Spike: 5
Title: Game State Management

Author: Krishna Adhikari, 4953193

Goals / deliverables:
Summarise from the spike plan goal
Besides this report, what else was created?

- Paper Design for the program
- A console C++ implementation of Zorkish in C++

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

- Visual Studio 2015
- Using Design Pattern State Article:
  http://www.cplusplus.com/articles/zAqpX9L8/

Tasks undertaken:
List key tasks likely to help another developer
This section should resemble a tutorial – the goal is to allow another coder to reproduce your work following these steps.

- Start with the Paper design for the program.
- Once you have the paper design, you can see the link between various classes.
- Start by implement the base classes. For this program, GameState, GameStateManager and MainMenu were created at the start.
- After the base classes, start creating class as per the need.
- Test and debug after implementin. each class.

What we found out:
Describe the outcomes, and how they relate to the spike topic + graphs/screenshots/outputs as needed

Before the spike, I had no knowledge about state pattern design. In this state pattern design, GameStateManager is a context class which contains different states.

```cpp
#include "stdafx.h"
#include "GameStateManager.h"
#include "MainMenu.h"

using namespace std;

GameStateManager::GameStateManager()
{
    _state = new MainMenu(this);
}

void GameStateManager::Handle()
{
    _state->Handle();
}

void GameStateManager::setState(GameState* state)
{
    _state = state;
}
GameStateManager::~GameStateManager()
{
}
```

As shown in the screenshot above, when GameStateManager class is called at the start, it creates an instance of MainMenu and loads it.

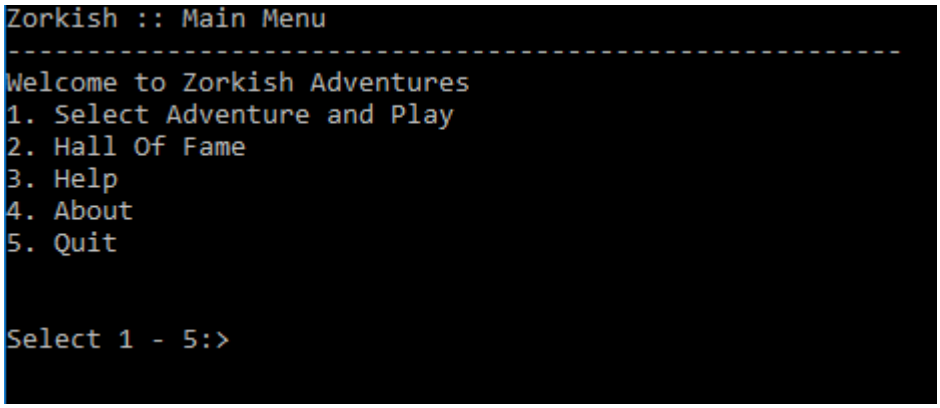
```
void MainMenu::Handle()
{
    do
    {
        DisplayMainMenu();
        switch (userInput)
        {
        case '1':
            _context->setState(new Select_Adventure(_context));
            _context->Handle();
            break;
        case '2':
            _context->setState(new Hall_of_Fame(_context));
            _context->Handle();
            break;
        case '3':
            _context->setState(new Help(_context));
            _context->Handle();
            break;
        case '4':
            _context->setState(new About(_context));
            _context->Handle();
            break;
        case '5':
            _context->setState(new Quit(_context));
            _context->Handle();
            break;
        }
    } while (!isQuit);
}
```

