# Virtual Drum Set

6.111 Final Project Proposal

Rishi Naidu
Luis Fernandez
11/05/2012

## Abstract

It is said that air drumming provides the same level of satisfaction to the user as drumming on a physical drum set does. Knowing this, why should everyone have to own a drum set to start drumming? For our final project, we have the intention of implementing a gesture-controlled virtual drum set on the 6.111 FPGA lab kit to provide a simple and fun way for people who wish to learn or simply enjoy drumming.

More specifically, the users hand motion would be tracked by a camera system. The user wears two distinct color gloves to easy the motion tracking. The motion tracking system will distinguish between hitting motions in different areas in front of the user and emulate sound of distinct drums based on hitting area. At a basic level, the implementation would consist of a limited number of drum set elements and basic playback of sounds of the drums the user plays.

Depending on available time, other features could be added. As an example, a game/learning mode in which the user loads pre-processed songs into the lab kit and air drums in tandem to them. Another idea is to provide the user with an increased number of drum set elements for increased sound options. We expect that by successfully implementing the basic functionality, the system will be robust enough so that a wide range of additional features could be completed easily.

## Overview

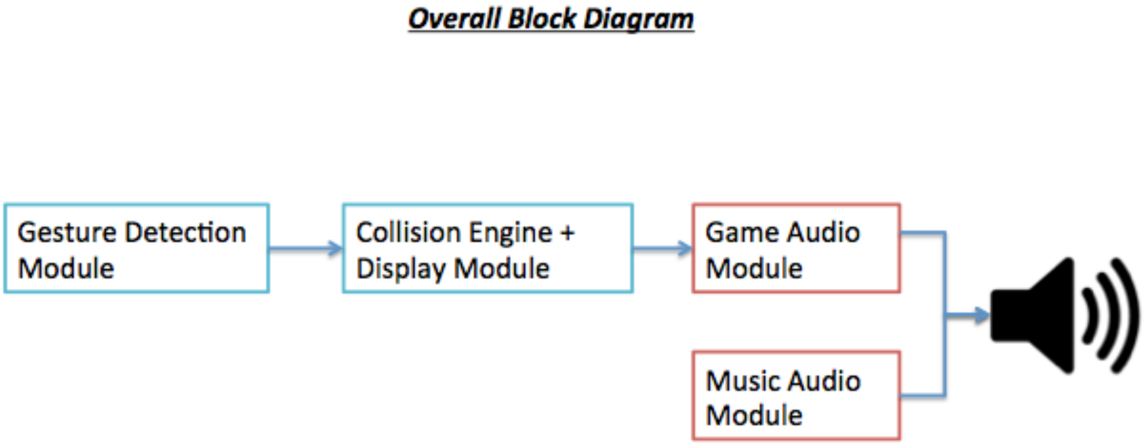
The project is partitioned into roughly two distinct sections: the *video module* and the *audio module*. These two modules interact with each other and as their name suggest deal with the video and audio parts of the project respectively.

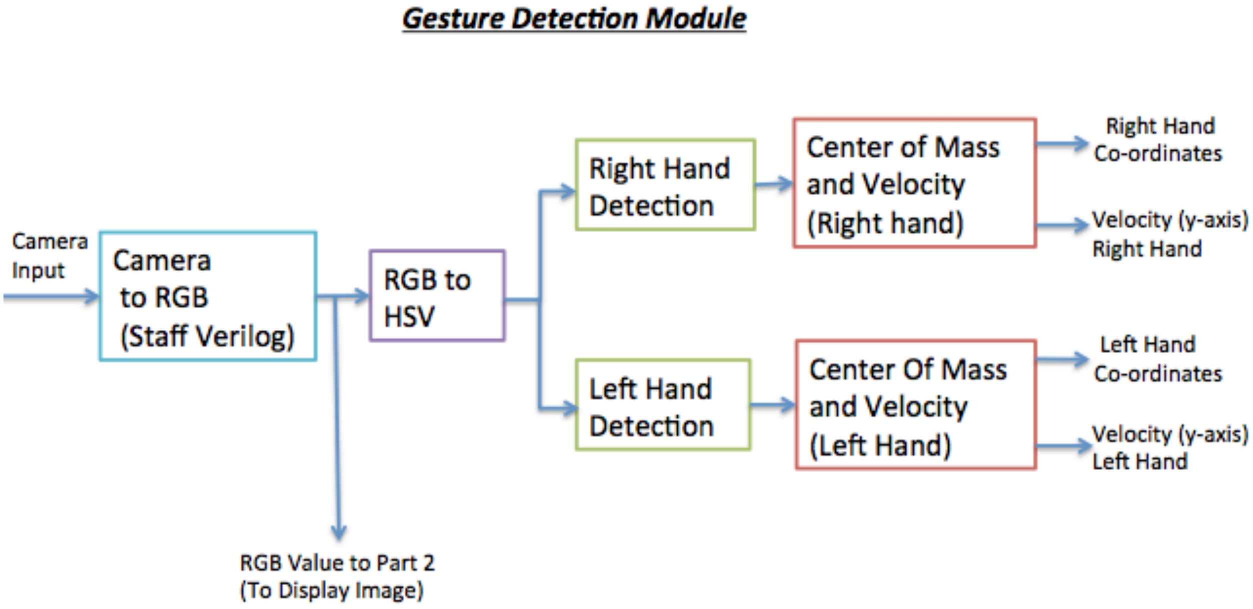
The video module is divided into two separate parts: the *gesture detection module* and the *collision engine and display module*. The *gesture detection module* is in charge of using the input stream of the video camera to detect the position and velocity of the user's hands and output a specific pair of coordinates corresponding to the location of the center of mass of the user's hands. The *collision engine and display module* then takes this information and not only displays it on the screen along with a graphical interface that represents the hands and the drums but also determines at what point a specific drum is being played. In short, this module is responsible for generating the graphical interface of the project and is also responsible with determining what drum is being played at what time by the user.

