NGPTest

This skateboarding prototype contains three systems: the movement system, the obstacles system and the points component. All this are mostly independent from each other but use delegates to communicate when needed, providing a soft and extendable approach to class communication.

Movement System (7h)

This is an adaptation of the UE5 Character Movement Component, it uses movement input as the main way to push/stop the character, this approach was chosen given the strict time limitation but I feel it ended up not being ideal and requiring a few too many "tricks" to make it feel ok. Perhaps implementing a custom movement mode or even looking into the new MoverComponent would have been better options, but given the time constraint I chose to see it through and move on to the other features.

Pushing (1h)

Pushing was implemented using an animation notify state in coordination with the push animation montage to ensure the push is applied when it makes sense for the given animation, this approach worked quite well and is very versatile and future proof. Ideally the notify state would have more options to customize how the push is applied, like a FloatCurve asset or a blueprint implementable event to be able to customize how we calculate the push force based on time, speed, combo size, etc.

Obstacles System(3h)

The obstacles system is comprised of a single component, the ObstacleBox component. This component can be added to any static mesh and tracks weather or not the player hits the static mesh while in the obstacle box. when the player exits the obstacle this result is broadcast together with some stats about how the player traversed the obstacle.

Points System (2h)

The points component works by listening to obstacle broadcasts and processing their results accordingly. It includes a combo system where if you chain consecutive jumps withing a time frame your next jump will have a combo bonus.

UI (3h)

UI is coordinated by using the output broadcasts from the points system know update combo score and total score accordingly and avoided using function bindings as this run every frame to recalculate the widget texts and it's unnecessary. For the progress bar we used a function binding given that this value will change almost every frame

Assessment (Total: 16h)

In general, even though the spec is met, I feel like there are definitely some areas of improvement specially with the movement system and game feel. A lot of time was lost trying to shoehorn the existing Character Movement Component to fit the requirements, perhaps a different approach might have yielded better results in less time, but custom movement is not an area I'm too familiar with so I paid the price.
In general I like most of the code output and think it's clean and decently organized, there are some system design decisions that I would change to make the system more extendable but they were put aside (though they are hinted) given the no blueprints requirement and the time constraint.