

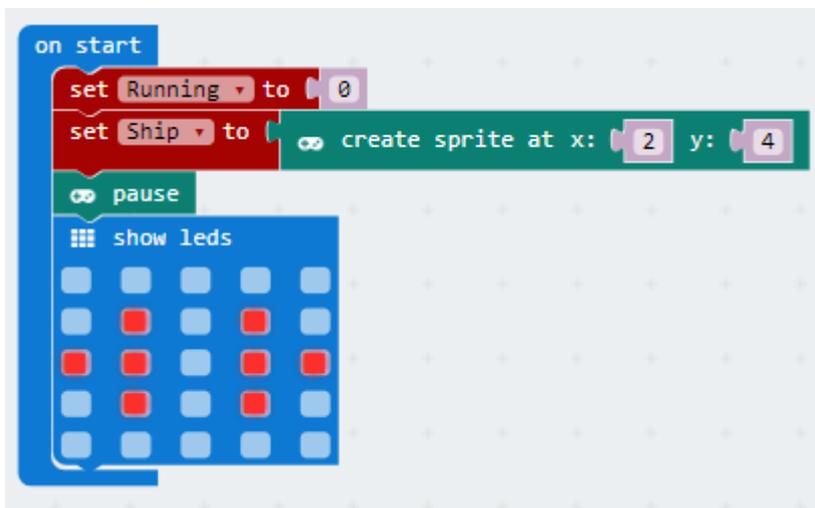
## Space Blaster Single Player

The following instructions will take you through the steps of creating a game where you pilot a spaceship and try to blast away at the aliens invading your home worlds territory. This game makes use of the micro:bit game blocks and also contains extension points where you can develop the game further. The following link is the reference page for the micro:bit game blocks.

<https://makecode.microbit.org/reference/game>

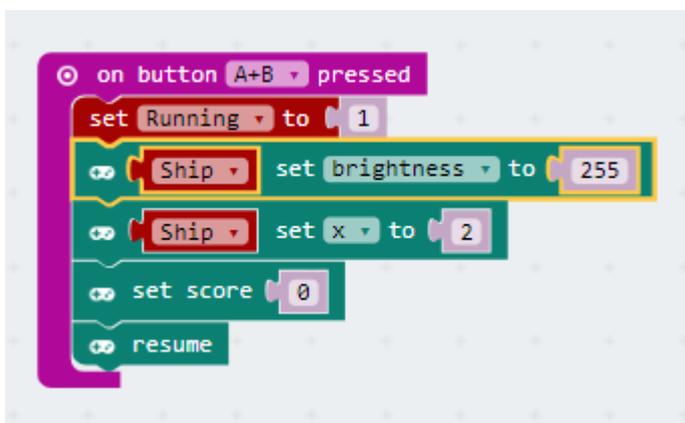
### Step 1 – Create the spaceship

The spaceship will be created as a sprite that can be moved along the bottom line of the micro:bit LED matrix when the game is running. A variable running is used to control if the game is currently running or not and a variable ship is used for the spaceship sprite. When the game first starts, we do not want it to start running.



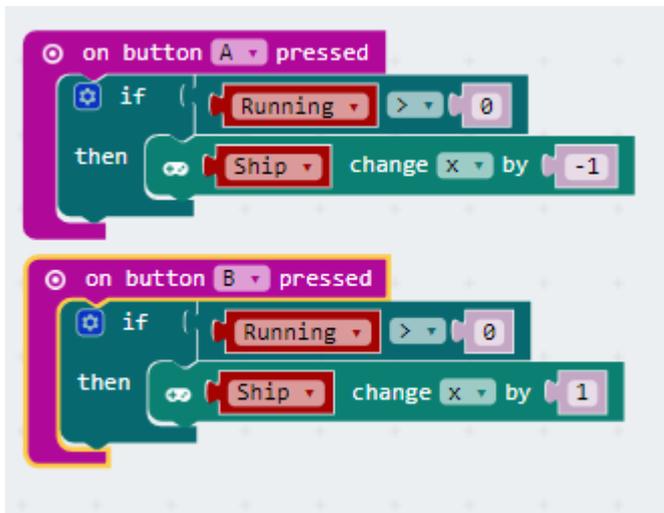
### Step 2 – Start the game

The A+B buttons used together will be used to start the game, by setting running to 1 and resetting the game variables.



