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

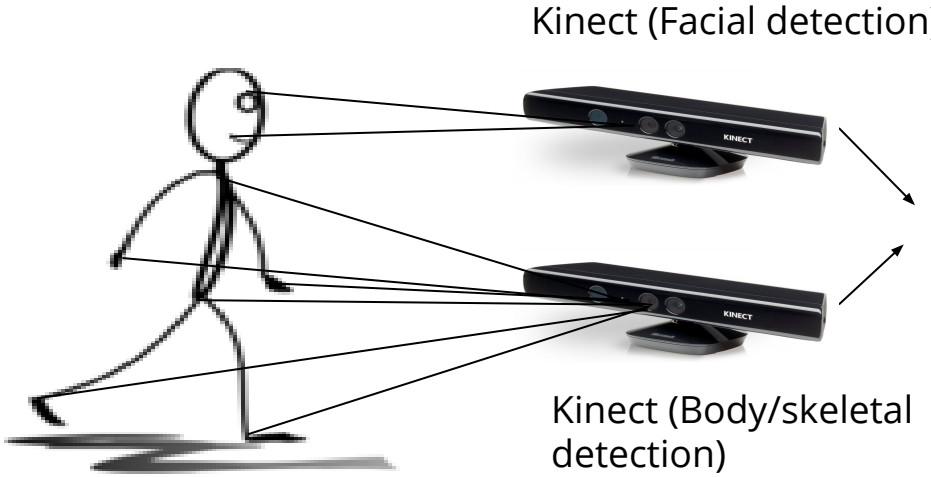
- OPERAcraft: Combines video games and opera to produce live productions
- Opera + Minecraft = OPERAcraft

