StakeFree Digital Casino: Sprint 3

EECS Team 6

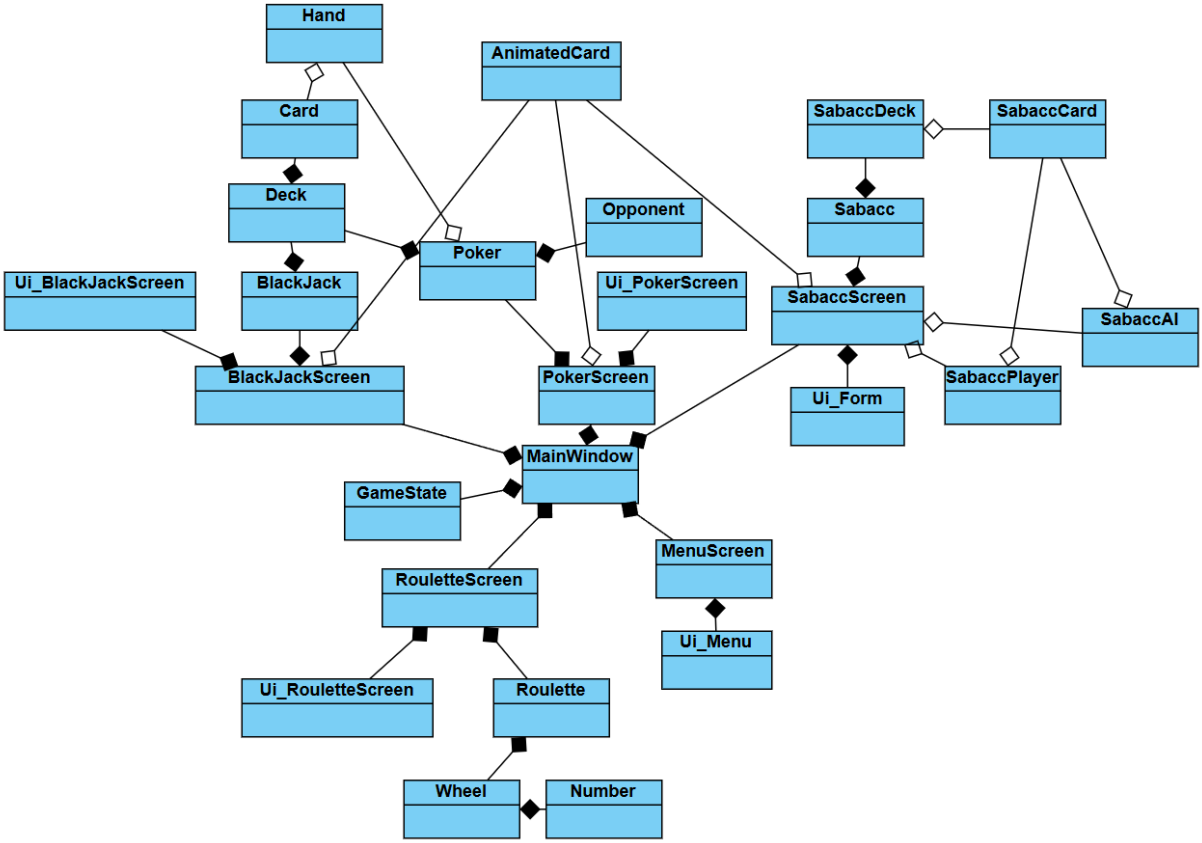
Bisshoy Bhattacharjee, Gavin Billinger, Maximilian Biundo, Mark Kitchin, Joshua Welicky

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 **high-level** UML Class Diagram for the 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

A high-level description of the architectural components is given below:

MainWindow

Core skeleton of the project. Manages switching between the menu GUI and game GUIs, as well as ultimate program termination.

GameState

Holds the global amount of the player's chips across games. This is passed to each of the GameScreen classes, further described below.

MenuScreen

A very simple GUI manager for the menu screen. It merely contains buttons that redirect to the games.

Ui_MenuScreen

A custom PyQt6.Ui_Form that is used to generate and manipulate the menu GUI.

[GAME]Screen

Each game implemented in StakeFree Digital Casino has a corresponding [GAME]Screen class.

- BlackJackScreen for BlackJack
- RouletteScreen for Roulette
- PokerScreen for Poker
- SabaccScreen for Sabacc

Each of these connects their corresponding [GAME], defined below, logic classes with the actual Ui_[GAME]Screen classes, defined below, to effectuate game logic actions in the actual GUI.

[GAME]

Each game implemented in StakeFree Digital Casino has a corresponding [GAME] class.

- BlackJack for BlackJack
- Roulette for Roulette
- Poker for Poker
- Sabacc for Sabacc

These are the game logic functions. They carry out the actual playing of each game. After these classes perform game actions, the corresponding [GAME]Screen class implements the change in the GUI.

Ui_[Game]Screen

Each game implemented in StakeFree Digital Casino has a corresponding Ui_[GAME]Screen class.

- Ui_BlackJackScreen for BlackJack
- Ui_RouletteScreen for Roulette
- Ui_PokerScreen for Poker
- Ui_Formfor Sabacc

Each on of these is a custom PyQt6.Ui_Form that is used to actually generate and manipulate GUI for the corresponding game by the [GAME]Screen class.

Card

A basic representation of a standard playing card used in Poker or BlackJack. It stores useful information, such as suit and rank

SabaccCard (aka Sabacc_Card)

A basic representation of a Sabacc playing card, storing useful information such as suit, rank, and sign (positive or negative).

Deck

A representation of a standard 52-card deck as used in Poker and BlackJack. It includes methods for shuffling a deck and drawing from it.

SabaccDeck(aka Sabacc_Deck)

Almost an exact copy of the Deck class, only configured for a 62-card Sabacc deck. It also has functionality for shuffling and drawing

AnimatedCard

This is a PyQt6 QGraphicsObject configured to serve as an animation for a moving card. It is configured to be used in both Sabacc, Poker, and BlackJack.

Number

This class represents a number as understood in Roulette. It contains important properties of a number, such as its color, row, column, which third of the range it is in, among other properties.

Wheel

This represents a Roulette wheel, and stores a collection of Number instances, in the order they appear on a physical wheel. It contains functionality for "spinning", which simply returns the index of the number the wheel randomly "landed" on.

SabaccPlayer

A class that represents the human player during Sabacc mode. It stores various Sabacc-related items for ease of access, including the current hand; an array to store AnimatedCard widgets for each card in the hand; the player's chips; the player's stake in the current game; and more. Also contains functionality for returning the value of the player's hand.

SabaccAI

A class that contains the same information and functionality as SabaccPlayer, but this represents a digital Sabacc opponent. As such, it contains additional functionality to make a game decision

based on its hand, the current discard pile, and the round number. It also contains functionality to make betting decisions based on the quality of its hand.

Opponent

A class similar to the SabaccAI class, but for Poker. It stores very similar information. However, it only makes Poker-related decisions. It bases on the strength of its hand, with some randomness integrated for unpredictability.