


Marie Oei  
A0240698U

## NM2207 Final Project Interim Report

The player encounters a lost blob/slime in the forest that they have to bring home. Through a series of 3 mini games, the player will be able to get clues about where the home is and the code to enter the home. After the 3 games are played, the options to choose the correct home will be shown.

Depending on which games are played first, the clues and partial code given are non-sequential. It is up to the player to decipher the right order of the code and the right home based on the clues.

 storyboard.jpg

Week	Concept	How I've used it	Filename
2	Styling with CSS	<pre>1  html { 2    width: 100%; 3    height: 100%; 4    overflow: hidden; 5  } 6 7  body { 8    background-image: url("../resources/forestBG.jpg"); 9    font-family: -apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen, 10     Ubuntu, Cantarell, Fira Sans, Droid Sans, Helvetica Neue, sans-serif; 11    color: white; 12    padding-left: 12px; 13    padding-top: 4px; 14    width: 100%; 15    height: 100%; 16    text-align: center; 17 18    /* fade in animation upon homepage loading from https://www.geeksforgeeks.org/ */ 19    animation: fadeInAnimation ease 2s; 20    animation-iteration-count: 1; 21    animation-fill-mode: forwards; 22  }</pre> <pre>196 .note { 197   height: 250px; 198   width: 50%; 199   background-color: #784b13; 200   display: flex; 201   position: relative; 202   margin: auto; 203   margin-top: 75%; 204   z-index: 200; 205 } 206 207 .message { 208   text-align: left; 209   margin: auto; 210   margin-left: 5%; 211   margin-right: 5%; 212   color: white; 213   font-family: 'Courier New', Courier, monospace; 214 }</pre> <p>Used CSS to style the body of the webpage and class names, amongst many others. Mainly working on trying to center the text in div, and div in div. Learnt about z-index on the internet whereby I can choose to adjust which element is to be positioned on top/in front. Use of percentages over pixels so that when</p>	appstyle.css

		the window is resized, the text (hopefully) does not overflow.	
3	Declaring variables, using if statements and console.log	<pre> 60 let gameState = 0; 61 62 // initial click event listener to change text 63 mainDiv.addEventListener('click', function(){ 64     if (gameState == 0) { 65         intText.innerHTML = "Is this your first time in this forest?"; 66 67         yesButton.className = "button"; 68         yesButton.innerHTML = "Yes"; 69         yesButton.onclick = function () { 70             /* console.log("yesButton is clicked"); */ 71             yesButtonClicked(); 72         }; 73         buttonContainer.appendChild(yesButton); 74         /* console.log("yesButton created"); */ 75 76         noButton.className = "button"; 77         noButton.innerHTML = "No"; 78         noButton.onclick = function () { 79             noButtonClicked(); 80             /* console.log("noButton is clicked"); */ 81         }; 82         buttonContainer.appendChild(noButton); 83         /* console.log("noButton created"); */ 84 85         click2Cont.remove(); 86 87     } 88 }) </pre> <p>Using the state is also a concept from week 5, but here I declared the variable gameState using “let” and also used an if statement to check the state of the game. If the state of the game is a certain number, then the code can run, if not, the code will not run.</p>	main.js
4	Creating functions, engaging with DOM elements and using console.log	<pre> 1 // div variables 2 let mainDiv = document.getElementById("mainDiv"); 3 let intText = document.getElementById("h1Text"); 4 let content = document.getElementById("content"); 5 let buttonContainer = document.getElementById("buttonContainer"); 6 7 // button variables 8 let yesButton = document.createElement("button"); 9 let noButton = document.createElement("button"); 10 let theEndButton = document.createElement("button"); 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 // clicking the no button 91 function noButtonClicked() { 92     intText.innerHTML = "Well then, this won't be much fun"; 93     intText.style.fontSize = "36px"; 94 95     yesButton.remove(); 96     noButton.remove(); 97 98     theEndButton.className = "button"; 99     theEndButton.id = "buttonToHome"; 100     theEndButton.innerHTML = "Leave"; 101     theEndButton.onclick = function () { 102         goBack(); 103         /* console.log("theEndButton is clicked"); */ 104     }; 105     buttonContainer.appendChild(theEndButton); 106 } </pre> <p>Creating the noButtonClicked() function (see week</p>	main.js

