# Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-006-S2024/it114-milestone-2-chatroom-2024/grade/owe

IT114-006-S2024 - [IT114] Milestone 2 Chatroom 2024

#### Submissions:

Submission Selection

1 Submission [active] 4/28/2024 5:44:26 PM

Instructions

^ COLLAPSE ^

Implement the Milestone 2 features from the project's proposal document:

https://docs.google.com/document/d/10NmvEvel97GTFPGfVwwQC96xSsobbSbk56145XizQG4/view

Make sure you add your ucid/date as code comments where code changes are done

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.

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: Milestone2

Tasks: 12 Points: 10.00

Demonstrate Usage of Payloads (2 pts.)

ACOLLAPSE A



Task #1 - Points: 1

Text: Screenshots of your Payload class and subclasses and PayloadType

Checklist

\*The checkboxes are for your own tracking

#	ronnes	Details
#1	1	Payload, equivalent of RollPayload, and any others
#2	1	Screenshots should include ucid and date comment
#3	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```
📢 File Edit Selection View Go Run Terminal Help
                                                                                                                                                                                    J Payloadjava A ● ▷ ∨ to III ···
<u>C</u>
        > M2JavaProblems
                                                       import java.io.Serializable:
        > M3NumberGuesser4
See
G
                                                          public long getClientId() {
    return clientId;
                                                           public void setClientId(long clientId) {
                                                           // Read https://www.baeldung.com/java-serial-version-uid
private static final long serialVersionUID = 11; // Serialization version UID
ط
                                                           private PayloadType payloadType;
         J TextFX.class
                                                           public PayloadType getPayloadType() {
```

#### payload class with implications

Checklist Items (0)

# payload continued

Checklist Items (0)

```
The lidt Selection Vew Go Aun Terminal Help 6 -> Downstrony and A J Poloutions III in the Control of the Contro
```

# payload type class

Checklist Items (0)

```
J PayloadTypejaw A 22 setPayloadType(PayloadType.LEST_ROOMS);

J RoomResultPayloadjava A 25

J TextFX_lars A 26 return rooms;

J TextFX_lars A 27

J TextFX_Scolot.class A 28

V Server 0 29 public void setRooms(ListCString> rooms) {

J Room_class A 30 this rooms = rooms:

J Room_java A PROBLIMS OUTPUT DEBUG CONSOLE TEMPANL POSTS

+ v ··· ^ X
```

subclasses for payload

# Checklist Items (0)



#### Task #2 - Points: 1

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

Checklist	t	*The checkboxes are for your own tracking
#	Points	Details
<b>#</b> 1	1	Demonstrate flip
#2	1	Demonstrate roll (both versions)
#3	1	Demonstrate formatted message along with any others
#4	1	Each screenshot should be clearly captioned

## Task Screenshots:

Gallery Style: Large View

Small Medium Large 📢 File Edit Selection View Go Run Terminal Help d ta III ... √ OWE-114-006 java \$1.Server.Server elif [ "\$2" = "client" ]; then java \$1.Client.Client # In Milestoned changes Client to ClientUI wlif [ "\$2" = "ui" ]; then jave \$1.Client.ClientUI ∨ Part3HW\Module4\Part3 ∨ Client J Client.class J Clientjava J Client\$1.class J Client\$2.dass J ConnectionPayload.class ٠ J Payload java J PayloadType.class J PayloadType java J RoomResultsPayload.class ■ RoomResultsPayload.iava J TextFX.class J TextFX java J TextFX\$Color.class m info INFO: Room[poop]: Sending message to 2 clie mts INFO: Debug Info: Type[MESSAGE], Message[Di ient processPayload ce roll result: 0], ClientId[2] deex: Dice roll result: 0 INFO: "deez connected" Apr 29, 2024 12:37:28 AM Project.Client.Cl L [] java nts
Apr 29, 2824 12:37:47 AM Project.Server.Ser
verthread info
3NFO: Thread[deez]: Received from client: T
ype[MESSABE], Hessage[Coin flip result: Hea
ds], ClientId[0]
Apr 29, 2824 12:37:47 AM Project.Server.Roo ient\$2 run Apr 29, 2024 12:37:47 AM Project.Client.Cli ent\$1 run IMFO: Debug Info: Type[MESSAGE], Message[D ice roll result: 0], ClientId[2] INFO: Waiting for input Apr 29, 2024 12:37:47 AM Project.Client.Cli deez: Dice roll result: 0 Apr 29, 2024 12:37:47 AM Project.Client.Cl ent\$2 run



## flip and roll

#### Checklist Items (0)

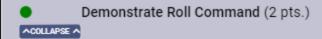


#### Task #3 - Points: 1

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

## Response:

In our system, payloads are structured data packages that help clients and servers communicate with each other. They contain a variety of commands and messages, including CREATE\_ROOM, JOIN\_ROOM, MESSAGE, CONNECT, DISCONNECT, and LIST\_ROOMS. We have added a new payload type called ROLL, which includes the data needed to simulate dice rolls. The system's functionality is improved by this new feature, which lets users participate in interactive games like dice.





# Task #1 - Points: 1

Text: Screenshot of the following items

e.hi Tiichrack II acc(),

else if (text.equalsIgnoreCase(FLIP\_COMMAND)) {

return true;

Checklist		*The checkboxes are for your own tracking		
#	Points	Points Details		
#1	1	Client code that captures the command and converts it to a RollPayload (or equivalent) for both scenarios /roll # and /roll #d#		
#2	1	ServerThread code receiving the payload and passing it to the Room		
#3	1	Room handling the roll action correctly for both scenarios (/roll # and /roll #d#) including the message going back out to all clients		
#4	1	Code screenshots should include ucid and date comment		
#5	1	Each screenshot should be clearly captioned		

#### Task Screenshots:

Gallery Style: Large View

	SIIIaii	Mediuiii	Laiye	

```
sendFlip();
} catch (IOException e) {
    e.printStackTrace();
}
return true;
} else if (text.startsWith(ROLL_COMMAND)) {
    String rollString = text.replace(target:"/roll", replacement:"").trim();
    try {
        int result = roll(rollString);
        sendRoll(result);
    } catch (IOException e) {
        sendMessage(message:"Wrong Format. Use : '/roll 123' or '/roll 456'.");
}
return true;
}
return false;
}
```

#### the client code for the dice for roll

## Checklist Items (0)

