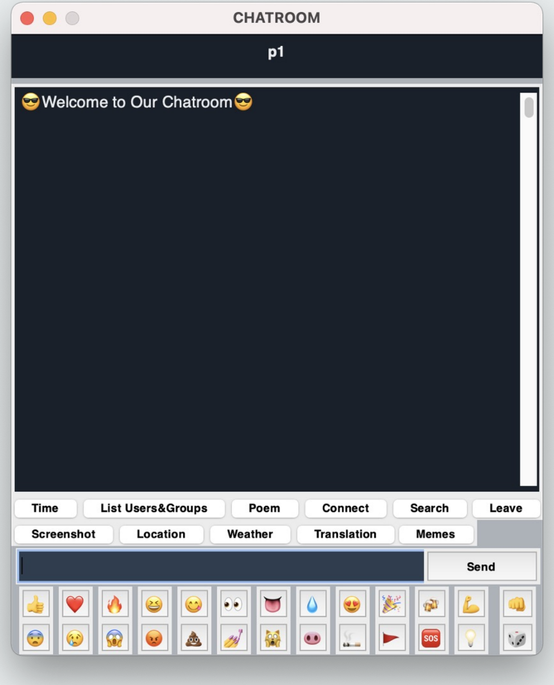
ICS Final Project GUI

by Ruoheng Du & Xiao Peng



😎Welcome to Our Chatroom😎

Overview

- Introduction of GUI & Our Project
- Project Demo
- Explaination of Codes
- Further Improvements

Introduction of GUI & Our Project

What is GUI?

The graphical user interface (GUI) is a form of user interface allows users to interact with electronic devices through items such as graphical icons instead of text-based user interfaces, typed command labels or text navigation.

(Wikipedia)

Introduction of GUI & Our Project

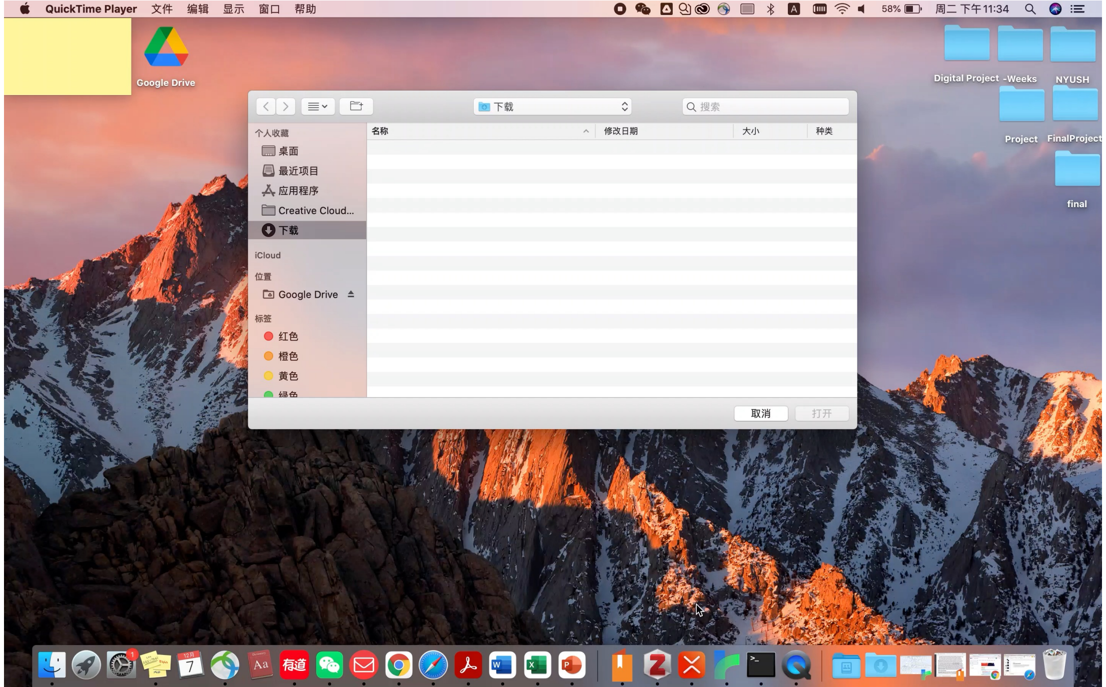
Our Project–Main Functions

We want to improve the previous chat system to make it more user friendly.

