Assignments

Assignment #3 - In progress

Draft - In progress
Submitted
Returned

Assignment Details

Title
Assignment #3
Due
Nov 8, 2022 11:55 PM
Number of resubmissions allowed
Unlimited
Accept Resubmission Until
Nov 10, 2022 11:55 PM
Status
Not Started
Grade Scale
Points (max 100.00)
Modified by instructor
Oct 31, 2022 5:53 PM

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Instructions

CS3357 Assignment #3

Fall Session 2022

Purpose of the Assignment

The general purpose of this assignment is to continue to explore network programming and more advanced concepts by a extending the text adventure game built in Assignment #2. This assignment is designed to give you further experience in:

- writing networked applications
- the socket API in Python
- writing software supporting a simple protocol

Assigned

Tuesday, October 18, 2022 (please check the main course website regularly for any updates or revisions)

Due

The assignment is due Tuesday, November 8th, 2022 by 11:55pm (midnight-ish) through an electronic submission through the OWL site. If you require assistance, help is available online through OWL.

Late Penalty

Assignment Project Exam Help

Late assignments will be accepted for up to two days after the due date, with weekends counting as a single day; the late penalty is 20% of the available marks per day. Lateness is based on the time the assignment is submitted.

https://powcoder.com

Individual Effort

Your assignment is expected to be an individual effort. Feel free to discuss ideas with others in the class; however, your assignment submission must be your own work. If it is determined that you are guilty of cheating on the assignment, you could receive a grade of zero with a notice of this offence submitted to the Dean of your home faculty for inclusion in your academic record.

Add WeChat powcoder

What to Hand in

Your assignment submission, as noted above, will be electronically through OWL. You are to submit all Python files required for your assignment. If any special instructions are required to run your submission, be sure to include a README file documenting details. (Keep in mind that if the TA cannot run your assignment, it becomes much harder to assign it a grade.)

Assignment Task

You are required to implement a more robust client-server text adventure game, leveraging the client and server you implemented as part of Assignment #2. A sample implementation of Assignment #2 will be provided in the near future (after all submissions are in) that you can use as a basis for development in this assignment, if you would rather do that than use your own previous work.

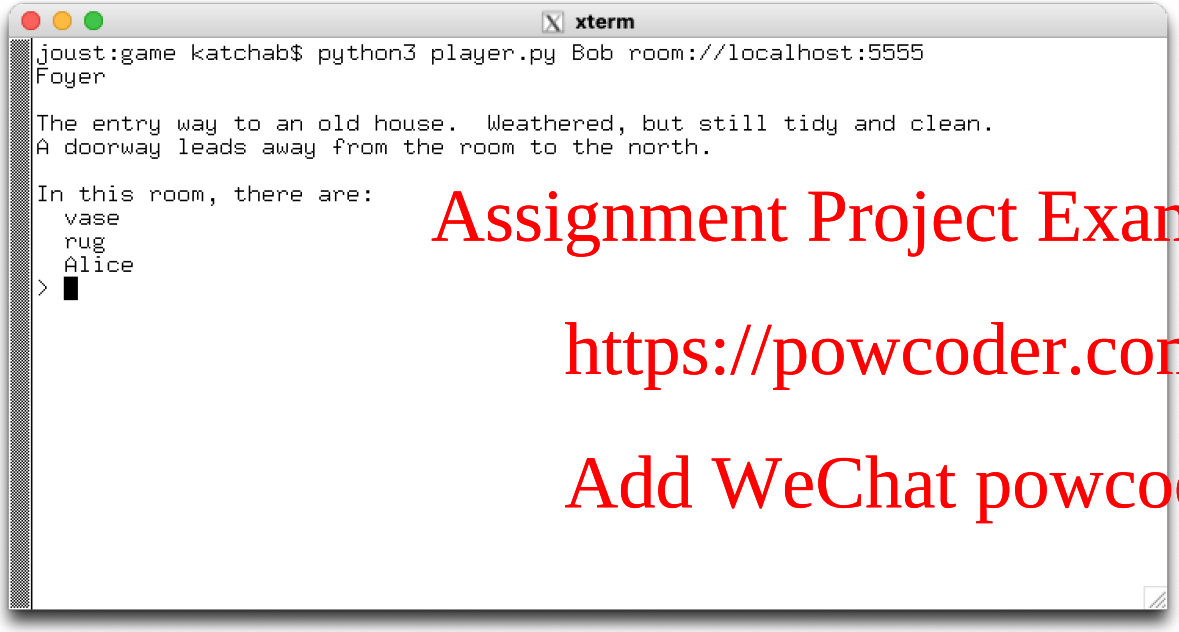
For this assignment you will be adding a collection of new features to the application. The main addition is that the game now must support multiple simultaneous players across multiple servers. Players will be able to see and say things to one another and also move from room to room

(which means from server to server).

Some Particulars

Here are some specific requirements and other important notes:

- When a player receives a room description from the room server, either by joining a room or by using the look command, other players will now be listed among what they can see in the room. A player should never see themselves listed as being present in the room; only other players in the room should be visible. For example, if Alice is in a room and Bob enters the room, Bob will see Alice in the description of the room.

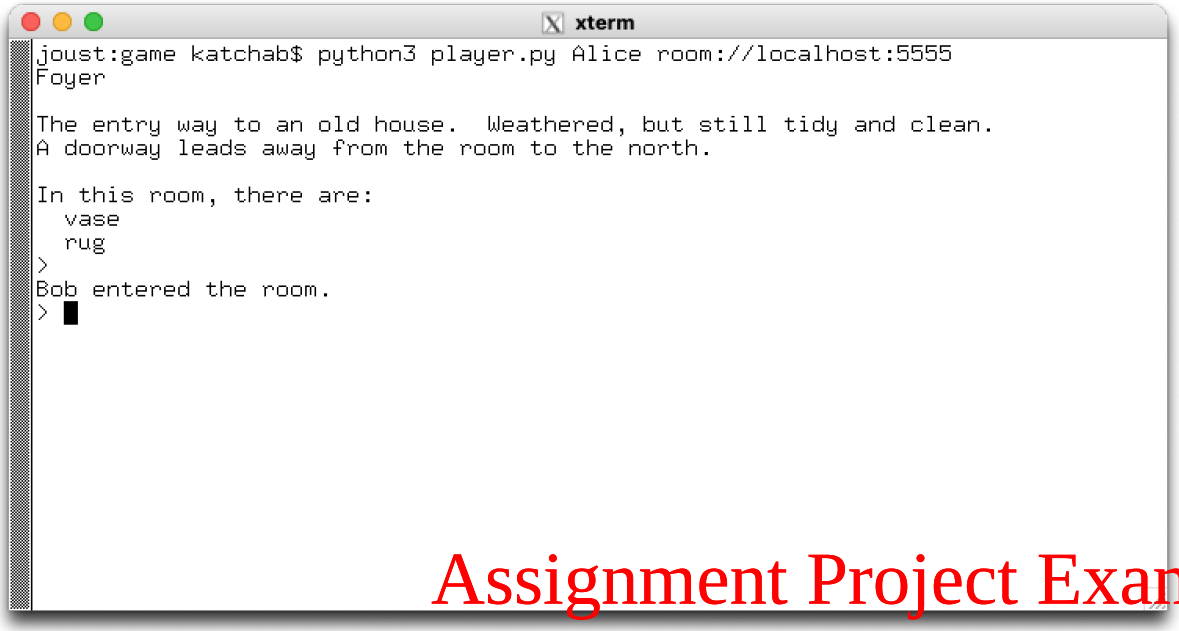


Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- Likewise, when another player enters the room a player is in, the server should notify them of the new player so it can be reported accordingly. Continuing the above example, when Bob enters the room and Alice is already in the room, Alice will be notified of Bob's entry.



