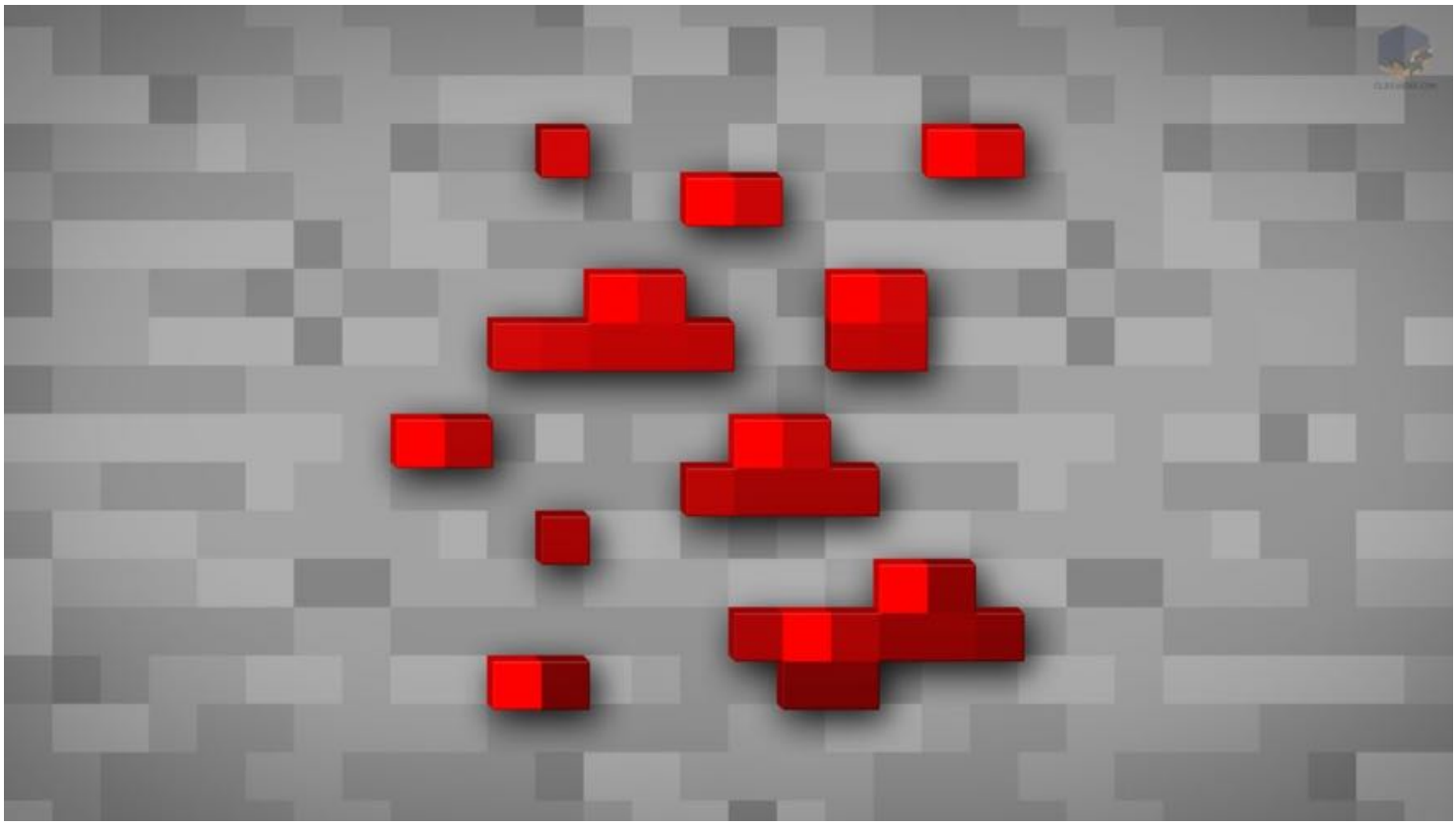


Minecraft U Sequence 2: Engineering with Redstone

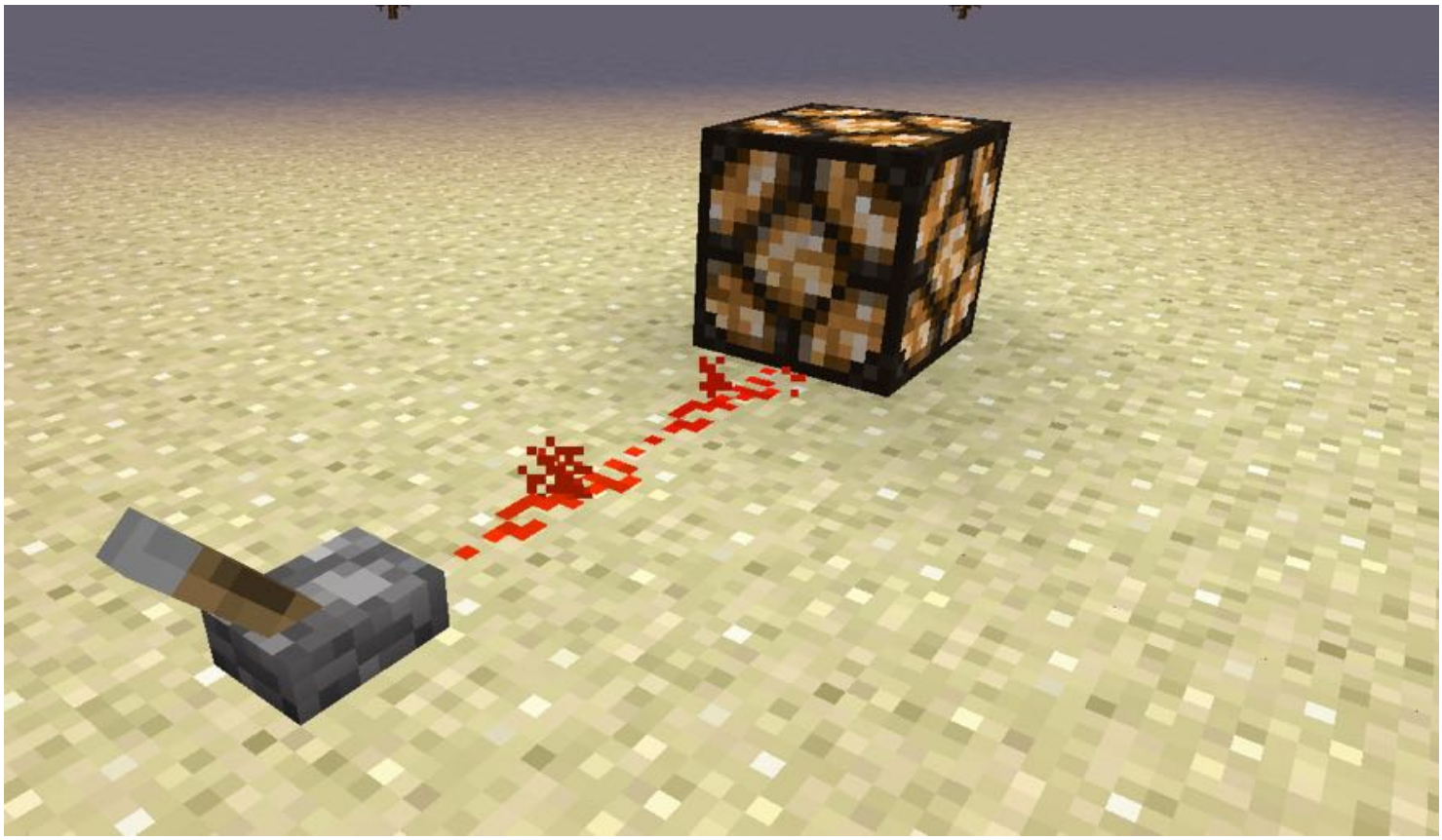
Learn the basics of electricity, with in game and real world applications. We'll go over digital circuits, like those found in smartphones and tablets, and how to use them in the Minecraft world to build yourself awesome contraptions. At the advanced level, we'll go over what binary really is, and how to design binary logic machines with Redstone!

Table of Contents

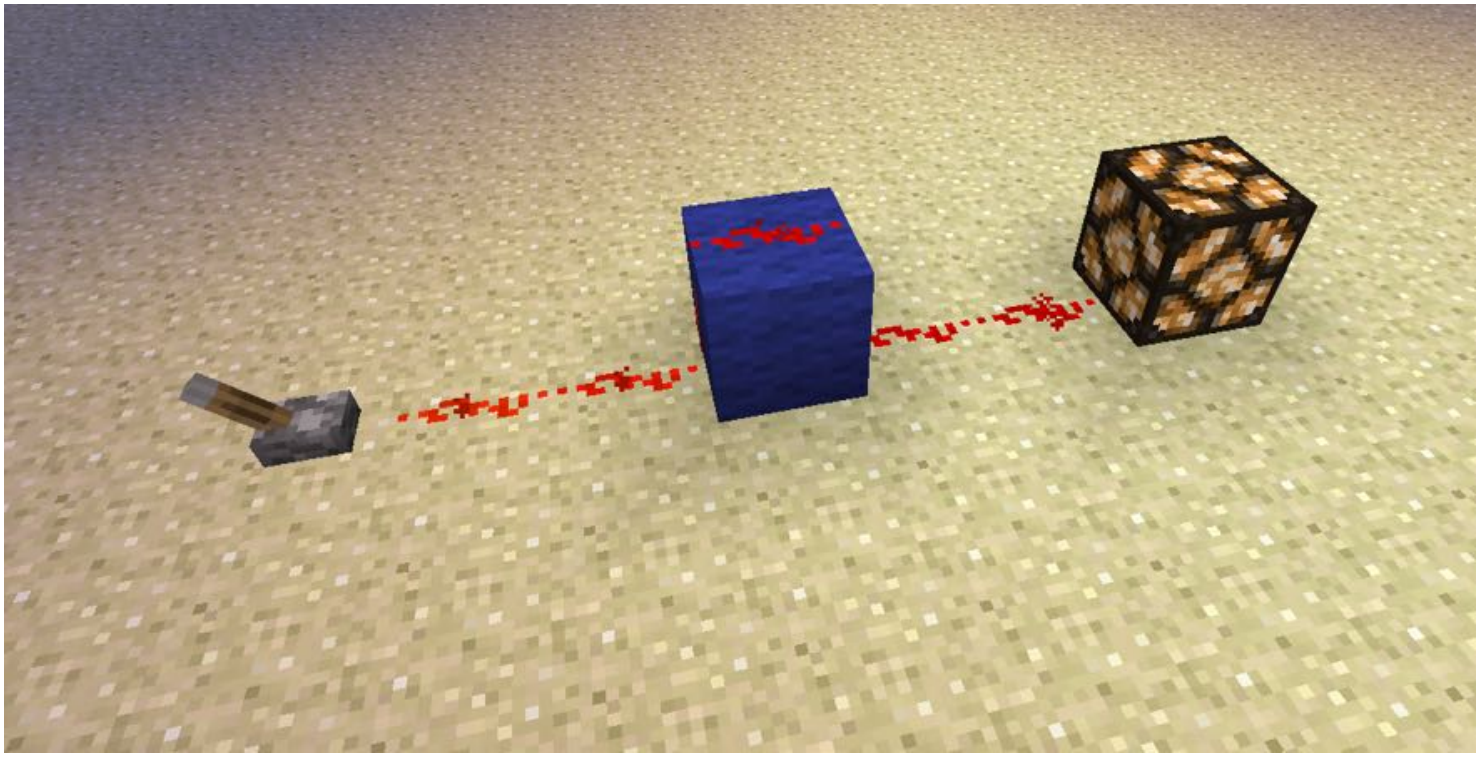
Section 1: Redstone Basics
Section 2: Basic Applications (Doors, Clocks, Cannons)
Section 3: Logic Gates
Section 4: Building with Logic Gates

Section 1: Redstone Basics

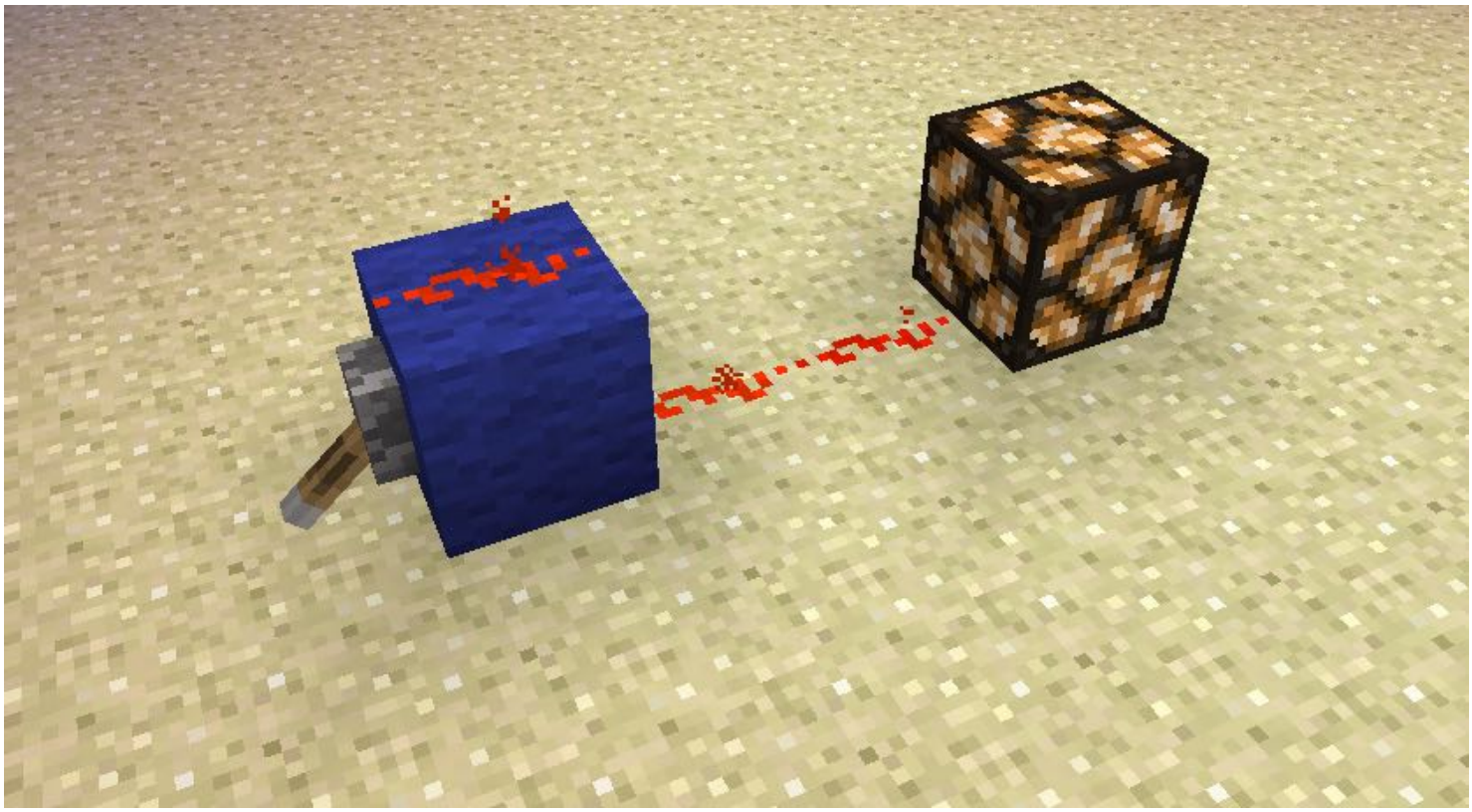
Redstone in Minecraft is a bit like electricity in real life. A redstone source transmits power through redstone dust (or repeaters) and can be used to power other blocks. For example, a lever in the ON position can be used to power a redstone lamp if at least two redstone dusts are placed between the two. THe two



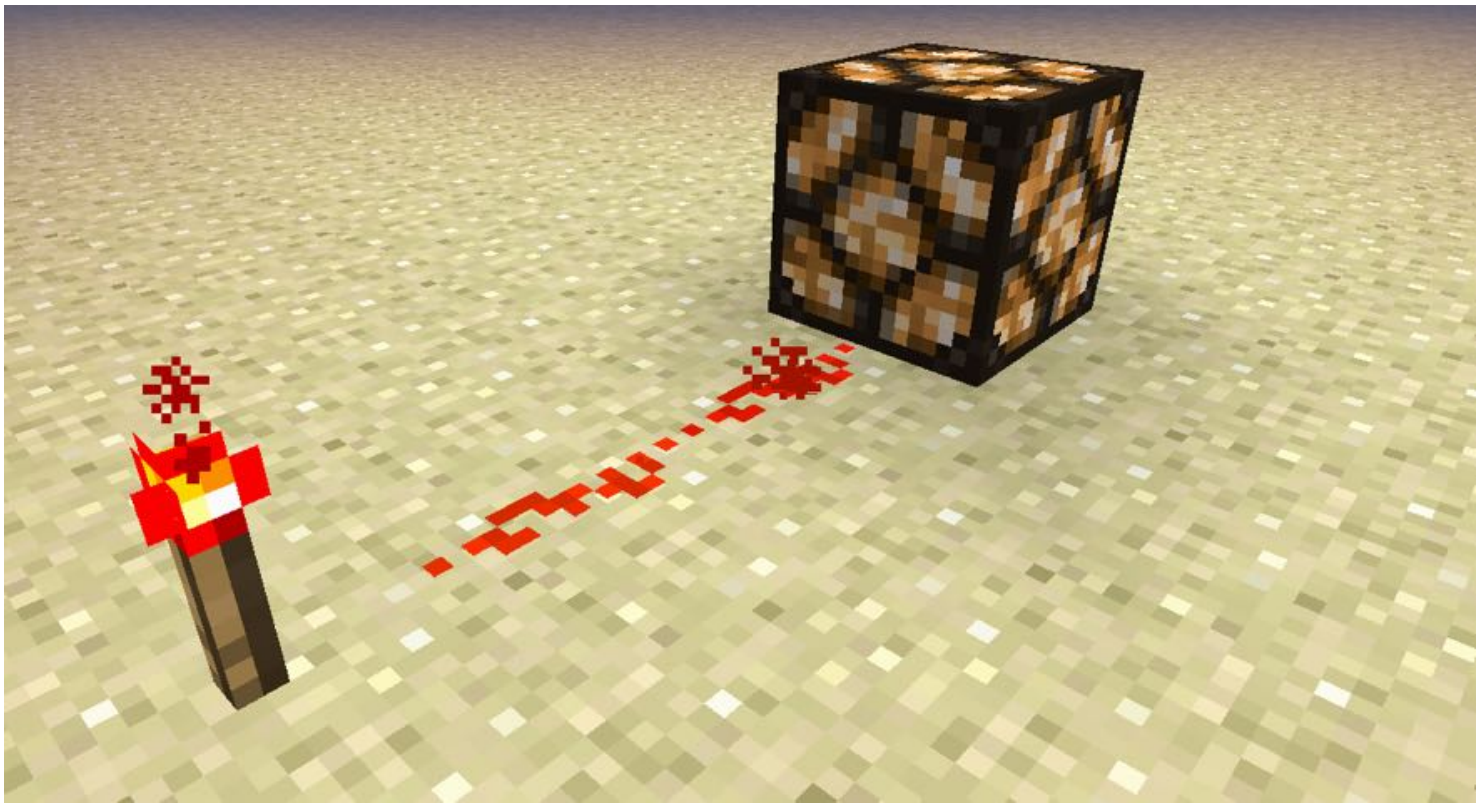
Redstone dust (also called redstone wire) can also bridge over individual blocks and still transmit power (also called redstone current).



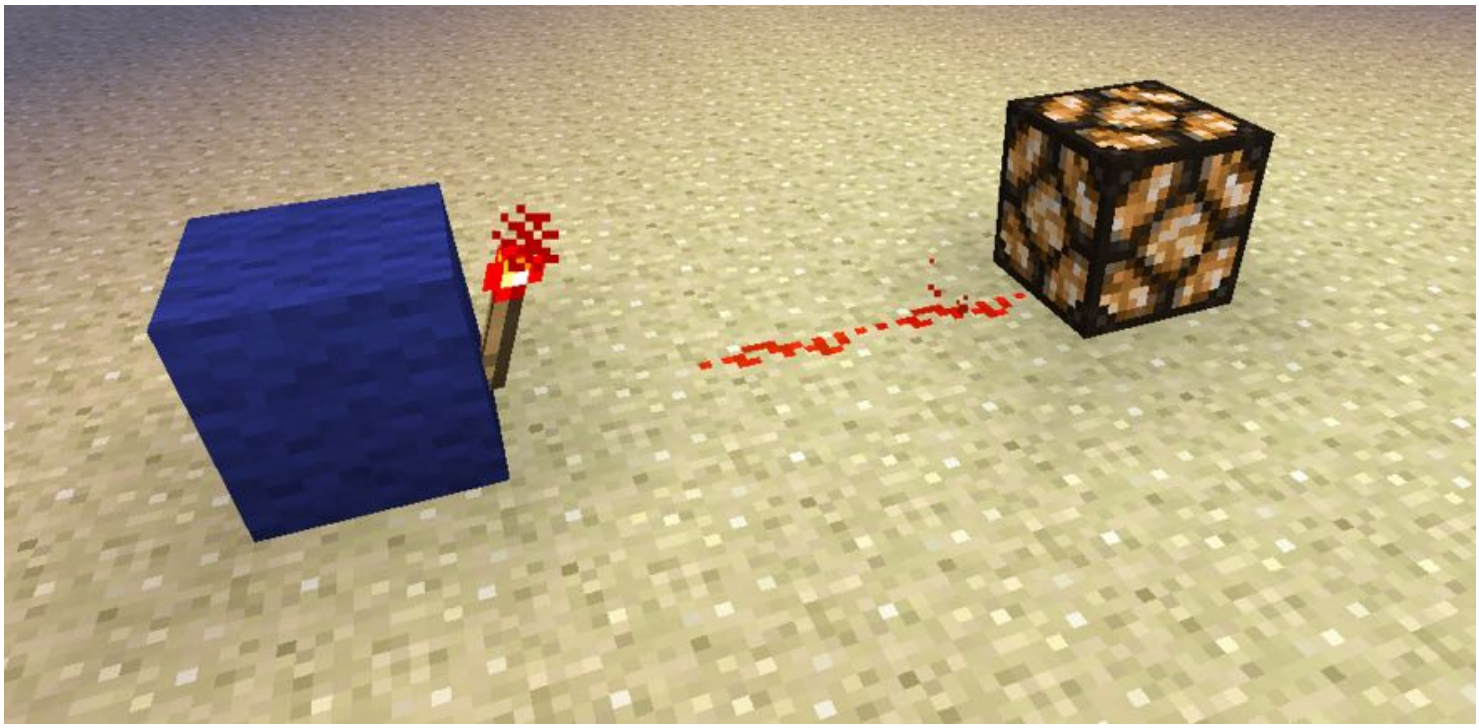
A lever placed directly onto a block powers the block itself, and redstone wire can carry current from the block to a lamp.



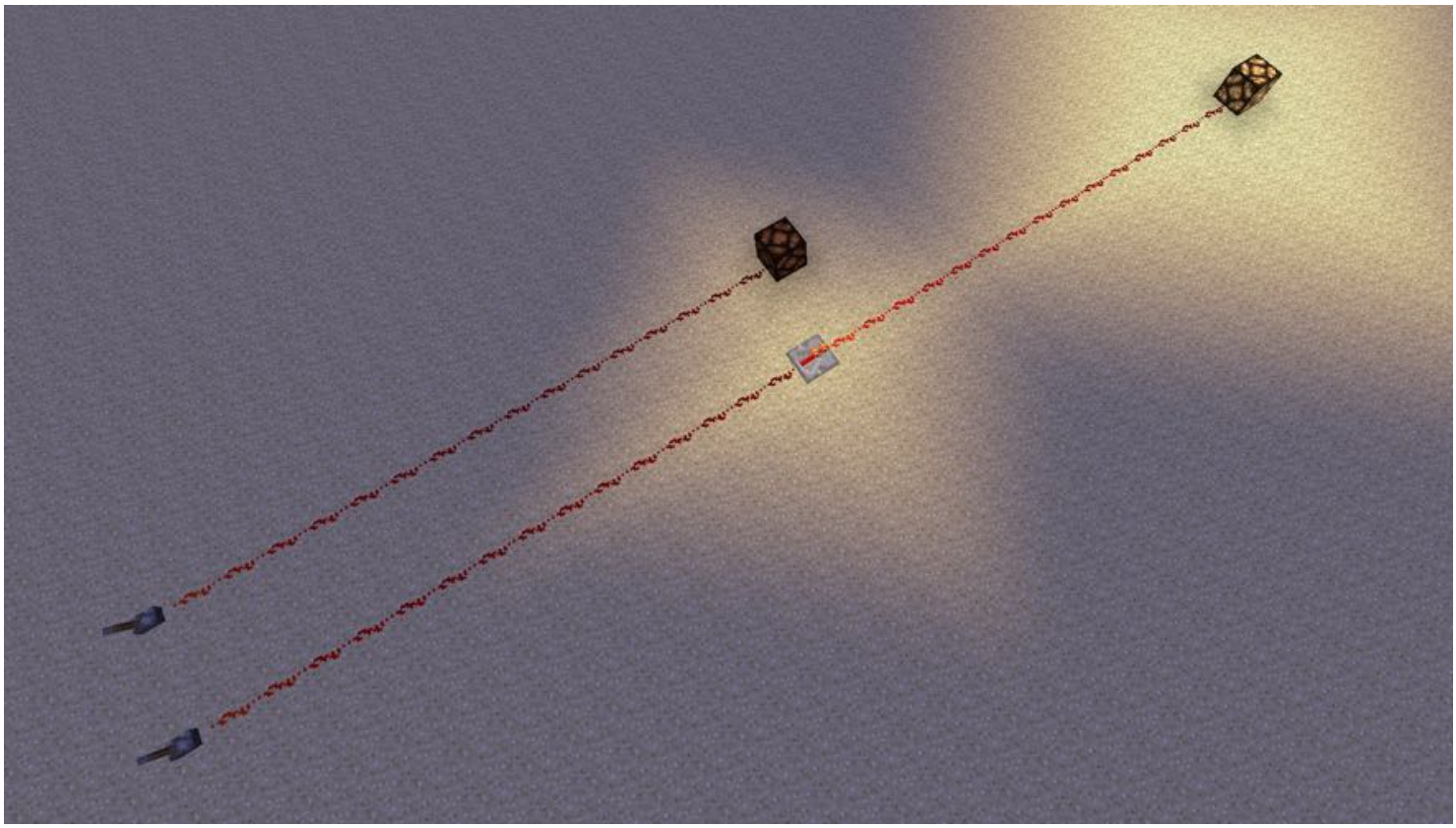
Redstone torches are a permanent source of redstone power; they are always ON and will always power adjacent redstone wire.



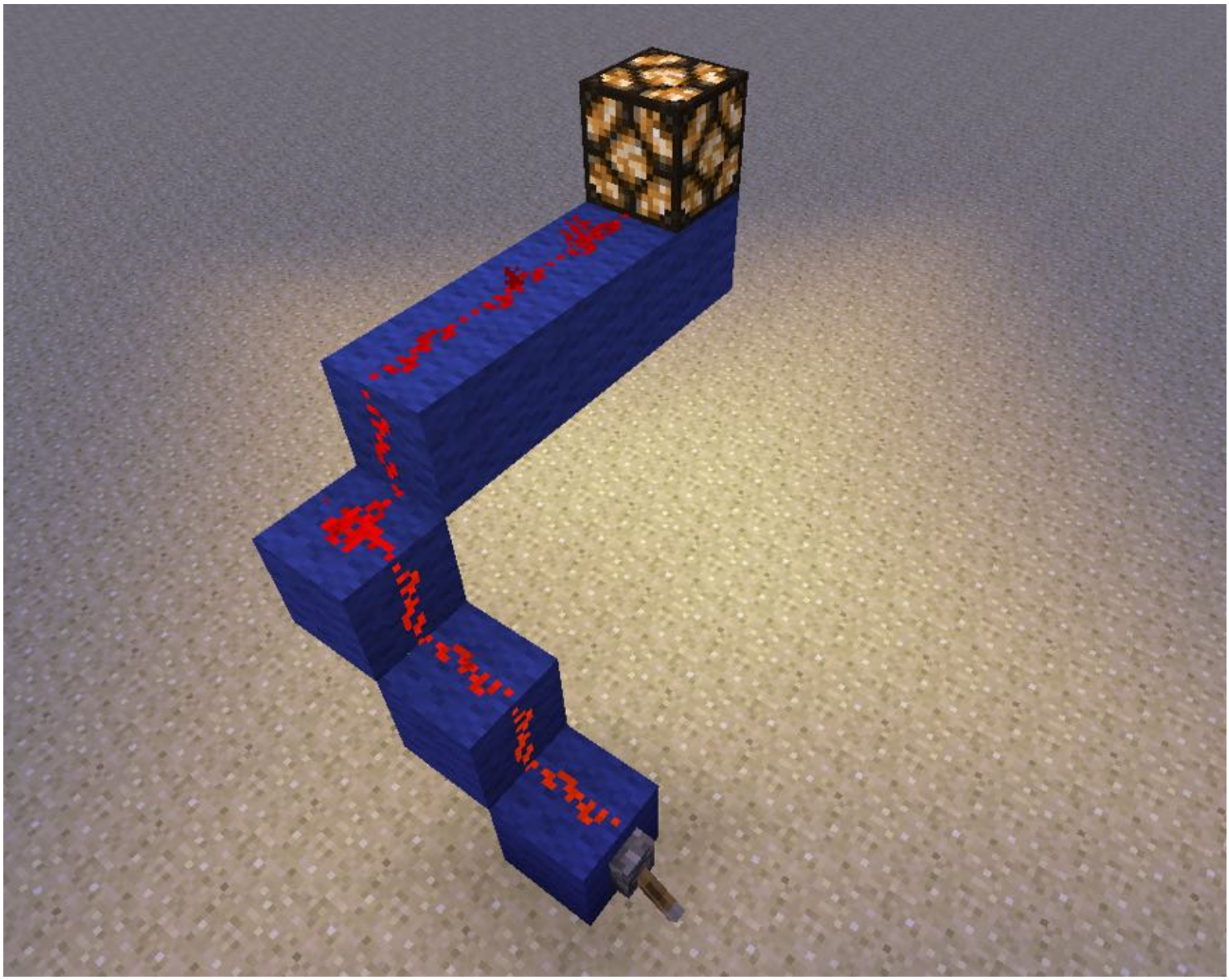
A redstone torch can be placed on the side of a block, too. Any adjacent redstone dust will also be powered.



Redstone current dies out after 15 blocks of redstone dust. Placing a repeater (which *repeats* the signal) will let a power source reach blocks even further away.

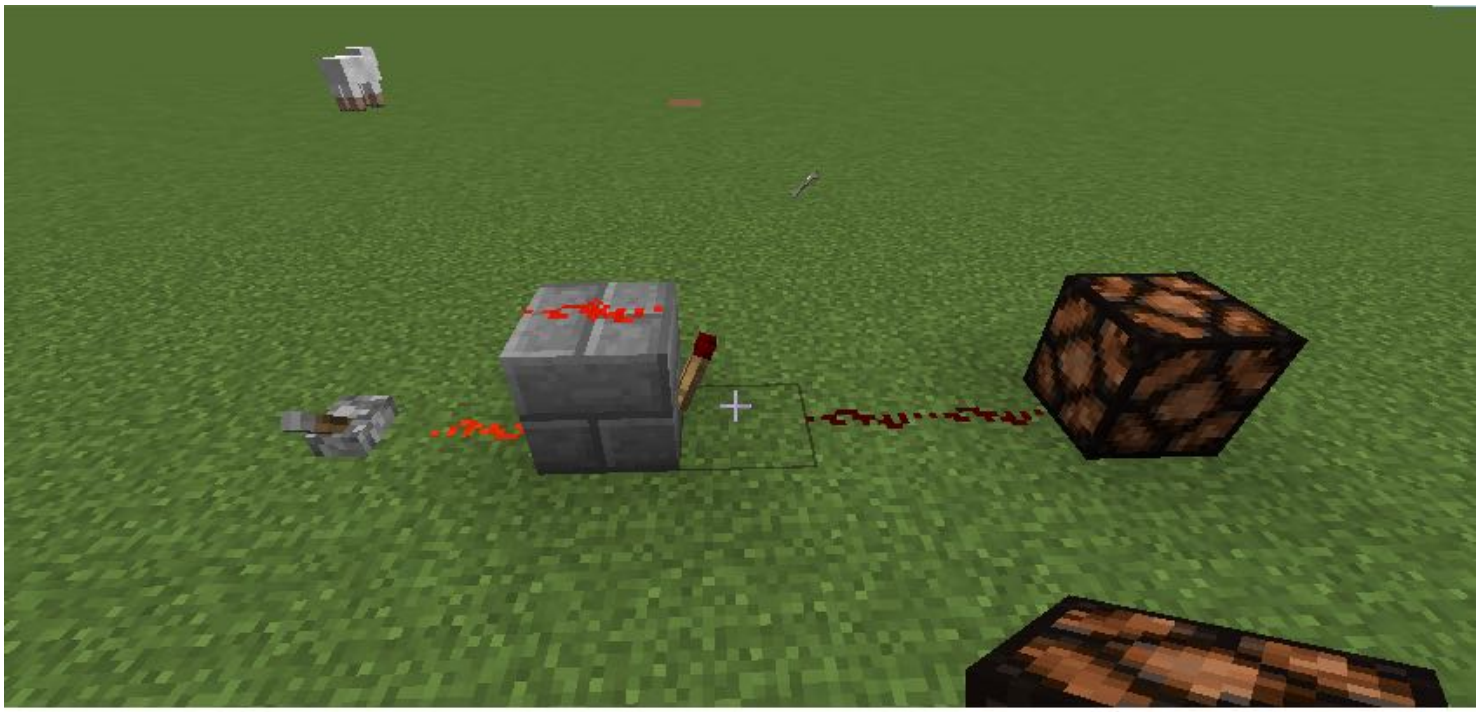


To move current upward, you must make stairs and connect the redstone dust going upward.



