ICDS Final Project

Rongyao Li rl4785
Kehan Luo kl4747

CONTENTS

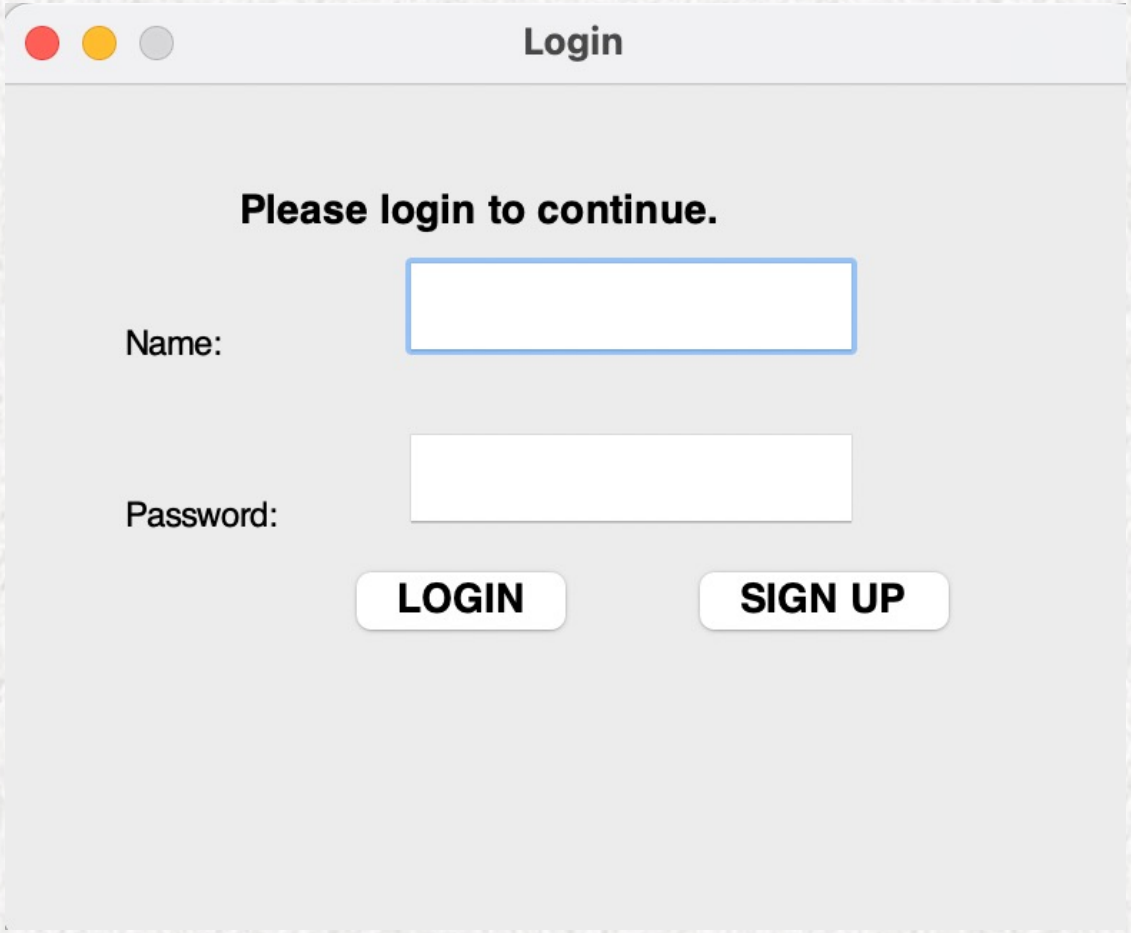
1 Introduction

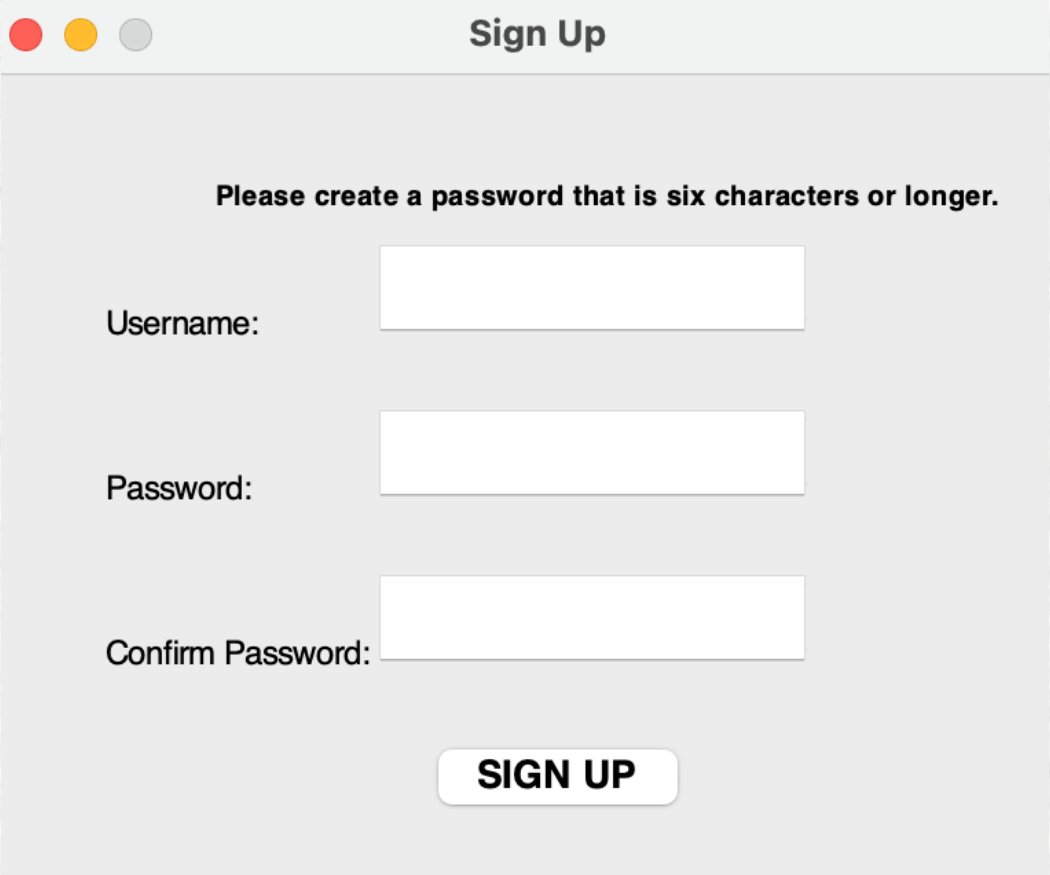
1

Introduction—— Motivations

Graphical User Interface	Develop a GUI for the chat system.
Online Game	Create an easy online game to add some fun to the chat system.

