**Ben Odom**

**ADGP 225 Major Productions**

**Post Mortem**

**Programming**

**What went well?**

Communication started out great. UML documentation started out great and kept the project on track. The start of the project was overall exactly what I would like to see in the future.

Team members contributed a fair share. Each person on the programming side had a large part in either the coding or management aspect of the project.

The final product from a programming standpoint is impressive. What is completed and what is functional given the amount of time is commendable.

**What went poorly?**

Communication as the project dragged on became less frequent. At the start I knew how every system worked in the game without having looked at all of the code. Later on, scripts were created without any communication and systems became more complex without any explanation.

As the most knowledgeable programmer, I feel my input was not utilized during much of the design process aside from towards the start. This would have helped some of the more messy code cleaner.

**How can this be avoided?**

More communication needs to be used when changing or creating new systems.

**Artists**

**What went well?**

Any art that was usable was of good quality. The artists are creative and talented.

The original workflow presented was followed for as long as it was in effect.

**What went poorly?**

Very few instructions given were followed past the first thing it applied to. If a mistake was made, it was corrected and then made again for the next asset.

When the workflow changed, it was ignored. Assets were never reviewed by the programming instructor, and as such assets still had the same issues they always did.

Artists had very little experience with the engine we worked in. This meant that an artist could not/did not test the asset before sending it to the programmers. As a programmer, I have some experience with photoshop which was the main tool used to create much of the art in this game. I feel some experience in Unity for all of the artists would have been useful beyond measure.

The artist’s workload was almost entirely shouldered by a single member. The amount of time and assets presented by this member are not easily enumerable; the assets and time from the rest of the team can almost entirely be condensed to a few weeks and counted on one hand. I believe this was unfair.

Programmers sometimes worked beyond regular school hours to finish some things. This meant coming in on a day when there was not normally class (Thursdays/Fridays/Holidays). The same cannot be said for artists. This would be understandable if artwork needed for the current version was complete, but this is not the case.

**How can this be avoided?**

Artists should have some experience in Unity while in school. Programmers should have some experience in Photoshop/Maya while in school. Class time should be dedicated to teach these tools/game engines or any others that are used at the time.

**My Contributions**

* Gem Behaviours
* Grid Behaviours
* Pause Behaviour
* Combat Management
* Combat Scene Input
* Enemy Health Bar Behaviours
* Enemy Animator Controllers
* Most combat UI behaviors
* Input Recording/Playback
* Random Number Generator Serialization
* Stage Selection Swipe Input
* Art Integration Between All Scenes
* Rect Transform Layout Polish in All Scenes
* Programming Design Input Between All Other Complex Systems
* Photoshop, Github, Unity, and Resharper lessons for other team members