## horizontal line



**Whist22**

Multiplayer Online Game with AI Bot Offline

Name: Zain-Ul-Abideen

**Overview**

Whist22 is a card game similar at some point to the game “Oh Hell” but simpler than this because there is no suit. It can be played with the tarot deck cards from which only 21 trumps and a fool card. The rules for the game will be mentioned below as there are many rules. I build this game for multiplayer purposes in which you can play this game with your friends worldwide using the internet, and if there is no one to play with, the bots will take place offline and play with you like a real player. It will be happening through an automatic and efficient UI code. The total score for everyone at the start of the game will be 14 and everyone will bid for themselves like how many tricks he/she will win if any of them is unable to win the exact amount of tricks so he will lose the game.



# 

# **Game Rules (How to Play Game)**

Summary of the rules: Ref boardgamegeek.com

1. If played in partnerships, partners sit across the table from each other.
2. Players begin the game with 14 points. The game ends when one player reaches 0. The player scoring the most points wins.
3. The dealer deals the cards so that the lowest number of cards remains undealt: for example with four players, five cards are dealt to each player and two cards are put aside.
4. The player to his left announces how many tricks he thinks he will win, then the player on his left, etc... This is the contract. The dealer is the last to announce, but the sum of announced tricks must not equal the number of cards dealt to each player.
5. Let us follow the same example, where 5 cards were dealt to each player: player A announces 2 tricks, and players B and C each announce 1 trick. Then the dealer cannot announce 1 trick because the total would be 5 tricks and each player could fulfill his contract. The player to the left of the dealer starts. The winner of a trick is the one who will start the next trick. The owner of the fool must assign its value (0 or 22) when he plays it.
6. At the end of the round, each player counts the number of tricks he has won and compares it to the number of tricks he has announced. No point is awarded for fulfilling the contract, but 1 point is deducted for each missed trick, either over or under the number of tricks the player initially announced.
7. The next round takes place in the same way, but with one less card per player.
8. After several rounds where one less card is dealt in each round, there is a round where each player only receives one card. Each player then places the card on its forehead without looking at it, so that all players except him can see it. Each player then announces if he will win the trick or not. As was the case in previous rounds, the sum of announced tricks cannot equal the number of cards dealt to each player (i.e. one).

