Asteroids

# Introduction



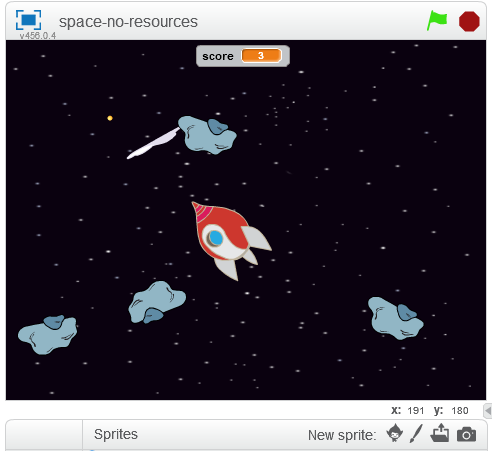
The Original

This project is based on the 1979 Atari arcade space shooter game ***Asteroids***.

In the original game, the player controls a spaceship in an asteroid field which is periodically traversed by flying saucers.

The object of the game is to shoot and destroy asteroids and saucers while not colliding with either or being hit by the saucers' counter-fire.

Our version includes only the spaceship and the asteroids, but has much better graphics than the original – and can be made using resources already in Scratch!



Our Project

The player controls a spaceship which moves around deep space dodging the asteroids coming from all directions. Each asteroid destroyed earns a point.

The asteroids move in random directions – then disappear. The game ends when an asteroid collides with the spaceship.

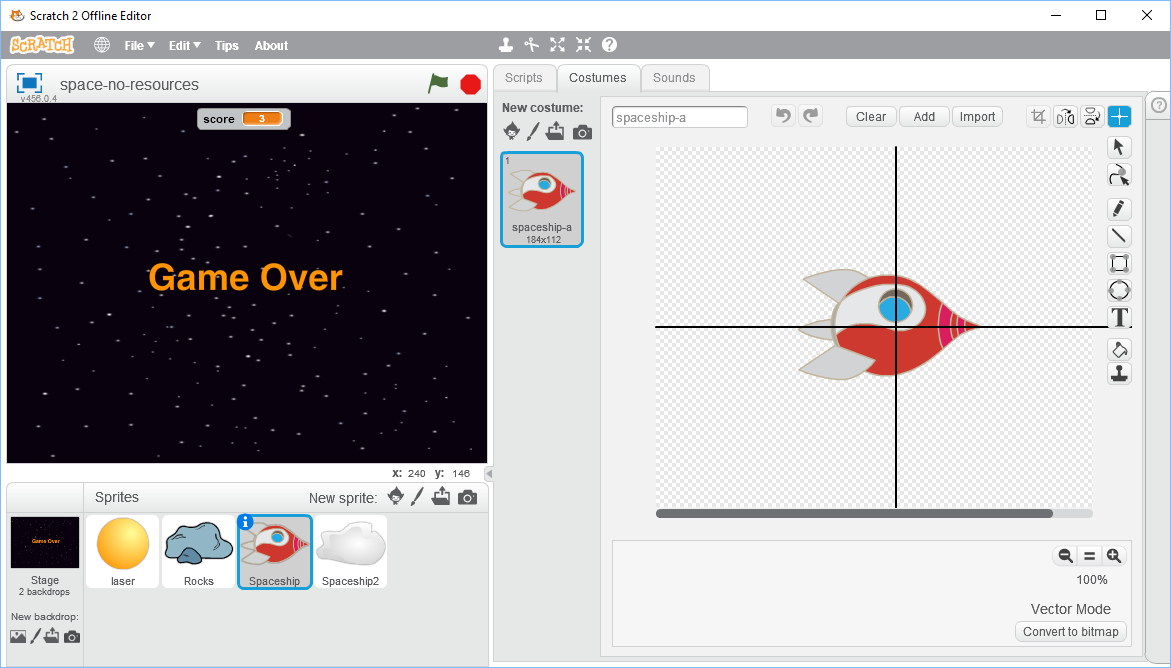
The following **parameters** will be used to change how the game behaves:

* How quickly the spacecraft speeds up when the rocket engine is fired (*thrust*)
* How quickly it slows down when the rocket engine is off (*friction*)
* How far the laser gun will reach (*max\_laser*)
* How many asteroids can be visible at the same time (*max\_rocks*)
* How close to the spaceship new asteroids can appear (*min\_distance*)

# Step 1 - Make the Initial Spaceship

Start Scratch. Set the background to *stars* and delete the cat sprite.

Add the *Spaceship* sprite from the library. Rotate the *spaceship-a* costume to be pointing to the right (90 degrees) and set the centre of the spaceship to line up with the nose and the middle of the window.



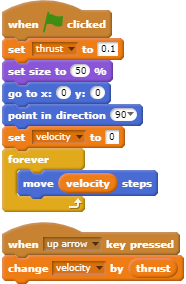
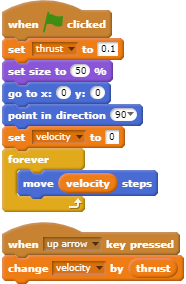
Duplicate the *Spaceship* sprite to make *Spaceship2*, which you will need later.

Go back to the script of *Spaceship* before continuing.

# Make the Spaceship Move

Create two variables which are for this sprite only, *thrust* and *velocity*. The variable *thrust* will control how quickly the spaceship will accelerate and *velocity* will hold the speed at which it is currently moving.

Write these two short sequences of code.

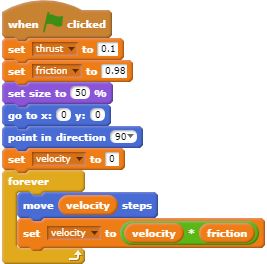


Start the game running by pressing the green flag. The spaceship will appear in the middle of the stage. Press the *up arrow* key and it will move to the right. The longer you hold it down, the faster it will go. It will stop when it hits the right-hand edge because Scratch cannot draw off the stage. Press the red button to stop the game.

# Add friction

We have no way of slowing down or stopping the spaceship. We could use the *down arrow* key to slow it down, but a more interesting idea would be to pretend that a spaceship travelling through space will slow down by itself. If you know your physics, you will know that this is not the case in real life but it makes the game more interesting.

Create a new variable *friction* which should be just for this sprite, then change the code slightly as shown.



If you run the game now, pressing the *up arrow* key will start the spaceship moving and when you release it, the spaceship will gently slow down.

# Add initial steering

The spaceship moves but only in one direction, so we will now add some steering using the left and right arrow keys. Add these two sequences to the scripts of *Spaceship*.



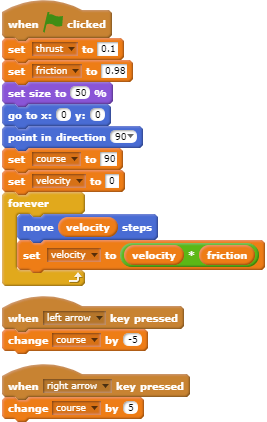
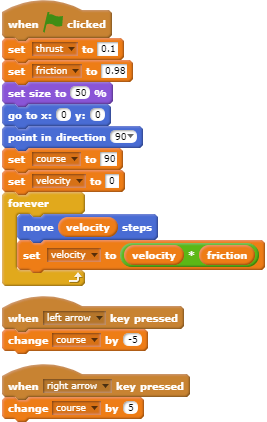
If you run the game now, you can steer the ship. But… the trouble is that this steers the spaceship like a car, not like a spaceship! Changing the direction the spaceship is pointing in should not make it change direction unless the rocket engine is being fired (up arrow) at the same time. But Scratch will always move a sprite in the direction it is pointing.

One way around this would be to have a different costume for each angle at which the spaceship can be. There are 360 possible and since we move in 5 degree steps, that means we would only need 72 costumes! Ouch. The second problem is that we need to be able to choose the right one for the direction the spaceship is travelling when the rocket motor is fired.

