CoderDojo Stirling

Scratch Tutorial: Shooter Game

Attack of the Bug-Eyed Fish Monsters

Alexander Drummond

V0.1 (Draft)

Contents

Challenges

1 Introduction

In this tutorial you will work through the development of a simple Scratch Shooter game called “Attack of the Bug-Eyed Fish Monsters”. This will show you the following areas of Scratch: -

Events – Reacting to pre-defined events and messages broadcast from other sprites.

Messages – Broadcasting messages between sprites.

Sensing – Collision detection and capturing the position of other sprites.

Variables – Setting Global and Local variables.

Cloning – Creating multiple copies of a single sprite

You should be able to complete this after completing the Hour of Code tutorial. If you have any questions at all, regarding this tutorial, just get the attention of a Mentor.

The Template for the Game for the project is available here: -

https://scratch.mit.edu/projects/201287695/

Remix this game, and save it. When saving the game, save it as **<your name>_BugEye_game.sb2.**

2 Setting up the sprites

2.1 The Spaceship

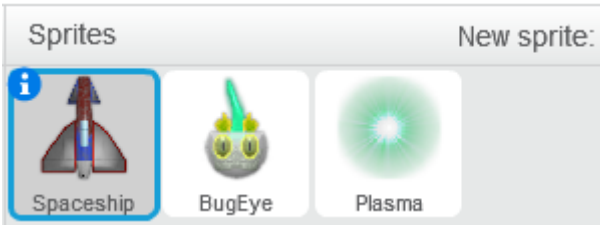
In this section you'll setup your Spaceship.

If you need help at any time during this section, remember to raise your  hand to get help from a mentor.

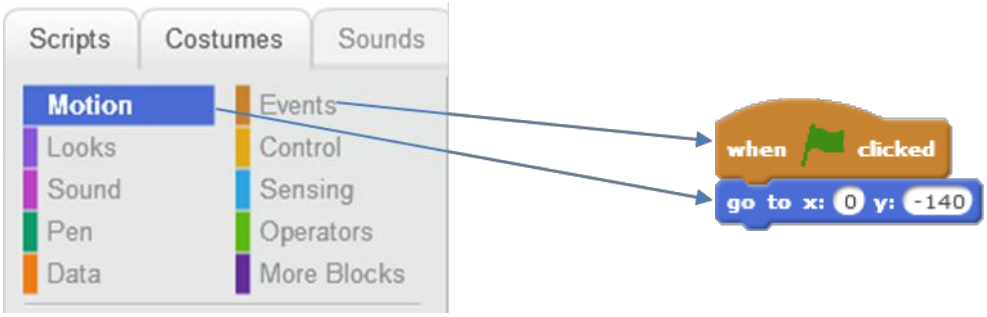
2.1.1 Setting the Start position…

For this game you need to ensure that your Spaceship always starts at the same position when the game starts.

Select the Spaceship sprite: -



To ensure the Spaceship returns to it's starting position every time you start the program, the first thing you'll do is set the Spaceship's "x" and "y" positions on the screen, in Scratch the centre of the screen is x=0, y=0 (0,0). The starting position of the Spaceship will be x=0, y=-140 (0,-140), this places the spaceship just above the bottom, and half way across the screen from left to right.

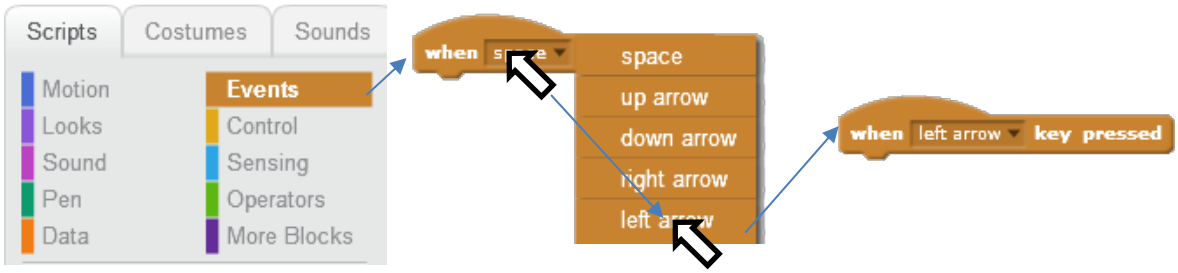


Now every time the green flag is clicked the Spaceship will return to its start point.

2.1.2 Moving the Spaceship

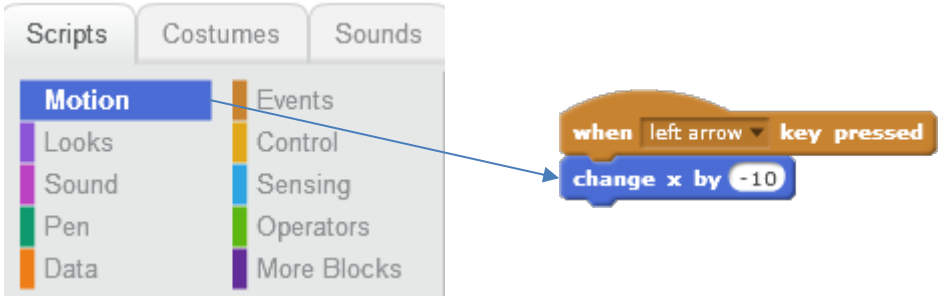
Now you'll make the Spaceship move left and right. To do this you need to know when a key is pressed, and you need to make sure that your code reacts to the correct key. The Event used for this is: -

To move the Spaceship, you'll capture 2 new events one for the "Left Arrow" key being pressed, and the other for the "Right Arrow" key being pressed. Drag the "when … key pressed" on to the script workspace. Click on "space" and select "left arrow" from the list.



You now want to move the space ship left by 10 every time the left arrow is pressed. To move left by 10, you need to subtract 10 from the current "x" position, so we need to change "x" by -10.

From the Motion menu take the "change x by" instruction and attached it to the key pressed event. Select the value and update it to "-10".



Now test your program by clicking the 

Press the left arrow key on the keyboard and the Spaceship should move left.

Now SAVE your work.

Challenge 1 – Move Right

Make the Spaceship move right when the Right Arrow Key is pressed.

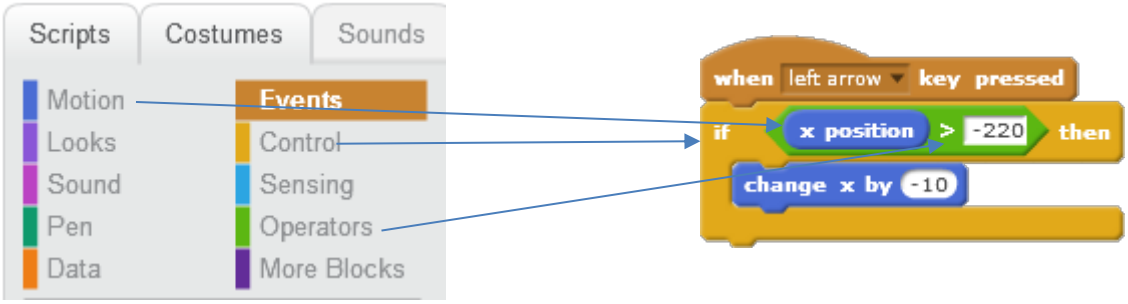
Clue: You can duplicate the "When key pressed" event you've just done, and update it.

If you need some help… 

Remember to SAVE your work.

2.1.3 Stopping the Spaceship moving off the screen…

You may have noticed that the Spaceship can move off the edge of the screen. To stop this, you need to make sure that the Spaceship stops when it reaches the edge of the screen. To do this you need to update the "when left arrow key pressed" event. Detach "change x by -10" from the event and insert the following ***if*** condition. Place the "change x by -10" inside the if condition.



Challenge 2 – Set the right boundary

Set the boundary for the right-hand side of the screen. The ***x position*** must be less than (<) 220.

If you need some help… 

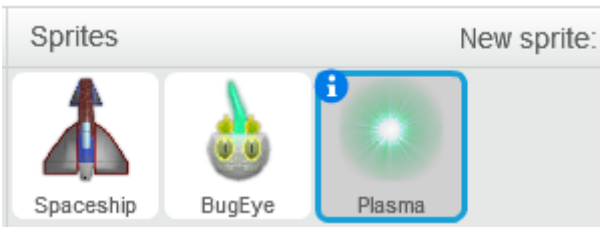
Remember to SAVE your work.

Now test your program by clicking the 

2.2 The Plasma Torpedo

2.2.1 Starting position

Now select the Plasma Sprite.



Like the Spaceship, we want to set the starting position of the Plasma Torpedo. The starting position for this sprite is x: 0, y: -124. The Plasma Torpedo should also be hidden, we don't want it to appear until it's triggered, more about that later.

Challenge 3 – Starting point for the Plasma Torpedo

Set the starting position of the Plasma Torpedo when the green flag is clicked and hide it. You'll find ***hide*** in the **Looks** section.

Need help?

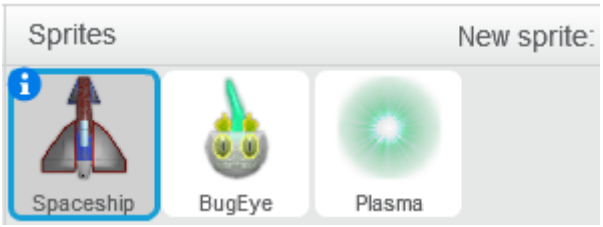
Remember to SAVE your work.

Now test your program by clicking the

You should no longer see the Plasma Torpedo.

2.2.2 Firing the Torpedo (Introduction to Sprite messages)

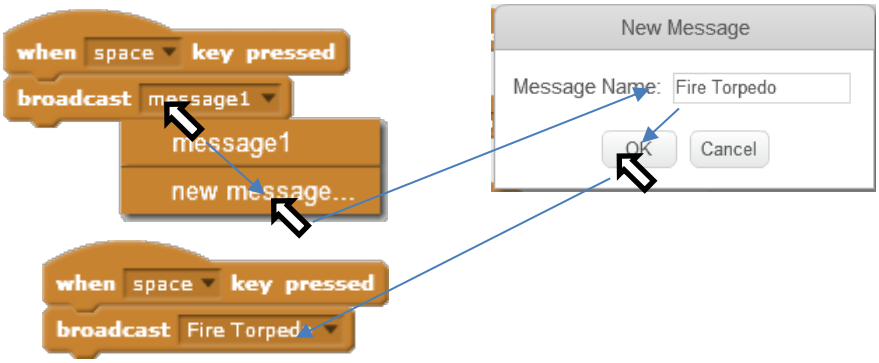
Now you need to be able to fire the Plasma Torpedo. After all, you need to blast the Bug-Eyed Fish Monsters with something. It is the Spaceship that fires the Torpedo, so you need to start with the Spaceship. In this section you will broadcast a message called "Fire Torpedo", and the Torpedo will react to the message.



Setup a Key pressed event and set it to space: -

In Scratch a Sprite can send (broadcast) a message that other Sprites can react to. This is how you will tell the Plasma Torpedo to fire. Go to the Events section and attach. A "broadcast" event to the bottom of the when *space* key pressed event. Select "message1" and click "new message…". Call the new message "Fire Torpedo".[1]

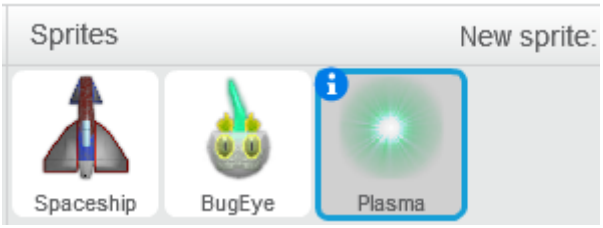
[1] This could be done using a Key pressed event on the Plasma sprite. But the aim here is to introduce you to Sprite messaging.



Remember, if you need help…

Remember to SAVE.

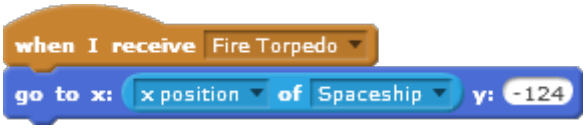
Now select the Plasma Sprite.



You must now make the Torpedo react to the "Fire Torpedo" message. To get a sprite to react to a message we use the "when I receive…" event, from the Events section

The Spaceship may have moved since you last time you pressed the space bar, or this may be the first time you've pressed the space bar. You must make sure the Torpedo moves to the Spaceship's x position before it's shown. This could be done using variables, but you will use a block in the Sensing section to capture the Spaceship's position, it's less complicated.

First attach a "go to" block from the Motion section to the "When I receive Fire Torpedo", as you did for the Spaceship, and the torpedo when the game starts. Set the y position to -124. Now select the Sensing Section, you should see a block "***x position*** of **…**" drag it into the x setting for the "go to" and select the Spaceship. This will ensure that the Torpedo starts at the Spaceships position.



Now make it play a sound. Select the **Sound** section and drag the "play sound" block and attached it below the goto. I've used the "zoop" sound, but feel free to use what you like.

Challenge 4 – Shoot the Torpedo to the top of the Screen.

Show the torpedo, then make it ***glide*** to the top of the screen in ***1*** second and ***hide*** it again. Everything you need is in the **Motion** and **Looks** sections. Remember only the y position will change.

If you need some help… 

Remember to SAVE your work.

Now test your program by clicking the 

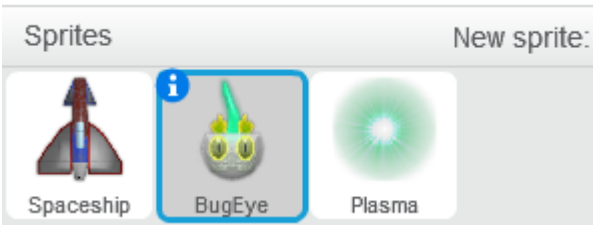
2.3 The Bug-Eyed Fish Monster

2.3.1 Overview of Controlling the Bug-Eye

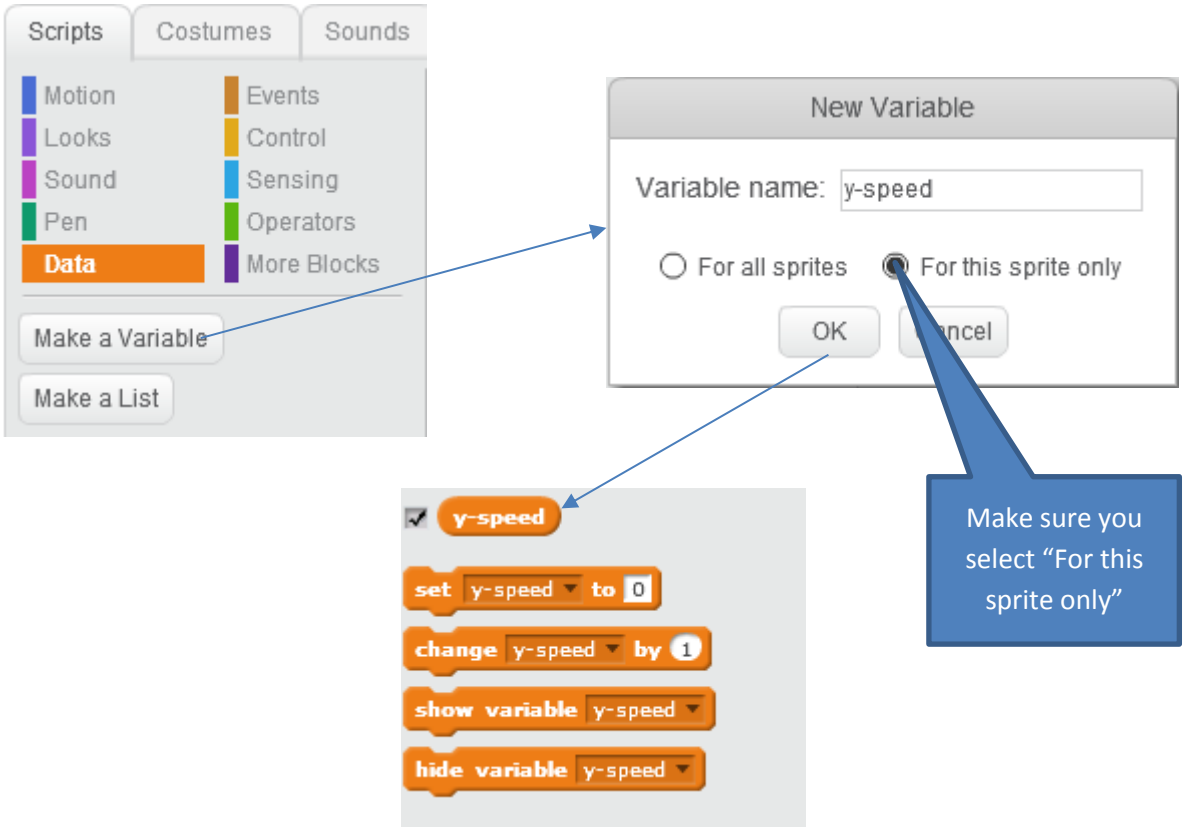
The Bug-Eye is a little more complicated than the Spaceship and the Plasma, but don't worry you'll manage to set it up easily. The difference between the Spaceship/Plasma and the Bug-Eye is the Spaceship and Plasma are under the player's control and reacting to key presses. But the programmer controls the movement of the Bug-Eye. You will just setup the Bug-Eye to move from the top of the screen to the bottom. But to make it a bit more difficult for the player you'll make it start from a random x position and travel at a random speed. The benefit of this setup will be shown off in a later section when we clone the Bug-Eye… but more about that later.

2.3.2 Setting up the Bug-Eye

Select the Bug-Eye:



Before you start programming the Bug-Eye you'll need to create a variable that will only be used by the Bug-Eye sprite. This variable will be called "y-speed". A variable is used to store a value that can change, you can then use the variable name instead of a value in your code. A good example of a variable is the Score in a game. But just now we are going to use the "y-speed" variable to change the speed of the Bug-Eye, each time it starts at the top of the screen.



You'll see that there is a little tick next to the variable "y-speed", this displays the value of the variable on the Screen. Remove the tick by clicking on it, this removes the variable from the screen.

Because the Bug-Eye won't be reacting to keys presses and you want the Bug-Eye to start again after it reaches the bottom of the screen, you'll need to put everything inside a forever loop.

Using what you've learned from the previous sections you should know enough to complete the rest of the Challenges in this section. But just raise your hand and get the attention of a Mentor if you're not sure.

Challenge 5 - Set the starting point of the Bug-Eye

You've already setup the starting point for the Spaceship and the Plasma Torpedo, so you should be able to do the following for the Bug-Eye. Remember to do it inside the *forever* loop: -

- Hide the Bug-Eye.
- Goto position x: 0 and y:185
- Set the y-speed variable to -5 (you haven't set a variable in this tutorial yet, but you'll find the set block in the **Data** section, just put your hand up if you get stuck.).

If you need some help… 

Remember to SAVE your work.

Now test your program by clicking the 

You've setup the starting point for the Bug-Eye and now you want to make it move. You could do something like the movement you did for the Plasma Torpedo, and glide from the top of the screen to the bottom of the screen, but you need more control over the movement of the Bug-Eye and when it needs to restart. For this you will use another loop inside the forever loop, a ***repeat until*** loop.

But before you set up the loop, you will create a new variable called ***restart.*** This will be used as a flag to exit the loop when the Bug-Eye needs to be restarted. You won't just restart the Bug-Eye when it reaches the bottom of the screen, it could be hit by a Plasma Torpedo or it could have hit the Spaceship. More about collisions later though. So, the easiest way to exit the repeat until loop is to create a flag.

The flag, ***restart***, you are creating here is just a variable, there is nothing special about it. But you will only ever set it to one of two values, 1 or 0. 1 means restart (or ***restart*** is true) and 0 means don't restart (or ***restart*** is false).[2]

Challenge 6 - Setup the Bug-Eye movement loop

In this challenge you'll setup a variable called ***restart*** in the same way you did for ***y-speed.*** The make the following changes: -

- Before the ***repeat until*** loop set restart to 0.
- Show the Bug-Eye.
- Set up the ***repeat until*** and loop until restart = 0. This will use the same type of operator you used for the Spaceship to stop it going off the edge of the screen.

[2] For those that are interested, a flag is usually represented by a bit (binary digit) and it can only ever have a value of 1 (true/on) or 0 (false/off).

If you need some help… 

Remember to SAVE your work.

Now test your program by clicking the 

Challenge 7 - Moving the Bug-Eye

You've already used ***change x by*** when moving the Spaceship, and you've prevented the Spaceship moving off the end of the screen, so try and work out how to do the following, all this code should go inside the ***repeat until*** loop: -

- Change the ***y position*** by ***y-speed***
- When the ***y position*** is less than -185 ***restart*** the Bug-Eye.

 if you need some help.

SAVE your work.

Now test your program by clicking the 

This is going to be a very easy game if the Bug-Eye just drops from the centre of the screen at a constant speed. So the next challenge is to make the Bug-Eye move in an un-expected way. Use the following operator to complete the next challenge: -

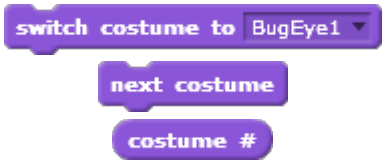
This operator will randomly pick a value between and including the 2 values.

Challenge 8 - Bug-Eye Random movement

This will change where the Bug-Eye will start and how fast it will move in a random way: -

- Change the starting point of Bug-Eye so that it starts at a random position between -200 and 200.
- Change the starting value of ***y-speed*** to between -3 and -10.

Now to finish off the BugEye, you will animate it using the following blocks in the **Looks** section: -



The table below shows the costume numbers for the costumes used to animate the Bug-Eye. This does seem obvious but there are 16 costumes used for the Bug-Eye 8 for the animation and 8 for the explosion.

costume #	Costume
1	BugEye1
2	BugEye2
3	BugEye3
4	BugEye4
5	BugEye5
6	BugEye6
7	BugEye7
8	BugEye8

Challenge 9 - Animate the BugEye

To animate the Bug-Eye you'll have to make the following changes: -

- Before you show the Bug-Eye, set the costume to "BugEye1".
- At the very start of the ***repeat until*** loop, goto the ***next costume***.
- After moving to the next costume check the costume number and if it's greater than 8, set the costume back to "BugEye1".

 if you need some help.

SAVE your work.

Test your work 

WELL DONE!!!! The assets for your game are now setup. Next, we'll add Collisions, Explosions, and a Score.

3 Configuring the Game

3.1 Overview

You'll now start configuring the game it's self. Just to be sure we're starting at the same place. Here's an overview of the code so far. Yours might not be exactly the same.

3.1.1 Spaceship

3.1.2 Plasma Torpedo

3.1.3 Bug-Eye


when clicked
forever
hide
go to x: pick random -200 to 200 y: 185
set y-speed to pick random -3 to -10
set restart to 0
switch costume to BugEye1
show
repeat until restart = 1
next costume
if costume # > 8 then
switch costume to BugEye1
change y by y-speed
if y position < -185 then
set restart to 1

3.1.4 Configuring the Bug-Eye

The BugEye sprite is the source of most of the Action. It is responsible for increasing the score when it's hit and broadcasting hit events to the object that hit it. The reason for this is there will only ever be 1 Plasma Torpedo and 1 Space ship, but by the end of this Chapter there will be a number of Bug-Eyes.

The first challenge you will have to detect a hit from the Plasma Torpedo. To do this you'll use the following block from the **Sensing** Section, everything else you need you've used before.

You can change the ***mouse-pointer*** to a sprite by selecting the sprite in the drop down.

Challenge 10 - Plasma Torpedo Collision

In this challenge you'll detect the collision with a Plasma Torpedo. You will also create a new sprite only variable called Monster Hit.

