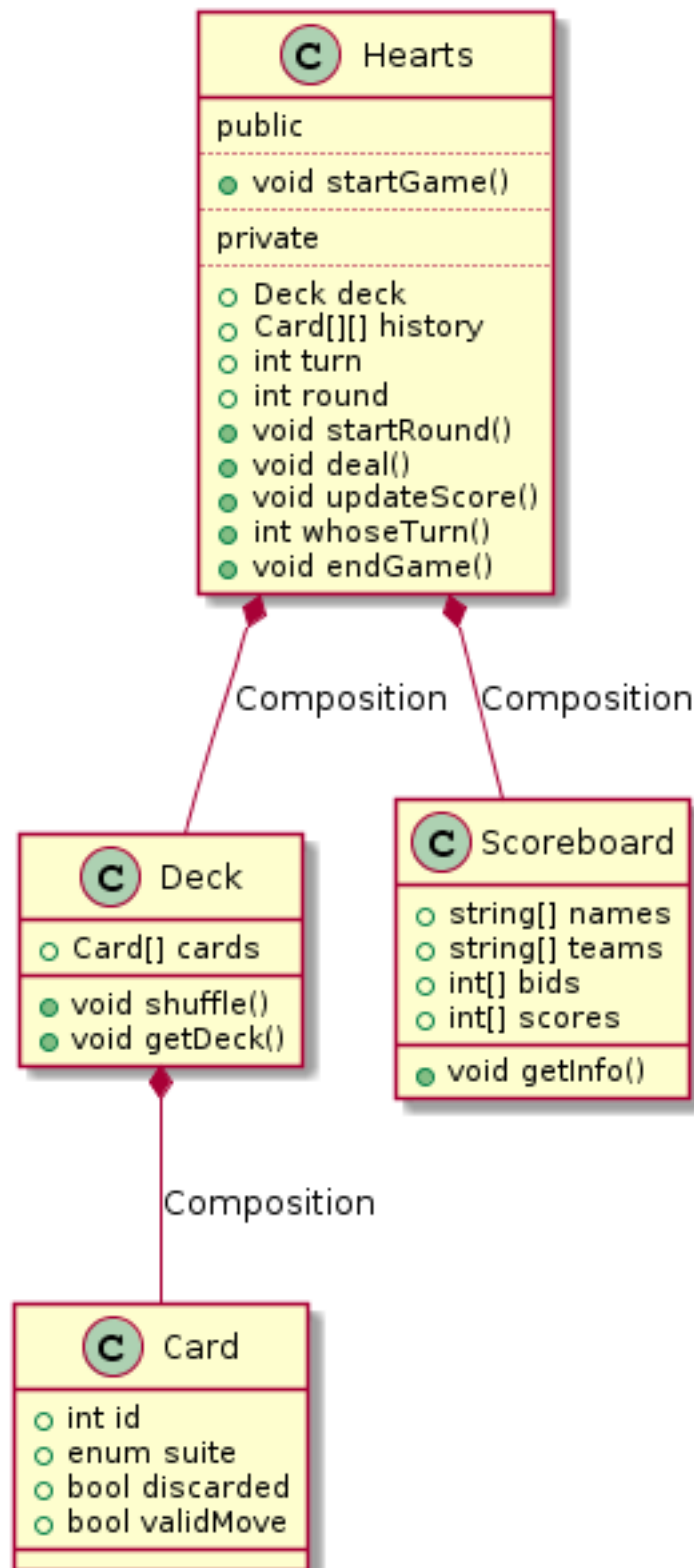


# **SDD -Hearts Low Level Design**

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

November 2, 2016

# Hearts - Class Diagram



Hearts Low Level Design Diagram

## 1 Hearts Class

### 1.1 `std::string suite(int i)`

converts enum ints to string of suit

## **1.2 std::vector<Card> initializeDeck()**

creates deck of cards taken from Card.cpp

## **1.3 std::string makename(int n)**

converts Player + int into Player names

## **1.4 std::vector<Player> creatPlayers(int p)**

Creates a vector of Players to play the game.

## **1.5 std::string value(int i)**

## **1.6 void displayCards(std::vector<Card> & hand)**

displays the deck for screen purposes.

## **1.7 void dealCards(std::vector<Player> & players, std::vector<Card> & Deck)**

## **1.8 int findTwoOfClubs(std::vector<Player> & p)**

looks through each hand to find the 2 of clubs

## **1.9 Card getCardsToPass(std::vector<Card> & h, std::string p)**

gets and stores cards for passing

## **1.10 int fixPass(int r, int p, int c)**

## **1.11 void assignPassedCards(std::vector<Player> & p, std::vector<Card> & h, int r)**

takes the passed cards and redistributes based on round

## **1.12 int passCards(std::vector<Player> & p, int round)**

function for passing cards at beginning of round

## **1.13 int scoretrick(std::vector<Card> & center, std::vector<Player> & players, int& turn)**

## **1.14 void displayroundScores(std::vector<Player> p)**

## **1.15 void displayScores(std::vector<Player> p)**

## **1.16 bool allhearts(std::vector<Card> h)**

checks to see if a players hand is all hearts.

## **1.17 bool noLeadSuit(Suit s, std::vector<Card> h)**

compares hand against the lead suit

## **1.18 bool validateMove(Player& p, bool& broken, std::vector<Card> Center, Card move, int t, int i)**

## **1.19 Card getmove(Player& p, bool& b, std::vector<Card> c, int t, int i)**

## **1.20 void updateScore(std::vector<Player> & p)**

adds round score to Score

**1.21** `bool playRound(std::vector<Player> & players, std::vector<std::vector<Card> >& history, int round)`

**1.22** `void startGame(std::vector<Player> & players, std::vector<std::vector<Card> >& history)`

uses `players` and calls `round` until game is over

**1.23** `void play_Hearts(int num)`