) <!doctype html><html lang="en"><head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<meta name="description" content="">

<meta name="author" content="">

<title>The Randomeister for TI4</title>

<!-- Bootstrap core CSS -->

<!--<link href="vendor/bootstrap/css/bootstrap.css" rel="stylesheet">-->

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet">

<!-- <link rel="stylesheet" href="ti4styles.css"> -->

<style type="text/css" id="dcoder\_stylesheet">

:root {

--gunmetal-color: rgba(52, 58, 64, 0.959);

--default-grey: rgba(224, 224, 224, 0.829);

}

body {

background-color: rgba(219, 219, 219, 0.411);

height: auto;

margin: 0px;

padding: 0px;

}

.mt-4 {

box-sizing: border-box;

margin: 5px;

}

.grid-container {

display: grid;

margin: auto;

padding: 0px;

margin-top: 25px;

grid-template-columns: minmax(min-content, 300px) minmax(350px, 1fr);

/\* grid-template-columns: minmax(min-content, 1fr) minmax(min-content, 1fr); \*/

grid-column-gap: 0.3em;

grid-row-gap: 0.3em;

grid-auto-rows: minmax(15px, auto);

background: rgba(255, 255, 255, 0.151);

}

.results-container {

width: auto;

padding-left: 0px;

padding-right: 0px;

background: rgba(52, 58, 64, 0.719);

color: rgb(230, 230, 230);

border: 1px solid rgb(255, 255, 255);

border-top: 0px solid rgba(255, 255, 255, 0.363);

border-radius: 15px;

box-shadow: 1px 0px 3px rgb(0, 0, 0);

}

#result-list {

display: none;

width: auto;

margin-top: 15px;

margin-left: 15px;

margin-right: 15px;

padding: 20px;

padding-right: 0px;

padding-left: 10px;

font-size: 1.1em;

color: rgba(52, 58, 64, 0.842);

background-color: rgb(255, 255, 255);

border: 3px solid rgba(156, 156, 156, 0.986);

border-radius: 15px;

box-shadow: 3px 3px 5px rgba(10, 10, 10, 0.651);

}

.output-id {

font-weight: 500;

}

.output-playername {

padding-left: 5px;

/\* font-weight: 500; \*/

}

.player-header {

margin: 0px;

padding: 5px;

padding-left: 10px;

font-size: 1em;

background: rgb(214, 214, 214);

border-radius: 15px 0px 0px 15px;

border-bottom: 2px solid rgba(156, 156, 156, 0.336);

border-left: 2px solid rgba(156, 156, 156, 0.349);

}

.player-details {

padding-left: 15px;

margin-bottom: 0px;

margin-top: 2px;

font-size: 1em;

/\* font-weight: 500; \*/

}

.setting-inputs {

box-sizing: border-box;

position: relative;

width: 100%;

;

margin: auto;

justify-content: space-between;

width: 100%;

padding: 15px;

background: rgba(52, 58, 64, 0.842);

border: 1px solid rgb(255, 255, 255);

border-radius: 15px;

box-shadow: 1px 0px 3px rgb(0, 0, 0);

}

.race-container {

position: relative;

margin: auto;

padding-left: 15px;

padding-right: 15px;

background: rgb(200, 200, 201);

border: 2px solid rgba(37, 37, 37, 0.61);

border-bottom: 0px solid rgba(37, 37, 37, 0.61);

border-radius: 10px 10px 0px 0px;

box-shadow: 3px 3px 5px rgba(10, 10, 10, 0.651);

}

#player-inputs {

position: relative;

width: auto;

margin: auto;

background: rgb(200, 200, 201);

padding: 15px;

padding-bottom: 15px;

padding-top: 10px;

border: 2px solid rgba(37, 37, 37, 0.61);

border-top: 0px;

border-radius: 0px 0px 15px 15px;

box-shadow: 3px 3px 5px rgba(10, 10, 10, 0.651);

}

.player-settings-interior {

margin-top: 0px;

background-color: rgb(253, 251, 251);

border-left: 3px solid var(--default-grey);

border: 1px solid rgba(40, 44, 48, 0.493);

border: 0.15em inset rgba(52, 58, 64, 0.623);

border-radius: 15px;

padding-bottom: 10px;

margin: auto;

}

.player-counter-container {

width: auto;

margin: auto;

margin-bottom: 10px;

background: rgba(218, 216, 216, 0.438);

padding: 5px;

padding-bottom: 5px;

padding-left: 10px;

padding-right: 10px;

border-bottom: 1px solid rgba(40, 44, 48, 0.219);

vertical-align: middle;

}

/\* BUTTONS \*/

.button-container {

width: 100%;

display: flex;

position: relative;

justify-content: space-between;

/\* padding-left: 15px; \*/

margin: auto;

}

.list-buttons {

display: flex;

justify-content: space-between;

}

#select-btn {

background: rgba(50, 55, 59, 0.87);

font-size: 0.8em;

color: rgba(255, 255, 255, 0.836);

border: 2px solid rgba(214, 216, 219, 0.918);

margin-bottom: 10px;

border-radius: 5px;

padding-top: 0px;

padding-bottom: 0px;

}

#clear-btn {

background: rgba(50, 55, 59, 0.87);

font-size: 0.8em;

color: rgba(255, 255, 255, 0.842);

border: 2px solid rgba(214, 216, 219, 0.918);

margin-bottom: 10px;

border-radius: 5px;

}

.display-header {

margin: auto;

width: auto;

padding: 15px;

padding-bottom: 10px;

padding-top: 10px;

font-size: 1.2em;

height: auto;

margin-bottom: 0px;

vertical-align: middle;

}

.submit-container {

display: flex;

justify-content: space-between;

height: auto;

width: 80%;

margin: auto;

padding: 15px;

margin-right: 0%;

padding-top: 5px;

padding-bottom: 5px;

vertical-align: middle;

}

#submit-btn {

margin-bottom: 10px;

border: 1px solid rgba(214, 216, 219, 0.767);

display: none;

border-radius: 15px;

font-size: 1.1em;

background: rgba(22, 175, 43, 0.651);

padding: 2px;

padding-left: 10px;

padding-right: 10px;

box-shadow: 0px 1px 2px rgba(10, 10, 10, 0.493);

border: 1px solid rgba(224, 225, 226, 0.705);

}

#reset-btn {

border-radius: 15px;

font-size: 1em;

width: auto;

margin: auto;

margin-left: 0px;

padding-bottom: 4px;

padding-left: 5px;

padding-right: 5px;

display: none;

vertical-align: middle;

border: 1px solid rgba(255, 255, 255, 0.705);

}

.display-header-controls {

display: flex;

position: relative;

/\* z-index: 0; \*/

justify-content: space-between;

vertical-align: middle;

background: rgba(45, 48, 51, 0.473);

/\* height: auto; \*/

width: 100%;

margin: 0;

/\* margin-left: -15px; \*/

padding-top: 5px;

padding-left: 0px;

padding-right: 5px;

margin-bottom: 5px;

border-radius: 15px 15px 0px 0px;

border-bottom: 2px solid rgba(214, 214, 214, 0.473);

}

.flexInput {

display: flex;

justify-content: space-between;

vertical-align: middle;

margin: auto;

width: auto;

padding: 0px;

padding-left: 5px;

padding-right: 5px;

margin-bottom: 5px;

}

#player-counter {

width: 40px;

margin-bottom: 10px;

font-size: 0.9em;

background: rgb(219, 219, 219);

border: 1px solid rgba(40, 44, 48, 0.377);

border-radius: 5px;

padding: 2px;

vertical-align: middle;

}

.player-choice-label {

padding: 0px;

margin: 0;

vertical-align: middle;

}

.player-count-label {

padding: 0px;

margin: 0;

vertical-align: middle;

}

#choice-counter {

width: 40px;

margin-bottom: 0px;

font-size: 0.9em;

background: rgb(219, 219, 219);

border: 1px solid rgba(40, 44, 48, 0.377);

border-radius: 5px;

padding: 2px;

vertical-align: middle;

