Budget Hero Documentation

Technologies Used

Backend Libraries

- NodeJS
- MongoDB
- Express

Front End Libraries

- HTML5
- JavaScript5
    - Canvas
    - jQuery
    - Chart.js
    - Survey.js
- CSS3
    - Bootstrap

Level Creation Tools

- GIMP
- Tiled
-

Source Code Instructions

How to Run Code

1. Download and extract BudgetHero.zip file
2. Ensure node is installed
3. Run *npm install* to gather the required dependencies from package.json
4. Run *index.js* file to start the server
5. Access the URL the server is hosting to start the game/survey

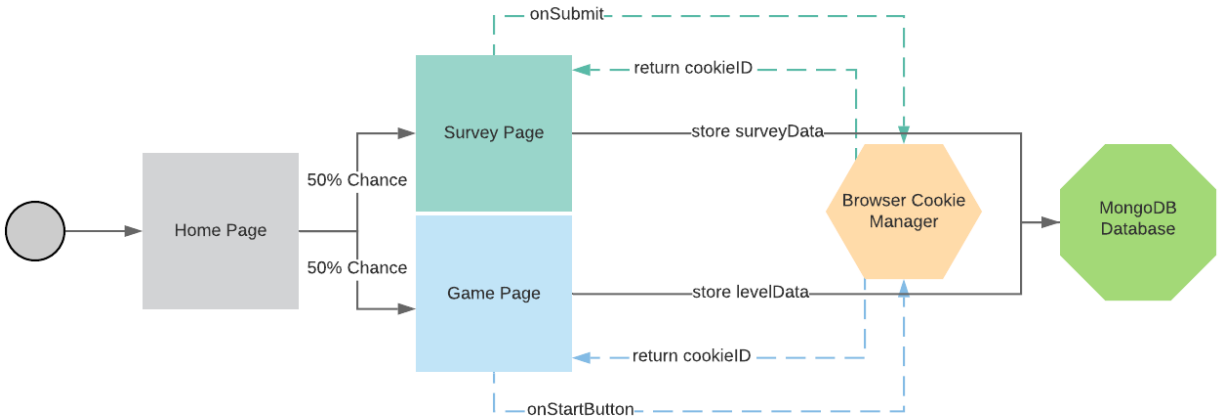
* If you want data to be stored properly, ensure MongoDB connection key is correct in *index.js*

How to Extract Data

1. Extract .json file from MongoDB '*users'* collection
2. Run the dataloader.py file from the zip extract
3. Results in a .csv file with data displayed cleaner

Generic User Flow

Flow Between Data and Events



Implementation Process

Generic Timeline and Details of Implementation

1. Generic template for game adapted from a [Rabbit-Trap implementation]
    a. Template followed the format of having individual files to handle its own tasks across the entire game
        i. *controller.js* to handle user inputs
        ii. *display.js* for rendering everything onto the HTML Canvas
        iii. *game.js* for all game logic throughout the game
        iv. *enginer.js* to control the fps and runs of the game
        v. *main.js* to communicate logic between all files
2. The template served as a boilerplate platformer for the Budget Hero game to be customized onto it
3. The following are the major changes needed to make the generic platformer into Budget Hero
    a. Create new sprites for the avatar
    b. Create new levels using custom tiles and collision maps
    c. Create coin sprite and coin functionality
    d. Create a specific bin level where the user can deposit coins
    e. Create controls to deposit coins
    f. Create Attack Functionality
    g. Create Enemies
    h. Create the survey interface
    i. Add a Homepage with a button to randomly choose to go to the survey or game
    j. Create backend service to store data to database
    k. Create cookie functionality to store user trials