

---

# **PeridexisErrant's DF Walkthrough Documentation**

***Release 0.2***

**PeridexisErrant**

**2018-05-21**



---

## Contents

---

<b>1</b>	<b>The Walkthrough</b>	<b>3</b>
<b>2</b>	<b>Tutorials</b>	<b>99</b>
<b>3</b>	<b>Masterclass</b>	<b>169</b>
<b>4</b>	<b>Other Information</b>	<b>187</b>
<b>5</b>	<b>About</b>	<b>199</b>





The learning curve for Dwarf Fortress has been described as a spiked overhang - but many of us have fought our way up, and have more fun than any other game around. This walkthrough, along with the [DF Starter Pack](#), are my attempts to dangle a rope down, as the community first did for me years ago.



# CHAPTER 1

---

## The Walkthrough

---

Dwarf Fortress is a rare game: every time you start a new game, an entire world is created anew - everything from geology and weather, to the patterns of trade in wheat and wars between jealous kings. The fact that there is no script, no standard storyline that every player goes through, makes a traditional game walkthrough impossible.

Instead this is more like a tutorial campaign in a strategy game - right down to playing on the same map, if you download the suggested package. You're free to follow along exactly, but also encouraged to experiment (and maybe back up your save occasionally).

Remember as you play that no matter how skilled you become, without a winning condition your fortress will inevitably fail - then pick yourself up, reflect on what you learnt, and recite the motto: **Losing is Fun!**

## 1.1 Getting Started

Dwarf Fortress is an extremely interesting and complex game, but one which people find exceptionally difficult to get into. Although it isn't hard to find tutorials, many assume some basic knowledge of the interface and what the player is looking at. They also don't take into account different tilesets and versions and the difficulty new users have in finding a good site for that first fortress.

But all that is in the past Dwarf Fortress Walkthrough! It comes with a pre-made world and save and full game download for you to play along with. And a ton of screenshots. Yes, many, many screenshots.

So grab a drink, get your thinking cap on and be prepared for some dwarfy fun! And carp. Never forget the carp...

### 1.1.1 Set Up

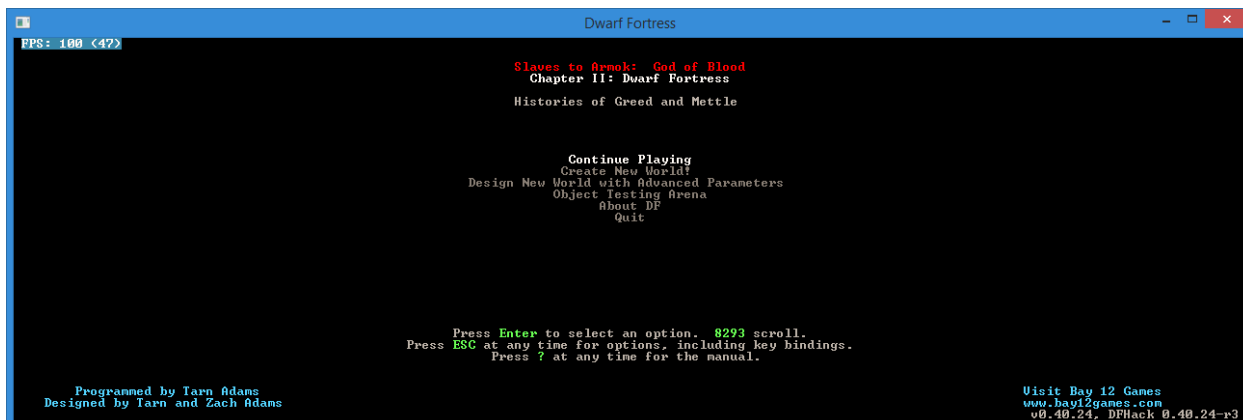
[Download the walkthrough pack here.](#)

This tutorial uses a pre-configured install of DF, with the same save and all the extras you need - but no more. If that's not going to work for you, see the guide to *Setting up your install*.

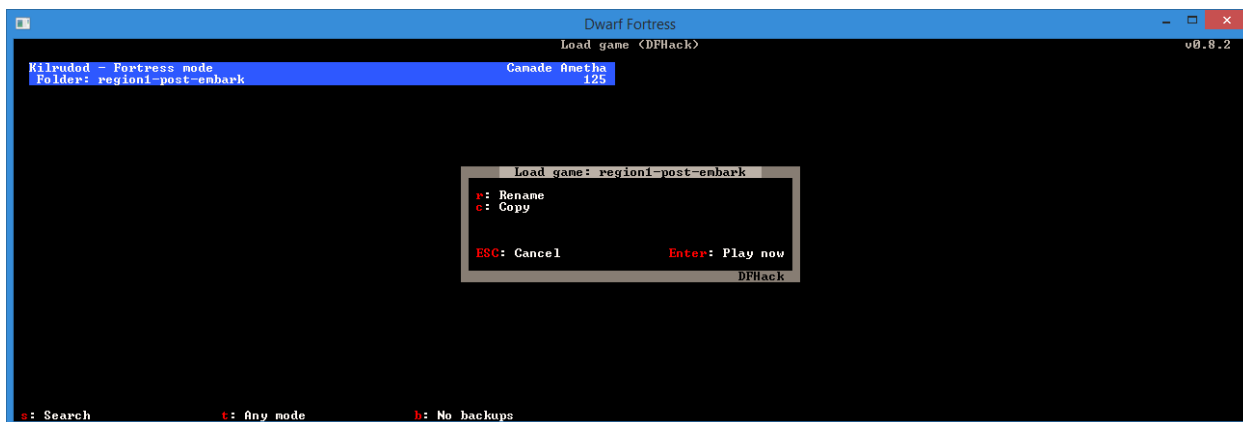
After downloading the pack, you need to unzip the folder onto your hard drive (but *not* in Program Files). Then create a shortcut on your desktop leading to the launcher or directly to `Dwarf Fortress.exe`.

### 1.1.2 Run the Game

Now we're ready to get going! Click on your new shortcut, hit "Play" if you're using the launcher, and you should get the screen below.



So, you've got the game running and enjoyed the exciting ASCII intro movie. Well done! I can tell you'll go far! Next step is to hit **Enter** on *Continue Playing*, which should be highlighted. You'll then be presented with this screen.



This is the save game list, which gets cluttered very quickly as seasonal saves build up. But for now all you have to do is use the down-arrow key (not the number pad arrow key) to select *region1-post-embark*. Hit **Enter** and the game will load.

### 1.1.3 First Look

Once the game loads you will be presented with this screen. Although it's confusing on first glance, don't panic! It will all be made clear shortly. First thing though, hit **Space** to pause the game. **Esc** backs out of any menu you're currently in.



Lets talk about what you can see. First up, the screen is divided into three sections: The left shows the local view. The center shows the action menu, the right shows the area map. To make things a lot neater hit Tab, which cycles through various menu arrangements. Stop when your view matches the one below.



Now we're really cooking with gas! Soon you'll be melting foes with magma-falls, and drowning goblins in artfully engineered traps! But before then, lets look around. Use the arrow keys to navigate around the local area. You'll notice the we've got a decent amount of space to work with.

While you examine your surroundings you should be able to spot tree trunks and the grassy areas easy enough, same with (frozen) ponds and the stream and bushes. Other tiles won't make much sense, these tend to be the slightly-arcane





You'll see that what was black dirt is now grass and trees. It's like we've gone up a floor in a lift and we're no longer looking at a slope, now we're looking across a grassy plain. Below us is open space and tree tops. Make sense? Now, try going back down to 108 and then down to 107, a level below "ground" level.





Your view should look something like this. We're underground now with mostly dirt around and earth between the pool, the tree roots, and the rest of the ground. You may have noticed the pool doesn't appear to have changed much, but don't be confused. The difference is that on this lower level we're on the same level as the body of the pool itself, at the level above, we're actually above the pool a little looking at the top of it.

Continue having a look around. When you're done, return to Z-level 108 and find the wagon with our dwarves. The helpful graphic below explains what you're looking at.



### 1.1.4 Lets get digging!

So now what do we do? In Dwarf Fortress we're usually trying to get a bustling community of dozens of dwarves up and running, while fending off goblins, the whims of nobles and assorted other evils. To do this we need to build ourselves a mighty fortress! Preferably underground. So lets find a cliff face to dig into and get this show on the road!

If you go west (left) from where our dwarves are (using the arrow keys) you'll see a nice cliff face. Lets dig an entrance into this spot and plan to eventually dig down way underground where it's safe and cosy. First up, get the map centered around about where we want to dig (shown on the screenshot below). Once you're ready:

- Hit the `d` key. You'll see the menu on the right changes and in the local view a yellow X has appeared. The menu on the right should show the *Mine* option highlighted. If it doesn't, hit `d` again, to select it.
- Move the cursor to the edge of the cliff, like this:





- Now hit enter and move your cursor down with the arrow keys. You'll see that you've dropped a flashing "anchor". This is going to mark out what your dwarves will dig. Move it across 20 and up 3 and hit enter, your screen should look like this:



The browned out area shows where your miners are going to come along and dig. But they won't act until you back out of the *Designations* menu as the game is paused. Hit `ESC` and you should see the game unpause and the menu reset to its master list. Oh, when you have the `d` menu up you can actually click on the map with your mouse and select areas to mine. Some people prefer to select areas this way.

With the game unpaused you should notice two dwarves race to the cliff face and start digging. With our entrance under way we should also think about laying out some rooms for our dwarves to live in, who wants to spend time out

under that hot yellow disk when lovely rock and earth beckon!

Right now we're not worried about making our fortress perfect and creating the strongest entrance, we're simply trying to scratch out a space to live! To that end we'll need three or four rooms off this main entrance.

See if you can match the layout below by marking out areas to dig, as you've learnt. If you make a mistake you can hit `x` from the *Designations* (`d`) menu and you'll note that the menu on the right has *Remove Designation* highlighted. Now when you hit enter and select an area, any area set for digging will be cleared of that designation. Anyway, enough detail, on with the room building!



Look at the little dwarves go! Aren't they industrious! They're quickly digging away and leaving a lot of dirt floor behind them. Good lads! Lets leave them to their work while we sort out some other important jobs.

### 1.1.5 Chopping trees, setting up piles

Dirt and rocks are handy, but so is wood to make nice dwarven beds, and we're short on it right now. To get wood, we need to chop down some trees. And we do that by designating an area of trees to be chopped. Lets get cracking and clear the trees in front of the entrance.

Designating trees to chop is like designating rocks to be dug, so lets decimate the local environment!

- Move the map around so you're looking out front of your fortress.
- Hit `d`. Hopefully this time you'll notice the game has automatically got *Chop Down Trees* selected, if it doesn't, hit `t` and it will be highlighted.
- Move the cursor to the upper left, hit *Enter*, and move the cursor to the bottom right, hit *Enter* again. Any trees in that area will now be set to be cut, as indicated by them all being marked brown.

Although you've done well neither man nor dwarf can live on wood and dirt alone! Luckily for us there are plenty of bushes out there loaded with ripe berries. Lets set some of them to be picked:

- Hit `d` again and then `p`. One the right *Gather Plants* is now selected.
- Designate about the same area as we did with the wood cutting.
- All the bushes will be highlighted now, and when you unpause you should see a plant harvesting dwarf join the woodcutter.

Your view now should look something like this, note the brown 'highlights' and fallen logs:



Now lets sit back and watch the dwarves work for a minute! It won't take long before our miners have cleared out our temporary living quarters and our woodcutter and plant gatherer have begun their work. Once the interior space is clear we have lots of work to do, so lets break it down into useful chunks.

### 1.1.6 Our First Farm

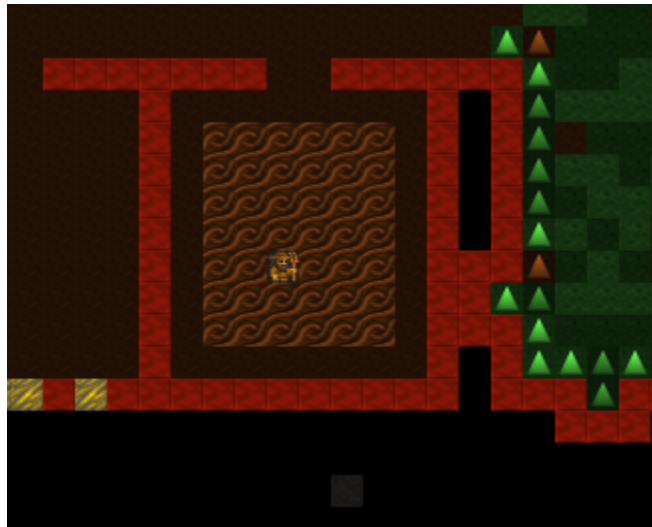
Farming is how you will make most of your food in Dwarf Fortress, and it's important to get farming quite quickly. If food runs out your dwarves will starve and your game will end prematurely, and we don't want that, right? dwarves usually farm underground and handily, we have a space all prepared. So follow along, and lets get some tasty mushrooms growing for our stumpys to chow down on!

- Find the bottom right room.
- Hit `b` for the *Building* menu and then `p` for *Farm Plot*.
- Move the cursor to your farm area.
- The screen instructions tell you how you can increase the size, we need a good 6 by 6 plot, so hit `u` and `k` a few times until you've got a big green grid, like the one below.

- Move the grid around with the arrow keys until it's in about the same position as the one in the screenshot. If you get the size wrong, `h` and `m` will reduce the plot.



- Hit `Enter` and the green area marker will change to a flashing brown field marker. When stuff flashes it indicates that dwarves are on their way to come build the construction, in this case, a field. And lookee-here! A farmer has come to build our field for us!



He will take a short while to build your field. When you see the dwarf run off, you know he's done. Now it's time to set the field to grow delicious plump helmets all year round.

- Hit the `q` key. This is the key for *Set Building Tasks and Preferences*. Again you'll notice a cursor on the local

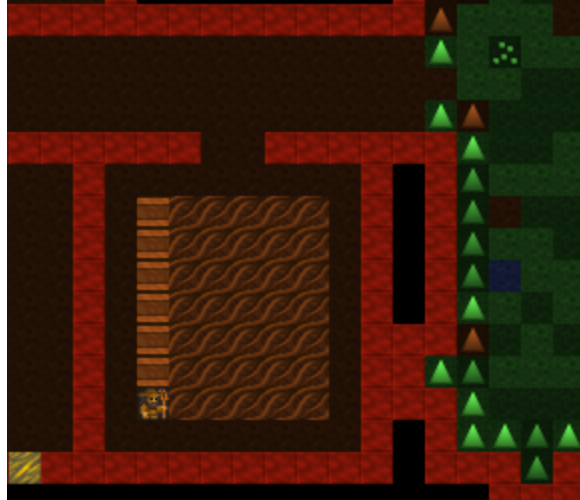
screen and if it's anywhere near the field, the field will be flashing. You can move that cursor around with your arrow keys, this is useful later on when you need to select different workshops and buildings.

- You'll note that on the right the menu now shows you specific task details associated with that field, like so:



- What we want to do is set *Plump helmets* as the food to be planted every season. You can see on the top right of the menu we have the various plants we could try and grow and in the middle area we have the different seasons. Right now *Dimple cups* are selected (but not highlighted, note) and the season is set on *Spring*, as it's Spring currently. We need to change these settings.
- Now comes an important skill! Menu scrolling! Using = (not the down-arrow), scroll down through the list to *Plump helmets* and hit enter. You should now see *Plump helmets* highlighted. By the way, -/= is the way to scroll in menus; DF usually uses - and +, but having to use the shift key gets annoying.
- Good work! But right now we've only set the spring planting. It's time to set the planting for Summer, Winter and Autumn too. To do this, with the field task menu up, hit b for Summer and then scroll to *Plump helmets* and hit enter again. Hit c for Autumn, set *Plump helmets* and then d for Winter, repeating your selection. You may have noticed that what you can plant changes with the seasons, but don't worry about that for now.
- Hit Esc to back out of the menu and resume the game. Your dwarves will quickly begin planting in your new field, well done!

Now we've got a field down hopefully none of your dwarves will starve, go crazy and resort to eating rats, or each other. Pretty soon you'll have a few seeds in the ground and your farm will begin to look a bit like this:



Oh, by now you may have noticed the announcements along the bottom of the screen. This is generally because the game wants to let you know that something important has happened. You can hit space to let the game resume, or hit a to see the alert if you miss it at the bottom of the screen). Hitting `ESC` will take you back out of the alerts menu and resume the game.

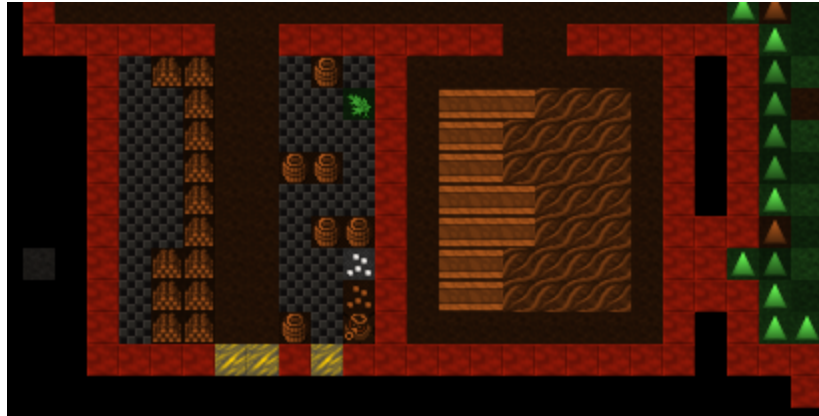
### 1.1.7 Stockpiles

While you can just leave all of your dwarves' stuff strewn around the countryside, it's much more efficient to have it all inside near where it's needed. You may have noticed your farmer dwarves running in and out of the fortress to grab the seeds that they need (when they flash between their icon and a little red dot (the seed icon) you know they are carrying seeds).

This is where stockpiles come in. They make everything more organised. A stockpile for all our food and seeds right next door to the farm would probably be pretty handy, right? We don't want the dwarves running across the map to get a bite to eat or a seed to plant, so lets make a food stockpile inside!

- Find the room opposite the farm.
- Hit the `p` key (for "piles" of course), a cursor will appear.
- Hit `f` to select *Food*, we want to make a food stockpile, after all.
- Move the cursor to the bottom right of the room, hit enter, move it to the top center and hit enter again. You have now set that space as a food stockpile, well done!
- Hit `w` to make a *Wood* stockpile, and place it on the other side of the room.
- After a few seconds your spare dwarves will start moving food inside, you'll see barrels and bags being hauled and pretty soon the stockpile will look something like this:





You don't have much room inside the fortress yet, so no more piles for now. In time you're going to want to put stockpiles everywhere to help manage your production and resource gathering. For now, well done, you've got food production up and running, you've set up a stockpile, you've set wood to be chopped and plants to be gathered and you know how to do some basic digging! You've come far!

Before reading Chapter 2, how about setting some more plants to be gathered and more wood to be chopped. Then, head [to the next installment](#) and we'll learn all about workshops, bedrooms, dining rooms and stairs! It will be ever so exciting, I promise!

## 1.2 Workshops, Furniture, and Stairs

Lets continue on with our exciting and fascinating Dwarf Fortress Walkthrough! In this section we'll work on getting some workshops set up and we'll dig out a lot more space for our shorties. There's lots to do, so lets get cracking!

### 1.2.1 Building a Workshop

Workshops are a central feature of DF so lets get to grips with them right away. Workshops are places your dwarfs work, usually turning base materials into useful goods. The list of workshops can be a little overwhelming (b then w), but don't worry, we only need to worry about a couple early on.

First up we need a carpenter's workshop, this workshop allows us to turn wood logs into furniture and other items. You should see a lot of wood scattered around in front of the fort, where the wood cutter has left it after felling trees. We have no wood pile for it to be moved to, you see.

Ideally, we'd like to build the carpenter's workshop inside, but we haven't dug a nice enough space for the workshop yet, so lets chuck it outside for now:

- Move the view to the front of the fortress.
- Hit **b** for *Building*, and then **w** for *Workshops*. You can also just scroll down to *Workshops* using **=** and hit **Enter**.
- You will now see a list of workshops. It scrolls off the bottom of the page (scroll around if you like)! Hit **c** for *Carpenters workshop*.
- The menu will vanish and the placement menu will appear. The green X's mark out the workshop's footprint. The dark green Xs are squares that will be impassible once the workshop is built, keep this in mind when building in an enclosed space, you don't want to block the door!



- Chose a spot in front of the fortress. Somewhere around where mine is marked should do. Once you're ready to place your workshop hit **Enter**.
- The menu on the right will now change to a list of items you could use to build the workshop with. Mostly it will be a list of wood and it will look something like this:



- At this point you could just hit enter and the first item on the list will be used to build the workshop. Instead, for fun, hit **x** and *Expand* the view to show a list of every single item the workshop can be built from. Expanding the item selection is a nice way to choose exactly the item you want to be used. Most of the time you can ignore this option, but it will come in handy later on.
- Close the expanded menu with **x** again. For your interest note that items are sorted by distance from the workshop's current location. Usually you want to build with the closest items to save your dwarfs a slow walk.
- Once you've hit enter (selecting the building material) the workshop will be placed in an un-built state on the



map. It will look like this:



Your workshop is now about to be built. You will have to hit `ESC` a couple of times to un-pause the game, but as soon as you do I bet you that a helpful wee dwarf will run over and start building it. If you're quick you can hit `q` and see the status of the workshop, it should say *Construction initiated..*

A few seconds later, the workshop is built!



It doesn't look like much, but it's really useful! I promise! So lets get turning some of this lumber into stuff our dwarfs can use. Just don't tell the Elves, ok? They really hate us chopping down trees. Pointy-eared gits, a magma-steam death is too good for them!

### 1.2.2 Making beds

Dwarfs don't like sleeping in the dirt, they like nice, comfy beds - and solid gold statues in their bedrooms, but those will come later.

Follow along with me:

- Hit `q` and move the cursor over the carpentry workshop.
- The menu on the right now shows you the actions you can take, specifically, we're interested in a *Add new task*.
- Hit `a` (duh)!
- Note the right menu is now a large list of stuff we can make. We're after beds.
- You can scroll down to beds using `=`, or simply hit `b`.
- The menu now changes back to the `q` list and you'll see *Construct bed* listed. But we want more than one bed!
- Hit `a` again, and then `b`. Repeat until we've got five beds up there. We don't need a bed for every dwarf just yet, luckily for us they seem happy to sleep in shifts.

Your carpentry workshop menu should now look like this:



If your workshop menu looks like this, well done! Five beds are queued up and hitting `Esc` a couple of times will un-pause the game and set a dwarf working, building your beds. You may have noted that there are a lot of other workshop options available to us now we have items on the build list. We can `s` suspend any construction, set any item to be `r`, repeat built, or we can hit `c` and cancel the construction of the current listed item. We can also have the entire workshop removed using `x`. For now, let's just watch the beds get built!

Note, you won't see the beds being put anywhere. Want to know why? The answer is simple, and if you're extra-special clever you may have already worked it out: We have no furniture stockpile! Let's fix that:

- Hit `p` from the main menu.
- hit `u` for *Furniture Storage*.
- Move the `X` near the workshop, hit `Enter`, and designate a pile about 5 tiles by 5 tiles.

Here's my pile after a few seconds of the game resumed:



You'll note the dwarfs have moved the constructed beds to the pile as well as some spare barrels, our anvil and a bag. As stated above, we don't usually like to leave our stuff lying around outside, even in tidy piles, but as a temporary solution it helps keep things organised and gets the furniture moved from the workshop. If items aren't pulled out

from under the carpenter's feet then eventually the workshop will get cluttered (marked with *[CLT]* when you *q* over the workshop) and a cluttered workshop works slower.

Later on we can remove piles by using *p* and then *x* and selecting the whole area of the stockpile. What's better is that helpful dwarfs will grab those items and move them to another suitable pile. But enough of that, lets get on with building our fortress!

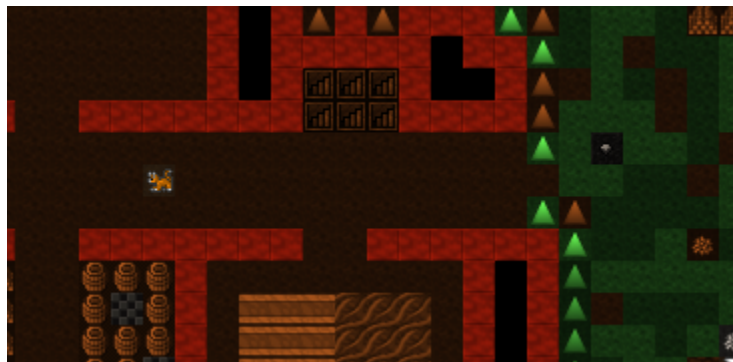
### 1.2.3 Digging down, down, down!

What kind of Dwarfs would we be if we only scratched around on the top of this mound like dirty hobbits? We want to dig down, down, down! The easiest way of doing this is to build some stairs. Well, to be precise, we're going to designate some stairs using the *d* menu. We are going to designate some un-dug space to be stairs and then see what's going on a few metres down. So follow along, and lets get expanding!

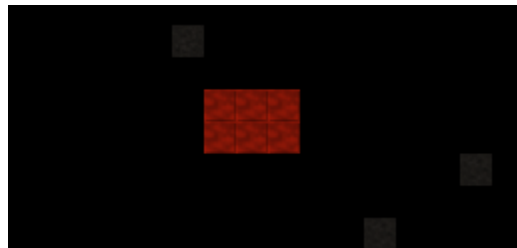
- Move the map to the top side of our corridor.
- Hit *d* and then *j* for *downward stairway*.
- Move the *X* into the black and hit enter, move the cursor down one, and hit enter again.

You should see six downwards stairs marked. You may be wondering why I have set six stairs down. The reason is that dwarfs need space to move past each other. If the corridor, or stairs, are only 1 wide, then they have to pause to let each other pass. This slows movement around your fortress down terribly once you have dozens of dwarfs running around. Six stairs will be plenty for a long time.

Hit space until the game is running and let that area get built. It should look something like this when it's done:



Super! A hallway and two downward staircases! At this point you may be feeling pretty chuffed with yourself. But if you go down a level (*>*, you'll recall) you'll not see any stairs going up, or much of anything, just these tan blobs:



Where are our stairs going up? Where are our vast stores of mineral wealth? Well, this is where stairs get crazy-strange. Let me explain...

To dwarves a down stair is just an exploratory stair dug down to see what's on the level below. It's like knocking a hole in the floor so you can look down on your neighbours in the flat below, hoping perhaps to work out what the weird noises are.

To actually go down to the level below, you need to build stairs going up from the level you're digging to. Or in our case, we want to build an *Up/Down Stairway* as we are looking to dig down a long, long way. An Up/Down stairway pokes the proverbial holes in both directions. It links up with a stairway above (if there's one there) and tries to link up with a stairway below (if there's one there). If there's no stairway to link up with it will provide access to those other levels so you can build the needed stairs.

Here's a handy side-on diagram:



To build an up/down stairway, do this:

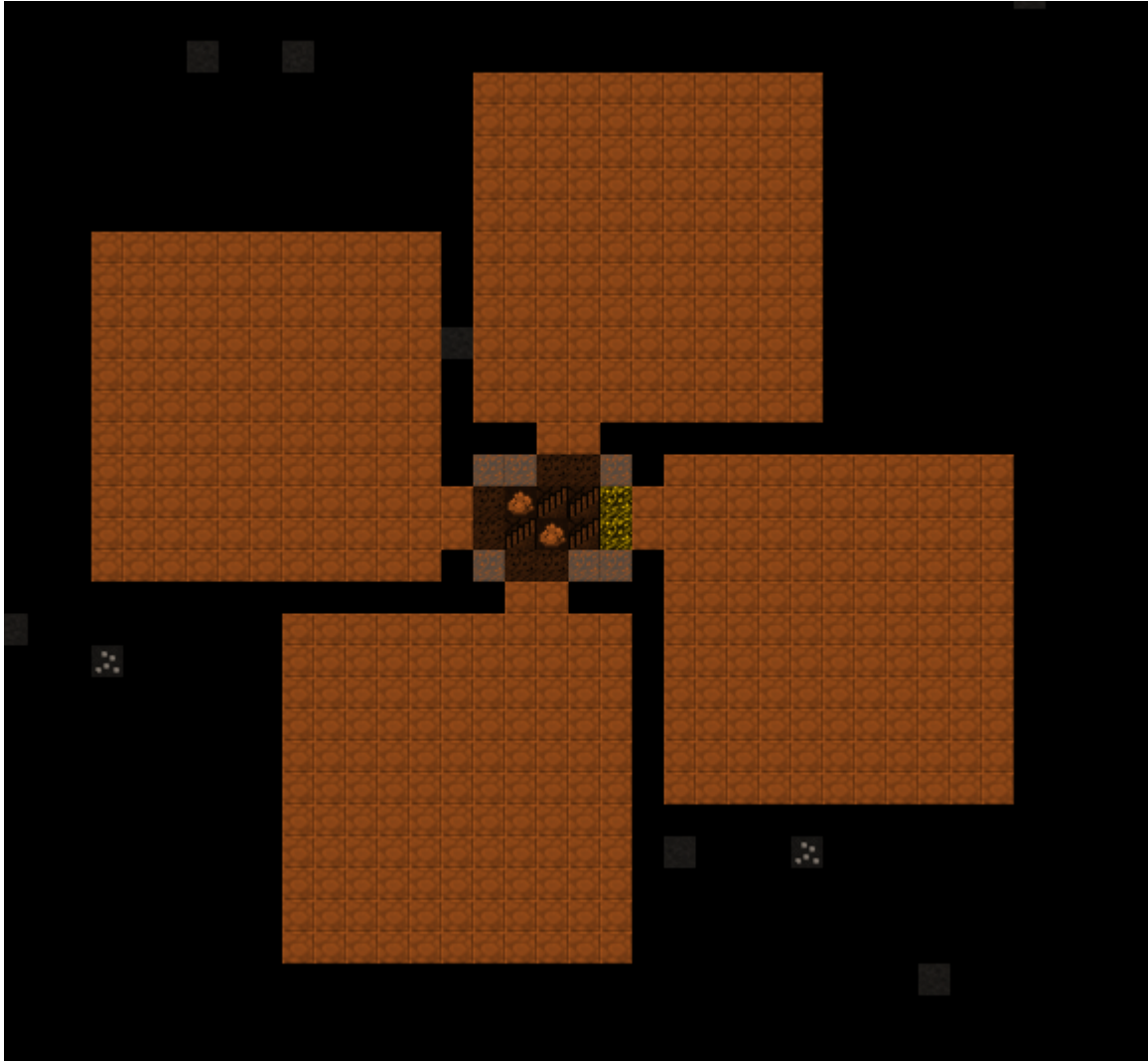
- Go down one level from our main area.
- At this level, below our down stairs, hit `d, i` to designate an *Up/Down Stairway*.
- Move the X over the tan blocks. Hit `Enter`, move down one level, and hit `Enter` again.
- Brown X's now show that your up-down stairway has been designated.
- If you mess it up, `d` and then `x` can un-designate the space for you.
- Resume the game and the up/down stairs will be dug. The designation looks like this:



Lovely! Perfect up-down stairs. But we're not done yet.

So far you've been selecting two corners to designate a rectangle in two dimensions - but laying out a staircase would get pretty tedious like that! Luckily DF supports three-dimensional designations as well, so lay it out in 2D, then `>` to level 103 to finish the column.

Rock! We've got some rock around us! This is the sort of place a good dwarf loves! How about we dig out some of this space? I suggest a set of square rooms just off the staircase, with two tile wide doorways. It's a simple layout, but enough for now. See if you can match mine.



Now let your boys dig! Dig boys, dig! And make sure all your areas set to be dug connect back to your stairs!

Along the way you may get various messages about striking various ores. That's good news! Metal is very handy, as you can imagine. But we'll worry about metal later.

### 1.2.4 What's that on the ground?

While they're busy digging, why not have a look at what is scattered all over the ground. To do that, from the main menu, hit `k` to look around. An `X` will appear and you can direct it using the arrow keys. As you move it around you'll see what's under the arrow. Check my example:



As you can see, underneath my X is a dwarf, some limestone, and a limestone cavern floor. I can use = to scroll down through those three items, and hitting `Enter` will display me some info about each. Ok, except for the floor. I mean, what do you want to know? It's a floor!

*Look around* is a very handy way to see what is in a square. Sometimes items get stacked a few deep and you're not sure what is what, `k` will show you. Also, it is very handy way to find out what the walls and ground are made of. For example, without irrigation you won't be able to build a farm plot inside on anything but soil, sand, silt, clay or loam. *Look around* will show you what the ground is made of.

Toy with `k` for a while, and see what minerals are being dug up. Then lets leave the dwarfs digging, we've got a very important job to do...

### 1.2.5 Chucking out the trash!

By now, believe it or not, some rubbish may be building up in your fortress. I've taken a shot of some rubbish near our food pile. Have a look at those slimy bones in with the barrels of booze:

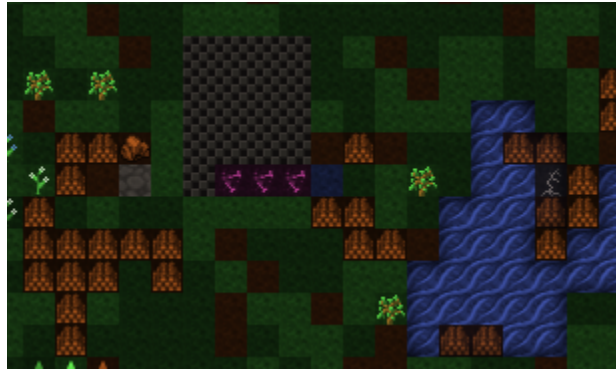


Using `k` I can have a closer look and see exactly what this trash is... *Rat remains*. Yuck! We can't have this left lying around our tidy fortress, can we? Well, we could, but the end result would be a stinking cloud of purple miasma.

Miasma makes our dwarfs unhappy. While they may never shower and probably reek to high-heaven, they really don't like the smell of rotting leftovers. Go figure.

So how do we get rid of the refuse? Easy! We build `p` *Stockpiles* for `r` *Refuse*. But note, we need to build this pile outside or we'll get miasma build-up. Outside, the wind blows the stink away.

See if you can set up a refuse pile on your own like mine. It's just like making any other pile, except we set `r` for refuse as we plot it.



Look, some of the refuse has already been moved! Phew, miasma criss averted!

But we still have one really, really important job to do. . .

### 1.2.6 Making booze!

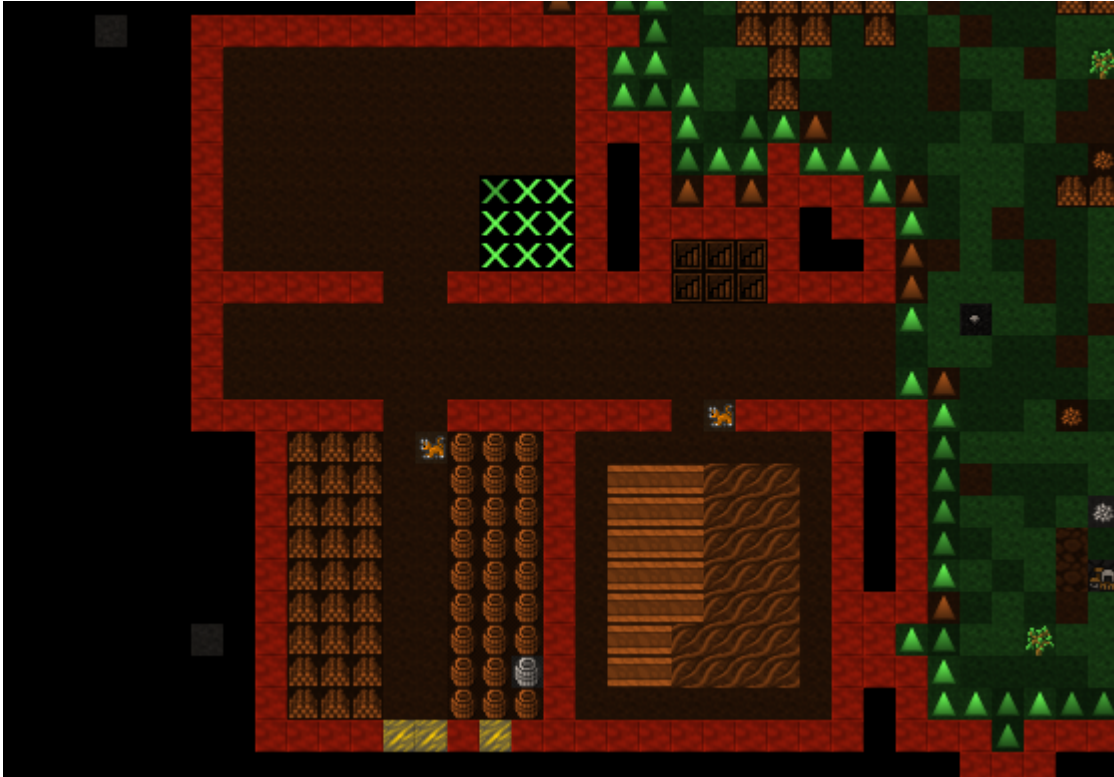
Booze is the lifeblood of dwarven society. Literally so. Dwarfs, unless injured, will prefer to drink booze over water 100% of the time. If there is no booze, they will drink water, and that's terrible - they work slower, and are more prone to murderous tantrums.

So what do we do about the dwarf booze demands? Simple! We build a still!

Follow along, dear reader:

- Hit `b` for *Build*.
- Hit `w` for *Workshops*.
- Hit `1` for *Still*, or scroll through and find the still yourself, and hit enter.
- Place the still in the room above the food stockpile, like this.





Placing it near food makes sense, as it will turn food items into booze. The less walking for our brewer, the better. Once you've placed it and selected the materials (any will do), your still should be quickly built and look something like this:





Before we brew some booze, there's one more thing we need: barrels.

- Head to the carpenter.
- Hit `q`.
- Hit `a`.
- Scroll down to *Make wooden Barrel*, or just hit `v`.
- Fill the queue with barrels.

Now to make it start churning out the brewskies!

- Hit `q` and move the cursor to the still.
- Hit `a` for *Add new task*.
- Hit `b` for *Brew drink from plant*
- Hit `a` and `b` another 7 or so times.

Yay! You have now queued up a lot of beer to be made.

To keep booze production at an acceptable rate, there's a couple of options. One is to keep checking manually, but you only have to be late once to cause a tantrum! The second is to set up repeating jobs (`q`, select job, `r`), but then it's easy for production of drinks and barrels to get out of sync and waste valuable materials.

The third option is a plugin called `workflow`, which lets you set a target amount and will suspend and unsuspend production jobs to keep you at that level. `q`, select job, `Alt-w`, `A` to *Add limit*, `R` adjust the range. This might seem complicated, but setting 50-100 drinks and 10-20 barrels will keep things flowing smoothly until your population is a lot larger.

End result, happy dwarfs! An important thing to note at this point is that brewing alcohol uses plants, but doesn't destroy the seeds. Cooking does destroy the seeds of any plant cooked, which can seriously compromise your ability to grow more food! So for now, let's just stick to making some extra booze. We should be ok for food for a while yet with our farm running, since dwarves love raw mushrooms.

Time to head back down stairs to see how our miners are going. Let them dig at least one room out before continuing with the next step. Hopefully they're not sleeping on the job (The big `Z` flashing on them)!

### 1.2.7 The World of Doors!

We need doors. Lots of doors. We need beds, doors, chairs, tables. We need lots of them. If you're starting to feel like an Ikea salesman, don't worry, when you see a bed menacing with spikes of cat leather, engraved with an image of a dwarf striking down a goblin with other dwarfs laughing, then you'll know you're no longer in Sweden. So how are we going to get all of this wonderful furniture?

Well, we could build some doors and assorted other stuff in our carpenters' workshop. But that would use valuable wood. Much better would be to use all that stone you can see strewn about. So how do we do that? I'm glad you asked! We build a masons workshop. Here's how:

- Hit `b` for *Build*.
- `w` for *Workshops*.
- `m` for *Masons Workshop*.

Chose an area near our main stairs up, in one of our new rooms. You can check the shot below to see where I'm going to place mine.



- Select some stone and get the thing built!

Once it's up it's time to get it producing some useful items:

- `q` over the Mason's Workshop.
- `a` for *Add new task*, then `d` for door. Do this four times.
- `a` for add task, then `t` for table, do this twice.
- `a` for add task, then `c` for chair, do this twice. Note that in DF, a chair made from stone is called a *throne*. They're not the only thing to have different names depending on the material, so if in doubt check the wiki.

Wee! Look at all that lovely furniture queued up! Soon our mason will turn up and start cutting blocks of stone into something more useful. Unfortunately, someone is going to end up trying to haul all that furniture up stairs to the furniture stockpile, and we can't have that, so let's make a big stockpile in the middle of this room.

While you're at it, why not remove the furniture stockpile upstairs and get all of that stuff out of the rain. Go up to the pile, hit `p` for pile, then `x` and then hit `Enter` at one corner of the outdoors furniture stockpile, and then move the cursor to the other end and hit `Enter` again. Here's my downstairs stockpile, complete with some just-moved furniture and a mason hard at work!



Now it's time to get those doors and beds into use, and when the tables and chairs are built, we'll use those too!

### 1.2.8 Home Sweet Home

You may have noticed that our dwarfs have been sleeping on the ground when they get tired. This is really not much fun for them and we'd like to give them a place to stay. With some beds built, let's set up a big bedroom on the top floor so our dwarfs can get some shuteye.

- Head to the top floor, and dig out a small room at the end of the hallway.
- Hit `b` and then `b` again (for *Bed*).
- Move the cursor and place a bed in the corner (hit `enter`, select a bed and hit `enter` again).
- Spread the five beds around the room.
- Hit `ESC` until we're back at the main menu.
- Hit `b` again, and `d` for *Door*.
- Place doors across the room entrance.

Here's my room layout. The shadowy beds and door show that the beds and doors aren't placed yet. The green X shows me about to place the next door.

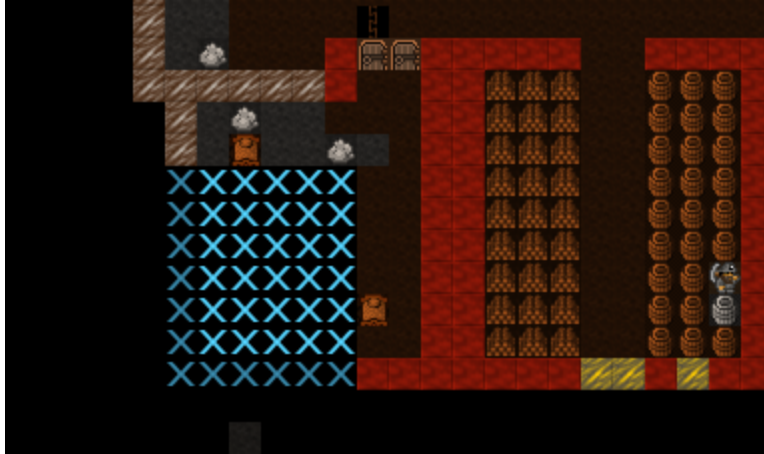


Pretty soon the room will be laid out and we'll be ready to use it to house our little workers. Here's mine finished, doesn't it look pretty?



But the dwarfs won't use it as a bedroom yet. We have to specify what the room is to be used for first. To do this:

- Hit `q`.
- Move the cursor near one of the beds. It will start flashing green.
- On the right, you'll see the option `r Make Bedroom`.
- Hit `r`.
- You will now see a flashing blue square. Here's mine, as selected from the bottom-left bed.



We could hit enter now and set the room at this size, but that would leave two beds out. So we need to make the room size bigger. With the blue square flashing...

- Hit `=` and this will make the blue square bigger until it fills the whole room.

You will note that the room won't "leak" beyond the boundaries of the walls and doors. This is why we need doors, to prevent leaky rooms! Makes sense? Right? Ok... moving on...

- Hit `Enter` with the room set at max size, a new menu will appear on the right.

This menu gives you options for the room. It will always appear when you `q` over the item you set a room's use from. Note, you don't have to set every bed in the room as a bedroom (although the game will let you do that). DF is smart enough to know that the room is a bedroom, and all the beds in the room should be used.

On the new menu you want to:

- Hit `d` for *Dormitory*, this will turn the *(N)* to a *(Y)*.

With the room set as a dormitory any dwarf without their own room will use the beds in the dormitory to sleep in. When you have a military, a *Barracks* is where dwarfs will spar and sleep when off duty.

Well done! We have a shared bedroom for our shorties to sleep in!

In *the next chapter of the walkthrough*, we'll build a fine dining room, set up some more workshops, and start to build some proper living quarters for our dwarves. I can't wait!

## 1.3 From Cave to Home

In this tutorial we'll set up some more living space for our dwarves as well as more workshops and stockpiles. This will just be a short one as I'm going to give you some tasks to do on your own at the end. Are you up to the challenge?! Lets do it!

### 1.3.1 Recap time

We've covered quite a lot so far, and if you've played through, and you're still reading, you're well on your way to being a master of Dwarf Fortress. Lets look at what we've learned:

- How to look around and work out what you're looking at.
- How to dig out space and how to dig stairs.
- How to set up rooms and workshops.

- How to get some basic resource production and gathering going.

One thing we haven't covered is saving your game! Perhaps an oversight?! Simply hit `ESC` and then select *Save Game*. But don't do it now! *Save Game* assumes you want to exit and it will take you back to the main menu. Whatever you do, **don't** choose *Abandon the Fortress* - that ends the game!

I should note that saves can get confusing - if you start to play with a seasonal autosave, then save that, the names get longer and don't always sort in a useful way. When you come back to DF and hit *Continue Game* choosing the last save in the list won't necessarily result in your most recent game starting. Instead, you may have to look in your `<DF>/data/save/` folder and see which folder has the most recently modified files. Then you might rename or delete the older folders.

If you choose the wrong save, there are two options: you can save as usual and return to the main menu, or force-close DF with the DFHack command `die`. Enter it in the ingame console (`Ctrl-P`), but be careful - this will quit without saving or checking again!

