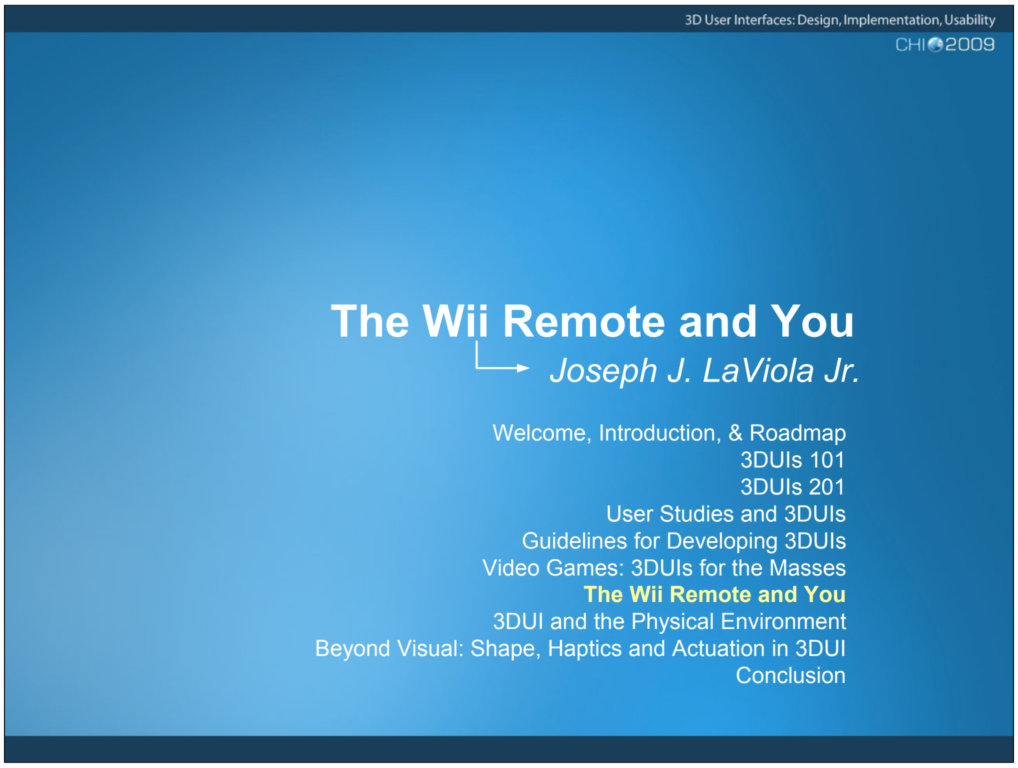
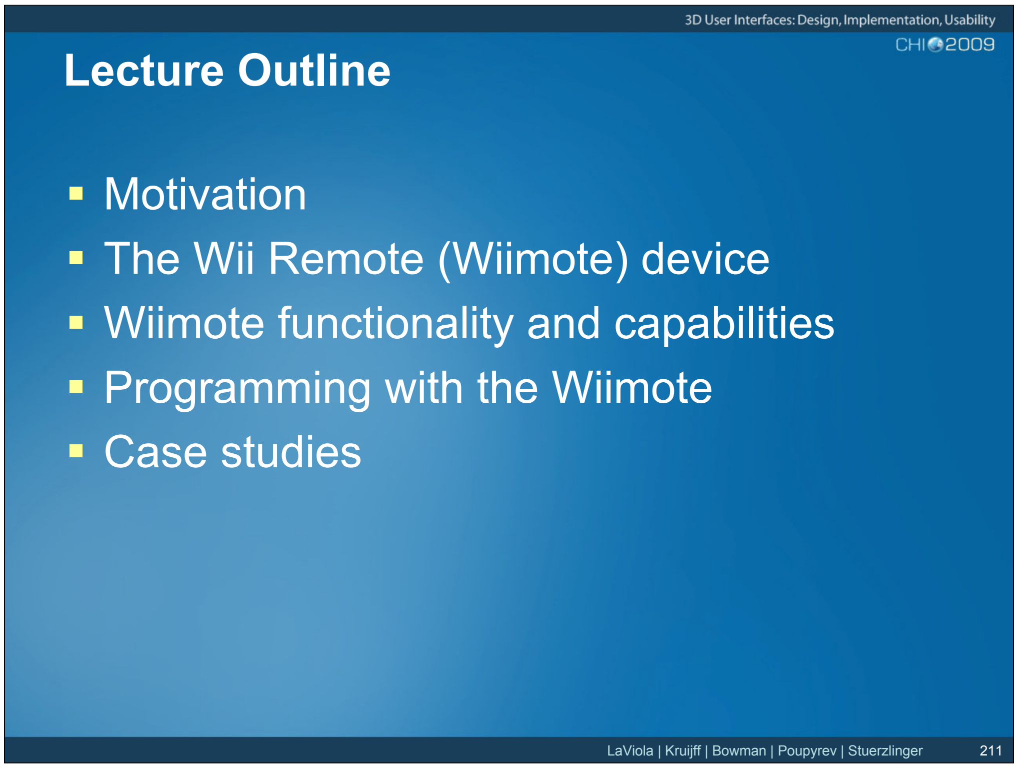


The Wii Remote and You

Joseph J. LaViola Jr.

Welcome, Introduction, & Roadmap
3DUIs 101
3DUIs 201
User Studies and 3DUIs
Guidelines for Developing 3DUIs
Video Games: 3DUIs for the Masses
The Wii Remote and You
3DUI and the Physical Environment
Beyond Visual: Shape, Haptics and Actuation in 3DUI
Conclusion


3D User Interfaces: Design, Implementation, Usability

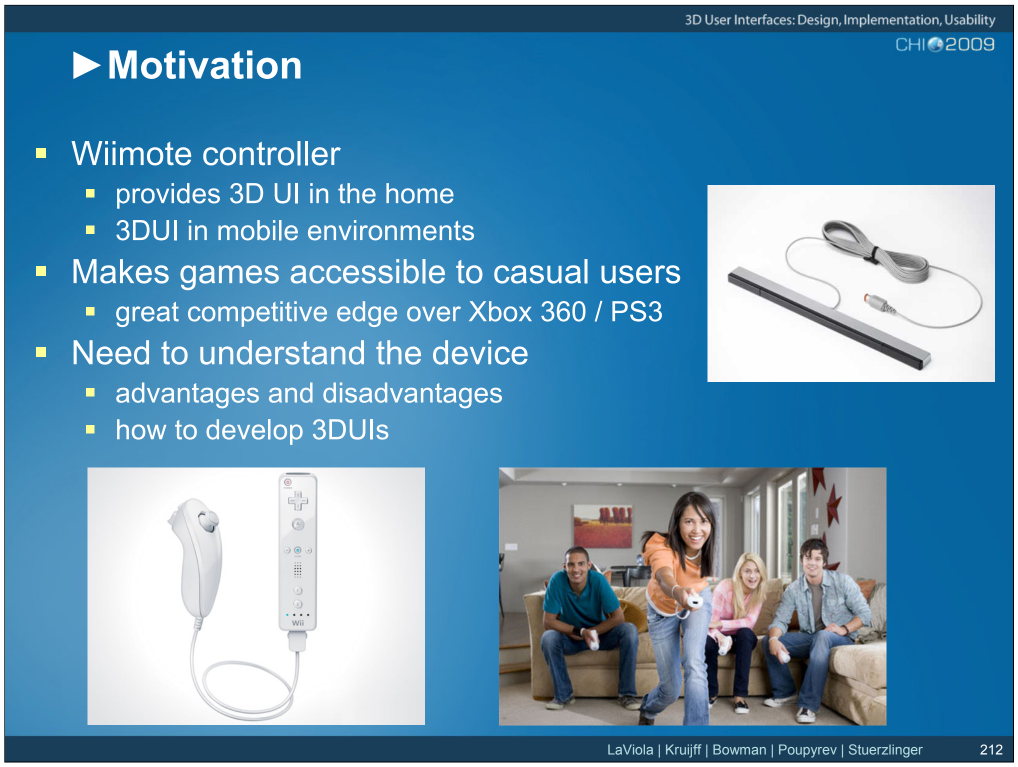
CHI 2009

Lecture Outline

- Motivation
- The Wii Remote (Wiimote) device
- Wiimote functionality and capabilities
- Programming with the Wiimote
- Case studies

LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger 211

In this lecture, we are going to discuss the Nintendo Wii Remote input device and how it can be used to develop 3D user interfaces. We will examine the device's functionality and capabilities as well as its strengths and weaknesses. We will also look at how to program using the Wii Remote (Wiimote) and connect it to a PC. Finally, we will go through several case studies showing how researchers have used the Wiimote to explore 3D spatial interaction in video games and robotics.

