

Alejandro Muñoz's technical test documentation

Summary

This is a very rudimentary implementation of a third-person shooter in which to save some time implementing animation and aiming system, guns just hover in front of the character's face and shoot in the direction the characters are looking at. Projectiles damage all characters, including team members (friendly fire). When a character dies, it respawns in a very basic way: its health is restored and it is moved to a random player start location. It only has one type of hero (projectile gun hero) and thus there is no hero selection screen, but the architecture has been created with multiple heroes in mind, so adding a new one build be a fairly simple task. The game has no HUD, as I decided to not implement it to save time

Notes

- Custom classes related to this test project are prefixed by 'UnrealTest'. This was the original prefix used for the template of the test, so I respected that. But I think it is worth mentioning that using a shorter prefix like 'UT' would make this more readable.
- Some initial values for variables like `UnrealTestHeroBase::MaxHealth` or `UnrealTestWeaponBase::Damage` are set in the header of the class, but ideally would be set using a data table or something similar. This way, artists and designers can modify them easier
- As the only way a character can die is through the take damage event, I made `UnrealTestHeroBase::CurrentHealth` a private variable and created a method `UnrealTestHeroBase::ModifyCurrentHealth(...)` that will call `UnrealTestHeroBase::Die()` if the conditions for it are met. This ideally would be implemented more elegantly like using an observer pattern.
- Some initialization values related to game design are hard coded in some headers. This would ideally be done using a data table exposed to artists and designers, but I decided not to do that to save time.
- The game has only one hero (`AUnrealTestProjectileGunHero`). Adding a new one (i.e `AUnrealTestHitscanGunHero`) should be as easy as extending from `AUnrealTestHeroBase`, however, the blueprint child classes should be handled accordingly as `BP_ThirdpersonCharacter` currently inherits directly from `AUnrealTestProjectileGunHero` (i.e creating one blueprint class for each C++ hero class)

- I started to implement the logic for sessions and host/join them but I did not have enough time to finish it. I also started to create the UI for host/join too. The game seems to be able to host sessions but for some reason, clients, can find those sessions.