Higher or Lower

https://github.com/gargsand/HigherorLowerGame

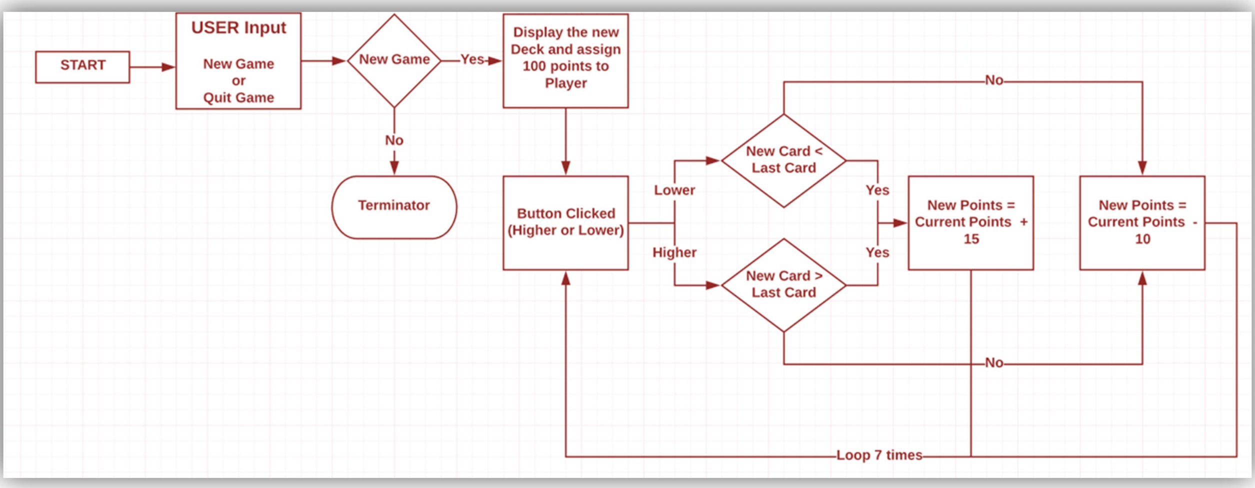
By: Sandeep Garg

Introduction

I have developed this Game in Python. The player has to guess if the next card will be Higher or Lower as compared to the last card. For every correct answer, points are increased by 15, and for wrong guess the points are reduced by 10. The player will be given 100 points initially to start with.

