Futuristic Pepper’s Ghost Approximation (FPGA)

Jeremy McCulloch, Adam Potter, Sreya Vangara
Inspiration

- Pepper’s Ghost tabletop simulation
- Render from the user’s perspective
Our Take

- Transform and project onto table
- Track user, re-render to simulate object
Functionality Goals

Minimum: Projection rendered from multiple perspectives
Target: User tracking and render updating
Reach: User interaction OR animation
BLOCK DIAGRAMS
Computer Vision
Render and Projection
Hardware

- Camera
- VGA Projector
- Projector and Camera Mount
- Green Hat

Limitations

- Board memory, frame rate, and camera specs (FOV)
MODULES
Computer Vision

- Input: Camera Data
- Output: \((x,y)\) of user’s head
- Use chroma keying to pick out green hue
- Use erode and dilate to find largest green blob
Projection

- Calculate where to draw a point on table based on user’s position and model position
- Adjust brightness of triangle based on angle of plane to user
Rasterize

- Use projection to map all vertices of triangle
- Iterate points in triangle, interpolate $z$ coordinate
- Framebuffer to store RGB and $z$ coordinate for each pixel
Timeline

- **Rasterize**
  - Yesterday!

- **Projection**
  - 11/17

- **Shading**
  - 11/17

- **CV**
  - 11/17

- **Integrate**
  - 11/24
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeremy</td>
<td>Rasterize + Framebuffer</td>
<td>Render Simple Model</td>
<td>Integration</td>
<td>Attempt to Add Animation</td>
</tr>
<tr>
<td>Adam</td>
<td>CV Prototyping</td>
<td>Shader, Better Projection</td>
<td>Integration, Mounting</td>
<td>Demo</td>
</tr>
<tr>
<td>Sreya</td>
<td>Camera working</td>
<td>Computer Vision</td>
<td>Integration</td>
<td>Add Wireframe</td>
</tr>
</tbody>
</table>