# Appendices

# Source Code

## HTML Code

<!DOCTYPE html>

<html>

<head>

<title>Horse Racing</title> <!-- set the title of the html web page. display on tab -->

<link rel="stylesheet" href="game.css" /> <!-- link game.css to this html page -->

<link href="https://fonts.googleapis.com/css?family=Lobster" rel="stylesheet"> <!-- adding google fonts -->

<script src="game.js"></script> <!-- link game.js javascript file to this html -->

</head>

<body>

<!-- this is a custom div to check(compare) when the horses touches the finishline. This is use because the provided finishline is set using MARGIN-LEFT property, due to which the offsetleft of that finishline is difficult to retrieve -->

<div class="finishLine">

</div>

<!-- this div covers horse and its rider -->

<div id="horse1" class="horse standRight">

<!-- this div only covers rider -->

<div class="rider">

<!-- this div covers riders head -->

<div class="head ">

</div>

<!-- this div covers riders body -->

<div class="body">

</div>

</div>

</div>

<div id="horse2" class="horse standRight">

<div class="rider">

<div class="head">

</div>

<div class="body">

</div>

</div>

</div>

<div id="horse3" class="horse standRight">

<div class="rider">

<div class="head">

</div>

<div class="body">

</div>

</div>

</div>

<div id="horse4" class="horse standRight">

<div class="rider">

<div class="head">

</div>

<div class="body">

</div>

</div>

</div>

<div class="track">

<div id="startline">

</div>

<div class="inner">

<div id="start" class="startButton">

Start Race

</div>

<div id="changeMode" class="startButton">

Play Relay Mode

</div>

<br>

<!-- table that display relay result -->

<table id="relayResult" style="left: 45vw; top: 22vh; margin-left: 0px; padding: 10px;">

<thead>

<tr>

<th>Relay </th>

<th>Results</th>

<th></th>

</tr>

</thead>

<tr>

<td>1st</td>

<td class="horse1"></td>

<td class="horse3"></td>

</tr>

<tr>

<td>2nd</td>

<td class="horse2"></td>

<td class="horse4"></td>

</tr>

</table>

<!-- table that disply normal game rank/ result -->

<table id="results">

<thead>

<tr>

<th>Results</th>

<th></th>

</tr>

</thead>

<tr>

<td>1st</td>

<td class="horse1"></td>

</tr>

<tr>

<td>2nd</td>

<td class="horse2"></td>

</tr>

<tr>

<td>3rd</td>

<td class="horse3"></td>

</tr>

<tr>

<td>4th</td>

<td class="horse4"></td>

</tr>

</table>

<div style="display: flex; flex-direction: column; padding: 0px; ">

<div class="bet" style="margin: 10px; margin-left: 30px; padding-top: 10px; margin-top: 7vh; top: 10vh;">

<p>You currently have <span id="funds">100</span></p>

<label>Bet Amount (£)</label>

<input type="number" id="amount" />

<label>Set Lap(1-5):</label>

<select id="lap"> <!-- Let user select the lap value -->

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

</select>

<label>Bet on horse:</label>

<select id="bethorse">

<option value="horse1">White</option>

<option value="horse2">Blue</option>

<option value="horse3">Green</option>

<option value="horse4">Brown</option>

</select>

<label style="font-weight: bold; color: blue;">Relay Setting:</label>

<select id="relayTeam" disabled="">

<option value="1">Team 1 (White &amp; Green)</option>

<option value="2">Team 2 (Blue &amp; Brown)</option>

</select>

</div>

<!--

<ul id="results" style="width: 30px; background-repeat: no-repeat; background-position: 0px -64px;">

<li class="horse1" style="height: 30px;"></li>

<li class="horse2" style="height: 30px;"></li>

<li class="horse3" style="height: 30px;"></li>

<li class="horse4" style="height: 30px;"></li>

</ul> -->

<!-- table that display current odds value for each character -->

<table id="odds" style="margin-top: 7vh; margin-left: 23vw; padding-bottom: 2px; padding-top: 2px;">

<thead>

<tr>

<th>Odds</th>

<th>Chart</th>

<th></th>

</tr>

</thead>

<tr>

<td class="horse1"></td>

<td>X</td>

<td></td>

</tr>

<tr>

<td class="horse2">x</td>

<td>X</td>

<td></td>

</tr>

<tr>

<td class="horse3"></td>

<td>X</td>

<td></td>

</tr>

<tr>

<td class="horse4"></td>

<td>X</td>

<td></td>

</tr>

</table>

</div>

</div>

</div>

<!-- rule board -->

<div class="notice" style=" margin-top: 25vh;

margin-left: 88vw;

margin-bottom: 0px;

width: 9vw;

height: 60vh;

left: 2vw;

top: 1vh;

padding: 0px;

padding-left: 5px;"><label style="color: red; padding: 3px; padding-left: 10px; margin: 0px;">Odds Rule => </label> if you bet on the player with odds X 2, on winning the bet you will get 2 times the money you bet. 3 times the money on odds X 3 and so on.. <br /> You will not loose more money then your bet amount, on single game. <br/ > <br /> <br /> <font color="blue">Note: odds system doesn't apply on RELAY MODE</font></div>

<!-- sound file use on game -->

<audio id="horseSound">

<source src="sound\horse.ogg" type="audio/ogg">

</audio>

<audio id="winSound">

<source src="sound\win\_sound.ogg" type="audio/ogg">

</audio>

<audio id="loseSound">

<source src="sound\lose\_sound.ogg" type="audio/ogg">

</audio>

<audio id="gallopSound">

<source src="sound\horse\_Gallop.ogg" type="audio/ogg"> <!-- Horse Gallop Sound Effect freesound. 2017. YouTube. -->