- interactive - command computers in graphical icons
- useful - easier and clearer to understand how to command

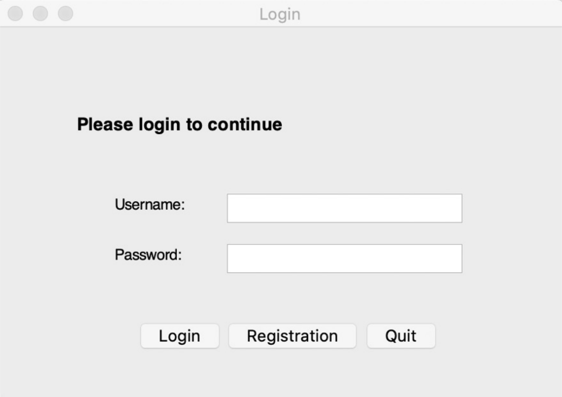
- registration and log-in
- group chat & search chat history
- get weather/location/translation
- basic game (dice & handgame)
- emoji& memes

Project Demo



Explaination of Codes

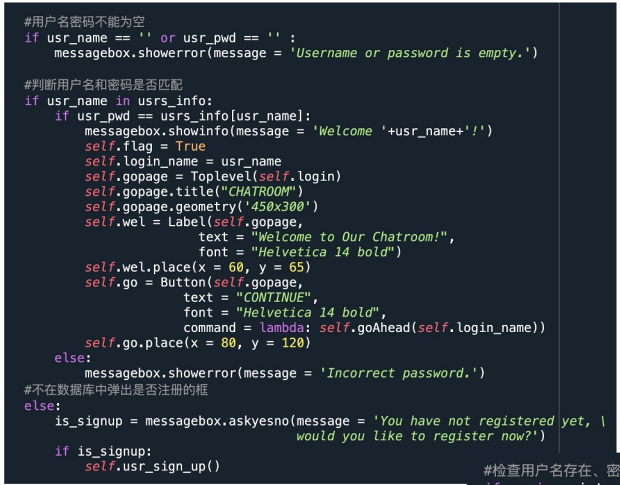
Registration & Log-in



Please login to continue

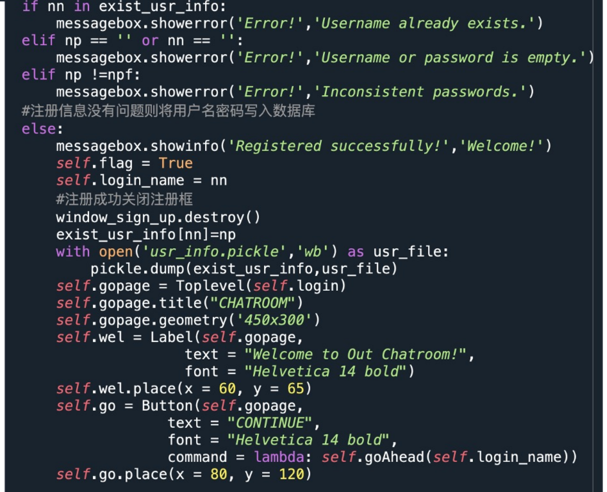
Username:

Password:



```
#用户名密码不能为空
if usr_name == '' or usr_pwd == '' :
    messagebox.showerror(message = 'Username or password is empty.')

#判断用户名和密码是否匹配
if usr_name in usrs_info:
    if usr_pwd == usrs_info[usr_name]:
        messagebox.showinfo(message = 'Welcome '+usr_name+'!')
        self.flag = True
        self.login_name = usr_name
        self.gopage = Toplevel(self.login)
        self.gopage.title("CHATROOM")
        self.gopage.geometry('450x300')
        self.wel = Label(self.gopage,
                       text = "Welcome to Our Chatroom!",
                       font = "Helvetica 14 bold")
        self.wel.place(x = 60, y = 65)
        self.go = Button(self.gopage,
                     text = "CONTINUE",
                     font = "Helvetica 14 bold",
                     command = lambda: self.goAhead(self.login_name))
        self.go.place(x = 80, y = 120)
    else:
        messagebox.showerror(message = 'Incorrect password.')
#不在数据库中弹出是否注册的框
else:
    is_signup = messagebox.askyesno(message = 'You have not registered yet, \
                                    would you like to register now?')
    if is_signup:
        self.usr_sign_up()
```

