

Project Report for UNO Game

Hongyi Yang(s2573830)

Overall

The software has been divided into two major parts Server and Client, which is obvious from the package structure. The Client is dealing with the USER interface and I/O operation, so it requires the list of commands and how to solve them. The Server oversees controlling the game process and the DATA storage and exchange. Both parts need to understand the protocol which means the protocol should be reachable for both sites.



Class Analyze

At the Client site:

- Class GameServerConnector is designed for DATA exchange and Network Connection.

- Class Main is the viewer which dealing with user I/O.

At the Server site:

- class Card generating a object called card which have all the element of a card like type and color and number.

- class Stack is the model of a standard card stack, it have standard card and have the method to get a card and shuffle the stack.

- class MyGameClient is dealing with each single client storage the client information and data communication.

- class MyClientManager is the working on storage the all the client, it helps the server to communicate and manage the independent client thread at server.

- class CommanParser is the model of processing the communication. It transfer protocol to command and call the method.

- class AbstrunctGameServer A server with basic method includes start a socket, create connection and stop.

- class MyGameServer the main controller of the software, it get the server function from the Abstrunct one and adding method of accepting player, start a game and deliver the message to the model.

-class TGameProcess the model of the software, it judges other the game is player follow the correct rule, it calculating the result. Deliver certain type of message, tell controller to end the game.

-class server.Main is the runnable class to start the server.

MVC

MVC pattern in this case is naturally applied. Client is born to be the viewer and Server is a Controller. The TGameProcess is generally the model, there are other classes like stack is also part of the game logical. Other class within the server package that helps the server to run and process the game is the model or game logic. So, if I want to upgrade to GUI, I only need to change the client part since its already have all necessary information from the Server or Controller. If I want to improve the possibility of player number, change the Server is more than enough. I want to play some new special rules, I could only change the model, in real application, it does need some other change at other part, but Model-Viewer-Controller are theoretically individual program. For bigger project with multiple people join, Frontend engineer is focus on the user interface and the backend is focus on the server. The only concern is the API or interface need to be built together. In this case it simplified the project control.

PRE/POST condition:

The most significant class is the TGameProcess in my software:

System Test:

-Starting the Game: Running one server instance and two client instance,

Client1 send message: HELLO&SINO|, server reply: WELCOME&SINO|

Client2 send message: HELLO&FISH|, server reply: WELCOME&FISH|

Client1 send message: PLAY&2|, server reply: QUEUE&SINO|

Client2 send message: PLAY&2|,server reply to client 2:

START&SINO,FISH&R1,B9,Yblock,Rturn,Rplus2|

Server reply to client 1: START&SINO,FISH&B7,R2,Bblock,Y3,B0|;

The game start successfully, player 1 and 2 got the initial cards on hand.

-PLACE/DRAW card:

Client1 send message: MOVE&B7|

Server reply: HAND&Yblock,?changer,Y4,Y9|; Server to Client2: UPDATE&A&B7|

Client2 send message: MOVE&DRAW|

Server reply: HAND&R8,?3,Rplus2,R3,R7,Y8|;

Client1 send message: MOVE&?changer|

Server reply: ERROR&Index 1 out of bounds for length 1|;

Conclusion:

There are still some unfinished parts like the game logic of function card is not running properly the AI player is not a AI, it just going through the entire hand cards and try which one is possible to place otherwise it draw. The networking is not able to recognize the port is occupied.