

Create Your Own World



Introduction

In this project you'll learn how to create your own open world adventure game.

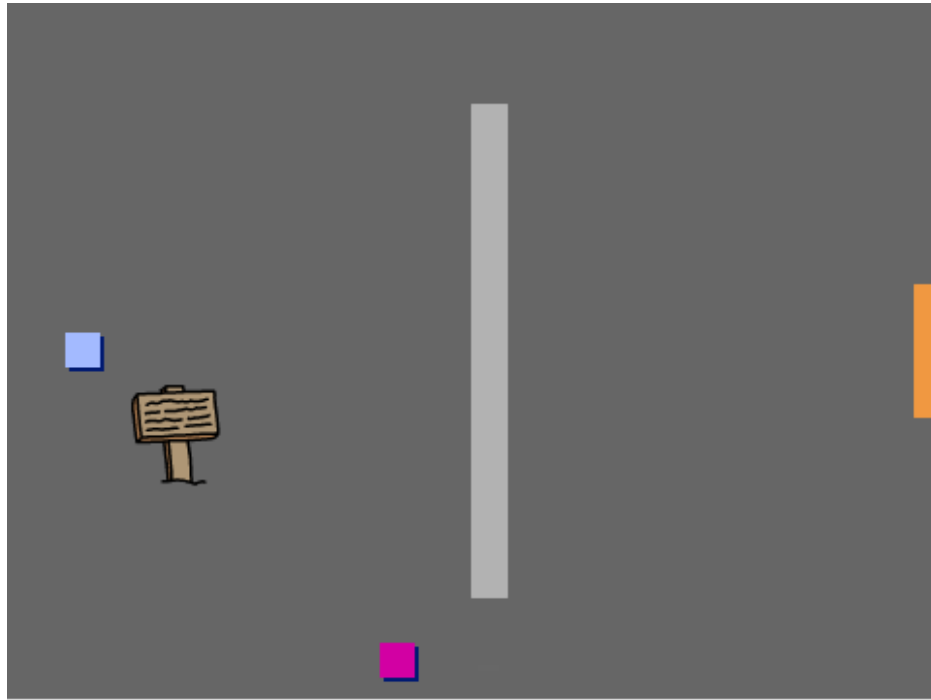


Step 1: Coding your player

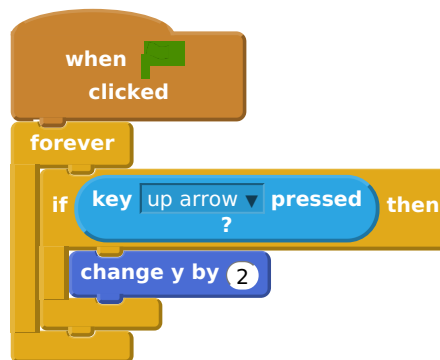
Let's start by creating a player that can move around your world.

✓ Activity Checklist

- ☐ Open the 'Create Your Own World' Scratch project online at jump.to/cc/world-go or download from jump.to/cc/world-get and then open if you are using the offline editor.



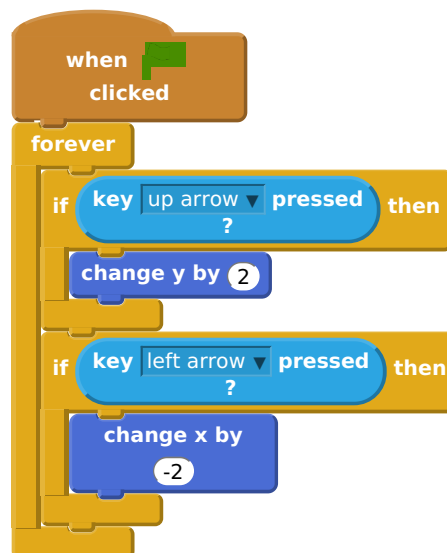
- Let's use the arrow keys to move the player around. When the player presses the up arrow, you want the player to move up, by changing its y coordinate. Add this code to the `player` sprite:



- Test out your player by clicking the flag and then holding down the up arrow. Does your player move up?



