



Computer Science

37 Lessons

Lower Secondary | Grades 6-8 | Ages 11-14

Coding with Minecraft

[EDUCATION.MINECRAFT.NET](https://education.minecraft.net)

Table of Contents

<u>GETTING STARTED WITH MINECRAFT</u>	<u>3</u>
<u>COMPUTER SCIENCE PATHWAY PROGRESSION</u>	<u>4</u>
<u>CURRICULUM SUMMARY</u>	<u>5</u>
<u>INTRODUCTION</u>	<u>7</u>
<u>MINECRAFT: EDUCATION EDITION- TEACHING & LEARNING FRAMEWORK</u>	<u>8</u>
<u>INSTRUCTIONAL SEQUENCE</u>	<u>9</u>
<u>EDUCATIONAL STANDARDS</u>	<u>15</u>
<u>COMPUTER SCIENCE CONCEPTS</u>	<u>17</u>
<u>VISUAL GLOSSARY</u>	<u>18</u>

Getting Started with Minecraft

Install Minecraft: Education Edition

Minecraft: Education Edition can be installed on Chromebook, iPad, Mac, and PC. To ensure your experience with Minecraft: Education Edition is top-notch, make sure your devices meet the minimum system requirements. To check if your device supports Minecraft: Education Edition, see [System Requirements](#).

If you have questions about setting up Minecraft: Education Edition, the following link will provide you with some [frequently asked questions and additional information](#) about set-up. On this page, you will find assistance for:

- Get Started
- Purchase Licenses
- Administration and License Management
- Installation
- Troubleshooting

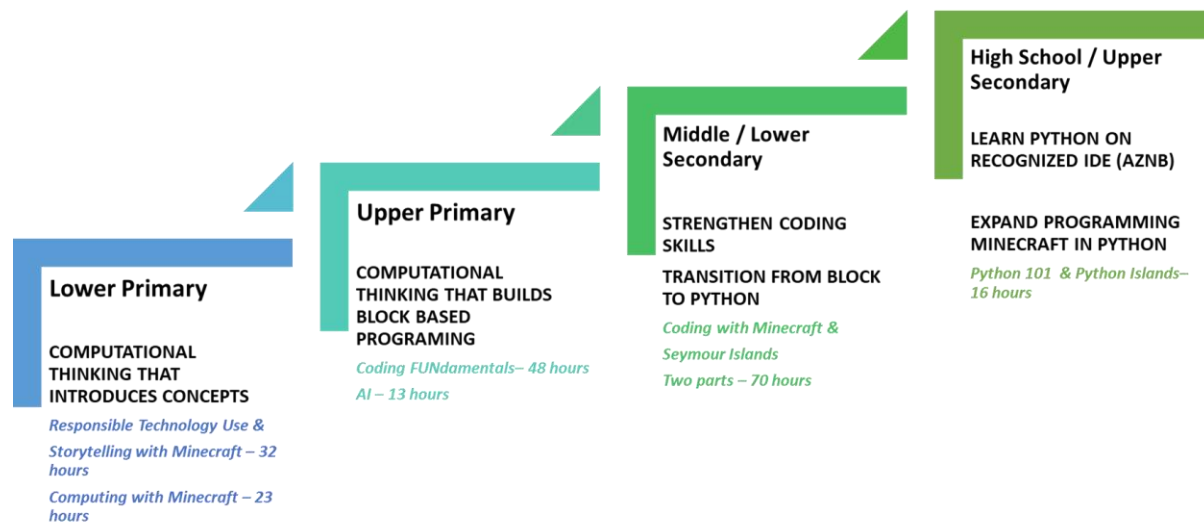
Preparing to Teach with Minecraft: Education Edition

Teachers do not need to have any prior computer science experience; however, they should familiarize themselves with a basic understanding of what is Minecraft: Education Edition. Support for building out teacher knowledge can be found here:

[Minecraft: Education Edition: Teacher Academy](#)

[Minecraft: Education Edition Webinar Series](#)

Computer Science Pathway Progression



Coding with Minecraft is a part of the Lower Secondary (Grades 6-8 | Ages 11-14) computer science progression. In this part of the progression, students are strengthening their core computer programming concepts, such as events, coordinates, variables, conditionals, functions, iteration, arrays, and artificial intelligence.

