



## **Educator Guide**

3 Lessons

Single Student

# **Computing with Minecraft: 4- A Zoo**

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

## THEME OVERVIEW

This fourth unit in Computing with Minecraft provides students with the opportunity to explore how code can be used to facilitate the imagination and creativity. Students will focus on creating code with increasingly complexity, while using the BUILDER blocks.

## LESSONS OVERVIEW

All of the following lessons are intended to be completed in 45-60 minutes, depending on how much time is given to students to explore coding activities.

Lesson 1: Code a Zoo Entrance

Lesson 2: Code Zoo Paths

Lesson 3: Code Animal Enclosures

## LESSON OBJECTIVES

- Recognize more than one way to achieve the same result through code
- Utilize design thinking to create code
- Merge multiple coding concepts to complete large, varied tasks
- Use block code to change their Minecraft world











## THINGS TO KEEP IN MIND



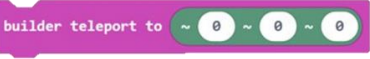

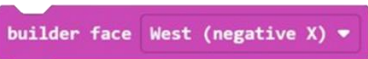

- Students are given a whistle in the first slot of their hotbar. This item allows students to call the Agent directly to them (i.e., instead on having students to code the Agent to teleport to them). The student will need to be standing and facing in the direction they would like the Agent to teleport to.
- Students are given a compass in the second slot of their hotbar. This item points to the world spawn point.
- Remind students that there may be more than one solution for each of the activities.

## MINECRAFT MECHANICS

C	<b>C</b> Summons the Agent and opens the MakeCode interface
T	<b>T</b> Opens chat panel in Minecraft for commands to be typed
ESC	<b>ESC</b> When a student wants to leave the game, leave chat, or pause the game

## CODING BLOCKS

	<b>On chat command</b> Runs the code when the student types the chosen text in the chat window
	<b>Agent Move</b> Tells the Agent to move in a certain direction by a defined amount
	<b>Agent Turn</b> Tells the Agent to turn left or right
	<b>Agent Place</b> Tells the Agent where to place a block
	<b>Agent Destroy</b> Tells the Agent to break a block in a specific direction
	<b>Agent Set Active Slot</b> Sets the agent's active inventory slot; Slots are counted from left to right and then top to bottom, starting at the top-left corner of the agent's inventory
	<b>Repeat</b> Creates a loop (which repeats a section of code until a specific condition is met)
	<b>Fill</b> Fill a volume of blocks in between two positions
	<b>Relative</b> Creates a new relative position; A relative position is the distance in each direction from the player's feet.
	<b>World</b> Creates a new world position; A world position is the distance in each direction from the world's origin, which is (0, 0, 0). The distance is measured in blocks.

	<b>Move</b> Moves the builder in the specified direction
	<b>Turn</b> Turns the builder in the specified direction
	<b>teleportTo</b> Teleports the builder to the specified direction
	<b>tracePath</b> Traces the path travelled since the last marked position with the specified block type
	<b>Face</b> Makes the builder face the specified direction
	<b>Spawn</b> Bring a creature into the game at a position you choose

## KEY VOCABULARY

**Minecraft** – game-based learning platform with endless creativity and possibilities

**Controls** – provides the ability to move during game play (will differ based on devices)

**MakeCode** – the coding editor program used in-game to create algorithms

**Agent** – your own personal Robot you can program to complete tasks for you in the game

**Loops** – repeats code a certain number of times until a condition is met

**Positions** – represents a location in the Minecraft world

**Blocks** – blocks that make the world of Minecraft can be manipulated by code: place, fill, clone, and replace are all possible

**Builder** – tool that makes it easier to build complex structures in Minecraft

**Mobs** – beings, or entities, in a Minecraft world

## ADDITIONAL RESOURCES

If you would like additional support about the concepts and skills covered in these lessons, review the following resources:

- **AGENT** blocks: <https://minecraft.makecode.com/reference/agent>
- **LOOPS** blocks: <https://minecraft.makecode.com/blocks/loops>
- **BLOCKS**: <https://minecraft.makecode.com/reference/blocks>
- **POSITIONS** blocks: <https://minecraft.makecode.com/reference/positions>
- **BUILDER** blocks: <https://minecraft.makecode.com/reference/builder>
- **MOBS** blocks: <https://minecraft.makecode.com/reference/mobs>

- For additional support with Minecraft: Education Edition, contact the support team at [aka.ms/meesupport](https://aka.ms/meesupport) or engage with the educator community at <https://educommunity.minecraft.net/hc/community/topics>

## LESSON ACTIVITIES

### LESSON 1: CODE A ZOO ENTRANCE

#### Direction Instruction (Teacher-Led; “I Do”)

We are going to continue to work on Computing with Minecraft (**slide 1**). Today’s lesson is called “Code a Zoo Entrance” (**slide 2**).

