Chat System: Application with Graphic User Interface and Online Game

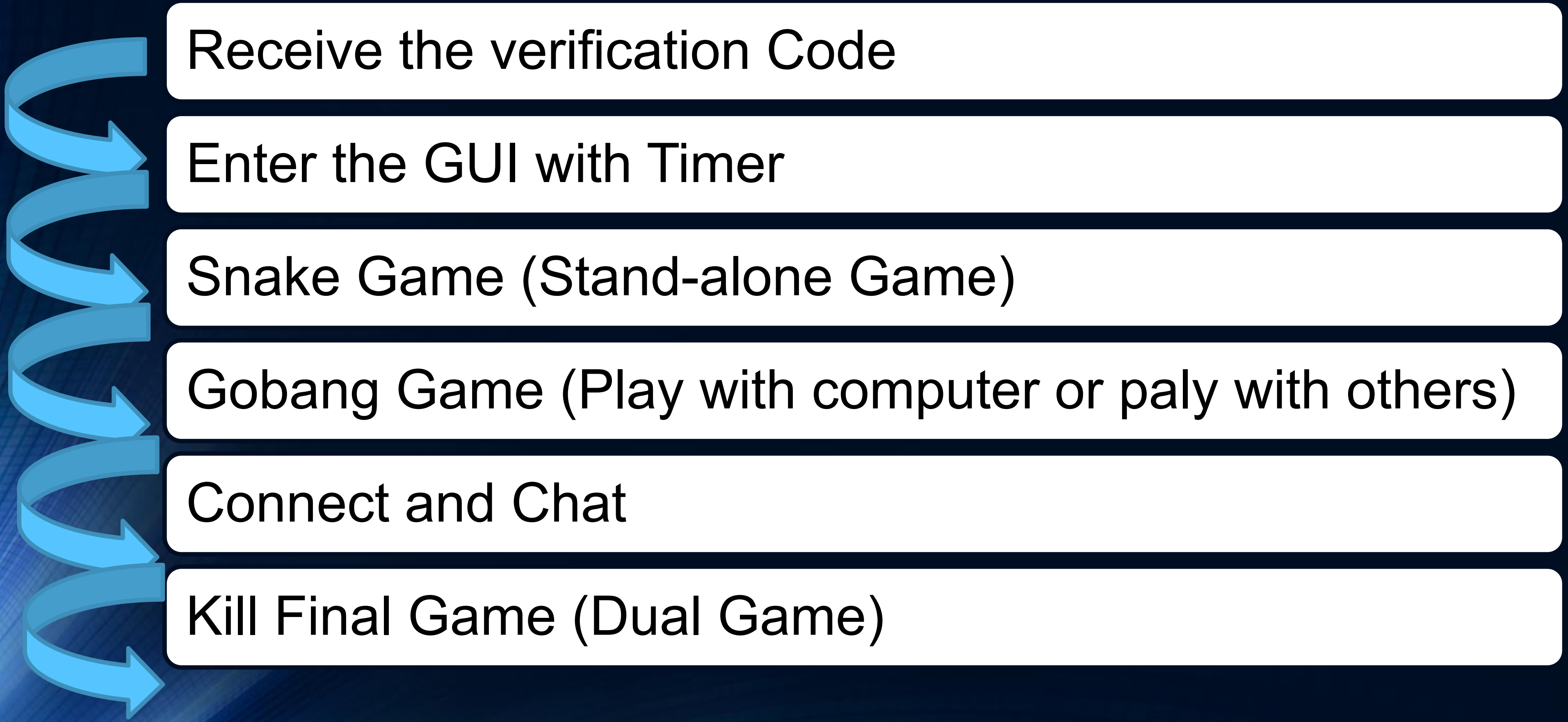
ICS Final Project

Qianyu Zhuc qz1086
Sihan Liu sl6964
Xiaoyan Tang xt534

Presentation Agenda

- Project Structural Introduction
- Verification Code Explanation and Demo
- Graphic User Interface Explanation and Demo(GUI)
- Chat System Explanation and Demo
- Snake Game Explanation and Demo
- “Kill Final” Game Explanation and Demo
- Gobang Game Explanation and Demo

Project Structural Demonstration



Receive the verification Code

Enter the GUI with Timer

Snake Game (Stand-alone Game)

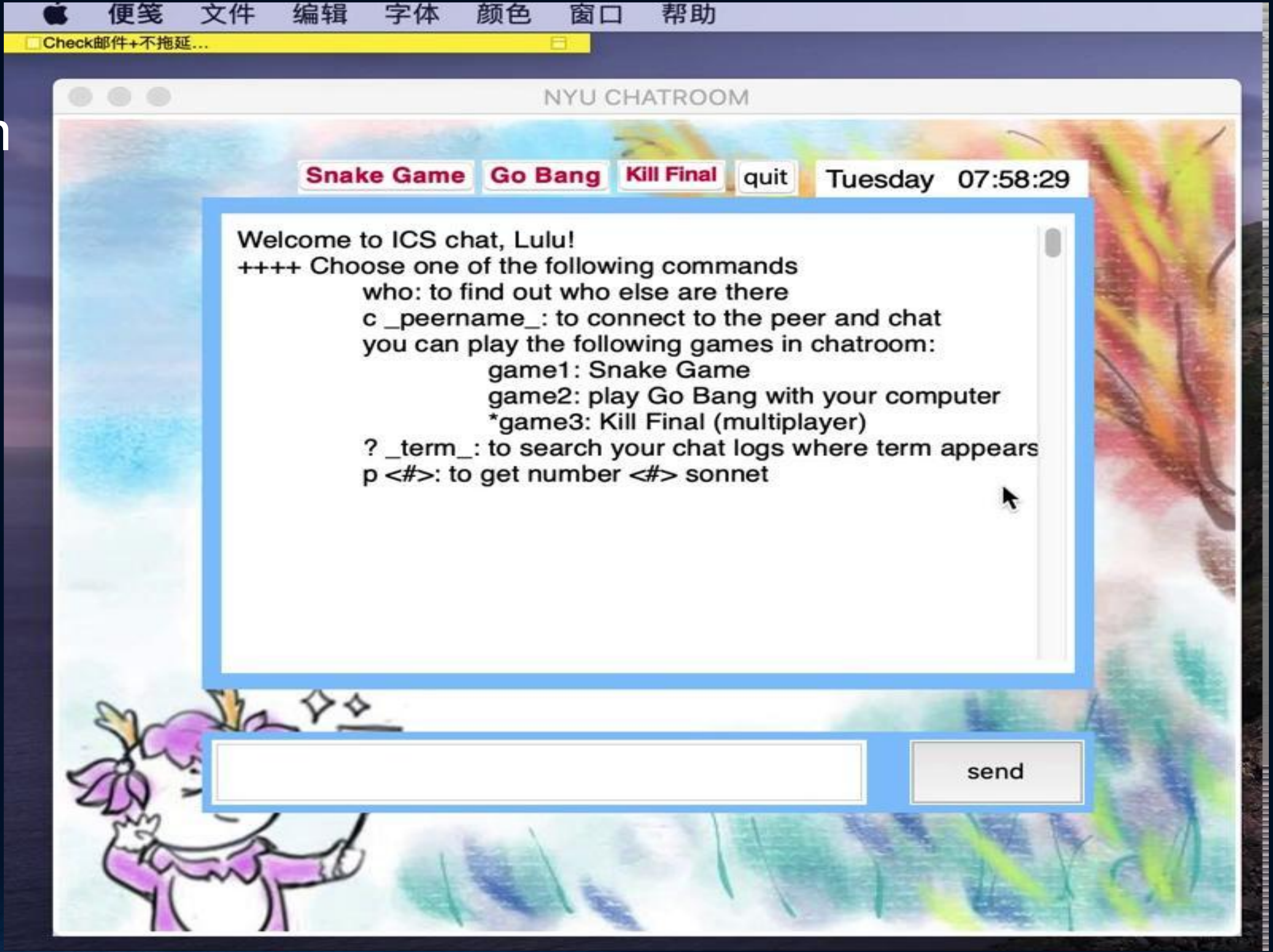
Gobang Game (Play with computer or paly with others)

