

# Chapter 1: The Challenges of Networked Games

Magda El Zarki

Dept. of CS

UC Irvine

[elzarki@uci.edu](mailto:elzarki@uci.edu)

<http://www.ics.uci.edu/~magda>

# Networked Virtual Environments (NVE)/Networked Games (NG)

- There are many genres of games
  - FPS
  - RPG
  - Simulations
  - Sports
  - RTS
- Each genre has different **system requirements**

# Common Themes in Games

- 3D virtual environment
- Real-time changes
- Collaboration with other users
  - Representation of users in the world (typically as avatars)
  - Text communication
  - Voice
- Virtual environment might mirror a real place but typically is a fantasy place

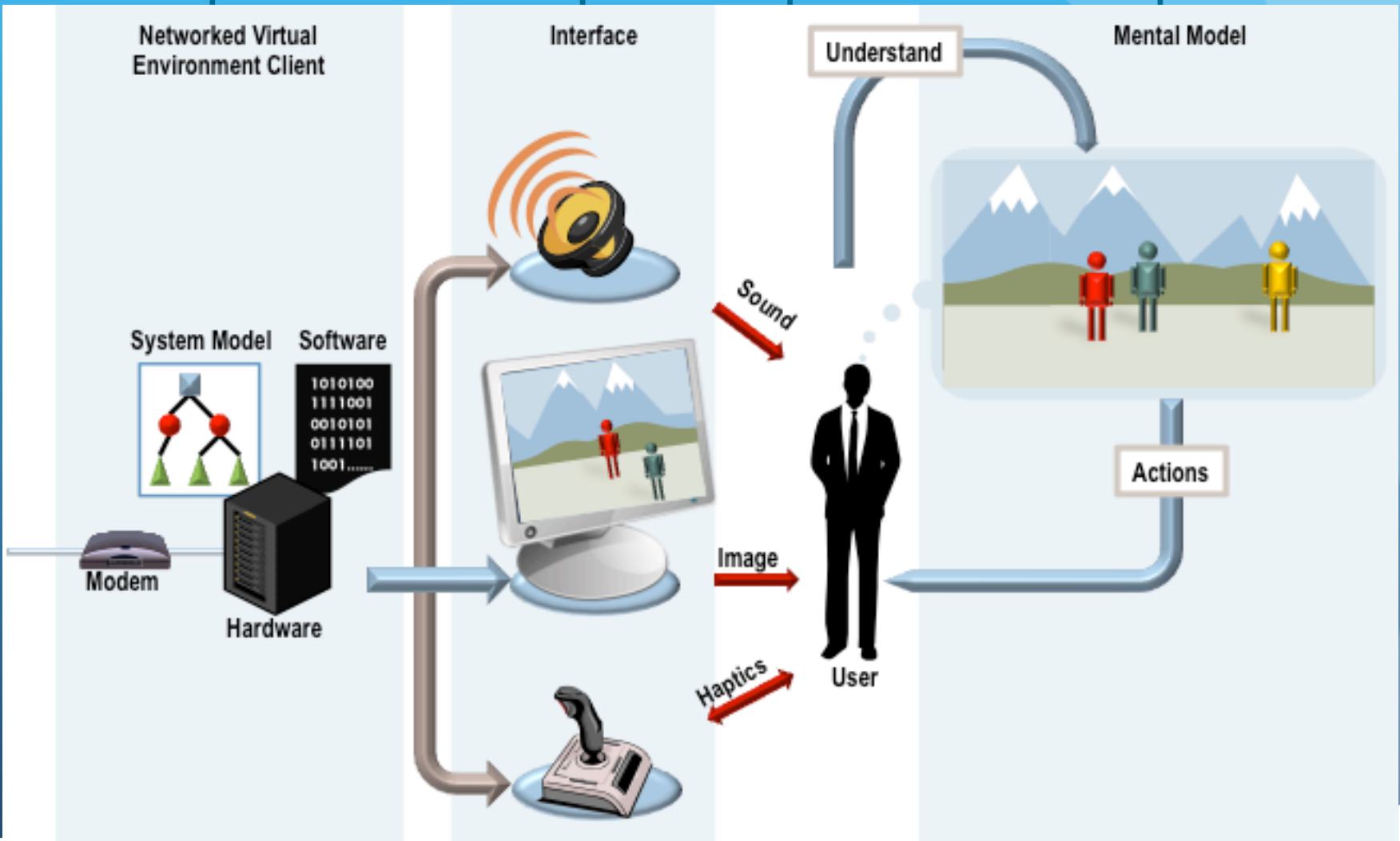
# Common Themes

- One *client* is usually responsible for generating the view for *one user*
- A *set of clients* creates the *illusion* of a shared space
- “Illusion” because
  - Virtual environments can involve detailed *models*
  - Information about changes in models takes time to travel across communication links

