COURSE OUTLINE

Course Number	Course Title	Credits
GAM 145	Game Programming I	**3**
Hours: Lecture/Lab/Other 1/4/0	**Pre-requisite** GAM 120	**Implementation Semester & Year SP 2022**

Catalog description:
Exploration of different facets of object-oriented programming (OOP) as it relates to game development. Students examine aspects such as methods, variables, loops, lists, arrays, game managers, health systems, timers, and other relevant components. Emphasis is on programming fundamentals, efficient prototyping techniques, essential game systems, and applying unique game mechanics.

General Education Category: **Not GenEd**

Course coordinator: (Ric Giantisco, x3458, giantisr@mccc.edu)

Required texts & Other materials:
None

Course Student Learning Outcomes (SLO):

Upon successful completion of this course the student will be able to:

1. Understand the use of variables and components in games. [ILG 1-4, 10, 11; PLO 1-4, 6, 8, 9]
2. Create loops, lists, and arrays to enable efficient and successful game interactions. [ILG 1-4, 10, 11; PLO 1-4, 6, 8, 9]
3. Design methods and employ relevant parameters to organize and connect data. [ILG 1-4, 10, 11; PLO 1-4, 6, 8, 9]
4. Program objects to detect, react, and follow predefined paths. [ILG 1-5, 10, 11; PLO 1-4, 6, 8, 9]
5. Construct game managers, collection, scoring, and health systems. [ILG 1-4, 10, 11; PLO 1-4, 6, 8, 9]
6. Demonstrate proper use of syntax in the C# programming language. [ILG 1-4, 10, 11; PLO 1-3, 6, 8, 9]
7. Design and program effective game states and game managers. [ILG 1-5, 10, 11; PLO 1-5, 6, 8, 9]
8. Create player movement via translation and physics. [ILG 1-4, 10, 11; PLO 1-4, 6, 8, 9]

Course-specific Institutional Learning Goals (ILG):

Institutional Learning Goal 1. Written and Oral Communication in English. Students will communicate effectively in both speech and writing.
Institutional Learning Goal 2. Mathematics. Students will use appropriate mathematical and statistical concepts and operations to interpret data and to solve problems.
Institutional Learning Goal 3. Science. Students will use the scientific method of inquiry, through the acquisition of scientific knowledge.
Institutional Learning Goal 4. Technology. Students will use computer systems or other appropriate forms of technology to achieve educational and personal goals.
Institutional Learning Goal 5. Social Science. Students will use social science theories and concepts to analyze human behavior and social and political institutions and to act as responsible citizens.
Institutional Learning Goal 10. Information Literacy: Students will recognize when information is needed and have the knowledge and skills to locate, evaluate, and effectively use information for college level work.

Institutional Learning Goal 11. Critical Thinking: Students will use critical thinking skills understand, analyze, or apply information or solve problems.

Program Learning Outcomes for Game Programming (PLO)

1. Understand the historical development of games;
2. Describe and reference industry trends and technologies in video gaming;
3. Apply the design process to research and develop professional video game concepts;
4. Create diagrams and prototypes to specify game design concepts;
5. Create a professional sales pitch for a game concept;
6. Program game engine components such as resource management, entity-based systems, physics simulation, and user interfaces;
7. Create a custom 2-D game engine;
8. Develop skills to be a self-learner and problem solver;
9. Work effectively on interdisciplinary teams producing functioning games and levels.

Units of study in detail – Unit Student Learning Outcomes:

Unit I Programming Fundamentals [SLO 1-3, 6]

This unit introduces students to basic game programming concepts such as variables, Boolean operations and conditionals, loops, and collections. Students will develop a project that provides the player with dynamic controls to construct and design a simple 2D platformer in real time. Emphasis will be placed on designing code with proper syntax, naming conventions, and optimized conditional structures.

Learning Objectives

The student will be able to:

- Instantiate basic objects into a scene within a game engine.
- Procedurally generate visual structures within a game setting.
- Collect and organize a group of specified objects during gameplay.
- Manipulate game elements in real time with multiple, specified inputs.

Unit II Player Movement [SLOs 1-3, 6, 8]

This unit explores various techniques for applying movement to active game objects and manipulating basic physics during gameplay. Students will program game avatars to move and perform physical actions through specific inputs. Emphasis will be placed on developing appropriate solutions to specific problems and achieving accurate results. This unit will also introduce strategies for basic input mapping strategies and balancing avatar response time.

Learning Objectives

The student will be able to:

- Translate an active game object dynamically via player inputs.
- Move an active game object dynamically by applying forces and manipulating physics.
- Design a core game play loop for a simple platformer environment.
- Clamp the movement of game avatars to a specified region and vector.

Unit III Event Driven Interactions [SLOs 1-3, 6-8]

This unit focuses on developing engaging gameplay interactions through the activation of specified triggers or events. Students will engineer various mechanisms such as timers, alerts, switches, basic collision detection, areas of effect (AoE), and other relevant interactions to achieve specified results. Emphasis will be placed on designing logical progressions and constructing efficient and modular code.

Learning Objectives

The student will be able to:

- Design simple interactions using standard collision detection techniques.
- Explain and demonstrate the difference between collision and trigger detection.
- Construct game objects that can change their state based on specified conditions.

- Deploy areas of effect (AoE) to calculate real-time results based on distance and/or other active game metrics.

Unit IV **Managers and Systems [**SLOs 1-3, 5-8]

This unit examines various system types commonly found in modern video games. Students will design and implement specific systems to dynamically control and manipulate health, inventory, time, score, and other relevant in-game commodities Emphasis will be placed on optimizing scope, maintaining consistent conventions, debugging, and class inheritance.

Learning Objectives

The student will be able to:

- Create unique variables and methods to pass information between scripts.
- Construct basic health, inventory, and scoring systems for active game avatars.
- Design a simple user interface to display game statistics.
- Troubleshoot game systems using specified editor debugging tools.

Unit V **Detection & Avoidance [**SLOs 1-8]

This unit introduces basic navigation for non-playable characters (NPCs) in video games. Students will apply standard collision detection techniques to achieve specified outcomes. Emphasis will be placed on situational assessment, problem solving, and efficient application to produce optimal results.

Learning Objectives

The student will be able to:

- Detect active game objects within a specified region using collision and triggers.
- Reorient active game objects to look at and follow specified targets.
- Construct a simple point and click level navigation system using game engine tools.

Evaluation of student learning:

PROJECTS

Project 1: Collect and Grow	- Collect different power-ups to help your avatar grow and evolve!
Project 2: Duck, Dodge, and Double Jump	- Run, duck, and double jump to reach home safely!
Project 3: Interactive Level Construction	- Design and construct your own 2D platformer from scratch!
Project 4: Splash Damage	- Avoid the incoming danger and survive for as long as you can!
Project 5: The Maze	- Escape the maze before the monsters find you!

EVALUATION CRITERIA

Each project will be evaluated on several factors. The specific goals, deliverables, and requirements of each project will be identified in the description sheet for each project.

GRADING

Projects	60%
Quizzes	20%
Homework	10%
Attendance	10%
Total	100%