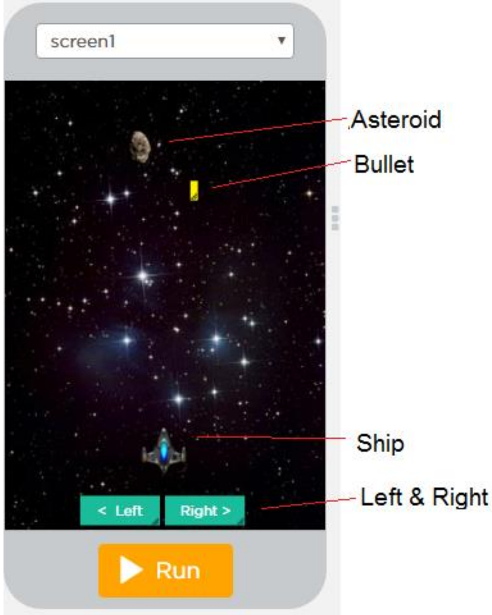
C7 Space Shooter

Screen 1's background is the space picture.

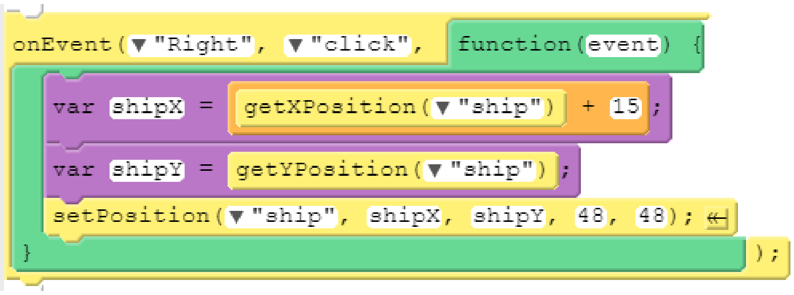
Asteroid, Bullet and Ship are all images.

Left and Right are buttons (to control the ship).



Ship

- The ship moves back and forth based on the buttons.
- You can actually avoid tracking the ship's position by using its accessors – getXposition and getYposition.

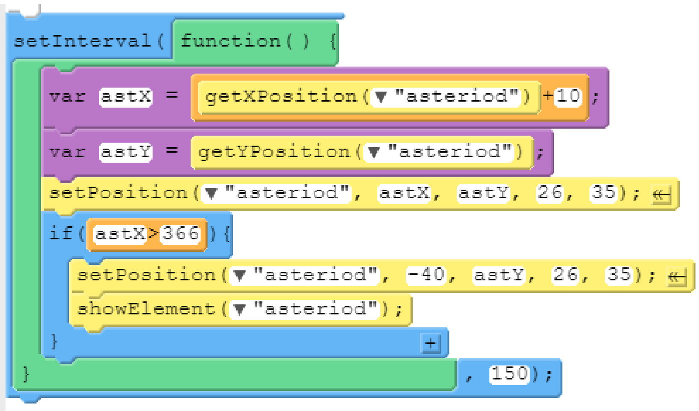


```
onEvent("Right", "click", function(event) {
  var shipX = getXPosition("ship") + 15;
  var shipY = getYPosition("ship");
  setPosition("ship", shipX, shipY, 48, 48);
});
```

- If you are really brave, you can do all three lines in one.
- Make the left button too. Verify that they work.

Asteroid

The asteroid starts on the left, travels to the right and then restarts again on the left.

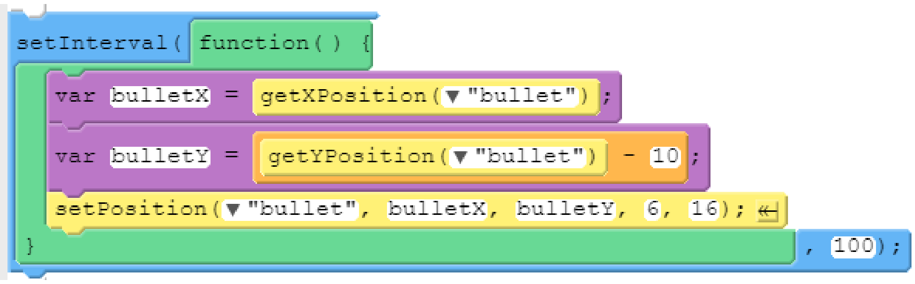


```
setInterval(function() {
  var astX = getXPosition("asteriod")+10;
  var astY = getYPosition("asteriod");
  setPosition("asteriod", astX, astY, 26, 35);
  if (astX>366) {
    setPosition("asteriod", -40, astY, 26, 35);
    showElement("asteriod");
  }
}, 150);
```

Put in the showElement line to deal with the fact it may have disappeared when it was hit by the bullet.

Bullet

Start the bullet by making it move up.

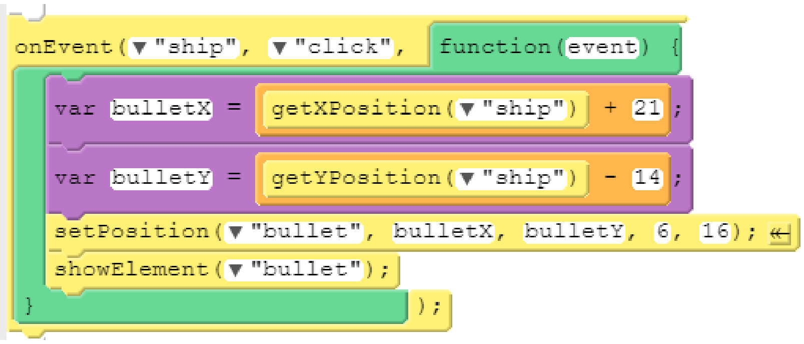


```
setInterval(function() {
  var bulletX = getXPosition("bullet");
  var bulletY = getYPosition("bullet") - 10;
  setPosition("bullet", bulletX, bulletY, 6, 16);
}, 100);
```

Verify that is working.

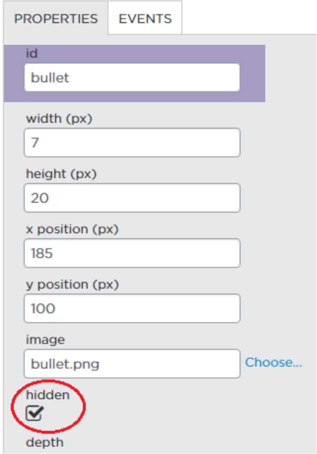
Now we need to be able to fire the bullet:

When the ship is clicked, we will move the bullet back to the ship's position and send it on its way.



```
onEvent("ship", "click", function(event) {
  var bulletX = getXPosition("ship") + 21;
  var bulletY = getYPosition("ship") - 14;
  setPosition("bullet", bulletX, bulletY, 6, 16);
  showElement("bullet");
});
```

Go back to design mode and set the default position of the bullet to be hidden.



id
bullet

width (px)
7

height (px)
20

x position (px)
185

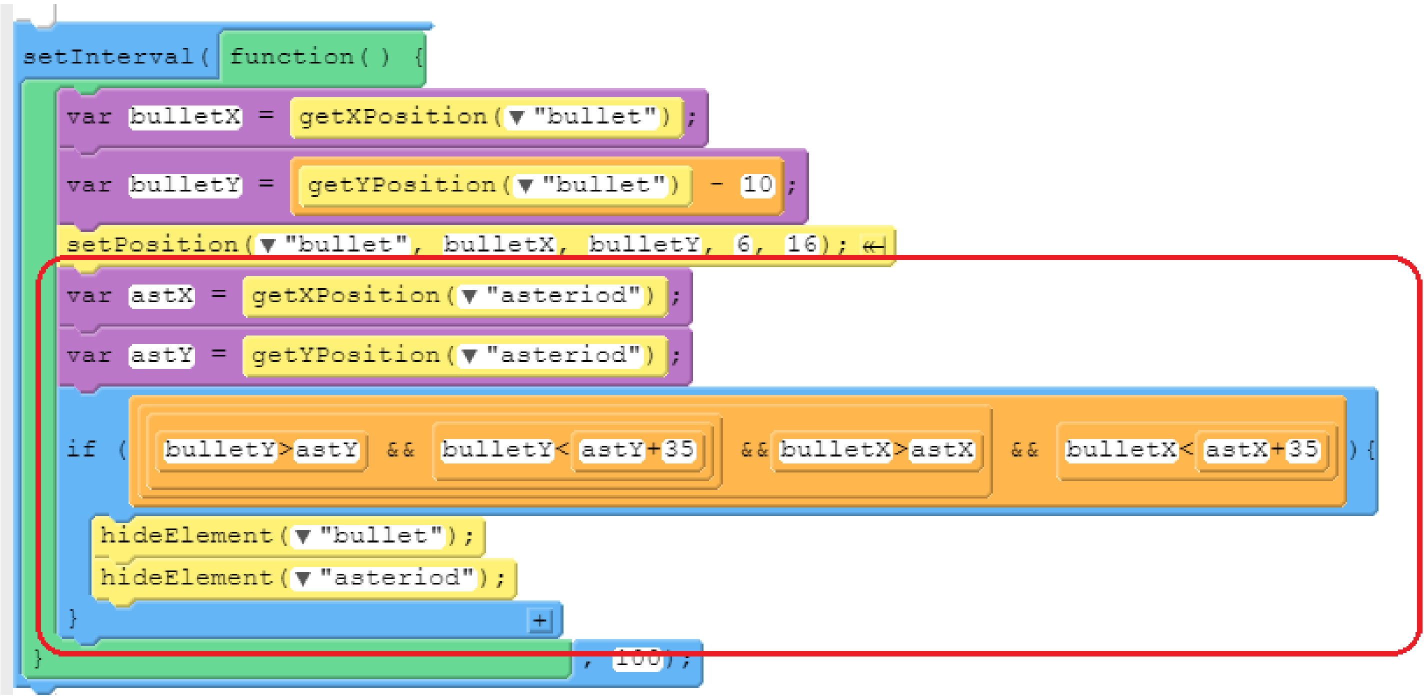
y position (px)
100

image
bullet.png

hidden
☑

depth

Then, handle the collision with the asteroid.



```
setInterval(function() {
  var bulletX = getXPosition("bullet");
  var bulletY = getYPosition("bullet") - 10;
  setPosition("bullet", bulletX, bulletY, 6, 16);
  var astX = getXPosition("asteriod");
  var astY = getYPosition("asteriod");
  if (bulletY>astY && bulletY<astY+35 && bulletX>astX && bulletX<astX+35) {
    hideElement("bullet");
    hideElement("asteriod");
  }
}, 100);
```

I typed the Boolean expression to detect the collision:

```
    if (bulletY>astY && bulletY<astY+35 &&bulletX>astX && bulletX<astX+35)
```

Feel free to cut and paste.

Essentially, you are checking if the bullet is inside the space occupied by the asteroid.

Verify that everything works.

Bonus: (Try some of the following to increase the difficulty)

- Add a score to the screen.
    - If an asteroid is hit, it goes up. If it is missed, it goes up.
- Add a second asteroid to the screen.
- Give the user a limited number of bullets.