

This is a work in progress...

```
' Card DLL Contest
' Anandis
' http://www.pagat.com/invented/anandis.html
' August, 2011
' Uses Qcard32.dll, a freeware library of playing card images.
' DLL by Stephen Murphy. Qcard32.DLL website:
' http://www.telusplanet.net/public/stevem/
```

```
' Rules
' 2 Players
' 17 card deck (4 Aces, 12 face cards, one Joker)
' 8 cards alternately dealt to each player
' Last card is placed face down in middle of table
' Object of the game is to guess the card in the middle
' Player opposite the dealer starts first
' Play options are
' Guess the card in the middle
' Correct guess = WIN game
' Incorrect guess = LOSE game
' Either way, game ends
' Play a card
' Opponent lays down all cards of the
' same suit and same rank
' Play then passes to the opposing player
' Pass the Joker
' If held, the player may pass the Joker
' to the opposing player, forcing that
' player to play
' A player with less than 2 cards MUST
' guess the middle card
' Winner is the best of 7 hands
```

```
WindowWidth = 800
WindowHeight = 500
UpperLeftX = Int((DisplayWidth - WindowWidth) / 2)
UpperLeftY = Int((DisplayHeight - WindowHeight) / 2)
graphicbox #main.g, 0, 0, 800, 500
```

```
open "Simple Anandis" for window as #main
#main "trapclose [quit]"
```

```
hGbox = hWnd(#main.g)
#main.g "down; color 0 192 0; bgcolor 0 192 0"
#main.g "font verdana 12 bold"
#main.g "place 0 0; boxfilled 800 500; flush bg"
#main.g "place 355 305; boxfilled 425 385"
#main.g "flush clearGuess"
#main.g "color black"
#main.g "place 365 325"
#main.g "\ ????"
#main.g "\Guess"
#main.g "\ Card"
#main.g "\ ????"
#main.g "flush playerGuess"
#main.g "color 0 192 0"

' Initialize the deck
call InitializeDeck hGbox
call SetCurrentBack 3

' Select (by number) only the cards to be used
' 11 = Jack Clubs, 24 = Jack Diamonds
' 37 = Jack Hearts, 50 = Jack Spades
dim dCards(17)
dCards(1) = 11
dCards(2) = 24
dCards(3) = 37
dCards(4) = 50
' 12 = Queen Clubs, 25 = Queen Diamonds
' 38 = Queen Hearts, 51 = Queen Spades
dCards(5) = 12
dCards(6) = 25
dCards(7) = 38
dCards(8) = 51
' 13 = King Clubs, 26 = King Diamonds
' 39 = King Hearts, 52 = King Spades
dCards(9) = 13
dCards(10) = 26
dCards(11) = 39
dCards(12) = 52
' 1 = Ace Clubs, 14 = Ace Diamonds
' 27 = Ace Hearts, 40 = Ace Spades
dCards(13) = 1
dCards(14) = 14
dCards(15) = 27
dCards(16) = 40
' 110 = Joker
```

```
dCards(17) = 110

' Set up array to hold nCards to shuffle
dim nCards(17)

' Set up an array to hold visibility of cards
dim iCards(17)

' Set up array to hold xCards to determine best guess
dim xCards(17)

' Fill arrays
call InitCards

' Set up array to hold cards
dim hand1(2, 8)
dim hand2(2, 8)
' (1, = computer's hand
' (2, = player's hand
' , 1-8) = nCard
' , 9: 1-8 = nCard

' Set GameInProgressFlag
GameInProgressFlag = 0

' Select dealer
dealer = int(rnd(1) * 2) + 1
if dealer = 1 then
human = 2
computer = 1
else
human = 1
computer = 2
end if

' Define coordinates
x1 = 50 ' x of dealt cards
x2 = 50 ' x of revealed cards
y1 = 300 ' y of human cards
y2 = 30 ' y of computer cards

[AnnounceDealer]
call AnnounceDealer dealer

[ShuffleCards]
' Remove cards from hand
```

```
redim hand1(2, 8)
redim hand2(2, 8)

' Shuffle the cards
call ShuffleCards dealer

' Deal the cards
call NewDeal dealer, hGbox

[AnnouncePlayer]
if dealer = human then
playerTurn = computer
else
playerTurn = human
end if
call AnnouncePlayer playerTurn

if playerTurn = human then
' Human's Turn
#main.g "when leftButtonDown [PlayerClick]"
wait
end if

' Computer's Turn
wait

[quit]
close #qc
close #main
end

[PlayerClick]
xPos = MouseX
yPos = MouseY
clickCard = -1
select case
case yPos < 300
clickCard = 0
case yPos > 395
clickCard = 0
case xPos > 400
clickCard = 0
case xPos > 352
clickCard = 9
case xPos > 260
```

```
if xPos < 330 then
clickCard = 8
else
clickCard = 0
end if
if clickCard = 8 then
if hand1(1, 8) = 0 then
clickCard = 0
if hand1(1, 7) > 0 then
if xPos < 300 then
clickCard = 7
end if
end if
end if
end if
case xPos > 230
if xPos < 300 then
clickCard = 7
else
clickCard = 0
end if
if clickCard = 7 then
if hand1(1, 7) = 0 then
clickCard = 0
if hand1(1, 6) > 0 then
if xPos < 270 then
clickCard = 6
end if
end if
end if
end if
case xPos > 200
if xPos < 270 then
clickCard = 6
else
clickCard = 0
end if
if clickCard = 6 then
if hand1(1, 6) = 0 then
clickCard = 0
if hand1(1, 5) > 0 then
if xPos < 240 then
clickCard = 5
end if
end if
end if
end if
```

```
end if
case xPos > 170
if xPos < 240 then
clickCard = 5
else
clickCard = 0
end if
if clickCard = 5 then
if hand1(1, 5) = 0 then
clickCard = 0
if hand1(1, 4) > 0 then
if xPos < 210 then
clickCard = 4
end if
end if
end if
end if
case xPos > 140
if xPos < 210 then
clickCard = 4
else
clickCard = 0
end if
if clickCard = 4 then
if hand1(1, 4) = 0 then
clickCard = 0
if hand1(1, 3) > 0 then
if xPos < 180 then
clickCard = 3
end if
end if
end if
end if
case xPos > 110
if xPos < 180 then
clickCard = 3
else
clickCard = 0
end if
if clickCard = 3 then
if hand1(1, 3) = 0 then
clickCard = 0
if hand1(1, 2) > 0 then
if xPos < 150 then
clickCard = 2
end if
end if
```

```
end if
end if
end if
case xPos > 80
if xPos < 150 then
clickCard = 2
else
clickCard = 0
end if
if clickCard = 2 then
if hand1(1, 2) = 0 then
clickCard = 0
if hand1(1, 1) > 0 then
if xPos < 120 then
clickCard = 1
end if
end if
end if
end if
case xPos > 50
if xPos < 120 then
clickCard = 1
else
clickCard = 0
end if
end select
if clickCard < 1 then
call RefreshCards hGbox, 1
call RefreshCards hGbox, 2
wait
end if
nCard = hand1(1, clickCard)
hand2(1, clickCard) = hand1(1, clickCard)
hand1(1, clickCard) = 0
call RefreshCards hGbox, 1
```

```
[computerDiscardsLikes]
' Find all invisible cards
for i = 1 to 17
iC = iCards(i)
for j = 1 to 8
if hand2(1, j) = iC then
iCards(i) = 0
end if
if hand1(2, j) = iC then
iCards(i) = 0
```

```
end if
if hand2(2, j) = iC then
iCards(i) = 0
end if
next j
next i
sort iCards(), 17, 1

nChoices = 0
for i = 1 to 17
if iCards(i) > 0 then
nChoices = nChoices + 1
end if
next i
cValue = GetCardValue(hand2(1, clickCard))
if cValue = 99 then
cValue = -1
end if
cSuit = GetCardSuit(hand2(1, clickCard))
for i = 1 to 8
if GetCardValue(hand1(2, i)) = cValue then
call SetCardStatus hand1(2, i), 1
hand2(2, i) = hand1(2, i)
hand1(2, i) = 0
end if
if GetCardSuit(hand1(2, i)) = cSuit then
call SetCardStatus hand1(2, i), 1
hand2(2, i) = hand1(2, i)
hand1(2, i) = 0
end if
next i

call RefreshCards hGbox, 2
wait
```

```
sub Pause ms
call dll #kernel32, "Sleep", _
ms as long, _
result as void
end sub
```

```
sub InitializeDeck hGbox
open "qcard32.dll" for dll as #qc
call dll #qc, "InitializeDeck", _
hGbox as ulong, _
```

```
    result as long
end sub
```

```
sub SetCardStatus nCard, face
'nC is number of card - 1-52 in first deck and
'53-104 in second deck, if used
'face: 0=facedown,1=faceup
calldll #qc, "SetCardStatus", _
nCard as long, _
face as long, _
result as void
end sub
```

```
sub DealCard hGbox, nCard, xPos, yPos
'places card on window whose handle is hGbox at xPos, yPos
'nC is number of card - 1-52 in first deck and
'53-104 in second deck, if used
calldll #qc, "DealCard", _