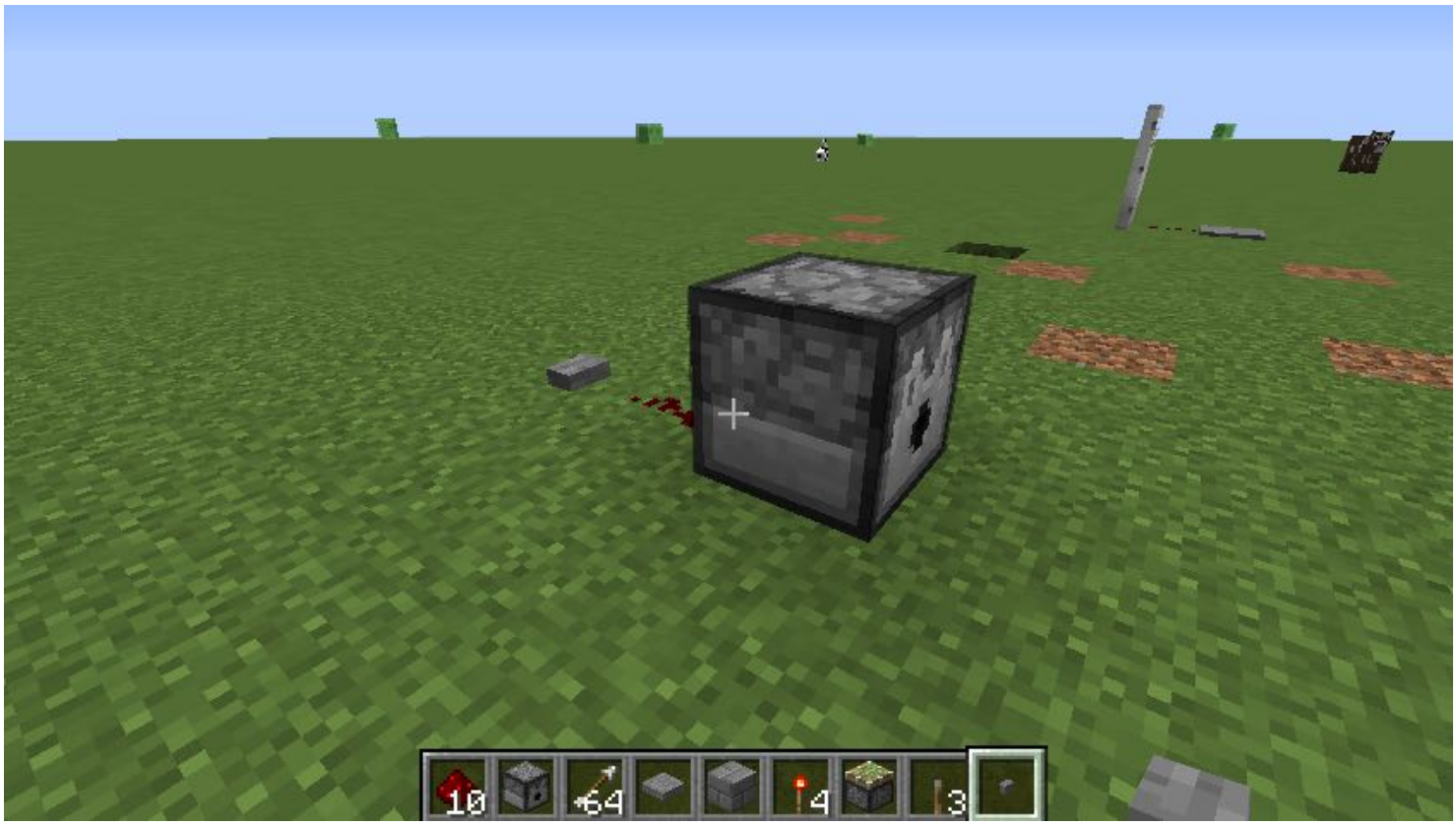
Now that you've seen some of the basics of redstone, experiment more with interactions between redstone and blocks. For example, you could try placing repeaters next to blocks and see how the current behaves.

Redstone torches are "turned off" when the block attached to it is turned on. In the setup below, when the lever is switched on, the redstone power travels to the stone block that the redstone torch is attached to. Because the block the redstone torch is attached to is now powered, the torch is turned off.



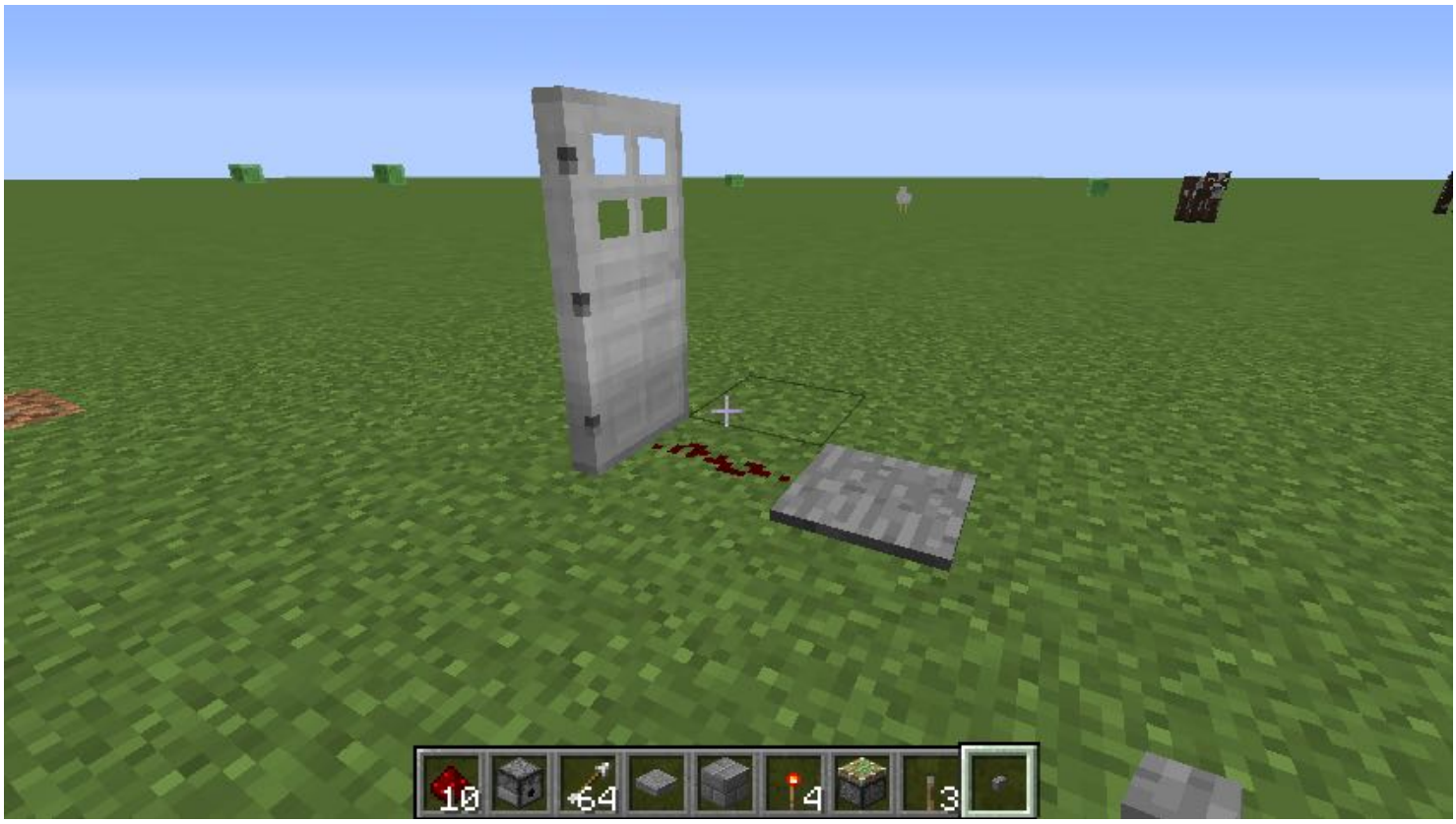
Redstone can also be used to power devices such as the ones shown below:

- Dispenser powered by button. Dispenser may be filled with any material (e.g. Arrows, diamond, etc).

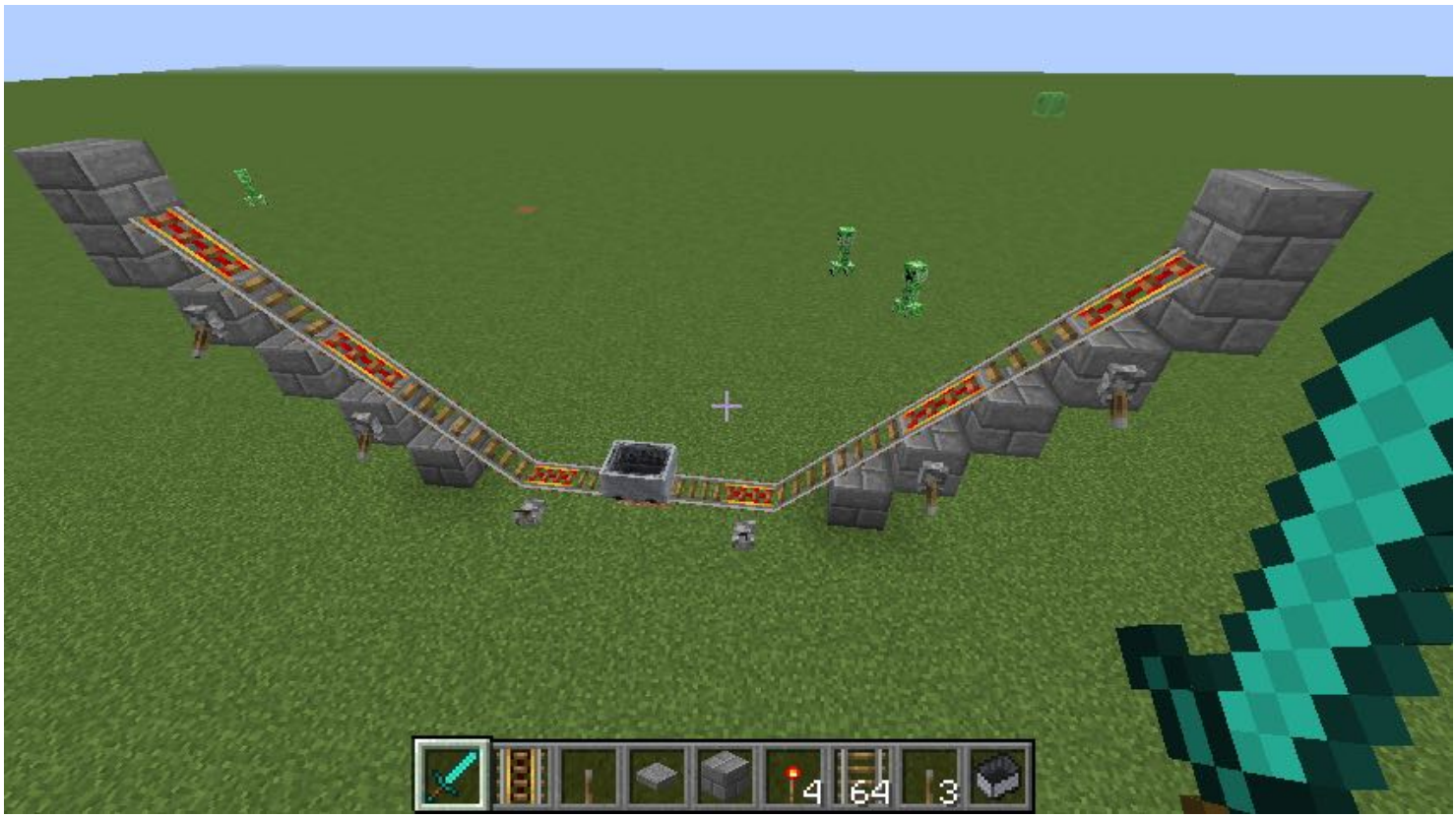


- Sticky piston powered by lever.

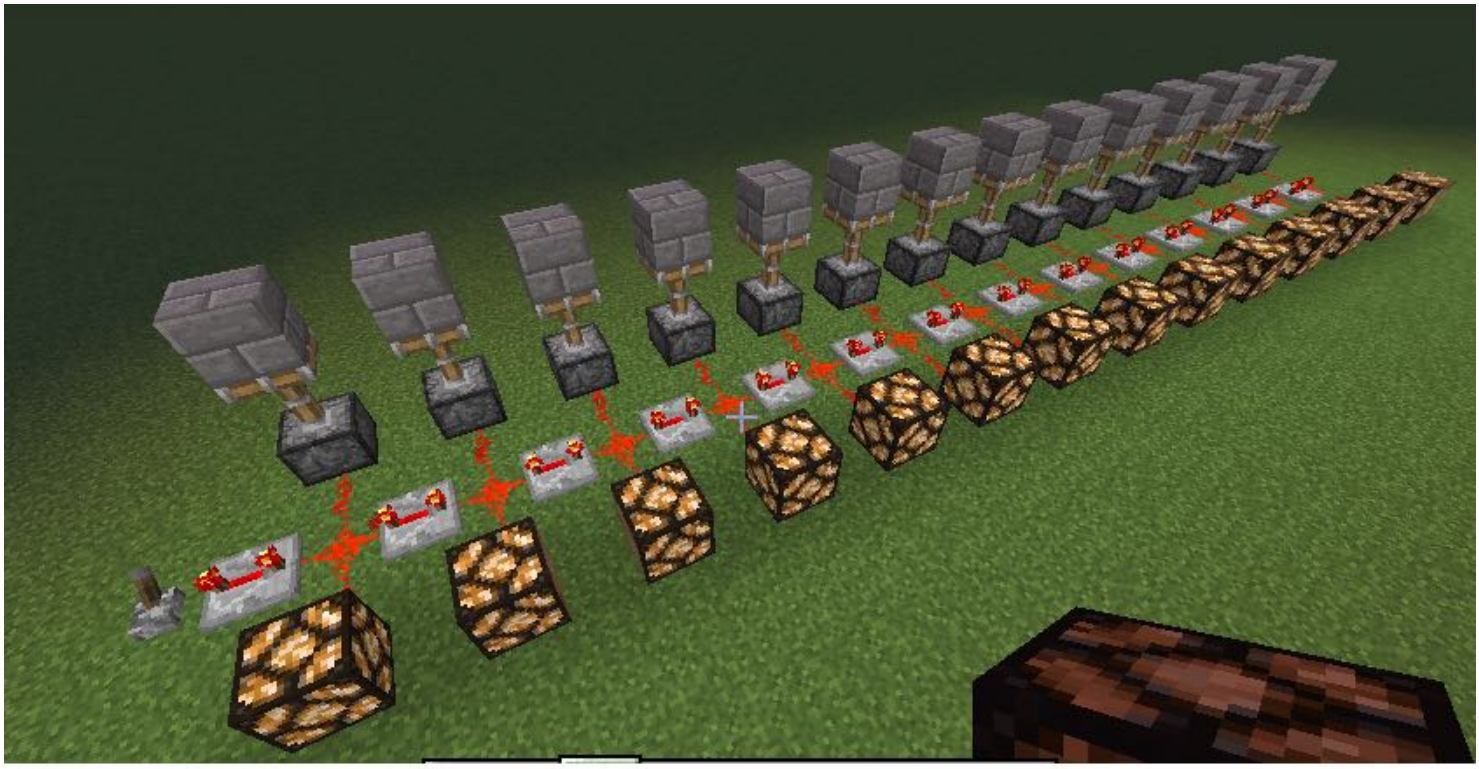
- Door powered by pressure plate.



- Powered rails powered by levers.



You can also combine these elements to make more complex contraptions.

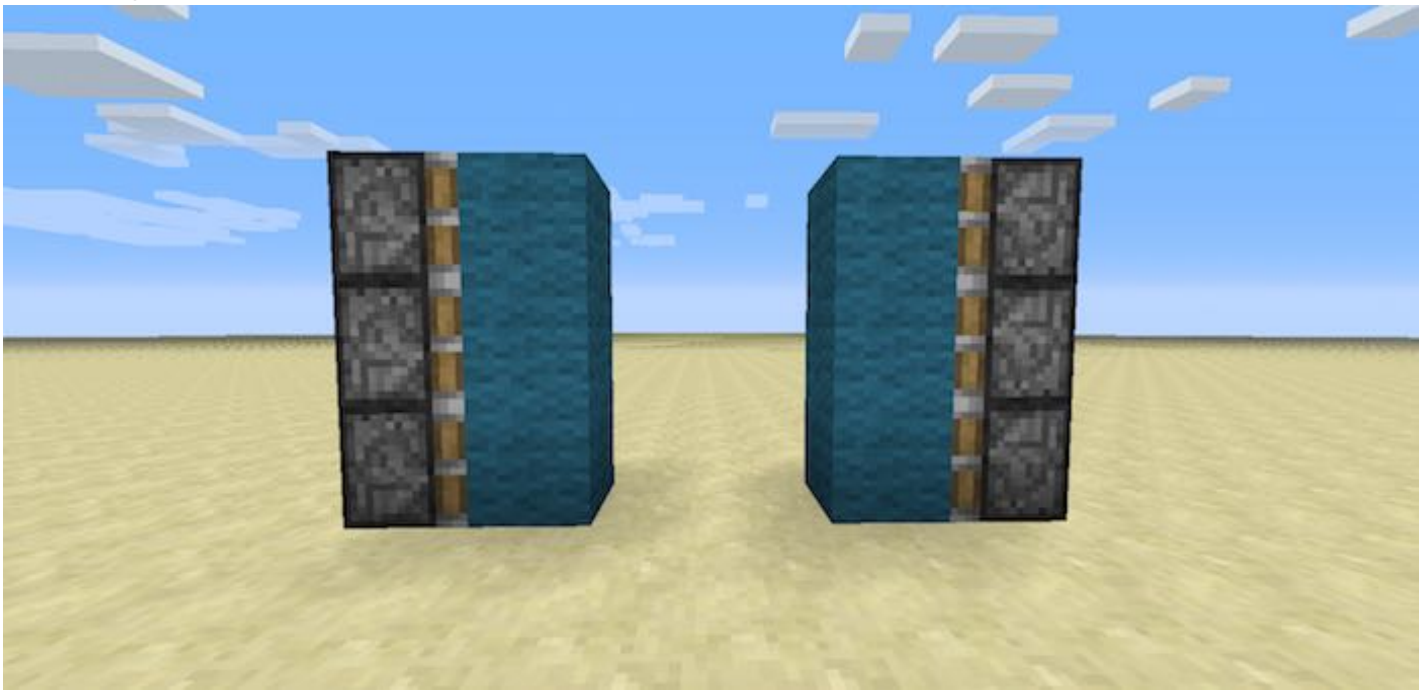


Section 2: Applications of Redstone Basics

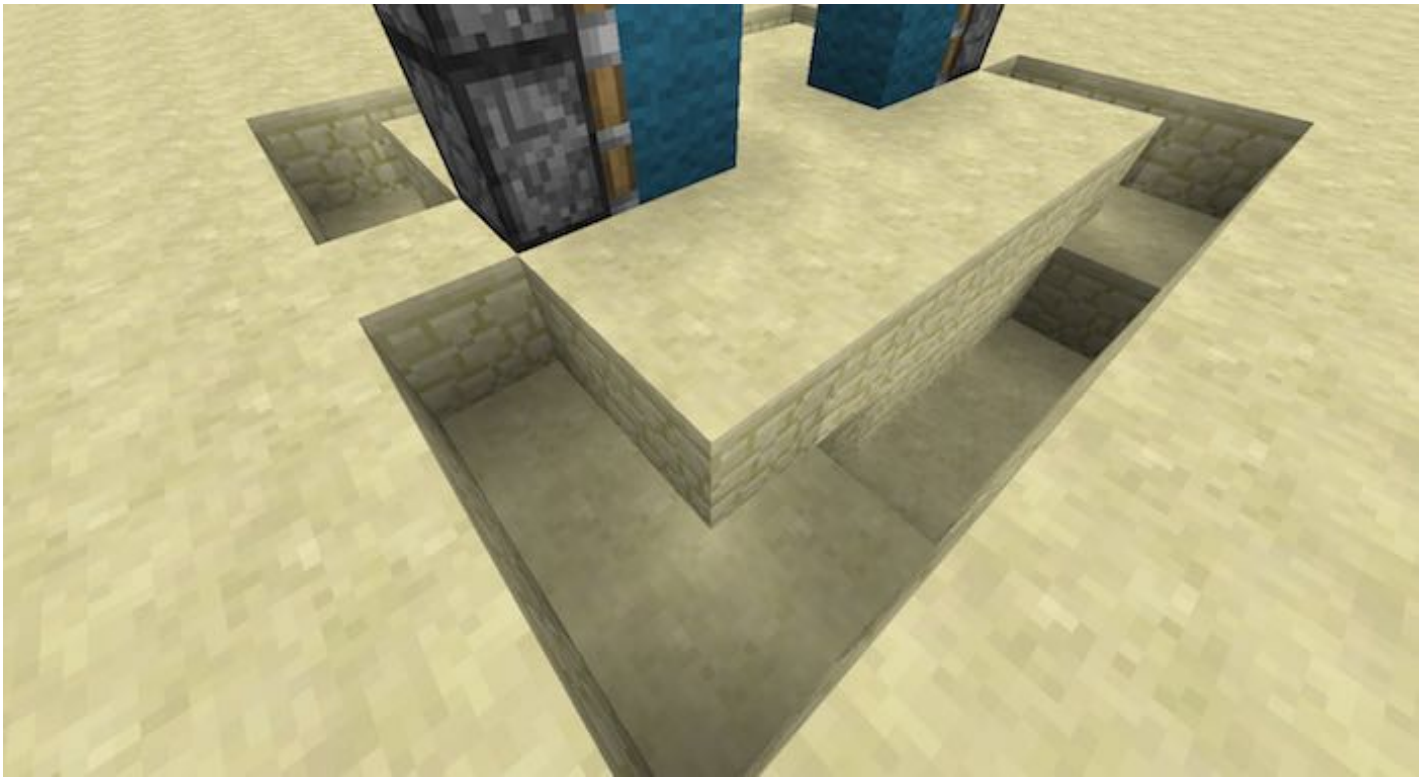
Piston Doors

This section concerns making doors powered by pistons and redstone. Rather than using wood or iron doors, we'll be using pistons to make large doors using blocks. For this you will need redstone, redstone torches, and sticky pistons. Since sticky pistons can both push and pull blocks, they're ideal for making something like a door. The redstone circuit used is simple but still requires some space, which can be hidden inside a wall (especially since doors normally appear in walls!).

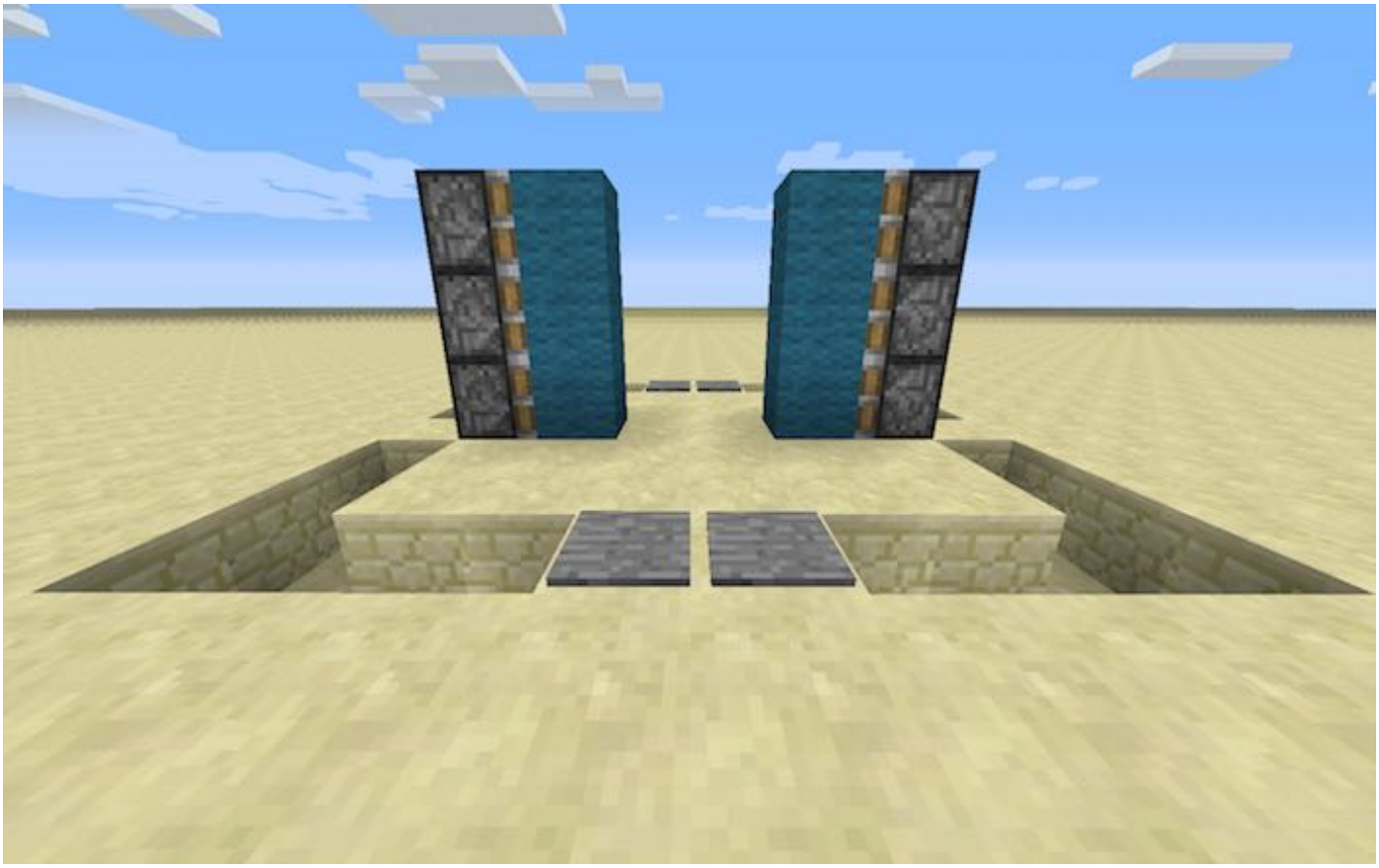
1. The door will be two blocks wide and three blocks tall. Place two columns of three sticky pistons facing each other with four empty air blocks between them. Then place your door blocks (lapiz, gold, etc) onto each of the sticky piston faces. When unpowered, there will be a two-block gap between the pieces of the door.



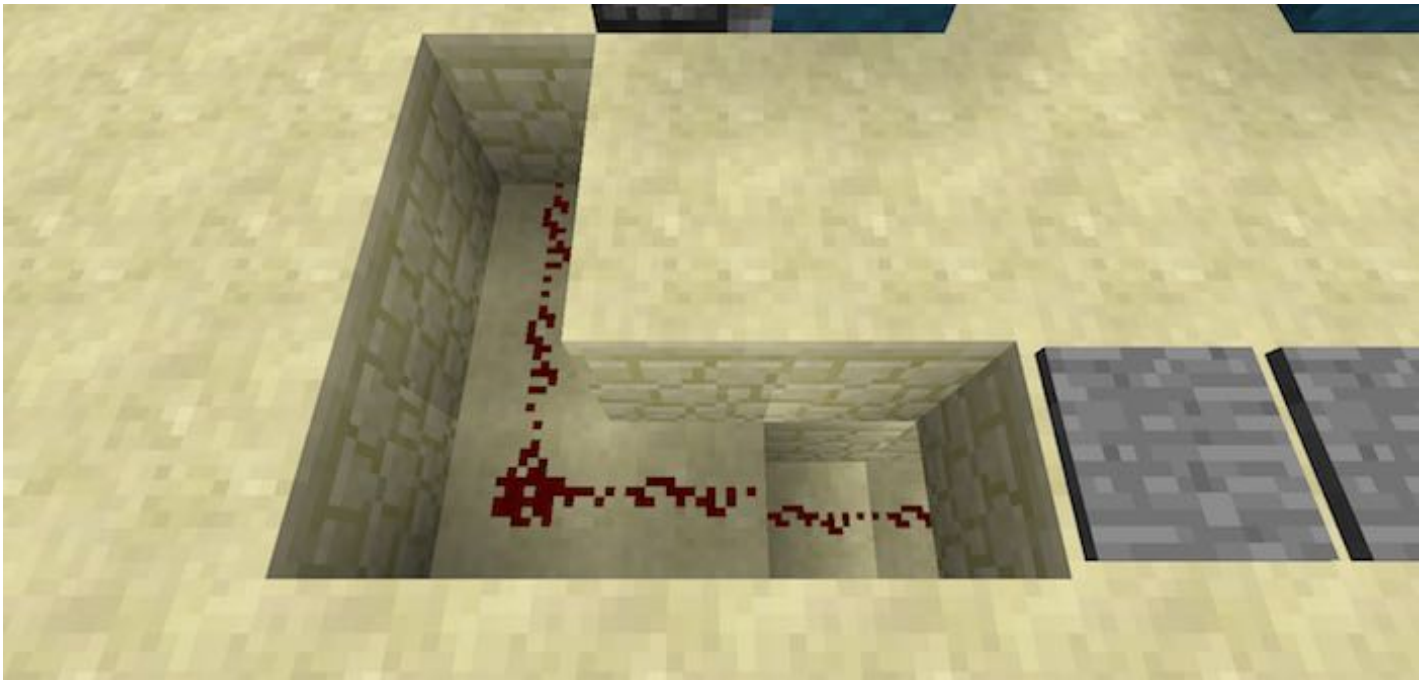
2. Dig a one block deep channel around the doors as shown in this picture. We will hide the redstone we're using to power the doors within this channel. Dig out two more blocks from the front of the door (consult the screenshot if you are confused).



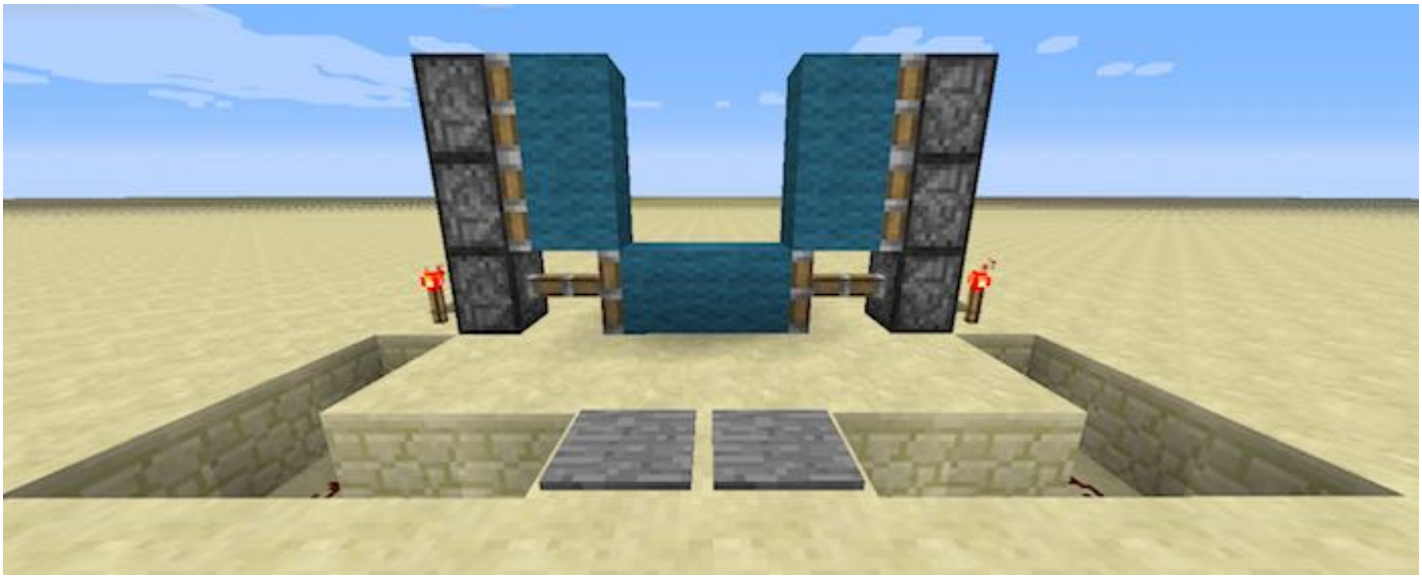
3. Then put two ground blocks back in the middle (keeping the just-mined block underneath empty) and put two pressure plates on top of them.



