
CMSC 498M: Chapter 12

Final Review

Chapter 12, Slide 1
Copyright © David Mount and Amitabh Varshney

Semester Review

What we covered:

- Introduction (history, game architecture) (2 lectures)
- Game Architecture (1 lecture)
- OpenGL Overview (3 lectures)
- Game and Graphics Engines (2 lectures)
- Modeling for Games (3 lectures)
- Animation (1 lecture)
- GPU Programming and Shaders (2 lectures)
- Artificial Intelligence for Games (4 lectures)
- Networking and Multiplayer Games (3 lectures)
- Sound for Games (1 lecture)
- Game Physics (3 lectures)

Chapter 12, Slide 2
Copyright © David Mount and Amitabh Varshney

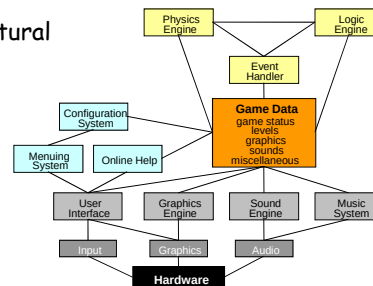
Introduction

History:

- Timeline
- Game genres
- Game demographics

Game Program Design:

- Basic elements of game design
- Game architecture - basic structural elements and relations
- Sample game program structure and timing
- Scene graphs



Chapter 12, Slide 3
Copyright © David Mount and Amitabh Varshney

Interactive graphics and Geometry

OpenGL:

- Graphics Pipeline and General Structure
- OpenGL Drawing Primitives
- GLUT: Window Manipulation and User Interaction
- OpenGL Transformations and 3D viewing
- Lighting, shading, texture mapping in OpenGL

Geometry for graphics:

- Affine Geometry
- Coordinate systems
- 2-d and 3-d Affine transformations
- Transformations in OpenGL

Graphics architecture and shaders:

- Vertex and fragment shaders
- GLSL programming language



Chapter 12, Slide 4
Copyright © David Mount and Amitabh Varshney

Graphics Engines and Ogre

Game and Graphics Engines:

- Engine core functionality
- Basic engine architecture

The Ogre 3D game engine:

- Ogre scene graph
- Ogre subsystem overview
- Examples

Game/graphics engine architecture:

- Core engine systems
- Scene graphs
- Renderers



Chapter 12, Slide 5
Copyright © David Mount and Amitabh Varshney

Geometric Modeling

Geometric modeling:

- Modeling forms (meshes, blobs, parametric)
- Bezier and B-Spline curves and surfaces
- Modeling strategies for games
- Geometry instancing

Level of detail:

- Static and dynamic LOD
- LOD based on: Distance, size, eccentricity, velocity, depth of field

Animation:

- Kinematics, dynamics, interpolation
- Keyframing
- Skeletal animation
- Skin and bones



Chapter 12, Slide 6
Copyright © David Mount and Amitabh Varshney

GPUs and Programmable Shaders

GPU Architecture:

- Rendering pipeline
- Vertex and fragment processing

Shaders:

- Vertex shaders
- Fragment shaders

GLSL Shader Language:

- Basic language structure
- Compiling/linking a GLSL program
- Vertex/Fragment shader examples
 - Phong shading
 - Toon shading
 - Normal mapping



Chapter 12, Slide 7
Copyright © David Mount and Amitabh Varshney

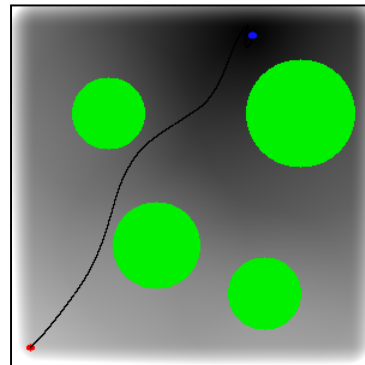
Artificial Intelligence

Basic AI for games:

- AI Basics
- Agents
- Rule-based Approaches
- Finite-State Machines

Path finding and Motion Planning:

- Configuration spaces
- Waypoints and path planning algorithms
- Dynamic planning
- Potential-based path planning
- Flocking
- Particle systems



Chapter 12, Slide 8
Copyright © David Mount and Amitabh Varshney

Networking, Online Games, and Sound

Networking:

- Networking: Protocols and packets
- OSI Network Layers: Physical, Data Link, Network, Transport, ...
- Socket programming and RakNet

Online Games:

- MMOG: latency and consistency
- Cheating and countermeasures

Sound:

- Basic definitions
- Sound engines and the aural pipeline
- Sound design



Chapter 12, Slide 9
Copyright © David Mount and Amitabh Varshney

Physics

Basics:

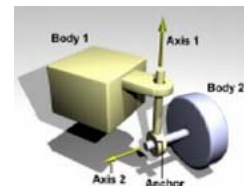
- Basic physical quantities
- Kinematics (for particles and rigid bodies)

Rotation and force:

- Geometry and transformations
- Rotations (Euler angles and quaternions)
- Force: Springs, dampers, buoyancy, friction

Kinetics and integration:

- Force and torque
- Kinetics and the laws of force and motion
- Collision detection and response
- Numerical integration (Euler)
- Motion constraints



Chapter 12, Slide 10
Copyright © David Mount and Amitabh Varshney

Student Presentations

Greg Benjamin	BSP Trees
Steven Dobek	Pygame Engine
Trevor Hill	Cloth Simulation
Andrew Ferguson	Partially Visible Sets and Partially Audible Sets
Peter Enns	Automated Terrain Generation
Tiffany Chao	Unity 3D Engine
Todd Watson	Game Cheating Tools
Jess Gonzales	Sound for Games
William Goh	Designing Imperfect AI
Sina Iman	Network-Based Cheating
Richard Johnson	OpenTK
Nick Van Aarsten	Smooth Animation and Camera Control
Chris Morton	Modeling in Blender
Patrick Bracken	XNA Game Studio
Mike Meyers	XNA Game Studio
Damon Earp	(TBA)
Douglas Hummel	(TBA)

Chapter 12, Slide 11
Copyright © David Mount and Amitabh Varshney

What We Didn't Cover

Topics we didn't cover:

- 2-D game programming (sprites)
- DirectX
- Story telling and game design
- Business issues
- Software development tools and project management
- Artistic elements and tools (Maya and 3ds Max)
- User-interface issues
- Deeper aspects of game design (flow, balance, rewards)
- Ethical issues (and game ratings)
- Serious games

Chapter 12, Slide 12
Copyright © David Mount and Amitabh Varshney