

Assignment – Object Oriented Design (COMP3004L)

UML Diagrams for Switch and Rummy card games
Final Report



Team name	6droids
14208971	Tharkana D Kodagoda
14208893	Sahitha Nelanga H De Silva
14208910	H W Srimal Priyanga Fonseka
14209059	Dilina Namal Weerasinghe
14209074	P W Poorni Yasodara
14209759	Kavindu Yudeesha Lakshan Narathota

Contents

Introduction.....	
1. User case Diagram	
2. State Diagram.....	
3. Sequence Diagram	
4. Domain Analysis.....	
5. Class Diagram	
5.1. Class Responsibility Collaborator	
6. Object Diagram.....	
7. Deployment Diagram.....	

Introduction

This Report is a collection of UML diagrams which will give a complete example of Object Oriented Analysis, Design and Programming applied to 2 simple card games, Switch and Rummy!

In this case we studied all the possible ways of winning each game and came up with the optimal solution to get the best outcome.

1. User case Diagram

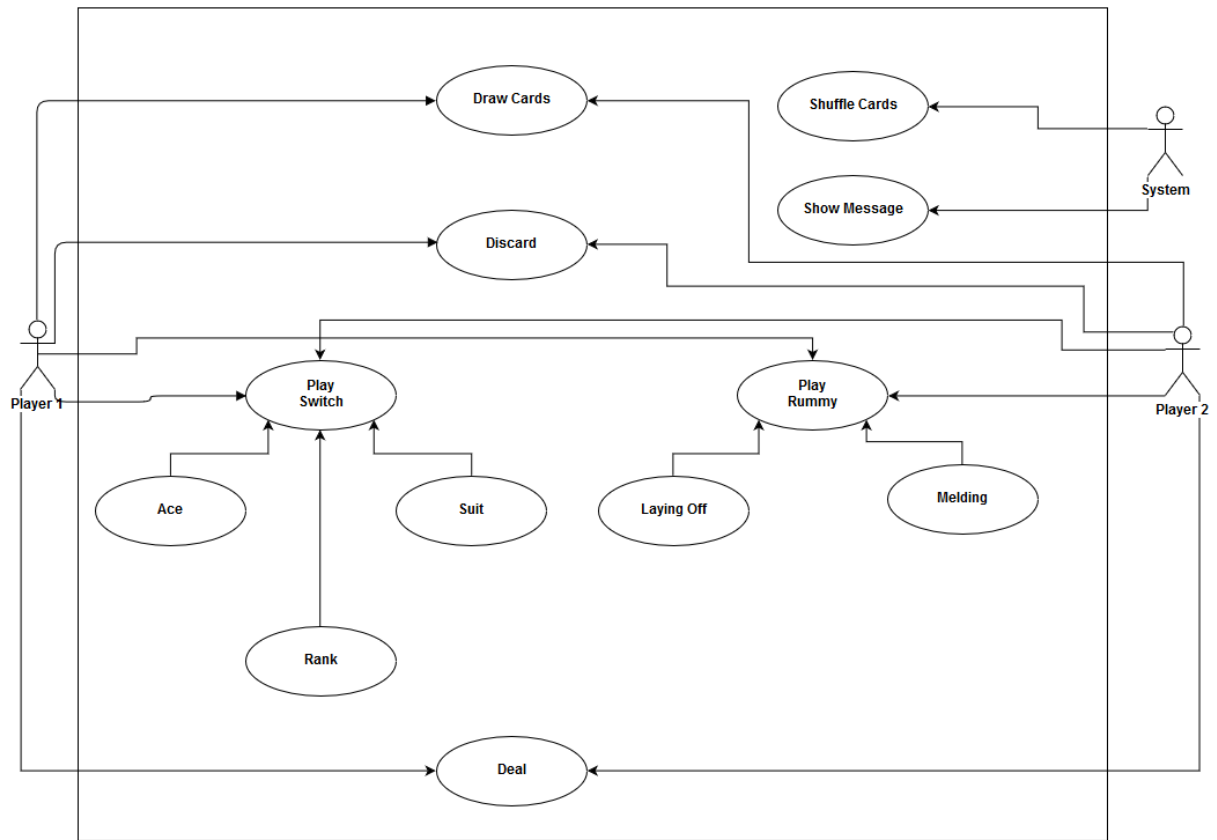


Figure 1 - Use Case Diagram

2. State Diagram

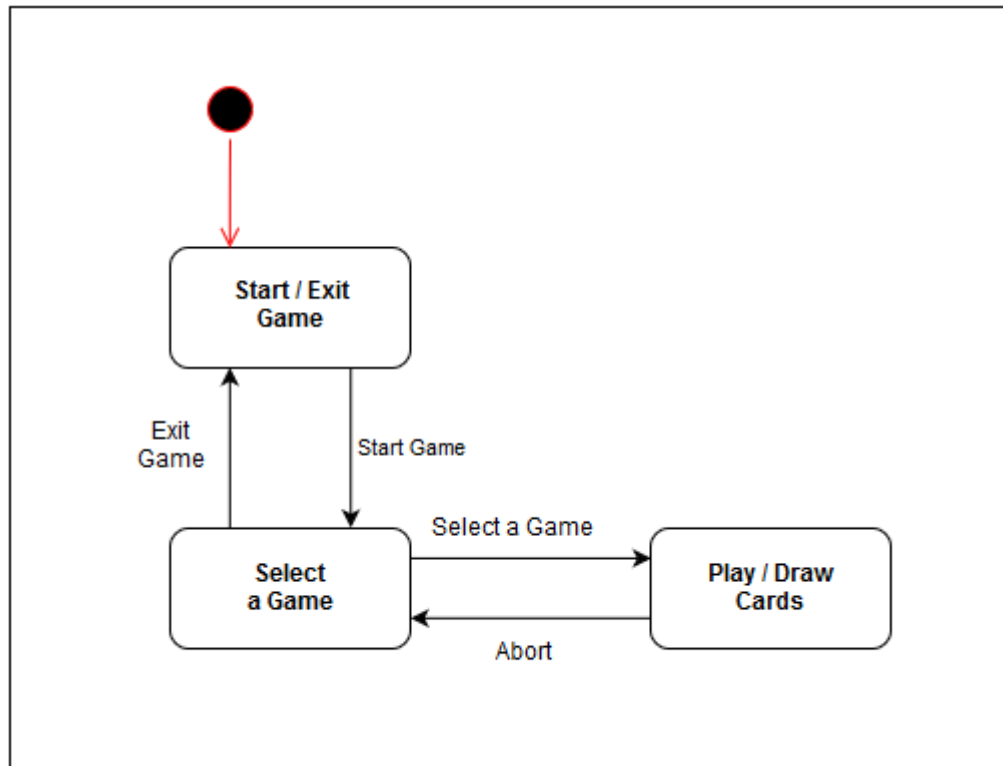


Figure 2 - State Diagram

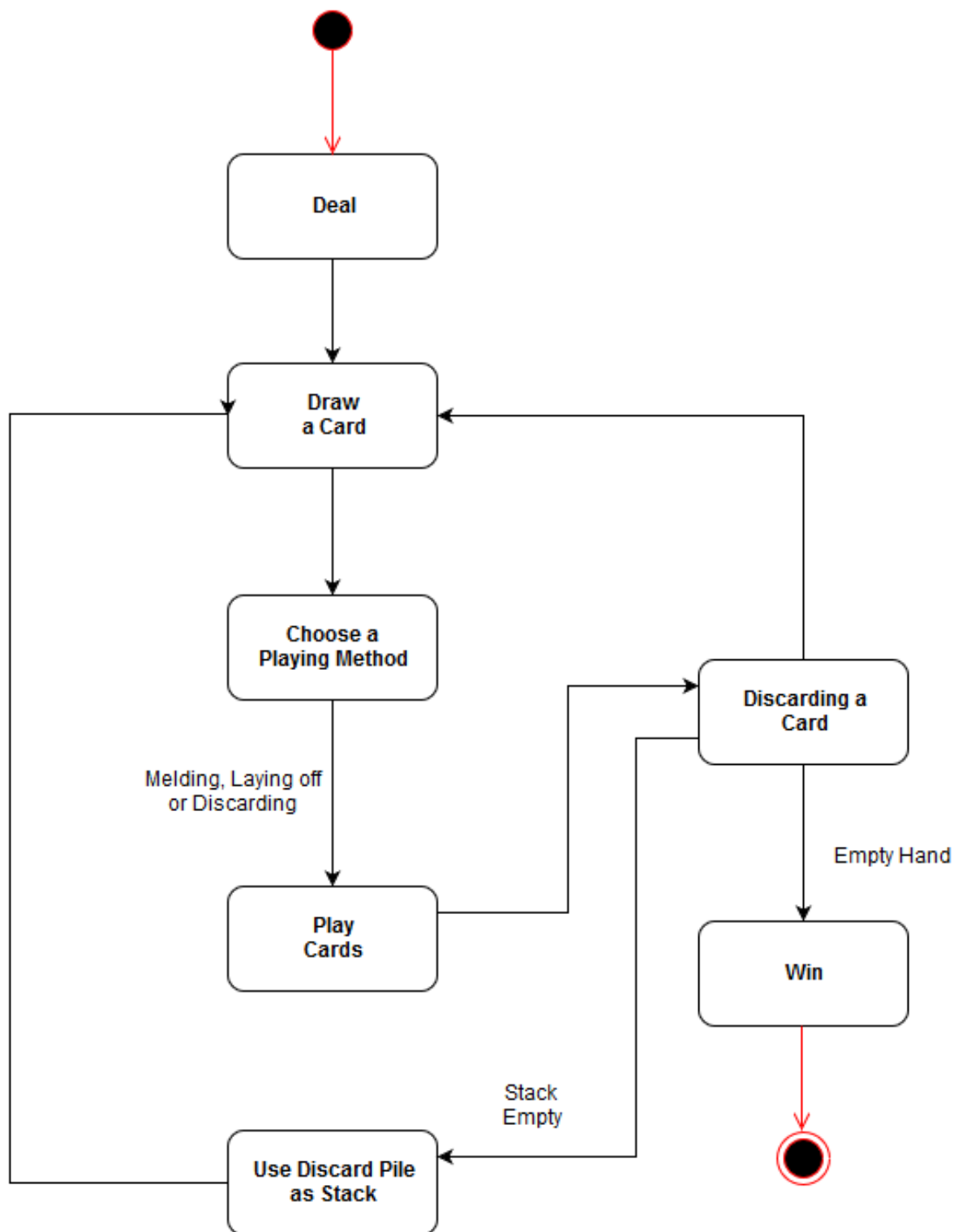


Figure 3 - State Diagram (Rummy)

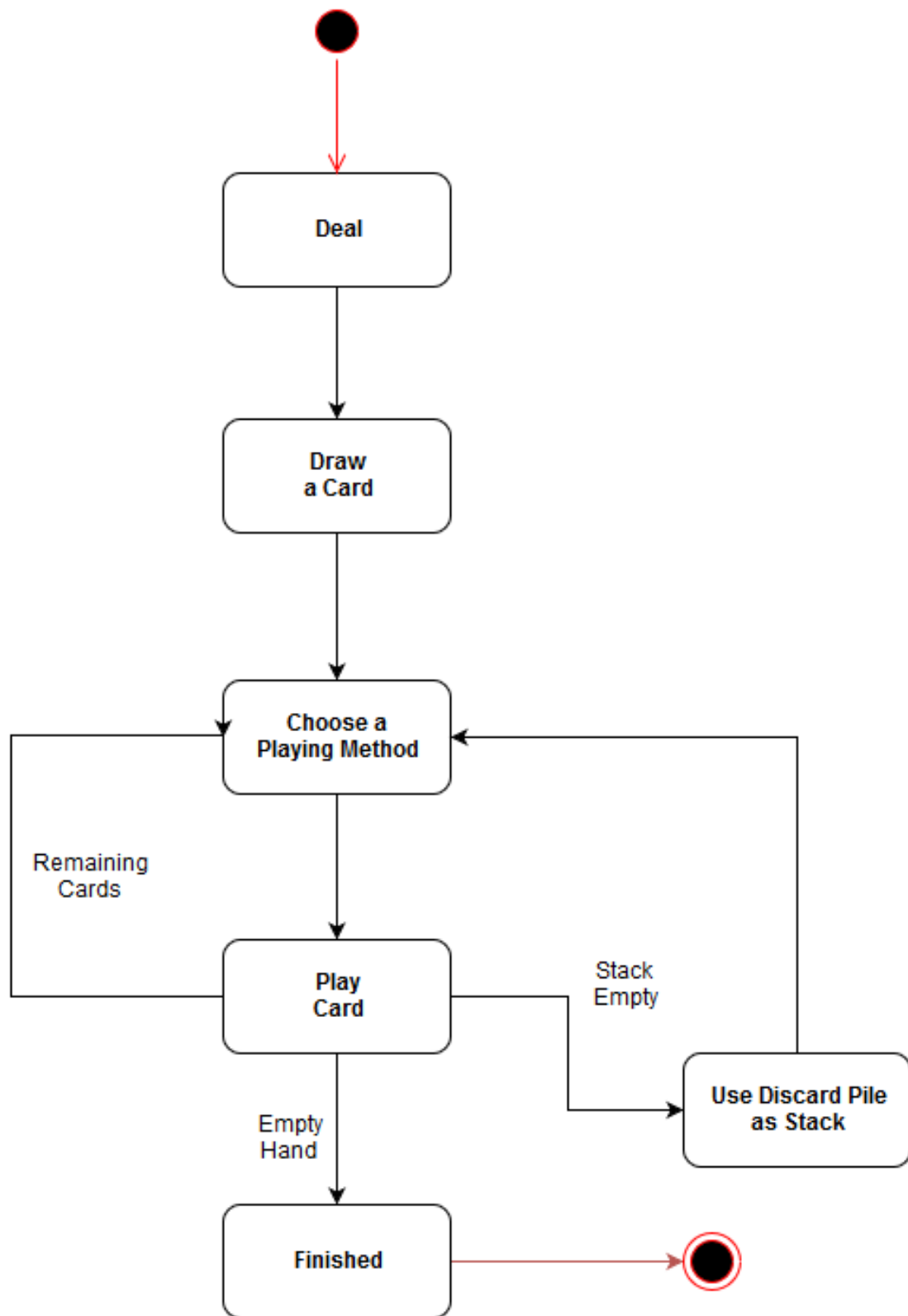


Figure 4 - State Diagram (Switch)

