Unit #4: Variables, Conditionals, and Functions

Content Area: **Template**
Course(s):
Time Period:
Length:
Status: **Published**

State Mandated Topics Addressed in this Unit

State Mandated Topics Addressed in this Unit	
N/A	N/A

Unit #4: Variables, Conditionals, and Functions

Learning Objectives

- Apply coding concepts such as variables, broadcasting, and collision detection in their game projects. Demonstrate problem-solving skills by debugging and refining their game mechanics.
- Artists will be able to write code from scratch using variables, conditionals, and functions one at a time and collectively. Artists will work in groups and independently.
- Artists will understand and apply a simplified version of the concept of functions as a way to name a collection of commands so that they can be used in multiple places within your code..
- Artists will understand the concept of conditionals and apply those concepts to coding.
- Artists will understand the concept of variables in the context of programming and apply those concepts.
- Design and build a basic Mario-style platformer game in Scratch.
- Students will be able to create a simple program using block-based coding that demonstrates core programming concepts, user interface design, and problem-solving skills.
- Students will be able to create and design a simple video game using Code.org, incorporating game elements and music elements effectively.
- Students will be able to create and design a simple video game using Pixicade, incorporating game elements and music elements effectively.
- Students will be able to design and code an interactive story using Code.org, demonstrating understanding of coding concepts and storytelling elements.
- Students will be able to design and code an interactive story using Scratch, demonstrating understanding of coding concepts and storytelling elements.
- Students will be able to explain algorithms, syntax, debugging, syntax bugs, logic bugs, variables, names, and values in the context of computer programming.
- Students will be able to understand and apply various coding principles as they complete chapters 2 and 3 of Ozaria in Code Combat.

Essential Skills

- Coding Proficiency: Understanding Scratch programming language and its components. Game Design: Utilizing design principles to create engaging game levels and mechanics. Critical Thinking: Analyzing problems in their code and developing solutions. Collaboration: Working effectively with peers during game testing and feedback sessions.
- Debug programs that use boolean expressions and conditional statements
- Debug programs that use functions
- Debug programs that use variables and expressions
- Describe the way a function call interrupts the normal flow of execution within a program
- Evaluate expressions that include Boolean values, comparison operators, and logical operators
- Evaluate expressions that include numbers, strings, and arithmetic operators.
- Explain the purpose of those programming patterns with boolean expressions and conditional statements both in terms of how they work and what they accomplish
- Explain the purpose of those programming patterns with variables both in terms of how they work and what they accomplish
- Identify common programming patterns using boolean expressions and conditional statements
- Identify common programming patterns when using variables as part of an app
- Identify opportunities to use functions to reduce repeated code within a program
- Implement a function using programming patterns while developing a functional app
- Implement programming patterns with boolean expressions and conditionals statements to develop a functioning app
- Implement programming patterns with variables to develop a functioning app
- Modify apps that make use of common programming patterns with boolean expressions and conditional statements to adjust their functionality
- Modify apps that make use of common programming patterns with variables to adjust their functionality
- Modify programs that declare and call functions to adjust their functionality
- Recognize the need for a function to reduce repeated code while developing a functional app
- Recognize the need for programming patterns with Boolean expressions and conditional statements as part of developing a functioning app
- Recognize the need for programming patterns with variables as part of developing a functioning app
- Scratch Interface: Understanding the layout and features of Scratch, including sprites, backdrops, and code blocks.
- Trace simple programs that use Boolean expressions and conditional statements
- Trace simple programs that use variables, expressions, and variable assignment.
- Trace the flow of execution in programs that declare and call functions
- Use appropriate vocabulary to describe Boolean expressions and conditional statements
- Use appropriate vocabulary to describe the declaring and calling of functions
- Use appropriate vocabulary to describe variables, expressions, and variable assignment.
- Use debugging skills as part of developing an app
- Use debugging skills as part of developing an app
- Use debugging skills as part of developing an app

- Utilizing Variables, Conditionals, Loops, Functions and Debugging:
- Write comments to clearly explain both the purpose and function of different segments of code within an app
- Write comments to clearly explain both the purpose and function of different segments of code within an app
- Write comments to clearly explain both the purpose and function of different segments of code within an app
- Write programs that use boolean expressions and conditional statements with the support of sample code.
- Write programs that use functions with the support of sample code
- Write programs that use variables and expressions with the support of sample code.