Students will have multiple authentic opportunities to learn, practice, and apply these coding concepts within Minecraft: Education Edition. Students will use a Blocks of Grass world to start out with, as a “sandbox” for their code, and as they learn and experiment, they will be able to see the progress of their learning visually in their world as a series of different physical projects and constructions.

Each of the curriculum includes an independent project where students will have the opportunity to create and demonstrate their new content knowledge and skills in a meaningful way.

Curriculum Summary

Coding with Minecraft is specially designed computer science content for students in Lower Secondary/Grades 6-8/Ages 11-14. Students will learn, practice, and apply relevant computer science skills and concepts as well as literacy skills in both unplugged and digital experiences. The lessons are designed to provide students with an opportunity to build knowledge of the concept(s) in an unplugged version (i.e., demonstrate the concept on paper), practice the concept with the direct support of their teacher in the Minecraft world, and then finally by completing the task independently.

Lesson Design

Each EDU guide contains multiple activities that are intended to be taught over the specified amount of sessions (explained in the **Instructional Sequence** part of this document); however, you should use discretion and modify/adapt the lesson activities based on your students' needs and abilities. Within the lessons, the instructional sequence will contain three parts correlating with the gradual release model:

Direct Instruction—Teacher-Directed, "I Do"	In the first step, the teacher introduces and models the appropriate way of performing the skills included in the new concept being taught.
Guided Instruction— Teacher Modeling, "We Do"	After the teacher models the correct way to understand or perform the new concept being taught, teacher will guide the students as they work through some examples together.
Independent Practice—Teacher Support, "You Do"	This step is where students demonstrate their initial level of understanding of the new concept being taught through independent practice.

Instructional Materials

Curriculum Overview	That is this document you are reading now! This will provide you with insight about the curriculum and what is taught within the curriculum.
Educator's Guides (EDU Guides)	An educator's guide is provided for each of the lessons. The guide provides a high-level overview of the lesson, learning goals, standards addressed, required preparation for the activities, the lesson plans for the activities, and any additional materials needed.
Classroom Presentations	Each unit is supported by its own PowerPoint presentation to provide structure and guide the educator through the activities for the lesson.

Assessment Guide	<p>To communicate expectations and support the educator in evaluating student work and final projects, assessment materials are provided for the course, as well as each unit and its associated lessons:</p> <ul style="list-style-type: none"> • Formative and summative assessment answer keys and printer-friendly versions to distribute to your students • Project scoring rubrics
Student Workbooks	<p>Students are provided with a resource workbook to guide their learning throughout each of the curriculum units.</p>
Minecraft World Files	<p>The specific world files needed to experience the instructional activities have been linked directly within the Educator's Guides. All of the worlds can also be accessed through the in-game library.</p>

Introduction

Level: Lower Secondary | Grades 6-8 | Ages 11-14

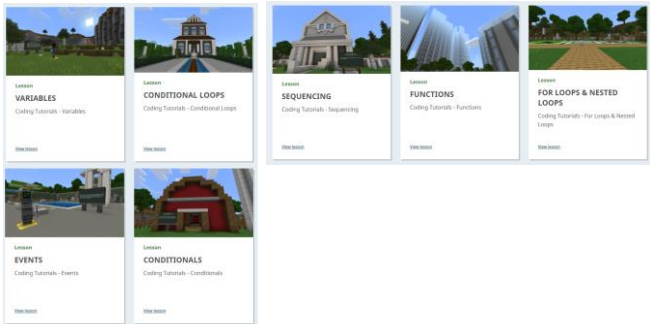
Essential Question: How can computational thinking help us to solve problems and identify solutions?

