

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT114-450-M2024/it114-milestone-2-chatroom-2024-m24/grade/vk686>

IT114-450-M2024 - [IT114] Milestone 2 Chatroom 2024 (M24)

Submissions:

Submission Selection

1 Submission [active] 7/8/2024 11:31:25 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: 8 Points: 10.00



Payloads (2 pts.)

^ COLLAPSE ^



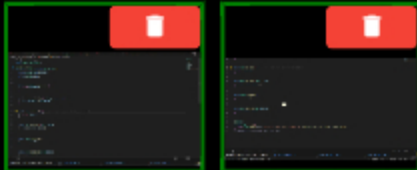
Task #1 - Points: 1

Text: Base Payload Class

Details:

All code screenshots must have ucid/date visible.

#1) Show screenshot of the Payload.java



Caption (required) ✓

Describe/highlight what's being shown

Showing screenshot of the Payload.java

Explanation (required) ✓

Briefly explain the purpose of each property and serialization

PREVIEW RESPONSE

When a client wants to send a message to the server, it creates a Payload object with the corresponding payloadType, clientId and message. This object is then serialized into a stream of bytes and sent over the network to the server. The server deserializes the byte stream back into a Payload object, reads the properties, and processes the message accordingly.

#2) Show screenshot examples of the terminal output for



Caption (required) ✓

Describe/highlight what's being shown

Showing screenshot examples of the terminal output for base Payload objects



^COLLAPSE ^

Task #2 - Points: 1

Text: RollPayload Class

Details:

All code screenshots must have ucid/date visible.

#1) Show screenshot of the RollPayload.java (or equivalent)



Caption (required) ✓

Describe/highlight what's being shown

Showing screenshot of the RollPayload.java

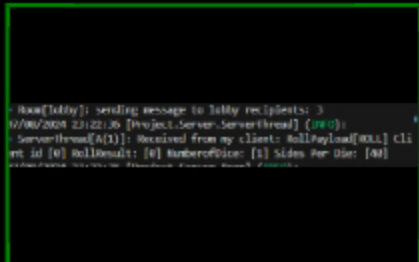
Explanation (required) ✓

Briefly explain the purpose of each property

PREVIEW RESPONSE

The RollPayload class is intended for storing all the information about a dice roll order, such as the number of dice, the sides of each die, and the roll's outcome. It extends common properties to the Payload class, such as payloadType and clientId, which help identify and process the payload in the context of a network.

#2) Show screenshot examples of the terminal output for



Caption (required) ✓

Describe/highlight what's being shown

Showing screenshot examples of the terminal output for base RollPayload objects

Client Commands (4 pts.)

^COLLAPSE ^

Task #1 - Points: 1

Text: Roll Command

i Details:

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.

Briefly explain the logic

PREVIEW RESPONSE

handleRollPayload(RollPayload rollPayload)

This method processes the roll command from the client. It retrieves the number of dice and sides per die from the RollPayload. It calls the rollDice method to simulate rolling the dice and sets the result in the RollPayload. Depending on the number of dice, it formats a result message and sends it to the current room.

the response

Explanation (required) ✓

Briefly explain the logic

PREVIEW RESPONSE

These methods ensure that the roll and flip commands from the clients are processed on the server side, and the results are sent back to all clients in the room.

Task #2 - Points: 1

Text: Flip Command

#1) Show the client side code for handling /flip



```
if (payload.type === 'flip') {
  // Send the flip command to the server
  socket.emit('flip', payload);
} else if (payload.type === 'roll') {
  // Send the roll command to the server
  socket.emit('roll', payload);
}
```

Caption (required) ✓

Describe/highlight what's being shown

Showing the client side code for handling /flip

Explanation (required) ✓

Briefly explain the logic

PREVIEW RESPONSE

The sendFlip method sends a flip command to the server.

