Project 1.4.7

Project Start Date: November 28, 2016

Project Due Date: December 8, 2016

Task Description:

You work for a software company that is producing a photo editing tool similar to Photoshop® software. The selling point of your flagship product is a large number of detailed algorithmic photo manipulations. Your team is to create a script to manipulate images into one composite in a unique way, ideally one not possible through simple Photoshop workflow. Your team wants the end product of the manipulations to be a photomontage made from at least two distinct original images. They believe that an exciting feature will result if you surprise the user by allowing them to combine two images that do not naturally occur together, especially if one of the images is then changed in some way to emphasize the effect of the combination. You know that your geometric patterns often are a selling point of your software, so your team is also considering how to incorporate geometric shape into the image – drawn on, as a border, or as a mask. Since this script will work as part of a cohesive software package and you have no way of knowing in advance what images a user is going to choose to manipulate with your company's product, you cannot make any assumptions about the images that you will have to work with. However, your team plans to offer a range of options (as a parameter) for one of the image operations you perform to allow the user to customize the operation.

Rough Outline:

To reach our goal, we will need to incorporate ideas from each of the other clients and combine them to reach our product. This will involve elements such as pixel detection, as well as pixel rendering.

Written Proposal:

In a constantly growing world of technology, there is a definite need for new and great software to work with changing needs. We have been assigned to create software that will manipulate multiple images into a single image in a unique way. Our proposed solution is a revolutionary tool that will go above and beyond high end editing software, such as Photoshop. The software combines images by overlapping them and merging the colors. This is done by a series of intricate code and scripts. The software allows for the importing user-designated images. The photos are then resized using a mathematical algorithm. This is important, as it allows both images to evenly altered. Next, the photos are overlayed onto the canvas. This is done by using the paste method. Then the color's of the pixels will be altered by changing the pixels that are overlapping. This software is beneficial to everyone alike. It can be used by teens looking to create unique images to upload to social media, design artists in seek of new and unique tools, and otherwise careers which involve a need for graphics. It is a known fact that Photoshop is currently the leading standard in photo editing software. While this tool isn't designed to overtake this software, it is however, designed to be used alongside it. It offers a much more novice level approach that allows for anyone to understand it and be able to pick it up with ease. Knowing there is great demand for such software, it can make a great investment financially. With a week of work and no cost to develop, the software can be priced competitively. This software will be distributed as a plugin through popular web stores, such as CodeCanyon. It can be used within other applications and website this wish to implement this plugin. It will be available with a commercial license for the price of $99 dollars. This is a fair price considering the amount of functionality this software allows for. This distribution method will be beneficial as it allows for anyone to make use of this plugin, whether it be a small indie developer or a large company. Through this, we will bring in great revenue from a large amount of revenue channels.

Task List:

- Import Images
- Merging Images
- Resize Images
- Merge colors of pixels

Day 1- November 28:

We picked our client and began working on an outline. We also started testing bits of code with small strides.

Day 2- November 29:

We made excellent progress today, as we were able to get Image.Show to work properly. We also have code that allows navigation to the image file. We were able to get a start on pixel color detection.

Day 3- November 30:

The day began with success on merging images. We then began to tackle the issue of merging colors, as well as resizing both images.

Day 4- December 1:

The day began by completing script which would allow for resizing the images. We then implemented the merging images script, which was successful. We then began to tweak the code to allow more functionality.

Day 5- December 5:

The day began where we left off tweaking the code. We then worked on our merging colors script to add better functionality.

Day 6- December 6:

We were able to complete our code and ensure it's functionality.

Day 7- December 7:

We began the day by exploring adding support for .JPG's as well as .PNG's, which was successful.

Day 8 - December 8:

Wrapped things up, made final modifications and submitted files.