3. Sequence Diagram

Read more - <https://goo.gl/CScmSn>

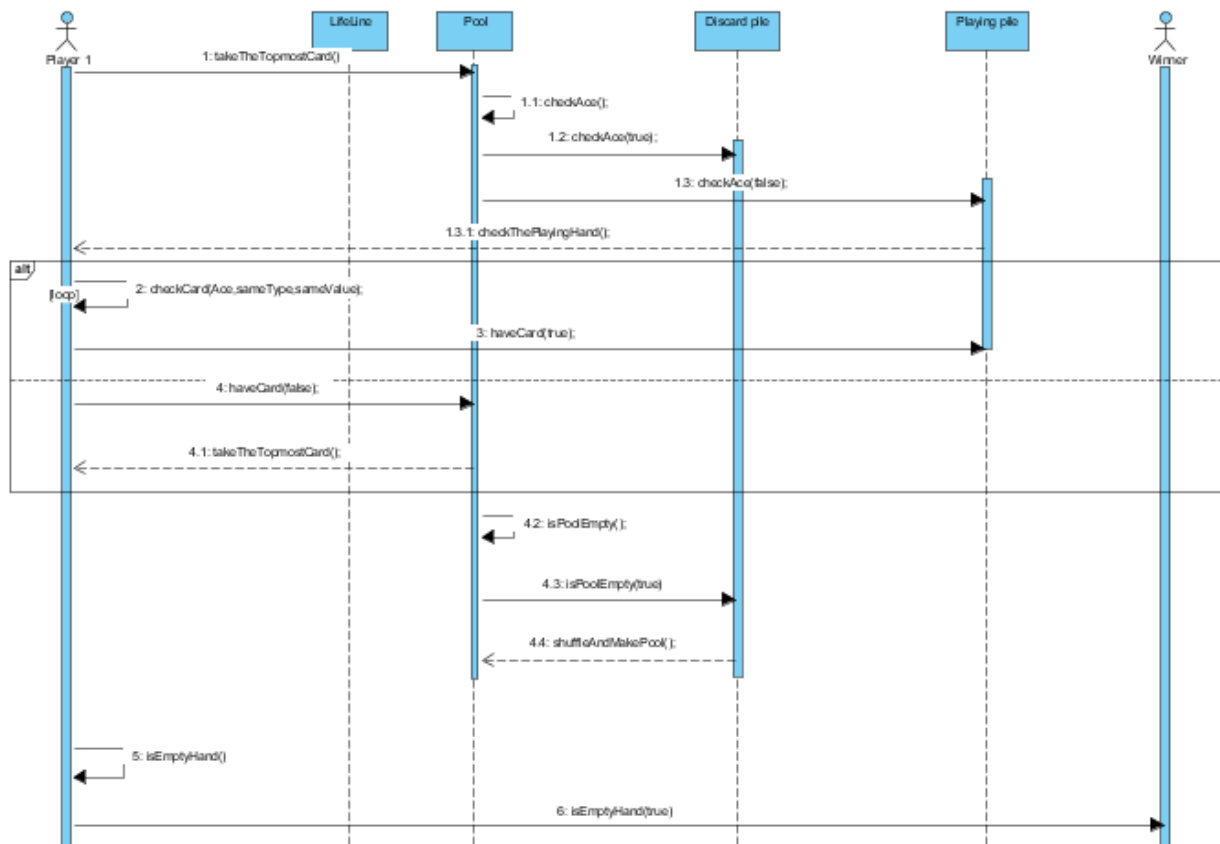


Figure 5 - Sequence Diagram (Switch)

