Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-006-S2024/it114-sockets-part-1-3-checkpoint/grade/kb96

IT114-006-S2024 - [IT114] Sockets Part 1-3-Checkpoint

Submissions:

Submission Selection

1 Submission [active] 2/21/2024 4:18:42 PM

Instructions

^ COLLAPSE ^

Create a new branch for this assignment

Go through the socket lessons and get each part implemented (parts 1-3)

You'll probably want to put them into their own separate folders/packages (i.e., Part1, Part2,

Part3) These are for your reference

Part 3, below, is what's necessary for this HW https://github.com/MattToegel/IT114/tree/Module4/Module4/Part3

Create a new folder called Part3HW (copy of Part3)

Make sure you have all the necessary files from Part3 copied here and fix the package references at the top of each file

Add/commit/push the branch

Create a pull request to main and keep it open

Implement two of the following server-side activities for all connected clients (majority of the logic should be processed server-side and broadcasted/sent to all clients if/when applicable)

Simple number guesser where all clients can attempt to guess while the game is active

Have a /start command that activates the game allowing guesses to be interpreted Have a /stop command that deactivates the game, guesses will be treated as regular messages (i.e., guess messages are ignored)

Have a guess command that include a value that is processed to see if it matches the hidden number (i.e., / guess 5)
Guess should only be considered when the game is active

The response should include who guessed, what they guessed, and whether or not it was correct (i.e., Bob guessed 5 but it was not correct)

No need to implement complexities like strikes

Coin toss command (random heads or tails)

Command should be something logical like /flip or /toss or /coin or similar

The result should mention who did what and got what result (i.e., Bob Flipped a coin and got heads)

Dice roller given a command and text format of "/roll #d#" (i.e., roll 2d6)

Command should be in the format of /roll #d# (i.e., roll 1d10)

The result should mention who did what and got what result (i.e., Bob rolled 1d10 and

Math game (server outputs a basic equation, first person to guess it correctly gets congratulated and a new equation is given)

Have a /start command that activates the game allowing equaiton to be answered Have a /stop command that deactivates the game, answers will be treated as regular messages (i.e., any game related commands when stopped will be ignored)

Have an answer command that include a value that is processed to see if it matches

the hidden number (i.e. / answer 15)

The response should include who answered, what they answered, and whether or not it was correct (i.e., Bob answered 5 but it was not correct)

Private message (a client can send a message targetting another client where only the two

can see the messages)
Command can be /pm, /dm followed by the user's name or an @ preceding the users name (clearly note which)

The server should properly check the target audience and send the response to the original sender and to the receiver (no one else should get the message)

Alternatively (make note if you do this and show evidence) you can add support to private message multiple people at once. Evidence should show a larger number of clients than the target list of the private message to show it works. Note to grader: if this is accomplished add 0.5 to total final grade on Canvas

Message shuffler (randomizes the order of the characters of the given message) Command should be /shuffle or /randomize (clearly mention what you chose) followed by the message to shuffle (i.e., /shuffle hello everybody)

The message should be sent to all clients showing it's from the user but randomized Example: Bob types / command hello and everyone recevies Bob: lleho

Fill in the below deliverables Save the submission and generated output PDF Add the PDF to the Part3HW folder (local) Add/commit/push your changes Merge the pull request Upload the same PDF to Canvas

Branch name: M4-Sockets3-Homework

Tasks: 7 Points: 10.00

Baseline (2 pts.) ^COLLAPSE ^

^COLLAPSE ∧

Task #1 - Points: 1

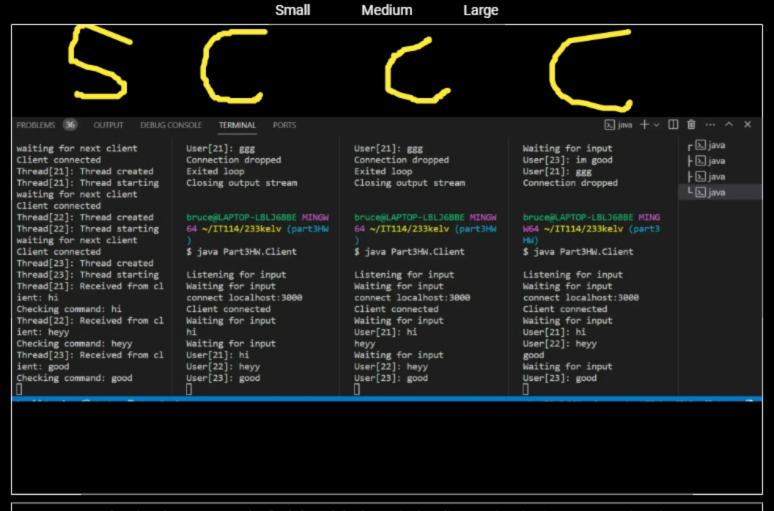
Text: Demonstrate Baseline Code Working

Details:

This can be a single screenshot if everything fits, or can be multiple screenshots

Checklist		*The checkboxes are for your own tracking
#	Points Details	
#1	1	Server terminal/instance is clearly shown/noted
#2	1	At least 3 client terminals should be visible and noted
#3	1	Each client should correctly receive all broadcasted/shared messages
#4	1	Captions clearly explain what each screenshot is showing
#5	1	Include a screenshot showing you grabbed Parts 1-3 correctly and have them in your repository alongside Part3HW

Gallery Style: Large View



I showing the Server on the far left and the last 3 is the clients I showing it connect and texting

Checklist Items (2)

#4 Captions clearly explain what each screenshot is showing

#5 Include a screenshot showing you grabbed Parts 1-3 correctly and have them in your repository alongside Part3HW



^COLLAPSE ^

Task #1 - Points: 1

Text: What feature did you pick? Briefly explain how you implemented it

Checklist		*The checkboxes are for your own tracking
# Points Details		Details
#1	1	Feature is clearly stated (best to copy/paste it from above)
#2	1	Explanation sufficiently and concisely describes implementation (should be aligned with code snippets in related task)

Response:

Coin toss the coin will flip head or tails



Task #2 - Points: 1

Text: Add screenshot(s) showing the implemented feature working (code and output)

Details:

Add screenshots of the relevant code changes AND relevant output during runtime

Checklist		*The checkboxes are for your own tracking
#	Points	Details
#1	1	Output is clearly shown and captioned
#2	1	Code shows relevant snippets that accomplish feature, UCID and date are present in all code screenshots. Relevant captions are included for each screenshot of the code.

Task Screenshots:

Gallery Style: Large View

waiting for next client client connected Thread[21]: Thread created Thread[22]: Thread created Thread[22]: Thread created Thread[22]: Thread starting waiting for next client Client connected Thread[22]: Thread starting waiting for next client Client connected Thread[22]: Thread starting waiting for next client Client connected Thread[22]: Thread starting waiting for next client Client connected Thread[23]: Thread starting waiting for next client Client connected Thread[23]: Thread starting Thread[23]: Thread starting Thread[23]: Thread created Thread[23]: Thread created Thread[23]: Thread created Thread[23]: Thread created Thread[23]: Received from cl ient: his Chacking command: hey Thread[23]: Received from cl ient: hey Checking command: hey Thread[23]: Received from cl ient: psy Checking command: good User[23]: waiting for input User[23]: hi User[23]: psy User[23]: good		Small	Medium Large		
waiting for next client client connected Thread[21]: Thread created Thread[21]: Thread starting waiting for next client Client connected Thread[22]: Thread starting waiting for next client Client connected Thread[22]: Thread starting waiting for next client Client connected Thread[22]: Thread starting waiting for next client Client connected Thread[22]: Thread starting waiting for next client Client connected Thread[23]: Thread starting waiting for next client Client connected Thread[23]: Thread starting Thread[23]: Received from cl ient: hi Checking command: hi Thread[22]: Received from cl ient: heyy Checking command: heyy Thread[23]: Received from cl ient: good Client connected Thread[23]: Received from cl ient: heyy Checking command: heyy Thread[23]: Received from cl ient: good Client command: good Client					
Client connected Thread[21]: Thread created Thread[21]: Thread starting waiting for next client Client connected Thread[22]: Thread created Thread[22]: Thread created Thread[22]: Thread created Thread[22]: Thread starting waiting for next client Client connected Thread[22]: Thread created Thread[23]: Thread starting Thread[23]: Thread starting Thread[23]: Thread created Thread[23]: Thread created Thread[23]: Thread created Thread[23]: Thread created Thread[23]: Received from cl ient: hi Checking command: hi Checking command: heyy Thread[23]: Received from cl ient: good Checking command: good Connection dropped Exited loop Closing output stream Connection dropped Exited loop Closing output stream bruce@LAPTOP-LBLJ68BE MINGW 64 ~/IT114/233kelv (part3HW) 64 ~/IT14/233kelv (part3HW) 64 ~/IT14/233kelv (part3HW) 64 ~/IT14/23k	PROBLEMS 36 OUTPUT DEBUG CO	DNSOLE TERMINAL PORTS		∑java +∨ [[) 前 ··· ^ ×
Thread[22]: Thread created Thread[22]: Thread starting Waiting for next client Client connected Thread[23]: Thread created Thread[23]: Thread created Thread[23]: Thread starting Thread[23]: Thread starting Thread[21]: Received from cl ient: hi Checking command: hi Thread[22]: Received from cl ient: heyy Checking command: heyy Thread[23]: Received from cl ient: good Checking command: good Checking Checking Chec	Client connected Thread[21]: Thread created Thread[21]: Thread starting waiting for next client	Connection dropped Exited loop	Connection dropped Exited loop	User[23]: im good User[21]: ggg	├∑ java ├∑ java
	Thread[22]: Thread created Thread[22]: Thread starting waiting for next client Client connected Thread[23]: Thread created Thread[23]: Thread starting Thread[21]: Received from cl ient: hi Checking command: hi Thread[22]: Received from cl ient: heyy Checking command: heyy Thread[23]: Received from cl ient: good	64 ~/IT114/233kelv (part3HW)) \$ java Part3HW.Client Listening for input Waiting for input connect localhost:3000 Client connected Waiting for input hi Waiting for input User[21]: hi User[22]: heyy	64 ~/TT114/233kelv (part3HW) \$ java Part3HW.Client Listening for input Waiting for input connect localhost:3000 Client connected Waiting for input User[21]: hi heyy Waiting for input User[22]: heyy User[23]: good	W64 ~/IT114/233kelv (part3 HW) \$ java Part3HW.Client Listening for input Waiting for input connect localhost:3000 Client connected Waiting for input User[21]: hi User[22]: heyy good Waiting for input	

