

Sound Swarm

[HTTPS://GITHUB.COM/SUBLIMINALJ/SOUNDSWARM](https://github.com/SubliminalJ/SoundSwarm)

Jay Hixson
ACM, IEEE

Revision History

Last User to make Changes	Date	Summary of Changes
J. Hixson	02/03/2018	Initial Release
J. Hixson	02/17/2018	Completed Sprint 1
J. Hixson	02/18/2018	Planning for Sprint 2
J. Hixson	3/3/2018	Completed Sprint 2 / Planning for Sprint 3
J. Hixson	03/17/2018	Completed Sprint 3
J. Hixson	03/18/2018	Planning for Sprint 4
J. Hixson	04/08/2018	Completed Sprint 4
J. Hixson	04/22/2018	Completed Sprint 5 / Planning for Sprint 6
J Hixson	05/06/2018	Completed Sprint 6

Table of Contents

Vision Statement	4
Requirements	5
Product Backlog.....	6
Sprint #1	8
Review	9
Retrospective	10
Sprint #2	11
Review	11
Retrospective	12
Sprint #3	13
Review	14
Retrospective	16
Sprint #4	17
Review	18
Retrospective	20
Sprint #5	21
Review	22
Retrospective	23
Sprint #6	24
Review:	25
Retrospective	26

Vision Statement

For music fans, DJs, and producers who want to mix, create, and share music with the world, SoundSwarm is an android app that allows users to create and share multitrack audio creations in an easy and intuitive way.

Goals:

Produce a launch ready multitrack audio program for android in one semester.

Create a website with demo videos and DJ testimonials before release in two years.

Create and launch custom sound banks from featured musicians and producers in two years.

Scope:

Included in the project are the sounds, samples and tones generated by the app, along with the means to combine them to make portable sonic creations to share or perform live.

Essential features:

Add Music of your own

Organize the sounds to create mixes, or soundscapes

save and restore creations

Use provided sound effects and soundscapes

Create sounds and tones using the app

Desirable features:

Access music from the Google Play Store

Access youtube videos

Generate Binaural Beats

Record mixes as .wav files

Requirements

DJs - Users who are familiar with beat matching, track juggling, arranging, and pitch-shifting songs.

Musicians - Users who are familiar with notes, intervals, chords, melodies and song writing.

Podcasters - Users who have radio shows or live stream with sound banks

ACTOR	GOAL	
DJ	import songs	
		from storage
		from google play
		from youtube
	Pitch shift songs	
	Loop audio	
Musician	Create song structures	
	Save and Restore mixes	
Podcasters	play samples	
	allow live input from mic	
	save session as audiofile	
	save session as multitrack audio files	

Product Backlog

Story ID	Story	Story Points	Priority	Status
S15	As a DJ I want to be able to have an HUD for each track and see playback settings	15	5	Complete
S1	As a DJ, I want access to my song collection	25	5	Complete
S2	As a DJ, I want to be able to loop Sounds	3	5	
S6	As a DJ, I want to be able to control the speed of songs	7	5	
S7	As a musician, I want to be able to audition sounds in the app so I can pick the right one	15	5	Sprint 5
S8	As a musician, I want to be able to save and restore my work.	50	5	
S14	As a DJ, I want to be able to play more than one sound file at once	50	5	Sprint 5
S5	As a DJ, I want to be able to add a track from Google Play	10	4	
S10	As a podcaster, I want to be able to export my work as a single audio file	25	4	
S11	As a podcaster, I want to be able to use the microphone over music	25	4	
S12	As a podcaster, I want to be able to play samples during recording	15	4	

S9	As a musician, I want to be able to export my tracks into individual files.	25	3	
S3	As a DJ, I want to be able to set and change loop points	25	1	
S4	As a DJ, I want to be able to add a track from youtube	15	1	
S13	As a podcaster, I want to be able to export shows with graphics or video	250	1	

Sprint #1

Sprint Backlog

Story ID	Story / Task	Estimated Hours	Actual Hours
S7	As a musician, I want to be able to audition sounds in the app so I can pick the right one	15	
	Design UI for file selection	3	
	Add UI functionality	2	
	Implement file chooser	5	
	TEST	5	
S1	As a DJ, I want access to my song collection	25	15
	research file browser interface	5	5
	create a file browser activity	5	2
	research audio APIs	5	2
	test file browser	5	1
	create Sound SwarmNode class	1	1
	create Swarm Selector Activity and layout	2	3
	test Swarm Selector Activity layout	2	1

