CS 74.42A Course Outline as of Fall 2017

CATALOG INFORMATION

Dept and Nbr: CS 74.42A   Title: GAME DEVELOPMENT 1
Full Title: Game Development 1
Last Reviewed: 11/26/2018

Units		Course Hours per Week		Nbr of Weeks	Course Hours Total	
Maximum	3.00	Lecture Scheduled	3.00	17.5	Lecture Scheduled	52.50
Minimum	3.00	Lab Scheduled	0	8	Lab Scheduled	0
		Contact DHR	0		Contact DHR	0
		Contact Total	3.00		Contact Total	52.50
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 105.00   Total Student Learning Hours: 157.50

Title 5 Category: AA Degree Applicable
Grading: Grade or P/NP
Repeatability: 00 - Two Repeats if Grade was D, F, NC, or NP
Also Listed As:
Formerly:

Catalog Description:
An introduction to game development for students interested in the technical aspects of making video games. This course provides the technical and mathematical background to develop a 2D arcade game using JavaScript and HTML 5, with brief introduction to other high-level languages and their applicability to game development. During the course, students will be introduced to modern game platforms, and the effect of their differences, evolution, and limitations, on game programming. In addition, students will learn the rudiments of game design and the common work flow practices within the industry.

Prerequisites/Corequisites:

Recommended Preparation:
Course completion of CS 110A and/or programming experience

Limits on Enrollment:

Schedule of Classes Information:
Description: An introduction to game development for students interested in the technical aspects of making video games. This course provides the technical and mathematical

background to develop a 2D arcade game using JavaScript and HTML 5, with brief introduction to other high-level languages and their applicability to game development. During the course, students will be introduced to modern game platforms, and the effect of their differences, evolution, and limitations, on game programming. In addition, students will learn the rudiments of game design and the common work flow practices within the industry. (Grade or P/NP)
Prerequisites/Corequisites:
Recommended: Course completion of CS 110A and/or programming experience
Limits on Enrollment:
Transfer Credit: CSU;UC.
Repeatability: Two Repeats if Grade was D, F, NC, or NP

ARTICULATION, MAJOR, and CERTIFICATION INFORMATION:

			Effective:	Inactive:
AS Degree:	**Area**		Effective:	Inactive:
CSU GE:	**Transfer Area**		Effective:	Inactive:
IGETC:	**Transfer Area**		Effective:	Inactive:
CSU Transfer:	Transferable	Effective:	Spring 2011	Inactive:
UC Transfer:	Transferable	Effective:	Fall 2020	Inactive:

CID:

Certificate/Major Applicable:
Both Certificate and Major Applicable

COURSE CONTENT

Outcomes and Objectives:
Upon completion of the course, students will be able to:
1. Discuss the history of computer programming languages, in particular the trend of managed code as a safer alternative to native code.
2. Practice the rudiments of Linear Algebra, using vectors and matrices to solve numerous analytical/scientific problems.
3. Create 2D arcade-style game simulations such as Space Invaders or Pac-Man.
4. Identify best practices for memory-constrained devices such as gaming consoles and discuss the particular issues of developing software for a console as opposed to a personal computer.

Topics and Scope:

I. JavaScript Language Fundamentals
   A. Introduction to JavaScript and HTML5
      1. History
      2. Interpreted vs. Native code
      3. Important JavaScript frameworks
   B. Project design and organization
      1. Creating project file and directory tree
      2. Implementing a simple JavaScript application
      3. Running and debugging in Firefox and Firebug
      4. Statements

C. Programming language fundamentals
    1. Data types
    2. Variables
    3. Operators
    4. Functions
    5. Garbage collection
    6. Strong vs. weak typing
D. Object-oriented programming (review)
    1. Class definitions
    2. Object instantiation
    3. Inheritance
    4. Encapsulation
    5. Polymorphism
E. Containers
    1. Arrays and lists
    2. Loops and iterating
F. Exceptions
G. Delegates and Events
H. String manipulation

II. Mathematics
  A. Cartesian Coordinate Systems
  B. World space, object space, camera space
  C. Vectors
    1. Addition and subtraction
    2. Dot and cross product
    3. Normalization
    4. Distance formula
  D. Matrices
    1. Matrix dimension and notation
    2. Transposition
    3. Scalar and matrix multiplication
    4. Row vs. column matrices
    5. Linear transformation
      a. Rotation
      b. Translation
      c. Scale
      d. Orthographic projection
  E. Physics simulation
    1. Numerical representation and inaccuracy
    2. Fixed vs. variable time step
    3. Performance considerations

III. Game Design and Development
  A. Cultural and function definition of games
    1. Serious game development
    2. Game vs. puzzles vs. play
  B. Rudiments of game design
    1. History
    2. Player feedback
    3. Risk vs. Reward
    4. Game vs. Simulation
  C. Development processes
    1. Scrumm

        a. User stories
        b. Tasks
        c. Prioritization
    2. Iterative design of game mechanics
  D. Documentation
    1. Concept document
    2. Game design document
    3. Technical design document

IV. Game Development Technical Aspects
  A. User input
    1. Gamepad
    2. Keyboard
    3. Mouse
    4. Filtering and other forms of signal processing
  B. Graphics
    1. Digital image representation
    2. Sprites and movement
  C. Audio
    1. Digital audio representation
    2. Streaming audio
    3. Audio event prioritization
  D. Game Engine Architecture
    1. The game loop
        a. Fixed
        b. Variable time step
    2. Separation of simulation and presentation
  E. Collision Detection
    1. What is collision detection
    2. Responding to collision detection
  F. Publication
    1. PC publication
    2. Publication on consoles
  G. Lessons from the Underground: DIY/Indie techniques
    1. Agile methodologies for content generation
    2. Intellectual property and copyright issues
    3. Free resources
  H. XNA Content Pipeline
    1. What are assets
    2. Importing assets into applications

Assignment:

1. Read approximately 25-30 pages a week
2. Prepare three written documents that closely mirror the documentation process used in the games industry: a concept document, game design document, and technical design document. (3-7 pages each)
3. Weekly programming assignments that solve particular technical challenges
4. Midterm
5. Final Project: a working 2D arcade game simulation that demonstrates the use of the techniques developed in this class

Methods of Evaluation/Basis of Grade:

Writing: Assessment tools that demonstrate writing skills and/or require students to select, organize and explain ideas in writing.

Concept document, game design document, technical design document	Writing 10 - 20%

Problem Solving: Assessment tools, other than exams, that demonstrate competence in computational or non-computational problem solving skills.

Programming assignments -- technical challenges	Problem solving 20 - 40%

Skill Demonstrations: All skill-based and physical demonstrations used for assessment purposes including skill performance exams.

Programming assignments with C#; Final project	Skill Demonstrations 30 - 60%

Exams: All forms of formal testing, other than skill performance exams.

Mid-term exam: multiple choice, true/false, matching, short answer	Exams 10 - 20%

Other: Includes any assessment tools that do not logically fit into the above categories.

None	Other Category 0 - 0%

Representative Textbooks and Materials:

Hall, Joseph. XNA Game Studio Express: Developing Games for Windows and the Xbox 360. Course Technology PTR, 2007. (classic)

Lobao, Alexandra Santos. Beginning XNA 3.0 Game Programming: From Novice to Professional. Apress, 2009.

Nitschke, Benjamin. Professional XNA Game Programming. Wrox, 2008. (classic)

Reed, Aaron. Learning XNA 3.0: XNA 3.0 Game Development for the PC, Xbox 360, and Zune. O'Reilly Media, 2008. (classic)

Instructor prepared materials