- ☐ To move the player to the left, you need to add another **if** block to your player, which changes the x coordinate:



Challenge: Moving in all four directions

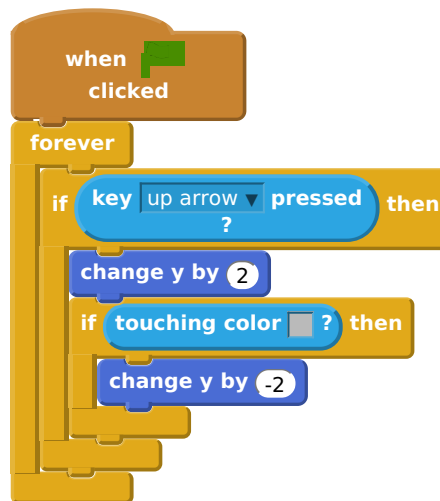
Can you add more code to your player, so that they can move up, down, left and right. Use the code you already have to help you!

Save your project

- Test out your player again, and you'll see they have the ability to walk through the light grey walls.



- To fix this, you need to move the player, but then move them back if they're touching a light grey wall. Here's the code you'll need:



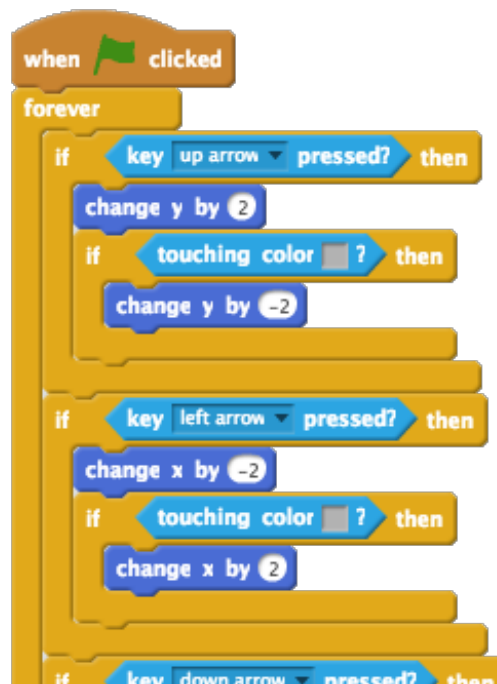
Notice that the new `if touching color` block is *inside* the `if key [up arrow]` block.

- Test this new code by moving below the wall - you shouldn't be able to move up into it.





Let's do the same for the left arrow, moving back if the player is touching a wall. This is how your player code should look so far:



Challenge: Fixing your player's movement

Add code to your player so that you can't walk through walls in any direction. Use the code you already have to help you!

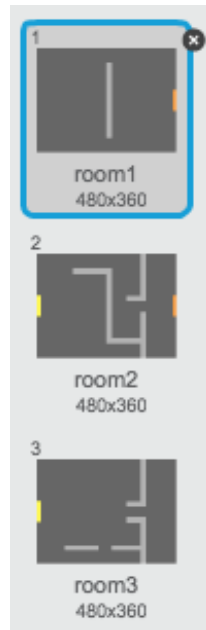
[Save your project](#)

Step 2: Coding your world

Let's allow the player to walk through doors into other rooms!

✔ Activity Checklist

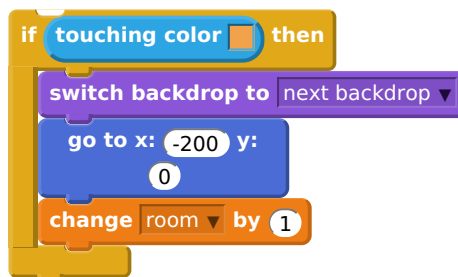
- ☐ Your project contains backdrops for additional rooms:



- ☐ You'll need a new 'for all sprites' variable called **room**, to keep track of what room the player is in.