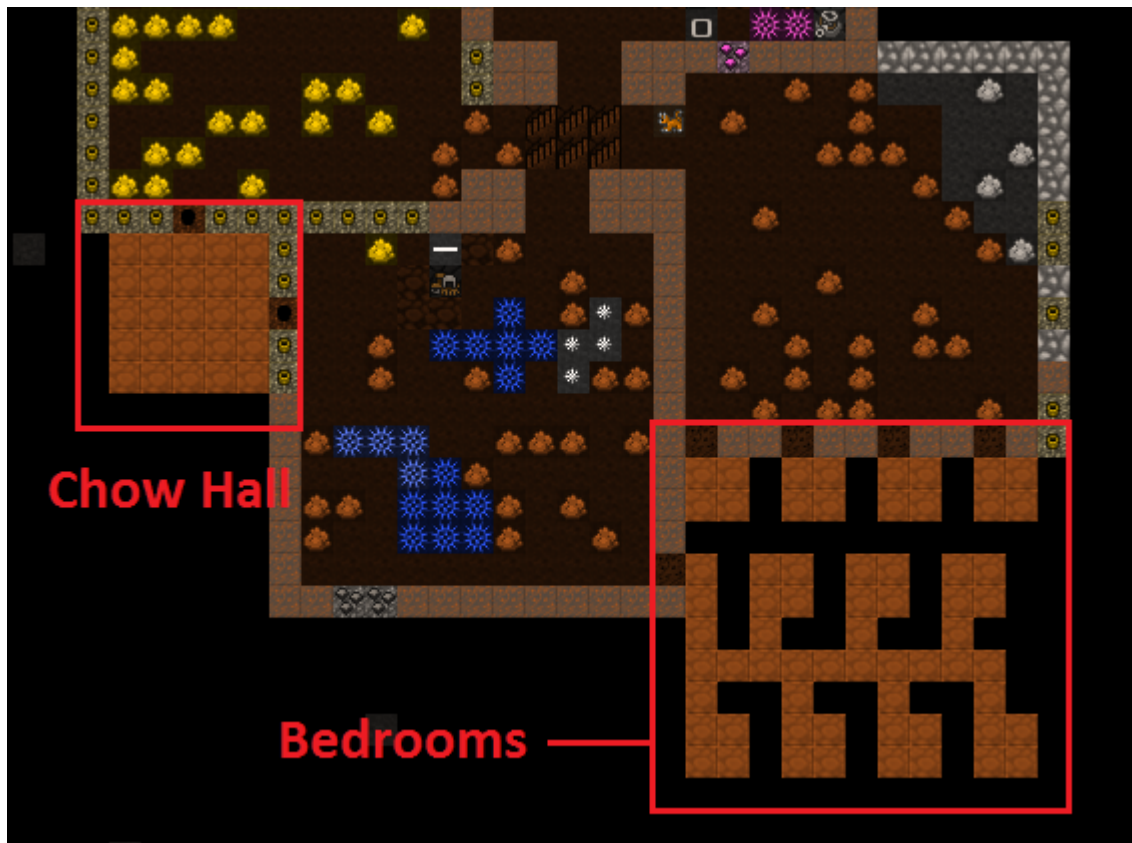
Good luck! And lets continue...

### 1.3.2 Making our Hole our Home

Lets look at making this place a little more comfortable for our dwarves. First up, they really need a nice place to eat. Just picking at food on the floor isn't much fun, dwarves want a beautiful hall in which they can quaff beer and eat cat biscuits (yes, you can make biscuits out of cats. Ugggh). Lets help them set one up.

For now we're going to set it up near our newly-dug space. I'm also going to make a bunch of 2 by 2 rooms, which will serve as permanent bedrooms for our dwarves. There's no such thing as a perfect design, so lets just get the function right and worry about the perfect layout in your next fortress.

This is how I set things up:





While it's being dug, go add a bunch of beds to the carpenter's shop, as well as doors and tables and chairs at the masons. Report back in when the rooms are dug and you've got a bunch of furniture and another couple of tables and chairs.

... Ok, are you done? Good!

Now you need to go and place a bed in each room, a door on each doorway, and tables and chairs in the dining room-to-be. You should know how to do all of this using `b` for *Build*, then `d` for *Door*, `t` for *Table*, and `c` for *Chair*... which in a Mason's workshop is called a *Throne*. Yep, the same object can have different names depending on the material - another reason using the shortcuts can be easier than scrolling.

Here are my rooms partly completed. If you look closely you can see a slacker dwarf having a nap.



Can spot my chairs, tables, beds and doors? Pretty aren't they!? You may have noticed that bits of your local map are flashing. Don't worry, this just indicates that objects are sharing the same space with other objects, but yes, all that stone does make things look messy. If you have a tidy-fetish, GIVE IT UP NOW! Dwarf Fortress forts often look messy with stone and junk strewn everywhere. You can, however, do a few things to fix that, but we'll worry about that some other time. Meanwhile, we need to make ourselves a dinning room!

### 1.3.3 What's that? Oh bugger!

Just got a message which reminded me that I've forgotten to do something...

*The Outpost Liason from Vushuvash has arrived.*

*A caravan from Vushuvash has arrived.*

*Their wagons have bypassed your inaccessible site.*

*The merchants need a trade depot to unload their goods.*

Damn! A trade caravan arrived but couldn't make it to our fortress because, 1: it is inaccessible, 2: we don't have a trade depot. This may have happened to you already, if so, don't worry about it much, we'll get that problem fixed soon. Trade caravans come by fairly regularly and represent other civilisations wanting to trade their valuables for yours. They are important, and handy, but we'll discuss them later some time.

Oh, you may get various windows pop up from the visiting trade liaison. Just `ESC` back out of them for now.

### 1.3.4 Back to the eating place thing!

By now you should have the dining room furniture set up, so it's time to let the dwarves know it's the official dining room of the fortress. We do that in much the same way we set up bedrooms:

- Hit `q` and move the cursor over any one of the tables (and for good fun, move it over a chair and read what sort of rooms chairs set up).
- When over a table, hit `r`, and again you should see a flashing blue box which doesn't quite fill the room.
- Use `=` to expand the room to fill the dining room space.
- Hit `Enter`.
- Now hit `h` to turn the dining room into a meeting hall as well. The *(N)* on the menu will become a *(Y)*.
- `ESC` back out to resume the game. All tables and chairs in the entire space will now be used.

Well done! A dining room and meeting hall is now set up! Without a meeting space immigrant dwarves get confused and don't know where to go when they arrive, milling about at the edge of the map. A meeting space seems to send out invisible mind-control rays and any newly-arrived dwarves will immediately home in on it and into your fortress. Handy!

For amusement, go back up stairs and find the wagon we arrived with. I bet you it has a few dwarves hanging around it. Know why? Because by default your wagon is your first meeting area!

If you press `F1`, the map will move to the wagon - because it's also your first hotkey location. The `H` hotkeys menu allows you to scroll through the hotkeys, name each of them, and set one to zoom to your current location - which saves time once you spread out vertically. Try setting `F2` to zoom to our big underground rooms now.

But the wagon is outside and a long way from the fort and we'd much prefer our dwarves to be safe and sound within our walls. We should probably remove that temptation to stand outside and get killed by carp, elephants, monkeys, unicorns, skeletal whales, zombies, giant eagles, deer, goblins, etc, so lets remove the wagon. Hit `q` and move the `X` over to the wagon, then hit `x` and the wagon will be *Slated for removal*. This should free up a three logs (which the wagon is made up) and will prompt any lazy, slacker dwarves to head back inside!

### 1.3.5 Everyone gets their own room!

Dwarves love to have their own room, much preferring it to sharing a dormitory (which they will do by default otherwise), and conveniently we have set up a number of little rooms. Lets declare that they're bedrooms, so our dwarves will come and claim one:

- Go to your empty bedrooms, hit `q`.
- Chose a bedroom, moving the `X` over a bed until it's flashing.
- Hit `r`. The blue selection area probably fills the room.
- Hit `Enter`.

You can assign the bedroom to a specific dwarf, but unless you want to play favorites there's no need - if a dwarf wants a nap and doesn't have a bedroom, they'll claim one of the unowned ones. Once that happens, the bedroom status will now look something like this:





If you get confused about a bedroom (or any room's status), just hit `q` again and move around over each object. Of interest is the fact that you can assign one room to have multiple uses, for example, put a bed and table in the same room and specify that the room is both a bedroom and a dining room, but doing so reduces the overall quality of both rooms. Don't bother unless you have some clever reason to.

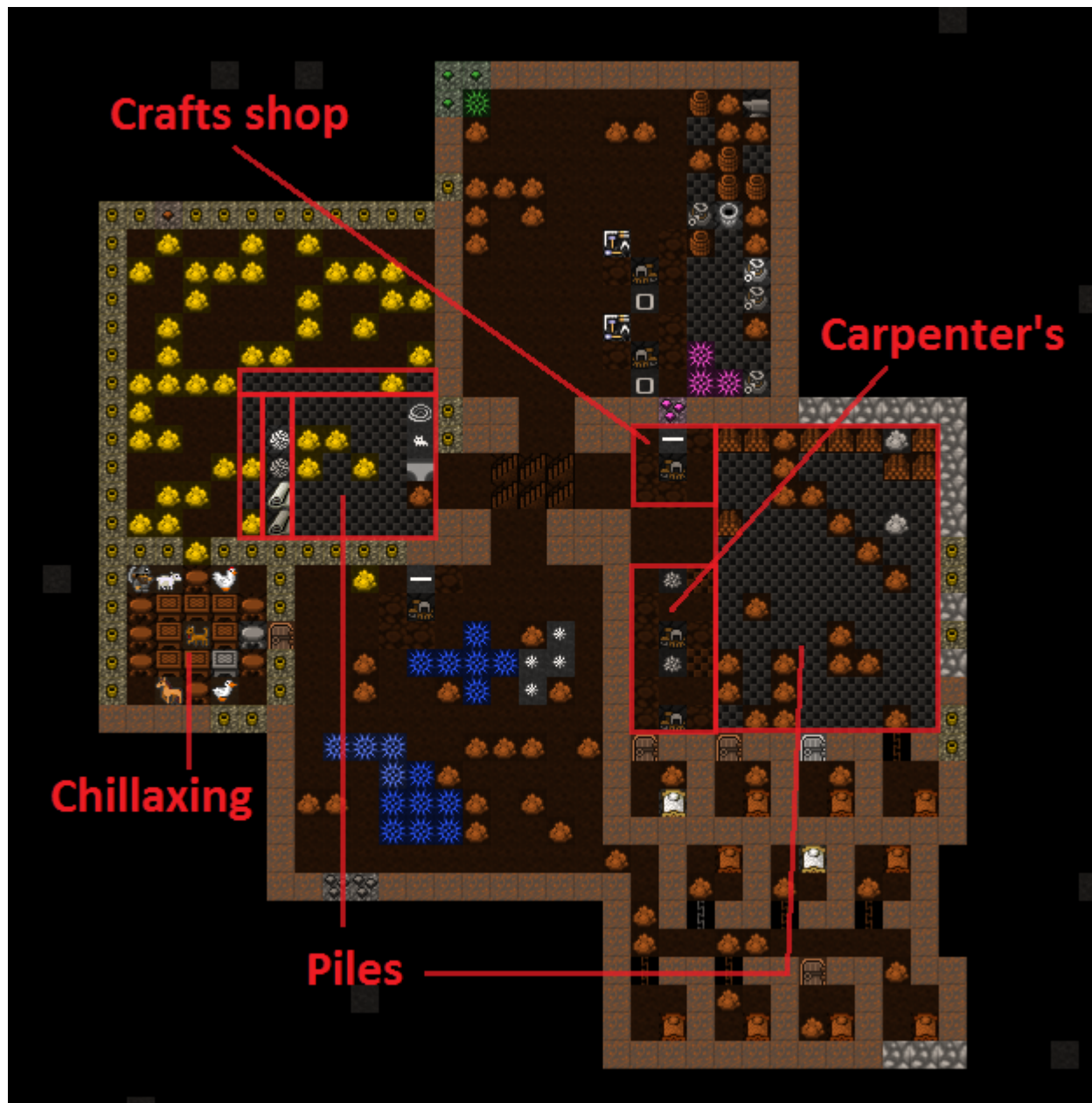
While you've got your cursor up, move it over the door and look at those options. You can lock doors and you can also make them impassable to pets. Don't bother with that for now, just have a look around.

### 1.3.6 Workshop fun!

Lets set up more workshops. First, put another mason's workshop in with the other one. It will be handy in the long run, I am sure. Set up a couple of carpenter's workshops in the room next door. While you're at it, set up a big wood pile (`p`, `w`). The next thing to do is to set up some more piles. How about we go through and make a *Finished Goods* pile? Build it using `p`, `g`. Lets keep making stockpiles in those rooms. A few squares for cloth (`h`), leather, (`l`), and bars/blocks (`b`) makes sense. Where one pile starts and another stops can get confusing, but give it your best shot.

Finally, lets also add a Craftsduwarf's Workshop to our room. Use `b`, `w`, `r`, chose the materials and place the workshop. These are particularly handy and profitable workshops, but more on them later!

This is how mine looks:



That's all for now, except before I go I have some homework for you. But don't worry, this is the fun kind of homework. See if you can complete these tasks before we meet again:

1. Destroy the carpenter's workshop upstairs.
2. Designate another wide area of trees to harvest. If the seasons have changed to autumn by now you'll notice all the trees a pretty gold colour. Admire them before you saw them down.
3. Designate a bunch of plants to be harvested.
4. Make a lot more beer and a lot more barrels, or set up workflow for some of the furniture you'll need later (beds, doors, tables, chairs).
5. Make some *bins* in the carpenter's shop - around 15 should do. You'll have to scroll to find them in the carpenter's *Add new task* list, or use *a, n*. Bins are what everything that isn't food or booze are stored in.
6. Go to the mason's workshop and set it to build blocks on repeat (*q, a, b, r*). Do this only if you've already

managed to make a LOT of bins or your Bar/Block pile will get filled quickly!

7. See if you can make some stone crafts. You will find them under the *g* (rock) sub-menu from the *Add new task* menu on the crafts dwarf's workshop. Make sure you have lots of bins for your crafts to go in though, and a big finished goods stockpile too!

Good luck! And see you soon, in *the next chapter*!

## 1.4 It almost makes sense!

Welcome to Part 4 of the Dwarf Fortress Walkthrough. No nancying around, lets get right back into it! I hope you've diligently completed your homework. If you look below, you'll see I have...

### 1.4.1 Extreme Home Makeover: Dwarf Edition!



...Lots of Crafts! I've gone through and completed all of the tasks at the end of chapter 3, and to prove it, here is a picture of my pile of crafts. Looks like I haven't made much in the way of bins yet, so the crafts haven't been tidily stored in a single bin. Don't worry, once bins are produced the dwarfs will sort all this out.

I've also got piles everywhere, workshops set up and food and booze production going. Things are looking good! But we have yet more useful jobs to do.

Before getting to those though, let's set up a trade depot outside so merchants can bring us wagons of useful goods. Place that with *b*, *D* outside - wagons need a three tile wide pathway without traps, so this is the easiest way to keep it accessible.

Next time traders turn up, just remember to check out the tutorial on *Trading for Fun and Profit* for a guide to the process.

First up, we need to move our booze production downstairs. I don't want to dig out any more of our top-floor space just now, and rock walls can be smoothed and engraved later - which keeps dwarves happy. Of course, we have many options for how we expand the fortress, but we'll go for simple right now, so lets get shifting stuff.

Lets get get on with some digging and our new booze space. Hell, lets move the food downstairs as well and keep it nice and close to the dining room, where dwarfs are going to want to eat it.

Here's how I've plotted things out:



Once that digging is done, set up the big food storage hall as a food stockpile. Then build a still downstairs using b, w, 1. This should be easy-peasy for you now! Your next job is go to upstairs and remove the old food stockpile and still.

### 1.4.2 Look what the cat dragged in!

At some point you're going to get immigrants. I got some right now.

*Some migrants have arrived.*

Soon after the message a stream of new loafers streak into the fortress. And what's the first thing they do? Eat and drink! This has me a little concerned about our food stocks, and if this has happened to you, I suggest you do what I do and sort out some more booze and dig some more bedrooms downstairs and assign those new dwarfs to it.

### 1.4.3 Outdoor farming for fun and profit

You know about farming inside, now how about farming outside? As you may recall we've had some plant gathering going on outside. In temperate climates that means we're generally gathering berries. And once we've eaten berries,

what do we have? Well, if you're a dwarf, you end up with seeds. And wouldn't it be great to plant them? Yes it would!

We'll cover looking at our stocks of goods later, by the way, so just trust me for now.

But berries are going to require being planted outside, as they love the sun don't they? Of course, we don't want to go outside where it could be nasty and dangerous, so what do we do?. We can't really expect strawberries to grow in a dark cave, can we? So how about we compromise with these fruity demons. We'll build an outdoor farm, but we'll lock it off from the world with a wall and an entrance only available to us.

To achieve our goals we first need to dig some handy exit to the outdoors. Perhaps near our existing farm. This is what I did:



I've expanded the farm room a little and set a passage to the outside to be dug. Once the space is dug you'll notice that the slope icons still exist around the exit. This could be a problem. If we built walls around a nicely enclosed farm now enemies would still be able to get to it from above, by walking down the slopes! So we need to remove the slopes. To remove it, we hit `d` for *Designations* and then `z` for *Remove Up Stairs/Ramps*.

I've selected almost all of the slopes across the front of my fortress. I don't want any surprises 'dropping' in anywhere along our front. Here you can see my miners hard at work stripping away the outside ramps so there's essentially a sharp drop between the level above and this level.



While you're at it, you could tidy up the outside edge of the fortress with digging and ramp removal. Here's my much tidier fortress entrance:





Later on we might incorporate some complex defenses into this area.

---

**Note:** Recent versions of Dwarf Fortress added the ability to climb, so to be truly secure a wall must smoothed natural stone, or built two levels high with an overhand at the top. Using stone blocks also helps.

That's pretty tricky though, so for now we'll just hope the goblins don't bother.

---

So, lets get on with this farm! We need to surround a nice large area with walls, right? Keep our dwarfs safe from wandering critters. To build walls we need to:

- Hit `b`
- Hit `C`, (that's `Shift-c`, remember the keys are case-sensitive), or scroll through the list and look for *Wall/Floor/Stairs/Track* and hit `Enter`.
- *Wall* is selected by default, hit `Enter`.
- You now have a green X. Like the farm plot you can change the size with `u`, `m`, `h` and `k`.
- Hit `u` until you've got a max-height wall.

- Place the wall right next to the entrance, hit `Enter` (below you can see how I placed mine).
- Select a material using `=` and `Enter`
- Hit `Enter` until the list goes away and you see the wall outline. You're selecting one item for each segment of the wall.
- Hit `Esc` until the game resumes.

You will now have a wall under construction! Again, hit `q` and move it down your wall, you'll see the construction status. Don't worry, your dwarfs will get to the wall pretty quickly.

While you're waiting, clear all the trees and bushes from inside your soon-to-be farm space using `d, t` to *Chop Down Trees* and `d, p` to *Gather Plants*.

Once you've got this under way, build two lengths of wall across to the right go down a length and a bit, and back to the cliff face. Here's how my outdoor farm plot looks so far:



Isn't it coming along nicely? Soon we'll be able to hide inside and behind our walls and ignore the nasty outside world. Yay!

Once your walls are complete you could easily build two 6x6 farm plots inside this space. One point though, make sure all those trees are cut down and plants harvested, otherwise you're going to end up with a patchy farm plot. Also, if there are any trees in the way of a wall being built you won't be able to place the wall. Get them cleared and the problem will go away.

Here's my private outdoors farm yard with the farm plots built as well:



There you go! Beautiful! Now, just like farms inside, you need to specify what the fields will build (q). On the first, I've set strawberries for every season (don't forget to cycle through the seasons using a, b c, d). When I tried to select strawberries on the second field they were red, suggesting to me we won't have enough seeds that this isn't the season for planting those items. So instead, I selected some other random plant. Not sure we've got seeds for those, but we'll find out all about that later! And later on you can come back and fix up some better planting instructions. Oh, don't select *Seas Fert* or *Fertilize*. We don't have any fertilizer yet.

While I remember things, let's take a moment to build a wall along the top edge above our outside farm. We don't want any goblins walking up to the edge of the cliff, looking down, and shooting up our farmers with their crossbows! So go up a level, using good-old < and plan out your wall. Here I've built a wall, which should once and for all block off any possible approach to my farm.



There's one last thing we should take care of today - hungry animals! It's pretty common for livestock to need grass (or cavern moss, or...) to graze on, so we'd better set up a pasture zone.

- Head up near the wagon, where there's open space and plenty of grass.
- Hit i for a zone, select a large area, then n for pasture.
- Hit N, then scroll with = and select grazing animals with Enter.





Done! Our livestock will now happily graze under the trees, turning grass into a foundation of dwarven industry and cuisine.

I'll see you in [chapter five](#) to look at expanding our industries to something truly impressive!

## 1.5 An Industrial Revolution

You have come a long way, young jedi! Your dwarf herding skills are strong! You now know the basics of building, production, trading and managing your dwarves.

This is probably a good time to go read the tutorial on trading - it ties in pretty well with a large industrial base. Tutorials are designed be skippable though, so if the thought of heavy industry excites you, we can dive right in!

### 1.5.1 More Production!

Lets start with some new workshops you should get to grips with. I won't list the keys you need to use for every task, you should be able to remember the basic stuff from earlier lessons, so lets just get building these workshops around our four big production rooms:

- Mechanic's Workshop
- Leather Works
- Farmer's Workshop

- Kitchen
- Butchers
- Metalsmith's Forge

In addition, go to the Furnaces menu (from **b** and then **e** for *Furnaces*) and build:

- Wood Furnace
- Smelter

While those places get built let's look at expanding a bit. In the middle of our four room area we have three stairs going down. Right now they go nowhere, so let's go down about 5 levels. Remember, **d** and then **i** for *Up/Down Stairs*, set to the level below, will get that construction going.

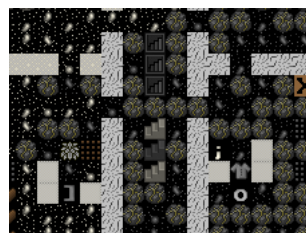
We also have room to expand between our production floor and our entrance, so let's go up a level and put some storage there. I suggest the easiest way to do this will be to **b**, build a **C** constructed up-stair. So choose the space I highlight below, and then follow along!



Find the spot in the picture above, just below those down stairs.

- Hit **b**
- Hit **C** (**Shift-c**)
- Hit **u** for *Up Stair*.
- Chose some stone.
- Repeat to set three up stairs to be built.

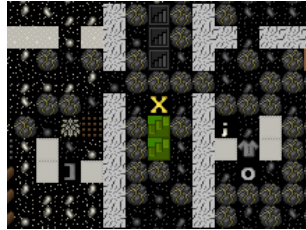
Once the job is done you'll have some stairs like this:



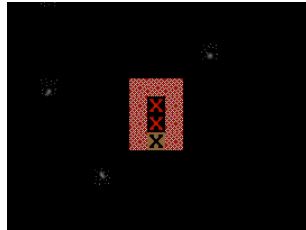
Now we want to connect this level to the one above. If you go up a level you won't see anything but dirt, but we know the stairs are leading up to this level, so it's a matter of constructing some, right? **WRONG!** We want to designate some stairs, because our miners will carve out some matching stairs! We only use *Construct* when we've got an empty space to deal with.

One challenge you'll face is working out where on the floor above the spot is you need to dig out. So try this technique:

- Hit **d**
- Hit **i** for *Up/Down Stair* (we want a series of stairs almost back up to the surface).
- Put your **X** over the top stair in the line, like the screenshot

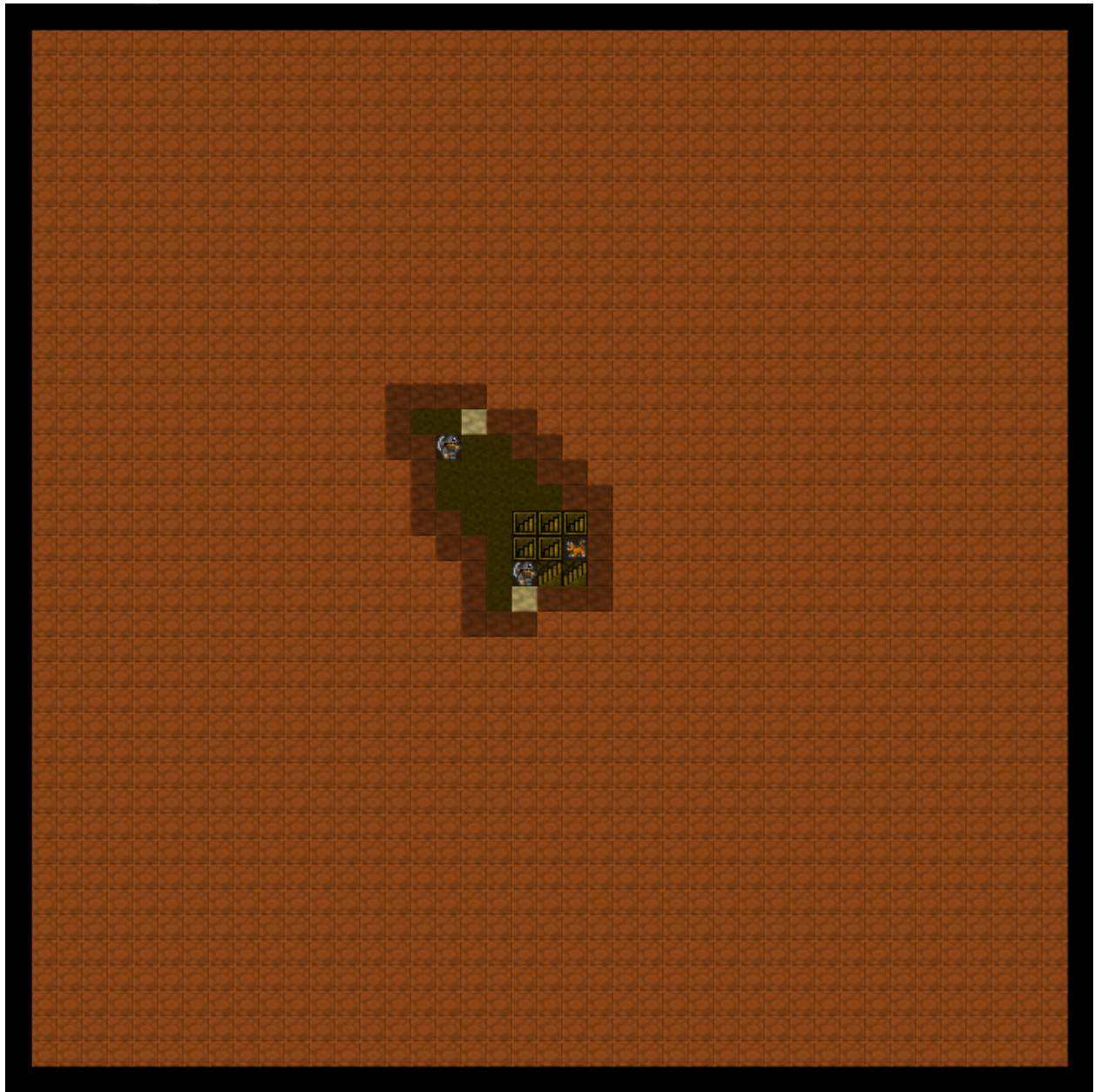


- Now go up a level and at the above level, designate three stairs in a row (ie, hit enter, then hit down-arrow a couple of times, and then hit enter), the above level should look like this:



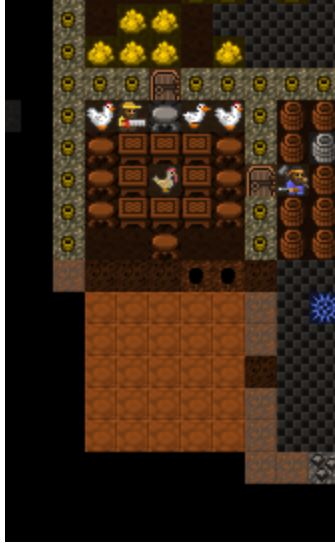
You'll note I wasn't quick enough when I took this picture and two of my three stairs are already dug out. Good dwarves!

With the new stairs dug out (providing easy access to all the space we're going to create for our production dwarves) lets make a massive space for them to stockpile goods. Here's how much I dug out:



In case you're wondering, that's about a 40×40 box with our first stairs at the bottom. And one nice thing about digging in dirt (which this layer is), is that it doesn't leave any messy rock around to clutter up our nice stockpiles!

While that's completing, lets start an important job, making our dining room awesome! You see, dwarves love to spend time in an attractive meeting hall. Right now mine is packed with loafers. They clearly need a bit more room! A bit of digging will sort that:

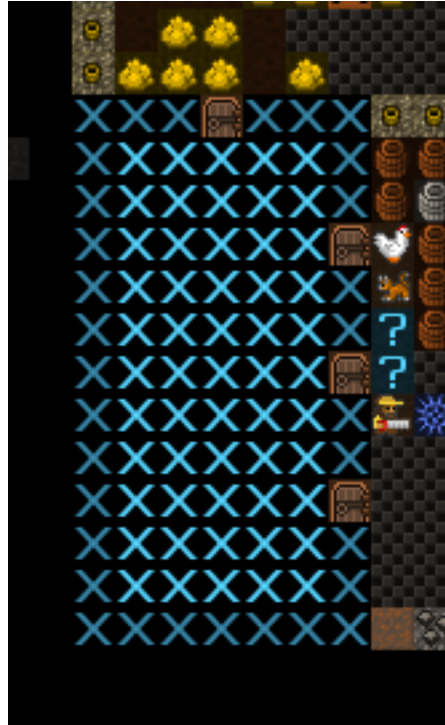


Once the room is expanded, fit doors and some more tables and chairs. Once that's done you'll need to do something important, and that's resize the room. As you may remember we set the room up from one of the tables. But if you *q* over the dining room table now you'll see it doesn't fill the space:



The room will be more valuable and widely used if it's set to be bigger, to fill this space. Lets do that now:

- Hit *q* and move the X over the table which is setting the room up. . .
- Hit *r* for *Resize Room*.
- Using = expand the room size till you fill the space:



- Hit `Enter`

Job done! The room is resized and better. But we want to make it MUCH better because I noticed a dwarf with a red down arrow flashing and when I viewed his info it turned out he was pretty unimpressed with his surroundings. Lets get to impressing him!

How do we do that? Simple! We smooth the walls and then engrave them with fine carvings all about our fortress! To do this:

- Hit `d`.
- Hit `s` for *Smooth Stone*.
- Select the entire dining room and walls using `Enter`, move cursor, `Enter`.

The room will now look all flashy like this (until you back out of the menu, anyway):



Any dwarf with the *Stone detailing* labour on will now set about smoothing the walls and floors. The next step, once the space is smooth, is to designate the room to be engraved using `d`, `e`. Engraved walls make dwarves happy and increase the value of your fortress. You can even look at them by using `k` and hitting enter with the engraving highlighted. Some walls have some quite amusing engravings (randomly generated and based on the history of your dwarves and your fortress), so it's worth looking around and finding the good ones. The better the quality of the engraving the more text there is to read, so keep an eye out for the engravings with the metal-bars icon next to them.

With the smoothing under way, lets get back to the piles. In that big space upstairs you can pretty much set up a space for everything it's possible to make a pile for (except refuse, of course, which we want to be outside). So go do that now. Here's how mine is laid out.





Pretty complete huh? Well, almost, I'm sort of tempted to make a stone pile and using custom settings have it as metal ores only. It would make things a bit tidier, but on the other hand probably doesn't gain me that much efficiency right now. Be very careful with stone stockpiles, hauling stone can consume a heap of your dwarves' time, which is a bit pointless.

### 1.5.2 Hotel Califortress!

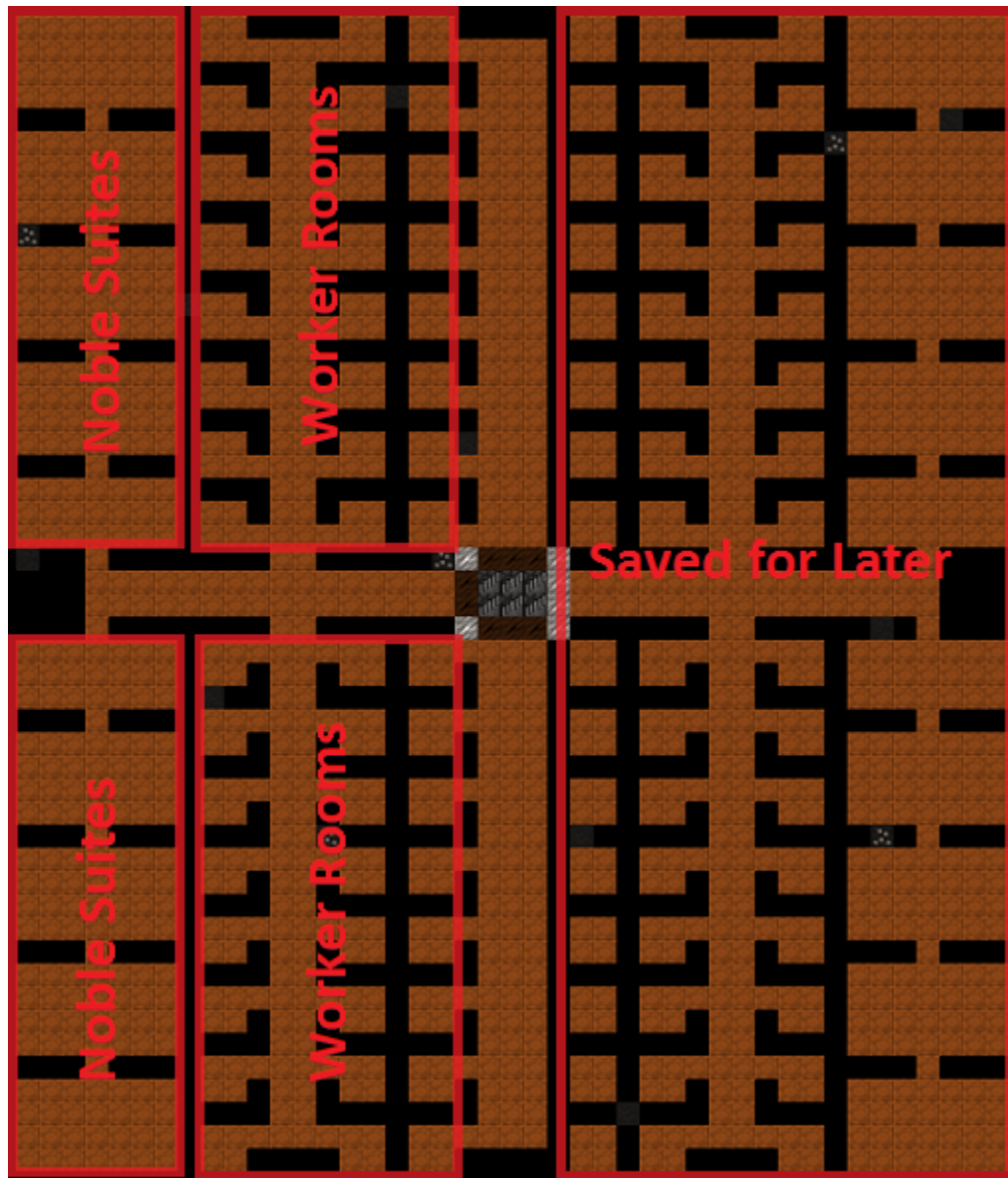
We've got some new workshops set up, we've got some great storage, and we've dug down a few levels. Good stuff! Unfortunately, (or fortunately?) you've probably had a bunch of immigrants arrive over the course of the past couple of tutorials and they haven't been assigned anywhere to live. We also don't have any space dug out for the inevitable arrival of nobles, and these boys and girls are one set of dwarves that need hard work to keep happy!

Fortunately we've dug down a few levels and we have a lot of nice rock down there. So lets go take some time to lay out some great bedrooms for our dwarves. The majority of your rooms should be 2x2, but lets make some space for



nobles too.

Nobles usually want two-to-four rooms. So lets make things easy and give them all four rooms of size 3x3 or so, that should be good enough. Here's how I have planned out my bedroom level:



A couple of points. Firstly, the whole right side of my new bedroom level won't be dug because it's not connected to the stairs, etc. I'll connect it up later once we've got the left side done. Second, my layout is pretty boring, go and check out some of the bedroom designs on the wiki. I love the fractal pattern! Very efficient! Finally, you'll notice the big wide corridors for the main arterial routes. dwarves need space to move around, remember!

Also, remember that you will need a lot of new doors and beds for our beautiful hotel. While you're at it, build a ton of tables and chairs and at least a half dozen coffers, cabinets, armor stands, weapon stands. You'll find all those items under the masonry workshop menus. Oh, and lets connect up the southern most stairs (the first ones we built way back in tutorial 1 or 2) with this level as well. Construct down stairs from the workshop floor and then designate Up/Down stairs till we're all connected up.

While that is being built (oh, we're going to find some gems too, cool!) lets continue with-

### 1.5.3 Dwarves and their strange moods!

Oh dear! Something is going down in dwarf land!

*Endok Oltarisos, Tanner, withdraws from society...*

If you get a dwarf in a strange mood, find them using `u`, looking for their name, and then `c`. You will see the dwarf flashing with a grey exclamation mark (red is very bad, by the way). Follow this dwarf closely. This dwarf has got a strange mood and is off to claim a workshop to start building some amazing object based on their whim and fancy. We can't control what they build, all we can do is hope they build something cool and that they can find all the materials they want for their fancy. If not, they go suicidal or homicidal. Oh dear! Lets watch and see what happens. Of course, this event is random, so it might not happen to you at this point in the game, but it will happen sooner or later.

Right, my dwarf, Endok Oltarisos has rushed off to claim a leather workshop. No surprise, he's a tanner after all. Once he claims the workshop you can `q` and see the status of the workshop, and if you wait, it will scroll through what items the dwarf is looking for. Endok is looking for *stacked leather* and *skeletons*. I'm not sure if I've got any. I'll find out pretty quickly though, the dwarf will either run off and start fetching stuff, or sit in the workshop, seemingly doing nothing...



Well, the leather isn't a problem, I just bought a ton from a trader, and the bones, I think he's grabbed some from the refuse pile. Thankfully, my worries about the dwarf not getting stuff are put to rest when I get this message:

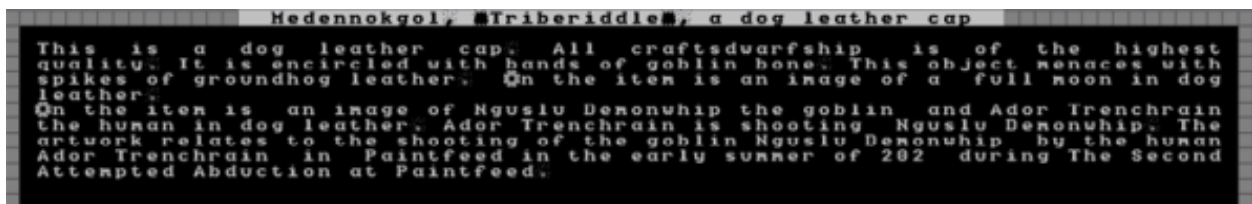
*Endok Oltarisos has begun a mysterious construction!*

Endok has begun a mysterious construction! Great! Now we just wait and see what crazy object the dwarf produces. With luck it will be useful!

*Endok Oltarisos, Tanner, has created Modonnokoi, a dog leather cap!*

..or maybe not! Our dwarf has made a cap, a simple hat, out of dog leather! Damn! On the plus side though the dwarf has become a legendary tanner. If they gain skill from a mood it often leaves them *Legendary*, which is pretty neat. With legendary skill I could use Endok to make leather armour and it would be almost as good as metal armour.

Lets look at the item. If we hit `1` we get a list of artefacts. With only one artefact there's no list, so we can go straight in to hitting `v` to `:guilabel'View'`. Behold! Triberiddle, the dog leather cap!



A nice hat depicting when one of the trade caravan guards shot a goblin. No one will wear it though, it's an *artifact*, and only champions are important enough to grab artifacts from stockpiles.

We were lucky this time with our moody dwarf. He was able to get everything he needed to make his artifact. If he couldn't find it he would go quite crazy in the workshop, or if a suitable workshop isn't available, in his room. When you see the dwarf start to go crazy (flashing down arrows and not moving from their workshop are a good sign) it's time to either assign the dwarf some war dogs (more on that later) or to construct some walls and wall them in to their workshop, or to lock the door on their quarters. At some point the dwarf will go berserk and either get attacked by nearby military dwarves or war dogs, or if locked inside, slowly starve to death.

If the crazy dwarf is ignored they will destroy stuff and attack dwarves, probably killing a couple before they are put down, so watch those moody stunts closely!

In *the next chapter*, we'll cover traps - used to "mine" a kind of iron ore the community calls "goblinite". See you then!

## 1.6 It's a trap!

Traps form the majority of many people's defences, so it's best we get sorted and make some. First up, we will need a lot of mechanisms, so go find that mechanic's workshop and fill it's job queue with mechanisms. They'll end up in a finished goods pile when done.

Next, go to the Carpenter's workshop and add a ton of cages (j is the shortcut from the *Add Task* menu). Try and get ten built. Add another carpenter's workshop to your workshop floor if you fancy, makes it easier to queue up more than a few of any item and get things made faster.

### 1.6.1 Defence and Traps!

You've got mechanisms, you've got cages, now lets make some traps! Head to the front entrance of your fort and we'll get building traps and then ponder the various strategies one might employ in setting up the defence of your fortress. Follow along:

- At the front entrance, hit b.
- Scroll to *Traps/Levers* and hit Enter, or just hit T.
- You will now see a list of traps, lets briefly discuss them.

**Stone-fall traps** These are simple one-shot traps that drop a big rock on an enemy's head. These are great for a quickly built defence, but clearly, as one shot traps, a little limited. Note, with all traps, if your dwarfs fall onto them, unconscious, they will trigger the trap!

**Weapon Trap** These will probably be your bread and butter as they will attack any enemy triggering them and with multiple weapons can be quite powerful.

**Lever** Not a trap, but a device used to manually trigger things like bridges, cages, doors, floodgates, supports, etc. More on these later some time.

**Pressure Plate** Can be part of a complex trap or can be used to trigger other objects when conditions are met (such as water being on them, creatures stepping on them, magma flowing over them, etc). More on those later too.

**Cage Trap** What we're going to be working with in a moment! A trap that catches the enemy in a cage. You can then do amusing things with the trapped bad guys, like tossing them off a tall bridge, dropping them into magma, or stripping them and using them for military sparing practice. Good fun!

**Upright Spear/Spike** A peculiar sort of trap device that can only be triggered by a pressure plate or lever.

You should read about [traps](#) and [trap design](#) on the wiki for a lot more detail, but only after you've completed this tutorial, ok?! For now, get back to work!

- Scroll down to *Cage Trap* and hit `Enter`.
- You now have a green X, place it in front of your entrance somewhere with `Enter`.
- Do this a few times, you can see below how I've set a few traps to be built.



This is hardly an impressive defence, but it will do for now. With luck it might help against any casual raiders, although sneaky ‘trapavoid’ monsters will still be able to sneak past your traps. To beef our defence up a bit we need to get some weapon traps installed as well as sort out some way to strike back at attackers who aren’t so gracious to step on all of our traps. So with that in mind, get producing a lot more mechanisms (at least 10) and follow along as we produce some trap components!

### 1.6.2 Smelting and other dwarfy things!

This wouldn’t be a *dwarf* fortress if we didn’t fill the place with the clang of forge hammers and the stink of furnaces, right? Lets have some fun then and make some sharp things to hurt our enemies!

---

**Note:** If you haven’t dug up any *Magentite* yet, this is a good time to go find it - it’s a form of iron ore, and we can’t smelt iron without any ore!

---

First up, let me point out that item production can be fairly confusing. Essentially what we’re going to do is turn some wood into charcoal (at a wood furnace), then use the charcoal to turn metal ores into metal bars (at a smelter), then we’re going to use more charcoal to smith metal bars into items (at a metalsmith’s forge). As you can no doubt imagine the problem we will face is the charcoal bottleneck. It will take a lot of wood to fuel all our furnaces! Most of the time this problem is got around by players using magma smelters and magma forges. We haven’t struck magma yet, so we’re going to plough ahead with the hard way, at least for a while. The goal of all this is to produce a few fearsome trap components to fit out some weapon traps – think spinning disks from Indiana Jones! So follow along:

- Find your wood furnace, hit `q`, a

- Select *Make Charcoal*, on `r` repeat.
- With the job selected, `Alt-w` to open workflow. Set this to keep 5 to 10 charcoal bars available.
- Now go to the smelter, and try to `a`, *Add new task*. You'll see a list of the tasks available at this workshop. Note, the tasks will change as you uncover more ores or deplete older ores. For interest, note that you can set the workshop to melt objects you wish to dispose of (turns the object back into a metal bar). More on that some other time.
- Chose *Smelt Magnetite Ore*, hit `Enter`. Magnetite is a form of iron ore, so some of the rocks lying around will become iron bars.
- Fill the queue and then fill it again (or build another smelter and fill both queues).
- Once we're done with this tutorial why not go and smelt all of that native platinum ore you saw on the list as well? If you've dug downstairs out you may have gold to smelt too!

Your dwarfs should be pretty busy now, hauling charcoal, wood, ores and bars around. While you're at it, make sure you've got enough wood around for charcoal by cutting more down outside!

If you want to check your progress, go look at your Bar/Block stockpile and see what is being built up. If you suspect all of your production is in bins, use `k`, find the bin, hit `Enter` and look inside it.

### 1.6.3 A little on food processing

While all this is going on, why not go to your farmer's workshop and add a task of *Process Plants (to bag)*. Set it on repeat. Then add a task of *Process Plants (to barrel)*, again, set it on repeat. Now some dwarves will process anything you've grown that needs processing. I won't list those plants here now (go look at crops for all the detail), but suffice to say, some plants need to be processed so they can be eaten or used in other sorts of manufacture (eg, cloth making).

Hopefully you're not short on bags. If you are, try and pick up a bunch off a trader. Alternately, buy leather and/or cloth off a trader and using the Leather Workshop or Clothier's Shop (we've not built one yet) make bags. Bags are very handy for storing seeds and processed plants, you can never have too many of them. They're like barrels and bins in that way.

I also just noticed that our farm entrance was entirely undefended, so I added some doors and cage traps. Hopefully that'll be enough!



### 1.6.4 Back to the iron!

With a few iron bars made, head on back to the smelter - we're ready to rock! I hope, at this point, that you've not been invaded, attacked or otherwise molested. With our fortress around two years old now it's likely you'll start to see the odd goblin ambush – especially if you send your dwarfs outside a lot. I hope you survive those ambushes just fine! Perhaps we should have started this chapter sooner. . .

Anyway, head over to your metalsmith's forge, and:

- Hit `q`, a You will now see a list! Explore the list, see all the neat things we can make! You might need to hit `Tab` to expand the menu size so you can see everything.
- Scroll to *Trap Components* and hit `Enter`.
- Scroll to *Iron*, hit `Enter`.
- Hit `Tab`, `Tab` so we can see the full item names.
- Chose to make a *large, serrated Iron disc*. I love the sound of that!
- Queue up three disks.

At this point you might want to think about other items you'd like to make in due course. Hit `a` again and then hit `space` to back down the make-item tree. Chose *Weapons and Ammunition*, chose *Iron* and let's make three Iron warhammers. These will come in handy once we start our military. I like warhammers because it's funny to watch goblins fly a half dozen tiles and then expire in a heap.

Once you're confident your items are made, go back and make sure you've got some more charcoal in production, you've got more ore being smelted and add, under *Armor*, three suits of iron chainmail and three iron shields. This will all come in handy later.

Oh, you may be getting annoyed at slow production. I know it's been bugging me. Perhaps it's time to pack your production rooms with more smelters and charcoal burners. One issue you'll face is you need to make or buy an anvil

off the traders in order for you to make another metalsmith's forge. Keep it in mind for your next trading deal.

The other thing to consider is setting up Workflow for every step - automating complex industries like steelmaking (or soap) is exactly what it's designed for.

### 1.6.5 Deploy Zee Traps!

Once you've got your nice, big spikey discs, head back to your entrance. Get ready for trap placing fun!

