Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-002-S2024/it114-rps-milestone-4-2024/grade/pd438

IT114-002-S2024 - [IT114] RPS Milestone 4 2024

Submissions:

Submission Selection

1 Submission [active] 4/30/2024 9:41:53 AM

Instructions

COLLAPSE

Implement the Milestone 4 features from the project's proposal document: https://docs.google.com/document/d/11SRMo7JkLAMM-PuuiGwl_Z-QXP3pyQ7xN3lRxwmcwCc/view
Make sure you add your ucid/date as code comments where code changes are done
All code changes should reach the Milestone4 branch
Create a pull request from Milestone4 to main and keep it open until you get the output PDF from this assignment.
Gather the evidence of feature completion based on the below tasks.
Once finished, get the output PDF and copy/move it to your repository folder on your local machine.
Run the necessary git add, commit, and push steps to move it to GitHub
Complete the pull request that was opened earlier
Upload the same output PDF to Canvas

Branch name: Milestone4

Tasks: 15 Points: 10.00

Implement Extra Options (2.25 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details

#	Points	Details
#1	1	Show how the extra options are displayed and utilized (including during the "battle" check)
#2	1	There should be an odd number of total choices (at least add 2 for this feature) https://wrpsa.com/different-variations-of-rock-paper-scissors/
#3	1	Screenshots should include ucid and date comment
#4	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The two extra options that are displayed are the sword and the shield.

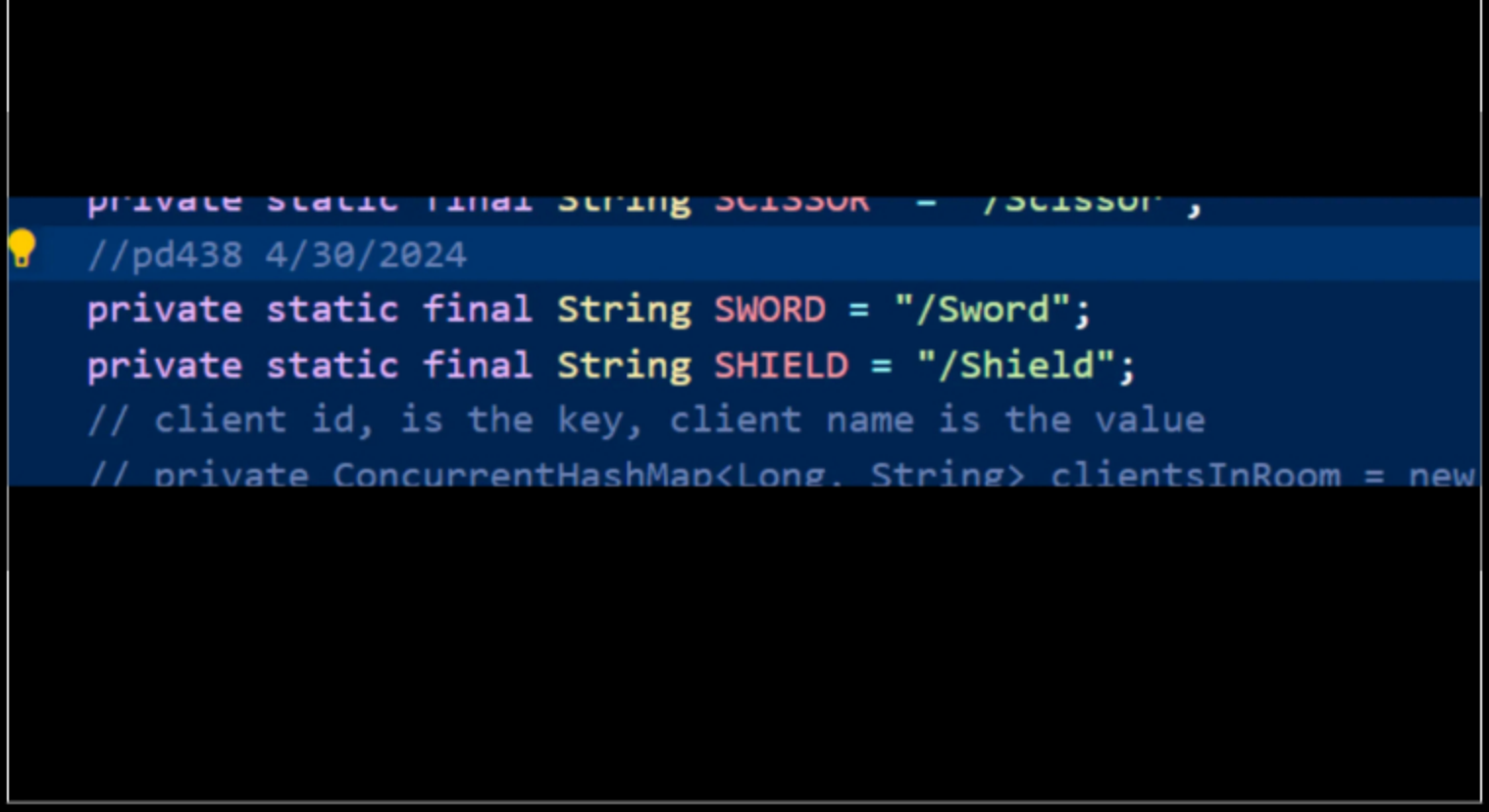
Checklist Items (4)

#1 Show how the extra options are displayed and utilized (including during the "battle" check)

#2 There should be an odd number of total choices (at least add 2 for this feature) https://wrpsa.com/different-variations-of-rock-paper-scissors/