}

#player-details-container {

display: flex;

justify-content: space-evenly;

border: 1px solid rgba(255, 255, 255, 0.863);

width: auto;

padding-left: 15px;

margin: auto;

border-radius: 4%;

margin-top: 0px;

padding-top: 0px;

margin-bottom: 10px;

}

.textbox-label {

vertical-align: middle;

padding-bottom: 5px;

margin: auto;

width: 100%;

font-size: 1.1em;

}

.name-input-field {

padding-top: 0px;

padding: 5px;

width: 100%;

height: 30px;

margin: auto;

/\* width: auto; \*/

vertical-align: middle;

color: var(--gunmetal-color);

font-size: 0.9em;

margin-bottom: 5px;

background: rgba(40, 44, 48, 0.082);

border: 1px solid rgba(40, 44, 48, 0.377);

border-radius: 5px;

/\* margin-left: 20px; \*/

}

.top-container {

padding: 0px;

margin: 0px;

background: rgba(52, 58, 64, 0.952);

grid-area: 1/span 2;

width: 100%;

border: 2px solid rgb(124, 124, 124);

border-radius: 3px;

height: 20px;

}

.bottom-container {

padding: 0px;

background: rgba(52, 58, 64, 0.993);

grid-area: 2/span 2;

width: auto;

border: 2px solid rgb(124, 124, 124);

border-radius: 3px;

height: 100px;

}

.section-header {

position: relative;

border-bottom: 2px solid rgba(52, 60, 68, 0.397);

width: auto;

margin: auto;

margin-bottom: 10px;

padding-top: 5px;

font-size: 1.3em;

}

/\* RACE SELECTION BOX STYLES \*/

.raceList {

height: 225px;

overflow: auto;

width: 220;

border: 0.15em inset rgba(52, 58, 64, 0.623);

margin: auto;

padding: 5px;

margin-bottom: 5px;

margin-top: 10px;

font-size: 1.1em;

background: rgb(255, 255, 255, 1);

}

ul {

list-style-type: none;

margin: 5px;

padding: 5px;

background-color: rgba(243, 243, 243, 0);

}

li {

padding-left: 0px;

width: fit-content;

/\* background-color: rgba(243, 243, 243, 1); \*/

}

label {

display: block;

color: WindowText;

background-color: Window;

margin: 0px;

padding: 0px;

width: 100%;

border: 2px solid white;

align-self:flex-end;

}

.raceList label:hover {

background-color: Highlight;

color: HighlightText;

border: 2px solid Highlight;

font-weight: 500;

}

@media screen and (max-width: 400px) {

.grid-container {

grid-template-columns: 1fr;

width: 95%;

margin: auto;

margin-top: 10px;

}

}

@media screen and (max-width: 400px) {

.bottom-container {

padding: 10px;

grid-area: 3/span 1;

}

.top-container {

grid-area: 1/span 1;

width: auto;

}

}

</style></head>

<body>

<!-- Navigation -->

<nav class="navbar navbar-expand-lg navbar-dark bg-dark static-top">

<div class="container">

<a class="navbar-brand" href="#">The Randomeister</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarResponsive">

<ul class="navbar-nav ml-auto">

<li class="nav-item active">

<a class="nav-link" href="#">Home

<span class="sr-only">(current)</span>

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="./auxillary-docs/ti-rando-mock.svg">ti4 rando mockup</a>

</li>

<li class="nav-item">

<a class="nav-link" href="https://www.soundcloud.com/hamilsauce">Ham's Jams</a>

</li>

<li class="nav-item">

<a class="nav-link" href="https://hamilsauce.github.io">hamhome</a>

</li>

<li class="nav-item">

<a class="nav-link" href="https://seahag.sharepoint.com/sites/ti4/SitePages/TI-Game-Result-Data.aspx">Crazy TI

Game Data</a>

</li>

</ul>

</div>

</div>

</nav>

<!-- Page Content -->

<div class="container">

<div class="row">

<div class="col-lg-12 text-center">

<h4 class="mt-4" style="width: 100%">Get your random on</h4>

<p class="lead" style="width: 100%; margin: 0; line-height: 1; padding: 5px">ya dig?</p>

