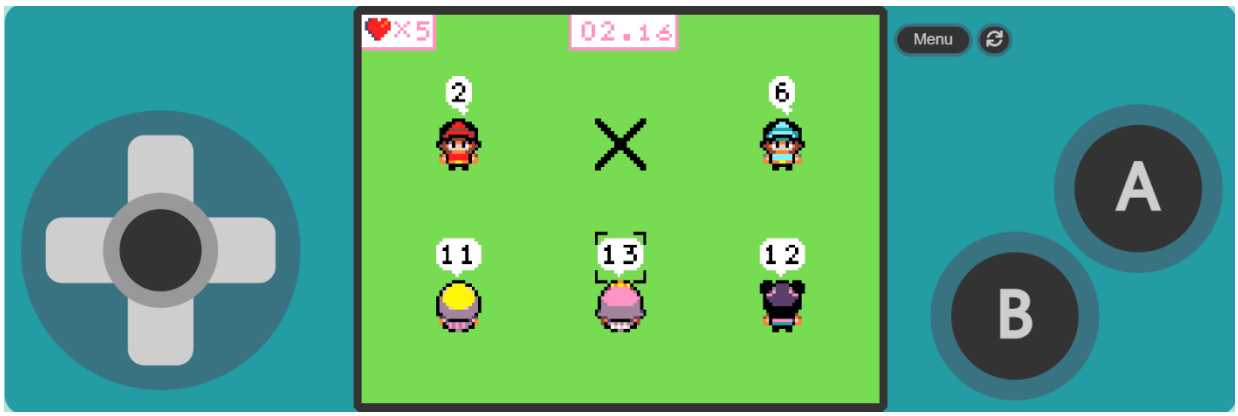
Segundo ejercicio Arrays



Descripción

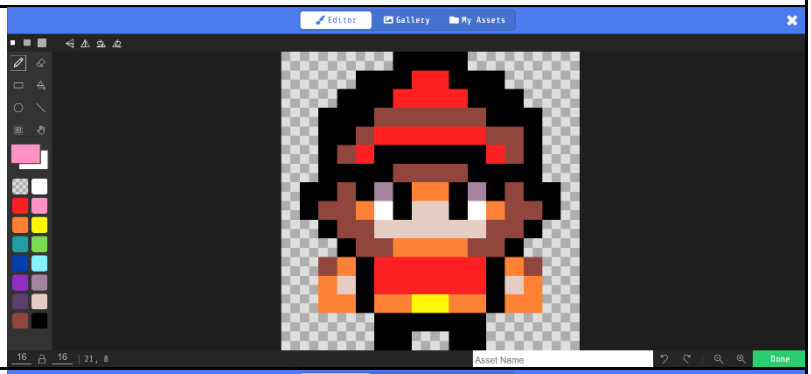
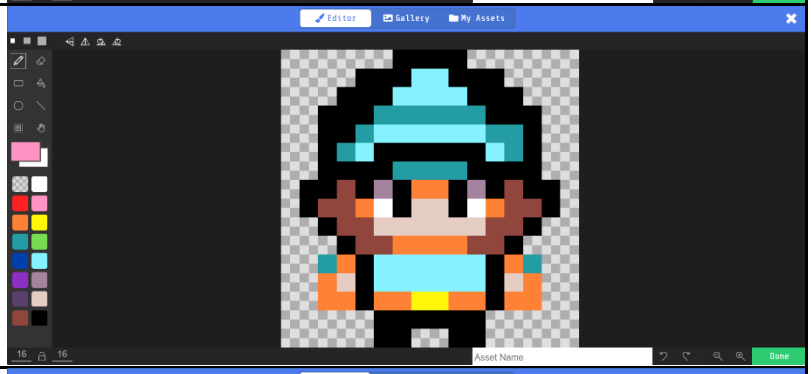
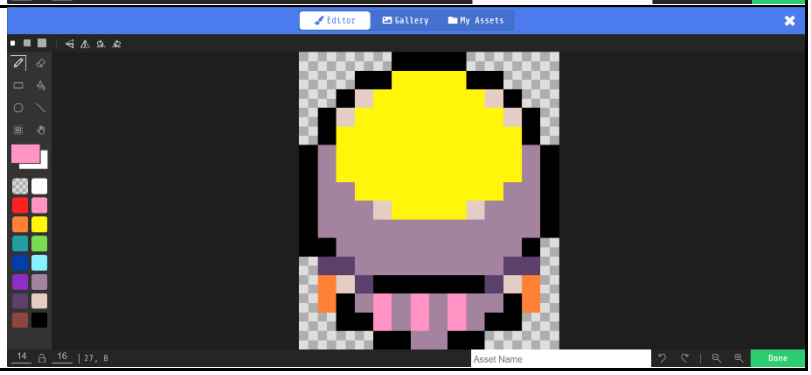
En este ejercicio crearemos un videojuego educativo de multiplicaciones.