Introduction

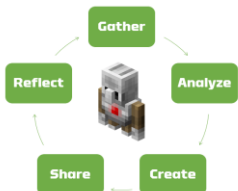
In this unit, students will explore the coding concepts of events and event handlers, coordinates, variables, conditionals, functions, iterations, arrays, and artificial intelligence. Students will get familiar with the concept through a kinesthetic unplugged activity. Then they get hands-on experience with guided coding activities to practice using MakeCode, complete various tasks and challenges to practice their coding skills, and opportunities to evaluate and improve existing code. In the final activity for each of the mini-units, students will use design thinking and programming techniques to creatively solve problems.

Pacing: 58 hours (could be more if tutorials and/or supplemental lessons are included)

Materials

Hardware	<ul style="list-style-type: none">The teacher will need a laptop or tablet with a projector for the plugged-in coding lessons.Internet access will be required for a portion of the lessons and activities.Each student will need a device to complete the plugged-in coding activities.
Software	<ul style="list-style-type: none">Minecraft: Education Edition needs to be deployed on the devices utilized within these lessons. Use this link to find information about Deploying Minecraft: Education Edition.The teacher and students will need the relevant MCworld files downloaded for each lesson.
Other Materials	<ul style="list-style-type: none">M:EE Tutorials in Blocks (if needed) M:EE Supplemental Lessons (if needed)

Minecraft: Education Edition Teaching and Learning Framework

GOAL: Provide students with the necessary skills to address the complex societal issues of our time and their future	
<p>Coding Mindset</p> <p>Although Minecraft: Education Edition provides rigorous and engaging academics, we also deliver a holistic education- providing instruction for the social, emotional, and physical needs for our students. We want to foster a distinctive set of attributes. These qualities prepare our students to make exceptional contributions both in school and outside of school.</p> <p>CURIOUS COMMITTED COOPERATIVE CONSIDERATE CONFIDENT</p>	<p>Computational Thinking Skills</p> <p>Computational thinking provides a vital skill set in which students must possess in order to fulfill the industry's needs in the jobs of tomorrow. Our ever-changing workforce creates a critical need for innovation. Our students need computational thinking skills not just to solve the problems within their educational journey, but to also meet the challenges of adapting to our constantly changing workforce.</p> <p>DECOMPOSITION PATTERN RECOGNITION ABSTRACTION ALGORITHMS</p>
<p>Computer Science Units of Study</p> <p>Minecraft: Education Edition provides meaningful, relevant, and engaging units of study. The units of study will possess a conceptual lens to allow for depth and complexity to develop conceptual understanding—knowledge which transfers through time, across cultures, and across situations.</p> <p>DIGITAL CITIZENSHIP PROGRAMMING CYBERSECURITY IMPACTS OF COMPUTING</p>	<p>Community</p> <p>As students gain and possess new knowledge and skills, we strive for them to find a greater purpose of “why do I need to know this” or more importantly, “how can I use this information?”.</p> <p>We aim to empower students develop confidence and self-efficacy into a commitment to serve the community in which we live in and beyond.</p>  <pre> graph TD Gather --> Analyze Analyze --> Create Create --> Share Share --> Reflect Reflect --> Gather </pre>

This unit will focus on **Programming**, as students will investigate the coding concepts in block-based programming via MakeCode.

This unit will focus on the coding mindset of being **confident**, as they hone in on specific coding skills to build out their programming knowledge and expertise.

This unit will focus on **decomposition** as students will need to think methodically on how to break down task into smaller sub-tasks to design, create, test, and debug their programs.

This unit will provide students with the opportunity to consider how they can utilize their new knowledge of programming to bring awareness to the **community**. Students can explore how coding and programming is presently being used in their school, community, and daily lives to consider what problems and/or tasks could be innovated through coding.

Instructional Sequence

This next section will provide you with an overview of the activities included in this lesson sequence. The lesson sequence is presented in chronological order—we suggest working in order, as the content will build upon skills presented in the previous session. A session is equivalent to one class period, or a 45-60 minute session. However, educators should feel empowered to modify and adapt the lesson sequence to best meet the needs of their students.

Lesson Sequence Overview

Session	Objectives	Teacher will	Students will	Resources
1*	Students will be able to explain the learning goals and concept of computer science, Minecraft, and Microsoft MakeCode.	Introduce Minecraft and demonstrate how to launch the game; Introduce MakeCode and how it will be the coding platform for the lesson.	Students will demonstrate how to use the Minecraft blocks in Microsoft MakeCode by navigating through Minecraft game play and the Chicken Rain tutorial.	Coding with Minecraft: Introduction
2*	Students will learn how to effectively use events and event handlers in their programs.	Introduce the concept of events and event handlers and how to use events and event handlers in Minecraft	Students will practice and use events and event handlers in the Yellow Brick Road coding activity.	Coding with Minecraft: Events- Lesson A: Introduction to Events
3*	Students will learn how to effectively use events and event handlers in their programs.	Practice the concept of events and event handlers in Minecraft	Students will write programs using events and event handlers in the Sing A Song of Sixpence and Last Stand at the Alamo coding activities.	Coding with Minecraft: Events- Lesson B: Coding with Events
4*	Students will learn how to effectively use events and event handlers in their programs.	Practice the concept of events and event handlers in Minecraft	Students will write programs using events and event handlers in the Linked Wall coding activity.	Coding with Minecraft: Events- Lesson C: Linking Events
5-6*	Students will demonstrate how to effectively use events and event handlers in their programs.	Apply the concept of events and event handlers in Minecraft	Students will write programs using events and event handlers in their Independent Coding project and Minecraft Diary entry.	Coding with Minecraft: Events- Lesson D: Get Creative with Events

7*	Students will learn how to effectively use coordinates in their programs.	Introduce the concept of coordinates in Minecraft and lead the World Landmarks unplugged activity and the Create a Compass coding activity	Students will practice utilizing programs with coordinates in the Create a Compass coding activity.	Coding with Minecraft: Coordinates- Lesson A: Introduction to Coordinates
8*	Students will learn how to effectively use coordinates in their programs.	Review the concepts of coordinates in Minecraft and difference between relative coordinates and absolute coordinates	Students will practice utilizing programs with coordinates in the Minecraft Moving Company coding activity.	Coding with Minecraft: Coordinates- Lesson B: Coding with Coordinates
9*	Students will learn how to effectively use coordinates in their programs.	Review the concepts of coordinates in Minecraft	Students will practice utilizing programs with coordinates in the Auto-Farmer coding activity.	Coding with Minecraft: Coordinates- Lesson C: Code with Coordinates and Moving Objects
10-11*	Students will demonstrate how to effectively use coordinates in their programs.	Introduce and facilitate the independent coding project in Minecraft	Students will write programs using coordinates in their Independent Coding project and Minecraft Diary entry.	Coding with Minecraft: Coordinates- Lesson D: Get Creative with Coordinates
12*	Students will learn how to effectively use variables in their programs.	Introduce the concept of variables in Minecraft and lead the Slap, Clap, Slap unplugged activity and the Chicken Storm coding activity	Students will practice utilizing programs with variables in the Chicken Storm coding activity.	Coding with Minecraft: Variables- Lesson A: Introduction to Variables
13*	Students will learn how to effectively use variables in their programs.	Review the concepts of variables (number, string, Boolean, position) in Minecraft	Students will practice utilizing programs with variables in the Arrow Counter and Fall is in the Air coding activities.	Coding with Minecraft: Variables- Lesson B: Coding with Variables
14*	Students will learn how to effectively use variables in their programs.	Review the concepts of variables (number, string, Boolean, position) in Minecraft	Students will practice utilizing programs with variables in the Wordsmith coding activity.	Coding with Minecraft: Variables- Lesson C: Combining Variables

