**SPRINT ONE**

Project Manager: Dr. Blanche Cohen

Scrum Master: Jacob Watters

Product Owner: Dustin Shaver

Team Members: Omar Bitar, Francesco Limoni, and Nick Miller

Introduction

Team 3 is creating a simple drawing program we are naming “Pixel Image Editor” (or PIE for short). It is coded in Ruby, using the FXRuby gem libraries. This program is written specifically for a desktop computer or laptop, not a tablet or mobile device. The GitHub repository link is: https://github.com/FrancescoLimoni/Pixel-Image-Editor.

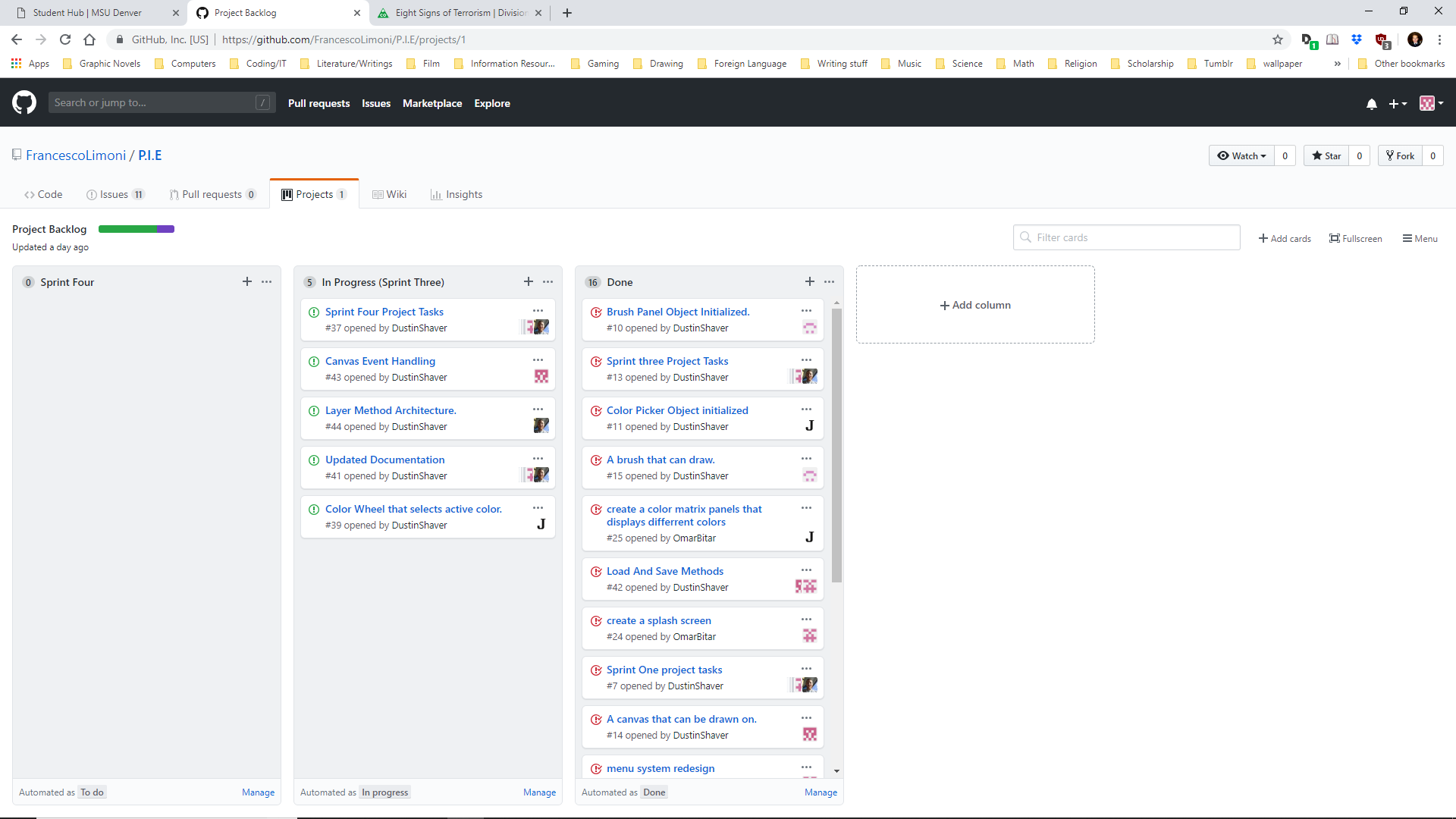
Research Progress and Project Demonstration