		<p>3, line 79). Manipulating DOM elements to be declared with certain class names and id, also playing with the innerHTML and style. Use of appendChild() to display the element in the webpage on the onset of the function.</p>	
5	Using event listeners	<pre> 207 // event listener to show the guide again 208 document.body.addEventListener('keydown', function(e){ 209     if (e.keyCode == 191) { 210         content.appendChild(note); 211         note.appendChild(noteMessage); 212         /* note.style.display = "flex"; */ 213     } 214 }); 215 document.body.addEventListener('keyup', function(e){ 216     noteMessage.remove(); 217     note.remove(); 218 }); </pre> <p>Creating a keydown event listener within the body of the document (though using 'window' might also work? (Did not try)). When a certain key is held down (ie. the '/' key as labelled by keyCode 191), elements note and noteMessage will pop up, but when the key is released, the elements disappear.</p>	main.js
6	Using session06.practice with the keyboard	<pre> 300 // declaring an empty array to store the user's key presses 301 let userKeys = []; 302 303 function keyTune() { 304     window.addEventListener('keydown', function(e) { 305         if (gameState == 7) { 306             playNote(e); 307 308             // store the key code in the userKeys array 309             userKeys.push(e.keyCode); 310 311             // compare the user's key presses to the correct sequence 312             if (userKeys.length == pianoAudio.length &amp;&amp; userKeys.every((value, index) =&gt; value == pianoAudio[index])) { 313                 // if the user's key presses match the correct sequence, show the clue and number 314                 mainPiano.style.display = "none"; 315                 listen.remove(); 316 317                 intText.innerHTML = "Here's your clue and number."; 318                 click2Cont.innerHTML = "Click to continue."; 319 320                 gameState = 7.5; 321                 clue2(); 322             } else if (userKeys.length == pianoAudio.length &amp;&amp; userKeys.some((value, index) =&gt; value != pianoAudio[index])) { 323                 // if the user's key presses do not match the correct sequence, reset the userKeys array 324                 window.alert("Try again."); 325                 userKeys = []; 326             } 327         } 328     }); 329 } 330 331 // clue2 function 332 function clue2() { 333     if (gameState == 7.5) { 334         content.appendChild(note); 335         note.appendChild(noteMessage); 336         note.style.backgroundColor = "rgb(235, 205, 124)"; 337         note.style.color = "rgb(130, 95, 0)"; 338         noteMessage.innerHTML = "It looks cool!" + "&lt;br&gt;" + "&lt;br&gt;" + "2"; 339         console.log("clue printed"); 340     } 341 } 342 343 344 function playNote(e) { 345     console.log("playNote is triggered! e.keyCode is " + e.keyCode); 346 347     //keycode is a number eg 65, but what are the ids of the audio elements in index.html like? 348     let pianoKey = document.querySelector('audio[data-key="' + e.keyCode + '"]'); 349 350     //use getAttribute() to get the attribute of an HTML element, e.g. the note which is being played. 351     if (pianoKey != null) { 352         let audio = new Audio(pianoKey.getAttribute("src")); 353 354         audio.currentTime=0; 355         audio.play(); 356     } 357 } </pre> <p>Trying to make a playable keyboard where if 5 keys are pressed correctly and in the right sequence, a hint will show up. If the wrong keys are pressed or keys are pressed in the wrong sequence, there will be a reset of the array. (ie. the event listener will push the keycode pressed into the userKeys array</p>	main.js

		<p>and if the array does not correspond with what was set in the pianoAudio array (not shown in the screenshot), then there will be a reset).</p> <p>However, here there is the issue of the hint ('note' element) not showing up. My thinking is that it is because the function clue2() is within a keydown event listener but I do not know how to fix this yet.</p>	
7	[Not using Chartjs, thus my own reflection on progress]	<pre> 359 // ----- GAME 3 - collect/sort out the berries 360 // gameState = 0; 361 // have a timer event, maybe 25 seconds; player has to either sort or collect 20 berries within the timing 362 // when timer runs out and the game is uncompleted, reset </pre> <p>I would eventually want to try out the timer code that was shown in the bonus videos for the ones to make a game, but a timer that counts down instead. This would probably be for Game 3, which I am still undecided about but I think I want to try making a sorting game or maybe a game that has falling objects (ie. berries) and the player has to collect the right berries.</p>	main.js
8	[Not using Chartjs, thus my own reflection on progress]	<pre> 220 // implementing the game buttons 221 function clickBlob3() { 222     mainDiv.addEventListener('click', function(){ 223         if (gameState == 4) { 224             gameState = 5; 225 226             intText.innerHTML = "I guess we have to play some games now..."; 227             click2Cont.innerHTML = "Choose a game."; 228 229             playGames(); 230         } 231     }); 232 } 233 234 // function to append the game buttons 235 function playGames() { 236     buttonContainer.appendChild(game1); 237     buttonContainer.appendChild(game2); 238     buttonContainer.appendChild(game3); 239 } 240 </pre> <p>At this point I realised the most effective method for my code would be to constantly use functions and place an event listener within it, and within the event listener itself, that is where the gameState variable should be checked.</p> <p>Other arrangements that I have tried usually end up glitching the entire code/webpage, so while it seems rather inefficient, this way unfortunately works the best for me and also helps me understand my own code since I think and work quite systematically.</p> <p>I placed the button appending into a function so I can call the function in other codes as well.</p>	main.js