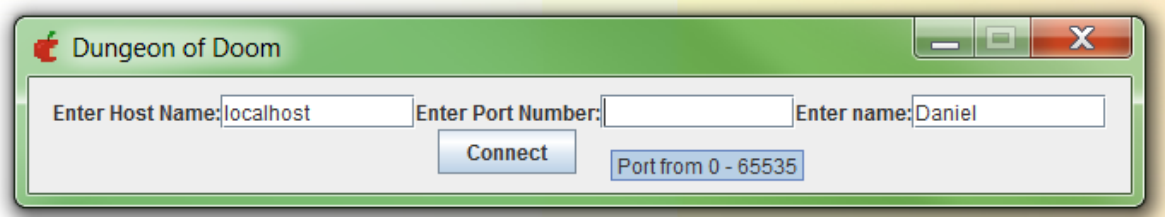
Dungeon of Doom Documentation

Launcher



Dungeon of Doom

How to play

In your preferred choice of kernel use the following commands to play the game.

How to run Dungeon of Doom:

$ Java StartGame

How to run an AI Bot play for you:

$ Java Bot *servername portnumber* - Where *servername* and *portnumber* can be replaced with desired variables. Eg localhost 54879

How to setup and run your own server:

$ Java Server *mapname* – Where mapname can be filled in **optionally** with a file path to a custom map to run on the server. You can make your own maps in notepad.

Aim of the game

The aim of the game is to collect the amount of gold required by the map and make your way to the exit before other players. You have 6 Action point per turn. There are also some items to help you along the way.

Keyboard Shortcuts

wasd – move north, move west, move south and move east.
Arrows keys – Move in direction of arrow
h key – Help
p – pickup
e – endturn
return/enter key – send message when in chat

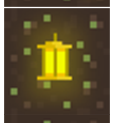
Item and Enemies

Blue player – This is you on the map.

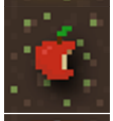
Red player – This is an enemy on the map

Gold coin – Collect as much the map requires.

Exit – Make your way here after collecting the required.



Lantern – Will expand how far you can see on the map.

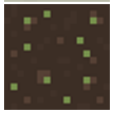


Health – Apples are healthy, pick them up to increase your health (No way to currently lose health)

Armour –Pick up to increase your armour (No way to currently lose armour). Can only be picked up once.

Sword –Pick up to increase your attack (No way to currently lose attack). Can only be picked up once.

Wall – You cannot move through or into these



Floor – You can walk on this, some floor tiles will have items on

Chat

The chat supports rich text and HTML. There are some colour shortcuts implemented into the chat. The chat is also the way they game informs you of any information you need to know such as gold required, action points left etc.

/r – red
/b – blue
/o – orange
/g – green

/bld – bold
/itl – Italics

Example /b this is the colour blue

Critical analysis

What did you do right?

As an aspiring game developer, I'm fairly proud of what I produced considering I have little to no experience of GUI programming I definitely achieved all the goals I wanted to with this project. The GUI in my personal opinion looks clean, clear and is easy to use.

I'm most proud of the artwork used within in the game as most of the core mechanics are not mine, I felt the artwork should be original and of high quality. The artwork is inspired by older pixel based games such as Pokémon and also by modern games such as Minecraft. I wanted to blend this old pixel art style and put a slightly modern twist on it. I most probably spent too much time on the artwork, even creating GIFS for some of the items in game however I feel as if it paid off and really adds to the overall polish of the game.

What went wrong?

Nothing went explicitly wrong when creating Dungeon of Doom, more often than not it was inexperience with SWING that caused the most frustration. GridBagLayout requires careful planning and while I stuck to my design I often would setup components in the wrong positions. The solution to this problem was to break the problem down, and put all components that belong together on a single panel, and then add those panels to a "master" panel.

The JUnit and Maven testing did not go as planned either. One of the main causes of this is due to the fact that they should be created when initially creating the system. I spent too much time trying to learn how to use them and therefore simply had to stop and focus on the actual game.

Cleverness!

One really cool feature that I enjoyed adding into the game was using HTML to make the chat a more enjoyable fun experience, especially if the user knows HTML they can do limitless thing with the font and even "program" within the game!

Other minor features to do with the aesthetics such as the player being on a separate layer so items can be seen under their feet and animated GIFs for some items add to the overall polish of the game.

Improvements

There were naturally many additions that could have been made to the base game. For example, while printing the commands in chat is informative and very old school RPG, fallout-esque, the chat box itself can get quite crowded and it would make more sense to create status bars for health, gold and action points. Other minor additions in the specifications that could have been implemented are attack, animated player movements, images over chat etc. Most couldn't be done purely to time constraints.

As well as minor generic additions, one idea that I thought could be interesting and suitable for the theme of our game, would be to randomly generate maps similar to the game "Rogue." It would provide a way to balance the game more as neither the Bot nor human would rely on memory to win the game and would certainly improve replay-ability.

On reflection I would have completely changed my AI to use a Heuristic algorithm such as A* or even D*, constraints on time and my lack of knowledge means that I did not implement such algorithms; however once I feel more confident and comfortable with them I will more than likely switch to them for both their efficiency (A* being roughly O(n) length of the solution path) and speed. While a perfect AI is of course what any Computer Scientist would strive to create, even more so in an assessment situation, it should be noted that the Bot must be beatable and display some sort of realism. Building on this it may be interesting to create distinct AI's such as an aggressive chaser AI or a more passive defensive AI that could defend you.