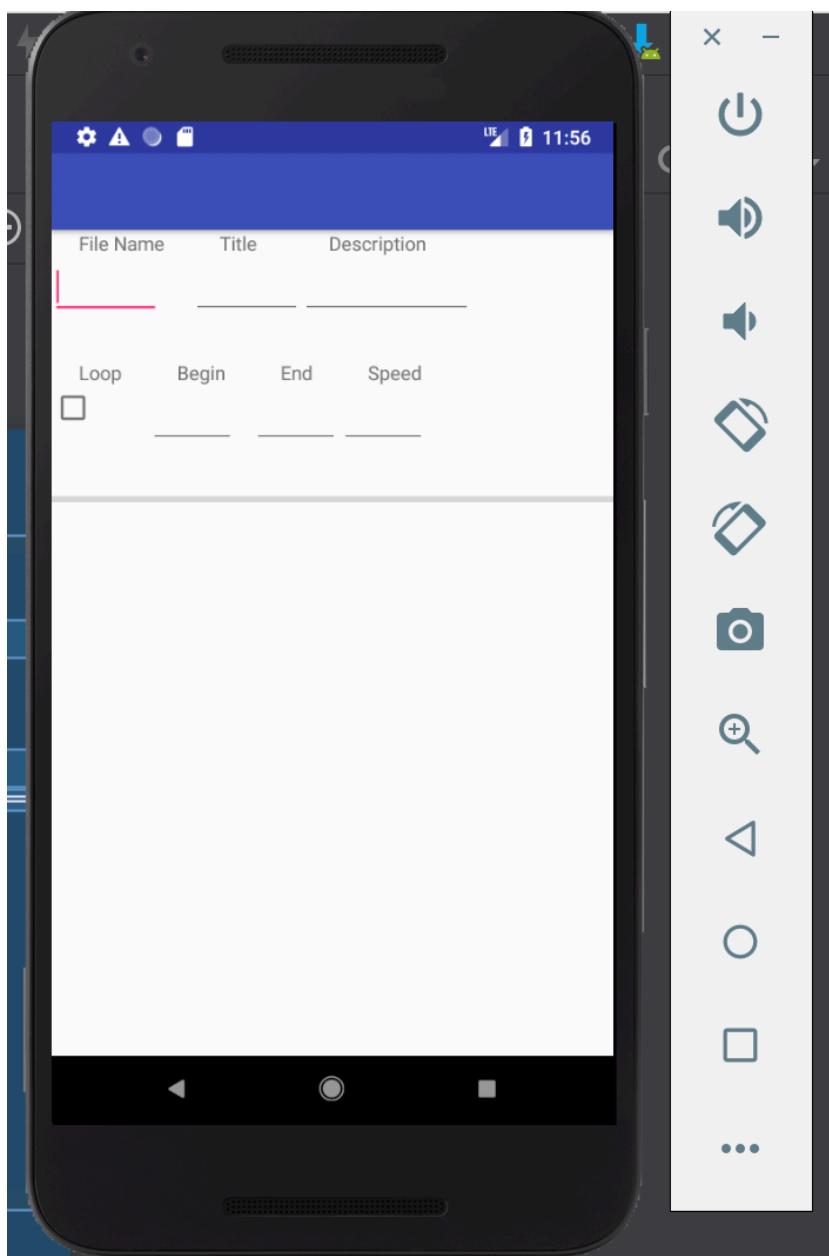
Review

We bit off a bit more than we could chew with this first story. Building an interface was about all we could manage with the file explorer code we tried not working out. We tried many different ways to get a file browser going, and in the end chalked it up to research time that didn't pan out.

We plan to go with the file access methods in the Big Nerd Ranch guide in the future instead of trying to build a file browser which may not be necessary.

Did lots of research and I think we will need to use sound pool for this app, although Google OBOE looks like it is a great wrapper for some advanced audio tools and may come in handy in the late stages of this product. So far there was no luck setting up the c++ libraries to do that, but spent a couple of hours working on it. The project I think will need to be restarted from scratch to do that.

In summary, we managed to do the layout and spent the rest of the time trying to use code that didn't work out.



Retrospective

No completed stories this time, so this first story will need to be broken up into smaller stories. Completed a build with a template for audio data but no functionality.

The first story was a little ambitious but it seemed the logical starting place, since you can't make music without loading any files.

Project velocity: 0 (15)

No stories completed so no velocity. Although gaged effectiveness at about an hour a day of work on average totaling about 15 story points per sprint is possible.

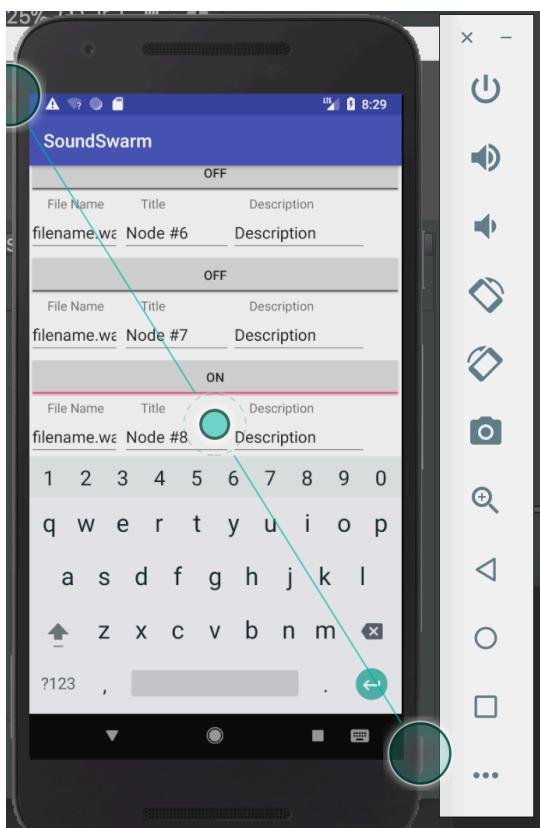
Sprint #2

Sprint Backlog

Story ID	Story / Task	Estimated Hours	Actual Hours
S15	As a DJ I want to be able to have an HUD for each track and see playback settings	15	5
	Make SoundSwarm Selector Activity a fragment	2	2
	create a recycler view	3	3
	create cards for each Swarm Node to put in recycler	5	
	test and refine layout of Sound Swarm Selector layout	5	2

Review

'We' made more progress this week but didn't complete the task with the cards. Made a fragment layout which will be more versatile and completed the recycler view with a fragment recycling but not cards. I think that the overall design needs to be more thoroughly fleshed out. Having all the sounds in a list that scrolls is great, and all the bindings for the data involved are working so the titles and descriptions are loading.



