

CompuScholar, Inc.Alignment to Arkansas **Advanced Programming: Game Design** Standards**Arkansas Course Details:**

Course Title:	Advanced Programming: Game Design
Course Code(s):	465650 / 465660 (Level 1 / Level 2)
Grade Level:	9th - 12th Grade
Standards Link:	Advanced Programming 082016.pdf

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

The Arkansas Advanced Programming Course focuses on the skills necessary to design and develop reliable programs and software. Through these standards, students will explore, apply, and advance toward mastery of data structures and algorithms, data representation, algorithm design, and program efficiency. Students will accomplish tasks and solve problems independently and collaboratively with the tools and skills needed to be successful in college and careers. This document includes the Game Design supplemental requirements for Advanced Programming.

Course Standards

Strand: Computational Thinking and Problem Solving		CITATION(S)
Content Cluster 1: Students will analyze problem-solving strategies.		
Level 1	Level 2	
APL1.1.1 Judge the effectiveness of several different problem-solving strategies (e.g., look for a pattern, try a simpler case, work backwards) and determine the most effective solution for a given problem. To meet this standard, include physics and mathematics principles for game mechanics.	APL2.1.1 Continuation of this standard is not specifically included or excluded	Chapter 5 Chapter 21 Supplemental Chapter 3, Lesson 4

<p>APL1.1.2 Utilize an abstract model (e.g., computational model, computer model) to simulate a system by reproducing behavior of a system. To meet this standard, include physics and mathematics principles for game mechanics.</p>	<p>APL2.1.2 Continuation of this standard is not specifically included or excluded</p>	<p>Chapter 5 Chapter 21 Supplemental Chapter 3, Lesson 4</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------	--------------------------------------------------------------------------

Strand: Computational Thinking and Problem Solving		CITATION(S)
Content Cluster 2. Students will solve problems cooperatively and collaboratively.		
Level 1	Level 2	
<p>APL1.2.1 Utilize an appropriate development life cycle process (e.g., spiral, waterfall) as a member of a development team for a given project (e.g., community service project, real world) of level-appropriate complexity</p>	<p>APL2.2.1 Utilize an appropriate development life cycle process (e.g., spiral, waterfall) while building a development team for a given project (e.g., community service project, real world) of level-appropriate complexity</p>	<p>Chapter 25 Chapter 26</p>
<p>APL1.2.2 Observe global collaboration in the development of a computational artifact (e.g., review the process through which an open-source software project hosted on GitHub has improved over time)</p>	<p>APL2.2.2 Contribute to global collaboration in the development of a computational artifact (e.g., assist in resolving a bug in an open-source software project hosted on GitHub)</p>	<p>N/A</p>

Strand: Data and Information		CITATION(S)
Content Cluster 3. Students will analyze various ways in which data is represented.		
Level 1	Level 2	
<p>APL1.3.1 Compare the differences in ways level-appropriate data structures (e.g., graphs, linked lists, maps, queues, sets, stacks, trees) organize data</p>	<p>APL2.3.1 Compare the differences in ways level-appropriate data structures (e.g., graphs, linked lists, maps, queues, sets, stacks, trees) organize data</p>	<p>Chapter 9 Chapter 12</p>
<p>APL1.3.2 <i>This standard is not specifically required until Level 2</i></p>	<p>APL2.3.2 Compare media formats (e.g., graphics, sounds) for traits such as lossiness and compression performance</p>	<p>Chapter 18, Lesson 1 Chapter 23, Lesson 3</p>

Strand: Data and Information		CITATION(S)
Content Cluster 4. Students will collect, arrange, and represent data.		
Level 1	Level 2	
APL1.4.1 Select and use data structures (e.g., graphs, linked lists, maps, queues, sets, stacks, trees) based on functionality, storage, and performance tradeoffs	APL2.4.1 Implement data structures (e.g., graphs, linked lists, maps, queues, sets, stacks, trees) to support the creation of larger computational artifacts	N/A
APL1.4.2 Create and populate tables in databases	APL2.4.2 <i>Continuation of this standard is not specifically included or excluded</i>	N/A
APL1.4.3 <i>This standard is not specifically required until Level 2</i>	APL2.4.3 Use various data collection techniques for different types of problems (e.g., system sensors, mobile device GPS, open data sets, social media data sets, user surveys)	N/A

Strand: Data and Information		CITATION(S)
Content Cluster 5. Students will interpret and analyze data and information.		
Level 1	Level 2	
APL1.5.1 Issue queries against databases to glean meaning from stored data	APL2.5.1 Discuss real-world data sources that can be mined to produce new knowledge	Supplemental Chapter 3 Lesson 4

Strand: Algorithms and Programs		CITATION(S)
Content Cluster 6. Students will create, evaluate, and modify algorithms.		
Level 1	Level 2	
APL1.6.1 Evaluate multiple classical algorithms in terms of time and space complexities (e.g., Big O notation)	APL2.6.1 Evaluate multiple student-created algorithms in terms of time and space complexities (e.g., Big O notation)	N/A

APL1.6.2 Develop algorithms to solve student-identified problems of appropriate complexity	APL2.6.2 Develop algorithms to solve student-identified problems of appropriate complexity	Chapter 21
APL1.6.3 Decompose problems of appropriate complexity into well-defined steps to produce computational artifacts	APL2.6.3 Decompose problems of appropriate complexity into well-defined steps to produce computational artifacts	Chapter 9 Chapter 21

