End-User Modeling Final Project

Excel Blackjack

April 13, 2009

Professor Rafael Nico Fermin

Group 9:

Manny Elawar
Joel Read
Kendy Lau
Jennifer Lam
Natalie Deschamps

Overview

The program is a simulation of the standard North American version of blackjack using one deck of cards and including insurance, doubling down, and dealer stands on soft 17 (ie ace-six). The program is driven primarily through user commands (ie betting amounts, hit, stand, double down, etc.), although there is significant back end functionality that creates the game environment using object oriented programming. Since explaining the entire operation in detail would be extensive and often trivial, we have highlighted the more interesting components of the program: deck creation and shuffling, dealing with Aces, strategy recommendations, and our probability of victory calculator.

The Deck

When the game initially opens, a deck of cards is created through a loop of four iterations (one for each suit), run thirteen times (one for each card value). The elements are stored in a matrix. Each card is assigned a suit, face value, and actual value (ie a Jack's face value would be "J" but it's actual value would be 10). Aces are initially assigned a value of eleven. The suit and face value are used to call the relevant card images that the user sees in the interface. Additionally, all cards are assigned a Boolean variable IsUp = True . That is, the card will displayed face up, unless it is the dealers first card, which is dealt facedown by changing the variable to False during the dealing operation

Once the 52 card deck is created, it is then shuffled (this operation is cued every time the deals selects the "deal" function). A random card is selected from the deck/array. The preceding cards are then shifted down the array and the selected card is placed in the first position of the array. This is repeated 300 times, creating a sufficiently randomized deck of cards that can be dealt from the top exactly like a real deck.

Dealing with Aces

Aces represent a unique challenge in coding blackjack, since the card can have two possible actual values (11 or 1). To solve this problem, we have created a logic test within the sub routine that determines a hand's value. Logically, the only time a player would prefer an Ace to be a 1 instead of 11 would be when the player's hand value exceeds 21. This insight allows the following test to work.

The hand is checked card by card to identify if there are any aces (if not, this operation is not carried out). If an Ace is found, and the function checks to see if the value of the hand up to the ace is greater than 21 (ie 5-Ace-5, the value of the hand up to the ace would be 5). If so, the actual value of the Ace is switched to 1 and the hand's value is recalculated. Otherwise, the value of the ace is unchanged and added to the running hand value (in the previous example, the running hand value would become 16).

The risk of this operation is that an Ace might not initially cause a hand to exceed 21. To account for this, the function activates a Boolean variable isAce if at least one ace is found in the initial review of the hand. If the variable has a value of True at the end of the review, a second test is launched (contingent on the hand's value still exceeding 21 after the first review). The function checks each Ace and determines if its value has already been adjusted to 1. If not, the second test adjusts the Ace's value and recalculates the player's hand.

This set of tests is run after either the player or the dealers adds another card to their hand (either by hit or double down), and before the hand is assessed for outcome (win, push, bust).

Suggested Strategy

We have added two additional buttons for the user immediately after the cards are initially dealt. One of these functions is a suggested strategy function, which calculates the player's hand value, takes the dealer's face card value and then uses the information to look up the recommended action from a standard blackjack strategy card. The strategy card itself has been divided into two tables in order to deal with aces in a simple manner. When the user clicks on the suggested strategy button, the function returns a prompt with the statistically best option to take give the player's cards and the dealer's face up card.

Probability of Success Calculator

Perhaps the most interesting component of this game is the option for the player to determine the probability of winning a round, based on the player's initial cards and the dealer's face card. The program first determines the suggested strategy for the player and carries the strategy out in an off table simulation. The program plays the hand as if it were the player and then plays the dealer's hand in the event the player has not busted. This is carried out 2000 times against the dealer with a new shuffled deck each time. The number of times the player wins (or the dealer busting) is kept in a running tally. Once the loop is complete, the number of wins is divided by 20 to get a percent chance of victory, which is then displayed in a prompt to the user. This simulation takes roughly 6 seconds to run to completion

Splitting Cards

When the player gets dealt a hand where the 2 initial cards have identical values, they have the option of doubling their bet and splitting the 2 cards up into 2 separate hands. The first card is then dealt a second card to complete the initial hand and the player plays out that hand. Once the first hand is complete, the second card from the original hand is dealt a second card to complete that hand and the player plays it out normally. At the end the dealer plays out its hand and compares to the player's two hands separately.

The player is limited to only splitting once, so if another identical valued card is dealt to the first or second initial cards they cannot split again. The player is allowed to carry out each individual hand created like any normal hand by hitting, doubling down, getting the suggested strategy, getting the probability of success, and standing.

Code

Below you will find all of the code uncommented for reference. Each module and class is labeled. To see a commented version of the code please look at code in the Visual Basic Editor.

Class Bankroll

```
Dim bankroll As Double
Dim pBet As Integer

Public Sub Class_Initialize()
   bankroll = 1000
   pBet = 0
End Sub


Public Sub MakeBet(m_value)
   pBet = pBet + m_value
   bankroll = bankroll - m_value
End Sub

Public Sub Win()
   bankroll = bankroll + pBet * 2
   pBet = 0
End Sub

Public Sub WinSplit(ByVal ratio As Double)
   bankroll = bankroll + pBet * ratio * 2
   'pBet = pBet - pBet * ratio
End Sub

Public Sub LoseSplit(ByVal ratio As Double)
   'pBet = pBet - pBet * ratio
End Sub

Public Sub PushSplit(ByVal ratio As Double)
   bankroll = bankroll + pBet * ratio
   'pBet = pBet - pBet * ratio
End Sub

Public Sub Lose()
   pBet = 0
End Sub

Public Sub BlackJack()
   bankroll = bankroll + pBet * 2.5
   pBet = 0
End Sub

Public Sub InsuranceWin()
```

```
    bankroll = bankroll + pBet
    pBet = 0
End Sub

Public Sub InsuranceLose()
    bankroll = bankroll - pBet * 0.5
End Sub

Public Function GetBankroll() As Integer
    GetBankroll = bankroll
End Function

Public Function GetBet() As Integer
    GetBet = pBet
End Function

Public Sub Tie()
    bankroll = bankroll + pBet
    pBet = 0
End Sub
```

