****

**A**

**PROJECT REPORT**

**ON**

**“PYTHON CARD GAME”**

**SUBMITED**

**TO**

**SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE**

**FOR THE AWARD OF**

**MASTER OF COMPUTER APPLICATION**

**(MCA-I, SEM.-2)**

**BY**

SHRADDHA SHINDE(21256)

**UNDER THE GUIDANCE OF**

**Dr.ASHWINI BRAMHE**

**SINHGAD INSTITUTE OF MANAGEMENT AND COMPUTER APPLICATION (SIMCA), NARHE, PUNE**

**(AY. 2021-2022)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **SINHGAD TECHNICAL EDUCATION SOCIETY’S**  **SINHGAD INSTITUTE OF**  **MANAGEMENT & COMPUTER APPLICATION**  **(Affiliated to SavitribaiPhule Pune University &Appoved by AICTE)**  **‘NAAC’ Accredited with ‘A’ Grade** | | |  |
| **S. No. 49/1, Off Westernly Bypass, Pune – Mumbai Expressway, Narhe, Pune – 411041, Tel : (020) 66831896 / 66831908 / 66831907**  **E-mail : director\_mca\_simca@sinhgad.edu Website : www.sinhgad.edu** | | | | |
| **33**  **Prof. M. N. Navale**  M.E. (Elec), MIE, MBA  FOUNDER PRESIDENT | | **Dr. (Mrs.) Sunanda M. Navale**  B.A., P.P.M., Ph.D.  FOUNDER SECRETARY | **Dr. Vilas Nandavadekar** M.C.A., M.P.M., M.B.A., Ph. D.DIRECTOR, SIMCA - MCA | |

**CERTIFICATE**

This is to certify that,the project entitled "**python playing card game"** submitted for the partial fulfillment of **Master of Computer Application** by her/him to **Sinhgad Institute of Management and Computer Application affiliated to Savitribai Phule Pune University, Pune** is the result of the original work completed by AISHWARYA SHINDE under the guidance of Amar Sir

To the best of our knowledge and belief, this work has not been previously submitted by the award of any degree or diploma of Savitribai Phule Pune University or any other University.

PLACE:

DATE:

**Dr.Ashwini brahme Dr. Vilas Nandavadekar**

**Amar Sir**

**Internal Guide Project Coordinator Director SIMCA-MCA**

**External Examiner**

|  |
| --- |
| **DECLARATION**  I, the undersigned hereby declare that the project titled“Pyhon card game**”,**being submitted for partial fulfillment of **Master of Computer Application** by me to **Shinhgad Institute of Management and Computer Application(SIMCA) affiliated to Savitribai Phule Pune University** is the result of an independent work carried out under the guidance of **Ashwini bramhe,** is my original work . Further I declare that this project has not been submitted to this or any Institution for the award of any degree.  Shraddha shinde  (Student)  PLACE: PUNE  DATE: |
|  |
|  |
|  |
|  |

**ACKNOWLEDGEMENT**

The project developed for the MCA was not possible without the persons and organizations that helped me in completing this. I am deeply grateful to all whose enthusiasm and energy transformed my vision of this study into reality.

I take this opportunity to thank my guide **guide name**, project coordinator **Dr. Ashwini Bramhe** and our Director **Dr. Vilas Nandavadaker**, for encouragement and guidance throughout the progress of this report.

**Student Name**

INDEX

Project Report Format

1. **Application Development**

|  |  |  |
| --- | --- | --- |
| **Chapter No** |  | **Details** |
| **1** |  | **Introduction** |
|  | 1.1 | Company Profile / Institute Profile / Client Profile |
|  | 1.2 | Abstract |
|  | 1.3 | Existing System and Need for System |
|  | 1.4 | Scope of System |
|  | 1.5 | Operating Environment - Hardware and Software |
|  | 1.6 | Brief Description of Technology Used   * + 1. Operating systems used (Windows or Unix)     2. RDBMS/No Sql used to build database (MySQL/ oracle, Teradata, etc.) |
| **2** |  | **Proposed System** |
|  | 2.1 | Study of Similar Systems (If required research paper can be included) |
|  | 2.2 | Feasibility Study |
|  | 2.3 | Objectives of Proposed System |
|  | 2.4 | Users of System |
| **3** |  | **Analysis and Design** |
|  | 3.1 | System Requirements (Functional and Non-Functional requirements) |
|  | 3.2 | Entity Relationship Diagram (ERD) |
|  | 3.3 | Table Structure |
|  | 3.4 | Use Case Diagrams |
|  | 3.5 | Class Diagram |
|  | 3.6 | Activity Diagram |
|  | 3.7 | Deployment Diagram |
|  | 3.8 | Module Hierarchy Diagram |
|  | 3.9 | Sample Input and Output Screens (Screens must have valid data. All  reports must have at-least 5 valid records.) |
| **4** |  | **Coding** |
|  | 4.1 | Algorithms |
|  | 4.2 | Code snippets |
| **5** |  | **Testing** |
|  | 5.1 | Test Strategy |
|  | 5.2 | Unit Test Plan |
|  | 5.3 | Acceptance Test Plan |
|  | 5.4 | Test Case / Test Script |
|  | 5.5 | Defect report / Test Log |

1. **Embedded Systems / IoT**

|  |  |  |
| --- | --- | --- |
| **Chapter No** |  | **Details** |
| **1** |  | **Introduction** |
|  | 1.1 | Company Profile / Institute Profile / Client Profile |
|  | 1.2 | Abstract |
|  | 1.3 | Existing System and Need for System |
|  | 1.4 | Scope of System |
|  | 1.5 | Operating Environment - Hardware and Software |
|  | 1.6 | Brief Description of Technology Used   * + 1. Operating systems used (Windows or Unix)     2. Database (if applicable) |
| **2** |  | **Proposed System** |
|  | 2.1 | Study of Similar Systems (If required research paper can be included) |
|  | 2.2 | Feasibility Study |
|  | 2.3 | Objectives of Proposed System |
|  | 2.4 | Users of System |
| **3** |  | **Analysis and Design** |
|  | 3.1 | Technical requirements – H/W, S/W |
|  | 3.2 | System Architecture / Block Diagram |
|  | 3.3 | System Hardware Details |
|  | 3.4 | Pin Diagrams |
|  | 3.5 | Interface diagrams |
|  | 3.6 | Design Sequence |
|  | 3.7 | System Software Details |
|  | 3.8 | Process / System Flow chart |
| **4** |  | **Coding** |
|  | 4.1 | Algorithms |
|  | 4.2 | Code snippets (if applicable) |

**Abstract :**

The modern web has quickly become a viable platform not only for creating stunning, high quality games, but also for distributing those games.

The range of games that can be created is on par with desktop and native OS counterparts. With modern Web technologies and a recent browser, it's entirely possible to make stunning, top-notch games for the Web. And primarily here we are talking about simple card games.

* **Acknowledgments :**

We consider ourself priveledge to express gratitude and respect towards all those who guided us through the completion of this project .

We convey thanks to our project guide **Prof. ASHWINI BRAMHE** Ma’am from Information Technology Department for providing support and guidance which was great help to complete this project successfully.

* **Introduction**

Our project topic is basically a two player multiplayer card game development using a simple python programming structures.Which includes Multidisciplinary Process, Input Control ,Game Logic,etc.

* **Problem Statement & Objectives**

This concept is designed to teach you the foundations in order to make simple programs in Python using the most common structures. No previous exposure to programming is needed.

Design a Game which can be played in-house between two players while players can sit and get engaged physically and mentally.

* **Literature Survey**

Computer games and video games have become very popular in children and adolescents’ life and play a prominent role in the culture of young people

Games can now be played everywhere in technology with laptops, smart phones, game consoles and other digital devices.

Besides of an abundant appearance of games in young students life, game development technology has matured and became more advanced than before Based on various existing game development software, the whole duty of game development process can be divided into several domains and roles such as game programmers, 3D model creators, game designers, musicians, animators, and play writers.

This makes it possible for students to modify existing games or develop their ownnew games with or without programming. They can design and implement their own game concepts with these game creation tools, learn the developing skills and relevant knowledge, and accumulate related practical experience.

* **List Of Tables**

PLAY

OPTION

QUIT

WINNER

* **Proposed System And Methodology**

Python is the most popular programming language or nothing wrong to say that it is the next-generation programming language. In every emerging field in computer science, Python makes its presence actively.

Python has vast libraries for various fields such as Machine Learning (Numpy, Pandas, Matplotlib), Artificial intelligence (Pytorch, TensorFlow), and Game development (Pygame,Pyglet).

Pygame

* Pygame is a cross-platform set of Python modules which is used to create video games.
* It consists of computer graphics and sound libraries designed to be used with the Python programming language.
* Pygame was officially written by Pete Shinners to replace PySDL.
* Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable
* **How To Play**

1. A standard 52-card deck comprises 13 ranks in each of the four suits: clubs (♣), diamonds (♦), hearts (♥) and spades (♠).

2. Two players can play card game against each other.

From the pack of 52 cards randomly 4 cards are distributed to both players.

3. Player have to select required card from the pack of cards provided by the system randomly.

4. Opportunity to replace a card from pack of card is given in one by one.

5. For win the game, players have to make a collection of 4 cards having the same rank.

6. The player who collect 4 card of same rank at first ,will win a game.

7.The Congratulation SMS is send to the player who win the game.

* **Details of Language, Libraries and Software**

Software used :- visual Studio Code

Programming language :- python

Module :- pygame, random, twilio

* **Code Of Game**

# card pixel -->  horizontal -> 104 vertical-> 160

frommultiprocessing.connectionimportwait

fromtimeimportsleep

importpygame

importrandom

fromtwilio.restimport Client

COLOR\_INACTIVE = pygame.Color('lightskyblue3')

COLOR\_ACTIVE = pygame.Color('dodgerblue2')

player1\_name=''

player2\_name=''

player1\_no=''

player2\_no=''

pygame.init()

screen = pygame.display.set\_mode((1200, 700))

image\_string={"1🔸":r"C:\Users\shahi\Pictures\python\_card\_image\AD.jpeg",

              "2🔸":r"C:\Users\shahi\Pictures\python\_card\_image\2D.jpeg",

              "3🔸":r"C:\Users\shahi\Pictures\python\_card\_image\3D.jpeg",

              '4🔸':r"C:\Users\shahi\Pictures\python\_card\_image\4D.jpeg",

              '5🔸':r"C:\Users\shahi\Pictures\python\_card\_image\5D.jpeg",

              '6🔸':r"C:\Users\shahi\Pictures\python\_card\_image\6D.jpeg",

              '7🔸':r"C:\Users\shahi\Pictures\python\_card\_image\7D.jpeg",

              '8🔸':r"C:\Users\shahi\Pictures\python\_card\_image\8D.jpeg",

              '9🔸':r"C:\Users\shahi\Pictures\python\_card\_image\9D.jpeg",

              '10🔸':r"C:\Users\shahi\Pictures\python\_card\_image\10D.jpeg",

              '11🔸':r"C:\Users\shahi\Pictures\python\_card\_image\JD.jpeg",

              '12🔸' :r"C:\Users\shahi\Pictures\python\_card\_image\QD.jpeg",

              '13🔸':r"C:\Users\shahi\Pictures\python\_card\_image\KD.jpeg",

              "1❤":r"C:\Users\shahi\Pictures\python\_card\_image\AH.jpeg",

              "2❤":r"C:\Users\shahi\Pictures\python\_card\_image\2H.jpeg",

              "3❤":r"C:\Users\shahi\Pictures\python\_card\_image\3H.jpeg",

              '4❤':r"C:\Users\shahi\Pictures\python\_card\_image\4H.jpeg",

              '5❤':r"C:\Users\shahi\Pictures\python\_card\_image\5H.jpeg",

              '6❤':r"C:\Users\shahi\Pictures\python\_card\_image\6H.jpeg",

              '7❤':r"C:\Users\shahi\Pictures\python\_card\_image\7H.jpeg",

              '8❤':r"C:\Users\shahi\Pictures\python\_card\_image\8H.jpeg",

              '9❤':r"C:\Users\shahi\Pictures\python\_card\_image\9H.jpeg",

              '10❤':r"C:\Users\shahi\Pictures\python\_card\_image\10H.jpeg",

              '11❤':r"C:\Users\shahi\Pictures\python\_card\_image\JH.jpeg",

              '12❤':r"C:\Users\shahi\Pictures\python\_card\_image\QH.jpeg",

              '13❤':r"C:\Users\shahi\Pictures\python\_card\_image\KH.jpeg",

              "1☘":r"C:\Users\shahi\Pictures\python\_card\_image\AC.jpeg",

              "2☘":r"C:\Users\shahi\Pictures\python\_card\_image\2C.jpeg",

              "3☘":r"C:\Users\shahi\Pictures\python\_card\_image\3C.jpeg",

              '4☘':r"C:\Users\shahi\Pictures\python\_card\_image\4C.jpeg",

              '5☘':r"C:\Users\shahi\Pictures\python\_card\_image\5C.jpeg",

              '6☘':r"C:\Users\shahi\Pictures\python\_card\_image\6C.jpeg",

              '7☘':r"C:\Users\shahi\Pictures\python\_card\_image\7C.jpeg",

              '8☘':r"C:\Users\shahi\Pictures\python\_card\_image\8C.jpeg",

              '9☘':r"C:\Users\shahi\Pictures\python\_card\_image\9C.jpeg",

              '10☘':r"C:\Users\shahi\Pictures\python\_card\_image\10C.jpeg",

              '11☘':r"C:\Users\shahi\Pictures\python\_card\_image\JC.jpeg",

              '12☘':r"C:\Users\shahi\Pictures\python\_card\_image\QC.jpeg",

              '13☘':r"C:\Users\shahi\Pictures\python\_card\_image\KC.jpeg",

              "1 ":r"C:\Users\shahi\Pictures\python\_card\_image\AS.jpeg",

              "2 ":r"C:\Users\shahi\Pictures\python\_card\_image\2S.jpeg",

              "3 ":r"C:\Users\shahi\Pictures\python\_card\_image\3S.jpeg",

              '4 ':r"C:\Users\shahi\Pictures\python\_card\_image\4S.jpeg",

              '5 ':r"C:\Users\shahi\Pictures\python\_card\_image\5S.jpeg",

              '6 ':r"C:\Users\shahi\Pictures\python\_card\_image\6S.jpeg",

              '7 ':r"C:\Users\shahi\Pictures\python\_card\_image\7S.jpeg",

              '8 ':r"C:\Users\shahi\Pictures\python\_card\_image\8S.jpeg",

              '9 ':r"C:\Users\shahi\Pictures\python\_card\_image\9S.jpeg",

              '10 ':r"C:\Users\shahi\Pictures\python\_card\_image\10S.jpeg",

              '11 ':r"C:\Users\shahi\Pictures\python\_card\_image\JS.jpeg",

              '12 ':r"C:\Users\shahi\Pictures\python\_card\_image\QS.jpeg",

              '13 ':r"C:\Users\shahi\Pictures\python\_card\_image\KS.jpeg",

              }

classInputBox:

    def\_\_init\_\_(self, x, y, w, h, text=''):

        FONT = pygame.font.Font(None, 40)

        self.rect = pygame.Rect(x, y, w, h)

        self.color = COLOR\_INACTIVE

        self.text = text

        self.txt\_surface = FONT.render(text, True, self.color)

        self.active = False

    defhandle\_event(self, event):

        ifevent.type == pygame.MOUSEBUTTONDOWN:

            # If the user clicked on the input\_box rect.

            ifself.rect.collidepoint(event.pos):

                # Toggle the active variable.

                self.active = notself.active

            else:

                self.active = False

            # Change the current color of the input box.

            self.color = COLOR\_ACTIVEifself.activeelseCOLOR\_INACTIVE

        ifevent.type == pygame.KEYDOWN:

            f= pygame.font.Font(None, 40)

            ifself.active:

                ifevent.key == pygame.K\_RETURN:

                    print(self.text)

                    self.text = ''

                elifevent.key == pygame.K\_BACKSPACE:

                    self.text = self.text[0:len(self.text)-1]

                    # self.txt\_surface=f.render(self.text,True,'white')

                    # self.txt\_surface=f.render(self.text,True,self.color)

                else:

                    self.text += event.unicode

                # Re-render the text.

                self.txt\_surface = f.render(self.text,False, self.color)

    # def update(self):

        # Resize the box if the text is too long.

        # width = max(410, 10)

        # self.rect.w = width

    defdraw(self, screen):

        # Blit the text.

        screen.blit(self.txt\_surface, (self.rect.x+5, self.rect.y+5))