Description and Logic

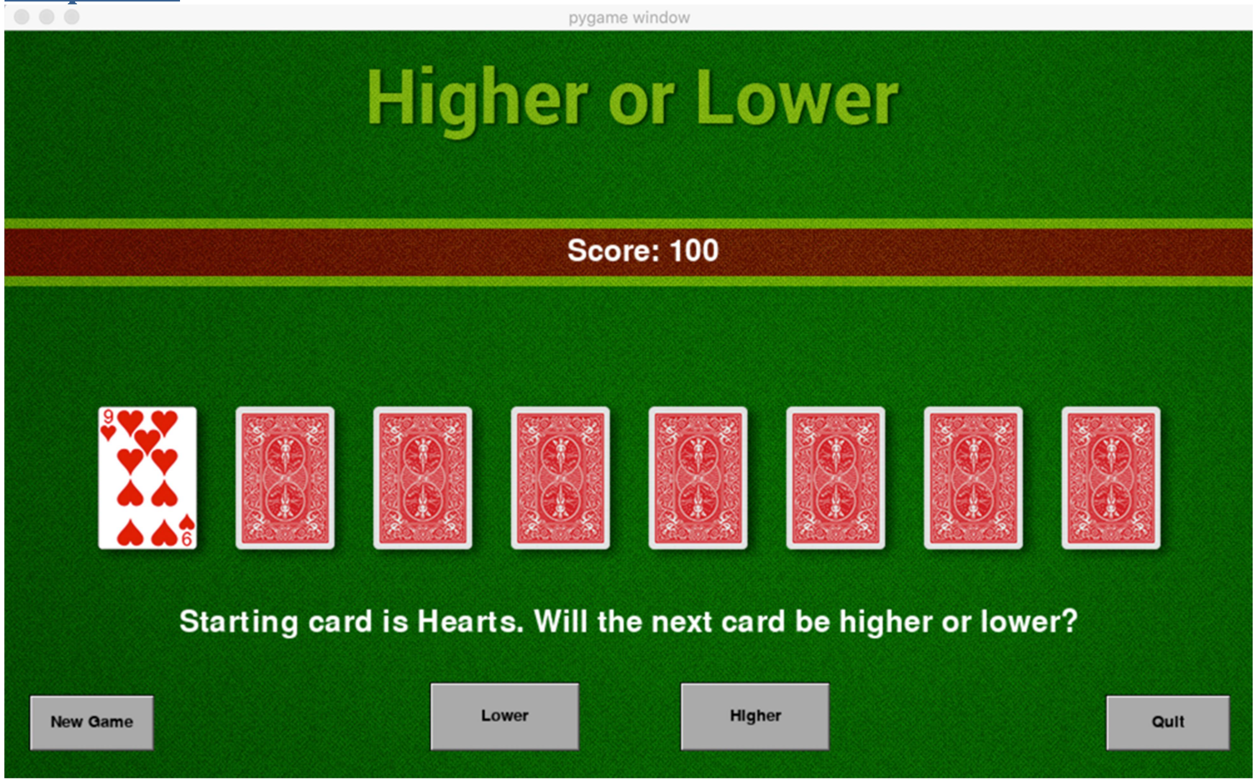
1. Major logic is written in three classes – Card, Deck, Game. All the final variables are defined in the Constants file and imported in all the classes and functions to be re-used.
    a. Class Card – Contains all the properties related to a Card
    b. Class Deck – Collection of Cards defined in step (a). Each Deck is collection of 52 cards, each card object having a face value and color associated to it
    c. Class Game – Uses the Deck Class with additional properties. e.g. When user clicks on New Game, Deck Class is shuffled and the points are reset.
2. I am using Library – "pygwidgets" written by Prof. Irv Kalb. This library is wrapper on collection of user interface widgets (e.g. buttons, frames, etc) written in Python. This takes care of building the complete user interface and refreshing it on user action. Details about this can by found at location https://pygwidgets.readthedocs.io/en/latest/
3. Concepts used in the included projects
    a. Classes {Game.py}
    b. Functions {Game.py, Main.py}
    c. Importing external modules {Game.py}
    d. Data Structures – tuple {CONSTANTS.py}
    e. User created iterators {Game.py && Main.py}
    f. Error checks using try-except {Main.py}
    g. User Input {Main.py}
    h. Decorator @staticmethod {Game.py}


START
USER Input
New Game or Quit Game
New Game
Yes
No
Terminator
Display the new Deck and assign 100 points to Player
Button Clicked (Higher or Lower)
Lower
Higher
New Card < Last Card
New Card > Last Card
Yes
Yes
No
No
New Points = Current Points + 15
New Points = Current Points - 10
Loop 7 times

