

Nexus - Design Theory

Shiranka Miskin, Dhruv Lal, Sam Maier, Navin Mahabir, Justin Sacbibit

December 19, 2016

1 Brief Description

Our project aims to create a platform for dynamic and interactive desktop widgets and themes. A simple example is a desktop background that displayed the weather, or a volume slider widget. The advantage we have in accomplishing this is the new open source library Electron, which allows cross-platform desktop applications to be built with web technologies. We would be able to create user interfaces very quickly in JavaScript, and would not have to make significant efforts to be cross-platform, unlike many existing applications.

2 Design Disciplines

The needs of our project are entirely related to software. We will mainly be using JavaScript, which the majority of our team members are already familiar with through co-op experiences. Where we will need to learn a substantial amount is in interfacing with the native APIs of each operating system. Our plan to acquire this knowledge is mainly by going through online resources. We do not have any collaborators, however there is an electron community slack channel where we can periodically ask for assistance.

3 Creating Alternatives

A fitting local analogy to our task is the Rainmeter platform for Windows, which lets users customize their desktops with clickable images and text. Users create their own themes using a custom file format that Rainmeter specifies, and over its many years of existence, there have been a huge number of widgets and themes people have created. What we

would need to create would have to be at least as powerful as Rainmeter, which we plan to achieve through the benefits of using web technologies.

Another possible local analogy is the widgets on Android phones. There are widgets to see the weather, view one's calendar, view their notes or todo lists, or even control applications such as Spotify.

A distant analogy would be that of a command centre. What comes to mind are scenes in science fiction movies where people are looking at complicated screens displaying a number of different stats and buttons. The goal of our application is to make common tasks such as checking the weather much closer at hand compared to having to open a web browser and search it up every day.

4 Selecting from Alternatives

The main choice that we had to make was between focusing our efforts on creating more of a dynamic and interactive desktop background, compared to focusing on separate widgets that one could bring up perhaps by pressing a hotkey.

The dynamic and interactive desktop background path was our initial goal, however the tradeoff was that the functionality in Electron required to implement this only works on Mac and Linux. To succeed we would need to fork Electron and implement this functionality ourselves.

The separate widget path would also adhere to our "command centre" ideal and would avoid the Windows compatibility issue, however after much brainstorming we could not come up with a well designed product idea around this.

We decided to take the risk of having to implement the Windows functionality we need in Electron and focus on creating a platform for dynamic and interactive desktop backgrounds. It would be more difficult, however it would both be a learning opportunity and would allow us to create a product which we can be more proud of.

5 Prototyping

The prototypes we built for this SE390 project were exploratory and horizontal in nature. Electron was unfamiliar to all but one of our group members, and there were numerous unknowns about the feasibility of our idea that we had to confirm. We tested out creating a native node module, as they will be crucial in accomplishing operating-system specific native tasks which we cannot do through JavaScript. We also tested out implementing

multiple simple widgets which use existing node libraries to familiarize our team members with Electron and figure out.

6 Normal and Radical Design

Our project is a slightly normal design as it follows the same ideas as the existing Rainmeter platform. Rainmeter is, however, the only application like this and is more limited in its functionality. Therefore, there could be more radical leaps we could make throughout the project. Our more radical approach is using Electron, which opens up many more possibilities than what is possible with Rainmeter.

7 Laws of Software Design

All the work we put in for this 390 project has been exploratory, and we plan to throw it all away as described by Brooks on Prototyping. Since our plan is for this to be a platform for others to create their own themes and widgets on top of, we will have to work towards designing a very simple and intuitive system for those to be built upon. To accomplish this, we may take inspiration from laws such as Einstein's Razor, Saint-Exupery's Razor, and Hoare's Razor.

8 Understanding Project Risk

The main risk involved is one of skills. In order to accomplish certain native tasks and ensure that our product can be ran on Mac, Linux, and Windows, we will have to learn how to accomplish all of our native tasks on all three of these operating systems. We will also have to learn about the Electron internals and we are putting our trust in our ability to eventually be able to implement the functionality we require that is currently missing in Electron. One way we can mitigate this risk is by contacting folks with experience in these areas, such as an Electron or a Rainmeter maintainer.