P2 - what type of song is this?

george mcfarlane

The basic, stripped down idea:

You type on your keyboard, letters show up, little musical things happen as you type each letter, and spelling a secret word taken from a small dictionary plays a little song. In order to provide some hints about which words to type, there will be some floating text across the screen every once in a while that will nudge the player towards a word or two. The ending state occurs if the user is able to type in all the words.

The interaction between sound and code is something that interests me a lot, more specifically the influence sound has on mood and tone, and how that could be implemented to make you feel ~things~ in a p5 based game. Depending on the words and music used, and the combinations thereof, there's an opportunity here to tell a sort of hidden story between the cracks, something that connects the words and the songs they are tied to. Don't get me wrong, I'm not writing the next blockbuster, but I've been deeply fascinated with storytelling like the visuals in Julien Mauve's After Lights Out and Jon Bois' 17776 , and would love to replicate something in this same low-key style.

In addition to external influences, I've done some very minor text-based artwork using a video-soft synth called Cathodemer, which has a similar mood to this projects aims. I feel it encapsulates the general tone I was looking for.

here it is
01
02

Achieving this will be an interesting challenge, thematically, as it relies on a few things simultaneously. Firstly, my own ability to create a story thread that is not obvious but still exists, and being able to guide the user along this thread. Secondly, needing to pick words that are both interesting and not extremely unexpected. Adding the coding requirements onto that, this is something I'm nervous about, but feel I can pull off.

Technical challenges:

Oh boy. The way I see it, there are about three main "sectors" in this project: the sound, the text, and everything in between. They all present unique challenges and require very different solutions, and so I've made three very loose plans for each of them, graded on the difficulty I foresee myself having.

-sound-

This one is very difficult and very easy at the same time. In terms of composing the sfx and pieces that will be played in the final product, no problems there. I know what I am capable of as a musician and won't struggle to finish that aspect. The main difficulty with the sound sector relies on tying it so intricately to the other two. This project is very interlinked with itself. There are no freestanding elements. In addition, in a previous project I made a rudimentary "collect the thing and it plays a sound effect!" simulation, and got most of the grunt work done, but missed a few key elements. The functions checking to see if the things were collected was flawed, and due to that, the sound effect would play over and over each frame as long as the user was within the space where the thing *used* to be. This is a very important issue that needs fixing, especially if I am creating multiple tracks and risking them playing over one another and combining into a terrible musical apocalypse.

The *plan*, then, goes like this.

1. make songs or placeholder songs

2. take old sfx collection code and debug debug debug
3. once old code is working properly, add new checks (instead of is_Thing_Collected it's is_Word_Typed)
4. Implement and debug debug debug
5. ????
6. profit.

-type-

This is the hardest of the three, mostly because of my unfamiliarity. I have done extremely basic work with type in past assignments, but incorporating word recognition and is_Typed and strings and all that is going to require a lot of research and trial and error. This is the thing I will be spending the most time on. In addition to the technical aspect of coding the type to behave the way I want it to, there's also plenty of research to be done on the visuals and making the simulation aesthetically interesting, in addition to technically interesting. For this, I'm looking to typographic artists like Bill Wurtz and the aforementioned Jon Bois.

The plan here is:

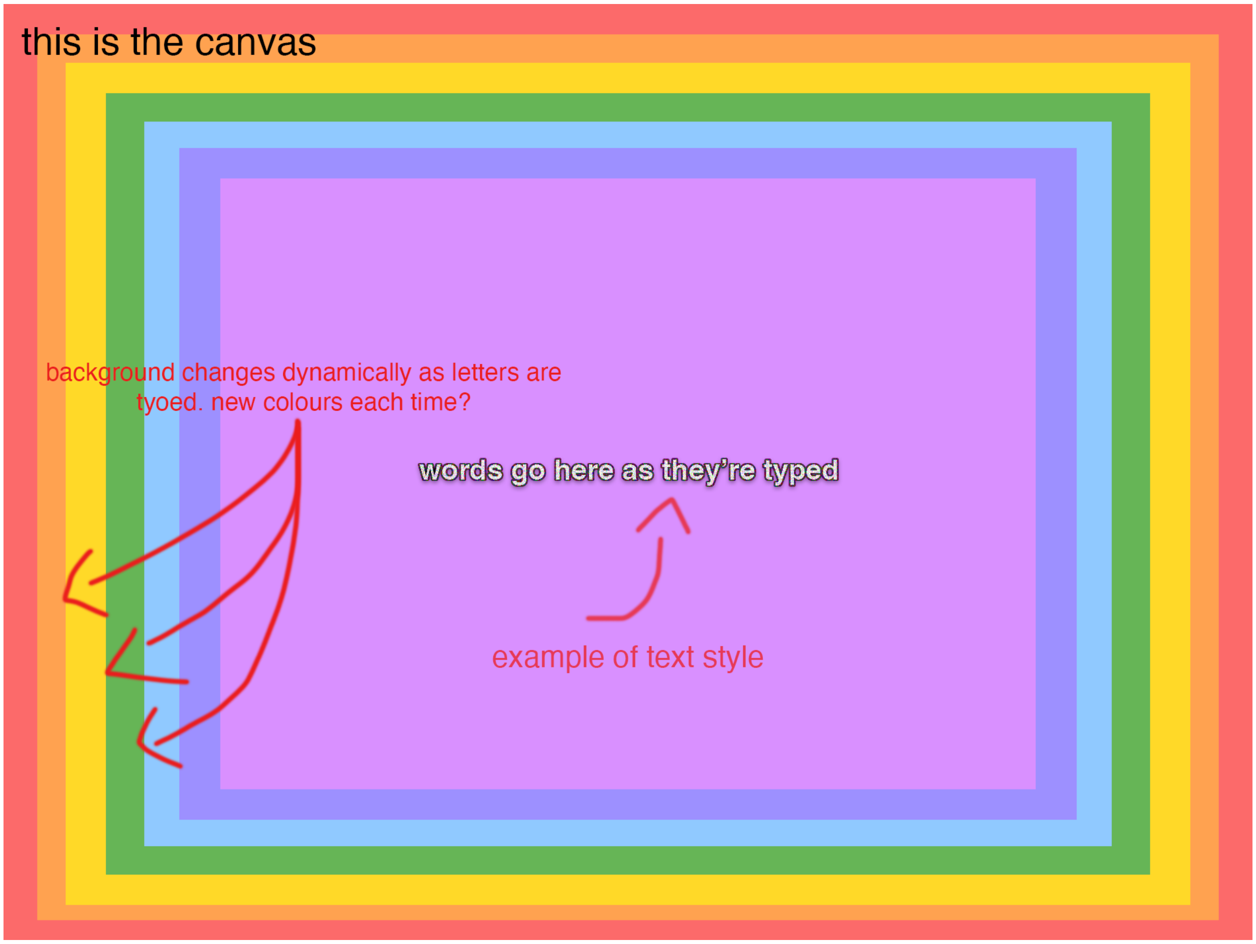
1. research. a lot of it. on various things
2. this prototype! focussing on the text element
3. implementing is_Typed and a dictionary of words (arrays???)
4. implementing interesting visuals
5. debug a lot probably
6. finalize ending state if all words are typed
7. sleep peacefully

-everything else-

It's everything else! All the housekeeping and nonspecific parts of the code that keep the gears turning, as well as other stuff like the background and defining states and all that. I think with this one it'll just be a matter of staying organized and having a clear picture of what I want to do. I don't know how far out of my comfort zone I'll be branching in this particular sector, I just know that this checklist is the sort of "fill in the gaps that I left by messing around with typography code" checklist.

plan!

1. classic template setup. see if there are any old functions/objects that will be useful and don't have to be remade.
2. background creation
3. code organization (class/function/object/etc)
4. everything else
5. die happy



this is the canvas

background changes dynamically as letters are tyoed. new colours each time?

words go here as they’re typed

example of text style