Standards

CS.9-12.8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
CS.9-12.8.1.12.AP.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
CS.9-12.8.1.12.AP.3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
CS.9-12.8.1.12.AP.4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
CS.9-12.8.1.12.AP.7	Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
CS.9-12.8.1.12.AP.9	Collaboratively document and present design decisions in the development of complex programs.
CS.9-12.8.1.12.CS.4	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
CS.9-12.8.2.12.NT.2	Redesign an existing product to improve form or function.
CS.9-12.8.2.12.ITH.1	Analyze a product to determine the impact that economic, political, social, and/or cultural factors have had on its design, including its design constraints.

Instructional Tasks/Activities

- Classroom Discussions
- Debugging
- Digital Worksheets
- Exploration/Investigation/App Lab - Design Mode Activities
- Formative Assessments
- Journaling
- Pair Programming
- Peer Feedback
- Project Decision Maker App

- Students will complete a coding challenge that requires them to create a simple program using the concepts learned (algorithms, variables, etc.) and identify and correct any syntax or logic bugs in provided code snippets.
- Students will present their interactive stories to the class, showcasing their use of Scratch features (such as sprites, sounds, and programming blocks) and explaining their coding choices. They will be evaluated based on a rubric that assesses creativity, functionality, and engagement.
- Students will work individually on their projects in code.com, ensuring to follow the minimum program requirements. They will document their coding process and any challenges faced.

Assessment Procedure

- Classroom Total Participation Technique
- Classwork
- DBQ
- electronic active responders
- Essay
- Exit Ticket/Entrance Ticket/Do Now
- identify the error problems
- Journal / Student Reflection
- Kahoot
- Other named in lesson
- Peer Review
- Performance
- Problem Correction
- Progress in Ozaria/Code Combat
- Project
- Quiz
- response and analysis questions
- Rubric
- Teacher Collected Data
- Test
- Worksheet

Recommended Technology Activities

- App Lab
- Appropriate Content Specific Online Resource
- Code.org
- Decision Maker App
- Gimkit
- GoGuardian

- Google Classroom
- Google Docs
- Google Slides
- Google Slides
- Kahoot
- MagicSchool AI
- Other- Specified in Lesson
- Ozaria/Code Combat
- Photo Liker App
- Quiziz
- Scratch Program
- Screencastify

Accommodations & Modifications & Differentiation

Accommodations and Modifications should be used to meet individual needs. Their IEP and 504 plans should be used in addition to the following suggestions.

Gifted and Talented

- Compare & Contrast
- Conferencing
- Debates
- Jigsaw
- Peer Partner Learning
- Problem Solving
- Structured Controversy
- Think, Pair, Share
- Tutorial Groups

Instruction/Materials

- alter format of materials (type/highlight, etc.)
- color code materials
- eliminate answers
- extended time
- extended time
- large print

- modified quiz
- modified test
- Modify Assignments as Needed
- Modify/Repeat/Model directions
- necessary assignments only
- Other (specify in plans)
- other- named in lesson
- provide assistance and cues for transitions
- provide daily assignment list
- read class materials orally
- reduce work load
- shorten assignments
- study guide/outline
- utilize multi-sensory modes to reinforce instruction

Environment

- alter physical room environment
- assign peer tutors/work buddies/note takers
- assign preferential seating
- individualized instruction/small group
- modify student schedule (Describe)
- other- please specify in plans
- provide desktop list/formula

Honors Modifications

Resources

- code.org
- https://codecombat.com/
- https://csfirst.withgoogle.com/s/en/home
- Internet Simulator