</audio>

</body>

</html>

## CSS Code

body {background-image: url(images/bg.png); margin: 0; padding: 0; font-family: calibri, sans-serif;}

aside {display: none; width: 200px; position: fixed; height: 100vh; left: 100%; margin-left: -240px; background-color: #ddd; border-left: 5px solid #bbb; padding: 20px;}

#closeside {float: right; font-weight: bold; cursor: pointer;}

.character {position: absolute; top: 200px; left: 200px; display: block;}

.character .body {width: 32px; height: 32px; background-image: url(images/body0.png); background-position: 64px 0px; margin-top: 14px; }

.character .head {width: 32px; height: 32px; background-image: url(images/head0.png); margin-bottom: 0;position: absolute; background-repeat: no-repeat; background-position: 0px -64px;}

aside ul {list-style-type: none; margin-left: 0; padding-left: 0; margin-bottom: 20px; overflow: auto;}

.heads li, .bodies li {width: 32px; height: 32px; background-position: 0 -64px; float: left; cursor: pointer; border: 2px solid #ddd; margin-right: 4px;}

.heads li:hover, .bodies li:hover {border: 2px solid red;}

#head0 {background-image: url(images/head0.png);}

#head1 {background-image: url(images/head1.png);}

#head2 {background-image: url(images/head2.png);}

#head3 {background-image: url(images/head3.png);}

#head4 {background-image: url(images/head4.png);}

.bodies li {background-position: 64px 0px;}

#body0 {background-image: url(images/body0.png);}

#body1 {background-image: url(images/body1.png);}

#body2 {background-image: url(images/body2.png);}

#body3 {background-image: url(images/body3.png);}

#body4 {background-image: url(images/body4.png);}

.tree {float: right; margin-top: 12vh; margin-right: 2vw;}

#opponent {left: 500px; top: 500px;}

#opponent .body {background-image: url(images/body1.png);}

#opponent .head {background-image: url(images/head1.png);}

.character.standUp .head {

background-position: 0px 0px;

}

.character.standDown .head {

background-position: 0px -64px;

}

.character.standLeft .head {

background-position: 0px -32px;

}

.character.standRight .head {

background-position: 0px -96px;

}

.character.standUp .body {

background-position: 0px 0px;

}

.character.standDown .body {

background-position: -64px 0px;

}

.character.standLeft .body {

background-position: -32px 0px;

}

.character.standRight .body {

background-position: -96px 0px;

}

.character.walkUp .head {

background-position: 0px 0px;

animation: headbob 0.5s infinite;

}

.character.walkUp .body {

animation: walkup 1s steps(3) infinite;

}

.character.walkDown .head {

background-position: 0px -64px;

animation: headbob 0.75s infinite;

}

.character.walkDown .body {

animation: walkdown 1s steps(3) infinite;

}

.character.walkLeft .head {

background-position: 0px -32px;

margin-top: -3px;

margin-left: -2px;

animation: headbob2 0.75s infinite;

}

.character.walkLeft .body {

animation: walkleft 1s steps(3) infinite;

}

@keyframes walkleft {

0%, 100% { background-position: -32px -32px; }

50% { background-position: -32px -128px; }

}

.character.walkRight .head {

background-position: 0px -96px;

margin-top: -3px;

margin-left: 4px;

animation: headbob3 0.75s infinite;

}

.character.walkRight .body {

animation: walkright 1s steps(3) infinite;

}

@keyframes walkright {

0%, 100% { background-position: -96px -32px; }

50% { background-position: -96px -128px; }

}

@keyframes walkup {

0%, 100% { background-position: 0px -32px; }

50% { background-position: 0px -128px; }

}

@keyframes walkdown {

0%, 100% { background-position: -64px -32px; }

50% { background-position: -64px -128px; }

}

@keyframes headbob {

0%, 100% { margin-top: 0px; }

50% { margin-top: 1px; }

}

@keyframes headbob2 {

0%, 100% { margin-left: -2px; }

50% { margin-left: 0px; }

}

@keyframes headbob3 {

0%, 100% { margin-left: 2px; }

50% { margin-left: 0px; }

}

.track {border-style: solid;

border-collapse: separate;

display: table;

border-width: 16px;

-moz-border-image: url(outer.png) 14 15 14 16 repeat;

-webkit-border-image: url(images/outer.png) 14 15 14 16 repeat;

-o-border-image: url(outer.png) 14 15 14 16 repeat;

border-image: url(images/outer.png) 14 15 14 16 repeat; background-image: url(images/mud.png);

position: absolute; height: 80vh; width: 80vw; margin: 5vw; margin-top: 5vh;

}

.track .inner {

background-image: url(images/bg.png);

border-image: url(images/inner.png); border-image-slice: 16 16 16 16 fill; border-image-width: 16px 16px 16px 16px; border-image-outset: 0px 0px 0px 0px; border-image-repeat: round round;

border-collapse: separate; margin-top: 15vh; margin-left: 10vw; display: table; solid transparent; background-color: green; width: 60vw; height: 50vh;}

.horse {background-image: url(images/horse2.png);}

.horse.runLeft {

width: 96px;

height: 60px;

display: block;

background-position: 0px -256px;

animation: horseLeft steps(1) 0.5s infinite;

}

@keyframes horseLeft {

0%, 100% {background-position: 0px -256px; }

50% {background-position: 0px -450px;}

}

.horse.runRight {

width: 96px;

height: 60px;

display: block;

background-position: 0px -320px;

animation: horseRight steps(1) 0.5s infinite;

}

@keyframes horseRight {

0%, 100% {background-position: 0px -320px; }

50% {background-position: 0px -512px;}

}

.horse.runDown {

width: 48px;

height: 64px;

background-position:48px 192px;

animation: horseDown steps(1) 0.75s infinite;

