HOUR OF CODE 2022:
Escape Estate

SOLUTIONS GUIDE

This Educator Guide includes possible solutions for Hour of Code 2022: Escape Estate in both blocks and Python.

Hour of Code 2022: Escape Estate

So exactly how do you actually escape from the mansion?

Great question!

This solution guide will provide you with a comprehensive overview of sample solutions in both blocks and Python for each of the coding puzzles in every pathway of the Escape Estate.

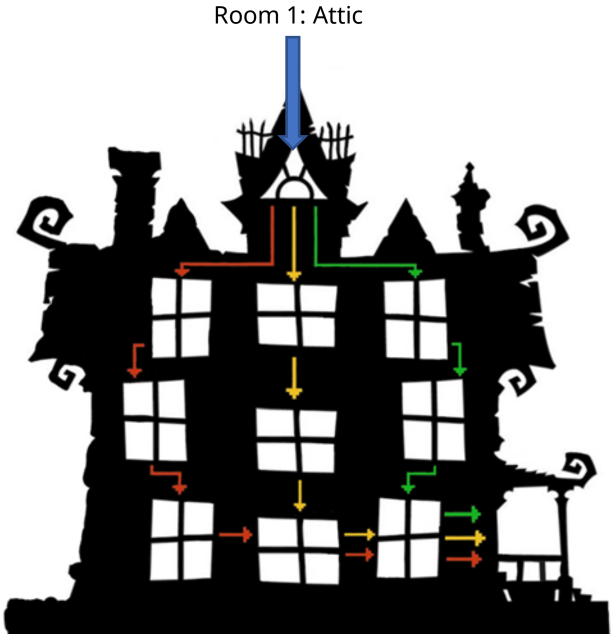
REMEMBER: These are just sample solutions. These are possibilities on how you *could* solve the coding puzzle, not how to *have* to solve the coding puzzle. These solutions will come in handy if a player gets stuck or if you need some assistance yourself! These solutions are **NOT** comprehensive or exhaustive of all plausible solutions.

Most of these coding puzzles have numerous ways to solve the puzzle. The game will consider the puzzle to be “solved” if they provide any coding solution that can actually solve the puzzle. It does not track for efficiency or specific use of blocks/lines of code. The game will allow the player to continue once they have provided an operational code to solve the puzzle.

This guide will provide the solutions by pathway so it’s easier to follow along and track your students’ progress.

Have fun and good luck!

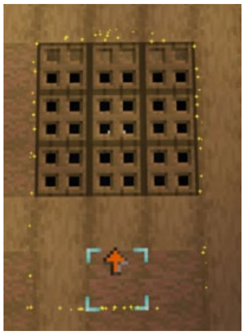
SOLUTIONS



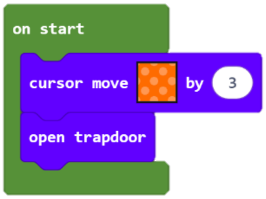
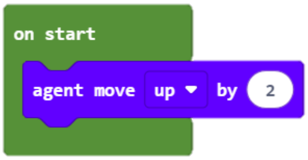
Room 1: Attic

All players will start in the attic. While they are in the attic, they will select their pathway.

ATTIC – Coding Puzzle 1: Rescue the Agent



The Agent is trapped! Can you help rescue the Agent?
Part 1: You can rescue the Agent by moving the cursor 3 blocks forward and opening the trapdoor.
Part 2: You need to help the Agent by moving the Agent 2 blocks up.

Blocks	
Coding Snippet: Part 1	Coding Snippet: Part 2
 on start cursor move 🟧 by 3 open trapdoor	 on start agent move up ▾ by 2

Python

```
move_cursor("orange")
move_cursor("orange")
move_cursor("orange")
open_trapdoor()
```

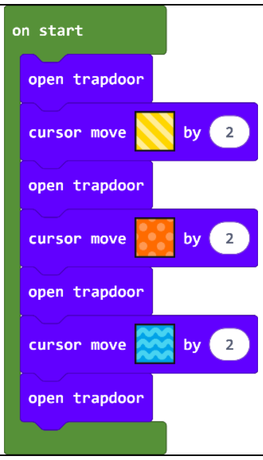
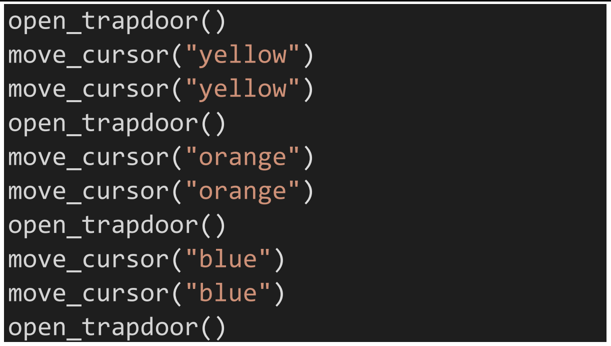
```
agent.move("up")
agent.move("up")
```

Upon completion of this puzzle, the **GREEN** pathway door will open.

ATTIC – Coding Puzzle 2: Ceiling Trapdoors

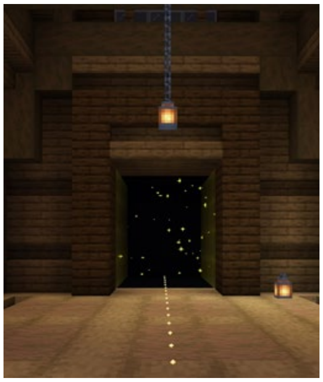
We need to open the skylight to help the Agent gain more battery power. If we open up the four trapdoors in the ceiling, we will be able to charge up the Agent even more!

Blocks	Python

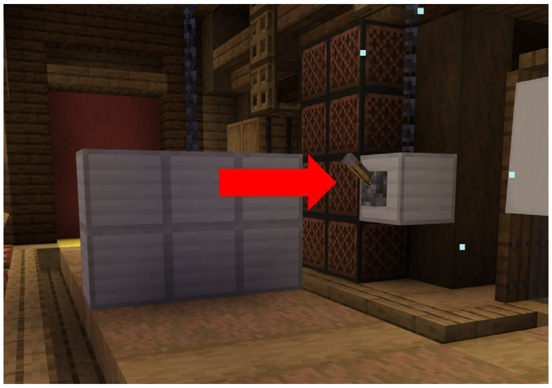

on start
open trapdoor
cursor move 🟨 by 2
open trapdoor
cursor move 🟧 by 2
open trapdoor
cursor move 🟦 by 2
open trapdoor
	
	`open_trapdoor()`
	`move_cursor("yellow")`
	`move_cursor("yellow")`
	`open_trapdoor()`
	`move_cursor("orange")`
	`move_cursor("orange")`
	`open_trapdoor()`
	`move_cursor("blue")`
	`move_cursor("blue")`
	`open_trapdoor()`

REMEMBER: There are MULTIPLE ways to solve this puzzle. Players may go in different directions—that is okay! This also serves as a great teachable moment. In coding, there are often multiple algorithms that can solve the same problem/task. Although some ways may be more effective or efficient, the game will accept any solution that successfully solves the problem or completes the task.

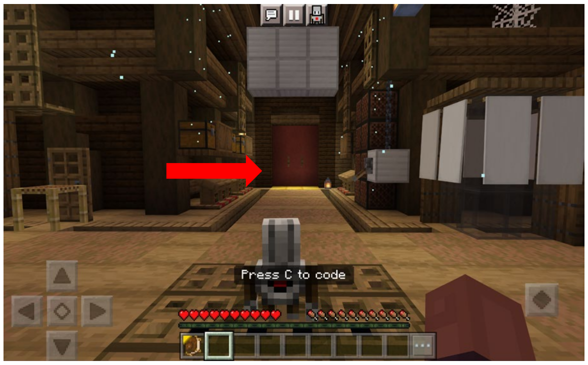
Upon completion of this puzzle, the YELLOW pathway door will open.



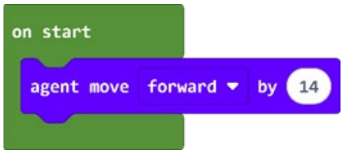
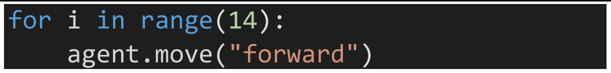
ATTIC – Coding Puzzle 3: Gold Blocks
First, you will need to switch the lever. When the lever is switched, this brick wall will lift and provide a clear path to the gold blocks in front of the red door.



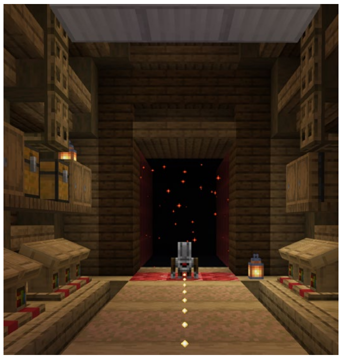
Then, you will need to code the Agent to move and end up on the gold blocks.



Blocks	Python

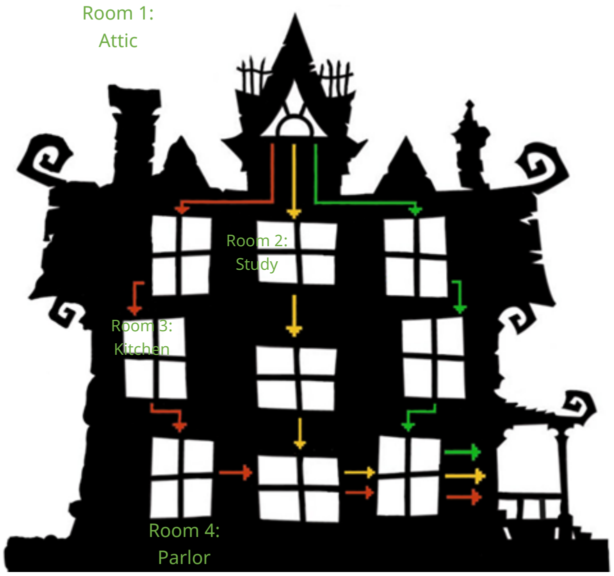

on start
agent move forward ▾ by 14
	
	`for i in range(14):`
	`    agent.move("forward")`

Upon completion of this puzzle, the **RED** pathway door will open.



Now, all players MUST select a door to start their pathway (if they have not already selected their desired pathway).

SOLUTIONS: GREEN PATHWAY



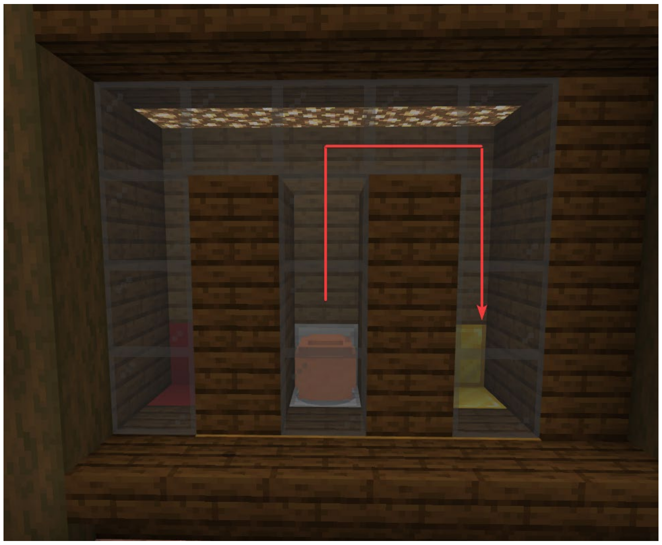
Room 2: Study

In the study, there are 3 coding puzzles:

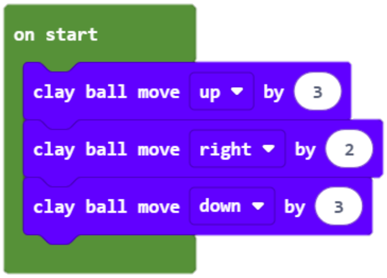
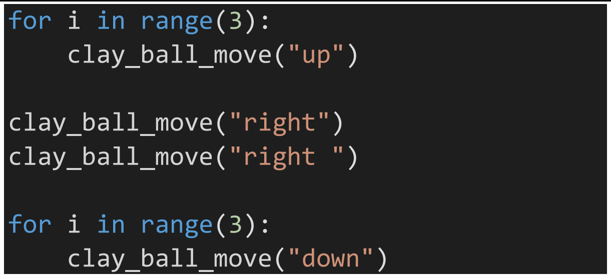
- Move the Clay Ball Puzzle
- Mirror, Mirror Puzzle
- Speedy Knight Puzzle

STUDY – Coding Puzzle 1: Move the Clay Ball

You will need to move the clay ball over to the gold block.

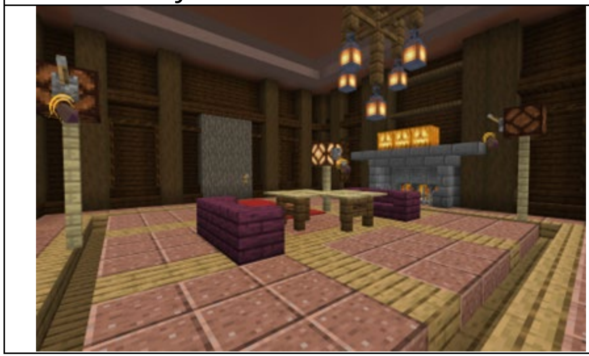
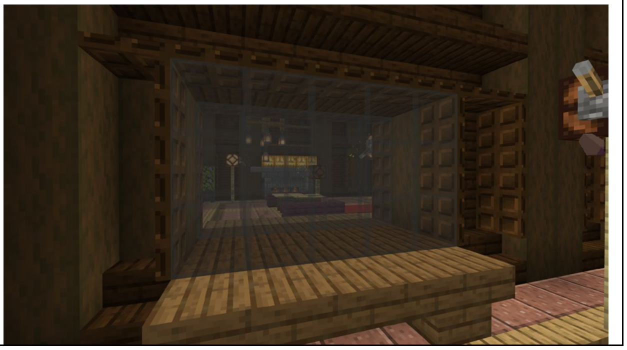


Blocks	Python

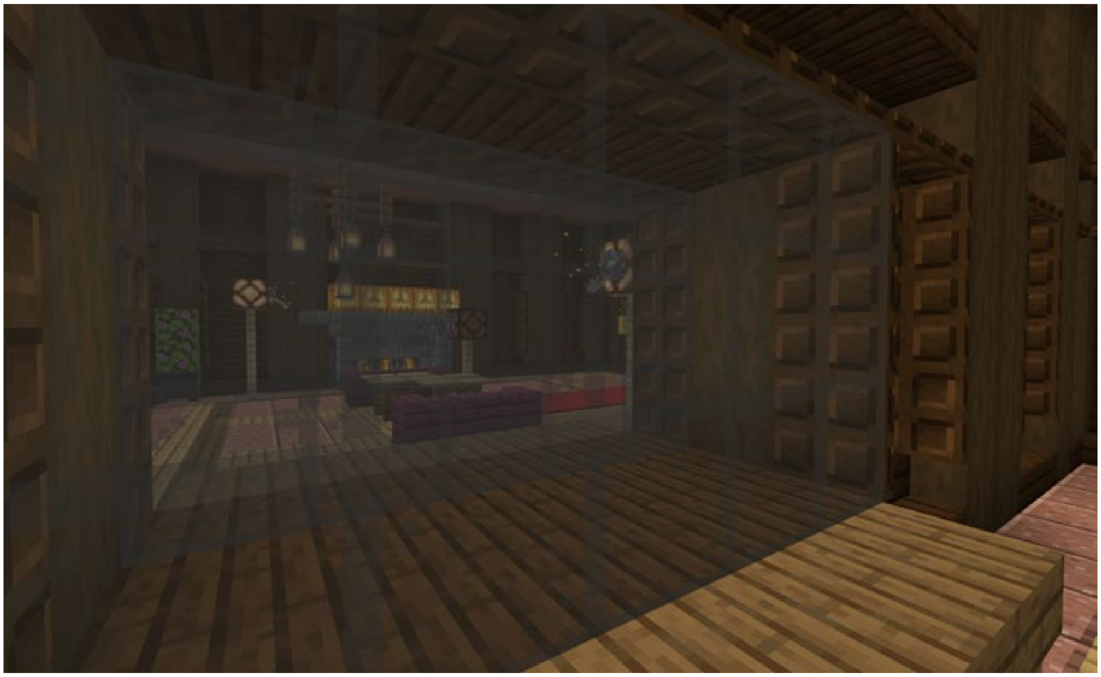

on start
clay ball move up by 3
clay ball move right by 2
clay ball move down by 3
	
	`for i in range(3):`
	`    clay_ball_move("up")`
	
	`clay_ball_move("right")`
	`clay_ball_move("right ")`
	
	`for i in range(3):`
	`    clay_ball_move("down")`

STUDY – Coding Puzzle 2: Mirror, Mirror

A mirror has suddenly appeared in the room! Let's look through this "mirror" to see if we can find any similarities—or differences!

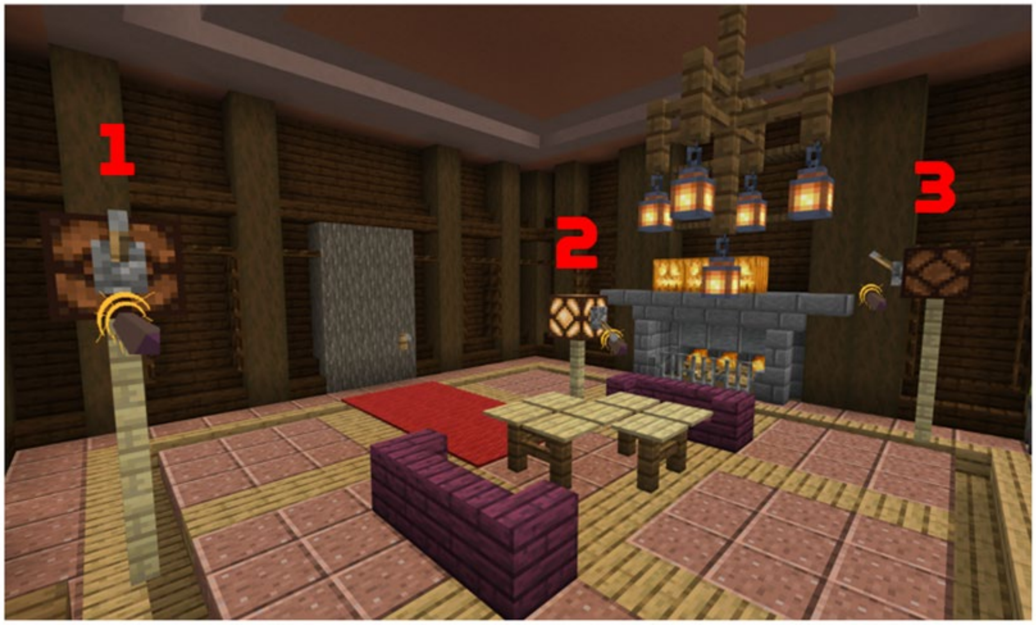
Actual Study	Study in the Mirror
	

Part 1: Make It Match – Lights



The lights need to be reversed to match the lights in the mirror.

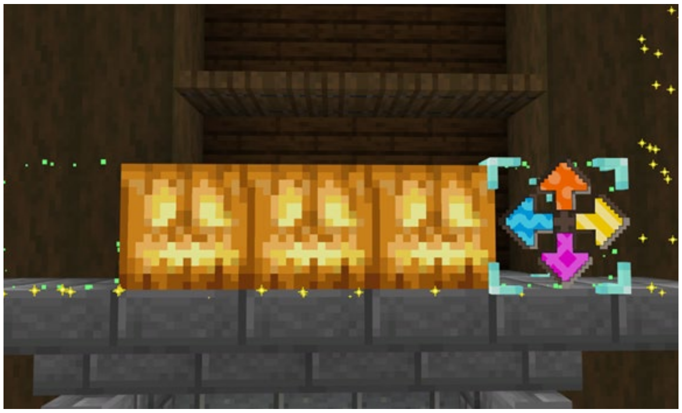
- Light 1 needs to be turned ON.
- Light 2 needs to be turned OFF.
- Light 3 needs to be turned ON.