Checklist Items (1)

#2 Code shows relevant snippets that accomplish feature, UCID and date are present in all code screenshots. Relevant captions are included for each screenshot of the code.





Task #1 - Points: 1

Text: What feature did you pick? Briefly explain how you implemented it

Checklist		*The checkboxes are for your own tracking
#	Points	Details
#1	1	Feature is clearly stated (best to copy/paste it from above)
#2	1	Explanation sufficiently and concisely describes implementation (should be aligned with code snippets in related task)

Response:

Dice roller the Dice would roll



Task #2 - Points: 1

Text: Add screenshot(s) showing the implemented feature working (code and output)

Details:

Add screenshots of the relevant code changes AND relevant output during runtime

C	hecklist		*The checkboxes are for your own tracking
	#	Points	Details
	#1	1	Output is clearly shown and captioned
	#2	1	Code shows relevant snippets that accomplish feature, UCID and date are present in all code screenshots. Relevant captions are included for each screenshot of the code.

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```
反 java + ∨ □ 前 ··· ∧ ×
PROBLEMS (36)
                                         TERMINAL
                                                                                                                                          r ∑ java
Client connected
                                   User[21]: /coin
                                                                                                        connect localhost:3000
                                                                                                                                          - I java
Thread[23]: Thread created
                                   User[22]: The Coin got head
                                                                     User[22]: /coin
                                                                                                        Client connected
                                                                                                                                          - ≥ java
Thread[23]: Thread starting
                                                                     /coin
                                                                                                        Waiting for input
Thread[23]: Received from cl
                                   User[22]: /coin
User[22]: The Coin got tail
                                                                     Waiting for input
User[22]: The Coin got tail
                                                                                                        /coin
                                                                                                                                          L ≥ java
                                                                                                        Waiting for input
ient: /coin
Checking command: /coin
                                                                                                        User[23]: The Coin got tai
                                   User[22]: /coin
User[23]: The Coin got tail
                                                                     User[22]: /coin
User[23]: The Coin got tail
Checking command: The Coin g
ot tails
                                                                                                        User[23]: /coin
Thread[23]: Received from cl
                                                                                                        /coin
                                                                     User[23]: /coin
ient: /coin
                                   User[23]: /coin
                                                                                                        Waiting for input
Checking command: /coin
                                   User[23]: The Coin got tail
                                                                     User[23]: The Coin got tail
                                                                                                        User[23]: The Coin got tai
Checking command: The Coin g
ot tails
                                   User[23]: /coin
                                                                     User[23]: /coin
                                                                                                        User[23]: /coin
                                                                        code
```

Checklist Items (1)

#2 Code shows relevant snippets that accomplish feature, UCID and date are present in all code screenshots. Relevant captions are included for each screenshot of the code.





Task #1 - Points: 1

Text: Reflection: Did you have an issues and how did you resolve them? If no issues, what did you learn during this assignment that you found interesting?

Checklist		*The checkboxes are for your own tracking
#	Points	Details
#1	1	An issue or learning is clearly stated
#2	1	Response is a few reasonable sentences

Response:

In this project, I worked on processing commands like "/coin" and "/roll" and implemented the fundamental features of a server-client system. By including additional features, boosting speed, and improving the user experience, developers may build on this framework even further.



Task #2 - Points: 1
Text: Pull request link



URL should end with /pull/# and be related to this assignment

URL #1

https://github.com/bigbabykev/233kelv/pull/7

End of Assignment