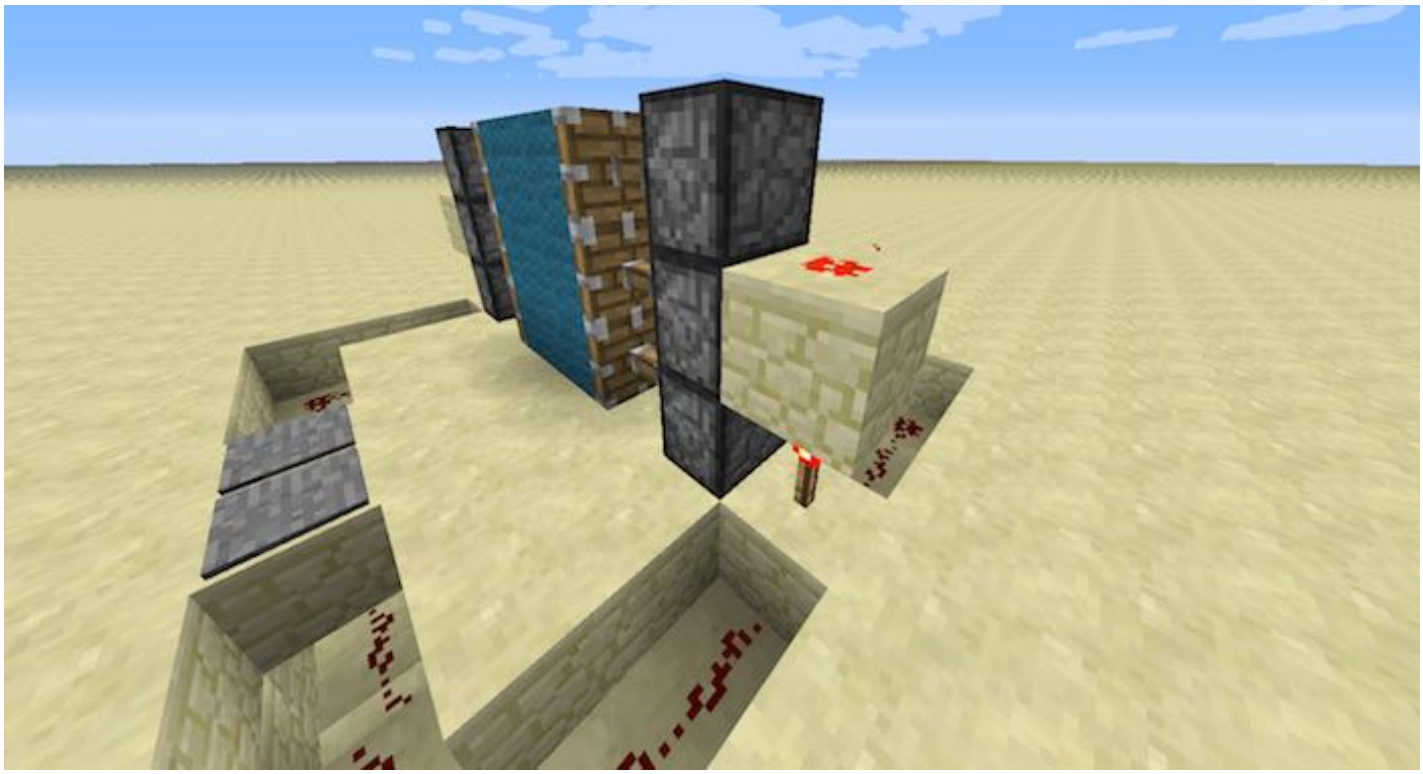
4. Now run redstone from the very bottom space all the way around the bottom of the channel, with the ends pointing into the remaining sandstone block on either side.



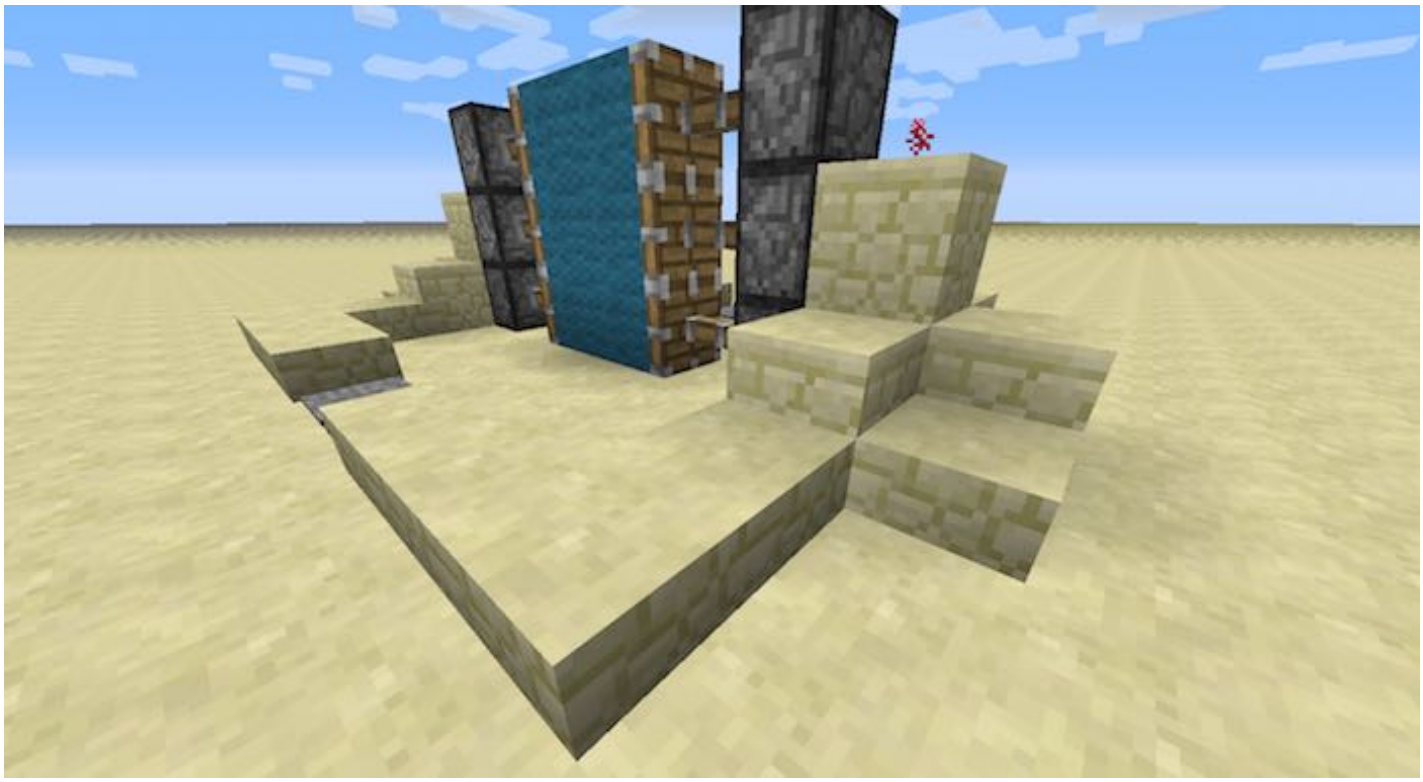
5. Place a redstone torch on top of that remaining sandstone (the torch should be next to the bottom-most sticky piston).



Place a sandstone block on top of that torch and redstone dust on top of that sandstone block. Repeat this for the other side of the door.



6. Cover up the redstone using slabs, which give the best aesthetics. You can also cover up the side towers (pistons and all) with sandstone, wood, or other attractive patterns.

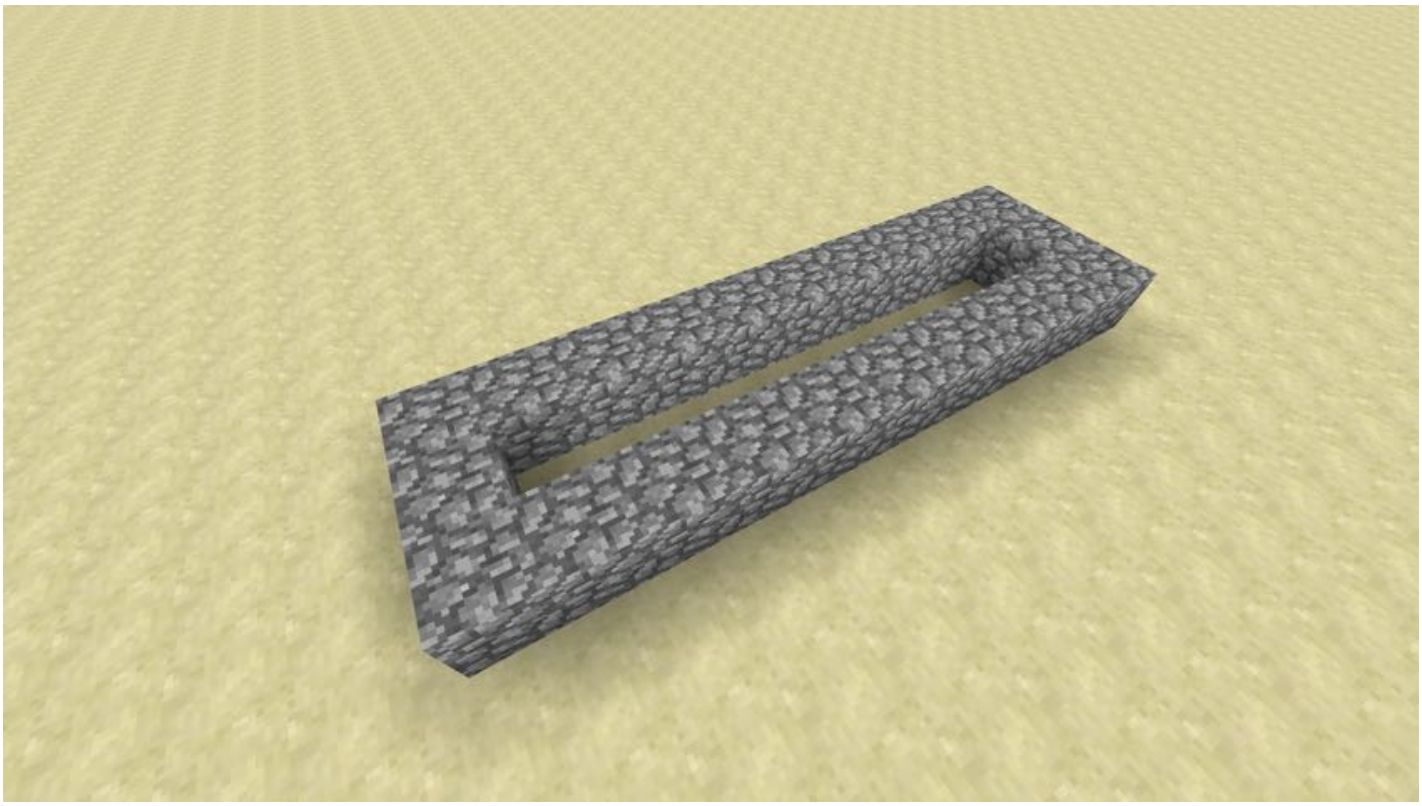


Ultimately, only the two-block space in the middle of the door is what shouldn't be covered. Our suggestion is to build this door as part of a wall where it looks best.

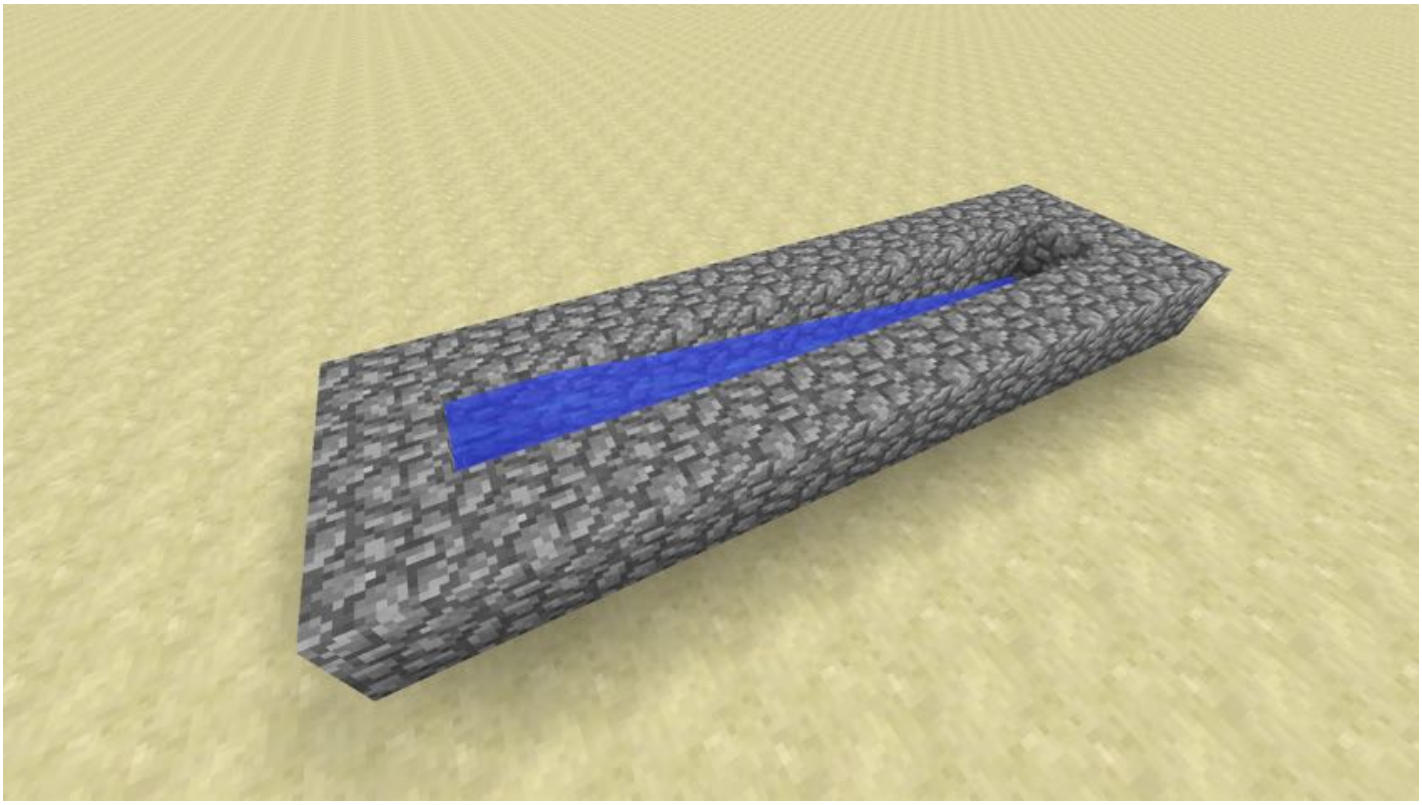
Cannons

By exploiting some of the physics of Minecraft, we can make cannons that fire TNT! As you've probably seen, explosions normally destroy blocks and throw items. Explosions in water, however, will not destroy blocks but will still affect items. Lit TNT is one item that can be thrown, and so we can use TNT and water to shoot a lit TNT block out of our cannon. We will be using redstone to ignite our TNT.

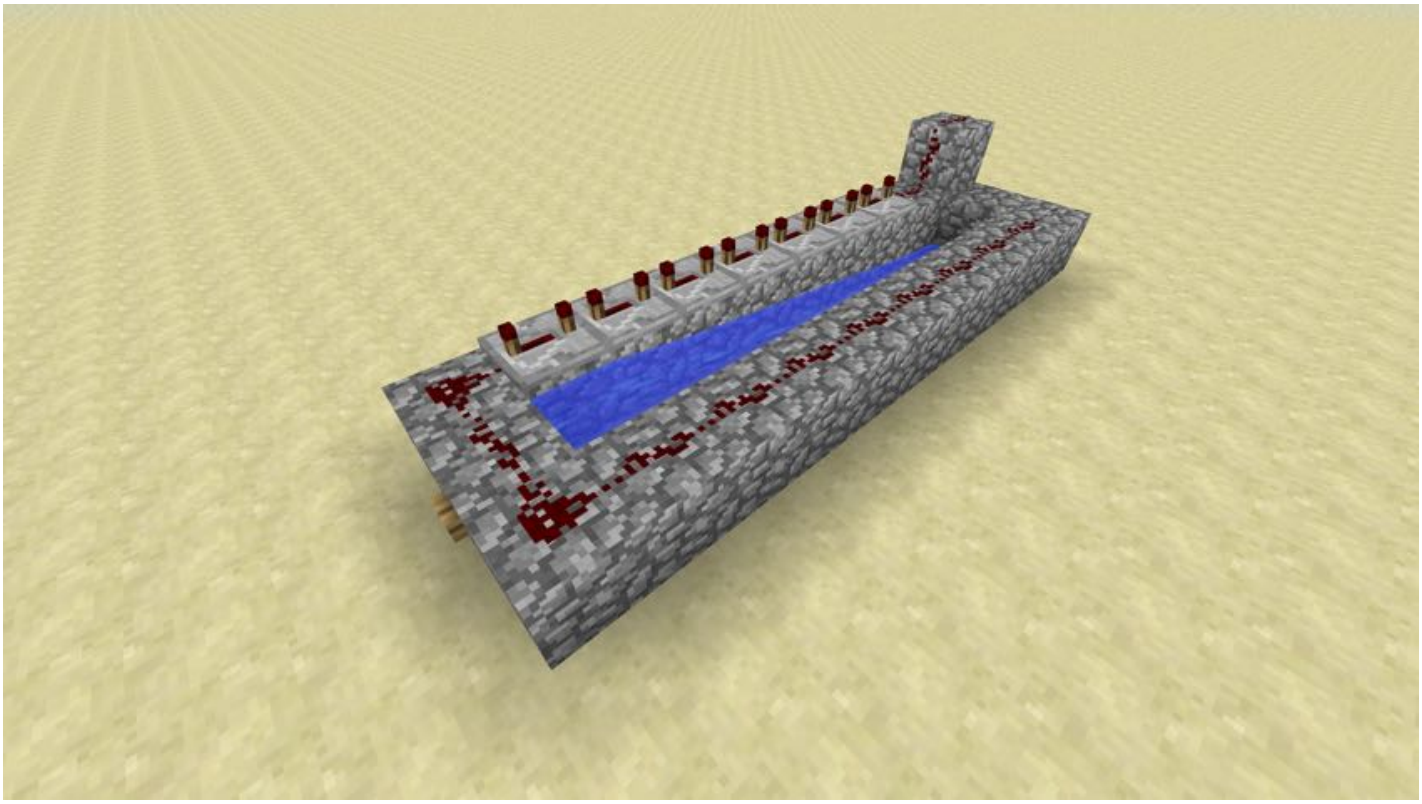
1. Set up the body of the cannon. Cobblestone is what we will be using, though most normal building blocks will work.

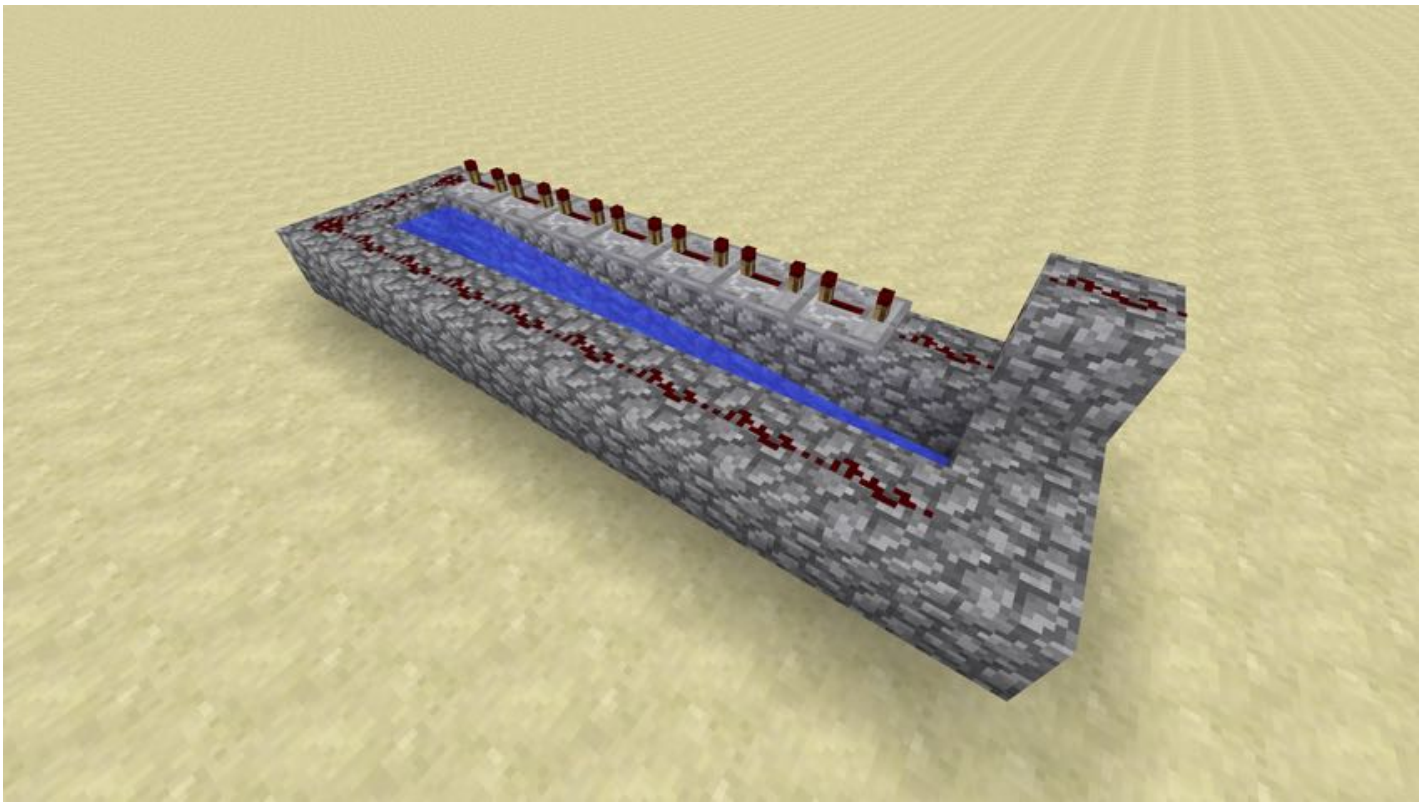


2. Fill the body with water, which will protect the body of the cannon from the blast.

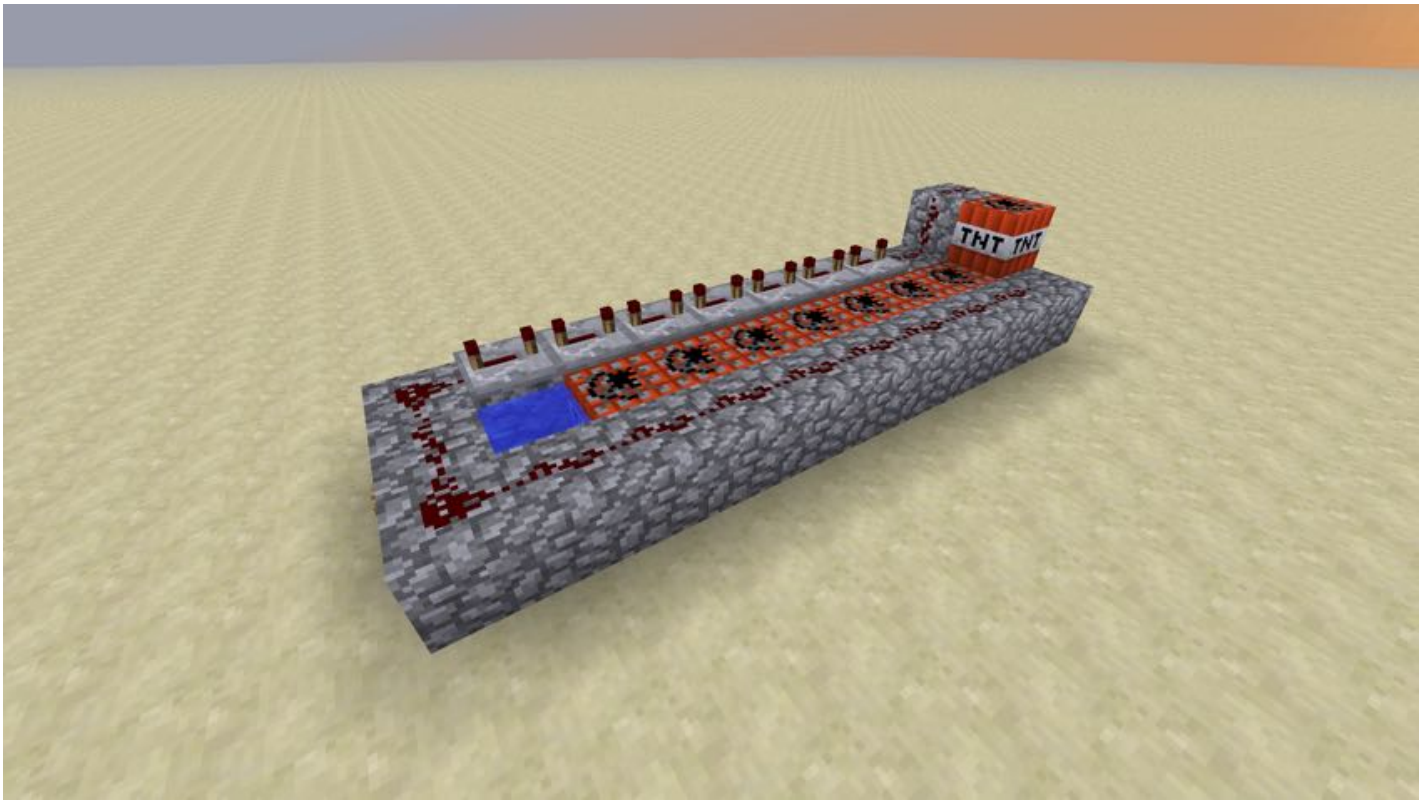


3. Place our redstone and repeaters as well as a button to trigger them. The trail of redstone will ignite our propellant TNT. The repeaters will ensure that the projectile TNT is lit right before it fires.





4. To fire the cannon, we will fill the body with TNT. Be sure not to accidentally place TNT over the source block of water. Then, place the projectile on the very end.



5. Press the button to fire!

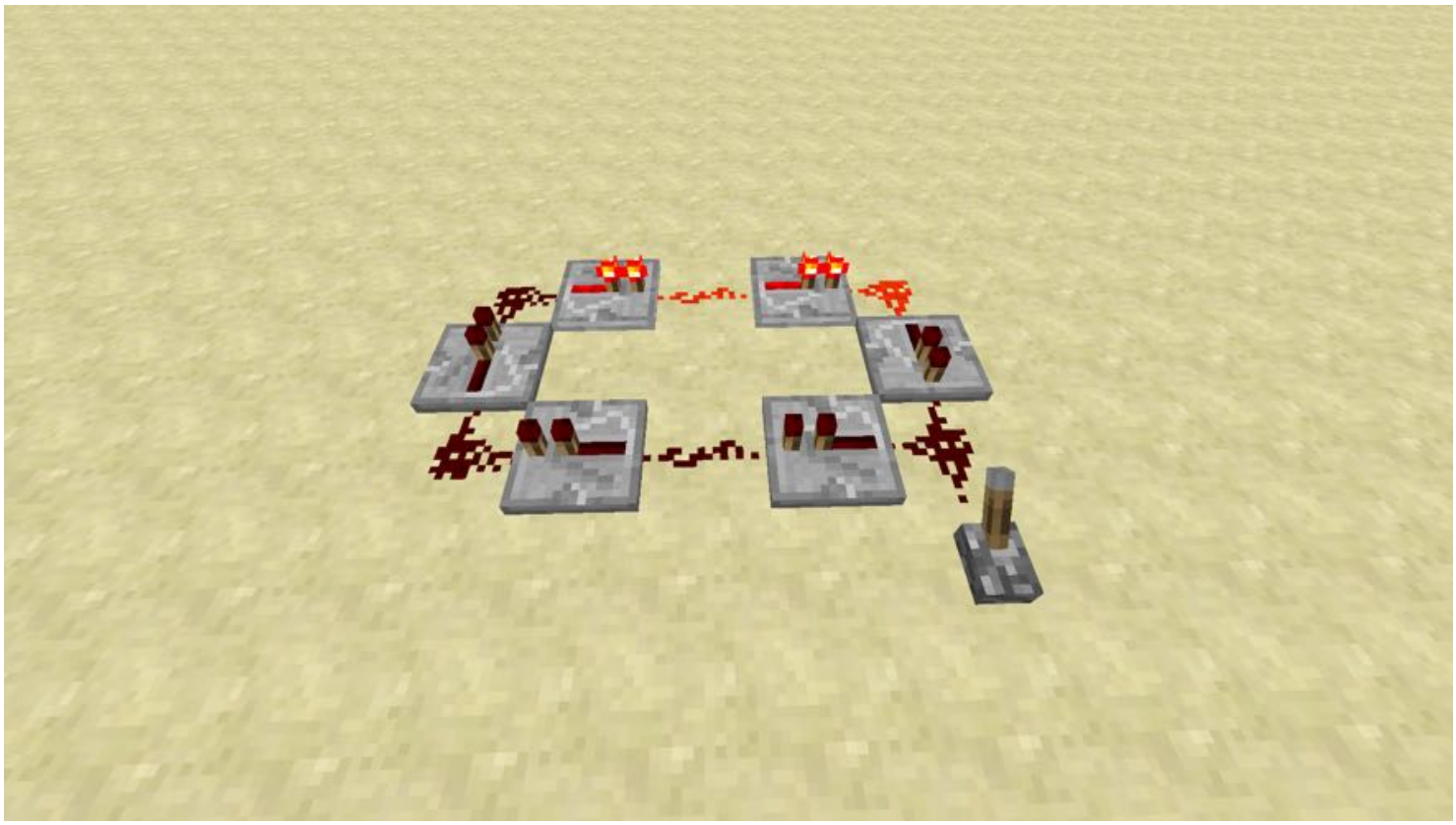
We know that we're limited to using TNT seven blocks away from whatever we want to fire. How could we increase our firing power with this limitation? Come up with some ideas and designs and test them out. Also, experiment with the height of the projectile TNT. How does placing it onto a slab change the distance shot?

TNT Cannon Competition

Try challenging your neighbor with a TNT cannon competition! Modify the example design and see who can fire the farthest and with the most accuracy.

Redstone Clocks

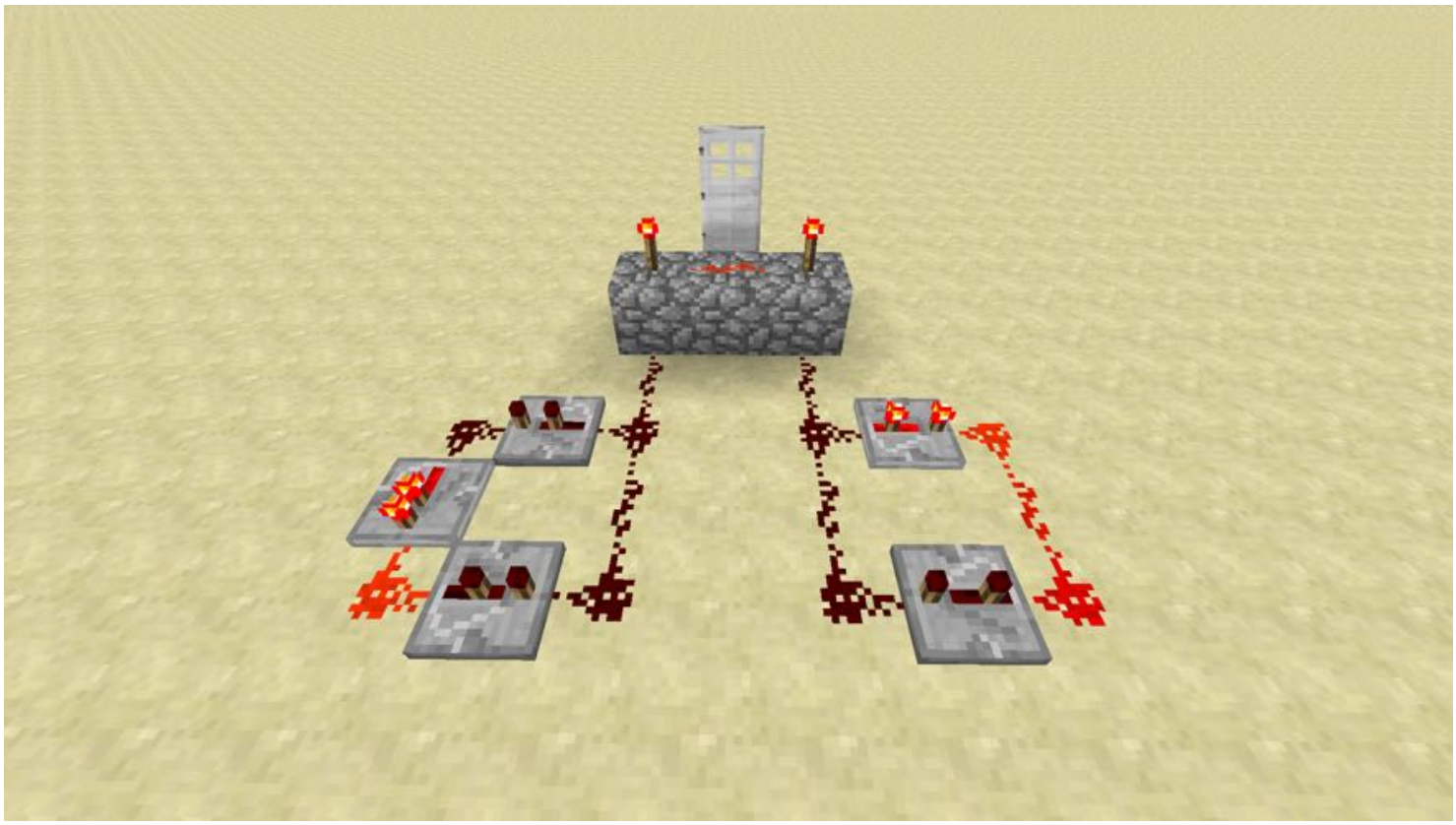
Interestingly enough, you can use redstone to make simple clocks in Minecraft. Redstone signals do not travel instantaneously. Instead, they update based on the timing of "ticks". A "tick" is one update in the world of Minecraft; for redstone, ten ticks occur per second. Generally torches, repeaters, and other blocks take one tick to update. Repeaters are unique in that they can be set to a specific number of ticks; the "setting" on a repeater (moving the torch back and forth along it) changes it to 1, 2, 3, or 4 ticks. So repeaters can be used to make clocks that pulse at a regular rate.



You can make more complicated clocks by using a simple mathematical principle and something called a logic gate. Let's say we wanted to make a 30-tick clock, which would pulse every 3 seconds. To use our above design we'd need 8 total repeaters (7 of 4 ticks and 1 of 2 ticks). However, we can actually get away with using just 4 repeaters. To do this, we're going to make two separate blocks of length 5 and 6. Individually, these clocks only require two repeaters each.

