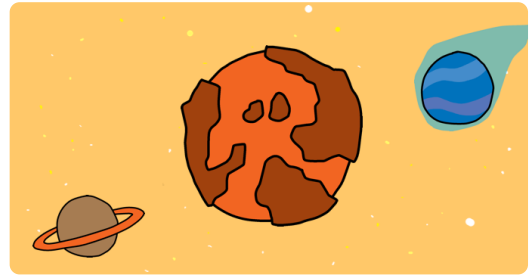


Create your own world

Create your own open-world adventure game

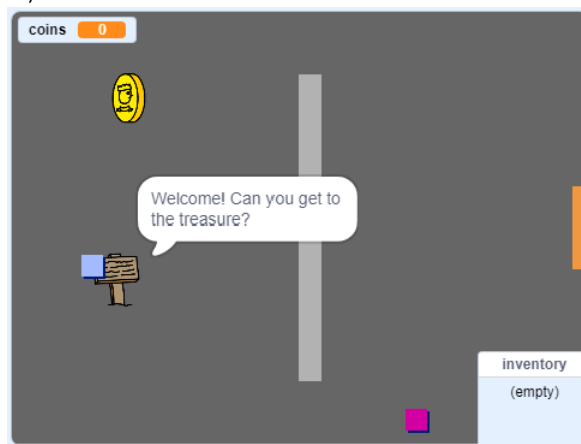


Step 1 Introduction

In this project, you'll learn how to create your own adventure game world with multiple levels to explore.

What you will make

You'll use the arrow keys to move your character around in the world.



What you will need

Hardware

- A computer capable of running Scratch 3

Software

- Scratch 3 (either **online** (<http://rpf.io/scratchon>) or **offline** (<http://rpf.io/scratchoff>))

Downloads

You can find everything you need to complete this project at rpf.io/p/en/create-your-own-world-go (<https://rpf.io/p/en/create-your-own-world-go>).



What you will learn

- Use conditional selection to react to key presses
- Use variables to store a game's state
- Use conditional selection based on the value of a variable
- Use lists to store data



Additional information for educators

If you need to print this project, please use the **printer-friendly version** (<https://projects.raspberrypi.org/en/projects/create-your-own-world/print>).

You can find the **completed project here** (<https://rpf.io/p/en/create-your-own-world-get>).

Step 2 Move the player sprite

Start by creating a `player` sprite that can move around your world.

Open the 'Create your own world' Scratch starter project.

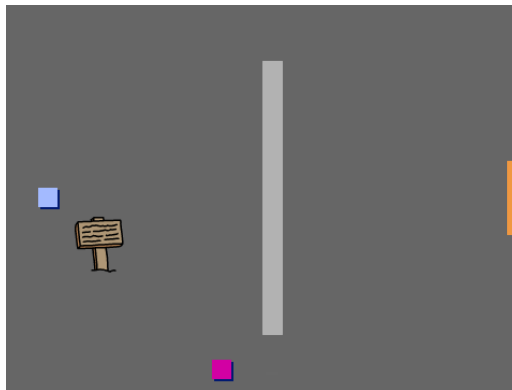


Online: open the online starter project at rpf.io/create-your-own-world-on (<http://rpf.io/create-your-own-world-on>).

If you have a Scratch account you can make a copy by clicking **Remix**.

Offline: download the starter project rpf.io/p/en/create-your-own-world-go (<http://rpf.io/p/en/create-your-own-world-go>), and then open it using the offline editor.

If you need to download and install the Scratch offline editor, you can find it at rpf.io/scratchoff (<http://rpf.io/scratchoff>).



Pressing the arrow keys should move the `player` sprite around. When the up arrow is pressed, the `player` sprite should move upwards on the Stage in response.

Add this code to the `player` sprite:



```
when green flag clicked
  forever loop
    if key up arrow pressed? then
      point in direction 0
      move 4 steps
```

Click the flag and then hold down the up arrow. Does the `player` sprite move up?



