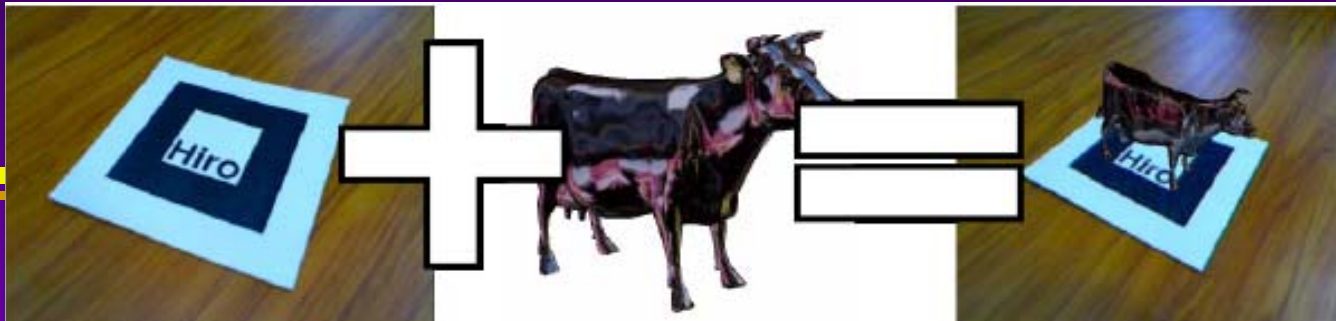
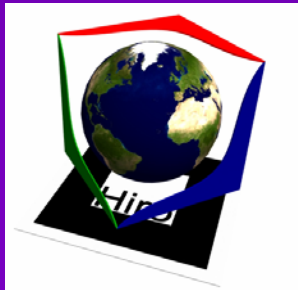


OSGART: A Pragmatic Approach to MR

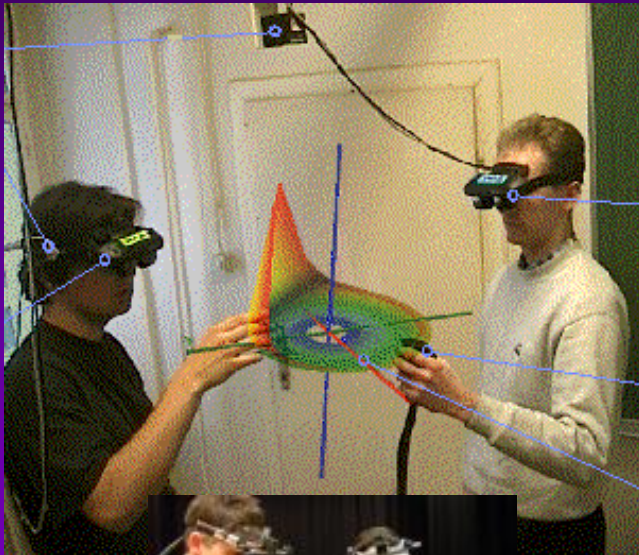


Julian Looser, Raphaël Grasset,
Hartmut Seichter, Mark Billinghurst



Motivations

- Augmented Reality Applications

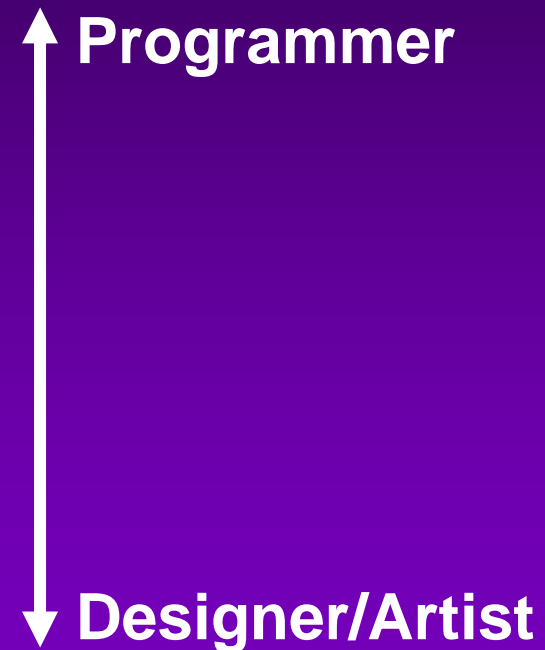


Need Tools, Utilities, SDK

Motivations

- Design an AR Application ([Ponder05]):

- Start from 'scratch'
- Extend VR/Game Solution
- AR Libraries
- AR ToolKit
- AR Framework
- AR Authoring Tools



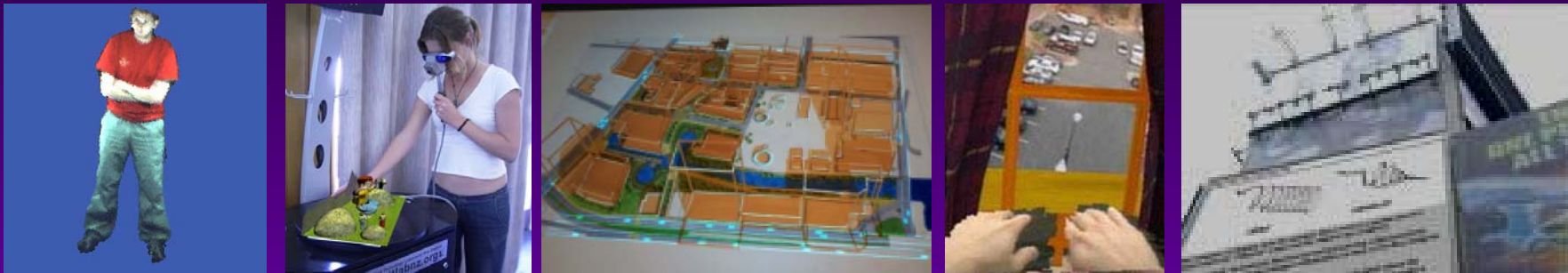
Existing Solutions

- **Libraries/Toolkit:** MR-Platform, MXToolkit, ARTToolkit, MXRToolkit, ImageTclAR, ARTag
- **Framework:** COTERIE, Studierstube, DWARF, AMIRE, ARTHUR, VHD++, Shared-Reality, Metaio
- **Authoring Tools:** APRIL, CMIL++, DART

Do we need a new tool ?

Emerging Trends

- More spaces, more mixing (along Milgram's Continuum)



- More tangible, more electronic smart devices



- More alternative collaborative system



Objective 1: Supporting New Issues

- No Real Dedicated and Flexible Solution
- Intrinsically supporting:

- TANGIBILITY



- TRANSITIONALITY



- COLLABORATION AWARENESS



Objective 2: Supporting Different Capabilities

- From Programmer to Designer
- From Learning to Prototyping to Developing
- → Be pragmatic: Multi-platform, Simple, Easy to use, All-in-one

ARToolKit?

ARToolkit

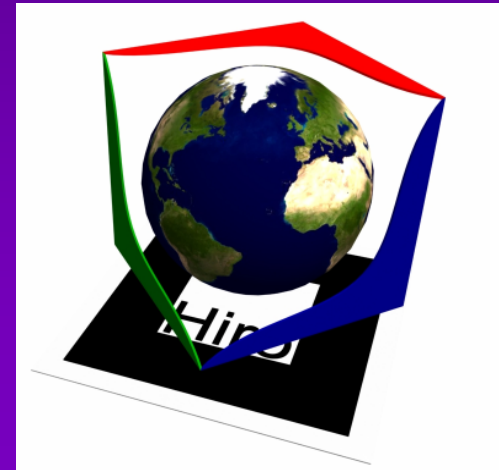
- Most used AR Library
- ~1000 download/month
- 100000 google references
- Multiplatform, all-in-one, small package, wrapping other libraries, **simple API, easy learning curve**
- **BUT limited rendering, lack robustness, alternative solutions (ARTPlus), etc.**

→ Enhance it and Replicate the approach

OSGART

= OpenSceneGraph + ARToolkit

- ToolKit supporting easy development of AR/AV (MR), **MediatedR** Applications
- Multi-level programming interfaces
 - C++, Lua/Python, Built-in Authoring Tools
- Wrapping than redeveloping
- More features than OSGAR
- Base Layer for Objective 1



Features

- Functional Services
 - High Level Video Layer
 - Video Objects
 - Geometric/Spatial Registration
 - Photometric/Visual Registration
- Developer Services
 - C++/Scripting (RAD)/Simple Authoring Tools
 - Documentation/Samples/Tutorials
 - Utilities
 - Free for Academic, License for Industrial



OpenSceneGraph



- High Quality Rendering
- Import/Export Max/Maya + Utilities
- Community Plugins: OSGAL, ReplicantBody, osgVortex, etc → Wrapping

Functionalities Overview..

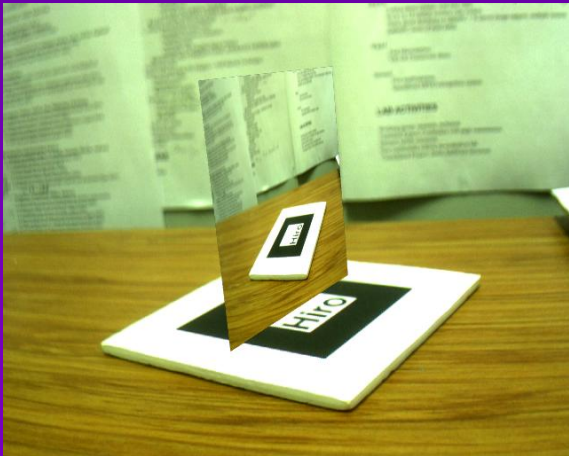
Video Input

- Multiple Video Support:
 - Video Input: WebCam (USB, Firewire), PtGrey, DCAM, etc.
 - Video Files: AVI, Quicktime, etc.
 - Video Streaming: Gstream, RTSP
- Integrating others video libraries: ARVideo, VideoWrapper, VideoCapture, CVCam, etc.
 - → Usual tool of developers