Now, think of the tick numbers that these clocks will pulse on. The first one will pulse at 5, 10, 15, 20, 25, and 30. The second one will pulse at 6, 12, 18, 24, and 30. Both of them will pulse at 30! Since 5 and 6 are both factors of 30, the two clocks will always pulse on multiples of 30. Thirty is the least common multiple (LCM) of the two numbers, and we can use the LCM to make clocks out of far fewer repeaters than with our simple design.

Next, we will use a logic gate called an AND gate. Basically, it's a series of blocks that takes two redstone inputs. The output will only be ON if the two inputs are both ON. Otherwise, they'll be off. If we AND the two outputs of our 5 and 6 clocks, the output of that AND will pulse only every 30 ticks, or 3 seconds.

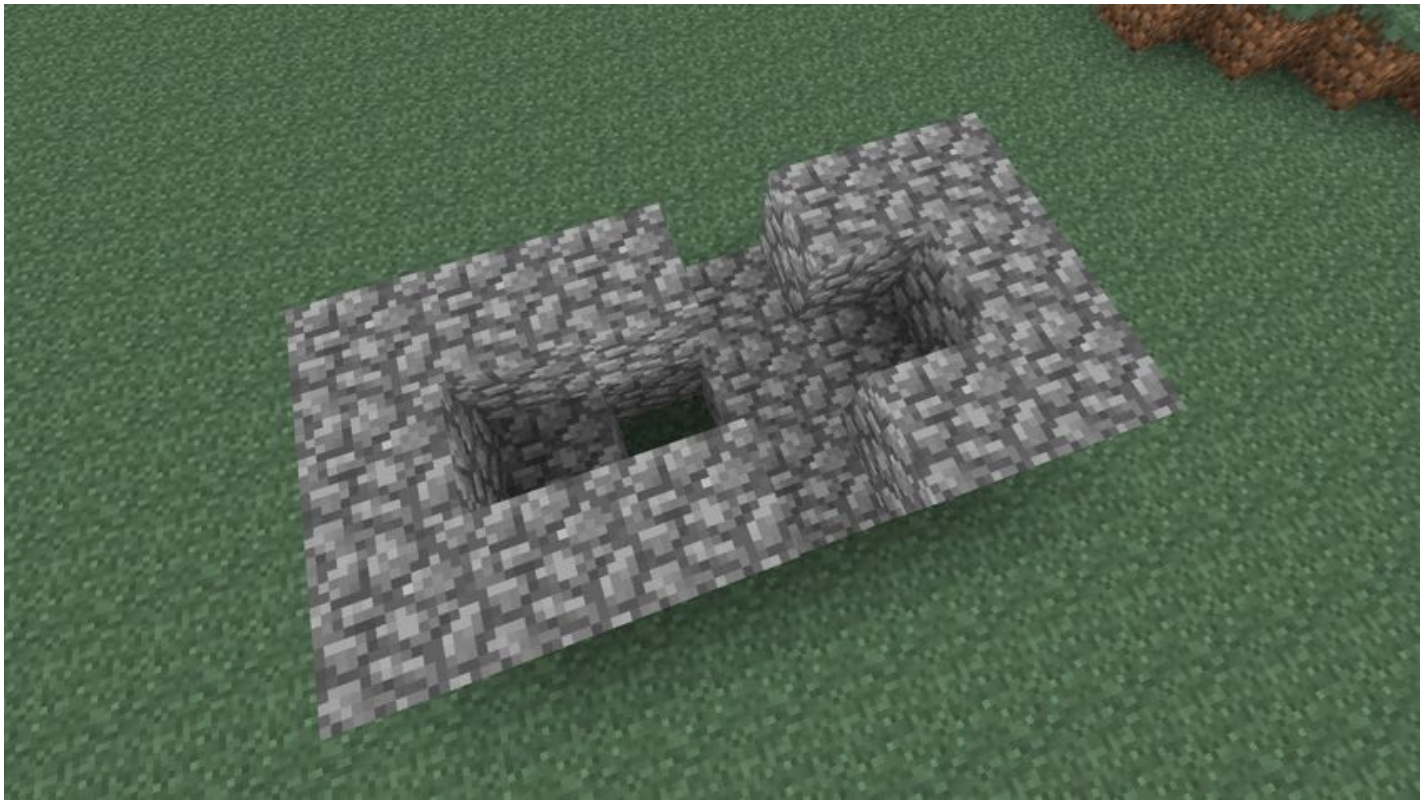


Redstone clocks have any number of uses. They can be used to make dispensers output something every set period of time, timers for playing games, or activate pistons for making cobblestone generators. Try to think of some productive uses for redstone clocks and implement them yourself!

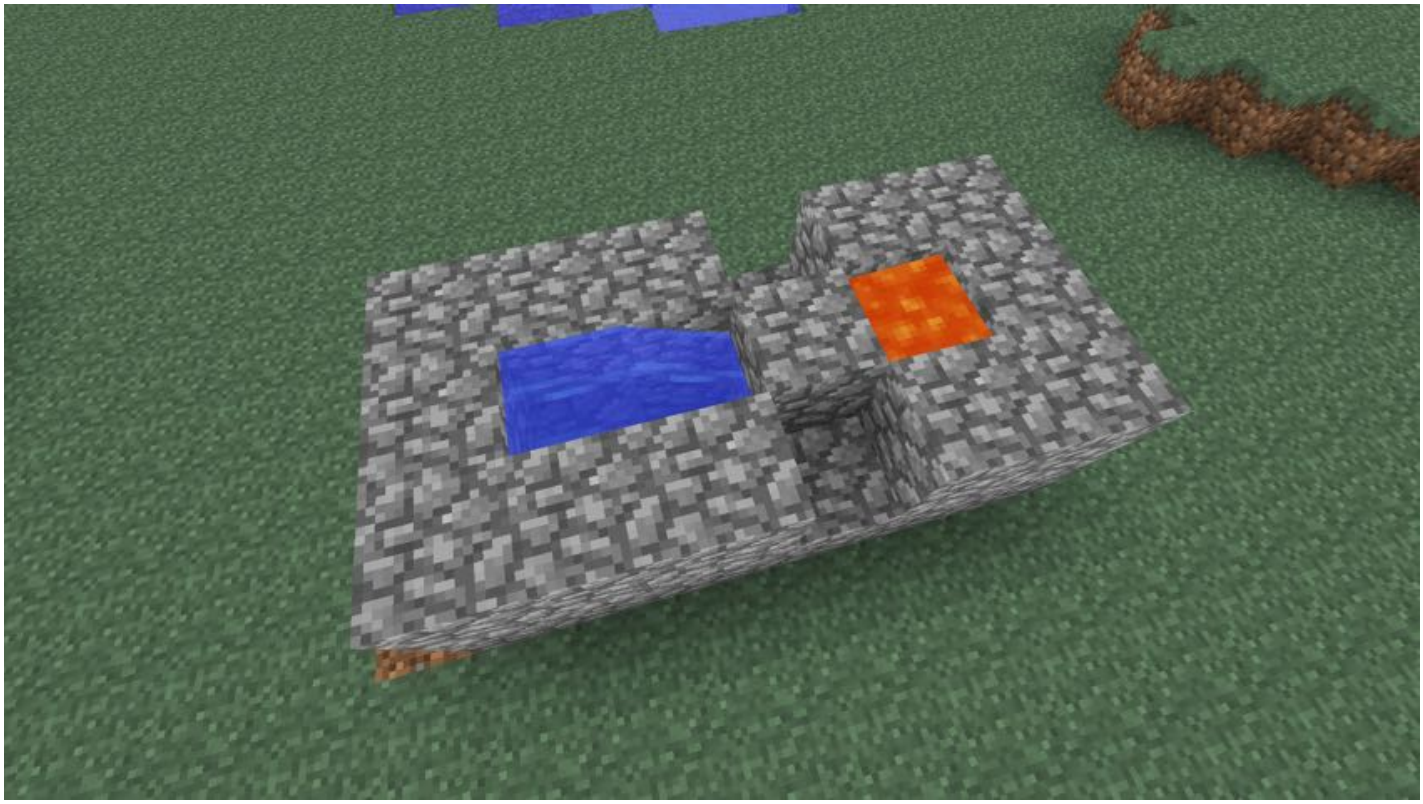
Cobblestone Generator

A cobblestone generator is both less costly and more useful than a TNT cannon. Cobblestone is created when flowing water and lava combine and is a valuable resource, especially on resource-light worlds. By controlling the flow of water and lava and utilizing a piston circuit, we can create a contraption that automatically builds cobblestone for us to harvest.

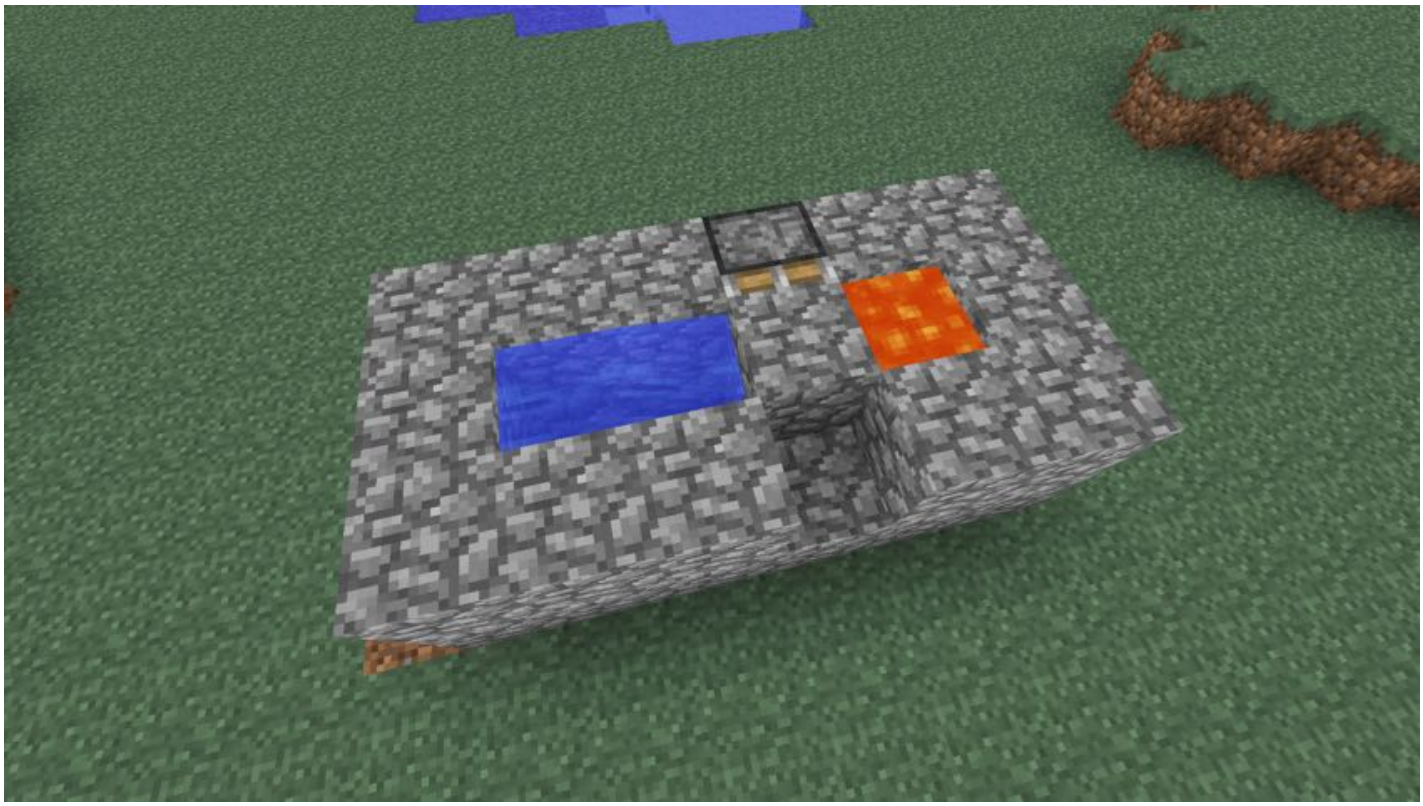
1. Build the body of the generator. Be sure to include the hole in the middle, as without the hole water would flow over the lava source block and turn it into obsidian.



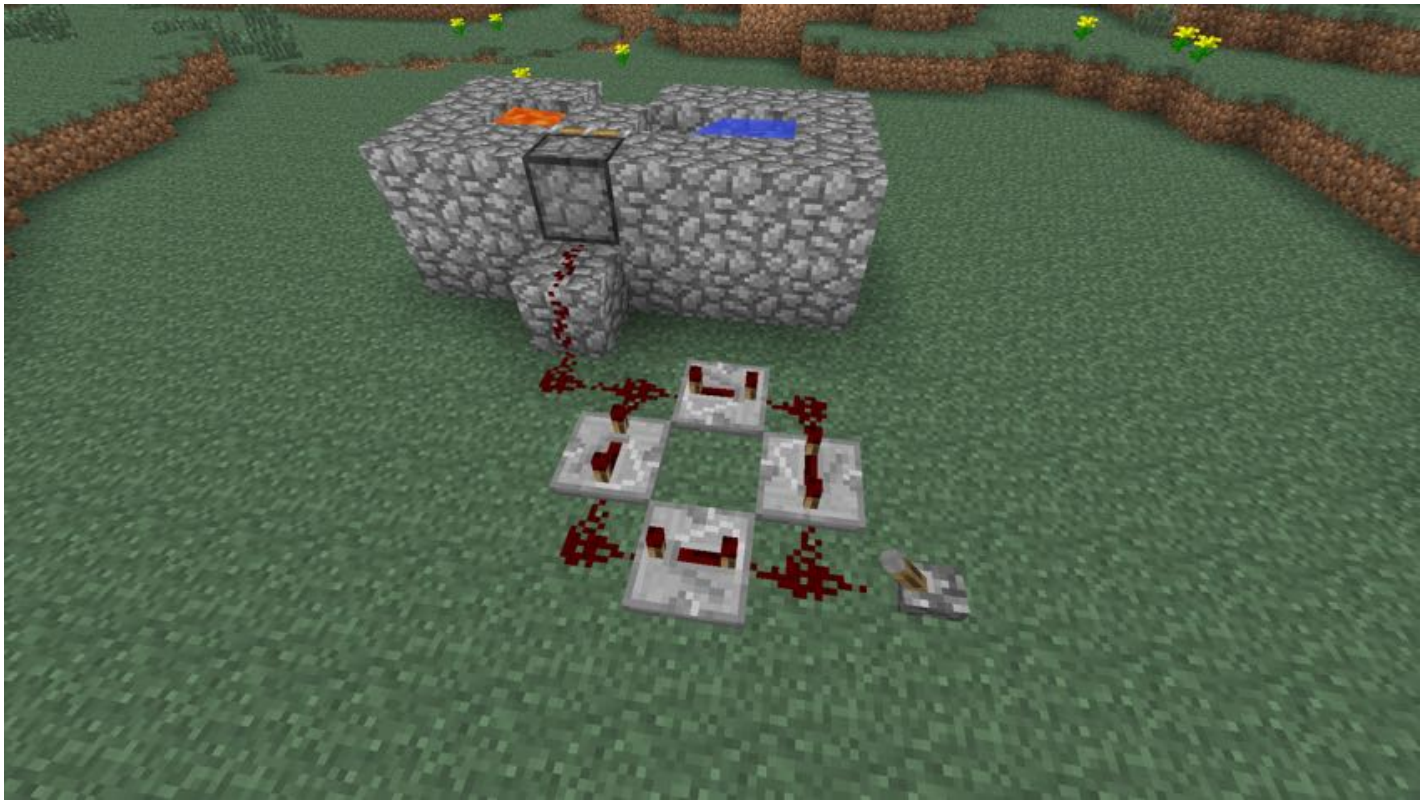
2. Place the water and lava source blocks from a bucket. Notice how cobblestone is formed in the middle!



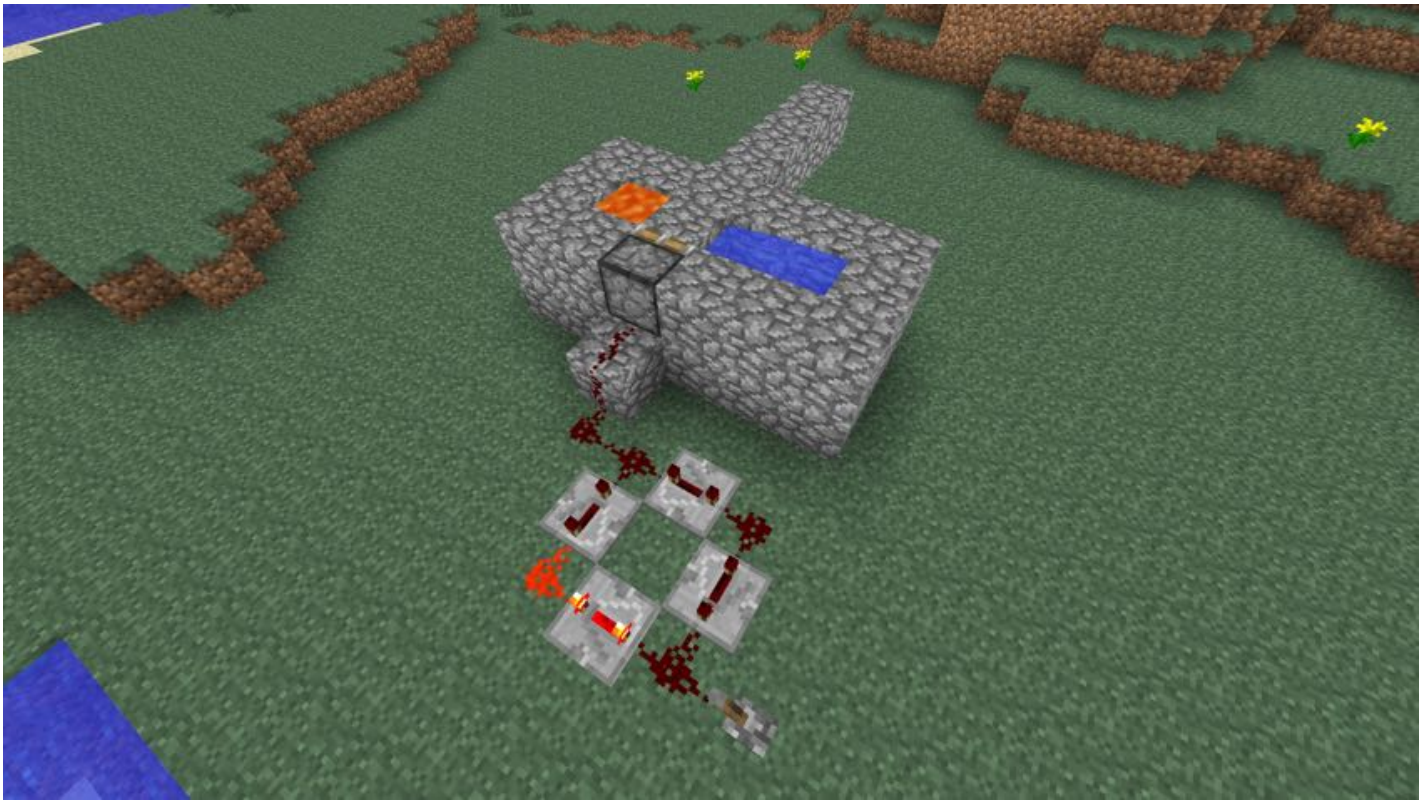
3. Place the piston that will push out the cobblestone blocks.



4. Create the redstone circuit that will cause the piston to automatically extend and then retract. We'll be talking about redstone clocks later in the lesson, but this one is simple enough that we can make use of it right now.



5. Quickly flip the lever on and off to start the generator. Pistons will push up to 12 blocks before stopping, but if you keep mining out the cobblestone column this generator will run forever.



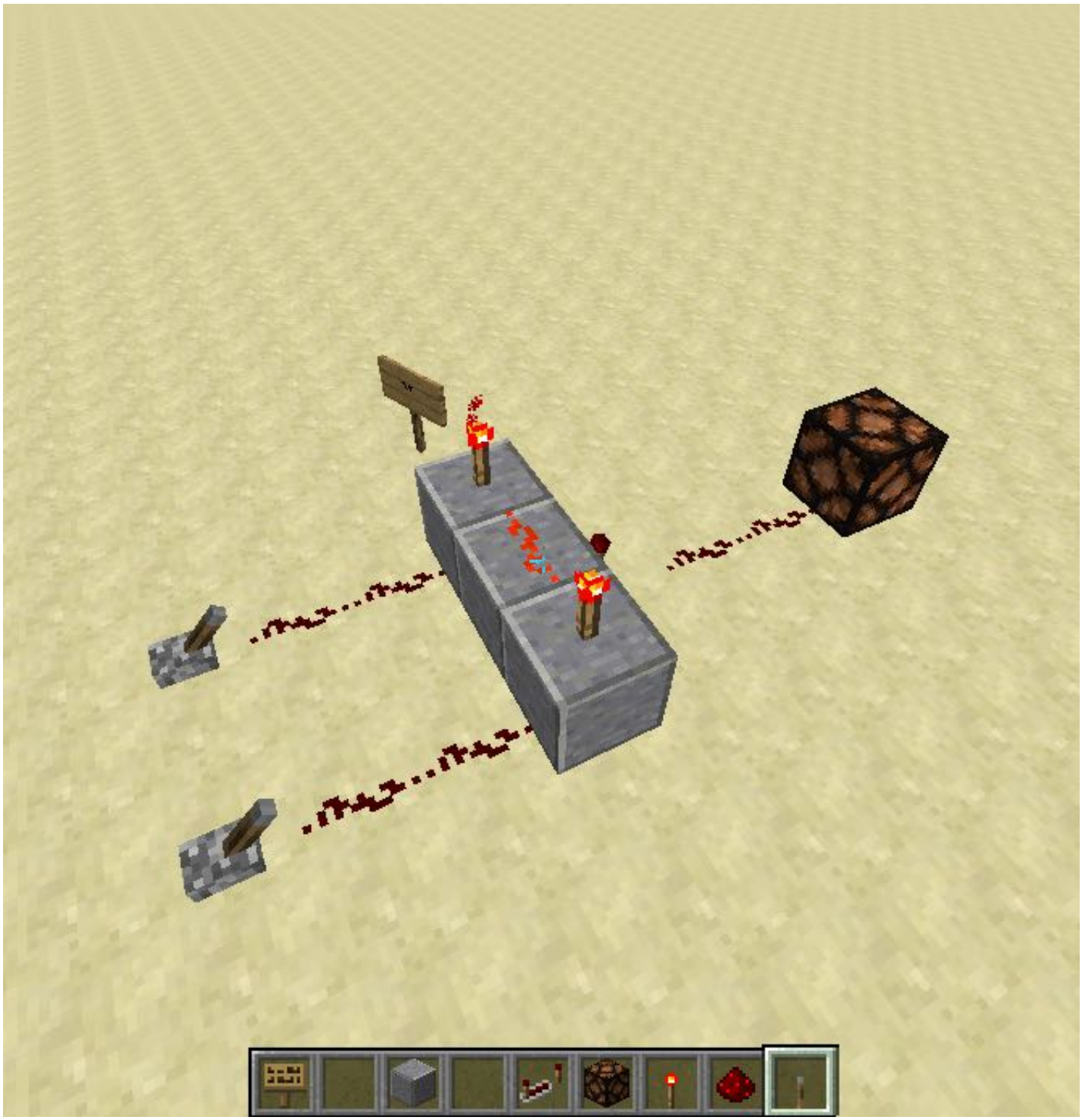
Section 3: Logic Gates

Logic Gates are a fundamental building block for digital circuits. They perform *boolean* functions and usually have 2 inputs and 1 output

AND

The AND gate performs the AND logic function which mathematically works like multiplication.

In Minecraft, your AND gate will take two redstone inputs which will either be ON or OFF. The output depends on the combination of inputs.



ON AND ON => ON
ON AND OFF => OFF
OFF AND ON => OFF
OFF AND OFF => OFF

Try these out on the above gate and see what happens. Remember that a lever is OFF if it is pointed upward.

Truth Tables

Truth tables are an easy way for us to organize the various outputs of logic gates given different inputs.
For boolean logic, “ON” is replaced with “True” or “1” and “OFF” is replaced with “False” or “0”. For example, the AND Gate could look like this:

TRUE AND TRUE => TRUE
TRUE AND FALSE => FALSE
FALSE AND TRUE => FALSE
FALSE AND FALSE => FALSE

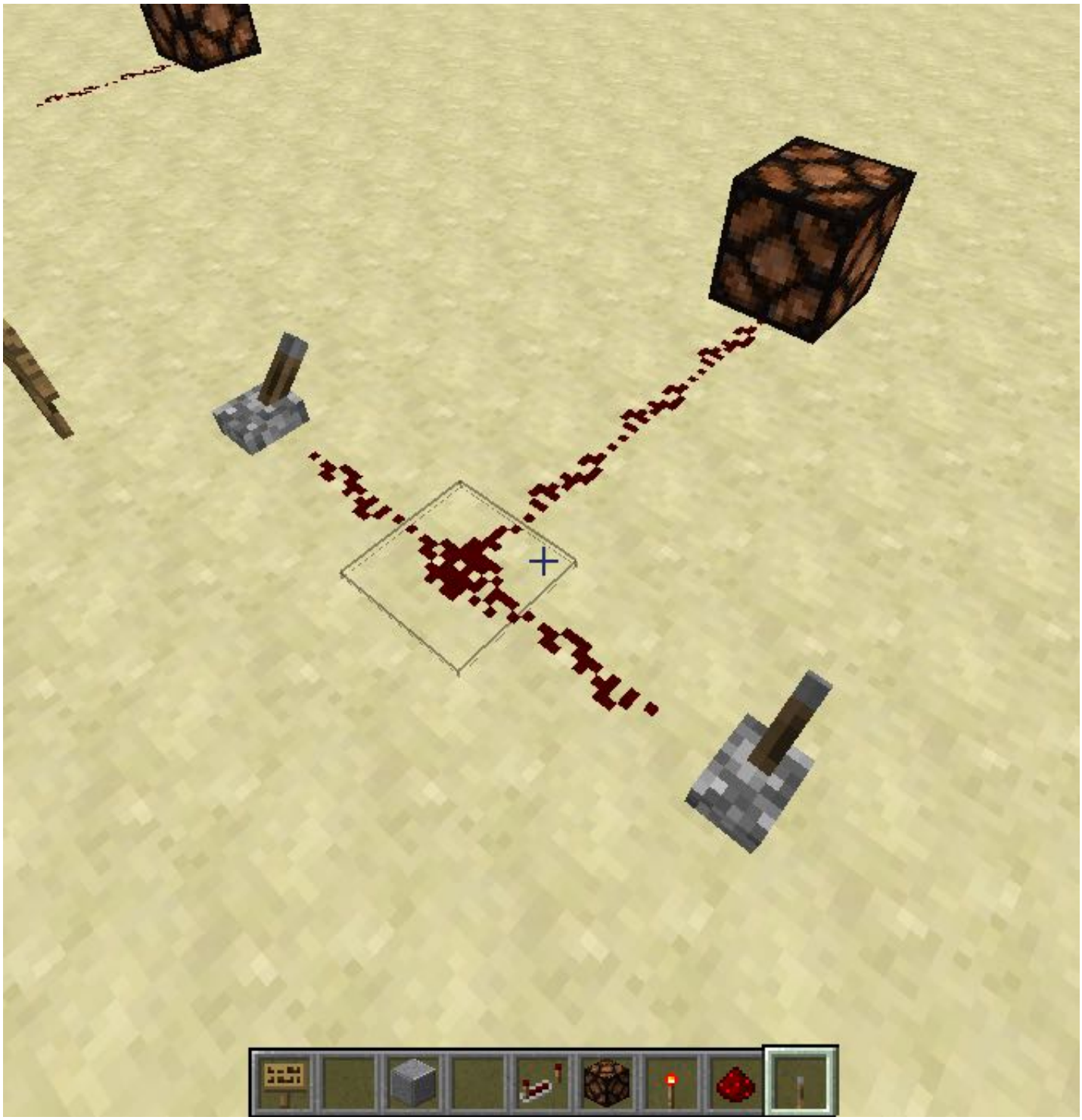
Which is the same as:
1 AND 1 => 1
1 AND 0 => 0
0 AND 1 => 0
0 AND 0 => 0

We can simplify this further to just a table:

Input	Input	Output
1	1	1
1	0	0
0	1	0
0	0	0

OR

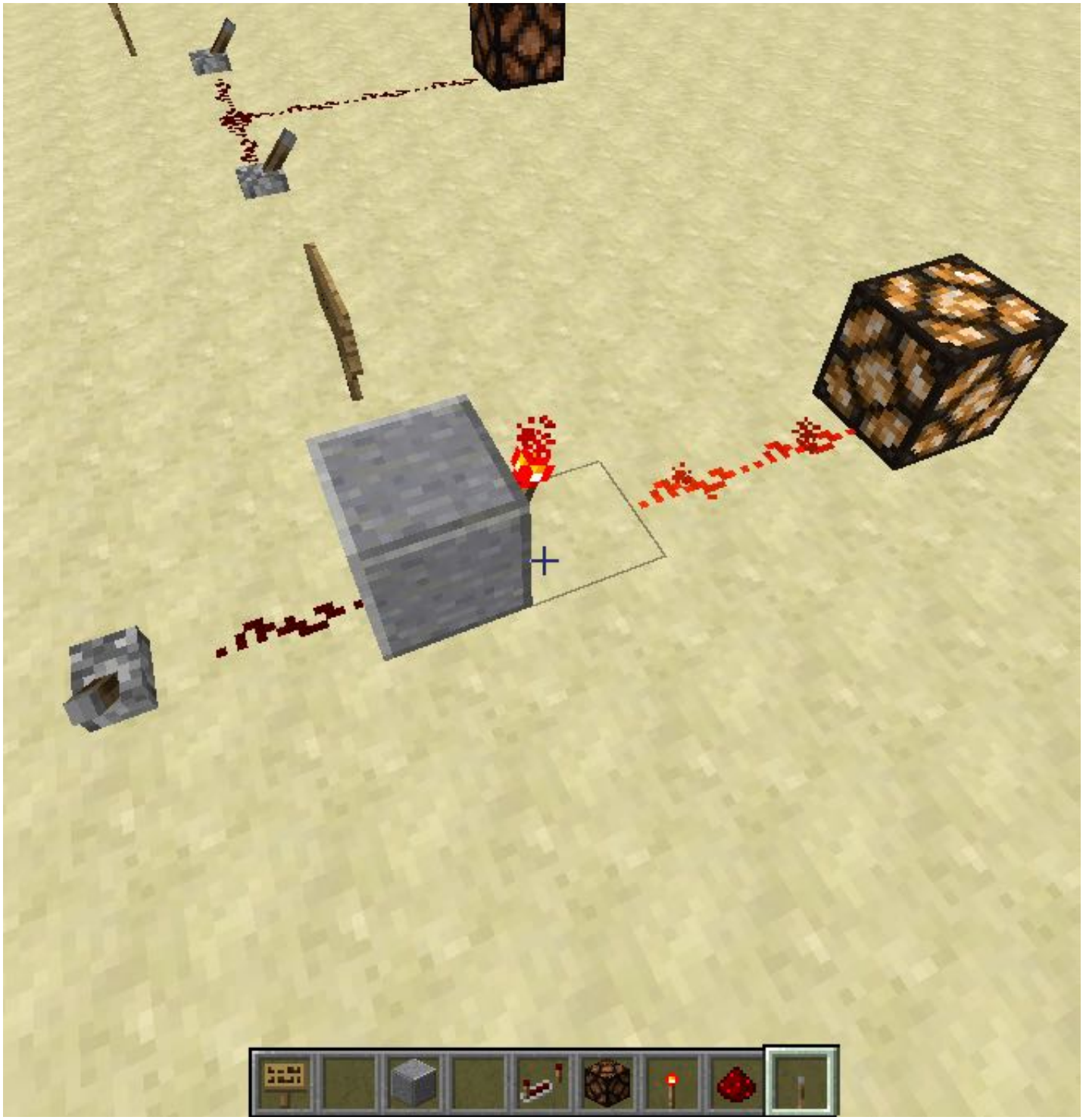
The OR gate will only output OFF if both inputs are OFF. If either input is ON or if both are ON, the output will be on.



Input	Input	Output
1	1	1
1	0	1
0	1	1
0	0	0

NOT

A NOT gate only has one input and simply reverses that input. An ON input leads to an OFF output, and vice-versa.



