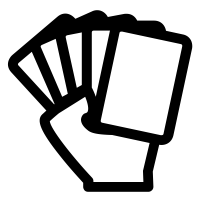
OOD Project

Card Game Interface



Author: Abigail Herron

ID: S00200536

Module: Object Orientated Development

Lecturer: Keith McManus

GitHub Link: <https://github.com/AbigailHerron/ObjectOrientatedDevelopment/tree/main/CardGameInterfaceProject>

Contents

[1. Summary 2](#_Toc67240796)

[2. Idea 2](#_Toc67240797)

[1. Initial Concept 2](#_Toc67240798)

[2. Elaborated Idea 2](#_Toc67240799)

[1. Interface 2](#_Toc67240800)

[2. Possible Classes 3](#_Toc67240801)

[3. Possible Button Logic 6](#_Toc67240802)

[3. First Idea Iteration 8](#_Toc67240803)

[1. Main Window 8](#_Toc67240804)

[3. Stats Class Code 8](#_Toc67240805)

[4. Card Class Code 9](#_Toc67240806)

[5. Deck Class Code 9](#_Toc67240807)

[6. Game Class Code 11](#_Toc67240808)

# Summary

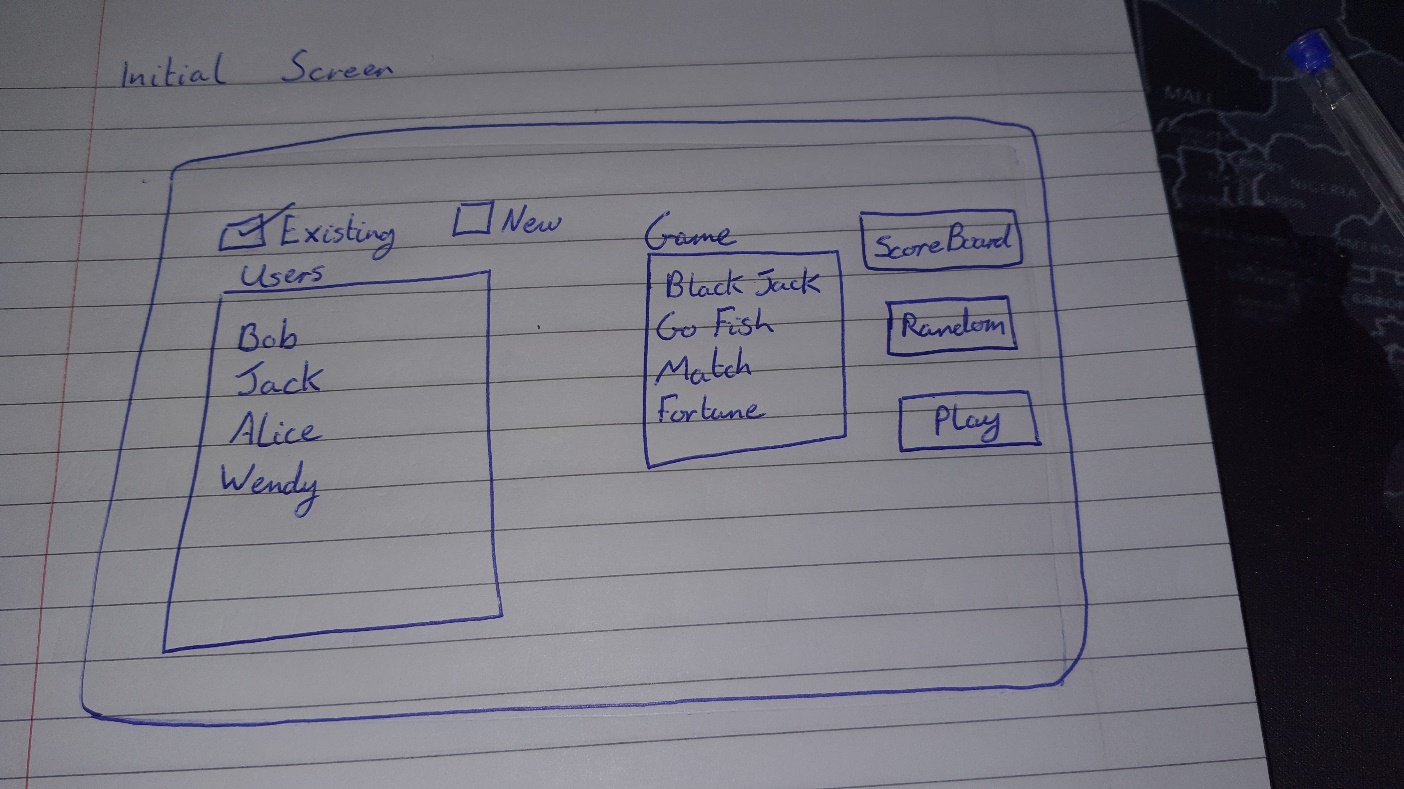
# Idea

## Initial Concept

* Card Game Interface
* Provides a list of profiles to play as
* Lists a few simple card games
* Creates a game based on which game you choose
* Plays game

Can use abstract class ‘Deck’ and then create new Decks based on the kind of game we’re playing

