Retrospective – Project 3

Official Fortran Fanclub

Meeting Log

1. Date: Monday, October 8, 2018

Location: In-class

Attended: All members

Outcomes: Set meeting time for Monday Oct 8 meeting. Discussed basic idea for project, discussed which functions needed to be made before next meeting.

2. Date: Monday, October 8, 2018

Location: Spahr

Attended: All members

Outcomes: Discussed prototype game features and final game features, began Gantt chart, divided workload, classes, and decided on prototype song choice.

3. Date: Wednesday, October 10, 2018

Location: In-class

Attended: All members

Outcomes: Set meeting time for Friday, Oct 12 meeting. Discussed progress made since Monday meeting including beatmap generator and button class.

4. Date: Friday, Oct 12, 2018

Location: In-class

Attended: All members

Outcomes: Discussed progress since previous meeting (none), did not get much done since we would be meeting later.

5. Date: Friday, Oct 12, 2018

Location: Spahr

Attended: All members

Outcomes: Made all UML charts, started Gantt chart, briefly revisited workload distribution.

6. Date: Wednesday, Oct 17, 2018

Location: In-class

Attended: All members

Outcomes: Discussed progress over break. Set meeting time for out of class Wednesday Oct 17 meeting.

7. Date: Wednesday, Oct 17, 2018

Location: In-class

Attended: All members

Outcomes: Began initial integration testing and discussed next steps.

8. Date: Friday, Oct 19, 2018

Location: In-class

Attended: All members

Outcomes: Discussed progress since last meeting (Game menus, game logic, beatmap improvements, button improvements) and set meeting time for out of class Friday Oct 19 meeting

9. Date: Friday, Oct 19, 2018

Location: Spahr

Attended: All members

Outcomes: Integrated menu and gamelogic classes, improved gamelogic functionality, worked on UML diagrams (class diagram, use-case diagram).

Workload Division

Sydney Combs

* Game Logic
* Timing

Daniel Gonzalez

* Button classes
* UML diagrams

Daniel Hidalgo

* Beatmap Generator
* Gantt chart
* Documentation generation

Nathan Pelletier

* Menus
* Product and Sprint Backlogs

Challenges

Sydney Combs - I was responsible for the game logic portion, which meant I had to convert the beatmap and deal with timing. Converting the beatmap was difficult only because of unexpected issues (easily remedied) with reading from file. Timing was confusing, but I came up with a solution based on mapping notes to a predetermined location.

Daniel Gonzalez - My portion of this project involved using mostly methods we had used in a previous project, so the main goal/challenge for me was to ensure that my code was straightforward for the other members to use.

Daniel Hidalgo - The biggest challenges were a result of being thinking too imperatively. Haskell requires such different thinking and I use it so rarely that it took some getting used to again. Every issue was my own fault… but besides Haskell, the only thing worth mentioning was writing my own wrappers for Python and C.

Nathan Pelletier- For this project, the main problems I had were both the creation of the menus as well as the navigating between them. I couldn’t just open up a new menu every time because each new menu would open a continuous while loop, which would be extremely inefficient. My first attempt at this was to make one outer loop, and then with each botton create an inner loop. When they press back, instead of creating a new loop it just ended the inner loop and brought it back out, but I had a few problems so instead I made it one loop that used an array of menus which it alternated between based on the selections. It worked, however that brought me to one of the parts that bugged me the most because apparently Python doesn’t have global variables, so it couldn’t increment my global variable to keep track of what menu I was on. I solved this by placing my counter inside an array of only one value, which it is able to store and change correctly.

Unused Features

· Difficulty implementations, osu mode, scoring multipliers, hold trails, polish, background image(s), note images.

Retrospective

· Sydney Combs - I feel like I did a better job than I expected of myself, but I also know that this is about as far from my best work as possible. I would rather the functions I wrote make more sense than they do. I feel like gamelogic generally looks messy, and in the future I would like to improve my work.

· Daniel Gonzalez - I would have liked to make the buttons look nicer, but there is still time to add a graphic for project 4 and it was not a necessity for the prototype. I do not like how many changes I had to (and probably will still have to) make to the buttons to account for unforeseen difficulties such as allowing for arguments in function calls. I may end up switching to a different module such as Tkinter which seems better suited to the menu buttons and which might be easier to extend upon. I feel like my contribution to this project was much less than previous projects, and I would like to make up for it in project 4.

· Daniel Hidalgo - My original inclination was to something difficult (grown up signal processing), but then I realized that would take a lot of work and went with an easier version of the idea. After deciding upon a new approach I began reading up on different languages to quickly test that approach. The three options were: Python with scientific packages, MATLAB, and Octave. Because of my familiarity with Python I chose to avoid it and chose to go with Octave due to its portability.

I tested my approach by listening to the results and they seemed to make sense, but getting visual confirmation was exciting once we did an actual test run. As much as I don’t want to make games, seeing a project come together like this was enjoyable.

· Nathan Pelletier- I am honestly rather happy at some of the solutions that I managed to find to help my code be more efficient, however I believe that there are still a lot of improvements that can be made. First, I think that I can definitely do better in terms of how I format the menus and change the button size and text size according to their text. I would like to make something that will automatically do that for each of the menus so it is easier to use, and would also like to do some formatting for the background to make it look more interactive. In addition, I think that I might have to use some form of Tkinter for project 4 to help with menus that include more buttons, since right now I’m limited in the number of buttons that I can put on my screen.