Connect and Chat

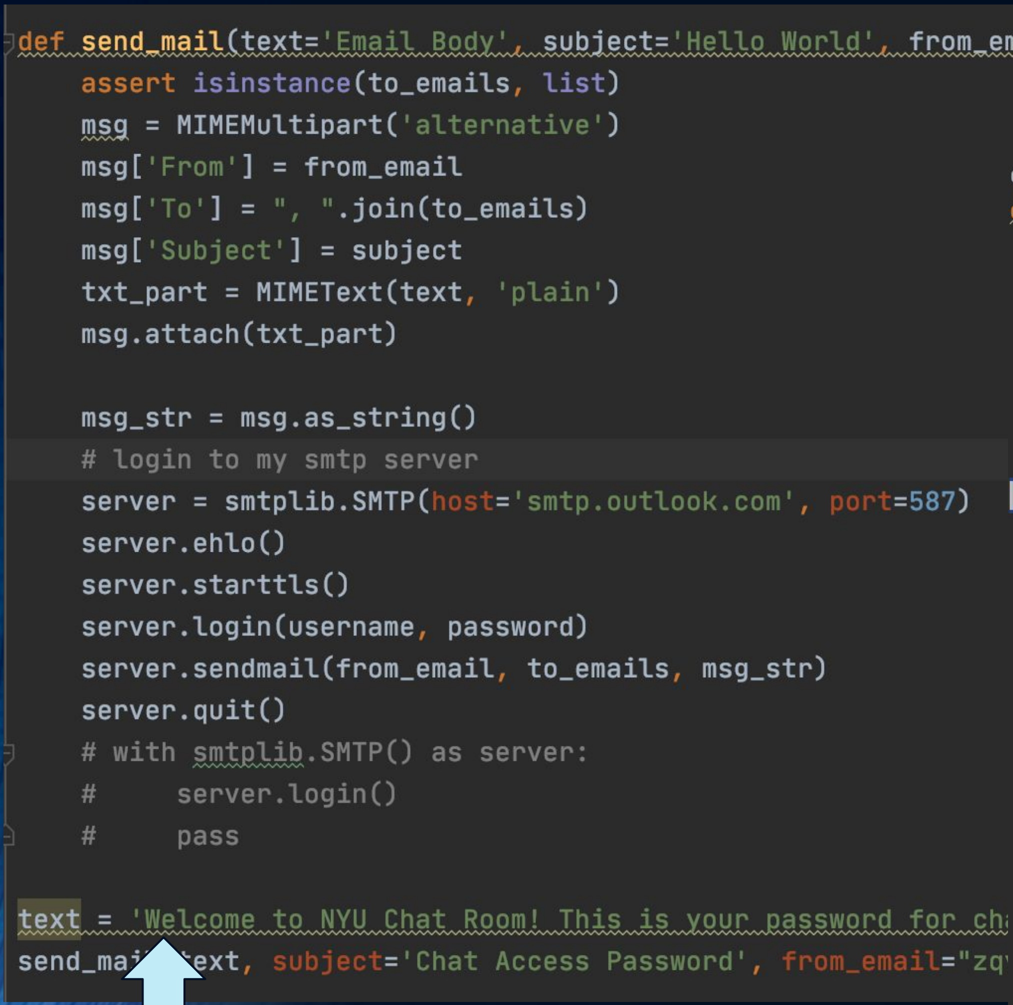
Kill Final Game (Dual Game)

Verification Code

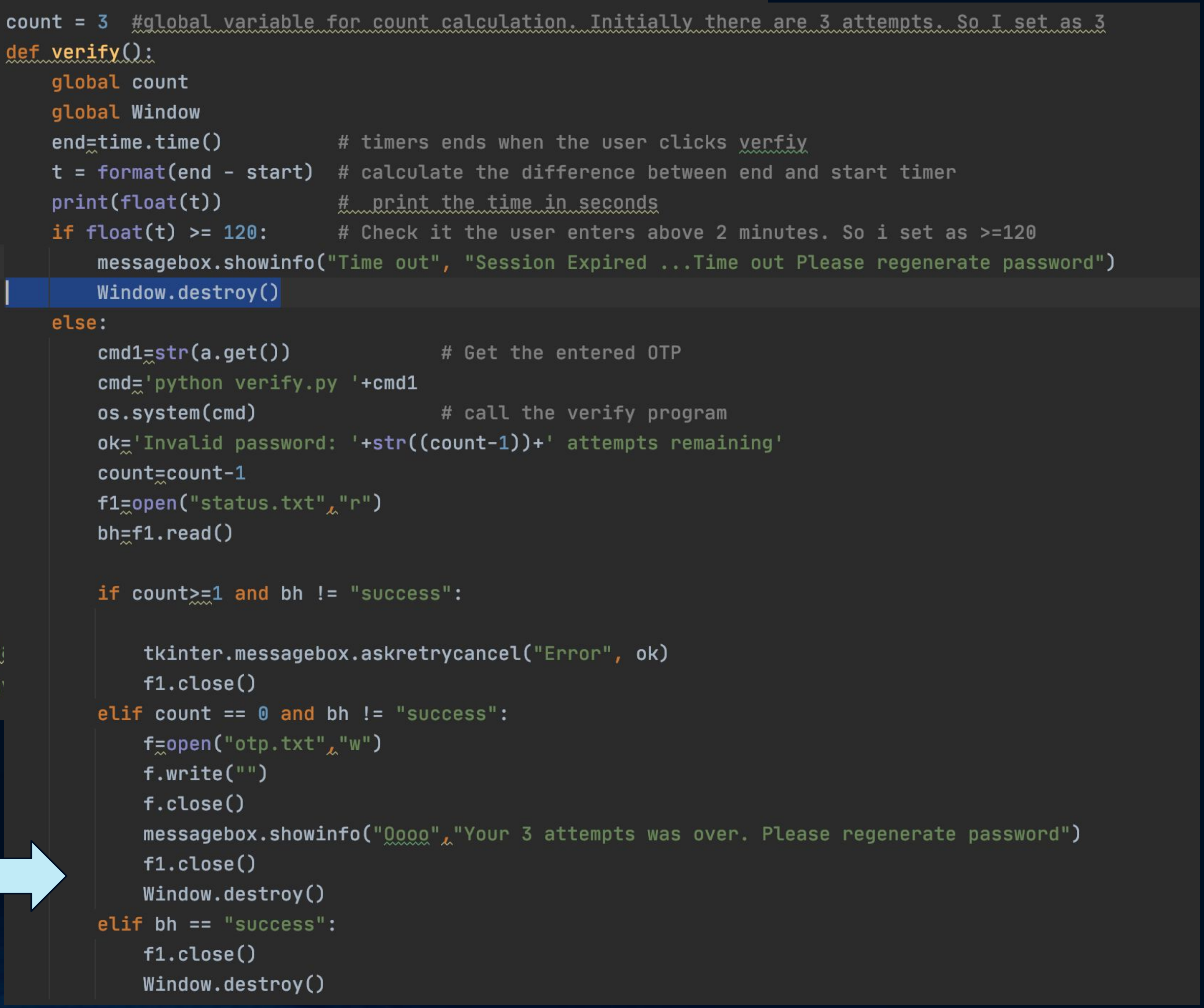
- create windows, buttons
- create email using python
- sending random password & verify using .txt
- time & number restrictions



```
Welcome to ICS chat, Lulu!
++++ Choose one of the following commands
        who: to find out who else are there
        c _peername_: to connect to the peer and chat
        you can play the following games in chatroom:
                game1: Snake Game
                game2: play Go Bang with your computer
                *game3: Kill Final (multiplayer)
        ? _term_: to search your chat logs where term appears
        p <#>: to get number <#> sonnet
```



