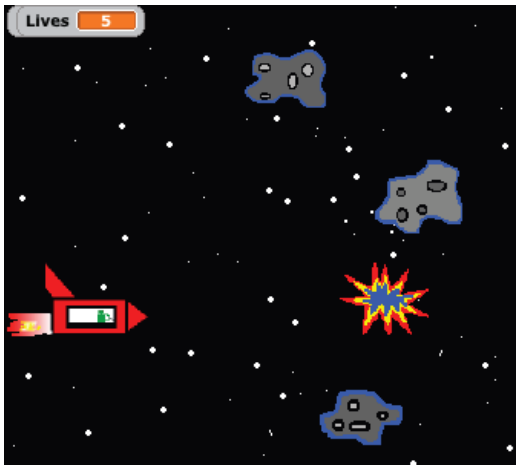
Lesson 27 – Shooter Games: Asteroids

Shooter games are based on an environment whereby a main character secures points for touching another character or set of characters. This genre is often quite challenging and can test a player's speed and reaction time. Oftentimes, the player-character, if he/she survives obstacles, doesn't lose lives and claims a certain score, can then moves forward onto another level or mission.

Asteroids

In this simplified version of the classic game Asteroids, students will be introduced for the first time on how to create a sprite that imitates the rapid action motions of a laser (or bullet).



Game Play - Coding Plan Summary

Spaceship enters an asteroid field. The pilot has to blast the oncoming astral rocks order to save the ship.

The ship can only survive three direct hits before it is destroyed

But for every direct hit, the pilot gains a point.

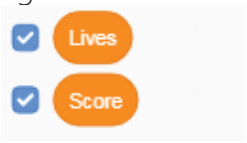
First, select the *Stars* backdrop from the *Nature* folder.

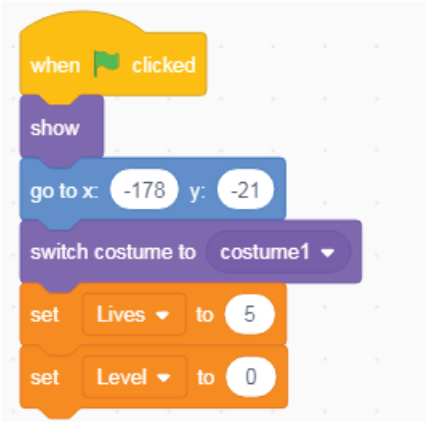
Spaceship sprite

Draw (or select from the sprite gallery) a spaceship with three costumes, one of which signifies an explosion.

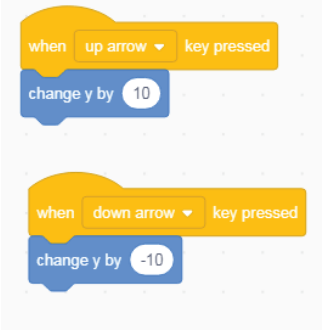
The first two costumes when coded in will give the impression that the spaceship is moving rapidly through space due to the small difference in the size and the shape of the spaceship particularly the width or length of the rocket booster flames.

Set up *Lives* using the *My Variable* option in the *Variables* folder.
Select a maximum of 5 lives.
Create a *Score* monitor using the same procedure using zero (0) as the starting figure.

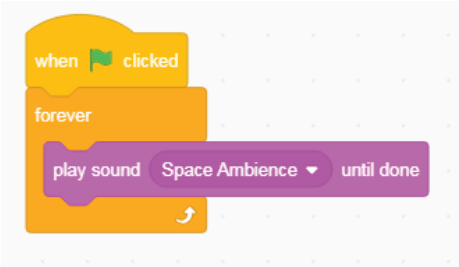




Control the *movement* of the spaceship using the *Up* and *Down Arrow* keys only.



