

## Segundo ejercicio Sprites



### Descripción

En este proyecto se trabaja el aspecto de variables, creación de un oponente, cambio de imagen y animación de un sprite.

Un sprite es un elemento gráfico dibujado en un mapa de bits. Con esto se trabajará en el ejercicio. A este elemento se le aplicarán distintos atributos como posición, velocidad, aceleración, etc.

El juego trata de un pez que necesita conseguir las perlas que le ofrece la ostra, pero el tiburón no se lo pondrá fácil, ya que ira siempre en su busca.

Para ello accederemos a [MakeCode Arcade](#) y realizaremos las operaciones necesarias.

### Objetivos de programación y diseño

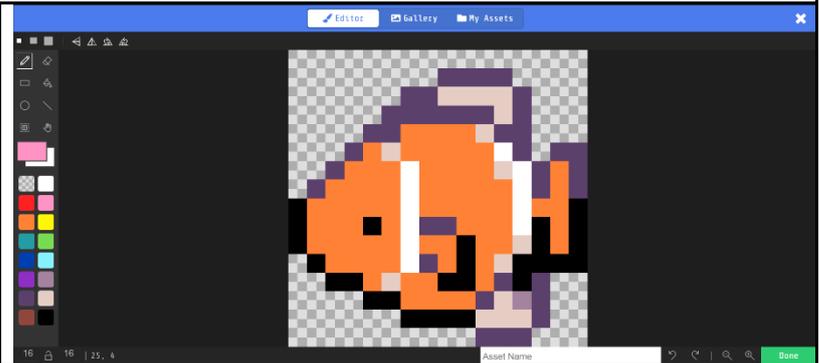
- Crear un sprite de tipo "Enemy" que esté animado y siga al jugador.
- Crear un sprite de tipo "Food" que cambie de estado
- Programar la interacción entre sprites.

## Programación del juego

### CREACIÓN DE ASSETS

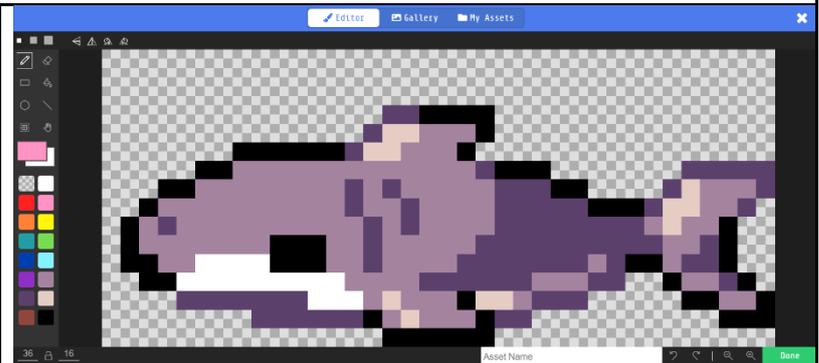
#### CREACIÓN SPRITE PROTAGONISTA

Te recomendamos utilizar una matriz de 16x16 px para el **Sprite** de **Protagonista**.



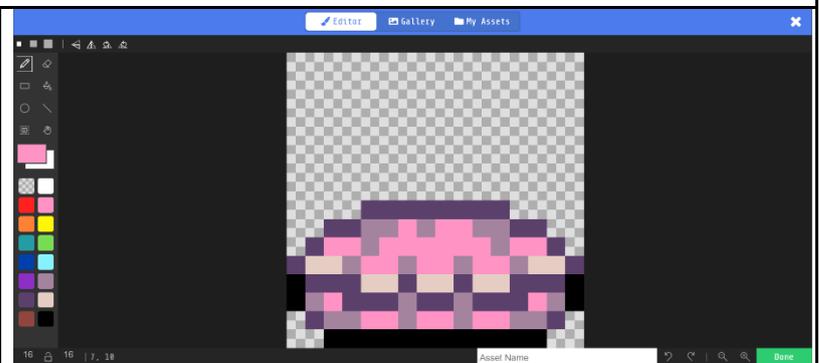
#### CREACIÓN SPRITE ENEMIGO

Te recomendamos utilizar una matriz de 36x16 px para el **Sprite** de **Tiburón**.