#3 Screenshots should include ucid and date comment

#4 Each screenshot should be clearly captioned



Client side code being revealed

Checklist Items (4)

#1 Show how the extra options are displayed and utilized (including during the "battle" check)

#2 There should be an odd number of total choices (at least add 2 for this feature) https://wrpsa.com/different-variations-of-rock-paper-scissors/

#3 Screenshots should include ucid and date comment

#4 Each screenshot should be clearly captioned

COLLAPSE

Task #2 - Points: 1

Text: Screenshots of the demo

Checklist

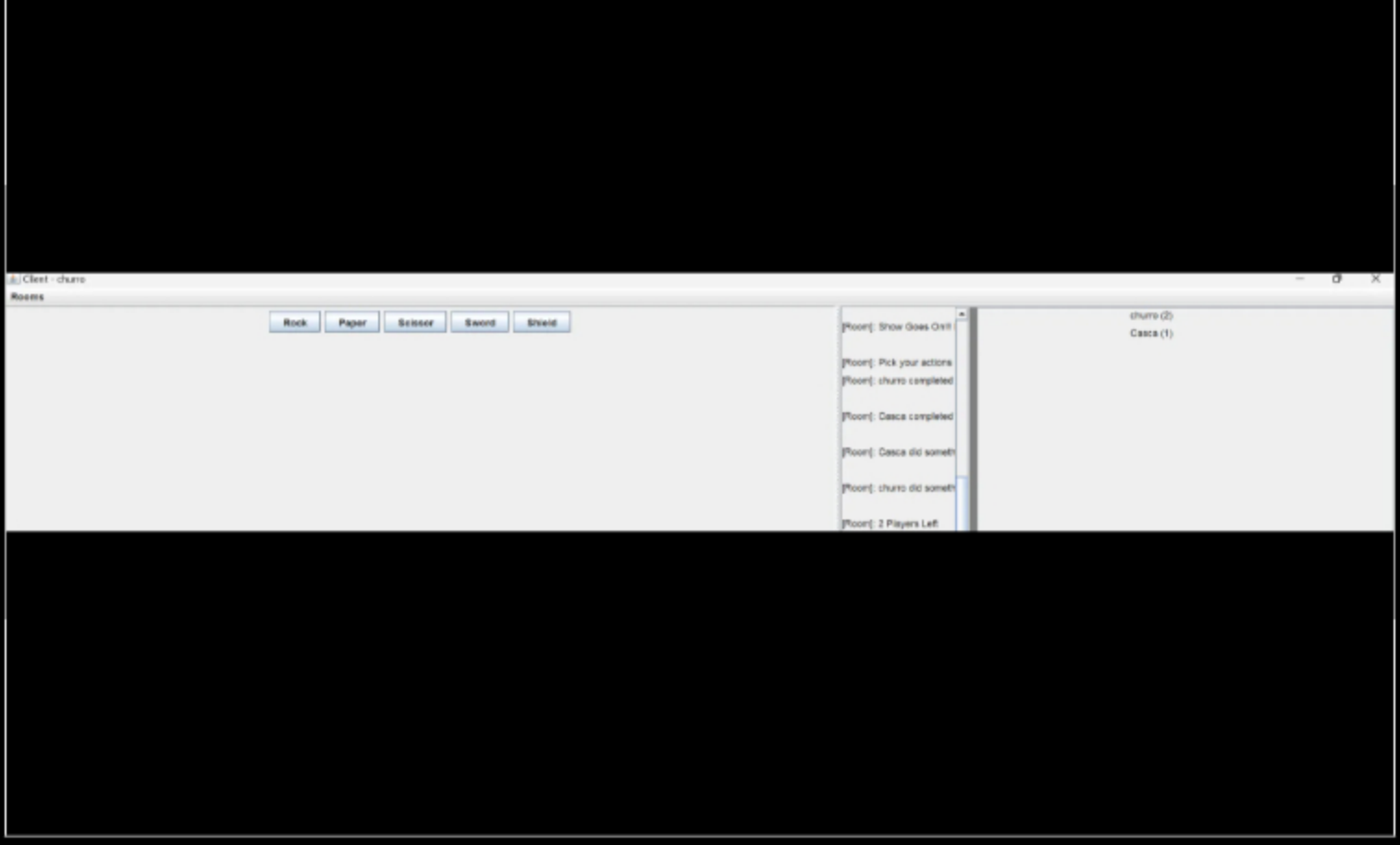
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Show how the options appear in the UI
#2	1	Show results/events containing the new options
#3	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small Medium Large



This Displays options appearing with the UI and having the new options.

Checklist Items (3)

#1 Show how the options appear in the UI

#2 Show results/events containing the new options

#3 Each screenshot should be clearly captioned

COLLAPSE

Task #3 - Points: 1

Text: Explanation of Solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Mention what options were added and their battle logic
#2	1	Mention how you incorporated into the current system

Response:

The options that were added, are the sword and shield. I had their own unique choices. I incorporated them as their own unique choices. While being selectable options, the new choices only interact with each other.

Implement a Cooldown for Choices Per Player (2.25 pts.)

