

Design Manual

For our final project, our group had the idea of creating a GUI focused game of Texas Holdem' Poker. In order to complete this idea we used the design format of model, view, and controller classes, otherwise known as MVC design. The model classes help hold the logic of the poker game itself. These classes will take in the information from the controller classes and interpret how the game would handle it and spits back the results to the controller classes based on what methods and classes were called in the model classes. The view classes are what present the GUI to the user and take information from the controller classes in order to change the appearance of the GUI based on the new information that comes from the controller classes via the model classes. The controller classes are the middle man between view and model. They will take information from either view or model and configure it so that the other one can use it and help the game progress. We used this model to condense the class sizes and make it a more organized design for the game overall.

User Stories

As a dealer , I want to be able to give each player two cards so that hands can be dealt
As a dealer , I want to be able to deal the flop
As a dealer , I want to be able to deal the turn
As a dealer , I want to be able to deal the river
As a dealer , I want to deal no duplicate cards
As a dealer , I want to be able to judge which hand won the pot
As a dealer , I want to deal cards first to the player that was dealt cards second the previous hand
As a dealer , I want to be able to deal cards clockwise
As a dealer , I want to deal the first cards on the first hand to the player directly to my left
As a player , I want to be able to choose the starting chip amount I have

As a player , I want to be able to see the cards I am dealt for my hand
As a player , I don't want any other player to be able to see the cards I am dealt for my hand
As a player , I want to be able to call, raise, or fold before the flop
As a player , I don't want to be able to check or bet before the flop
As a player , I want to be able to call, raise, or fold after the flop
As a player , I want to be able to bet if no other player has bet after the flop
As a player , I want to be able to check if I do not have to match another player's bet after the flop
As a player , I want to be able to call, raise, or fold after the turn
As a player , I want to be able to bet if no other player has bet after the turn
As a player , I want to be able to check if I do not have to match another player's bet after the turn
As a player , I want to be able to call, raise, or fold after the river
As a player , I want to be able to bet if no other player has bet after the river
As a player , I want to be able to check if I do not have to match another player's bet after the river
As a player , I want to show my cards if there is a showdown
As a player , I want to win the game when I have all the chips
As a player , I want to be out of contention for the game when I have no chips left
As a player , I want to have no time limit when making a decision about my hand
As a player , I want to win all the chips in the pot if I have the best hand at the end of the hand
As a player , I want to be able to get half the pot if there is a tie
As a player , I want to make an initial bet (blind) before the hand is dealt
As a player , I want to make the small blind if I am the first one receiving cards for this hand
As a player , I want to make the large blind if I am the second one receiving cards for this hand
As a player , I want to have the option to show my hand to other players after the end of the hand if there is no showdown
As a player , I want to have the option to show my chip count to other players
As a player , I want to choose how many opponents I am against
As a player , I want to choose the color of the poker table
As a player , I want to choose the color that I appear as
As a player , I want to choose my name

Object Oriented Design

Our project uses many objects interacting with each other. Some of them are GamePot, PlayerChips, Card, Hand, River, and Player. The GamePot and PlayerChips interact

with each other in order to exchange the money. Whenever a player places a bet or raises the bet in a game, the money is then taken out of the player's chip stack and placed into the game pot. Once all beginning player and computer's bets are placed, the game may begin. Players are dealt their cards and the river gets proceeded to be played out. As each part of the river is laid out the player can choose to check, call, fold, or raise. Once the river is played out, the player and computers begin to see who is bluffing and who is not. Once the winner is determined, the pot money is then given to the winner or split between the two players in the chance of a tie. The player can then choose whether or not to play again.

PokerMain <ul style="list-style-type: none"> - Set up GUI - Set up logic <ul style="list-style-type: none"> - Controller classes - View classes - Game logic manager class 	GameView <ul style="list-style-type: none"> - Set up GUI for the actual Poker game <ul style="list-style-type: none"> - SettingsController - PokerMain - MenuController 	MenuView <ul style="list-style-type: none"> - Set up GUI for the initial menu that the user is brought to <ul style="list-style-type: none"> - PokerMain 	GamePot <ul style="list-style-type: none"> - Handle how many chips are currently in the pot for the hand
SettingsView <ul style="list-style-type: none"> - Set up GUI for the settings menu that holds customization options <ul style="list-style-type: none"> - PokerMain 	ViewUtility <ul style="list-style-type: none"> - Utility class that provide final variables so that the use of magic numbers is avoided 	GameController <ul style="list-style-type: none"> - Set up connection between the game's GUI view and the logic behind the game's GUI <ul style="list-style-type: none"> - GameView - PokerMain - PlayerChips - Dealer 	PlayerChips <ul style="list-style-type: none"> - Handle how many chips an individual player has for a game
MenuController <ul style="list-style-type: none"> - Set up connection between the menu's GUI view and the logic behind the menu's GUI <ul style="list-style-type: none"> - MenuView - SettingsView - GameView - PokerMain - Dealer - GamePot 	SettingsController <ul style="list-style-type: none"> - Set up connection between the settings's GUI view and the logic behind the setting's GUI <ul style="list-style-type: none"> - SettingsView - MenuView - PokerMain 	Dealer <ul style="list-style-type: none"> - Handle adding money to the game pot - Handle taking money away from player chips - Handle giving money to winner(s) of a hand <ul style="list-style-type: none"> - PlayerChips - GamePot - PokerMain 	PlayerEnum <ul style="list-style-type: none"> - Properly sets up the different colors that a player can appear as
PokerChip <ul style="list-style-type: none"> - Properly sets up the values of the different chips that may be used to bet 	PokerTableEnum <ul style="list-style-type: none"> - Properly sets up the different colors that the poker table can appear as 	AnalyzeHands <ul style="list-style-type: none"> - Determines the kind of hand (three of a kind, flush, etc.) the player has including the flop, turn, and river <ul style="list-style-type: none"> - River - Hand 	CPU <ul style="list-style-type: none"> - Handle what action the computer will make - Handle channing how smart or risky the CPU is <ul style="list-style-type: none"> - PlayerChips - GamePot

<div>Card</div> <div><ul style="list-style-type: none">- Contains the suit of the card- Determines the rank of the card- Determines the color of the card based on suit</div>	
<div>DeckOfCards</div> <div><ul style="list-style-type: none">- Contains a full deck of standard Poker cards- Handles shuffling Cards</div>	<ul style="list-style-type: none">- Card
<div>Hand</div> <div><ul style="list-style-type: none">- Contains the cards that an individual player has for the hand</div>	<ul style="list-style-type: none">- Card
<div>River</div> <div><ul style="list-style-type: none">- Contains the cards that the dealer deals for the flop, river, and turn</div>	<ul style="list-style-type: none">- Card
<div>GameLogic</div> <div><ul style="list-style-type: none">- Sets up the entirety of the game logic</div>	<ul style="list-style-type: none">- GamePot- River- DeckOfCards- Dealer
<div>Player</div> <div><ul style="list-style-type: none">- Abstract class that sets methods for the player and the CPU players</div>	<ul style="list-style-type: none">- GamePot- Hand- PlayerChips

