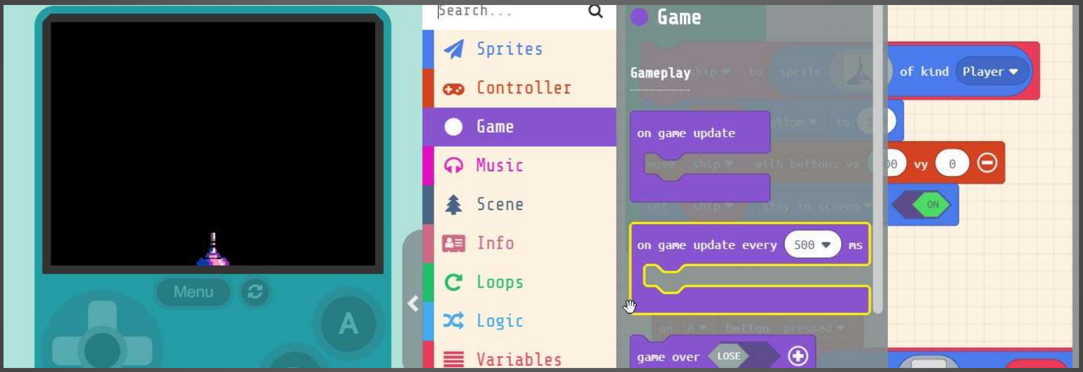
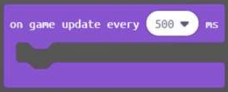
SPACE SHOOTER – EPISODE 02

In this presentation we'll add more elements to our Space Shooter game.

Let’s add something to shoot at in our new game!!

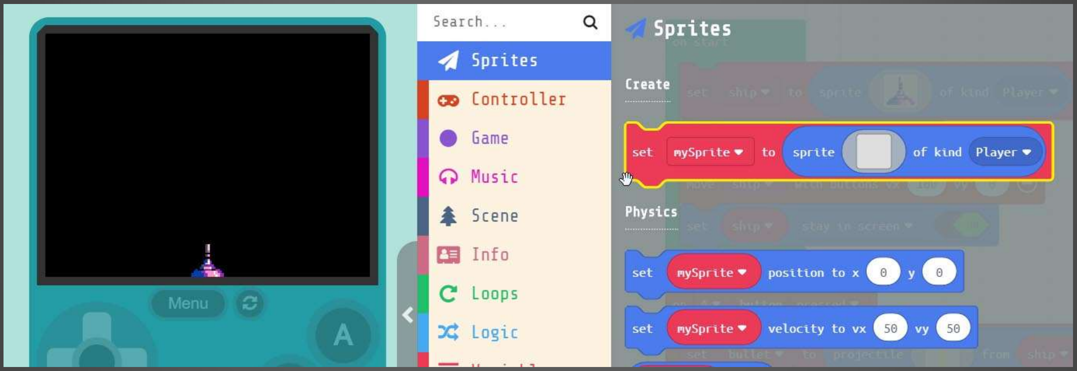
Let’s add something to shoot at in our game.



Grab the  block.

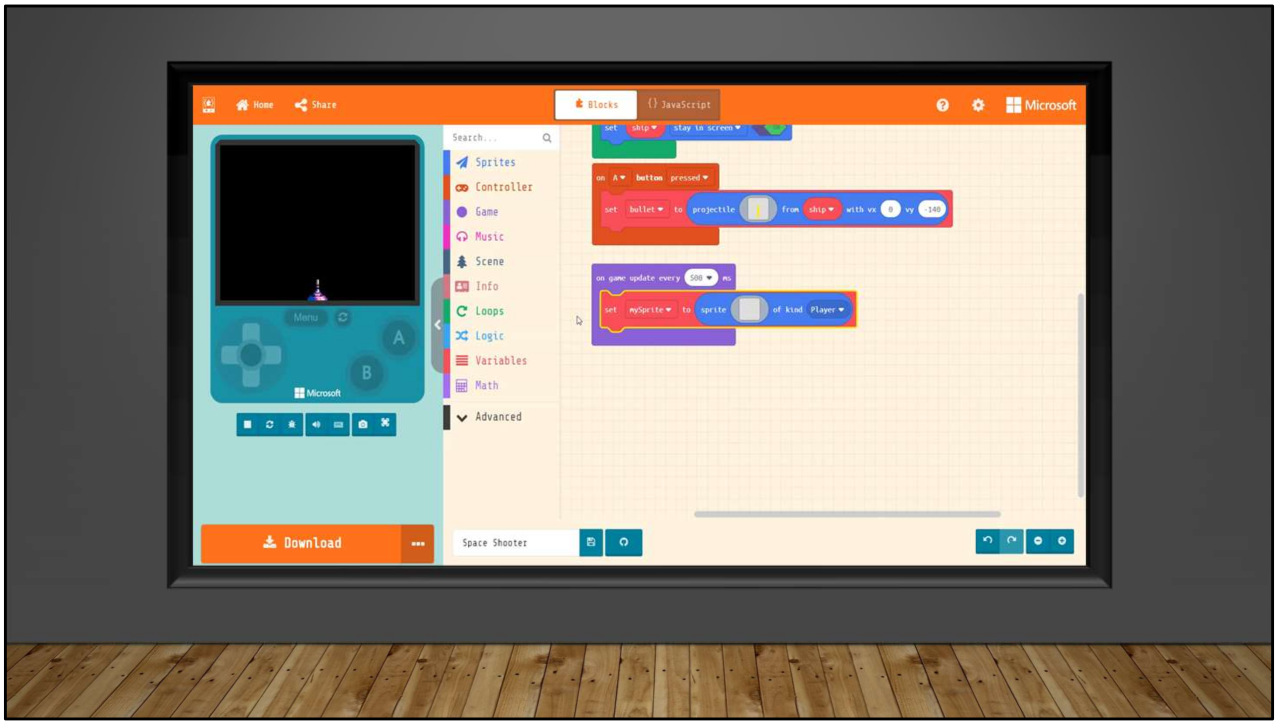
This event block runs the code at specific intervals.
In this case every 500 milliseconds.

First we need to grab the '**on game update every 500 ms**' block. It can be found under 'Game' in the menu. This type of event block runs whatever code we put inside it every ½ a second which is 500 milliseconds. Drag this block into the work area.


