

AR Sandbox for Construction Planning

Traffic Simulation Feature Update

Winter Term Progress Update

Team Augmented Construction Education

McKenzie Gray, Jonah Spencer, Adam Sunderman

April 15, 2019

Abstract

This document describes the progress our team has made over Fall and Winter terms on the AR Sandbox for Construction Planning. Overall project goals will be discussed first, followed by a demonstration and explanation in current beta level functionality of the new traffic simulation feature.

Contents

1	Purpose	3
2	Goals	3
3	Current State of Development	3
3.1	Winter Term Changes	4
4	Outstanding Task List	4
4.1	Network Creation and Loading	4
4.2	Vuforia Integration	4
4.3	Simulation Visuals and Logic	4
4.4	Upgrades to Existing AR Sandbox	5
5	Problems	5
6	Beta Functionality Demonstration	6
7	Traffic Simulation Network Creation Classes	12

1. Purpose

The purpose of the AR Sandbox is mainly educational and meant as a means to visualize large-scale construction projects. The AR Sandbox offers a new and unique way for students to visualize the topics about which they learn in class with real-time feedback supported by various augmented reality modules.

2. Goals

The primary goal of this project is to introduce a traffic simulation mode, which includes AR marker functionality, to the AR Sandbox. The traffic simulation mode will project a visualization of a traffic simulation. Markers will be used to interact with the simulation in various ways, such as turning sections of road into work zones or introducing traffic lights to an intersection. After the scene has been modified by a marker, the traffic simulation will automatically update accordingly.

Another important goal of this project is to improve the existing functionality of the AR Sandbox. Several aspects of the current AR Sandbox are not working properly, while other aspects need updates to make them more user-friendly. These improvements are described in the Problems section.

Another feature to be added to the AR Sandbox is the ability to save and load scenes from any mode. For example, in depth mode, loading a previously-saved scene would project the heights from the saved scene onto the sand, thus indicating where sand must be placed in order to match the previous topography. In traffic simulation mode, loading a scene would result in the same road network as that of the saved scene.

3. Current State of Development

As the traffic simulation update is building on an existing project, the AR Sandbox is already fully-functional and supports three user modes: Depth Mode, Design Mode, and Cut & Fill Mode. In Depth Mode, various colors are projected into the sandbox representing the current height of the sand. In Design Mode, a segment of road can be constructed by manipulating a variable number of control points. [1] Once the road segment has been created, it will be displayed using different colors representing whether sand needs to be added or removed in order for the road to be at the appropriate height. Finally, in Cut & Fill Mode, a table collects and displays data from Design Mode that indicates the area and volume of cut and fill to be performed for the segment of road.

The AR Sandbox currently uses a Microsoft Kinect V2 sensor for depth sensing and an Optoma ST50 projector for displaying the program window in the sandbox. The software is built on the Unity game engine, and the Kinect SDK is used to read depth data from the Kinect sensor. Additionally, a Logitech C920 webcam has been introduced to the AR Sandbox for use with Vuforia.

3.1. Winter Term Changes

The majority of the code involved in creating and loading a simulation network into Unity is complete but needs a few minor changes. Simulation road networks and terrains are building with no errors but still need to be properly textured, scaled, and aligned in the scene.

Vuforia interactions are ready for use in some places but need to be integrated with the rest of the system. The current user interface will be changed to use Vuforia's Virtual Buttons and marker interaction will be introduced once simulations are fully-functional.

The simulation logic and visuals that will run on networks and terrains is one of the more challenging tasks and requires that network creation be stable before implementation. This feature is in progress but cannot yet be demonstrated for this document.

The original AR Sandbox depth mode has been upgraded. The feature is now able to display contour lines on the elevation projection. As the projection changes color gradients, and therefore height, a black line is drawn between the change. This helps a user more precisely determine height information in places where there are greater rates of change in elevation.

4. Outstanding Task List

4.1. Network Creation and Loading

- 1) Adjust textures and materials for models.
- 2) Ensure all models are properly scaled, rotated, and translated.
- 3) Finish the building model procedural algorithm to add buildings to a simulation.
- 4) Add scene objects with interaction. Traffic lights, stop signs, etc.
- 5) Add scene objects with no interactions. Benches, trash cans, street lights, etc.

4.2. Vuforia Integration

- 1) Connect user interface to Vuforia using Virtual Buttons.
- 2) Apply marker interaction to traffic simulation scene objects.

4.3. Simulation Visuals and Logic

- 1) Create a client-server based class for interacting with Traci to get simulation data.
- 2) Create models for different vehicle representations.
- 3) Define the specific macroscopic logic for simulation color coding.

- 4) Create a user interface to switch between the macroscopic and microscopic simulation modes.
- 5) Develop algorithm to determine passing lanes for simulation vehicles. (Line-Of-Sight)

4.4. Upgrades to Existing AR Sandbox

- 1) Saving and loading of terrains.
- 2) Depth mode calibration improvement.
- 3) Fix exception errors in the old 'design' mode.

5. Problems

Remounting the projector and depth camera: In order to create a more permanent and appealing mount for the projector and camera we researched a few mounting options. For the Kinect depth camera we will just drill a couple holes in the plastic frame and attach it with screws to the AR Sandbox. Mounts are available, but not necessary. The projector mount needed to be low-profile and have adjustments for rotation and pitch. After looking at GoPro action camera mounts and projector mounts we ended up using a mount designed for trail cameras used to photograph wildlife.

