Zulama™

Powered by Carnegie Learning

AP Computer Science Principles

Detailed Syllabus

Zulama AP Computer Science Principles

Zulama™
Powered by Carnegie Learning

Course Overview

The Zulama AP Computer Science Principles Course is structured by the Conceptual Framework established by the College Board. This includes six Computational Thinking Practices: Skills and five Big Ideas.

The content of the Zulama AP Computer Science Principles course was originally written and structured by Ruth Comley, faculty at the Entertainment Technology Center at Carnegie Mellon University in Pittsburgh, Pennsylvania. GameMaker Studio 2 was chosen as the game engine to teach computer science principles due to its professional game design interface, high interest skill development, and functional GML scripting language. The computer game design principles and skills high school students develop using GameMaker Studio 2 interfaced also are taught in the over 400 university game design programs in the United States, both at the undergraduate and graduate levels.

Through a study of computer science principles and game design, students in the Zulama AP Computer Science Principles course:

- apply computer science principles and transferable skills by building playable digital games,
- program using GML scripting language and the GameMaker Studio™ interface,
- engage in creative design opportunities by applying knowledge of game mechanics,
- collaborate in design teams using the iterative design process,
- expand knowledge of the global impact of computer innovations,
- explore career and advanced education opportunities in the study of computer science and computer game design through the Explore Computer Innovations activity
- design and code a playable game that meets the expectations of the Performance Task
- prepare for the College Board AP CSP Exam

Pedagogy

This AP Computer Science Principles course develops computational thinking practices that students use to problem solve and to critically analyze innovations in computing. The course is appropriate for students who have completed a high

school algebra course. Students learn advanced computer science principles by completing rigorous computer game projects. They plan, design, code, and test software using the scripting language GML in GameMaker: Studio. Students gain a deep understanding of the global impact of the Internet through the study of game design, game programming, and the fast growing and diverse global video game industry.

This course emphasizes building computer science vocabulary and applying computer science principles and essential knowledge of coding practices. Students engage in a variety of activities where they design, code, iterate, and share playable games in a 2d environment defined through the course Big Ideas and Learning Objectives.

Total course time: 140 hours = approximately 37 weeks of instruction when delivered in 45 minute periods and 5 classroom periods per week. The course fits flexibly into other formats, such as block scheduling.

Course Delivery

Teachers and students each have their own login and password to access the digital curriculum through the browser-based Passport platform. This digital curriculum provides:

Features of Passport include:

- Content delivery
- Course scheduling
- Edulastic online assessment integration
- Real time gradebook
- Teacher to student messaging
- Teacher control of student rosters and passwords
- Teacher keys and other resources

Primary Programming Environment

GameMaker Studio 2

Zulama AP Computer Science Principles Resources

- Instructional Videos
- Assessment rubrics

- Module quizzes
- Digital Portfolio template

Teacher Resources

- Schell, Jesse. *The Art of Game Design: A Book of Lenses.* New York: CRC Press, 2008.
- Dale, Nell and Lewis, John. *Computer Science Illuminated*. Jones & Bartlett, 2015.
- Interactive book "How to Think Like a Computer Scientist: Interactive Edition" Runestone Interactive Project at Luther College

AP CSP Conceptual Framework

Computational Thinking Practices

CPT1: Computational Solution Design
CPT2: Algorithms and Program Development
CPT3: Abstraction in Program Development
CPT4: Code Analysis
CPT5: Computing Innovations
CPT6: Responsible Computing

Big Ideas

The AP Computer Science Principles course is built on five Big Ideas. For specific Enduring Understandings, Learning Objectives, and Essential Knowledge Statements that support each Big Idea, please refer to the College Board Course and Exam Description (CED) binder for AP CSP.

(CRD) Big Idea 1: Creative Development

When developing computing innovations, developers can use a formal, iterative design process or experimentation. While using either approach, developers will encounter phases of investigating and reflecting, designing, prototyping, and testing. Additionally, collaboration is an important tool to use at any phase of development because considering multiple perspectives allows for improvement of innovations

(DAT) Big Idea 2: Data

Data is central to computing innovations because it communicates initial conditions to programs and represents new knowledge. Computers consume data, transform data, and produce new data, allowing users to create new information or knowledge to solve problems through the interpretation of this data. Computers store data digitally, which means that the data must be manipulated in order to be presented in a useful way to the user.

(AAP) Big Idea 3: Algorithms and Programming

Programmers integrate algorithms and abstraction to create programs for creative purposes and to solve problems. Using multiple program statements in a specified order, making decisions, and repeating the same process multiple times are the building blocks of programs. Incorporating elements of abstraction, by breaking problems down into interacting pieces, each with their own purpose, makes writing complex programs easier. Programmers need to think algorithmically and use abstraction to define and interpret processes that are used in a program.

(CSN) Big Idea 4: Computing Systems and Networks

Computer systems and networks are used to transfer data. One of the largest and most commonly used networks is the Internet. Through a series of protocols, the Internet can be used to send and receive information and ideas throughout the world. Transferring and processing information can be slow when done on a single computer but leveraging multiple computers to do the work at the same time can significantly shorten the time it takes to complete tasks or solve problems.

(IOC) Big Idea 5: Impact of Computing

Computers and computing have revolutionized our lives. To use computing safely and responsibly, we need to be aware of privacy, security, and ethical issues. As programmers, we need to understand how our programs will be used and be responsible for the consequences. As computer users, we need to understand how to protect ourselves and our privacy when using a computer.

AP CSP Curricular Requirements

The Zulama AP CSP course supports each of the Big Ideas throughout the course content, collaborative and individual activities, and student reflections. The lesson activities support lesson content and provide opportunities for students to apply newly learned computer science concepts and skills. Thus, **students are expected to complete all lesson activities listed in the Module Guide**.

