Chapter 15

Artificial Intelligence

INVITATION TO

Computer Science

6TH EDITION

Objectives

After studying this chapter, students will be able to:

- Define “artificial intelligence” and describe the range of techniques and applications
- Explain the pros and cons of various knowledge representation methods
- Explain the parts of a simple neural network, how it works, and how it relates to real neurons
- Construct a state-space tree for simple state-space problems
- Describe how state-space search algorithms work

Objectives (continued)

After studying this chapter, students will be able to:

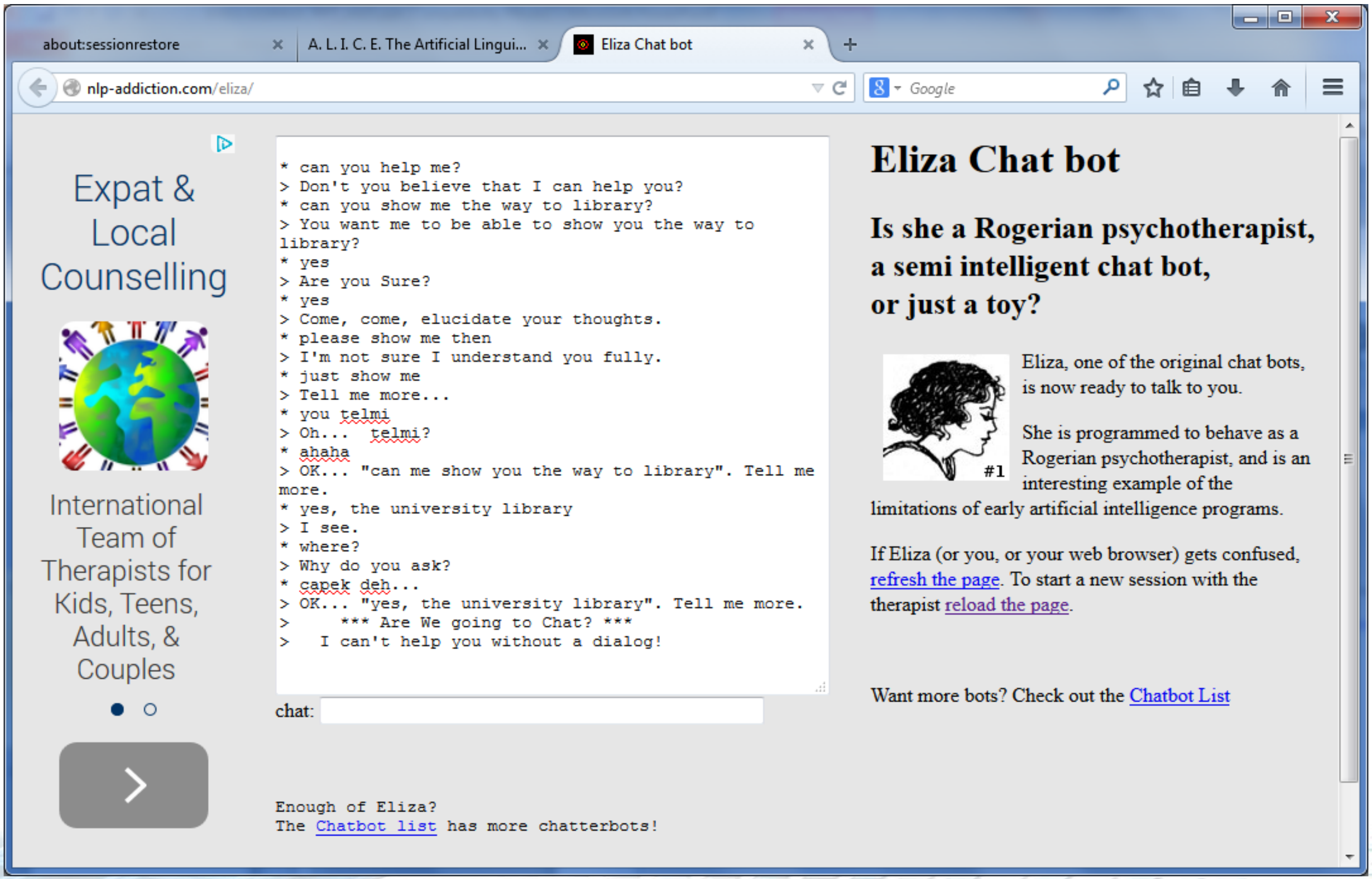
- List the reasoning techniques described here, and explain how each works
- Describe the use of state-space search to play games, and why certain games are easier or more difficult
- Describe how Watson approached the Jeopardy! problem, and why the problem is difficult
- Explain what a robot is, and the tasks robots are currently suited to do

Introduction

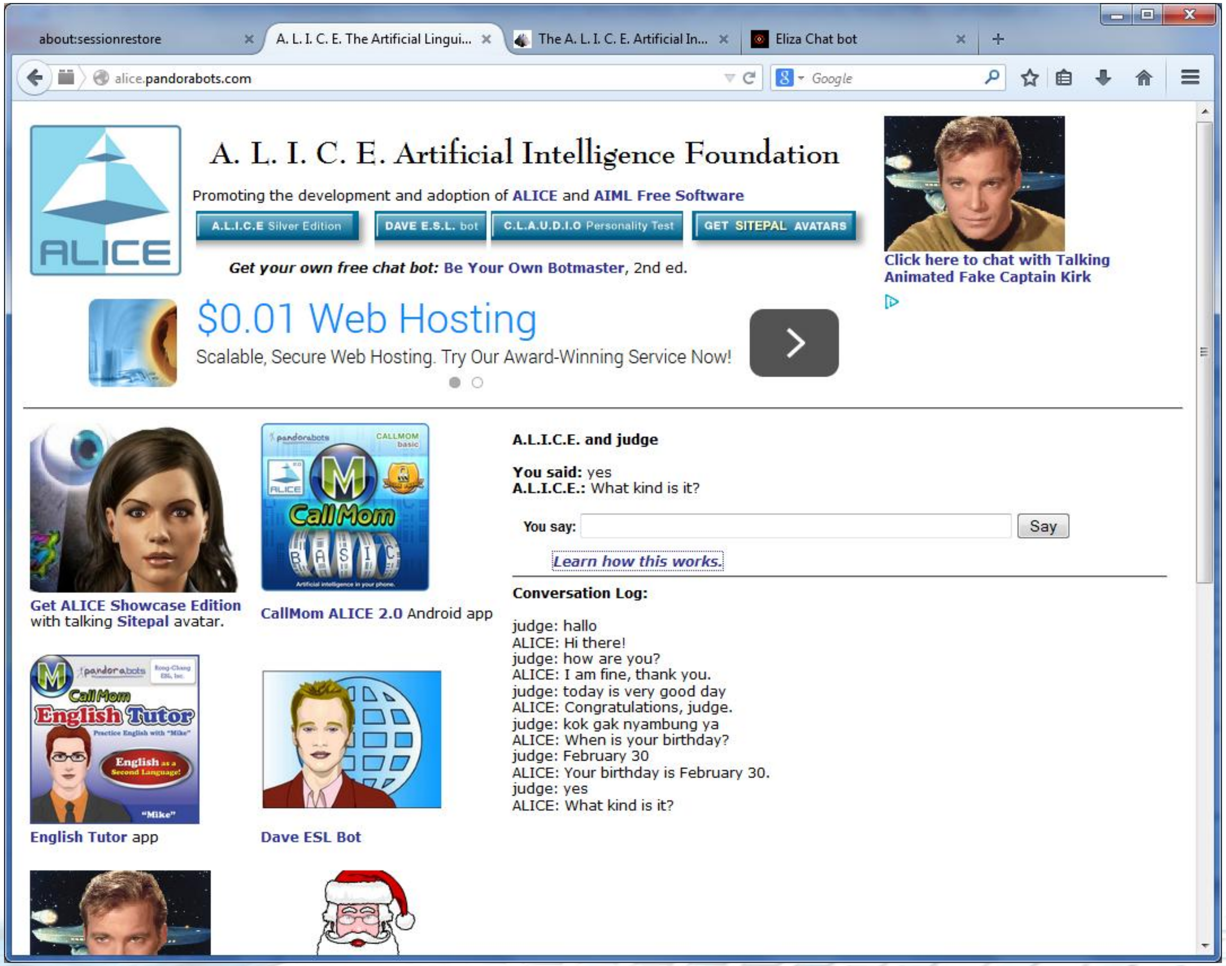
- **Artificial Intelligence (AI)**: creating computer systems that exhibit aspects of intelligence
- What is intelligence?


●●●○○ TSEL 3G
20:49
64%
"Mary Chris"
Sorry, [redacted], I don't see 'Mary Chris' in your contacts.
"Mary Chris Mas"
I'm not sure I understand.
"Mary Christmas"
I'm not sure I understand.
"Merry I said Merry Christmas"
Wait... Christmas Day isn't today, it's on December 25th!
"Stupid"
tap to edit
I'm just trying to help you.
?

http://nlp-addiction.com/eliza/



http://alice.pandorabots.com/



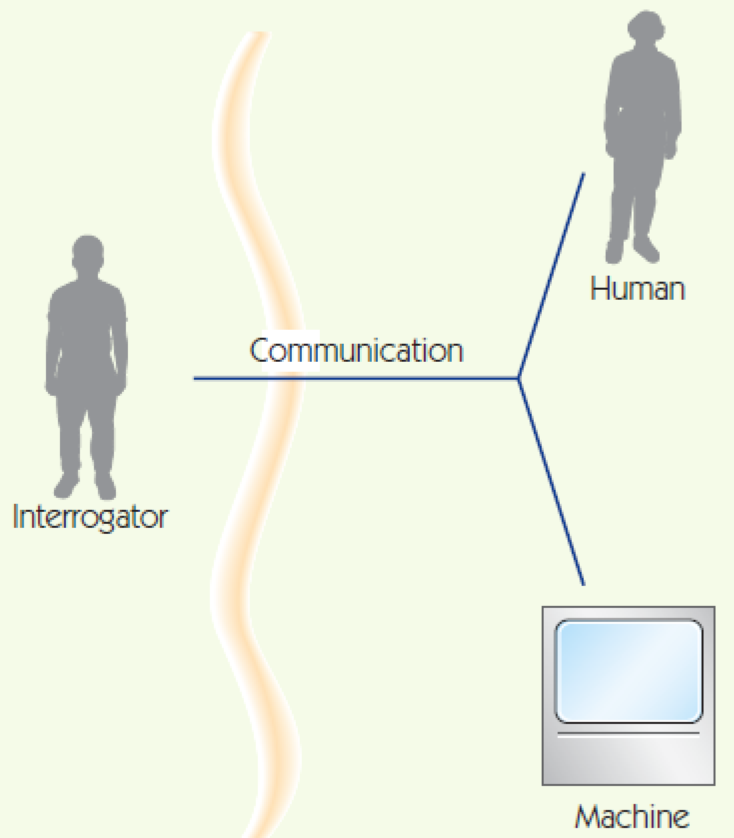
http://alice.pandorabots.com/

Introduction

- The **Turing Test**:
  - Human judge questions two hidden entities
  - One entity is a person
  - One entity is a computer
  - If judge cannot distinguish computer from person, then computer is intelligent!