Replacing the sand in the AR Sandbox: The sand in the AR Sandbox was determined to be too glassy by the client and he wished to find a more suitable sand that didn't reflect light as well. Research found that most sands other than play sand (which the AR Sandbox has) contain carcinogens. We have decided that it is better to deal with the reflection from the projector rather than risk of harm to the users of the AR Sandbox.

Cleaning up previous work on the AR Sandbox: Several aspects of the current AR Sandbox are not working properly, while other aspects need updates to make them more user-friendly. Some known needed improvements include:

- The settings for minimum and maximum height in calibration mode are very sensitive and it can be very difficult to get them to the right setting.
- Switching modes via the menu does not always work, and keyboard shortcuts must be used instead.
- In Design Mode, adding a control point throws an exception and causes one of the endpoints to stop functioning properly. There are likely many small bugs like this one that we will need to fix

Making the AR Sandbox usable during development: To make the AR Sandbox usable by students and professors during development of new features we will build a version of the app in its current state. This version will be named ARSandbox2018-Stable and will be available as source code on GitHub in it's current state as well.

Using Vuforia with the Kinect camera: For the first several weeks of development, Vuforia was unable to track images consistently. We determined early on that the Kinect was somehow distorting the image, making it next to impossible for Vuforia's computer vision to recognize images. After modifying settings within Unity, Vuforia, and the Kinect API, we were unable to resolve the problem. Instead, we installed a new webcam on the AR Sandbox for use solely with Vuforia.

6. Beta Functionality Demonstration

The following images and captions give a detailed example of creating a traffic simulation network with the AR Sandbox. First, a real world location will be selected to be used as the simulation model and it's data will be downloaded from Open Street Map. Next, the locations roads, buildings, and a terrain will be built as Unity GameObjects and loaded into the game engine where simulations will be run with Vuforia interaction.

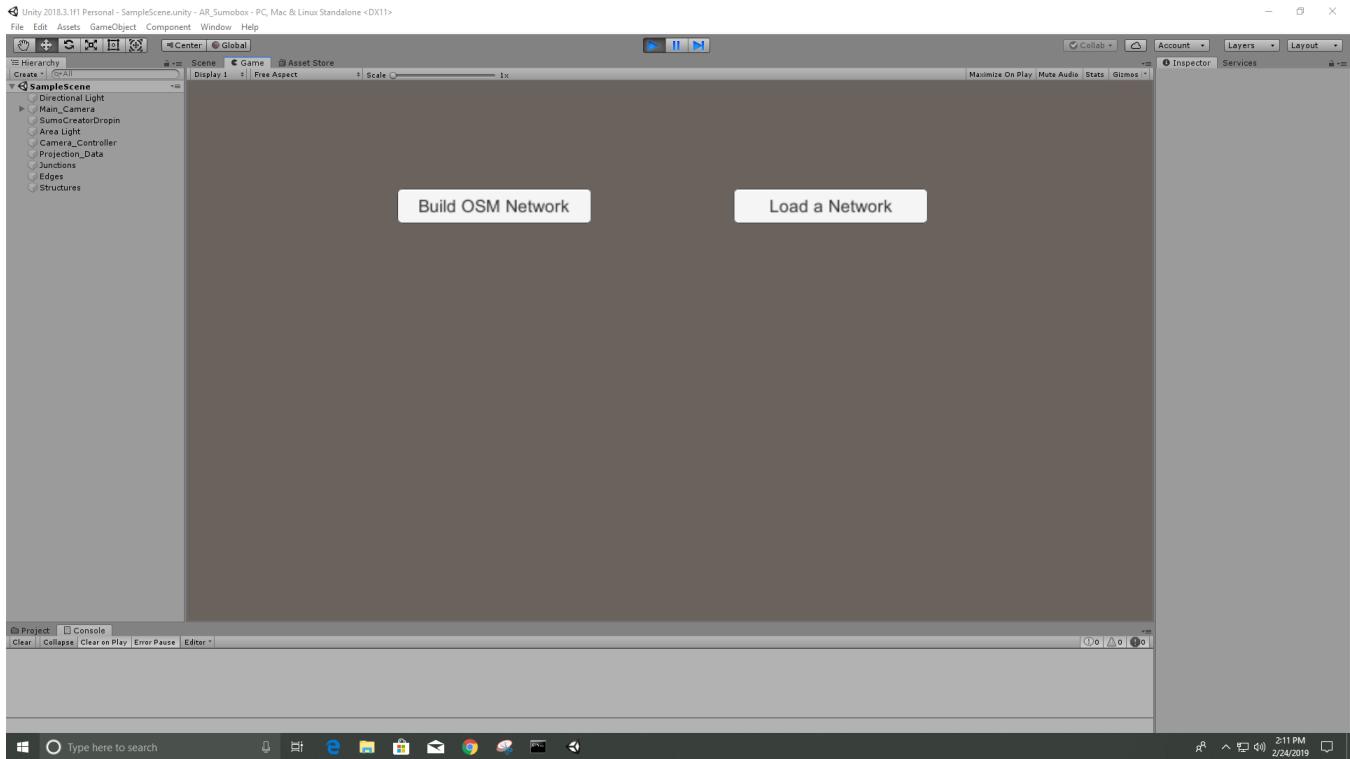


Fig. 1: The temporary menu screen for development and selecting the option "Build OSM Network".