Review the objectives on **Slide 3**.

We are going to build our zoo entrance. We will handle this in 4 parts: code the pillars; position your Agent and run the code; code the sign; and run the code. (**slide 4**)

For this task, you are going to have students log into Minecraft: Education Edition. (**slide 5**). As they arrive into the Teleporter, they should walk towards the yellow portal to teleport to the Zoo. (**slide 6**)

#### Guided Instruction (Teacher Modeling; “We Do”)

##### Activity: Code a Zoo Entrance (Slides 7-18)

Welcome to the Zoo! We are going to begin the lesson (**slide 7**). Talk to the NPC (**slide 8**) and then select the “tutorial” button to begin coding (**slide 9**).

Now, it is time to begin creating the code for the coding task (**slides 10-16**).

**Step 1:** We need to rename the **on chat command** block to **build\_gates**. Drag the **agent set active slot** into the workspace. Then add an **agent move [forward]** and use the drop-down menu to change it to **up**.

**Step 2:** Drag a **repeat [4] times** block into your **on chat command**. Set the number to **15**.

**Step 3:** Drag the **agent place [forward]** block into the coding workspace; set it to **down**. Add an **agent move [forward]** to your main code. Use the drop-down menu to change it to **right** and leave the number at 1.

**Step 4:** Drag another **agent place [forward]** block into the coding workspace; set it to **down**. Add an **agent move [forward]** to the workspace, leaving it set to **forward** and 1.

**Step 5:** Drag another **agent place [forward]** block into the coding workspace; set it to **down**. Add an **agent move [forward]** to your main code. Use the drop-down menu to change it to **left** and leave the number at 1.

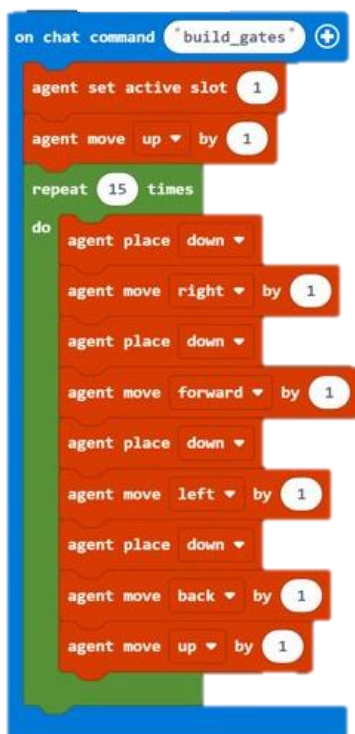
**Step 6:** Drag another **agent place [forward]** block into the coding workspace; set it to **down**. Now, we need to get the Agent to the next starting position to complete this part of gate's code.

**Step 7:** Drag two more **agent move [forward]** blocks to the workspace. Connect under the **agent place [down]** block. Use the drop-down menu to change **forward** to **back** in the first one and the **forward** to **up** in the second one.

Now, we are going to test our code for the first pillar. Use the whistle to summon your Agent to the location: **-34, 69, -561 (slide 17)**. Run your code.

Then change the coordinates for the second pillar: **-34, 69, -571**. Run the code. (slide 18)

### Coding Solution:



## Independent Work (Teacher Support; “You Do”)

Students will now be responsible for creating the zoo sign. (slides 19-25)



### Possible Coding Solution:



## LESSON 1 CONCLUSION

Upon completion of this lesson, students should be able to answer the following questions: (slide 26)

1. What does the **LOOPS** drawer help you do and when did we use it in this lesson?  
Answer: To build the gate to repeat code without having to rewrite each repeated line
2. What else could you add to your zoo gate to make it more exciting for your visitors?  
Answer: (responses will vary)



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

Allow students to reflect and share on their experience. (**slide 27**)

## LESSON 2: CODE A BUILDING

### Direction Instruction (Teacher-Led; “I Do”)

“We are going to work on Lesson 2 today. It is called ‘Code Zoo Paths.’” (**slide 28**)

Review the objectives on **Slide 29**.