Retrospective

No completed stories this time, so this story will need to be finished next sprint. Completed a build with a list of SwarmNodes but doesn't link to the recycler view with cards because it's not complete yet. Lots of coding this time with guidance from the text and results were surprisingly effective. What we accomplished wasn't in line with the story we were doing but the only changes that are needed are to either edit the layout file and copy in the layout from the first sprint.

Project velocity: 0 (7)

No stories completed so no velocity. Accomplished less than half of what was being targeted due to exams and time not available to work on the project.

Sprint #3

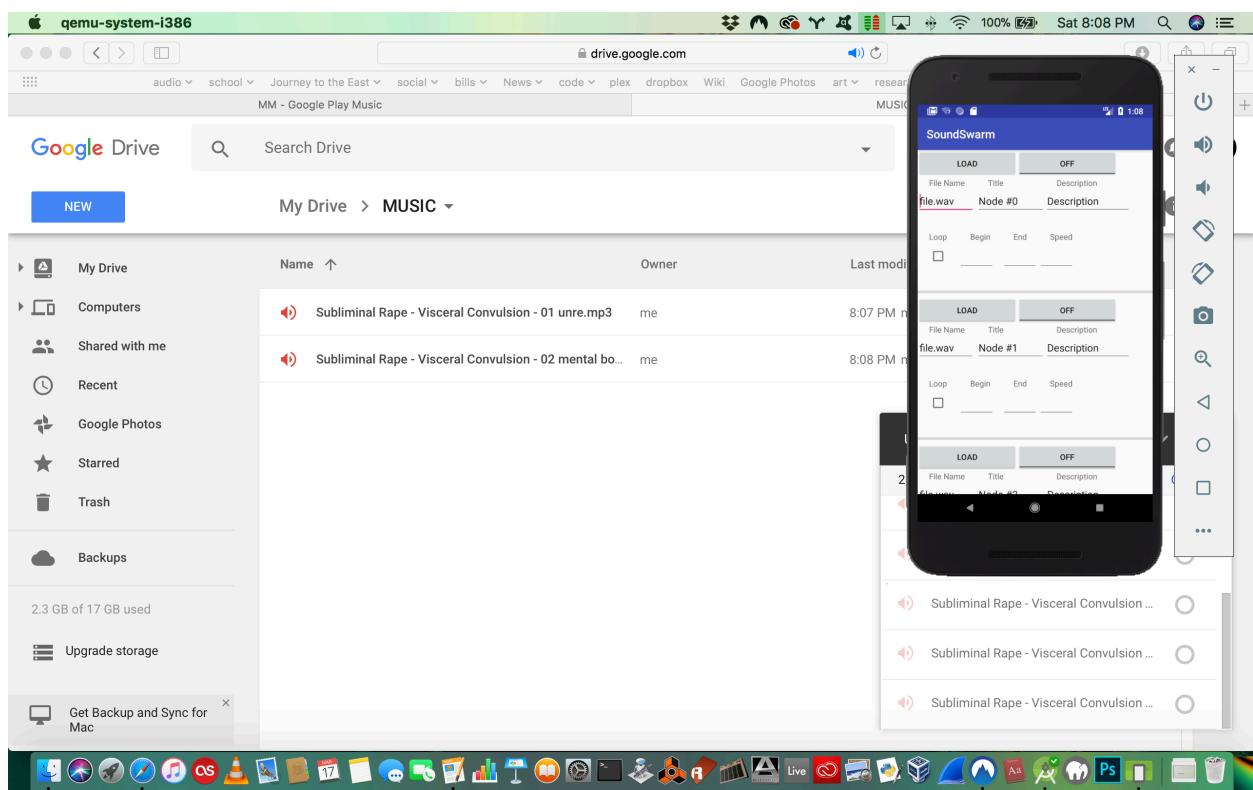
Sprint Backlog

Story ID	Story / Task	Estimated Hours	Actual Hours	Earlier Sprints	Sprint 3
S15	As a DJ I want to be able to have an HUD for each track and see playback settings	15	12		
	Make SoundSwarm Selector Activity a fragment	2	2	2	
	create a recycler view	3	3	3	
	create cards for each Swarm Node to put in recycler	5			
	Update Recycler view to show all options	2	1		1
	Update bindings and inflation for recycler	3	1		1
	test and refine layout of Sound Swarm Selector layout	5	5	2	1
S1	As a DJ, I want access to my song collection	25	18	15	3
	research file browser interface	5	5	5	
	create a file browser activity	5	2	2	
	research audio APIs	5	2	2	
	test file browser	5	1	1	
	create Sound SwarmNode class	1	1	1	
	create Swarm Selector Activity and layout	2	3	3	
	test Swarm Selector Activity layout	2	1	1	
	Use an intent to load audio file types	3	3	3	3

