GameCode

7 Lessons

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

Curriculum Overview

Education.minecraft.net

INTRODUCTION

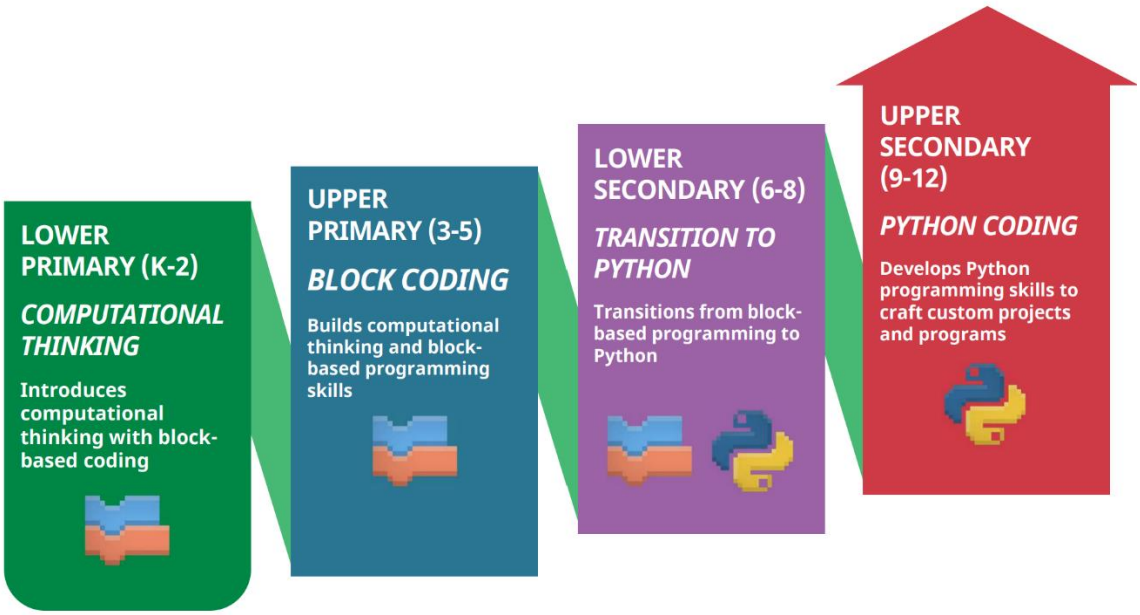
GameCode offers an immersive and innovative approach to coding education, harnessing the captivating world of Minecraft Education to ignite the passion for programming in students. Designed as a dynamic curriculum, GameCode blends the thrill of gaming with the fundamentals of coding, tasking students with crafting their very own arcade-style mini-games within the Minecraft universe. Through a series of engaging video tutorials, PowerPoint presentations, and comprehensive instructional content, participants embark on an exciting journey where creativity and logic intertwine.

In this hands-on program, students delve into coding by designing and programming their arcade games, honing their problem-solving skills while unleashing their imagination. They learn to manipulate Minecraft's mechanics, utilizing coding concepts to build interactive game experiences. With GameCode, students not only grasp the essentials of the MakeCode block coding language, but also cultivate critical thinking and teamwork as they collaborate with peers in bringing their arcade visions to life within the Minecraft world. This curriculum fosters a learning environment where innovation thrives and students emerge not just as coders but as inventive game designers, empowered to shape their digital landscapes.


AIM

Table of Contents

Computer Science Pathway progression



Game Code: My Arcade 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 coding skills by learning and building mini games in the Game Code: My Arcade. Students will watch self-guided videos to build their code to create mini games in the arcade. Students will learn important coding concepts and the associated MakeCode word blocks.

All of these lessons will provide robust learning experiences for students to build out their own knowledge of coding concepts. These coding concepts will be used in the final summative assessment – using all of their coding knowledge and skills to build their own arcade game!

Curriculum Design

Welcome to Game Code: My Arcade!

Minecraft Education is extremely excited to release new computer science lessons, but equally excited to share our new curriculum design to help even more teachers to engage in computer science instruction!

To support this high-quality learning experience, Game Code: My Arcade leverages video playlists as a learning tool for coding in Minecraft.

Why video playlists?

1. Greater Engagement

The human brain can process visuals faster than written text. Video playlists combine the elements of sound, text, and movement to help students better connect and understand the content.

2. Interactive

Videos are interactive – individuals can pause and move to different parts of the video to get more information or to follow the content. This feature makes videos more interactive and easier to use than text-only content. In Game Code: My Arcade, students are able to follow along with the video playlists to see the computer science concept/skill demonstrated and then to check for their own understanding as they compare their versions of the mini game to the example.

