

# Sound Blocks

## An interactive environment for real-time music creation

6.111 Final Project Proposal  
Iris Cheung & Dennis Miaw

### PROJECT DESCRIPTION

The purpose of the Sound Blocks is to provide a unique and interactive interface for creating music in real-time. A user is able to mix, match, and modify sounds by placing and manipulating blocks on an interactive table. There are three types of blocks: source, local-effect, and global-effect. A source block generates sound, such as a drum loop or a bass line. Local-effect blocks, when placed near a source block, modify the waveform of the source block, producing effects such as filtering, reverb, modulation, etc. Global effect blocks control properties that affect the entire system, such as volume and tempo. To show the interaction between the blocks, the table will display the audio waveforms resulting from each block, and also the output waveform of the entire system. Sound Blocks was inspired by research exhibited at the 2006 Siggraph convention.

#### Types of Sound Blocks

The first type of block is the “source block.” A source block causes a sound to be produced when it is placed on the table. The audio is played through a set of speakers or an amplifier connected to the labkit. These sounds could be drum loops, bass lines, melodies or musical phrases, hand claps, square waves, etc. Each sound is around four to eight measures long, and will constantly be looping whenever the block is on the table.

The second type of block is the “local-effect block.” A local-effect block modifies the audio signal generated by a source block. Each local-effect block has a specific effect associated with it, and placing the local-effect block near a source block causes the audio signal generated by the source block to be modified accordingly. Examples of some possible effects include: band-pass filters, reverb, delay, phasing, and flanging.

The third type of block is the “global-effect block.” These blocks are similar to the local-effect blocks, except that they do not target a specific source block to modify. These blocks alter parameters which affect the entire system. Global-effect blocks could change parameters such as master volume, or tempo.

In addition to placing blocks on the table, the user is also able to modify the properties of the blocks by rotating them after they have been placed on the table. When a source block is rotated clockwise, the volume of the audio signal associated with the source block increases. Similarly, rotating the source block counter-clockwise decreases the volume of the audio signal associated with the source block. Rotating local-effect and global-effect blocks modify a property unique to each block. For example, rotating a band pass filter block changes the frequency range of the filter, while rotating a delay block adjusts the time of the delay.

#### Interactive Table

The table consists of a translucent, circular tabletop surface, upon which the user places blocks, and the table also houses a video camera and a projector. The surface of the table will roughly be around 3 to 3.5 feet above the ground. The video camera and the projector will both be placed near the floor, underneath the surface of the table, pointed upwards towards the table surface. Each block will have a unique pattern printed on it, and the verilog code will determine the position of the blocks that are currently on the table by looking for these patterns using the video

camera. The patterns will be printed on the bottom of the block, so that when the blocks are placed onto the translucent table surface, the camera will be able to see the patterns from below.

The projector will be used to display images on the table. Since the table surface is translucent, when the projector displays an image onto the bottom of the table from underneath, the user will be able to see the mirror image of the display from above. Therefore, the projector will have to display the image backwards so that the user sees the correct image.

The projector will display audio waveforms connecting source and local-effect blocks currently on the table, so as to provide visual feedback to the user as to what is happening to the sound. On the center of the table is a dot designated as the "output." When a source block is placed onto the table, a display of the waveform of the audio signal associated with the source block will be displayed beginning where the block has been placed on the table and ending at the "output" dot in the center of the table.

When a local-effect block is then placed near the source block, the original and unmodified waveform associated with the source block will be displayed beginning where the source block has been placed on the table and ending where the local-effect block has been placed on the table. Additionally, the waveform of the original sound that has been modified by the local-effect block will be displayed beginning where the local-effect block has been placed on the table and ending at the "output" dot. So, in this manner, the paths of each audio waveform, and the manner in which each waveform is altered by local-effects, will be visually displayed on the table.

The waveform of the summation of all of the audio signals on the table will also be displayed. One slice of the table will be reserved for displaying this waveform. This waveform will start at the "output" at the center of the table, and end at the edge of the table, staying within its pre-portioned slice.