OPERAcraft Controls

- Limited to the keyboard and predefined controls

*avatar not pictured

Kinect Sensing + OPERAcraft = CINEMAcraft



Pd-L2Ork →

OPERAcraft Visual Display

- 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

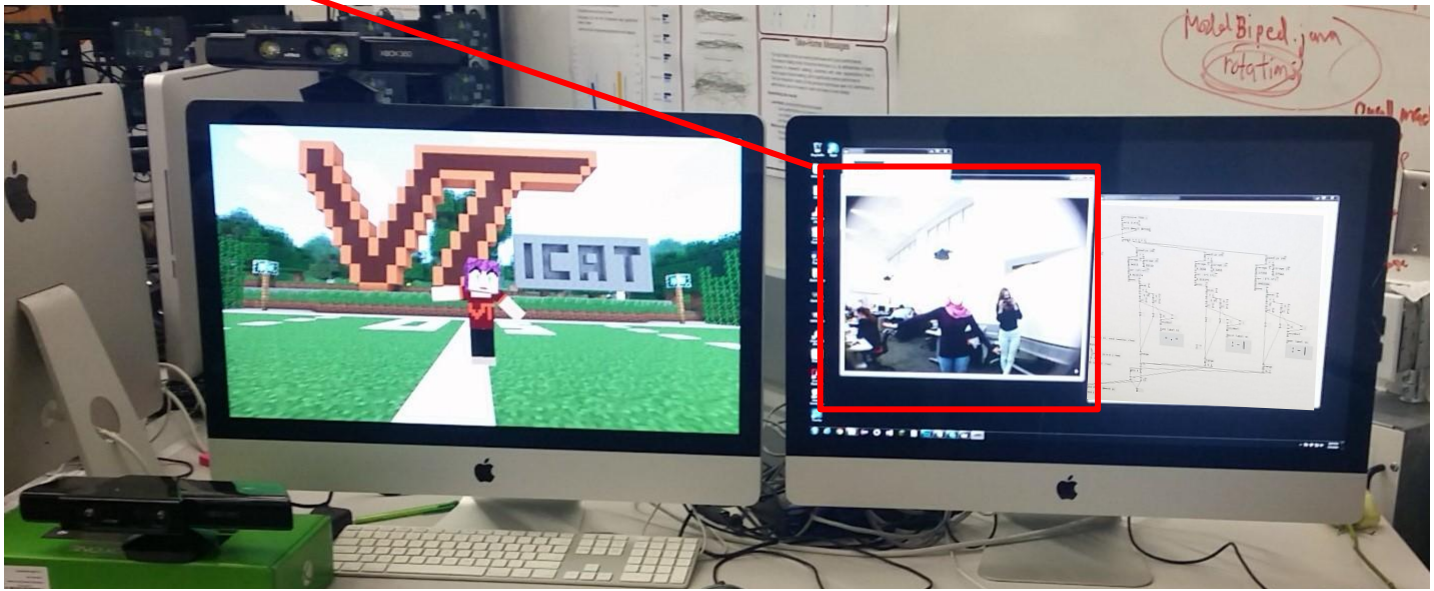
Software

- 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)

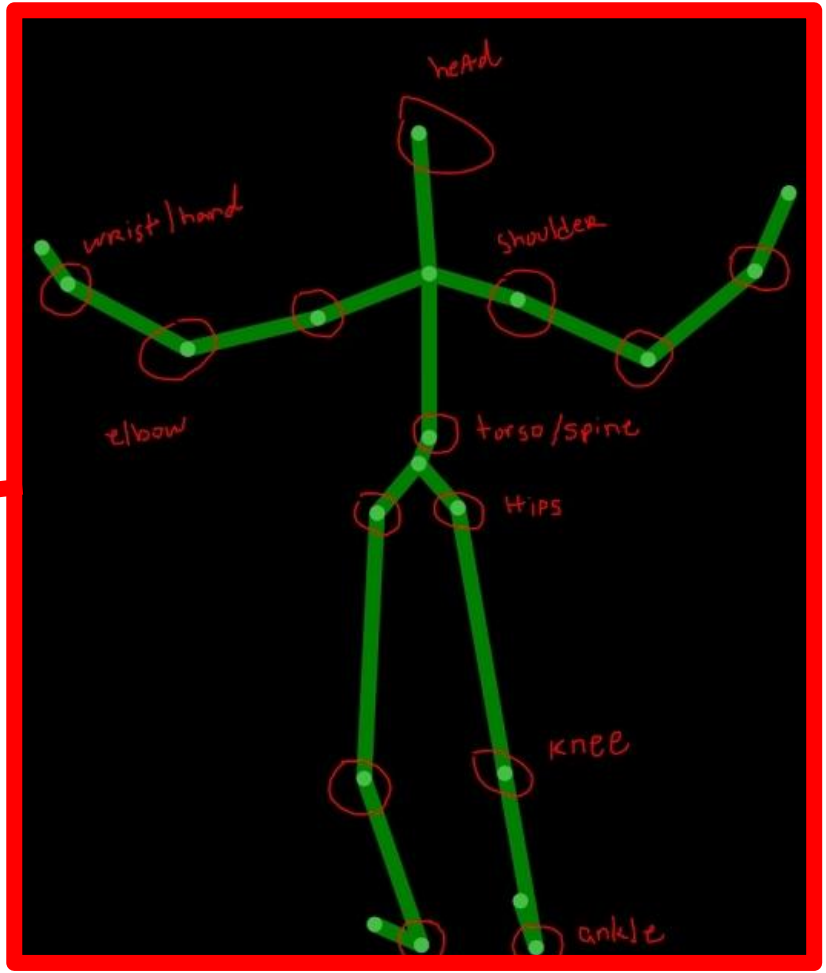
What's Done?

Integral Feature		Base Level of Functionality	Does it need to be expanded?	Could it be expanded?
Skeletal:	arm movement	complete	no	no
	leg movement	complete	no	no
	torso movement	complete	no	no
	head movement	complete	no	yes
Facial:	eyebrow movement	in progress	yes	yes
	mouth movement	in progress	yes	yes
Minecraft:	avatar art	complete	no	yes
	world art	complete	no	yes

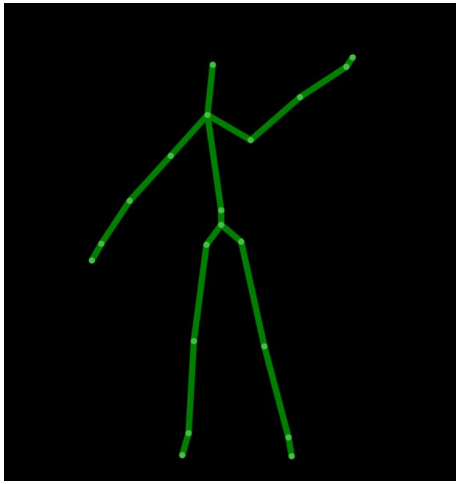
Kinect → Pd-L2Ork → Minecraft



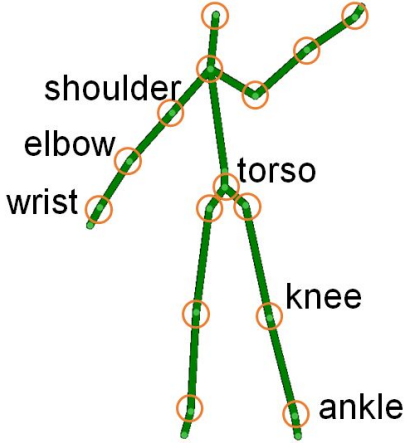
Skeletal Recognition



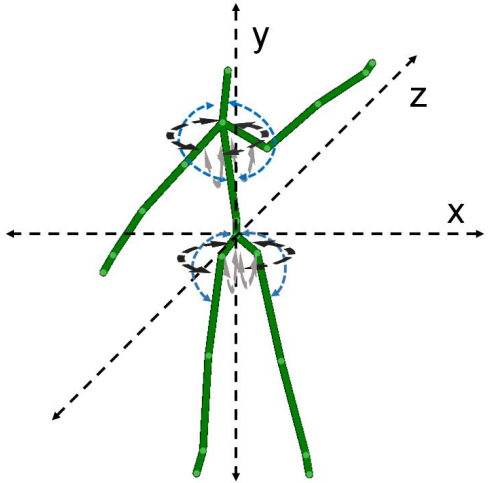
Translating Kinect data -> OPERAcraft movement



Kinect Skeletal Tracking



Infer points



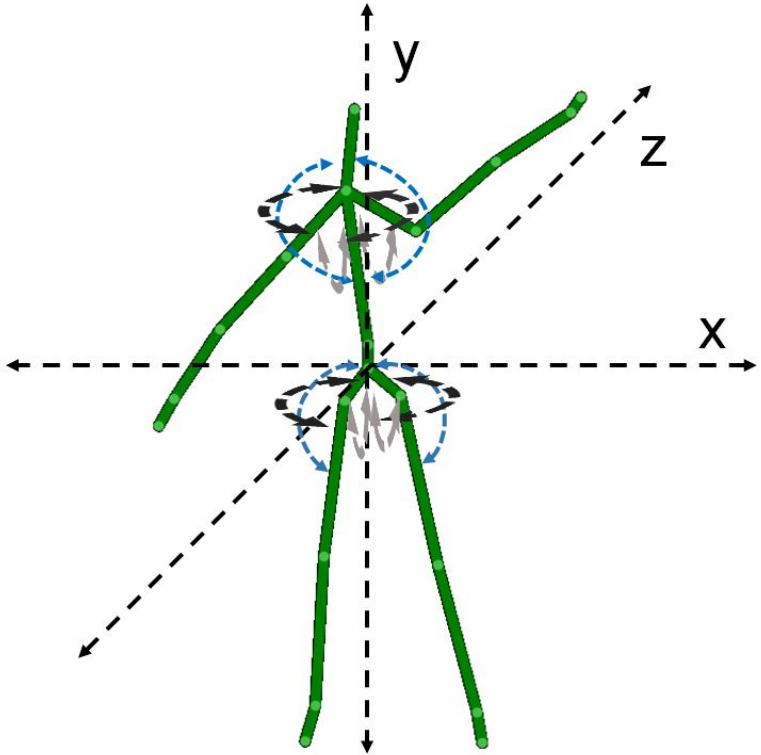
Calculate angles of rotation

UDP Packet:

```
@karm
ICAT01 24 36
…
```

Send **angles** to OPERAcraft

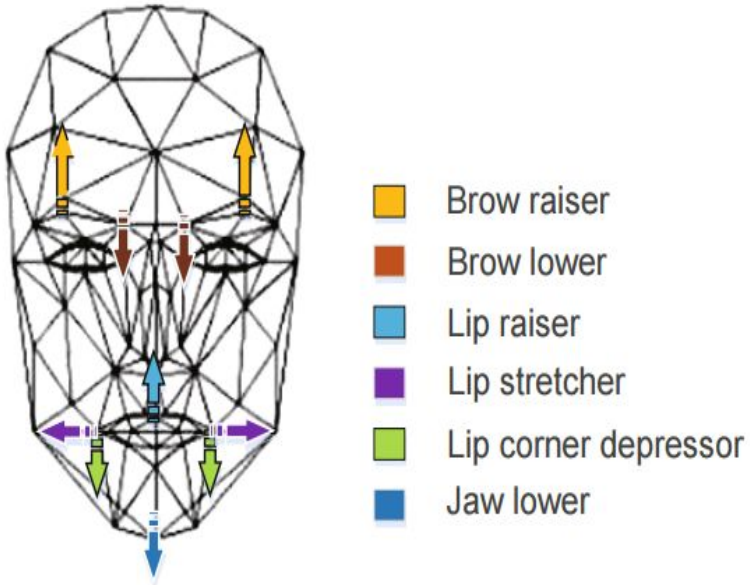
Angle Calculations



X Rotation
Y Rotation
Z Rotation

- 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



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).

