Charlie Devivero
Javier Garcia

Space Force Duo
Overview

- Two-Player space shooter game
- Two FPGA’s connected through Serial port for two-way communication
- Features voice-communication
- Log into the game using RFID tags

http://www.shockinglyfun.com

http://www.harrowaudiohifi.co.uk
Networking Module

- Packetizes incoming data.
- 2 byte Header + body

### Some Packet Types:

<table>
<thead>
<tr>
<th>Game Data:</th>
<th>16 bytes</th>
<th>128 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>11 bits</td>
<td></td>
</tr>
<tr>
<td>Y1</td>
<td>10 bits</td>
<td></td>
</tr>
<tr>
<td>X2</td>
<td>11 bits</td>
<td></td>
</tr>
<tr>
<td>Y2</td>
<td>10 bits</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>X6</td>
<td>11 bits</td>
<td></td>
</tr>
<tr>
<td>Y6</td>
<td>10 bits</td>
<td></td>
</tr>
<tr>
<td>Unused</td>
<td>2 bits</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Audio Samples:</th>
<th>8 bytes</th>
<th>64 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample 1</td>
<td>1 byte</td>
<td></td>
</tr>
<tr>
<td>Sample 2</td>
<td>1 byte</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Sample 7</td>
<td>1 byte</td>
<td></td>
</tr>
<tr>
<td>Sample 8</td>
<td>1 byte</td>
<td></td>
</tr>
</tbody>
</table>

Frame Rendering

Host Data

Client Data

Frame Buffering
Audio Module

- Microphone samples from AC97 are recorded, downsampled to 6kHz, and passed through 3kHz low-pass filter.
- Samples sent through network.
- Incoming samples are played back.
- Samples are 8-bits wide.
Serializer/Deserializer

- Transmits/Receives data serially using RS-232 scheme
- Data rate of 200 kbps

http://www.wikipedia.org
Master FSM/Game Logic

- Determines the Host and the Client
- Control the state of the game
- Determine collisions and coordinates of all sprites
- Game runs at 1024x768 @ 60Hz
- Color information is compressed to 3 bits for RGB.
- Early version will render sprites as mono-colored rectangles.
- If time permits, detailed image sprites will be rendered instead.
RFID Module

- RFID Reader
- Asynchronous receiver (RS-232):
- Database: match 10-byte ID number to a user name

http://www.parallax.com/
Major Challenges

- Information flow control; preventing buffer overflow
- Handling packet corruption
- Bandwidth limitations
- Reading/writing to frame buffer fast enough
- Hiding unused sprites
- Scalability of maximum sprites
- Interfacing with the RFID reader
<table>
<thead>
<tr>
<th>Week</th>
<th>Charlie (Audio/Network)</th>
<th>Javier (Game/RFID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/16</td>
<td>Packetization of audio; serial transfer</td>
<td>One-Player game with 5 enemy ships</td>
</tr>
<tr>
<td>11/23</td>
<td>Game data packetization; bi-directional serial communication</td>
<td>Two-Player game with 4 enemy ships; frame buffering</td>
</tr>
<tr>
<td>11/30</td>
<td>RFID login information packetization, debugging</td>
<td>RFID login using generic transponders, scale number of enemies and bullets, add levels.</td>
</tr>
<tr>
<td>12/07</td>
<td>System integration; debugging; polishing</td>
<td>System integration; debugging; polishing</td>
</tr>
</tbody>
</table>