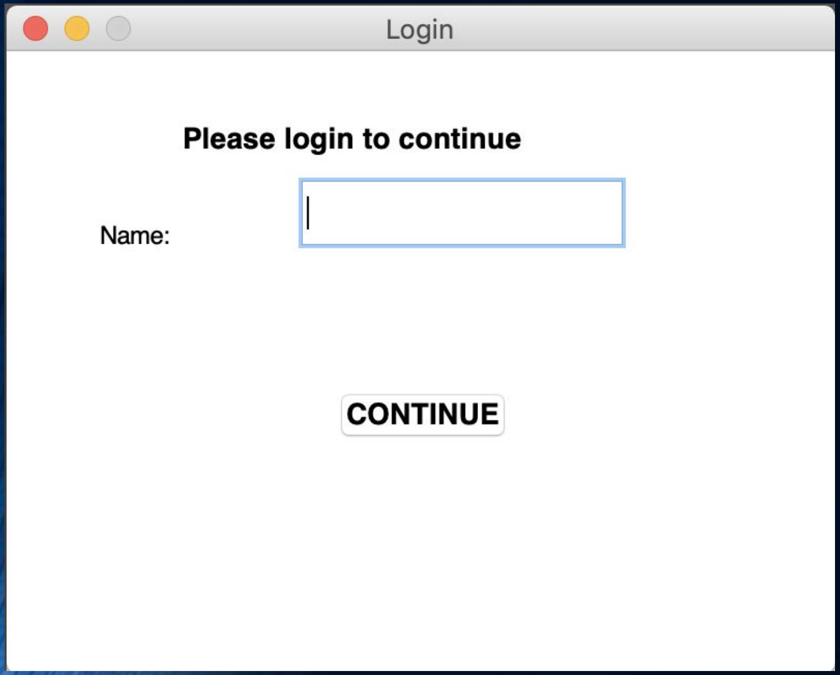
```
def send_mail(text='Email Body', subject='Hello World', from_email='Chat Room <zqy0927@outlook.com>', to_emails=None):
    assert isinstance(to_emails, list)
    msg = MIMEMultipart('alternative')
    msg['From'] = from_email
    msg['To'] = ", ".join(to_emails)
    msg['Subject'] = subject
    txt_part = MIMEText(text, 'plain')
    msg.attach(txt_part)

    msg_str = msg.as_string()
    # login to my smtp server
    server = smtplib.SMTP(host='smtp.outlook.com', port=587)
    server.ehlo()
    server.starttls()
    server.login(username, password)
    server.sendmail(from_email, to_emails, msg_str)
    server.quit()
    # with smtplib.SMTP() as server:
    #     server.login()
    #     pass

text = 'Welcome to NYU Chat Room! This is your password for ch
send_mai  ext, subject='Chat Access Password', from_email="zq
```

