Goal

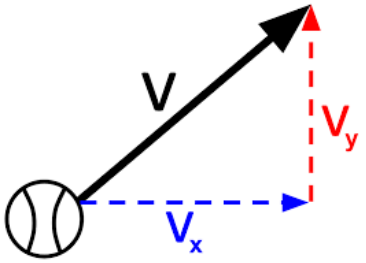
The goal is to collect **10** coins. But you will need to make some changes to the game to be able to collect them all!

Task 1

The first coin is just to your right. Unfortunately, at the moment you can only move left with the **left arrow** key.

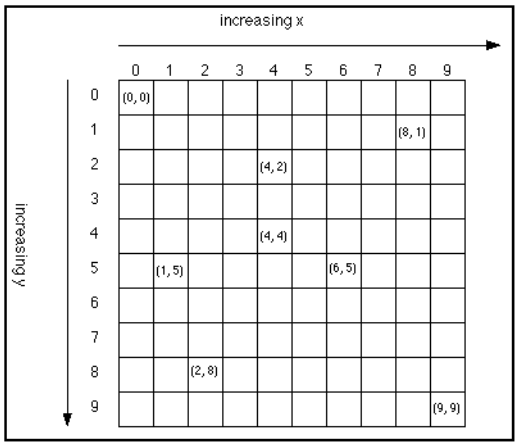
To collect the coin you will need to add code to detect the **right arrow** key and change the players x velocity.

Velocity just means speed with a direction:



A **positive x velocity** means an object is moving to the **right**.
A **negative x velocity** means the object is moving to the **left**.

A **positive y velocity** means an object is moving **down**.
A **negative y velocity** means the object is moving **up**.
This is because for computers graphics, the y-axis points down on the screen:



Steps

1. Open **keyboard.js**.
2. Find the block of code that detects the **left arrow key** being pressed and sets the x velocity to **-3.2**.
3. Copy and paste that block under the comment “Task 1”.
4. Set the x velocity to **3.2** when the **right arrow key** is being pressed.
5. Collect the coin.

Task 2

Good job, you now have the first coin! The second coin is on a platform. You can jump using the **space bar** or **up arrow key**.

Try collecting the coin now. You will have noticed that you can't jump high enough. So let us make the character jump higher.

Steps

1. In **keyboard.js** find the line of code that sets the **y velocity** when the **space bar** or **up arrow key** are pressed.
2. Set the **y velocity** to **-6**, when the character jumps.
3. Collect the coin.

Task 3

Two coins collected now, you're on a streak. You probably noticed the third coin is floating in the air above a canyon. To reach this we will create a platform that bridges the canyon.

Steps

1.In **world.js c**opy and paste the existing line for creating a platform under "Task 3":

```
platforms.push(new Platform(1500, 450, 200, 20));
```

The word **platforms** is a list containing all of the platforms in the game. And we are adding (or pushing) a new **Platform** object to that list.

The part you will need to change is the numbers. These are called **arguments**. They have special meanings as described below:

Argument Position	Name	Description
1st	x	The x co-ordinate of the top left of the platform.
2nd	y	The y co-ordinate of the top left of the platform.
3rd	width	The width of the platform in pixels.
4th	height	The height of the platform in pixels.

3. Change the line you pasted so that the platform is at the following position:

x: 2050
y: 530

Give the platform the following width and height:

width: 600
height: 30

4. Collect the coin.

Task 4

That's 3 coins in the bag now. To collect the next coin we will use a different type of platform that can move.

The new type of platform is called a **MovingPlatform**. It is the same as a **Platform** but it takes a few extra arguments:

Argument Position	Name	Description
1st	x	The x co-ordinate of the top left of the platform.
2nd	y	The y co-ordinate of the top left of the platform.
3rd	width	The width of the platform in pixels.
4th	height	The height of the platform in pixels.
5th	range	The distance the platform travels in pixels.
6th	speed	How fast the platform moves in pixels/frame.
7th	direction	The direction the platform moves in. This can have the values HORIZONTAL or VERTICAL.

Steps

1. Under the line for task 4, type the code to create a **MovingPlatform**:

    platforms.push(new MovingPlatform*(<x>, y>, <width>, <height>, <range>, <speed>, <direction>*));

2. Change the parameters to the following:

    x: 2980
    y: 500
    width: 50
    height: 20
    range: 400
    speed: 2
    direction: VERTICAL

3. Collect the coin.

Task 5

The fourth coin is over a canyon again. This time, instead of using a bridge we will reach it with a horizontally moving platform.

Steps

1. Under the “Task 5” comment in **world.js,** copy and paste the line you added for the **MovingPlatform** as part of task 4.
2. Change the parameters so that it moves across the canyon in a HORIZONTAL direction. I won’t give you the values this time, but you can use the table in task 4 to help work them out.
3. Hopefully you have a platform that moves across the canyon, and you can collect the coin now.

Task 6

Ok that’s 5 coins collected, halfway to the goal. The 6th coin is guarded by a slime. To get to the coin we will place a small canyon so that the slime falls into it, letting us reach the coin safely.

Steps

1. Find an example for creating a **Canyon**, then copy and paste it under the “Task 5” comment.

2. The **Canyon** object takes the following arguments:

Argument Position	Name	Description
1st	x	The x co-ordinate of the left side of the canyon.
2nd	width	The width of the canyon.

Place a canyon with **width 50**, just under the slime.

You will need to work out the value for **x** yourself using trial and error.

3. The slime should fall into the canyon now, allowing you to safely collect the coin.

Task 7

Only 3 more coins to collect! The next one is sat on top of a platform. We are going to position the platforms so we can use them as steps to reach it.

Steps

1. Find the for loop under the “Task 7” comment. The loop repeats 3 times, and adds 1 to the value of **i** each time. We can use the value of **i** to calculate an **offset** to change the position of each **Platform** that it creates.

2. Use **i** to create a variable called **yOffset**, whose value changes by **100** each time the loop runs.

   HINT: Look at how **xOffset** is set, the code will be very similar.

3. Subtract **yOffset** from the **y co-ordinate** argument of the **Platform** that is inside the for loop. This is similar to how **xOffset** is added to the **x co-ordinate.**

   NOTE: We are subtracting **yOffset**, because we want the platforms to be progressively higher. And in computer graphics a smaller y value is further up the screen.

4. You should be able to collect the coin now.

Task 8

Great work, you're almost at the goal of 10 coins now. The 8th coin is guarded by 3 tightly grouped slime monsters. To get to the coin we will space them out so we can easily jump over them.

Note

1. Find the for loop for creating the 3 slimes under the "Task 8" comment.

2. An **Enemy** patrols between two positions, **x minimum** and **x maximum.**

   It takes the following arguments:

Argument Position	Name	Description
1st	x minimum	The x co-ordinate of how far left the enemy will patrol to.
2nd	x maximum	The x co-ordinate of how far right the enemy will patrol to.
3rd	y position	The y co-ordinate of the enemy.
4th	speed	How fast the enemy moves in pixels/frame.

   Update the arguments to add an **offset** to the **x minimum** and **x maximum** arguments, so that the enemies will be more spaced out.

The value of **offset** should increase by **300** each time the loop runs.

HINT: Look at how we applied an offset in the for loop from the previous task.

3. You should now be able to jump over the enemies to reach the coin.

Task 9

Only 1 more coin left to collect after this one! This coin is sitting on a platform in the air and is guarded by a slime.

Steps

1. Use any of the techniques that you have learnt to collect the coin!

Task 10

The final coin! Unfortunately, it does not exist in the world. Place your own coin to reach the final goal of 10 coins.

Steps

1. Place a **Coin** anywhere in the world that is easy to reach. A **Coin** object takes the following arguments:

Argument Position	Argument	Description
1st	x	The x co-ordinate of the coin.
2nd	y	The y co-ordinate of the

		coin.