Types of card decks (examples): Normal playing deck, Picture deck, Tarot deck, etc.



## Elaborated Idea

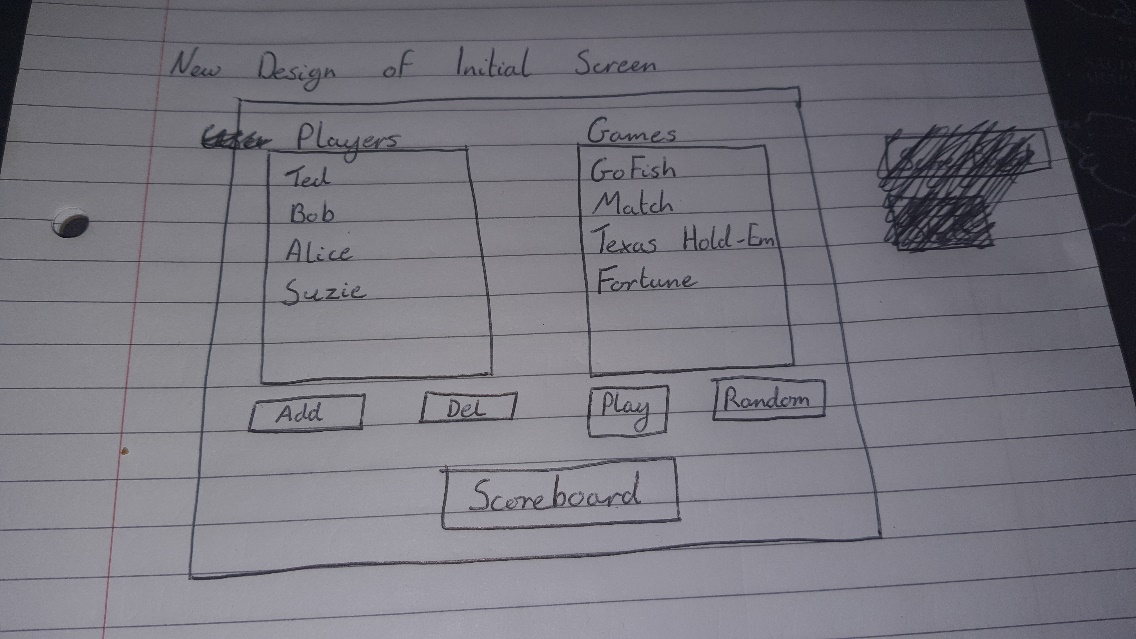
### Interface

Initial window has a listbox populated with a list of player profiles to choose from. These can be added or removed from the list via the buttons underneath the listbox.

It will also contain a listbox populated with a list of simple games that can be played by selected one item, or by hitting the random button which will highlight a selected item in the listbox, and then hitting the play button underneath. This should close the initial window and launch a second one that is designed based on what kind of game has been selected.

***Note:*** *A player profile must be selected before the game can be played. A message box will appear to prompt the user to make this choice.*

The user will also have the option to view the current scoreboard statistics by hitting the scoreboard button at the bottom of the window. This should link to a database where the Wins, Losses, Draws, etc statistics are kept, and updated (‘pushed’ and/or ‘pulled’).