Place in a suitable Sound effect code

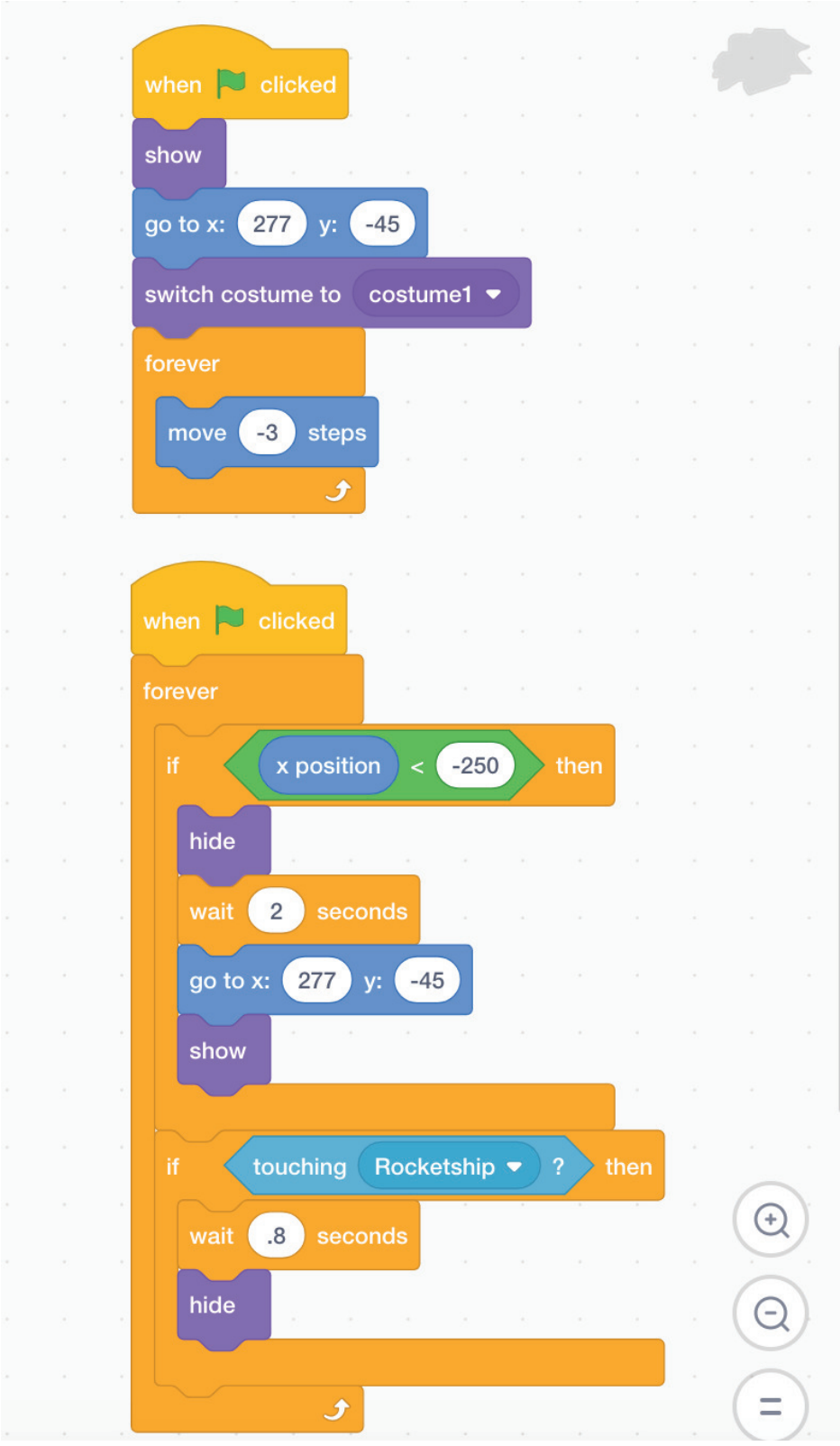


Draw an asteroid sprite with two costumes.

In the first costume, the asteroid should be grey.
Select the following numbers when drawing:
Colour: 25; Saturation: 0; Brightness: 65.

The second costume should represent an explosion.
Now replicate the asteroid at least four more times.
Each asteroid script should have a different X and Y setting to ensure that the sprites do not start from the same location nor have the same flight path.

Furthermore it would be atmospheric if these moving space rocks disappear just prior to reaching the left edge of the screen and to reappear a second or two later moving out of the right side.
This is done by using the following code (with each rock sprite having different X and Y coordinates):

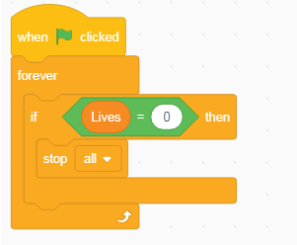


Now we return to the Spaceship script as we have to programme in commands that will signify the impact of a collusion with an asteroid.

As there are multiple asteroids, it would take a lot of code to recognise each of the individual sprites. However we collectively identify all the asteroids by using the identifying colour grey.

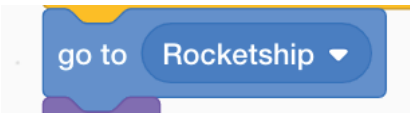
So the main coding script for the spaceship should now read as follows:

Finally, input code that will stop the game once all the spaceship's lives are lost:



The Laser Sprite

Draw a new sprite that consists of a short relatively thick line.
Copy its costume six or seven times.
Give each costume a different colour.
This will give the effect of *firing* when coded in.

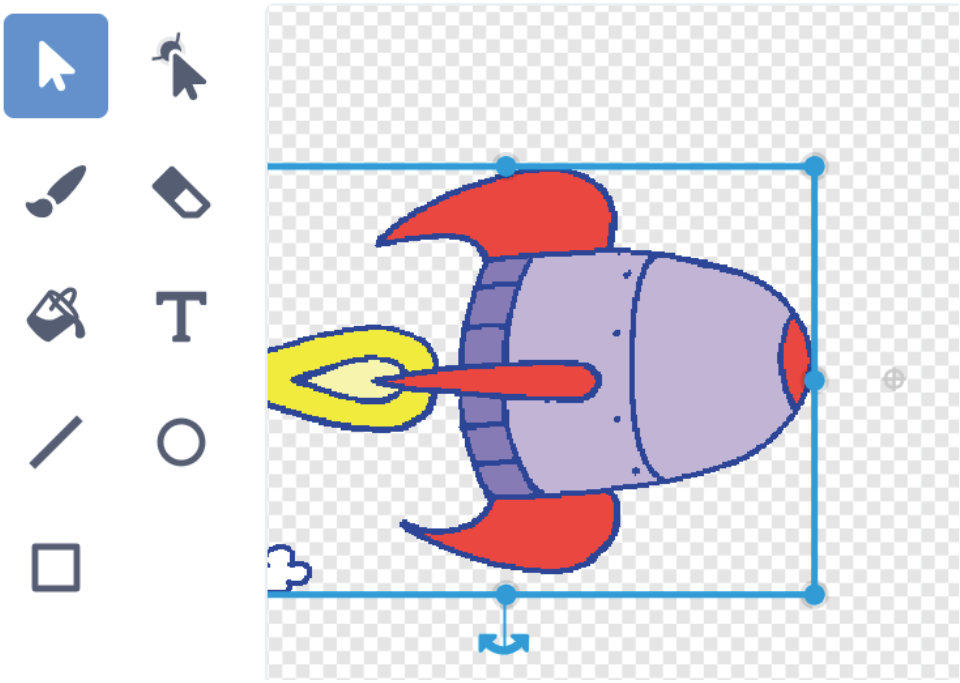
Align the Laser sprite with the Spaceship by using the 

in the Motion category.

However you need to go into the costumes' *Paint Editor* of the Rocketship sprite to get a better alignment between both sprites.

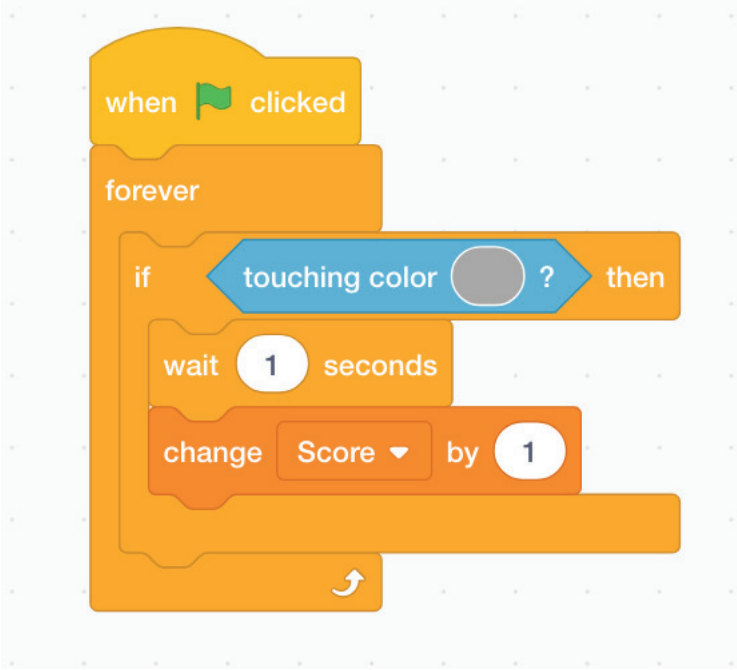
After doing so, move the Rocketship sprite out of the middle of the screen in the *Paint Editor* using the *Arrow* positioning icon located at the Top left hand corner of the tools menu within *Vector* mode.
However if the sprite comprises multiple parts that may move independently of each other using this tool, go to the tool  located at the top right of the menu which allow it to be manipulated as one entity.

You will see a very small ***target*** marking. Move the spaceship sprite to the left and in front of the target.



Run the full script project and see if it works.
You may though have to experiment a few times with moving the rocketship sprite in the Paint Editor until you feel the alignment between both sprites is satisfactory.
Now, we build a code that will give the visual impression that, when the spacebar is touched, a laser is shot from the spaceship in a straight firing line towards the direction of the asteroid.

The code below includes commands that registers the laser hits of an asteroid:



However please ensure that, in the

block, you input the same numbers in the colouring box as you did when colouring the asteroid. Otherwise no score will be registered when you touch an asteroid. These numbers are:
Colour: 25; Saturation: 0; Brightness: 65.