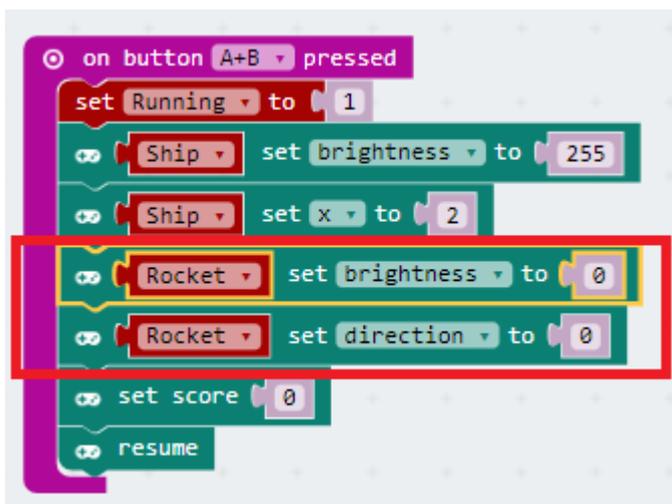
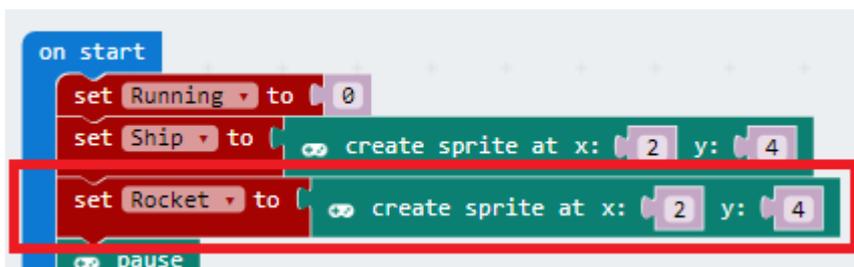
### Step 3 – Move the spaceship

The A button will be used to move the spaceship left and the B button will be used to move the spaceship right.



### Step 4 – Add the rocket

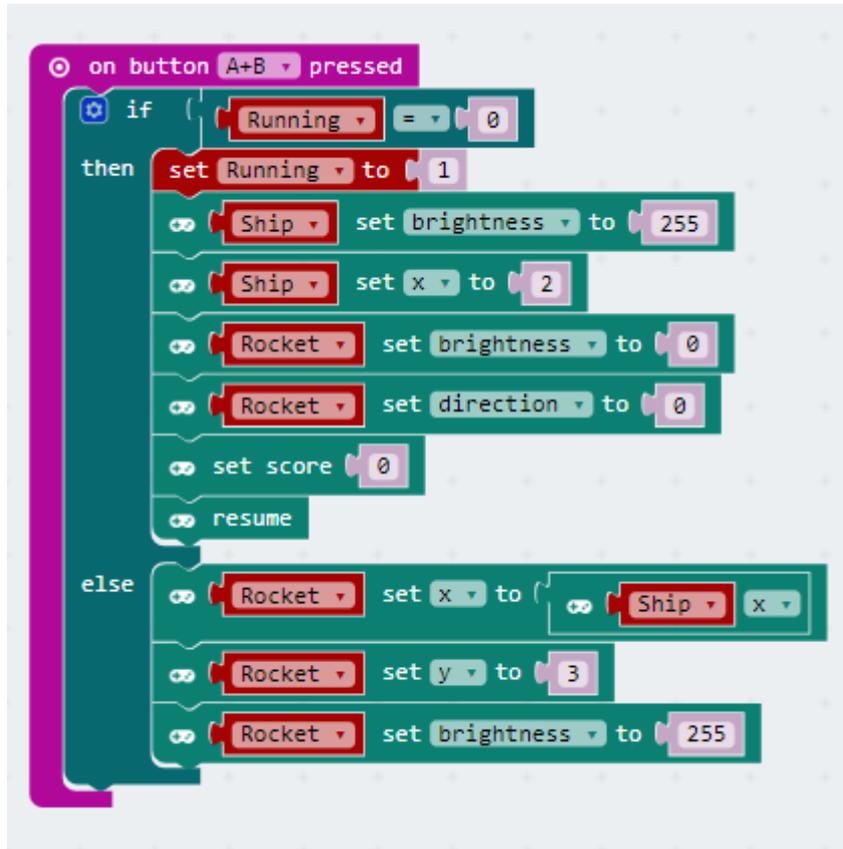
Create a new sprite called Rocket under the existing during the on start section and also set the default values for the rocket in the on button A+B event.