### Possible Classes

|  |  |
| --- | --- |
| Player Class | |
| Properties | String PlayerName  List<Stats> GameResutls {GameType1, GameType2, etc} |
| Methods | PullStats (void)  PushStats (void) |
| Constructor | Default, All |

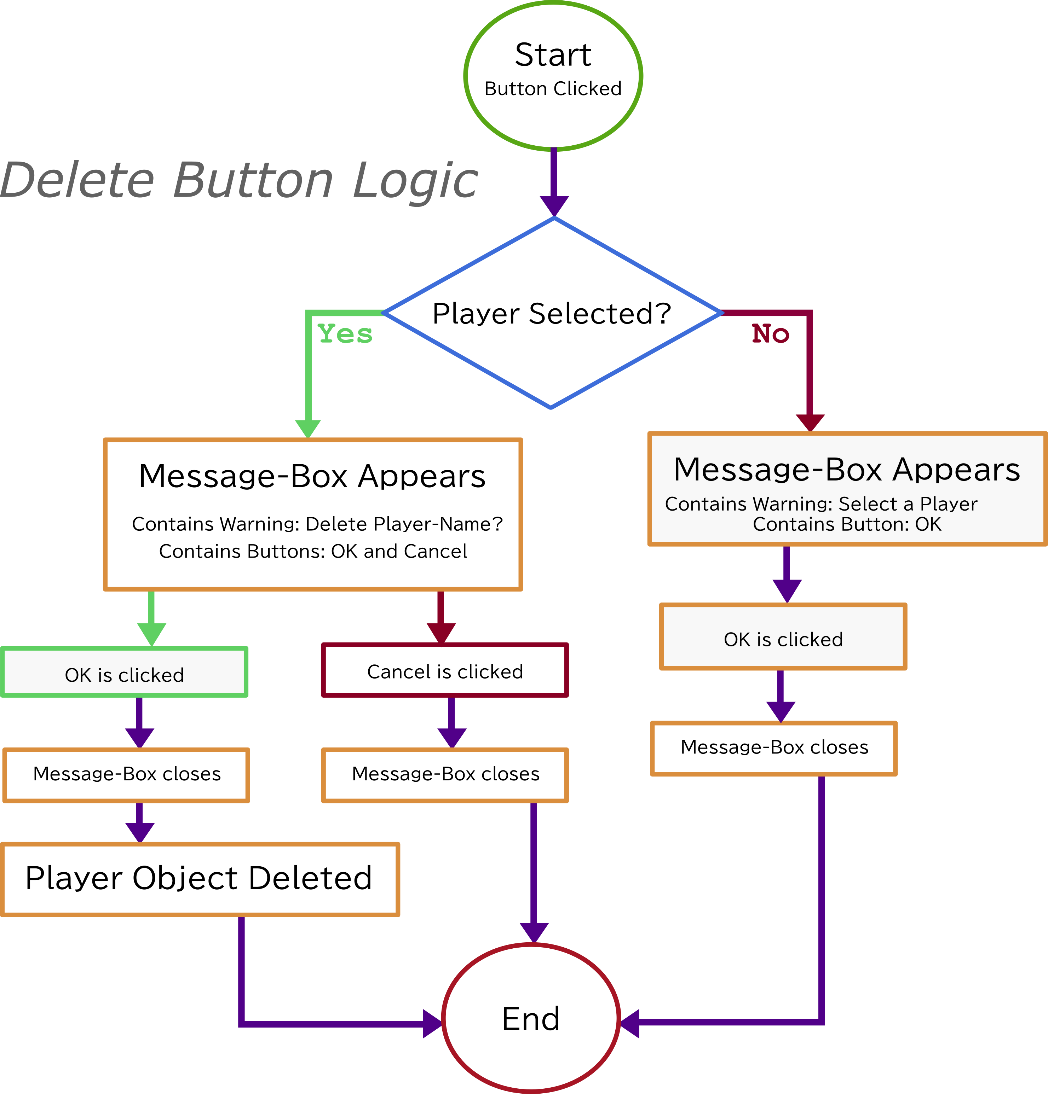
|  |  |
| --- | --- |
| Stats Class | |
| Properties | Int Wins  Int Losses  Int Draws  String GameCode |
| Methods | UpdateStats (void) |
| Constructor | Default |

