UNITED INDEPENDENT SCHOOL DISTRICT
AGENDA ACTION ITEM

TOPIC: Approval of Innovative Courses: Video Game Design, Video Game Design II, and Video Game Design III

SUBMITTED BY: Alicia G. Carrillo, Ph.D., Director of Career and Technical Education

OF: Curriculum and Instruction Department

APPROVED FOR TRANSMITTAL TO SCHOOL BOARD: _____

DATE ASSIGNED FOR BOARD CONSIDERATION: August 19, 2015

Recommendation:

Administration recommends approval of three innovative courses for Video Game Design:
Career Cluster: Visual Arts; Career Pathway: Arts, Audio/Video Technology, Communications.

Video Game Design, (VIDEOGD), N1300993
Video Game Design II, (VIDEOGD2), N1300994
Video Game Design III, (VIDEOGD3), N1300995

Rationale:

The three courses in this pathway provide interested students with opportunities to design, program and create functional video games. Video Game Design will introduce basic programming language and skills essential to developing a video game. Topics covered include math, physics, design and computer programming.

Video Game Design II & Video Game Design III expand on the foundation created in Video Game Design through programming languages such as: C+ programming, XNA game studio, Java, & Android Applications. In VGD2, students will dive into 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. In VGD3, students will develop mobile applications.

Budgetary Information:

Program will be funded using through the 2015-16 Perkins Grant.

Board Policy Reference and Compliance:

EHBF (LEGAL): Each public school student shall master the basic skills and knowledge necessary for managing the dual roles of family member and wage earner and for gaining entry-level employment in a high-skill, high-wage job or continuing the student's education at the postsecondary level. *Education Code 29.181*

TEA

Course: Video Game Design

PEIMS Code : N1300993* **Abbreviation:** VIDEOGD

Number of credits that may be earned: 1 credit

Brief description of the course (150 words or less):

The student will be provided the opportunity to design, program, and create a functional video game. The course will introduce basic programming language and skills that are essential to developing a video game. Topics covered are math, physics, design, and computer programming.

Essential Knowledge and Skills of the course:

1. Foundation- The student demonstrates knowledge and appropriate use of hardware components, software programs, and their connections.

   (A) demonstrate knowledge and appropriate use of operating systems, software applications, and communication and networking components;
   (B) compare, contrast, and appropriately use the various input, processing, output, and primary/secondary storage devices;
   (C) make decisions regarding the selection, acquisition, and use of software taking under consideration its quality, appropriateness, effectiveness, and efficiency;
   (D) delineate and make necessary adjustments regarding compatibility issues including, but not limited to, digital file formats and cross platform connectivity;
   (E) differentiate current programming languages, discuss the use of the languages in other fields of study, and demonstrate knowledge of specific programming terminology and concepts; (F) differentiate among the levels of programming languages including machine, assembly, high-level compiled and interpreted languages;
   (G) demonstrate coding proficiency in a contemporary programming language.
   H) identify object-oriented data types and delineate the advantages/disadvantages of object data;
   (I) demonstrate coding proficiency in contemporary programming languages including an object-oriented language; and
   (J) survey the issues accompanying the development of large software systems such as design/implementation teams, software validation/testing, and risk assessment.

2. Foundation - The student uses data input skills appropriate to the task. The student is expected to:
   (A) demonstrate proficiency in the use of a variety of input devices such as keyboard, scanner, voice/sound recorder, mouse, touch screen, or
   (B) use digital keyboarding standards for the input of data.

3. Foundation - The student complies with the laws and examines the issues regarding the use of technology in society. The student is expected to:
   (A) discuss copyright laws/issues and model ethical acquisition and use of digital information, citing sources using established methods
   (B) demonstrate proper etiquette and knowledge of acceptable use policies when using networks, especially resources on the Internet and intranet;
   (C) investigate measures, such as passwords or virus detection/prevention, to protect computer systems and databases from unauthorized use and tampering;
   (D) discuss the impact of computer programming on the World Wide Web community, and
   (E) code modules for the World Wide Web (WWW) community.

4. Information and Acquisition - The student acquires electronic information in a variety of formats, with appropriate supervision. The student is expected to:
(A) acquire information in and knowledge about electronic formats including text, audio, video, and graphics;
(B) use a variety of resources, including foundation and enrichment curricula, together with various productivity tools to gather authentic data; and
(C) design and document sequential search algorithms for digital information storage and retrieval.

5. Information and Acquisition - The student evaluates the acquired electronic information. The student is expected to:
(A) determine and employ methods to evaluate the design and functionality of the process using effective coding, design, and test data; and
(B) implement methods for the evaluation of the information using defined rubrics.

6. Information and Acquisition - The student acquires electronic information in a variety of formats, with appropriate supervision. The student is expected to:
(A) acquire information in and knowledge about electronic formats including text, audio, video, and graphics; and
(B) use a variety of resources, including foundation and enrichment curricula, together with various productivity tools to gather authentic data as a design and document sequential search algorithms for digital information storage and retrieval.

7. Solving Problems - The student uses appropriate computer-based productivity tools to create and modify solutions to problems. The student is expected to:
(A) apply problem-solving strategies such as design specifications, modular top-down design, step-wise refinement, or algorithm development;
(B) use visual organizers to design solutions such as flowcharts or schematic drawings;
(C) develop sequential and iterative algorithms and codes programs in prevailing computer languages to solve practical problems modeled from school and community;
(D) code using various data types;
(E) demonstrate effective use of predefined input and output procedures for lists of computer instructions including procedures to protect from invalid input;
(F) develop coding with correct and efficient use of expressions and assignment statements including the use of standard/user-defined functions data structures, operators/proper operator precedence, and sequential/conditional/repetitive control structures;
(G) create and use libraries of generic modular code to be used for efficient programming;
(H) identify actual and formal parameters and use value and reference parameters;
(I) use control structures such as conditional statements and iterated, pretest, and post test loops;
(J) use sequential, conditional, selection, and repetition execution control structures such as menu-driven programs that branch and allow user input;
(K) identify and use structured data types of one-dimensional arrays, records, and text files;
(L) use appropriately and trace recursion in program design comparing invariant, iterative, and recursive algorithms;
(M) manipulate data structures using string processing;
(N) use notation for language definition such as syntax diagrams or Backus-Naur forms;
(O) identify, describe, and use sequential/non-sequential files; multidimensional arrays and arrays of records; and quadratic sort algorithms such as selection, bubble, or insertion, and more efficient algorithms including merge, shell, and quick sorts;
(P) create robust programs with increased emphasis on design, style, clarity of expression and documentation for ease of maintenance, program expansion, reliability, and validity;
(Q) apply methods for computing iterative approximations and statistical algorithms;

(R) define and develop code using the concepts of abstract data types including stacks, queues, linked lists, trees, graphs, and information hiding;
(S) identify and describe the correctness and complexity of algorithms such as divide and conquer, backtracking, or greedy algorithms;
(T) develop software to solve a school or community problem such as customer relations, design, modular programming, documentation, validation, marketing, or support; and
(U) research advanced computer science concepts such as applied artificial intelligence, expert systems, robotics, depth-first/breadth-first and heuristic search strategies, multitasking operating systems, or computer architecture, such as reduced instruction set computer (RISC) and complex instruction set computer (CISC).

8. Solving Problems - The student uses technology applications to facilitate evaluation of work, both process and product. The student is expected to:
(A) design and implement procedures to track trends, set timelines, and review/evaluate progress for continual improvement in process and product;
(B) use correct programming style to enhance the readability and functionality of the code such as spacing, descriptive identifiers, comments, or documentation;
(C) seek and respond to advice from peers and professionals in delineating technological tasks;
(D) resolve information conflicts and validate information through accessing, researching, and comparing data;
(E) create technology specifications for tasks/evaluation rubrics and demonstrate that products/product quality can be evaluated against established criteria;
(F) demonstrate the ability to read and modify large programs including the design description and process development;
(G) analyze algorithms using "big-O" notation, best, average, and worst case space techniques;
(H) compare and contrast design methodologies including top-down and bottom-up;
(I) analyze models used in development of software including software life cycle models, design objectives, documentation, and support; and
(J) seek and respond to advice from peers and professionals in delineating technological tasks.

9. Solving Problems - The student uses research skills and electronic communication, with appropriate supervision, to create new knowledge. The student is expected to:
(A) participate with electronic communities as a learner, initiator, contributor, and teacher/mentor;
(B) demonstrate proficiency in, appropriate use of, and navigation of LANs and WANs for research and for sharing of resources;
(C) extend the learning environment beyond the school walls with digital products created to increase teaching and learning in the foundation and enrichment curricula; and
(D) participate in relevant, meaningful activities in the larger community and society to create electronic projects.

10. Communication - The student formats digital information for appropriate and effective communication. The student is expected to:
(A) annotate coding properly with comments, indentation, and formatting; and
B) create interactive documents using modeling, simulation, and hypertext.