Class Card

```
Dim suit As String
Dim denom As String
Dim value As Integer
Dim IsUp As Boolean

Public Sub Class_Initialize()


End Sub

Public Sub SetCard(ByVal m_suit As Integer, ByVal m_value As Integer)
  value = m_value
    IsUp = True

    Select Case m_suit
      Case 1
        suit = "H"
      Case 2
        suit = "C"
      Case 3
        suit = "D"
      Case 4
        suit = "S"
    End Select


    Select Case m_value
      Case 1
        denom = "A"
        value = 11
      Case 11
        denom = "J"
        value = 10
      Case 12
        denom = "Q"
        value = 10
      Case 13
        denom = "K"
        value = 10
      Case Else
        denom = m_value
    End Select

    'SetCard = 1
```

```
End Sub

Public Sub TurnCardDown()
   IsUp = False
End Sub

Public Sub TurnCardUp()
   IsUp = True
End Sub

Public Function IsCardUp() As Boolean
   IsCardUp = IsUp
End Function

Public Function GetValue()
   GetValue = value
End Function

Public Function GetDenom()
   GetDenom = denom
End Function

Public Function GetSuit()
   GetSuit = suit
End Function

Public Sub SetValue(ByVal m_value As Integer)
   value = m_value
End Sub
```

Class Deck

```
Dim cardDeck() As card
Dim shuffConst As Integer
Dim position As Integer
Dim numCards As Integer

Public Sub Class_Initialize()
   numCards = 52
   ReDim cardDeck(numCards)
   Call CreateDeck
   shuffConst = 300
   position = 1
End Sub

Public Function DrawCard() As card
   position = position + 1

   Set DrawCard = cardDeck(position - 1)
End Function

Public Sub CreateDeck()
   Dim dcount As Integer
   Dim sCount As Integer
   Dim tmpCard As New card
   Dim i  As Integer

   tmpCard.Class_Initialize

   i = 0

   For dcount = 1 To 13
      For sCount = 1 To 4
         i = i + 1
         Set tmpCard = Nothing
         Call tmpCard.SetCard(sCount, dcount)

         Set cardDeck(i) = tmpCard
      Next sCount
   Next dcount



End Sub
```

```
Public Sub RemoveCard(ByVal m_denom As String, ByVal m_suit As String)
   Dim count, rCount As Integer

   For count = 1 To numCards
      If cardDeck(count).GetDenom = m_denom Then
         If cardDeck(count).GetSuit = m_suit Then
            For rCount = count To numCards - 1
               Set cardDeck(rCount) = cardDeck(rCount + 1)
            Next rCount

            numCards = numCards - 1
            ReDim Preserve cardDeck(numCards)
            Exit Sub
         End If
      End If
   Next count
End Sub

Public Sub Shuffle()
   Dim bigCount, sCount As Integer
   Dim rndVar As Integer
   Dim tmpCard1 As card

   position = 1
   Randomize

   For bigCount = 1 To shuffConst
      rndVar = Rnd() * (numCards - 1) + 1
      Set tmpCard1 = cardDeck(rndVar)

      For sCount = rndVar To 2 Step -1
         Set cardDeck(sCount) = cardDeck(sCount - 1)
      Next sCount

      Set cardDeck(1) = tmpCard1

   Next bigCount
End Sub
```

Class Hand

```
Dim hand() As card
Dim numCards As Integer
Dim handValue As Integer

Public Sub Class_Initialize()
   numCards = 2
   ReDim hand(numCards)
   handValue = 0
End Sub

Public Sub AddCard(ByVal card As card)
   numCards = numCards + 1
   ReDim Preserve hand(numCards)
   Set hand(numCards) = card
End Sub

Public Sub ClearHand()
   numCards = 2
   ReDim hand(numCards)
End Sub

Public Sub ResetHand()
   numCards = 2
   ReDim Preserve hand(numCards)
End Sub

Public Function GetNumCards()
   GetNumCards = numCards
End Function

Public Function GetCard(ByVal index As Integer) As card
   Set GetCard = hand(index)
End Function

Public Function value()
   Dim count As Integer
   Dim tmpValue As Integer
   Dim aceExists As Boolean

   aceExists = False
   tmpValue = 0
```

```
    For count = 1 To numCards
        If hand(count).GetDenom = "A" Then
            aceExists = True
            If tmpValue + hand(count).GetValue > 21 Then
              hand(count).SetValue (1)
            End If
        End If

        tmpValue = tmpValue + hand(count).GetValue
    Next count

    If tmpValue > 21 Then
        If aceExists Then
            For count = 1 To numCards
                If hand(count).GetDenom = "A" Then
                    If hand(count).GetValue = 11 Then
                        hand(count).SetValue (1)
                        tmpValue = tmpValue - 10
                    End If
                End If
            Next count
        End If
    End If

    value = tmpValue
End Function

Public Function GetHand() As card()
    Set GetHand = hand
End Function

Public Sub TurnCard(ByVal index As Integer, ByVal Direction As Boolean)
    If index <= numCards Then
        If Direction = True Then
            Call hand(index).TurnCardUp
        Else
            Call hand(index).TurnCardDown
        End If

    End If
End Sub

Public Sub SetFirst(ByVal first As card)
    Set hand(1) = first
End Sub
```

```
Public Sub SetSecond(ByVal second As card)
   Set hand(2) = second
End Sub

Public Sub SetHand(ByVal m_hand As hand, ByVal numCards As Integer)
   hand = m_hand
End Sub
```