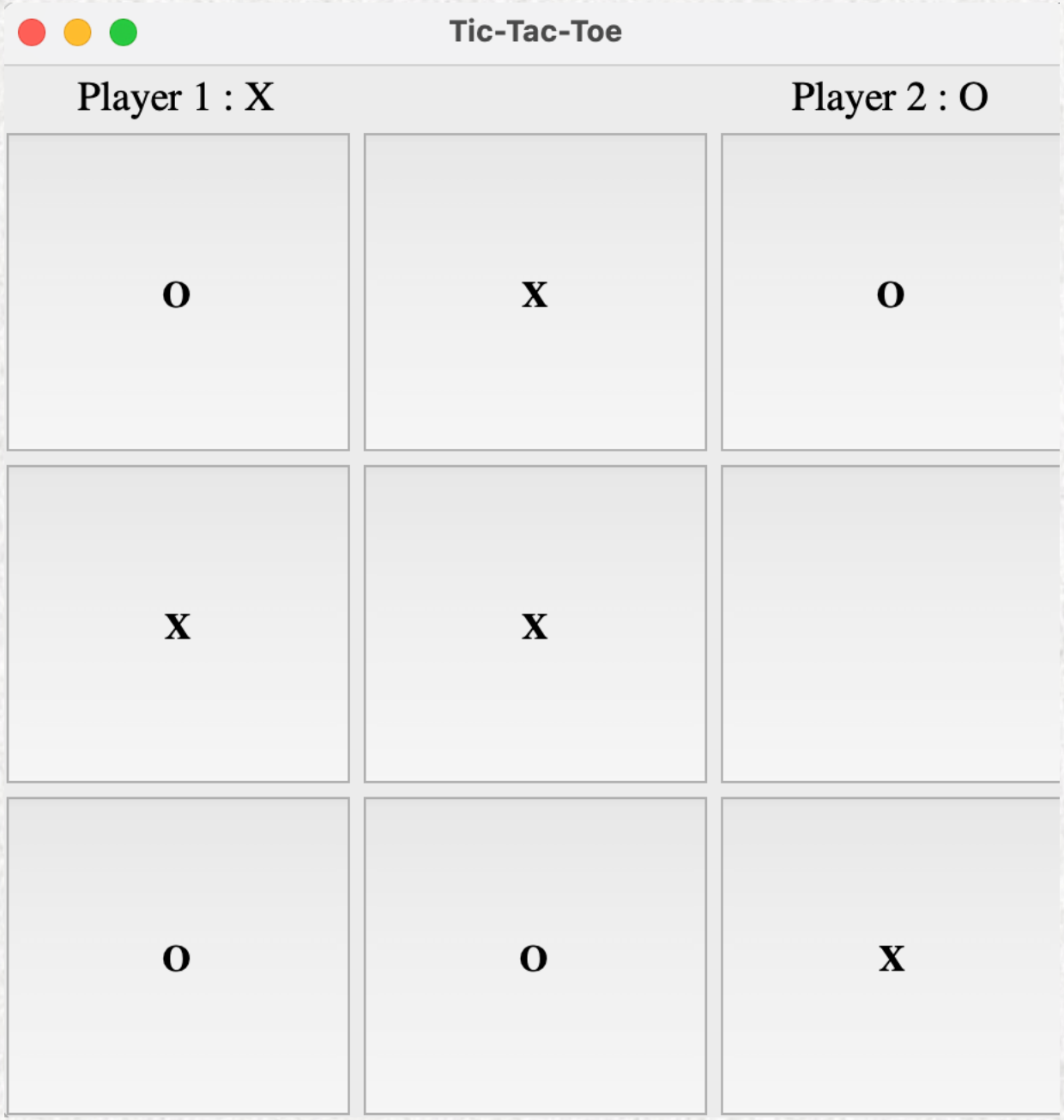
1 Introduction—— GUI





1

Introduction —Game



2 Demo

3 Discussion

3 Discussion—— GUI

Sign-up Memory

```python
# Read & Load Sign-up Memories
def load_registered_users(self):
    try:
        with open("registered_users.json", "r") as file:
            self.reg_names = json.load(file)
    except FileNotFoundError:
        self.reg_names = {}

def save_registered_users(self):
    with open("registered_users.json", "w") as file:
        json.dump(self.reg_names, file)
```

The sign-uped message will be saved as a dictionary in "registered_users.json".

3 Discussion—— GUI

Sign-up Algorithm

```
# Check if the user exists already
if username in self.reg_names.keys():
    messagebox.showerror("Error", "The user exists already!")
    return

# Check if password meets criteria
if len(password) < 6:
    messagebox.showerror(
        "Error", "Please use passwords equal to or longer than 6 characters!"
    )
    return

# Check if passwords match
if password != confirm_password:
    messagebox.showerror(
        "Error", "Passwords do not coincide! Please confirm again!"
    )
    return

self.reg_names[username] = password
self.save_registered_users()
messagebox.showinfo("Success", "Registration successful!")
self.signup.destroy()
```

We added a condition for the length of password.

3 Discussion—— GUI

Log-in Window

```
# Go-ahead from Sign-up to Log-in
def goAhead(self, name, password):

    if len(name) == 0 or name not in self.reg_names.keys():
        messagebox.showerror("Error", "User does not exist, please sign up first!")
        return

    if password != self.reg_names[name]:
        messagebox.showerror("Error", "Incorrect password!")
        return
```

3 Discussion—— GUI

Chat Window

```
# function to basically start the thread for sending messages
def sendButton(self, msg):
    self.textCons.config(state=DISABLED)
    self.my_msg = msg
    self.textCons.config(state=NORMAL)
    self.textCons.insert(END, f"You: {msg}\n\n")
    self.textCons.config(state=DISABLED)
    self.entryMsg.delete(0, END)
```

We updated the Chat Window so that users can see their own messages.

