

Video Game Programming

PEIMS Code: : N1300994

Abbreviation: VIDEOGD2

Grade Level(s): 10-12

Award of Credit: 1.0

State Approved Innovative Course

- Districts must have local board approval to implement innovative courses.
- In accordance with Texas Administrative Code (TAC) §74.27, school districts must provide instruction in all essential knowledge and skills identified in this innovative course.
- Innovative courses may only satisfy elective credit toward graduation requirements.
- Please refer to TAC §74.13 for guidance on endorsements.

Course Description:

Video Game Programming expands on the foundation created in Video Game Design through programming languages such as: C# programming, XNA game studio, Java, and Android App. In this course, students will investigate the inner workings of a fully functional role-playing game (RPG) by customizing playable characters, items, maps, and chests and eventually applying customizations by altering and enhancing the core game code.

Essential Knowledge and Skills:

- (a) General Requirements. This course is recommended for students in Grades 10-12. Recommended prerequisite: Video Game Design. Students shall be awarded one credit for successful completion of this course.
- (b) Introduction.
 - (1) Career and technical education instruction provides content aligned with challenging academic standards and relevant technical knowledge and skills for students to further their education and succeed in current or emerging professions.
 - (2) The Information Technology (IT) Career Cluster focuses on building linkages in IT occupations for entry level, technical, and professional careers related to the design, development, support, and management of hardware, software, multimedia, and systems integration services.
 - (3) Video Game Programming expands on the foundation created in Video Game Design through programming languages such as: C# programming, XNA game studio, Java, and Android App. In this course, students will investigate the inner workings of a fully functional role-playing game (RPG) by customizing playable characters, items, maps, and chests and eventually applying customizations by altering and enhancing the core game code.
 - (4) Students are encouraged to participate in extended learning experiences such as career and technical student organizations and other leadership or extracurricular organizations.
 - (5) Statements that contain the word "including" reference content that must be mastered, while those containing the phrase "such as" are intended as possible illustrative examples.

(c) Knowledge and Skills.

- (1) The student demonstrates professional standards/employability skills as required by business and industry. The student is expected to:
 - (A) identify and demonstrate positive work behaviors and personal qualities needed to be employable;
 - (B) demonstrate skills such as building a resume related to seeking and applying for employment;
 - (C) create a career portfolio to document information such as work experiences, licenses, certifications, and work samples; and
 - (D) compare and evaluate employment opportunities in the game programming industry.
- (2) The student applies programming skills related to software development and computer programming. The student is expected to:
 - (A) develop software applications;
 - (B) analyze the basic programming structure of application and be able to debug, compile, and run an application;
 - (C) create, name, and assign values to variables;
 - (D) create custom methods that can return values and take parameters;
 - (E) apply common built-in objects and reference types;
 - (F) apply common programming statements to implement flow control, looping, and exception handling;
 - (G) create, initialize, and use collections; and
 - (H) design and create custom class-constructors and use the object-oriented techniques of inheritance, abstraction, polymorphism, and encapsulation.
- (3) The student applies game development skills. The student is expected to:
 - (A) demonstrate significant understanding of game development tools including graphic design, game engines, animation, editors, and programming
 - (B) apply core programming logic and techniques that are used in building games;
 - (C) identify the code, structure, and layout of a fully functional role-playing game;
 - (D) create and customize new game elements such as characters, items, chests, quests, and monsters;
 - (E) create enhancements to the combat engine logic with role-playing game;
 - (F) research the inner workings of the role-playing game system, for the purpose of modifying simulated game actions; and
 - (G) describe how a two-dimensional tile-based rendering and collision system works to create maps in a game.
- (4) The student applies creativity, innovation, and critical-thinking skills to video game programming methodology. The student is expected to:
 - (A) demonstrate the ability to enhance existing game program(s) by customizing screens, adding levels, adding characters, and adding graphics;
 - (B) create, design, and program original working game features;

- (C) explain how separated game logic fits together to form a cohesive game application;
 - (D) critique beta applications and provide solutions to fix bugs and ensure performance;
 - (E) conduct a self-evaluation and discuss findings with peers
 - (F) compare projects with the required established game specifics; and
 - (G) interpret technical and increasingly complex programming instructions in order and in detail.
- (5) The student applies communication and collaboration skills as an individual and as part of a team. The student is expected to:
- (A) demonstrate the concepts of the original game and conduct in-class presentations including demonstration of original game concepts;
 - (B) analyze and solve program errors individually or in teams and collaborate with classmates in problem solving and debugging program errors; and
 - (C) apply technical writing skills to explain game design concepts, document programming logic, and document development processes.
- (6) The student applies the use of appropriate and available digital tools for research and learning. The student is expected to:
- (A) review and research websites, wiki's, and blogs for appropriate content, ideas, and best practices to engage other users; and
 - (B) investigate websites to explain concepts learned and to reference coding syntax.
- (7) The student applies engineering, physics, and mathematical concepts critical to game development. The student is expected to:
- (A) discuss and describe the principles of software engineering design within complex functional games;
 - (B) apply the principles of software engineering to enhance a complex functional game including multiple movements and multiple controls;
 - (C) apply the principles of software engineering within a complex fully-functional game/bug free program;
 - (D) reverse engineer existing game functionality to understand game design; and
 - (E) demonstrate the use of mathematics and physics to evaluate behavior in an existing game to enhance core logic.

Recommended Resources and Materials:

Software Requirements: Gaming and programming software such as Visual Studio 2010 or XNA Game Studio 4.0

Hardware Requirements: Computer – 1.6GHz or faster processor; 1 GB (32 Bit) or 2 GB (64 Bit) RAM (Add 512 MB if running in a virtual machine); 3GB of available hard drive space; 5400 RPM Hard Disk Drive; DirectX 9.0c capable video card running at 1024 x 768 or higher resolution display; Recommended graphics card that supports DirectX 10, with a supporting WDDM 1.1 driver; DVD-ROM Drive; Projector/Printer /Copier

Resources: Gaming and programming resources such as: Stempfuse.com; Microsoft Developers Network; [http://msdn.microsoft.com/en-us/library/ms228593\(v=vs.100\)](http://msdn.microsoft.com/en-us/library/ms228593(v=vs.100)); Xbox Indie Games; <http://xbox.create.msdn.com/en-US>

Recommended Course Activities:

Students will explore the physics, artificial intelligence, 3D modeling, alternative interfaces, and story & character design required in game and simulation design. Students will interact with videos, text, pictures, and animations to simulate a real-world environment.

Students will create the following game types: side scrolling; platform; maze; simulations; and racing within the program environment.

Suggested methods for evaluating student outcomes:

- Worksheets and quizzes embedded within the curriculum,
- Rubric for assessing final product and participation,
- Electronic Portfolio of process for creating video game,
- Final product of a personalized video game.

Teacher qualifications:

- (1) Any business or office education certificate.
- (2) Business and Finance: Grades 6-12.
- (3) Business Education: Grades 6-12.
- (4) Secondary Industrial Arts (Grades 6-12).
- (5) Secondary Industrial Technology (Grades 6-12).
- (6) Technology Education: Grades 6-12.
- (7) Technology Applications: Early Childhood-Grade 12.
- (8) Technology Applications: Grades 8-12.
- (9) Trade and Industrial Education: Grades 6-12. This assignment requires appropriate work approval.
- (10) Trade and Industrial Education: Grades 8-12. This assignment requires appropriate work approval.
- (11) Vocational Trades and Industry. This assignment requires appropriate work approval.
- (12) Computer Science: Grades 8-12.
- (13) Secondary Computer Information Systems (Grades 6-12)

Additional information: