Rohith, Stephanie, Sydney
CS 222 Group 13

Interactive Web-based Escape Room Game

Pitch

Our website will offer an interactive experience simulating an escape room game. Players can enjoy the mysteries and challenges of an actual escape room from within the comfort of their own home by typing into a webpage to find clues, solve puzzles, and change environments.

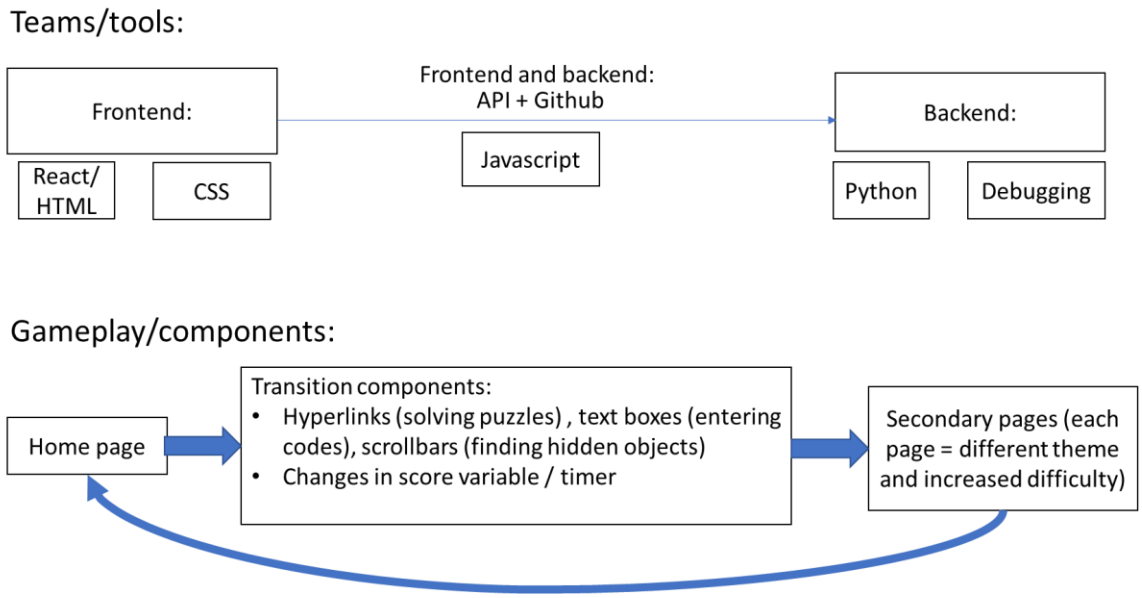
Functionality

1. Users can explore a webpage where performing certain actions (e.g. clicking buttons, scrolling, and typing on the screen) leads to on-screen feedback
2. Users can navigate to different web pages by clicking on on-screen objects.
3. Users can solve puzzles by opening pop-ups, typing codes, and interacting with hyperlinks, trying to earn as many points as possible before time is up.
4. Users can keep track of their scores and times throughout multiple web pages, and save statistics to a profile
5. Users can access and interact with an inventory of items they have collected throughout the game.
6. Users can advance levels that increase in difficulty, each of which offers different environments or challenge topics.

Components

- Backend: We will use Python for the backend, since most of our team members have experience working with Python, and Python will give us access to various libraries and frameworks to help us write our website. We're considering using Django as our framework because it will give us a large toolset for creating our website.
- Frontend: We plan to use CSS for the frontend. We will implement internal CSS style with HTML code in our basic web pages, and will likely use the React framework to supplement Javascript addons (we could also use Vue if it fits constraints better).
- We will create an API (using the Django REST framework) so that the frontend and backend can communicate.

(components diagram on next page)



Weekly Planning

1. Familiarize ourselves with which programming languages to use and who is most comfortable in which languages needed for the website. List down some interactive features we could implement and the purpose of each one (how it affects the game). Also explore suggested tools (TailwindCSS, DaisyUI, ReactRouter, etc.)
2. Create an outline describing each webpage, how they are connected, and each interactive feature (and its impact on the website)
3. Code (backend) the main home page (basic web page containing main game screen before any interactive features are implemented)
4. Code (backend) interactive features that will be used in our game, such as a text box or scrollbar, which impacts the result of the game
5. Improve backend code: debug errors and edit code to be more efficient (consider runtime, memory, etc.)
6. Assign 2 members to begin frontend development (finalizing the main home page, including visual styles like CSS). Other members should continue backend development of interactive features, as well as writing code (backend) embedding each feature onto the main web page
7. Frontend developers finalize each interactive feature and test them to ensure they serve the purpose for our game/website. Backend developers begin to add game score/time trackers, as well as a log-in/profile page to save results in a database.
8. Frontend should finalize the tracker and profile pages. Improve visual quality/readability of the website (CSS), and add small visual effects that show during the game.

Potential Risks

1. Some elements we incorporate (including visual effects, counters that track progress, etc.) will take up lots of memory that needs to be saved even when a user clicks on a different web page. Debugging these features may be time consuming as the website/platform can easily crash when too much storage/memory is being used. We may need to implement/utilize web tools (like cookies) to reduce memory challenges.
2. Later in the process, because frontend and backend developers are working on different components to the website, backend developers may notice a bug or necessary change to an older page/component; if that component is already being finalized by frontend, it will be difficult to communicate information about the errors between the two teams. This makes it important to scan for errors often and debug them as soon as possible (preferably in the backend stage).
3. When we publish this website online or integrate it into an app, there may be certain restrictions regarding the file formats accepted or the amount of storage available. Therefore, we need to decide on a publishing platform in advance and conform to its requirements throughout frontend development. This also means that some interactive features may not work with all publishing platforms and devices that the website will run on.

Teamwork

We will use a GitHub repository to collect all of our code in one place. We plan on communicating with each other primarily on a Discord group chat we set up, and we will also meet up regularly to check in on each other's progress. All three of us will start out working on the backend and will eventually split up into two teams, one of which will continue working on the backend while the other develops the frontend. To prevent conflict, we will communicate clearly with each other and divide up the workload fairly.