Names and IDs

B01902028 Shu-Ho, Chou. (Ben)

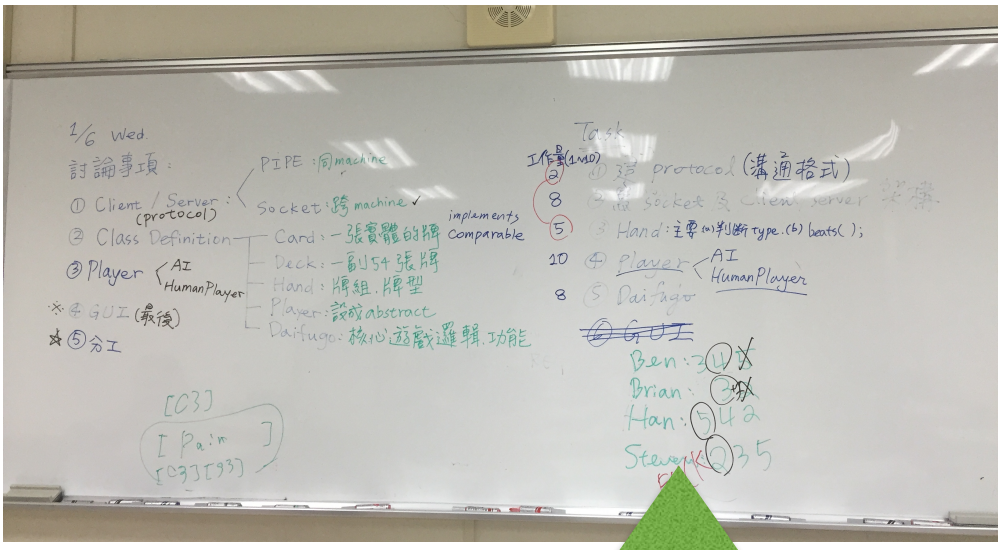
B01902044 Yu-Chih, Lin. (Steven)

B01902068 Po-Chih, Huang. (Brian)

B01902078 Han-Hao, Chen. (Han)

How to Divide the Responsibilities

Firstly, Steven helped build the basic structure of this project.
And we spend a night together at CSIE building to figure out how many parts were left to be filled.
And then each of us picked 3 favorite items to determine the ultimate distribution of the work.



The discussion for dividing responsibilities at CSIE building. All team members were present.

Ultimate work distribution.

Relations Between the Classes

The classes we defined are as follows, and the relations as well as the functionalities are described in text.

class Daifugo

the core logic, workflow, and special effects of this game.

class Card

each instance of this class represents a real-world poker card.

class Hand

a combination of cards that can be played (by players).

class Player

the participants of this game, instantiated by Daifugo class.

class HumanPlayer

a child-class of Player, to interact with human players.

class AIPlayer

an AI player with simple strategy, also a child-class of Player.

class Deck

represents a standard deck of cards.

class Server

providing server utilities for Daifugo to “listen to human players”.

class Client

responsible for communication between program and real people.

class Message

facilitate the communication between game’s core logic and players.

Advantages of Our Design

1. The work of each member is independent of one another. It's better for group projects since we didn't have to wait for others to finish their work first.
2. The efforts it takes to find bugs are alleviated significantly.

Disadvantages of Our Design

1. Difficult to extend our project with GUI (Graphical User Interface) because we didn't intend to build GUI in the first place.

Packages We Used

None.

How to Play Our Games

The steps are described below.

1) clone the repository first. (repository name: `foop15_final01`)
2) use `$make run` to start the game as well as the server if necessary.
3) for human players, use `$make run-client` to connect to server.

(P.S. human players need to know the IP address of the server.)

The game logic and the rules can be found on its wiki page, and some of the rules that we have implemented are listed on the next page.

The rules that we implemented in our Daifugo game.

Titles for the players
- The grand millionaire (大富豪)
- The millionaire (富豪)
- The commoner (平民)
- The needy (貧民)
- The extreme needy (大貧民)

(If there are 4 players, then no one would be commoner. If there are more than 4 people, there would be multiple commoners.)

Order of dealing cards and playing cards
* dealing cards: from position "0" to position "(player_count - 1)".
* playing cards: Extreme Needy -> Grand Millionaire -> Millionaire -> Commoner -> Needy.

(If first round, player with [S3] plays first, and then follow the clockwise fashion.)

The end of a round
* until only 1 player left. (with cards in his/her hands)

Winning the Game
* The winning/losing at the end of the game is determined by the points a player has.

(Grand Millionaire earns 2 points per round, Millionaire 1 point per round, Commoner no point, Needy loses 1 point per round, Extreme Needy loses 2 points per round.)

Revolution
- Straight Flush triggers "Revolution".
- Four of a kind also triggers "Revolution".

(The power, i.e., strengths, of hands is reversed indefinitely.)

Jack-Back
* If a hand with J is played, the power of hands is reversed in the current trick.

Eight Enders
* If a hand with 8 is played, the trick is immediately over, and the player becomes leader.

Skip 5
* If a hand with 5 is played, multiple players are skipped according to how many 5's are played.

Jokers
* Jokers can represent any card, and is a little more powerful than that specific card.

Tight
* When the hand being played has the exact same fashion of suits as the previous hand on the table, it triggers a "Tight" situation.
* All hands later in that trick should follow the same fashion to be legal.

Hands
* any type of hands never beats hands of a different type.
* types of hands supported
  * single
  * pair (exactly 2 cards)
  * three of a kind (exactly 3 cards)
  * four of a kind (exactly 4 cards)
  * straight flush (at least 4 cards)
    * K can be followed by A, while 2 cannot be followed by 3.

AbandonTen
* If a hand with 10 is played, the player should abandon several cards according to how many 10's are played.

Give Seven
* If a hand with 7 is played, the player should give several cards to the next player according to how many 7's are played.