Assets: 4h

The character and animations for the character were downloaded from Mixamo. However, they needed some fixes that were corrected in Blender, particularly the addition of a root bone and keeping the character in place. Animations were also slightly adjusted to better fit the project.
Skate and some ramps were downloaded from Sketchfab and CGtracer, while other ramps were made in Blender.
The city is from a pack in the Epic marketplace.

Level Design: 2h

Some assets were removed from the demo level of the city, and ramps were added trying to follow the road directions.
On the center of the level, multiple ramps were added to create a more compact playing space.
Connecting the start and the center of the map there is a tunnel made with simple cylinders

Animation: 3h

The animation contains the main state and locomotion state machines, similar to the normal third person.
On the locomotion state machine, there is the logic to move forward (thrust) a “first thrust” state was added to cut the first part of the animation and so make it faster. It also implements the crouch state that can be actioned at any time
On the main state machine, the locomotion is the default and the rest is part of the jump/falling logic

Thrust: 2h

The thrust input changes the state of the animation to make it move forward. The animation has a notify that will call an event in the pawn to add force forward

Breaking: 1h

Breaking changes the “braking deceleration flying” to a bigger number, making the breaking look natural, an animation is missing for this state, which doesn’t reflect visually the effect

Rotation: 4h

When pressing the rotate input, the velocity is rotated by a value, and in the tick, the character will rotate to match the velocity.
If the character is falling or if the velocity is almost zero, the opposite happens and the character is rotated and when it becomes relevant again (landing) the velocity is rotated to match the forward vector.
A first iteration was created, where all the rotation was handled by the character and the velocity tried to match the rotation. Although it would cause problems when falling facing up.
Currently, the character only has a forward vector X+ although the skate should be able to move forward and backward (X+ and X-). A check every time the character lands would need to be done to invert the character. This was not done due to time limit and the need to adjust animations and other systems

Physics: 4h

The main solution to achieve the best result possible was to change the movement mode to flying and add gravity to each tick. The walking mode had too many restrictions that were limiting the character. Falling in check by the default pawn, although can cause some problems when getting stuck in gaps. Also, the default pawn was used, limiting to one capsule, an extra capsule on the skateboard could have helped the simulations.

Ground Normals: 3h

The character does two line traces, one for each skate truck location to the ground. Then calculates the average normal of both. The skateboard rotates following that vector.
The character uses IK (inverse kinematics) in the animation to adjust the feet to the board. This check is not performed while falling.

Point system: 2h

The system for points is based on the air time, character rotation while in the air, and if a gap was crossed.
Gaps are checked using a BP and hand placing in the map

Conclusion:

Using the third person character was useful to make the project in 2 days although it brings quite some limitations.
If I would make this project to achieve the maximum potential would need to create a custom pawn and custom movement modes to better express the skating movement.

Time-wise, it took me a lot of time fighting against the walking movement mode to finally understand that the flying mode is better.
In general, I am satisfied witch what I was able to achieve in such a short time.