OPERAcraft + Kinect = Cinemacraft

Client: Ivica Ico Bukvic, Institute for Creativity, Arts & Technology (ICAT)

Members: Brittany Barnes, Elsi Godolja, Marina Kiseleva

CS 4624, Virginia Tech, Blacksburg VA.

Spring 2016
OPERAcraft: Combines video games and opera to produce live productions

Opera + Minecraft = OPERAcraft
OPERAcraft Controls

- Limited to the keyboard and predefined controls

*avatar not pictured
Kinect Sensing + OPERAcraft = CINEMAcraft

- Allows for production creation in real-time from unique perspectives
- **Goal**: Get Minecraft avatar to ‘mirror’ user’s facial and skeletal movements via Kinect sensors
Software

- Kinect code
  - Language: C#
  - Environment: Visual Basic (64 or 32 bit)

- Pd-L2Ork

- Minecraft code
  - Language: Java
  - Environment: Eclipse Luna/Mars (64 or 32 bit)
<table>
<thead>
<tr>
<th>Integral Feature</th>
<th>Base Level of Functionality</th>
<th>Does it need to be expanded?</th>
<th>Could it be expanded?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Skeletal:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arm movement</td>
<td>complete</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>leg movement</td>
<td>complete</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>torso movement</td>
<td>complete</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>head movement</td>
<td>complete</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Facial:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eyebrow movement</td>
<td>in progress</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>mouth movement</td>
<td>in progress</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Minecraft:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar art</td>
<td>complete</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>world art</td>
<td>complete</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Kinect → Pd-L2Ork → Minecraft
Skeletal Recognition
Translating Kinect data -> OPERAcraft movement

Kinect Skeletal Tracking

Infer points

Calculate angles of rotation

Send angles to OPERAcraft

UDP Packet:
@karm
ICAT01 24 36
...
Angle Calculations

❖ Compute angles of rotation in each plane
❖ Points -> 3D Vectors -> compute angle between
❖ *Around Y axis*: rotation in X plane
❖ *Around Z axis*: rotation in Y plane
❖ *Around X axis*: rotation in Z plane
Facial Recognition

Eyebrow and Mouth Combination Matrix. The 20 possibilities for avatar helmets, based on the user’s facial expressions.

Tracked Animation Units (AUs). The areas of 2D points that would affect various AUs to yield the delta values from a face’s neutral position (4).
Kinect → **Pd-L2Ork** → Minecraft
# UDP Packet Constraints

<table>
<thead>
<tr>
<th>Delimiter(s)</th>
<th>Line Ending(s)</th>
<th>Avatar Movement Keywords</th>
<th>Movement Position Value(s)</th>
<th>Movement Position Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>whitespace</td>
<td>\n</td>
<td>eyes, mouth, head, shoulder, arm, leg, torso</td>
<td>{0-3}, {0-4}, {(-90)-(90)}, {((-180)-(180))}, {(y)}, {(-180)-(180)}, {((-180)-(180))}, {(x)}, {(y)}, {(-180)-(180)}</td>
<td>integer, integer, float (xrot, yrot), float (y), float (xrot, zrot), float (xrot, zrot), float (x, y, yrot)</td>
</tr>
</tbody>
</table>
Pd-L2Ork: Averages/Normalization

- Feature position data gets sent from Kinect for each frame
- Pd-L2Ork averages the last five frames to reduce jumpiness in expressions/gestures
- Sends the “patched” data to Minecraft
Kinect → Pd-L2Ork → Minecraft
Minecraft Code: Dependencies and Flow

1. printChatMessage() OperaCraftPlayer.java
2. GuiInGame.java (draws HUD) set player location POSITION
2. ModelBibed.java ROTATIONS

Kinect
→ pd-L2Ork
(add username) @view 1
→ OpenSocketThread.java
sendChat -> network
→ GuiNewChat.java

1. GuiInGame.java (draws HUD) set player location POSITION
2. ModelBibed.java ROTATIONS
Minecraft Art: Facial Expression Helmets

- Microsoft Paint
- 16x16 pixels
- Female and male

Sad
- Anxious brows
  + frown

Surprised
- Surprised brows
  + open lips

Angry
- Furrowed brows
  + frown

Kiss Face
- Neutral brows
  + puckered lips
Minecraft Art: Avatar Skin

- Microsoft Paint
- 32x64 pixels
- Female and male

800% zoom

100% zoom
Minecraft Art: Background World

- In-game with various block types
Prototype

1. Primary display monitor
2. Computer to process skeletal movements
3. Computer to process facial movements
4. Kinect sensor for skeletal tracking
5. Kinect sensor for facial tracking
SXSW Prototype Demo

Video: https://drive.google.com/file/d/0B12jba-1Ut5jT2pPYTZoOExNWDg/view
QUESTIONS?
Resources

[Image] https://hidale.com/shop/dp-kinect/

[Image] http://vector.me/search/walking-stick-figure

[Image] https://www.icat.vt.edu/funding/operacraft