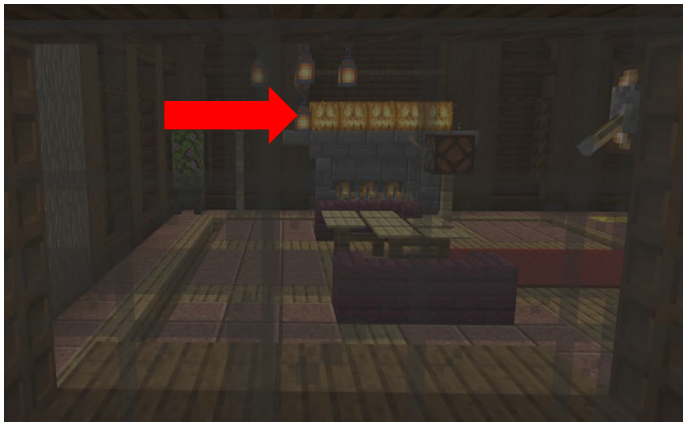
Part 2: Make it Match – Pumpkins

The pumpkins above the fireplace in the mirror is somehow different than the ones in this room. Try making them match.

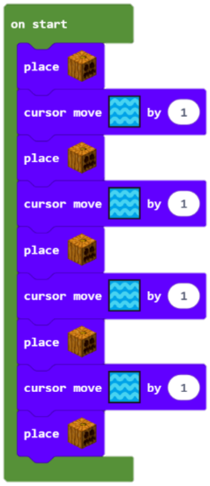
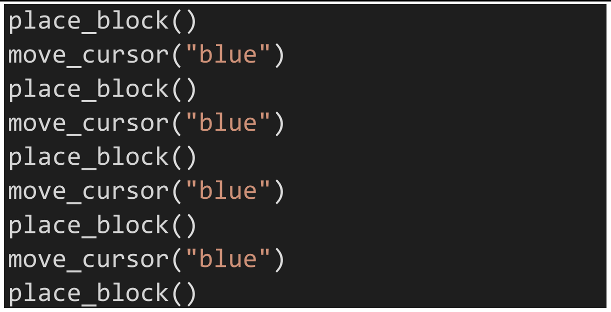
Current Image



Mirror Image



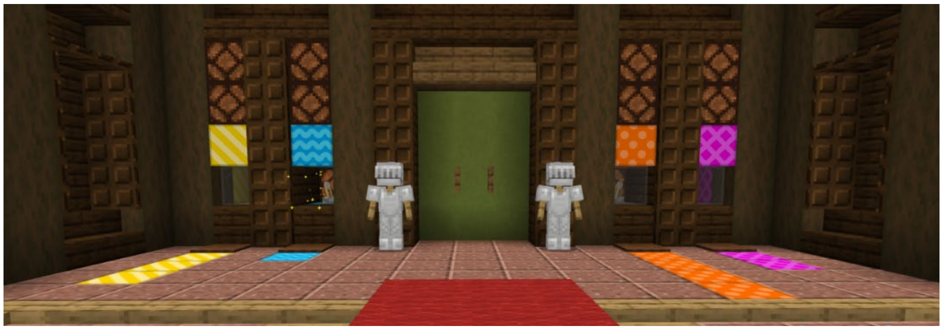
Blocks	Python

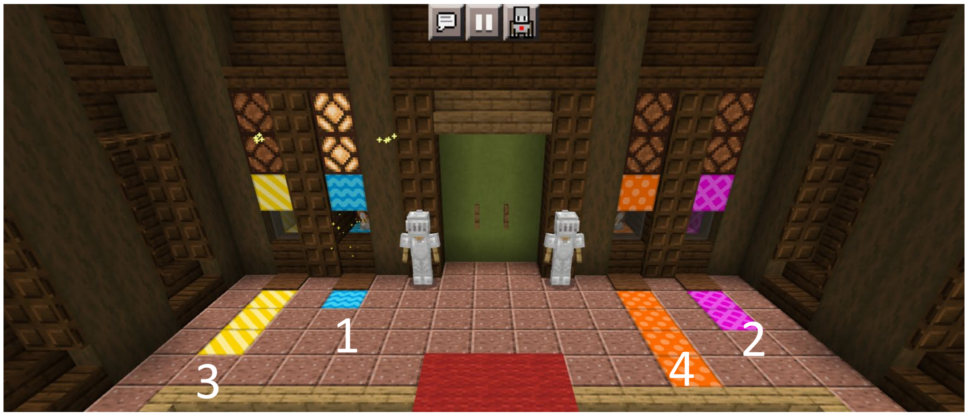

on start
place
cursor move by 1
place
cursor move by 1
place
cursor move by 1
place
cursor move by 1
place
	
	place_block()
	move_cursor("blue")
	place_block()
	move_cursor("blue")
	place_block()
	move_cursor("blue")
	place_block()
	move_cursor("blue")
	place_block()

STUDY – Coding Puzzle 3: Speedy Knight

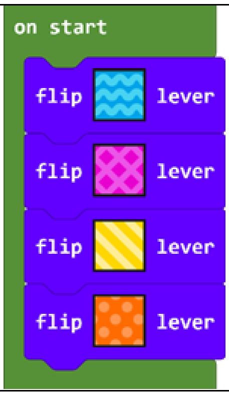
We need to flip all four levers to open the door, but we are not quick enough to do it manually. Let’s get the Speedy Knight to help us! We need to figure out the order the levers need to be flipped to program the Speedy Knight to do it for us!

The number of colored carpet squares will explain the order for this task. The flashing glowstone above the colors will also indicate the order of the levers.





Blocks	Python


on start
flip 🟦 lever
flip 🟪 lever
flip 🟨 lever
flip 🟧 lever
	move_knight("blue")
	move_knight("magenta")
	move_knight("yellow")
	move_knight("orange")

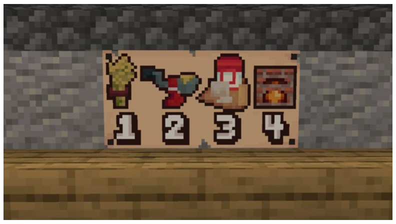
Room 3: Kitchen

In the kitchen, there are 3 coding puzzles:

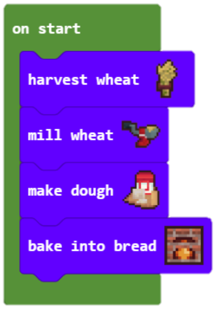
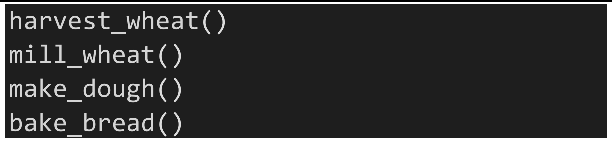
- Making Bread Recipe Puzzle
- Hungry House Puzzle
- Mixed-Up Meals Puzzle

KITCHEN – Coding Puzzle 1: Making Bread Recipe

The Zombie Chef is making bread. Use the recipe on the wall to guide the creation of your algorithm to make the bread.

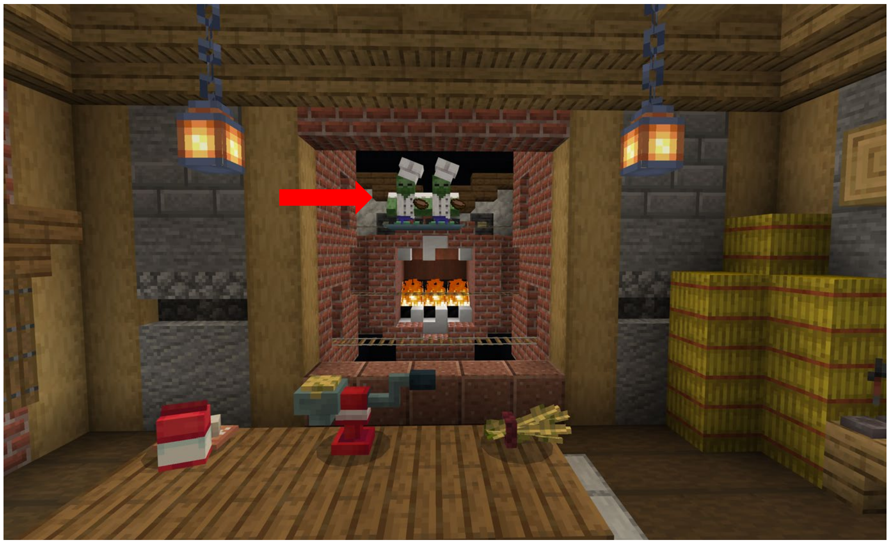


Blocks	Python

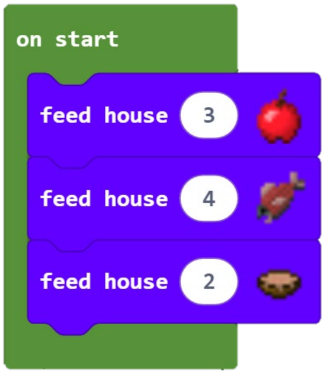
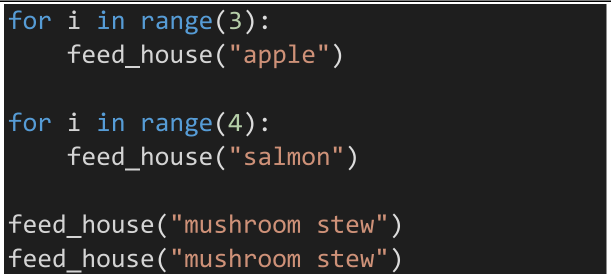

on start
harvest wheat
mill wheat
make dough
bake into bread
	
	`harvest_wheat()`
	`mill_wheat()`
	`make_dough()`
	`bake_bread()`

KITCHEN – Coding Puzzle 2: Hungry House

The House is hungry! What could be better than apples, salmon, and mushroom stew?! Look at the moving conveyor belt behind the table. The number of ingredients will be represented by the other zombie chefs passing by on the conveyor belt. The conveyor belt runs numerous times so don't worry if you miss it the first time!



Blocks	Python

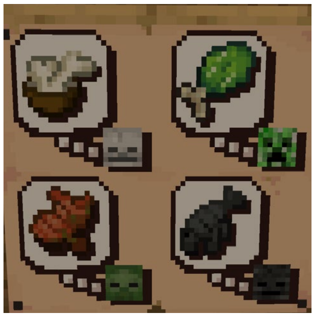
 on start
feed house 3
feed house 4
feed house 2
	 for i in range(3):
	    feed_house("apple")
	
	for i in range(4):
	    feed_house("salmon")
	
	feed_house("mushroom stew")
	feed_house("mushroom stew")

KITCHEN – Coding Puzzle 3: Mixed-Up Meals

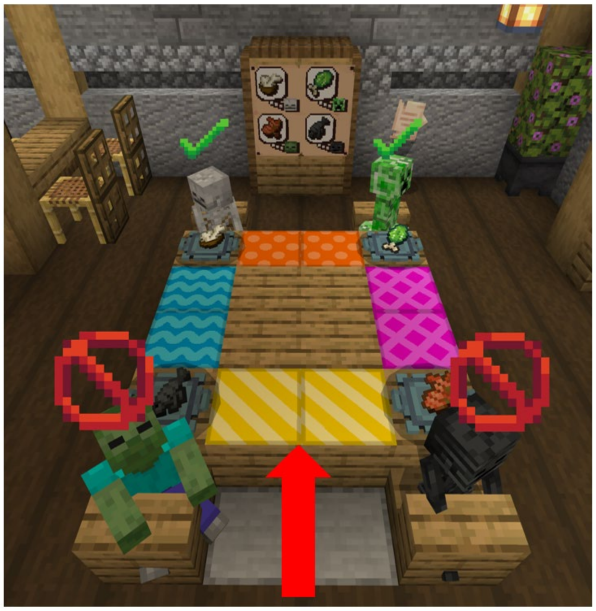
The Zombie Chef mixed up all the meals. Search the kitchen for the menu and make sure everyone ends up with the correct meals.

In this puzzle, you must swap the menu items until each mob gets their correct meal based on the menu.

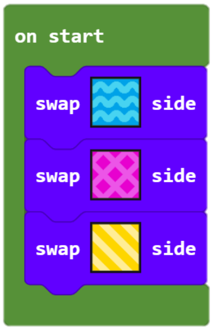
Here is what each of the mobs should have:

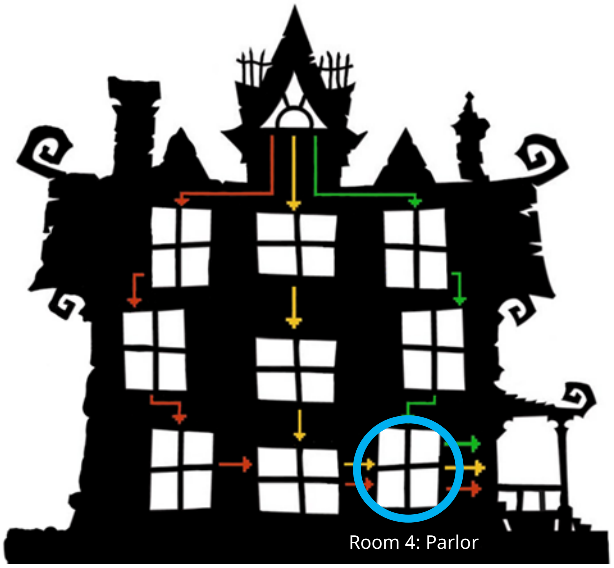


When you run the initial code, it will swap the blue side first, and then the magenta side. We will need to make one more swap on the yellow side to correct all of the meals.



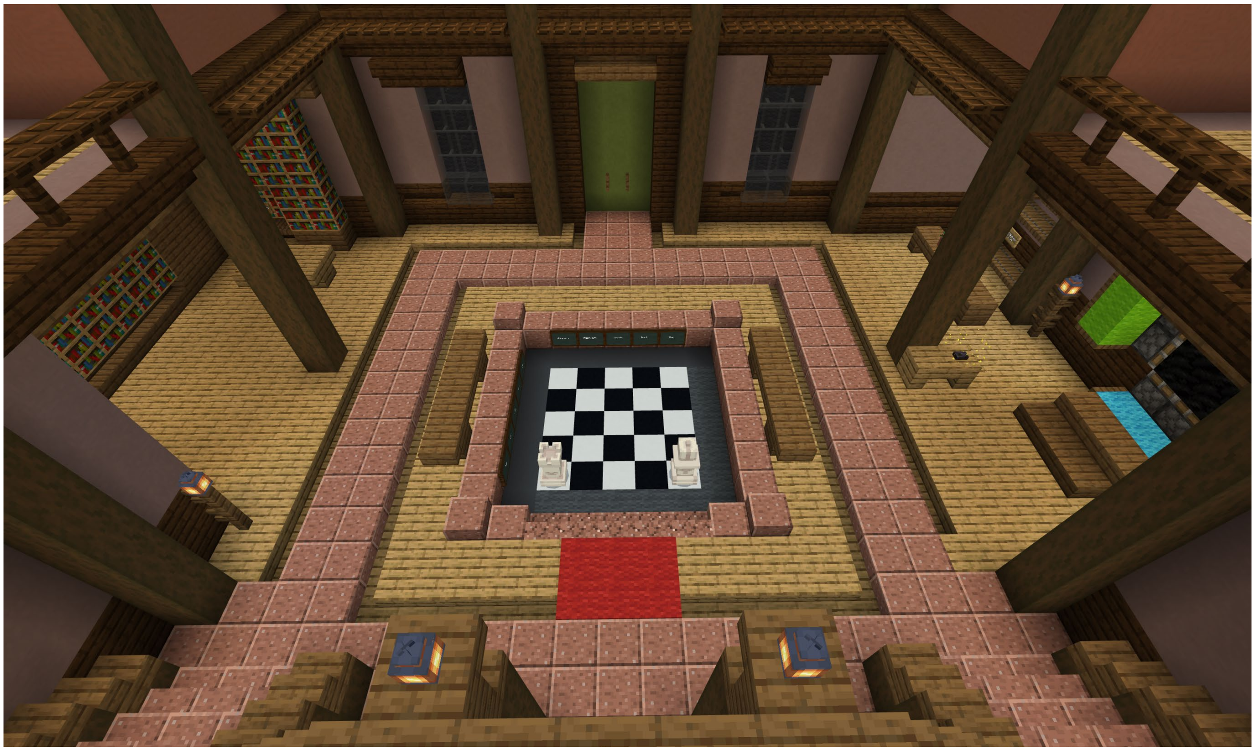
Blocks	Python


on start
swap 🟦 side
swap 🟪 side
swap 🟨 side
	swap_plates("blue")
	swap_plates("magenta")
	swap_plates("yellow")



Room 4: Parlor

All players will end their Escape Estate adventure in the parlor. This is the last room for all players (regardless of which pathway they took).

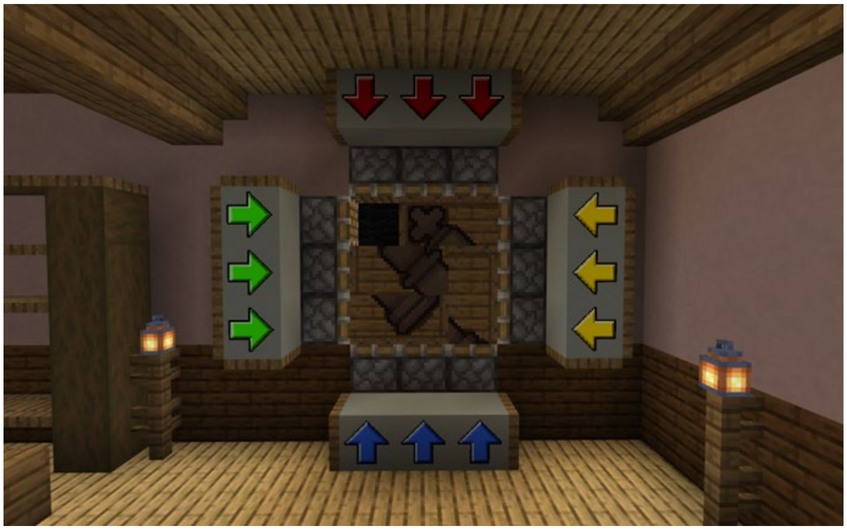


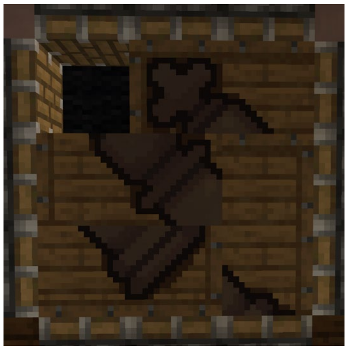
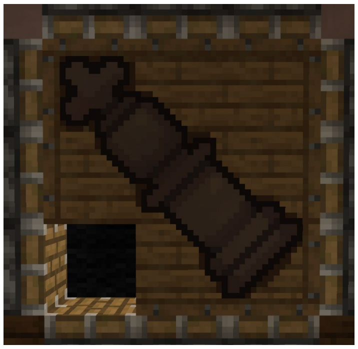
In the parlor, there are 3 coding puzzles:

- King’s Scrambled Picture
- Save the Queen
- Wrong Birthdays

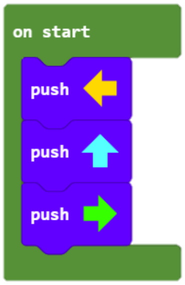
PARLOR – Coding Puzzle 1: King’s Scrambled Picture

The picture has somehow gotten scrambled! Let’s push some blocks around to see if we can fix it! The colors will signify which way the blocks will be pushed.



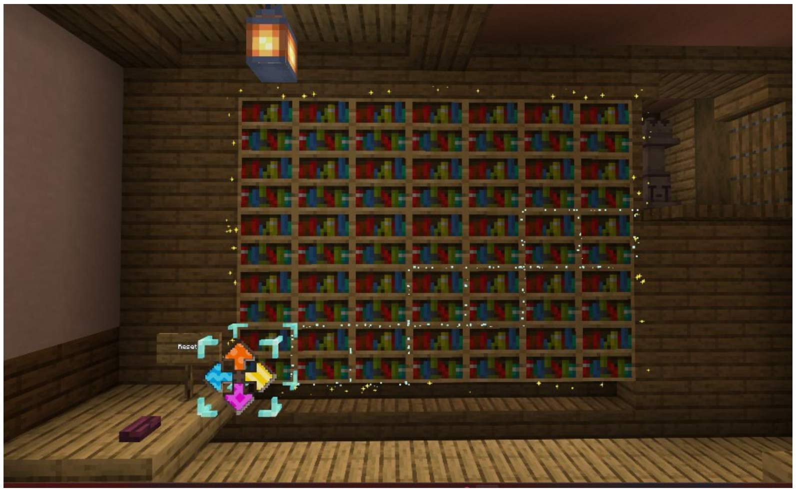
BEFORE	AFTER
	

Blocks	Python


on start
push
push
push
	`push_pieces("left")`
	`push_pieces("up")`
	`push_pieces("right")`

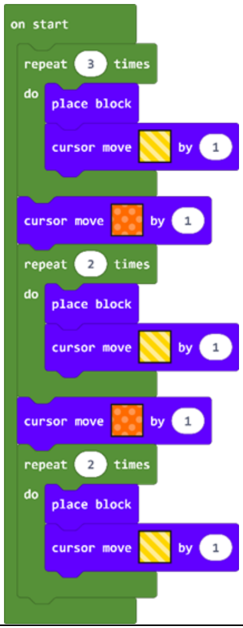
PARLOR – Coding Puzzle 2: Save the Queen

There is something at the top of the bookcase! Create something to get up there!



