Programmer's Guide

Project Name: Fraction Runner
Team Name: Team DBA
Team Members:

- Gregory Shelton gpsc7c@umsl.edu
- Maija Garson mmgzzn@umsl.edu
- Kayla Thurman kethurman@mail.umsl.edu
- Sedaf Shakeel ssmkh@missouri.edu
- James Platt jbpkcd@mail.umsl.edu

Revision History:

04/04/2023 – 1st Draft
04/11/2023 – Revision, publish to GitHub
04/12/2023 – Revision per team discussion
04/14/2023 – Revision per team discussion
05/02/2023 – Screenshots and polishing
5/09/2023 – Added screenshots
5/09/2023 — Final updates to Implementation Code

Section 1: What a Programmer should know about Fraction Runner

Programming the Fraction Runner game: The game is built using HTML, CSS, and JavaScript. The uses a simple game loop to update the game state and render the graphics.

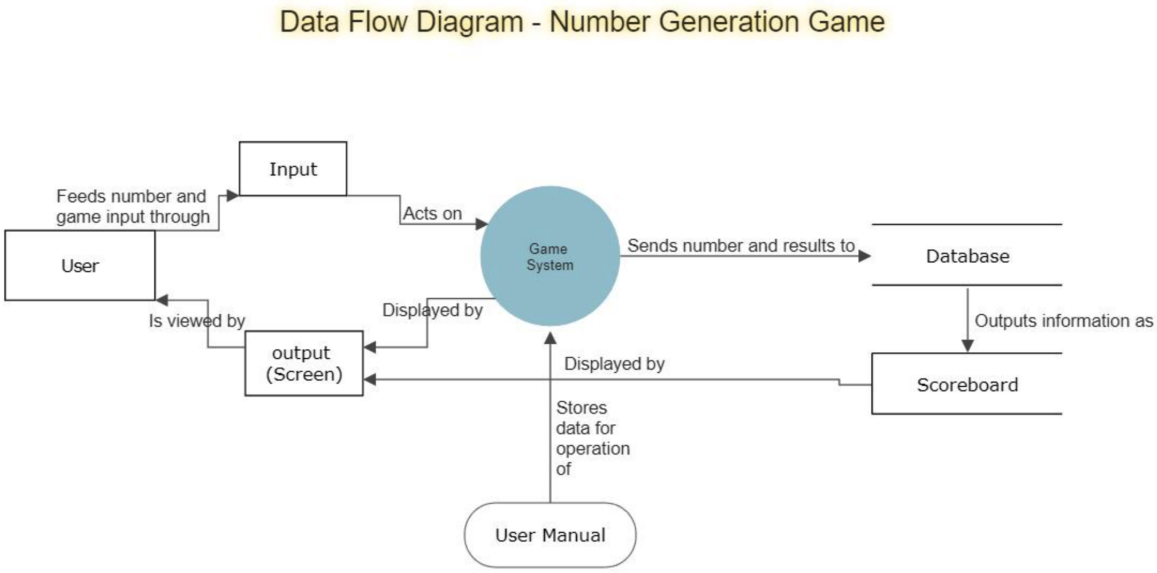
To program the Fraction Runner game, you will need a basic understanding of HTML, CSS, and JavaScript. It is recommended to use an integrated development environment (IDE) to write and test your code.

Here are the basic steps to programming the Fraction Runner game:

1. Pull files from GitHub.
2. Get a domain name and a web host.
3. Recreate the file structure inside code bases / webpages (directory / folders)
4. After testing on local machine, export database using MySQL server 8.0.
5. Change password, hostname and database name listed in the scoreDatabaseFunctions.php file under function()makeConnection.
6. If webhost errors for requirements, all file locations referenced in code must be changed to fix this.
7. Create 2 Databases on web server or local MySQL by use of the included by theMySQL database set up code file in codebase / database. This will create the table that will store the high score, fractions and user information database between game sessions. We choose this method because locally hosting the database off-server is not a valid option in this case, and the numbers and user data need to be stored on separate tables to prevent column bloat. We are using MySQL within php to access the server.
   [IMPORTANT]
   Firing query "Database Setup Code" in dba/codebase/database using MySQL Workbench, phpMyAdmin, or similar is REQUIRED to set up a local database.

For purposes of our current endeavor, everything here should already be handled, but note usage in case of errors mysql database has two important user types, "root" (administration, passcode is set as "VfX!565WW!t552") intended to be set with all permissions, and "siteuser" (average access to database, passcode set as "edcvfr43edcvfr4") intended to be set with permissions to DELETE, INSERT. SELECT, and UPDATE records.
Finally, intended servername should be at "127.0.0.1", for testing, Machine Local Network IP (usually 198.68.0.*) Or webserver IP ,subject to change based on webserver settings.

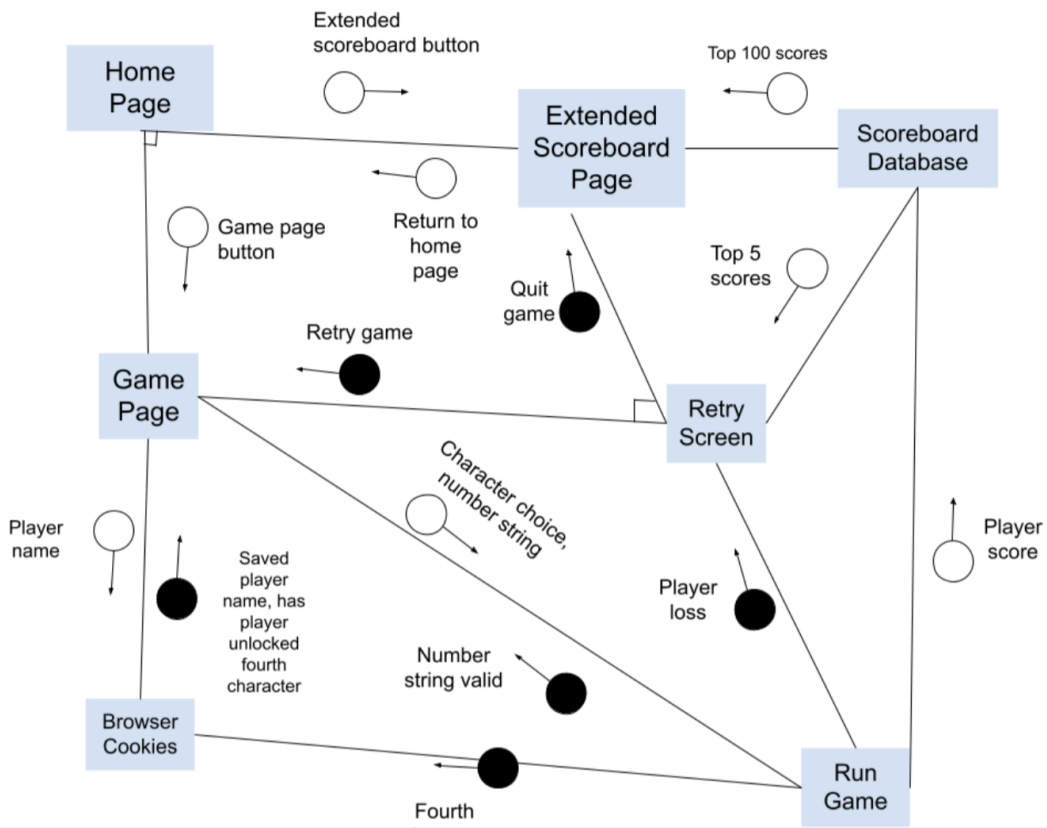
Section 2: High Level Design



Section 3: More Detailed Designs

Team DBA Detailed Designs

Structure Chart



Pseudocode Detailed Design for Fraction Runner

Front page:

CSS: all centered

<header (Welcoming)>

<close header>

<introduction to concept and game>

<link to video>

<close link>
<close intro>
<images>
<static images>
<close static images>
<gif of game>
<close gif of game>
<close images>
<container>
<button>
<link to game>
<close link>
<close button>
<button>
<link to extended scoreboard>
<close link>
<close button>
<close container>
<footer>
<close footer>
Game page:
CSS: all centered
<header (Welcoming)>
<close header>
<game border>
<game window>
<close window>
<close border>
<container>
<button>
<link to intro>
<close link>
<close button>
<button>
<link to extended scoreboard>
<close link>
<close button>
<close container>
<footer>
<close footer>
Extended Scoreboard Page:

CSS: all centered
<header (Welcoming)>
<close header>
<score border>
<scoreboard table (100 rows linked to database)>
<close table>
<close border>
<container>
<button>
<link to intro>
<close link>
<close button>
<button>
<link to game>
<close link>
<close button>
<close container>
<footer>
<close footer>
JS:
Javascript psuedocode:
Function to layout data(information: table name, rank, name, points)
Variables:
Table name
Table row
Table Divider (Rank)
Table Divider (Name)
Table Divider (points)
Table Divider (Repeating String Generated)
Set dividers = children of row
Set row = child of table
Function to repeat above function 100 times onload (tablename, name, points)
if(i = 0, i< 100, i++){
Function to layout data(tablename, rank(i), name at rank I, points at name at rank i
}
Game:
Game start
Print brief explanation of math that was referenced on the intro page
Event listener for pressing of mute button
IF audio is NOT muted
mutes audio

ELSE
unmutes audio
Event listener for pressing of quit button
Pauses gameplay
Asks confirmation that player wants to return to home page
IF player clicks YES to return to home page
returns user to intro page
IF player clicks NO to continue game
resume gameplay
Cookie check
IF no cookie
input window for name to put on scoreboard
IF cookie present
checks for unlocked fourth character
Number input window, onhover:
add color around box
onclick: change color around box
Event listener for number input window
IF number input is out of range (<1 or >9)
display some appropriate error messaging
allow user to redo input
IF number is valid
Generate repeating decimal
Count number of significant digits in number
Divide number by equal number of 9s (ex. 221332/999999)
Show brief fake load screen showing the number get generated in a flashy way.
proceed with related game logic
Number input window, onhover:
add color around box
onclick: change color around box
Event listener for character select to choose avatar
Onclick: sets variable that determines character
Sets strings so that character sprites display correct character
Event listener for GAME START button
Onclick:
start game
begin generating ground made of numbers as well as obstacles
Event listener for jump:
IF character is not jumping:
trigger jump
set character state to jumping

when jump action is complete
reset character state
ELSE
do nothing
Event listener for duck:
IF character is not ducking:
trigger duck
set character state to ducking
when player releases duck button
reset character state
Event listener for attack:
IF character is not dodging
trigger attack
set character state to attacking
when attack action is complete
reset character state
ELSE
do nothing
event listener for minute passed:
Check transition number
If transition number > 5
transition back to first background
set transition number back to 1
Trigger transition linked to number
reset minute time for next transition
event listener for obstacle collision with character:
Trigger game over
stop movement on page on collision
set player state to loss
Record score
Checks cookies for high score on browser
display top 5 scoreboard with retry/quit buttons
Event listener for retry button, onclick:
Reset score to zero
Reset other relevant variables (character state, transition number, timer/elapsed time, repeating decimal, etc) back to initial values
Reopens input window for repeating decimal and character select
Event listener for quit button, onclick:
Take user to extended scoreboard page
Event listener for home button on extended scoreboard page
Return user to homepage

scoreboard databases
XML file including rules of database

```
CREATE TABLE users (
#Variable/column name/ids and rules
#NOT NULL
'user_id' int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
'user_rank' int(10) UNSIGNED NOT NULL,
'user_name' varchar(50) NOT NULL,
'user_score' bigint(10) UNSIGNED,
'fraction' decimal(13,12) UNSIGNED CHECK(fraction>0) CHECK(fraction<1),
'time_set' TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (user_id)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

Stored in above cookie
Accessed by:
Scoreboard page
Game Scoreboard
MYSQLAdmin
Trigger or loop to remove lowest score and lower all scores below an added score

Section 4: Installation Instructions

Fraction Runner game runs in a web browser. The installation instructions are quite straight forward. You will simply open a modern web browser. Google Chrome is recommended. Please navigate to our site: https://www.fractionrunner.com

Installation for a fresh install: Please see Section 1 which discusses installation of the game.

Permissions for a fresh install: Will need to create a user that has Admin privileges on all capabilities. A user profile that is based off the user listed in the scoreDatabaseFunctions.php file under function()makeConnection. With abilities to insert, delete, select, and update.

User can then be brought to our homepage where there are several options including to start the game.

Appendix A: Implementation Code

WEB PAGE

index.php

```
<?php session_start();?>
<!DOCTYPE html>
<html>
<head>
   <title>Home</title>
   <link href="https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap"
rel="stylesheet">
   <link rel="stylesheet" href="homestyle.css">
 </head>
 <body>
   <!-- NAVBAR -->
   <nav class="navbar">
      <div class="navbar-container">
         <!-- home button/logo -->
         <a href="./mainpage.html" id="home-button">Fraction Runner</a>

         <!-- other navbar items -->
         <ul class="navbar-menu">
            <li class="navbar-item">
               <!-- SESSION USAGE -->
               <!-- line below displays username, put it in the navbar -->
               <p id="user-greet"><?php
               if (isset($_SESSION["username"])){
                 echo 'Hello, '; echo $_SESSION['username'];
               }
               ?></p>
            </li>
            <li class="navbar-item">
               <?php
               if (isset($_SESSION["username"])){
                  echo '<a class="navbar-link" href="./GameLogin/signout.php">Account</a>';
               }
               else{
                  echo '<a class="navbar-link" href="./GameLogin/signlog.php">Log In</a>';
               }
            ?>
```

```
            </li>
            <li class="navbar-item">
                <a class="navbar-link" href="./Gamepage/game.php">Play</a>
            </li>
            <li class="navbar-item">
                <a class="navbar-link" href="./Scorepage/ScorePage.php">Scoreboard</a>
            </li>
        </ul>
    </div>
</nav>

<!-- CONCEPT SECTION -->
<div class="main" id="about">
    <div class="main-container">
        <div class="about-content">
            <h1>WHAT IS FRACTION RUNNER?</h1>
            <h2>Did you know?</h2>
            <div class="concept-txt1">
                <p>If you divide any number by the same number of 9s...</p>
                <p>You'll get a repeating decimal with the same digits as the numerator!</p>
            </div>
            <div class="about-img-container1">
                <img class="concept-gif" src="./division1.gif" alt="123/999">
                <img class="concept-gif" src="./division2.gif" alt="21693/99999">
            </div>
        </div>
        <div class="about-content">
            <div class="concept-txt1">
                <p>
                    Fraction Runner is an endless runner game based on this mathematical idea.
Select your character, input a series of up to 9 digits, and increase your score by avoiding
obstacles as they approach!
                </p>
            </div>
            <div class="about-img-container">
                <img class="concept-img" src="./game-options.png" alt="Character select and
number input options">
            </div>
        </div>
            <p>If you're interested in learning more about the math behind Fraction Runner and
the link between the number 9 and recurring decimals, click <a target="_blank"
href="https://www.youtube.com/watch?v=daro6K6mym8">here</a> to watch a video on the
topic.
            </p>
```

```
            <!-- add a tag to scroll to controls and one for login/scoreboard sections -->
            <p>If you're itching to jump in, scroll to our <a href="#controls">controls</a> section
to learn how to play!</p>
            <!-- add a tag to scroll to login/scoreboard section -->
            <p>Interested in the <a href="#log-and-score">scoreboard</a>? Scroll down to learn
how to get your name at the top of the rankings!</p>
        </div>
    </div>