However, a better solution is to have two spaceship sprites, where the second one does nothing except provide the costume which you can see. Only the first one moves around the screen, but is hidden so you can’t see it. The second spaceship is visible and will rotate to whatever angle you want and will track the first spaceship’s position and direction. Confused?

Create a new variable *course* which needs to be available for all sprites because both spaceships will need to use it.

Modify the code of *Spaceship* slightly to make it possible for the second spaceship to know where it is and which way its costume should point. Note that you also need to change the two steering sections (left and right arrow).

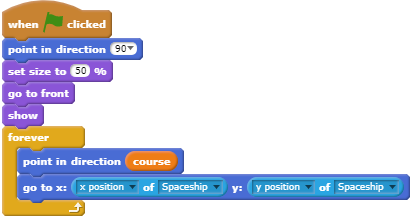


Run the game and fly the spaceship around. When you press the left or right arrow, nothing seems to happen - until you press the up arrow to fire the rocket engine then the spaceship goes in the new direction. The spaceship is travelling in the right way, but the costume isn’t always facing in the right direction.

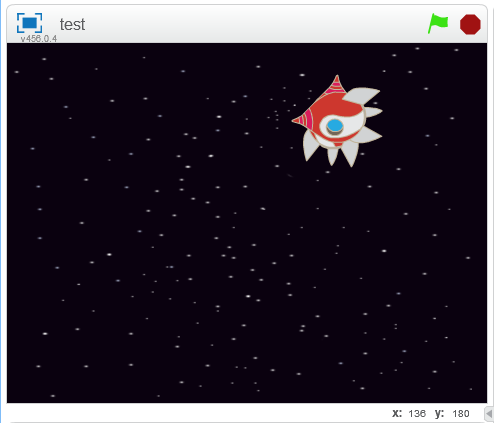
# Add Code to Spaceship2

The second spaceship, which you created at the start of this project, will fix this missing part of the jigsaw.

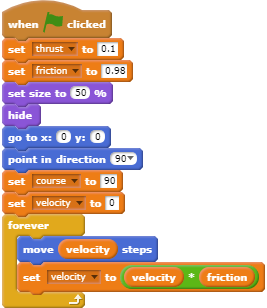
Add the code needed in the scripts for *Spaceship2* to show the costume correctly. The first part points the spaceship in the same direction and size as *Spaceship*. The forever loop points *Spaceship2* in the same direction as *Spaceship* using the variable *course*. Then it moves to the same position that *Spaceship* is currently on. All it is doing is pointing in the direction we want and jumping to where *Spaceship* is (too quickly for you to see).



Start the game and drive the spaceship around again. Now you will see that there are two spaceships visible, one on top of the other. When you turn using the left and right arrow keys, you can see one of these changes direction but the spaceship carries on in the same direction as before until you fire the rocket engine using the up arrow.



Change the script of *Spaceship* slightly so that we can’t see it anymore by adding a Hide block.