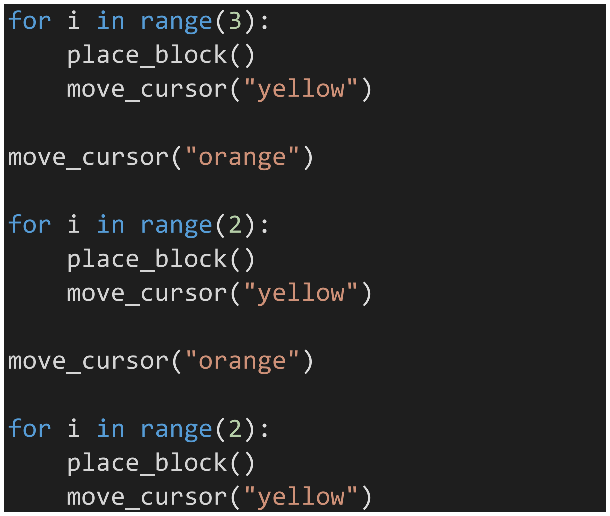
Blocks

Code Snippet



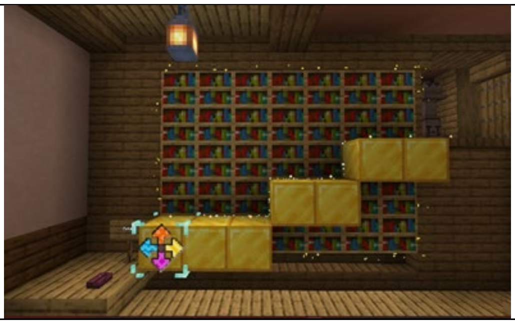
```
on start
  repeat 3 times
  do  place block
      cursor move 🟨 by 1
  cursor move 🟧 by 1
  repeat 2 times
  do  place block
      cursor move 🟨 by 1
  cursor move 🟧 by 1
  repeat 2 times
  do  place block
      cursor move 🟨 by 1
```

Python



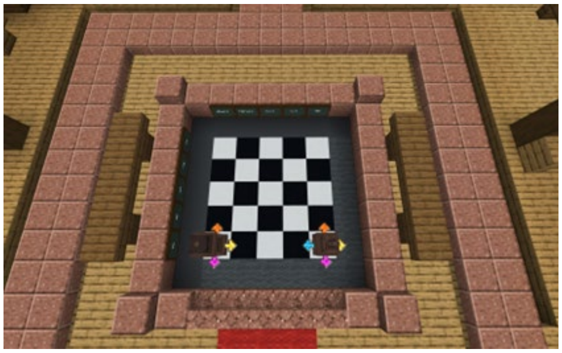
```python
for i in range(3):
    place_block()
    move_cursor("yellow")

move_cursor("orange")

for i in range(2):
    place_block()
    move_cursor("yellow")

move_cursor("orange")

for i in range(2):
    place_block()
    move_cursor("yellow")
```

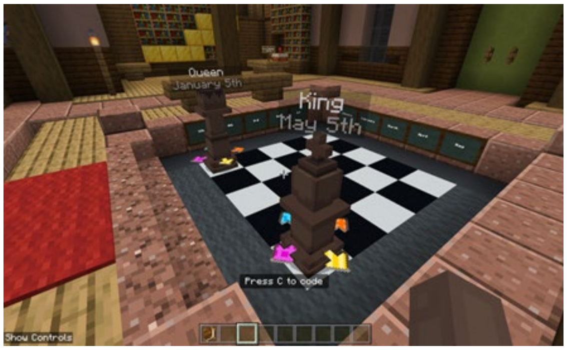
RESULT



PARLOR – Coding Puzzle 3: Messed Up Birthdays

Although our king and queen have been freed, the door still isn't opening! They must be in the wrong positions on the chessboard. Move them into the correct positions so you can finally escape this place!

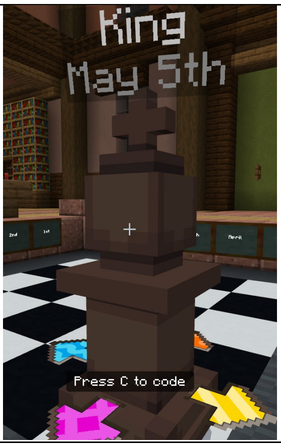




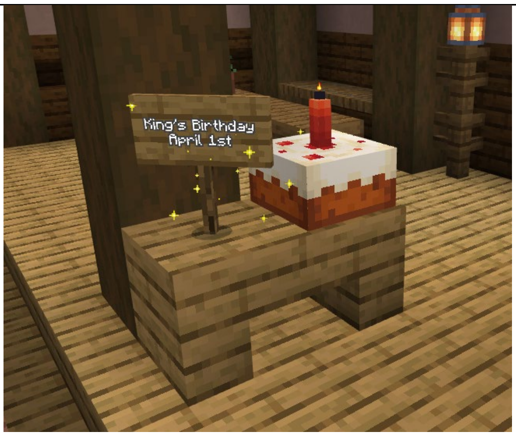
The months and days are listed on the chalkboards inside the chessboard. You will need to need to move the pieces from their current position to their new position (i.e., birthday).

King

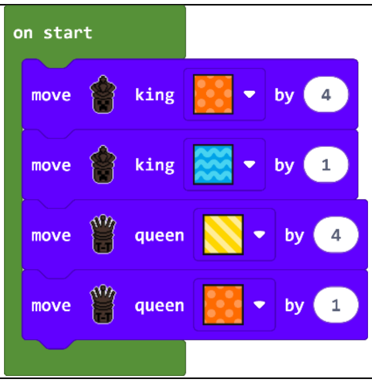
<table>
<tr><td>Current Position</td><td>New Position</td></tr>
<tr><td>



</td><td>

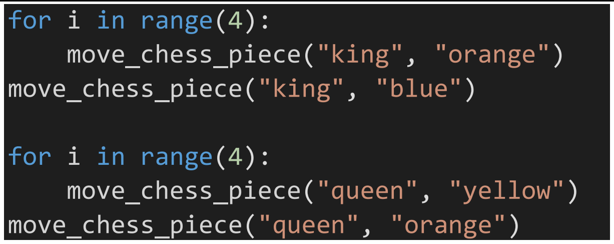


</td></tr>
<tr><td>Blocks</td><td>Python</td></tr>
<tr><td>



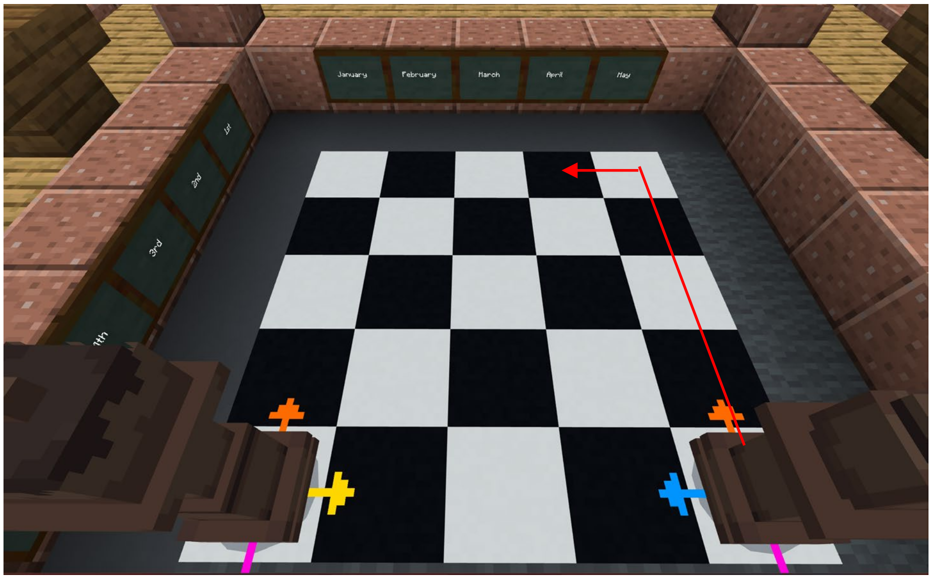
```
on start
  move king 🟧 by 4
  move king 🟦 by 1
  move queen 🟨 by 4
  move queen 🟧 by 1
```

</td><td>

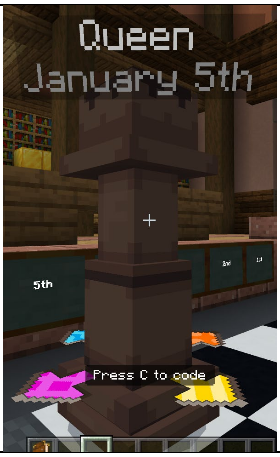
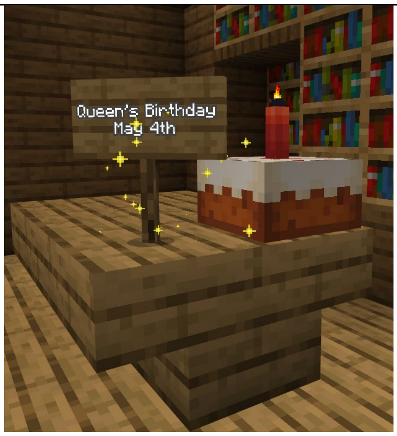


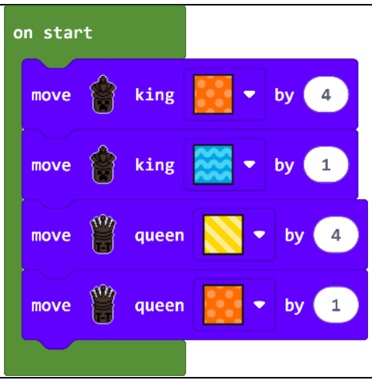
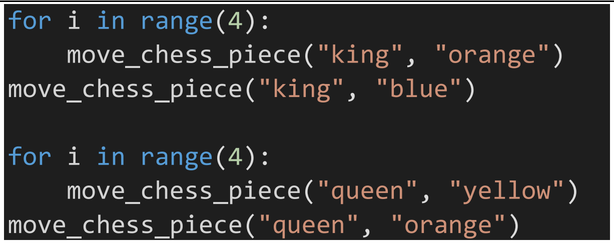
```
for i in range(4):
    move_chess_piece("king", "orange")
move_chess_piece("king", "blue")

for i in range(4):
    move_chess_piece("queen", "yellow")
move_chess_piece("queen", "orange")
```

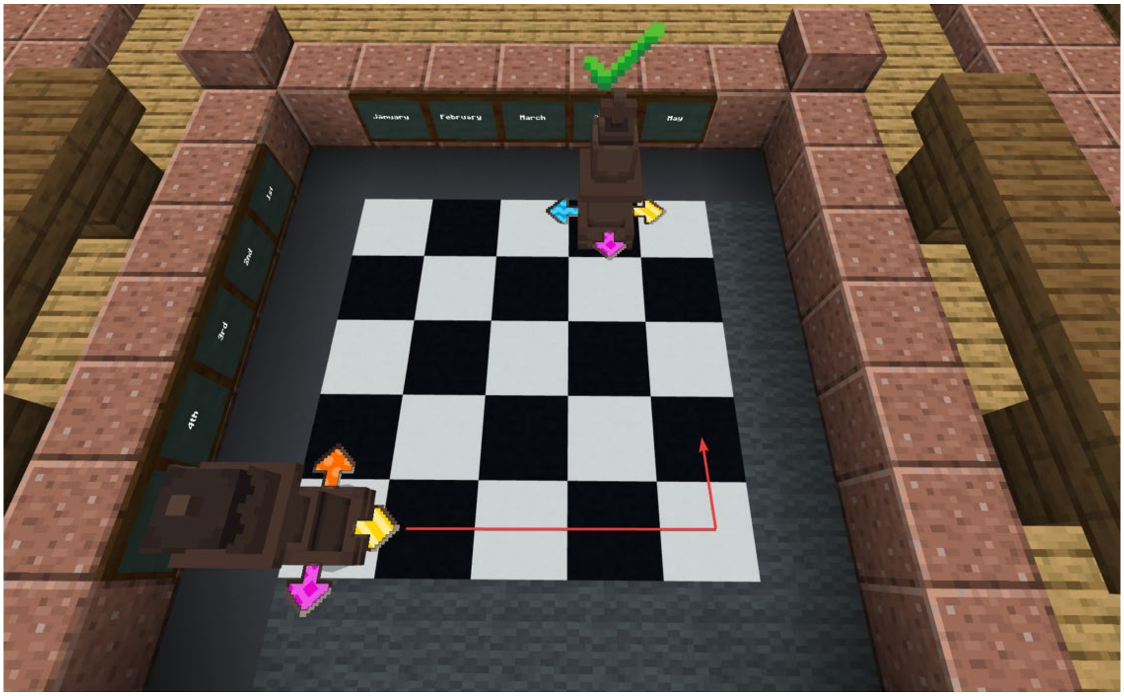
</td></tr>
</table>



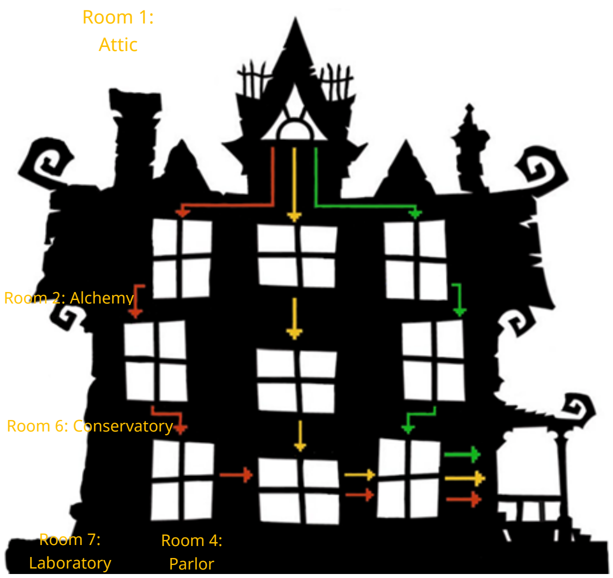
Queen

Current Position	New Position
 Queen January 5th 5th 2nd 1st Press C to code	
Blocks	**Python**

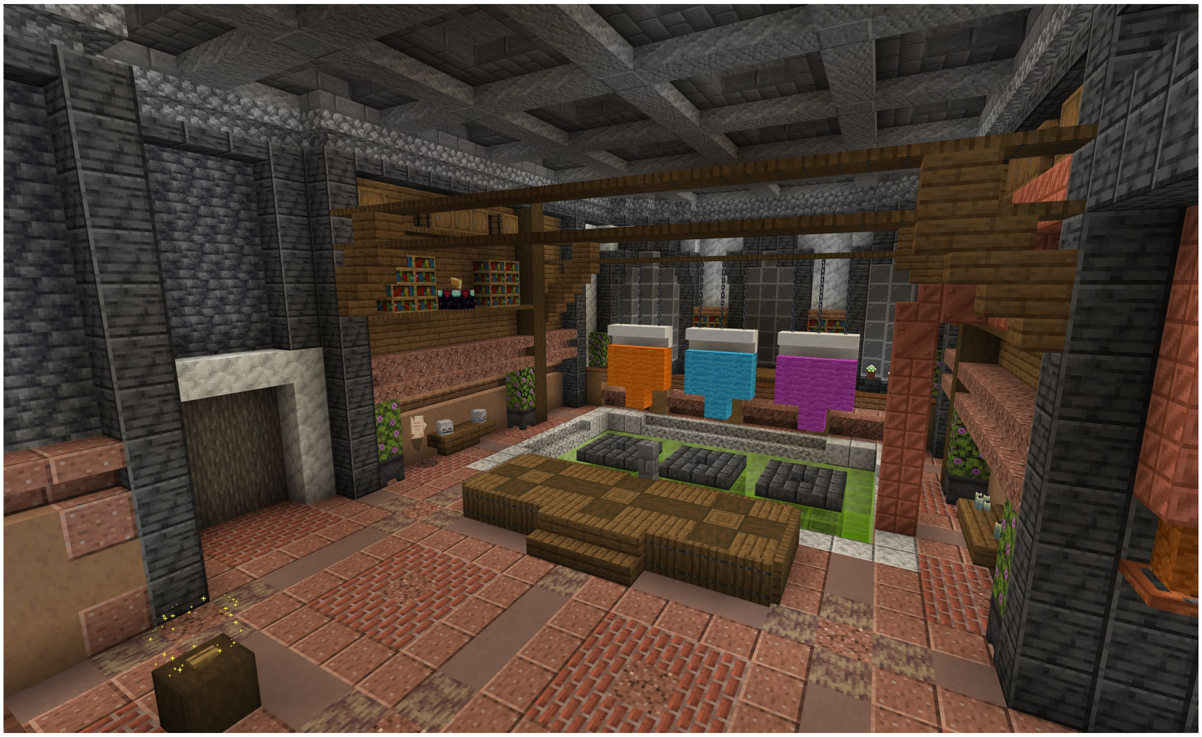

on start
move king 🟧 by 4
move king 🟦 by 1
move queen 🟨 by 4
move queen 🟧 by 1
	
	`for i in range(4):`
	`    move_chess_piece("king", "orange")`
	`move_chess_piece("king", "blue")`
	
	`for i in range(4):`
	`    move_chess_piece("queen", "yellow")`
	`move_chess_piece("queen", "orange")`


January
February
March
May
1st
2nd
3rd
4th

SOLUTIONS: YELLOW PATHWAY



Room 5: Alchemy Room



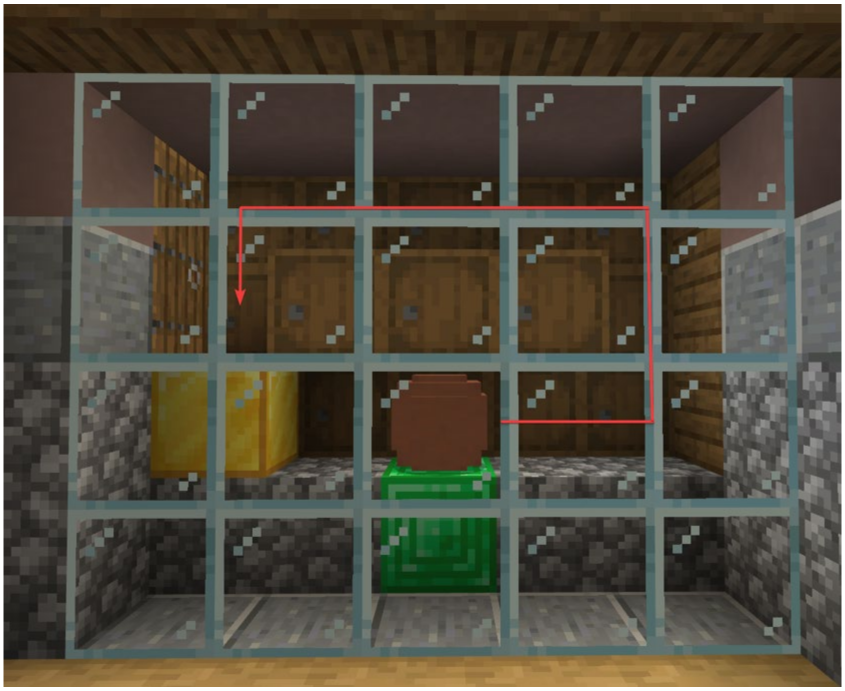
In the Alchemy Room, there are 3 coding puzzles:

- Move the Clay Ball Puzzle

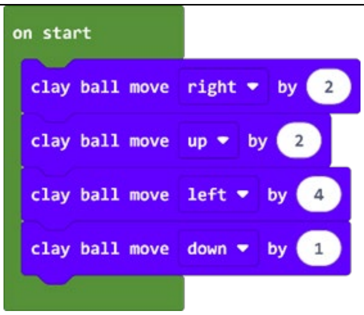
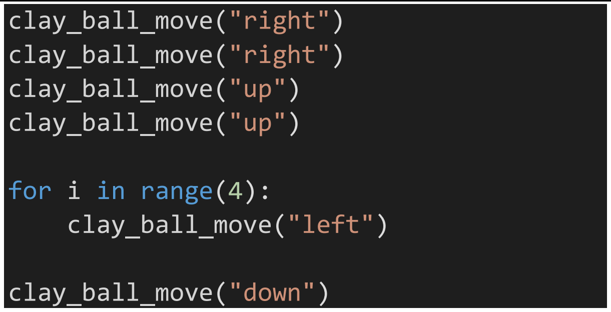
- Clay Golem Puzzle
- Potion Puzzle

ALCHEMY ROOM – Coding Puzzle 1: Move the Clay Ball

Players need to move the clay ball to end up on the gold block.

