University of Idaho
College of Engineering

MEMORANDUM

To: Bruce Bolden

From: Joe Leister

Subject: Second Sprint (Assignment 6)

Date: April 11, 2017

Summary of Team Efforts

While communicating via video conference is not ideal (The sound tends to go from I can't hear it to blowing out my ear drums in the meeting room we have reserved for the semester. It appears to be random and I have no idea why or how to fix it. Lectures in the classroom seem great though) I am pleasantly surprised at the group. I am used to one or two people pulling the weight while the rest of the group sits back and enjoys the ride. I was really expecting something similar on this project and I kind of expected it to be even more exaggerated because of my physical distance from the group. Instead I feel very included in the conversations and I feel my comments are valued by the group. Also, I feel the project is being pretty evenly divided. I am not sure how the GitHub repo looks like it is being divided with commits and what not but, those that may not have as many lines of code in the project make up for it with documentation and input in the group/Slack discussions. One great thing that I experienced this sprint was feeling overwhelmed at the tasks I took on. Without saying anything to the team I think they felt it because at our last scrum, people just volunteered to do little bits and pieces of hooking everything together and getting stuff working that it took a huge load off my efforts.

My Two Cents

My main accomplishment this sprint was to read in the frames from a tan file and create icons of the output so we could have a preview of each frame and be able to edit it. I started off with the idea of writing thumbnail images and setting them as the icons for the frame buttons. This didn't seem efficient as there was no need for the files except during runtime and that is a lot of IO to write to disk for temp files. After branching the repo to work on this feature I came up with the idea to just take the bufferedImage and cast it as an imageIcon since if I wrote it to disk I would have to read it back into a bufferedImage anyway. This saved code, IO read and writes, and helped me better understand how

bufferedImage class worked. Other accomplishments this sprint, I added code to color the nodes in the grid with the appropriate value when a tan file is opened so you can see the current frame in the grid editor as well as the previews for other frames. I also got the team to switch to using a tanFile class with read and write methods since we had been using a readFile and writeFile class and using the constructor to do the reading and writing. We didn't have any issues except the fact that we were just trusting that since Java was pass by reference that we could continue to access the data. After some discussion I was able to convince the team that we wanted to explicitly return the reference back to the program so that it stayed in scope and garbage collection didn't release it and end up corrupting the data.

Appendicies

Design Specification Document

The current complete version of the Design Specifications can be accessed at our team Github page via the following link. https://github.com/GoofyGlasses-CS383-S17/Design-Specification/blob/master/Working%20Draft/designspec.pdf

Updates

Below is the updates made during this sprint the full size PDF of changes is located in the team's notebook.

Updating Design Specification for Second Sprint Report

Team 3 (Java)

Contents

A Changes/Updates

The changes made to the Design Specification was relatively small. These changes are listed below.

A.1 Updating the Diagram for Grid Editor

The change made to the diagram for the Grid Editor was to reconfigure the diagrams to accurately represent the current build of the system for the second sprint. The Color Class was removed from the diagram as the team opted to utilize Java's own built in Color Class. Two new classes were added in the diagram, a FrameButtonActionListener class, which does as it is named, and utilized to handle actions made to the "Frame Button" in the GUI. The other class was the NodeActionListner, which also performs as it is described, handling the button press of a node in the GUI. These diagrams thus were updated and reflected in the updated Design Specification for Sprint 2.

A.2 Updating the Timeline

The timeline was edited to reflect the progress the team has currently made since Sprint 2 and also display the goals the team hopes to accomplish in the following sprints. Originally, we had not exactly planned for a frame editor (which include the frame preview bar), but that is now reflected in the timeline, but since it is not done, it was moved to be completed at a later sprint. The Mulit Node Editor is the next goal as well, since the Single Node Editor is fully functioning now, as well as most of the GUI elements. The timeline reflects the idea that due to the success of the team's progress, there will be additional time to add or edit features, however this is not set in stone as difficulties in the future may arise.

B Added Section(s)

Only one section was required to be added to Sprint 2, which was the TAN file section. These section(s) are further discussed below.

B.1 TAN File Section

The TAN file section was only just implemented into the Design Specification under Design Decisions due to the team not having the full picture of the requirements of the TAN file until the end of the last sprint. The TAN file section briefly explains the purpose of the TAN file, as well as providing a brief description of the components that make up the the TAN file. These components are then explained to be utilized so that the editor can read in files, and correctly output them to the GUI, as well as having the editor write out to files, allowing users to save these TAN files. The section was fairly extensive at explaining the properties of the TAN file and its functionality within the scope of the project.

C New/Changed Diagrams for Update

The Following are diagrams were created/updated for relevancy in the updating the Design Specification for the second sprint, these diagrams are presented in the following sections below.

C.1 Class Diagram for Grid Editor

The Class Diagram below in Figure displays the structure of the Grid Editor (so far), in which the Color, Node, and ActionListener classes holds the main mechanism of the Gird Editor, also showing the relationships between these classes. This has been updated from the previous sprint, by removing the Color class (utilizing Java's default Color class) and adding the NodeActionListener class and FrameButtonActionListener class. The diagram is shown in Figure 1 below.

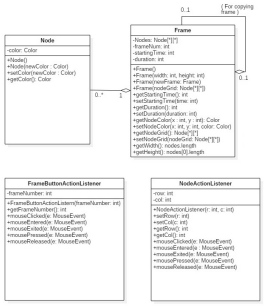

Figure 1: Class Diagram of Grid Editor

C.2 Frame Preview Bar Diagram

The diagram in the figure below displays the structure of the Frame Preview Bar, which contains a list of frames, which have images to show the frame's configuration. There also exists a scroll bar as shown, which allows the user to scroll to show other frames. This diagram was mainly added to present the user with a visual cue to how the Frame Preview Bar should function. The diagram is shown in Figure 2 below.

Figure 2: Diagram of Frame Preview Bar

C.3 Grid Editor Diagram

The diagram in the figure below displays the structure of the Grid Editor in regards to the second sprint. This was shown to display the progress made on the Grid Editor GUI since the first sprint. The Grid Editor now displays a color when the user enter the RGB values of that color in the node dialog box, This effect was supposed to be focal point of the diagram. The diagram in shown in Figure 3 below.

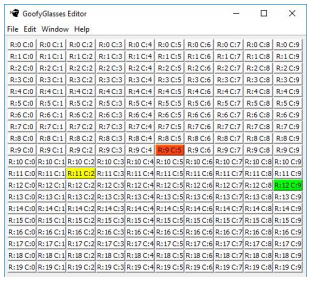

Figure 3: Diagram of Grid Editor