3. Accessibility

Videos offer a different modality to support various types of learners. The learning design in Game Code: My Arcade aims to encompass universal design principles and provide accessibility to as many learners as possible. Flexibility, multiple modes of access, and equitability are built into the design.

Curriculum Summary

Minecraft Education's Game Code is specially designed computer science content for secondary students. Students will learn, practice, and apply relevant computer science skills and concepts in Minecraft. The lessons are designed to provide students with an opportunity to build knowledge of the coding concept(s) through self-paced video content. Students will build out mini games to demonstrate and apply coding concepts and computational thinking.

Lesson Design

Each EDU guide contains multiple activities guided by video content; depending on the needs and abilities of students, educators should use discretion and modify/adapt the lesson activities appropriately, as they are also intended to be self-paced. Within the lessons, the activities will follow the instructional segments:

Connect	In the first step, the student is introduced to the new skill/concept. The skill/concept will **connect** to their prior learning and assess their current understanding of the skill/concept.
Construct	Next, the students will work alongside the video content to **construct** an example of the skill/concept in Minecraft.
Contemplate	The students will be provided with an opportunity to **contemplate** about their learning and respond to questions to demonstrate their learning.
Continue	Then, students will have a preview of what the next activity in the lesson will be and how it will **continue** their learning.

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 lesson is supported by its own PowerPoint presentation to provide structure and guide the educator through the activities for the lesson. The video content is embedded within the presentation.
Formative Assessments	After each activity in the EDU guide, there is an opportunity to check for student understanding of the concept taught within the lesson. These formative assessments are typically comprised of 2-4 questions directly related to the learning that just took place. These questions are included as the "Reflection Questions" slide in the EDU Presentations and the Student Workbooks.
Video Playlists	Every single lesson is fully supported by its own video playlist. Each of the mini games has been decomposed into smaller, more manageable pieces to support just-in-time learning. The video playlists support the entire student experience from start to finish.
Student Workbooks	Each lesson will provide a student-facing workbook. This workbook will help guide the students through each of the activities in the lessons.
Summative Assessment	At the end of the entire lesson sequence, students will be provided with a performance-based task to demonstrate their new knowledge and skills learned throughout the computer science unit, Game Code: My Arcade. This performance-based task can be assessed using the provided rubric.
Minecraft World Files	The specific world files needed to experience the instructional activities have been linked directly within the Educator's Guides. For Game Code: My Arcade, there is one comprehensive world file that will support all of the lessons within the EDU guides.

THEME Introduction

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

Essential Question: How can we use various coding concepts and their associated MakeCode word blocks to create different mini games in Minecraft?

Overview

The Game Code: My Arcade Curriculum is a series of block-based coding lessons intended to introduce students to critical coding concepts in the fun and engaging environment of Minecraft Education using MakeCode blocks with CodeBuilder.

Students will have the opportunity to participate in the following lessons:

- Getting Started (GameCode Tutorials)
- Parkour
- Hide and Seek
- Rock, Paper, Scissors
- Aim Trainer
- Dunk Tank
- Build Your Own Arcade Game

Pacing: 15 hours

Materials

Hardware	• The teacher will need a laptop or tablet with a projector for the plugged-in coding lessons. • Internet access will be required for the lessons and activities. • Each student will need a device to view the videos and complete the coding activities.

Software	• Minecraft Education needs to be installed on the devices to utilize these lessons. Use this link to find information about deploying Minecraft Education. • The teacher and students will need the relevant MCworld files for each lesson. These can be found in the in-game library.
Other Materials	• Student Workbooks (printed or in digital format) • Pencils/Pens

Minecraft Education Teaching & learning Framework

GOAL:
Provide students with the necessary skills to become creative coders and content creators

Coding Mindset

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.

CURIOUS
COMMITTED
COOPERATIVE
CONSIDERATE
CONFIDENT

Computational Thinking Skills

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.

DECOMPOSITION
PATTERN RECOGNITION
ABSTRACTION
ALGORITHMS

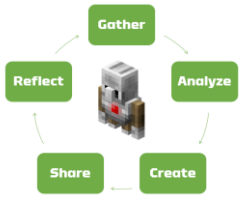
Computer Science Units of Study

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.

DIGITAL CITIZENSHIP
PROGRAMMING
CYBERSECURITY
IMPACTS OF COMPUTING

Community

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?".
We aim to empower students, develop confidence and self-efficacy into a commitment to serve the community in which we live in and beyond.



This unit will focus on **Programming**, as students will learn all about how to use Code Builder in Minecraft. Students will explore the various MakeCode blocks they can use to create mini games in their arcade.

