

# MUSICAL INSTRUMENT SIMULATOR

Sailendra Bathi Babu - 170050081

Charan Teja Narisimsetti - 170050095

## Description:

The project is about simulation of an actual Piano, Electronic Drum-pad, Guitar using scheme language. The simulation not only allows user to create music according to the input but also have features such as “record” , “play” and also a “REMIX” feature. This remix is very interesting feature as allows user to create some beautiful soundtracks by overlapping music created using the simulator or predefined music.

### More About Remix:

Cut : cuts the sound to required length.

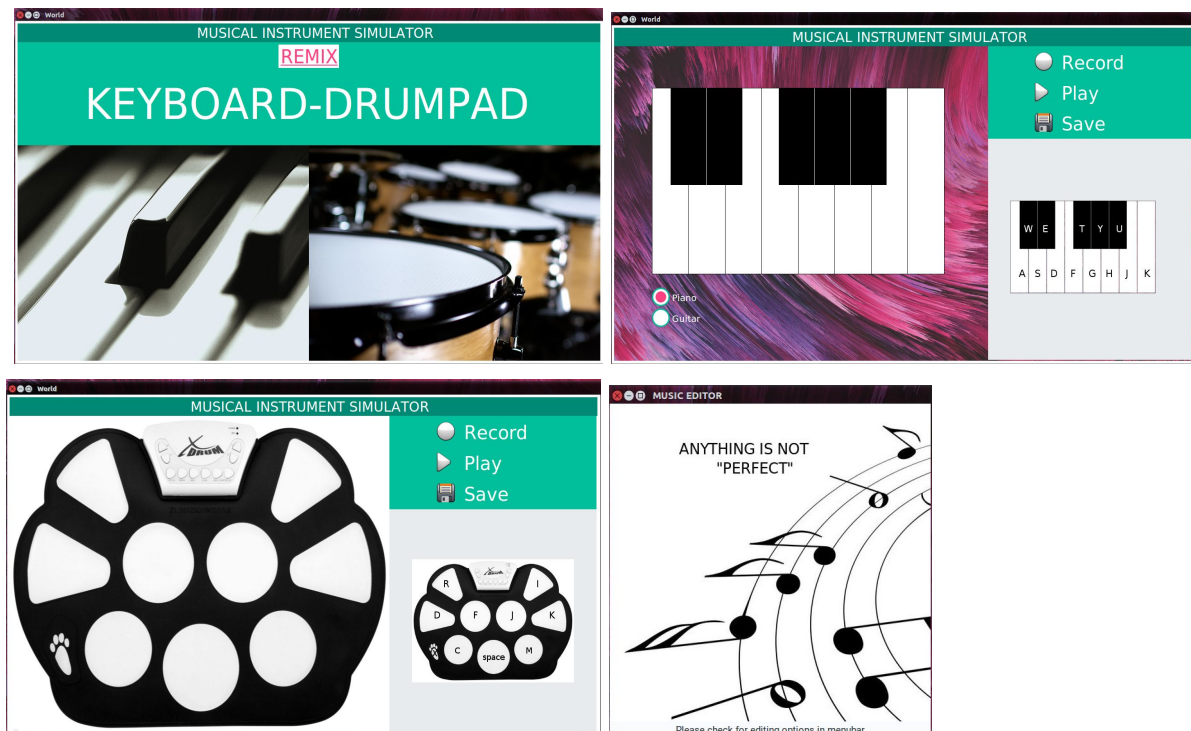
Append : joins two sounds.

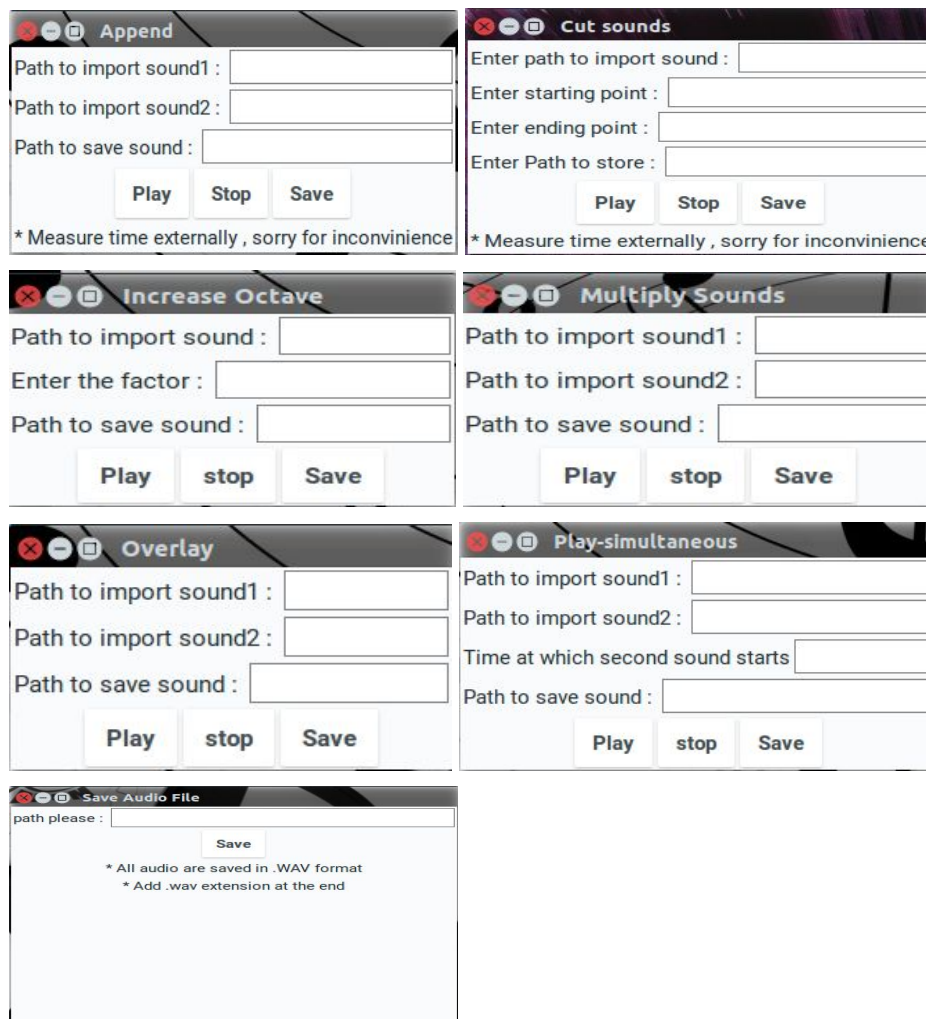
Overlap : as the name says overlaps any two sounds.

Play-simultaneous : This is like overlap except that one sound starts playing after some time as other sound starts playing. The time gap is specified by user.

Multiply : multiplies two sounds with respect to their wave functions and gives the sound of resultant wave function. (not same as overlap)

Increase Octave : Increases the octave number of the tunes.





## Design Idea:

**GUI :** The User interface is designed to look as close as possible to reality while being attractive and convenient to use. The input (discussed under next heading) is also decided keeping user convenience in mind. The theme is based on “Google’s material design principle” found in Android OS so as to look minimal and decent.

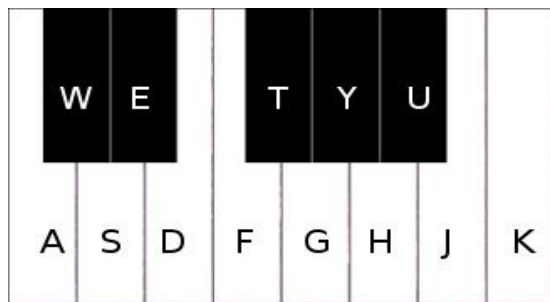
**Program:** The code for the program is developed by dividing into two parts GUI part (UI abstraction) and sound part (audio abstraction) and finally are interlinked using both Functional and Object Oriented Programming. Both of these paradigms are used because of their advantages in some particular aspects. Sound parts were implemented using the packages “rsound”, “racket/gui”. The first one was used to get musical tones while the latter was used combined with rsound to implement features like record, play, remix.

## Input-Output:

**PIANO :**

**INPUT :** specific keys from the keyboard are allocated as piano keys as shown in figure.

**OUTPUT :** Similar sound as obtained from an original piano for .



#### GUIAR :

INPUT : keys similar to that of piano, just the tunes are changed.

OUTPUT : guitar tunes corresponding to keys

#### DRUMS :

INPUT : each drum component is bound to a specific key on the keyboard.

OUTPUT : The sound corresponding to the drum component.



## Limitations:

It is to be noted that we are limited by the size of keyboard on the laptop so we cannot create a full-size 88 key piano. We could use all the keys on the keyboard but it would be very messy to use. So we took the middle 13 keys of the piano.

Also for guitar, we used the fact that the guitar sounds are similar to that of sounds obtained when decreasing 12 notes in piano-tones. So not all features that you'd expect from an acoustic guitar is simulated here. While using "REMIX" feature, the only inconvenience is that time has to be measured externally while using features like cut, play-simultaneous etc.

## Point of Interests:

This project involved a vast variety of packages to be used. They are racket/gui, rsound, rsound/piano-tones, 2htdp/image, 2htdp/universe, images/icons/control, images/icons/icons, images/icons/style. This made the project more interesting because we explored a lot of features in DrRacket, which we wouldn't have seen if we didn't take up this. Also this project is very much different from the previous projects we've heard of. Therefore we wanted to give our best shot by including many features and enjoyable to users.