Module Guide

Module 1

Overview of Game Design and Computer Science Principles

Timeframe: 3 Lessons, approximately 16 hours

Module 1	
Big Ideas	CRD, DAT, AAP, CSN, IOC
Computational Thinking Practices	CTP 1, 2, 5, 6
Lessons	**Essential Questions**
Lesson 1: **Introduction Game Design** CRD, CP1; CP5 **Lesson 2**: **Data and Computational Thinking** CSN; CTP 2, 6 **Lesson 3**: **The Internet and Global Impact** CSN; CTP 6	• Why are games a computing innovation? • How is computational thinking used in game development? • What are the positive and negative impacts of computers and the Internet?
Computer Science / Game Design Topics	**Coding Concepts**
• Game design principles • Working collaboratively in design teams • Game mechanics	
Questions	• Module 1 Quiz • AP CSP Topic Formative Assessment
Outcomes	• Read binary data • Determine impact of computing • Analyze data collection and reporting • Know Internet characteristics • Gauge Internet usage • Evaluate cybersecurity issues

Instructional Activities

Lesson 1: **IDEA Teams and Computing Innovations** Students pretend their IDEA Teams were the creators of Side Scroller, and discuss ways in which their collaboration might have made different choices. **CRD**

Lesson 2: **Tell Time Using Binary Clocks** Students study the examples of binary clocks, read the Binary Data section in Lesson 1, convert the LED binary settings to decimal numbers so that they can confirm the times. Then they will also complete exercises to represent binary data using hexadecimal digits and make comparisons. **DAT**

Lesson 2: **Research the Impact of Games** Students research and create a multimedia presentation that explores the impact of the global game design industry on today's society. The presentation should include statistics on how games are used for the greater good and careers in gaming as well as data of the student's choice. Sources should be cited and students should organize the data in a clear way. **IOC**

Lesson 2: **Using Data for Information and Knowledge** Students describe an app they use frequently and write a short summary of its purpose. Students give an example of data that it collects and how it uses and presents the data back to the user. Students share with the class what information or knowledge is gained from the reporting. **DAT**

Lesson 2: **Database Design Problems: The Card Game** Students consider how to best design a database for users. They consider search factors, variable, and how to simplify a database. Students consider the problem of scalability and develop a working blueprint that could be used to design a preliminary test site. **DAT**

Lesson 3: **Compare Usage** Students compare their Internet usage to what statistics indicate and support their comparison with a graphic. **CSN**

Lesson 3: **Unauthorized Access** Students research how unauthorized access to computing resources is gained and how they can stop or prevent that access. **CSN**

Lesson 3: **Data and Security** Students essay on the benefits and dangers of collecting and storing personal data on online databases. **CSN**

Lesson 3: **Crowdsourcing and Large-Scale Problem Solving** Students choose a crowdsourced research project online and explain how the people involved are participating in large-scale problem solving. **CSN**

Lesson 3: **Ethical Use of Computers** Students access their favorite search engine and search for articles related to the harmful effects of computing.

After reading and summarizing the article, students discuss ways to counteract the effect. **CSN**

Lesson 3: **Bringing Equity** Students research equity and how it can be achieved in computing. Students write an essay summarizing their findings. **IOC**

Lesson 3: **Think Beyond the Code** Students research an example of a computing innovation that ended up having consequences that the programmers probably didn't intend and decide how they might have done things differently. **CRN**

Lesson 3: **Good and Bad** Students research a well-known computing innovation and discuss both the pros and the cons of that innovation for society. **CRD**

Lesson 3: **Internet Usage** Students share their Internet usage habits, including how much time is spent per day on the Internet and how it is spent. Students describe their use, such as playing games, accessing social media, using it for school-related projects, watching videos, or independently learning. **CSN**

Lesson 3: **Internet Characteristics** Students work in pairs and choose a topic from the Digital Ocean article to investigate and prepare to teach their classmates in a short presentation. Topics chosen should include IP addresses, protocols, and packets. **CSN**

Lesson 3: **Impact of the Internet** Students explain how the Internet has changed how individuals communicate and learn. **IOC**

Lesson 3: **Cybersecurity** Students read a selection of articles pertaining to cybersecurity from looking at encryption to recent security breaches. They respond to a writing prompt and describe a current cybersecurity concern and consider ways to address the issue. **CSN**

Lesson 3: **Computer Vocabulary Word Art** Students choose 15 words from their reading that describe computer science. They find a word cloud generator of their choice to complete the assignment. *Wordle* is one option, but others are available. Students then design a creative word art showcasing the computer science vocabulary found in Module 1. **CRD**

Lessons 1 - 3: **Game Design Journal** Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Instructional Resources

- **Article:** An Introduction to Networking Terminology, Interfaces, and Protocols, online DigitalOcean

https://www.digitalocean.com/community/tutorials/an-introduction-to-networking-terminology-interfaces-and-protocols

- **Article:** Internet Protocol (http://searchunifiedcommunications.techtarget.com/definition/Internet-Protocol)

Module 2

Using GameMaker: Studio

Timeframe: 3 Lessons -- Approximately 4 hours

Module 2	
Big Ideas	CRD, DAT, CSN, IOC
Computational Thinking Practices	CTP 1, 3, 5, 6
Lessons	**Essential Questions**
• **Lesson 4: Setting Up GameMaker Projects** CRD; CTP 1 • **Lesson 5: GameMaker: Studio Interface** CSN; CTP 5, 6 • **Lesson 6: Game Assets and GameMaker** DAT, CSN; CTP 3	• How is a GameMaker project stored? • What does the GameMaker interface look like? • How are assets and projects transferred?
Computer Science / Game Design Topics	**Coding Concepts**
• Downloading and installing software (GameMaker: Studio) • Project file structure	
Questions	• Module 2 Quiz • AP CSP Topic Formative Assessment
Outcomes	• Set up file structures • Explain software releases and version numbers • Navigate GameMaker: Studio interface • Use file compression formats

Instructional Activities

Lesson 6: Storing, Securing, and Compressing Data Students work in small, collaborative groups to research and create a multimedia presentation on compressed file formats. Include historical information, how data is secured, and technical information. **DAT**

Lesson 6: Investigating Data Sets Students work individually or in pairs to investigate metadata. Use file explorer and GameMaker project files to draw conclusions. **DAT**

Lesson 6: Computing Innovations Through the first two modules students have been exploring how computing has evolved and had an impact. Now they will look beyond video games and research how computing, and shared access to resources, has impacted innovation in other fields. Students present their findings in the form of a digital poster. **CSN**

Lessons 4 - 6: Game Design Journal Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Instructional Resources

- **Instructional Video:** How to Extract Zip Files

Module 3

Zulama Pinball

Timeframe: 5 Lessons -- Approximately 5 hours

Module 3	
Big Ideas	CRD, DAT, AAP, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4, 5, 6
Lessons	**Essential Questions**
• **Lesson 7**: **Game Design Documents** CRD; CTP 1, 5 • **Lesson 8**: **Parts of a GameMaker Game** DAT; CTP 3, 5	• What does a game document include? • What are the main resources used in a GameMaker Studio project? • What resources are used in a game level?