- Create a a new variable called ***MonsterHit*** set it to 0 just after you set restart to 0.
- After the ***y position*** check insert an ***if*** block to test if the Bug-Eye is hitting the plasma.
- If it is touching plasma do the following: -
  - Broadcast a ***PlasmaHit*** message, you'll have to create the message.
  - Set the new ***MonsterHit*** variable to 1
  - Set the ***restart*** variable to 1

 if you need some help.

SAVE your work.

Test your work 

Now when the Plasma Torpedo hits the Bug-Eye the Bug-Eye disappears and starts again. Now you'll need to make the Bug-Eye explode when it's hit. To do this you'll need to animate the explosion.

costume #	Costume
9	Explosion1
10	Explosion2
11	Explosion3
12	Explosion4
13	Explosion5
14	Explosion6
15	Explosion7
16	Explosion8

Challenge 11 - Animate the Bug-Eye explosion.

At the bottom of the ***repeat until restart = 1*** loop check for ***MonsterHit*** being set to 1. If it is, set it back to 0 and play a sound, then inside this if loop through the explosion costumes until it gets to the last costume. Now the BugEye should make a sound and explode when it's hit by the Plasma sprite.

 if you need some help.

SAVE your work.

Test your work 

Challenge 12 – Spaceship Collision

Repeat Challenge 10, but this time for SpaceshipHit. Place the Spaceship check under the plasma check.

 if you need some help.

SAVE your work.

Test your work 

You should find it's starting to look more like a game, without any points though. So you'll now add the scoring add the score.

Challenge 13 - Adding the Score

Create a new variable for all sprites, called Score. Leave it displayed on the screen but move it to the very top corner. Increase the score by as much as you like inside the ***if MonsterHit*** Block. Make sure the Score is set back to 0 at the very start of the game i.e. off the 

 if you need some help.

SAVE your work.

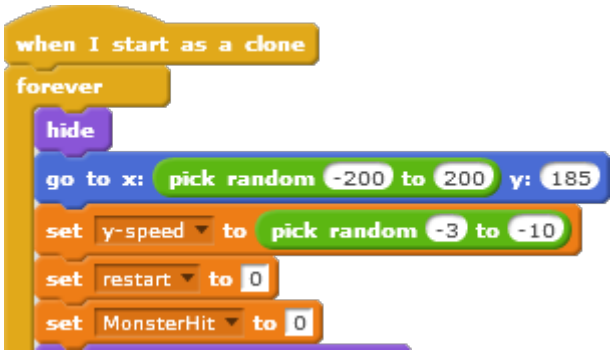
Test your work 

You should now see that every time the Bug-Eye explodes the Score is increased by 10. Even if the Bug-Eye hits the Spaceship.

But it's no fun with just one Bug-Eye, lets add more. We could just duplicate the sprite, but there is a better way that gives you more control over how many Bug-Eyes there are, and it's called cloning.

To clone the Bug-Eye you'll use the following blocks from the **Control** Section.

Detach the Forever loop from the ***when flag clicked*** block and attach it to the ***when I start as clone*** block.



And attach the following to the ***when flag clicked*** block.

SAVE your work.

Test your work 

You now have 10 Bug-Eyes to contend with. But your Spaceship doesn't react to being hit.

3.1.5 Configuring the Spaceship

Select the space ship again.

You'll make the Spaceship react to the Bug-Eye and explode when it's hit as a challenge. You'll also need to create a hit flag for this sprite only, use it to prevent a torpedo being fired and prevent the Spaceship from moving when it's hit. I've outlined what you need to do in this Challenge below: -

Challenge 14 - Spaceship Explodes

Make the following changes: -

- Use a hit flag to prevent the ship moving and firing a torpedo when it's been hit.
- When a ShipHit message is received: -
  - Set the hit flag to 1
  - Play a sound for the ship exploding.
  - Loop through the Spaceship explosion costumes.
  - Once the explosion is complete hide the spaceship
  - Broadcast a Reset message (more about that later).
  - Place the Spaceship back at it's starting position.
  - Wait 3 seconds
  - Clear the Hit Flag, set it to 0.
  - Show the space ship.