3D Virtual World

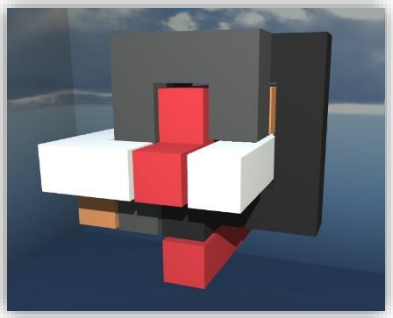
Project Report 2018

Éanna Glavin – G00331702

Galway Mayo Institute of Technology

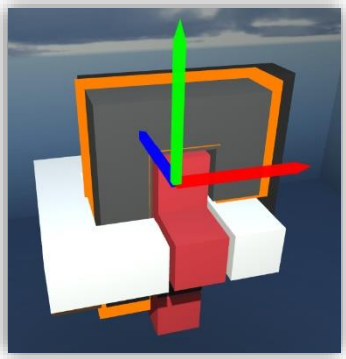
Overview

For the project I wanted to reproduce a level from a game called Interlocked. Interlocked is a game for all ages which I want to stay true to with my own project I would intend that the game would be played by all people. I decided to go with level 13 to give myself a challenge and to see how well I could make the game work. I started of by playing the Interlocked game to get a good feel of the way the game currently works. After playing the game, I realised that I would need to make some blender assets for the interlocked pieces and also find a texture online to use as a skybox which would look a bit nicer than just empty world.



My rendition of the game objects in Interlocked.

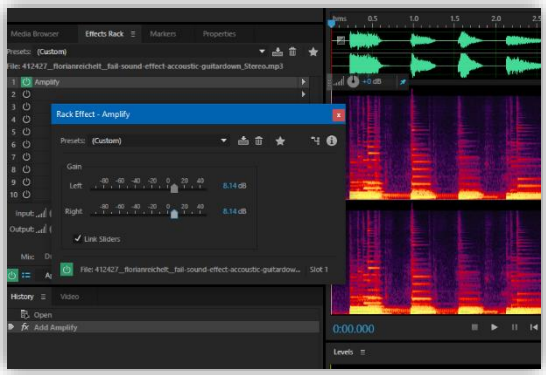
I had to write some C# scripts to detect the movement of the blocks and destroys them when the player gets each piece far enough away from the centre of the map. I also made a Game Piece Manager which keep track of the amount of game pieces that have been destroyed. I decided to change the game slightly and add a timer to the game so after a set time the game will time out. I also had to write a script which allowed the player to click on menu options to choose to play the game and to quit the application. I also made a script which allows the player to look around the game objects. So, if the player presses the space bar the game will recognize this and enable the camera movement within the environment.



In-Game Gizmo.

I spent a good week trying to get a working version with the blocks moving the way I wanted to and after not getting the movement that I wanted I was informed about a script on GitHub from Hidden Monk called Unity3DRuntimeGizmo. This is an asset which I downloaded from GitHub and imported into the Unity assets. This script gave me the functionality I wanted to emulate and worked perfectly so I attached the script to my main camera in the game and then attached a Box Collider to the game pieces which the script uses to find any item in the game you want to make moveable. Also included in the script is the ability to translate the objects by position, rotation, and scale but I only wanted the player to be able to move the objects and so disabled the mapping for the buttons which the player would use to rotate and scale, there were other mapped buttons which I disabled in the script so I would be able to use the other keys for other options in the game. I also used the TextMeshPro script which made the text look nicer in the menu systems for the play and quit buttons.

I found a few song and sound effects which allowed me to give the player a nicer experience while there playing. The play can listen to a song in the start menu and in game and then at the end of the game they will hear a sound of success or failure depending on the outcome of the game. If the player wins the game by moving the pieces away from the centre in time a success sound effect will play but if the time runs out before the player can successfully move the pieces the game will play a failure sound and switch to another game scene. The failure sound was really quiet in game so from what I could tell it had been recorded on an acoustic guitar which from what I can tell wasn't close to the mic during the recording, so to fix this I had to increase the gain on the source file to make it louder. I imported the file into Adobe Audition and used the Amplify Effect in the effect rack to increase the gain to a point before the sound would start clipping.