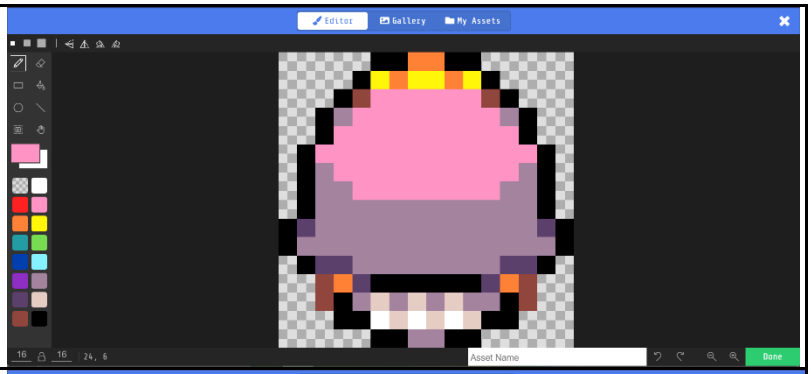
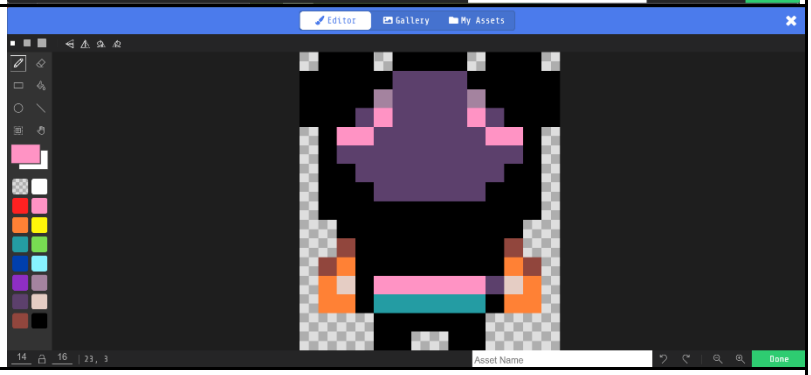
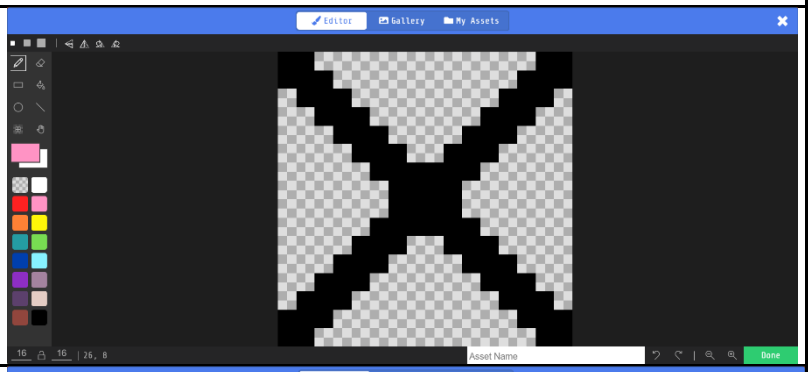
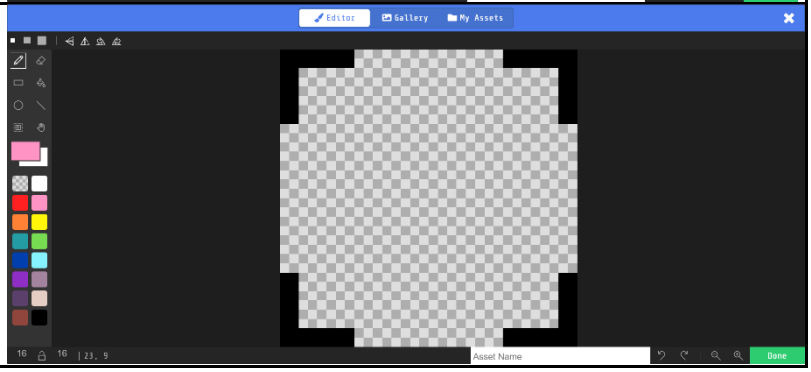
Para ello accederemos a **MakeCode Arcade** y realizaremos las operaciones necesarias.

