Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-003-F2024/it114-milestone-2-chatroom-2024-m24/grade/is2637

Course: IT114-003-F2024

Assigment: [IT114] Milestone 2 Chatroom 2024 (M24)

Student: Jack S. (js2637)

Submissions:

Submission Selection

1 Submission [submitted] 11/11/2024 4:34:07 PM

•

Instructions

↑ COLLAPSE ↑

- Implement the Milestone 2 features from the project's proposal document: https://docs.google.com/document/d/10NmvEvel97GTFPGfVwwQC96xSsobbSbk56145XizQG4/view
- 2. Make sure you add your ucid/date as code comments where code changes are done
- 3. All code changes should reach the Milestone2 branch
- Create a pull request from Milestone2 to main and keep it open until you get the output PDF from this assignment.
- 5. 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.
- 7. 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: Milestone2

Group



Group: Payloads

Tasks: 2 Points: 2

^ COLLAPSE ^



Group: Payloads

Task #1: Base Payload Class

Weight: ~50% Points: ~1.00





All code screenshots must have ucid/date visible.



Columns: 1



Group: Payloads

Task #1: Base Payload Class

Sub Task #1: Show screenshot of the Payload.java

Task Screenshots

Gallery Style: 2 Columns

2



base payload class

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown

Task Response Prompt

Briefly explain the purpose of each property and serialization

Response:

The purpose of the payload class is the check the payload and use the information to send/receive messages, set/send client ID, and set/get payload type.



Group: Payloads

Task #1: Base Payload Class

Sub Task #2: Show screenshot examples of the terminal output for base Payload objects

Task Screenshots



RollPayload code

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown

■ Task Response Prompt

Briefly explain the purpose of each property

Response:

It is a child class of payload. It takes in two values: the number of dice and the number of sides of the dice. We have a getter and setter for the dice side and the number of dice. We also have a override too string method.



Group: Payloads

Task #2: RollPayload Class

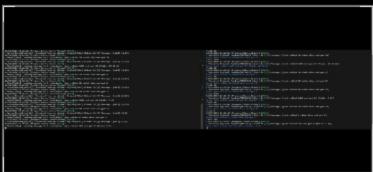
Sub Task #2: Show screenshot examples of the terminal output for base RollPayload objects

Task Screenshots

Gallery Style: 2 Columns

2

4



Roll payload on left terminal

Caption(s) (required) <

Caption Hint: Describe/highlight what's being shown

End of Task 2

End of Group: Payloads Task Status: 2/2

Group



Group: Client Commands

Tasks: 2 Points: 4

^ COLLAPSE ^

Task

Group: Client Commands



Weight: ~50%

Points: ~2.00

^ COLLAPSE ^



All code screenshots must have ucid/date visible.

Any output screenshots must have at least 3 connected clients able to see the output.

All commands must show who triggered it, what they did (specifically) and what the outcome was:

Columns: 1



Group: Client Commands Task #1: Roll Command

Sub Task #1: Show the client side code for handling /roll #

Task Screenshots

Gallery Style: 2 Columns

2

Roll code

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown

■ Task Response Prompt

Briefly explain the logic

Response:

The roll command checks to see if it has a d for the d for multi-dice rolls. Then, it uses the send method to create a payload and send it to the server thread to process the number of dice and sides.

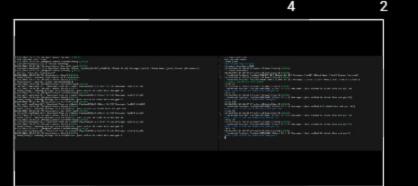
Sub-Task 100%

Group: Client Commands

Task #1: Roll Command

Sub Task #2: Show the output of a few examples of /roll # (related payload output should be visible)

Task Screenshots



Rolls on the right terminal

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown



Group: Client Commands Task #1: Roll Command

Sub Task #3: Show the client side code for handling /roll #d# (related payload output should be

visible)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

```
The state of the s
```

Roll Command

Caption(s) (required) <

Caption Hint: Describe/highlight what's being shown

=, Task Response Prompt

Briefly explain the logic

Response:

The roll command checks to see if it has a d for the d for multi-dice rolls. Then, it uses the send method to create a payload and send it to the server thread to process the number of dice and sides.



