Kevin Yang- kyang01@college.harvard.edu

Christina Chen- christinachen@college.harvard.edu

Diana Smith- dianasmith@college.harvard.edu

Michael Farrell- michaelfarrell@college.harvard.edu

Texas Hold’em – Final Spec

**Signatures/Interfaces:**

Player: The general class that has children that are AI’s as well as human players.

* Private Variables
  + Money – Will remain private because users should not be able to alter the amount of money directly without using a helper method.
  + Bet- The amount of money the player is betting, also shouldn’t be allowed to be altered from outside.
  + IndividHand – The cards the player has should be kept private so not to be seen by others.
  + Hand- Full hand including the cards on the table, specific to each person because they have an additional two cards.
* Methods
  + getMoney – returns money
  + playBlind – Allows the player to play the blind, alternating between big and small
  + isAllI- any player can be all in and so this is defined here, allows the interface to be aware that the rules are changing because someone is all in
  + addMoney – Updates a players funds after winning a hand
  + clearHand – removes the cards in a hand so the next round can begin
  + addToHand- adds a card to the hand
  + getIndividHand- returns the players hand
  + adjustMoney- removes money after a bet is made
  + getWholeHand- allows the whole 7 card return for a player in order to check what hand that player has at the momeny.

All of the methods are used in order to keep some of the variables private, and the methods within the players class are universal to any player in the game whether they are human or not. This allows us to treat a computer and a human the same when comparing their hands and making bets and other such mundane actions in poker.

Computer: The class that holds for any computer AI’s

Human: The human class

* Unique to both is the make Bet class because the human will make their own bet while the computer must calculate their chances to make a bet and has helper functions to create that bet

Card: General class declaring a card that is inherited by different classes of each type of card (Ace, king, queen….)

* Private variables:
  + valueOfCard- Returns the value of the corresponding card
  + suitOfCard
* Methods
  + returnValue
  + returnSuit
  + toString- Turns a card into a string in order to be viewed
  + CompareTo- Allows a comparison between cards by comparing their values

**Timeline:**

Week of 4/20

* Finish general game setup
* Complete research of game AI’s
* Write out the pseudocode or background for the computer AI’s
* Complete statistic calculator for AI’s to figure out their chances

Week of 4/27

* Begin assembly GUI
* Complete AI and test against humans to find holes in the system

**Progress Report**

At the moment we are debating between making one AI and split up the implementation or rather each creating different ones.

We attached the main files for the key components of the game