Course: IT114-010-S2025
Assignment: IT114 - Milestone 3 - RPS
Student: Hector M. (hem)
Status: Submitted | Worksheet Progress: 93%
Potential Grade: 9.33/10.00 (93.30%)
Received Grade: 0.00/10.00 (0.00%)
Grading Link: https://learn.ethereallab.app/assignment/v3/IT114-010-S2025/it114-milestone-3-rps/grading/hem

Instructions

1. Refer to Milestone3 of Rock Paper Scissors
    1. Complete the features
2. Ensure all code snippets include your ucid, date, and a brief description of what the code does
3. Switch to the `Milestone3` branch
    1. `git checkout Milestone3`
    2. `git pull origin Milestone3`
4. Fill out the below worksheet as you test/demo with 3+ clients in the same session
5. Once finished, click "Submit and Export"
6. Locally add the generated PDF to a folder of your choosing inside your repository folder and move it to Github
    1. `git add .`
    2. `git commit -m "adding PDF"`
    3. `git push origin Milestone3`
    4. On Github merge the pull request from `Milestone3` to `main`
7. Upload the same PDF to Canvas
8. Sync Local
    1. `git checkout main`
    2. `git pull origin main`

100%
Section #1: (1 pt.) Core Ui

100%
Task #1 (0.50 pts.) - Connection/Details Panels

Combo Task:

Weight: *50%*
Objective: *Connection/Details Panels*

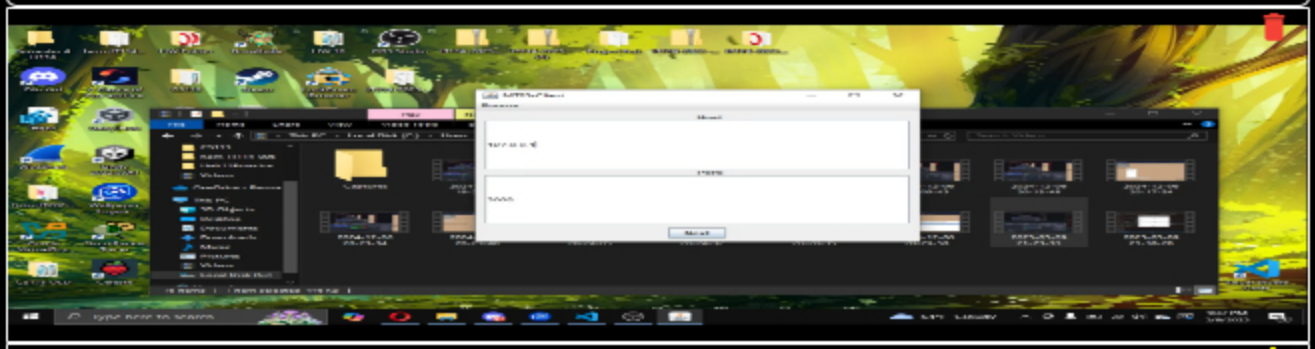
Objective: *Connection/Details Panels*

Image Prompt

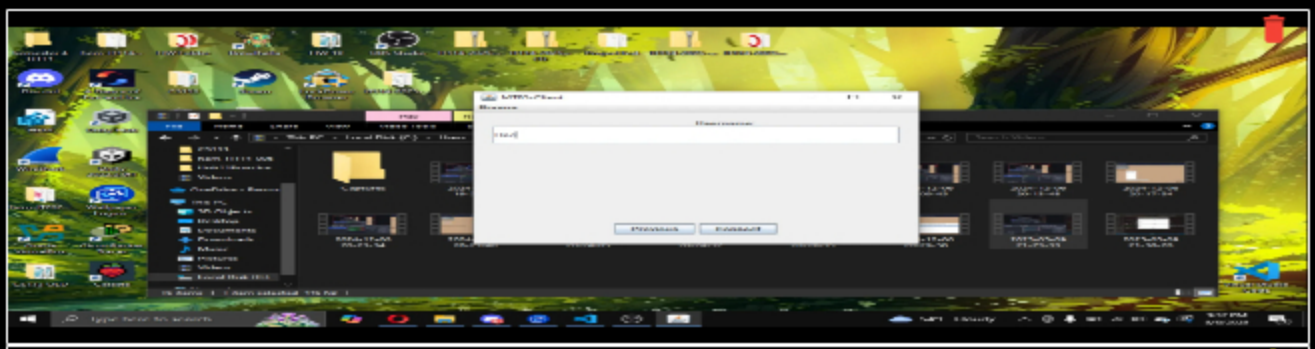
Weight: *50%*

Details:

- Show the connection panel with valid data
- Show the user details panel with valid data



Connection panel



User Panel

Saved: 5/9/2025 10:52:23 PM

Text Prompt

Weight: *50%*

Details:

- Briefly explain the code flow from recording/capturing these details and passing them through the connection process

Your Response:

The ConnectionPanel class provides a GUI for entering a host IP and port number, validating the port input, and advancing to the next screen if valid. The UserDetailsPanel class creates a screen for

entering a username, ensuring the field isn't empty before saving it and calling the connect command to continue. Both panels use Swing components and are part of a card-based navigation system managed through the ICardControls interface.

Saved: 5/9/2025 10:52:23 PM

100%

Task #2 (0.50 pts.) - Ready Panel

Combo Task:

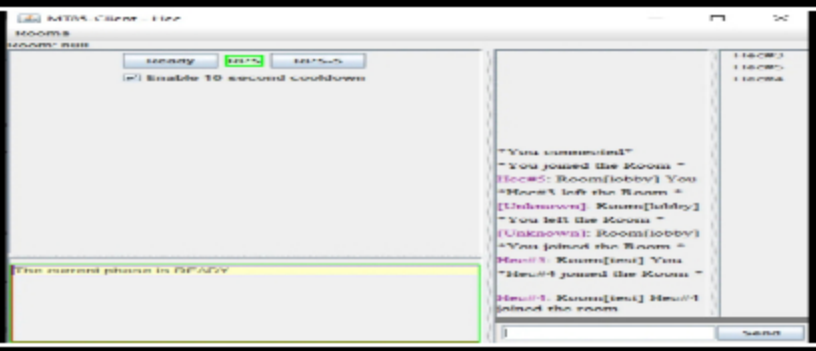
Weight: *50%*
Objective: *Ready Panel*

Image Prompt

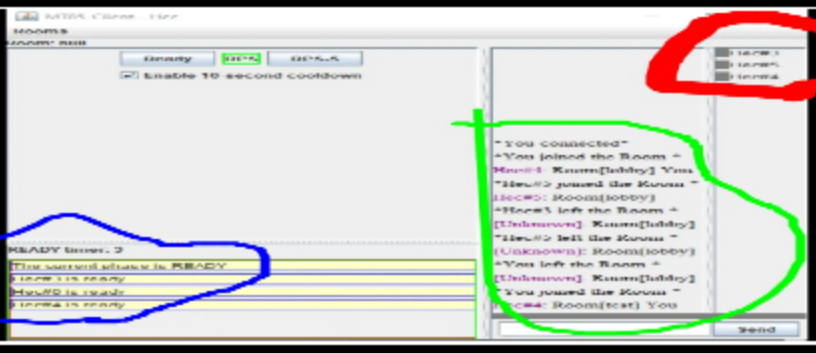
Weight: *50%*

Details:

- Show the button used to mark ready
- Show a few variations of indicators of clients being ready (3+ clients)



ready button panel



The indications of clients being ready are circled

Three clients connected

Saved: 5/11/2025 3:18:19 AM

Text Prompt

Weight: *50%*

Details:

- Briefly explain the code flow for marking READY from the UI
- Briefly explain the code flow from receiving READY data and updating the UI

Your Response:

The ready flow starts when a user clicks the ready button in the UI, triggering Client.sendReady which sends a ReadyPayload to the server. The server's processPayload handles the ready type, validating the player's status before calling handleReady, which toggles the ready state and broadcasts updates. On the UI side, the UserListPanel receives the update via onReceiveReady, which then visually updates the player's ready status and refreshes the sorted player list. This makes it so all clients see real-time changes while maintaining game state's consistency.