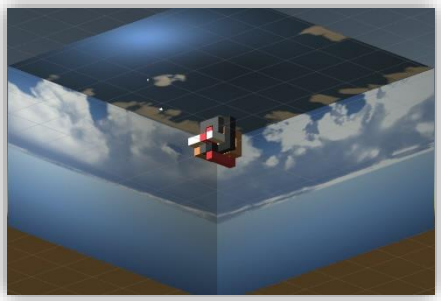


Amplifying the audio.

I recorded a 360-degree rotation of the game pieces with the screen recording software Open Broadcasting Software and by using a ruler and trying to line up the mouse, so it will go in a straight line. I then used Adobe Premiere to edit the video, so it seamlessly tracks around the centre of the game pieces and then allows the video to be looped in the game editor.

Game background video in game main menu.

I found a texture on *kisspng.com* which I used in the game to give the player the belief that they are floating in the air. The texture was one large file which wouldn't work so I used Photoshop to separate the files out in to each section.



Skybox

Conclusions

From attempting this project, I have learnt a lot about the development of games. I learnt how much back and forth there is between the different stacks within this small project. I seen how you should ensure the correctness of your mesh's in Blender and make sure that you clean up and abnormalities when creating the mesh's as it might have weird affects on the way the material interacts in the game world.

I also learnt a bit more about the physics in the game world and the way the game takes input from a mouse. And translates it around the world objects. I had a problem with the game objects which meant when I had both a "InnerConnects Kinematic-Disabled" build and a "InnerConnects DestroyGamePieces-Disabled" game build.

With the "InnerConnects Kinematic-Disabled" build the player wins straight away because the with "Convex Mesh Collider" enabled and "isKinematic Rigidbody" enabled on the game piece's, the objects fly away and fly through the colliders I made to detect when an object should be destroyed. The player won't be able to touch the pieces of the game object either due to the Knematic-Disabled property.

"InnerConnects DestroyGamePieces-Disabled" build allows the player to see what the time out screen is like, as well as this the player can now highlight and move the game pieces but unfortunately the game pieces still float through each another. I believe this is something to do with the colliders I chose but am not so sure would like to come back to this in the future.

To conclude, I learnt a bit about the Unity game engine with this project and the other one we are doing for Virtual & Augmented Reality, and I have started to think more about game development again. I really enjoyed using blender for the project and am starting to feel a bit more comfortable with the program. Although I didn't get everything working I wanted I think I did a good job with the general layout of the objects in the game world. I think that the songs and sound effects I used where good quality and made the outcome of the game pretty nice.

Ps Got it to stop flying off from the centre with help of the other class mates so I made another build and put it in the folder "InnerConnects Gravity Working".

References

Scripts:

Hidden Monks: Unity3DRuntimeTransformGizmo
URL: https://github.com/HiddenMonk/Unity3DRuntimeTransformGizmo

Unity Technologies: Text Mesh Pro
URL: https://assetstore.unity.com/packages/essentials/beta-projects/textmesh-pro-84126

Songs:

ARealDUNDERHEAD: Banjo Shop
URL: https://www.newgrounds.com/audio/listen/814557

DarkHorseOrchestra: Go
URL: https://www.newgrounds.com/audio/listen/791459

Sound Effects:

Florianreichelt: Fail Sound Effect - Accoustic Guitar
URL: https://freesound.org/people/florianreichelt/sounds/412427/

Grunz: success.wav
URL: https://freesound.org/people/grunz/sounds/109662/

Textures:

Anonymous: Skybox Texture mapping Cube mapping - sky cloud
URL: https://www.kisspng.com/png-skybox-texture-mapping-cube-mapping-sky-cloud-920475/

Software:

Blender – Object Creator
URL: https://www.blender.org/

Unity – Game Engine/Compiler
URL: https://unity3d.com/
Docs: https://docs.unity3d.com/ScriptReference/

Open Broadcaster Studio – Screen Cast Recording
URL: https://obsproject.com/

Adobe Photoshop – Photo/Texture Editor
URL: https://www.adobe.com/ie/products/photoshop.html

Adobe Premiere – Video Editor
URL: https://www.adobe.com/ie/products/premiere.html

Adobe Audition – Audio Editor
URL: https://www.adobe.com/ie/products/audition.html

Microsoft Visual Studio – C# Code Editor
URL: https://visualstudio.microsoft.com/