Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-004-S2024/it114-milestone-2-chatroom-2024/grade/msa224

IT114-004-S2024 - [IT114] Milestone 2 Chatroom 2024

Submissions:

Submission Selection

1 Submission [active] 5/1/2024 5:45:04 PM

Instructions

COLLAPSE

1. Implement the Milestone 2 features from the project's proposal document: https://docs.google.com/document/d/1ONmvEvel97GTFPGfVwwQC96xSsobbSbk56145XizQG4/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
4. 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.
6. 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
8. Complete the pull request that was opened earlier
9. Upload the same output PDF to Canvas

Branch name: Milestone2

Tasks: 12 Points: 10.00

Demonstrate Usage of Payloads (2 pts.)

COLLAPSE

EXPAND

Task #1 - Points: 1

Text: Screenshots of your Payload class and subclasses and PayloadType

EXPAND

Task #2 - Points: 1

Text: Screenshots of the payloads being debugged/output to the terminal

EXPAND

Task #3 - Points: 1

Text: Explain the purpose of payloads and how your flip/roll payloads were made

COLLAPSE

Demonstrate Roll Command (2 pts.)

EXPAND

Task #1 - Points: 1

Text: Screenshot of the following items

EXPAND

Task #2 - Points: 1

Text: Explain the logic in how the two different roll formats are handled and how the message flows from the client, to the Room, and shared with all other users

COLLAPSE

Demonstrate Flip Command (1 pt.)

EXPAND

Task #1 - Points: 1

Text: Screenshot of the following items

EXPAND

Task #2 - Points: 1

Text: Explain the logic in how the flip command is handled and processed and how the message flows from the client, to the Room, and shared with all other users

COLLAPSE

Demonstrate Formatted Messages (4 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshot of Room how the following formatting is processed from a message

Details:

Note: this processing is server-side

Slash commands are not valid solutions for this and will receive 0 credit

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Room code processing for bold
#2	1	Room code processing for italic
#3	1	Room code processing for underline
#4	1	Room code processing for color (at least R, G, B or support for hex codes)
#5	1	Show each one working individually and one showing a combination of all of the formats and 1 color from the terminal
#6	1	Must not rely on the user typing html characters, but the output can be html characters
#7	1	Code screenshots should include ucid and date comment
#8	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```
//msa224 4/30/24 begins
public String processMessage(String message) {

    message = message.replaceAll(regex:"\\*(.*?)\\*", replacement:"<b>$1</b>");
    message = message.replaceAll(regex:"\\*(.*?)\\", replacement:"<i>$1</i>");
    message = message.replaceAll(regex:"\\^r\\((.*?)\\)", replacement:"<span style='color:red'>$1</span>");
    message = message.replaceAll(regex:"\\^g\\((.*?)\\)", replacement:"<span style='color:green'>$1</span>");
    message = message.replaceAll(regex:"\\^b\\((.*?)\\)", replacement:"<span style='color:blue'>$1</span>");
    message = message.replaceAll(regex:"\\~(.*?)\\~", replacement:"<u>$1</u>");

    return message;      You, 1 hour ago • milestone 2 edits
} <- #37-47 public String processMessage(String message)
//msa224 4/30/24 ends
```

I could not understand or figure out how to show this in my terminal but I knew this is how the code should look somewhat, because for the bold italics underline and colors that is the layout you would need to capture the message and then wrap it and then spit out whatever change you need. Same thing for the color but using the HTML span color you would change it to whatever specified.

Checklist Items (8)

#1 Room code processing for bold

#2 Room code processing for italic

#3 Room code processing for underline

#4 Room code processing for color (at least R, G, B or support for hex codes)

#5 Show each one working individually and one showing a combination of all of the formats and 1 color from the terminal

#6 Must not rely on the user typing html characters, but the output can be html characters

#7 Code screenshots should include ucid and date comment

#8 Each screenshot should be clearly captioned

COLLAPSE

Task #2 - Points: 1

Text: Explain the following

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Which special characters translate to the desired effect
#2	1	How the logic works that converts the message to its final format

Response:

So using regex, "\(.?)*", is how we get the text of the message that are wrapped and then we can bold it using <b>. the same goes for italic and underlined.

Now for the color we used span style colors HTML tags to change the text to a color of our choice for example blue green red. We use the same regex to get the wrapped messages as well.

message = message.replaceAll is self explanatory.

COLLAPSE

Misc (1 pt.)

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/msa224/msa224-it114-004/pull/12

COLLAPSE

Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

I had a bunch of issues that I had to work through to get most things to work. I still couldn't figure out the message formatting, the only thing I knew using W3 schools and my prior knowledge with regex and HTML was a code for the formatting but I couldn't figure out how to show it works.

As for everything else, i used a lot of the baseline codes provided to figure out how to format things and make them work so most of it is just using that and prior knowledge and experience from previous homework like java sockets to help me out, like for the actual roll code and flip code. I used the flip i had from previous homework and I had used a diceroll a while ago and I kind of already knew how to setup the "math" part of the code if that makes sense.

but after working through all the errors I finally got everything to work, the roll was the hardest for sure...after the message formatting since its still not working lol.

COLLAPSE

Task #3 - Points: 1

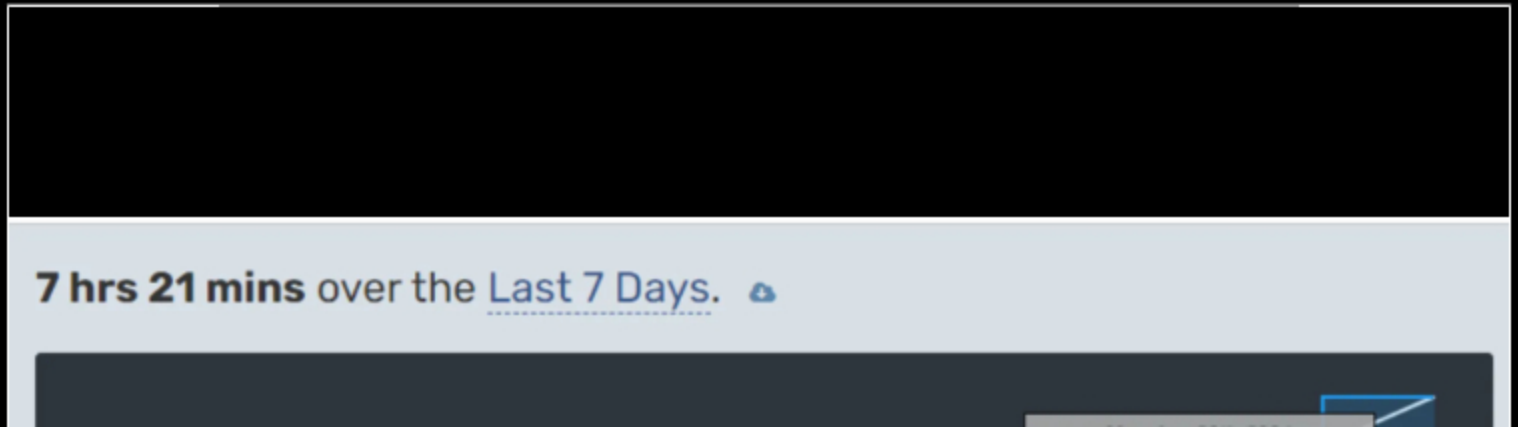
Text: WakaTime Screenshot

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



Shows wakatime

End of Assignment