Final Project Proposal - Blackjack Analysis

Overview of Blackjack

Blackjack is a common casino game that involves luck and strategy. Depending on the casino, there are up to five players and a dealer. Players are originally dealt two cards with each card having a numerical value associated with it. Aces can be played as either an eleven or a one, cards Two through Ten have a numerical value that is associated with its rank (Two, Three, etc.) and face cards (Jack, Queen, and King) have a value of 10. The goal of blackjack is to obtain a final hand value that is greater than that of dealer's. If any player or dealer has a hand total of 22 or higher they lose on the current hand that they are playing. Players have an opportunity to split hands if they have duplicate ranks in their hand and they have the currency left in their bank to do so. Other options are available to players, such as doubling down when they have enough currency to do so, as well as buying insurance if a dealer is showing an Ace and they have the currency to place their insurance bet as well. For a complete list of rules and objectives in blackjack, please see the following website: **Blackjack** [1].

Goal of Project

The goal of this project is to create a functioning blackjack game that will process the logical steps in a user playing blackjack. On top of the typical logic that is involved in playing blackjack, the extended goal of this project is to create an analysis of a blackjack game as well as providing strategies that better help a player gain an advantage while playing blackjack. These strategies include giving a player hints to how they should play as well as a training aspect that teaches players how to count cards (a common strategy for gaining an advantage while playing) in a game of blackjack. Along with these strategies an analysis will be conducted after a game has been finished to show players how they faired after playing. If time permits, a simulation aspect will be implemented to show how a user will fair if they play with perfect blackjack strategy.

Files & Classes

For this program to run, the following classes and files will be created for the program to function:

- **Binary Search Tree.hpp** - This file includes the structure that contains the templated binary search tree that will be used in this program.
- **Card.h & Card.hpp** - These files make up a Card class that represents a Card object that has multiple string values representing things such as rank and suit as well as an integer value that represents the cards value. In this class, there is a private data member called 'PlayingCard' that is a structure that encapsulates all of data members that are necessary to represent a card.
- **Core.hpp** - These files encapsulate all the previous classes and functions that are used to make the classes operate. These files mainly implement the previous classes to produce the game logic as well as other logic.
- **Hand.h & Hand.hpp** - These files make up a Hand class that represents a Hand object that is symbolic of a players hand while playing blackjack. Inside the Hand class their is a private data member called 'Player' where a myriad of boolean, floats, integers, strings, and container values that hold data about a players hand. In total, as this proposal is being written, there is a total of 37 data members that belong to this structure that encapsulate the information of a player.
- **Includes.h** - This file includes all of standard C++ libraries that are needed to make other classes and functions operate.
- **Linked List.hpp** - This file includes the structure that contains the templated linked list that will be used in this program.
- **Shoe.h & Shoe.hpp** - These files make up a Shoe class that represents a Shoe object that holds the Card objects in a container where they can be pulled from and added to players hands. This class has a private data member called 'Deck' which represents a singular deck that is present in a shoe. The 'Deck' data member is where the Card objects are stored.
- **Utility.hpp** - These files encapsulate functions and data types that are used in other classes and functions in other files.

Some of the aforementioned containers are arrays, vectors, and linked lists. The linked lists in this program are templated, meaning that they can hold an arbitrary data type. This is because each linked list that is being used has to be able to hold different types of data. Instead of creating a linked list that holds all the different data types, I will create the aforementioned templated linked list. The data that is produced from each hand will be stored in a linked list and then pushed into a binary search tree. The data that is being tracked are the wagers that are placed each hand, the net total of each hand, the bank total after each hand, and the total number of hands that have been played. Similar to that of the linked list in this program, the binary search tree will also be templated. After this data has been collected, numerical graphs will be displayed showing the data of the game after playing has ended.

Testing

The testing of this project will be done by testing each class and the operations the functions in the class.

References

[1] "Learn To Play Blackjack." Blackjack, bicyclecards.com/how-to-play/blackjack/. Accessed 19 July 2023.