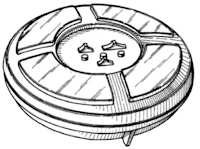
**Welcome to PYmon**

PYmon is an online multiplayer version of the famous Hasbro game simon

[](http://lh3.googleusercontent.com/qXdmXKyRLzxf0SspNm8QBsOSsXUoDADTo-3q-zLf0kd3qdk2P9fXsoeg-wV7b8cW0MXt6yVktsFrWsJZ2Q6OS6u4=s200)

**The game flow:**

1. Players click on an "open" game (open for players to join)
2. As long as the game status is open other players can join.
3. Clicking the "play" button will play the sequence once
4. After all players listened to the sequence the game status changes to "on" (from now other players can join as viewers)
5. The first player who joined the game will get the turn
6. On each turn the player should click the next color of the sequence (one guess per turn)
7. If you failed, you are out.
8. If you are correct the turn goes to the next player
9. If the sequence is finished all remaining players are considered winners

**Built With**

* [Bottle (Python)](https://bottlepy.org/docs/dev/) - The web framework used
* [React (JS)](https://reactjs.org/) - FrontEnd Framework
* [Webpack](https://webpack.js.org/) - Frontend module bundler

**Authors**

* **Gilad Navot** - *Initial work* - [navotgil](https://github.com/navotgil)

**Getting Started**

First, visit [The Demo](https://py-mon.herokuapp.com/) (hopefully it will be awake) Clone the project and make sure you have all of the tools below installed.

**Prerequisites**

* [A computer](https://www.ebay.com/itm/386-Computer-AMD-80386DX40-40Mhz-8Mb-RAM-Windows-3-11-Old-DOS-Game-PC/153285622134)
* [Python 2.7.x](https://www.python.org/downloads/release/python-2715/)
* [Pip](https://pypi.org/project/pip/)
* [Npm](https://www.npmjs.com/)
* [Heroku CLI](https://devcenter.heroku.com/articles/heroku-cli) (needed for deployment)
* [Remote MySql Server](https://www.db4free.net) (needed for deployment)

**Installing**

1. Create a new "pymon" folder under your dev folder.
2. cd into the folder.
3. clone this repository

git clone https://gitlab.com/itc-bootcamp/assignment12-pymon.git

1. run pip and install the required libraries

pip install -r requirements.txt

1. cd into "/frontend" folder, run npm and install the required libraries
2. cd frontend

npm install

1. run webpack to build the react files into the dist folder. webpack is configured to watch fie changes, so this cmd window should remain open.

npm run build

1. create a local MYSQL DB on your machine for now follow the instruction on this [section](#installing-the-db) Update your credentials in the dbutils.py file!!!!

**Installing the DB**

Update your credentials in the dbutils.py file!!!!

connection = pymysql.connect(host='db4free.net',

user='<your user>',

password='<your password>',

db='<your db>',

charset='utf8',

autocommit=True,

cursorclass=pymysql.cursors.DictCursor)

Create a database for the project. You can use:

CREATE DATABASE <database\_name>

And run the following commands on your MySql workbench to define the database

CREATE DATABASE pymon;

use pymon;

CREATE TABLE `game` (

`id` int(11) NOT NULL,

`name` varchar(30) NOT NULL,

`status` varchar(30) DEFAULT 'open',

`created` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`sequence` varchar(100) NOT NULL,

`creator` varchar(30) NOT NULL DEFAULT 'anonymous',

`step` int(11) NOT NULL DEFAULT '0'

) ;

CREATE TABLE `player` (

`id` varchar(30) NOT NULL,

`avatar` varchar(100) NOT NULL DEFAULT 'anonymous',

`created` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ;

CREATE TABLE `playergame` (

`game` int(11) NOT NULL,

`player` varchar(30) NOT NULL,

`status` varchar(30) NOT NULL DEFAULT 'new',

`created` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ;

ALTER TABLE `game`

ADD PRIMARY KEY (`id`);

ALTER TABLE `player`

ADD PRIMARY KEY (`id`);

ALTER TABLE `playergame`

ADD KEY `game` (`game`),

ADD KEY `player` (`player`);

ALTER TABLE `game`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=51;

ALTER TABLE `playergame`

ADD CONSTRAINT `game` FOREIGN KEY (`game`) REFERENCES `game` (`id`) ON DELETE RESTRICT ON UPDATE RESTRICT,

ADD CONSTRAINT `player` FOREIGN KEY (`player`) REFERENCES `player` (`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

COMMIT;

**Deployment**

The project is "Heroku Deploy Ready"

In order for the project to work online you will have to use a remote MySql DB one option is to use the following [service](https://www.db4free.net)

1. Set a DB onine (following this [step](#installing-the-db) again, this time online)
2. Don't forget to update the dbutils.py credentials
3. Create a new Heroku app
4. Git push heroku master

**License**

This project is licensed under the MIT License

**TODO:**

Split into groups of up to 10 people :) AND GET THE JOB DONE!

* Implement the filters on /games page
  + **description:** (should allow users to filter games according to status)
  + **desing:** A selected filter should be css "brown" color ("OPEN" should be selected on load)
  + **constraints:** frontend implementation (HTML/CSS/JS)
* Implement a back button in /game page.
  + **description:** allow users to go back to the /games page
  + **design:** total freedom
  + **constraints:** Add React component "BackBtn"
* Fix Bug - player can press more than one button per turn
  + **description:** This is against the rules, we need to somehow block the user from being able to do that
* Implement a more impressive "Game Over"/"Game Won" prompt
  + **description:** currently there are none, try to make something nice...
  + **design:** total freedom
  + **constraints:** total freedom (follow application structure)
* Implement highscores page
  + **description:** A standalone page showing the top 10 users with the heighest score (most games won) descending order.
  + **design:** total freedom (follow application design)
  + **constraints:** total freedom (follow application structure)
* Implement the ability for a player to select one of 5 avatars
  + **description:** On the /start page add the ability to select an avatar (the DB already contains that field) The avatar should be shown in the "players" component next to the players name
  + **design:** total freedom
  + **constraints:** total freedom (follow application structure)
* Update project to Python 3.x
  + **description:** currently the project runs Pyhton 2.7.x update the project and make sure all third party libraries works. Also verify deploying to Heroku is not borken.
* Limit the number of players per game to be less than the number of steps in the sequence
  + **description:** If there are more players than steps in the sequence the last players to join will win without playing. don't allow a player to join a "full" game
  + **constraints:** total freedom (follow application structure)
* Implement delete button for game in the /games page
  + **description:** users should be able to delete old games
  + **design:** show the delete button only for games the user can delete
  + **constraints:** only the game creator should be able to delete a game (follow application structure)
* Fix responsiveness on mobile
  + **description:** The simon itself is quite responsive but we need to fix side menu somehow
  + **design:** total freedom
  + **constraints:** use media queries
* Serve mp3s from a CDN
  + **description:** Currently the mp3 serving is not reliable, find a (free) CDN upload the files and serve from there
* Add comments to the entire project (ALL OVER!)
* Find and fix any bug
* Add some missing features of your own

**Enjoy!**