

Final Project

Abstract

We created a musical game similar to Guitar Hero or PianoKeys. The game has squares that come down from the top of the screen in 4 columns, signifying notes. When they reach a marked area in the bottom of the screen, the player must press a button on the numpad corresponding to the column that the square is in. If they press the button at the right time, the next note in the song “Never Gonna Give You Up” by Rick Astley is played. If they miss the note, an error sound is played and the player’s lives are decremented. When the player’s lives run out, the game ends and the reset button must be pressed to restart the game. Whenever the song finishes, it loops from the beginning with the speed increased. The 7 segment display keeps track of the amount of time the player has been alive, representing their score.

High Level Components

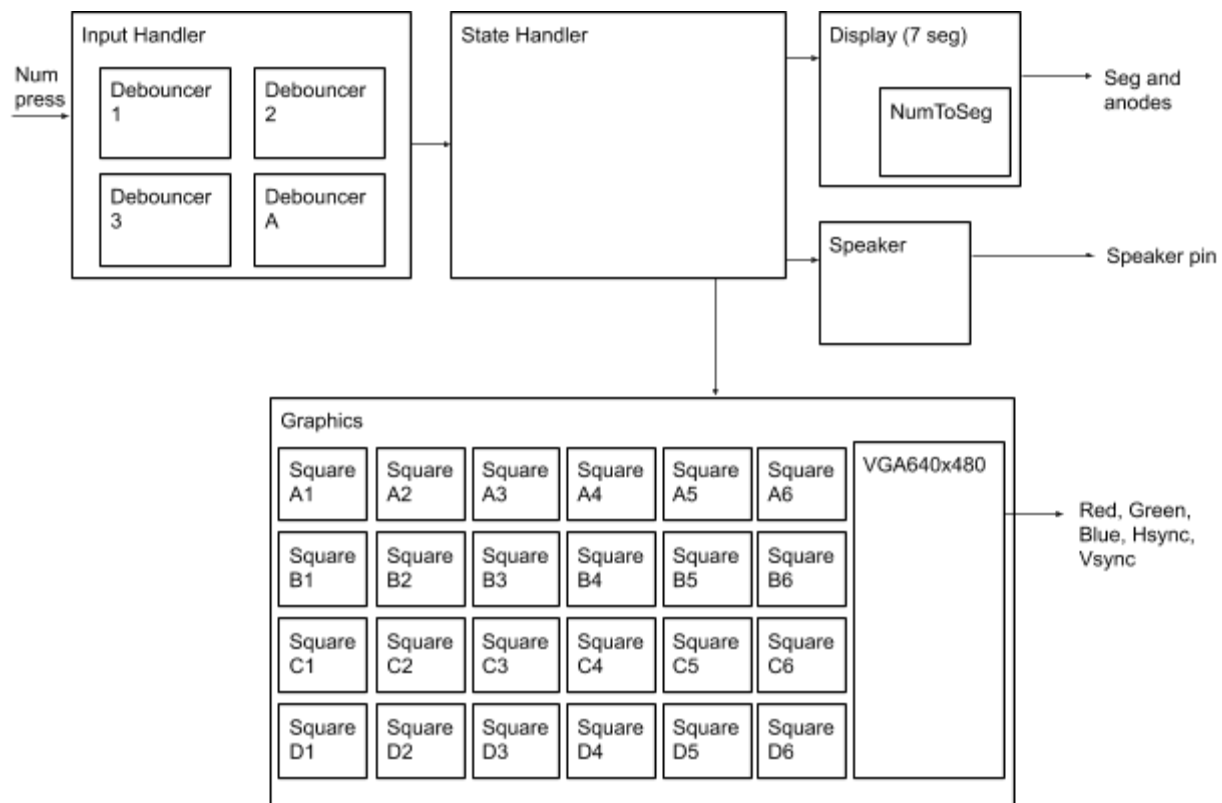


Figure 1: Modular Design: The game is designed with 5 main modules, the input handler, the state handler, graphics, the 7-seg display and the speaker module.

The input handler takes input from 4 buttons on the numpad and debounces them, and converts the button press into a 4 bit instruction to the state handler, where each bit signifies a button being pressed in that column.

The state handler maintains the state of the game, including the player's lives, the progression through the song, and checking if the notes were played correctly or not. It also receives the current state of each square from Graphics, which are between lines, at the end of a column, or neither. It outputs the number of lives left to the leds, displays the current score, sends an instruction to display a square in graphics when necessary, and sends a frequency to the speaker module to play the corresponding note sound when a button is pressed correctly, or a bad note sound if the button press is incorrect. If the player runs out of lives, the game is recorded as ended in the state handler and a corresponding instruction is sent to the graphics module.

The graphics module takes a 4 bit instruction from the state handler signifying which column to create a new square in. There are 6 possible squares that can exist in each column (A, B, C, or D), and the module displays them as necessary and hides them when not in use. This module is based on the VGA module in the sources section below.

The speaker module receives a frequency, and plays the corresponding note of that frequency for 1s. A simple square wave is used to produce a note through the speaker. If a note has not finished playing but another note is sent to the speaker from the state handler, the speaker will play the new note instead.

The 7-seg display module is taken from project 3 with minor modifications, as it performs the same functionality as the 7-seg display in project 3. It displays the current score of the player, and if the game ends, the current score of the player is fixed on the display.

A speaker module and a variable resistor is connected to pin 1 and the ground pin of pmodA, and a numpad is connected to pmodB of the board. The first column of the numpad is used as the buttons corresponding to each column in the screen.

Sources:

VGA Module based on: <https://timetoexplore.net/blog/artty-fpga-vga-verilog-01>

Numpad Input Module based on:

<https://reference.digilentinc.com/reference/pmod/pmodkypd/start>