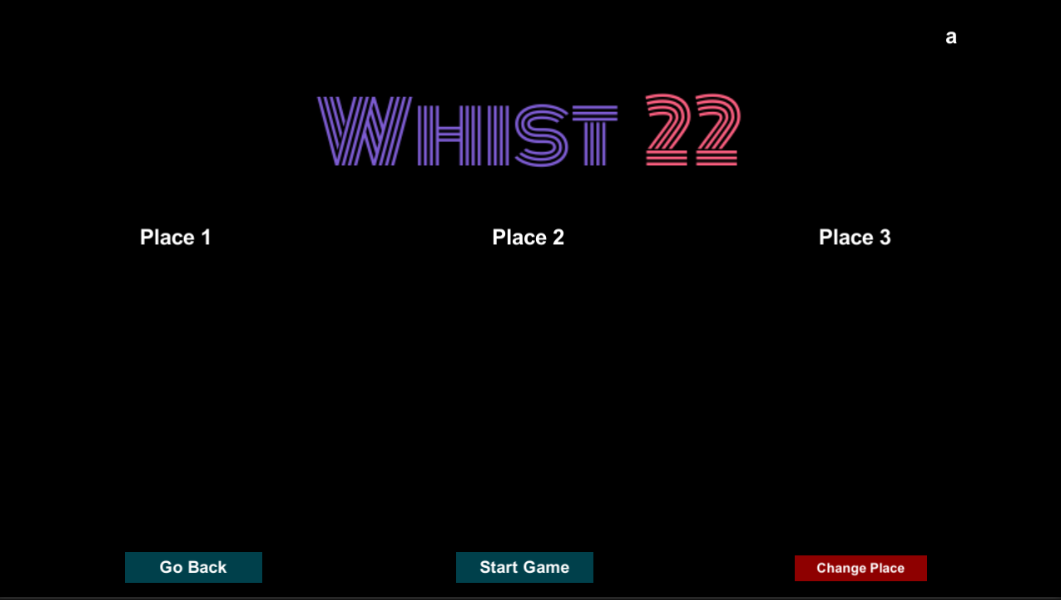
# **Gameplay**

**Play Offline or Online Screen:**

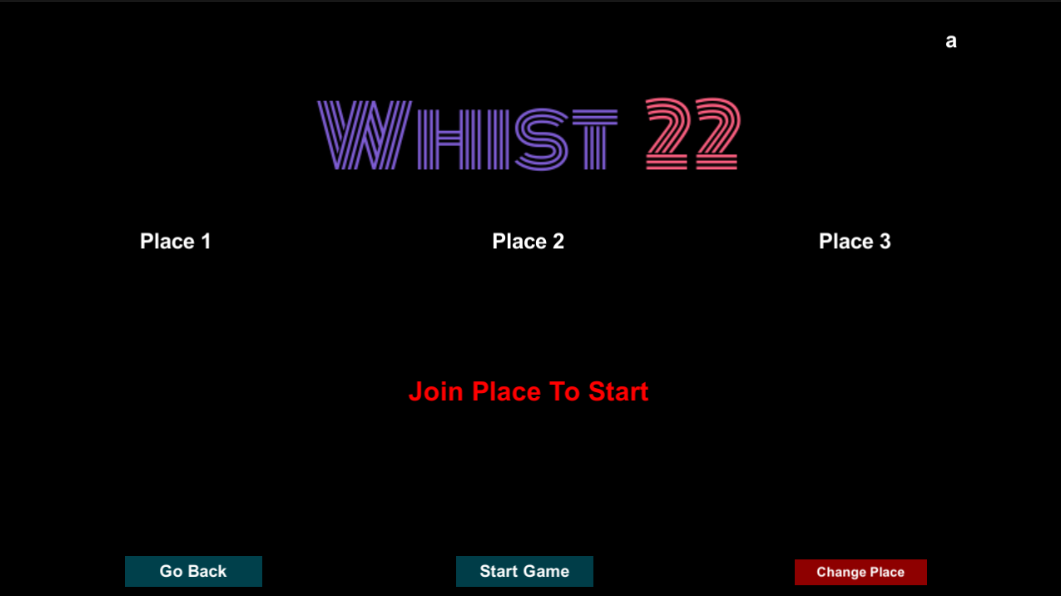


On this screen, you have to select whether you want to play Online or Offline. For the online purpose, you must have an internet connection and 3 friends to join you. Offline the Bots will take place with you to play.

**Offline Selected:**

****

On This screen, you can take place either you want middle, first, or second, at the moment you click on your desired place the **Bots** will automatically take place in the other two places. You cannot start the game without joining any place the game will throw an error message “Join Place To Start” as bellow.