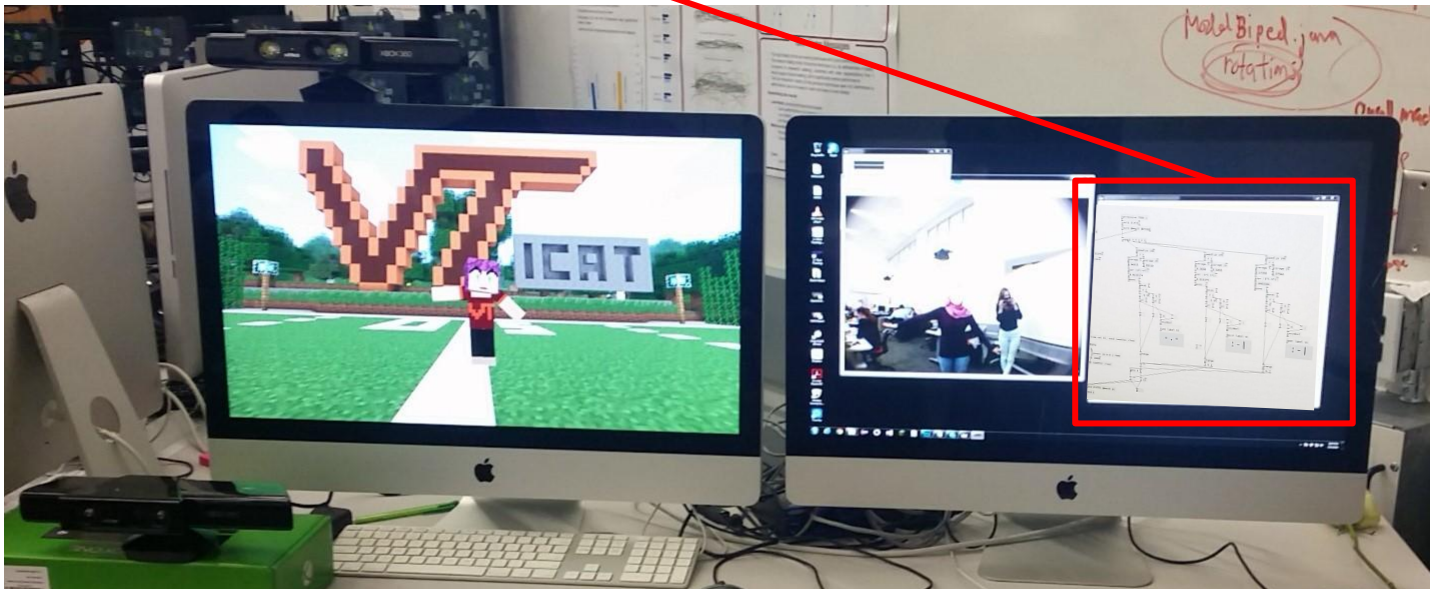
EYEBROW

MOUTH

0, 0	0, 1	0, 2	0, 3
1, 0	1, 1	1, 2	1, 3
2, 0	2, 1	2, 2	2, 3
3, 0	3, 1	3, 2	3, 3
4, 0	4, 1	4, 2	4, 3

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

Kinect → Pd-L2Ork → Minecraft

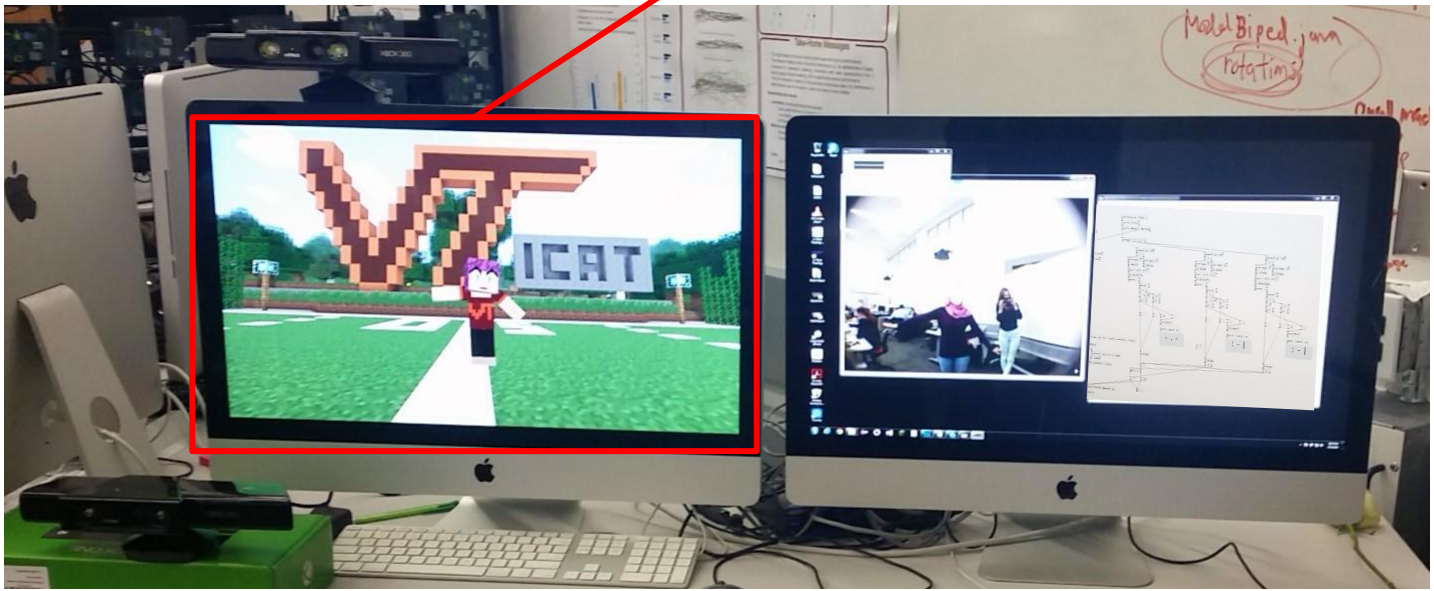


UDP Packet Constraints				
Delimiter(s)	Line Ending(s)	Avatar Movement Keywords	Movement Position Value(s)	Movement Position Types
whitespace	;\n	eyes mouth head shoulder arm leg torso	{0-3} {0-4} {(-90)-(90)} {(-180)-(180)} {(y)} {(-180)-(180)}{(-180)-(180)} {(-90)-(90)} {(-90)-(90)} {(x)} {(y)} {(-180)-(180)}	integer integer float (xrot, yrot) float (y) float (xrot, zrot) float (xrot, zrot) float(x, y, yrot)