The program demonstrated today is the result of our sprint goal to update and complete functionality with regards to the saving and loading of pixel data. A great deal of research was required on the effective use of event handlers and how Windows interacts with applications using the ruby compiler. A series of event handlers were created to handle save and load buttons and to link the about button in the menu to our github account. Inputs were created to set the active color using RGB values. The integration of this functionality brings the P.I.E. application into the final stages of development. Each team member worked solely on creating their own “module,” each of which were combined at the end of the Sprint. The modules are listed on the next page.

* **Canvas Module** – The blank canvas where the user will be able to draw. The canvas is resizable with can be saved as a PNG image file.
* **Brush Module** – A side bar with different “brush sizes” for the pixel editing. This ranges from using a brush that is 1 pixel by 1 pixel to some larger size. The buttons are not currently implemented.
* **Color Module** – A side bar with preset color palletes and RGB text inputs for color seting. Eventually, this will be where the user can define their own colors.
* **Toolbar Module** – The toolbar across the top of the screen where the buttons to Save and access program documentation are.
* **Layering Module** – A side bar where the user will organize the different layers to their pixel project.

All of the modules mentioned above then communicate through a Main file, where they are formatted using frames provided in FXRuby. The Layering Module was considered extra for this week, not part of our main priority.

During the Sprint 4 presentation, the Project Backlog was described, and the successes and failures of the team to meet those tasks. On the next page is a screenshot of the Project Backlog.



Comparison to Project Plan

Our current version is close to our program concept (the concept art is shown on the next page). Advancements in design and layout have made the program more stylized and functional. So far we are on track to finish the pixel editor by the end of the semester. There are no current plans to scale back the project.

|  |  |
| --- | --- |
| https://media.discordapp.net/attachments/539956456070578190/540601305069256704/team3ProjectConcept_1.png?width=1179&height=664  Concept art | https://cdn.discordapp.com/attachments/539956456070578190/545354495803129856/unknown.png  Version 1.1 |

Sprint Retrospective

This last Sprint was not as organized as we would have liked, but our team is finding it very difficult to find times where we are all available to meet. Inclement weather and conflicting schedules has made coordinating difficult. The team is largely de-centralized, and the trust in our fellow team member to meet their backlog despite the communication difficulties has been largely successful. As the simpler tasks get accomplished more quickly the work load has become very uneven, with some team members logging much longer hours than others. Future sprints will attempt to more evenly assign work to increase productivity of the team.

Most things went well for the team during the fourth sprint. Despite the necessity of major design changes to the canvas class to account for user data storage, the restructuring was relatively smooth with no major bugs or problems generated.

A potential improvement our team could make is to have clearer documentation and updated comments in our individual files. As more of the work is shared on in the same files the ability to understand team members code is paramount.

Our plan for implementing improvements is to dedicate more work time to formatting and readability. Our team works very well together and can often find solutions to potential tasks very quickly and clearly. We look forward to implementing future functionality in sprint five

Hours

|  |  |
| --- | --- |
| Team Member | Hours Worked for Sprint 1 |
| Omar Bitar | 6 hr 0 min |
| Francesco Limoni | 5 hr 0 min |
| Nick Miller | 9 hr 0 min |
| Dustin Shaver | 17 hr 0 min |
| Jacob Watters | 22 hr 0 min |

Tools and Resources

Language used: Ruby - ruby-lang.org/en/

Gem for GUI used: FXRuby - fxruby.org/

IDE for Ruby: Aptana Studio 3 - aptana.com/

Documentation for FXRuby: Online - rubydoc.info/gems/fxruby/Fox/

Book on FXRuby: FXRuby - https://media.pragprog.com/titles/fxruby/tables.pdf

Guide for Scrums and Sprints: The Scrum Guide - https://www.scrumguides.org/docs/scrum

guide/v2017/2017-Scrum-Guide-US.pdf

Repository Hosting: GitHub – github.com/