Module 1

```
Option Explicit

Dim playerHand() As hand
Dim dealerHand As hand
Dim gameDeck As Deck

Dim money As bankroll
Dim isFirst As Boolean
Dim isSplit As Boolean
Dim handIndex As Integer
Dim handDouble() As Boolean
Dim baseBet As Integer
Const CARDS_PER_HAND As Integer = 7

'Enum Strategy

'End Enum

Public Sub Auto_Open()
   Call GameLoad
End Sub

Public Sub Bet1()
   If isFirst = False Then
      Call GameLoad
      isFirst = True
   End If

   If money.GetBankroll = 0 Then
      Call GameOver
      Exit Sub
   End If

   If money.GetBankroll > 1 Then
      money.MakeBet (1)
   Else
      money.MakeBet (money.GetBankroll)
      Call NoMoney
   End If

   Call SetCaptions
End Sub

Public Sub Bet5()
```

```
    If isFirst = False Then
        Call GameLoad
        isFirst = True
    End If

    If money.GetBankroll = 0 Then
        Call GameOver
        Exit Sub
    End If

    If money.GetBankroll > 5 Then
        money.MakeBet (5)
    Else
        money.MakeBet (money.GetBankroll)
        Call NoMoney
    End If

    Call SetCaptions
End Sub

Public Sub Bet25()
    If isFirst = False Then
        Call GameLoad
        isFirst = True
    End If

    If money.GetBankroll = 0 Then
        Call GameOver
        Exit Sub
    End If

    If money.GetBankroll > 25 Then
        money.MakeBet (25)
    Else
        money.MakeBet (money.GetBankroll)
        Call NoMoney
    End If

    Call SetCaptions
End Sub

Public Sub Bet100()
    If isFirst = False Then
        Call GameLoad
        isFirst = True
    End If
```

```
    If money.GetBankroll = 0 Then
        Call GameOver
        Exit Sub
    End If

    If money.GetBankroll > 100 Then
        money.MakeBet (100)
    Else
        money.MakeBet (money.GetBankroll)
        Call NoMoney
    End If

    Call SetCaptions
End Sub



Public Sub Bet500()
    If isFirst = False Then
        Call GameLoad
        isFirst = True
    End If

    If money.GetBankroll = 0 Then
        Call GameOver
        Exit Sub
    End If

    If money.GetBankroll > 500 Then
        money.MakeBet (500)
    Else
        money.MakeBet (money.GetBankroll)
        Call NoMoney
    End If

    Call SetCaptions
End Sub

Public Sub GameOver()
    Call SetCaptions
    MsgBox "You have no more money, you must start over", vbOKOnly, "Game Over"

End Sub

Public Sub NoMoney()
```

```
    MsgBox "You have no more money, hand will start", vbOKOnly, "Out of Money"
    Call Deal
End Sub

Public Sub GameLoad()
    handIndex = 1
    ReDim playerHand(handIndex)
    Set playerHand(handIndex) = New hand
    Set dealerHand = New hand
    Set gameDeck = New Deck
    Set money = New bankroll
    isFirst = True
    isSplit = False
    ReDim handDouble(2)

    Call SetCaptions

    Sheet1.Shapes("lblHit").Visible = msoFalse
    Sheet1.Shapes("lblStand").Visible = msoFalse
    Sheet1.Shapes("lblDouble").Visible = msoFalse
    Sheet1.Shapes("lblSplit").Visible = msoFalse
    Sheet1.Shapes("btnSuccess").Visible = msoFalse
    Sheet1.Shapes("Strategy").Visible = msoFalse
    Sheet1.Shapes("lblBet1").Visible = msoTrue
    Sheet1.Shapes("lblBet5").Visible = msoTrue
    Sheet1.Shapes("lblBet25").Visible = msoTrue
    Sheet1.Shapes("lblBet100").Visible = msoTrue
    Sheet1.Shapes("lblBet500").Visible = msoTrue
    Sheet1.Shapes("btnDeal").Visible = msoTrue

    Call ClearCards


End Sub

Public Sub ClearCards()
    Dim count As Integer
    Dim obj As Object

    For count = 1 To CARDS_PER_HAND * 2
        For Each obj In Sheet1.OLEObjects
            If obj.name = "Image" + CStr(count) Then
                Sheet1.Shapes("Image" + CStr(count)).Visible = msoFalse
            End If
        Next obj
    Next count
```

```
   Call Sheet1.changeValue(0)
End Sub

Public Sub DrawCards()

   Dim t As Object
   Dim obj As Object
   Dim SRC_PATH_STR As String
   Dim count As Integer

   SRC_PATH_STR = ThisWorkbook.Path + "\cards\"

   For count = 1 To dealerHand.GetNumCards
      For Each obj In Sheet1.OLEObjects
         If obj.name = "Image" + CStr(count) Then
            If dealerHand.GetCard(count).IsCardUp = True Then
               obj.Object.Picture = LoadPicture(SRC_PATH_STR +
dealerHand.GetCard(count).GetSuit + dealerHand.GetCard(count).GetDenom +
".bmp")
            Else
               obj.Object.Picture = LoadPicture(SRC_PATH_STR + "CARDBACK.bmp")
            End If

            Sheet1.Shapes("Image" + CStr(count)).Width = 40
            Sheet1.Shapes("Image" + CStr(count)).Height = 60
            Sheet1.Shapes("Image" + CStr(count)).Visible = msoTrue
         End If
      Next obj
   Next count

   For count = dealerHand.GetNumCards + 1 To CARDS_PER_HAND
      For Each obj In Sheet1.OLEObjects
         If obj.name = "Image" + CStr(count) Then
            Sheet1.Shapes("Image" + CStr(count)).Visible = msoFalse
         End If
      Next obj
   Next count

   For count = 1 To playerHand(handIndex).GetNumCards
      For Each obj In Sheet1.OLEObjects
         If obj.name = "Image" + CStr(count + CARDS_PER_HAND) Then
            If playerHand(handIndex).GetCard(count).IsCardUp = True Then
               obj.Object.Picture = LoadPicture(SRC_PATH_STR +
playerHand(handIndex).GetCard(count).GetSuit +
playerHand(handIndex).GetCard(count).GetDenom + ".bmp")
```

```
            Else
               obj.Object.Picture = LoadPicture(SRC_PATH_STR + "CARDBACK.bmp")
            End If

            Sheet1.Shapes("Image" + CStr(count + CARDS_PER_HAND)).Width = 40
            Sheet1.Shapes("Image" + CStr(count + CARDS_PER_HAND)).Height = 60
            Sheet1.Shapes("Image" + CStr(count + CARDS_PER_HAND)).Visible =
msoTrue
         End If
      Next obj
   Next count

   For count = playerHand(handIndex).GetNumCards + 1 To CARDS_PER_HAND
      For Each obj In Sheet1.OLEObjects
         If obj.name = "Image" + CStr(count + CARDS_PER_HAND) Then
            Sheet1.Shapes("Image" + CStr(count + CARDS_PER_HAND)).Visible =
msoFalse
         End If
      Next obj
   Next count

   For Each obj In Sheet1.OLEObjects
      If obj.name = "lblpValue" Then
         Call Sheet1.changeValue(playerHand(handIndex).value)
         'Sheet1.Shapes("lblpValue").Select
         'Selection.Text = CStr(playerHand(handIndex).value)
         'Range("P14").Select
      End If
   Next obj

