



Project Description

INTRODUCTION

This project is oriented to deliver the recipient a simulation regarding a complete gaming experience intended for amusement purposes, precisely developed to entertain users with a modernized version of a classic game experience focused on "Snake", a game made renowned also by mobile devices around the year 2000.

The website has been built based on a classic development modality, related to the pure usage of only basic/essential tools like HTML5, CSS3, JavaScript, PHP, and some further libraries as Phaser 2, etc, for educational purposes.

Pages structure has been shaped as per usual by enclosing logo, menu bar, footer, pictures, texts and links.

DESCRIPTION

- **Game description:**

1. (**Logic**) - As previously mentioned, the game is a JavaScript replication of **Snake** (a famous classic). It is built and implemented into the website home page.
The game consists of a snake who runs/slithers around, able to be controlled by 4 keys (directional left and right arrow, and up and down arrow to change speed) of the keyboard or by the mouse as well, in order to make Snake change direction, with the purpose of getting longer and wider by colliding/eating objects randomly appearing around it. The progressive extension of the snake due to the repeated feeding represents the score increase, as well as the difficulty because the longer the snake gets the less available space remains to slither around since as soon as the snake head touches the rest of its body the game goes into a game over state. Alternatively, if the user manages to keep the snake alive until the filling of the game environment then a new level starts with a faster running snake, hence increasing the game difficulty.
2. (**Source code**) - This Game is powered by the powerful **Phaser 2.6.2** game engine. Initially I started using *Phaser 3*, but since the most of material and tutorials are about Phaser 2, I eventually had to switch using the previous version of Phaser and indeed it ended up being the best option due to a much quicker retrieving of examples, advises, and tutorials by a broader community. Several objects and prototypes have been utilized rather than classes since these latter are an EcmaScript2015 (*ES6) and

above feature, and sadly any Phaser versions does support yet up to ES5 and older versions only of JS. The source code is organized in components files containing objects and global functions and three main files which are respectively the first state (start), the main state (game) and the last state (game over). The “start” state is the first image displayed whit a “play” button when the home page loads, while the “game” state might be considered the main scene, the one where the snake slither through, and the “game over” is the last step, reached only after the snake death and in this latter the user can choose to play again or get suggested to save the score whenever he/she is not logged-in yet. As a last reflexion I could say that Phaser 2 is a nice and powerful tool which helped me a lot in building a challenging game in half the time it would have taken by using just the canvas, but most surely if I ever build I game again I would definitely go for Phaser 3 that time, hopping I will have much more time to get properly familiar with this new technology, since during my recent researches I find out it is for a plethora of reasons much better, performing, and amusing to work with, despite its earlier community and subsequential difficulties to commence using it.

