Jake Gendreau

CS210

February 19, 2025

**PROJECT PROPOSAL**

**PRODUCT DESCRIPTION**

My project proposal is a Bitmap image filterer made with rust. The app will take in a picture in the format of .bmp, and will allow the user to choose from a number of effects to apply. These effects will include downscaling, upscaling, mirroring in x, mirroring in y, edge detection, blurring, negative, and greyscale. The app will allow the user to make these changes sequentially and as many times as they would like before saving the image to the disk.

**WHY THIS?**

I chose this project because it will expose me to many of Rust's features. I will be working extensively with image processing, which entails 2d arrays, memory management, logic, and data structures. Additionally, hosting the application on the web will force me to learn how to use one of Rust's many web libraries, and how to handle web development with Rust as a backend.

Another reason why I chose this project is to practice my web development skills. I have historically been a web development hater, but after participating at Crimson Code 25 and making a web app with Tavin, I learned that I kind of enjoy it. Plus, it's super easy to show off web applications in portfolios, or just for fun. I will have to make sure that HTML and Javascript are both able to work off of the Rust web server.

**CORE FEATURES**

The core features that I wish to implement are

1) Rust web server

2) Webpage for uploading .bmp images and interfacing with the backend processing

3) Downscaling

4) Upscaling

5) Mirror X

6) Mirror Y

7) Edge detection filter

8) Blurring filter

9) Negative filter

10) Greyscale filter

Furthermore, the web app should be easily accessible and usable on

1) Desktop

2) iOS

3) Android

I split up iOS and Android because web development for iOS is NOT the same as web development

for Android.

Document Design:

**TABLE OF CONTENTS**

1. Components

i. Rust web server

ii. Rust image processing

iii. Website frontend

2. Implementation

i. Rust web server

ii. Rust image processing

iii. Website frontend

**COMPONENTS**

1. Rust web server:

The web server through which the user will interface with the website will be running using Rust as a backend. This will be responsible for serving HTML, JS, and CSS information in a reliable and efficient manner.

2. Rust image processing:

The Rust image processing will be responsible for taking the images that are uploaded to the website, and processing them. It will store the base image as a temporary file, apply filters of the user's selection to this file, and then return the modified file for downloading at the end. The available filters to the user are:

1) Upscale with bilinear interpolation

2) Downscale with bilinear interpolation

3) Mirroring the image in X

4) Mirroring the image in Y

5) Edge detection using the Sobel Operator

6) Blurring with Box Blur

7) Negative filter

8) Greyscale filter

3. Implementation:

1) The implementation of the Rust web server will be done with Rocket

2) The implementation of the image filtering will be done without any libraries so that I really learn how to use Rust and all of its features

3) The website frontend will be very simple, consisting of HTML, CSS, and Javascript to dynamically change the website.

DATAFLOW DIAGRAM BELOW:

Dataflow diagram:

