

Virtual Environments



Shyam M Guthikonda
20 April 06

Background

- Who am I?
 - Year: Senior
 - Major: Computer Science
 - Minors: Physics, Math, Computational Science

Purpose of this Talk

- Explain some of the things I've been working on for the past year @ the Ohio Supercomputer Center (OSC) in Columbus
- Demonstrate some of the projects going on at OSC
- Thanks
 - Dr. Noyes, Kevin, Don, Dennis

Goals of this Talk:

- Theoretical:
 - Familiarize audience with the concept of VEs and demonstrate their importance
- Practical (implementation):
 - Introduce various hardware/software requirements for VEs in order to show how these projects can be implemented

What are Virtual Environments?

- Popularly known as “Virtual Reality”
 - VR has become a sort of 'layman' term to describe a broad range of virtual simulations
 - Don has a pet peeve towards the term “Virtual Reality”
 - Therefore: we use “Virtual Environment”



What does “VE” mean?

- Virtual – created; simulated; carried out by a computer
- Environment – The circumstances or conditions that surround one; surroundings
- Combining the two:
 - A computer simulation that places the user in a real or imaginary surrounding that enables the user to perform operations on the simulated system and shows the effects in **real time**.

Definitions from: (opera.answers.com)

Why are VEs Important?

- We can place a person in *any* environment (real *or* imaginary)
 - (usually) without any harm to the person
 - Extremely expensive or impossible to send the person to the environments (Mars, inside the human body, etc)
 - Endless possibilities

Objectives of VEs

- To better train individuals, preventing real life injuries and/or deaths
- To help find the most cost-effective best-practice information regarding the environment
- Identify possible high-risk individuals

Taken from: Stredney, "Simulation of a PTO Environment".

Types of VEs

- Agriculture
 - Tractor/Power-Take-Off (PTO) Safety
- Architecture
 - 3D Visualization of Designs
- Medicine
 - Pre-Operative Care Planes
- Military
 - Battlefield strategies and tactics
- Any type of simulation is possible

How do we Create VEs?

- Equipment (Hardware Requirements)
 - Intersense – tracking devices
 - Immersion – force-feedback devices
 - Very expensive!
 - Tracking System:
 - A grid (on the ceiling)
 - Acoustic (position) and Inertia (orientation) based sensors
 - Allow for ~ few mm precision

Equipment (cont.)

- Head-Mounted Display (HMD)
 - Perhaps the most important piece of equipment
 - 2 extremely high resolution screens
 - Can be rendered in quad-buffer stereo
 - Allows the user to 'see' the virtual world



Equipment (cont.)

- Interaction devices
 - Need a way to interact with the environment (no hands!)
 - Wand:
 - 3D mouse
 - Laser-pointer for selection of objects
 - Buttons for other operations
 - Gloves:
 - Virtual pair of hands
 - “Vibrating” devices on each finger
 - Pain or force-feedback



Equipment (cont.)

- Misc. Devices
 - Headphones
- Attempt to bring every one of the senses into the Virtual Environment
 - O Sight
 - O Touch
 - O Hearing
 - X Taste
 - X Smell

Equipment (cont.)

- Force-feedback – preventing the user from performing a certain physical action
 - Very difficult to do completely
 - Can't injure the user
 - Expensive
- Interesting point:
 - All this VE software can be run on a standard desktop PC

