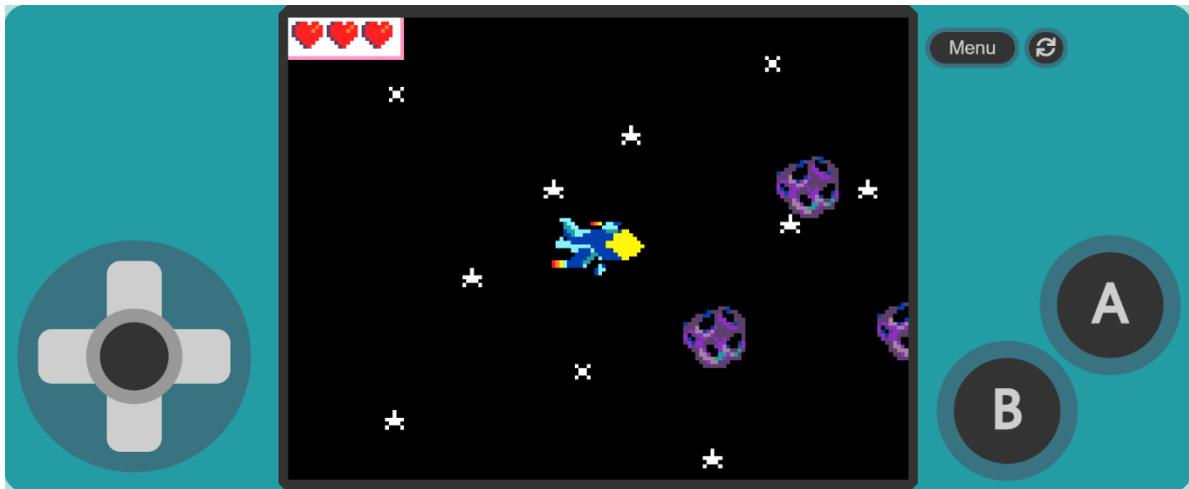


Primer Ejercicio Motion



Descripción

En este proyecto realizaremos un juego de efecto de desplazamiento lateral. Esto se consigue con la aparición y movimiento de asteroides, estrellas y con proyectiles.

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

Objetivos de programación y diseño

- Crear Sprite del Jugador, una Nave.
- Crear Un Sprite Enemigo, un Asteroide.
- Crear un Sprite de Proyectil.
- Crear Sprites de ambientación del escenario, Estrellas.
- Mecánica de desplazamiento lateral de Asteroide, Estrellas y Proyectil.
- Mecánica de colisión del Proyectil con el Asteroide.
- Mecánica de colisión del Asteroide con la Nave.
- Mecánica de movimiento del Jugador por el escenario.

Programación del juego

Aquí os dejamos los Assets y parte de la programación inicial

<https://makecode.com/EKC79cR9ogDi>

CREACIÓN DE ASSETS

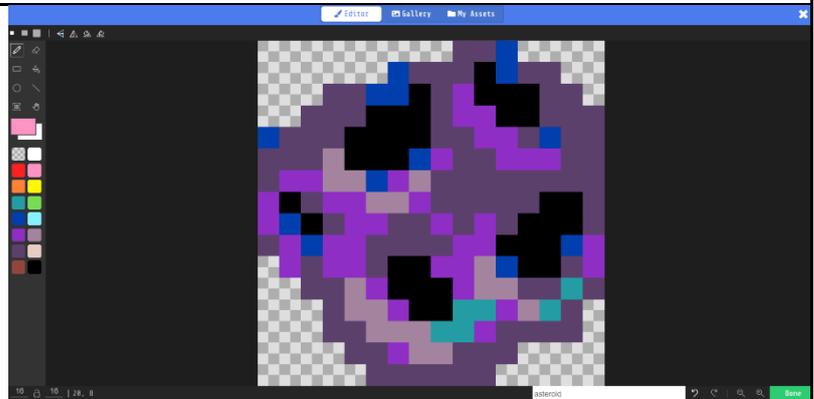
CREACIÓN SPRITE PRINCIPAL

Te recomendamos utilizar una matriz de 24x16 px para el Sprite de "Ship".

