

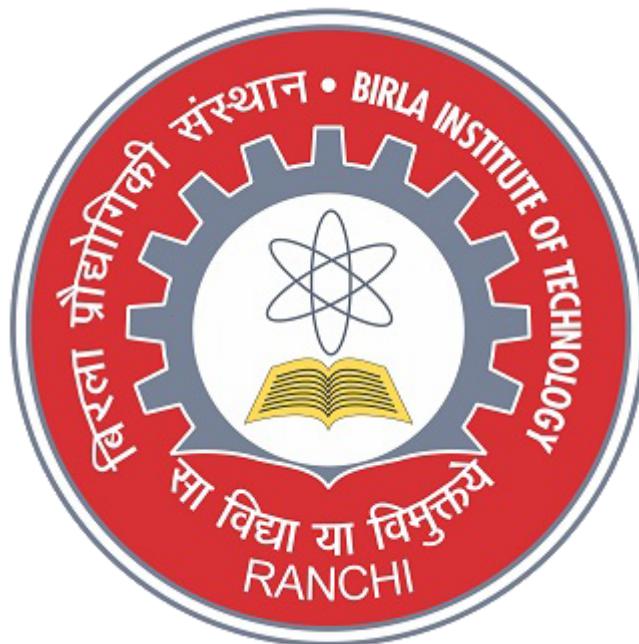
GAME USING UNREAL ENGINE

A Thesis

submitted in partial fulfilment of the requirements for
the award of the degree of
Bachelor Of Engineering

BY

VANGALAPUDI APURV(BE/10152/2017)



DEPARTMENT OF INFORMATION TECHNOLOGY

BIRLA INSTITUTE OF TECHNOLOGY

MESRA-835215, RANCHI

2020

APPROVAL OF GUIDE

This is to certify that the work presented in the project/thesis entitled “Game Using Unreal Engine” for the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering in Information Technology of Birla Institute of Technology Mesra, Ranchi is absolutely based on their authentic work carried out under my supervision and guidance.

I further declare, to the best of my knowledge and belief, that the content of this

thesis does not contain any reproduction of any work which has been submitted for the award of any previous Degree to anyone else.

DATE:14/05/2021

(Dr. Satish Chander)

Dept. of Computer Science and engineering

Birla Institute of Technology Mesra, Ranchi

DECLARATION CERTIFICATE

The mentioned thesis entitled “Game Using Unreal Engine”, is hereby approved as a reliable study of the research topic and has been presented in adequate manner to justify its acceptance as necessary to the degree for which it has been submitted.

It is understood that by this approval, the undersigned does not necessarily endorse any conclusion drawn or opinion expressed therein, but approve the thesis for the purpose for which it is submitted.

VANGALAPUDI APURV
(BE/10152/2017)

CERTIFICATE OF APPROVAL

The mentioned thesis entitled “Game Using Unreal Engine”, is hereby approved as a reliable study of the research topic and has been presented in adequate manner to justify its acceptance as necessary to the degree for which it has been submitted.

It is understood that by this approval, the undersigned does not necessarily endorse any conclusion drawn or opinion expressed therein, but approve the thesis for the purpose for which it is submitted.

(Internal Examiner)

(External Examiner)

**(Chairman)
Head of the Department**

INTRODUCTION

I have built a game which takes inputs from user and puts those values in blueprints which makes the character move at all directions using force systems and it also contains a weapon, boosters, elevators using sequencer, destructible objects. Using a Radial Force Actor, you can apply a constant force (listed as *force strength* in the editor) or an impulse. Impulses need to be fired from a blueprint, and apply their entire force in only one frame, as compared to every frame from a Constant Force. In the content example, only a Constant Force is being applied. If an Object has mobility set to movable and it has a simple collision, you can click the check flag simulate physics in the physics rollout and the Object will be simulated when simulate in editor or play in editor is clicked. Skeletal Meshes can only be created with external software like 3D Studio Max or Maya; however, their collision is setup in Unreal Editor. When importing, a Physics Asset can be automatically created if create physics asset is checked, but the asset created will be an estimation based on the joints and skin weights applied to those joints, so some editing is necessary by using the physics asset tool or PhAT.

SYSTEM REQUIREMENTS

UNREAL ENGINE:-Unreal Engine is the world's most open and advanced real-time 3D creation tool. Continuously evolving to serve not only its original purpose as a state-of-the-art game engine, today it gives creators across industries the freedom and control to deliver cutting-edge content, interactive experiences, and immersive virtual worlds.

BLUEPRINTS:-Language used.Blueprints is the visual scripting system inside Unreal Engine 4 and is a fast way to start prototyping your game. Instead of having to write code line by line, you do everything visually: drag and drop nodes, set their properties in a UI, and drag wires to connect.

In addition to being a fast prototyping tool, Blueprints also makes it very easy for non-programmers to dive in and start scripting.

This system is extremely flexible and powerful as it provides the ability for designers to use virtually the full range of concepts and tools generally only available to programmers. In addition, Blueprint-specific markup available in Unreal Engine's C++ implementation enables programmers to create baseline systems that can be extended by designers.

The following approach was taken to make this game:

Press the right mouse button and blueprint class can be seen then select it

and another layout is going to pop up like shown in this picture and in that

select character blueprint to create a character. Character blueprint has all

the components related to the desired movement of the character.

In the components folder are those components which are used for my desired character and some components are made child to some other components for the convenience of the desired input control system and the weapon(Futuristic_Weapon_Lowpoly61) is a free asset downloaded from internet.

Action and axis mappings can be created from projects settings under input

section. Axis events for mouse's x and y axis values are created for turn and look up events respectively. These values are used for turn the character left to right and to turn the weapon top to down.

Event tick is an event that is executed every second. So, it is used to make

the camera and the weapon stay with the sphere at all times.

Axis events to make the sphere move forward and backward using force system are used in the above slide.

Action mapping to make the sphere jump using force system is created in

above slide.

when the left mouse button is pressed then from the socket(created manually) of the muzzle, a bullet (blueprint in upcoming slides) is spawned

with a sound of gun shot at that location.

An action mapping for pausing the game is created. Two functions are created to pause and un-pause (widgets in upcoming slides) the game.

In this project, two levels are created which are for main menu section and

game world area as shown .A widget for main menu is created which consists of various boxes with text block in it which defines what's supposed to happen with each button.

The widget for pause menu when the action mapping for pause is activated

and what each button is supposed to do.

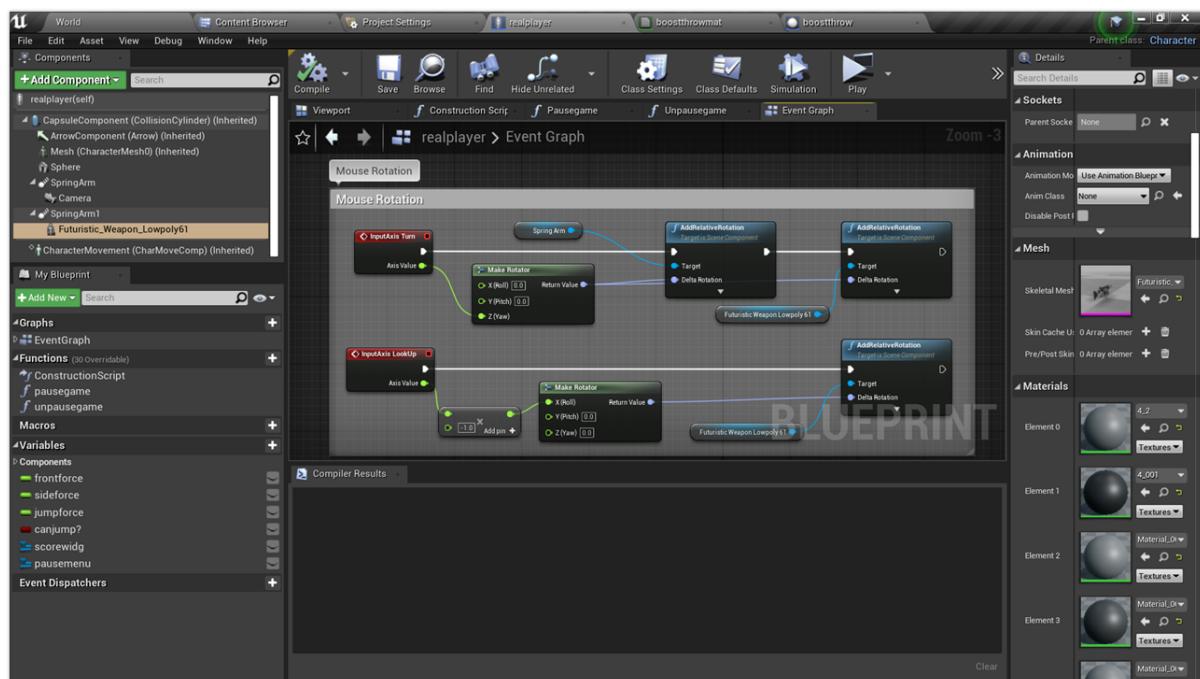
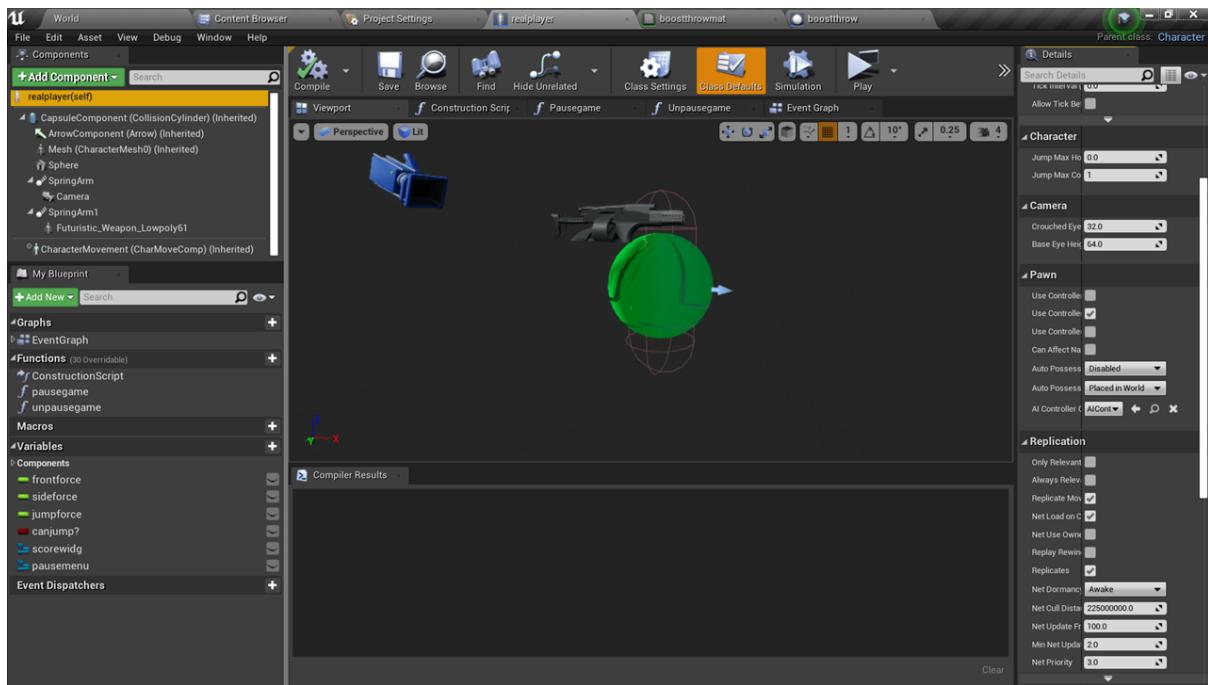
An actor blueprint is present whose purpose is to force boost the character