Strand: Algorithms and Programs		CITATION(S)
Content Cluster 7. Students will create programs to solve problems.		
Level 1	Level 2	
APL1.7.1 Demonstrate code reuse by creating programming solutions using level-appropriate libraries and application program interfaces (API) (e.g., graphics libraries, maps). To meet this standard, include the function of a game engine and supporting libraries.	APL2.7.1 Demonstrate code reuse by creating programming solutions using level-appropriate libraries and APIs (e.g., graphics libraries, maps). To meet this standard, include the function of a game engine and supporting libraries.	Unity SDK used throughout the course. See also: Chapter 9 Activity Chapter 15, Lesson 4
APL1.7.2 <i>This standard is not specifically required until Level 2</i>	APL2.7.2 Break a large system down into progressively smaller classes or objects that are responsible for some part of the problem domain	Chapter 9
APL1.7.3 Create programs to solve problems of level-appropriate complexity	APL2.7.3 Create programs to solve problems of level-appropriate complexity	Multiple programs created throughout the course

Strand: Computers and Communications		CITATION(S)
Content Cluster 8. Students will analyze various components and functions of computers		
Level 1	Level 2	
APL1.8.1 Investigate the functionality of various hardware components as they relate to modern game design (e.g., game controllers, GPU acceleration, input and output devices such as sensors, robotics components, virtual reality).	APL2.8.1 Integrate various hardware components (e.g., game controllers, input and output devices, robotics components, sensors) as they relate to student-developed computational artifacts	Chapter 4, Lesson 3

Strand: Computers and Communications		CITATION(S)
Content Cluster 9: Students will utilize appropriate hardware and software.		
Level 1	Level 2	
<p>APL1.9.1 Investigate the role and impact of laws and regulations on the development and use of software (e.g., privacy, security). To meet this standard, include game rating system (ERSB) and copyright laws concerning assets.</p>	<p>APL2.9.1 Clarify the role and impact of laws and ethical decisions of student-led projects</p>	<p>Supplemental Chapter 1 Supplemental Chapter 3, Lesson 3</p>
<p>APL1.9.2 Evaluate security issues that lead to compromised computer programs (e.g., circular references, lack of error checking and field size checking)</p>	<p>APL2.9.2 Implement security policies by comparing encryption and authentication strategies (e.g., safeguarding keys, secure coding)</p>	<p>Supplemental Chapter 1, Lesson 3</p>
<p>APL1.9.3 Use an ethical decision-making process to justify decisions made in creating computational artifacts</p>	<p>APL2.9.3 Defend an ethical decision-making process to justify decisions made in creating computational artifacts</p>	<p>Chapter 18, Lesson 1 Supplemental Chapter 1</p>

Content Cluster 10. Game Design Addendums		CITATION(S)
Level 1		
<p>APL1.10.1 Compare and contrast game elements present in board games and digital representations.</p>		<p>Chapter 13, Lesson 1</p>
<p>APL1.10.2 Describe how the parts of a game contribute to its overall behavior (e.g., addiction, balance, character, enjoyment factor, goal, immersion, interactive, level, movement, replay value, rules, score, story, theme, victory/goal conditions).</p>		<p>Chapter 13</p>

<p>APL1.10.3</p> <p>Describe the core areas of digital game design:</p> <ul style="list-style-type: none"> • Characters and Development • Core Mechanics • Gameplay Modes • Game World • Level Design • Mode Elaboration • Story Elaboration 	Chapter 13
<p>APL1.10.4</p> <p>Describe the concept of a game loop in digital games.</p>	Chapter 3, Lesson 4
<p>APL1.10.5</p> <p>Describe the functions of a game engine and supporting libraries (eg. images, sounds, sprites, text effects)</p>	Chapter 13, Lesson 1
<p>APL1.10.6</p> <p>Discuss common asset creation techniques (e.g., images, music, sounds, 3-D models)</p>	Chapter 1, Lesson 3 Chapter 17, Lesson 1 Chapter 18, Lesson 1 Chapter 23
<p>APL1.10.7</p> <p>Classify the roles and responsibilities of each member on a game design team</p> <ul style="list-style-type: none"> • Artist/Animation • Audio • Designer • Producer • Programmers • Project manager • Quality Assurance 	Chapter 25, Lesson 3
<p>APL1.10.8</p> <p>Summarize the history of games and their significance to digital games</p>	Supplemental Chapter 2
<p>APL1.10.9</p> <p>Summarize the role of play in human culture</p>	Supplemental Chapter 2
Level 2	
<p>APL2.10.1</p> <p>Design a game following the core areas of digital design (e.g. level maps, rules, script writing, storyboarding, storytelling).</p>	Chapter 14 Chapter 15 Chapter 26
<p>APL2.10.2</p> <p>Use concepts related to updating and drawing within the game loop.</p>	Chapter 3, Lesson 4
<p>APL2.10.3</p> <p>Utilize custom assets for a game.</p>	Chapter 18, Lesson 1 Chapter 23

APL2.10.4 Discuss diverse careers that are available for game designers and their educational requirements.	Supplemental Chapter 3, Lesson 5
APL2.10.5 Identify the hard and soft skills required to be an effective game design team member (e.g., analytical competence, creativity, initiative, punctuality, teamwork).	Chapter 25, Lesson 3 Supplemental Chapter 3, Lesson 5
APL2.10.6 Explore advancing and emerging technologies (e.g., artificially intelligent agents, robotics) as they relate to game design.	N/A