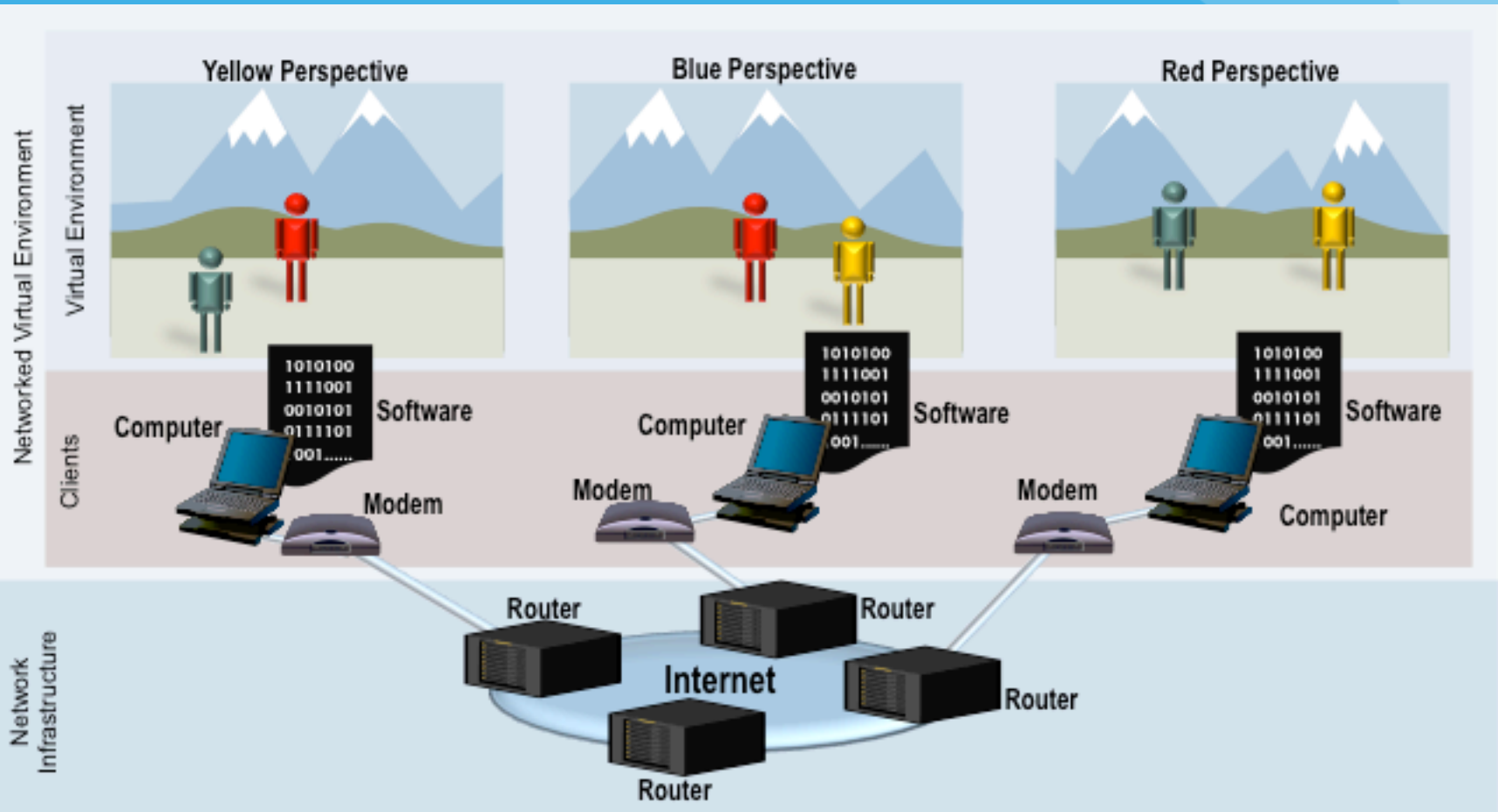
# Virtual Environment: Client and User

Client

User



# Networked Virtual Environment



# Consistency and Plausibility

- *Local plausibility* is the appearance of consistency of only local actions
- *Shared plausibility* is the appearance of properties being the same as observed by users
  - *Objects that are in the background need not be consistent*

# Definition of MultiPlayer Online Game (MPOG)

- By definition an online/network game **must involve a network** - a digital connection - that connects two or more computers
- **Multiplayer games are not necessarily networked games**. Many early multiplayer games were hosted on one computer with players taking turns to make moves on a shared or split screen

And.....

