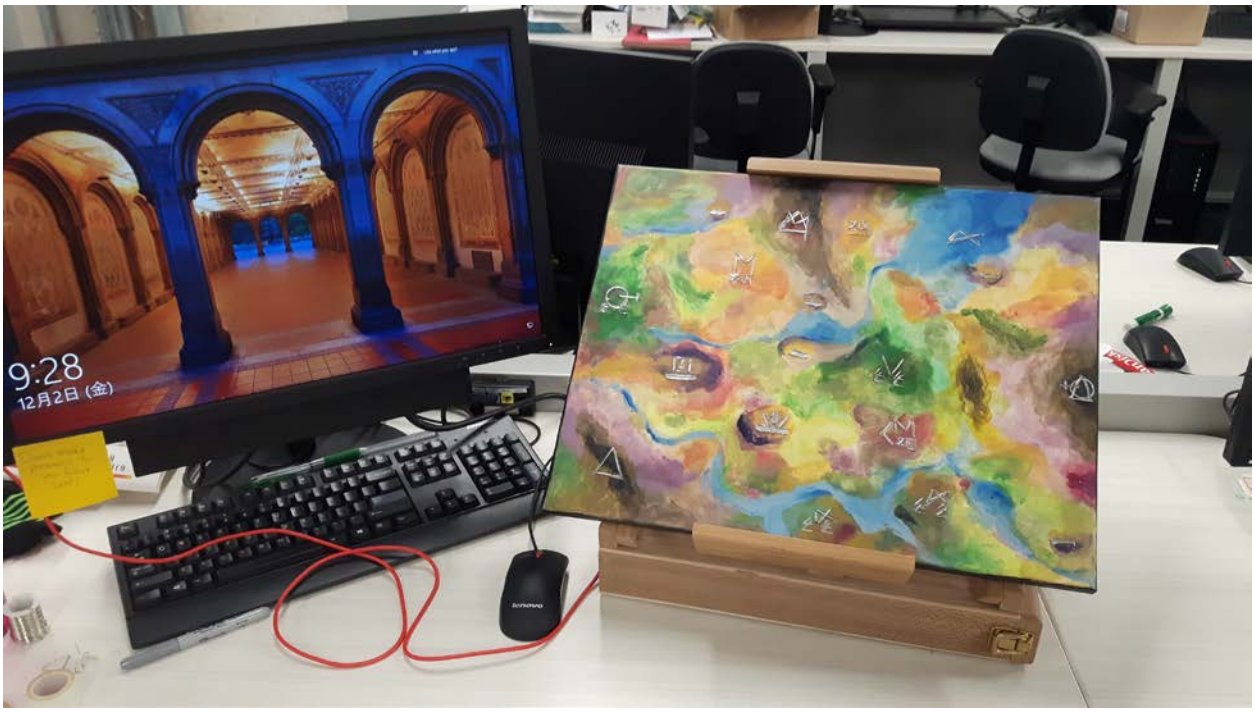
IMM Design Studio 1 – December 2016
Meagan Ellen Peat

Makey-Makey Puzzle Map
Breakthroughs Stumbles and Victories



Introduction

Following ancient instructions and cryptographic symbols navigate your way through a colourful map to solve the puzzle!

How it works

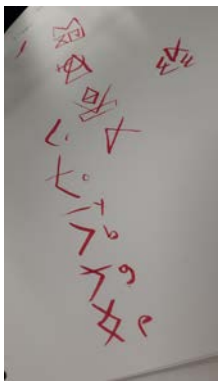
The project uses a computer, a browser page with some JavaScript, and a Makey Makey board in combination with beading wire, electrical wire, a physical painted canvas, paper and a desk easel. The beading wire on the front of the canvas connects to wires on the back to the Makey Makey board and then the computer. Holding the ground connected to the board the user then completes the circuit to enter input into the browser. The browser listens for the input and plays a victory sound if the player presses the keys in the correct order. The page is currently hosted at http://peatm.dev.fast.sheridanc.on.ca/techStudio/.

Each symbol in the solution is attached to a different part of the Makey Makey board. They use the connections for the board for WASDFG. All of the other symbols are wired together and then to the Makey Makey board's space entry part. The "mystical artifact" that the player must guide is attached to the ground.

The JavaScript on the page takes every input from the map interface and stores them in an array. The array holds up to six values. Whenever a new value is added, it removes the oldest entry of the six and checks to see if the last entered six values matches the expected or correct entry. If this is the case the browser plays victory music.

