Coding in Minecraft and Computer Science Discoveries Integration

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 1: Programming for a Purpose	Students will be able to: • Create a prototype of an animation or game design to meet the needs of a user using the problem-solving process • Identify features of an animation or game design that match the needs of users • Understand the steps of the problem-solving process	CSTA K-12 Computer Science Standards AP - Algorithms & Programming 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	1
Computer Science Discoveries: Lesson 2: Plotting Shapes	Students will be able to: • Communicate how to draw an image in Game Lab, accounting for shape position, color, and order • Reason about locations on the Game Lab coordinate grid	CSTA K-12 Computer Science Standards (2017) 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 3: Drawing in Game Lab	Students will be able to: • Sequence code correctly to overlay shapes. • Use a coordinate system to place elements on the screen.	CSTA K-12 Computer Science Standards (2017) 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1
Computer Science Discoveries: Lesson 4: Shapes and Parameters	Students will be able to: • Use and reason about drawing commands with multiple parameters	CSTA K-12 Computer Science Standards (2017) 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1
Computer Science Discoveries: Lesson 5: Variables	Students will be able to: • Identify a variable as a way to label and reference a value in a program	CSTA K-12 Computer Science Standards (2017)	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
	• Use variables in a program to store a piece of information that is used multiple times	2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	
Coding in Minecraft – Intermediate Coding using MakeCode: Review Lesson (Lesson 1)	Students will be able to: • Navigate the Minecraft environment • Access the MakeCode environment • Understand the basics of MakeCode blocks • Understand the role of the Agent	CSTA K-12 Computer Science Standards 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-09 Create programs that use variables to store and modify data.	2
Coding in Minecraft – Intermediate Coding using MakeCode: Lesson 2: Variables	Students will be able to: • Learners will be able to effectively use standard programming blocks in MakeCode to create variables	CSTA K-12 Computer Science Standards 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.	2

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
	• Learners will be able to design and develop a programmatic solution for a given problem	2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	
Computer Science Discoveries: Lesson 6: Random Numbers	Students will be able to: • Generate and use random numbers in a program • Update a value stored in a variable	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1
Computer Science Discoveries: Lesson 8: Sprites	Students will be able to: • Create and use a sprite	CSTA K-12 Computer Science Standards (2017)	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	
Computer Science Discoveries: Lesson 9: Sprite Properties	Students will be able to: • Use dot notation to update a sprite's properties	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	
Computer Science Discoveries: Lesson 10: Text	Students will be able to: • Place text on the screen using a coordinate plane. • Use arguments to control how text is displayed on a screen.	CSTA K-12 Computer Science Standards (2017) 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 11: Mini-Project - Captioned Scenes	Students will be able to: • Use a structured process to plan and develop a program.	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	2
Computer Science Discoveries: Lesson 12: The Draw Loop	Students will be able to: • Explain how the draw loop allows for the creation of animations in Game Lab • Use the draw loop in combination with the randomNumber() command, shapes, and sprites to make simple animations	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures,	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	
Computer Science Discoveries: Lesson 13: Sprite Movement	Students will be able to: • Identify which sprite properties need to be changed, and in what way, to achieve a specific movement • Use the counter pattern to increment or decrement sprite properties	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures,	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	
Computer Science Discoveries: Lesson 14: Mini-Project - Animation	In this lesson, students are asked to combine different methods that they have learned to create an animated scene.	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	2

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	
Computer Science Discoveries: Lesson 15: Conditionals	Students will be able to: • Use conditionals to react to changes in variables and sprite properties	CSTA K-12 Computer Science Standards (2017) 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Coding in Minecraft – Intermediate Coding using MakeCode: Lesson 3: Logic	Students will be able to: • Identify standard programming blocks in MakeCode such as loops, variables and logic • Understand the outcome of a program without running the code • Effectively use standard programming blocks in MakeCode such as loops, variables and logic • Design and develop a programmatic solution for a given problem	CSTA K-12 Computer Science Standards 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	2
Coding in Minecraft – Intermediate Coding using MakeCode: Lesson 4: Advanced Logic	Students will be able to: • Identify standard programming blocks in MakeCode such as loops, variables and logic • Understand the outcome of a program without running the code • Effectively use standard programming blocks in MakeCode such as loops, variables and logic • Design and develop a programmatic solution for a given problem	CSTA K-12 Computer Science Standards 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	2

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	
Computer Science Discoveries: Lesson 16: Keyboard Input	Students will be able to: • Move sprites in response to keyboard input • Use conditionals to react to keyboard input	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 17: Mouse Input	Students will be able to: • Respond to a variety of types of user input. • Use an if-else statement to control the flow of a program.	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Coding in Minecraft – Intermediate Coding using MakeCode: Lesson 5 – Loops	Students will be able to: • Identify standard programming blocks in MakeCode such as loops, variables and logic • Understand the outcome of a program without running the code • Effectively use standard programming blocks in MakeCode such as loops, variables and logic Design and develop a programmatic solution for a given problem	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. CSTA K-12 Computer Science Standards 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 Document programs in order to make them easier to follow, test, and debug.	2