^COLLAPSE ^

^COLLAPSE ^

Task #1 - Points: 1

Text: Screenshots of the code

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Show the code that limits a player's choice to not repeat twice in a row (include client and server-side code) (i.e., A User who chose Rock round 1 can't choose Rock again until round 3)
#2	1	Screenshots should include ucid and date comment
#3	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```
//Pd438 previous Choice Implementation
if (sp.getpreviousChoice() == choice) {
    client.sendMessage(Constants.DEFAULT_CLIENT_ID, message:"You Already Chose this!!!");
    return;
}
```

Previous Choice Implementation and it displays that they cannot choose until the third turn.

Checklist Items (0)

^COLLAPSE ^

Task #2 - Points: 1

Text: Screenshot of the UI

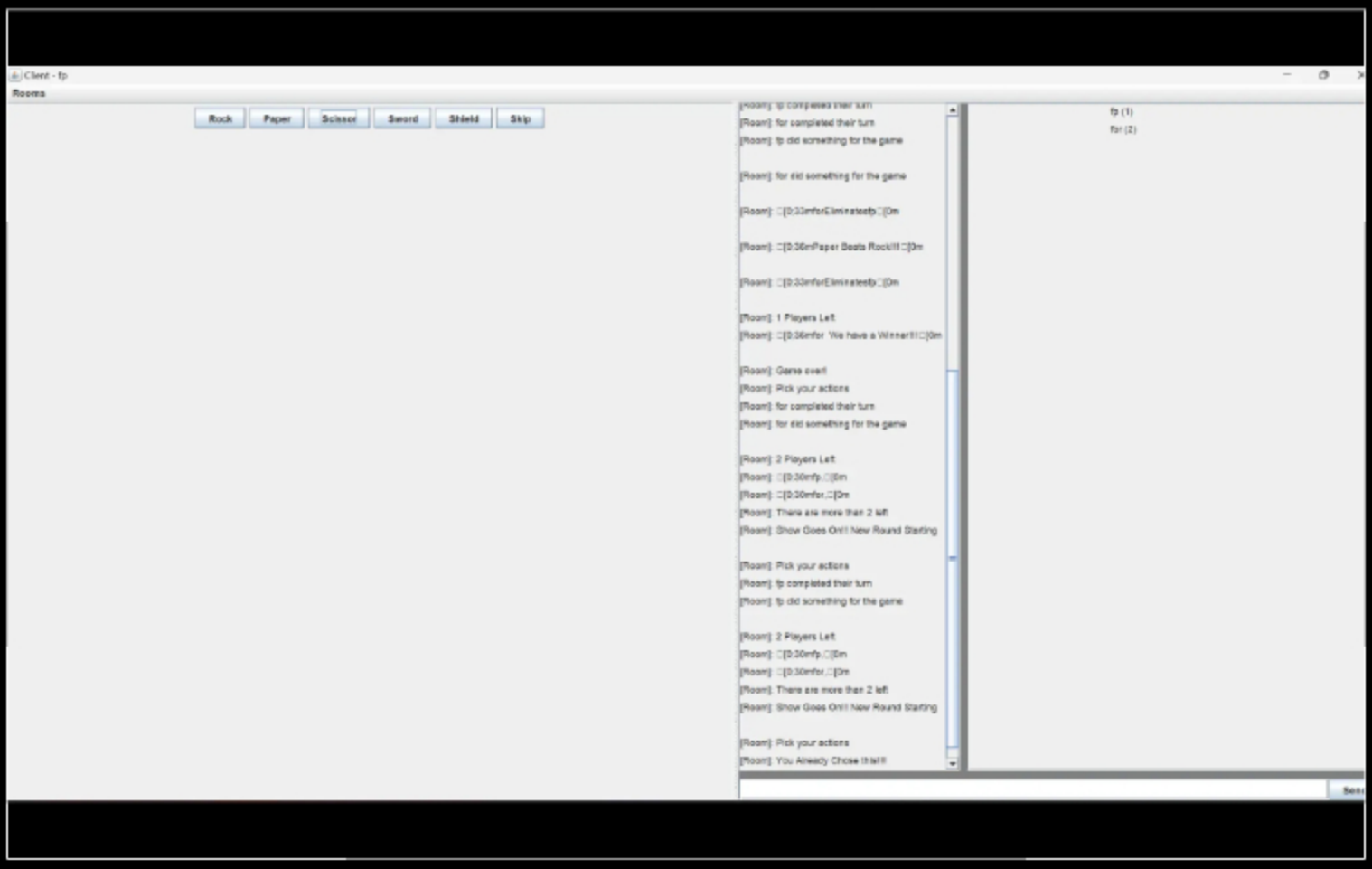
Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Show how the UI reflects a choice can't be chosen for the current round (i.e., UI elements, messages, etc)
#2	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small Medium Large



displays the option about the user already choosing the option.

Checklist Items (2)

#1 Show how the UI reflects a choice can't be chosen for the current round (i.e., UI elements, messages, etc)

#2 Each screenshot should be clearly captioned

COLLAPSE

Task #3 - Points: 1

Text: Explanation of Solution

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Mention how you handled the logic to add this limitation
#2	1	Mention the flow of the code across client-side and server-side
#3	1	Mention if you made this option toggleable or not

Response:

I handled the logic by creating a getprevious choice option inside the gameroom code. Once i have done that since the choices are already being given since the user has selected them. I made this option not toggleable due to save time and effiecency.

Client can mark themselves away (2.25 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of the code

Checklist

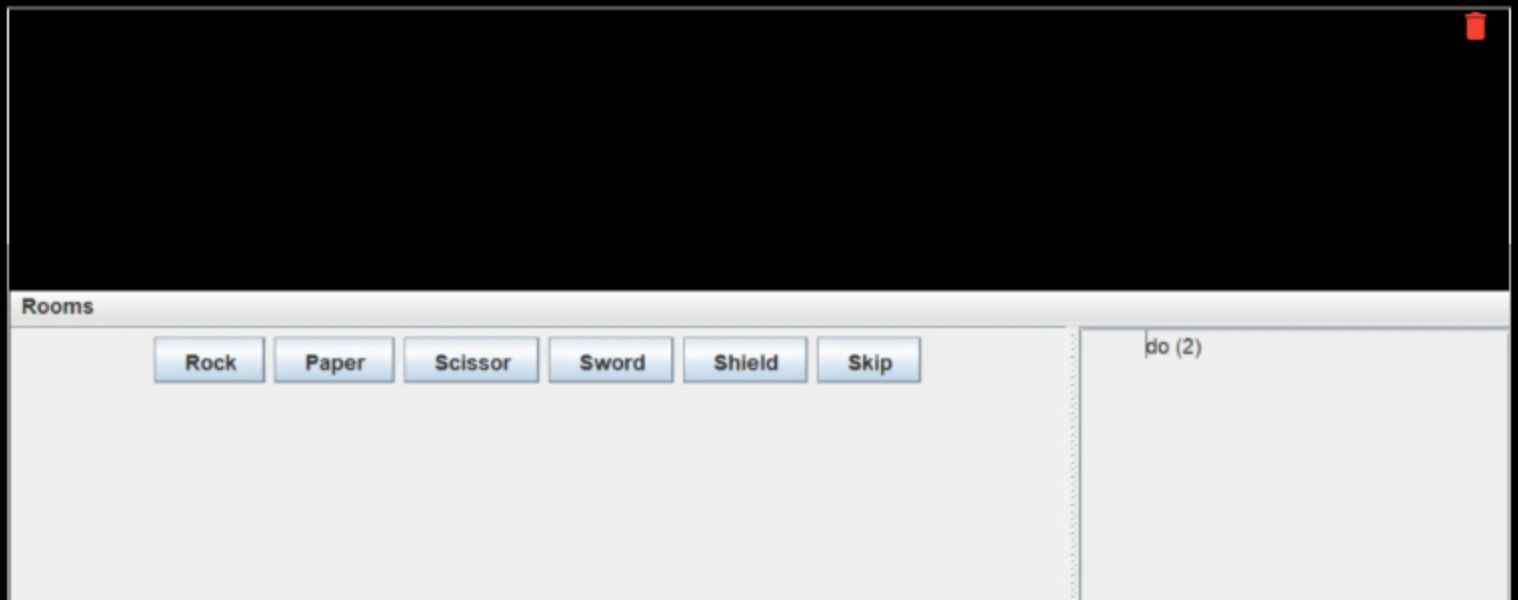
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Show how the client marks/unmarks away and the flow of the action
#2	1	Show how the "away" state is handled during the game events (i.e, blocking choosing answers from that person and similar)
#3	1	Show the code that updates the visual (UI) of a person away (i.e., can gray out their name in the list for example)
#4	1	Screenshots should include ucid and date comment
#5	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small Medium Large



SHows implementation of skip.

Checklist Items (0)

Implementation of the code and how it looks on the UI

Checklist Items (5)

#1 Show how the client marks/unmarks away and the flow of the action

#2 Show how the "away" state is handled during the game events (i.e, blocking choosing answers from that person and similar)

#3 Show the code that updates the visual (UI) of a person away (i.e., can gray out their name in the list for example)

#4 Screenshots should include ucid and date comment

#5 Each screenshot should be clearly captioned

COLLAPSE

Task #2 - Points: 1

Text: Screenshots of the demo

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Show the results of marking and unmarking away
#2	1	Show how away status is visualized and handled with game events
#3	1	Each screenshot should be clearly captioned

Task Screenshots:

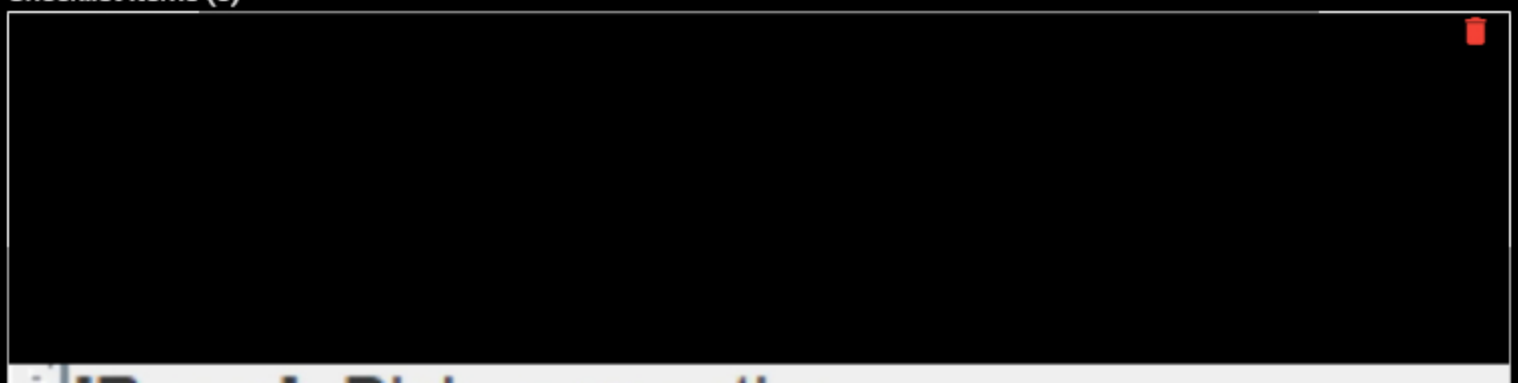
Gallery Style: Large View

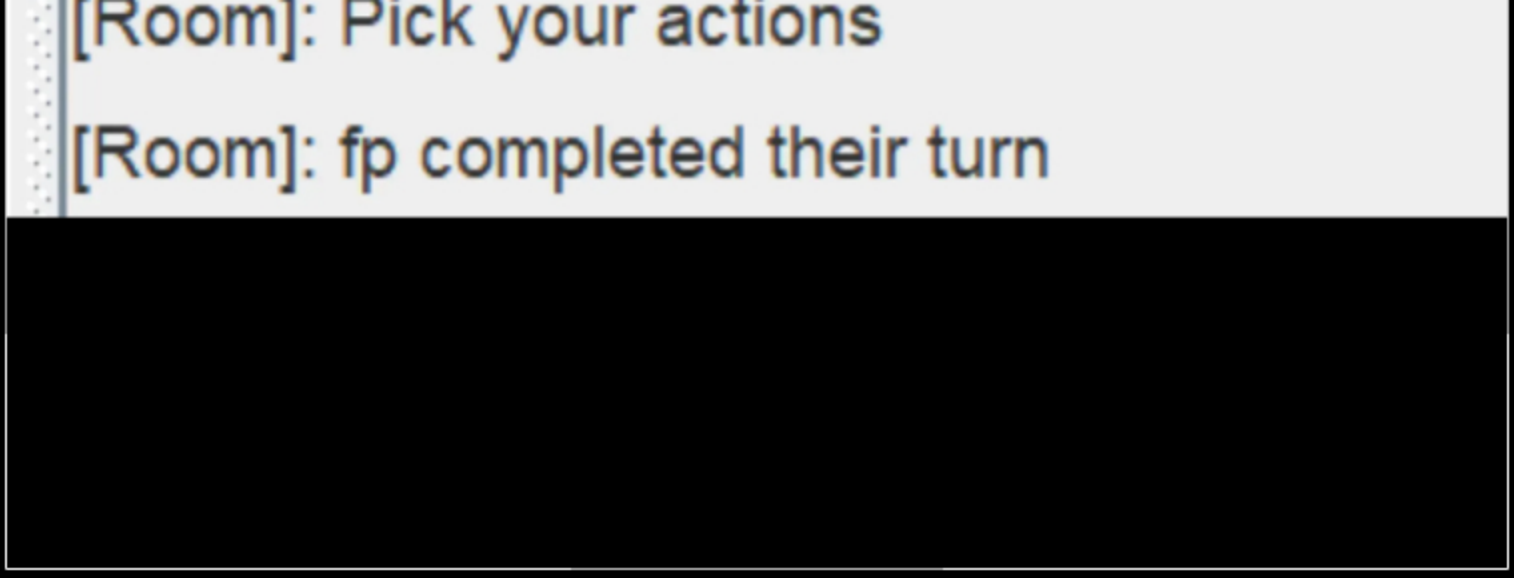
Small Medium Large