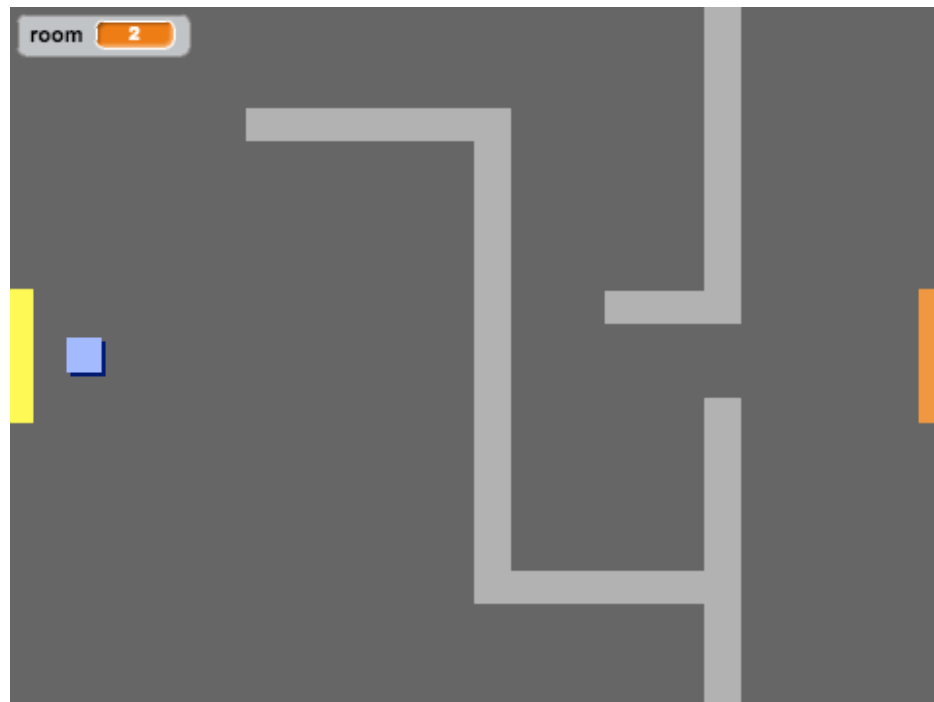
- ☐ When the player touches the orange door in the first room, the next backdrop should be displayed, and the player should move back to the left side of the stage. Here's the code you'll need - it should go inside the player's **forever** loop:



- ☐ Add this code to the *start* of your player code (before the **forever** loop) to make sure that everything is reset when the flag is clicked:

```
set room to 1
go to x: -200 y: 0
switch backdrop to room1
```

- ☐ Click the flag and move your player over the orange door. Does your player move to the next screen? Does the **room** variable change to 2?



Challenge: Moving to the previous room

Can you make your player move to the previous room when they touch a yellow door? Remember that this code will be very similar to the code you've already added for moving to the next room.

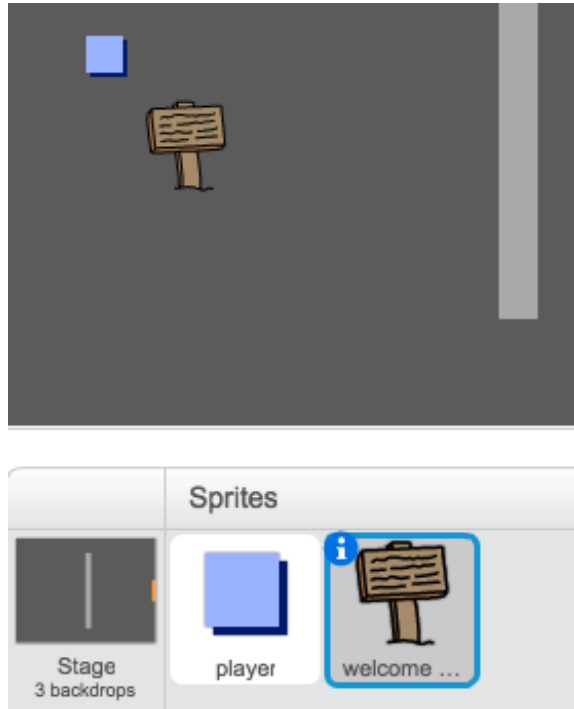
Save your project

Step 3: Signs

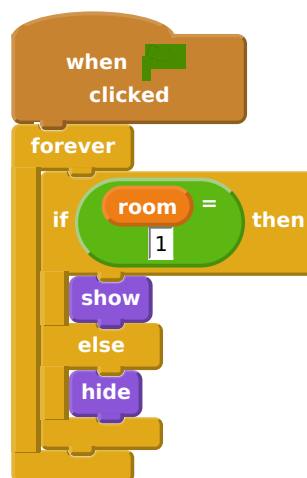
Let's add signs to your world, to guide your player on their journey.