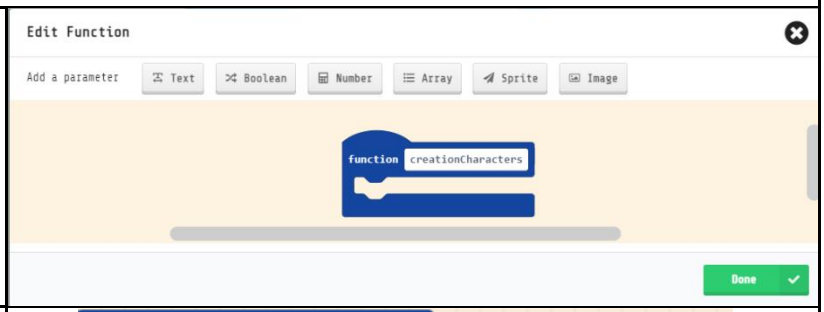
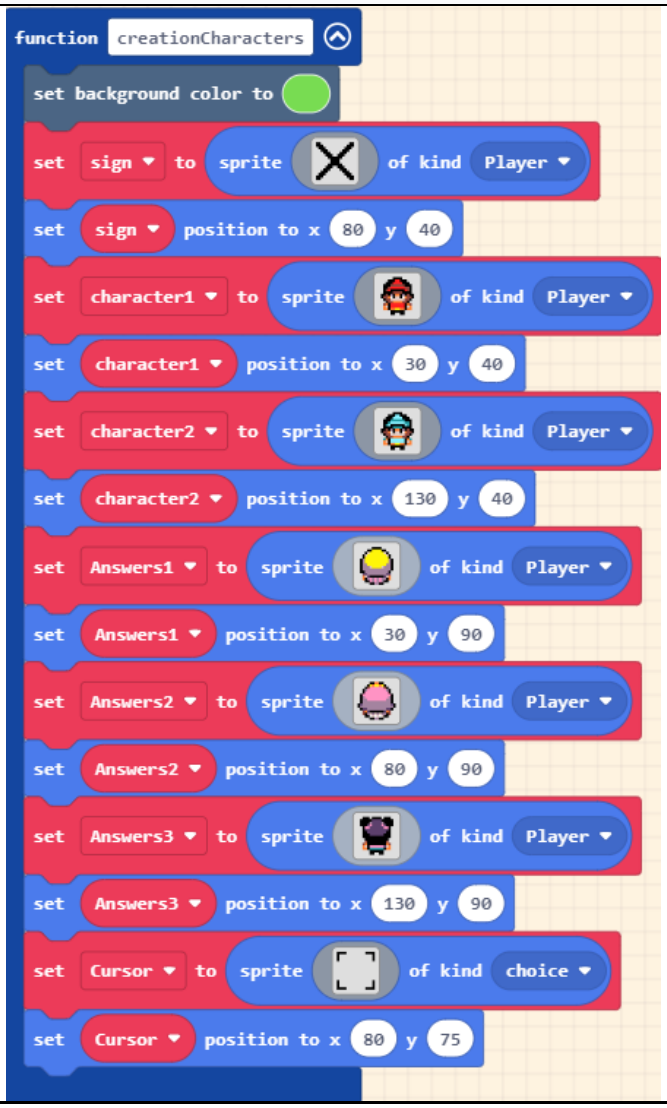
Objetivos de programación y diseño

