Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-004-S2024/it114-milestone-2-chatroom-2024/grade/rn364

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

Submissions:

Submission Selection

1 Submission [active] 4/28/2024 9:06:00 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.)

^COLLAPSE ^



Task #1 - Points: 1

Text: Screenshots of your Payload class and subclasses and PayloadType

Checklist

*The checkboxes are for your own tracking

#	Points	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

```
> OPEN EDITORS
                       Project > Common > () RollPayload.java > & RollPayload > (j) getSides()
                                  private int dice, sides;
   formatting
                                 public RollPayload(int dice, int sides) {
  () ConnectionPayloa...
                                       setPayloadType(PayloadType.ROLL);
  () Constants.class
  () Constants.java
                                 public int getDice() {
  () Payload.class
                                 return dice;
  () Payload java
  () PayloadType.class
  ( ) PayloadType.java
  () RollPayload.class
  () RollPayload.j... M 18
  () RoomResultsPaylo...
                                  public int getSides() {
                                  return sides;
  ( ) RoomResultsPaylo...
  () TextFX.class
∨ OUTLINE
   → Project.Common
                                      this sides = sides;
 RollPayload
   @ dice
    RollPayload(int, int)
```

RollPayload, payload and payloadtype. does include my ucid //rn364

Checklist Items (3)

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

```
() Payload.class
                                   public void setClientId(long clientId) {
   ( ) PayloadType.class
   () PayloadType.java
  () RollPayload.class
  () RollPayload.j... M 18
   () RoomResultsPaylo...
   () TextFX.class
                                   private PayloadType payloadType;
   Project.Common
                                   public PayloadType getPayloadType() {
                                      return payloadType;
    clientid
    @ getClientId() : long

    setClientid(long)...

                                   public void setPayloadType(PayloadType payloadType) (
                                       this payloadType - payloadType;
> TIMELINE
> JAVA PROJECTS
```

RollPayload, payload and payloadtype. does include my ucid //rn364

Checklist Items (3)

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

```
> OPEN EDITORS
                         Project > Common > () PayloadType.java > @ PayloadType
V RN364-1...[] [] [] [] [] []
                             public enum PayloadType {
CONNECT, DISCONNECT, MESSAGE, CREATE_ROOM, JOIN_ROOM, LIST_ROOMS, CLIENT_ID, SYNC_CLIENT, ROLL, FLIP,
   formatting
     Common
   () ConnectionPayloa...
   () ConnectionPayloa...
   () Constants.java
   () Payload.class
   () Payload java
   () PayloadType.class
   () PayloadType.java
   () RollPayload.j... M
   () RoomResultsPavio...
   () RoomResultsPaylo...
   () TextFX.class
∨ OUTLINE
   1 Project.Common
 ₽ DISCONNECT
```

RollPayload, payload and payloadtype. does include my ucid //rn364

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



Task #2 - Points: 1

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

Small

Checklist		*The checkboxes are for your own tracking
#	Points	Details
#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:

> TIMELINE

Gallery Style: Large View

Medium

Large

formatting 盘 () ConnectionPayloa... () ConnectionPayloa... () Constants.class () Constants.java PROBLEMS (36) OUTPUT DEBUG CONSOLE TERMINAL PORTS 日加・十・四 自 … へ () Payload.class r 🖸 java INFO: Room[lobby]: Sending message to 1 clients () Payload.java Apr 28, 2024 9:18:34 PM Project.Server.ServerThread info INFO: Thread[Ryan]: Received from client: Type[ROLL], Message[null], Cl Apr 28, 2024 9:18:26 PM Project.Client.Client\$2 run INFO: Debug Info: Type[MESSAGE], Message[coinflip is: Tails], ClientI () PayloadType.class () PayloadType.java ientId[0] d[-1] Apr 28, 2024 9:18:34 PM Project.Server.Room info () RollPayload.class INFO: Room[lobby]: Sending message to 1 clients Apr 28, 2024 9:18:34 PM Project.Server.ServerThread info INFO: Thread[Ryan]: Received from client: Type[MESSAGE], Message[/roll /roll 2d5 () RollPayload.j.. M Apr 28, 2024 9:18:34 PM Project.Client.Client\$1 run () RoomResultsPaylo... INFO: Malting for input
Apr 28, 2024 9:18:34 PM Project.Client.Client\$2 run
INFO: Debug Info: Type[MESSAGE], Moscago[Dice roll result is 8], Clie () RoomResultsPaylo... 2d5], ClientId[0] Apr 28, 2024 9:18:34 PM Project.Server.Room info () TextFX.class INFO: Room[lobby]: Sending message to 1 clients V OUTLINE Apr 28, 2024 9:19:01 PM Project.Server.ServerThread info [MODE]: DECOMPOSE PROJECT LS 8 Apr 28, 2024 9:18:34 PM Project.Client.Client\$2 run INFO: Debug Info: Type[MESSAGE], Message[/roll 2d5], ClientId[1] INFO: Thread[Ryan]: Received from client: Type[MESSAGE], Message[.*Ryan Project.Common *.], ClientId[0] Apr 28, 2024 9:19:01 PM Project.Server.Room info INFO: Room[lobby]: Sending message to 1 clients PayloadType d[™] CONNECT Apr 28, 2024 9:19:01 PM Project.Client.Client\$1 run ₱ DISCONNECT INFO: Waiting for input ■ MESSAGE Apr 28, 2024 9:19:01 PM Project.Client.Client\$2 run INFO: Debug Info: Type(MESSAGE), Message(Ryan), ClientId[1]

