Team name: Goon Squad

Deliverable 1- Design Document

1. Project Background and Description:

For our project, we are doing Go Fish, which uses a standard pack of 52 cards. Cards are dealt to the players. Depending on the number of players there are, the cards dealt are different. For example, if two or three people are playing, each player receives seven cards. If there are 4 or more players, each player receives five cards. The rest of the cards are placed face down. The aim of the game is to have the most amount of “books”, and “books” are any four of a kind. The game goes in clockwise rotation. When a player is selected, the player can ask anybody for a certain card. If someone has that card, the player is required to give that card to the person who asked for it. If nobody has that card the asker has to “Go Fish”, which means to grab a card from the shared pool of cards in the middle. The player must have at least one card of the set or “book” in their hand in order to ask for that card. If you receive the card, your turn continues. When you receive all four cards you put it on the side.

2. Project Scope:

In our case, we are done when all the rules are implemented in the game logic/tested. As a team of two, we will split the workload 50/50. One person does half of the classes, the other person does the other half, and the rest of the requirements will be dynamically split.

3.High-Level Requirements

- Ability for each player to register with the game
- Ability for the game to communicate a win or loss
- Ability for players to know their status (score) at all times
- Randomisation of cards
- Card pick validation
- Turn based system
- Gameplay loop of Go Fish.

4.Implementation Plan

GitHub Link: https://github.com/YvesDonato/GoFish-GoonSquad

All files are stored appropriate files e.g. text files go into a text files folder. For our timeline, we can do two classes a week. One class for one person, the other class for the other. For the third week we will do the biggest class, game logic of Go Fish. Both of us are doing it together, because it is going to be the biggest class in the project.

5.Design Considerations

In order to make our project more cohesive, we can try to separate classes like game logic into smaller classes. We can focus on classes that aren’t inheriting any class to make it less coupled. We can be more sure we encapsulate a lot to make it more cohesive.