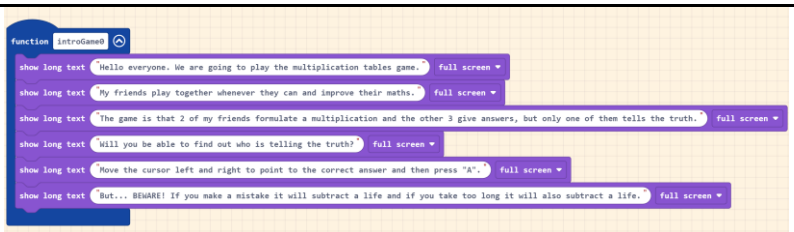
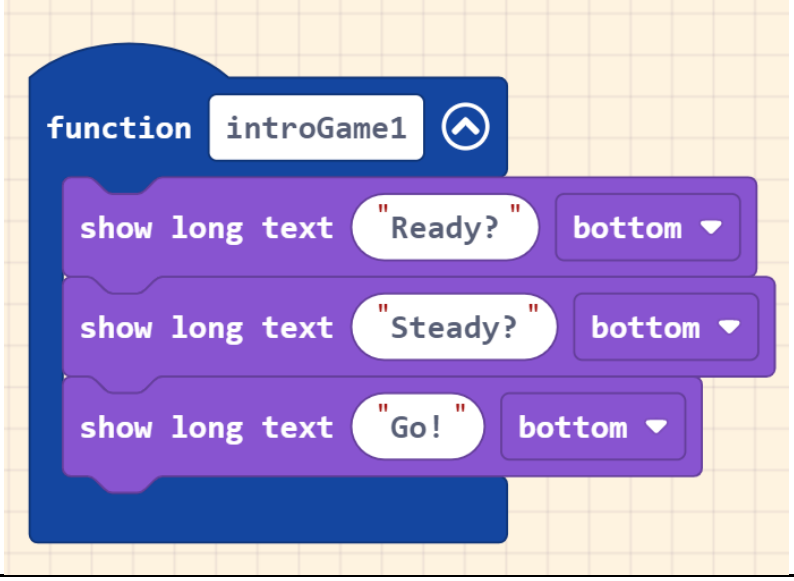
- Trabajar arrays mediante un control del juego en MakeCode Arcade.
- Trabajar y entender las variables en MakeCode Arcade.
- Asignar una posición a cada elemento del juego.
- Utilizar operaciones matemáticas para resolver problemas.
- Convertir valores numéricos en textos (string).
- Aumentar la dificultad.

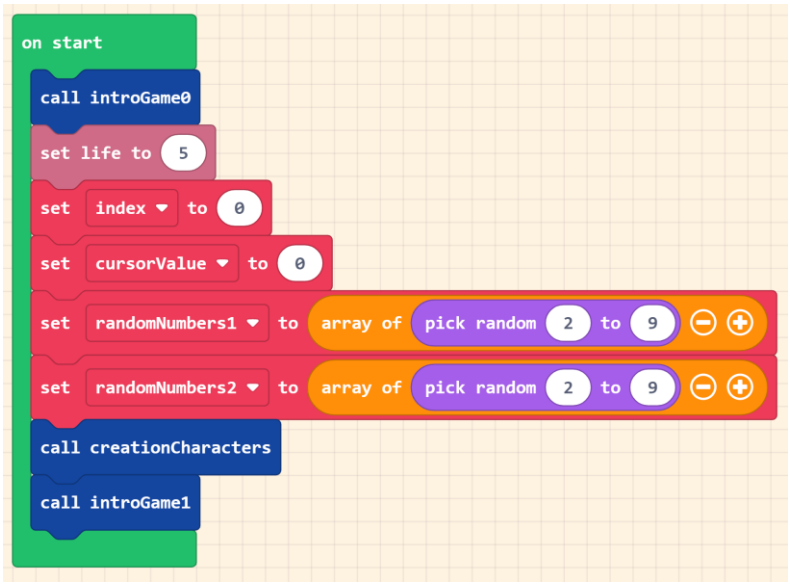
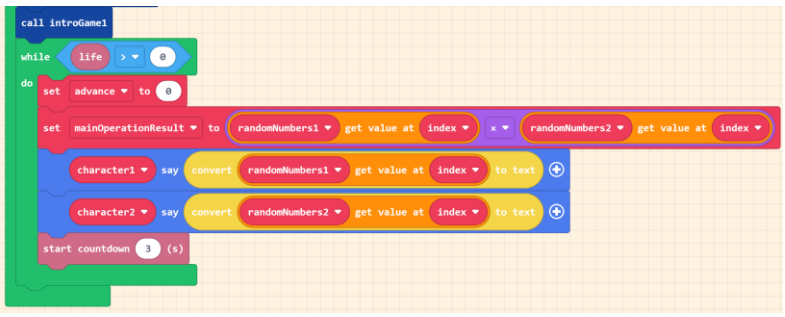
Programación del juego

CREACIÓN DE ASSETS	
CREACIÓN SPRITE PROTAGONISTAS	
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de character1.	
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de character2.	
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de Answers1.	

