Component 3 - Map Interaction

This is a simple component which showcases three map additions that the player can interact with - a sticky moving platform, a pressure plate and a button. It also adds the option of having the player ‘die’ when they fall off the map.

How it works
The player movement in this script is only used as a tool to showcase how the map works, and more information on this can be found in my first component - player movement.

Platform:
Within start the function movement is called. This function causes the platform to continuously move to the right, and then when the platform collides with another object with the tag “Floor” the direction changes to left.
There are then many mechanisms at play to ensure that the player moves with the platform rather than it moving independently. Firstly a reference is created to the game object that is the player and this is set to null. Within the on trigger stay function an if statement checks is the collision was with an object tagged “Player”. If this is the case the player variable is declared as the object which has collided, making it no longer null. Within a function that references the separate player movement script and is running constantly within start, an if statement checks if the player variable is not null. If this is the case the position of the player is set to the position of the platform plus an offset. The offset is defined within the on trigger stay function as the player’s position minus the position of the platform. Then in the on trigger exit function the player status is set back to null so that when the player leaves the platform they are no longer being forced to move with the platform.

Pressure plate:
Within the on collision stay function is being called, an if statement checks if the collision was with an object with the tag “Player”. If this is the case a bool called “Player on” (which is originally set to false) becomes true, and the moveable wall moves up. When the player exits the pressure plate player on then becomes false. Within update an if statement checks if both player on is false and the y coordinate of the moveable wall is below a certain y value, and if this is true the wall moves back up.

Button:
Within update a function called position check runs constantly. This function contains an if statement which checks if the player is between two x coordinates, and if this is true a bool called player in position becomes true. If this is not the case an else statement makes the bool false. Also within update, an if statement checks whether the player is both in position, pressing the E key, and if the moveable wall is below a certain y coordinate. If all of these conditions are met, the moveable wall moves upwards.

Fall death:
Within the player movement script a function is constantly running called fall. This function contains an if statement which checks if a player's position is below a certain y value. If it is below the player is then transformed to the position of an empty object named respawn.

How to modify

- Each script and object reference can of course be applied to any in game object.
- The speed at which the sticky platform moves can be changed by simply putting a new value in either the unity inspector or at the top of the script.
- All of the tags used can be assigned different names in both the code and unity.
- The speed of the moveable walls in both the pressure plate and button script can be changed easily in both the scripts and the unity inspector.
- The moveable walls can also move in a different direction by changing the vector2 values given in the scripts.
- The button can have a smaller or larger radius that allows the player to press it by simply changing the y coordinates in the if statement.
- The key to interact with the button can be changed - simply give a different key in the script.
- The Y value at which the player dies and is sent to the respawn point can be changed to anything within the player movement script.

How I implemented into my game

To implement this component into my final prototype I simply duplicated the project to form the basis of my prototype. I then moved each component into different positions and the map interaction was implemented.