



## Tercer ejercicio Logic



### Descripción

En este ejercicio crearemos un videojuego estilo Duck Hunt, donde los elementos irán apareciendo de forma aleatoria en la pantalla y nuestro jugador tendrá que atraparlos con la pelota.

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

### Objetivos de programación y diseño

- Trabajar lógica mediante un control del juego en MakeCode Arcade.
- Trabajar y entender las variables en MakeCode Arcade.
- Asignar un ángulo en función al tiempo que pulse la tecla.

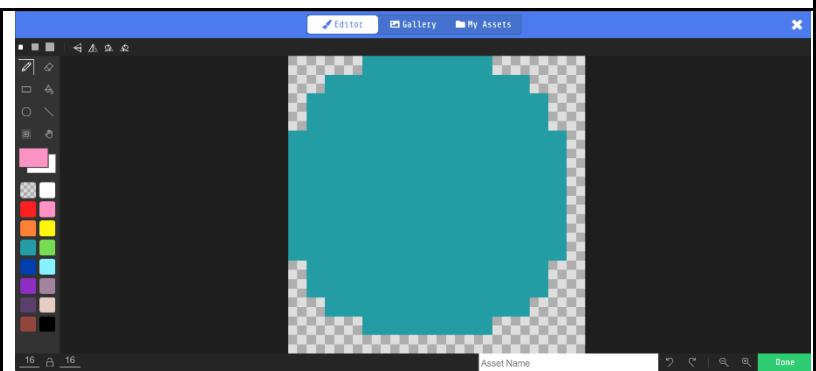


## Programación del juego

### CREACIÓN DE ASSETS

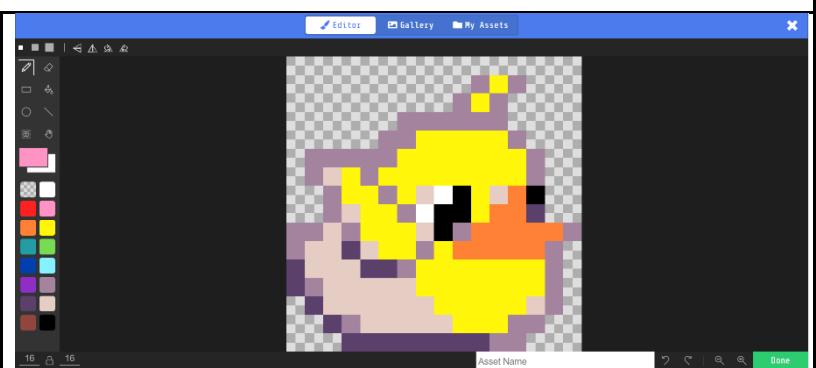
#### CREACIÓN SPRITE PLAYER

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



#### CREACIÓN SPRITES ADICIONAL

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





## PROGRAMACIÓN PRINCIPAL

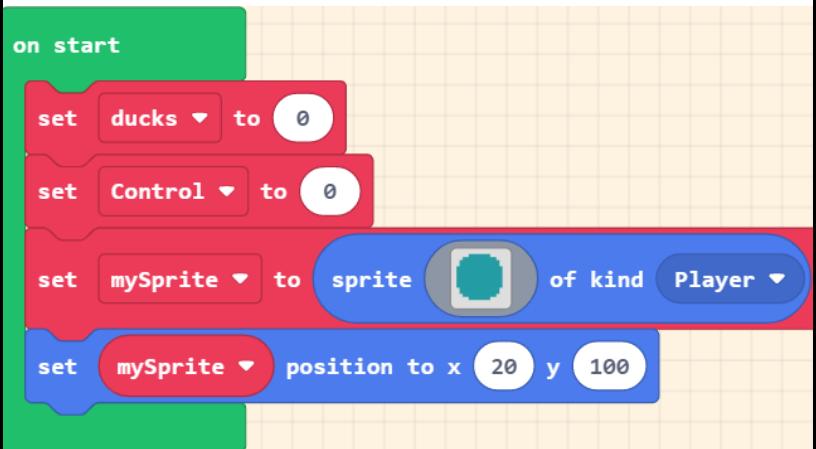
### CREACIÓN INICIO DEL JUEGO

En el “on start” colocaremos las **variables** (previamente creadas) **ducks** y **control** y les asignaremos el valor 0.

También incluiremos un **sprite** (player) y lo asignaremos como **Player** dándole la **posición** **x=20 e y=100**.

La **variable** **control**, nos indicará la **velocidad** con la que se lanza la bola.