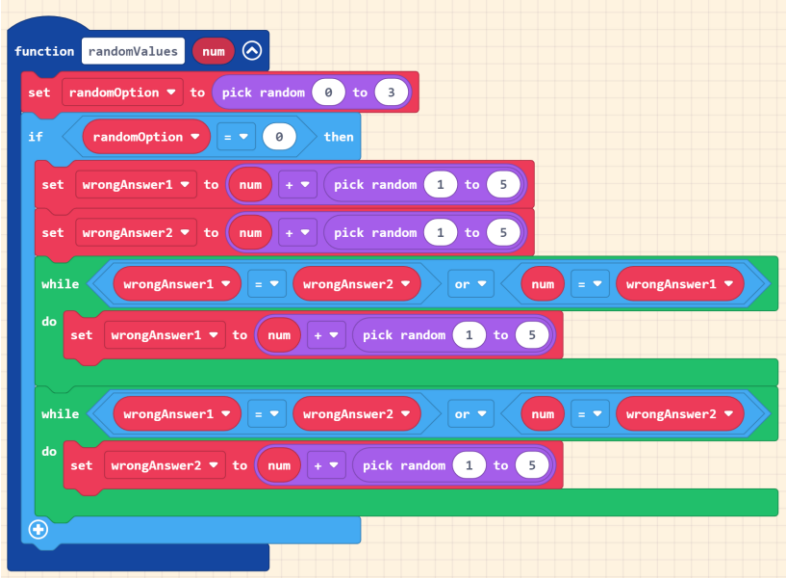
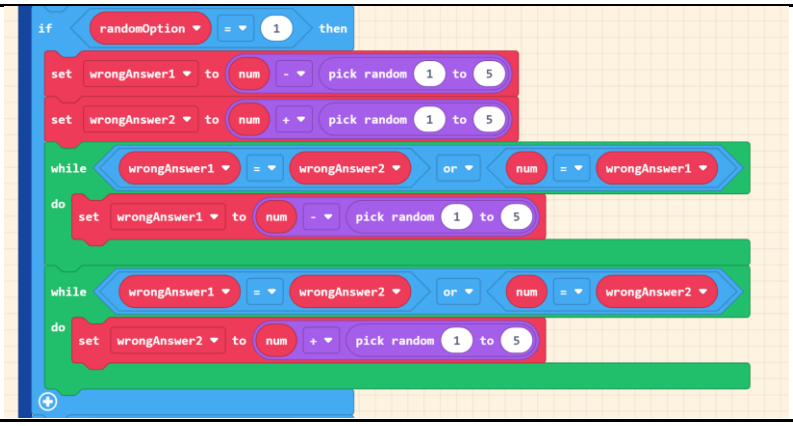
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de Answers2.	
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de Answers3.	
CREACIÓN SPRITES ADICIONALES	
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de sign.	
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de Cursor.	