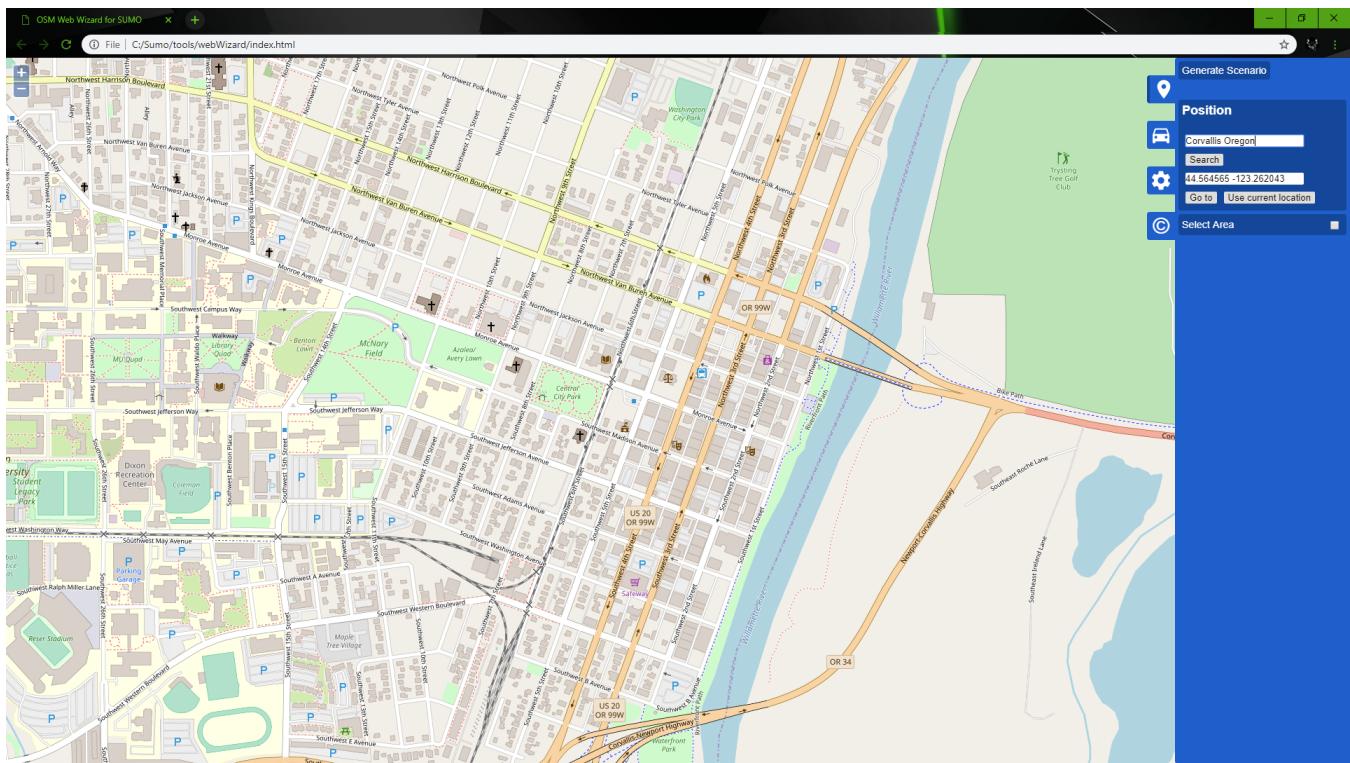


Fig. 2: The import screen after typing "Corvallis Oregon" in the Position text box

