StakeFree Digital Casino: Sprint 3

EECS Team 6

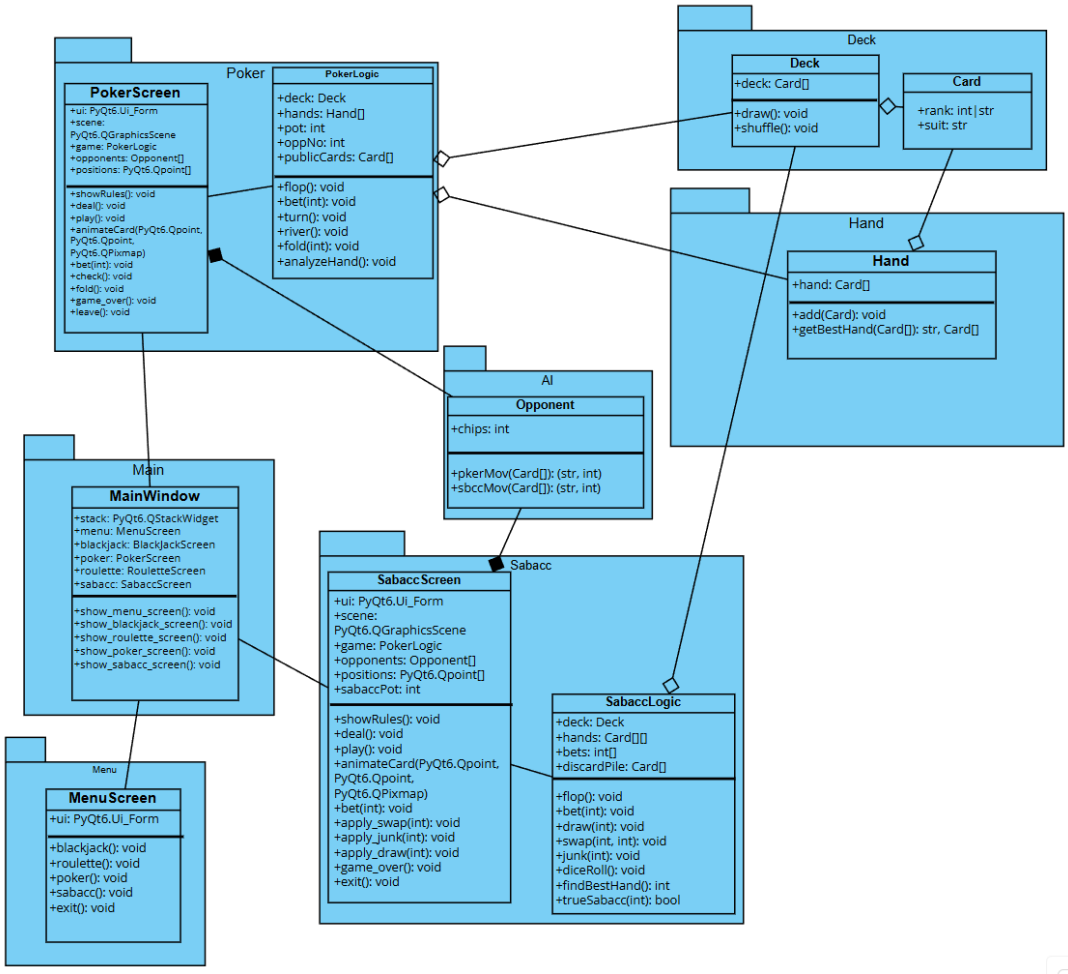
Bisshoy Bhattacharjee, Maximilian Biundo, Gavin Billinger, Mark Kitchin, Joshua Welicky

Project Synopsis

StakeFree Digital Casino provides gamblers with the opportunity to play games such as Roulette, Blackjack, Poker, and Sabacc without the risk of financial loss.

Architecture Description

The UML Class Diagram for the Sprint 3 architecture of StakeFree Digitial Casino is shown below.



Dependencies

This project utilizes the PyQt6 libraries to implement GUIs. To install all required libraries, enter “pip install pyqt6-tools". This library provides a designer program that allows for the customization of a GUI in a user-friendly manner, saving the resulting GUI as a .ui file. This project also takes advantage of another tool provided by PyQt6, pyuic6, which converts a .ui file into a Python class for ease of use. Each element in the GUI is made available as a class attribute.

Components

MainWindow

- Description:
  - Core skeleton of the project. Manages switching between the menu GUI and game GUIs, as well as ultimate program termination.
- Attributes:
  - stack: A PyQt6.QStackedWidget instance that will store all of the GUI handlers, allowing switching.
  - menu: A MenuScreen instance which handles the main menu GUI.
  - blackjack: A BlackJackScreen instance which handles the blackjack GUI.
  - poker: A PokerScreen instance which handles the poker GUI.
  - roulette: A RouletteScreen instance which handles the roulette GUI.
  - sabacc: A SabaccScreen instance which handles the sabacc GUI.
- Methods:
  - show_[screen]_screen():
    - Simply switches to the desired screen (menu, blackjack, poker, sabacc, or roulette).

