Laser Pinball

WESTON BRAUN – JAKE ISENHART – PAULINE VARLEY
6.111 FALL 2014
Project Overview

- Reimagined arcade game
- User-customizable
- Three-block implementation:
  - Image processing and camera interface (gameboard recognition)
  - Physics and game engine
  - Laser display driver
Camera interface

- Reads camera image to develop basic game board
- Camera interface (VGA, I2C)
  - Preset camera for correct output (RGB vs chrominance)
  - Camera can be driven between 10 and 48MHz
  - Camera’s internal pixel clock based off of system clock; camera interface module can read based on pixel clock
- Memory interface (frame buffer)
- 32-bit interface with game engine to transmit static object locations
Game Engine & Physics

*Sprite Lookup*

- 8-bit sprite ID – allows for 256 game elements
- Laser projector module will have vector graphics defined for each sprite

*Collision Detection*

- Collisions with different objects will set specific flags visible to the physics module

*Friction and Gravity*

- Operations carried out whenever updating the frame, regardless of collision status

*User Input*

- Game will be controlled by a pair of accelerometers mounted to gloves
64 bits per sprite
- [7:0] Sprite ID
- [2:0] Color
- [4:0] Update count
- [23:0] reserved – may later be used for scaling, rotation
Laser projector display

- Laser projector consists of a RGB laser and galvanometers to steer the beam
- Limited number of points can be plotted in each frame
  - Variable frame rate
- Vector graphics engine translates sprite locations and coordinate sets from the game engine to vectors
- Path finding module optimizes the plotted path
- Galvanometer position set with SPI DAC, laser color set over a parallel interface.
## Timeline

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal and planning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source components</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camera interface and object recognition</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physics design and development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camera and game engine interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Galvo testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game engine and display module interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pathfinding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bonuses (calibration, sound effects, accelerometers)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Debugging and final touches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contributors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pauline</td>
</tr>
<tr>
<td>Jake</td>
</tr>
<tr>
<td>Weston</td>
</tr>
</tbody>
</table>