• Goals

• Create a multi-user god game which uses a novel combination of finger and hand-stroke gestures

• Create a testbed kiosk where multiple users can interact with the system without need of training or formal introduction
• Assumptions

• Gross body gestures are being adequately explored by modern consoles as an alternative to a traditional controller

• Physical proximity of users can enhance gameplay experience

• Most interactions with games involves physical disconnect between user and image

• Casual games have largest growth potential

• Gesture computing (base) technology has reached sufficient maturity for rapid prototyping.
Introduction

• Related Work (Gesture-Based Computing)
  • Sutherland’s Sketchpad (1963)
  • Jones and Katyl’s GRAPHPAK (1974)
  • Minsky’s gesture recognition system (1984)
  • Hong and Landay’s SATIN (2000)
  • Benko, Wilson, Baudisch’s Stretch, X-Menu, and Slider techniques (2006)
  • Microsoft’s Tablet SDK (2007)
• Related Work (Multi-point Touch Systems)

  • Krueger’s VIDEOPLACE (1991)
  • Wellner’s Digital Desk (1993)
  • Dietz and Leigh’s DiamondTouch (2001)
  • Rekimoto’s SmartSkin (2002)
  • Han’s Low-Cost FTIR (2005)
  • Morris et. al.’s Cooperative Gestures (2006)
• Related Work (Gesture-based Gaming)
  • Lost Magic (Ubisoft/Taito Corp.)
    • Draw symbols to cast spells
  • WarioWare: Smooth Moves (Nintendo)
    • “The Umbrella”: player holds WiiMote like an umbrella handle
    • “The Elephant”: player holds WiiMote against nose, like an elephant’s trunk
    • “Darts”: player simulates action of throwing a dart with the WiiMote
• Multi-touch Display
  • Leverages:
    • open source software libraries (OpenCV)
    • standards-based 3D protocols (X3D)
    • Simple, low-cost hardware
    • FTIR technology
      • Based on Han’s research (2005)
Hardware

• Multi-touch Display
  • 22” x 35”
  • LEDs: 850 nm
    • parallel series of banks of five
    • 1.03 A/9.5V
  • PixeLINK 1394 camera
    • 1024 X 768 monochrome uncompressed
  • NEC WT610E Projector
    • 2000 lumens
    • 1024 x 768 native resolution
• Server
  • Intel Core 2 Duo 2.66 GHz
  • 4 GB RAM
  • PCI Express
  • Nvidia 8800 GTX
• Multi-touch Display

Diagram courtesy Drexel infinitouch
System Architecture

**Touchscreen**
- Point data from multiple blobs (individual fingers) as CBlocks

**C++**
- MTD Library
  - Condense point groups into hands
  - Classify gesture
  - Send over socket

**X3D Engine**
- MTClient
  - Retrieve from socket, parse, & store to local buffer
  - Hand data (4 byte buffer): action, hand, x, y, z (finger, gesture)
  - Did any hand hit a diggum?
    - Yes: send notification of gesture update (front)
    - No

**ECMAscript**
- “Brain”

- Condensed multiple input points (blobs) to a single input point and determines a particular hand gesture if a finger gesture is recognized
- Takes input off the socket and parses data for ECMAscript
- Controls diggum state machine

---

Digital Media
College of Media Arts and Design
Drexel University
Server (C++)

- Implementation of MTD library developed by Drexel’s infinitouch team
  - Based on OpenCV
    - Bradski, 2002
  - Retrieves **blob** information from table using cvblobslib
    - Based on Dave Grossman’s algorithm
    - Extract blob components from binary/grayscale image
    - Filter to get objects of interest
System Architecture

```c
CBlobResult blobs;
blobs = CBlobResult( inputImage, NULL, 100, true );
blobs.Filter( blobs, B_INCLUDE, CBlobGetArea(),
              B_GREATER, 5000 );
CBlob blobWithBiggestPerimeter, CBlob blobWithLessArea;
blobs.GetNthBlob( CBlobGetPerimeter(), 0,
                  blobWithBiggestPerimeter );
blobs.GetNthBlob( CBlobGetArea(), blobs.GetNumBlobs() -
                  1, blobWithLessArea );
IplImage *outputImage;
outputImage = cvCreateImage( cvSize( inputImage->width,
                                    inputImage->height ), IPL_DEPTH_8U, 3 );
cvMerge( inputImage, inputImage, inputImage, NULL,
         outputImage );
blobWithBiggestPerimeter.FillBlob( outputImage, CV_RGB(255, 0, 0 ));
blobWithLessArea.FillBlob( outputImage, CV_RGB(0, 255, 0 ));
```

