08356 ACW2

We’re all Doomed

Evaluation

Nicholas Taylor

Matei Giurgiu

Nathan Glasper

Joshua Morley-Stone

Word Count: 1092

Contents

Programming Evaluation

For the project an OBJ loader designed by Nathan Glasper was used, this accelerated the development as we were able to easily progress through the initial part of the project and be able to render objects in the program straight away, the small issue with this was that the other members of the group had to learn how to use the OBJ loader though objectively the time saved by doing this was much less than what was lost in learning to use it. Later in the engine development process the OBJ loader was switched to another one made by Matei Giurgiu because this one generates an element buffer as well and is reusing similar vertices instead of duplicating them for memory optimization. This change made the development process a bit slow because it needed some changes in the render system in order to make it compatible and fully integrated. But then the render system was completely decoupled from the main game logic which lead to a increased workflow.

The engine that was provided as a base for the project allowed for a rapid start to the project however limited the scope slightly as modifying that engine to allow extra features would be time consuming. The problems from this were mild as it was possible to accomplish everything we wished within the scope of what was provided.

After the original plan was put in place for which member of the team was doing which task it became clear that this plan was not completely viable, this was due to the requirement of certain features to be completed before others for the project to be able to be developed properly. This lead to problems where someone may need to complete part or all another person's task to be able to either complete or test their own. This could have been mitigated using a Gant Chart which will be discussed in the management evaluation.

In the project there is some commented out code related to camera controls and movement, this is due to the original design of the game matching more of a full first person shooter. Later in the project we modified the controls and camera to match the original provided specification.

The way Components are build using the IComponent interface lead to some design problems such as a component cannot know the entity to which is attached to because we couldn't define any variables in the interface only methods. Sometimes we needed to get the transform component attached to the same entity that the current entity was attached to. A better approach would be for the Component to inherit from an abstract class which will have a variable *parentEntity.* We overcome this by the consideration that in an Entity-Component-System pattern, components should not contain any logic, only data and the system will know every component of an entity and they should have the behaviour. In an ideal system this would work fine but sometime you still need to expose some code to the components which might require to know another component. We solve this by forcing a component to require the other needed component in the constructor. (e.g. ComponentCollider_Sphere).

Management Evaluation

The project took place over Christmas this year which added some difficulty in the development. Due to the fact that people in the group went home for the break we lost the ability to have meetings in person and discuss in an efficient way the progression of the project. To mitigate this, we were able to continue discussing the project through online methods such as Facebook messenger this however was not as efficient as meeting in person and did lead to a few communication issues as to who was working on what part at what time and ensuring properly that people did not overwrite others code in the SVN.

Another problem with the timing of the project was that everyone in the group had other group projects and assignments to be working on at the time, this made meeting a little difficult as we had to fit meetings around that. As the project started after the others at the time we had to fit our meetings around those already arranged for the other projects.

During the project, and especially over the Christmas break, the lack of meetings due to people not being available or in the city made collaborative working a little bit more difficult. This meant that when we were developing features for the game if work was committed to the SVN it sometimes had errors or bugs within the code that made it difficult for others to work on or test their own additions to the code. This problem could have been avoided if we utilised the features of the SVN a little more and used forks of the code when adding new features and only merged it in when they were completed. During the project we did not do this however due to people being mostly free over the break just not present in the city the problem was not overly pressing as we could contact each other over online methods to remove the bugs in the program causing these issues in a reasonable time frame.

As discussed earlier parts of the project were hindered by the order that tasks were completed in, to mitigate this we could have used a Gant Chart to better plan which features needed to be completed before others to break down the project better. This would have removed the problem of having tasks not completed in the right order and given a better time plan and time frame to the project.

Issues

At the beginning of the project the provided framework did not work properly on the computers available in some of the rooms at the university, this made it difficult to properly work on the project as a group as the rooms it would build properly in were not always available. This did not create the biggest roadblock and there were ways around the problem however it did slow down development slightly.

Towards the end of the project there were some issues with the SVN in which the transfer rate ground to a halt. This caused it to take close to half an hour to commit and update for very small amounts of information (20-30 megabytes), as this only lasted a few hours it did not create the largest issue however the timing of it was less than ideal with only a few days left to work on the project.