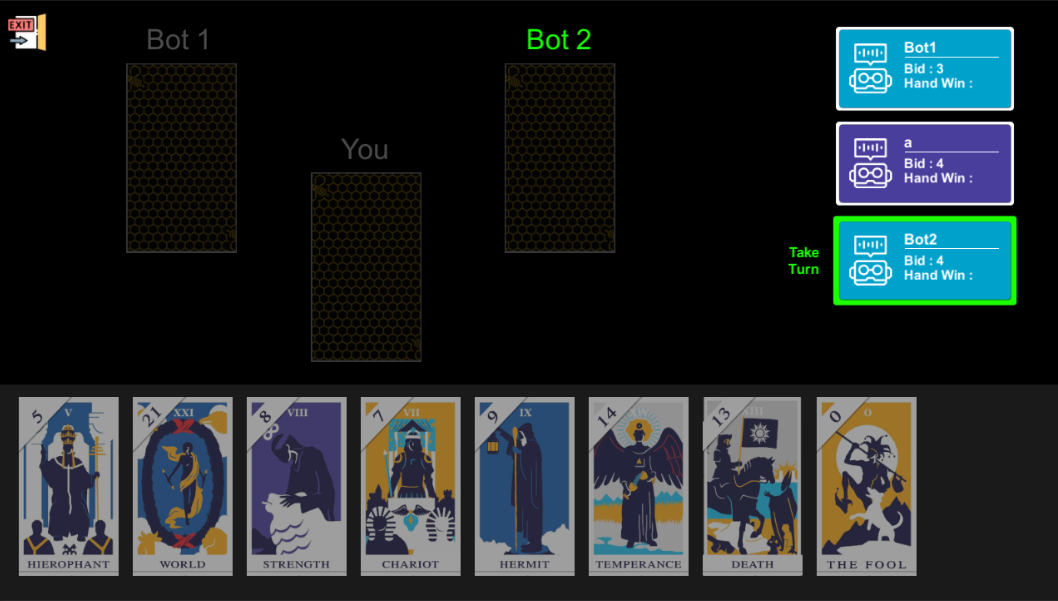


For example: I click on Place 2 so below you can see **Bots taken the other both places.**

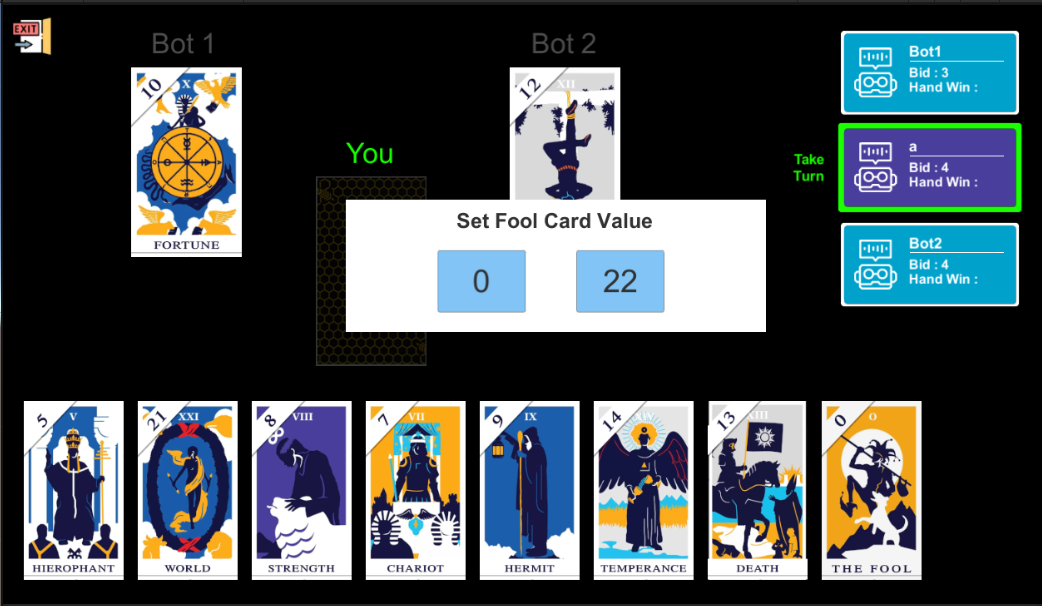


Now You can start the game.

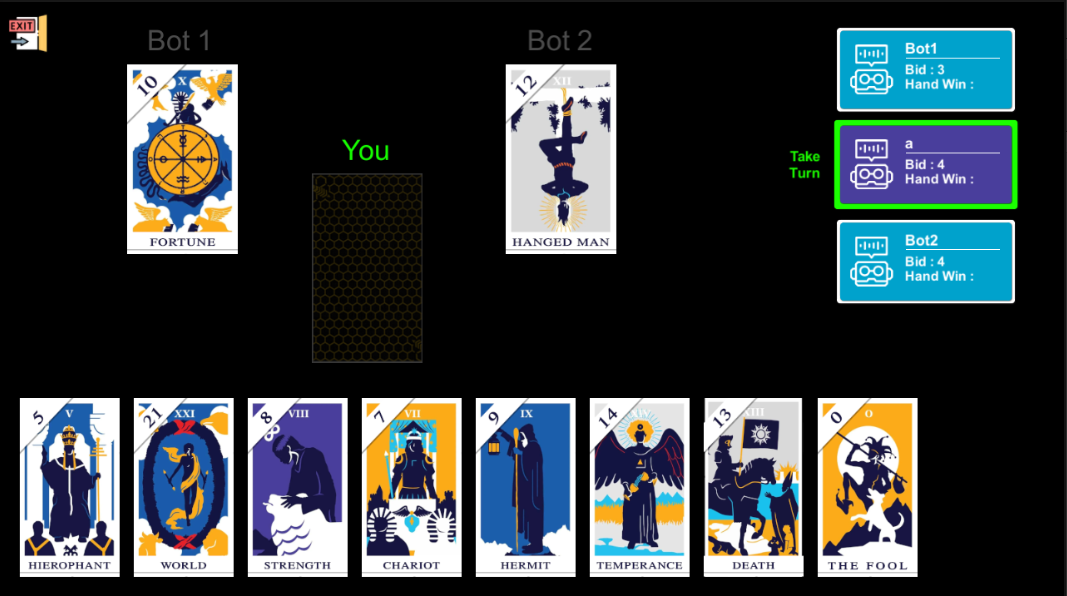
**Gameplay Screen:**

****

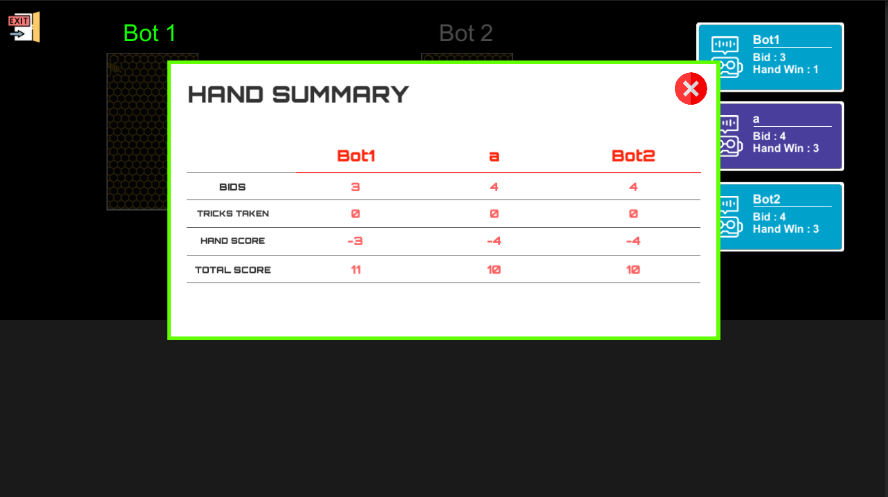
1. **PlayerStatCard:**  As every player have some stats to play the game, the Player stat card on the **right** side of the screen will show every information about the player. **Bid Value, HandWins, and when the player turn is so the outline will be green and take turn text highlighted**.
2. **LocalPlayerCard:**  On the Bottom, The local player card will be shown, and other players' cards will be hidden.
3. **Fool Card Window:** will appear when the fool card is for the local player and he tries to through it on stage.



1. **Turn:** Bots will take turns by themselves automatically. And the player have to select the card. The player cards are on bottom of the screen and when its bot turns so the player will unable to click on card.

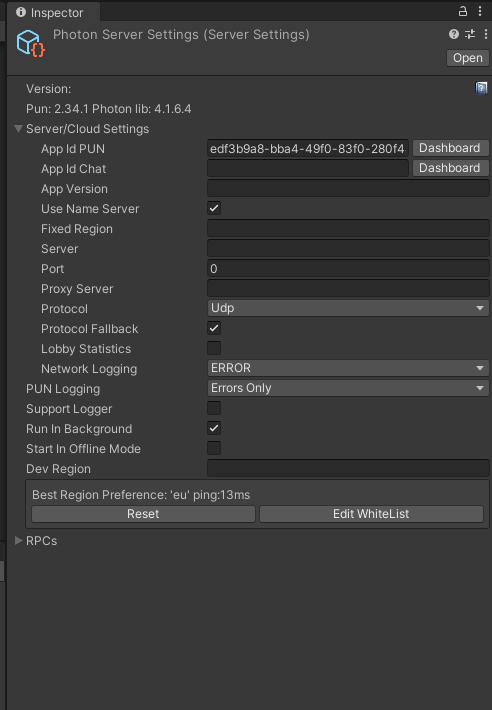


Following the game rules, the game will continue like this until any of the players have 0 cards left then the summary screen will appear as below.