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Coding in Minecraft – Intermediate Coding using MakeCode: Lesson 6 – Advanced Loops	Students will be able to: • Identify standard programming blocks in MakeCode such as loops, variables, logic and functions • Understand the outcome of a program without running the code • Demonstrate the ability to determine the differences in the output of a program, after making changes to the program, given only the program and its input without running the program • Identify errors in code • Effectively use standard programming blocks in MakeCode such as loops, variables and logic	CSTA K-12 Computer Science Standards 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 Document programs in order to make them easier to follow, test, and debug.	2
Coding in Minecraft – Intermediate Coding using MakeCode: Lesson 7: The Project	Students will design and develop a solution to a problem while demonstrating their coding abilities using variables, logic, and loops.	CSTA K-12 Computer Science Standards	2 – 10 (depends on the scope of the project the

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	teacher chooses)
Computer Science Discoveries: Lesson 18: Project - Interactive Card	Students will be able to: • Apply an iterator pattern to variables or properties in a loop • Sequence commands to draw in the proper order • Use conditionals to react to keyboard input or changes in variables / properties • *(Students could tie this card to their Coding in Minecraft project)*	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	2 – 3

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 19: Velocity	Students will be able to: • Describe the advantages of simplifying code by using higher level blocks • Use the velocity and rotationSpeed blocks to create and change sprite movements	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 20: Collision Detection	Students will be able to: • Describe how abstractions help to manage the complexity of code • Detect when sprites are touching or overlapping, and change the program in response.	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 21: Mini-Project - Side Scroller	Students will be able to: • Use what they have learned about collision detection and setting velocity to create a simple side scroller game.	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases.	1
Computer Science Discoveries: Lesson 22: Complex Sprite Movement	Students will be able to: • Explain how individual programming constructs can be combined to create more complex behavior • Use sprite velocity with the counter pattern to create different types of sprite movement	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values.	1-2

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 23: Collisions	Students will be able to: • Describe how abstractions can be built upon to develop even further abstractions • Model different types of interactions between sprites.	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 24: Mini-Project - Flyer Game	Students will be able to: • Use what they have learned about simulating gravity and the different types of collisions to create simple flyer games.	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1-2
Computer Science Discoveries: Lesson 25: Functions	Students will be able to: • Create and use functions for blocks of code that perform a single high-level task within a program • Explain how functions allow programmers to reason about a program at a higher level	CSTA K-12 Computer Science Standards (2017)	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
	• Explain the advantages of using functions in a program	2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 26: The Game Design Process	Students will be able to: • Implement different features of a program by following a structured project guide	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 27: Using the Game Design Process	Students will be able to: • Implement different features of a program by following a structured project guide • Identify core programming constructs necessary to build different components of a game	CSTA K-12 Computer Science Standards (2017) 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Computer Science Discoveries: Lesson 28: Project - Design a Game	Students will be able to: • Create a plan for building a piece of software by describing its major components • Implement a plan for creating a piece of software • Independently scope the features of a piece of software	CSTA K-12 Computer Science Standards (2017) 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution.	3-5

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	
Coding in Minecraft – Advanced Coding using JavaScript Lesson 2: Algorithms	Students will be able to: • Understand that code is reusable • Demonstrate the ability to determine the output of a program given only the program and its input without running the program • Identify syntax errors in existing code and provide explanations of the syntax error and suitable resolution(s) • Understand what an algorithm is. • Write and understand basic algorithms.	CSTA K-12 Computer Science Standards 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse. 2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-19 Document programs in order to make them easier to follow, test, and debug. 2-AP-17 Systematically test and refine programs using a range of test cases.	1

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
Coding in Minecraft – Advanced Coding using JavaScript Lesson 3: Moving to JavaScript	Students will be able to: • Identify the equivalent JavaScript code of MakeCode blocks • Understand the difference between block-based and text-based programming languages. • Make changes to JavaScript code.	CSTA K-12 Computer Science Standards 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	2
Coding in Minecraft – Advanced Coding using JavaScript Lesson 4: JavaScript Loops	Students will be able to: • Identify iteration and selection in both MakeCode blocks and JavaScript • Identify and modify repeat, while and for loops in JavaScript • Understand loops in JavaScript • Identify equivalent JavaScript looping code for MakeCode blocks	CSTA K-12 Computer Science Standards 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	2
Coding in Minecraft – Advanced Coding using JavaScript Lesson 5: JavaScript Logic	Students will be able to: • Identify iteration and selection in both MakeCode blocks and JavaScript • Understand why logic & selection is required and when you might use them. • Identify and modify IF THEN and IF THEN ELSE JavaScript statements in MakeCode.	CSTA K-12 Computer Science Standards 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	2

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
	• Use comparison and Boolean operators in JavaScript		
Coding in Minecraft – Advanced Coding using JavaScript Lesson 6: Errors	Students will be able to: • Demonstrate the ability to determine the output of a program given only the program and its input without running the program • Identify syntax errors in existing code and provide explanations of the syntax error and suitable resolution(s)	CSTA K-12 Computer Science Standards 2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-19 Document programs in order to make them easier to follow, test, and debug. 2-AP-17 Systematically test and refine programs using a range of test cases.	2
Coding in Minecraft – Advanced Coding using JavaScript Lesson 7: Project	Students will design and develop a solution to a problem while demonstrating their coding abilities using variables, logic, and loops.	CSTA K-12 Computer Science Standards 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 Decompose problems and subproblems into parts to facilitate the	3 – 10 (depends on the scope of the project the teacher chooses)

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CS Standards	Average Completion Time (in hours)
		design, implementation, and review of programs.	

© Prodigy Learning | prodigylearning.com | codingcredentials.com