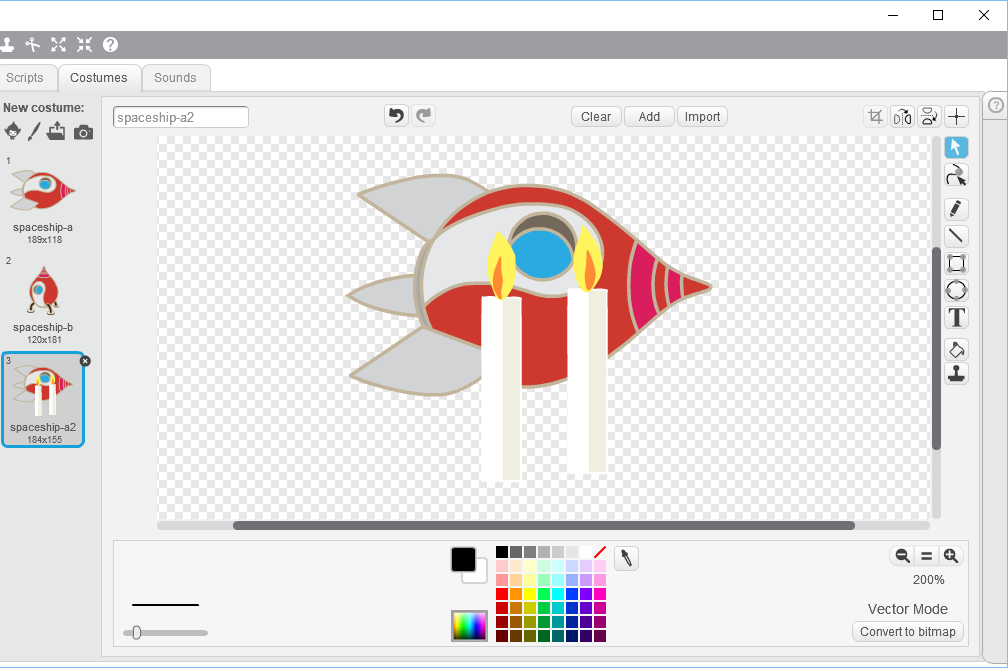
Start the game again and drive the spaceship around. Now when the spaceship is moving and you are not using the rocket engine, if you rotate the spaceship it does not change direction - until you use the rocket engine using the *up arrow*.

# Show When the Rocket Engine is being Used

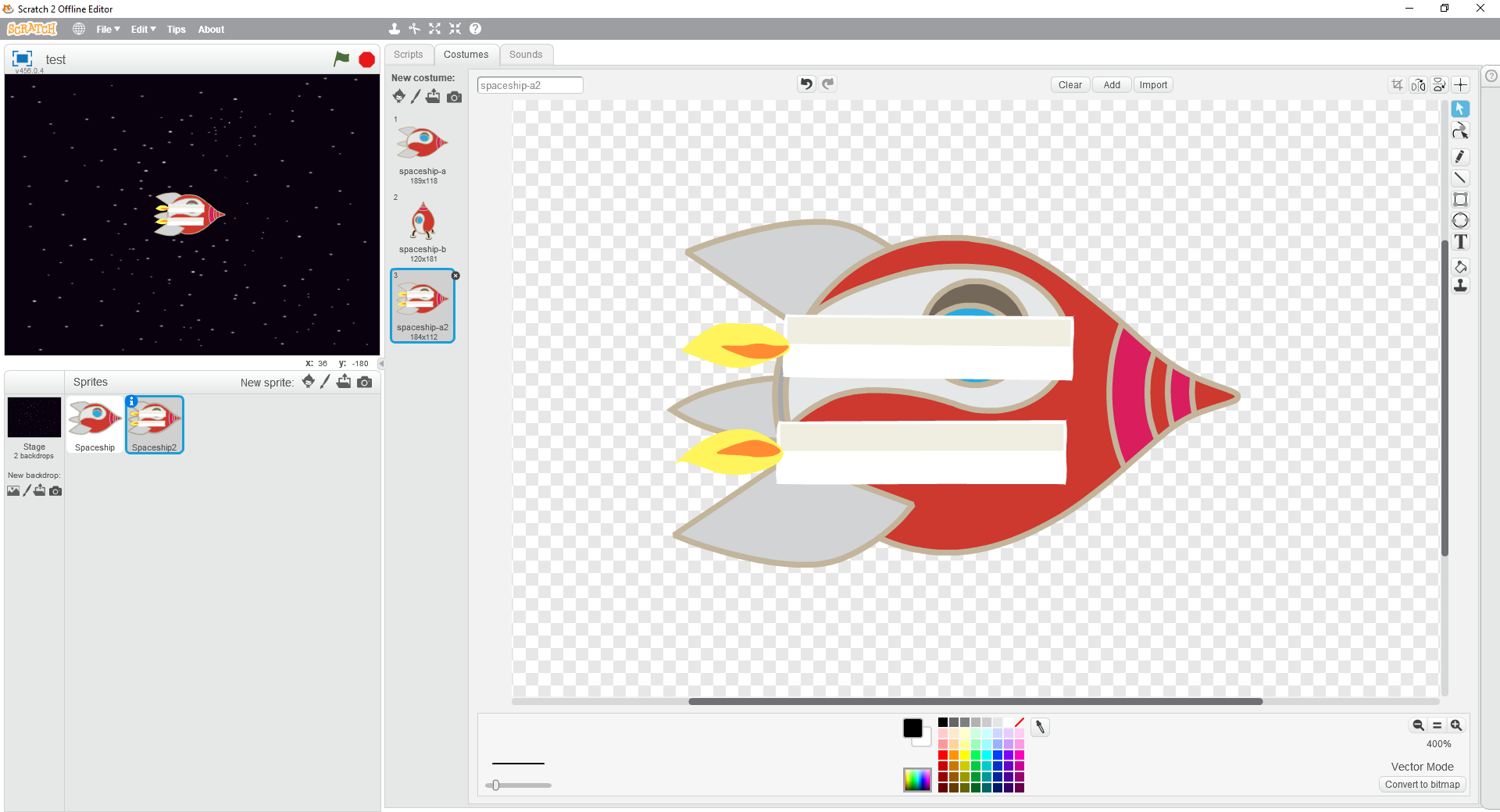
Something which we can do easily now, which would be very difficult with 72 costumes, would be to add a flame for the rocket engine when it is being fired. In fact, we will add two so that it will flicker slightly when we are pressing the *up arrow* key.