        # Blit the rect.

        pygame.draw.rect(screen, self.color, self.rect, 2)

# card shuffle method

defshufflePack(l):

    newPack=[]

    total=len(l)

    foriinrange(0,total):

        n=random.randint(0,len(l)-1)

        newPack.append(l[n])

        l.pop(n)

    returnnewPack

defhow\_to\_play():

    screen.fill('orange')

    how\_to\_play\_image=pygame.image.load(r"C:\Users\shahi\Pictures\python\_card\_image\how\_to\_play.jpeg")

    screen.blit(how\_to\_play\_image,(0,0))

    pygame.display.update()

    run=True

    whilerun:

        pygame.display.flip()

        mouse\_pos=pygame.mouse.get\_pos()

        foreinpygame.event.get():

            ife.type==pygame.QUIT :

                run=False

            ifmouse\_pos[0] inrange(464,652) andmouse\_pos[1] inrange(612,667):

                ifpygame.mouse.get\_pressed()[0]==1:

                    run=False

defsend\_sms(no):

    globalplayer1\_name

    globalplayer2\_name

    globalplayer1\_no

    globalplayer2\_no

    account\_sid="ACad9c0d950e06b746a9fb432d062a5a65"

    auth\_tokan="4e7b8c1fcdc6c1950a8635ff1365e975"

    client=Client(account\_sid,auth\_tokan)

    ifno==1andplayer1\_no!='':

        client.messages.create(body='Congratulations '+player1\_name+",\n"+"You win the card game \n\n\n\n\n from team,\n Python Card Game Project",from\_='+16078003847',to="+91"+player1\_no)

    elifno==2andplayer2\_no!='':

        client.messages.create(body='Congratulations '+player2\_name+",\n"+"You win the card game \n\n\n\n\n from team,\n Python Card Game Project",from\_='+16078003847',to="+91"+player2\_no)

defuser\_data():

    globalplayer1\_name

    globalplayer2\_name

    globalplayer1\_no

    globalplayer2\_no

    screen.fill('Black')

    clock = pygame.time.Clock()

    user\_data\_bg=pygame.image.load(r'C:\Users\shahi\Pictures\python\_card\_image\user\_data.jpeg')

    screen.blit(user\_data\_bg,(0,0))

    # print(player1\_name,player2\_name,player1\_no,player2\_no)

    p1\_name = InputBox(90,210, 410, 50)

    p2\_name = InputBox(690, 210, 410, 50)

    p1\_no = InputBox(90, 400, 410,50)

    p2\_no = InputBox(690,400 ,410,50)

    input\_boxes = [p1\_name,p1\_no,p2\_name,p2\_no]

    done=False

    whilenotdone:

        m\_pos=pygame.mouse.get\_pos()

        foreventinpygame.event.get():

            ifevent.type == pygame.QUIT:

                done = True

                return'back'

            forboxininput\_boxes:

                box.handle\_event(event)

        # for box in input\_boxes:

        #     box.update()

            ifm\_pos[0] inrange(485,685) andm\_pos[1] inrange(573,645):

                ifpygame.mouse.get\_pressed()[0]==1:

                    player1\_name=p1\_name.text

                    player2\_name=p2\_name.text

                    player1\_no=p1\_no.text

                    player2\_no=p2\_no.text

                    print(player1\_name,player2\_name,player1\_no,player2\_no)

                    done=True

        forboxininput\_boxes:

            box.draw(screen)

        pygame.display.flip()

        clock.tick(30)

defwelcome\_screen():

    globalplayer1\_name

    globalplayer2\_name

    globalplayer1\_no

    globalplayer2\_no

    screen.fill('red')

    welcome\_image=pygame.image.load(r"C:\Users\shahi\Pictures\python\_card\_image\welcome\_bg.jpeg")

    screen.blit(welcome\_image,(0,20))

    pygame.display.set\_caption('welcome')

    pygame.display.update()

    # top 360  left 490  w 230  h 65

    # rec1=pygame.draw.rect(screen,'black',pygame.rect.Rect(490,360,230,64))

    # how to play (490,440,230,65)

    # quit

    # rec2=pygame.draw.rect(screen,'blue',pygame.rect.Rect(490,530,230,65))

    pygame.display.flip()

    ru=True

    whileru:

        pygame.display.flip()

        mouse\_pos=pygame.mouse.get\_pos()

        foreinpygame.event.get():

            ife.type==pygame.QUIT:

                pygame.quit()

                ru=False

            ifmouse\_pos[0] inrange(490,720) andmouse\_pos[1] inrange(360,425):

                ifpygame.mouse.get\_pressed()[0]==1 :

                    ru=False

                    player1\_name='player1'

                    player2\_name='player2'

                    player1\_no=''

                    player2\_no=''

                    return'play'

            ifmouse\_pos[0] inrange(490,720) andmouse\_pos[1] inrange(530,595):

                ifpygame.mouse.get\_pressed()[0]==1 :

                    ru=False

                    return'quit'

            ifmouse\_pos[0] inrange(490,720) andmouse\_pos[1] inrange(440,505):

                ifpygame.mouse.get\_pressed()[0]==1 :

                    return'how\_to\_play'