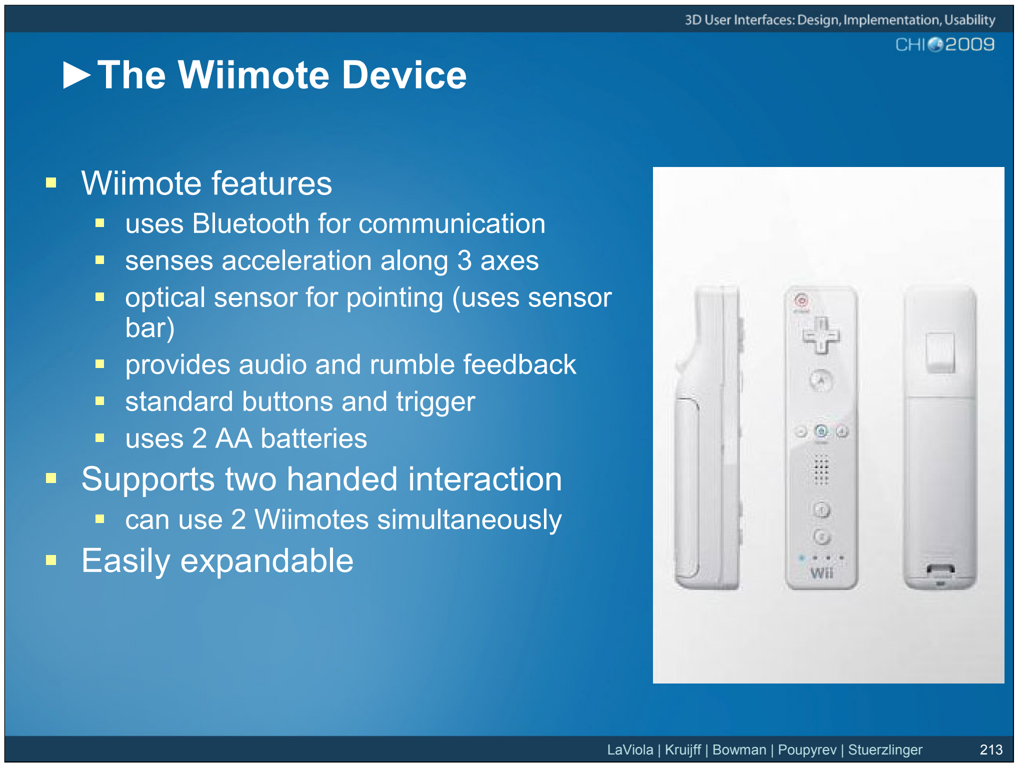


▶Motivation

- Wiimote controller
  - provides 3D UI in the home
  - 3DUI in mobile environments
- Makes games accessible to casual users
  - great competitive edge over Xbox 360 / PS3
- Need to understand the device
  - advantages and disadvantages
  - how to develop 3DUIs

The latest gaming console from Nintendo, the Wii, has made one of the most important technological innovations in gaming technology with respect to 3D user interfaces. The key innovation of the Wii is its controller, the Wiimote. This input device not only acts as a gamepad, but makes games accessible to the casual gamer because it can sense 3D motion. The device not only makes 3D spatial interfaces available to the casual gamer, but also provides researcher, developers, and hobbyists with the ability to explore their own ideas because the device can be connected to a standard computer.

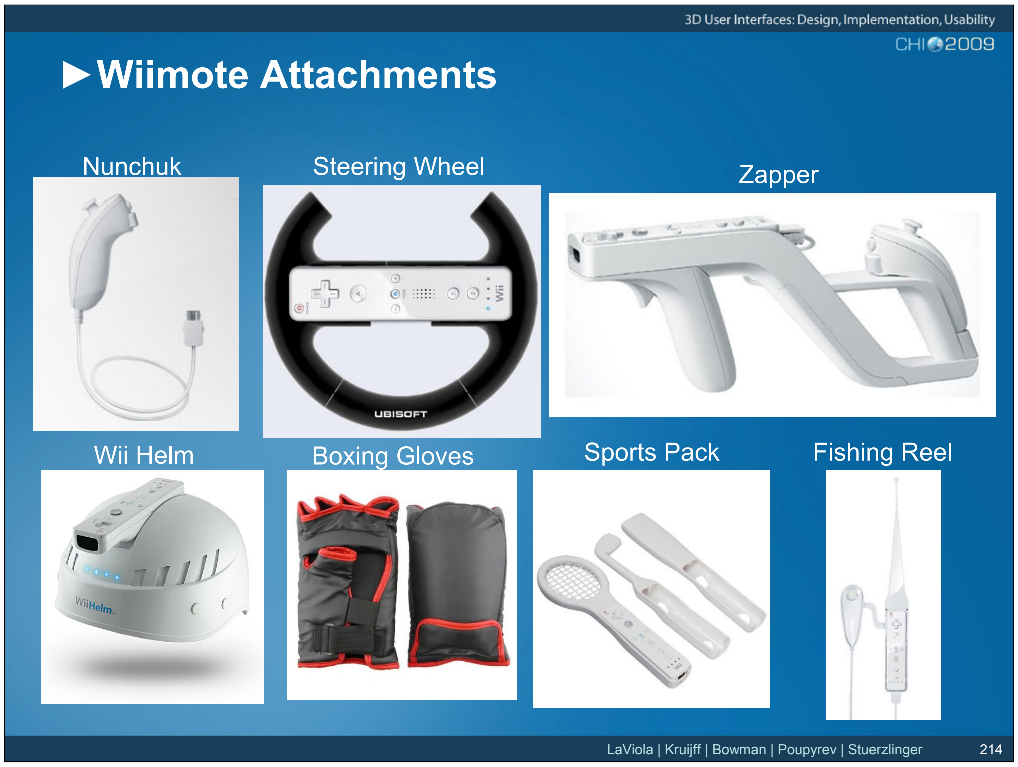
Images from Nintendo and http://blog.mlive.com/manzero/



►The Wiimote Device

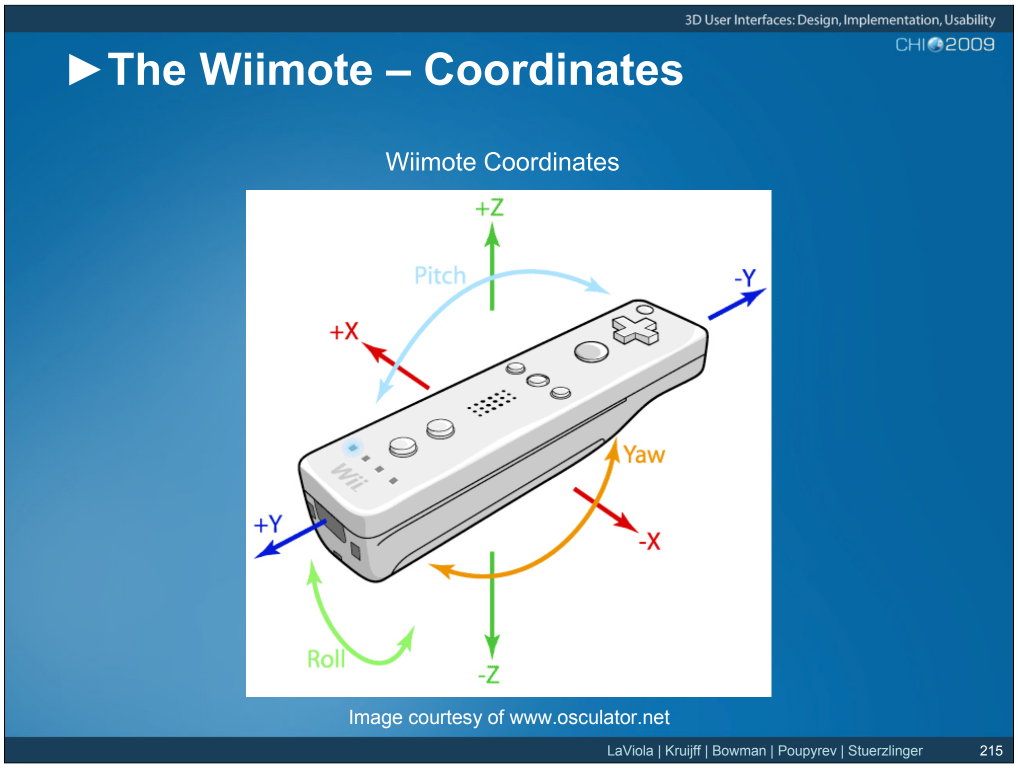
- Wiimote features
  - uses Bluetooth for communication
  - senses acceleration along 3 axes
  - optical sensor for pointing (uses sensor bar)
  - provides audio and rumble feedback
  - standard buttons and trigger
  - uses 2 AA batteries
- Supports two handed interaction
  - can use 2 Wiimotes simultaneously
- Easily expandable

The Wiimote has several interesting features. First, it uses Bluetooth for communication making it a wireless device. Second, it senses acceleration along 3 axes and has an optical sensor for pointing (when a sensor bar is used). This acceleration detection gives the Wiimote its power in that it lets users interact with games spatially (e.g., swinging a bat or tennis racket, or golf club). The device also has audio and rumble feedback. One of the best features of the Wiimote is it was designed to be easily expandable in terms of the input devices that can be attached to it.



▶Wiimote Attachments

This slide shows the many input peripherals that can be used with the Wiimote. Note that the Nunchuk also has accelerometers embedded in it.



▶ The Wiimote – Coordinates

Wiimote Coordinates

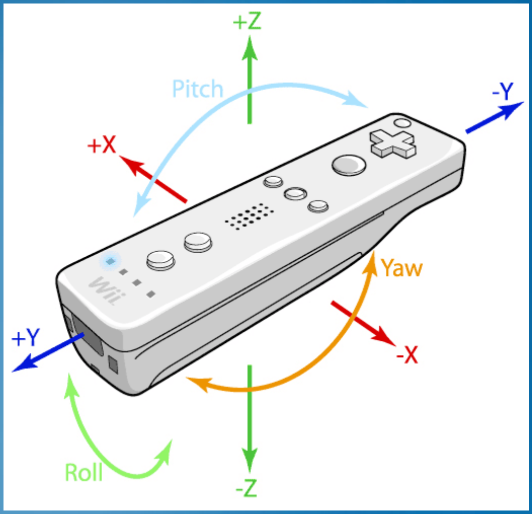
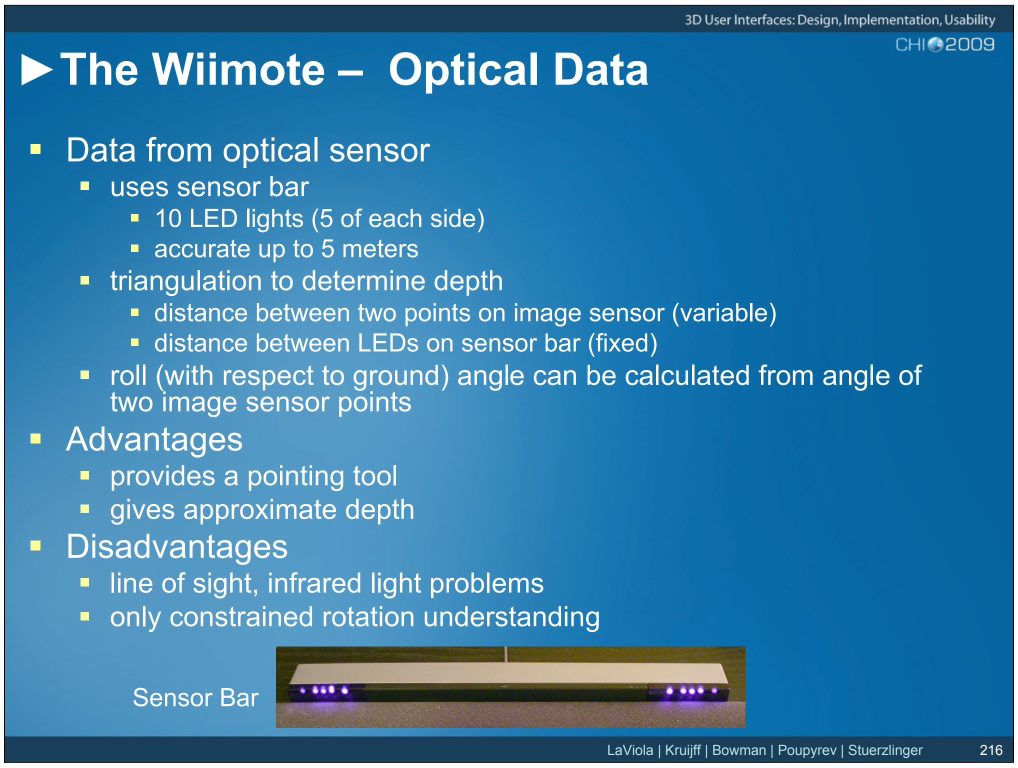


