# Advanced Tech Report:Open World

**Ryan Manning**
**18026856**

*University of the West of England*

April 27, 2021

**This report outlines the implementation of creating an open world game**

## 1 Introduction

The problem is to research and develop an streaming open world game with the illusion of a living world around the player.Also implementing the loading and unloading of world chunks as the player moves around the environment. To make the world seem more alive adding NPC's which will have to load and unload in the world and also perform their behavior. As well as this objects in the game had to be loaded into the world by a file for example text, JSON etc.

## 2 Related Work

Some related work to this and other solutions such as some of the games in P'erez, 2015-2016. Some of these are GTA San Andreas by Rockstar North(2004), World of WarCraft by blizzard(2004) and Minecraft by Mojang(2011). All of these games use some system that manages chunks and loads objects in the game in and out around the player. As stated in P'erez, 2015-2016 the way to do this is to break up bits of the world up into different sections and then load in each of these sections when the player is near or can see objects in that chunk. A form of this is also mentioned here Tanel Teinemaa, 2015. As well as this a similar approach also using chunking can be seen hereVogler, 2015. This breaks down chunking and how to chunk using map data which can be really useful in creating a solution to this problem.

Also this solution requires some AI for the NPC's. This Toma´s Plch, 2014 shows some examples of AI in open world games and also evaluates a AI system. This states how AI in many open world games can lack as complicated tasks can take up a lot of processing power which could be used on loading more objects into the world. For the solution that im going to create this is a bit over the top however it does raise questions on how much will NPC's do? how many NPC's are actually going to be in the system.

## 3 Method

First was implementing a file reading system that could read a file and from that load in the correct gameobject into the world.This is done though a switch statement and a for loop that checks through each line of the file and stores everything in a struct. The structure of the text file is the name of the object on the first line and then the position of the object on lines 2,3 and 4 and then rotation on 5,6 and 7. This was done so that objects could be placed in the world at specific points.

Next was actually loading the floor around the player and unloading them when the player walks a certain distance away. This was done by chunking the world up by 20 units and storing the all the newly created gameobjects in a dictionary and moving them to different lists to either generate them or to destroy them. However the chunks aren't actually destroyed as when they need to be unloaded they are set to false and then moved in front of the player.This improves optimisation of the game as only a certain amount of gameobjects will ever be active at one point. To also help optimisation i added in a Ienumerator function that delays the building of chunks these makes sure that chunks don't instantly pop in and only allow them to slowly load in

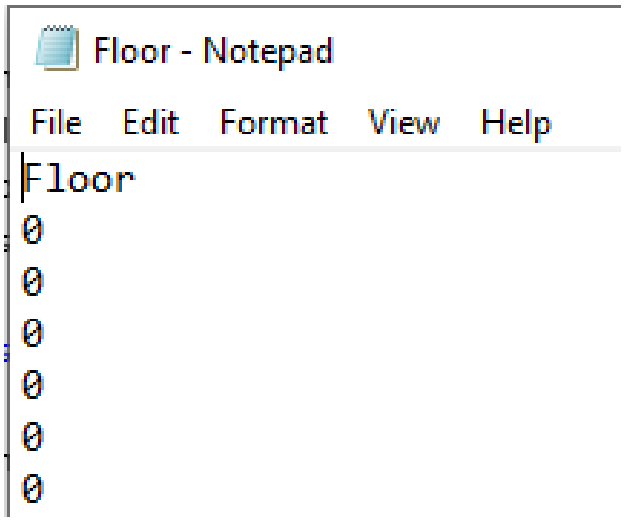


**Figure 1:** *The layout of the text file for a gameobject to be loaded into the scene*

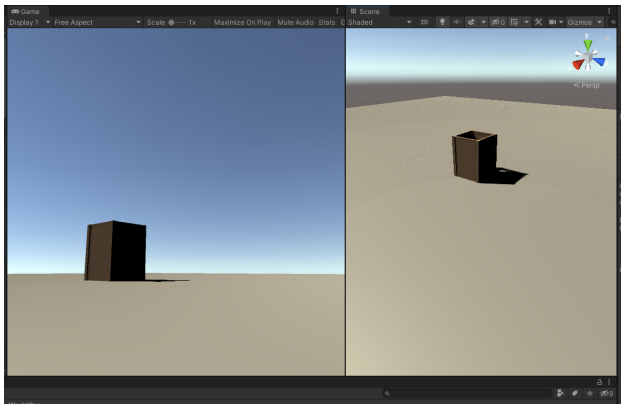

**Figure 2:** *House loaded in to the world*

to the game world.

