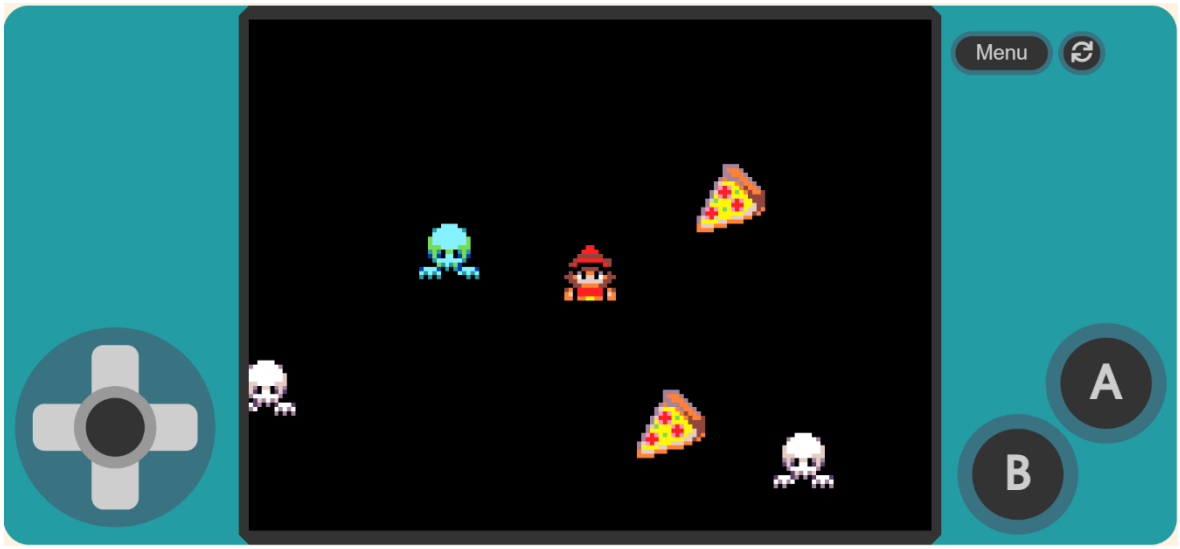
FIRST FUNCTIONS TASK



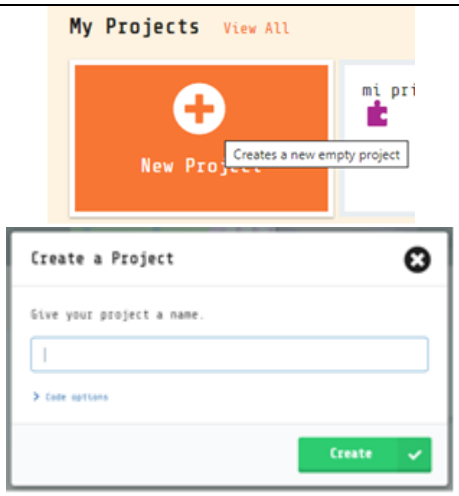
Description

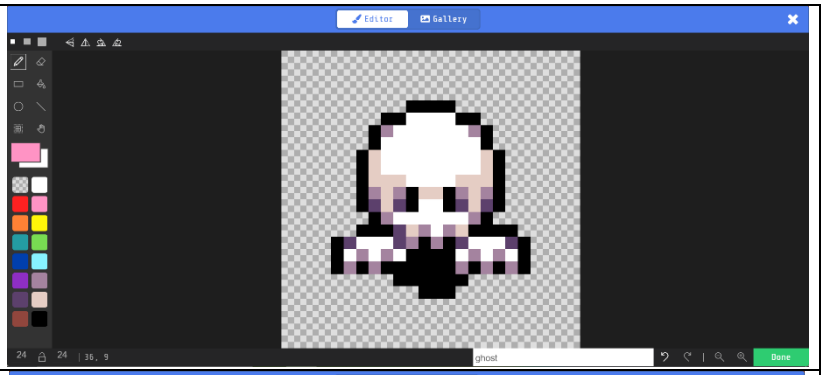
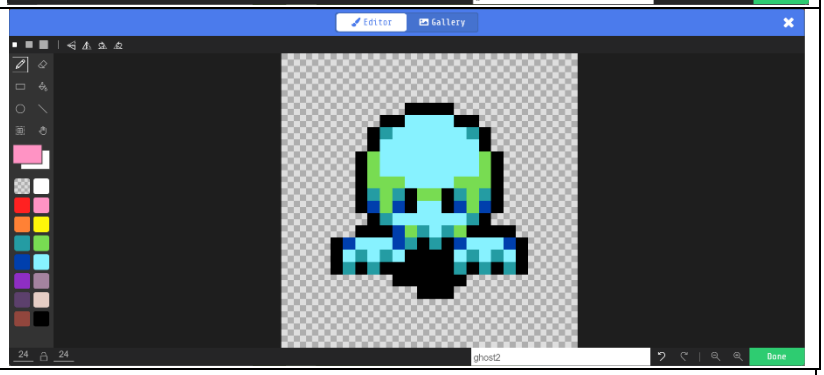
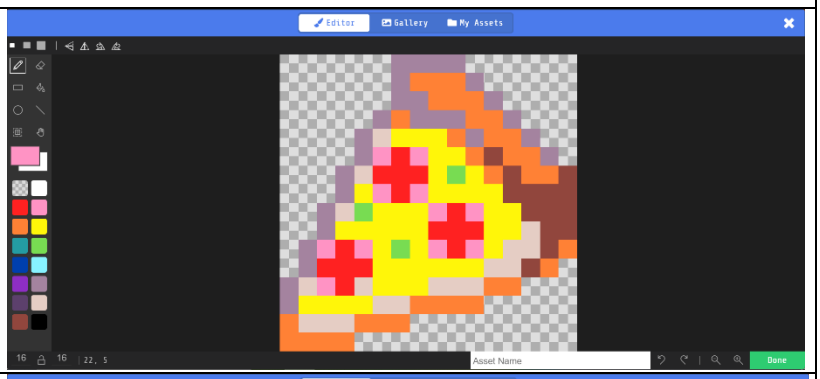
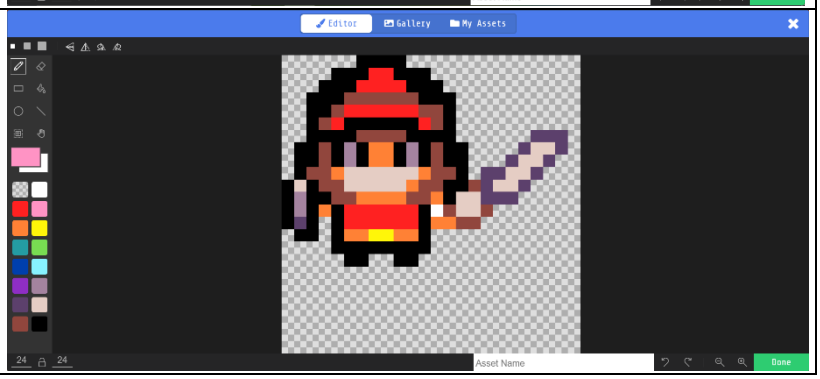
In this project we will work on a game that uses different programming and video gaming concepts, such as the use