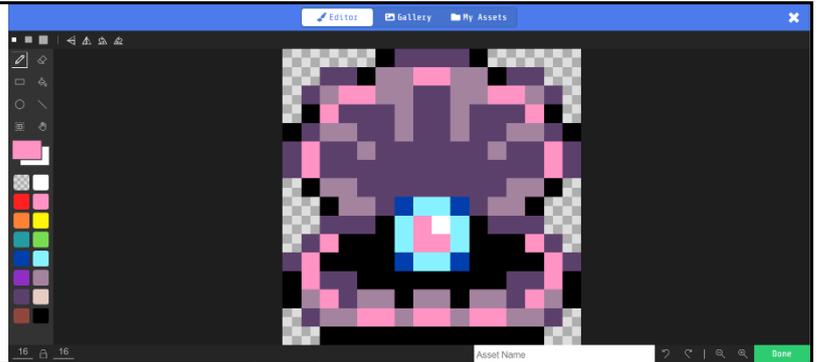


#### CREACIÓN SPRITES ADICIONALES

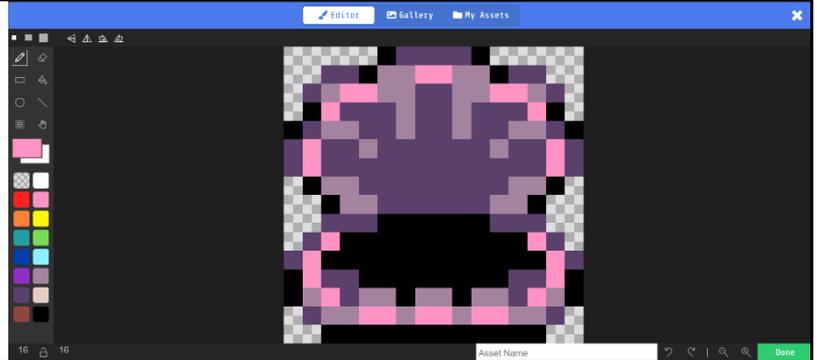
Te recomendamos utilizar una matriz de 16x16 px para el **Sprite** de **Ostra**.



Te recomendamos utilizar una matriz de 16x16 px para el **Sprite** de **Ostra2**.

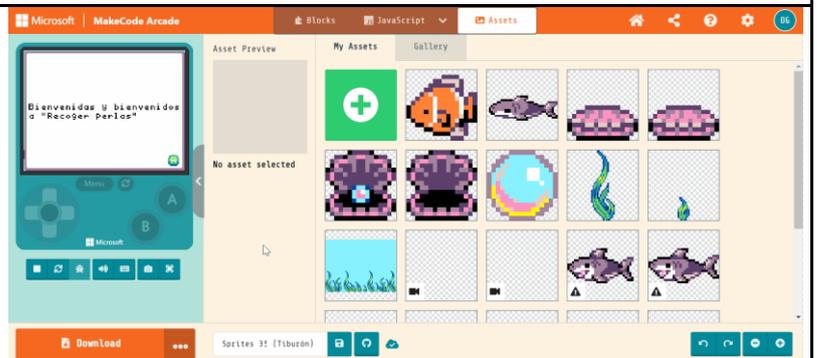


Te recomendamos utilizar una matriz de 16x16 px para el **Sprite** de **Ostra3**.



## CREACIÓN ANIMACIÓN

En la pestaña Assets, haremos clic en el icono "+". Ahí elegiremos la pestaña Gallery y elegiremos nuestra animación del tiburón. Por último, le ponemos nombre y clicamos en Done.



## PROGRAMACIÓN PRINCIPAL

### CREACIÓN INICIO DEL JUEGO

Empezamos con la creación de los distintos **sprites** que aparecerán en el ejercicio y su posición dentro del escenario. Le daremos control al **sprite** de tipo "Player" que en este caso será el pez. Al **sprite** de tipo "Enemy" le indicaremos que siga al pez con el bloque "set myEnemy follow **mysprite**". Haciendo clic en el signo + de ese bloque, es posible disminuir la velocidad, por ejemplo, a un valor de 20, para dar más posibilidad de jugar al "Player". Además, crearemos el Sprite de ostra, que será de tipo "Food"

Añadiremos un color al **fondo del escenario**.

