Development Standards & Practices Used

List all standard circuit, hardware, software practices used in this project. List all the Engineering standards that apply to this project that were considered.

Summary of Requirements

List all requirements as bullet points in brief.

Applicable Courses from Iowa State University Curriculum

List all Iowa State University courses whose contents were applicable to your project.

- Com S 227/228
- EE 224
- EE 285
- EE 321

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

- Android development
- Generation of envelope functions

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1 Introduction

1.1 Acknowledgement

- Drs. Randall Geiger and Degang Chen served an advisory role and offered technical advice.

1.2 Problem and Project Statement

While there exist fully-featured samplers on iOS, samplers on Android have a limited feature set. As well, existing sampler applications for smart devices fail to offer users an intuitive view of the effects of their actions. This project is a fully-featured Android sampler app geared around offering a clear picture of what is happening and an accessible UI.

1.3 Operational Environment

The end product will be an application for Android devices. The operational environment is a tablet, phone, or other smart device running the Android OS. One of the constraints that comes with this is the wide variation in processing power between different devices.

1.4 Requirements

- The product must sound "good" to musicians
    - Otherwise, the product will not be economically viable
- The development costs of the product need to be covered entirely by in-app ads or donations
- Android platform
    - The app must be available for the Android platform because there are few fully-functional samplers on Android
        - Functionality of a "fully-functional" sampler is covered later
    - Written in Java
- Easy to use UI
    - Drum pads
- The sampler must have an envelope feature
    - An envelope feature prevents unpleasant pops when samples are triggered
    - It also allows sound modulation in ways musicians enjoy

- ○ The envelope and the filtered sample should be shown on screen to paint a more intuitive picture of what envelopes do
- Pitch-shifting feature
  - ○ To prevent samples in different keys from creating dissonances (or to allow dissonances to be created)
  - ○ Two or more samples in different keys playing at once will often sound unpleasant
- Speed-changing feature
  - ○ For musicians in drum 'n' bass or hip-hop, the ability to change the speed of samples is desirable
  - ○ The UI for this feature should include presets based on record player speeds
- Sampler must have Bandpass filter that can be applied to audio
- Samples should be saved as WAV files and be available for later use
- Presets should be possible to save locally
- Haptic feedback will be used to provide musicians a "feel" response

1.5 Intended Users and Uses

The product has two major user groups: hobbyist and performing musicians. Hobbyist musicians will use the app as a more robust version of existing sampler apps that better substitutes for a physical sampler. Performing musicians will use the app to create and perform music on a more portable device.

1.6 Assumptions and Limitations

Assumptions:

- The application will run on a fairly modern (released within last five years) Android device
- The maximum number of simultaneous inputs to the app is less than ten

Limitations

- The application will need to be lightweight enough to run on phones
- The application will need to output audio in a format compatible with how Android interfaces with speakers

1.7 Expected End Product and Deliverables

End product: sampler application for Android
Delivery date: week of April 27, 2019
The end product will be a sampler application for Android with the features listed in section 1.7.
It will also include refinements based on feedback from alpha and beta testing.

Deliverable: equalizer component
Delivery date: TBD

Deliverable: speed-shifting component
Delivery date: TBD

Deliverable: pitch-shifting component
Delivery date: TBD

Deliverable: envelope component
Delivery date: TBD

Deliverable: UI
Delivery date: TBD

2. Specifications and Analysis

2.1 Proposed Design

Include any/all possible methods of approach to solving the problem:

– Discuss what you have done so far – what have you tried/implemented/tested, etc?

– We want to know what you have done

• Approach methods should be inclusive of **functional and non-functional requirements** of the project, which can be repeated or just referred to in this section

If your project is relevant to any **standards** (e.g. IEEE standards, NIST standards) discuss the applicability of those standards here

Things tried so far:

2.2   Design Analysis

2.3   Development Process

2.4   Design Plan

3. Statement of Work

3.1 Previous Work And Literature

The iMPC sampler app is no longer available for Android as of October 2019.

3.2 Technology Considerations

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

Android Studio

Strengths:

- Intuitive UI that allows all members to contribute

3.3 Task Decomposition

Most of the tasks in this project require that we have a working way to load audio from the Android file system, then play it from our app's memory. Once that works, the audio processing tasks can all be handled independently. Then, they will need to be combined and integrated into our UI.

3.4 Possible Risks And Risk Management

One risk that our project faces is that the implementations of our equalizer and pitch-shifter could be more complex than initially envisioned. Another risk is our team's limited Java programming experience.

3.5 Project Proposed Milestones and Evaluation Criteria

The first key milestone is having a functional equalizer. In order to test the equalizer functionality, we plan to compare the drawn function shape to the actual shape. We also plan to test a series of random inputs using Monkey.

