





# FOURTH MOTION TASK



### Description.

In this project we are going to work with different programming concepts, we can find functions with some elements, text, scene designing...

#### Goals.

- Work with sprites movement.
- Programming interactions between sprites.
- Set a score that changes depending on the situation.
- Understand the functions and call them appropriately.







### Game Programming.

Г

NEW PROJECT	
We start creating a project, we should stablish the name, for example "Ping Pong" and then press "create" button.	My Projects View All mi pri New Project Create a Project Give your project a name. 1 > Ceet estima Create
ASSETS CREATION	
Sprite Player Creation	
We create one sprite with 16 x 16 p dimensions. We create in "Assets" ar then we look for a sprite that we like "Gallery". After that we click on "Don	x nd in e









We will create the "ball\_creation" function, here we design the "ball" and we turn on the "bounce on wall" so it bounces on the edge of the map in the opposite direction. Then, we will set the speed in both axes' "x" and "y", it will be 100, we will also have to set a random position in "y" axe between 0 - 120.



#### PADDLES CREATION

The next step is the formation of "left\_paddle\_creation" function and we add inside the "left\_paddle" function left\_paddle\_creation  $\odot$ sprite (player 1), we will set the of kind left\_paddle • left\_paddle 🔻 movement and 150 speed only in y to sprite axe using "move with buttons". We left\_paddle 💌 with buttons vx 🔞 vy 150 👄 do not want the paddle to exit the screen, so we set "stay in screen". left\_paddle 💌 stay in screen After that, we give the paddle a left\_paddle 💌 left 🔻 to 12 set position in the screen, in this case it will be 12. function right\_paddle\_creation  $\odot$ This block is the same as the of kind right\_paddle • right\_paddle 🔻 to sprite previous one, but with set "right paddle creation" name right\_paddle 💌 with buttons vx 🙆 100 🔾 (player 2). However, we will also set right\_paddle 🔻 stay in screen the initial position to 148. right\_paddle 💌 right 🔻 to 148 **POINTS MECHANIC** 







Here, we will add a "forever" block so the programming inside repeats over and over, then, we will drag inside the loop two different conditions, so they both can occur. We want to get that situation, if the ball touches one of player's wall it will bounce but the player will lose 1 point in the score, so we will add "change player score" and give –1 value.



#### **INTERACTION MECHANIC**

With this "overlaps" block we are going to get that, if the ball touches the paddle, it has to bounce and change the direction but with same speed, so we add "set sprite vx velocity to sprite" and we set -1 so it changes its way. After that we set "change player score by" and add 1 so the score increases one point.

This block is the same as the previous one but using player 2 instead of player 1.











With that programming, each player will control a paddle and they will try to bounce the ball on it towards the player's wall to get points, if it reaches the wall the player loses 1 point. When 30 seconds pass by, the programming will check the score and the payer with higher score will win or they will tie.







## Glossary

Loops: A sequence of code that is executed repeatedly.

Example: Forever, while, for.

**Conditionals**: A sequence of instructions that are executed based on the value of a condition.

Example: If, If...Else.

**If...Else if**: A sequence of conditionals in which we move, in an orderly manner, from one condition to another until one of them is met.

**Functions**: A subprogram that collects a set of instructions and can be executed from the main program by calling it.

Scene: The space where the video game takes place.

Walls: Objects or areas where the various elements of the game cannot pass through.

**Score**: The total points obtained by a player through certain interactions.

**Countdown**: A set time that runs down, and when it reaches zero, something happens, such as the end of the game.

**Game Over**: The game session has ended. It usually displays scores and asks if you want to play another game.

Music: Combination of sounds and silences that create a rhythm.