Search...
Sprites
Controller
Game
Music
Scene
Info
Loops
Logic
Menu
A
Sprites
Create
set mySprite to sprite of kind Player
Physics
set mySprite position to x 0 y 0
set mySprite velocity to vx 50 vy 50

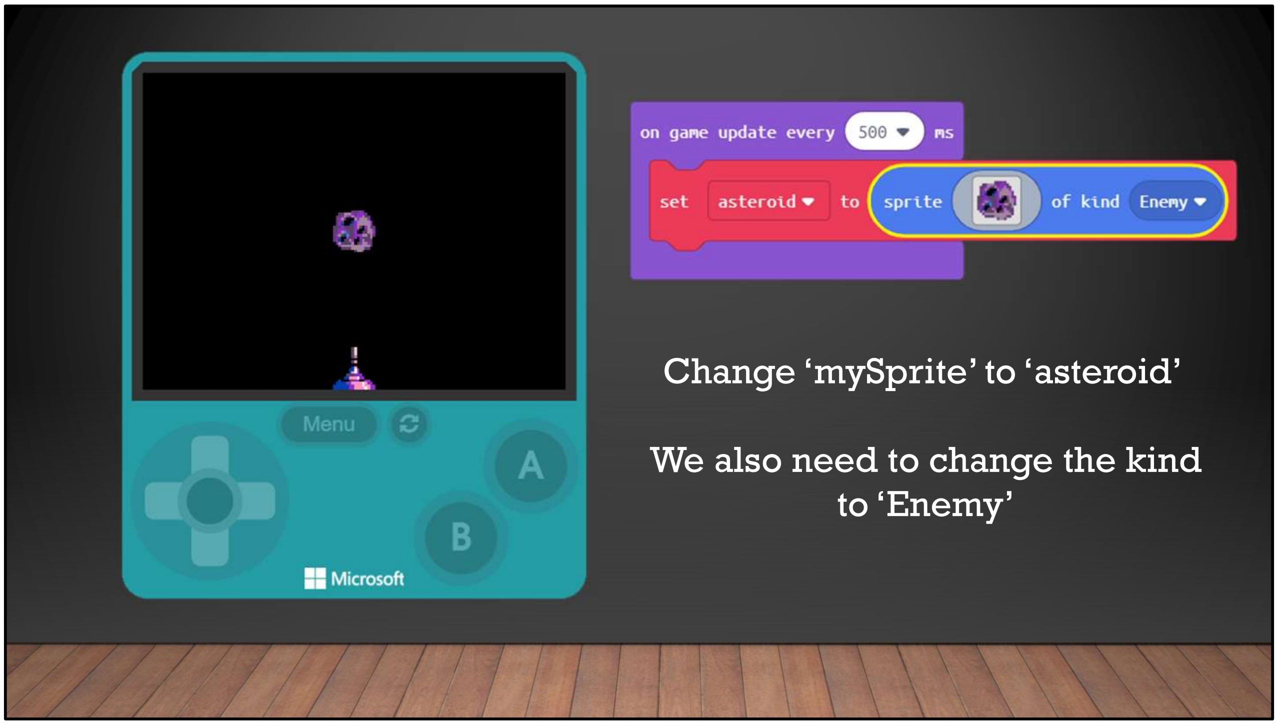
Grab the `set mySprite to sprite of kind Player` block.
We need this block to create a new Sprite.
Drag into the `on game update every 500 ms` block

Next, we'll create an enemy to shoot at. To do this we need to add another Sprite to the project. Grab the '**set mySprite to**' block and put it inside the '**on game update**' block we just added to the work area.



```
set ship to stay in screen
on A button pressed
    set bullet to projectile from ship with vx 0 vy -140
on game update every 500 ms
    set mySprite to sprite of kind Player
```

Now let's add an Asteroid Sprite to the block. To do this click on the blank gray box inside the ‘**set mySprite**’ block. This will bring up the Sprite editor. This is where Sprites are created or selected. We will grab one of the pre-created Sprites from the ‘Gallery’. Click on gallery and scroll down until you see one of the ‘Asteroids’, select it and press ‘DONE’

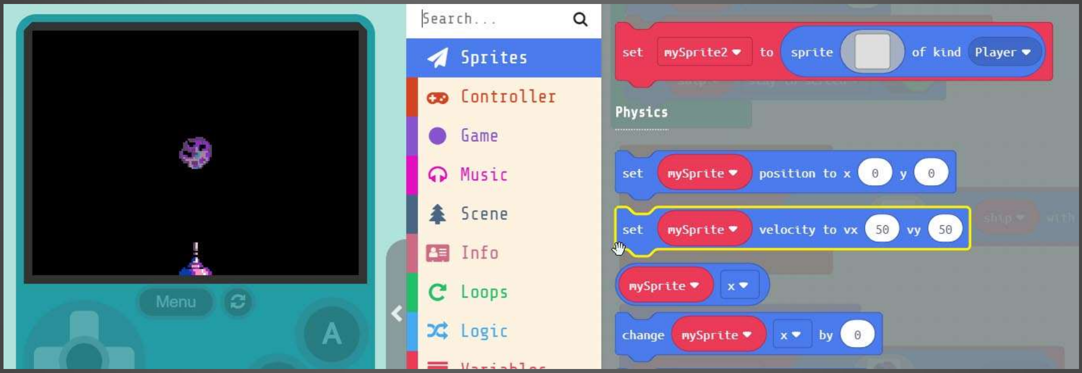


```
on game update every 500 ▾ ms
  set asteroid ▾ to sprite [ ] of kind Enemy ▾
```

Change 'mySprite' to 'asteroid'

We also need to change the kind to 'Enemy'

Now that we have an asteroid on the screen lets change the parameters inside the block. First change 'mySprite' to 'asteroid'. Remember what we said about meaningful names for our variables? We also need to change what kind of Sprite it is. In this case we need to change it to 'Enemy'