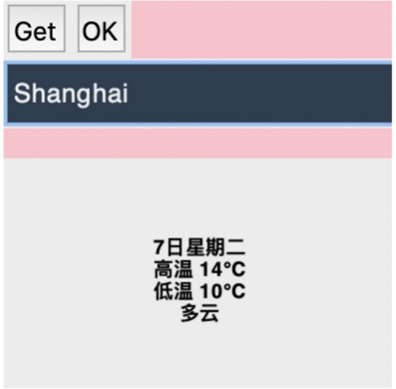
GUI



```
#检查用户名存在、密码为空、密码前后不一致
if nn in exist_usr_info:
    messagebox.showerror('Error!','Username already exists.')
elif np == '' or nn == '':
    messagebox.showerror('Error!','Username or password is empty.')
elif np !=npf:
    messagebox.showerror('Error!','Inconsistent passwords.')
#注册信息没有问题则将用户名密码写入数据库
else:
    messagebox.showinfo('Registered successfully!','Welcome!')
    self.flag = True
    self.login_name = nn
    #注册成功关闭注册框
    window_sign_up.destroy()
    exist_usr_info[nn]=np
    with open('usr_info.pickle','wb') as usr_file:
        pickle.dump(exist_usr_info,usr_file)
    self.gopage = Toplevel(self.login)
    self.gopage.title("CHATROOM")
    self.gopage.geometry('450x300')
    self.wel = Label(self.gopage,
                   text = "Welcome to Out Chatroom!",
                   font = "Helvetica 14 bold")
    self.wel.place(x = 60, y = 65)
    self.go = Button(self.gopage,
                 text = "CONTINUE",
                 font = "Helvetica 14 bold",
                 command = lambda: self.goAhead(self.login_name))
    self.go.place(x = 80, y = 120)
```

