**Software Implementation and Testing Document**

**For**

**Group 4**

Version 1.0

**Authors**:

Cristhian Prado

Caleb Rachel

Lucas Compton

Mason Day

# Programming Languages (5 points)

*Unity (C#)*

* Used in Scene Management as well as general button functionality
* Will be used in Single Player functionality primarily
* C# was chosen as the primary language of choice due to its built-in functionality with the Unity Engine. An alternative was C++, but that would have required more overhead that we would have to accrue to get that set up as well.

# Platforms, APIs, Databases, and other technologies used (5 points)

*Platform: Desktop / Laptop*

*Unity Engine: Used as the Game Engine to run scenes as well as house the game itself*

*Virtual Studio Code: Works in tandem with Unity Engine when opening the main scripts for the game*

*GitHub Desktop: Chosen as opposed to Unity VCS given the teams familiarity with GitHub as well as the issue tracker as a point of usage for project contribution paired with easy-access commit history to review*

# Execution-based Functional Testing (10 points)

* *Menu Screen*
  + *Testing of scene switching from Menu to Single Player*
  + *Testing of Quit functionality*
  + *Testing of Tutorial Scene*
* *Asset Sourcing*
  + *All Cards and Chips sourced and working*
* *Single Player Functionality*
  + *Testing of raise functionality*
  + *Testing of deck creation*
  + *Testing of deck shuffling*
  + *Testing of card dealing*
  + *Testing of Game Flow*
    - *Cards dealt*
    - *Rounds of betting and decisions*
    - *Community card reveals*
  + *Testing of card dealing sound effect*

# Execution-based Non-Functional Testing (10 points)

*N/A – Focused on completing general game functionality first*

# Non-Execution-based Testing (10 points)

*We communicated general code walkthroughs of what new features each of us had added and how to go about reading the code to understand it most efficiently.*