   If dealerHand.GetCard(1).IsCardUp = True Then
      For Each obj In Sheet1.OLEObjects
         If obj.name = "lbldValue" Then
            Call Sheet1.changeDValue(dealerHand.value)
         End If
      Next obj
   Else
      Call Sheet1.changeDValue(0)
   End If

   Application.ScreenUpdating = True
End Sub

Public Sub SetCaptions()
   Call Sheet1.changeBet(money.GetBet)
   Call Sheet1.changeBankroll(money.GetBankroll)
```

```
    If isSplit = True Then
        Call Sheet1.handVisible
        Call Sheet1.changeHand(handIndex)
    Else
        Call Sheet1.handNotVisible
    End If
End Sub

Public Sub Deal()

    If isFirst = False Then
        Call GameLoad
        isFirst = True
    End If

    If money.GetBet = 0 Then
        MsgBox "You must make a bet"
        Exit Sub
    End If

    Sheet1.Shapes("lblHit").Visible = msoTrue
    Sheet1.Shapes("lblStand").Visible = msoTrue
    Sheet1.Shapes("lblDouble").Visible = msoTrue
    Sheet1.Shapes("btnSuccess").Visible = msoTrue
    Sheet1.Shapes("btnDeal").Visible = msoFalse
    Sheet1.Shapes("Strategy").Visible = msoTrue
    Sheet1.Shapes("lblBet1").Visible = msoFalse
    Sheet1.Shapes("lblBet5").Visible = msoFalse
    Sheet1.Shapes("lblBet25").Visible = msoFalse
    Sheet1.Shapes("lblBet100").Visible = msoFalse
    Sheet1.Shapes("lblBet500").Visible = msoFalse

    Call playerHand(handIndex).ClearHand
    Call dealerHand.ClearHand
    Call gameDeck.Shuffle

    Call playerHand(handIndex).SetFirst(gameDeck.DrawCard)
    Call dealerHand.SetFirst(gameDeck.DrawCard)
    Call playerHand(handIndex).SetSecond(gameDeck.DrawCard)
    Call dealerHand.SetSecond(gameDeck.DrawCard)
    Call dealerHand.TurnCard(1, False)
    Call DrawCards

    Dim tmpHand() As card
    Dim count As Integer
```

```
    ReDim tmpHand(playerHand(handIndex).GetNumCards)

    For count = 1 To playerHand(handIndex).GetNumCards
        Set tmpHand(count) = playerHand(handIndex).GetCard(count)
    Next count

    If tmpHand(1).GetDenom = "A" Or tmpHand(2).GetDenom = "A" Then
        If tmpHand(2).GetValue = 10 Or tmpHand(1).GetValue = 10 Then
            Call money.BlackJack
            MsgBox "You have a blackjack!", vbOKOnly, "Blackjack"
            Call EndHand
            Exit Sub
        End If
    End If

    If tmpHand(2).GetDenom = tmpHand(1).GetDenom Then
        Sheet1.Shapes("lblSplit").Visible = msoTrue
    End If

    For count = 1 To dealerHand.GetNumCards
        Set tmpHand(count) = dealerHand.GetCard(count)
    Next count

    If tmpHand(2).GetDenom = "A" Then
        Call Insurance(tmpHand(1).GetValue)
    End If


    'Set tmpHand = dealerHand.GetHand

    'If tmpHand(2).GetDenom = "A" Then

    'End If
End Sub

Public Sub Insurance(ByVal value As Integer)
    Dim answer As String

    answer = MsgBox("Dealer shows an Ace, would you like insurance?", vbQuestion +
vbYesNo, "Insurance")

    If answer = vbYes Then
        If money.GetBankroll < money.GetBet / 2 Then
            MsgBox "You do not have the money to cover half of your bet, you cannot have
insurance", vbOKOnly, "Insurance"
```

```
            Exit Sub
        End If


        If value = 10 Then
            MsgBox "Dealer has BlackJack", vbOKOnly, "Insurance"
            Call money.InsuranceWin
            Call dealerHand.TurnCard(1, True)
            Call DrawCards
            Call EndHand
        Else
            Call money.InsuranceLose
            MsgBox "Dealer has no BlackJack, continue hand.", vbOKOnly, "Insurance"
        End If
    Else
        If value = 10 Then
            MsgBox "Dealer has BlackJack", vbOKOnly, "Insurance"
            Call money.Lose
            Call dealerHand.TurnCard(1, True)
            Call DrawCards
            Call EndHand
        Else
            MsgBox "Dealer has no BlackJack, continue hand.", vbOKOnly, "Insurance"
        End If
    End If

    Call SetCaptions
End Sub

Public Sub Hit()
    Call playerHand(handIndex).AddCard(gameDeck.DrawCard)
    Call DrawCards

    If playerHand(handIndex).value > 21 Then
        Call Bust
        Exit Sub
    End If

    If playerHand(handIndex).GetNumCards > 2 Then
        Sheet1.Shapes("lblDouble").Visible = msoFalse
        Sheet1.Shapes("btnSuccess").Visible = msoFalse
        Sheet1.Shapes("Strategy").Visible = msoFalse
        Sheet1.Shapes("lblSplit").Visible = msoFalse
    End If
```

```
End Sub

Public Sub Bust()
  If isSplit = False Then
    Call money.Lose
    Call dealerHand.TurnCard(1, True)
    Call DrawCards
    Call SetCaptions
    MsgBox "You Busted", vbOKOnly, "You Lose"
    Call EndHand
  Else
    If handIndex = 1 Then
      If handDouble(1) = True Then
        Call money.LoseSplit(2 / 3)
      Else
        Call money.LoseSplit(0.5)
      End If

      Call DrawCards
      MsgBox "You Busted", vbOKOnly, "You Lose"
      handIndex = 2
      Call DrawCards
      Call SetCaptions
    Else
      If handDouble(1) = True And handDouble(2) = True Then
        Call money.LoseSplit(0.5)
      End If

      If handDouble(1) = False And handDouble(2) = False Then
        Call money.LoseSplit(0.5)
      End If

      If handDouble(1) = True And handDouble(2) = False Then
        Call money.LoseSplit(1 / 3)
      End If

      If handDouble(1) = False And handDouble(2) = True Then
        Call money.LoseSplit(2 / 3)
      End If

      MsgBox "You Busted", vbOKOnly, "You Lose"
    End If
  End If
End Sub

Public Sub DDown()
```

```
If playerHand(handIndex).GetNumCards = 2 Then
   If isSplit = True Then
      If money.GetBankroll < baseBet Then
         MsgBox "You do not have the funds to double", vbOKOnly, "Double Down"
         Exit Sub
      End If

      handDouble(handIndex) = True
      Call money.MakeBet(baseBet)
      Call playerHand(handIndex).AddCard(gameDeck.DrawCard)
      Call DrawCards
      Call SetCaptions

      If handIndex = 2 Then
         Sheet1.Shapes("lblDouble").Visible = msoFalse
         Sheet1.Shapes("btnSuccess").Visible = msoFalse
         Sheet1.Shapes("Strategy").Visible = msoFalse
         Sheet1.Shapes("lblSplit").Visible = msoFalse

         Call DrawCards
         Call SetCaptions
         Call DealerTurn
      Else
         handIndex = 2
         Call DrawCards
         Call SetCaptions
      End If
   Else
      If money.GetBankroll < money.GetBet Then
