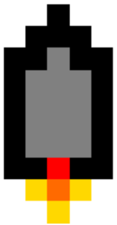
Week 1 Game
Astral Insanity

Neil Notman, Sam Serrels
Edinburgh Napier University
Advanced Games Engineering(SET10110)



1 Introduction

This report documents the delivery of the Advanced Games Engineering module's week 1 assignment. The created game was is based on the classical "Bullet hell / Shoot'em up'" game genre. The development time assigned to this project was 5 working days, so the design and feature scope was kept to a minimum to deliver a complete project.

2 Technical Features

C++ The game was written in C++, making use of the most modern standards and compilers where available. Using C++ for "Jam" style development can cause an increased level of difficulty, as quick game Development is focussed on the amount of features developed in the smallest amount of time. Efficient C++ code can be difficult to maintain when undergoing constant refactoring and code additions when compared to a high level language or game framework. Rough estimates put approximately 50% of development time spent on non-gameplay features i.e Bug fixing and usability testing.

SFML To get started quickly, the SFML library was used as a platform layer and rendering engine. Without a library as featured as SFML significant development time would have been spent on low level engine features, which would also increased the amount bugs that would need to be fixed.

Controller Support Any standard PC compatible controller should be able to play the game, this includes navigating any menus.

Windows x64/x86 + Linux SFML allowed the project to target multiple platforms easily, compiling for Linux did not require any changes to the codebase. However patches were required to the file layout and libraries to be compatible with a modern Ubuntu Linux system, so multi-platform did not go without some required developer time.

Installer For windows platforms an installer was created. It was discovered during deployment testing that all but the very newest versions of windows did not have upto date libraries for the C runtime, so the installer was programmed to also download dependencies if the system requires them.

3 Evaluation

Platform dependency issues and installer configuration encroached significantly into gameplay development time, and as such features were cut back in the interest of delivering a stable product.

Code Quality and Design As the game grew in size, the code design had to move a way for a simplistic "one .cpp file" design to compartmentalising systems and subsystems. With the use of engineering short cuts that would be frowned upon in larger projects, cross-system communication and synchronisation was kept efficient and easy maintainable. If the project was to be developed further, the codebase is more than ready to be built upon.

4 Conclusion

This project was a good experience in rapid game development with a list of must-have features and quality standards. We believe that we used the development time allocated as efficiently as possible and delivered a product that although lacking in features, is a complete game with minimal outstanding issues.

Web Page Project web page:
`https://dooglz.github.io/Astral-Insanity/`