Nathan Nickerson Tasks

= DONE

= Started, not complete

= Not started

~~~~~ASSIGNED TASKS~~~~~

-Week 1

 Set up the applet.

Load an image into the applet.

Display a loaded image into the applet.

Be able to load an image with JAI and manipulate the pixels

within the image.

* Interesting RenderedOP documentation: <http://docs.oracle.com/cd/E17802_01/products/products/java-media/jai/forDevelopers/jai-apidocs/javax/media/jai/RenderedOp.html>
* Documentation for PlanarImage: <http://docs.oracle.com/cd/E17802_01/products/products/java-media/jai/forDevelopers/jai-apidocs/javax/media/jai/PlanarImage.html>
* JAI API Tutorial (PDF)

<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CC8QFjAA&url=http%3A%2F%2Fseer.ufrgs.br%2Frita%2Farticle%2Fdownload%2Frita_v11_n1_p93-124%2F3555&ei=Y8zdUY2jDMGmigKCtoDwDQ&usg=AFQjCNEY9Ze1dtZ2x6gTj7C4vtt0QT3h1A&sig2=AgDbWpvjptsJgyzzCc_DOw>

* Remember this: Each pixel has bands. A band represents R (red), G (green), or B (blue).

Add scroll bars to the applet to see the whole image.

Start manipulating a loaded image.

Figure out what color values red eye has.

- I couldn’t find set in stone red eye RGB values.

- I played around with a color chooser and found RGB values

that would make sense for the average person. If a person

had an eye disease, the red eye color would not be red.

~~(175, 0, 0) for the darkest red.~~

~~(255, 0, 0) for the purest red.~~

~~(255, 75, 75) for the minimal red.~~

(Document under the documentation directory with red eye

sample data.)

Create a bounding box for the pixels that contain the red eye

values.

Find the eye in a picture.

* Best way proposed so far is to click the eye and have the user dynamically change the size of the eye or oval.

~~Create a bounding box for the eye.(The user now dynamically chooses where the eye is along with the size of the eye.)~~

* + ~~Make the bounding box a different color than the bounding box for red eye colors.~~

----------------------------Goal for week 1 ends here----------------------------

~~Get the original eye color. (Not needed because red eye focuses inside of the pupil.~~

* Check to see if it is possible there’s some color on the edge of the red.
* Check to see if the other eye is not red. If not, grab that color.

~~Change the redness in the eye to the original color.~~

* Cannot change the red eye to the original color because red eye only exists in the pupil.

Correct the redness in the eye.

----------------------------Stretch goal for week 1 end here--------------------

Week 2 –

Be able to draw in the applet.

* Hold down the left click and drag with drawing happening as the pointer moves through the applet.

Custom Brushes

* Instead of drawing a pixel, have a preset of pixels to be drawn anywhere the pointer chooses.

Make the ability to create lines.

Be able to bend the line created.

* Accomplish something cool!

Week 3 –

|  |  |  |
| --- | --- | --- |
| Date | Day | Description |
| July 8th, 2013 | Monday | * Applet Setup, Load and display image. * Experiment with JAI |
| July 9th, 2013 | Tuesday | * Experiment with JAI |
| July 10th, 2013 | Wednesday | * Experiment with JAI |
| July 11th, 2013 | Thursday | * Start Red Eye Removal |
| July 12th, 2013 | Friday | * Red Eye Removal |
| July 13th, 2013 | Saturday | * Red Eye Removal |
| July 14th, 2013 | Sunday | * Red Eye Removal |
| July 15th, 2013 | Monday | * Finish Red Eye Removal |
| July 16th, 2013 | Tuesday | * Start Drawing Tool |
| July 17th, 2013 | Wednesday | * Finish Drawing Tool |
| July 18th, 2013 | Thursday | * Start Custom Brushes |
| July 19th, 2013 | Friday | * Custom Brushes |
| July 20th, 2013 | Saturday | * Custom Brushes |
| July 21st, 2013 | Sunday | * Finish Custom Brushes |

~~~~~END OF ASSIGNED TASKS~~~~~

~~~~~UNASSIGNED TASKS~~~~~

1. Create a logo and implement it somehow into the application.
2. Be able to restart the project.
3. Make images bigger or smaller.
4. Be able to center text on a graphic.
5. Put multiple graphics into my graphic.
6. Multiple copies (ctrl+c copies) stored to choose from later.
7. Keep track of the history as changes are made to an image.
8. Let the user create custom shapes or custom logos.
9. Add shadows to a 2D picture.
10. Work on multiple areas on a picture?
11. Have the ability to zoom for precision.
12. Draw straight lines.
13. Be able to turn on a “snapping” effect.
14. Create anti-aliasing for image resizing or for a smoother graphic.
15. Different stroke types for painting.
16. Make any graphic sharper or put different effects onto it.
17. Save as multiple file types.

~~~~~END OF UNASSIGNED TASKS~~~~~

~~~~~Things to keep in mind~~~~~

~~~~~End of things to keep in mind~~~~~

~~~~~Meetings~~~~~

1st Meeting: July 10, 2013 – Need to make more realistic goals with some stretches to know if the work is good enough. Need to re-word vague tasks for example: “Dig into JAI” could be, “Be able to load an image with JAI and manipulate the pixels within the image.”

* Need to put project files into the code directory and use .gitignore to ignore any unwanted files to commit.
* No status color for this week.

2nd Meeting: July 17, 2013 – 15 to 17hrs put in this week. Put in more hours next week. Learn to either step over, or walk around brick wall. Putting more time could have changed what I delivered this week. Think towards user stories (Agile) instead of specifications (Waterfall). Waterfall commits too much towards this and has to be accomplished in that specific way. Think out different solutions and why not try them. Commit to drawing a line and the Bezier curve manipulating the line. Make something cool! Finish the red eye and do not deliver something that paint could do (Spray Can). Trying to recreate something another application already does is okay for Capstone, but more innovation is needed in industry, and recreation is not as okay there.

* Hours: 15-17hrs.
* Status: Yellow (Room for improvement, but have not lost hope)
* Technicality for accomplished: Moderate
* Professionalism: Dressed well and minimal stutters.

~~~~~End of Meetings~~~~~