**Gun Game Project**

By Tom Nguyen

**Overview**

This project serves as practice building and working with object-oriented programming concepts: primarily classes and interacting objects to create a system where the player can battle an opponent with their unique selected character.

**Scenario**

***NOTE:*** **this project is based on a game of the likes of rock-paper-scissors, now being converted to a program to eventually allow for ideas of making an online app, possibly being a competitive video game!**

Gun Game is a competitive, class-based fighting game based on simultaneous action during combat and deception. Featuring multiple classes, Gun Game challenges the player’s ability to strategize and risk-management with its combat system. Based on famous roleplaying strategy games from the twentieth century, Gun Game encourages friendly competition and the self-actualization of its players through combat.

**Requirements**

This project will require several classes and functions: First it will require the main game’s main() entry point that the user will play the game through. Next, a class that organizes the result of the players’ actions is required, returning to the main game the result of how different classes interact with each other in battle (battle\_action). Fourth, a wiki class that returns how to use each character (wiki). Finally, a class is needed for each character in the game, with this first edition having two characters: Stock (typical pistol gun man) and Samurai (typical anime Shonen swordsman).

Main() (This is a function):

When running, the program here will:

1. Display a welcome message
2. Display the menu
3. Prompt user for input and navigate the program
4. Allow for ending program

1. Start Game

Starting the game should then ask the player what character they want to choose, showing them the list of available characters. Then it should ask what character the enemy AI will play. Finally, BATTLE START will be displayed, letting the player know it is time to play the game where their inputs now influence combat.

2. Character Wiki

This prompts the wiki class, using it to allow the player to find out what each character does! Should allow player to go back to the menu at any point.

3. Quit

Exits game.

Battle\_action():

Should contain every possible interaction and cause changes in the instances of each player. It should look like a large collection of if statements. Finally, it should also return strings to the main program to be displayed and let the player know the results of actions.

Wiki():

Contains information on how to use each character. Returns strings.

Stock():

Should have a constructor that assigns value of starting with zero bullets. Stock’s moveset should include: reload (+1 bullet), shoot (-1 bullet, is an attack and can kill), block (starts with 5, -1 each use, protects from attack), reflect (-1 bullet, if enemy shot gun, then kills enemy).

Samurai():

Should have a constructor that assigns starting value of unsheathed = False. Samurai’s moveset should include: unsheathe (changes unsheathed to True if False, else does nothing to not waste block), slash (if unsheathed is True, then this is an attack and can kill unless is blocked. If it is blocked, then the sword breaks and unsheathed becomes False), block (starts with 5, -1 each use, protects from attack). With slash, if enemy also attacks at same time, if the enemy attack is a bullet, the sword cuts through the bullet and the samurai wins. Otherwise, if the enemy attack is also melee (another samurai), then neither player kills the other.

**Goal**

To have this game be bang-ass fun, with fighting AI feeling like a proper challenge, and having the program being able to be improved upon in the future for future project use.