**SPRINT ONE**

Project Manager: Dr. Blanche Cohen

Scrum Master: Dustin Shaver

Product Owner: Nick Miller

Team Members: Omar Bitar, Francesco Limoni, and Jacob Watters

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

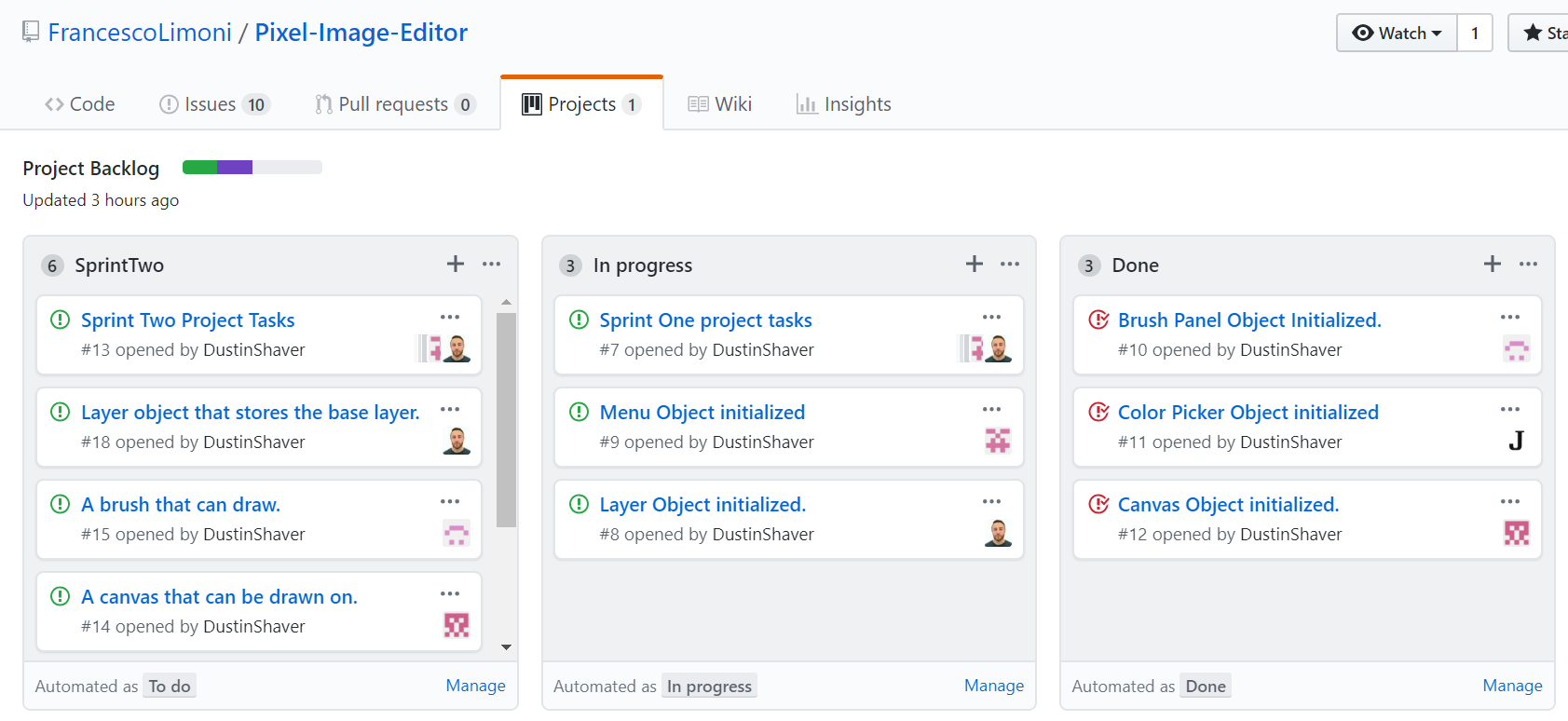
Each team member researched for this project individually depending on their experience. Everyone learned basic syntax in Ruby and looked for libraries created for GUI applications. Eventually the team settled on using FXRuby because it was easiest to install and find documentation for online.

The program demonstrated today is the result of our sprint goal to create a skeleton of the pixel editor. This was planned to be an extremely simple framework on which features of the pixel editor would later be placed. Formatting was not important for this Sprint. The priority was having methods defined for each feature and displaying placeholders in a GUI environment. 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. Eventually the canvas will be resizable and the user will have the option to zoom in or out.
* **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, which is currently not decided upon.
* **Color Module** – A side bar with some form of color picker. Eventually, this will be where the user can change the color of the brush they are currently using, and 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 1 presentation, the Project Backlog was displayed from GitHub on the screen. On the next page is a screenshot of the Project Backlog.



Comparison to Project Plan

As is expected, our current skeleton is nowhere near as impressive as our program concept (the concept art is shown on the next page). FXRuby has a legacy graphic style. This was not necessarily what we expected when we started programming, but we like it. 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. When the team can meet, it is never everyone at once. It is not that big of a problem because the group members are understanding of jobs and other classes. It took a long time to get FXRuby to work for each of us, and now that the installation process is complete, there will be more time to deal with debugging and formatting errors.

Most things went well for the team during the first sprint. Downloading the necessary gems and programs was a longer process than it should’ve been, but creating the framework after getting used to the FXRuby notation worked out well for us. The research was really easy and there is a lot of documentation online. Also, integration of all of our separate files was simple enough. Overall, we achieved our goal of displaying the skeleton of our program on the screen.

A potential improvement our team could make is to have better communication in general, even though things worked out this time. Also, all of our modules should be organized on the screen in a more user-friendly way. That work will be part of Sprint 2.

Our plan for implementing improvements is to perfect our communication. Our team gets a lot of work done and can normally stay on-task pretty well. But when we are apart, we forget to tag each other on Discord and forget to check our notifications. With better communication, our Sprint will be even more productive next time.

Hours

|  |  |
| --- | --- |
| Team Member | Hours Worked for Sprint 1 |
| Omar Bitar | 34 hr 0 min |
| Francesco Limoni | 15 hr 0 min |
| Nick Miller | 18 hr 30 min |
| Dustin Shaver | 22 hr 30 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/