

ANIMATION AND SOUND WITH UNREAL ENGINE 4



DESIGNER GUIDE DOCUMENT

Jordan Kaspers

Index

| | |
|--------------------------------|-----------|
| Index | 1 |
| Introduction | 2 |
| Finding your way around | 2 |
| Playing the Demo | 2 |
| Input | 2 |
| Goal | 2 |
| Not working? | 3 |
| Characters | 3 |
| Player | 3 |
| Description | 3 |
| How to - Player | 4 |
| Customisation | 4 |
| Animation | 5 |
| Gameplay Elements | 6 |
| Doors | 6 |
| Description | 6 |
| How to - Doors | 7 |
| Customisation | 7 |
| Lights | 8 |
| Data | 8 |
| Analysis | 9 |
| Conclusion | 9 |
| Audio System | 10 |
| How to - Sound | 10 |

Introduction

The purpose of this document is to show any designers working on this Animation and Sound Demo how to control and use any part of the project. The basics of the Demo are explained, as well as modifying certain areas to customise the overall experience. There will be an area going through what the effect light has on performance, as well as an audio system walk-through.

Finding your way around

Once you open the project, you should automatically find yourself looking at the editor with the Game_P map showing. From here, you can hit play and the demo will start. Once the demo is playing, use the Escape key on the keyboard (Esc) to quit the demo.

Playing the Demo

Input

| | |
|---------------------------|--------------------|
| W | Move forward |
| S | Move backward |
| A | Move left |
| D | Move right |
| Space Bar | Jump |
| Mouse Movement | Control camera |
| Left Mouse Button | Pick up object |
| Right Mouse Button | Cast a light spell |

Goal

The goal of the Demo is to navigate through different puzzles and to unlock a different room every time a puzzle is solved. The Player is to find and pick up a box, and drop the box in the designated areas next to the doors in order to pass through to the next area. Since the world has no light, the Player is to use the light spell in order to see where to go.

Not working?

- If the Game_P map did not load automatically, go to the Maps folder and select the desired map.
- Go to project properties, maps and modes, and select the game mode to be BP_Practica_kaspersGameMode.

Characters

The demo only has one character at the time of the creation of this document.

Player



Description

The Player is the main (and only) character of the game, playable and to be controlled by the user. The Player is used to navigate the level and to pick up the necessary boxes in order to go on to the next room. The Player will need to cast the light spell in order to see, as without the light spell, the Player will not be able to see much due to a lack of other light sources. The Player, apart from moving around, can also jump, and will be required to do so in order to access certain areas.

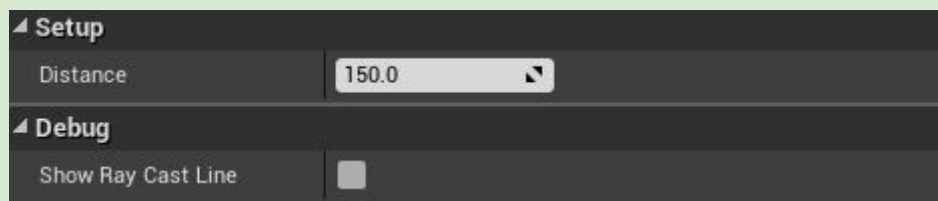
How to - Player

There can only be one Player in the world. The PlayerStart of UE4 can be used to spawn the Player into the world when the game begins. Another way of placing the Player would be to go to the Player Blueprint (BP_PlayerCharacter) in the Blueprints folder, and dragging it to a desired position.

Customisation

The Player can be customised in several ways.





In the Blueprints folder, open the Player Blueprint (BP_PlayerCharacter), and by looking in the Components Tab, select the ObjectHandler Component. Once selected, in the Details Tab, there are some properties that can be changed:



- Distance
 - Category = Setup
 - Default value = 150.0
 - Represents the minimum distance the Player is required to be from an object in order to be able to pick it up.
- Show Ray Cast Line
 - Category = Debug
 - Default value = false (unchecked)
 - As the category name implies, used for debugging and if checked, will show a line representing what the player can see and the range.