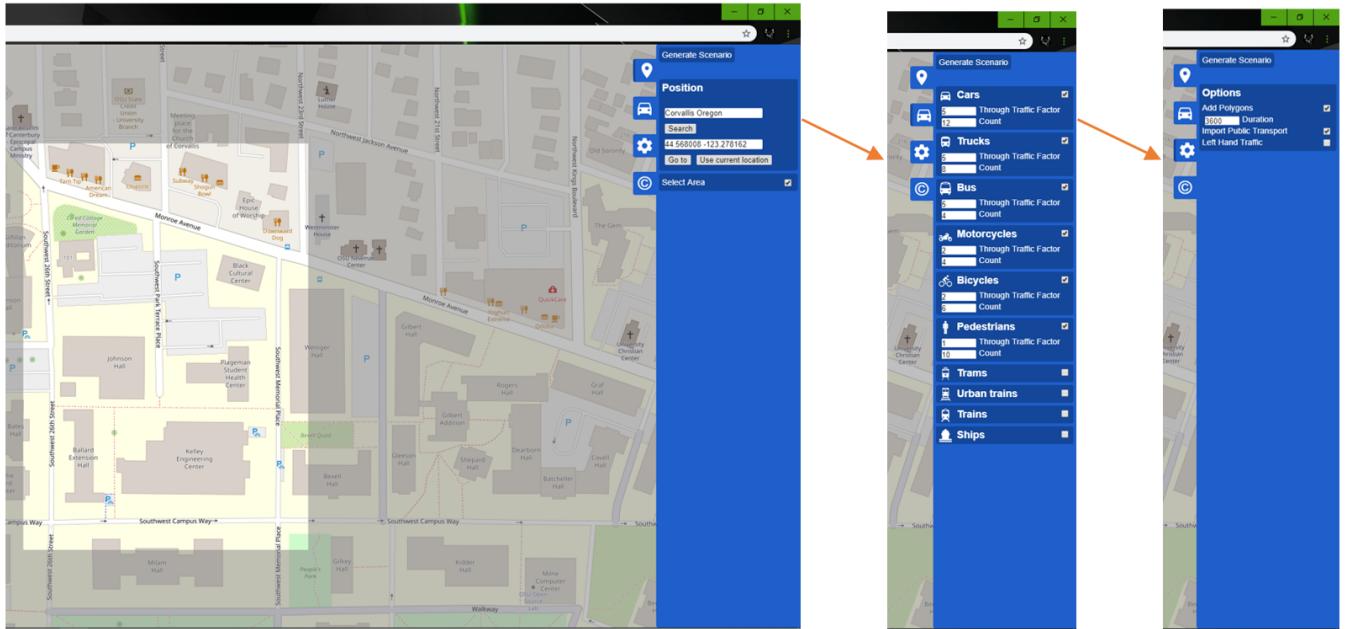


Fig. 3: Map data is converting from Open Street Map format to SUMO required format.

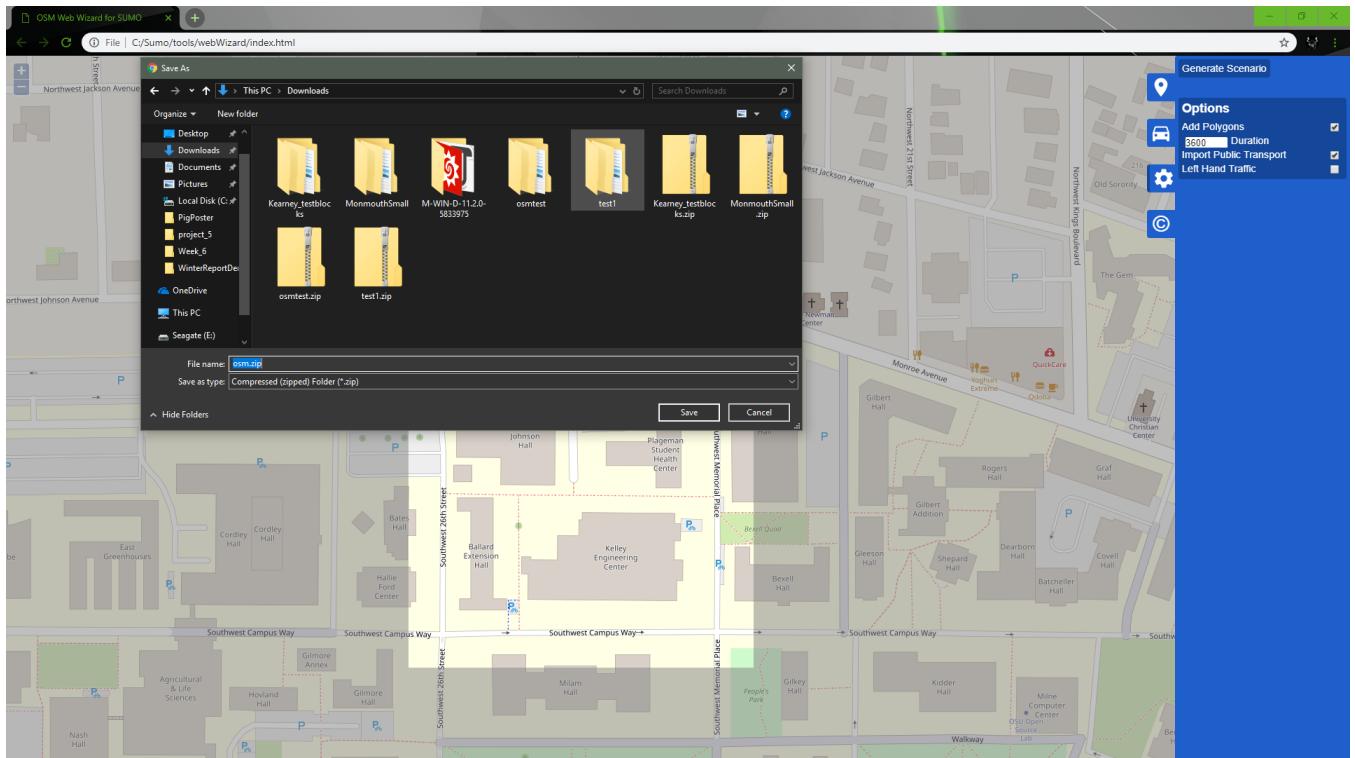


Fig. 4: A dialog is opened which allows the user to save the new network as a zip file.

