

CompuScholar, Inc.

Alignment to Ohio "145090 - Game Design" Course Standards

Ohio Course Details:

Course Title:	Information Technology
Course Code(s):	145090 - Game Design
Credit:	1
Grade Level:	9th-12th
State Standards Link:	http://education.ohio.gov/Topics/Career-Tech/Information-Technology-Career-Field

CompuScholar Course Details:

Course Title:	Unity Game Programming
Course ISBN:	978-0-9887070-7-8
Course Year:	2019

Note 1: Citation(s) listed may represent a subset of the instances where objectives are met throughout the course.

Note 2: Citation(s) for a "Lesson" refer to the "Lesson Text" elements and associated "Activities" within the course, unless otherwise noted. The "Instructional Video" components are supplements designed to introduce or re-enforce the main lesson concepts, and the Lesson Text contains full details.

Course Description

This course will prepare students to design and program games using commercial and open source programs and applications. Students will learn industry standard programming language constructs to write programs that integrate classes, class methods, and class instances. Students will learn input method handling, animation, collision detection, game physics and basic artificial intelligence.

Course Standards

Strand 2 - IT Fundamentals	
Learners apply fundamental principles of IT, including the history of IT and its impact on society, common industry terms, systems theory, information storage and retrieval, database management, and computer hardware, software, and peripheral device configuration and installation. This base of knowledge and skills may be applied across the career field.	
2.9. Project Concept Proposal: Develop a project concept proposal.	CITATION(S)
2.9.1. Identify and incorporate branding strategies.	Chapter 13, Lesson 1
2.9.2. Determine the scope and purpose of the project.	Chapter 13, Lessons 1 - 2
2.9.3. Determine the target audience, client needs, expected outcomes, objectives, and budget.	Chapter 13, Lessons 1, 4
2.9.4. Develop a conceptual model and design brief for the project.	Chapter 13, Lessons 1, 2, 4

2.9.5. Develop a timeline, communication plan, task breakdown, costs (e.g., equipment, labor), deliverables, and responsibilities for completion.	Chapters 14, 25, 26
2.9.6. Develop and present a comprehensive proposal to stakeholders.	Chapter 13, Lessons 1, 4 Chapter 25, Lesson 2
2.12. Performance Tests and Acceptance Plans: Develop performance tests and acceptance plans.	CITATION(S)
2.12.1. Create a written procedure agreed by the stakeholders and project team for determining the acceptability of the project deliverables.	Chapter 25, Lesson 2
2.12.2. Develop a test system that accurately mimics external interfaces.	N/A
2.12.3. Develop test cases that are realistic, compare with expected performance, and include targeted platforms and device types.	Chapter 11, Lesson 2 Chapters 14, 26 Activity 3
2.12.4. Develop, perform, and document usability and testing integration.	N/A
2.12.5. Make corrections indicated by test results.	Chapter 11, Lesson 2 Chapters 14, 26 Activity 3
2.12.6. Seek stakeholder acceptance upon successful completion of the test plan.	Chapters 14, 26 Activity 3
2.13. Rollout and Handoff: Plan rollout and facilitate handoff to customer.	CITATION(S)
2.13.1. Include overall project goals and timelines in the rollout plan.	Chapters 14, 26 Activity 1
2.13.2. Communicate rollout plans to key stakeholders in a timely manner.	Chapters 14, 26 Activity 1
2.13.3. Conduct final review and approvals according to company standards.	Chapters 14, 26 Activity 3
2.13.4. Identify support staff, training needs, and contingency plans in the rollout plan.	N/A
2.13.5. Test delivered application to assure that it is fully functional for the customer or user and meets all requirements.	Chapters 14, 26 Activity 3
Strand 5 - Programming and Software System	
Learners apply principles of computer programming and software development to develop code; build, test, and debug programs; create finished products; and plan, analyze, design, develop, implement, and support software applications.	
5.2. Computational and String Operations: Develop code that performs computational and string operations.	CITATION(S)
5.2.1. Compare and contrast primitive types of numeric and nonnumeric data (e.g., integers, floats, Boolean, strings).	Chapter 6, Lesson 1
5.2.2. Identify the scope of data (e.g., global versus local, variables, constants, arrays).	Chapter 6, Lesson 3
5.2.3. Write code that uses arithmetic operations.	Chapter 6, Lesson 2

5.2.4. Write code that uses subtotals and final totals.	As needed throughout (e.g. keeping score in each game)
5.2.5. Write code that applies string operations (e.g., concatenation, pattern matching, substring).	N/A
5.3. Logical Operations and Control Structures: Develop code that uses logical operations and control structures.	CITATION(S)
5.3.1. Explain Boolean logic.	Chapter 7, Lesson 1
5.3.2. Solve a truth table.	N/A
5.3.3. Write code that uses logical operators (e.g., and, or, not).	Chapter 7, Lessons 1 - 2
5.3.4. Write code that uses relational operators and compound conditions.	Chapter 7, Lessons 1 - 2
5.3.5. Write code that uses conditional control structures (e.g. if, if-then-else).	Chapter 7, Lessons 1 - 2
5.3.6. Write code that uses repetition control structures (e.g., while, for).	Chapter 12, Lessons 2 - 3
5.3.7. Write code that uses selection control structures (e.g., case, switch).	Chapter 7, Lesson 3
5.3.8. Write code that uses nested structures and recursion.	Chapter 12 Activity (nested loops)
5.3.9. Write code that creates and calls functions.	Chapter 9, Lesson 3
5.3.10. Code error-handling techniques.	Chapter 11, Lesson 1
5.3.11. Write code to access data repositories.	N/A
5.3.12. Write code to create classes, objects, and methods.	Chapter 9
5.4. Integrated Development Environment: Build and test a program using an integrated development environment (IDE).	CITATION(S)
5.4.1. Configure options, preferences, and tools.	Chapter 2, Lesson 1
5.4.2. Write and edit code in the IDE.	Chapter 3
5.4.3. Compile or interpret a working program.	Chapters 2, 3 and throughout the course
5.4.4. Define test cases.	Chapters 14, 26 Activity 3
5.4.5. Test the program using defined test cases.	Chapters 14, 26 Activity 3
5.4.6. Correct syntax and runtime errors.	Chapter 11, Lessons 2 - 3 Chapters 14, 26 Activity 3

