RMIT UNIVERSITY VIETNAM

COSC2659 - IOS Development

Assignment 2

Tutor: Dr. Tom Huynh

Student Name: Huynh Dac Tan Dat   Student Number: s3777091

- Introduce the app.

Here is video Link

https://drive.google.com/file/d/13CVCku0Xondu-vmmGexA8i_L_RsUPqBk/view?usp=sharing

App review:

Icon App:

I started with Viet Nam Big Two app, it's very difficult about AI algorithm but it took me almost two weeks to complete it but there are still some errors like threeOfKind cards can be blocked by straight or i can 't replay the game to play and it's very difficult to make the menu for the card game so I decided to change to playing poker with AI in poker, there are still some bugs but promise there's someone smarter than a flexible strategy than.

App Review:

Main Menu:

Contain user avatar and user money (instance user score i change to user money that currency that user can play with Ai bot). Next is paperView: paperView can automate scroll to next place depending on time.

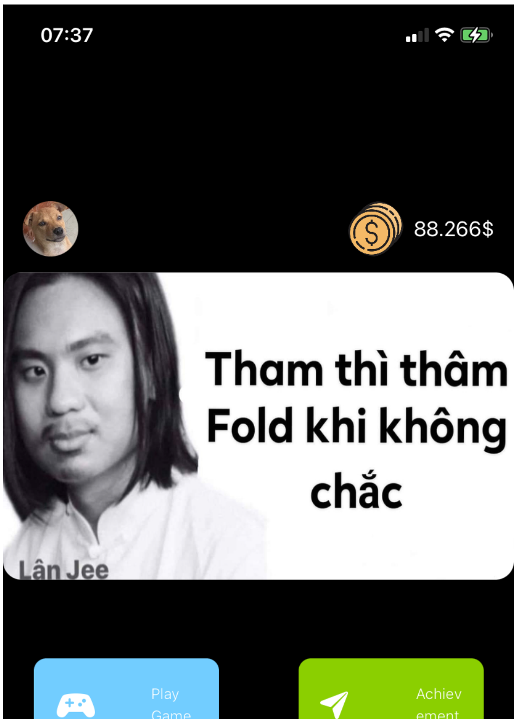
Menu contain 4 screen:

Play Game View : Click go to Poker Screen is the screen where you can play Poker with Ai Bot

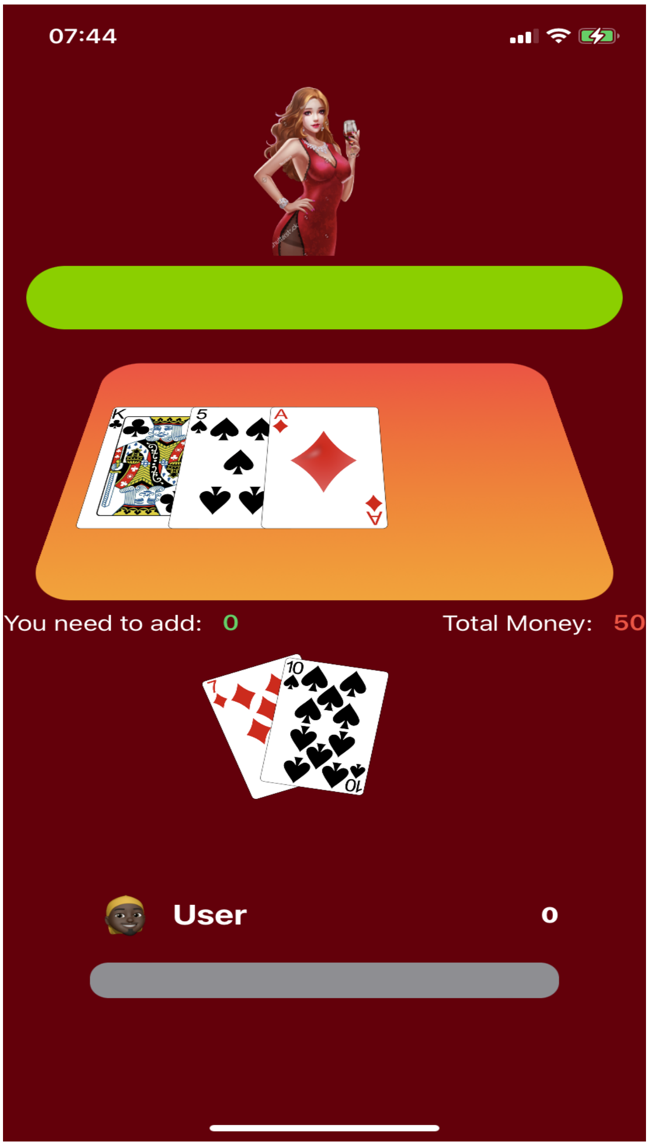
Add Money View : Click you need to scratch a card to earn 50000 dola to your user account. (Scratch as many as you want due to poker being an easy game to lose all of your money so I added this function to keep users playing).

Achievement View: While you play or win game with different type of poker in hand for example: Flush, fourOfKind, royalFlush you will earn achievement that contain Avatar so you can click and change avatar for user View.

Rules View : View contains only link that help us understand about poker and this report will illustrate more detail about rule.



Play Game View:



Play View:

- □ Dealer
- □ Messenger Box
- □ Desk Table
- □ User Keep track money
- □ User Cards
- □ User Slider
- □ 3 button Fold, Call with money

,Call Any

In this Screen, these are a lot of function and condition to control so most of time i will go detail with How i code AI bot:

AI Bot:

Start as defaults all of checking flag = false

```
var pairExist = false, threeExist = false,  fourExist = false, fullHouseExist = false, straightExist = false, flushExist = false
```

```
    var rankCount = [Rank : Int]()
    var suitCount = [Suit : Int]()
```

Then i call RankCount and SuitCount to Count Value of Bot Hand:

```
var playerCardsByRank : [Card] = []

    if DealerCards.count == 3 {
        //Start add Money in first Start And Check for first Cards
        //Only Check Cards In hand
        //Start Money
        self.TotalAmount = TotalAmount + 50
        playerCardsByRank = player.cards
    } else {
        playerCardsByRank = GetCards(HandStack: player.cards, DeskCards: DealerCards)
    }
```

As Start game bot will add 50 dola to play and it will check these cards in hand first then next turn i will check card in hand with the desk table.

```
    for card in playerCardsByRank {
        if rankCount[card.rank] != nil {
            rankCount[card.rank]! += 1
        } else {
            rankCount[card.rank] = 1
        }
        if suitCount[card.suit] != nil {
            suitCount[card.suit]! += 1
        } else {
            suitCount[card.suit] = 1
        }
    }

    var cardsRankCount1 = 1
    var cardsRankCount2 = 1

    for rank in Rank.allCases {
        var thisRankCount = 0

        if rankCount[rank] != nil {
            thisRankCount = rankCount[rank]!
        } else {
            continue
        }

        // Check if there are ranks > 1. This is to detect pair, three, four, fullhouse
        if thisRankCount > cardsRankCount1 {
            if cardsRankCount1 != 1 {
                cardsRankCount2 = cardsRankCount1
            }
            cardsRankCount1 = thisRankCount
        } else if thisRankCount > cardsRankCount2 {
            cardsRankCount2 = thisRankCount
        }

        pairExist = cardsRankCount1 > 1
        threeExist = cardsRankCount1 > 2
        fourExist = cardsRankCount1 > 3
        fullHouseExist = cardsRankCount1 > 2 && cardsRankCount2 > 1

        if straightExist {
            continue
        } else {
            straightExist = true // start off with true then check below
        }

        for i in 0 ... 4 {
            var rankRawValue = 1

            if rank <= Rank.Ten {
                rankRawValue = rank.rawValue + i
            } else if rank >= Rank.Ace {
                rankRawValue = (rank.rawValue + i) % 13
                if rankRawValue == 0 {
                    rankRawValue = 13
                }
            }

            if rankCount[Rank(rawValue: rankRawValue)!] != nil { // if all 5 consecutive rank exist
                straightExist = straightExist && rankCount[Rank(rawValue: rankRawValue)!]! > 0
            } else {
                straightExist = false // if one of consecutive rank does not exist
            }
        }
    }
```