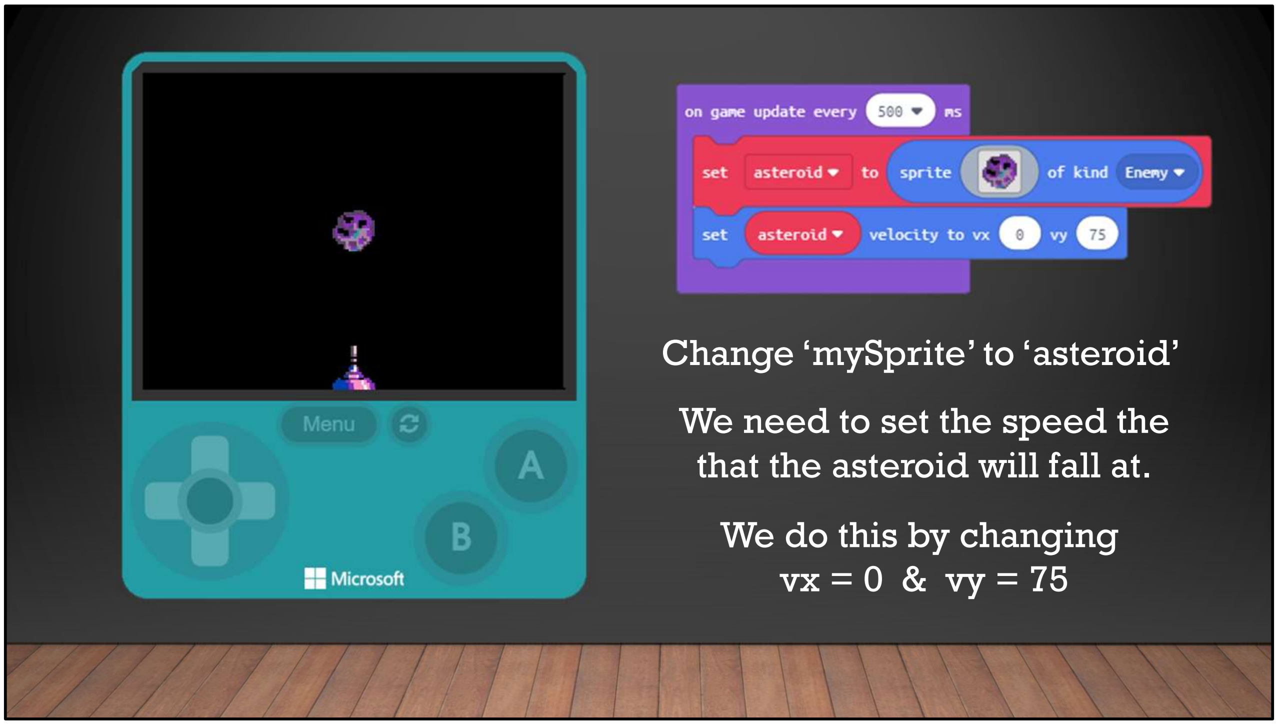

Search...
Sprites
Controller
Game
Music
Scene
Info
Loops
Logic
Menu
A
set mySprite2 to sprite of kind Player
Physics
set mySprite position to x 0 y 0
set mySprite velocity to vx 50 vy 50
mySprite x
change mySprite x by 0

Grab the set mySprite velocity to vx 50 vy 50 block.

Add this block just below the 'asteroid' sprite block

inside the on game update every 500 ms block

Let's make our 'asteroid' fall from the sky. To do this we need to grab the '**set mySprite velocity**' block. Add this block to the bottom of our '**on game update**' block in the work area.



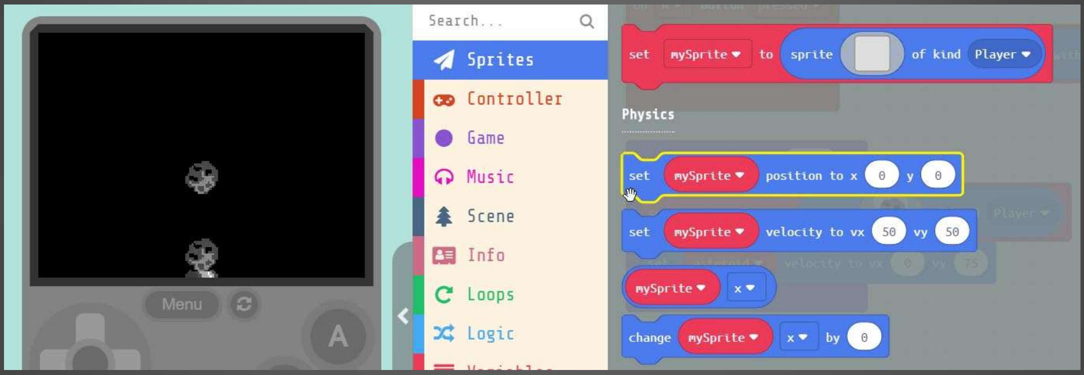
```
on game update every 500 ms
    set asteroid to sprite [ ] of kind Enemy
    set asteroid velocity to vx 0 vy 75
```

Change ‘mySprite’ to ‘asteroid’

We need to set the speed the that the asteroid will fall at.

We do this by changing
vx = 0 & vy = 75

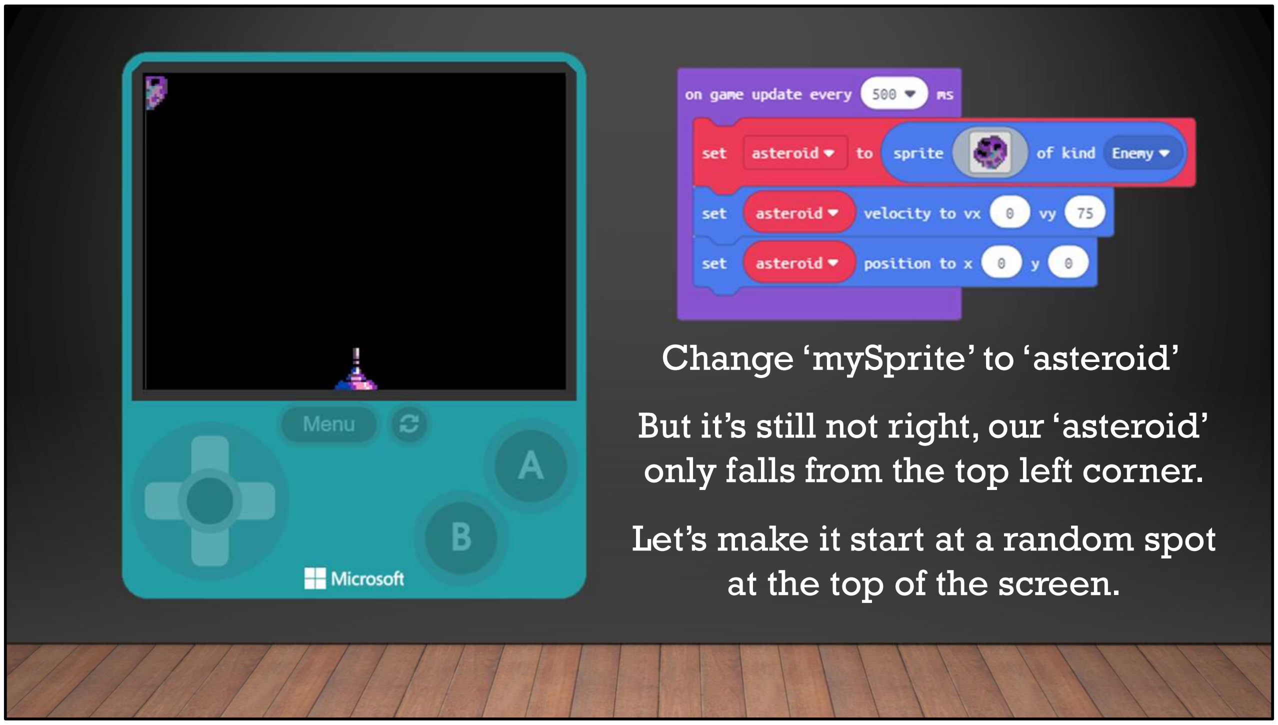
Change ‘mySprite’ to asteroid. Next change the vx=0 which means that it won’t move in the X direction. Set the vy to ‘75’ this is how fast the asteroid will fall on the Y – axis. But as you can see our ‘asteroid’ starts in the default center and just falls straight down on the player. Let’s add a new block to make the ‘asteroid’ start at the top of the screen.