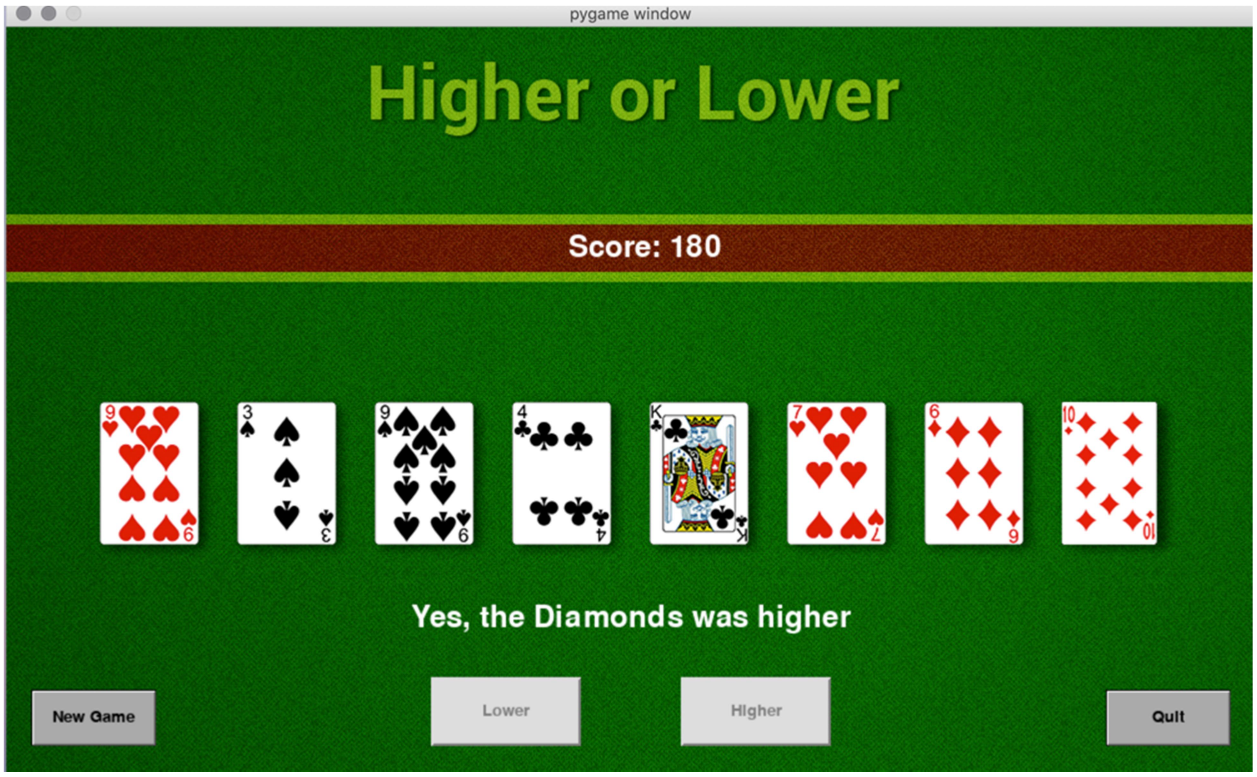
Requirements

- Pygame – This package is provided by Python to create User Interfaces

Snapshots

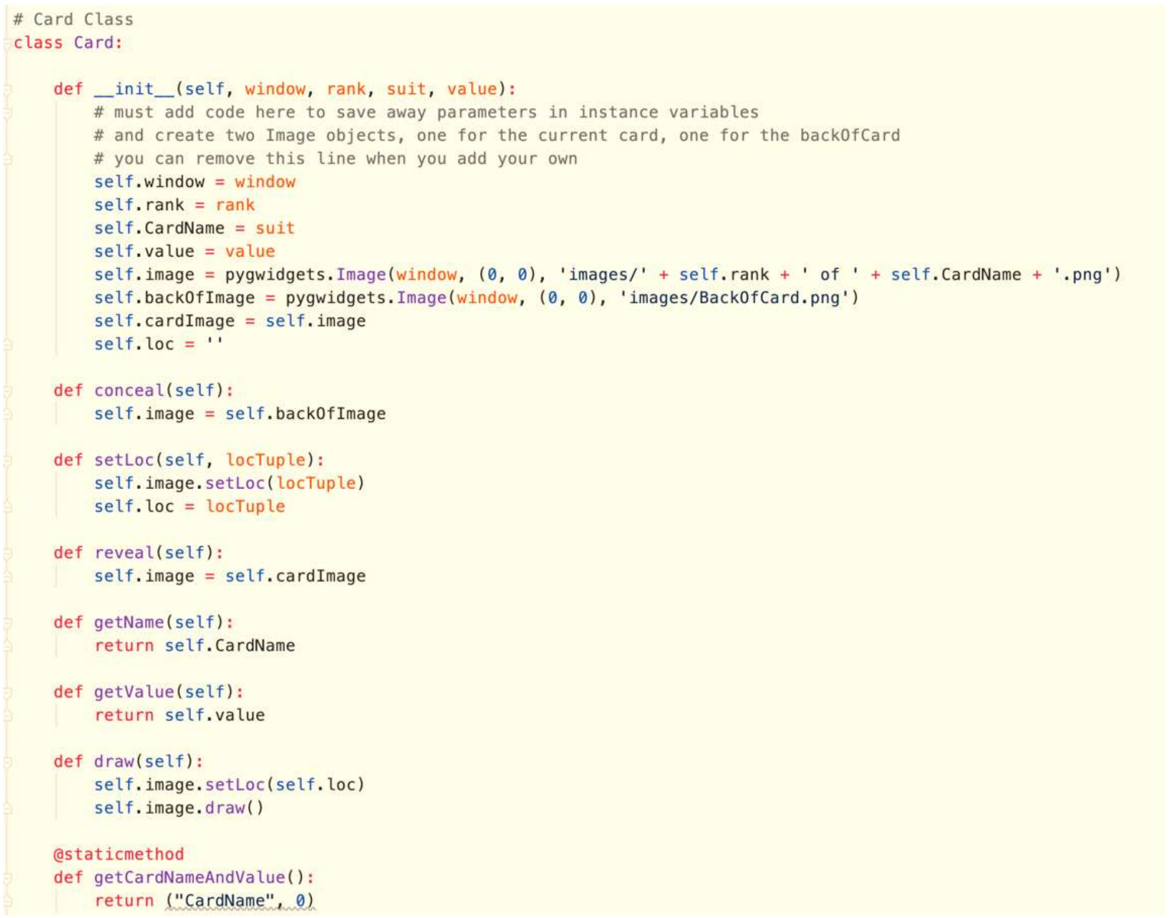


