The Weakest Link

By: Nicholas Defranco

AND

Steven Bouttarath

Table of Contents:

1. Code
2. Screen shots with description

Code:

Index.html

<!DOCTYPE html>

<!-- *By: Nicholas Defranco and Steven Bouttarath*-->

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width">

<title>repl.it</title>

<link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<h1 id="round"></h1>

<h2 id="timer"></h2>

<p id="question"></p>

<form id="startform">

<input id="text" type="text" placeholder="Name" >

<label id="m">Male</label>

<input id="male" type="radio" name="gender" value="Male" > &nbsp;

<label id="f">Female</label>

<input id="female" type="radio" name="gender" value="Female" > &nbsp; <br>

<input type="button" value="Start!" id="startbut">

<input type="button" value="Resume!" id="resumebut">

</form>

<h3 id="next\_earn"></h3>

<h3 id="risk"></h3>

<form id="bankform">

</form>

<h3 id="bank"> </h3>

<form id="saveform">

</form>

<script src="script.js"></script>

</body>

</html>

Script.js

\*Game as well as local storage DO NOT work\*

// question work the first time, but the next time onward, even if you select the correct answer it will still say it is incorrect. This has been a recurring issue for a long time and we were not able to solve the problem.

//question array

const questions = [];

class Question {

constructor(str, a, b, c, d) {

this.question = str;

this.options = [a, b, c, d];

this.correct = a;

}

}

questions.push(new Question("Which of these games were released in 1996", "Super Mario 64", "Super Contra", "Super Mario World", "Donkey Kong"));

questions.push(new Question("Where is Nintendo World located", "New York City", "Toronto", "Buffalo", "LA"));

questions.push(new Question("The princess is in \_\_\_\_\_ \_\_\_\_\_", "Another castle", "Your house", "The wild", "The Kingdom"));

questions.push(new Question("Mario's last name is: ", "Mario", "Bro", "Luigi", "Man"));

questions.push(new Question("What does NES stand for?", "Nintendo Entertainment System", "Nintendo's Extra Stuff", "Nintendo's Encyption System", "Nintendo Education System"));

questions.push(new Question("When was Minecraft created?", "2009", "2010", "2008", "2011"));

questions.push(new Question("My body is \_\_\_\_\_\_", "Ready", "Not ready", "Here", "There"));

questions.push(new Question("What is the name of Megaman in japan?", "Rockman", "Metalman", "Cutman", "Still Megaman"));

questions.push(new Question("Who is the main character is the Legend of Zelda?", "Link", "Zelda", "Shiek", "Tetra"));

questions.push(new Question("Princess Zelda from The Legend of Zelda games by Nintendo has gone by a few names. Which of these names has she NOT been called in a video game?", "Malon", "Tetra", "Sheik", "She has only been called Zelda"));

questions.push(new Question("What was Mario originally refered to as?", "Jump man", "It was always Mario", "Red Luigi", "The Plummer Guy"));

questions.push(new Question("What video game console has the highest number of video game console sales of all time?", "Playstation 2", "Wii", "Xbox 360", "NES"));

questions.push(new Question("Who is the fastest of these video game characters?", "Sonic", "Mario", "Donkey Kong", "Little Mac"));

questions.push(new Question("What is the red ghost's name in pacman?", "Blinky", "Inky", "Clyde", "Pinky"));

questions.push(new Question("In Super Mario Bros, what colours was Mario's outfit?", "Red and Brown", "Red and blue", "Green and blue", "Orange and Brown"));

questions.push(new Question("Mario's first time saying 'It's-a me Mario!' in what game?", "Super Mario 64", "Super Mario World", "Super mario Sunshine", "Super Mario Galaxy"));

questions.push(new Question("What end of what year was the last Nintendo Power issue released?", "2012", "2010", "2008", "2014"));

questions.push(new Question("What year was the first Nintendo Power released", "1988", "1990", "1984", "1986"));

questions.push(new Question("Which of these games is considered a successor to Mario Paint", "Super Mario Maker", "Super Smash bros", "Mario Paint 2", "Hotel Mario"));