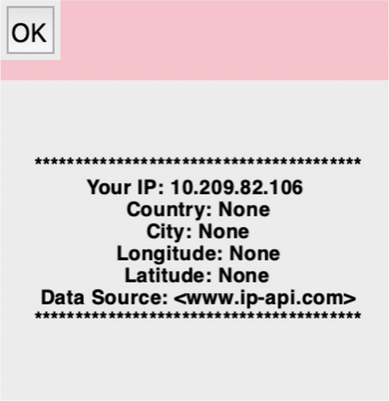
Explaination of Codes

Get Weather & Location



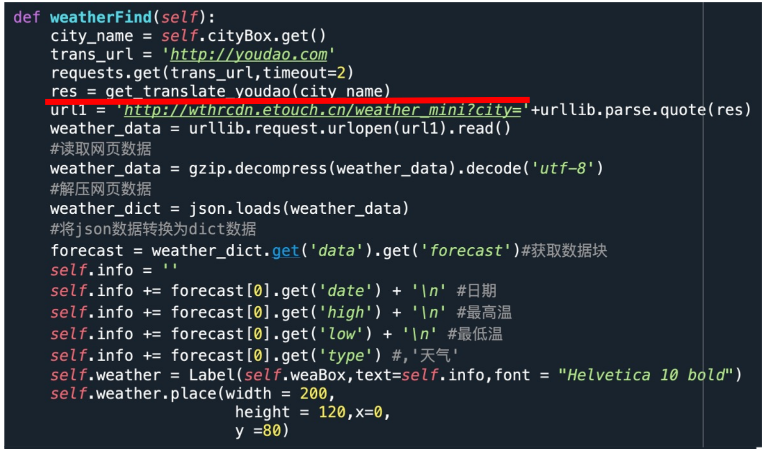
Shanghai

7日星期二
高温 14°C
低温 10°C
多云

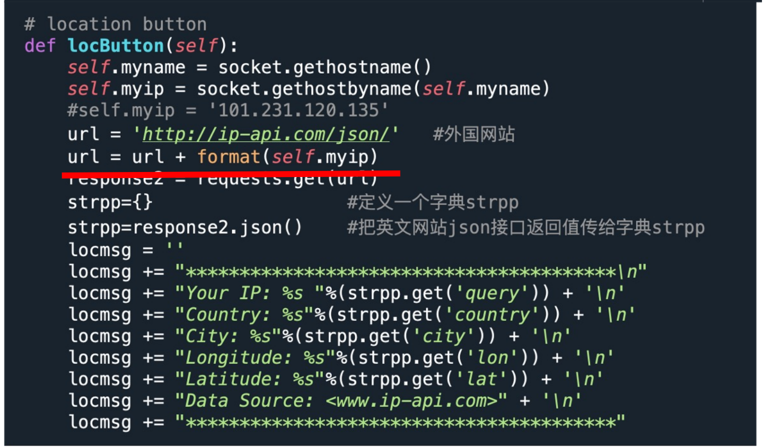


```
******************************************
Your IP: 10.209.82.106
Country: None
City: None
Longitude: None
Latitude: None
Data Source: <www.ip-api.com>
******************************************
```

GUI



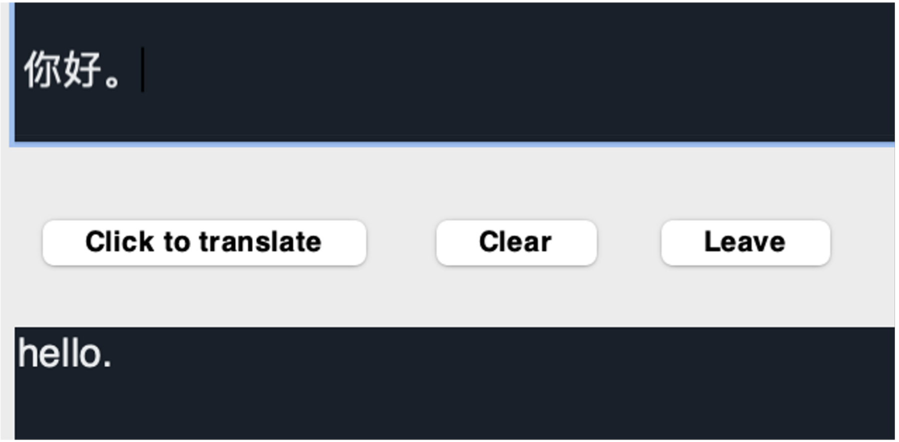
```
def weatherFind(self):
    city_name = self.cityBox.get()
    trans_url = 'http://youdao.com'
    requests.get(trans_url,timeout=2)
    res = get_translate_youdao(city_name)
    url1 = 'http://wthrcdn.etouch.cn/weather_mini?city='+urllib.parse.quote(res)
    weather_data = urllib.request.urlopen(url1).read()
    #读取网页数据
    weather_data = gzip.decompress(weather_data).decode('utf-8')
    #解压网页数据
    weather_dict = json.loads(weather_data)
    #将json数据转换为dict数据
    forecast = weather_dict.get('data').get('forecast')#获取数据块
    self.info = ''
    self.info += forecast[0].get('date') + '\n' #日期
    self.info += forecast[0].get('high') + '\n' #最高温
    self.info += forecast[0].get('low') + '\n' #最低温
    self.info += forecast[0].get('type') #,'天气'
    self.weather = Label(self.weaBox,text=self.info,font = "Helvetica 10 bold")
    self.weather.place(width = 200,
                       height = 120,x=0,
                       y =80)
```



```
# location button
def locButton(self):
    self.myname = socket.gethostname()
    self.myip = socket.gethostbyname(self.myname)
    #self.myip = '101.231.120.135'
    url = 'http://ip-api.com/json/'    #外国网站
    url = url + format(self.myip)
    response2 = requests.get(url)
    strpp={}                         #定义一个字典strpp
    strpp=response2.json()     #把英文网站json接口返回值传给字典strpp
    locmsg = ''
    locmsg += "*****************************************\n"
    locmsg += "Your IP: %s "%(strpp.get('query')) + '\n'
    locmsg += "Country: %s"%(strpp.get('country')) + '\n'
    locmsg += "City: %s"%(strpp.get('city')) + '\n'
    locmsg += "Longitude: %s"%(strpp.get('lon')) + '\n'
    locmsg += "Latitude: %s"%(strpp.get('lat')) + '\n'
    locmsg += "Data Source: <www.ip-api.com>" + '\n'
    locmsg += "*****************************************"
```