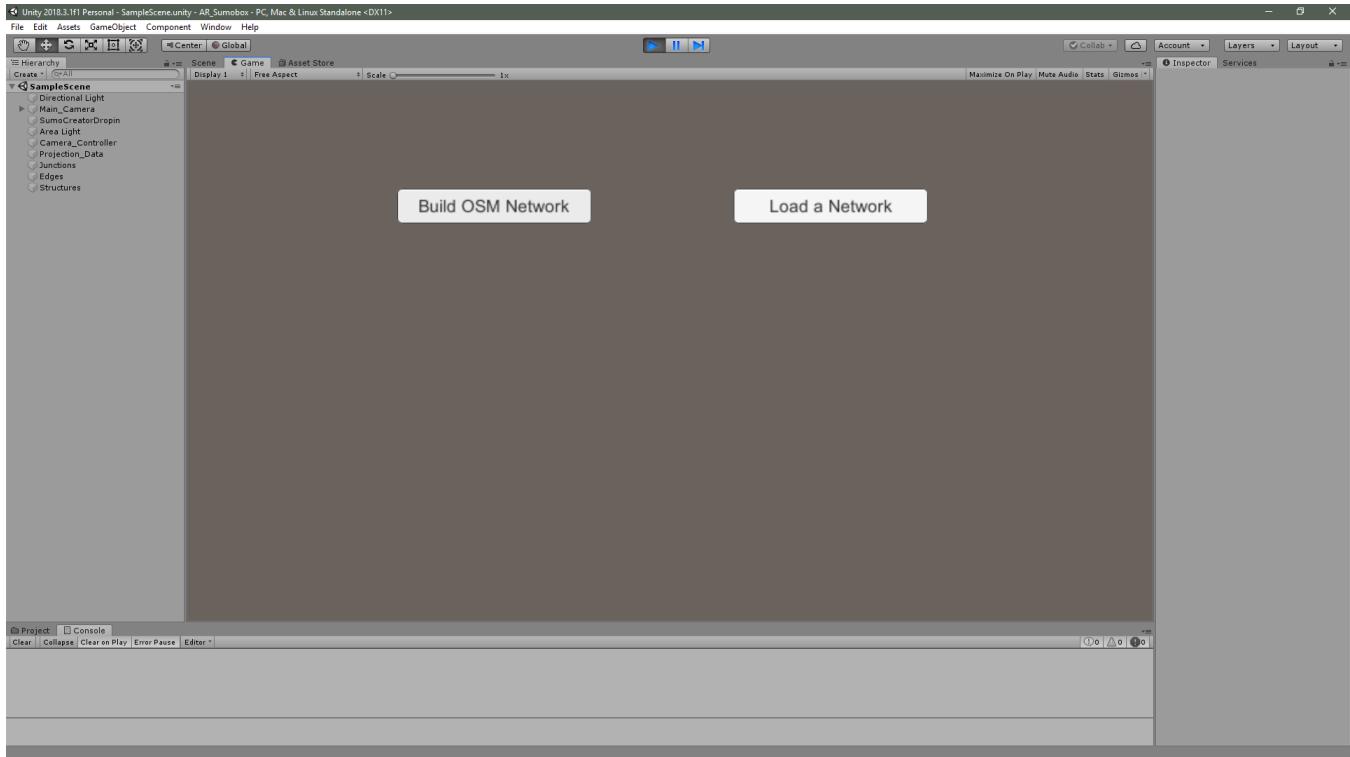


Fig. 5: Back to the main menu and select the option "Load a Network".

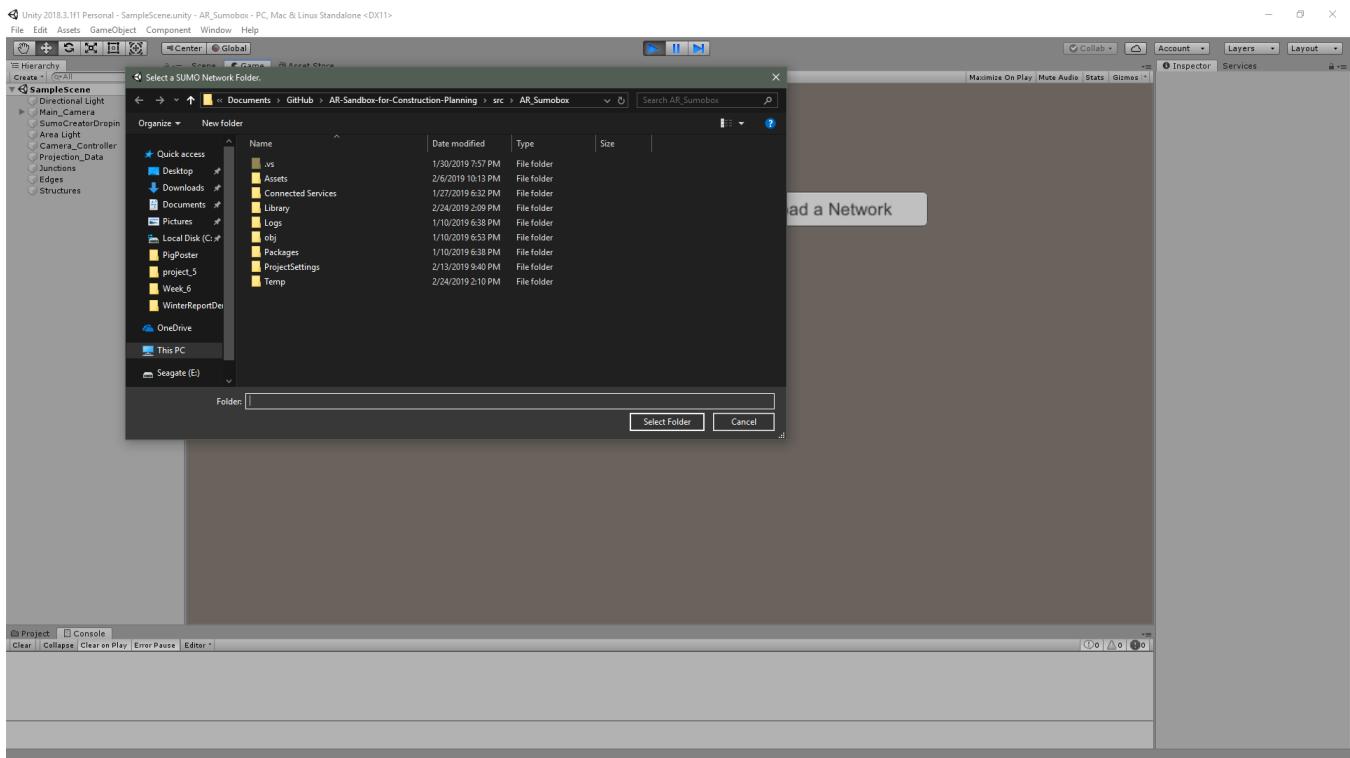


Fig. 6: A dialog is again opened so the user can choose a network to load into Unity.

