Spike: Task 08
Title: Game State Management

Author: Luke Valentino, 103024456

Goals / deliverables:
The goal is to deliver a console program that implements “Zorkish: Phase 1” game using a flexible game state management method,
For example: UML diagram, code, reports

- Code see https://github.com/LukeValentino138/COS30031-2023-103024456
- Spike Report
- Design doc

Technologies, Tools, and Resources used:
List of information needed by someone trying to reproduce this work

- Swinburne Games Programming Lectures
- StackOverflow
- W3School C++

Tasks undertaken:
Steps Taken:

- Watched Swinburne Games Programming Lecture on Game State Management
- Implemented the code from the lecture that uses a stack to organise game states.
- Created a design document. The design document first details the requirements of each menu/class, then it details the contents of each classes’ functions and constructor.
- In multiple scenarios the stack needs to be popped multiple times. This causes issues as popping the stack from within the class on top of the stack leads to errors. A way around this is to use a queuing system. When a stack would be popped multiple times, instead of popping it while in the class, a counter is updated to reflect how many times the stack needs to be popped, then after render() and update() executepop() is called from the GameManager class itself.
- In the class “HallOfFame” a way to determine if the top state of the stack is “MainMenu” is needed. This is because HallOfFame can be accessed in multiple ways, so just will not work. Initially I tried to do this using dynamic cast, but ran into issues with the order in which classes are declared. The current solution is to keep track of the previous state (e.g. mainmenu, halloffame, etc.) and use this information to determine how many times the stack needs to be popped.

What we found out:

Using a similar code structure to what was shown in the lectures I was able to implement phase 1 of the Zorkish game. This came with a few issues that were addressed in the Tasks undertaken section.