</div>

<div class="grid-container">

<div class="setting-inputs">

<form>

<div class="race-container">

<h5 class="section-header">Select Races</h5>

<ul class="raceList">

<li>

<label for="arborecChk">

<input checked class="raceBox" type="checkbox" name="arborecChk" id="arborecChk" value="Arborec"> Arborec

</label>

</li>

<li>

<label for="baronyChk">

<input checked class="raceBox" type="checkbox" name="baronyChk" id="baronyChk" value="Barony of Letnev"> Barony of Letnev

</label>

</li>

<li>

<label for="clanChk">

<input checked class="raceBox" type="checkbox" name="clanChk" id="clanChk" value="Clan of Saar"> Clan of Saar

</label>

</li>

<li>

<label for="embersChk">

<input checked class="raceBox" type="checkbox" name="embersChk" id="embersChk" value="Embers of Muaat"> Embers of Muaat

</label>

</li>

<li>

<label for="emiratesChk">

<input checked class="raceBox" type="checkbox" name="emiratesChk" id="emiratesChk" value="Emirates of Hacan"> Emirates of Hacan

</label>

</li>

<li>

<label for="federationChk">

<input checked class="raceBox" type="checkbox" name="federationChk" id="federationChk" value="Federation of Sol"> Federation of Sol

</label>

</li>

<li>

<label for="ghostsChk">

<input checked class="raceBox" type="checkbox" name="ghostsChk" id="ghostsChk" value="Ghosts of Creuss"> Ghosts of Creuss

</label>

</li>

<li>

<label for="l1z1xChk">

<input checked class="raceBox" type="checkbox" name="l1z1xChk" id="l1z1xChk" value="L1Z1X Mindnet"> L1Z1X Mindnet

</label>

</li>

<li>

<label for="mentakChk">

<input checked class="raceBox" type="checkbox" name="mentakChk" id="mentakChk" value="Mentak Coalition"> Mentak Coalition

</label>

</li>

<li>

<label for="naaluChk">

<input checked class="raceBox" type="checkbox" name="naaluChk" id="naaluChk" value="Naalu Collective"> Naalu Collective

</label>

</li>

<li>

<label for="nekroChk">

<input checked class="raceBox" type="checkbox" name="nekroChk" id="nekroChk" value="Nekro Virus"> Nekro Virus

</label>

</li>

<li>

<label for="sardakkChk">

<input checked class="raceBox" type="checkbox" name="sardakkChk" id="sardakkChk" value="Sardakk NOrr"> Sardakk N'Orr

</label>

</li>

<li>

<label for="universitiesChk">

<input checked class="raceBox" type="checkbox" name="universitiesChk" id="universitiesChk" value="Universities of Jol-Nar"> Universities of

Jol-Nar

</label>

</li>

<li>

<label for="winnuChk">

<input checked class="raceBox" type="checkbox" name="winnuChk" id="winnuChk" value="Winnu"> Winnu

</label>

</li>

<li>

<label for="xxchaChk">

<input checked class="raceBox" type="checkbox" name="xxchaChk" id="xxchaChk" value="Xxcha Kingdom"> Xxcha Kingdom

</label></li>

<li>

<label for="yinChk">

<input checked class="raceBox" type="checkbox" name="yinChk" id="yinChk" value="Yin Brotherhood"> Yin Brotherhood

</label>

</li>

<li>

<label for="yssarilChk">

<input checked class="raceBox" type="checkbox" name="yssarilChk" id="yssarilChk" value="Yssaril Tribes"> Yssaril Tribes

</label>

</li>

</ul>

<div class="button-container">

<!-- need to replace function in the onclick attrbute -->

<div class="list-buttons">

<input id="select-btn" class="btn btn-dark btn-sm" type="button" value="Select All" onclick="">

<input id="clear-btn" class="btn btn-dark btn-sm" type="button" value="Clear" onclick="">

</div>

<!-- <div class="submit-container">

<input id="submit-btn" class="btn btn-success btn-sm" type="button" value="Dispense!" onclick="getRaceList(this.form)">