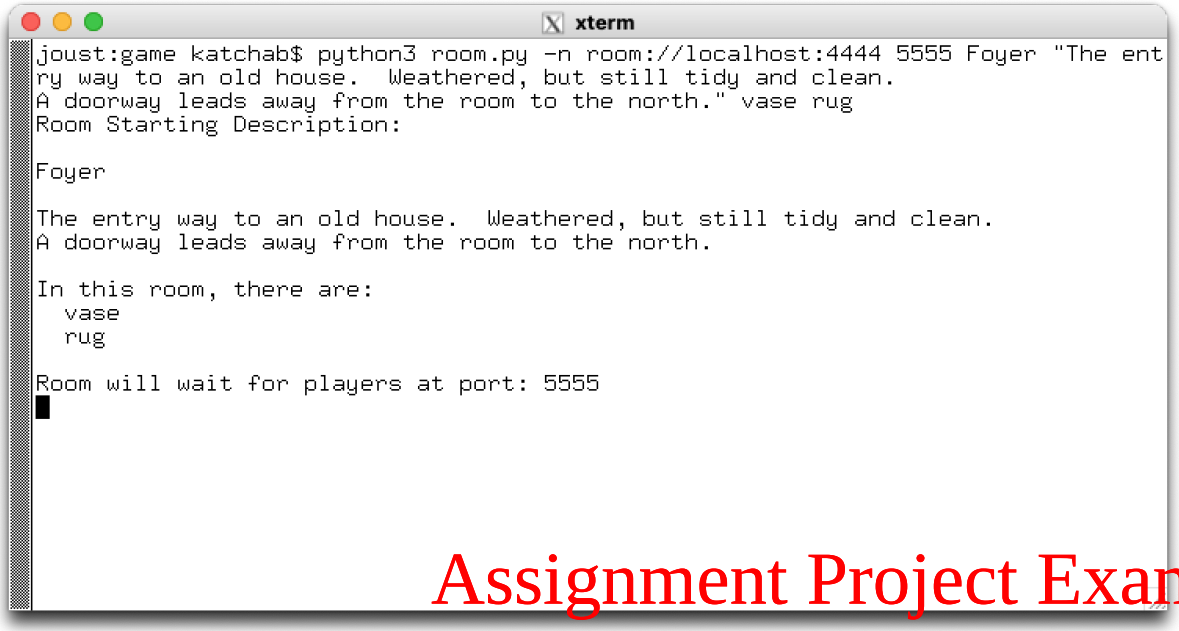
Assignment Project Exam Help

This does mean that the server must be able to send messages to the client on its own and not simply in response to messages from the client. More on that below.

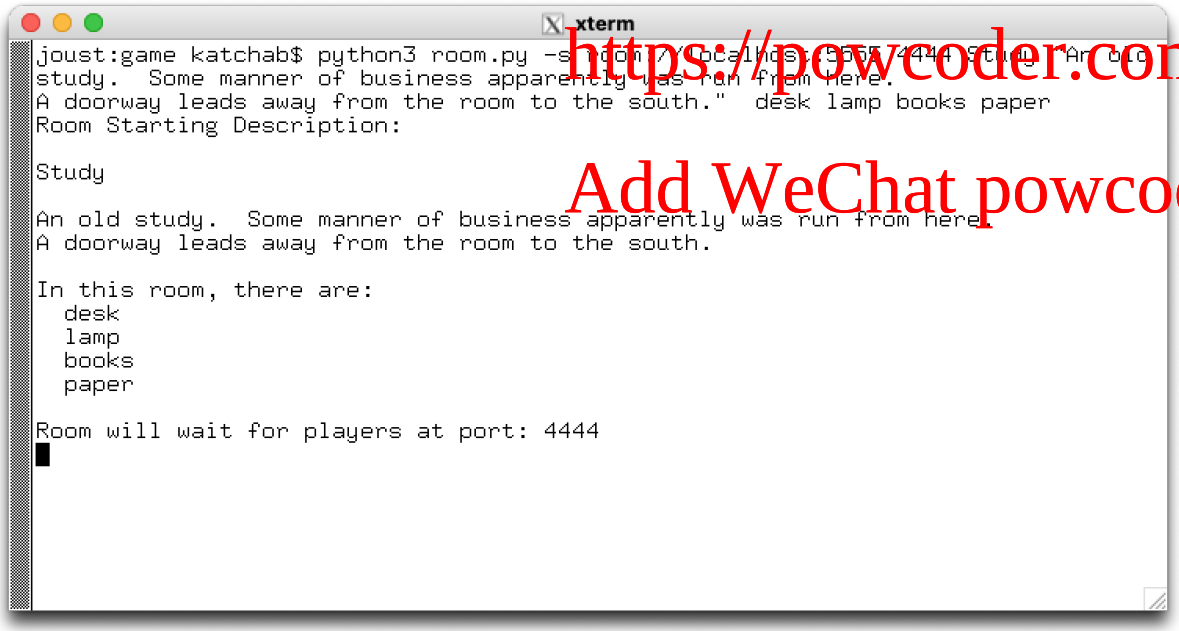
https://powcoder.com

- Players will be able to move from room to room now, using the commands north, south, east, west, up, and down. When a room server is started, optional parameters must now be given for those directions leading away from the room hosted by the server. These parameters will indicate the direction and give an address of the room in that direction in the form room://host:port (like the client uses in connecting with servers.) For example, if the Foyer was running on port 5555 on localhost, and the Study was located on port 4444, they might be executed as follows:

Add WeChat powcoder

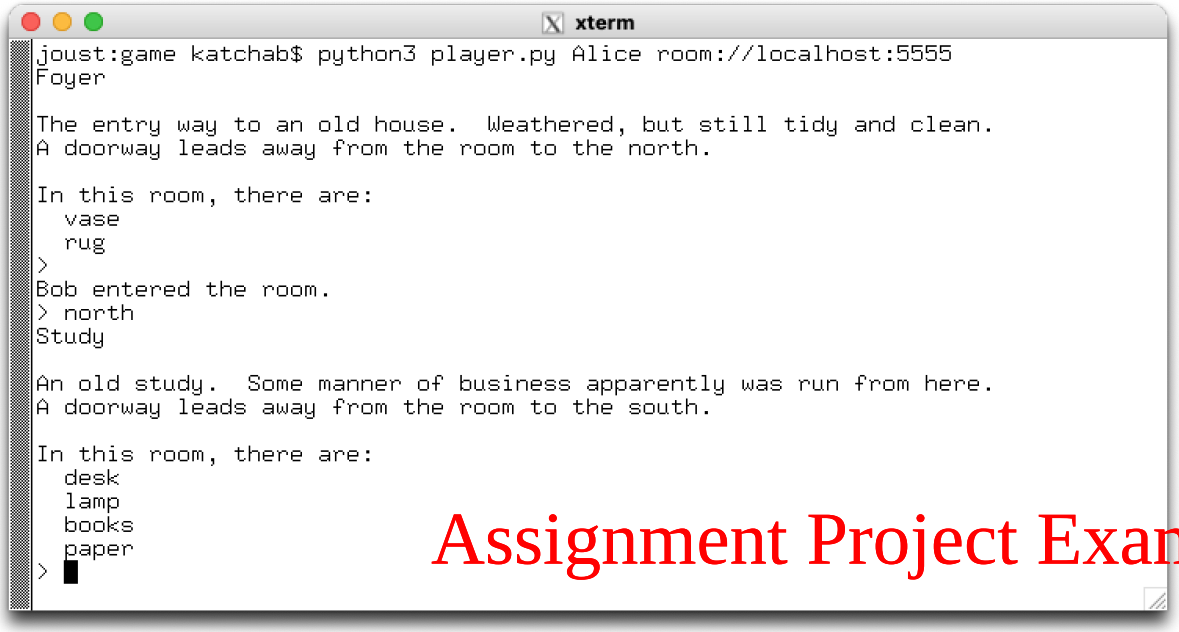


Assignment Project Exam Help



https://powcoder.com
Add WeChat powcoder

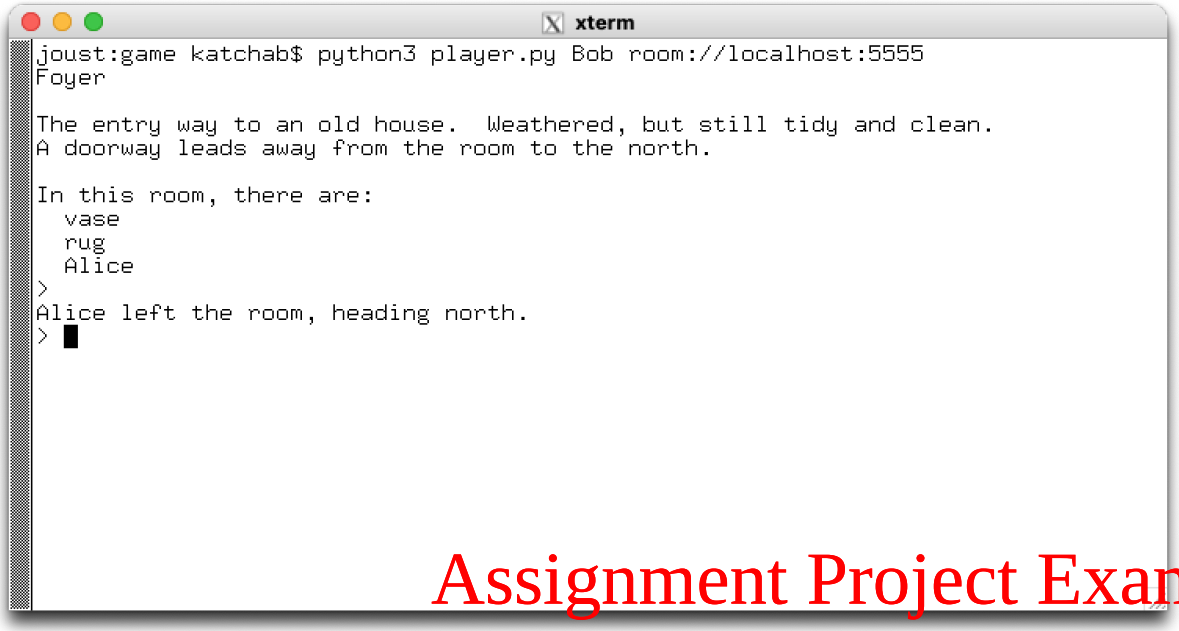
As you can see, the -n and -s options are used for specifying rooms to the north and the south. Likewise, -e, -w, -u, and -d will allow you to specify rooms to the east, west, up, and down directions respectively. Rooms do not need connections to other rooms in every direction; you only need to specify the directions that are actually needed and used in the game. Continuing the previous example, if Alice was to head north using the north command, it would appear like this:



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

