

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT114-003-F2024/it114-milestone-3-rps-2024-m24/grade/jns>

Course: IT114-003-F2024

Assignment: [IT114] Milestone 3 RPS 2024 M24

Student: Jimmy S. (jns)

Submissions:

Submission Selection

1 Submission [submitted] 11/26/2024 12:32:50 PM ▾

Instructions

▲ COLLAPSE ▾

Implement the Milestone 3 features from the project's proposal document: https://docs.google.com/document/d/11SRMo7JkLAMM-PuuiGwl_Z-QXP3pyQ7xN3IRxwmcwCc/view Make sure you add your ucid/date as code comments where code changes are done All code changes should reach the Milestone3 branch Create a pull request from Milestone3 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: Milestone3

Group



Group: Basic UI

Tasks: 1

Points: 2

▲ COLLAPSE ▾

Task



Group: Basic UI

Task #1: UI Panels

Weight: ~100%

Points: ~2.00

[COLLAPSE](#)

i Details:

All code screenshots must include ucid/date.

App screenshots must have the UCID in the title bar like the lesson gave.



Columns: 1

Sub-Task

100%

Group: Basic UI

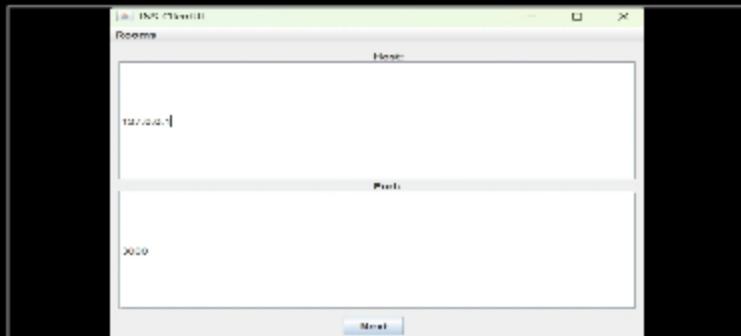
Task #1: UI Panels

Sub Task #1: Show the ConnectionPanel by running the app (should have host/port)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



connection panel

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

100%

Group: Basic UI

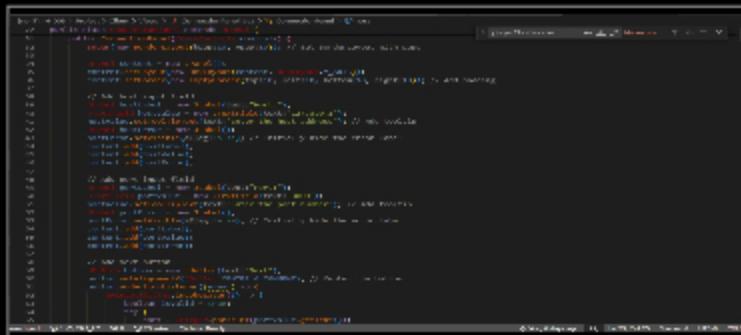
Task #1: UI Panels

Sub Task #2: Show the code related to the ConnectionPanel

Task Screenshots

Gallery Style: 2 Columns

4 2 1



connectionpanel in project/client/views

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain how it works and how it's used

Response:

this is the first panel that shows when the game is initially started, asks for connection information and name, in which it will then attempt to connect to a server with that provided information.

Sub-Task

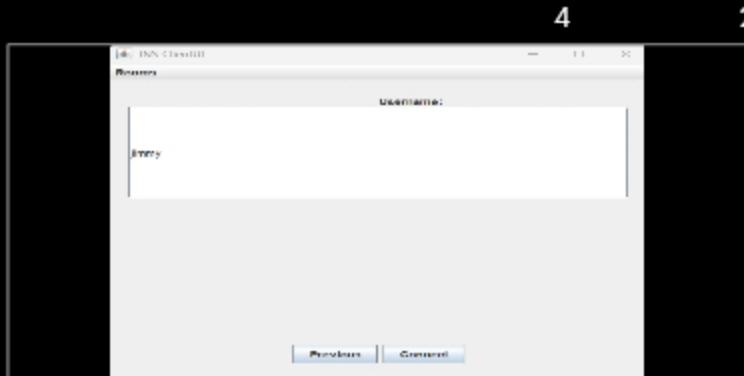
Group: Basic UI

Task #1: UI Panels

Sub Task #3: Show the UserDetailsPanel by running the app (should have username)

Task Screenshots

Gallery Style: 2 Columns



user details panel

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

Group: Basic UI

Task #1: UI Panels

Sub Task #4: Show the code related to the UserDetailsPanel

Task Screenshots

Gallery Style: 2 Columns

user details code, pulled and merged into my code from

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain how it works and how it's used

Response:

this is merged from the starter code from milestone tree, creates panel for username after user enters connection information, prompts the user for a username before they attempt to connect to the server.