         MsgBox "You do not have the funds to double", vbOKOnly, "Double Down"
         Exit Sub
      End If

      Call money.MakeBet(money.GetBet)
      Call playerHand(handIndex).AddCard(gameDeck.DrawCard)
      Call DrawCards
      Call SetCaptions

      Sheet1.Shapes("lblDouble").Visible = msoFalse
      Sheet1.Shapes("btnSuccess").Visible = msoFalse
      Sheet1.Shapes("Strategy").Visible = msoFalse
      Sheet1.Shapes("lblSplit").Visible = msoFalse
      Call DrawCards
      Call SetCaptions
      'TODO: Disable Action Buttons
      Call DealerTurn
```

```
      End If
    End If
End Sub

Public Sub Split()

    If money.GetBankroll < money.GetBet Then
        MsgBox "You do not have the funds to split", vbOKOnly, "Split"
        Exit Sub
    End If

    Dim secondCard As New card
    isSplit = True

    Set secondCard = playerHand(1).GetCard(2)
    Call playerHand(1).SetSecond(gameDeck.DrawCard)

    ReDim Preserve playerHand(2)

    Set playerHand(2) = New hand

    Call playerHand(2).SetFirst(secondCard)
    Call playerHand(2).SetSecond(gameDeck.DrawCard)
    baseBet = money.GetBet
    Call money.MakeBet(money.GetBet)
    Sheet1.Shapes("lblSplit").Visible = msoFalse

    Call DrawCards
    Call SetCaptions
End Sub

Public Sub Stand()
    If isSplit = True Then
        If handIndex = 1 Then
            handIndex = 2
            Call DrawCards
            Call SetCaptions
            Exit Sub
        Else
            Call DealerTurn
            Exit Sub
        End If
    Else
        Call DealerTurn
    End If
End Sub
```

```
Public Sub Lose()
   If isSplit = False Then
      Call money.Lose
   Else
      If handDouble(1) = True And handDouble(2) = True Then
            Call money.LoseSplit(0.5)
      End If

      If handDouble(1) = False And handDouble(2) = False Then
         Call money.LoseSplit(0.5)
      End If

      If handDouble(1) = True And handDouble(2) = False And handIndex = 2 Then
         Call money.LoseSplit(1 / 3)
      End If

      If handDouble(1) = False And handDouble(2) = True And handIndex = 2 Then
         Call money.LoseSplit(2 / 3)
      End If

      If handDouble(1) = True And handDouble(2) = False And handIndex = 1 Then
         Call money.LoseSplit(2 / 3)
      End If

      If handDouble(1) = False And handDouble(2) = True And handIndex = 1 Then
         Call money.LoseSplit(1 / 3)
      End If

   End If

   MsgBox "Dealer Wins", vbOKOnly, "You Lose"
End Sub

Public Sub Win()
   If isSplit = False Then
      money.Win
   Else
      If handDouble(1) = True And handDouble(2) = True Then
            Call money.WinSplit(0.5)
      End If

      If handDouble(1) = False And handDouble(2) = False Then
         Call money.WinSplit(0.5)
      End If
```

```
      If handDouble(1) = True And handDouble(2) = False And handIndex = 2 Then
        Call money.WinSplit(1 / 3)
      End If

      If handDouble(1) = False And handDouble(2) = True And handIndex = 2 Then
        Call money.WinSplit(2 / 3)
      End If

      If handDouble(1) = True And handDouble(2) = False And handIndex = 1 Then
        Call money.WinSplit(2 / 3)
      End If

      If handDouble(1) = False And handDouble(2) = True And handIndex = 1 Then
        Call money.WinSplit(1 / 3)
      End If
    End If

    MsgBox "You Win!", vbOKOnly, "You Win"
End Sub

Public Sub Push()
    If isSplit = False Then
      money.Tie
    Else
      If handDouble(1) = True And handDouble(2) = True Then
        Call money.PushSplit(0.5)
      End If

      If handDouble(1) = False And handDouble(2) = False Then
        Call money.PushSplit(0.5)
      End If

      If handDouble(1) = True And handDouble(2) = False And handIndex = 2 Then
        Call money.PushSplit(1 / 3)
      End If

      If handDouble(1) = False And handDouble(2) = True And handIndex = 2 Then
        Call money.PushSplit(2 / 3)
      End If

      If handDouble(1) = True And handDouble(2) = False And handIndex = 1 Then
        Call money.PushSplit(2 / 3)
      End If

      If handDouble(1) = False And handDouble(2) = True And handIndex = 1 Then
        Call money.PushSplit(1 / 3)
```

```
      End If
    End If

    MsgBox "You pushed with the dealer", vbOKOnly, "Push"
End Sub

Public Sub DealerTurn()
    Call dealerHand.TurnCard(1, True)
    Call DrawCards

    If isSplit = False Then

      Call DrawCards
      Call SetCaptions

      While dealerHand.value < 17
        Call dealerHand.AddCard(gameDeck.DrawCard)
        Call DrawCards
        Call SetCaptions
      Wend

      If dealerHand.value > 21 Then
        Call DealerBust
        Exit Sub
      Else
        If dealerHand.value > playerHand(handIndex).value Then
          Call Lose
        Else
          If playerHand(handIndex).value > dealerHand.value Then
            Call Win
          Else
            Call Push
          End If
        End If
      End If

      Call EndHand
    Else
      If playerHand(1).value > 21 And playerHand(2).value > 21 Then
        Call EndHand
        Exit Sub
      End If

      If playerHand(1).value < 22 And playerHand(2).value < 22 Then
        While dealerHand.value < 17
          Call dealerHand.AddCard(gameDeck.DrawCard)
```

```
      Call DrawCards
    Wend

    If dealerHand.value > 21 Then
      Call DealerBust
      Exit Sub
    Else
      If dealerHand.value > playerHand(handIndex).value Then
        Call Lose
      Else
        If playerHand(handIndex).value > dealerHand.value Then
          Call Win
        Else
          Call Push
        End If
      End If

      handIndex = 1
      Call DrawCards

      If dealerHand.value > playerHand(handIndex).value Then
        Call Lose
      Else
        If playerHand(handIndex).value > dealerHand.value Then
          Call Win
        Else
          Call Push
        End If
      End If
    End If

    Call EndHand
    Exit Sub
  End If

  If playerHand(1).value < 22 And playerHand(2).value > 21 Then
    handIndex = 1
  End If

  Call DrawCards

  While dealerHand.value < 17
    Call dealerHand.AddCard(gameDeck.DrawCard)
    Call DrawCards
  Wend
```

```
      If dealerHand.value > 21 Then
        Call DealerBust
        Exit Sub
      Else
        If dealerHand.value > playerHand(handIndex).value Then
          Call Lose
        Else
          If playerHand(handIndex).value > dealerHand.value Then