## MODULE DESCRIPTION & PROJECT IMPLEMENTATION

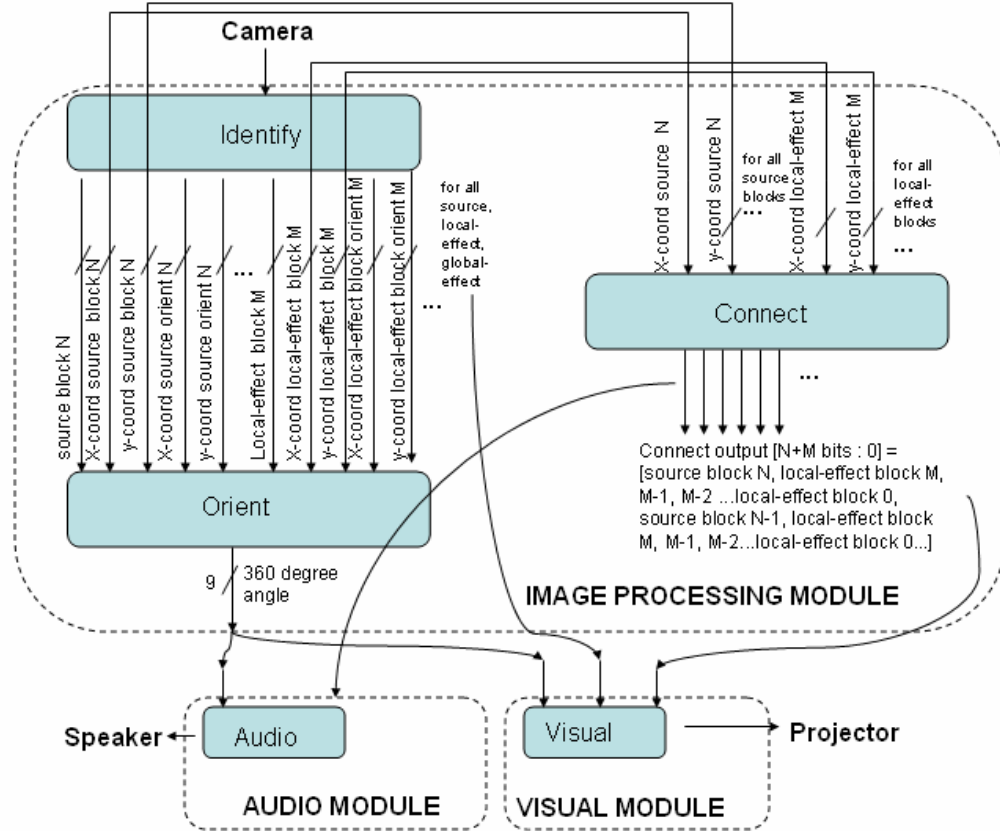


Figure 1: Block Diagram of Sound Blocks Controller

The design approach for this project is broken into three large sections: image processing, audio processing, and video processing.

### Image Processing Module

The image processing module interprets the data received by the video camera to determine which blocks are currently on the table, and the relation of the blocks to each other. It consists of three smaller modules:

#### Identify

This Identify Module receives input from the video camera. The Identify Module scans the incoming frames, trying to locate any patterns. For every block it finds, it outputs a set of bits containing data about the type of block it is, the x and y coordinates of the block, and the x and y coordinates of a special mark embedded in the pattern of the block which is used to indicate the rotational orientation of the block. The Identify Module requires a register large enough to store all of the outputs of the module so that the Orient Module can reference past positions of blocks in order to determine changes in the blocks' orientation. To test the Identify Module we will use the Visual Module to display a graphical representation of the location of blocks that have been placed on the table. Dennis Miaw is responsible for designing this module.

## **Connect**

The Connect Module receives input from the Identify Module. This module receives the x and y coordinates of every source block and every local-effect block. Based on the locations of the blocks, the module determines which local-effect blocks should be connected to which source blocks using a pre-determined algorithm. The algorithm will involve “drawing” a circular perimeter around which intersects the origin of a source block and the output dot. Local-effect blocks which lie within the perimeter of a particular source block will then become associated with that source block. For each source block, the Connect Module outputs a set of bits containing data about the effects (and the order of effects) associated with the source block. Iris Cheung is responsible for designing this module.

## **Orient**

Like the Connect Module, the Orient Module receives input from the Identify Module. This module receives the x and y coordinates of every block on the table, as well as the x and y coordinates of a specific attribute in the pattern of each of the blocks used to determine the blocks' orientation. The purpose of the Orient Module is to output the 360 degree orientation associated with each block on the table. Iris Cheung is responsible for designing this module.

## **Audio Module**

The Audio Module receives data from the Connect Module, in order to know what effects need to be put onto each sound. It also receives data from the Orient Module, in order to know the magnitude/properties of each effect. The Audio Module then retrieves the samples of the appropriate source-block sounds from a memory, and applies the necessary effects to the sound. The output is sent through the AC97 to speakers. We decided to use a tempo of 120beats/min for our estimations of the size of internal memory connected to the audio module. For four measures at 120beats/min, that translates into 8 seconds for a four measure audio file in 4/4 time. In order for the audio to play at 128kbs,  $8\text{seconds} \times 128\text{kbs} = 1024\text{kbits}$  per audio file we store. Dennis Miaw is responsible for designing this module.

## **Visual Module**

The Visual Module receives data from the Connect Module, so that it will be able to correctly connect the different blocks together when it displays the waveform paths on the table. It also receives data from the Orient Module, so that it can correctly represent the magnitude of the effect on the appropriate waveforms. The waveforms for the unaltered sounds will be retrieved from a memory, and then the required effects will be applied. The output is sent to the projector to be displayed on the table. Iris Cheung is responsible for designing this module.

## **EXTERNAL COMPONENTS**

### **Blocks:**

This project uses a number of small blocks as part of its basic concept. The size, shape, and material of the blocks are not critical, but they will likely be either hockey-puck or cube-shaped, on the order of a couple inches in size, and made of plastic or wood. Each block will need to have a distinct pattern printed onto the bottom.

### **Table:**

The table is the surface on which users will place the blocks. The surface of the table needs to be translucent (a piece of clear plexiglass with a thin paper covering, for example), but aside from that, the materials and the construction can be fairly basic. The camera and projector will also be mounted to this structure, near the floor.

**Video camera:**

A camera will be used to detect the blocks placed on the table. The camera will connect to the labkit through the ADV7185 video input chip.

**Projector:**

A projector will be used to display information about the system onto the table surface. Hopefully, we will be able to interface the projector to the labkit through either the ADV7125 VGA output, or the ADV7194 video output.

**Speakers/Amplifier:**

In order to output the audio generated by the system, a set of speakers or an amplifier will need to be connected to the labkit, via the AC97 outputs.