



WiiR3D

User manual

2008 CSCI321 Project GA1

Jessica Lloyd

Adam Parkes

James Leskovar

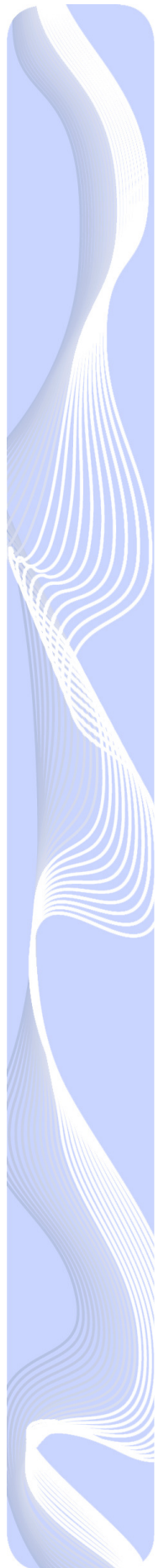
VinhBuu To

<http://zim.cs.uow.edu.au:50321/~cs321ga1>

<http://home.exetel.com.au/wiir3d>

Contents

1	Introduction	2
a	Thank you.....	2
b	Intended readership	2
c	About.....	2
2	System requirements	3
3	Software requirements	3
4	Building a three-point infrared headset.....	4
a	Tools and components	4
b	Electrical circuit diagram.....	5
c	Instructions.....	5
d	Choice of infrared LED	5
5	Installation	6
6	Configuration	7
a	Placement of Wiimote	7
b	UDP ports client and server	7
c	XML configuration file.....	8
d	Creating IDriver and ICalculator plugins.....	9
i	Step by step instructions to make a class implementing IDriver	9
7	User experience.....	12
a	Running the server	12
b	Running a demo.....	17
8	Troubleshooting.....	18
9	FAQs	19
10	Table of figures	21
11	Index.....	22





I Introduction

a Thank you

Thank you for choosing WiiR3D. We have produced a software framework which will assist open source developers to utilise head tracking technology. In addition, we have developed a variety of technical demonstrations to exemplify some of the possible uses of the WiiR3D product.

WiiR3D endeavours to provide you with a reliable product with which you will be able to expand and develop your own technology utilising head tracking.

b Intended readership

This user manual is aimed at those who wish to use the demonstrative head tracking applications that we have provided as well as use the WiiR3D server in the development of their own head tracking applications.

This manual will guide you through installation and use of the WiiR3D product, and includes detailed instructions on how to construct your own three-point head tracking headset. For those who would like to gain some insight into the thinking behind WiiR3D and its development, please refer to the technical manual.

c About

The WiiR3D product is a .NET plugin-based framework for building head tracking applications. Head tracking is a process by which an input device can observe the movement and position of a user's head relative to its own position. It identifies the six degrees of freedom that describe motion in three-dimensional space: rise, run and depth (translation along three perpendicular axes), and pitch, yaw and roll (rotation about three perpendicular axes).

Plugins can be written in any .NET-compliant language, including but not limited to C#, VB.NET and C++/CLI. The two main classes of plugin for WiiR3D are driver plugins and calculator plugins, which respectively determine the input device and calculation algorithms for providing head tracking functionality.

The WiiR3D product provides a set of plugins which offer head tracking using the Nintendo Wii Remote as the input device and Alter's equations¹ to compute user gaze from three points under weak-perspective projection.

The WiiR3D application is a lightweight server which provides the necessary head tracking data using the chosen plugins to client applications over UDP. Provided with the product are a collection of demonstrative applications which provide the intended user with examples of, and ideas for, the possible functions of head tracking using the WiiR3D product.

¹ <ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-1378.pdf>

2 System requirements

- PC with 300 megahertz or higher processor clock speed recommended; 233 MHz minimum required (single or dual processor system); Intel Pentium/Celeron family, or AMD K6/Athlon/Duron family, or compatible processor recommended
- 128 megabytes (MB) of RAM or higher recommended (64 MB minimum supported; may limit performance and some features)
- 600 kilobytes (KB) of available hard disk space for standalone server
 - Additional 3 megabytes (MB) needed for source code installation
 - Additional 50 MB needed for Panda3D² installation to run samples.
- Super VGA (800 x 600) or higher-resolution video adapter and monitor
- Keyboard and mouse or compatible pointing device

3 Software requirements

Stand-alone server usage:

- .NET 2.0 (x86) framework
- Bluetooth stack compatible with the Nintendo Wii Remote³

Server with sample applications:

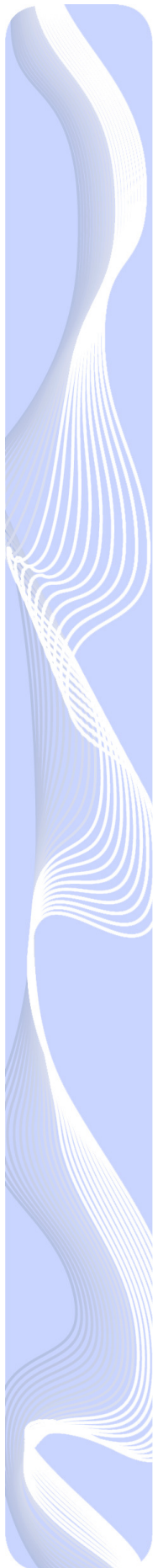
- Panda 1.5.0+
 - Python-based 3D game engine needed to run the graphical samples.
- Python 2.5.2+ (packaged with Panda installer)
 - Needed to run sample applications
- Mechanize 0.1.7b
 - Programmable client browser needed for “20Questions” sample

For driver development:

- Microsoft Visual Studio 2005 Express / Professional

² Available for free download at <http://panda3d.org/>

³ At time of writing, Microsoft BT stack and BlueSoleil 6.05.85.20070815 have been confirmed to be compatible.





4 Building a three-point infrared headset

When building a three-point infrared headset two major factors must be considered.

The first consideration is in regard to the size, dimensions and orientation of the triangular arrangement of LEDs. The triangle's base must be parallel to the user's eyes. The top of the triangle needs to lean either forwards or backwards. Also, experiments over the course of the project have found that triangles with taller centre points and more biased leans work more effectively.

The second consideration is the viewing range on the infrared source points. Most infrared LEDs have a viewing angle of approximately 17 degrees. An infrared source that provides a greater range is advised: the WiiR3D team used LEDs with a 24 degree viewing angle⁴.