Search...
Sprites
Controller
Game
Music
Scene
Info
Loops
Logic
set mySprite to sprite of kind Player
Physics
set mySprite position to x 0 y 0
set mySprite velocity to vx 50 vy 50
mySprite x
change mySprite x by 0
Menu
A

Grab the set mySprite position to x 0 y 0 block.

Add this block to the bottom of the on game update every 500 ms block

Now grab the ‘**set mySprite position**’ block from the menu under ‘Sprites’ drag this block to the bottom of our ‘**on game update every 500 ms**’ block.



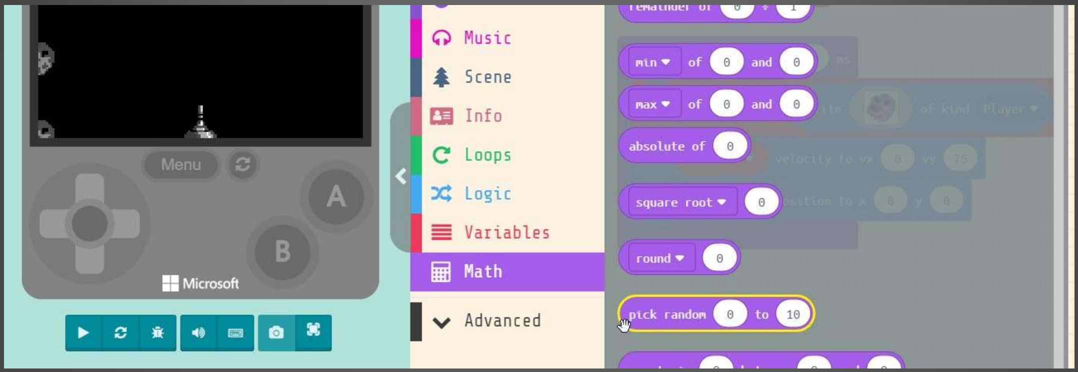
```
on game update every 500 ms
    set asteroid to sprite of kind Enemy
    set asteroid velocity to vx 0 vy 75
    set asteroid position to x 0 y 0
```

Change ‘mySprite’ to ‘asteroid’

But it’s still not right, our ‘asteroid’ only falls from the top left corner.

Let’s make it start at a random spot at the top of the screen.

Change ‘mySprite’ to asteroid. This sets the ‘asteroid’ start position to the top left corner of the screen. But it’s still not exactly what we want. The ‘asteroid’ only falls from the corner. Let’s make it start at a random spot at the top of the screen.

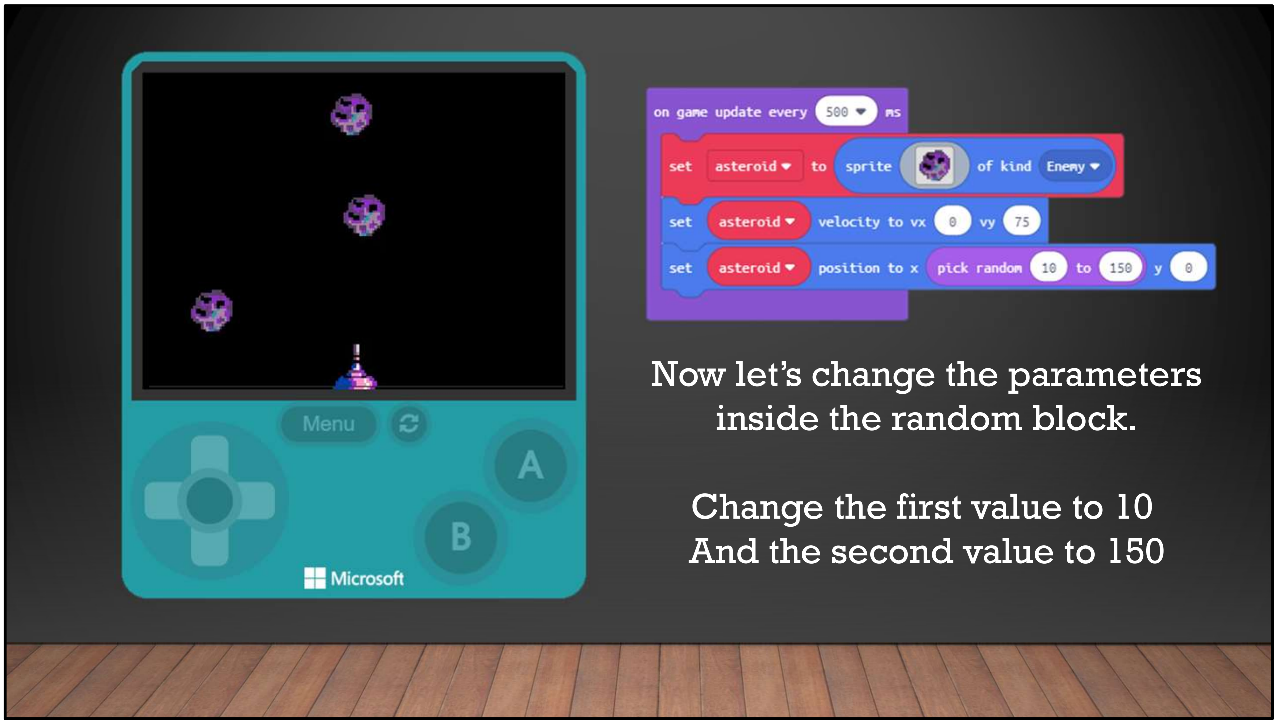


Grab the `pick random 0 to 10` block.

Put it inside the 'x' parameter

in the `set asteroid position to x 0 y 0` block

To make our asteroid start in a random location we need to grab the '**pick random 0 to 10**' block found under 'Math' in the menu. Drag this and put it inside the 'X' value window inside our '**set asteroid position**' block.