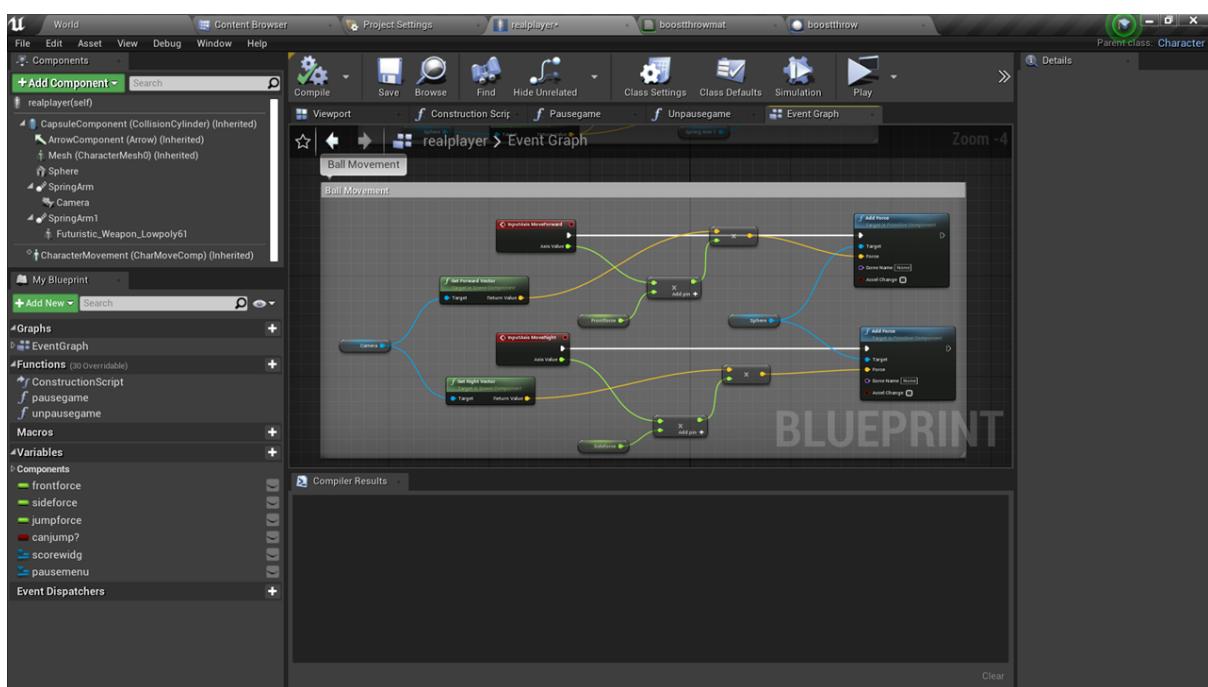
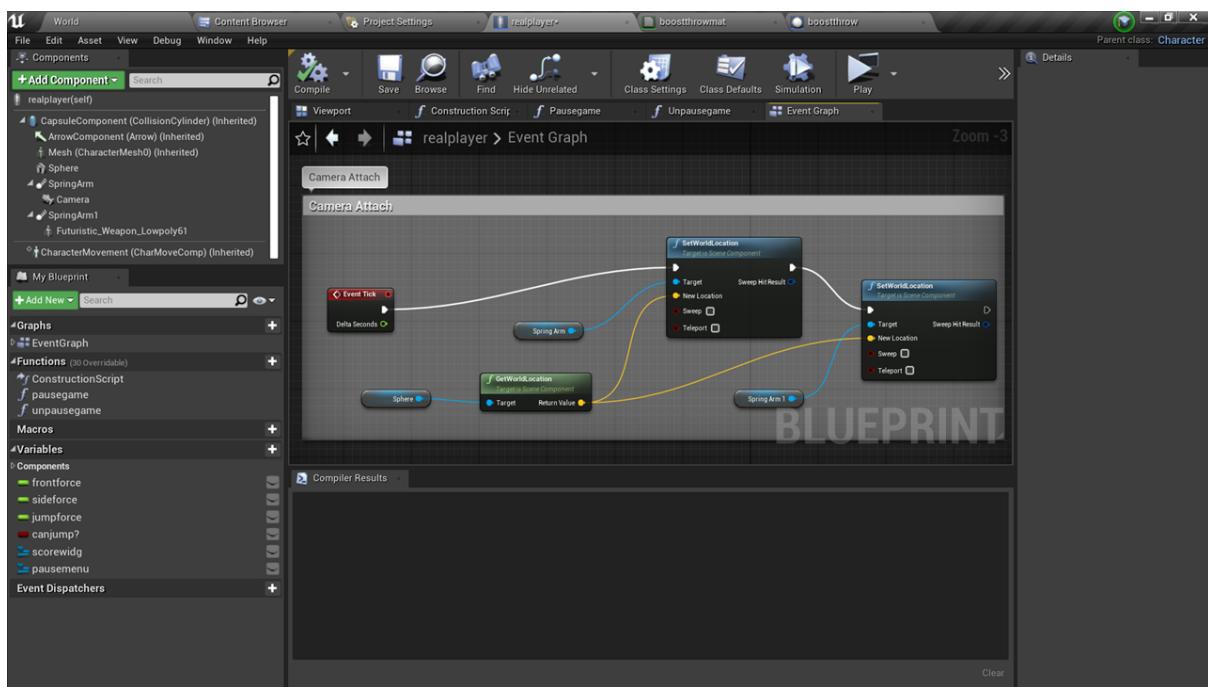
in manually changeable coordinates.

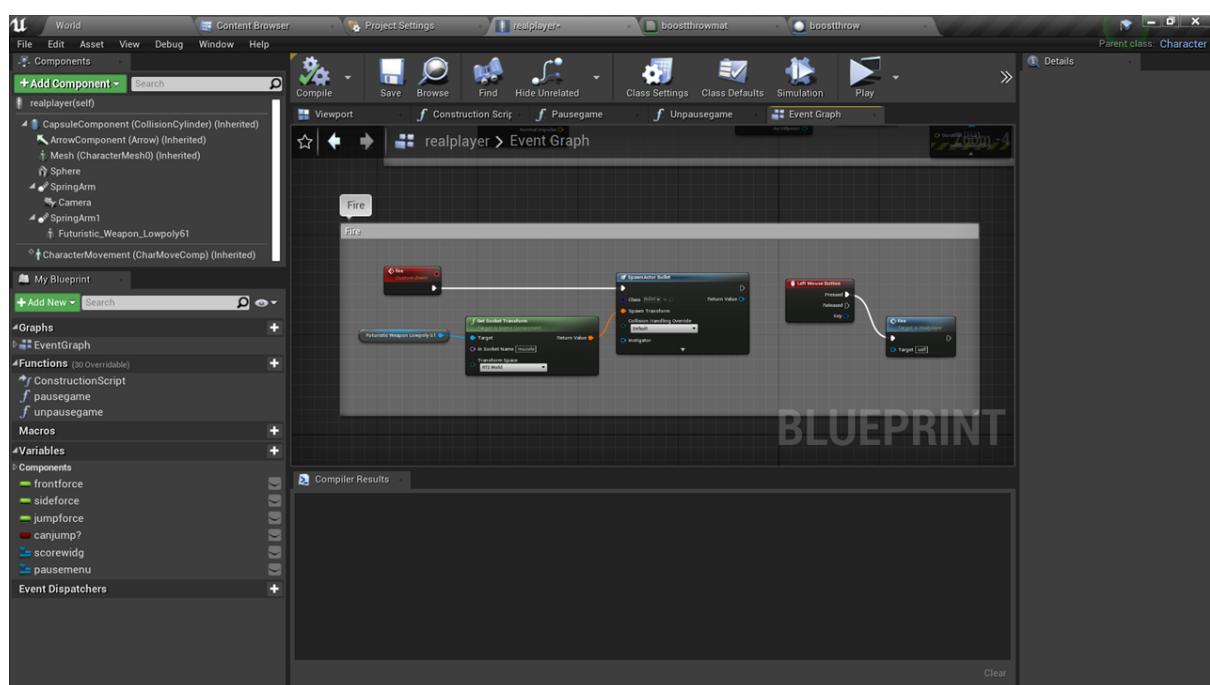
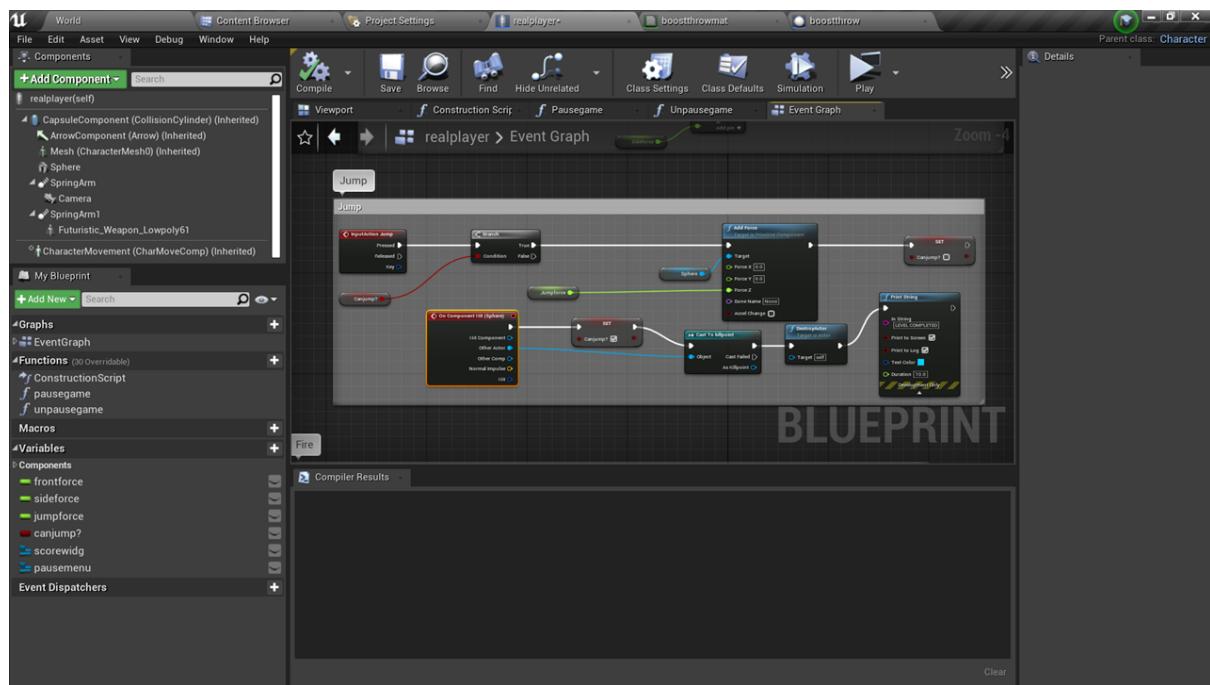
An actor blueprint is present which can be also referred as bullet that is spawned from the socket of the weapon with a gun shot at that location sound when the left mouse button is pressed.

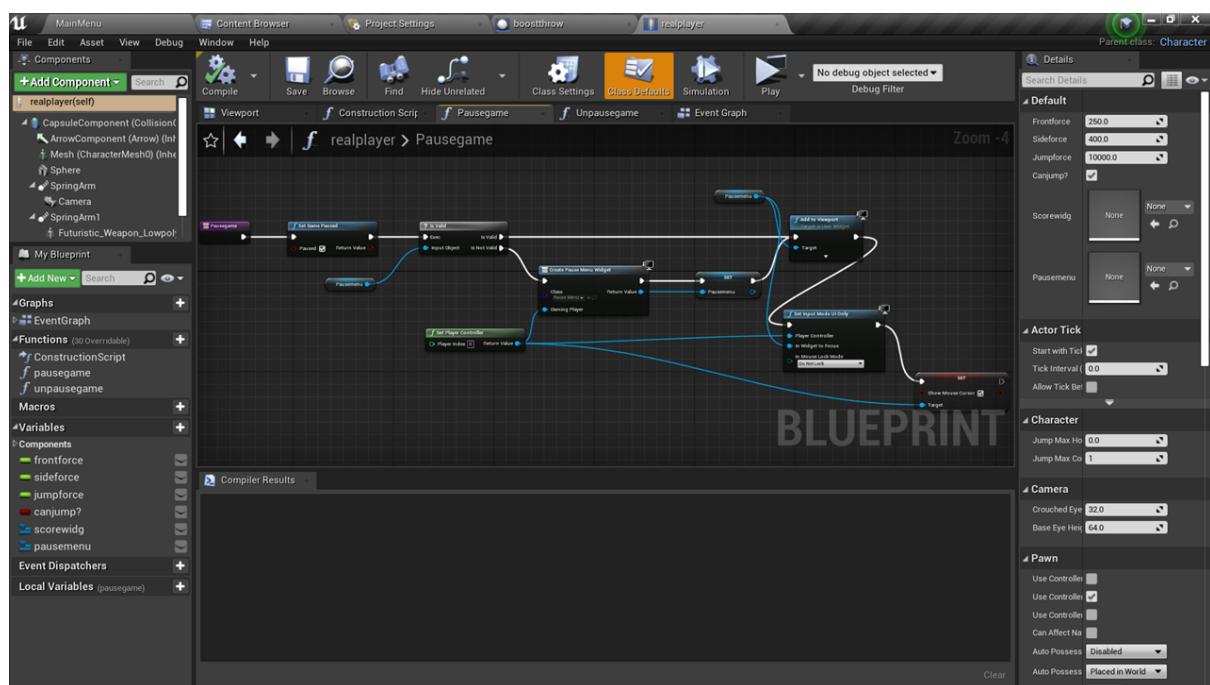
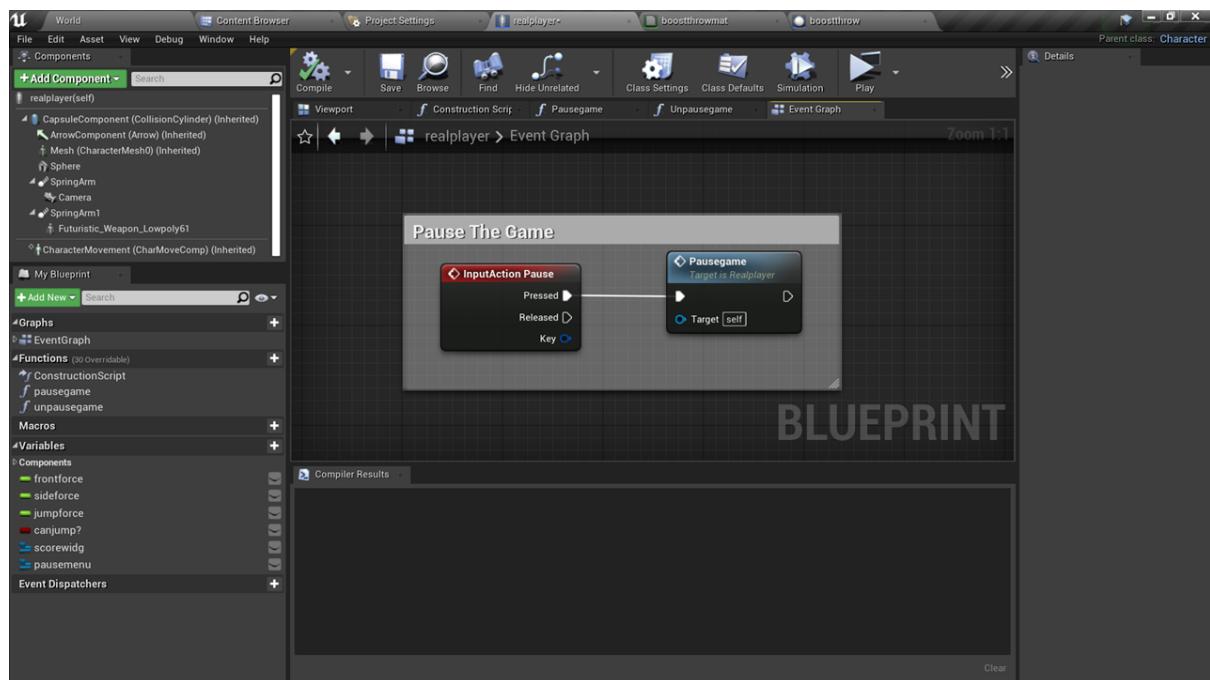
An actor blueprint is present which can also be referred as coin whose purpose is to rotate in z axis as if it is floating and when the character touches the coin it destroys with a sound at that location and the score value increases by one point.A widget for score value is visible and how that widget is affected by the character in game.