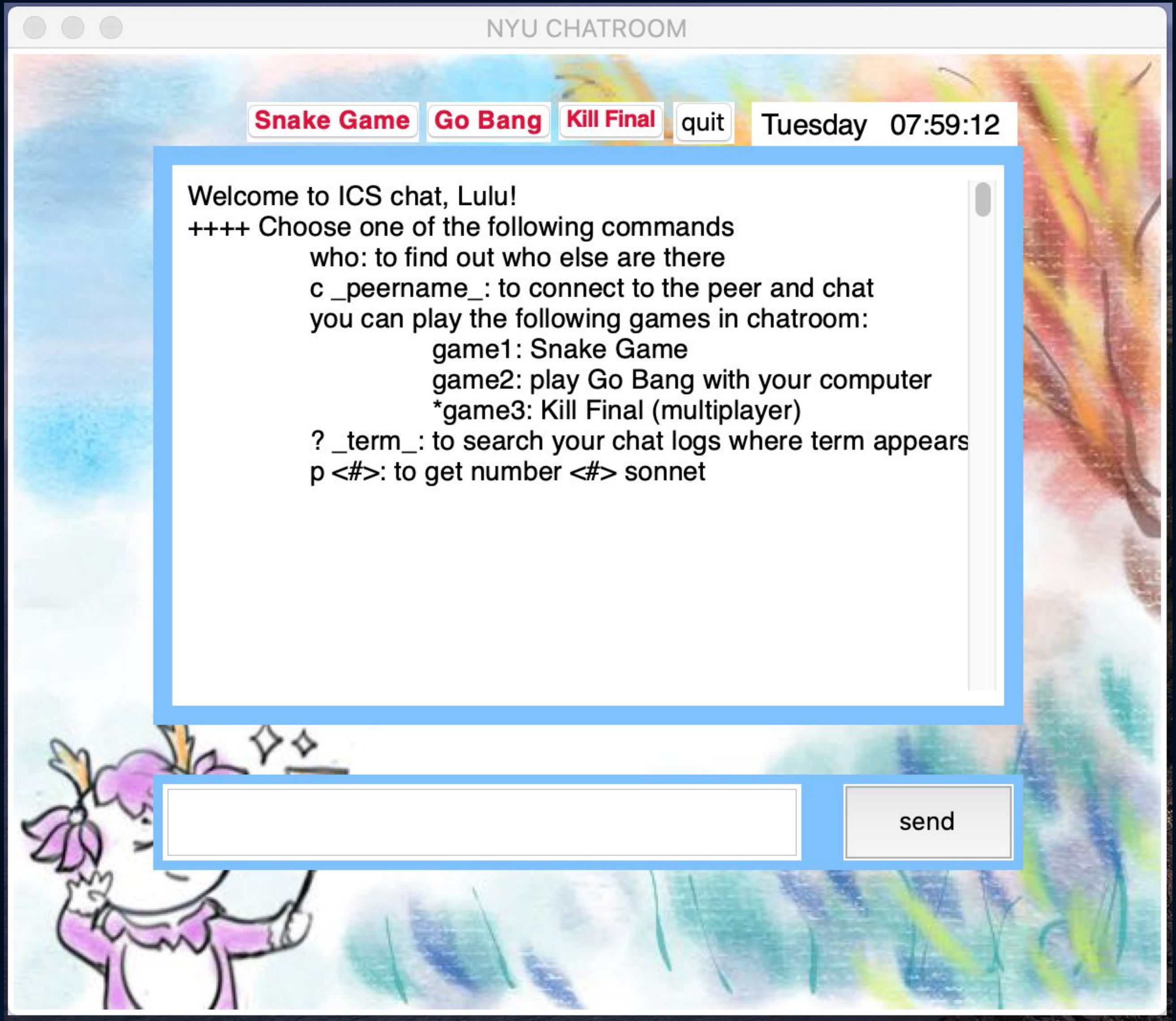


```
count = 3  #global variable for count calculation. Initially there are 3 attempts. So I set as 3
def verify():
    global count
    global Window
    end=time.time()            # timers ends when the user clicks verfiy
    t = format(end - start)    # calculate the difference between end and start timer
    print(float(t))            #  print the time in seconds
    if float(t) >= 120:        # Check it the user enters above 2 minutes. So i set as >=120
        messagebox.showinfo("Time out", "Session Expired ...Time out Please regenerate password")
        Window.destroy()
    else:
        cmd1=str(a.get())               # Get the entered OTP
        cmd='python verify.py '+cmd1
        os.system(cmd)                  # call the verify program
        ok='Invalid password: '+str((count-1))+' attempts remaining'
        count=count-1
        f1=open("status.txt","r")
        bh=f1.read()

        if count>=1 and bh != "success":

            tkinter.messagebox.askretrycancel("Error", ok)
            f1.close()
        elif count == 0 and bh != "success":
            f=open("otp.txt","w")
            f.write("")
            f.close()
            messagebox.showinfo("Oooo","Your 3 attempts was over. Please regenerate password")
            f1.close()
            Window.destroy()
        elif bh == "success":
            f1.close()
            Window.destroy()
```

python email
verify

GUI



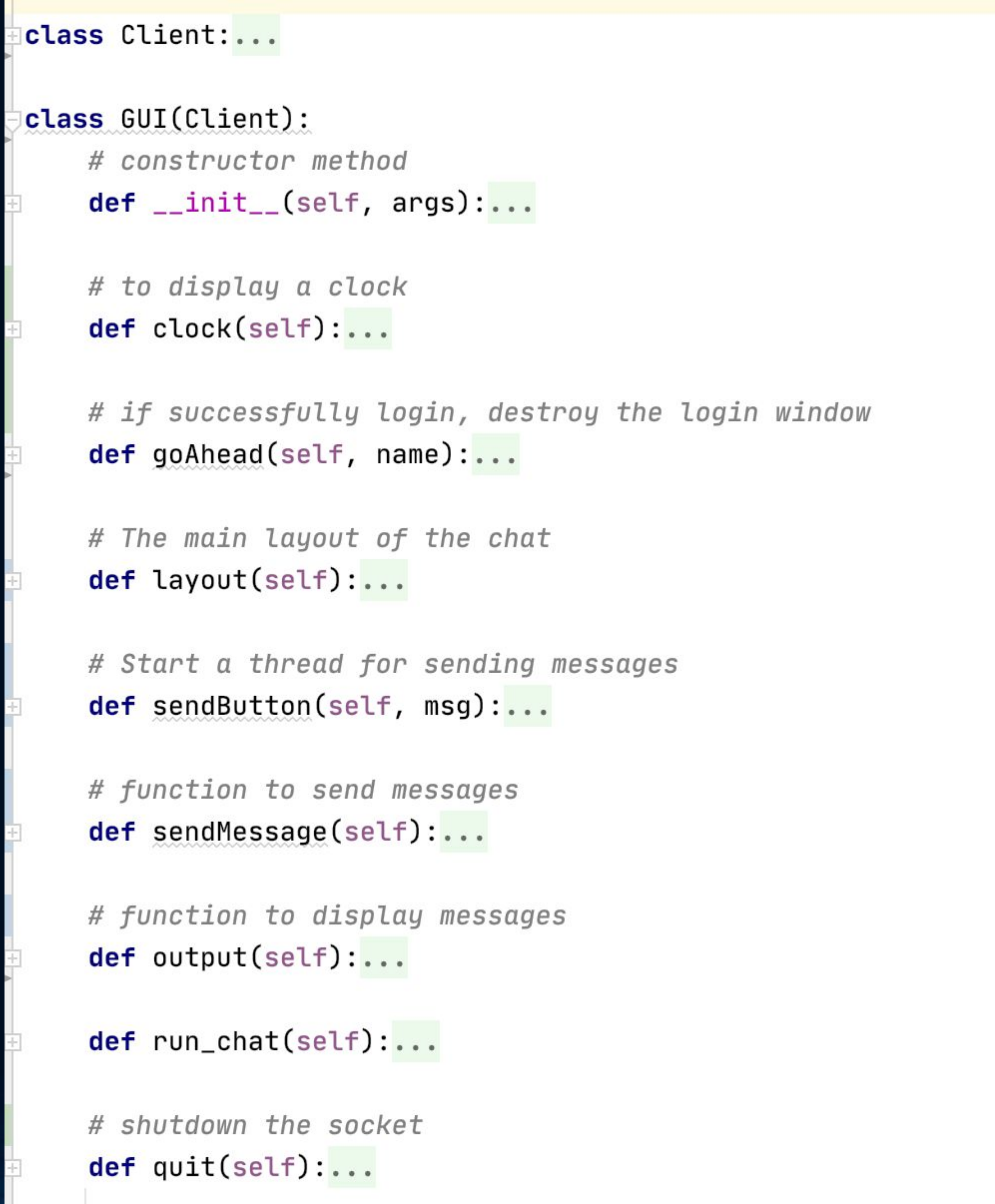


```
Welcome to ICS chat, Lulu!
++++ Choose one of the following commands
          who: to find out who else are there
          c _peername_: to connect to the peer and chat
          you can play the following games in chatroom:
                    game1: Snake Game
                    game2: play Go Bang with your computer
                    *game3: Kill Final (multiplayer)
          ? _term_: to search your chat logs where term appears
          p <#>: to get number <#> sonnet
```

