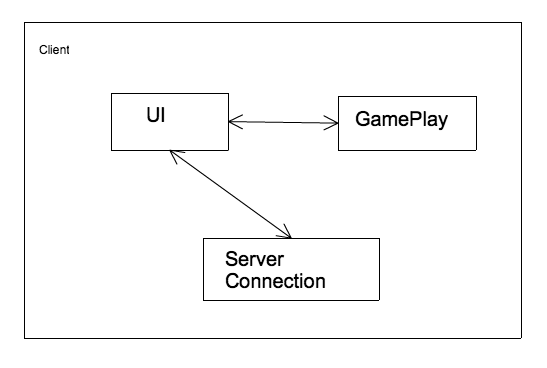
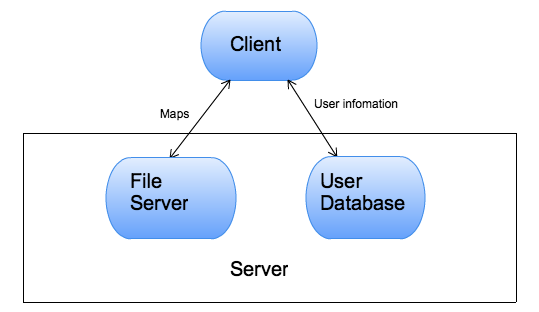
**Incremental and Regression Testing**

Authors: Jeanne Deng, Huyue Gu, Chengcheng Xu, Neng Gu, Jiali Du

**Classification of Components:**



The project can be divided into two main components, server part and client. For the server part, we have two independent components, the user information server on EC2 and the map server on S3. For the client side, we have three components, UI, gameplay and server connection. Gameplay and server connection is built within the UI.

Form of incremental testing:

We will use the bottom up approach which we will test each subparts that have no dependence (gameplay, client) and then test the part which has dependence (UI).

UI:

- UI will call gameplay

input to gameplay: user input

output from gameplay: next stage of the game

- UI will call server connection

input to server connection: user requests

output from server connection: response from server

Client:

- server connection will call map server

input: request to upload/download/listing the maps

output: the content of the map or a list of maps

- server connection will call user server

input: username/password/game process

output: game process list

\*server and game play are independent

**Incremental and Regression Testing**

Testing client with S3 (S3 is a service of AWS, it cannot be tested by us):

*Incremental tests:*

A. Client 001, Upload file to S3, Severity 2

B. Create a new S3 uploader and upload a file

C. The file successfully uploaded and I can see it in the S3 bucket

A. Client 002, Retrieve file list from S3, Severity 1

B. Call the MapList method of client to retrieve a list of maps from S3

C. The method returns a list of all the existing maps, but some of the prefix for the map names is not correct and some of the maps inside user folder are not returned

A. Client 003, Download file from S3, Severity 1

B. Call the download method with the file name

C. A new file is created in the local dir with the correct content

A. Client 004, Retrieve user saved data, Severity 3

B. Call the server to get the list saved states

C. The server returns the correct list of the saved states

A. Client 005, Create user, Severity 1

B. Call the server with a new username and password

C. The server returns true upon successfully create of a user, but the server will allow me to create a user multiple times. Which is a bug. We fixed it so that a user can be created only once.

*regression test:*

I fixed defects that the user cannot have private maps, but after fixing the problem, I find out the listing of the map is wrong, the listing will not list the maps that are created by the user. - severity 3

Testing GamePlay:

*Incremental tests:*

A. Client Gameplay, move north, Severity 1.

B. When a player is in a room, type ‘n’ to move to the room which is bordered in the north.

C. Description of new room is printed to screen.

A. Client Gameplay, move south, Severity 1

B. When a player is in a room, type ‘s’ to move to the room which is bordered in the south.

C. Description of new room is printed to screen.

A. Client Gameplay, move west, Severity 1

B. When a player is in a room, type ‘w’ to move to the room which is bordered in the west.

C. The message “Can’t go that way” is printed to screen.

A. Client Gameplay, move east, Severity 1

B. When a player is in a room, type ‘e’ to move to the room which is bordered in the east.

C. Description of new room is printed to screen.

A. Client Gameplay, inventory command, Severity 1

B. Type inventory command (i) in order to list all items in player’s inventory.

C. Print “Inventory: empty”.

A. Client Gameplay, take item command, Severity 1

B. Type “take item” command to change item ownership from container to player’s inventory.

C. Print “Item(item) added to inventory”.

A. Client Gameplay, open container command, Severity 1

B. Use “open (container)” to check the items in the container.

C. Prints contents of container in format "(container) contains (item), (item), ..." and makes those items available to pick up.