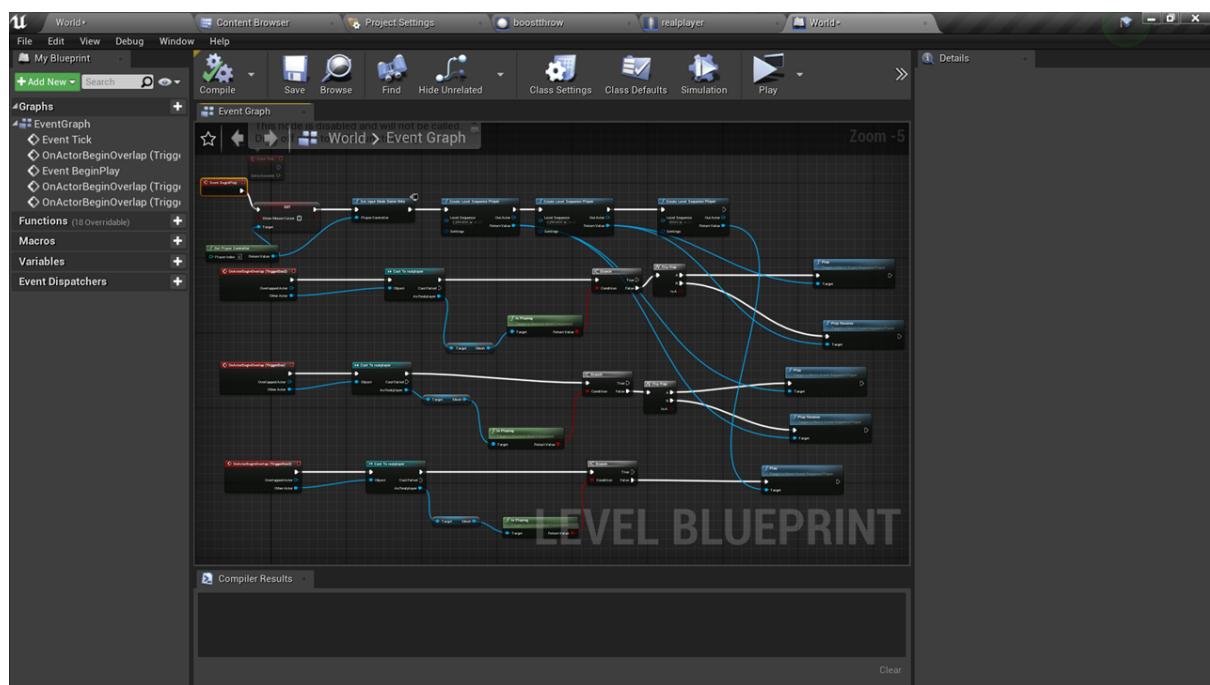
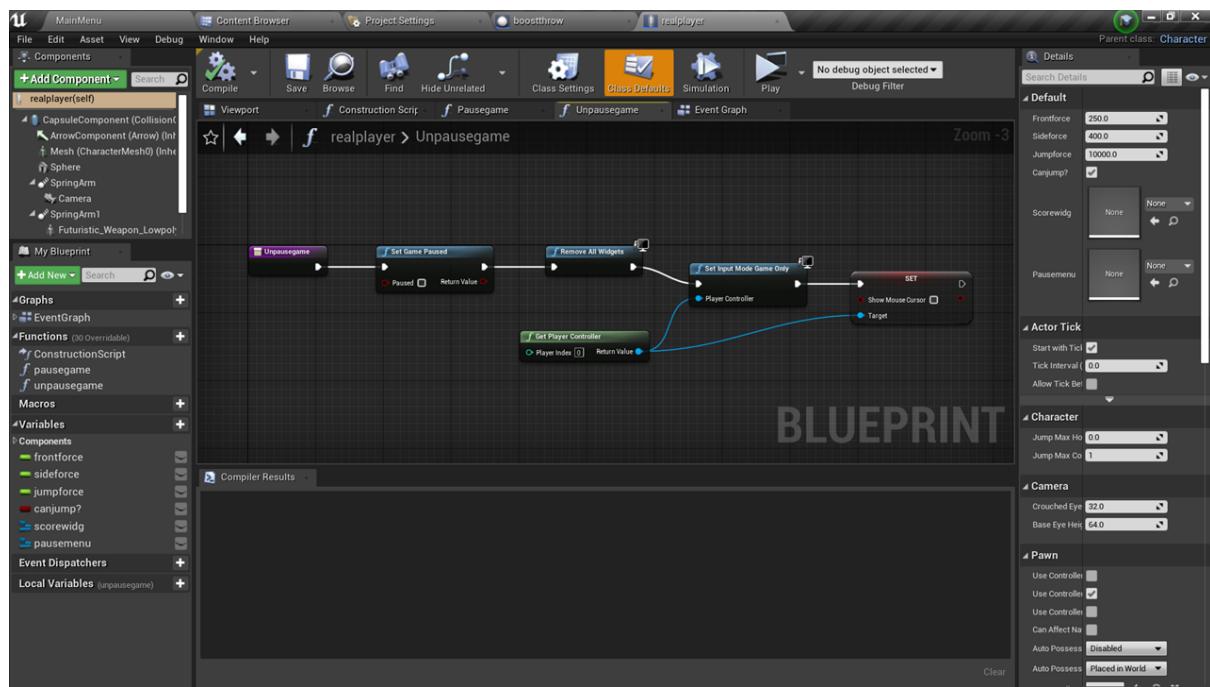
An actor blueprint is present whose purpose is to deal the damage when the bullet touches this actor and after a particular damage this is supposed to destroy with a pain sound at that location.

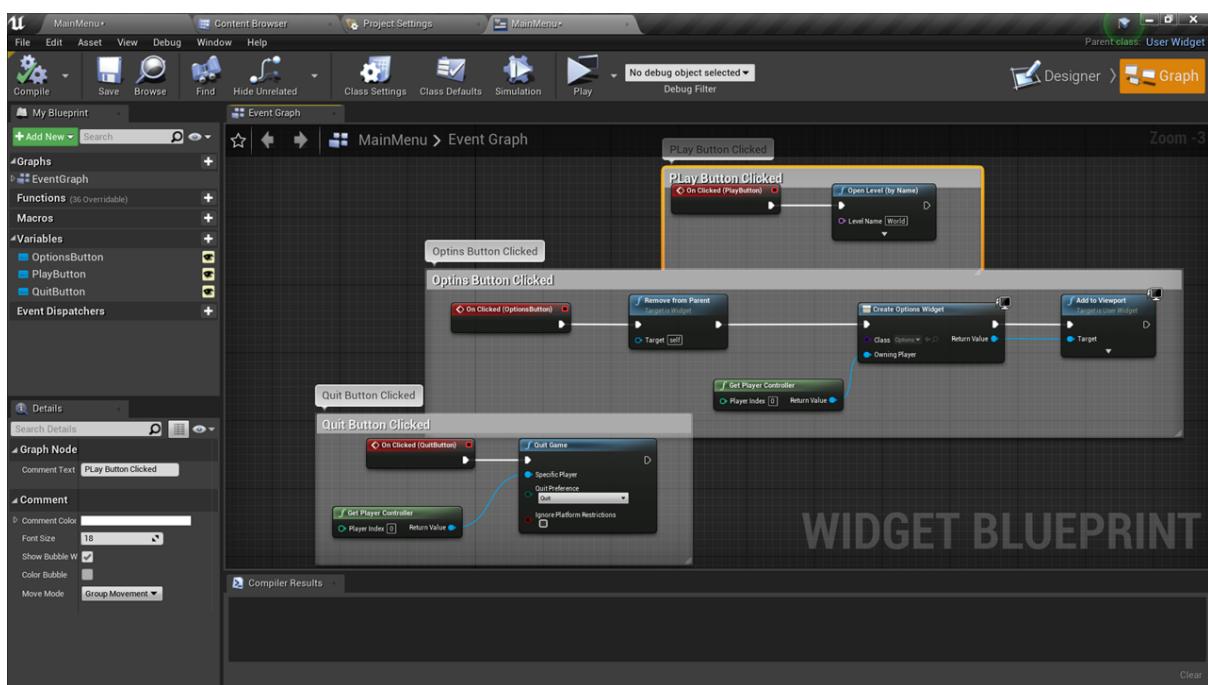
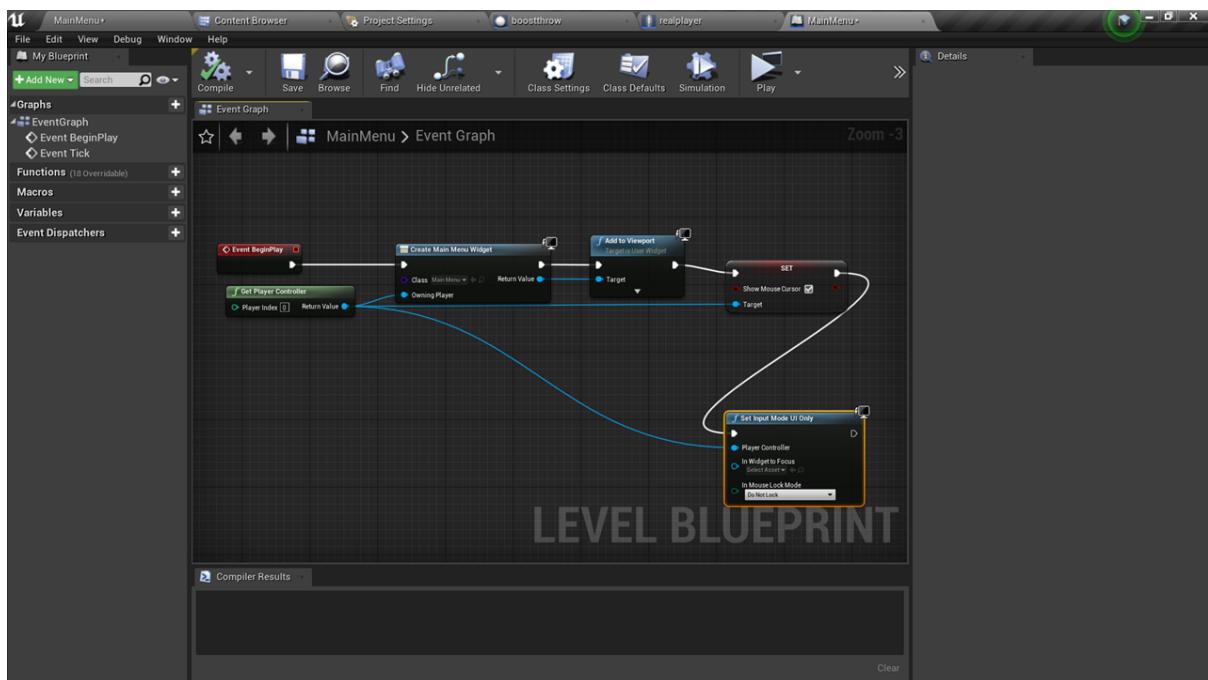