• **Lesson 9: Backgrounds and Rooms** CRD, DAT; CTP 2, 3 • **Lesson 10: Adding Code** AAP; CTP 2, 3 • **Lesson 11: Complete Navigation Workshop** CRD, DAT, CSN; CTP 1, 4, 6	• How do objects react to events? • How is room navigation handled?
Computer Science / Game Design Topics	**Coding Concepts**
• One page design • Game document • Pair programming • Create GameMaker project • Sprites • Objects • instance	
Questions	• Module 3 Quiz • AP CSP Topic Formative Assessment

Outcomes	• Manipulate the GameMaker interface
	• Add rooms and backgrounds
	• Determine game flow
	• Apply GML scripting skills, including:
	  • Create events and collisions
	  • Add functions and variables

Instructional Activities

Lesson 8: Designing New Playing Pieces Students work as a design team to add two new playing pieces to the game. Design team may be pair programmers or two pairs may form one team. This activity will begin here and will finish at the end of Zulama Pinball, and includes design of game art, application of game mechanics, and adding objects to the computer game environment. **CRD**

Lesson 10: Upload Completed Project File When students have finished all the tasks described in the lessons and tested that their game works, students export their GameMaker project and upload the GMZ file for review. CRD, **DAT**

Lesson 10: Levels of Abstraction Students respond to a writing prompt to discuss the multiple levels of abstraction they have seen from the game document to running their code (highest to lowest - rule set, pseudocode, GML code, compiled code, executed code.) **AAP**

Lesson 11: Upload Navigation Workshop Students export and upload their completed Navigation Workshop GMZ file. This Workshop includes making independent changes to the game to assess module outcomes **CRD, DAT**

Lessons 7 - 11: Game Design Journal Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Instructional Resources

- **Instructional Video:** Creating a Sprite
- **Instructional Video:** Creating an Object
- **Instructional Video:** Creating a Background
- **Instructional Video:** Creating a Room / Adding Objects
- **Instructional Video:** Writing Code
- **Instructional Video:** Exporting GMZ Files

Module 4

Making the Game Work

Timeframe: 4 Lessons -- Approximately 6 hours

Module 4	
Big Ideas	CRD, DAT, AAP, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4
Lessons	**Essential Questions**
• **Lesson 12: Controlling the Paddle** AAP; CTP 1, 2, 3, 4 • **Lesson 13:** Using Mouse input AAP; CTP 1, 2, 3, 4	• How is the paddle movement controlled by the player? • What events are generated by mouse clicks?

• **Lesson 14:** Collision with Ball AAP; CTP 1, 2, 3, 4 • **Lesson 15: Adding More Assets** CRD, DAT; CTP 1, 2, 3, 4	• How are program errors interpreted and corrected? • Why is the User Interface important?
Computer Science / Game Design Topics	**Coding Concepts**
• Keyboard events • Mouse events • Randomization • Collision handling • Debugging • User interface • Draw events • Font resource	• Conditional statement • Random number functions • Boolean variables • Variable scope • Global variables • Constants • "with" statement • Interpreting error messages • Draw functions
Questions	• Module 4 Quiz • AP CSP Topic Formative Assessment

Outcomes	• Apply GML scripting skills
	  ○ Use conditional statements, random number functions, variables, constants, and variable scope
	  ○ Dynamically allocate game resources

Instructional Activities

Lesson 13: Practice with if Statements Students complete an if statement worksheet to assess their ability to trace logical programming statements. **AAP**

Lesson 14: GameMaker Tips Students respond to the writing prompt by posting to an online discussion thread: What GameMaker shortcuts have you discovered on your own that you find helpful? What web searches have you done and what online resources have helped you? How did you verify their credibility? Students then respond to a partner's suggestion after trying it out or using the online reference. **CRD, CSN**

Lesson 14: **Debugging Exercise** Students work as pair programmers to correct all errors in given project. They then export and upload it to the Zulama Learning and Content Management System. **AAP**

Lesson 15: Is the Game Fair? Students provide written answers to questions posed to address game design, iteration, and game balance. **CRD**

Lesson 15: Module 4 Guided Lessons Completed When students have finished all the tasks described in the lessons, tested, and iterated their game so they know it works correctly, students export their GameMaker project and upload the GMZ file to the Zulama Learning and Content Management System for review. **CRD, DAT**

Lessons 12 - 15: **Game Design Journal** Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Instructional Resources

- **Instructional Video:** Using Keyboard Events
- **Instructional Video:** Adding Code to Respond to Mouse Input
- **Instructional Video:** Adding a Rule
- **Instructional Video:** Making Objects Operational
- **Instructional Video:** Drawing the Score

Module 5

Finishing Zulama Pinball

Timeframe: 5 Lessons -- Approximately 8 hours

Module 5	
Big Ideas	CRD, DAT, AAP, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4
Lessons	**Essential Questions**
• **Lesson 16: Add Game Balance** AAP; CTP 1, 2, 3, 4, • **Lesson 17: Add Reward**s APP; CTP 1, 2, 3, 4 • **Lesson 18: Player Feedback** AAP; CTP 2, 3, 4 • **Lesson 19: Designing a Level** CRD, AAP; CTP 2, 3, 4 • **Lesson 20: Final Playtest** CRD, IOC; CTP 4	• What is game balance and how can it be achieved in your game? • How can variables be used to control game flow? • How is winning or losing determined and communicated to the player? • What needs to be considered to build a second level? • What does playtesting involve? • Why is playtesting a crucial step in game design?
Computer Science / Game Design Topics	**Coding Concepts**
• Game balance • Dynamic resource allocation • Building and testing game feedback • Game development cycle • Working collaboratively in a design team • Game design principles • Playtest	• Conditional statement (if-else) • Increment and decrement operators • Compound if statement • Global variables • Develop and test • Create application executable
Questions	• Module 5 Quiz • AP CSP Topic Formative Assessment
Outcomes	• Apply GML scripting skills

	○ Use compound conditional statements, increment and decrement operators, and variable scope ○ Dynamically allocate game resources ● Engage in the iterative process to design a second game level ● Playtest a game

Instructional Activities

Lesson 17: Add More Balance Students choose one of several brainstormed ideas from Lesson 15 and implement it in the game. Students consider choosing an original idea that will make the game more interesting but also take into account a change that can be successfully implemented with the deadlines set by the teacher. The explanation and file are uploaded for teacher review. **AAP**

Lesson 18: Guided Lessons Completed After finishing all tasks directed in the lessons, testing the game, and iterating as needed so the game works, export their GameMaker project. Students upload the GMZ file for teacher review. **CRD, DAT**