This unit will focus on the coding mindset of being **curious**, as they create, test, and debug various programs throughout their coding journey to build out their Game Code: My Arcade.

This unit will focus on **algorithms** as students create programs that utilize step-by-step instructions to solve problems, complete tasks, and be creative!

This unit will provide students with the opportunity to consider how they can utilize their new knowledge of how MakeCode blocks can be used to create mini games in Minecraft. Students can bring together individuals within their **community** to play, test, and enjoy one another's mini games!

Ways to use this learning content

How can these lessons be used?

Integrated into computer science courses	Whether it's a digital tools course or an introduction to coding class, Game Code: My Arcade are great lessons to integrate into your existing course/class structure! Refer to the standards and/or the coding concepts to see where you could easily map these lessons into your curriculum for the school year.
STEAM Labs or Digital Makerspaces	Who wouldn't love to use Minecraft in a fully integrated learning environment where everything from the furniture and technology to curriculum and assessment work together to support hands-on, minds-on learning?! Game Code: My Arcade content could be used to support students' technology skills and develop their coding skills.
Afterschool learning opportunities	After a full instructional day, students need something fun and engaging to provide motivation for learning. What could possibly be more fun and engaging than Minecraft Education?! Not only will students love playing Minecraft, but they will also learn educational concepts and develop their skills

	in creativity, communication, collaboration, and critical thinking.
Summer camps	Are you planning or hosting a summer camp for students? Who wouldn't love a game-based learning camp?! This learning content could be included in a summer camp that supports student learning AND high engagement through the world of Minecraft!
Homeschool groups and other youth organizations	You don't have to be exclusively at a school or part of a school district to utilize Minecraft Education. Homeschools, families, and other organizations can purchase and manage Minecraft Education through our commercial offer. You can find more information here.

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.

Game Code: My Arcade

Day	Lesson	Coding Concept	Students will:	MakeCode Block Progression
*1**	*Hello?!*	Hello World	Use say block to create their first message in Minecraft	On start ___say
*2**	*Lights On*	Events	Use event blocks to turn on the lights in the dark arcade	___on chat ___on arrow shot ___on item used ___agent move ___agent interact
*2**	*Cobweb Cleanup*	Sequencing	Code the agent to move and turn in order to clean up the	___agent turn

			cobwebs in the arcade's switchboard	
*3**	*Cleaning Crew*	Loops (Repeat loops)	Code the agent to destroy the bugs that have infested the arcade	___agent destroy
*4**	*Circuit Corrections*	Conditionals (IF statements)	Code the agent to detect the missing glowstone and place a new block when necessary	___agent detect ___agent set block or item ___agent place
*5**	*Eliminate the Slime*	Loops (While loops)	Code the agent to inspect the arcade for any slime blocks and remove the blocks when found	___agent inspect ___block value comparison (is equal to, ==) comparison (is not equal to, !=)
*6**	*Parkour*	Variables Storing Data	Create a fun parkour game to play and challenge their peers	___agent teleport to player ___execute ("clear") ___give ___change game mode pause
*7**	*Hide and Seek*	Variables Loops (Nested loops) Conditionals (IF/Else statements) Randomization	Build a structure to play Hide and Seek against the agent	value block (for a constant or variable) pick random if/else logical not ___title
*8**	*Rock, Paper, Scissors*	Functions Conditionals	Program the agent to play rock, paper, scissors against a player	call function ___agent set active slot if/elseIf/…/else
*9**	*Aim Trainer*	Function Parameters Arrays	Create a target practice game that automatically counts your score	empty array add value get random value for element
*10**	*Dunk Tank*	Return Functions	Play a number guessing game to see	find index of length of array

			if a player can dunk the villager	comparison (greater than or equal to)
*11-15**	*Build Your Own Arcade Game*	All previous coding concepts	Demonstrate their understanding of coding concepts by coding to create their own arcade game	All previous blocks can be used in this experience

Educational Standards – Computer Science

Computer Science Teachers Association (CSTA) – United States

Lesson	CSTA Standard(s)
Lesson 1 - Tutorials	
Tutorial 1 – Hello?!	**2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. **2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
Tutorial 2 – Lights On	
Tutorial 3 – Cobweb Cleanup	
Tutorial 4 – Cleaning Crew	
Tutorial 5 – Circuit Corrections	
Tutorial 5 - Eliminate the Slime	
Lesson 2 – Parkour	**2-AP-11** Create clearly named variables that represent different data types and perform operations on their values.
Lesson 3 – Hide and Seek	**2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
Lesson 4 – Rock, Paper, Scissors	**2-AP-14** Create procedures with parameters to organize code and make it easier to reuse.