Y la **variable** **ducks**, nos indicará el número de patitos que hay en pantalla.

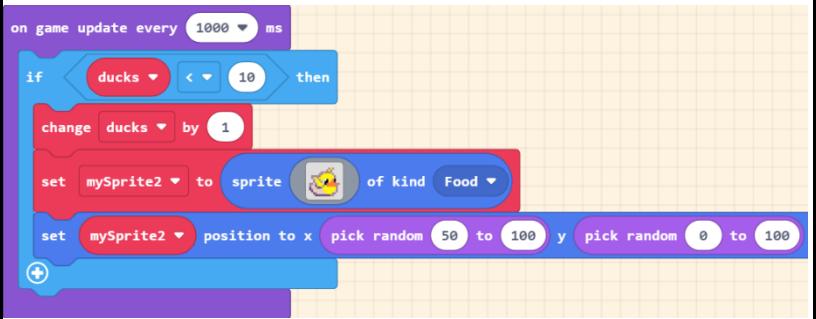


### CREACIÓN DE LA MECÁNICA APARICIÓN DE LOS PATOS

En un “on game update every” colocaremos un condicional “if”, cuya condición es si **ducks** es menor a 10, le introduciremos lo siguiente: **change ducks by 1.**

**set mySprite2 to sprite (sprite duck) of kind Food.**

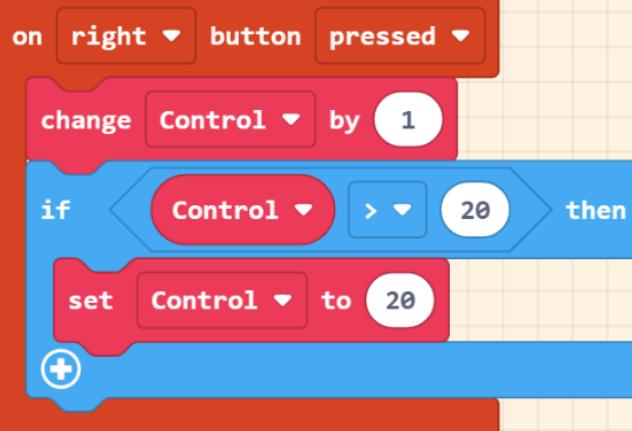
Por último, le decimos que **aleatoriamente** lo coloque.



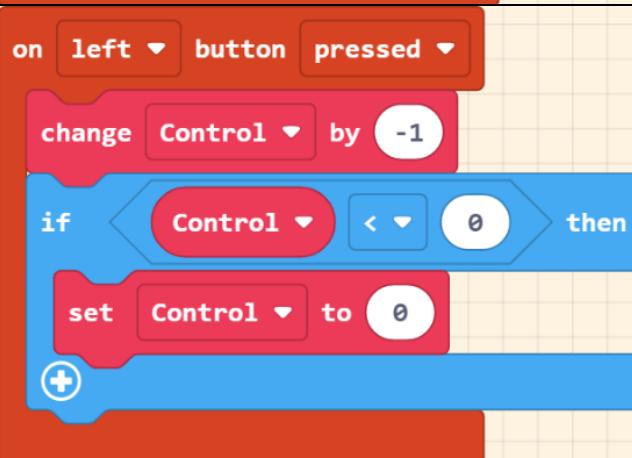


## CREACIÓN MECÁNICA LANZAMIENTO DE LA PELOTA DEL JUGADOR

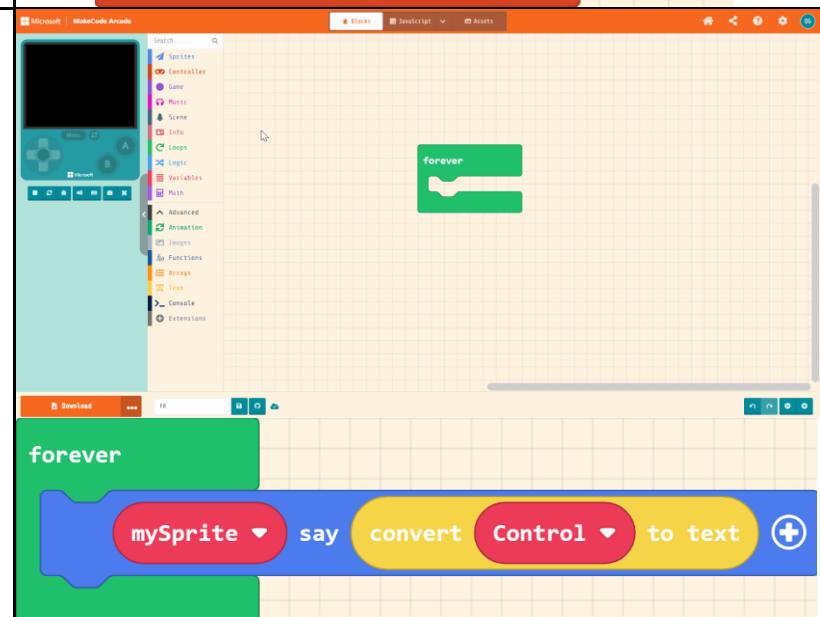
Ahora es momento de darle movimiento a la pelota para que pueda recoger o cazar los patitos que van saliendo, para ello, le decimos que, cada vez que pulse right (derecha) incrementa en 1 la variable control y que su máximo sea 20. Esto último lo logramos con un "if".



