Game Engines in AI Exercise 04

Waldemar Zeitler

December 4, 2018

Task 1

Für diese Aufgabe wurde ein Actor erstellt, welcher 3 Boxen beinhaltet. Diese Boxen werden, im Code, mit den passenden Eigenschaften versehen und weitere Eigenschaften werden im Editor gesetzt, wie zum Beispiel die static Eigenschaft der ersten Box.

```
ACubeActor::ACubeActor()
{
	// Set this actor to call Tick() every frame. You can turn
	    ↪  this off to improve performance if you don't need
	    ↪ it.
	PrimaryActorTick.bCanEverTick = false;

    RootComponent = CreateDefaultSubobject<USceneComponent>(TEXT("
        ↪ RootComponent"));

    // Setup for the three boxes
    FirstBox = CreateDefaultSubobject<UStaticMeshComponent>(TEXT("
        ↪ FirstBox"));
    BoxSetup(FirstBox, FVector(0.8, 0.8, 0.8), FVector::ZeroVector
        ↪ );

    SecondBox = CreateDefaultSubobject<UStaticMeshComponent>(TEXT
        ↪ ("SecondBox"));
    BoxSetup(SecondBox, FVector(0.5, 0.5, 0.5), FVector(40, 40, 0)
        ↪ );

    ThirdBox = CreateDefaultSubobject<UStaticMeshComponent>(TEXT("
        ↪ ThirdBox"));
    BoxSetup(ThirdBox, FVector(0.3, 0.3, 0.3), FVector(80, 80, 0))
        ↪ ;

    // Find the meshes for the boxes and set them
    static ConstructorHelpers::FObjectFinder<UStaticMesh> CubeMesh
        ↪ (TEXT("StaticMesh'/Engine/BasicShapes/Cube.Cube'"));
    if (CubeMesh.Succeeded())
    {
        FirstBox->SetStaticMesh(CubeMesh.Object);
```

```
        SecondBox->SetStaticMesh(CubeMesh.Object);
        ThirdBox->SetStaticMesh(CubeMesh.Object);
    }
}

void ACubeActor::BoxSetup(UStaticMeshComponent* Box, FVector
    ↪ Scale, FVector RelativePosition)
{
    // Box settings for the rotation with constraints
    Box->SetupAttachment(RootComponent);
    Box->SetWorldScale3D(Scale);
    Box->SetRelativeLocation(RelativePosition);
    Box->SetSimulatePhysics(true);
    Box->SetEnableGravity(false);
}
```

Der Actor wurden dann in die Welt gesetzt und die Boxen wurden noch mal passend verschoben, um die Rotation besser zu erkennen. Das folgende Bild zeigt die 3 Boxen und den Editor, wo die constraints und der SmallCube zu sehen sind.

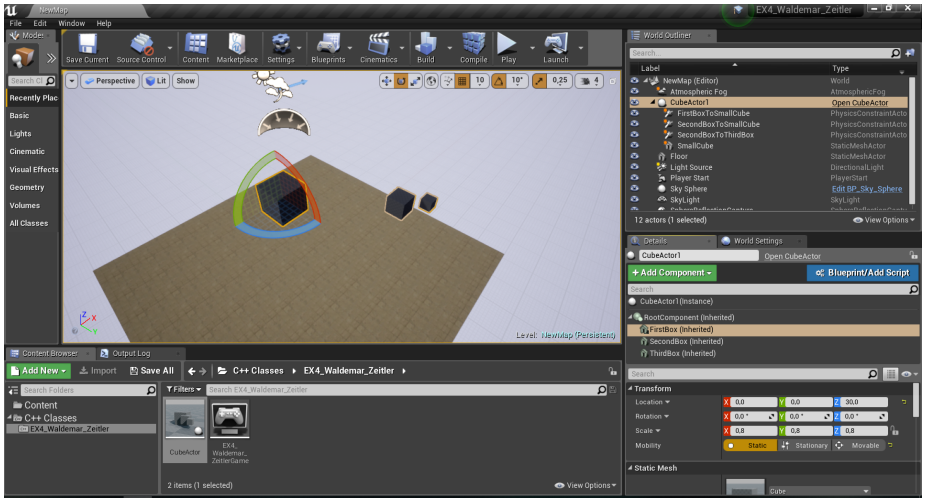

Figure 1: Box Positionen und Constraint Aufteilung

Das Rotieren der zweiten Box um die erste und das Rotieren der zweiten Box um sich selbst funktioniert mit den constraints, nur leider treten bei mir ständig Probleme auf wenn ich die dritte Box hinzunehme. Es ist wahrscheinlich ein Fehler bei der Einstellung der korrekten Geschwindigkeit. Diesbezüglich habe ich die dritte Box so stehen lassen. Diese rotiert nun um sich selbst, mit offenen constraints zu zweiten Box. Die Geschwindigkeit für die dritte Box ist ebenso im Code ausgeklammert.

Task 2

Für diese Aufgabe habe ich einen extra Actor mit drei Boxen erstellt, welche den Roboter simulieren sollen. Alle Teile sind als RobotPart collison deklariert und wie in der Aufgabe beschrieben gesetzt. Die Kanone wurde mit "Brushes" und zu einem Static Mesh umgewandelt. Dies führte allerdings zu Problemen und es ist nicht richtig möglich ein constraint dazu herzustellen, das man damit keine Physik simulieren kann.
Da ich mit der Aufgabe leider zu spät angefangen habe und das Problem nicht gelöst bekommen habe, sind beide Aufgaben nun in einem unfertigen Zustand. In der UMoveRobotComponent ist auch nur ein Beispiel für ein FTransform, der wie bereits erwähnt nicht funktioniert.