Introduction (continued)

FIGURE 15.1



The Turing test

A Division of Labor

- Computational Tasks
  - E.g., managing a payroll
- Recognition Tasks
  - E.g., Understanding the spoken word
- Reasoning Tasks
  - E.g., Planning your major in college

A Division of Labor (continued)

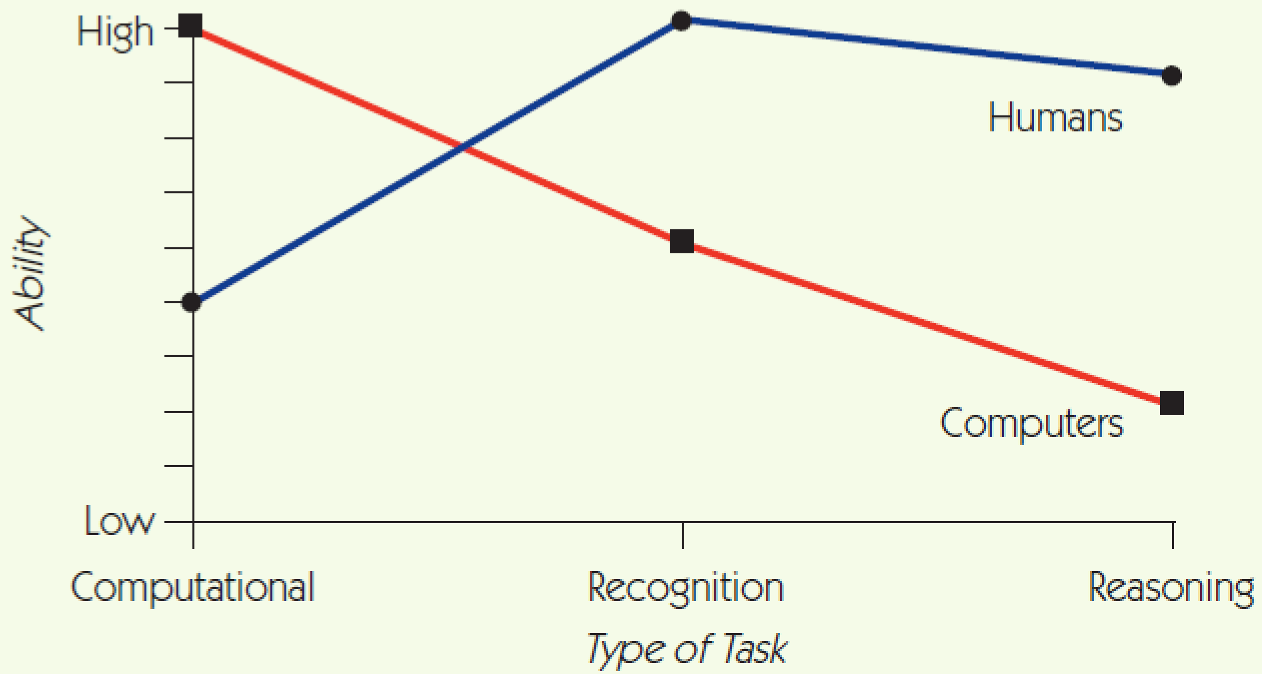
- Computational Tasks
  - Typically have algorithmic solutions
  - Computers perform faster than humans
  - Computer perform more accurately than humans
- Recognition Tasks
  - Process massive amounts of sensory information
  - Access massive amounts of past experience
  - Require approximation
  - Humans perform much better than computers

A Division of Labor (continued)

- Reasoning Tasks
  – Formal reasoning can be automated to some extent
    - Problems become intractable quickly
  – Common-sense reasoning
    - Requires great experience and knowledge

A Division of Labor (continued)

FIGURE 15.2



Human and computer capabilities

Knowledge Representation

- How can we represent knowledge for the computer?
- Natural language
  - Use requires understanding of the meanings of words and combinations of words
  - “Spot is a brown dog”
  - “Every dog has four legs”

Knowledge Representation (continued)

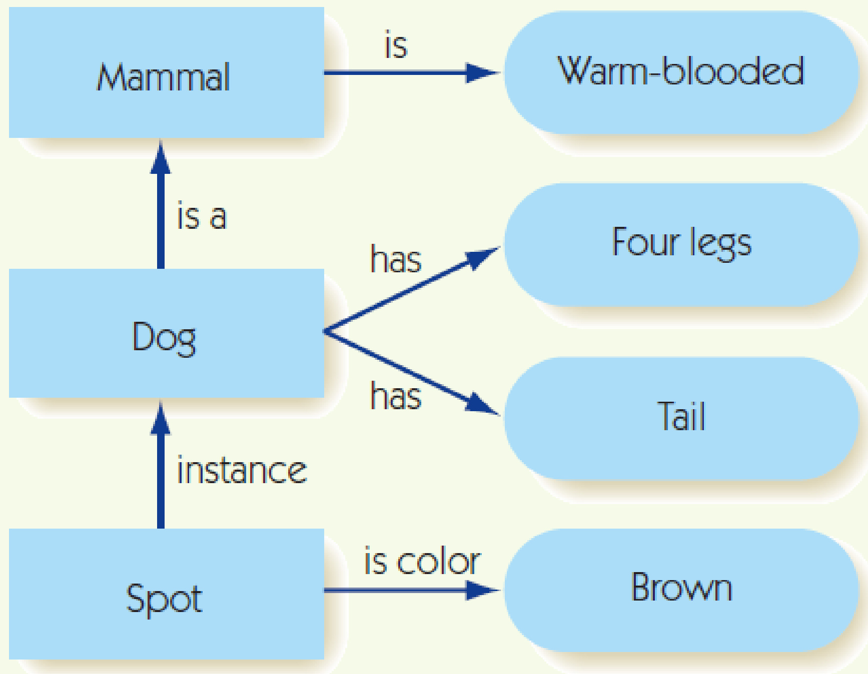
- **Formal language**
  - Language of formal logic
  - “Spot is a brown dog” becomes
    - *dog(Spot) AND brown(Spot)*
  - “Every dog has four legs” becomes
    - *For every x, if x is a dog then x has four legs*
    - *(∀x) dog(x) -> four-legged(x)*

Knowledge Representation (continued)

- Pictorial representation
  - Knowledge as a digital picture
  - Cannot represent categorical information
    - E.g., Every dog has four legs
- Graphical representation
  - Knowledge as nodes connected by edges
  - **Semantic net**:
    - Nodes for objects or categories of objects
    - Edges for relationships
    - Nodes inherit features through “isa” relationships

Knowledge Representation (continued)

FIGURE 15.3



A semantic net representation

Knowledge Representation (continued)

- Requirements of a representation
  - Adequacy: must capture all relevant information
  - Efficiency: avoid redundant information
  - Extendability: easy to add new knowledge
  - Appropriate: easy to use for particular purpose

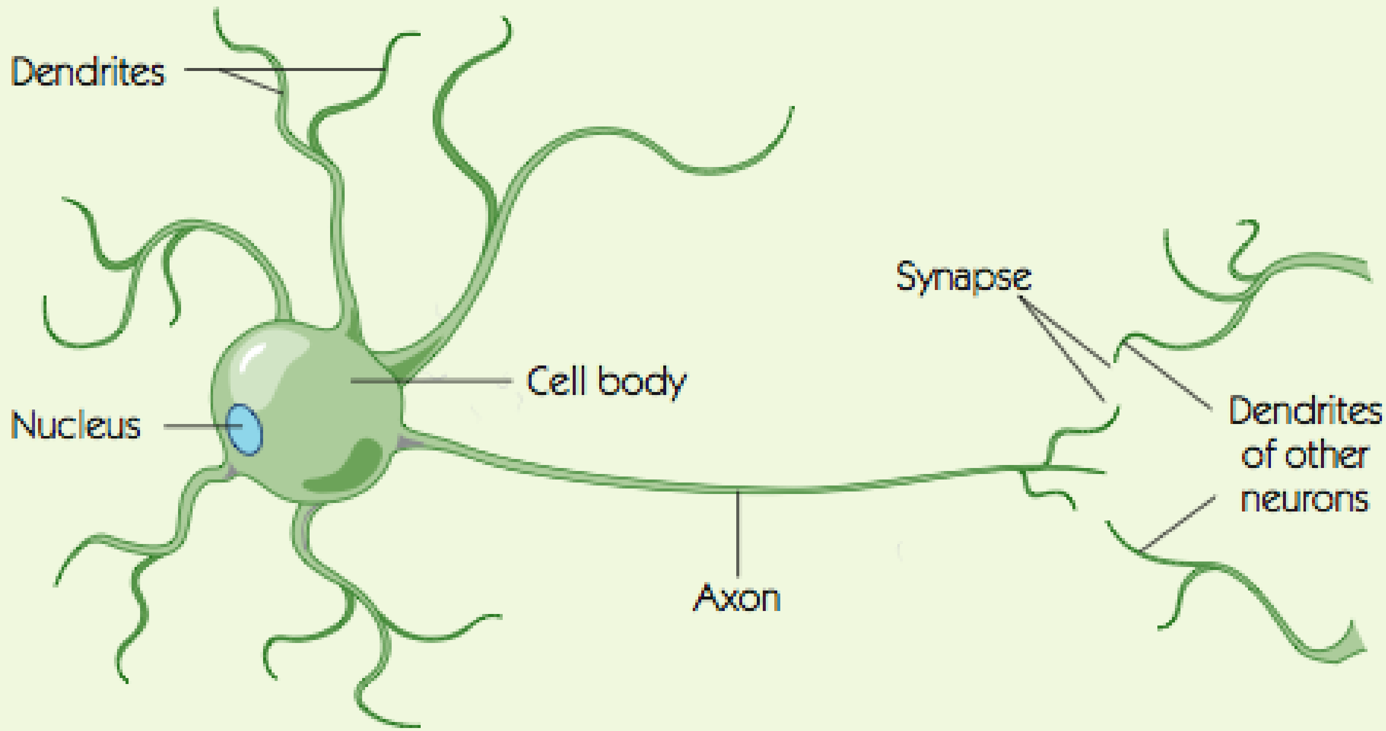
Quick Quiz 1

- Name four of the characteristics any knowledge representation must have.
- Answer: adequacy, efficiency, extendability, and appropriateness
- (True or False) Images and sounds easily represent general information about categories of objects.
- Answer: False
- (True or False) Graph representations easily represent relationships between objects.
- Answer: True

Recognition Tasks

- Some AI work attempts to mimic the brain
- Humans have 10^{12} neurons
- A neuron receives electrical stimuli from other neurons through dendrites
- A neuron sends electrical stimuli through its axon
- Signals pass through gaps, synapses
- Some synapses cause increased activation, others inhibit activation
- Neurons are like very simple computational devices

FIGURE 15.4



A neuron

Recognition Tasks (continued)

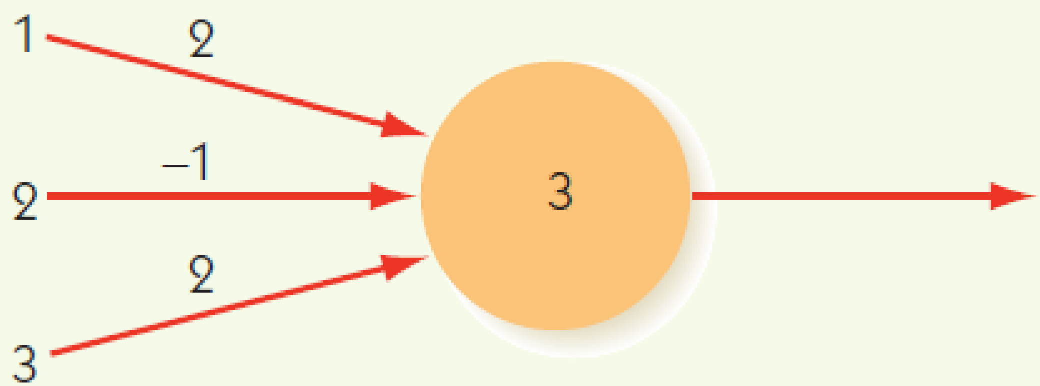
- The nervous system is like a **connectionist architecture**
  - Processing arises from many simple processors with rich and complex connections
- Processing in the brain occurs in a massively parallel way
  - Individual neurons are slow compared to computer computational speeds
  - Allows for redundancy and neuron failure (fault tolerant)

Recognition Tasks (continued)

- Artificial **neural networks** mimic the connectionist approach
- Individual artificial “neurons” have:
  - A threshold for generating output
  - An activation level
  - Incoming weighted edges
  - Outgoing weighted edges

Recognition Tasks (continued)

FIGURE 15.5

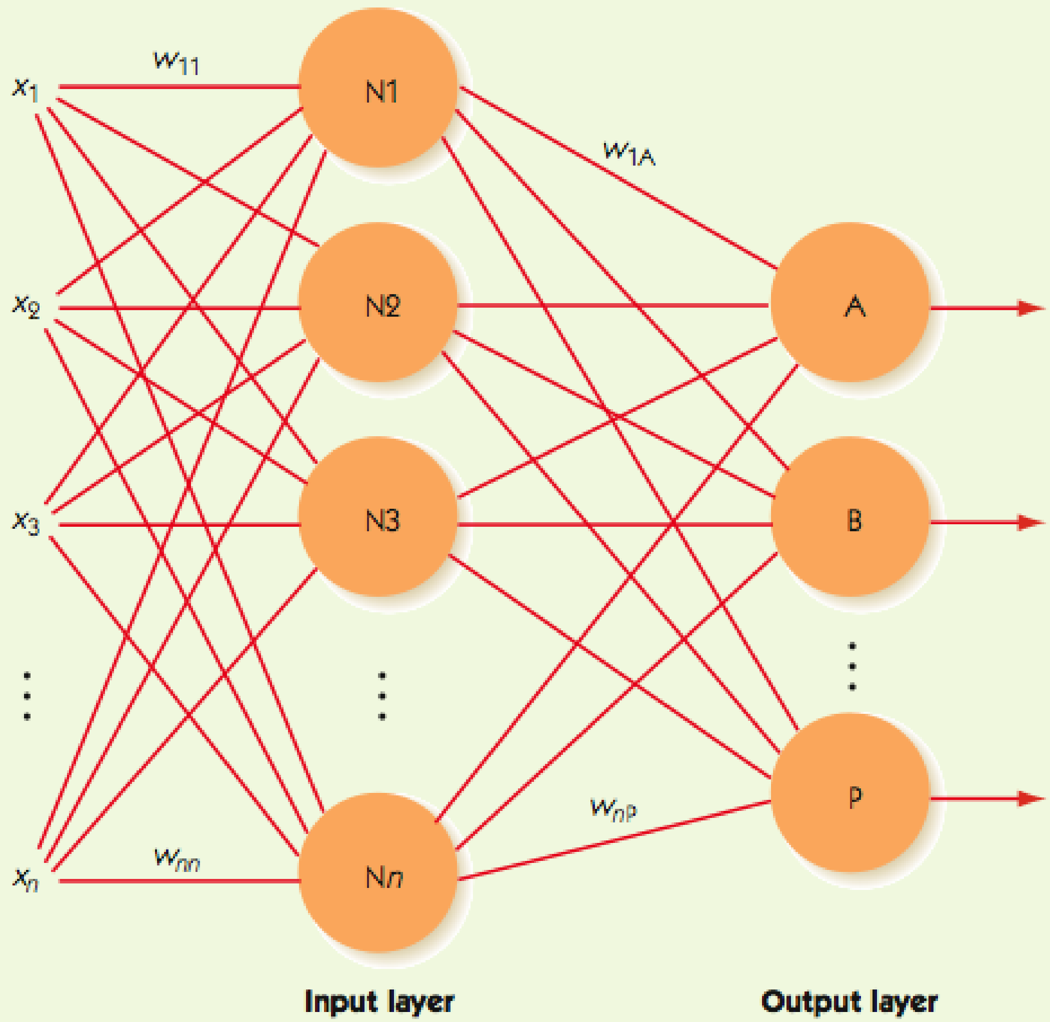


One neuron with three inputs

Recognition Tasks (continued)

- Neural networks often organized into input and output layers
- To provide an input to the network, fix the values of the input layer to 0 or 1
- Output nodes compute weighted sum of all inputs
  - Activation from node *i* to node *j* is $w_{ij} * x_i$

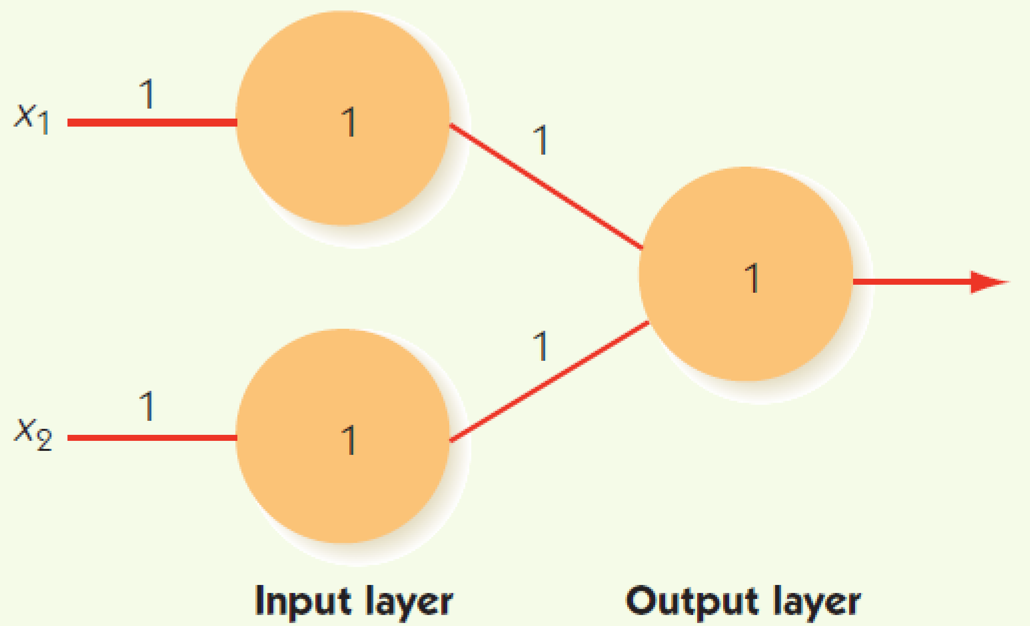
FIGURE 15.6



Neural network model

Recognition Tasks (continued)

FIGURE 15.7



A simple neural network—OR gate

Recognition Tasks (continued)