Image courtesy of www.osculator.net

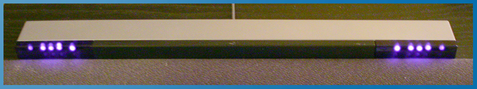
Although the Wiimote is a relatively simple device in terms of what it provides to the user, compared to other more sophisticated input devices, it can still be somewhat confusing to talk about in terms of what data it actually gives the user. The combination of its accelerometers and image sensor can provide a variety of useful information to the research and developer. Some of this information comes directly from the sensors, but other pieces of information come from deriving it mathematically from the base sensors. Thus, it is important to have a frame of reference in which to talk about the Wiimote. The image in the slides shows a canonical coordinate frame we will use to discuss the details of the device and is important to remember (for visualization purposes) when developing 3D spatial user interfaces with it.



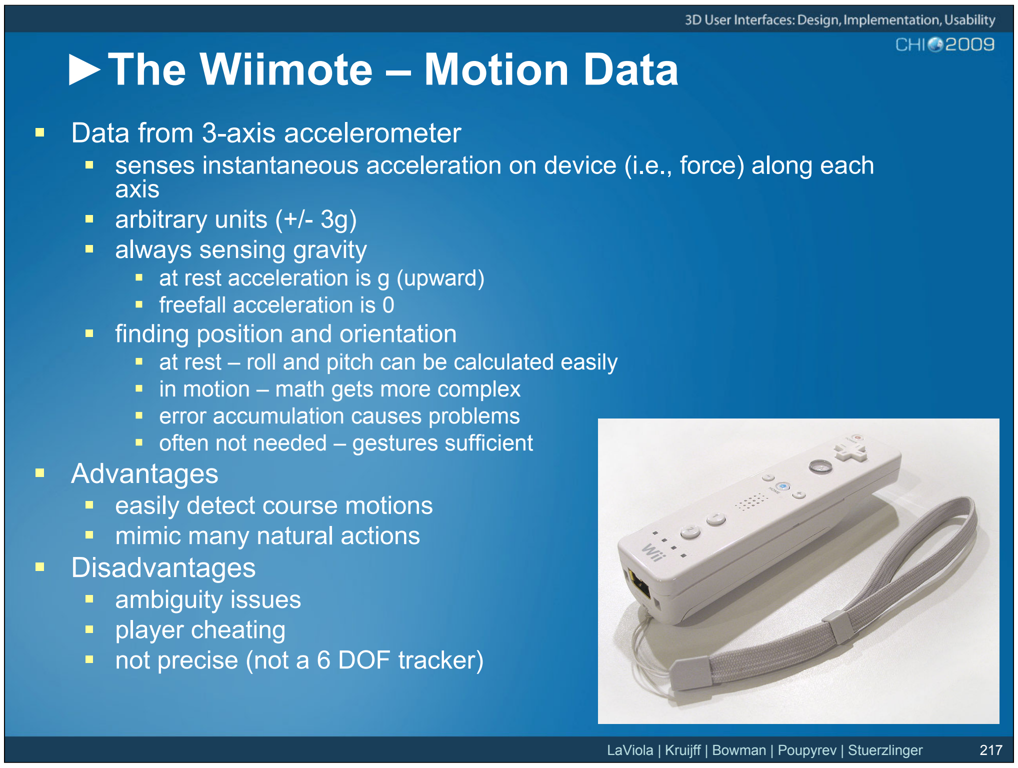
▶The Wiimote – Optical Data

- Data from optical sensor
  - uses sensor bar
    - 10 LED lights (5 of each side)
    - accurate up to 5 meters
  - triangulation to determine depth
    - distance between two points on image sensor (variable)
    - distance between LEDs on sensor bar (fixed)
  - roll (with respect to ground) angle can be calculated from angle of two image sensor points
- Advantages
  - provides a pointing tool
  - gives approximate depth
- Disadvantages
  - line of sight, infrared light problems
  - only constrained rotation understanding

Sensor Bar



There are two primary inputs that can be utilized with the Nintendo Wiimote. The first one is information from its optical sensor. The optical sensor makes use of a sensor bar, a device that has 10 LED lights (5 on each size). The LEDs farthest away from the center are pointed slightly away from the center, the LEDs closest to the center are pointed slightly inwards, while the rest are pointed forward. This spread helps to improve the optical sensor's range. When the Wiimote is pointed at the sensor bar, it picks up two points from the LED arrays. This allows the Wiimote to act as a pointing device. Given a fixed distance between these points on the sensor bar and the variable distance between the two points on the image sensor, the Wiimote hardware can determine how far it is from the sensor by using triangulation. Thus, when the Wiimote is pointed in the direction of the sensor bar, it can provide information on depth. However, this calculation works best when the Wiimote device is pointing directly at the sensor bar (orthogonally) since the image sensor cannot tell that the distance between the two points it sees from the sensor bar is because the Wiimote is at a given distance away or it is rotated. This represents one of many ambiguities with the Wiimote device. One other issue with the Wiimote's optical sensor is that it will pick up any infrared light in the room, which can cause accuracy problems.



3D User Interfaces: Design, Implementation, Usability

CHI 2009

►The Wiimote – Motion Data

- Data from 3-axis accelerometer
  - senses instantaneous acceleration on device (i.e., force) along each axis
  - arbitrary units (+/- 3g)
  - always sensing gravity
    - at rest acceleration is g (upward)
    - freefall acceleration is 0
  - finding position and orientation
    - at rest – roll and pitch can be calculated easily
    - in motion – math gets more complex
    - error accumulation causes problems
    - often not needed – gestures sufficient
- Advantages
  - easily detect course motions
  - mimic many natural actions
- Disadvantages
  - ambiguity issues
  - player cheating
  - not precise (not a 6 DOF tracker)

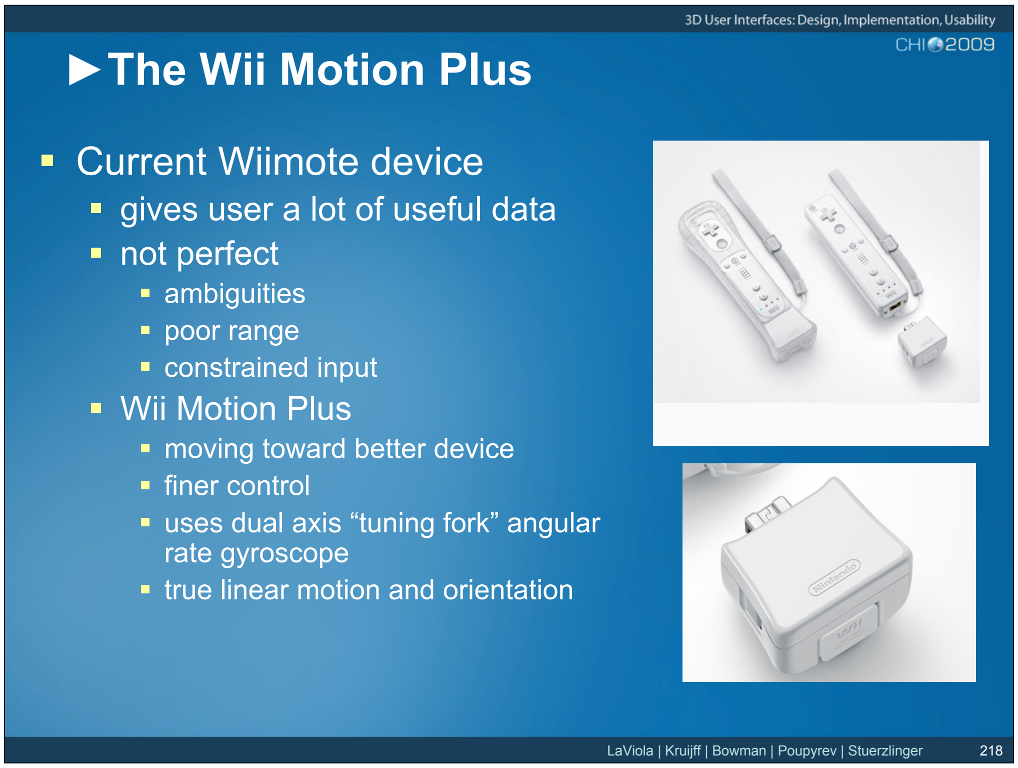
LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger 217

The second input from the Wiimote that researchers and developers can use to create 3D spatial interfaces is based on the device's 3-axis accelerometer. Thus, the Wiimote has the ability to sense motion. More precisely, the Wiimote device senses instantaneous acceleration on itself along each axis. Some people say that what the Wiimote is really sensing is force on the small masses in the accelerometer and it subject to some debate. However, for the purposes of this lecture we will use acceleration. The 3-axis accelerometer provides three values in arbitrary units that represent the acceleration on the device in the x, y, and z directions. When we talk about acceleration measurements with the Wiimote we can express them in g's. Typically, motions the Wiimote senses are between 3g and -3g, which is fine for dealing with typical user motions. One thing that must be considered when using the accelerometer data is the notion that the device is always sensing gravity. In other words, when the Wiimote is completely stationary (i.e., at rest), it will show an acceleration of 1g in the upward direction (i.e., +Z direction). If someone was to drop the Wiimote and let it freefall, its cumulative acceleration readings would be approximately zero. This simple fact regarding the Wiimote device must be taken into account when using the device. This is the main reason why, when the device is not moving, it is not possible to calculate yaw (the sensor bar and image sensor can do this). However, when the device is at rest, calculating the roll and pitch of the device is straightforward (driving games such as Mario Kart make use of this information). The following equations calculate roll and pitch under this condition:

roll = $\arctan2(a_z,a_x)$

pitch = $\arctan2(a_z,a_y)$

When the device is moving, the mathematics gets more complicated (see www.wiili.org/motion_analysis for details). Typically the data needs to be doubly integrated to get the orientation and this can be problematic due to accelerometer error accumulation. Kalman filters and the like are often used to help deal with the error accumulation problem. In many cases, finding precise orientation or position is not needed as the raw data values can be used to recognize various gestures. With a reasonable classifier, there are many different gestures that can reliably be detected. Of course, ambiguity issues also play a role in the number of gestures one can recognize with the Wiimote device.



3D User Interfaces: Design, Implementation, Usability

CHI 2009

▶The Wii Motion Plus

- Current Wiimote device
  - gives user a lot of useful data
  - not perfect
    - ambiguities
    - poor range
    - constrained input
  - Wii Motion Plus
    - moving toward better device
    - finer control
    - uses dual axis "tuning fork" angular rate gyroscope
    - true linear motion and orientation

LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger 218

Overall, the Wiimote give the 3DUI developer a lot of data to create interesting 3D spatial interfaces. It is, however, by no means perfect. The ambiguities in the accelerometer data and the image sensing can cause problems when trying to detect interesting gestures and to find orientation and depth information. Another issue with the device is that the image sensor range can be problematic. Given the sensor bar is fixed, it can cause problems with large size TVs where the signal is lost at the edges. Finally, and probably most importantly, the device provides "constrained input". In other words, you can only extract certain types of data from the Wiimote under given assumptions such as how it is pointing and whether it is at rest or in motion. This makes transitions between interface techniques difficult and is also one of the main reasons why you see most Wii games ask users to hold the Wiimote in certain positions before games start.

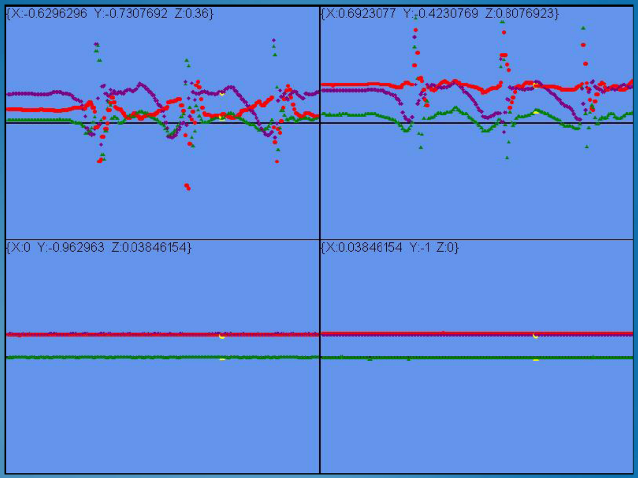
One area of improvement to the Wiimote is the Wii Motion Plus. This device should be hitting stores at about the time of this conference. The Wii Motion Plus make use of dual axis angular rate gyroscopes to improve motion detection. Thus, in combination with the accelerometers and, to a lesser extent, the image sensors, the device will be able to detect true linear motion and orientation. Having this data will go a long way to improving the device's accuracy and reducing some of its ambiguities. However, it still will not be able to provide reliable position information in 3D.

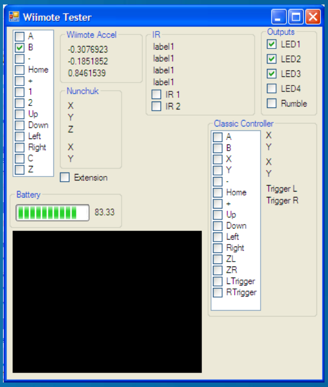
3D User Interfaces: Design, Implementation, Usability

CHI 2009

Visualizing Wiimote Data

- Important to see data to understand device





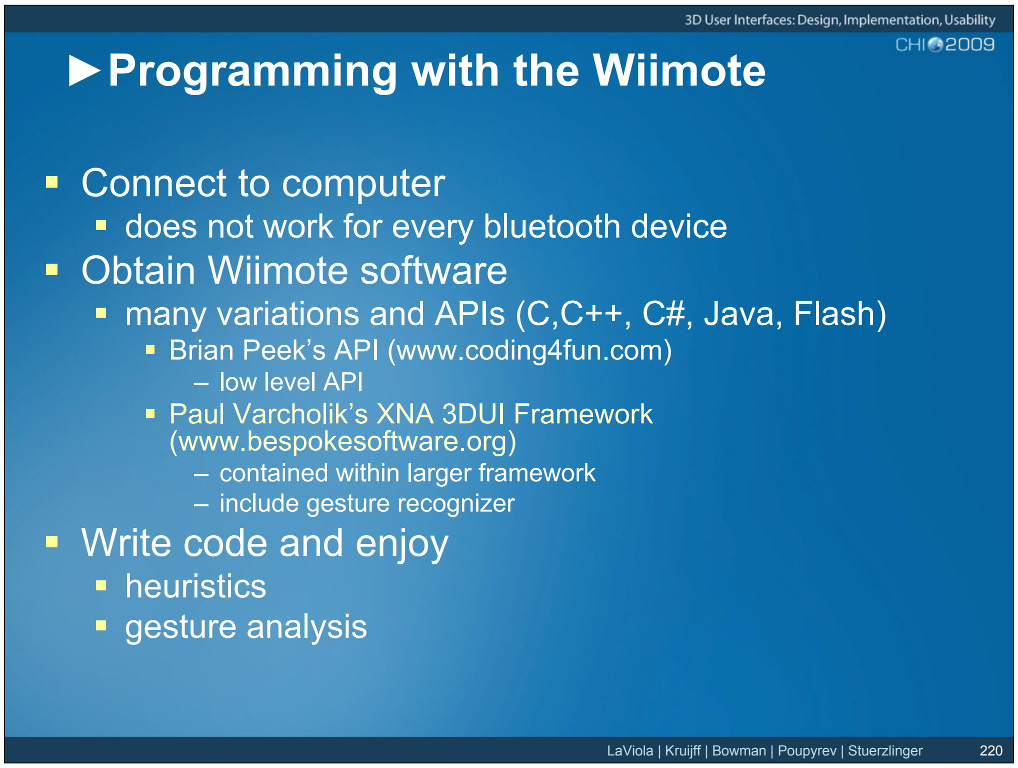
LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger 219

Working with Wiimote data can be somewhat challenging to the beginner because the accelerometer data presents a counter-intuitive picture of what the device is doing. Most people who think about 3D spatial interfaces, think about them in terms of the position and orientation of the input device in 3D space. However, the Wiimote does not provide all of this information. Thus, it is important to be able to visualize the motion data in some way to get a better feel for how the motion signals directly correspond to Wiimote motion. The image on the left shows a screen shot of a simple application that plots out the motion data while the image on the right shows a screenshot of Brian Peek’s Wiimote tester, an application that is useful for showing that the Wiimote is working properly by showing the numerical values for the accelerometers. This tester is available on Brian Peek’s website and it is also incorporated into the Bespoke XNA 3DUI Framework.

References:

http://www.brianpeek.com/blog/

http://www.bespokesoftware.org



3D User Interfaces: Design, Implementation, Usability

CHI 2009

▶Programming with the Wiimote

- Connect to computer
  - does not work for every bluetooth device
- Obtain Wiimote software
  - many variations and APIs (C,C++, C#, Java, Flash)
    - Brian Peek's API (www.coding4fun.com)
      - low level API
    - Paul Varcholik's XNA 3DUI Framework (www.bespokesoftware.org)
      - contained within larger framework
      - include gesture recognizer
- Write code and enjoy
  - heuristics
  - gesture analysis

LaViola | Kruijff | Bowman | Poupyrev | Sturzlinger 220

At this point in the lecture, we are going to look at some of the things you need to get started using Wiimotes to create 3D spatial interfaces. The first step is to get the Wiimote(s) connected to the PC. This may sound trivial, but there are subtle nuances that you need to be aware of. The following excerpt comes from Brian Peek's article on connecting Wiimotes to PCs (http://blogs.msdn.com/coding4fun/archive/2007/03/14/1879033.aspx). Note there is also other valuable information in this article as well.

The Wiimote will not pair and communicate successfully with every Bluetooth device and stack on the planet.

1. Start up your Bluetooth software and have it search for a device.

2. Hold down the 1 and 2 buttons on the Wiimote. You should see the LEDs at the bottom start flashing. ***Do not let go of these buttons until this procedure is complete.***

3. The device should show up in the list of devices found as ***Nintendo RVL-CNT-01****. If it's not there, start over and try again.*

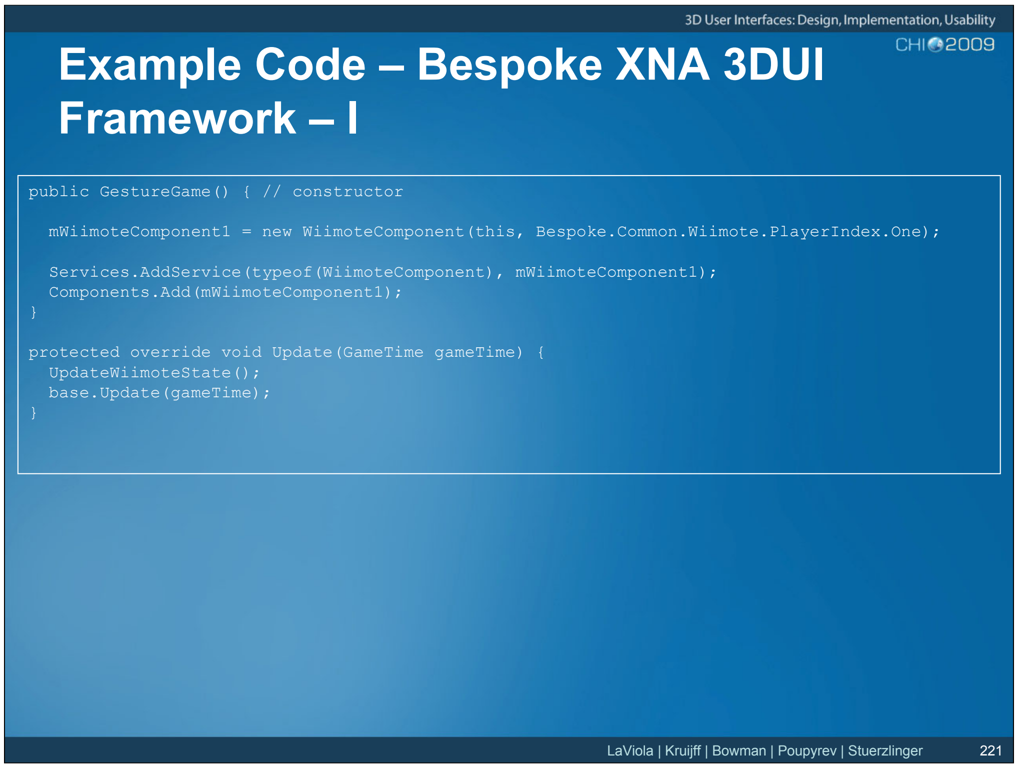
4. Click ***Next*** *to move your way through the wizard. If at any point you are asked to enter a security code or PIN, leave the number blank or click* ***Skip****. Do not enter a number.*

5. You may be asked which service to use from the Wiimote. Select the keyboard/mouse/HID service if prompted (you should only see one service available).

6. Finish the wizard.

That's it. The LEDs at the bottom should continue to flash and you should see the device listed in your list of connected Bluetooth devices. If you run the test application (the Wiimote Tester – a screen shot of the app is shown on the previous slide) included with the source code and you see the numbers change, you are all set. If you don't see them change or you get an error, try the above again. If it continues to not function, you are likely stuck with an incompatible device or stack.

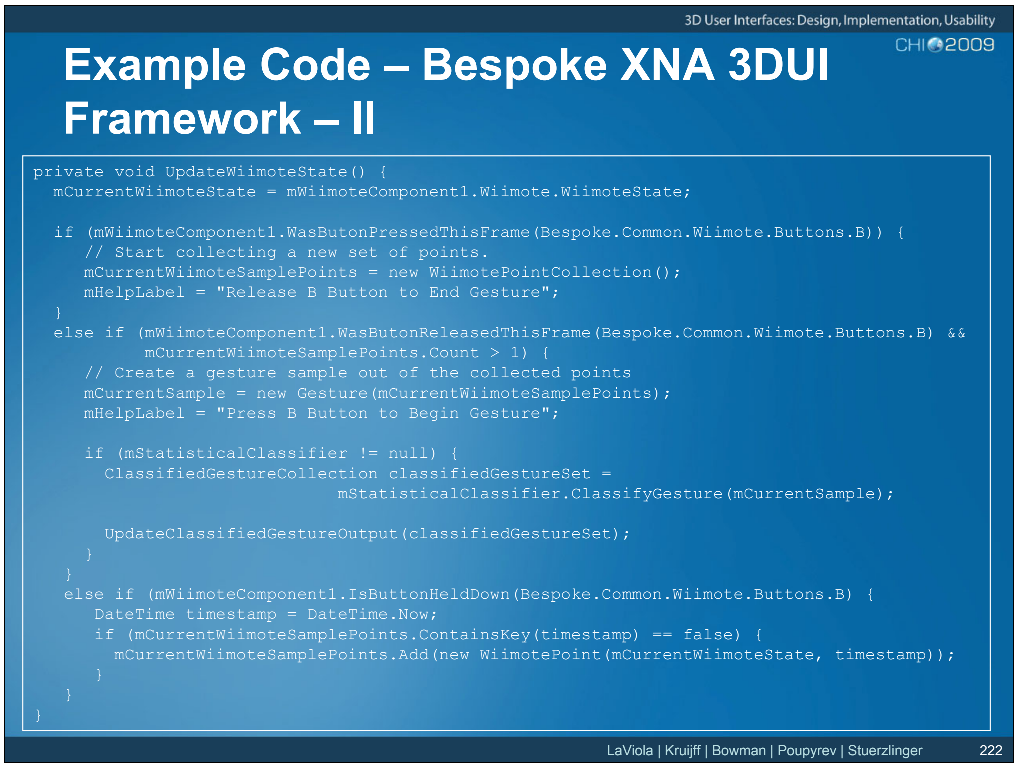
Once you know you have a viable connection with your PC, you can then begin to start using Wiimotes to develop 3D spatial interfaces. The next step in the process is to find an appropriate software API or SDK that you can incorporate into you game application code. There are a variety of different systems out there today. If you are a C# programmer, I recommend Varcholik's Bespoke XNA 3DUI Framework. It has many useful tools, and incorporates Brian Peek's Wiimote library. Note with Bespoke, you can use as many Wiimotes as you want. Bespoke also has a gesture recognition engine which make it easy to train and use gestures. Other Wiimote libraries include WiiCade for Flash developers, GlovePie, and a host of others (simply do a google search for Wiimote and AP)I.



Example Code – Bespoke XNA 3DUI Framework – I

```
public GestureGame() { // constructor

  mWiimoteComponent1 = new WiimoteComponent(this, Bespoke.Common.Wiimote.PlayerIndex.One);

  Services.AddService(typeof(WiimoteComponent), mWiimoteComponent1);
  Components.Add(mWiimoteComponent1);
}

protected override void Update(GameTime gameTime) {
  UpdateWiimoteState();
  base.Update(gameTime);
}
```

The following code snippet is from taken from the gesture test application found in the Bespoke XNA 3DUI Framework. This code shows how to initialize the Wiimote Component. The UpdateWiimoteState function is called in XNA's Update method which gets called once every frame.



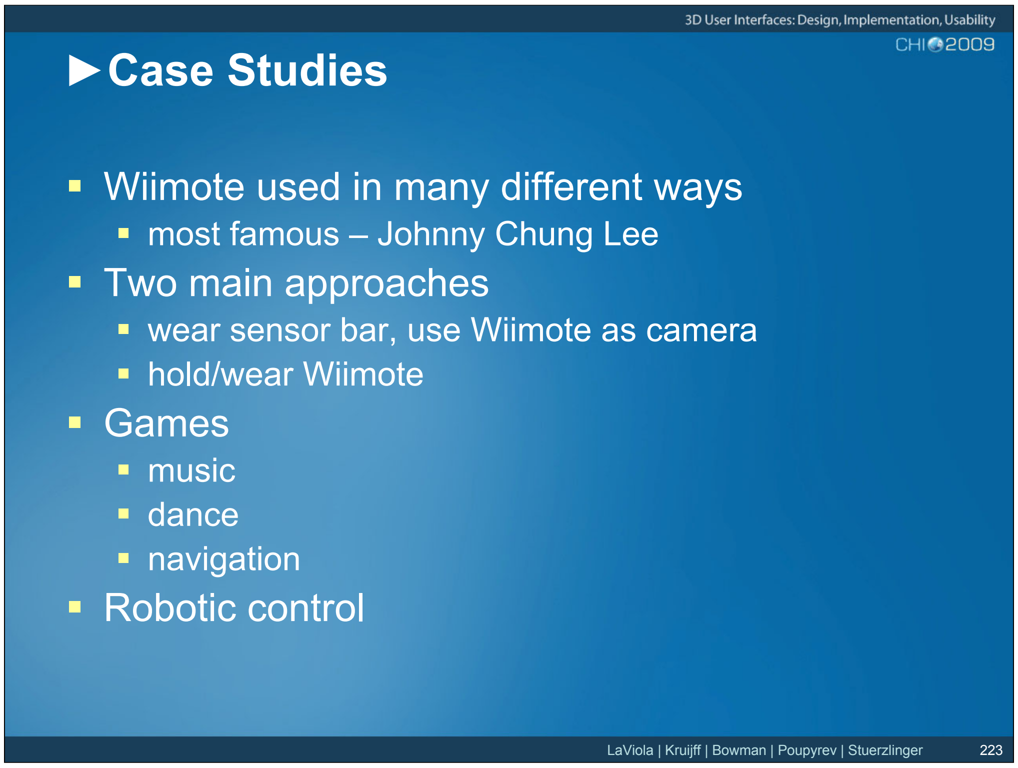
Example Code – Bespoke XNA 3DUI Framework – II

```
private void UpdateWiimoteState() {
  mCurrentWiimoteState = mWiimoteComponent1.Wiimote.WiimoteState;

  if (mWiimoteComponent1.WasButonPressedThisFrame(Bespoke.Common.Wiimote.Buttons.B)) {
     // Start collecting a new set of points.
     mCurrentWiimoteSamplePoints = new WiimotePointCollection();
     mHelpLabel = "Release B Button to End Gesture";
  }
  else if (mWiimoteComponent1.WasButonReleasedThisFrame(Bespoke.Common.Wiimote.Buttons.B) &&
           mCurrentWiimoteSamplePoints.Count > 1) {
     // Create a gesture sample out of the collected points
     mCurrentSample = new Gesture(mCurrentWiimoteSamplePoints);
     mHelpLabel = "Press B Button to Begin Gesture";

     if (mStatisticalClassifier != null) {
       ClassifiedGestureCollection classifiedGestureSet =
                               mStatisticalClassifier.ClassifyGesture(mCurrentSample);

       UpdateClassifiedGestureOutput(classifiedGestureSet);
     }
   }
   else if (mWiimoteComponent1.IsButtonHeldDown(Bespoke.Common.Wiimote.Buttons.B) {
      DateTime timestamp = DateTime.Now;
      if (mCurrentWiimoteSamplePoints.ContainsKey(timestamp) == false) {
        mCurrentWiimoteSamplePoints.Add(new WiimotePoint(mCurrentWiimoteState, timestamp));
      }
   }
}
```

This code snippet show what is contained in the UpdateWiimoteState function. In this case, the function is collecting Wiimote motion data as long as the B button is pressed. When it is released, the data is sent to a Bespoke's gesture classifier for processing.

You can download the Bespoke XNA 3DUI framework at

http://www.eecs.ucf.edu/isuelab/downloads.php?theme=4



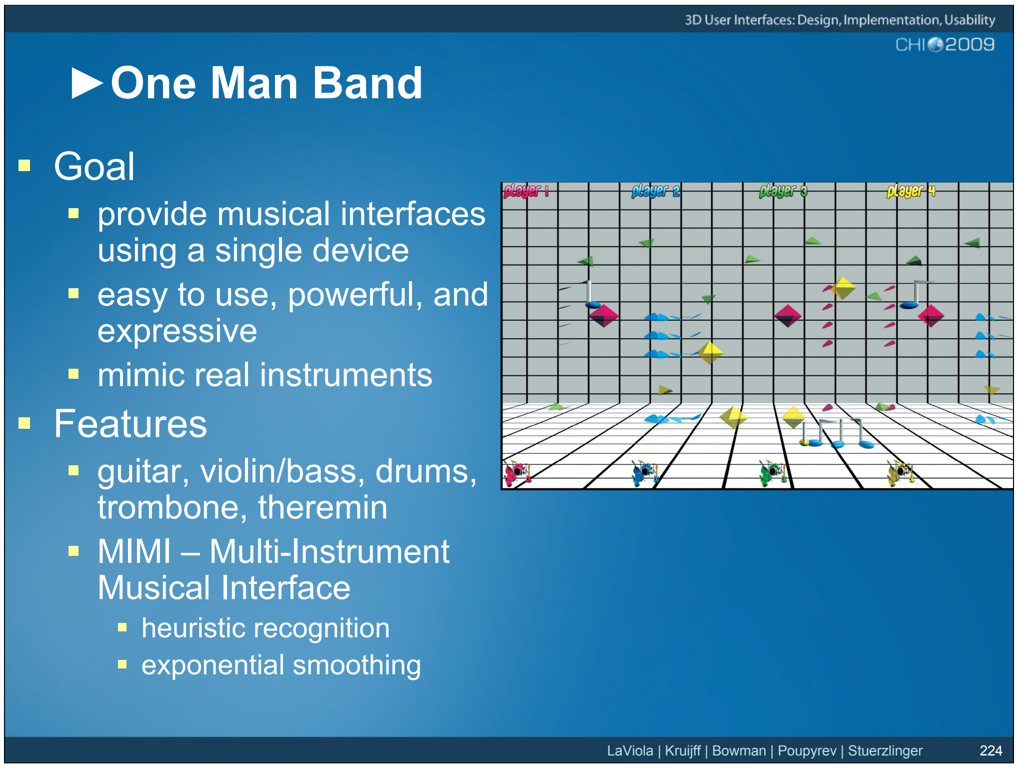
▶Case Studies

- Wiimote used in many different ways
  - most famous – Johnny Chung Lee
- Two main approaches
  - wear sensor bar, use Wiimote as camera
  - hold/wear Wiimote
- Games
  - music
  - dance
  - navigation
- Robotic control

The Wiimote has been used in a variety of different ways. Of course, Johnny Chung Lee has done a variety of interesting projects and is probably the most famous Wiimote “hacker”. However, there are many other researchers who are using Wiimotes for research in spatial 3D interfaces. Typically, there are two main approaches people use. The first is to actually wear the sensor bar and mount the Wiimote in some stationary position, using it as a camera. The second approach is to simply hold the Wiimote or attach it to the body. This approach makes more use of the accelerometer data than the image sensor data. The following slides illustrated some interesting research projects that use the Wiimote in both computer games and robotics.

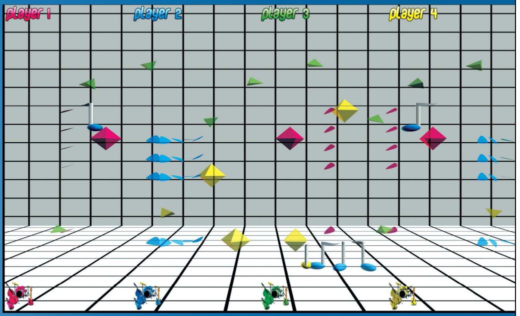
References:

http://www.wiimoteproject.com/index.php

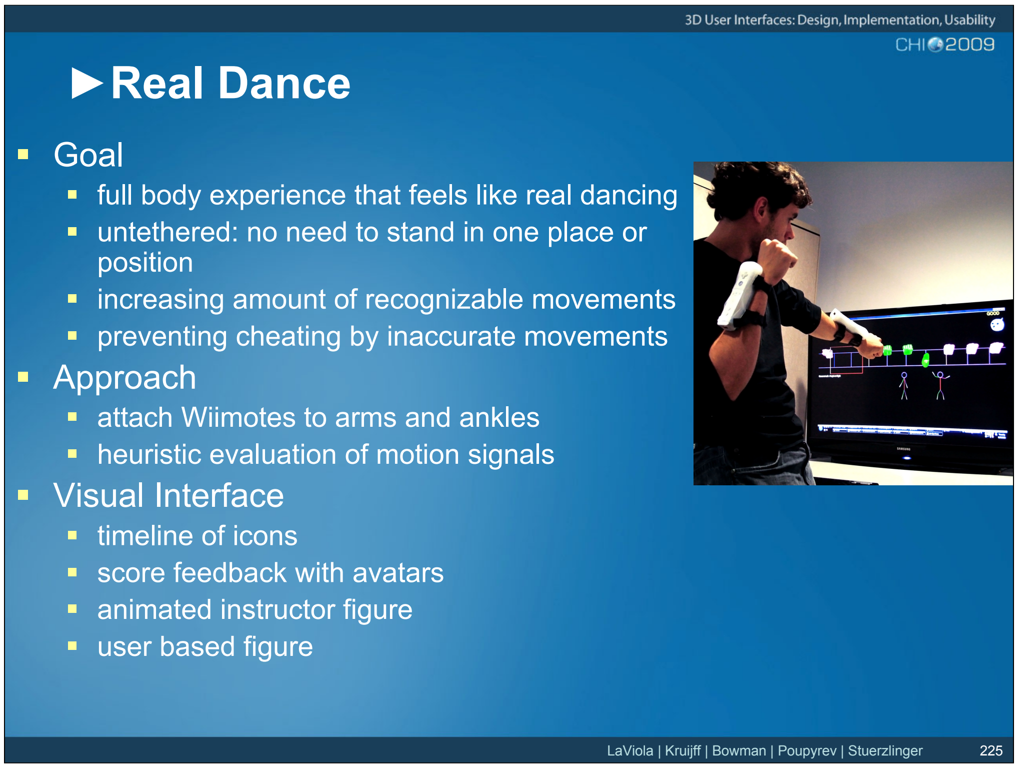


►One Man Band

- Goal
  - provide musical interfaces using a single device
  - easy to use, powerful, and expressive
  - mimic real instruments
- Features
  - guitar, violin/bass, drums, trombone, theremin
  - MIMI – Multi-Instrument Musical Interface
    - heuristic recognition
    - exponential smoothing

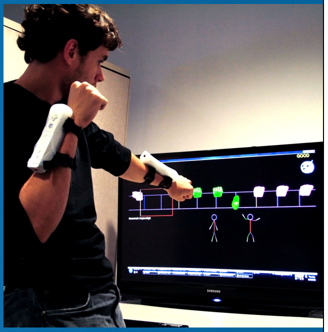


One Man Band's goal is to explore how a single Wiimote device can be used to create music. It was developed at the Interactive Systems and User Experience Lab at the University of Central Florida. The current system supports the guitar, violin, bass, drums, trombone, and therein. One Man Band also has a multi-instrument musical interface called the MIMI. The idea behind the MIMI is that game players might want to quickly and easily transition from one musical instrument to another without any mode switching. The MIMI uses heuristics recognition and exponential smoothing to detect 5 different instruments. A study was recently conducted comparing One Man Band and Wii Music and the results showed users significantly preferred One Man Band.

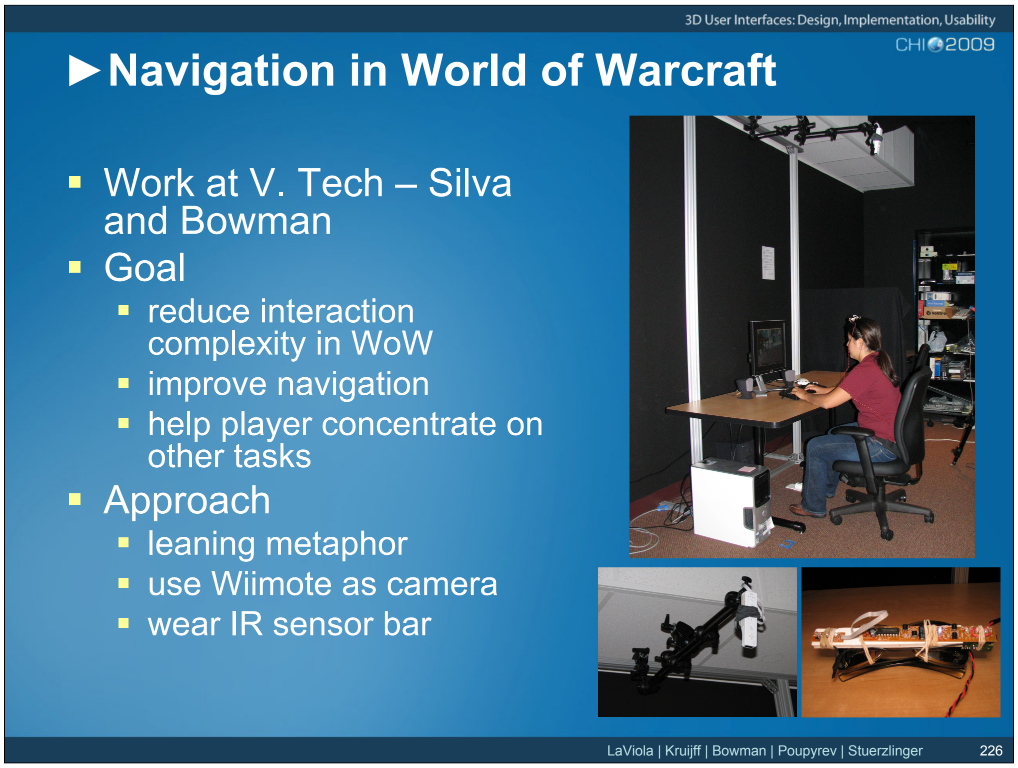


▶Real Dance

- Goal
  - full body experience that feels like real dancing
  - untethered: no need to stand in one place or position
  - increasing amount of recognizable movements
  - preventing cheating by inaccurate movements
- Approach
  - attach Wiimotes to arms and ankles
  - heuristic evaluation of motion signals
- Visual Interface
  - timeline of icons
  - score feedback with avatars
  - animated instructor figure
  - user based figure

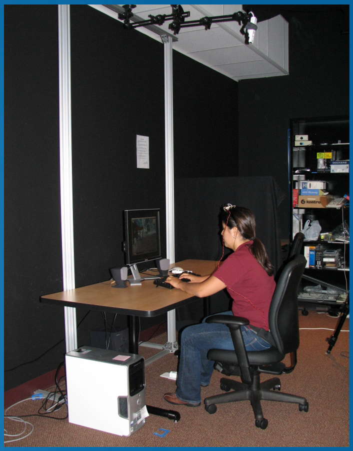


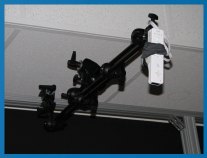
Real Dance is a game prototype, developed at the Interactive Systems and User Experience Lab at the University of Central Florida, that is exploring more natural, full body interfaces for dancing games. In the game, users wear four Wiimotes attached to their wrists and ankles using velcro strips. This provides an untethered experience, meaning that the user does not need to stand in one place or position. Another goal of Real Dance is to explore how to increase the number of recognizable dance movements. The current prototype detects kicks, stomps, punches, and static poses. It also employs a visual interface for teaching users to perform the various dances. The visual interface includes a timeline of icons, score feedback with avatars, and an animated instructor figure.

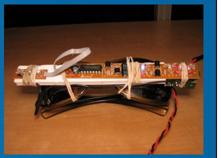


►Navigation in World of Warcraft

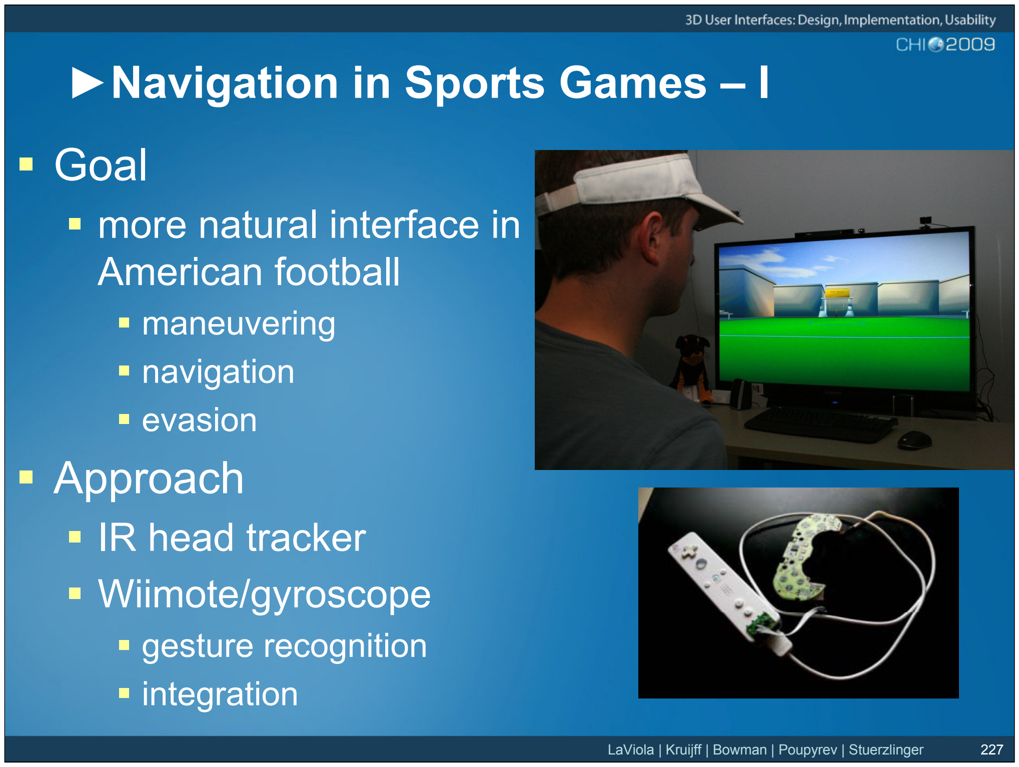
- Work at V. Tech – Silva and Bowman
- Goal
  - reduce interaction complexity in WoW
  - improve navigation
  - help player concentrate on other tasks
- Approach
  - leaning metaphor
  - use Wiimote as camera
  - wear IR sensor bar





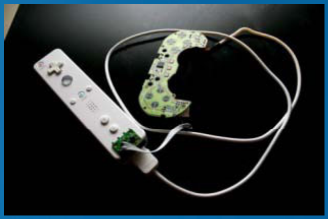


Mara Silva and Doug Bowman at Virginia Tech are exploring how body-based interaction can be used to complement and reduce interaction complexity in World of Warcraft. The idea behind this work is to using body based controls to offload keyboard and mouse navigation, helping players to concentrate on other tasks. In their configuration, the user wears a modified sensor bar and mount a Wiimote on the ceiling, using the device as a camera. Navigation is based on a leaning metaphor. Starting from a neutral body position, a small amount of forward or backward movement makes the character walk, and leaning farther forward makes the character run (rate control). There is a “dead zone” surrounding the neutral point in which the character stands still. Leaning to the side rotates the character, with the amount of rotation proportional to the distance the player leaned (position control). Note that a foot pedal is used to activate and deactivate movement. Preliminary experimentation has shown that body-based interaction in addition to keyboard and mouse can help players perform more tasks at the same time and can be especially attractive and helpful to new players.



►Navigation in Sports Games – I

- Goal
  - more natural interface in American football
    - maneuvering
    - navigation
    - evasion
- Approach
  - IR head tracker
  - Wiimote/gyroscope
    - gesture recognition
    - integration



Current user interfaces for sports games can be made more natural and intuitive to a user by using natural motions. The goal of this project is to examine the hardware and software needed to accomplish three main tasks in a game of American football.

-- Maneuvering – The quarterback needs real time responses to maneuver in a small area of opportunity.

-- Navigation – The ball carrier needs to be able to run down the field

-- Evasion – The ball carrier also needs to recognize some simple gestures that can be performed in order to evade tackles.

Three techniques were developed. The first technique utilized an IR head tracker from Natural Point and was designed as the ideal natural interface in order to compare against Wiimote-based solutions. The second technique used a single Wiimote with an attached gyroscope, places on the user's chest. Software was used to recognize patterns in the data to interpret movement on the field. The results from this technique shows:

-- Maneuvering is possible, but had a great deal of latency and unintended movements.

-- Running down the field is possible with a Running-In-Place metaphor.

-- Not enough data was present from the hardware to accurately perform the gestures.

The third technique uses the Wiimote configuration, but integrated the accelerometer data to maneuver on the field.

-- Maneuvering was cleaner and in more real time, though still not as good a with the head tracker and drift was present.

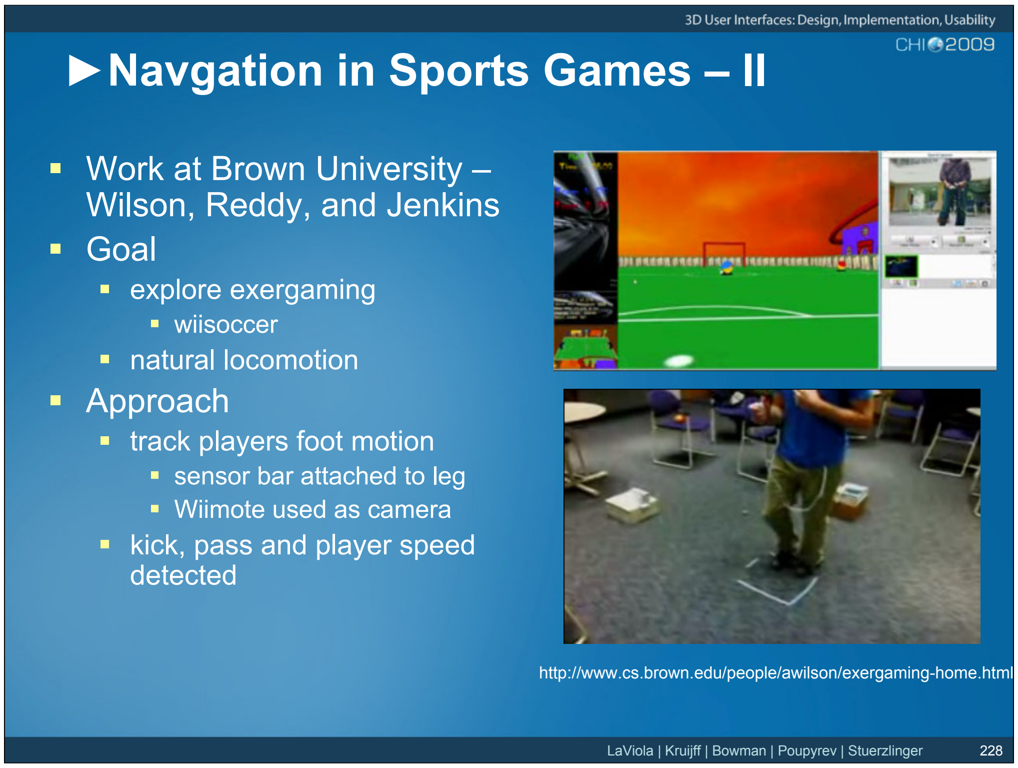
-- Running down the field was accomplished.

-- Not enough data was present from the hardware to accurately perform the gestures.

Maneuvering is a hard task without some clue to position, as the head tracker can provide. With the third technique it still seems possible though.

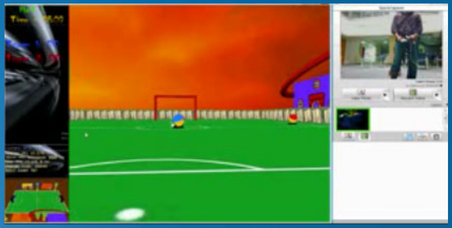
Gesture recognition is hindered when the hardware data is already being utilized for some other process (such as whether the user is running in place or not).

This project is still in its early stages at the Interactive Systems and User Experience Lab at UCF.



▶Navgation in Sports Games – II

- Work at Brown University – Wilson, Reddy, and Jenkins
- Goal
  - explore exergaming
    - wiisoccer
  - natural locomotion
- Approach
  - track players foot motion
    - sensor bar attached to leg
    - Wiimote used as camera
  - kick, pass and player speed detected

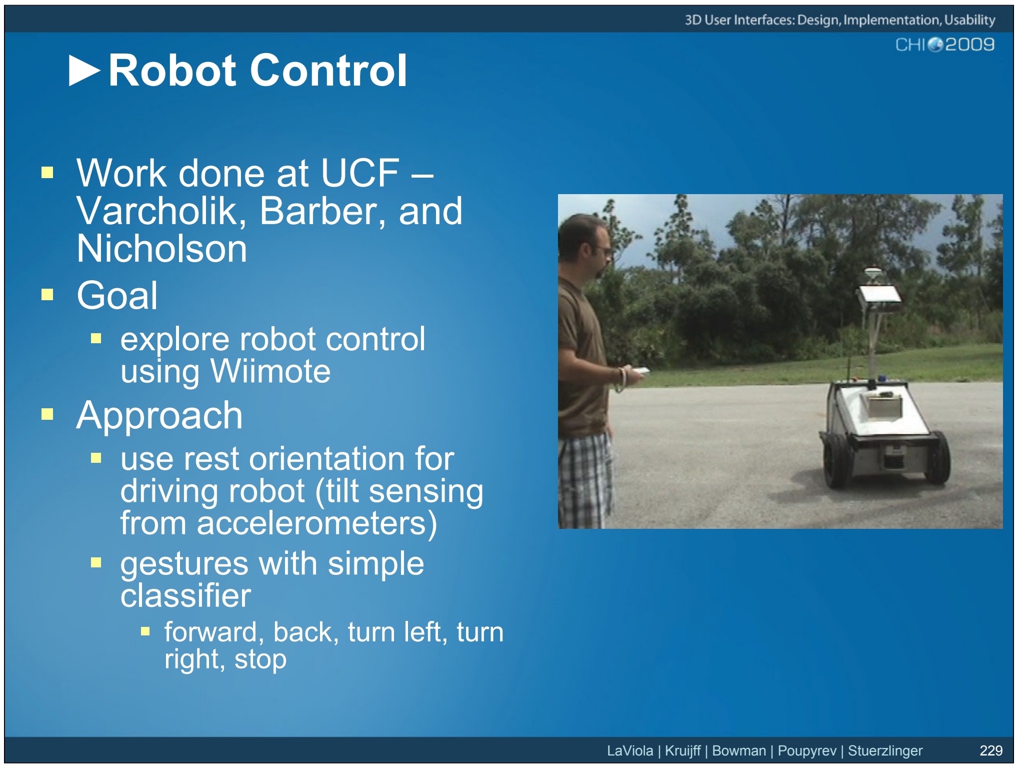


http://www.cs.brown.edu/people/awilson/exergaming-home.html

The focus of this project, developed at Brown University under the direction of Chad Jenkins, is on exergaming, an important and up-and-coming research area that examines how to build effective exercise-based games. Wiisoccer is a game that uses natural locomotion to move players on a soccer field. The key innovation is that the IR sensor bar is attached to the user's leg and a Wiimote is used as a camera and placed on the side of the player. The Wiimote detects the player's running motion as well as kicks and passes.

References:

http://www.cs.brown.edu/~awilson/exergaming-home.html



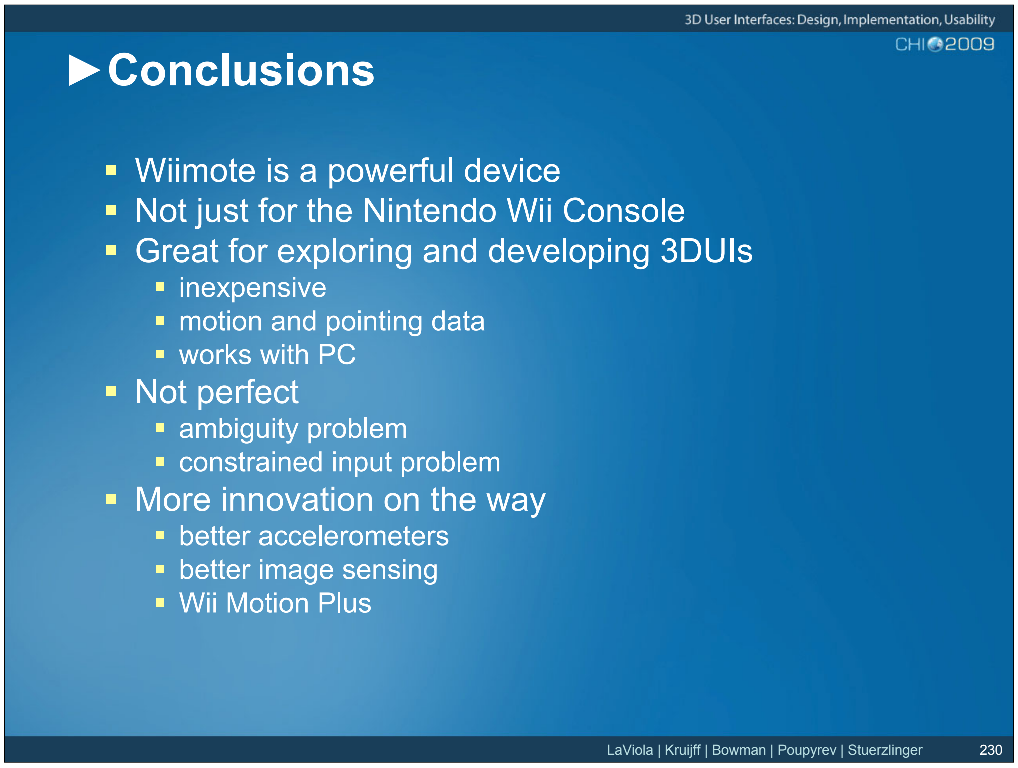
▶Robot Control

- Work done at UCF – Varcholik, Barber, and Nicholson
- Goal
  - explore robot control using Wiimote
- Approach
  - use rest orientation for driving robot (tilt sensing from accelerometers)
  - gestures with simple classifier
    - forward, back, turn left, turn right, stop

In our final case study, a Wiimote is used to control a robotic vehicle. This work was done by Paul Varcholik, Daniel Barber, and Denise Nicholson at the Institute for Simulation and Training at UCF. In their approach they use an interface similar to Mario Kart. They make use of a resting Wiimote to calculate roll and pitch for moving the robot forward and backward and for turning left and right. In another control scheme, they use Wiimote gestures to control the robot. The project show an example of how a Wiimote can be used in a mobile setting.

References:

Varcholik, P., Barber, D.and Nicholson (2008). Interactions and Training with Unmanned Systems and the Nintendo Wiimote. *Proceedings of the 2008 Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*.



▶Conclusions

- Wiimote is a powerful device
- Not just for the Nintendo Wii Console
- Great for exploring and developing 3DUIs
  - inexpensive
  - motion and pointing data
  - works with PC
- Not perfect
  - ambiguity problem
  - constrained input problem
- More innovation on the way
  - better accelerometers
  - better image sensing
  - Wii Motion Plus

In conclusion, the Wiimote is a powerful input device that is not just for playing games with the Nintendo Wii console. It is a great device for exploring 3D spatial interfaces because it is inexpensive compared to other tracking systems, it works with a PC, and provides motion and pointing data. Although the device is not perfect because of the ambiguity and constrained input problem, it can still be used in a variety of different contexts. The Wiimote continues to evolve and Nintendo is planning on improving the device as a whole with better accelerometers and image sensing hardware. In addition, the Wii Motion Plus should make a significant impact on the device's quality and expand its interface capabilities. Of course, there is still a lot of research to be done in finding the best ways of using the Wiimote in 3D spatial interfaces for games and other applications.

For more information, the following websites can be useful:

http://wiibrew.org/wiki/Main_Page

http://www.wiili.org/