            Call Win
          Else
            Call Push
          End If
        End If
      End If

      Call EndHand
    End If
End Sub

Public Sub EndHand()
    Sheet1.Shapes("lblHit").Visible = msoFalse
    Sheet1.Shapes("lblStand").Visible = msoFalse
    Sheet1.Shapes("lblDouble").Visible = msoFalse
    Sheet1.Shapes("lblSplit").Visible = msoFalse
    Sheet1.Shapes("btnSuccess").Visible = msoFalse
    Sheet1.Shapes("Strategy").Visible = msoFalse
    Sheet1.Shapes("lblBet1").Visible = msoTrue
    Sheet1.Shapes("lblBet5").Visible = msoTrue
    Sheet1.Shapes("lblBet25").Visible = msoTrue
    Sheet1.Shapes("lblBet100").Visible = msoTrue
    Sheet1.Shapes("lblBet500").Visible = msoTrue
    Sheet1.Shapes("btnDeal").Visible = msoTrue

    Call dealerHand.TurnCard(1, True)
    isSplit = False
    handIndex = 1
    Call money.Lose
    Call DrawCards
    Call SetCaptions

    handDouble(1) = False
    handDouble(2) = False

    Dim count, count2 As Integer

    handIndex = UBound(playerHand)
```

```
    For count2 = 1 To handIndex
        For count = 1 To playerHand(handIndex).GetNumCards
            If playerHand(handIndex).GetCard(count).GetDenom = "A" Then
                playerHand(handIndex).GetCard(count).SetValue (11)
            End If
        Next count
    Next count2



    For count = 1 To dealerHand.GetNumCards
        If dealerHand.GetCard(count).GetDenom = "A" Then
            dealerHand.GetCard(count).SetValue (11)
        End If
    Next count

    handIndex = 1
    ReDim playerHand(1)
    Set playerHand(1) = New hand
    'TODO: prepare interface for new hand (needed in multiple areas), check if all
money gone
End Sub

Public Sub DealerBust()
    MsgBox "Dealer Busted!", vbOKOnly, "You Win"

    If isSplit = False Then
        Call money.Win
    Else
        If playerHand(handIndex).value < 22 Then
            Call Win
        End If

        handIndex = 1
        Call DrawCards
        Call SetCaptions

        If playerHand(handIndex).value < 22 Then
            Call Win
        End If
    End If



    Call EndHand
```

```
End Sub

Public Sub ProbSuccess()
   'Dim stratTable As Variant
   Dim tmpHand() As card
   Dim tmpDeal() As card
   Dim tmpCard As card
   Dim tmpDeck As Deck
   Dim handsWon As Integer
   Dim netMoney, baseBet, totalBet As Integer
   Dim strategy As String
   Dim count As Integer
   Dim cont As Boolean


   handsWon = 0
   netMoney = 0
   totalBet = 0
   Set tmpCard = New card

   'Set tmpHand = playerHand(handIndex).GetHand()
   'Set tmpDeal = dealerHand.GetHand()

   ReDim tmpHand(playerHand(handIndex).GetNumCards)
   ReDim tmpDeal(dealerHand.GetNumCards)
   Set tmpDeck = New Deck

   For count = 1 To playerHand(handIndex).GetNumCards
      Set tmpHand(count) = playerHand(handIndex).GetCard(count)
   Next count

   For count = 1 To dealerHand.GetNumCards
      Set tmpDeal(count) = dealerHand.GetCard(count)
   Next count

   Set tmpCard = tmpDeal(1)
   baseBet = money.GetBet

   'If tmpHand(1).GetDenom = "A" Or tmpHand(2).GetDenom = "A" Then
   '    ReDim stratTable(2 To 9, 2 To 11)
   '    Call LoadAceTable(stratTable)
   'Else
   '    ReDim stratTable(4 To 20, 2 To 11)
   '    Call LoadNoAceTable(stratTable)
   'End If
```

```
cont = True

Call tmpDeck.RemoveCard(tmpHand(1).GetDenom, tmpHand(1).GetSuit)
Call tmpDeck.RemoveCard(tmpHand(2).GetDenom, tmpHand(2).GetSuit)
Call tmpDeck.RemoveCard(tmpDeal(1).GetDenom, tmpDeal(1).GetSuit)
Call tmpDeck.RemoveCard(tmpDeal(2).GetDenom, tmpDeal(2).GetSuit)

For count = 1 To 2000
   Call tmpDeck.Shuffle
   totalBet = baseBet
   netMoney = netMoney - totalBet


   While strategy <> "Stand"
      strategy = GetStrategy()

      If strategy = "Hit" Then
         Call playerHand(handIndex).AddCard(tmpDeck.DrawCard)
         If playerHand(handIndex).value > 21 Then
            strategy = "Stand"
         End If
      Else
         If strategy = "Double" Then
            totalBet = baseBet * 2
            Call playerHand(handIndex).AddCard(tmpDeck.DrawCard)
            strategy = "Stand"
         End If
      End If
   Wend

   If playerHand(handIndex).value < 22 Then
      Call dealerHand.SetFirst(tmpDeck.DrawCard)

      While dealerHand.value < 17
         Call dealerHand.AddCard(tmpDeck.DrawCard)
      Wend

      If dealerHand.value < 22 Then
         If playerHand(handIndex).value > dealerHand.value Then
            netMoney = netMoney + totalBet * 2
            handsWon = handsWon + 1
         End If

         If playerHand(handIndex).value < dealerHand.value Then
            netMoney = netMoney - totalBet
         End If
```

```
            Else
               netMoney = netMoney + totalBet
               handsWon = handsWon + 1
            End If
         Else
            netMoney = netMoney - totalBet
         End If

         Call playerHand(handIndex).ResetHand
         Call dealerHand.ResetHand
      Next count

      Call dealerHand.SetFirst(tmpCard)
      Dim success As Double

      success = handsWon / (10 * 2)

      MsgBox CStr(success) + "%"
   End Sub

   Public Function GetStrategy() As String
      Dim tmpHand() As card
      Dim tmpDHand() As card
      Dim strat As String
      Dim count As Integer

      ReDim tmpHand(playerHand(handIndex).GetNumCards)
      ReDim tmpDHand(dealerHand.GetNumCards)

      For count = 1 To playerHand(handIndex).GetNumCards
         Set tmpHand(count) = playerHand(handIndex).GetCard(count)
      Next count

      For count = 1 To dealerHand.GetNumCards
         Set tmpDHand(count) = dealerHand.GetCard(count)
      Next count

      'Set tmpHand = pHand.GetHand
      'Set tmpDHand = dHand.GetHand


         If (tmpHand(1).GetDenom = "A" Or tmpHand(2).GetDenom = "A") And
   playerHand(handIndex).GetNumCards > 2 Then
            Dim offValue As Integer

            If tmpHand(1).GetDenom = "A" Then
```

```
            offValue = tmpHand(2).GetValue
          Else
            offValue = tmpHand(1).GetValue
          End If

          If offValue >= 1 And offValue <= 6 Then
            strat = "Hit"
          Else
            strat = "Stand"
          End If
        Else
          If playerHand(handIndex).value >= 4 And playerHand(handIndex).value <= 9
Then
            strat = "Hit"
          End If