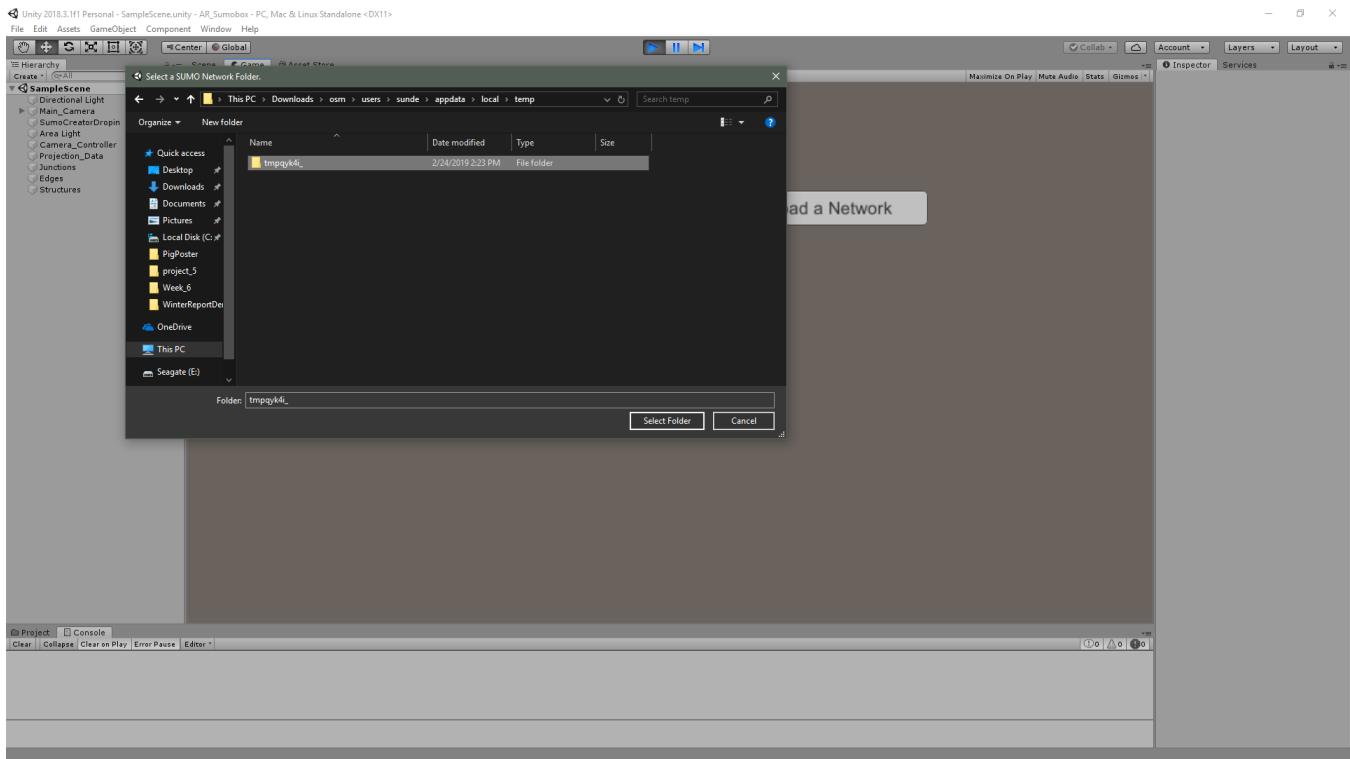


Fig. 7: We navigate to our networks root folder and choose "Select Folder".