- Go to place another trap, a *Weapon Trap* this time.
- Chose a mechanism.
- A new screen will be displayed looking something like this:



- From this screen you can chose the weapons you want to deploy with your trap. Facing goblins, we really only need one awesome iron disc per trap I think, but we're able to select more if we wish.
- Scroll to the *large, serrated Ir...* and hit *Enter*, you'll see *1/3 selected*. Weapon traps can have up to ten weapons each, but we need to save some for the other traps!
- Hit *d* for *Done Selecting*.
- Your first weapon trap is now placed! Place three more.

Weapon traps are great ways of getting rid of a bunch of trash weapons you might have lying around – this is particularly the case once you've killed off a few goblins and have their rubbish cluttering up your piles. When filling a trap with weapon trash you might want to put ten low-quality weapons per trap, to ensure maximum damage!

Here are my traps, laid out. I'm actually going to add another row of slicey dicey weapon traps as I don't have a military and I'm a bit paranoid. Do something similar!





Well done on the trap placing! But we're not done with our defences just yet. We need to stop our drunken citizens from casually strolling into an invading army - and that means the Burrows system!

### 1.6.6 Learn to love Burrows

"Wait!" you're probably thinking, "Why do we need filthy burrows when we have a perfectly good fortress?". Well, I'm glad you asked!

A burrow isn't a physical part of your fortress, it's a way of designating where dwarves are allowed to live and work. More importantly, you can set a *Civilian alert* to confine their movements too - perfect for stopping Urist McSuicidal from wandering off to make friends with a troll. Here's how to set it up:

Want some more detail? You're in the right place! Start by hitting `w` for the burrows menu, and `a` to add. As shown in the gif above, `Enter` to designate is the important thing. Cover the whole underground area, since it's safe there, and `n` name it something like "Inside".



```

FPS: 99 <47>
Siege: no squads, 1 burrow
c: Add alert d: Delete alert Enter: Set civilian alert N: Name alert
Civilian alert: Siege

ALERTS          SQUADS          BURROWS
Inactive
Active/Training
Siege [CIV]
Civilians set to "Siege" alert, will not exit activated "Inside" burrow

p: Positions a: Alerts e: Equip n: Uniforms u: Supplies f: Ammunition
ESC: Done      234689: Move selector s: Schedule

```

The key feature here - for civilians, the military is complicated - is that on the left we have *ALERTS* and on the right we have *BURROWS*. This means that we can set non-military dwarves to be confined to a burrow, and change which if any that is with a single alert setting.

This is the *m* military *a* alerts screen, which is also used for for civilian alerts.

```

FPS: 100 <46>
Siege: no squads, 1 burrow
c: Add alert d: Delete alert Enter: Restrict civs to burrow N: Name alert
Civilian alert: Inactive

ALERTS          SQUADS          BURROWS
Inactive [CIV]
Active/Training
Siege
List of All Burrows
Selected Alert
Current Civilian Alert
Inside
(more burrows would be listed here if there were any)
A = Burrow active during selected alert

p: Positions a: Alerts e: Equip n: Uniforms u: Supplies f: Ammunition
ESC: Done      234689: Move selector s: Schedule

```

Now, let's set up a civilian alert in our new burrow, to make sure nobody gets caught on the wrong side of the traps:

- *c* to add an alert, then *n* to name it. I called mine "Siege" because it's to use during a siege. Yep.
- Now move the cursor to the *BURROWS* column, and hit *Enter* to associate that burrow with your new alert. You'll see a green *A* next to the burrow when you have that alert selected.
- Hit *Esc* a couple of times to get out of the military menus.

You're all set up! In case of invasion *m*, *a*, select the Siege alert, *Enter*, and you should see a green *[CIV]* next to it.

Congratulations, your civilians are now confined to the burrow! Watch any civilians who were outside drop what they were doing and come running back inside. Note they will not go for food, so your safe room may become a death sentence if there's no supplies (but that shouldn't be a problem by now, right?).

When the 'fight' (against brave... traps) is over, come back and select *Inactive* and hit `Enter` to cancel the alert and let your civilians back out of the burrow - that's all there is to it.

Anyway, that's enough to now. Get on with building lots of traps, making bags, armor, barrels, food, bins, shields, weapons and so on! Go to it, and I'll see you in *the next chapter!*.

## 1.7 Nobles and other useless sods

You've come a long way, padawan, but don't worry, I won't let any stupid angsty teenager come and strike you down because he's in a tizzy having lost his mum! No, nothing of the sort! Baby Dwarf Fortress managers like yourself will prosper and grow strong like a seedling, going on to bear nice dwarven fruit. And not the Class-C sort of fruit that gets picked and sold on the streets, no, nice, rare, delicious fruit! Anyway, where was I? Oh, yes, chapter 7, Nobles and other layabouts. Get on with it!

### 1.7.1 Interior decorating

Decorating your massive housing project should be complete now. Here's how mine looks. Every room has a bed and a door and the noble rooms (they will need four rooms each, as a rule) are kitted out with assorted stuff to keep any nobles who visit happy. Lets cover off the details.



Nobles (from the `n` menu) generally want a bedroom (assigned from a bed), a dining room (assigned from a table), an office (assigned from a chair) and often a tomb as well (assigned from a coffin). For simplicity's sake, set up each room in its own contained area and then throw in some decorations to keep the noble happy, like armor stands, weapon stands, containers, statues, cabinets, cages (with their favourite animal in them), that sort of thing. Usually nobles want at least two of most of those items so produce a ton and get placing!

What is all this about nobles I hear you ask? Well, once your fortress gets to a certain size, those purple slacker dwarves will come to live in your fortress. They usually don't do much (other than haul stuff) but they do make demands. Their biggest demand is for rooms, which need to be of a certain quality. Quality can be increased through smoothing and engraving the walls of the room as well as adding more furniture, especially high quality furniture.

The other thing nobles do is make mandates. Mandates are demands by the noble for things to be done, like the production of certain goods, or a ban on the export of certain goods. It pays to try and meet the demands of your nobles or they may go crazy and cause all kinds of problems.

Lets have a look at nobles. Go ahead and hit `n`. This is your noble list, right now it should be fairly short, but in time it will grow. Chances are it looks like this:

```

king          Rakust Telushat, king          [REQUIRE][DEMAND][MANDATE]
expedition leader  Urdim Erithteshkad, expedition [REQUIRE][DEMAND][MANDATE]
militia commander
sheriff
hammerer
manager
chief medical dwarf
broker
bookkeeper
Kivish Cubordodok, broker      [REQUIRE][DEMAND][MANDATE]
UACANT
UACANT
UACANT
UACANT
UACANT
UACANT
UACANT

Enter: View Unit/Fill Vacancy  r: Replace  s: Settings
8293: Scroll                   ESC: Done

```

In most new fortresses only one dwarf is set as expedition leader. As the fortress grows more noble ranks will appear in this list and more positions will be filled. However we've been 'lucky' - one of our dwarves has become king, presumably because all the other candidates died. This is kinda cool, but a king can also be unreasonably demanding.

Lets look at the king. Hit `Enter` and you'll see info about the state of his stuff. Our king owns 14 objects, mostly clothes, and has *meagre quarters*. Rooms and item he 'needs' are in red, and things he has (but aren't good enough) are in yellow. Start building a sweet suite, and we'll come back to him later.

```

Owned Objects:      14

Holdings:  No Office          Needs: Royal Throne Room
           Meager Quarters    Needs: Royal Bedroom
           No Dining Room     Needs: Royal Dining Room
           No Tomb            Needs: Royal Mausoleum
           No Chests          Needs: 10 Chests
           No Cabinets        Needs: 5 Cabinets
           No Weapon Racks    Needs: 5 Weapon Racks
           No Armor Stands    Needs: 5 Armor Stands

Mandates:  Export of bucklers Prohibited

Set Follow Hotkey <F1 etc.>: None
Enter: Thoughts and preferences  r: Relationships  y: Customize

```

Next job for you is to assign a book keeper. Usually, if you embark with at least one dwarf with the *appraiser* skill (good idea, by the way), this job would already be filled. But for now it's vacant. This skill is what makes our stock count accurate (more on that later). Fortunately for us we can chose a dwarf to do the job. They might not be doing a good job at first (having no skill), but they will learn, and they provide a useful function, accurate stocktaking!

So, in the nobles screen use the arrow keys to scroll down to *Bookkeeper* and hit `Enter`. Find Eshtan and hit `Enter`,

assigning him to this job. The bookkeeper requires an office to work, just as well we just assigned him one, right?

Next, with Bookkeeper still highlighted hit `s` for *settings* (the only noble role you can assign settings for). The screen you'll see now dictates how much time your dwarf will spend counting. Use the arrow to scroll to *Highest* and hit enter. Eshtan will now work very hard to get all our stock counts accurate!

Finally, let's assign a room to our bookkeeper - he's not demanding any, but it'll be useful to know how when the royal rooms are finished. `q` to see the room, `a` to assign, select the dwarf in question, and finally Enter to set. Now go back to the noble list and you'll now see his holdings have changed!

There you go, a quick overview of nobles. As you play the game more you will have to deal with nobles a lot, but don't worry, the [Dwarf Fortress Wiki](#) should provide you with heaps of tips and advice should you get confused.

## 1.7.2 Accounting is fun! Really!

We've mentioned stocks, but not discussed the stock screen yet, so lets get to it now. Hit `z` and you'll see this screen:

Hamlet Kilrudod, "Bronzetunnels" FPS: 100 (50)									
Animals Kitchen Stone Stocks Justice									
x: Additional options (DFHack)									
Created Wealth:		369000*?	Population:		38				
Weapons:		None	Miners		3	Axe Lord		3	None
Armor and Garb:		None	Woodworkers		3	Swordsdwarves		2	None
Furniture:		12700*?	Stoneworkers		2	Swordmasters		1	None
Other Objects:		202000*?	Rangers		1	Macedwarves		6	None
Architecture:		77500*?	Metalsmiths		6	Mace Lords		1	None
Displayed:		76100*?	Jewelers		1	Hammerdwarves		4	None
Held/Worn:		None	Craftsdwarves		4	Hammer Lords		1	None
Imported Wealth:		35304*	Nobles/Admins		1	Speardwarves		None	None
Exported Wealth:		1210*	Peasants		None	Spear Masters		3	None
Food Stores:		500?	Dwarven Childrn		None	Marksdwarves		14	None
Meat	None	Seeds 200?	Fishery Workers		3	Elite Mrksdwrvs		None	None
Fish	None	Drink 20?	Farmers		14	Wrestlers		1	None
Plant	300?	Other 3	Engineers		None	Elite Wrestlers		62	None
			Trained Animals		A 1	Recruit/Others			None
			Other Animals		A 62				

Lets go over what you're seeing here:

- Across the top are sub menu options. You can scroll through them using the arrow keys (`←` and `→`) and then hit Enter to learn more.
- There's a description of your wealth on the left. We can get more accurate details with a skilled bookkeeper.
- There are food store stats, but those question marks indicate we really aren't that sure how much food or drink we've got. The bookkeeper should turn these figures accurate fairly soon.
- Then we have population information. Not too interesting I find.

With *Animals* selected, hit enter. You can see all the animals in your fortress on the following screen. This is also the screen you can set animals to be butchered en mass. If you set a few to be butchered a dwarf with *small animal dissection* and *butchery* skills enabled will run off and start grabbing those animals and chopping them up in the Butchery. You won't need to set anything at that workshop, the butcher tasks gets added automatically.

Butchering animals is a great way to get a lot of food, bones and skin (for tanning into leather) but be careful! If you assign a big list of animals to be slaughtered by the time the butcher gets to the last one the animals may have become someone's pet. This results in the butcher walking up to the animal, as it follows its master around, dragging it off and turning it into steak. This is somewhat disturbing to the dwarf whose pet you've just killed! So, don't tag too many animals at once. Also, make sure you leave at least one breeding pair (male and female symbols to the right of the pet's name), or you won't get any more baby animals popping out, will you?

Now go back to the stocks menu and hit *Kitchen*. This is where you see a list of what your dwarves will consider using to cook meals in the kitchen when ordered (dwarves like cooked meals). Remembering that cooking destroys plant seeds, here are some guidelines for you:

- Cooking meat (ie, *cook* is blue) is fine.
- Don't cook spawn or seeds (*cook* is red) unless you have way, way too many.
- All plants can be brewed as brewing returns seeds.
- Turn off cook for *Plump Helmet*, you don't want to cook them out of stock, right?
- Cooking booze is great as you end up magically creating more food than you started with. Trust me on this (or look it up on the wiki).

Good work! Now back out of that menu and chose stone. In the stone menu you see (in red) all the stones your dwarves are forbidden to build with/use and in green, all the ones they are allowed to use. Scroll through with the arrows and hit enter over every stone that does not have a use description appear on the right.

The reason you are doing this is so that the dwarves have much more choice in the stone they will use to make doors, walls, etc etc. This stops them running half way across the map to get some boring stone when a perfectly useful stone (that just happens to be yellow) is sitting next to them, but ignored.

Backing out to *z* the final menu is *stocks*. This is a master list of all items in your fortress and is a good way to see exactly how much of certain items you've got. Until your bookkeeper gets working you won't get any detail, but trust me, you'll see heaps, in time.

You can use this menu to manage the items in your fortress more easily. For example, you might find, after a goblin raid, that you've got a ton of their crappy weapons lying around, what to do with them? Well, you could add them to weapons traps, or you could find them in the stocks menu and hit *m* for *Melt* when they are selected and then any smelter with the *Melt metal object* task on will have a dwarf grab the item and then melt it back to a metal bar.

You can also use this menu to forbid items. If you have a bunch of low-quality junk you don't want your dwarves to use, hit *f* for *Forbid* with the item highlighted and they will ignore it. This is handy when setting weapons and armor up for your military as you can prevent them from picking up wooden swords and thus push them to pick up your non-forbidden steel swords!

The last menu *Justice* doesn't come in to play until we have more nobles who will start wanting to throw dwarves into jail if they ignore their mandates. Worry about that latter!

### 1.7.3 A bedroom fit for a King!

Remember that sweet suite the king wanted? If you didn't decide it was a royal pain, here's an example of what you might build:





- Four large rooms, smoothed and engraved on the walls and floor
- Plenty of valuable furniture
- One bedroom, one office, one private dining room, and one tomb

```

Owned Objects:      14
Holdings:  Splendid Office      Needs: Royal Throne Room
           Quarters             Needs: Royal Bedroom
           Fine Dining Room     Needs: Royal Dining Room
           Burial Chamber       Needs: Royal Mausoleum
           2 Chests            Needs: 10 Chests
           2 Cabinets          Needs: 5 Cabinets
           2 Weapon Racks      Needs: 5 Weapon Racks
           2 Armor Stands      Needs: 5 Armor Stands

Mandates:  Export of bucklers Prohibited

Set Follow Hotkey <F1 etc.>: None
Enter: Thoughts and preferences      r: Relationships      y: Customize

```

He's still not happy of course, but it's something. At this point you can continue piling on the valuables, or arrange an... [unfortunate accident](#).

### 1.7.4 Homework time!

You've got a tricky job to do now, ok, not so tricky really. I want you to dig exploratory tunnels in every direction from your living room and the below levels. Try and cover most of the map with your grid of tunnels. We're looking for three things: metal ore, gems, and magma. You may want to assign another dwarf or two to mining duty, and to make a few more picks as well (at the metalsmith's forge, under *weapons*).

We'll see what this map is like when we *return next chapter!*

## 1.8 Soldiers and your army

While you're exploring and digging around, how about we set up a military? You're going to need one, goblins are pesky bastards and likely to bother you more and more (if they haven't already). Soldiers can be a bit fiddly to manipulate, but once you get the hang of it you'll find it second nature.

### 1.8.1 Drafting some dwarves

The first thing you will need to do is draft some dwarves into your military. There are a few different strategies for this. First up, you can pick complete newbies and let them work up their skills and strength through sparing. Another strategy is to have some peasants work on screw pumps to quickly build up their strength and toughness before drafting them. Others like to use miners (which train up quickly and are thus easy to replace) as they usually are quite strong and tough already.

As a further consideration, some don't like to recruit women as soldiers. When they give birth it's not uncommon to see them running into battle carrying a baby. This is both somewhat alarming and somewhat risky as you're going to have potentially dead babies on your hands due to combat, something that might cause a lot of emo among your dwarves.

For this tutorial we're not going to worry about detail too much and we're just going to grab 6 peasants – which is 10% of my current fortress population. Something similar will work for you for now. So here's what we'll do:

- Hit `m` for the *military* screen.



You'll note this screen is not unlike the `u`, *units* screen. The difference is that there are a few different commands at the bottom of the panel. Primarily, this screen lets you organise your squads and it lets you set squad weapons. We're going to aim for two squads of three dwarves. Smaller squads helps your response time to enemy threat and gives you a bit more control. Lets get on with it:

- Hit `c` to create a squad, and using metal armour. (You *did* make the iron armour in chapter 6, right?)
- Scroll left then down with the arrow keys, to one of your no-life peasant dwarves, hit `Enter`. You will see the peasant is marked and you can chose another dwarf. Chose another pleb, hit `Enter`.
- Scroll, chose another dwarf, and hit `Enter` again.
- Hit space to stop promoting. We've created our first squad.
- Chose another spare peasant, hit `Enter` to promote and then choosing two more dwarves, create squad two.

You will now note that the dwarves in the same squad are now sharing the same squad name, *The Razonrs of Handling* and *Fortresses of Sorcery* for me!

```

Date:126-05-24  Idlers: 20
a: Assign Bed
f: Free Bed
b: Barracks (Y)
d: Dormitory (N)
r: Resize Room

n: Give name

The Razors of Handling ZTIQ
Fortresses of Sorcery ZTIQ

z: Sleep      t: Train
i: Indiv Eq   q: Squad Eq
p: Position   +-*/: Select
  
```

But they're not doing anything yet, they're assigned to squads but aren't active doing soldierly stuff, so:

- Scroll to the first squad leader, hit `a` for activate.
- Scroll to the second squad leaders and `a` activate him as well.
- Scroll to the top of your military list, your squads are now up there with their names in grey.
- On the first squad leader, hit `v` for view squad.
- If the squad is *on duty* hit `t` to stand them down.
- Hit `f` for *Food carried* and they'll go get backpacks and put a stock of food in them for long duties.
- Leave other settings and repeat for squad two.

Our dwarves are now soldiers and are going to go do soldierly things. What's that you ask? Well, drink, eat and hang around the barracks sparing mostly. But shouldn't we give them some weapons and armor? Maybe, is the answer, lets show you how:

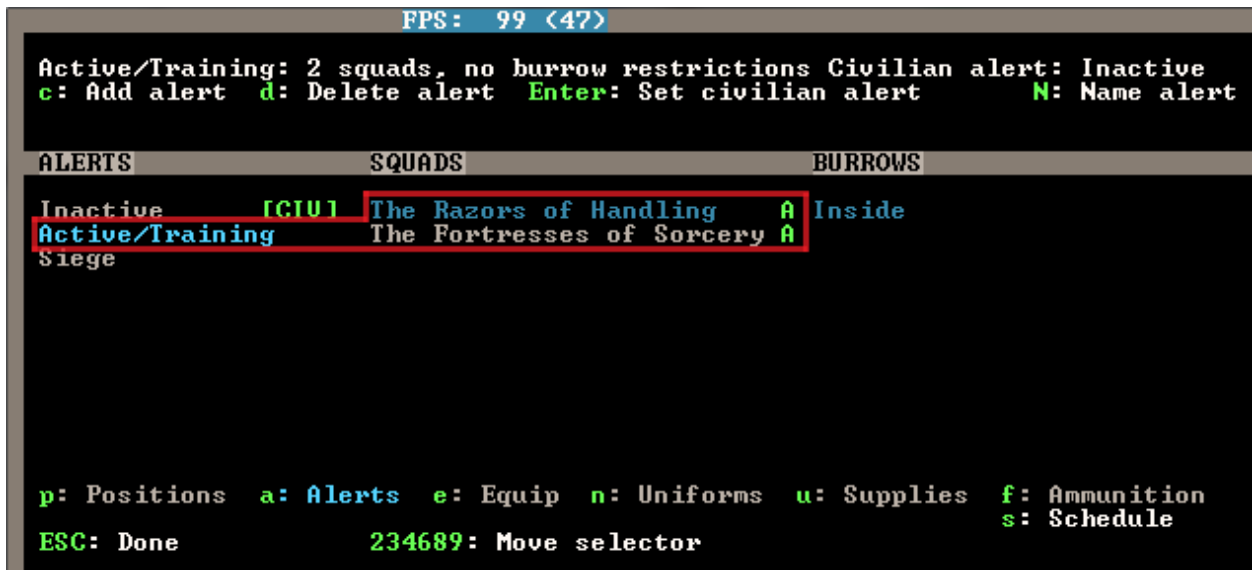
- With the military screen up, hit `w` for weapons.
- You now have a field you can move around using the arrow keys and cycle using `Enter`. Leave the dwarves unarmed.
- Scroll to the first `A`, which is for *armor* and hit `Enter` until the field shows *Lth* for leather. Then go to the next column and hit `Enter` until you see *Shd* for shield.

- If you look further down your weapons list you may note how woodcutters are armed with an axe. And note that any you've told to be woodcutters who don't have an axe won't cut wood... although they might try and arm themselves due to being tasked. Anyway...
- Space back out.

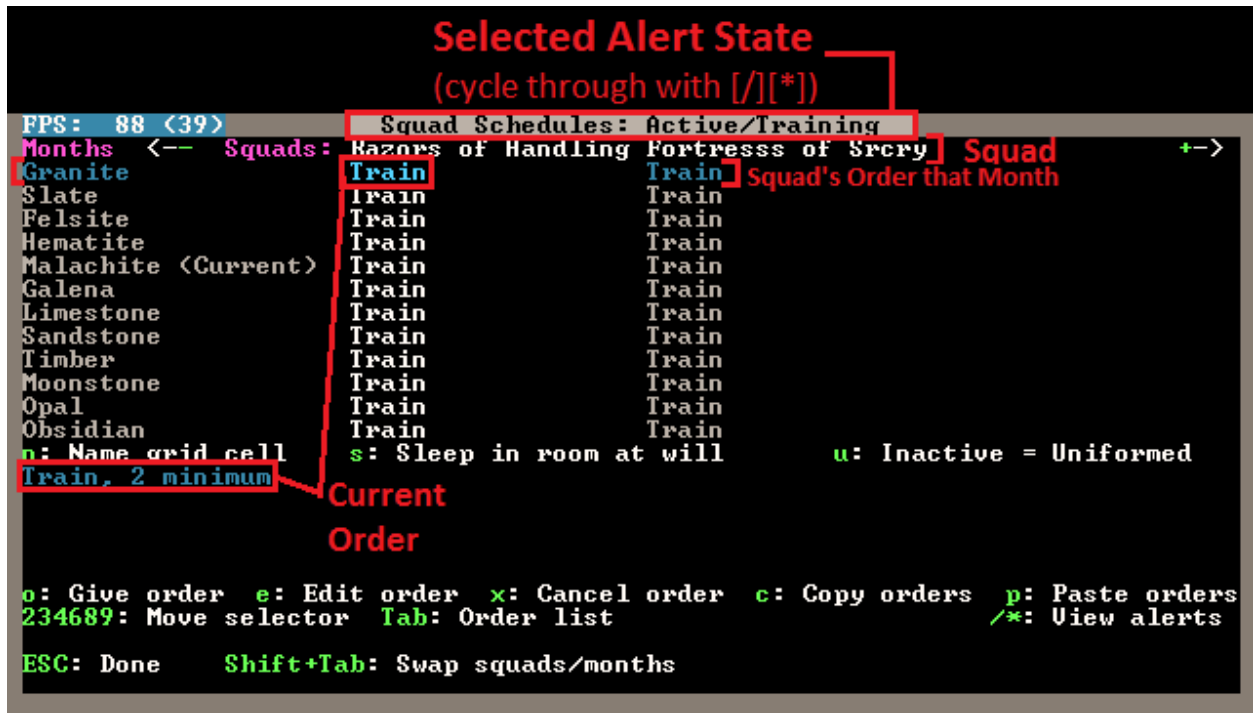
Your soldiers will now go and get a bunch of leather armor and a shield each. You did remember to make a bunch of armor as instructed earlier, right? They will then go to the barracks and spar. Why do we want them to spar unarmed? A few reasons:

1. Sparring unarmed trains the wrestling skill, and builds the key stats (like all training) but also trains quickly. Hammer weapon skill is the skill used by crossbow dwarves when out of bolts. Yes, they bash the enemy to death with their crossbows. We can train hammering latter.
2. Wrestling helps when an enemy disarms your dwarves as they have to fight unarmed then, and yes, a few enemies can disarm your lads.
3. Finally, picture newby dwarves with no skill training with razor sharp steel battleaxes. Yes, picture a few graves. Wrestling gets them tough enough to spar with decent weapons without a lot of weapon access micromanagement in the stocks screen (ie. forcing them to pick up wooden weapons and later on dropping them in favour of proper weapons).

Using the m military a alerts system, start training!



Specifically: we want them training in small groups, in every month. This is the fastest way to build martial skill, since they don't all stop if one lazy midget wants a drink.



One thing you might want to think about is providing food and drink and somewhere to eat close to your military dwarves. That way they will hopefully spend more time training and less time walking!



...hmm, actually, I have so many bloody dwarves I'm going to put each squad up to four members each. That means we'll have up to 8 dwarves sparing at once, good stuff. To add more dwarves, go to the military screen again, hit enter on the leader and then chose a dwarf to add to the squad.

Note: It's possible to end up with really convoluted command structures by accidentally making the boss of one squad the boss of another squad. This is annoying and pointless, as far as I can tell. To fix it, chose the problematic squad or person, hit enter on them, and then hit enter again. Hopefully that will remove them back up the chain of command. Or, hit enter on the boss and then enter on the dwarves 2 steps down, that will pull them back up to being directly under the boss's command.

While the boys are sparring go to your leather workshop and make sure you've got some backpacks, armor and helms on order. You can check your stocks too, to make sure there's enough kit to go around. In time we'll get our lads into steel chainmail and plate, but for now, we need to get them into leather, and yes, armor does stack later.

### 1.8.2 War Puppies!

A nice addition to your military are some war dogs. How do you get war dogs? Easy! Find a space in your workshop and **b** build a **k** kennel. This is a large building, so you may have to free up some space to place it. Then ensure you have someone with the animal training labour on (so it gets built and so you will be training animals).

Then, go to the kennels using **q** and hit **a** for add task and then **w** for *Train War Dog*. Hit **r** to set the task on repeat. Hopefully all the adult dogs will now go about being trained. You'll soon start getting messages like *An animal has become a stray war dog*. Good!

How can these help us? Well, we can either assign them to restraints we build - use **b**, **v** and choose either a rope or a chain; once placed use **q** to assign an animal to the restraint - or we can assign them to soldiers. Lets assign one to each soldier:

- Hit **m** to bring up your military.
- Hit **c** to look at a soldier.
- Hit **p** for *Preferences*.
- Hit **e** for *Work Dogs*.
- Hit **r** to assign a dog to that dwarf. Once you have enough trained, assigned one each.

You can also train hunting dogs at a kennels and they are quite handy for hunters, since they too have the ambush skill that Hunters use. I don't usually bother with hunter dwarves so wardogs is all I want!

### 1.8.3 Get tough, little guys!

So your boys are training, putting each other in arm bars and whatnot. If you use **v** and look around your soldiers, hitting **g** for General, you should see their skills climbing. Use **=** to see the full range.

Let them keep this up until they become champions or heroes. At this point they will no longer accept work dogs and become impossible to assign other jobs. The good news though, is that champions will actually use artifact items! Yay! So, let them train and while that goes on, keep digging as per the last chapter. Grid up all the digable space on the various levels and find gems, ore and whatever else is around!

Ok, lets assume your boys have trained up to pretty damn good wrestlers, getting very tough and agile in the process. What's the next step? Well, we'd like them to use chainmail, right? We could set them right into plate, but plate slows them down if they aren't very good armor users (they won't be yet). Chain will be just fine. Hopefully you've build a good pile of iron chainmail, helms, shields, grieves and whatever else you can see on the list. Now it's time to assign.

At the same time you've set a lot of iron warhammers to be built, right? Good!

But before we assign the weapons we want to make sure our soldiers don't pick up the wrong armor, right? So this is what we do:

- Go to your stocks menu under **z**. You should now see all your stocks updated by your bookkeeper, handy!
- Find the *weapons* first. Scroll through and look for any other hammers. Chances are *iron hammers* are all you have. For good measure, set everything that looks useless (ie, not an axe or a pick) to **m** for melt. Or put that trash into weapon traps.
- Go to the *armor* list (and subsequent clothing lists) and look for chainmail. Any chain item that isn't iron chainmail set to *forbid* using **f**. Now your boys won't go picking up the rubbish when you change their armor

setting. Make sure you've got enough suits of chainmail armor, shields, helms, etc, to equip your lads. Or near enough.

- If you're unhappy with how your armor settings turn out, set everyone to no armor and once they've dumped all their items, go back through the equipping process again.

Now, you can go to the `m` military menu and hit `w` for the weapons and change them from *Lth* to *Chn* and the weapon to *Ham*. Your dwarves will rush off, grab a warhammer and throw some chainmail on top of their leather armor, great! If a few of them get to be awesome armor users you can change them to plate and they will even put the plate on top of their chain on top of their leather. A tiny, cute, drunk, ball of fury!

## 1.8.4 MAD: Mutually Assured Dwarfiness

How do you use your dwarves then? Well, when they aren't mucking about your barracks working on their skills they can be out fighting bad guys! How I do it, when invaded is:

- Hit `m`.
- Go to a squad leader, hit `v` (*not* `a`, that turns them back to peasants – but `a` again will turn them back to squadies).
- Hit `t` to make the squad respond to orders.
- hit `z` to zoom to the commander.
- I then go to the level and place I want the squad to go.
- Hit `s` to station the soldiers there, or hit `p` to place a patrol point and then move your cursor around and hit `p` again to place another point (and keep hitting `p` to place more).

Your soldiers will now stop what they are doing and rush to the station point. Sometimes it's a good idea to set it in a safe place first, so you can at least get a few moving all at the same time. If you just drop the station point in the middle of the bad guys you risk your lads charging in one by one.

There's a lot of ways of handling military, and we haven't even covered crossbow wielding marksdwarves, multiple weapons, archery targets, or any of that stuff. For now, go and build your military and come back for *the next and final chapter* where we'll round out your lesson with some final touches!

## 1.9 That's All, Folks!

This is the final chapter of the Dwarf Fortress walkthrough.

### 1.9.1 Marksdwarves for Pest & Goblin Control

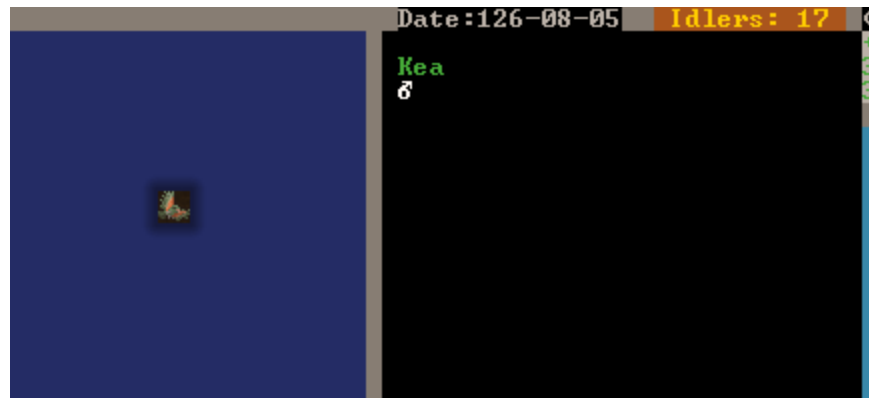
It was about this time that I realized that the grazing animals in their pastures outside had scattered and the dwarves attempting to round them up were fleeing in abject terror. The reason was clear: at the bottom of the screen I saw:

*Etur Ozarlolor, Farmer cancels Pen/Pasture Stray Water Buffalo Calf (Tame): Interrupted by Kea.*

Bringing up the *Units* screen by pressing `u` and scrolling right to *Others*, I select a *Kea*. Bracing myself, I press `v` and lay my eyes on the beast.

*A small, green, intelligent mountain parrot.*

Terrifying stuff. Let's locate it on the map. Go back to the *Units* screen, select a *Kea*, and press `z` to *Go to Unit*.



The keas are flying around up in the sky, out of reach of our military. This seems like a great time to organize a marksdwarf militia! Marksdwarves are an excellent tool for repelling invasions of non-undead creatures because they can engage from behind fortifications without exposing themselves to much danger.

First, let's build some archery targets.

- Press `b`, `A` (or find *Archery Target* at the bottom).
- Place four or more in a room somewhere. I built mine in the spare room next to the melee squad barracks:



- Hit `q` over a target.
- Hit `q` for *Make Archery Range*.
- Increase the blue size until it covers each of the targets neatly, hit `Enter`.
- Look at the top right. You'll see *Shoot: From Left to Right*. Use `wasd` to choose the proper direction. For me, I want them shooting down so I hit `s`.
- Repeat this on each target you placed. Unfortunately, a bug necessitates this micromanagement.
- Job done! Range is now ready!

Now, we need some warm bodies to fill the ranks.

- Make a squad of six or more slackers. Ten is great.
- Set them to use the *Archer* preset uniform. This includes leather armor (if you have any) and a crossbow of any material.

- Make a bunch of wood crossbows. Crossbow ranged attacks are identical across material type, so wood ones are fine. Wood crossbows are made at a *Bowyers Workshop* (b, w, b). Material type for ranged weapons only effects their damage in melee.
- Unlike shields, bucklers can be used simultaneously with crossbows but give a penalty to hit. Build some if you're feeling generous.
- Marksdwarves should only be used from behind *Fortifications* or against harmless pests, so armor is not necessary. But, if you like, armor certainly helps when an *Elite Goblin Crossbowman* starts threading bolts through your carefully-constructed *Fortifications*. If you decide to use metal armor, be sure to edit the archer squad's uniform accordingly with m, e, U.
- Queue up some more backpacks and quivers at the *Leather Works*.

Unfortunately, a bug often prevents marksdwarves from switching between bolts of different materials for combat and training. Therefore, we have to ensure that we use a single material type of bolt for training and combat. We have lots of bone lying around outside in our refuse stockpile, so let's make bone bolts for now:

- Build one or two *Craftsdwarfs Workshops* next to the outdoor refuse pile.
- Queue up *Make bone bolts* and set it on repeat or use workflow to make about 400.
- Hit m, a, f, to bring up the *Ammunition* menu.
- Select your marksdwarf squad. To ensure that only bone bolts are used in combat and training:
- Select the exiting bolts and hit d to delete them.
- Hit c, select *bolts*, press Enter, then hit M and select *bone*.
- Ensure that the resulting *bone bolts* are used in training and combat by pressing T and C.
- Change the amount to 200 with +/- . Your ammo screen should look like this:

FPS: 99 <49>		The Military of Kilrudod	
c: Add item	-+/*: Change amount	C: Use in combat	
d: Remove item	M: Material	T: Use in training	
SQUADS/GROUP	AMMUNITION	ASSIGNED	
Hunters	bone bolts <200>	CT	-reindeer bone bolts [5]-
Razors of Handling			-reindeer bone bolts [5]-
Fortresses of Sorcery			*reindeer bone bolts [5]*
Merchants of Conjrnjg			reindeer bone bolts [4]
			+reindeer bone bolts [5]+
			*reindeer bone bolts [5]*
p: Positions a: Alerts e: Equip n: Uniforms u: Supplies f: Ammunition			
s: Schedule			
ESC: Done		234689: Move selector	

- Make sure that your marksdwarf squad is set to *Active/Training* in the *Military* screen (m, a).
- Set up your Marksdwarves training schedule like you did with your melee squads. Hit m, s to bring up the *Squad Schedules* panel, navigate to *Squad Schedules: Active/Training* with / \*, and set your new Marksdwarf squad to train with at 2 or 3 minimum for every month.
- With everything set up correctly, your marksdwarves should grab some equipment and start practicing!



Due to a bug, dwarves sometimes cannot drop carried ammo in order to pick up ammo of a different material type, such as iron. This can cause them to never practice or even fail to engage the enemy because they cannot drop their ammo in order to pick up practice or war ammo. You may be affected by this bug if you eventually decide to switch from bone bolts to metal bolts, which are much deadlier. If you have problems, consult the DF Wiki.

### 1.9.2 Putting the Fortress in Dwarf Fortress

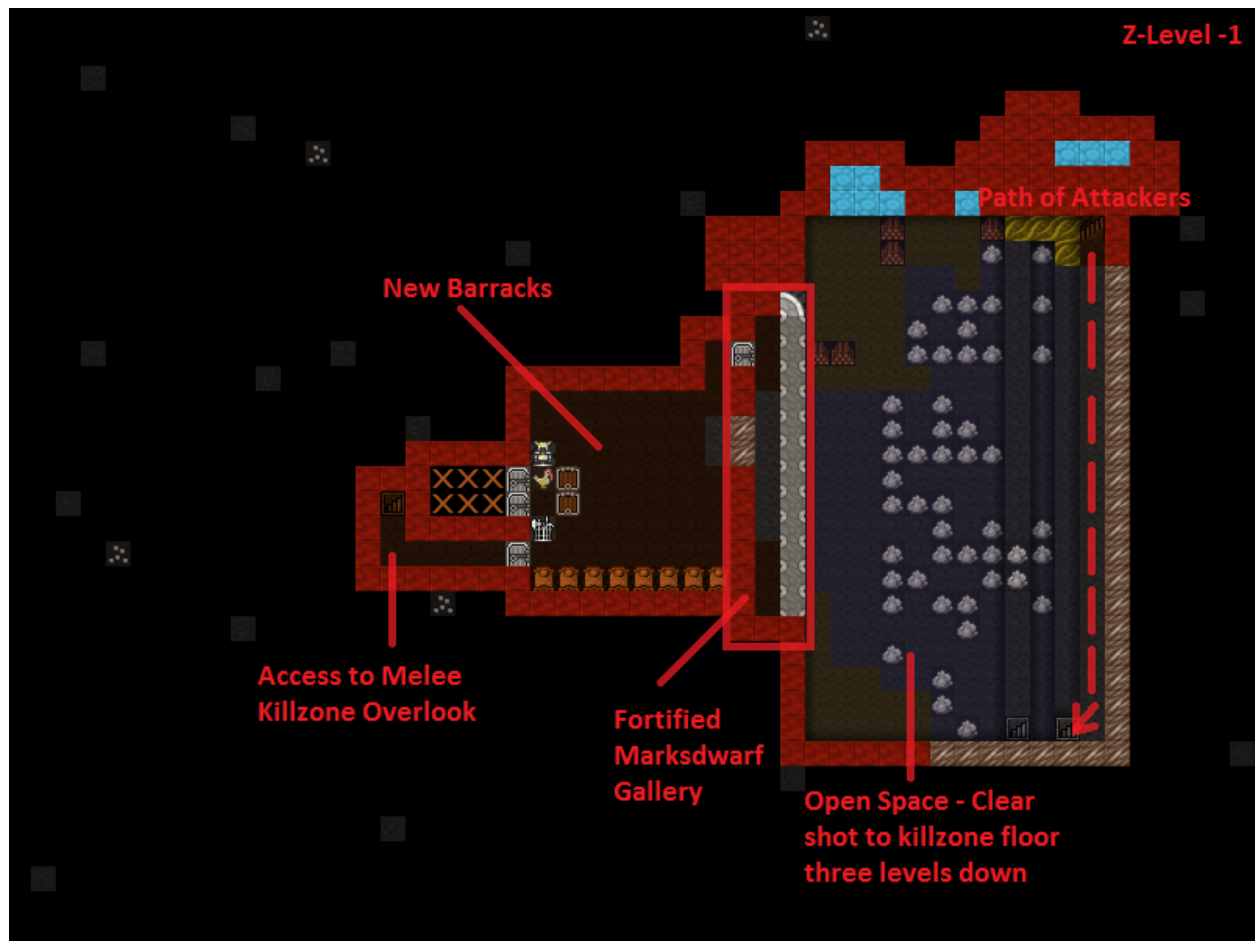
Congratulations on your new squad of marksdwarves! Besides *Kea* sniping, marksdwarves are great at defending from behind *fortifications*. Let's make some!

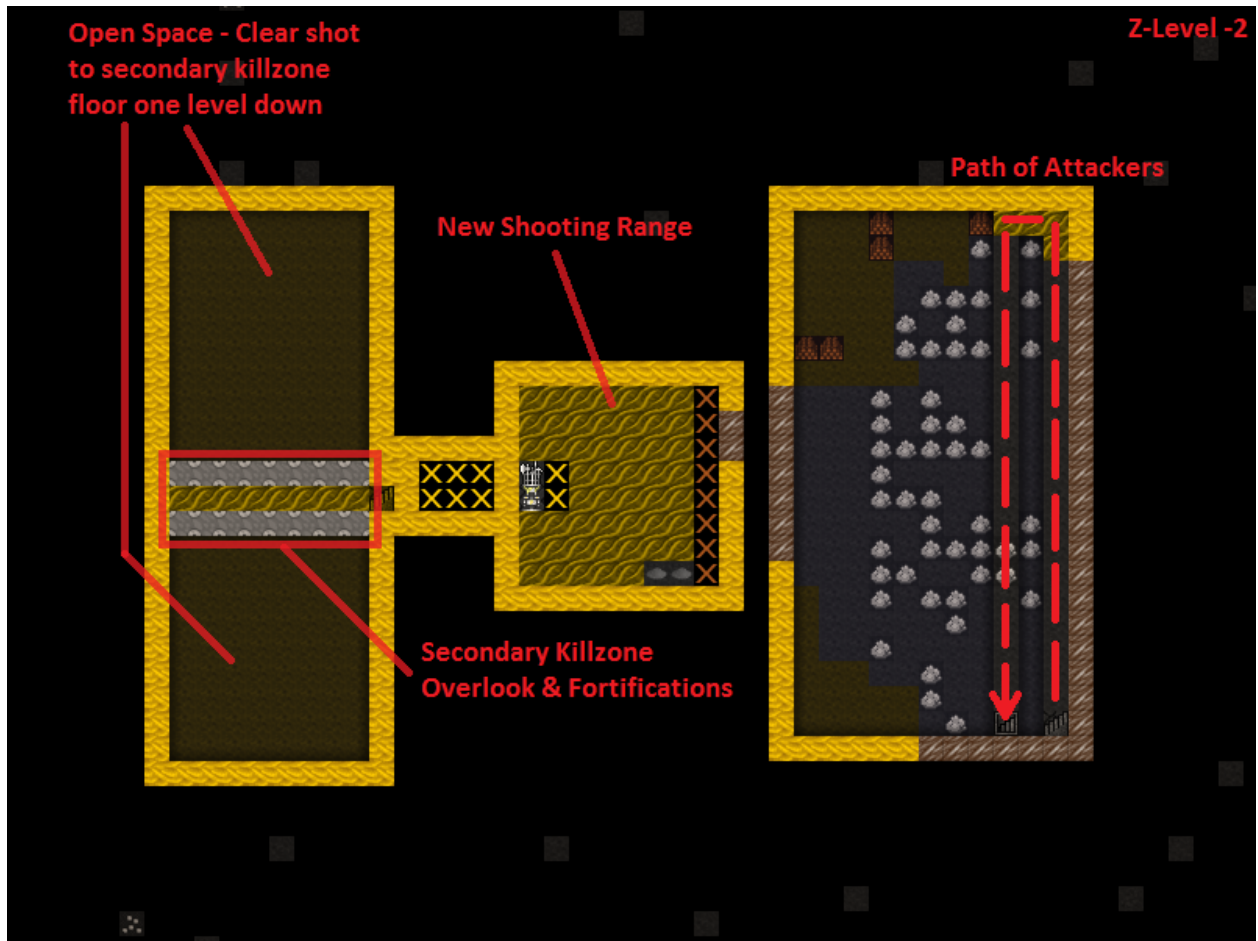
There are a few important things to keep in mind when building complex fortress defenses:

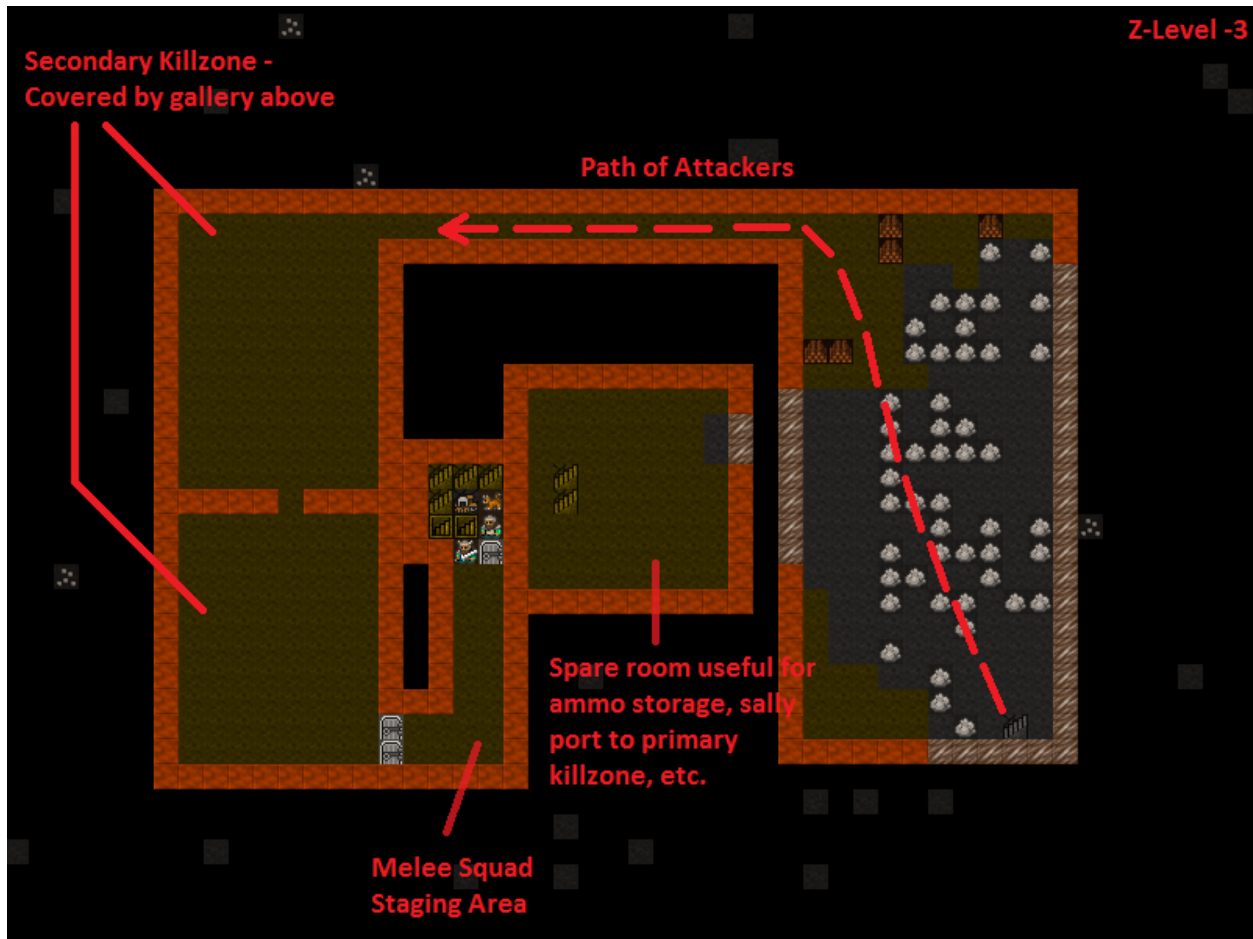
- Enemies will take the shortest path from the outside into your fortress.
- Some enemies can destroy doors and other furniture, but none can destroy closed draw bridges or constructed walls.
- Marksdwarves must stand adjacent to *fortifications* in order to shoot through them, so it's a good idea to make 1-tile wide halls adjacent to *fortifications* and use burrows to ensure they stand in the right spot.
- You should also account for the unexpected - attackers that can fly, jump, climb, etc.