questions.push(new Question("How many Mario Kart games have been made? (excludes releases and must have been released on a nintendo console)", "8", "9", "6", "10"));

questions.push(new Question("Mike \_\_\_\_\_\_'s Punch-Out!!", "Tyson", "Huckabee", "Trout", "Sullivan"));

questions.push(new Question("When did Tetris come out?", "1984", "1980", "1986", "1989"));

questions.push(new Question("When did Super Mario World release in North America", "1991", "1988", "1992", "1995"));

questions.push(new Question("Super Mario World 2 \_\_\_\_\_\_\_\_\_\_\_\_", "Yoshi's Island", "6 Golden Coins", "Bowser's revenge", "Wario World"));

questions.push(new Question("When did the virtual boy release in North America?", "1995", "1991", "1998", "2000"));

const tree1 = [1000, 5000, 10000, 50000, 75000, 125000, 250000, 500000];

const tree2 = [1000, 10000, 75000, 125000, 500000, 1000000];

let risk = 0;

let bank = 0;

const startbut = document.querySelector("#startbut");

const form = document.querySelector("#startform");

let name;

startbut.addEventListener("click", function() {

name = document.querySelector("#text").value;

game();

});

const resume = document.querySelector("#resumebut");

let timer\_obj;

let timer = 120;

let storage = JSON.parse(localStorage.getItem('Data'));

if( typeof storage == 'undefined' ) {

game.counter = 'undefined';

}

resume.addEventListener("click", function() {

game.counter = --parseInt(storage[0]);

//let arr = [game.counter, timer,set\_ques.money\_index, set\_ques.counter, risk, bank];

timer = parseInt(storage[1]);

set\_ques.money\_index = parseInt(storage[2]);

set\_ques.counter = parseInt(storage[3]);

risk = parseInt(storage[4]);

bank = parseInt(storage[5]);

let rand = parseInt(storage[6]);

const form = document.querySelector("#startform");

form.removeChild(startbut);

let tmp = document.createElement("input");

tmp.setAttribute("type", "button");

tmp.setAttribute("value", `Bank`);

tmp.setAttribute("id", `bankbut`);

const bankform = document.querySelector("#bankform");

bankform.appendChild(tmp);

game();

});

function game() {

let treelen;

startform.removeChild(resume);

startform.removeChild(document.querySelector("#male"));

startform.removeChild(document.querySelector("#female"));

startform.removeChild(document.querySelector("#f"));

startform.removeChild(document.querySelector("#m"));

startform.removeChild(document.querySelector("#text"));

if( typeof game.counter == 'undefined' ) {

game.counter = 0;

} else {

for(let i = 0; i < questions[rand].options.length; i++) {

let tmp = document.createElement("input");

tmp.setAttribute("type", "button");

tmp.setAttribute("value", `${questions[rand].options[i]}`);

tmp.setAttribute("id", `option\_${i + 1}`);

tmp.setAttribute("class", `${rand}`);

tmp.addEventListener("click", function(id) {

if(document.querySelector(`#${this.id}`).getAttribute("value") == questions[this.className].correct && game.counter !== 3) {

risk = tree1[set\_ques.money\_index];

set\_ques.money\_index++;

if(set\_ques.money\_index === treelen) {

set\_ques.counter = 0;

set\_ques.money\_index = 0;

timer = 0;

// if haven't reached max risk for the round then increment round counter

} else {

set\_ques.counter++;

next(treelen);

}

} else if(document.querySelector(`#${this.id}`).getAttribute("value") == questions[this.className].correct && game.counter === 3) {

if(treelen === 0 && timer === 0) {

alert(`you win ${bank}`);

document.getElementsByTagName("body").removeChil(document.querySelector("#timer"));

} else {

next(treelen);

}

} else if(document.querySelector(`#${this.id}`).getAttribute("value") !== questions[this.className].correct && game.counter === 3) {

alert(`you lose`);

document.getElementsByTagName("body").removeChild(document.querySelector("#timer"));

} else {

risk = 0;

set\_ques.money\_index = 0;

set\_ques.counter++;

next(treelen);

}

questions.splice(this.className, 1);

});

form.appendChild(tmp);