Loop for all rank in table to easy understand this one will simple it with add 5 cards more in Stack like this:

First loop i will return all single cards in hand like this then it start with counting sort algorithm:

Ace, Two, Four, Ace, King, Ten, Two

Then:

If Ace count 1 if Two count 1 if Four count 1 if Ace count 2 if King count 1 if Ten count 1 if Two count 2 while 2 cards count that meant it bring flag pairExist = True or if count 3 it will bring flag ThreeOfKind = True

With Straight and another Hand Type of Stack need to by Sort:

I Start algorithm sort in Extension of Stack:

```
extension Stack where Element == Card {
    func sortByRank() -> Self {
        var sortedHand = Stack()
        var remainingCards = self

        for _ in 1 ... remainingCards.count {
            var highestCardIndex = 0
            for (i, _) in remainingCards.enumerated() {
                if i + 1 < remainingCards.count {
                    if remainingCards[i + 1].rank >
                        remainingCards[highestCardIndex].rank ||
                        (remainingCards[i + 1].rank == remainingCards[highestCardIndex].rank &&
                         remainingCards[i + 1].suit > remainingCards[highestCardIndex].suit) {
                        highestCardIndex = i + 1
                    }
                }
            }

            let highestCard = remainingCards.remove(at: highestCardIndex)
            sortedHand.append(highestCard)
        }

        return sortedHand
    }
}
```

So i will sort all Cards in Stack for Example:

Seven, Four, Six, Five, Three => Sort => Seven, Six, Five, Four, Three

Then if
Seven - Six = 1 that mean true
Six - Five = 1 that mean true
Five - Four = 1 that mean true
Four - Three = 1 that mean true

=> StraightExist = true

Then while bring What type exist in hand of Cpu

Cpu will start actions such as Block user, bluff user, add money , increase money or Fold card even check Card.

All actions will start by Strategy:

```
// Need Animation
func HumandBot(of playerCpu: PlayerPorker){
    Messager = ""
    if handScore(GetCards(HandStack: playerCpu.cards, DeskCards: DealerCards)) < handScore(GetCards(HandStack: UserCards, DeskCards: DealerCards)) {
        Messager = "\(playerCpu.playerName) Add \((200 + MoneyPluss))$"
        self.MoneyUserNeedAdd += (200 + MoneyPluss)
        self.TotalAmount += (200 + MoneyPluss)
    } else if MoneyPluss == 0 && handScore(GetCards(HandStack: playerCpu.cards, DeskCards: DealerCards)) > handScore(GetCards(HandStack: UserCards, DeskCards: DealerCards)) {
        Messager = "\(playerCpu.playerName) Add \((200 + MoneyPluss))$"
        self.MoneyUserNeedAdd += (200 + MoneyPluss)
        self.TotalAmount += (200 + MoneyPluss)
    } else if MoneyPluss > 0 && handScore(GetCards(HandStack: playerCpu.cards, DeskCards: DealerCards)) > handScore(GetCards(HandStack: UserCards, DeskCards: DealerCards)) {
        Messager = "\(playerCpu.playerName) Add \((200 + MoneyPluss))$"
        self.MoneyUserNeedAdd += (200 + MoneyPluss)
        self.TotalAmount += (200 + MoneyPluss)
    } else if let playerIndex = players.firstIndex(where: { $0.id == playerCpu.id }) {
        if handScore(GetCards(HandStack: playerCpu.cards, DeskCards: DealerCards)) < handScore(GetCards(HandStack: UserCards, DeskCards: DealerCards)) {
            Messager = "\(playerCpu.playerName) Fold cards"
            players[playerIndex].cards = []
        }
    } else {
        Messager = "\(playerCpu.playerName) Check cards"
    }
}
```

Cpu will check your cards and their cards and check money you add in to give back a decision. Currently, I only use a handScore that only calculates if their cards are that larger than yours or not. So a handScore is just like a prediction score that can't get the true value of cards (True value of card means know your handType of Not).

Active Next Player:

While current user dealing their card with money or while start at begin of stage, Bot need automative active to play with user

Here is how i do:

First:

```
struct PlayerPorker: Identifiable, Equatable {
    var cards = Stack()
    var playerName = ""
    var playerImage = ""
    var checkingScore = 0
    var UserMoney = 0
    var playerIsMe = false
    var activePlayer = false
//    var playStyple : Int = 1
    var id = UUID().uuidString

    var offset: CGFloat = 0

    static func == (lhs: PlayerPorker, rhs: PlayerPorker) -> Bool {
        return lhs.id == rhs.id
    }
}
```

Create activePlayer : Bool in PlayerPorker struct to while it goes to that user turn i will activate that player to True.

Second:

I create Function findFistUserPlay() by random
in this case i will random 4 player

```
    func findStartingPlayer() -> PlayerPorker {
        let randomInt = Int.random(in: 0..<4)
        return players[randomInt]
    }
```

While i know first firstPlayer

First TotalRound is 0 While start game TotalRound = 1 it will check and find first people starting. TotalRound will increase through one second time

```
 if Games.TotalRound == 1 {
                    // First Player with Random
         nextPlayer = Games.findStartingPlayer()
 } else {
                nextPlayer = Games.getNextPlayerFromCurrent()
 }
```

```
    let timer = Timer.publish(every: 1, on: .main, in: .common).autoconnect()
```

I will use timer and TotalRound to active nextPlayer

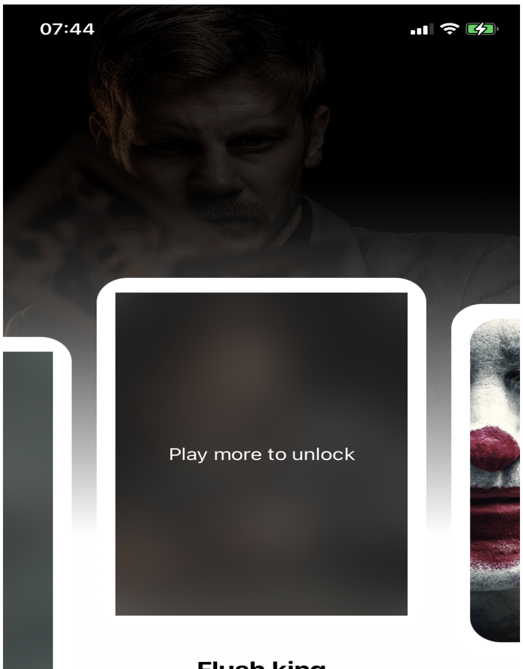
```
    .onReceive(timer) { time in
        var nextPlayer = PlayerPorker()
        counter += 1
        if counter >= 2 {
            if Games.TotalRound == 1 {
                // First Player with Random
                nextPlayer = Games.findStartingPlayer()
            } else {
                nextPlayer = Games.getNextPlayerFromCurrent()
            }

            Games.activatePlayer(nextPlayer)
            if nextPlayer.playerIsMe && !Games.players[3].cards.isEmpty {
                counter = -100
                NowIsYourTurn = true
            } else {
                counter = 0
                NowIsYourTurn = false
                if Games.players[3].cards.isEmpty {
                    WinGamePoker()
                }
            }
        }
    }
```

