JavaScript Project

In this practice project you will design and develop a web based product using HTML, CSS and JavaScript. As a reminder: HTML defines the content of web pages, CSS defines the appearance and JavaScript provides the functionality. (Refer to chapters 24 & 25 in your text book)

You should use Notepad to write the HTML, any CSS file(s) and the JavaScript files. The file extensions for these, respectively are: .html, .css and .js. Hint: Make use of the <div> tag to help with layout.

Some things in the project you’ve not been taught e.g. writing to a text file in JavaScript. Use the Internet to find out how!

# The Product

The product will be a game in which the user has to attack an enemy e.g. a bug which appears at random locations on a map, repeatedly. It can be attacked by clicking the mouse on it, but it mustn’t be too easy to do as the enemy should not stay still for long.

If the strike hits the enemy, the user gains points e.g. 10. If an attack is made but the strike misses, then the user will lose a life. The user starts the game with 10 lives and if they lose all of their lives the game is over. The user wins the game if they don’t lose all of their lives and their score is e.g. 100 points.

Note: you can opt for an enemy of your choosing and the map can also be of your choosing e.g. Europe. You could make the game more complex by varying the number of points based on where on the map the enemy is struck. You could also deduct points if the enemy is struck if safe areas e.g. over the sea rather than in a country.

The game should be suitable for teenagers aged 13-18. The style of the webpages should appeal to this age range and there should be a consistency of style. The design should make good use of page space. Note, white web pages with standard size grey buttons wouldn’t be appealing. Spoil me with your creativity!

In order to play the game a user must be a registered user and they need to login.

Requirements:

1. The name of the game (e.g. Bug Invaders) must be displayed on all web pages.
2. The game must have a home page with 4 buttons which will let the user do the following:
   1. Register
   2. Login
   3. View instructions
   4. Play Game
3. Register –
   1. Clicking on the ‘Register’ button should cause a new page to be opened with a form where the user should enter their details. The data to be gathered is the user’s name, a valid email address and a password. You can opt to do validate here if you want in order to make your system more robust.
   2. The form such have a submit button and a reset button. The latter button should clear the form and the submit button should cause the user’s details to be saved in a registration file (i.e. text file) in the format: email address, password, name .

For example,

[boydn001@chartersschool.org.uk](mailto:boydn001@chartersschool.org.uk) , KitCat1, Nick

* 1. The user should receive a message to tell them when they have successfully registered. For example, you could use the JS alert command for this.
  2. There should be a home or back button on the page.

1. Login –
   1. Clicking on the ‘Login’ button should cause a new page to be opened with a form where the user should enter their email address and password.
   2. The form such have a submit button and a reset button. The latter button should clear the form and the submit button should cause the product to read the registration file to find a match.
      1. If a match isn’t found, the user should receive a message telling them their email address or password wasn’t recognised. The message should remind them that they need to have registered before they can login.
      2. If a match is found, the user should receive a message to tell them they have logged in successfully.
   3. There should be a home or back button on the page.
2. View Instructions –
   1. Clicking on the ‘View Instructions’ button should cause a new page to open that provides instructions on how to play the game and its aim.
   2. There should be a home or back button on the page.
3. Play Game –
   1. Clicking on the ‘Play Game’ button will take the user to a new page where they can start to play the game.
   2. The page should have a HUD (Heads-up display) that shows the user how many lives they have remaining, their current score, the strike status when they attack the enemy (e.g. hit or miss), and how many points they’ve gained or lost when they make a strike. The HUD should display the number of lives remaining graphically e.g. 5 love hearts to show 10 lives remain. The graphic should change each time a life is lost.

I’d love to see some use of Object-Oriented programming in your project but not to worry if that’s just asking too much.