Software Requirements

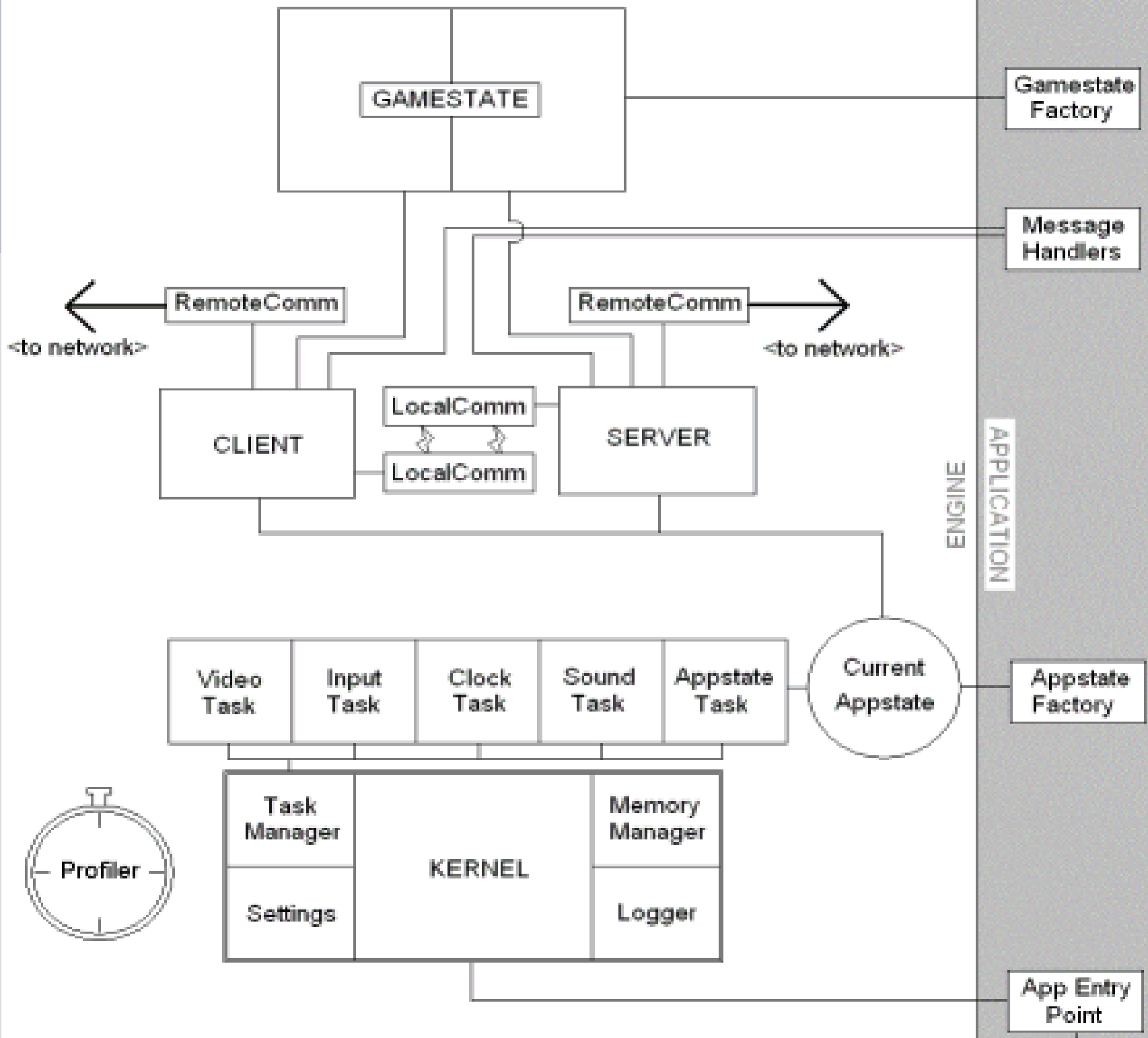
- Real-Time
 - Movies vs Video Games
 - Video Game = Virtual Environment = Real-Life
 - Must always be considered when dealing with software for both video games and VEs
 - A slow algorithm must be re-done
 - Sometimes quality must be sacrificed for performance

Utilizing Existing Technologies

- Basic rule: “Don't reinvent the wheel.”
- Existing technology is probably more stable due to multiple developers/users over a long period of time
 - i.e. Writing your own linked-list vs STL

Engine

- Quake Engine, Source Engine (Half-Life 2), Grand-Theft-Auto Engine, etc.
- Engine performs 'fundamental' tasks
 - Rendering (drawing to the screen)
 - Audio
 - Physics
 - Animation
 - Input
 - Networking
 - etc.
- Application sits on top of engine



Engine

- Delta3D Gaming and Simulation Engine
 - Main reason:
 - Open-Source (we have all the code)
 - Other reasons:
 - OpenSceneGraph (OSG)
 - Open Dynamics Engine (ODE)
 - Cal3D
 - OpenAL
 - Python bindings
 - XML
 - Networking
 - STAGE
 - Combines all features into single API

veLib

- Project got big
 - ~6000 lines of code
 - 44 classes/structs
 - 43 files
 - Greater complexity/functionality
- Create a VE library layer
 - Future distribution of library

veLib

- Layers (figure) – application on top
- 3 main applications:
 - Main Demo
 - The original project, but using Delta3D + new features
 - Physics Demo
 - Shows off physics capabilities
 - Power-Take-Off (PTO)
 - Agricultural Safety
 - Central focus (most of the funding)

The PTO Environment

- Meets 2 VE objectives:
 - “To better train individuals, preventing real life injuries and/or deaths”
 - “Identify possible high-risk individuals”
- Significance
 - “Machine related incidents remain one of the most common causes of death in the agricultural workplace... with unintentional injuries among persons less than 20 years of age for the period 1982-1996, 35.6% of the 2174 deaths were from machinery, and overall, the 3rd leading cause of death” (Stredney, *Simulation of a PTO Environment*)
 - Summary: 35.6% of 2174 = 774 avoidable deaths

The PTO Environment

- Equipment used:
 - HMD
 - Headphones
 - Gloves (x2)
 - Vital statistics – heart rate, temperature, blood pressure, etc.
- Scenario Explanation
 - Farm setting: tractor + PTO
 - Perform various tasks
- If hand comes into contact with running PTO
 - Gloves vibrate to signal loss-of-finger(s)
 - and...

The PTO Environment



The PTO Environment

- Contributions
 - Sound Manager (replaced by Delta3D)
 - Ambient (birds)
 - Transition Sounds (engine)
 - Triggers/Events
 - Proximity – user position
 - Timed – timer
 - Radial – object selection

Video(s)

- PTO Scenario

Future Plans

- Release library as Open Source
- Networking between OSC-C and OSC-S VEs
 - Interaction between multiple users
 - Creates even more possibilities for simulations

Summary

- We covered...
 - Purpose/Objectives of VEs
 - Applications of VEs
 - Necessary Equipment
 - Software Requirements of VEs
 - Specifics pertaining to the PTO environment

¿Questions?