15-16*	Students will demonstrate how to effectively use variables in their programs.	Introduce and facilitate the independent coding project in Minecraft	Students will write programs using variables in their Independent Coding project and Minecraft Diary entry.	Coding with Minecraft: Variables- Lesson D: Get Creative with Variables
17*	Students will learn how to effectively use conditionals in their programs.	Introduce the concept of conditionals in Minecraft and lead the Simon Says If Then Else unplugged activity and the How Old Are You? coding activity	Students will practice utilizing programs with conditionals in the How Old Are You? coding activity.	Coding with Minecraft: Conditionals- Lesson A: Introduction to Conditionals
18*	Students will learn how to effectively use conditionals in their programs.	Review the concept and purpose of conditionals in Minecraft	Students will practice utilizing programs with conditionals in the Agent Tree Chopper and All Mine! coding activities.	Coding with Minecraft: Conditionals- Lesson B: Coding with Conditionals
19-20*	Students will demonstrate how to effectively use conditionals in their programs.	Introduce and facilitate the independent coding project in Minecraft	Students will write programs using conditionals in their Independent Coding project and Minecraft Diary entry.	Coding with Minecraft: Conditionals- Lesson C: Get Creative with Conditionals
21*	Students will learn how to effectively use functions in their programs.	Introduce the concept of functions and how to use functions in Minecraft	Students will practice and use functions in the Leap of Faith mini-game coding activity.	Coding with Minecraft: Functions- Lesson A: Introduction to Functions
22*	Students will learn how to effectively use functions in their programs.	Review the concept of functions in Minecraft	Students will write programs using functions in the Zombie Pig coding activity.	Coding with Minecraft: Functions- Lesson B: Coding with Functions
23*	Students will learn how to effectively use functions in their programs.	Review the concept of functions in Minecraft	Students will write programs using functions in the Lead the Burger coding activity.	Coding with Minecraft: Functions- Lesson C: Building on Functions
24-25*	Students will demonstrate how to effectively use functions in their programs.	Review the concept of functions in Minecraft	Students will write programs using functions in their Independent Coding project and Minecraft Diary entry.	Coding with Minecraft: Functions- Lesson D: Get Creative with Functions

26*	Students will learn how to effectively use iteration in their programs.	Introduce the concept of iteration in Minecraft and lead the Walk Around the House unplugged activity and the Introduction to the Agent and the Dance Dance Agent coding activities.	Students will practice utilizing programs with iteration (with loops) in the Introduction to the Agent and Dance Dance Agent coding activities.	Coding with Minecraft: Iteration- Lesson A: Introduction to Iteration
27*	Students will learn how to effectively use iteration in their programs.	Review the concepts of iteration (using loops) in Minecraft	Students will practice utilizing programs with iteration in the One Block at a Time coding activity.	Coding with Minecraft: Iteration- Lesson B: Coding with Iteration
28*	Students will learn how to effectively use iteration in their programs.	Review the concepts of iteration (with loops) in Minecraft	Students will practice utilizing programs with iterations in the Help Your Agent Farm coding activity.	Coding with Minecraft: Iteration- Lesson C: Debugging with Iteration
29-30*	Students will demonstrate how to effectively use iteration in their programs.	Introduce and facilitate the independent coding project in Minecraft	Students will write programs using iteration in their Independent Coding project and Minecraft Diary entry.	Coding with Minecraft: Iteration- Lesson D: Get Creative with Iteration
31*	Students will learn how to effectively use arrays in their programs.	Introduce the concept of arrays in Minecraft and the purpose of using arrays in programs.	Students will practice the concept of arrays by completing pseudocode for the Bubble Sort activity.	Coding with Minecraft: Arrays- Lesson A: Introduction to Arrays
32*	Students will learn how to effectively use arrays in their programs.	Review the concepts of arrays in Minecraft and associated terms: array length, sort, index, type	Students will practice utilizing programs with arrays in the We Built a Zoo coding activity.	Coding with Minecraft: Arrays- Lesson B: Build A Zoo with Arrays
33*	Students will learn how to effectively use arrays in their programs.	Review the concepts of arrays in Minecraft	Students will practice utilizing programs with arrays in the Warp Belt coding activity.	Coding with Minecraft: Arrays- Lesson C: Teleport with Arrays
34*	Students will learn how to effectively use arrays in their programs.	Review the concepts of arrays in Minecraft	Students will practice utilizing programs with arrays in the Blocks by ID coding activity.	Coding with Minecraft: Arrays- Lesson D: Blocks with Arrays
35-36*	Students will demonstrate how to effectively use arrays in their programs.	Introduce and facilitate the independent coding project in Minecraft	Students will write programs using arrays in their Independent	Coding with Minecraft: Arrays- Lesson D: Get

