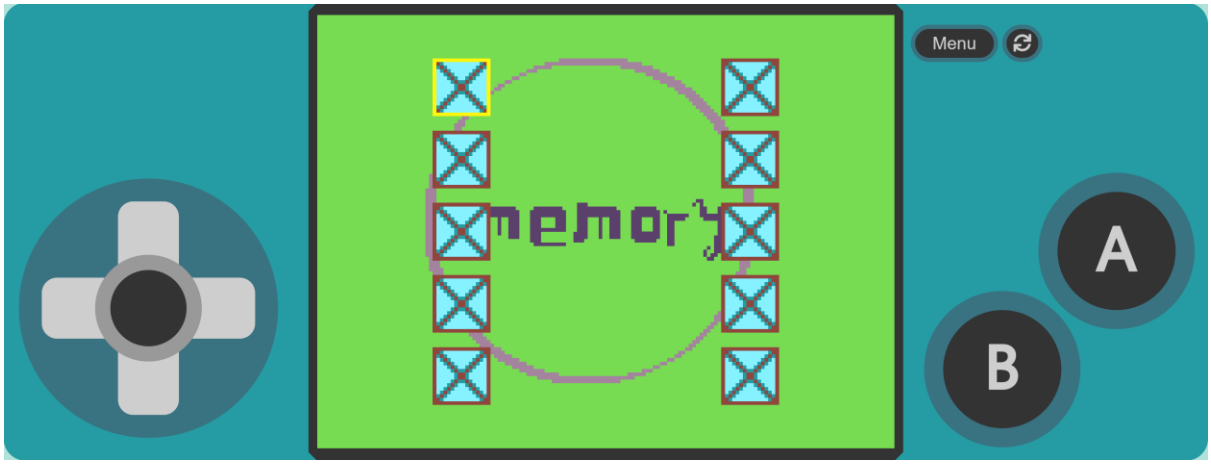
Tercer Ejercicio Arrays



Descripción.

En este proyecto crearemos una inspiración del juego clásico “Memory” donde el jugador tiene una mesa con cartas y tiene que escoger dos veces seguidas cartas con la misma imagen.

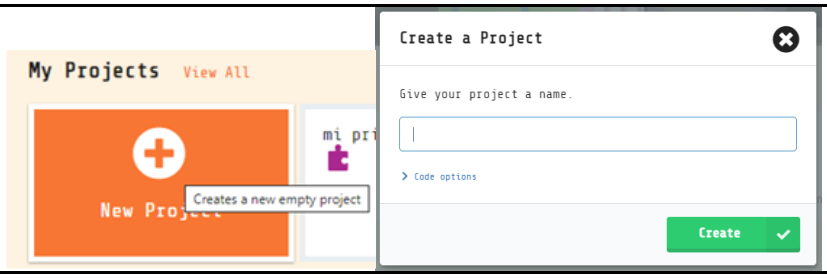
Se usará muchos de los elementos aprendidos aunque mayormente se trabaja el apartado “Array” donde se guardará en listas las posiciones de cada carta, las imágenes que están relacionadas, etc.

In order to do that, we use MakeCode Arcade to create the game.

Objetivos

- Crearemos un sprite que sirva como de cursor para navegar entre distintas opciones.
- Navegaremos entre array para conseguir el resultado requerido, tanto para poner contenido aleatorio como imágenes asociada a cierta posición.
- Programming the interaction between sprites.

