

6.111 Project Checklist

Rahul Yesantharao

Jacob Pritzker

Commitment:

Our commitment is to have two major modes in our project:

First, it will have a keyboard mode, in which a user can play a full piano-scale of notes, using the switches on the Nexys4 to choose which note to play. Sound will be generated by the Nexys4 and can be heard with headphones out of the audio jack. This will be demonstrated by playing different notes by flipping switches.

Second, it will have a game mode, in which a user tries to match a song (with only one note played at any one time). The desired notes will be displayed on the Nexys4 LEDs, and the user will try to flip the corresponding switches as quickly and accurately as possible in order to play the song. The user's final score will be displayed on the LED segment display on the Nexys4. This functionality will be demonstrated by playing a game and seeing the final score displayed.

Major Modules:

- Keyboard: takes user input from switches, outputs sine wave through audio jack
 - Demonstration: flip different switches, hear different notes playing
- Game Controller: Runs the whole game by maintaining an FSM of current state and score
 - Demonstration: Play the game and see score at the end
- Song Storage Module: Serially outputs notes of the song after a start signal.
 - Demonstration: Accurate notes showing up on LEDs.

Goal:

Our goal is to improve our baseline goals in the following manners:

First, instead of using switches on the Nexys4 to play different notes, the user will play notes on a MIDI keyboard, which will communicate using UART with the Nexys4. Audio will then be heard from the keyboard, rather than the audio jack on the Nexys4. This functionality will be demonstrated by playing the keyboard.

Second, the game display will use the lab monitor, instead of the LEDs on the Nexys4. Specifically, the monitor will show notes falling (with only one note played at a time), indicating which note a player needs to play when. The user's final score will still be displayed on the LED segment display on the Nexys4. This functionality will be demonstrated by playing through a game.

Third, a menu to select between the keyboard mode and the game mode will be displayed on a lab monitor, with the user able to select between them either with buttons on the Nexys4, or by singing a high vs. low note into a microphone. Similarly, a menu to select which song to use in

the game mode will be displayed on a lab monitor. This functionality will be demonstrated by showing that menu navigation can be accomplished either with buttons or with singing.

Major Modules

- Menu Module: Provides a standard interface for any menu, and allows the user to choose between options using the FPGA buttons or singing into the microphone.
 - Demonstration: navigate menu both with singing and with FPGA buttons
- VGA Interface: Takes in a set of standard signals representing various display states in the game and menus and constructs the appropriate output.
 - Demonstration: menus displayed properly; falling notes (one note played at a time) during gameplay
- MIDI Interpreter: takes in MIDI output from keyboard, interprets messages to determine which note is currently being played
 - Demonstration: light up LEDs on FPGA corresponding to which note is currently being played on keyboard
- Microphone Note Analyzer: Uses the FFT hardware to determine whether a low or high note is being hummed (for menu navigation).
 - Demonstration: Proper menu navigation by humming.
- Song Select: A further refinement on the Song Storage module above, allows for selecting a song to be read from memory.
 - Demonstration: Use the song selection menu to choose different songs.
- Game Controller: same as in Commitment level

Stretch Goals:

Our stretch goal is to accomplish our expected goal, with three additional modes:

First, we hope to create a singing mode for our game, whereby a user can sing into the microphone, instead of playing the keyboard, in trying to match notes. This mode will be demonstrated by playing a singing-mode game.

Second, we hope to create a song-creation mode, in which a player can play the keyboard in order to design their own song that can then be used in the game mode, as opposed to only being able to use the pre-selected songs. This will be demonstrated by creating a new song, and then using that song in the game mode.

Third, we hope to create a learning mode, in which a player can learn one of the songs instead of being scored for it in the game mode. Specifically, this mode will show the notes a user should play for the song, but will wait until the user has hit that note before moving on, thus teaching the user this song. This will be demonstrated by “learning” one of the song options.

Major Modules

- Microphone Note Analyzer: extension of same module from expected goal level; uses the FFT hardware to determine which note the user is currently singing into the microphone
 - Demonstration: light up LEDs on FPGA corresponding to which note is currently being sung

- Song Creation Module: Take in notes from the keyboard and save them to BRAM so that the player can later play these songs in game mode.
 - Demonstration: Create a song and then use it in game mode.
- Song Learning Module: Provides an interface similar to the game controller, but does not score and does not progress in the song until the player gets the current note correct.
 - Demonstration: Learn a song.