# def take\_name():

#     take\_name\_screen=pygame.surface.Surface((400,400))

#     take\_name\_screen.fill('orange')

#     screen.blit(take\_name\_screen,(400,150))

#     pygame.display.update()

# set images on GUI

defbind\_images\_User1(l1,screen,image\_string):

    card1=pygame.image.load(image\_string.get(l1[0]))

    card2=pygame.image.load(image\_string.get(l1[1]))

    card3=pygame.image.load(image\_string.get(l1[2]))

    card4=pygame.image.load(image\_string.get(l1[3]))

    screen.blit(card1,(70,470))

    screen.blit(card2,(190,470))

    screen.blit(card3,(310,470))

    screen.blit(card4,(430,470))

defbind\_images\_User2(l1,screen,image\_string):

        card5=pygame.image.load(image\_string.get(l1[0]))

        card6=pygame.image.load(image\_string.get(l1[1]))

        card7=pygame.image.load(image\_string.get(l1[2]))

        card8=pygame.image.load(image\_string.get(l1[3]))

        screen.blit(card5,(670,470))

        screen.blit(card6,(790,470))

        screen.blit(card7,(910,470))

        screen.blit(card8,(1030,470))

# random card for selection

defrandom\_card(pack):

    n=random.randint(0,len(pack)-1)

    main\_card=pygame.image.load(image\_string.get(pack[n]))

    screen.blit(main\_card,(550,100))

    returnn

defchange\_card(num,pack,user,n):

    num=num-1

    pack.append(user.pop(num))

    user.insert(num,pack[n])

    pack.pop(n)

defdisplay\_winner(user,w):

    globalplayer1\_name

    globalplayer2\_name

    globalplayer1\_no

    globalplayer2\_no

    print(player1\_name,player2\_name,player1\_no,player2\_no)

    try:

        if (w == 1):

            send\_sms(1)

        if (w== 2):

            send\_sms(2)

    exceptException:

        pass

    try:

        screen.fill('Blue')

        winner\_bg=pygame.image.load(r"C:\Users\shahi\Pictures\python\_card\_image\winner\_bg.jpeg")

        screen.blit(winner\_bg,(0,0))

        pygame.draw.rect(screen,'green',pygame.rect.Rect(200,250,150,30))

        text\_1=pygame.font.SysFont('carbel',50)

        t=text\_1.render("Winner :- ",False,'Black')

        screen.blit(t,(200,250))

        ifw==1:

            if( player1\_name!=''):

                text\_winner\_name=pygame.font.SysFont('carbel',50)

                text=text\_winner\_name.render(player1\_name,False,'Black')

                screen.blit(text,(250,300))

            else:

                text\_winner\_name=pygame.font.SysFont('carbel',50)

                text=text\_winner\_name.render(user,False,'Black')

                screen.blit(text,(250,300))

        ifw==2:

            if (player2\_name!=''):

                text\_winner\_name=pygame.font.SysFont('carbel',50)

                text=text\_winner\_name.render(player2\_name,False,'Black')

                screen.blit(text,(250,300))

            else:

                text\_winner\_name=pygame.font.SysFont('carbel',50)

                text=text\_winner\_name.render(user,False,'Black')

                screen.blit(text,(250,300))

    exceptException:

        pass

    pygame.display.flip()

    r=True

    whiler:

        foreventinpygame.event.get():

            ifevent.type==pygame.QUIT:

                r=False

defmain\_game():

    background\_colour ='green'

    screen.fill(background\_colour)

    pygame.display.set\_caption('Card game')

    pygame.display.flip()

    s

# card pack GUI

    pygame.draw.rect(screen,'red',pygame.Rect(550,100,104,160))

    card\_pack\_image=pygame.image.load(r"C:\Users\shahi\Pictures\python\_card\_image\cardPack.jpeg")

    screen.blit(card\_pack\_image,(545,98))

# Red color GUI

    pygame.draw.rect(screen,'red',pygame.Rect(50,450,500,200))

    pygame.draw.rect(screen,'red',pygame.Rect(650,450,500,200))

# button for skip

    pygame.draw.rect(screen,'Blue',pygame.Rect(550,400,90,40))

    text\_skip=pygame.font.SysFont('carbel',50)

    text=text\_skip.render('SKIP',False,'Black')

    screen.blit(text,(555,400))

#   indicaterects

    indicate\_image=pygame.image.load(r"C:\Users\shahi\Pictures\python\_card\_image\indicate\_user.jpeg")

    screen.blit(indicate\_image,(500,420))

    hide\_indicate\_image=pygame.image.load(r"C:\Users\shahi\Pictures\python\_card\_image\hide\_indicate.jpeg")

    screen.blit(hide\_indicate\_image,(700,420))

# string and image relation

    pack=["1🔸","2🔸","3🔸",'4🔸','5🔸','6🔸','7🔸','8🔸','9🔸','10🔸','11🔸','12🔸'

          ,'13🔸',"1❤","2❤","3❤",'4❤','5❤','6❤','7❤','8❤','9❤','10❤','11❤','12❤','13❤',

          "1☘","2☘","3☘",'4☘','5☘','6☘','7☘','8☘','9☘','10☘','11☘','12☘','13☘',

          "1 ","2 ","3 ",'4 ','5 ','6 ','7 ','8 ' ,'9 ','10 ','11 ','12 ','13 ']

    user1=[]

    user2=[]

    foriinrange(4):

        n=random.randint(0,len(pack)-1)

        user1.append(pack[n])

        pack.pop(n)

        n=random.randint(0,len(pack)-1)

        user2.append(pack[n])

        pack.pop(n)

    bind\_images\_User1(user1,screen,image\_string)

    bind\_images\_User2(user2,screen,image\_string)

    n=random\_card(pack)

    ready\_user1=True

    ready\_user2=False

    w=0

    run=True

    whilerun:

        main\_mouse\_pos=pygame.mouse.get\_pos()

        pygame.display.flip()

        foreventinpygame.event.get():

            ifevent.type==pygame.QUIT:

                run=False

            ifmain\_mouse\_pos[0] inrange(50,550) andmain\_mouse\_pos[1] inrange(450,650):

                print("user1 ")

                ifmain\_mouse\_pos[0] inrange(70,174) andmain\_mouse\_pos[1] inrange(470,630):

                    ifpygame.mouse.get\_pressed()[0]==1  andready\_user1==True:

                        print("card1")

                        change\_card(1,pack,user1,n)

                        bind\_images\_User1(user1,screen,image\_string)

                        n=random\_card(pack)

                        ready\_user1=False

                        ready\_user2=True

                ifmain\_mouse\_pos[0] inrange(190,294) andmain\_mouse\_pos[1] inrange(470,630):

                    ifpygame.mouse.get\_pressed()[0]==1andready\_user1==True:

                        print("card2")

                        change\_card(2,pack,user1,n)

                        bind\_images\_User1(user1,screen,image\_string)

                        n=random\_card(pack)

                        ready\_user1=False

                        ready\_user2=True

                ifmain\_mouse\_pos[0] inrange(310,414) andmain\_mouse\_pos[1] inrange(470,630):

                    ifpygame.mouse.get\_pressed()[0]==1andready\_user1==True:

                        change\_card(3,pack,user1,n)

                        print("card1")

                        bind\_images\_User1(user1,screen,image\_string)

                        n=random\_card(pack)

                        ready\_user1=False

                        ready\_user2=True

                ifmain\_mouse\_pos[0] inrange(430,534) andmain\_mouse\_pos[1] inrange(470,630):

                    ifpygame.mouse.get\_pressed()[0]==1andready\_user1==True:

                        print("card1")

                        change\_card(4,pack,user1,n)

                        bind\_images\_User1(user1,screen,image\_string)

                        n=random\_card(pack)

                        ready\_user1=False

                        ready\_user2=True

            ifmain\_mouse\_pos[0] inrange(650,1150) andmain\_mouse\_pos[1] inrange(450,650) :

                print('user2')

                ifmain\_mouse\_pos[0] inrange(670,774) andmain\_mouse\_pos[1] inrange(470,630):

                    ifpygame.mouse.get\_pressed()[0]==1andready\_user2==True:

                        print("card1")

                        change\_card(1,pack,user2,n)

                        bind\_images\_User2(user2,screen,image\_string)

                        n=random\_card(pack)

                        ready\_user1=True

                        ready\_user2=False

                ifmain\_mouse\_pos[0] inrange(790,894) andmain\_mouse\_pos[1] inrange(470,630):

                    ifpygame.mouse.get\_pressed()[0]==1andready\_user2==True:

                        print("card2")

                        change\_card(2,pack,user2,n)

                        bind\_images\_User2(user2,screen,image\_string)

                        n=random\_card(pack)

                        ready\_user1=True

                        ready\_user2=False

                ifmain\_mouse\_pos[0] inrange(910,1014) andmain\_mouse\_pos[1] inrange(470,630):

                    ifpygame.mouse.get\_pressed()[0]==1andready\_user2==True:

                        print("card3")

                        change\_card(3,pack,user2,n)

                        bind\_images\_User2(user2,screen,image\_string)

                        n=random\_card(pack)

                        ready\_user1=True

                        ready\_user2=False

                ifmain\_mouse\_pos[0] inrange(1030,1134) andmain\_mouse\_pos[1] inrange(470,630):

                    ifpygame.mouse.get\_pressed()[0]==1andready\_user2==True:

                        print("card4")

                        change\_card(4,pack,user2,n)

                        bind\_images\_User2(user2,screen,image\_string)

                        n=random\_card(pack)

                        ready\_user1=True

                        ready\_user2=False

            ifmain\_mouse\_pos[0] inrange(550,640) andmain\_mouse\_pos[1] inrange(400,440):

                # skip

                ifpygame.mouse.get\_pressed()[0]==1:

                    ifready\_user1==True:

                        ready\_user1=False

                        ready\_user2=True

                    else:

                        ready\_user2=False

                        ready\_user1=True

                    sleep(1)

                    n=random\_card(pack)

            ifready\_user1==True:

                screen.blit(indicate\_image,(500,420))

                screen.blit(hide\_indicate\_image,(700,420))

            ifready\_user2==True:

                screen.blit(indicate\_image,(700,420))

                screen.blit(hide\_indicate\_image,(500,420))

            if(user1[0][0]==user1[1][0]==user1[2][0]==user1[3][0]):

                if(user1[0][1] in ['0','1','2','3'] oruser1[1][1] in ['0','1','2','3'] oruser1[2][1] in ['0','1','2','3'] oruser1[3][1] in ['0','1','2','3']):

                    if(user1[0][1]==user1[1][1]==user1[2][1]==user1[3][1]):

                        run=False

                        w=1

                else:

                    run=False

                    w=1

            if(user2[0][0]==user2[1][0]==user2[2][0]==user2[3][0]):

                if(user2[0][1] in ['0','1','2','3'] oruser2[1][1] in ['0','1','2','3'] oruser2[2][1] in ['0','1','2','3'] or  user2[3][1] in ['0','1','2','3']):

                    if (user2[0][1]==user2[1][1]==user2[2][1]==user2[3][1]):

                        run=False

                        w=2

                else:

                    run=False

                    w=2

    ifw==1:

        returndisplay\_winner('Player 1',1)

    elifw==2:

        returndisplay\_winner('Player 2',2)

background\_colour ='green'

screen.fill(background\_colour)

pygame.display.set\_caption('Card game')

pygame.display.flip()