}

const bankbut = document.querySelector(`#bankbut`);

bankbut.addEventListener("click", function() {

bank += risk;

risk = 0;

document.querySelector("#risk").innerHTML = `Risk Money: ${risk}`;

document.querySelector("#bank").innerHTML = `Bank Money: ${bank}`;

set\_ques.money\_index = 0;

if(game.counter === 1) {

document.querySelector("#next\_earn").innerHTML = `next earning: ${tree1[set\_ques.money\_index]}`;

} else if(game.counter === 2) {

document.querySelector("#next\_earn").innerHTML = `next earning: ${tree2[set\_ques.money\_index]}`;

}

});

const saveform = document.querySelector("#saveform");

let savebut = document.createElement("input");

savebut.setAttribute("type", "button");

savebut.setAttribute("value", `Save`);

savebut.addEventListener("click", function(){

let arr = [game.counter, timer,set\_ques.money\_index, set\_ques.counter, risk, bank];

localStorage.setItem('Data', JSON.stringify(arr));

});

saveform.appendChild(savebut);

}

game.counter++;

document.querySelector("#round").innerHTML = `Round ${game.counter}`;

if(game.counter === 1) {

treelen = tree1.length;

const form = document.querySelector("#startform");

form.removeChild(startbut);

let tmp = document.createElement("input");

tmp.setAttribute("type", "button");

tmp.setAttribute("value", `Bank`);

tmp.setAttribute("id", `bankbut`);

const bankform = document.querySelector("#bankform");

bankform.appendChild(tmp);

set\_ques(treelen, parseInt(Math.random() \* questions.length));

timer\_obj = setInterval(count, 1000);

timer = 120;

} else if (game.counter === 2) {

treelen = tree2.length;

timer\_obj = setInterval(count, 1000);

timer = 120;

set\_ques(treelen, parseInt(Math.random() \* questions.length));

} else if (game.counter == 3){

document.getElementsByTagName("body")[0].removeChild(document.querySelector("#timer"));

document.querySelector("#bankform").removeChild(document.querySelector("#bankbut"));

document.getElementsByTagName("body")[0].removeChild(document.querySelector("#next\_earn"));

document.getElementsByTagName("body")[0].removeChild(document.querySelector("#risk"));

treelen = 0;

timer = 60;

set\_ques(treelen, parseInt(Math.random() \* questions.length));

}

}

// sort randomizer

function ran(a, b) {

return 0.5 - Math.random();

}

function next(t) {

set\_ques(t, parseInt(Math.random() \* questions.length));

}

function set\_ques(treelen, rand) {

if( typeof set\_ques.counter == 'undefined' ) {

set\_ques.counter = 0;

}

if( typeof set\_ques.money\_index == 'undefined') {

set\_ques.money\_index = 0;

}

if(game.counter === 1) {

document.querySelector("#next\_earn").innerHTML = `next earning: ${tree1[set\_ques.money\_index]}`;

} else if(game.counter === 2) {

document.querySelector("#next\_earn").innerHTML = `next earning: ${tree2[set\_ques.money\_index]}`;

}

if(game.counter === 1 || game.counter === 2) {

document.querySelector("#risk").innerHTML = `Risk Money: ${risk}`;

document.querySelector("#bank").innerHTML = `Bank Money: ${bank}`;

}

document.querySelector("#question").innerHTML = `question #${set\_ques.counter + 1}: ${questions[rand].question}`;

questions[rand].options.sort(ran);

// set up game if first round and first question

if(game.counter === 1 && set\_ques.counter === 0) {

for(let i = 0; i < questions[rand].options.length; i++) {

let tmp = document.createElement("input");

tmp.setAttribute("type", "button");

tmp.setAttribute("value", `${questions[rand].options[i]}`);

tmp.setAttribute("id", `option\_${i + 1}`);

tmp.setAttribute("class", `${rand}`);

tmp.addEventListener("click", function(id) {

if(document.querySelector(`#${this.id}`).getAttribute("value") == questions[this.className].correct && game.counter !== 3) {

risk = tree1[set\_ques.money\_index];

set\_ques.money\_index++;

if(set\_ques.money\_index === treelen) {

set\_ques.counter = 0;

set\_ques.money\_index = 0;

timer = 0;

