FPGuitAr Hero

Sarah Spector, Alejandro Diaz
Proof of intellectual property

To: Alejandro Diaz

10/17/19, 1:52 PM

Do you know the game Rock Band
What even is your game?

- Notes fall down from the top of the screen
- User moves their hands to hit notes (camera tracking of colored patches/LEDs, audio feedback)
- Lots of room for gameplay development
Camera interface + image processing

- **Set thresholds**
  - up/down/left/right buttons select threshold to be set (LH, RH, LF, RF)
  - sw[7:0] set upper thresh, sw[15:8] set lower thresh

- **Buttons**
  - buttons 4

- **sw[15:0]**
  - sw[15:0] set thresholds (LH, RH, LF, RF)

- **Camera read**
  - pclk 16
  - vsync 16
  - href 16
  - pixel 8

- **RGB to HSV**
  - output_pixels 16
  - valid_pixel
  - frame_done_out

- **Threshold**
  - Determine if pixel meets each of 4 thresholds

- **Hex display**
  - 16

- **Thresholds**
  - LH, RH, LF, RF

- **Set BRAM write addresses**
  - Write address, valid_pixel, pclk

- **Erosion**
  - 3x3 kernel
  - Separate + parallel for each of 4 channels

- **Frame buffers**
  - 4 BRAM frame buffers, each holding binary values

- **Centroid detection**
  - Average x, y values of thresholded, eroded pixels
  - x 9
  - y 8
  - * 4 (LH, RH, LF, RF)
Threshold control

- Hue (HSV) thresholding
- Set separate thresholds for each of 4 limbs
- Use buttons, switches to control

Set thresholds
- up/down/left/right buttons select threshold to be set (LH, RH, LF, RF)
- sw[7:0] set upper thresh, sw[15:8] set lower thresh
Hue thresholding + buffering

- Fast RGB -> HSV (6-13 cycles)
- Threshold separately for each of 4 hue ranges
- Four BINARY frame buffers
Erosion + centroid detection

- Erode with 3x3 kernel
- Binary buffers
- Avg x, y val of resultant pixels
- Separate + parallel image processing for each of 4 channels
Gameplay Logic Module
Display on Monitor

- Have to create all note blobs from the start
- BPM, Timbre, Pitch, Note Length
- Note = 4 bits for pitch and 2 for length
- Falling speed depends on BPM
Note Hand Interactions

- Pixel overlap on a frame by frame basis to detect Intersections
- Detect note hit by intersection of a hand and the front end of a note and x, y position
- Note Intersections on a frame by frame basis along a horizontal axis
- Score is a function of its inputs (initially on hex display)
Audio Player and Mixer Module

- Naive Audio mixing is just adding and dividing by n
- 13 channels feed into Audio Mixing
Minimum viable product

- Accurate x, y tracking of at least 1 colored patch/light (hand)
- Notes streaming down screen based on a stored song
- Notes play on intersection with hands (audio output)
- Score displayed
Stretch goals

- Menu settings selection
- Tracking hands AND feet
- More sophisticated image processing—no colors?
- Hand y-coordinate to control volume
- Variable gameplay speed
- Gameplay add-ons: visual effects streaks, multipliers, multiplayer, levels, gestures, etc.
- Load a song from SD card (if needed)
## Timeline

<table>
<thead>
<tr>
<th>Due date</th>
<th>Task Description</th>
<th>Responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/11</td>
<td>Implementation of music storage, notes falling and playing when hit by cursors moved by buttons, multiple notes playing</td>
<td>Alejandro</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sarah</td>
</tr>
<tr>
<td>11/18</td>
<td>More complicated songs playing with perhaps some extra gameplay features/graphics</td>
<td>Alejandro</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sarah</td>
</tr>
<tr>
<td>11/25</td>
<td>Done with all essential features for basic gameplay- basic song playing with according notes playing</td>
<td>Alejandro</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sarah</td>
</tr>
<tr>
<td>12/2</td>
<td>Addons and extra features to be added</td>
<td>Alejandro</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sarah</td>
</tr>
</tbody>
</table>

- Alejandro's tasks include implementing music storage, handling notes falling and playing, and more complicated songs with additional gameplay features. By 11/25, all essential features are completed, and by 12/2, add-ons and extra features are added.
- Sarah's tasks consist of completing the camera interface, handling HSV conversion and thresholding, erosion/dilation, and tracking specific patches. By 11/18, additional gameplay features are added, and by 12/2, improvements to the tracking algorithm are made.
Questions?