11. Communication - The student delivers the product electronically in a variety of media, with appropriate supervision. The student is expected to:
(A) publish information in a variety of ways including, but not limited to, printed copy and monitor displays; and
(B) publish information in a variety of ways including, but not limited to, software, Internet documents, and video.

12. Communication - The student uses technology applications to facilitate evaluation of communication, both process and product. The student is expected to
(A) write technology specifications for planning/evaluation rubrics documenting variables, prompts, and programming code internally and externally;
(B) seek and respond to advice from peers and professionals in evaluating the product; and
(C) debug and solve problems using reference materials and effective strategies.

Description of the specific student needs this course is designed to meet:

Students apply technical skills to address business applications of emerging technologies. Knowledge and skills in programming and creating video games are delivered through totally interactive software. Through text, pictures, animations, digital video and a highly interactive and self-directed curriculum, students are led through the exciting world of video game design.

This course fulfills the technology applications requirement for graduation. This course will instruct students on how to apply the design or problem solving process in order to create a real world solution and learn the basics of programming structure and functions.

Major resources and materials to be used in the course:

Video Game Design curriculum by I Support Learning, Inc.
DarkBASIC is a program distributed by The Game Creators which includes a 3D game engine and its own BASIC programming language.

Required activities and sample optional activities to be used:

Students will follow the design process to create their own video game.
Students will interact with videos, text, pictures, and animations to simulate a real-world environment.
They will use video tutorials and activities to learn how to program in DarkBASIC.

Methods for evaluating student outcomes:

Worksheets and Quizzes that are embedded within the curriculum; Rubric for assessing final product and participation; Portfolio of process for creating video game; Final product of a personalized video game.

Teacher qualifications:

Business 120, Tech Ed 123, Tech App 126, or Trade & Industrial Education 125

Additional information (optional):

This course is recommended for students in Grades 9-12.

TEXAS EDUCATION AGENCY

Course: Video Game Design II & III
PEIMS Code: N1300994, N1300995
Abbreviation: VIDEOGD2, VIDEOGD3
Grade Level(s): 10-12
Number of Credits: 0.5-1.0

Course description:

Video Game Design II & Video Game Design III expands on the foundation created in Video Game Design through programming languages such as: C# programming, XNA game studio, Java, & Andriod App. In VGD2, students will dive into 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. In VGD3, students will develop mobile applications.

Essential knowledge and skills:

Video Game Design II

(a) Knowledge and skills.

(1) The student applies Programming Skills related to software development and computer programming. The student is expected to:

(A) demonstrate an understanding of the core principles of programming;

(B) develop software applications;

(C) analyze the basic programming structure of application and be able to debug, compile, and run an application;

(D) create, name, and assign values to variables;

(E) create custom methods that can return values and take parameters;

(F) utilize common built-in objects and reference types;

(G) utilize common programming statements to implement flow control, looping, and exception handling;

(H) create, initialize, and use collections; and

(I) design and create custom classes and use the object-oriented techniques of inheritance, abstraction, polymorphism, and encapsulation.

(2) The student applies game development skills. The student is expected to:

(A) demonstrate significant understanding game development tools;

(B) employ 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 characters, items, chests, quests, and monsters;

(E) create enhancements to the combat engine logic with role-playing game;

(F) understand 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.

(3) The student applies creative, innovative, and critical thinking. The student is expected to:

(A) demonstrate ability to enhance existing game program;

(B) create, design, and program original working game features;

(C) explain how separated game logic fits together to form a cohesive game application;

(D) troubleshoot existing applications and game programs to fix bugs and ensure performance

(E) perform self-evaluations of projects against the required established directives; and

(F) follow technical and increasingly complex programming instructions in order and detail.

(4) The student applies communication and collaboration as an individual and as part of a team. The student is expected to:

(A) conduct in-class presentations including demonstration of original game concepts;

(B) collaborate with classmates in problem solving and debugging program errors; and

(C) use technical writing skills to explain game design concepts, document programming logic, document development processes.

(5) The student applies the use of appropriate and accessible digital tools for research and learning. The student is expected to:

(A) use appropriate websites, wiki's, and blogs to engage other users for research, ideas, and help; and

(B) use websites to explore concepts learned and to reference coding syntax.