GUI

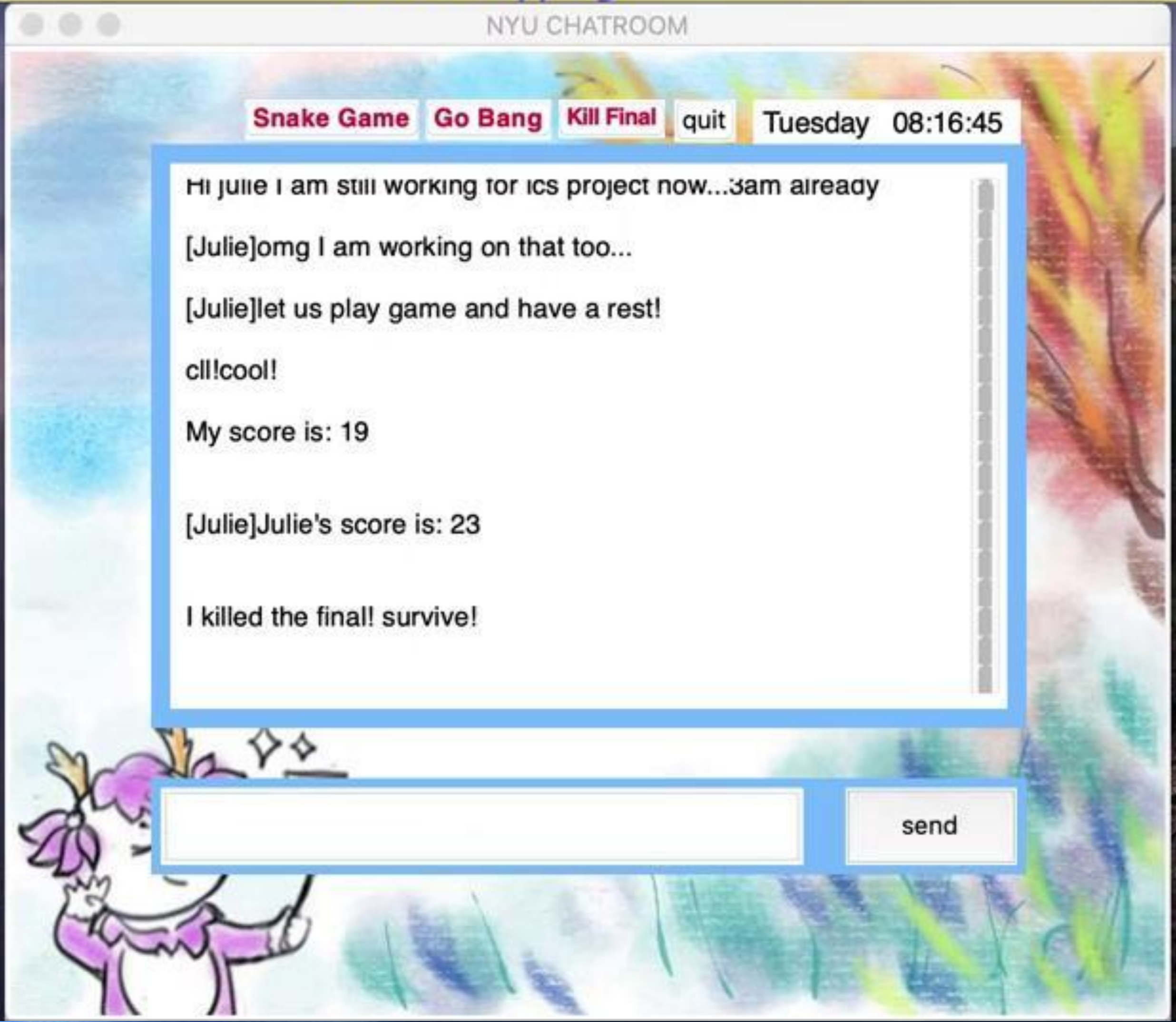
chat_client_class.py

- A box to display messages
- An entry box to send messages
- some buttons:
  - games
  - quit
  - time



```python
class Client:...

class GUI(Client):
    # constructor method
    def __init__(self, args):...

    # to display a clock
    def clock(self):...

    # if successfully login, destroy the login window
    def goAhead(self, name):...

    # The main layout of the chat
    def layout(self):...

    # Start a thread for sending messages
    def sendButton(self, msg):...

    # function to send messages
    def sendMessage(self):...

    # function to display messages
    def output(self):...

    def run_chat(self):...

    # shutdown the socket
    def quit(self):...
```

User 1: Tiffany



Hi julie i am still working for ics project now...3am already

[Julie]omg I am working on that too...

[Julie]let us play game and have a rest!

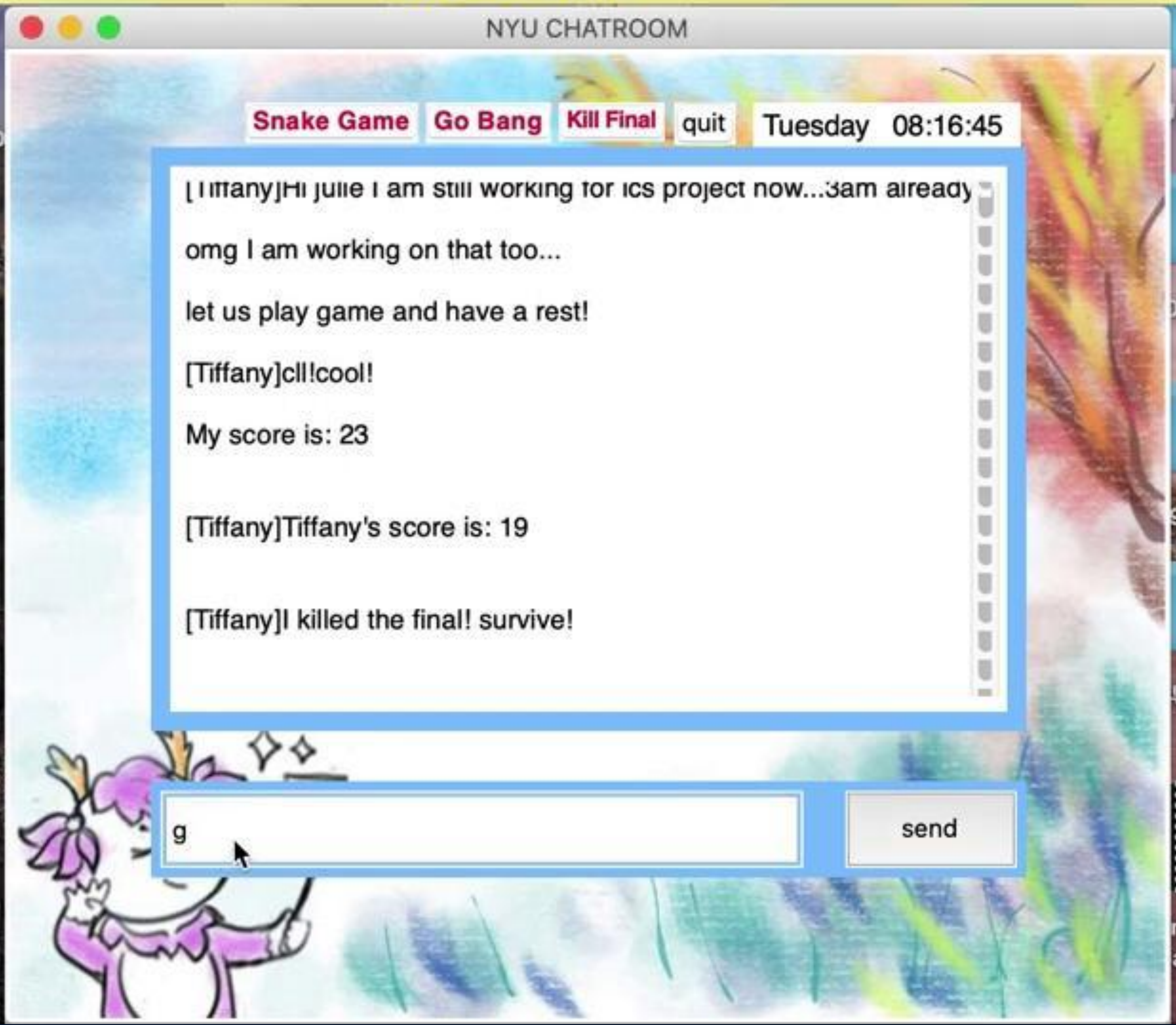
cll!cool!