    <!-- CONTROLS SECTION -->
    <div class="main" id="controls">
        <div class="alt-container">
            <div class="alt-content">
                <h1>CONTROLS</h1>
                <div class="about-img-container">
                    <img src="./controls demo.gif" alt="Demonstration of game controls"
class="controls-gif">
                </div>
                <p>
                    The controls in Fraction Runner are simple. Press the Up Arrow key to jump over red
obstacles, press the Down Arrow key to duck under blue obstacles, and press the Enter key to
destroy green obstacles.
                </p>
                <p>
                    Press the Space key during gameplay to pause or to restart the game after a game
over.
                </p>
            </div>
        </div>
      </div>

      <!-- LOGIN/SCOREBOARD SECTION -->
      <div class="main" id="log-and-score">
        <div class="alt-container">
            <div class="alt-content">
                <h1>HOW DO YOU RANK?</h1>
                <p>
                    Want to see if your name is at the top of the scoreboard? Just trying to keep track of
your personal best score? Start by creating an account!
                </p>
                <p>
                    Fraction Runner can be played without creating an account or signing in, but your
scores will not be saved.
                </p>
```

```
            <p>
                Check out the rankings <a href="./Scorepage/ScorePage.php">here</a>.
            </p>
            <!-- </p> <a href="./Scorepage/ScorePage.php"><button>Scoreboard</button></a> -->
        </div>
      </div>
    </div>

  </body>
</html>
```

homestyles.css

```
/* COLOR PALETTE */
/* BG: 131313*/
/* NAVBAR: 000000*/
/* TEXT: fff4e0*/
/* BUTTONS: 809b4c*/
/* ACCENTS/LINK: 449489*/
/* LINK AFTER: 285763 */

:root {
    --bg-main: #131313;
    --nav-bg: #000000;
    --text-main: #fff4e0;
    --btn-main: #809b4c;
    --accent: #449489;
    --link-aft: #285763;
    --errortxt: red;
}

* {
    padding: 0;
    margin: 0;
    box-sizing: border-box;
    scroll-behavior: smooth;
    /* put the font family in here later */
    font-family: sans-serif;
    font-size: 18px;
    color: var(--text-main);
    text-align: center;
}

body {
    background-color: var(--bg-main);
```

```
}

a {
   color: var(--accent);
   text-decoration: none;
   transition: all 0.3s ease;
   font-size: 1.4rem;
}

p {
   font-size: 1.4rem;
}

a:hover {
   color: var(--text-main);
}

.navbar {
   background: var(--nav-bg);
   height: 70px;
   display: flex;
   justify-content: center;
   align-items: center;
   font-size: 1.2rem;
   position: sticky;
   top: 0;
   /* z index high so that it is always visible */
   z-index: 999;
}

#user-greet {
   font-size: 1rem;
}

.navbar-container {
   display: flex;
   justify-content: space-between;
   height: 90px;
   z-index: 1;
   width: 100%;
   margin: 0 auto;
   padding: 0 20px;
}
```

```
#home-button {  /*add text shadow later*/
   font-family: 'Press Start 2P', sans-serif;
   background-color: var(--btn-main);
   background-image: linear-gradient(to bottom, var(--text-main) 0%, var(--bg-main) 100%);
   background-size: 100%;
   background-clip: text;
   -webkit-background-clip: text;
   -moz-background-clip: text;
   -webkit-text-fill-color: transparent;
   -moz-text-fill-color: transparent;
   display: flex;
   align-items: center;
   cursor: pointer;
   text-decoration: none;
   font-size: 1.4rem;
   width: 200px;
}

.navbar-menu {
   width: 40%;
   display: flex;
   align-items: center;
   list-style: none;
   justify-content: space-evenly;
}

.navbar-item {
   display: flex;
   width: 100px;
   font-size: 1rem;
   margin: 0 10px;
   justify-content: space-evenly;
}

.navbar-link {
   color: var(--accent);
   display: flex;
   align-items: center;
   justify-content: space-evenly;
   text-decoration: none;
   height: 100%;
   transition: all 0.3s ease;
   font-size: 1rem;
}
```

```
button {
   color: var(--nav-bg);
   display: flex;
   justify-content: center;
   align-items: center;
   text-decoration: none;
   padding: 1rem 2rem;
   border: none;
   outline: none;
   cursor: pointer;
   border-radius: 4px;
   background: var(--btn-main);
   font-size: 1.5rem;
   transition: all 0.2s ease;
}

button:hover {
   scale: 1.1;
   color: var(--text-main);
}

h1 {
   font-family: 'Press Start 2P', sans-serif;
}

.navbar-link:hover {
   color: var(--text-main);
   transition: all 0.3s ease;
}

/* ABOUT */
.main {
   background-color: var(--bg-main);
   padding: 5rem 0;
}

.main-container {
   display: flex;
   /* grid-template-columns: 1fr 1fr; */
   align-items: center;
   justify-content: center;
   flex-direction: column;
   /* margin: 0 auto; */
```

```
    height: 90%;
    /* z-index: 1; */
    width: 100%;
    /* max-width: 1500px; */
    padding: 0 50px;
}

.about-content {
    display: flex;
    flex-direction: column;
    color: var(--text-main);
    width: 100%;
}

.about-content h1 {
    font-size: 2.5rem;
    margin-bottom: 32px;
    text-align: center;
}

.about-content h2 {
    font-size: 1.6rem;
    text-align: center;
}

.about-content p {
    margin-top: 1rem;
    font-size: 1.4rem;
    font-weight: 400;
    text-align: center;
}

.about-img-container1 {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}

.about-img-container {
    margin: 1.5rem 0;
}

.concept-txt1 {
```

```
    margin: 1.5rem 0;
}

.concept-gif {
    width: 550px;
    height: 550px;
    margin: 1rem 0;
}

.alt-container {
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    margin: 0 auto;
    height: 90%;
    z-index: 1;
    width: 90%;
    /* max-width: 1500px; */
    padding: 0 50px;
}

.alt-content {
    display: flex;
    flex-direction: column;
    color: var(--text-main);
    align-items: center;
    justify-content: center;
}

.alt-content h1 {
    font-size: 3rem;
    margin-bottom: 32px;
}
```

signup.php

```
<?php
session_start();
include '../Scorepage/scoreDatabaseFunctions.php';
$ranks = new scoreDatabaseFunctions();
$servername ="127.0.0.1";
$username = "fractio3_user";
$password = "edcvfr43edcvfr4";
$dbname = "fractio3_dba";
```

```
$conn = new mysqli($servername, $username, $password, $dbname);

if($conn->connect_error){
          die("connection failed");
}

$name = $_POST["name"];
$pass = $_POST["password"];
$salt = "fractio3_dba";
$pass_encrypted = sha1($pass.$salt);

try {
      if (strlen($name) < 4 || strlen($name) > 49){
         echo "<script>alert('ERROR: Username must be longer than 3 characters and less than
50');</script>";
      }
      else if (strlen($pass) < 4 || strlen($pass) > 49){
         echo "<script>alert('ERROR: Password must be longer than 3 characters and less than
50');</script>";
      }
      else{
         #$insertUser = $ranks->addNewUser($ranks->dbconn, $name, $pass);
         $insertUser = $ranks->addNewUser($ranks->dbconn, $name, $pass_encrypted);
         if (!is_String($insertUser)){
            echo "<script>alert('ERROR: Username already exists');</script>";
           echo "<script type='text/javascript'>location.assign('./signlog.php');</script>";
         }

         else{

            $_SESSION['username'] = $name;
            $_SESSION['loggedin'] = true;

            echo "<script>alert('Successful new user addition! Welcome!');</script>";
           echo "<script type='text/javascript'>location.assign('../index.php');</script>";
         }
      }

   //error states
   } catch(mysqli_sql_exception $e2){
   echo "<script>alert('ERROR: Incorrect database permissions or disconnection. ');</script>";
   echo "<script type='text/javascript'>location.assign('./signlog.php');</script>";
}
```

```
?>
```

signin.php

```
<?php
session_start();
include '../Scorepage/scoreDatabaseFunctions.php';
$ranks = new scoreDatabaseFunctions();



$name = $_POST["name"];
$pass = $_POST["password"];
$salt = "fractio3_dba";
$pass_encrypted = sha1($pass.$salt);

try{

      //fire login function
      #$sql = $ranks->logIn($ranks->dbconn, $name,$pass);
      $sql = $ranks->logIn($ranks->dbconn, $name,$pass_encrypted);
      //in cases where login successful and no error out
      if(!is_string($sql)){

         $_SESSION['username'] = $name;
         $_SESSION['loggedin'] = true;

          echo '<script>alert("Login successful");</script>';
         echo "<script type='text/javascript'>location.assign('../index.php');</script>";
      }

      //failure states
      else{
          echo "<script>alert('ERROR:".$sql."');</script>";
          echo "<script type='text/javascript'>location.assign('./signlog.php');</script>";
      }
   //error states
}catch(mysqli_sql_exception $e2){
   echo "<script>alert('ERROR: Incorrect database permissions or disconnection. ');</script>";
   echo "<script type='text/javascript'>location.assign('./signlog.php');</script>";
}

?>
```

signlog.php

```
<?php
   session_start();
   include 'connection.php';
   if(isset($_SESSION['username'])){
      echo "<script type='text/javascript'>location.assign('../index.php');</script>";
   }
?>
<!DOCTYPE html>
<html>
<head>
         <title>Sign Up and Login</title>
         <link href="https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap"
rel="stylesheet">
         <link rel="stylesheet" type="text/css" href="./style.css">
         <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>

         <!-- NAVBAR -->
   <nav class="navbar">
      <div class="navbar-container">
         <!-- home button/logo -->
         <a href="../index.php" id="home-button">Fraction Runner</a>

         <!-- other navbar items -->
         <ul class="navbar-menu">
            <li class="navbar-item">
               <!-- SESSION USAGE -->
               <!-- line below displays username, put it in the navbar -->
               <p id="user-greet"><?php
               if (isset($_SESSION["username"])){
                 echo 'Hello, '; echo $_SESSION['username'];
               }
               ?></p>
            </li>
            <li class="navbar-item">
               <?php
               if (isset($_SESSION["username"])){
                  echo '<a class="navbar-link" href="../GameLogin/signout.php">Account</a>';
               }
               else{
                  echo '<a class="navbar-link" href="../GameLogin/signlog.php">Log In</a>';
```

```
            }
        ?>
        </li>
        <li class="navbar-item">
            <a class="navbar-link" href="../Gamepage/game.php">Play</a>
        </li>
        <li class="navbar-item">
            <a class="navbar-link" href="../Scorepage/ScorePage.php">Scoreboard</a>
        </li>
      </ul>
    </div>
  </nav>


<div class="container" id="container">
<div class="form-container sign-up-container">

<form action="./signup.php" method="post">
        <h1>Create Account</h1>
        <input type="text" name="name" placeholder="Username">
        <input type="password" name="password" placeholder="Password">
        <button type="submit" name="signup">SignUp</button>
</form>
</div>
<div class="form-container sign-in-container">
        <form action="./signin.php" method="post">
                <h1>Sign In</h1>
        <input type="text" name="name" placeholder="Username">
        <input type="password" name="password" placeholder="Password">
        <!-- Cut because we do not have high enough level encryption to take email addresses
<a href="#">Forgot Your Password</a>-->

        <button type="submit" name="signIn">Sign In</button>
        </form>
</div>
<div class="overlay-container">
        <div class="overlay">
                <div class="overlay-panel overlay-left">
                        <h1>Welcome Back!</h1>
                        <p>Login to track your scoreboard rank!</p>
                        <button class="ghost" id="signIn">Sign In</button>
                </div>
                <div class="overlay-panel overlay-right">
                        <h1>Hello</h1>
```

```
                        <p>Enter your details and start playing!</p>
                        <button class="ghost" id="signUp">Sign Up</button>
                    </div>
            </div>
</div>
</div>
<?php
   #include 'signin.php';
            #include 'signup.php';
?>

<script type="text/javascript">

            const signUpButton = document.getElementById('signUp');
            const signInButton = document.getElementById('signIn');
            const container = document.getElementById('container');

            signUpButton.addEventListener('click', () => {
                        container.classList.add("right-panel-active");
            });
            signInButton.addEventListener('click', () => {
                        container.classList.remove("right-panel-active");

            });

</script>
</body>
</html>
```

style.css

```
:root {
   --bg-main: #131313;
   --nav-bg: #000000;
   --text-main: #fff4e0;
   --btn-main: #809b4c;
   --accent: #449489;
   --link-aft: #285763;
   --errortxt: red;
}

* {
            box-sizing: border-box;
            padding: 0;
```

```
        margin: 0;
        font-size: 18px;
        font-family: sans-serif;
}

body {
        background: var(--bg-main);
        display: flex;
        /* justify-content: center; */
        /* align-items: center; */
        flex-direction: column;
        /* color: var(--text-main); */
        height: 100vh;
        /* margin: -20px 0 50px; */
}

/* NAVBAR */

.navbar {
   background: var(--nav-bg);
   height: 70px;
   display: flex;
   justify-content: center;
   align-items: center;
   font-size: 1.2rem;
   position: sticky;
   top: 0;
   /* z index high so that it is always visible */
   z-index: 999;
}

.navbar-container {
   display: flex;
   justify-content: space-between;
   height: 90px;
   z-index: 1;
   width: 100%;
   margin: 0 auto;
   padding: 0 20px;
}

#home-button {  /*add text shadow later*/
   font-family: 'Press Start 2P', sans-serif;
   background-color: var(--btn-main);
```

```
    background-image: linear-gradient(to bottom, var(--text-main) 0%, var(--bg-main) 100%);
    background-size: 100%;
    background-clip: text;
    -webkit-background-clip: text;
    -moz-background-clip: text;
    -webkit-text-fill-color: transparent;
    -moz-text-fill-color: transparent;
    display: flex;
    align-items: center;
    cursor: pointer;
    text-decoration: none;
    font-size: 1.4rem;
    width: 200px;
}

.navbar-menu {
    width: 40%;
    display: flex;
    align-items: center;
    list-style: none;
    justify-content: space-evenly;
}

.navbar-item {
    display: flex;
    width: 100px;
    font-size: 1rem;
    margin: 0 10px;
    justify-content: space-evenly;
}

.navbar-link {
    color: var(--accent);
    display: flex;
    align-items: center;
    justify-content: space-evenly;
    text-decoration: none;
    height: 100%;
    transition: all 0.3s ease;
    font-size: 1rem;
}

.navbar-link:hover {
    color: var(--text-main);
```

```
    transition: all 0.3s ease;
}

#user-greet {
    font-size: 1rem;
}

/* PAGE CONTENTS */

h1 {
        /* font-weight: bold; */
        font-family: 'Press Start 2P', sans-serif;
        font-size: 1.8rem;
        margin: 0.7rem 0;
}

h2 {
        text-align: center;
}

p {
        font-size: 1rem;
        font-weight: 100;
        /* line-height: 20px; */
        letter-spacing: 0.5px;
        margin: 20px 0 30px;
}

span {
        font-size: 12px;
}

a {
        color: var(--accent);
        font-size: 1rem;
        text-decoration: none;
        margin: 15px 0;
}

button {
        border-radius: 10px;
        /* border: 1px solid #000; */
        outline: none;
        border: none;
```

```
        background-color: var(--btn-main);
        color: var(--nav-bg);
        font-family: 'Press Start 2P', sans-serif;
        font-size: 1rem;
        /* font-weight: bold; */
        padding: 12px 45px;
        /* letter-spacing: 1px; */
        /* text-transform: uppercase; */
  transition: all 0.2s ease;
        cursor: pointer;
}

button:hover {
  scale: 1.1;
  color: var(--text-main);
}

button:active {
        transform: scale(0.95);
}

/* button:focus {
        outline: none;
} */

/* button.ghost {
        background-color: transparent;
        border-color: var(--text-main);
} */

form {
        background-color: var(--text-main);
        display: flex;
        align-items: center;
        justify-content: center;
        flex-direction: column;
        padding: 0 50px;
        height: 100%;
        text-align: center;
}

input {
        background-color: var(--text-main);
        /* border: none; */
```

```
        border: none;
        outline: none;
        border-bottom: 2px solid var(--accent);
        padding: 12px 15px;
        margin: 8px 0;
        width: 100%;
}

.container {
        background-color: var(--bg-main);
        border-radius: 10px;
        box-shadow: 0 14px 28px rgba(0,0,0,0.25),
                    0 10px 10px rgba(0,0,0,0.22);
        position: relative;
        top: 50%;
        left: 50%;
        transform: translate(-50%, -60%);
        overflow: hidden;
        width: 768px;
        max-width: 100%;
        min-height: 480px;
}

.form-container {
        position: absolute;
        top: 0;
        height: 100%;
        transition: all 0.6s ease-in-out;
}

.sign-in-container {
        left: 0;
        width: 50%;
        z-index: 2;
}

.container.right-panel-active .sign-in-container {
        transform: translateX(100%);
}

.sign-up-container {
        left: 0;
        width: 50%;
        opacity: 0;
```

```
	z-index: 1;
}

.container.right-panel-active .sign-up-container {
	transform: translateX(100%);
	opacity: 1;
	z-index: 5;
	animation: show 0.6s;
}

@keyframes show {
	0%, 49.99% {
		opacity: 0;
		z-index: 1;
	}

	50%, 100% {
		opacity: 1;
		z-index: 5;
	}
}

.overlay-container {
	position: absolute;
	top: 0;
	left: 50%;
	width: 50%;
	height: 100%;
	overflow: hidden;
	transition: transform 0.6s ease-in-out;
	z-index: 100;
}

.container.right-panel-active .overlay-container{
	transform: translateX(-100%);
}

.overlay {
	background: var(--bg-main);

	background-repeat: no-repeat;
	background-size: cover;
	background-position: 0 0;
	color: var(--text-main);
```

```
	position: relative;
	left: -100%;
	height: 100%;
	width: 200%;
	transform: translateX(0);
	transition: transform 0.6s ease-in-out;
}

.container.right-panel-active .overlay {
	transform: translateX(50%);
}

.overlay-panel {
	position: absolute;
	display: flex;
	align-items: center;
	justify-content: center;
	flex-direction: column;
	padding: 0 40px;
	text-align: center;
	top: 0;
	height: 100%;
	width: 50%;
	transform: translateX(0);
	transition: transform 0.6s ease-in-out;
}

.overlay-left {
	transform: translateX(-20%);
}

.container.right-panel-active .overlay-left {
	transform: translateX(0);
}

.overlay-right {
	right: 0;
	transform: translateX(0);
}

.container.right-panel-active .overlay-right {
	transform: translateX(20%);
}
```

```
.social-container {
	margin: 20px 0;
}

.social-container a {
	border: 1px solid #DDDDDD;
	border-radius: 50%;
	display: inline-flex;
	justify-content: center;
	align-items: center;
	margin: 0 5px;
	height: 40px;
	width: 40px;
}
```

signout.php

```
<?php session_start();
include '../Scorepage/scoreDatabaseFunctions.php';
?>
<!DOCTYPE html>
<html>
<head>
	<link href="https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap"
rel="stylesheet">
	<link rel="stylesheet" href="./signoutstyle.css">
	<title>Log Out and Account Maintenance</title>
</head>
<body>

	<!-- NAVBAR -->
   <nav class="navbar">
      <div class="navbar-container">
         <!-- home button/logo -->
         <a href="../index.php" id="home-button">Fraction Runner</a>

         <!-- other navbar items -->
         <ul class="navbar-menu">
            <li class="navbar-item">
               <!-- SESSION USAGE -->
               <!-- line below displays username, put it in the navbar -->
               <p id="user-greet"><?php
               if (isset($_SESSION["username"])){
                 echo 'Hello, '; echo $_SESSION['username'];
               }
```

```
            ?></p>
          </li>
          <li class="navbar-item">
            <?php
            if (isset($_SESSION["username"])){
              echo '<a class="navbar-link" href="../GameLogin/signout.php">Account</a>';
            }
            else{
              echo '<a class="navbar-link" href="../GameLogin/signlog.php">Log In</a>';
            }
          ?>
          </li>
          <li class="navbar-item">
            <a class="navbar-link" href="../Gamepage/game.php">Play</a>
          </li>
          <li class="navbar-item">
            <a class="navbar-link" href="../Scorepage/ScorePage.php">Scoreboard</a>
          </li>
        </ul>
      </div>
    </nav>



        <!-- PAGE CONTENT -->
        <div class="page-container">
                <h1 class="page-header">Account Maintenance</h1>
                <div class="content-container">
                        <form action="./logout.php" method="post">
                        <h2>Sign Out</h2>
                        <button type="submit" name="logOut">Log Out</button>
                        </form>
                </div>
                <div class="content-container">
                        <form action="./changepass.php" method="post">
                        <h2>Change Password</h2>
                        <input type="password" name="oldpassword" placeholder="Current
Password">
                        <input type="password" name="newpassword" placeholder="New
Password">
                        <button type="submit" name="changePass">Change Password</button>
                        </form>
                </div>
                <div class="content-container">
```

```
                <form action="./areyousure.php" method="post">
                <h2>Delete Account</h2>
                <input type="password" name="password" placeholder="Password">
                <button type="submit" name="deleteAccount">Delete Account</button>
                </form>
            </div>
        </div>
</body>
</html>
```

signoutstyle.css

```
/* COLOR PALETTE */
/* BG: 131313*/
/* NAVBAR: 000000*/
/* TEXT: fff4e0*/
/* BUTTONS: 809b4c*/
/* ACCENTS/LINK: 449489*/
/* LINK AFTER: 285763 */

:root {
   --bg-main: #131313;
   --nav-bg: #000000;
   --text-main: #fff4e0;
   --btn-main: #809b4c;
   --accent: #449489;
   --link-aft: #285763;
   --errortxt: red;
}

* {
   padding: 0;
   margin: 0;
   box-sizing: border-box;
   scroll-behavior: smooth;
   /* put the font family in here later */
   font-family: sans-serif;
   font-size: 18px;
   color: var(--text-main);
   text-align: center;
}

body {
   background-color: var(--bg-main);
}
```

```
a {
   color: var(--accent);
   text-decoration: none;
   transition: all 0.3s ease;
   font-size: 1.4rem;
}

p {
   font-size: 1.4rem;
}

a:hover {
   color: var(--text-main);
}

.navbar {
   background: var(--nav-bg);
   height: 70px;
   display: flex;
   justify-content: center;
   align-items: center;
   font-size: 1.2rem;
   position: sticky;
   top: 0;
   /* z index high so that it is always visible */
   z-index: 999;
}

#user-greet {
   font-size: 1rem;
}

.navbar-container {
   display: flex;
   justify-content: space-between;
   height: 90px;
   z-index: 1;
   width: 100%;
   margin: 0 auto;
   padding: 0 20px;
}

#home-button {  /*add text shadow later*/
```

```
    font-family: 'Press Start 2P', sans-serif;
    background-color: var(--btn-main);
    background-image: linear-gradient(to bottom, var(--text-main) 0%, var(--bg-main) 100%);
    background-size: 100%;
    background-clip: text;
    -webkit-background-clip: text;
    -moz-background-clip: text;
    -webkit-text-fill-color: transparent;
    -moz-text-fill-color: transparent;
    display: flex;
    align-items: center;
    cursor: pointer;
    text-decoration: none;
    font-size: 1.4rem;
    width: 200px;
}

.navbar-menu {
    width: 40%;
    display: flex;
    align-items: center;
    list-style: none;
    justify-content: space-evenly;
}

.navbar-item {
    display: flex;
    width: 100px;
    font-size: 1rem;
    margin: 0 10px;
    justify-content: space-evenly;
}

.navbar-link {
    color: var(--accent);
    display: flex;
    align-items: center;
    justify-content: space-evenly;
    text-decoration: none;
    height: 100%;
    transition: all 0.3s ease;
    font-size: 1rem;
}
```

```
button {
   font-family: 'Press Start 2P', sans-serif;
   color: var(--nav-bg);
   display: flex;
   justify-content: center;
   align-items: center;
   text-decoration: none;
   padding: 1rem 1rem;
   border: none;
   outline: none;
   cursor: pointer;
   border-radius: 10px;
   background: var(--btn-main);
   font-size: 1.3rem;
   transition: all 0.3s ease;
   /* width: max-content; */
}

button:hover {
   scale: 1.1;
   color: var(--text-main);
}

input {
   font-size: 1.2rem;
   width: 60%;
   padding: 12px 10px;
   color: var(--nav-bg);
   background-color: var(--text-main);
   border: none;
   outline: none;
   border-radius: 10px;
   margin-bottom: 1.1rem;
}

form {
   display: flex;
   flex-direction: column;
   width: 100%;
   align-items: center;
}

.page-container {
   margin: 2rem 0;
```

```
    display: flex;
    flex-direction: column;
    align-items: center;
}

.page-header {
    font-family: 'Press Start 2P', sans-serif;
    font-size: 2rem;
    margin-bottom: 1rem;
}

h2 {
    font-family: 'Press Start 2P', sans-serif;
    margin: 1rem 0;
    font-size: 1.5rem;
}

.content-container {
    margin: 1.5rem 0;
}
```

logout.php

```
<?php session_start();
session_unset();
session_destroy();
echo "<script type='text/javascript'>location.assign('./signlog.php');</script>";
?>
```

deleter.php

```
<?php session_start();
include '../Scorepage/scoreDatabaseFunctions.php';
$ranks = new scoreDatabaseFunctions();
$name = $_SESSION['username'];
$score = $_GET['userscore'];

try{

        //fire user delete function
        #$sql = $ranks->logIn($ranks->dbconn, $name,$pass);
        $sql= $ranks->deleteUser($ranks->dbconn, $name, $pass_encrypted);
        //in cases where delete unsuccessful and error out
        if(!is_string($sql) && !is_array($sql)){
```

```
            echo '<script>alert("ERROR: User does not exist, or you are no longer logged
in.");</script>';
           echo "<script type='text/javascript'>location.assign('./signout.php');</script>";
        }
        else if(is_array($sql)){
           echo "<script>alert('ERROR: Incorrect Password');</script>";
            echo "<script type='text/javascript'>location.assign('./signout.php');</script>";}
        //Success State
        else{
            echo "<script>alert('Account deleted.');</script>";
            session_destroy();
            echo "<script type='text/javascript'>location.assign('./signlog.php');</script>";
        }

    //no databaseerror states
}catch(mysqli_sql_exception $e2){
    echo "<script>alert('ERROR: Incorrect database permissions or disconnection. ');</script>";
    echo "<script type='text/javascript'>location.assign('./signlog.php');</script>";
}
?>
```

game.php

```
<?php
session_start();
include '../Scorepage/scoreDatabaseFunctions.php';
$ranks = new scoreDatabaseFunctions();
// if (!isset($_SESSION['username'])){
//     $loggedoutmsg = "You are not currently logged in, your data will not be saved.";
//     echo $loggedoutmsg;
// }else{echo "logged in";}
?>

<!DOCTYPE html>

<html lang="en">
 <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Game</title>
    <link href="https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="gamestyles.css" />
</head>
```

```
<body>
  <!-- NAVBAR -->
  <nav class="navbar">
    <div class="navbar-container">
      <!-- home button/logo -->
      <a href="../index.php" id="home-button">Fraction Runner</a>

      <!-- other navbar items -->
      <ul class="navbar-menu">
        <li class="navbar-item">
          <!-- SESSION USAGE -->
          <!-- line below displays username, put it in the navbar -->
          <p id="user-greet"><?php
          if (isset($_SESSION["username"])){
            echo 'Hello, '; echo $_SESSION['username'];
          }
          ?></p>
        </li>
        <li class="navbar-item">
          <?php
          if (isset($_SESSION["username"])){
            echo '<a class="navbar-link" href="../GameLogin/signout.php">Account</a>';
          }
          else{
            echo '<a class="navbar-link" href="../GameLogin/signlog.php">Log In</a>';
          }
        ?>
        </li>
        <li class="navbar-item">
          <a class="navbar-link" href="../Gamepage/game.php">Play</a>
        </li>
        <li class="navbar-item">
          <a class="navbar-link" href="../Scorepage/ScorePage.php">Scoreboard</a>
        </li>
      </ul>
    </div>
  </nav>

  <!-- HTML canvas element -->
  <div id="canvas-container" class="game-container hide">
    <canvas id="canvas1"></canvas>

    <!-- game assets to be handled with load function to avoid image errors -->
    <img id="cactusImage" class="gameImg" src="./img/cactus.png">
```

```
<!-- CHARACTER 1 IMAGES -->
<img id="charBAttack0" class="gameImg" src="./img/charb-attack-0.png">
<img id="charBDuck0" class="gameImg" src="./img/charb-duck-0.png">
<img id="charBDuck1" class="gameImg" src="./img/charb-duck-1.png">
<img id="charBDuck2" class="gameImg" src="./img/charb-duck-2.png">
<img id="charBDuck3" class="gameImg" src="./img/charb-duck-3.png">
<img id="charBHurt0" class="gameImg" src="./img/charb-hurt-0.png">
<img id="charBJump0" class="gameImg" src="./img/charb-jump-0.png">
<img id="charBFall0" class="gameImg" src="./img/charb-jump-1.png">
<img id="charBRun0" class="gameImg" src="./img/charb-run-0.png">
<img id="charBRun1" class="gameImg" src="./img/charb-run-1.png">
<img id="charBRun2" class="gameImg" src="./img/charb-run-2.png">
<img id="charBRun3" class="gameImg" src="./img/charb-run-3.png">
<img id="charBStatic0" class="gameImg" src="./img/charb-static-0.png">
<!-- CHARACTER 2 IMAGES -->
<img id="charRAttack0" class="gameImg" src="./img/charr-attack-0.png">
<img id="charRDuck0" class="gameImg" src="./img/charr-duck-0.png">
<img id="charRDuck1" class="gameImg" src="./img/charr-duck-1.png">
<img id="charRDuck2" class="gameImg" src="./img/charr-duck-2.png">
<img id="charRDuck3" class="gameImg" src="./img/charr-duck-3.png">
<img id="charRHurt0" class="gameImg" src="./img/charr-hurt-0.png">
<img id="charRJump0" class="gameImg" src="./img/charr-jump-0.png">
<img id="charRFall0" class="gameImg" src="./img/charr-jump-1.png">
<img id="charRRun0" class="gameImg" src="./img/charr-run-0.png">
<img id="charRRun1" class="gameImg" src="./img/charr-run-1.png">
<img id="charRRun2" class="gameImg" src="./img/charr-run-2.png">
<img id="charRRun3" class="gameImg" src="./img/charr-run-3.png">
<img id="charRStatic0" class="gameImg" src="./img/charr-static-0.png">
<!-- CHARACTER 3 IMAGES -->
<img id="charVAttack0" class="gameImg" src="./img/charv-attack-0.png">
<img id="charVDuck0" class="gameImg" src="./img/charv-duck-0.png">
<img id="charVDuck1" class="gameImg" src="./img/charv-duck-1.png">
<img id="charVDuck2" class="gameImg" src="./img/charv-duck-2.png">
<img id="charVDuck3" class="gameImg" src="./img/charv-duck-3.png">
<img id="charVHurt0" class="gameImg" src="./img/charv-hurt-0.png">
<img id="charVJump0" class="gameImg" src="./img/charv-jump-0.png">
<img id="charVFall0" class="gameImg" src="./img/charv-jump-1.png">
<img id="charVRun0" class="gameImg" src="./img/charv-run-0.png">
<img id="charVRun1" class="gameImg" src="./img/charv-run-1.png">
<img id="charVRun2" class="gameImg" src="./img/charv-run-2.png">
<img id="charVRun3" class="gameImg" src="./img/charv-run-3.png">
<img id="charVStatic0" class="gameImg" src="./img/charv-static-0.png">
<!-- OBSTACLES -->
<img id="jumpObs" class="gameImg" src="./img/jump obs.png">
```