- Networks with only input and output layers
  - Can solve many problems, but
  - Cannot solve XOR (or many others)

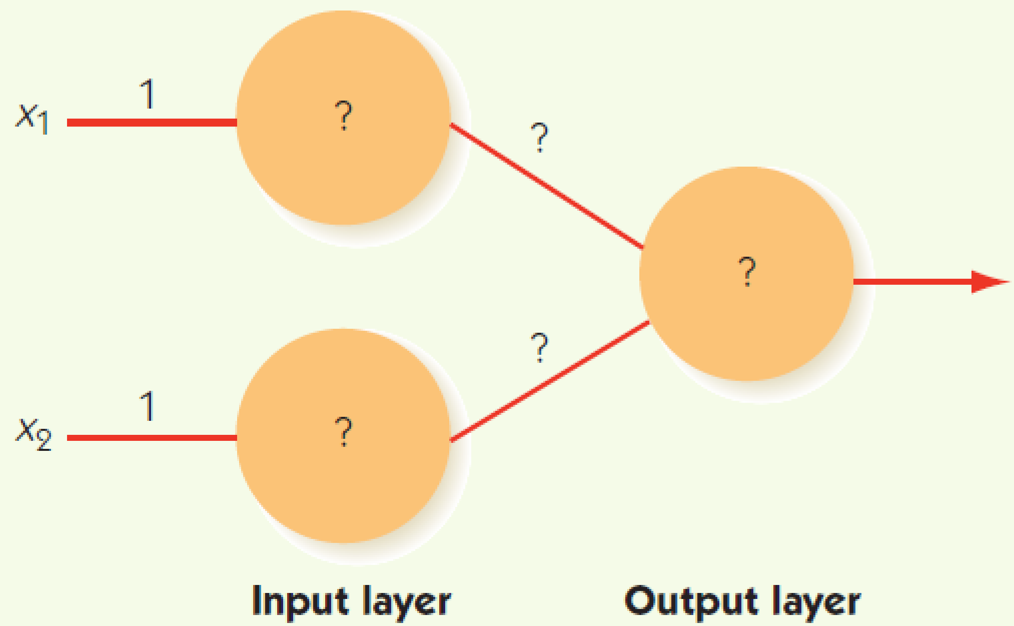
FIGURE 15.8

Inputs		Output
X_1	X_2	
0	0	0
1	0	1
0	1	1
1	1	0

The truth table for XOR

Recognition Tasks (continued)

FIGURE 15.9



An attempt at an XOR network

Recognition Tasks (continued)

- Add an intermediate layer between input and output
  - Hidden layer
- Can solve most problems given the right weights
- How can we determine the correct weights?
- Neural networks are “trained” on sample data
  - Machine learning: the network “learns” correct responses to inputs

Recognition Tasks (continued)

Training neural networks:

- **Training data**: input/output pairs where output is known to be correct for input
- Output nodes that are incorrect have quantifiable error
- Use error to update weights to generate less error
- **Backpropagation**: algorithm that propagates errors back through hidden layer(s) to input

Quick Quiz 2

- The __________ algorithm trains neural networks by passing errors from the output toward the input.
- Answer: backpropagation
- (True or False) Artificial networks are programmed by determining, by hand, the weights between neurons in the network.
- Answer: False
- (True or False) A connectionist architecture includes many simple computing devices connected in a complex network.
- Answer: True

Reasoning Tasks
Intelligent Searching

- Decision tree represents possible next items to search for
- Linear search and Binary search assume:
  - Data is organized linearly
  - Exact match is required
- What if we relax the requirements?
  - What if data is not linear?
  - What if an approximate match is okay?

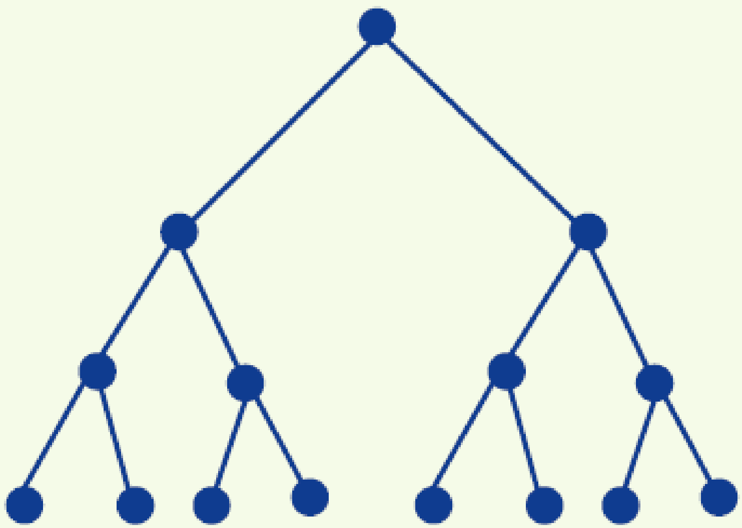
Reasoning Tasks

Intelligent Searching (continued)

FIGURE 15.10

Decision tree for sequential search

FIGURE 15.11



Decision tree for binary search

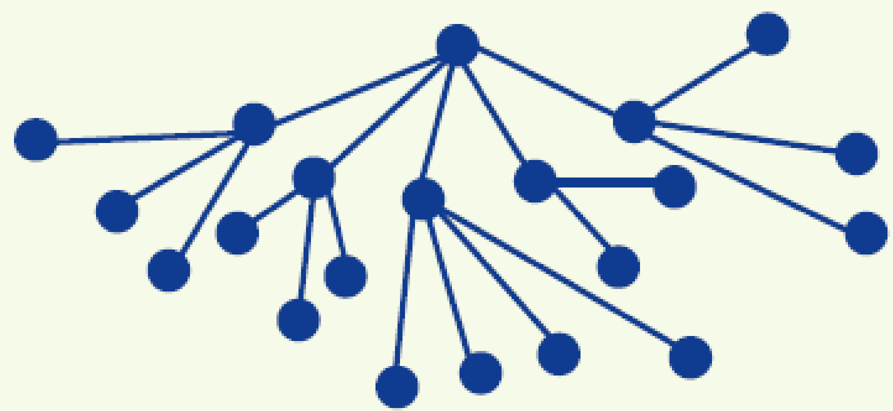
Reasoning Tasks
Intelligent Searching (continued)

- **State-space graph**
  - Each node is a state of our problem
  - A node connects to another if that state can be directly generated by the node
  - Examples: Tic-Tac-Toe, Eight-Puzzle, Maze-solving
  - Each node has many children
  - May be many paths to a goal
- **State-space search**: seeks a path from start state to goal state

Reasoning Tasks

Intelligent Searching (continued)

FIGURE 15.12



A state-space graph with exponential growth

Reasoning Tasks
Intelligent Searching (continued)

- **State-space graph**
  - Each node is a state of our problem
  - A node connects to another if that state can be directly generated by the node
  - Examples: Tic-Tac-Toe, Eight-Puzzle, Maze-solving
  - Each node has many children
  - May be many paths to a goal
- **State-space search**: seeks a path from start state to goal state

Reasoning Tasks
Intelligent Searching (continued)

- Searching for path to goal
  - Brute force: trace all branches of decision tree
    - Too slow
  - Heuristics: use educated guess to guide which branches to search
- Example: chess
  - Brute force impossible
  - Good heuristics make computer players at grand master level
  - Chess is the last “easy” hard problem

Reasoning Tasks
Swarm Intelligence (continued)

- **Swarm intelligence model**
  - Model communities of simple agents
  - Ants, termites, etc.
- Ant colonies
  - Individuals exhibit simple behaviors
  - Colonies accomplish great things
    - Finding shortest path to food
    - Constructing nests
- Ant Colony Optimization: route-finding using simulated ants

Reasoning Tasks
Intelligent Agents (continued)

- **Intelligent agent** works with human user
  - Learns user's preferences and takes actions on user's behalf
- Current examples:
  - Personalized web search
  - E-commerce site that tailors suggestions to your interests
- Future:
  - Personal travel planner: buys tickets for user
  - Office manager: screens calls, arranges meetings

Reasoning Tasks
Expert Systems (continued)

- **Expert system**: mimic reasoning in some specific domain
- **Knowledge base**: knowledge about domain
- **Inference engine**: rules for reasoning with knowledge
- Often use formal language to represent knowledge and rules for inference
- Deductive reasoning, e.g., modus ponens

Reasoning Tasks

Expert Systems (continued)

Expert system reasoning

- Forward chaining:
  - Start with assertions, look for rules to deduce new assertions
  - Given assertion A, and rule “if A then B”, then deduce B
- Backward chaining:
  - Start with a query, look for rules that could deduce query
  - Given question “Is B true?” and rule “if A then B”, then try to determine “Is A true?”

Reasoning Tasks
Expert Systems (continued)

- **Explanation facility**
  - Users can see explanation based on the reasoning chain
- **Knowledge engineering**
  - Human system builders must spend time with experts
  - Listing and codifying the expert knowledge

Reasoning Tasks
The Games We Play (continued)

Board games

- Many programs use forms of state-space search
- Tic-Tac-Toe:
  - Small state space
  - Brute force works to play perfectly
- Checkers:
  - Chinook project built and searched complete state space
  - Results can be embedded in computer player
  - Chinook can never be beaten

Reasoning Tasks
The Games We Play (continued)

Board games

- Chess
  - State space too large to solve
  - Computer players depend on heuristics
  - Deep Blue defeated world champion Gary Kasparov (1997)
- Go
  - Huge search space
  - Difficult for computer to play well
  - Current research underway to reach top levels

Reasoning Tasks
The Games We Play (continued)

FIGURE 15.13

Garry Kasparov vs. Deep Blue

Source: P. Morgan/Reuters/Landov

Reasoning Tasks
The Games We Play (continued)

Quiz games: *Jeopardy!*

- Watson defeated *Jeopardy!* champions, 2011
- Given a quiz "answer" and category:
  - Applies multiple kinds of AI agents to search database (of information from the Web)
    - Produces 300-500 candidate answers
  - Narrow to one answer and evaluate its certainty in real time
    - Scoring and evaluation done in parallel

Reasoning Tasks

The Games We Play (continued)

FIGURE 15.14

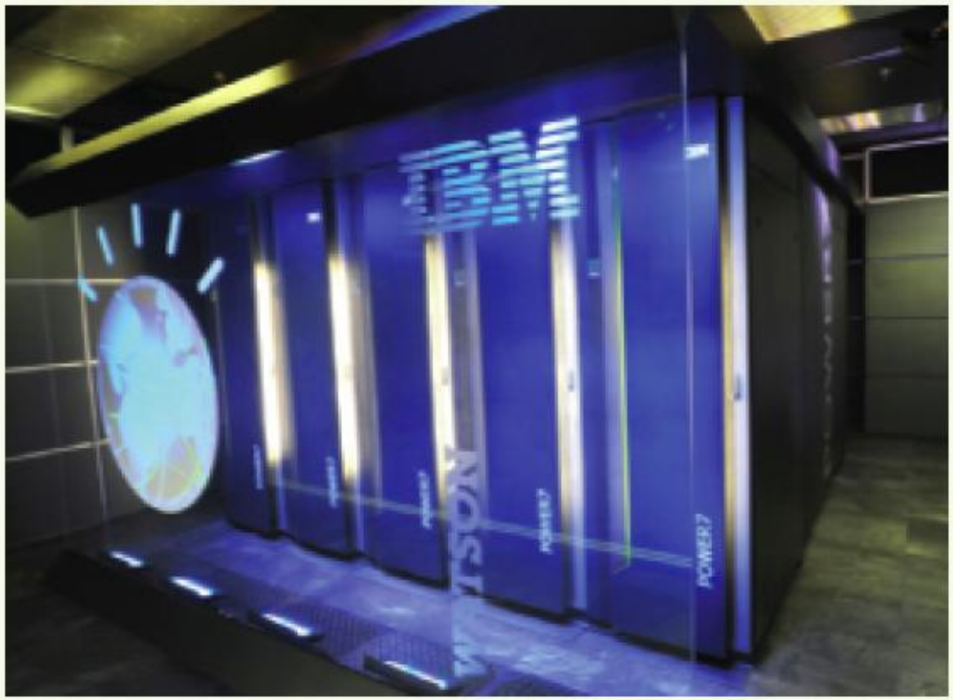
IBM's Watson beats its human opponents on *Jeopardy!*

Source: Jeopardy!/Landov

Reasoning Tasks

The Games We Play (continued)

FIGURE 15.15



IBM's Watson supercomputer

Source: Courtesy of IBM

http://www.cim.mcgill.ca/~jer/courses/java.html

Robotics

http://www.youtube.com/watch?v=W1czBcnX1Ww

Robotics

- A robot is a physical device that takes in sensory data and makes autonomous responses
- Current robot tasks:
  - Repetitive or dangerous for humans
  - Manufacturing, bomb disposal, search-and-rescue
- New research on multiple cooperating robots:
  - Schools of robot fish for studying sea life
  - Swarms of robot flies for reconnaissance
  - Groups of robot snowmobiles to study climate change

Robotics (continued)

- Humanoid robots designed for interacting with people
  - Help elderly or hospital patients
  - Monitor small children
- Japan is a leader in humanoid robots
  - Aging population needs support
- Asimo, by Honda
  - Designed to walk and move fluidly and robustly
  - Can open/close a door to go through, serve refreshments, etc.

Robotics (continued)

FIGURE 15.16

Honda's ASIMO conducting

Source: Courtesy of Honda

Robotics (continued)

- **Deliberative strategy** for robot control programs
  - Maintain detailed internal model of the world
  - Reason about sensory inputs and choose best response
- **Reactive strategy**
  - Limit/eliminate internal model
  - React immediately to sensory inputs
  - Rapid cycle from inputs to responses to more inputs

Summary

- Artificial intelligence programs solve problems in “intelligent” ways
- Knowledge may be represented in many different ways; choice of representation depends on task
- Neural networks simulate the connectionist structure of the nervous system
- Neural networks are trained to produce the correct responses to inputs
- Reasoning may often be state-space search

Summary (continued)

- Swarm intelligence uses colonies of simple agents to solve problems
- Intelligent agents would be artificial personal assistants
- Expert systems reason with expert domain knowledge
- Game-playing is a common application for AI
- Robots perform tedious and dangerous tasks
- Robots may fill in for routine human-interaction tasks in the future

Glossary

- **Artificial intelligence (AI)**: The branch of computer science that explores techniques for incorporating aspects of intelligence into computer systems.
- **Back propagation algorithm**: A training algorithm for neural networks that passes error estimates from the output layer back to earlier neurons so they can adjust connection weights.
- **Connectionist architecture**: A system with a large number of simple processors that are heavily interconnected; describes the human brain with its many simple interconnected neurons.

Glossary

- **Deliberative strategy**: An approach to robotics that says a robot must have an internal representation of its environment that guides a reasoned response to some stimulus from that environment.
- **Expert system**: Software that mimics the expertise of a human in a certain area, using facts and rules of inference to draw conclusions from those facts.
- **Explanation facility**: A process within an expert system that allows the user to see the assertions and rules used in arriving at a conclusion.

Glossary

- **Formal language**: The language of formal symbolic logic, as opposed to natural languages such as English, Japanese, Spanish, etc.
- **Intelligent agent**: Software designed to interact collaboratively with a user as a personal assistant.
- **Knowledge base**: A set of facts known to an expert system.
- **Neural network**: A massively parallel network of hardware devices simulating individual biological neurons or, alternatively, a software system that simulates an interconnected arrangement of biological neurons; sometimes called an artificial neural network.

Glossary

- **Reactive strategy**: An approach to robotics that says a robot should respond directly to environmental stimuli using heuristics but no chain of reasoning.
- **Semantic net**: A graphical representation of classes, objects, properties, and relationships.
- **State-space graph**: A representation of a set of "states" representing different configurations in a problem.
- **State-space search**: The process of finding a path through a state-space graph from an initial state to the goal state.

Glossary

- **Swarm intelligence model**: A model of artificial intelligence based on a group of simple agents (like an ant colony) that operate independently but can communicate in some way to perform cooperative tasks.
- **Training data**: Sets of input values for a neural network for which correct output values are known; weights and thresholds in the neural network are repeatedly adjusted until the network output is sufficiently close to the correct output.
- **Turing test**: A test for intelligent behavior of machines, proposed by Alan Turing in 1950; if a human can interrogate a person and a computer and not be able to detect which is which, then the computer has passed the test.

Chapter 16

Computer Graphics and Entertainment: Movies, Games, and Virtual Communities

INVITATION TO

Computer Science

6TH EDITION

Objectives

After studying this chapter, students will be able to:

- Explain the history of computer games
- Describe the transformation of movie animation caused by CGI
- List the phases of the graphics pipeline
- Explain how a tessellation represents a 3D graphics object
- Apply a transformation matrix to translate a vertex
- Describe the ray tracing approach to image rendering

Objectives (continued)

After studying this chapter, students will be able to:

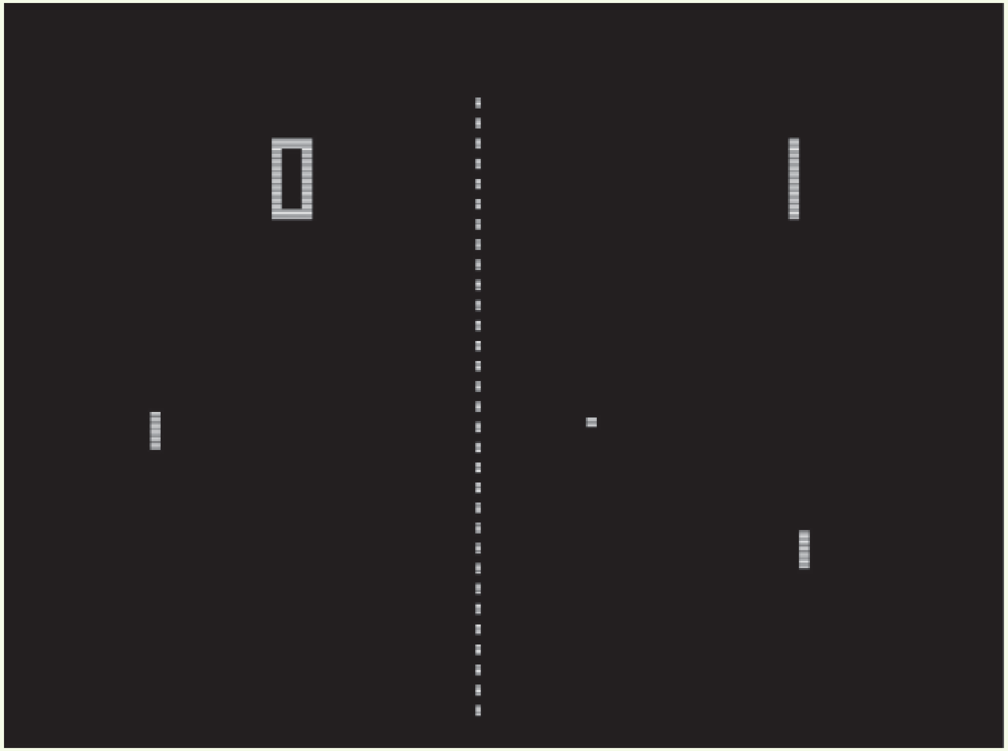
- Describe the issues faced by video game graphics rendering
- Explain the purpose and function of a GPU
- List and explain other techniques used for real-time graphics
- Describe how MMOGs work, and the particular challenges they face
- Explain how virtual communities like *Second Life* operate as MMOGs

Introduction

- Early computers were too expensive and rare for use as entertainment (1950s-1960s)
- When **minicomputers** arrived, college students began writing games for them (1970s)
- Atari created Pong, first commercial game (1972)
- Use in movies grew in early 1990s (Terminator 2, Jurassic Park)
- Now entertainment is a huge segment of the industry ($40-$60 billion industry)

Introduction (continued)

FIGURE 16.1



Pong—One of the first computer-based video games

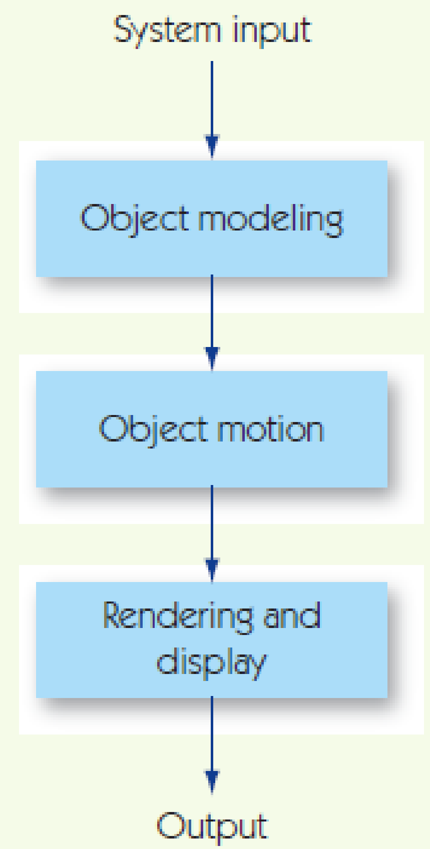
Computer-Generated Imagery (CGI)

- Movie animation in the past
  - Hand-drawn (e.g., Walt Disney)
  - **Stop-motion animation**
    - Clay figures posed, photographed, then moved slightly
- Computer animation developed in 1980s and 1990s
  - Terminator 2: Judgment Day
  - Jurassic Park
- **Computer-generated imagery (CGI)** produces **photorealistic animation**

Computer-Generated Imagery (CGI) (continued)

- The **graphics pipeline**: sequence of tasks to generate CGI images/movies
- **Object modeling**: create model of 3D object
- **Object motion**: apply transformations to move objects
- **Rendering and display**: incorporate lighting, shadows, and textures

FIGURE 16.2



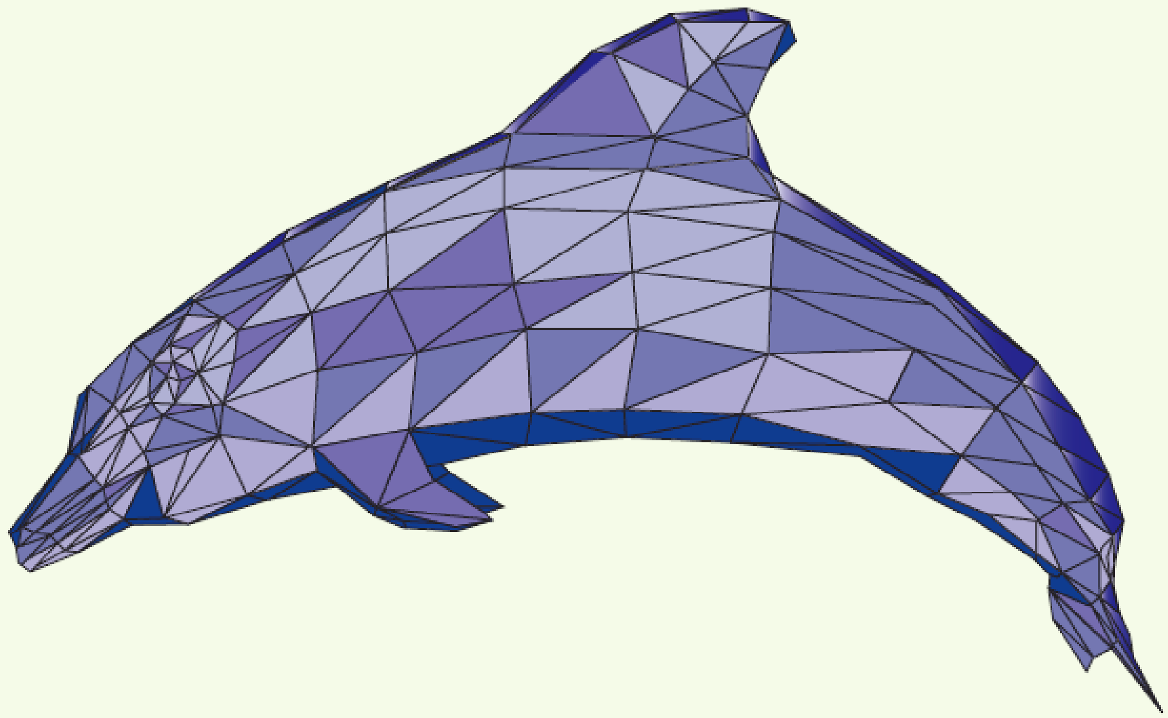
A simplified three-stage graphics pipeline

Computer-Generated Imagery (CGI) (continued)

- **Wireframe modeling**, common approach to object modeling
- Take an image and create a **tessellation**
  - Divide object surface into planar polygons
    - Polygons share edges and vertices
    - Polygons completely cover object surface
    - Polygon mesh: result looks like a wire model
  - Convert to three dimensions
  - Generate a **vertex list**, a table of all vertices and their 3D position

Computer-Generated Imagery (CGI) (continued)

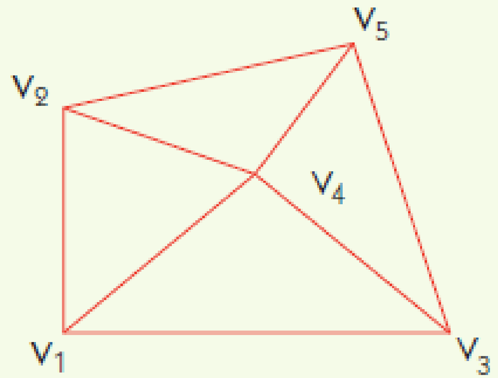
FIGURE 16.3



Wireframe model of a dolphin (based on image in Wikipedia entry on polygon meshes)

FIGURE 16.4

(a)



(b)

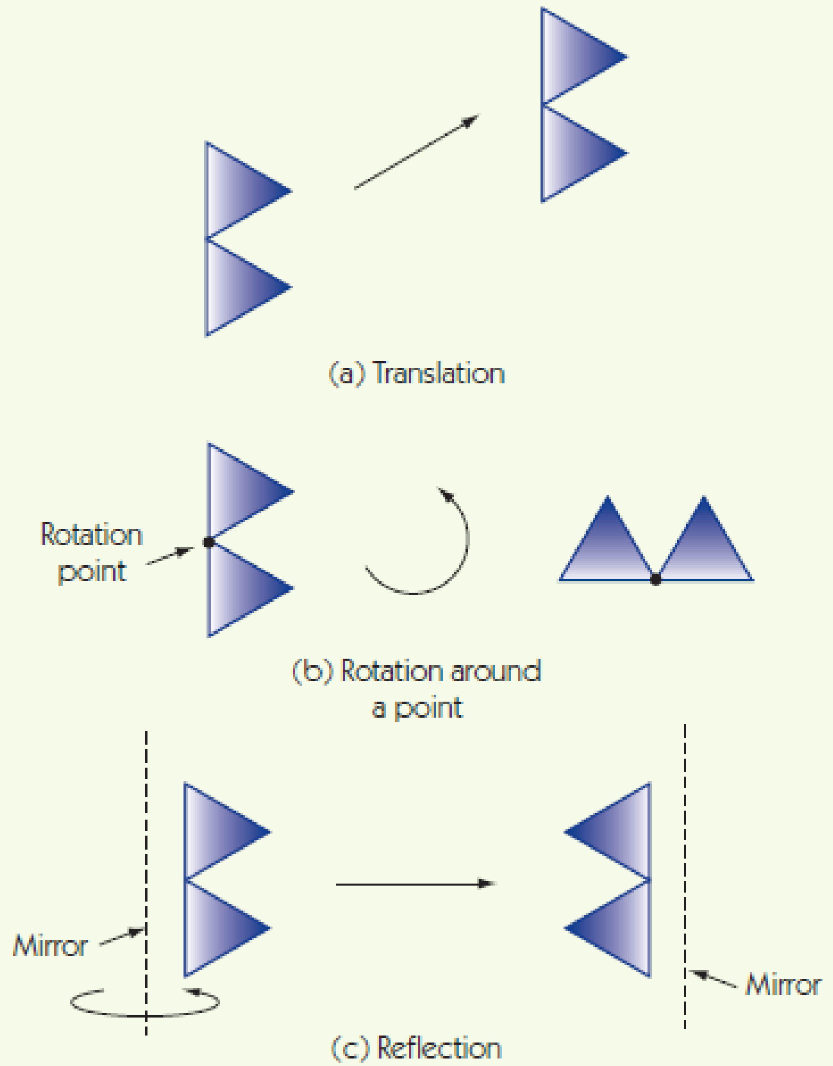
	Vertex	x	y	z	Connected to
(Origin)	V_1	0	0	0	V_2, V_3, V_4
	V_2	0	1.0	0	V_1, V_4, V_5
	V_3	1.6	0	0	V_1, V_4, V_5
	V_4	0.7	0.5	0.5	V_1, V_2, V_3, V_5
	V_5	1.4	1.1	0	V_2, V_3, V_4

Simple tessellation and corresponding vertex list

Computer-Generated Imagery (CGI) (continued)

- Object motion: rigid or deforming
- **Rigid motion**: movement that does not change object's shape
  - **Translation**:
    - Lateral movement, every point changes by the same amount in all three dimensions
  - **Rotation**:
    - Circular movement around a fixed point/axis
  - **Reflection**:
    - Mirror image of object

FIGURE 16.5



The three types of rigid motion

Computer-Generated Imagery (CGI) (continued)

- Implement movements using matrix multiplication
- **Transformation matrix**:
  - Defines a particular motion
  - Multiply vector for vertex position (x, y, z) by transformation matrix
  - Result is new location for that point, as a vector
- Example: move point at (x, y, z) to (x+a, y+b, z+c)
- Create motion by repeating for small a, b, c
- **Keyframing**: user gives start and end positions, computer generates the intermediate movements