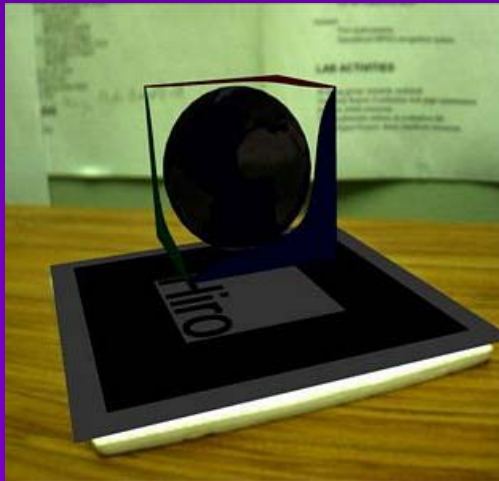
Video Objects

- Video Textures
- Video Shader
- Video Objects: VideoPlane, VideoBillboard, VideoLayers, etc.



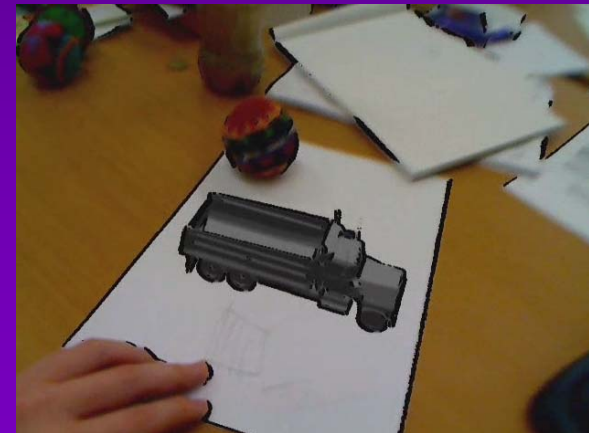
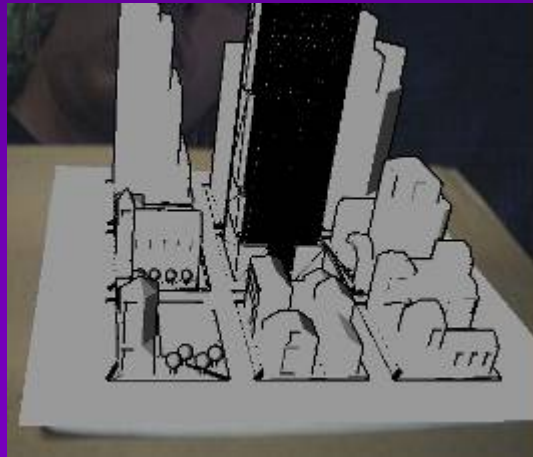
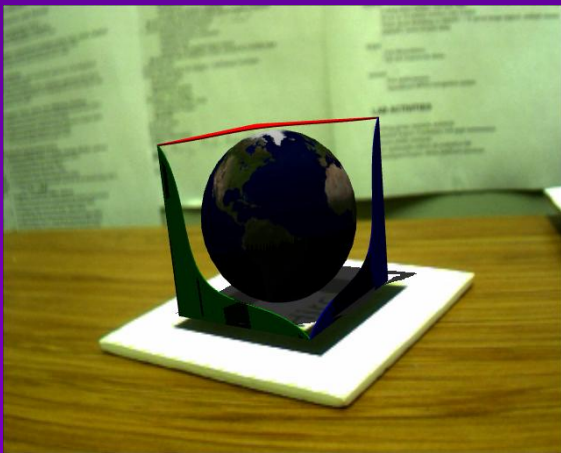
AR Spatial Registration

- Integrated on the SceneGraph (*ARTTransform*)
- Multiple Markers, Multi-Marker support
- Tracker Plugins: ARToolKit, ARToolKit4, ARToolKit4+NFT, etc.



AR Visual Registration

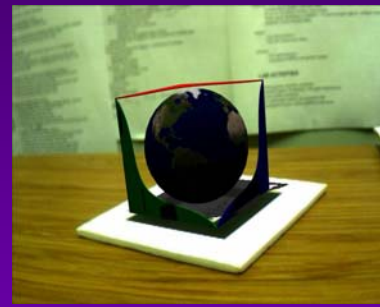
- Photorealistic: Occlusion, Shadow
- Non Photorealistic: on video, on the content, on both (stylized AR)



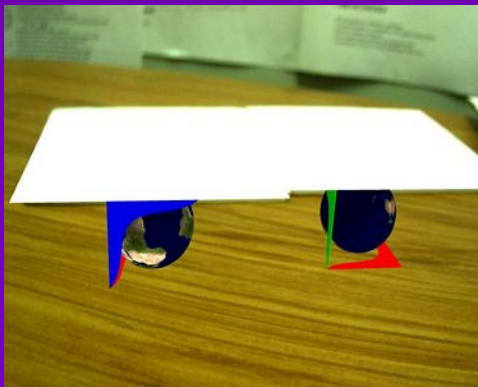
Developer Overview..

ARToolKit Examples

- simpleVideo, simpleTest, multi, simpleShadow



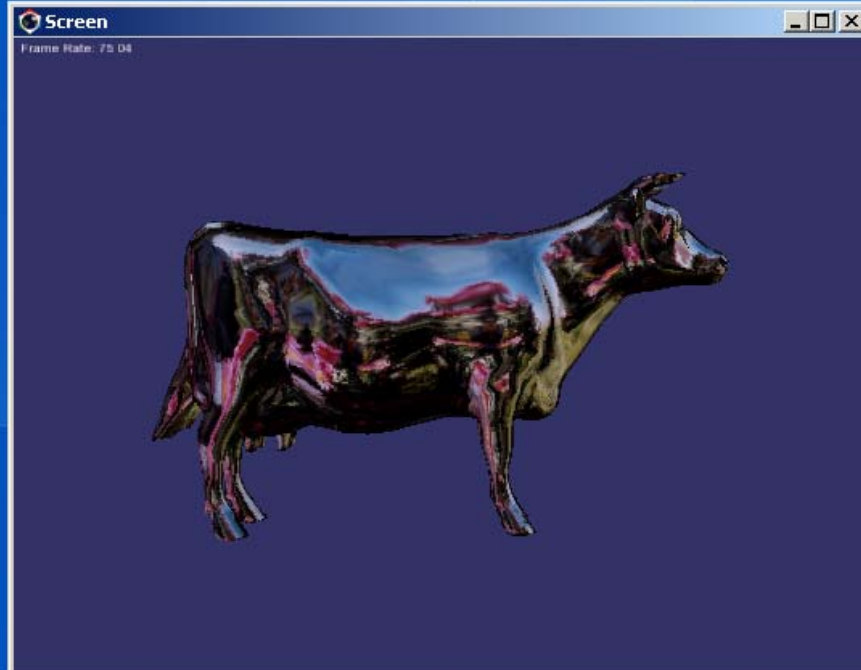
- Collide, PaddleInteraction



Scripting Coverage

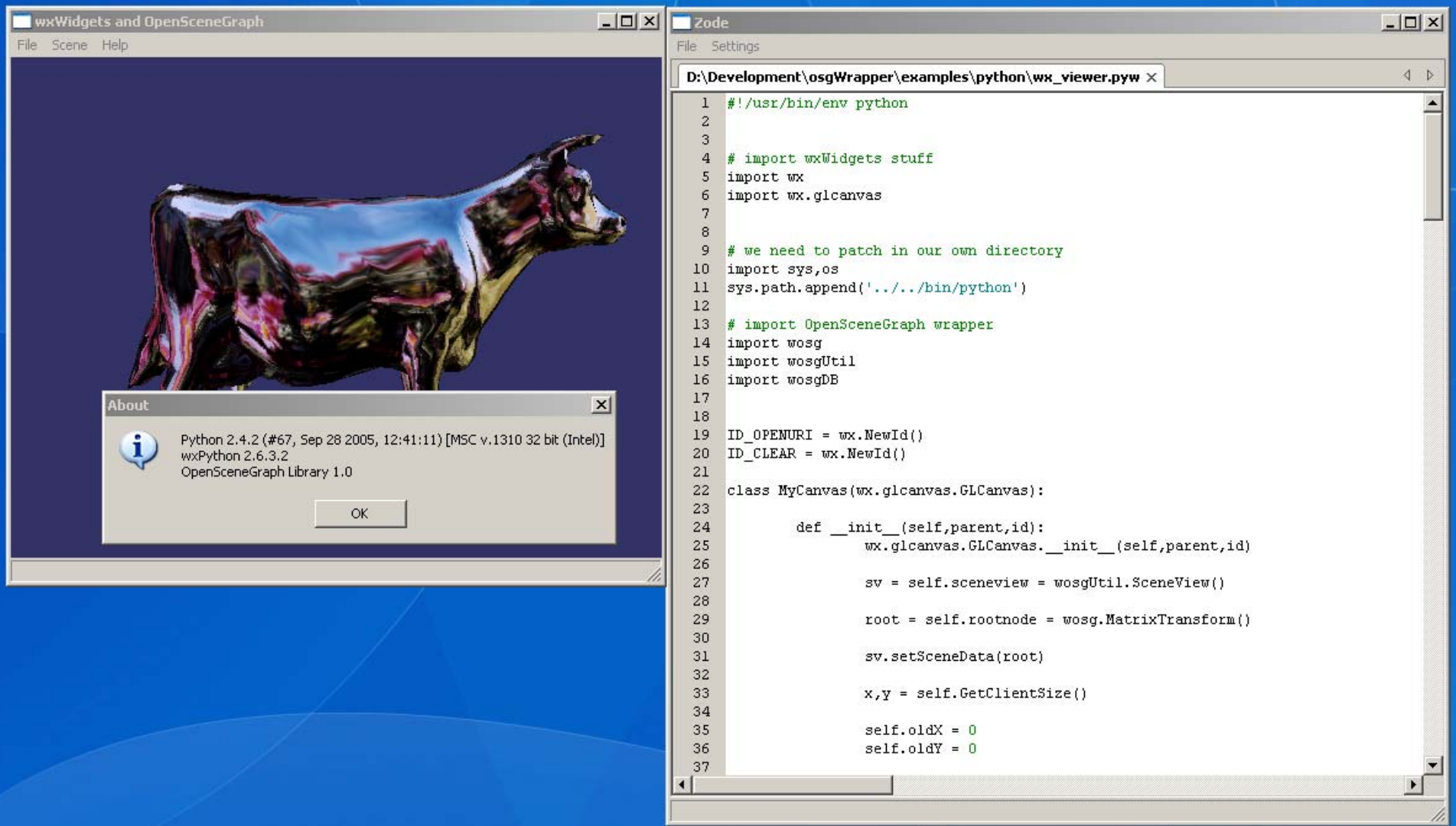
- Hand tuned SWIG wrapper scripts
- Wrapper interfaces use unmodified OpenSceneGraph header file
- Platforms: Windows, Linux, Unix (incl. Mac OS X)
- Simple integration into wxPython, Ruby GTK+ etc.

Ruby Version



```
File Settings
D:\Development\osgWrapper\examples\ruby\viewer.rb x
1  #!/usr/bin/ruby
2
3  #
4  # Demonstation for Ruby
5  #
6
7  # extend the load path for wosg
8  $LOAD_PATH << '../bin/ruby'
9
10
11 # include all necessary libs
12 require 'wosg'
13 require 'wosgDB'
14 require 'wosgProducer'
15
16 class Viewer
17
18     def initialize()
19
20
21         puts "Open a window ... "
22
23         @viewer = WosgProducer::Viewer.new
24
25         @viewer.setUpViewer(WosgProducer::Viewer::STANDARD_SETTINGS)
26
27         puts "Reading Data ... "
28
29         n = WosgDB::readNodeFile("cow.osg")
30
31         root = Wosg::Group.new()
32
33         root.addChild(n)
34
35         @viewer.setSceneData(root)
36
37         puts "Show the Window ... "
38
39         @viewer.realize()
40
```

wxPython / OSG / Windows



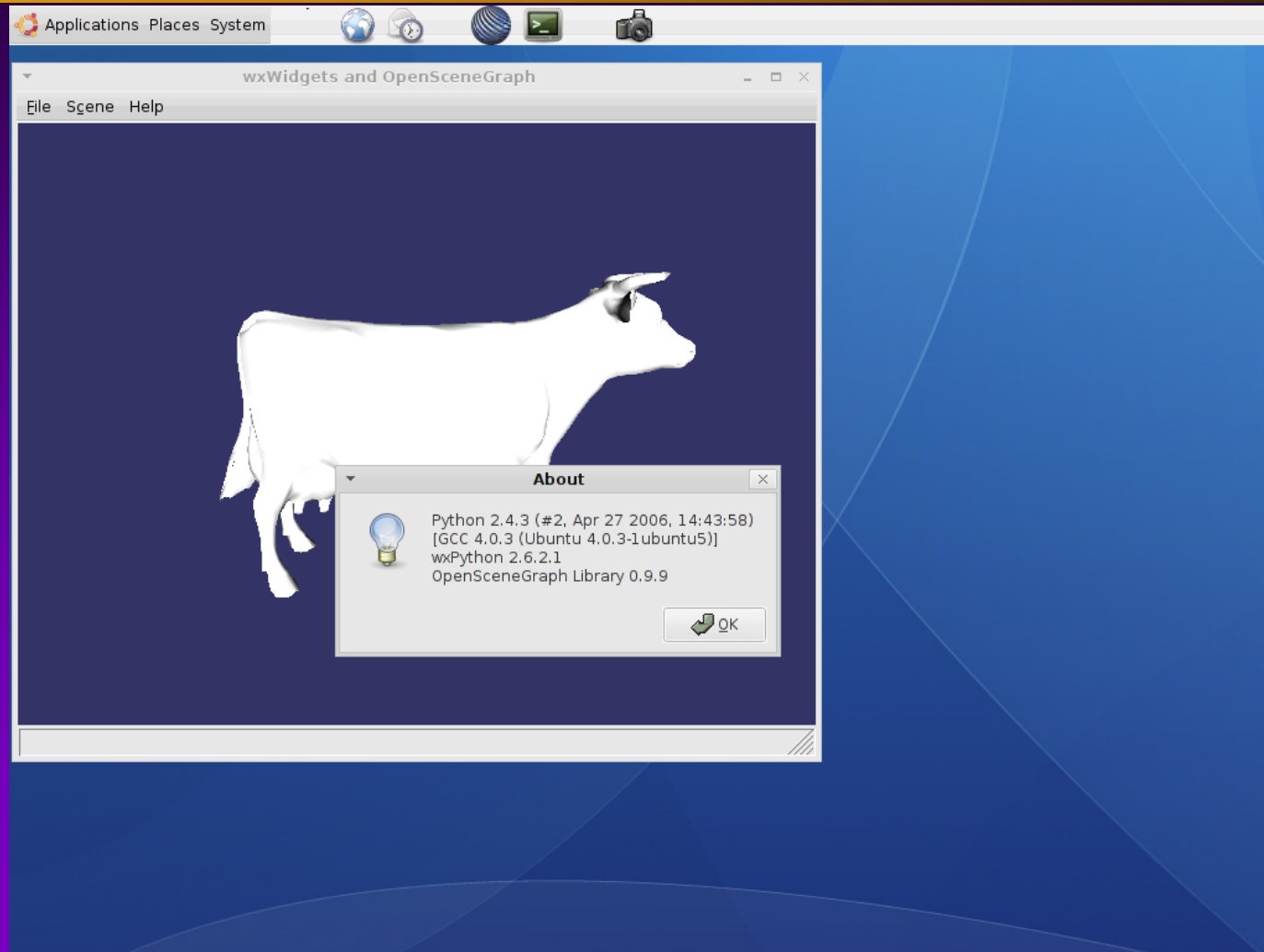
The screenshot displays a Windows desktop environment. On the left, a window titled "wxWidgets and OpenSceneGraph" is open, showing a 3D rendering of a metallic cow. An "About" dialog box is overlaid on this window, displaying the following information:

- Python 2.4.2 (#67, Sep 28 2005, 12:41:11) [MSC v.1310 32 bit (Intel)]
- wxPython 2.6.3.2
- OpenSceneGraph Library 1.0

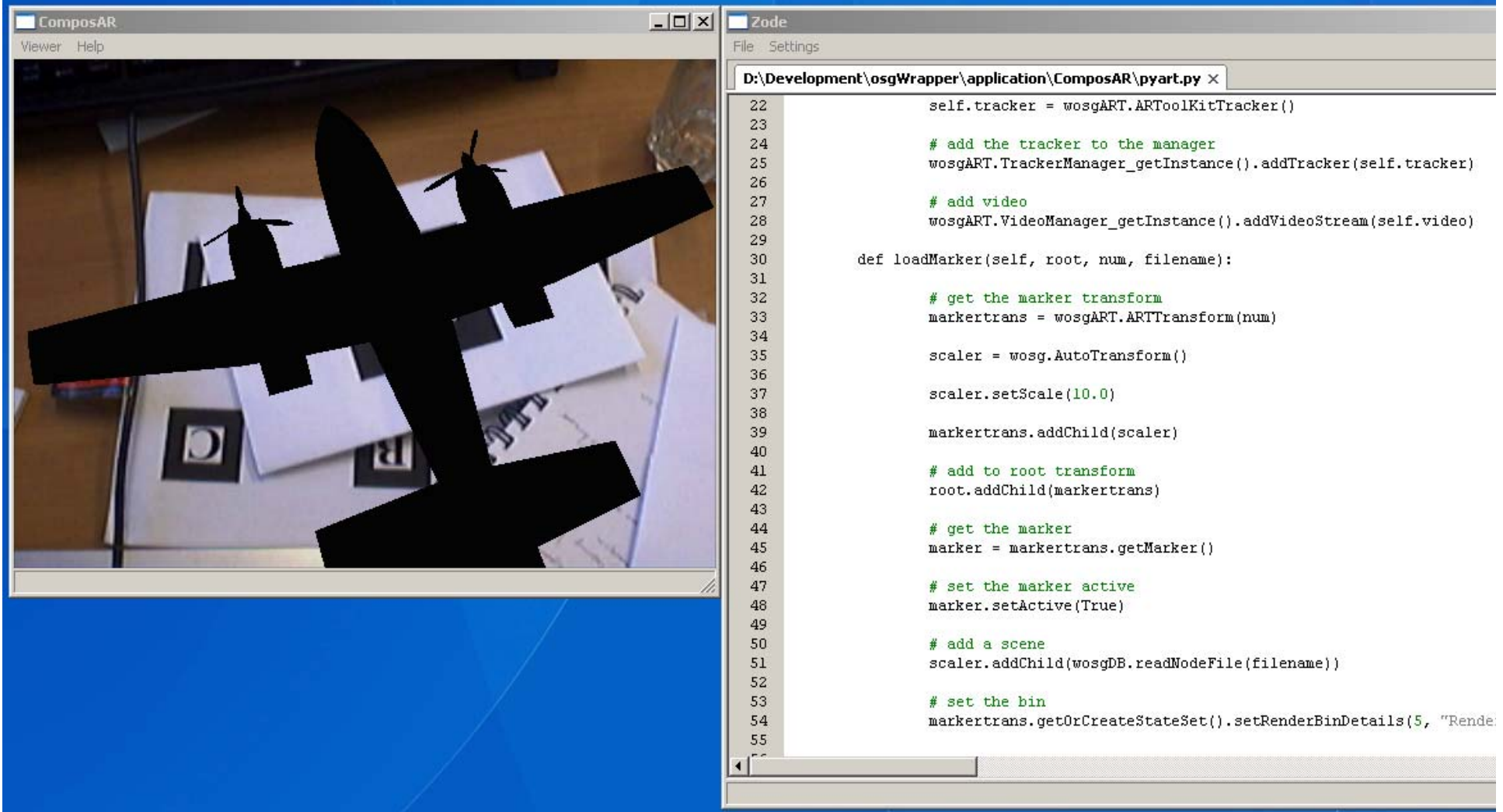
An "OK" button is visible at the bottom of the dialog. On the right, a "Zode" code editor window is open, displaying the source code for a Python viewer application. The code is as follows:

```
D:\Development\osgWrapper\examples\python\wx_viewer.pyw x
1  #!/usr/bin/env python
2
3
4  # import wxWidgets stuff
5  import wx
6  import wx.glcanvas
7
8
9  # we need to patch in our own directory
10 import sys,os
11 sys.path.append('../..bin/python')
12
13 # import OpenSceneGraph wrapper
14 import wosg
15 import wosgUtil
16 import wosgDB
17
18
19 ID_OPENURI = wx.NewId()
20 ID_CLEAR = wx.NewId()
21
22 class MyCanvas(wx.glcanvas.GLCanvas):
23
24     def __init__(self,parent,id):
25         wx.glcanvas.GLCanvas.__init__(self,parent,id)
26
27         sv = self.sceneview = wosgUtil.SceneView()
28
29         root = self.rootnode = wosg.MatrixTransform()
30
31         sv.setSceneData(root)
32
33         x,y = self.GetClientSize()
34
35         self.oldX = 0
36         self.oldY = 0
37
```