### Step 5 – Firing the Rocket

We are going to allow the rocket to be fired when both the A+B buttons are pressed together at the same time. Because we are also using A+B to start the game, we will need to add an if-then-else condition to the A+B event to only fire the rocket when the game is running. Modify the A+B button code to be as follows and test it out.

What happens when you fire the rocket?



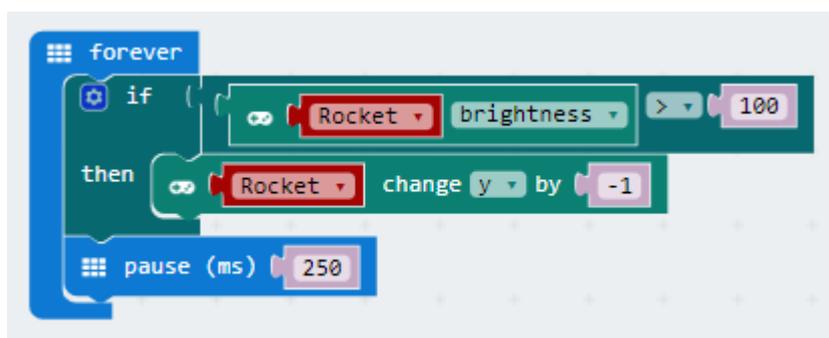
```
on button A+B pressed
  if (Running = 0)
  then
    set Running to 1
    Ship set brightness to 255
    Ship set x to 2
    Rocket set brightness to 0
    Rocket set direction to 0
    set score 0
    resume
  else
    Rocket set x to Ship x
    Rocket set y to 3
    Rocket set brightness to 255
```

### Step 6 – Moving the rocket

Did you find that the rocket that was being fired stayed just above the spaceship? That's because we haven't added any code to move the rocket yet. We can move the rocket by adding in some special code in the forever loop.

What happens when the rocket gets to the top of the LED matrix?

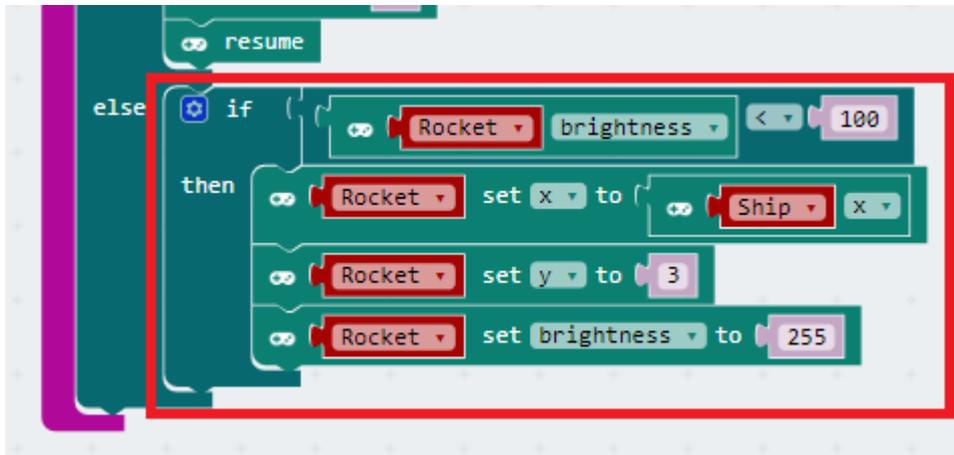
What happens if you press A+B again, before the rocket gets to the top of the LED matrix?



```
forever
  if (Rocket brightness > 100)
  then
    Rocket change y by -1
  pause (ms) 250
```