✓ Activity Checklist

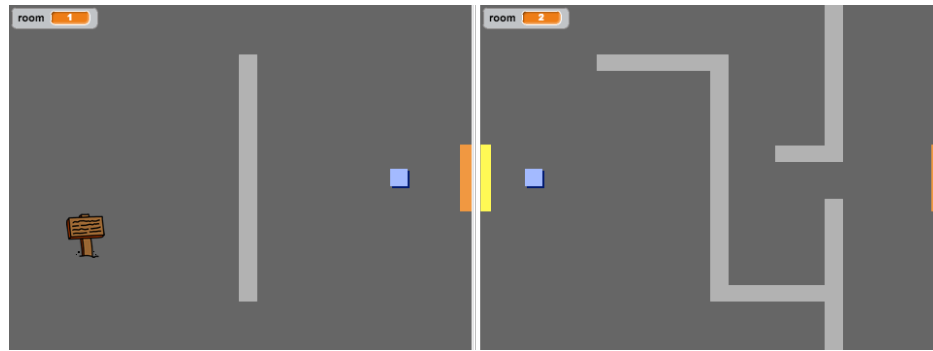
- ☐ Your project includes a welcome sign sprite:



- ☐ This sign will only be visible in room 1, so let's add some code to the sign to make sure that this happens:



- ☐ Test your sign by moving between rooms. Your sign should only be visible in room 1.



- A sign isn't much good if it doesn't say anything! Let's add some more code (in another separate block) to display a message if the sign is touching the player:



- Test out your sign, and you should see a message when the player touches it.

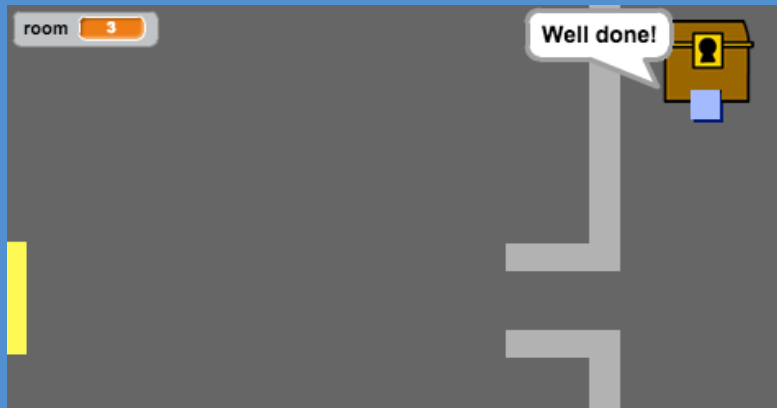


Save your project

Challenge: Treasure!

Right-click on the treasure chest sprite and choose 'show'.

Can you make the treasure chest sprite appear only in room 3 and say 'Well done!' when the player touches it.



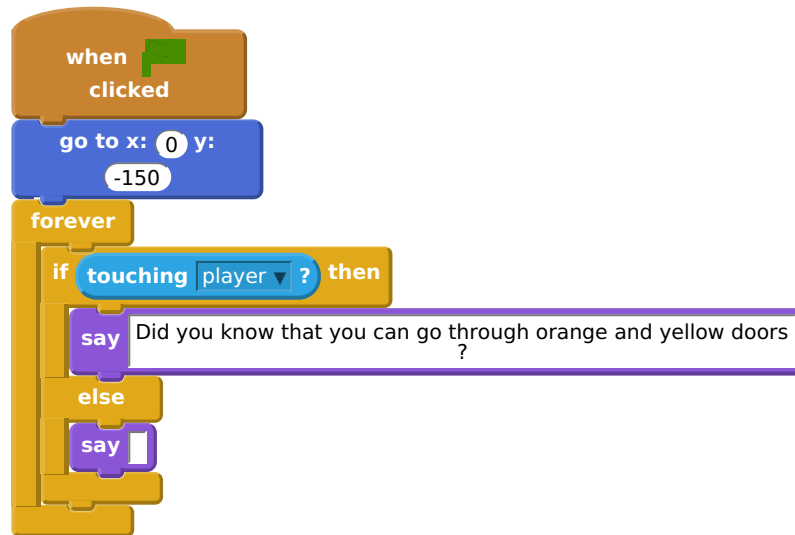
Save your project

Step 4: People

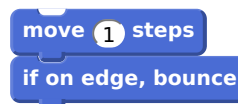
Let's add other people to your world that your player can interact with.

