import simplegui

import random

# load card sprite - 936x384 - source: jfitz.com

CARD\_SIZE = (72, 96)

CARD\_CENTER = (36, 48)

card\_images = simplegui.load\_image("http://storage.googleapis.com/codeskulptor-assets/cards\_jfitz.png")

CARD\_BACK\_SIZE = (72, 96)

CARD\_BACK\_CENTER = (36, 48)

card\_back = simplegui.load\_image("http://storage.googleapis.com/codeskulptor-assets/card\_jfitz\_back.png")

# initialize some useful global variables

in\_play = False

outcome = ""

script = ""

score = 0

# define globals for cards

SUITS = ('C', 'S', 'H', 'D')

RANKS = ('A', '2', '3', '4', '5', '6', '7', '8', '9', 'T', 'J', 'Q', 'K')

VALUES = {'A':1, '2':2, '3':3, '4':4, '5':5, '6':6, '7':7, '8':8, '9':9, 'T':10, 'J':10, 'Q':10, 'K':10}

class Card:

def \_\_init\_\_(self, suit, rank):

if (suit in SUITS) and (rank in RANKS):

self.suit = suit

self.rank = rank

else:

self.suit = None

self.rank = None

print "Invalid card: ", suit, rank

def \_\_str\_\_(self):

return self.suit + self.rank

def get\_suit(self):

return self.suit

def get\_rank(self):

return self.rank

def draw(self, canvas, pos):

card\_loc = (CARD\_CENTER[0] + CARD\_SIZE[0] \* RANKS.index(self.rank),

CARD\_CENTER[1] + CARD\_SIZE[1] \* SUITS.index(self.suit))

canvas.draw\_image(card\_images, card\_loc, CARD\_SIZE,

[pos[0] + CARD\_CENTER[0], pos[1] + CARD\_CENTER[1]], CARD\_SIZE)

class Hand:

def \_\_init\_\_(self):

self.hand = []

def \_\_str\_\_(self):

return "Hand contains " + " ".join(self.hand)

def add\_card(self, card):

self.hand.append(card.\_\_str\_\_())

def get\_value(self):

hand\_value = "".join(self.hand)

values\_1 = 0

for n in hand\_value:

if n in VALUES:

values\_1 += VALUES[n]

if "A" not in hand\_value:

return values\_1

else:

#values\_11 = values\_1 + 10 \* hand\_value.count("A")

values\_11 = values\_1 + 10

if values\_11 <= 21:

return values\_11

else:

return values\_1

def draw(self, canvas, pos):

for c in self.hand:

cd = Card(c[0],c[1])

cd.draw(canvas, [pos[0]+self.hand.index(c)\*90,pos[1]])

class Deck:

def \_\_init\_\_(self):

self.card\_list = []\*52

for suit in SUITS:

for rank in RANKS:

self.card\_list.append(Card(suit, rank))

def shuffle(self):

random.shuffle(self.card\_list)

def deal\_card(self):

deal = random.choice(self.card\_list)

self.card\_list.remove(deal)

return deal

def \_\_str\_\_(self):

check = []

for c in self.card\_list:

check.append(c.\_\_str\_\_())

return "Deck contains " + " ".join(check)

#define event handlers for buttons

def deal():

global outcome, script, in\_play, deal\_cards, dealer, player, score

if in\_play == True:

score = score - 1

deal\_cards = Deck()

deal\_cards.shuffle()

dealer = Hand()

player = Hand()

dealer.add\_card(deal\_cards.deal\_card())

dealer.add\_card(deal\_cards.deal\_card())

player.add\_card(deal\_cards.deal\_card())

player.add\_card(deal\_cards.deal\_card())

script = "Hit or Stand ?"

outcome = ""

in\_play = True

def hit():

global outcome, script, in\_play, deal\_cards, player, score

if in\_play == True:

player.add\_card(deal\_cards.deal\_card())

if player.get\_value() > 21:

outcome = "You Loss"

script = "New deal?"

score = score - 1

in\_play = False

elif player.get\_value() == 21 and dealer.get\_value() < 21:

outcome = "You Win!"

script = "New deal?"

score = score + 1

in\_play = False

def stand():

global outcome, script, in\_play, dealer, player, score

if in\_play:

while dealer.get\_value() <= 17:

dealer.add\_card(deal\_cards.deal\_card())

if dealer.get\_value() > 21:

outcome = "You Win!"

script = "New deal?"

score = score + 1

else:

if player.get\_value() > dealer.get\_value():

outcome = "You Win!"

script = "New deal?"

score = score + 1

else:

outcome = "You Loss"

script = "New deal?"

score = score - 1

in\_play = False

def new\_game():

global outcome, script, in\_play, deal\_cards, dealer, player, score

deal()

score = 0

def draw(canvas):

canvas.draw\_text('Blackjack', (50, 80), 50, 'Yellow')

canvas.draw\_text("score : " + str(score), (350, 80), 30, 'black')

canvas.draw\_text("Dealer", (50, 120), 30, 'black')

canvas.draw\_text("Player", (50, 320), 30, 'black')

canvas.draw\_text(outcome, (150, 120), 30, 'black')

canvas.draw\_text(script, (150, 320), 30, 'black')

canvas.draw\_text(str(player.get\_value()), (350, 320), 30, 'black')

#canvas.draw\_text(str(in\_play), (400, 350), 30, 'black')

if in\_play == True:

dealer.draw(canvas, [50, 150])

player.draw(canvas, [50, 350])

canvas.draw\_image(card\_back, CARD\_BACK\_CENTER, CARD\_BACK\_SIZE,

[50+ CARD\_BACK\_CENTER[0] , 150 + CARD\_BACK\_CENTER[1]],

CARD\_BACK\_SIZE)

else:

canvas.draw\_text(str(dealer.get\_value()), (350, 120), 30, 'black')

dealer.draw(canvas, [50, 150])

player.draw(canvas, [50, 350])

frame = simplegui.create\_frame("Blackjack", 600, 500)

frame.set\_canvas\_background("Green")

frame.add\_button("Deal", deal, 200)

frame.add\_button("Hit", hit, 200)

frame.add\_button("Stand", stand, 200)

frame.add\_label('')

frame.add\_button("New Game", new\_game, 200)

frame.add\_label('')

frame.add\_label('Rules:')

frame.add\_label('The dealer wins ties.')

frame.set\_draw\_handler(draw)

deal()

frame.start()