**Final Project Report**

Group: Courtney Jo Young and Christian Harris

Contact: young2cj@mail.uc.edu and harri3cp@mail.uc.edu

**The Game of Battleship**

**I. Introduction**

The project, which requires the design, construction, and implementation of a C++ game that my team has decided to undertake is Blackjack. This game was given implementations that allowed the game to run effective and in a manner that embodies the reality of this game but niche in a way that allows the user to play against artificial intelligence (the computer). The game allows the user to feel as if they are playing a real game of battleship. This game went through many staging processes and development issues that were overcame. The game is free of errors and runs effectively that embodies the rules, techniques and concept of the game of battleship.

**II. Project Description**

This C++ game of Battleship embodies code that constructs the game to follow rules, have a displayed a board, and allows the players to create an arrangement of ships. The user makes a move that is then followed by the Artificial Intelligence (AI) this grants the user experience to be exemplified as the original game of battleship would.

**III. Original Design and Specification**

The design and development specifications of battleship was responsible for ensuring a functionable game that executes effectively as well as compiles efficiently without warnings. This was accomplished successfully due to following the best practices of C++ as well as keeping the code organized with coding comments. The project assured a successful user experience and user interface against the computer that made the game resemble reality.

**IV. Program Feature List**

•  Implements the rules and guidelines of how the game of battleship is preformed

• Includes the option of multiple players

• Encompasses the playing procedures of battleship

• Allows the user to place its ships down to their preference and generates random placement of the artificial intelligences’ ships

•  Distinguishes both the user (or users) and Artificial Intelligence placement and board status within the game

•  Accounts for user error for uneducated / mistakes that the user may make or have

•  Originating classes that executes the game in an effective and efficient manner that resembles the playing rules and guidelines of battleship

•  Executes properly without crashing and embodies the qualifications of the game of battleship as outlined

**V. Testing Methodologies**

Testing methodologies were implemented to make sure every error a user could make, or the game would have, was weaned out and resolved before the project was launched. One testing methodology implemented was combinatorial testing, this type of testing implemented a “worse case scenario”. Every possible key stroke that was not encompassed in the game of battleship was hit to ensure the game did not crash, as well as reporting back to the user if they made a mistake in the game or had an issue with placement of their hit. Another testing methodology applied was functionality testing which ensured a functioning game free of bugs or errors that could affect the user interface and experience. This methodology looked for playing issues, graphic issues, and technical issues. When it came to the testing of different issues that were faced when implementing the game of battleship, divergent thinking occurred to ensure that the user experience would be consistent, and free of errors. Partaking and implementing multiple testing methodologies assisted our team in gaining confidence that the game will functions properly.

**VI. Program usage guidelines**

When the game begins, the user is granted the option to choose one player or two players. Each player places their ships down while the artificial intelligence randomly places its ships down. Once the process of choosing ships is over, the game starts by having the user shoot targets. Each player has as many shots as the number of ships they have placed for that turn. For example: if a user places down 5 ships, they are allotted five shots. When a ship has sank, the players are notified, and the player is then granted with one less shot than they started with. When all five ships on one side have been sunk, the player that still remains with ships is the winner. To implement this program, two classes were used. One class was used to keep track of each player's ships and board status. The other class was used to keep track of all the players. These classes ensured a functioning and efficient game free of errors that lead to a successful user experience.

Christian Harris

Professor Montjoy

CS 1021C

7 December 2018

Lessons Learned: The Game of Battleship

The lessons learned by taking on the development of battleship with my team was a lack of deciphering issues along the way. The game which involved the usage of the data structure, class, came across bug issues that were resolved later in the development that would have made the development process smoother if solved earlier. The game overall allowed me to learn more about the coming togetherness that accompanies a game against artificial intelligence.

Partaking in a project that involved Artificial Intelligence lead me to become more knowledgeable on the process of implementing a game both the user and Artificial Intelligence take part in. Implementing rules that allotted for error was also accounted for, which became troublesome when trying to figure out the most efficient way to implement error that may occur.

Overall the game of battleship was an exciting project to take on due to its unique features that make the game of battleship run efficiently against Artificial Intelligence. This project taught me a lot about how Artificial Intelligence works when accompanied by a user as well. Combining the two allowed me to understand how the process works and brought up issues I had never been faced with before.

Courtney Jo Young

Professor Montjoy

CS 1021C

7 December 2018

Lessons Learned: The Game of Battleship

Taking on the game of battleship lead me to understanding more how C++ data structures work as well as being faced with errors that I had never been faced with before. One issue I had but overcame was the knowledge in which the original game of battleship worked. I overcame this issue by locating a game online that explained how the dynamics of battleship worked. I then relayed this information to Christian, we then took that information gathered and created the most efficient game that worked exactly like the game of battleship does.

When it came to the actual process of creating this game, a few road blocks held us back that took us some time to figure out. When we took time to sort through our code, and resolve these issues, we became equipped with the knowledge to overcome the next similar challenge that came our way. That was really neat to me to resolve issues as a team and grow together.

The game overall functions phenomenally due to proper C++ practices being used and the testing methodologies that took place to ensure the best outcome of the project. After taking on this project, I felt confident that I could accomplish another similar project, which accomplished the challenge of making the code reusable. The project kept my interest throughout and taught me lessons I will apply in the future.