

# From Simulation to Games

## Architectures for Large Scale MMOGs

Fabio Biselli

Università di Bologna  
Laurea Magistrale in Informatica  
Corso di Simulazione di Sistemi

Bologna, 11 Dicembre 2014

- 1 From Visual Simulation to Virtual Reality to Games
  - Defining Serious Games
  - Creating a Science of Games
  - Game Research Agenda
- 2 An Architecture for Distributed Simulation Games
  - Drivers for the Architecture
  - Requirements for the Architecture
  - An Architecture for Distributed Games
- 3 An Architecture Supporting Large Scale MMOGs
  - Requirements
  - An Overlapping Zone-Based Architecture
  - Analysis

- Beginning in 1997 with the publication of an NRC report titled *“Modeling and Simulation — Linking Entertainment and Defense”*, the video game community has pushed into spaces previously the domain of the VR community.

- Beginning in 1997 with the publication of an NRC report titled *“Modeling and Simulation — Linking Entertainment and Defense”*, the video game community has pushed into spaces previously the domain of the VR community.
- Because much of the research and development being conducted in the games community parallels the VR communitys efforts, it has the potential to affect a greater audience.

- Beginning in 1997 with the publication of an NRC report titled *“Modeling and Simulation — Linking Entertainment and Defense”*, the video game community has pushed into spaces previously the domain of the VR community.
- Because much of the research and development being conducted in the games community parallels the VR communitys efforts, it has the potential to affect a greater audience.
- Researchers who want their work to remain relevant must realign to focus on game research and development. Research in the games arena affects not just the entertainment industry but also the government and corporate organizations that could benefit from the training, simulation, and education opportunities that serious games provide.

# Outline

- 1 From Visual Simulation to Virtual Reality to Games
  - Defining Serious Games
  - Creating a Science of Games
  - Game Research Agenda
- 2 An Architecture for Distributed Simulation Games
  - Drivers for the Architecture
  - Requirements for the Architecture
  - An Architecture for Distributed Games
- 3 An Architecture Supporting Large Scale MMOGs
  - Requirements
  - An Overlapping Zone-Based Architecture
  - Analysis

# What is a game?

## A Dictionary Definition

A physical or mental contest, played according to specific rules, with the goal of amusing or rewarding the participants.

# What is a game?

## A Dictionary Definition

A physical or mental contest, played according to specific rules, with the goal of amusing or rewarding the participants.

## A Video Game Definition

A game played **against a computer**, which would more accurately be worded as a game played **with a computer**.



# What is a serious game?

## A Serious Game Definition

A mental contest, played with a computer in accordance with specific rules, **that uses entertainment to** further government or corporate training, education, health, public policy, and strategic communication objectives.

# What is a serious game?

## A Serious Game Definition

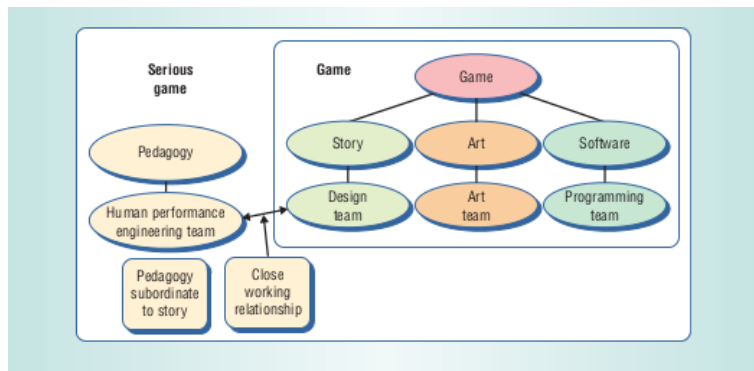
A mental contest, played with a computer in accordance with specific rules, **that uses entertainment to** further government or corporate training, education, health, public policy, and strategic communication objectives.

## A Developer Definition

Bing Gordon, chief creative officer of video and computer games developer Electronic Arts, once told that he defines video games as **“story, art, and software”**.

# What is a serious game?

Serious games have more than just story, art, and software, however.



# Outline

- 1 From Visual Simulation to Virtual Reality to Games
  - Defining Serious Games
  - **Creating a Science of Games**
  - Game Research Agenda
- 2 An Architecture for Distributed Simulation Games
  - Drivers for the Architecture
  - Requirements for the Architecture
  - An Architecture for Distributed Games
- 3 An Architecture Supporting Large Scale MMOGs
  - Requirements
  - An Overlapping Zone-Based Architecture
  - Analysis

# Creating a Science of Games

## Growing Interest

- The development and wide release of the America's Army game began a **revolution** in thinking about the potential role of video games for **nonentertainment** domains.

# Creating a Science of Games

## Growing Interest

- The development and wide release of the America's Army game began a **revolution** in thinking about the potential role of video games for **nonentertainment** domains.
- Those who have grown up playing games indicated that a game-centered research and educational program could offer many positive benefits.

# Creating a Science of Games

## Growing Interest

- The development and wide release of the America's Army game began a **revolution** in thinking about the potential role of video games for **nonentertainment** domains.
- Those who have grown up playing games indicated that a game-centered research and educational program could offer many positive benefits.
- The announcement of America's Army at the 2002 Electronics Entertainment Expo (E3) prompted the US Army to commission a study of the game to see if it could be used for training.

# Creating a Science of Games

## An America's Army History

- In the summer of 2002, a US Army captain from Fort Benning indicated that there “was insufficient fidelity in the game for it to be of any use in training”.



# Creating a Science of Games

## An America's Army History

- In the summer of 2002, a US Army captain from Fort Benning indicated that there “was insufficient fidelity in the game for it to be of any use in training”.
- In October 2002, when a staff sergeant from Fort Benning approached the America's Army booth, an enthusiastic gamer who had played the game from day one, he told the staff, “We love this game at Benning. We use it for training”.

# Creating a Science of Games

## An America's Army History

- In the summer of 2002, a US Army captain from Fort Benning indicated that there “was insufficient fidelity in the game for it to be of any use in training”.
- In October 2002, when a staff sergeant from Fort Benning approached the America's Army booth, an enthusiastic gamer who had played the game from day one, he told the staff, “We love this game at Benning. We use it for training”.

“Street finds its own use for things”

The sergeant had bypassed the Army's requirements documents and formal studies and deployed the game on his own initiative.

# Outline

- 1 From Visual Simulation to Virtual Reality to Games
  - Defining Serious Games
  - Creating a Science of Games
  - Game Research Agenda
- 2 An Architecture for Distributed Simulation Games
  - Drivers for the Architecture
  - Requirements for the Architecture
  - An Architecture for Distributed Games
- 3 An Architecture Supporting Large Scale MMOGs
  - Requirements
  - An Overlapping Zone-Based Architecture
  - Analysis

# Game Research Agenda

To influence the future of both serious and entertainment games

Developers must create a research and development agenda with three components:

# Game Research Agenda

To influence the future of both serious and entertainment games

Developers must create a research and development agenda with three components:

- infrastructure;

# Game Research Agenda

To influence the future of both serious and entertainment games

Developers must create a research and development agenda with three components:

- infrastructure;
- cognitive game design;

# Game Research Agenda

To influence the future of both serious and entertainment games

Developers must create a research and development agenda with three components:

- infrastructure;
- cognitive game design;
- immersion.

# Game Research Agenda

## Infrastructure

The underlying software and hardware necessary for developing interactive games include **MMOG architectures**, game engines and tools, streaming media, next-generation consoles, wireless and mobile device.



# Game Research Agenda

## Infrastructure

The underlying software and hardware necessary for developing interactive games include **MMOG architectures**, game engines and tools, streaming media, next-generation consoles, wireless and mobile device.

MMOGs pose the fundamental research question of how to develop dynamically extensible and semantically interoperable software architectures.

# Game Research Agenda

## Infrastructure

The underlying software and hardware necessary for developing interactive games include **MMOG architectures**, game engines and tools, streaming media, next-generation consoles, wireless and mobile device.

MMOGs pose the fundamental research question of how to develop dynamically extensible and semantically interoperable software architectures.

This work, of interest to gaming in general, has special relevance for the large governmental game-based simulation sector.

# Game Research Agenda

## Cognitive Game Design and Immersion

### Cognitive Game Design:

- modeling and simulating computer characters, story, and human emotion;
- analyzing large-scale game play;
- innovating new game genres and play styles;
- integrating pedagogy with story in the interactive game medium.

# Game Research Agenda

## Cognitive Game Design and Immersion

### Cognitive Game Design:

- modeling and simulating computer characters, story, and human emotion;
- analyzing large-scale game play;
- innovating new game genres and play styles;
- integrating pedagogy with story in the interactive game medium.

### Immersion:

- computer graphics, sound, and haptics;
- affective computing/sensing human state and emotion;
- advanced user interfaces.

- Simulation has long been established as a useful method for inquiry into complex, ill-structured problem situations.

- Simulation has long been established as a useful method for inquiry into complex, ill-structured problem situations.
- One of its related fields is gaming, in which the tools are called simulation games.

- Simulation has long been established as a useful method for inquiry into complex, ill-structured problem situations.
- One of its related fields is gaming, in which the tools are called simulation games.
- There is a **need to develop an architecture** on which distributed simulation games can be built.

# Outline

- 1 From Visual Simulation to Virtual Reality to Games
  - Defining Serious Games
  - Creating a Science of Games
  - Game Research Agenda
- 2 An Architecture for Distributed Simulation Games
  - Drivers for the Architecture
  - Requirements for the Architecture
  - An Architecture for Distributed Games
- 3 An Architecture Supporting Large Scale MMOGs
  - Requirements
  - An Overlapping Zone-Based Architecture
  - Analysis



# Drivers for the Architecture Technology

In order to understand the value of an architecture for gaming, it is useful first to focus on its drivers.

# Drivers for the Architecture

## Technology

In order to understand the value of an architecture for gaming, it is useful first to focus on its drivers.

Technology is considered to be the driver for the development of a multi-player, distributed architecture for gaming.

# Drivers for the Architecture

## Technology

In order to understand the value of an architecture for gaming, it is useful first to focus on its drivers.

Technology is considered to be the driver for the development of a multi-player, distributed architecture for gaming.

The question is:

Why? To what extent have the technological developments of the last decade improved understanding and acceptance?

# Drivers for the Architecture

## Why Technology?

### The First Driver

Incorporating state of the art web-enabled technologies provides players with a simulated environment which actually resembles the environment for which they are trained.

# Drivers for the Architecture

## Why Technology?

### The First Driver

Incorporating state of the art web-enabled technologies provides players with a simulated environment which actually resembles the environment for which they are trained.

### A Second Driver

Incorporating these technologies provides opportunities to construct a more complex and realistic environment.

# Drivers for the Architecture

## Why Technology?

### The First Driver

Incorporating state of the art web-enabled technologies provides players with a simulated environment which actually resembles the environment for which they are trained.

### A Second Driver

Incorporating these technologies provides opportunities to construct a more complex and realistic environment.

### A Third Driver

Allows to embed algorithms for operational decision making in simulation games.

# Outline

- 1 From Visual Simulation to Virtual Reality to Games
  - Defining Serious Games
  - Creating a Science of Games
  - Game Research Agenda
- 2 An Architecture for Distributed Simulation Games
  - Drivers for the Architecture
  - Requirements for the Architecture
  - An Architecture for Distributed Games
- 3 An Architecture Supporting Large Scale MMOGs
  - Requirements
  - An Overlapping Zone-Based Architecture
  - Analysis

# Requirements for the Architecture

## The three U's

When considering simulation games, there are three “U”'s that are important: the **usefulness** of the tools and methods, their **usability**, and finally their **usage**.



# Requirements for the Architecture

## Usefulness & Usability

### Usefulness



# Requirements for the Architecture

## Usefulness & Usability

### Usefulness

- Use simulated players to model players to increase complexity, or to increase dynamics.

# Requirements for the Architecture

## Usefulness & Usability

### Usefulness

- Use simulated players to model players to increase complexity, or to increase dynamics.
- Depending on the complexity of the problem at hand, it is useful to support long ( $> 1$  day) playing times.

# Requirements for the Architecture

## Usefulness & Usability

### Usefulness

- Use simulated players to model players to increase complexity, or to increase dynamics.
- Depending on the complexity of the problem at hand, it is useful to support long ( $> 1$  day) playing times.
- Persistency of data and a scenario service suitable for long time control are needed to support long games.

# Requirements for the Architecture

## Usefulness & Usability

### Usefulness

- Use simulated players to model players to increase complexity, or to increase dynamics.
- Depending on the complexity of the problem at hand, it is useful to support long ( $> 1$  day) playing times.
- Persistency of data and a scenario service suitable for long time control are needed to support long games.

### Usability

User interfaces placed on top of generic services.

# Requirements for the Architecture

## Usage & Other Requirements

### Usage

# Requirements for the Architecture

## Usage & Other Requirements

### Usage

- **Scalability** is needed to support variable amounts of players.

# Requirements for the Architecture

## Usage & Other Requirements

### Usage

- **Scalability** is needed to support variable amounts of players.
- **Persistency** is needed to support the saving of data when a (human) player is deliberately not on-line, or when the connection with a player is unexpectedly lost.



# Requirements for the Architecture

## Usage & Other Requirements

### Usage

- **Scalability** is needed to support variable amounts of players.
- **Persistency** is needed to support the saving of data when a (human) player is deliberately not on-line, or when the connection with a player is unexpectedly lost.
- Saving occurs at fixed and/or flexible points in time.

# Requirements for the Architecture

## Usage & Other Requirements

### Usage

- **Scalability** is needed to support variable amounts of players.
- **Persistency** is needed to support the saving of data when a (human) player is deliberately not on-line, or when the connection with a player is unexpectedly lost.
- Saving occurs at fixed and/or flexible points in time.

### Other Requirements

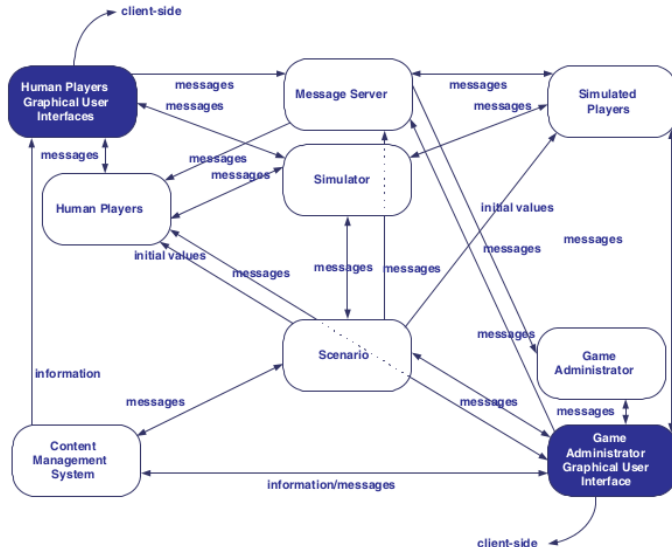
Other requirements are related to **reliability**, **robustness**, credibility and adaptivity.

# Outline

- 1 From Visual Simulation to Virtual Reality to Games
  - Defining Serious Games
  - Creating a Science of Games
  - Game Research Agenda
- 2 An Architecture for Distributed Simulation Games
  - Drivers for the Architecture
  - Requirements for the Architecture
  - An Architecture for Distributed Games
- 3 An Architecture Supporting Large Scale MMOGs
  - Requirements
  - An Overlapping Zone-Based Architecture
  - Analysis

# An Architecture for Distributed Games

## An Overview of the Architecture



# An Architecture for Distributed Games

## Technologies and Tools Used for Implementation

- The reference implementation is a J2EE system which is hosted on a JBOSS & Tomcat combination.

# An Architecture for Distributed Games

## Technologies and Tools Used for Implementation

- The reference implementation is a J2EE system which is hosted on a JBOSS & Tomcat combination.
- The simulation core was provided by the DSOL (Jacobs, Lang and Verbraeck 2002) suite for simulation which was accessed through RMI.

# An Architecture for Distributed Games

## Technologies and Tools Used for Implementation

- The reference implementation is a J2EE system which is hosted on a JBOSS & Tomcat combination.
- The simulation core was provided by the DSOL (Jacobs, Lang and Verbraeck 2002) suite for simulation which was accessed through RMI.
- A combination of HTML and JSP pages provided most of the content to the actual players.

# An Architecture for Distributed Games

## Technologies and Tools Used for Implementation

- The reference implementation is a J2EE system which is hosted on a JBOSS & Tomcat combination.
- The simulation core was provided by the DSOL (Jacobs, Lang and Verbraeck 2002) suite for simulation which was accessed through RMI.
- A combination of HTML and JSP pages provided most of the content to the actual players.
- Persistency of the game was provided by MySQL and game scenarios were parsed from locally accessible XML-files.