Source: http://opencvlibrary.sourceforge.net/
System Architecture

**Server (C++)**

- Initialize tracker
- Initialize output handler
- Capture single frame

```
while (blobs) {
    Determine blob state (add, update, delete)
    On add: add to hand or create new hand and determine gesture
    On update: update position/gesture
    On delete: remove from current point list
}
```

Send hand data (action, id, centerX, centerY, area, numFingers, gesture) over socket
X3D Engine

- Custom Nodes added via DLLs (C++)
  - Interface between server and X3D world file
  - Custom MTClient node
  - Modified PlaneSensor/TouchSensor
System Architecture

X3D Engine

- Retrieve hand data from socket
- Write hand data to local buffer
- Upon request, retrieve hand data from local buffer
  
  if (diggumHit)
  
  Send gesture from local buffer to ECMAScript
ECMAscript

- Controls “brain” logic of diggums
- If the diggum is hit, retrieve current gesture from custom nodes and change diggum state accordingly
• How diggums think (“the brain”)
  • Each has a “temperament”
    • Six different variables (bored, scared, religious, happy, hungry, tired) on a (-1,1) scale
    • Different combinations of these “temperament” variables, as well as possible user input, location (water, etc.), and a little randomization result in 15 different states
The Diggum
The Diggum

Swimming

- User Gesture
- User Picks Up
- Picked Up
- Increases Fear
- Searches for Land
- Finds Land
- Doesn't Find Land
- Reverts to Idle

Not On Land

Digital Media
College of Media Arts and Design
Drexel University
Gestures

• Poke
• Grab ‘n’ Drag
• Slingshot
• Zoom in/zoom out
• Five-finger drag navigation
Future Work

• Gross gesture recognition
  • Experimentation needed to determine whether SATIN will be suitable for our needs
    • Downsides: heavily reliant on Java2D and Swing, currently does not natively support multiple users
    • Alternatives: Microsoft Tablet SDK, in-house HMM library by Louis Kratz
  • Recording and analysis of user input to refine recognition
  • Address tracking latency
<table>
<thead>
<tr>
<th>Digital Media</th>
<th>Computer Science</th>
<th>Electrical Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Paul Diefenbach</td>
<td>Dr. Frank Lee</td>
<td>Dr. Youngmoo Kim</td>
</tr>
<tr>
<td>William Muto</td>
<td>Louis Kratz</td>
<td>Timothy Kurzweg</td>
</tr>
<tr>
<td>Matthew Smith</td>
<td>Ko Nishino</td>
<td>Vijay Balchandani</td>
</tr>
<tr>
<td>Chester Cunanan</td>
<td>Craig Polakoff</td>
<td>Eric Effinger</td>
</tr>
<tr>
<td>Justin Dobies</td>
<td></td>
<td>Jeevan Kotha</td>
</tr>
<tr>
<td>Arvind Neelakantan</td>
<td></td>
<td>Pannha Prak</td>
</tr>
<tr>
<td>Sara Colucci</td>
<td></td>
<td>Joseph Romeo</td>
</tr>
<tr>
<td>James Grow</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Drexel infinitouch**

<table>
<thead>
<tr>
<th>Boris Block</th>
<th>Dan Hennessey</th>
<th>Zenko Klapko</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dan Hennessey</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Zenko Klapko</td>
<td></td>
</tr>
<tr>
<td></td>
<td>David Millar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>William Morgan</td>
<td></td>
</tr>
</tbody>
</table>