

6.111 Project Proposal  
Humberto Evans  
Alex Guzman

## Bass Hero

Bass Hero is a game implemented on the 6.111 lab kit. To play Bass Hero the user will use a real electric bass hooked up to the lab kit. The idea of Bass Hero is for the user to play along with the bass line of a real song. While playing the game the user will hear the song and see a visual map of which notes he should play at which time on the VGA display. The notes on the display scroll to the left. When the note reaches a marker at the left of the screen, the user should play the note. The object of the game is to keep up with the song and hit as many notes as possible correctly. The user will get continuous feedback about how he is doing with a score that is displayed on the screen. Scoring will be based on the time that the user was playing the correct note times a constant minus the time that the user was playing an incorrect note times a constant. There will be a short period between note transitions where no scoring is tabulated to allow the user enough time physically switch finger positions.

The game will be implemented on the FPGA. The bass guitar will be hooked up to the FPGA via the AC97 chip's 1/8" microphone input. A full song will be stored in memory that will be played back through the AC97 chip when the user is playing the game. A representation of which notes need to be played in the proper order will be kept somewhere else in memory. While the user is playing, a video module will request the next set of notes to be played. It will display that to the user as well as tell a game logic module which set of notes it is asking the user to play. The game logic will compare that to the actual notes being played by the user at the time and assign a score based on how well they match. The score is then passed back to the video module and displayed.

### FFT

The FFT module will be responsible for performing a Fast Fourier Transform of what is being played by the user. It will output the FFT bins and the value for each bin.

### Note Detector

The note detector will be responsible for mapping the output of the FFT module to notes that are being played by the user. The module will take the output from the FFT bin and check to see if a set of bins at the right frequency add up to a certain threshold value. If they do, it decides the note has been played by the user. It does this for the entire octave we are looking at and outputs three 3 bit numbers. Each corresponds to a string and its value is the fret on that string.

### Game Logic

The game logic module is the central logic of the game. It interfaces mostly with the video module to run and time the game. It takes data from the note detector module about which set of notes the user is currently playing, and data from the Video module about which set of notes the user is expected to be playing. The video module also outputs an enable bit which tells the game logic when it should be scoring the user. While the enable bit is high the Game Logic will compare the notes the user is playing versus the notes the user is expected to be playing and adjust the score accordingly.

## Video

The video module will handle the visual representation on the screen. It will keep in registers the next several notes that need to be played and scroll those across the screen to tell the user which notes to play. When a set of notes gets to a region when the video expects the user to be playing a note, it will send that set of notes to the game logic as well as set the enable signal high. As long as the enable is high, the game logic will be scoring the user.

## Template

The template module will hold the sequence of notes the user is expected to play for the entire song. It will be one large memory where 9 bit words are held corresponding to the three strings and a fret number for each string. When the video module needs another note it will request it by asserting the next note signal.

## Song

The song to be played will be kept in memory. The song module will send the song to the AC97 chip when it is enabled by the FSM.

## FSM

The FSM handles the current state of the game. At first this module will only handle the pausing and restarting of the game. It will have two states, a playing and paused state and will switch back and forth at the push of a button. On playing state it will assert the enable button to all modules, in the paused state it will set the enable low, which will freeze all modules. If time allows, we will make more than one song available to the user. The FSM will have a startup state where the user can select a song to play. It will send this information to the template module and move to the playing state.

## Event Flow of the Game:

FFT and Which Note module constantly give to the Game Logic which notes the user is playing.

The Video module essentially dictates the timing of the game, it requests the next set of notes from the Template module when it needs it. It then displays it on one side of the screen and begins scrolling it across. When the notes get to the other end of the screen, and the video module deems that the user should be playing the note, it sends to the game logic a high enable signal along with which notes the user should be playing. As long as the enable signal is high the game logic will be adding or subtracting from the score. This way if the user holds the note too long, he will + points for the time he had the note right and – points for the time he had the note wrong. The song will be playing over the AC97 the whole time.

## Task Distribution:

- Humberto
  - FFT
  - Note Detector
  - FSM
  - Song
- Alex
  - Video

- Template
- Game Logic

