CompuScholar, Inc.

Alignment to the Oklahoma Academic Standards (OAS) for

Computer Science (9th - 12th Grade)

Unity Game Programming

Oklahoma Standards Information:

CS Page	Oklahoma Computer Science Standards
Standards Link:	2023 Oklahoma Academic Standards for Computer Science

CompuScholar Course Details:

Course Title:	Unity Game Programming
Course ISBN:	978-0-9887070-7-8
Course Year:	2024

Course Description

CompuScholar's ***Unity Game Programming*** curriculum is commonly used for **Video Game Design** courses in many states. The course covers introductory game design concepts within the Unity framework and introductory C# coding concepts, including decision-making, iteration, data structures, algorithms, OOP, and other classic CS topics.

Oklahoma Subject Codes

This course is best used as a primary resource for the following subject:

2511 - Advanced Programming

This is a specialty course focused on game design topics. Some items in the generic OAS do not fit into this context and are marked as N/A in the citations below.

Oklahoma Academic Standards (OAS) for Computer Science (High School)

Level 1 Requirements in White
Level 2 Requirements in Blue

Note 1: Citation(s) for a "Lesson" refer to the "**Lesson Text**" page where instruction of concepts is found. Additional hands-on practice can be found in the nearby "**Chapter Activity**" pages within that chapter.

Note 2: The "Instructional Video" components are optional supplements designed to introduce or reinforce the main lesson concepts and are not cited as standards-bearing content.

Note 3: Citation(s) to "Supplemental" or "Suppl." Chapters refer to Supplemental Chapters found at the end of the course.

Computing Systems	CITATIONS
Devices	
L1.CS.D.01 Model how abstractions hide the underlying implementation details of computing systems embedded in everyday	Chapter 1, Lesson 1 (game engines) Chapter 5, Lesson 1 (physics engine)
Hardware & Software	
L1.CS.HS.01 Analyze the levels of abstraction and interactions between application software, system software, and hardware.	N/A
L2.CS.HS.01 Identify and categorize the roles of a variety of operating system software.	N/A
Troubleshooting	
L1.CS.T.01 Develop and apply criteria for the systematic discovery of errors and systematic strategies for the correction of errors in computing systems.	Chapter 11, Lessons 2, 3 Chapter 14, Activity 3 Chapter 26, Activity 3
L2.CS.T.01 Illustrate how understanding the ways hardware components facilitate logic, input, output, and storage in computing systems will support troubleshooting.	N/A

Network & The Internet	CITATIONS
Network Communication & Organization	
L1.NI.NCO.01 Evaluate the scalability and reliability of networks by identifying and illustrating the basic components of computer networks (e.g., routers, switches, servers, etc.) and network protocols (e.g., IP, DNS).	N/A
L2.NI.NCO.01 Describe the issues that impact network functionality (e.g., bandwidth, load, latency, topology).	N/A
Cybersecurity	
L1.NI.CY.01 Compare physical and cybersecurity measures by evaluating trade-offs between the usability and security of a computing system and the risks of an attack.	Supplemental Chapter 1, Lesson 3
L2.NI.CY.01 Compare and refine ways in which software developers protect devices and information from unauthorized access.	Supplemental Chapter 1, Lesson 3
L1.NI.CY.02 Recommend security measures to address various scenarios based on information security principles.	Supplemental Chapter 1, Lesson 3
L1.NI.CY.03 Explain trade-offs when selecting and implementing cybersecurity recommendations from multiple perspectives, such as the user, enterprise, and government.	N/A

Data Analysis	CITATIONS
Storage	
L1.DA.S.01 Convert and compare different bit representations of data types, such as characters, numbers, and images	Chapter 6, Lesson 1 Supplemental Chapter 3, Lesson 2

L1.DA.S.02 Evaluate the trade-offs in how data is organized and stored digitally.	Chapter 6, Lesson 1 Chapter 8 Chapter 9, Lesson 5
Collection, Visualization & Transformation	
L1.DA.CVT.01 Use tools and techniques to locate, collect, and create visualizations of small and largescale data sets (e.g., paper surveys and online data sets).	Supplemental Chapter 3, Lesson 4
L2.DA.CVT.01 Use data analysis tools and techniques to identify patterns from complex real-world data.	Supplemental Chapter 3, Lesson 4
L2.DA.CVT.02 Generate data sets that use a variety of data collection tools and analysis techniques to support a claim and/or communicate information.	Supplemental Chapter 3, Lesson 4
Inference & Models	
L1.DA.IM.01 Illustrate and explain the relationships between collected data elements using computational models.	Supplemental Chapter 3, Lesson 4
L2.DA.IM.01 Use models and simulations to help plan, conduct, and refine investigations.	Supplemental Chapter 3, Lesson 4

Algorithms & Programming	**CITATIONS**
Algorithms	
L1.AP.A.01 Create a prototype that uses algorithms (e. g., searching, sorting, finding shortest distance) to provide a possible solution for a real- world problem.	Chapter 21, Lessons 2, 3
L2.AP.A.01 Model and use appropriate terminology to describe how artificial intelligence algorithms drive many software and physical systems (e.g., autonomous robots, pattern recognition, text analysis).	Chapter 21
L2.AP.A.02 Develop an artificial intelligence algorithm to play a game against a human opponent or solve a real-world problem.	Chapter 21 Activity
L2.AP.A.03 Critically examine and trace classic algorithms (e.g., selection sort, insertion sort, binary search, linear search).	Chapter 21, Lesson 3
L2.AP.A.04 Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency and clarity.	Chapter 21, Lesson 1
Variables	
L1.AP.V.01 Demonstrate the use of lists (e.g., arrays) to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	Chapter 12, Lessons 1, 2
L2.AP.V.01 Compare and contrast data structures and their uses (e.g., lists, stacks, queues).	Chapter 9, Lessons 1, 5 Chapter 12, Lesson 1
Control	
L1.AP.C.01 Justify the selection of specific control structures (e.g., sequence, conditionals, repetition, procedures) considering program efficiencies such as readability, performance, and memory usage.	Chapter 7 Chapter 9, Lesson 3 Chapter 12, Lessons 2, 3