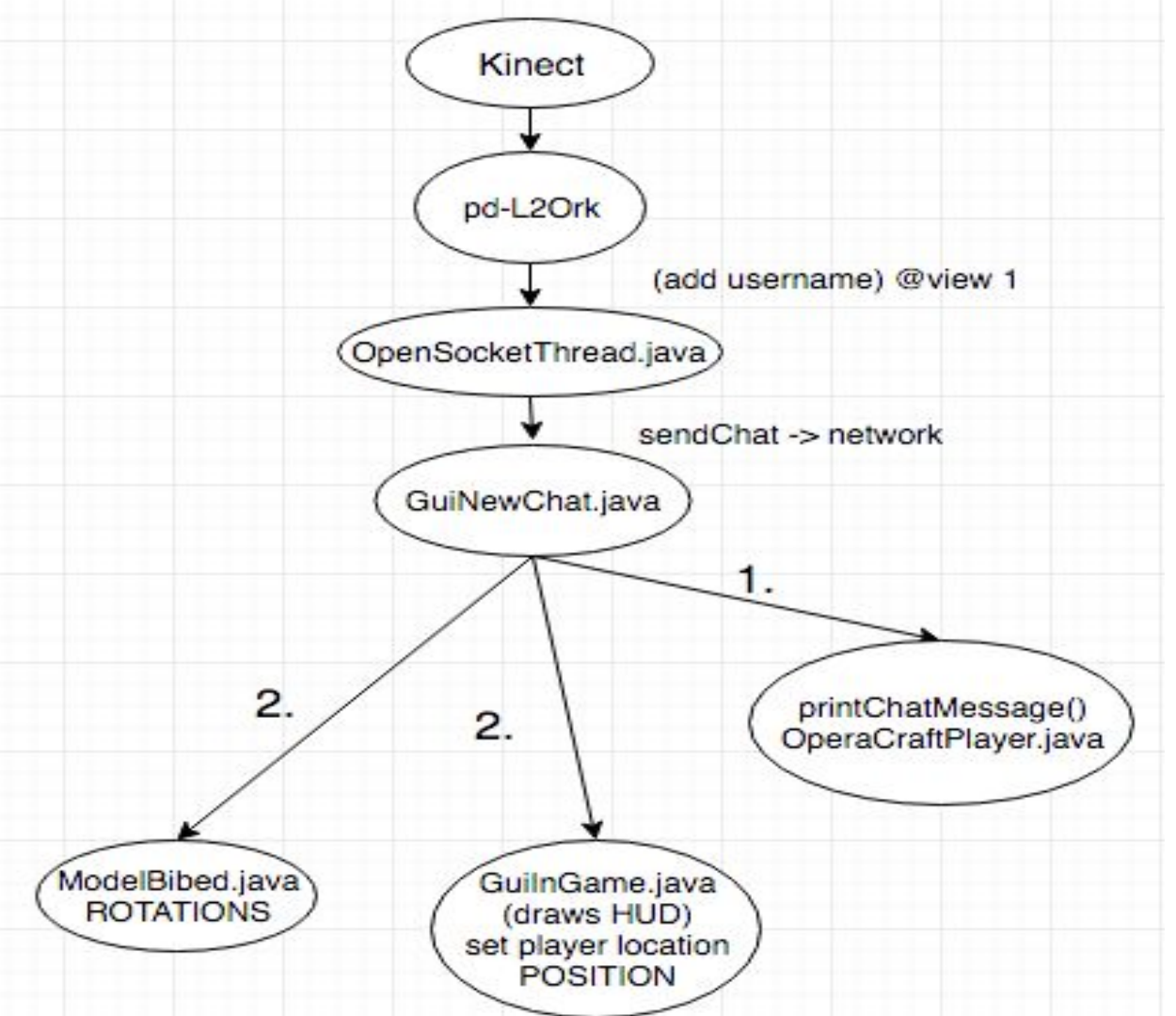
Pd-L2Ork: Averages/Normalization

- 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



