Coursework 1: Game website

Ignas Lamanauskas M00655420

Project report

**Game:** “Flappy bird”

**Description:** Game objective is to get the highest score by navigating the object (Bird) through the small space in between other objects (Pipes).

**Scoring system**: The player scores one point for each pair of pipes he passed.

**Controlling**: You can control the bird by just simply pressing space button on keyboard.

**Game over scenario**: Whenever the bird is going to hit bottom or upper wall or a pipe in any direction - the game will be over.

**Brief project description:**

To create this project I used HTML, JSON local storage, CSS, PHP, JavaScript. The structure of the website is mainly created with HTML but for styling I used CSS and for website functionality I used PHP. It let me to create the functions with whom I did not needed to duplicate my code in every page and made my life easier in other cases.

The game, rankings, registration, validation, login and log out – all of those I made using JavaScript and I stored the users in JSON local storage.

**Validation:** I have a function which gets all registration fields from HTML and puts them through “Validate” function to check whether every field is done correctly. For some of the validation requirements I used “Reg Expressions” which helped me to create better validation.

**Registration:** For registration to be successful, user must complete all the required fields correctly and input their username, password, email, phone number. If user passes all the validation requirements inside the function, I am creating an object in which I am storing the user information and passing it to local storage. Once registration is completed user gets and message and gets redirected to Log In page.

**Login:** Firstly, function checks if entered email exists in local storage and then by the email user has given function checks if password is correct. If Everything is correct user is getting a message of successfully login and get redirected to game page. If not, errors with what is wrong is being displayed.

**Error handling:** If any of the entered information is not correct, user is getting an error with requirements of what he is missing or which part he needs to correct.

**Log out:** If user is logged in, text of his email and “Log Out” button appears in navigation bar right side. If “Log Out” button is pressed user is removed from “sessionStorage” where logged in users are stored and text of his email and button disappears.

**Game:** “Flappy bird” is created using Canvas. To fully create the game I used images, functions, objects, arrays, events listeners, switch cases, loops. The game itself has a “collision” logic which is probably the most important part of the complete game. To start the game, you need to click on the canvas with mouse and then start “flapping” with space button on your keyboard. Whenever user losses, score is displayed, and user is been asked if he wants to play again and what instructions he needs to follow to do that. Top score of each user is stored in local storage. **Please read “Readme.txt”** file before playing the game.

**Ranking:** Getting score from the game and putting it inside every user in local storage. Function checks if email matches with logged in user and user inside object and if just obtained score is higher than the current user top score, then its automatically changed to a new one. After it is passed to an user inside an object it displays in rankings table with username and its top score achieved.

**Difficulties I faced during this project:**

As I have never touched any CSS, PHP, JSON or JavaScript before everything was very new for me. Therefore, I started to do this project very early, step by step by watching tutorials on YouTube, reading, getting a logic behind everything. **The most difficult things to do for me was:**

1. Creating game Pipes in array and make them go towards bird non-stop until the game is over. I did not used a lot of arrays, so I was not very familiar with them. Had to learn how array works and how I can use it.
2. Implementing game states. Even though it seems simple thing to do, it was kind of difficult for me, because after creating it, something was not working, or I was getting errors all the time.
3. Rankings – this part was the most difficult one as I was struggling to get the score displayed inside object of user. After that I was trying to display all the users top score’s in rankings table but couldn’t manage to do that. My problem was that I was only getting one user inside an array and not all of them, or if I got all of them in my array, I could not get username or top score from it. So, my rankings table is only displaying the top score of current user online even if other users has their own top score’s stored in local storage as well.