Read more - <https://goo.gl/zOXLRG>

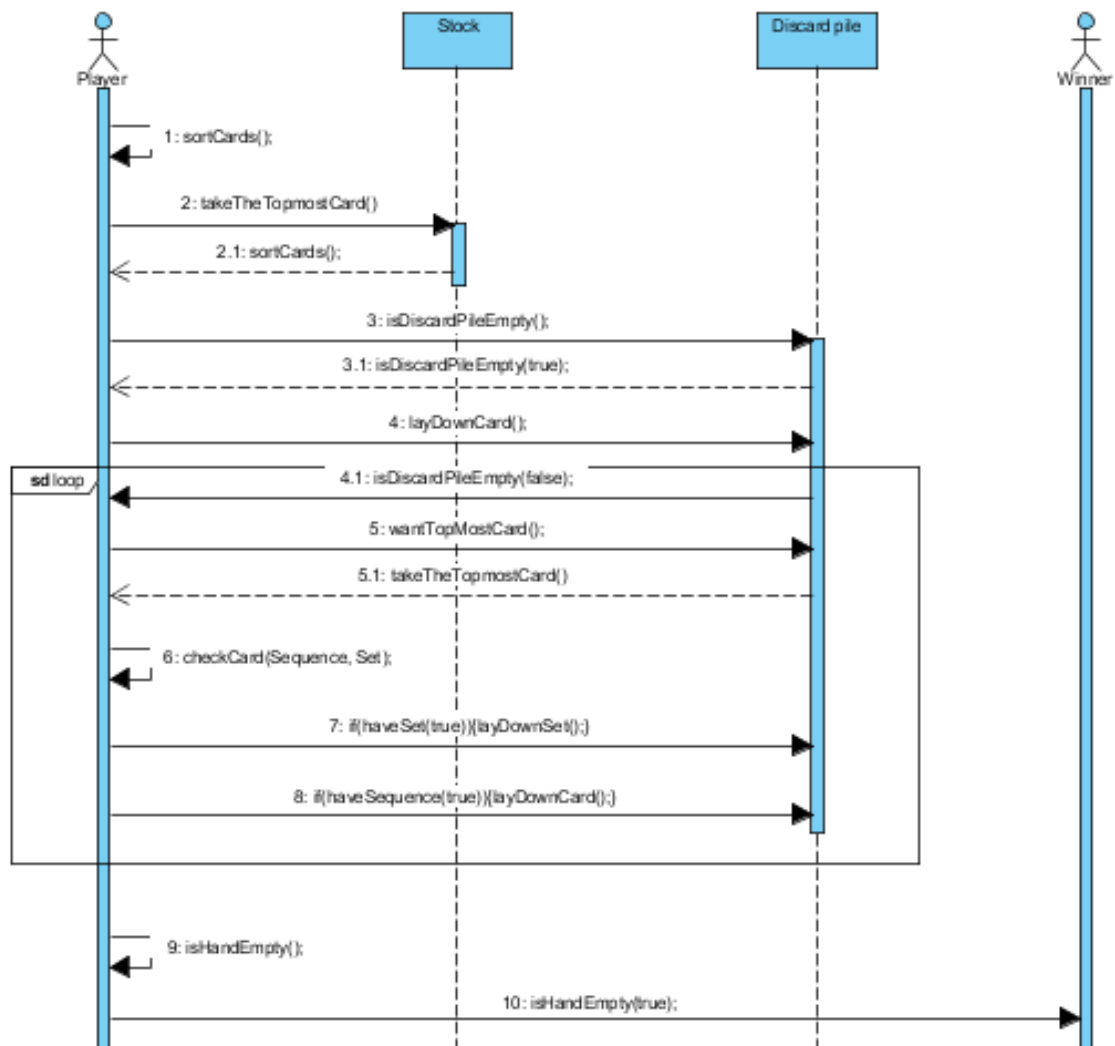


Figure 6 - Sequence Diagram (Rummy)