}

@keyframes horseDown {

0%, 100% {background-position:48px -192px; }

33% {background-position:48px -380px; }

66% {background-position:48px 0px;}

}

.horse.runUp {

width: 48px;

height: 64px;

background-position:0px 192px;

animation: horseUp steps(1) 0.75s infinite;

}

@keyframes horseUp {

0%, 100% {background-position:0px -192px; }

33% {background-position:0px -380px; }

66% {background-position:0px 0px;}

}

#startline {

width: 2vw;

height: 15vh;

background-color: #eee;

background-image: linear-gradient(45deg, black 25%, transparent 25%, transparent 75%, black 75%, black),

linear-gradient(-45deg, black 25%, transparent 25%, transparent 75%, black 75%, black);

background-size:1vw 1vw;

position: absolute;

margin-left: 20vw;

opacity: 1;

}

#horse1 {

position: absolute; top: 1vh; z-index: 100;

}

#horse4 .rider .head {background-image: url(images/head2.png);}

#horse4 .rider .body {background-image: url(images/body3.png);}

#horse3 .rider .head {background-image: url(images/head1.png);}

#horse3 .rider .body {background-image: url(images/body1.png);}

#horse2 .rider .head {background-image: url(images/head4.png);}

#horse2 .rider .body {background-image: url(images/body4.png);}

#horse2 {

position: absolute; top: 5vh; z-index: 200;

background-image: url(images/horse3.png);

}

#horse3 {

position: absolute; top: 9vh; z-index: 300;

background-image: url(images/horse1.png);

}

#horse4 {

position: absolute; top: 13vh; z-index: 400;

background-image: url(images/horse4.png);

}

.horse {left: 0px;}

.horse.standLeft {background-position: 0px -64px; width: 96px; height: 60px;}

.horse.standLeft .rider .body {background-position: -32px -320px;}

.horse.standLeft .rider .head {background-position: 0px -32px;}

.horse.standRight {background-position: 0px -128px; width: 96px; height: 60px;}

.horse.standRight .rider {margin-left: -15px;}

.horse.standRight .rider .body {background-position: -96px -320px;}

.horse.standRight .rider .head {background-position: 0px -96px;}

.horse .rider .head {

position: absolute;

margin-left: 40px;

margin-top: -16px;

background-image: url(images/head0.png);

width: 32px;

height: 32px;

}

.horse .rider .body {

background-image: url(images/body0.png);

width: 32px;

height: 32px;

margin-left: 40px;

margin-top: 8px;

}

.horse.runDown .rider, .horse.runRight .rider, .horse.runLeft .rider{

animation: riderBob1 0.75s infinite;

}

.horse.runLeft .rider .body {

background-position: -32px -320px;

}

.horse.runRight .rider {

margin-left: -15px;

}

.horse.runRight .rider .head {

background-position: 0px -96px;

animation: headbob4 0.5s infinite;

}

.horse.runRight .rider .body {

background-position: -96px -320px;

}

.horse.runDown, .horse.runUp {

margin-left: 32px;

}

.horse.runDown .rider {

margin-left: -32px;

}

.horse.runDown .rider .head {

background-position: 0px -64px;

animation: headbob5 0.5s infinite;

}

.horse.runDown .rider .body {

background-position: -64px -320px;

margin-top: -16px;

}

.horse.runUp .rider {

margin-left: -32px;

animation: riderBob2 0.75s infinite;

}

.horse.runUp .rider .head {

background-position: 0px 0px;

animation: headbob5 0.5s infinite;

}

.horse.runUp .rider .body {

background-position: 0px -320px;

margin-top: -16px;

}

.horse.runLeft .rider .head {

background-position: 0px -32px;

animation: headbob4 0.5s infinite;

}

@keyframes headbob4 {

0%, 100% { margin-left: 40px; }

50% { margin-left: 42px; }

}

@keyframes headbob5 {

0%, 100% { margin-top: -10px; }

50% { margin-top: -8px; }

}

@keyframes riderBob1 {

0%, 100% { margin-top: 8px; }

50% { margin-top: 10px; }

}

@keyframes riderBob2 {

0%, 100% { margin-top: 18px; }

50% { margin-top: 20px; }

}

#start, #changeMode {

float: right;

margin: 1vw;

padding: 1vw;

background: #3498db;

background-image: -webkit-linear-gradient(top, #3498db, #2980b9);

background-image: -moz-linear-gradient(top, #3498db, #2980b9);

background-image: -ms-linear-gradient(top, #3498db, #2980b9);

background-image: -o-linear-gradient(top, #3498db, #2980b9);

background-image: linear-gradient(to bottom, #3498db, #2980b9);

-webkit-border-radius: 28;

-moz-border-radius: 28;

border-radius: 28px;

font-family: Arial;

color: #ffffff;

font-size: 20px;

padding: 10px 20px 10px 20px;

text-decoration: none;

cursor: pointer;

text-shadow: 2px 2px 2px #333;

}

#start:hover , #changeMode:hover{

color: #000;

}

.bet, #results, #odds, #relayResult, .notice{

background-color: #ffefb4;

padding: 1vw;

width: 18vw;

border: 2px solid #555;

border-radius: 20px;

margin-top: 5vh;

margin-left: 2vw;

overflow: auto;

float: left;

}

#funds {font-weight: bold;}

#funds:before {content: "£";}

label, input, select {float: left; display: block; width: 8vw; margin-bottom: 0.5vh;}

label {clear: left;}

#results {width: auto; position: absolute; margin-top: 8vh; margin-left: 48vw;}

#results td {height: 30px; width: 30px; background-repeat: no-repeat; background-position: 0px -64px;}

#results .horse1 { background-image: url(images/head0.png);}

#results .horse2 { background-image: url(images/head4.png);}

#results .horse3 { background-image: url(images/head1.png);}

#results .horse4 { background-image: url(images/head2.png);}

#relayResult {width: auto; position: absolute; margin-top: 8vh; margin-left: 48vw;}