✓ Activity Checklist

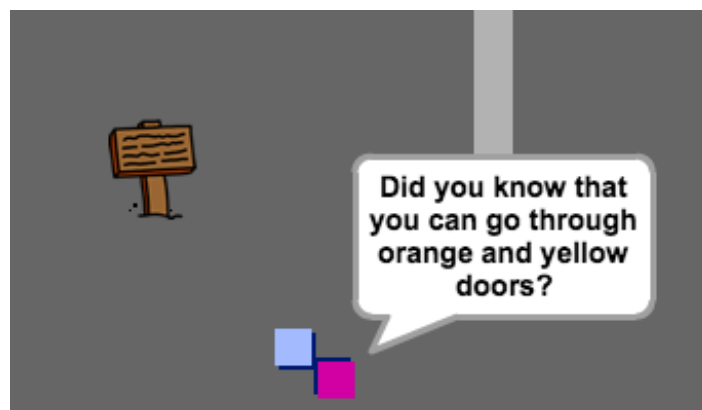
- ☐ Add this code to the person sprite, so that the person talks to your player. This code is very similar to the code you added to your sign:



- ☐ You could also allow your person to move, by using these two blocks:



Your person will act differently, depending on whether you place this code inside the **forever** loop or the **if** block. Try both and see which you prefer.



- ☐ Have you noticed that your person flips upside-down. To stop this, click the sprite's information icon (**i**), and click the dot to fix to rotation style.

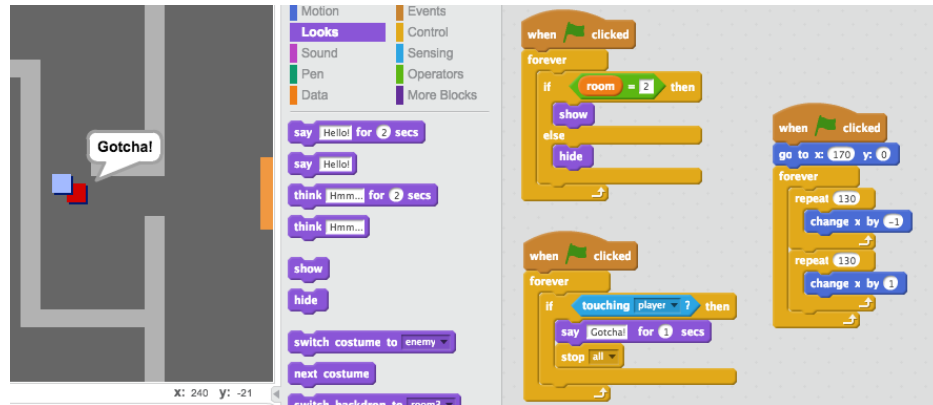


Challenge: Improving your person

Can you add code to your new person, so that they only appear in room 1? Make sure you test out your new code!

Save your project

- ☐ You can also add in patrolling enemies, who end the game if the player touches them. Add in a new enemy sprite, and change the rotation style, just like you did with the 'person' sprite.
- ☐ Add code to your enemy, so that they only appear in room 2.
- ☐ You'll also need to add code to move the enemy, and to end the game if the enemy touches the player. It's easier to do this in separate code blocks. Here's how your enemy code should look:



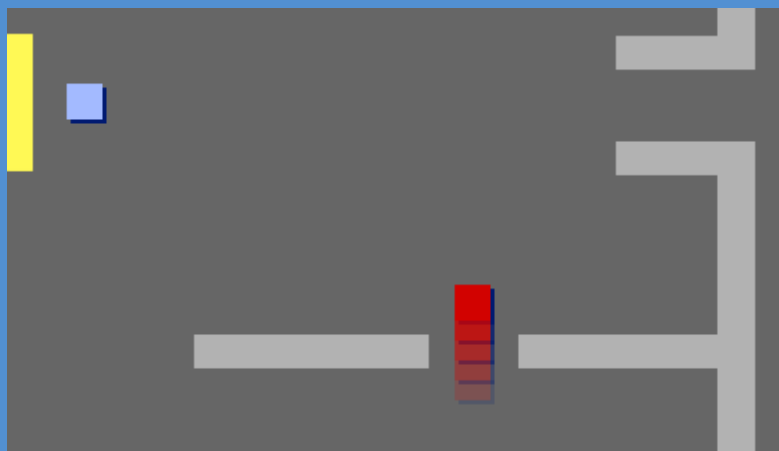
Test out your enemy, to make sure that:

- It's only visible in room 2;
- It patrols the room;
- The game ends if the player touches it.