The specific design of your fortifications and traps is limited only by your creativity. The design below is an example of maximizing the distance that attackers must cover while being exposed to marksdwarf fire. As the attackers wind down the descent, they are exposed to fire from the fortified gallery above. Then, they must get past the melee squads in the secondary killzone which is also covered by a fortified gallery. You can make far more complex (and satisfying) defenses than this - TinyPirate's automatic minecart shotgun is a noteworthy example!









Though formidable, these defenses are incomplete. Can you see the flaw? Our main stairwell is wide open, protected only by some traps! Remember that attackers will use the shortest route to get into your fortress, and right now the shortest route is over those traps. Let's build a drawbridge that we can raise to block off our main entrance and force attackers to take the route through the killzone.

- Hit `b`, `g` to bring up the *Bridge* placement menu.
- Use `u`, `m` and `k`, `h` to create a 3x3 footprint and place it at the entrance to the fortress, adjacent to the first line of traps.
- By default, bridges retract when activated. We want ours to raise to the left so that when raised, it blocks access to the fortress. Hit `a` to accomplish this.
- Hit `Enter` to place the bridge and use stone blocks to build it.
- A Dwarf with the Architecture labor enabled will come to design the bridge and a Mason will then construct it.

Now we need to hook the bridge up to some kind of mechanism to raise it. The most straightforward method is to connect it to a lever. Levers do not need to be near the machines they operate, so let's build ours in a high traffic area near the dining hall.

- Hit `b`, `T`, `l` and place the lever near the dining hall.
- Once the lever and bridge are built, hit `q`, place the cursor on the lever, and hit `a`, `b` to bring up a list of bridges it can be linked to.
- You should have only one bridge. If you have more, select the entrance bridge and hit `Enter` to *Assign* it to the lever.

- Use any old mechanisms. Mechanism quality only effects the time delay for traps to fire when sprung, not bridges.
- You'll see the *Link a Building to Trigger* job on the lever's `q` screen. A Mechanic will use the two mechanisms you selected to hook them up.
- To raise/lower the bridge, hit `q`, place the cursor on the lever, hit `a` to *Add new task*, `P`. If the situation is critical - eg during a siege - hit `n` for *Do task now!* and the closest Dwarf will run to pull it.
- Go ahead and test it out. Be careful: any object or creature on the bridge when it raises will be flung across the map! Bridges can be weaponized in this way.
- Lastly, build a stone wall north of the bridge to seal off the diagonal gap. Good job! Now we can funnel attackers into our kill zone with a pull of a lever!

The finished bridge in the raised position:



### 1.9.3 Assigning Battle Stations

Finally, it's a great idea to set up burrows for each killzone gallery. Although you can use `s`, `m` to move squads around, Marksddwarves have a bad habit of never standing in the right position to shoot the massive army bearing down on them. A failsafe method is to use the *Alerts* screen (`m`, `a`) in conjunction with burrows.

Remember the *Inside* burrow we set up to keep civilians inside? The same principle applies here. We will make a tiny burrow for each killzone gallery, create cooresponding alert states, and set up orders to defend the specific burrow when the alert state is active. This will cause the squad subject to the alert to immediately move to the correct burrow.

- Start by creating a burrow on the main killzone gallery encompassing the areas where marksddwarves must stand to shoot - directly adjacent to the fortifications. Name this burrow "Prm Killzn Gallery".
- Do the same for the secondary killzone gallery. Name it "Sec Killzn Gallery".
- Make a burrow in the melee staging area as well. Name it "Melee Staging".
- Hit `m`, `a` and create three new alerts with descriptive names like "Prim. Killzn. Gallery", "Sec. Killzn. Gallery", and "Melee Staging Area." Use `c` to *Add alert* and `N` to *Name alert*. Here is how the alerts panel should look:

```

Prim. Kllzn. Gallery: no squads, no burw rstrctns Civilian alert: Inactive
c: Add alert d: Delete alert Enter: Set civilian alert N: Name alert

ALERTS          SQUADS          BURROWS
Inactive        [CIU]          The Razors of Handling      Inside
Active/Training The Fortresses of Sorcery  Prm Kllzn Gallery
Siege           The Merchants of Conjuring Sec Kllzn Gallery
Prim. Kllzn. Gallery
Sec. Kllzn. Gallery
Melee Staging Area
Melee Staging

p: Positions a: Alerts e: Equip n: Uniforms u: Supplies f: Ammunition
ESC: Done      234689: Move selector s: Schedule

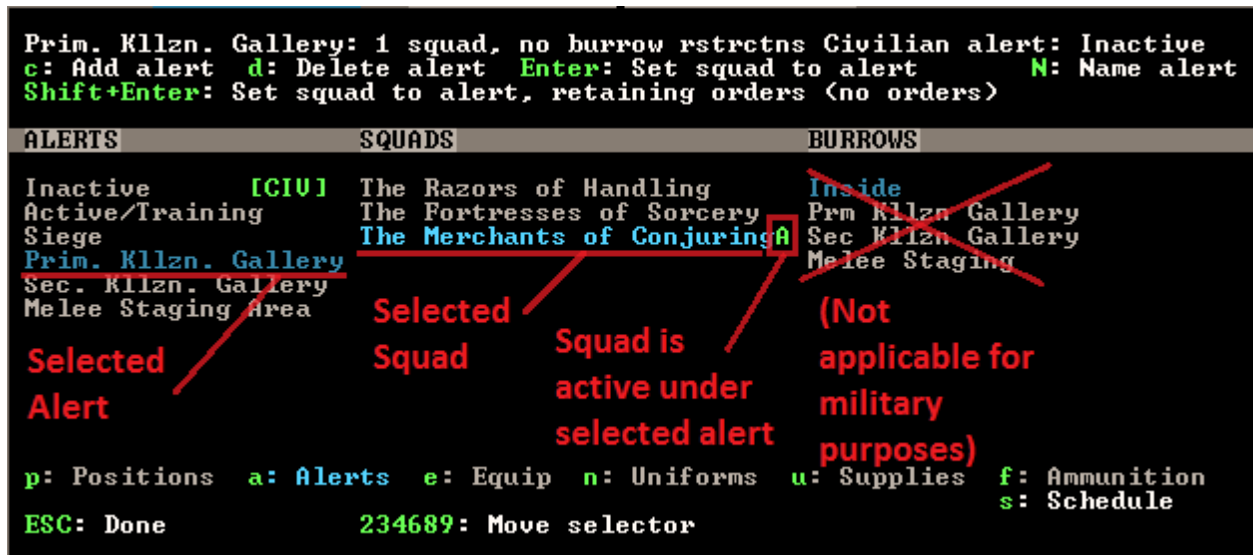
```

Now we will associate each of these alerts with a cooresponding burrow so that when a squad is set to, for example, “Prim. Kllzn. Gallery” alert, it moves to occupy the “Prm Kllzn Gallery” burrow. The game calls this an *Order* and manages orders with the *Squad Schedules* interface.

- Hit `m`, `s` to bring up the *Squad Schedules* interface. Make sure our first melee squad column is highlighted.
- Use `/`/`*` to cycle to the *Prim. Kllzn. Gallery* alert (displayed in the gray bar at the top of the screen).
- Hit `o` to *Give order*. Hit `o` again until the *Defend burrows* order appears.
- Highlight *Prm Kllzn Gallery* and press `Enter` to *Select* it.
- Use `-`/`+` to set the minimum number of soldiers to the number in the squad as displayed on the right. Each of our melee squads should have three soldiers, so get it down to *3 soldiers minimum*.
- Press `Shift-Enter` to enter the changes.
- Hit `c` to copy the order and `p` to paste it to each month for our first melee squad. We can also paste the order for each month under our second melee squad because that squad also has three dwarves.
- For the Marksdwarf squad, do the same steps above but instead of three soldiers minimum, use however many soldiers are in the Marksdwarf squad. I have ten and I want them all to show up when I give the order, so I set 10 minimum.
- “Prim. Kllzn. Gallery” alert is set up!
- Now just do the same steps for each separate alert, but use the appropriate burrow for each. On the *Squad Schedules* panel, press `/`, `*` to cycle to the next alert over, *Sec. Kllzn. Gallery* and set it up so that the *Sec Kllzn Gallery* burrow is used. After you’re done setting up that alert, do the same for the last alert - *Melee Staging Area*.

To order a squad to man a position, hit `m`, `a`, highlight an alert, use `6` or the arrow keys to highlight the squad you want to move, and press `Enter` so that a green *A* appears next to the squad. The squad will execute the *Defend Burrow* order we set above and move to the burrow with weapons out and armor on. Although somewhat cumbersome, this will guarantee that your Marksdwarves man an effective fighting position at the critical time. Go ahead and give it a try.





With “Prim. Killzn. Gallery” highlighted, the green A shows that The Merchants of Conjuring are under this alert and are moving to the “Prm Killzn Gallery” burrow.

## 1.9.4 Further Fortifications

The surface entrance to our underground deathtrap is a choke point for attackers and will cause them to bunch up as they try to get in. Let’s build a tower to shower them with death! Here is a design I came up with that uses the existing stairs that run through the barracks. As a bonus, it covers the entrance way as well.

The main Marksdwarf gallery is directly above the drawbridge. It uses the terrain to cover the north and is completely sealed off from the outside.



A stone roof seals it from above. Create roofs by building floors on the level above: b, c, f. Note: walls create floors on the square directly above them, but fortifications do not. Remember to build floors above all fortifications or nasties will drop in from above!



Access is through a staircase...



... that leads to the barracks below ground.



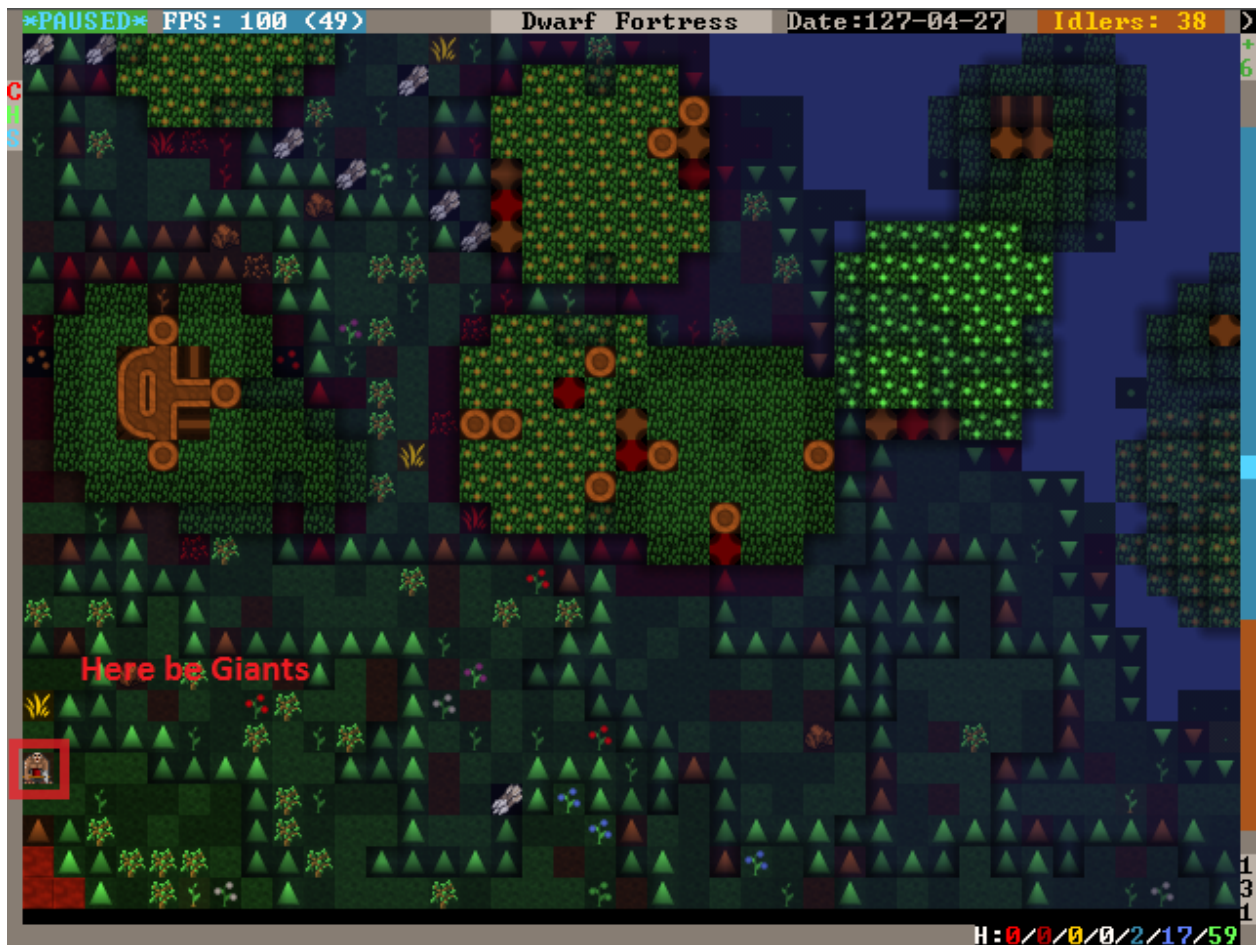
Don't forget to set up a burrow for this tower just like in the previous section!

### 1.9.5 Man Your Battle Stations!



The first test of my defenses came somewhat after this when a *Giant* appeared on the southwest corner of the map:





Lets see how tough our defenses are. Our priority is to get everyone inside, batten down the hatches, and man the outer defenses.

- Set the civilian alert to *Siege* (m, a, then select siege and press Enter). This restricts civilians to the *Inside* burrow and will cause them to drop everything and run inside. Now is also a good time to ensure that your *Inside* burrow is in fact inside and out of the line of fire!
- Station our Marksdwarf squad in the primary killzone gallery. Hit a, select the *Prim. Kllzn. Gallery Alert*, move right to the *Squads* column, highlight the Marksdwarf squad, and hit Enter. A green A next to the Marksdwarf squad means that they will move to that burrow.
- Station our Melee squads in the melee staging area. Use the same procedure as above, but activate the *Melee Staging Area Alert* for them.
- Now the tricky part. We need to shut our main drawbridge when everyone is inside but before the *Giant* arrives. A good method is to pause the game and use . to advance time incrementally. Keep your eye on that Giant! If anyone is left outside, they will have to outrun the Giant as it comes in through the killzones.
- If you lose track of the Giant, use the *Units* screen (u) to find it in the *Others* submenu and hit z to *Go to Unit*. Then press f to follow it.
- When the time is right, close the drawbridge by hitting q, moving the cursor onto the lever, and hitting a, P, n. Bridges shut slowly, so give lots of time!

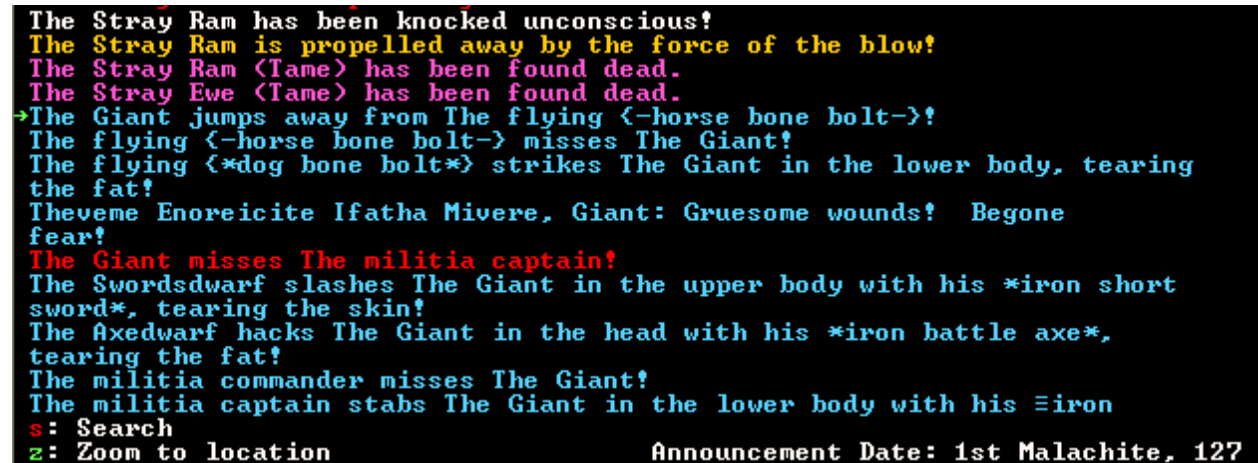
The gate shuts, our dwarves are in position, and the *Giant* is making a beeline for the killzones. Perfect. Just remember to reposition your Marksdwarves if the *Giant* makes it underground!

I mistime the throw of the lever but the *Giant* is distracted by the strategic diversionary livestock pastured outside, giving time for the gate to shut! Those poor sheep, though...

The Giant pauses to destroy the Crafts Dwarf's Workshops outside, then descends into the killzones...

and is struck down by the waiting melee squads.

Our defenses worked! It's always fun to look at the combat log - hit `r`, scroll down to *the Giant Theveme Enoreicite Ifatha Mivere is fighting!* and hit `Enter`.



```
The Stray Ram has been knocked unconscious!
The Stray Ram is propelled away by the force of the blow!
The Stray Ram <Tame> has been found dead.
The Stray Ewe <Tame> has been found dead.
→The Giant jumps away from The flying <-horse bone bolt->!
The flying <-horse bone bolt-> misses The Giant!
The flying <*dog bone bolt*> strikes The Giant in the lower body, tearing
the fat!
Theveme Enoreicite Ifatha Mivere, Giant: Gruesome wounds! Begone
fear!
The Giant misses The militia captain!
The Swords Dwarf slashes The Giant in the upper body with his *iron short
sword*, tearing the skin!
The Axedwarf hacks The Giant in the head with his *iron battle axe*,
tearing the fat!
The militia commander misses The Giant!
The militia captain stabs The Giant in the lower body with his ðiron
s: Search
z: Zoom to location      Announcement Date: 1st Malachite, 127
```

Scrolling up past the pages of Dwarves beating on the *Giant*, I notice that the *Giant* dodged a bolt when it entered the killzone. Look back at the second GIF above. Do you see what happened? The *Giant* appeared on the first level of the killzone walkway and immediately came under fire. When it dodged, it went over the side and landed on the second walkway! It was only a 1-z level drop so it did no damage, but what if we lined the killzone with iron spikes? There's always room for improvement in fortress defenses!

---

**Note:** Don't forget to set your squads back to *Active/Training* and civilians back to *Inactive* on the alerts menu!

---

## 1.9.6 Medic!!

If you've made it this far without a dwarven casualty, you are extremely lucky! Injury is a certainty in Dwarf Fortress, but adequate healthcare will keep your maimed citizens from succumbing to blood loss or infection. Dwarven healthcare is complex to set up, but a hospital system will keep even badly mangled dwarves in the army/workforce and prevent fort-ending tantrum spirals due to loss of loved ones. Once running, the whole thing is automated and requires little management.

A hospital requires:

- A source of fresh water - preferably not muddy to minimize infection.
- Cloth - for bandages.
- Thread - for sutchurs.
- Splints - to set broken limbs.
- Crutches - allow injured Dwarves to be useful while recovering.
- Buckets - to wash and give water to patients (patients can drink only water in the hospital!)



- Beds - allow patients to be diagnosed, rest, and receive water.
- Tables - for surgical operations.
- Traction benches - immobilize patients when setting bones.
- Soap - to clean wounds and prevent infection.
- Gypsum powder - for casts.
- A chief medical dwarf - to diagnose injuries, assigned in the nobles screen (n).
- Doctors - Dwarves with the various healthcare jobs enabled.

That sounds like a whole lot, but you likely have most of it on hand. Get started by designating a hospital zone. I evicted everyone from our original set of bedrooms and hollowed out those rooms to be my hospital zone.

- Hit i, place the zone and h to make it a hospital.
- Build and place beds and traction benches in the hospital zone with tables adjacent to them. Adjacent tables make it easier to move a patient for surgery.
- You probably will need to build some chains or ropes for the traction benches. The traction benches themselves are made at the Mechanic's workshop using one chain/rope, one mechanisms, and a table.

Here is a setup with tables and traction benches clustered around the beds:



- Hospitals require containers to store supplies. Hit b, h to build six or so *Containers* within the hospital zone. Coffers are best because bags are used in other industries.

- Once the containers are placed, Dwarves will start stocking the hospital zone with supplies. Let's take a look at those parameters. Hit `i`, place the cursor over the hospital zone, and hit `H` to *Set hospital parameters*.



Already, we have enough *Thread* and *Cloth*. Note that hospital item counts are a thousand times the actual units, so you don't really need 75,000 stacks of cloth. Let's move on to the other items.

- That bucket count looks low. Select *Buckets: 0/2* with `2/8` and use `6/4` to increase it to 10.
- Queue up 10 splints, 10 crutches, and 20 buckets at the Carpenter's Workshop.
- Build a *Kiln* with `b, e, k` and queue up 10 *Make plaster powder* jobs. Each job requires a bag and a boulder of gypsum or other plaster-making stone, and we have gypsum in abundance.
- Build a *Soap Makers Workshop* (`b, w, S`) and an *Ashery* (`b, w, y`).
- Soap's recipe is:  $\text{Lye} + \text{Tallow/Oil} = \text{Soap}$ . Lye is made from ash at a Wood Furnace. Queue up 10 units of ash at the Wood Furnace and 10 units of Lye at the Ashery. Finally queue up 10 units of soap (*Make soap from tallow*) at the Soap Maker's Workshop. If you have no tallow, butcher an animal and a Dwarf will automatically render the fat into tallow at a Butcher's Shop.

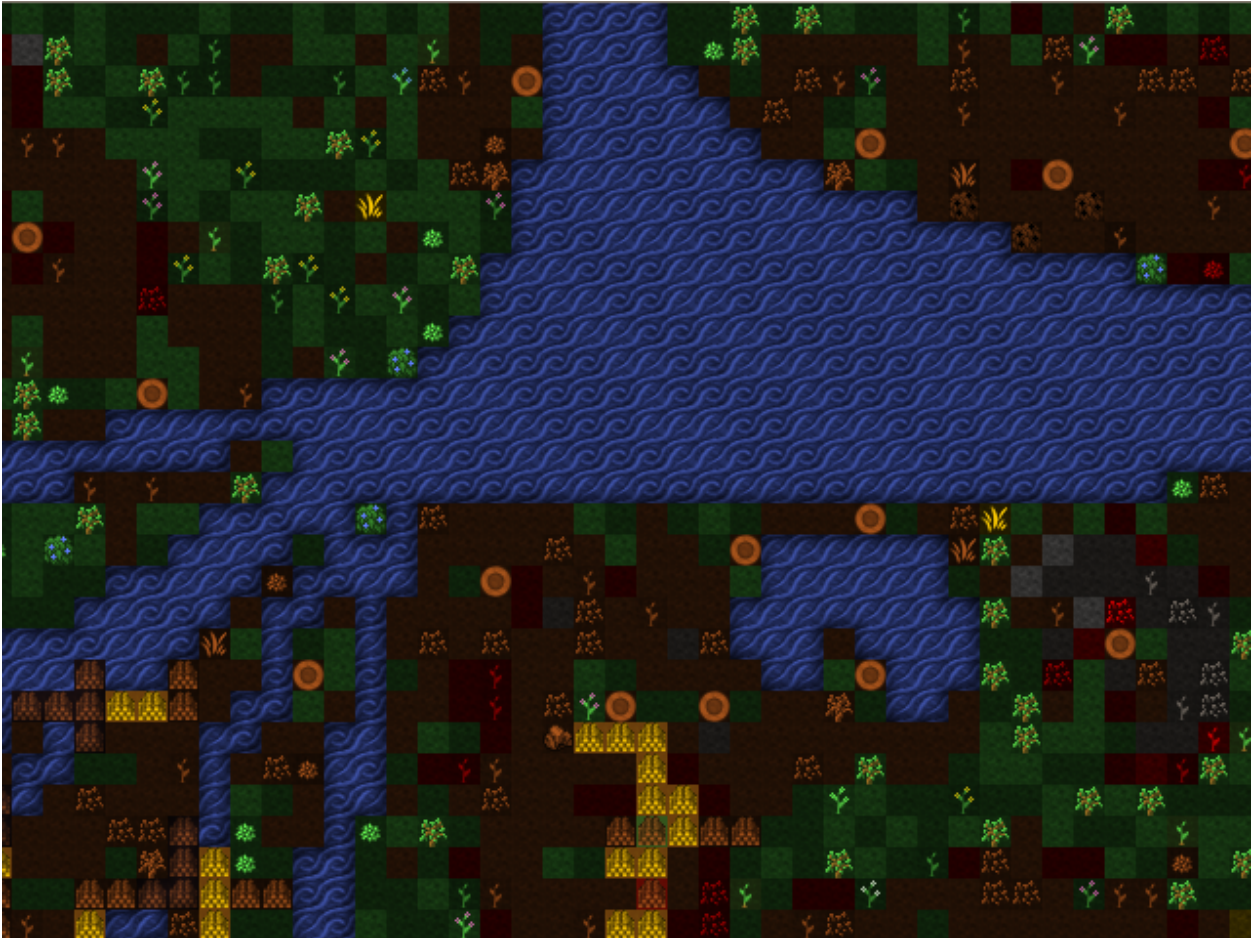
Great work! While your dwarves execute all the orders you've put in, let's secure a source of fresh water. Easy access to fresh water is essential because injured dwarves cannot drink booze. Moreover, water must be brought to them as they lay injured in bed. Therefore, it's critical to have a year-round source of fresh water close to our hospital. Otherwise, injured dwarves will die from dehydration in their hospital beds.

Let's take stock of the fresh water sources we have available.



To the east we have a number of small ponds. When we look at the water in them with `k`, we see *Murky Pool* filled with *Stagnant Water* [7/7]. Murky pools are no good to us because *Stagnant Water* increases the risk of infection when used to wash wounds and also creates an unhappy thought in any Dwarf that drinks it. Moreover, these pools are frozen for half the year! That pretty much rules these out unless we have no other choice.





Fortunately, we have a small river in the northeastern portion of the map. Using `k` to look at its water, we see that it's fresh, wholesome, *Water* [7/7]. This is what we've been looking for. The only problem is that like the *Murky Pools*, the river is frozen much of the year and it's very far from the hospital. However, if we divert the water into an underground cistern, it will not freeze and we can accumulate enough to last us through the cold months. Let's do some hydrological engineering!

First, let's dig out our cistern. It's best to locate it near the hospital and it needs to be at least two z-levels deep to avoid mud contamination. If water is taken from a 1 z-level source and the floor has a *Pile of mud* on it, the water will cause infections. The easiest way to avoid this is to make the cistern two or more layers deep. Here is where I placed mine:

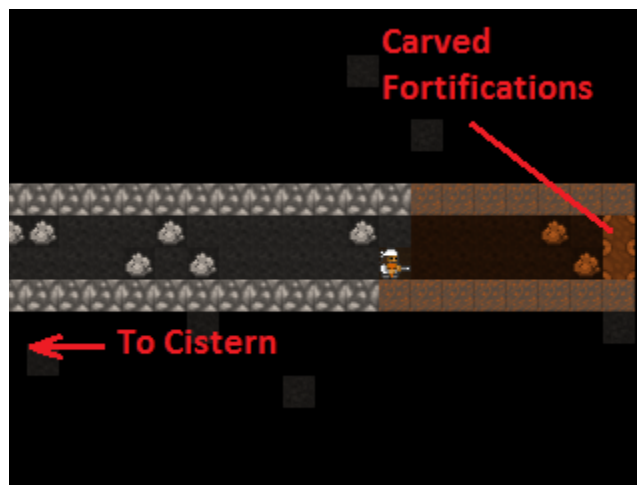


It goes down an additional 3 z-levels directly below the well room:



Now we have to bring the water to the cistern. Easy, right? Just dig a really long trench from the river and connect it up. Actually, that's a recipe for a flooded fortress! Water pressure will act to force the river water - flowing from above our cistern - up over the top of it and into the rest of the fortress. So, we need to use floodgates controlled by levers to manage the water flow. The only wrong way to accomplish this is the one that floods the fortress, but below is an example scheme that prioritizes flood safety.

- Queue up 20 floodgates from the Masons.
- Stick a door on the well room. If the worst happens, you can always lock that door to stop the water until you figure out what to do. Also, keep pets out of it by pressing `o` with the door selected - pets have a bad habit of drowning in cisterns and spillways.
- Dig a spillway tunnel from the top of the cistern to the edge of the map, then *Smooth* (`d, s`) and *Carve fortifications* (`d, F`) on the stone at the edge of the map. That will allow water to flow off the edge of the map in the event of overflow, relieving pressure in the cistern.

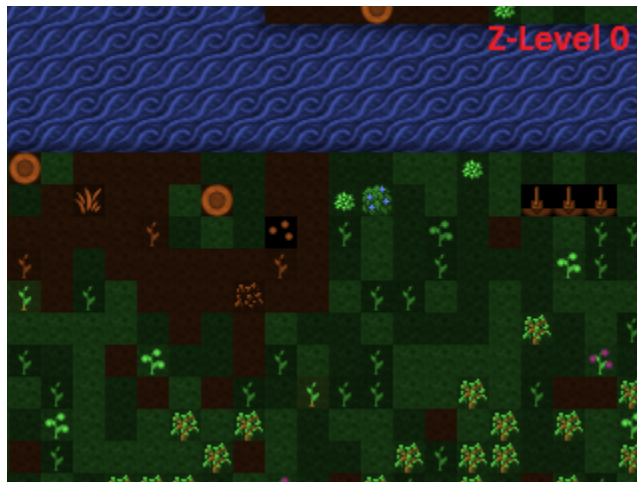


- Build *Floodgates* (`b, x`) between the cistern and the spillway. Make sure to leave one tile free adjacent to the floodgates to allow access for when we hook them up to the control lever.
- Build a *Lever* (`b, T, 1`). Place it in a location where you'll remember what it does and where it will be accessible in the event of a flood. You can use a *Note* (`N`) to mark it, too.
- Connect the *Lever* to the *Floodgates*. Hit `q`, place the cursor on the *Lever*, `f`, and select one of the floodgates you placed, using any old mechanisms. Then hit `q`, `f` again and select the other floodgate to hook it up, too.

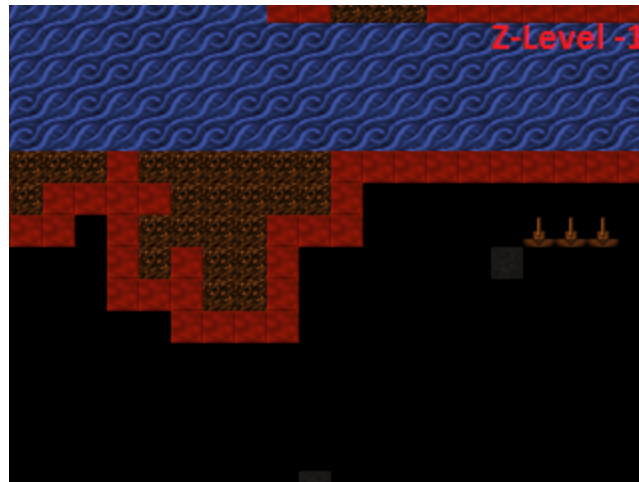
- When the Mechanic finishes these jobs, test out the *Lever* by queueing a *Pull the Lever* job (a, P). If both floodgates open, it's all set up! Close them again when you're satisfied.



Next, let's dig the water channel from the river. Start at the river and *Channel* (d, h) a 3x1 section as pictured. Make sure to leave at least one tile intact between the river and the channel! Otherwise, the channel will flood prematurely. When we are ready to tap the river, we will *Channel* out this canal to let the water flow.



- We need to make a ramp all the way down to the top of our cistern at z-level -6. Remembering that each *Channel* job will create a ramp on the tile below it, we move one tile south and one z-level down for each subsequent *Channel* designation.



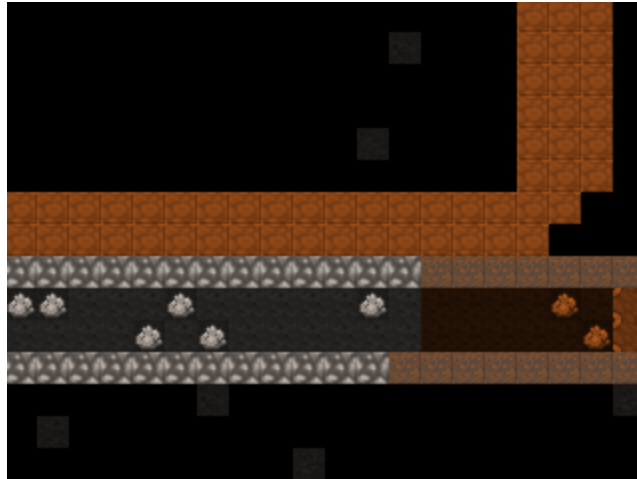




- At z-level -6, *Dig* a 3x1 tunnel south to the spillway tunnel...



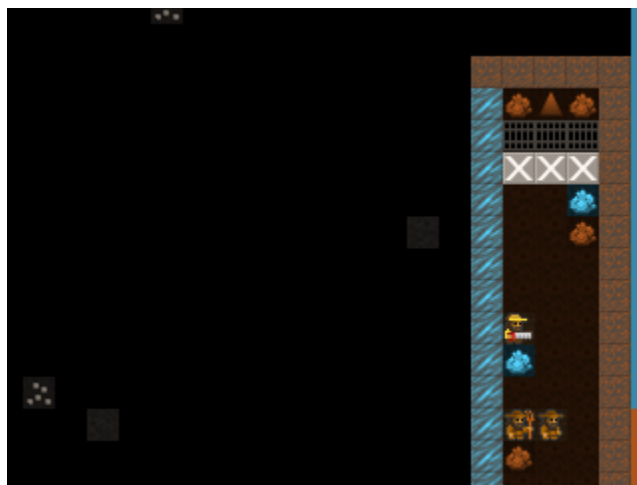
- then dig west until the cistern.



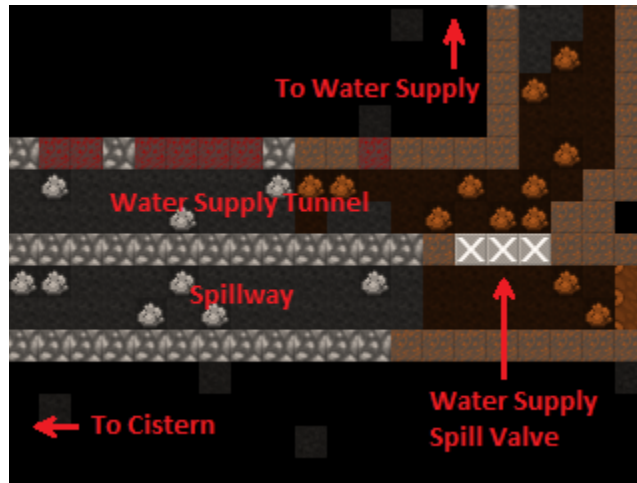
- Dig out two wall tiles at the cistern.



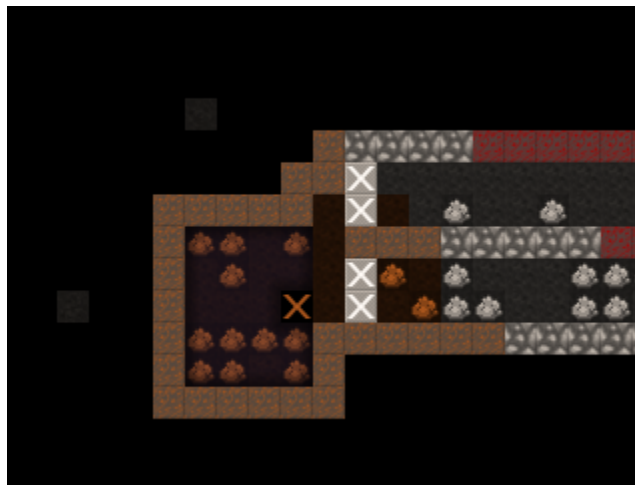
- Build *Vertical Bars* (b, B) across the water supply tunnel to keep the surface riffraff out. Although bars can be destroyed by a building destroyer (like that *Giant* from earlier), we will make sure that the bars stay submerged and inaccessible to all but aquatic building destroyers.
- After the *Vertical Bars* are built, place three *Floodgates* across the tunnel to the south of the *Vertical Bars*.



- Dig a 3x1 relief valve to connect the water supply tunnel to the spillway tunnel. Then install *Floodgates* across it. Safety first!



- Install the last *Floodgates* between the water supply tunnel and the cistern. If dwarves get stuck behind them, move on quickly to the next step where we install levers to open the gates.



Build three *Levers* (b, T, 1) to control the three sets of floodgates. It's important to keep track of which lever controls which floodgate, so I like to build a mini facsimile of the tunnel network and place the levers where the floodgates are located. It's also a great idea to put a *Note* on each lever (N, p) and name it with n so you don't forget. Here is the scheme that I came up with:



- Hook up each *lever* to the corresponding set of *floodgates*. Select the correct *floodgate* with +/- and use any old mechanisms.
- Test out the levers after they are hooked up to ensure that each one operates the appropriate set of floodgates.
- Stick a door on the entrance to your lever room in order to further floodproof it.

Floodgates can become stuck open if debris like boulders block their tile when open, and moving water tends to push boulders under open floodgates. This floods the fortress. Therefore, we must remove all boulders from our waterworks.

- Hit `i`, designate a 1x1 zone somewhere outside the waterworks, and hit `d` to make it a *Garbage Dump*.
- Hit `d`, `b`, `d` to enable *Dump Items* designation. Designate all stone in the water supply tunnel and spillway tunnel to be dumped. Our workforce should be large enough by this point to get the job done relatively quickly.

It's time to remove the dike between the river and the water channel, but it's a good idea to take some precautions when doing so because dwarves tend to path through fast-moving water and get swept away. We'd like to keep our Legendary +5 Miners alive despite their idiocy.

- Ensure all floodgates are closed.
- Hit `:kbd'd'`, `:kbd'o'`, `:kbd'r'` and designate a *Restricted Traffic Area* encompassing the dike and the ditch. This will make dwarves prefer any route other than over the designated area and hopefully clue them in that walking over it means certain death.
- *Channel* out (`d`, `h`) the middle tile of the dike at the river. Then, *Channel* out the outer two tiles.
- If anyone drowns, build them a *rock Slab* at a mason's workshop and have a Crafts dwarf *Engrave Memorial Slab*. Then place it next to the river so the idiot's family will know why he died. Disregard any dead bodies in the water supply.

We now have *Water [7/7]* from the river held back by the water supply tunnel floodgates at the bottom of the ramp we dug. That means that everything above that point is completely submerged by water. Therefore, if we opened both the water supply tunnel floodgates and the cistern floodgates at this point, the water pressure would push water all the way back up to z-level -1 - the level of the river. Consequently, we must ensure that at no point is the cistern opened directly to the river.

- Ensure all floodgates are closed.
- Open the water supply tunnel floodgates at the base of the ramp and allow the water to fill the entire water supply tunnel.
- When the water has reached *[7/7]* for most of the water supply tunnel, close the water supply tunnel floodgates.

- Confirm that the water supply tunnel floodgates at the base of the ramp are closed, then open the cistern floodgates.
- Allow the cistern to fill. Due to our design, the cistern will not flood so long as the floodgates leading to the river remain closed. There's nothing to do but wait!
- While the cistern is filling up, hit `b, 1` to build a *Well* on the open space above the cistern. Use the highest-quality materials available because dwarves get happy thoughts when they use a well-put-together well.
- When the well is complete, hit `i`, designate a 3x3 zone in the well room, and hit `w` to make it a water source.
- If you feel like enhancing the nanny state of your fortress, you can put some *Horizontal Bars* over the river water intake ramps at z-level 0 and a *Hatch* over the well access stairs. Forbidding the *Hatch* will at least keep the dwarves from drowning themselves at that location.

The cistern is filling nicely.

- Check the water level the top level of the cistern (one z-level below the well room) using `k`. When it is at `[7/7]` or `[6/7]`, close the cistern floodgates. If it stabilizes before reaching at least `[6/7]`, close the cistern gates, refill the water supply tunnel, seal it from the river, and reopen the cistern gates.
- At this point we could leave the cistern floodgates open indefinitely. So long as the water supply floodgates are closed to the river, the cistern will not overflow.

In the (extremely likely) event that something goes wrong and the fortress begins flooding, take immediate action to:

- *Forbid* and *Keep Tightly Closed* (`q, 1, o`) the door to your lever room and all doors leading to the flooding area. Doors are watertight so long as they stay shut.
- Open all floodgates to your spillway. Use `a, P, n` on the *Levers* for maximum haste.
- Close all other floodgates.
- If things are particularly dire, extend your central staircase downward until you hit the caverns. The water will tend to flow down your staircase and into the caverns where there is usually enough room for it to dissipate. That may buy you enough time to dig a permanent spillway or seal off the flooded parts of the fortress, but it also opens yourself to visits from uninvited guests.

Our last task is to staff our new hospital. We need to appoint a *chief medical dwarf* so that patients get timely care.

- Hit `n` to bring up the *Nobles* panel. Scroll down to *chief medical dwarf* and hit `Enter`.
- The first dwarf listed is the best diagnostician available, so highlight him/her and hit `Enter`.

With that, we have a functional hospital! Dwarves can recover from very severe injuries with a good healthcare system in place, so don't be afraid to send your melee units into combat. Now, broken bones will build character, not body count!

### 1.9.7 The End!

And welcome to the end of the tutorial! By now you should be able to play the game on your own. You can also check out the tutorials in the [index](#), for short chapters that cover a particular topic.

You might consider a mega project like an enormous tower, a castle, or a fortress entirely surrounded by magma. There are many options, and just surviving the increasingly severe raids from goblins and other critters can be a challenge.

You're also ready to upgrade to the [Starter Pack](#), and explore the wider world of DF addons. From alternative graphics to advanced tools or a real 3D visualiser... the world is your oyster!



The walkthrough covers the topics you'll need in every fortress, in just enough detail to get you started. After that, it's your decision what to focus on - the options are endless, from beekeeping to soap-making to digging too deep, or learning to make and install mods.

Each tutorial is a short, self-contained introduction to a topic which is likely to be of interest to players who have just finished the walkthrough. And if you're most interested in something not listed here, [the wiki](#) and the Bay12 Forums will be your best friends.

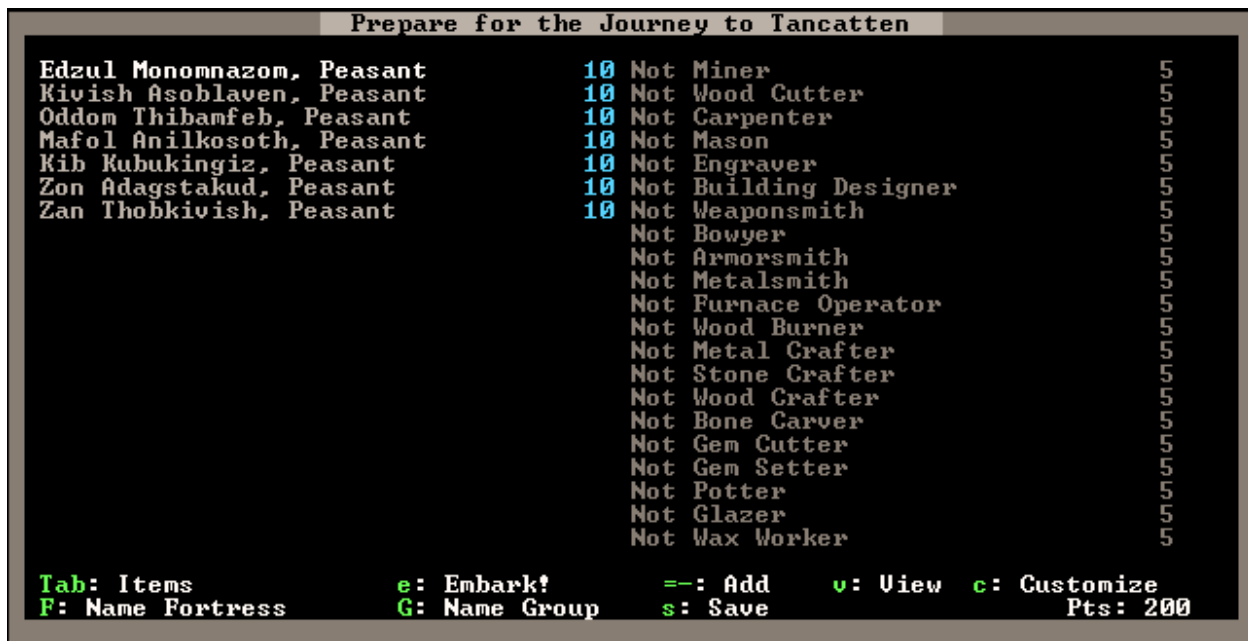
## 2.1 Custom Embark Strategies

The custom embark feature of Dwarf Fortress is a handy part of the game as it lets you customize exactly what sort of dwarfs you take with you and allows you to set their supplies to your exact specifications. Read on for more information!

### 2.1.1 Embarking the custom way

So you've chosen your perfect location, as per the world generation tutorial, and now you're ready to embark. What you want to be doing now is selecting *Prepare for your journey carefully* from the embark menu. You will now be presented with your list of dwarfs and the skills they can access.





Have a fiddle with your controls. The left-right arrows move between the list on the left (dwarfs) and the list on the right (dwarf skills). Select your first dwarf, then arrow over to the skill list and scroll through it. Currently this dwarf has no skills assigned. Find *Mining*, hit `-` and keep hitting it until the dwarf is set as a *Proficient Miner*. Do the same for *Mason*.

You have now made your dwarf a pretty good mason and a miner! Note, there's a zero next to his name? This means he can't accept any more skills. Most of the time you're going to be giving each dwarf a couple of skills, but one or two dwarfs might be given some outlier skills (leadership skills, for example) to ensure you have a broker from the get-go. There are many strategies when it comes to skill selection, we'll cover some samples later.

Meanwhile, get back to a dwarf using the arrow key. You'll see at the bottom the `c` menu for *Customize* and `v` for *View*. Hit `v` and have a read about your dwarf's personality. I usually look for the "friendliest" and mark them in my head for any possible leadership skills.

Head back to the list (`Esc`) and then hit `c`. Here you can modify a dwarf's nickname and profession name. This is handy if you want to make a note of how you've skilled each dwarf, or just for your own amusement, of course! Once you're done there, hitting `F` or `G` takes you to a group and fortress menu list. Fiddle around here and come up with a creative name if you want!