Group: Client Commands Task #1: Roll Command

Sub Task #4: Show the output of a few examples of /roll #d#

Task Screenshots



muti dice roll

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown



Group: Client Commands Task #1: Roll Command

Sub Task #5: Show the ServerThread code receiving the RollPayload

Task Screenshots

Gallery Style: 2 Columns

1

2



payload processing flip and roll

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown

=, Task Response Prompt

Briefly explain the logic

Response:

The roll payload creates a roll and then sends it to the room class to handle the logic to handle the roll in the room class



Group: Client Commands Task #1: Roll Command

Sub Task #6: Show the Room code that processes both Rolls and sends the response

Task Screenshots



handle roll

Caption(s) (required) <

Caption Hint: Describe/highlight what's being shown

=, Task Response Prompt

Briefly explain the logic

Response:

The handle method gets the sender's info the parameter for the dice roll and then processes the logic with a for loop. then send the message from the server side with null.

End of Task 1

Task



Group: Client Commands Task #2: Flip Command

Weight: ~50% Points: ~2.00

^ COLLAPSE ^

Columns: 1



Group: Client Commands Task #2: Flip Command

Sub Task #1: Show the client side code for handling /flip

Task Screenshots

Gallery Style: 2 Columns

4 2 1

] = #250-256 private void seroFlip()

flip command

Caption(s) (required) <

Caption Hint: Describe/highlight what's being shown

=, Task Response Prompt

Briefly explain the logic

Response:

Create and send the payload to be processed and handled in the server thread and room class. The room handles the logic while the server thread processes the payload.



Group: Client Commands Task #2: Flip Command

Sub Task #2: Show the output of a few examples of /flip (related payload output should be visible)

Task Screenshots

Gallery Style: 2 Columns

2

1

```
The state of the s
```

flip

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown

End of Task 2

End of Group: Client Commands

Task Status: 2/2

Group

100%

Group: Text Formatting

Tasks: 1 Points: 3

^ COLLAPSE ^

Task

Group: Text Formatting



Task #1: Text Formatting

Weight: ~100% Points: ~3.00

^ COLLAPSE ^



All code screenshots must have ucid/date visible.

Any output screenshots must have at least 3 connected clients able to see the output.

Note: Having the user type out html tags is not valid for this feature, instead treat it like WhatsApp,

Discord, Markdown, etc

Columns: 1



Group: Text Formatting
Task #1: Text Formatting

Sub Task #1: Show the code related to processing the special characters for bold, italic, underline,

and colors, and converting them to other characters (should be in Room.java)

1

Task Screenshots

Gallery Style: 2 Columns

2

text format

Caption(s) (required) <

Caption Hint: Describe/highlight what's being shown

Task Response Prompt

Briefly explain how it works and the choices of the placeholder characters and the result characters Response:

It replaces all the asterisks and underlines and #color with the correct format for the next milestone. the class is used in the send message method.



Group: Text Formatting

Task #1: Text Formatting

Sub Task #2: Show examples of each: bold, italic, underline, colors (red, green, blue), and combination of bold, italic, underline and a color

Task Screenshots



Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown

End of Task 1

End of Group: Text Formatting

Task Status: 1/1

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 ^

Details:

Note: the link should end with /pull/#



⇔Task URLs

URL #1

https://github.com/Jackshii/Js2637-IT114-003/pull/11

https://github.com/Jackshii/Js2637-IT114-003/p

1

End of Task 1

Task



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

Weight: ~33% Points: ~0.33

^ COLLAPSE ^

■ Task Response Prompt

Response:

I had to look up how to change the format for the bold and color with * and _. I also had to look back on the slides to check where to put the codes because sometimes I'll be confused about where things go.

End of Task 2

Task

100%

Group: Misc

Task #3: WakaTime Screenshot

Weight: ~33% Points: ~0.33

^ COLLAPSE ^



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

1

2



Waka time

End of Task 3

End of Group: Misc Task Status: 3/3

End of Assignment