```
J ServerThreadjava A X D ∨ 🖏 🗓 -
G
                                   Project > Server > → ServerThread.java > 1 ServerThread > ⊕ processPayload(Payload)
       V OWE.114.006
                                             21 public class ServerThread extends Thread (
        > M2JavaProblems
        > M3NumberGuesser4
å.
                                                                  @SuppressMarnings("unused")
private void sendFlip() throws IOException {
                                                                     Payload p = new Payload();
p.setPayloadType(PayloadType.MESSAGE);
                                                                        out.writeObject(p);
                                                                  @SuppressWarnings("unused")
                                                                       return (int) (Math.random() * 6) + 1;
                                                                  @SuppressWarnings("unused")
private void sendRoll(int result) throws IOException (
           J RoomResultsPayload.java
           J TextFX.class
                                                                     Payload p = new Payload();
p.setPayloadType(PayloadType.MESSAGE);
                                                                       p.setMessage("Roll result:
out.writeObject(p);
                                                                                                             64 /c/MEM JERSEY INSTITUTE OF TECHNOLOY oner
elhossary/NJIT SPRING 2824/IT 114/OME-114-
806 (Milestone2)
5 []
                                                      64 /c/MBW JERSEY INSTIUTE OF TECHNOLOY ona-
elhossary/NJIT SPRING 2024/IT 114/DME-114-
006 (Milestone2)$ git status
      > OUTLINE
                                                                                                                                                                                       Screenshot copied to clipboard and saved 
Select here to mark up and share.
      > TIMELINE
      > JAWA PROJECTS
P Milestone2*+ @ @ 0 A 0 W 0 & 50 3 hrs 5 mins & Java: Ready
                                                                                                                                                                                             Ln 216, Col 18 Spaces: 4 UIF-6 CRLF () Java 😃
                                                                                                                                                                                                         ^ ♥ Q0 🖢 6D6 PM # 🦺
                                                                Q Search
                                                                                                  (a) 📭 🕩 🔚 💕 🤣 😥 🖂 💆 💐 🗒
```

more server thread code for roll

#### Checklist Items (0)

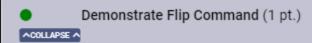


#### Task #2 - Points: 1

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

#### Response:

In our system, the Room class receives a roll message sent from the client, which is then processed to extract the roll details. All other users in the room receive this roll message after it is broadcast by the Room class. Real-time communication is made possible during gameplay as each user receives the message and can see the outcome of the roll. This logic improves the user experience by keeping all users informed and synchronized about the current dice rolls.





#### Task #1 - Points: 1

Text: Screenshot of the following items

Checklist		*The checkboxes are for your own tracking	
#	Points Details		
#1	1	Client code that captures the command and converts it to a payload	
#2	1	ServerThread receiving the payload and passing it to the Room	
#3	1	Room handling the flip action correctly	
#4	1	Code screenshots should include ucid and date comment	
#5	1	Each screenshot should be clearly captioned	

Task Screenshots:

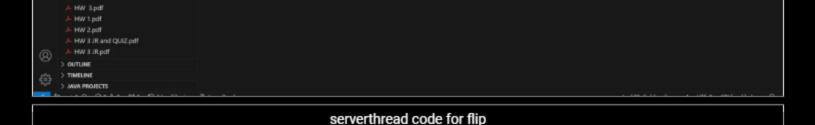
Gallery Style: Large View

Medium

Large

Small

**□** □ □ 08 File Edit Selection View Go Run Terminal Help Φ DIPLORER ... J Clientiava X V OWE-114-006 Project > Client > J Client,java > d Client > ⊕ getClientName() private boolean processClientCommand(String text) throws IOException ( } else if (text.equalsIgnoreCase(FLIP\_COMMAND)) { sendFlip(); ∨ Client e.printStackTrace(); J Client\$1.class } else if (text.startsWith(ROLL\_COMMAND)) {
 String rollString = text.replace(target:"/roll", replacement:"").trim(); J Client\$2.class J Client\$3.class J ConnectionPayload.class sendRoll(result);