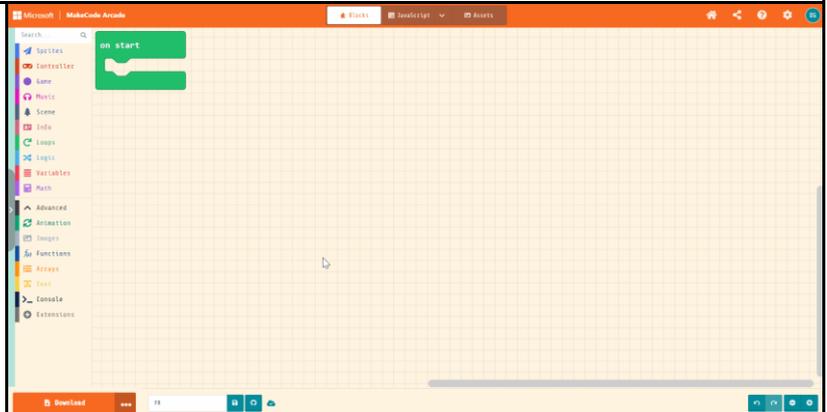
Usaremos el bloque "animate" del bloque avanzado "Animation" para crear una animación. Importante dejar puesto el interruptor de "loop" para que una vez termine la animación, se repita de nuevo sin parar.

```

on start
  set hero to sprite  of kind Player
  set hero position to x 20 y 59
  move hero with buttons +
  set shark to sprite  of kind Enemy
  set shark position to x 120 y 61
  set shark follow hero with speed 20 -
  set background color to 
  set clam to sprite  of kind Food
  animate shark
  frames 
  interval (ms) 500
  loop ON
  start countdown 60 (s)
  
```

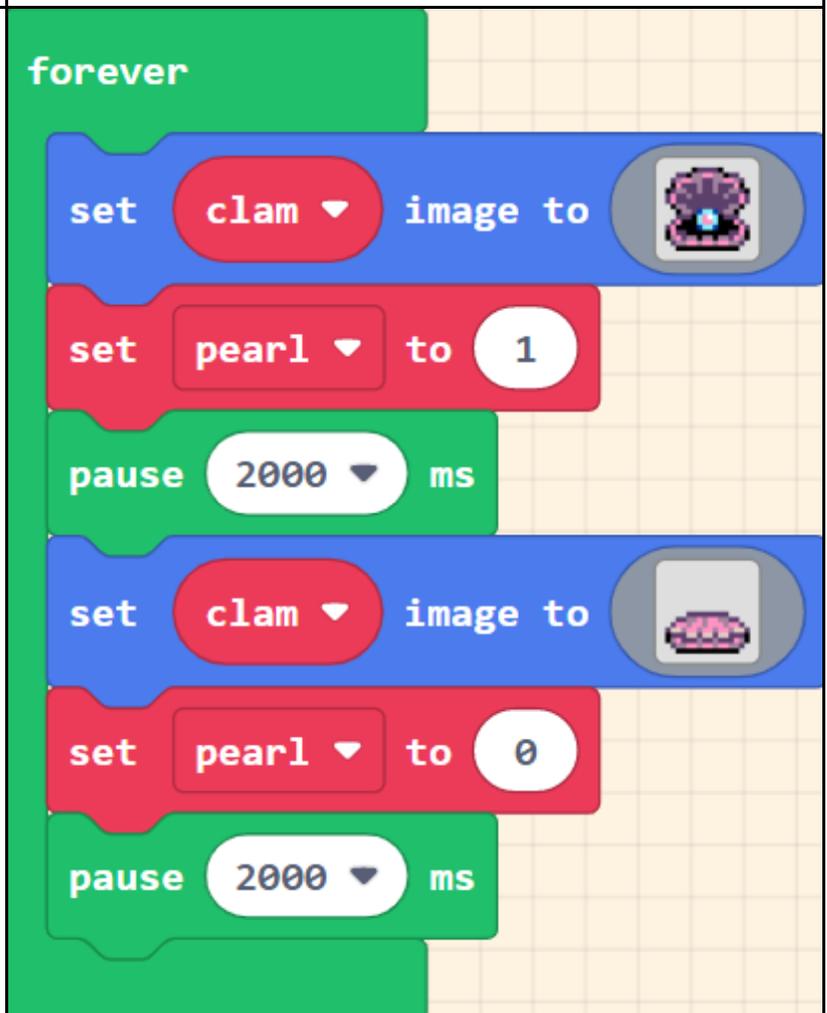
## CREACIÓN DE LA MECÁNICA APERTURA DE LA OSTRAS