Input	Output
0	1
1	0

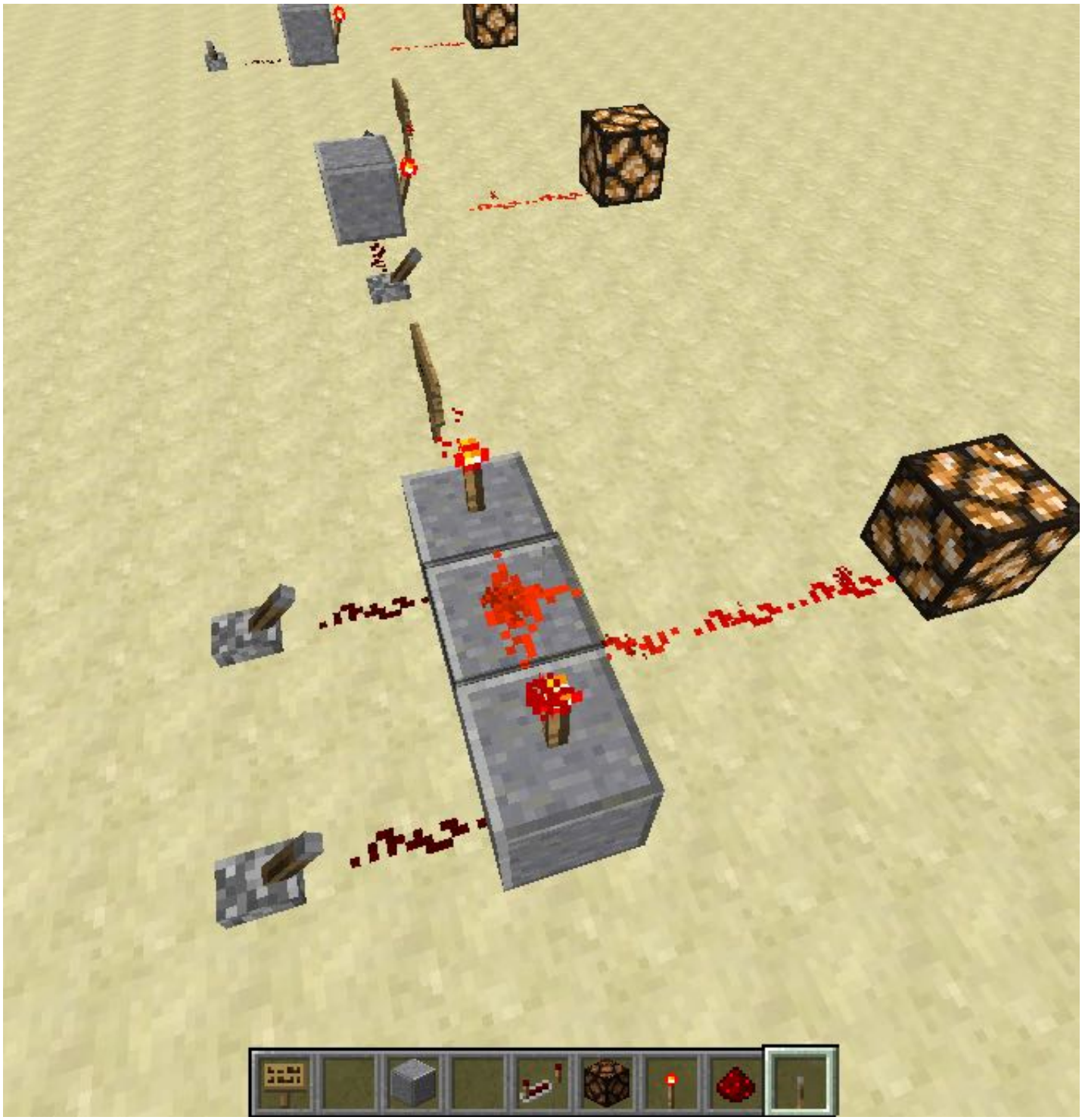
NOR

A NOR gate is just an OR gate with its outputs reversed. So a NOR gate will only output ON if both inputs are OFF. Otherwise the output is OFF.

Input	Input	Output
1	1	0
1	0	0
0	1	0
0	0	1

NAND

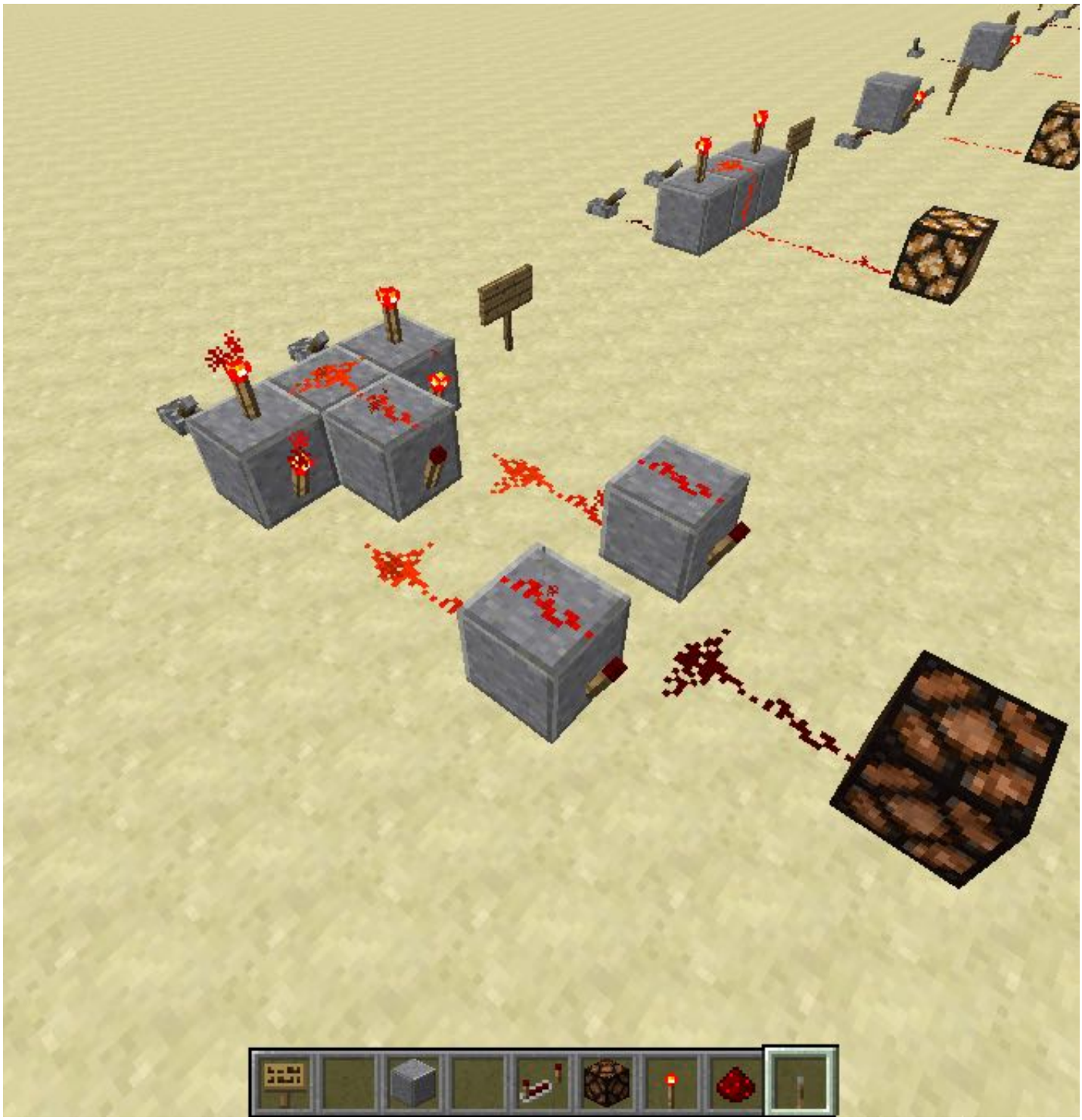
Just as a NOR gate has the opposite outputs of an OR gate, a NAND gate has the opposite outputs of an AND gate. It will output OFF only if both inputs are ON. If either input is OFF, the output will be ON.

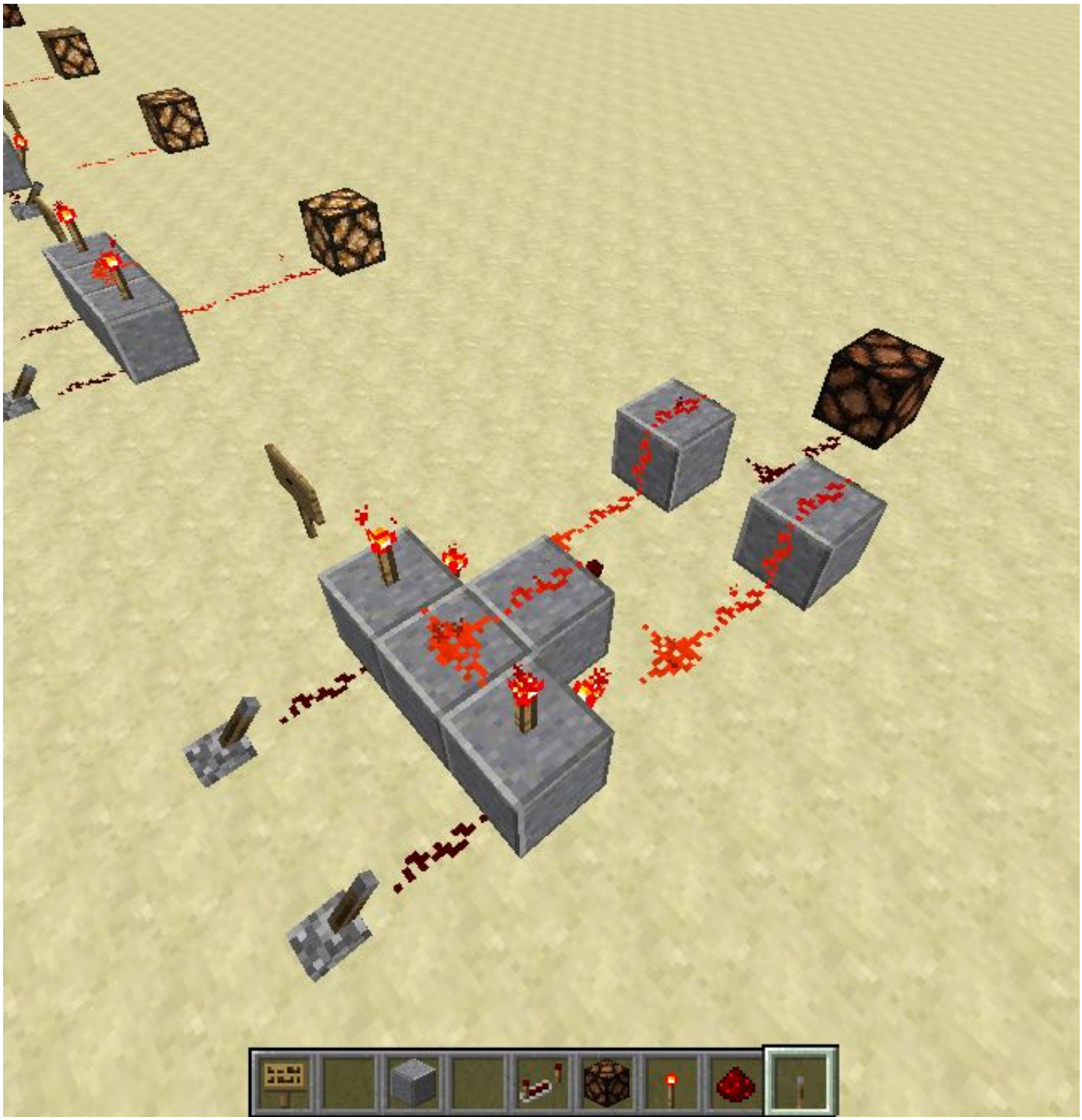


Input	Input	Output
1	1	0
1	0	1
0	1	1
0	0	1

XOR

An XOR (ex-or) gate is also called an “exclusive OR” gate. It will only output ON if either lever is ON. If both levers are either off or on, it will output OFF.

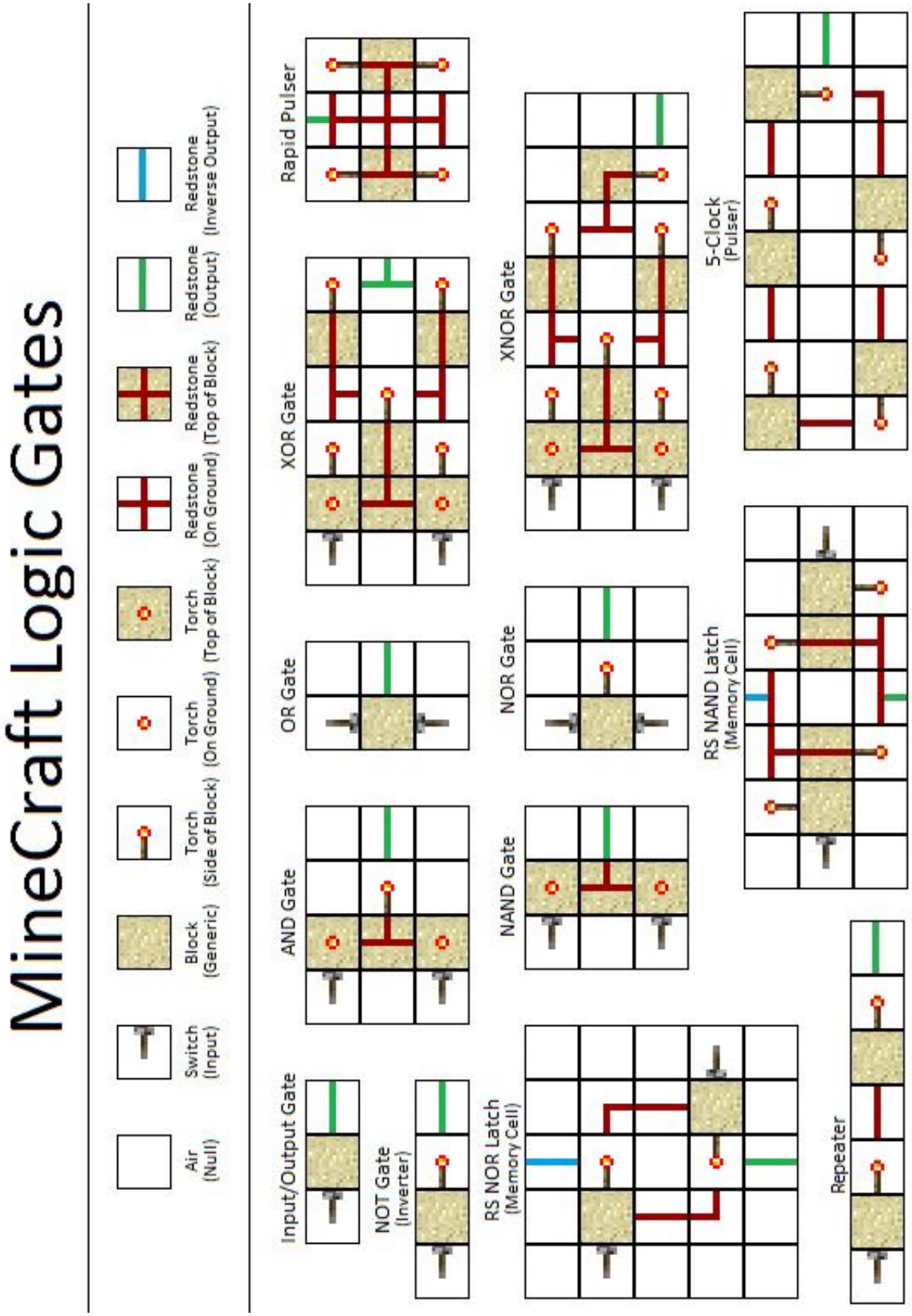




Input	Input	Output
1	1	0
1	0	1
0	1	1
0	0	0

Reference

This diagram has most of the logic gates that we've gone over, as well as some more that you may find useful.

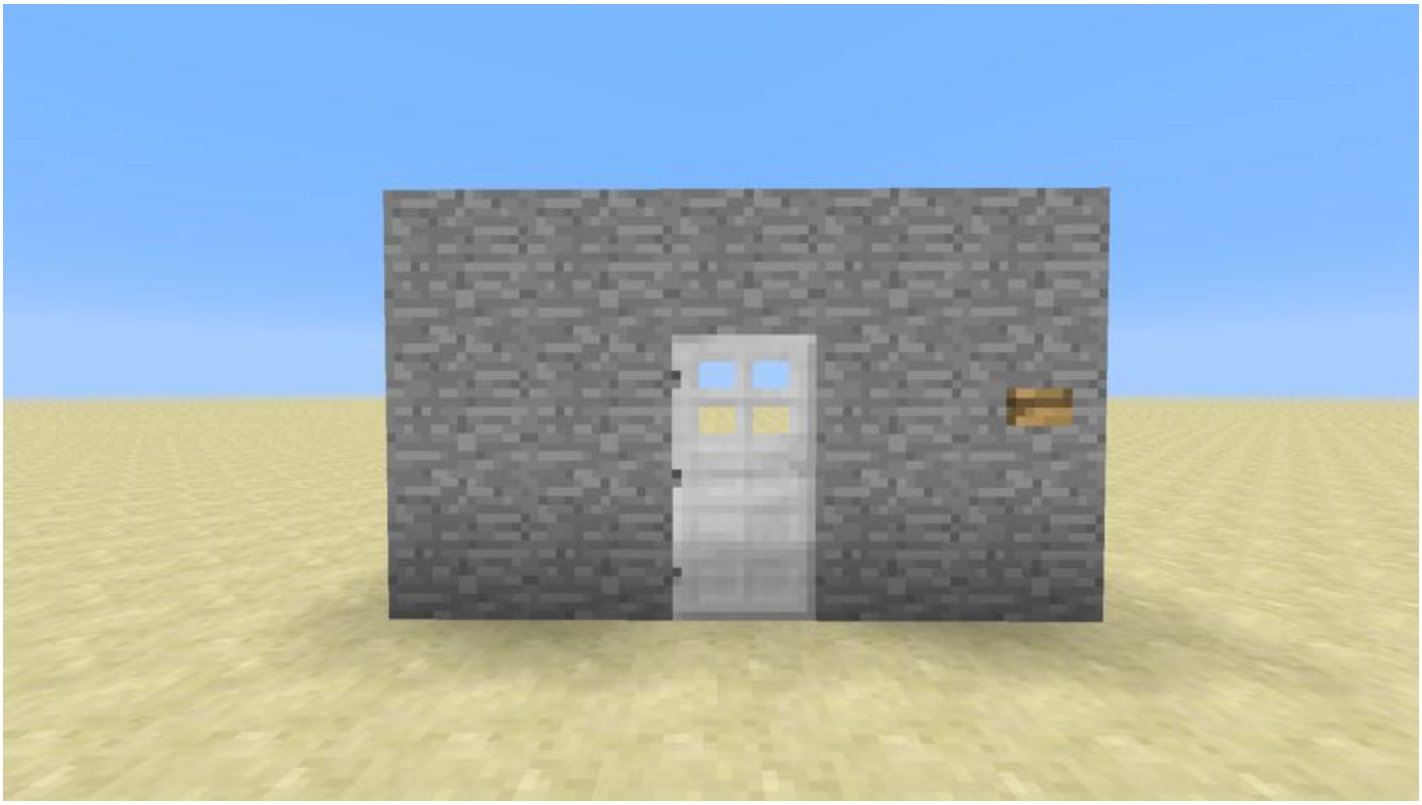


Section 4: Applications of Logic Gates

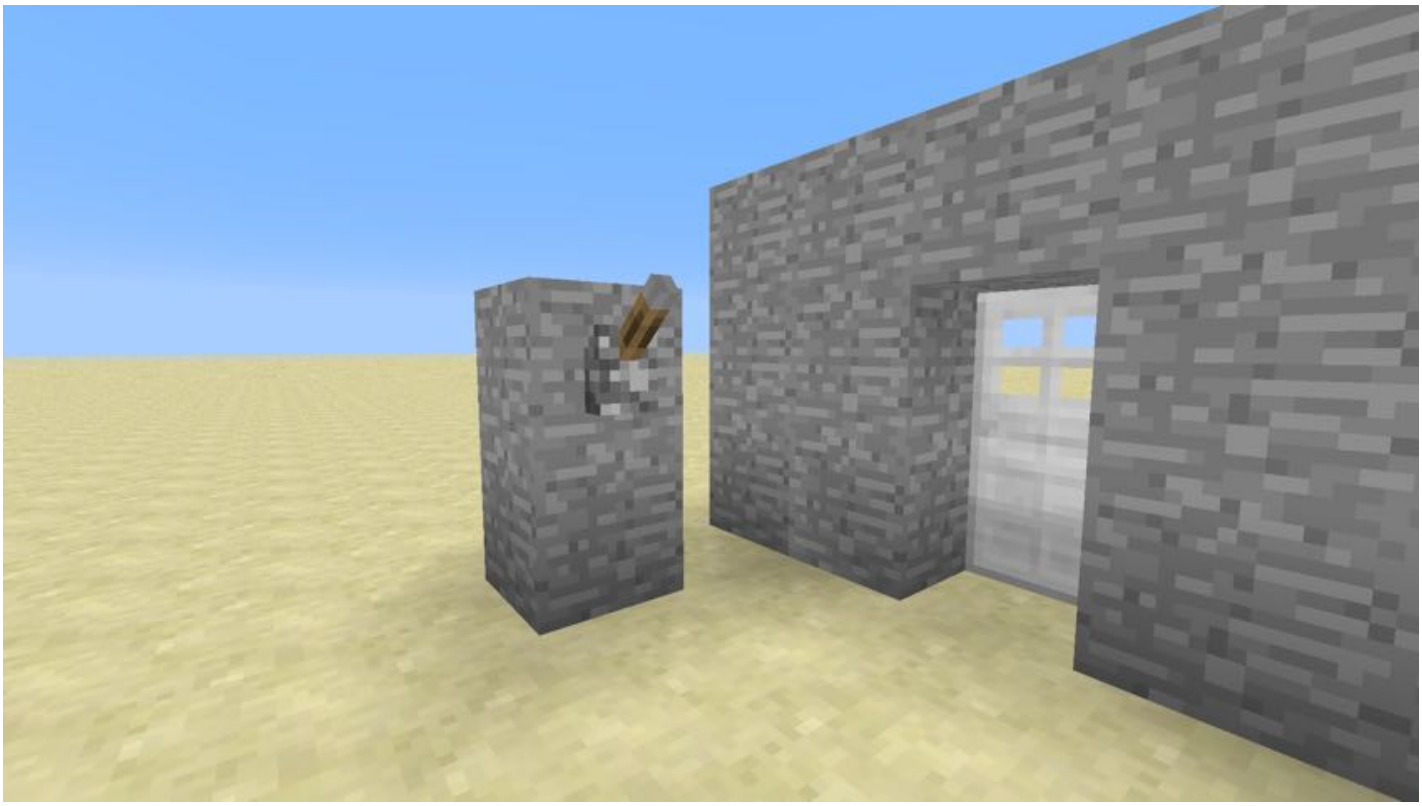
Locked Doors

Sometimes you may want to prevent people from entering a door using a lock. By having a lever on the inside, you can prevent the door from being opened unless the lever is in a specific state.

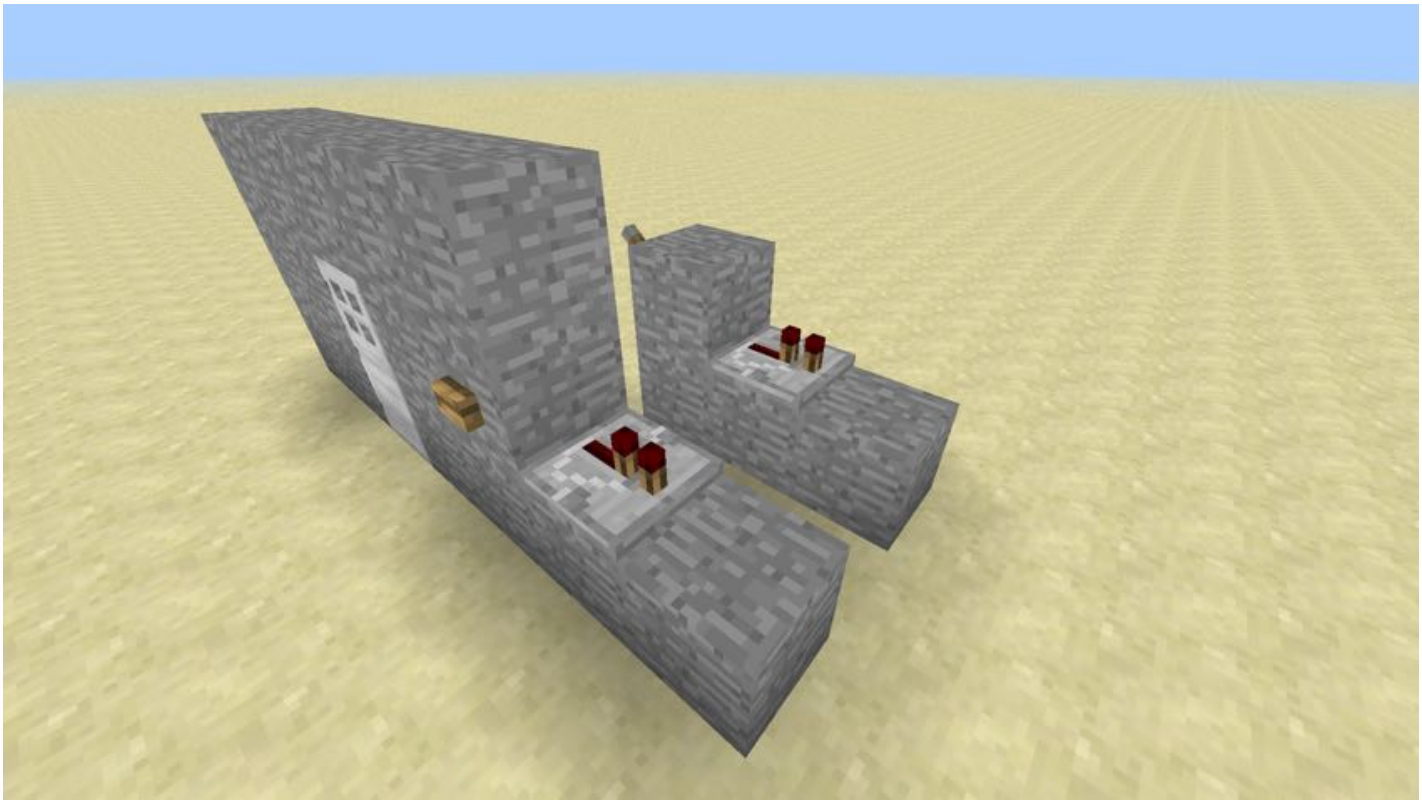
1. Place a door and build a wall around the door. A button should be placed one block away from the door; its block cannot be touching the door or it will open the door every time.



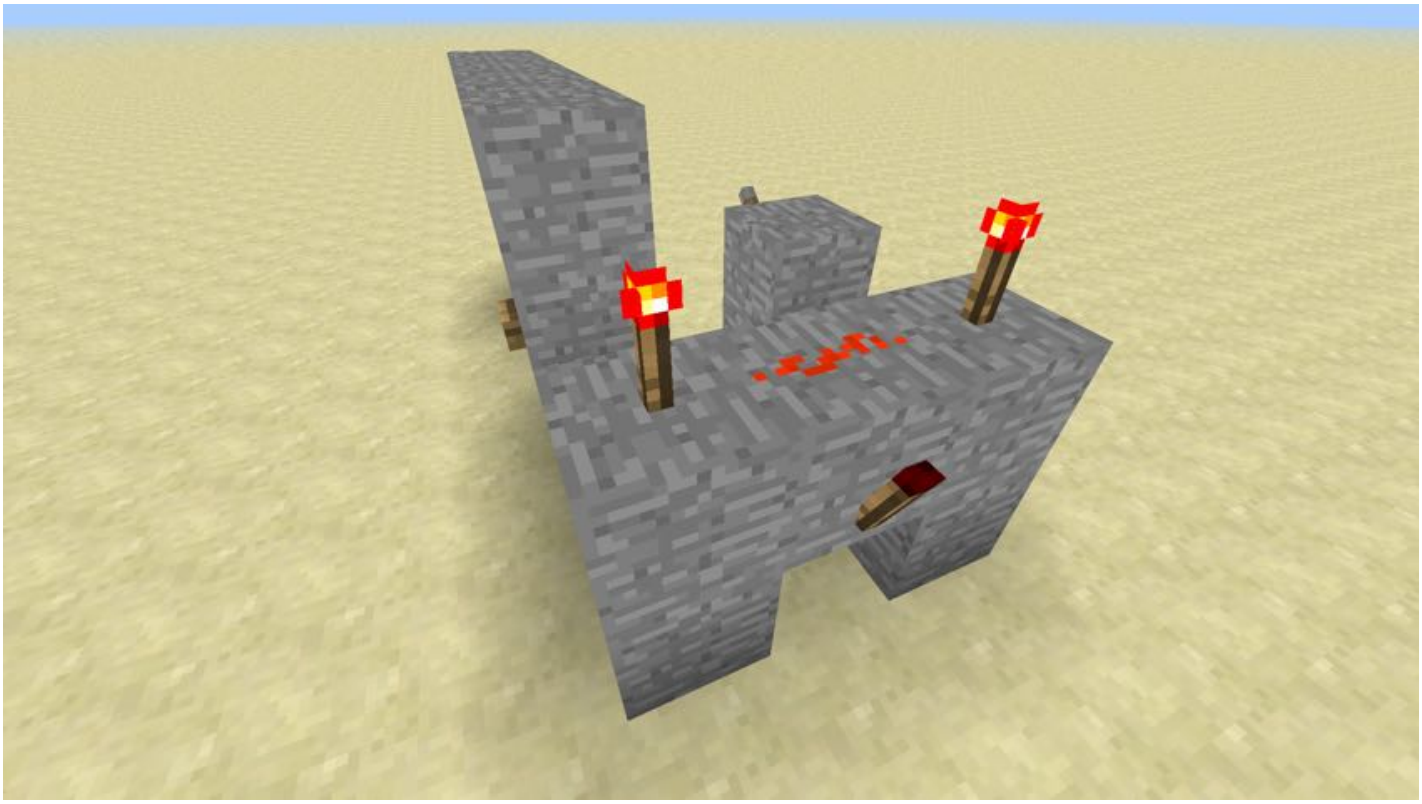
2. The location of the lock lever inside the door.



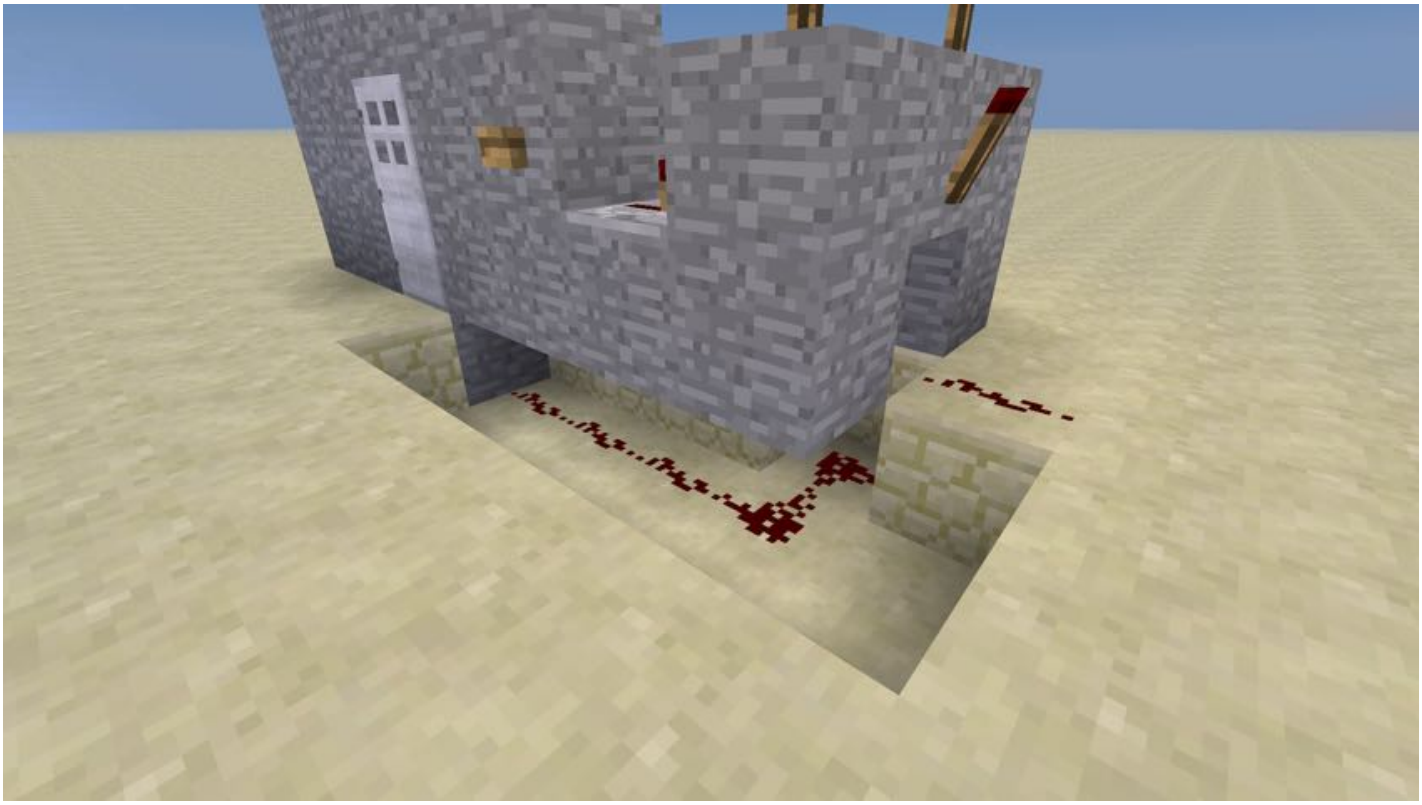
3. Repeaters that carry the redstone signal from both the button and the lever. Be sure to place the layer of stone underneath them so they are powered by the button and lever.



4. An AND gate. The two inputs are the lever and the button, and the output goes to the door. The output will only be ON when both the button and lever are ON as well.