) catch (IOException e) { J ConnectionPayload.java J Constants.class sendMessage(message: "Wrong Format. Use : '/roll 123' or '/roll 456'."); J Constants.java J Payload.class J PayloadType.class J PayloadType,java J RoomResultsPayload.java private void sendDisconnect() throws IDException ( ConnectionPayload cp = new ConnectionPayload(); cp.setPayloadType(PayloadType.DISCONNECT); J TextFX java out.writeObject(cp); J TextFX\$Color.class > Server Filter (e.g. text, "7".ts. I"... ▼ Ø 를 ^ X owe\_it114-project-milestone-1\_IT114-0... No problems have been detected in the workspace



# Checklist Items (0)



## 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

#### Response:

In order to handle the flip command, it is sent from the client to the Room class, where it is processed to ascertain the flip's outcome. The message is then broadcast to all other users in the room with this outcome. Real-time communication and coordinated gameplay are made possible by the message that each user receives and the ability to view the flip's result. This logic makes sure that everyone is aware of the outcome of the flip, which makes it easier for people to interact and communicate in the shared virtual environment.

Demonstrate Formatted Messages (4 pts.)



#### Task #1 - Points: 1

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

①Details:

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		
<b>#</b> 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
<b>#</b> 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

Medium

Large

Small

File Edit Selection View Go Run Terminal Help <u>G</u> J ServerThread java A D ∨ 10. III ·· EXPLORER J RoomResultsPayload.iava A ∨ OWE-114-006 10 public class Room implements AutoCloseable {
120 private synchronized void syncClientList(ServerThread joiner) { > M2JavaProblems > M3NumberGuesser4 protected static void createRoom(String roomName, ServerThread client) { if (Server.INSTANCE.createNewRoom(roomName)) { Room.joinRoom(roomName, client); client.sendMessage(Constants.DEFAULT\_CLIENT\_ID, String.format(format:"Room %s already exists", roomName)); protected static void joinRoom(String roomName, ServerThread client) ( if (IServer.INSTANCE.joinRoom(roomName, client)) {
 client.sendMessage(Constants.DEFAULT\_CLIENT\_ID, String.format(format:"Room %s doesn't exist", roomName)); protected static List<String> listRooms(String searchString, int limit) ( return Server.INSTANCE.listRooms(searchString, limit); client.setCurrentRoom(room:null);
client.disconnect(); PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS J Roomjava r la bash 64 /c/NEW DERSEY INSTRUTE OF TECHNOLGY omar-elhossary/NDIT SPRING 2024/IT 114/OME-114-006 (Milestone2)\$ git status 64 /c/NEW JERSEY INSTITUTE OF TECHNOLOGY owar elhossary/NOIT SPRING 2024/IT 114/ONE-114-006 (Milestone2) \$ [] SEV INSTIUTE OF TECHNOLSY onar elhossary/N JIT SPRING 2024/IT 114/OME-114-006 (Milest l L bash cmd 🖂 powershell

# code for room was not finished

# Checklist Items (0)

> OUTLINE



Task #2 - Points: 1

Text: Explain the following

Checklist		*The checkboxes are for your own tracking
#	Points	Details
<b>#</b> 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: i did not get to do this Misc (1 pt.) ^COLLAPSE ^ Task #1 - Points: 1 ACOLLAPSE A Text: Add the pull request link for the branch Details: Note: the link should end with /pull/# **URL #1** https://github.com/WayguBeef5/OWE-114-006/pull/9 Task #2 - Points: 1 ^COLLAPSE ^ Text: Talk about any issues or learnings during this assignment Response: i did not get it to work was moving quickly to get to milestone 2 was in contact with professor to make sure it somewhat working for demo

↑COLLAPSE ↑

Task #3 - Points: 1

**Text: WakaTime Screenshot** 

Details:

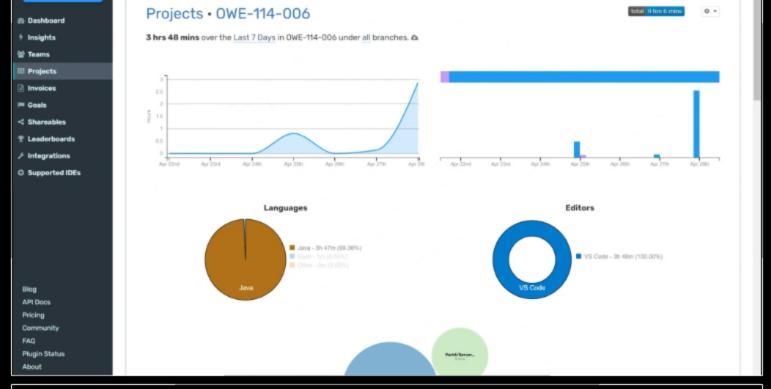
Grab a snippet showing the approximate time involved that clearly shows your repository. The duration isn't considered for grading, but there should be some time involved

Task Screenshots:

Gallery Style: Large View

Small Medium Large

| Small Medium Large | Small Medium Large | Small State | Small S



screen shot for the waka time

**End of Assignment**