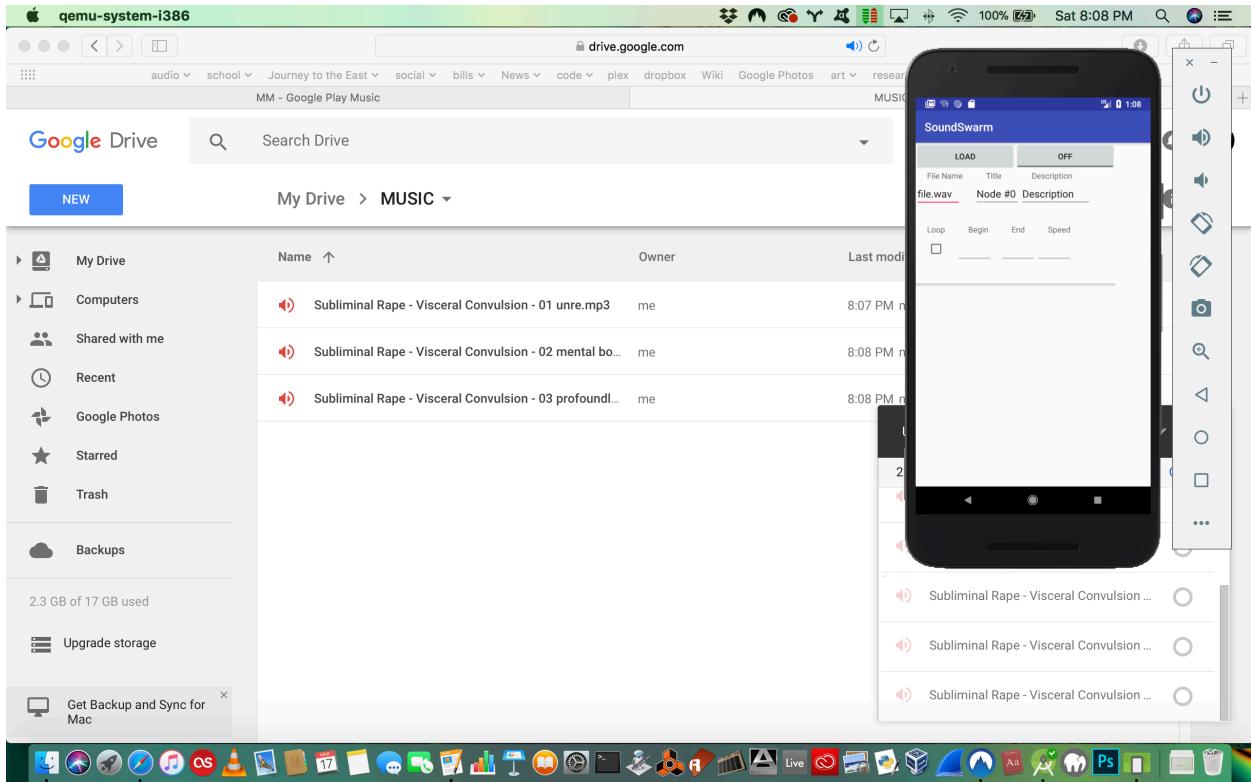
Review

Sprint 3 brought the completion of story 1 and 15. Great progress this week, although some ideas at inception needed to be modified. Instead of using Cards in the recycler, I just modified the existing recycler to hold all the fields for the SwarmNodes. Also, file loading is now possible with an intent instead of building a file browser from scratch. This was a huge time saver and much cleaner to code. There were no extra file adapters to code and no extra Activities or Fragments. For now, the file loads and associates a URI with the SwarmNode making it ready to be wired up which will be the subject of next weeks sprint.

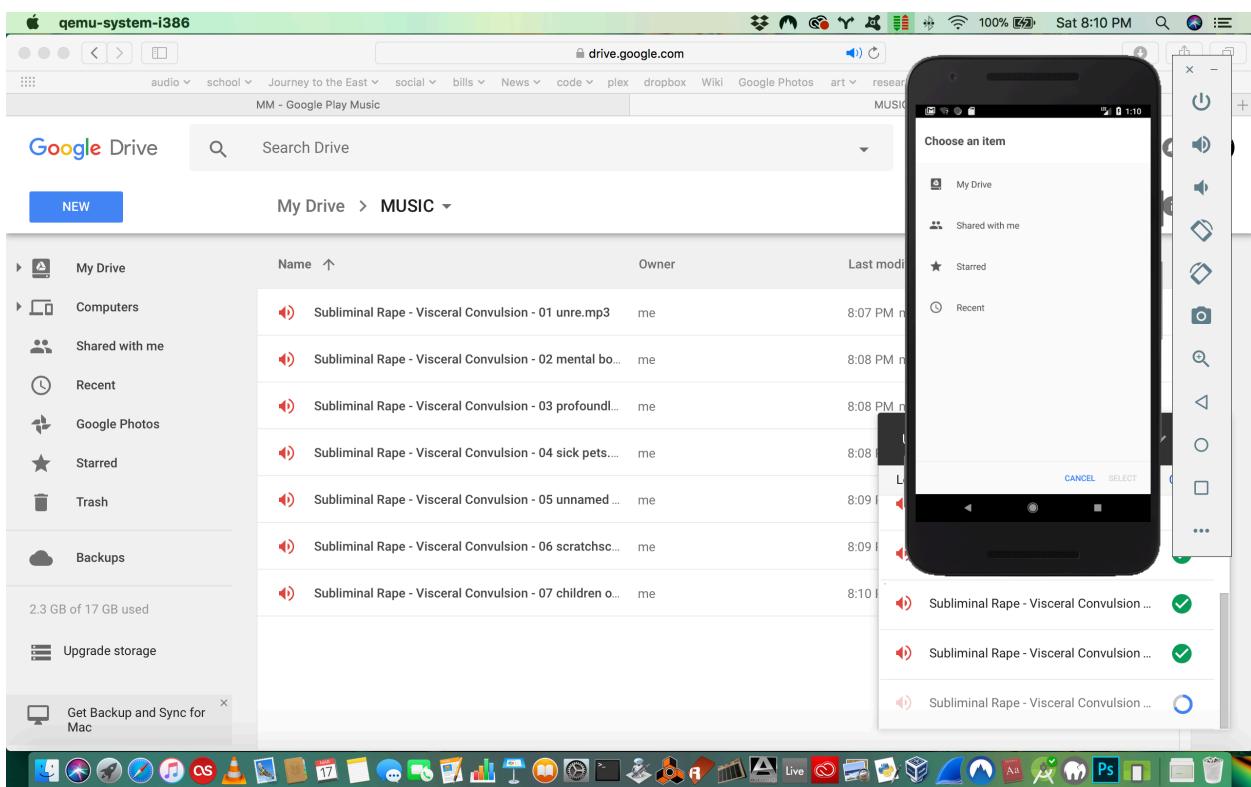
Main Swarm List - A recycler view showing a HUD of all the options for each SwarmNode



This is the original SwarmNode fragment which will be the detail view for the nodes. It is accessible with a click on the background of the desired Node.