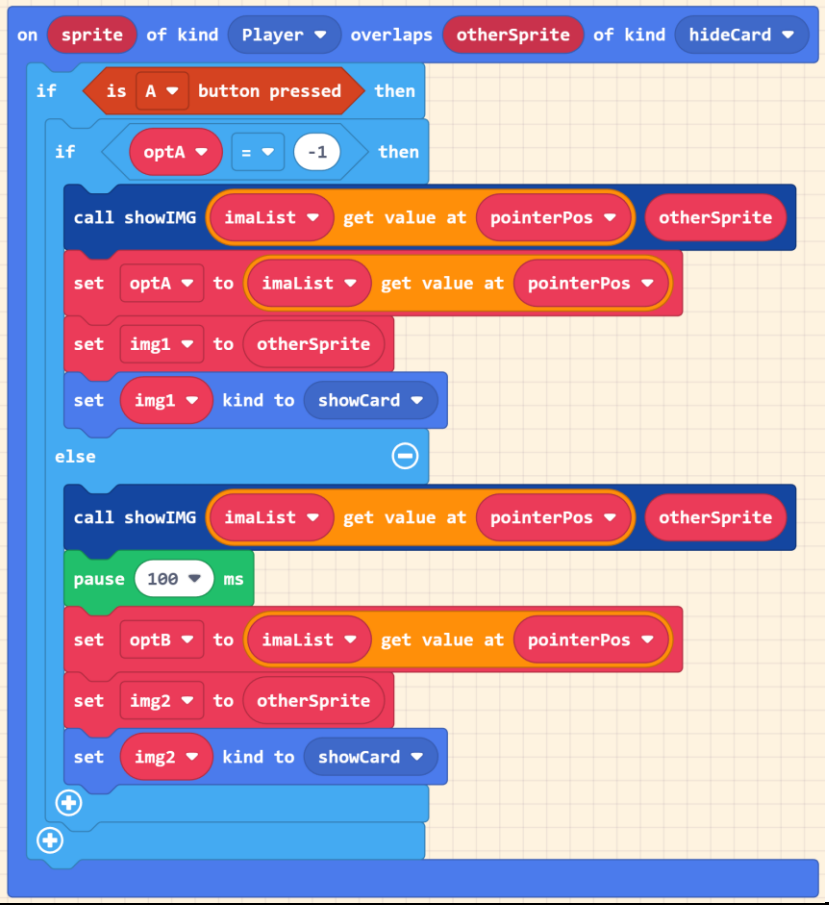
Guía de programación

NEW PROJECT	
We start creating a project, we should stablish the name, for example ¨Picking up food¨ and then press ¨create¨ button.	

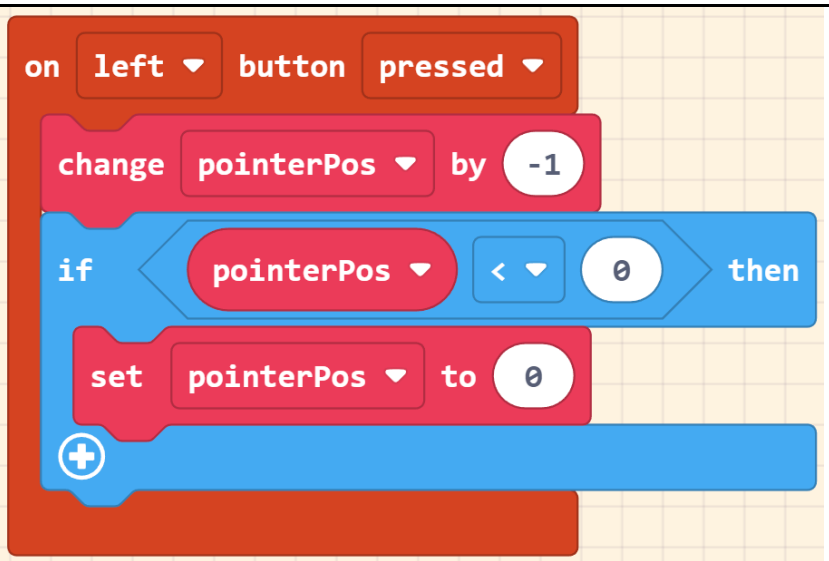
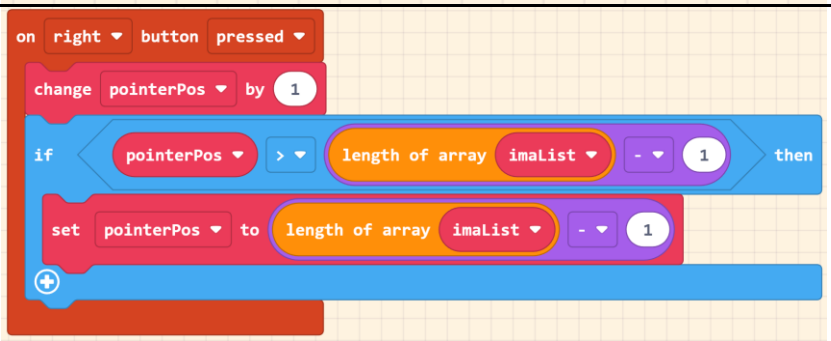
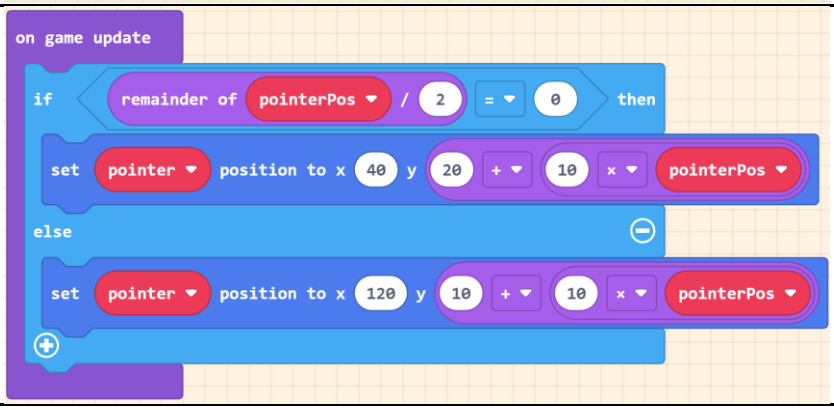
PROGRAMACIÓN PRINCIPAL	
ASIGNACIÓN IMAGENES EN EL ARRAY	
Crearemos la función “showIMG” con dos tipos de parámetros, uno numérico y otro de tipo Sprite. En función del número que entre desde el parámetro numérico, se asignará al Sprite que también entre desde el parámetro una imagen u otra. Se pone una condición por cada pareja de cartas para asignar una imagen.	

MECÁNICA DE ESCOGER UNA OPCIÓN

Programamos la interacción entre el jugador y la carta escondida. Aquí se comprobará si cuando se pulsa el botón “A” en función de si se había hecho una elección o las dos, se cambia la imagen de la carta escondida y la variable “optA” o “optB” cambie el valor en relación a la carta que realmente es.



MECÁNICA MOVIMIENTO CURSOR

Programaremos el botón izquierdo para que disminuya el valor de “pointerPos”. Pondremos una condición que no permita tener menos valor del mínimo, en este caso 0.	
El botón hacia la derecha será igual que el anterior grupo de bloques, pero incrementaremos el valor de “pointerPos” y el máximo será 9 que depende de la cantidad de cartas que usemos (10 cartas, 0-9 posiciones).	
Arrastraremos un bloque “on game update” al espacio de trabajo. Con esta instrucción cambiaremos la posición del Sprite “pointer”. Se hará una comprobación de si es número par el valor de “pointerPos”.	

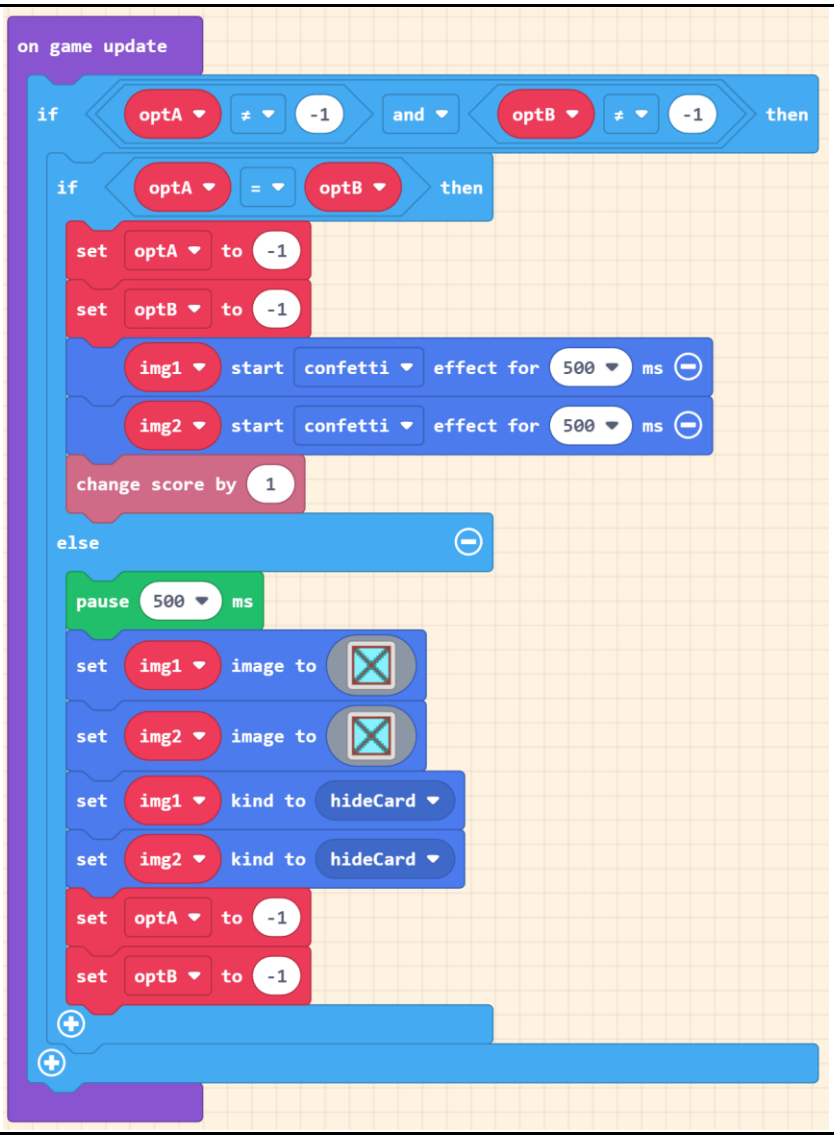
MECÁNICA DE RESPUESTA CORRECTA O INCORRECTA

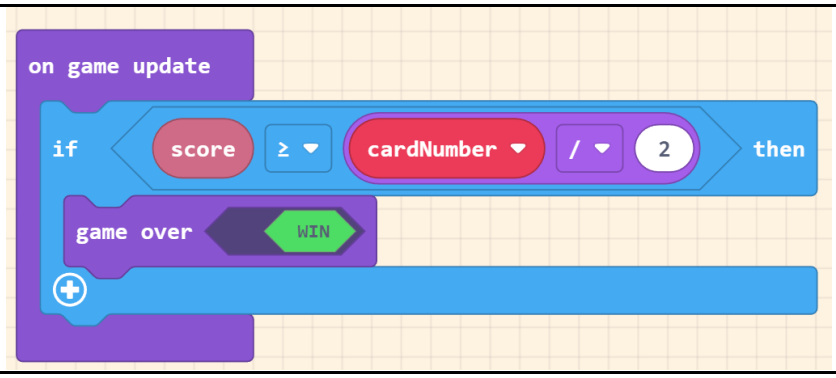
Arrastraremos otro bloque “on game update” para comprobar si la elección de las dos cartas es correcta o no.

Primero haremos una comprobación de si las variables que guardan la elección de las cartas son distintas a -1. Recuerda que hacemos esto porque ese valor es el que está asignado en caso de no haber ninguna elección.

Después de la primera comprobación, dentro de esta, comparará si las dos opciones escogidas son la misma. Si ocurre eso, se reinicia las opciones escogidas, aumentará en uno la puntuación y se mostrará un efecto de celebración.

En caso contrario las imágenes que estaban mostradas cambiarán y también devolveremos el tipo de cada sprite a “hideCard”: Además que se reinician las variables “optA” y “optB”.



MECÁNICA DE FIN DE PARTIDA	
Añadiremos un bloque más de “on game update” para terminar la partida. Pondremos una condición que mire si la puntuación es igual o superior a la cantidad de parejas de cartas. Si se cumple la partida terminará.	

Con esta programación podremos navegar el cursor entre las distintas cartas y podemos escogerlas para descubrirlas y ver que cartas son. En caso de acertar una pareja obtenemos un punto y cuando consigamos todas las parejas, ganamos la partida.