End of Task 1

End of Group: Basic UI

Task Status: 1/1

Group



Group: Game Area

Tasks: 5

Points: 7

▲ COLLAPSE ▲

Task



Group: Game Area

Task #1: ReadyCheck UI Panel

Weight: ~20%

Points: ~1.40

▲ COLLAPSE ▲

ⓘ Details:

All code screenshots must include ucid/date.

App screenshots must have the UCID in the title bar like the lesson gave.



Columns: 1

Sub-Task

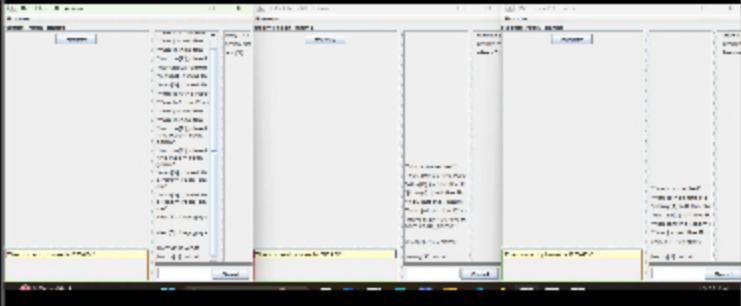
Group: Game Area

Task #1: ReadyCheck UI Panel

Sub Task #1: Show the screen with the ready panel open in a fresh session

Task Screenshots

Gallery Style: 2 Columns



3 players in game lobby, with option to ready up

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task



Group: Game Area

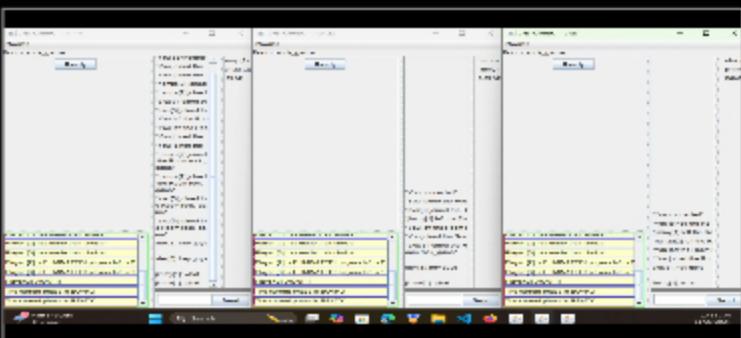
Task #1: ReadyCheck UI Panel

Sub Task #2: Show the screen with the ready panel open after a session ends (there should be output in other parts of the UI showing this)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



session has occurred, values are reset, and players are sent back to the ready panel when the phase is changed back to ready.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

End of Task 1

Task



Group: Game Area

Task #2: User List

Weight: ~20%

Points: ~1.40

COLLAPSE

i Details:

All code screenshots must include ucid/date.

App screenshots must have the UCID in the title bar like the lesson gave.

- Show the username and id of each Player

Columns: 1

Sub-Task

Group: Game Area

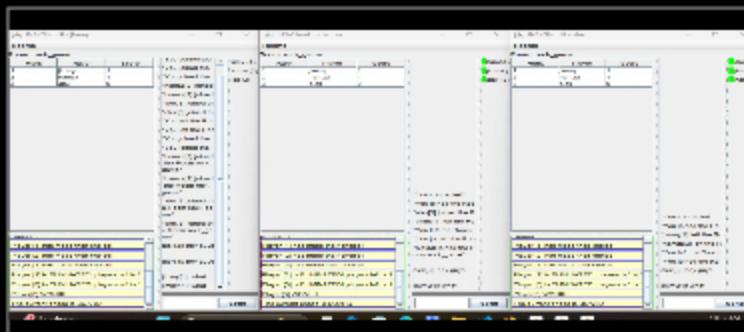
Task #2: User List

Sub Task #1: Show the username/id of each player, current points, pending-to-pick indicator, eliminated-indicator (list should appear in score order across all clients) (show a few examples)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



leaderboard, displays after round end, and the game enters board phase for 10 seconds before returning back to ready.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

Group: Game Area

Task #2: User List

Sub-Task #2: Show the related code (from server-side to UI) that marks the user list item properly.

