Explanation of the system:

For the development of the simulator, I used a simple yet robust approach that should allow for project scalability if desired. The main systems are as follows:

- Skateboard Character

- Obstacle

- Controller

Skateboard Character: Contains movement, camera, and scoring functionalities. All its internal components are created using C++ and later configured in a child class in Blueprints. This should allow for different variants of the Skateboard Character to be generated without altering any functionality. The initial configuration variables are available for editing from the class in Blueprints.

- **Movement:** Input is detected through Input Actions that are directly connected to their corresponding function. Movement itself is handled through the Character Movement Component.

- **Score:** Private variable that contains accessibility methods: Get and Set.

Obstacle: Contains elements that allow for collision detection to award points to the player when required.

Controller: Manages the interface elements available for the simulator.

Personal Assessment of Performance:

- **Strengths:** I believe I correctly reflected the mechanics requested in the task. The time taken for each component of the task was optimal, and during testing, I did not detect any bugs that affected the correct functioning of the project.

- **Weaknesses:** I would have liked to achieve a level of independence for the skateboard component from the character. This would allow the player to experiment with skateboards of different configurations, providing a varied gameplay experience.

Times:

- **Skateboard Character:** 6 Hours

- **Obstacle, UI, and Controller:** 2 Hours

- **Environment and setup:** 4 Hours

- **Document:** 30 Hours

-**Total Time:** 12 Hours and 30 minutes Approximately.