When you're done, (you may have to hit `Esc` to head back out of the menu), hit `Tab` to go to the item list. Many items will be on your list already, on the left, with animal options on the right.

Prepare for the Journey to Tancatten			
(copper picks [2])	44	No turkey gobbler, ♂	6
(copper battle axes [2])	68	No poult, ♂	6
(steel anvil)	300	No turkey hen, ♀	6
(dwarven rum [20])	2	No poult, ♀	6
(dwarven beer [40])	2	No horse, ♂	101
(plump helmet spawn [5])	1	No horse foal, ♂	101
(pig tail seeds [5])	1	No horse, ♀	101
(cave wheat seeds [5])	1	No horse foal, ♀	101
(sweet pod seeds [5])	1	No dog, ♀	16
(rock nuts [5])	1	No puppy, ♀	16
(dimple cup spawn [5])	1	No hunting dog, ♀	31
(prepared aardvark heart [15])	2	No war dog, ♀	31
(pond turtle, ♀ [15])	2	No dog, ♂	16
(plump helmets [15])	4	No puppy, ♂	16
(pig tail fiber thread [5])	12	No hunting dog, ♂	31
(pig tail fiber cloth [5])	14	No war dog, ♂	31
(pig tail fiber bags [5])	20	No cat, ♀	11
(pig tail fiber ropes [3])	20	No kitten, ♀	11
(goose leather quivers [3])	10	No cat, ♂	11
(ginkgo wood buckets [3])	10	No kitten, ♂	11
(ginkgo wood splints [3])	10	No mule, ♂	101
<b>Tab:</b> Dwarves <b>e:</b> Embark! <b>==:</b> Add <b>n:</b> New <b>F:</b> Name Fortress <b>G:</b> Name Group <b>s:</b> Save <b>Pts:</b> 200			

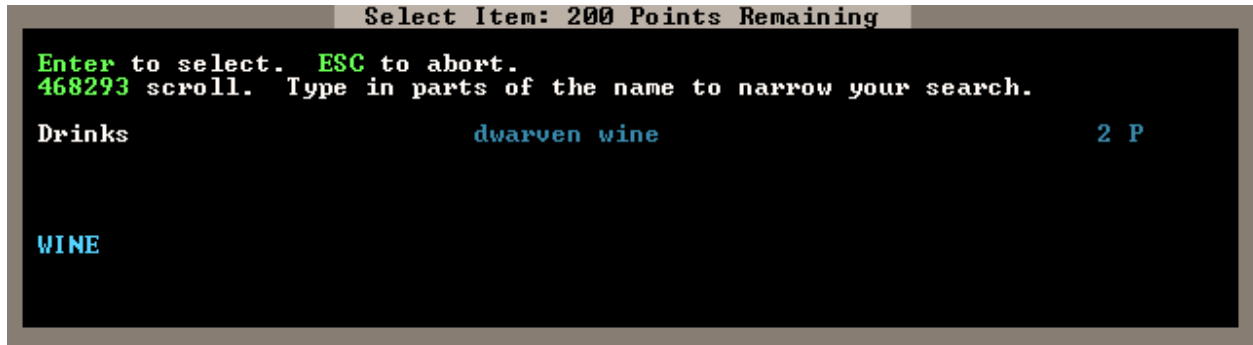
As in the last menu, use the arrows to move left and right and the up down arrows to move through the list. Head across to the animal list and hit – a couple of times to add 2 dogs to your party. Go down, add a couple of cats as well. No, make that three of each, then you've got a pretty good chance of getting one of each sex. That's handy for reproduction, so my mum tells me!

= reduces the number of items. Reduce the number of battle axes to one. I usually don't bother taking two steel battle axes as one skilled woodsman with a single battleaxe is quite enough and getting rid of one will free up some points you can use for other things. As for the rest of the items, that depends on your strategy. For now, hit n for *New*. You'll be taken to a huge menu of purchasable items.

Select Item: 200 Points Remaining		
Enter to select. ESC to abort.		
468293 scroll. Type in parts of the name to narrow your search.		
Meat	silver maces	170 P
Fish	silver spears	180 P
Eggs	silver short swords	140 P
Garden Vegetables	silver war hammers	90 P
Drinks	copper maces	34 P
Digging Implements	copper spears	36 P
Weapons	copper short swords	28 P
Training Weapons	copper war hammers	18 P
Ammo	bronze maces	85 P
Leather	bronze spears	90 P
Cloth (Silk)	bronze short swords	70 P
Cloth (Yarn)	bronze war hammers	45 P
Wood	bronze battle axes	170 P
Stone	bismuth bronze maces	102 P
Metal Bars	bismuth bronze spears	108 P
Small Cut Gems	bismuth bronze short swords	84 P

Lets add some turtles to the list of stuff we're going to take. Scroll down to *Fish*, then across to the right and down to *Turtle*. Hit Enter, you will now be taken back to the main goods menu and you'll see turtles added to your list. Hit – until you've got 20 turtles. Note, the points available to you will go down as you do this – your available points are in the bottom right corner and the point cost of each item is listed next to the item.

Lets add some Dwarven Wine the other method. Hit `n` for new, then start typing *wine*. Pretty soon you'll be left with only one option, *Dwarven Wine*. You'll note that the *Drinks* part of the menu is highlighted, you'll need scroll across *Dwarven Wine*, then hit `Enter`.



You can now increase and decrease the number in the usual fashion. Lots and lots of drink is a good idea!

### 2.1.2 Embark Strategies

You now know enough to navigate around the somewhat confusing menus. But what you don't have is an strategy for what to buy when you're embarking. Thankfully, there are some really good ideas presented on the Dwarf Fortress Wiki Starting Builds guide and of course, more ideas over on the Dwarf Fortress forums. I suggest going and reading the starting builds guide, there are some fun ideas there – for example, taking no axes, a little wood and then forging your own axes on site (saves you points for other items!).

The [DF Starter Pack](#) includes a collection of embark builds, ranging from good for new players to advanced options designed to maximise the impact of each point. You can save your settings with `s`, if you're happy with them.

Take all the advice there and then think about the location you're embarking to. If it seems likely to be a frozen waste, you may chose to modify the builds presented as many are tailored for fairly ideal locations. That being said, there are some interesting ideas for builds centered around extreme fortresses too!

When you've spent all your points and modified everything you want, hit `e` for *Embark* and enjoy your new fortress!

## 2.2 Dig Too Deep

The traditional activity of a dwarven civilisation: digging ever downwards, driven by greed for some marvelous material. But beware, for there are older and fouler things than mere goblins in the deep places of the world.

No tutorial is necessary for this one. Just go down, through the cavern layers, and keep digging...

*If you need [spoilers](#), click that link. You've been warned.*

## 2.3 Kísat Dur: the Dwarven Martial Art

This book is a translation of a manuscript believed to have been written by Thiktog Idashtathur in Kogsakozon. For an authoritative retranscription of the original runes, visit [our office in Telingvath](#).

---

**Note:** While written in the somewhat eccentric style of an in-game martial scholar, the combat advice in this tutorial really does work and can make your adventurer unstoppable.

---

### 2.3.1 Become Strong by Fighting Alligators

Before you become a master of dwarven combat techniques, you must become a master of your body.

This usually works. It probably works even if you're an elf. (I don't know.) You should fight according to this technique:

**Attack** Close Combat

**Charge-Defense** Stand Ground

**Dodge** Stand Ground

If you are more interested in dodging than using shields, use this technique instead. I don't care:

**Attack** Close Combat

**Charge-Defense** Dodge Away or Move Around

**Dodge** Stand Ground

Get a shield and a weapon, and a full suit of armor. Then find a river and some alligators (this is easy) and slap them with your weapon. You want to inflict as little damage as possible while still raising the conflict level to Lethal.

I know everything about alligators. For instance, I know you are now surrounded by them.

Enter a Creeping Swim. Lie down in the water (s). Now move in circles slowly around the alligators. They won't be able to kill you, because they're really stupid. In about five minutes you will be really good at dodging, observing, using shields, using armor and fighting.

The next thing you should do: wrestle them until you're tired. (maybe using the techniques described later in this manual) You will probably be an expert wrestler at that point.

After that, just punch them to death.

### 2.3.2 Getting Started

Prepare mentally to start a fight.

During your fight, just flip to the page that starts with whatever situation you're in right then and see what the book says you should do. If you follow all the steps you will probably win the fight.

These techniques are not especially good for fighting more than three people at once and that's a situation you should try to avoid.

### 2.3.3 Initiation

#### Against One Enemy

Do anything you want if your enemy doesn't realize you're going to fight them.

If your enemy can close the distance between yourself and them, you should probably skip to the next section.

#### Nish Vafig

Before initiating, try offering your enemy something expensive in exchange for their weapon. This only works if you have something expensive.

### Against Two or Three Enemies

You should try to disable at least one of your enemies instantly, if you can. The tricks to do this usually involve exploiting distance, which means you won't be able to execute them more than once in the same fight unless one of your enemies is very insistent about using ranged weapons only.

These techniques are still worth a shot if you don't have distance! But I'm not going to guarantee they will work. As a general rule, remember that your efforts to wrestle don't commit until you move, attack, jump, or let time advance – this enables you to do a whole lot of things at once that you might not have realized you could.

### Korumid Strike

The korumid strike will disarm, stagger, and often cripple one enemy while leaving you with his weapons and prepared to face another. You'll need to be about five steps away to employ it.

Sprint as fast as you can towards the enemy you'd most like to disarm. When in melee range, grab their weapon and shield with your weapon and shield hands respectively. (you can grab their arms if they're missing one of those – or grab only their weapon for more reliability and speed) Then jump over them as best you can do, and body-slam them. This will probably snap their arm or leg and leave them on the ground.

If you're a novice this will be easier than disarming using wrestling alone (see the Nabasuthar technique below) but you will probably have to catch an incoming strike, which won't work for you.

If you're a vampire you can drink people's blood while midair, which will increase the power of your tackle and cause it to hit everybody in the way instead of just one. Drinking blood will not affect the efficacy of any other grabs you queue up while in the air.

### Momguzumid Strike

Start a korumid strike with a longer-than-usual running start. You must run at least three dwarfmiles per hour, and you've got to jump about four steps. Instead of grabbing both weapon and shield, grab only one thing belonging to a nearby enemy before jumping. Start the jump before your first grab has actually connected and then make a second grab while in midair. It will be impossible for most enemies to evade your first grab. (the second grab won't usually connect, but will prevent the enemy from evading)

Whatever you're grabbing doesn't necessarily have to belong to the enemy you're jumping over, which makes this technique potentially useful against groups as large as three. But fights like that are always dangerous.

Sometimes you can even use this to knock your enemies into your other enemies, but I don't think this is very reliable.

### Against Four or More Enemies

I don't know. Throw rocks at them? That sounds like a good idea.

You can also try standing somewhere important and waiting for someone to try to crawl under you. Then you can stomp on their throat. That also sounds like a good idea.

If you're very fast, try charging into the center of groups, then charging isolates at the edges. Attack ranged weapon users first and exploit full-body strikes (like the previous two) as much as possible. Obviously, melee weapon users can't attack you unless you enter melee range.

### Against Enemies Who Don't Realize You're Trying To Fight Them

There's a technique for fighting while maintaining the appearance of not fighting:

## Ar Ētul

Stand pretty close to them and jump on them a bunch of times until they die. (or fall unconscious) This takes much less preparation than the other whole-body techniques, and most enemies will assume you didn't mean to lethally hurt them. (provided you don't take a significant running start)

You shouldn't rely on enemies assuming this isn't a combat action, though.

### 2.3.4 Standing Combat Against an Enemy

Your goals are to get the enemy into any of these conditions:

- off-balance (by using wrestling tricks to stop their strike)
- unarmed (by taking their weapon)
- incapacitated by pain/nerve damage (by attacking any of the four preferred zones, tearing muscle tissue, or locking a joint if you're strong)

To do this you need either to strike opportunistically at one of the four weak zones, or to defend by grappling until you can set them off-balance and lock a joint. (or quickly strike a weak zone then) The point is to inflict suffering, so they can't fight you toe-to-toe.

Often your enemy is armed. Weapons are very dangerous to you but they also open up some initial techniques to disable your enemy. It's almost always better to use grappling if they have a weapon, because they will use it very predictably.

#### Determining What Your Enemy Wants To Do

This is the standard observational trick everyone uses:

#### Rashgur Nitem Technique

If an enemy is apparently attacking you, figure out what the easiest place to strike would be. If one of their limbs looks easier to hit, that probably means that's the limb they're attacking with.

#### Disarming

You've got no good reason not to do this first. These are techniques (predominantly grappling) that will remove your enemy's weapon. You may also be able to take the enemy's armor, which will make them more vulnerable to unarmed attacks.

#### Nabasuthar Technique

If an enemy is attacking with a weapon you can probably grab the weapon and that will confuse them. This almost always works. You might even be able to get their weapon.

If they attack you with a shield (they usually do that next, if they have one) you can use wrestling tricks (the sodel ar technique) to block the attack. It's easier to employ wrestling tricks against someone's shield than against their bare hands.

### Gakit Sodel

If you grab your enemy's shield, even when they're still holding onto it, you might automatically be able to block your enemy's attacks. This means you don't need to maintain possession of it to use it.

This isn't that reliable.

### Gakit Kun

Grab your enemy by the weapon as above, but instead of using wrestling to disarm them, scratch their hand until you sever a nerve and then take the weapon after they drop it. Even if you don't hit a nerve, you'll probably loosen their grip. This works well on enemies that are stronger than you.

### Tosidcabnul

If the enemy is wearing armor (particularly plate armor) you can grab it, but dodge out of melee range before the grab connects. This will protect you from an imminent melee attack, and remove the armor from the enemy's body. (I don't know how, but it's true.)

### Other Grappling

Offensive wrestling works best on large enemies, but defensive wrestling works even if your enemy is really big. (particularly if they're trying to punch you instead of using weapons) Against a large enemy you should use wrestling to set them off balance before taking an opportunistic strike

When fighting reactively, you should use wrestling tricks to interrupt the enemy.

Your most common trick will be to block an attack midswing with a grapple:

### Sodel Ar Technique (Wrestling Tricks)

If they attack you with their left hand, you can use your right hand to catch it. If they use their right hand you can do the same with your left. (Remember – the limb easiest-to-hit is probably the limb they're attacking with!) This will probably leave them off-balance.

As a novice you will not reliably be able to do this: dodge frequently and use the Zimesh strike instead.

There's also a defensive striking technique to accomplish this, but it won't leave your enemy off-balance. (it may inflict pain, since it can damage nerves and muscle tissue) Here it is:

### Rashgur Break

If you scratch up the enemy's hand or do another really quick strike while they're about to strike, they might get confused or miss. If you get lucky it won't hurt very much. This works really well when they're punching you. Since this won't affect enemy balance or put them in preparation for a joint-lock, you will probably need to do this a lot of times before you inflict enough pain to disable the enemy.

Another disabling technique is to stand on top of them, but you need to be strong enough to throw them to the ground first. This is a lot like taking them off-balance by grappling but it lasts longer and will allow you to make a lot of quick strikes before they can do anything about it:



## Nidostegul

Throw your enemy to the ground (preferably a weak throw, if possible) and stand on top of them before they land. Don't let your enemy strike you first or else they'll stand on top of you. (They might also knock themselves down, mind.)

## Striking

The four zones likely to disadvantage an enemy are:

### The Gut

If you hit your enemies here they will be sick. They might even vomit.

### The Lungs

There are ribs in the way, but if you bruise your enemies in the lung they will probably have trouble breathing. If you bruise both lungs they will probably drop dead.

### The Neck (Throat and Spine)

If you break their spine it will paralyze your enemy. If you crush their throat your enemy will probably suffocate. If you tear their throat your enemy will probably bleed to death. If you strangle their throat for a little while you can choke your enemy unconscious. (This is a good idea!)

### The Joints

If you break joints your enemy will be in a lot of pain and probably pass out. (This isn't very hard to do if you're strong.)

Of these, the guts and throat are least likely to be protected by bone, but they might be protected by armor.

Remember these guidelines when about to deliver a strike:

- When you decide to hit something, you should hit it a lot of times, so long as you're at least bruising the muscle or the bone. It will eventually break. (this principle is called zilirakith)
- Punching is almost as good as kicking, and much faster. (this principle is called nonub) Scratching is also fast, and scratching muscles can inflict a lot of pain.
- Most enemies can be incapacitated by pain and sickness.

It isn't strictly necessary to grapple before striking. Your key technique to get good blows against the enemy without having to wrestle defensively or significantly disadvantage the enemy is the opportunistic dodge-and-strike:

## Zimesh Strike

If you dodge to your enemy's side (by moving diagonally), you can punch or scratch your enemy a lot more easily. You should probably try to use fast (not wild) attacks so that you hit them before they turn and so you aren't left off guard.

Using biting against the throat, hands, or feet is called idrazoluth and requires too much committal to be safe against an enemy on equal footing. But scratching those areas is generally safe and can inflict pain against enemies not vulnerable to defensive wrestling.

### 2.3.5 Standing Combat Against a Disadvantaged Enemy

If your enemy is off-balance or stunned, you should take advantage of high-risk striking techniques that will more permanently disable them.

If your enemy is unconscious from pain or extremely nauseated, instead flip down to the next section.

The combat advice from the previous section applies, but a few more striking attacks are viable now that weren't before. Your goal is still to permanently incapacitate the enemy, and many techniques from the previous section can still accomplish that. A brief list of those you should still consider:

- gakit kun (tearing the muscles and motor nerves)
- nidostegul (standing on the enemy)
- zinesh strike (dodging diagonally and striking: doing so will still reduce the enemy's ability to counter)

### Wrestling

#### Asas Break (Rapid Break)

If you grabbed the enemy's hand, lock the wrist. You can usually break the joint if you're strong enough. Most of the time your enemy will double over in pain. This is one of the most reliable techniques you can employ, even against armored enemies.

#### Enkos Nitem

This only works on enemies weaker than you. Grab your enemy by the head and gouge out their eyes. It's often easier to grab their neck but pinch their throat. (Grabbing their throat itself is more difficult)

### Striking

#### Stod-Suton Strike

If you can use wrestling tricks (the sodel ar technique) to interrupt your enemy, you will probably leave them off-balance. While they're off-balance you can easily quick-attack the guts or the throat. Being very fast usually works since there aren't any bones protecting those places.

#### Mezakod Break

If you've already incapacitated your enemy by stabbing the guts, (the stod-suton strike) you will probably be able to attack the lungs. Your goal is to bruise at least one of them. Breaking the ribs helps but isn't necessary: throwing a wild punch will probably work because your enemy won't be able to counterattack.

You can also punch the head, which might kill them. You probably shouldn't rely on being able to do that, because you have to get lucky.

Using the zinesh strike against the spine has a special name. While the initial dodge is a mildly defensive move, you've got to have an opening to punch again and again, because single spine strikes aren't very reliable.

### Ostlurit Strike

You have to be really strong to do this, but if you punch your enemy in the back of the neck they'll probably break their spine and die. You might need to catch them offguard but if you're actually doing this you probably don't care, because you're very strong.

This last technique involves a wrestling technique but unlike most wrestling techniques, it doesn't work as a defensive option and it requires too much committal to be safe against an enemy on equal footing. Unlike most wrestling techniques, this will work offensively on larger enemies too:

### Idrazoluth ("Dog Combat": Tearing Technique)

If you bite people you will be able to tear them up (skin and muscle) and cause blood loss without using a weapon.

The other way to do this is scratching, but biting lets you hold on afterwards. You can shake the body part to injure it more and you can often completely sever the body part if it's small, like an ear or a finger. This leads to even more blood loss and pain.

Good places to do a lot of damage with this technique are the throat, hands, and feet, because disabling the throat will kill your enemy and disabling the hands and feet will keep them from using a weapon or walking respectively. (provided you disable *both* feet)

This is effective against animals because animals are typically susceptible to muscle pain, but latching onto an animal might hurt your mobility. If that worries you, use scratching instead.

## 2.3.6 When You Have Permanently Incapacitated Your Enemy

### If You Only Have One Enemy

#### If You Have a Weapon

Follow the advice under Striking and use wild strikes to the four preferred zones to defeat your enemy.

#### If Not

Stomp on their neck (to suffocate them) or on their throat. (to bleed them to death) Punching or wildly kicking the head several times will often kill them more quickly but is less reliable. (this is often the most effective way to finish off very large enemies)

If they're wearing lots of armor, choke them instead, or take the armor off first. But armor doesn't usually protect the neck very well, so try stomping on that first.

#### If You Have More

Ignore the one you just knocked down and subdue the others.

## 2.3.7 If Your Enemy Can't Walk (or Doesn't Want To)

Many of the techniques described in 'Initiation' will now be possible again, because you can get back the distance you had at the beginning of the fight. Don't give too much time to ranged enemies, but if they're already at a distance you may be able to exploit that distance. (e.g. by disarming them via the korumid strike)

There's at least one technique where you'll have to have disabled the enemy to succeed.

### Aknunakith Strike

Walk a little ways away from your enemy (who is lying on the ground) and then sprint as fast as you can in their direction. Then, when you're in melee range, attempt to jump over your enemy. You'll fail but you'll clobber them with both feet and generally you'll disembowel them too. (I don't understand how it works, but it does.) Unfortunately this technique isn't very practical.

## 2.3.8 When You Are On The Ground

If the enemy isn't standing over you, you should stand up. If the enemy is standing over you, you have probably lost the fight and will die. But you still have one chance to dramatically turn the fight around.

### Egnul Spurn

Interrupt your enemy's attack with a body part other than your hand. Then use a throw to put them on the ground and stand on top of them. This works well when you're already on the ground. You can usually use your hold to break their joint after you're done.

## 2.4 World Generation

Welcome to the tutorial on world building. If you've played along with the walkthrough you may be itching to try out making your own world and finding a new place to build your next dwarfy citadel (perfectly laid out – as detailed on the notepaper you doodled on at work all day). Well, you're in luck, because that's what we're talking about today!

### 2.4.1 Why World Build?

World building is a subtle and arcane art in Dwarf Fortress. You can do it the easy way (default world generation), the slightly complex way (custom parameter setting) or the really hard way (world gen file manipulation). We're going to be covering just the first way as the others make my brain hurt and I'm not a masochist! If you are interested in doing things the hard way you can head over to the wiki and check out the guides to advanced world generation and pre-generated worlds.

Er, where was I, why do we want to generate our own worlds? Well, generally it is because we want to create a specific type of environment (haunted glaciers with skeletal whales FTW!) or we want to make sure we have an interesting and exciting place to build (build a volcano inside a fortress inside a volcano!). Or perhaps you just want to start a new game in a new land.

### 2.4.2 Making Worlds the Easy Way

Fire up Dwarf Fortress and from the start menu scroll down to *Create New World!* and hit `Enter`. You'll see a screen with a variety of basic options which can be selected with the arrow keys. I suggest the following settings, but by now you can decide for yourself:

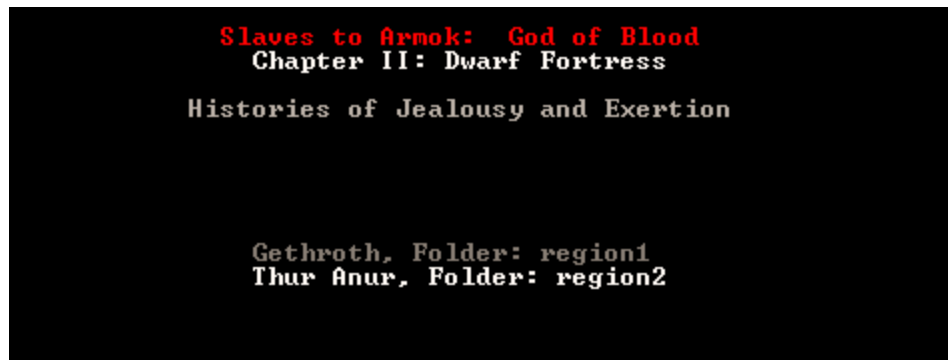


You like the settings, hit **y** and the world will start to generate. You'll get a lot of stuff happening on the screen as the world is generated. Mountains will form and be eroded, forests will spread, rivers will spawn, civilisations will rise and fall. It's pretty cool actually!

Here's a shot of the world history being generated, which happens after the geography is formed. Did you know you can actually go and read all of this history in *Legends* mode? Yes, it's all there! Tons of history and mythology to soak in. We'll have a look at that later.



Once the world is created you'll end up with a menu as well as a little information, such as the name of your world. Mine, *The Universe of Wonder* looks mountainous and pretty frozen up North! (I swear, I got that name the first time!) Use the arrow keys to look around and get a rough idea for the kind of world created. Hit **Enter** to accept the world. The world is now being stored and you will be taken back to the main menu. You can now choose *Start Playing* and a new world will be there, probably the attractively named (*untranslated name*), *Folder: Region 2*.



### 2.4.3 You're a bloody legend, mate

With your new region selected, hit “Enter” and you can see the three game modes available. *Dwarf Fortress*, which you know well by now, *Adventurer*, which you may have dipped into and *Legends*. Select *Legends* and hit Enter. After some loading you’ll get a screen showing a list of histories to read. Scroll around, hit Enter to drill down and Esc to go back up a menu layer. The first history I read is about Sidya Athetipyi, hag spouse of evil - born an elf, abducted at age four, and transformed into a creature of the night by Momuz Shadeberry. She later escaped and wandered for years, before being struck down at age 41 in the Pure Prairies. There are happy stories too, I promise!

If the in-game interface for Legends mode isn’t your cup of tea, don’t despair! From the main legends screen, you can press x, p, d to export most of the information - or use the DFHack command `exportlegends all` to do so automatically. These files are created in the *Dwarf Fortress* folder, and can be read with Legends Viewer - which incorporates the maps, and adds everything from hyperlinks to population statistics.

When you’re bored, Esc back out to the main menu.

### 2.4.4 Play the Damn Game Already!

Head back in to “Start playing” and select your region and then “Dwarf Fortress”. You’ll then get a screen somewhat like this. It is the “settle” screen where you chose where you’re going to found your fortress. I’ve scribbled on the picture below so as to give you a better idea of what you’re looking at. The image below explains the three map windows you can see:



On the right is more information and it relates entirely to what's in the embark box in the left window. Let's see what we can do about navigating around! Start off by hitting the arrow keys a while. What you'll notice is that the arrow keys move the X on the region map. Holding down an arrow key results in the local map zipping past, the region map moving square by square and the world map X moving fairly slowly. As you move the cursor around you should see the info on the right change. Note, this info does NOT relate to the region window! Instead, it relates to the local window, but as you've noticed by now we don't seem to have much control over the placement of that local window box... or do we?

If you read the text on screen you'll see you can move the local area box around using w, a, x, and d. Try it now! You'll notice lots of information changing on the right. You can also resize the local space using u, h, m, k. You probably want to leave the box at 3x3, but 4x4 will give you more space when you're playing. 5x5 should only be used by those who like the game going particularly slowly as those sort of sizes can become somewhat crippling for the CPU to keep up with. 1x1 micro-fortresses are also doable, if you want speed more than space.

Hit u and make your space 3x5. Make it small again by m. Easy huh? But rather than just selecting some random area and hitting e for *Embark!*, let's use the *Find* system to get us a nice site.

## 2.4.5 Finding somewhere to live

Follow along:

- Hit "f" for *Find*. A new menu will pop up with settings to fiddle with.
- Set *Savagery* and *Evil* to *Low*.
- Set *Flux Stone Layer* to *Yes*. Flux stone is one of the ingredients you need to make steel so we really do want it.
- Leave *Aquifer* as *N/A* - while they're a pain, you'll get a better site if you're willing to use the DFHack command `drain-aquifer` after you embark
- Set *River* to *Yes*.
- Set *Shallow Metal* and *Deep Metal* to *Multiple*.
- Set *Soil* to *< Deep*.



- Set *Clay* to *Yes*.
- Hit *Enter* for *Do Search*.

DF will now look for a location matching your preferences. Now, I've chosen the particular settings above to set up a fairly straight forward fortress location. You can have a lot of fun by not going for some of these settings, so don't be afraid to experiment.

Once the search is run a suitable site will be found (hopefully) and the local map will be centered on it. Hit *Space* to back out of the find menu. Here's what I've found.



You can see the climate looks quite good. The temperature is warm, it's sparse trees (which is plenty). It's wilderness, which will be somewhat challenging and fun and there's some sand too – always nice if you want to make glass. However, one thing I don't see is an aquifer... or do I? Perhaps it's in another biome? Biomes are how land is divided up in DF. Plains will be one biome, a mountain range another, a forest yet another. Clearly there's a grassland biome here, and there are three options next to *View Biome*; F1, F2, and F3. By default we're viewing the biome that dominates this space inside the local embark box and this is the F1 biome. If I hit F2 I will switch to seeing what's on the other biome as well as what land it covers. The flashing Xs show the boundary and I can hit F1 to see how it compares to the other biome. Here's what I see when I check the second biome. Note, it's possible to have one, two, four or even more biomes all intersecting on the area you're looking at!

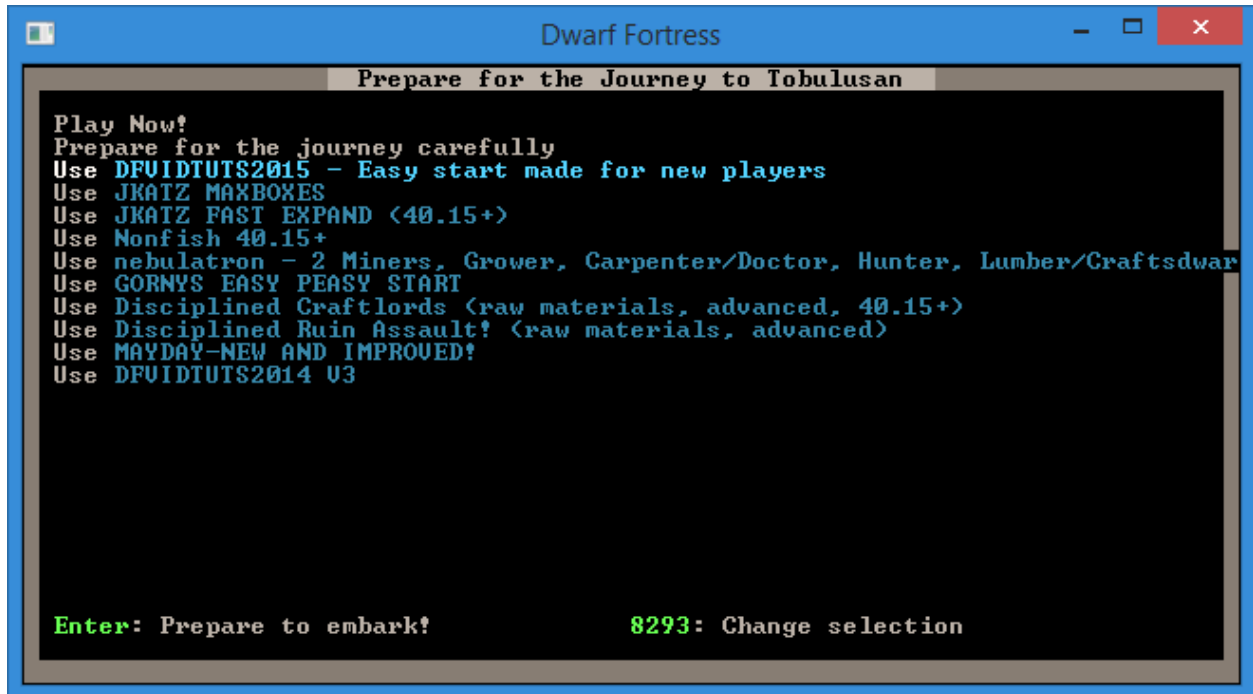


I'm pretty pleased with that! A good sized area, a water source in the corner, magma (deep down), sand, flux stone, lots of wood, perfect! Of interest is that I have a brook instead of a full river (DF tells me that in the info box to the right of the maps) which somewhat reduces the difficulty of dealing with the challenges of plumbing!

I am now ready to embark!

## 2.4.6 Your Basic Embarking

When you're comfortable with your start zone hit `e` for "Embark". You will get a confirmation dialog, hit `Enter`. You now have options that probably look something like this:



The best option, if you want to play straight away, is to select *Use DFVIDTUTS2015* (the preset the walkthrough is based on) and hit enter, you will now be taken to another menu which you want to hit *e* from. Voila! You're in and playing! Alternately, you can *Prepare for the journey carefully* and individually skill and equip each dwarf - but that's another tutorial.

Righto, you should be able to do basic world construction to your heart's content now. World generation is as complex as you want it to be and you can see some of those options under *Design New World with Advanced Parameters* from the main menu. Don't worry about those unless you get super keen on custom building worlds to ensure certain features, in which case, head off to the Bay12 forums already! There are lots of useful threads there and links to map databases (and info on how to use them). It's a hobby in itself and people seem to enjoy competing to create the most perfect starting location or the most perfect micro-fortress.

## 2.5 Managing Dwarven Jobs

**Warning:** This tutorial is based on an old version of DF, and needs to be updated.

In this tutorial we'll look at managing our dwarfs, a topic we have ignored until now. Dwarf management isn't that hard, but there's lots of subtlety to it, as I'm sure you can imagine. You also have to learn to navigate the interface, which is its own special challenge!

So lets dive right in! In the end you will find managing your dwarfs well quite rewarding, trust me!

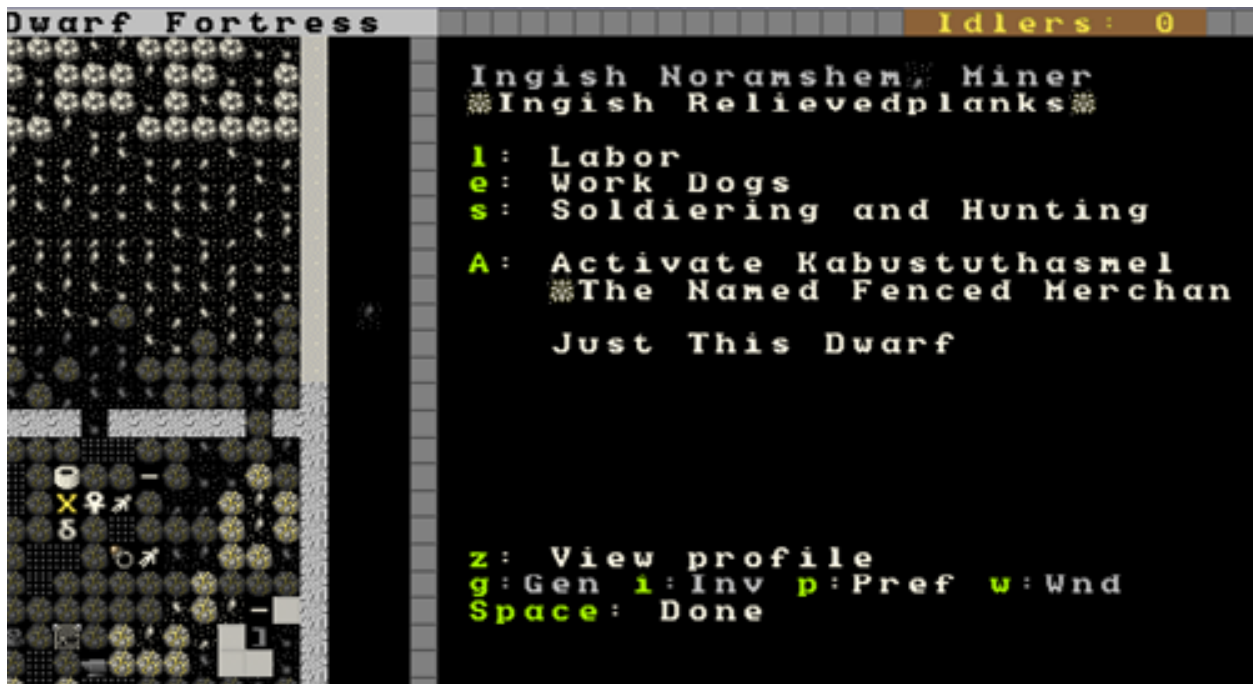
### 2.5.1 In-game interface

Hit *u*! You will be presented with a list much like this:



On the left, a list of dwarfs with their name and current profession (the game works out the dwarf's profession based on their best skill, unless you assign them a custom profession name, more on that later). On the right, we have the current job the listed dwarf is doing. You will note that the list includes more than just dwarfs. You can scroll up and down using PageUp and PageDown as well as the arrow keys so have a look around. This is the full list of all living creatures in the current area (excluding vermin), so it's a good way to see how many goblins you're facing when the inevitable siege occurs!

Had a good look? Great! Now let's get on with working with our dwarfs. With the first dwarf selected, hit **c** for *Zoom-Cre*. Using this shortcut will become extremely familiar to you as you jump to this list, select a dwarf, and whack **c** to see what on earth they are up to now!



Let's examine what we can see here. On the left we have a yellow X over the dwarf we selected. If we were to move the X around using the arrow keys we'd see the info on the right change as the closest dwarf to the X changes. By the way, you can pull up the X directly without going through **u** just by hitting **v** from the main map. This is really handy for when you're trying to work out which dwarf is skulking around the dining room not doing any work!

So, let's focus back on Ingish Noramshem, Miner. You can see some info about this stunty. First up, you can go to the **l** for "Labour" menu, the **e** for "Work Dogs", **s** for "Soldiering and Hunting" or you can **A** "activate" the dwarf.

Don't worry about any of those just yet! Instead, look at the bottom menu list. We've got z, "View profile" and a list of other shortcuts.

Things can get a little confusing from here, so follow along carefully. For some reason known only to Toady, the coder, the menus get extremely circular, but we'll get there in the end!

So, hit z, and you'll see this:



Not much to see. We can see that this dwarf owns 14 things (*snore*), has a modest quarters. The text is in blue, which I'm guessing is a good thing which means that it is over and above the quality demanded by that noble (thank Ookpik). Below, we can see we have three options, we can hit enter, view relationships with "r" or customize with "y". Lets go ahead and hit "enter". Oof! Wall of text!



You'll have to read this on your own screen to see what it says. In essence it's a blow by blow of the happy thoughts and sad thoughts of the selected dwarf as well as some bio details. As you will read from the first paragraph Ingish is quite happy, except for the vermin problem (erm, my fault, I forgot to bring any cats with us. A couple of cats will hunt vermin away). On the whole, a content dwarf, which is good. Upset dwarfs go crazy and kill other dwarfs. It's annoying.

The text in blue concerns the religious and group affiliations of this dwarf. I've never found much need to worry about this stuff, although of note is that the god the dwarf worships can be read about in Legends mode. Reading about the gods, creatures and civilisations in your world can be quite an amusing post-game sideline.

Next you'll see a list of things the dwarf likes. Over time it's quite possible your dwarf will accumulate items based on the things they like, especially once the dwarven economy kicks in. Of note is that this dwarf likes "cows for their haunting moos", how strange! Mind you, we could take advantage of this later on by putting a cage in this dwarf's quarters and shoving a cow in it. Then the dwarf will be happy ever time they see the cow, because it's a favoured animal of theirs.

Food preferences are interesting as well. This dwarf likes quarry bush leaves. We're not growing any right now (you need to grow them in an indoors field and then at a Farmer's Workshop, process them to a bag), but it might be a good idea to grow some if we ever worried about this dwarf's happiness, or just to ensure those dwarfs that do like quarry bush leaves are kept extra happy.

The next block of text covers the personality traits of the dwarf. There's not much to act on here, although it's interesting to note the dwarf's tolerance for the outside world. Over time dwarfs that spend all their time underground will become nauseous at the sight of the sun, which slows them down when they go outside. This can have a big impact on your military and so some fortress designers like to build open air spaces into their fortress. But more on that later.

Don't be alarmed by the note on drinking. All dwarfs love drink!

With this examining done, space back a menu and hit `r`.



To be honest, I seldom view this screen and I'm not sure if we really need to. I just noticed that with the selection over the deity you can hit `v` and get information on the god worshipped, cool! This dwarf worships Agesh, who "often takes the form of a female dwarf and is associated with fame, rumors, agriculture, the rain and rainbows", sounds delightful!

You also get a list of friends here, and in time, partners and children. It's interesting, but not particularly useful

Anyway, back out of that menu and hit `y` for "Customize". You get an option now, customize nickname and customise profession. Some people use these to tag their dwarfs for more easy viewing, or to name them after friends and family for general hilarity, I leave this option alone because I have plans for them using the Dwarf Manager utility, but more on that later.

Hit `Esc` and back out to the main dwarf menu. Hit `g` for "Gen". You'll see some skill information and stat information for this dwarf:





As you can see, all of this mining and digging has made our dwarf tough and agile. Ingish is also becoming a skilled miner and is also an adept mason! Other skills have been built up through time in our fortress. There's more to see than this list though, I've highlighted the down arrow which indicates that there are more skills and so use "alt"+"down arrow" and have a look. You should see "Dabbling Grower" at the bottom, looks like our shorty has been picking some plants in the fields!

Now, next menu item, hit **i** for "Inventory". You will see a list of items your dwarf is carrying, it should look something like this:





If you want to make this list easier to read, hit **Tab** a few times until the menu takes up half the screen. Then you can see all the item names in full (this tip is especially helpful with some workshop menus where item names are long). You can use this list of inventory items to check closely what your dwarf is carrying. Scroll through each item using the alt-scroll method, and hit **Enter** and **v** if you want to learn more. The inventory screen is one you'll use often when dealing with the military side of Dwarf Fortress. Don't worry too much about it for now.

Lets move on and hit **w**. This is a list of wounds:



White is good. Nothing injured. Light grey indicates bruised, yellow indicates broken, red, mangled and grey is “lopped off”. In a game I’m playing now I currently have a goblin I stuck in a special goblin pit with both eyes missing after a fight with my dwarfs (and those are the only wounds! Good shot marksdwarfs!). He keeps wandering around and passing out before waking up and wandering around again, no doubt bumping into other goblins before collapsing again. Ok, so I found it amusing, anyway...

What you can’t immediately see in this list are organs. Yes, Dwarf Fortress does model internal organs! If your dwarfs, or critters, get wounded, you can see exactly how much and to what degree from this screen. Dwarfs that get wounded will be taken (or make their own way) to your barracks where they will be fed, watered (you need a bucket made in the carpenters for your dwarfs to fetch water for the injured dwarf) and left to recover. This can take a substantial amount of time, weeks, months or even years! And they may never recover from some injuries.

Hit **p** to be taken back to the info-screen about that dwarf. For now, ignore the “Activate” and “Work Dogs” option (we have no trained dogs yet) and lets focus on “Labour”. When you hit **l** you’ll see a screen like this:



What you're looking at is a big, big list (it goes down a couple of pages) of different "labours" that the dwarf is enabled for. The jobs in white are turned on, the ones in grey, off. At the bottom of the list are all the hauling jobs, from stone hauling to cleaning.

An important job in DF is thinking about and managing the labours of your dwarfs. For example, do you want your wood cutter to also have hauling jobs on? If they have that job on they're going to spend their time cutting wood and then walking half way across the map to move a piece of food to a food store, and then walk back across the map to go back to chopping. Perhaps it would be best if you turn off the hauling jobs on your woodcutter? How about your miner? Do you have a lot of mining work to do? Turn off all the jobs except mining and your digger will drink, eat, sleep, rest and dig, and that's all. No distractions!

But hauling jobs are important, so how about turning all the extraneous labours of all those lye makers and potash maker immigrants and turning them into full-time peasant haulers. You can't go wrong with up to 1/4 of your workforce just hauling stuff and the rest divided between important jobs. At times you'll want to do a lot of bulk job modifying so you can get a lot of hauling done, or a lot of digging, or whathaveyou. This is especially the case once your fortress grows to a substantial size and you have multiple dwarfs doing about the same sort of job.

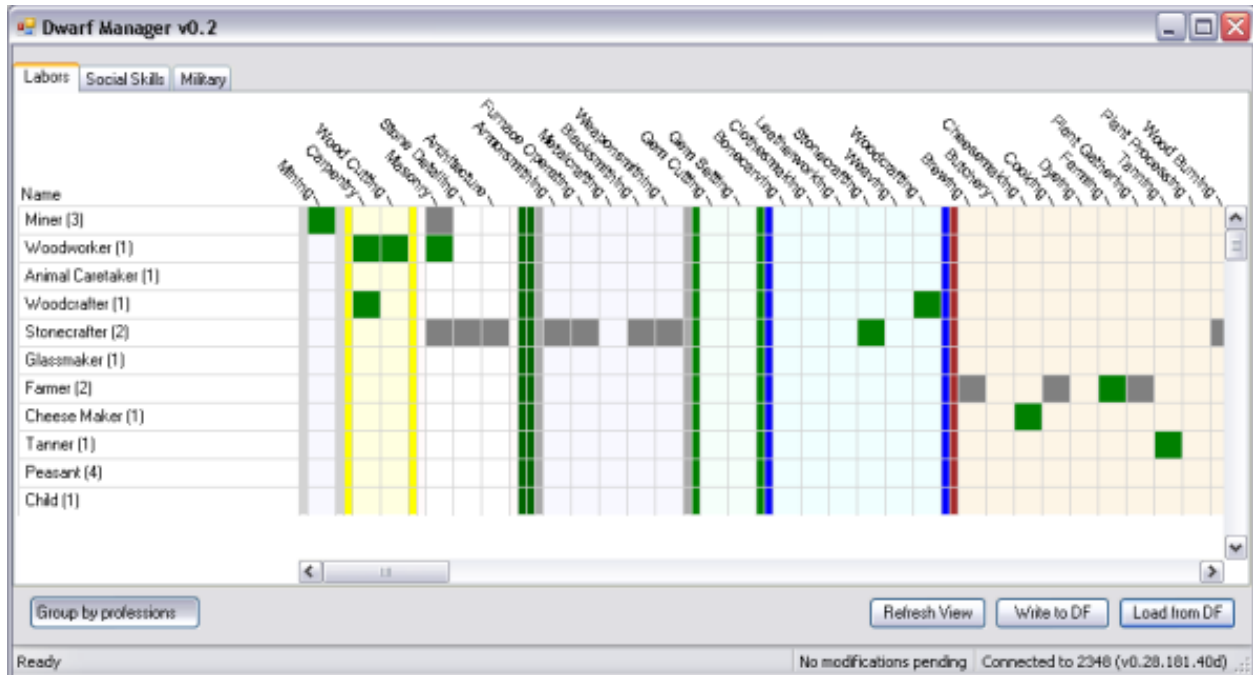
## 2.5.2 Dwarf Therapist

**Warning:** This section uses the precursor to current tools. Look up Dwarf Therapist instead, and admire the fantastic user manual it comes with.

But how does one do this without working ones way through dozens of labour settings for dozens of dwarfs? Well thankfully the answer is at your fingertips! Remember the "Dwarf Manager" shortcut we made right at the start? Good! Run it! If you skipped that step you'll find "Dwarf Manager" in your Dwarf Fortress folder. Run the exe in there.

If the program doesn't run, you may need to do some .net installing. You can check this all out at the Dwarf Manager website. \*\* NOTE: Dwarf Therapist is currently the favoured dwarf management program. You may wish to try it instead. It is fairly self-explanatory and works in much the same way. \*\*

Once you get the program up and running, the first thing you should do is hit the "Load from DF" button in the bottom right of the program. The program will pull all the dwarf data from DF and you should see a screen something like this:



The extremely cunning among you may have noticed that my Dwarf Manager is v0.2, but the one on the website is v0.6. I suggest we all go to the Dwarf Manager website, grab the latest version, and install it on top of the Dwarf Manager folder in your DF folder. Come back when you're done!