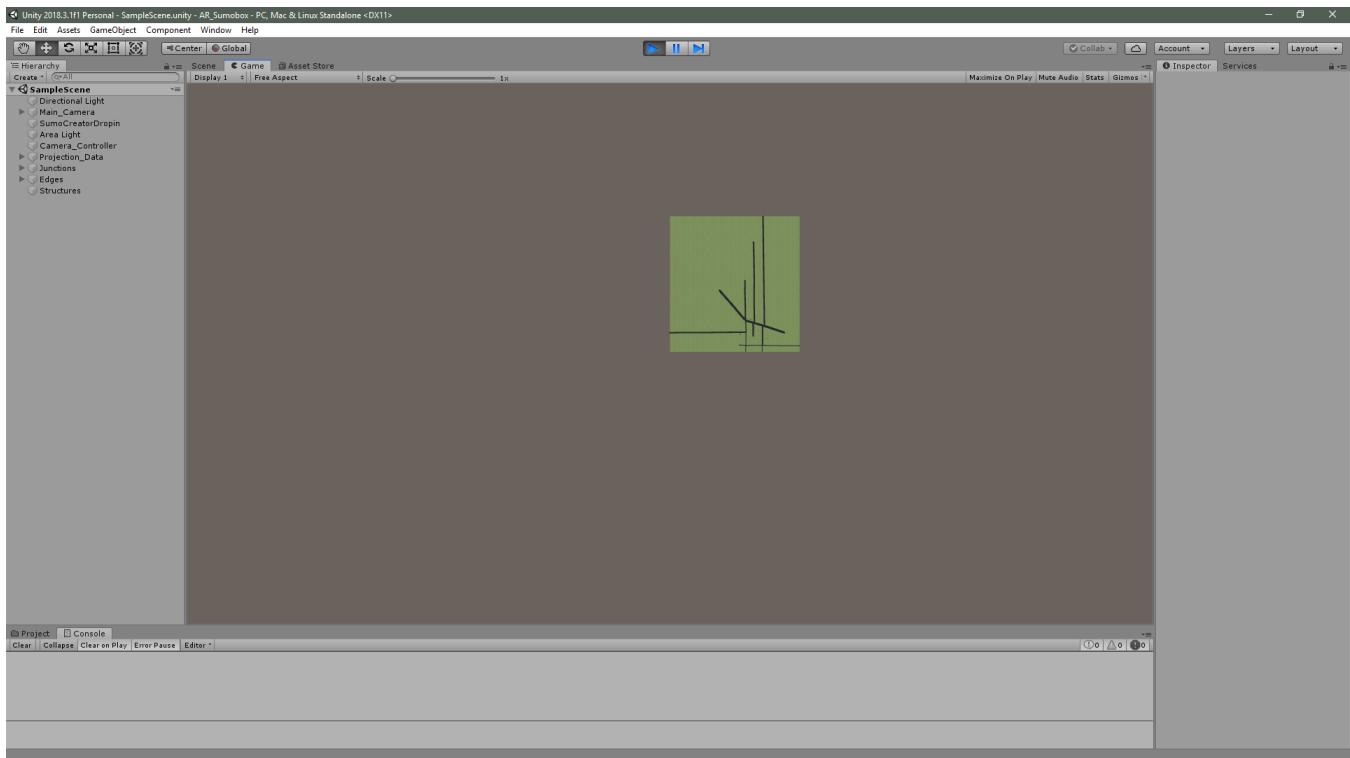


Fig. 8: The network is loaded and we are presented a topdown view.

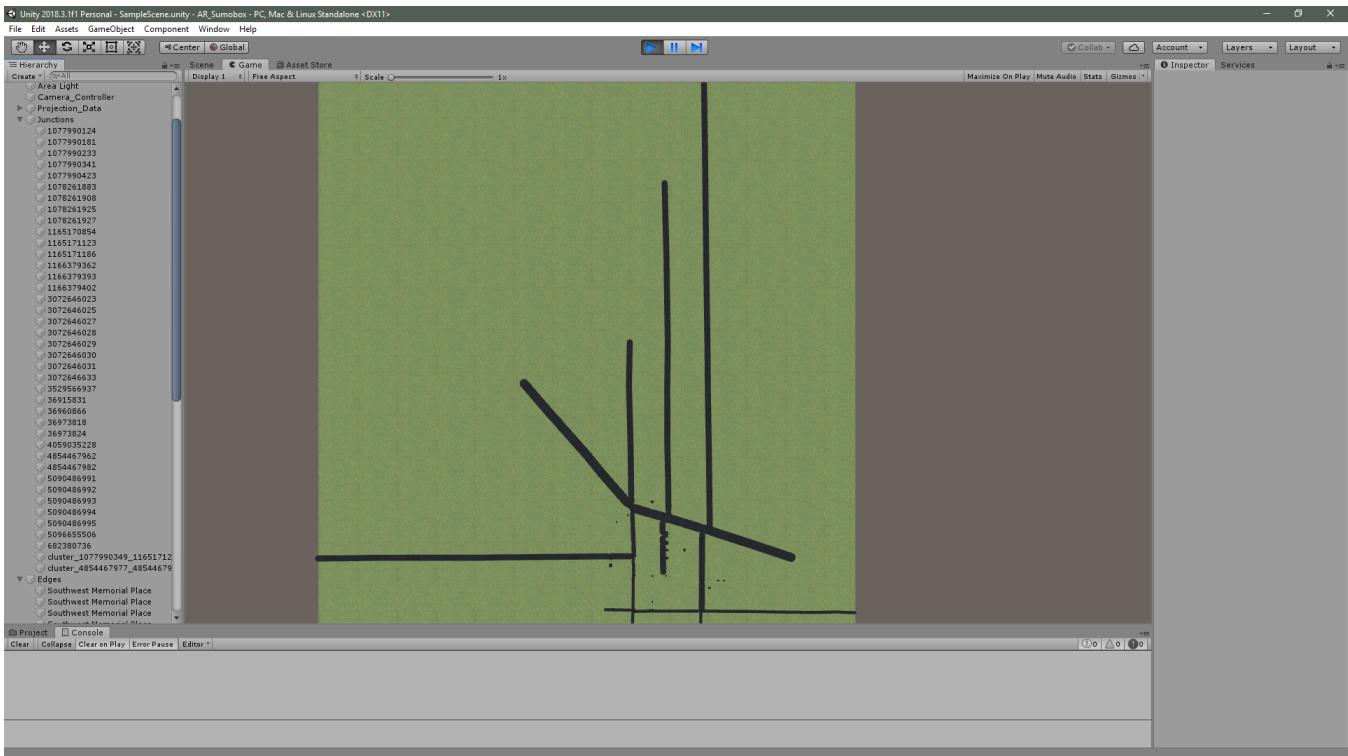


Fig. 9: Another view of the network with the list of its "Junctions" expanded for reference.

