

A revised submission of simple made-up game.

There is one "quirk" not yet fixed: when selecting the last card in a row, clicking outside the area of a normal card lap will cause an error.

```
'KenoCard by jaba
'Sept 09, 2011
'Rev. 1

'This program uses the qcard32.dll which must be in the same folder as
>this
'program.

dim card(52)                                'card index for full deck before shuffle
dim x(52), y(52)
dim d(20)
dim cs(7)

nomainwin
  WindowWidth=750:WindowHeight=600
  UpperLeftX=1:UpperLeftY=1

  menu #1, "&File", "P&lay again", [start], "E&xit", [quit]
  graphicbox #1.g, 0, 0, 750, 570
  open "KenoCard Game" for window_nf as #1
  #1 "trapclose [quit]"

hBox=hWnd(#1.g)

open "qcard32.dll " for dll as #qc
call InitializeDeck hBox

[start]
  #1.g "cls"
  cnt = 0
  cardSelected = 0
  redim cs(7)
  #1.g "font arial 12 bold;color black"
  #1.g "down; fill 10 225 127"
  #1.g "backcolor 10 225 127;place 20 20;\PLAY KENOCARD - ";
  "Click to select 7 cards. ";
  "3 or more matches and you win!"

[fillCardArray]
  for i = 1 to 52
```

```
        card(i)=i      ':print card(i)
next

[shuffleCards]
  for i = 1 to 52
    newIndex=int(rnd(0)*52)+1
    tempCard=card(i)
    card(i)=card(newIndex)
    card(newIndex)=tempCard
next

[dealDeck]
  y=30
  j=0
  for i = 1 to 52
    j=j+1
    call SetCardStatus card(i), 1
    call DealCard hBox, card(i), j*24, y
    x(i)=j*24:y(i)=y

    call Pause 20
    if i mod 26 = 0 then y = y+110:j=0
  next i

  gosub [drawBoxes]

#1.g "setfocus; when leftButtonDown [selectCard]"
wait

[drawBoxes]
#1.g "color yellow; size 2"
h=80:y=266
for i = 1 to 7
  x=i*h
  #1.g "place ";x;" ";y
  #1.g "box ";x+76;" ";y+104
next i
return

[selectCard]
mx=MouseX:my=MouseY
  'if mouse is within visible part of card, get
  'that card's index
  if my > 30 AND my < 130 then

    for i = 1 to 26
```

```
        if mx >x(i) AND mx <x(i)+24 then
            cardSelected=card(i)
            exit for
        end if
        next i
    end if

    if my > 140 AND my < 240 then

        for i = 27 to 52
        if mx >x(i) AND mx <x(i)+24 then
            cardSelected=card(i)
            exit for
        end if
        next i
    end if

'check if user selects same card more than once
gosub [checkDoubles]
if dbl then wait

'count number of selections
cnt=cnt+1

#1.g "place 24 260;color black;\ " ;7-cnt
cs(cnt)=cardSelected

call SetCardStatus cs(cnt), 1
call DealCard hBox, cs(cnt), ((cnt)*80)+2, 270
call Pause 20

if cnt=7 then [stopSelecting]

wait

[stopSelecting]
#1.g "place 30 410;color yellow;\Please wait..."
#1.g "setfocus; when leftButtonDown"

call Pause 1000

cnt=0
for i = 1 to 52
    newIndex=int(rnd(0)*52)+1
    tempCard=card(i)
```

```
    card(i)=card(newIndex)
    card(newIndex)=tempCard
next

for i = 1 to 20
    call SetCardStatus card(i), 1
    call DealCard hBox, card(i), i*31, 420  '24
    call Pause 20
    for j = 1 to 7
        if card(i)=cs(j) then cnt=cnt+1
    next j
next i
#1.g "place 30 535;color black"
#1.g "\You got " ;cnt; " matches."
if cnt >2 then
    gosub [youWin]
else
    gosub [youLose]
end if

    goto [playagain]
wait

[youWin]
#1.g "place 280 400;font arial 20 bold;color red;\YOU WIN!!"
return

[youLose]
#1.g "place 280 400;font arial 20 bold;color black;\YOU LOSE!!"
return

[checkDoubles]
dbl=0
for k = 1 to cnt
    if cardSelected = cs(cnt) then
        notice "Please make another selection."
        dbl=1
        cardSelected = 0
    end if
next k
return

[playagain]
timer 1000, [ok]
wait
[ok]
```

```
timer o

confirm "Play again?";yn$
if yn$ = "yes" then goto [start]

[quit]
close #qc
close #1
end

' =====
'subs and functions
' =====

sub Pause ms
  'pause ms number of milliseconds
  calldll #kernel32, "Sleep",_
  ms as long,_
  re as void
end sub

sub InitializeDeck hndle
  calldll #qc, "InitializeDeck",_
  hndle as ulong,_
  r as long
end sub

sub DealCard hndle,nC,x,y
  'places cards on window whose handle is hndle at x,y
  'nC is number of card - 1-52 in first deck and
  '53-104 in second deck, if used
  calldll #qc, "DealCard",_
  hndle as ulong,_
  nC as long,_
  x as long,_
  y as long,_
  r as void
end sub

sub SetCardStatus nC, 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",_
  nC as long,_
  face as long,_
end sub
```

```
r as void
end sub
```