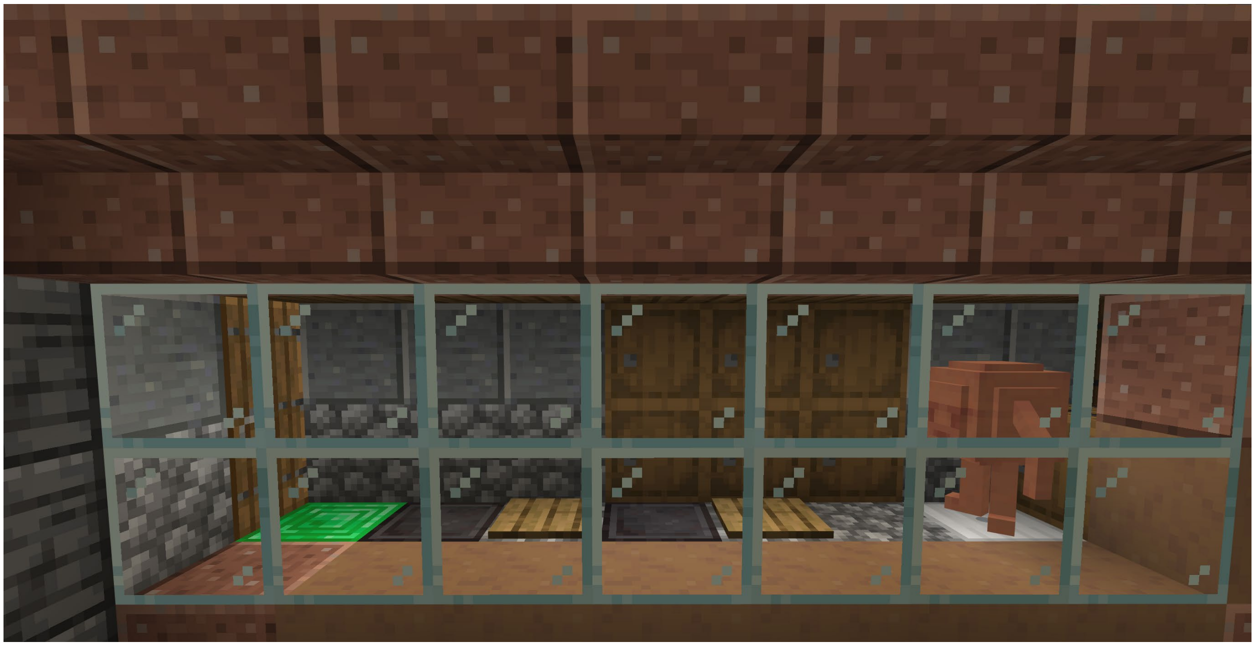


Blocks	Python

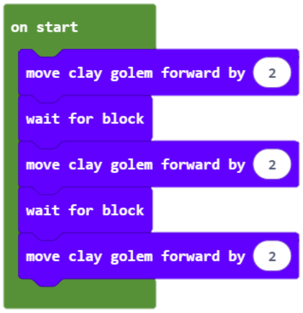

on start
clay ball move right ▾ by 2
clay ball move up ▾ by 2
clay ball move left ▾ by 4
clay ball move down ▾ by 1
	
	`clay_ball_move("right")`
	`clay_ball_move("right")`
	`clay_ball_move("up")`
	`clay_ball_move("up")`
	
	`for i in range(4):`
	`    clay_ball_move("left")`
	
	`clay_ball_move("down")`

ALCHEMY ROOM – Coding Puzzle 2: Clay Golem

The clay golem needs to move forward 6 blocks; When the golem steps on a pressure plate, it will open a pit for 1 second so the golem will need to wait for the block to come back before moving.

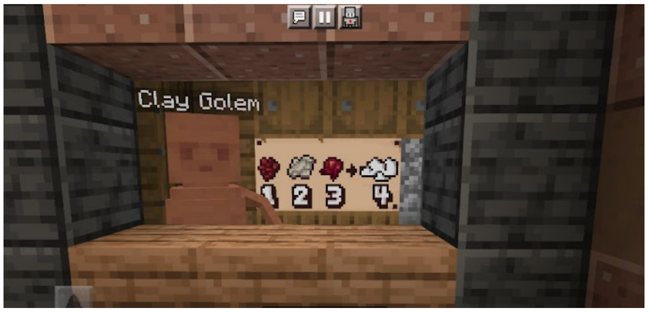


Blocks	Python

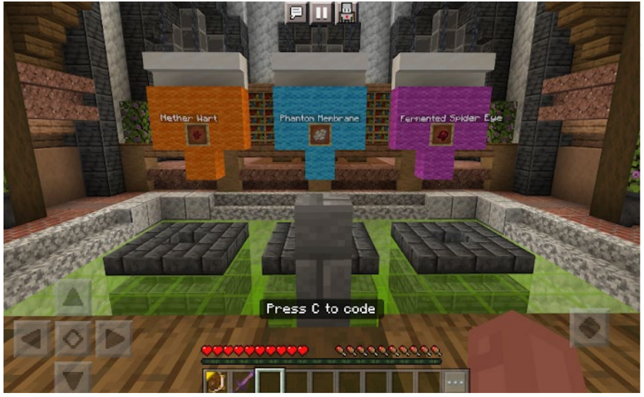

on start
move clay golem forward by 2
wait for block
move clay golem forward by 2
wait for block
move clay golem forward by 2
	move_golem_forward()
	move_golem_forward()
	wait()
	move_golem_forward()
	move_golem_forward()
	wait()
	move_golem_forward()
	move_golem_forward()

ALCHEMY ROOM – Coding Puzzle 3: Potion Puzzle

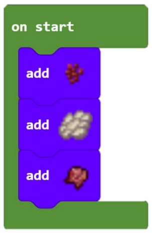
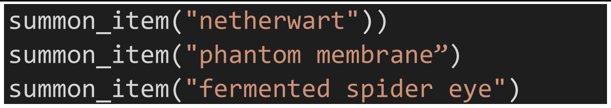
The player needs to create a potion—the ingredient list can be found on the wall.



After you press the button, you will see these ingredients appear in these spots.



Blocks	Python


 	
	
	`summon_item("netherwart"))`
	`summon_item("phantom membrane”)`
	`summon_item("fermented spider eye")`

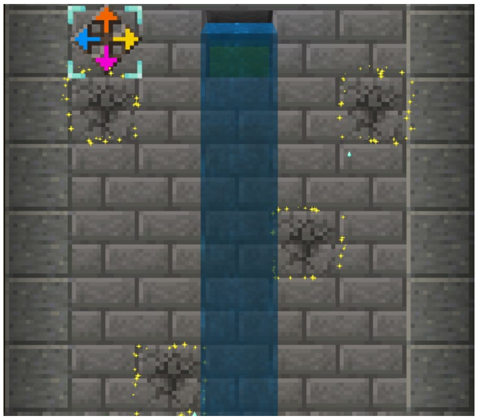
Room 6: Conservatory

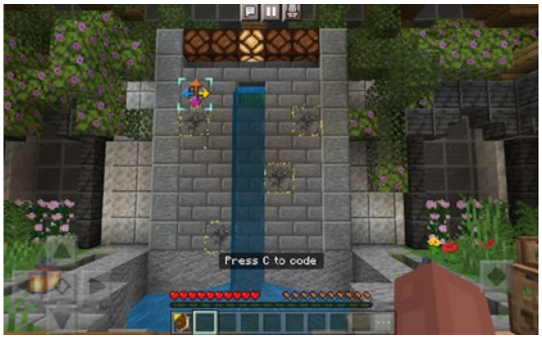
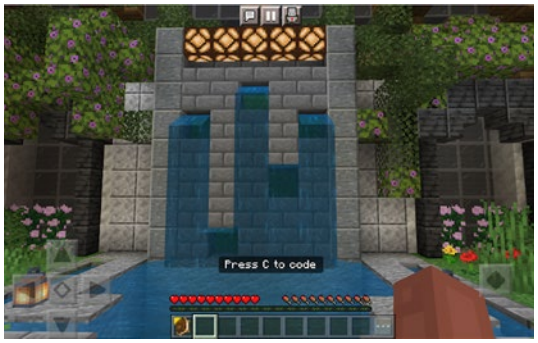
In the Conservatory, there are 3 coding puzzles:

- Breaking Blocks (Part 1) Puzzle
- Water Fountain (Part 2) Puzzle
- Flower Patterns

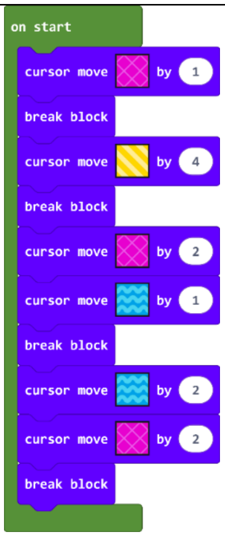
CONSERVATORY – Coding Puzzle 1: Breaking Blocks (Part 1) Puzzle

Players need to break blocks to allow for the water to start flowing.



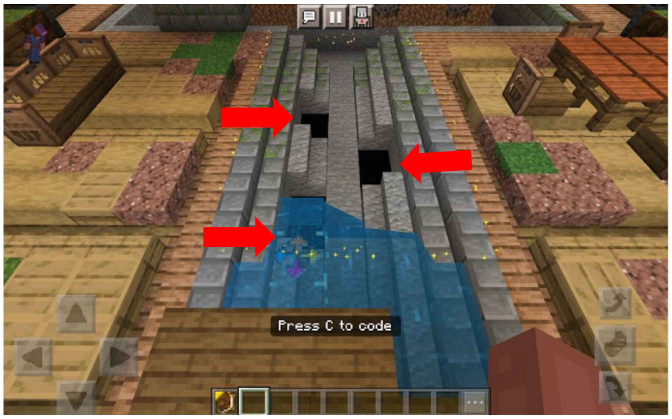
BEFORE	AFTER
	

Blocks	Python

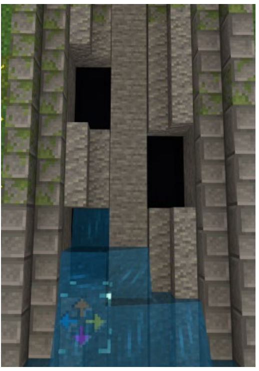

on start
cursor move 🟪 by 1
break block
cursor move 🟨 by 4
break block
cursor move 🟪 by 2
cursor move 🟦 by 1
break block
cursor move 🟦 by 2
cursor move 🟪 by 2
break block
	move_cursor("magenta")
	break_block()
	
	for i in range(4):
	    move_cursor("yellow")
	break_block()
	
	move_cursor("magenta")
	move_cursor("magenta")
	move_cursor("blue")
	break_block()
	
	move_cursor("blue")
	move_cursor("blue")
	move_cursor("magenta")
	move_cursor("magenta")
	break_block()

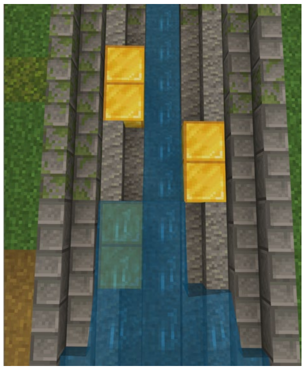
CONSERVATORY – Coding Puzzle 2: Plugging Holes Puzzle

Players need to use the cursor to place gold blocks to plug the holes in the fountain.

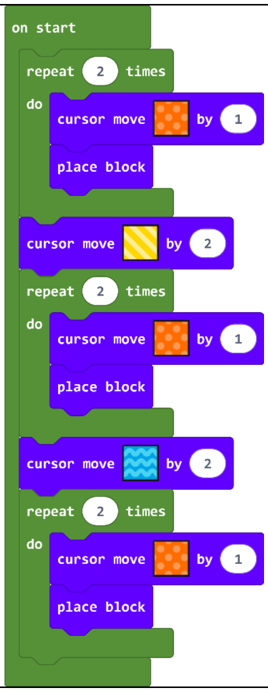


C



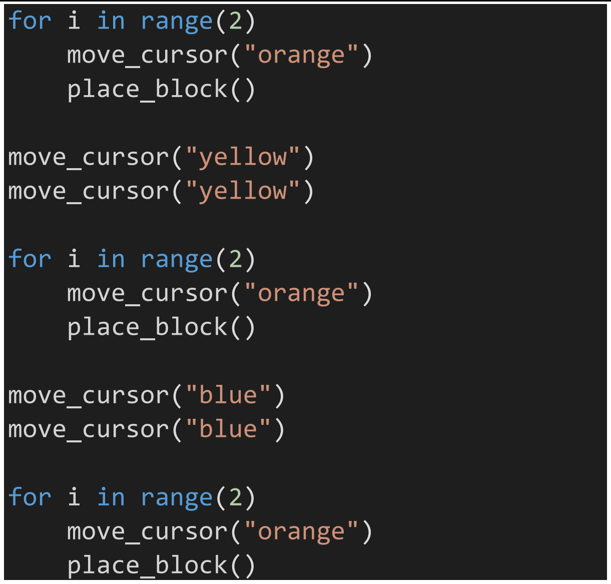


<table>
<tr><th>Blocks</th><th>Python</th></tr>
<tr><td>



```
on start
  repeat 2 times
  do
    cursor move 🟧 by 1
    place block
  cursor move 🟨 by 2
  repeat 2 times
  do
    cursor move 🟧 by 1
    place block
  cursor move 🟦 by 2
  repeat 2 times
  do
    cursor move 🟧 by 1
    place block
```

</td><td>



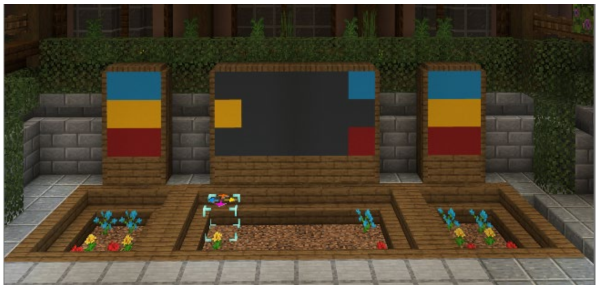
```
for i in range(2)
    move_cursor("orange")
    place_block()

move_cursor("yellow")
move_cursor("yellow")

for i in range(2)
    move_cursor("orange")
    place_block()

move_cursor("blue")
move_cursor("blue")

for i in range(2)
    move_cursor("orange")
    place_block()
```

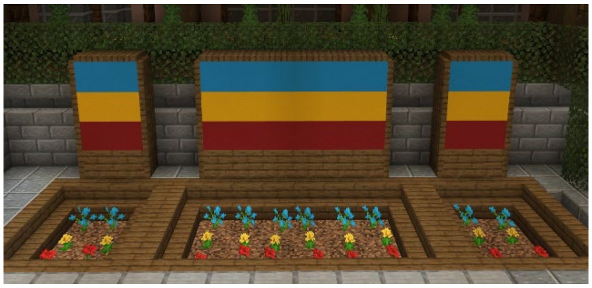
</td></tr>
</table>

CONSERVATORY – Coding Puzzle 3: Flower Patterns Puzzle

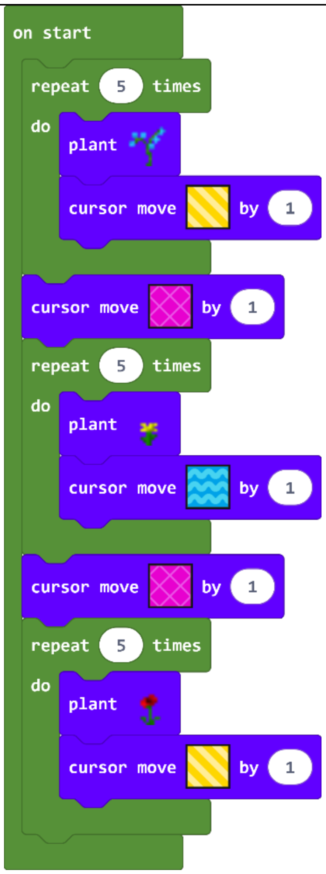
Players need to use the cursor to create a code to fill in the middle flower bed based on the colored flowers (blue, yellow, red) displayed on the side flower beds.

HINT: Use the colored blocks on the wall to help you count the number of spaces within the flower bed.



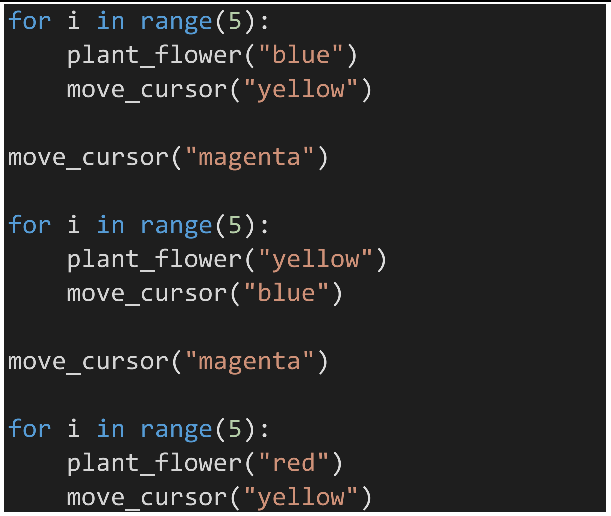


<table>
<tr><th>Blocks</th><th>Python</th></tr>
<tr><td>



```
on start
  repeat 5 times
  do
    plant
    cursor move by 1
  cursor move by 1
  repeat 5 times
  do
    plant
    cursor move by 1
  cursor move by 1
  repeat 5 times
  do
    plant
    cursor move by 1
```

</td><td>



```python
for i in range(5):
    plant_flower("blue")
    move_cursor("yellow")

move_cursor("magenta")

for i in range(5):
    plant_flower("yellow")
    move_cursor("blue")

move_cursor("magenta")

for i in range(5):
    plant_flower("red")
    move_cursor("yellow")
```

</td></tr>
</table>

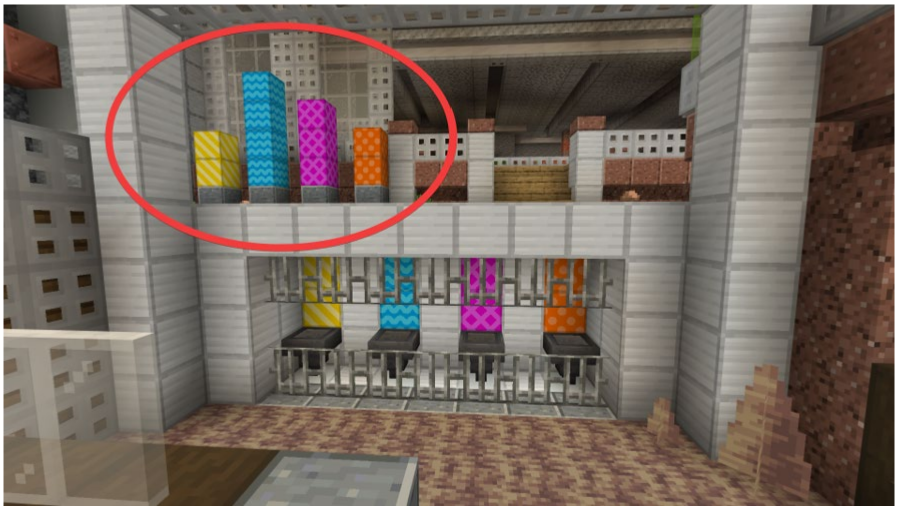
Room 7: Laboratory

In the Laboratory, there are 3 coding puzzles:

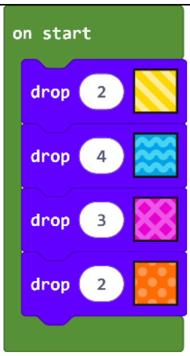
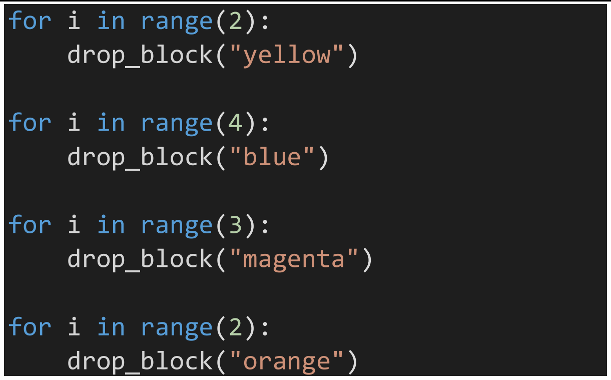
- The Hopper Puzzle
- Animal Scale Puzzle
- Mob Minecart Passageway Puzzle

LABORATORY– Coding Puzzle 1: Hopper Puzzle

Players need to drop additional color blocks to have the hopper start. Use the number of colored blocks shown here to build your code.