			Coding project and Minecraft Diary entry.	Creative with Arrays
37*	Students will learn the importance of artificial intelligence.	Introduce the concept of and purpose of artificial intelligence	Students will explore the concept of AI through the Paper AI unplugged activity.	Coding with Minecraft: AI- Lesson A: Introduction to Artificial Intelligence
38*	Students will learn how to effectively use artificial intelligence via the intelligent Agent in their programs.	Review the concept and purpose of artificial intelligence in Minecraft	Students will practice utilizing programs with AI in the Maze Generation coding activity.	Coding with Minecraft: AI- Lesson B: Explore Intelligent Code
39*	Students will learn how to effectively use artificial intelligence via the intelligent Agent in their programs.	Review the concept and purpose of artificial intelligence in Minecraft	Students will practice utilizing programs with AI in the Maze Pathfinding coding activity.	Coding with Minecraft: AI- Lesson C: Code an Intelligent Agent
40*	Students will learn how to effectively use artificial intelligence via the intelligent Agent in their programs.	Review the concept and purpose of artificial intelligence in Minecraft	Students will practice utilizing programs with AI in the Tree Hunter coding activity.	Coding with Minecraft: AI- Lesson D: Code a Tree Hunting Agent
41*	Students will demonstrate how to effectively use artificial intelligence in their programs.	Introduce and facilitate the independent coding project in Minecraft	Students will write programs using AI in their Independent Coding project and Minecraft Diary entry.	Coding with Minecraft: AI- Lesson E: Get Creative with Artificial Intelligence
42*	Students will learn about the final project for the course.	Introduce the final project and lead the class in a review of the course (Part 1 & Part 2 of Coding with Minecraft)	Students will work in groups to review concepts and begin project planning.	Coding with Minecraft: Final Project- Lesson A: Course Review
43-57**	Students will collaborate with peers to create a computational artifact.	Review the final project task and provide guidance to collaborative groups as needed.	Students will work together to plan, design, test, and refine their solution.	Coding with Minecraft: Final Project- Lesson B: Coding a Great Journey
58*	Students will present their final projects in their final showcase.	Review the final deliverables and lead students through the	Students will share their final projects in the showcase.	Coding with Minecraft: Final

finalization of their
project

[Project- Lesson C:](#)
[Course Wrap-Up](#)

**Teachers should use discretion and modify the lessons as needed for students based on needs.*

***The final project may last more than 15 days—use discretion and modify the time as needed.*

Educational Standards

CSTA Standards

- **2-DA-08** Collect data using computational tools and transform the data to make it more useful and reliable.
- **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-14** Create procedures with parameters to organize code and make it easier to reuse.
- **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.

ISTE Standards

- **1.2.b** Students engage in positive, safe, legal and ethical behavior when using technology, including social interactions online or when using networked devices.
- **1.3.c** Students curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions.
- **1.4.c** Students develop, test and refine prototypes as part of a cyclical design process.
- **1.5.a** Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.
- **1.5.c** Students break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
- **1.6.b** Students create original works or responsibly repurpose or remix digital resources into new creations.
- **1.7.c** Students contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.