Saved: 5/11/2025 3:18:19 AM

66%

Section #2: (2 pts.) Project Ui

25%

Task #1 (0.67 pts.) - User List Panel

Combo Task:

Weight: *33.33%*
Objective: *User List Panel*

Details:

- Show the username and id of each user
- Show the current points of each user
- Users should appear in score order, sub-sort by name when ties occur
- Pending-to-pick users should be marked accordingly
- Eliminated users should be marked accordingly

Image Prompt

Weight: *50%*

Details:

- Show various examples of points (3+ clients visible)
  - Include code snippets showing the code flow for this from server-side to UI
- Show that the sorting is maintained across clients
  - Include code snippets showing the code that handles this
- Show various examples of the pending-to-pick indicators
  - Include code snippets showing the code flow for this from server-side to UI
- Show various examples of elimination indicators
  - Include code snippets showing the code flow for this from server-side to UI

3 clients connected

Missing Caption

Saved: 5/11/2025 3:19:10 AM

Text Prompt

Weight: *50%*

Details:

- Briefly explain the code flow for points updates from server-side to the UI
- Briefly explain the code flow for user list sorting
- Briefly explain the code flow for server-side to UI of pending-to-pick indicators
- Briefly explain the code flow for server-side to UI of elimination indicators

Your Response:

Missing Response

Saved: 5/11/2025 3:19:10 AM

75% Task #2 (0.67 pts.) - Game Events Panel

Combo Task:

Weight: *33.33%*
Objective: *Game Events Panel*

Details:

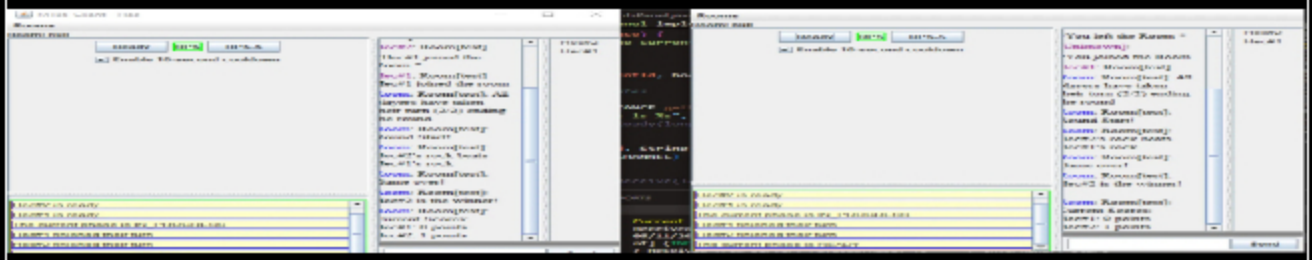
- Show the status of users picking choices
- Show the battle resolution messages from Milestone 2
  - Include messages about elimination
- Show the countdown timer for the round

Image Prompt

Weight: *50%*

Details:

- Show various examples of each of the messages/visuals
- Show code snippets related to these messages from server-side to UI



game events panel showing off what happens in a game

Missing Caption

Saved: 5/11/2025 3:33:32 AM

Text Prompt

Weight: *50%*

Details:

- Briefly explain the code flow for generating these messages and getting them onto the UI

Your Response:

The GameEventsPanel initializes with a panel using GridBagLayout to display messages. Whenever addText is called, it creates a non-editable JEditorPane, adds it above a vertical glue component to pin messages to the top, and makes it scroll to the latest message. The glue is temporarily removed and reinserted during updates to maintain layout consistency, while revalidate and repaint make the UI refresh. Timer updates and event notifications are routed through this system via interface callbacks.

Saved: 5/11/2025 3:33:32 AM

Task #3 (0.67 pts.) - Game Area

Combo Task:

Weight: *33.33%*
Objective: *Game Area*

Details:

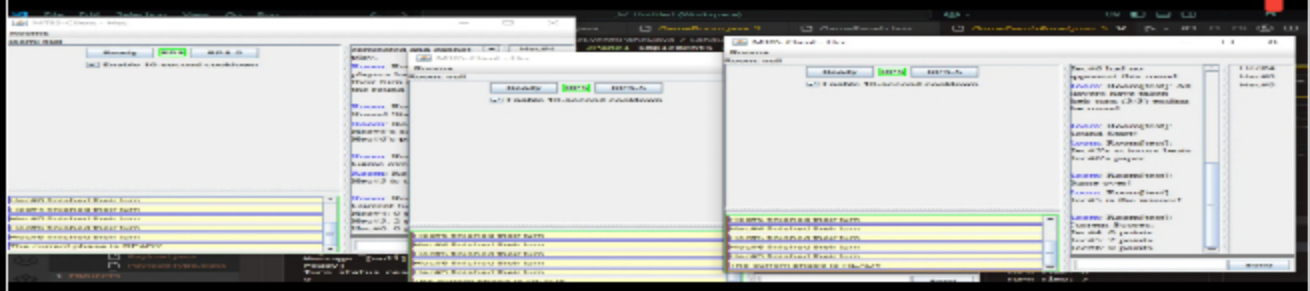
- UI should have components to allow the user to select their choice