#2) Show the output of a few examples of /flip (related payload)



```
client: [10] Message: [10]
  type: 'flip', payload: {
    type: 'flip',
    payload: {
      type: 'flip',
      payload: {
        type: 'flip',
        payload: {
          type: 'flip',
          payload: {
            type: 'flip',
            payload: {
              type: 'flip',
              payload: {
                type: 'flip',
                payload: {
                  type: 'flip',
                  payload: {
                    type: 'flip',
                    payload: {
                      type: 'flip',
                      payload: {
                        type: 'flip',
                        payload: {
                          type: 'flip',
                          payload: {
                            type: 'flip',
                            payload: {
                              type: 'flip',
                              payload: {
                                type: 'flip',
                                payload: {
                                  type: 'flip',
                                  payload: {
                                    type: 'flip',
                                    payload: {
                                      type: 'flip',
                                      payload: {
                                        type: 'flip',
                                        payload: {
                                          type: 'flip',
                                          payload: {
                                            type: 'flip',
                                            payload: {
                                              type: 'flip',
                                              payload: {
                                                type: 'flip',
                                                payload: {
                                                  type: 'flip',
                                                  payload: {
                                                    type: 'flip',
                                                    payload: {
                                                      type: 'flip',
                                                      payload: {
                                                        type: 'flip',
                                                        payload: {
                                                          type: 'flip',
                                                          payload: {
                                                            type: 'flip',
                                                            payload: {
                                                              type: 'flip',
                                                              payload: {
                                                                type: 'flip',
                                                                payload: {
                                                                  type: 'flip',
                                                                  payload: {
                                                                    type: 'flip',
                                                                    payload: {
                                                                      type: 'flip',
                                                                      payload: {
                                                                        type: 'flip',
                                                                        payload: {
                                                                          type: 'flip',
                                                                          payload: {
                                                                            type: 'flip',
                                                                            payload: {
                                                                              type: 'flip',
                                                                              payload: {
                                                                                type: 'flip',
                                                                                payload: {
                                                                                  type: 'flip',
                                                                                  payload: {
                                                                                    type: 'flip',
                                                                                    payload: {
                                                                                      type: 'flip',
                                                                                      payload: {
                                                                                        type: 'flip',
                                                                                        payload: {
                                                                                          type: 'flip',
                                                                                          payload: {
                                                                                           ...
                                                                                          }
                                                                                        }
                                                                                      }
                                                                                    }
                                                                                  }
                                                                                }
                                                                              }
                                                                            }
                                                                          }
                                                                        }
                                                                      }
                                                                    }
                                                                  }
                                                                }
                                                              }
                                                            }
                                                          }
                                                        }
                                                      }
                                                    }
                                                  }
                                                }
                                              }
                                            }
                                          }
                                        }
                                      }
                                    }
                                  }
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Caption (required) ✓

Describe/highlight what's being shown

Showing the output of a few examples of /flip

Text Formatting (3 pts.)

COLLAPSE

^COLLAPSE ^

Task #1 - Points: 1

Text: Text Formatting

Details:

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

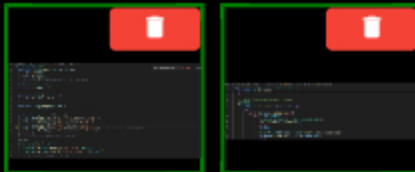
Note: Each text trigger must wrap the text that you want to affect

Note: Slash commands are not an accepted solution, the text must be transformed

Note: You do not need to use the same symbols in the below example, it's just an example, also, the below example doesn't show the "correct" output for colors, I'm leaving the proper conversion up to research on your own.

See proposal for an example.

#1) Show the code related to processing the special characters



Caption (required) ✓

Describe/highlight what's being shown

Show the code related to processing the special characters for bold, italic, underline, and colors, and converting them to other

Explanation (required) ✓

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

PREVIEW RESPONSE

The TextFX.java class is a utility class responsible for formatting text. It contains methods to convert placeholder characters into HTML-like tags that represent bold, italic, underline, and colors.

#2) Show examples of each: bold, italic, underline colors (red



Caption (required) ✓

Describe/highlight what's being shown

Show examples of each feature

^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/VK686NJ/vk686-IT114-450/pull/11>

Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

One of the main challenges was managing the complexity of command processing within the chat server. The implementation of commands such as coin tosses, dice rolls, and text formatting required careful logic to ensure correct functionality. I also caused problems understanding the payload, how to send it back and forth, and what to send. I watched all the lectures to find out how it works.

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

Files		Branches	
1hr 35 mins	Project/Client/Client.java	5 hrs 41 mins	Milestone2
1 hr 12 mins	..ct/Server/ServerThread.java	54 mins	Milestone2-Prep
50 mins	Project/Common/TextFX.java	27 mins	main
39 mins	Project/Common/Payload.java		
36 mins	Project/Server/Room.java		
32 mins	..ect/Common/RollPayload.java		
21 mins	..ect/Common/PayloadType.java		
16 mins	Project/Server/Server.java		
13 mins	server-0.jpg		
8 mins	Project/sources.txt		
7 mins	Project/ServerThread.java		