The audio module is also divided into two separate parts: the *game audio module* and the *music audio module*. The game audio module is responsible for using the output of the video module to determine what drum sound to play at what time and correctly output it to the ac97 interface for playback in the lab kit. The music audio module is responsible for reading a predetermined song from compact flash memory and also interfacing with the ac97 interface for playback in the lab kit.

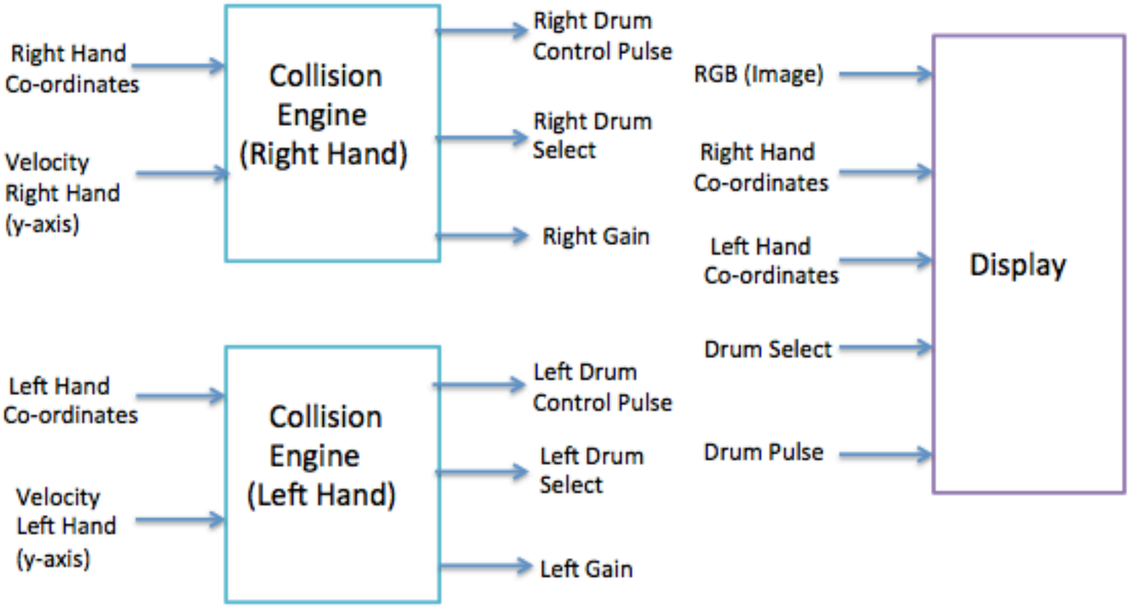
Overall, the user would air drum freely and hear playback from the drums he is virtually playing, along with listening to a song in tandem to his actions. A schematic of our modules is provided below in Figure 1.

**Figure 1. Schematic**

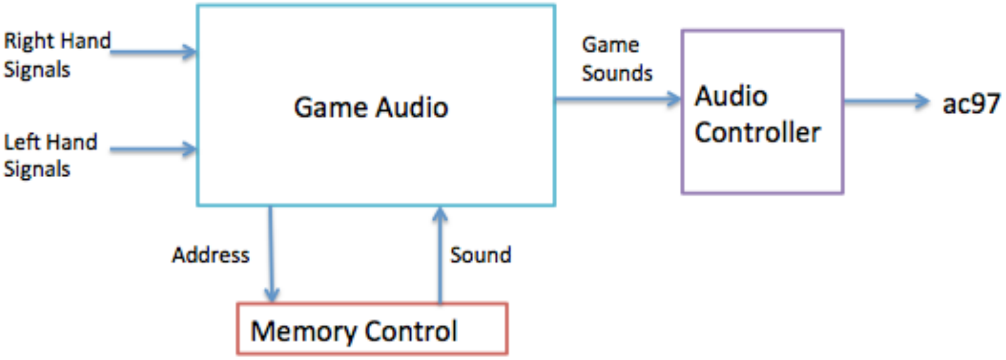



Gesture Detection Module
Camera Input
Camera to RGB (Staff Verilog)
RGB to HSV
Right Hand Detection
Left Hand Detection
Center of Mass and Velocity (Right hand)
Center Of Mass and Velocity (Left Hand)
Right Hand Co-ordinates
Velocity (y-axis) Right Hand
Left Hand Co-ordinates
Velocity (y-axis) Left Hand
RGB Value to Part 2 (To Display Image)

