AR Sandbox Milestone 2

Team 7

Zach Bishop, Tyler Rivas, Andrew White, and Ryan Stachura

October 27, 2020

Table of Contents

1. Background

This will be the third year that students from the Oregon State University Computer Science and Mechanical Engineering departments will work on the Augmented Reality Sandbox Project. In the last two years, students have worked on implementing the core features of the sandbox. They created a physical sandbox that has a depth sensor and a kinect sensor attached to it. This hardware, combined with software that computer science students in the past have written, resulted in an augmented reality sandbox where terrain and mountains are augmented onto the physical sand where people can view it and interact with it. If the sand is moved or altered, the terrain will change allowing you to see live updates of what happens with different environments. In the first year, students physically set up the sandbox, obtained the sand, built the components of the unique rolling contraption, and used this time to set up the motion-activated camera and depth sensor. Essentially, during this time, the physical hardware devices that were required to get the project up and running were installed. During the second year, students implemented a lot of core software functionality. Students used C# and the Unity game engine to interact with the sandbox and the physical devices that the students had set up during the previous year. The second-year students created the software that allows people to actually interact with the sand and to see visual landscape changes displayed on the sand. The students left a codebase on GitHub and left the physical sandbox intact and ready to use. Aside from where the previous students left the project, this type of project has been around for a few years now. Multiple institutions across the country have implemented similar AR sandboxes. It is worth noting that the implementations that the students worked on the last two years were not necessarily unique and they were not the first people to create this type of project.

As far as problems go, there currently aren't any problems specifically relating to the sandbox. There are new features that our team is going to have to problem solve and work through, but there are no issues that require resolution that we are aware of. As previously mentioned, the system used to implement core features is the Unity game engine, along with an SDK called Vuforia. Vuforia is a software development kit that will allow us to create augmented reality applications and features. Vurforia uses computer vision technology to track images and 3D projects in realtime. The pertinent history for this project is how everything works together and how the features of the current AR sandbox were implemented. With that being said, the first challenge our team faces is to understand how all of the code from the codebase that we were given works together. Understanding each script and each file that is in the codebase is going to be essential to implementing more advanced features.

2. Vision Statement

We have two major goals over the course of this project. The first goal is to implement a feature that allows people using the AR sandbox to load and save terrains to the sandbox. More specifically, this means that we want the sandbox to interpret a piece of land data that we give it and display that piece of terrain onto the sand. With this technology we would be able to pick different terrains to demonstrate with, and we would be able to run more simulations. The second goal of this project is to implement a disaster response simulation. Given data from FEMA, which would tell us the estimated damage that a particular catastrophic event would have on a specific city or location, we would be able to take this data and implement an augmented simulation on the sand that would show where debris and destruction is in the city. We will hopefully have game pieces that the users of the sandbox can interact with. These pieces are going to represent resources to help the location that endured the catastrophic event. Depending on where the game pieces are placed, the user will see an expected duration of days representing how long the city would take to recover. This will allow users to run different simulations testing where the prime location in the city is to put resources.

2.1 Growth and Value Hypothesis

2.1.1 Growth Hypothesis

Users will adopt the sandbox when they need to represent simulation data in a more tangible way, helping engineers get better insight into the project they are a part of.

2.1.2 Value Hypothesis

Users will gain valuable feedback from running simulations, and they will be able to see conclusive situational data that can be used to better understand disaster relief and terrain manipulation.

2.2 Functional and Non-Functional Requirements

2.2.1 Functional Requirements

- Users can run simulations of traffic, road construction, and other civil engineering tasks.
- Users can load terrain data, from common GIS formats, into the system, and the system will show the user how to sculpt the sand-terrain into the desired terrain shape.
- Users can also save the terrain data that they have made in the sandbox into common GIS formats.
- The user can use special game pieces that each represent different simulated objects on the sandbox. These game pieces can represent a variety of things from road closures to debris clearing equipment.
- The system should not be able to generate its own terrain.

2.2.2 Non-Functional Requirements

- The system must be able to recognize the status of the sandbox terrain and simulate changes to it.
- The system must be able to read common GIS formats, and also export modified sandbox terrain data to them.
- The system needs to make use of computer vision to recognize game pieces that are placed into the sandbox.

3. Prioritized Project Constraints

3.1 Time

The main time constraint we will face is our capstone project length. It is a three term project, after which we will no longer be able to work on it. If we are unable to complete the project by the end, our portion of the project may be scrapped or pushed back to the next group of capstone students. We also have a limit to the amount of time per week we can work on the project. Each of us has classes outside of the capstone project which limits how much we can do in a given period of time. In addition to the above constraints we also have limited time periods to test features with the sandbox due to the current pandemic.

3.2 Resources

So far, resources do not seem to be a constraint. We have access to the sandbox built by previous students who worked on the project. All of the equipment required for the project has been provided already. All members of the team need to be able to use the Unity3d game engine as well as the Vuforia framework for computer vision. As not all members of the team are familiar with Vuforia, this may incur additional time constraints.

