INT2001 BASIC GAME ENGINE PROGRAMMING - ASSESSMENT 3 - RUINS OF OBLIVION POST MORTEM

ADAM WILLIAMS
10241

Ruins of Oblivion - Post Mortem:

Project Completion:

Ruins of Oblivion was completed with most of the conceptual design implemented into the final game. Due to time constraints stemming from the scope of the game's systems, visual design and gameplay, extra time could not be taken to add better visuals and sound. Design reconsiderations during development also meant that some elements of the game were scrapped and could not be developed further. The time spent making this project, however, was thoroughly enjoyed and this game may be developed further in the future.

Design Changes:

Many changes were made from the initial design. A revision of the weapon and enemy system resulted in the EnemyArcher and EnemyMelee classes not being necessary. The EnemyBoss class could not be added due to their complexity within the time allocated for the project's development. The design of the HUD health and dash had to be changed to text as a sprite implementation was complex without a revision. The inheritance tree was rearranged for the Weapon and HUD classes due to quality-of-life features that the more specialised classes possess. Menu buttons were not added as their concept was revised and deemed unnecessary as the implementation already exists to easily change GameState with the press of a keyboard key. Additionally, many changes were made across the program to include new features and functionality for the developed designs.

What Went Well?:

The Weapon system and Menu system both came along quite well. A good inheritance design allowed for easy development and aided in separating the core functionalities of each element of the game. The visuals were very enjoyable to create and did not take as much time as was expected. Particle effects and collectables also went very smoothly and could be added wherever necessary with little effort.

What Could Have Been Done Better?:

More time could have been spent designing the initial class diagrams. This would have resolved many unexpected functional requirements during development which led to numerous changes and wasted time. Many issues arose that wasted time for the major systems .For example, ten minutes were spent trying to figure out why the player's sprite was not rendering, only to then realise that the player was being created before the background and was thus behind it.

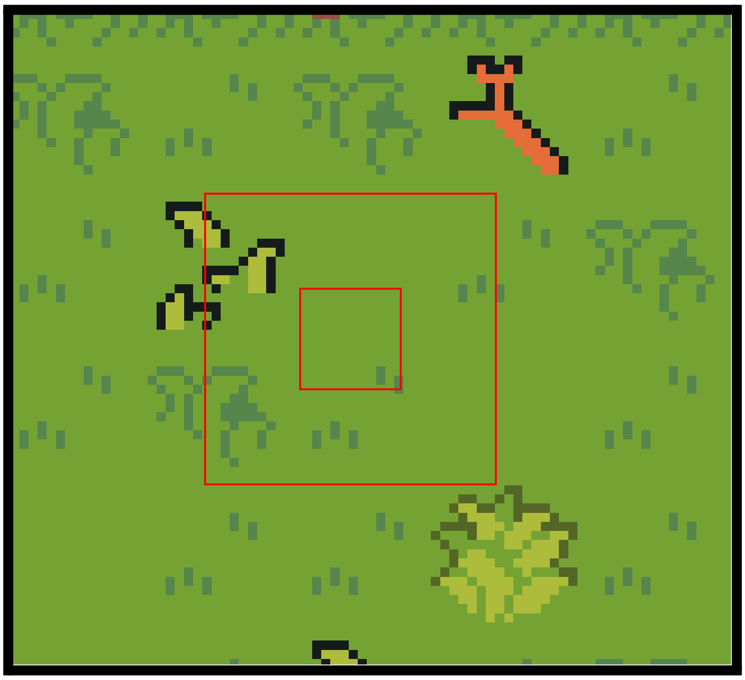

Figure 1: Player Behind Background

Additionally, an hour was lost trying to determine why the game was crashing as the crash log was filled with unrelated messages. The issue was that enemy weapons were being spawned during OnStart() which is during the object initialisation from the stack, thus the stack was indexed out of bounds. Many such strange issues and errors occurred across the game's development

Improvements For Next Time:

Spending more time in advance to better plan the project would have resolved many complexities during development. Improved time allocation and a better project scope would have allowed more detail and designed elements being implemented into the project. Generally, having more experience and knowledge with inheritance and OOP would aid to resolve issues with developing new systems. Regardless, *Ruins of Oblivion* turned out very well and I am satisfied with the final product that I got to develop.