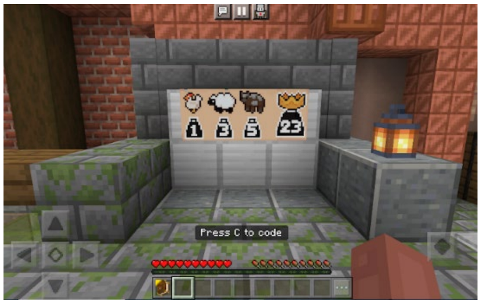
Blocks	Python

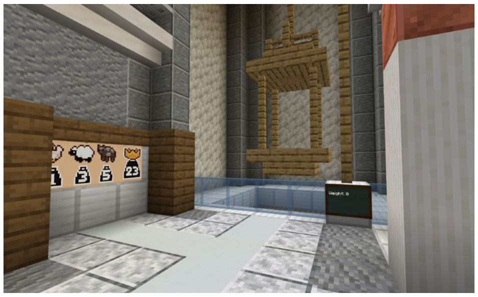

on start
drop 2 ▨
drop 4 ▨
drop 3 ▨
drop 2 ▨
	
	for i in range(2):
	    drop_block("yellow")
	
	for i in range(4):
	    drop_block("blue")
	
	for i in range(3):
	    drop_block("magenta")
	
	for i in range(2):
	    drop_block("orange")

LABORATORY – Coding Puzzle 2: Animal Scale Puzzle

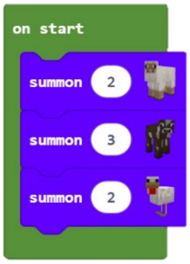
Players will need to place the appropriate combination of animals on the scale to total 23 pounds.

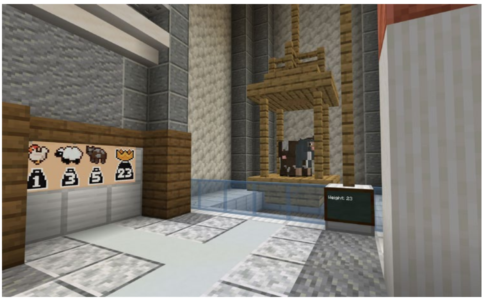
NOTE: There are numerous ways to complete this puzzle! This is one possible way to solve the puzzle.





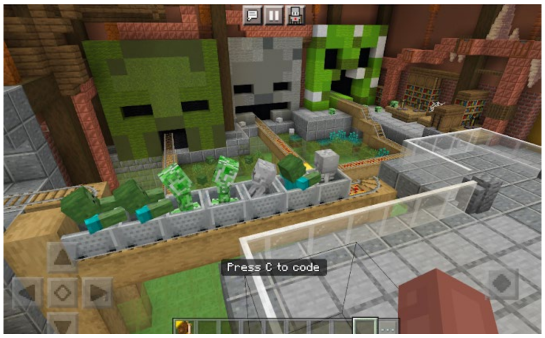
Blocks	Python

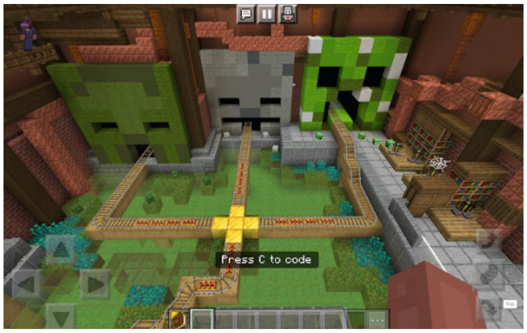

on start
summon 2
summon 3
summon 2
	`for i in range(2):`
	`    summon_animal("sheep")`
	
	`for i in range(3):`
	`    summon_animal("cow")`
	
	`for i in range(2):`
	`    summon_animal("chicken")`

a 

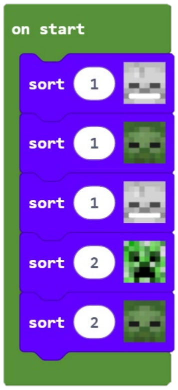
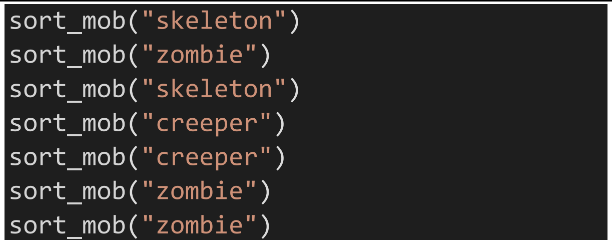
LABORATORY– Coding Puzzle 3: Mob Mine Cart Passageway

The mobs need to be sorted into the right passageways. The mobs are already queued up in the mine carts—they just need to be sorted into their passageways!

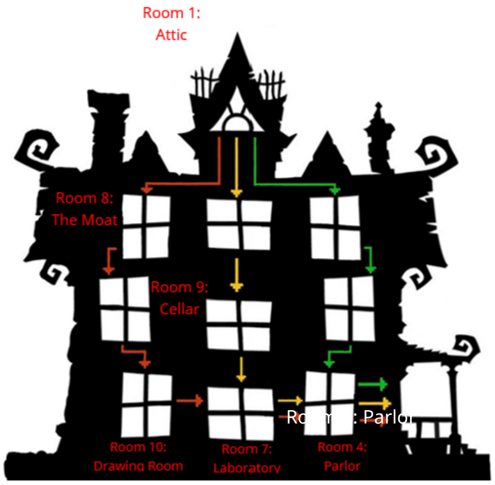




Blocks	Python


on start
sort 1
sort 1
sort 1
sort 2
sort 2
	
	sort_mob("skeleton")
	sort_mob("zombie")
	sort_mob("skeleton")
	sort_mob("creeper")
	sort_mob("creeper")
	sort_mob("zombie")
	sort_mob("zombie")

SOLUTIONS: RED PATHWAY



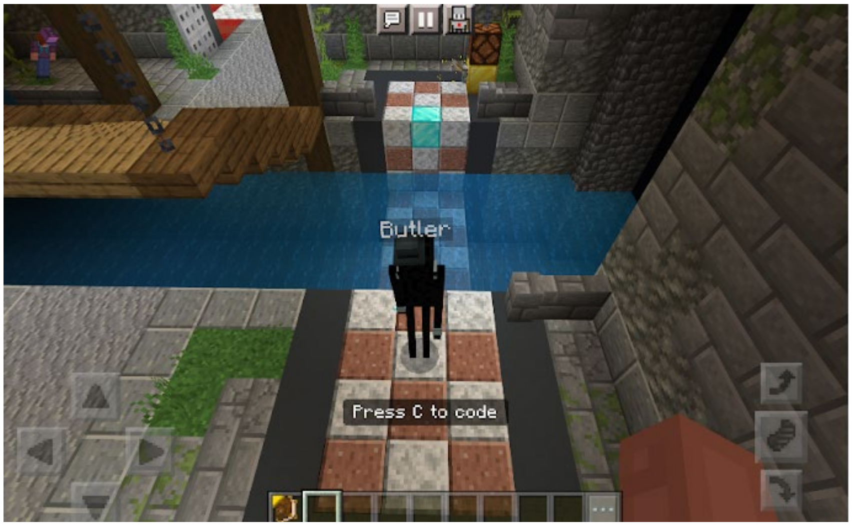
Room 8: The Moat

In the Moat, there are 3 coding puzzles:

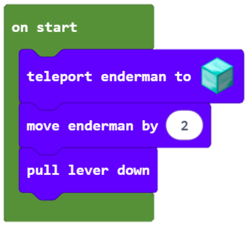
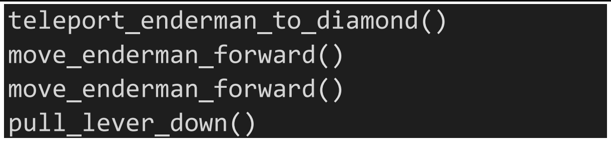
- Crossing the River Puzzle
- Enderman Maze Puzzle
- Flashing Lights Puzzle

THE MOAT – Coding Puzzle 1: Crossing the River

Players need to create a code to help the Enderman Butler cross the river. The Enderman Butler will need to teleport from the starting point across the moat over to the diamond block. The Enderman Butler then needs to flip the lever.



Blocks	Python

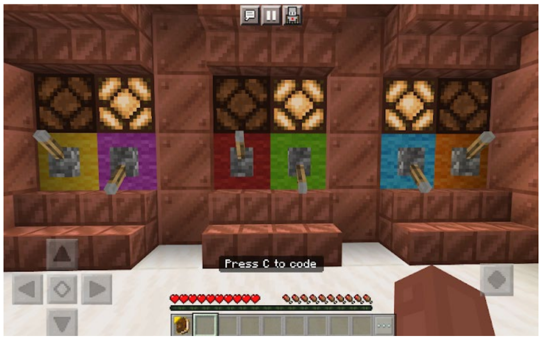

on start
teleport enderman to
move enderman by 2
pull lever down
	
	teleport_enderman_to_diamond()
	move_enderman_forward()
	move_enderman_forward()
	pull_lever_down()

THE MOAT – Coding Puzzle 2: Enderman Maze

The Enderman Maze needs to be solved in two steps:

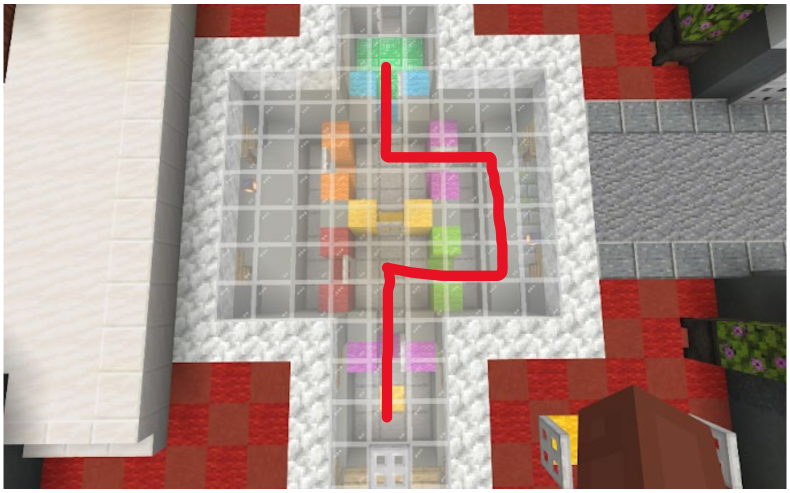
Step 1: Flipping Light Switches

When light switches are flipped, doors within the maze will either open or close. You need to flip the switch to ensure the purple door is open and the blue door is open.

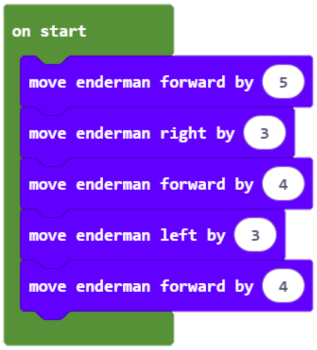
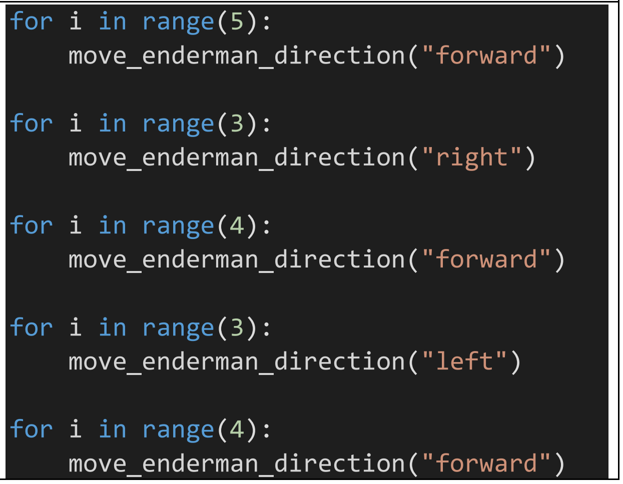


Step 2: Move Through the Maze

The Enderman, starting on the gold block, needs to move through the maze to reach the emerald blocks at the end of the maze.

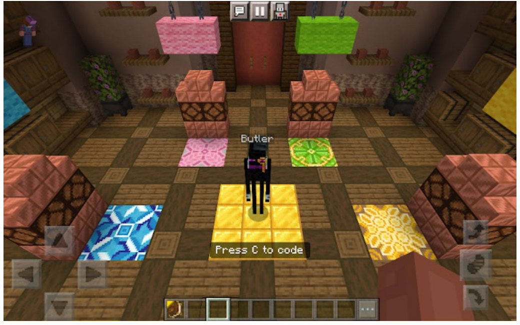


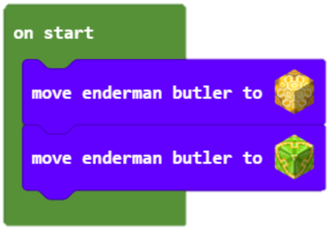
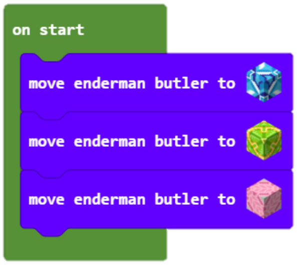
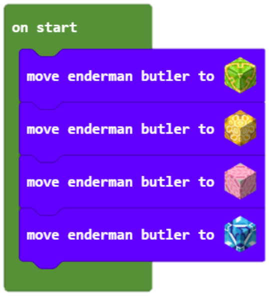
Blocks	Python


`on start`
`move enderman forward by 5`
`move enderman right by 3`
`move enderman forward by 4`
`move enderman left by 3`
`move enderman forward by 4`
	
	`for i in range(5):`
	`    move_enderman_direction("forward")`
	
	`for i in range(3):`
	`    move_enderman_direction("right")`
	
	`for i in range(4):`
	`    move_enderman_direction("forward")`
	
	`for i in range(3):`
	`    move_enderman_direction("left")`
	
	`for i in range(4):`
	`    move_enderman_direction("forward")`

THE MOAT – Coding Puzzle 3: Flashing Lights

There's seems to be a pattern with the way the lights are flashing. Move the enderman to each position following the pattern the lights are making. Players need to identify the pattern and then code the three different light pattern sequences.

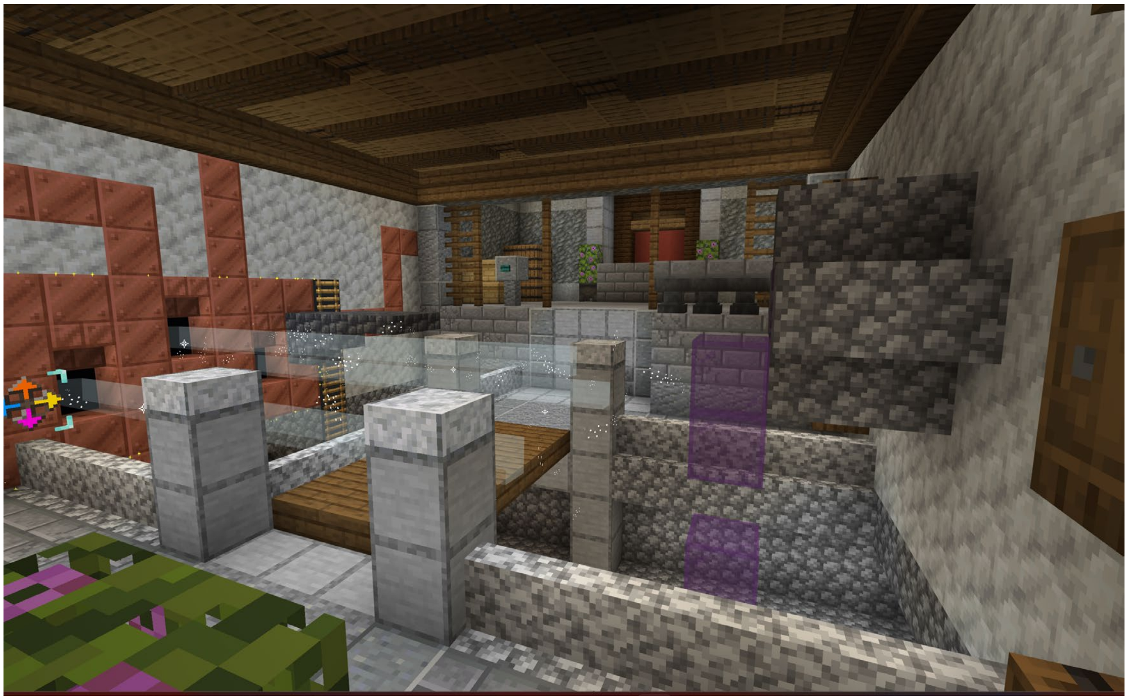


Blocks		
Light Pattern 1	Light Pattern 2	Light Pattern 3
 on start move enderman butler to move enderman butler to	 on start move enderman butler to move enderman butler to move enderman butler to	 on start move enderman butler to move enderman butler to move enderman butler to move enderman butler to

Python		
Light Pattern 1	Light Pattern 2	Light Pattern 3

`move_enderman("yellow")`
`move_enderman("green")`
	`move_enderman("blue")`
	`move_enderman("green")`
	`move_enderman("pink")`
		`move_enderman("green")`
		`move_enderman("yellow")`
		`move_enderman("pink")`
		`move_enderman("blue")`

Room 9: The Cellar

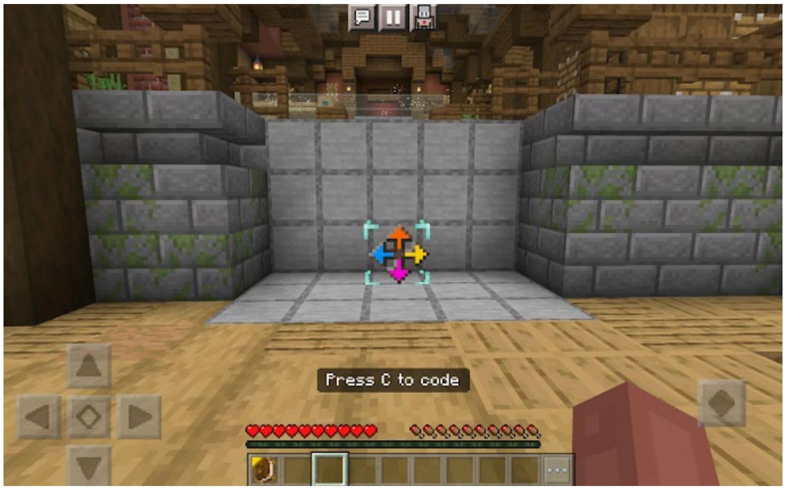


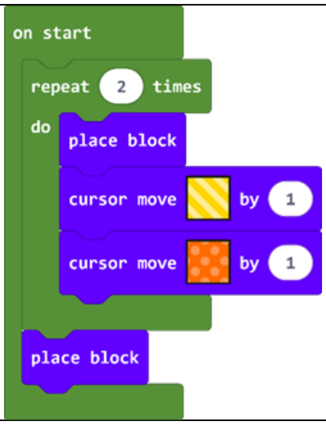
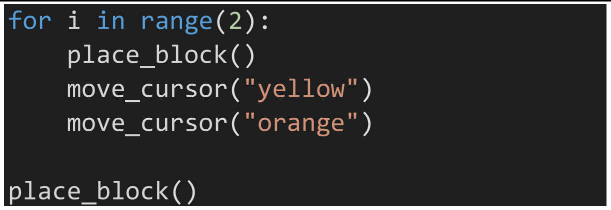
In the Cellar, there are 3 coding puzzles:

- Staircase Puzzle
- Blocking the Wind Puzzle
- Build a Bridge Puzzle

THE CELLAR – Coding Puzzle 1: Staircase

Build a staircase to reach the top of the platform in the cellar.

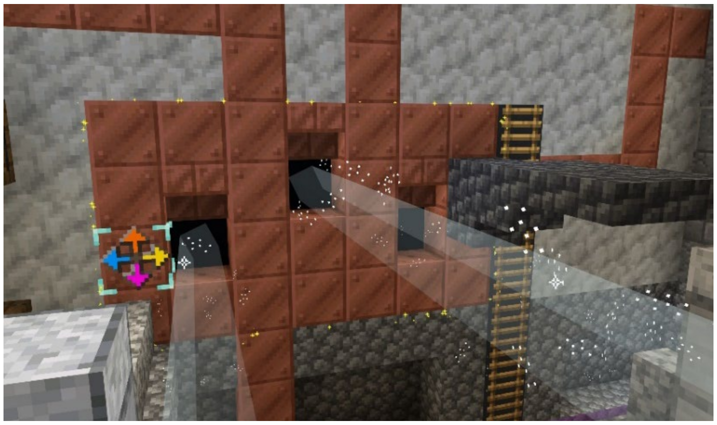


<table>
<tr><th>Blocks</th><th>Python</th></tr>
<tr><td>
<pre>on start
  repeat 2 times
  do place block
     cursor move 🟨 by 1
     cursor move 🟧 by 1
  place block</pre></td><td>
<pre>for i in range(2):
    place_block()
    move_cursor("yellow")
    move_cursor("orange")

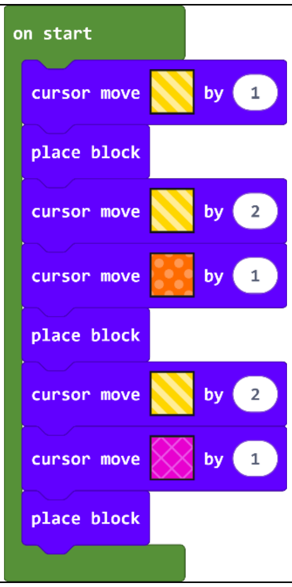
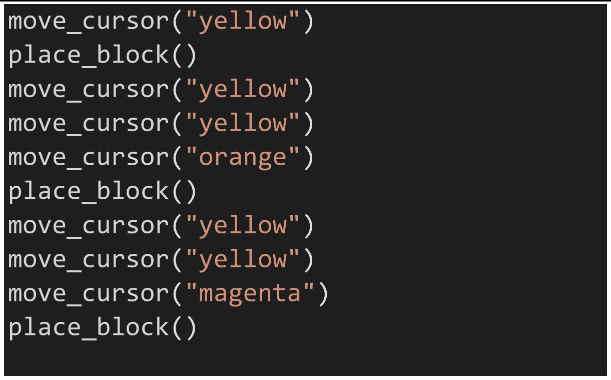
place_block()</pre></td></tr>
</table>

THE CELLAR – Coding Puzzle 2: Blocking the Wind

It’s super windy in here! Can you place blocks over those open holes to control the heavy winds so you can walk across the bridge?

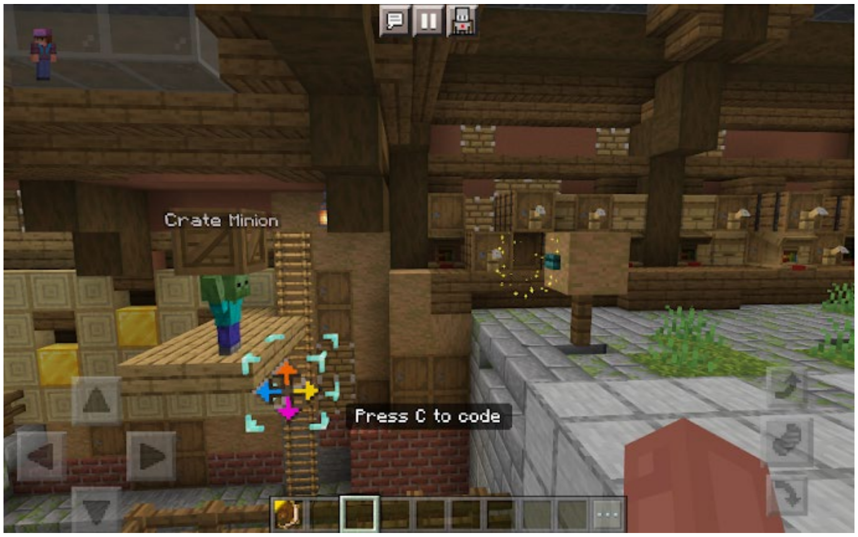


Blocks	Python

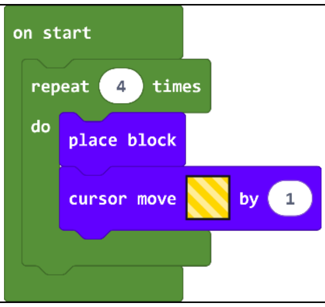
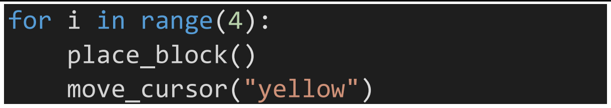

on start
cursor move 🟨 by 1
place block
cursor move 🟨 by 2
cursor move 🟧 by 1
place block
cursor move 🟨 by 2
cursor move 🟪 by 1
place block
	
	`move_cursor("yellow")`
	`place_block()`
	`move_cursor("yellow")`
	`move_cursor("yellow")`
	`move_cursor("orange")`
	`place_block()`
	`move_cursor("yellow")`
	`move_cursor("yellow")`
	`move_cursor("magenta")`
	`place_block()`

THE CELLAR – Coding Puzzle 3: Build a Bridge

Use your cursor to build a bridge that can help the crate minion reach the button! A staircase up will appear once the crate minion has pressed the button.

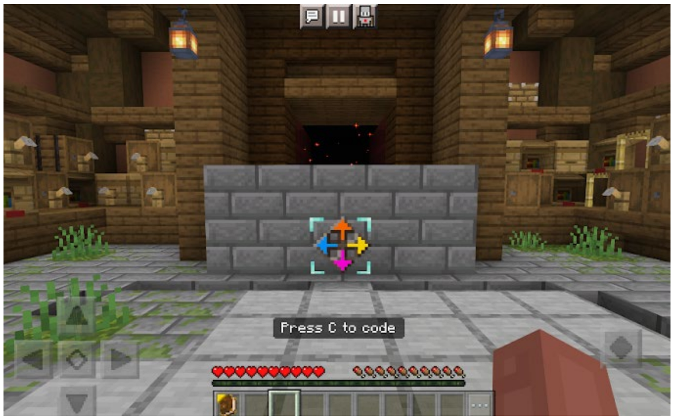


Blocks	Python

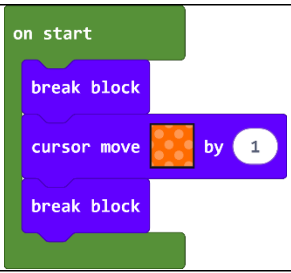
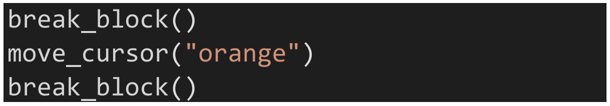

on start
repeat 4 times
do
place block
cursor move 🟨 by 1
	
	`for i in range(4):`
	`    place_block()`
	`    move_cursor("yellow")`

THE CELLAR – Coding Puzzle 4: Block Breaker

Use the cursor to break the blocks in front of the door to escape.



Blocks	Python


on start
break block
cursor move 🟧 by 1
break block
	
	break_block()
	move_cursor("orange")
	break_block()

Room 10: The Drawing Room

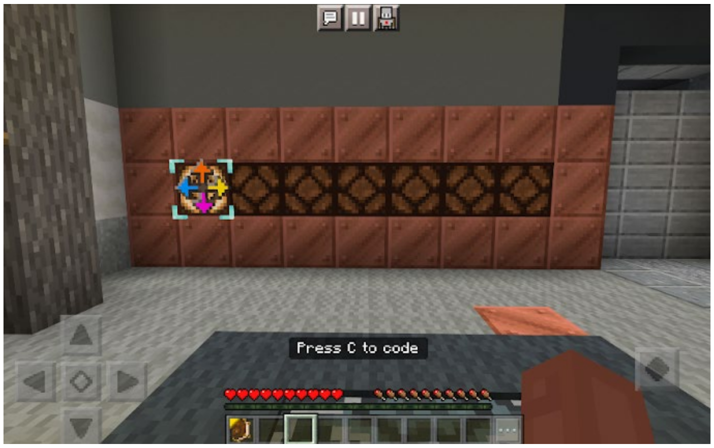
In the Drawing Room, there are 3 coding puzzles:

- Light Power Puzzle
- Red Curtain Puzzle

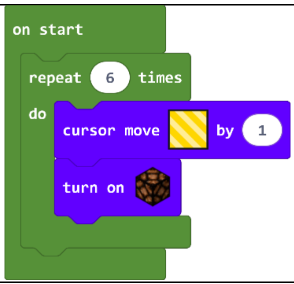
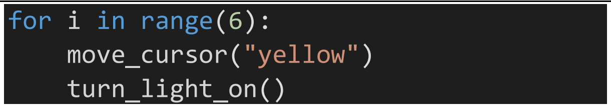
- Laser Show Puzzle

THE DRAWING ROOM– Coding Puzzle 1: Light Power

Use the cursor to turn on all of the lights. Once all the lights are on the door will open.

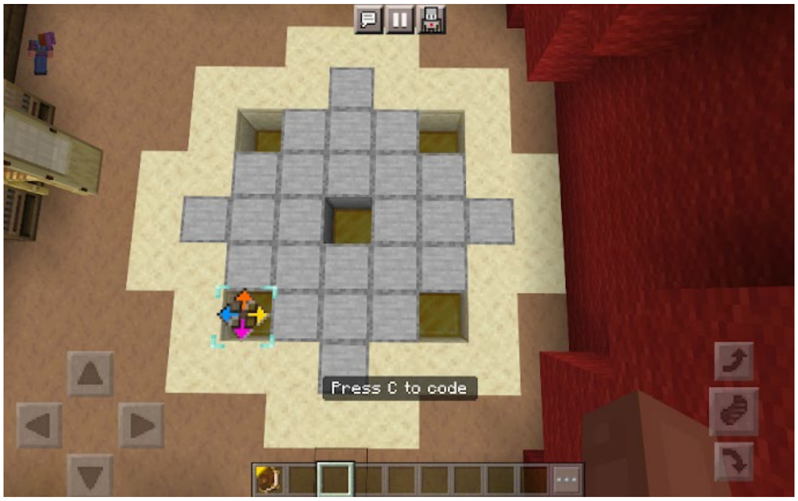


Blocks	Python

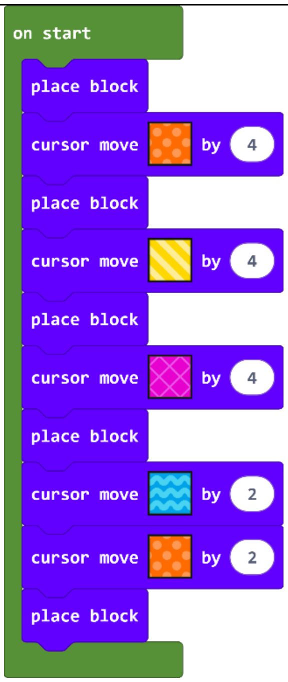
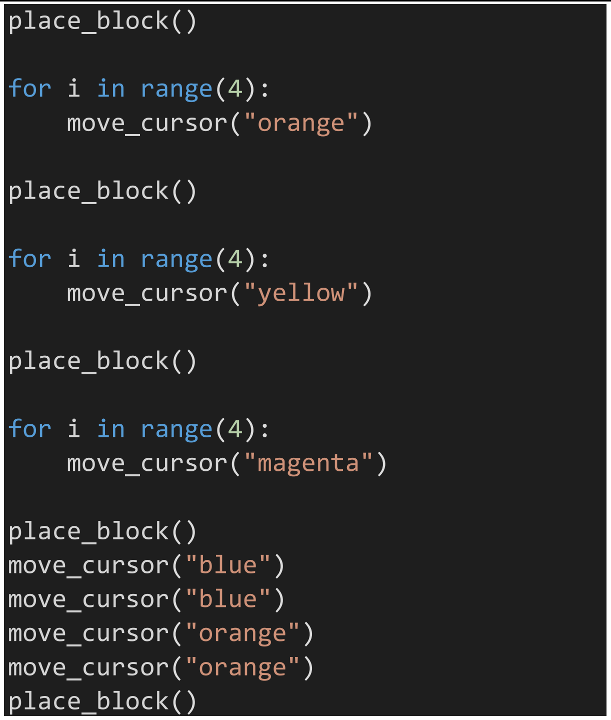

on start
repeat 6 times
do
cursor move by 1
turn on
	
	`for i in range(6):`
	`    move_cursor("yellow")`
	`    turn_light_on()`

THE DRAWING ROOM– Coding Puzzle 2: Red Curtain

Fill in all the holes in the floor with the cursor to open the red curtain.



Blocks	Python

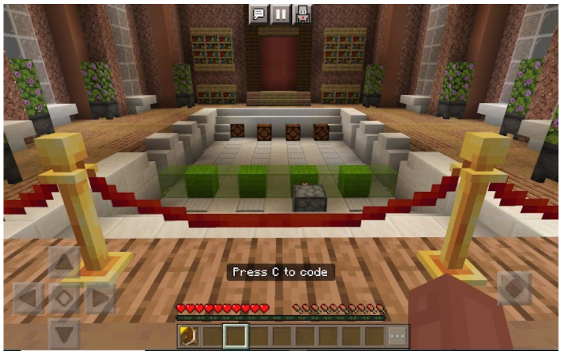

on start
place block
cursor move 🟧 by 4
place block
cursor move 🟨 by 4
place block
cursor move 🟪 by 4
place block
cursor move 🟦 by 2
cursor move 🟧 by 2
place block
	
	place_block()
	
	for i in range(4):
	    move_cursor("orange")
	
	place_block()
	
	for i in range(4):
	    move_cursor("yellow")
	
	place_block()
	
	for i in range(4):
	    move_cursor("magenta")
	
	place_block()
	move_cursor("blue")
	move_cursor("blue")
	move_cursor("orange")
	move_cursor("orange")
	place_block()

THE DRAWING ROOM– Coding Puzzle 3: Laser Show

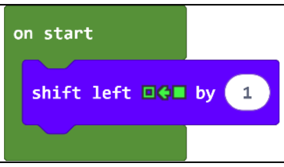
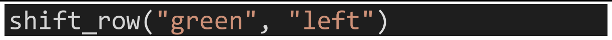
Solid blocks stop the laser!

Step 1: Shift the row of green blocks to make the laser pass-through the glass blocks and turn on the light at the other side.
Step 2: Shift the green and yellow rows on the floor to make the laser pass-through the glass blocks and turn on the light at the other side.

Use the shift blocks to select which row you want to move and the direction it should move in. The block at the end of the row you are shifting will get bumped to the other side.

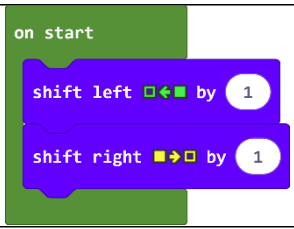
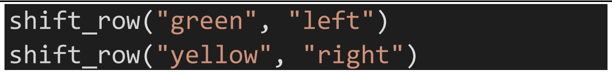


PART 1

Blocks	Python
 on start shift left □←■ by 1	 shift_row("green", "left")

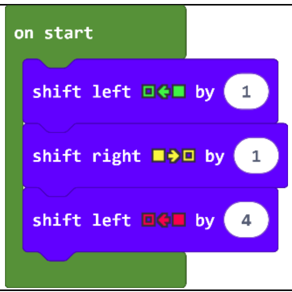
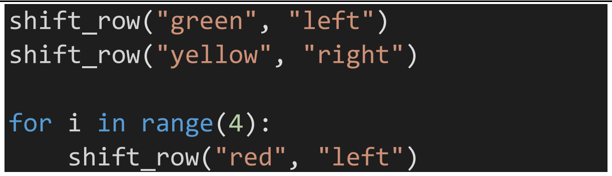
PART 2

Blocks	Python

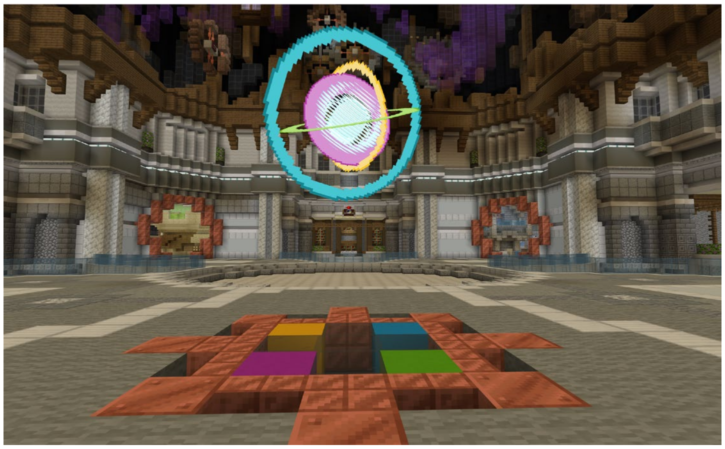

on start
shift left □←■ by 1
shift right ■→□ by 1
	
	shift_row("green", "left")
	shift_row("yellow", "right")

PART 3

Blocks	Python


on start
shift left □←■ by 1
shift right ■→□ by 1
shift left □←■ by 4
	
	shift_row("green", "left")
	shift_row("yellow", "right")
	
	for i in range(4):
	    shift_row("red", "left")

BOSS BATTLE SHOWDOWN

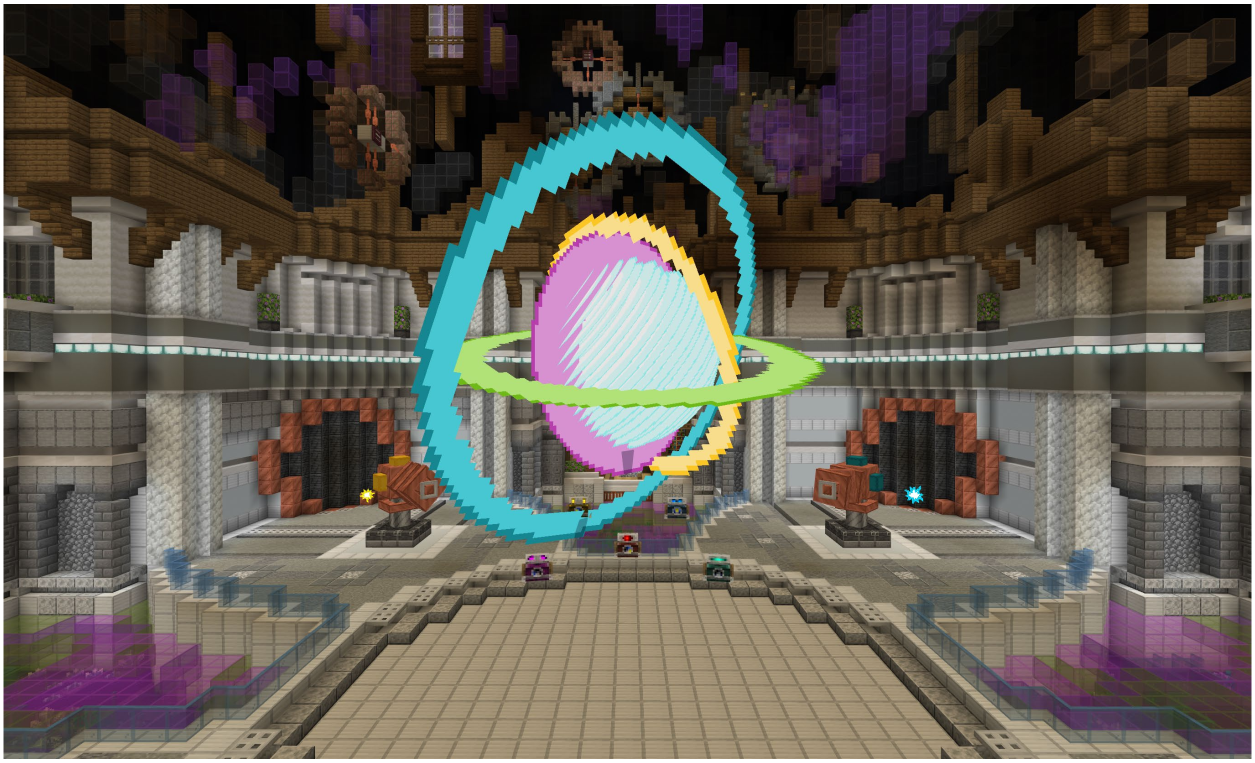


All players will now engage in the epic BOSS BATTLE! Players need to defeat the Time Agents!

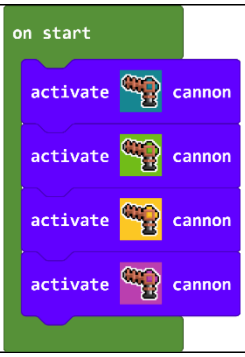
Depending on the pathway, there may be multiple steps within the boss battle sequence.

GREEN PATHWAY – BOSS BATTLE

Player must use the Chrono Cannons to trap the Time Agents! Activate the cannons in the same order as the colored rings surrounding the Time Orb.