Starting card is Hearts. Will the next card be higher or lower?

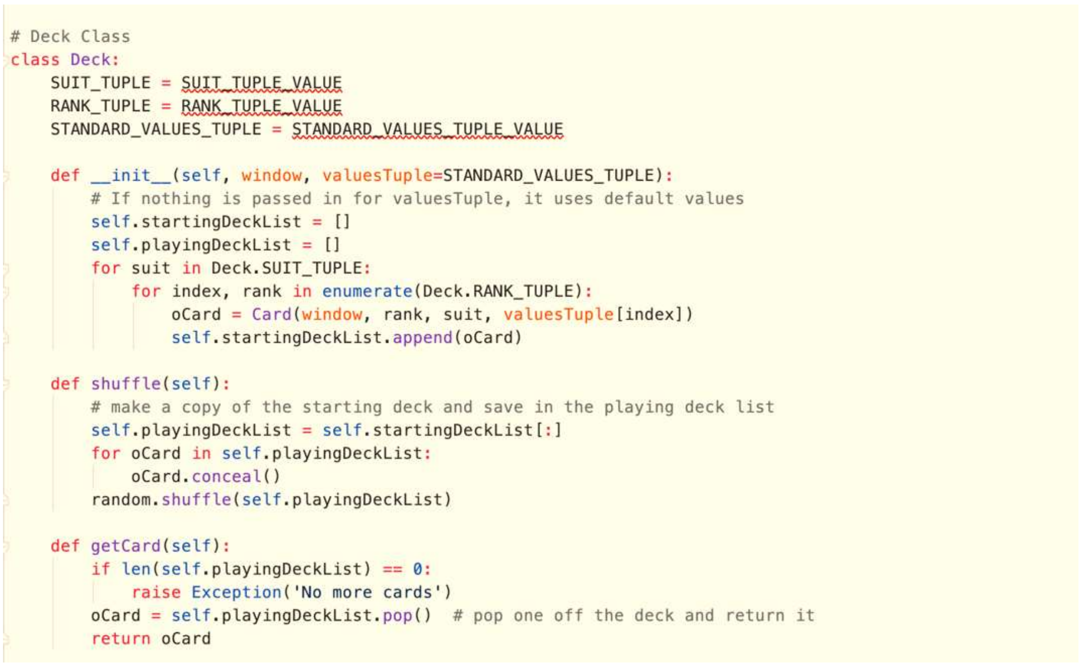


Yes, the Diamonds was higher

CARD, DECK and GAME Class –
https://github.com/gargsand/HigherorLowerGame/blob/master/Game.py