Image Prompt

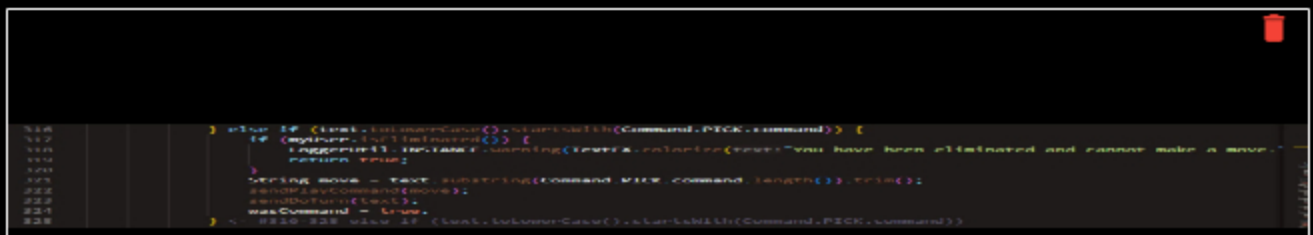
Weight: *50%*

Details:

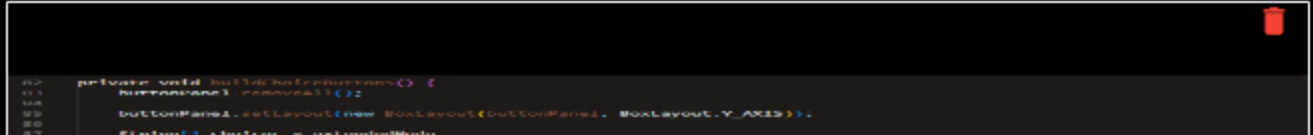
- Show various examples of selections across clients (3+ clients visible)
- Show the code related to sending choices upon selection
- Show the code related to showing visually what was selected



3 clients after a game. choices made can be seen in the game events panel



the process payload method for my players choice



how it is visually represented in game panel through buttons

Saved: 5/11/2025 5:13:43 AM

Text Prompt

Weight: *50%*

Details:

- Briefly explain the code flow for selecting a choice and having it reach the server-side
- Briefly explain the code flow for receiving the selection for the current player to update the UI

Your Response:

When a player clicks a choice button (like Rock, Paper, etc.) in the `GamePanel`, the `buildChoiceButtons()` method links that button to `Client.INSTANCE.sendDoTurn()`, which sends the selected move to the server. The server then processes this move as part of the game logic. Once the server updates the game phase (like switching to READY or IN_PROGRESS), it sends that information back to the client. The `onReceivePhase()` method handles this by switching the visible panel—showing either the ready screen or the play screen—and updates the UI accordingly, such as enabling or hiding the choice buttons based on the phase.

Saved: 5/11/2025 5:13:43 AM

100%

Section #3: (4 pts.) Project Extra Features

100%

Task #1 (2 pts.) - Extra Choices

Combo Task:

Weight: *50%*

Objective: *Extra Choices*

Details:

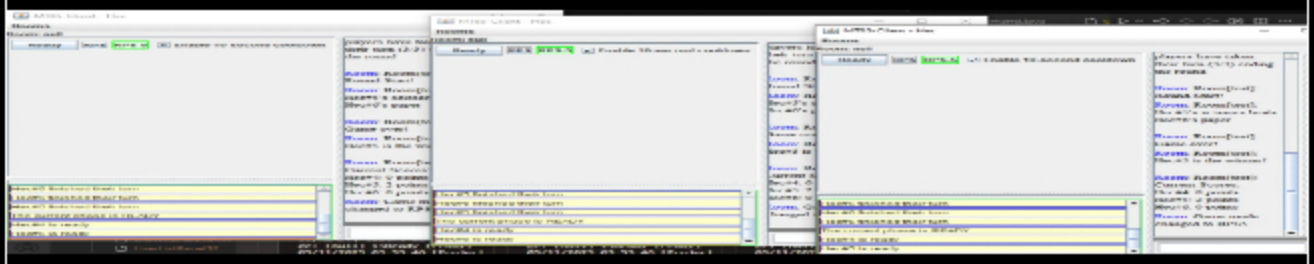
- Setting should be toggleable during Ready Check by session creator
  - (Option 1) Extra choices are available during the full session
  - (Option 2) Only activate extra options at different stages (i.e., last 3 players remaining)
- There should be at least 2 extra options for rps-5

Image Prompt

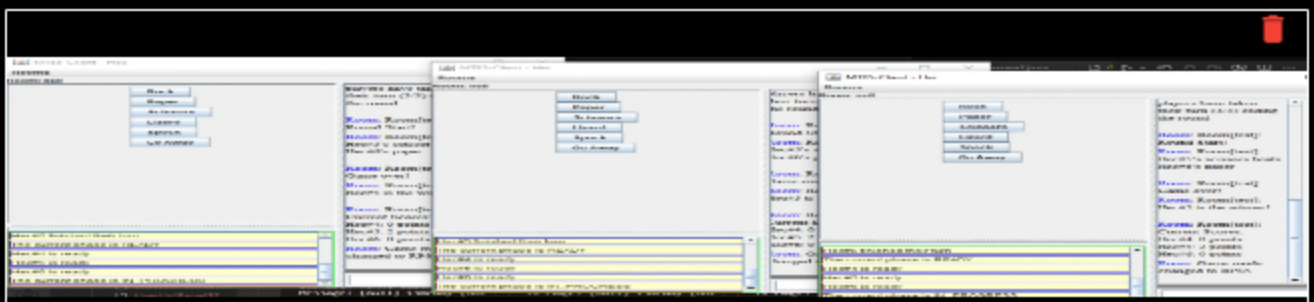
Weight: *50%*

Details:

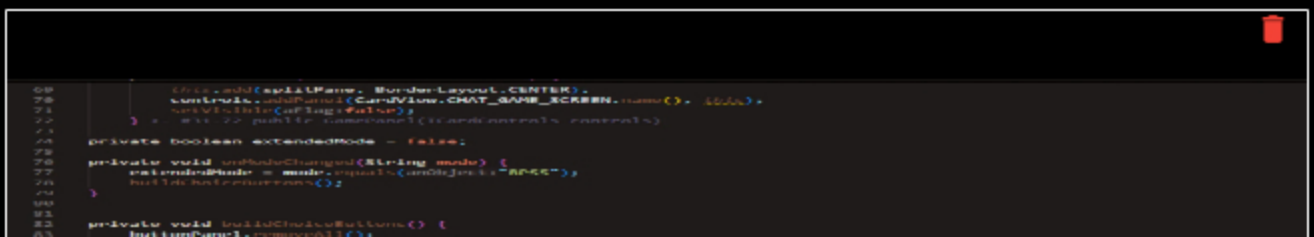
- Show the Ready Check screen with the option for the host (3+ clients must be visible)
  - Show the related code that makes this interactable only for the host
- Show the play screen with the extra options available
  - Show the related code for the UI and handling of these extra options (including battle logic)

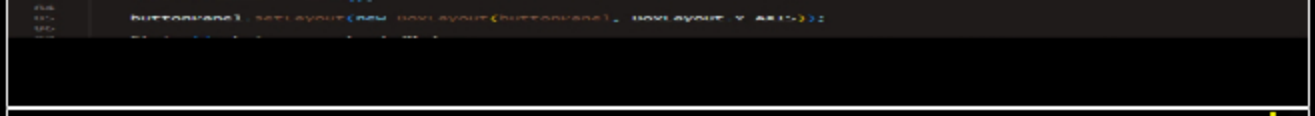


The option for RPS5



All 5 options available





My game UI method that updates the UI with new options if the game mode is changed

My game room logic that uses my Gamemode.java file to check for what game mode the user set the game to