L2.AP.C.01 Model the execution of repetition (e.g., loops, recursion) of an algorithm illustrating output and changes in values of named variables.	Chapter 12, Lessons 2, 3 Chapter 21, Lessons 2, 3
Modularity	
L1.AP.M.01 Decompose problems into procedures using systematic analysis and design.	Chapter 9, Lesson 3 Chapter 13, Lessons 1, 2, 4 Chapter 25, Lessons 1, 2
L2.AP.M.01 Construct solutions to problems using student-created components (e.g., procedures, modules, objects).	Chapters 9, 14, 23, 26
L1.AP.M.02 Create computational artifacts by systematically organizing, manipulating and/or processing data.	Chapter 9, Lessons 1, 5 Chapter 10 Chapter 12, Lessons 1, 2
L2.AP.M.02 Design or redesign a solution to a large-scale computational problem by identifying generalizable patterns.	Chapter 21
L2.AP.M.03 Create programming solutions by reusing existing code (e.g., libraries, Application Programming Interface (APIs), code repositories).	Students use Unity libraries throughout the course, e.g.: Chapters 5, 10, 17, 18, 19
Program Development	
L1.AP.PD.01 Create software that will provide solutions to a variety of users using a software development process.	Chapters 25, 26
L2.AP.PD.01 Create software that will provide solutions to a variety of users using multiple software development processes.	Chapters 25, 26
L1.AP.PD.02 Evaluate a variety of software licensing schemes (e.g., open source, freeware, commercial) and discuss the advantages and disadvantages of each scheme in software development.	Supplemental Chapter 1, Lesson 2
L2.AP.PD.02 Design software in a project team environment using integrated development environments (IDEs), versioning systems, and collaboration systems.	Chapter 2 (IDE) Chapters 14, 25, 26 (team project) (excluding versioning and collaboration systems)
L1.AP.PD.03 While working in a team, develop, test, and refine event-based programs that solve practical problems or allow self-expression.	Chapter 3, Lesson 4 (events) Chapters 14, 25, 26 (team projects)
L2.AP.PD.03 Develop programs for multiple computing platforms.	Chapter 24, Lessons 2, 3, 4
L1.AP.PD.04 Using visual aids and documentation, illustrate the design elements and data flow (e.g., flowcharts, pseudocode) of the development of a complex program.	Chapter 21, Lesson 2 Chapter 25, Lesson 2 Chapter 14 & 26, Activities 1, 2
L2.AP.PD.04 Systematically examine code for correctness, usability, readability, efficiency, portability, and scalability through peer review.	Chapter 11, Lesson 2 Chapter 25, Lesson 2 Chapters 14 & 26, Activity 3
L1.AP.PD.05 Evaluate and refine computational artifacts to make them more user-friendly, efficient and/or accessible.	Chapter 21, Lessons 1, 3 Chapters 14 & 26, Activity 3
L2.AP.PD.05 Develop and use a series of test cases to verify that a program performs according to its design specifications.	Chapter 11, Lessons 2, 3 Chapters 14 & 26, Activity 3
L2.AP.PD.06 Explain security issues that might lead to compromised computer programs.	Supplemental Chapter 1, Lesson 3

L2.AP.PD.07 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).	Chapters 16, 17, 18 Activities (incremental additions to a program) Chapters 14 & 26, Activity 3 (Iterative testing)

Impacts of Computing	**CITATIONS**
Culture	
L1.IC.CU.01 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	Supplemental Chapter 1, Lessons 1, 3 Supplemental Chapter 2, Lesson 2 Supplemental Chapter 3, Lessons 3, 4
L2.IC.CU.01 Evaluate the beneficial and harmful effects that computational artifacts and innovations have on society.	Supplemental Chapter 2 Supplemental Chapter 3, Lessons 3, 4
L1.IC.CU.02 Test and refine computational artifacts to ensure access to a variety of user audiences.	N/A
L2.IC.CU.02 Evaluate the impact of location and user audience on the distribution of computing resources in a global society.	N/A
L1.IC.CU.03 Demonstrate ways a given algorithm can help solve computational problems across disciplines.	Chapter 21, Lesson 3 Supplemental Chapter 3, Lesson 4
L2.IC.CU.03 Design and implement a study that evaluates or predicts how creating, testing, and refining computational artifacts has revolutionized an aspect of our culture and how it might evolve (e.g., education, healthcare, art/entertainment, energy).	N/A
Social Interactions	
L1.IC.SI.01 Demonstrate and debate how computing increases and decreases connectivity and communication among people of various cultures.	Supplemental Chapter 3, Lesson 1
Safety, Law & Ethics	
L1.IC.SLE.01 Describe the beneficial and harmful effects that intellectual property laws can have on innovation.	Supplemental Chapter 1, Lesson 2
L2.IC.SLE.01 Debate laws and regulations that impact the development and use of software.	Supplemental Chapter 1, Lessons 2, 3
L1.IC.SLE.02 Describe and discuss the privacy concerns related to the large-scale collection and analysis of information about individuals (e.g., how websites collect and uses data) that may not be evident to users.	N/A
L1.IC.SLE.03 Evaluate the social and economic consequences of how law and ethics interact with digital aspects of privacy, data, property, information, and identity.	Supplemental Chapter 1