Lesson 19: Designing a Level Workshop Before making the changes in GameMaker, planning needs to be done. Students document their plans to implement an original level by completing a one page design document as well as adding to the game document used throughout Zulama Pinball. Updating the game document includes writing pseudocode. Students upload their completed planning documents for teacher review. **CRD, DAT**

Lesson 19: New Level GMZ Workshop Once students have coded and tested their design changes, they export and upload their completed GMZ GameMaker project file for teacher review. **CRD, DAT**

Lesson 20: Evaluate the Playtest Students work in their design teams to evaluate the test results and brainstorm ideas for changes to games to make them more playable and fun. Students then provide written responses to writing prompts. **CRD**

Lessons 16 - 20: Game Design Journal Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Instructional Resources

- **Instructional Video:** Adding Player Lives
- **Instructional Video:** Adding a Bonus Feature, part 1
- **Instructional Video:** Adding a Bonus Feature, part 2
- **Instructional Video:** Win Conditions and Feedback
- **Instructional Video:** Making the Play Again Button Functional

Module 6

Ball Bouncer

Timeframe: 4 Lessons -- Approximately 6 hours

Module 6	
Big Ideas	CRD, DAT, AAP, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4
Lessons	**Essential Questions**
• **Lesson 21: Rooms and Backgrounds** AAP; CTP 1, 2, 3, 4 • **Lesson 22: Ball and Wall Objects** AAP; CTP 1, 2, 3, 4 • **Lesson 23: Adding the Goa**l AAP; CTP 1, 2, 3, 4 • **Lesson 24: Create Playing Pieces** **Workshop** CRD, DAT; CTP 1, 2, 3, 4	• What is required to set up game navigation when you begin a new game? • How can instances be added easily to a room? • How is the game won? • What steps are required to set up the game's playing pieces?
Computer Science / Game Design Topics	**Coding Concepts**
• Pair Programming • Create GameMaker project • Set up game flow	• Using functions • Random number functions

• Level design • Object instances • Randomization • Create game objects • Project organization	
Questions	• Module 6 Quiz • AP CSP Topic Formative Assessment

Outcomes	• Apply GML scripting skills
	  ○ Set up game flow in the Ball Bouncer game
	  ○ Use random number functions
	  ○ Use create events and collision events to control game flow

Instructional Activities

Lesson 21: Ball Bouncer Room Navigation Students set up the room navigation for their game according to the game document and upload their GMZ file for teacher review. **CRD, AAP, DAT**

Lesson 23: Guided Lessons Completed Student files show that room navigation, including moving to the end room when the ball hits the goal is complete. Students export and upload their tested game GMZ file for teacher review. **CRD, DAT**

Lesson 24: Create Playing Pieces Workshop Game flow should begin at the start screen and move to the main level when the play button is clicked. All five playing piece objects should be created and placed in the main room at the appropriate location in the bin area. Once students have completed the lessons and successfully playtested their game, they need to export the GMZ file and upload it for teacher review. **CRD, AAP, DAT**

Lessons 21 - 24: Game Design Journal Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Instructional Resources

- **Instructional Video:** Creating More Sprites

- **Instructional Video:** Coding Blocks
- **Instructional Video:** Incorporating the Goal

Module 7

Ball Bouncer Game Mechanics

Timeframe: 6 Lessons -- Approximately 10 hours

Module 7	
Big Ideas	CRD, DAT, AAP, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4
Lessons	**Essential Questions**
• **Lesson 25**: **Placing the Playing Pieces** AAP; CTP 1, 2, 3, 4 • **Lesson 26: Taking a Closer Look** AAP; CTP 1, 2, 3, 4 • **Lesson 27**: **User Interface** AAP, DAT; CTP 1, 2, 3, 4 • **Lesson 28: Global Variables** AAP; CTP 2, 3, 4 • **Lesson 29: Adding a Power Up** AAP; CTP 2, 3, 4 • **Lesson 30: On Your Own Workshop** CRD, DAT, IOC; CTP 1, 2, 3, 4	• How can Boolean variables be used to control dragging and dropping playing pieces? • What fundamentals are used throughout games programming? • What game elements need to be added to complete level design? • Why is a global variable needed? • What is the purpose of a power-up? • How does a power-up add interest to the game? • What should you be evaluating when playtesting your game?
Computer Science / Game Design Topics	**Coding Concepts**
• Program logic • Boolean logic • Truth tables • User Interface • Game development cycle	• Boolean variables • Compound conditionals • Coordinate math • Variable scope • Variable types • Constants • Objects and instances

	• Nested if statements • Logical operators • Dot notation
Questions	• Module 7 Quiz • AP CSP Topic Formative Assessment

Outcomes	• Apply GML Scripting skills
	  ○ Use Boolean variables, Boolean logic, constants, and nested if statements
	  ○ Make decisions using computations based on instance's coordinates
	• Engage in the iterative process to test and debug the game

Instructional Activities

Lesson 25: The Rest of Pieces Students modify other playing pieces using the diamond object as a guide. Students playtest, address errors as needed, and upload the GMZ file for teacher review. **AAP**

Lesson 26: Variables and Conditionals Practice Students provide written answers to a series of questions to show their code tracing prowess. They describe the error or determine the outcome when the code shown is run. **AAP**

Lesson 28: Relational Operators Students evaluate (and write) expressions using relational operators. **AAP**

Lesson 28: Instances and Variables Students read a scenario that analyzes instance vs. global variables. They then create a graphic organizer or multimedia presentation to demonstrate their understanding. Their visual, in turn, can be shared and discussed with other students to lead to deeper understanding. **AAP**

Lesson 28: Progress Check Students evaluate game elements and game mechanics that they feel would improve the game. Students identify these elements and provide a written plan for iterations. **CRD**

Lesson 29: How to Prevent Paddle from Rotating into Wall This is an online discussion question that has students suggesting ways to prevent the paddle in the game from rotating into one of the four walls. This activity gives

students an opportunity to brainstorm and sort through possible iterations for game improvement. **AAP**

Lesson 29: Module 7 Guided Lessons Completed Once all Module 7 game mechanics have been coded and tested, students export and upload their completed GMZ file for teacher review. CRD, DAT

Lesson 30: Make It Your Own Students complete the "Finish the Game" tasks. Then they choose at least three enhancements from their Game Design Journal or the Make it Your Own section to implement. **CRD**

Lesson 30: Is it Ready for Prime Time? Through a writing prompt, students evaluate their game and consider changes that would be effective to reach a wider audience and be required for widespread distribution. **CRD**

Lessons 25 - 30: Game Design Journal Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Instructional Resources

- **Instructional Video:** The Drag Function, part 1
- **Instructional Video:** The Drag Function, part 2
- **Instructional Video:** Creating User Interface
- **Instructional Video:** The Goal Top
- **Instructional Video:** Limiting Player Cards
- **Instructional Video:** Adding Power-Ups
- **Instructional Video:** Fixing the Bug

Module 8

Matching Game

Timeframe: 3 Lessons -- Approximately 5 hours

Module 8	
Big Ideas	CRD, DAT, AAP, CSN, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4
Lessons	**Essential Questions**
• **Lesson 31**: **Matching Game Setup** AAP; CTP 1, 2, 3, 4 • **Lesson 32**: **Card Sprites** CRD, AAP; CTP 2, 3, 4 • **Lesson 33: Controller Object** AAP; CTP 2, 3, 4	• How is a simple, matching card game programmed? • What is the purpose of one sprite containing multiple images? • What is the purpose of the controller object?
Computer Science / Game Design Topics	**Coding Concepts**
• Pair Programming • Set up game flow • Game timing • Controller object	• Using alarms • Using functions • Coordinate math
Questions	• Module 8 Quiz • AP CSP Topic Formative Assessment

Outcomes	• Apply GML Scripting skills
	  ○ Manage variables at an advanced level
	  ○ Use the controller object
	  ○ Program alarms and timers, and turning cards over by changing sprites

Instructional Activities

Lesson 33: **Research Simulation Games Activity** Video games simulations are widespread. Students research examples of how video game simulations are used for training purposes in and create a multimedia presentation to report their findings. **CRD, CSN, IOC**

Lesson 33: **Module 8 Guided Lessons Completed** Students upload their fully playtested Matching game GMZ file for teacher review. **CRD, DA**T

Lessons 31 - 33: **Game Design Journal** Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Instructional Resources

- **Instructional Video:** Creating the Card Sprite
- **Instructional Video:** Creating the Alarms

Module 9

Finding Matches
Timeframe: 5 Lessons -- Approximately 9 hours
Explore - Impact of Computing Innovation - 8 hours

Module 9	
Big Ideas	CRD, DAT, AAP, CSN, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4, 5, 6
Computing Innovations	Explore - Impact of Computing Innovations
Lessons	**Essential Questions**
• **Lesson 34: Managing Variables** AAP, CTP 2, 3, 4 • **Lesson 35: Game Timing** AAP; CTP 2, 3, 4 • **Lesson 36**: **Randomizing the Game** AAP; CTP 2, 3, 4	• How are matches determined and what role do variables play in controlling the game? • How are alarms used to handle game timing? • How can playability be improved?

• **Lesson 37: Game Improvements** CRD, IOC; CTP 1, 4 • **Lesson 38**: **Level Up** AAP; CTP 2, 3, 4	• What improvements can be made to the matching game? • What changes are necessary to change a sprite's subimages?
Computer Science / Game Design Topics	**Coding Concepts**
• Flowcharts • Pseudocode	• Conditionals • Global variables
Questions	• Module 9 Quiz • AP CSP Topic Formative Assessment

Outcomes	• Apply GML scripting skills
	  ○ Use conditionals and modulus operator
	  ○ Apply randomization to the game
	  ○ Use alarms
	• Engage in the iterative process to test and debug the game
	• Gain a deep understanding of computer innovations

Instructional Activities

Lesson 36: **Module 9 Guided Lessons Completed** Students upload their fully playtested Matching game GMZ file for teacher review. This represents the game version following completion of the guided lessons. **CRD, DAT, CSN**

Lesson 37: **Finish the Basic Game Workshop** Students upload their completed Matching Game GMZ file. Teacher review includes whether the UI is properly set up, gives appropriate feedback, and handles special test cases appropriately. **CRD, DAT, CSN**

Lesson 38: **Edit Sprites Workshop** Students add new images to their Matching game. Students access Creative Commons images to understand how computing has enabled broader access. Images should be resized and edited so that the game has a unique theme. Once inserted into the game and

playtested students should upload their GMZ file for teacher review. **CRD, DAT, CSN**

Lessons 34 - 38: **Game Design Journal** Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

At end of Module 9: **Explore - Impact of Computing Innovations** Students complete the Computing Innovations activities utilizing 8 hours of class time. Students should use the AP rubric and task guidelines to ensure all requirements are met. **CSN, IOC; CTP 1, 5, 6**

Instructional Resources

- **Instructional Video:** Checking for Matches
- **Instructional Video:** Fixing the Win Condition
- **Instructional Video:** Random Card Placement

Module 10

31 Game Setup

Timeframe: 6 Lessons -- Approximately 8 hours

Module 10	
Big Ideas	CRD, DAT, AAP, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4
Lessons	**Essential Questions**
• **Lesson 39**: **Set Up Playing Board** CRD, DAT; CTP 1, 2, 3, 4 • **Lesson 40: For Loop** AAP; CTP 2, 3, 4 • **Lesson 41: Arrays** AAP; CTP 2, 3, 4 • **Lesson 42: Managing the Deck** AAP; 1, 2, 3, 4	• What is required to set up game navigation when a game is started? • How can lines of code be repeated? • How are arrays used in programming? • How is the deck of cards represented virtually? • What programming is necessary to be able to score the cards?

• **Lesson 43: More Scripts** AAP; CTP 1, 2, 3 • **Lesson 44: Deal the Hand Workshop** DAT; CTP 1	• What planning and programming is necessary to set up the cards?
Computer Science / Game Design Topics	**Coding Concepts**
• Pair Programming • Set up game flow • Loops • Arrays • Writing algorithms • Implement game mechanics	• For loops • Arrays • Scripts • Nested for loops • Scripts (user-defined functions) • Alarms
Questions	• Module 10 Quiz • AP CSP Topic Formative Assessment
Outcomes	• Apply GML scripting skills ○ Demonstrate game flow in the Project 31 game ○ Describe data structures ○ Create one and two dimensional arrays, and loops ○ Identify variable scope ○ Use debug messages to test data structure ○ Consider additional detail on for loops and array indexing ○ Create and use GML scripts ■ Passing arguments ■ Returning values

Instructional Activities

Lesson 39: Playing 31 This activity provides an opportunity for students to post and respond to a discussion thread explaining their analysis of game mechanics used in the traditional card game 31 and how they will be applied in creating a digital version of the game. **CRD, IOC**

Lesson 41: **Manipulating Arrays** In this assignment students have the opportunity to use array notation to access cells of tables and to manipulate the contents. **AAP**

