



Al Imam Muhammad ibn Saud Islamic University
College of Computer and Information System
Computer Science Department
2nd Semester 1443 H – 2022 G



CS 438 – Internet Technologies

Project

[The Jumping Drake]

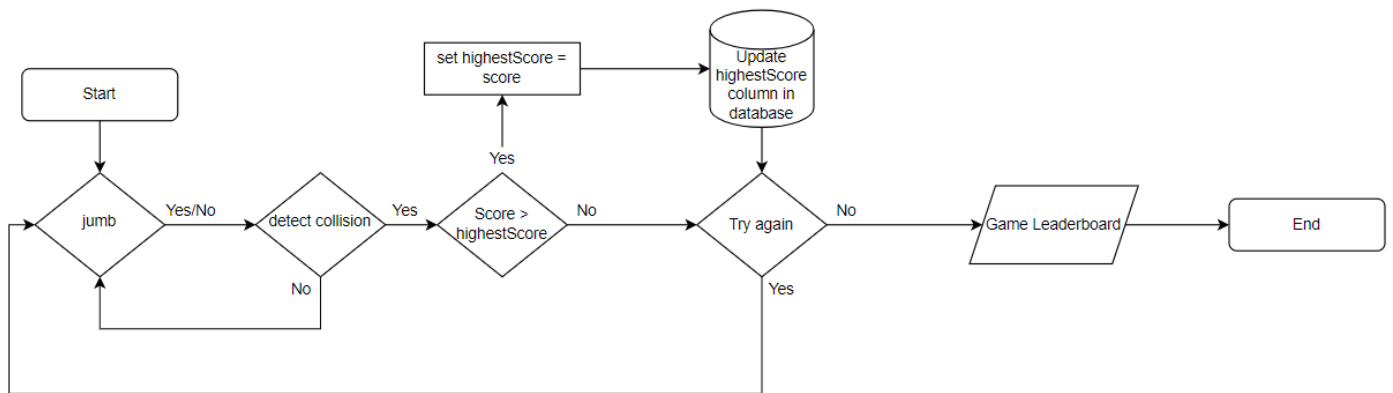
BY

STUDENT NAME	ID
Khaled Mohammed Alzouman	439012271
Abdulilah Saleh Alqasem	439014750
Mohammad Abdulaziz Zouman	439012798

[May 11, 2022]

The Jumping Drake website is a game in which participants compete to earn the greatest score. The game is intended for a wide range of audiences. The player takes on the role of the drake, avoiding incoming knights by hopping over them; if a knight contacts the drake, the game is over.

1. Flow Chart



2. Look & Feel

The Jumping Drake game design was created by using CSS style sheet (**DragonStyle.css**). The Drake character made by a dragon GIF [1]. The background is made by a lava field GIF [2]. The lava floor was made by lavaland GIF [3]. The knights were made by a knight GIF [4]. The design was created in such a manner that the drake would jump over the knights who were on their way to hunt him. The background and the floor are moving with the drake to make it feel real and fun.

3. Dynamic Components

We have only one script which is **DragonScript.js** file and is linked in the **game.php** file using this line: `<script src="DragonScript.js"></script>`

4. Business Logic

The database was created using XAMPP, and we implemented it in the same manner as we learned in class. The database is called "the jumping drake," and it has one table called "users" and five columns: "id", "username", "email", "password", and "HighestScore".

In the **DragonScript.js** file we used XML to send the value of score to **game.php**, to do that we wrote the following script:

In **DragonScript.js**:

```
var xhttp = new XMLHttpRequest();
xhttp.open("POST", "game.php", true);
var ourFormData = new FormData();
ourFormData.append("score", score);
xhttp.send(ourFormData);
```

and to receive the value in **game.php**:

```
$score = $_POST["score"];
```

MySQL queries are described below for each file that has queries.

server.php:

- 1) The query "SELECT * FROM users WHERE username='\$username' OR email='\$email' LIMIT 1"; was used to check whether the user is already a registered member or not.
- 2) The query "INSERT INTO users (username, email, password) VALUES ('\$username', '\$email', '\$password')"; was used to add a new user in the database.
- 3) The query "SELECT * FROM users WHERE username='\$username'; AND password='\$password'"; was used to login an already registered user.

game.php:

- 1) The query "SELECT * FROM users WHERE username = '\$username'"; was used to get access to highestScore column of the player to compare its value with his newest score value.
- 2) The query "UPDATE users SET HighestScore = '\$score' WHERE username = '\$username'"; was used to update the value of HighestScore column.

leaderboard.php:

- 1) The query "SELECT username, HighestScore FROM users ORDER BY HighestScore DESC"); was used to get all users HighestScores in a descending order.

References

- [1] <https://acegif.com/dragons-gifs/>
- [2] [https://www.pinterest.nz/pin/385550418076972645/?amp_client_id=CLIENT_ID\(\)&mweb_unauth_id=%7B%7Bdefault.session%7D%7D&simplified=true](https://www.pinterest.nz/pin/385550418076972645/?amp_client_id=CLIENT_ID()&mweb_unauth_id=%7B%7Bdefault.session%7D%7D&simplified=true)
- [3] <https://www.vectorstock.com/royalty-free-vector/texture-soil-burnt-earth-with-geysers-lava-vector-36918393>
- [4] <https://imgur.com/r/animation/KvEH5Ey>
- [5] <https://app.diagrams.net/>
- [6] <https://www.w3schools.com/>
- [7] <https://stackoverflow.com/>
- [8] https://www.youtube.com/watch?v=AiE6CDACWT4&ab_channel=CodeBoxx