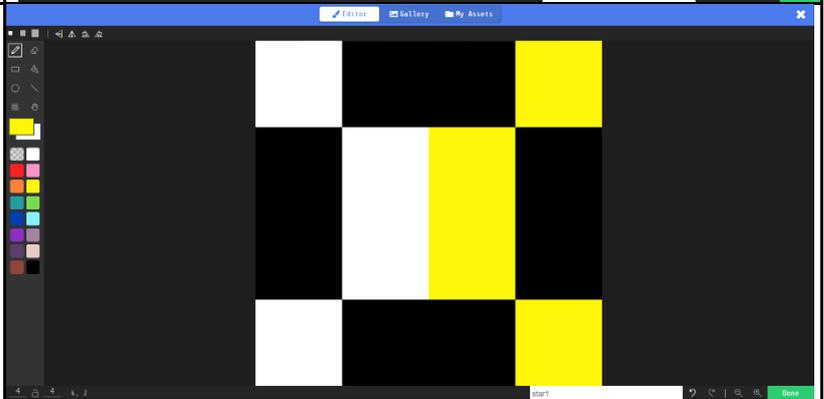


CREACIÓN SPRITES ADICIONALES

Crearemos el Sprite de asteroid en una matriz de 16x16 px.



Crearemos el Sprite de star1 en una matriz de 4x4px.





Crearemos el **Sprite** de **star2** en una matriz de 5x5px.

Crearemos el **Sprite** de **projectile** en una matriz de 8x8px.

PROGRAMACIÓN PRINCIPAL

CREACIÓN INICIO DEL JUEGO

Primero crearemos la **variable** **asteroidSpeed**. En ella guardaremos la velocidad que vaya a tener el asteroide.



Ahora, crearemos un bloque **on start**. Y en el metemos los siguientes bloques de código.

set asteroidSpeed to -50. Establecemos la velocidad del asteroide en -50 y se pone en negativo para que se mueva hacia la izquierda.

set Ship to Sprite (Sprite Ship) of kind Player. Aquí declaramos el **Sprite Ship** como **Player** y este aparecerá en el juego.

Por último, **set life to 3**. Le ponemos 3 vidas al jugador.

on start

set asteroidSpeed to -50

set Ship to sprite of kind Player

set life to 3

Ahora, colocaremos dos bloques de código.

El primero sirve para poder mover la nave, **move Ship with buttons vx 200 vy 200**. Para darle esta velocidad, hay que darle al símbolo +.

set Ship stay in screen on. Esto sirve para decirle al juego que la nave no se salga de la pantalla.

on start

set asteroidSpeed to -50

set Ship to sprite of kind Player

set life to 3

move Ship with buttons vx 200 vy 200

set Ship stay in screen on

MECÁNICA DE DISPARO

Vamos a crear la mecánica de disparo, con ello haremos que la nave lance proyectiles.

Para ello, en **Controller** elegiremos el bloque **on A button pressed**.

Dentro colocaremos el siguiente bloque: **set projectile to (projectile "introducir el Sprite de projectile" from Ship with vx 200 vy 0)** que se encuentra en **Sprites** apartado **Projectiles**. Con esto le decimos al juego que cada vez que pulsemos el botón **"A"** cree el **Sprite proyectil** desde la nave con una velocidad de 200 para que se mueva a la derecha.

```

on A button pressed
  set projectile to projectile from Ship with vx 200 vy 0
  
```

MECÁNICA GENERACIÓN DE ASTEROIDES

Ahora vamos a generar los asteroides.

Haremos que salgan cada medio segundo, así que, en **Game** cogeremos el bloque de código: **on game update every 500 ms**. Esto significa que compruebe el juego cada 500 milisegundos, es decir, medio segundo.

```

on game update every 500 ms
  
```

Ahora colocamos dentro: **set Asteroid to (Sprite "Introducir el Sprite de asteroid" of kind Enemy)**. Con esto lograremos crear el asteroide y lo declaramos Enemy.

change asteroidSpeed by -1. Con esto el juego le suma, a la **velocidad** de **-200** que tiene el asteroide, **-1**. Así le aumentamos la velocidad en **-1** al asteroide cada **medio segundo**.

```

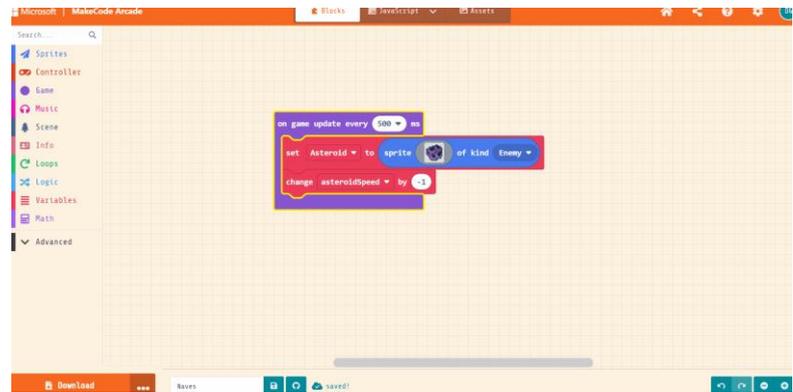
on game update every 500 ms
  set Asteroid to sprite of kind Enemy
  change asteroidSpeed by -1
  
