Card

* Find rank & suit of a card
* this class connect to Hand, DiscardPile, GameUI & DrawPile classes with a one(from Card) to many(other) relationship.
* used with both Rummy & Switch games
* Dependencies : None

Hand

* used to insert and discard card on hand while playing the game
* this class connect to Card classe with a one(from Hand) to many(Card) relationship.
* Dependencies : RummyGame & SwitchGame

DiscardPile

* This class support to access card pile & use it while playing the Rummy or Switch game
* this class connect to Card classe with a one(from DiscardPile) to many(Card) relationship.
* When there is no any card to draw Discard Pile support to use all the discard cards as DrawPile
* Dependencies : RummyGame & SwitchGame

DrawPile

* Start the game with shuffle the cards
* this class connect to Card classe with a one(from DrawPile) to many(Card) relationship.
* Draw the first card on both Rummy & Switch game after the divide cards among players
* Dependencies : RummyGame & SwitchGame

GameUI

* Graphical user interface of both game
* this class connect to Card, GameServer & Client(Player) classes with a one(from GameUI) to many(GameServer) & 1-6 (Client) relationship.
* support to start new games , pause the game & exit
* Players can switch between games using buttons which represent the game (Rummy or Switch)
* Contains all the components that used to play the games using mouse clicks & drag & drop
* Display the game results of each game when the game is over
* Dependencies : RummyGame & SwitchGame

SwitchGame

* Main controller class of Switch card game
* this class connect to Message classe with a one(from SwitchGame) to many(Message) relationship.
* Start the switch game from deal and continue the game with both parties
* Control the flow of game interacting with Client Player
* Create rules & give instructions according to the game.
* Dependencies : None

RummyGame

* Main controller class of Rummy card game
* this class connect to Message classe with a one(from RummyGame) to many(Message) relationship.
* Start the rummy game from deal and continue the game with both parties
* Control the flow of game interacting with Client Player
* Create rules & give instructions according to the game.
* Dependencies : None

Message

* This class manage all the communication between game server & client players
* this class connect to SwitchGame, RummyGame, Client & GameServer classe with a one(from other) to many(Message) relationship.
* maintain the connection for each player & support to send messages separately
* Dependencies : ConnectionToClient

ConnectionToClient

* Create new connection for players & maintain IDs of each player
* this class connect to Client(Player) classes with a one(from ConnectionToClient) to 1-6 (Client) relationship.
* Support to Message class by passing incoming and outgoing messages
* Dependencies : GameServer

ClientPlayer

* Manage Players of both Rummy & Switch games
* this class connect to ConnectionToClient & GameUI with a one to 1-6(from Client) relationship.
* also this class connect to Message classes with a one(from Client) to Many (Message) relationship.
* Store the game host and use it to connect to the game server
* Maintain Access up to 6 players to play the game
* Dependencies : None

GameServer

* Main Pc of the games
* this class connect to Message with a one to Many(from Message) relationship. & also GameUI with a one to one relationship.
* Use web sockets & port to work efficiently
* Deal with the each player connections and maintain player ids
* Send and receive all the messages to & from the players
* Generate game results
* Support to Shutdown & Restart when needed.
* Dependencies : None