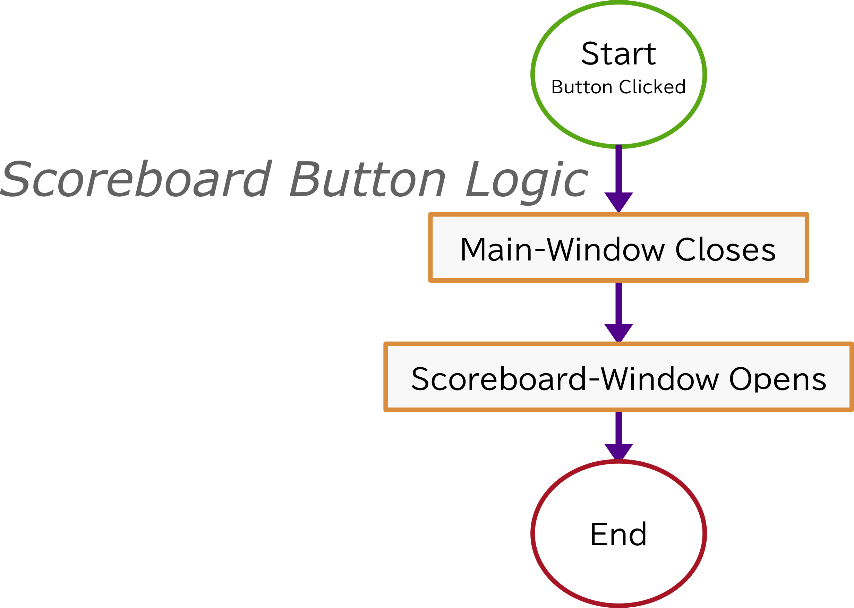
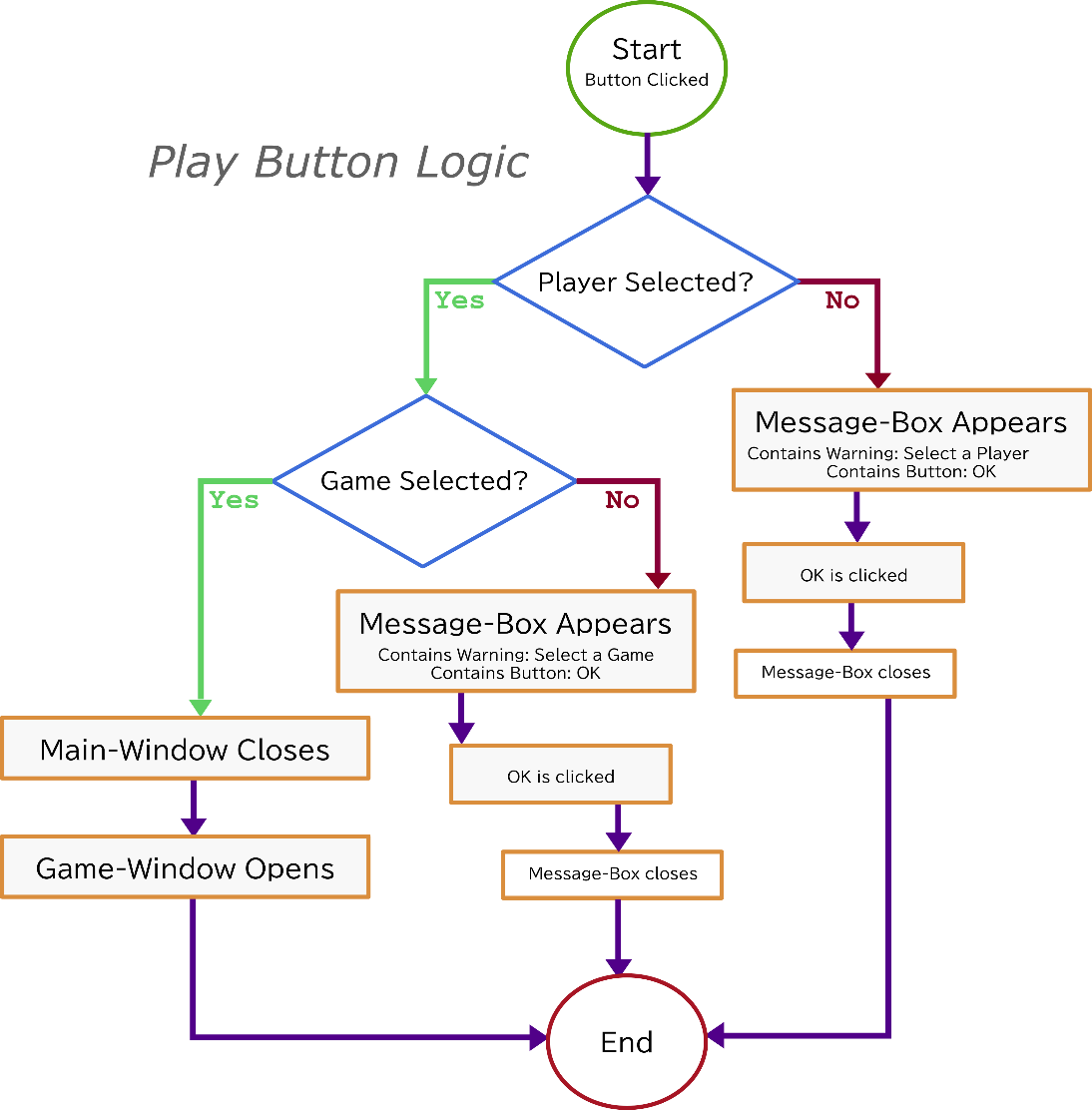
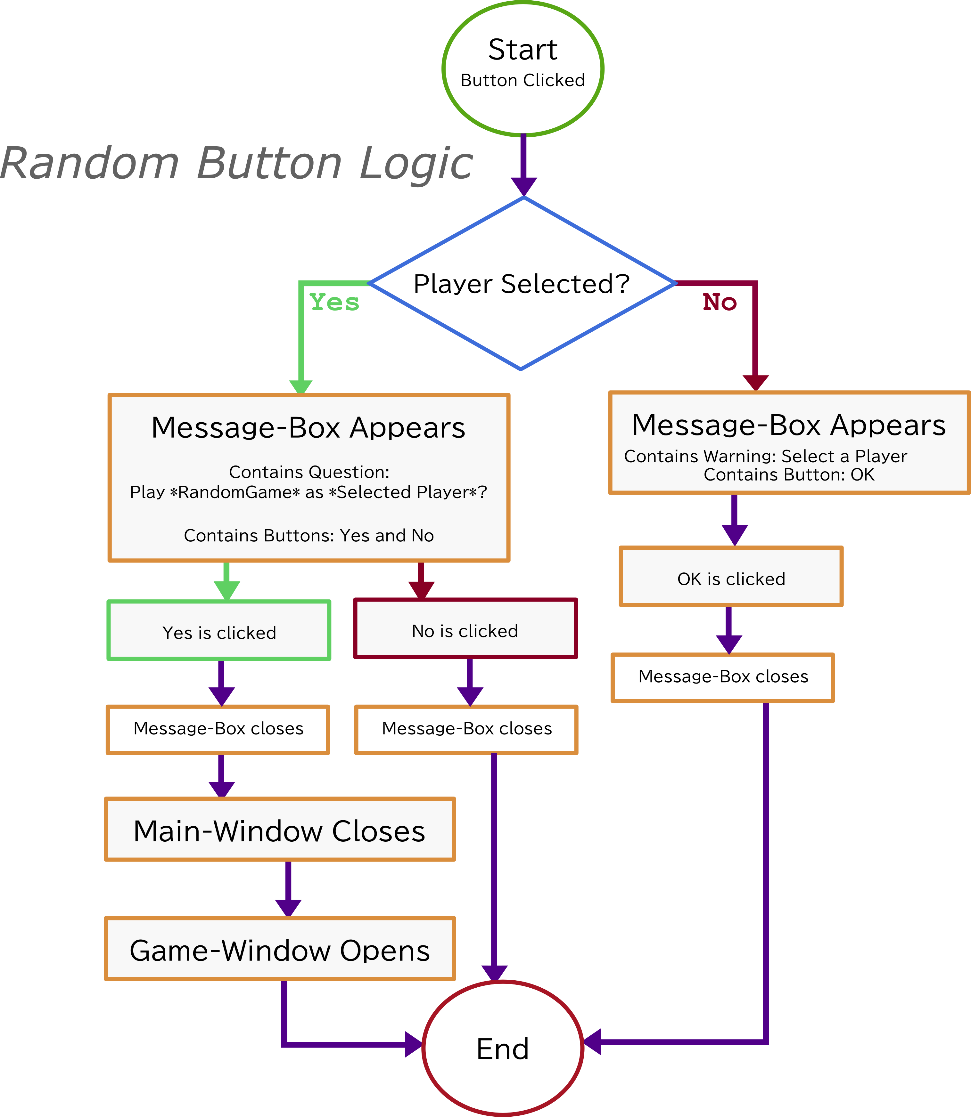
|  |  |
| --- | --- |
| Card Class | |
| Properties | String Suit, String Rank, Int Value, Bool Position |
| Constructor | Default |