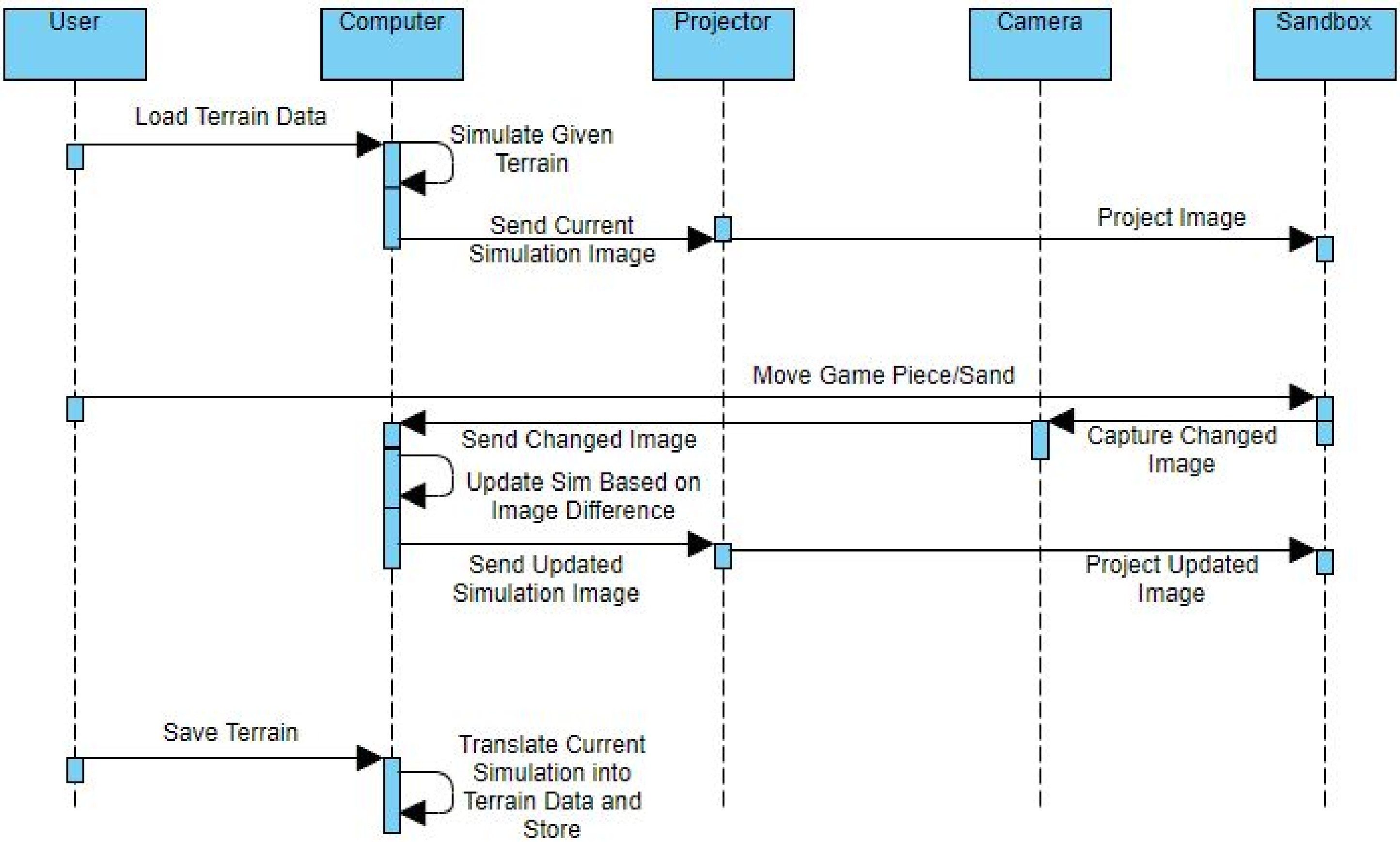
3.3 Scope

The project has a straightforward goal, allowing users to physically interact with the sandbox to produce effects on the simulation, and load/save terrain data using common GIS formats. This includes the use of computer vision to recognize game pieces, and their placement, to alter the outcome of the simulation. For example, a game piece that is defined as a bridge could be used to give the user an alternate way of designing a roadway through terrains. Game pieces could also be used to represent debris clearing equipment, so that a simulation of an earthquake could be made to assist in disaster preparation, giving information such as road blockage clearing times depending on the location of the aforementioned debris clearing equipment.

4. Scope

4.1 Process Flows

The main goal of this project is to allow users to run simulations and interact physically through the sandbox with the simulation to understand the potential effects of different actions. An example of this is simulating a terrain hit by a natural disaster. The user could move physical pieces representing resources to discover different distributions and how they affect the speed of recovery. There are a few subgoals that needed to be achieved to make this possible. These include loading and saving terrain data, registering game pieces, defining the effects of game pieces, recognizing when game pieces are moved, and adjusting the simulation depending on the placement of game pieces. The following is a UML sequence diagram demonstrating the flow of actions that should follow each user action.



4.2 User Stories

1. As a user, I need to be able to quickly set up the physical system and easily use the software interface so that I can focus more time on learning valuable information by simulating different scenarios.

2. As an engineer, I need to load and save terrain data, alter the sand to represent the loaded terrain data, move game pieces, and test different scenarios so that I can gain a better understanding of the effect of different decisions.
3. As a developer, I need the system to be easily extensible with thorough documentation so that I quickly understand the system to add more features and allow for different types of simulations.
4. As the seller of the product, I need to show the potential of the sandbox through showcasing demo use cases so that I can sell its worth to customers.

5. Iteration Plan and Estimate

Within the next few weeks we plan to gain access to the sandbox so that we can test the current software and learn how to use what has already been implemented. Currently, we are looking into FEMA Hazus which will provide us with the information necessary to estimate potential losses from earthquakes, floods, tsunamis and hurricanes. We are also learning to use Unity and Vuforia in order to implement the required features.This will be the goal of our first sprint. After access to the sandbox has been gained and we have a solid grasp on how to use the development tools, we can begin focusing on sprint 2. The goal of this sprint will be to implement an AR program that can save and load terrain data. The terrain data will be simulated and projected onto the sandbox, showing what sand needs to be altered to recreate the terrain. This should take approximately 2-3 weeks. Our final sprint will focus on implementing a feature to recognize physical objects that will have some effect on the simulation. The simulation should change as a reaction to the physical objects being moved within the sandbox. This should take approximately 3-4 weeks.