hGbox as ulong, _
nCard as long, _
xPos as long, _
yPos as long, _
result as void
end sub
```

```
sub RemoveCard hGbox,nCard
'removes a card from screen that was
'drawn with DealCard, replacing screen background
calldll #qc, "RemoveCard", _
hGbox as ulong, _
nCard as long, _
result as void
end sub
```

```
sub SetCurrentBack nDesign
'nDesign can be 1,2,3,4,5,6 for 6 possible designs
calldll #qc, "SetCurrentBack", _
nDesign as long, _
result as void
end sub
```

```
sub RefreshCards hGbox, player
for i = 1 to 8
call RemoveCard hGbox, hand1(player, i)
call RemoveCard hGbox, hand2(player, i)
next i
```

```
if player = 1 then
#main.g "place 40 295; boxfilled 340 400"
yPos = 300
else
#main.g "place 40 45; boxfilled 340 150"
yPos = 50
end if
x1Pos = 50
x2Pos = 450
for i = 1 to 8
n1Card = hand1(player, i)
if n1Card > 0 then
call DealCard hGbox, n1Card, x1Pos, yPos
end if
n2Card = hand2(player, i)
if n2Card > 0 then
call DealCard hGbox, n2Card, x2Pos, yPos
end if
x1Pos = x1Pos + 30
x2Pos = x2Pos + 30
next i
end sub
```

```
sub InitCards
for i = 1 to 17
nCards(i) = dCards(i)
iCards(i) = dCards(i)
xCards(i) = dCards(i)
next i
end sub
```

```
sub ShuffleCards dealer
for i = 17 to 2 step -1
n = int(rnd(1) * i) + 1
temp = nCards(i)
nCards(i) = nCards(n)
nCards(n) = temp
next i
if dealer = 1 then
hand1 = 2
hand2 = 1
else
hand1 = 1
hand2 = 2
end if
ct = 0
```

```
for i = 1 to 15 step 2
ct = ct + 1
hand1(hand1, ct) = nCards(i)
hand1(hand2, ct) = nCards(i + 1)
hand2(hand1, ct) = 0
hand2(hand2, ct) = 0
next i
questCard = nCards(17)
end sub

sub NewDeal dealer, hGbox
  xPos = 50
  if dealer=1 then
    hand1 = 2
    hand2 = 1
    vis1 = 0
vis1 = 1 ' Preserve computer CardStatus 0 for actual play
    vis2 = 1
    y1Pos = 50
    y2Pos = 300
  else
    hand1 = 1
    hand2 = 2
    vis1 = 1
    vis2 = 0
vis2 = 1 ' Preserve computer CardStatus to 0 for actual play
    y1Pos = 300
    y2Pos = 50
  end if
  for i = 1 to 8
    call Pause 250
    nCard = hand1(hand1, i)
    call SetCardStatus nCard, vis1
    call DealCard hGbox, nCard, xPos, y1Pos
    call Pause 250
    nCard = hand1(hand2, i)
    call SetCardStatus nCard, vis2
    call DealCard hGbox, nCard, xPos, y2Pos
    xPos = xPos + 30
  next i
  call Pause 500
  nCard = nCards(17)
  ' Reset Middle (Guess) CardStatus to 1 for actual play
  ' call SetCardStatus nCard, 0
  Call DealCard hGbox, nCard, 162, 175
end sub
```

```
sub AnnounceDealer dealer
  if dealer = 1 then
    msg$ = "Your Deal"
  else
    msg$ = "Computer Deals"
  end if
confirm msg$;yn$ ' Validates proper deal of cards
end sub
```

```
sub AnnouncePlayer playerTurn
  if playerTurn = 1 then
    msg$ = "Your Turn"
    y = 420
  else
    msg$ = "Computer's Turn"
    y = 40
  end if
  #main.g "stringwidth? msg$ width"
  #main.g "place ";int(200 - width/2);" ";y
  #main.g "\";msg$
end sub
```

```
function GetCardSuit(nCard)
  calldll #qc, "GetCardSuit", _
  nCard as long, _
  GetCardSuit as long
  'returns 1=Clubs, 2=Diamonds, 3=Hearts, 4=Spades
end function
```

```
function GetCardValue(nCard)
  calldll #qc, "GetCardValue", _
  nCard as long, _
  GetCardValue as long
  'ace=1, deuce=2...jack=11, queen=12, king=13, joker=99
end function
```

At this point, the only action is for the human to click a card, and the computer then reveals cards of the same value and same suit. Turn is never passed to the computer, so human can keep clicking away. All cards are face up. Eventually the computer's hand will be face down (with discards face up) and the middle (guess) card will be face down as well.