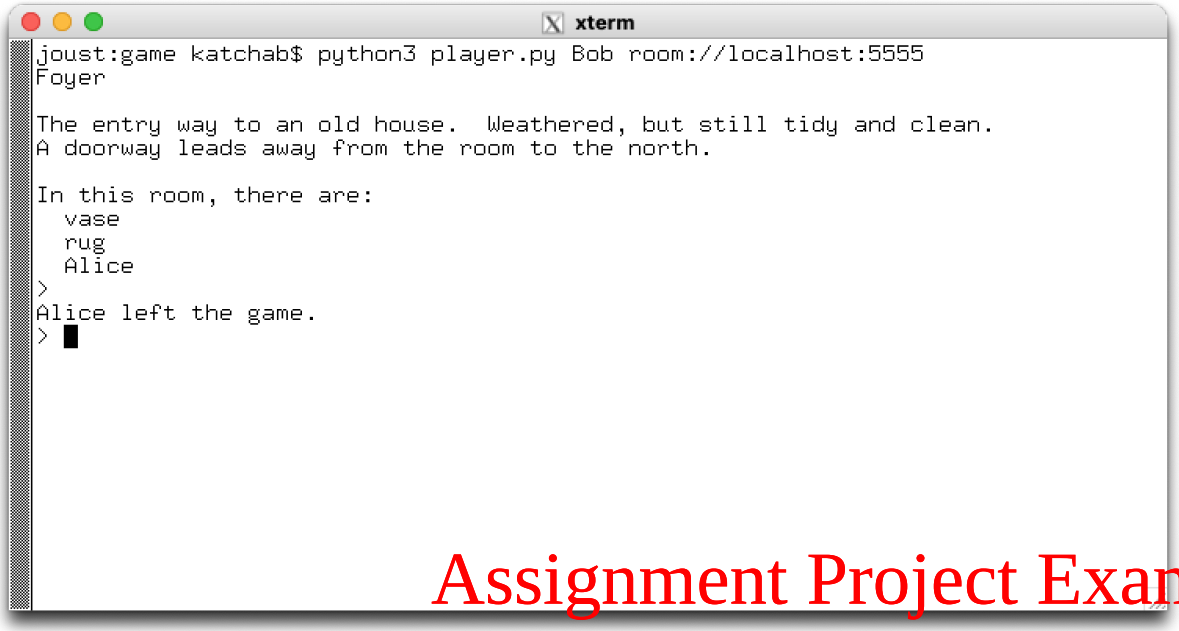
When a player leaves a room and moves to another, they take all their inventory items with them to the new room. This does mean that players can pick something up in one room and drop it in a different room. (Please note that players can never pick up other players though!) In effect, moving from one room to another amounts to exiting the one server and joining the other.

- Players will also see other players leave the room that they share with them, being notified of such occurrences by the server. Continuing the above example, if Alice headed north, the server would notify Bob that Alice left, indicating the direction she headed in at the same time.