A. Client Gameplay, open exit command, Severity 1

B. If a player reaches the end of the game, use “open exit” command to end the game.

C. The game is not ended properly, no message is printed. Not pass the test

A. Client Gameplay, read item command, Severity 1

B. Type “read (item)” in order to read the information on a object.

C. Prints writing on object.

A. Client Gameplay, drop item command, Severity 1

B. When a player wants to change item ownership from inventory to present room, use “drop  (item)”.

C. Prints “(item) dropped”.

A. Client Gameplay, put item command, Severity 1

B. When players want to change item ownership from inventory to declared container, use “put (item) in (container)”.

C. Prints “put item fails.”

A. Client Gameplay, turn on item command, Severity 1

B.  If players want to activate item which is available in inventory, type “turn on (item)”.

C. Prints “You activate the (item).”

A. Client Gameplay, attack with command, Severity 1

B. When players want to attack creatures with some item in inventory, use “attack (creature) with (item).”

C. Prints “You assault the (creature) with the (item).” Otherwise, prints “attack fails.” It passed the test.

*regression test:*

I fixed the API with “open exit” command. However, after fixing this bug, our game would not be loaded properly.  ---Severity 1

Testing ZorkGUI

*Incremental tests:*

A. Zork UI 001, Login Panel, Severity 1

B. Click login from initial screen, user name and password request pop up, enter and continue with OK button.

C. I originally set them to be in different class, say Login.java + Register.java but I found out later that its not a good design when considering the whole GUI, I have to keep track of Login Panel and Register Panel by different threads even for the easiest solution like keep a static external variable. Otherwise the Observer Pattern must be applied to the original ZorkGUI.java. So I solved by putting them together and put all according action event in the ActionPerformed{......}in and also the global variables into ZorkGUI.java and solved it.

A. Zork UI 001, Verify User Name and Password, Severity 1

B. Click login from initial screen, user name and password request pop up, enter and continue with OK button.

C. It was built separately from sever at the beginning so I stored these info at local directory. Later on I changed it to the way that call the methods in sever to build an online connection.

A. Zork UI 002, Save Game, Severity 2

B. Pull down the tool bar from upper left corner, click ‘Save’ to store the current game status.

C.Even when it is logged  in successfully, we still need to set the save button from ‘Archive’ to be non-clickable at the beginning since when there is not even a game scene, we cannot save. Soled it by set the save game button at the beginning.

A. Zork UI 003, Load Game, Severity 2

B. Click the ‘Load’ from the pulled down tool bar on left upper corner, choose the stored file you wanted from correct directory confirm by ’OK’ and it will import the saved archive about previous game status.

C. The file chooser was always going back to origin directory but not goal directory and referenced API and solved this issue.

A. Zork UI 004, Import Map, Severity 3

B. Click Import Map from the tool bar on upper left corner and import the map you made or downloaded for this game through out correct directory.

C. The panel includes ‘Next Page’ , ‘Previous Page’, after tested click them for several times the error occurs. It was because of the count of page number and finally fixed by setting the right current page number.

*regression test:*

-Naming for lots of variables created additional confusion so I renamed them to make them more readable next time when making changes to dode.

-Panel display had some error about not displaying properly the reason is not refresh the panel every time after adding it to the main panel contained in JFrame.

Testing User Server

*Incremental tests:*

A. Server 001, Connection Command, Severity 1

B. Telnet ec2-23-23-28-183.compute-1.amazonaws.com 8089

C. No defects. Terminal shows:

“Trying 23.23.28.183...

Connected to ec2-23-23-28-183.compute-1.amazonaws.com.

Escape character is '^]'.”

A. Server 002, Register Command, Severity 1

B. While telneting server, send “Register:;:testcase:;:testing”, then send the same message again.

C. Defect found. Terminal shows “Register:;:success” follow with “Register:;:success”

A. Server 003, Login Command, Severity 1

B. While telneting server, send “Login:;:testcase:;:testing”, then send “Login:;:testcase:;:wrongpw” and “Login:;:wronguser:;:testing” and “Login:;:wronguser:;:wrongpw”

C. No defect. Terminal shows “Login:;:success” follow with “Login:;:fail” and “Login:;:fail” and “Login:;:fail”

A. Server 004, User Info Upload Command, Severity 2

B. While telneting server, send “SaveData:;:testcase:;:1:;:test cases”, then send “SaveData:;:wronguser:;:1:;:test cases”

C. Defect found. Terminal shows “SaveData:;:succes” follow with “SaveData:;:success”

A. Sever 005, User Info Download Command, Severity 2

B. While telneting server, send “GetData:;:testcase”, then send “GetData:;:wronguser”

C. Defect found. Terminal shows:

“GetData:;:0:;:(empty):;:1:;:hello world:;:2:;:(empty):;:3:;:(empty):;:4:;:(empty):;:5:;:(empty):;:6:;:(empty):;:7:;:(empty):;:8:;:(empty):;:9:;:(empty)” follow with “Connection closed by foreign host.”

*regression test:*

- I fixed “Register” defect by making username in table User as a unique key. I tested it again, no new defect found.

- I fixed “SaveData” and “GetData” defects by changing how server requests query since query itself has no error, so now server does not send query if user is not exist. I tested it again, no new defect found.