Task Screenshots

Gallery Style: 2 Columns

4 2 1

`sendReady` in client, goes to serverthread and tells room that user is ready, and start ready timer.

case handler in serverthread, will only work if user is game room, was more important in milestone two, but with buttons now ...

```
    // execute data tasks sequentially (not in parallel)
    protected void handleDelayed(HandlerThread sender) { ... }
    protected void handleNormal(HandlerThread sender) {
        try {
            // early exit checks
            checkLayerIncons(sender);
            checkCurrentPhase(sender, Phase.NORMAL);
            ...
        } catch (Exception e) {
            Log.e("Finsky", "Error in normal task", e);
        }
    }
    ...
    protected void handlePriority(HandlerThread sender) {
        try {
            ...
        } catch (Exception e) {
            Log.e("Finsky", "Error in priority task", e);
        }
    }
}
```

```
You, 6 days ago | 1 author (You)
public class ReadyPanel extends JPanel {
    public ReadyPanel() {
        JButton readyButton = new JButton();
        readyButton.setText("Ready");
        readyButton.addActionListener(l_ -> {
            ...
        });
    }
}
```

handler of ready in base game room, toggles ready and starts ready timer + send ready status for other people to see.

ready panel, creates ready button, and adds listener for when user clicks on button, which sends to sendReady in client.

`userlistpanel`, when receives `setturn` (should be named better) calls other class, and repaints tile next to name green

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

explained in each caption of the images. but basically, user clicks ready button, sendsReady in client, goes to serverthread, gameroom, starts ready timer, and in userlistpanel, marks tile next to username and id green in both the players ui and others players, as well as sending a ready message in the game terminal.

End of Task 2

Task

Group: Game Area

Task #3: GameEventPanel

Weight: ~20%

Points: ~1.40

COLLAPSE

• i Details:

All code screenshots must include ucid/date.
App screenshots must have the UCID in the title bar like the lesson gave.

Columns: 1

Sub-Task

Group: Game Area

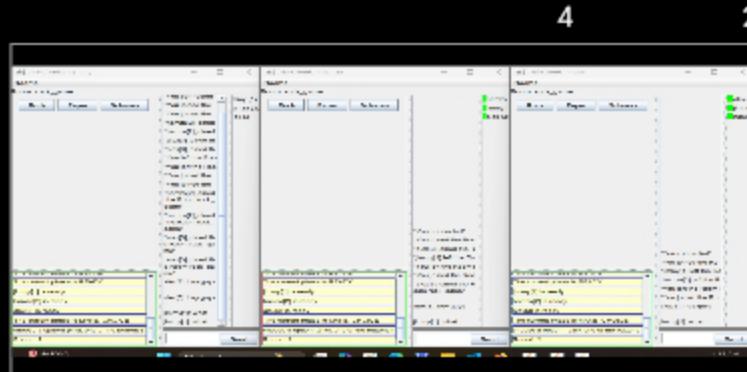
100%

Task #3: GameEventPanel

Sub Task #1: Show the picking choices and battle log messages from Milestone 2

Task Screenshots

Gallery Style: 2 Columns



make choice panel, user are able to make choices and the terminal below tells when user gets message.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Note how the battle win/lose is determined/handled

Response:

handled A LOT BETTER then in milestone 2, there was a lot of issues with the logic before, mostly from me not ordered the playerchoices list in some way, bc concurrenthashmaps have no ordering, but now they are ordered by client ID and also calculating logic is also correct now. Additionally, players are removed from choices list bf reset, but instead isAlive value is the new conditional check.

Sub-Task

Group: Game Area

100%

Task #3: GameEventPanel

Sub Task #2: Show the code for the UI flow (Client receiving to UI) for each example

Task Screenshots

Gallery Style: 2 Columns

```

private void handlePlayerChoices() {
    List<String> choices = new ArrayList<String>();
    for (Client client : clients) {
        if (client.isAlive()) {
            choices.add(client.getName());
        }
    }
    Collections.sort(choices);
    for (Client client : clients) {
        if (client.isAlive()) {
            client.setChoices(choices);
        }
    }
}

```

```

private void handleBattleLog() {
    String log = "Battle Log:\n";
    for (Client client : clients) {
        if (client.isAlive()) {
            log += client.getName() + " is alive.\n";
        } else {
            log += client.getName() + " has died.\n";
        }
    }
    System.out.println(log);
}

```

game panel code, merged with starter code, additional code more on game panel, added for leaderboard information, and JTable.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

merged with started, updates based of phase information from gameroom. Additionally add section for board phase, in which it will display leaderboard information for all client through leaderboardPayload and leaderboardRecords.

End of Task 3

Task

Group: Game Area

Task #4: Choices Area

Weight: ~20%

Points: ~1.40

COLLAPSE

● Details:

All code screenshots must include ucid/date.

App screenshots must have the UCID in the title bar like the lesson gave.