Animation

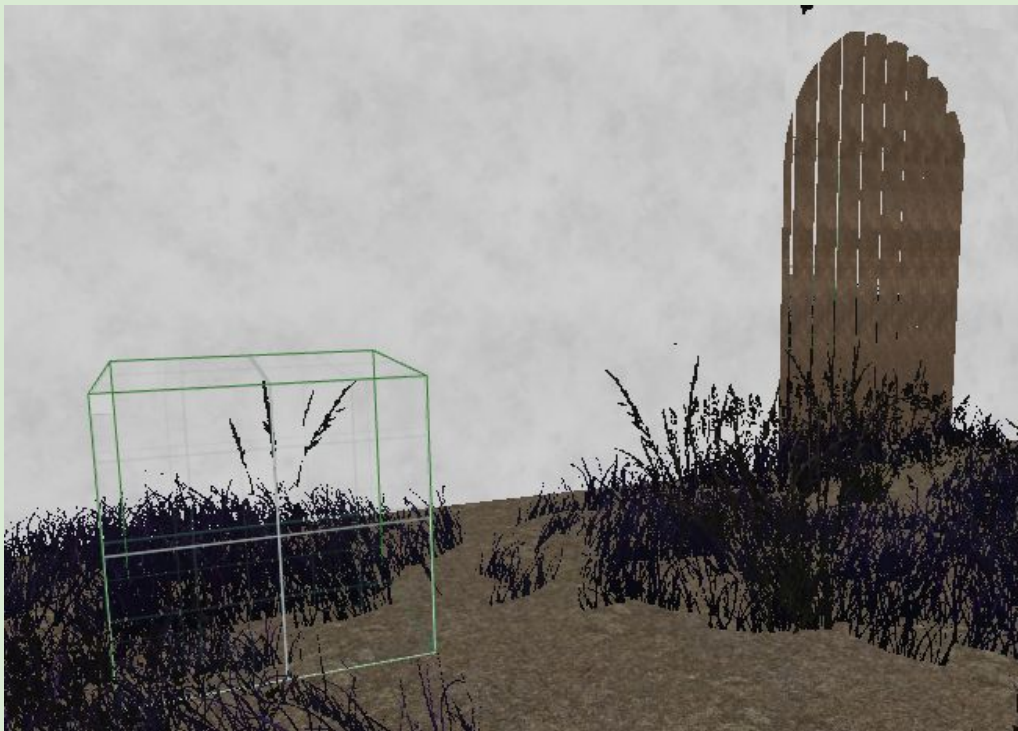
The Player has different animations, which are used for several cases:

| | |
|---|---------------|
|  | Idle |
|  | Moving |
|  | Jumping |
|  | Picking up |
|  | Casting spell |

Gameplay Elements

The main purpose of the game is for the Player to get through the different rooms by solving the puzzles.

Doors



Description

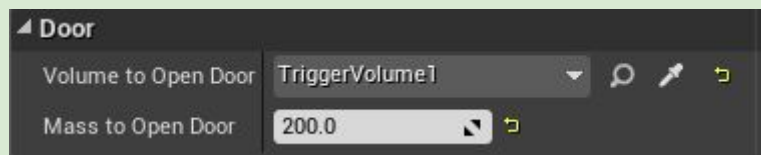
The Doors are the main obstacle for the Player in the game. In order to access the next room, the Player needs to unlock the door. To unlock the door, the Player needs to place a box with a certain mass in the designated area (box hologram). Some doors will require more mass than others, and not all boxes have the same mass. The larger the hologram, the larger the amount of mass it requires. If a box is larger than another box, it means it is also heavier. Once the Player places the correct box(es), the door will open and access to the next room will be granted. It is useful to know that if the Player removes the box after placing it, the door will not remain open but return to its initial closed state.

How to - Doors

Doors are fairly simple to use, navigate to the Door Blueprint (BP_Door) in the Blueprints folder, then drag it to the desired location in the world. Make sure to add a Trigger Volume to the world and assign it to the Door.

Customisation

To customise a Door, if there is one already in the world, select the Door, then select the Door Component.



