Skateboard Project

The current document has the objective to inform about the process behind the Skateboard Project, the main systems and finally what could be improved.

General Structure

In this project, it was necessary to make adjustments every time not just in the parameters values of the player but in the animations and its states. For this reason, the project has methods for each behavior and values that can be editable from the instance of the class.
The structure of the project can be simplified in this four items:

- **SkateboardCharacter:** The skateboard character is where all the logic of the movement coexist because it gets the input of the player and calls the methods related to them.
- **MovementComponent:** This is a class that has the values of every behavior the player has, from the speed to the deceleration.
- **Main HUD:** This is a class where the score is calculated and then shown on the screen of the player.
- **Obstacles:** The main purpose of this class is to be instanced in the map so the player can jump over it and get more points.

Systems

The main focus of this section is to describe the existing systems in the project and the logic behind.

- **Movement:** For this project, there are 2 types of movements. The forward movement and the horizontal movement.
  Forward movement is a state triggered when the player impulses himself. This impulse directly affects the velocity of the player, turning into an increase of the max speed. The player’s maximum speed is limited by a parameter.
  Horizontal movement is a state where the player rotates depending on the input (left or right). Horizontal movement is actually a direction change due to the behavior of skateboarding. Player is able to move horizontally just when is moving forward and the impulse animation has already finished.
- **Deceleration:** Player is always decelerating, even when is impulsing himself. The current deceleration rate depends on the current movement. There are three deceleration rates: Impulse Deceleration,Stop Deceleration and Horizontal Deceleration. Impulse Deceleration rate is used when the player is moving forward but not trying to decelerate. On the other hand, Stop Deceleration is used when the player is trying to decelerate (opposite input of moving forward) and Horizontal Deceleration is the deceleration rate when the player is changing his direction. In the case the player is moving horizontally and wants to decelerate, the final deceleration rate is the sum of Stop Deceleration and Horizontal Deceleration.
- **Jump:** Player is able to jump just when is moving and the impulse animation is not playing.
- **Score:** Player can get points to be added to the current score when he jumps. The more time in the air, the more points he gets. Player gets extra points for every obstacle it

jumps over. The extra points gained for jumping over an obstacle depends on the height of it.

- **ResetRotation:** When the player gets stuck, he can use the Restore Rotation button to change the rotation to the current view rotation.

Complements

This section is for additional implementations indirectly related to the main systems.

- **Main Menu**: Simple main menu creation.
- **Main HUD:** Visualizable UI to show the score and the current player speed.
- **Control Rig:** Real time logic for changing transform root position.
- **Horizontal Rotation:** Tthe player has an inclination depending on the horizontal move (left or right)

Personal assessment

For this project I focus on the movement mainly, setting a really simple UI with not too much code for it. It was difficult to replicate the movement, primarily because of the animations and fixes that were needed to implement a proper behavior. With this project I get more feedback about what I should improve and I learn a lot of new things and put in practice some others that I already know. I have a long path to go through but I know I’m the way.

Future work

The movement can be improved even more because there are some cases when it feels weird, and makes it more realistic. Also implements a proper UI.