Make sure you are looking at *Spaceship2* and make a copy of its costume *spaceship-a*. This will be called *spaceship-a2*.

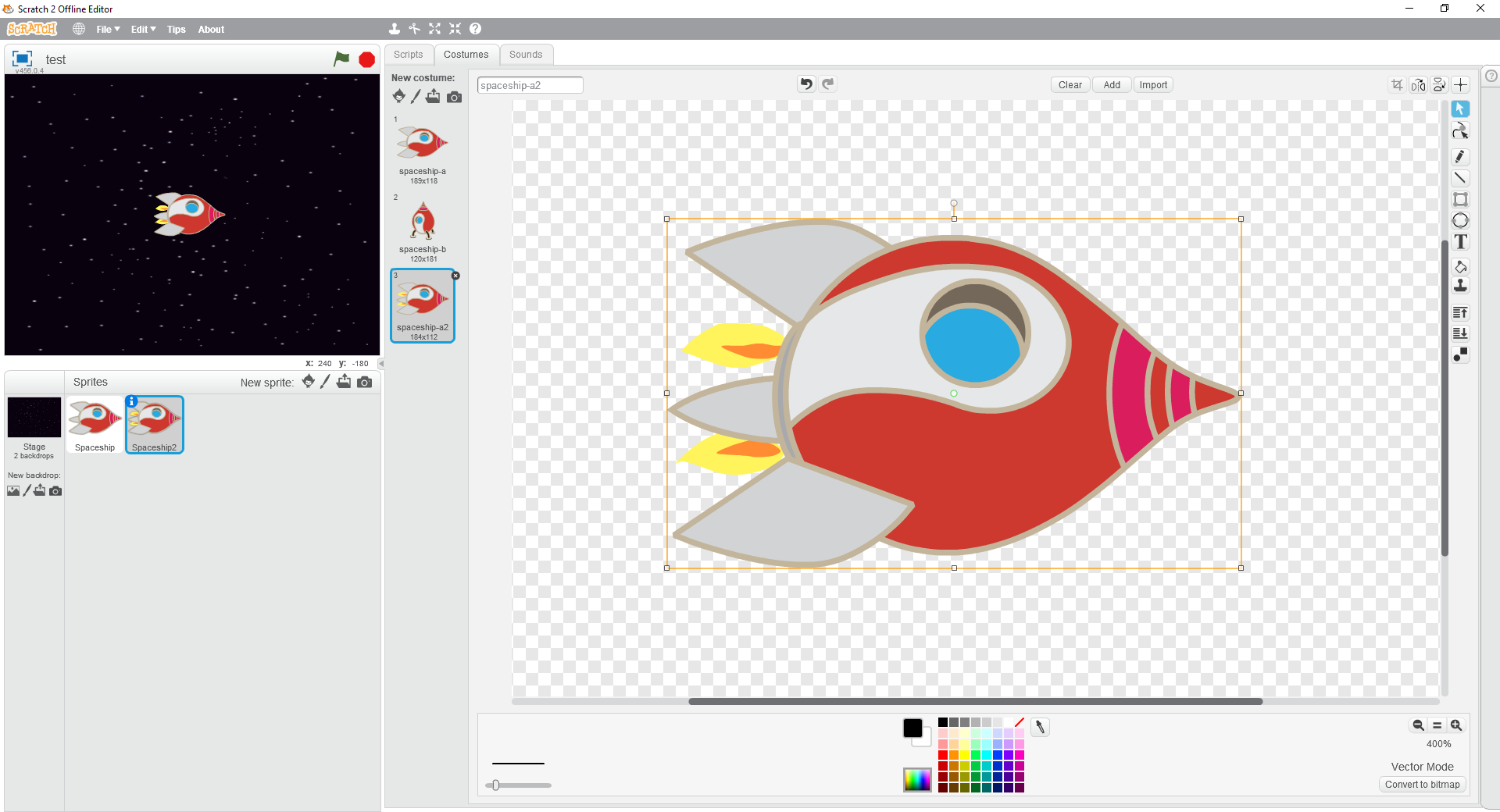
Click “*Add*” at the top of the screen and select *candle1-a* and *candle1-b* from the library so they are added to the costume (you can do this in one step if you know how, or in two steps).