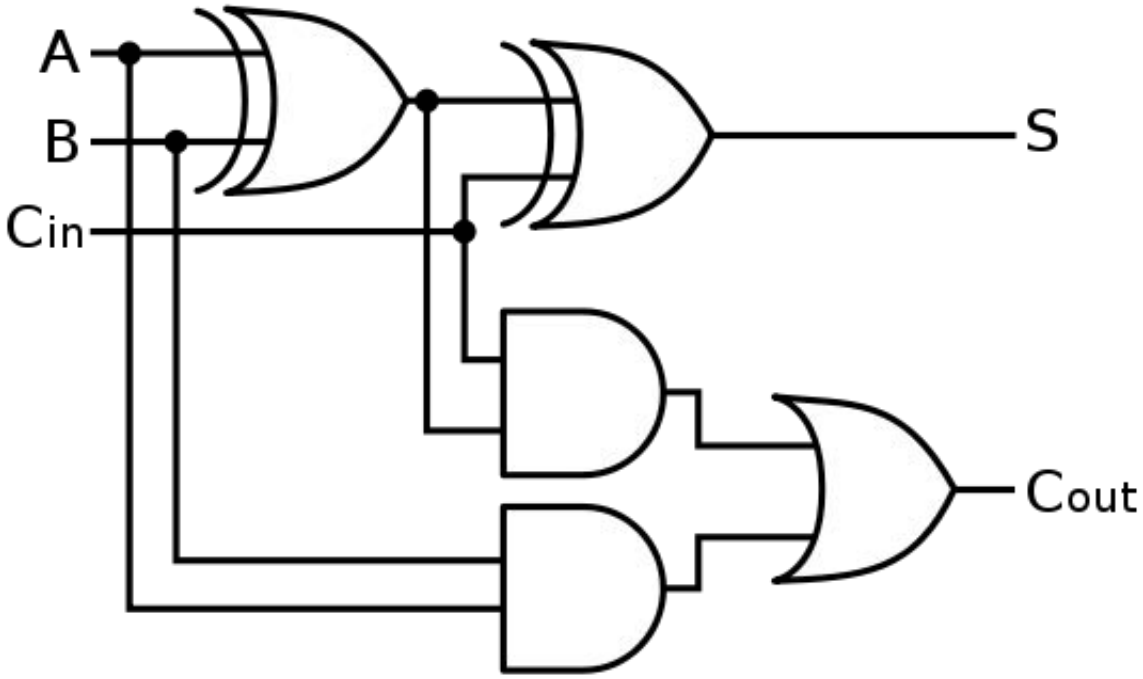
5. The redstone that carries the output of the AND gate to the block directly underneath the door. Now, the door will only open when the lever is set and the button is pressed.



Redstone Adder Circuit

An *adder* uses logic gates to perform addition. There are three inputs to the whole circuit: A, B, and carry. The carry is the result from the previous addition (think of how you carry a 1 when you add large numbers). The two outputs are the result and the carry for the next addition.

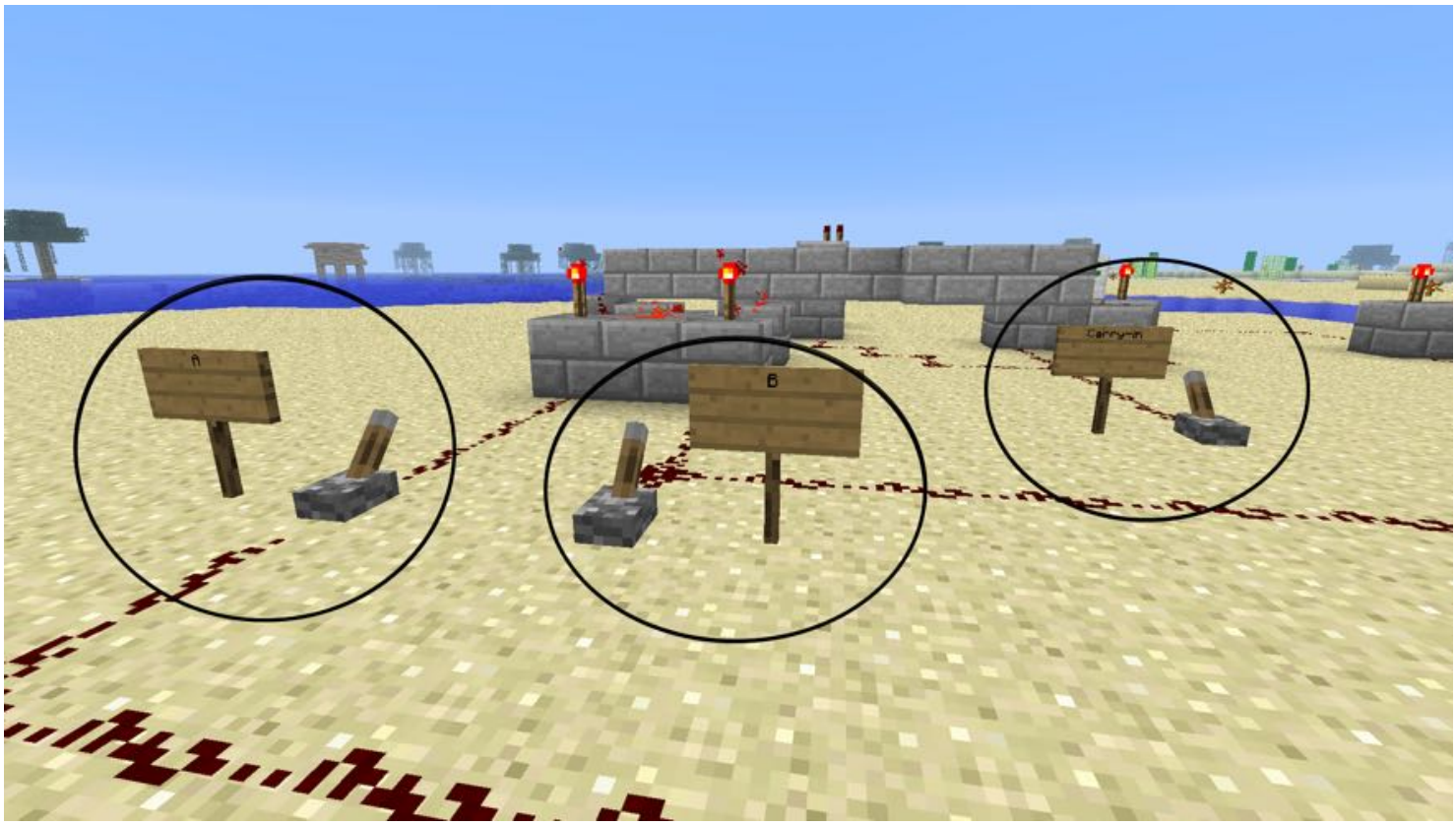
This diagram shows the overall structure of an adder.
Note: A dot means the two lines of redstone should connect. Other overlaps should not connect.



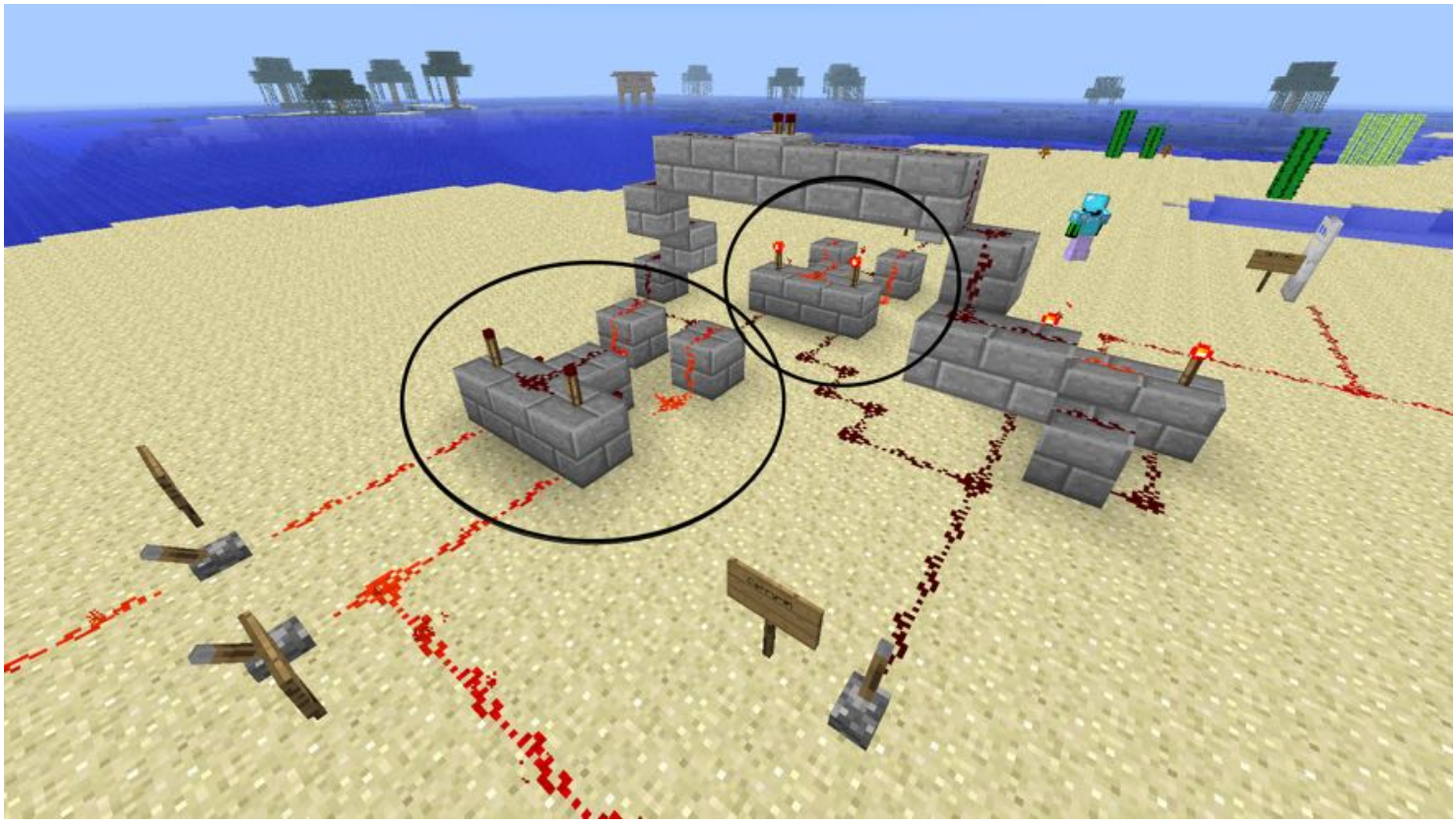
Look at the logic gates key so you can begin to understand how the adder works.

Now, we're going to build an adder using redstone! Follow along with the steps as best you can. This is a complex circuit so be sure to ask questions. The doors at the end represent the output. The door to the left represents the normal output, and the door to the right represents the carry. An open door is a 1 and a closed door is a 0.

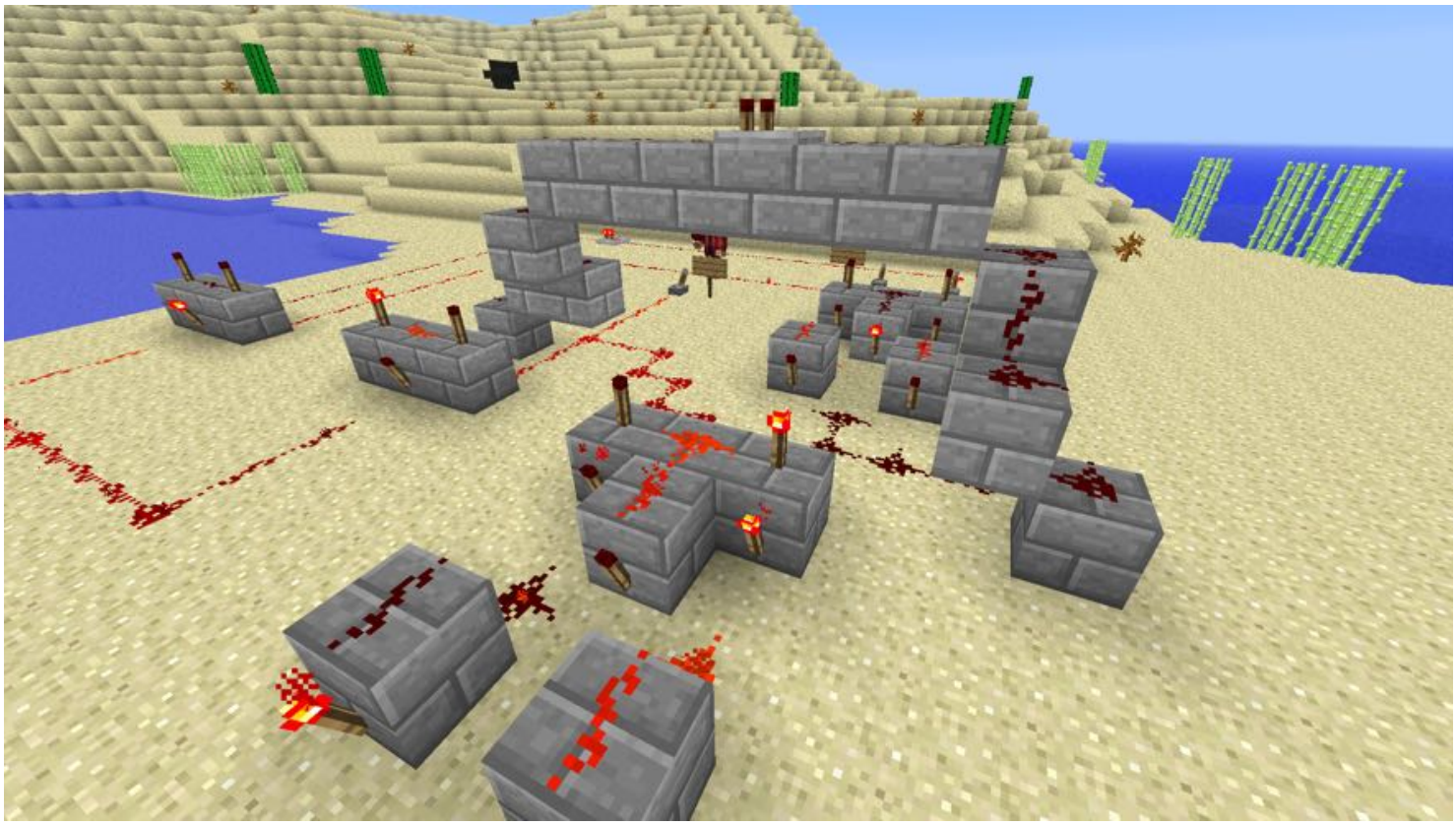
There are three inputs, so be sure to put signs to label each one. Levers are best since we can switch them between ON and OFF easily.



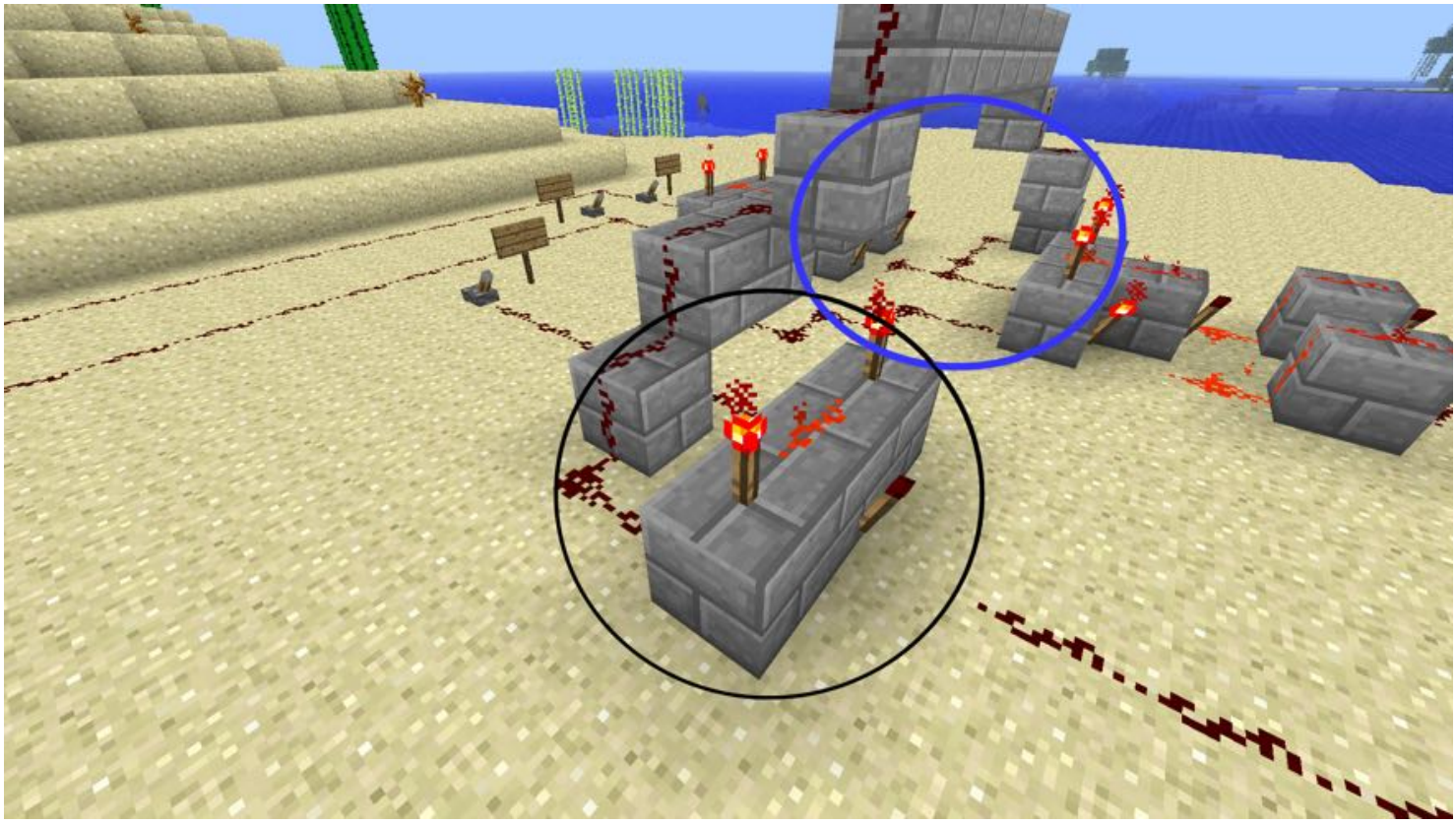
Next, build two XOR gates. The first one (on the left; we'll call it XOR gate 1) takes input from A and B. The second one (XOR gate 2) takes input from the output of the first gate and the carry.



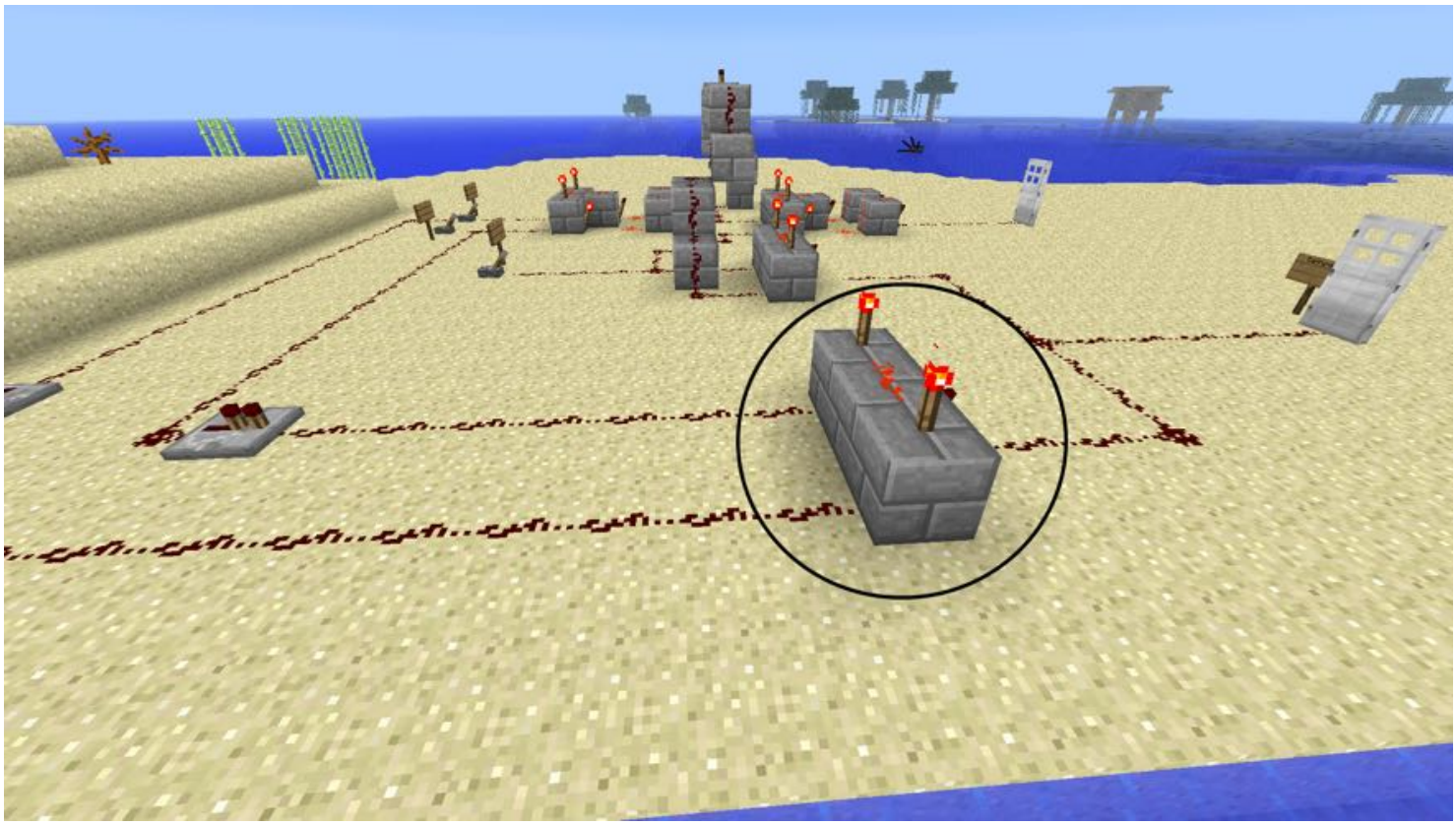
A second perspective of the XOR gates.



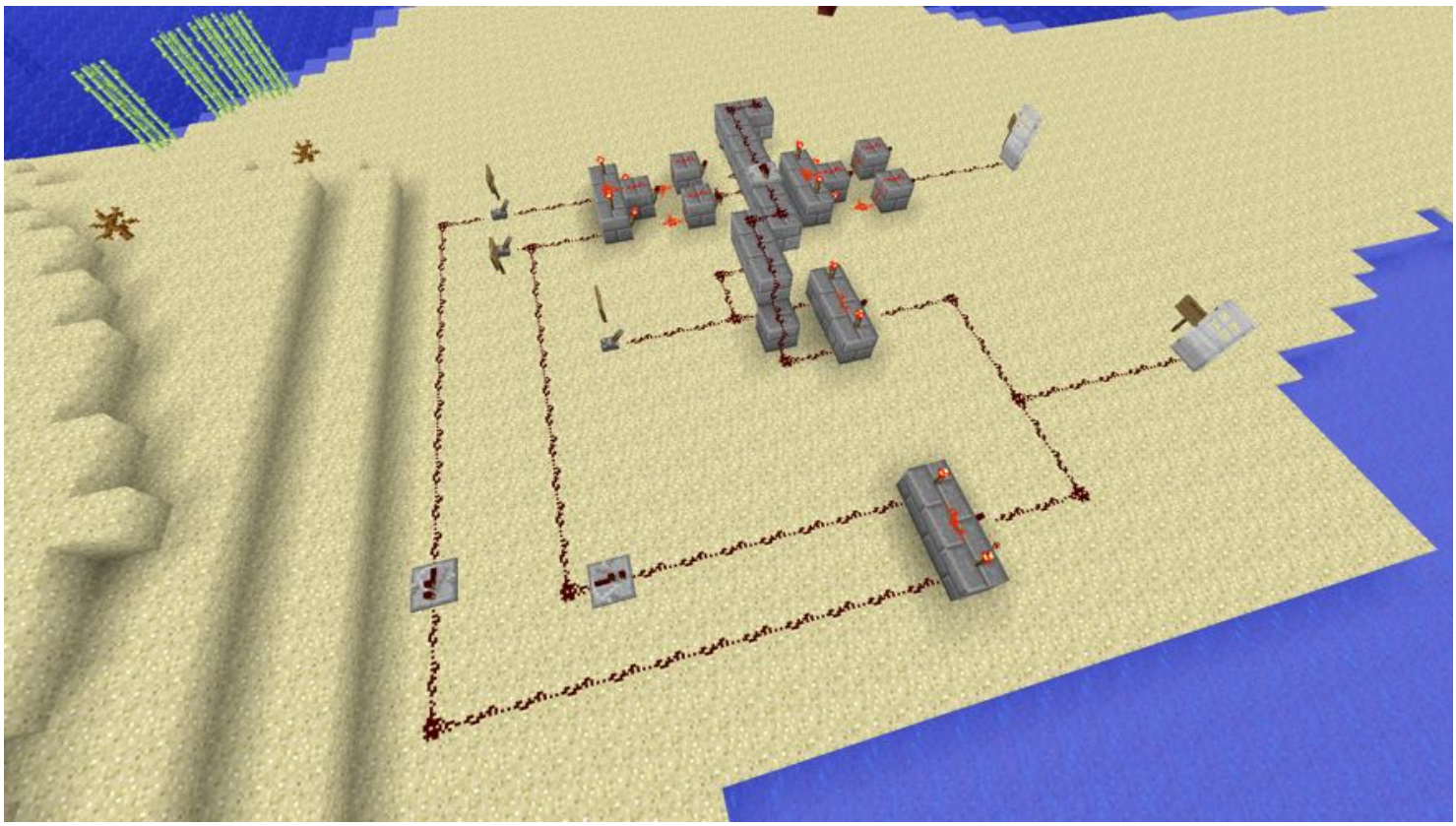
This AND gate takes input from the carry and the output of XOR gate 1. The blue circle shows where the gate 1 output goes up and over to this AND.

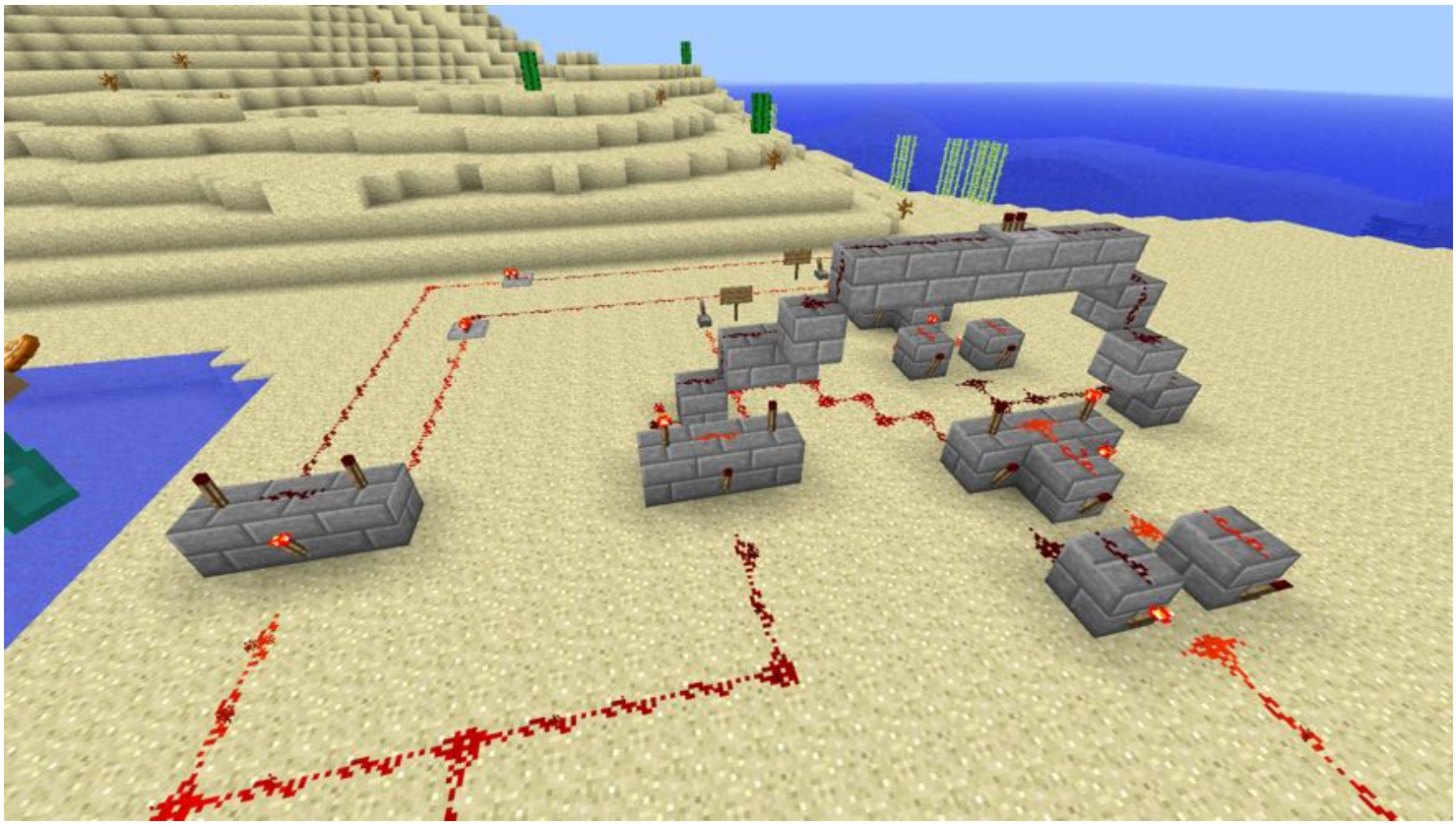


The second AND gate. This one takes input from the original A and B inputs. Notice how it and the output of the other AND gate combine and go to the door representing the carry.



Two overviews of the entire adder from different perspectives.





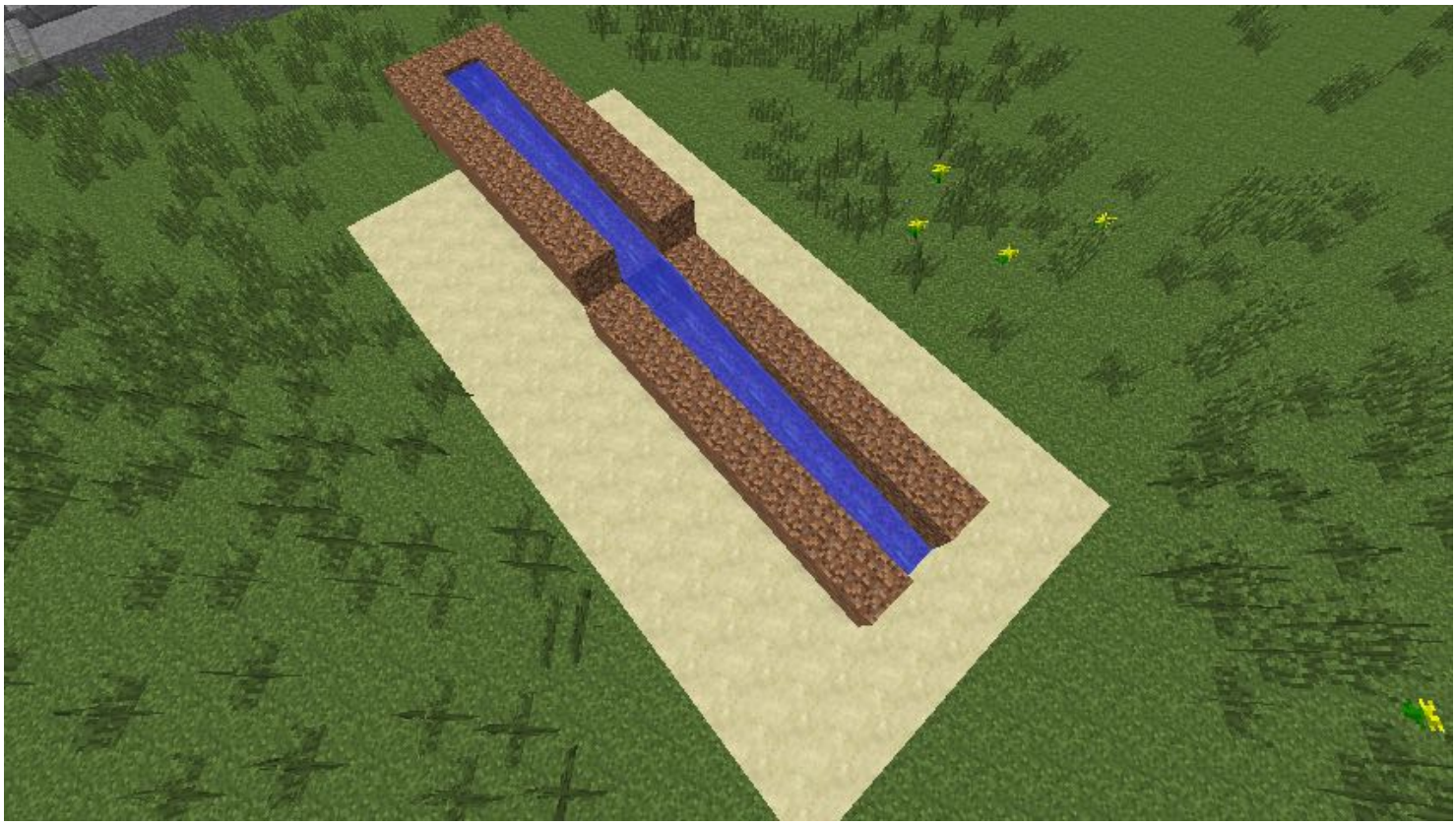
Automatic Farming

Farming isn't exactly difficult, and can be quite relaxing, but if you wanted to speed things up a bit there is a simple redstone contraption you can construct to automatically harvest your crops.