The second key milestone is having a working sample speed shifter. This can be tested using samples of various lengths and comparing the output of our app to a working MATLAB equivalent.

The third milestone is implementing frequency-shifting. This will be testing again using a set of samples of various lengths, initial frequencies, and amplitudes. This will let us cover a wide variety of types of inputs.

3.6 Project Tracking Procedures

Our group will use Google Drive and Git to track progress.

3.7 Expected Results and Validation

The desired outcome is that we would be able to create a sampler app that lets users understand and feel involved in the process of performing with a sampler. Our plan to ensure our solutions work is to perform beta and final testing with performing and hobbyist musicians.

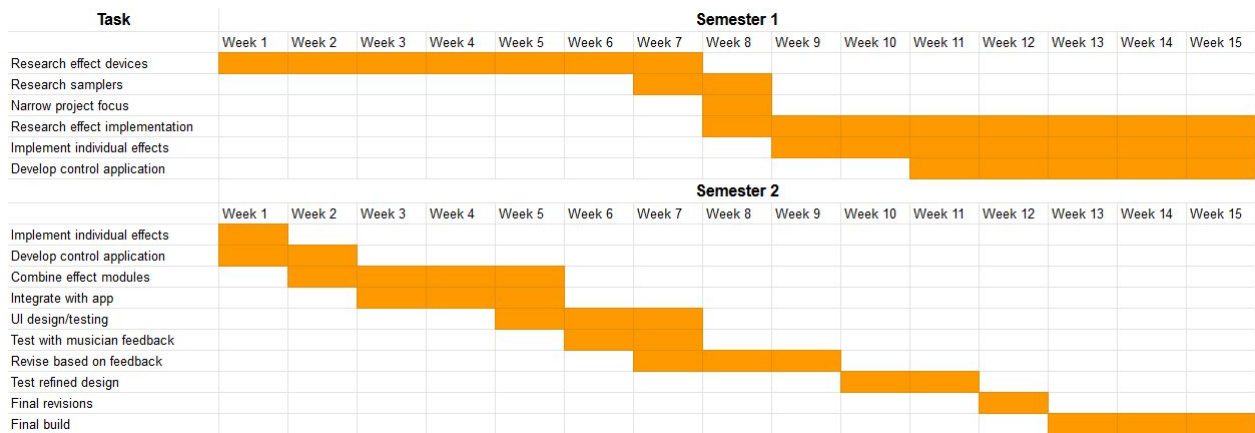# 4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

• A realistic, well-planned schedule is an essential component of every well-planned project

• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

Task	Semester 1														
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15
Research effect devices	███	███	███	███	███	███	███								
Research samplers							███	███							
Narrow project focus															
Research effect implementation								███	███						
Implement individual effects									███	███	███	███	███	███	███
Develop control application												███	███	███	███

Task	Semester 2														
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15
Implement individual effects	███														
Develop control application	███	███													
Combine effect modules			███	███											
Integrate with app					███										
UI design/testing					███	███									
Test with musician feedback							███	███							
Revise based on feedback								███	███						
Test refined design										███	███				
Final revisions												███	███		
Final build													███	███	███

To-do: replace Gantt chart with version based on final design plan

4.2 Feasibility Assessment

Challenges that will be faced throughout the project process include implementing envelopes, UI design, and working with Java.

4.3 Personnel Effort Requirements

4.4 Other Resource Requirements

- Computers with the Android Studio IDE
  - The group members will be using their personal computers

4.5 Financial Requirements

As Android Studio is free software, no financial resources will be needed to conduct the project.

5. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)

2. Define the individual items to be tested

3. Define, design, and develop the actual test cases

4. Determine the anticipated test results for each test case 5. Perform the actual tests

6. Evaluate the actual test results

7. Make the necessary changes to the product being tested 8. Perform any necessary retesting

9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

Functional Testing

-

Non-Functional Testing

-

Modeling and Simulations

Challenges

5.1 Interface Specifications

– Discuss any hardware/software interfacing that you are working on for testing your project

5.2 Hardware and software

Monkey is a piece of software that allows a pseudo-random set of inputs to be sent to an app for testing purposes.

5.3 Functional Testing

- The application will be compared to pre existing software for PC that performs similar effects
- Expected output and obtained output will be compared for a variety of length, amplitude, and frequency content values
- The app will be tested with musicians to obtain feedback and make any necessary changes

5.4 Non-Functional Testing

- The app will be tested with musicians to obtain feedback and make any necessary changes
- Monkey will be used to test the app against a random series of inputs

5.5 Process

5.6 Results

6. Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.