

Educator Guide

Block 3 – Lesson 1
60 minutes
Single Student

Running Out of Time: The Experiment

EDUCATION.MINECRAFT.NET

THEME OVERVIEW

Tell students that while testing our latest time traveling device, we got stuck in a time loop! This time loop is sending us to random time periods in history. We must use our coding powers to overcome the obstacles and find our way to the time gate.

In Block 3, students will be exploring creativity in coding. Over the past 2 blocks (units), students have learned many coding concepts and how to use these coding concepts in their programs to accomplish goals in the lessons. Throughout those lessons, it has been reiterated that there may be more than one solution to an activity. This unit (Block 3) will allow students to explore that autonomy and use creativity in their coding solutions. These activities will either be student validated (i.e., the student achieves the objective successfully) or world validated (i.e., the student uses a specific code and/or block that lets the game know the objective has been achieved). This unit is intended to allow students to be creative with coding, persevere, problem solve, and try new things.

LEARNING OBJECTIVES

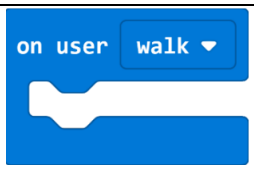
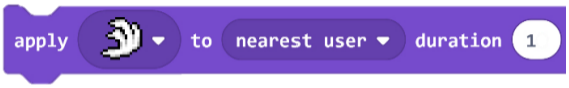
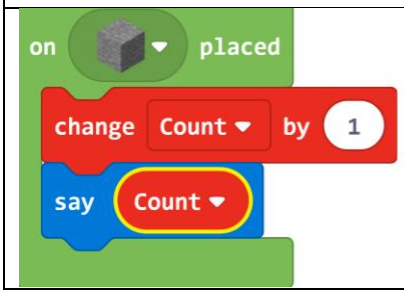
By the end of this lesson, students will:

- Use the coding concepts of events, event handlers, and variables to complete tasks
- Test and debug code
- Recognize there are multiples algorithms to solve the same problem and/or task
- Embrace a coding mindset

THINGS TO KEEP IN MIND

- Students are given a radio in the first slot of their hotbar. This item allows students to reset the coding activity.
- Remind students that there may be more than one solution for each of the activities.
- When using a while loop, students will need to press “C” again to deactivate the code (i.e., stop the code). To stop the Count variable, students will need to jump.
- Ask students to take a screenshot of their coding snippet and write an explanation of what the code does. (This can also be used as a homework assignment.)

NEW CODING BLOCKS

	**On user travelled** Runs the code when a student travels a certain way (walk, climb, swim)
	**Apply effect** Applies an effect to a Minecraft character or mob
	**On item used** Runs the code when the student right clicks while holding an item
	**On block placed** Runs the code when a defined block type is placed. Used along with VARIABLE container name

KEY VOCABULARY

Event – the change in the state of an object—this change is brought about as a result of some sort of user interaction; for example, when a block is placed, something else happens, or when the "Start" is activated

Event Handler – In programming, an event handler is part of your program that runs when a specific event happens (i.e., it "handles" the event). In MakeCode, an event handler block looks like a square with a gap in the middle and usually starts with the word "on": on block placed, on block broken, etc.

In real life, there are also events that might be associated with an action, like cause and effect. Here are some examples:

Event	Action
It starts raining	People open umbrellas
The bell rings	Students go to class
The power button is pressed	The computer turns on
The mouse button is clicked	An application opens

Event Handler – Most schools have lockers for students to keep their books in. Variables are a lot like lockers; variables are places in which computers keep

things. Computers need containers in which to keep numbers, text, and other information so that they can later make decisions. The code tells the computer when to look in these lockers and what to look for.
Here are some examples of things computers might store in variables:

EXAMPLE	VARIABLES
Weather Computer	temperature, time, day of the week, amount of rainfall, etc.,
Sports Computer	Touchdowns, player names, home runs, bases stolen, birdies, etc.
Clothing Computer	What is clean, what is dirty, what I want to buy, number of shoes, number of socks, etc.

LESSON ACTIVITIES

Lesson Introduction

This lesson is all about students learning how to think for themselves and problem solve. They will utilize the skills they have accrued over the past two units (blocks) to reach their objectives. Some new coding blocks are introduced primarily to help the students succeed. However, using variables is introduced as a method of storing data that can be reused. When the student opens MakeCode for this lesson, the code displayed for them is just a suggestion. These activities are player validated, in that, when the student/Agent achieves the goal, the activity is complete. It is not based on the game reading the "correct" code. This is to give the student the autonomy in coding and to allow them to be creative and resourceful in achieving their goals,

Direct Instruction (Teacher-Led; "I Do")

Welcome to Coding Fundamentals, Part 3! Today, our lesson will be "The Experiment". (**slide 1**)
Review the lesson objectives. (**slide 2**)
Review the important vocabulary. (**slide 3**)
Review the coding blocks that we will used today. (**slide 4**)

The students will be placed in a world where they must complete five obstacles to make it to a time gate. The obstacles will be made up of both player-validated activities and world-validated activities.

The code for player-validated activities are just suggestions and the student can use whatever method they can think of to move past the obstacle. World-validated activities will have strict objectives and will be communicated to the student.

Tell students:
Today, you are testing the latest invention, the Time Distortion Device (TDD). Before it can be activated, you will have to become familiar with the tools you will need for the journey. You will be placed in the testing course and must get to the time gate at the end. You will need to be creative and use all the tools at your disposal (i.e., your coding skills) to make it to the end of the course. (**slide 6**)

Guided Instruction (Teacher Modeling; "We Do")

At this time, have students log into Minecraft: Education Edition. When they spawn into the world, they will need to talk to the NPC, Dr. Rubin to begin their time travel. **(slides 7-8)**

Activity 1: Lava Swim (Player Validation) [Slides 9-11]

The students should use On Player Swim with Fire Resistance (with the default values) to give themselves the fire resistance ability to cross the lake of lava. The lava lake is only one block deep (i.e., the player does not need to swim). However, the lake is wide enough that the student cannot jump over it without assistance from the effects. If they try to cross the lake without fire resistance, they will burn up in the lava. The student's objective is to get to the other side.



Independent Work (Teacher Support; "You Do")

Activity 2: The Climb (Player Validation) [Slides 12-15]

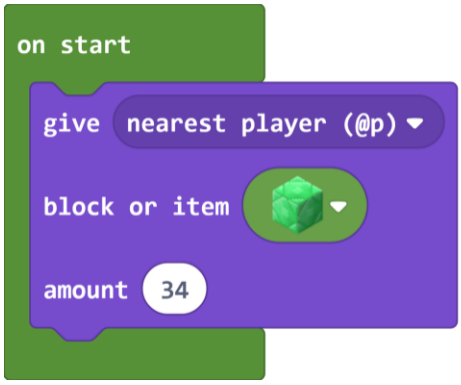
The student should use the Forever event to give themselves the Jump boost (using default values) to climb up the inaccessible area. There will be an area where the student must climb to reach the next area. Jump boost 2 (which the student should give themselves) gives them a jump boost to allow them to jump up 2.5 blocks. The students' objective will be to reach the top to access the next area.

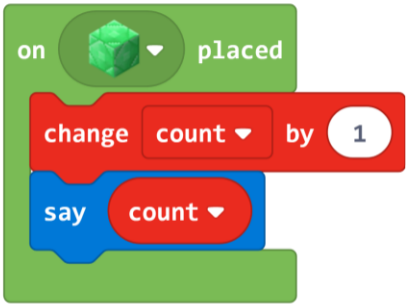


Activity 3: Building (World Validation) [Slides 16-19]

The students will be required to use On Block Placed event to start counting up a variable and saying the current number after each block is placed. The students will need to place at least 34 blocks to move on. They should be copying the pyramid they will see. The building blocks can be iron, diamond, gold, or emerald. In the event of a successful completion, the beacon will appear on top of the pyramid. In order for the game to validate the build and help keep students on track, they are going to use the On Block Placed event along with a Variable.

Tell students:
Imagine cookie jar in your family's home. What does it do? Holds cookies, right? But it doesn't always hold the same amount of cookies or the same kind of cookies, But the cookie jar does have the same purpose and can be used over and over again. A Variable works the same way. We are going to create a Variable called Count. The variable will be there, but we can change the number to whatever we need it for to achieve our objective. Right now, we want the game to count by 1 every time we place a block.

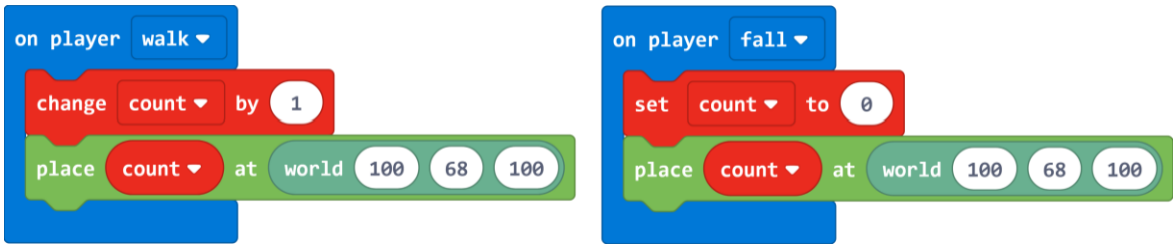




Activity 4: Change the World! (World Validation) [Slides 20-23]

