

Final project - workshop in communication networks

Introduction:

• You are going to implement an online version of the game "The Chase" (for those who don't know).

The game is recommended to be viewed on the YouTube channel [here](#)).

• The project focuses on server-client and socket type programming .programming

• Recall that in the server-client architecture there are two sides: the client - the receiver Service and the server - which provides a service.

• In our case-

o The client side will be a computer user who will receive questions and answer them.

o The server side will be the computer responsible for managing the game (its services include).

Satisfy questions, track game progress, report to user about

readiness and regarding his progress, etc. (in addition, the server side will be the chaser in the case

Ours will be automatic and answer questions correctly with high probability.

game rules-

Below is the schedule of the game-

steps board	steps
chaser	chaser step
	1
	2
	3
	4
	5
	6
bank	7

start phase-

At the beginning of the game the player receives an opening message in which the player is asked if

He wants to play and he has two options-

1. The player answers yes and then a connection is made with the server. The game continues to stage one.

2. The player answers no and then the connection is not made and we will end the running of the user program.

Level A' -

The user receives three questions according to which he will determine how much money he has in the account, only if

A user got at least one question right, he continues to the next step, otherwise he starts over at first-

1. If he was right in 0 questions: 0 NIS and he moves to the beginning stage.
2. If he was right in 1- questions: NIS 5000 and he continues to the next level.
3. If he was right in 2 questions: NIS 10,000 and he continues to the next level.
4. If he was right in 3 questions: NIS 15,000 and he continues to the next level.

stage B'-

The user has 3 options-

1. Start from step 3 with the current amount.
2. Get one step closer to the chaser and start from step 2 for twice his current amount
3. Take one step away from the chaser and start from step 4 for 1/2 of his current amount.

For example, if the user was right in step A in two questions, then his account currently has 10,000

and it will have 3 options as in the following drawing-

	1
chaser	2
	3
player	4
	5
	6
bank	7

Step C-

Now the chase begins -

Ask the user a question and show him 4 possible answers. The user has 2

Options

1. Choose any answer.
2. The user will be able to request a "saving wheel" and download 2 wrong options out of 4 The answer options (the user has only one lifeline).

- If the user chooses the correct answer, he advances one step towards

The bank, if he is wrong, he stays at that stage.

- The chaser also has to answer the question but in our case the chaser is a computer And so we programmed it so that it would have a probability of 0.75 to answer correctly.

- If the chaser chooses the correct answer he also moves one step towards

The bank, if he is wrong, he stays at that stage.

• After the user and the chaser have answered, an update of the current board will be sent to the user

which includes 3 things -

1. How much money the user has.
2. What stage is the user at?
3. What stage is the chaser at?
4. Is there a lifeline left?

We will return to phase C if the game is not over yet or we will go to the final phase if the game is over

Finished (the meaning of finished is explained in the finishing stage).

For example - suppose that the current turn is the state of the board -

	1
chaser	2
	3
player	4
	5
	6
bank	7

chaser and the player are right -

	1
	2
chaser	3
	4
player	5
	6
bank	7

Chaser is right and player is wrong-

	1
	2
chaser	3
player	4
	5
	6
bank	7

Chaser is wrong and player is right

	1
chaser	2
	3
	4
player	5
	6
bank	7

Chaser is wrong and player is wrong (no change) -

	1
chaser	2
	3
player	4
	5
	6
bank	7

Finishing stage -

The game will end in one of two of the following options-

1. The user reaches the bank (step 7) before the chaser has enough time to reach his step.

A win for the user.

2. The chaser reaches the user's stage before the user reaches the bank. Victory of the chaser.

After the game ends, the server will send a summary message of the game (victory of the chaser or the player (and will offer the user to start a new game and if so we will go back to step 1

Otherwise the server will disconnect the user and terminate the user program run.

For example-

- If this is the state of the board -

	1
	2
	3
chaser	4
player	5
	6
bank	7

So if the chaser gets the current question right and the player gets it wrong then the chaser wins.

)Note- If the user is right and the chaser is right in this case then it doesn't count that the player but simply both advance a step to the bank).

- If this is the state of the board -

	1
	2
	3
chaser	4
	5
player	6
bank	7

So if the player is right and the chaser is right or wrong (doesn't matter) then the player conductor.

Implementation instructions-

- You need to implement the code in the Python language in two Ubuntu operating systems In one the server will run and in the other the client will run (and there is communication between them).
 - You are responsible for preparing a selection of at least 15 questions that includes 4 answers (correct and three incorrect).
 - The chaser is automatic and not a human user. The chaser's probability of success is .0.75
 - You need to allow up to 3 players to connect to the server at the same time.
 - Each player who connects is a game by himself and has no connection to other players.
 - If a fourth player tries to connect, send him a rejection message stating that there is already one three players
 - I want you to invest the best of your time in
 - o Writing quality and documented code.
 - o Accompanying documentation.
 - o Understanding the theoretical material behind the code.
- Documentation guidelines -
- Document the code carefully.
 - Add a documentation page in PDF format that contains explanations in the following order -
 2. Operating system, programming code in Python + Python version (3 or higher).
 3. Theoretical background.
 4. An explanation for each code file you made -< classes and functions.
 5. Explanation of opening -> socket with which protocol did you open, what are the commands and an explanation of each command and so on.
 6. Screenshots of running the program from start to finish for both options of the progress of the game (winning player + winning chaser).

**MAKE SURE THAT THE SERVER USER WILL BE CALLED YUVAL-SERVER
and CLIENT SIDE WILL BE CALLED YUVAL-CLIENT
MAKE SURE THE SCREENSHOTS AND EVERYTHING HAS NOTHING TO DO WITH
YOUR NAME OR ANY OF UR LANGUAGE.**