4. Domain Analysis

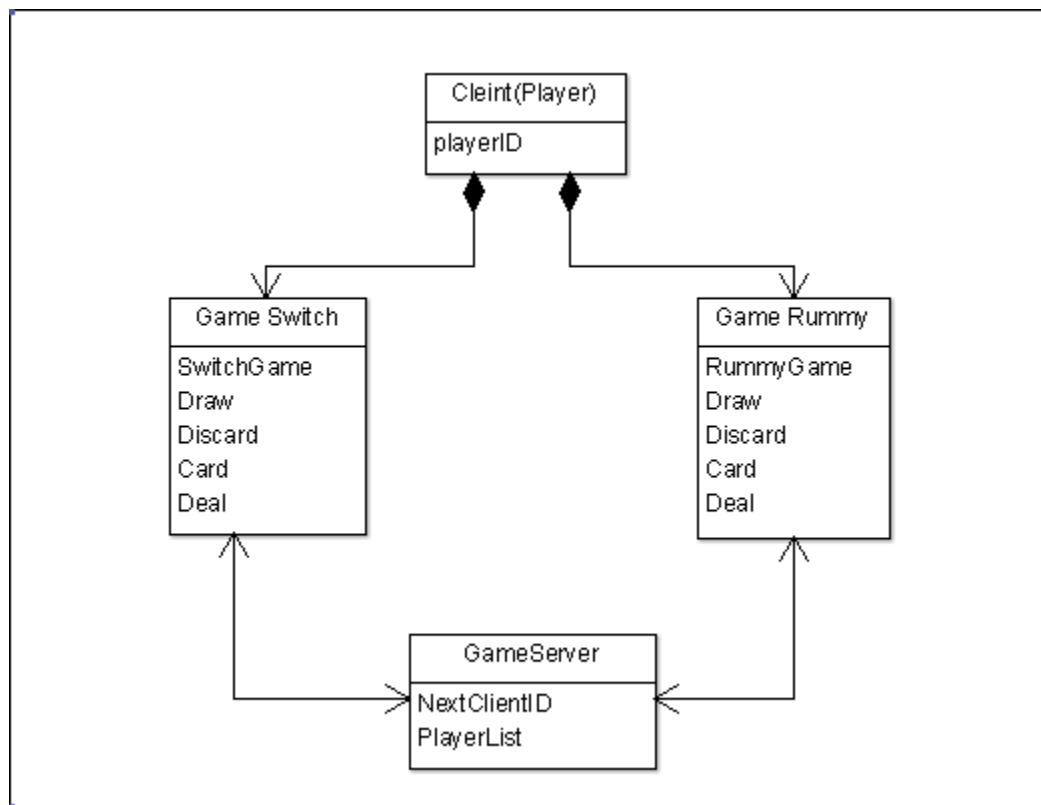


Figure 7 - Domain Analysis

5. Class Diagram

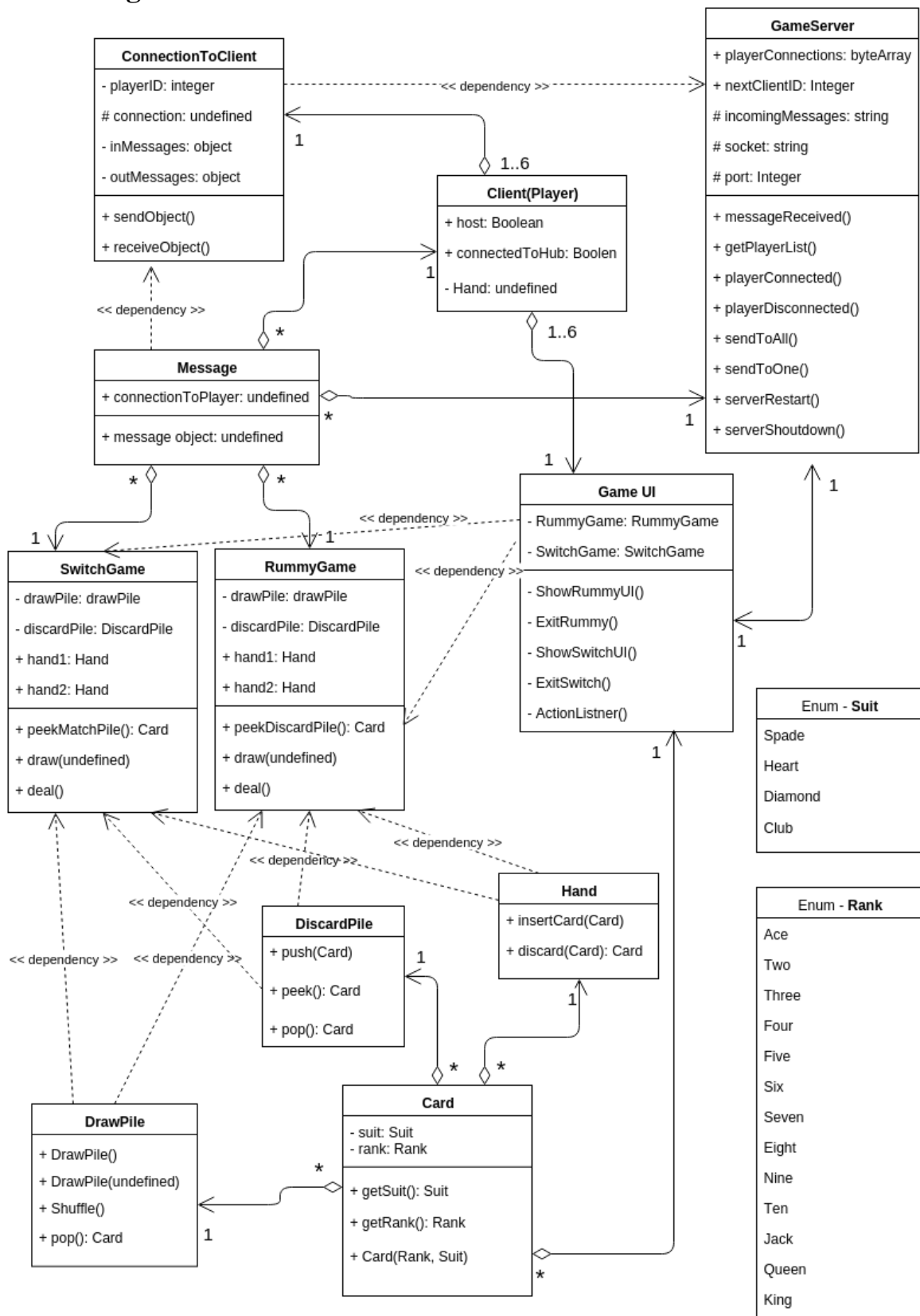


Figure 8 - Class Diagram

5.1. Class Responsibility Collaborator

Class Card

Responsibilities

- This finds Suit & Rank of a card
- Used with both Rummy & Switch Games

Collaborators

DiscardPile
GameUI
DrawPile
Hand

<<Dependencies: None>>

Class Hand

Responsibilities

- This used to insert and discard card on hand while playing the game

Collaborators

Card

<<Dependencies: RummyGame & SwitchGame>>

Class DiscardPile

Responsibilities

- This support to access card pile and use it while playing the Rummy or Switch game
- When there is no any card to draw, Discard Pile support to use all the discard cards as DrawPile

Collaborators

Card

<<Dependencies: RummyGame & SwitchGame >>

Class DrawPile

Responsibilities

- Start the game with shuffling the cards
- Draw the first card after dealing the cards among the players on both Rummy & Switch game

Collaborators

Card

<<Dependencies: RummyGame & SwitchGame >>

Class GameUI

Responsibilities

- This is the graphical user interface of the both games
- Support to start new games, pause game & exit

Collaborators

Game Server

ClientPlayer

- 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 or drag & drop
- Display the game results of each game when the game is over

Card

<<Dependencies: RummyGame & SwitchGame >>

Class SwitchGame

Responsibilities

- Main controller class of Switch card game
- Start the game from dealing and continue the game with players
- Control the flow of game interacting with Player
- Create rules & give instructions according to the game

Collaborators

Message

<<Dependencies: None >>

Class RummyGame

Responsibilities

- Main controller class of Rummy card game
- Start the game from dealing and continue the game with players
- Control the flow of game interacting with Player
- Create rules & give instructions according to the game