No Player winds as none of them win the hand as exact their bid value.

# **Photon Server Implementation Functionality.**

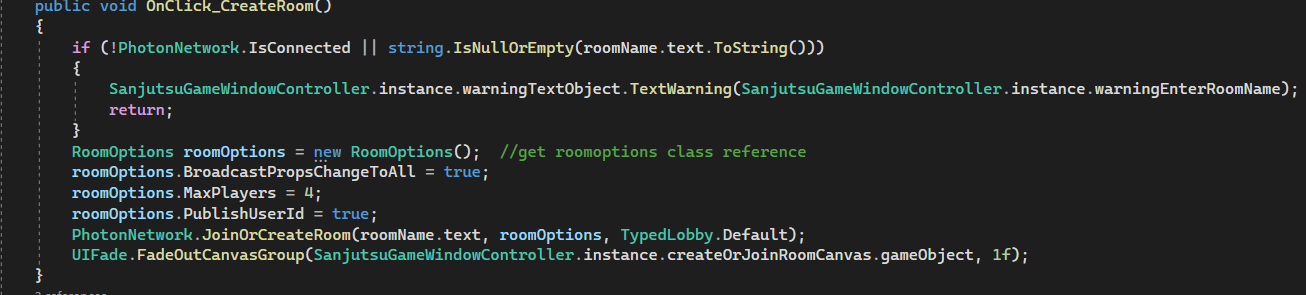
**Note: due to the huge codebase I am unable to put every important snippet here as the length of the report will be long.**

Photon Server Integration Setup

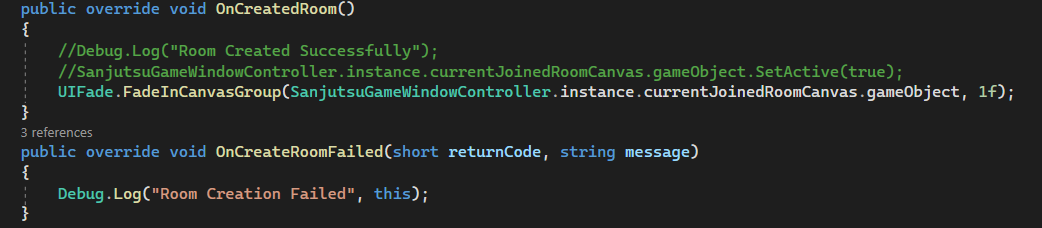
* App Id (Generated from photon account)
* Use Name Server

# 

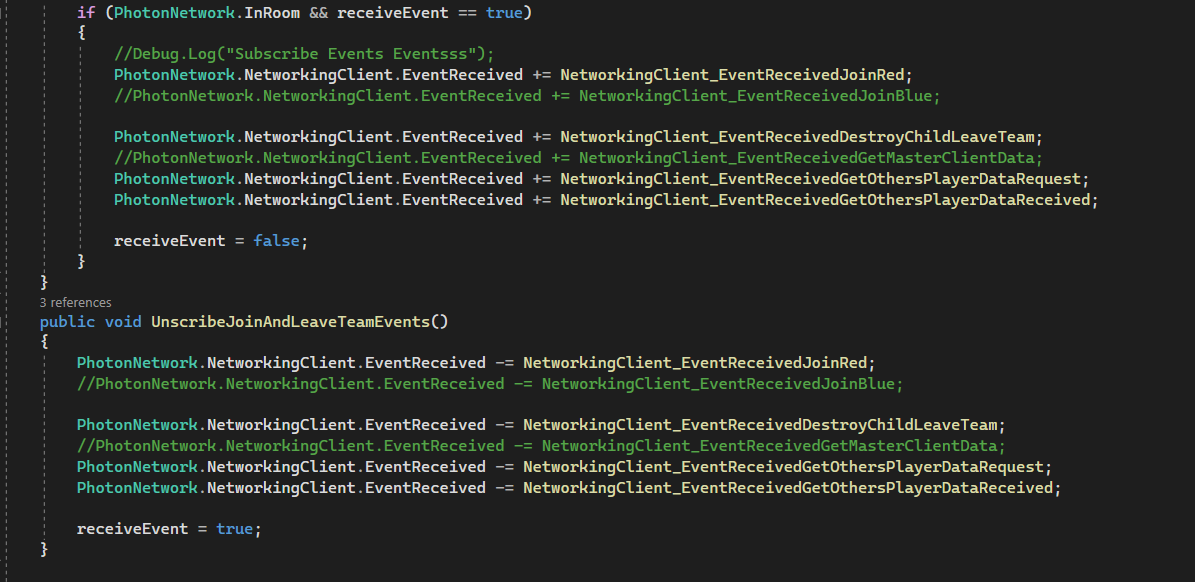
**Create Room** functionality will take the room name and create a room with that name so any player can find it after joining the lobby (the available rooms to join will be listed in dropdown on UI)



