Super Tusa Fox is a platform game similar to the known “Super Mario Bros”, in which the main objective is to take the maximum number of diamonds in the time (180 seconds). I made the videogame with the program Unity Hub.

To enter the game, you should start by playing the scene “menu”.When you start the game, it appears a first image of the game in which you have to click on “Start” to access the game. I have done this by creating a new scene called “menu” and then I added an image to the background and some text to finish the name of the game. The text “Start” is a button so by clicking on it you access wherever you want, in this case to the game.

Just when you click on it, it appears the game and the song “Tusa” start. This is possible because I created an empty GameObject called “SOUND” and inside of it, I put an Audio Score and add on it the song.

The scene of the principal game is called “my_game”. I started by joining some platforms and creating big ones according to the size I wanted. For making it more entertained I also put some moving platforms that can move either right-left, up-down or even I have some with diagonal movement. This is possible by moving the target each platform contains. Apart of that, there are also three platforms that when they detect the player on it, wait for some seconds and then falls down, and with the delay they have programmed they appear in the same position after the delay time had finished.

Controlling the player is very easy:

- Pressing on the right arrow, goes to the right.
- Pressing on the left arrow, goes to the left.
- Pressing the up arrow, it jumps.
- Pressing the space, the game is pause until you press again the space.

When no arrow is pressed, it is on an idle state in which it makes a small movement on the same position.

If you try to jump from one platform to another and you fall down to nowhere, I have programmed it to start again from the beginning of the scene, but you do not lose the diamonds you have.

<u>SCENE my_game</u>

This scene is the one with the game and consists of some GameObjects like:

- Main camera: I use it to select a solid color for the background, this color was chosen because of the colors of the background. Also, it contains a script called “follow_player” used for following the player with the camera. The camera has two important variables:
    - Min camera position: is the position in which the camera starts, and the most left position it can go.
    - Max camera position: is the position in which the camera ends, and the rightest position it can go.
- Ground: is the platform in which the game starts.

- Player: it contains a Rigidbody 2D and a Circle Collider 2D in order to control the collisions the player can have with the platforms and the diamonds. In addition to that, it also has a script called “Player_controller” in which I made the implementations of the functions to obtain the movement and velocity of the player. Inside of “player” there is an object called “ground collider” with the script “CheckGround” in which I coded the collisions between the player and the platform and ground, and when he has to detect if it is in the ground or it is not. That was possible because each thing has a tag according to what I decide them to be, and they have different grades of friction.

- Platforms in general: here I have grouped all the platforms I made: the 1x1, 3x1, and also the moving and falling platforms.

- Diamonds: is what you have to collect. They have a polygon collider, because as it was not a normal circle, was easier to do the shape with this one. They also have a script called “purple_controller” in which they have a function for the diamond´s movement and to destroy and count the object when the player collides with it.

- Canvas_counter: is a canvas object with an image (a diamond) and a text with the “score_value” script used to print in the screen the new score after collecting a diamond.

- Canvas_pause: is a canvas object with a text containing the “pause_script” used to detect when the space is pressed and to stop the game when it is, and restart the game when it is pressed again. It is formed by:

    - pause_menu: is the menu that appears in the screen when you pressed the space. It contains a script called “pause_menu” with a function used for quitting the game, a button for “Options” in which where you click on it the options_menu is active and the pause_menu is not active, and a button for “Quit” to finish the game when you click on it. As finishing the game is impossible, when you click on it nothing happens but, on the console, appears a message saying that you have quit the game.

    - options_menu: it has a text and a slider for the volume but it is just to simulate a normal game, it is not programmed to do so, and also a back button to return to the pause_menu.

- Canvas_timer_text: is just a text formed by “0000” used to replace the time.

- Canvas_image: is the timer image and text. The image is an animation of the same time of the time programmed to the game (180 seconds) and “timer_image” has a script to control the counter and to restart the game after the time. I decide it to be 180 seconds but you can change it if you want in “timer_image” with the variable “Count_time”.