

Feature Documentation: LiquidX UE Game Programmer Test

Test 1: Please explain what is wrong with this code. (5-10mins)

```
TArray < AActor * > MyActors;
void PopulateArray(int n) {
    for (int i = 0; i < n; ++i) {
        AActor * ActorToAdd = GetWorld() ->SpawnActor<AActor>();
        MyActors.Add(ActorToAdd);
    }
}
void PrintArray(TArray < AActor * > Array) {
    for (auto Actor: Array) {
        UE_LOG(LogTemp, Warning, TEXT("Actor's name is: %s"), *Actor-> GetName());
    }
}
void BeginPlay() {
    PopulateArray(100000);
    PrintArray(MyActors);
}
```

Notes:

- 1) I think it can be done in one for loop instead of two and use pre allocation for the array to avoid reallocations when array items are added.

```
TArray<AActor*> MyActors;
void PopulateArray(int n)
{
    MyActors.Reserve(n); // Preallocate memory for n actors
    for(int i = 0; i < n; ++i){
        AActor* ActorToAdd = GetWorld()->SpawnActor<AActor>();
        MyActors.Add(ActorToAdd);
        // if the names are required for further use
        UE_LOG(LogTemp, Warning, TEXT("Actor's name is: %s"), *ArrayToAdd->GetName());
    }
}

void BeginPlay()
{
    PopulateArray(100000);
    // in case we just need the number of actors
    //UE_LOG(LogTemp, Warning, TEXT("Number of actors: %d"), MyActors.Num());
}
```

- 2) There could still be a problem with garbage collection when the Actors are destroyed. Unless, the array is used for object pooling

Test 2: Implement gameplay features (4 hours max)

1. A jetpack.

- Used the simple flying by using the CharacterMovementComponent setting and changed the move code in the `AGamePlayTestCharacter` to simulate flying.
- Improvements:
 - Need Animations
 - Need to be activated when a JetPack Pick up actor is in the player's Inventory
 - Fuel cells to limit the usage of the JetPack
 - Need an equipment system to attach and detach Jetpacks to the player.
 - Audio and particle effects
- Class Name: `AGamePlayTestCharacter`
- Input: IA_Fly
- Time: 30 Minutes

2. Simple world interaction mechanics

- Opening doors
 - Triggering switches or buttons
 - Speaking with an NPC
- Created a C++ Interface to handle various types of interaction by adding the interface to the Interactable_base class (a to c).
 - Class Name: `IIInteractWithObjects`
 - Input: IA_Interact
 - Improvements:
 - Speaking with NPC would be better with a dialogue system tool
 - Fix the issues below
 - Problems:
 - Had trouble implementing blueprint functions like DoesImplementInterface. Therefore, Interaction is implemented in Blueprints.
 - Also, the box collision for the Door is disappearing sometimes in the StaticMesh editor so added an extra collision box.
 - Time: 1 hour

3. Picking up and throwing the cubes

- Implemented Using PhysicsHandle Component
- Improvements:
 - To fix the jittering caused when the cubes are moved around

- Need to add Projectile path prediction and projectile movement while throwing the cube.
- Input: IA_GrabAndThrow
- Class Name: [AGamePlayTestCharacter](#)
- Time: 30 minutes

4. Punching and damaging the cubes

- Implemented using impulse and master field along with the destructible mesh
- Improvements:
 - Need to add animations and Audio
- Problems:
 - Did not have access to the FSMasterFeild Class through C++ as they are Blueprints in the Engine Content folder.
- Class Name: [BP_ThirdPerson](#)
- Time: 30 minutes

5. Double jumps

- Implemented by changing the JumpMaxCount variable defined in the UE's character class.
- Class Name: [AGamePlayTestCharacter](#)
- Time: 1 minutes

Remaining time (1 and a half hour) was spent in Testing and making corrections and look for other ways of implementing the features

Files and Path:

- **Test Level:**
 - Map: ThirdPersonMap
 - Path: Content/ThirdPerson
- **Blueprints:**
 - Path: Content/Blueprints