

## Keys Type Mouse - Introduction

Cody Walker

This work is an experiment in digital storytelling. Two stories and a playground. In parallel with this work, I built an engine for creating (small) works of interactive fiction.

My motivation for creating an engine for this project was borne from my frustration with the structures that contemporary interactive fiction authoring tools provide (twine: <http://twinery.org>, inform7: <http://inform7.com>).

The text engine uses love2d (<https://love2d.org>): a minimal framework that's commonly used for small game engines. It acts as a lua (<https://www.lua.org>) binding to SDL (<https://www.libsdl.org>): a library that provides access to graphical and audio hardware. This project was written in lua.

The text engine has multiple components: a system of glyphs and graphs that renders and orders the typographical characters; an editor to move and arrange collections of those characters; a tweening and event system to make those collections of characters into interactable elements, and to animate them.

I wanted to create a text engine that was capable of manipulating typography, words and passages in a programmatic way. My initial approach to the rendering aspect of the engine was to convert a font into an image that I could then parse, character by character, into its constituent grid of pixels. This grid of pixels allowed me to change the properties of each typographical character (scaling, distorting, rendering pixels as ellipses instead of square pixels, etc.). However, this level of granularity proved to be computationally intensive. Manipulating even a couple hundred dozen characters at this granularity was visibly slow, and wouldn't scale well to the hundreds of characters I wanted to be visible.

I moved away from this this approach. Instead of considering each character as a table of points representing the glyph, I opted to treat each glyph of the font as a unit that could be parsed and read into a grid of other characters, that is, a text field.

The text field had its own set of difficulties. Namely, it needed to be able to parse strings (the text for each story, stored as plaintext) into grids. Thus, I needed to implement lines, words, and whitespace in the text engine. These properties are at the core of the text engine, and they account for the variety of textual forms represented in the work.

The other half of the text engine was that of linking the passages of the text to the reader's input.

For "keys", the process was straightforward: each line of text fades and moves up incrementally with each of the reader's key-presses, when it fades out completely, its position is reset to the bottom, and the next line of text populates it.

The system that "mouse" uses is wholly different: as the mouse moves over each typographical character, the engine checks if there is a function. Some functions simply alter the moused-over character, others trigger a timer that fades in or moves the whole field of typographical characters. The writing of "mouse" was a hybrid exercise. I alternated between traditional fiction writing and a script-based programmatic writing to chain together the sequences of textual interactions.

"Computer keys simulate mental processes. These glowing images are [...] images drawn from the brain outward." (Flusser 28)

With "type" I wanted to explore the connection between computer keys and vision in a way that was more exploratory than the linear

as a textual playground for graphemes and computer keys to entangle themselves, and partly to expose the inner workings of the text engine itself.

Vilém Flusser describes the bifurcation of numbers and alphabetic forms as they migrate into digital expression:

"As they migrate from alphanumeric into digital codes, numbers behave differently. They no longer form islands of algorithms rich in complex and creative visionary power, they form heaps that can be picked at". (Flusser 26-27)

In "type", along with each character's respective function (vowels emanating from corners and center, consonants coming from the sides), each numeral, descending from 9, accelerates the movement, bringing it closer to the alien speed of the computer. At 1, the text is almost undecipherable: the letterforms whir into dizzying shapes. 0, however (directly to the right of 9 on most keyboards) freezes the movement of the text, rendering it as a tableau.

The work began with two principal constraints: no visual elements other than the white text over the black background, and no branching story structures. My hope was that the minimal visual aesthetic of the work would create a distance from image-rich computer reading practices. Although there are three parts of the work, the stories themselves do not branch. Each story has only one end - the interactions are not there to suggest an alternate ending, but to encourage a deliberate, contemplative reading practice.

#### Works Cited

Flusser, Vilém. *Does Writing Have a Future?* Ed. Mark Poster. Trans. Nancy A. Roth. Minneapolis: U of Minnesota, 2011. Print.