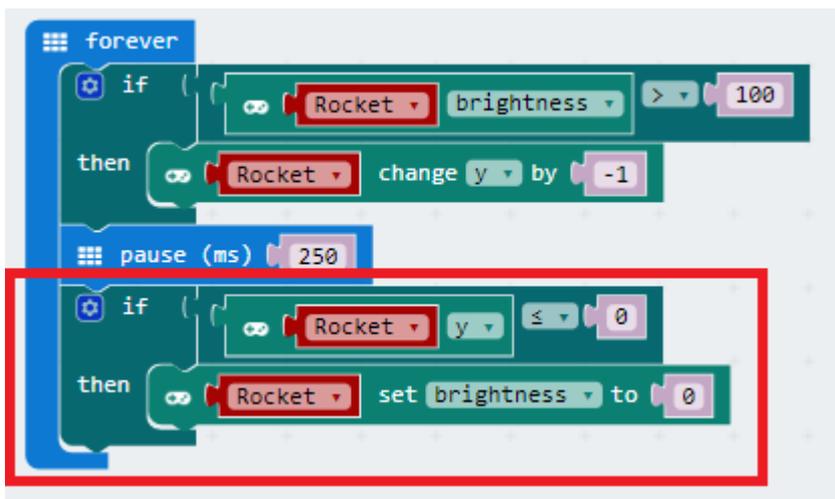
### Step 7 – Only one rocket and disappearing at the top

We only want the player to be able to fire a rocket if they have not already fired one and it is in motion. We can test the brightness of the rocket and only reset it if the brightness is less than 100. Change the code in else statement of the A+B event as follows.



```
else
  if (Rocket brightness < 100)
  then
    Rocket set x to Ship x
    Rocket set y to 3
    Rocket set brightness to 255
```

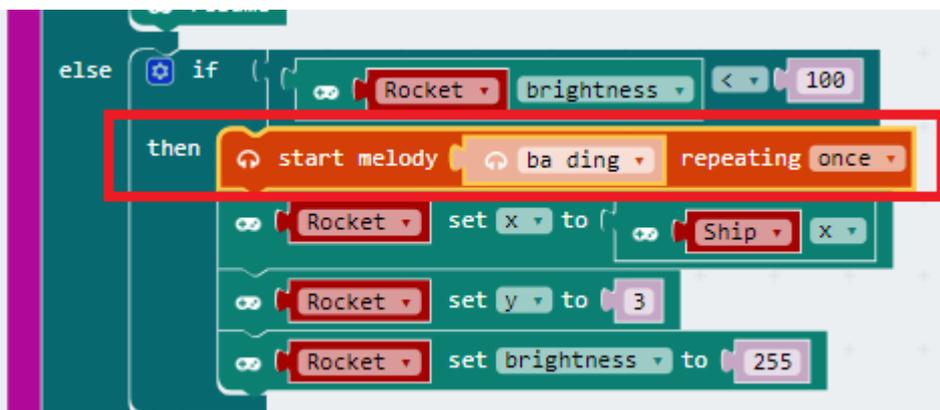
We also want the rocket to disappear when it reaches the top of the LED Matrix. Therefore, add the following code to the bottom of the forever loop.



```
forever
  if (Rocket brightness > 100)
  then
    Rocket change y by -1
  pause (ms) 250
  if (Rocket y <= 0)
  then
    Rocket set brightness to 0
```

### Step 8 – Add sound

It's much more fun to add sounds to indicate that the rocket has fired. Add in a sound effect of your choice to the code where the rocket gets fired.



```
else
  if (Rocket brightness < 100)
  then
    start melody ba ding repeating once
    Rocket set x to Ship x
    Rocket set y to 3
    Rocket set brightness to 255
```