```
      <img id="duckObs" class="gameImg" src="./img/duck obs.png">
      <img id="attackObs" class="gameImg" src="./img/attack obs.png">
      <!-- BG IMAGE LAYERS -->
      <img id="bgLayer1" class="gameImg" src="./img/bg-layer1.png">
      <img id="bgLayer2" class="gameImg" src="./img/bg-layer2.png">
      <img id="bgLayer3" class="gameImg" src="./img/bg-layer3.png">
      <img id="bgLayer4" class="gameImg" src="./img/bg-layer4.png">
   </div>

   <!-- input for number string -->
   <div id="start-container" class="popup options">
      <h3 id="charselect-text">Select Character</h3>
      <div class="character-select">
         <div class="characters">
            <img class="char-image" src="./img/charb-static-0.png" id="charblue">
            <img class="char-image" src="./img/charr-static-0.png" id="charred">
            <img class="char-image" src="./img/charv-static-0.png" id="charvio">
         </div>
         <p id="no-char-selected" class="invisible">Please select your character.</p>
      </div>
      <div class="num-input">
         <label for="rep-digits">
            <h3>Input your repeating digits:</h3>
         </label>
            <input type="number" name="rep-digits" id="user-num" class="user-num-input">
            <p id="invalid-num" class="invisible">Please enter a nonzero number with 1 to 9
digits.</p>
            <button id="game-start">START GAME</button>
      </div>
   </div>

   <script type= "module" src="modules/main.js"></script>

 </body>
</html>
```

gamestyles.css

```
/* COLOR PALETTE */
/* BG: 131313*/
/* NAVBAR: 000000*/
/* TEXT: fff4e0*/
/* BUTTONS: 809b4c*/
/* ACCENTS/LINK: 449489*/
/* LINK AFTER: 285763 */
```

```
:root {
    --bg-main: #131313;
    --nav-bg: #000000;
    --text-main: #fff4e0;
    --btn-main: #809b4c;
    --accent: #449489;
    --link-aft: #285763;
    --errortxt: red;
}

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-size: 18px;
    font-family: sans-serif;
    font-size: 18px;
    color: var(--text-main);
    text-align: center;
}

body {
    background-color: var(--bg-main);
}

a {
    color: var(--accent);
    text-decoration: none;
    transition: all 0.3s ease;
}

a:hover {
    color: var(--text-main);
}

.navbar {
    background: var(--nav-bg);
    height: 70px;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 1.2rem;
    position: sticky;
```

```
    top: 0;
    /* z index high so that it is always visible */
    z-index: 999;
}

.navbar-container {
    display: flex;
    justify-content: space-between;
    height: 90px;
    z-index: 1;
    width: 100%;
    margin: 0 auto;
    padding: 0 20px;
}

#home-button {  /*add text shadow later*/
    font-family: 'Press Start 2P', sans-serif;
    background-color: var(--btn-main);
    background-image: linear-gradient(to bottom, var(--text-main) 0%, var(--bg-main) 100%);
    background-size: 100%;
    background-clip: text;
    -webkit-background-clip: text;
    -moz-background-clip: text;
    -webkit-text-fill-color: transparent;
    -moz-text-fill-color: transparent;
    display: flex;
    align-items: center;
    cursor: pointer;
    text-decoration: none;
    font-size: 1.4rem;
    width: 200px;
}

.navbar-menu {
    width: 40%;
    display: flex;
    align-items: center;
    list-style: none;
    justify-content: space-evenly;
}

.navbar-item {
    display: flex;
    width: 100px;
```

```
    font-size: 1rem;
    margin: 0 10px;
    justify-content: space-evenly;
}

.navbar-link {
    color: var(--accent);
    display: flex;
    align-items: center;
    justify-content: space-evenly;
    text-decoration: none;
    height: 100%;
    transition: all 0.3s ease;
    font-size: 1rem;
}

.navbar-link:hover {
    color: var(--text-main);
    transition: all 0.3s ease;
}

button {
    font-family: 'Press Start 2P', sans-serif;
    color: var(--nav-bg);
    display: flex;
    justify-content: center;
    align-items: center;
    text-decoration: none;
    padding: 1rem 1rem;
    border: none;
    outline: none;
    cursor: pointer;
    border-radius: 10px;
    background: var(--btn-main);
    font-size: 1.5rem;
    transition: all 0.3s ease;
}

input {
    font-size: 1.5rem;
    width: 60%;
    padding: 12px 10px;
    color: var(--nav-bg);
    border: none;
```

```
    outline: none;
    border-radius: 10px;
}

button:hover {
    scale: 1.1;
    color: var(--text-main);
}

#user-greet {
    font-size: 1rem;
}

/* GAME CANVAS */

#canvas1 {
    border: 5px solid var(--nav-bg);
    position: absolute;
    top: 55%;
    left: 50%;
    transform: translate(-50%, -50%);
    width: 960px;
    height: 540px;
    /* cursor: pointer; */
}

.gameImg {
    display: none;
}

.popup {
    background-color: var(--bg-main);
    position: absolute;
    top: 55%;
    left: 50%;
    transform: translate(-50%, -50%);
    width: 80%;
    height: 70%;
    border: 3px var(--nav-bg) solid;
    border-radius: 10px;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
```

```
    font-size: 2rem;
}

h3 {
    font-family: 'Press Start 2P', sans-serif;
    font-size: 1.5rem;
    position: relative;
    margin-bottom: 15px;
    text-align: center;
}

#charselect-text {
    position: absolute;
    top: 25px;
}

.character-select {
    width: 100%;
    display: flex;
    flex-direction: column;
    text-align: center;
    justify-content: center;
    align-items: center;
    position: absolute;
    top: 65px;
}

.characters {
    display: flex;
    flex-direction: row;
    width: 100%;
    align-items: center;
    justify-content: center;
}

.char-image {
    width: 96px;
    height: 96px;
    padding: 1px;
    border: 2px solid var(--accent);
    border-radius: 10px;
    flex-direction: row;
    align-items: center;
    justify-content: center;
```

```
    margin: 5px;
    cursor: pointer;
}

.selected {
    border: 4px solid var(--text-main);
    border-radius: 10px;
}

.num-input {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    margin-top: 170px;
}

#invalid-num, #no-char-selected {  /*styling for error text on game option container*/
    color: var(--errortxt);
    font-size: 1rem;
    margin: 5px 0;
}

.invisible {
    visibility: hidden;
}

.error {
    outline: none;
    border: 2px solid var(--errortxt);
    border-radius: 4px;
}

.hide { /*can be toggled on and off; gameImg should NOT be toggled, will be displayed using
JS*/
    display: none;
}
```

scoresaver.php

```
<?php
session_start();
include '../Scorepage/scoreDatabaseFunctions.php';
$ranks = new scoreDatabaseFunctions();
$userscore = $_POST['score'];
```

```
echo $userscore;
$name = $_SESSION['username'];
$digits = $_SESSION['digits'];
if (isset($_SESSION['username'])){
   try{
      $sql = $ranks->setUserScore($ranks->dbconn, $name, $userscore, $digits);
      //error codes
      if (is_string($sql)){
         echo "<script>alert('ERROR:".$sql."');</script>";

      }
      else{
         echo "<script>alert('successfully uploaded');</script>";
      }

   }catch(mysqli_sql_exception $e){
      echo "<script>alert('ERROR: Incorrect database permissions or disconnection. ');</script>";
      echo "<script type='text/javascript'>location.assign('./signlog.php');</script>";
   }
}
else{
      echo "<script>alert('successfully uploaded');</script>";
   }
?>
```

fracsaver.php

```
<?php session_start();
include '../Scorepage/scoreDatabaseFunctions.php';
$ranks = new scoreDatabaseFunctions();
$digits = $_POST['userInput'];
$decimal = $_POST['decimal'];
$divisor = $_POST['divisor'];
$_SESSION['digits'] = $digits;

try{

      //fire storage function
      $sql = $ranks->addNewDigits($ranks->dbconn, $digits, $decimal, $divisor);
      //in cases where login successful and no error out
      if(is_string($sql) && $sql == "0"){
          //echo '<script>alert("INPUT successful, First generation of these numbers");</script>';
         $_SESSION['timesgenned'] = 0;
         $timesgenned = $sql;
         echo $timesgenned;
```

```
        }
        else if (is_int($sql)){
          // echo '<script>alert("INPUT successful, numbers generated before.");</script>';
           $_SESSION['timesgenned'] = $sql;
           $timesgenned = $sql;
           echo $timesgenned;
        }
        //failure states
        else{
            //echo "<script>alert('ERROR:".$sql."');</script>";
            $_SESSION['timesgenned'] = "";
            $timesgenned = $sql;
            echo $timesgenned;
        }

    //error states
}catch(mysqli_sql_exception $e2){
    echo "ERROR: Incorrect database permissions or
disconnection.".mysqli_error($ranks->dbconn);
}

?>
```

main.js

```
import { Player } from './player.js';
import { InputHandler } from './input.js';
import { Background, NumberString } from './background.js';
import { JumpObstacle, DuckObstacle, AttackObstacle } from './obstacle.js';
import { UI } from './UI.js';
var script = document.createElement('script');
script.src = 'https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js'; // Check
https://jquery.com/ for the current version
document.getElementsByTagName('head')[0].appendChild(script);

//waits for webpage to fully load before executing function
window.addEventListener("load", function() {
    //get HTML page elements

    const gameContainer = document.getElementById("canvas-container");
    const optContainer = this.document.getElementById("start-container");
    const canvas = document.getElementById("canvas1");
    const startBtn = document.getElementById("game-start");
    const inputBar = document.getElementById("user-num");
    const numMessage = document.getElementById("invalid-num");
```

```
const charMessage = this.document.getElementById("no-char-selected");
const charRed = document.getElementById("charred");
const charBlue = this.document.getElementById("charblue");
const charVio = this.document.getElementById("charvio");
let selectedChar = "";  //variable to pass selected character to game
//DB variables
let divisor = "";
let numerator;
let userDecimal;
let timesnumgenned = "";


//canvas mode set to 2d
const ctx = canvas.getContext("2d");


//manually setting canvas width and height for correct image scaling
//please come back and adjust aspect ratios at some point :')
const CANVAS_WIDTH = canvas.width = 960;
const CANVAS_HEIGHT = canvas.height = 540;

//REFACTORING FOR BETTER ORGANIZATION
//GAME CLASS: instantiates objects required for game to function
class Game {
   constructor(width, height) {

      this.width = width;
      this.height = height;
      this.speed = 3; //set initial game speed
      this.maxSpeed = 3;
      this.groundHeight = 72; //height of ground that player/obstacles needs to stand on top of
      this.scroll;
      this.userNum = inputBar.value;
      this.bg = new Background(this);
      this.player = new Player(this, "charB");
      this.input = new InputHandler(this);
      this.UI = new UI(this);
      this.pause = false;
      this.obstacles = []; //array to hold existing game obstacles
      this.particles = []; //array to hold particle effects
      this.spawnTimer = 0; //when timer reaches value in interval, spawn new obstacle
      this.spawnInterval = 1000; //initialize time to new obstacle to one second (measure in
ms)
      this.testMode = false;   //set test mode to true; hitboxes will be visible
```

```
        this.gameTimer = 0;         //initialize game timer
        this.gameOver = false;
        this.intro = true;     //indicates whether or not game is in division intro
        this.introActive = true;    //toggle whether or not game shows division intro
        this.score = 0; //initialize game score
        this.fontColor = 'black';   //color for UI text
        this.player.currentState = this.player.states[0];
        this.player.currentState.enter();
    }
    update(dt) {
        if (!this.pause && !this.gameOver) {  //calls update function if game is not paused or
game over
            //update game timer
            this.gameTimer += dt;
            //call bg update function
            this.bg.update();
            //call player update function
            this.player.update(this.input.keys, dt);
            //handle obstacle update here
            if (this.spawnTimer > this.spawnInterval) {
                this.addObstacle();
                this.spawnTimer = 0; //reset timer back to zero
                this.spawnInterval = Math.floor(Math.random() * (3000 - 1000 + 1)) + 1000;
//randomize spawn interval to between 1-3s
            }
            else {
                this.spawnTimer += dt; //add delta time to spawn timer
            }
            this.obstacles.forEach(obstacle => {
                obstacle.update(dt);
            });
            //updating particle effects
            this.particles.forEach((particle, index) => {
                particle.update();
            });
            //use filter method to remove particles and obstacles that are off screen to avoid
image jitter
            this.particles = this.particles.filter(particle => !particle.offScreen);
            this.obstacles = this.obstacles.filter(obstacle => !obstacle.offScreen);
            }

            if (this.gameTimer > 3000) {
                this.speed *= 1.01;
                this.maxSpeed *= 1.01;
```

```
            this.gameTimer = 0;
            console.log(this.speed);
        }
    }
    draw(context) {
        this.bg.draw(context);
        this.player.draw(context);
        this.obstacles.forEach(obstacle => {
            obstacle.draw(context);
        });
        this.particles.forEach((particle) => {
            particle.draw(context);
        });
        this.UI.draw(context);
    }
    addObstacle() {
        let getRandom = Math.random(); //gets a random value to determine type of object to
spawn
        if (getRandom < 0.3) {
            this.obstacles.push(new JumpObstacle(this));
        }
        else if (getRandom >= 0.3 && getRandom < 0.6) {
            this.obstacles.push(new AttackObstacle(this));
        }
        else {
            this.obstacles.push(new DuckObstacle(this));
        }
        //console.log(this.obtacles);
    }
    togglePause() {      //pause menu will need to draw some things to the canvas on top of
player and enemy and stuff, may need to look into a way to do that WITHOUT stopping
requestAnimationFrame
        this.pause = !this.pause;   //sets whatever pause boolean is to the opposite
    }
    showInitMenu() {
        //hide game container
        gameContainer.classList.add("hide");