3 Discussion—— GUI

```python
def proc(self):
    MAX_LINES = 100  # Set the maximum number of lines appearing on GUI
    while True:
        read, write, error = select.select([self.socket], [], [], 0)
        peer_msg = []
        if self.socket in read:
            peer_msg = self.recv()
        if len(self.my_msg) > 0 or len(peer_msg) > 0:
            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\n")
            lines = self.textCons.get("1.0", "end").split("\n")
            if len(lines) > MAX_LINES:
                self.textCons.delete("1.0", f"{len(lines)-MAX_LINES}.0")
            self.textCons.config(state=DISABLED)
            self.textCons.see(END)
```

Chat Window

We updated the Chat Window so that previous messages will not appear repeatedly.

3 Discussion—— Game Tic Tac Toe

```
self.buttonG = Button(
    self.labelBottom,
    text="Game",
    font="Helvetica 10 bold",
    width=20,
    bg="#ABB2B9",
    command=self.start_game)
self.buttonG.place(relx=0.88,
    rely=0.008, relheight=0.06,
    relwidth=0.11)
```

Interface


```
def start_game(self):
    self.my_msg = "request_to_start_a_game"
```

3 Discussion—— Game Tic Tac Toe

```
if self.state == S_LOGGEDIN:
    # todo: can't deal with multiple lines yet
    if len(my_msg) > 0:
        if my_msg == "request_to_start_a_game":
            self.out_msg += (
                "Unsuccessful! You need two users to play the game!\n"
            )
```

```
elif self.state == S_CHATTING:
    if len(my_msg) > 0:
        if my_msg == "request_to_start_a_game":
            mysend(
                self.s,
                json.dumps({"action": "game", "from": "[" + self.me + "]"}),
            )
```

3 Discussion—— Game Tic Tac Toe

```
elif peer_msg["action"] == "game":
    if peer_msg["status"] == "fail":

        self.out_msg += peer_msg["results"]

    elif peer_msg["status"] == "success":
        self.out_msg += peer_msg["results"]
```

Client_State_Machine will update the clients' states into "gaming".

```
elif peer_msg["action"] == "gaming":
    if peer_msg["status"] == "continue":

        if peer_msg["operation"] == 1:
            self.out_msg += "systeminfo1" + peer_msg["mark"]
```

3 Discussion—— Game Tic Tac Toe

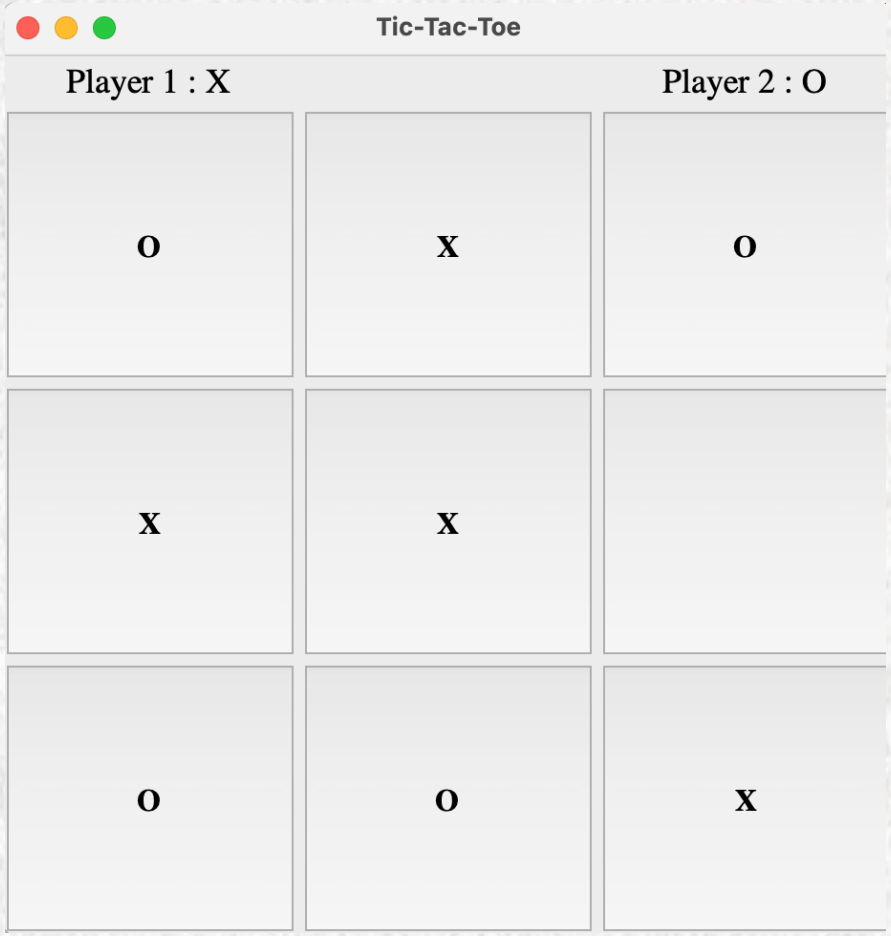
```
elif msg["action"] == "gaming":
    from_name = self.logged_sock2name[from_sock]
    The_guys = self.group.list_me(from_name)
    in_group, groupnum = self.group.find_group(from_name)

    choose = msg["operation"]
    if choose in self.gameinfo[groupnum][0]:
        self.gameinfo[groupnum][0].remove(choose)
        if self.gameinfo[groupnum][1] % 2 == 0:
            self.gameinfo[groupnum][3] = "X"

            self.gameinfo[groupnum][2][choose] = self.gameinfo[groupnum][3]

        elif self.gameinfo[groupnum][1] % 2 != 0:
            self.gameinfo[groupnum][3] = "O"
            self.gameinfo[groupnum][2][choose] = self.gameinfo[groupnum][3]
        self.gameinfo[groupnum][1] += 1
```

Chat server will connect these two users to the game.

3 Discussion—— Game Tic Tac Toe

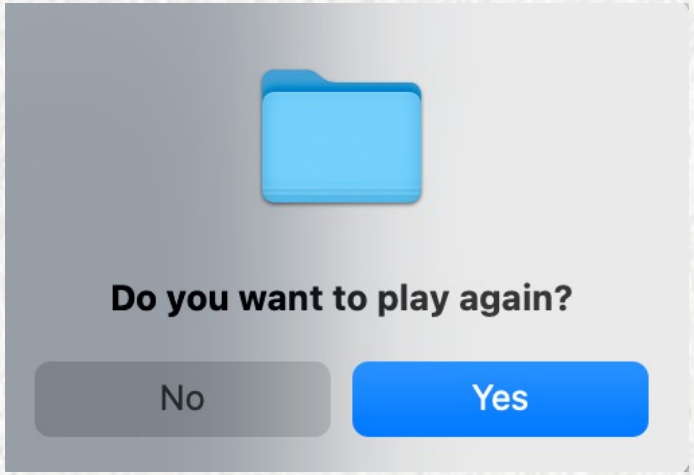
```python
def game_layout(self):
    self.gameWindow = Toplevel(self.Window)
    self.gameWindow.title("Tic-Tac-Toe")
    Label(self.gameWindow, text="Player 1 : X", font="times 18").grid(
        row=0, column=1
    )
    Label(self.gameWindow, text="Player 2 : O", font="times 18").grid(
        row=0, column=3
    )
    self.button1 = Button(
        self.gameWindow,
        width=15,
        height=7,
        font=("Times 16 bold"),
        command=self.checker1,
    )
```

```python
def win(self, panels, sign):
    return (
        (panels[1] == panels[2] == panels[3] == sign)
        or (panels[1] == panels[4] == panels[7] == sign)
        or (panels[1] == panels[5] == panels[9] == sign)
        or (panels[2] == panels[5] == panels[8] == sign)
        or (panels[3] == panels[6] == panels[9] == sign)
        or (panels[3] == panels[5] == panels[7] == sign)
        or (panels[4] == panels[5] == panels[6] == sign)
        or (panels[7] == panels[8] == panels[9] == sign)
```



3 Discussion—— Game Tic Tac Toe

```
elif self.system_msg == "serverinfoPlayer 1":
    messagebox.showinfo(
        title="Result", message="Player 1 wins the game!"
    )
    self.play_again_dialog()
elif self.system_msg == "serverinfoPlayer 2":
    messagebox.showinfo(
        title="Result", message="Player 2 wins the game!"
    )
    self.play_again_dialog()
```




```
# Ask players whether to play again
def play_again_dialog(self):
    choice = messagebox.askquestion("Play Again", "Do you want to play again?")
    if choice == "yes":
        self.gameWindow.destroy()
        self.start_game()
    else:
        self.gameWindow.destroy()
```

4
5

Analysis Improvement

4 Analysis & Improvement

Security & Reliability
Online Game

Thank You!

Feel free to contact if you have any questions!

rl4785@nyu.edu
kl4747@nyu.edu