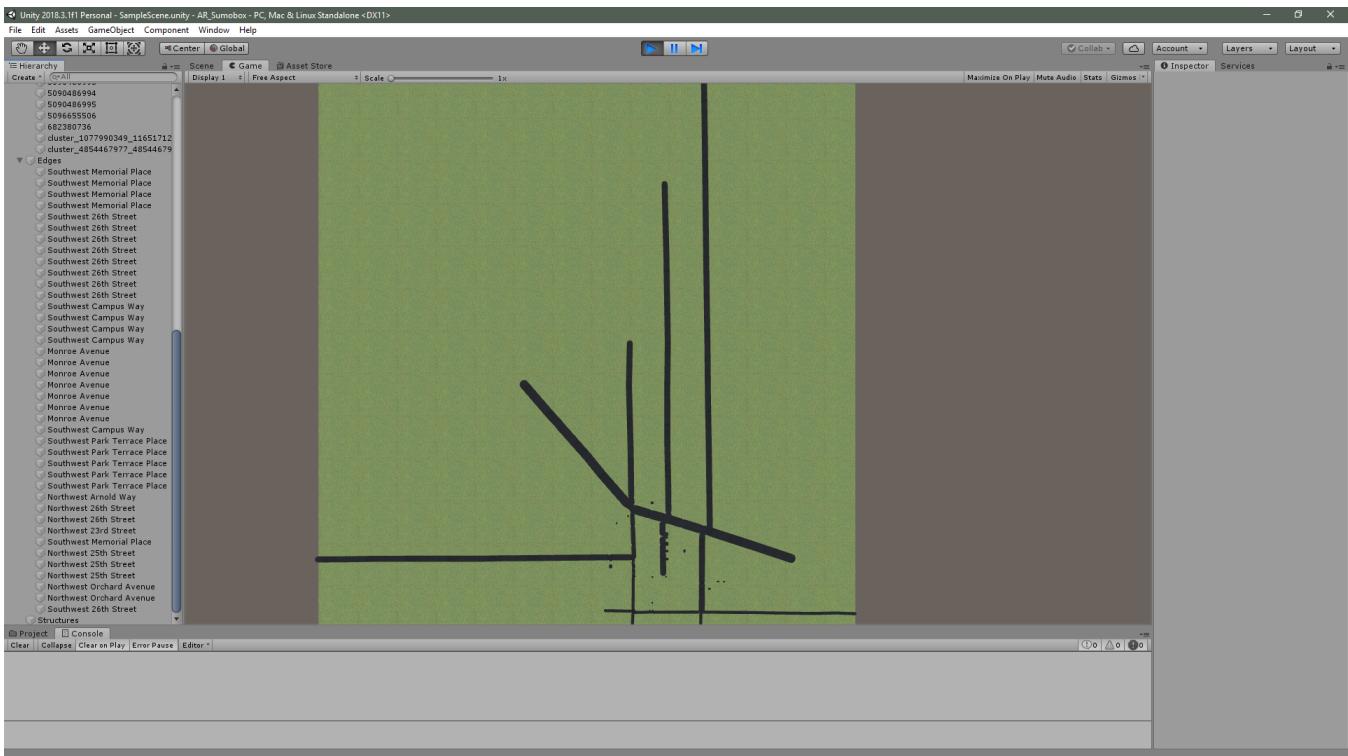


Fig. 10: Another view of the network with the list of its "Edges" expanded for reference.

7. Traffic Simulation Network Creation Classes

Listing 1 SumoCreator.cs

```
// SumoCreator class is used for creating Open Street Map networks with
// SUMO's
// OSM Web Wizard and reading SUMO generated files that describe a networks
// logic
// and layout.

public class SumoCreator : MonoBehaviour
{
    // ProjectionData Parent GameObject and script.
    private GameObject Projection_Data_GO;

    // Junctions Parent GameObject and script.
    private GameObject Junctions_GO;

    // Edges Parent GameObject and script.
    private GameObject Edges_GO;

    // Structures Parent GameObject and script.
    private GameObject Structures_GO;

    // Starts off the SumoCreator
    private void Start()
    {
        Projection_Data_GO = GameObject.Find("Projection_Data");
        Junctions_GO = GameObject.Find("Junctions");
        Edges_GO = GameObject.Find("Edges");
        Structures_GO = GameObject.Find("Structures");
    }

    // Builds the network pieces by reading a Sumo network file.
    // Reads through a network and saves all the network data to the
    // handling class.
    // There are classes for ProjectionData, Edge, and Junction.
    // After all data is read in each class builds its own shapes.
```

```
private void BuildNetwork(string file)
{
    ...
}

// Builds the structures of the given file
private void BuildStructures(string file)
{
    ...
}

// Open Up OSMWebWizard and let the user build a real network.
// The user will save the new network to a zipfile when done.
// The processes remain open so the user can build multiple network at
// once.
public void GenerateOsmNetwork()
{
    ...
}

// Go through all network description files and build the network into
// Unity.
// Most files will be passed over at this point but there are some
// handles left
// in case we decide we need access to them later.
public void LoadNetwork()
{
    ...
}
}
```

References

- [1] R. Petroff, A. Soltesz, and M. Sprouse, “Cs capstone design document,” <https://github.com/soltesza/AR-Sandbox-for-Construction-Planning>, 2017.