          If playerHand(handIndex).value = 10 Then
            If tmpDHand(2).GetValue < 10 Then
              strat = "Double"
            Else
              strat = "Hit"
            End If
          End If

          If playerHand(handIndex).value = 11 Then
            strat = "Double"
          End If

          If playerHand(handIndex).value = 12 Then
            If tmpDHand(2).GetValue > 3 And tmpDHand(2).GetValue < 7 Then
              strat = "Stand"
            Else
              strat = "Hit"
            End If
          End If


          If playerHand(handIndex).value > 12 And playerHand(handIndex).value < 17
Then
            If tmpDHand(2).GetValue > 1 And tmpDHand(2).GetValue < 7 Then
              strat = "Stand"
            Else
              strat = "Hit"
            End If
          End If
```

```
        If playerHand(handIndex).value > 16 Then
          strat = "Stand"
        End If
      End If

  GetStrategy = strat
End Function


Public Function GetMessageStrategy() As String
  Dim tmpHand() As card
  Dim tmpDHand() As card
  Dim strat As String
  Dim count As Integer

  ReDim tmpHand(playerHand(handIndex).GetNumCards)
  ReDim tmpDHand(dealerHand.GetNumCards)

  For count = 1 To playerHand(handIndex).GetNumCards
    Set tmpHand(count) = playerHand(handIndex).GetCard(count)
  Next count

  For count = 1 To dealerHand.GetNumCards
    Set tmpDHand(count) = dealerHand.GetCard(count)
  Next count

  'Set tmpHand = pHand.GetHand
  'Set tmpDHand = dHand.GetHand


    If (tmpHand(1).GetDenom = "A" Or tmpHand(2).GetDenom = "A") And
playerHand(handIndex).GetNumCards > 2 Then
      Dim offValue As Integer

      If tmpHand(1).GetDenom = "A" Then
        offValue = tmpHand(2).GetValue
      Else
        offValue = tmpHand(1).GetValue
      End If

      If offValue >= 1 And offValue <= 6 Then
        strat = "Hit"
      Else
        strat = "Stand"
      End If
    Else
```

```
        If tmpHand(1).GetDenom = tmpHand(2).GetDenom And
tmpHand(1).GetDenom = "A" Then
           strat = "Split"
        End If

        If tmpHand(1).GetValue = tmpHand(2).GetValue Then
           If tmpHand(1).GetDenom = "A" Then
              strat = "Split"
           End If

           If tmpHand(1).GetValue = 10 Then
              strat = "Stand"
           End If

           If tmpHand(1).GetValue = 9 Then
              If tmpDHand(2).GetValue = 7 Or tmpDHand(2).GetValue > 9 Then
                 strat = "Stand"
              Else
                 strat = "Split"
              End If
           End If

           If tmpHand(1).GetValue = 8 Then
              If tmpDHand(2).GetValue < 10 Then
                 strat = "Split"
              Else
                 strat = "Hit"
              End If
           End If

           If tmpHand(1).GetValue = 7 Then
              If tmpDHand(2).GetValue < 8 Then
                 strat = "Split"
              Else
                 strat = "Hit"
              End If
           End If

           If tmpHand(1).GetValue = 6 Then
              If tmpDHand(2).GetValue < 7 Then
                 strat = "Split"
              Else
                 strat = "Hit"
              End If
           End If
```

```
            If tmpHand(1).GetValue = 5 Then
               If tmpDHand(2).GetValue < 10 Then
                  strat = "Double"
               Else
                  strat = "Hit"
               End If
            End If

            If tmpHand(1).GetValue = 4 Then
               If tmpDHand(2).GetValue = 5 Or tmpDHand(2).GetValue = 6 Then
                  strat = "Split"
               Else
                  strat = "Hit"
               End If
            End If

            If tmpHand(1).GetValue = 2 Or tmpHand(1).GetValue = 3 Then
               If tmpDHand(2).GetValue < 8 Then
                  strat = "Split"
               Else
                  strat = "Hit"
               End If
            End If

         Else
            If playerHand(handIndex).value >= 4 And playerHand(handIndex).value <=
9 Then
               strat = "Hit"
            End If

            If playerHand(handIndex).value = 10 Then
               If tmpDHand(2).GetValue < 10 Then
                  strat = "Double"
               Else
                  strat = "Hit"
               End If
            End If

            If playerHand(handIndex).value = 11 Then
               strat = "Double"
            End If

            If playerHand(handIndex).value = 12 Then
               If tmpDHand(2).GetValue > 3 And tmpDHand(2).GetValue < 7 Then
                  strat = "Stand"
```

```
            Else
               strat = "Hit"
            End If
         End If


         If playerHand(handIndex).value > 12 And playerHand(handIndex).value <
17 Then
            If tmpDHand(2).GetValue > 1 And tmpDHand(2).GetValue < 7 Then
               strat = "Stand"
            Else
               strat = "Hit"
            End If
         End If

         If playerHand(handIndex).value > 16 Then
            strat = "Stand"
         End If
      End If


   End If

  GetMessageStrategy = strat
End Function


Public Sub LoadAceTable(ByRef table As Variant)
  Dim rCount, cCount As Integer

  For rCount = 2 To 6
    For cCount = 2 To 11
      table(rCount, cCount) = "H"
    Next cCount
  Next rCount

  For rCount = 7 To 9
    For cCount = 2 To 11
      table(rCount, cCount) = "S"
    Next cCount
  Next rCount

  table(7, 9) = "H"
  table(7, 10) = "H"
End Sub
```

```
Public Sub LoadNoAceTable(ByRef table As Variant)
   Dim rCount, cCount As Integer

   For rCount = 4 To 9
      For cCount = 2 To 11
         table(rCount, cCount) = "H"
      Next cCount
   Next rCount

   For rCount = 10 To 11
      For cCount = 2 To 9
         table(rCount, cCount) = "D"
      Next cCount
   Next rCount

   table(10, 10) = "H"
   table(10, 11) = "H"
   table(11, 10) = "D"
   table(11, 11) = "D"

   For rCount = 12 To 16
      For cCount = 2 To 6
         table(rCount, cCount) = "S"
      Next cCount
   Next rCount

   table(12, 2) = "H"
   table(12, 3) = "H"

   For rCount = 12 To 16
      For cCount = 7 To 11
         table(rCount, cCount) = "H"
      Next cCount
   Next rCount

   For rCount = 17 To 20
      For cCount = 2 To 11
         table(rCount, cCount) = "S"
      Next cCount
   Next rCount

End Sub
```