In the Door Category in the Details Tab, there are two properties to be adjusted:

- Volume to Open Door
 - Category = Door
 - Default value = None
 - Use this to tell the Door which Trigger Volume is used in order for it to be activated
- Mass to Open Door
 - Category = Door
 - Default value = 250.0
 - Use this to tell how much mass the Door requires to be activated

Lights

Data

| | NO Light | 1 Light (No shadows) | 1 Light (Shadows) | 10 Lights (Shadows) |
|--|----------|----------------------|-------------------|---------------------|
| Scene | 4.95 | 5.70 | 6.55 | 6.07 |
| PrePass DDM_AllOpaque | 0.35 | 0.39 | 0.34 | 0.34 |
| ComputeLightGrid | 0.02 | 0.02 | 0.02 | 0.02 |
| BasePass | 0.81 | 0.92 | 0.81 | 0.86 |
| HZB SetupMips Mips:1..9 512x256 | 2.09 | 2.26 | 3.41 | 2.49 |
| ShadowDepths | 0.00 | 0.00 | 0.01 | 0.01 |
| RenderVelocities | 0.02 | 0.02 | 0.02 | 0.02 |
| LightCompositionTasks_PreLighting | 0.51 | 0.59 | 0.54 | 0.69 |
| ClearStencilFromBasePass | 0.01 | 0.01 | 0.01 | 0.01 |
| ClearTranslucentVolumeLighting | 0.05 | 0.05 | 0.04 | 0.05 |
| Lights | 0.00 | 0.17 | 0.22 | 0.41 |
| Direct Lighting | 0.00 | 0.17 | 0.22 | 0.41 |
| NonShadowedLights | 0.00 | 0.17 | 0.00 | 0.07 |
| StandardDeferredLighting | 0.00 | 0.17 | 0.00 | 0.06 |
| BeginRenderingSceneColor | 0.00 | 0.00 | 0.00 | 0.00 |
| IndirectLighting | 0.00 | 0.00 | 0.00 | 0.00 |
| UpdateLPVs | 0.00 | 0.00 | 0.00 | 0.00 |
| ShadowedLights | 0.00 | 0.00 | 0.22 | 0.34 |
| ClearQuad | - | - | 0.02 | 0.02 |
| ShadowProjectionOnOpaque | - | - | 0.03 | 0.02 |
| InjectTranslucentVolume | - | - | 0.00 | 0.00 |
| StandardDeferredLighting | - | - | 0.17 | 0.14 |
| FilterTranslucentVolume 64x64x64 Cascades 2 | 0.32 | 0.36 | 0.31 | 0.31 |

Analysis

The previous table shows four different tests taken to see what the effect of a variety of types of light have on the performance of the game. There was a test with no light in the scene, a test with only one light which did not cast shadows, another test with only one light which did cast shadows, and finally a test with ten lights which all cast shadows. A higher number represents a higher impact on performance.

It became apparent that more lights meant more of an impact on the performance, especially if the lights were casting shadows.

If we take a look at the overall scene:

| | NO Light | 1 Light (No shadows) | 1 Light (Shadows) | 10 Lights (Shadows) |
|-------|----------|----------------------|-------------------|---------------------|
| Scene | 4.95 | 5.70 | 6.55 | 6.07 |

There is a clear increase in the impact on performance when lights are introduced to the scene, with a strange abnormality at the test with the most lights which seems to be affecting performance slightly less than when there are less lights in the scene, though always more than when there are absolutely no lights at all.

Going into more detail, the most obvious category to check was the Lights category:

| | NO Light | 1 Light (No shadows) | 1 Light (Shadows) | 10 Lights (Shadows) |
|--------|----------|----------------------|-------------------|---------------------|
| Lights | 0.00 | 0.17 | 0.22 | 0.41 |

This shows a steady increase in the way the light affects performance, where a scene with no light was not affected at all, up to a scene with 10 lights casting shadows having significantly increased the impact on performance.

From the data found in the tests, it also shows that lights casting shadows have more of an impact on performance than lights that do not cast shadows.

