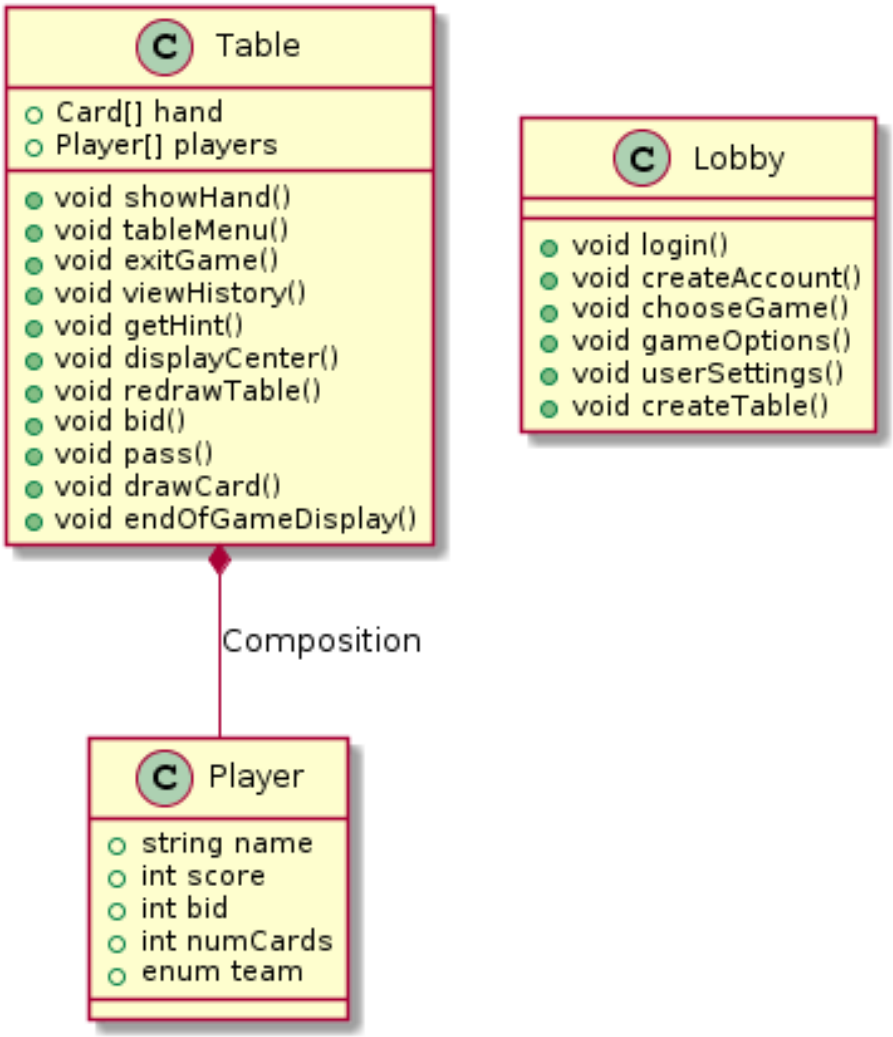
SDD -Client Low Level Design

Brandon Smith, Nieka Gutenberger, Joseph Coppin, Ryan Frazier, Trevor Jewkes

October 8, 2016



Client Class Diagram

1 Table

1.1 Card[] hand

An array of cards holding each players hand.

1.2 Player[] players

An array of player objects.

1.3 void showHand()

A function that prints out the players hand onto the GUI.

1.4 void tableMenu()

A menu dropdown that includes the following: (plus any other options we see fit to add in the future)

1.4.1 void exitGame()

Facilitates a call to the server about early game withdrawal.

1.4.2 void viewHistory()

Shows a history of each trick in the given round

1.4.3 void getHint()

Provides the user with a good playable card. (Possibly using the logic from the hard AI)

1.5 void displayCenter()

Controls the display of the center play area as the game is played.

1.6 void redrawTable()

Updates the table after each move is made, in between tricks, and after each round.

1.7 void bid()

Used in Spades to handle the bid scenario.

1.8 void pass()

Used in Hearts to handle the pass scenario

1.9 void drawCard()

Used in Crazy 8s to handle the draw card scenario.

1.10 void endOfGameDisplay)

Is called at the end of a game to display winners, everyones final scores, and an option to play again or return to the main menu.

2 Player

2.1 string name

String that holds the name of a player.

2.2 int score

Int that holds the score of a player.

2.3 int bid

Int that holds the value bidded in a game of Spades.

2.4 int numCards

Int that holds the number of cards player has left to play.

2.5 enum team

Marks player as in team 1, team 2 or on no team depending on game played.

3 Lobby

3.1 void login()

This function allows for the user to enter their credentials and will send a login request to the server.

3.2 void createAccount()

This function allows for the user to create an account and send that information to the server

3.3 void chooseGame()

This function will take the user's request to start one of the games and send the corresponding request to the server.

3.4 void gameOptions()

This function will take the user's request to play a public or private game.

3.5 void userSetting()

This function will open the window for the user to see their stats and a set amount of options.

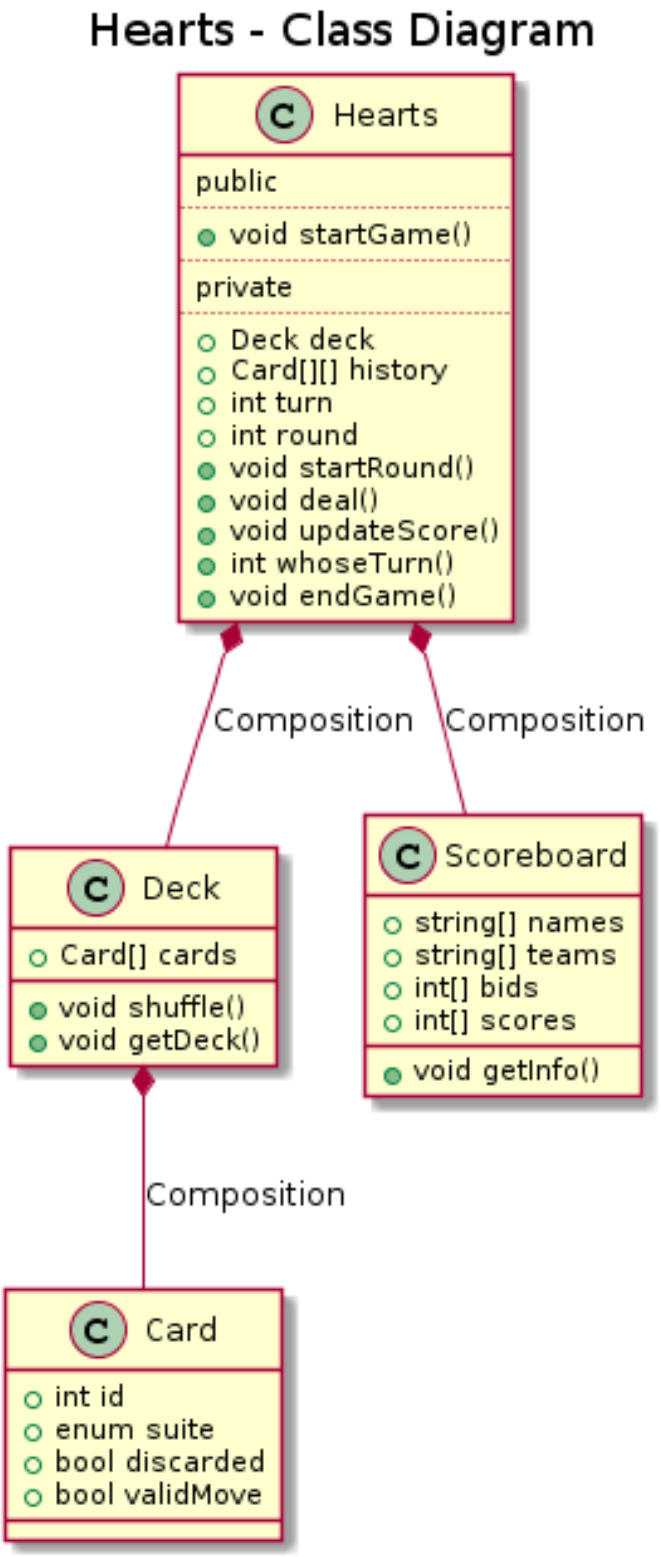
3.6 void createTable()

This function will be called to start the game and will create the table for play to begin. It will also send a start game request to the server.

SDD -Hearts Low Level Design

Brandon Smith, Nieka Gutenberger, Joseph Coppin, Ryan Frazier, Trevor Jewkes

October 8, 2016



Hearts Low Level Design Diagram

1 Hearts Class

1.1 Deck deck

Object deck which holds an array of 52 cards.

1.2 Card[][] history

Array holds previous cards in play, allows client to view history.

1.3 int turn

Variable to hold turn number for use in game logic.

1.4 int round

Variable to hold round number for use in game logic.

1.5 void startRound()

Start round will first update each client with their hands and then ask which cards need to be passed, it then will call a private function take turn.

1.6 void deal()

This function gives each player the appropriate cards at the beginning of each game or round.

1.7 void updateScore()

This function updates the score after each player goes (or after each round depending on specific game)

1.8 void whoseTurn()

This function keeps track of which player is next to play.

1.9 void endGame()

This funcion allows the client to exit or play an additional game.

2 Scoreboard Class

2.1 string[] names

String of player name.

2.2 string[] teams

String of team player is on.

2.3 int bids

Int of player bid.

2.4 int scores

Int of player score.

2.5 void getInfo()

This function calculates and updates information needed for displaying score for player.

3 DeckClass

3.1 Card[] cards

This is an array (of size 52) of card objects to be used in a game.

3.2 void shuffle()

This function changes the id values to different array elements to randomize a deck to be played in a game.

3.3 void getDeck()

This allows the game logic to pull the information of the Deck class and use it for a game.

4 Card Class

4.1 int id

This variable represents and corresponds to a specific card in a standard playing deck.

4.2 enum suite

The card object will be one of four suites, enumerated to represent hearts, diamonds, spades, and clubs.

4.3 bool discarded

This indicates whether a card has been discarded or played in a game.

4.4 bool validMove

This indicates whether a card is playable in the current hand of play.