Minecraft Art: Facial Expression Helmets

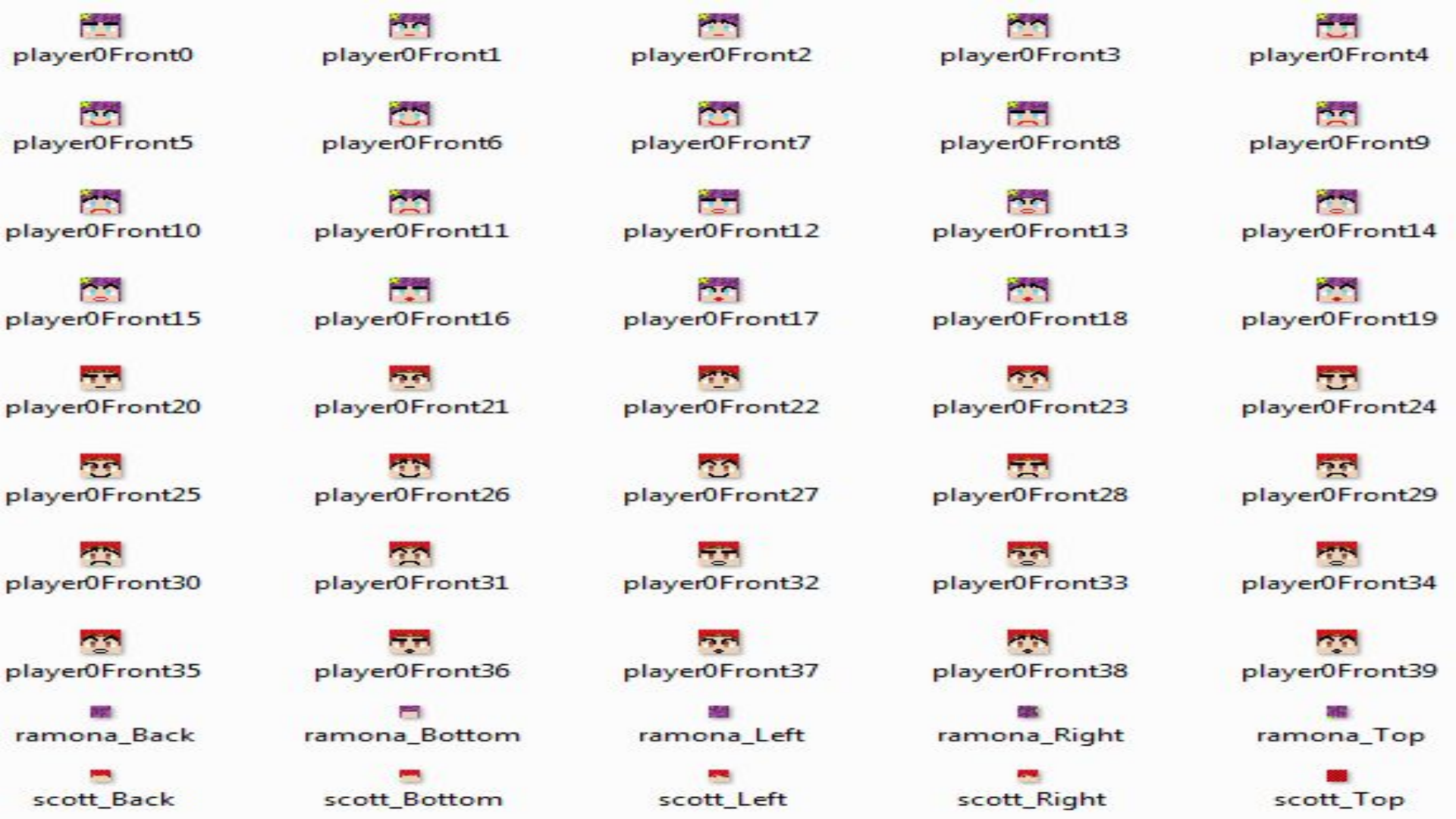
- Microsoft Paint
- 16x16 pixels
- Female and male

