Omar Alkendi

M. Lehr, PhD

CIS/CSC-17B

13 December, 2019

Group Project: Lehr's Lair

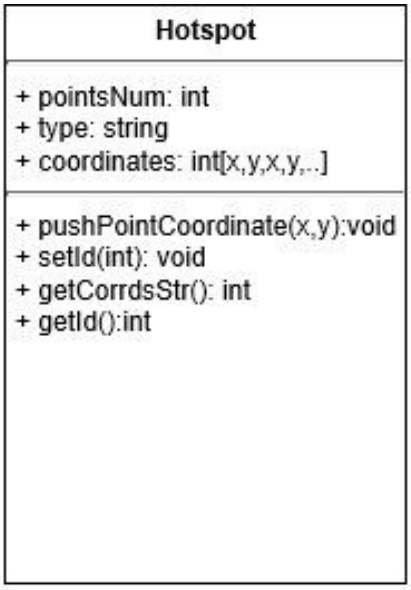
Introduction:

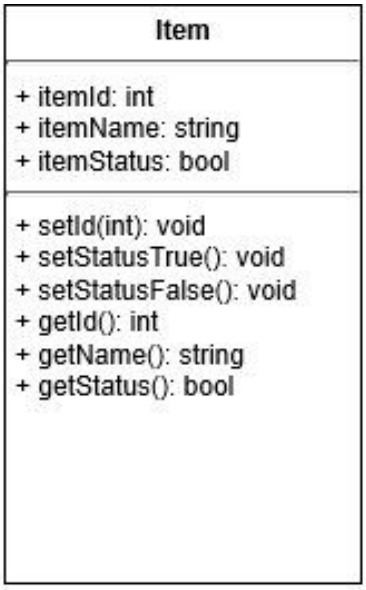
Lehr's Lair is a basic escape game in which one of Dr. Lehr's students attempts to steal the sample code for the final project from Dr. Lehr's personal computer. However, the professor finds out and decides to teach the student a lesson—a harsh lesson. The student gets knocked out and transported to an unknown location. Locked in the unknown location, the student tries to escape and save his life.

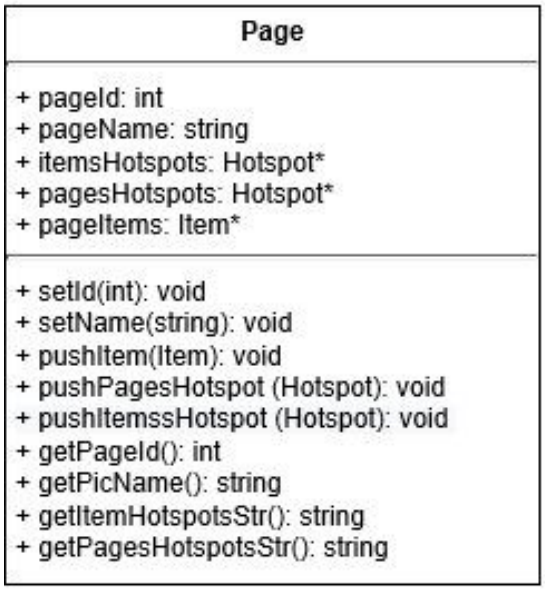
Concept:

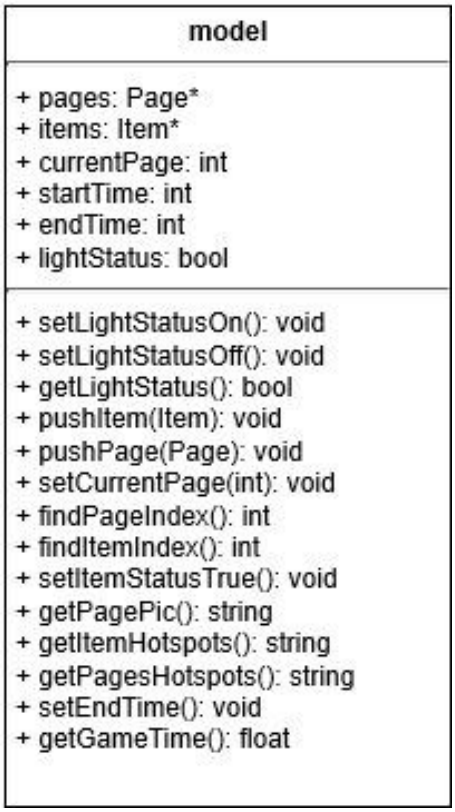
The game consists of a level that includes different location of the presented as Pages. Each Page has a set HotSpots that links it to other pages and allows the user to interact with the items on the page. Once an item has been clicked-on (obtained) it disappears from the user's field of view.

To achieve such a goal, multiple classes were implemented.









For Lehr's Lair code, click here.

MVC:

Lehr's Laor utilizes the MVC design by incorporating a controller (trunkController.js) that modifies the currentPage variable in the model (model.js) and then calls the update() function which refreshes the view for the user with the elements of the new page.

```
function update () {
    var image = document.getElementById("pageImage");
    image.src = level.getPagePic();
    document.getElementById("pageHotspots").innerHTML = level.getPagesHotspots();
    document.getElementById("pageItems").innerHTML = level.getItemsHotspots();
}
```

Objects:

There are a total of 4 JavaScript classes and one PHP class in the project.