Suitable head tracking equipment set is vital for proper operation and usage of the WiiR3D product. Whilst solutions without batteries do exist⁵, they generally require more work to set up correctly, and may have degeneracy issues at longer ranges.

a Tools and components

Below are the tools and components required for construction of the three-point infrared head tracking gear:

- 1 x Construction hat
- 1 x Soldering iron
- 1 x Coil of solder wire
- 1 x approx. 5cm by 10cm piece of perfboard
- 3 x TSAL7400 LED
- 3 x SPST switch
- 3 x Zener Diodes 3.3V
- 3 x Resistors 220 Ω
- 2 x Diodes IN4001
- 1 x Nine volt battery

⁴ We used the Vishay High Power Infrared Emitting Diode TSAL7400

⁵ <http://www.pixelpartner.de/openKMQen.htm> - OpenKMQ Kit #2 Freeware HT solutions

b Circuit diagram

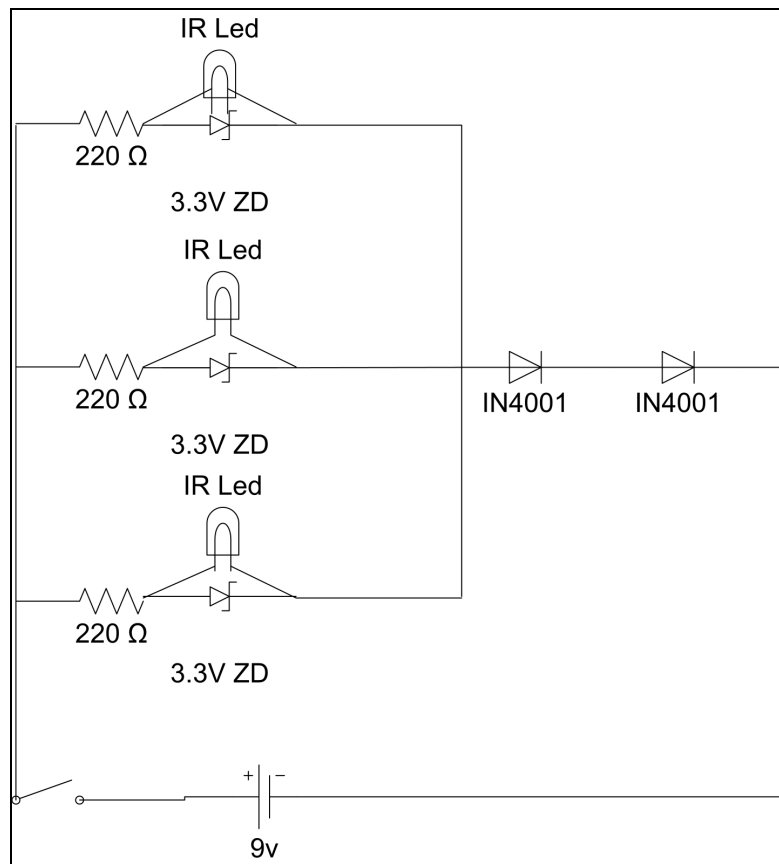


Figure 1 LED circuit diagram

c Instructions

Build the circuit to match the diagram in Figure 1. Make sure you leave 20-30 cm of wire between the circuit board and the LEDs. From the experiments carried out in this project we found that a triangle orientation with approx 6.5cm X distance⁶, 2.5cm Y distance and 6cm Z distance provided a stable model to work with. Be sure to mount the triangle on a surface that will maintain the structure of the triangle model.

d Choice of infrared LED

If a higher amp infrared LED is desired, remember to adjust the resistance flowing through the LEDs as they are quite fragile. LEDs with a higher ampere rating will produce a more intense light, and hence may be usable at greater distances. The amount of resistance required can be computed with the following formula⁷:

$$\text{resistance in ohms} = \frac{\text{power supply voltage} - \text{LED voltage drop}}{\text{LED current rating}}$$

⁶ X distance is measured by taking the base of the triangle and dividing it by two.

⁷ http://en.wikipedia.org/wiki/LED_circuit

5 Installation

Run the WiiR3D installer (installer.exe). Source code may optionally be installed.

- If the .NET 2.0 framework is not installed, it will be downloaded and installed during main installation. The installer will use the default proxy settings in Internet Explorer if a proxy is needed. Otherwise, you can download the .NET 2.0 redistributable package⁸.

What next?

If you want to:	Then
Run the server to begin using 6DoF head tracking data for your own applications	Refer to User Experience section (7a) on server usage
Run the sample applications utilising 6DoF head tracking data	Refer to the WiiR3D website ⁹ for further download and installation instructions. Additionally, refer to the User Experience section below on how to run a demo (7b).
Begin developing additional driver and calculator plugins for the server	Refer to the section below on creating plugins (7d) for more details. Be aware that plugins for performing head tracking with the Wiimote are included by default with the WiiR3D software package.

⁸ <http://download.microsoft.com/download/5/6/7/567758a3-759e-473e-bf8f-52154438565a/dotnetfx.exe>

⁹ <http://home.exetel.com.au/wiir3d>

6 Configuration

a Placement of Wiimote

The Wiimote should be in line with the user's head when they are in their start location. The Wiimote's placement is important as it greatly affects the calculation when the player is moving forward and backward.

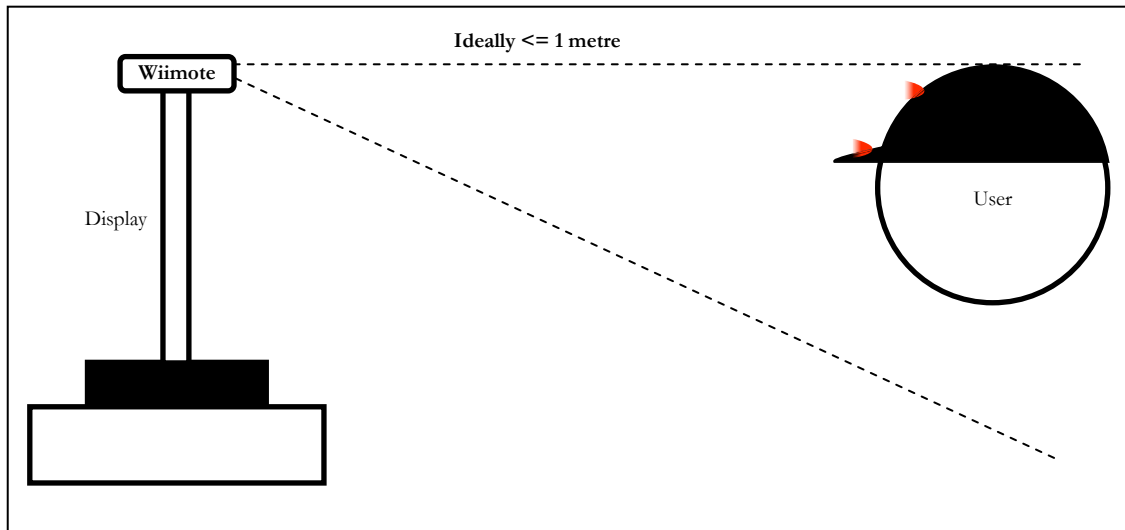


Figure 2 Placement of the Wiimote

b UDP ports client and server

The default server port is 4500.

The client may bind to any port.

c XML configuration file

The client must send through a configuration XML file before the server can start sending data back to the client. This XML document will contain a number of Gesture nodes which are used to define Recognition Patterns. These consist of:

- An attribute called *type* which is either simple or complex. Currently the only recognised type is simple.
- An attribute called *name* which is the name of the gesture which is to be registered.
- A number of children nodes called *segment*.

A Segment is used to define a section of a gesture. A segment contains:

- An attribute called *direction* which is either left, right, up, or down
- An attribute called *magnitude* which is a number representing the minimum number of degrees used to recognise the segment.

Below is an example of a simple XML document that defines a Nod and a Shake.

```
<wiir3d>

  <gesture type="Simple" name="Shake">

    <segment direction="Left" magnitude="15" />

    <segment direction="Right" magnitude="15" />

    <segment direction="Left" magnitude="15" />

  </gesture>

  <gesture name="Nod" type="Simple" >

    <segment direction="Up" magnitude="15" />

    <segment direction="Down" magnitude="15" />

    <segment direction="Up" magnitude="15" />

  </gesture>

</wiir3d>
```

d Creating IDriver and ICalculator plugins

Plugin creation involves creating class libraries in a .NET language which implement these interfaces. Please refer to the source code and the accompanying help file (WiiR3D.chm) for more details.

Implementing this interface involves creating a class in C# that implements all the interface functions. Please read the documentation on each function carefully to ensure that your class functions as required by the API. A developer only needs to implement their own driver if they wish to use a head tracking device other than the Wiimote.

i Step by step instructions to make a class implementing IDriver

Note: Similar steps can be taken to implement ICalculator.

1. Create a new project. Pick a Class Library as the project's type.
2. Add references:

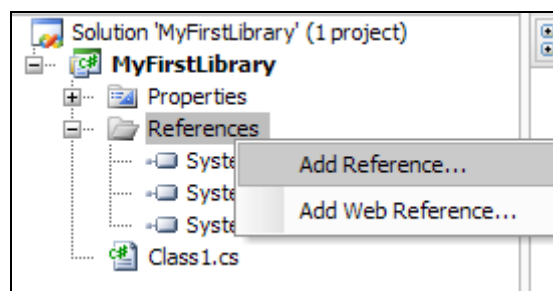


Figure 3 Implementing IDriver: add references

Browse your file system. Find the Interface you wish to use, navigate to the bin directory then either Debug or Release and load in any dll files that are in that directory.

3. Place a colon (:) and the name of the interface you wish to extend after the name of your class.

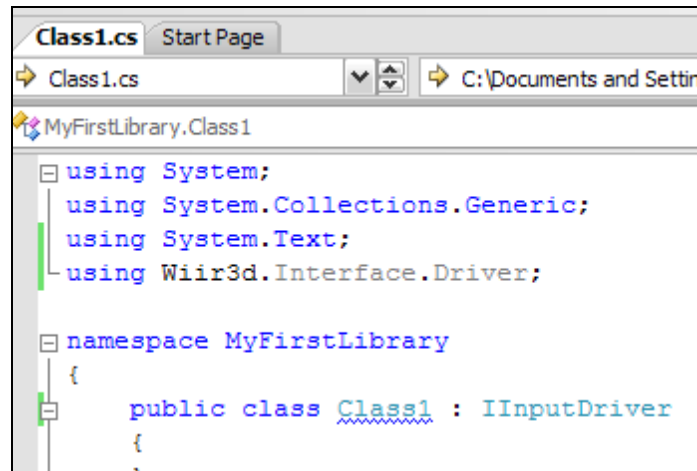


Figure 4 Implementing IDriver: colon and name of interface you wish to extend

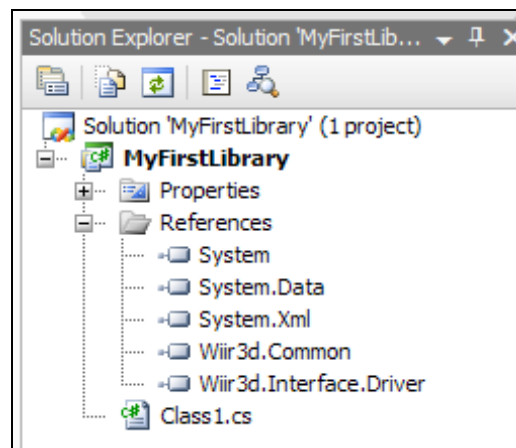


Figure 5 Implementing IDriver

4. Right click on the interface and navigate down and select Implement Interface.

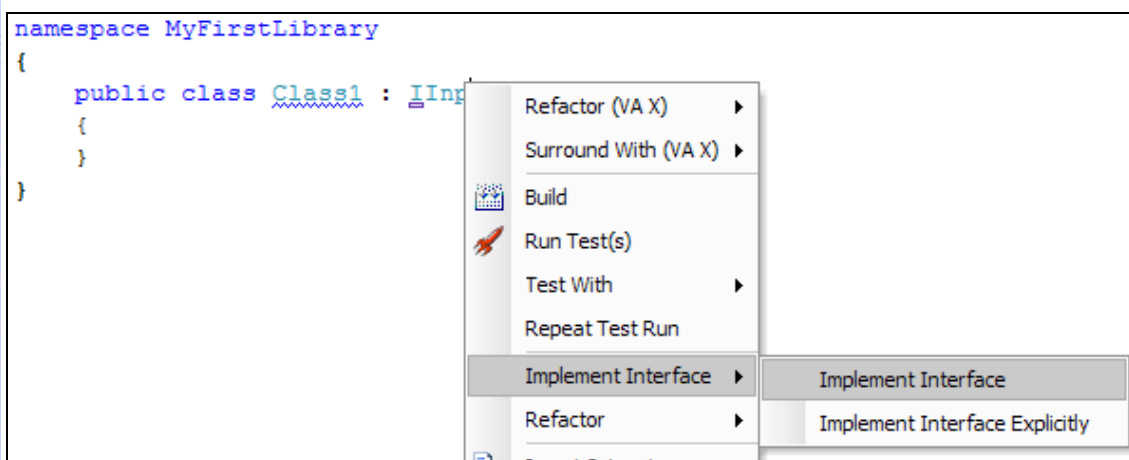
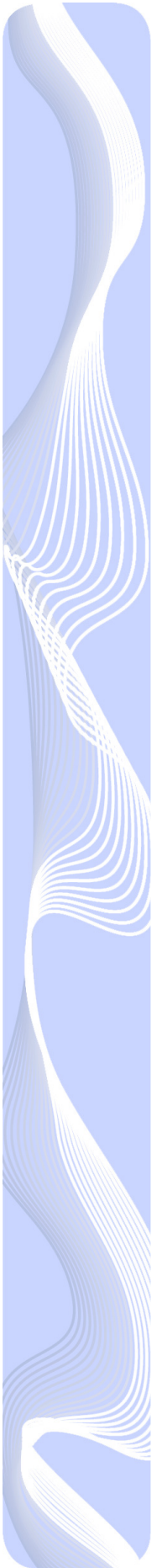


Figure 6 Implementing IDriver: right-click and implement interface

5. By this stage you should have a class that will build and become a useable, but rather worthless dll that can be used as a plugin into the WiiR3D server. To develop this class further it is recommended that the documentation in the WiiR3D API.chm file and also developers' reference code provided in predefined Drivers and Calculators is read and continually referenced.



7 User experience

a Running the server

1. Open the WiiR3D Server.

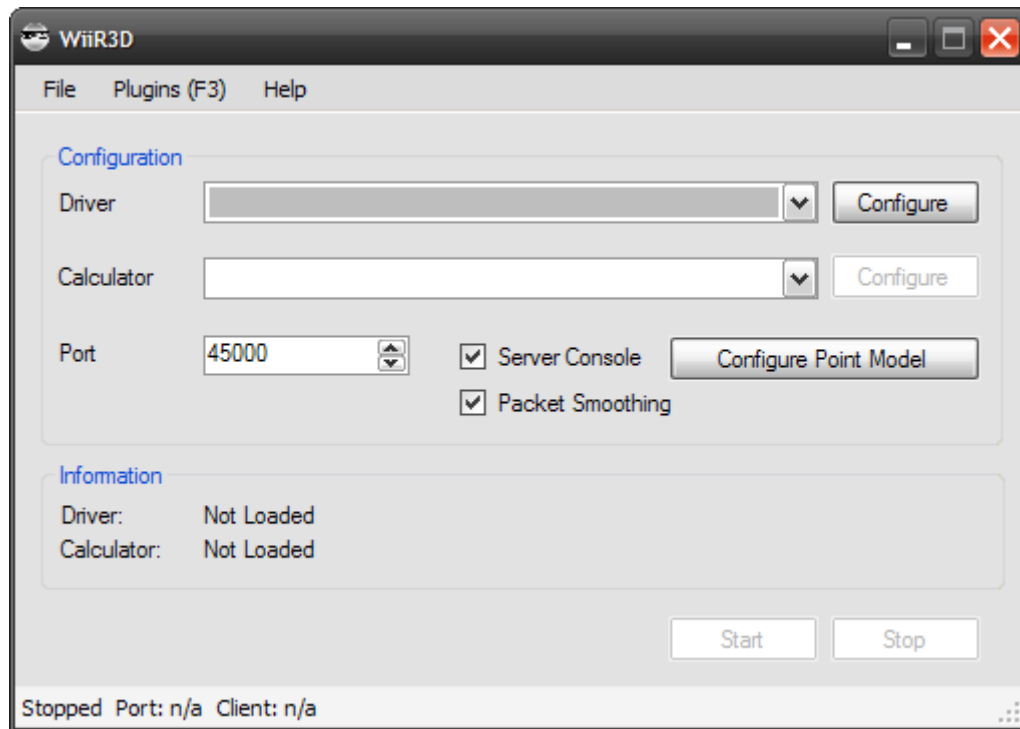


Figure 7 WiiR3D Server

2. Select the Plugins option beside the File menu item to navigate to your plugins directory.

Click 'Open' to load your selected directory.

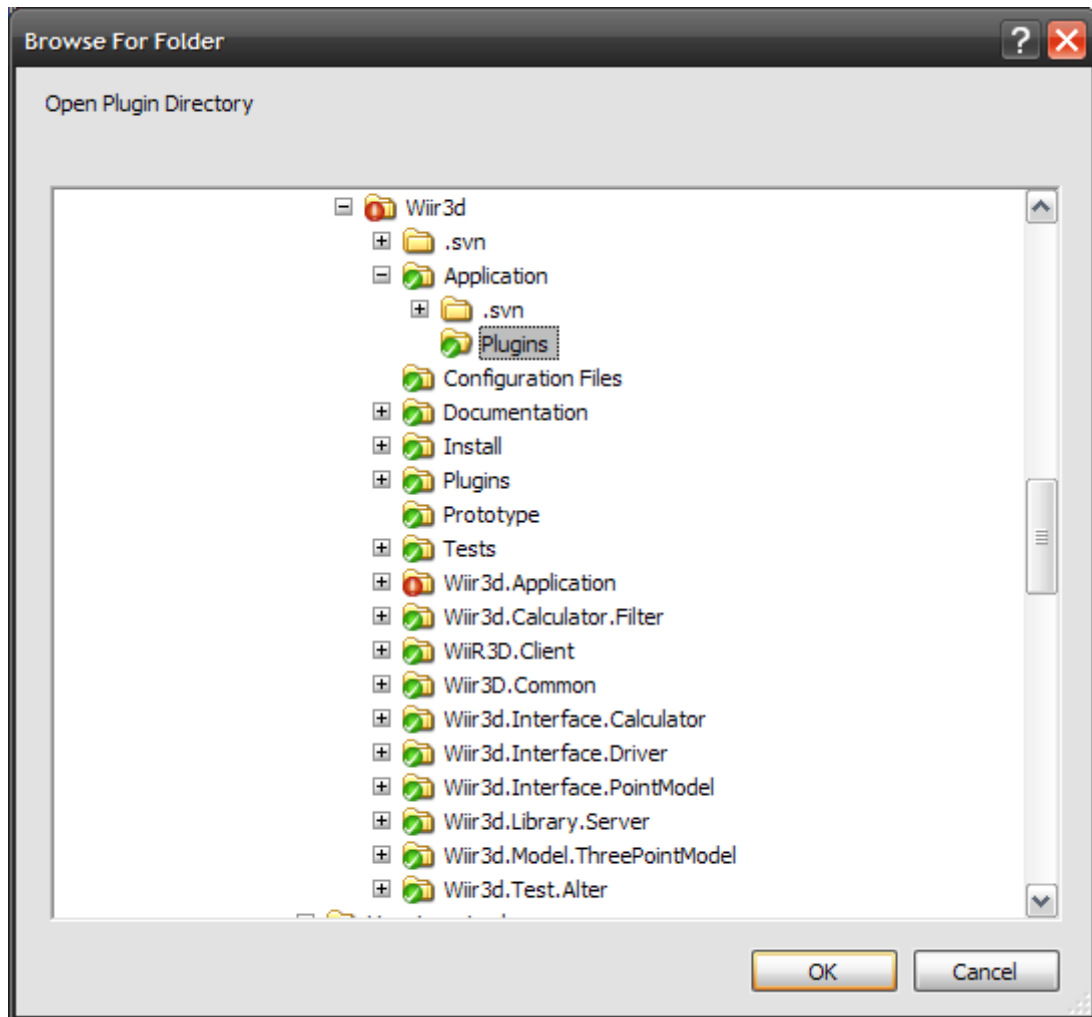


Figure 8 Selecting the plugins directory for the server

3. The drop down boxes for Driver and Calculator will now be populated with all the appropriate loadable dlls. You can now select a driver and a calculator from the drop down boxes.

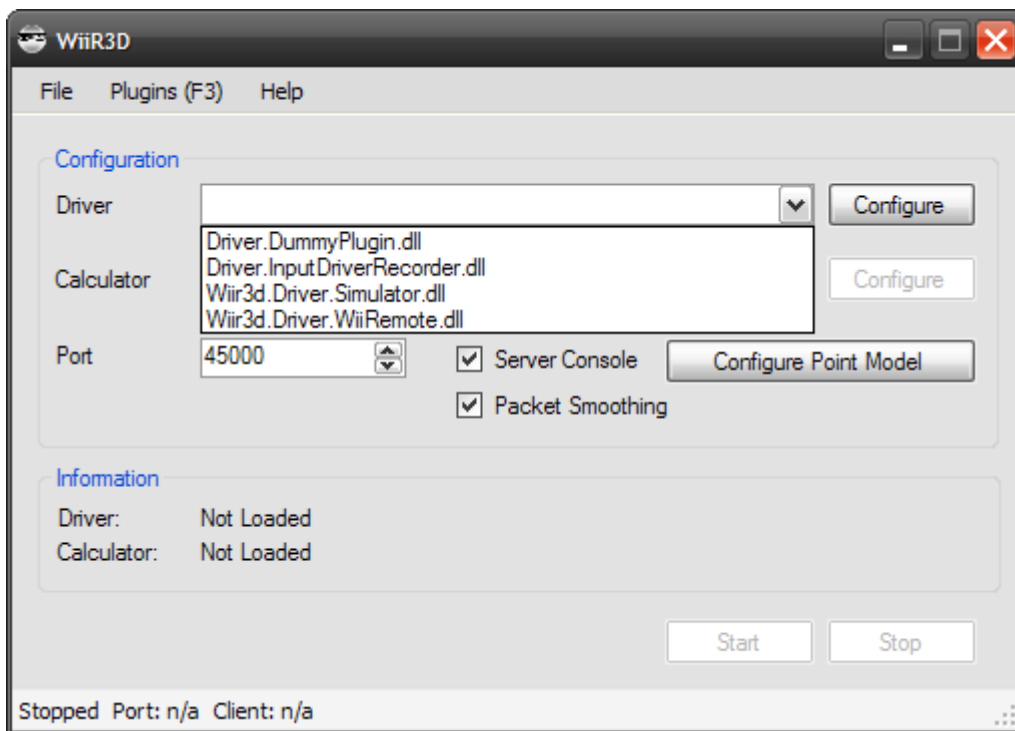


Figure 9 Server with populated drop down boxes

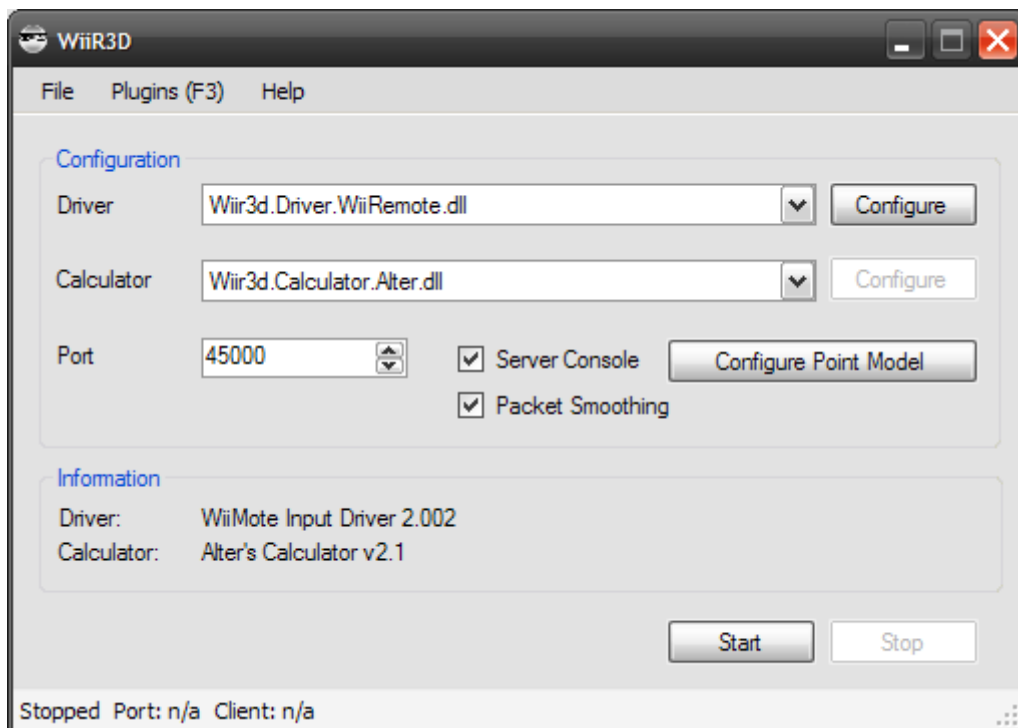


Figure J Server with driver and calculator loaded

4. Select the 'Configure Point Model' button to bring up the Point Configuration window. Identify and measure the distances on your own headset as indicated in the diagrams, then enter the x , y and z distances (in centimetres) here.

Click 'Apply' to confirm your changes.

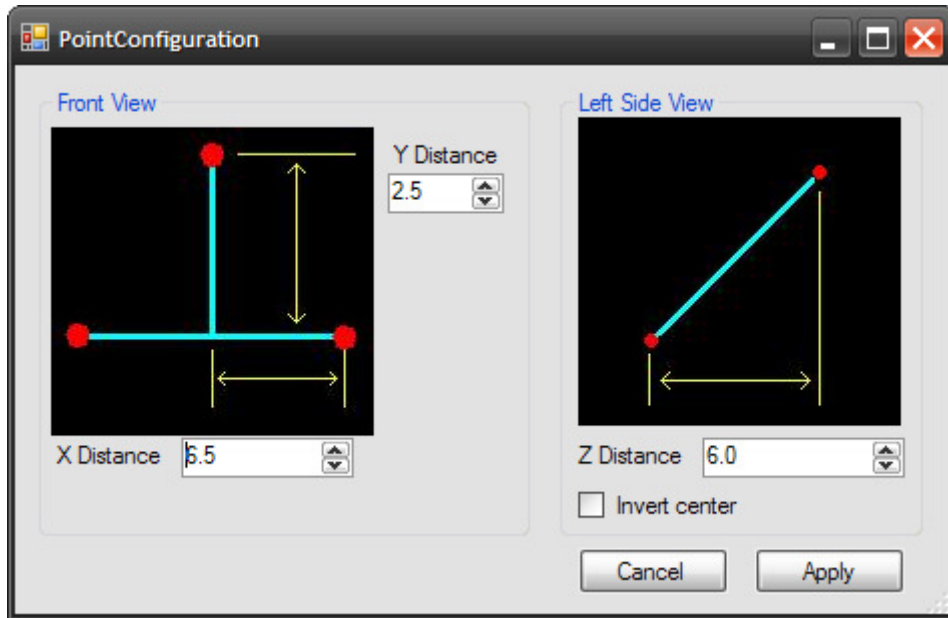


Figure 11 Configuring the point model

5. If you would like the server console to display while the server is running, check the 'Server Console' checkbox.

6. Click to 'Start' to begin running the WiiR3D server. If you have checked the 'Server Console' checkbox, the server console will appear.

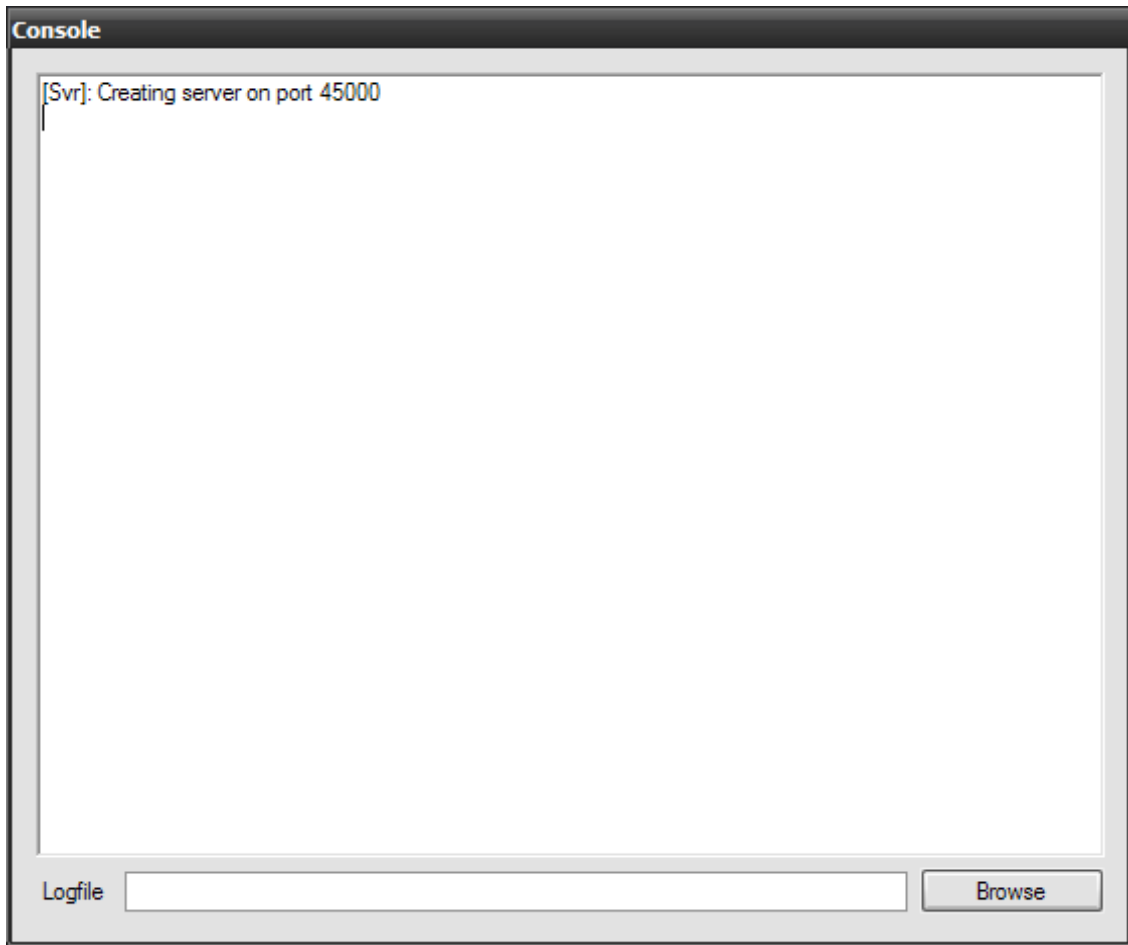


Figure 12 Server console

b Running a demo

To run a demo the server need to be running and waiting for a client to connect.

1. Connect you input device.
2. Follow the steps above (7a) to launch and run the server.
3. After the server has been launched, navigate to the demo directory.
4. Pick a demo and double click on it. The demo should then appear in a new window.

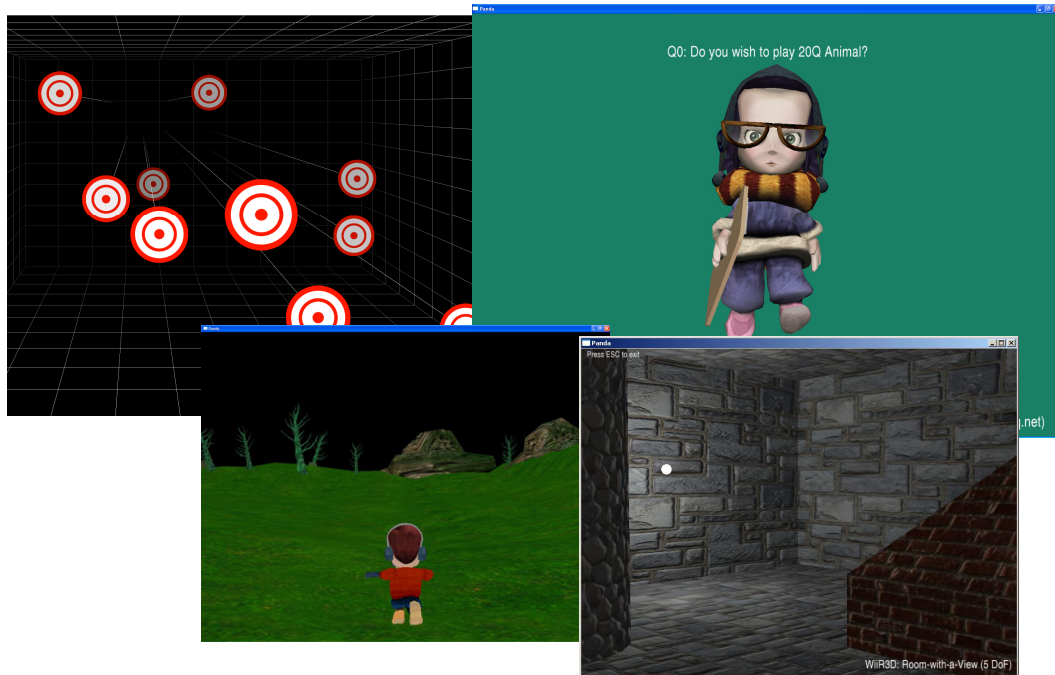


Figure 13 Demonstrative applications



8 Troubleshooting

My view is jittery.

Have you got smoothing activated? If smoothing is not activated, then activate it by checking the box in the WiiR3D server.

If you have smoothing activated and your view is still jittery, make sure that you are not sitting too far away from your input device. For the Wiimote, for example, ideally you should be within one metre of the infrared camera. Move closer and try again.

If you have smoothing activated and you are not sitting too far from your input device but your view is still jittery, there may be a problem with your hardware causing your LEDs to flicker off and on, which appears as a jittery view. Check your headset.

If you have exhausted all of the above solutions and your view is still jittery, you may need to reconfigure the formation of your LEDs. Refer to section 4 (Building a three-point infrared headset) for information regarding the ideal arrangement of your LEDs for the Alter calculator. For further information on the ideal triangle formation, you can check Alter's paper¹⁰.

Why can't my Wiimote connect, and how can I fix it?

Errors in Wiimote connectivity typically occur if more than one Wiimote is paired in the device manager. Due to the way devices are managed in Windows, there is no way to tell the difference between a live connected Wiimote and a Wiimote which was paired previously, but not currently connected. One possible fix is to bring up the Device Manager (Right-click My Computer > Properties > Hardware > Device Manager), scroll down to Human Interface Devices, and disable anything titled 'Bluetooth HID Unclassified Device' or 'Bluetooth Joystick/Gamepad.' Next, begin re-enabling each device one at a time and attempt to reconnect from within the WiimoteLib configuration form.

Even though I am looking straight at the screen, the demo thinks I am looking in a different direction.

The LEDs in your headset may not be uniformly horizontal. Ensure that your infrared LEDs are pointing straight ahead when you are looking directly at the screen; the length of the LEDs should be horizontal, parallel to the floor.

When I double click on one of your demos, it opens but it doesn't perform head tracking. Why not?

Make sure that your input device is correctly connected. If it is, then check that you have the server running.

¹⁰ <http://dspace.mit.edu/bitstream/handle/1721.1/6611/AIM-1378.pdf?sequence=2>

9 FAQs

What is head tracking?

Head tracking is a process by which an input device can observe the movement and position of a user's head relative to its own position. It identifies the six degrees of freedom that describe motion in three-dimensional space: rise, run and depth (translation in three perpendicular axes), and pitch, yaw and roll (rotation about three perpendicular axes). Tracking head position and pose allows programmes to produce user-dependant images.

Do I have to use the Wiimote as my input device?

No. WiiR3D uses a driver plugin, which means that any input device can be used as long as there exists an appropriate driver plugin for it. Plugins can be written in any .NET-compliant language. If you have a different input device and driver, simply navigate to it using the browse button for the driver in the WiiR3D server.

I want to use my own head tracking calculations. Is this possible?

Yes. WiiR3D uses a calculator plugin, which means that you can write your own head tracking calculations as long as you use a .NET-compliant language. If you have written your own calculator simply navigate to it using the browse button for the calculator in the WiiR3D server.

Why do you use three infrared LEDs?

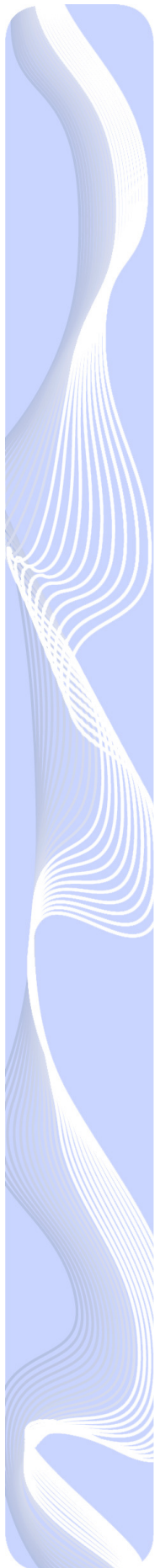
Three is the minimum number of points required to observe six degrees of freedom in motion. While it is possible to use more LEDs for head tracking, the equations which we utilised in our head tracking calculator only required the minimum three points.

How does gesture recognition work?

The gesture recognition component treats a gesture as an ordered sequence of segments. A gesture segment is simply a direction (up, down, left, or right) representing the user's head movement together with a magnitude in degrees. Each registered gesture operates as a tiny finite state machine which remembers segments it has seen and what segments it needs to look out for in order to complete a gesture. Simple pre-processing of the points is applied to weed out noisy points, or points which are considered to not have changed considerably since the last poll. Once a gesture is recognised, the client is notified via a gesture alert packet and may choose to process or ignore the event.

How does the packet smoothing work?

The packet smoothing function takes the head position coordinates from the calculator and applies a weighted moving average to a list of previous raw head positions. The algorithm takes into account each of the previous head positions according to a ratio that defines how important the previous positions are, and then averages them.



How do you observe six degrees of freedom from three LEDs?

We followed Alter's equations for our calculations. Alter's equations assume weak-perspective projection of model points onto the camera (an orthographic projection with a fixed scaling), solved for the constant scale S and heights h_1, h_2 from one of the image points to get back the original 3D model points m_0, m_1 and m_2 , as shown below:

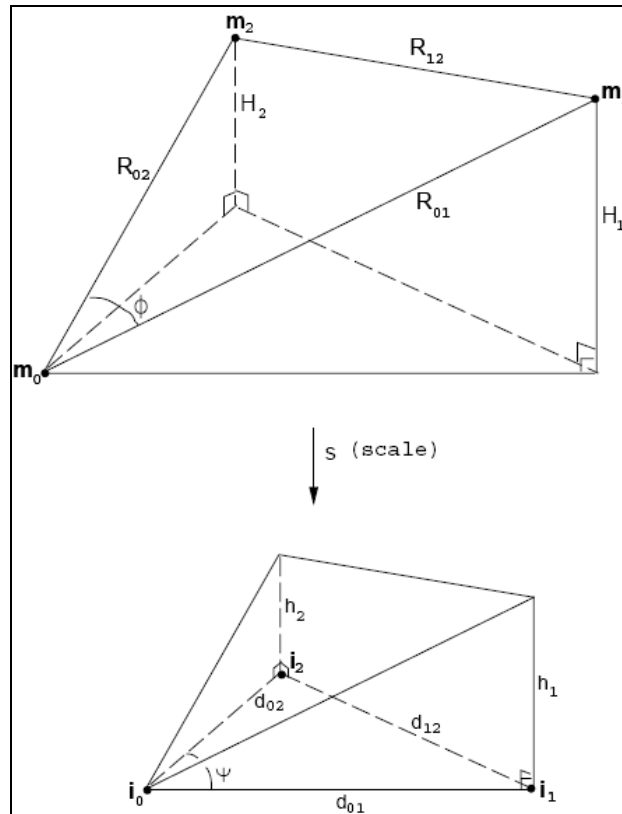


Figure 14 Model points m_0, m_1 , and m_2 undergoing orthographic projection plus scale to produce image points i_0, i_1 , and i_2 ¹¹.

The rigid transformation which aligns the sensed model to the base model is computed, and from this the pitch, yaw and roll angles are extracted.

Finally, depth translation is simply estimated from the image triangle and the base point model configuration.

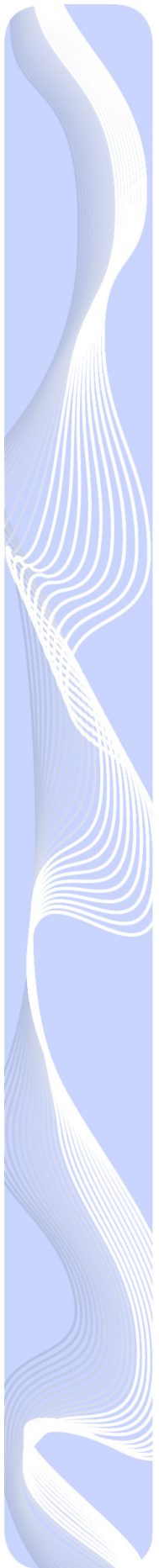
What does 'WiiR3D' mean?

WiiR3D is the name we came up with for the project in the very early days. 'Wii' pays tribute to the Wiimote, the input device on which we focussed and for which we built a driver plugin. 'R3D' is for 'revolutionising 3D': head tracking technology really opens the door for a revolution in the way that users interact with 3D environments. WiiR3D is open source, so developers everywhere can begin taking advantage of head tracking.

¹¹ From '3D Pose from 3 Corresponding Points under Weak-Perspective Projection'
<ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-1378.pdf>

I O Table of figures

Figure 1 LED circuit diagram	5
Figure 2 Placement of the Wiimote	7
Figure 3 Implementing IDriver: add references	9
Figure 4 Implementing IDriver: colon and name of interface you wish to extend	10
Figure 5 Implementing IDriver	10
Figure 6 Implementing IDriver: right-click and implement interface	10
Figure 7 WiiR3D Server	12
Figure 8 Selecting the plugins directory for the server.....	13
Figure 9 Server with populated drop down boxes.....	14
Figure J Server with driver and calculator loaded.....	14
Figure 11 Configuring the point model.....	15
Figure 12 Server console	16
Figure 13 Demonstrative applications.....	17
Figure 14 Model points m_0 , m_1 , and m_2 undergoing orthographic projection plus scale to produce image points i_0 , i_1 , and i_2	20





Index

- Alter, T.D, 1, 17, 19
- Bluetooth, 2, 17
- Building a three-point infrared
 - headset, 4
- Connection, 16
- Demonstrative applications, 16, 20
- Gesture recognition, 7
- Head tracking, 1, 18
 - Pose, 19
 - Translation, 4, 5
- Headset, 14
- ICalculator, i, 8
- IDriver, i, 8, 9, 20
- Infrared, 3
- Input, i, 8, 9, 13, 20
- Install, i, 5
- LED, infrared, i, 3, 4, 17, 18, 19, 20
- Mechanize, 2
- Panda3D, 2
- Plugin, 8
 - Caclulator, i, 8
 - Calculator, 1, 13, 17, 19
 - Driver, i, 8, 9, 13, 20
- Point Model, 14
- Python, 2
- Server, 2, 11, 13, 14, 15, 20
- Software requirements, i, 2
- System requirements, i, 2
- UDP, i, 1, 6
- Wiiimote, i, 1, 2, 5, 6, 8, 17, 18, 19, 20
- XML, i, 7