My score is: 19

[Julie]Julie's score is: 23

I killed the final! survive!

User 2: Julie



[Tiffany]Hi julie i am still working for ics project now...3am already

omg I am working on that too...

let us play game and have a rest!

[Tiffany]cll!cool!

My score is: 23

[Tiffany]Tiffany's score is: 19

[Tiffany]I killed the final! survive!

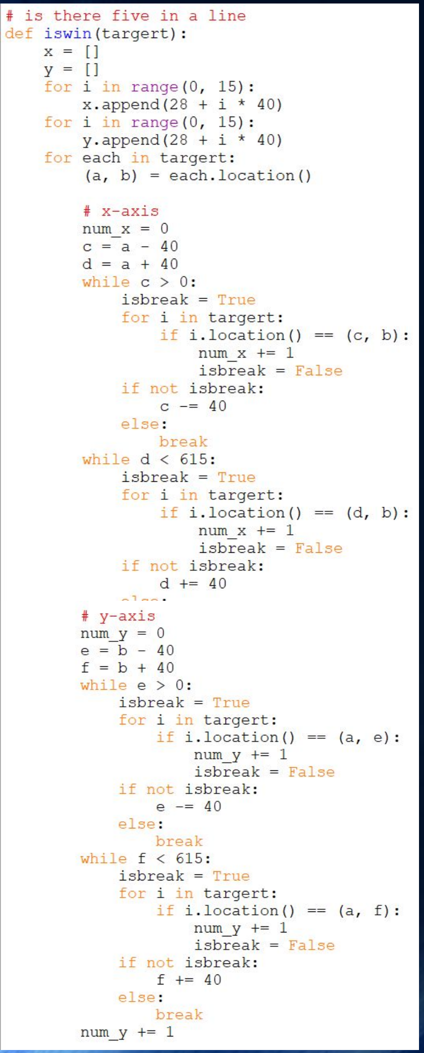
g

Snake Game Video Demo

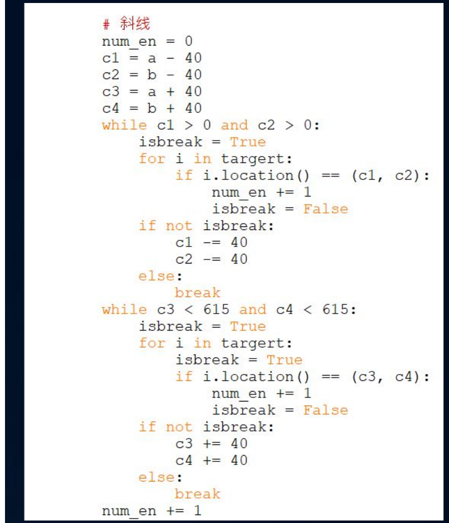
- movement with .turtle
- extend body length using list
- changing color
- update score
- background music

Gobang Game

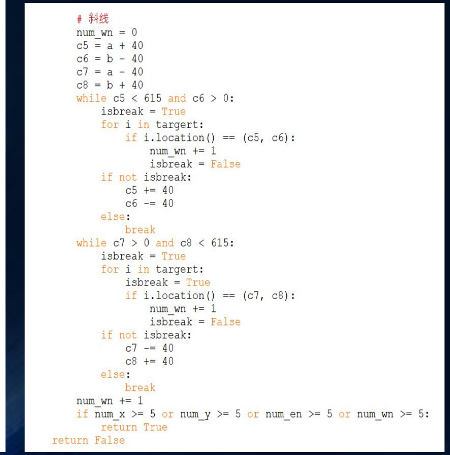
- Whether there are five chessmen in a line



```
# is there five in a line
def iswin(targert):
    x = []
    y = []
    for i in range(0, 15):
        x.append(28 + i * 40)
    for i in range(0, 15):
        y.append(28 + i * 40)
    for each in targert:
        (a, b) = each.location()

        # x-axis
        num_x = 0
        c = a - 40
        d = a + 40
        while c > 0:
            isbreak = True
            for i in targert:
                if i.location() == (c, b):
                    num_x += 1
                    isbreak = False
            if not isbreak:
                c -= 40
            else:
                break
        while d < 615:
            isbreak = True
            for i in targert:
                if i.location() == (d, b):
                    num_x += 1
                    isbreak = False
            if not isbreak:
                d += 40
            else:
        # y-axis
        num_y = 0
        e = b - 40
        f = b + 40
        while e > 0:
            isbreak = True
            for i in targert:
                if i.location() == (a, e):
                    num_y += 1
                    isbreak = False
            if not isbreak:
                e -= 40
            else:
                break
        while f < 615:
            isbreak = True
            for i in targert:
                if i.location() == (a, f):
                    num_y += 1
                    isbreak = False
            if not isbreak:
                f += 40
            else:
                break
        num_y += 1
```