This is the intent that shows a file browser it is accessible by pressing the LOAD button.



Retrospective

Two stories were completed this sprint. A lack of design preparation up front led to some confusion this sprint. The informal plan was to make a whole new recycler view for the HUD and link to that from the list with just a play/stop button. Instead, I made the recycle list show everything and drill down to a single view on pressing the background. Some interface refinements will clearly be needed as the presentation is a bit messy. However, the functionality is good so far. The file browser is a big chunk and it now loads an URI into the swarmNode class for use with Soundpool which will be the subject of the next sprint.

Average Project velocity: 13

Two stories completed (25pts + 15pts) for a total of 40 points over three iterations.

Sprint #4

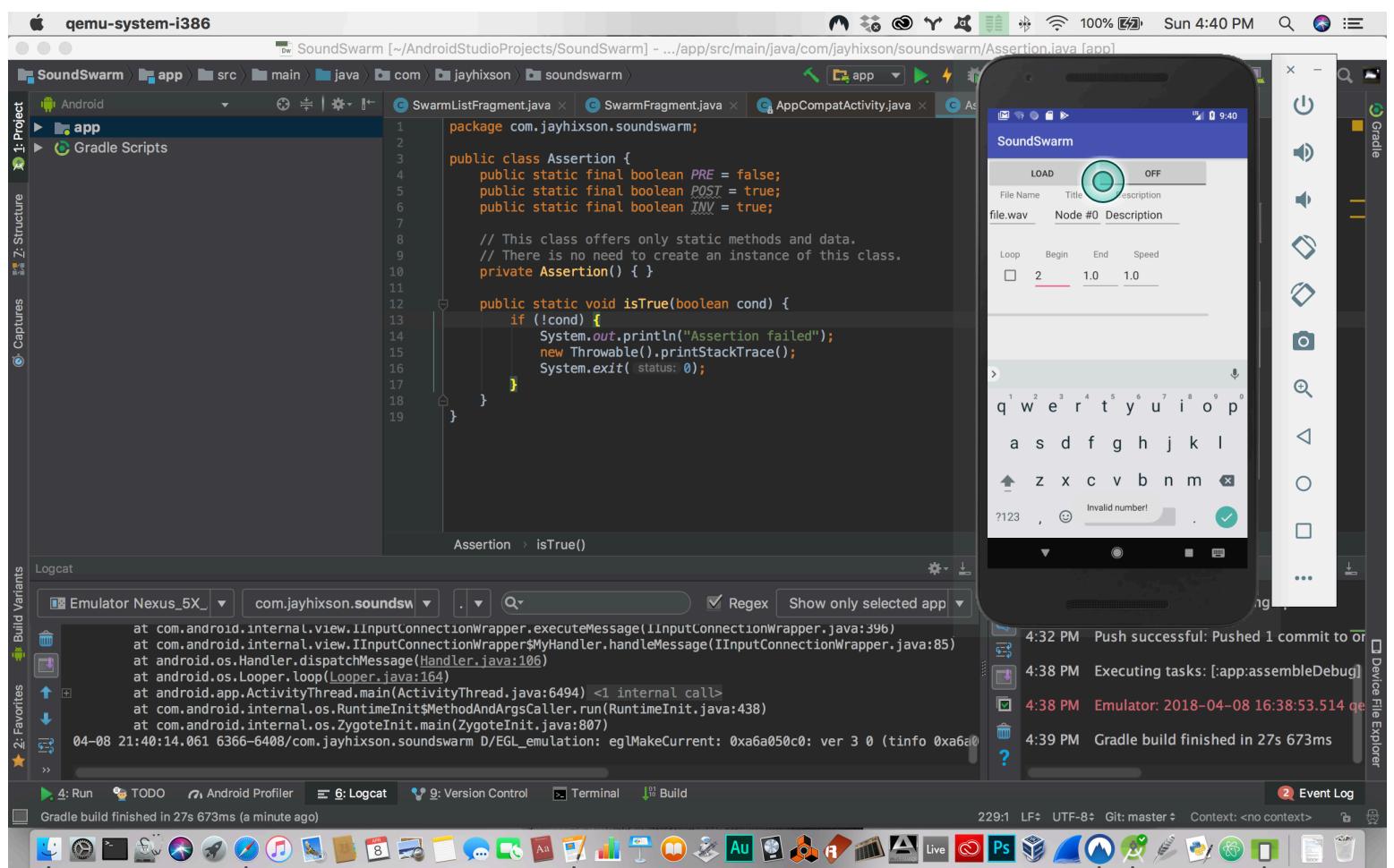
Sprint Backlog

Story ID	Story / Task	Estimated Hours	Actual Hours
S7	As a musician, I want to be able to audition sounds in the app so I can pick the right one	7	5
	Create URI to file source string logic	1	1
	Test sound playback in mediaplayer	2	
	Add exceptions to test Data Types	1	1
	Add assertions to test Data Types	1	1
	test exceptions	1	1
	test assertions	1	1
S14	As a DJ, I want to be able to play more than one sound file at once	10	
	Create Soundpool back end	3	
	add sounds to assets	1	
	autoload sounds	2	
	wire up play buttons	2	
	attach progress bars to playback	2	