The image is in vector mode which means that we can easily rotate the two candles and position them roughly so that the flames are at the back of the spaceship.



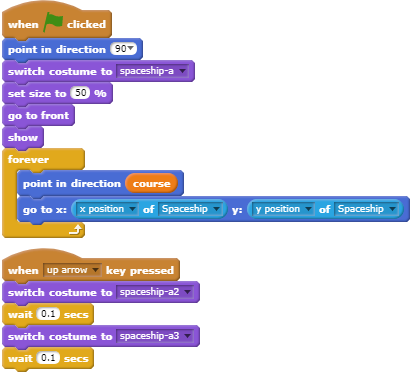
This doesn’t look very good, but remember that in vector mode each object is on a different layer – one on top of the other. At the moment, the candles are higher up than the spaceship so we need to move the spaceship to the top. Click the spaceship to select it and press the “*Forward a layer*” button (circled in red below) on the right-hand side twice - and the two candles will now be below the spaceship with only the flames showing.



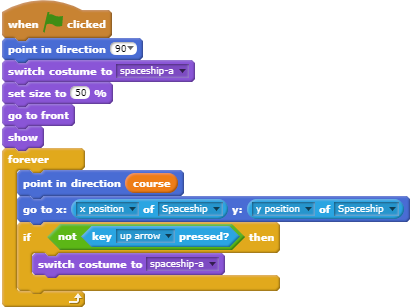
You can move the two candles so that the flames look how you like them, the candlestick parts remain hidden underneath the spaceship.

Duplicate this costume to create *spaceship-a3*. Swap the two candles around to make the flames slightly different between the two costumes.

Now we can add the code to make this work by setting the starting costume, then to use the two new costumes when the *up arrow* key is pressed.



Run the game and press the *up arrow* key to fire the rocket engine. Hey that looks good – but… when you release the up arrow, the rocket engine still looks like it is firing. There is no easy way to detect when you release the *up arrow* key, but it is easy to check if it is **not** being pressed in the forever loop as shown below.



Run the game again and move around the screen. This should now look pretty good!