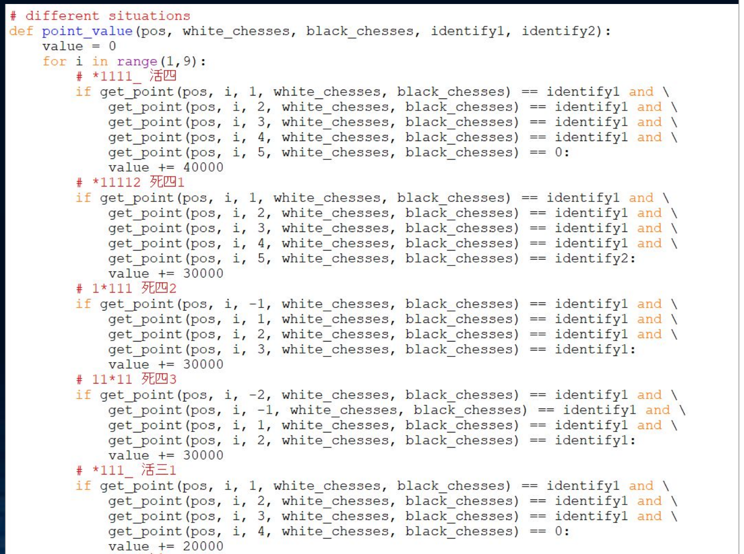
```
        # 斜线
        num_en = 0
        c1 = a - 40
        c2 = b - 40
        c3 = a + 40
        c4 = b + 40
        while c1 > 0 and c2 > 0:
            isbreak = True
            for i in targert:
                if i.location() == (c1, c2):
                    num_en += 1
                    isbreak = False
            if not isbreak:
                c1 -= 40
                c2 -= 40
            else:
                break
        while c3 < 615 and c4 < 615:
            isbreak = True
            for i in targert:
                isbreak = True
                if i.location() == (c3, c4):
                    num_en += 1
                    isbreak = False
            if not isbreak:
                c3 += 40
                c4 += 40
            else:
                break
        num_en += 1
```



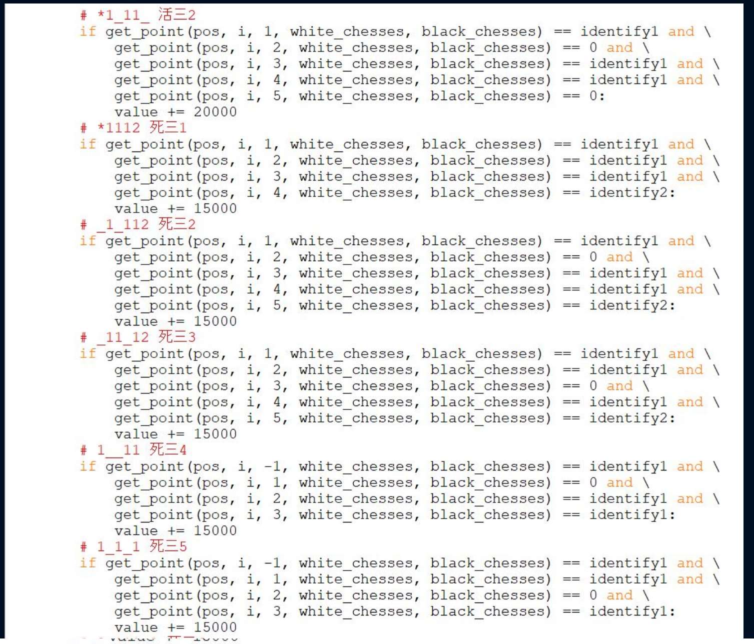
```
        # 斜线
        num_wn = 0
        c5 = a + 40
        c6 = b - 40
        c7 = a - 40
        c8 = b + 40
        while c5 < 615 and c6 > 0:
            isbreak = True
            for i in targert:
                if i.location() == (c5, c6):
                    num_wn += 1
                    isbreak = False
            if not isbreak:
                c5 += 40
                c6 -= 40
            else:
                break
        while c7 > 0 and c8 < 615:
            isbreak = True
            for i in targert:
                isbreak = True
                if i.location() == (c7, c8):
                    num_wn += 1
                    isbreak = False
            if not isbreak:
                c7 -= 40
                c8 += 40
            else:
                break
        num_wn += 1
        if num_x >= 5 or num_y >= 5 or num_en >= 5 or num_wn >= 5:
            return True
    return False
```

- Initialize the chessboard and chessmen
- Background music and click buttons

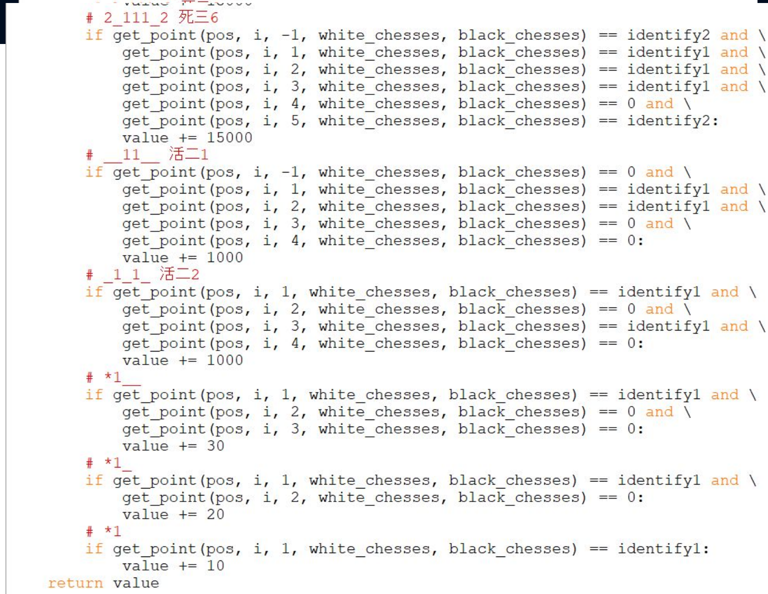
- Different situations in Gobang



```
# different situations
def point_value(pos, white_chesses, black_chesses, identify1, identify2):
    value = 0
    for i in range(1,9):
        # *1111_ 活四
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 4, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 5, white_chesses, black_chesses) == 0:
            value += 40000
        # *11112 死四1
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 4, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 5, white_chesses, black_chesses) == identify2:
            value += 30000
        # 1*111 死四2
        if get_point(pos, i, -1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1:
            value += 30000
        # 11*11 死四3
        if get_point(pos, i, -2, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, -1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == identify1:
            value += 30000
        # *111_ 活三1
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 4, white_chesses, black_chesses) == 0:
            value += 20000
```