UK National Curriculum: Computing - Key Stage 3

- Develop their capability, creativity and knowledge in computer science, digital media and information technology
- develop and apply their analytic, problem-solving, design, and computational thinking skills
- understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to identify and report a range of concerns

Australian F-10 Curriculum: Digital Technologies - Year 7 and 8

- Developing a preliminary specification for an opportunity or a need that typically contains a problem statement, a set of solution needs expressed as functional and non-functional requirements, any assumptions or constraints to be considered and the scope or boundaries of the solution (ACTDIP036)
- Designing algorithms to solve real-world problems and describing algorithms using flow charts and structured English (ACTDIP040)
- Recognising that different algorithms can solve a problem with different trade-offs (ACTDIP040)
- Tracing algorithms to predict results and program state for a given input (ACTDIP040)
- Coding separate modules that perform discrete functions but collectively meet the needs of the solution (ACTDIP041)
- Considering different algorithms and selecting the most appropriate based on the type of problem (ACTDIP041)
- Selecting different types of data structures such as an array, record and object to model structured data (ACTDIP041)
- Creating online interactive solutions for working with others by combining or modifying online software tools to support project work (ACTDIP043)
- Managing and modifying the development of a solution (ACTDIP044)
- Developing an evolutionary prototype iteratively and incrementally (ACTDIP044)

Computer Science Concepts: Coding with Minecraft

Lesson	Concept(s)
Unit 1 - Introduction	Block-based coding Sequencing Decomposition
Unit 2 – Events and Event Handlers	Events Event Handlers
Unit 3 - Coordinates	Coordinates Relative Position Absolute World Position Positioning programming
Unit 4 - Variables	Variables
Unit 5 - Conditionals	if Conditionals if else Conditionals elif Conditionals Boolean logic
Unit 6 - Functions	Functions
Unit 7 - Iteration	Repeat Loops For Loops While Loops Forever Loops
Unit 8 - Arrays	Arrays Lists
Unit 9 – Artificial Intelligence	Machine Learning Agent commands
Unit 10 – Final Project	Decomposition Collaboration Pair Programming

MINECRAFT VISUAL GLOSSARY

Agent

personal robot in Minecraft



Blocks

the basic units of structure in
Minecraft that make up the game's
world



Book & Quill

an item used to create written books
in Minecraft



Camera

allows you to take screenshots and
selfies in Minecraft: Education Edition



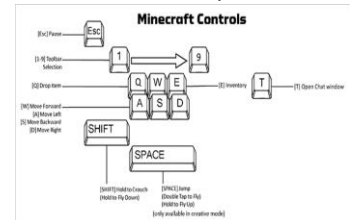
Chalkboards

special blocks that allow you to write
and display text in Minecraft



Controls

(keyboard)
keyboard buttons that help you move
around and complete tasks



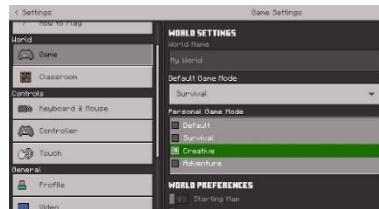
Controls

(touch)
the touch pad that helps you move
around and complete tasks



Creative

game mode that gives you unlimited resources, ability to fly, and lets you destroy blocks instantly when mining



Hotbar

selection bar that appears on the bottom of the screen



Inventory

pop-up menu the player (or Agent) uses to manage the items they carry



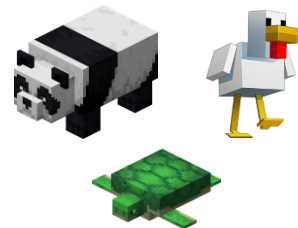
Minecraft: Education Edition

a game-based learning platform



Mob

game character resembling a living creature



Portfolio

saves all of the photos that you have taken with a camera; allows you to add captions



NPC

non-player character



Spawn Point

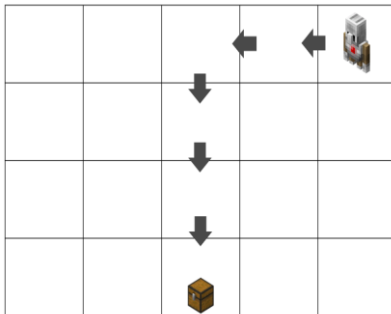
the location where a player begins game play



