|  |
| --- |
| Feature Documentation |
|  |

Date: 31-01-23

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());

}

1. There could still be a problem with garbage collection.

Test 2: Implement gameplay features (4 hours max)

**Prepare your work environment.**

1. A jetpack.

2. Simple world interaction mechanics

a. Opening doors

b. Triggering switches or buttons

c. Speaking with an NPC

* Created a C++ Interface to handle various types of interaction (a to c)
* Class Name: IIInteractWithObjects
* Improvements:
  + Speaking with NPC would be better with a dialogue system tool
* 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
* Class Name: AGamePlayTestCharacter
* Time: 30 minutes

4. Punching and damaging the cubes

5. Double jumps

* Implemented by changing the JumpMaxCount variable defined in the UE’s character class.
* Time: 1 minutes

6.YOUR CREATIVE IDEA HERE

Test 3: Documentation (20 mins)

Please take some time to explain which features you’ve implemented the thought process behind them and how long it took for you to complete each feature.

Mention what you would have liked to spend more time on for polish, issues and blockers you had faced.

It is preferred if you include your documentation alongside the .uproject file in a text or markdown file.

Alternatively, you may also provide a screen recording with voice over explaining what you’ve done.