|  |  |
| --- | --- |
| Deck Class (Abstract) | |
| Properties | List<Card> Pack |
| Constructor | Default |
| Methods | Shuffle (void) |
| PlayingDeck Class (Sub-Class of Deck) | |
| Inherited Parts | *All but the SpecialCards array* |
| Added Methods | ToString (override) |
| TarotDeck Class (Sub-Class of Deck) | |
| Inherited Parts | *Everything* |
| Added Methods | ToString (override)  Shuffle (override) |

|  |  |
| --- | --- |
| Game Class (Abstract, uses IComparable) | |
| Properties | String GameType  Deck GameDeck |
| Constructor | Default |
| Methods | Deal (void, abstract)  ResetGame (void, abstract)  CompareTo (override) |
| GoFish Class (Sub-Class of Game) | |
| Inherited Parts | *Everything* |
| Added Properties | Int Pairs |
| Added Methods | CheckPairs (void)  CompareTo (override)  PullCards (void) |
| Match Class (Sub-Class of Game) | |
| Inherited Parts | *Everything* |
| Added Properties | Int Pairs  Int TriesLeft |
| Added Methods | IsMatch (bool)  CompareTo (override) |
| TexasHoldEm Class (Sub-Class of Game) | |
| Inherited Parts | *Everything except CompareTo method* |
| Added Properties | Double Bet |
| Added Methods | Raise (double)  Fold (void) |
| Fortune (Sub-Class of Game, uses FileStream or StreamReader) | |
| Inherited Parts | *Everything except CompareTo method* |
| Added Properties | DateTime today |
| Added Methods | GetPast (string)  GetPresent (string)  GetFuture (string)  CalcFortune (int)  ToString (override) |