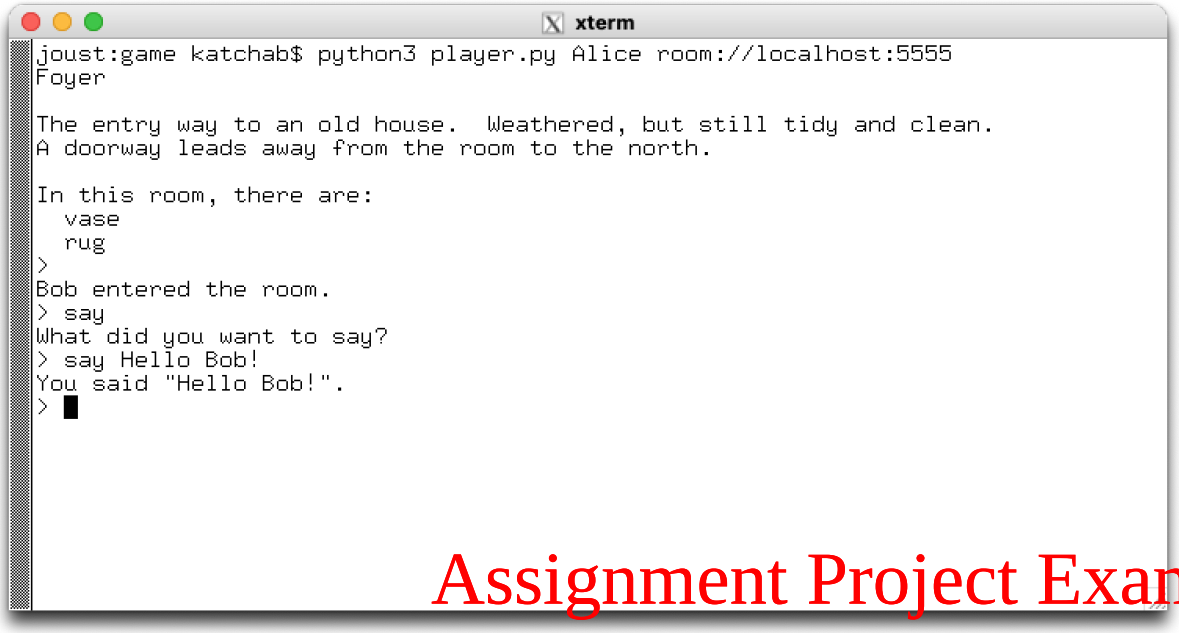
Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Alternatively, if a player in the same room leaves the game entirely, the server should still notify the players in that room, but use a slightly different message. For example, if Alice had instead exited the game using the exit command, Bob would have seen a message like what is depicted below.



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

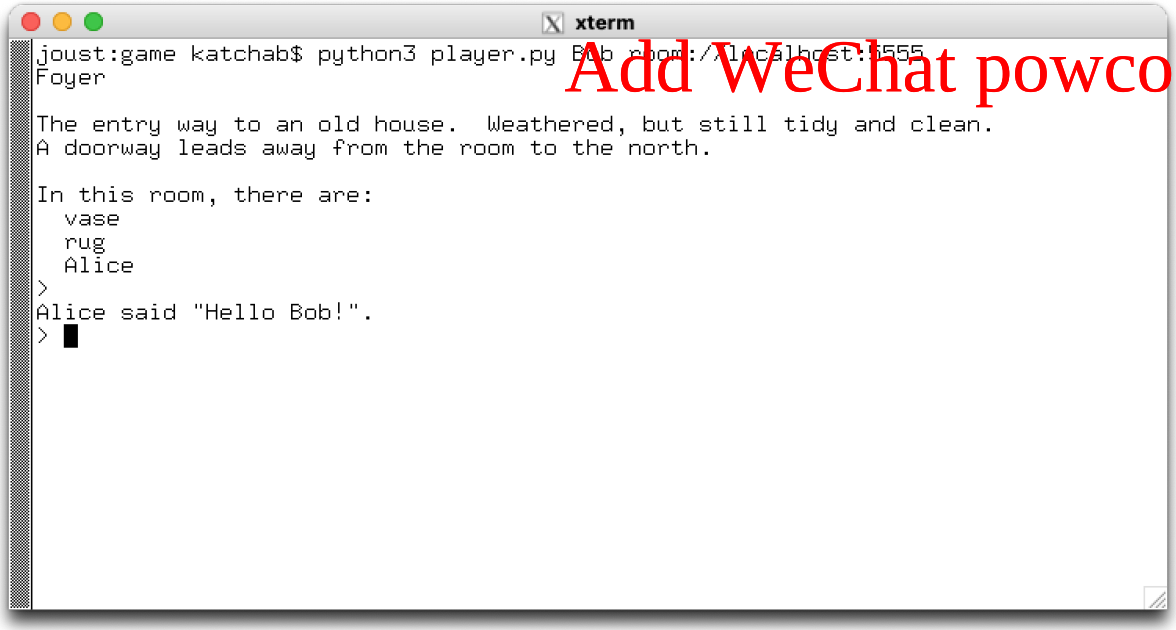
- Players can now say things using the "say" command. Whatever message they write on the line following the say command is passed to the server, and the server then distributes it to all other players currently in the room, who then immediately display it. (The player who spoke does not receive a copy of things in that fashion, but rather gets an acknowledgement that they said what they said.) If no additional text is given, the player is given an error message notifying them of this. For example, if Alice said something in the same room as Bob, her display would look like this.



