**Las Positas College**
3000 Campus Hill Drive
Livermore, CA 94551-7650
(925) 424-1000
(925) 443-0742 (Fax)

**Course Outline for CIS 48**

**INTRO TO GAME PROGRAMMING**

**Effective: Spring 2008**

I. CATALOG DESCRIPTION:
CIS 48 — INTRO TO GAME PROGRAMMING — 3.00 units

Want to Play? You have played plenty of games. Now it is time to create your own! You are interested in programming games, but you don't know where to begin. This course covers the basics of game programming with an emphasis on hands-on development of games using a Rapid Application Development prototyping tool such as Dark Basic or BlitzPlus. These tools, based on the Basic language, feature powerful graphics engines, and make it possible to demonstrate high-level subjects using a minimum amount of code. This first programming course provides experience and skills writing every element of your first video game—from graphics and animation to sound and music. Programming experience is not required to get started. Although this course has a programming focus, other topics briefly covered will include the history of computer/video game technology, game genres and design principles, and the social impact of games. Students may enroll in Computer Information Systems 48 and/or CS 48 for a total of 2 times.

2.50 Units Lecture 0.50 Units Lab

**Grading Methods:**
Letter or P/NP

**Discipline:**

| | MIN |
|---|---|
| **Lecture Hours:** | 45.00 |
| **Lab Hours:** | 27.00 |
| **Total Hours:** | 72.00 |

II. NUMBER OF TIMES COURSE MAY BE TAKEN FOR CREDIT: 2

III. PREREQUISITE AND/OR ADVISORY SKILLS:

IV. MEASURABLE OBJECTIVES:

**Upon completion of this course, the student should be able to:**

A. Examine and categorize the significant events in the development of the Electronic Game Industry;
B. Describe the basic principles of computer game programming;
C. List and develop steps that are involved in program development;
D. Design correct and efficient algorithms to solve programming problems;
E. Write pseudocode to solve programming problems;
F. Draw flowcharts to indicate the flow of algorithms;
G. Evaluate the basic process of creating a game;
H. Demonstrate an understanding of programming theory;
I. Display an understanding of game and level design;
J. Identify and discuss game genres and playing perspectives;
K. Use skills gained in the course to effectively present a prototype game
L. Construct small programs using various elements, such as variables, I/O, if-else, loops, functions, expressions, and parameters.

V. CONTENT:
A. Programming Concepts
1. History of Computation and programming languages
2. Types and purposes of programming languages
B. Program Development
1. The Notion of an Algorithm
2. Top-Down design
3. Pseudocode
4. Flowcharts
5. Design of correct, concise, and effective algorithms
6. Control flow, basic logic, and Boolean expressions
7. Usage of data types, arithmetic expressions, input/output, functions and arguments
8. Effective methods for checking and debugging programs
9. Programming in an integrated development environment (IDE)
C. The Basics of BASIC
1. Getting Started
2. Getting to Know BASIC interface

3. Variables, types, Loops, Functions, Arrays
4. The Style Factor

D. Structure and Syntax
1. Declaring, Using Variables
2. Input
3. Conditionals, IF...THEN…ELSE
4. Logical Operators
5. GOTO command

E. Loops, Functions, Arrays, and Types
1. Understanding Loops: For..Next, While, Wend, Repeat..Until
2. Understanding functions: scope, when to use
3. Understanding Arrays: Multi-Dimensional
4. Using Types: Coordinate Systems
5. Types and purposes of programming languages

F. Style Factor
1. Developing Style: white space and indentation
2. Comments: Pre-program, main program, function comments
3. Function and Variable Names: Names, Names Format

G. Getting Graphical
1. Beginning Graphics
2. Images: Loadimage, drawimage, createimage, maskimage
3. Colors: RGB, Color, Cls and ClsColor
4. Page Flipping and Pixel Plotting
5. Basic Image Programming
6. Animation
7. Collision Detection

H. Animation
1. Using bitmaps in animation
2. Making bitmaps
3. Displaying movement

I. Collision Detection
1. Basic collisions
2. Bounding Circles
3. Bounding Boxes
4. Pixel-Imperfect Collisions
5. Pixel-Perfect Collisions

J. Handling Input
1. Keyboard: KeyDown, KeyHit
2. Mapping the Mouse to the Screen
3. Handling Joystick input

K. Sounds and Music
1. Sounds: Loading Sounds, Playing Sounds
2. Music: Channels and PlayMusic()

L. Artificial Intelligence
1. Random Numbers: millisec() timer
2. Chasing and Evading

VI. METHODS OF INSTRUCTION:
A. **Lecture** -
B. Hands-on exercises in the laboratory
C. **Discussion** -
D. **Demonstration** -

VII. TYPICAL ASSIGNMENTS:
A. Convert a series of decimal numbers into their binary, octal, and hexadecimal equivalents. B. Using pseudocode and flow charts, create the programming logic that will allow a user to enter a starting and ending value and display the even integer values (display the odd integer values). C. Using pseudocode and flow charts, create the logic that will read in records from a sequential data file and process the records (emphasize that the file has an “unknown” number of records and what type of loop you would use).

VIII. EVALUATION:
A. **Methods**

1. Exams/Tests
2. Quizzes
3. Home Work
4. Other:
    a. Methods
        1. Programming assignments—All assignments are evaluated on correctness, completeness, timely submission, documentation (both internal and external), style, structure, testing, and reporting of results.
        2. Written homework
        3. Exams and/or quizzes
        4. Final examination

B. **Frequency**

1. Frequency
    a. At least two midterm examinations, or one midterm examination and several quizzes
    b. Additional midterm examination(s) and/or quizzes
    c. Comprehensive final examination
    d. Programming assignments to cover each topic within the course content (contents can be combined).
    e. One homework assignment each week to cover that week’s discussion. This might be written or a programming assignment.

IX. TYPICAL TEXTS:

1. Adams/Rollings *Fundamentals of Game Design.*, Prentice Hall, 2007.
2. Harbour, Jonathan *DarkBASIC Pro Game Programming.* 2nd ed., Course Technology, 2006.
3. Finney, Kenneth *3D Game Programming All In One.* 2nd ed., Delmar/Cengage Learning, 2007.

X. OTHER MATERIALS REQUIRED OF STUDENTS: