eDynamic Learning Course Title: Game Design 2a / 2b

State: TX
State Course Title: Video Game Programming
State Course Code:
State Standards: Video Game Programming
Date of Standards: 2009-2010

Standards	Course Title. (a or b), if applicable, e.g. Game Design 1a	Unit Name(s)	Lesson(s) Numbers
(1) The student demonstrates professional standards / employability skills as required by business and industry.			
(A) identify and demonstrate positive work behaviors and personal qualities needed to be employable;	Game Design 2a	Unit 1: Get Paid to Make Games!	Lesson 1
(B) demonstrate skills such as building a resume related to seeking and applying for employment;	Game Design 2a	Unit 2: Give Yourself Some Wow Factor	Lesson 1
(C) create a career portfolio to document information such as work experiences, licenses, certifications, and work samples;	Game Design 2a	Unit 2: Give Yourself Some Wow Factor	Lesson 1
(D) compare and evaluate employment opportunities in the game programming industry.	Game Design 2a	Unit 1: Get Paid to Make Games!	Lesson 2
(2) The student applies programming skills related to software development and computer programming.			
(A) develop software applications;	Game Design 2b	Unit 7: Betas, Packaging and Publishing	Lesson 1
(B) analyze the basic programming structure of application and be able to debug, compile, and run an application;	Game Design 2b	Unit 6: Enemies, Interfaces and Testing	Lesson 4
(C) create, name, and assign values to variables;	Game Design 2b	Unit 3: Introduction to Level Design	Lesson 3
(D) create custom methods that can return values and take parameters;	Game Design 2b	Unit 5: Finishing Touches	Lesson 4
(E) apply common built-in objects and reference types;	Game Design 2b	Unit 1: Taking Control	Lesson 2
(F) apply common programming statements to implement flow control, looping, and exception handling;	Game Design 2b	Unit 3: Introduction to Level Design	Lesson 4
(G) create, initialize, and use collections;	Game Design 2b	Unit 3: Introduction to Level Design	Lesson 2
(H) design and create custom class-constructors and use the object-oriented techniques of inheritance, abstraction, polymorphism, and encapsulation.	Game Design 2b	Unit 3: Introduction to Level Design	Lesson 4
(3) The student applies game development skills.			
(A) demonstrate significant understanding of game development tools including graphic design, game engines, animation, editors, and programing	Game Design 2b	Unit 4: Extending Unity	Lesson 2

(B) apply core programming logic and techniques that are used in building games;	Game Design 2b	Unit 1: Taking Control	Lesson 2
(C) identify the code, structure, and layout of a fully functional role-playing game;	Game Design 2b	Unit 3: Introduction to Level Design	Lesson 3
(D) create and customize new game elements such as characters, items, chests, quests, and monsters;	Game Design 2b	Unit 1: Taking Control	Lessons 1, 3
(E) create enhancements to the combat engine logic with role-playing game;	Game Design 2b	Unit 5: Finishing Touches	Lesson 2
(F) research the inner workings of the role-playing game system, for the purpose of modifying simulated game actions;	Game Design 2b	Unit 3: Introduction to Level Design	Lesson 2
(G) describe how a two-dimensional tile-based rendering and collision system works to create maps in a game.	Game Design 2b	Unit 3: Introduction to Level Design	Lesson 2
(4) The student applies creativity, innovation, and critical-thinking skills to video game programming methodology.			
(A) demonstrate the ability to enhance existing game program(s) by customizing screens, adding levels, adding characters, and adding graphics;	Game Design 2b	Unit 1: Taking Control	Lesson 1
(B) create, design, and program original working game features;	Game Design 2b	Unit 3: Introduction to Level Design	Lesson 4
(C) explain how separated game logic fits together to form a cohesive game application;	Game Design 2b	Unit 4: Extending Unity	Lesson 3
(D) critique beta applications and provide solutions to fix bugs and ensure performance;	Game Design 2b	Unit 7: Betas, Packaging and Publishing	Lesson 1
(E) conduct a self-evaluation and discuss findings with peers	Game Design 2b	Unit 6: Enemies, Interfaces and Testing	Lesson 4
(F) compare projects with the required established game specifics;	Game Design 2b	Unit 6: Enemies, Interfaces and Testing	Lesson 4
(G) interpret technical and increasingly complex programming instructions in order and in detail.	Game Design 2b	Unit 3: Introduction to Level Design	Lesson 3
(5) The student applies communication and collaboration skills as an individual and as part of a team.			
(A) demonstrate the concepts of the original game and conduct in-class presentations including demonstration of original game concepts;	Game Design 2b	Unit 7: Betas, Packaging and Publishing	Activity 1
(B) analyze and solve program errors individually or in teams and collaborate with classmates in problem solving and debugging program errors;	Game Design 2b	Unit 1: Taking Control	Lesson 1
(C) apply technical writing skills to explain game design concepts, document programming logic, and document development processes.	Game Design 2a	Unit 2: Give Yourself Some Wow Factor	Lesson 4
(6) The student applies the use of appropriate and available digital tools for research and learning.			
(A) review and research websites, wiki's, and blogs for appropriate content, ideas, and best practices to engage other users;	Game Design 2b	Unit 4: Extending Unity	Lesson 4
(B) investigate websites to explain concepts learned and to reference coding syntax.	Game Design 2b	Unit 4: Extending Unity	Lesson 4
(7) The student applies engineering, physics, and mathematical concepts critical to game development.			

(A) discuss and describe the principles of software engineering design within complex functional games;	Game Design 2b	Unit 4: Extending Unity	Lesson 4
(B) apply the principles of software engineering to enhance a complex functional game including multiple movements and multiple controls;	Game Design 2b	Unit 5: Finishing Touches	Lesson 3
(C) apply the principles of software engineering within a complex fully functional game/bug free program;	Game Design 2b	Unit 7: Betas, Packaging and Publishing	Lessons 2, 4
(D) reverse engineer existing game functionality to understand game design;	Game Design 2b	Unit 3: Introduction to Level Design	Lesson 3
(E) demonstrate the use of mathematics and physics to evaluate behavior in an existing game to enhance core logic.	Game Design 2b	Unit 5: Finishing Touches	Lesson 4