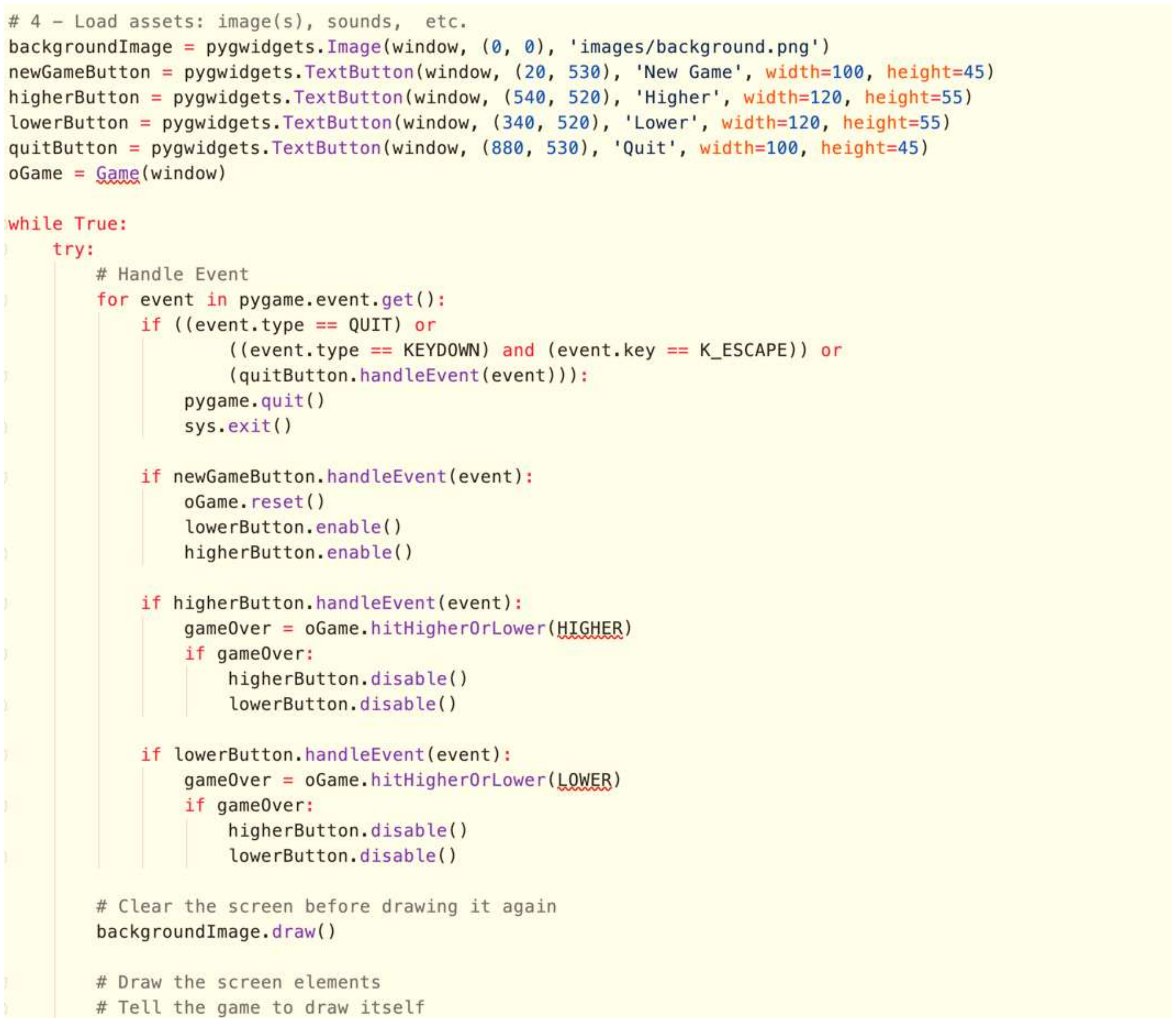


```python
# Card Class
class Card:

    def __init__(self, window, rank, suit, value):
        # must add code here to save away parameters in instance variables
        # and create two Image objects, one for the current card, one for the backOfCard
        # you can remove this line when you add your own
        self.window = window
        self.rank = rank
        self.CardName = suit
        self.value = value
        self.image = pygwidgets.Image(window, (0, 0), 'images/' + self.rank + ' of ' + self.CardName + '.png')
        self.backOfImage = pygwidgets.Image(window, (0, 0), 'images/BackOfCard.png')
        self.cardImage = self.image
        self.loc = ''

    def conceal(self):
        self.image = self.backOfImage

    def setLoc(self, locTuple):
        self.image.setLoc(locTuple)
        self.loc = locTuple

    def reveal(self):
        self.image = self.cardImage

    def getName(self):
        return self.CardName

    def getValue(self):
        return self.value

    def draw(self):
        self.image.setLoc(self.loc)
        self.image.draw()

    @staticmethod
    def getCardNameAndValue():
        return ("CardName", 0)
```



```python
# Deck Class
class Deck:
    SUIT_TUPLE = SUIT_TUPLE_VALUE
    RANK_TUPLE = RANK_TUPLE_VALUE
    STANDARD_VALUES_TUPLE = STANDARD_VALUES_TUPLE_VALUE

    def __init__(self, window, valuesTuple=STANDARD_VALUES_TUPLE):
        # If nothing is passed in for valuesTuple, it uses default values
        self.startingDeckList = []
        self.playingDeckList = []
        for suit in Deck.SUIT_TUPLE:
            for index, rank in enumerate(Deck.RANK_TUPLE):
                oCard = Card(window, rank, suit, valuesTuple[index])
                self.startingDeckList.append(oCard)

    def shuffle(self):
        # make a copy of the starting deck and save in the playing deck list
        self.playingDeckList = self.startingDeckList[:]
        for oCard in self.playingDeckList:
            oCard.conceal()
        random.shuffle(self.playingDeckList)

    def getCard(self):
        if len(self.playingDeckList) == 0:
            raise Exception('No more cards')
        oCard = self.playingDeckList.pop()  # pop one off the deck and return it
        return oCard
```