Collaborators

Message

<<Dependencies: None >>

Class Message

Responsibilities

- This class manage all the communication between game server & players
- Maintain the connection for each player & support to send messages separately

Collaborators

SwitchGame
RummyGame

Client
GameServer

<<Dependencies: ConnectionToClient >>

Class ConnectionToClient

Responsibilities

- Create new connection for players & maintain IDs of each player
- Support to Message class by passing incoming and outgoing messages

Collaborators

ClientPlayer

<<Dependencies: GameServer >>

Class ClientPlayer

Responsibilities

- Manage Players of both Rummy & Switch games
- Store the game host and use it to connect to the game server
- Maintain Access up to 6 players to play the game

Collaborators

ConnectionToClient
GameUI

<<Dependencies: None >>

Class GameServer

Responsibilities

- Use web sockets & ports 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

Collaborators

Message

<<Dependencies: None >>

6. Object Diagram

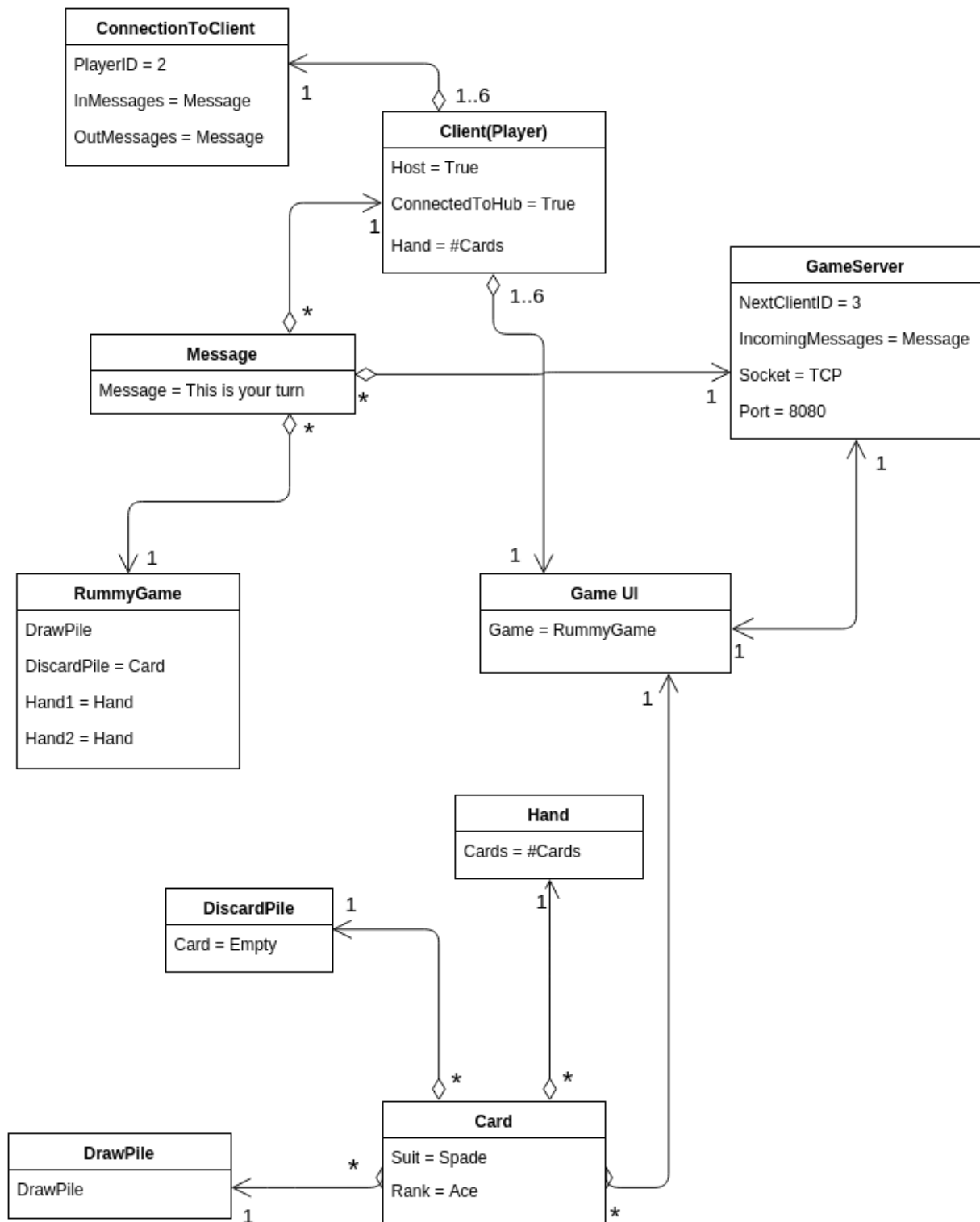


Figure 9 - Object Diagram (Rummy)

