

Documentation:

Before I get into my reflection/documentation I want to start by stating that this was my **second time using Unreal Engine with C++**. The first time being yesterday in preparation for this.

Most of my experience with Unreal is through Blueprints and my C++ knowledge is limited to one Unit at University I completed last year. I'm confident that with the right mentor and guidance I will be proficient using C++ in Unreal in 1 – 2 months.

Task 1:

Explain what is wrong with this code:

```
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);
}
```

Answer 1:

To the best of my knowledge the code is syntactically correct however there are some potential issues such as the code not handling any errors/exceptions that might occur during spawning the actors.

Something else I noticed is that the “GetWorld()” should have a reference to the UWorld so it should look something like this: “MyWorld->GetWorld()”. Under where you declare the array you could declare the world with “UWorld* MyWorld”.

The **biggest issue** though is probably trying to spawn 100,000 Actors on BeginPlay() that would most definitely crash Unreal.

Task 2:

Implement two gameplay features.

Answer 2:

I decided to implement the 'Jetpack' and 'Picking up and throwing the cubes.'

You can see my demo video [here](#) and download the repo [here](#).

Jetpack: (2hrs to implement)

The Logic:

- Player Collides with Jetpack (sphere collider).
- Jetpack attaches to player's back.
- Jetpack bool 'isAttached' is set to true.
- Whenever the player presses Space bar it tries to get the jetpack and if valid checks if 'isAttached' is true.
- If true, it will toggle the bool 'usingJetpack'.
- In the Tick function we run the ActivateJetpack() function whenever 'usingJetpack' is true.
- This function adds force to the Character Movement component on the player.

I created the GetJetpackAttached() and ToggleJetpack() functions to use as nodes in the BP_ThirdPersonCharacter.

Result:

When the player holds down Space Bar they start to fly.

With more time:

I would have liked to give the player the ability to remove the jetpack, give the jetpack fuel and a fuel indication on the HUD. With even more time I would have liked to add a jetpack model, correctly attaching the jetpack to right part of the back, animations for equipping and removing the jetpack and of course sounds and visual effects.

I don't like how the input is controlled through the BP_ThirdPersonCharacter Blueprint. If I had more time, I would have researched how to have the Jetpack listen for input.

Throwing Cubes: (2hrs to Implement)

The Logic:

- Player collides with the interactable object (sphere collider).
- The Object is attached to the players hand and collisions are disabled.
- Whenever the player clicks the Left Mouse Button it finds all Interactable objects and if 'beingHeld' is true the ThrowObject() function is called.
- Collisions are re-enabled, physics turned on and force is applied.

Result:

The player can run around grabbing cubes and press the Left Mouse button to throw them.

With more time:

I would have liked to make it so the player can only pick up one object at a time (or maybe one on each hand). Also, once you throw an object you can't pick it up again, I want to fix that. With even more time I'd like to give the player the ability to hold down the Left Mouse button to throw the cube further, adding a curved line indication to where it will land. And of course, throwing/grabbing animations.

Again, I don't like that the input is handled through the BP_ThirdPersonCharacter. I didn't know of another way, so I had to use the hacky approach of getting all the Interactable Objects in the scene and attempt to throw them (checking against the 'beingHeld' bool). Ideally, the player character would have a reference to the object they are holding and when they throw it only that object's throw function is called.

Conclusion:

Thank you for taking the time to review my application/test.

At the very least this was a great learning exercise for me!

If you can I'd love to hear any feedback about my approach to the gameplay features and my code.

Thanks,
Matthew Stasinowsky