Skeletal Input for User Interaction in X3D
Manuel Olbrich - Tobias Franke - Jens Keil - Sven Hertling
Fraunhofer IGD
What we want

- Naturally interacting with VR applications
- Intuitive interaction for unsupervised installations
- Vandalism proof and hygienic
What we did before

2011

- UserSensor Node
- Getting skeleton data into X3D

```
UserSensor : X3DDirectSensorNode {
  ...
  SFInt32 [in,out] userID
  SFImage [out] userMask
  MFVec3f [out] jointPositions
  MFRotation [out] jointOrientations
  SFString [out] gesture
}
```
What we did before

2012

- Improved Skeletal Gesture Recognition
- JSON container as a unified interface
- Guideline to extend the system by writing own recognizers in JavaScript

```javascript
UserSensor : X3DDirectSensorNode {
    ...
    SFString   [out]    gestureJSON
}
```

```json
{
    Gesture:  Circle
    Info {
        Times: 1.3
        Radius: 4.2
        Center: [2.2, 1.0, 3.9]
    }
}
```
Data Source
Frameworks and Devices

- Microsoft Kinect SDK
  - Microsoft Kinect

- OpenNI
  - Microsoft Kinect
  - Asus Xtion Pro
**Scenarios**

Active Installations in daily use

- Virtually visit the reconstructed home of the Brothers Grimm
  - Select rooms
  - Look around
- Explaining Molecular Modelling to exhibition visitors.
  - Examine proteins
  - Position agent into protein pocket
  - Switch through different views
UserSensor : X3DDirectSensorNode {
...
SFInt32    [in,out] userID
SFImage    [out]    userMask
MFVec3f    [out]    jointPositions
MFRotation [out]    jointOrientations
SFString   [out]    gesture
SFString   [out]    gestureJSON
}

- Skeleton joint positions in 3D
Data Flow
Steps in the Pipeline

- Data preparation
  - Sensor orientation,
- User Selection
  - Switching
  - Reacting to lost user
- User Interaction
  - Scale
  - Filtering
  - Interpreting
Implementation in X3D

- All elements in the pipeline are implemented as Script Nodes.
- Data preparation and user selection keep the jointPosition data structure.
- Interaction nodes use jointPosition format.
User Selection

- **Lift-Hands**
  - Tracked user skeletons are displayed on the screen
  - First user who raises his arms takes control

- **Step-On-Map**
  - First user who steps on the marked interaction area takes control
  - Takeover if user is out of the area and new user is inside
  - Application reset if no user is in the area for 10 sec
Interaction Types

- Examine
- Object Placement
- Look-Around
- Map Input
- 2D GUI Interaction
Examine

Rotate an object in front of your camera

- User can
  - Rotate object around all axis
  - Zoom in and out
Examine

(Video)
Object Placement

Position an object with your hands in 3D

- Object is bound to the users hands
- User can rotate the object around all axis
- User can place the object in 3D
- User can continuously rotate the object at the limits of normal rotation
Object Placement

(Video)
Look Around
Camera rotation around a fixed point

- Camera panning with hand movements
- Trying to understand the user
  - Left and right handed users
  - Hand changes
  - Two hands
  - „Unused“ hand
- Allowing the user to relax
Look Around
(Video)
Map Input

Floor printed map as an user input

- Feet position on map select the room
- Hints to the user
  - Map display on leave
  - Feet marker
  - Position lights up on contact
Map Input

(Video)
GUI Interaction

Skeleton interaction with UI elements

- Interaction with TouchSensors
- Hands act like cursors
  - Displayed skeleton helps the user to understand the handling
- UserBody and Timers for Interaction
[Video] GUI Interaction
Thank you!
Questions?
[Short description]

Video