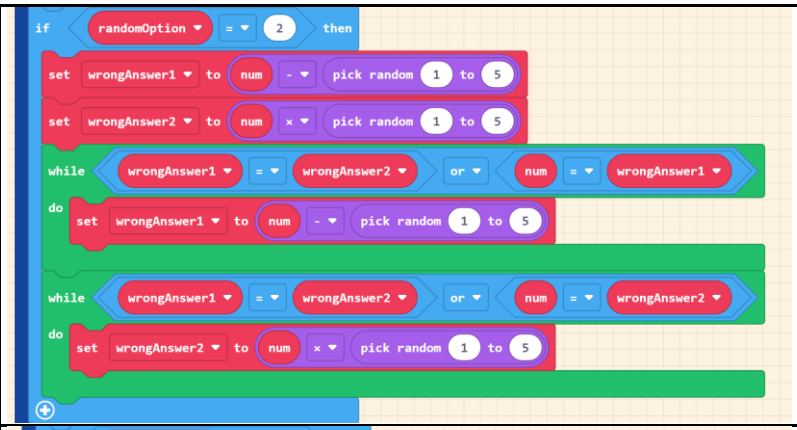
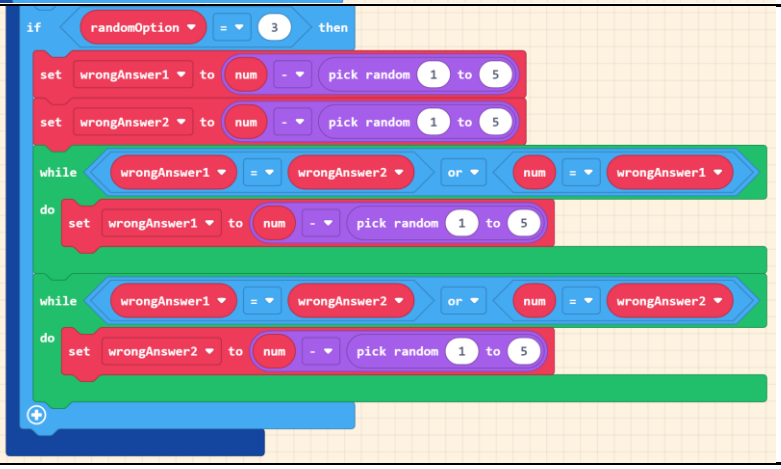
PROGRAMACIÓN PRINCIPAL	
CREACIÓN INICIO DEL JUEGO	
CREACIÓN FUNCIÓN creationCharacters	
Este juego empieza ya a complicarse a la hora de programar. Por eso tenemos que empezar por crear una función para crear el escenario y los elementos del juego.	
Aquí introduciremos que el color de nuestro escenario sea verde y también pondremos todos nuestros sprite de tipo player y los colocaremos en su posición en el escenario, por último, colocaremos el sprite cursor y lo asignaremos tipo choice.	

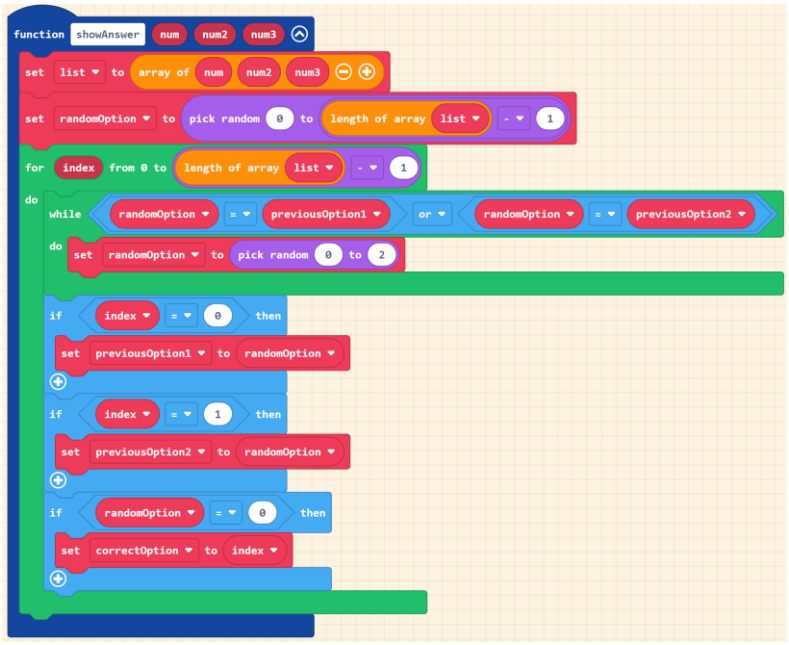
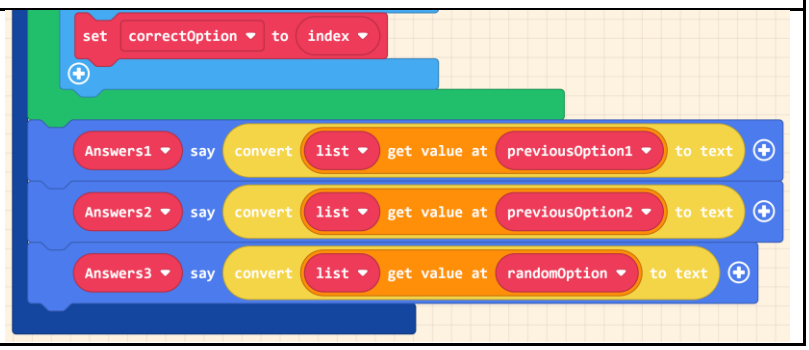
CREACIÓN FUNCIÓN introGAME0	
Empezamos mostrando mensaje en pantalla para que el jugador tenga información de cómo funciona el juego	
CREACIÓN FUNCIÓN introGAME1	
Mostramos más mensajes de inicio de la partida	