Explaination of Codes

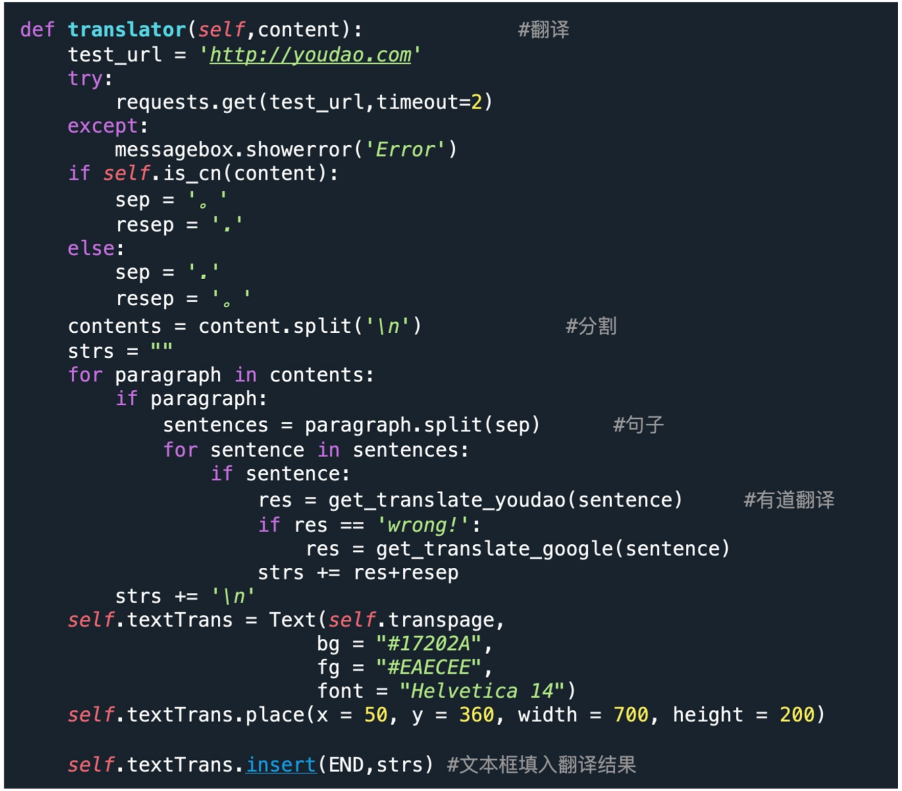
Translation



你好。

hello.

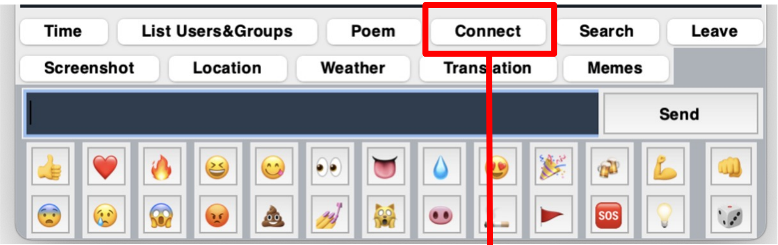
GUI

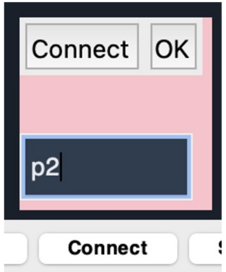


```
def translator(self,content):                       #翻译
    test_url = 'http://youdao.com'
    try:
        requests.get(test_url,timeout=2)
    except:
        messagebox.showerror('Error')
    if self.is_cn(content):
        sep = '。'
        resep = '.'
    else:
        sep = '.'
        resep = '。'
    contents = content.split('\n')                  #分割
    strs = ""
    for paragraph in contents:
        if paragraph:
            sentences = paragraph.split(sep)        #句子
            for sentence in sentences:
                if sentence:
                    res = get_translate_youdao(sentence)     #有道翻译
                    if res == 'wrong!':
                        res = get_translate_google(sentence)
                    strs += res+resep
        strs += '\n'
    self.textTrans = Text(self.transpage,
                         bg = "#17202A",
                         fg = "#EAECEE",
                         font = "Helvetica 14")
    self.textTrans.place(x = 50, y = 360, width = 700, height = 200)

    self.textTrans.insert(END,strs) #文本框填入翻译结果
```