Module 3

```
Sub lblBet1_Click()
   Call Bet1
End Sub

Sub lblBet5_Click()
   Call Bet5
End Sub

Sub lblBet25_Click()
   Call Bet25
End Sub

Sub lblBet100_Click()
   Call Bet100
End Sub

Sub lblBet500_Click()
   Call Bet500
End Sub

Sub lblHit_Click()
   Call Hit
End Sub

Sub lblStand_Click()
   Call Stand
End Sub

Sub lblDouble_Click()
   Call DDown
End Sub

Sub btnDeal_Click()
   Call Deal
End Sub

Sub btnSuccess_Click()
   Call ProbSuccess
End Sub
```

Module 5

```
Sub Button40_Click()
   Call GameLoad
End Sub
```

Module 6

```
Sub Button50_Click()
   Dim strat As String
   strat = GetMessageStrategy
   MsgBox strat
End Sub
```

Module 7

```
Sub lblSplit_Click()
   Call Split
End Sub
```

Sheet1

```
Public Sub changeValue(ByVal value As Integer)
   lblpValue.Caption = CStr(value)
End Sub

Public Sub changeDValue(ByVal value As Integer)
   lbldValue.Caption = CStr(value)
End Sub

Public Sub changeBankroll(ByVal value As Integer)
   lblBankroll.Caption = CStr(value)
End Sub

Public Sub changeBet(ByVal value As Integer)
   lblBet.Caption = CStr(value)
End Sub

Public Sub changeHand(ByVal value As Integer)
   lblHand.Caption = "Hand " + CStr(value)
End Sub

Public Sub handVisible()
   lblHand.Visible = True
End Sub

Public Sub handNotVisible()
   lblHand.Visible = False
End Sub

Private Sub Image7_Click()

End Sub
```