To move the `player` sprite to the left, you need to add another `if` block with similar code:



```
when green flag clicked
  forever loop
    if key up arrow pressed? then
      point in direction 0
      move 4 steps
    if key left arrow pressed? then
      point in direction -90
      move 4 steps
```

Add more code to your `player` sprite so it can move down and to the right as well. Use the code you already have to help you.



Here is how your code should look:



```
when green flag clicked
  forever loop
    if key up arrow pressed? then
      point in direction 0
      move 4 steps
    if key left arrow pressed? then
      point in direction -90
      move 4 steps
    if key down arrow pressed? then
      point in direction 180
      move 4 steps
    if key right arrow pressed? then
      point in direction 90
      move 4 steps
```

Step 3 Solid walls

Test your `player` sprite again. Do you see that it can walk through the light grey walls.

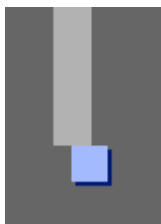


To fix this, you need to make the **player** sprite move back if it touches a light grey wall. Here's the code you need to add inside your **forever** block below the direction blocks:



```
when green flag clicked
  forever loop
    if key up arrow pressed? then
      point in direction 0
      move 4 steps
    if key left arrow pressed? then
      point in direction -90
      move 4 steps
    if key down arrow pressed? then
      point in direction -180
      move 4 steps
    if key right arrow pressed? then
      point in direction 90
      move 4 steps
    if touching color light grey? then
      move -4 steps
```

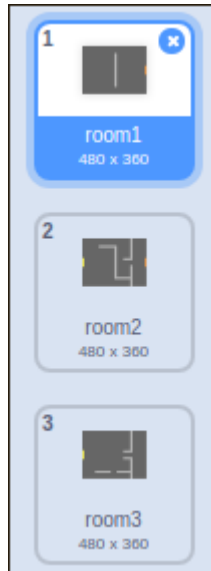
Try to make the `player` sprite move through a wall. If your new code works, this shouldn't be possible.



Step 4 Move around your world

The `player` sprite should be able to walk through doors into other rooms.

Your project contains backdrops for additional rooms:



Create a new 'for all sprites' variable called `room` to keep track of which room the `player` sprite is in.



When the **player** sprite touches the orange door in the first room, the game should display the next backdrop, and the **player** sprite should move back to the left side of the Stage. Add this code inside the **player** sprite's **forever** loop:



```
when green flag clicked
  forever loop
    if key up arrow pressed? then
      point in direction 0
      move 4 steps
    if key left arrow pressed? then
      point in direction -90
      move 4 steps
    if key down arrow pressed? then
      point in direction -180
      move 4 steps
    if key right arrow pressed? then
      point in direction 90
      move 4 steps
    if touching color [grey] ? then
      move -4 steps
    if touching color [orange] then
      switch backdrop to next backdrop
      go to x: -200 y: 0
      change room by 1
```

Every time the game starts, the room, character position, and backdrop need to be reset.



Add code to the **start** of your **player** sprite code above the **forever** loop, to reset everything when the flag is clicked:

Here's what your finished script should look like:



when  clicked

set room to 1

go to x: -200 y: 0

switch backdrop to room1

forever

if key up arrow pressed? then

point in direction 0

move 4 steps

if key left arrow pressed? then

point in direction -90

move 4 steps

if key down arrow pressed? then

point in direction -180

move 4 steps

if key right arrow pressed? then

point in direction 90

move 4 steps

if touching color  ? then

move -4 steps

if touching color  then

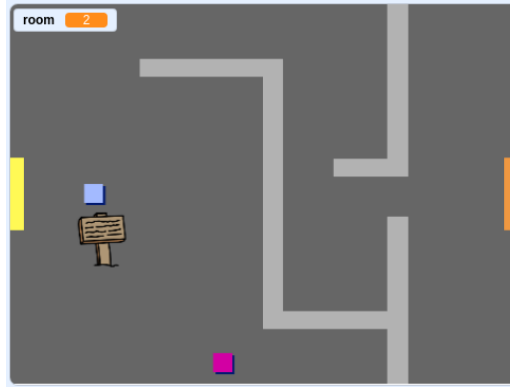
switch backdrop to next backdrop

go to x: -200 y: 0

change room by 1



Click the flag, and then move your **player** sprite until it touches the orange door. Does the sprite move to the next screen? Does the **room** variable change to 2?