```



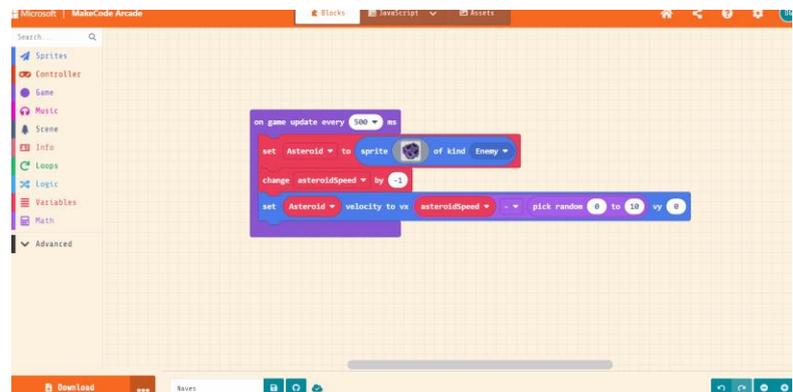
Para que los asteroides, no tengan todos la misma **velocidad**, además del **-1** anterior, vamos a sumarle un **número aleatorio de 0 a 10** y así tendrán distinta **velocidad**. Para lograr esto, vamos a colocar: **set Asteroid velocity to vx 50 vy 50**.

Como la velocidad es **vx**, entramos en **Math** y dentro de **vx** colocamos el bloque de código **0 - 0**: **set Asteroid velocity to vx (0 - 0) vy 0**.

Posteriormente, buscamos la **variable asteroidSpeed** (en **Variables**) y la colocamos dentro del primer **0**: **set Asteroid velocity to vx (asteroidSpeed - 0) vy 0**. Pero así no hay velocidad aleatoria, para ello volvemos a **Math** y usamos **pick random 0 to 10** y lo colocamos en el segundo cero quedando el resultado así: **set Asteroid velocity to vx (asteroidSpeed - pick random 0 to 10) vy 0**.



Ahora le vamos a decir donde queremos que aparezca. Para que no se coloque en el mismo sitio (en medio de la pantalla), le vamos a decir al juego lo siguiente: **set Asteroid position to x 180 y (pick random 0 to 120)**. Con esto le decimos que se cree el asteroide en el punto **180** en **x** (este punto está fuera de la pantalla hacia la derecha) y en la **y** como hicimos con la **velocidad**, le damos un punto **aleatorio** entre **0 y 120** que es lo que mide la pantalla de alto.



MECÁNICA DE COLISIÓN CON ASTEROIDE

Vamos a empezar con las colisiones que tendrán los asteroides tanto con el láser que dispara la nave, como contra la propia nave.

```

on sprite of kind Projectile overlaps otherSprite of kind Enemy
  destroy otherSprite
  destroy sprite with fire effect for 100 ms
  change score by 1

on sprite of kind Player overlaps otherSprite of kind Enemy
  destroy otherSprite
  change life by -1
  
```

En Sprites cogemos el bloque de código ya conocido: `on sprite of kind Projectile overlaps otherSprite of kind Enemy`. Con esto le decimos al juego que cuando el Sprite de tipo Projectile (láser de la nave) toque otro Sprite de tipo Enemy (asteroide) hará lo siguiente:

Primero: `destroy otherSprite` (Asteroide)

Segundo: `destroy sprite with fire effect for 100 ms` (destruye el laser con efecto de fuego durante 100 milisegundos)

Por último: `change score by 1` (sumamos un punto)

```

on sprite of kind Projectile overlaps otherSprite of kind Enemy
  destroy mySprite
  destroy mySprite
  change score by 1

on sprite of kind Projectile overlaps otherSprite of kind Enemy
  destroy otherSprite
  destroy sprite with fire effect for 100 ms
  change score by 1
  
```

En Sprites cogemos de nuevo el bloque de código: `on sprite of kind Player overlaps otherSprite of kind Enemy`. Con esto le decimos al juego que, el Sprite de tipo Player (la nave) toque a otro Sprite de tipo Enemy (asteroide) hará lo siguiente:

Primero: `destroy otherSprite` (Asteroide)

```

on sprite of kind Player overlaps otherSprite of kind Enemy
  destroy otherSprite
  change life by -1
  
```



Segundo: **change life by -1** (resta una vida al jugador)

CREAR AMBIENTACIÓN

Vamos a darle una ambientación al juego, es decir, para que parezca que realmente estamos en el espacio y que la nave se desplace hacia la derecha.

```

on game update every 500 ms
  set star1 to sprite of kind Background
  set star1 position to x 160 y pick random 0 to 120
  set star1 vx (velocity x) to -100
  set star1 auto destroy ON

on game update every 200 ms
  set star2 to sprite of kind Background
  set star2 position to x 160 y pick random 0 to 120
  set star2 vx (velocity x) to -100
  set star2 auto destroy ON
  
```

Haremos que la primera forma de estrella salga cada medio segundo, así que, en **Game** cogeremos el bloque de código: **on game update every 500 ms**.