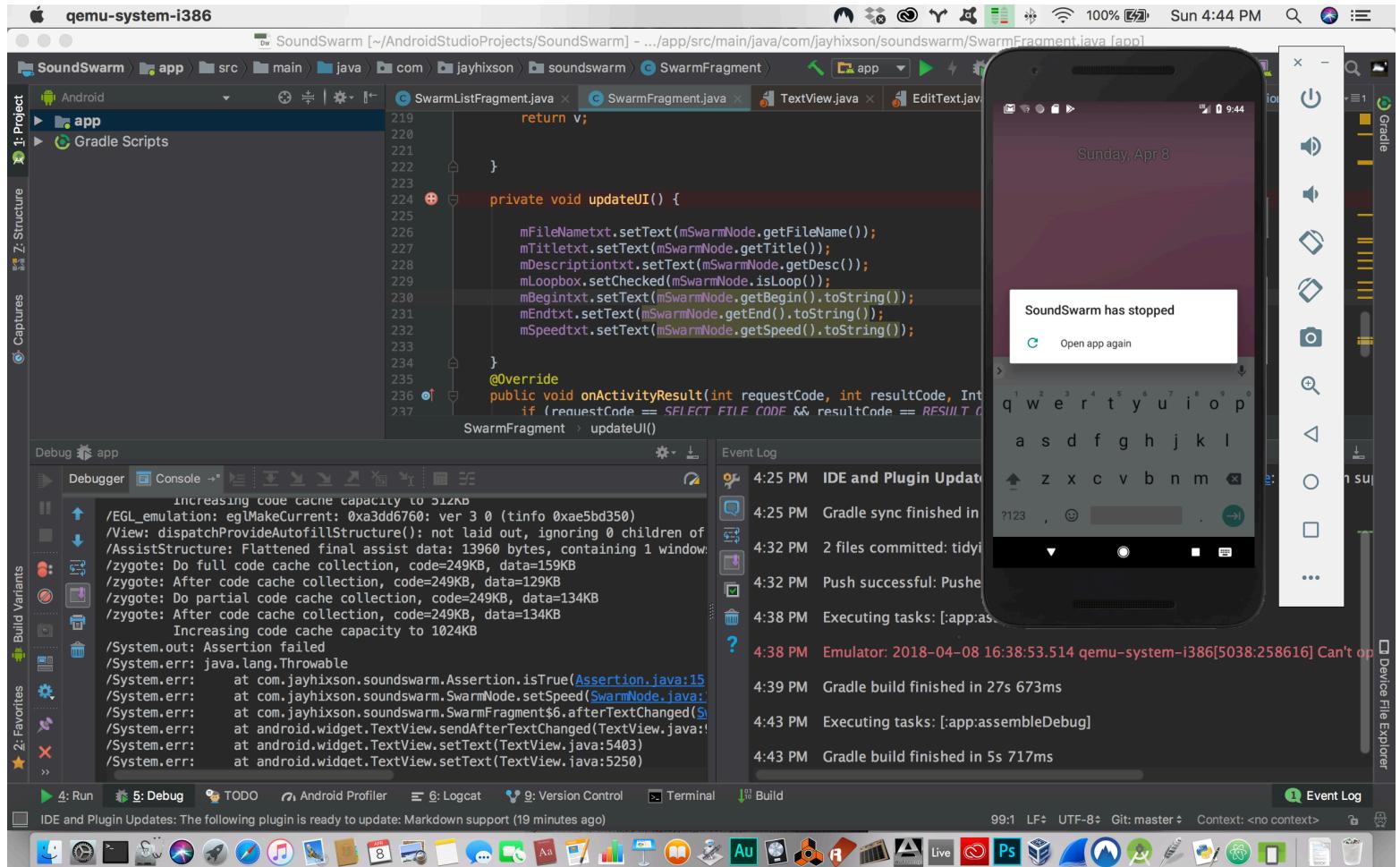
Review

Sprint 4 was planned to be the sprint where we finally turned the sound on and got to hear music, but that didn't happen. Some tidying up commits and refactoring got done which will make the code more robust. Also added in exceptions which are very tight and then added in assertions before them to fulfill this week's requirements.

Notice the small Toast near the space bar that shows that 2 is an invalid number for the start position. Only values between 0.0 and 1.0 are allowed and this entry threw an exception which called the toast.



Here: an assertion failed. Compiling them out by setting the PRE member variable to FALSE doesn't check the precondition and compiles normally.



Retrospective

Two stories were planned to be completed this sprint. Neither were completed. Doing some more research on how to use the mediaplayer is necessary since that isn't really covered in the text for the class. I'll have to get another book or look on line for guidance.

Average Project velocity: 10

No stories completed with 40 points over four iterations.