// if haven't reached max risk for the round then increment round counter

} else {

set\_ques.counter++;

next(treelen);

}

} else if(document.querySelector(`#${this.id}`).getAttribute("value") == questions[this.className].correct && game.counter === 3) {

if(treelen === 0 && timer === 0) {

alert(`Congrats ${name}! You win ${bank}`);

document.getElementsByTagName("body").removeChil(document.querySelector("#timer"));

} else {

next(treelen);

}

} else if(document.querySelector(`#${this.id}`).getAttribute("value") !== questions[this.className].correct && game.counter === 3) {

alert(`you lose`);

document.getElementsByTagName("body").removeChild(document.querySelector("#timer"));

} else {

risk = 0;

set\_ques.money\_index = 0;

set\_ques.counter++;

next(treelen);

}

questions.splice(this.className, 1);

});

form.appendChild(tmp);

}

const bankbut = document.querySelector(`#bankbut`);

bankbut.addEventListener("click", function() {

bank += risk;

risk = 0;

document.querySelector("#risk").innerHTML = `Risk Money: ${risk}`;

document.querySelector("#bank").innerHTML = `Bank Money: ${bank}`;

set\_ques.money\_index = 0;

if(game.counter === 1) {

document.querySelector("#next\_earn").innerHTML = `next earning: ${tree1[set\_ques.money\_index]}`;

} else if(game.counter === 2) {

document.querySelector("#next\_earn").innerHTML = `next earning: ${tree2[set\_ques.money\_index]}`;

}

});

const saveform = document.querySelector("#saveform");

let savebut = document.createElement("input");

savebut.setAttribute("type", "button");

savebut.setAttribute("value", `Save`);

savebut.addEventListener("click", function(){

let arr = [game.counter, timer,set\_ques.money\_index, set\_ques.counter, risk, bank, rand];

localStorage.setItem('Data', JSON.stringify(arr));

});

saveform.appendChild(savebut);

// if already set up, only change text showing on the buttons

} else {

for(let i = 0; i < questions[rand].options.length; i++) {

let tmp = document.querySelector(`#option\_${i + 1}`);

tmp.removeAttribute("value");

tmp.removeAttribute("class");

tmp.setAttribute("value", `${questions[rand].options[i]}`);

tmp.setAttribute("class", `${rand}`);

}

}

}

function count() {

timer -= 1;

if(timer >= 0) {

document.querySelector("#timer").innerHTML = `${timer}`;

} else {

clearInterval(timer\_obj);

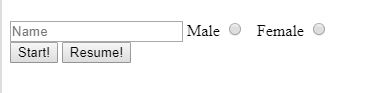
set\_ques.counter = 0;

game();

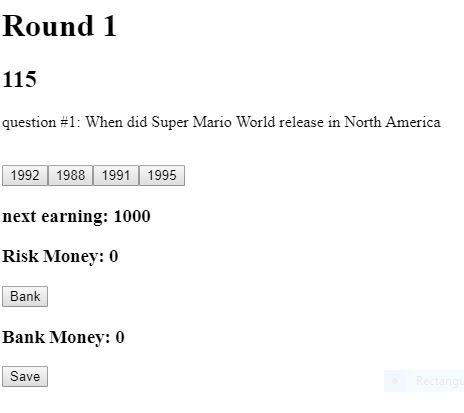
}

}

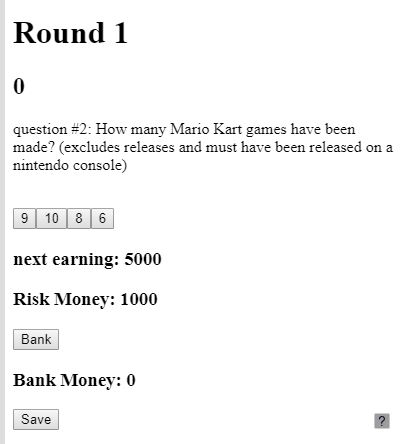
ScreenShots



* Start menu with a form



* Game plays in this format



* If the user gets a correct answer