Sound Effect Devices for Musicians

sdmay20-56.sd.ece.iastate.edu

Dalton Sherratt, Eric Stablein, Zach Besta
Advisor/Client: Dr. Randy Geiger

Some sound effect devices

- Synthesizers
- Effect pedals
- Loopers
- Keyboards
- Electronic Drum Pads
- Samplers

What is a Sampler?

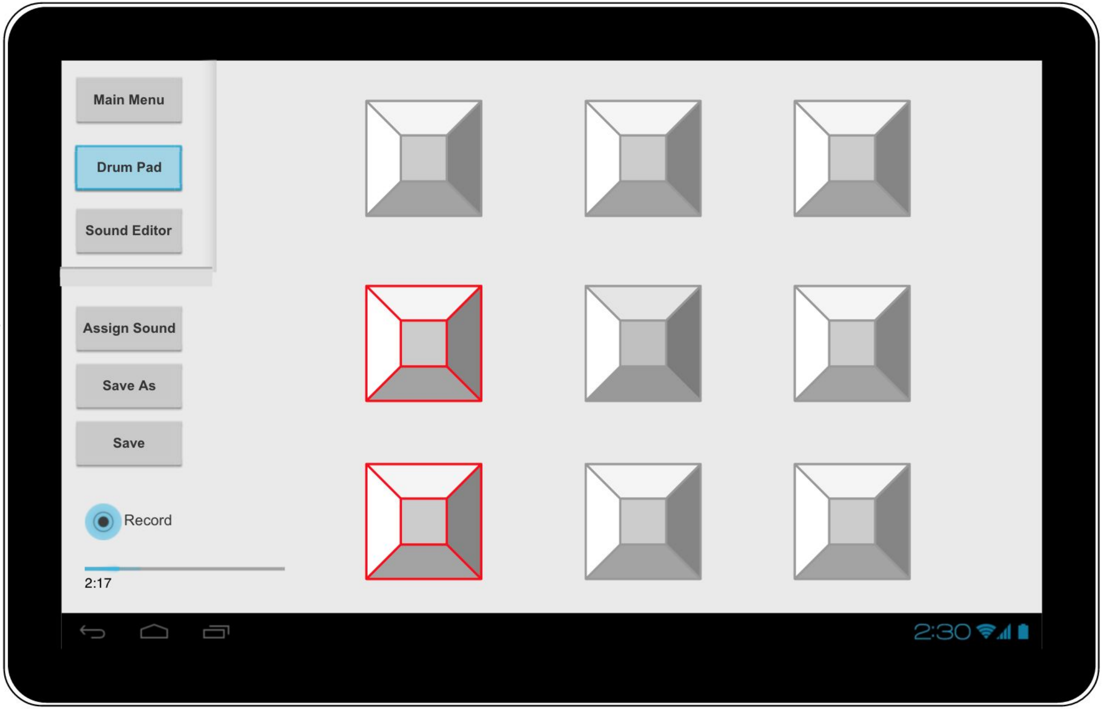
- Stores a snippet of audio and plays it when triggered
  - Some have step sequencers
- Can be used to synthesize drum patterns or for even more
- Used heavily in rap, pop, and electronic dance music
  - Kanye West
  - Dr. Dre

Problem Statement

- Existing sampler apps for Android lack important features and are inaccessible
  - Hard to understand without or prior knowledge
  - Missing envelopes, pitch-shifting, etc...

Top-Level GUI - conceptual sketch

r



Functional Requirements

- The application must be able to run on a modern Android device (API Level 28, Android 9.0 “Pie”)
- The application must have an easy-to-use, intuitive UI
- The application must have a “full” feature set (to be discussed soon)

More functional Requirements

- The application must conform to the requirements of the Google Play store
- The application needs to have an appealing sound to musicians, both hobbyist and professional
- The application must conform to Google’s material design standards

Non-functional requirements

- Warn users about hearing damage at high volumes
- Make sure users know to take breaks to avoid repetitive stress injuries

Technical/Other Constraints & Considerations

- Different processing capabilities between devices
  - Chose to focus on modern devices (API level 28+ - Android 9.0)
- Screen size/shape difference between devices
  - Tablet vs phone
  - Landscape vs portrait