```
} <- #267-272 else if (p2.getChoice().equals("Sword")&& p1.getChoice().equa...
else if (p2.equals(obj:"Pass")) {
    sendMessage(ServerConstants.FROM_ROOM,TextFX.colorize(p1.getClientName() + "Has Skipped", Color.BLACK));
}
```

The code option to display user is away.

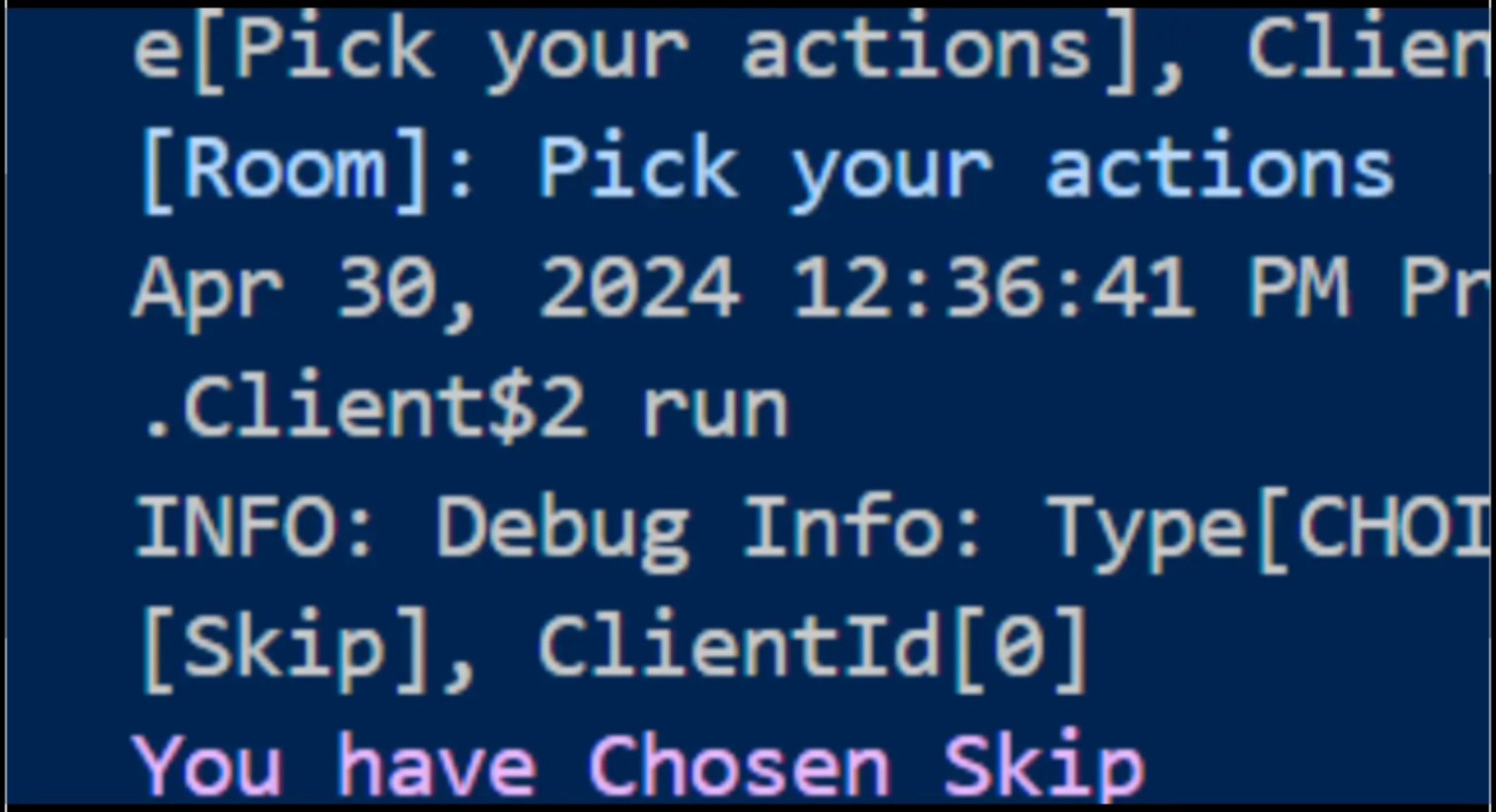
Checklist Items (0)





to display the individual has skipped their turn.

Checklist Items (0)



Displaying user

Checklist Items (3)

#1 Show the results of marking and unmarking away

#2 Show how away status is visualized and handled with game events

#3 Each screenshot should be clearly captioned

COLLAPSE

Task #3 - Points: 1

Text: Explain the solution

Checklist *The checkbox are for your own tracking

#	Points	Details
#1	1	Mention the flow/process of marking away