Save your project

Challenge: More enemies

Can you create another enemy in room 3, that patrols up and down through the gap in the wall?



Save your project

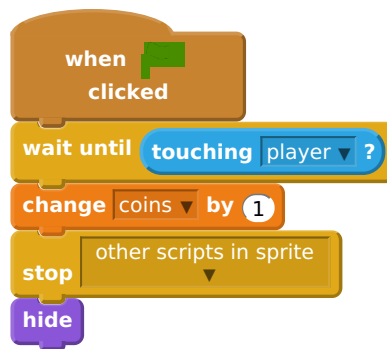
Step 5: Collecting coins

✓ Activity Checklist

- ☐ Add a new variable called **coins** to your project.
- ☐ Right-click on the 'coin' sprite and choose 'show'.



- ☐ Add code to your coin, so that it only appears in room 1.
- ☐ Add code to your coin sprite, to add 1 to your **coins** once they've been picked up:



The code **stop other scripts in sprite** is needed so that the coin stops being displayed in room 1 once it's been collected.

- ☐ You'll also need to add code to set your **coins** variable to 0 at the start of your game.
- ☐ Test your project - collecting your coins should change your score to 1.

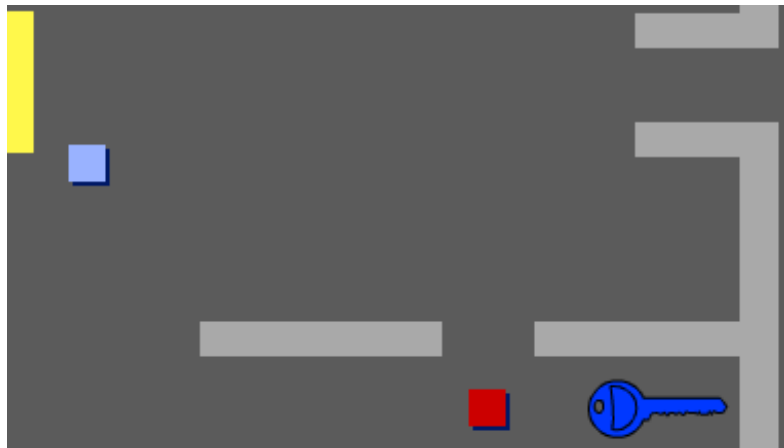
Challenge: More coins

Can you add more coins to your game? They can be in different rooms, and some coins could even be guarded by patrolling enemies.

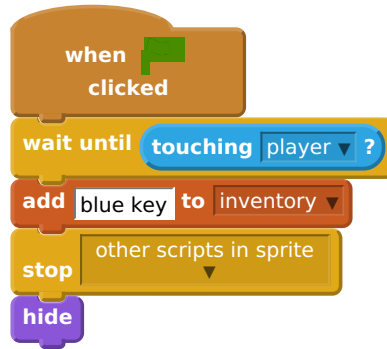
Step 6: Doors and keys

✓ Activity Checklist

- ☐ Edit the key sprite's costume so that it's blue. Right-click the key sprite and choose 'show' so that it appears on the stage. Switch your stage to backdrop 3, and place the key somewhere difficult to reach!



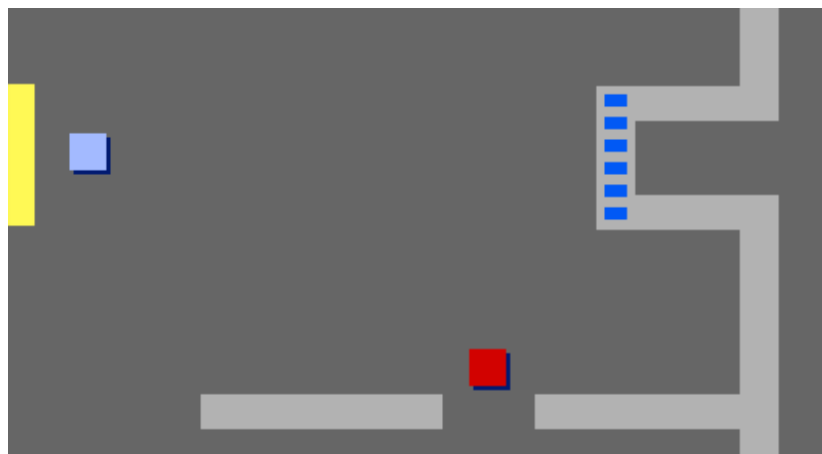
- ☐ Make sure that your key is only visible in room 3.
- ☐ Create a new list variable called **inventory**. This will be where you store all of the items your player collects.
- ☐ The code for collecting the key is very similar to the code for collecting coins. The difference is that you add the key to your inventory.