Sad
Anxious brows
+
frown

Surprised
Surprised brows
+
open lips

Angry
Furrowed brows
+
frown

Kiss Face
Neutral brows
+
puckered lips

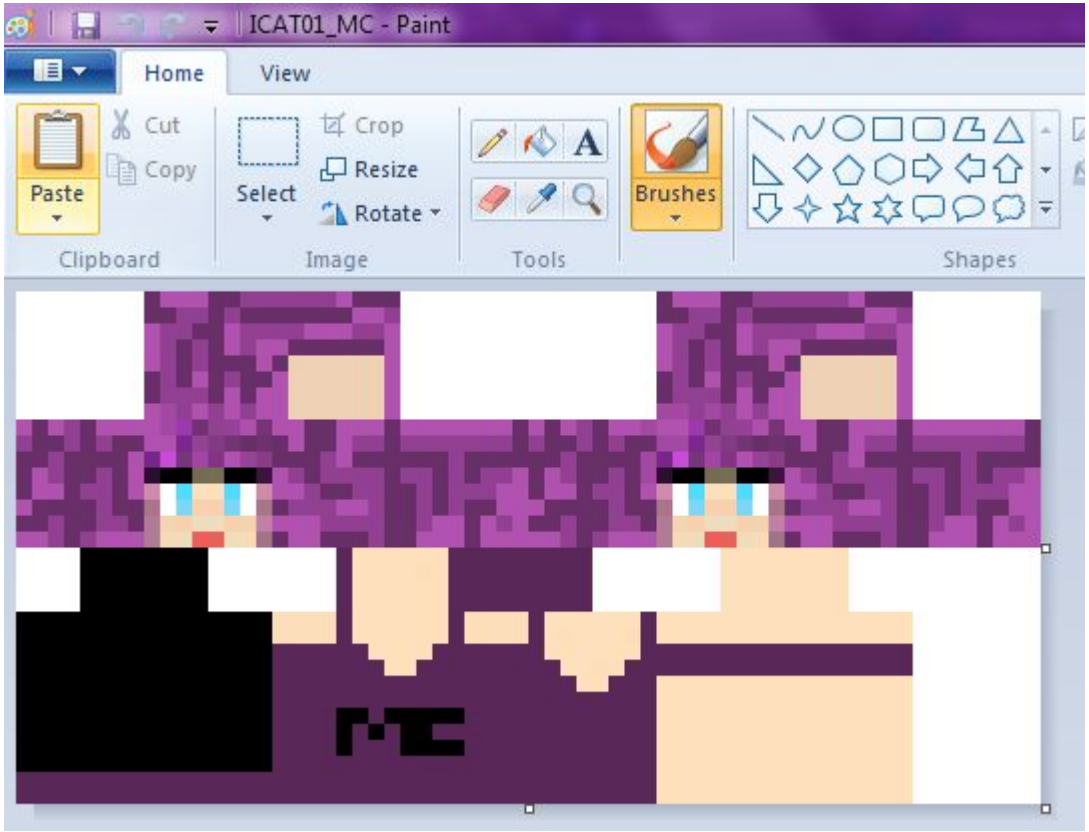

player0Front0
player0Front1
player0Front2
player0Front3
player0Front4
player0Front5
player0Front6
player0Front7
player0Front8
player0Front9
player0Front10
player0Front11
player0Front12
player0Front13
player0Front14
player0Front15
player0Front16
player0Front17
player0Front18
player0Front19
player0Front20
player0Front21
player0Front22
player0Front23
player0Front24
player0Front25
player0Front26
player0Front27
player0Front28
player0Front29
player0Front30
player0Front31
player0Front32
player0Front33
player0Front34
player0Front35
player0Front36
player0Front37
player0Front38
player0Front39
ramona_Back
ramona_Bottom
ramona_Left
ramona_Right
ramona_Top
scott_Back
scott_Bottom
scott_Left
scott_Right
scott_Top

Minecraft Art: Avatar Skin

- Microsoft Paint
- 32x64 pixels
- Female and male

100% zoom

800% 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