The photon server also notifies us whether the **room is created or failed**. So that we can take action on the basis of the current failed situation.



This is how **subscribing and unsubscribing** of the event is handled. So which player who subscribed to the event is the only one who receives that particular event? The receiver must have to event code in order to get the data from the received event.



# **Functionality**

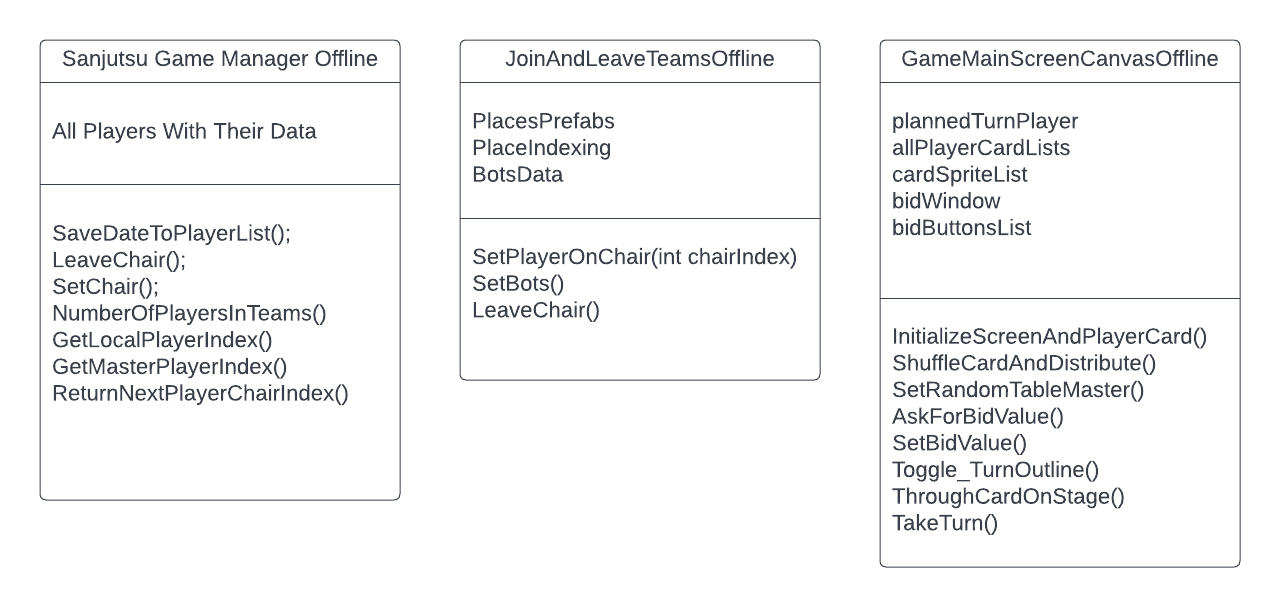
By following the rules stated above whist22 can be played with 3, 4, or 5 players. **But because of the testing concerns, I was unable to arrange for 4 or 5 players to test every module in the game so I just developed it for 3 players. I Implemented the backend for Online but Unfortunately due to the hard rules and huge codebase I got some logical errors while updating the database That is why I stopped working on the multiplayer section as it is available on many websites already so I just focused on the Offline version of the game**. As there are many rules in this game we have to write efficient code for multiplayer purposes and every player must be updated at the same time as others.