Assignment Project Exam Help

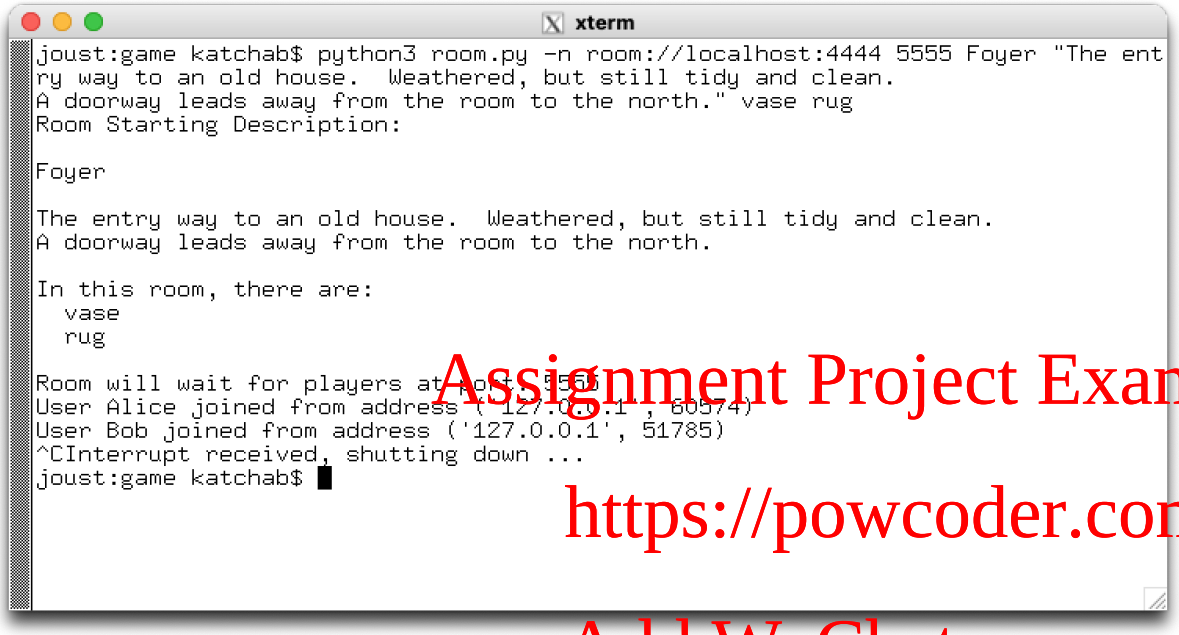
Bob would get immediately notified of Alice speaking by the room server, without having to enter any commands. His display would appear as follows.