Conclusion

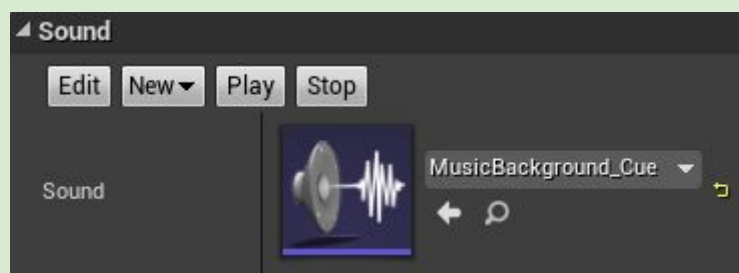
After researching the different effects different types of light have on the overall performance of the game, it is safe to assume that using no lights at all is the best way to improve performance. But if lights have to be used, it is significantly better to use lights which do not cast shadows as that affects performance less than when lights are used that do cast shadows.

Audio System

The Animation and Sound Demo uses a fairly simple sound system.

How to - Sound

In order to get sound into the world, for example some background music, find the Sound Blueprint (BP_Sound) in the Blueprints folder. Drag and drop the Blueprint in any location in the world, then, while still selected, find the Audio Component in the Details Tab. Select the Audio Component, and in the Sound Category add a desired Sound Cue.



If you want to activate the Sound automatically, make sure the Auto Activate is checked in the Activation Category. If the Sound should not automatically be playing, uncheck the same box.

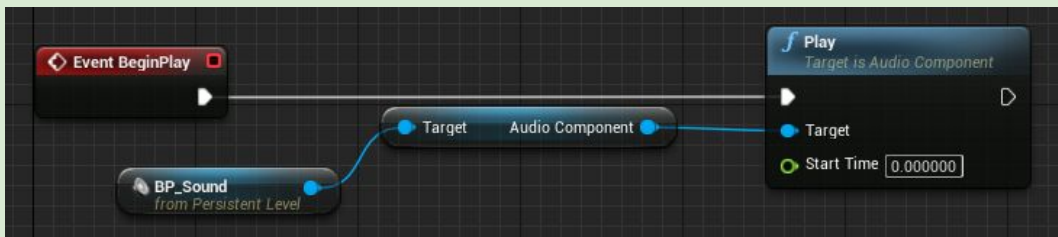


The Animation and Sound Demo uses Trigger Volumes in the world in order to switch between two songs everytime the Player enters a new room. This is done by using the Level Blueprint, which is accessed in the Toolbar Tab, then clicking on Blueprints. Once the sub-menu has opened, open the Level Blueprint by going to the Level Blueprints Category and selecting Open Level Blueprint.

In order to start the Level with a sound playing, make sure to have gone through the previous step of dragging a Sound Blueprint (BP_Sound) into the world and assigning a suitable sound to it.

Then select the Sound Blueprint in the world and without deselecting it, navigate into the Level Blueprint. By now clicking with the right mouse button on any open area in the Event Graph, selecting the Create a Reference to BP_Sound will create a reference to the Sound Blueprint in the world. From there on, all that is left is playing the sound. Drag a blue line from the Sound Blueprint Reference, let go and look for a Play option.

The remaining task to to connect the Event BeginPlay with the Play, which is done by dragging from the arrow slot from one end to the arrow slot on the other end. If there is no Event BeginPlay, simply right click on any open area, search for BeginPlay and it will appear in the Event Graph after selecting it.



Changing sounds is relatively easy as well. All that is needed is a Trigger Volume in the world in an area where it can be reached by the Player. After placing the Trigger Volume in a desired position in the world, while having it selected, navigate to the Level Blueprint once again, then right click on an open space in the Event Graph. Once the menu pops up, select Add Event for Trigger Volume, the open the Collision Category, and choose Add On Actor Begin Overlap. Now using the same Sound Blueprint Reference as before, or by creating a copy of the same Reference and using that, drag the blue line, releasing it on an open area, and search for Stop. Make Sure the arrows are connected with a white line in between the OnActorBeginOverlap(TriggerVolume) Event and the newly created Stop. This will prevent the sound already playing from playing at the same time as the new sound. In order to play a new sound, repeat the previous step when creating and playing the first Sound Blueprint, making sure to add a different suitable Sound Cue. Then connect the Stop and the Play with another white line across the arrows and it will be done.

