

COMP3150 ASSIGNMENT 2 REPORT

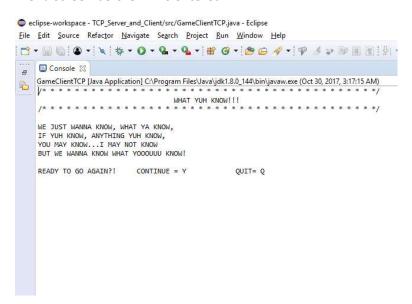


A.Mohammed

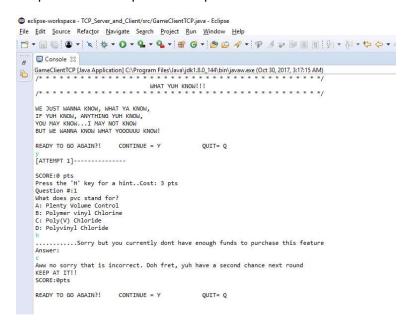
814000595

INPUT & SCREENSHOTS of running program

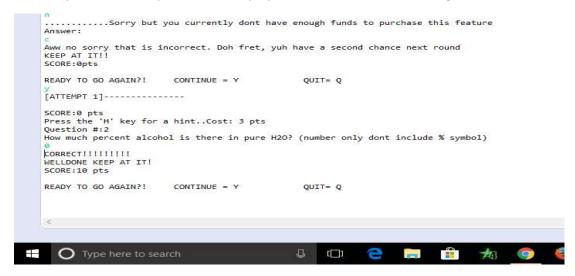
Menu screen before "Y" is entered:



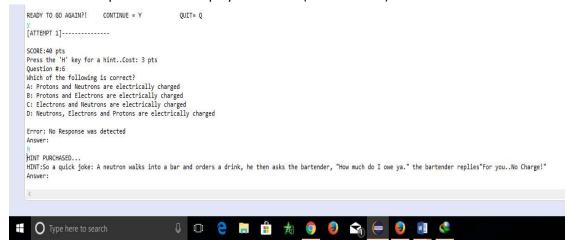
1st question loads up and attemp to use the hint while score is 0 then incorrect answer is given



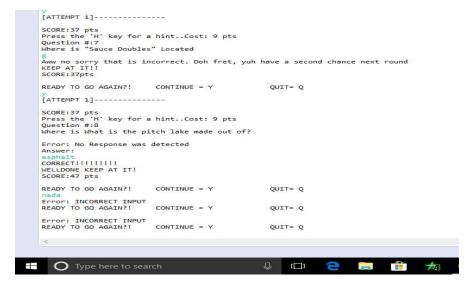
Selects "y" to continue question 2 is displayed and the correct answer is given



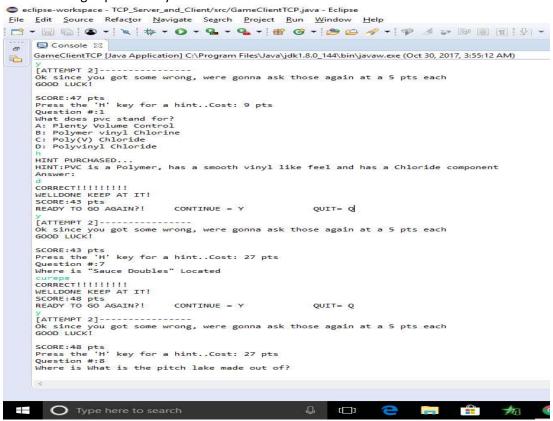
If the player pushes enter with no typed response it lets them know there was no response detected and a successful purchase and display of the hint (extra feature) shown



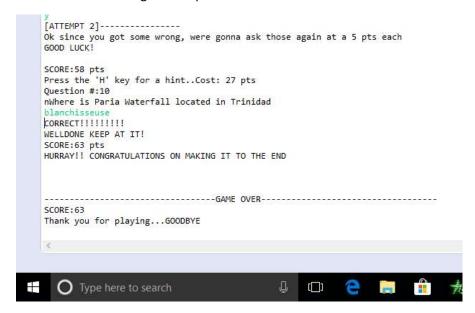
Hint cost is seen to have increased from 3 to 9 pts here, hence an exponential increase in the cost of hints the more they are used. And when players enter anything but "Y" or "Q" at intervals between questions



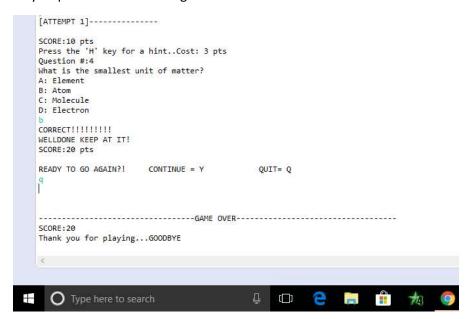
Attempt 2 questions are in order and points increasing by 5 with ever correct answer and cost of hints still increasing exponentially 3^x where x is the hint counter.



Game over after doing out all questions



Player quits in mddle of the game



CONCLUSION:

I Antonio Leonardo Mohamemd hearby say that he work submitted is all my own and was done soully by me.

My program seemed to execute and function quite well from all the scenarios I could think of to test. Main difficulty was really getting the client to read more than one line at a time and getting it to ignore the regular readLine ending, the "\n" character. However I managed to fix this by using a loop to stop reading data from server upon reaching a "#" character.

The extra feature I had to method overload it cause I thought it was more functional to have one ExtraFeature(int count) and one ExtraFeature(int count, int points) which both return an integer cost amout with the only differnce being the possibility of the latter returning a -1 denoting that the player would not be able to purchase the hint statement.

Along with the ExtraFeature function use a temporary queue to house the hints for the many questions.

What was quite usueful and what I was quite happy I included was the catch exception for when the server could not send a message. This helped in debugging as I knew the message being sent to the client was lacking a "\n#" at the end of the message.

I think I completed 100% of the requirements and the program runs fairly smoothly