</div> -->

</div>

</div>

<div id="player-inputs">

<h5 class="section-header">Players</h5>

<div class="player-settings-interior">

<div class="player-counter-container">

<div class="flexInput">

<p class="player-count-label">Player Count:</p>

<!-- <label for="choice-counter">Player Count:</label> -->

<select name="player-counter" id="player-counter">

<option value="0">-</option>

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

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

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

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

<option value="6">6</option>

</select>

</div>

<div class="flexInput">

<p class="player-choice-label">Player Choice:</p>

<!-- <label for="choice-counter">Player Choices:</label> -->

<select name="choice-counter" id="choice-counter">

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

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

</select>

</div>

</div>

<!-- need this markup for template literals -->

<div class="player-details-container">

</div>

</div>

</div>

</form>

</div>

<div class="results-container">

<div class="display-header-controls">

<h5 class="display-header">Race Assignments</h5>

<div class="submit-container">

<div><input id="submit-btn" class="btn btn-success btn-sm" type="button" value="Dispense!" onclick="getRaceList(this.form)"></div>

<div><input id="reset-btn" class="btn btn-outline-secondary btn-sm" type="button" value="Reset"></div>

</div>

</div>

<p id="result-list"> </p>

</div>

<div class="bottom-container">

<!-- </div>

<div class="sidebar">

</div>

<div class="sidebar"> -->

</div>

</div>

<!--grid-container -->

</div>

</div>

<!-- Bootstrap core JavaScript -->

<script src="vendor/jquery/jquery.slim.min.js"></script>

<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="ti4script.js"></script>

<!--<script src="ti4script2Revisions.js"></script> -->

<script type="text/javascript" id="dcoder\_script">const playerCounter = document.querySelector('#player-counter');

const raceListCard = document.querySelector('.race-container');

const resultListContainer = document.querySelector('.results-container');

const nameInputContainer = document.querySelector('.player-details-container');

const raceDisplay = document.getElementById('result-list');

const submitButton = document.querySelector('#submit-btn');

const resetButton = document.querySelector('#reset-btn');

const selectAllButton = document.querySelector('#select-btn');

const clearButton = document.querySelector('#clear-btn');

//const raceItems = document.querySelectorAll('.raceBox');

let textBoxes;

let playerNames = [];

let players = [];

let raceList = [];

let usedBoxes = [];

let nameBoxes = playerCounter.addEventListener('change', () => {

//Changes name textboxes displayed when player counter changes, returns the textbox form elements

let playerCount = parseInt(playerCounter.value);

return addInputs(playerCount);

});

submitButton.addEventListener('click', () => {

//on submit button, dole races and display

let nameArr = trimNames(playerNames);

console.log(trimNames);

raceList = getRaceList()

getPlayerData(nameArr, raceList);

doleOut(raceList);

displays();

});

selectAllButton.addEventListener('click', () => {

const raceItems = document.querySelectorAll('.raceBox');

raceItems.forEach(r => {

r.checked = true;

});

return;

});

clearButton.addEventListener('click', () => {

const raceItems = document.querySelectorAll('.raceBox');

raceItems.forEach(r => {

r.checked = false;

});

return;

});

function getPlayerNames(boxes) {

//function is run everytime a user deselects a text input - adds text content to playername list

let box = document.querySelector(boxes);

// let document.querySelectorAll('.name-input-field')]m0;

let [bName, bVal, pos] = [box.name, box.value, parseInt(box.name.split('').pop()) - 1];

if (bVal == '' || bVal === undefined) /\*&& playerNames[pos].length > 1) \*/{

console.log(`pName elem = ${playerNames[pos]}, getting erased`);

playerNames[pos] = '';

box.style.color = 'black';

box.style.border = '1px solid rgba(40, 44, 48, 0.377)';

// playerNames.splice(pos, 1);

} else if (bVal == playerNames[pos]) {

console.log(`${bVal} is already in ${bName}`);

//return;

} else {

if (playerNames.length > 6) {

alert("Already Got enough players. Click Reset to start over.");

} else {

playerNames[pos] = bVal;

// box.disabled = true;

box.style.color = 'grey';

box.style.border = '2px solid green';

console.log('last conditon reached');