```

6 mins  _ct/Server/ServerPlayer.java
3 mins  viad_output.txt
2 mins  run.sh
2 mins  Project/Client.java
1 min   _mmon/ConnectionPayload.java
1 min   _mon/RoomResultsPayload.java
1 min   _server/BaseServerThread.java
1 min   Project/Client/ClientData.java
1 min   gitignore
1 min   Project/BaseServerThread.java
1 min   _ct/Common/ReadyPayload.java
55 secs Project/TestFX.java
47 secs Project/Common/LoggerUtil.java
47 secs Project/PayloadType.java
32 secs _mmon/ConnectionPayload.java
31 secs _ct/Client/ClientPlayer.java
30 secs Project/Common/TimedEvent.java
26 secs Project/Room.java
21 secs build.sh
17 secs _ct/Server/BaseGameRoom.java
16 secs Project/Common/Phone.java
15 secs Project/Payload.java
12 secs Project/ClientData.java
11 secs Project/Common/Player.java
10 secs _mon/RoomResultsPayload.java
10 secs Project/ConnectionPayload.java
9 secs  Project/Server.java
6 secs  Project/Server
1 sec   client-0.log.1
0 secs  client-0.log

```

Detailed view

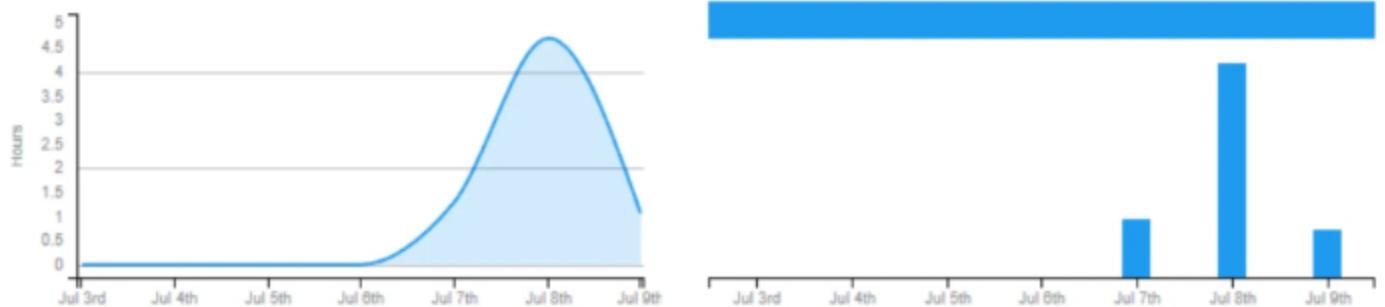


Projects • vk686-IT114-450

total 17 hrs 54 mins



7 hrs 3 mins over the Last 7 Days in vk686-IT114-450 under all branches. 📈



Overall

End of Assignment