Colocamos los siguientes bloques de código:

Primero: **set star1 to Sprite "Introducir Sprite star1" of kind Background**. Creamos el **Sprite star1** y lo declaramos **Background**.

Segundo: **set star1 position to x 160 y (pick random 0 to 120)**. Como vimos antes para que salgan fuera de la pantalla y en posición aleatoria.

Tercero: **set star1 vx (velocity x) to -100**. Establecemos la velocidad.

Por último: **set star1 auto destroy on**. Con esto hacemos que la estrella se auto destruya cuando se salga de la pantalla.

```

on game update every 500 ms
  set star1 to sprite of kind Background
  set star1 position to x 160 y pick random 0 to 120
  set star1 vx (velocity x) to -100
  set star1 auto destroy ON
  
```



Haremos la segunda forma de estrella igual que la anterior, pero haciendo que salga cada 200 milisegundo, así que, en **Game** cogemos el bloque de código: **on game update every 200 ms**.

```

on game update every 200 ms
  set star2 to sprite of kind Background
  set star2 position to x 160 y pick random 0 to 120
  set star2 vx (velocity x) to -100
  set star2 auto destroy to ON
  
```

ÚLTIMOS DETALLES

Para que le demos el toque final. Añadiremos unos efectos de **sonido**.

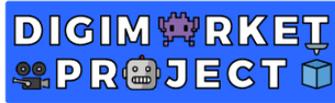
```

when green flag clicked
  when green flag clicked
  set position x to pick random 0 to 120
  set position y to pick random 0 to 120
  when green flag clicked
  set vx (velocity x) to -100
  set vx (velocity y) to -100
  when green flag clicked
  set vx (velocity x) to -100
  set vx (velocity y) to -100
  when green flag clicked
  set vx (velocity x) to -100
  set vx (velocity y) to -100
  when green flag clicked
  set vx (velocity x) to -100
  set vx (velocity y) to -100
  
```

INCLUSIÓN DE EFECTOS DE SONIDO Y DE CÁMARA

<p>Empezaremos por darle sonido a la nave, para ello colocamos en un Loops forever: play sound thump. Con esto simulamos el motor de la nave y suena siempre.</p>	
<p>A continuación, vamos a darle sonido al laser de la nave, para ello colocamos en la programación anterior que generábamos el láser incluiremos: play sound pew pew. Con esto simulamos el disparo de la nave y suena cada vez que se pulse el botón.</p>	
<p>Haremos que cada vez que se destruya un asteroide por el láser suene un pequeño sonido. Así que en la programación incluiremos: play sound small crash. Así, cada vez que el láser colisione con el asteroide y se destruya, se escuchará un pequeño crash.</p>	
<p>Por último, colocaremos un efecto de sonido y de cámara cuando el asteroide colisione con la nave. Para ello Music y play sound big crash. Y para que el efecto sea mucho mayor, también colocaremos: camera shake by 4 pixels for 500 ms. Esto hace que la cámara tiemble cuando un asteroide colisione con la nave. Se encuentra en Scene.</p>	

Con toda esta programación, hemos aprendido a realizar un juego de naves de desplazamiento lateral básico y sencillo. Si queréis realizar mejoras e incluirle detalles extra es vuestro momento. Aquí os dejamos una versión mejorada de una franquicia galáctica muy conocida y muy, muy lejana: https://makecode.com/_i814A5hDTYzm



Glosario

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

Ciclo de Vida: Duración de un elemento de un programa desde su creación hasta su destrucción.

Cámara: Objeto dentro de un escenario que sirve como vista del jugador respecto al juego.

Vida: Recurso que tiene el jugador para poder seguir en la misma partida. Una vez que terminen las vidas, la partida se acaba.

Efecto: Algo que se aplica sobre el escenario, objeto, personaje y más elementos para transmitir realismo o una sensación dentro del juego.

Aleatoriedad: La generación de números que tienen la misma probabilidad de generarse.

Memoria Ram: Componente que se almacena información a corto plazo ya que se utiliza para programas y datos que estén ejecutándose.