Fpglappy Bird

Wei Low, Nicholas McCoy, and Julian Mendoza
Problem

- popular, difficult and frustrating game
- software implementation for mobile devices
- “gone forever”
Overview

- Implement on NEXYS 4
- OV7670 camera points at player and tracks “beak” on their face
- Game images built from predefined sprites
- Sound effects from SD card
- Player jumps => bird jumps
Design

Video In

Vision Processing Block

Game Logic Block

Video Block

Audio Block

VGA Out

Audio Out

Button Input
Implementation: Vision Processing Block

Video In

Preprocessing

Object Identification

Kalman Filter

Player coordinates

Video Frame

BRAM
Implementation: Vision Processing Block

Video In

Preprocessing

Vision Processing Block

Object Identification

Kalman Filter

Player coordinates

Video Frame

BRAM
Preprocessor

- Gray-scale Conversion
- Thresholding
- Noise Filtering
Preprocessor

- Gray-scale Conversion
- Thresholding
- Noise Filtering
Preprocessor

- Gray-scale Conversion
- Thresholding
- Noise Filtering
Preprocessor

- Gray-scale Conversion
- Thresholding
- Noise Filtering
Implementation: Vision Processing Block

Video In → Preprocessing → Object Identification → Kalman Filter → Player coordinates

Video Frame → BRAM
Implementation: Vision Processing Block

Video In

Preprocessing

Object Identification

Kalman Filter

Video Frame

Player coordinates

BRAM
Implementation: Gameplay Logic Block

Gameplay Logic Block

- Player coordinates
- Button input
- Game State
- Physics
- High Scores
- BRAM
- SD Card
- Obstacle location
- Face pixels
- Audio selection
- Save temp highsco
Implementation: Game States

- Start
- Play
- Pause
- Lose
- High Score

- After 3 seconds, enter button to return to start state.
- "CONTINUE" to restart from play state.
- "END" to return to lose state.
Implementation: Video Block

Video Memory (for player face)

Sprite Pipeline

Sprite Image Memory (BRAM)

Load at startup

SD Card

Object Locations

VGA Controller

VGA Out

- Takes input of sprite locations from game logic block
  - Player sprite, background, up to 3 obstacles
- Sprite pipeline like in Lab 2
- Textures from SD card
- Takes input of a picture of the player's face from the vision processing block and uses it as a sprite for the player
Implementation: Audio Block

- Takes enable signals from the Game Logic block for each audio track
  - Jumping sound effect
  - Crashing sound effect
  - Background music
- Sounds files loaded from SD card as they are played (.wav format)
- Uses mono audio DAC onboard the NEXYS 4

Diagram:

- Audio Controller
  - "Jump" FSM
  - "Crash" FSM
  - Music FSM
- DAC
- SD Card (.wav files)
- Audio Out
- Track enable bits
- Load tracks as they play
## Schedule

<table>
<thead>
<tr>
<th>Task</th>
<th>11/1</th>
<th>11/8</th>
<th>11/15</th>
<th>11/22</th>
<th>11/29</th>
<th>12/06</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface with FPGA</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Tracking Module</td>
<td>J</td>
<td>J</td>
<td>J</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio/Video Module</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game Logic Module</td>
<td>W</td>
<td>W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preliminary Testing</td>
<td>All</td>
<td>All</td>
<td>J</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration: Game Logic, Audio Video</td>
<td>N, W</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration: Object Tracking</td>
<td>All</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing</td>
<td></td>
<td></td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buffer Time/Stretch Goals</td>
<td></td>
<td></td>
<td></td>
<td>All</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demo/Final Presentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>All</td>
<td></td>
</tr>
</tbody>
</table>
Complexities

- Memory management during image processing
- Noise management
- Distributing access to SD card
Stretch Goals

- Multiple FPGAs running the game and comparing high scores
- Sprite rotates as it jumps
- Multiplayer version