

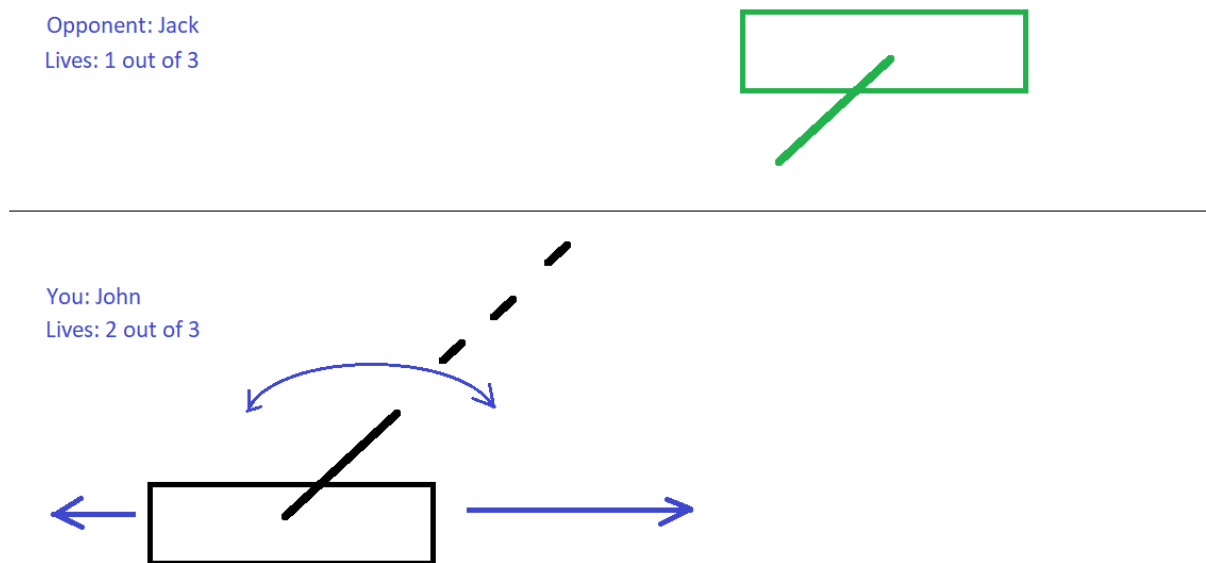
## Individual assignment 1

### Hide Folder Information

#### Instructions

You have chose from these options

- 1-** Adding user management to HangMan game you already developed ( sing up, ranking etc )
- 2-** Adding user management to Quiz creator game you already developed ( sing up, rating, multiple users taking same quiz and see their ranking etc ). wysiwyg editors like Markdown or ckEditor could be used too
- 3-** Real time multiplayer game . you can use Websocket ( or Socket IO) to develop this.( No database related development needed for this project)
  - Two users play a very simple tank game, trying to shoot each other. tanks only move left and right with arrow keys.
  - At the beginning of the game prompt user to pick a name



Some resources: Your game is much much much simpler compare to this game was implemented using websocket: <http://tankanarchy.herokuapp.com/>

**4-** Online video chat ( no database needed) . You use WebRTC or compatible ones to develop this ( No database related development needed for this project)

- 2 or more users to do online video chat .
- at the beginning of the chat, before entering, the users get prompted to enter a name

Here is one of the many good references: <https://www.pusher.com/tutorials/webrtc-video-call-app-nodejs/>

More detailed requirements will be posted here including

- you need to host
- no game engine is allowed. For front end only HTML5, CSS3, JavaScript, JQuery, React, Bootstrap are allowed . No External APIs that provide video chat are allowed. Socket.IO is allowed. WebRTC libraries ( not services) are allowed .
- For the assignment you cannot use Django . but nodeJS is allowed. that's because Django is a framework\*, nodeJS is not. Express is not. Please email me if you had any questions please .
- you need to post it on git ( from day one) and to record feature developments, improvements, bug fixes ect
- Code has to be fully OOP
- all arrow functions ( new: if possible)
- preferred use of let and const
- no bugs at all
- all messages displayed to users have to be stored in a separate file
- mobile friendly
- clear steps to use your app ( no user wondering around what-to-do-now?!)
- No hard coded string messages in your code. Place the strings ( messages get displayed to users) in a separate j file
- passwords have to be **encrypted** in your DB

#### **FAQ:**

- Are we allowed to use Facebook/G+/ AWS Cognito login?
- Yes as you still need to develop db and back-end

- Can we develop different multiplayer game than the Tank one?

-- Yes but 1- you still are limited to using what mentioned above for development; 2- You need to inform me and get my approval

- Can we use firebase for backend?

-- Yes

- Can we use passportjs?

-- No

**deliverable:**

Entire source code, files in their own sub folder folders in a zip file yourLastName\_1\_x.zip where x is the option number of the project you've selected

together with a backup of your database filled with test data ( or fake data)

**Note:** Please remove the *node\_modules* folder if you have it

+ a pdf containing the title of your assignment and instructions on how to install your app for testing/marketing

in the comment section provide the url of your hosted assignment and url of your gitup

**Attn:**

You will lose -0.5 mark for every hour late submission

You have to understand every single line of your code. You may be asked technical questions about your assignment, for every wrong answer you will lose -2 marks

The project you demo during marking has to be identical with the one you submitted onto D2L ( configs server related setting could be different), if not you lose additional 50% mark or you may get totally 0 for all your labs and this assignment

\* What is a framework? One easy way to test is that if you install and set up a "software" framework and don't do any extra programming, you still have a fully functioning web application.

Due Date

Nov 3, 2018 11:30 PM