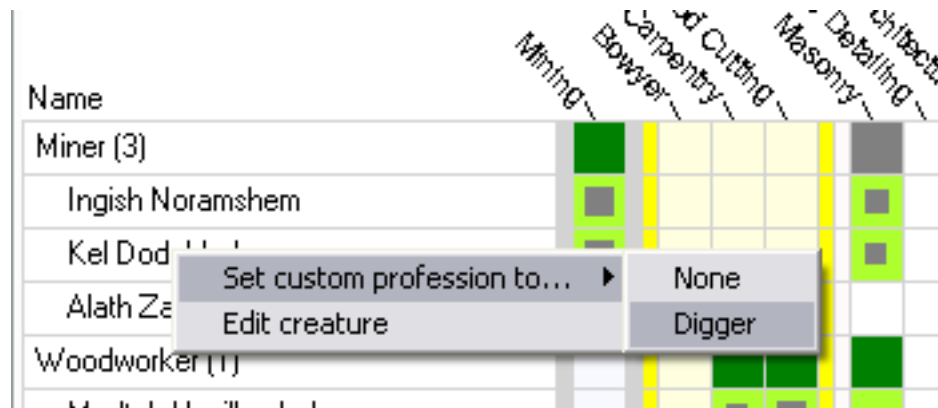
What you're looking at now is a very neat way of adjusting the labour of large numbers of dwarfs at once. It also gives you a great way of quickly seeing what your dwarfs are set to do, and it allows grouping in useful ways. Please note, in my screenshot I've had a few immigrants, so don't worry if your numbers next to professions don't match mine.

Now, go through and double click on each profession and expand them out. The green blocks show what labour the dwarf has, the grey dot in the middle shows how skilled they are at that labour – the bigger the dot, the better they are. If you double click on any dwarf name you can pull up more information about the dwarf including their name and profession and what they're skilled in.

Note that the grey bars across the profession names indicate what is turned on in that profession group. Clicking on those boxes turns on jobs for everyone in that profession. And this is where we can make things easier for ourselves. I'll show you how:

- Look at your miners in Dwarf Manager.
- Double click on the first one, change their "Profession" to "Digger".
- Hit enter.

You have now created a custom profession. Now right-click on the next dwarf, chose custom profession, click on "Digger", like the picture.



Do this for all of your miners, and then hit the bottom right button “Write to DF” (you need to do this to make any change stick, and always click “Load from DF” before making any modifications to!). With this custom profession I now have easy control over groups of dwarfs. I group dwarfs into the following groups, how you do it is up to you though:

- Diggers (miners)
- Farmers (dwarfs who farm!)
- Minions (dwarfs that only haul)
- Crafters (dwarfs with all the craft skills on)
- Brickies (dwarfs who engrave and do masonry)
- Chippies (dwarfs who cut wood and do carpentry)
- Metalers (dwarfs who smelt/armor/weaponsmith)
- Foodies (dwarfs who cook and brew)

...And so on. I'm sure you can come up with a scheme of your own. Anyway, the point is, once you get beyond a dozen dwarfs Dwarf Manager is an awesome way of herding your shorties easily and quickly. Give it a go yourself!

Managing labour is how you get things done. If you see a building or workshop not being built, hit “q” and check to see if there's a “need XYZ labour” message. If there is, check to see if anyone has the labour on. If they don't, find someone, and turn it on. Your dwarfs can do just about anything you set them to do, but remember they do it better when they are good at it, not only quicker, but with higher quality.

Keep in mind that mining and wood cutting require specialist tools (a pickaxe and a waraxe respectively). Also note that farmers with no skill may screw up and lose your seed, producing no food. While this is ok when you've got a lot of food and farmers and you're training up some new recruits, it's not so useful when you are low on food and supplies!

I think this may be enough for this lesson. I am sure I will come up with more useful tips as we go, but now you have the tools to look after your dwarfs so taking advantage of my tips will be no problem!

Next lesson we're going to get back to expanding the fortress and increasing our uberness!

## 2.6 !!Fun with Magma!!

This tutorial covers navigating the caverns, locating magma, and raising it to the surface for use. It builds on the fort from the DF Walkthrough Pack and picks up at the end of Chapter 9 of the DF Walkthrough.

Note: Unless otherwise stated, this tutorial uses the Workflow utility for building items. We will also use the Quickfort utility during the pump stack construction process. All these utilities are bundled in the latest [Starter Pack](#) maintained by PeridexisErrant.

The save I used is [DFFD file 11179](#).

### Contents

- *!!Fun with Magma!!*
  - *Delving for Magma*
  - *A Sea of Fire*
  - *A Well-Laid Foundation*
  - *Stacks on Stacks*
  - *Deus Ex Machina*
  - *Tower of Babel*

## 2.6.1 Delving for Magma

Welcome to the magma tutorial! Magma is an incredibly useful - and dangerous - substance in Dwarf Fortress. It eliminates much of the *fuel* needed for smelting and smithing, allows for mass production of steel, will burn your enemies to a crisp, and opens the possibility of creating infinite quantities of glass and clay objects. It also tends to light everything on *!!Fire!!*, melt important parts of your fortress, and is usually located inaccessibly deep underground. But if you put in the work and understand magma's unique properties, you will discover how much fun magma adds to your fortress.

First, we must locate magma on our map. Nearly all forts lack magma at the surface and in only a lucky few is it found above z-level -50. That means that we will have to dig deep to find it, through caverns full of unknown horrors. Mitigation of the risk posed by the caverns is essential, and the steps below outline important precautions to take as you delve into the depths of the world.

- Start by digging a *Down Stair* adjacent to the lowest part of our staircase.
- Place a *Floor Hatch* (b, H) over the *Down Stair*.



This small hatch will act as the final line of defense between anything that comes up from the caverns and our fortress. It will stop anything directly below it so long as we are able to *Forbid* the hatch in time.

- Assign a war dog to each of our Miners. Hit *u*, highlight the first *Miner*, and hit *z* to *Go to Unit*. Hit *p*, *e* to bring up a list of work animals and hit *Enter* to select a *Stay war dog*.
- Note: If you've pastured your war dogs in a spare room as I have (to save a few FPS), be sure to un-pasture them or else they will not join their new owners.
- It's a good idea to create a burrow at the *Hatch* in anticipation of stationing melee troops there. See DF Walkthrough Chapter 9.3 for instructions on using burrows to control troops.
- Speaking of melee troops, we have enough cannon fodder to fill the ranks of our two melee squads. Hit *m* and top off the ranks with new draftees.
- Don't forget to ensure that we have enough arms and armor to issue to the troops. A full set of armor includes a *helm*, *mail shirt*, *breastplate*, *gauntlet* (each job makes a pair), *greaves* (each job makes a pair), *high boot* (each job makes a pair), and a *shield*.
- Take a second to edit the squads' schedules to respond to combat alert states with *10 minimum* soldiers. Hit *m*, *s*, select *Prim. Kllzn. Gallery* with */\**, and hit *e*, *\**, *Shift+Enter*. Then hit *c* to *Copy orders* and *p* to paste them in for every month in each squad. Do the same thing for the other alerts, cycling between them with */\**.
- Finally, if you're up for it, create a military squad manned only by your Miners, assign as a uniform a metal helm and mail shirt, and keep their schedule on *Inactive*. This will cause the miners to don some protective gear to increase their chances of surviving unexpected encounters.

That's about all we can do to up the odds when dealing with the caverns. It's time to delve for magma. Dig a *U/D Stair* straight down from the *Hatch* as far as you can go, and see what happens.

You have discovered an expansive cavern deep underground.  
Press **Enter** for more

Welcome to the underworld.

- Make sure you keep the game paused for now. Hit *Enter* to dismiss the first notification.

You have discovered a downward passage.  
Press **Enter** to close window

These popups notify you of two distinct underground features. The first *expansive cavern deep underground* refers to this cavern layer itself. A *downward passage* is a rift that connects two or more cavern layers. The discovery of a downward passage is not good news for us because it gives things from deep below access to our fortress.

Let's assess the situation. Take a good look around the cavern using *k*. Note the underground lake with giant mushroom trees growing in it, the muddy, farmable cavern floor, and the various wiggling things moving around. Take a look at one of the wiggling *{* things.





It's an olm, whatever that is. Notice that there's no option to press `Enter` to look closer at it like you could with, for example, a dog. Furthermore, hitting `u` and scrolling over to *Others*, we see that there's no mention of any *Olm*. We can't even hit `s` and order a squad to attack it on the map. We'll find the same thing to be true for the frog-looking things in the water, called *cap hoppers*.

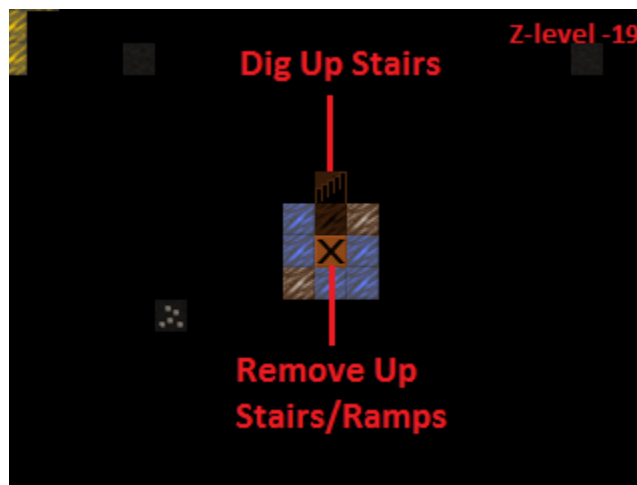
Breathe a sigh of relief; these are just harmless vermin. In fact, looking at the *Others* tab on the *Units* screen, we only see a *Tiercel Peregrine* on the surface and a *Kea* that flew into a trap last season. Currently, there are no known hostiles anywhere on the map.

Although we do not immediately see anything bent on our utter destruction, we lack line of sight on all parts of the cavern, not to mention whatever is at the bottom of the *downward passage*. Consequently we must close off access to the cavern before anything nasty shows up. Something unseen could be on its way up already, having sensed the minute change in pressure as long-closed stillness is disturbed by our miner's pick.

- Keeping the game paused, find the *Up/Down Stairway* that's been carved at the floor of the cavern.



- Go up one z-level, *Mine* one tile north, then dig an *Up stair* on the tile north of that.
- Hit *z* to *Remove Up Stairs/Ramps* on the existing *Up/Down Stair*.
- Connect our central staircase to the *Up stair*. Go up one z-level, hit *d*, *j*, and designate a *Down Stair* to be dug over the *Up stairs* below. Then, connect the *Down Stair* to your central staircase with a tunnel.



- Hit *d*, *x* and remove the dig up/down stair designation under the cavern floor. Leaving it there will cause a miner

to merrily continue burrowing downward.

- Finally, amend the Inside burrow so dwarves don't enter the caverns. Hit `w`, highlight the *Inside* burrow, hit `Enter`, `r` to change to *Currently erasing*, and erase the Inside burrow for all of z-level -20 as well as for any parts of the cavern visible on other z-levels.

The *Remove Up Stairs/Ramps* designation will leave us with only a downward stairway into the caverns. We must cover that up with a hatch and forbid it as quickly as possible to seal the cavern. Alternatively, we could build a floor over it with the closest available rock, so if you do not have a *Floor Hatch* on hand, that's the best backup plan.

- Use `.` to advance time tick by tick or unpause and repause in rapid succession. Advance time only long enough for the designations to be dug out.
- Hit `b`, `H` and place the *Floor Hatch* over the *Downward Stairway* leading to the caverns.
- Advance time only long enough for the hatch to be put in place, then immediately hit `q`, `l`, `o` to *Forbid* it and set it to *Keep tightly closed*.



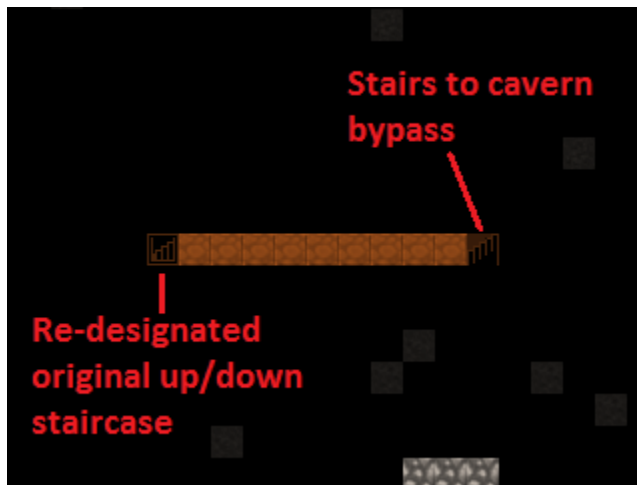
If nothing came up from below, exhale and toast the dwarf who affixed that *Floor Hatch* because no creature can get through it so long as it remains *Forbidden*.

The first cavern layer's muddy floor would make excellent underground farms and there's probably exotic things to trap down there, but for now we have to continue the magma hunt. Let's find a pillar thick enough to run a stairwell bypass through and link it back up to the main stairwell under the cavern floor. I see an appropriate one just east of our main up/down stairwell.

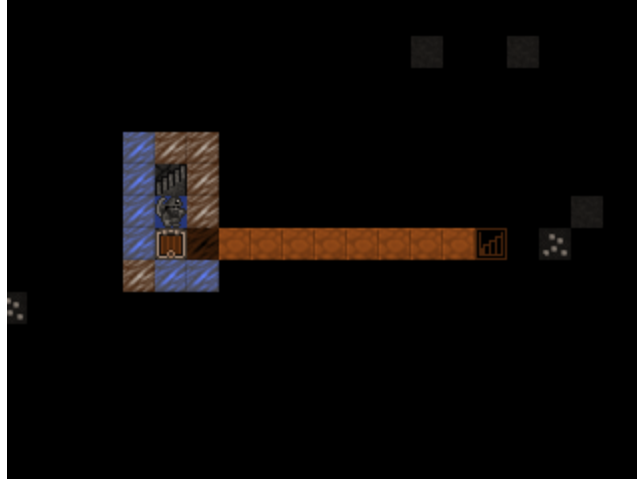
- Designate an *U/D Stair* to be dug in the pillar west of the main up/down stairwell on the cavern layer, bypassing the cavern.



- Connect the bypass stairwell to our original Up/Down stairwell designation: go down one z-level and designate another *U/D Stair*, and under that, an *Up Stair*. Connect the *Up Stair* to the original Up/Down stairwell designation with a normal tunnel.
- Complete the connection by hitting `d`, `j` and re-designating the original Up/Down stairwell to be a *Down Stair*



- Connect the bypass stairwell to the Up/Down stairwell leading to the fortress: go up one z-level from the bypass stairwell designation, hit `d`, `j`, and designate a *Down Stair*. Then *Mine* straight west to link back up with the main Up/Down stairwell.



- After the digging is complete, put a *Floor Hatch* on every *Down Stairway* you can. These will act as additional buffers if something makes it in from below.

Unpause the game and allow your miners to dig deeper. We are only at z-level -20; there's a very long way still to go.



We've hit a second cavern layer and another downward passage. Like before, keep the game paused until you set up your dig designations to secure the breach.

- Hit `u` and scroll to *Others* to see if there's any immediate threats. Thankfully, nothing is immediately visible. There are lots of *bats*, though, which are icky. Better seal this up fast.
- Repeat the steps above to seal the stairwell. Remove the up stairs above the breached stairwell, dig upward stairs to reconnect with the stairwell, throw a hatch or floor over the breach, and bypass.
- Don't forget to delete the sections of the *Inside* burrow, remove the dig designation under the cavern floor, and cover all downward staircases with a floor hatch.

This time, I elect to construct a floor instead of installing a hatch because the stone was readily available and the closest hatch was 152 tiles away.





Nothing to do but continue to dig.



Not long after the second cavern, we stumble into a third. As usual, hit `u` and scroll over to *Others* to see if some miners are about to meet a bloody end.

There's a new creature on the *Others* list, a *Jabberer*. Hit `v` to examine it.

*A huge monster that lurks in caverns deep under the earth.  
It uses its wide beak to reach down and pluck up unsuspecting intruders.*

Jabberers are terrifying. They are as big as elephants and much more aggressive, making a beeline to any exposed dwarf to rip it apart. Moreover, our military likely cannot take one down without suffering many casualties.

Hit `u` again, highlight the *Jabberer*, and hit `z` to locate it on the map.





It's not far from the breach we made in the caverns, but it's our lucky day - unlike the other breaches we made in the upper caverns that led directly onto the cavern floor, this one is in a rock pillar one z-level above the cavern floor. Unless the Jabberer can climb, it cannot path into the fortress.

- Pray to Armok that this Jabberer hasn't been training its *climb* skill and make the usual designations to seal and bypass the cavern layer, remembering to place hatches and trim the *Inside* burrow.
- Keep an eye on the monster - place your cursor over it with `v` and hit `⌘` to *follow* it. If it starts moving toward the breach, we must immediately evacuate the up-down staircase and seal the floor hatches we've placed.

## 2.6.2 A Sea of Fire

Let's push further downward and locate that elusive magma. So far we've had no luck finding magma close to the surface, so we'll continue mining down until we encounter signs of it.

- Reconnect the bypass staircase to the main up/down staircase, allowing your miners to continue downward.
- While we're down in the depths of the world, hit `H`, `F3`, `z` to create a hotkey to take us straight down here. Give it a name with `n` - in a fit of creativity, I called it *Basement*.

**Digging designation cancelled: warm stone located.**

Bingo. The miners have stopped work because a tile designated to be mined was found to be warm to the touch. That means there is magma on the other side.

- Hit `d` so you can see the flashing yellow signifying *warm stone* and find the tile where the dig job was cancelled.
- Hit `x` and erase all remaining dig designations on that level and below it.
- Designate an up/down staircase on the cancelled tile. Let's get a look at that magma.



We've found what we've been looking for - the magma sea at the bottom of the world.



Actually, we've found quite a bit more than what we've been looking for. *Adamantine* is the strongest, sharpest, and lightest metal in existence, but it requires special precautions to exploit. Adamantine is beyond the scope of this tutorial and can be very dangerous, so we'll leave it alone and focus on working with magma.

The magma sea is full of nasty critters, just like the caverns, so let's seal it off until we're prepared to deal with it on our own terms.

- Follow the same procedure as before, removing the up stairs, fixing a floor hatch or building a floor, and rerouting back to the main up/down staircase.
- If you get a warm stone cancellation, verify that you're not digging into the magma sea itself and re-designate the tile to override it.

Take some time to get acquainted with the magma sea and its inhabitants. Some parts are made of normal stone and others are *Semi-molten Rock*, which cannot be mined through. *Magma Crabs* are unpleasant crustaceans that spit fire, setting your dwarves alight. *Fire snakes* are vermin that normally keep to themselves but can cause problems if trapped - they are hotter even than magma, so when a dwarf transports a caged one to a stockpile, the cage burns away and looses the fire snake to ignite everything as it wiggles around.

We didn't risk the fortress dodging Jabberers and other forgotten horrors just to admire the view. The magma sea isn't much good to us all the way down here. To harness magma's potential, we need to raise it to the surface. We'll do so by building a 'pump stack.'

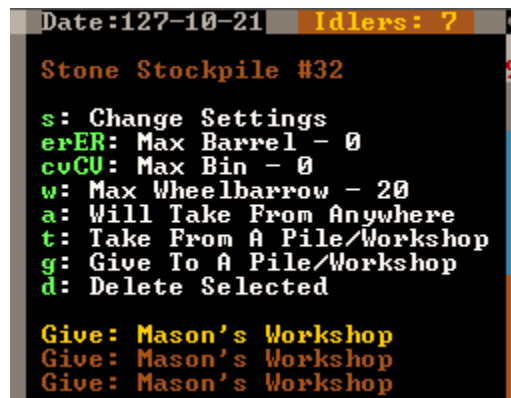
### 2.6.3 A Well-Laid Foundation

A pump stack is a series of pumps stacked directly on top of one other, each alternating in orientation so that they pump liquid from bottom to top. The pumps can be operated by dwarves, but the extreme height of ours makes it more practical to use generated power. We must also consider the temperature of magma when building the pump stack, using only magma-safe materials whose melting point is higher than the temperature of magma.

Before we start digging, it's a good idea to start building the materials needed for a pump stack. A single *Screw Pump* (b, M, s), as the game calls it, is built with one *Block*, one *Enormous Corkscrew*, and one *Pipe Section*. Each of these must be made out of a magma-safe material or else the pump will deconstruct when it touches magma. Additionally, we will need a supply of magma-safe doors to seal the pumps' output chambers.

The *Blocks* and *Doors* can be made out of some kind of magma-safe rock, and I noticed that we have lots of *quartzite* available on the upper levels of our fortress.

- Stick three or more Mason's Workshops in the quarry area at level -9. There is lots of quartzite already on the ground, but we probably will need to mine for more eventually.
- Streamline the production process by creating a quartzite-only stockpile around the Mason's Workshops. Hit p, t to create a custom stockpile that only allows *quartzite*. It's found in the *Stone* -> *Other Stone* submenu.
- Hit q over the quartzite stockpile, w, and change the maximum wheelbarrows to 20.
- Speaking of which, hit j, m, q and put in an order for 30 wooden wheelbarrows. Go to the Carpenter's Workshop and cancel some of the bucket orders to make room in the queue.
- Configure the quartzite stockpile to *give to* the three Mason's Workshops. Hit q over the quartzite stockpile, g, and Enter over each Mason's Workshop you just built. This is what you should see:



Now, the only source that these workshops will use when executing jobs is the linked quartzite stockpile. This ensures that all products made here will be magma safe.

- At the three mason's workshops, add orders for an additional 150 *doors* and an additional 150 *blocks*. You will need to edit the current workflow ranges to accomplish this.
- Make sure to delete the orders for doors and blocks at the two Masons' Workshops on z-level -5. Otherwise we may fall short of our magma-safe door and block targets.

Magma-safe *Enormous Corkscrews* and *Pipe Sections* are more difficult to produce. We have two options - forge them from iron at a Metalsmith's Forge, or make them from glass at a Glass Furnace. Although we have the infrastructure already in place to start making iron components, we shouldn't expend our finite iron ore reserves on corkscrews and pipe sections. Each corkscrew takes one bar of iron to produce and each pipe section takes three, so making 127 of each out of iron would take 508 iron bars total! Glassmaking requires only sand, which is infinite, and 127 units of charcoal for fuel. Therefore, the glass option is clearly the better course of action.

Let's beef up our glass production facilities. Yellow sand is found on z-level -2, so that's where we will locate our Glass Furnaces.

- Build a set of three Glass Furnaces to pump out *Screw Pump* components and another five to run *Collect Sand* jobs so that we have a steady supply. Here is my setup:



- Make sure to leave room for a *sand bag* stockpile near the producing Glass Furnaces. Sand bags are located at the bottom of the *Furniture/Siege Ammo* menu.
- On each of the three production Glass Furnaces, set up a production run of 150 *Make green glass tube* and *Make enormous green glass corkscrew*.
- On the five spare Glass Furnaces, order *Collect Sand* with a Workflow limit of 25-30 *powder of any sand*. It's important to keep sand bags available or glass production will cease.
- Sand collection and storage requires lots of bags, so increase the workflow limit for bags at the Clothier's Shop to 30-40.
- If the clothiers can't keep up, we'll have to resort to making leather bags and micromanaging the Farmer's Workshop to supply our clothesmaking industry. For this reason, it's also a good idea to clean out all traders of cloth and leather. Bags are critical to our magma ambitions!
- Hit *i*, *e*, fill the glass factory with a zone, and hit *s* to designate the zone as a source of sand. The dwarves will now collect sand here.

Now to expand our charcoal production capacity. Each operation at the Glass Furnaces burns a unit of fuel, and right now we have only a single Wood Furnace supplying the fort. Let's build some more to keep up with the demand for charcoal.

- Place four new Wood Furnaces in our stockpile room close to the wood. We have a large excess of space in our Armor and Ammo stockpiles, so cut into those to make room for the Wood Furnaces.
- Queue up a *Make Charcoal* job in each and change the Workflow limits to 55-60. There's currently plenty of excess wood laying around outside, but it never hurts to chop down more old growth forest. Wood stock is something we'll need to keep an eye on.



Let's do something to mitigate the extreme distances involved in hauling the pump components. We'll make a large stockpile midway to the magma sea to store our quartzite *doors*, quartzite *blocks*, glass *tubes*, and glass *enormous corkscrews*.

- Go down to z-level -60 and dig out three large rooms for our doors, tubes, and corkscrews, as well as some space for blocks which will be neatly packed into bins. We will need 150 of each, so size the rooms accordingly
- While you're down here, hit H and make a hotkey for this spot.
- I accidentally carved my way into the caverns at this point. If that happens to you, immediately wall up the breach (b, C, w) and modify the dig designation to leave at least a 1-tile wall between the caverns and fortress. Here's my setup:





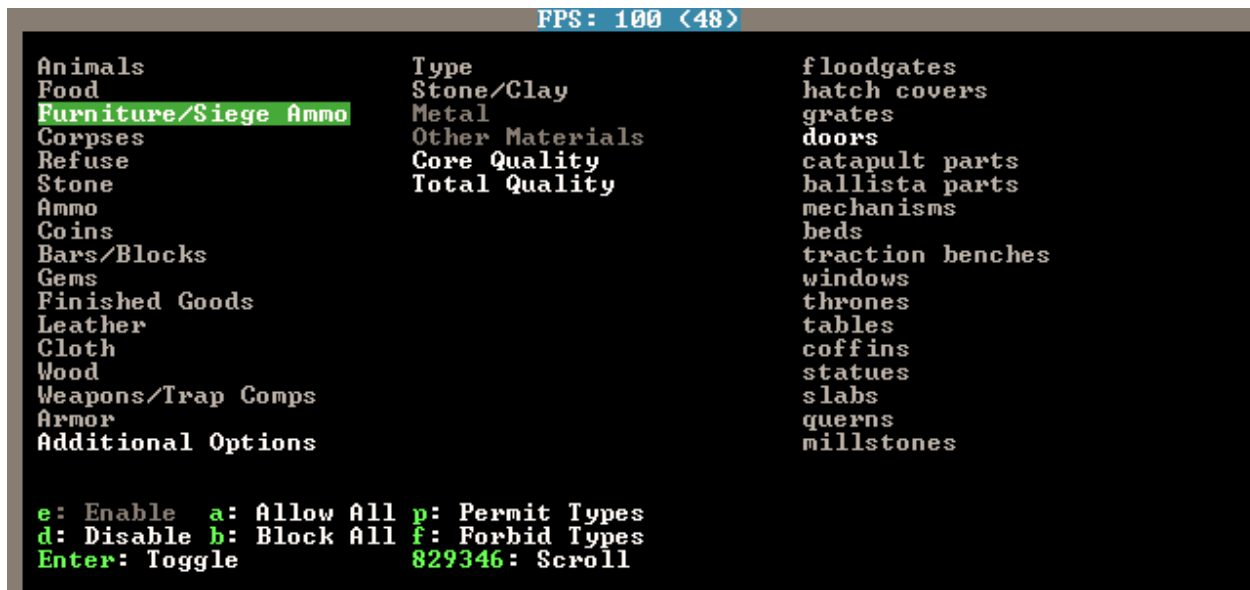
You'll end up with *marble*, *sphalerite*, and other detritus strewn everywhere. We need that space for our stockpiles, so let's clear it:

- Hit `i` and designate a 1x1 tile zone in a corner of one of the rooms, *Place* it with `Enter`.
- Hit `d` to make it a garbage dump. Hit `d`, `b`, `d` and mark everything you dug out to be dumped.

Let's set up our stockpiles, starting with the quartzite doors.

- Hit `p`, `t`, and use `d` to disable everything in the first column. Everything should be dark gray.
- Navigate to *Furniture/Siege Ammo* and hit `e`, `b`. Move right to *Type* and press `Enter` on *doors* so that it toggles to white.
- Move back to the middle column and navigate to *Stone/Clay*. Move to the right column and find *quartzite* (it helps to hit `s` and start typing "quartzite"). Hit `Enter` to toggle *quartzite* to white.
- We do not care about door quality, so navigate to the middle column, highlight *Core Quality*, and hit `p` so that all core qualities are permitted in the stockpile. Do the same for *Total Quality* directly below.
- Your screen should look like this:





- Press Escape. Now, we are designating a stockpile that will accept only *quartzite doors* of any quality. Place the stockpile in one of your 11x11 rooms.

In the same way, create stockpiles for quartzite blocks, glass corkscrews, and glass tubes. Remember to start with everything disabled when setting the parameters. If you prefer, instead of starting with *p*, *t*, you can place the stockpiles first and use *q* to edit the options. Below are sets of keystrokes certain to result in the proper stockpile in case you get lost, but it's a good idea to figure it out on your own.

- Quartzite blocks: *p*, *t*, move to and hit *d* on everything in the left column, move to *Bars/Blocks*, *e*, *b*, move to *Blocks: Stone/Clay*, move to *quartzite*, Enter, Esc, and place the stockpile.
- Glass corkscrews (located in *Trap Components*): *p*, *t*, disable everything with *d*, move to *Weapons/Trap Comps*, *e*, *b*, move to *Trap Components*, move to *Enormous Corkscrews*, Enter, move to *Other Materials*, move to *Green Glass*, Enter, move to *Core Quality*, *p*, move to *Total Quality*, *p*, Escape, and place the stockpile.
- Glass tubes (called *pipe sections* in *Furniture/Siege Ammo*): *:kbd:'p*, *t*, disable everything with *d*, move to *Furniture/Siege Ammo*, *e*, *b*, move to *Types*, move to *pipe section*, Enter, move to *Other Materials*, move to *Green Glass*, Enter, move to *Core Quality*, *p*, move to *Total Quality*, *p*, Escape, and place the stockpile.

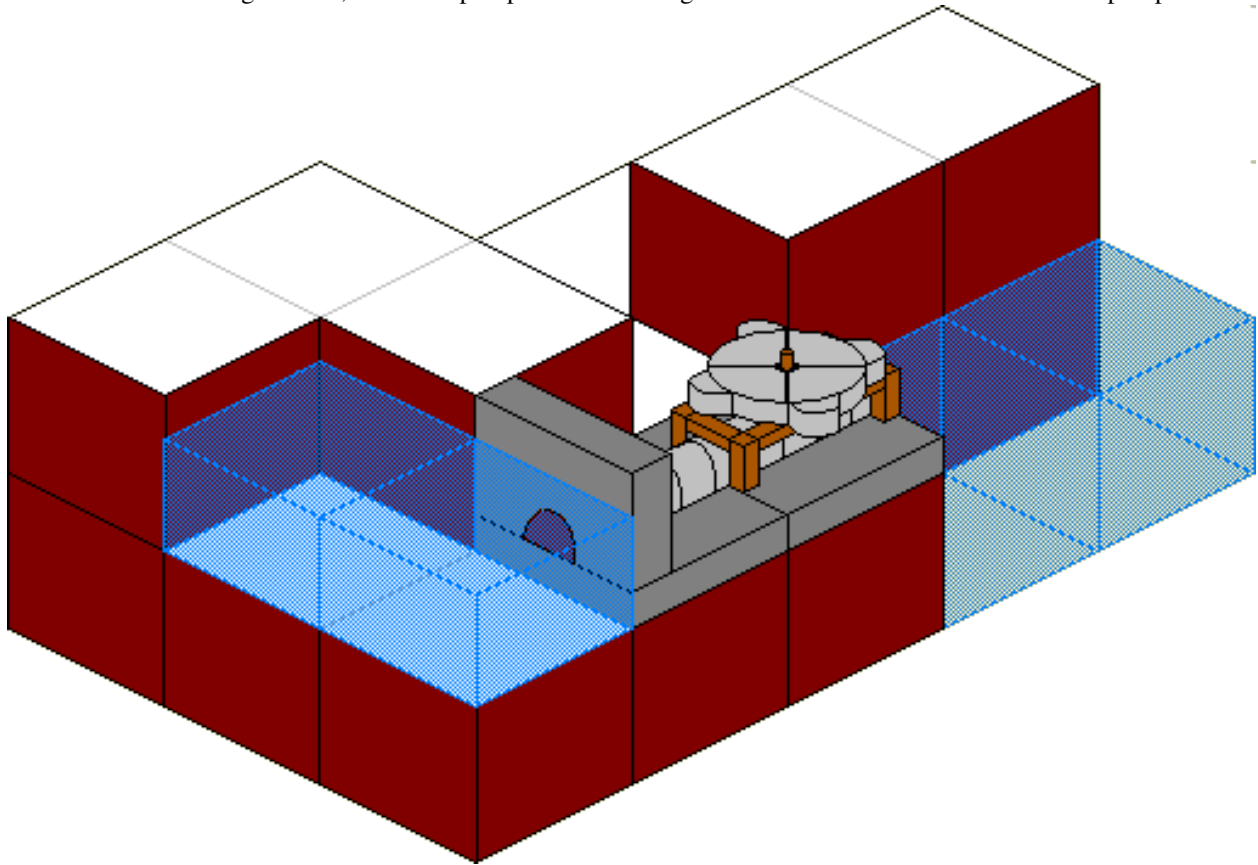
There's some housekeeping to do before we're done with stockpile configuration. We have to disable quartzite doors, quartzite blocks, glass tubes, and glass corkscrews from our other stockpiles or else these products will wind up in our existing stockpiles instead of our new ones.

- Go to our main stockpile level on z-level -4. Hit *q* over *Furniture Stockpile #24*, then *s* to bring up the settings. Disable *quartzite* in the *Stone/Clay* menu and *Green Glass* in the *Other Materials* menu.
- Hit *q* over *Weapon Stockpile #18* to the west of the stairs. Hit *s* and disable *green glass* in the *Other Materials* section.
- Go down one level and open the settings of *Furniture Stockpile #5* in the northeast. Disable *quartzite* in the *Stone/Clay* menu and *Green Glass* in the *Other Materials* menu.
- Staying on the same level, open the settings for *Bar/Block Stockpile #10* in the northwest. Disable *quartzite* in the *Blocks: Stone/Clay* menu. Now quartzite blocks will go only to the new block stockpiles on z-level -60.

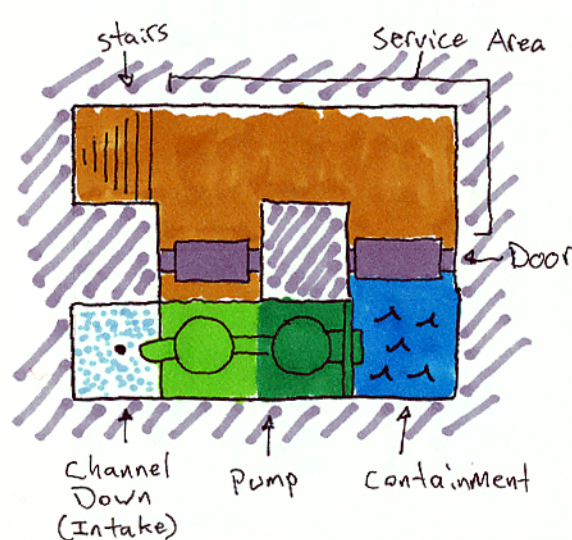
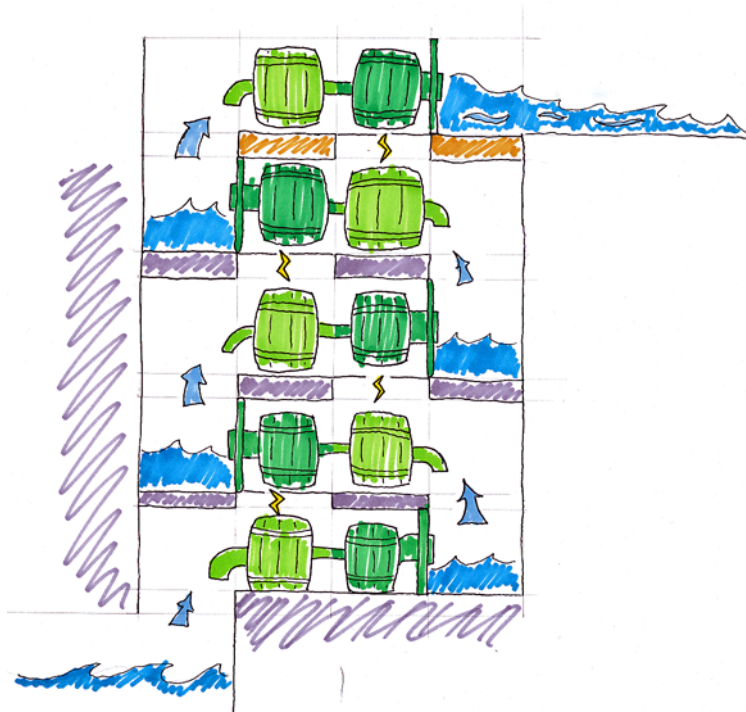
## 2.6.4 Stacks on Stacks

Let's turn our attention to digging out the pump stack casing. First, we should go over pump operation in order to understand what we're doing. A pump lifts liquids - water or magma - from the level below it up to

its level. In the image below, water is pumped from the right to the left and is retained on the pump's level.



A pump is 2x1 tiles large and operates under dwarf or generated power. One tile of the pump is impassible so as to retain the pumped liquid. The other is passable to allow a dwarf access to the pump. If we build a pump so that its passable tile rests not on a floor but atop the impassible tile of a pump below, the upper pump will transmit power downward to the lower pump. This power transmission property is the key to the pump stack's utility because it allows us to stack an infinite number on top of one another.



Orientation of pumps in a pump stack is critical. In the illustration above, notice how the bottom pump draws water from the channeled out tile adjacent to it, pumps it up to its level, and outputs the water on the other side. Consequently, the next pump up must be rotated 180 degrees in order to draw from the output water of the first pump.

Therefore, when digging out the casing for our pump stack, we have to ensure that:

- Each pump's intake tile has been channeled out to allow access to the magma beneath it,
- Each pump's passable tile has been channeled out to allow power transmission downward, and
- Dwarves can access the passable tile of the pump as well as the output tile for maintenance and installation.

It helps to visualize what we need to do. The animation below outlines the pump stack construction process.

We'll use a modified design from the one above:

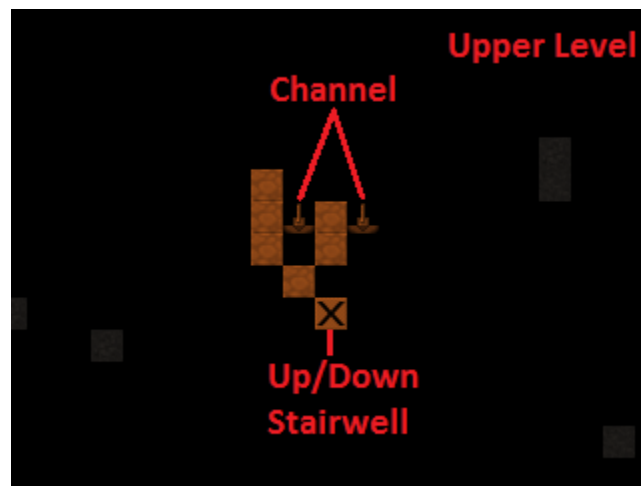
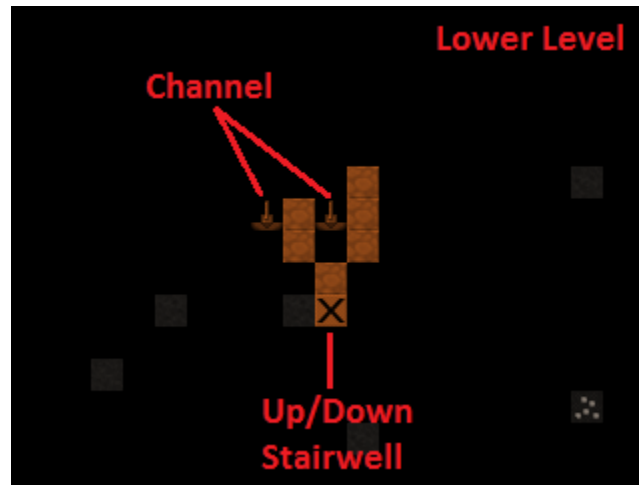
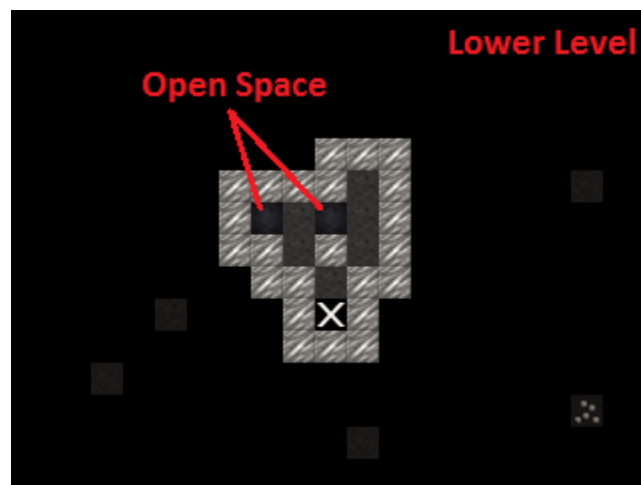


Fig. 1: Designated



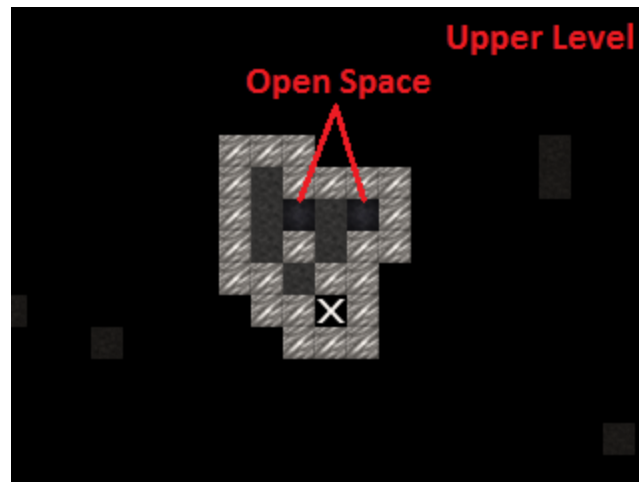


Fig. 2: Dug out

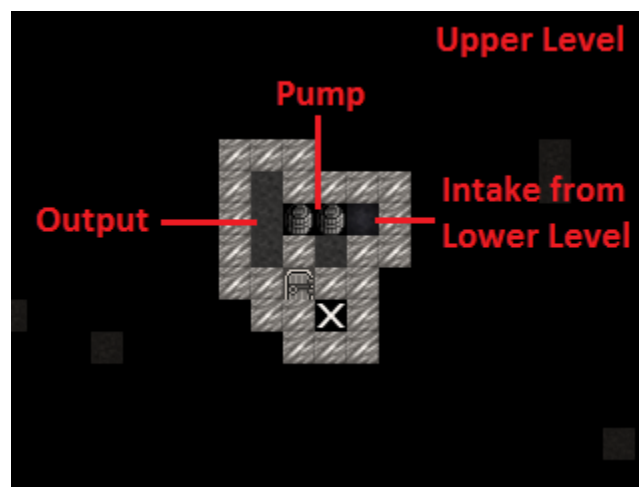
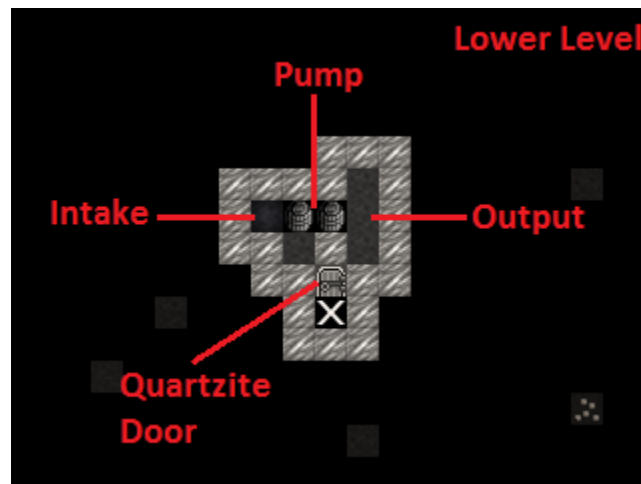


Fig. 3: Populated

Some notes about the design:

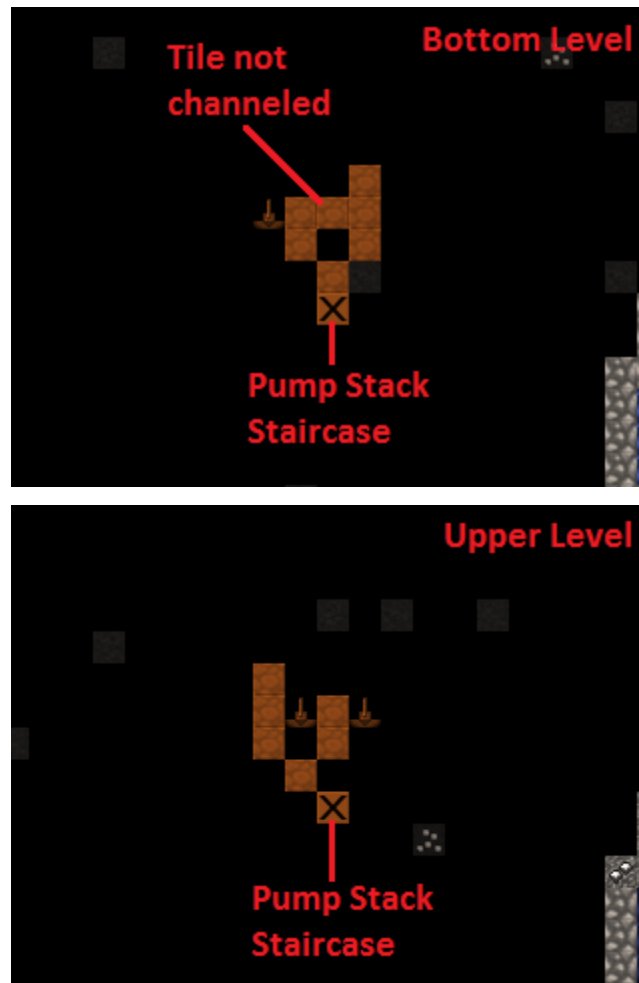
- Our design incorporates !!Science!! to mitigate framerate loss. Pump stacks are notorious for plunging the game's framerate into single digits, so we've used a 3x1 output zone for each pump instead of the previous illustrations' 1x1 zone.
- The quartzite door blocks access to the output zone of each pump. When built, each must be forbidden and made pet impassible to prevent magma leaks.
- Unlike the rest of the pumps, the bottom-most pump in the stack must be placed on solid ground or a constructed floor. All subsequent pumps are built on top of one another due to the channeled-out portion of the casing. This allows power to be transmitted from above, but the downside is that if one pump deconstructs, all will fall apart.

The first order of business is to locate a suitable vertical route for our pump stack. Including the outer walls, we need 6x7 tiles of space for our design. Let's look for a solid 6x7 column of rock that runs from magma to open air.

