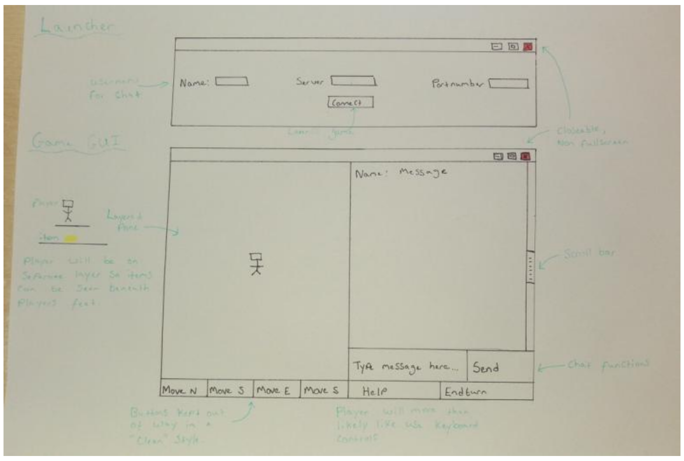
Dungeon of Doom Documentation

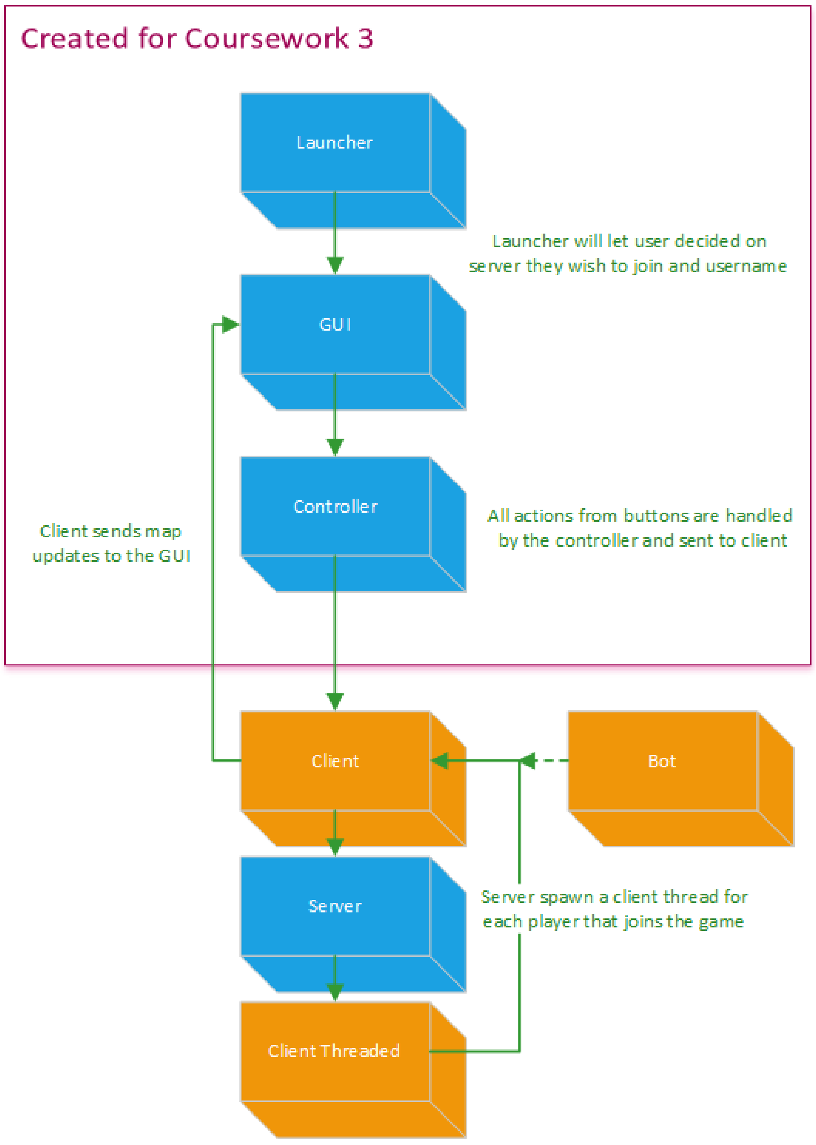
Intended features	Points	Self evaluation	Works	Special Instructions
A) Basic GUI	15	Yes		none
B) Advanced Client GUI	10	Yes, in the time I had I feel as if my GUI sticks to my original design. It's both functional and easy to use. If I had more time I would of liked to add status bars such as health, gold amount etc		none
C) Updating GUI	10	Yes, the GUI not only updates when other players are in the game but also if other players spawn in your vision they will appear. The same as if they quit the game, they will disappear from the map.		none
D) Advanced Chat	15	Yes. As chat is such a broad term it was very difficult to keep the scope relevant to the project. I feel as if coloured text and html support allows an adequate amount of options to the end user. Additional features such as player icons or emojis would have been nice; however I simply did not have enough time to implement these features.		Supports use of HTML in chat and there are colour/font changing commands.