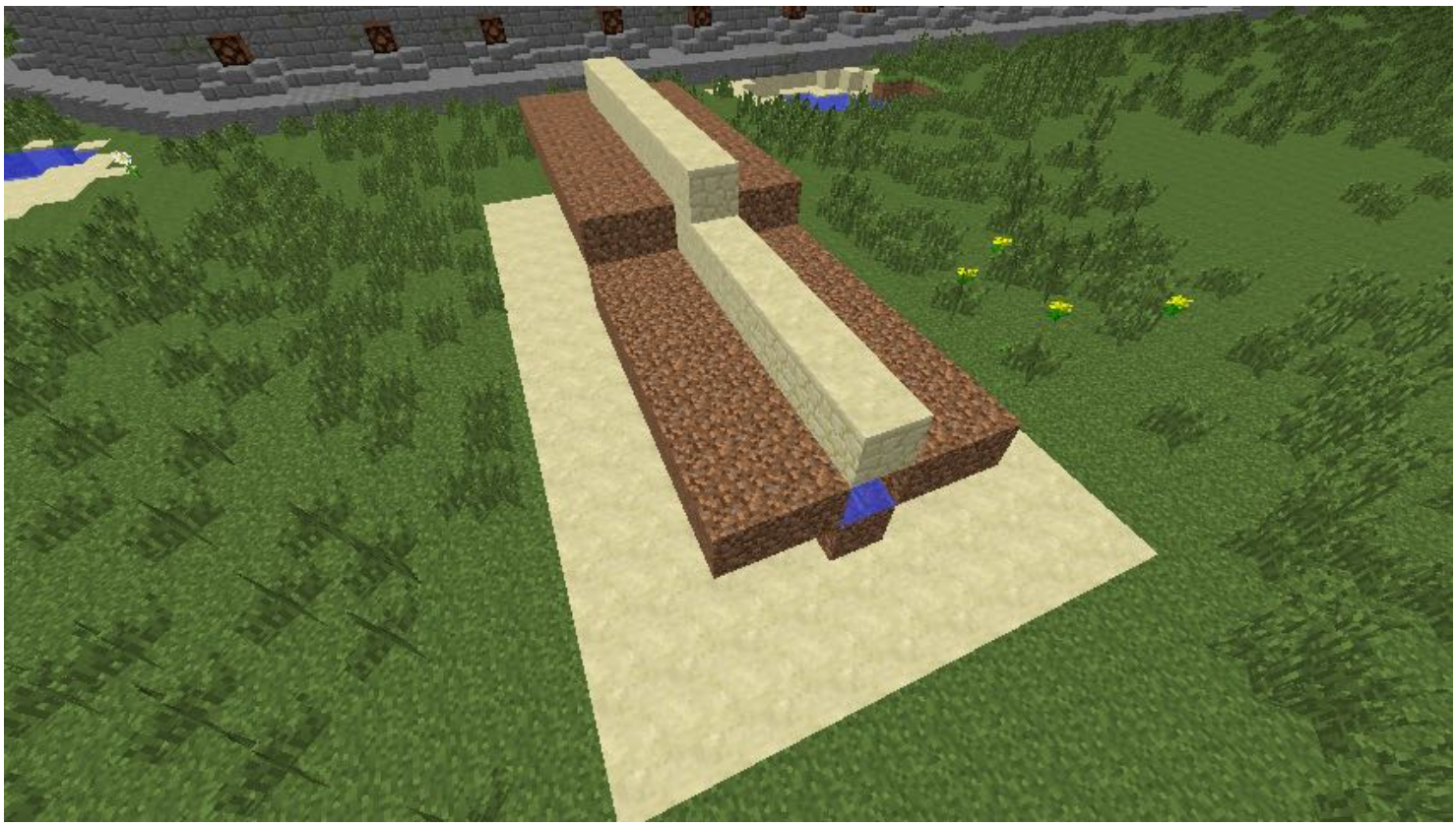
Water will flow 8 blocks, which is an important fact to know for many purposes. This farm will use water flowing downhill over the crops to automatically harvest them and drop them into hoppers, and then a central chest where you can pick up the foodstuffs.

Create a new creative world for this exercise.

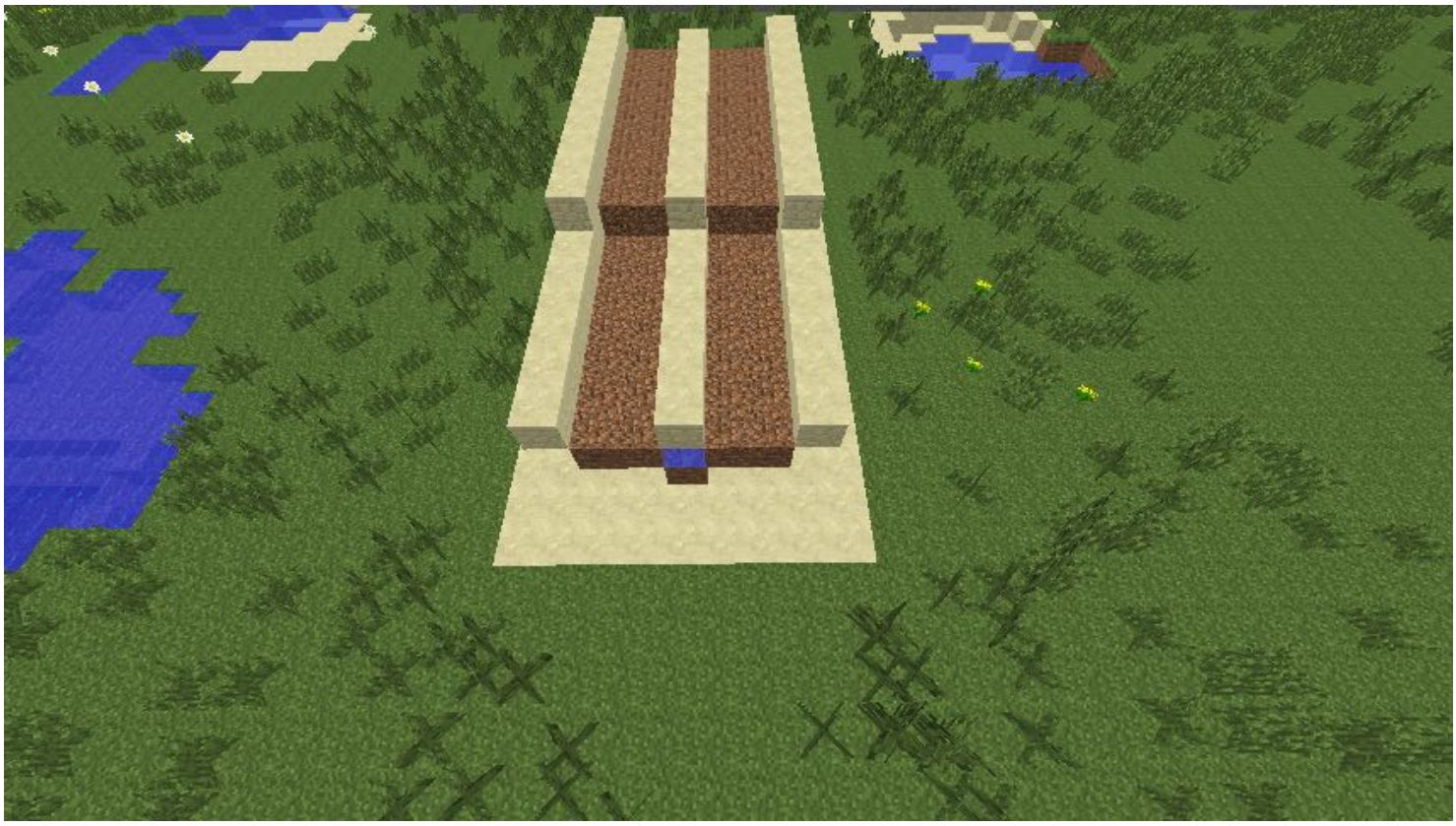
First we'll set up the irrigation system for our farm by creating a canal down the center of our farming area. Both levels are eight blocks long, but the water starts at the second block so that it will flow off the end of the first level and down the second level.



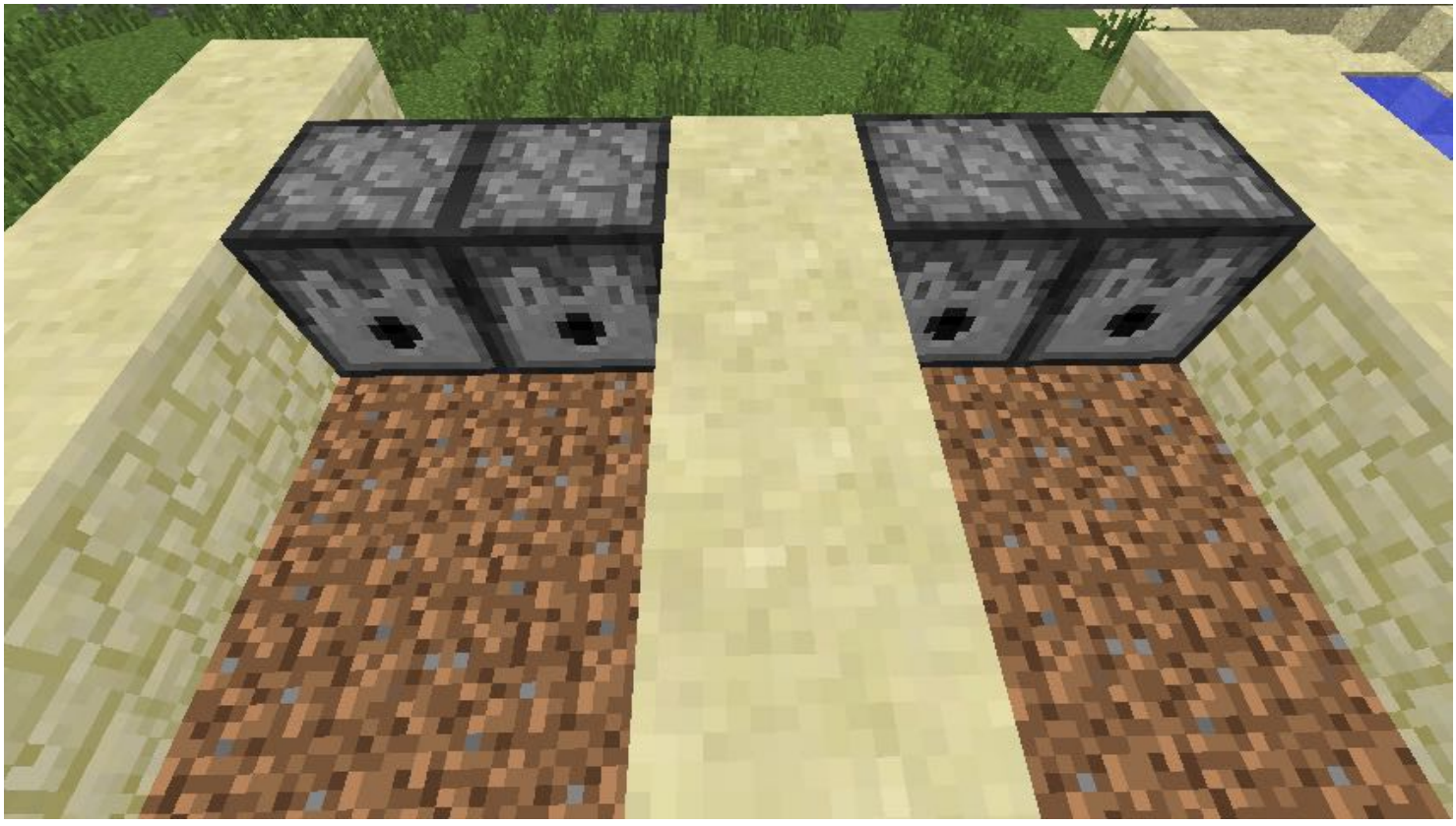
Next cover the irrigation canal (using whatever block you want) and create the farm plots next to the canal:



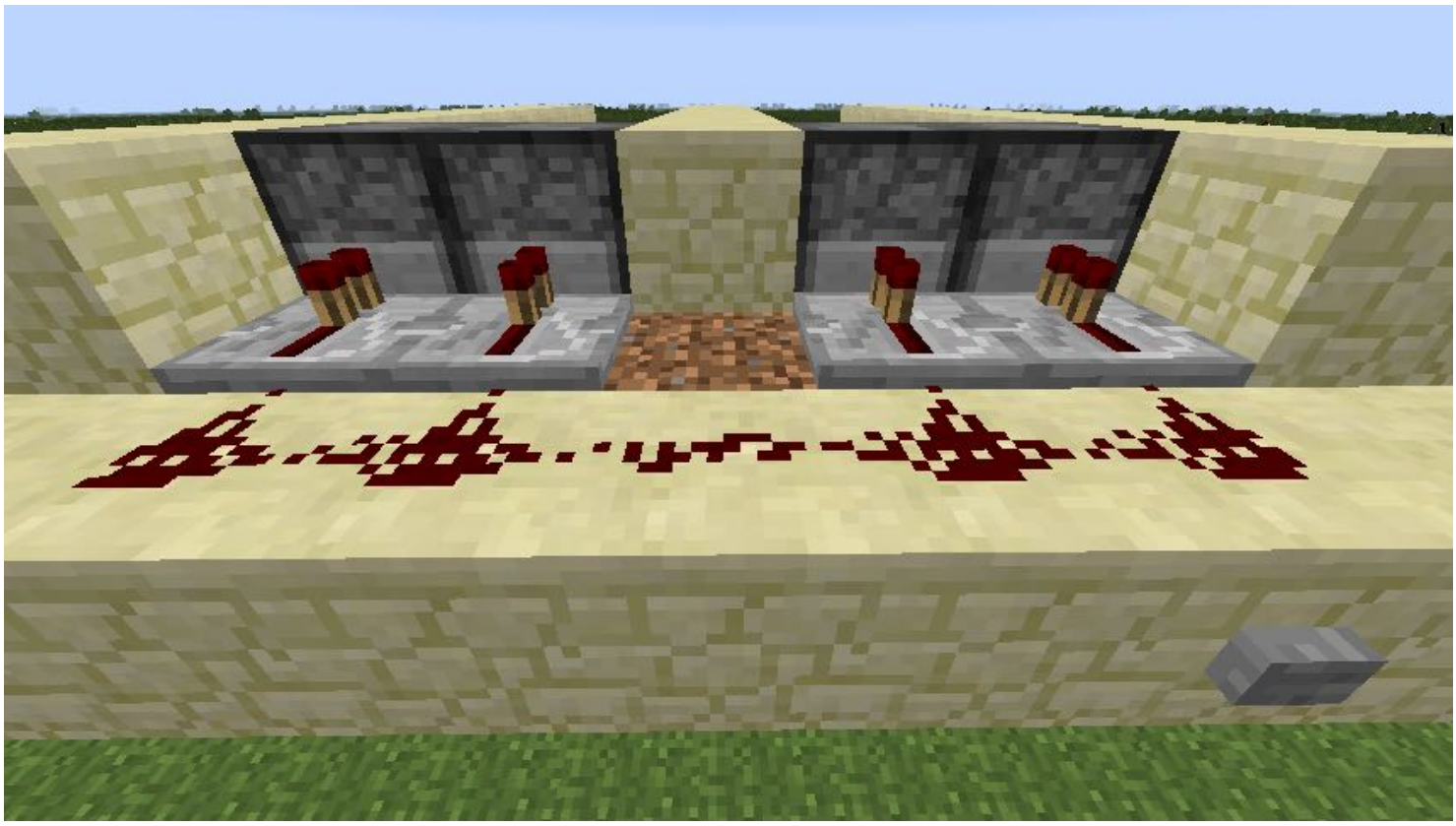
Now create walls along the side of your plots. These will keep the water from flowing over the sides.



Now we'll place dispensers along the top of our farm. These will dispense the water that will harvest our crops and carry them down to the bottom of the farm.

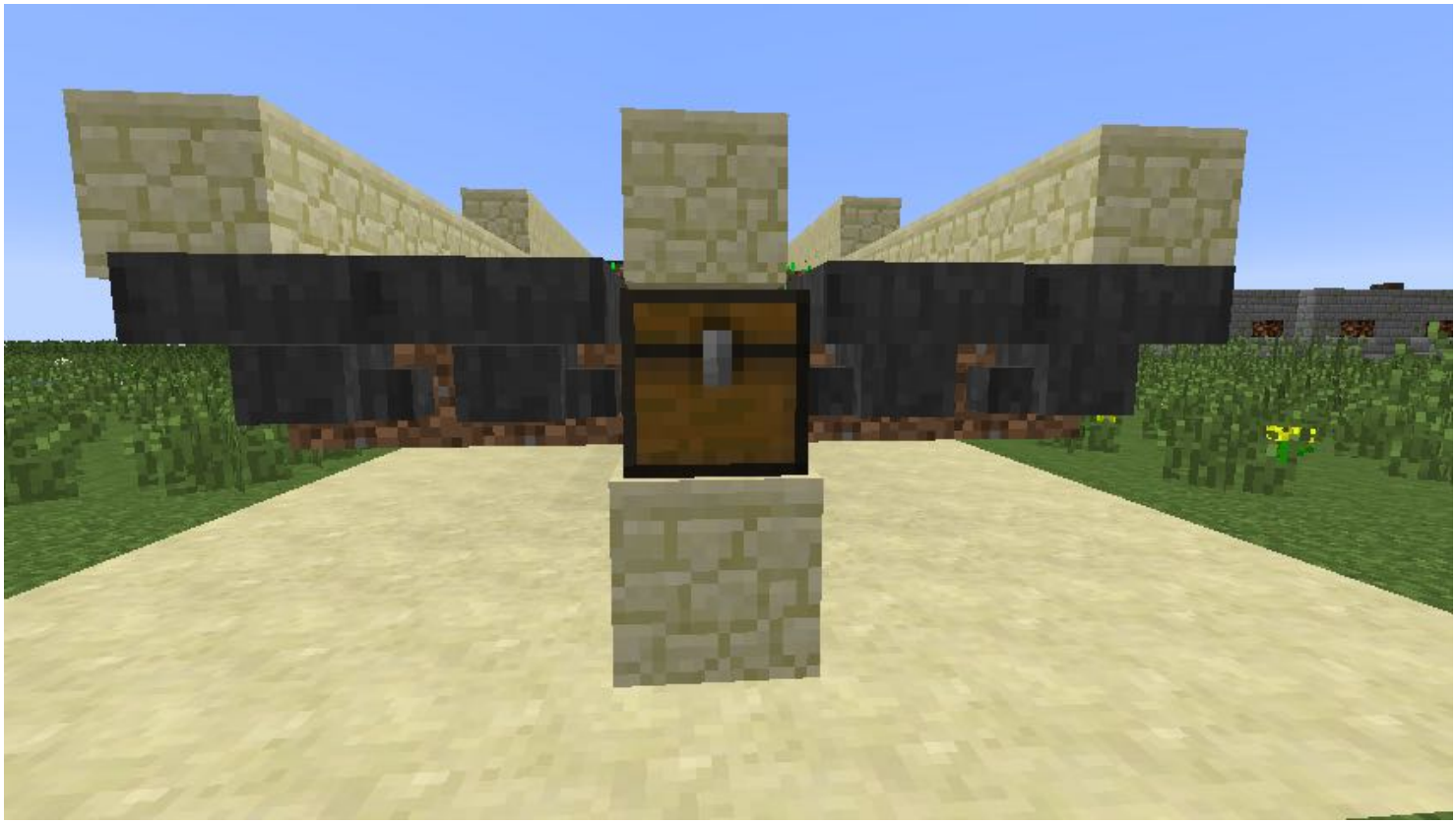


To power the dispensers, you'll need redstone repeaters, redstone and a button, like so:

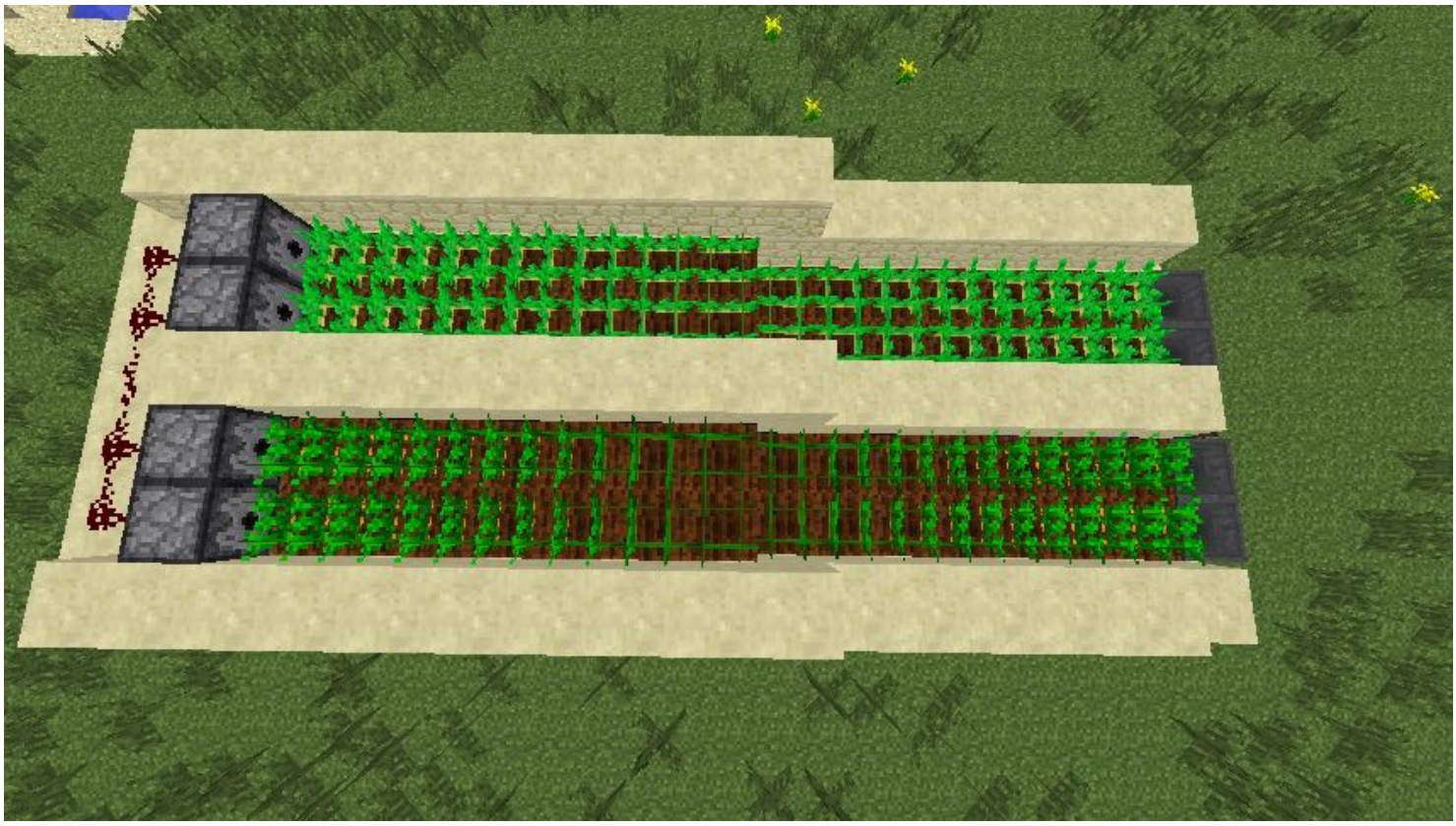


Place a bucket of water in each dispenser. The nice thing about this setup is the water buckets will remain full even after the dispenser dispenses the water. Plant crops of your choice. Here we've planted some carrots on the left and potatoes on the right:

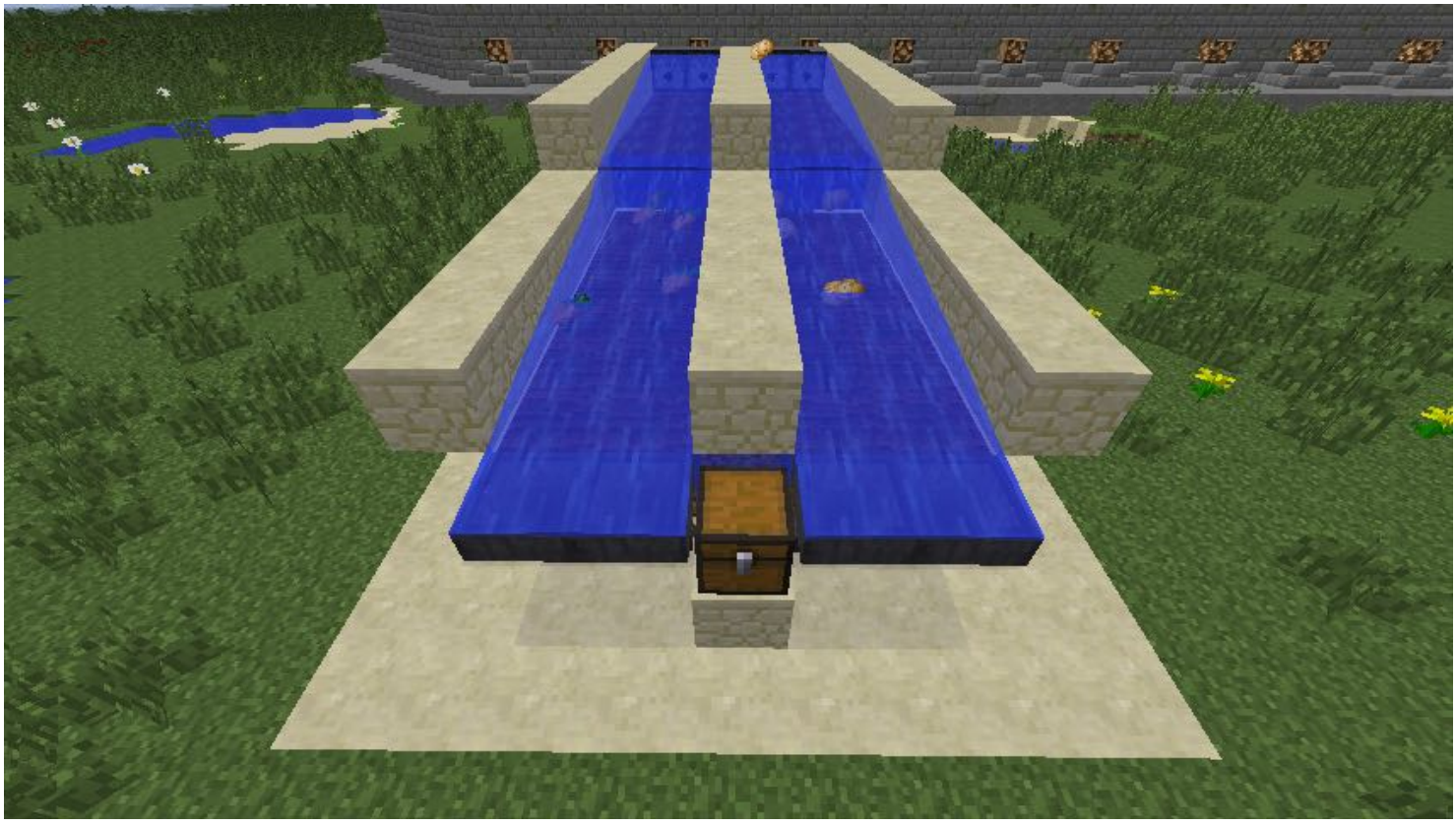
Now we need to get the collection system in place. Place a chest at the bottom-center of your farm, directly in front of where the irrigation canal stops. Then place hoppers on either side of the chest. To do so take a hopper and then shift-click on the side of the chest to place the hopper. Do the same on the side of the hopper with another hopper. Now everything falling into these four hoppers will go into the chest in the center:



Now we have to wait for our crops to grow, but in creative we have a shortcut. Go ahead and take some bonemeal in your hand and right-click on your crops until they’re fully grown.



To harvest, go press the button connected to the hoppers. The water will flow down over your crops, harvesting them and carrying them down to the hoppers.

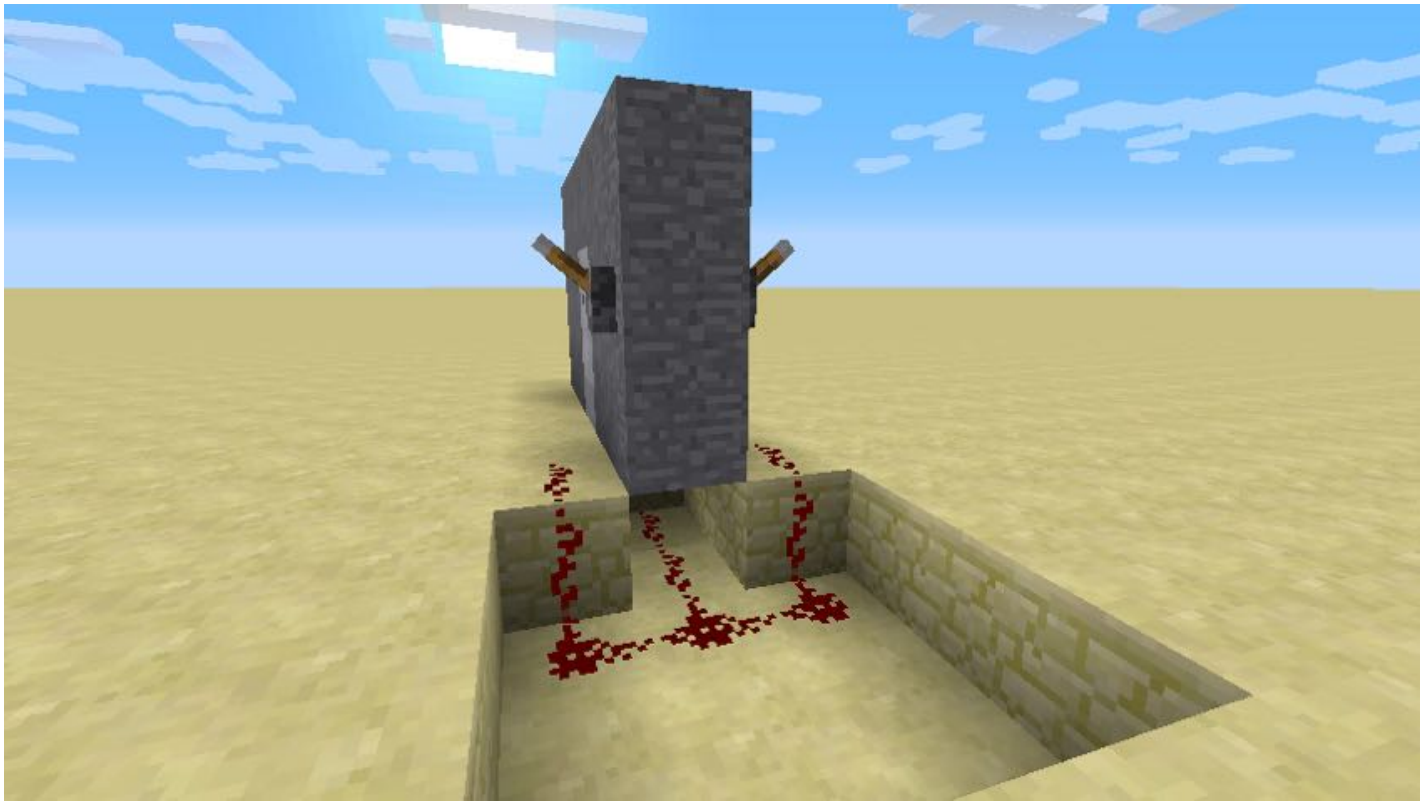


When all the items have flowed down into the hoppers, press the button to stop the water flow, and replant!

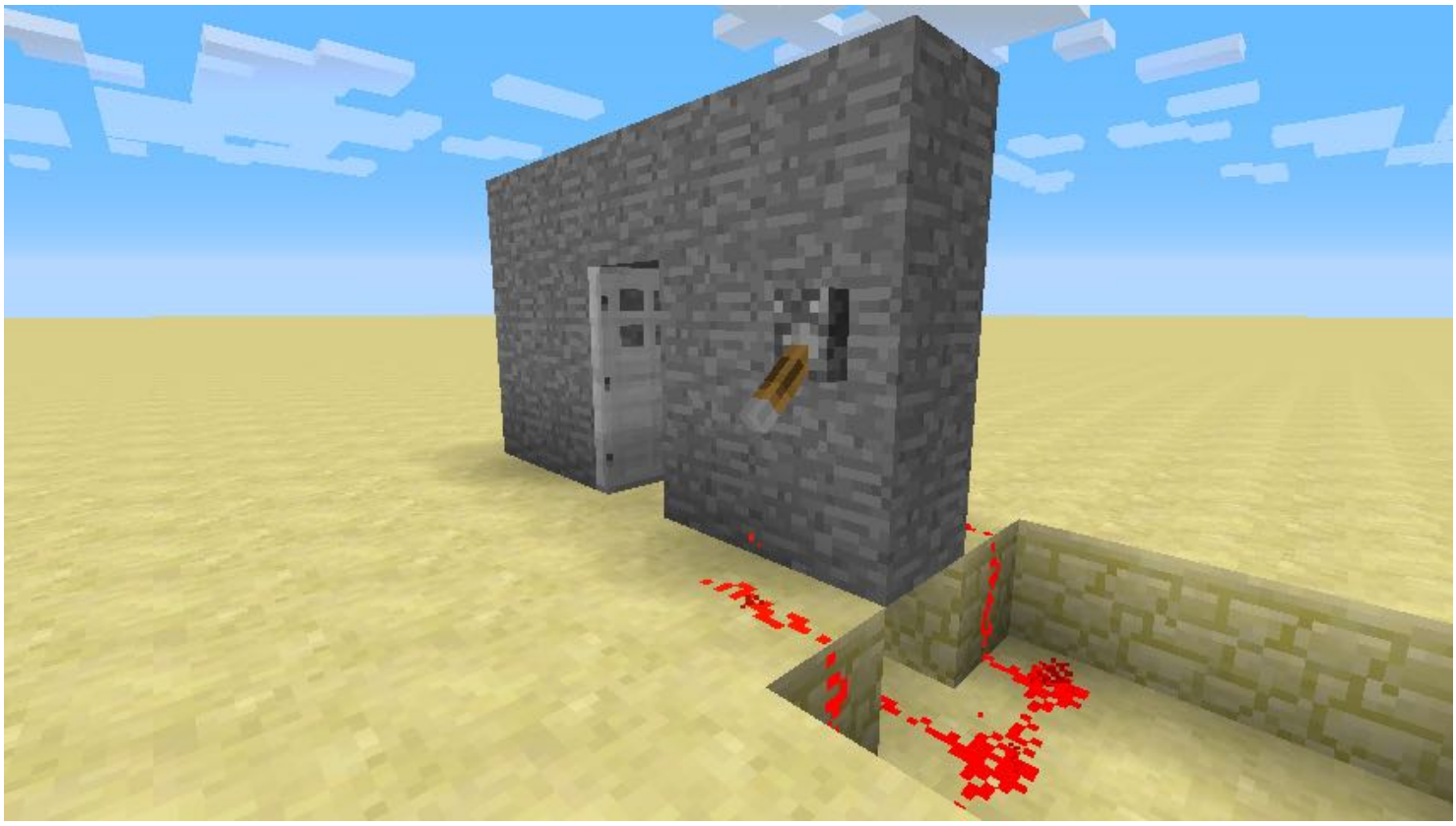
You can open doors using levers and redstone, and using an OR gate, you can open them from both sides.