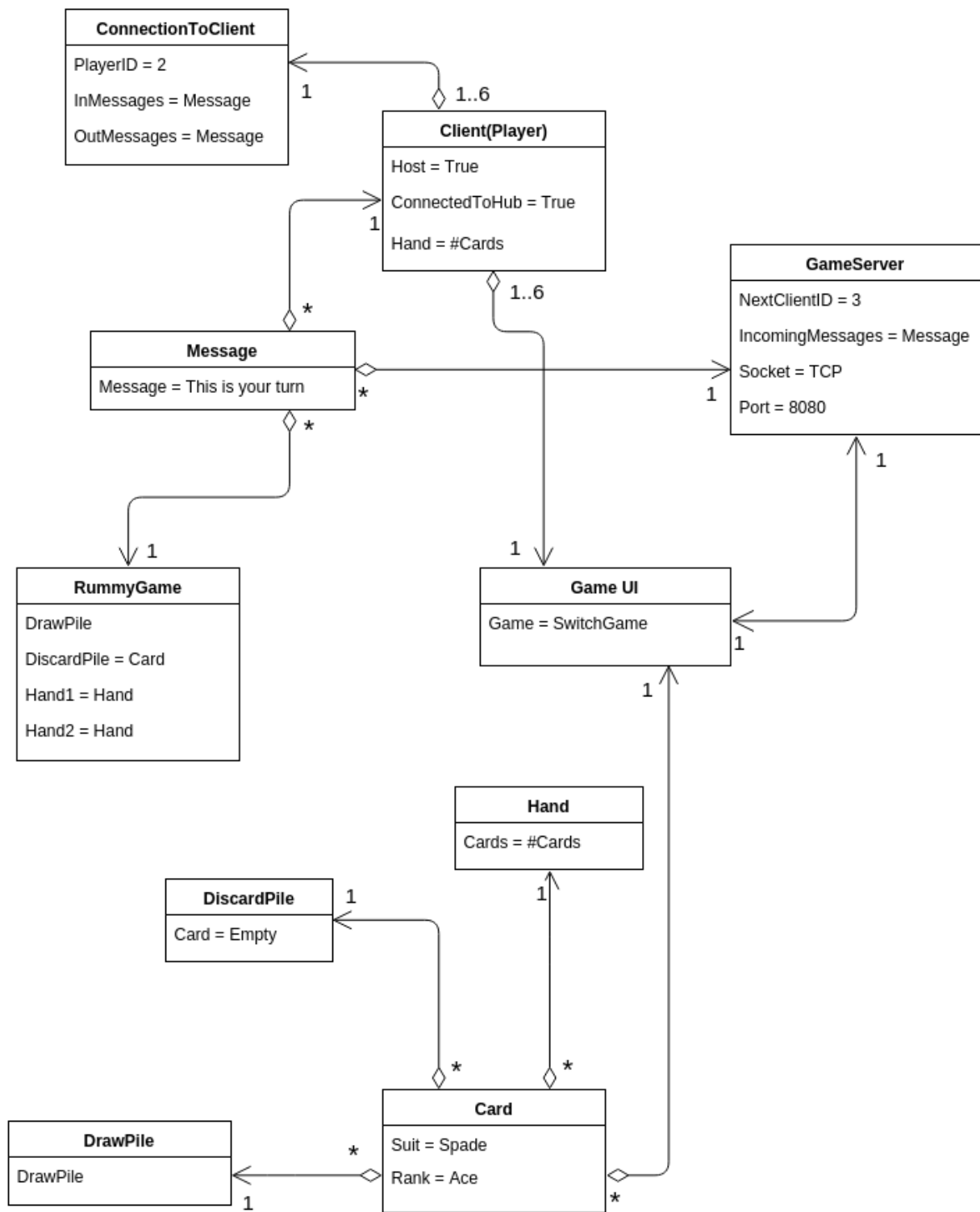


Figure 10 - Object Diagram (Switch)

7. Deployment Diagram

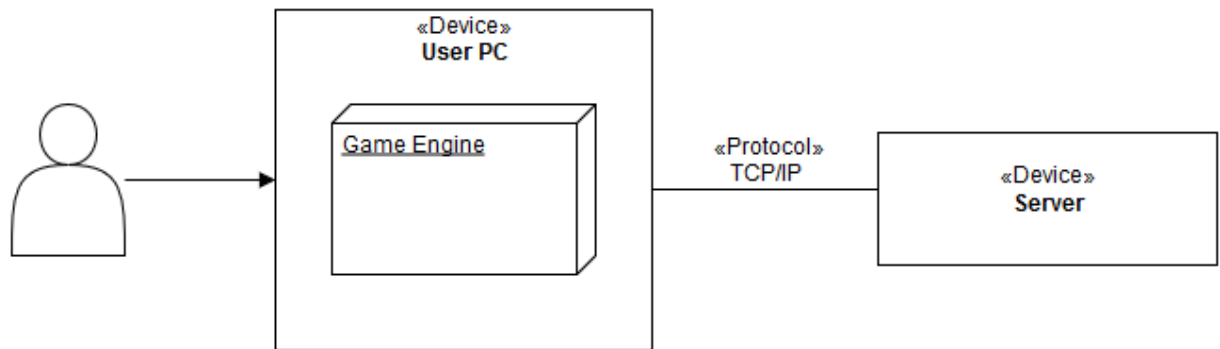


Figure 11 - Deployment Diagram