**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

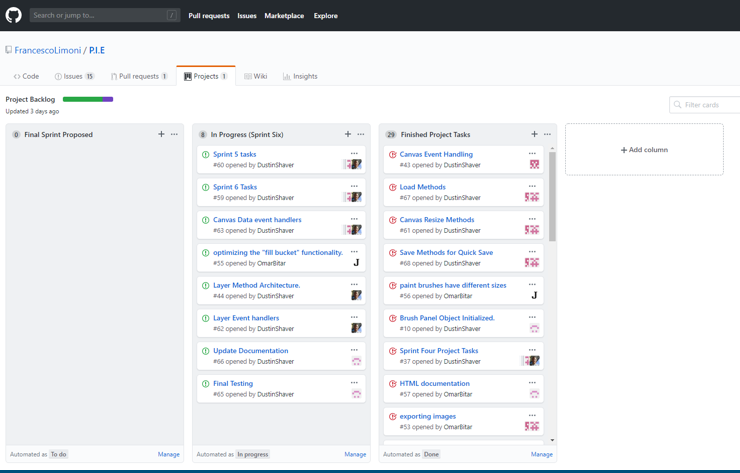
Team 3 is creating a simple drawing program we are naming “Pixel Image Editor” (or PIE for short). It is coded in Ruby, using the FXRuby gem libraries. This program is written specifically for a desktop computer or laptop, not a tablet or mobile device. The GitHub repository link is: https://github.com/FrancescoLimoni/Pixel-Image-Editor.

Research Progress and Project Demonstration

The program demonstrated today is the result of all work done on the program during the semester. Most of the time this sprint was spent on getting the program ready for final demonstration. This included bug fixing, extensive testing, creating a presentation plan and practicing that plan, as well as updating all the documentation for P.I.E. We also updated the HTML help file that is opened when a user clicks on the about button in the program. An important task that we got completed this sprint was the ability to load in PNG files into the program. As a bonus, were also able to implement a fully functional color wheel and RGB sliders. Our team was also able to create and run a Docker image of the program which was a big success in terms of the deployment of P.I.E

* **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 pallets and RGB text inputs for color setting. 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.



Comparison to Project Plan

Our Final version meets all the original goals set forth in the initial project proposal and even contains some extra bits we had time to implement. As a group we made a decision to try and improve the fill brush before the final demonstration but due to time constraints this complex, optional task did not get completed. Members mostly worked on their own modules but also collaborated with other team members on different aspects of the program.

|  |  |
| --- | --- |
|  |  |

Sprint Retrospective

This sprint was well executed. The team members all communicated effectively and many of them met outside of class to discuss the project. I am proud to say that the debugging and testing was extensive enough to ensure there are no major unknown bugs. Any bugs that still exist in the program were deemed arbitrary or too complex to fix in a limited amount of time. An example of this is the way the layering works in the program. If you show a layer you won’t be able to see any layer’s underneath because the program does not support transparent colors.

Most things went well for the team during the final sprint. We completed all mandatory task set forward only not having time to complete the fill optimization as it was realized to be much more complex than originally thought.

Our team updated a lot of documentation but one thing we could have done better was come up with a easy way for a user to download our program.

We have all learned a lot from this process and will take these lessons into our next classes and into our careers.

Hours

|  |  |
| --- | --- |
| Team Member | Hours Worked for Sprint 1 |
| Omar Bitar | 15 hr 0 min |
| Francesco Limoni | 7 hr 0 min |
| Nick Miller | 5 hr 0 min |
| Dustin Shaver | 13 hr 57 min |
| Jacob Watters | 28.1 hr |

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/