Potential Risks & Mitigation

- Lack of business experience
  - Hard to judge economic requirements
  - Limited knowledge of the app marketplace
- To mitigate
  - Monitor economic requirements and extrapolate as new data is obtained
  - Continually monitor app marketplace

Market survey

- iMPC no longer offered
- Several simple samplers available
  - Mainly in “low-end” range (under $2.99)
- Samplers with more features are cluttered and unclear
  - Mainly in “premium” range (over $10)

Market survey examples

App	Price	Functions
nanoloop	$3.49	Envelope (AD only) Start offset
Pocket Sampler - DJ Launchpad	$2.99	Online database
iMPC (Apple only)	$6.99	Time correction Effect modules Recording and overdubbing Step sequencer
G-Stomper Studio	$12.99	Step sequencer Effect modules Real-time modulation

Resource/Cost Estimate

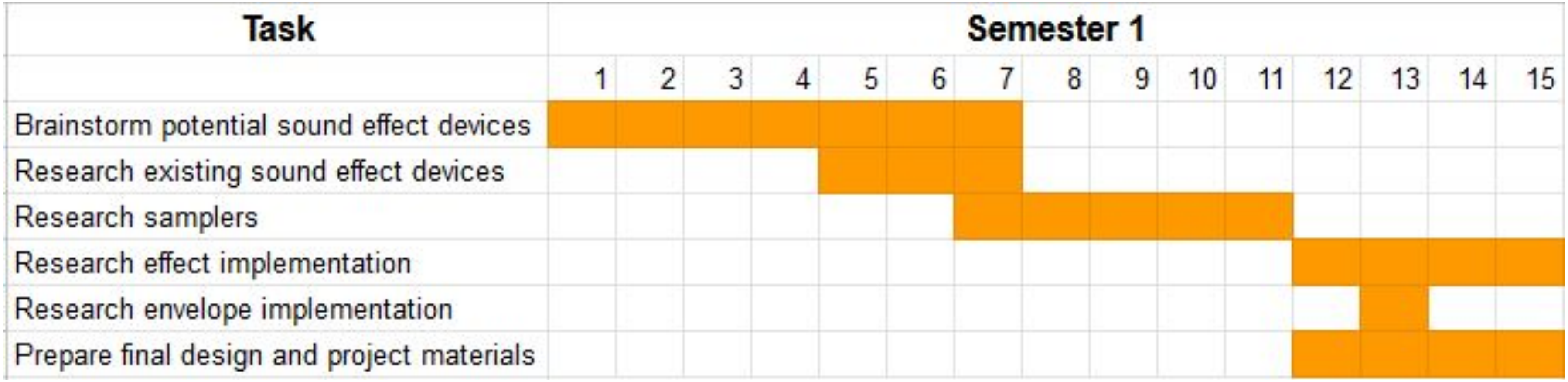
Tentative resources:

Item	**Price**	**Purpose**
Lenovo Tab M10	$100	Low-end tablet
MPC Studio	$300	Major physical sampler

Note: not purchased due to university closure.

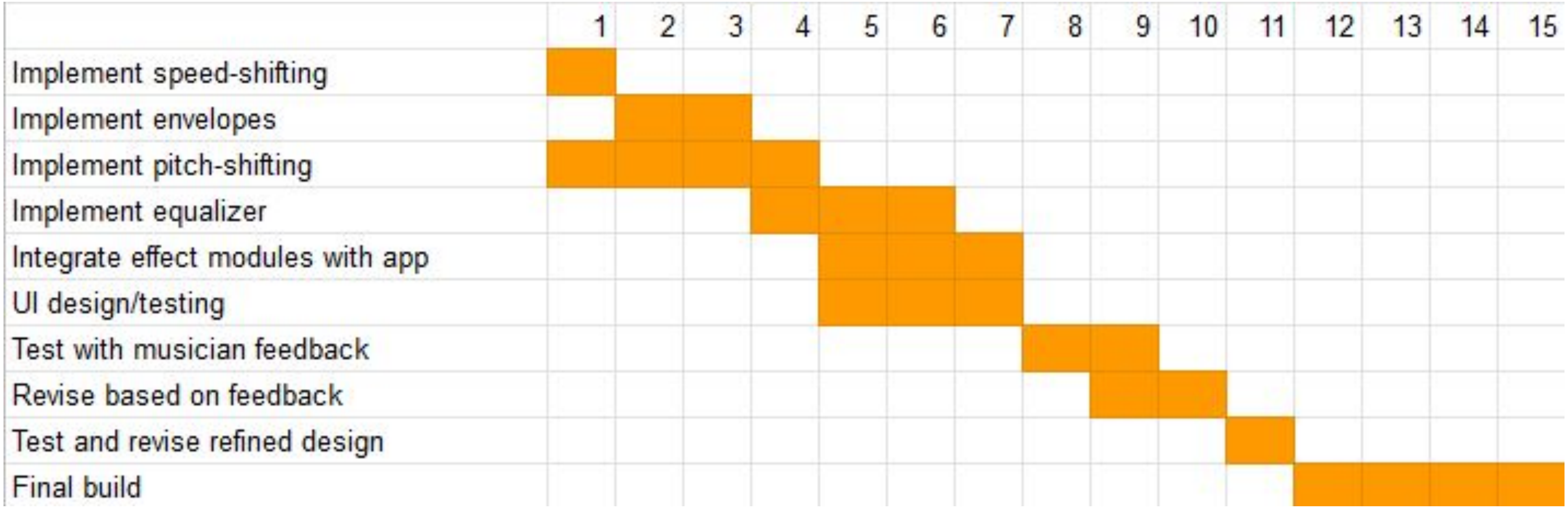
Project Milestones & Schedule (1)

- Each column represents 1 week



Task	Semester 1														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Brainstorm potential sound effect devices	■	■	■	■	■	■	■								
Research existing sound effect devices					■	■	■								
Research samplers							■	■	■	■	■				
Research effect implementation												■	■	■	■
Research envelope implementation													■		
Prepare final design and project materials												■	■	■	■

Project Milestones & Schedule (2)