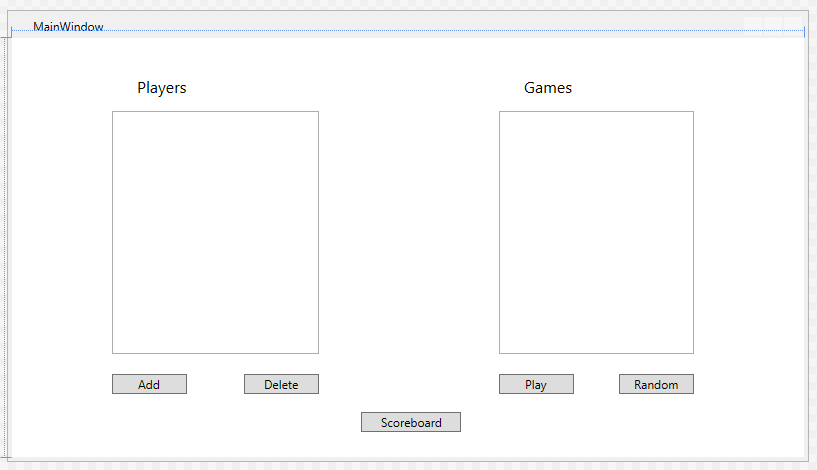
## Possible Button Logic



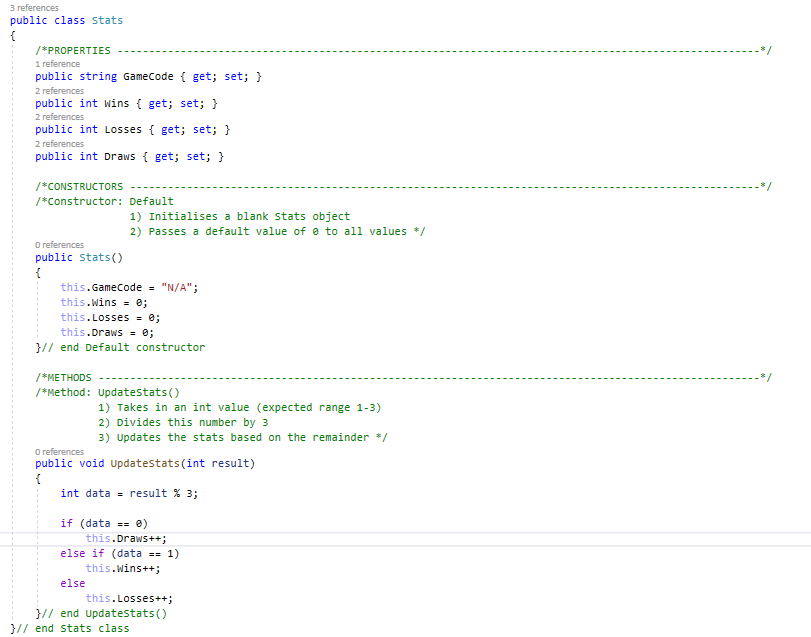


# First Idea Iteration

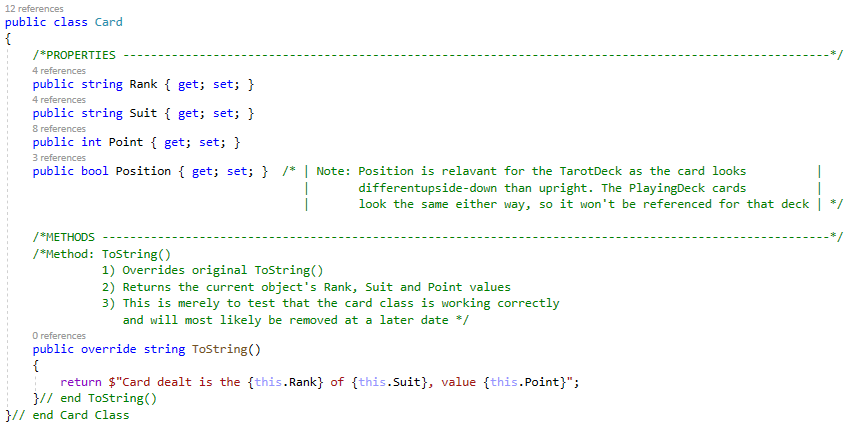
### Main Window



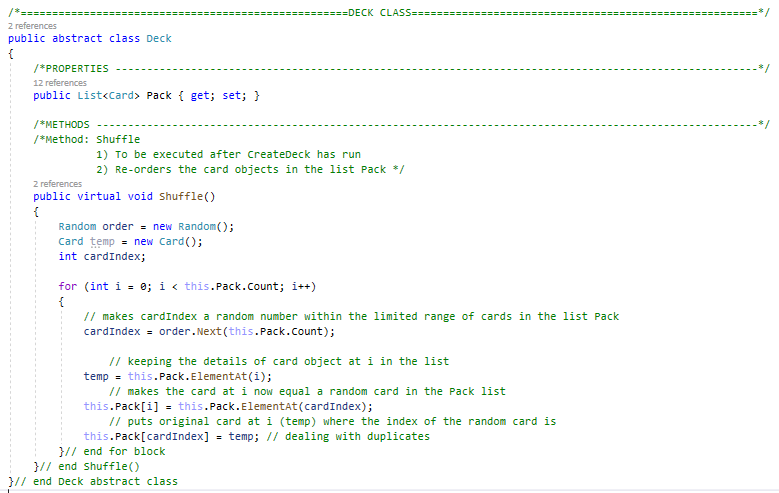
### Stats Class Code

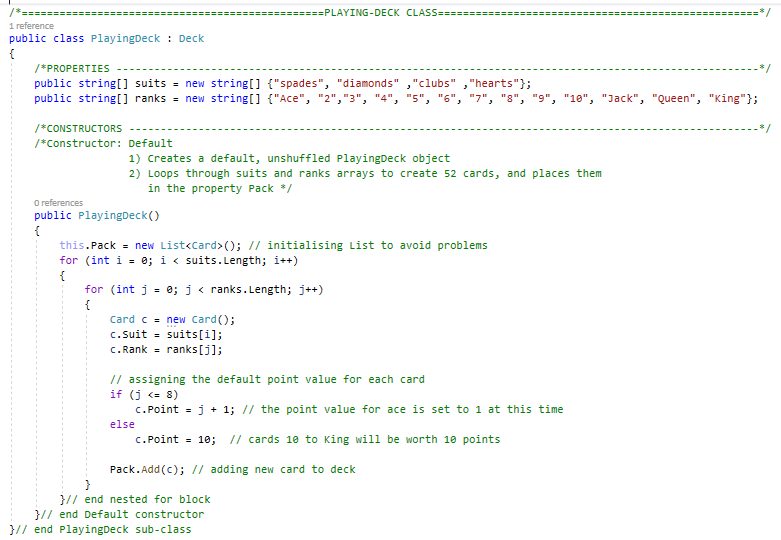


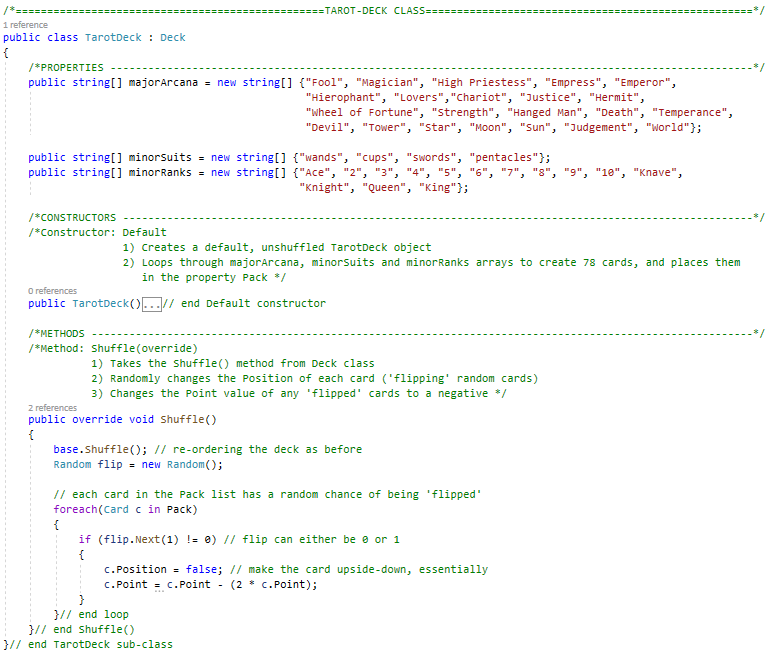
### Card Class Code



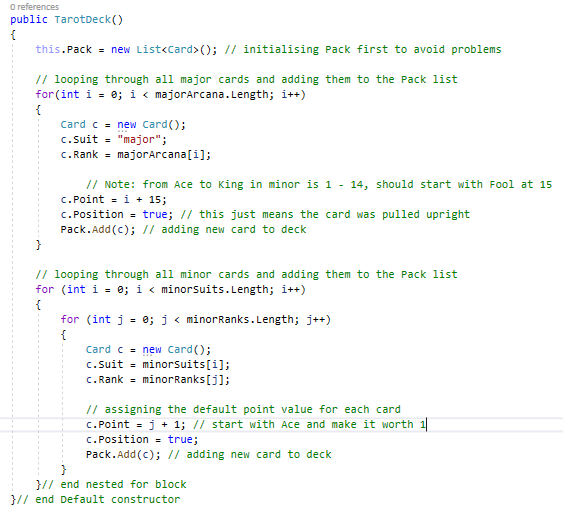
### Deck Class Code







* *Closer look at TarotDeck() Constructor*



### Game Class Code