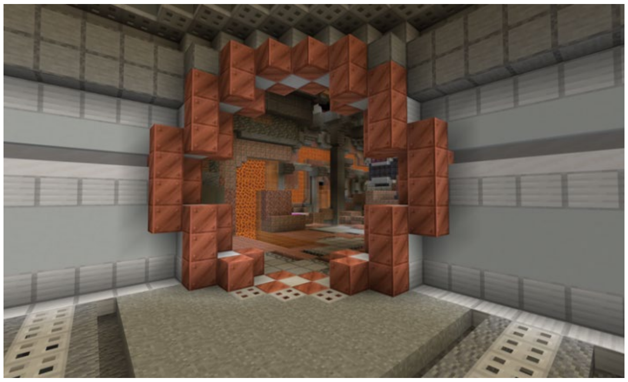
Blocks	Python

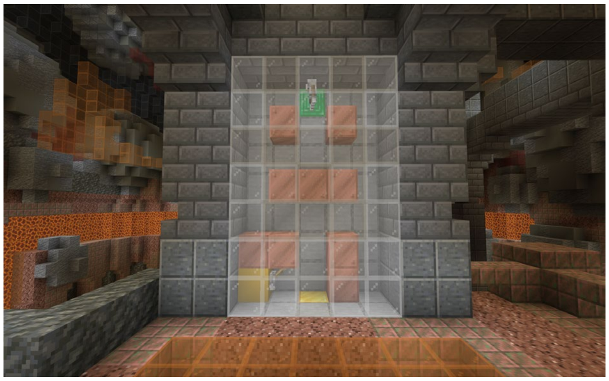

on start
activate cannon
activate cannon
activate cannon
activate cannon
	`activate_cannon("blue")`
	`activate_cannon("green")`
	`activate_cannon("yellow")`
	`activate_cannon("magenta")`

YELLOW PATHWAY – BOSS BATTLE

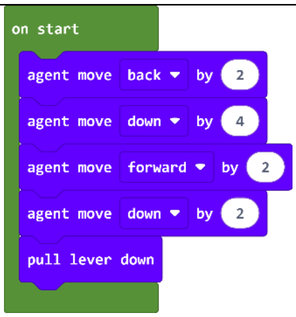
Step 1: Complete the coding puzzles in all four side rooms to activate the Chrono Cannons.

Room 1: Pixie

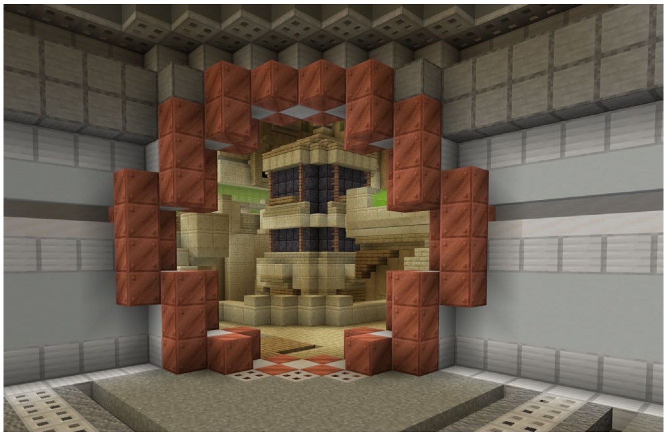


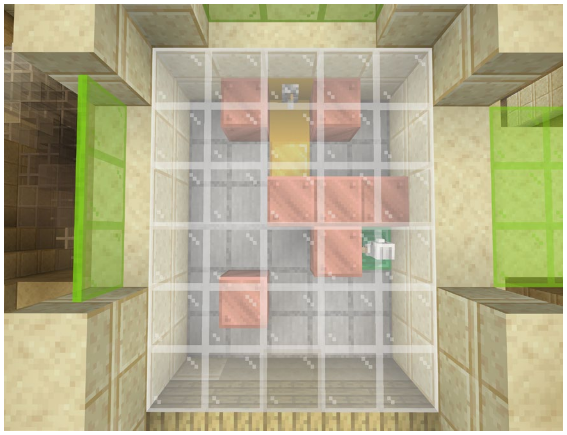


Blocks	Python

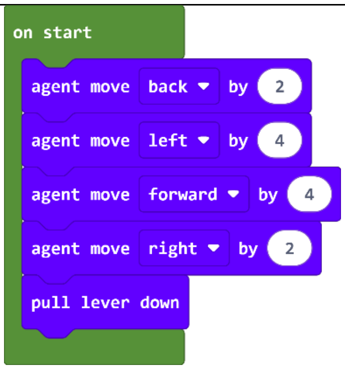
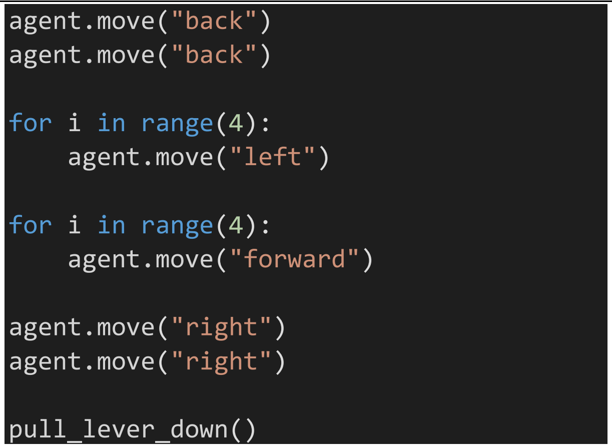

on start
agent move back by 2
agent move down by 4
agent move forward by 2
agent move down by 2
pull lever down
	`agent.move("back")`
	`agent.move("back")`
	
	`for i in range(4):`
	`    agent.move("down")`
	
	`agent.move("forward")`
	`agent.move("forward")`
	`agent.move("down")`
	`agent.move("down")`
	
	`pull_lever_down()`

Room 2: Yeti

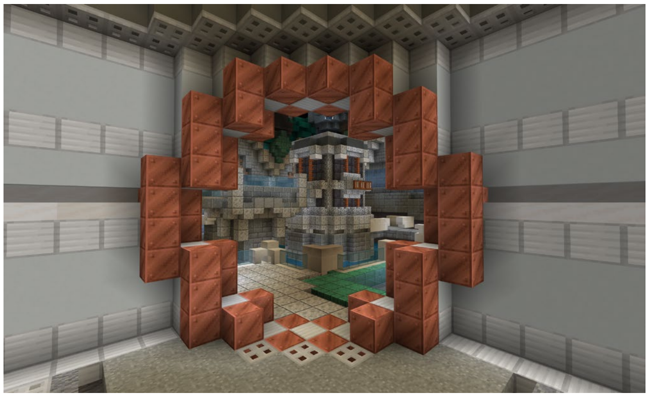


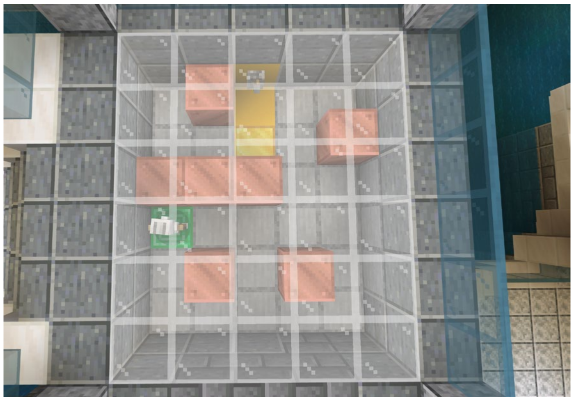


Blocks	Python

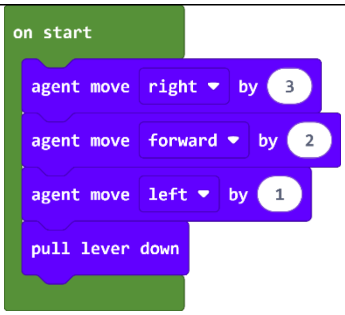

on start
agent move back by 2
agent move left by 4
agent move forward by 4
agent move right by 2
pull lever down
	
	agent.move("back")
	agent.move("back")
	
	for i in range(4):
	    agent.move("left")
	
	for i in range(4):
	    agent.move("forward")
	
	agent.move("right")
	agent.move("right")
	
	pull_lever_down()

Room 3: Banshee



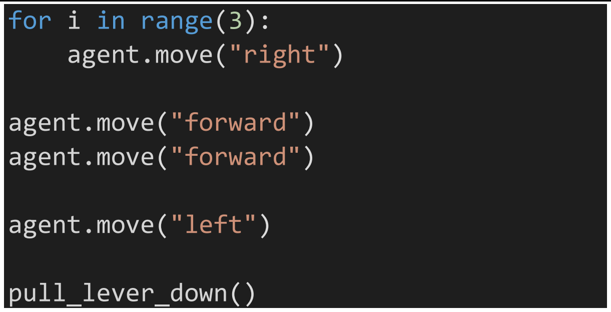


Blocks



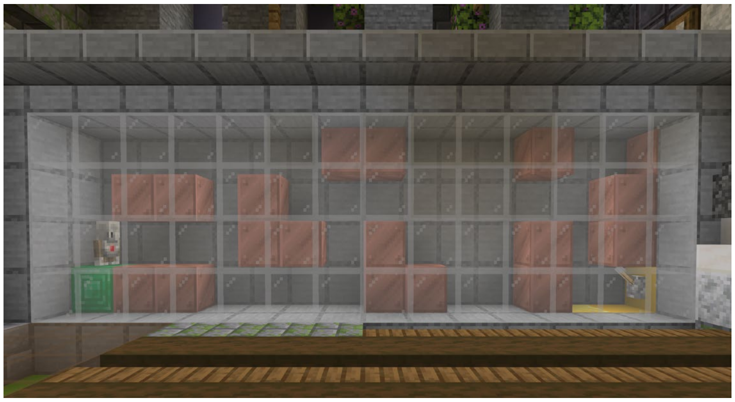
```
on start
    agent move right by 3
    agent move forward by 2
    agent move left by 1
    pull lever down
```

Python

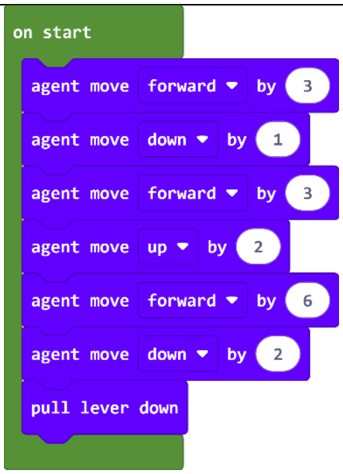


```
for i in range(3):
    agent.move("right")

agent.move("forward")
agent.move("forward")

agent.move("left")

pull_lever_down()
```

Room 4: Gremlin

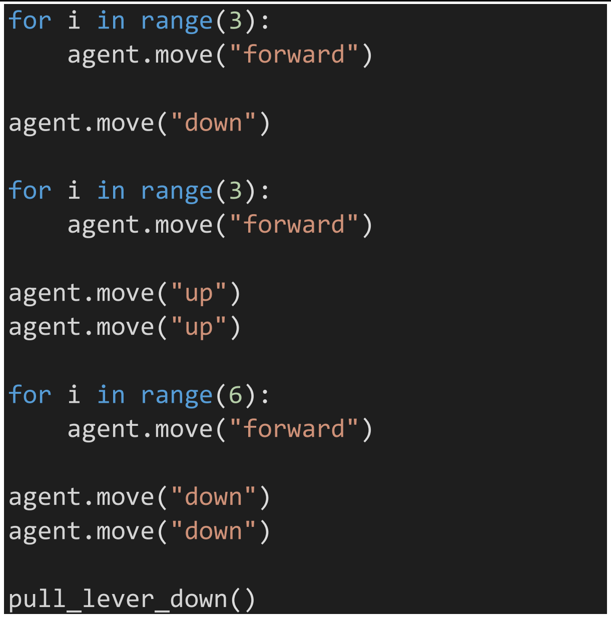


Blocks



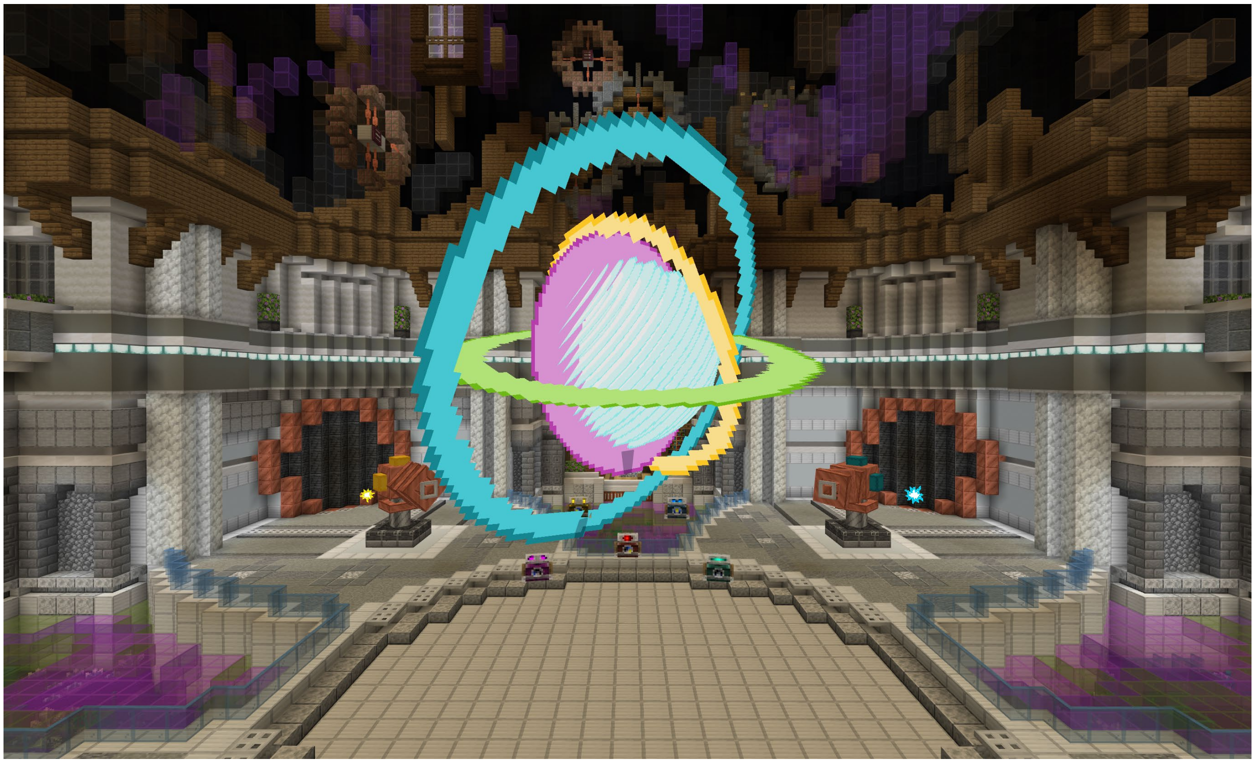
```
on start
    agent move forward by 3
    agent move down by 1
    agent move forward by 3
    agent move up by 2
    agent move forward by 6
    agent move down by 2
    pull lever down
```

Python

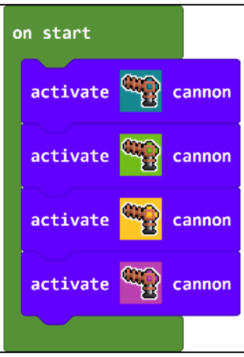


```
for i in range(3):
    agent.move("forward")

agent.move("down")

for i in range(3):
    agent.move("forward")

agent.move("up")
agent.move("up")

for i in range(6):
    agent.move("forward")

agent.move("down")
agent.move("down")

pull_lever_down()
```

Step 2: Activate the cannons in the same order as the colored rings surrounding the Time Orb.

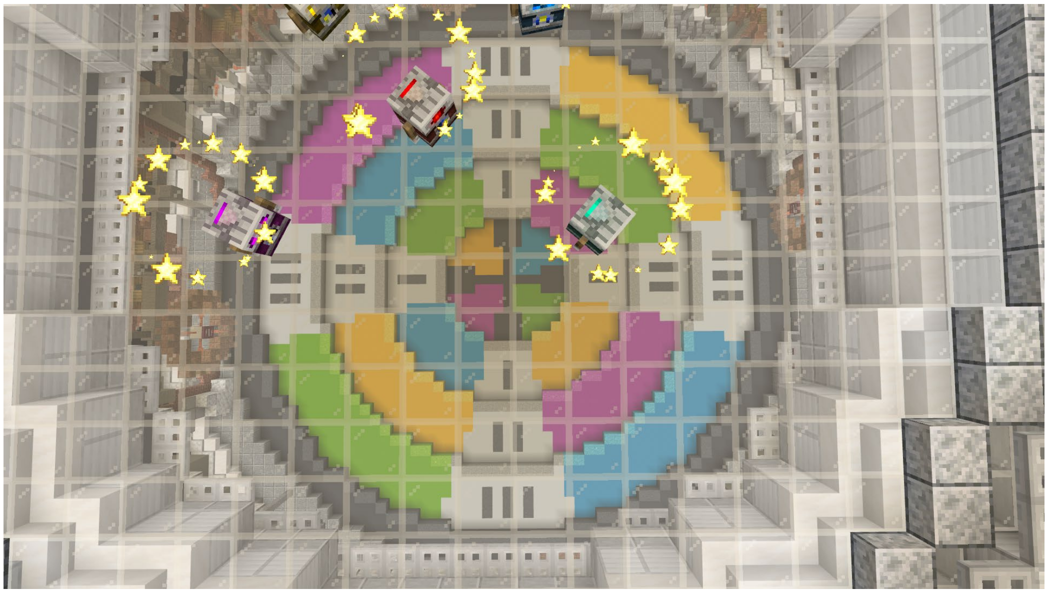


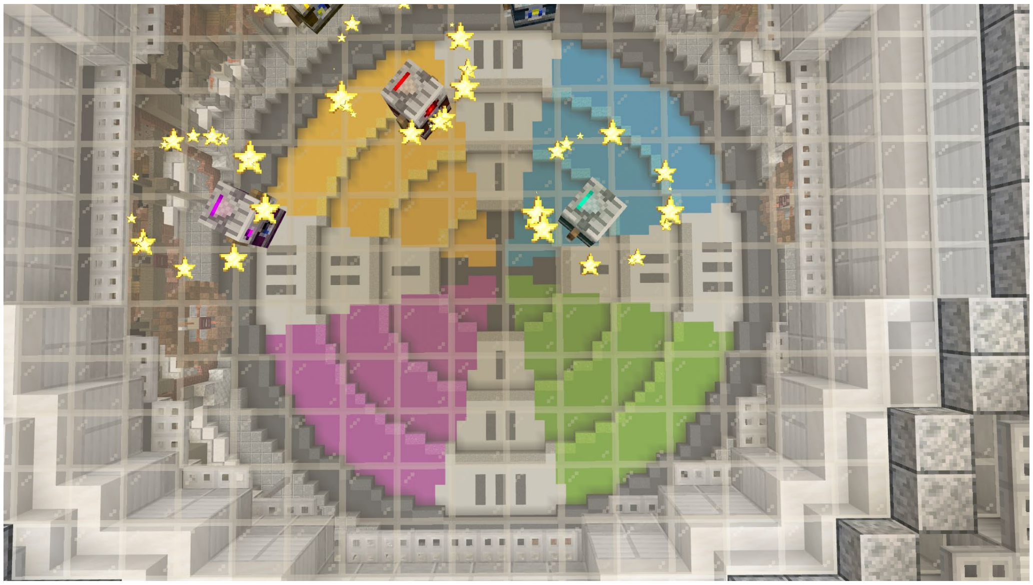
Blocks	Python


on start
activate cannon
activate cannon
activate cannon
activate cannon
	`activate_cannon("blue")`
	`activate_cannon("green")`
	`activate_cannon("yellow")`
	`activate_cannon("magenta")`

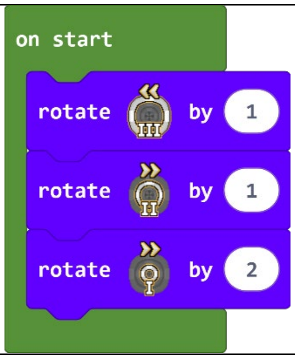
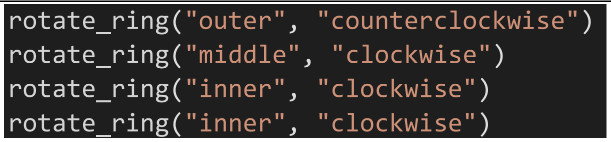
Step 3: Rotate the magnetic discs beneath the floor to match—align all of the sections to trap the time agents!