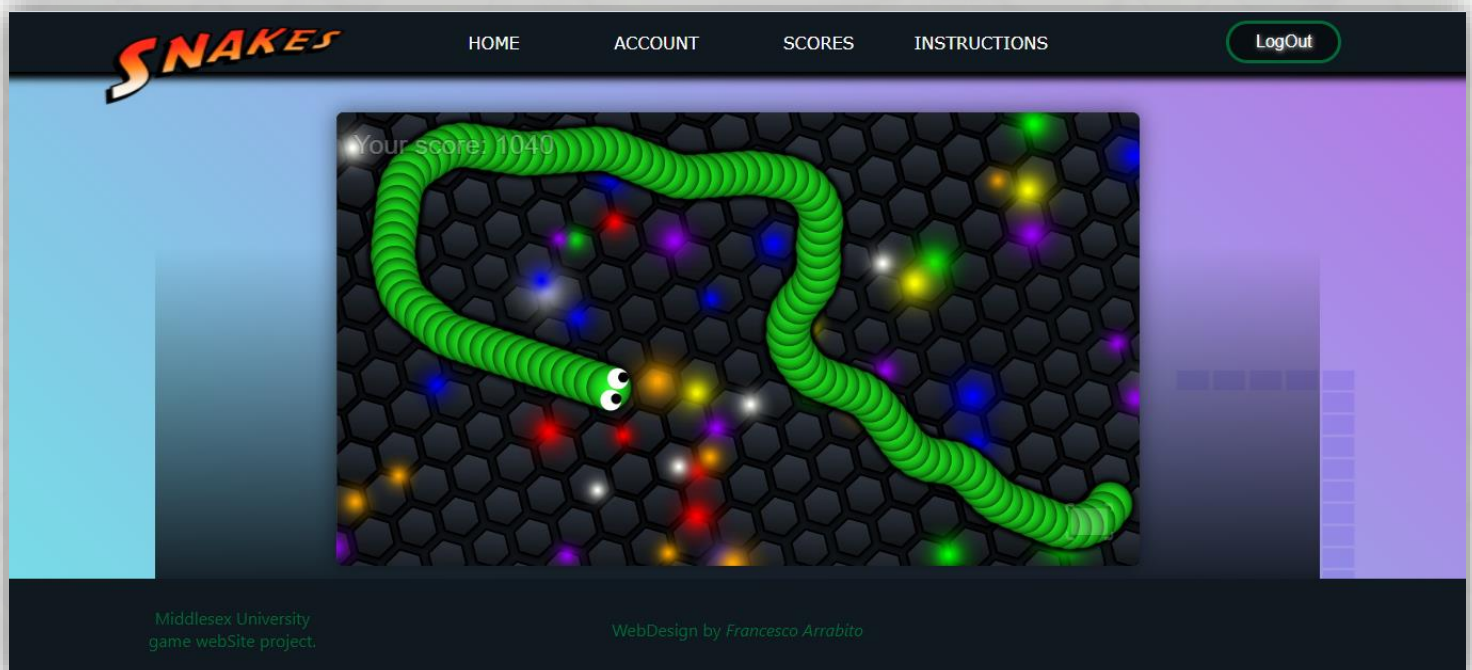
- **Website scripting:**

1. (**Home**) – The “home”, as scripting, and apart from the game canvas script, covers the login button only, since the login button contains an event listener which detect if a user is logged, and if so, it changes its text from log-in to log-out and vice versa, and once pressed, it log-out the user.
2. (**Account**) – *This page JS code is focused on the validation of data entered from users into the sign-in or sign-up form. It detects if de data entered are correct before submitting and accessing to the user page. It checks every input box and other than verifying that data are of an allowed format it minds whether the email and nickname have been registered already from someone else and therefore denies the double registration of such data by displaying red errors messages for each interested input box of the DOM.*
 - a) Name, Surname, Country: these fields will accept letters only.
 - b) Date: will automatically adjust the format.
 - c) Email: check if the format includes text before the “@”, after and some text after a dot, lastly it checks if the email is registered into the JSON format database (localStorage) by another user.
 - d) Nickname: accepts letters, number, no initial spaces and a dot.
Furthermore, it checks if the Nickname is registered already into the JSON format database (localStorage) by another user.
 - e) Password: has to be 8 characters minimum.
 - f) Repeat-Password: has to match the “Password” field.
3. (**User Page**) – The user page has two functionality (scripting) “borrowed” from other pages like account page and ranking table page. Indeed, it is capable to show the user his/her personal top scores, date and time into a dedicated personal top scores table, and moreover, it provides the facility to check personal data and edit them whenever needed. The table score script and the data editor form script are essentially the same script used for the public ranking table and SignIn/SignUp forms of the account page.

4. (*Ranking table*) – The public ranking table is handled by a script which check the database (localStorage) and get the score data of every user registered, in order to sort it and insert it into the table by cycling every table cell and user score data.
5. (*Instructions*) – This page, as scripting, covers the login button only, since the login button contains an event listener which detect if a user is logged, and if so, it changes its text from log-in to log-out and vice versa, and once pressed, it log-out the user.

GRAPHICAL REPRESENTATION

1. Home page



2. Account page

The screenshot shows the 'ACCOUNT' page of the SNAKES game. The header includes the SNAKES logo, navigation links (HOME, ACCOUNT, SCORES, INSTRUCTIONS), and a 'Login' button. The main content area has two forms: 'SignIn' and 'SignUp'.