### Step 9 – An alien to shoot

We create the alien that shoots at us in the same way as we have created the spaceship sprite. The primary difference is that the alien will be at the top of the screen rather than at the bottom which is where the spaceship is.

```
on start
  set Running to 0
  set Ship to create sprite at x: 2 y: 4
  set Rocket to create sprite at x: 2 y: 4
  set Alien to create sprite at x: 2 y: 0
```

```
on button A+B pressed
  if Running = 0
  then
    set Running to 1
    Alien set brightness to 255
    Alien set x to 2
    Ship set brightness to 255
    Ship set x to 2
```

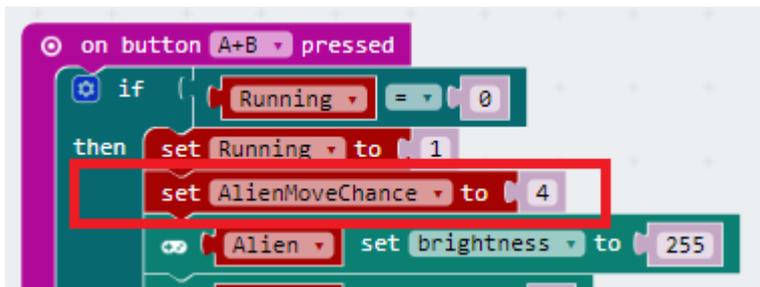
### Step 10 – Hitting the alien

We need to add in the code to detect if the rocket hits the alien and if it does, we need to increase the players score. To do this, we change the code that hides the rocket when it reaches the top of the LED matrix. This looks complicated code but all it is doing is checking if the rocket is in the same place as the alien and the rocket has not yet been hidden. This is also a good place to add another sound effect.

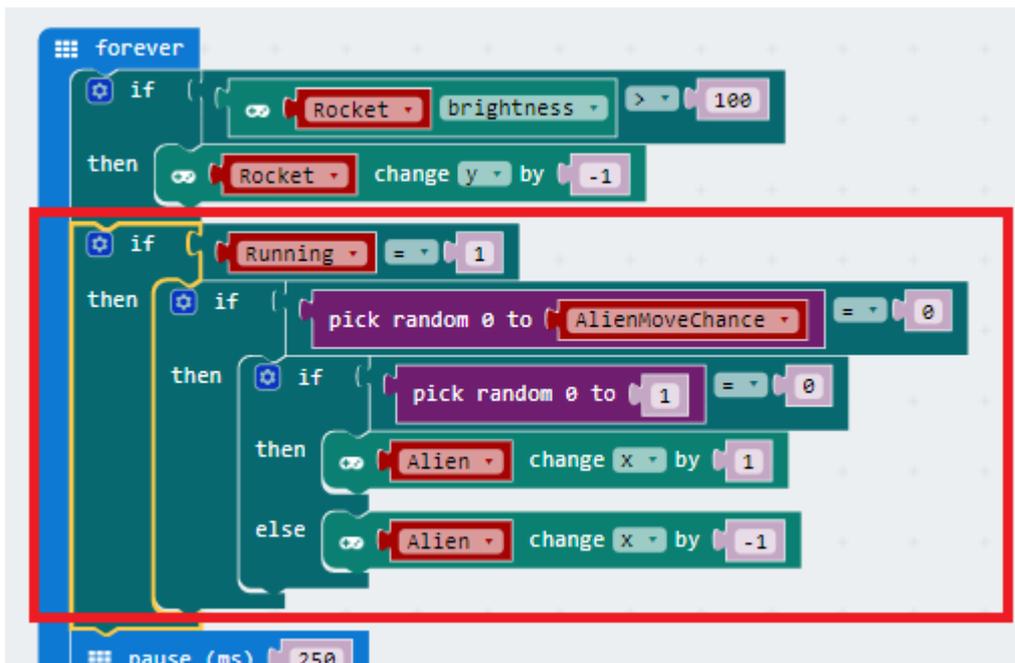
```
pause (ms) 250
if Rocket y <= 0
then
  if Rocket brightness >= 100
  then
    Rocket set brightness to 0
    if Rocket x = Alien x
    then
      start melody power up repeating once
      change score by 1
```

