CS 42 Course Outline as of Fall 2020

CATALOG INFORMATION

Dept and Nbr: CS 42 Title: INTRO TO GAME CODING
Full Title: Introduction to Game Coding
Last Reviewed: 11/26/2018

Units		Course Hours per Week		Nbr of Weeks	Course Hours Total	
Maximum	4.00	Lecture Scheduled	4.00	17.5	Lecture Scheduled	70.00
Minimum	4.00	Lab Scheduled	0	8	Lab Scheduled	0
		Contact DHR	0		Contact DHR	0
		Contact Total	4.00		Contact Total	70.00
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 140.00 Total Student Learning Hours: 210.00

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:	CS 74.42

Catalog Description:
This course introduces students to the design, development, and coding of simple graphical computer-based games. During the course, students will be introduced to various game engines and development environments. Students will gain experience working individually and in a team environment. Emphasis is placed on engaging players through compelling application of game mechanics, dynamics, and aesthetics, as well as on playtesting and iterative development to ensure user-centered design goals are met.

Prerequisites/Corequisites:

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

Limits on Enrollment:

Schedule of Classes Information:
Description: This course introduces students to the design, development, and coding of simple graphical computer-based games. During the course, students will be introduced to various game engines and development environments. Students will gain experience working individually and

in a team environment. Emphasis is placed on engaging players through compelling application of game mechanics, dynamics, and aesthetics, as well as on playtesting and iterative development to ensure user-centered design goals are met. (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:

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

Student Learning Outcomes:
At the conclusion of this course, the student should be able to:

1. Understand and apply systems-level thinking and game development methodology best practices to the design and development of simple graphical computer-based games.
2. Develop team-based game creation skills that cover game logic sequencing; storyboarding; artistic and technical coordination; and understanding of technical requirements and limitations.

Objectives:
At the conclusion of this course, the student should be able to:

1. Explain factors that motivate game players and apply those to the design of games that satisfy the needs of players. This includes the design of a logical sequence of game rules, play flow, and interaction opportunities.
2. Develop game coding skills to implement simple versions of games they design. Desired functions include: graphical, interactive user interface; sprite creation, collision detection, applied simulation of physical forces; classes, object instantiation; and modular implementation.
3. Determine that games are of measurable high quality and error-free through playtesting, iterative development, and an adherence to best practices in quality assurance, including a proper and documented testing process.
4. Work independently and in teams using an iterative process to prepare and create Game Design Documents, storyboards, game assets and scripting to create game simulations.
5. Utilize the concepts of game mechanics, dynamics, and aesthetics to discuss game play of games of different genres on varied platforms.

Topics and Scope:

I. Game Development Fundamentals
- A. Introduction to systems thinking
  - 1. Flow control and diagramming
  - 2. Visual scripting systems
- B. Introduction to scripting
  - 1. Variables and data types
  - 2. Expressions and operators
  - 3. Control structures
  - 4. Functions
  - 5. Objects and classes
  - 6. Events and triggers
- C. Mathematics and physics fundamentals
  - 1. Cartesian coordinate systems
  - 2. World space, object space, camera space
  - 3. Vectors, forces and physics simulations
- D. Development methodologies
  - 1. Unified and agile processes
  - 2. Iterative design
  - 3. Human-centered design
  - 4. Design patterns
  - 5. Project management and quality assurance

II. Considerations of Game Creation
- A. Types of games
- B. Game mechanics, dynamics, and aesthetics
  - 1. Mechanics of gameplay
  - 2. Dynamics of gameplay
  - 3. Aesthetics of gameplay
- C. Motivational and emotional aspects of playing games
- D. Generic game design rules
- E. Technologies, platforms, tools
- F. Game development frameworks and engines

III. Game Development Process
- A. Setting goals for the game
- B. Developing a storyboard and designing gameplay
- C. Drafting a Game Design Document (GDD)
- D. Iterative prototyping
- E. Stakeholder feedback loop and playtesting
- F. Features, functions, and program components
  - 1. Game loop
  - 2. Assets
  - 3. User input
  - 4. Sprites and collisions
  - 5. Audio representation
- G. Game rules
- H. Navigation and wayfinding
- I. Graphical user interfaces

IV. Supporting Disciplines
- A. Game playtesting and validation
  - 1. Quality assurance testing plan
  - 2. Bug tracking and resolution

B. Working as a team
C. Productization and packaging
D. Art work

Assignment:

1. Read approximately 25-30 pages a week
2. Prepare 1-2 written Game Design Document(s) (GDD) that closely mirror the documentation process used in the field. (3-7 pages each)
3. Regular group discussion contributions (0 - 12) such as:
   A. Play logic and navigation design
   B. Storyboarding, player motivation
   C. Theories of game development
   D. Design and implementation of game logic (rules and programmatic navigation)
   E. Use of design patterns
   F. Graphical user interface design
   G. Implementation technologies and validation
4. Game creation and scripting assignments that solve particular technical challenges (6 - 12)
5. Midterm and final examinations that evaluate critical thinking skills such as:
   A. Game requirements analysis and validation
   B. Game storyboard design and implementation
   C. Software games, techniques, and technologies
   D. Tools and techniques used in the software industry
6. A final class project that includes topics such as:
   A. Selection of a viable, realistic project
   B. Interviews with players and project stakeholders to collect requirements and risks
   C. Creation of a comprehensive software-based game
   D. Summary of playtesting findings and suggested improvements for future versions

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.

Game design document(s)	Writing 10 - 30%

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

Game creation and scripting assignments	Problem solving 20 - 40%

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

Final project	Skill Demonstrations 30 - 60%

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

Midterm and final exams	Exams 10 - 30%

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

Attendance and participation, discussion contributions	Other Category 0 - 10%

Representative Textbooks and Materials:
An Introduction to HTML5 Game Development with Phaser.js. Faas, Travis. CRC Press. 2017

Blueprints Visual Scripting for Unreal Engine. Sewell, Brenden. Packt Publishing. 2015

Unity in Action: Multiplatform Game Development. Hocking, Joe. Manning Publications. 2015

Introduction to Game Design, Prototyping, and Development. Bond, Jeremy Gibson. Addison-Wesley Professional. 2014 (classic)

Invent Your Own Computer Games with Python. 4th ed. Sweigart, Al. No Starch Press. 2016