PROXIMITY TOOLKIT

Rob Diaz-Marino, Interactions Lab
rob.diazmarino@gmail.com
OVERVIEW

- Premise of Proximity Toolkit
- The Home Space
  - Intro to Vicon System
  - Intro to Proximity Server
- Proximity Data Model
  - Presence Properties
  - Relation Properties
- Using the Toolkit
  - Controls
- Help & Resources
- Demo
PREMISE OF PROXIMITY TOOLKIT
Any potential proximity input hardware

- Vicon
- Motion Sensor
- ...

ProximityToolkit

Your Application
UNIFIED DATA MODEL

The Space

Things that Change (Subjects)

Fixed Features (Volumes)

Relations
Accessibility

- Potentially distributed applications

Computer A - Input

Computer B - Display

Mobile Device - Display
USING THE HOME SPACE
GENERAL MACHINE USE

- **SmartBoard**
  - Switch off after use

- **2\textsuperscript{nd} Monitor**
  - Input: DVI-D 2

- **Computer**
  - Always on
  - Leave logged in
  - Ignore updates
INTRODUCTION TO VICONS

Ultranet → Computer → Vicon Nexus → Proximity Server → Your Program

Camera 1 → Camera 2 → … → Camera 8
VICON NEXUS APPLICATION
SUBJECTS

Pencil  WhiteHat  BlackHat
LOADING SUBJECTS

- Subjects Tab
- Load Subject button
- Check/Uncheck

DO NOT TOUCH ANY OTHER SETTINGS!
TRACKED SUBJECTS
DISPLAYS
VOLUMES
EXPLORING THE SPACE

Modules
- Vicon Input Module

Presences
- Pencil
  - IsVisible
  - Confidence
  - LastUpdate
  - Location
    - IsVisible
    - Confidence
    - LastUpdate
    - Location
      - X: 1.52149  Y: 1.12833  Z: -113.06

Data Value
- homespace, Pencil, Location, Location
- Vector3
  - X: 1.52149  Y: 1.12833  Z: -113.06
VISUALIZING RELATIONS
HOMESPACE SUMMARY

- **Starup Procedure**
  - Turn on Machine + Display(s)
  - Turn on Ultranet (switch on back)
  - Run Vicon Nexus
  - Load Subjects
  - Run ProximityServer
  - Connect ProximityServer to Nexus

- **Shutdown Procedure**
  - Turn off Ultranet
  - Turn off Display(s)
PROXEMIC DATA
Identity Base
- Decorator Data Model
  - Different Inputs = Different Data
- Optional Groups of Data
  - Location
  - Motion
  - Orientation
  - Direction
  - Rotation
- Certainty
LOCATION

(300, 200, -300)

(300, 0, 300)
Motion

Velocity

Acceleration

(300, 200, 300)  (300, 200, 0)  (300, 200, -500)

(0, 0, -300)  (0, 0, -500)

(0, 0, -200)
Incline

Up

Front

30°

Ground (XZ Axis)
Azimuth

Front

Ground (XZ Axis)

180°
ORIENTATION

Roll

Up

Perp Up

-30°

Ground (XZ Axis)

Front
Simplification of Orientation
  - Incline
  - Azimuth
  - NO Roll

Anything Orientable is Directable

Used for Pointers
Client-Side Calculation on Request

Relation Types

- Location
- Motion
- Orientation
- Direction
- Rotation
- Pointing
- Collision
Distance (X/Y/Z)
Horizontal Distance
Vertical Distance
Incline A to B
Incline B to A
A Moving Toward B
A Moving From B
Etc.
Is A Moving
Is B Moving
X/Y/Z Velocity Agrees
X/Y/Z Acceleration Agrees
Velocity Difference
Acceleration Difference
Incline Agrees
Incline Difference
Azimuth Agrees
Azimuth Difference
Roll Agrees
Roll Difference
A Towards/From B
B Towards/From A
A Tangential to B
B Tangential to A
Parallel
Horizontally Perpendicular
*Open to Suggestions
Points At
Points Toward
Distance
Is Touching
Intersection Location
Intersection Direction
*Screen Coordinates
Collision Relation

Collision
Containment
Distance
Intersection
Nearest Point
USING THE TOOLKIT
Adding References

- Proximity Toolkit Installation
  - C:/Program Files/ProximityToolkit

- Windows Forms
  - ProximityToolkit.dll
  - ProximityToolkit.WinForms.dll

- WPF
  - ProximityToolkit.dll
  - ProximityToolkit.WPF.dll
  - System.Drawing
Add a Tab
- Right click on Toolbox
- Choose “Add Tab”
- Give Tab a name like “ProximityToolkit”

Add Items
- Right click on Toolbox in new tab
- Choose “Choose Items…”
- Browse for .NET components & select libraries
- Ensure the following are checked
  - PresenceManager
  - PresenceControl
  - RelationControl
Properties:
- IP
- Port
- AutoDismissEmbodyDialog
- MaintainEmbodiment

Events:
- OnConnectionDropped(Asynch)
- OnStarted(Asynch)
- OnStopped(Asynch)
- OnReadyChanged(Asynch)
- OnUpdating(Asynch)
- OnDataUpdated(Asynch)
- OnUpdated(Asynch)

```csharp
public Form3()
{
    InitializeComponent();
    presenceManager1.Start();
}

private void Form3_Load(object sender, EventArgs e)
{
    if (!presenceManager1.IsStarted)
    {
        Application.Exit();
        return;
    }
}
Set PresenceName (match ProximityServer)

Set PresenceType
  - (Presence/Display/Volume/Device)

Set Monitor
  - None, Location, Direction, Orientation, Rotation, Motion, All

Set IsOptional / CustomImage

Properties:
- PresenceName
- PresenceType
- Monitor
- IsOptional
- CustomImage

Events:
- OnLocationUpdated
- OnDirectionUpdated
- OnOrientationUpdated
- OnRotationUpdated
- OnMotionUpdated
- OnUpdated
Properties:
- A
- B
- Monitor

Events:
- OnLocationUpdated
- OnDirectionUpdated
- OnOrientationUpdated
- OnRotationUpdated
- OnMotionUpdated
- OnCollisionUpdated
- OnPointingUpdated
- OnUpdated

- Set A to a Presence Control
- Set B to a Presence Control
- Set Monitor
  - None, Location, Direction, Orientation, Rotation, Motion, Collision, Pointing, All
Embodiment Dialog

- Appears on Program Startup
- Availability Verifiction
  - Connection
  - Presences
PresenceManager not a container
- Can organize with Stack Panel, set to Hidden

PresenceControl
- Monitors are bool properties
  - MonitorLocation, MonitorDirection, etc.

RelationControl
- Monitors are bool properties
  - MonitorPointing, MonitorCollision, etc.
- Must specify NameA and NameB
GUI-INDEPENDENT SUPPORT

- ProximityClientConnection
- PresenceBase for Presences
  - DataUpdating/DataUpdated events
  - GetRelationInterface for types ILocatable, IMovable, IOrientable, IDirectable, etc.
- RelationPair for Relations
  - Same Events as RelationControl, all Asynch
HELP & RESOURCES
HELP AND RESOURCES

- Cookbook Page
  - http://grouplab.cpsc.ucalgary.ca/cookbook
  - Reference

- E-Mail
  - rob.diazmarino@gmail.com
THE END
(DEMO TIME?)