- Hit F3 and go down the rest of the way to the very bottom of your pump stack. Start moving up and see if any routes stand out. It's best to keep the pump stack well away from the central staircase in order to dodge the upper levels' bedrooms, quarry, and stockpile rooms.
- After ten minutes of scrolling up and down the map, I found a suitable route to the north of the central stairwell. Run an up/down stairway from z-level -1 to z-level -126 through the location marked on the screen shot below:



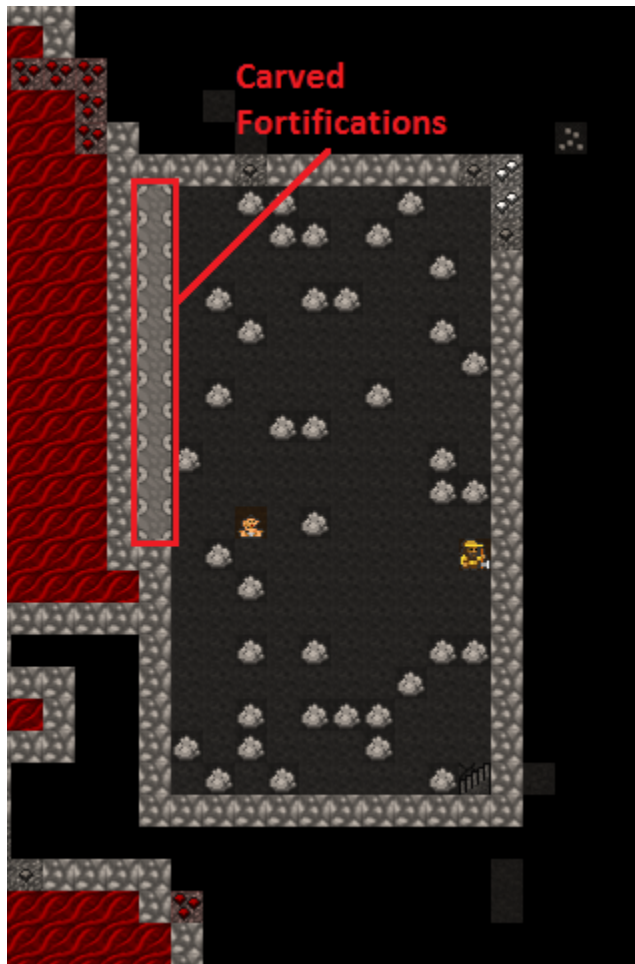
- Connect it to the stockpiles at z-level -60.
- Dig out the first two layers of the pump stack casing. Make sure to only channel out the intake tile of the bottom-most layer, not the tile under the pump itself.



Now, let's dig out the magma intake pool below the bottom-most level of the pump stack. We will use *Fortifications* to keep out the magma creatures and channel from above to safely flood the intake pool.

- Dig an up stair below the pump stack staircase and then mine out one of the flatter sections to the north, leaving a two-tile wide wall between you and the magma pool.
- When the job is completed, *Smooth Stone* on western walls with `d, s`, then *Carve Fortifications* with `d, F`. This is what it should look like completed:





- We will flood the intake pool by channeling out its walls from above. Dig an access tunnel above the *Fortifications*.
- From the access tunnel level, channel out the tiles to the west of the *Fortifications*. Make sure not to channel out tiles that are not in front of fortifications or you'll expose the fortress to the magma sea and wind up with flaming crabs in your basement.



- The level below will flood with magma. Block off the access tunnel with a constructed wall to seal it.

Now, let's dig out the last 120 z-levels of the pump stack casing.

Actually, let's not. This may be the most tedious, mind-numbing task in any video game you'll ever play. Fortunately, we have the Quickfort utility to do it all for us, and today is your lucky day - I've committed our pump stack design to a series of blueprints using the Picturefort utility so you don't have to!

- Extract Pump Stack - Quickfort - DF Magma Tutorial.xlsx to a convenient location. We will use this file as a blueprint to dig the casing and place the pumps and doors.
- Run the Quickfort utility from the DF Starter Pack window, found in the Utilities tab.
- Quickfort will now take over your `Alt` key in Dwarf Fortress. You can toggle Quickfort on/off at any time with `Shift+Alt+Z` or exit it entirely with `Shift+Alt+X`.
- Go back to the Dwarf Fortress window. Hit `d` and place your cursor on the *Up/Down Staircase* on z-level -124, directly above the first two levels of pump stack casing.
- Hit `Alt+F` and open Pump Stack - Quickfort - DF Magma Tutorial.xlsx from the convenient location you extracted it to. Then select Pump Stack - Dig from the list on the left and click OK.
- Take a moment to read the Quickfort cursor tooltip, and then hit `Alt+D` to execute the macro.

Quickfort's magic designates the dig squares for us automatically! Take a moment to appreciate how the old-timer Urists had to designate all this by hand. 'You kids have it too easy,' they'd tell ya, 'makes ya soft and complacent!' But we digress - let's finish digging out the pump stack casing.

Our macro has three steps - first it digs the 'Lower'-type pump stack, then it moves up a level and digs the 'Upper'-type pump stack, and finally moves up another level to be ready for the next operation. Our job is to ensure that we start each operation on a fresh, undesignated level. If we get the dig designations out of order, the pump stack will not work.

- We've designated two levels to be dug out so far, so place the cursor on the *Up/Down Stair* above the top level designated: z-level -122.
- Hit `d` to activate *Mine* mode, and hit `Alt+D`. Quickfort will designate two levels to be dug out and spit the cursor out on the fresh layer above the top level designated.
- Ensure that the cursor is resting on the *Up/Down Stair* where Quickfort spit it out (just don't move it). Hit `Alt+D` again to execute another two levels.
- Continue hitting `Alt+D` in this manner until we end up on z-level -1. Then, remove the dig designation created by Quickfort on z-level -1. We will save this level for power-related mechanical linkages.
- Finally, go back through each level and ensure that we haven't made any mistakes. Check the dig designations against the patterns above. When you're satisfied, unpause and let the miners get to work.
- If you get spammed with *Urist McPickaxe, Miner, cancels Dig: Inappropriate dig square*, just ignore it. Our design does not require babying to be dug out correctly.
- Finish the process by hitting `d`, `b`, `d`, and designating everything in the doorframes of the pump stack to be dumped. Loose stone in this area combined with the diagonal angles of our design can cause dwarves to suspend door construction.

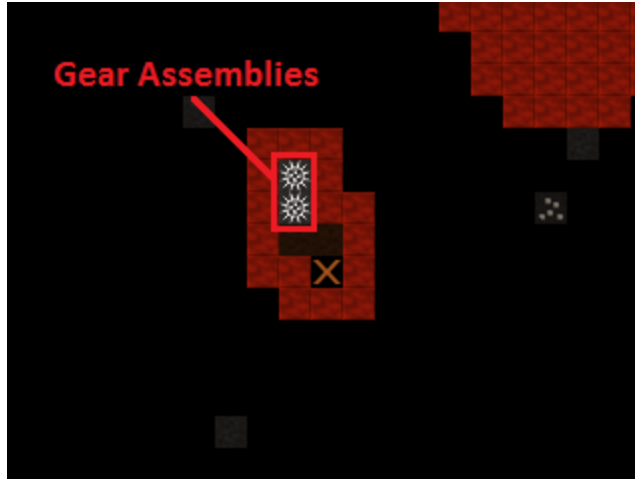
Placement of the pumps can be the most time-consuming part of building the pump stack. Each pump requires an Architect to 'design' it and then a Mechanic to put it together. To mitigate this, let's install a *Gear Assembly* above the top of the pump stack. This will anchor it from above, allowing pumps to be built from above and below simultaneously, thereby halving the time required to complete the stack.

An added benefit of two anchor points is that if one pump deconstructs due to, for example, accidentally being built from non-magma-safe materials, the entire stack shouldn't also deconstruct for lack of a foundation.

- On z-level -1, channel out the tile leading to where the impassible tile of the pump below will be.



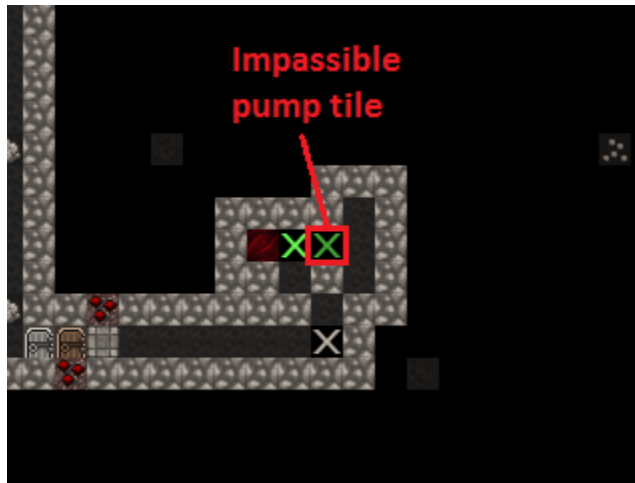
- Hit `b`, `M`, `g` and build two *Gear Assemblies* - one on solid ground adjacent to the channel and the second hanging over the channel. The hanging gear assembly is anchored by the one on solid ground next to it and therefore provides a foundation for pumps built below it.



It's time to build the pumps and place the doors. We'll do the first two levels manually to illustrate important considerations that need be taken and then use Quickfort to finish the stack.

- Go to the bottom of the pump stack, z-level -122.
- Hit `d`, `M`, `s` to bring up the *Screw Pump* build panel.

Recall that pumps have a passible tile and an impassible tile. The pumped liquid is output on the far side of the impassible tile. Therefore, each pump must be oriented so that the impassible tile forms a seal on the 3x1 output chamber. Use `umkh` to orient the pump so that the dark green impassible tile is adjacent to the 3x1 output chamber. It should look like this:



- Hit `Enter`.
- Make absolutely sure you choose only magma-safe components made from magma-safe materials! Otherwise, the whole stack will fall apart the instant it's activated. Use only *quartzite blocks*, *enormous green glass corkscrews*, and *green glass tubes*.
- Finish it off by placing a *quartzite door* in the tile between the *Up/Down Staircase* and the 3x1 output chamber. When we turn on the stack, we will make sure these doors are locked tight.
- Go to the next level and build another setup in the same manner, but with the pump turned 180 degrees so that the impassible tile seals the 3x1 output chamber.

Keep going if you want - it helps to do a few levels manually to really understand why a pump stack works. When you're ready, let's finish building with Quickfort.

First, delete the Workflow orders for glass tubes, corkscrews, quartzite blocks, and quartzite doors. We don't want to keep producing these as they are used to construct the pump stack, and the number of items in the game impacts framerate.

Quickfort does not yet permit manual material selection - whatever is first on the list is used when build orders are put in, and the materials are sorted only by distance. That opens the possibility that we could unknowingly use a non-magma-safe door or block in our pump stack. Therefore, we need to forbid all non-quartzite doors and blocks.

- Hit `z` and select the *Stocks* panel. Go down to *doors*, move to the right section of the screen, and press `f` on all non-quartzite doors listed.
- Do the same for blocks.
- We can check our success by starting to build a pump/door and verifying that quartzite is the only option for either.
- Build the pump stack using Quickfort. Ensure that you start on a 'Lower'-type level with the 3x1 output chamber to the right.
- Hit `Alt+F`, load Pump Stack - Quickfort - DF Magma Tutorial.xlsx, and select Pump Stack - Build from the menu on the left.
- Read the tooltip instructions, hit `b`, `o`, place the cursor on the pump stack's *Up/Down Staircase*, and hit `Alt+D`.
- When the Quickfort macro spits us out on the next level after executing the build commands, hit `Alt+D` again, and repeat until all pumps and doors are placed.
- Verify that the pumps and doors are placed correctly, then unpause and let the dwarves get to work.

### 2.6.5 Deus Ex Machina

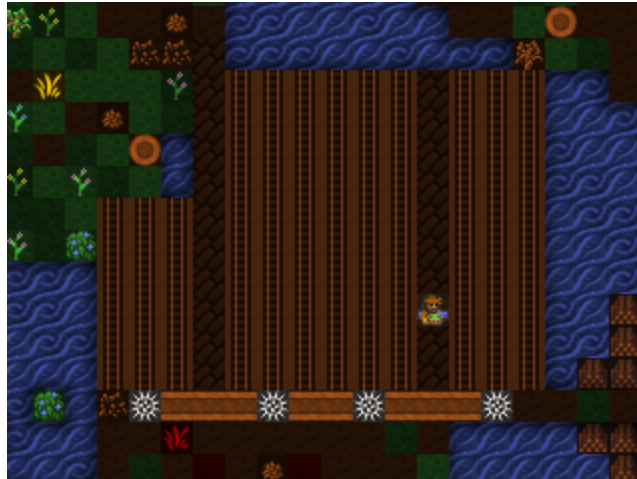
Now that everything is built, we must devise a way to power the pump stacks. We have 124 pumps to drive, each using 10 power, for a total of 1240 required power. Additionally, the mechanical linkages necessary to transmit all that power push our power generation needs well above 1500.

We'll use *Water Wheels* for power generation. Unsurprisingly, these require flowing water. The only natural flowing water on the map is the river in the northwest, which is exposed to invaders. Another option is to create an artificial underground river that empties into the caverns, but these only work well with low-flow sources like aquifers (that's right - aquifers are useful!). Unfortunately, the high flow rate of our river will overwhelm the game's water flow model in an artificial channel, resulting in no power.

We will install about 30 *Water Wheels* (`b`, `M`, `w`) along the river. This almost certainly is overkill - 17 would probably be fine, but more power never hurts and we have tons of wood.

- A water wheel requires a stable foundation adjacent to it, so first build a *Gear Assembly* on the river bank.
- Each water wheel can support additional ones constructed adjacent to it, so build more water wheels across the river. Construct floors as needed to reach these.
- Make three or four sets of water wheels.
- Connect the gear assemblies with *Horizontal Axles* (`b`, `M`, `h`).

Here is what I came up with. It generates more than 3500 power, plenty for our purposes!



Let's transmit all this power to our pump stack by means of *Horizontal Axles*, *Vertical Axles*, and *Gear Assemblies*. Doing so means creating a permanent opening in our fortress, so we'll have to build walls to deter invaders.

- Find the two gear assemblies at the top of the pump stack. Dig a tunnel ten or so tiles north, then dig an *Upward Ramp* (d, r).
- Remove the ramp when it's dug out. In its place, build a *Gear Assembly*.

We'll now have a hole in our fortress that we cannot plug - the space will be occupied by a *Vertical Axle*. Let's build a small tower around it to mitigate the danger. It's a great idea to add traps as well. Here's a design I came up with (the access stairs will be removed after axle and gear installation):





Let's build a series of axles and gear assemblies to transmit power to the pump stack. Use *Gear Assemblies* for junctions, *Horizontal Axles* to connect power on the same z-level, and *Vertical Axles* to connect between the z-levels. Always build from the bottom up, as well.

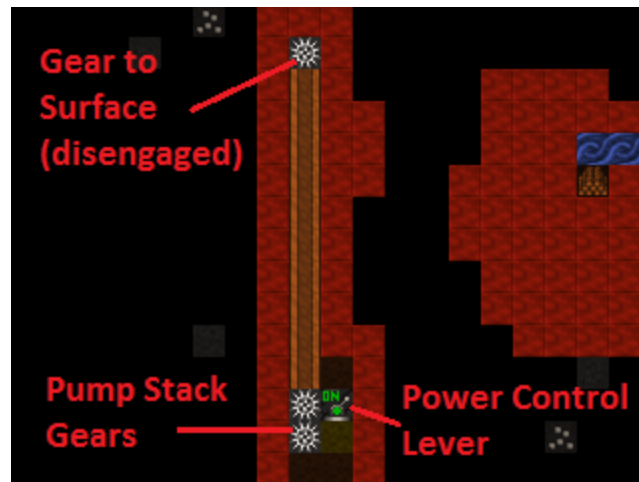
- Start by building a *Vertical Axle* over the channeled-out over the underground gear assembly in our tower.
- One z-level above that, build a *Gear Assembly*.
- Remember: build from bottom up - determine the path of the axle leading out over the tower, then build a *Gear Assembly* on the ground outside it to transmit the power over the walls.
- Go up a z-level and build a *Gear Assembly* on top of that one.
- Connect the two upper gear assemblies with a *Horizontal Axle*.
- Connect the *Gear Assembly* on the ground outside the tower to the water wheel power plant. Cut down any trees that stand in your way.
- When the machinery is completed, remove (d, n) the access stairs on the tower.

We must also devise a way to turn off power to the pump stack. Let's connect a *Lever* to the *Gear Assembly* below the tower. Pulling it will disengage the gear assembly and break the 'chain' of power to the pumps. Just be sure not to hook the lever to the supporting gear assembly to the south of the pump stack gear - disconnecting that one may cause the pump stack to collapse for sudden lack of a foundation.

- After connecting the lever to the gear assembly, *Pull the Lever* to disengage the gear. We don't want to pump magma before we decide where it will go!
- Complete the power linkage with a *Horizontal Axle* leading from the disengaged gear assembly to the gear assembly atop the pump stack.

Here is an overview of the setup I devised:







With power connected, the only thing we have left to do is dig tunnels to direct the pumped magma and throw the master power lever.

### 2.6.6 Tower of Babel

Magma has a wide range of applications, but for now, I've dug out a massive magma-powered factory that will handle all smelting, forging, glassmaking, and kiln operations of the fortress. The specific layout of your magma infrastructure is completely up to you! Perhaps you'd rather build a magma weapon?





Before activating our pump stack, it's a good idea to *Forbid* and *Keep Tightly Closed* the doors on each level of the stack. Otherwise, a cat or child will open one and run flaming throughout the fortress, igniting everything and horrifying everyone.

With that done, it's time to turn it all on!

- Verify that no one is in the magma tubes. Pumps work fast and leave little time to get away. Use `d`, `o`, `r` to restrict access if necessary.
- Throw the master power lever and cross your fingers!



The magma flows! It's working! And our frames per second is in the teens! Turn off the pumps to fix that, by the way.

Surface magma is the key to true mass production of steel, glass, and clay products. We can also weaponize it to burn our enemies into *Piles of Ash* (see [:wiki:' magma mist <Magma\\_mist>'](#)), combine it with water to create *obsidian*, or make an awesome magma moat. Let's start by relocating our smelting, forging, glassmaking, and kilns to the magma factory floor and setting up steel production.

- Hit `b`, `e` to find the *Magma Smelter*, *Magma Glass Furnace*, and *Magma Kiln*. The *Magma Forge* is built with `b`, `w`, `v`.
- Create more magma-safe materials for these workshops if you run short.
- Each magma-powered workshop requires a channeled-out tile leading to the magma below. For maximum safety, plan out your channeling so that the workshops' impassible tiles cover the exposed magma.
- Delete the old smelters and forges - they are obsolete now.

Below is an example layout for steel production using magma-powered workshops. I connected the various stockpiles to the appropriate workshops, but that's not strictly necessary and may be more trouble than it's worth. I also dug an additional magma tunnel to accommodate a lower level dedicated to magma forges. The new magma tunnel doubles as a trap, too!







This is just the beginning of magma's possibilities! I decided to start converting my outdoor garden to a greenhouse with the addition of a roof made from *Clear Glass* blocks. No more pesky goblins digging in my potatoes!





My steel production is coming along so well that I've had to clear cut the entire map of trees, too. What fun! Once my squads are outfitted, I plan on taking control of the caverns and maybe mining the Adamantine pillar we found in the magma sea.

Good luck with your own magma-fueled adventures and congratulations on completing this tutorial! If you have a chance, I hope you post your volcanic contraptions and resulting Fun on the Bay 12 forums. Praise Armok and strike the earth!

## 2.7 Trading for Fun and Profit

This will be a bit shorter, but we're going to cover some important information, none-the-less. So stay tuned for more dwarfy goodness!

### 2.7.1 Blankets, beads, muskets and cat skull totems!

Trading is very important for your fortress. Every few months you're likely to get a trade caravan and envoy arrive at your fortress looking to trade with you. They offer a wide range of goods and one of the most common things we trade back to them is crafts made from the crafts dwarf's workshop. And the easiest crafts to trade are rock ones. Rock mugs and instruments are popular as well, so get a bunch made of all of the above when you've got the time.

But for the traders to visit properly they need a trade depot. This is a big structure which requires a 3-wide corridor for the trader wagons to access. Often people build them close to the fortress entrance, but there's no reason you couldn't build a trade depot deep inside your fortress to keep it nice and safe.

For this tutorial, we'll just set up the depot outside. You can build it wherever you want, so long as it's accessible to wagons!

- Hit **b**, **D** for *Trade Depot* (remember that keys are case-sensitive, so that's **Shift-d**)

Now that we have the depot up we can expect to see traders turn up to unload their goods here now and then. To trade with them you **q** over the building and follow the options: **g** to move goods to the trade depot, and **r** to request your boss dwarf to go do the trading. Once he turns up you hit **t** and then enter the trade screens. Note again, do not trade anything wooden to the elves, including wooden barrels and bins! They will get pissy, leave, and attack you some years later! Trading is beyond the scope of this current tutorial as there's quite a lot to it, but you can read a lot more about it on the Dwarf Fortress Wiki. I suggest you make an effort to trade as trading encourages new immigrants to come to your fortress and provides you with items you can't make or find yourself. Speaking of which...

## 2.7.2 Trading for fun and profit

At some point you're going to get some traders turn up and with your depot up, they'll soon make themselves at home in your depot. They'll unload all their goods and stand around waiting for something to happen. This is how it will look when traders have arrived and unloaded:

Look at them, filling our trade depot with their mess! Lets get trading so they'll bugger off and we can make the place all tidy again. Yes, I like things tidy in my fortress, it helps my fortress run like a well-oiled machine! So how do we gouge the visitors, I mean, trade with our friends?

Essentially, trading is a five step process:

1. Set goods to be moved to the trade depot for us to trade with.
2. Call the trader dwarf to the depot.
3. Engage in trade.
4. Deal with the trade liaison and his questions.
5. Wave goodbye to the traders and enjoy your loot!

So lets get through these stages! First up, move goods. Go `q` over the trade depot, and with traders in there, the options will be quite different to what you see when it's empty. What we're looking for is hitting `g` for *Move goods to/from the depot*. You'll get this screen:

```

All           Finished Goods Bin <alder> < Distance: 20 [PENDING]
Plants        Finished Goods Bin <hazel wo Distance: 21 [PENDING]
Seeds
Drinks
Weapons
Cloth <Silk>
Cloth <Yarn>
Wood
Stone
Sand
Stone Blocks
Powders
Extracts
Crafts
Barrels
Buckets
Bags <Leather>

s: Select _      Shift+Enter: Move visible goods
Sorting by distance  p: Pending on top      m: Culling on mandates
Enter: Move good   v: View good      468293: Scroll  ESC: Done

```

This is a list of everything in our fortress. On the left we have item categories. In the middle, we've got the items themselves, and on the right, the distance from the trade depot. Looking at the distance measure you now know why some people prefer to put the depot all the way inside – remembering the 3-wide corridor or ramp all the way to the depot, of course.

Don't worry about all of this detail for now, just scroll down to *Crafts* using `↓`. Once you're there, this is what you'll see:

```

Trade Goods
All           Limestone 20
Plants        Limestone 20
Seeds         Limestone 20
Drinks        Limestone 20
Weapons       Limestone 20
Cloth <Silk>  Limestone 20
Cloth <Yarn>  Limestone 20
Wood          Limestone 20
Stone         Limestone 20
Sand          Limestone 20
Stone Blocks  Limestone 20
Powders       Limestone 20
Extracts      Limestone 20
Crafts        Limestone 20
Barrels       Limestone 20
Buckets       Limestone 20
Bags <Leather> Limestone 20

s: Select _      p: Pending on top      m: Culling on mandates
Sorting by distance  v: View good      468293: Scroll  ESC: Done
Enter: Move good

```

What do you see? All of the crafts we've been making listed on the right! You'll also notice weird symbols to the left and right of each named item, these are the quality symbols. Later on you may want to manage the quality of goods your dwarfs use, especially weapons, so knowing that `==` much good item, is worth remembering.

Now we want to move these goods to the trade depot so we can trade them away. I mean really, how many *Sandstone earrings* do we really need? So, hit the `→` to move to the item list (rather than the category list) and then hit `Enter`. On the right you'll see *Pending* next to the item we've selected. Scroll down and set most of the items to *Pending*.

At the bottom of the list you'll see some *Finished Goods* bins. This is good! Moving a single bin full of stuff is much better than moving dozens of items one by one! How do I know there's heaps of stuff in those bins? I hit `v` with a bin highlighted!



You don't need to select those individual items to move, selecting the bin in the menu above will move the bin and everything it contains.

Right! Once we `Esc` back out our dwarves will start moving the trade goods we ordered. From here the trade depot is still selected so let's take the chance to hit `x` which cycles the trader status. It now says *Trader requested at depot*, which is what we're aiming for. The trader should now consider heading up to talk to the traders.

With the game resumed goods are tagged to be moved and pretty soon dwarfs will start hauling goods to your trade depot. The hauling of trade goods seems to be set at quite a high priority, so it shouldn't take long for them all to be offloaded. Of course, it would be much quicker if I had more bins, that way a dwarf isn't wasted carrying a single earring, and we'll have lots of wasted time with this hauling! More bins are on my to-do list.

After a minute, hit `q` again and move over the trade depot, you'll see the status of the depot. If *t: Trade* is grey, the trader hasn't got there yet. Below the menu field it tells me my trader is *On break*. oh no, I lie, now he's asleep! Hurry up dude, I've got crap to offload to the visitors!

While you're waiting, why not go and set a bunch of barrels and bins and booze to be made. You can never, ever have too much of any of that stuff!

Oh, look! The trader finally bothered to turn up to trade! See, it says *Trade at Depot* for his current job, and the *Trade* option is white and therefore, usable.

```

g: Move Goods to/from Depot
t: Trade

r: Trader requested at depot
b: Only broker may trade

Broker:
Kivish Cubordodok, broker
Trade at Depot
Broker can access depot

```

Now hit `t` and we'll be taken to the trade screen, which looks like this:



On the left, we have what the traders are offering. On the right, we have our stuff. At this point you can start selecting stuff to trade away. Left and right arrows move between the panels and up and down through the list. Remember, don't trade bins or wooden items to elves! Not sure if they're elves? Look at the top of the screen and you'll see who you're trading with, in my case I'm trading with traders from "Riseilime". If I `Esc` back out of the trading to the main menu and hit `c` I will see a list of in this world civilisations. I see that Riseilime is an elf civilisation. I can hit `Enter` and learn more about it and then `Tab` through more info. Interesting.

In case you're wondering you can actually read all of the history of each of those civilisations in `legends` mode. There's some crazy stuff that the game auto-generates and the detail is remarkable (like stories about dwarf children, kidnapped by goblins, who go native and in adulthood lead goblin raiding parties against dwarf cities!). But don't worry about that for now, let's get back to trading.

Go through your list of items, and being careful not to select the wooden bins, hit `enter` on all those crafts.



As you can see, I've got lots of stuff up for trade (marked with a `T`). On the left bottom there it tells me how much my junk is worth, 4980! That's a lot of loot I can get off the elves! On the right, allowed weight is 14. This number starts much higher and drops as you add goods and represents the amount of weight the traders can take away with them.

So now we've set a huge pile of junk for us to sell, let's pick a bunch of junk off the elves! I suggest buying food,

booze, animals in cages, weapons (but not the very best ones) like steel crossbows, chainmail (steel and iron), shields, etc. But don't stress, buy what amuses you! Oh, definitely buy a few cloth or leather bags (what seeds get stored in) and some rope (for setting up animals as guards). Type of rope or bags doesn't really matter. So much shopping to do!

One important point! Your dwarfs can't wear anything *Narrow* or *Large*, so don't buy it.

Once you've chosen all that's useful you should have a look at the bottom left, *Trader Profit*. If it's more than a token tip you will want to un-select some of your own trade goods. No point in giving away lots of junk for free!

When you're happy with your trading, hit `t`, the trade will either go through, or you'll get a counter offer. `Enter` lets you consider the counter offer, which you can adjust down more in your favour. Hit `t` again and hopefully the trade completes. Keep working at it until it pans out. Until your trader is experienced your tip could be quite large.

If the trade completes you will see the message *[name] seems pleased with the trading*. Job done! Hit `Esc` to go back to the local map and let the game resume. You don't need to do anything else now, dwarfs will come and haul your spoils and the traders will leave happy with their trinkets. Even better, they will take news of your awesome limestone mugs, earrings and bracelets back to the world and that will attract even more immigrants. Super!

The next likely thing to happen is that the trade liaison will want to talk to you about what goods you want. Feel free to suggest to the trader items you'd like (using the arrow keys to scroll and upgrade the priority of an item). Note, the more you want the item, the higher the price they will charge when they bring it, so mild suggestions for preferred items are probably better than setting the meter all the way over to the right.

The trader will also tell you what they're interested in getting in trade goods. The higher the priority the more they will pay for the good. I largely ignore this as, frankly, if you churn out a ton of good crafts, and trade regularly, you should have plenty of junk to trade for most things you want, let alone junk from dead invading goblins etc. If/when you do get those screens, read the screen prompts carefully, and experiment! It can't hurt!

That's pretty much all there is to it!



## CHAPTER 3

---

### Masterclass

---

A masterclass is a tutorial on an advanced topic, aimed at experienced players. You'll be taken through some of the most difficult challenges DF offers, and see how an expert does things.

### 3.1 Training Archers

Source: [Mechanixm's Archery Training Primer](#)

I've had several requests to create a guide as to how I set up the Archery Training Room of Highsteppes. I'll go through the core steps to train crossbow dwarves. I assume you have some DF experience, so I won't tell you literally everything.

#### 3.1.1 Building the training room

Dig out a simple 11x11 room and seal it off with doors. There is nothing special about this... it's just a room. Then build the targets (b, A) in a vertical line, one space from the walls.

The material of the blocks used has no effect on anything. Dwarves with Masonry enabled will come and build the targets.

Dig a channel next to the walls, to save any bolts that miss a target.

This way, if a bolt flies and hits the north or south wall, it will be saved - bolts that are fired from one z-level and land on another are not destroyed when they hit the ground. Reclaiming ammunition like this is more useful for combat (ie metal) bolts, so this is just to demonstrate the technique.

q, r each target and set its room size. Yes, you have to do this for each of them.



If you're not planning to shoot from left to right, press `w/a/s/d` to change the firing direction.

### 3.1.2 Creating the squad

In my example, I haven't created any squads yet. I assign them *No Uniforms* because I literally have no armor in this fortress. Rename your squad if you wish. Assign all of your Crossbow dwarves to the squad.

If you have uniforms created that contain crossbows, you should be able to just assign your guys that uniform. Otherwise, manually assign a crossbow to each dwarf as the example shows.

Highlight the squad name and press `c` to add a new item. Add Wood Bolts and Bone Bolts and set the amounts to 600 of each. I've never had an issues with ammo when setting 100 bolts per dwarf. The bolts are set for both Combat and Training.

When you graduate your dwarves out of a Training Squad to a Legit Crossbow squad, only use metal ammo. Crappy materials are for training.

You'll need some other supplies - waterskins, quivers, and backpacks.

Quivers are mandatory - marksdwarves won't train if they can't carry ammo! Waterskins or flasks are optional, but your military dwarves need them to carry booze and thus avoid drink breaks. Backpacks are used to carry food with them, so they don't need a dining room to eat.

Make some crossbows and wooden bolts. I hardly ever ever ever use wooden crossbows, but use whatever you want.

I have 6 dwarves in this Training Squad, so I'm going to create minimum training orders. I then delete that unnecessary Train 10 minimum order by pressing `Tab`, `x`.

Press `Tab` to get back up to the schedule grid where it says *Train for each Month*. On the month I created the 6 Train 1 Soldier minimum orders, I press `c` to copy, then `p` paste those orders for each month of the year. *Sleep in room* and *uniformed inactive* stay at default settings.

Finally, `s` for squads, `a` to choose my squad, `t` to set active.

### 3.1.3 Assign your squad to the Archery Training Room.

Yes. You need to do this for the room set from every target, individually. Press `q` and highlight an Archery Target Training Room, highlight your squad and press `t`. My squad is already highlighted because I only have one squad right now. Set your squad to train for each target... and let the training begin!

If you've done everything correctly, the dwarves will come in and shoot. Once they are done, they will go on break. It is not unusual for your dwarves to shoot an entire quiver while training and then just sort of hang around for a month.

Instead, you can give the squad a Kill command to go kill some helpless animal. After they kill it, they should go right back to the training to train some more. Be patient, and your dwarves will continue to train. They just like to take breaks after emptying each quiver.

You'll need to create an ammunition stockpile in the room. DO NOT try to store ammunition in bins; it's very buggy. Use a quantum stockpile or learn to live with it.

Reclaim ammo from the channel with d, b, c and select the items to recover.

Your dwarves will grab the bolts and bring it over to your ammo stockpile. In my case, it gets quantum stockpiled which allows for easy grabbing when refilling quivers. Here is a gif showing ammo reclamation during training. Notice how no one gets hit in the crossfire. That is normal:

Slowly but surely their skills will rise.

Thanks for taking the time to read. A complete list of my other guides can be [found here](#). If you have any questions or comments, please let me know. **-Mech**

## 3.2 Use of Minecarts

This masterclass on Minecarts, containing significant original research, was written by Larix. [See the original here](#), or check out [the wiki on minecarts](#).

The wiki article contains a lot of varied information, but i've delved quite a bit into minecart pathing; i.e. where a minecart goes when you let it run free, how its paths change and so on. There's almost no in-depth information on all this stuff on the wiki and it seems to me that much of it isn't well understood.

I encourage all readers to replicate my designs and experiments and offer corrections or alternative interpretations. In order to properly trace what's going on, you will need to look at events closely and that means (unless you have an infallible hack script to do it for you) you'll need to pause the game, advance by single steps and count the steps exactly. You can't just eyeball the speed as "pretty fast" or "sort of sluggish", you'll have to e.g. count out a hundred steps and look how far a cart travels in that time, so you can definitely tell whether a cart moves at 45000 or 55000 speed.

### Contents

- *Use of Minecarts*
  - *Lesson One: Track on flat floor*
  - *Lesson Two: Ramps, basics*
  - *Lesson Three: rollers and guided carts*
  - *Lesson Four: Flight*
  - *Lesson Five: diagonal movement and how to fix it*
  - *Lesson Six: False ramps*
  - *Lesson Seven: Pathing across levels*
  - *Lesson Eight: Meet the checkpoint bug*
  - *Lesson Nine: Practical implications of the checkpoint bug*
  - *Lesson Ten: Corner ramps*

Let's start simple.

### 3.2.1 Lesson One: Track on flat floor

In short, the only type of track that matters to a free-running cart are corners. All other types are irrelevant.

A sweeping statement, sure, but I found it to be perfectly true. The basic rule is that a minecart will move in a straight line. The only exception is when it encounters a track corner (the two-connection type, not T-junctions) that's connected to the direction the cart is coming from. Let's take an example:

a    b

Cart gets pushed east from a/b, moves east until the corner and turns south/north there.

a    b

Cart gets pushed and *behaves exactly the same as above*.

The cart incidentally also behaves like that when the route before the corner is entirely non-tracked floor, it'll path just the same, it will only slow down more thanks to higher friction.

Why is this so? As far as I can tell, the game doesn't calculate any sort of "heading" for the cart, it just keeps track of the velocity, probably split between x- and y- axis. When the cart moves over flat floor, all that'll happen on "direction-neutral" track is deceleration. When the cart is re-pathed by a legal corner, the whole input speed is taken and turned into speed in the exit direction of the corner.

"Straight" tracks have no pathing power over minecarts, they don't keep them "on track", because carts don't consider themselves "on track" in the first place. They're in contact with the floor and react to corners/constructions, or they're in flight and don't. That's it.

### 3.2.2 Lesson Two: Ramps, basics

Of course, everyone who works with carts for a while will probably get to love ramps. They allow carts to climb levels, they can provide speed, they even allow perpetual motion.

First of all, what makes a ramp tick?

A ramp is only fully traversible for carts and can only provide acceleration if it's properly connected by track. There are two arguments that get checked, and they concern track connections and nothing else. It doesn't matter where the cart comes from, whether it changes level or not, it's all about how the ramp is built.

- Requirement one: the ramp must have track connection to wall. One, two or three connections are all acceptable.
- Requirement two: the ramp must have exactly one track connection to a non-wall tile. This can be an adjacent ramp on the same level, flat floor or a hole (e.g. containing a down ramp).

Examples of functional track ramps:

#		#		#		#		#		#
			+		#+		##			
+									+	

All shown track engraved on up ramp.

up ramp

down ramp (i.e. hole containing a ramp)

+ flat floor

Examples of non-functional track ramps:

```
###   ###   ##
++    ++    #
```

- first example: no connection to wall
- second: more than one connection to floor
- third: no connection to floor

When a ramp is properly connected, it provides acceleration towards its “down” direction; ~5000 speed units for every step a cart moves across it. Ten steps of acceleration give as much speed as a highest-speed roller, but you’ll need multiple ramps for that.

The important part is that the game only checks if the ramp is properly connected, it doesn’t check where the cart’s coming from. This is the foundation of the fabled impulse ramp - a cart entering this ramp:

```
###
++
```

from the west will be accelerated towards the east, the same as a cart going down a level down such a ramp:

```
#+
```

(cart’s coming from track or a ramp on the level above). Impulse ramps thus grant speed without needing to sacrifice height, no more. They do not provide more or a different acceleration, just the exact same amount (which is quite a lot considering it means perpetual motion at practically any speed up to ~250.000 that you desire).

If a cart moves onto a ramp from the ramp’s down direction, it’ll be accelerated in the direction it was coming from, i.e. it decelerates (at the normal ramp rate). Excepting a rather powerful bug that’ll come up later, this deceleration will stop carts that are moving at the speed of a medium-speed roller or less before they reach a ramp’s top, whereupon they’ll roll back down from the place they’ve reached. The resulting speed when leaving the ramp again will be less than the speed the cart entered with, a cart bouncing between two ramps separated by one tile of level floor.

```
#####   #####   #####
##       ##       ##
ramps           track variants
```

will after about a dozen bounces stop on the flat middle tile. There’s no observable difference between the two ramp layouts.

I built a fifteen-level straight ramp slope to measure the speeds different numbers of ramps will give to a cart. While the speeds found were just as expected and only of minor practical use (as reference to “generate” carts of specific speeds), the experiment provided a few valuable pointers, stuff that has been worked out by others before but doesn’t seem to be widely known:

- a dropped cart (I always dropped it off a hatch) will land in the middle of the tile below and only roll down half the ramp it lands on.
- the speed rises with the number of turns the cart spends on the ramps (just under 5000 speed for every turn, ~130 000 for a cart sent down a fifteen-level ramp), but one turn is subtracted from the count; i.e. the cart is charged ~5000 speed for leaving the ramps in the end.
- the “length” of a ramp is bigger than that of a flat tile. Since I only had full steps to calculate with, the numbers aren’t super-precise, but it appears to be  $\sqrt{2}$  times the length of a flat tile (the “lost” acceleration step mentioned above actually is needed for the length calculation to best fit the results).

Out of curiosity, I checked “catching” a cart falling down a vertical shaft with a track ramp since i’ve seen reports that paying attention to the exact level and different designs were required. I drilled down a 40-z shaft and built a ramp at the bottom of it (ordinary EW ramp with wall to the west and floor to the east). The cart fell down, landed half-way up the ramp and rolled off at the usual half-ramp speed of ~20.000. I tried it at different adjacent levels, and the result

was always the same: none of the vertical speed was preserved (over 1 zlevel per step), the cart never failed to be accelerated. If you got different results, i'd like to hear how you got them. I dropped the cart via hatch, which afaik is the easiest way to guarantee a clean drop without colliding with the shaft's walls.

Enough for now, more to follow.

As a sum of lesson two, i'd offer:

Ramps' main parameter is the direction they accelerate to. The exact track engraved on them and where a cart enters the ramp is secondary. If there's a corner engraved on the ramp and the cart actually moves following that corner, the corner is respected (and things get weird), otherwise exact track is just as irrelevant as on level floor.

### 3.2.3 Lesson Three: rollers and guided carts

Rollers are the powered means of providing speed to a cart. As has been widely observed

- for practical purposes, it's easiest to assume that rollers simply *set* the cart's speed to a fixed value. *see below, "Late P.S." - I found that rollers really provide acceleration, it's just a very large amount and the acceleration gets capped at the roller's set speed. The effect rarely materialises, only when working with high-speed carts and only when speed and attitude are just "right".*
- rollers will not slow down a cart moving faster than the roller's set speed; they will "brake" and turn around a cart moving in the opposite direction
- rollers working laterally to a cart's current movement direction result in *diagonal* movement
- rollers only affect free-running carts (not guided carts) and only when they are on top of track. Rollers on ordinary floor are ignored.
- rollers which are not powered are completely ignored, you just get the effects of the tile underneath.

The "braking" power of rollers is impressive: each step spent on an opposing roller slows down a cart by 100 000 speed units. The strongest track stops slow a cart by 50.000 per step. You need a very speedy cart to keep moving past a single-tile roller. A cart moving more than one tile per step will, however, not be affected by the friction of tiles it "skips" over during its turns; only the tiles it's in at the end of a step do count. Two rollers (of any speed) in the correct spots are enough to stop and turn around a cart moving at maximum ramp speed (~270.000, that's eight tiles in three steps on average).

Since non-corner track doesn't matter, it also doesn't matter much what kind of track you build a roller on. A west-pushing roller on an E-W track tile works the same way as a west-pushing roller on N-S track, an E "track end" or a NSE t-junction.

Corners do affect the way rollers work, however:

a) a roller pushing "from" a connection of a track corner results in movement towards the corner's exit direction, *not* towards the roller's push direction. b) a roller on a track corner pushing from a direction the corner does *not* connect to pushes to its normal direction (of necessity one of the track corner's connections) and may cause diagonal movement.

a):



Roller pushing east

No matter where the cart comes from, it exits to the south as long as the roller is powered (and the cart isn't super-fast). The most evidently useful application is with a cart coming from the east, because that allows a simple powered switch: when the roller is off, the cart moves off to the west, if the roller is on, the cart goes south.

Still, it's interesting to see that carts coming from the north or south are not thrown into a diagonal, although the roller's nominal push direction is lateral to their movement. It looks like the corner sort of turns the roller's effect around.

b):



Roller pushing west

Results: when the cart comes from the east or west, it moves west. Carts coming from the north go on a diagonal heading southwest. Carts coming from the south go on a diagonal heading northwest.

Since the roller's "push from" direction is not in line with the corner, it keeps its "push to" direction *and causes the cart coming from the south to ignore the corner track*. The latter isn't some kind of cumulative speed derailing, it happens when combining two low-speed effects which even theoretically can't add up to more than 40.000 (50.000+ is the derail threshold). The only rationale I can find for it is that the roller indeed "overrules" the corner when active.

More likely, however, the laterally-working roller just "adds" its movement speed to the cart's velocity and leaves it to the corner to sort things out. The observation remains valid that a roller pushing "into" a corner is less likely to cause wild diagonal movement even when working laterally, while a roller working towards a corner's exit often causes trouble. On the whole, however, cart motion is most predictable and controllable when working with rollers in line or opposed to cart direction, not with lateral rollers. Corners appear to apply at the end of a turn, after all speed changes on the tile are done with, so in the example above, the "bend the cart to the south" effect of the corner happens after the last "set speed towards east" effect of the roller and the leaving cart goes off with southward-only speed and no eastward component.

**A cart encountering a laterally-working roller which does not sit on a corner will generally be thrown off onto a diagonal trajectory.** Diagonally-moving carts are great fun, because (Lesson One) only corners matter, so the carts'll merrily barrel all across your carefully laid-out track, smacking into walls and stopping or going places you don't want them (most cases of unexplicably-stopping carts are due to diagonal movement and wall collisions). Unless you manage to thread them through track corners, that is, because a cart properly taking a corner will move precisely in its exit direction and will not retain any diagonal movement component. More on that later.

Late PS: rather confusing results of a recent roller-based device show that rollers indeed accelerate carts, at 100.000 subtiles/step<sup>2</sup> in their given direction, *but capped at roller's set speed*. When a cart moves from roller to roller, this won't matter: since the highest speed that can be imparted by a roller is 50.000, the 100.000 acceleration is enough to neutralise the speed of a cart incoming at max. roller speed *and* impart max speed, all in a single step. However, if the cart moves at higher speeds, one step of acceleration may only change it from, say, -70.000 to +30.000 and when the cart leaves the roller's tile on the following turn, it will move at the received 30.000 speed, even if it was affected by a highest-speed roller with a set speed of 50.000. In addition, the cart actually calculates the distances it moves on the roller's tile, so the right combination of cart speed and "offset" can result in very irregular speeds. A rather bare-bones test allowed achieving different non-max speeds from a highest-speed roller by slightly varying input speed - from 12.000 to 22.000, both from a "highest" roller.

I'll make **guided carts** short, because there's not much about them: guided carts ignore special track buildings like rollers or track stops, the pushing dwarf just moves them at their walking speed, much like a wheelbarrow. Track must be "connected" for dwarfs to actually guide a cart. If they find no connection, they'll lug the cart by hand, which ranges from much slower to abysmally slow. "Connectivity" is quite lenient, however - in most cases, a tile only needs one track connection in the correct direction, and bridges are accepted as track, too. It's best to just engrave/build an identifiable unbroken track, though. Guided track can go up/down ramps without trouble, all at the normal dwarven walking speed.

### 3.2.4 Lesson Four: Flight

Carts can be sent over ramps or over the lips of cliffs, and the game will trace a ballistic trajectory. Carts in flight are not subject to air friction (according to hack scripts), but they are subject to gravity. Someone did the calculation, I forget. Anyway, observation tells us that a free-falling cart takes as long to reach the bottom of a shaft as one rolling down a flight of ramps. Thus, the acceleration is the same - corrected for the greater length of ramps ( $\sqrt{2}$  times length of a flat tile), we get something just under 0,035 zlevels/step<sup>2</sup>. (Which shows that dwarven physics are screwy, acceleration on a ramp should be lower than free-fall acceleration.)

A cart released by a hatch takes six steps before it's displayed on the next level down, which suggests - hm, that the cart is considered to start falling about 2/3 up the current level? There's some tricky stuff going on with the decision whether a cart's actually in contact with the floor (and thus subject to corners, rollers and track stops): carts can make small jumps in some cases which don't move them to a different level, and in those cases it seems to take ~those six steps before they start registering as "on floor" again.

A cart pushed off a cliff follows an ordinary downward curve. It keeps its horizontal velocity and will keep moving at the same speed when it lands, while vertical speed will build up during the fall and will completely disappear when it hits the ground.

If a cart is sent over an upward ramp into the open sky, it can go up several levels, depending on its speed. A highest-speed roller will barely manage a hop, the cart won't even reach the level above the ramp, but it'll be in flight for a few steps. A cart accelerated by a long downward slope or an impulse ramp array can go over the ramp at much higher speeds and can reach heights of up to 26 z-levels (or more with added trickery). The "launch ramp" converts the horizontal speed of the incoming cart into ramped-upward velocity, and the upward component will grant height while gravity nibbles away at it.

Counting steps and trying to calculate out the results, my best estimation for ramp launches is as follows: The baseline is the speed on horizontal track. This speed is converted into speed calculated for ramps. When released, the cart moves vertically at 1/2 the original speed and horizontally at ~70% of the original speed. Assuming this is all ramp stuff, it's likely  $\sqrt{1/2}$  the original speed horizontally. As per usual, vertical speed disappears upon landing and if the cart is launched off a ramp again, its horizontal speed will be 1/2 the original, vertical speed  $\sqrt{1/8}$  (ca. 35%) and the height reached will only be about one half of what the first jump achieved (a bit less because of ramping speed costs).