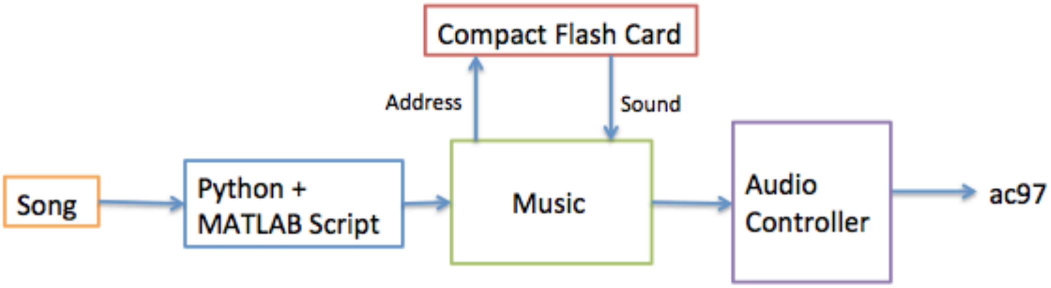
## ***Collision Engine + Display Module***



## ***Game Audio Module***



## ***Music Audio Module***



**Modules**

***Video:***

*Camera to RGB: (Luis)*

This module is provided by the 6.111 staff and is only modified to output color video in RGB rather than the default black and white video.

Inputs: clock, video camera stream
Outputs: RGB bit stream

*RGB to HSV: (Luis)*

This module is provided by the 6.111 staff and converts the RGB bit stream to the HSV protocol.

Inputs: clock, RGB bit stream
Outputs: HSV bit stream

*Hand Detection: (Luis)*

This module is responsible for taking in the HSV stream of bits captured by the video camera and detecting which pixels correspond to the hand of the user. This is done by comparing the hue of the incoming pixels to the specific color of the glove that the user would be wearing. If a pixel is determined to match the color of the glove, then it is sent to the next module.

In the actual project two of these modules would be instantiated, changing a parameter in each to specify which color they are looking to match.

Parameters: glove color
Inputs: clock, HSV bit stream
Outputs: matching pixels

*Center of mass and velocity: (Luis)*

This module is responsible for determining the center of mass of the user's hand and the velocity at which this one is moving. We find such center of mass by using the data from the Hand Detection module and averaging the position of all the matching pixels. The module also keeps track of the previous location of the center of mass and then compares the change in location for every frame in order to calculate the velocity at which the hand is moving.

Inputs: clock, matching pixels
Outputs: co-ordinates of right hand, co-ordinates of left hand, velocity of right hand (y-axis), velocity of left hand (y-axis)

*Collision Engine: (Rishi)*

There are two different collision engine modules, one for each hand. This module is responsible for detecting the hit on the drum and also to determine which drum is being hit. The hit is detected when a hand crosses a specific threshold level with some velocity along the vertical plane. Drum sound is selected based on position of hand along the horizontal plane.

Inputs: clock, co-ordinates of right hand, co-ordinates of left hand, velocity of right hand (y-axis), velocity of left hand (y-axis)
Outputs: Drum Control Pulse (left, right), Drum Select (left, right), Gain (left, right)

*Display: (Rishi)*

This module is responsible for displaying the drum set, hand position and image of user (if required). It will also show the drum being played. These displays will basically be images on the screen, whose motion will be controlled by the user's gesture movements.

Inputs: clock, RGB (image), co-ordinates of right hand, co-ordinates of left hand, gain, and drum select.
Output: combined RGB stream to VGA display

***Audio:***

*Game Audio: (Rishi)*

This module takes in right hand and left hand control signals as input and fetches the necessary sound from memory control. Sound is then processed based on the gain (determined by collision engine module) and fed to the audio controller. The audio controller does necessary sampling and output it to ac97, which plays the sound through speakers. The audio controller is also responsible for correctly 'merging' the game sounds and the music fed into it by the music audio module.

Input: clock, right hand control signals, left hand control signals
Output: Game sound through speakers

*Music audio: (Luis)*

This module is responsible for reading a song that has been stored in compact flash in the lab kit (preprocessed by some MATLAB and Python scripts) and correctly interfacing with the ac97 for playback while playing drums. It feeds the music bit stream into the game audio module in order for it to be 'merged' with the game sounds.

Inputs: clock, music control signal, compact flash out
Outputs: music bit stream

**Schedule**

Week of 11/05:

- Start implementing video modules
- Block diagram conference

Week of 11/12:

- Finish implementation of video modules
- Project design presentation
- Checklist conference
- Start implementing audio modules

Week of 11/19:

- Debug and testing of video modules
- Debug audio modules

Week of 11/26:

- Finish implementing audio modules
- Wrap up video modules testing
- Start testing and debug of audio modules

Week of 12/3:

- Debug all modules
- Test whole design
- Prepare final presentation