Main.py https://github.com/gargsand/HigherorLowerGame/blob/master/Main.py



```
# 4 - Load assets: image(s), sounds,  etc.
backgroundImage = pygwidgets.Image(window, (0, 0), 'images/background.png')
newGameButton = pygwidgets.TextButton(window, (20, 530), 'New Game', width=100, height=45)
higherButton = pygwidgets.TextButton(window, (540, 520), 'Higher', width=120, height=55)
lowerButton = pygwidgets.TextButton(window, (340, 520), 'Lower', width=120, height=55)
quitButton = pygwidgets.TextButton(window, (880, 530), 'Quit', width=100, height=45)
oGame = Game(window)

while True:
    try:
        # Handle Event
        for event in pygame.event.get():
            if ((event.type == QUIT) or
                    ((event.type == KEYDOWN) and (event.key == K_ESCAPE)) or
                    (quitButton.handleEvent(event))):
                pygame.quit()
                sys.exit()

            if newGameButton.handleEvent(event):
                oGame.reset()
                lowerButton.enable()
                higherButton.enable()

            if higherButton.handleEvent(event):
                gameOver = oGame.hitHigherOrLower(HIGHER)
                if gameOver:
                    higherButton.disable()
                    lowerButton.disable()

            if lowerButton.handleEvent(event):
                gameOver = oGame.hitHigherOrLower(LOWER)
                if gameOver:
                    higherButton.disable()
                    lowerButton.disable()

        # Clear the screen before drawing it again
        backgroundImage.draw()

        # Draw the screen elements
        # Tell the game to draw itself
```