While change playerActive = true

```
.onChange(of: Games.activePlayer) { players in
            if !players.playerIsMe {
                Games.CheckingCpuHand(of: players)
            }
        }
```

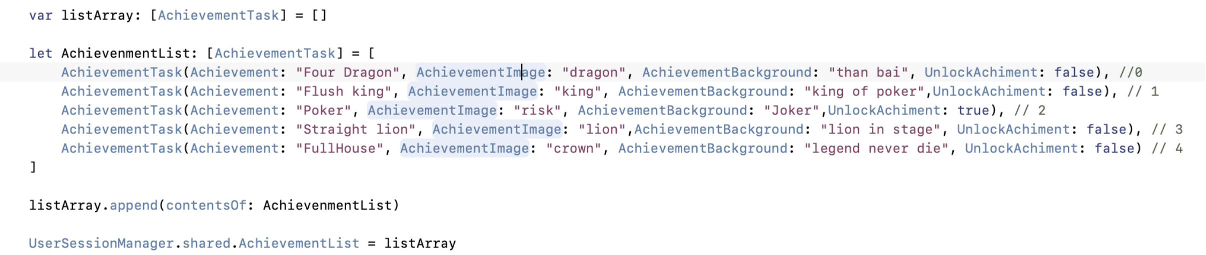
It will check Cpu hand and give decision money.

In Achievement View:



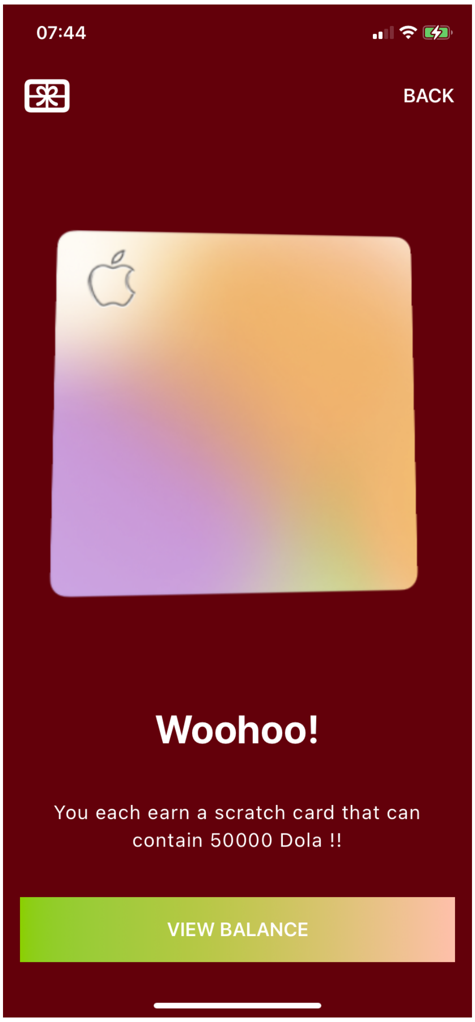
□ List userDefaults of userAchievement

□ Config user Avatar



```
var listArray: [AchievementTask] = []

let AchievenmentList: [AchievementTask] = [
    AchievementTask(Achievement: "Four Dragon", AchievementImage: "dragon", AchievementBackground: "than bai", UnlockAchiment: false), //0
    AchievementTask(Achievement: "Flush king", AchievementImage: "king", AchievementBackground: "king of poker",UnlockAchiment: false), // 1
    AchievementTask(Achievement: "Poker", AchievementImage: "risk", AchievementBackground: "Joker",UnlockAchiment: true), // 2
    AchievementTask(Achievement: "Straight lion", AchievementImage: "lion",AchievementBackground: "lion in stage", UnlockAchiment: false), // 3
    AchievementTask(Achievement: "FullHouse", AchievementImage: "crown", AchievementBackground: "legend never die", UnlockAchiment: false) // 4
]

listArray.append(contentsOf: AchievenmentList)

UserSessionManager.shared.AchievementList = listArray
```