5.4.7. Debug logic errors.	Chapter 11, Lessons 2 - 3 Chapters 14, 26 Activity 3
5.5. Programming Conventions: Develop programs using applications security practices.	CITATION(S)
5.5.1. Develop programs using data validation techniques.	N/A
5.5.2. Develop programs that use reuse libraries.	Chapter 1 and throughout the course (Unity library)
5.5.3. Develop programs using operating system calls.	Chapter 3 and throughout the course (via .NET library)
5.5.4. Develop programs that call other programs.	N/A
5.5.5. Use appropriate naming conventions and apply comments.	Chapter 3, Lesson 3 Chapter 6, Lesson 1
5.5.6. Format output (e.g., desktop, mobile, enterprise, reports, data files).	All output is game-specific (e.g. game scores, mini-maps)
5.6. Software Development Lifecycle: Apply the software development lifecycle (SDLC).	CITATION(S)
5.6.1. Determine requirements specification documentation.	Chapter 13, Lesson 1
5.6.2. Identify constraints and system processing requirements.	Chapter 13, Lessons 1, 2, 4
5.6.3. Develop and adhere to timelines.	Chapters 14, 26 Activity 1
5.6.4. Identify a programming language, framework, and an integrated development environment (IDE).	Chapter 1
5.6.5. Identify input and output (I/O) requirements.	Chapter 13, Lessons 1, 2, 4
5.6.6. Design system inputs, outputs, and processes.	Chapter 13, Lessons 1, 2, 4
5.6.7. Document a design using the appropriate tools (e.g., program flowchart, dataflow diagrams, Unified Modeling Language [UML]).	Chapter 13, Lessons 1, 2, 4
5.6.8. Create documentation (e.g., implementation plan, contingency plan, data dictionary, user help).	Chapters 13, 14, 26 Chapter 25, Lesson 2, Activity
5.6.9. Review the design (e.g., peer walkthrough).	Chapters 14, 26 Activity 2
5.6.10. Present system design to stakeholders.	Chapters 14, 26 Activity 2
5.6.11. Develop the application.	Chapters 14, 26 Activity 3
5.6.12. Compare and contrast software methodologies (e.g., agile, waterfall).	Chapter 25, Lesson 1
5.6.13. Perform code reviews (e.g., peer walkthrough, static analysis).	Chapter 11, Lesson 2 Chapters 14, 26 Activity 3

5.6.14. Ensure code quality by testing and debugging the application (e.g., system testing, user acceptance testing).	Chapter 11, Lessons 2 - 3 Chapters 14, 26 Activity 3
5.6.15. Train stakeholders.	N/A
5.6.16. Deploy the application.	Chapter 25, Lessons 2 - 4
5.6.17. Collect application feedback and maintain the application.	N/A
5.7. Configuration Management: Describe configuration management activities.	CITATION(S)
5.7.1. Explain version management and interface control.	Chapter 25, Lesson 1
5.7.2. Explain baseline and software lifecycle phases.	Chapter 25, Lesson 1
5.7.3. Analyze the impact of changes.	Chapter 25, Lesson 1

Strand 7 - Digital Media	
Learners apply principles of digital media to produce interactive media; develop and produce multimedia applications; integrate typography into media; create 3D models and 2D and 3D animation; and create digital video, audio, and photographs.	
7.2. Multimedia Tools: Develop navigational structures, scripts, storyboards, and flowcharts for multimedia applications.	CITATION(S)
7.2.1. Choose a navigational menu structure (e.g., rollovers, dropdowns, disjointed).	Chapter 22, Lessons 1 - 2 Chapter 14, Lesson 1
7.2.2. Construct and place navigational units.	Chapter 21, Lesson 3
7.2.3. Build in interactive elements.	Chapter 4, Lesson 3
7.2.4. Determine uses and needs for site maps, multimedia scripts, storyboards, and flowcharts.	Chapter 21, Lesson 2
7.2.5. Make preliminary sketches showing placement of images and text on screen.	Chapter 13, Lesson 1
7.2.6. Place buttons and navigational graphics.	Chapter 22, Lessons 1 - 2 Chapter 14, Lesson 1
7.2.7. Select colors based on color theory and psychology.	Chapter 23, Lesson 2
7.2.8. Describe music, video, and special effects to be used.	Chapters 17 - 18
7.2.9. Provide a sample layout to stakeholders for review.	Chapter 13, Lessons 1, 2, 4
7.2.10. Select and create visual design elements appropriate for the intended audience and use.	Chapter 13, Lessons 1, 2, 4 Chapters 14, 22, 26
7.2.11. Develop characters and narrative to support intended outcomes.	Chapters 13, 14, 26