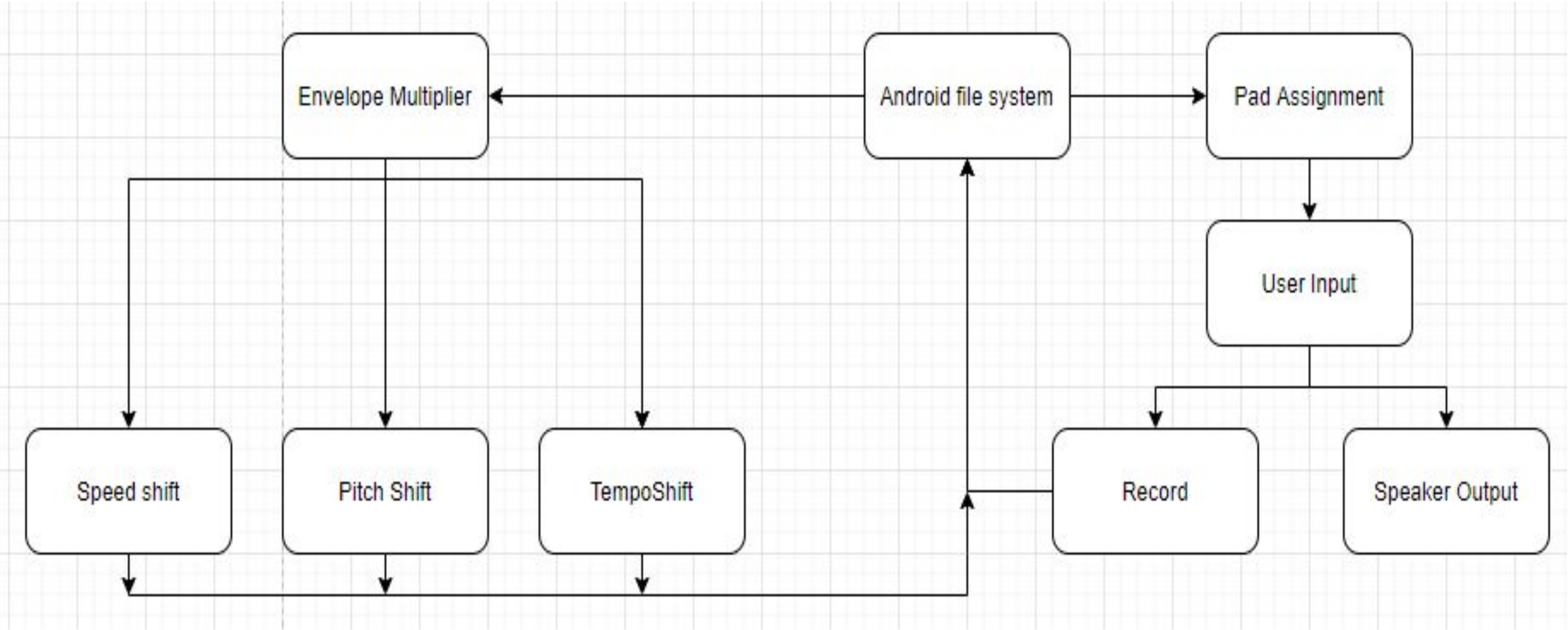


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Implement speed-shifting	🟧														
Implement envelopes		🟧	🟧												
Implement pitch-shifting	🟧	🟧	🟧	🟧											
Implement equalizer				🟧	🟧	🟧									
Integrate effect modules with app					🟧	🟧	🟧								
UI design/testing					🟧	🟧	🟧								
Test with musician feedback								🟧	🟧						
Revise based on feedback									🟧	🟧					
Test and revise refined design											🟧				
Final build												🟧	🟧	🟧	🟧

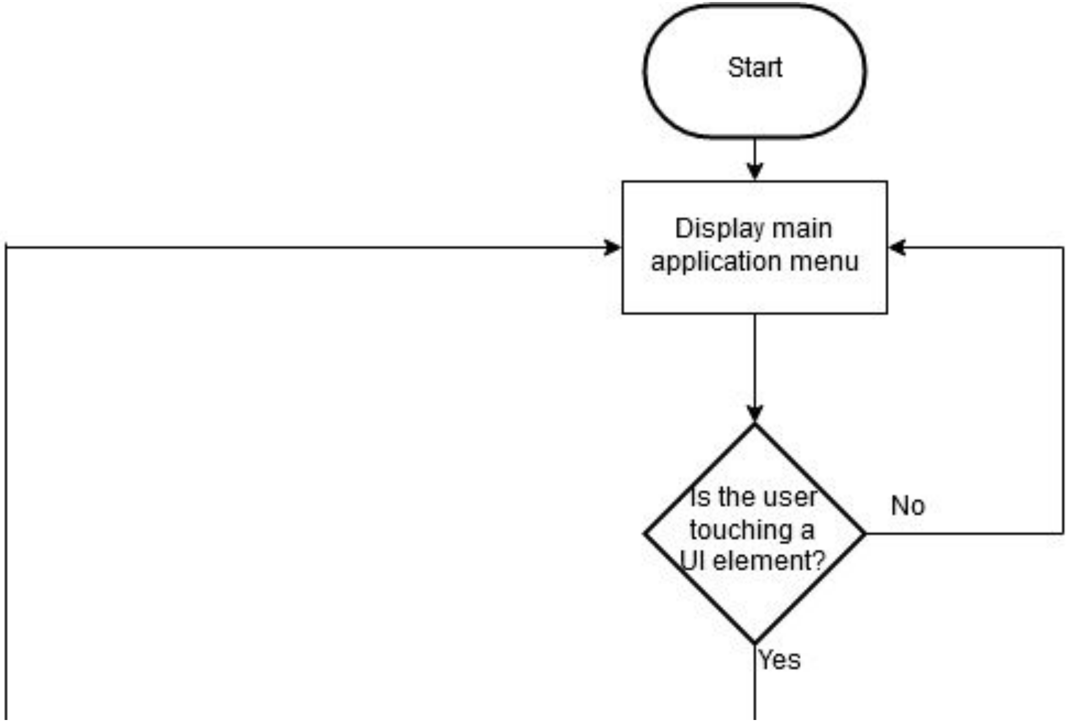
Functional Decomposition

1. Loading audio from Android file systems
2. Audio processing tasks
   a. Create individually, then merge
   b. Order of effects matters
3. Integrate into UI