In Add Money View:

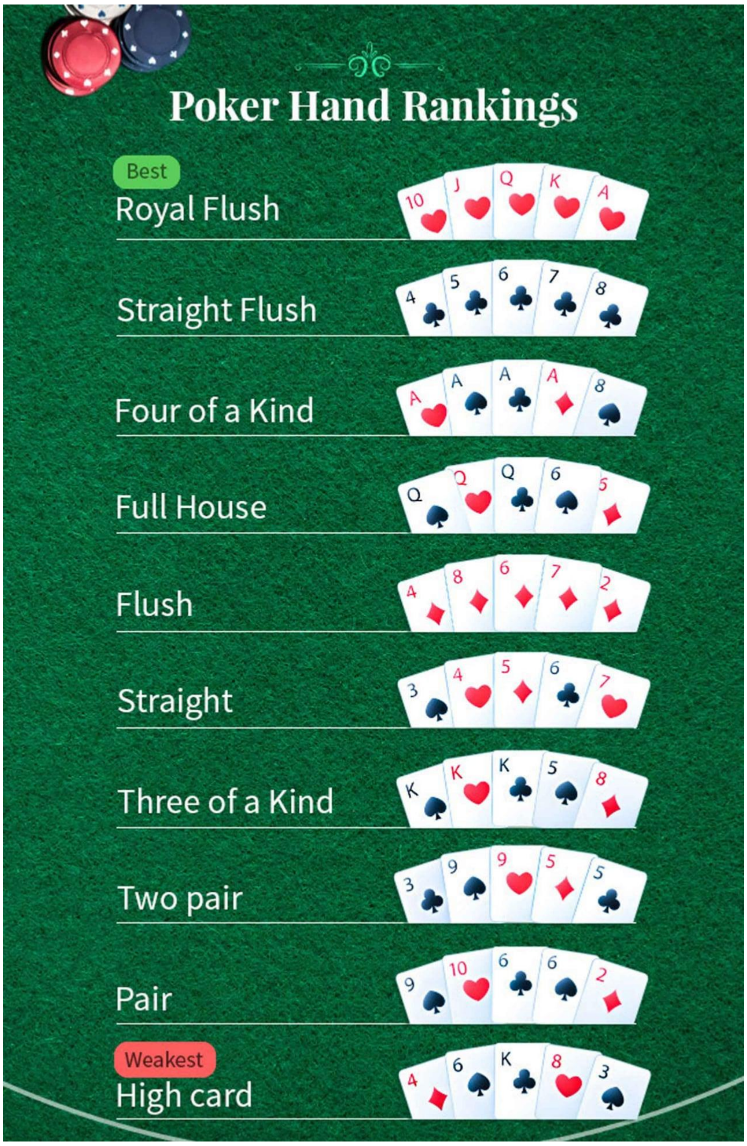


Woohoo!

You each earn a scratch card that can contain 50000 Dola !!

- □ Scratch Card to earn 50000 dola
- □ View Balance with alter about your account

How to Play:



First Round, user will add 50 dola for play (program will do it for you) then 3 cards in table, random player first if is you :

Three option:

one Fold Card mean you give up because your cards so low or the money you need adds too high.

Call with money mean you not satisfy with money while you have high cards you want to add more money. Bot Will add same money with you for example if you add 50 dola so => 150 dola if all 3 bot add money without folding their cards.

Call any mean you add money the same as same with money you need to add.
For Rank of Cards

Three, Four, Five, Six, Seven, Eight, Nine, Ten, Jack, Queen, King, Ace, Two

Mean Three is lowest and Two is highest one

For Suit of Cards

Spade, Club ,Diamond, Heart

Whoever has a high hand type in hand will Win the game and earn all Total Money.

Github code Here:

https://github.com/s3777091/TienLen.git

In conclusion, this project still has several errors such as can’t replay games and several errors while checking who wins.