**Code Structure For Offline:**

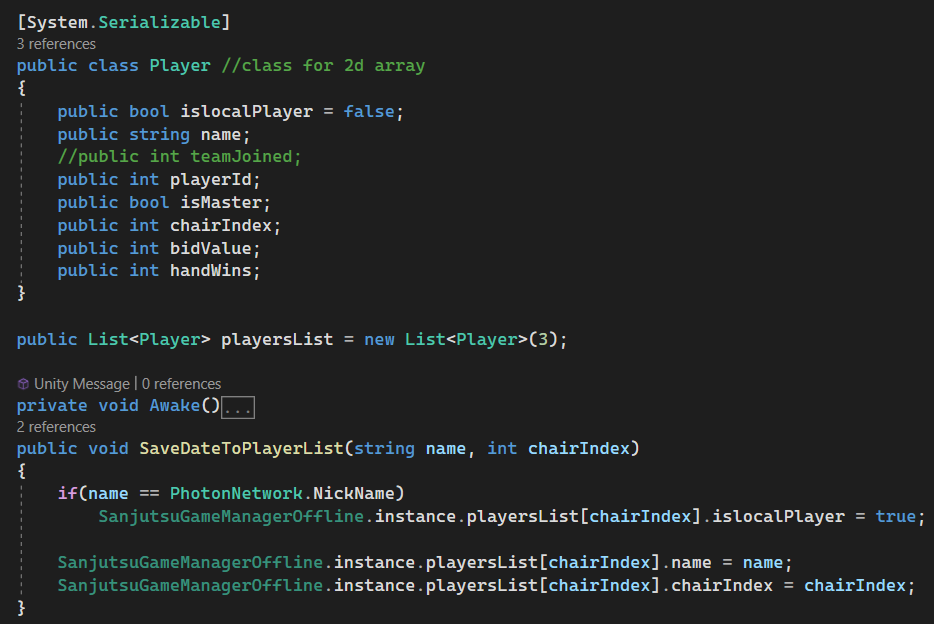
Here are the main Scripts that are handling the game.

Note: Sanjutsu is a name for this game in the ancient Japanese era.

Here are the main scripts.



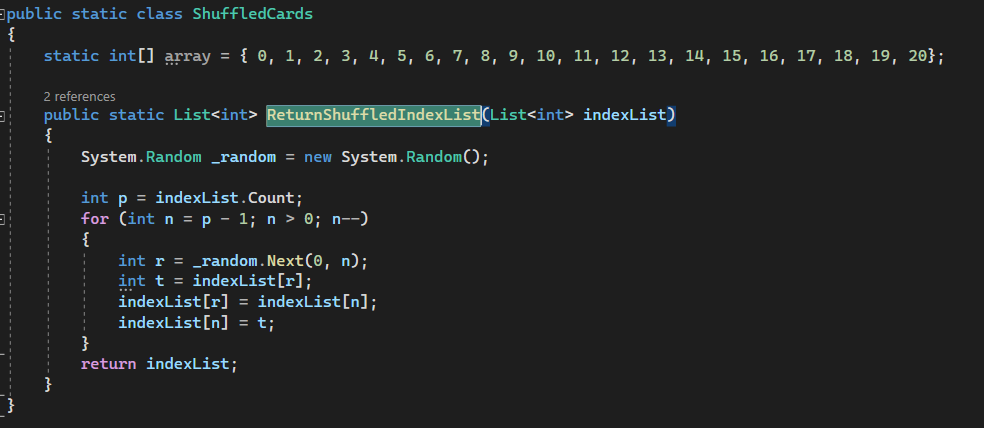
The main component of the games will be the players, the Game Manager who will manage the data of players and provide core function as above. This is how Game Manager saves data in a 2d list of players.



Playing offline the **bots will take place in empty places right after you take place**.