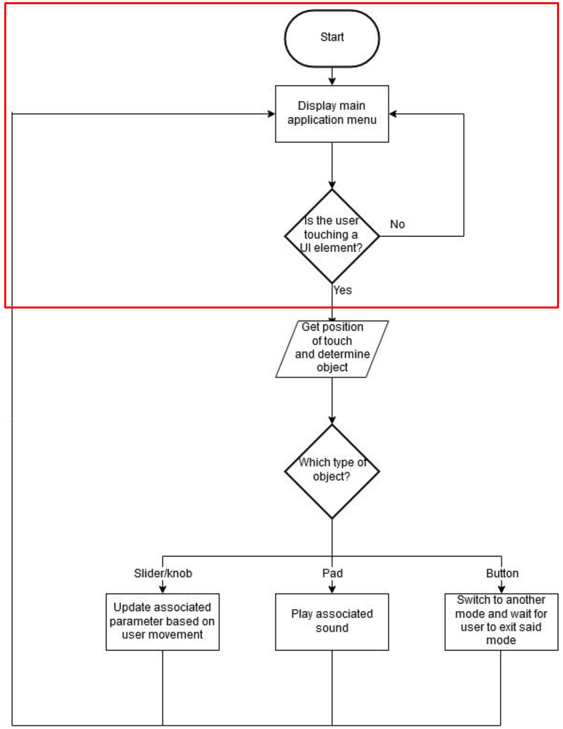
Detailed Design



Detailed Design

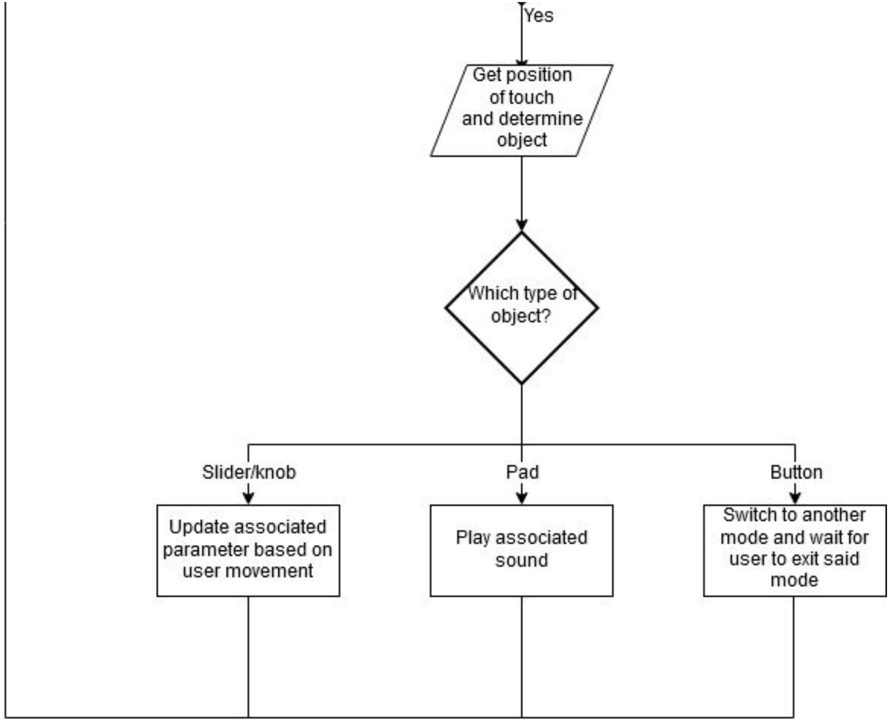


Start
Display main application menu
Is the user touching a UI element?
No
Yes

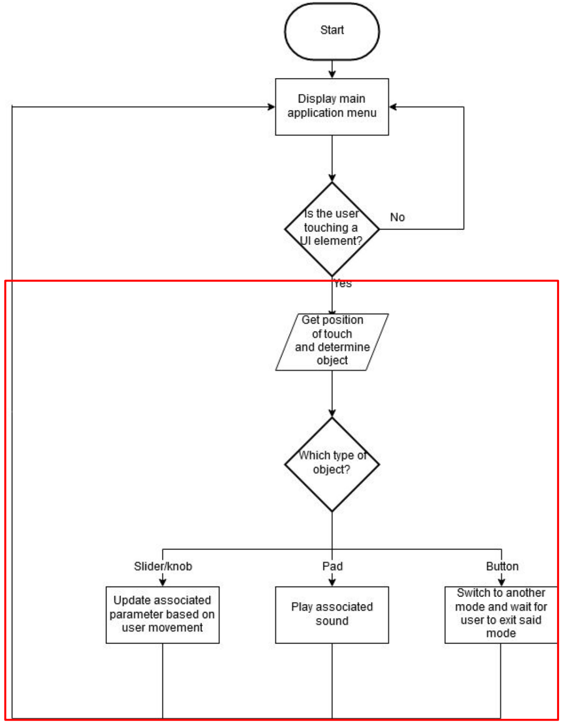


Start
Display main application menu
Is the user touching a UI element?
No
Yes
Get position of touch and determine object
Which type of object?
Slider/knob
Pad
Button
Update associated parameter based on user movement
Play associated sound
Switch to another mode and wait for user to exit said mode

Detailed Design



Yes
Get position of touch and determine object
Which type of object?
Slider/knob
Pad
Button
Update associated parameter based on user movement
Play associated sound
Switch to another mode and wait for user to exit said mode



Start
Display main application menu
Is the user touching a UI element?
No
Yes
Get position of touch and determine object
Which type of object?
Slider/knob
Pad
Button
Update associated parameter based on user movement
Play associated sound
Switch to another mode and wait for user to exit said mode

UI design

- Separate screens
  - Play
  - Edit
  - Create
- Buttons for sound editing
- Material Theme to target a beginner electronic music artist
- Simplicity for accessibility

HW/SW/Technology Platform(s) used

- Android OS (Pie and newer)
- Using Android Studio to develop
- Personal computers or ISU lab computers
  - Virtual machines an option to handle lab closures
- Emulator within Android Studio and personal Android devices for testing
- Git used for collaboration

Test Plan

1. Tests (Mockito/JUnit) will be implemented by each team member