The sections can be rotated clockwise or counterclockwise. Pay attention to the type of discs—outer disc, middle disc, and the inner disc.



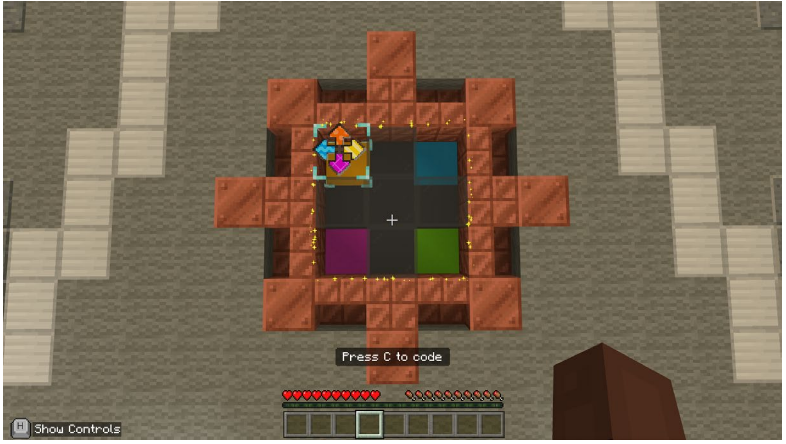


Blocks	Python

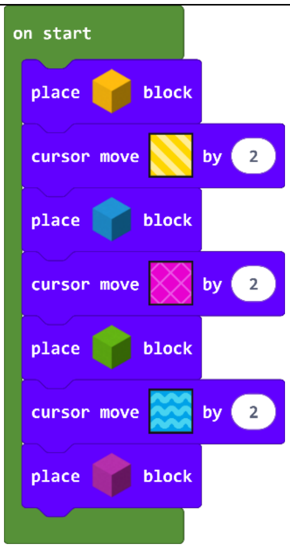
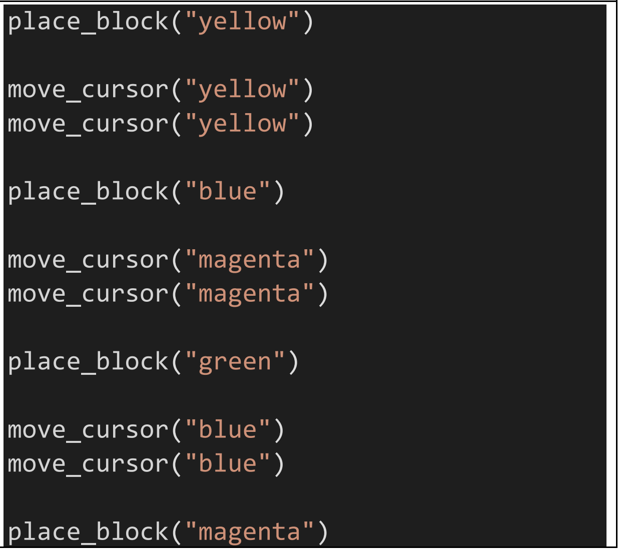

on start
rotate III by 1
rotate II by 1
rotate I by 2
	
	rotate_ring("outer", "counterclockwise")
	rotate_ring("middle", "clockwise")
	rotate_ring("inner", "clockwise")
	rotate_ring("inner", "clockwise")

RED PATHWAY – BOSS BATTLE

Step 1: Fill in the colored key blocks.

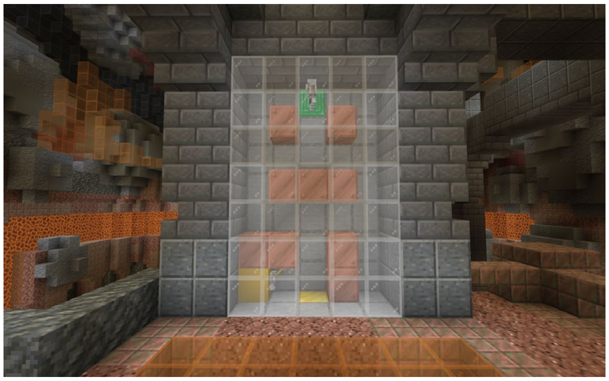


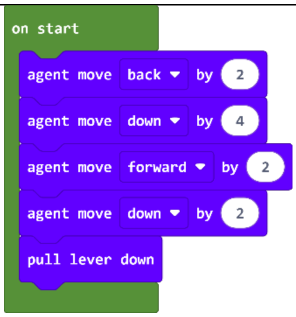
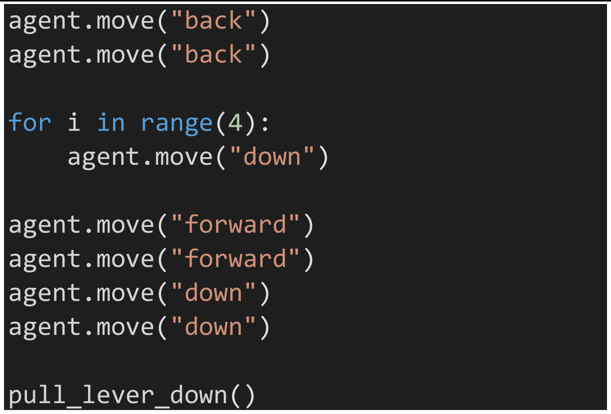
Blocks	Python


on start
place 🟨 block
cursor move 🟨 by 2
place 🟦 block
cursor move 🟪 by 2
place 🟩 block
cursor move 🟦 by 2
place 🟪 block
	
	`place_block("yellow")`
	
	`move_cursor("yellow")`
	`move_cursor("yellow")`
	
	`place_block("blue")`
	
	`move_cursor("magenta")`
	`move_cursor("magenta")`
	
	`place_block("green")`
	
	`move_cursor("blue")`
	`move_cursor("blue")`
	
	`place_block("magenta")`

Step 2: Complete the coding puzzles in all four side rooms to activate the Chrono Cannons.

Room 1: Pixie



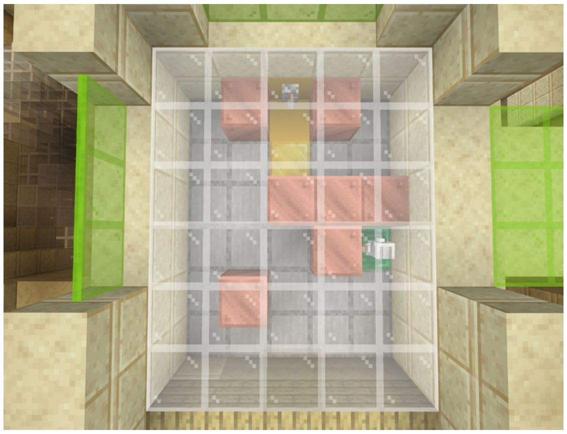
<table>
<tr><th>Blocks</th><th>Python</th></tr>
<tr><td>

<pre>
on start
    agent move back by 2
    agent move down by 4
    agent move forward by 2
    agent move down by 2
    pull lever down
</pre>
</td><td>

<pre>
agent.move("back")
agent.move("back")

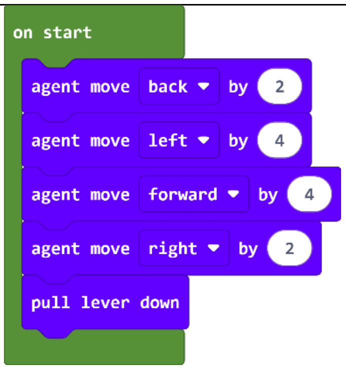
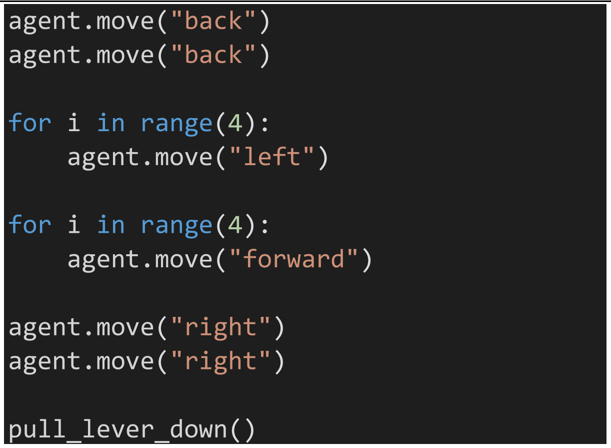
for i in range(4):
    agent.move("down")

agent.move("forward")
agent.move("forward")
agent.move("down")
agent.move("down")

pull_lever_down()
</pre>
</td></tr>
</table>

Room 2: Yeti

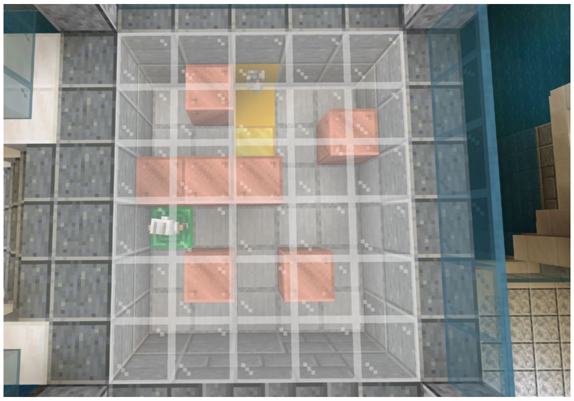


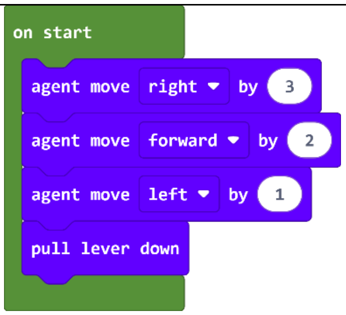
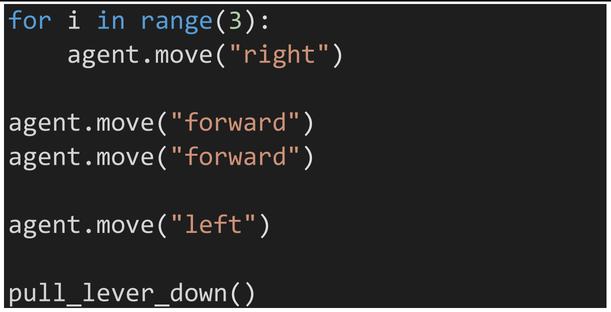
Blocks	Python
	

```
on start
    agent move back ▾ by 2
    agent move left ▾ by 4
    agent move forward ▾ by 4
    agent move right ▾ by 2
    pull lever down
```

```python
agent.move("back")
agent.move("back")

for i in range(4):
    agent.move("left")

for i in range(4):
    agent.move("forward")

agent.move("right")
agent.move("right")

pull_lever_down()
```

Room 3: Banshee

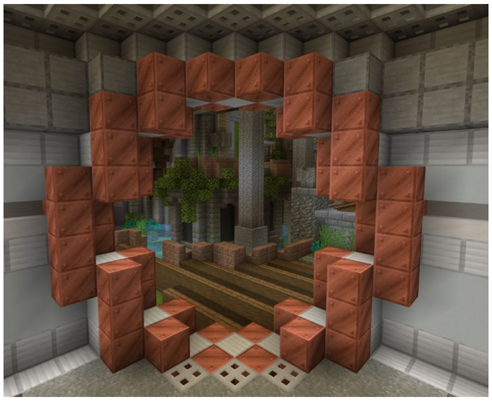


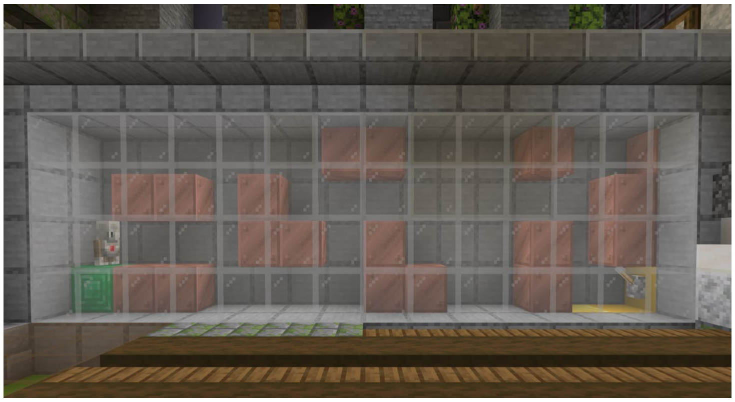
Blocks	Python
	

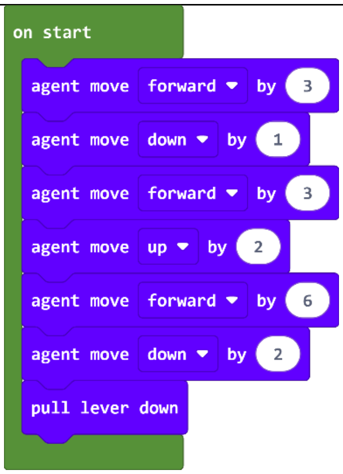
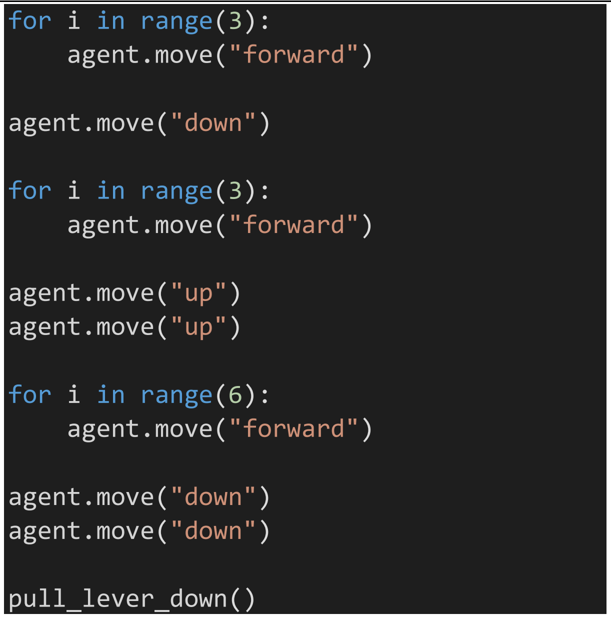
```
on start
    agent move right ▾ by 3
    agent move forward ▾ by 2
    agent move left ▾ by 1
    pull lever down
```

```python
for i in range(3):
    agent.move("right")

agent.move("forward")
agent.move("forward")

agent.move("left")

pull_lever_down()
```

Room 4: Gremlin



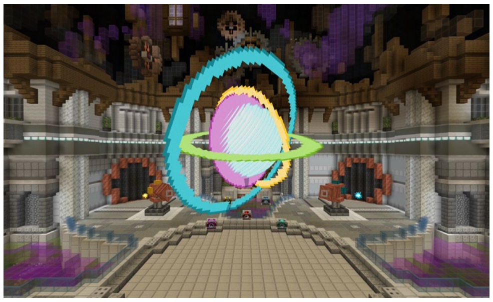


Blocks	Python
	

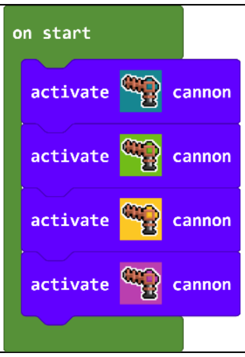
```
on start
    agent move forward by 3
    agent move down by 1
    agent move forward by 3
    agent move up by 2
    agent move forward by 6
    agent move down by 2
    pull lever down
```

```python
for i in range(3):
    agent.move("forward")

agent.move("down")

for i in range(3):
    agent.move("forward")

agent.move("up")
agent.move("up")

for i in range(6):
    agent.move("forward")

agent.move("down")
agent.move("down")

pull_lever_down()
```

Step 3: Activate the cannons in the same order as the colored rings surrounding the Time Orb.

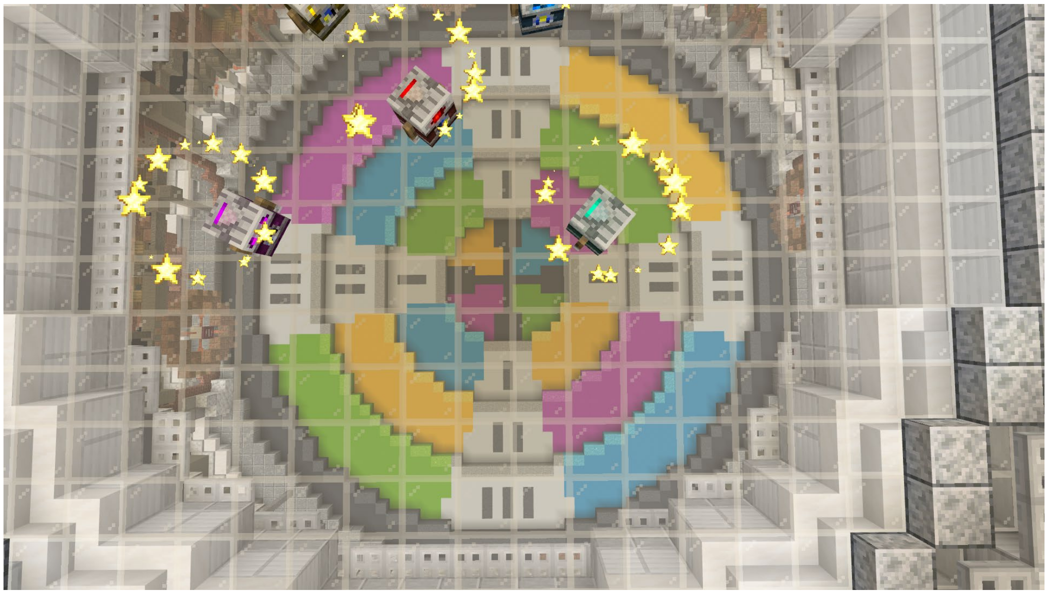


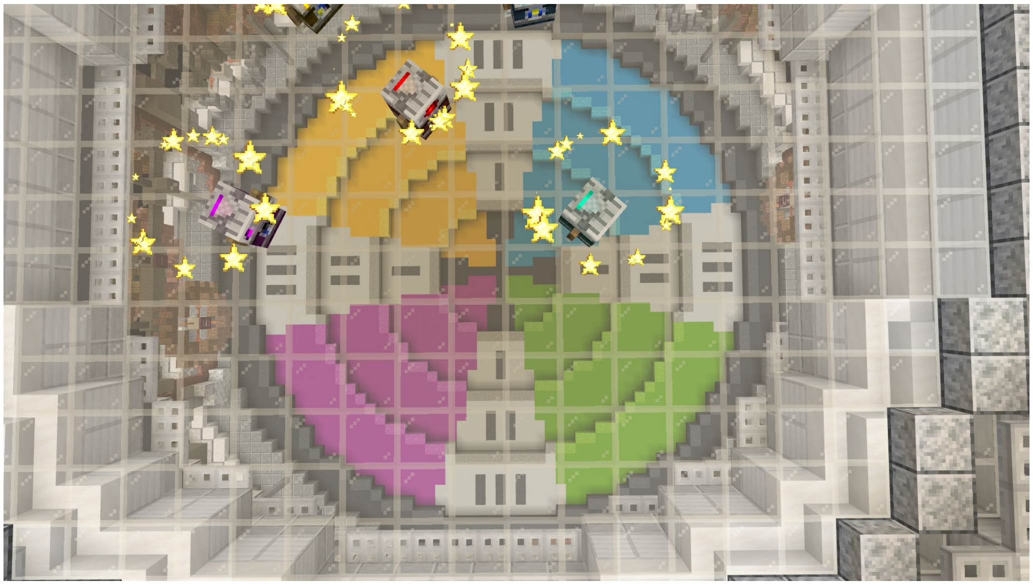
Blocks	Python


on start
activate cannon
activate cannon
activate cannon
activate cannon
	`activate_cannon("blue")`
	`activate_cannon("green")`
	`activate_cannon("yellow")`
	`activate_cannon("magenta")`

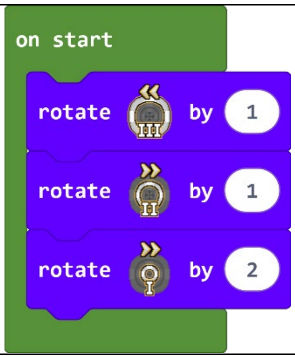
Step 4: Rotate the magnetic discs beneath the floor to match—align all of the sections to trap the time agents!

The sections can be rotated clockwise or counterclockwise. Pay attention to the type of discs—outer disc, middle disc, and the inner disc.





Blocks	Python


on start
rotate « III by 1
rotate » II by 1
rotate » I by 2
	rotate_ring("outer", "counterclockwise")
	rotate_ring("middle", "clockwise")
	rotate_ring("inner", "clockwise")
	rotate_ring("inner", "clockwise")