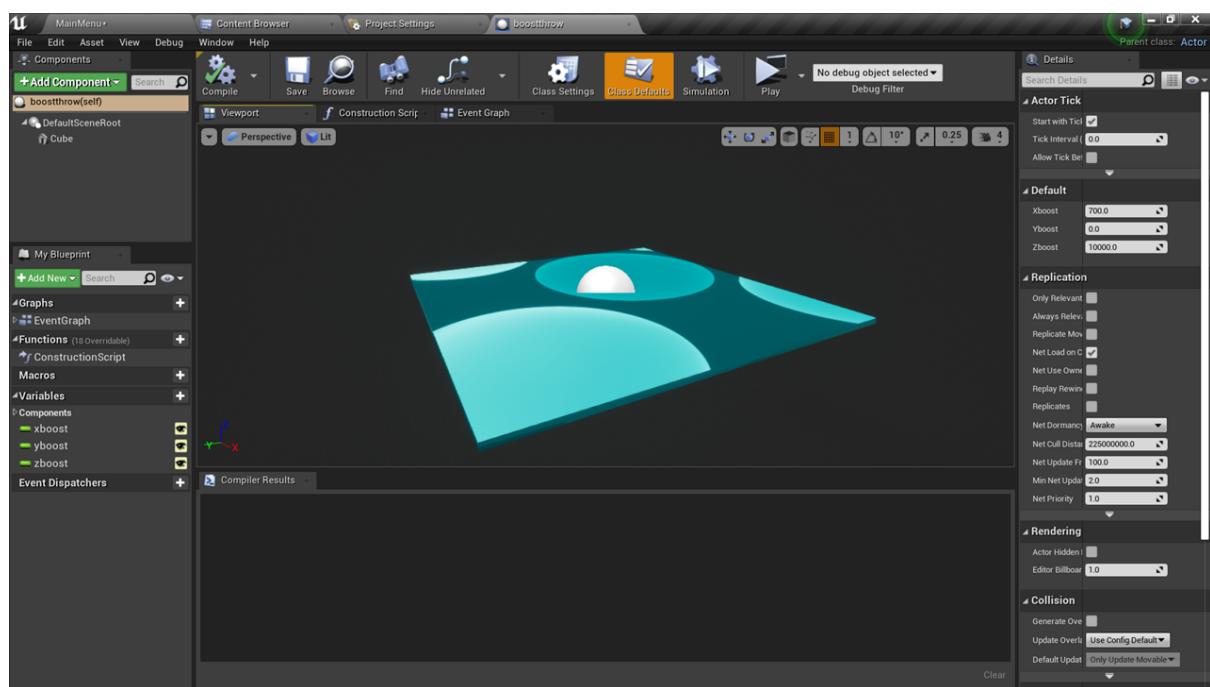
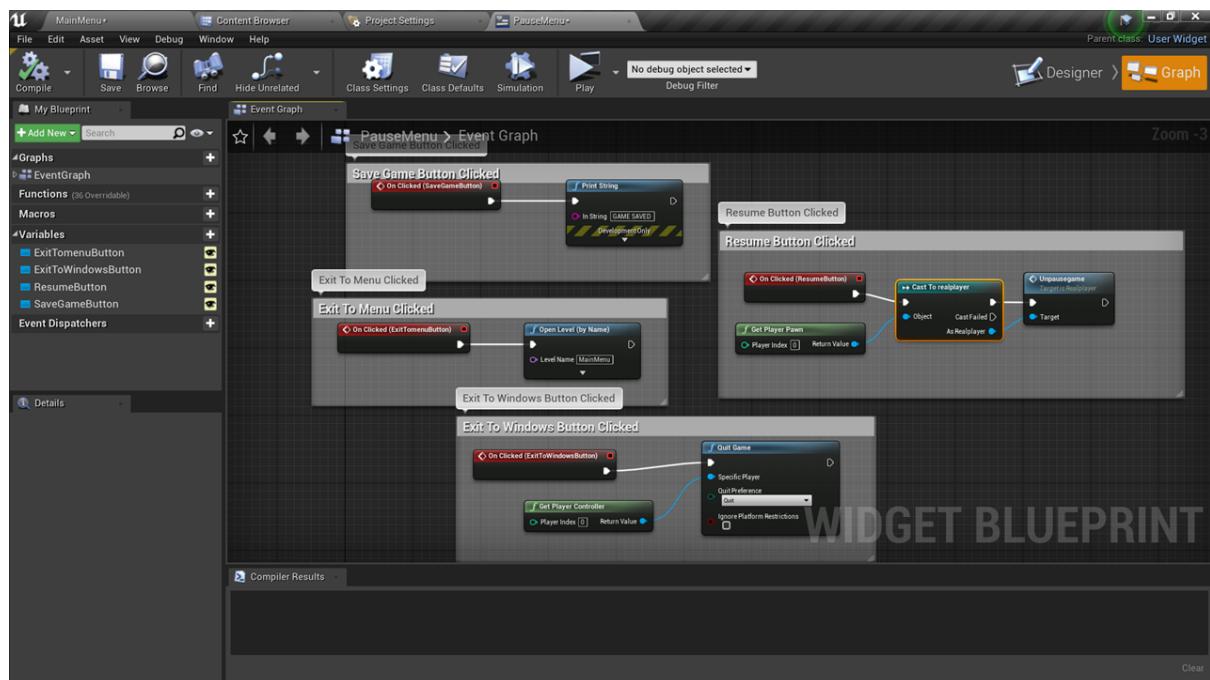


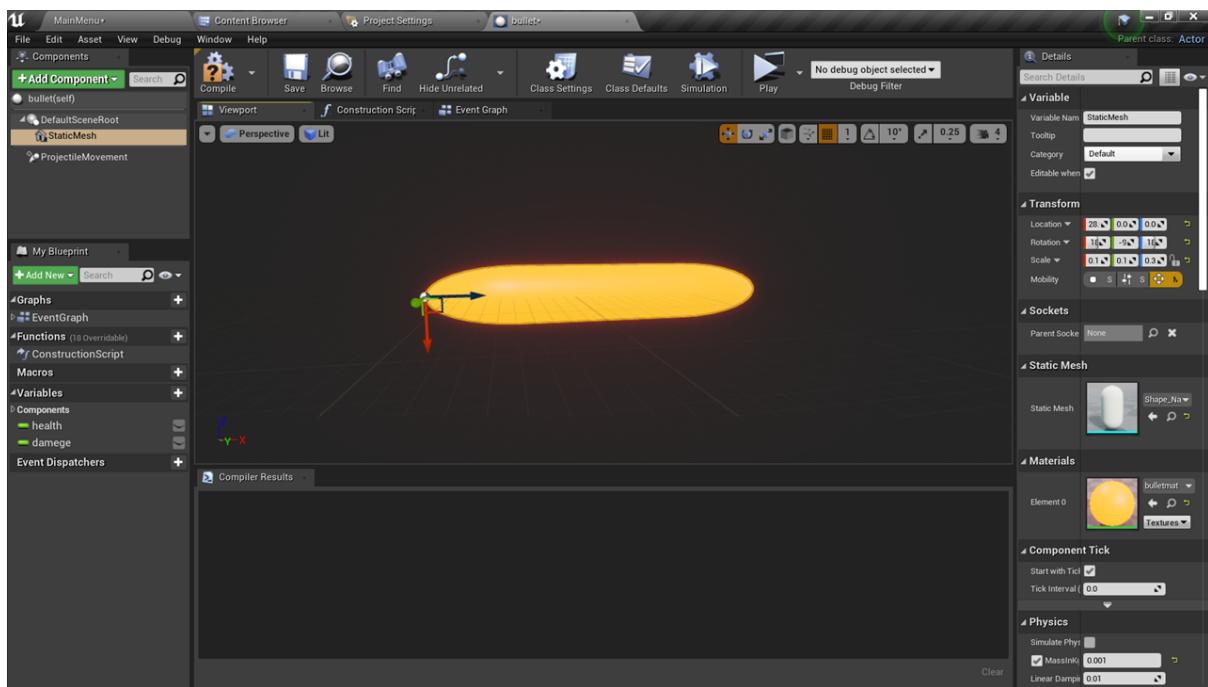
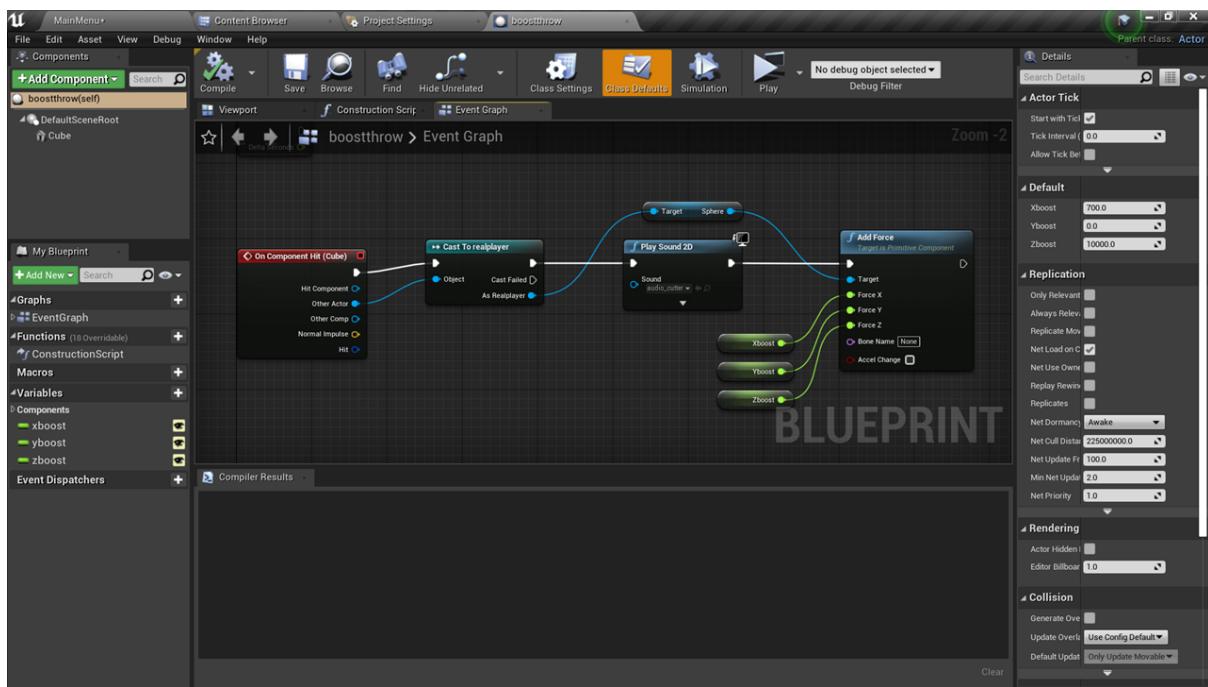


