Winter Term Retrospective

1. Introduction

This is the winter capstone project retrospective for team C10. The primary purpose of this retrospective is to evaluate current implemented technical approaches and make comparisons to alternative solutions. The retrospective contains a brief overview of main technical tasks that were assigned by the project partner. An evaluation of tools and technical approaches is provided, evaluating the efficiency of current technical implementations. A review of team project management assesses the pros and cons of the current project management approaches. Finally, some recommendations are provided for the project partner and future teams who are interested in the project.

The goal of this project is to validate an efficient visualization workflow that geographic scientists use to visualize their geographic researches. The primary task proposed by the project partner Dr. Raffaele de Amicis is to visualize a tsunami occurring on the coast of Seaside, Oregon. The requirements are that the visualization must be realistic, use real-world geographic datasets, and be done in Unreal Engine. To visualize this tsunami stimulation, the following tasks are assigned to team members: use geographic datasets to generate terrain, integrate modeling software with Unreal Engine to programmatically generate large-scale city models, and stimulate the water system.

2. Tools and Technical Approaches

For gathering digital elevation models and consolidating them, we used the ESRI software ArcGIS Pro. ArcGIS is very powerful but requires some time to understand the software. Much of the early development period was dedicated to understanding ArcGIS's interface and tools. Some disadvantages of the software are the confusing documentation that confuses ArcGIS Pro with other ESRI softwares. There are also issues with error messages being unspecific on what issues the program had. When considering other choices, there are a few open source alternatives to ArcGIS, but have similar issues with requiring proficiency. OSU gives us access to ArcGIS for free, so it was the best option for us.

The current solution we used for city building and street generation is a modeling software called CityEngine. CityEngine can import geographic data sets, generate a large number of buildings and street models, and export the generated models to Unreal Engine. The disadvantage is that it takes a long time to learn new technical software and test whether it is suitable for our task. In addition, the integration of CityEngine and Unreal Engine is still in beta testing. Models exported from CityEngine cannot be adjusted effectively in Unreal. A more efficient alternative solution would be the Cesium for Unreal plug-in, which can be used directly in Unreal Engine. In this

case, the focus of the project development will be in Unreal Engine. However, the Cesium plugin is not available until the end of March, 2021.

When it came to the simulation aspect of the project, it was a requirement for us to use Unreal Engine. There are many alternatives to doing this, such as Unity or a homebrew solution, but the project guidelines outlined Unreal Engine as being essential, as part of the deliverable is to create documentation for repeating our project in Unreal. Unreal Engine is very different from many game engines, as it primarily uses visual scripting, leading to many issues when it comes to understanding how the engine flows when compared to more standard engines like Unity, that instead use a scripting language for simulation logic, like C#. Regardless, Unreal was still very effective in simulating our project, and has many native solutions for advanced graphics such as Megascans, and a built-in lighting engine.

3. Project Management

The management of our project was split up into three different main responsibilities. Tsunami simulation and terrain, AI and DEM creation, and structure and player design. Each one was given to one person. The objective was to allow each individual to become specialized in an area of expertise. However, this structure changed a little as our goals became more clear and the tasks were split up as tsunami and terrain generation, foliage and AI creation, and building and road generation. We did very well in terms of sprint planning and knowing abstractly what needs to happen, but a big hindrance to that progress was our lack of knowledge with the softwares we had to use. If we could start over, we would definitely spend more time in better understanding the “direction” we needed to face in order to learn effectively.

Because of our struggle in knowing the proper “direction” of learning, one of the main struggles our team faced was that we didn’t stick to the agile sprint format well, and met sparingly. I feel that we could have gotten a lot more progress, and understood each other’s issues better if we had met previously. The lack of a dedicated leader may have been a large contributor to this. From what was understood from the coursework, a leader and their responsibilities would shift from person to person, however I feel that if we had simply assigned one person to lead at the very beginning we would have been able to have a more tight schedule when it came to meetings and deliverables.

4. Advice and Recommendations

Given that the biggest obstacle in progress was our lack of experience with the softwares we had to use, I think it would be helpful if we had some sort of abstract introduction to different ways we can achieve our goals. For example, learning the basics of Unreal Engine effectively took the entirety of the first term - mainly because the learning curve is so large and the documentation is

difficult to traverse as a beginner - and it was not until taking an actual class on Unreal Engine that significant progress was made. Youtube videos were helpful as well, but it is much more difficult to gauge if the skills being presented are actually useful or not, or to know if the video is all instruction as opposed to education. To someone without any experience in Unreal Engine, the breadth and depth of its capabilities are very intimidating, and knowing where exactly to start is very strenuous. As such, if we could have had a better foundation in our direction of learning, I believe we would have made more progress. This is not to say that we didn't learn anything useful - on the contrary, we all gained incredibly useful skills along the way - but only that the quality of our learning might have been improved.

Leading into advice for inheriting teams, they should definitely consider taking a brief formal class on Unreal Engine and complete several projects that cover all the main features of the game engine. Because of how much there is to learn, this should take at least a full sprint session, if not more. This way, they learn what Unreal Engine is capable of along with having an abstract understanding of how to accomplish them, which will speed up the development process.