

A Simpler Approach to 3D Scene Design for Civil Engineering

- Dr. Louis has worked with previous capstone teams to develop simulations of several civil engineering processes in VR. However, creating these simulations typically requires extensive modeling, programming, and editing in development environments.
- As a result, the aim of this project is to create an application that reads text commands and translates them into an animated 3D environment in Unity, reducing the development time.
- This scene and animation tool aims to decrease the cost of creating simulations of potential or current construction in civil engineering processes. The creation of these simulations in 3D is costly and largely dependent on individual skill and effort in simulation.
- Enabling faster development of these simulations will improve the overall flow of documenting or proposing these processes, while also providing a viewing interface.



Text-based Scene Building

Creating animated 3D environments from written instructions using Unity.

Project Intent

The text based scene building program utilizes Unity's engine to create and display 3D scenes dynamically. Users provide instructions in a text file to control assets imported into unity. Unity will then generate the animation, while simultaneously allowing viewers to freely explore the scene.

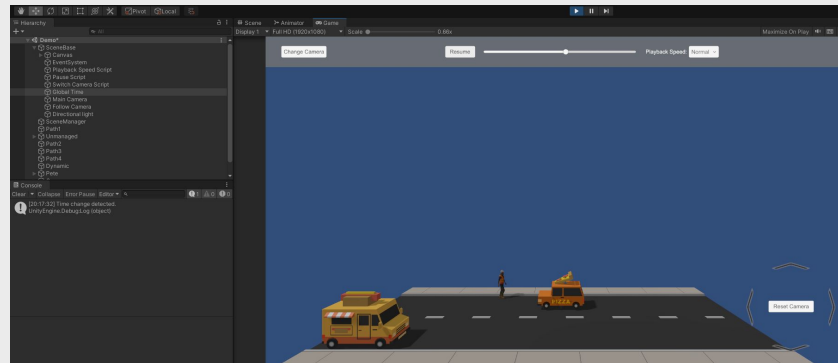
How it Works

First the text instruction file is parsed, gathering time stamps, references, and creating the list of commands. Each command has its own structure, taking in input parameters to target an object and tell it what to do. After the text file is read by the program, it will execute each action line by line, stringing them together to create the desired animation.

Example Text Instructions

```
1  INCLUDE Reference.txt
2
3  FUNCTION test
4  CREATE Pete Worker -10 0 -2
5  SETOBJCELL Pete Transform.scale 2 2 2
6  CREATE Car Vehicles.PizzaCar 6 0 -7
7  SETOBJCELL Car Transform.rotation.y 90
8  SETOBJCELL Pete Animator WorkOnDevice
9  TIME 3
10 SETOBJCELL Pete Animator Walk
11 TIME 8
12 SETOBJCELL Pete Animator Empty
13 SETOBJCELL Pete Transform.rotation.y 90
14 TIME 9
15 DESTROY Pete
16 MOVE Car Path3 2
17 TIME 11
18 DESTROY Car
19 END test
```

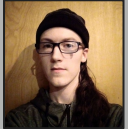
User Interface (Editor View)



Project Team

Darren Garnett: darrenqarnett_20@gmail.com

- Text Parsing & Command Functionality
- CS Simulation and Game Programming Major



Tun Aung Thuang: lobythuang@gmail.com

- Object Initializing & Animation Rendering
- CS Artificial Intelligence Major



Jack Tang: jacktango403@gmail.com

- Animation & Users Manual
- CS Data Analysis Major



Morel Kopcho: mkopcho12521@gmail.com

- UI Design & Implementation
- CS Artificial Intelligence Major / Statistics Minor



Dr. Joseph Louis - Project Partner

- Associate professor of civil engineering at OSU

