Documentation

**Introduction:** Our goal for this project is to make the game Stratego. This will be a multiplayer game played on the same computer. This game will follow all the rules of Stratego. We intend to create a working GUI that will allow the players to easily set up, play, and win the game. At the beginning, each player will set up their 40 Units through the GUI. After the setup, gameplay will begin. Both players will take turns moving their Units to accomplish the goal of capturing the opposite player’s flag.

To create this project, we will need to implement the usage of a superclass, subclasses, and data structures to create a ranking system, methods that move and attack Units, and other attributes that will complete the game.

**IO:** The interface of this game will be made up of a GUI. Generally, the Stratego board will be made up of a 10x10 board of squares. There will be two 2x2 obstacles in that board that won’t allow units to move there. The interface of the board will alter as players set up their board and continue into the actual gameplay. To set up and move their units, players will click on units and a square on the board. During the gameplay, the players will see their own units and which unit is what (Flag, bomb, spy, etc.). In order to hide one player’s units from the other player, the units’ ranking value will disappear when it is the opposite player’s turn. For example, when it is Player 1’s turn, he or she cannot see the value of the units of Player 2. The number of each Units are: Flag (1), Bomb (6), Spy (1), Scout (8), Miner (5), Sergeant (4), Lieutenant (4), Captain (4), Major (3), Colonel (2), General (1), and Marshall (1).

**Detailed Description:** The goal in Stratego’s game is to seize the flag of the other side. For this purpose, each player has an army of 40 units classified according to the military ranks, which allows to easily retain the value of the units. The ranks and values are: Flag, Bomb, Spy (1), Scout (2), Miner (3), Sergeant (4), Lieutenant (5), Captain (6), Major (7), Colonel (8), General (9), and Marshall (10). The rules are as follows:

1. Units move 1 square per turn, horizontally or vertically
   1. Scout can move over any number of squares
2. To attack, one unit is moved to the same square as another. The higher value wins, except for the exceptions that follow:
   1. If both pieces have the same rank value – both are removed
   2. If the Spy attacks the Marshall – Spy wins. But if Marshall attacks Spy – Marshall wins
   3. The attacking unit is whichever one engages in attack (the one that moves)
   4. The Bomb defeats everything besides the Miner
3. Bombs and the Flag cannot be moved after the initial set up

To create this game, we plan to create a superclass called Unit, with a subclass for each specific unit. For example, a subclass for Spy, Scout, Miner, etc. This will allow us to define certain methods and attributes in the superclass that will apply to all Units, but allow us to define more specific methods and attributes for each Unit. For example in the superclass, we’ll initialize the ranking attribute, since each Unit will have a rank. In each subclass, we’ll define the ranking value and define methods that will cover the rule exceptions. For example, when a Spy attack a Marshall, and the Spy wins.

The final part we’ll need to implement is the GUI and graphics part. To do this, we’ll use the PyGame library. From this, we will be able to create a GUI interface. At the beginning, the interface will consist of a blank 10x10 board. Each player will click a specific Unit, then click a square on the board to place their Unit. After both players are set up, the game is ready to play. Both players will take turns clicking a square to move to and attacking if they so choose. The goal of the GUI is to provide a visual demonstration for the two players. We plan to keep it simple at the beginning, to focus on the actual gameplay algorithms. If time is left over, we will improve the GUI.

**Scope:** In this project, all the rules of Stratego are to be implemented. The fact that this two-player game is going to be played on the same computer will limit us. This is because in the board game, players cannot see the other player’s Units. Our group will need to take that into account and devise a way to hide the opposite player’s Units when a player is taking their turn. Another limiting factor will be our knowledge of creating a GUI. This will be a good experience to learn the process of what it takes to create one from scratch.