MenuScreen

- Description:
  - A very simple GUI manager for the menu screen. It merely contains buttons that redirect to the games.
- Attribute:
  - ui: A PyQt6.Ui_Form that generates and manages the menu GUI.
- Methods:
  - blackjack(): Redirects to the blackjack game.
  - roulette(): Redirects to the roulette game.
  - poker(): Redirects to the poker game.
  - sabacc(): Redirects to the sabacc game.

- exit(): Requests clean program termination.

Card

- Description:
  - Basic representation of a playing card.
- Attributes:
  - rank: The integer value for a numbered card, the string name for a face card or ace.
  - suit: A string name of the suit of the card (spade, club, heart, diamond).

Deck

- Description:
  - A representation for a deck of cards.
- Attributes:
  - deck: An array of Card instances.
- Methods:
  - draw():
    - Randomly selects a Card instance and removes it from the deck attribute, returning the Card as well.
  - shuffle():
    - Reinitializes the deck attribute, clearing it and replacing every Card instance.

Hand

- Description:
  - A representation of a Poker player's hand, which is their specific two cards.
- Attributes:
  - hand: An array of Card instances.
- Methods:
  - add():
    - Simply adds a Card instance to the hand.
  - getBestHand():
    - Takes in the public cards of the Poker game. It then determines the best possible hand, using the Card instances of hand and the public cards. It returns the string name of the best possible hand and an array of the Card instances used to construct it.

Opponent

- Description:
  - Represents an opponent in a game of Poker or Sabacc.
- Attributes:
  - chips: A randomly generated number of chips available for the opponent to bet.
- Methods:
  - pkerMov(Card[]):
    - Implements an opponent's decision on whether to bet, check, or fold during a turn in poker. Takes in an array of Card instances, which represents the opponent's hand. Returns a tuple containing a string choice("bet", "check", "fold") and an integer. The integer will represent the number of chips to bet if the choice is to bet. The number will be unused otherwise(set to zero).
  - sbccMov(Card[]):
    - Similarly implements the opponent's decision in a sabacc game. Returns a similar tuple, containing the determined move and possible bet amount.

PokerScreen

- Description:
  - Manages the GUI displayed while playing Poker. Connects actions on GUI elements to blackjack moves.
- Attributes:
  - ui: A PyQt6.Ui_Form that generates the initial state of the poker GUI.
  - scene: A PyQt6.QGraphicsScene that enables the addition of animated sprites to the GUI.
  - game: The PokerLogic instance handling the actual moves of the game.
  - deck_pos: A PyQt6.QPoint that represents the card deck's position on the GUI. Used for draw animations.
  - opponents: An array of Opponent instances, which represent the opponents in the game.
  - positions: An array of PyQt6.QPoint instances that stores the GUI positions of the player and all three AI opponents.
- Methods:
  - showRules():
    - Merely displays a form explaining the rules of poker to the user.
  - deal():

        - Initiates the initial dealing of the game, including the card animations and initial revealing of the public cards.
    - play():
        - Main function managing the gameplay loop (managing the exectutions of AI opponents' actions).
    - animateCard():
        - Manages the actual animation of the cards.
    - bet():
        - Used to effect a player's, specified by the integer parameter, bet, showing its addition to the pot.
    - check():
        - Used to move along the game.
    - fold():
        - Used to implement folding, effectively ending the game if the player folds.
    - game_over():
        - Manages end-game chip rewarding and game state resetting.
    - leave():
        - Simply exits to the main menu.

PokerLogic

- Description:
    - Main game logic class for the Poker gameplay.
- Attributes:
    - deck: A Deck instance representing the card deck.
    - hands: An array of Hand instances, one for each player.
    - pot: An integer storing the current prize amount for the game.
    - oppNo: The number of opponents specified by the user.
    - publicCards: The cards visible by all players, used to construct their hand.
- Methods:
    - flop():
        - Adds the first three public cards to the publicCards array.
    - bet():
        - Adds a bet to the pot.
    - turn():
        - Adds a card to the publicCards array.
    - river():
        - Also adds a card to the publicCards array.
    - fold():
        - Used to forfeit a player during the game.

    - analyzeHand():
        - Interacts with the Hand's methods to obtain the best possible hand for a player.

SabaccLogic

- Description:
    - Main game logic class for the Sabacc gameplay. Will be further fleshed out to integrate with the GUI later.
- Attributes:
    - deck: A Deck instance representing the card deck.
    - hands: An array of arrays of Card instances, one for each player.
    - bets: An array representing the chips bet by each player.
    - discardPile: A Card instance array representing the discard pile.
- Methods:
    - flop():
        - Initial assignment of hands to all players and discard pile.
    - bet():
        - Handles adding chips to the bet array.
    - draw():
        - Adds a card to a player's hand.
    - swap():
        - Swaps a specified card with the card at the top of the discard pile.
    - junk():
        - Handles discard pile additions from a player's forfeit.
    - diceRoll():
        - Implements the double-dice role at the end of each round, and the effects if has on players' hands.
    - findBestHand():
        - Determines whose hand is the closest to zero, and therefore the winner.
    - trueSabacc():
        - Simple function that will determine if a player's hand is a true sabacc, and therefore the best hand in the game.