JavasScript Classes:

1. Hotspot.js: Holds coordinates for the clickable spots and a unique id.
2. Item.js: Holds the information of an item in addition to a unique id shared between the item and its hotspot.
3. Page.js: Holds back-grouns image name of the pageplus a unique id shared with the hotspot leading to it.
4. model.js: Connects the Hotspot, Item, Page classes to structure the level of the game.

PHP Classes:

1. User.php: A class that assists the user in creating an account to save the user's score. It also allows the user to change their name, password, and email address. Moreover, it can check the login credentials to determine if a user with the entered credentials exists.

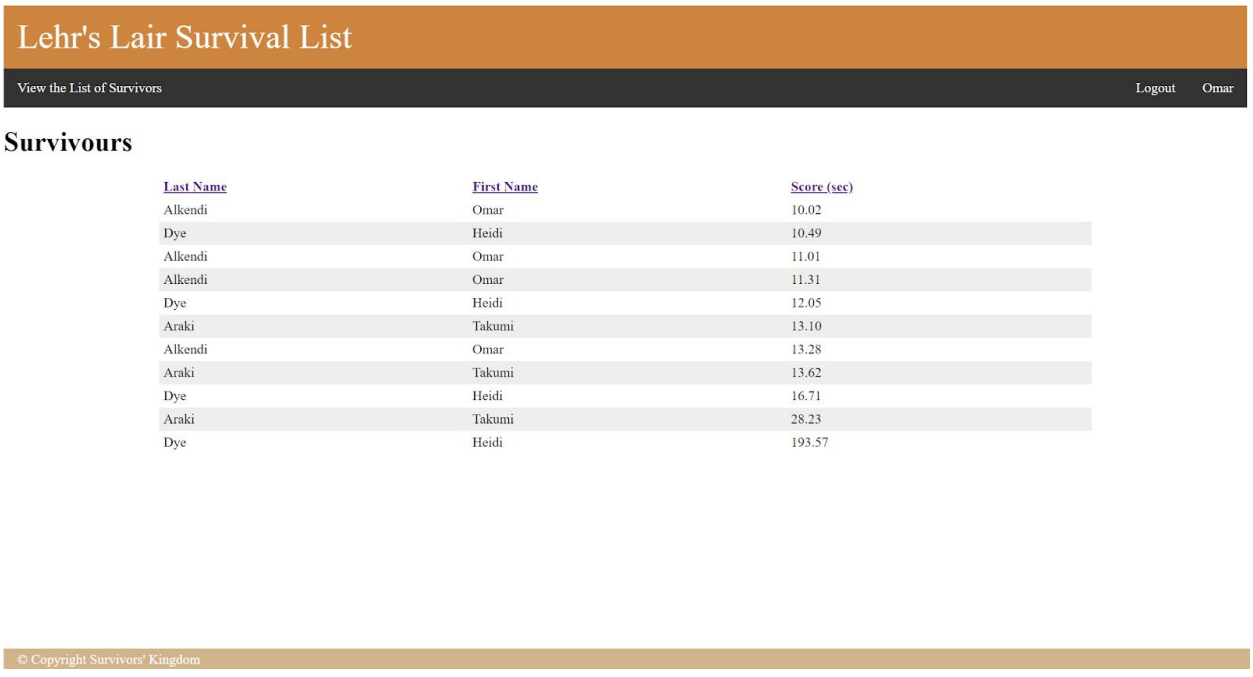
Reading/Writing Files, Local Storage, and JSON:

NA: No concept under this category was used in making Lehr's Lair.

Databases and MySQL:

A MySQL database was utilized to hold two tables:

1. entity_user: Holds users' information and assign it a unique auto-incremented id.
2. score: Holds a record of all users who finished the game in addition to the time it took the user to finish the game (Hall of Fame).



Form Validation:

Form validation was implemented in the sign_up.php page to prevent the user from creating an account without the proper format for:

1. First Name: Must be in English letters only.
2. Last Name: Must be in English letters only.
3. Password: Must be 6~20 characters length, include one upper-case, one-case, and one special character.
4. Email: Must follow the general format for an email (eg. a@b.com)

User-Admin Login:

The uni_login.php is a login page for both users and admins, and will redirect each to their proper page. However, no admin features were added so far.

Cookies, Sessions, and Securing Pages:

Cookies were used to transfer the the time it took the player to finish the game from JavaScript to PHP. Sessions on the other hand was used to determine that the user won't be logged-out unless the browser is closed or the logout option is clicked.

```
function endGame() {
    level.setEndTime();
    var time = level.getGameTime();
    setCookie("time", time, new Date().getTime()+ 3600);
    //checkCookie("time");
    alert("Congratulations! You were able to escape the lair in "+time+ "seconds!");
    window.open("update_list.php");
}
```