Thought Process

Originally the first idea I had was to implement some kind of minigame like a time trial to find a jump all the obstacles on the level or an implementation like Brain Wall but with the skate trying to make a particular jump to surpass the walls in an infinite way until the character lose. But because of the issues I had the first day that made any progress, I focused all my efforts on trying to do a minimal viable product that had the required features.

System Decisions and Implementations:

Code:

The code was totally implemented using Blueprints. I know that this type of code is less performant but because the goal of this task was to make a Prototype in a short amount of time the simplest and more simple iterative capability Blueprints provide made him the best option.

Character:

The Skater character Blueprint is a modification of the Third Person character blueprint provided by Unreal Engine. On this blueprint I took some of the ideas and implementations from the Youtuber CodeLikeMe and adapted them to my situation. Originally the idea was to implement the moving capabilities the project currently have with the adittion of the possibility that inclinations applied force, like ramp for example, but because of the lack of time I couldn't add.

Jumpable Obstacles

What I did for these elements was to make a blueprint actor that had a collision around to detect the player character once he enters the obstacle area. This element registers a reference to the player and starts making a box ray casting projecting above him so it can be sure that the player passes the element. In case it does, the element will call the GameMode function to increase the score.

Performance

Personally I think my performance took a big hit because of my lack of knowledge related to publishing UnrealEngine projects using the free Github account, that make me have almost totally waste the first day with troubleshooting related with the maximum files sizes this accepts to upload changes or the low capability free account have for using LFS. Besides that I think I could do something decent enough in 24 hours considering that I didn't use any pre-written or re-used code, if we don't consider any guide or youtube video I use to complement my implementations/ideas, but isn't something that had enough pride to make a presentation to sell it with a engaging pitch that why I didn't send the record.

Amount of hours spend

1. Character 10-12 hours approx: this hours consider the day I worked fighting with the repository. Also I consider here searching the character implementing the animations and the logic implementation for the movement of the character.
2. Jumpable objects 1-2 Hours approx: This includes the logic of the jumpable elements, the game mode logic for the scoring and the implementation of the UI to show the Scoring update.
3. Checking the different assets and fighting with GitHub 7-9 hours approx: This includes checking the resources provided by the task and dealing with the troubleshootings related with trying to upload these assets.
4. Building, documentations and presentation 3 hours: This includes making sure the executable could be made and work properly making this document and preparing and recording the presentation.