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>html {
		  width: 100%;
		  height: 100%;
		  overflow: hidden;
		}
		
		body {
		  background-image: url("../resources/forestBG.jpg");
		  font-family: -apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen,
		  Ubuntu, Cantarell, Fira Sans, Droid Sans, Helvetica Neue, sans-serif;
		  color: white;
		  padding-left: 12px;
		  padding-top: 4px;
		  width: 100%;
		  height: 100%;
		  text-align: center;
		
		  /* fade in animation upon homepage loading from https://www.geeksforgeeks.org/ */
		  animation: fadeInAnimation ease 2s;
		  animation-iteration-count: 1;
		  animation-fill-mode: forwards;
		}</pre><pre>.note {
		  height: 250px;
		  width: 50%;
		  background-color: #784b13;
		  display: flex;
		  position: relative;
		  margin: auto;
		  margin-top: 75%;
		  z-index: 200;
		}
		
		.message {
		  text-align: left;
		  margin: auto;
		  margin-left: 5%;
		  margin-right: 5%;
		  color: white;
		  font-family: 'Courier New', Courier, monospace;
		}</pre>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
			appstyle.css

		the window is resized, the text (hopefully) does not overflow.	
3	Declaring variables, using if statements and console.log	 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.	main.js
4	Creating functions, engaging with DOM elements and using console.log	  Creating the noButtonClicked() function (see week	main.js

```
let gameState = 0;

// initial click event listener to change text
mainDiv.addEventListener('click', function(){
    if (gameState == 0) {
        intText.innerHTML = "Is this your first time in this forest?";

        yesButton.className = "button";
        yesButton.innerHTML = "Yes";
        yesButton.onclick = function () {
            /* console.log("yesButton is clicked"); */
            yesButtonClicked();
        };
        buttonContainer.appendChild(yesButton);
        /* console.log("yesButton created"); */

        noButton.className = "button";
        noButton.innerHTML = "No";
        noButton.onclick = function () {
            noButtonClicked();
            /* console.log("noButton is clicked"); */
        };
        buttonContainer.appendChild(noButton);
        /* console.log("noButton created"); */

        click2Cont.remove();

    }
})
```

```
// div variables
let mainDiv = document.getElementById("mainDiv");
let intText = document.getElementById("h1Text");
let content = document.getElementById("content");
let buttonContainer = document.getElementById("buttonContainer");

// button variables
let yesButton = document.createElement("button");
let noButton = document.createElement("button");
let theEndButton = document.createElement("button");
```

```
// clicking the no button
function noButtonClicked() {
    intText.innerHTML = "Well then, this won't be much fun
    intText.style.fontSize = "36px";

    yesButton.remove();
    noButton.remove();

    theEndButton.className = "button";
    theEndButton.id = "buttonToHome";
    theEndButton.innerHTML = "Leave";
    theEndButton.onclick = function () {
        goBack();
        /* console.log("theEndButton is clicked"); */
      };
    buttonContainer.appendChild(theEndButton);
}
```

		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.	
5	Using event listeners	 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.	main.js
6	Using session06.practice with the keyboard	  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	main.js

		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). 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.	
7	[Not using Chartjs, thus my own reflection on progress]	 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.	main.js
8	[Not using Chartjs, thus my own reflection on progress]	 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. 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. I placed the button appending into a function so I can call the function in other codes as well.	main.js