Challenge!

Challenge: move back to the previous room

Can you make your `player` sprite move back to the previous room when it touches a yellow door? The code you need for this is very similar to the code you've already added for make the sprite move to the next room.

Step 5 Signs

Now add signs to your world to guide players on their journey.

Your project includes a `welcome sign` sprite:



The **welcome sign** sprite should only be visible in room 1, so add some code to the sprite to make sure that this happens:

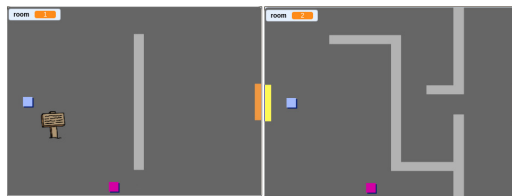


Here is the complete code:



```
when green flag clicked
  forever loop
    if room = 1 then
      show
    else
      hide
```

Test the code for your **welcome sign** sprite by moving between rooms. The sign should only be visible in room 1.



A sign isn't much good if it doesn't say anything! Add some more code to display a message if the `welcome sign` sprite is touching the `player` sprite:



```
when clicked
  forever
    if room = 1 then
      show
    else
      hide
    if touching player ? then
      say Welcome! Can you get to the treasure?
    else
      say
```

Test your `welcome sign` sprite again. You should now see a message when the `player` sprite touches the `welcome sign` sprite.



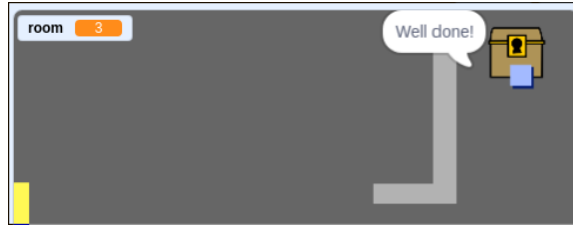


Challenge!

Challenge: treasure!

Can you add some treasure for the player to find?

Make the `treasure chest` sprite appear only in room 3, and have this sprite say 'Well done!' when the `player` sprite touches it.



Step 6 People

Add other people to your world who your **player** sprite can interact with.

Switch to the **person** sprite.



Add some code to the **person** sprite so that the person talks to the **player** sprite. This code is very similar to the code you added to your **sign** sprite:



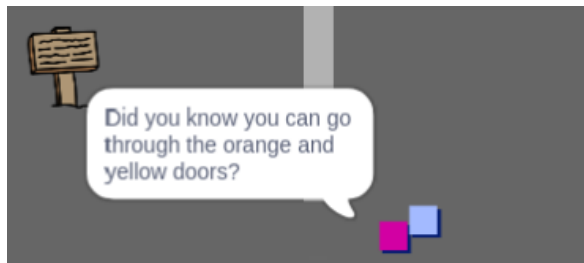
```
when green flag clicked
  go to x: 0 y: -150
  forever
    if touching player ? then
      say Did you know that you can go through orange and yellow doors?
    else
      say [ ]
```

Allow your **person** sprite to move by adding these two blocks in the **else** section of your code:



```
when green flag clicked
  go to x: 0 y: -150
  forever loop
    if touching player then
      say Did you know that you can go through orange and yellow doors?
    else
      say 
      move 1 steps
      if on edge, bounce
```

Your **person** sprite will now move, but will stop to talk to the **player** sprite.



Add code to your new **person** sprite so that the sprite only appears in room 1. The code you need is exactly the same as the code that makes the **sign** sprite only visible in room 1.



Make sure you test out your new code.

Challenge: add an enemy

If you want, you can also add patrolling enemies to your game. If the **player** sprite touches an enemy, the game ends.