Lesson 43: **Module 10 Guided Lessons Completed** After thoroughly playtesting their game, students upload their GMZ file for teacher review. Functionality includes setup of room navigations and one card used for testing. **CRD, DAT**

Lesson 44: **Deal the Hand** Students have written code to set up room navigation, build a virtual deck, create objects for the player and opponent's hands, and dealt the first card hand. After thoroughly playtesting their game, students export and upload their GMZ file for teacher review. **AAP, DAT**

Lessons 39 - 44: **Game Design Journal** Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Instructional Resources

- **Instructional Video:** How to Play Scat: Card Games
- **Instructional Video:** Setting the Sprites
- **Instructional Video:** For Loop
- **Instructional Video:** Nested For Loop
- **Instructional Video:** Building the Deck
- **Instructional Video:** Swapping Two Cards
- **Instructional Video:** Dealing the Cards

Module 11

Build 31

Timeframe: 4 Lessons -- Approximately 8 hours

Module 11	
Big Ideas	CRD, DAT, AAP, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4
Lessons	**Essential Questions**
• **Lesson 45**: **The Player's Turn** AAP; CTP 2, 3, 4 • **Lesson 46**: **The Computer's Turn** AAP; CTP 2, 3, 4 • **Lesson 47: End the Hand** AAP; CTP 2, 3, 4 • **Lesson 48: Finish the Game** DAT; CTP 1	• What steps are required to code the player's turn? • What steps are necessary for the computer to take a turn? • What needs to be done to end a hand? • What improvements can be made to 31?
Computer Science / Game Design Topics	**Coding Concepts**
• Implement game mechanics • Implement game feedback • Build and test feedback and game timing	• Scripts (user-defined functions) • Alarms • Develop and test
Questions	• Module 11 Quiz • AP CSP Topic Formative Assessment
Outcomes	• Utilize key elements of a game design document • Implement the game development process • Apply gained knowledge to build the Project 31 game

	• Apply iteration to debug the Project 31 game

Instructional Activities

Lesson 48: **31 Game Project** Students have created a digital game version of the card game 31. Once tested and all errors have been corrected, students export their project and upload the compressed file for teacher review. Students also describe the original change made to the game. **CRD, DAT, AAP**

Lesson 48: **31 Game Development Process Reflection** Students write an essay about the process they followed to write the card game 31. Their essay should include detailed explanation of a section of their code, an overview of their project plan, and challenges during testing and how they overcame them. **CRD, IOC**

Lessons 45 - 48: **Game Design Journal** Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Instructional Resources

- **Instructional Video:** Discard
- **Instructional Video:** Selecting a Card from Discard

Module 12

Sky is Falling Cut Scene

Timeframe: 5 Lessons -- Approximately 7 hours

Module 12	
Big Ideas	CRD, DAT, AAP, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4
Lessons	**Essential Questions**

• **Lesson 49: Evolution of Transmedia World** IOC; CTP 1 • **Lesson 50: Begin the Cut Scene** CRD, AAP; CTP 1, 2, 3 ,4 • **Lesson 51: Construct the Timeline** CRD, AAP; CTP 1, 2 • **Lesson 52: Finish the Cut Scene**, AAP; CTP 2, 3, 4, • **Lesson 53: Cut Scene Workshop**, DAT, IOC; CTP 1	• How is game navigation set up? • How is a timeline used in GameMaker? • What is a persistent object and when should it be used? • How can horizontal and vertical speed be controlled and used? • What can be done to make transitions smoother?
Computer Science / Game Design Topics	**Coding Concepts**
• Set up game flow • Use animation • Use timelines • Image alpha levels • Persistent objects	• Functions • Alarms • Coordinate math
Questions	• Module 12 Quiz • AP CSP Topic Formative Assessment
Outcomes	• Use cut scenes • Use timelines to control when something happens • Use animation • Use layers and depth

Instructional Activities

Lesson 50: **Make Lightning Move with Cloud** Students revisit the cut scene previously coded where lightning only strikes when a cloud is not moving. Students change this so that the lightning object moves with the cloud object. **CRD, AAP**

Lesson 53: **Refine the Cut Scene** Once students complete the cut scene they evaluate the timing and add additional steps to improve the scene. After coding and testing, they export and upload their The Sky is Falling GMZ file for teacher review. **CRD, AAP, DAT**

Lessons 49 - 53: **Game Design Journal** Students respond to writing prompts to make entries to their online Game Design Journal that details how they

engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Module 13

Sky is Falling Game

Timeframe: 5 Lessons -- Approximately 4 hours

Module 13	
Big Ideas	CRD, DAT, AAP, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4
Lessons	**Essential Questions**
• **Lesson 54: Animals Fall** AAP; CTP 2, 3, 4 • **Lesson 55: Saving the Animals** AAP; CTP 1, 2, 3, 4 • **Lesson 56: Game Timers and UI** AAP; CTP 1, 2 3, 4 • **Lesson 57: Restarting the Game** AAP: 2, 3, 4; CTP 2, 3, 4	• How are the animal's game mechanics implemented in code? • How is code used to move two instances in sync? • How can feedback be added to the game to let the players know how they are doing? • What variables must be reset when the game is restarted?
Computer Science / Game Design Topics	**Coding Concepts**
• Implement game mechanics • Controlling spawning using percentages • Build and test feedback and game timing	• Using dot notation • Alarms • Conditionals • Functions • Switch statements • Develop and test
Questions	• Module 13 Quiz • AP CSP Topic Formative Assessment
Outcomes	• Use conditionals at an advanced level

	• Move one instance with another instance • Create effective User Interface

Instructional Activities

Lesson 55: **Captured Animals Investigation** Students analyze the code that controls an animal's collision with the boat. They make suggested changes to the conditional checks to investigate the effect on playing the game. Through a writing prompt, they report their findings. **CRD, DAT**

Lesson 55: **Compare Search Algorithms** Students compare linear and binary searches. The algorithms, conditions for use, and efficiency comparisons are investigated. **DAT**

Lesson 56: **Can the Computer Solve all Problems?** Computational problems, that is. Students use given sources, or research their own, to investigate what it means to solve a problem in reasonable time, to be unsolvable, or to be undecidable. **IOC**

Lesson 57: **Play Again Workshop** Once all functionality is working from following the guided lessons, students add code to restart the game. After testing they export and upload their The Sky is Falling GMZ file for teacher review. **CRD, AAP, DAT**

Lessons 54 - 57: **Game Design Journal** Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Module 14

Sky is Falling Enhancements

Timeframe: 5 Lessons -- Approximately 8 hours

Module 14	
Big Ideas	CRD, DAT, AAP, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4, 5