#  for Running game

s=True

running = True

whilerunning:

    try:

        foreventinpygame.event.get():

            ifevent.type==pygame.QUIT:

                s=False

                running=False

            ifs==True:

                y=welcome\_screen()

                s=False

            if(y=='play'):

                next=user\_data()

                print(next)

                if(next!='back'):

                    main\_game()

                s=True

            ify=='how\_to\_play':

                how\_to\_play()

                s=True

            ify=='quit':

                running=False

                s=False

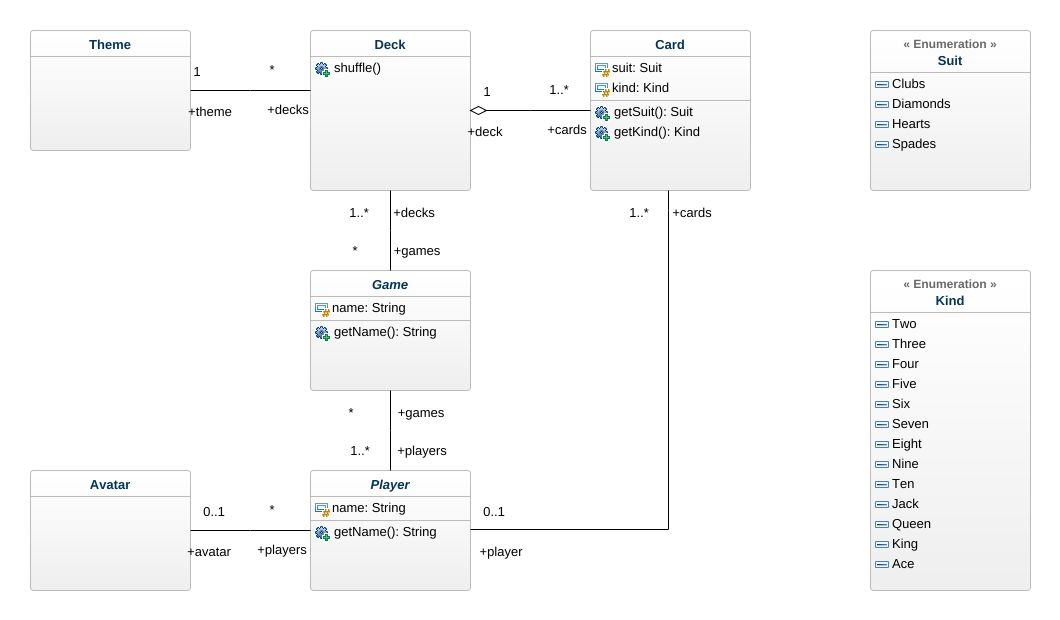
    exceptException:

        pass

Diagram:

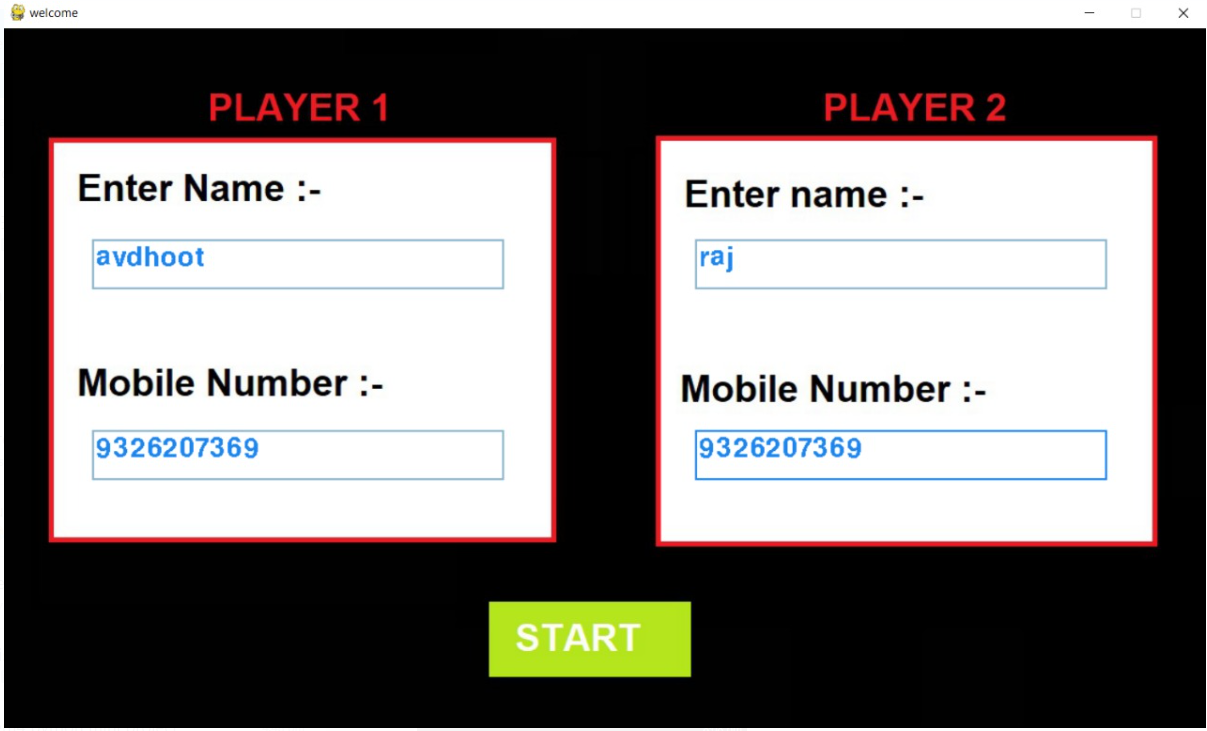
UML :