二

Columns: 1

Sub-Task

Group: Game Area

Task #4: Choices Area

Sub Task #1: Show the UI representing the possible choices

Task Screenshots

Gallery Style: 2 Columns

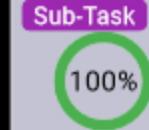
4 2 1



game panel for make choice phase

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

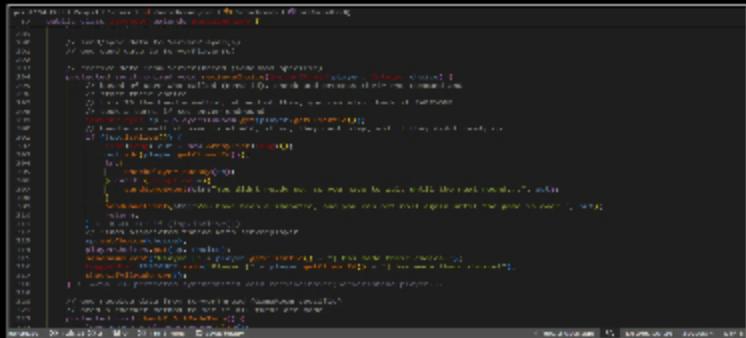


Group: Game Area
Task #4: Choices Area
Sub Task #2: Show the code related to these buttons and their interaction

Task Screenshots

Gallery Style: 2 Columns

4 2 1



receives choice in gameroom

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

mostly same from milestone, instead, each button in ui sendChoice to gameroom, which receive choice and continue game logic process.

End of Task 4

Task



Group: Game Area

Task #5: Countdown Timer UI

Weight: ~20%

Points: ~1.40

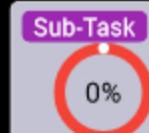
COLLAPSE

● Details:

All code screenshots must include ucid/date.

App screenshots must have the UCID in the title bar like the lesson gave.

Columns: 1



Group: Game Area

Task #5: Countdown Timer UI

Sub Task #1: Show the UI of the countdown (few examples to show it changes)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



Missing Caption

Caption(s) (required)

Caption Hint: *Describe/highlight what's being shown*

Missing caption(s)

Sub-Task



Group: Game Area

Task #5: Countdown Timer UI

Sub Task #2: Show the code related to managing the timer

Task Screenshots

Gallery Style: 2 Columns

4 2 1



Missing Caption

Caption(s) (required)

Caption Hint: *Describe/highlight what's being shown*

Missing caption(s)

Task Response Prompt

Explain in concise steps how this logically works, also note if you're doing two separate timers or just syncing the ticks (or something else)

Response:

don't have this implemented... timer shows in terminal, not in game panel... ran out of time but will implement for part 4

End of Group: Game Area

Task Status: 4/5

Group

Group: Misc

Tasks: 3

Points: 1

[^ COLLAPSE ^](#)**Task**

Group: Misc

Task #1: Add the pull request link for the branch

Weight: ~33%

Points: ~0.33

[^ COLLAPSE ^](#)**① Details:**

Note: the link should end with /pull/#

**🔗 Task URLs**

URL #1

<https://github.com/jnsnjit/jns-IT114-003/pull/15>

URL

<https://github.com/jnsnjit/jns-IT114-003/pull/15>

End of Task 1

Task

Group: Misc

Task #2: Talk about any issues or learnings during this assignment

Weight: ~33%

Points: ~0.33

[^ COLLAPSE ^](#)**📝 Task Response Prompt**

Response:

honestly... the hardest part with this milestone was merging code for milestone 3 with my existing code from milestone2. IT TOOK a long time, be very careful to merge everything correct, as well as learning how to merge them properly. I am pretty you did a example in class, but I couldnt find a video that showed this process in detail, so this part of the project ended up taking me the most time, probably roughly 8+ hours alone.

End of Task 2

Task

Group: Misc
Task #3: WakaTime Screenshot
Weight: ~33%
Points: ~0.33

▲ COLLAPSE ▾

1 Details:

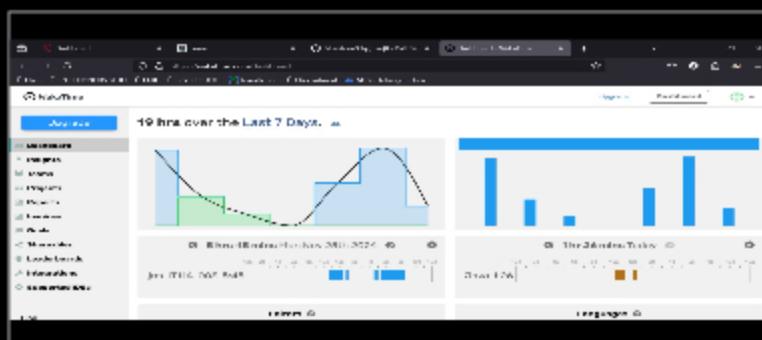
Grab a snippet showing the approximate time involved that clearly shows your repository. The duration isn't considered for grading, but there should be some time involved



Task Screenshots

Gallery Style: 2 Columns

4 2 1



wakatime screenshot

End of Task 3

End of Group: Misc

Task Status: 3/3

End of Assignment