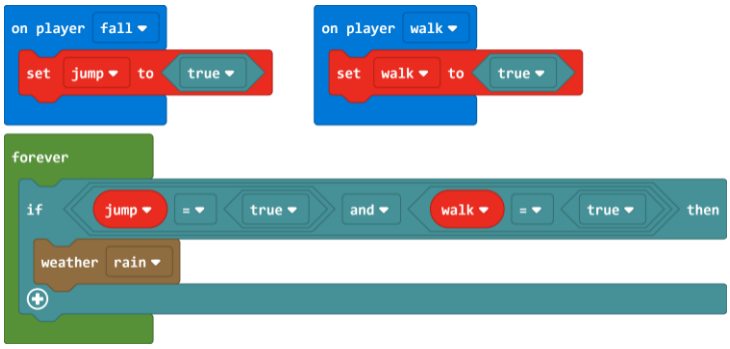
The student must figure out what the code is to open the door to the next area. They should use On Player Walk event to set a block at a specific set of

coordinates. Every time they walk, the count variable will increase by 1 and place a block that is associated with that block ID (e.g., 1=Stone, 2=Grass, 3=Dirt, etc.). Students should also use On Player Fall to reset the secret block. If they jump up, count will be reset to 0 and a block of air will be placed. Students need to determine the combination of blocks by walking around and jumping to reset the blocks.



Activity 5: Make it Rain! [Slides 24-27]

Students should string together multiple events to make it rain. This is done by setting a variable to "true" for each event when fired and then having a forever loop running and checking that all events are true; then making it rain. Students can use any event handlers to make it rain in Minecraft.



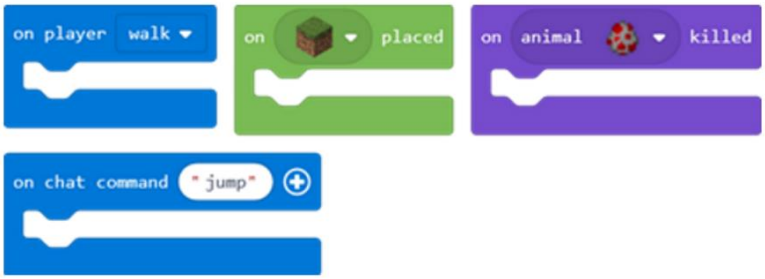
LESSON CONCLUSION

Ask students about the skills that they have learned during the lesson to reinforce the concepts learned. (**slides 28-29**)

1. What is an event?
   Answer: An event in computer science is an action or occurrence that is detected by a computer. For example, when someone selects the button on their mouse, it generates a "mouse click event" for the computer.
2. What is a variable?
   Answer: Variable are a lot like lockers: variables are places in which computers keep things. Computers need containers in which they keep

numbers, text, and other information so that they can use this information to make decisions later. The code tells the computer when to look in these lockers and what to look for.

3. What is an event handler? Name at least 2 event handlers.
   Answer: An event handler is a part of your program that runs when a specific event happens.
   Examples: On Player Walk/Swim/Fall/Swim Lava/Fly, etc.



4. What is one thing that you were excited about today?
   Answer: Answers may vary

These questions can also be used as a formative assessment. The formative assessment can be found at the end of this document that can be printed out.

EDUCATIONAL STANDARDS

UNITED STATES: CSTA

- 1B-AP-09 Create programs that use variables to store and modify data.
- 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.
- 1B-AP-10 Create programs that include sequences, events, loops, and conditionals.
- 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

UNITED STATES: ISTE

- 1.5b Students collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.
- 3D Students build knowledge by actively exploring real-world issues and problems, developing ideas and theories, and pursuing answers and solutions.
- 5C Students break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.

AUSTRALIA: DIGITAL TECHNOLOGIES (YEAR 3 AND 4)

- (ACTDIP016) Acquire, store and validate different types of data, and use a range of software to interpret and visualise data to create information
- (ACTDIP019) Following, modifying and describing the design of a game involving simple algorithms represented diagrammatically or in English
- (ACTDIP018) Experimenting with different ways of representing an instruction to make a repetition, for example loops in a flowchart diagram or using a 'REPEAT' statement
- (ACTDIP020) Planning and implementing a solution using a visual programming language, for example designing and creating a simple computer game involving decisions and repetitions, suitable for younger children, that requires user input to make selections, taking into account user responses

England: Computing (Key Stage 2)

- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- Use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

NAME: _____ **DATE:** _____

THE EXPERIMENT: FORMATIVE ASSESSMENT

What is an event?	
What is a variable?	
What is an event handler? Can you name at least two event handlers?	
What is one new and exciting thing you learned today?	