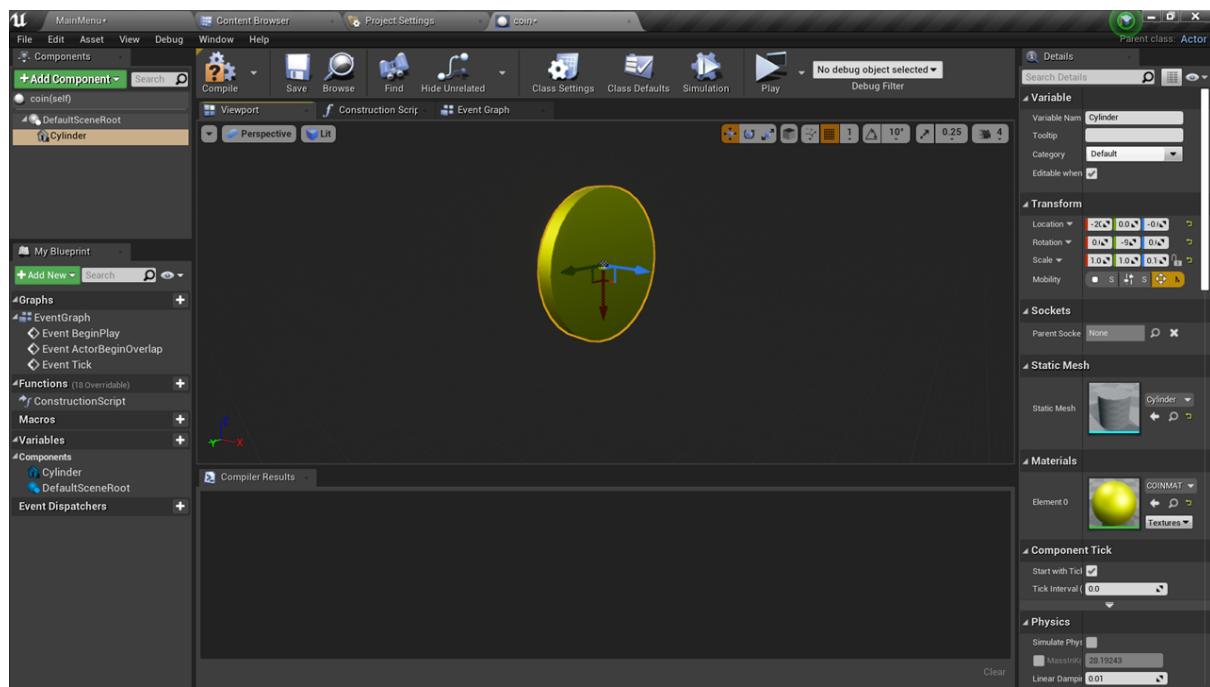
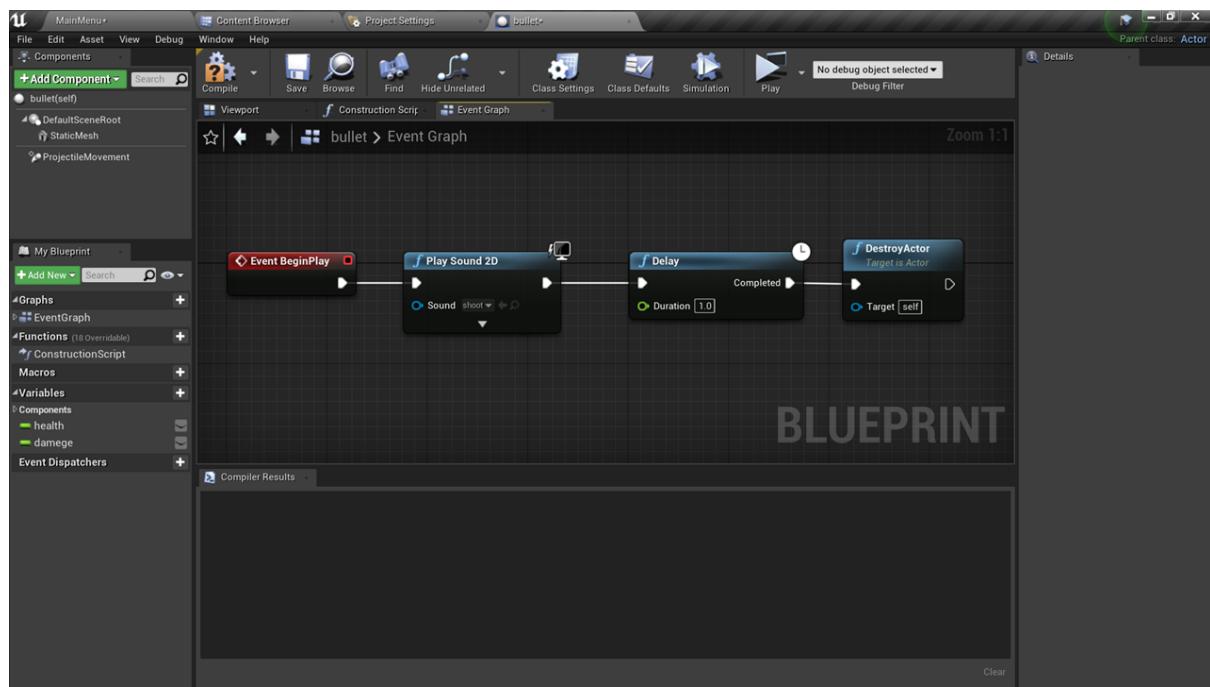


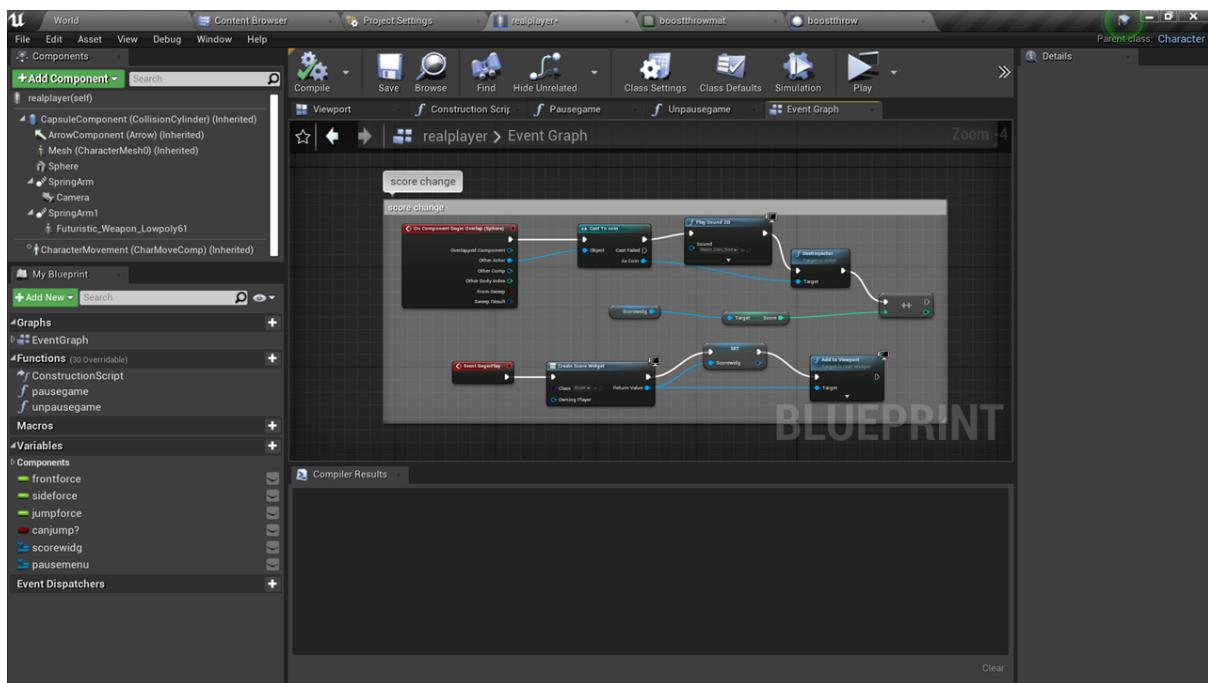
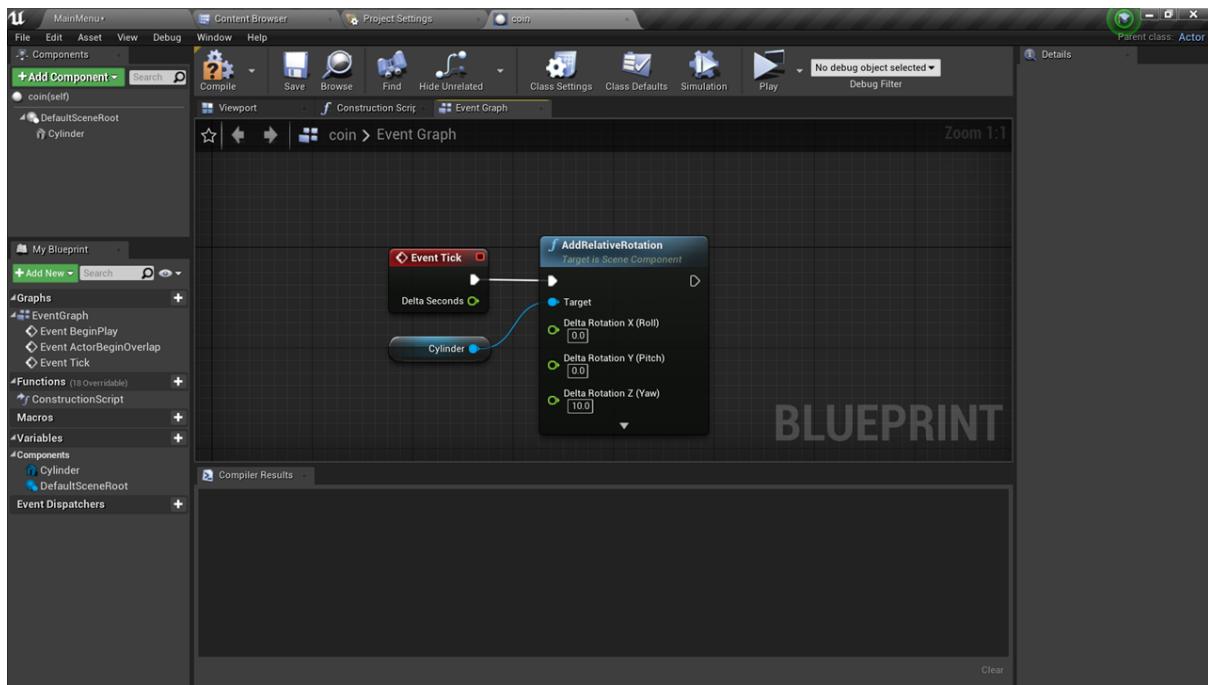