Explaination of Codes

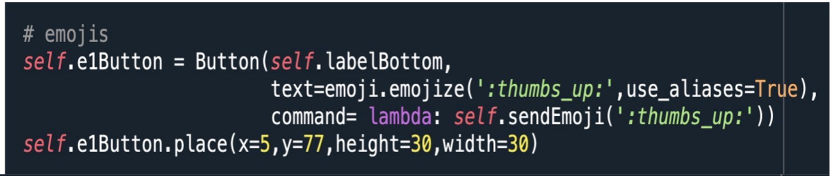
Add Buttons & Emojis



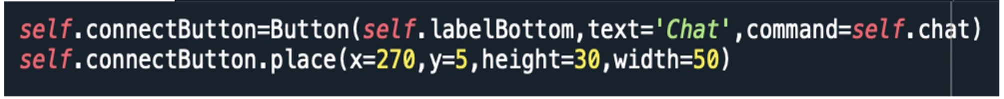


p2

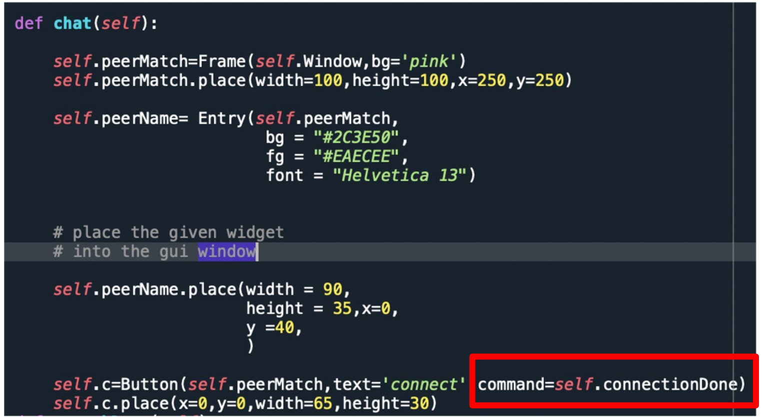
GUI



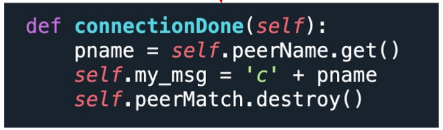
```
# emojis
self.e1Button = Button(self.labelBottom,
                       text=emoji.emojize(':thumbs_up:',use_aliases=True),
                       command= lambda: self.sendEmoji(':thumbs_up:'))
self.e1Button.place(x=5,y=77,height=30,width=30)
```



```
self.connectButton=Button(self.labelBottom,text='Chat',command=self.chat)
self.connectButton.place(x=270,y=5,height=30,width=50)
```



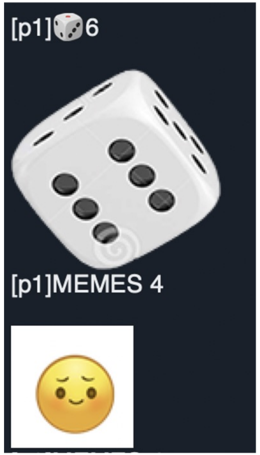
```
def chat(self):

    self.peerMatch=Frame(self.Window,bg='pink')
    self.peerMatch.place(width=100,height=100,x=250,y=250)

    self.peerName= Entry(self.peerMatch,
                         bg = "#2C3E50",
                         fg = "#EAECEE",
                         font = "Helvetica 13")


    # place the given widget
    # into the gui window

    self.peerName.place(width = 90,
                        height = 35,x=0,
                        y =40,
                        )

    self.c=Button(self.peerMatch,text='connect' command=self.connectionDone)
    self.c.place(x=0,y=0,width=65,height=30)
```



```
def connectionDone(self):
    pname = self.peerName.get()
    self.my_msg = 'c' + pname
    self.peerMatch.destroy()
```

Explaination of Codes

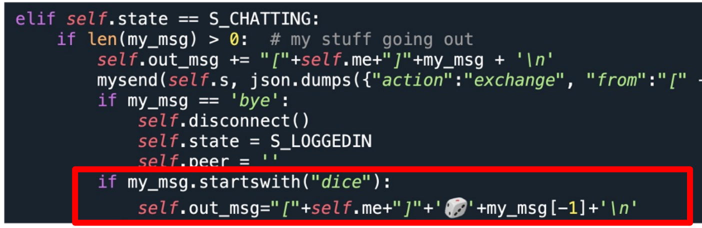
Dice Game & Memes



[p1]🎲6

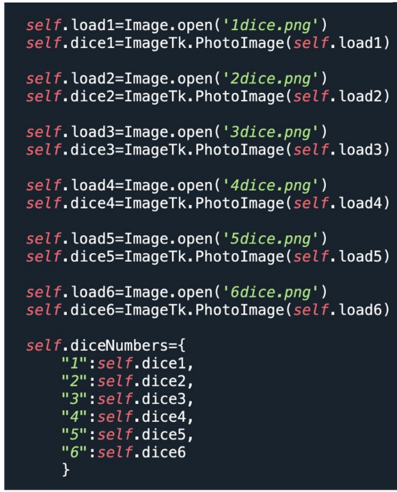
[p1]MEMES 4

Client_state_machine

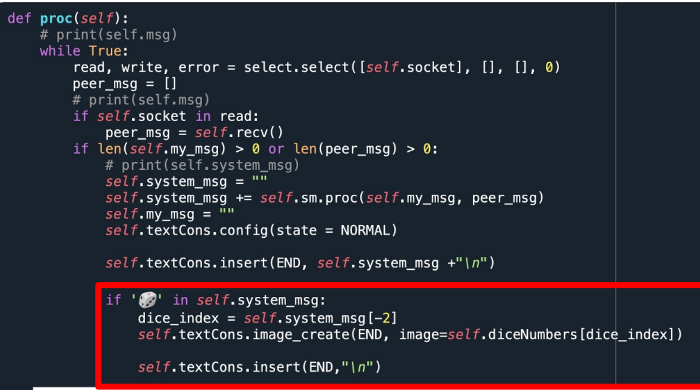


```
elif self.state == S_CHATTING:
    if len(my_msg) > 0:  # my stuff going out
        self.out_msg += "["+self.me+"]"+my_msg + '\n'
        mysend(self.s, json.dumps({"action":"exchange", "from":"["
        if my_msg == 'bye':
            self.disconnect()
            self.state = S_LOGGEDIN
            self.peer = ''
        if my_msg.startswith("dice"):
            self.out_msg="["+self.me+"]"+'🎲'+my_msg[-1]+'\n'
```

GUI



```
self.load1=Image.open('1dice.png')
self.dice1=ImageTk.PhotoImage(self.load1)

self.load2=Image.open('2dice.png')
self.dice2=ImageTk.PhotoImage(self.load2)

self.load3=Image.open('3dice.png')
self.dice3=ImageTk.PhotoImage(self.load3)

self.load4=Image.open('4dice.png')
self.dice4=ImageTk.PhotoImage(self.load4)

self.load5=Image.open('5dice.png')
self.dice5=ImageTk.PhotoImage(self.load5)

self.load6=Image.open('6dice.png')
self.dice6=ImageTk.PhotoImage(self.load6)

self.diceNumbers={
    "1":self.dice1,
    "2":self.dice2,
    "3":self.dice3,
    "4":self.dice4,
    "5":self.dice5,
    "6":self.dice6
    }
```



```
def proc(self):
    # print(self.msg)
    while True:
        read, write, error = select.select([self.socket], [], [], 0)
        peer_msg = []
        # print(self.msg)
        if self.socket in read:
            peer_msg = self.recv()
        if len(self.my_msg) > 0 or len(peer_msg) > 0:
            # print(self.system_msg)
            self.system_msg = ""
            self.system_msg += self.sm.proc(self.my_msg, peer_msg)
            self.my_msg = ""
            self.textCons.config(state = NORMAL)

            self.textCons.insert(END, self.system_msg +"\n")

            if '🎲' in self.system_msg:
                dice_index = self.system_msg[-2]
                self.textCons.image_create(END, image=self.diceNumbers[dice_index])

                self.textCons.insert(END,"\n")
```

Further Improvement

- Integrate translation function into the chat interface
- More individualized options such as profile picture
- Upload pictures and save as memes