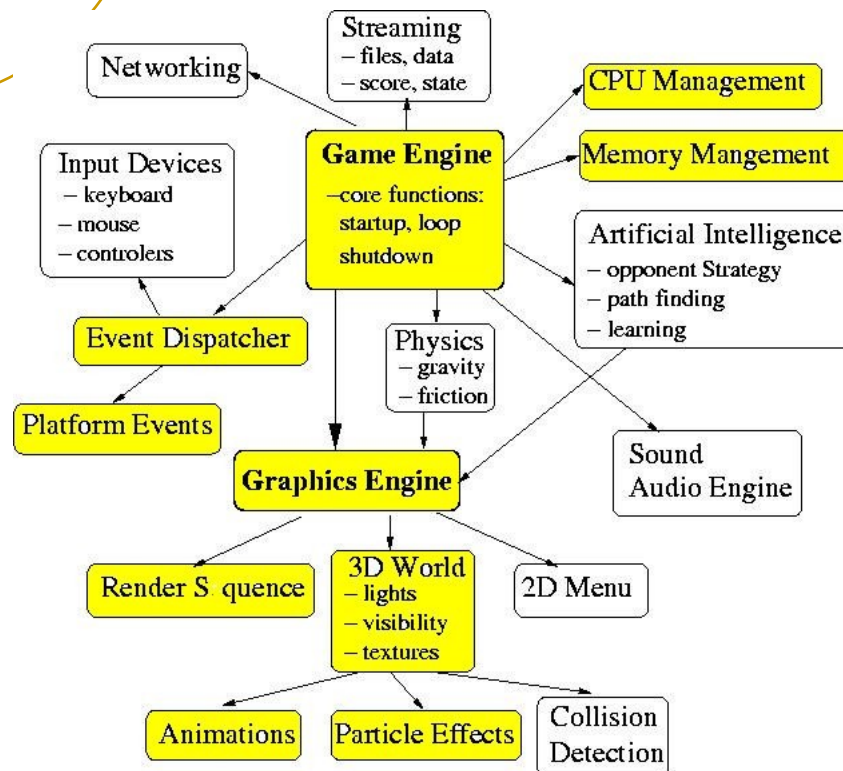
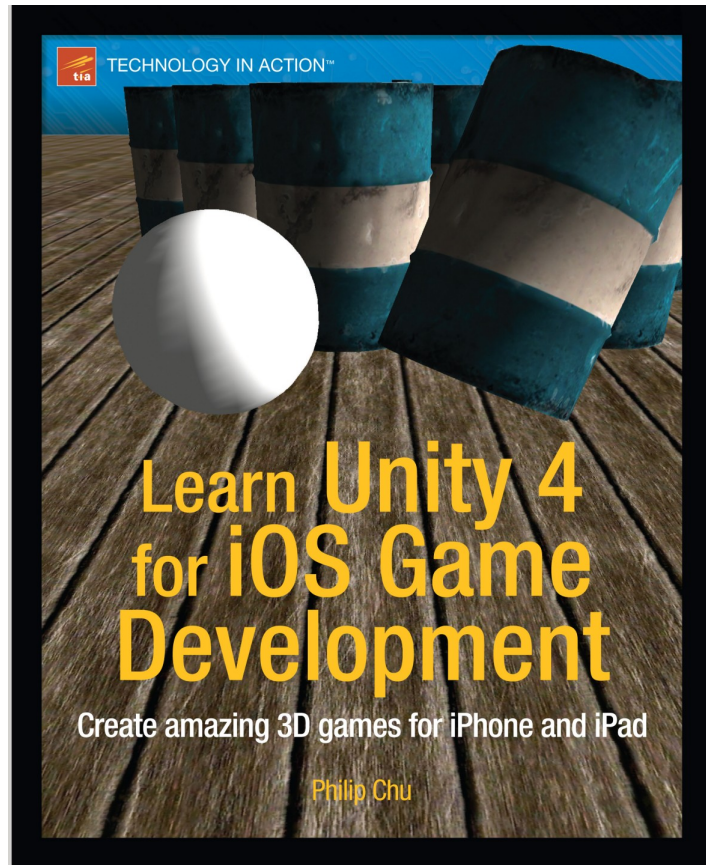


CS 38101 Game Programming, Fall 2014



Recommended Text

- Learn Unity 4
for iOS Game
Development,
Philip Chu,
2013,
Apress,
ISBN-13 (pbk): 978-1-4302-4875-0
ISBN-13 (electronic): 978-1-4302-4876-7,
www.apress.com.



Required Software

- Unity 4
- Studica

Course Outline

- Chapters 1-9 of recommended text.
- Material provided by the instructor.
- Weekly exercises and occasional quizzes class material.
- Around mid-semester, choose a project in Unity .
- Grades: 40% weekly exercises; 40% (group) project :planning, progress reports and presentations; 10% class participation, 10% attendance.

[Course Learning Objectives]

- Demonstrate the ability to program in event driven environments: to use game engine scripting; to build a simple game or interactive teaching tool; to program lighting and sound effects; to program collisions and other object interactions; and to use basic AI techniques.

Handing in Exercises

- Hand in weekly exercise by uploading them to the svn. The procedure will be explained later. Add a statement to the Blackboard assignment that you have completed the exercise. The exercise will not be grade without a response to the assignment in Blackboard.
- Do not upload spurious files (meshes, music, etc) when handing in exercise. Disregarding this will cost your points. Name exercises Unity1, Unity2, etc and put them in separate folders.

[Exercise & Project Submission]

- Projects and Exercises are submitted to an SVN repository at `classes.cs.kent.edu/courses/cs38101/svn/F2014`.
- SVN directly available on Linux and Macs
- For windows use TortoiseSVN from <http://tortoisesvn.net/>

Game Engines

- A game engine is a software system designed for the creation and development of video games. (Wikipedia).
- The core functionality typically provided by a game engine includes
 - rendering engine (“renderer”) for 2D or 3D graphics
 - a physics engine or collision detection (and collision response),
 - sound,
 - *scripting*,
 - animation,
 - artificial intelligence,
 - networking,
 - streaming,
 - memory management,
 - threading,
 - localization support,
 - and a scene graph.

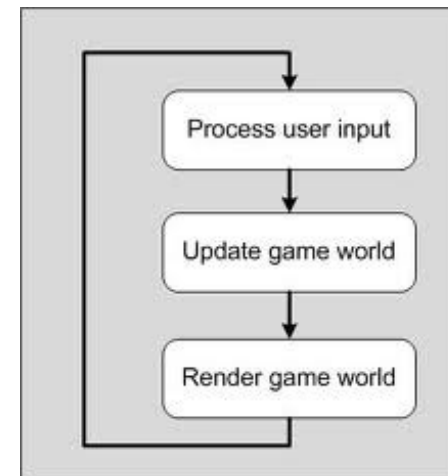
Game Engines

- Game Engines Examples:
 - Unreal Engine (Unreal Tournament)
 - Unity
 - CryEngine 2,3
 - Torque Game Engine
 - Second Life
- Game Engines are typically written (frequently in C++ as a large number of classes and that provide the functionality needed to define a game.
- Unity Runtime Classes

Game Engines

- Since the flow of the program (game) depends on the actions of the player, game engines are **event driven**:

- Mouse movements.
- Key presses.
- Avatar collision.
- Avatar near a bot.
- Network activity.
- The contents of a environment definition file.



Game Loop

- Programming a game engine is therefore a matter of programming what happens when an event occurs.

Game Engines

- To make the event programming (which defines the game) easy to create and modify, game engines often provide a high level scripting language that provide nearly direct access to the functions defined by the underlying game classes.
- Typically at the time the script program is defined, it is parsed and stored as a byte code.
- Loosely, the byte code is a list of indices into an array of functions together with the function's parameters.

Unity3d

- High quality game engine.
- Very nice game editor
 - Terrain editor
 - Scripting in C#, JavaScript, Boo (dialect of Python)
 - Intuitive scene graph
 - Intuitive disk file system
 - Great documentation

Concentration in Game Programming

- CS 38101 Introduction to Game Programming
- CS 47101 Computer Graphics
- CS 48101 Game Engine Concepts
- CS 48102 Game Development Practicum

Program Goals

- Preparation for working in
 - Game Development
 - Simulation Development
 - Educational Learning Environment Development
- Skills required
 - Understanding game engine programming
 - Game engine construction
 - Project Planning
 - Working with animators and artists

Program Goals

- Based on IGDA Curriculum Framework
- Game Programming Curriculum
 - Specifics and how well we cover the curriculum.
- Prepare students to work as (serious)game programmers.
 - Game programmers usually do not develop game engines.
 - They are usually trouble shooters who must know how to debug code, fix problems, develop new applications quickly.