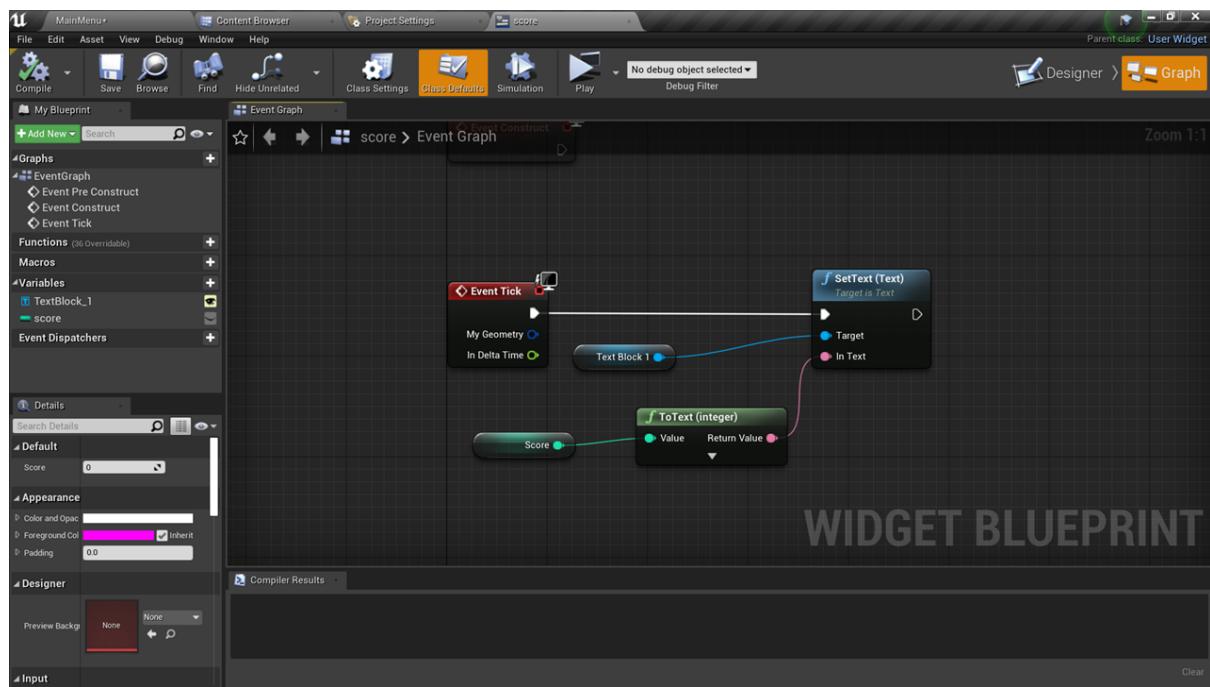




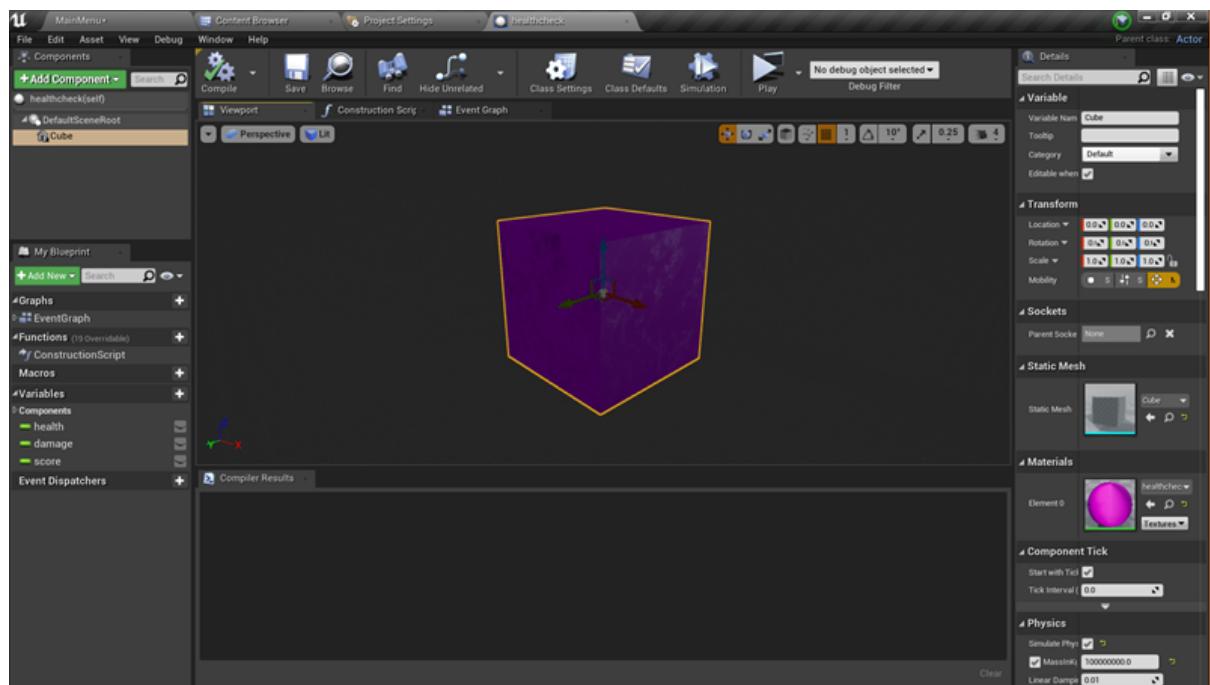


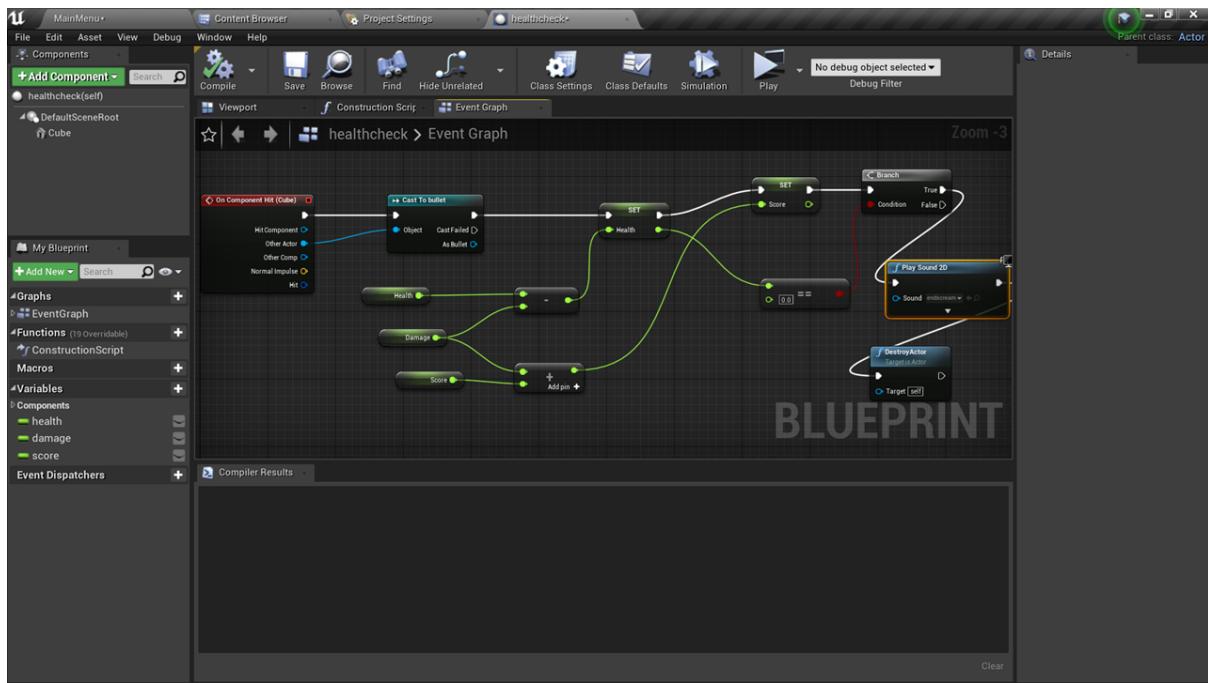






WIDGET BLUEPRINT





SEQUENCER

The Sequencer Editor gives users the ability to create in-game cinematics

with a specialized multi-track editor (similar to matinee). By creating level sequences and adding tracks, users can define the makeup of each Track,

which will determine the content for the scene. Tracks can consist of things

like Animations (for animating a character), Transformations (moving things

around in the scene), Audio (for including music or sound effects), and several other Track types.

The Level Sequence is the "container" for your cinematic scenes, and must

be created in order to begin working inside of the Sequencer Editor. You can create a Level Sequence directly in your Level from the toolbar under

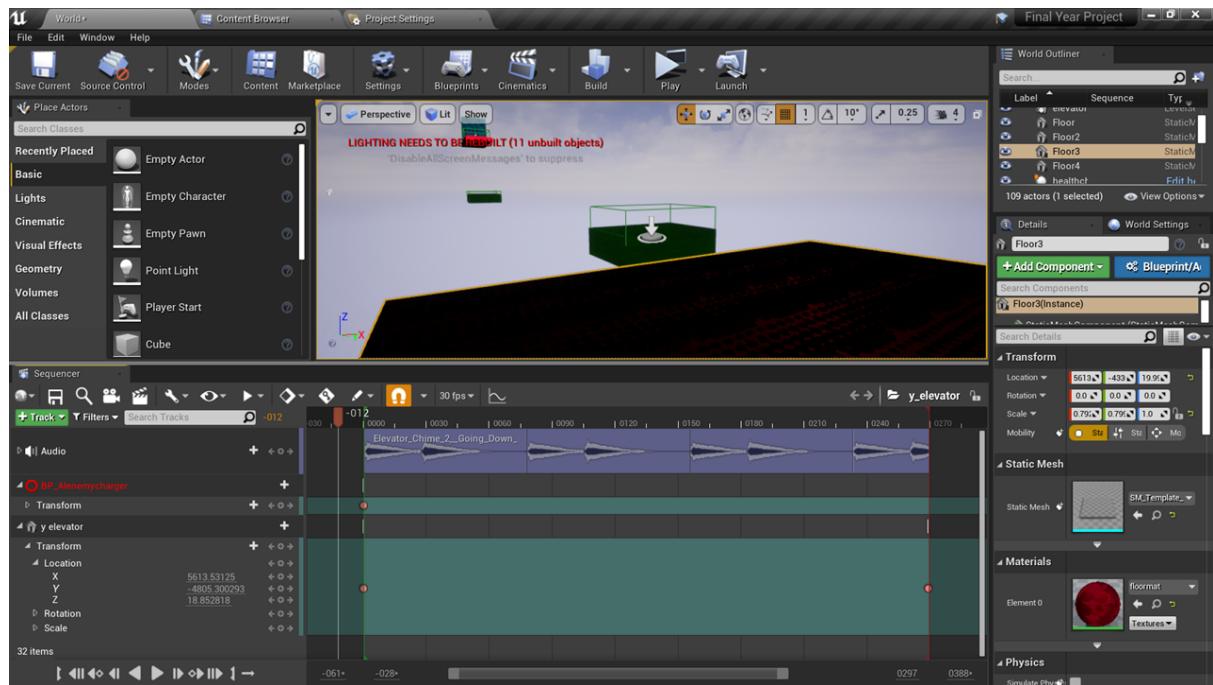
cinematics.