Standard design for a launch ramp:



Fast cart comes from the west, goes over the ramp, flight happens. It must be a proper track ramp :P

Carts that fail to enter a hole in the floor "jump" over it, and this also seems to count as flight: speedy carts will not follow a track going down a ramp when coming from level track, and they will ignore corners directly behind the hole because they haven't touched floor again.

The peculiar feature of speed supercharging still exists in 0.40.11: if two carts of similar speed collide frontally and the "pushing" cart is between 1 and 100% heavier than the "pushed" cart, momentum of the pusher will be conserved. That's to say, the pushed cart will move off at a speed higher than what the pushing cart brought to the collision. This allows breaking the speed limit on ramp and gravity acceleration (270.000 reportedly). Carts moving that fast are subject to an exceptional friction of 10.000 per step, all the time, thus only very short bursts of extreme speed are possible and since high-speed collisions are required, no cargo can be transported. In a quick-and-dirty test for 40.11, I just smashed two hazel wood carts together, one loaded to double weight, and right enough, the pushed cart moved 29 tiles in six steps. In .34.11, I managed burst speeds of up to 17 tiles/step through tiered collisions and ramped jumps of 45 z-levels. The latter was what I meant with "added trickery" above.

Bodycount: 6 dogs (+1 since last update), one mangled dwarf (survived and is fine, but keeps cleaning himself).



### 3.2.5 Lesson Five: diagonal movement and how to fix it

Diagonal movement, on the face of it just means that a cart is not moving in a cardinal direction and will eventually move off the “straight line” or bump into a wall, stopping dead.

I admit that this is just interpretation, but i’m reasonably certain that diagonal movement is not handled as a “heading” like “fifteen marks east off north” but rather as a combination of movement on the two flat axes.

Laborious example: A cart pushed north by one highest-speed roller, then east by a lowest-speed roller doesn’t move “north by northeast” but rather “50.000 north and 10.000 east” and each of these components is separately subject to floor friction. Letting the cart roll over higher-friction floor (like non-track floor) shows that the cart will only take five steps (and three tiles) to move the first step to the east (since its eastward movement started in the middle of the tile, it only needs to move half a tile to switch over to the next), twelve steps and six tiles for the next, 22 steps and nine tiles for the third, and it won’t make a fourth step to the east: after fifty steps, the eastward component of the cart’s movement should be entirely gone. (It would take a rather unfeasible 1000 steps on track-engraved floor.)

Admittedly, accepting the sideways aberration and trying to remove it by floor friction is rarely an option.

Diagonal movement commonly occurs *when a cart moves up a corner ramp*. Since minecarts don’t care about flat-floor track apart from corners, a long straight track line will do nothing to rule in a diagonally-moving cart, it’ll just move along and take its sideways step when it’s time. And if there’s a wall next to the track (e.g. because you’re trying to keep accelerating the cart via impulse ramps) it’ll just hit the wall and stop, at least temporarily. If it stops on flat track, it’ll stop for good, if it stops on a ramp, it’ll start moving again, but it may lose its load. As far as I can tell, that was the problem encountered in [this water gun design](#). Thanks to uncorrected truetype font turning all text into garbage, I can only guess (and you better ramp speed up to 1000+ and “step” the thing yourself by hitting forward/pause repeatedly).

Note: in my experience, a cart always gets one ramp-step’s speed (i.e. about 5000, 1/20 tile/step) to the “outside” of the curve on the corner ramp. It will step off the straight path on the eleventh step after the corner, i.e. after this lateral speed component has accumulated half a tile of distance. This holds both for a cart propelled by a highest-speed roller (50.000 speed) and a maximum-speed cyclotron (265.000); both will stop/go off the straight path after ten steps.

I’ve re-built WanderingKid’s impulse/something elevator and found the problem he faced ([reported here](#)) was also nothing fancier than diagonal movement: sending the output of a corner ramp onto a straight (i.e. inconsequential) track. In my re-build, the cart would move off the straight line on the eleventh step after the corner.

So how to avoid diagonal-movement troubles?

The easiest option is not to generate diagonal movement in the first place: don’t use corner ramps to move carts up levels. For moving carts up levels, straight ramps work just as well as corner ramps; better in fact, since they don’t cause the added 1000 speed loss from the corner (and don’t cause diagonal movement). There are some special cases of upward movement over multiple levels which require corner ramps, but if you only want to go up a single level, just use a straight ramp.

The other option, when corner ramps are used, is to use the one track type carts care about: corners.

If a cart tries to leave a corner tile, the game checks whether the border the cart tries to leave over is “blocked” by the corner: on a NW corner, those will be the E and S borders. If a cart tries to leave to the south, it’s treated as coming from the north, and it leaves towards the west. This rule appears to only care for the tile border the cart tries to leave over. A diagonally-moving cart is also subject to these checks: let’s assume a cart moving from the northwest towards the southeast: if the tile the cart’d leave to would be the one directly south of the corner, the cart will turn around to the west and will move west only. Notably, the resulting speed is the cart’s previous N->S velocity, the W->E velocity will disappear. If the cart would have left to the eastern tile, it’ll turn north (moving at the previous W->E velocity). If the cart’s go-to tile is the exact southeastern one, the corner will not affect it. Which of the two axial speeds is higher doesn’t matter. A cart moving from northeast to southwest will only be affected by the corner if its go-to tile is the southern one. If it tries to leave to the western (or southwestern) tile, it’ll stay on its diagonal course, because the border over which it attempts to leave isn’t blocked.

My standard approach to the output of corner ramps is to just put a corner on the tile immediately behind the ramp, like this:

```
z+0      z+1
####
#      ++

####
#
track on ramp
```

I've yet to see a case where this doesn't work (if necessary propped up by a wall behind the corner above when working with fast carts).

PS: my best interpretation is that a corner "sets" the cart's speed in the exit direction to its previous value in the "input" direction. Since the diagonal component is actually velocity on the corner's exit axis, that part of the cart's movement speed just gets overwritten. Result in any case: successfully rounded corners fix diagonal movement.

Example of weird behaviour:

```
#
```

roller pushing south, medium speed (I didn't check all speeds, but highest is too fast). Track under the roller - doesn't matter, something inconsequential like NS or EW.

Upon first being pushed, the cart goes around the circuit normally. But when it then reaches the roller again, it will move south into the corner after two steps, then north after one to two steps, then south again and then once more through the loop. Interpretation: the cart is pushed into a southeasternish course, which is recognised as coming from the west by the corner, so it gets bent around to the north, reflected by the roller and then goes through the corner normally, entering from the north and leaving to the west this time.

### 3.2.6 Lesson Six: False ramps

In the ramps section, I mentioned ramps which don't accelerate carts. Those may seem kind of pointless for building tracks, but the lack of acceleration can actually be a benefit.

If a ramp connecting levels doesn't cause friction, you can change level without losing/gaining speed (apart from ordinary floor friction). It's decidedly weird - my constructions only work when the cart enters at very low speed - around that of a dwarven push - but a single push can move a cart up/down 40+ levels without notably changing the cart's speed. (example (o hey, it was 47 z. You can safely speed past the end, I just showed that each ramp was a non-functional E-only one.) It's of course also possible to do this without dwarven labour, you just need sufficiently regulated cart speeds from proper ramps or rollers, if needed combined with a few track stops. A super-low-tech and low-risk way of lifting a cart up a huge number of levels.

Another application of false ramps is to make the loading of liquids into carts easier [pioneered by flameaway](#). I found it to be an impressively fast, fully-automatable loading mechanism for waterguns allowing cadences of up to one shot per ten steps (using multiple carts in one barrel). It works so well because it *doesn't accelerate/decelerate the carts*. The loader simply consists of a single channelled-out tile containing a track ramp with no actual down direction. Its track connections only go to wall, therefore it is treated as ordinary flat floor by the game. The cart is never at the "bottom" of the "ramp", because as far as the minecart engine is concerned, there's no ramp here. Thus, the cart also doesn't need to "climb" out of the hole, it just needs enough forward motion to roll to the next tile.

A cart moving slowly enough will pick up water/magma from a 7/7 tile; the speed imparted by a high-speed roller is just low enough. Dwarven pushes have the advantage that they "teleport" the cart to the middle of the first pushed-to tile, which makes them the fastest loading event. They're decidedly less automatable, though. There's no need to engrave a corner into the pond tile, a straight fake ramp works better.

Bodycount: nothing new! Well, one diagonal vs. roller test ended up giving a dog a bruised stomach. Big deal, I don't really count dogs if they don't end up in multiple parts, like the puppy that during the last round teleported its torso through a wall while leaving all its limbs on the other side. The highly irresponsible flying minecart test, however, didn't cause any harm at all.

### 3.2.7 Lesson Seven: Pathing across levels

Pathing on flat floor is easy enough: only corners matter. It's not quite so easy when minecart paths go to different z-levels, either up or down.

Getting a cart to move upwards is easy enough - just offer it a track ramp. Carts will not go up ramps without engraved track, and they will not reliably go up "false" ramps (i.e. ramps which don't accelerate/decelerate carts). You'll eventually want the cart to stop going up, and there things can go awry. A cart moving up a ramp with no closed ceiling (or building) immediately above the exit tile may get airborne. The speed from a highest-speed roller is enough for this, but high-speed rollers or equivalent speeds like the acceleration from a single down ramp can suffice, too. An airborne cart will not be in contact with the floor underneath it and will thus not care about track corners, rollers or track stops on that tile.

A closed ceiling or building (bridge, hatch cover etc.) above the exit tile will make the cart behave and stick to the floor, regardless of its speed - a high-speed roller cart will be reined in by a ceiling just the same as a highest-ramped-speed cart or a supercharged cart.

If there's open ceiling above the exit tile, a cart can still be ruled in by a *functional ramp* on the exit tile.

z+0		z+1, a)	b)	c)	d)
#####		#	#	#	#
#					
#####		#	#	#	#
#					

Cart comes from the west, accelerated by a series of impulse ramps, then goes over an up ramp. a) - no ramp (can be smoothed floor instead of straight track): cart goes into flight, several z-levels up. b), c), d): cart goes down the ramp to the east and follows the track. Notably, the orientation of the ramp on the top tile doesn't matter, it just needs to be a legal ramp. Carts can be made to "level out" via ramp, but as seen here, they can also be forced down an adjacent ramp this way.

So, if you send a cart up several levels to the surface and don't want it to go flying, put a ramp on the exit tile.

When you want a cart to enter a downward path, there are a few issues and solutions, as well:

A cart coming upon a hole in the ground will by default just jump across it. If the cart moves at a speed of at least 1/5th of a tile per step, it can jump over one tile of open space and continue moving on flat floor on the other side. A dwarven push or low-speed roller are enough for this purpose. A peculiar issue was found with dwarven pushes: a dwarf pushing a cart from right next to a hole in the floor cannot move the cart across. It will collide with the hole's edge and fall down into the pit. This seems to happen because the push "teleports" the cart to the middle of the adjacent tile, without giving it the "lift" gained by a jump. If there's one tile of "buffer" between the dwarf and the hole, the cart jumps just fine.

If there is a ramp in a hole (ordinary floor ramp or track ramp, both are recognised), a cart will treat the hole as an appropriate pathing destination and will directly move into it (i.e. without spending time in the "open space" above the hole) *as though it were rounding a "downward" track corner*. Carts moving at derail-capable speeds will not enter a downward ramp, they'll jump over the tile and continue beyond it. In addition, the tile before the ramp must be a "track" tile - either engraved track or a bridge. Carts coming from ordinary floor will jump, regardless of their speed.

As noted above, however, a cart coming *from* a legal track ramp (any orientation!) will enter a downward track ramp just fine. This allows sending very fast carts down ramps simply by putting an impulse ramp before the actual ramp entrance:

```
. #          #
```

Other ramp orientations seem to work just the same, as long as they're legal and don't open a diverging path. Ramps will *not* send a cart into a hole that doesn't contain a ramp.

### 3.2.8 Lesson Eight: Meet the checkpoint bug

Let's face the possibly most powerful feature/bug of minecarting. Nope, not impulse ramps. For demonstration purpose, let's take two sets of opposed ramps:

```
a)          b)
##          ##
##          ##
```

Offer open floor above and to the sides.

Drop a cart onto one of the ramps via hatch. In each case, the cart will start out by rolling along a ramp for five steps.

In a), the cart will then pass over the flat tile *in a single step*, spends eight steps on the opposing ramp, rolls across the middle tile in a single step again, spends seven steps on the first-touched ramp, then across in a single step etc., until after a few iterations it sits still in the middle tile.

In b), the cart goes onto the opposing ramp, passes over it *in a single step*, goes to the tile above and to the side, passes over that tile in a single step again and then moves off at about 1/5 tile per step (~19 000 speed). If you offer no exit, the cart will bounce between the two ramps forever, spending eight steps on each. You can temporarily stop it by blocking the opposite ramp with another minecart, but as soon as one cart is removed, the remaining cart starts bouncing again.

What we're seeing is an artefact of the game having to switch distance calculations as soon as ramps get involved. The upshot is that

a) if track changes from flat track to a ramp, the cart *must* step onto the new ramp tile. No matter how fast the cart is, the tile cannot be skipped. I'll call this a "half checkpoint". b) if track changes from a type of ramp to *anything else*, the "changed" tile cannot be skipped and the cart will spend *exactly* one step on it, regardless of its speed (as long as speed is above zero). Finally, the last speed increment the cart received on the ramp is erased, presumably by applying equivalent acceleration in the opposite direction. I'll call this a "full checkpoint".

"Anything else" notably means that checkpoints happen whenever the cart passes from a ramp to a *different* ramp, i.e. a ramp with a different slant (accelerate-to direction), and when passing to a non-ramp tile, preferably flat track.

The biggest effect here is that **checkpoints effectively divorce the rate of movement from internal speed of the cart.**

Cart propelled by a single ramp (about 1/3 tile per step) going over checkpoint? Spends exactly one step there. Cart propelled by maximum number of ramps (about 2,5 tiles per step) crossing checkpoint? Spends exactly one step there. In fact, if a cart is moving along a ramp- and corner-heavy track and crosses one tile each step, it's almost a given that you're dealing with chained-up checkpoints.

Simple example:

```
#####          #####
```

A cart going in at sufficient speed (must be ~72 000+) will cross this track spending one step on each tile and will come out on the east at almost exactly the speed it went in. This holds both for a 72 000 speed and a 265 000 speed cart, they'll move at the same rate through this track, they'll only lose the speed for normal track friction but the slower

cart will also not accelerate. Their actual internal speeds will only again assert themselves after the cart left this track section.

This happens because each impulse ramp is a half and each flat tile a full checkpoint. The slower cart is just fast enough to make it off the ramp in a single step (apparently a cart moves its full movement rate “into” a half-checkpoint (but not past it when moving faster than one full tile per step): a fast-enough cart makes it to just past the half-way point of the ramp upon entering, and just past the tile’s “exit” on the very next turn). PS: I haven’t checked this exact design, but as long as incoming speed is at least 80.000, this thing should work the same way *in both directions* - carts going “with” the impulse ramps won’t accelerate, and those going “against” them won’t slow down.

Let’s look at the first example with the double-ramp again and see what happens by checkpoint rules, dropping the cart onto the western ramp:

-cart goes “down” ramp to the east, picks up 25 000 speed. -cart enters ramp slanting west - checkpoint: accelerate 5000 to the west (compensating for last step of acceleration), go to end of tile -cart “accelerates” west by 5000 on the west-slanting ramp, has 15 000 speed left to cross the threshold to the next tile, thus reaches flat tile above and to the east - checkpoint: accelerate 5000 east (compensating for westward acceleration), go to end of tile - cart keeps moving on flat track to the east, now with normal distance calculations so it takes five steps per tile again.

Why the weird “accelerate backwards on the checkpoint” thing? Because in example a), the cart actually stops. It also explains why the highest speed i’ve got through ramps (measuring actual track covered) is not 270.000 but 265.000.

For a clearer example:

```
#+      #
```

+ lever-operated door

Station a cart on the ramp, then open the door. The cart instantly rolls onto the flat tile *and stops there*. This is, it picked up speed from the ramp, used that speed to pass over to the flat ground, but had no speed left thereafter (or it’d have moved to the next tile east on the next step). I interpret this so that the cart actually loses its speed after taking the move. Other evidence supports the interpretation.

This bug allows deriving speed from pits in the floor and moving carts up levels with ease. It’s the actual power behind the “impulse elevator” shown on the wiki. WanderingKid’s elevator uses impulse ramps to gain speed, but checkpoints to go up levels.

I’ll leave you with this for now. More to come.

Bodycount: kitty!

Someone’s pet cat wandered into the cyclotron. It’s the only contraption that has caused any real damage so far, and the only dwarf who was hurt remains the spinner/leatherworker who tried to “clean” puppy blood out of it while it was spinning.

### 3.2.9 Lesson Nine: Practical implications of the checkpoint bug

The checkpoint bug affects all manner of minecart constructions, as soon as ramps get involved. For a start, let’s look at the lowly single-ramp cyclotron:

```
#####      #####
##          ##
##          ##
#####      #####
```

Cart cycles counter-clockwise and its speed oscillates somewhere between 70,000 and 80,000.

It won’t go any faster, ever, although one step of ramp acceleration gives 4900 speed while four corners and, say, seven steps of movement cost no more than 4070. Evidently, if the cart spends only one step on the ramp, this acceleration

is eaten up by the checkpoint compensation when moving off the ramp to level floor. It'll only really pick up speed when it spends at least two steps on the ramp and it must be slower than ~72.000 for this to happen.

Indeed, the cart cycles at an oscillating speed: it goes five rounds at eight steps each (spending two steps on the ramp each time) and seven steps in the sixth round (spending only one step on the ramp).

For speed to keep building up, you need an unbroken stretch of three impulse ramps: due to the greater length of ramp tiles, the maximum speed available through ramps (270.000) is just less than two ramp tiles per step, so a cart will always spend at least two consecutive steps on the three-ramp stretch. Such a three-ramp cyclotron is enough to achieve maximum ramp speed.

When moving a cart laterally onto an impulse ramp track, the checkpoint effect can be used to prevent diagonal movement.

Throwing a cart directly into a sideways impulse ramp:

a)		b)	
####	####	####	####
		#	#

from the south like in a) will have the cart accelerate to the west on top of a pre-existing and lingering northward speed. It'll either bump into the wall and temporarily stop or exit the impulse stretch on a diagonal trajectory. Sending it through an immediately adjacent impulse ramp lets it pass right through the first ramp of the acceleration stretch via checkpoint effect, stopping it against the wall and cancelling the northward speed instantly, so that it can accelerate west on a straight course.

Of course, others have, often unknowingly, used checkpoint effects in their constructions. Take the "impulse elevator" on the wiki:

```
##### ### ##### # ### # ### ## ###
```

All track on ramps, going up from left to right.

Looking at the thing in action, we'll see that the cart moves at a rate of exactly one tile every step until after five levels or so it stops, rolls back from an "up" ramp in eight turns, spends another eight steps on the ramp behind, then starts going at the previous rate for another five levels. Clearly, this means that the cart moves at one ramp-length per step, i.e. 140.000 speed, right?

Haha, of course not. It's checkpoints all the way up. The cart hiccups and stops not because it's too fast, but because it ran all out of speed and had to checkpoint-cheat itself some new steam.

Observe the ramp slants in the example above: E, W, N, S, W, E. Slant changes every tile, thus every tile is a full checkpoint. The checkpoint bug runs the cart up at a rate of one ramp every step, until speed falls to zero. At that point, the cart makes it onto the next tile (and technically all the way "up" on it) but has no more speed to make it to the next tile (up), so it stays on the ramp and accelerates there for the full eight steps. This moves it back to the last (opposing) ramp, which it again fully crosses, but here it bumps against a wall and accelerates all the way forward again. With the shiny new 35.000 speed, it can take the up checkpoint and have speed leftover to keep moving.

It's peculiar that this thing loses speed so quickly - it appears to burn through its store of ~35.000 speed points in five levels, although it should only lose 1.000 speed per level for the corner. It's almost as if there's something fishy with corner ramps that enforces a higher speed loss.

Another ramp spiral was invented by WanderingKid and has the advantage of doing without the annoying back-and-forth every few levels. The cart in that design just keeps going. Let's check it out:

z+0	z+0, track	z+1	z+1, track	z+2 (z+0 mirrored)
####	####	####	####	#
###	###	###	###	###
#	#	###	###	####
####	####	####	####	####

This one surprised me at first, because it “somehow” manages to send a cart up *two* levels, seemingly with a single checkpoint. Spoiler: of course it's two checkpoints.

The east-pointing ramp on z+0 works as a proper speed-granting impulse ramp here, because the cart enters it from flat floor, not from another ramp. When I tried it out, the cart spent two or three steps (repeating pattern of different rates, like in the cyclotron above) on the ramp each time, so there was always speed gained here. The corner up ramp is, unsurprisingly, a checkpoint, the cart passes it in a single step. What I hadn't fully understood yet - the next, straight, ramp is *also* a checkpoint, because the slant of ramps changed, from west to south. The flat corner is yet another full checkpoint, which doesn't really matter in and of itself, but the fact that it's normal floor and not a ramp saves the following impulse ramp from being a full checkpoint, so it can actually do its impulse work.

Let's crack an old puzzle next: the 2x2 ramp spiral. It's a notoriously ill-behaved contraption, carts keep stopping on it for no discernible reason. At the same time, it looks so simple:

```
#####      #####      #####      #####      #####
##          ##          ###          #####          ###
##          #####          ###          ##          ###
#####          #####          #####          #####          #####
```

Spread over four levels, one corner on each level, each leading into the next. Throwing a cart down such a spiral lets the cart start going at one ramp per step, but after five, it stops, starts again, goes another five, stops again etc.

Ho hum. Is it picking up too much speed? I put a few stone blocks into a cart and sent it down there. The blocks stayed in the cart. Well, it was moving at one ramp per step, so it was probably checkpoint-hopping again. Makes sense, of course, since ramp slant changes on every tile. So it probably stopped simply because its speed dropped to zero. Still, a cart going *down* a ramp spiral and losing speed? I revved up a cart in the trusty cyclotron and sent it down a nice long spiral. It kept going and emerged 21 z-levels below - at 130.000 speed. The cart was definitely losing ~6.000 speed on every ramp, a few more tests confirmed this. In fact, a downward spiral slows down a cart exactly as much as an *upward* spiral.

Inspired by [rhesusmacabre's long table](#), I built a few simple test spirals, and yes, I was getting checkpoint-movement up the spirals, over nice large numbers of levels, and my eyeballed speed loss of 6.000 per level seemed to work out.

I definitely needed to crack the puzzle of corner ramps. But first, some light entertainment.

Since different-slant ramps work as checkpoints for each other and the compensating speed effects cancel out their acceleration, shouldn't it be possible to send *reallllly slow* carts along a line of impulse ramps, bouncing one ramp per step until ramps stopped and the actual speed reasserted itself? I built a line of 24 impulse ramps stretching from east to west and with wall to the south, alternating between NS and SW every step, hatch-dropped a minecart on the easternmost (SW) ramp and watched it. Yep, cart rolled down the usual five steps, then went forward at a rate of one ramp every step over the whole line, and once it emerged from the ramp line, it crawled along at the actual ~19.000 speed (five to six steps used for every tile).

But shouldn't the northward acceleration, although it's cancelled instantly, result in a minor northward displacement on every NS ramp that should eventually push the cart past the northern border? I expanded the row to ~40 ramps, and sure enough, after the thirtieth ramp (15th NS ramp) the cart moved off the ramp-line to the north. To make sure it's really displacement and not northward velocity, I covered ten ramps with a bridge so that north-pointing ramps #15 to #19 were obscured. The cart moved over this stretch without diverting, went over the SW ramp directly behind the bridge - and made its step to the north when it checkpoint-passed the NS ramp behind it, the twentieth northward ramp in the line, but this time, the fifteenth touched by the cart.

Fifteen pushes of presumably 4900 distance units give 73500 distance units, just over half the assumed length of a ramp (140.000 or so - I don't know the exact number Toady uses). Enough to move over the border to the next tile when starting in the middle of a tile. Seems that it works out.

Of course, northward displacement can simply be compensated by southward displacement. I dug out a track all across the embark (normal embark, so just 190ish tiles) and carved out a nice stretch of 160ish alternating track ramps. First ten “forward” ramps interspersed with 10 North-slanted ramps, then (changing the adjacent wall) 20 forward with 20 south-slanted, then another 20/20 stretch forward/north etc., finally a bit of flat track leading into a little loop at the far



end. The cart was dropped in via hatch as usual and moved all across the embark without falling off the row, passing one tile per step as long as it was bouncing over ramps, while the flat track at the end demonstrated its internal speed remained at the original 19.000. The loop itself contained a nice juicy acceleration rail, increasing speed on the route back to ~120.000, and the cart went back all the way, once again at 1 tile/step externally, unfazed by the 80 “opposing” impulse ramps.

### 3.2.10 Lesson Ten: Corner ramps

Corner ramps had been bugging me for a while now, so I built a simple test rig:

above	below
#	#

With a SE () track ramp.

First of all, send a cart up the ramp: no matter what I do, when given straight track, the cart will move diagonally and the first step aside happens after 11 steps, adequate for a lateral component of just under 5000 speed, i.e. the acceleration gained by a single step on a ramp. Curiously, while the corner should convert all south-to-north velocity of the cart into west-to-east velocity and the ramp slants to the south, the aberration was to the north.

Unsurprisingly, the culprit is the checkpoint bug: almost always, a corner ramp passed upward leads to a checkpoint - the ramp slants south and the most sensible connections above are flat track or a west-slanting ramp. Thus, the checkpoint effect is applied: a) the next tile is crossed in a single step. b) compensative acceleration is applied *which is opposed to the ramp's slant*. That's it - the corner outputs the cart on a pure-eastward path but *then* the “compensating” speed is applied and gives the much-abhorred diagonality to the cart.

So, putting it in numbers: when a cart checkpoint-hops up a corner ramp, it loses 5000 from its original incoming speed to ramp acceleration, loses another 1000 for the corner, and the checkpoint doesn't “refund” the 5000 speed but rather (since it's applied after the corner turn) applies it as lateral/diagonal speed towards the “outside” of the corner. **A cart going up a corner ramp at any speed loses 5000x(time on ramp)+1000(corner penalty) speed, and gains 5000 lateral.**

That was the easy part.

Let's send a cart *down* the ramp now.

If the cart is fast enough (about 45.000 minimum), it takes the corner and continues perfectly straight in the corner's exit direction, with a speed loss of ca. 6000. I tried it with a highest-speed roller, and the cart going through a corner ramp would emerge at 44.000 speed, while a cart going down a straight ramp would gain ca. 5000 and emerge at 55.000. Once again, we're dealing with checkpoints and a corner, so let's step through it: On the corner ramp, all acceleration goes to the side, it doesn't accelerate the cart in its original travel direction. Here, we have a cart going west, which is accelerated south. Unsurprisingly, the westward speed isn't increased by this event. At the *end* of the turn on which the cart wants to leave the tile, the corner comes into play, converts all westward to southward motion overwriting the extant southern vector, the acceleration gained is therefore lost. On the next step, the cart reaches a checkpoint and to compensate, it is “accelerated” 5000 units to the north. Summa: all southward acceleration was ignored because of the corner, but the compensative deceleration still applies, so the cart loses 5000 speed, plus 1000 for the corner. 6000 in total.

What's that about a 45.000 minimum speed? Ah well, losing speed on a down ramp is not the weirdest thing here. A cart moving at lower speeds than that is liable to malfunction even more blatantly. A cart propelled by a dwarven push emerges at a mostly-south-and-slightly-west trajectory, going off the straight line after two tiles. A cart entering the ramp at between 30.000 and 40.000 speed leaves at an almost-45° angle, a very sharp diagonal. It took me quite a while to think up a solution for that one, but I think it works out:

Corners are only checked when a cart tries to leave a tile, and they only check whether the side opposed to the “border” over which the cart is trying to leave is connected. In understandable: if a cart on a southwest heading is trying to leave

a tile going over the western border of the tile, the pathing algorithm checks if the tile underneath is a track corner with an eastern connection. If yes, the cart is turned around towards the corner's other connection. If the cart tries to leave over the southern border, the algorithm checks whether the tile is a north-connected corner. If the checked border is not connected or if the tile isn't a corner, the cart leaves normally and its speed(s) is (are) unchanged.

So what happens with these slower carts is this: they move so slowly to the west and thus pick up so much southward speed on the ramp, that the cart's "exit" direction from the tile is south (or SW (??) in the case of the somewhat-slow cart), and thus the corner has no power over them. Consequently, they move off on their screwy diagonal course.

**A cart going down a corner ramp, properly taking the corner, loses 5000(checkpoint compensation)+1000 (for the corner)=6000 speed, independent of lingering time on the ramp.**

If time on the ramp is too long, the corner starts checking the wrong (unconnected) side of the tile when the cart tries to leave and no longer applies. In that case, the output trajectory is purely diagonal, presumably incoming speed in the incoming direction +  $5000 \times (\text{lingering time} - 1)$  lateral (towards ramp slant), no corner penalty.

Bodycount: nothing new, no new tests required. I just wrote up what I had worked out previously.

This concludes our course on Minecarts. Annotations, corrections, claims of priority will be gracefully accepted and carefully considered. Possibly. I've tried to link to sources and earlier findings. I owe a large debt to other players for their research and inspiring inventions.



## 4.1 Sharing and resuse



This project is published under the Creative Commons Attribution-ShareAlike 4.0 International License. I think that this is the best license for community-driven creative projects (the GPL is best for software). To quote the [Creative Commons website](https://creativecommons.org/licenses/by-sa/4.0):

This license lets others remix, tweak, and build upon your work even for commercial purposes, as long as they credit you and license their new creations under the identical terms. This license is often compared to “copyleft” free and open source software licenses. All new works based on yours will carry the same license, so any derivatives will also allow commercial use. This is the license used by Wikipedia, and is recommended for materials that would benefit from incorporating content from Wikipedia and similarly licensed projects.

You should have received a copy of the license along with this work. If not, see <https://creativecommons.org/licenses/by-sa/4.0>

Creative Commons Attribution-ShareAlike 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept **and** agree to be bound by the terms **and** conditions of this Creative Commons Attribution-ShareAlike 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights **in** consideration of Your acceptance of these terms **and** conditions, **and** the Licensor grants You such rights **in** consideration of benefits the Licensor receives **from making** the Licensed Material available under these terms **and** conditions.

Section 1 -- Definitions.

(continues on next page)

(continued from previous page)

- a. Adapted Material means material subject to Copyright **and** Similar Rights that **is** derived **from or** based upon the Licensed Material **and in** which the Licensed Material **is** translated, altered, arranged, transformed, **or** otherwise modified **in** a manner requiring permission under the Copyright **and** Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material **is** a musical work, performance, **or** sound recording, Adapted Material **is** always produced where the Licensed Material **is** synched **in** timed relation **with** a moving image.
- b. Adapter's License means the license You apply to Your Copyright **and** Similar Rights **in** Your contributions to Adapted Material **in** accordance **with** the terms **and** conditions of this Public License.
- c. BY-SA Compatible License means a license listed at [creativecommons.org/compatiblelicenses](https://creativecommons.org/compatiblelicenses), approved by Creative Commons **as** essentially the equivalent of this Public License.
- d. Copyright **and** Similar Rights means copyright **and/or** similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, **and** Sui Generis Database Rights, without regard to how the rights are labeled **or** categorized. For purposes of this Public License, the rights specified **in** Section 2(b)(1)-(2) are **not** Copyright **and** Similar Rights.
- e. Effective Technological Measures means those measures that, **in** the absence of proper authority, may **not** be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, **and/or** similar international agreements.
- f. Exceptions **and** Limitations means fair use, fair dealing, **and/or** **any** other exception **or** limitation to Copyright **and** Similar Rights that applies to Your use of the Licensed Material.
- g. License Elements means the license attributes listed **in** the name of a Creative Commons Public License. The License Elements of this Public License are Attribution **and** ShareAlike.
- h. Licensed Material means the artistic **or** literary work, database, **or** other material to which the Licensor applied this Public License.
- i. Licensed Rights means the rights granted to You subject to the terms **and** conditions of this Public License, which are limited to **all** Copyright **and** Similar Rights that apply to Your use of the Licensed Material **and** that the Licensor has authority to license.
- j. Licensor means the individual(s) **or** entity(ies) granting rights under this Public License.
- k. Share means to provide material to the public by **any** means **or** process that requires permission under the Licensed Rights, such **as** reproduction, public display, public performance, distribution, dissemination, communication, **or** importation, **and** to make material

(continues on next page)

(continued from previous page)

available to the public including **in** ways that members of the public may access the material **from a** place **and** at a time individually chosen by them.

1. Sui Generis Database Rights means rights other than copyright resulting **from Directive** 96/9/EC of the European Parliament **and** of the Council of 11 March 1996 on the legal protection of databases, **as** amended **and/or** succeeded, **as** well **as** other essentially equivalent rights anywhere **in** the world.
- m. You means the individual **or** entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

## Section 2 -- Scope.

### a. License grant.

1. Subject to the terms **and** conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights **in** the Licensed Material to:
  - a. reproduce **and** Share the Licensed Material, **in** whole **or** **in** part; **and**
  - b. produce, reproduce, **and** Share Adapted Material.
2. Exceptions **and** Limitations. For the avoidance of doubt, where Exceptions **and** Limitations apply to Your use, this Public License does **not** apply, **and** You do **not** need to comply **with** its terms **and** conditions.
3. Term. The term of this Public License **is** specified **in** Section 6(a).
4. Media **and** formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights **in** all media **and** formats whether now known **or** hereafter created, **and** to make technical modifications necessary to do so. The Licensor waives **and/or** agrees **not** to **assert** any right **or** authority to forbid You **from making** technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a) (4) never produces Adapted Material.
5. Downstream recipients.
  - a. Offer **from the** Licensor -- Licensed Material. Every recipient of the Licensed Material automatically receives an offer **from the** Licensor to exercise the Licensed Rights under the terms **and** conditions of this Public License.
  - b. Additional offer **from the** Licensor -- Adapted Material.

(continues on next page)

(continued from previous page)

Every recipient of Adapted Material **from You** automatically receives an offer **from the** Licensor to exercise the Licensed Rights **in** the Adapted Material under the conditions of the Adapter's License You apply.

c. No downstream restrictions. You may **not** offer **or** impose **any** additional **or** different terms **or** conditions on, **or** apply **any** Effective Technological Measures to, the Licensed Material **if** doing so restricts exercise of the Licensed Rights by **any** recipient of the Licensed Material.

6. No endorsement. Nothing **in** this Public License constitutes **or** may be construed **as** permission to **assert or** imply that You are, **or** that Your use of the Licensed Material **is**, connected **with, or** sponsored, endorsed, **or** granted official status by, the Licensor **or** others designated to receive attribution **as** provided **in** Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such **as** the right of integrity, are **not** licensed under this Public License, nor are publicity, privacy, **and/or** other similar personality rights; however, to the extent possible, the Licensor waives **and/or** agrees **not** to **assert any** such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but **not** otherwise.
2. Patent **and** trademark rights are **not** licensed under this Public License.
3. To the extent possible, the Licensor waives **any** right to collect royalties **from You for** the exercise of the Licensed Rights, whether directly **or** through a collecting society under **any** voluntary **or** waivable statutory **or** compulsory licensing scheme. In **all** other cases the Licensor expressly reserves **any** right to collect such royalties.

### Section 3 -- License Conditions.

Your exercise of the Licensed Rights **is** expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including **in** modified form), You must:
  - a. retain the following **if** it **is** supplied by the Licensor **with** the Licensed Material:
    - i. identification of the creator(s) of the Licensed Material **and** **any** others designated to receive attribution, **in any** reasonable manner requested by the Licensor (including by pseudonym **if**

(continues on next page)



(continued from previous page)

- designated);
- ii. a copyright notice;
- iii. a notice that refers to this Public License;
- iv. a notice that refers to the disclaimer of warranties;
- v. a URI **or** hyperlink to the Licensed Material to the extent reasonably practicable;
- b. indicate **if** You modified the Licensed Material **and** retain an indication of **any** previous modifications; **and**
- c. indicate the Licensed Material **is** licensed under this Public License, **and** include the text of, **or** the URI **or** hyperlink to, this Public License.
- 2. You may satisfy the conditions **in** Section 3(a)(1) **in any** reasonable manner based on the medium, means, **and** context **in** which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI **or** hyperlink to a resource that includes the required information.
- 3. If requested by the Licensor, You must remove **any** of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.
- b. ShareAlike.

In addition to the conditions **in** Section 3(a), **if** You Share Adapted Material You produce, the following conditions also apply.

- 1. The Adapter's License You apply must be a Creative Commons license **with** the same License Elements, this version **or** later, **or** a BY-SA Compatible License.
- 2. You must include the text of, **or** the URI **or** hyperlink to, the Adapter's License You apply. You may satisfy this condition **in any** reasonable manner based on the medium, means, **and** context **in** which You Share Adapted Material.
- 3. You may **not** offer **or** impose **any** additional **or** different terms **or** conditions on, **or** apply **any** Effective Technological Measures to, Adapted Material that restrict exercise of the rights granted under the Adapter's License You apply.

#### Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

- a. **for** the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, **and** Share **all or** a substantial

(continues on next page)

(continued from previous page)

portion of the contents of the database;

- b. **if** You include **all or** a substantial portion of the database contents **in** a database **in** which You have Sui Generis Database Rights, then the database **in** which You have Sui Generis Database Rights (but **not** its individual contents) **is** Adapted Material, including **for** purposes of Section 3(b); **and**
- c. You must comply **with** the conditions **in** Section 3(a) **if** You Share **all or** a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements **and** does **not** replace Your obligations under this Public License where the Licensed Rights include other Copyright **and** Similar Rights.

Section 5 -- Disclaimer of Warranties **and** Limitation of Liability.

- a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.
- b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.
- c. The disclaimer of warranties **and** limitation of liability provided above shall be interpreted **in** a manner that, to the extent possible, most closely approximates an absolute disclaimer **and** waiver of **all** liability.

Section 6 -- Term **and** Termination.

- a. This Public License applies **for** the term of the Copyright **and** Similar Rights licensed here. However, **if** You fail to comply **with** this Public License, then Your rights under this Public License terminate automatically.
- b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:
  - 1. automatically **as** of the date the violation **is** cured, provided it **is** cured within 30 days of Your discovery of the

(continues on next page)

(continued from previous page)

violation; **or**

2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does **not** affect **any** right the Licensor may have to seek remedies **for** Your violations of this Public License.

- c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms **or** conditions **or** stop distributing the Licensed Material at **any** time; however, doing so will **not** terminate this Public License.
- d. Sections 1, 5, 6, 7, **and** 8 survive termination of this Public License.

#### Section 7 -- Other Terms **and** Conditions.

- a. The Licensor shall **not** be bound by **any** additional **or** different terms **or** conditions communicated by You unless expressly agreed.
- b. Any arrangements, understandings, **or** agreements regarding the Licensed Material **not** stated herein are separate **from and** independent of the terms **and** conditions of this Public License.

#### Section 8 -- Interpretation.

- a. For the avoidance of doubt, this Public License does **not, and** shall **not** be interpreted to, reduce, limit, restrict, **or** impose conditions on **any** use of the Licensed Material that could lawfully be made without permission under this Public License.
- b. To the extent possible, **if any** provision of this Public License **is** deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed **from this** Public License without affecting the enforceability of the remaining terms **and** conditions.
- c. No term **or** condition of this Public License will be waived **and** no failure to comply consented to unless expressly agreed to by the Licensor.
- d. Nothing **in** this Public License constitutes **or** may be interpreted **as** a limitation upon, **or** waiver of, **any** privileges **and** immunities that apply to the Licensor **or** You, including **from the** legal processes of **any** jurisdiction **or** authority.

## 4.2 Contributing to this project

Are you interested in contributing to this project, or just interested in learning what goes into it? Then this is the page for you!

You can [find the project on Github](#).

There are two main areas of work: maintaining and improving the existing content (walkthrough chapters and tutorials), and adding new tutorials. It would be great if you wanted to help out!

I'm also very happy to take bug reports - just send me a message about small errors, typos, or formatting problems. Larger issues about content should be reported on the Github issue tracker, so I'm not the only one who can work on them.

You can also [support me on Patreon](#) if this sounds like a lot of work.

### 4.2.1 How to turn text into the website

Anyone who wants to contribute content, fix a bug, or just have an offline copy needs to turn the text files into the website. Luckily, this is pretty easy! You only need two things installed:

- **Python**. If you're on OSX or Linux, Python is probably already installed. If it isn't, or you're on Windows, install it from the site above.
- **Sphinx**. With Python installed, just run `pip install -U Sphinx` in a terminal / command prompt. If you're using an old Python version and that doesn't work, the site has alternative install instructions.

Get the code from the GitHub repo, either by forking the repo if you want to contribute, or using the 'download ZIP' option and unzipping the folder.

Congratulations, you now have the text! There's only one more step: open a command prompt in the DF-Walkthrough folder, and run `make html`

That's it!

### 4.2.2 Authors and Contributors

*Anyone who's contributed more than a bug report should be here - add yourself if you're not!*

Person	Contribution
PeridexisErrant	Project leader, editor, maintainer; does some of everything
TinyPirate	Author and inspiration for project
/u/buschwacker	Contributed chapters and many, many images
/u/Mechanixm	Wrote a great masterclass series
Larix	Wrote the Minecart masterclass

### 4.2.3 Contribution standards

This section describes the target standard for the project. Not all pages will meet it, and that's OK. Don't let them stop you adding something - but feel free to apply them to existing content!

#### Content

The walkthrough chapters cover core knowledge, with a clear progression of skills. If a topic can be put in a self-contained tutorial, it should be. Prefer linking to a tutorial over adding a section to the walkthrough.

The tutorials each cover a single, self-contained topic. Tutorials are aimed at players who have just finished the walkthrough. They introduce core topics not covered in the walkthrough.

A masterclass is like a tutorial, but covering an advanced topic for experienced players.

## Style

Use clear, direct, and simple language. Avoid jargon or the passive voice. I find the [hemmingway editor](#) useful; I often ignore but always consider it's suggestions.

Keep all lines to 80 characters or less. Sphinx will automatically join everything between blank lines into one paragraph, and short lines make the raw text easier to read for editors. More importantly, short lines make version-control software much more useful. When writing new text, keep to about 70 characters to avoid many lines changing for small edits later. Do not use tabs or leave trailing whitespace.

You can check with `python misc/lint.py && make clean && make html` - this sequence of commands should always run without errors, or at least give warnings about specific files and line numbers!

## Markup

Markup should be fairly minimal and let readers focus on the text. Basic markup helps with this, and should be used where appropriate. If in doubt, just match the surrounding sections.

Headings:

- File headings over- and underlined with #####
- Section headings underlined with =====
- Subsections generally avoided, but underlined with ----- otherwise

Lists:

- Bulleted lists use \* and one space. Use these instead of a paragraph for sections such as a list of examples or sequence of commands.
- Use numbered lists only when the ordering is both important and unclear from context. Use # . and one space, so renumbering will be automatic if the list is changed later.

Special text:

- Use bold, italics, etc. very sparingly - add meaning, not just emphasis.
- For keybindings, use the `:kbd:`Key`` directive, which is rendered with the special style. Keys are case sensitive, so `:kbd:`Shift`` should never be used (unlike `Esc`, `Space`, etc.).
- For in-game text, use the `:guilabel:`In game text`` directive, which is rendered with the appropriate font and background. Ensure that phrasing and capitalisation matches DF exactly (except ' , which renders as ' - leave it out). Don't use this style for ASCII art, only ingame menus which should be read as text.

Tips:

- Image names must not contain spaces
- Text files should be encoded in UTF-8 (your editor should have an option for this)

### 4.2.4 TODO list

*There's a lot to do; this list is roughly in order of priority but items may be done in other orders for whatever reason.*

1. Update remaining images
2. Run everything through Hemmingway
3. Add more tutorials; eg modding, quantum stockpiles, graphics, etc
4. Add an adventure mode walkthrough

## 4.3 Setting up your install

For most people, the [standard pack](#) is the best way to get started - but not always. If you're on OSX or Linux, or have some other reason to avoid the preset walkthrough pack, here's what you'll need:

1. [Dwarf Fortress](#)
2. [DFHack](#)
3. [A graphics pack](#)
4. The save file from the standard pack (yep, download it anyway!)

The easiest way to get all this working is usually via a [Starter Pack](#), which are also suitable for more advanced players (eg if you want to dive into the tutorials or even a masterclass).

Just add `this file` to your DF folder; it configures DFBHack to automate away many of the worst pain points for new players.

Instructions for how to set up all of the components individually are beyond the scope of this quick guide, so check the link for each and experiment if anything is unclear.

## 4.4 What is Dwarf Fortress?

Above anything else, Dwarf Fortress is a complicated game.

Enough has been written about this already, so this will simply link to some of the better articles for a background understanding.

Dwarf Fortress is a single-player fantasy game. You can control a dwarven outpost or an adventurer in a randomly generated, persistent world.

Although Dwarf Fortress is still in a work in progress, many features have already been implemented. . .

—Tarn Adams (developer of \*Dwarf Fortress\*)

“Losing is Fun”

—Community motto

Dwarf Fortress is a mixture of classic turn-based fantasy adventure games and sophisticated city building that has been confounding observers since 2006. It's visuals appear incomprehensible and its learning curve not just steep and unending, but slippery and sometimes electrified.

To look at Dwarf Fortress it is easy to imagine it as an anachronistic throwback to the likes of Rogue and Nethack, a cute little sort of retro tribute. However the truth of the matter, which becomes rapidly more apparent as you learn exactly what is being done within the game, is that this is not a primitive or old fashioned game. Dwarf Fortress is spectacularly sophisticated. The world that the game generates for you is created in the sort of meticulous detail that would typically demand several hundred pages of appendices at the end of a JRR Tolkien novel. Each created world has a history, legendary figures, towns, cities and people. You can create a world that is artificially aged up to a thousand years, or you can start with a younger world, either approach bringing with it different challenges. A new world might be untamed when a more established one might be home to established communities of potential foes such as goblins.

—NewStatesman

Players are responsible for the cultivation and management of a virtual ecosystem — a colony of dwarves trying to build a thriving fortress in a randomly generated world. Dwarf Fortress unfolds as a series of

staggeringly elaborate challenges and devastating setbacks that lead, no matter how well one plays, to eventual ruin.

Many simulation games offer players a bag of building blocks, but few dangle a bag as deep, or blocks as small and intricately interlocking, as Dwarf Fortress. Beneath the game's rudimentary facade is a dizzying array of moving parts, algorithms that model everything from dwarves' personalities (some are depressive; many appreciate art) to the climate and economic patterns of the simulated world. ...

—The NY Times Magazine





## CHAPTER 5

---

### About

---

The source text and images are [hosted on Github](#) and the finished site is published immediately to [readthedocs.org](#) every time a change is pushed.

This project was inspired by [TinyPirate's DF tutorials](#), which taught me to play DF. The walkthrough started as an update of these tutorials. Other content is original or (for some tutorials and most masterclasses) links back to the primary source.

Consider [supporting my work](#) on Patreon.