Saved: 5/11/2025 4:06:39 AM

Text Prompt

Weight: *50%*

Details:

- Briefly explain the code for the host's option to toggle this feature
- Briefly explain the code related to handling these options including how it's handled during the battle logic
- Note which option you went with in terms of activating the choices

Your Response:

100% Task #2 (2 pts.) - Choice cooldown

Combo Task:

Weight: *50%*
Objective: *Choice cooldown*

Details:

- Setting should be toggleable during Ready Check by session creator
- The choice on cooldown must be disable on the UI for the User

Image Prompt

Weight: *50%*

Details:

- Show the Ready Check screen with the option for the host (3+ clients must be visible)
  - Show the related code that makes this interactable only for the host
- Show a few examples of the play screen with the choice on cooldown
  - Show the related code for the UI and handling of the cooldown and server-side enforcing it

enable cool down toggleable button on UI

Saved: 5/11/2025 3:45:20 AM

Text Prompt

Weight: *50%*

Details:

- Briefly explain the code for the host's option to toggle this feature
- Briefly explain the code related to handling and enforcing the cooldown period (include how this is recorded per user and reset when applicable)

Your Response:

I couldn't get this feature to work properly I was only able to get a check box on my UI to be toggled on and off. I misunderstood the prompt and made a timed cooldown and it just does work right

Saved: 5/11/2025 3:45:20 AM

100%

Section #4: (2 pts) Project General

Section #4: (2 pts.) Project General Requirements

100% Task #1 (1 pt.) - Away Status

Combo Task:

Weight: *50%*
Objective: *Away Status*

Details:

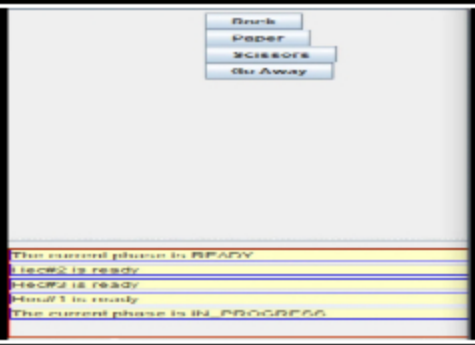
- Clients can mark themselves away and be skipped in turn flow but still part of the game
- The status should be visible to all participants
- A message should be relayed to the Game Events Panel (i.e., Bob is away or Bob is no longer away)
- The user list should have a visual representation (i.e., grayed out or similar)

Image Prompt

Weight: *50%*

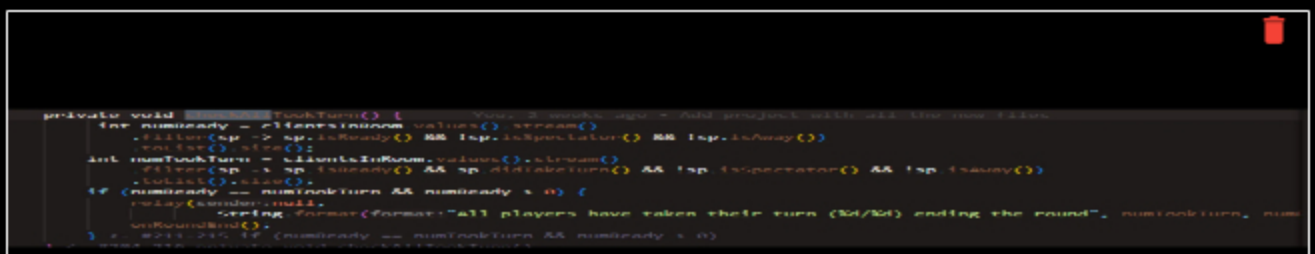
Details:

- Show the UI button to toggle away
- Show the related code flow from UI to server-side back to UI for showing the status
- Show the related code flow for sending the message to Game Events Panel
- Show various examples across 3+ clients of away status (including Game Events Panel messages)
- Show the code that ignores an away user from turn/round logic



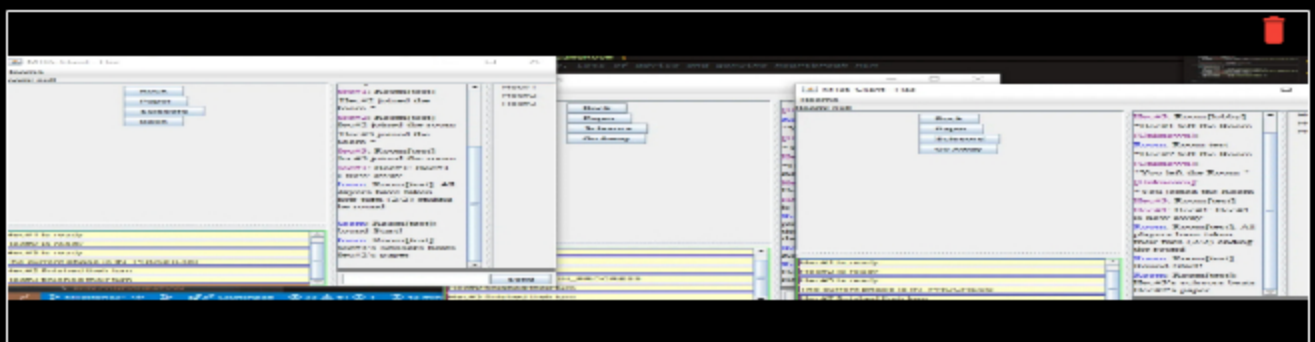
The away button in the UI

The code to make the button on the UI

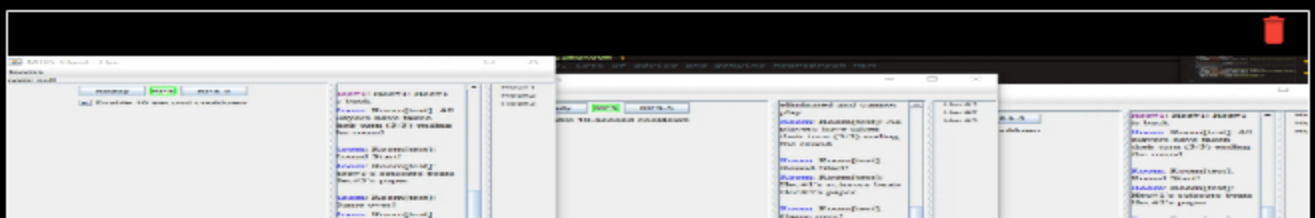


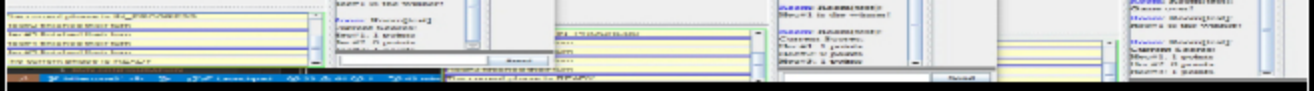
checkalltookturn modifications to ignore away players

the filtering I have set up in round so that away players are excluded from the total clients



away player being skipped over by the round logic and the two active players being able to duel it out





away player coming back and fighting last rounds winner

100% Task #2 (1 pt.) - Spectators

Combo Task:

Weight: *50%*
Objective: *Spectators*

Details:

- Spectators are users who didn't mark themselves ready
  - Optionally you can include a toggle on the Ready Check page
- The can see all chat but are ignored from turn/round actions and can't send messages
- Spectators will have a visual representation in the user list to distinguish them from other players
- A message should be relayed to the Game Events Panel that a spectator joined (i.e., during an in-progress session)

Image Prompt

Weight: *50%*

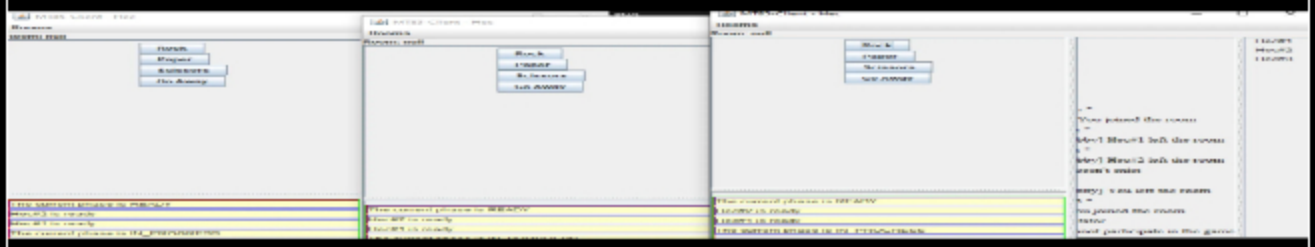
Details:

- Show the UI indicator of a spectator (visual and message)
- Show the related code flow from UI to server-side back to UI for showing the status
- Show the related code flow for sending the message to Game Events Panel
- Show various examples across 3+ clients of spectator status (including Game Events Panel messages)
- Show the code that ignores a spectator from turn/round logic
- Show the code that prevents spectators from sending messages (server-side)
- Show the spectator's view of the session
- Show the code related to the spectator seeing the session data (including things participants won't see)

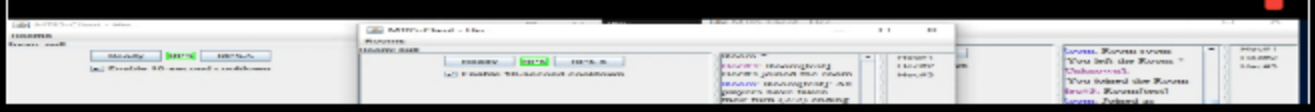
specattors being filtered out in round logic

check turn logic handling spectators

Stops a spectator from messaging



spectator joined but as show in game panel could not take a turn



100%

Section #5: (1 pt.) Misc

Task #1 (0.33 pts.) - Github Details

Combo Task:

Weight: *33.33%*
Objective: *Github Details*

Image Prompt

Weight: *60%*

Details:
From the Commits tab of the Pull Request screenshot the commit history

commits

Saved: 5/11/2025 5:54:54 AM

Url Prompt

Weight: *40%*

Details:
Include the link to the Pull Request for Milestone3 to main (should end in `/pull/#`)

URL #1
https://github.com/Hemogoblin04/hem-IT114-006/compare/main...Milestone3

URL
https://github.com/Hemogoblin04/

Saved: 5/11/2025 5:54:54 AM

Task #2 (0.33 pts.) - WakaTime - Activity

Image Prompt

Weight: *33.33%*
Objective: *WakaTime - Activity*

Details:

- Visit the WakaTime.com Dashboard
- Click `Projects` and find your repository
- Capture the overall time at the top that includes the repository name
- Capture the individual time at the bottom that includes the file time
- Note: The duration isn't relevant for the grade and the visual graphs aren't necessary

wakatime

Saved: 5/11/2025 4:07:28 AM

Task #3 (0.33 pts.) - Reflection

Weight: *33.33%*
Objective: *Reflection*

Sub-Tasks:

Task #1 (0.33 pts.) - What did you learn?

Text Prompt

Weight: *33.33%*
Objective: *What did you learn?*

Details:
Briefly answer the question (at least a few decent sentences)

Your Response:

I learned a lot about designing UI, handling payloads and among other things. The biggest thing I learned was the limits of human sanity. Jokes aside I also learned a lot about tracking cross files especially when desiging the buttons to switch gamemodes.

Saved: 5/9/2025 10:27:54 PM

100%

Task #2 (0.33 pts.) - What was the easiest part of the assignr

Text Prompt

Weight: *33.33%*
Objective: *What was the easiest part of the assignment?*

Details:
Briefly answer the question (at least a few decent sentences)

Your Response:

Quite literally nothing was, the assignment was really tedious. The ammount of new features needed to be added while working with UI we just learned to do was hard. the only somewhat easy part was designing the UI, but getting things right where I wanted was still extremely tedious.

Saved: 5/11/2025 4:09:19 AM

100%

Task #3 (0.33 pts.) - What was the hardest part of the assignı

Text Prompt

Weight: *33.33%*
Objective: *What was the hardest part of the assignment?*

Details:

Briefly answer the question (at least a few decent sentences)

Your Response:

It was figuring out how to make away and spectator work. They made no sense to me how I would make it so people can be skipped or ignored by the game logic. My logic was tracking everybody in the room and it was so hard coded to do so, excluding people would just break it. Other then that the implimentation of RPS5 wasnt hard in it of itself, but the handling of how to change the game logic so that it can process 5 options instead of 3 was challenging

Saved: 5/9/2025 10:27:10 PM