COMPUTER SCIENCE GLOSSARY

Algorithm

a sequence of defined steps



Code (Coding)

the method of giving a computer instructions to perform a specific task



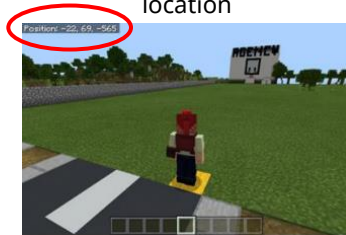
Code Builder

coding palette used to create programs



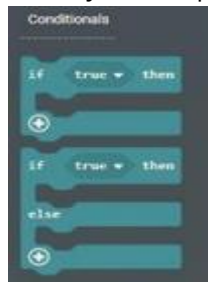
Coordinates

a coordinate represents a position or location



Conditionals

an action or occurrence that is detected by the computer



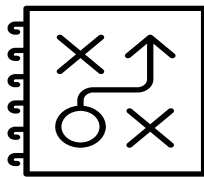
Debug

the process of finding (and correcting) errors in a computer program



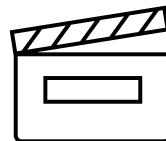
Decompose

breaking down a problem or task into smaller, more manageable parts



Event

an action or occurrence that is detected by the computer



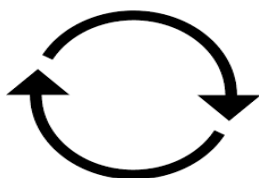
Event Handler

a part of your program that runs when a specific event happens (i.e., it "handles" the event)



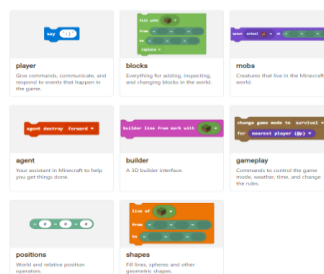
Loop

sequence that repeats a portion of code a set number of times until the desired task is complete



MakeCode Blocks

blocks used to create code



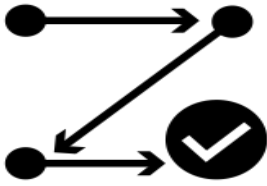
Sequence

a programmed (coded) algorithm



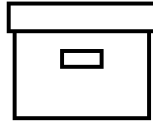
Test

the process of running a program to see if it is correct



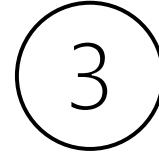
Variable

the places in which computers store values in a computer program



Number Variable

holds numeric data (e.g., person's age, a player's score, the year, etc.)



String Variable

a string of alphanumeric characters (e.g., a person's name, a password, the day of week, etc.)

{Friday}

Boolean Variable

only has two possible values: true or false



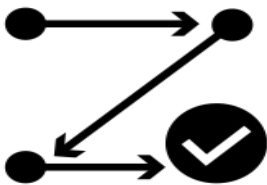
Position Variable

a special kind of variable that holds three numbers that describe a specific location in a three-dimensional space

(12, 4, -36)

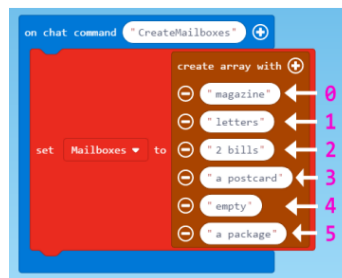
Test

the process of running a program to see if it is correct



Array

a series of places to store things



Variable

the places in which computers store values in a computer program

```
1 # Say Hello!
2 greeting = "Hello World!"
3 say(greeting)
4 # Introduce yourself!
5 myname = "Anonymous"
6 say(myname)
7 # try both
8 say(greeting, myname)
```

While Loops

repeats code when a condition is met

while

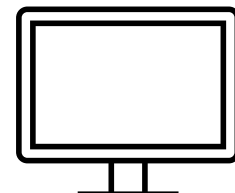
World Position

unique coordinates defined by three numbers

world(0, 0, 0)

Artificial Intelligence

a computer or software that mimics the human way of thinking



Iteration

the process of repeating steps

```
-----  
|   |  
|   v  
| Step 1 - "left foot forward"  
|   |  
|   v  
| Step 2 - "right foot forward"  
|   |  
|   v  
| - REPEAT
```

Function

a named section of a program that performs a specific task