# An Architecture for Distributed Games

## Technologies and Tools Used for Implementation

- The reference implementation is a J2EE system which is hosted on a JBOSS & Tomcat combination.
- The simulation core was provided by the DSOL (Jacobs, Lang and Verbraeck 2002) suite for simulation which was accessed through RMI.
- A combination of HTML and JSP pages provided most of the content to the actual players.
- Persistency of the game was provided by MySQL and game scenarios were parsed from locally accessible XML-files.
- The Java Message Service is used for the message service.

- Highly interactive virtual reality systems such as massively multiplayer online games (MMOGs) necessitates highly robust and efficient architectures.

- Highly interactive virtual reality systems such as massively multiplayer online games (MMOGs) necessitates highly robust and efficient architectures.
- Distributed implementations must deal with challenges such as: supporting very large numbers of users, the need to maintain robustness, balancing the processing load, reducing user latency, and minimizing thrashing effects. There are no unified methods that attack these problems cohesively.

- Highly interactive virtual reality systems such as massively multiplayer online games (MMOGs) necessitates highly robust and efficient architectures.
- Distributed implementations must deal with challenges such as: supporting very large numbers of users, the need to maintain robustness, balancing the processing load, reducing user latency, and minimizing thrashing effects. There are no unified methods that attack these problems cohesively.
- The next paper presents a software design intended for distributed high performance computing facilities in which the world is divided into a regular lattice of overlapping cells (providing redundancy), which are dynamically assigned to servers within the High Performance Computing.

# Outline

- 1 From Visual Simulation to Virtual Reality to Games
  - Defining Serious Games
  - Creating a Science of Games
  - Game Research Agenda
- 2 An Architecture for Distributed Simulation Games
  - Drivers for the Architecture
  - Requirements for the Architecture
  - An Architecture for Distributed Games
- 3 An Architecture Supporting Large Scale MMOGs
  - Requirements
  - An Overlapping Zone-Based Architecture
  - Analysis

# Requirements for Large-scale, Distributed System

## Robustness to Hardware Failure

Distributed systems should avoid single-point-of-failure problems where part of the world state is lost with a server failure.

# Requirements for Large-scale, Distributed System

## Robustness to Hardware Failure

Distributed systems should avoid single-point-of-failure problems where part of the world state is lost with a server failure.

## Balancing of Processing Load

Dynamic entity behaviors demand responsive methods for shifting processor load to conserve computational resources.

# Requirements for Large-scale, Distributed System

## Robustness to Hardware Failure

Distributed systems should avoid single-point-of-failure problems where part of the world state is lost with a server failure.

## Balancing of Processing Load

Dynamic entity behaviors demand responsive methods for shifting processor load to conserve computational resources.

## Mitigation of Thrashing

Distributed systems should avoid repeated inefficient transitions of objects between resources.



# Requirements for Large-scale, Distributed System

## Robustness to Hardware Failure

Distributed systems should avoid single-point-of-failure problems where part of the world state is lost with a server failure.

## Balancing of Processing Load

Dynamic entity behaviors demand responsive methods for shifting processor load to conserve computational resources.

## Mitigation of Thrashing

Distributed systems should avoid repeated inefficient transitions of objects between resources.

## Reduction of User Latency

Game systems must avoid unacceptable latencies between the user clients and the servers.

# Related Work

Distributed MMOGs and other multi-server, large-scale VR systems use a variety of methods to address the four desired capabilities mentioned above.

# Related Work

Distributed MMOGs and other multi-server, large-scale VR systems use a variety of methods to address the four desired capabilities mentioned above.

- Sharding method.

# Related Work

Distributed MMOGs and other multi-server, large-scale VR systems use a variety of methods to address the four desired capabilities mentioned above.

- Sharding method.
- Zoning methods.

# Related Work

Distributed MMOGs and other multi-server, large-scale VR systems use a variety of methods to address the four desired capabilities mentioned above.

- Sharding method.
- Zoning methods.
- Peer-to-Peer (P2P) paradigm.

# Outline

- 1 From Visual Simulation to Virtual Reality to Games
  - Defining Serious Games
  - Creating a Science of Games
  - Game Research Agenda
- 2 An Architecture for Distributed Simulation Games
  - Drivers for the Architecture
  - Requirements for the Architecture
  - An Architecture for Distributed Games
- 3 An Architecture Supporting Large Scale MMOGs
  - Requirements
  - An Overlapping Zone-Based Architecture
  - Analysis

# An Overlapping Zone-Based Architecture

- The virtual world is divided into a lattice of small, **overlapping cells** (zones).

# An Overlapping Zone-Based Architecture

- The virtual world is divided into a lattice of small, **overlapping cells** (zones).
- Cells must completely tessellate the region of interest.



# An Overlapping Zone-Based Architecture

- The virtual world is divided into a lattice of small, **overlapping cells** (zones).
- Cells must completely tessellate the region of interest.
- The cell overlap provides a mechanism for redundancy to protect against server crash, it facilitates a system for seamless migration across cell boundaries, and it allows load balancing methods.

# An Overlapping Zone-Based Architecture

- The virtual world is divided into a lattice of small, **overlapping cells** (zones).
- Cells must completely tessellate the region of interest.
- The cell overlap provides a mechanism for redundancy to protect against server crash, it facilitates a system for seamless migration across cell boundaries, and it allows load balancing methods.
- Cells are pre-distributed amongst the servers for processing.

# An Overlapping Zone-Based Architecture

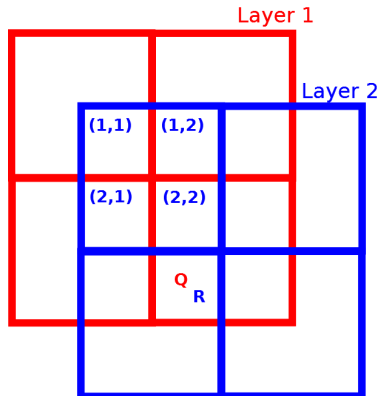
- The virtual world is divided into a lattice of small, **overlapping cells** (zones).
- Cells must completely tessellate the region of interest.
- The cell overlap provides a mechanism for redundancy to protect against server crash, it facilitates a system for seamless migration across cell boundaries, and it allows load balancing methods.
- Cells are pre-distributed amongst the servers for processing.
- Entity (object and user data in the environment) are assigned to a cell based on the location of the object in the world, and the server managing that cell can be considered the **master host** of such objects.

# An Overlapping Zone-Based Architecture

## Loose Cell Overlap Pattern

The master-slave relationship is determined by the overlap.

Server 1 (Layer 1) is master to object Q and slave to object R.  
Server 2 (Layer 2) is master to object R and slave to object Q.



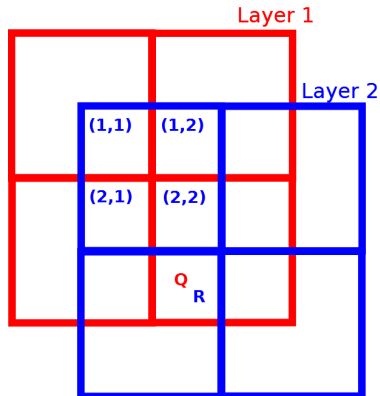
# An Overlapping Zone-Based Architecture

## Loose Cell Overlap Pattern

The master-slave relationship is determined by the overlap.

Server 1 (Layer 1) is master to object Q and slave to object R.  
Server 2 (Layer 2) is master to object R and slave to object Q.

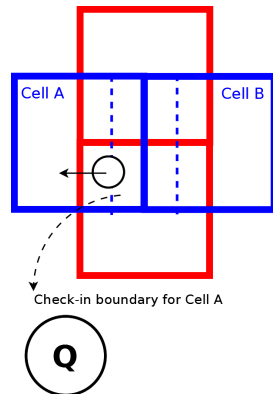
Each cell is divided into four quadrants:  
(1, 1); (1, 2); (2, 1); (2, 2).



# An Overlapping Zone-Based Architecture

## Update Responsibilities & Area of Interest

Update responsibilities of entities are transferred between servers as those entities are repositioned throughout the world.

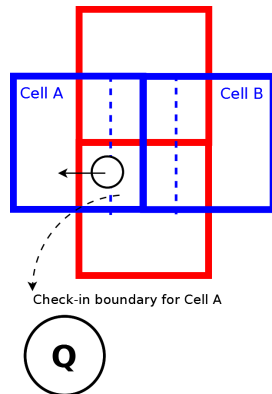


# An Overlapping Zone-Based Architecture

## Update Responsibilities & Area of Interest

Update responsibilities of entities are transferred between servers as those entities are repositioned throughout the world.

Such decisions are made by using defined check-in and check-out boundaries for entities moving between cells.



# An Overlapping Zone-Based Architecture

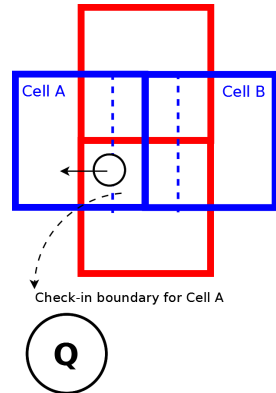
## Update Responsibilities & Area of Interest

Update responsibilities of entities are transferred between servers as those entities are repositioned throughout the world.

Such decisions are made by using defined check-in and check-out boundaries for entities moving between cells.

### Area of Interest

Area of Interest (Aoi) is defined around an entity, and when that Aoi intersects the boundary, management of the entity is shifted.



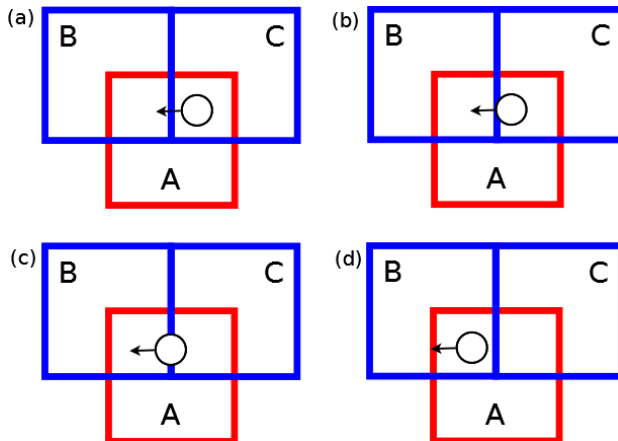


# Outline

- 1 From Visual Simulation to Virtual Reality to Games
  - Defining Serious Games
  - Creating a Science of Games
  - Game Research Agenda
- 2 An Architecture for Distributed Simulation Games
  - Drivers for the Architecture
  - Requirements for the Architecture
  - An Architecture for Distributed Games
- 3 An Architecture Supporting Large Scale MMOGs
  - Requirements
  - An Overlapping Zone-Based Architecture
  - Analysis

# Analysis













## Entity Motion



# Analysis

## Entity Motion

**Table :** Favorable Server-Motion allocation for Cost minimization.

For direction of motion of entity	Master server of entity should be that which is the master of the cell that hold the entity
  	(1,1)
  	(1,2)
  	(2,1)
  	(2,2)

# Analysis

## Server Crash & Dynamic Allocation

### Server Crash



# Analysis

## Server Crash & Dynamic Allocation

### Server Crash

- A natural fault tolerance is accomplished in this design because of the redundancy provided by overlapping cells.

# Analysis

## Server Crash & Dynamic Allocation

### Server Crash

- A natural fault tolerance is accomplished in this design because of the redundancy provided by overlapping cells.
- When a server crashes, any entities over which it had master control are already replicated on at least one other server.

# Analysis

## Server Crash & Dynamic Allocation

### Server Crash

- A natural fault tolerance is accomplished in this design because of the redundancy provided by overlapping cells.
- When a server crashes, any entities over which it had master control are already replicated on at least one other server.
- The system must merely transfer master control and needn't seek data from a crashed machine.

# Analysis

## Server Crash & Dynamic Allocation

### Server Crash




- A natural fault tolerance is accomplished in this design because of the redundancy provided by overlapping cells.
- When a server crashes, any entities over which it had master control are already replicated on at least one other server.
- The system must merely transfer master control and needn't seek data from a crashed machine.

### Dynamic Cell Allocation

Dynamic cell allocation methods between servers must accomplish a multitude of things with reasonable efficiency.



# References

-  S.A. van Houten, P.H.M. Jacobs, *An Architecture for Distributed Simulation Games*, Proceedings of the 2004 Winter Simulation Conference;
-  M. Zyda, *From Visual Simulation to Virtual Reality to Games*, IEE Computer Society, September 2005;
-  C. Ghosh, R.P. Wiegand, B. Goldiez, T.Dere, *An Architecture Supporting Large Scale MMOGs*, Proceedings of the 3rd International ICST Conference on Simulation Tools and Technique.