Sprint #5

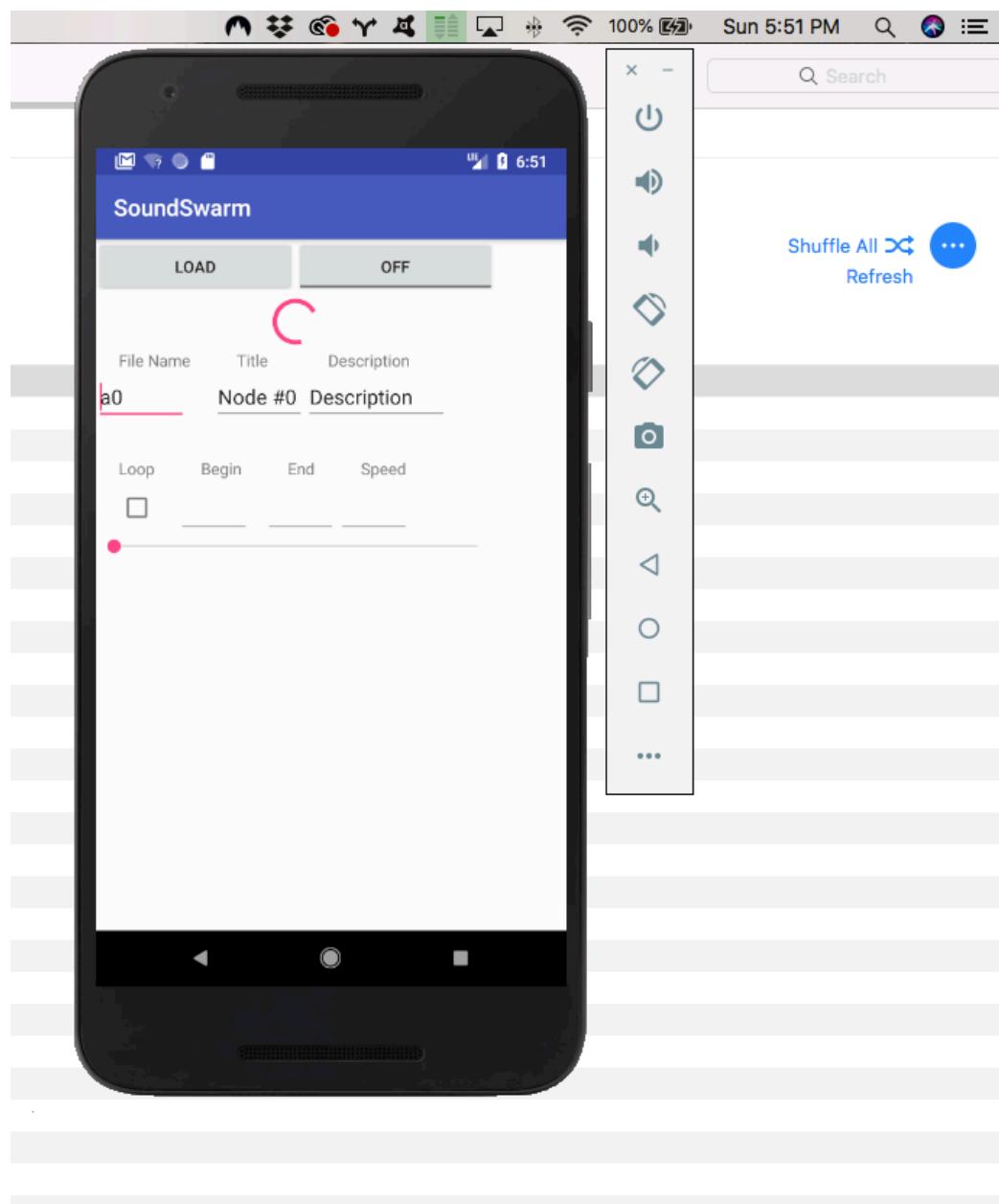
Sprint Backlog

Story ID	Story / Task	Estimated Hours	Last Sprint	Completed
S7	As a musician, I want to be able to audition sounds in the app so I can pick the right one	12	5	7
	Create URI to file source string logic	1	1	1
	Test sound playback in mediaplayer	2	2	2
	Add exceptions to test Data Types	1	1	1
	Add assertions to test Data Types	1	1	1
	test exceptions	1	1	1
	test assertions	1	1	1
	Load URIs into the mediaPlayer object	3		
	handle streaming	2		
	As a DJ, I want to be able to play more than one sound file at once	10		
S14	Create Soundpool back end	3		
	Create Threaded MediaPlayers for each Node	3		5
	add sounds to assets	1		1
	autoload sounds	2		1
	wire up play buttons	2		
	attach progress bars to playback	2		

Review

Sprint 4 was planned to be the sprint where we finally turned the sound on and got to hear music, but that didn't happen. In sprint 5 we have sound! Some cowboy coding and some confusion about the design led to some spaghetti code and boat anchors still to deal with. Instead of using SoundPool (for small files and sound effects) I went with using multiple instances of the MediaPlayer.

Shown below is the individual SwarmFragment. It has a progressbar (not fully implemented) and I updated the progress bar for a seekbar (also not fully implemented)



Retrospective

Story 7 is taking much longer than expected and story 14 is still not complete. Story 7 is missing the logic to handle URIs, there's only prototype code in place for now.

Story 14 has a problem with the file ID resource living in the Swarm class and being needed before it's generated. This code will have to move to make story 14 feasible. With less than 10 hours I could finish both stories, but I suspect finals might keep me from finishing. What works for now is that each individual SwarmNode will play sound in the drill down (accessed by pressing the background of the fragment) This is clunky and I'd like to clean that up quickly with a button before the final sprint is complete.

Average Project velocity: 8

No stories completed with 40 points over five iterations.