Computer-Generated Imagery (CGI) (continued)

FIGURE 16.6

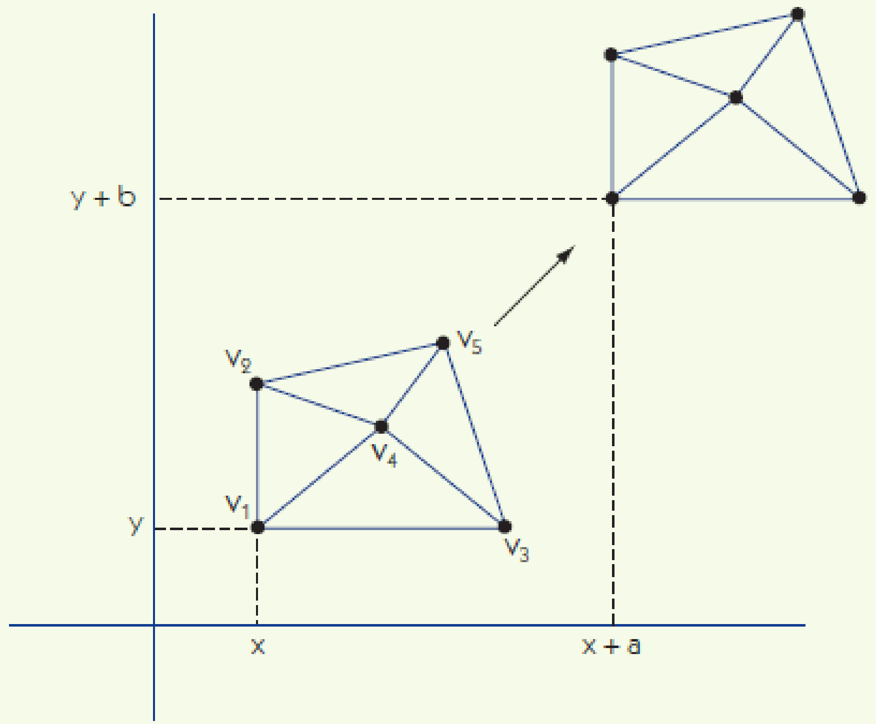
$$\begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ z+c \\ 1 \end{bmatrix}$$

Translation matrix *Current vertex coordinates* *New coordinates*

Using matrix multiplication to implement object translation

Computer-Generated Imagery (CGI) (continued)

FIGURE 16.7



Example of a translation performed on the object shown in Figure 16.4(a)

Computer-Generated Imagery (CGI) (continued)

- **Control point (animation variable)**
  - A point or axis used to control motion
- Multiple control points allow movement of parts of an object
- Example: two rotation axes
  - One rotates right circle around center point
  - One rotates both circles around external point

Computer-Generated Imagery (CGI) (continued)

FIGURE 16.8

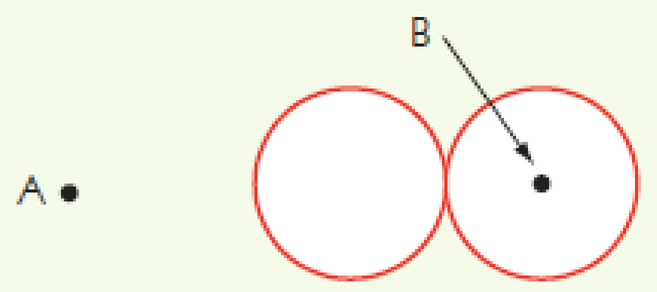


Figure-8 object containing two control points

Computer-Generated Imagery (CGI) (continued)

- **Rendering**: converting model to full 3D image
  - Lighting:
    - Fix location and intensity of all light sources
    - Determine effects on objects
  - Color shading
    - Assign colors to vertices, blend across polygons
  - Shadows
    - Account for shadows from other objects
  - Texture mapping
    - Apply variable surfaces to polygons
  - Blur: to show motion

Computer-Generated Imagery (CGI) (continued)

FIGURE 16.9

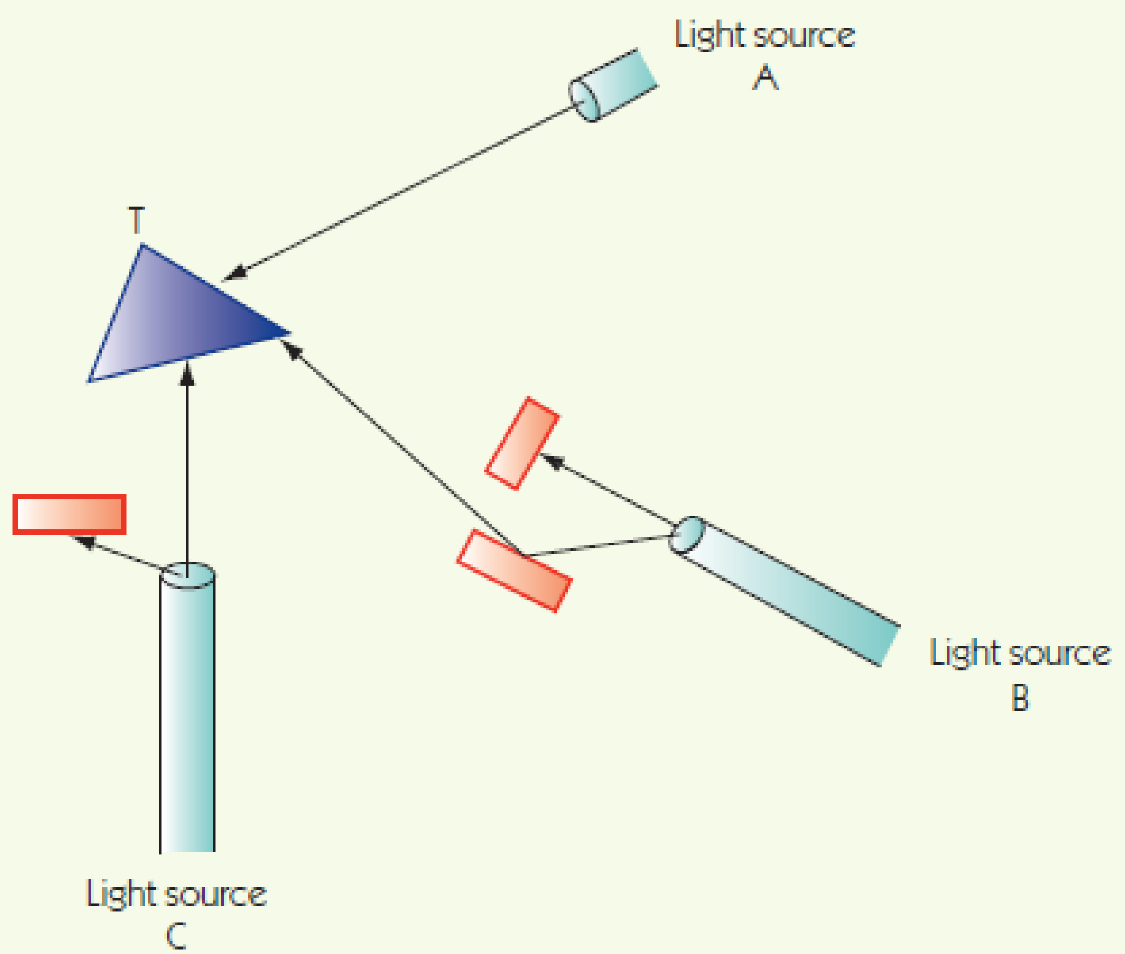
Example of a fully rendered frame

Source: Courtesy Gilles Tran

Computer-Generated Imagery (CGI) (continued)

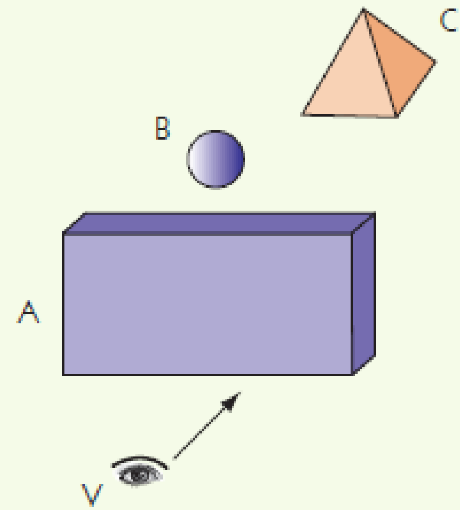
- **Ray tracing**: Most common rendering algorithm
- Repeats for each polygon in the object
  - For each light source
    - Determine the amount and direction of light from that source
  - Combine light sources and determine intensity, color shading, brightness
- Last step: convert 3D objects to a 2D image
  - Locate viewer, use geometry to determine the view

FIGURE 16.10

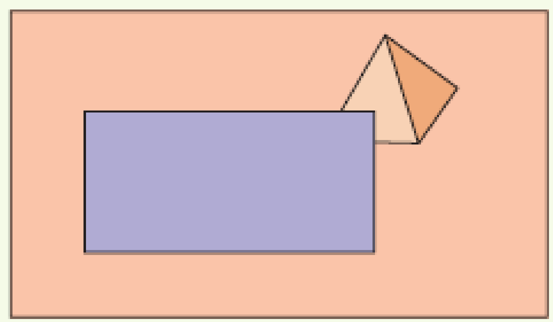


Three light sources illuminating triangle T

FIGURE 16.11



(a) Three-dimensional representation



(b) Two-dimensional representation

Converting an image to a two-dimensional representation

Video Gaming

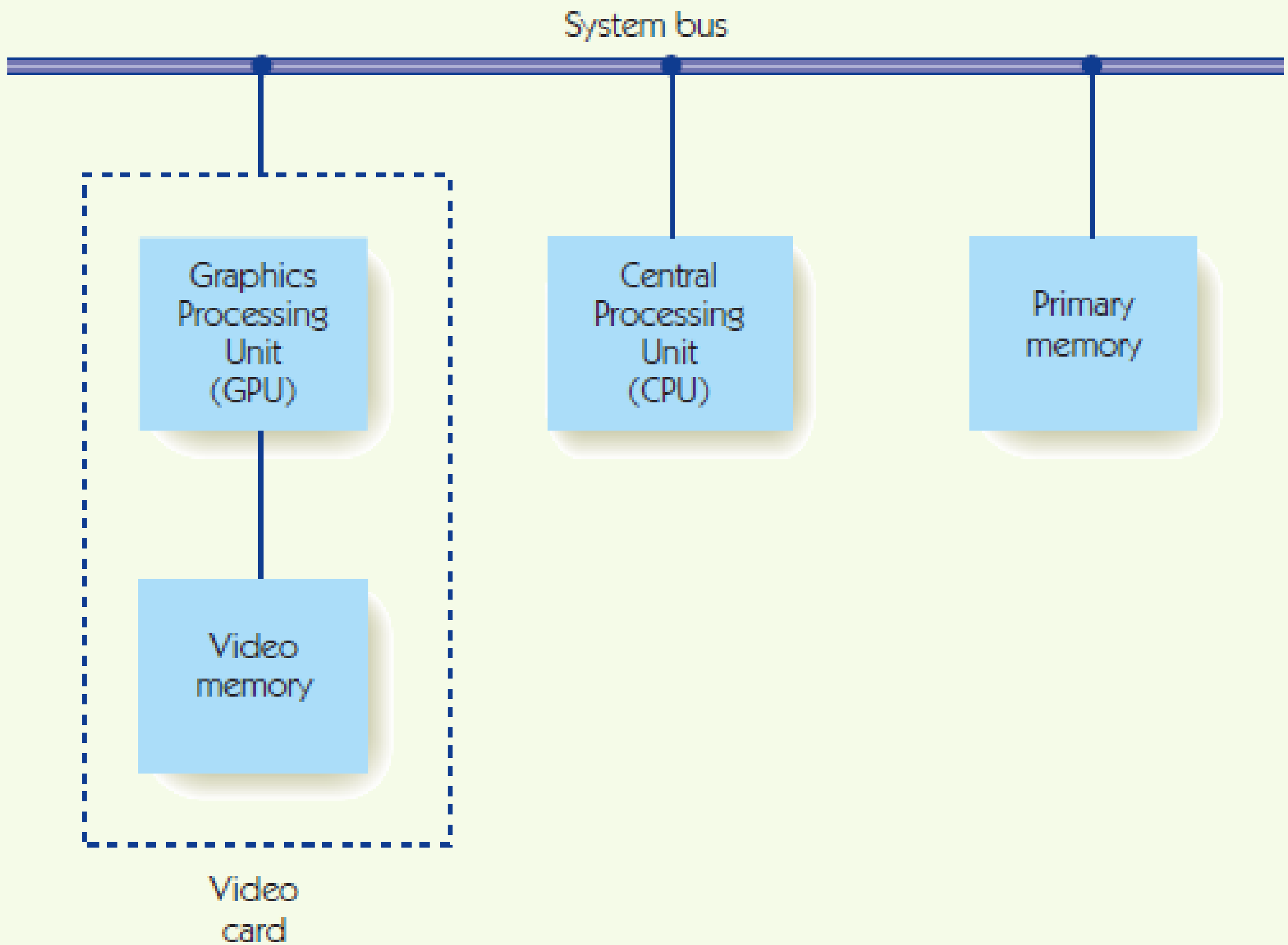
- Video game images similar to CGI
- CGI assumes a **static environment**
  - Unlimited time to produce the images
  - Created once and shown many times
- Video games are an **interactive environment**
  - User choices change what must be displayed
  - Must render images in real-time: r**eal-time graphics**

If necessary, sacrifice image quality for speed of display

Video Gaming (continued)

- **Frame rate**: image generation speed
- Computers improve frame rate with a GPU
- **Graphics Processing Unit (GPU)**
  - A separate processor optimized for CGI
  - Dedicated **video memory** on a **video card** (**graphics card)**
- Most modern computers have GPUs
  - Some have multiple GPUs (Apple iPad 2)

FIGURE 16.12



Typical architecture of a GPU and video memory

Video Gaming (continued)

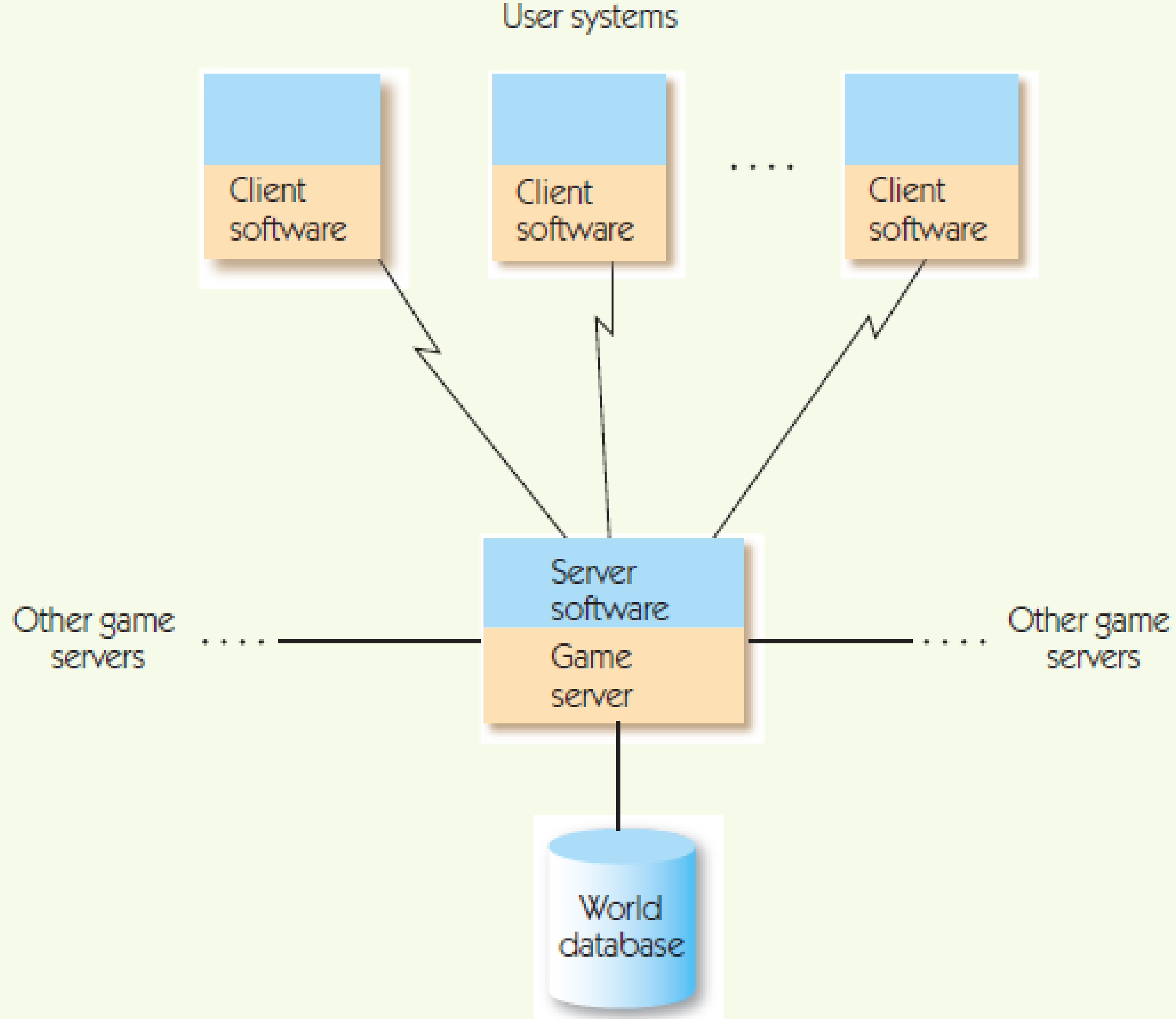
Other methods for achieving real-time graphics:

- Avoid time-consuming algorithms (e.g., ray tracing)
- Use fixed colors, with no shadows or shading
- **Culling**: first determine what is visible, and only render visible elements
- **Cut-ins:** pre-computed fully-rendered objects
  - Stored in a video library
  - Dropped into the frame as needed

Multiplayer Games and Virtual Communities

- **Massively multiplayer online games (MMOG)**
  - Thousands of players
  - Simulated virtual world
  - E.g., *World of Warcraft*
- Game provided by online **game servers**
- Game world is always available and running
- Users log in to game with client software

FIGURE 16.13



Architecture of an MMOG

Multiplayer Games and Virtual Communities

MMOG game designers must perform:

- Game play management
  - Control game actions and render game images
- Registration management
  - Control user accounts, new and returning users
- Client/server protocols for multiple servers and thousands of clients
- Security to ensure authorized access only
- Database design store huge amounts of data, retrieve quickly

Multiplayer Games and Virtual Communities

Virtual Communities:

- **Noncompetitive MMOG**, or **metaverse**
- Goal is not to score points or win, but to explore and interact, create, and engage in virtual commerce
- E.g., *Second Life*, by Linden Labs
  - Several thousand servers
  - 100 trillion bytes of data
  - 21 million “residents”
  - ~54,000 online at any one time

Summary

- As computing power has grown, so has the use of computers for entertainment, games and movies
- CGI allows for realistic animation
- The graphics pipeline includes object modeling, object motion, and rendering and display
- Objects are commonly modeled as tessellations of polygons in three dimensions
- Motion is created through matrix multiplication of transformation matrices with vertices of an object

Summary (continued)

- Rendering must apply color, shading and texture to images
- Ray tracing is a common technique for rendering
- Video games must render images in real-time
- GPUs perform video operations to speed up render time
- MMOGs are online games, large scale computing
- Virtual communities like *Second Life* are noncompetitive MMOGs