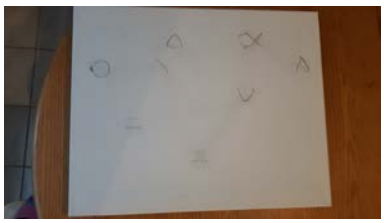
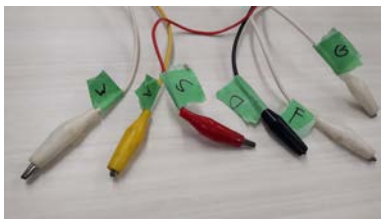
How it was made

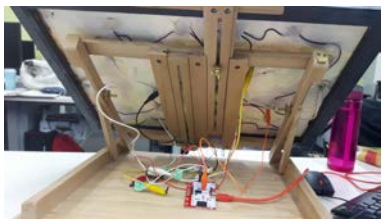
The first step of the project was to test the Makey Makey board. One of the first things I completed was a short JavaScript script that would check entries from the keyboard of Makey Makey against an entry that would solve the puzzle. The next step was to design the interface in which the player would solve the puzzle, and of course the puzzle that would be solved. Once I decided on combining a cryptogram like puzzle and a map I tested out a simplified version of the map using a canvas and some beading wire. When I was happy that the idea was going to work I started painting a second canvas. I used acrylic paints and my fingers after brushing down a layer of watered down brown paint as a base. When the main design was completed I also painted the edges of the canvas to help polish it off. Next I took my puzzle design and drew the symbols onto the map with a silver sharpie. From there took beading wire and made the shapes, poking the ends through to the other side of the canvas. I attached all the symbols that were not a part of the final solution to each other using wire. Next I placed the canvas on the easel. From there I attached and all of the other symbols to the board using labeled alligator clips and the dummy symbols to the space bar part of the board. I attached the ground to the "sacred" object that the player would have to lead home. For the hints I took thick paper and wet teabags to create an aged effect. Then I wrote the hints using a calligraphy pen after the paper had dried. I placed the hints in a slightly opened compartment of the easel for the player to find.




Design, Design, Design – Plan, Plan, Plan


Build, Test, Test, Build, Test, Test


Initial Plan

My original sketches featured a rectangular grid as an input panel with a separate panel for a handprint to connect the scan, but as I worked on designing the project I felt that my original idea did not fully embrace the imaginative capabilities of the Makey-Makey board. You can easily find a keypad in an office or make the panel using actual buttons that press down. I felt in my test that pressing a panel and not having any feedback was ultimately unsatisfying. While trying to solve the separate problem of how to setup my display I came across my easel and came up with the idea of making a map on a canvas. I feel that the texture of paint and a canvas enhances the experience. From the start I knew that I wanted to use some sort of cryptogram so I combined that with the idea of giving instructions to get to my final project.

Inspiration

I was heavily inspired by point and click adventure games, and escape room games (both digital and physical). This puzzle is inspired by a puzzle from a Broken Sword game, where you solve a cryptogram like puzzle to get directions. In the game you follow the directions in a separate puzzle and I wanted to combine it all into one map.

Breakthroughs

I chose this project to work on my design skills and I really feel that the map design was a major victory for me. Escape rooms and puzzles are often about seeing things from another perspective and I feel that up close the painting is not completely recognizable as a map, but from a distance the rivers make the map much more understandable. The painting style and mix of colors really achieves this effect. At first I was worried it would be too obvious from any perspective or not look like a map at all, but as I filled it in and added more colors and contrasts it properly took shape.

Stumbles

I found that attempting to press a button that did not feel or react as a button was ultimately unsatisfying. I spent a long time trying to come up with a solution that would not provide feelings of let down and disappointment. Eventually I came up with the idea of a design that would allow the user to drag their fingers.

In switching to dragging I encountered a technical problem. Since I made the shapes out of wire and moved away from a single press my original code had some problems with the Makey-Makey board registering the same input multiple times which made it difficult for me to code something that could check if the path was correct. This was a problem both for dragging and for extended presses. With the player trying to solve the puzzle step by step they are likely to linger on certain spaces.

At first I changed the expected input that the browser was looking for in an event listener from “keypress” to “keydown”, but that did fix the problem with the Makey Makey board involved.

```
window.onload = function(){
    //listen to the keyboard
    document.addEventListener('keypress', enterPassword);
}
```

```
window.onload = function(){
    //listen to the keyboard
    document.addEventListener('keydown', enterPassword);
}
```

To check for the correct path I use an array that always has six entries. Every time the player presses a new button it adds to the end of the array and removes the first entry and checks if the array matches the password. The final solution that I came up with was to only accept entries into the array that did not match the last inputted value which I did by wrapping the contents of the enter password function in an if statement.

```
if(password[5]!= key){
```

Victories

I added a sound that plays when the player successfully completes the puzzle. The code pics from one of three (at the moment) possible victorious sounds to showcase the different situations the puzzle or something like it could fit into. I find that this aspect really creates a sense of victory!

```
function checkWin(){
    var pass = password.toString();
    if (pass == endGoal){
        console.log("you win!");
        var victory = document.getElementById('victory');
        var success = document.getElementById('success');
        var zelda = document.getElementById('zelda');
        var sounds = [victory, success, zelda];
        var rand = Math.floor(Math.random()*sounds.length);
        sounds[rand].play();
        //document.getElementById('victory').play();
    }
}
```