## Step 11 – Making the alien move

The alien isn't hard to hit so we need to make it move around. The speed of the alien moving will control the difficulty in shooting it. To determine if the alien moves, we will use a random number (this can be found in the Math blocks). We first create a variable called AlienMoveChance and set it to a value (4 in the example below) in the A+B event. The smaller the number that AlienMoveChance is set to, the greater the chance the alien will move each time through the forever loop. The reason we use the variable AlienMoveChance is so that we can modify the value when the game is playing (reducing the value to make the alien move faster).

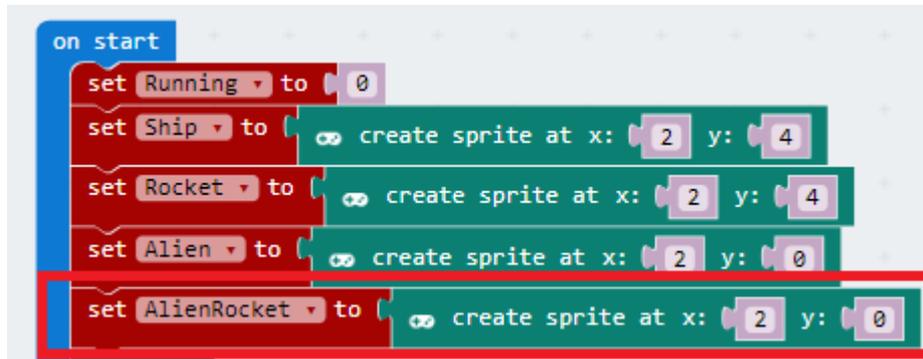


Now, in the forever loop we add in the code that determines if the alien is moving if the game is currently running. If the alien does move, we then make another random choice to see if the alien should move left or right.



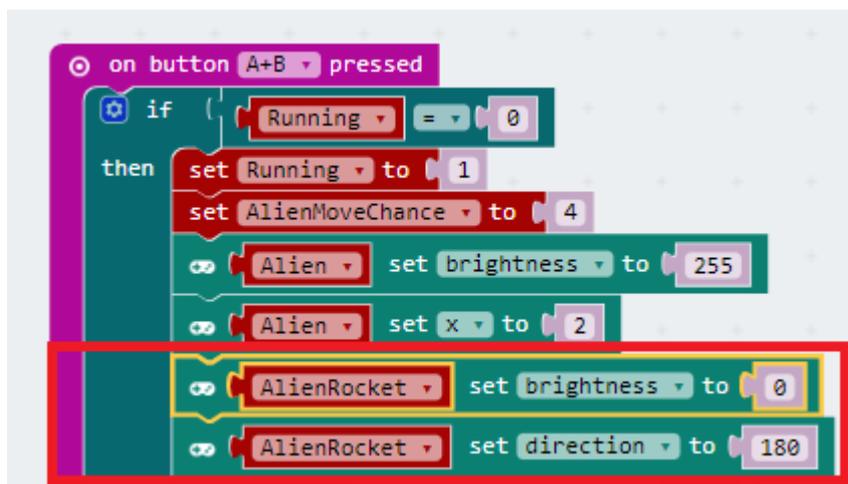
## Step 12 – Getting the alien to shoot back

So far, the rockets are only coming one way so there isn't much challenge to this game. It's time to let the alien shoot back. This section may seem like a lot of code but it is just replicating the code that you've already put together for the spaceship rocket. Start by creating a new variable called AlienRocket for the sprite.



```
on start
  set Running to 0
  set Ship to create sprite at x: 2 y: 4
  set Rocket to create sprite at x: 2 y: 4
  set Alien to create sprite at x: 2 y: 0
  set AlienRocket to create sprite at x: 2 y: 0
```

Then in the A+B button event, set the default values for the alien rocket at the start of the game.



```
on button A+B pressed
  if Running = 0
  then
    set Running to 1
    set AlienMoveChance to 4
    Alien set brightness to 255
    Alien set x to 2
    AlienRocket set brightness to 0
    AlienRocket set direction to 180
```

The rest of the code goes into the forever loop. The first section goes at the top (before the pause) and the second piece of code goes at the end of the forever loop (after the pause).