- **Not all networked games are multiplayer**. A user could be playing a game that is on a remote server without engaging any other players
- Strictly speaking it is a **software system that allows multiple users to interact with each other in real-time from different locations, usually remote, and preferably with immersive graphics**



**Large, distributed, real-time, interactive system**



# MPOG by Definition MUST

**Have a Network and Involve Multiple players**



# Characteristics of MPOGs

- A shared *sense* of space
- A shared *sense* of presence
- A shared *sense* of time
- *Communication channels*
- Ability to *manipulate* the environment



# Why NVEs **are not** Standard Network Applications

- Unlike video/audio streaming, or web browsing, in an NVE or NG client, networking is **NOT** the main activity: rendering probably is
- Some information changes very quickly and smoothly
  - E.G. player positions
- Can incorporate other web-enabled media
  - Audio/video
- Often require bulk download of assets

# Components of MPOG Systems

- **Graphics Engines and Displays, Physical Engine**
  - Real time rendering, collision detection, opaqueness
- **Control and Communication Devices**
  - Input devices for game control - mouse, keyboard, joy stick, etc.
  - Input devices for communication - keyboard, microphone, camera, etc
- **Processing Systems**
  - System Architecture - Client - Server, P2P, Hybrid, Mobile
  - Realtime distributed system - shared dynamic space
- **Data Network**
  - Wireless vs Wireline
  - Last mile



# Challenges of MPOG Design & Development

- Data Network
- Heterogeneity
- Distributed Interaction
- Real-time system design and resource management
- Failure Management
- Scalability - System Architecture



# Data Network - Bandwidth

- **More users** means **more information** that needs to be communicated
- **Higher end/complex devices** means **more information** that needs to be communicated
- The network is a **limited resource** that is shared by many
- Have to work within the **constraints** of the communication system
- **Control** over the network resource is **limited** - invariably non-existent unless the service is on a paid VPN (Battleping (battleping.com), WTFast ([www.wtfast.com](http://www.wtfast.com)), Pingzapper (pingzapper.com)), which can give some quality of service guarantees

# Heterogeneity

- **Network heterogeneity** -
  - Not all users are connected to the network via the same access link
  - Some users may be more distant from the server than others - require more hops to send the data
  - To maintain fairness - drop to the lowest common denominator (LCD), but that could impact playability for better connected users
- **Graphics and processing heterogeneity** -
  - Users have very different systems that can compute, display and generate different types of visual data
  - Choosing what to display and what to omit is not an easy task as it could create an unfair advantage - e.g., not displaying foliage could make some players position visible in a shooting game that normally would not be

# Distributed Interaction

- A networked game must support **accurate collision** detection, agreement, and resolution among participants
- **Dynamic state** - Accurate collision detection is difficult because at any given point in time, no user has accurate information about the other users' current positions
- **Network delay** means that all received information is out-of-date. Messages are delayed, incur different delays, arrive out of order, lost
- **Conflicts** - one user might conclude, based on stale information, that a collision occurred, while, in fact, the other user actually moved to avoid the collision during the network delay period.
- **Distributed acoustic** information is also impacted as position could impact the sound effects at a user's device



# Real-Time System Design and Resource Management

- The system needs to be **responsive** - react to a user's input - local plausability
- The system needs to process messages carrying position and action information from other users to maintain a “pseudo” **consistent view** - shared plausability
- Physics modeling and collision detection must be performed in **real-time** to provide a realistic environment for game play.

# Failure Management

- **Network failures** - users are disconnected from the other players
- **Host/Server failures** - lost state and service for a multitude of players
- **Host/Server closure** - current players are OK, but new incoming players maybe prevented from joining the system
- **Part failures** - some aspect of the system is malfunctioning and may not stop play but could affect the quality of play, e.g., the audio server

# Scalability

- Two forms of scalability:
  - Number of **end hosts** connected simultaneously to the system
  - Number of **game entities** that are part of the game environment that can change over time
- Both forms, when they go up in number, will require lots more processing power and network resources to communicate game state for consistent game views



# Intricacy of a MPOG System

- **Very complex** - designer/developer does not have control over all aspects of the system, e.g., the data network, the end user devices, etc.
- **Engineering tradeoffs** - balance the system and make the game “playable” at all levels for a very heterogeneous environment.
- **Many components** - other than the direct gaming components - databases, security, redundancy, accounting,.....