#relayResult td {height: 30px; width: 30px; background-repeat: no-repeat; background-position: 0px -64px;}

#relayResult .horse1 { background-image: url(images/head0.png);}

#relayResult .horse2 { background-image: url(images/head4.png);}

#relayResult .horse3 { background-image: url(images/head1.png);}

#relayResult .horse4 { background-image: url(images/head2.png);}

#odds {width: auto; position: absolute; margin-top: 8vh; margin-left: 48vw;}

#odds td {height: 30px; width: 30px; background-repeat: no-repeat; background-position: 0px -64px; text-align: center;}

#odds .horse1 { background-image: url(images/head0.png);}

#odds .horse2 { background-image: url(images/head4.png);}

#odds .horse3 { background-image: url(images/head1.png);}

#odds .horse4 { background-image: url(images/head2.png);}

/\* css for custom made finish line\*/

div.finishLine{

width: 1px;

height: 15vh;

color: white;

position: absolute;

top: 7vh;

left: 26vw;

z-index: 999;

}

## JAVA Script Code

/\* ::::::::::::: DECLARATION OF VARIABLES :::::::::::: \*/

var startButtonActivated = true;

var intervals = [];

var lap = 0; // number of laps that user inputs... currently manual.

var horse = ["","horse1", "horse2", "horse3", "horse4"]; // horse array variable to store horse-id, use on various functions.

// variables that stores values related with current width and height of screen

// use in movement and turing of horses.

// distance from top and left side of screen to inner part of track

var leftLaneInner = window.innerWidth \* (10/100); // stores 10% of total screen width indicates distance from left part of screen to left side of inner track

var topLaneInner = window.innerHeight \* (15/100); // 15% of total height, indicates distance from top part of screen to top side of inner track

var rightLaneInner = window.innerWidth \* (70/100); // stores 70% of total screen width, indicates distances from left side of screen to right side of inner track.

var bottomLaneInner = window.innerHeight \* (65/100);

// distance from top and left side of screen to outer part of track.

var leftLaneOuter = window.innerWidth \* (5/100); // stores 5% of total screen width, indicates distance from left side of screen to left side of outer track.

var topLaneOuter = window.innerHeight \* (5/100);

var rightLaneOuter = window.innerWidth \* (80/100);

var bottomLaneOuter = window.innerHeight \* (80/100);

var trackLength = window.innerWidth \* (75/100); // total length of track.

// variables use to, check number of lap completion.

var finishLine = 0; // distance from left side to custom created to finish line.(use of offsetLeft)

var lapFinished = 0; // varable that counts the number of lap that have been completed

var lapCounter = 0; // this lap couter value is increased by 1 each, any of the horses touches the finishline.

// variable use to declare the RANK of winners, these value are accessed only in last lap

var rankCount = 1;

var rank = []; // this array variable stores the horse name, in corresponding to array index value(example, index0 = 1st rank, index 1 = 2nd rank)

// variable for betting system

var betAmount = 0;

var balance = 0;

var selectedHorse; // variable that stores horse which the user selects

var updateAmount = 0; // amount to be updated after lap ends

var finalAmount = 0; // final amount after lap ends

// variable for user status/updates

var result = false;

// variable to store already taken odds values

var oddArray = [];

// variable to change game mode

var gameMode = 1;

// relay mode team selection variable

var selectedTeam = 0;

// variable to store team in winning order

var wonTeam = [];

// to store prompt input value

var promptInput = 0;

/\* :::::::::::::: BETTING SYSTEM :::::::::::::::: \*/

/\* :::::::::::::::::: RESULT SCREEN CHANGES ::::::::::::::: \*/

/\* this function is called when the game ends.

\* this function updates the result on screen \*/

function displayRank(){

for(var i = 0; i <=3; i++){

var rankedHorse = rank[i];

var rankTable = document.getElementById('results');

var rankHeads = rankTable.getElementsByTagName('tr')[i + 1];

var rankHead = rankHeads.getElementsByTagName('td')[1];

rankHead.className = rankedHorse;

}

}

function displayRelayResult(){

for(var i = 0; i <= 1; i++){

var wonHorses = wonTeam[i];

var resultTable = document.getElementById('relayResult');

var resultHeads = resultTable.getElementsByTagName('tr')[i + 1];

var resultHead1 = resultHeads.getElementsByTagName('td')[1];

var resultHead2 = resultHeads.getElementsByTagName('td')[2];

resultHead1.className = 'horse' + wonHorses;

resultHead2.className = 'horse' + (wonHorses + 2);

}

}

/\* :::::::::::::::::: Lap Setting ::::::::::::::::::::: \*/

function lapDisplay(){

var oldLapDisplay = document.getElementsByClassName('lapDisplay');

if(oldLapDisplay.length > 0){

console.log("value yes");

oldLapDisplay[0].parentNode.removeChild(oldLapDisplay[0]);

}

var getbody = document.getElementsByTagName('body')[0];

var lapDisplayDiv = document.createElement('div');

lapDisplayDiv.className = 'lapDisplay';

getbody.appendChild(lapDisplayDiv);

console.log('Lap is being Displayed');

/\*

\* styling on newly created element

\*/

var selectDiv = document.getElementsByClassName('lapDisplay')[0];

selectDiv.className = 'notice';

selectDiv.style.width = 10 + 'vh';

selectDiv.style.position = 'absolute';

selectDiv.style.left = 88 + 'vw';

selectDiv.style.top = 5 + 'vh';

selectDiv.style.borderStyle = 'dotted';

selectDiv.style.borderColor = 'rgba(' + randFloor(255) + ', ' + randCeil(255) + ', ' + randRange(10,245) + ', ' + 1 + ')';

selectDiv.style.fontSize = 25 + 'px';

selectDiv.style.fontFamily = 'Lobster';

selectDiv.style.padding = 10 + 'px';

selectDiv.style.textAlign = 'center';

var lapText0 = document.createTextNode('L ap');

var lapText1 = document.createTextNode((lapFinished + 1) + ' / ' + lap);

var paragraph = document.createElement('p');

paragraph.appendChild(lapText0);

paragraph.style.margin = 0 + 'px';

paragraph.style.padding = 0 + 'px';

selectDiv.appendChild(paragraph);

selectDiv.appendChild(lapText1);