}

}

console.log('pNames ' + playerNames);

console.log('pNames len' + playerNames.length);

console.log(`val at pos ${pos} = ${playerNames[pos]} vs ${bVal} in box`);

return playerNames;

}

function trimNames(names){

let trimmed = names.filter(n => {

if (typeof n === 'string' && (n != '' && n != ' ')) {

console.log('passing ' + n);

return n;

}

console.log('filteting thisnel');

});

return trimmed;

}

function getRaceList() {

//iterate over checklist of races, snag the checked ones, note that for loop is preferable to forEach here because need to reference index of each checklist item/race

const raceChecks = document.querySelectorAll('.raceBox');

console.log(raceChecks);

let selectBoxes = [];

for (let i = 0; i < raceChecks.length; i++) {

if (raceChecks[i].type == 'checkbox' && raceChecks[i].checked == true) {

selectBoxes.push(raceChecks[i].value);

}

console.log(selectBoxes);

}

return selectBoxes;

}

function addInputs(count) {

/\*for each player (according to counter selection), 1) add text input html to output array, each with player ID label/Id/Name/selectors incremented accordingly.

Then 2) join the array elems and write it to the inner html of the 'player-details-container' div \*/

const output = [];

for (let i = 1; i <= count; i++) {

output.push(`

<div class="flexInput">

<p class="textbox-label">Player ${i}</p>

<input type="text" class="name-input-field" name="playername-textbox${i}" id="playername-textbox${i}" value=""

placeholder="Enter a name..." onblur="getPlayerNames('#playername-textbox${i}')">

</div>

`);

}

nameInputContainer.innerHTML = ` ${output.join('')}`;

toggleSubmit();

}

function toggleSubmit() {

//show or hide subit buton

let textFields = document.querySelectorAll(".name-input-field");

if (textFields.length > 0) {

submitButton.style.display = "inline";

resetButton.style.display = "inline";

} else {

submitButton.style.display = "none";

resetButton.style.display = "none";

}

return textFields;

}

function getPlayerData(names, races) {

//for each name in array, add a new player object to array

let i = 1;

names.forEach(name => {

players.push({

playername: name,

id: i,

gotRaces: [],

displayData: function(c) {

let showRaces =

c == 2 ? this.gotRaces.join(', ') : this.gotRaces[0];

let Output = '';

output =

`<h4 class="player-header"><span class="output-id">Player ${this.id}</span></h4>

<p class="player-details">${this.playername} - ${showRaces}</p>`;

return output;

},

getRace: function (races) {

//randomize/shuffle race array and select first element, then remove that from array

for (let i = races.length - 1; i > 0; i--) {

let j = Math.floor(Math.random() \* (i + 1));

[races[i], races[j]] = [races[j], races[i]];

}

this.gotRaces.push(races.splice(0, 1));

console.log(this.gotRaces);

}

});

i++;

});

return players;

}

//Calls getRace method for each player, repeat once

function doleOut(races) {

let pl = players;

for (let i = 0; i < 2; i++) {

pl.forEach(p => {

p.getRace(races);

});

};

console.log(races);

}

//concats the player displayData() output and pushes to DOM

function displays() {

const resultList = document.querySelector('#result-list');

const choiceCount

= document.querySelector('#choice-counter').value;

let pl = players;

let outText = [];

pl.forEach(p => {

outText.push(p.displayData(choiceCount));

});

raceDisplay.innerHTML = outText.join('<br>');

resultList.style.display = "block";

}

// eed to clear players objects, races array, clean out the result list HTML,

// Hide buttons, and clear names in textboxes.

//When reset button is pressed, fire off page reload

resetButton.addEventListener('click', () => {

document.location.reload(true);

});

/\*

// resetButton.addEventListener('click',() => {

// const resultList = document.querySelector('#result-list');

// players.length = 0;

// // raceList.length = 0;

// console.log(players);

// console.log(playerNames);

// console.log(selectBoxes);

// // playerNames.length = 0;

// // selectBoxes.length = 0;

// resultList.style.display = "none";

// raceDisplay.innerHTML = "";

// // submitButton.style.display = "none";

// });

// // function resetData() {

// // }

\*/

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