Computing Innovations	
Lessons	**Essential Questions**
• **Lesson 58: Lives and Capacity Bar** AAP; CTP 1, 2, 3, 4 • **Lesson 59: Boat Control** AAP; CTP 2, 3, 4 • **Lesson 60: Creating Paths** AAP; CTP 2, 3, 4 • **Lesson 61: Sound** AAP**,** DAT; CTP 1, 2, 3, 4, 5 • **Lesson 62: Create an Original Cut Scene** CRD, AAP; CTP 1, 2, 3, 4	• How can online communities and coding forums help when adding new features? • What factors should affect the speed of the boat? • How are path resources used? • How is sound added to a game? • What steps are necessary to create an original cut scene?
Computer Science / Game Design Topics	**Coding Concepts**
• Use coding forums • Use sound in a game • Game Development cycle	• Sound functions • Control animations • Develop and test
Questions	• Module 14 Quiz • AP CSP Topic Formative Assessment
Outcomes	• Use sound in a game• Use paths in a game • Engage in the iterative process to design an original cut scene

Instructional Activities:

Lesson 59: **Improving Boat Control** Students make improvements to how the boat's movement and height is controlled and report on the effect of their changes. **CRD, AAP**

Lesson 61: **Game Sound Research** Students research different sound file formats and create a digital presentation to report what they have learned. **IOC**

Lesson 61: **Sound in Games You Play** This activity provides an opportunity for students to post and respond to a discussion thread talking about how sound is used in games they play. **IOC**

Lesson 61: **The Sky is Falling Completed Game** Students upload their completed Sky is Falling Game GMZ file. Teacher review includes whether the game mechanics work properly, the sun moves along a path, and sound is used effectively in the game. **CRD, AAP, DAT**

Lesson 62: **Original Cut Scene Design Document** Students document their plans to design and code an original level by completing a one page design document. Students upload their completed planning document for teacher review. **CRD, DAT**

Lesson 62: **Create Original Cut Scene** Once students have coded and tested their cut scene they export and upload their completed GMZ GameMaker project file for teacher review. **AAP, DAT**

Lesson 62: **Cut Scene Development Reflection** Students will write an essay about the process they followed to write the original cut scene. Their essay should include detailed explanation of a section of their code and challenges during testing and how they overcame them. **CRD, IOC**

Lessons 58 - 62: **Game Design Journal** Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Module 15

Create Performance Task

Timeframe: 1 Lesson -- 12 hours

Create - Applications from Ideas Performance Task (12 hours class time)

Module 15	
Big Ideas	CRD, DAT, AAP, CSN, IOC
Computational Thinking Practices	CTP 1, 2, 3, 4,5, 6
Computing Innovations	Create Performance Task
Lessons	**Essential Questions**
• **Lesson 63:** Overview of Create Performance Task CRD, DAT, AAP; CTP 1, 2, 3, 4, 5, 6	• What are the guidelines for the Performance Task: Create - Applications from Ideas?

Computer Science / Game Design Topics	Coding Concepts
• Iterate through the Software development cycle to create a digital product	• Summative/authentic assessment of coding concepts and skills learned during the course
Questions	• AP CSP Topic Formative Assessment
Outcomes	• Use design skills acquired in the course • Use coding concepts and skills acquired in the course • Create a digital artifact

Instructional Activities:

Lesson 63: **Create - Applications from Ideas Performance Task** Students complete the through-course assessment utilizing at least 12 hours of class time. Students should use the AP rubric and task guidelines to ensure all requirements are met. **CRD, DAT, AAP**

Lesson 63: **Create Performance Task Game Design Journal** Students write and essay detailing their notes on the process as shown in their Game Design Journals. **CRD, IOC**

Lesson 63: **Create Performance Task Review** Students upload their completed game for teacher review. **CSN, IOC**

Lesson 63: **Game Design Journal** Students respond to writing prompts to make entries to their online Game Design Journal that details how they engage in the idea - design - iterate process, code, and analyze digital artifacts process. **CRD, IOC**

Module 16 (post APCSP exam)

Game Design Digital Portfolio Website

Timeframe: 2 Lessons -- Approximately 16 hours

Module 16	
Big Ideas	CRD, CSN, IOC
Computational Thinking Practices	CTP 1, 5, 6
Lessons	**Essential Questions**
• **Lesson 64: Digital Portfolio Website** CRD, CSN, IOC; CTP 1, 5, 6 • **Lesson 65: Postmortem** IOC; CTP 6	• What does a professional digital portfolio website look like? • How can I showcase my computer science skills and talents?
Computer Science / Game Design Topics	**Coding Concepts**
• Build and Iterate a Digital Portfolio Website	
Assessments	none
Outcomes	• Develop a digital portfolio website • Publish a digital portfolio website

Instructional Activities

Lesson 64: **Showcasing Your Work** Students organize and plan visual structure of their digital portfolio website. Students then write descriptions of why they invested in designing games, what they enjoyed the most about computer game design and coding, explain their choice of game genres (serious games, education, adventure, puzzle, etc.), and articulate their iterative design process. **CRD, IOC**

Lesson 64: **Building Your Digital Portfolio Website** Students upload and showcase their playable games. Students include a one page design

document that serves as a quick view of their full game design document. **CRD, CSN**

Lesson 64: Portfolio Presentation Students present their digital portfolio unpublished website to their peers for feedback. Students improve their website digital portfolios based on peer feedback, publish it to the Internet, and schedule a formal presentation to local business and / or educational leaders. **IOC**

Lesson 65: Postmortem Students complete a full game design journal entry that deconstructs the game design process, describes their success as a computer game developer, and reflects on the importance of advanced computer programming skills in the game design industry. **IOC**

AP CSP Curriculum Summary Chart

Curriculum		Big Idea					Computational Thinking Practices						Computing Innovation
Mod	**Les**	**CRD**	**DAT**	**AAP**	**CSN**	**IOC**	**1**	**2**	**3**	**4**	**5**	**6**	
1	1	X				X	X					X	
	2	X	X		X	X		X				X	
	3	X			X	X						X	
2	4	X				X	X						
	5	X			X	X					X	X	
	6	X	X		X	X			X				
3	7	X				X	X				X		
	8	X	X			X		X	X				
	9	X	X			X		X	X				
	10	X		X		X		X	X				
	11	X	X		X	X	X			X		X	