- Your game already contains an **enemy** sprite. Add code to the **enemy** sprite so that it only appears in room 2.
- Add code to move the **enemy** sprite and to end the game if the **enemy** sprite touches the **player** sprite. It's easier to do this in separate code blocks. Here's how your **enemy** sprite code might look:

```

when clicked
  forever
    if room = 2 then
      show
    else
      hide
  
```

```

when clicked
  forever
    if touching player ? then
      stop all
  
```

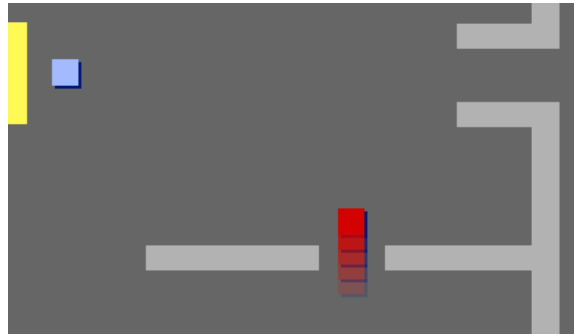
```

when clicked
  go to x: 170 y: 0
  forever
    repeat 130
      change x by -1
    repeat 130
      change x by 1
  
```

- Test out your new code to make sure that:
 - The **enemy** sprite only visible in room 2

- The **enemy** sprite patrols the room
- The game ends if the **player** sprite touches the **enemy** sprite

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



Step 7 Collect coins

Your `player` sprite should be able to collect coins as it moves through the world.

Add a new variable called `coins` to your project.



Select the `coin` sprite and click `show`.



Add code to your `coin` sprite so that it only appears in room 1.



```
when green flag clicked
  forever loop
    if room = 1 then
      show
    else
      hide
```

Add code to your **coin** sprite so that the sprite **hides** and **1** is added to the **coins** variable once the **player** sprite touches the **coin** sprite to 'pick it up'.



```
when green flag clicked
  wait until touching player
  change coins by 1
  hide
  stop other scripts in sprite
```

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

Now add code to the Stage to set your **coins** variable to **0** at the start of the game.



```
when green flag clicked
  set coins to 0
```

Test your game. Collecting a coin should change your **coins** score to **1**.



Step 8 Doors and keys

Now you are going to add code so that some of the doors in your game world are locked, and the player must find the key to open them and get to the next room.

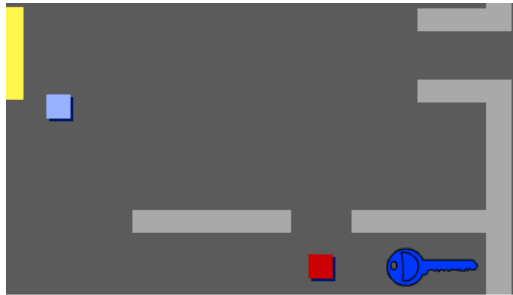
Switch to the `key` sprite. Click on `show` in the Scripts menu so that the sprite appears on the Stage.



Edit the `key` sprite's costume so that it is blue.



Switch your Stage backdrop to room 3, and place the `key` sprite somewhere difficult to reach!



Add code to the `key` sprite to make it only visible in room 3.



Create a new list called `inventory` to store the items your `player` sprite collects.



The code you need to add for collecting the key is very similar to the code for collecting coins. The difference is that you add the key to the **inventory**.



```
when green flag clicked
  wait until touching player ?
  add blue key to inventory
  hide
  stop other scripts in sprite
```

Add code to your Stage to empty your inventory at the start of the game.

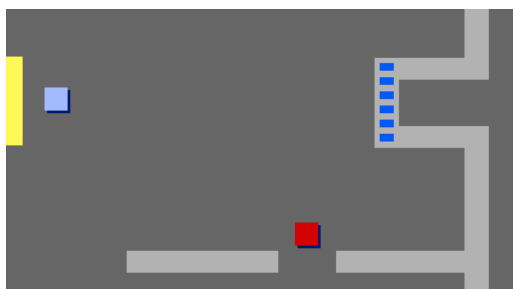


```
delete all of inventory
```

Test out your game to check whether you can collect the **key** sprite and add it to your inventory.



Now add the locked door. Select the **door-blue** sprite and click on **show** in the Scripts menu, and then position the sprite across the gap between the two walls.



Add code to the **door-blue** sprite so that it is only visible in room 3.



Add code to the `door-blue` sprite so that, when the key is in the `inventory`, the sprite `hides` to allow your `player` sprite to pass.



```
when clicked
wait until inventory contains blue key
stop other scripts in sprite
hide
```

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



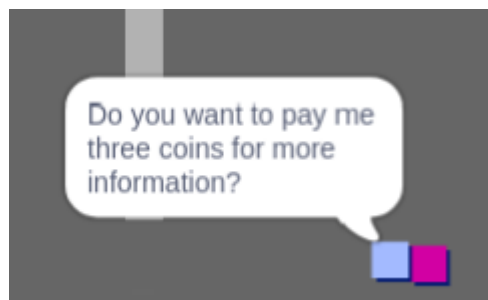


Challenge!

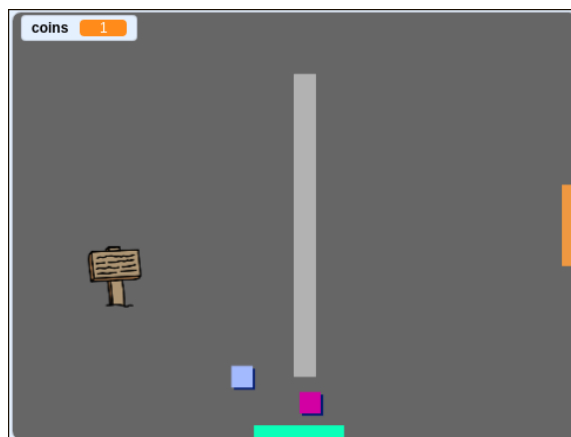
Challenge: extend your world

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

- Add more coins to your game in different rooms. Can you let some coins be guarded by patrolling enemies?
 - Change your game's backdrops
 - Add sound and music to your game
 - Add more people, enemies, and signs
 - Add red and yellow doors, and special 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 doors in the north and south walls of room 1, so that the player can move between rooms in all four directions. For example, your game can have nine rooms in a 3x3 grid. You can then add 3 to the room number to move down one level.

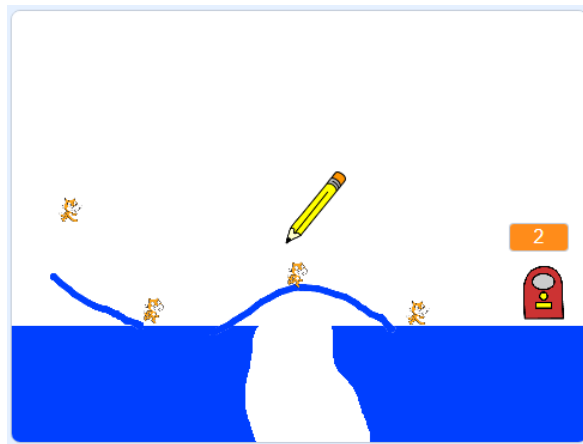


1	2	3
4	5	6
7	8	9

```
if touching color [ ] ? then
  switch backdrop to costume number + 3
  go to x: 0 y: 200
  change room by 3
```

Step 9 What next?

Have a go at creating another game by working through the **CATS!** (https://projects.raspberrypi.org/en/projects/cats?utm_source=pathway&utm_medium=whatnext&utm_campaign=projects) project.



If you want to make a game using Python instead of Scratch, try out the **RPG** (https://projects.raspberrypi.org/en/projects/rpg?utm_source=pathway&utm_medium=whatnext&utm_campaign=projects) project.

Published by **Raspberry Pi Foundation** (<https://www.raspberrypi.org>) under a **Creative Commons license** (<https://creativecommons.org/licenses/by-sa/4.0/>).

View project & license on GitHub (<https://github.com/RaspberryPiLearning/create-your-own-world>).