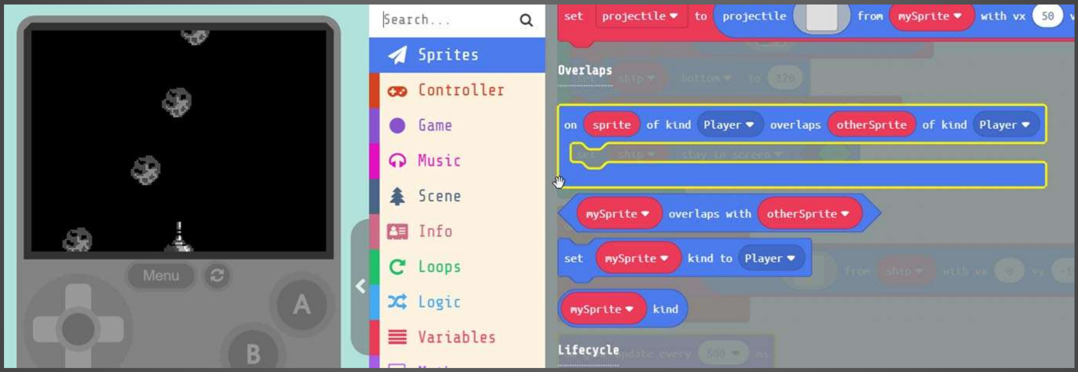


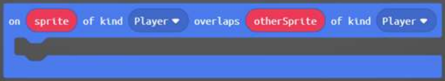
```
on game update every 500 ▾ ms
    set asteroid ▾ to sprite [ ] of kind Enemy ▾
    set asteroid ▾ velocity to vx 0 vy 75
    set asteroid ▾ position to x pick random 10 to 150 y 0
```

Now let's change the parameters inside the random block.

Change the first value to 10
And the second value to 150

Now we need to change the parameters inside the 'pick random' block we just added. Change the first parameter to '10' and the second one to '150'. Now our asteroids are randomly generated at the top of the screen and fall towards our player.

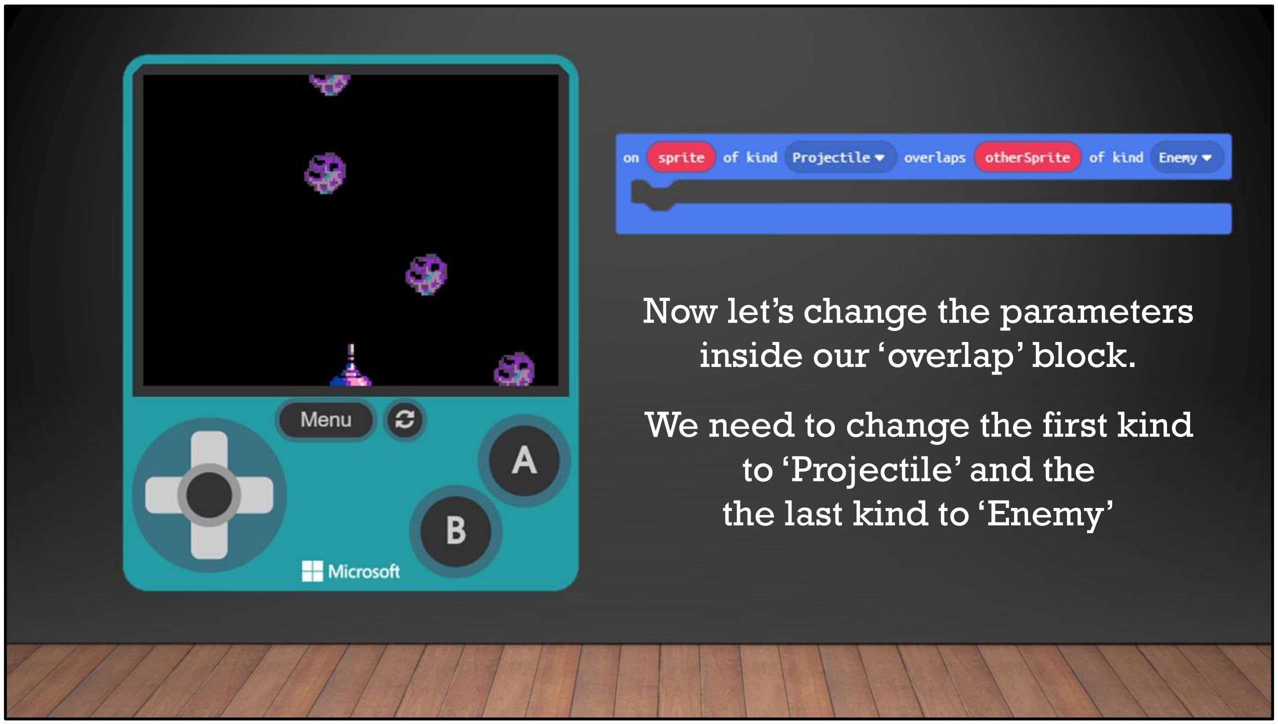

Search...
Sprites
Controller
Game
Music
Scene
Info
Loops
Logic
Variables
Menu
A
B
set projectile to projectile from mySprite with vx 50
Overlaps
on sprite of kind Player overlaps otherSprite of kind Player
mySprite overlaps with otherSprite
set mySprite kind to Player
mySprite kind
Lifecycle

Grab the  block.

on sprite of kind Player overlaps otherSprite of kind Player

Drag this ‘Event’ block into the work area

You may have tried to shoot at the ‘asteroid’ and nothing happened. We need to add collision detection to our game. Under ‘Sprites’ in the menu select the ‘**on sprite of kind player overlaps with otherSprite**’ Event block. Remember that events are standalone blocks that run the code inside them when something happens. In this case when we have an overlap of Sprites.

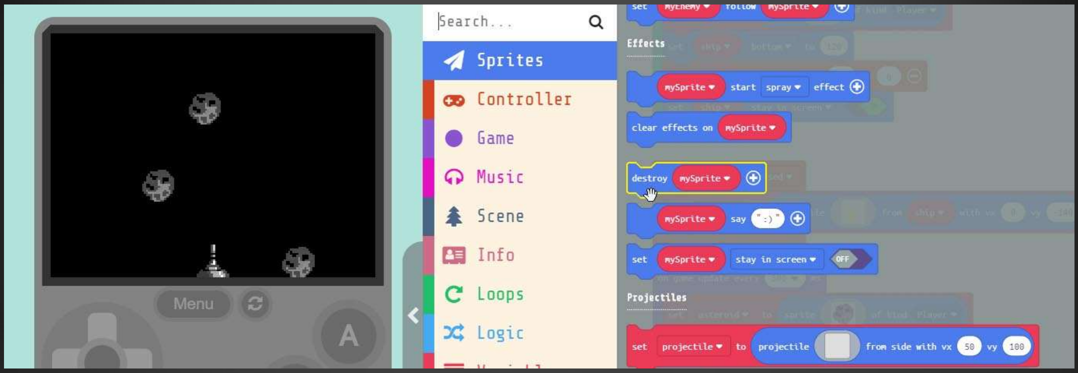


```
on sprite of kind Projectile ▾ overlaps otherSprite of kind Enemy ▾
```

Now let's change the parameters inside our 'overlap' block.