#2	1	Mention how it's utilized in the game events
#3	1	Mention how the UI reflects the state

Response:

The ui reflects the state when the user presses the skip. the process of it is when it is displayed in the game events that the user is the game and the game will end.

COLLAPSE

Spectators (2.25 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of the code

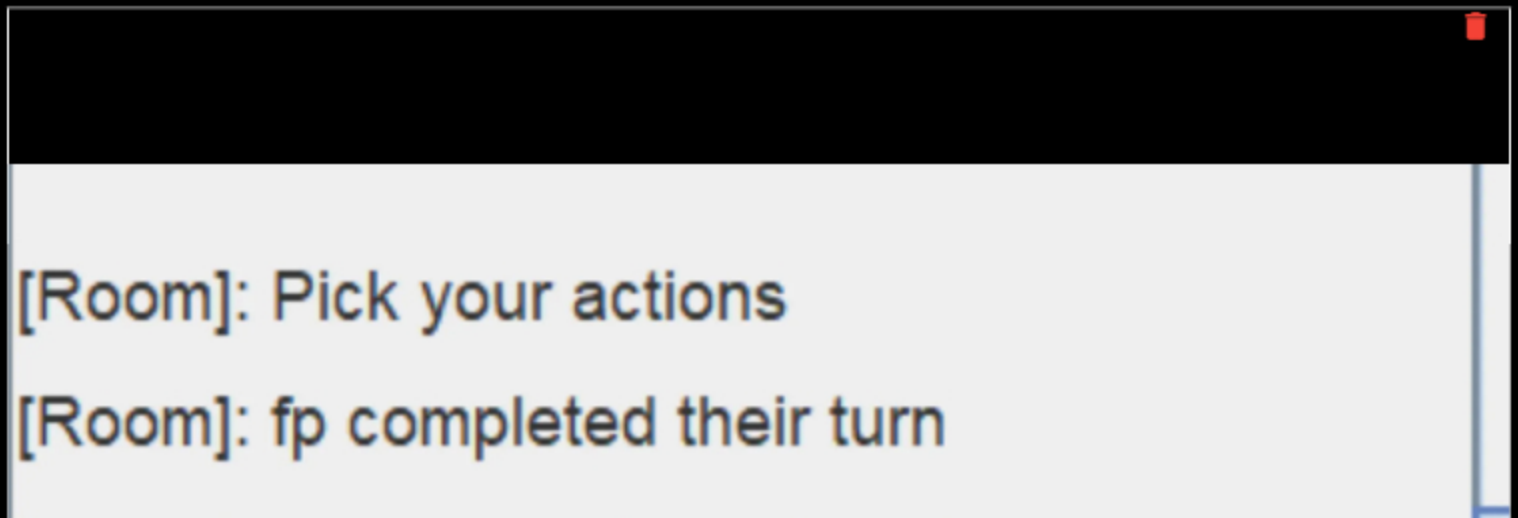
Checklist *The checkbox are for your own tracking

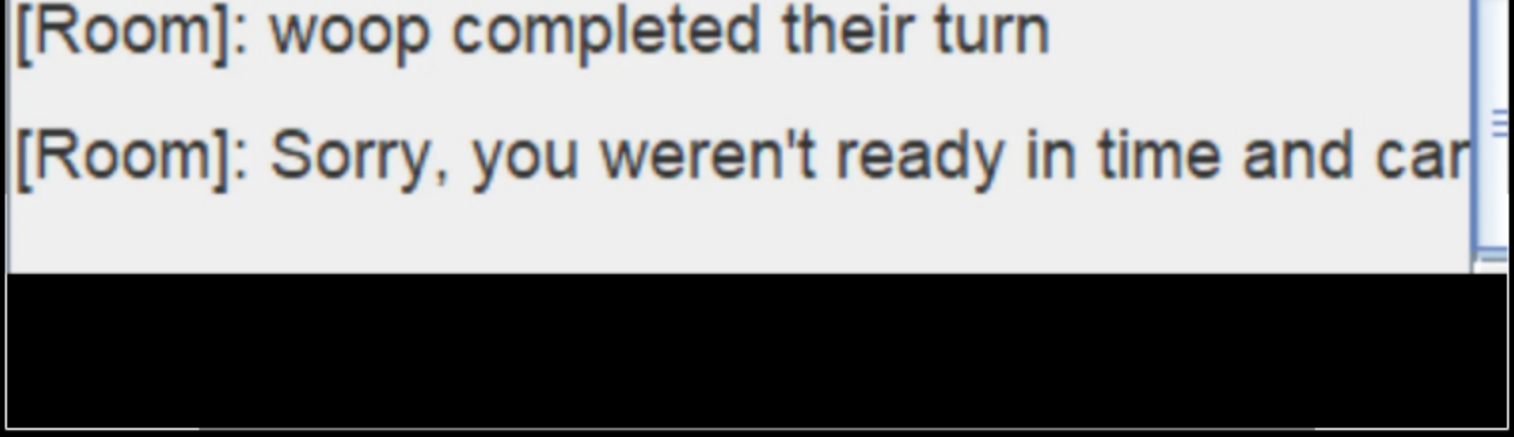
#	Points	Details
#1	1	Show how spectators are handled (including how they're ignored from game actions and game chat)
#2	1	Screenshots should include ucid and date comment
#3	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small Medium Large





displays to the user that they cannot participate

Checklist Items (0)

```
long clientId = client.getClientId();
if (players.containsKey(clientId)) {
    ServerPlayer sp = players.get(clientId);
    if (!sp.isReady()) {
        client.sendMessage(Constants.DEFAULT_CLIENT_ID, message:"Sorry, you weren't ready in time and can't participate");
        return;
    }
    //pd438 4/30/2024 Elimination Implementation
```

displays to the user that they cannot participate on time.

Checklist Items (3)

#1 Show how spectators are handled (including how they're ignored from game actions and game chat)

#2 Screenshots should include ucid and date comment

#3 Each screenshot should be clearly captioned

COLLAPSE

Task #2 - Points: 1

Text: Screenshots of the demo

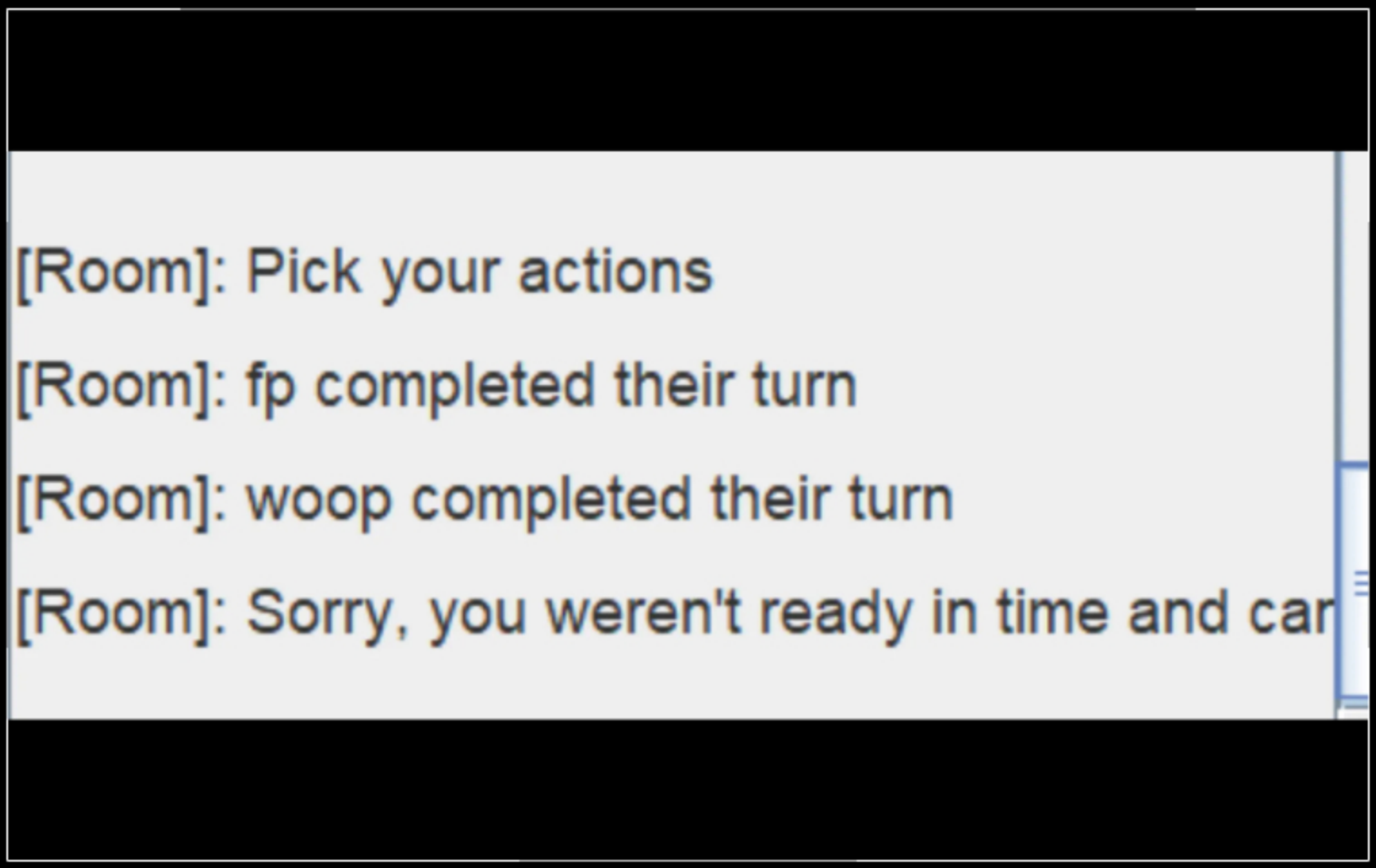
Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Show appropriate messages notifying the actions failed or can't be done
#2	1	The actions shouldn't be reflected in the game state (like normal user actions will appear)
#3	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Cant participate.

Checklist Items (3)

#1 Show appropriate messages notifying the actions failed or can't be done

#2 The actions shouldn't be reflected in the game state (like normal user actions will appear)

#3 Each screenshot should be clearly captioned

COLLAPSE

Task #3 - Points: 1

Text: Explain the solution

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Mention how spectators are determined
#2	1	Mention how they're handled regarding game actions and chat

Response:

Spectators are determined when they do not join the game on time. They are only able to chat when they spectate. But they cannot be able to play the game

Misc (1 pt.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Add the pull request link for the branch

Details:

Note: the link should end with /pull/#

URL #1

https://github.com/PD438/PD438_IT114_002/pull/15

COLLAPSE

Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

the one thing that i have an issue with in regards to the assignment. is to understand on how spectating would work. it does display that the user can work completely fine. But i def wish the logic was alot more understandable

COLLAPSE

Task #3 - Points: 1

Text: WakaTime Screenshot

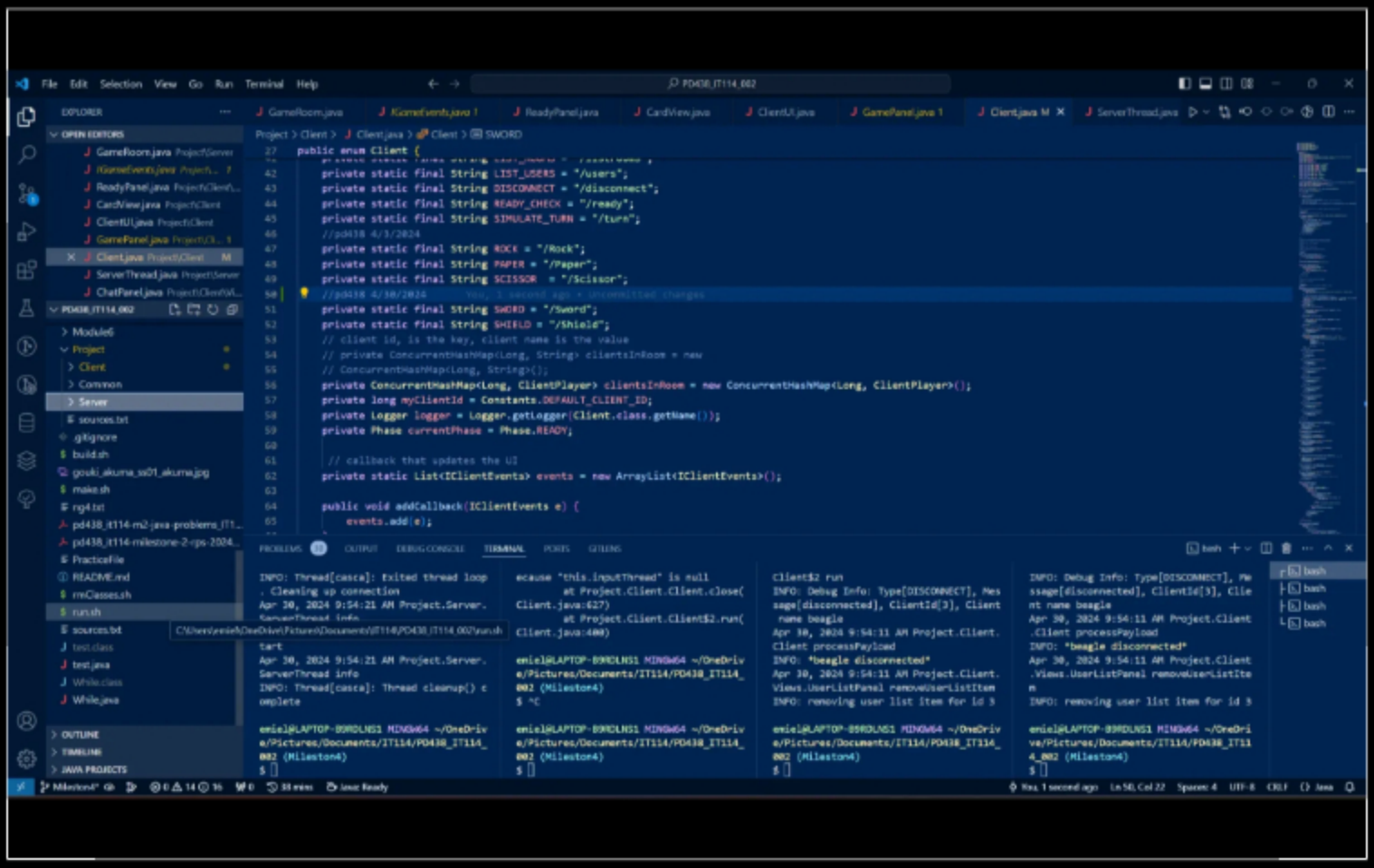
Details:

Grab a snippet showing the approximate time involved that clearly shows your repository. The duration isn't considered for grading, but there should be some time involved

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Wakatime and repository screenshot

End of Assignment