Checklist Items (4)

#1 Demonstrate flip

- #2 Demonstrate roll (both versions)
- #3 Demonstrate formatted message along with any others
- #4 Each screenshot should be clearly captioned



Task #3 - Points: 1

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

Response:

Payload prett ymuch carries to messages between the clients and the server. It sends the commands, the messages and the status. When the command of the flip is made the client creates a payload for the command then send it to server. It the gets the command and processes it. Same with roll

Demonstrate Roll Command (2 pts.)



Task #1 - Points: 1

Text: Screenshot of the following items

Checklist *The checkboxes are for your own tr		
#	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

Small Medium Large

} <- #178-184 else if (text.equalsIgnoreCase(LIST_USERS))

```
rn364_it114-sockets...
                                           String roll = text.replace(ROLL, replacement: "").trim();
                                          String[] rollParts = roll.split(regex:"d");
                                          if (rollParts.length >= 2) {
                                                   int lower = Integer.parseInt(rollParts[0]);
 () Client.class
                                                   int upper = Integer.parseInt(rollParts[1]);
                                                   sendRoll(lower, upper);
() Client$1.class
() Client$2.class
                                                   System.out.println(TextFX.colorize(text:"Socket error", Color.RED));
() Client$3.class
 formatting
                                                   e.printStackTrace();
  Common
() ConnectionPayloa...
                                          else {
() ConnectionPayloa...
() Constants.class
                                                   int lower - 1;
() Constants.java
                                                   int upper = Integer.parseInt(rollParts[0]);
                                                   sendRoll(lower, upper);
() Payload.class
OUTLINE

分 Project.Client

                                                   System.out.println(TextFX.colorize(text:"Socket error", Color.RED));
 ent Client
 ■ INSTANCE
                                          catch (Exception e) {
 server
                                              e.printStackTrace();
 @ out
TIMELINE
```

Client code

Checklist Items (1)

#1 Client code that captures the command and converts it to a RollPayload (or equivalent) for both scenarios /roll # and /roll #d#

```
() Serverflava 231
() Serverflavad case ROLL:
RollPayload roller = (RollPayload) p;
currentRoom.rollDice(roller.getDice(),roller.getSides());
break;
Case ROLL:
RollPayload p;
currentRoom.rollDice(roller.getDice(),roller.getSides());
break;
case FLIP: //ro364
currentRoom.flipCoin();
```

Serverthread code

Roomcode

Checklist Items (1)

#3 Room handling the roll action correctly for both scenarios (/roll # and /roll #d#) including the message going back out to all clients



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:

The server is notified when a player types anything like "/roll 6" or "/roll 2d6" in the game chat. The number of dice to roll and the number of sides on each die are then determined. All players in the game receive a message with the total roll once the dice have been rolled and the results have been added up

Demonstrate Flip Command (1 pt.)



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

Small Medium Large

```
else if (text.startsWith(ROLL)) {
                                           String roll = text.replace(ROLL, replacement:"").trim();
  m364_it114-sockets..
                                           String[] rollParts = roll.split(regex:"d");
                                           if (rollParts.length >= 2) {
                                                   int lower = Integer.parseInt(rollParts[θ]);
  () Client.class
                                                   int upper = Integer.parseInt(rollParts[1]);
                                                   sendRoll(lower, upper);
                                               } catch (IOException e) {
  () Client$1.class
                                                   System.out.println(TextFX.colorize(text:"Socket error", Color.RED));
   () Client$2.class
   () Client$3.class
                                                   e.printStackTrace();
   formatting
   Common
   () ConnectionPayloa...
                                              try [
                        200
   ( ) ConnectionPayloa...
                                                   int lower = 1;
   () Constants.class
                                                   int upper = Integer.parseInt(rollParts[0]);
   () Constants.java
                                                   sendRoll(lower, upper);
   () Payload.class

✓ OUTLINE

                                               catch (IOException e) {
   Project.Client
                                                   System.out.println(TextFX.colorize(text:"Socket error", Color.RED));
 ■ INSTANCE
                                           catch (Exception e) {
                                               e.printStackTrace();
    server
```

client code

Checklist Items (1)

