Process - Short Sounds

CSE 403 Spring

4/13/15

Joel Sigo	sigoj
Justin Yoon	yoonj4
Nick Jones	ntjones
Neil Harlow	nharlow
Seth Hampson	shampson
Mattie Carlson	mlc327
John Buscher	jbuscher
Casey Fischer	caseympf

Risk Assessment

Our risks have changed considerably since the alpha release, as we have realized that implementing effects and compiling multiple wave files into a single file is going to be much more difficult than anticipated. The main risk is still the same, but the second risk is a new one we have encountered. The last three risks are still the same, but we are less worried about them. We are no longer as worried about making the application backwards compatible because this has proven to be really easy in Android.

Risk One*: The ShortSounds application will take up too much processing power*

Because the application is based on computer audio manipulation, there is a possibility that the audio manipulation will take up too much processing power on a phone to be able to work quickly and efficiently without draining the phone's battery. Because there are other applications that offer similar sound manipulation without processing power being an obstacle, we believe the likelihood of this problem occurring is medium, while the effect of this problem occurring is high because adding effects efficiently is a huge part of what this application is meant to do. In order to mitigate this risk, we will develop the effects one by one, testing them as soon as they are functional. If a certain effect is taking up too much processing power, as determined by the team as a whole, we may make the decision to cut the effect from the application and replace it with a new effect.

Risk Two: *Figuring out how to compile the wave files into a single wave file to share*

This application requires that the wave files of each individual track be compiled into a single wave file for sharing. This is proving to be very difficult. We have had to change much of the backend implementation in the past few days and we have had to change the format of the files that we are saving. We are currently tackling this risk and with the aforementioned techniques but is proving to be time consuming and tricky.

Risk Three: *The ShortSound files will take up too much storage space*

A single ShortSound is composed of multiple mp3 files. Depending on how many ShortSound projects a user creates, this could possible cause some issues with storage space. The likelihood of this occurring is rather low because most phones have the capability to store many, many mp3 files, but if this does prove to be an issue, the level of impact would be high because a user should be able to store and save their projects. In order to mitigate this risk, we will see how much storage space is being used at the beta release and determine if there is an issue. If it is a large amount of space, we will either require users to have a lot of storage space to use the application or limit the amount of tracks that a user can save.

Risk Four: *The ShortSounds user interface may not be intuitive to use*

ShortSounds is an application that is geared towards beginners. In order to make the application accessible to beginners, it needs to be simple and intuitive to use. Unfortunately, many of the effects that we are hoping to make available are rather complicated. Because of this, the likelihood of application becoming too complicated is medium. The effect of the design not being intuitive would also be medium because our application would no longer be suitable for our target audience. To mitigate the risk, we have already designed and done paper prototyping with multiple different designs. For now, we have settled on a design that has proved to be most intuitive to users, but we will continue to do user testing at every stage of our project to ensure that the final project is intuitive.

Risk Five: *Finding a balance between robustness of features and ease of use*

Because our application is for beginners, we do not want to provide so many different effects and controls that it will overwhelm the user. At the same time, we still want to provide a useful and fun application that will allow the user to create a unique sound. In order to do this, we need to find a balance between overloading the user with editing options and making the application so simple that it is no longer useful. This risk has a medium likelihood of occurring because it would be easy to overload the application with so many experienced musicians on the project, but the effect that this mistake would have on our project is low because it would not be too difficult to get rid of functionality after it is implemented. To mitigate this risk, we have already done paper prototyping and intend to continue with usability testing during each phase of the project, redesigning if necessary.

Project Schedule

1) Zero-Feature release 5/01
    a) Action bar functionality designed/defined by the Library/Menu/Action Bar Team
    b) Navigation drawer buttons designed/defined by the Library/Menu/Action Bar Team
    c) All major elements of UI created such that the app is runnable by all teams with the help of Joel and Seth overseeing the UI design.
    d) All back-end classes/interfaces complete with method stubs by the ShortSounds team

2) Beta release part A (back-end push) 5/08
    a) The ShortSounds team will make sure that sounds:
        i) can be saved
        ii) are recordable

    iii) are playable
  b) The Library team will make sure that mixes are loadable
  c) The Effects team will make sure that effects can be added, parameters changed but need not work
  d) The effects team will implement track volume controls
  e) The effects team will have the back end implemented for:
    i) reverb
    ii) EQ

3) Beta release part B (effects and front end push) 5/15
  a) The Library team will make sure that the action bar is implemented and navigation drawer buttons are implemented
  b) The effects team will make sure that effects and short sounds can merge
  c) The effects team will make sure that back-end additional effects final decisions are made
    i) bitcrush? no
    ii) distortion? no
    iii) low pass filter w/ resonator? no
    iv) etc.
  d) Team redistributions will occur this week as needed. Probably moving people from the Library and ShortSounds teams to work on the UI and the effects.
  e) Joel will be in charge of user tests this week

4) Feature-complete release (aka every button works) 5/22
  a) The effects team will be in charge of additional back-end effects ready to test/tweak and additional effect design skeletons
  b) Joel and Seth will continue to gather user test feedback
  c) The library team will work on importing audio files
  d) The library team will work on miscellaneous tasks as needed, including the intro, options, help, share, etc

5) Release candidate 5/29
  a) The effects team will verify that all effects can be added and do what they are supposed to do
  b) The ShortSounds team will be in charge of recorded input/output quality testing
    i) when tracks are layered does it start to peak?
    ii) consider auto limiters/compressors
  c) The Library team will be in charge of system integration testing and debugging
  d) Joel and Seth will add finishing touches to the UI (animations)
  e) Joel and Seth will continue with user testing

6) Final project presentation 6/05
  a) The Library team will continue with system integration tests

b) The musicians in the group will create sample projects (or acquire them from users)
c) Joel and Seth will make any final UI tweaks

Team Structure

Our team is divided up into three main different subteams based on the features that we are going to implement so that every person in our team will have an opportunity to experience full stack Android development. The Short Sounds team is composed of Neil, Joel and Justin, and they are in charge of making the model that deals with the ShortSounds that are compilations of different mp3 files. So far, this team has set up a database prototype for both ShortSounds and ShortSoundTracks. They will also be in charge of figuring out how to save, play, and record a sound. The next team is the Tracks/Effects team which is composed of Nick, Casey, and Seth. Seth will focus on the UI implementation of the effects and Nick and Casey will focus on the backend implementation of these effects. This team will be responsible for adding effects to audio files, adjusting the effects, and the display of the different effects. Mattie and John will make up the Library/Menu/Action Bar team. This team will be in charge of determining how to display the library, connecting the library to the stored ShortSounds, and determining the look and function of the action bar, and implementing these things. Joel will oversee the overall user interface design in order to ensure that it is coherent and is intuitive to users. He will specifically help Mattie and John designing the library and action bar. So far, Joel and Seth have laid out and implemented a basic UI design. Our tech lead is Neil and he is in charge of making key design decisions about the backend. Our project manager is Nick and he is in charge of meeting agendas and scheduling. Each of the team's specific tasks can be seen in the project schedule above.

Our team's main source of communication will be via our Facebook page. Updates on anything completed or requests for help will all be posted here. All of our documentation and meeting notes are shared in a Google Docs folder. All of our source code is shared on GitHub. We will also be meeting Tuesdays at 9:30 each week to update each other and make our plans for the coming week. Subteams meet once a week on Thursday mornings, around 10:30 in the labs so the teams can communicate.

Test Plan

All unit tests will be designed and implemented by the developers as they develop. It's been recommended for everyone to use test driven development, so that the tests our written before the implementations. However, if they choose not to use test driven development, it's expected that unit tests will be written closely after the code is.

System tests will be written after major feature modules are implemented, to insure that they interact as expected. The teams which created the modules will be responsible for the writing the system tests.

Usability tests will be done before the release candidate is presented by Joel and John to ensure that the product is intuitive and easy to use, helping catch any flaws in our design. This will be done through testing with our peers outside of the computer science department, having each participant work through a series of use cases and then play around with the application while being supervised and communicating with the facilitators.

Ideally, we would prefer to have an automated testing suite, however, with the limit on time that we have, we don't have the time to implement a full testing suite. Running the tests from a server automatically has proven difficult with android tools. Instead, developers can run the test suite through a command line tool before making any changes to master. This has proved more time efficient, and just as effective as automated testing.

For tracking any bugs and issues in our code, we plan on using GitHub's built in bug tracker. That way every group member has easy access to the full list of bugs.

Documentation Plan

We have a plan to write one piece of external documentation in the form of a user guide. It will give an overview of the application, as well guide the reader how to use each of the features made available by the application. We also plan to have some internal documentation built into the application as well. This would be in the form of a help menu, which will share a lot of content as the user guide, but briefer, and easy for a user to read while attempting to use the application.

Coding Style Guidelines

For our programming, we agreed to use Google's style guidelines. The only two languages we plan on using, Java and XML, use the style guidelines provided by the google links below.
https://google-styleguide.googlecode.com/svn/trunk/javaguide.html
http://google-styleguide.googlecode.com/svn/trunk/xmlstyle.html
Each development team is working in at least pairs, so our plan to enforce the guidelines is to have each partner hold each other accountable. Since the pairs should be sharing a lot of code, it should not be too difficult to realize if your partner is violating the style guidelines.