}

/\* ::::::::::::::::: UPDATE THE ODDS VALUE ::::::::::::::: \*/

function changeOdds(){

for(var i = 0; i <=3; i++){

var rankedHorse = rank[i];

var oddsTable = document.getElementById('odds');

var headClass = oddsTable.getElementsByClassName(rankedHorse)[0];

var oddInsert = headClass.parentNode;

var textDiv = oddInsert.getElementsByTagName('td')[2];

textDiv.innerHTML = (i + 2); // generates new random number and set value in document

}

}

// function that checks if lap is over or not, and calls some onward functions

function checkLap(){

console.log("current GameMode = " + gameMode); // for checking purpose

// console.log(wonTeam); // for checking purpose

//console.log(lapCounter); // for checking purpose

/\* this condition is to prevent from getting the below condition to get true when lapCounter is '0'

\* without the above condition lapFinished value will increase from the beginning when garme starts. \*/

if(lapCounter > 0){

// condition to check if all the horses completes 1 lap.

if(lapCounter % 4 == 0){

lapFinished++; // when the above conditions get true then current lap count is increase by 1, ie lapFinished.

if(lapFinished < lap){

lapDisplay(); // current display on screen

}

lapCounter = 0;

}

}

//console.log(lapFinished); // for checking purpose

// condition to see if laps are finished

if(lapFinished >= lap){

console.log('Horse rank ' + rank); // for checking purpose

stopAudio(gallopSound);

wonOrLost(); // function to check if user won or lost

lapFinished=0; // reset the lap

// alert("lapOver"); // for testing purpose

if(gameMode == 1){

displayRank(); // function to change the result screen

changeOdds(); // updates the odds value

}else if(gameMode == 2){

displayRelayResult();

}

startButtonActivated = true; // enables start button

if(gameMode == 1 ){

intervals[5] = setInterval(stopAllIntervals, 2000); // creating interval that stops all interval, after 2 seconds when lap is over

}else if(gameMode == 2){

intervals[5] = setInterval(stopAllIntervals, 1);

}

}

}

/\* ::::::::::::::::::: function to check if the user wins or lose::::::::::::::: \*/

function wonOrLost(){

// console.log(rank[0]);

// console.log(selectedHorse); // for testing purpose

console.log('won team' +wonTeam);

console.log(selectedTeam);

if(gameMode == 1){

if(selectedHorse == rank[0]){

result = true;

}

}else{

if(selectedTeam == wonTeam[0]){

result = true;

}

}

announcement(); // announce the result to user

}

/\* :::::::::: function that announce the result to user ::::::::::: \*/

function announcement(){

if(result){

var winSound = document.getElementById('winSound'); // get element form HTML

playAudio(winSound); // function that plays the sound that is passed as parameter

alert('You won the bet:');

won(); // function to make update in user balance when user won

result = false; // reset result value

}else{

console.log("loss is working");

var loseSound = document.getElementById('loseSound');

playAudio(loseSound);

alert('You lose the bet:');

getBalance();

if(balance == 0){

var reset = confirm("You have 0 Amount remaining. Press OK to restart The Game.");

if(reset == true){

location.reload();

}

}

//lost(); // function to make update in user balance when user lose

}

}

/\* ::::::::::: function that make update on user balance when user wins :::::::: \*/

function won(){

if(gameMode == 1){

var oddsTable = document.getElementById('odds');

var headClass = oddsTable.getElementsByClassName(selectedHorse)[0];

var oddInsert = headClass.parentNode;

var textDiv = oddInsert.getElementsByTagName('td')[2];

var getOdds = textDiv.innerHTML;

console.log(getOdds);

updateAmount = betAmount \* getOdds;

} else{

updateAmount = betAmount \* 2;

}

finalAmount = balance + updateAmount;

setBalance(finalAmount);

}

/\* :::::::::: function that make update on user balance when user lost ::::::::: \*/

// not in use !!

function lost(){

updateAmount = betAmount;

finalAmount = balance - updateAmount;

setBalance(finalAmount);

}

/\* ::::::::::::: function to stop all intervals :::::::::::::::: \*/

function stopAllIntervals(){

for(var i = 0; i <= 5; i++){

clearInterval(intervals[i]);

if(i > 0 && i < 5){

var horses = document.getElementById(horse[i]);

horses.className = 'horse standRight';

}

}

}

/\* ::::::::: function to return random number that requires parameter :::::::::: \*/

function randFloor(number){ // function that takes 1 numeric parameter and returns integer random number from zero '0' to one number less than provided parameter.

var random = Math.floor(Math.random() \* number);

return random;

}

function randCeil(number){ // function that takes 1 numeric parameter and returns integer random number from '1' to given parameter.

var random = Math.ceil(Math.random() \* number);

return random;

}

function rand(number){ // function that takes 1 numeric parameter and returns floating point random number from '0' to given parameter.

var random = Math.random() \* number;

return random;

}

function randRange(number1, number2){ // function that takes 2 numeric parameter and returns 1 integer random number between the given parameters.

var inRange = false; // boolean variable

var temp = 0; // temporary variable to make swap possible.

if(number1 > number2){ // swap the value if number1 is greater then number2

temp = number1;

number1 = number2;

number2 = temp;

}

while(!inRange){ // while loop so that only value between range is given out

var random = Math.ceil(Math.random() \* number2);

if(random >= number1){

inRange = true;