SignIn Form:

- Fields: Your eMail, Your Password
- Button: SignIn

SignUp Form:

- Fields: Name (with format gg/mm/aaaa and a calendar icon), Surname, Country, eMail, Nikname, Password, Repeat-Pass
- Button: SignUp

Footer:

- Left: Middlesex University game webSite project.
- Right: WebDesign by Francesco Arrabito

3. User Account page

The screenshot shows the 'ACCOUNT' page for a user named Francesco. The header includes the SNAKES logo, navigation links (HOME, ACCOUNT, SCORES, INSTRUCTIONS), and a 'LogOut' button.

Header: Francesco's best score: 7910

User Profile Details:

- Name:** Francesco
- Surname:** Arrabito
- Country:** Italy
- eMail:** francuccio93@gmail.com
- Nikname:** Francy.93
- Repeat-Pass:** (masked with dots)

Update Button: Update

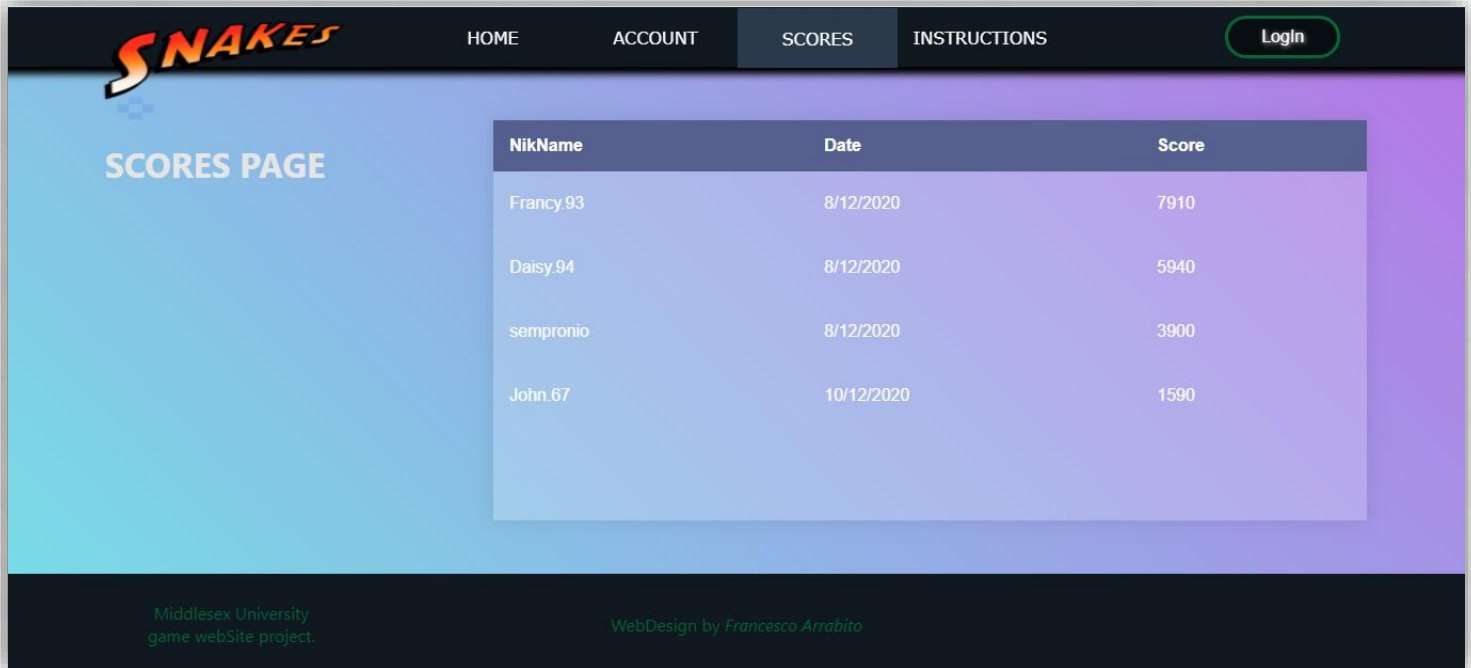
Score Table:

Date	Time	Score
8/12/2020	04:22:59	7910
9/12/2020	03:04:10	4950
8/12/2020	22:22:07	3140
10/12/2020	01:46:07	3130
8/12/2020	22:21:27	3000
9/12/2020	15:23:26	2220

Footer:

- Left: Middlesex University game webSite project.
- Right: WebDesign by Francesco Arrabito

4. Scores page



5. Instructions page