Al pulsar el botón izquierdo le disminuimos a la variable control 1 hasta llegar un máximo de 0.



Para saber cuánto incremento o decrecimiento de la variable control hay, usamos un forever y le introducimos los siguiente:  
**mySprite say convert Control to text**  
 Con esto convertimos los números de la variable en formato texto y nos lo muestra en pantalla.





Por último, al pulsar el botón A le damos a la pelota una aceleración en y de 100 y la velocidad en y que teníamos acumulado en la variable control.

Después esperamos 2 segundos y establecemos posición, aceleración y velocidad original.

```

on A button pressed
  set [mySprite v] [ay (acceleration y) to 100]
  set [mySprite v] [velocity to vx 50 vy Control x -10]
  pause (2000 ms)
  set [mySprite v] [position to x (20) y (100)]
  set [mySprite v] [ay (acceleration y) to 0]
  set [mySprite v] [velocity to vx 0 vy 0]
end

```

### CREACIÓN MECÁNICA OBTENCIÓN DE PUNTOS

Ahora, para que los patos puedan ser recogidos por la pelota, le daremos la orden que, al colisionar el sprite Player con uno de tipo Food se hará lo siguiente:

Restaremos 1 a la variable ducks.

Destruiremos ese sprite pato.

Y sumaremos 1 a la puntuación.

```

on [sprite of kind Player v] [overlaps [otherSprite of kind Food v]]
  change [ducks v] by (-1)
  destroy [otherSprite v]
  change [score v] by (1)
end

```

### CREACIÓN MECÁNICA FINAL DE LA PARTIDA

Para que la partida no se eterna y tenga un aliciente de mejora, en el on start, introduciremos un start countdown y lo pondremos en 60. Así cuando el tiempo se agote se verá la puntuación final y habrá que batirla.

```

on start
  set [ducks v] to (0)
  set [Control v] to (0)
  set [mySprite v] to [sprite of kind Player v]
  set [mySprite v] [position to x (20) y (100)]
  start [countdown (60) (s)]
end

```

Con esta programación, los patitos aparecerán en pantalla de manera aleatoria y nosotros con la pelota tendremos que recoger tantos como podamos en un minuto de tiempo. Hemos aprendido a incrementar el número de elementos en pantalla y también a darle valor a nuestro lanzamiento para que llegue más alto.

Ahora, es tu turno de personalizarlo y añadirle contenido. Aquí te dejamos el nuestro para que te inspires un poco: [https://makecode.com/\\_3kg7webwHats](https://makecode.com/_3kg7webwHats)

## Glosario

**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

**Operadores de comparación:** Son operadores que comparan un valor a otro y se introducen dentro de una condición.

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

**Evento:** Ejecuta una secuencia de instrucciones cuando ocurre un suceso externo al sistema.

Tipo de dato en programación:

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

**Física:** En videojuegos la física se refiere al comportamiento que tiene los distintos elementos dentro de un entorno. Suelen simular a las físicas del mundo real.

**Aceleración:** Es la variación de velocidad por unidad de tiempo.

**Velocidad:** Es una magnitud física que relaciona la posición con el incremento de tiempo.

**Vectores:** Es un segmento de recta dentro de un espacio representando una magnitud física o matemática.

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

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

**Puntuación:** Puntos totales que obtiene un jugador al realizar ciertas interacciones.

**Cuenta atrás:** Tiempo que se establece para que vaya corriendo y al terminar suceda algo, como terminar la partida.

**Game Over:** La partida se ha terminado. Se suele mostrar puntuaciones y te pregunta si quieres jugar otra partida.