#1 Client code that captures the command and converts it to a payload

```
{} Room.java 8 227 hpsak:
```

```
Server.class
                                              case JOIN_ROOM:
   () Server.java
                                                  Room.joinRoom(p.getMessage(), this);
   {} Server$1.class
   () ServerThread.class
                                              case ROLL:
                                                  RollPayload roller = (RollPayload) p;
                                                  currentRoom.rollDice(roller.getDice(),roller.getSides());
  () ServerThread$1.cla...
                                                  break;
  sources.txt
                                              case FLIP: //rn364
 build.sh
                                                  currentRoom.flipCoin();
 = ng4.txt
                                                  break;
() NumberGuesser4.class
                                              case LIST_ROOMS:
                                 Ö

✓ OUTLINE

                                                  String searchString = p.getMessage() == null ? "" : p.getMessage
                         239
                                                  int limit = 10:
   - Droiect Canvar
```

ServerThread receiving the payload and passing it to the Room

Checklist Items (1)

#2 ServerThread receiving the payload and passing it to the Room

```
325 } <- #320-325 public void close()
326 public void flipCoin()
327 {
328 Random random = new Random();
329 String result = random.nextBoolean() ? "Heads" : "Tails";
330 sendMessage(sender;null, "coinflip is: " + result);
1.cla... 331 } <- #327-331 public void flipCoin()
327 nublic void rollDice(int dice int sides) /
```

Room handling the flip action correctly

Checklist Items (1)

#3 Room handling the flip action correctly



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:

The message "/flip" is sent to the server by a client, and the server relays this information to the room. After flipping a coin, the room determines if it is heads or tails and relays the outcome to each and every client there. After receiving the outcome, each client presents it to the user. In essence, the server promotes an entire virtual coin flip among all users!





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

```
private String applyFormatting(String message)//rm364
                                                    if (message.contains(s:".")) && message.contains(s:"*.")
   () Client$3.class
                                                         message = message.replace(target:".*",replacement:"<b>");//rn364 BOLD
message = message.replace(target:"*.", replacement:"</b>");
    formatting
                                                    if ((message.contains(s:".|")) && message.contains(s:"|."))
   () Room.class
                                                         message = message.replace(target:".|",replacement:"<i>");//rn364 ITALICS
message = message.replace(target:"|.", replacement:"</i>");
   () Server.class
   () Serverjava
   () Server$1.class
                                                    if ((message.contains(s:"._")) && message.contains(s:"."))
    () ServerThread.class
                                                         message = message.replace(target:"._",replacement:"<u>"); //rn364 UNDERLINE
message = message.replace(target:"_.", replacement:"</u>");
   () ServerThreadi... 2
   () ServerThread$1.cla...
                                                    if ((message.contains(s:"R[")) && message.contains(s:"]R"))
 build.sh
                                                         message = message.replace(target:"R[",replacement:"<font> color = RED");//rn364
message = message.replace(target:"]R", replacement:"</font>");
 = ng4.txt
 () NumberGuesser4.class

∨ OUTLINE

                                                    if ((message.contains(s:"G[")) && message.contains(s:"]G"))
    Project.Server
                                                         message = message.replace(target:"G[",replacement:"<font> color = GREEN");//rn364
message = message.replace(target:"]G", replacement:"</font>");
     @ server
     @ name
                                                    if ((message.contains(s:"B[")) && message.contains(s:"]B"))
> TIMELINE
                                                         message = message.replace(target:"B[",replacement:"<font> color = BLUE");//rn364
message = message.replace(target:"]B", replacement:"</font>");
```

Done plus the comments on the side

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



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:

Using a message as input, this function looks for certain sequences such as ".*", ".|", "._", "R[...]R", "G[...]G", and "B[...]B". When it comes across one of these sequences, it replaces them with HTML elements so that the text can be formatted with bold, italics, underlining, or different colors (red, green, or blue). In this technique, it uses the special characters in the message to convert plain text into formatted text.





Task #1 - Points: 1

Text: Add the pull request link for the branch

① Details:

Note: the link should end with /pull/#

URL #1

https://github.com/ryann2n/rn364-IT114-004/pull/9



Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

Had a lot of help but was worth it to understand. Learned to format and type code.



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



End of Assignment