Portoflio task 1

You will design and implement a simple, social, text-based adventure game (like *Zork* <http://www.web-adventures.org/cgi-bin/webfrotz?s=ZorkDungeon>, or *Text-based Multiplayer Shooter* <http://eigen.pri.ee/shooter/> ). *Note: Zorg is also mentioned in the book* ***“Ready Player One”***[*http://www.goodreads.com/book/show/9969571-ready-player-one*](http://www.goodreads.com/book/show/9969571-ready-player-one) *(made into a movie in 2018).*Another inspiration could be this turn-based DOOM remake, with text-mode graphics: <https://www.dosgames.com/game/doom-the-roguelike>

***TASKS:***

1. (We did this together the first lecture) *Decide a style/theme for your game, collect images and ideas for the look and feel of your game; write down a title and a short description.*
2. **Task 1 –** Look at the code in **/99\_simpleGame/** which is the beginning of a game, very similar to the game you will create, and combine it with the code in **/readWrite3[client]/** which shows how a nodejs server can save and reload data to and from a JSON file, so that it can remember data even if the server goes down and is restarted.

Based on the code in these 2 examples, create a single webpage that is a complete, small game. The game has to be single-player; you will use node.js only to serve the game page, and javascript in that page to run the game for the single player playing the game. The goal of task is to use javascript to create a text-based, single-player game: you should jQuery to make the client webpage. To complete this task you need to fulfil the following **use case**:

* + Start your node.js server
  + Open localhost:3000/ and the HTML page containing the game should be loaded from a node.js server.
  + When the page opens it should show you a table with the best scores (i.e. the highscore table) for all players so far. This table should be loaded via AJAX from a JSON file on the node.js server.
  + The game page should then ask you your username (perhaps just with a simple *prompt()* ), and you might type “Alice”, so that you score will later be recorded as the score for player “Alice”.
  + The game will start by telling you information about a room (or location) where your player is. E.g. “You are in a dark room, no windows, but there is a candle and some strange shapes in the corners of the room. There is a door on the west wall. What do you want do to?”. You should implement a form to allow the player to type his/her commands (as you can see in the code for **/99\_simpleGame/**).
  + You have to implement at least 3 rooms (or places) for the player to explore, and provide basic navigation commands to the player, in the client. I.e. the player should start in the “main” room each time the game is loaded, and be able to move from room to room via simple text-based commands.
  + Some commands (for example “search” the room) could increase your score, and some might diminish your health. If you reach 0 HP you should die and the game should stop. You should also have a command to “quit” the game, which also makes it stop.
  + When the game stops, your client should do an Ajax call to the server and pass the score of the player (for example “Alice”, 1000 points) to the server, that keeps a table of all scores. If the score for Alice in the table is less than 1000, then the new score for Alice will become 1000, in the table in the server. This is the highscore table, and the server will also save it to a JSON file after every change, when the server gets an Ajax call from the client.
  + When the server is killed and starts again it should first see if there is a JSON file with the highscore table, and if there is, load it in its score table. If not, the score table is initiated as an empty table. See code in **/readWrite3[client]/**, for an example of this mechanism.

It is not important at this stage that the player’s avatar is able to interact with items or NPCs that might exist in the rooms.

**General Notes**

While developing your game, you would design some (simple) HTML-based GUI, using javascript. Simple static graphics can be used in the GUIs, for example to show a picture with a map of the rooms, or the state of the current room, or perhaps show score/health/energy in a HUD (<https://en.wikipedia.org/wiki/Head-up_display_(video_gaming)> ).

**If you have another, potentially very different idea** for the tasks, please write a 1-page proposal (like this one) and get it approved (by me), as soon as possible**.**

**Delivery dates**All 2 tasks by **mid May 2021.**Remember: **in any case both should be delivered** (and approved by me) **before** the end of the semester, because they will be the basis for our discussion at the exam.