

Week 6 Progress Report (3/3)

- + (y/n) The product is effective in delivering the knowledge in computer science to the product users.
- + The product is user friendly/appealing in terms of the following criteria:
 - (y/n) The landing page is attractive.
 - (y/n) Users are able to understand and play the puzzle game quickly.
 - (y/n) Users can just jump in and start playing (trying out) the game immediately without the registration process.
- + The product should have the following functions:
 - (y/n) Users can register with a username and a password.
 - (y/n) The performance of registered users are updated after each trial and can be displayed upon requests.
 - (y/n) Users can ask for hints and/or solutions.
 - (y/n) Administration account:
 - Have all the functionality like the regular registered users.
 - Have additional privileges like user account removals or password-reset.
- + (y/n) Do you have a brute-force method as the comparison basis for the puzzle solver?
- + (y/n) Do you have a better algorithm than brute-force?
 - Since there will be different puzzles with different numbers of nodes connected in different ways, it is hard to determine another more efficient algorithm that would be able to find all the solutions for all the puzzles.
- + Explain if the puzzle is targeted at a single user or multiplayer, competitive or non-competitive. If it is a multiplayer-game, address the possibility of direct peer-to-peer communications without going through the host.
 - The puzzle is targeted at a single user. There will be an element of competitiveness in the sense that there will be a user leaderboard so users can see where they rank among the rest of the site's users, but users will not be completely directly against each other.
 - The game is not a multiplayer game.
- + Explain how to deploy your product.
 - I will deploy the Flask project with Heroku. After creating a Heroku account and installing the Heroku CLI, I will create a file named Procfile in the project's root directory and update the requirements.txt file and commit the files to Git. Then, I will use the Heroku CLI to create a Heroku app and then push the Git repository to the remote branch created with the Heroku create command. Afterwards, the building and deployment process will occur; the app will be online at that point and will be able to be opened on the web browser.