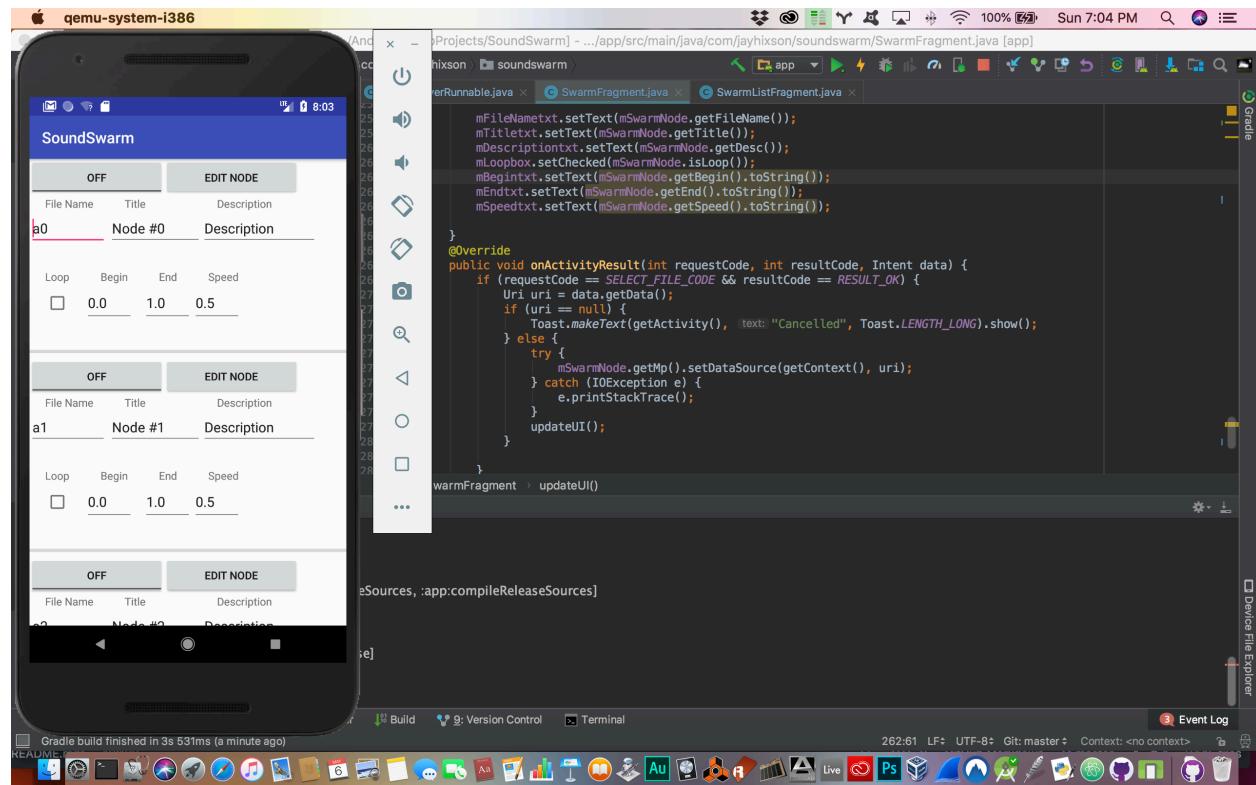
Sprint #6

Story ID	Story / Task	Estimated Hours	Completed	Sprint 6
S7	As a musician, I want to be able to audition sounds in the app so I can pick the right one	12	7	
	Create URI to file source string logic	1	1	
	Test sound playback in mediaplayer	2	2	
	Add exceptions to test Data Types	1	1	
	Add assertions to test Data Types	1	1	
	test exceptions	1	1	
	test assertions	1	1	
	Load URIs into the mediaPlayer object	3		1
	handle streaming	2		
	Unit test for file types in MediaPlayer	4		
S14	As a DJ, I want to be able to play more than one sound file at once	10		
	Create Soundpool back end	3		
	Create Threaded MediaPlayers for each Node	3	5	
	add sounds to assets	1	1	
	autoload sounds	2	1	
	wire up play buttons	2		
	attach progress bars to playback	2		
	add edit button for easy access to edit view	1		1

Review:

In this final sprint I had problems after merging a branch on github. I did it through the website instead of the IDE and this caused problems. I wrestled with it for over an hour before I was able to get the project back to syncing with github and working on my machine. I had to re-upload the audio files, and add release signing to my build. This also took some playing around with the settings to make it work right. After I had my project compiling again, I tried to accomplish three things and got only one to work. First, I added a button to switch to the edit view (swarm fragment activity) this works fine. Second, I tried to fix the MediaPlayer in the recycle view. I wired them up to the toggle button and set them to run in separate threads, but no luck. I couldn't get sound. Third, I tried to get the URI to load into the edit view when a sound file is loaded. This also didn't work.

Here, you can see the edit node button on the right near the play toggle button.



Retrospective

Story 7 still has prototype code, which is better than before but still non-functional. Story 14 added a task to make an edit button. This makes navigation easier and is the only real triumph in this sprint. This project had an easy beginning since I was using the textbook for guidance. Once I was in uncharted waters, however, things got a lot more difficult. This build is a release build made with a key, so it's signed. All together this was a great learning experience and I hope to rebuild this project from the ground up someday. For now it's just a lesson in a craft of which I barely understand the basics. All together, if I were to restart this project, I would build a screen something like an eight channel audio mixer and have an edit button for each channel take the user into a properties window for the MediaPlayer in question. This final build is just a prototype and not all the parts are finished. It's tough to really nail down each function before moving on. But once I start testing before coding, I expect that will get easier.

Average Project velocity: 7

No stories completed with 40 points over six iterations.