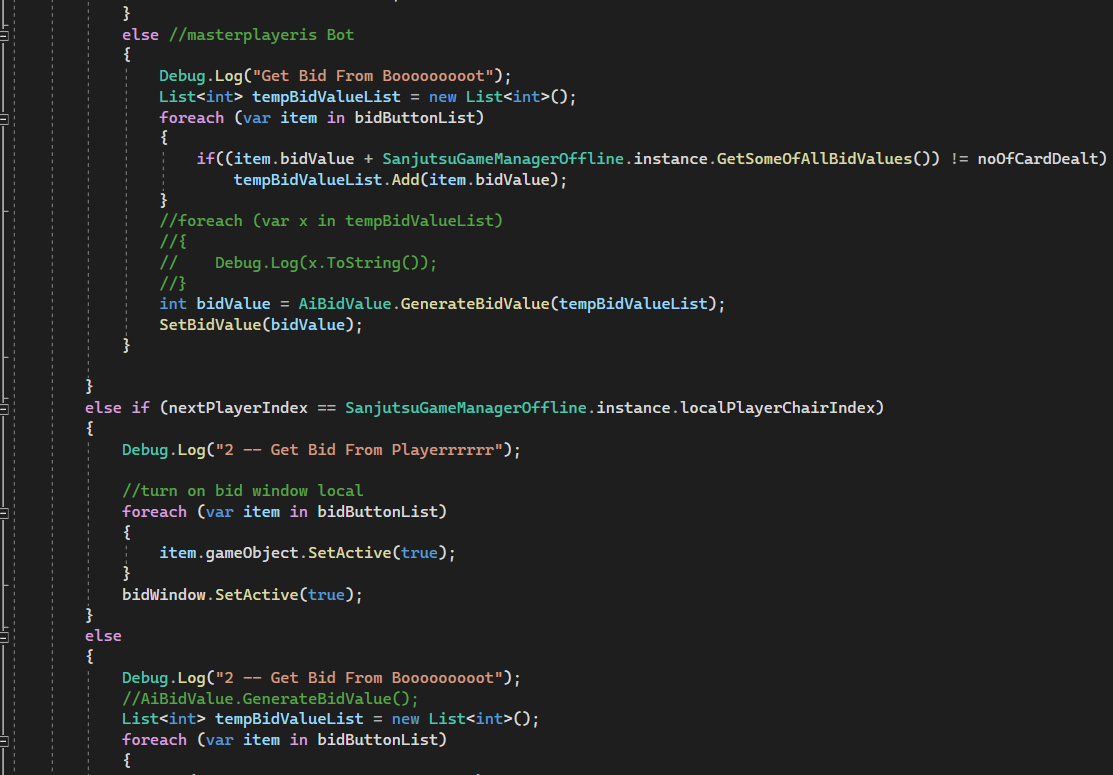


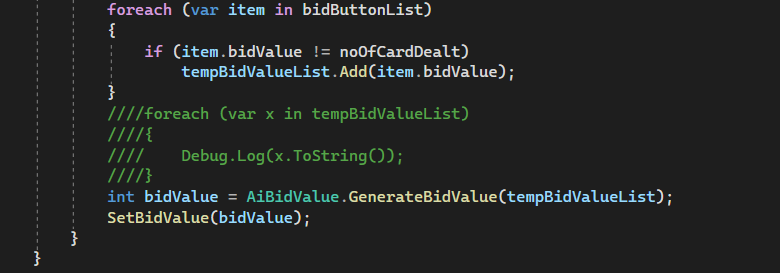
This is how I am **shuffling the card** through a utility class as below.



This is how code is **asking for bid Value** automatically from every player in the game (due to the many number of code lines I have to break the snippet in two)







**Note: due to the huge codebase I am unable to put every important snippet here as the length of the report will be long.**

# **Referencing Assets**

1. Multiplayer server Photon Pun 2 (https://assetstore.unity.com/packages/tools/network/pun-2-free-119922)
2. The cards used for this project will be from (<https://www.alittlesparkofjoy.com/tarot-cards-list/>)
3. Some Particle effects and animation by the animator will be used.
4. For sounds ([www.miskit.co](http://www.miskit.co) & www.pixabay.com)
5. For the background, Buttons, Input Fields, and table I will create my own design by myself.

**Note:**

* For UI and other graphical things, I created my own design using Adobe Photoshop and Xd.
* Due to the huge codebase, I am unable to demonstrate every important aspect of coding and features.
* The splash screen is added.