Same on Linux



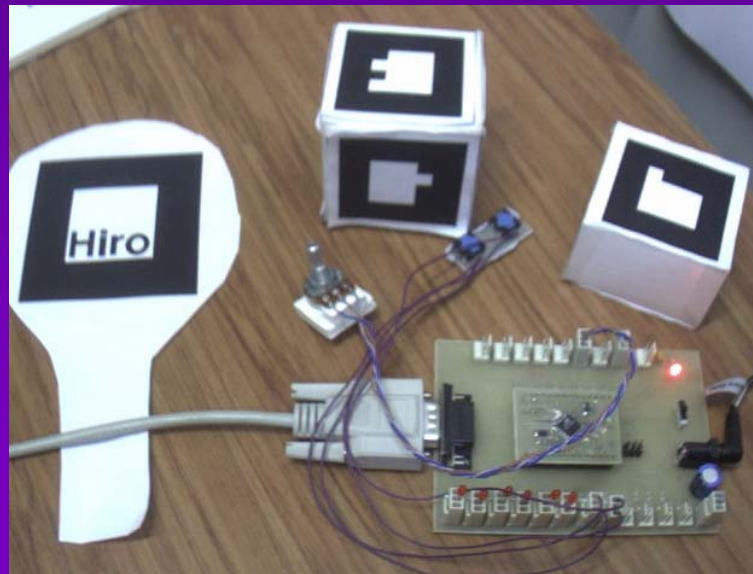
OSG / AR Toolkit / wxPython



And the Upper Level..

Manipulation: Tangible

- Spatial+Physical Input: Prototyping Interface
 - ARToolKit+homemade microcontroller
 - Large choices of sensor/actuators
 - (switch, pressure, LEDs, Motor)

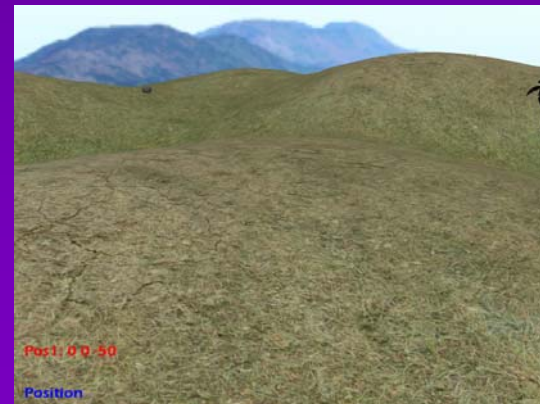


Navigation: Transitionality

- Transitional Framework:
 - Multispace support: VR, AR, AV
 - Multiscale, Multiviewpoint, Multirepresentation support
 - Navigation: Transitional Interface
- Transition Awareness



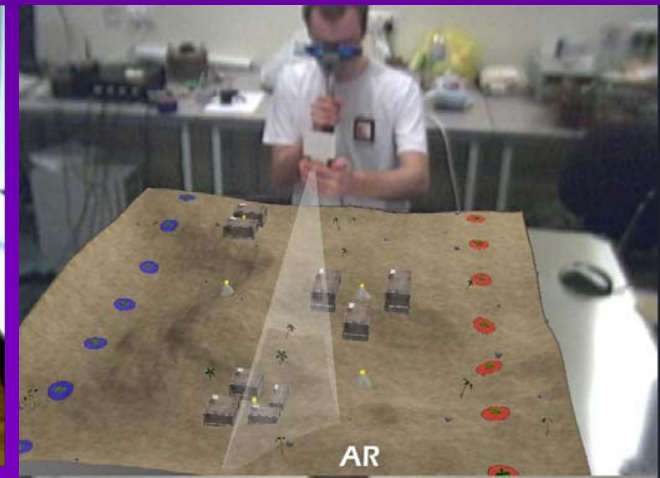
AR



VR

Collaboration Awareness

- Awareness: Head/Hand Position, Gaze Direction, Telepointers, etc.
- Dedicated Objects (VideoAvatar, GazeAwar..)
- Approach: ICE (Middleware)



Results

First Results

- Integration with OSG plugins



First Results

- Animalia (CONVERGE 05)

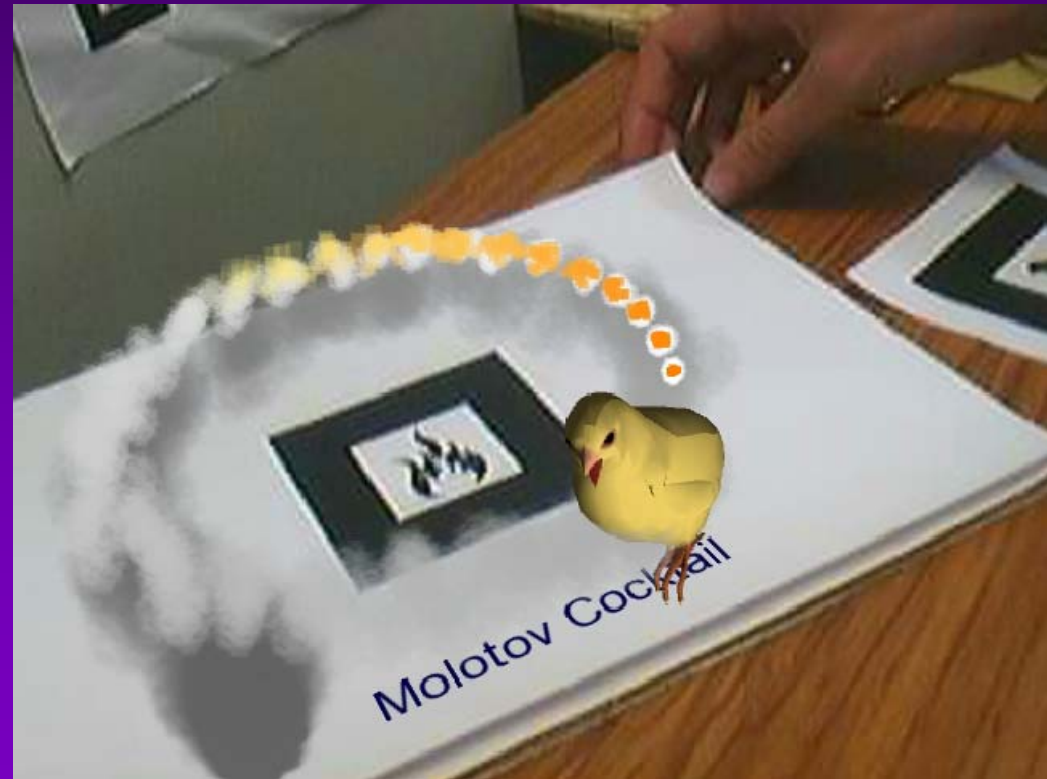






First Results

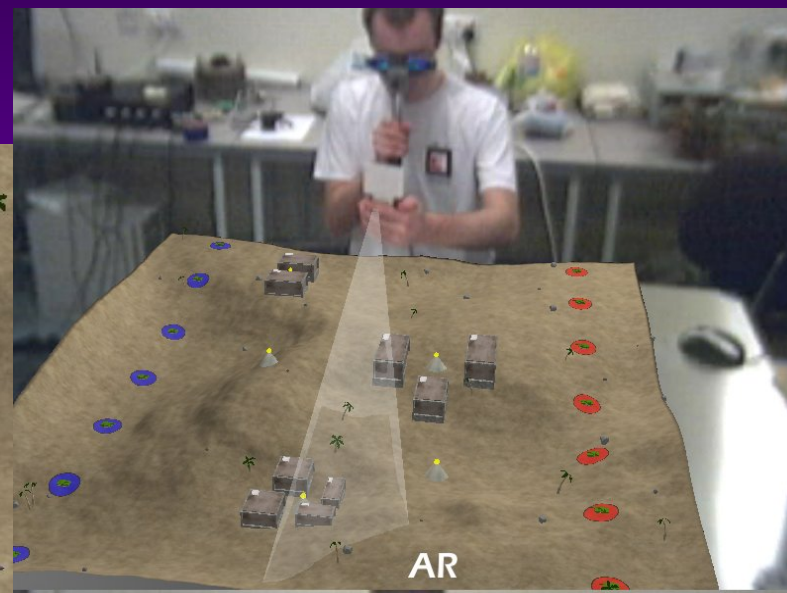
- Experimenta Vanishing Project (Australia)



MagicLens Research



Tank War



First Results

- Industrial Prototypes
 - NZ Army
 - Airways
 - W&M
- Commercial Project (MagicBook)



Contact, Information

- Website: <http://www.hitlabnz.org>
- Distribution, Availability:
 - Mark Billingham
 - Philip Lamb (Licensing)
- Development:
 - Julian Looser
 - Hartmut Seichter
 - Raphael Grasset
-
- Pre-Alpha Release: December 2006