1. Build a wall with a door in it. Place a lever on both sides of the door. Make sure the levers are at least one block away.

2. Create an OR gate around the edge of the wall, connecting the levers with redstone, and connecting the redstone current to the door underground.



Now that you have it built, you should be able to see that when either lever is pulled, the door opens. Compare this to the OR gate truth tables that you learned about in Section 3. Remember that when the lever is down, it equals 1, and when it is up, it equals 0.



You can use this in your own Minecraft world to hold open iron doors from either side.

Traps: Design and Execution

There are a few key features that every good trap must have. They must avoid suspicion and lure in victims with chests or rare items. They should have an entertaining or unique mechanism, since in the end all Minecraft traps are purely for entertainment. A trap has three main parts: the trap mechanism, the trigger, and the bait.

1. The first thing to consider when making a trap is the trap mechanism itself. How will you ensnare the player who has been snooping around your house? Obsidian can make for very strong traps as it requires diamond (and lots of time) to break; if the trapped player doesn't have a way to break obsidian, they're basically stuck! Water and lava traps can slow and even kill a player, but are significantly harder to conceal. For our trap we will be using stone and water. Stone is easier to obtain than obsidian but is still somewhat difficult to break, especially when the player is under the water we will be dropping on their head!

2. Triggers can take many forms. Tripwires and pressure plates are the most obvious, and you can place them in various ways to avoid detection (for example, only putting wood pressure plates on wood of matching color). However, buttons and levers can also be used. You can make it appear as if a button opens a specific door, when in reality it activates the trap. Levers can be used for dual purposes. For example, you may have 3 levers that act as a combination lock for a door. One combination will open the door, while incorrect combinations will trigger a trap for the trespasser. Our trap will use a trap chest as the trigger.

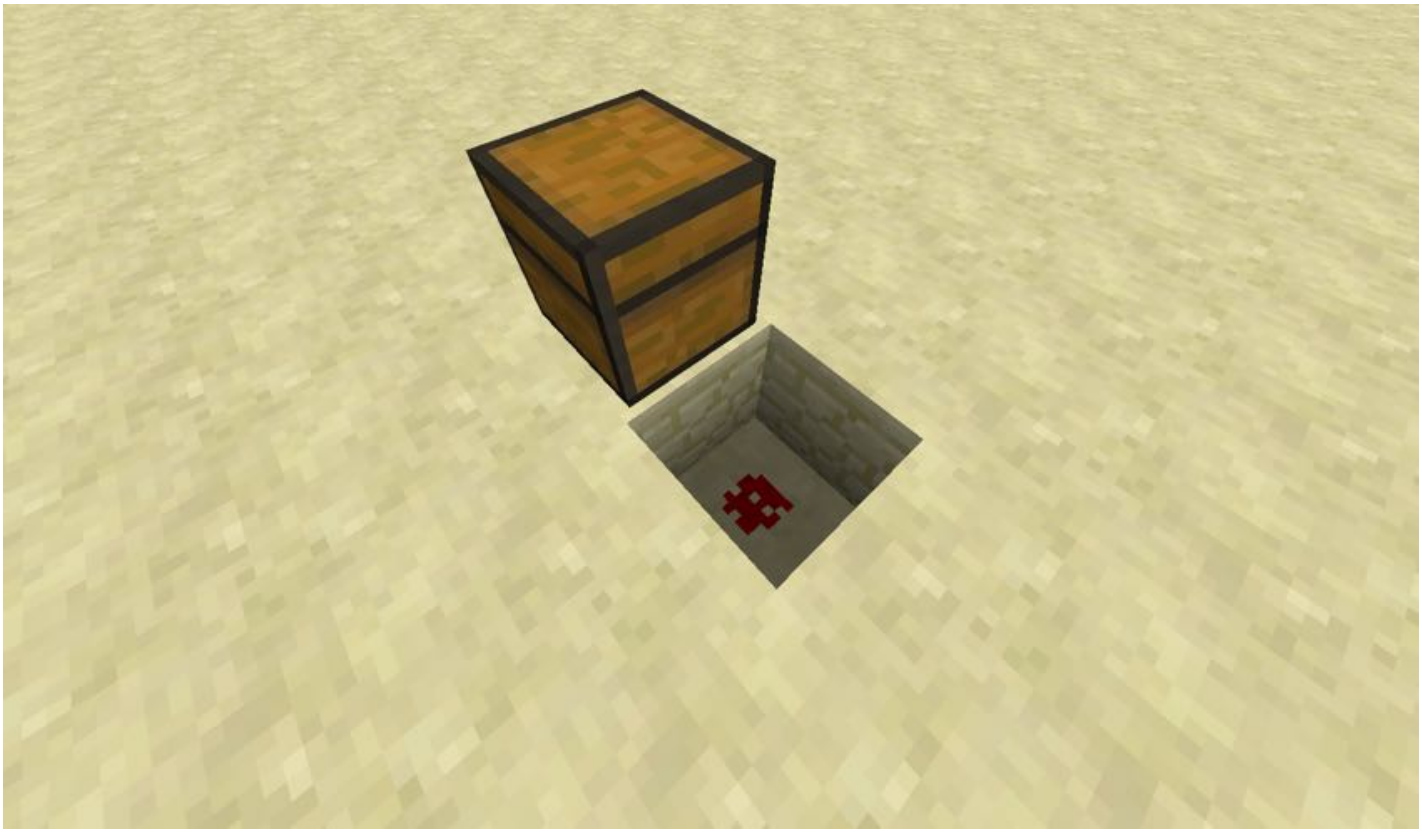
3. A trap chest filled with valuables is serving as the bait for this trap. However, interesting rooms, rare blocks, and even mine entrances can be used as the bait. As long as the object has something that a player would desire, it can serve as bait for a trap. Protective traps don't really have true "bait" as the trap is not supposed to be luring players in; rather, the item they are protecting is the bait in some sense. Our bait is the valuable items we will be placing inside of our trap chest.

Now that you've learned about the basics of traps, we're going to build an example trap using a trap chest, water, and pistons.

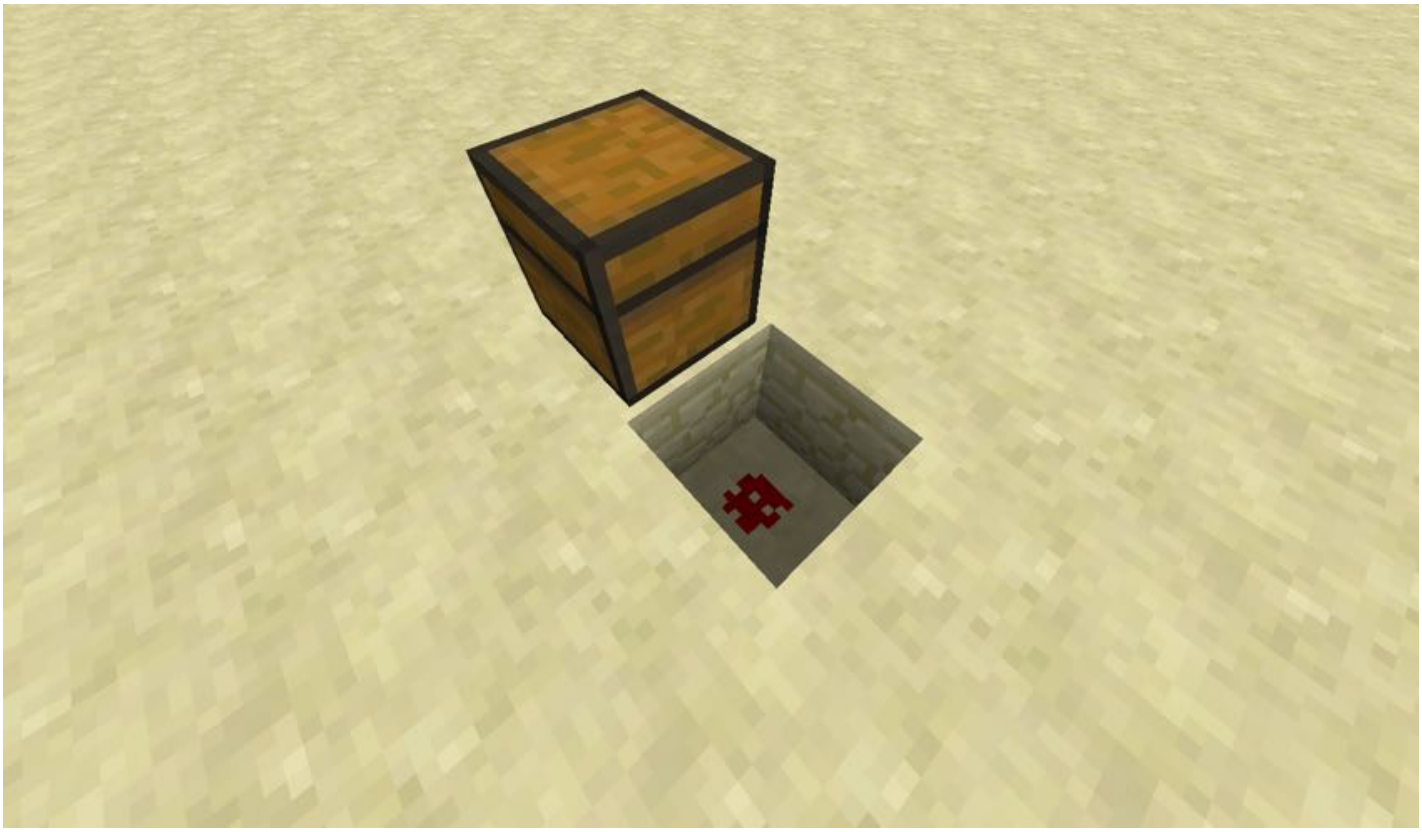
Trap Chests

Trap chests emit a redstone signal whose power is based on the number of people who are opening the chest. We can use this redstone signal to spring a trap designed for people opening the chest. It will open a flow of water when they open the chest, and it will not stop until a reset button has been pressed.

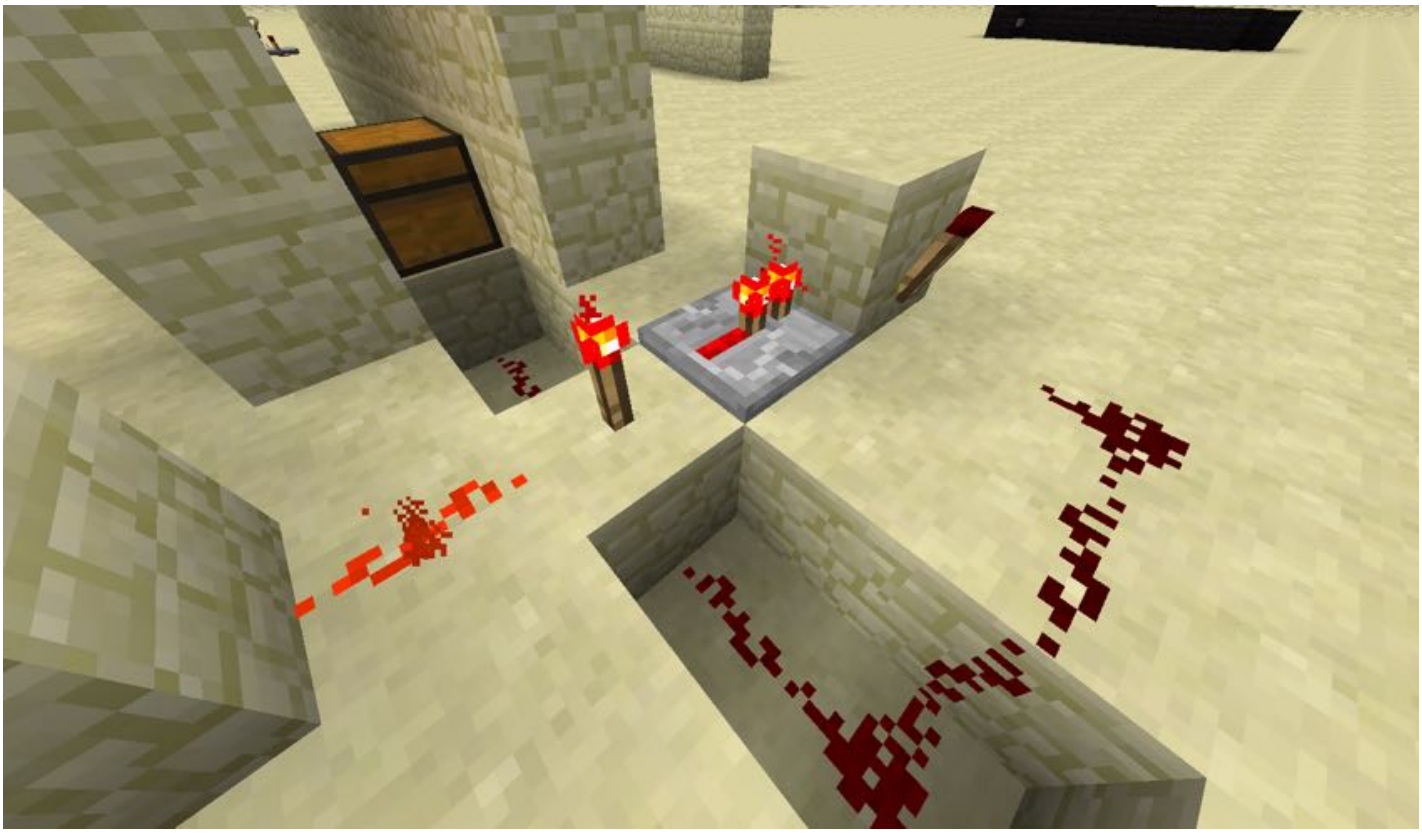
1. Place a trap chest somewhere. They’re made out of a regular chest and a tripwire. Dig a whole behind the chest and place one patch of redstone dust.

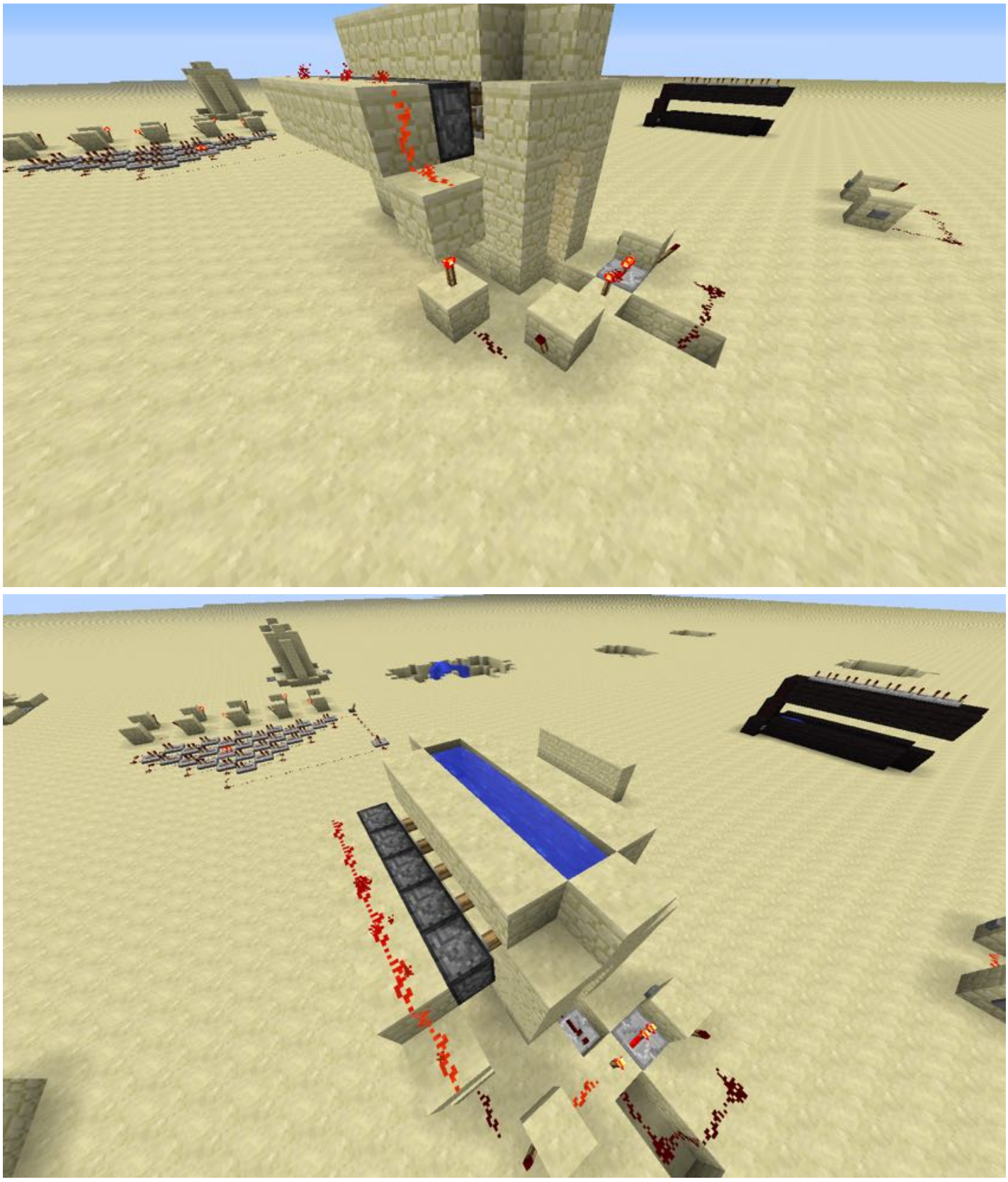


2. Dig a second hole and place a redstone repeater. Since the redstone power is proportional to the number of people opening the chest, this trap will trigger if just one person opens the chest.



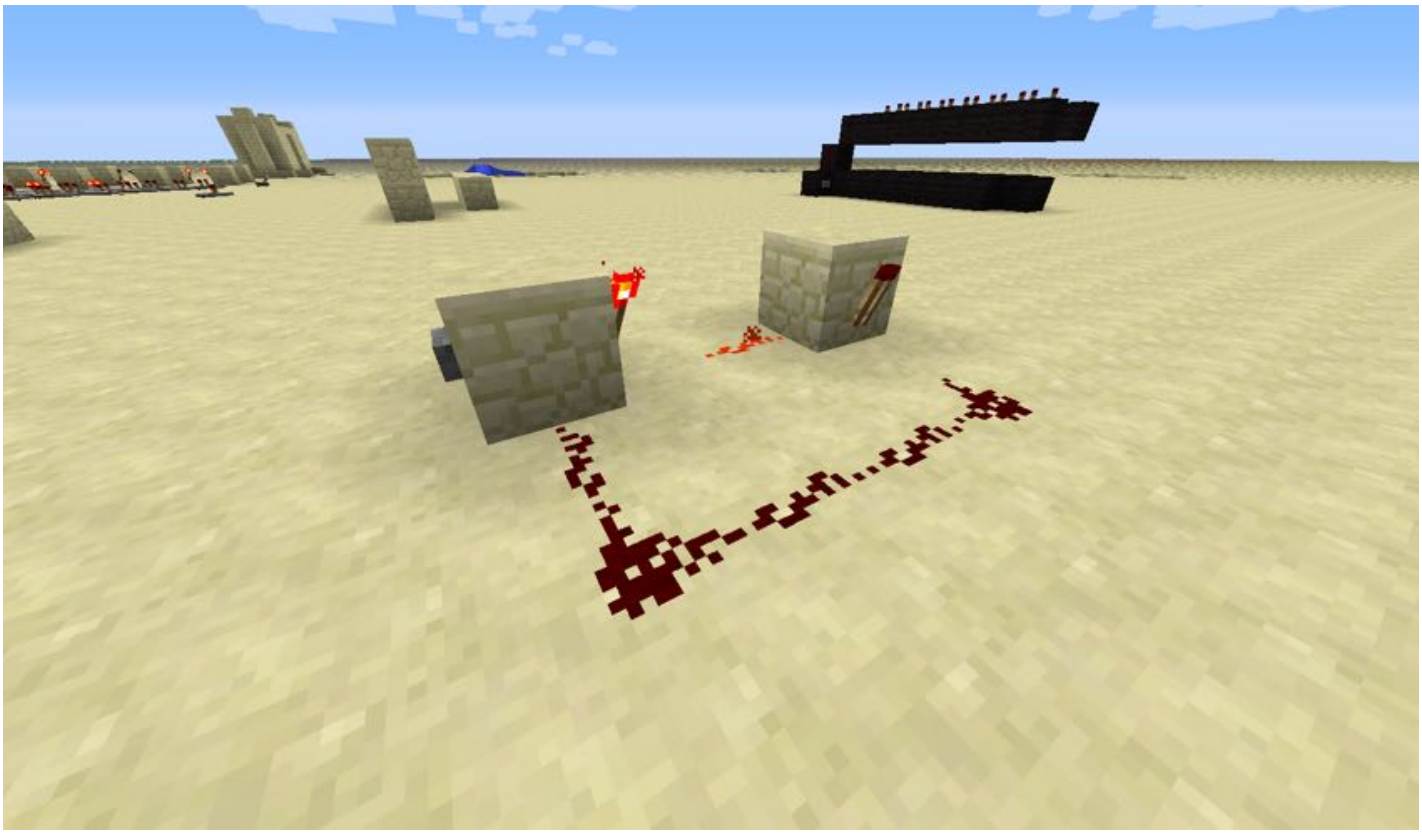
3. The wiring that will make the water gate open when the chest is opened. It's a bit tricky but further down there are additional pictures for you to follow.





4. This is the “one-way gate” that prevents the water from turning off even when the trap is closed. When the button on the left is pressed, the first torch is turned off and the second one turns on (since the first one is no longer disabling it). Now the second torch is disabling the first torch as its redstone leads into the block of the first one. Toggling the button (or chest in our case) will not alter the state of the

switch.

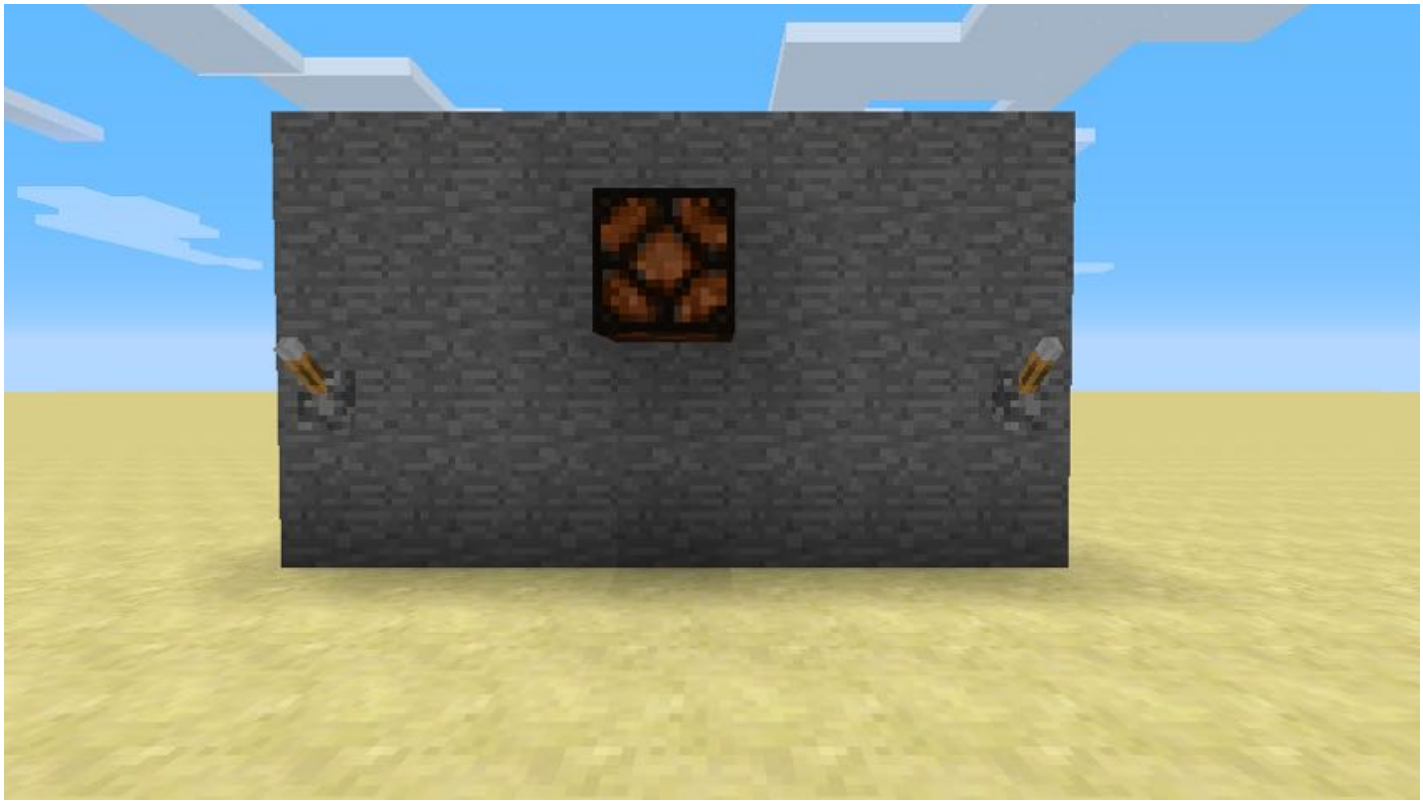


Try to think of some other ideas for traps that could be fun or useful. If we have time left over, feel free to experiment and share ideas with each other.

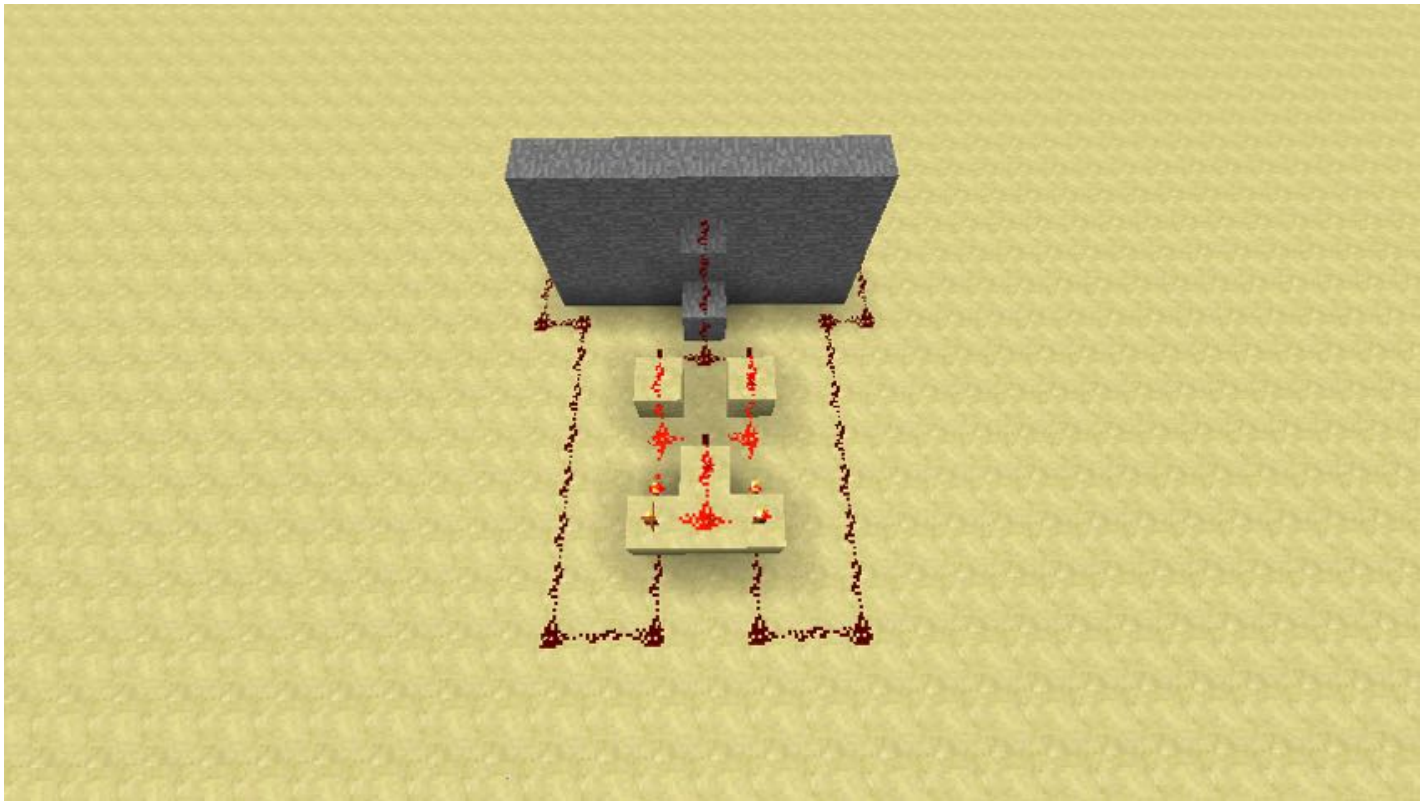
Light Switches

Using XOR gates, you can create a light that you can control with multiple switches. You might have lights that work this way in your actual home! Any light that has multiple switches to control it use a XOR gate. The most common use of these is lights in stairwells.

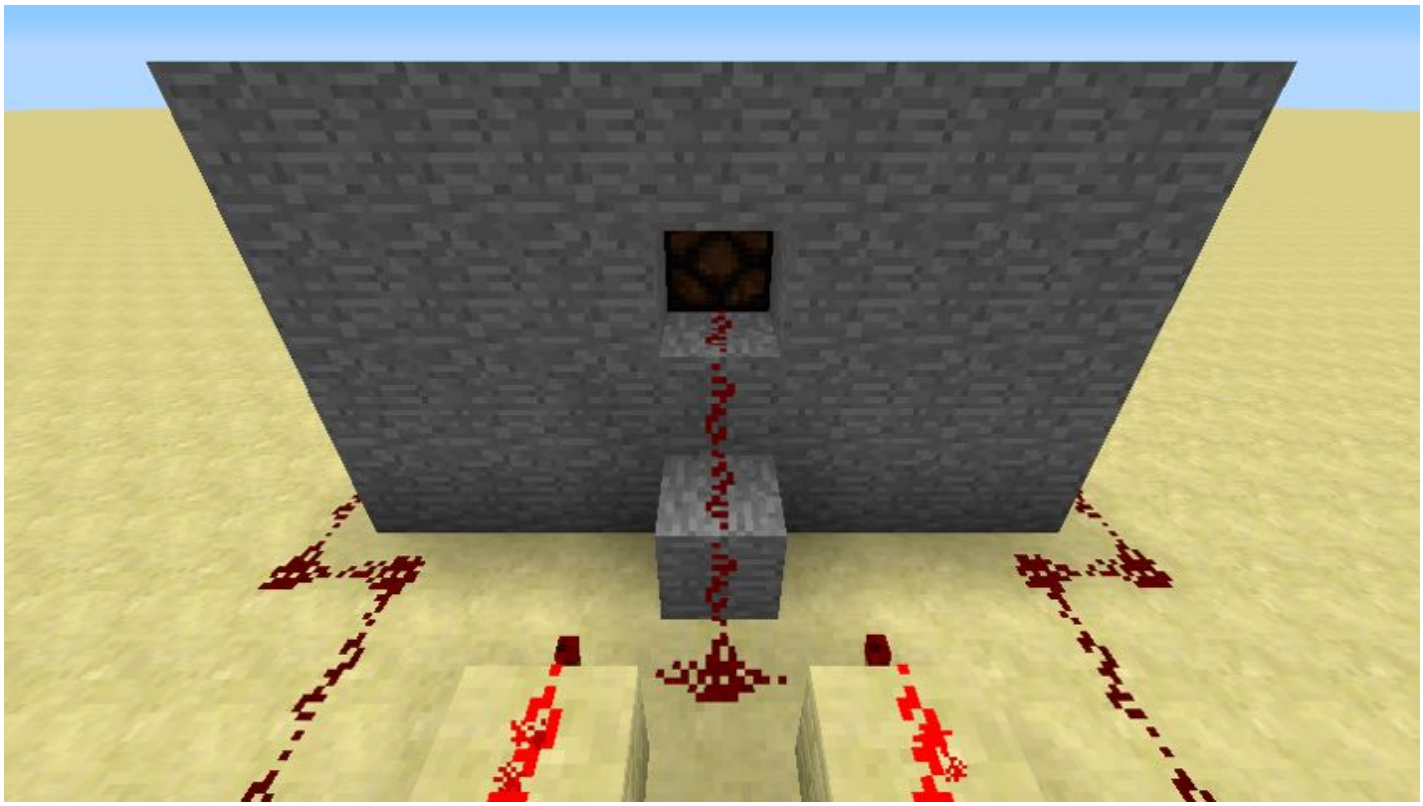
1. Build a wall and hang a redstone lamp on it. Put a lever on each end of the wall. The idea is that the levers can be as far away as you want, but this is a simple demonstration.



2. Create a XOR gate behind the wall, pointing the output towards the wall. Use the Section 3 reference picture if you need to. Remember that you don't need to place the levers, since the redstone signal is coming from the levers you placed on the front of the wall. Connect those to the XOR gate using redstone.



3. Lead the redstone signal coming from the output of the XOR gate to the lamp through the back of the wall, as shown.



4. You’re done! Flip one of the levers, and your lamp will turn on. Feel free to flip them as much as you want. The lamp will always be affected, no matter which way you flip either lever.