        //empty input bar and divisor variables in preparation for more input
        inputBar.value = "";
        divisor = "";

        //pull up game options window again, since it has the first call to animate() shouldn't
need to put that in manually?
```

```
        optContainer.classList.remove("hide");
    }
    reset() {   //reset all game data
        this.gameTimer = 0;
        this.score = 0;
        this.obstacles = [];    //empty the obstacle array
        this.speed = 3;        //reset initial game speed
        this.maxSpeed = 3;
        this.spawnTimer = 0;
        this.spawnInterval = 1000; //reset spawn interval
        this.gameOver = false; //reset game over and pause to false only before game start
        this.pause = false;
    }
    restart() {     //function to restart game, should be able to be called either from game over
screen or generally from a pause menu
        //call object reset functions
        this.player.reset();
        this.bg.reset();
        this.reset();
    }
}

const g = new Game(CANVAS_WIDTH, CANVAS_HEIGHT);
//console.log(g);

//value needed to calculate delta time for frame rate delta time for frame rate
let prevTime;

function animate(newTime) {
    if (!g.gameOver) {
        if (prevTime == null) {
            prevTime = newTime;
            requestAnimationFrame(animate);
            return;
        }

        const dt = newTime - prevTime;
        prevTime = newTime;
        ctx.clearRect(0, 0, CANVAS_WIDTH, CANVAS_HEIGHT);
        g.update(dt);
        g.draw(ctx);
        requestAnimationFrame(animate);
    }
}
```

```
//EVENT LISTENERS FOR CHARACTER SELECT
charBlue.addEventListener("click", ()=> {
    selectedChar = "charB";
    charBlue.classList.add("selected");
    charRed.classList.remove("selected");
    charVio.classList.remove("selected");
    charMessage.classList.add("invisible");
});

charRed.addEventListener("click", () => {
    selectedChar = "charR";
    charRed.classList.add("selected");
    charBlue.classList.remove("selected");
    charVio.classList.remove("selected");
    charMessage.classList.add("invisible");
});

charVio.addEventListener("click", () => {
    selectedChar = "charV";
    charVio.classList.add("selected");
    charBlue.classList.remove("selected");
    charRed.classList.remove("selected");
    charMessage.classList.add("invisible");
});

//validate user input from button click
startBtn.addEventListener("click", () => {
    if (selectedChar == "") {
        charMessage.classList.remove("invisible");
        return; //prevents game from firing any other code from this event listener until all
options are selected
    }
    else if (inputBar.value < 1 || inputBar.value > 999999999 || inputBar.value == "") {
        //invalid input, reveals error messaging, will not allow game start
        numMessage.classList.remove("invisible");
        inputBar.classList.add("error");
        return; //prevents game from firing any other code from this event listener until all
options are selected
    }
    else {
        //complete setup before hiding game start options and starting game
        g.scroll = new NumberString(g, inputBar.value, 1);  //sets scrolling number string
        g.player.character = selectedChar;  //sets player character
```

```
        numMessage.classList.add("invisible");
        charMessage.classList.add("invisible");
        inputBar.classList.remove("error");
        optContainer.classList.add("hide");
        gameContainer.classList.remove("hide");
        //call to math animation function here before game start (maybe include an option to
turn it off)
        //put any and all prerequisites to game play BEFORE first call to animate() in an event
listener (maybe also a start message)
            g.restart();
            animate(0);
    }
    //console.log(inputBar.value);

    //gets numbers for database entry
    while (divisor.length < inputBar.value.length) {
        divisor += "9";
    }
    //setting up user decimal
    userDecimal = "0." + inputBar.value + inputBar.value;

    //variables for database are in string format because anything else causes rounding errors
    divisor = divisor;
    numerator = inputBar.value;

    console.log("user input: " + numerator);
    console.log("divisor: " + divisor);
    console.log("decimal: " + userDecimal);

    //sends data to the database, returns amount of times the number was previously
generated.
    $.ajax({
    type:"POST",
    data:{"divisor": divisor, "decimal": userDecimal, "userInput": numerator},
    url:'./fracsaver.php',
    success: function (timesgenned) {
        console.log("success");
        timesnumgenned = timesgenned;
        window.TIMESGENNED = timesnumgenned;   //attempting to pass this variable as a
window attribute so it can be displayed in UI
        console.log(timesnumgenned)
        //USE TIMESNUMGENNED TO FILL PLACEHOLDER FOR TIMES NUMBER PREVIOUSLY
GENERATED
```

```
        }
      });

    });
});
```

background.js

```
//breaking up bg into layers for parallaxing with finalized art assets
class Layer {
    constructor(game, width, height, speedModifier, image) {
        this.game = game;
        this.width = width;
        this.height = height;
        this.speedModifier = speedModifier;
        this.image = image;
        this.x = 0;
        this.y = 0;
    }
    update(){
        if (this.x < -this.width)
            this.x = 0;
        else
            this.x -= this.game.speed * this.speedModifier;
    }
    draw(context){
        //draw first image
        context.drawImage(this.image, this.x, this.y, this.width, this.height);
        //draw second image for seamless repeat
        context.drawImage(this.image, this.x + this.width, this.y, this.width, this.height);
    }
}

export class NumberString {
    constructor(game, userNum, speedModifier) {
        //basic object attributes
        this.game = game;
        this.baseNum = userNum;
        this.speedModifier = speedModifier;
        this.x = 0;
        this.y = this.game.height - 20;
        this.numPrefix = "0.";
        this.scrollNum = "";
        this.scrollWidth;
        //fill a string with numbers to scroll across bottom of canvas area
```

```
        while (this.scrollNum.length < 25) {
            this.scrollNum += this.baseNum;
        }
        //console.log(this.scrollNum);
    }
    draw(context) {
        //set font properties
        context.font = "68px Courier New";
        //draw font portions
        //draw numPrefix first
        //context.fillText(this.numPrefix, this.x, this.y);
        //draw scroll num
        //get prefix width
        //let pWidth = context.measureText(this.numPrefix).width;
        //reset fill style to draw text in black
        context.fillStyle = 'black';
        context.fillText(this.scrollNum, this.x, this.y);
        //get width of repeating digits
        this.scrollWidth = context.measureText(this.scrollNum).width;
        //draw second instance of digit string
        context.fillText(this.scrollNum, (this.x + this.scrollWidth), this.y);

    }
    update() {
        this.x -= this.game.speed * this.speedModifier;
        //when object scrolls off screen, remove it and/or reset its position
        if (this.x < 0 - this.scrollWidth) {
            this.x = 0 - this.game.speed;
            }
            //console.log(this.scrollWidth);
    }
}

//for refactoring numstring class: probably wrap it up in along with layers in bg class?
//can try to fix the issue with the weird hitching at the same time
export class Background {
    constructor(game) {
        this.game = game;
        this.width = 960;
        this.height = 540; //would want to make each layer image same height as base canvas
height
        this.layer1image = document.getElementById("bgLayer1");    //this will match the id of an
image layer in the HTML document later; they will have the class .gameImg as well
```

```
        this.layer2image = document.getElementById("bgLayer2");    //this will match the id of an
image layer in the HTML document later; they will have the class .gameImg as well
        this.layer3image = document.getElementById("bgLayer3");    //this will match the id of an
image layer in the HTML document later; they will have the class .gameImg as well
        this.layer4image = document.getElementById("bgLayer4");    //this will match the id of an
image layer in the HTML document later; they will have the class .gameImg as well
        this.layer1 = new Layer(this.game, this.width, this.height, 0, this.layer1image);
        this.layer2 = new Layer(this.game, this.width, this.height, 0.4, this.layer2image);
        this.layer3 = new Layer(this.game, this.width, this.height, 0.2, this.layer3image);
        this.layer4 = new Layer(this.game, this.width, this.height + 10, 1, this.layer4image);
        this.backgroundLayers = [this.layer1, this.layer2, this.layer3, this.layer4]; //array to hold
each layer of bg
    }
    update(){
        //call update method for each layer in bg layers array
        this.backgroundLayers.forEach(layer => {
            layer.update();
        });
        this.game.scroll.update();
    }
    draw(context){
        this.backgroundLayers.forEach(layer => {
            layer.draw(context);
        });
        this.game.scroll.draw(context);
    }
    reset() {
        this.backgroundLayers.forEach(layer => {
            layer.x = 0;
        });
    }
}
```

effects.js

```
class Particle {
    constructor(game) {
        this.game = game;
        this.offScreen = false; //when too small to be visible, erase
    }
    update() {
        this.x -= this.speedX + this.game.speed;
        this.y -= this.speedY;
        this.size *= 0.95;
        if (this.size < 0.5)    //when size becomes smaller than half a pixel, remove from screen
```

```
        this.offScreen = true;
    }
}

export class Dust extends Particle {
    constructor(game, x, y) {
        super(game);
        this.size = Math.random() * 10 + 7;  //generate particle size between 10 to 17 px
        this.x = x;
        this.y = y;
        this.speedX = Math.random();    //x and y movement speeds: random number between 0 to
1
        this.speedY = Math.random();
        this.color = 'rgba(230, 230, 230, 0.2)';
    }
    draw(context) {
        //draw a square with width/height of size
        context.fillStyle = this.color;
        context.fillRect(this.x, this.y, this.size, this.size);
    }
}

export class Burst extends Particle {
    constructor(game, x, y) {
        super(game);
        this.size = Math.random() * 20 + 7;  //generate particle size between 20 to 27 px for bigger
dust cloud
        this.x = x;
        this.y = y;
        this.speedX = Math.random();    //x and y movement speeds: random number between 0 to
1
        this.speedY = Math.random();
        this.color = 'rgba(230, 230, 230, 0.2)';
    }
    draw(context) {
        //draw a square with width/height of size
        context.fillStyle = this.color;
        context.fillRect(this.x, this.y, this.size, this.size);
    }
}
```

input.js

```
//global values for keyboard controls
window.JUMP = 'ArrowUp';
```

```
window.DUCK = 'ArrowDown';
window.ATTACK = 'Enter';
window.PAUSE = ' ';
window.DEBUGMODE = 'd';

export class InputHandler {
   constructor(game) {
      this.game = game;
      //similar to pygame getpressed function; array tracks all keys currently pressed?
      this.keys = [];
      //keydown event to add key pressed into keys array
      window.addEventListener("keydown", e => {
         //select which keys to add to keys array (connected to constant key values above)
         if ((e.key === window.JUMP
           || e.key === window.DUCK
           || e.key === window.ATTACK)
          && this.keys.indexOf(e.key) === -1) { //if key is one of specified controls and is not in
keys array
            //add key in matched variable to keys array
            this.keys.push(e.key);
         }
         else if (e.key === window.DEBUGMODE)
            this.game.testMode = !this.game.testMode; //toggle game in and out of testing mode
         else if (e.key === window.PAUSE && !this.game.player.dead) //if game is not paused or
game over, you can pause
            this.game.togglePause();
         else if (e.key === window.PAUSE && this.game.gameOver)  //if pause button is pressed
and game is over, restart game?
            this.game.showInitMenu();
      });
      //remove key pressed from keys array on keyup event
      window.addEventListener("keyup", e => {
         if ((e.key === window.JUMP
           || e.key === window.DUCK
           || e.key === window.ATTACK
           || e.key === window.PAUSE)) {
            //splice method to remove key released from keys array
            //splice(i, num) takes index i of element to be removed and how many elements to
remove; i.e. splice(2, 3) would remove 3 elements starting at index 2
            this.keys.splice(this.keys.indexOf(e.key), 1);
         }
      });
   }
}
```

obstacle.js

```
class Obstacle {
   constructor() {
      //properties that may be needed if objects are animated
      //this.animFrame = 0    //animate obstacle/enemy sprite
      //measures for calculating delta time
      this.fps = 10;
      this.animInterval = 1000/this.fps;
      this.frameTimer = 0;
      this.offScreen = false;
   }
   update(dt) {
      this.speedX = this.game.speed;
      this.x -= this.speedX;
      this.y += this.speedY;
      //same handler as player for movement/animation fps
      if (this.frameTimer > this.animInterval) {
         this.frameTimer = 0;
         /*if(this.animFrame < this.maxFrame)
            this.animFrame++;
            else
               this.animFrame = 0; */
      }
      else {
         this.frameTimer += dt;
      }
      //check if obstacle has moved off screen to the left
      if (this.x + this.width < 0) {
         this.offScreen = true;  //obstacle is off screen and can be deleted
         this.game.score++; //test UI text; score increases when object moves offscreen
      }
   }
   draw(context) {
      //draw obstacle hitbox when test mode active
      if (this.game.testMode == true) {
         context.strokeRect(this.x, this.y, this.width, this.height);
      }
      context.fillStyle = 'red';
      //context.fillRect(this.x, this.y, this.width, this.height);
      //context.drawImage(this.image, this.x, this.y)
   }
}
```

```
export class JumpObstacle extends Obstacle {
   constructor(game) {
      super();
      this.game = game;
      this.type = "Jump";
      this.width = 64; //will need to be updated based on final sprite size
      this.height = 64;
      this.x = this.game.width;
      this.y = this.game.height - this.height - this.game.groundHeight - 10;
      this.image = document.getElementById("jumpObs");
      this.speedX = 0;
      this.speedY = 0;
      //this.maxFrame = 0; //in case this obstacle is animated
   }
   draw(context) {
      super.draw(context);
      //context.fillStyle = 'red';
      context.drawImage(this.image, this.x, this.y);
   }
}

export class AttackObstacle extends Obstacle {
   constructor(game) {
      super();
      this.game = game;
      this.type = "Attack";
      this.width = 62;
      this.height = 130;
      this.x = this.game.width;
      this.y = this.game.height - this.height - this.game.groundHeight - 10;
      this.image = document.getElementById("attackObs");
      this.speedX = 0;
      this.speedY = 0;
      //this.maxFrame = 0;    //in case this obstacle is animated
   }
   draw(context) {
      super.draw(context);
      //context.fillStyle = 'green';
      //context.fillRect(this.x, this.y, this.width, this.height);
      context.drawImage(this.image, this.x, this.y);
   }
}

export class DuckObstacle extends Obstacle {
```

```
  constructor(game) {
    super();
    this.game = game;
    this.type = "Duck";
    this.width = 64; //will be based on whatever sprite size is (hitbox will be separate)
    this.height = 64;
    this.x = this.game.width; //will need to start off screen
    this.y = this.game.height - 220; //since player MUST duck this obstacle, should be about the
height of standing sprite but too tall to jump over
    this.speedX = 0; //speed of movement on x-axis
    this.speedY = 0; //speed of movement on y-axis (maybe not needed unless we're doing
some sine wave type movement or something)
    //this.maxFrame; //sets up total number of frames on obstacle's spritesheet
    this.image = document.getElementById("duckObs");
  }
  update(dt) {
    super.update(dt); //call parent class update method
    //any extra behaviors for this particular child class can be called here
  }
  draw(context) {
    super.draw(context);
    //context.fillStyle = 'blue';
    //context.fillRect(this.x, this.y, this.width, this.height);
    context.drawImage(this.image, this.x, this.y);
  }
}
```

player.js

```
import { Running, Jumping, Falling, Ducking, Attacking, Lose } from './states.js';