Primero crearemos una **variable** llamada **"pearl"**.  
Para ello, vamos al panel de la izquierda en "Variable" y después en "Make" a variable.



Dentro de un bloque **"forever"** introduciremos los bloques para cambiar la imagen, estableceremos el valor de la variable **"pearl"** a 1 para cuando esté presente y 0 para cuando no. Esto sería cómo una abstracción de cuándo tenemos la perla o no.

Importante que la espera esté después del cambio de imagen y el establecimiento de la variable para que resulte visual y el jugador tenga un intervalo (en este ejemplo de 2 segundos) de tiempo para poder obtener puntos.



Entonces en este grupo de bloques vemos visualmente cuando está la perla y también se lo indicamos en el juego, teniendo una diferencia de dos segundos en cada estado.

**pearl = 1 -> La perla está y se puede atrapar.**  
**pearl = 0 -> La perla no está y no se puede atrapar.**

### CREACIÓN MECÁNICA OBTENCIÓN DE PERLA Y SUMA DE PUNTOS

Cuando el pez toque a la ostra, si hay perla, la coge cambiando la imagen, aumentando la **puntuación** y estableciendo que no hay perla.

**Si no hay perla, al tocar el pez a la ostra, no ocurre nada**

```

on sprite of kind Player overlaps otherSprite of kind Food
  if pearl = 1 then
    set clam image to [clam]
    change score by 1
    set pearl to 0
  
```

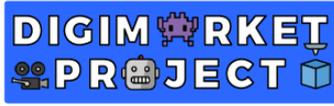
### CREACIÓN MECÁNICA FINAL DEL JUEGO

Al tocar el pez al tiburón, la partida termina con el bloque **game over**.

```

on sprite of kind Player overlaps otherSprite of kind Enemy
  game over LOSE
  
```

Con esta programación, usaremos al sprite de tipo "Player" para obtener puntos al tocar al sprite de tipo "Comida" mientras estos se generan por toda la pantalla con un temporizador de un minuto. Ahora, es tu turno de personalizarlo y añadirle contenido. Aquí te dejamos el nuestro para que te inspires un poco: [https://makecode.com/\\_AHWED8PpjPFs](https://makecode.com/_AHWED8PpjPFs)



## Glosario

**Sprite:** Un sprite es un elemento gráfico dibujado en un mapa de bits. A este elemento se le aplicarán distintos atributos como posición, velocidad, aceleración, etc.

**Mapa de bits:** Es una rejilla de píxeles que se usa para dibujar sprites.

**Escenario:** Espacio donde se desarrolla el videojuego.

**Variable:** Es un espacio asociado a un identificador, en ese espacio hay un valor que puede ser modificado.

**Animación:** La animación de un sprite es la sensación de movimiento que tiene por los diferentes frames que lo componen.

**Bucles:** Secuencia de código que se ejecuta repetidas veces.

Ejemplo: Forever, while, for.

**Condicionales:** Secuencia de instrucciones que se ejecuten en función del valor de una condición.

Ejemplo: If, If...Else

**If:** Sentencia condicional que, según el resultado de una operación lógica, ejecuta una secuencia de instrucciones o se omite.