GUI Design

While I have Photoshop and could of spent many hours creating a draft design, I thought a more realistic approach would be to hand draw a design and follow it as closely as possibly rather than drawing an optimistic GUI in Photoshop that cannot be achieved within the time frame. In the critical analysis section I will explain in further detail about why I wanted the layout like this and why I choose low resolution pixel art instead of highly detailed pixel art.


Launcher
Username for chat
Name:
Server
Portnumber
Connect
Connect game
Closeable, Non fullscreen
Game GUI
Player
Item
Layered Pane
Player will be on separate layer so items can be seen beneath players feet
Name: Message
Scroll bar
Type message here...
Send
Chat functions
Move N
Move S
Move E
Move S
Help
End turn
Buttons kept out of way in a "Clean" style
Player will more than likely like use keyboard controls

High level Design of Classes and Algorithms


Created for Coursework 3
Launcher
Launcher will let user decided on server they wish to join and username
GUI
Controller
All actions from buttons are handled by the controller and sent to client
Client sends map updates to the GUI
Client
Bot
Server
Server spawn a client thread for each player that joins the game
Client Threaded

This is a high level design of the game. The purple box represents classes which have been newly created for this coursework. Classes represented by an orange box will more than likely have to be updated or changed in order for certain features to work. For example updating the map as soon as a player spawns. Bot does not have its own GUI and therefore only has a client when it joins the game.

In the latest coursework version of “Dungeon of Doom” I’ am expecting to create three new classes are as well as using the previous classes from coursework two. The three new classes will consist of the Launcher class, GUI class and the Controller class. The GUI, Controller and Client class will attempt to follow the MVP model (Model-view-presenter), while the Launcher class is used as a setup screen to provide initial options to the user, such as which server they wish to join.

GUI class
The GUI class will add all components to a main window frame, this allows the player to interact with the game in a more intuitive way in comparison to command line. The visual representation of the map will be created with PNG image files and will be updated using the look reply. The GUI class will more than likely have to use a GridBagLayout, which gives the most flexibility when positioning components.

Controller class
The controller class will implement ActionListener and KeyListener. The buttons added in the GUI will interact with the client by sending String actions through to the Controller class. I'm planning to also allow the user to interact with WASD and the arrow keys.

Launcher class
The launcher will act as a main entrance point to the game, enabling the user to choose their name and sever they wish to join. Overall the class should be fairly short with a few methods to handle input error.

Software process model
While it may seem a little nonsensical to discuss which software model I used; especially since Dungeon of Doom is fairly small in comparison to many games not to mention the fact it's made entirely by a single person, I can't help but notice that I naturally seem to take an iterative design approach. I first collate ideas in Pseudo code (design), create an at least basic version of this in actual working code (implementation) , then refine this (evaluation) , and finally starting this cycle again when a new part of the program is being worked on. I think this type of software model matches my programming style and is suitable in most situations. While I appreciate that I'm still learning core fundamentals of Computer Science every day, the way in which I approach writing a program seems to be reaching a level that I'm happy with (Of course this will only be refined with time and experience).

Test plan

Test Scenario	Test case	Test steps	Expected Result
StartGame GUI	Text Fields handle wrong user input e.g. Port number greater than 65535.	Run launcher, enter characters or number larger than 65535 in port number text field.	Yes - Dialog box to alert user.
Game GUI	GUI is displayed correctly on different systems and screens.	Run Dungeon of Doom on both LCPU, NT Kernel. Run Dungeon of Doom on different resolutions.	Yes - Runs on both LCPU using Unix and Windows 7 using NT Kernel. Runs on 1920x1080 as well as 1360x768.
Game GUI	Buttons all function and carry out correct commands e.g. move n	Click all buttons on screen.	Yes- All carry out the correct functionality
Game GUI	Keyboard input can be	Use arrows	Yes - Only when the chat

	used	keys/WASD when game loads. Use those keys when chat is focused.	is not focused.
Game GUI	The map updates when other players join/leave game. Should work with Bot as well as players	Run two instances of Dungeon of Doom then see if player spawns in visually and when they leave they visually disappear.	Yes - Players/ bots joining game update map.
Game GUI	Chat prints in correct formatting.	Try sending multiple messages.	Yes - Colours, formatting and commands are all displayed.
Game GUI	Colour codes function correctly	Type /r and message into chat	Yes – chat colour commands work.
Network	Leaving game advances player turn.	Join game with two players and then quit with one player.	Yes. * Confirmed bug found with server code provided. If first person in server quits and other players join afterwards the player turn is not advanced.

I think it will be a good idea to use JUnit in conjunction with Maven to carry out some basic tests for Dungeon of Doom. These will not only speed up the process but also as I've never used this feature before it will hopefully become a learning experience.