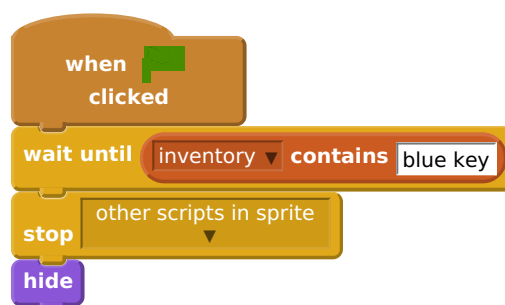
- ☐ Test out your key, to see if you can collect it, and add it to your inventory. Remember to add code to your stage to empty your inventory at the start.



- ☐ Place your blue door sprite across the gap in the two walls.



- ☐ Add code to your door, so that it is only visible in room 3.
- ☐ You'll need to hide your blue door to allow your player to pass once you have the blue key in your inventory.





Test out your project, and see if you can collect the blue key to open the door!

Save your project

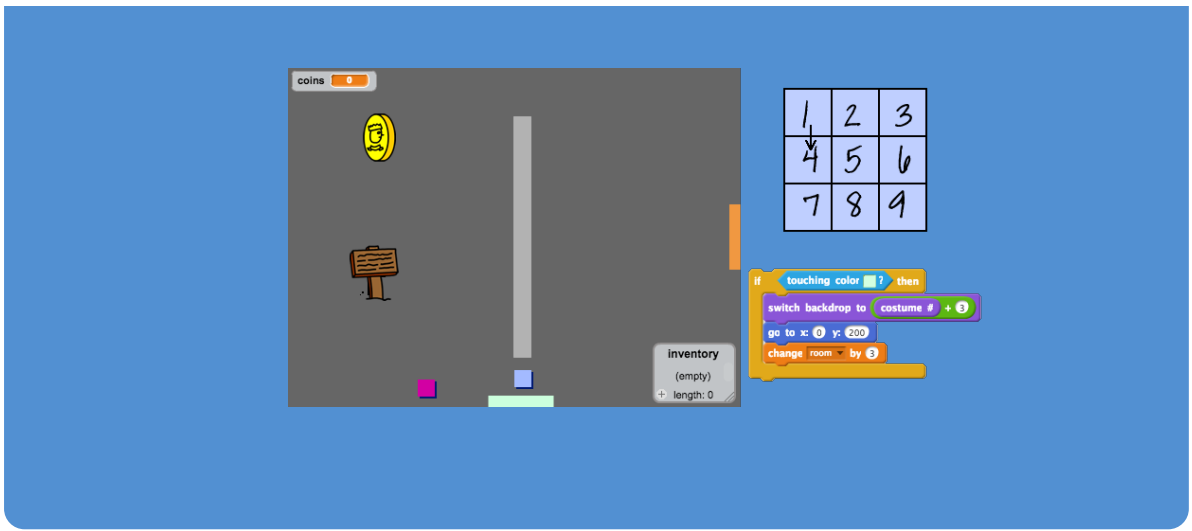
Challenge: Create your own world

You can now continue creating your own world. Here are some ideas:

- ☐ Change the setting of your game, and your game graphics;
- ☐ Add sound and music to your game;
- ☐ Add more people, enemies, signs and coins;
- ☐ Add red and yellow doors, that need their own keys to open them;
- ☐ Add more rooms to your world;
- ☐ Add other useful items to your game;
- ☐ Use coins to get information from other people;



- ☐ You could even add north and south doors, so that the player can move between rooms in all 4 directions. For example, if you had 9 rooms, you could think of them as being in a 3x3 grid. You can then add 3 to the room number to move down 1 level.



Save your project