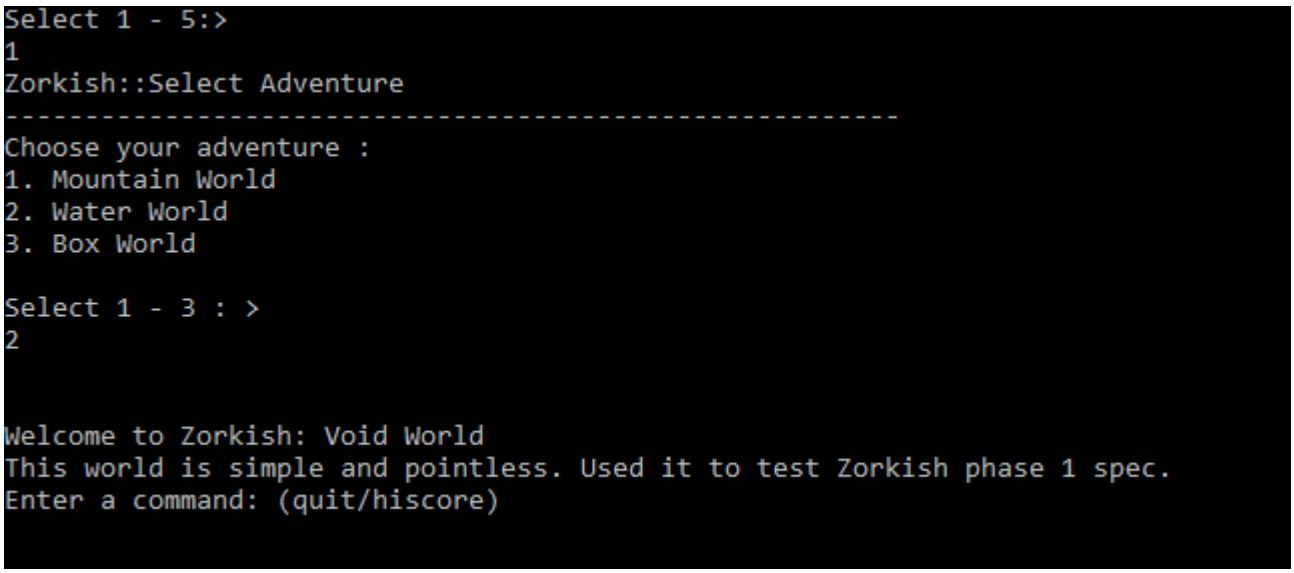
MainMenu has the main logic to call different state. Depending on the input from the user, different states are set by the use of context which is handled by GameStateManager. After setting the state, the Handle function function for that state is called.

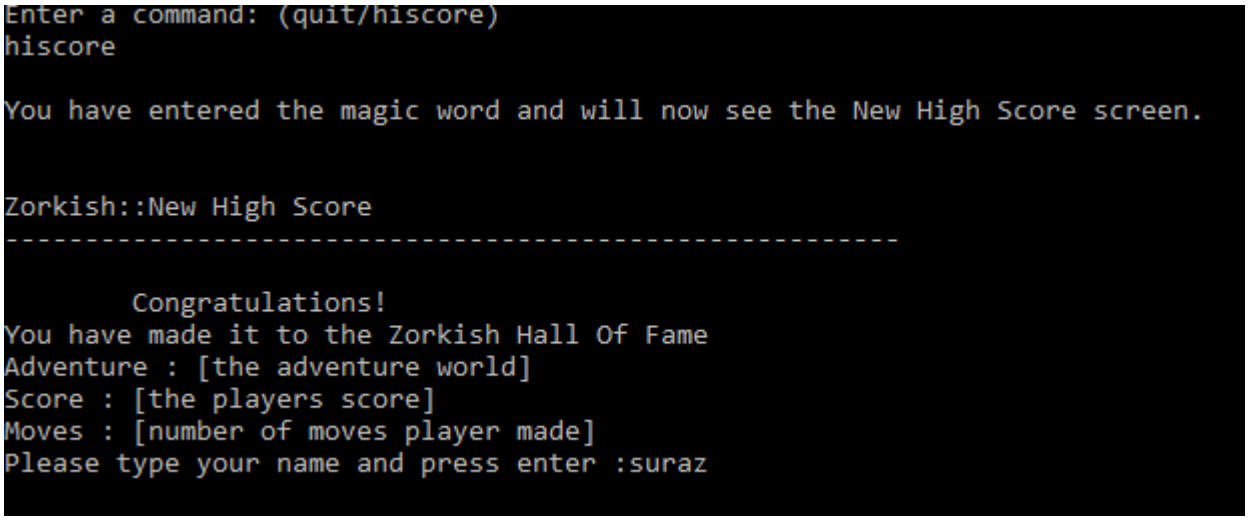
```
#pragma once
#include "GameStateManager.h"
class Select_Adventure: public GameState
{
private:
    GameStateManager* _context;
public:
    Select_Adventure();
    Select_Adventure(GameStateManager* context);
    ~Select_Adventure();
    void Handle();
};
```

The line class Select_Adventure: public GameState means that Select_Adventure is a child class of GameState and inherits the function. The handle function for this state is overridden.

Below are the screenshots of the program running:







```
Select 1 - 5:>
2
Zorkish::Hall Of Fame
------------------------------------------------------------
Top 10 Zorkish Adventure Champions
1. Fred, Mountain World, 5000
2. Mary, Mountain World, 4000
3. Joe, Water World, 3000
4. Henry, Mountain World, 2000
5. Susan, Mountain World, 1000
6. Alfred, Water World, 900
7. Clark, Mountain World, 800
8. Harold, Mountain World, 500
9. Julie, Water World, 300
10. Bill, Box World, -5


Press ESC or Enter to return to the Main Menu
```

```
Select 1 - 5:>
4
Zorkish :: About
-----------------------------

Written By: Krishna Adhikari


Press ESC or Enter to return to the Main Menu
```

Paper Design: