

Omi Online! CO324 Project II

7th of May 2016

GROUP 10 (E/12/103, E/12/323)

Department of Computer Engineering, University of Peradeniya

Table of Content

1 Introduction	3
2 Implementation	3
3 State diagram	3
4 Application messages	4
5 Design	5

1 Introduction

The game needs four players and uses a standard deck of 52 playing cards. The cards are ranked from highest to lowest *A K Q J 10 9 8 7 6 5 4 3 2*. Play begins by dealing the cards around the table until each player has 13 cards. The last card is turned face up and the suit on that card becomes the *trump suit* for that hand.

The first player may begin the game by playing any card. The rest of the players must play the same suit as the card that leads the trick, if they can. Otherwise, they may play any card. If a trump suit card is played, the highest trump wins. Otherwise the high card of the suit wins the trick. The winner of a trick leads the next turn.

When all 13 tricks of a round have been played, the player winning the most tricks scores one point for each trick won. The first player to earn ten points wins the game. This is the game logic of this game.

At the end of a round, the server updates each player's score and declare a winner.

2 Implementation

For this project Java servlets were used to program the server. HTTP requests are used to update the UI of the game. For that we use AJAX technology and jQuery library. Json-simple-1.1.1 library was used to handle Json objects in servlets. Knockout.js framework was used to update the interface real-time.

3 State diagram

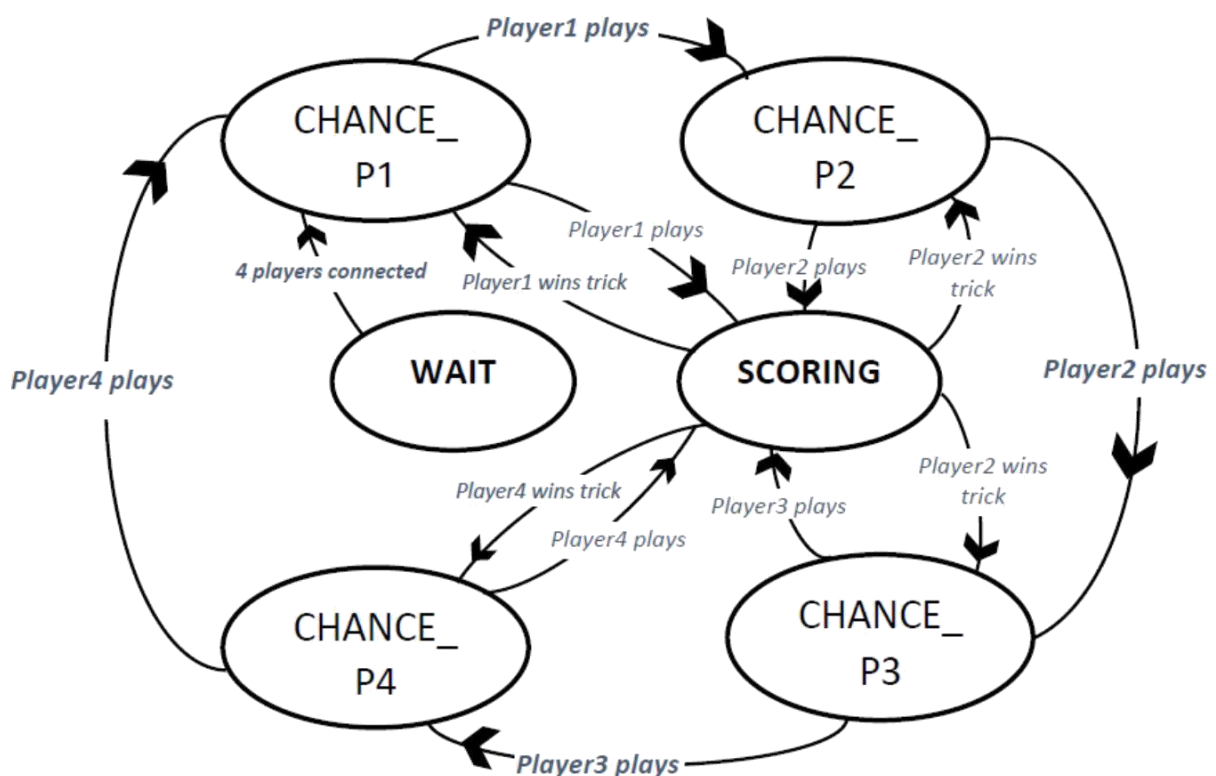


FIGURE 1

WAIT – Wait for 4 players to connect

CHANCE_P1 - Player P1 has the chance to play

CHANCE_P2 - Player P2 has the chance to play

CHANCE_P3 - Player P3 has the chance to play

CHANCE_P4 - Player P4 has the chance to play

SCORING – Calculating score

4 Application messages

By client

addPlayer – Send a request to get connected as a player.

Parameters – name entered by the client

cardPlayed – Send a request to the server with the played card. **Parameters** – name of the card

update – Send a request to the server asking for updates on current trick

Parameters – none

By server

Response to **addPlayer** – send a json object with a json array of cards of the particular client
Response to **cardPlayed** – send a json object to

1. Boolean 'valid' – whether the card play is valid
2. Json array 'cards' – hand of client

Response to **update** – send a json object to

1. Json array 'cards' – cards of current trick
2. Boolean 'shouldShowHand' – whether to reveal hand of client

5 Design

Four clients must connect to the server to start the game. When a client type and visit the URL of the game application, a web page with a form will be displayed. He/she Must enter his/her name in the text box and click on the 'Enter Game' button below it. Then the client will be entered to the game. The client won't be shown anything until four such players are connected. When four players are connected to the game, each player will be given 13 cards. The first player to connect the game will have the first chance to play a card. In the first trick, the players will have the chance in the order they are connected.

When a client enters his/her name and enter the 'Enter Game' button, he a http request will be sent to the server refering the GameLogic servlet. The request will send the name entered by the client and the type of the request he/she is sending. The type of the request is set to 'addPlayer' in this game. Then the servlet recognizes the request as an employer request and stores the information of the client (session will be created). Upon receipt of the request sent to add a player, the server respond with a json object with the set of cards of the particular player, boolean saying whether the cards must be shown to the client or not. The clients always send requests to the server checking whether any new updates are there.

This is done by Ajax http request sent every second. These requests are handled by GameLogic servlet. The boolean saying whether to display the hand or not will be set to true when all four players are connected.

Then the clients are allowed to play cards in order. When a player plays a card, an Ajax http request will be sent to the server with the card played by the client. The server records this and update the common variables used to update the every client. After the first trick, the client winning the previous trick will be given the chance to play first.

The score will be displayed near each of the client's card boxes. An enumeration class called State maintains the current state of the game. It changes according to the chance of the player or the waiting or scoring state.

If a client gets 10 points within the current round, the game will be stopped and he will be announced the winner. Otherwise, if all 13 cards are played without a client getting 10 points, client with the maximum points will be announced the winner.