https://powcoder.com



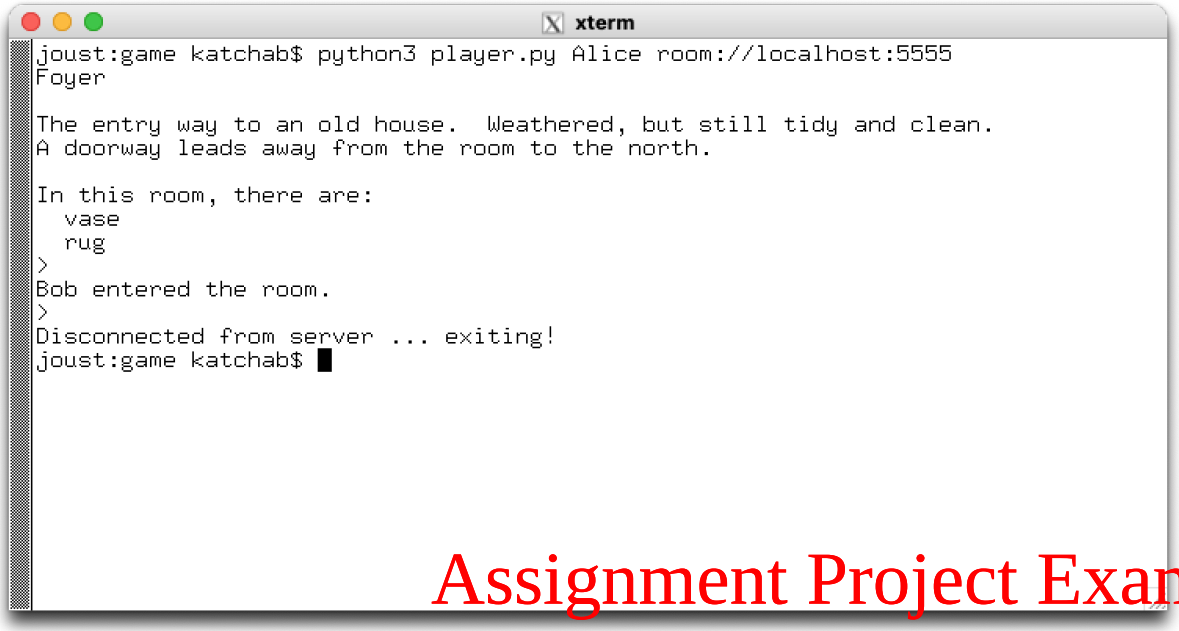
Add WeChat powcoder

- Now that the room server can message clients whenever they need to, and clients will receive them and process them even if they are waiting for user input, the server can now have a signal handler that notifies players when it is shut down using a Ctrl-C signal. When that is done, it should send a "disconnect" message to every client on the server. Clients receiving such messages should exit right away. For example, let's suppose the Foyer room server is shut down. Its display would look like:



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Note that it also prints a message when shutting down. Alice would receive the disconnect message from the server and shut down immediately. Her display would appear as follows.



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

- As noted above, your game clients now need to be able to both accept new commands typed by the user and messages sent by the server at the same time. (There is no way to tell which will be happening when ... both could happen at any time.) Messages from the server will include activity of other players in the same room (coming, going, saying things) as well as potential disconnect messages. To enable this, I recommend using the selectors package in Python or the lower-level select package. This will allow you to wait from input from multiple sources (the keyboard and your client's socket) at the same time instead of just a single source (the keyboard) in the previous assignment.

You are to provide all of the Python code for this assignment yourself, except for code used from the Assignment #2 implementation provided to you. You are **not** allowed to use Python functions to execute other programs for you, nor are you allowed to use any libraries that haven't been explicitly allowed. (If there is a particular library you would like to use/import, other than those listed above and things like socket, os, or sys, you must check first. You cannot use things like socketserver, for example.) Your server should be called room.py and your client should be called player.py. You may name any other files you create for your program in some other appropriate way (though you likely don't need any others). All of these files must be submitted with your assignment.

As an important note, marks will be allocated for code style. This includes appropriate use of comments and indentation for readability, plus good naming conventions for variables, constants, and functions. Your code should also be well structured (i.e. not all in the main function).

Please remember to test your program as well, since marks will obviously be given for correctness! You should experiment with various scenarios, including what is depicted above, to make sure that the game client and server are working correctly.

Submission

Assignment Text

This assignment allows submissions using both the text box below and attached documents. Type your submission in the box below and/or use the Browse button or the "select files" button to include other documents. **Save frequently while working**.

Source

Styles | Format | Font | Size

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Words: 0, Characters (with HTML): 0/1000000

Attachments

No attachments yet

Select a file from computer Choose File No file chosen or select files from 'Home' or site

Proceed Preview Save Draft Cancel Don't forget to save or proceed!