2. Test modules individually
   - Test-first approach
3. Test modules together
4. Test user interface
5. Beta testing with musicians for functionality (may be cancelled due to COVID-19)

Prototype implementations

- Different sound playback options
  - MediaPlayer
  - SoundPool
  - AudioTrack
- Currently have a working MediaPlayer-based implementation of effects due to features
- SoundPool is used for audio playback

Basic building block implementations

- Early semester 2: pitch-shifting, tempo adjustment, general sound playback, XML
- Spring break: equalization, envelopes
- Second half of semester 2: file IO, recording

Engineering Standards and Design Practices

- Google Play standards
  - Must target at least API level 28 (Android Pie)
- General good coding practice
  - Comments and JavaDoc
  - Implementing tests early on
  - Code Modularity

Main responsibilities

- Dalton: UI design/XML, Sound Processing, IO Stream, and testing framework
- Zach: UI design and audio processing
- Eric: Audio processing and UI integration

Future Options for the App

- The ability to share from the app to social media
- Porting the app to iOS
- More sound editing functionality
- An expansive built-in library of sounds
- More sound generation functionality

Live Demonstration

We will now give a live demonstration of our sound sampling application

Useful Links

- Project folder (requires an Iowa State University login): https://drive.google.com/drive/folders/1J6CoUSB8kGIspMl5Rx_ZolqqcnhsiKBc?usp=sharing

- Git repository: https://git.ece.iastate.edu/sd/sdmay20-56

- Team website: http://sdmay20-56.sd.ece.iastate.edu