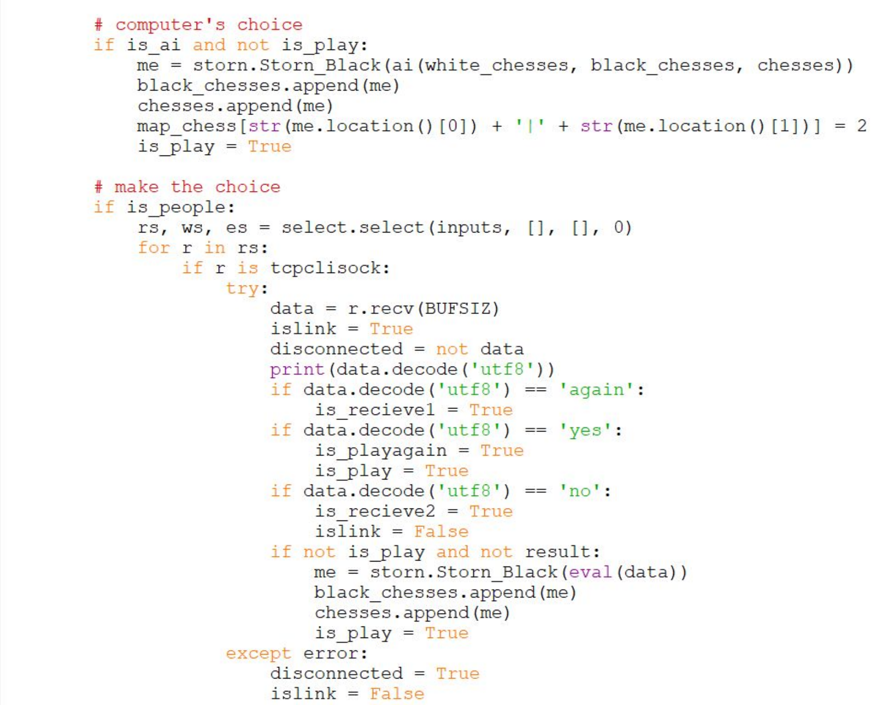
```
        # *1_11_ 活三2
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == 0 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 4, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 5, white_chesses, black_chesses) == 0:
            value += 20000
        # *1112 死三1
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 4, white_chesses, black_chesses) == identify2:
            value += 15000
        # _1_112 死三2
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == 0 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 4, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 5, white_chesses, black_chesses) == identify2:
            value += 15000
        # _11_12 死三3
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == 0 and \
            get_point(pos, i, 4, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 5, white_chesses, black_chesses) == identify2:
            value += 15000
        # 1__11 死三4
        if get_point(pos, i, -1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 1, white_chesses, black_chesses) == 0 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1:
            value += 15000
        # 1_1_1 死三5
        if get_point(pos, i, -1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == 0 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1:
            value += 15000
```



```
        # 2_111_2 死三6
        if get_point(pos, i, -1, white_chesses, black_chesses) == identify2 and \
            get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 4, white_chesses, black_chesses) == 0 and \
            get_point(pos, i, 5, white_chesses, black_chesses) == identify2:
            value += 15000
        # __11__ 活二1
        if get_point(pos, i, -1, white_chesses, black_chesses) == 0 and \
            get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == 0 and \
            get_point(pos, i, 4, white_chesses, black_chesses) == 0:
            value += 1000
        # _1_1_ 活二2
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == 0 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 4, white_chesses, black_chesses) == 0:
            value += 1000
        # *1__
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == 0 and \
            get_point(pos, i, 3, white_chesses, black_chesses) == 0:
            value += 30
        # *1_
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1 and \
            get_point(pos, i, 2, white_chesses, black_chesses) == 0:
            value += 20
        # *1
        if get_point(pos, i, 1, white_chesses, black_chesses) == identify1:
            value += 10
    return value
```

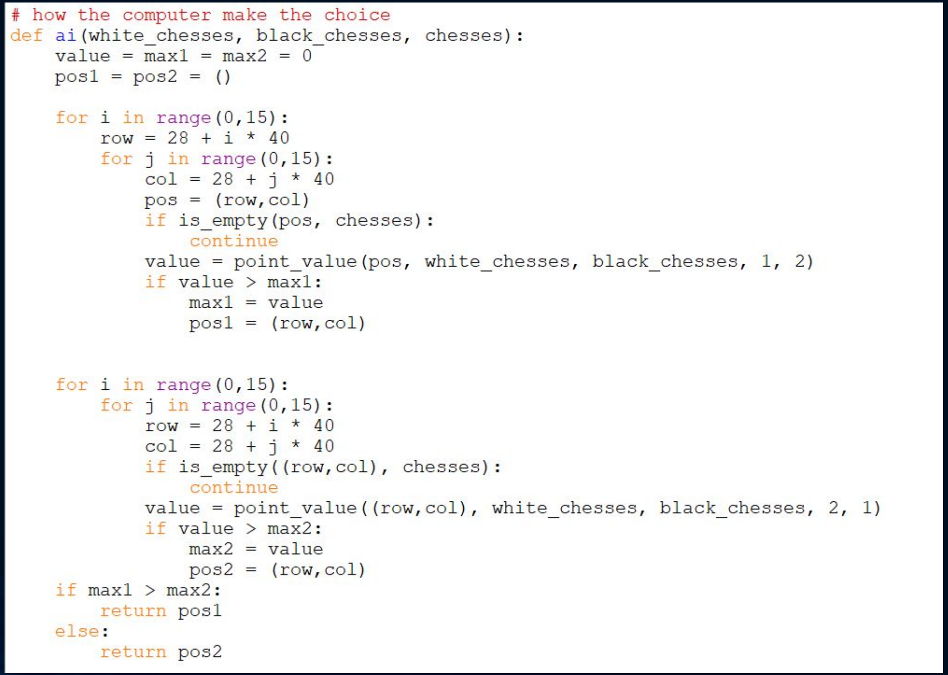
Gobang Game

- Play with Others
- (is_people)
- Ask for IP address



```
# computer's choice
if is_ai and not is_play:
    me = storn.Storn_Black(ai(white_chesses, black_chesses, chesses))
    black_chesses.append(me)
    chesses.append(me)
    map_chess[str(me.location()[0]) + '|' + str(me.location()[1])] = 2
    is_play = True

# make the choice
if is_people:
    rs, ws, es = select.select(inputs, [], [], 0)
    for r in rs:
        if r is tcpclisock:
            try:
                data = r.recv(BUFSIZ)
                islink = True
                disconnected = not data
                print(data.decode('utf8'))
                if data.decode('utf8') == 'again':
                    is_recieve1 = True
                if data.decode('utf8') == 'yes':
                    is_playagain = True
                    is_play = True
                if data.decode('utf8') == 'no':
                    is_recieve2 = True
                    islink = False
                if not is_play and not result:
                    me = storn.Storn_Black(eval(data))
                    black_chesses.append(me)
                    chesses.append(me)
                    is_play = True
            except error:
                disconnected = True
                islink = False
```

- Play with Computers (is_ai)
- Computer decide where to put the chess



```
# how the computer make the choice
def ai(white_chesses, black_chesses, chesses):
    value = max1 = max2 = 0
    pos1 = pos2 = ()

    for i in range(0,15):
        row = 28 + i * 40
        for j in range(0,15):
            col = 28 + j * 40
            pos = (row,col)
            if is_empty(pos, chesses):
                continue
            value = point_value(pos, white_chesses, black_chesses, 1, 2)
            if value > max1:
                max1 = value
                pos1 = (row,col)


    for i in range(0,15):
        for j in range(0,15):
            row = 28 + i * 40
            col = 28 + j * 40
            if is_empty((row,col), chesses):
                continue
            value = point_value((row,col), white_chesses, black_chesses, 2, 1)
            if value > max2:
                max2 = value
                pos2 = (row,col)
    if max1 > max2:
        return pos1
    else:
        return pos2
```

Gobang Game Video Demo

Kill Final Game
Video Demo

- use socket & server to share grades

- handmade graph

Thank for your watching !

Qainyu Zhu
Sihan Liu
Xiaoyan Tang

Good luck for all of you with your finals~~