//class for functions relating to player
export class Player {
  //updated constructor method?
  constructor(game, character) {
    this.game = game;
    this.width = 128; //width and height based on pixel sheet; base size is 32x32 but can be
scaled up in aseprite
    this.height = 128;
    this.ground = this.game.height - this.height - this.game.groundHeight; //variable to store
"ground" plane that player will stand on
    this.x = 100; //screen position x
    this.y = this.ground; //sets current y to ground
    //hitbox variables
    this.hitWidth = this.width * 0.3;
```

```
        this.hitHeight = this.height * 0.6;
        this.hitX = this.x + this.hitWidth;
        this.hitY = this.y + 30;
        this.duckYOffset = 20;
        this.jumpYOffset = -5;
        this.velY = 0; //velocity for jump
        this.gravity = 0.92; //counterbalance to velY variable; will allow character to fall after
reaching peak of jump
        //add in spritesheet info here
        this.character = character;
        this.stateImage = "Static";
        this.animFrame = 0; //tracks current frame in sprite animation
        //this.image = document.getElementById("dino" + this.stateImage + this.animFrame);
        this.maxFrame; //tracks total number of frames in each animation
        this.fps = 10;
        this.animInterval = 1000/this.fps;
        this.frameTimer = 0;
        this.attackTimer = 0;   //timer for how long attack action is active
        this.loseTimer = 0;     //timer for lose animation, will be replaced later with just frame
management from lose state
        this.dead = false;
        //this.image = document.getElementById("dinoStationary0");
        //state management
        this.states = [new Running(this.game), new Jumping(this.game), new Falling(this.game),
new Ducking(this.game), new Attacking(this.game), new Lose(this.game)];
        //console.log(this.ground);
    }
    //draw method for player sprite
    draw(context) {
        //draw hitbox when debug mode is on
        if (this.game.testMode == true) {
            context.strokeRect(this.hitX, this.hitY, this.hitWidth, this.hitHeight);
        }
        this.image = document.getElementById(this.character + this.stateImage + this.animFrame);
        //console.log(this.image);
        context.drawImage(this.image, this.x, this.y);
    }
    update(input, dt) {
        if (!this.dead)
        //check for collisions
        this.checkCollision();
        //call handleInput function in state manager
        this.currentState.handleInput(input);
        //updating player actions based on received input
```

```
        this.y += this.velY; //update player y position based on velY variable
        this.hitY += this.velY; //update hitbox y pos
        if (!this.grounded())
            this.velY += this.gravity; //adds gravity to velY to make player fall after jumping
        else {
            this.velY = 0; //should prevent player from falling past ground plane
            this.y = this.ground;
        }
        //fps management
        if (this.frameTimer > this.animInterval) {
            this.frameTimer = 0;
            //change animation frame number
            if (this.animFrame < this.maxFrame) {
                this.animFrame++;
            }
            else {
                this.animFrame = 0;
            }
        }
        else {
            this.frameTimer += dt;
        }

        //attack timer
        if (this.currentState === this.states[4] && this.attackTimer > 0) { //if player is in attacking
state and attack timer is not at zero
            this.attackTimer -= dt;
        }
        if (this.currentState === this.states[5] && this.loseTimer > 0) //replace later with frame
management from states
            this.loseTimer -= dt;
    }
    grounded() {
        //will return true or false; true if player is on ground, false if player is not on ground
        return this.y >= this.ground;
    }
    //switch between player states
    setState(state, speed) {
        //set current state
        this.currentState = this.states[state];
        //set game speed for current state (slightly faster during attack, normal during run)
        this.game.speed = this.game.maxSpeed * speed;
        //call enter method for passed state
        this.currentState.enter();
```

```
    }
    checkCollision() {
        this.game.obstacles.forEach(obstacle => {
            //STANDING COLLISION CHECK
            if ( //checks for collision between player hitbox and obstacle (may need refactoring for
obstacle hitbox later)
                obstacle.x < this.hitX + this.hitWidth &&
                obstacle.x + obstacle.width > this.hitX &&
                obstacle.y < this.hitY + this.hitHeight &&
                obstacle.y + obstacle.height > this.hitY
            ) {
                //obstacle.offScreen = true; //always remove obstacle on collision? may change that
later
                //if player is in attacking state and object is a wall, destroy obstacle
                if (this.currentState === this.states[4] && obstacle.type === "Attack") {
                    obstacle.offScreen = true;  //destroy obstacle
                    this.game.score++;
                }
                else {
                    obstacle.offScreen = true;
                    this.setState(5, 0);
                }

            }
        });
    }
    reset() {
        this.y = this.ground;   //reset so player starts out on the ground again
        this.hitY = this.y + 30;    //reset player hitbox position
        this.dead = false;
        this.setState(0, 1);
    }
}
```

states.js

```
import { Dust, Burst } from './effects.js';

//enum to track player states for readability and better control of spritesheet animation
const playerStates = {
    // STILL: 0,
    RUNNING: 0,
    JUMPING: 1,
    FALLING: 2,
    DUCKING: 3,
```

```
    ATTACKING: 4,
    LOSE: 5,
    RESTART: 6
}

class State {
    constructor(state, game) {
        this.state = state;
        this.game = game;
    }
}

// export class Still extends State {
//     constructor(player) {
//         super('STILL');
//         this.player = player
//     }
//     enter() {
//         this.player.stateImage = "Stationary";
//         this.player.animFrame = 0;
//         this.player.maxFrame = 0;
//         this.player.image = document.getElementById("dino" + this.player.stateImage +
this.player.animFrame);
//     }
//     handleInput(input) {
//         if (input.includes(window.JUMP) || input.includes(window.DUCK) ||
input.includes(window.ATTACK)) {
//             this.player.setState(playerStates.RUNNING);
//         }
//     }
// }

export class Running extends State {
    constructor(game) {
        //call constructor for parent class State
        super('RUNNING', game);
    }
    enter() {
        this.game.player.stateImage = "Run"
        this.game.player.animFrame = 0;
        this.game.player.maxFrame = 3;
        this.game.player.image = document.getElementById(this.game.player.character +
this.game.player.stateImage + this.game.player.animFrame);
        //reset hitbox radius
```

```
        this.game.player.hitY = this.game.player.y + 30;
    }
    handleInput(input) {
        //add dust particles while player is running
        this.game.particles.unshift(new Dust(this.game, this.game.player.x +
this.game.player.width/2 - 15, this.game.player.y + this.game.player.height - 20));

        if (input.includes(window.JUMP)) {
            this.game.player.setState(playerStates.JUMPING, 1);
        }
        else if (input.includes(window.DUCK)) {
            this.game.player.setState(playerStates.DUCKING, 1);
        }
        else if (input.includes(window.ATTACK)) {
            this.game.player.setState(playerStates.ATTACKING, 1.6);
        }
    }
}

export class Jumping extends State {
    constructor(game) {
        //call constructor for parent class State
        super('JUMPING', game);
    }
    enter() {
        //will need to change jump image later
        if (this.game.player.grounded())
            this.game.player.velY -= 21.5;
        this.game.player.stateImage = "Jump"
        this.game.player.animFrame = 0;
        this.game.player.maxFrame = 0;
        this.game.player.image = document.getElementById(this.game.player.character +
this.game.player.stateImage + this.game.player.animFrame);
        //move hitbox up just a bit
        this.game.player.hitY = this.game.player.hitY + this.game.player.jumpYOffset;
    }
    handleInput(input) {
        //logic for handling falling animation
        if (this.game.player.velY > this.game.player.gravity) { //once player hits peak of jump and
starts to fall, switches to falling state
            this.game.player.setState(playerStates.FALLING, 1);
        }
    }
}
```

```
export class Falling extends State {
   constructor(game) {
      //call constructor for parent class State
      super('FALLING', game);
   }
   enter() {
      //will need to change fall image later
      this.game.player.stateImage = "Fall"
      this.game.player.animFrame = 0;
      this.game.player.maxFrame = 0;
      this.image = document.getElementById(this.game.player.character + this.stateImage +
this.animFrame);
   }
   handleInput(input) {
      //logic for handling falling animation
      if (this.game.player.grounded()) { //changes state once player is on ground again
         this.game.player.setState(playerStates.RUNNING, 1);
      }
   }
}

//NOTE: seeing some odd -y offset with ducking state; test with character base sprites to see if
y-offset issue
//persists and if adjustment needs to be made with some kind of +y offset in enter method
export class Ducking extends State {
   constructor(game) {
      //call constructor for parent class State
      super('DUCKING', game);
   }
   enter() {
      this.game.player.stateImage = "Duck"
      this.game.player.animFrame = 0;
      this.game.player.maxFrame = 3;
      this.game.player.image = document.getElementById(this.game.player.character +
this.game.player.stateImage + this.game.player.animFrame);
      //change hitbox y coordinate on entering ducking state
      this.game.player.hitY = this.game.player.hitY + this.game.player.duckYOffset;
   }
   handleInput(input) {
      //add dust particles while player is running
      this.game.particles.unshift(new Dust(this.game, this.game.player.x +
this.game.player.width/2 - 15, this.game.player.y + this.game.player.height - 20));
```

```
        if (!input.includes(window.DUCK)) {
            this.game.player.setState(playerStates.RUNNING, 1);
        }
        // else if (input.includes(window.JUMP)) {
        //     this.game.player.setState(playerStates.JUMPING);
        // }
    }
}

export class Attacking extends State {
    constructor(game) {
        //call constructor for parent class State
        super('ATTACKING', game);
    }
    enter() { //attacking can be entered from running only for the time being
        this.game.player.stateImage = "Attack"
        this.game.player.animFrame = 0;
        this.game.player.maxFrame = 0;
        this.game.player.image = document.getElementById(this.game.player.character +
this.game.player.stateImage + this.game.player.animFrame);
        //maybe an attack timer of some sort? attack frame plays for x frames then stops?
        this.game.player.attackTimer = 500; //set attack timer to 0.5 second
    }
    handleInput(input) {
        //add dust particles while player is running
        this.game.particles.unshift(new Burst(this.game, this.game.player.x +
this.game.player.width/2 - 15, this.game.player.y + this.game.player.height - 25));

        if (this.game.player.attackTimer <= 0) { //when attack timer expires, switch back to running
state
            this.game.player.setState(playerStates.RUNNING, 1);
        }
    }
}

export class Lose extends State {
    constructor(game) {
        //call constructor for parent class State
        super('LOSE', game);
    }
    enter() {
        //GAME OVER STATE IS TRIGGERED HERE; THIS IS PROBABLY BEST PLACE TO OUTPUT SCORE
TO DATABASE
        console.log(this.game.score);
```

```
        $.ajax({
        type:"POST",
        data:{"score": this.game.score},
        url:'./scoresaver.php',
        success: function (scoresuccess) {
            console.log("successful score update");
        }
      });
        this.game.player.stateImage = "Hurt"
        this.game.player.animFrame = 0;
        this.game.player.maxFrame = 0;
        this.image = document.getElementById(this.game.player.character + this.stateImage +
this.animFrame);
        this.game.player.dead = true;
        this.game.player.loseTimer = 500;
    }
    handleInput(input) {
        //pause for a moment before displaying game over screen
        if (this.game.player.loseTimer <= 0) {
            this.game.gameOver = true;
        }
    }
}
```

UI.js

```
export class UI {
    constructor(game) {
        this.game = game;
        this.fontSize = 40;
        this.fontFamily = 'Courier New';
    }
    draw(context) {
        context.font = this.fontSize + 'px ' + this.fontFamily;
        context.textAlign = 'left';
        context.fillStyle = this.game.fontColor;
        //score text
        context.fillText('Score: ' + this.game.score, 20, 40);
        //number of players who generated same number string
        context.font = this.fontSize * 0.75 + 'px ' + this.fontFamily;
        context.fillText('Number string generated ' + window.TIMESGENNED + ' times', 20, 70);
//placeholder is where actual # will go
        //timer text (not working, dt is wonky :/)
        // context.font = this.fontSize * 0.9 + 'px ' + this.fontFamily;
```

```
        // context.fillText('Time: ' + (this.game.gameTimer/1000).toFixed(1), 20, 90);
        //game over message
        if (this.game.gameOver) {
            context.textAlign = 'center';
            context.font = this.fontSize * 3 + 'px ' + this.fontFamily;
            context.fillText('GAME OVER', this.game.width/2, this.game.height/2 - 40);
            context.font = this.fontSize + 'px ' + this.fontFamily;
            context.fillText('Press Space to restart game', this.game.width/2, this.game.height/2);
        }

        //pause screen text
        if (this.game.pause) {
            context.textAlign = 'center';
            context.font = this.fontSize * 3 + 'px ' + this.fontFamily;
            context.fillText('PAUSED', this.game.width/2, this.game.height/2 - 40);
            context.font = this.fontSize + 'px ' + this.fontFamily;
            context.fillText('Press Space to resume', this.game.width/2, this.game.height/2);

        }
    }
}
```

ScorePage.php

```
<?php session_start() ?>
<!DOCTYPE html>
<?php
include './scoreDatabaseFunctions.php';
$ranks = new scoreDatabaseFunctions();
?>

<html>
    <head>
        <meta charset="UTF-8">
        <title>Scoreboard</title>
        <link href="https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap"
rel="stylesheet">
        <link rel="stylesheet" href="./scorestyle.css" type="text/css">
    </head>
    <body>
        <!-- NAVBAR -->
        <nav class="navbar">
        <div class="navbar-container">
            <!-- home button/logo -->
            <a href="../index.php" id="home-button">Fraction Runner</a>
```

```
        <!-- other navbar items -->
        <ul class="navbar-menu">
            <li class="navbar-item">
                <!-- SESSION USAGE -->
                <!-- line below displays username, put it in the navbar -->
                <p id = "user-greet"><?php
                if (isset($_SESSION["username"])){
                  echo 'Hello, '; echo $_SESSION['username'];
                }
                ?></p>
            </li>
            <li class="navbar-item">
                <?php
                if (isset($_SESSION["username"])){
                    echo '<a class="navbar-link" href="../GameLogin/signout.php">Account</a>';
                }
                else{
                    echo '<a class="navbar-link" href="../GameLogin/signlog.php">Log In</a>';
                }
            ?>
            </li>
            <li class="navbar-item">
                <a class="navbar-link" href="../Gamepage/game.php">Play</a>
            </li>
            <li class="navbar-item">
                <a class="navbar-link" href="../Scorepage/ScorePage.php">Scoreboard</a>
            </li>
        </ul>
    </div>
</nav>

    <!-- PERSONAL SCOREBOARD -->
    <div class="user-score-container">
        <table class="scoretable">
            <thead>
                <tr id="table-title1"><th>Your Rank</th></tr>
                <tr>
                    <th>Rank</th>
                    <th>Name</th>
                    <th>Points</th>
                    <th>Repeating String</th>
                </tr>
            </thead>
```

```
            <tbody id = "scoreboard">
            <?php
            if (isset($_SESSION["username"])){
               try{
                  $loggedranking = $ranks->userRankingTable($_SESSION["username"]);
                  if (is_string($loggedranking)){
                     echo $loggedranking;
                  }
                  else if (is_null($loggedranking)){
                     echo "irrecoverable error";
                  }
                  else if ($loggedranking->num_rows > 0) {
                     // output data of each row
                     $i = 1;
                     // do not swap the order of the checks in the while statement
                     while($i <= 100 && $row = mysqli_fetch_array($loggedranking)) {
                        echo
"<tr><td>".$row["score_rank"]."</td><td>".$row["user_name"]."</td><td>".$row["user_score"
]."</td><td>".$row["digits"]."</td></tr>";
                        $i++;
                     }
                  } else {echo "user does not exist";}
               } catch(mysqli_sql_exception $e){
               echo $e;
               }
            }
            else{echo "<tr><td>user is not logged in</td></tr>";}
            ?>
            </tbody>
         </table>
      </div>