The first piece of code has two parts. An if statement that moves the alien's rocket downwards by 1 LED pixel if the rocket is showing and a piece of code that uses the same random chance method as for the alien move to decide if the alien is going to shoot or not. The alien only shoots though if its existing rocket is not already in motion.

The second piece of code checks to see if the alien rocket has reached the bottom of the LED matrix and whether it has hit the ship or not. If it has hit the ship, the game is over.

This section of code goes at the beginning of the forever loop.

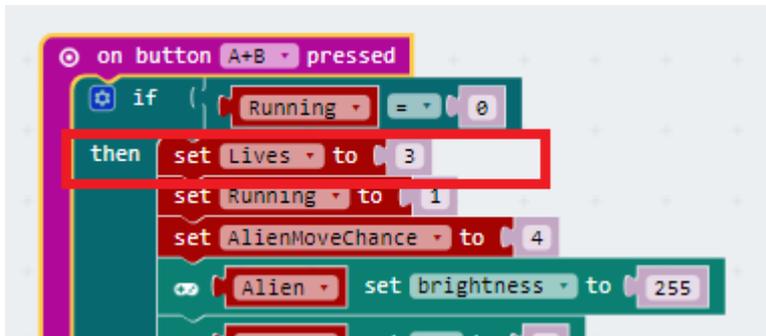
```
forever loop:  
  if (Rocket brightness > 100)  
    then (Rocket change y by -1)  
  if (AlienRocket brightness > 100)  
    then (AlienRocket change y by 1)  
  if (Running = 1)  
    then if (AlienRocket brightness ≤ 100)  
      then if (pick random 0 to AlienMoveChance = 0)  
          then (AlienRocket set x to Alien x)  
              (AlienRocket set y to 1)  
              (AlienRocket set brightness to 255)  
              (start melody ba ding repeating once)  
          if (pick random 0 to AlienMoveChance = 0)
```

This section goes at the end of the forever loop.

```
  then (start melody power up repeating once)  
  (change score by 1)  
  if (AlienRocket y ≥ 4)  
    then if (AlienRocket brightness ≥ 100)  
      then (AlienRocket set brightness to 0)  
          if (AlienRocket x = Ship x)  
              then (start melody power down repeating once)  
                  (game over)
```