	2-AP-17 Systematically test and refine programs using a range of test cases.
Lesson 5 – Aim Trainer	**2-AP-14** Create procedures with parameters to organize code and make it easier to reuse.
Lesson 6 – Dunk Tank	**2-AP-14** Create procedures with parameters to organize code and make it easier to reuse.
Lesson 7 – Build Your Own Game	**2-AP-19** Document programs in order to make them easier to follow, test, and debug.

Educational Standards – Computer Science

International Society for Technology in Education (ISTE) – United States

Lesson	ISTE Standard(s)
Lesson 1 - Tutorials	
Tutorial 1 – Hello?!	**1.1.d** Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.
Tutorial 2 – Lights On	
Tutorial 3 – Cobweb Cleanup	
Tutorial 4 – Cleaning Crew	
Tutorial 5 – Circuit Corrections	
Tutorial 5 - Eliminate the Slime	
Lesson 2 – Parkour	**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.
Lesson 3 – Hide and Seek	**1.5.c** Students break problems into component parts, extract key information, and develop

	descriptive models to understand complex systems or facilitate problem-solving.
Lesson 4 – Rock, Paper, Scissors	**1.4.c** Students develop, test and refine prototypes as part of a cyclical design process.
Lesson 5 – Aim Trainer	**1.5.d** Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.
Lesson 6 – Dunk Tank	**1.5.d** Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.
Lesson 7 – Build Your Own Game	**1.4.a** Students know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.

Educational Standards – Computer Science

Computing Programmes of study – National curriculum in England

Lesson	KEY STAGE 3
Lesson 1 - Tutorials	
Tutorial 1 – Hello?!	Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem
Tutorial 2 – Lights On	
Tutorial 3 – Cobweb Cleanup	
Tutorial 4 – Cleaning Crew	
Tutorial 5 – Circuit Corrections	
Tutorial 5 - Eliminate the Slime	
Lesson 2 – Parkour	Create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
Lesson 3 – Hide and Seek	Understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds

	and pictures) can be represented and manipulated digitally, in the form of binary digits
Lesson 4 – Rock, Paper, Scissors	Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem
Lesson 5 – Aim Trainer	Understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers
Lesson 6 – Dunk Tank	Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
Lesson 7 – Build Your Own Game	Undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users

Educational Standards – Computer Science

Australian f-10 Curriculum – Digital Technologies

Lesson	Year 7 and 8 Standards
Lesson 1 - Tutorials	
Tutorial 1 – Hello?!	Investigating and designing some common algorithms (ACTDIP029)
Tutorial 2 – Lights On	
Tutorial 3 – Cobweb Cleanup	
Tutorial 4 – Cleaning Crew	
Tutorial 5 – Circuit Corrections	
Tutorial 5 - Eliminate the Slime	
Lesson 2 – Parkour	Developing and modifying digital solutions by implementing instructions contained in algorithms through programs (ACTDIP030)

Lesson 3 – Hide and Seek	Identifying that problems can be decomposed into sub elements (ACTDIP027)
Lesson 4 – Rock, Paper, Scissors	Checking the accuracy of an algorithm before it is implemented (ACTDIP029)
Lesson 5 – Aim Trainer	Developing a digital game that manipulates models of real-world objects (ACTDIP030)
Lesson 6 – Dunk Tank	Creating digital solutions that provide user navigation and prompts with controlled repetitions (ACTDIP030)
Lesson 7 – Build Your Own Game	Identifying features that make an effective game, such as storyline, goal, reward, gameplay and environment (ACTDIP028) Applying the principles and elements of design to a series of solutions to evaluate the success of each solution to hold the viewer's attention (ACTDIP028)

How to Get Started – Minecraft Education

Minecraft Education offers a unique learning platform where students can engage in creative builds and various game elements.

If you are licensed to use Minecraft Education through your O365 EDU account, download directly at aka.ms/download or reach out to your IT department for assistance. More information on deployment and license assignment can be found at aka.ms/meedeployguide.

Here's how to get setup:
1. First, check here to see if your school account is eligible.

2. If you do not have a valid O365 EDU account, you can still download and try a free demo on Windows, Mac or iPad.
3. Download Minecraft Education for Windows, Mac or iPad.

Once you are set-up with Minecraft Education, you should:

- Learn about Game Code: My Arcade with this video.
- Read through and familiarize yourself with the instructional materials included with Game Code: My Arcade.
- Watch the Game Code: My Arcade video playlists.
- Play through Game Code: My Arcade to get a better understanding of the lessons and how to navigate through the Minecraft world.
- Download and preview the EDU Guides, EDU Presentations, and Student Workbooks for the lessons.