}

}

return random;

}

/\* :::::::::::: All horses Animation on movement ::::::::::::::::: \*/

/\* ALL the below run functions takes a parameter that indicates to which horse the animation should be applied

\* this reduces code redundancy for animation of all horses\*/

function runLeft(i){ /\* all horses animation for running left \*/

var horses = document.getElementById(horse[i]); // choose a particular horse div, depending on parameter/argument

horses.className = 'horse runLeft';

}

function runUp(i){ /\* all horses animation for running UP \*/

var horses = document.getElementById(horse[i]);

horses.className = 'horse runUp';

}

function runRight(i){ /\* all horses animation for running right \*/

var horses = document.getElementById(horse[i]);

horses.className = 'horse runRight';

}

function runDown(i){ /\* all horses animation for running DOWN \*/

var horses = document.getElementById(horse[i]);

horses.className = 'horse runDown';

}

/\* ::::::::: All horses movement speed ::::::::::::::::: \*/

/\* similar to above run function,

these move functions also takes parameters,

the first parameter 'i' tells which horse to choose to make move

the second parameter 's' tells by what factor the horse should move

As the track is circular the inner round and the outer round (perimeters) are different which requires each horse to run at different random

speed but also each horse should have fixed range of random speed to make the match even, not use of this will be in result, that horse near to innter track would win.

\*/

function moveLeft(i,s){ /\* make horses move LEFT \*/

var random = (rand(4) + (rand(0.1) \* s)); // takes random number from custom rand() function

var horses = document.getElementById(horse[i]); // selects specific horse div, depends on paramenter/ argument

var positionLeft = horses.offsetLeft;

horses.style.left = positionLeft - random + 'px';

}

function moveUp(i,s){ /\* make horses move UP \*/

var random = (rand(4) + (rand(0.1) \* s));

var horses = document.getElementById(horse[i]);

var positionTop = horses.offsetTop;

horses.style.top = positionTop - random + 'px';

}

function moveRight(i,s){ /\* make horses move RIGHT \*/

var random = (rand(4) + (rand(0.1) \* s));

var horses = document.getElementById(horse[i]);

var positionLeft = horses.offsetLeft;

/\* the below if\_\_else.. condition statement is to make sure that the horse div right edge touches the finishline

ie. offsetLeft of horseDiv + 96(horse div length) == offsetLeft of finishline.

doing so, we can determine that horse reached the finish line... \*/

if((positionLeft < finishLine) && ((positionLeft+96) > (finishLine-20))){

horses.style.left = positionLeft + 1 + 'px'; /\* why + 1px? . ie because if we randomize the horse movement at this state

due to random numbers the condition (offsetLeft of horseDiv + 96(horse div length) == offsetLeft of finishline)

might not be true \*/

}else{

horses.style.left = positionLeft + random + 'px';

}

}

function moveDown(i,s){ /\* make horses move DOWN \*/

var random = (rand(4) + (rand(0.1) \* s));

var horses = document.getElementById(horse[i]);

var positionTop = horses.offsetTop;

horses.style.top = positionTop + random + 'px';

}

/\* ::::::::: MOVE HORses ::::::::::::::::: \*/

function moveHorse1(){

var horse = document.getElementById('horse1');

var positionTop = horse.offsetTop;

var positionLeft = horse.offsetLeft;

if((positionTop - 40 <= topLaneInner) && (positionLeft <= rightLaneOuter)){

runRight(1);

moveRight(1, rand(6));

/\* lap count \*/

if(positionLeft + 96 == finishLine){

lapCounter++;

if(gameMode == 2){

if(lapCounter > (lap \* 4) - 2){

wonTeam.push(1);

}

horse.className = 'horse standRight';

clearInterval(intervals[1]);

intervals[3] = setInterval(moveHorse3, 11);

}

// declare rank

else if(lapFinished == lap - 1){

rank.push('horse1');

}

}

}

if((positionTop >= bottomLaneInner) && (positionLeft >= leftLaneOuter)){

runLeft(1);

moveLeft(1, rand(6));

}

if((positionLeft-100 >= rightLaneInner) && (positionTop <= bottomLaneOuter)){

runDown(1);

moveDown(1, rand(7));

} if((positionLeft <= leftLaneInner) && (positionTop >= topLaneOuter)){

runUp(1);

moveUp(1, rand(7));

}

}

function moveHorse2(){

var horse = document.getElementById('horse2');

var positionTop = horse.offsetTop;

var positionLeft = horse.offsetLeft;

if((positionTop+40 <= topLaneInner) && (positionLeft <= rightLaneOuter)){

runRight(2);

moveRight(2, rand(5));

/\* lap count \*/

if(positionLeft + 96 == finishLine){

lapCounter++;

if(gameMode == 2){

if(lapCounter > (lap \* 4) - 2){

wonTeam.push(2);

}

horse.className = 'horse standRight';

clearInterval(intervals[2]);

intervals[4] = setInterval(moveHorse4, 11);

}

// declare rank

else if(lapFinished == lap - 1){

rank.push('horse2');

}

}

} if((positionTop + 10 >= bottomLaneInner) && (positionLeft >= leftLaneOuter)){

runLeft(2);

moveLeft(2, rand(5));

}

if((positionLeft-20 >= rightLaneInner) && (positionTop <= bottomLaneOuter)){

runDown(2);

moveDown(2, rand(5));

} if((positionLeft-20 <= leftLaneInner) && (positionTop >= topLaneOuter)){

runUp(2);

moveUp(2, rand(5));

}

}

function moveHorse3(){

var horse = document.getElementById('horse3');

var positionTop = horse.offsetTop;

var positionLeft = horse.offsetLeft;

if((positionTop-40 <= topLaneInner) && (positionLeft <= rightLaneOuter)){

runRight(3);

moveRight(3, rand(5));

/\* lap count \*/

if(positionLeft + 96 == finishLine){

lapCounter++;

if(gameMode == 2){

if(lapCounter > (lap \* 4) - 2){

wonTeam.push(1);

}

horse.className = 'horse standRight';

clearInterval(intervals[3]);

intervals[1] = setInterval(moveHorse1, 11);

}

// declare rank

else if(lapFinished == lap - 1){

rank.push('horse3');

}

}

} if((positionTop-20 >= bottomLaneInner) && (positionLeft >= leftLaneOuter)){

runLeft(3);

moveLeft(3, rand(5));

}

if((positionLeft-30 >= rightLaneInner) && (positionTop <= bottomLaneOuter)){

runDown(3);

moveDown(3, rand(5));

} if((positionLeft-60 <= leftLaneInner) && (positionTop >= topLaneOuter)){

runUp(3);

moveUp(3, rand(5));

}

}

function moveHorse4(){

var horse = document.getElementById('horse4');

var positionTop = horse.offsetTop;

var positionLeft = horse.offsetLeft;

if((positionTop <= topLaneInner) && (positionLeft <= rightLaneOuter)){

runRight(4);

moveRight(4, rand(4));

/\* lap count \*/

if(positionLeft + 96 == finishLine){

lapCounter++;

if(gameMode == 2){

if(lapCounter > (lap \* 4) - 2){

wonTeam.push(2);

}

horse.className = 'horse standRight';

clearInterval(intervals[4]);

intervals[2] = setInterval(moveHorse2, 11);

}

// declare rank

else if(lapFinished == lap - 1){

rank.push('horse4');

}

}

}

if((positionTop >= bottomLaneInner) && (positionLeft >= leftLaneOuter)){

runLeft(4);

moveLeft(4, rand(4));

}

if((positionLeft-randRange(30, 60) >= rightLaneInner) && (positionTop <= bottomLaneOuter)){

runDown(4);

moveDown(4, rand(4));

}

if((positionLeft <= leftLaneInner) && (positionTop >= topLaneOuter)){

runUp(4);

moveUp(4, rand(4));

}

}

/\* ::::::: bring horses to their start line :::: \*/

function setHorsesPosition(){

var positionTop = 1;

for(var i = 1; i <= 4; i++){

var horse = document.getElementById('horse' + i);

horse.style.top = positionTop + 'vh';

var left = 0.26 \* window.innerWidth;

horse.style.left = left - 80 + 'px';

positionTop += 4;

}

}

/\* ::::::::::::: horse indexing manages index of horse while gaming ::::::::: \*/

// this function manages index value of horses.. for better Graphics

function horseIndexing(){

//console.log('hi');

for(var i = 1; i <=4; i++){

for(var j = 2; j<= 4; j++){

var firstHorse = document.getElementById('horse' + i);

var secondHorse = document.getElementById('horse' + j);

var firstZIndex = window.document.defaultView.getComputedStyle(firstHorse).getPropertyValue('z-index'); //Getting the z-index of a DIV in JavaScript?. 2017. Stackoverflow.com.

var secondZIndex = window.document.defaultView.getComputedStyle(secondHorse).getPropertyValue('z-index'); //Getting the z-index of a DIV in JavaScript?. 2017. Stackoverflow.com.

if((firstHorse.offsetTop > secondHorse.offsetTop && firstZIndex < secondZIndex) || (firstHorse.offsetTop < secondHorse.offsetTop && firstZIndex > secondZIndex)){

firstHorse.style.zIndex = secondZIndex;

secondHorse.style.zIndex = firstZIndex;

}

}

}

}

/\* ::: this function sets the value to finishline which we later check to complete laps and declare winners :::::::\*/

function customFinishLineSetting(){

var divLine = document.getElementsByTagName('div')[0]; // select the first div of html doc. ie. CUSTOM made finish line div

finishLine = divLine.offsetLeft; // distance between left side of screen to left side of finishLine div.

}

/\* ::::::::::::::::: all kind of intervals are added in this function ::::::::::::::::::::: \*/

function setAllIntervals(){

/\* interval to move all horses \*/

intervals[1] = setInterval(moveHorse1, 15);

intervals[2] = setInterval(moveHorse2, 15);

intervals[3] = setInterval(moveHorse3, 15);

intervals[4] = setInterval(moveHorse4, 15);

intervals[5] = setInterval(horseIndexing, 1);

intervals[0] = setInterval(checkLap, 1);

}

/\* :::::::::::::: First run for RElay mode :::::::::::: \*/

function setFirstRun(){

intervals[1] = setInterval(moveHorse1, 11);

intervals[2] = setInterval(moveHorse2, 11);

intervals[5] = setInterval(horseIndexing, 11);

intervals[0] = setInterval(checkLap, 1);

intervals[6] = setInterval(preventOverGoing, 1);

}

function preventOverGoing(){

if(lapCounter == (lap \* 4)){

stopAllIntervals();

}

}

/\* :::::::::: function that get user input lap value ::::::::::::::: \*/

function getLap(){

var inputLap = document.getElementById("lap");

lap = inputLap.value;

}

/\*:::::::::::: function that get user input betting amount value :::::::::::::::: \*/

function getBetAmount(){

var inputAmount = document.getElementById("amount");

betAmount = inputAmount.value;

}

function setBetAmount(promptValue){

document.getElementById("amount").value = promptValue;

}

/\* :::::::::::::: function to get Current balance :::::::::::::: \*/

function getBalance(){

var currentBalance = document.getElementById('funds');

balance = parseInt(currentBalance.innerHTML);

// console.log(balance + 'balance'); // testing purpose

}

/\* ::::::::::::: function to set balance ::::::::::::::::::: \*/

function setBalance(finalBalance){

var currentBalance = document.getElementById('funds');

currentBalance.innerHTML = finalBalance;

}

/\* ::::::::: function to get user input horse :::::::::::: \*/

function getHorse(){

var getHorse = document.getElementById('bethorse');

selectedHorse = getHorse.value;

//console.log(selectedHorse); // testing purpose

}

/\* :::::::: check all User Input Validation :::::::::::::: \*/

function checkUserInput(){

var userInputValue = true;

// condition to check valid lap value

if(lap < 1 || lap > 5){

// code update to select choose from user input .. no need of below code

alert('Please Set Lap value between (1-5)');

startButtonActivated = true; // enables start button

userInputValue = false;

}

// condition to check valid amount

if(betAmount > balance){ // check if balance is enough to bit

promptInput = prompt('Not enough balance: your current balance is:: ' + balance + ' Enter new Bet amount:');

setBetAmount(promptInput);

console.log('balance ' + balance); // checking purpose

console.log('betamt ' + betAmount); // checking purpose

console.log('difference' + (betAmount - balance)); // checking purposes

userInputValue = false;

startButtonActivated = true; // enables start button

}else if(betAmount < 1){ // check if balance is less then ZERO.

promptInput = prompt('Bet amount must be greater then zero !! Enter new Bet amount:');

setBetAmount(promptInput);

userInputValue = false;

startButtonActivated = true; // enables start button

}

return userInputValue; // return false if any of user input is invalid

}

/\*::::::::: function to update balance when game starts :::::::::: \*/

function initialBalanceUpdate(){

balance = balance - betAmount;

setBalance(balance);

}

/\* ::::::::::::::::: audio play and pause :::::::::::: \*/

function playAudio(sound){

sound.play(); // plays sound //HTML DOM Audio play() Method. 2017. W3schools.com.

}

function stopAudio(sound){

sound.pause(); // pause sound HTML DOM Audio pause() Method. 2017. W3schools.com.

}

/\* ::::::::: SCRIPT run when START RACE button is pressed ::::::::::::::::: \*/

function startGame(i){

if(startButtonActivated == true){

startButtonActivated = false; // disables start button

rank = []; //clear array variable 'rank' that will store rank of horses

getLap(); // get user input lap

getBalance(); // get current balance of user

getBetAmount(); // get user input amount

if(i == 1){

getHorse(); // get user input horse

setHorsesPosition(); // calls setHorsesPosition() function

if(checkUserInput()){

// alert('play'); // for checking purpose

var horseSound = document.getElementById("horseSound");

var horseGallop = document.getElementById('gallopSound');

playAudio(horseSound);

playAudio(horseGallop);

setAllIntervals(); // calls setAllIntervals(); function

lapDisplay();

initialBalanceUpdate();

}

}else if(i == 2){

//alert('Relay mode'); // for checking purpose

getTeam();

setHorsesPosition();

if(checkUserInput()){

var horseSound = document.getElementById("horseSound");

var horseGallop = document.getElementById('gallopSound');

playAudio(horseSound);

playAudio(horseGallop);

setFirstRun();

lapDisplay();

initialBalanceUpdate();

}

}

customFinishLineSetting(); // calls customFinishLineSetting() function

}

}

/\* ::::::::::::::; FOR RELAY MODE :::::::::::: \*/

function enableRelayMode(){

var relayTeamSelect = document.getElementById('relayTeam');

relayTeamSelect.disabled = false;

var horseSelect = document.getElementById('bethorse');

horseSelect.disabled = true;

var startButton = document.getElementById('changeMode');

startButton.innerHTML = 'Play Normal Mode';

setHorsesPosition();

}

function enableNormalMode(){

var relayTeamSelect = document.getElementById('relayTeam');

relayTeamSelect.disabled = true;

var horseSelect = document.getElementById('bethorse');

horseSelect.disabled = false;

var startButton = document.getElementById('changeMode');

startButton.innerHTML = 'Play Relay Mode';

setHorsesPosition();

}

function getTeam(){

var getTeam = document.getElementById('relayTeam');

selectedTeam = getTeam.value;

}

/\* :::::::::::::: ODDS GENERATOR AND MANAGEMENT :::::::::::::::::: \*/

// function that generates new Odds value;

function oddsGenerator(){

var newOdds = false;

while(!newOdds){ // loop is formed to get random number until new number is generated.

var oldOdds = false;

var randomOdds = randRange(2,5); // custom made function that returns random number between provied range

//console.log(randomOdds); // for checking purpose

for(var i = 0; i <= oddArray.length; i++){

if(randomOdds == oddArray[i]){ // check if the obtain random number is already taken

oldOdds = true;

}

}

if(!oldOdds){

newOdds = true;

oddArray.push(randomOdds); // store newly generated random number in array variable

}

}

return randomOdds; // returns newly generated random number

}

function oddsManager(){

for(var i = 0; i <=3; i++){

var oddsTable = document.getElementById('odds');

var oddsRow = oddsTable.getElementsByTagName('tr')[i + 1];

var odds = oddsRow.getElementsByTagName('td')[2];

odds.innerHTML = oddsGenerator(); // generates new random number and set value in document

}

}

/\* main function \*/

function startScript(){

oddsManager();

setHorsesPosition();

var startButton = document.getElementById('start'); // get the element of DOM whose id Is start and store it in variable in startButton

startButton.addEventListener('click', function(){

startGame(gameMode);

}); // when startButton(DOM ELEMENT) is clicked startGame Function is called.

var startButton = document.getElementById('changeMode'); // get the element of DOM whose id Is start and store it in variable in startButton

startButton.addEventListener('click', function(){

if(startButtonActivated == true){

if(gameMode == 1){

gameMode = 2;

enableRelayMode();

}else{

gameMode = 1;

enableNormalMode();

}

}

}

);

}

// after HTML doc file is loaded then function startScript function is called.

document.addEventListener('DOMContentLoaded', startScript);