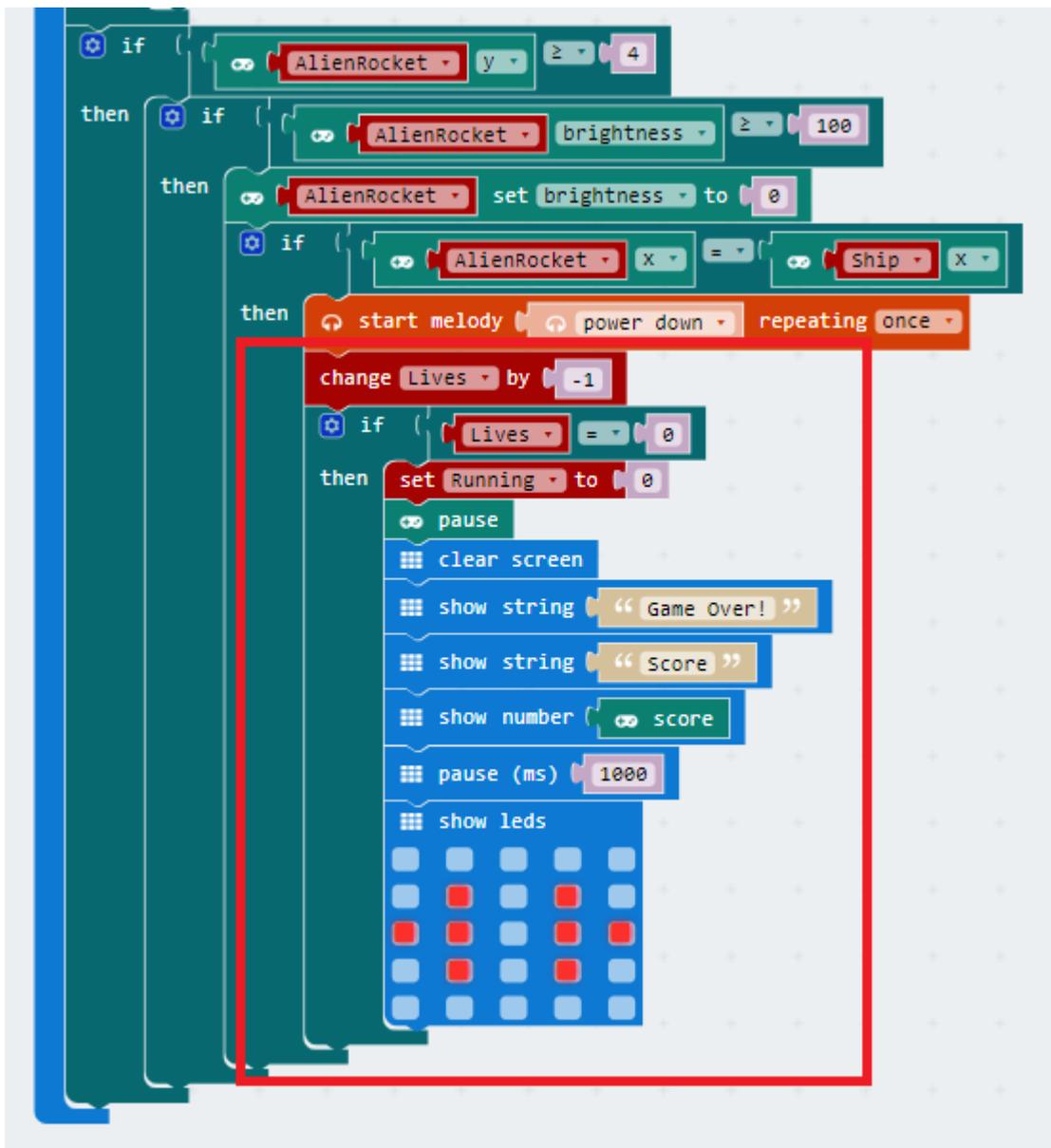
### Step 13 – Getting hit and ending the game

At present, getting hit by the alien causes the game to end and to restart and play again the micro:bit needs turning off. What we really want is for the game to finish and go back to the home screen if we lose all of our lives. Change the A+B event with a new Lives variable.



```
on button A+B pressed
  if (Running = 0)
  then
    set Lives to 3
    set Running to 1
    set AlienMoveChance to 4
    Alien set brightness to 255
```

Modify the end of the forever loop so that the game no longer ends but the score is shown and the game paused if the lives run out.



```
if (AlienRocket y >= 4)
then
  if (AlienRocket brightness >= 100)
  then
    AlienRocket set brightness to 0
    if (AlienRocket x = Ship x)
    then
      start melody power down repeating once
      change Lives by -1
      if (Lives = 0)
      then
        set Running to 0
        pause
        clear screen
        show string "Game Over!"
        show string "Score"
        show number score
        pause (ms) 1000
        show leds
```

## Extending the game

There are many ways that this game can be extended. Just a few ideas are given below.

- Speed up the game as the score increases. i.e. make the alien move faster and shoot more often. Good points to increase the speed are when the scores hit 5, 10, 15 and 20.
- Award the player an extra life for each 5 aliens that are shot.
- When the alien is shot by the player, move it to a different starting square rather than it reappear where it was shot.
- Add in extra sound effects to the game. Maybe play a melody on the home screen or something when the players game ends
- Maintain the high score between games and let the player know if they have beaten the highest score.
- Add in a second alien when the player scores 10 points.