4	12	X		X		X	X	X	X	X			
	13	X		X		X	X	X	X	X			
	14	X		X		X	X	X	X	X			
	15	X	X			X	X	X	X	X			
5	16	X		X		X	X	X	X	X			
	17	X		X		X	X	X	X	X			
	18	X	X	X		X		X	X	X			
	19	X	X	X		X		X	X	X			
	20	X				X				X			
6	21	X		X		X	X	X	X	X			
	22	X		X		X	X	X	X	X			
	23	X		X		X	X	X	X	X			
	24	X	X			X	X	X	X	X			
7	25	X		X		X	X	X	X	X			
	26	X		X		X	X	X	X	X			
	27	X	X	X		X	X	X	X	X			
	28	X		X		X		X	X	X			
	29	X		X		X		X	X	X			
	30	X	X			X	X	X	X	X			
8	31	X		X		X	X	X	X	X			
	32	X		X		X		X	X	X			
	33	X	X	X	X	X		X	X	X			
9	34	X		X		X		X	X	X			
	35	X		X		X		X	X	X			
	36	X	X	X		X		X	X	X			

	37	X	X	X		X		X	X	X			
	38	X	X			X	X			X			Explore
10	39	X	X			X	X	X	X	X			
	40	X		X		X		X	X	X			
	41	X		X		X		X	X	X			
	42	X		X		X	X	X	X	X			
	43	X		X		X	X	X	X				
	44	X	X			X							
11	45	X		X		X		X	X	X			
	46	X		X		X		X	X	X			
	47	X		X		X		X	X	X			
	48	X	X			X	X						
12	49	X		X		X	X						
	50	X	X	X		X	X	X	X	X			
	51	X		X		X							
	52	X		X		X	X	X					
	53	X	X			X	X						
13	54	X		X		X		X	X	X			
	55	X	X	X		X	X	X	X	X			
	56	X		X		X	X	X	X	X			
	57	X	X	X		X		X	X	X			
14	58	X		X		X	X	X	X	X			
	59	X		X		X		X	X	X			
	60	X		X		X	X	X	X	X			
	61	X	X	X		X	X	X	X	X	X		

	62	X	X	X		X	X	X	X	X			
15	63	x	x	x		x	x	x	x	x	x	x	Performance Task
16	64	x			X	x	X				X	X	
	65	x				x						X	

Digital Portfolio Website Guide

1. Your digital portfolio should provide working examples of the games you have designed. If you have uploaded your games to a game design site, provide links to the games. Make sure the links are active. While it is fine to provide more than one example of your best work, note which example you consider your finest work. A separate game design rubric is used to assess the game you are showcasing.
2. Include a one page quick view document of the full game design document you developed for your best work example.
3. If your work has been archived somewhere, make sure to include information and a working link to where that archived game is found.
4. Your digital portfolio should include a description about why you are invested in designing games, what you enjoy most about designing games, and what game design focus particularly interests you. For example, are you interested in designing games for education? Serious games? Action games? Your writing should be insightful and describe your depth of interest in game design.
5. Share a game design experience. This should include iterations made after building a prototype for an internship, game jam, or game you've built for a project. Explain the reasoning behind the iterations and how they improved the playability of the game.
6. Document reflections on a game project. This is called the project postmortem. What went well? What were some challenges? How were those challenges overcome? Focus on your design team experience. Your understanding of the game design process, in particular the role of iteration, needs to be clearly articulated.

7. If you created a digital game, make sure you include the games you have designed. Add information on the role you took in creating each game and your assessment of the role you took within a design team.
8. Enjoy creating your digital portfolio. It should reflect your approach to game design. It also should reflect who you are, your unique creative skills, and your passion for game design.

Zulama Computer Game Digital Portfolio Rubric

(included in course teacher resources)

Criteria	Unsatisfactory	Basic	Proficient	Exemplary	Rating
Score	1	2	3	4	
Selection of Artifacts	Artifacts and work samples do not relate to the purpose of the portfolio.	Some artifacts and samples relate to the purpose of the portfolio.	Several artifacts and samples strongly support the purpose of the portfolio.	All artifacts and samples relate to the purpose of the portfolio.	
Descriptions	No explanation is provided for the importance of included artifacts.	Limited explanation is provided for some of the included artifacts.	Clear explanations are given for included artifacts, but with missing or limited insight into process and iterations.	Clear explanations are provided for all artifacts included in the portfolio, including several that give insight into the candidate's process and iteration.	
Layout	Portfolio lacks organization,	Portfolio is fairly well	Portfolio is well	Portfolio is engaging,	

	inconsistent use of fonts, color, theme, and font styles (bold, underline, italics).	**organized but shows some inconsistencies in color, theme, use of fonts, and font styles.**	**organized, easily navigated and consistent in color, theme, use of fonts, and font styles.**	**well organized, and visually pleasing and consistent in color, theme, use of fonts, and font styles.**	
Writing Quality	**Writing shows several spelling, grammar, and syntax errors.**	**Writing shows few spelling, grammar, and syntax errors but is in need of a final review.**	**Writing is mostly error free.**	**Writing shows meticulous attention to detail and indicates close review of grammar, syntax, and font styles.**	

Game Design Document	**Major design details are missing from the Design Document. The scope and intent of the project is unclear.**	**Details are missing from the Design Document, leading to misinterpreta-tion of parts of the plan.**	**The Design Document is clear, unambiguous, and complete.**	**The Design Document includes a student designed summary cover page, is clear, and complete.**	
Aesthetics	**Minimal use of audio and graphics in designed games.**	**Some use of audio and graphics, but both are lacking in creativity.**	**Good use of audio and graphics that add high interest to the game.**	**Exceptional use of audio and graphics, creating a highly interesting and captivating game.**	

Implementation	**The game mechanics do not function properly, hindering the playability of the game.**	**The game mechanics generally function but are not engaging for players.**	**The game mechanics function well and succeed in engaging players at some points in the game.**	**The game mechanics are well-designed and function well, resulting in a highly playable and engaging game.**	
Design Team Collaboration	**Has some difficulty working with others. At times resists the ideas of others. Lacks leadership and resists iteration.**	**Shows respect for new ideas, works effectively with individuals of diverse opinions. Plans and interacts with team in productive ways.**	**Works effectively with individuals embraces brainstorming and iteration. Provides team leadership.**	**Allows others to take the lead, embraces the ideas of others, fulfills team role with extra effort, and embraces the iterative process.**	
Postmortem	**Reflections do not indicate an understanding of the iteration in the game design process.**	**Reflections indicate some understanding of iterative process in game design.**	**Reflections include a good description of the importance of iteration in game design.**	**Reflections indicate a complete description of the worth of the iterative process.**	
Total Score					

TEACHER NOTES