We need to change the first kind to 'Projectile' and the the last kind to 'Enemy'

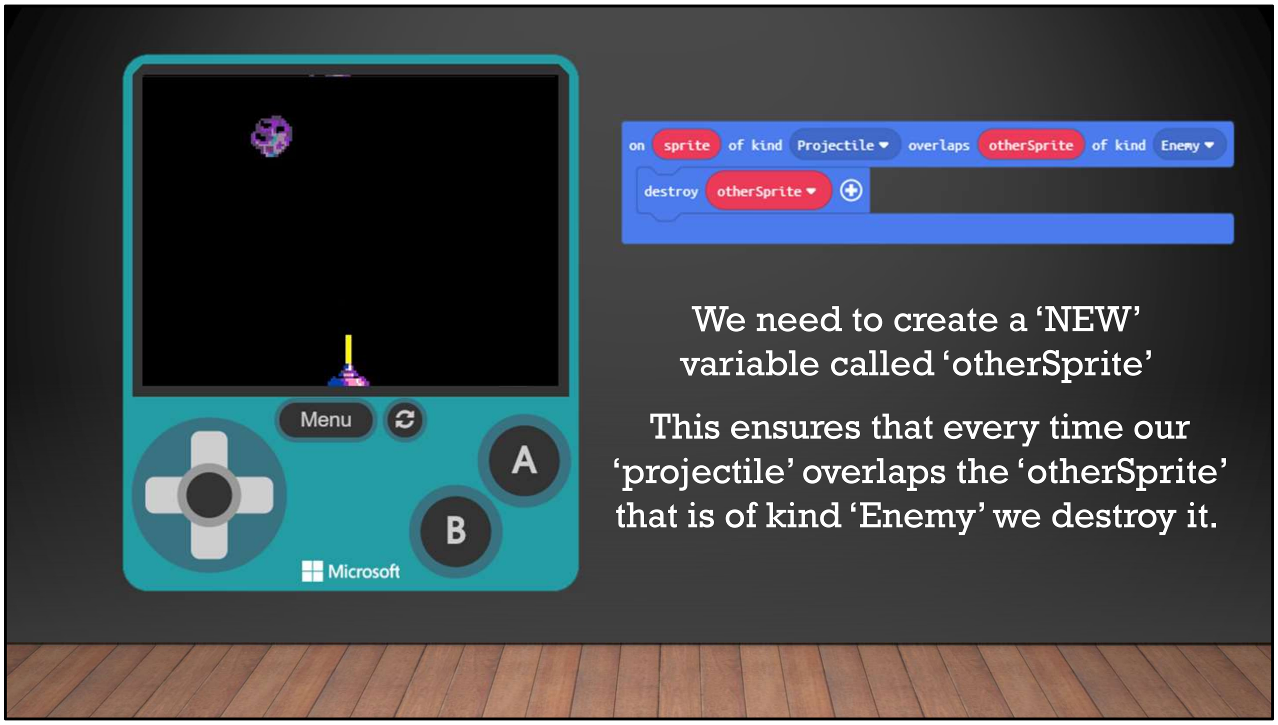
We need to change the kind of Sprites we want to detect when overlapping. Change the first kind from 'Player' to "Projectile', next change the last kind from 'Player' to 'Enemy'. So now when a 'Projectile', our 'bullet', overlaps an 'Enemy', our Asteroids, the code we place inside this block will run.


Search...
Sprites
Controller
Game
Music
Scene
Info
Loops
Logic
Menu
A
Effects
mySprite start spray effect
clear effects on mySprite
destroy mySprite
mySprite say ":)"
set mySprite stay in screen OFF
Projectiles
set projectile to projectile from side with vx 50 vy 100

Grab the destroy mySprite block.

Put inside our on sprite of kind Projectile overlaps otherSprite of kind Enemy Event block.

You may have tried to shoot at the ‘asteroid’ and nothing happened. We need to add collision detection to our game. Under ‘Sprites’ in the menu select the ‘**on sprite of kind player overlaps with otherSprite**’ Event block. Remember that events are standalone blocks that run the code inside them when something happens. In this case when we have an overlap of Sprites.

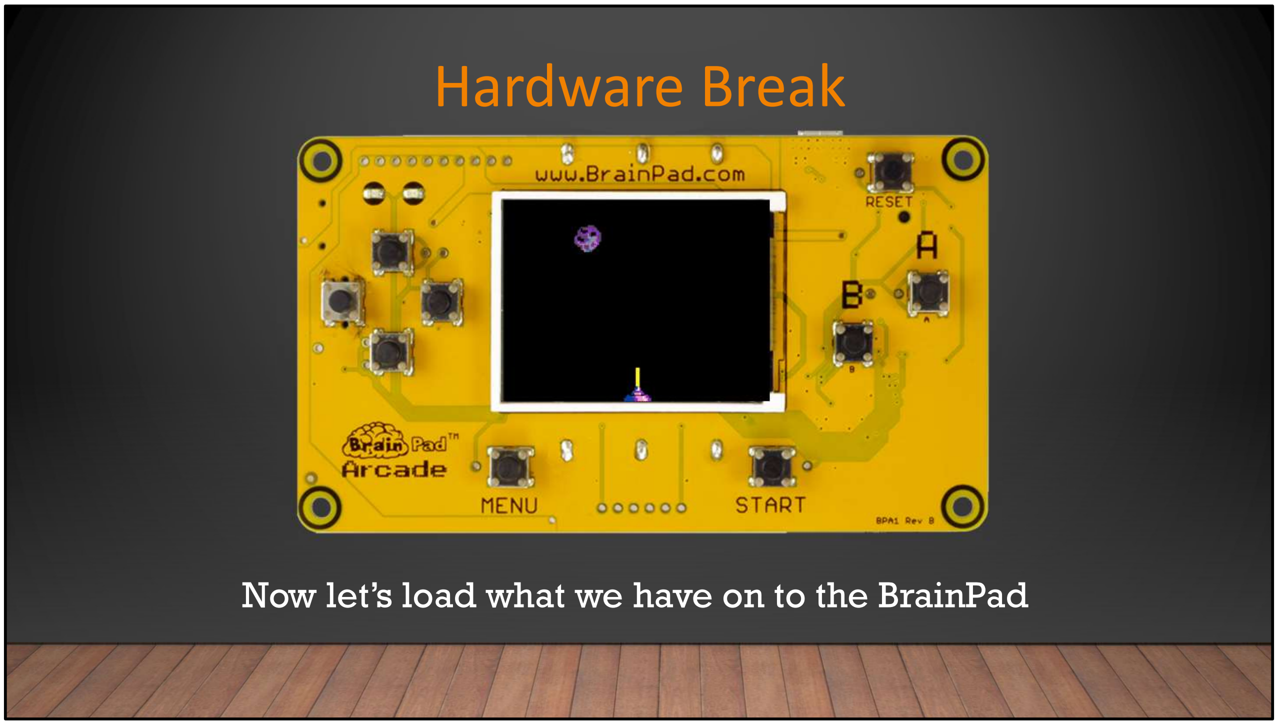


```
on sprite of kind Projectile ▾ overlaps otherSprite of kind Enemy ▾
    destroy otherSprite ▾ ⊕
```

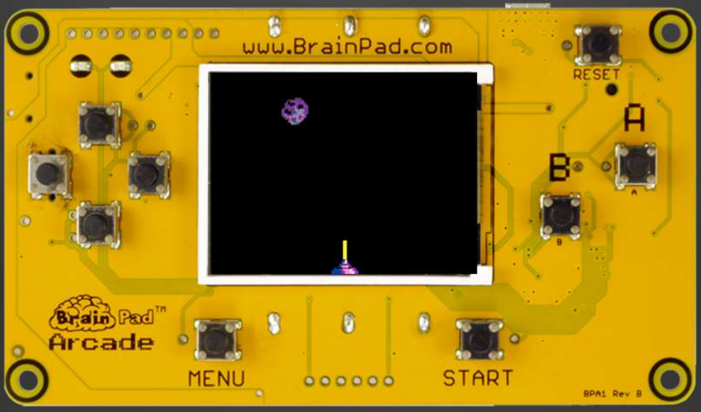
We need to create a 'NEW' variable called 'otherSprite'

This ensures that every time our 'projectile' overlaps the 'otherSprite' that is of kind 'Enemy' we destroy it.

We need to create a new variable by clicking the down arrow inside the '**destroy mySprite**' block. We need to name the new variable 'otherSprite'. This ensures that every time our 'projectile bullet' overlaps the 'otherSprite' that is an "Enemy' the 'otherSprite' will be destroyed. You ask, why we don't set this to destroy 'asteroid' instead? If we did this then hitting just one 'asteroid' Sprite with our bullet, would destroy ALL the asteroids on the screen.



Hardware Break

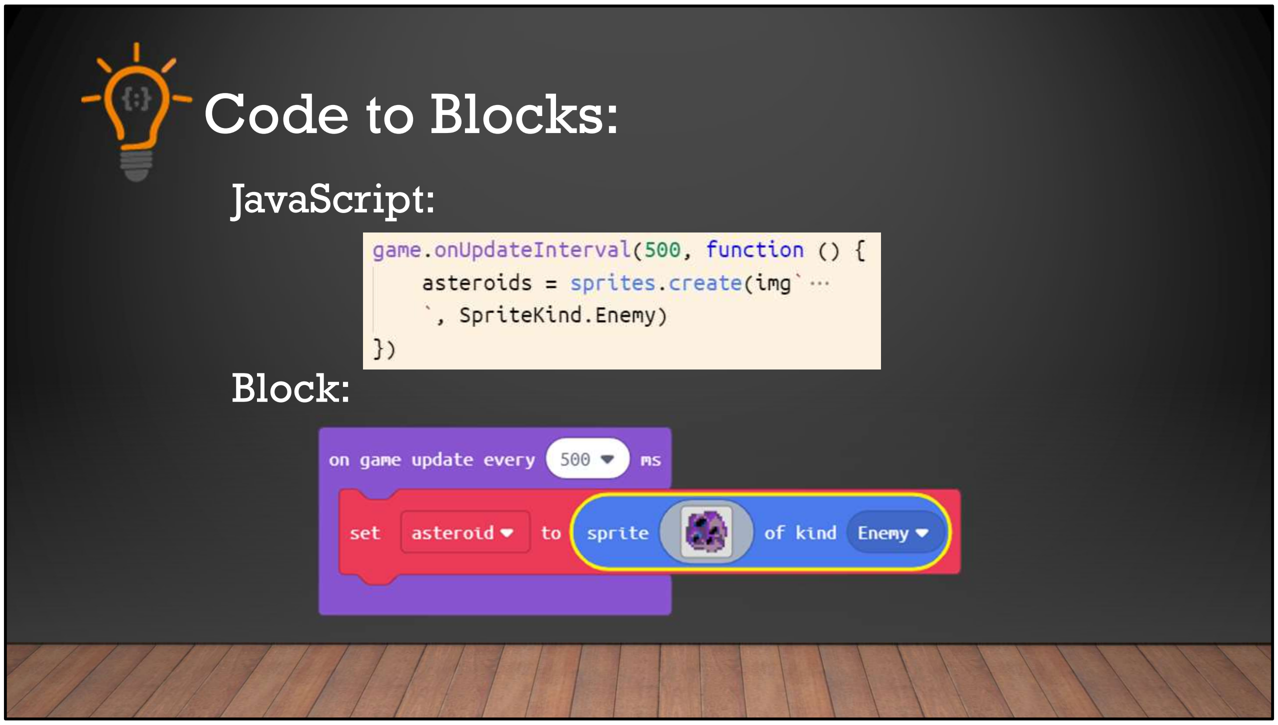


Now let's load what we have on to the BrainPad

Now let's load what we have so far on the BrainPad.

EXTRA CREDIT

Now let’s load what we have so far on the BrainPad.



Code to Blocks:

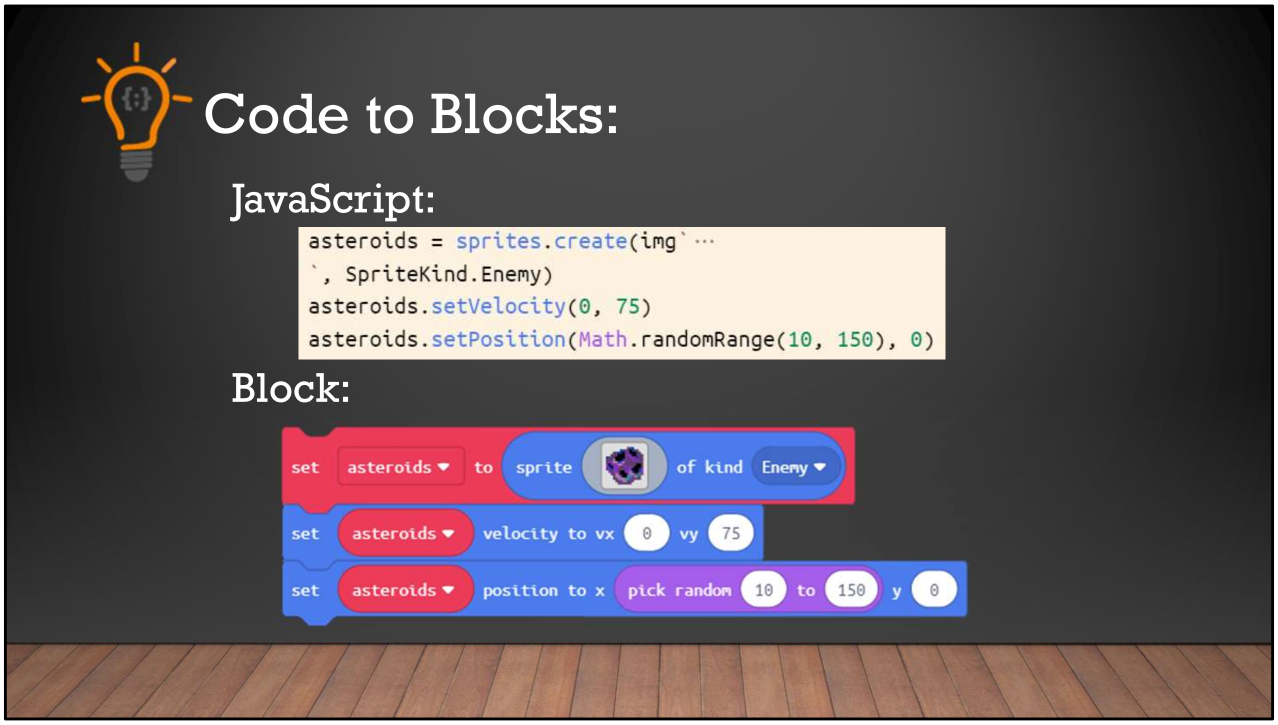
JavaScript:

```
game.onUpdateInterval(500, function () {
    asteroids = sprites.create(img`...
    `, SpriteKind.Enemy)
})
```

Block:

on game update every 500 ms
set asteroid to sprite of kind Enemy

Now you may have noticed that '**on A button pressed**' block looks different then the other blocks. It looks more like the '**on start**' block, like a square. This is because it's what is known as an 'EVENT' block. An 'EVENT' block runs the code we place inside of it when an 'EVENT' occurs. Like a button press. '**on start**' is an 'EVENT' too. The first time we run our program, everything inside of it runs.



Code to Blocks:

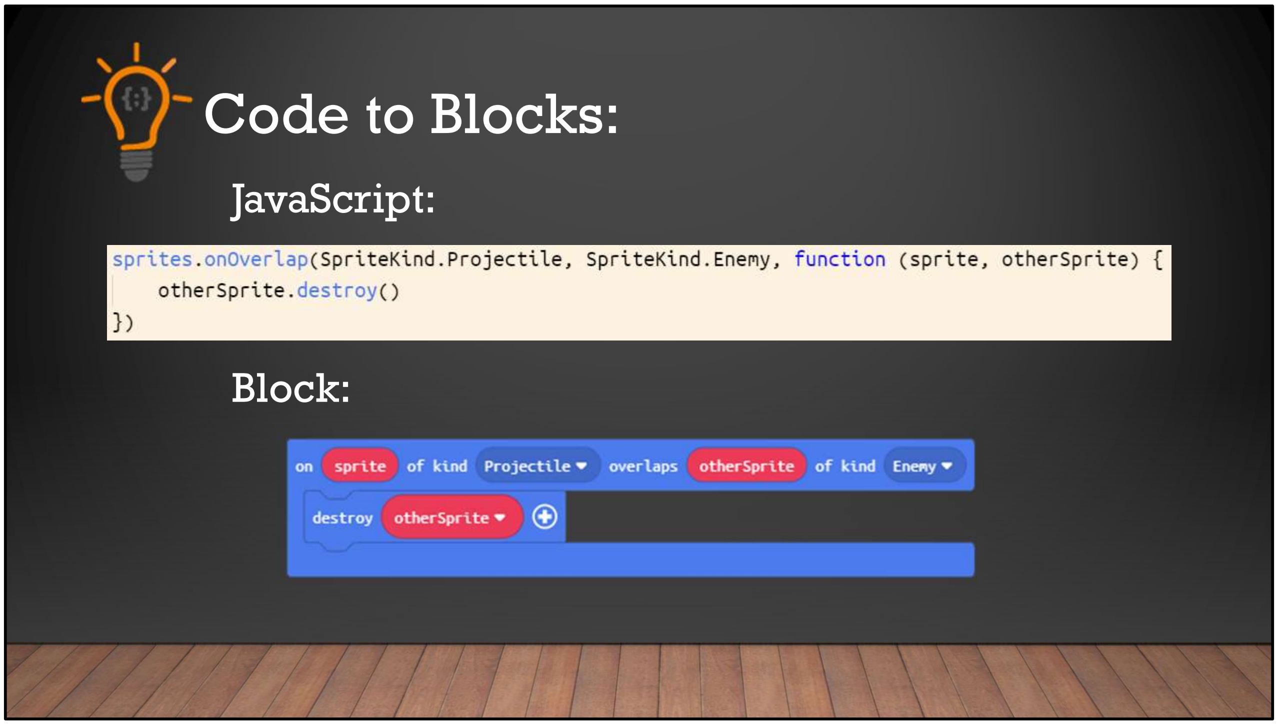
JavaScript:

```
asteroids = sprites.create(img`···
`, SpriteKind.Enemy)
asteroids.setVelocity(0, 75)
asteroids.setPosition(Math.randomRange(10, 150), 0)
```

Block:

```
set asteroids ▾ to sprite [image] of kind Enemy ▾
set asteroids ▾ velocity to vx 0 vy 75
set asteroids ▾ position to x pick random 10 to 150 y 0
```

Now you may have noticed that ‘**on A button pressed**’ block looks different then the other blocks. It looks more like the ‘**on start**’ block, like a square. This is because it’s what is known as an ‘EVENT’ block. An ‘EVENT’ block runs the code we place inside of it when an ‘EVENT’ occurs. Like a button press. ‘**on start**’ is an ‘EVENT’ too. The first time we run our program, everything inside of it runs.



Code to Blocks:

JavaScript:

```
sprites.onOverlap(SpriteKind.Projectile, SpriteKind.Enemy, function (sprite, otherSprite) {
    otherSprite.destroy()
})
```

Block:

on sprite of kind Projectile overlaps otherSprite of kind Enemy
destroy otherSprite

Now you may have noticed that ‘**on A button pressed**’ block looks different then the other blocks. It looks more like the ‘**on start**’ block, like a square. This is because it’s what is known as an ‘EVENT’ block. An ‘EVENT’ block runs the code we place inside of it when an ‘EVENT’ occurs. Like a button press. ‘**on start**’ is an ‘EVENT’ too. The first time we run our program, everything inside of it runs.