(6) The student applies engineering, physics, and mathematical concepts critical to game development. The student is expected to:

(A) apply mathematical calculations and formulas to programming logic, creating custom behavior and functionality in an application and a game;

(B) perform physics calculations on objects within a game:

(C) recognize software engineering design principles by working within and enhancing a complex fully functional game:

(D) reverse engineer existing game functionality in order to understand game design; and

(E) apply knowledge of math and physics to evaluate behavior in an existing game to enhance core logic.

Video Game Design 3

(a) Knowledge and skills.

(1) The student applies Technical Skills related to software development, computer programming, and graphic design. The student is expected to:

(A) introduce an application development for mobile application technology;

(B) program by writing code;

(C) follow technical and increasingly complex programming instructions in order and detail;

(D) design and program real working education based mobile application projects;

(E) use digital design resources and color theory to draw and animate sprites, objects, platforms, backgrounds, and loops;

(F) become familiar with common mobile application technologies and platforms; open files, save files, create and program original material, integrate separate files into a mobile application project, create and edit audio sound effects and music; and

(G) technical writing; user instructions, application rules and document development process within a development team.

(2) The student applies creative, innovative, and critical thinking. The student is expected to:

(A) learn steps of the engineering design cycle and how it works as a practical problem solving method;

(B) use gained technical skills to improve mobile application projects;

(C) use gained technical skills to create, design, and program original applications;

(D) troubleshoot applications to fix bugs and ensure performance:

(E) test fellow classmate's applications to ensure performance:

(F) perform self-evaluations of projects against the required established directives:

(G) perform evaluations of classmate's projects against the required established directives; and

(H) develop a marketing plan for original programmed application to include; target audience, current competition, delivery options, product pricing, logo design, and strategy to spend budgeted funds.

(3) The student applies communication and collaboration as an individual and as part of a team. The student is expected to:

(A) form mobile application development groups to achieve directive of completing working mobile applications;

(B) assign tasks to members of development group to achieve directive of creating a mobile application;

(C) use the engineering design cycle within a development team to achieve directive of creating a working mobile application;

(D) project management; students will have the opportunity to lead a development team, assign tasks, evaluate progress, facilitate communication among team members and ensure that project is completed within time deadline; and

(E) conduct in-class presentations including demonstration of application.

(4) The student applies the use of appropriate and accessible digital tools for research and learning. The student is expected to:

(A) use appropriate internet websites to gather and analyze research on a variety of subjects including: mobile application development, marketing statistics, color and design theory, post-secondary education options and careers in mobile application development and other STEM related industries; and

(B) use appropriate wiki's and blogs to engage other (distant) users of mobile application technology for research, ideas, and help.

(5) The student applies the use of engineering, physics, and mathematical concepts critical to mobile application development. The student is expected to:

(A) learn how the process used in designing and developing software can be applied to other design and development projects such as bridges, buildings, and machines;

(B) learn how basic physics concepts like gravity, acceleration, velocity, speed, trajectory, Newton's Laws of Motion, force and elasticity are used in mobile application development;

(C) use required mathematical techniques to perform physics calculations in determining how physics is used in mobile application development as compared to the real world; and

(D) use knowledge of math and physics to evaluate behavior within applications in the "virtual world" as compared to the "real world."

(6) The student applies post-secondary and career options and resources related to mobile application development and STEM. The student is expected to:

(A) research how technical and communication skills used in mobile application design translates to other technology industries and businesses;

(B) research required post-secondary diplomas, certificates, and degrees needed to gain employment in game development and other technology based industries; and

(C) research career trends, wage data, and employment opportunities in mobile application development and technology based industries.

Description of specific student needs this course is designed to meet:

These courses give students the opportunity to dive further into gaming development and provide them with real world processes and systems used in the professional world of video games and simulations.

Major resources and materials:

VGD 2 Software Requirements: Gaming and programming software such as: Visual Studio 2010; XNA Game Studio 4.0

VGD 2 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

VGD 2 Resources: Gaming and programming resources such as: Stemfuse.com; Microsoft Developers Network; http://msdn.microsoft.com/en-us/library/ms228593(v=vs.100); XBox Indie Games; http://xbox.create.msdn.com/en-US

VGD 3 Software Requirements: Gaming and programming software such as: Java Developer Kit; Eclipse Classic Version; Android SDK; ADT Plugin for Eclipse; SQLite Database Browser

VGD 3 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

VGD 3 Resources: Gaming and programming resources such as: Stemfuse.com; Android Developers; http://developer.android.com/index.html

Required activities and sample optional activities to be used:

Students will delve into 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.

Methods for evaluating student outcomes:

Worksheet, 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:

Same as the certification requirement for CTE's Computer Programming course (Business or Technology Education)