of sprites, the interaction between the different game elements or the use of functions.

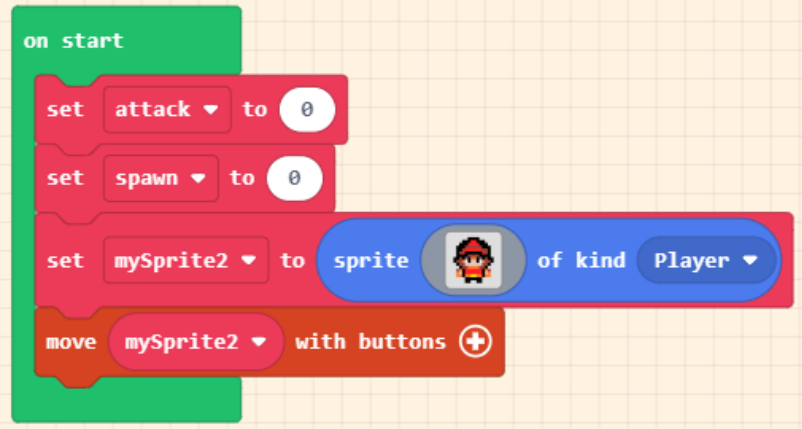
Goals

- Work on the sprites movement.
- Program interactions between the elements.
- Establish a score the changes depending on the situation.
- Use functions to program different events.
- Program the entry in scene of the enemies, their speed and acceleration.

Programming guide

NEW PROJECT	
We start creating a project, we should stablish the name, for example ¨ghost hunt¨ and then press ¨create¨ button.	
Here is the link with part of the programming and assets done. https://makecode.com/_7KvXtMcvX3Ej	
ASSETS CREATION	
PLAYER SPRITE CREATION	
We recommend using a 16x16 px grid for the Sprite of mySprite2	
ENEMIES SPRITE CREATION	

We recommend using a 24x24 px grid for the Sprite of ghost.	
We recommend using a 24x24 px matrix for the Sprite of ghost2.	
ADITIONAL SPRITES CREATION	
We recommend using a 16x16 px matrix for the Sprite of block2.	
We recommend using a 24x24 px matrix for the Sprite of block3	

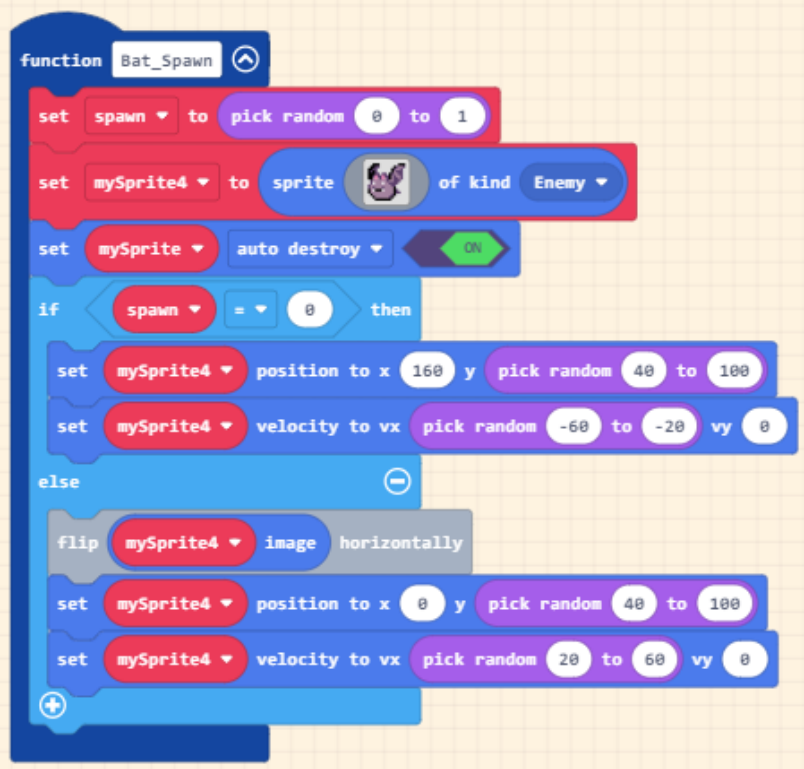
MAIN PROGRAMMING	
ON START GAME CREATION	
We will start establishing the variables “attack” and “spawn”, which we will use later. Also, we will create our character and we are going to give it movement by using the crosshead.	
ATTACK ANIMATION	
We will program that when pressing the “A” button, our character defends itself by using the variable “attack” and two different images of the sprite to simulate a hit.	

ENEMY SPAWN

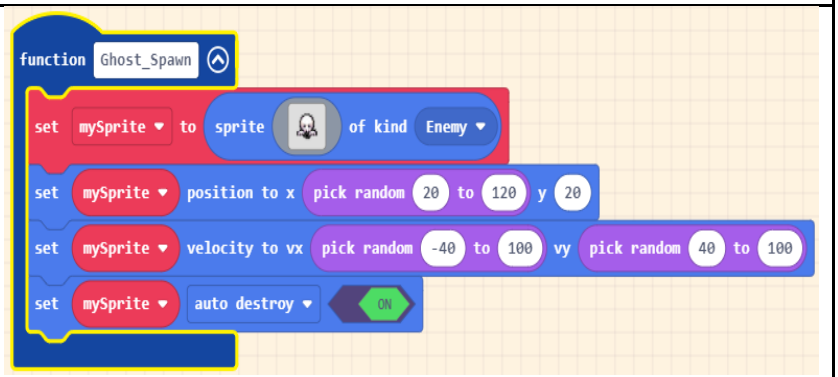
Now, we are going to create one of our enemies.

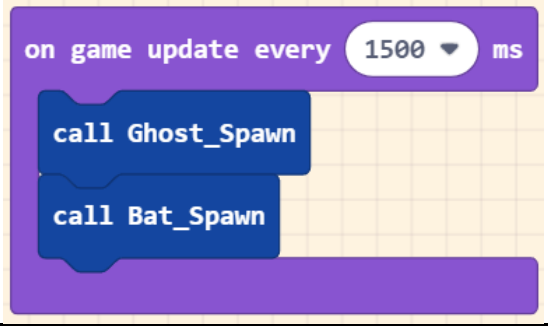
For this we will use a function. Inside it we will create the enemy sprite, establishing that it destroys itself when coming out of scene. Also, we will establish that the variable "spawn" chooses a random number between 0 and 1.

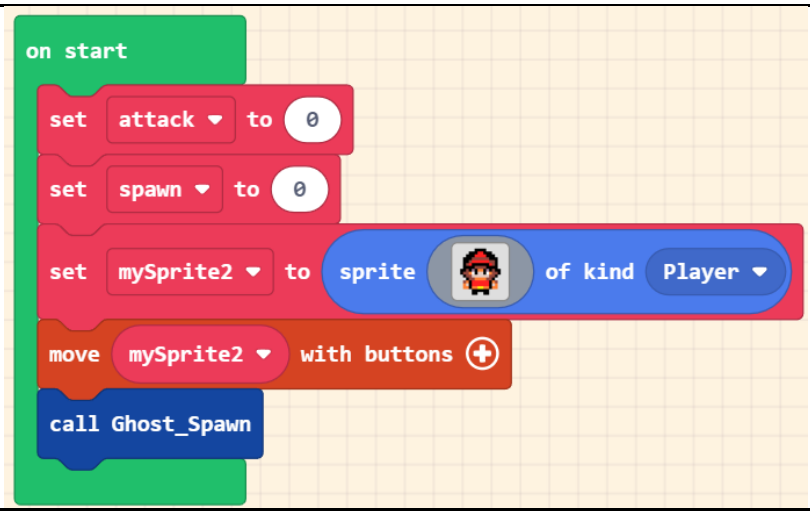
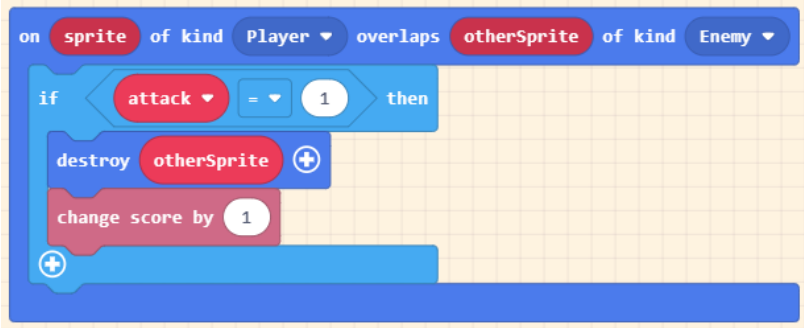
By using this "if else" and the blocks inside it, we will establish that the enemies will come from right or left side of the map, depending on whether the variable is 0 or 1.



We will use another function to create our second type of enemy. Inside it, we will create the sprite and establish it a speed and a random position to make it unpredictable.
When the enemy come out the scene, it will destroy itself.



FOOD SPAWN	
With this “game update” we are going to make these food type sprites appear in a random position every second.	
ENEMY SPAWN TIMINGS	
In this other “game update”, we will call the functions previously created to apply them every 1.5 seconds, creating the enemies.	
PICKING UP FOOD MECHANIC	
With the “overlaps” we will establish that if our character touches a sprite of kind food, this last one will disappear, and we will increase our score by one. Also, we create the code below the overlaps block to increase the difficult of the game. Inside it we will call the function, what will make a ghost appear every time we touch a food sprite.	

To make it a little bit more difficult, we will call this function of the ghost in the "on start", making two ghost appear instantly as soon as the game begins.	
ATTACK INTERACTION	
With this programming we will be able to defend ourselves from the enemies. When the variable "attack" is 1, when pressing "A", we will be able to destroy every enemy sprite that we touch and we will get a point for each one we destroy.	
When our attack is cooling-down, the variable will be equal to 0. If in this moment an enemy impact us, we will lose the game.	

With this programming, we will use functions and variables to help our sprite of a kind "Player" to get sprites of a kind "Food" and to destroy sprites of kind "Enemy" getting like this as much points as possible. If we impact with and enemy when our variable makes us vulnerable, the game will be over.

Glossary

Variables: Variables are spaces associated with an identifier, where a value can be stored and modified.

Functions: Functions are subprograms that contain a set of instructions and can be executed by making a call to them from the main program.

Event: An event triggers a sequence of instructions when a specific external occurrence happens.

If-Else: If-Else is a conditional statement that executes one sequence of instructions if a condition is true and another sequence if the condition is false.

Acceleration: Acceleration refers to the rate of change of velocity over time.

Velocity: Velocity is a physical quantity that relates position to the rate of change of time.

Lifecycle: The lifespan of an element in a program from its creation to its destruction.

Randomness: The generation of numbers with equal probability of occurrence.

Score: The total points a player earns by performing certain interactions.

Game Over: Game Over is the end of a game, often displaying scores and asking if the player wants to play again.