Then was creating a unique object that is spawned randomly in the world i decided to use a house as this could be used to help turn this system into more of a game.The file loading that was used to load in the floor was also used to load in the house object. However this time instead of using number values to set the position and rotation i wanted to make the house spawn in randomly. To do this i decided that each chunk would have a randomly generated value and based on what that value is the chunk would load in a house if its the first time the condition is met.

Once the house was loading in randomly, next was managing the loading and unloading of the chunks for this i decided to parent the house to the floor it spawns on as the house already spawns from a random number generator in each chunk. This solved the unloading problem as the floor already took care of this. Due to how the floor is coded and that chunks can be moved i had to set up some parameters that allowed the house to be reloaded in the correct position that the player would leave it. This is done by parenting the house to an empty gameobject that will store the house. Then

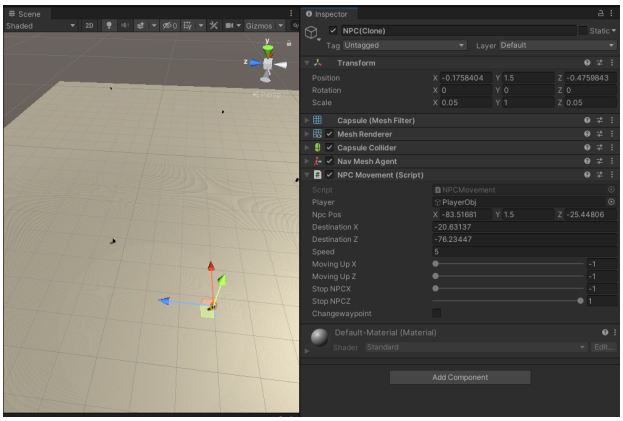

**Figure 3:** *NPC's walking around the game world and some of the variables in the inspector*

by using a raycast to check if there is any floor loaded into the world i can then load and unload the house when the player is in range of it.

Lastly was adding NPC's to the game. I wanted the NPC's to walk around the world randomly and load and unload. For the behavior i decided to make a waypoint system that would randomly generate a number for the x and z value. With this i used simple checks to see which direction the desired position would be in and then move the NPC in that direction. This can be seen in figure 3.Once the NPC had reached the required position i have a ienumerator that checks how long the player has been in the same position if it is over 3 seconds then i reset the NPC's waypoint as this either means that they are at the correct position or they got stuck along the way. For unloading them when they are out of a chunk i used another raycast down to parent the NPC to the floor as then this way the NPC would be able to carry on with different movements after the chunk has been moved. To make sure that the moving of the NPC's didn't cause any problems i added a function that would check to see if the x and z the NPC was moving too was outside of the loaded chunk range if this is the case then the function creates new values for the NPC to move too.

# 4 Evaluation

Overall the implementation of the system that's been created is good as it a streaming open world game with objects that get loaded in from a text file and NPC's that move randomly around the world.Some improvements to the implementation would be changing the actual loading in of the chunks. This is because the way i have done the loading in as stated before is to move chunks around this gives the illusion of a open world game and works fine. However if the environment becomes more unique then this could cause problems as this system will only ever move the same chunks in front of the user. This could cause boring game play as the scenery of the game would be similar all the time.However this

solution is also good for memory as outside of the inital load in to the scene the game would only have to load in unique objects into the game and NPC's as the floor only ever gets moved. Overall the other system work and behave as expected.

# 5 Conclusion

I created a file reading system that loads in objects into the world through a text file based on what is in the file. A chunking system that chunks the world out and loads in objects into these specific chunks as well as unique objects being randomly generated about the world. And some simple NPC's that randomly walk around the player using a way point system to navigate.

# Bibliography

P'erez, Alejandro Juan (2015-2016). "Open World Streaming:Automatic memory management in open world games without loading screens." In: *Open World Streaming:Automatic memory management in open world games without loading screens*. Vol. 1. 1. Universitat Politècnica de València, pp. 9–19 31–38.

Tanel Teinemaa Till Riemer, Noor Shaker (2015). "2D infinite world generation". In: *A Procedural Approach for Infinite Deterministic 2D Grid-Based World Generation*. Vol. 1. 1. University of Copenhagen, pp. 1–5.

Toma´s Plch Matej Marko, Petr Ondra´cek (2014). "An AI System for Large Open Virtual World". In: *An AI System for Large Open Virtual World*. Vol. 1. 1. Charles University in Prague, Faculty of Mathematics and Physics, pp. 2–6.

Vogler, Benedikt S. (2015). "Chunking in a isometric game". In: *Implementing an Open World Isometric Game Engine Based on a Block World Design*. Vol. 1. 1. gamasutra, All.