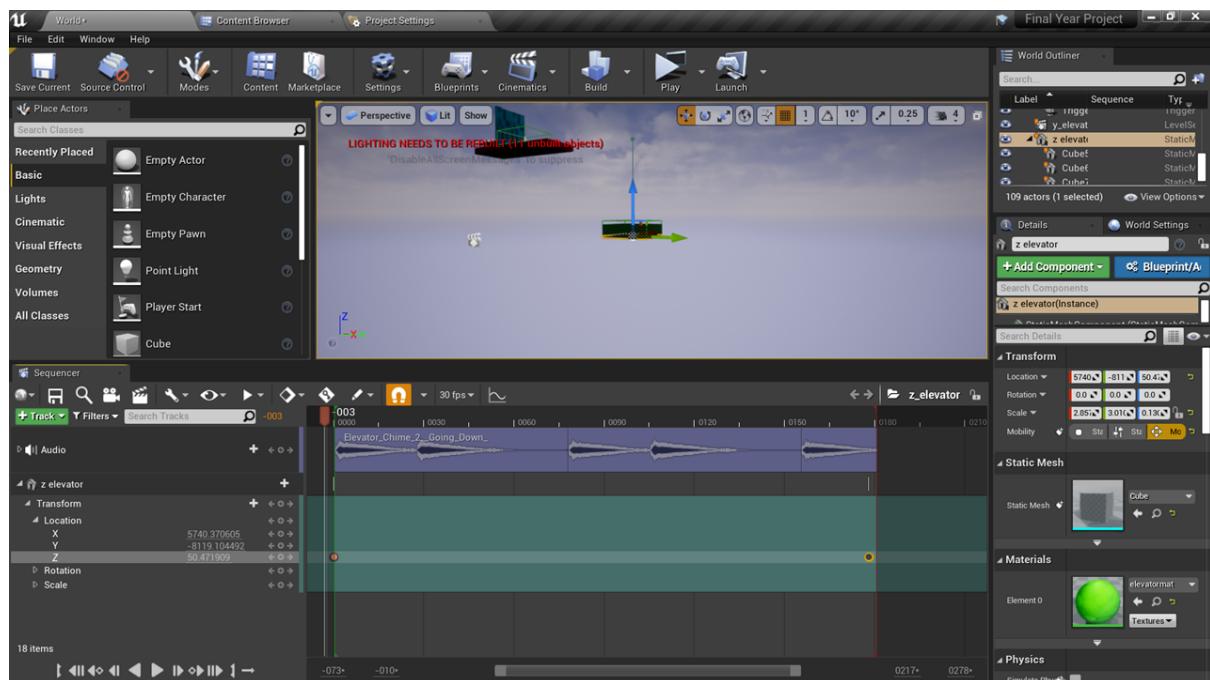
This project consists of level sequences for two elevators and an opening

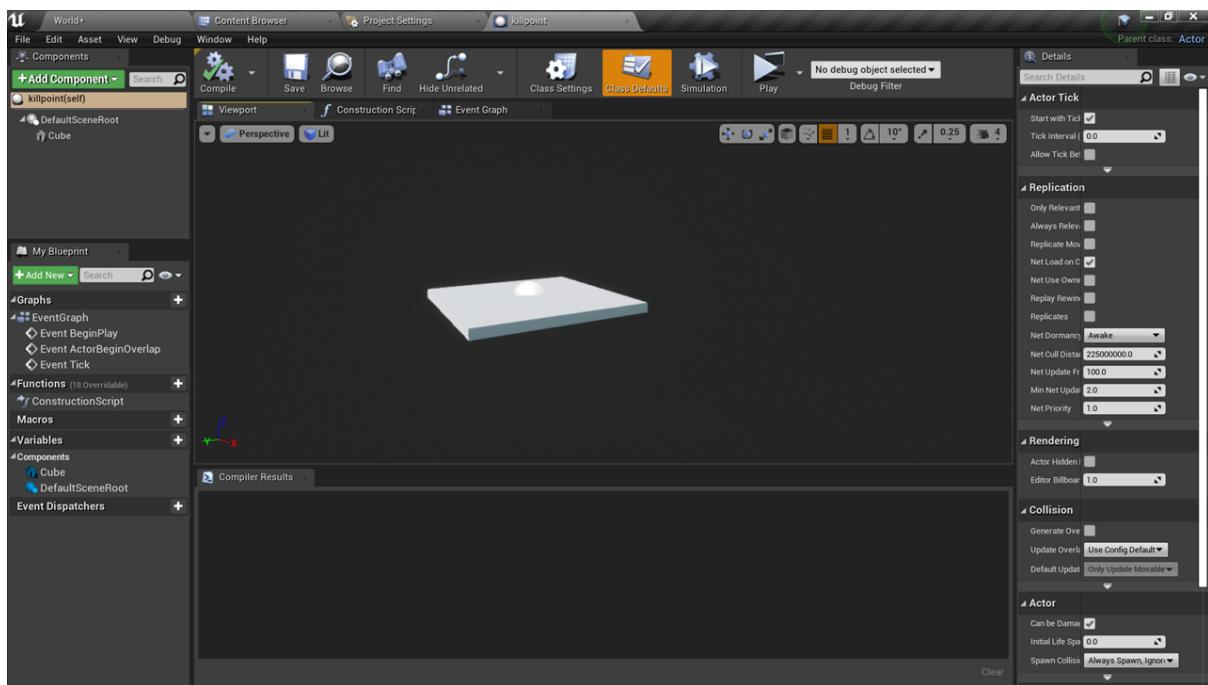
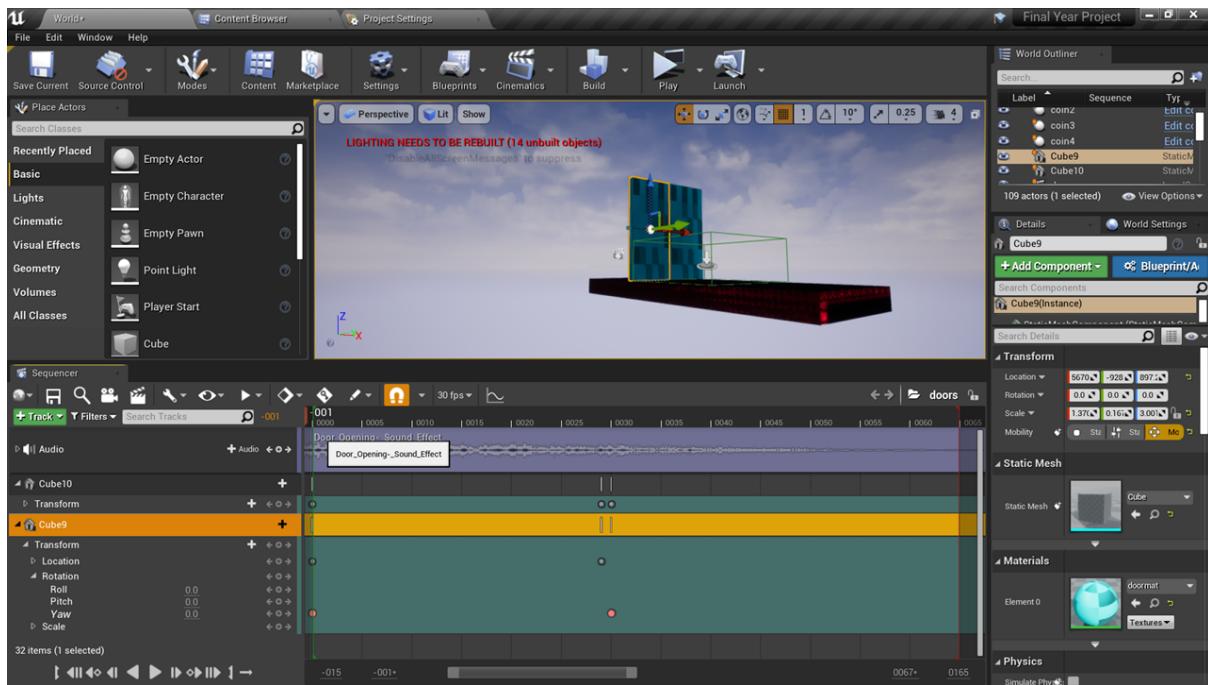
door with suitable audios generated from that specific location respectively

with trigger boxes which tells the editor when it is triggered. One elevator is

moving in y-axis, another in z-axis and the opens and rotates simultaneously when triggered.







And finally we have an actor blueprint whose purpose is to destroy the character when touched and print a statement that says, "LEVEL COMPLETED".

MATERIALS USED FOR ALL ACTORS AND CHARACTERS

A Material is an asset that can be applied to a mesh to control the visual look of the scene. At a high level, it is probably easiest to think of a Material

as the "paint" that is applied to an object. But even that can be a little misleading, since a Material literally defines the type of surface from which

your object appears to be made. You can define its color, how shiny it is, whether you can see through the object, and much more.

In more technical terms, when light from the scene hits the surface, a Material is used to calculate how that light interacts with that surface. These

calculations are done using incoming data that is input to the Material from

a variety of images (textures) and math expressions, as well as from various property settings inherent to the Material itself.