1 CLASS DIAGRAM:

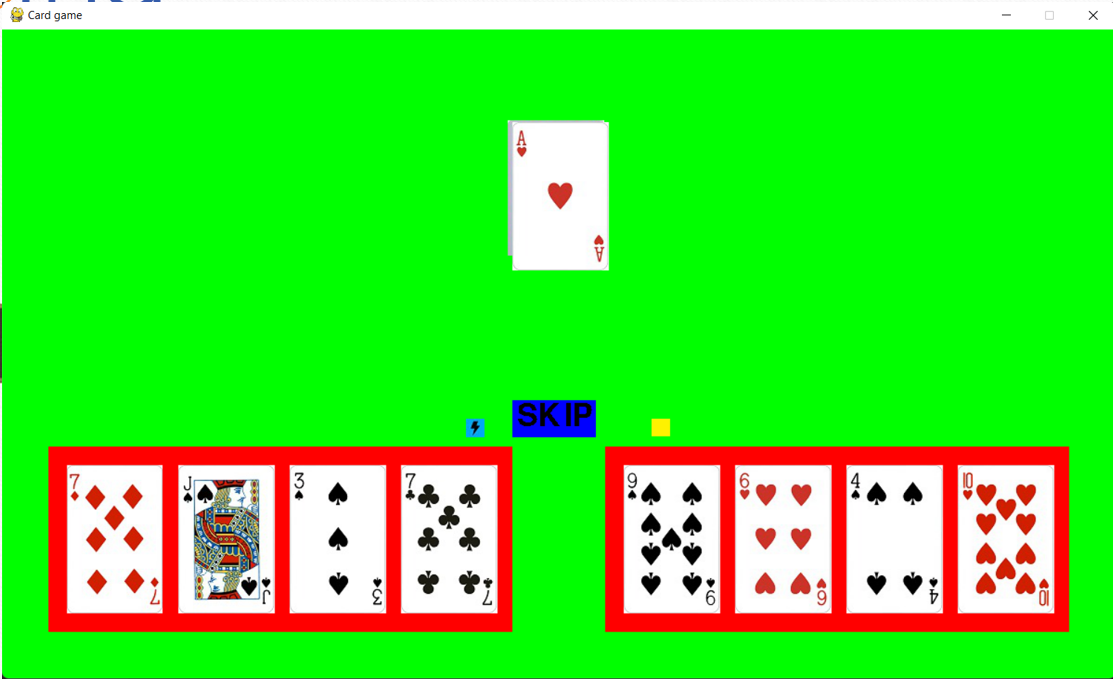


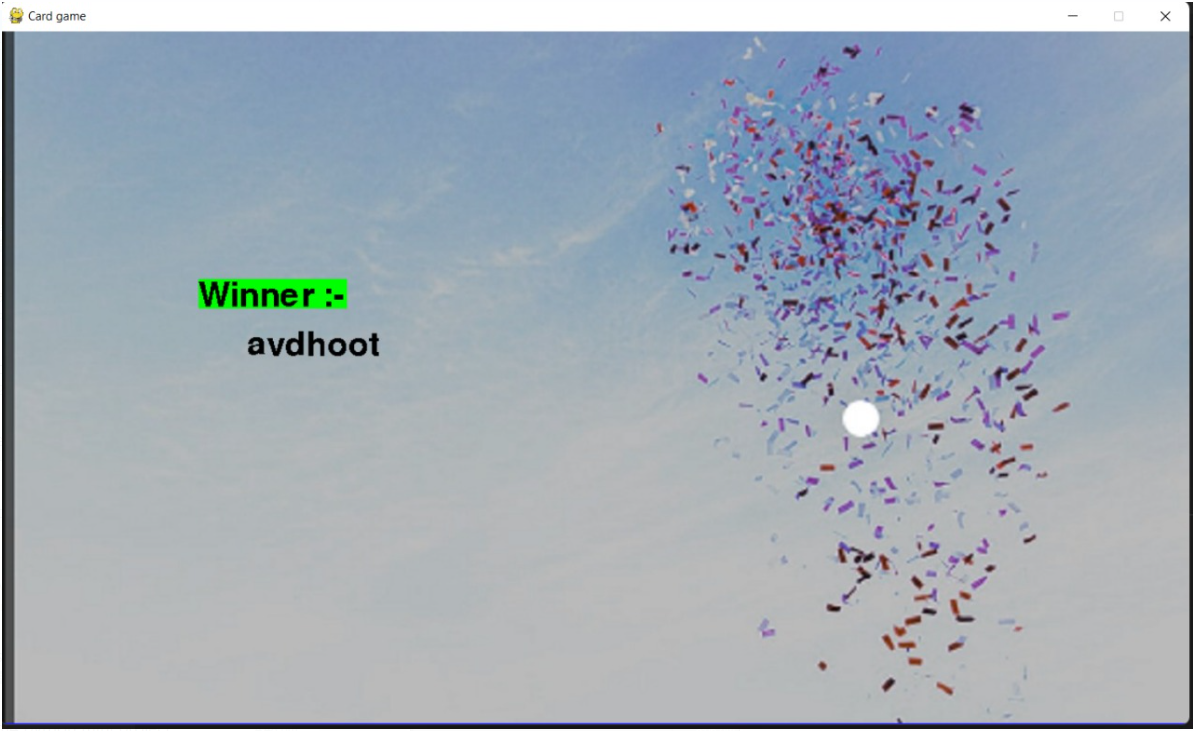
* Screenshots Of Game:-











* **Conclusion and Future work.**

We tried to implement this game in a straightforward, testable, and modular form. This made debugging simple. It also means that it is easier for another person to understand the code and its purpose.

* **Reference :-**

1. [https://gist.github.com](https://gist.github.com/csaez/5655129)

2. <https://github.com/gdwangh/coursera-introPython/blob/master/week5/Mini-project%205%20-%20Memory.py>

3. https://slideplayer.com/slide/222263/