PROGRAMACIÓN PRESENTACIÓN DE ELEMENTOS Y DECLARACIÓN VARIABLES INICIALES	
En el on start empezamos a establecer variables y activamos las funciones para mostrar una presentación del juego. En ambos introGame muestran mensajes al jugador y creationCharacters crea y coloca todos los sprites que tendrá el juego. La variable index servirá para mirar en ciertas posiciones en los arrays más adelante. La variable cursorValue servirá para tener un control sobre el cursor. Las variables randomNumbers tendrán un valor aleatorio que serán las que se muestre al jugador y adivine el resultado de multiplicar ambas variables	
PROGRAMAMOS BUCLE DE REINICIO DE PANTALLA PARA CADA INTENTO	
Hacemos un bucle que se ejecute mientras el jugador tiene alguna vida. Dentro estableceremos ciertas variables, mostraremos un mensaje de la operación actual usando personajes e iniciamos una cuenta atrás. La variable advance servirá para comprobar si hemos seleccionado algún resultado. La variable mainOperationResult tendrá de valor el resultado de la operación que debe acertar el jugador.	

PROGRAMACIÓN FUNCIÓN RANDOMVALUES

Crearemos la función randomValues con un parámetro numérico, esta sirve para generar las variaciones en las respuestas, que tendrán cierta aleatoriedad. Estableceremos un valor aleatorio en randomOption. La cantidad de números posibles que tendrá esta variable servirán para establecer pautas en los distintos resultados incorrectos. Empezando por randomOption igual a 0, hacemos que las variables wrongAnswer tengan el valor introducido en el parámetro de la función más una suma entre 1 y 5 de manera aleatoria. Después usaremos bucles while para comprobar que no hay números iguales en las distintas opciones a escoger por el jugador.	
Podremos duplicar el conjunto de bloques recogidos en “randomOption = 0” y cambiar valores para adecuarlo a otro posible resultado. En este caso se cambia wrongAnswer1 con una resta	

Lo haremos de nuevo, pero ahora, respecto al anterior grupo de bloques, pondremos una multiplicación para wrongAnswer2	
En la última variación de posibles resultados incorrectos pondremos en ambas una resta	

PROGRAMACIÓN FUNCIÓN SHOWANSWER	
Crearemos una función con 3 parámetros numéricos llamado shorAnswer. En los parámetros se introducirán los resultados a escoger por el jugador. Dentro de esta función se irá ubicando las distintas posiciones el resultado correcta, rellenado los otros espacios con los incorrectos. Crearemos un array con los distintos valores introducidos en los parámetros. Se creará una variable que tenga un valor aleatorio entre 0 y el tamaño del array list restando uno, en este caso, esa operación da de resultado 2. Usaremos un bucle para que recorra todos los espacios del array list. Se asignará en las variables las posiciones que se mostrarán más adelantes	
Para terminar con la función, programaremos a ciertos personajes para que digan las opciones que podrá escoger el jugador	

ACTIVAR FUNCIONES RANDOMVALUES Y SHOWANSWER	
Llamaremos a las funciones “randomValues” y “showAnswer”. En “randomValues” introducimos la variables “mainOperationResult” para que las opciones alternativas tengan relación a esta. En “showAnswer” introduciremos en los parámetros las variables “mainOperationResult”, “wrongAnswer1” y “wrongAnswer2” donde cambiarán la posición de las opciones que podrá escoger el jugador	

Con esta programación, se crearán los personajes que servirán como elementos gráficos para mostrar información, se establecerán los valores y posición de las distintas opciones a elegir de manera aleatoria.