      <!-- ALL USERS SCOREBOARD -->
      <div class="all-scores-container">
         <table class="scoretable">
            <thead>
               <tr id="table-title2"><th>Extended Scoreboard</th></tr>
               <tr>
                  <th>Rank</th>
                  <th>Name</th>
                  <th>Points</th>
                  <th>Repeating String</th>
               </tr>
            </thead>
```

```
            <tbody id = "scoreboard">
            <?php
            $userranking = $ranks->rankingTable();
            if ($userranking->num_rows > 0) {
               // output data of each row
               $i = 1;
               // do not swap the order of the checks in the while statement
               while($i <= 100 && $row = mysqli_fetch_array($userranking)) {
                  echo
"<tr><td>".$row["score_rank"]."</td><td>".$row["user_name"]."</td><td>".$row["user_score"
]."</td><td>".$row["digits"]."</td></tr>";
                  $i++;
               }
            } else {echo "0 results";}
            ?>
            </tbody>
         </table>
      </div>
   <!-- <div>
      <?php
      if (isset($_SESSION["username"])){
         echo '<a href="../GameLogin/signout.php"><button>Account Maintenance and Log
Out</button></a>';
      }
      else{
         echo '<a href="../GameLogin/signlog.php"><button>Log In and Sign Up</button></a>';
      }
      ?>
      <div>
         <a href="../index.php"><button>Front Page</button></a>
      </div>
      <div>
         <a href="../Gamepage/game.php"><button>Play Game</button></a>
      </div>
   </div> -->
   </body>
</html>
```

scorestyles.css

```
/* COLOR PALETTE */
/* BG: 131313*/
/* NAVBAR: 000000*/
/* TEXT: fff4e0*/
/* BUTTONS: 809b4c*/
```

```
/* ACCENTS/LINK: 449489*/
/* LINK AFTER: 285763 */

:root {
   --bg-main: #131313;
   --bg-main-2: #303030;
   --nav-bg: #000000;
   --text-main: #fff4e0;
   --btn-main: #809b4c;
   --accent: #449489;
   --link-aft: #285763;
   --errortxt: red;
}

* {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
   font-size: 18px;
   font-family: sans-serif;
   font-size: 18px;
   color: var(--text-main);
   text-align: center;
}

body {
   background-color: var(--bg-main);
}

a {
   color: var(--accent);
   text-decoration: none;
   transition: all 0.3s ease;
}

a:hover {
   color: var(--text-main);
}

.navbar {
   background: var(--nav-bg);
   height: 70px;
   display: flex;
   justify-content: center;
```

```
    align-items: center;
    font-size: 1.2rem;
    position: sticky;
    top: 0;
    /* z index high so that it is always visible */
    z-index: 999;
}

.navbar-container {
    display: flex;
    justify-content: space-between;
    height: 90px;
    z-index: 1;
    width: 100%;
    margin: 0 auto;
    padding: 0 20px;
}

#home-button {  /*add text shadow later*/
    font-family: 'Press Start 2P', sans-serif;
    background-color: var(--btn-main);
    background-image: linear-gradient(to bottom, var(--text-main) 0%, var(--bg-main) 100%);
    background-size: 100%;
    background-clip: text;
    -webkit-background-clip: text;
    -moz-background-clip: text;
    -webkit-text-fill-color: transparent;
    -moz-text-fill-color: transparent;
    display: flex;
    align-items: center;
    cursor: pointer;
    text-decoration: none;
    font-size: 1.4rem;
    width: 200px;
}

.navbar-menu {
    width: 40%;
    display: flex;
    align-items: center;
    list-style: none;
    justify-content: space-evenly;
}
```

```
.navbar-item {
   display: flex;
   width: 100px;
   font-size: 1rem;
   margin: 0 10px;
   justify-content: space-evenly;
}

.navbar-link {
   color: var(--accent);
   display: flex;
   align-items: center;
   justify-content: space-evenly;
   text-decoration: none;
   height: 100%;
   transition: all 0.3s ease;
   font-size: 1rem;
}

.navbar-link:hover {
   color: var(--text-main);
   transition: all 0.3s ease;
}

#user-greet {
   font-size: 1rem;
}

.container {
   max-width: 800px;
   margin: 0 auto;
   padding: 20px;
}

.user-score-container, .all-scores-container {
   display: flex;
   flex-direction: row;
   justify-content: center;
}

.scoretable {
   border-collapse: collapse;
   margin: 25px 0;
   font-size: 0.9em;
```

```
    font-family: sans-serif;
    min-width: 400px;
    /* box-shadow: 0 0 20px rgba(0, 0, 0, 0.15); */
    display: grid;
    border: none;
}

#table-title1, #table-title2 {
    display: grid;
    grid-template-columns: 1fr;
    /* column-span: all; */
}

.scoretable thead tr {
    background-color: var(--nav-bg);
    color: var(--text-main);
    text-align: left;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr;
}

.scoretable th, html .scoretable td {
    padding: 12px 15px;
}

.scoretable tbody tr {
    /* border-bottom: 1px solid midnightblue; */
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr;
}

.scoretable tbody tr:nth-of-type(even) {
    background-color: var(--bg-main);
}

.scoretable tbody tr:nth-of-type(odd) {
    background-color: var(--bg-main-2);
}

.scoretable tbody tr:last-of-type {
    border-bottom: 2px solid var(--link-aft);
}

button {
```

```
    background-color: green;
    color: white;
    font-size: 16px;
    padding: 10px 20px;
    border-radius: 4px;
    border: none;
    cursor: pointer;
}

button:hover {
    background-color: darkgreen;
}

footer p {
    font-size: 0.8rem;
}
```

scoreDatabaseFunctions.php

```
<?php
##############################################################################
#########################################
## This class offers all functions for the user database and fraction database, despite the name.
oops.
## Common user error checking can be accomplished by checking the types of dynamically
##############################################################################
#########################################
class scoreDatabaseFunctions
{
    //constructor for a ranking board
    public $ranking;
    public $dbconn;
    //constructor for a ranking board, sets values of $this->dbconn and $this->ranking, for use in
scoreboards
    function __construct(){
        $this->makeConnection();
        //$this->rankingTable();
    }

    //Connects to the database
    function makeConnection(){
        //This sets the connection up, also has the password included
        $servername = "127.0.0.1";
        $sqlusername = "fractio3_user";
        $sqlpassword = "edcvfr43edcvfr4";
```

```
        $dbname = "fractio3_dba";
        //these variables are for status reporting
        $connectionstatus = "Connection not attempted.";
        $connectbool = false;
        mysqli_report(MYSQLI_REPORT_STRICT | MYSQLI_REPORT_ALL);
        try {
            // Create connection, username and pw here are for the sql server
            $this->dbconn = mysqli_connect($servername, $sqlusername, $sqlpassword, $dbname);
            // Check connection and report errors
            error_reporting(E_ALL);
            mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
            $connectionstatus = "Connection Successful.";
            $connectbool = true;
        } catch (mysqli_sql_exception $e){
            $connectionstatus = "Connection to server failed";
            echo $connectionstatus;
        }

    }

    //function to make an initial ranking table.
    function rankingTable(){
        return mysqli_query($this->dbconn, "SELECT users.user_name, users.user_score,
users.digits, count(t2.user_name) score_rank
                FROM users
                LEFT JOIN users t2 ON t2.user_score >= users.user_score
                GROUP BY user_name, user_score, digits
                ORDER BY score_rank;");
    }
    //function to make ranking table for current singular user
    function userRankingTable(string $rankedusername){
        return mysqli_query($this->dbconn, "SELECT * from
                (SELECT users.user_name, users.user_score, users.digits, count(t2.user_name)
score_rank
                FROM users
                LEFT JOIN users t2 ON t2.user_score >= users.user_score
                GROUP BY user_name, user_score, digits
                ORDER BY score_rank) AS ranksubalias
                WHERE user_name = ('$rankedusername');");
    }
    //function to retrieve pre-existing digits strings, it returns a string as a status note, and
changes public variables
    function retrieveDigits(mysqli $dbconn, $digits){
        //This is basic security to prevent code injection
```

```
    $digits = mysqli_real_escape_string($dbconn, $digits);

    //we turn currentDigits into a mysqli_query that the information can be pulled from
    $currentDigits = mysqli_query($dbconn, "SELECT *
          FROM fractio3_dba.fractions
          WHERE digits = ('$digits');");
    //return status code or score
    if($currentDigits->num_rows < 1){
       return " This is the first time these digits have been generated. ";
    }
    else{
       $incrementOne = digitsUpdate();
       return $currentDigits;
    }
    return "ERROR: Incorrect database permissions or disconnection.";
  }


  //function to retrieve a specific user's score, it returns a string as a status note, and changes
public variables
  //check if is_string (Not !is_string) for error message
  function retrieveUserScore(mysqli $dbconn, $username){
    //this is security to prevent code injection
    $username = mysqli_real_escape_string($dbconn, $username);
    //we turn currentScore into a mysqli_query that the information can be pulled from
    $currentScore = mysqli_query($dbconn, "SELECT user_score
          FROM fractio3_dba.users
          WHERE user_name = ('$username');");
    //return status codes
    if($currentScore->num_rows < 1){
       return " User does not exist. ";
    }
    else{
       return $currentScore;
    }
  }

  //function to add a new user, dbconn must be mysqli, Additionally, error checking for
pre-existing user is
  //carried out by checking if !is_string($this->addNewUser)
  function addNewUser(mysqli $dbconn, string $newname, string $newpass){
    //this is security to prevent code injection
    $newname = mysqli_real_escape_string($dbconn, $newname);
    $newpass = mysqli_real_escape_string($dbconn, $newpass);
```

```
    $userChecker = mysqli_query($dbconn, "SELECT * FROM fractio3_dba.users WHERE
user_name = ('$newname')");
    if($newname == "" || $newpass == "" || $userChecker->num_rows > 0){
      return $userChecker;
    }
    else{
      $newuser = mysqli_query($dbconn, "INSERT INTO fractio3_dba.users VALUES
(0,('$newname'), 0, ('$newpass'), '0')");
      return "Successful new user insertion.";
    }
  }

  function addNewDigits(mysqli $dbconn, string $newDigits, float $newFrac, int $newDivisor){
    //this is security to prevent code injection
    $newDigits = mysqli_real_escape_string($dbconn, $newDigits);
    if ((int)$newDigits > 999999999 || (int)$newDigits < 0){
      return "digits out of bounds";
    }
    if ($newFrac > 1 || $newFrac < 0){
      return "Fraction out of bounds, math implemented incorrectly.";
    }
    $digitsChecker = mysqli_query($dbconn, "SELECT * FROM fractio3_dba.fractions WHERE
digits = ('$newDigits')");
    if($digitsChecker->num_rows > 0){
      $incrementor = mysqli_query($dbconn, "UPDATE fractio3_dba.fractions
        SET times_generated = (times_generated + 1)
        WHERE digits = ('$newDigits');");
      $returnTimesGenerated = mysqli_query($dbconn, "SELECT fractions.times_generated
FROM fractio3_dba.fractions
        WHERE digits = ('$newDigits');");
      $row = mysqli_fetch_array($returnTimesGenerated);
      return (int)($row["times_generated"]-1);
    }
    else{
      $newDigits = mysqli_query($dbconn, "INSERT INTO fractio3_dba.fractions VALUES
(('$newDigits'), 1, $newFrac, $newDivisor)");
      return "0";
    }
  }

  //Function to change user score and most recent input digits, digits should START as an int,
and then be cast to
  //A string before being put into this.
  function setUserScore(mysqli $dbconn, string $name, int $userscore, string $digits){
```

```
    $name = mysqli_real_escape_string($dbconn, $name);
    $digits = mysqli_real_escape_string($dbconn, $digits);
    $scoreCheck = mysqli_query($dbconn, "SELECT user_score
            FROM fractio3_dba.users
            WHERE user_name = ('$name');");
    //return status codes
    if($scoreCheck->num_rows < 1){
        return " User does not exist. ";
    }
    $scoreCheck = mysqli_query($dbconn, "SELECT user_score
            FROM fractio3_dba.users
            WHERE user_name = ('$name') AND ('$userscore') > user_score;");
    if($scoreCheck->num_rows < 1){
        return " Not higher than your current highest score, too bad! ";
    }
    $scoreUpdate = mysqli_query($dbconn, "UPDATE fractio3_dba.users
    SET user_score = ('$userscore'),
        digits = ('$digits')
        WHERE user_name = ('$name') AND ('$userscore') > user_score;");
    return $userscore;
}

//function to change user's password
function changePass(mysqli $dbconn, string $name, string $oldpass, string $newpass){
    //input sanitization
    $name = mysqli_real_escape_string($dbconn, $name);
    $oldpass = mysqli_real_escape_string($dbconn, $oldpass);
    $newpass = mysqli_real_escape_string($dbconn, $newpass);
    //check for user existence
    $passfinder = mysqli_query($dbconn, "SELECT user_score
            FROM fractio3_dba.users
            WHERE user_name = ('$name');");
    //return status codes
    if($passfinder->num_rows < 1){
        return " User does not exist. ";
    }
    //check for old password being correct
    $passfinder = mysqli_query($dbconn, "SELECT user_score
            FROM fractio3_dba.users
            WHERE user_name = ('$name') AND password = ('$oldpass');");
    if($passfinder->num_rows < 1){
        return " Incorrect Password ";
    }
    //check if it's the same password
```

```
    $passfinder2 = mysqli_query($dbconn, "SELECT user_score
          FROM fractio3_dba.users
          WHERE user_name = ('$name') AND password = ('$newpass');");
    if($passfinder2->num_rows > 0){
      return " This is already your Password ";
    }
    //finally, change the password
    $passfinder = mysqli_query($dbconn, "UPDATE fractio3_dba.users
    SET password = ('$newpass')
    WHERE user_name = ('$name') AND password = ('$oldpass');");
    return $passfinder;
  }

  //Function to delete a user
  function deleteUser(mysqli $dbconn, $name, $pass){
    //First query to find the entry password for this user and check for correct permissions
    $name = mysqli_real_escape_string($dbconn, $name);
    $pass = mysqli_real_escape_string($dbconn, $pass);
    $passfinder = mysqli_query($dbconn, "SELECT password
      FROM fractio3_dba.users
      WHERE user_name=('$name')
      ");
    if ($passfinder->num_rows == 0){
      return $passfinder;
    }
    $row = mysqli_fetch_array($passfinder);
    if ($row["password"] != $pass) {
      return $row;
    }
    //the second block is to find out if the user exists with the if statement, then check if the
password is correct
    mysqli_query($dbconn, "DELETE FROM fractio3_dba.users
      WHERE user_name = ('$name') AND password = ('$pass');");
    return "Successfully deleted";
  }

  //checks user against login database
  function logIn(mysqli $dbconn, $name, $pass){
    $name = mysqli_real_escape_string($dbconn, $name);
    $pass = mysqli_real_escape_string($dbconn, $pass);
     //check for user existence
    $passfinder = mysqli_query($dbconn, "SELECT user_score
          FROM fractio3_dba.users
          WHERE user_name = ('$name');");
```

```
        //return status codes
        if($passfinder->num_rows < 1){
            return " User does not exist. ";
        }
        //check for password being correct
        $passfinder = mysqli_query($dbconn, "SELECT user_score
                FROM fractio3_dba.users
                WHERE user_name = ('$name') AND password = ('$pass');");
        if($passfinder->num_rows < 1){
            return " Incorrect Password ";
        }
        else{
            return $passfinder;
        }
    }

}
```

DATABASE

Database Setup:

```
#initial creation of database, drop is delete in this case, use states we're using it as the base
database going forwards
DROP DATABASE IF EXISTS `scoreboard_dba`;
CREATE DATABASE `scoreboard_dba`;
USE `scoreboard_dba`;

#character sets
SET NAMES utf8mb4 ;
SET character_set_client = utf8mb4 ;

#creation of an actual table within the database, users is the database name
CREATE TABLE `users` (
#Variable/column name/ids and rules
#NOT NULL
  `user_id` int NOT NULL AUTO_INCREMENT,
  `user_name` varchar(50) NOT NULL,
  `user_score` bigint,
  `password` varchar(50) NOT NULL,
  `digits` varchar(9),
  #`time_set` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`user_id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `fractions` (
#Variable/column name/ids and rules
#NOT NULL
  `digits` varchar(9),
  `fraction` decimal(10,9) CHECK(fraction>0) CHECK(fraction<1),
  `divisor` int,
  PRIMARY KEY (`digits`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

Database Trigger Code:

```
SELECT * FROM scoreboard_dba.users;

DELIMITER $$

CREATE TRIGGER trigger1
BEFORE INSERT
ON users
FOR EACH ROW
BEGIN
  SELECT COUNT(*) INTO @count FROM users;
  IF @count >= 10000 THEN
    DELETE FROM users
    WHERE user_rank = (SELECT min(user_rank) FROM users);
  END IF;
END
$$

DELIMITER ;
```

Appendix B: User Manual

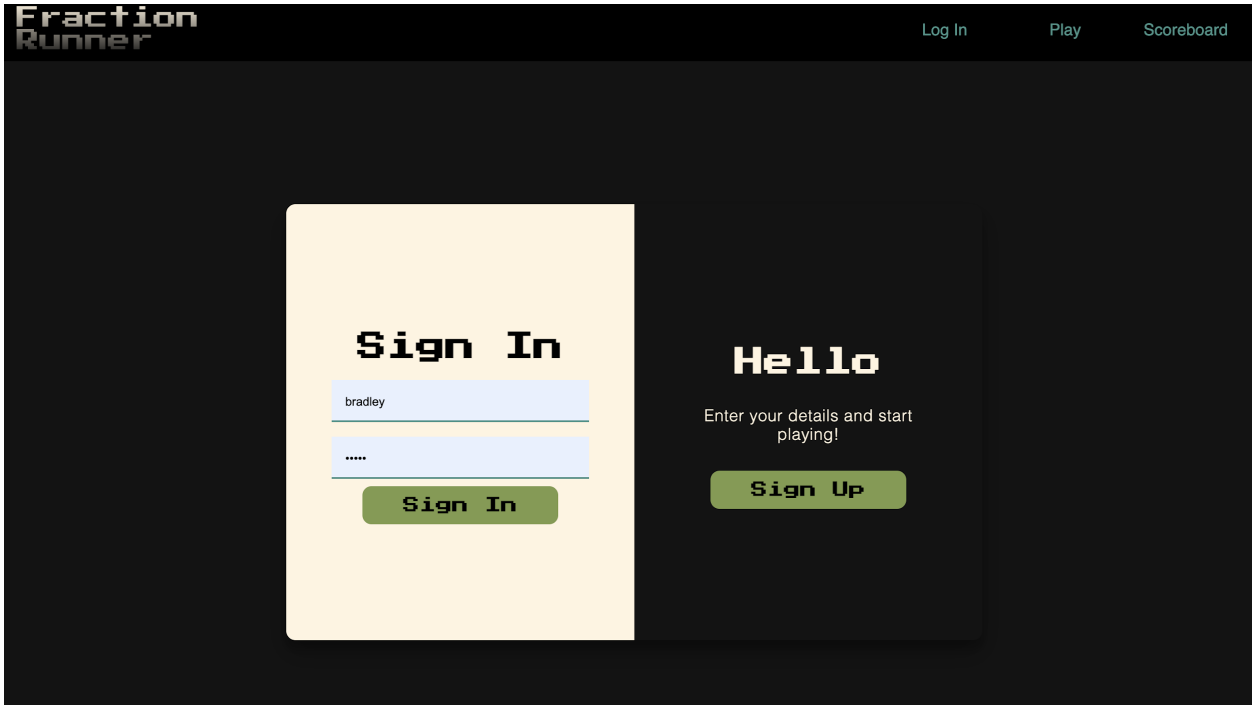
User Manual

Welcome to Fraction Runner, an educational running game!

In this game, the object is to see how long you can stay alive.
Upon start, the player is asked to write a number.
The game puts player's number in a fraction as the numerator over the same number of 9s in denominator.
(Example: 443 becomes 443/999)
The division produces a string of repeating decimals which will be displayed as the ground.
(From our earlier example: 443/999 becomes 0.443443443...)
The player runs on top of the numbers using Jump, Duck, or Attack to stay alive.
The player that stays alive the longest time is the winner.
When running, check out the repeating decimals.

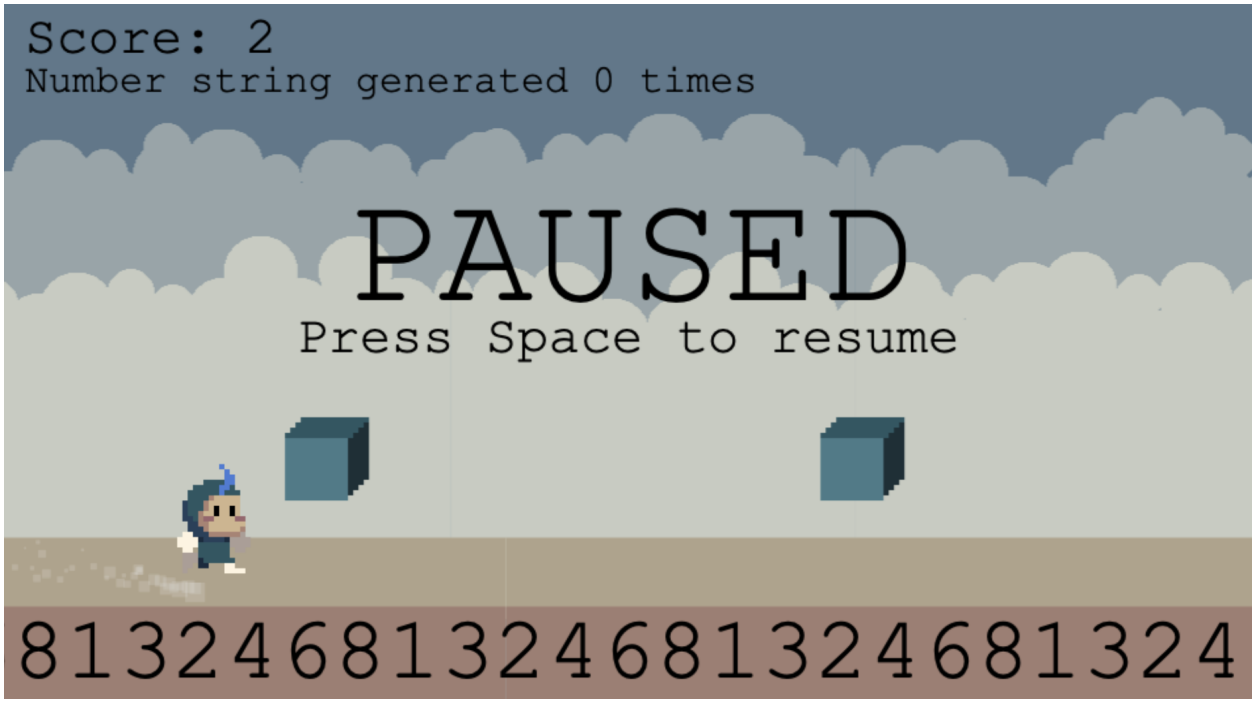
On the Home Screen you will see three choices:
Fraction Runner – select when you are ready to begin the game
Introduction Page – click to learn more about the game and its developers
Top 100 Scoreboard – pick this to see who has the high score

Player will log in on the Log in Page:



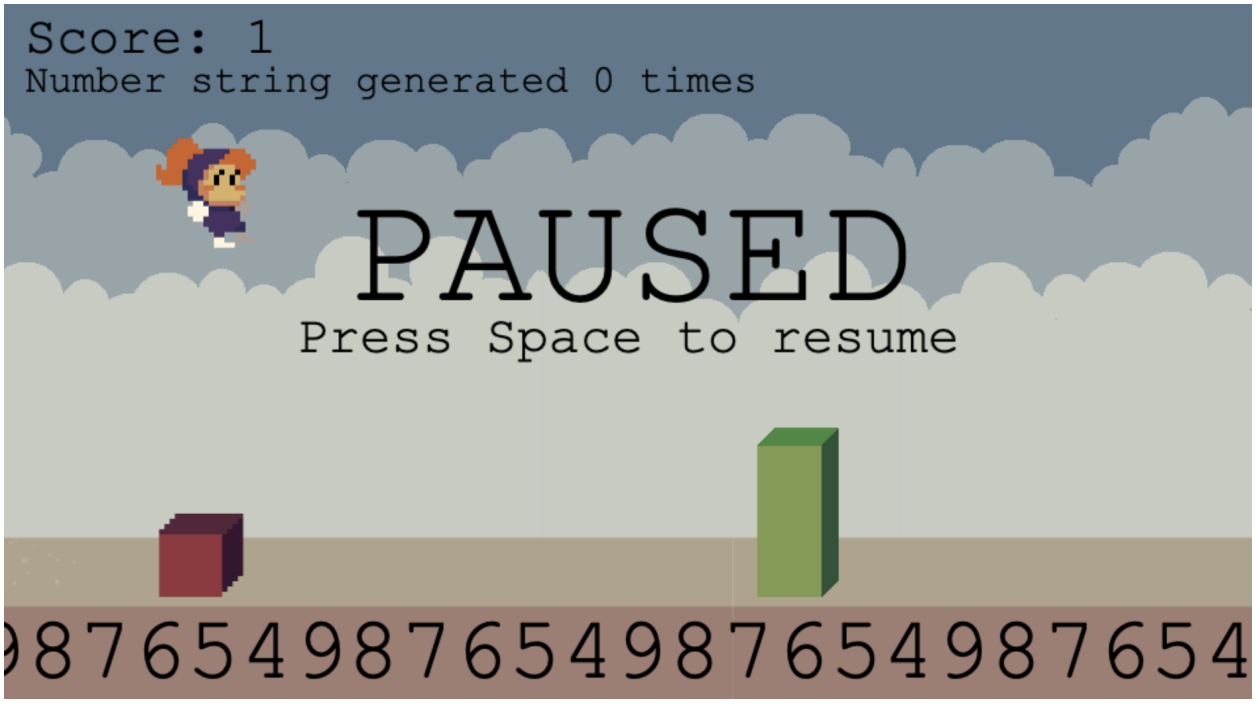
Controls:
Each character can perform three different actions:
*Jump – press the up arrow
*Attack – hit the attack button
*Duck – press the down arrow

(Screenshot of character needing to duck)



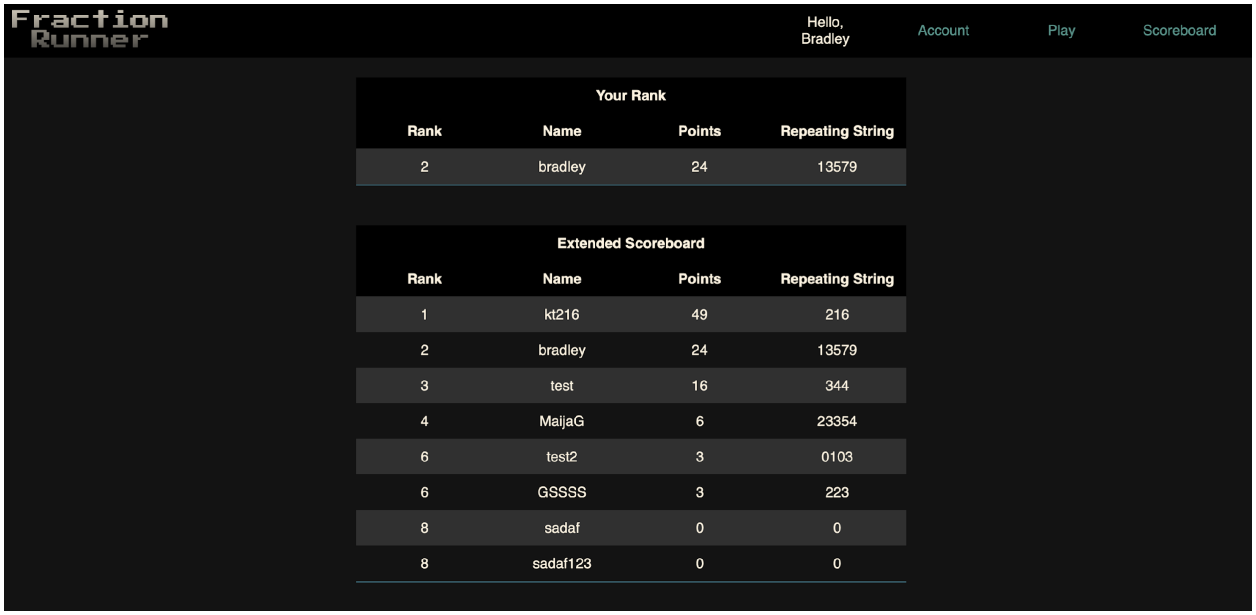
Gameplay:
At the start, the player is asked to select a character.
There is no skill difference between the characters.
When starting the game, the player must pick a number up to 999,999,999.
After entering the number, the running game begins.

(Screenshot of character jumping over an obstacle)



The object of the game is to see how long the player can stay alive.
The timer starts at the beginning of the game.
Once the player has been hit by an object, the game is over.
The time is calculated and added to the Scoreboard.

Fraction Runner Scoreboard:



Your Rank

Rank	Name	Points	Repeating String
2	bradley	24	13579

Extended Scoreboard

Rank	Name	Points	Repeating String
1	kt216	49	216
2	bradley	24	13579
3	test	16	344
4	MaijaG	6	23354
6	test2	3	0103
6	GSSSS	3	223
8	sadaf	0	0
8	sadaf123	0	0

Tips and Tricks:
*There are three different obstacles. Each one can only be defeated by the correct action.
*Use Jump when a red obstacle appears
*Use Duck to go under the blue block
*Use Attack to break a green wall
*The runner will speed up as time continues. Stay alert!

We Thank you for playing Fraction Runner.

Appendix C: **Test Plan**

Our test plan involves testing for the following:
*MySQL database – make sure insert, delete, update actions work from login and gamepage. Make sure Select actions work from score page and the gamepage.
*JavaScript – make sure character actions match the buttons.
*CSS – make sure the site loads correctly from the intro page. Check the loading, colors, size, etc, of the game during each step of input. Check scoreboard loads properly.
*HTML – make sure site loads and structure is intact.