In this lesson, students will create zoo paths. We are going to use FILL blocks to make this task much more efficient than manually digging and placing blocks. (**slide 30**)

At the time, have all students log into Minecraft: Education Edition. (**slide 31**)

### Guided Instruction (Teacher Modeling; “We Do”)

#### Activity: Code Zoo Paths (Slides 32-62)

Return to the Unit 3, Lesson 2 NPC. Press on the NPC to see the dialog box (**slide 32**). Select the tutorial button to launch Code Builder (**slide 33**). Then, demonstrate the following steps and have students follow along, step-by-step. (**slides 34-39**)

**Step 1:** Rename the run element of the **on chat command** to **zoo\_path\_1**. Select the **fill with** code block and drag into your **on chat command** block.

**Step 2:** Use the drop-down menu to change the type of material from **grass** to **gravel**.

**Step 3:** Drag two **world [0, 0, 0]** ovals into coding workspace and replace the **[~0, ~0, ~0]** ovals into the **from** and **to** elements.

**Step 4:** Add the coordinates **-40, 69, -575** to the **from** set of **world [0, 0, 0]** positions.

**Step 5:** Add the coordinates **-60, 69, -575** for the **to** set of **world [0, 0, 0]** positions.

**Step 6:** Change the center, or the Y coordinate, to one number lower. In this example, we are changing the coordinate to **68**. We must change this to one block lower so the code will replace the ground underneath us.

Use the whistle to summon the Agent to stand on the gold block. (**slide 40**) Test your code. (**slide 41**)



## Coding Solution:



## Independent Work (Teacher Support; "You Do")

Now, it is the students turn to create additional zoo paths. They are welcome to create whatever paths they would like within the space. (slide 42)

## Possible Coding Solutions:



## LESSON 2 CONCLUSION

Upon completion of this lesson, students should be able to answer the following questions: (slide 43)

1. Why must you pay close attention to whether each position number is a minus or not?  
Answer: There is no undo button so a mistake with the position numbers cannot always be fixed easily.
2. What's the difference between absolute world position and relative player position?  
Answer: Absolute world position = based on the position from the world's origin point (0, 0, 0); Relative player position = based on where the player is (~0, ~0, ~0)

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

Allow students to reflect and share on their experience. (**slide 44**)

## LESSON 3: CODE ANIMAL ENCLOSURES

### Direction Instruction (Teacher-Led; “I Do”)

We are going to work on Lesson 3 today. It is called “Code Animal Enclosures”. (**slide 45**)

Review the objectives on **Slide 46**.

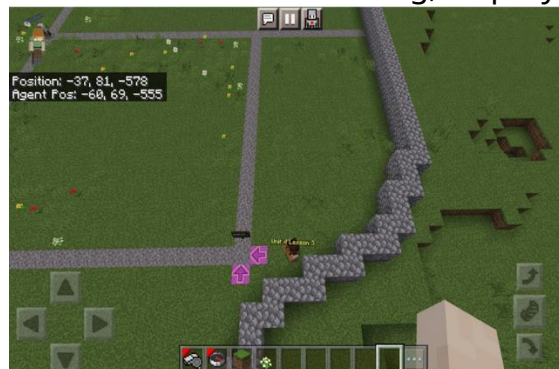
In this lesson, you will use new blocks. These blocks are called **BUILDER** blocks. **BUILDER** functions allow us to build using an invisible marker, as if a player or our **Agent** were present and building, but they are not! This method takes some practice to get used to, but its main benefit of building everything much faster is worth it! The **BUILDER** function builds everything much faster than the player or the **Agent** can. The placement of multiple blocks can happen in an instant and large structure are built within a blink of an eye. We will build a small structure to teach each of the most important elements of the **BUILDER** function. We will build in phases. (**slide 47**)

### Guided Instruction (Teacher Modeling; “We Do”)

#### Activity: Code Animal Enclosures

At the time, have all students log into Minecraft: Education Edition. (**slide 48**)

Return to the Unit 3, Lesson 3 NPC. Press on the NPC to see the dialog box (**slide 49**). Select the tutorial button to launch Code Builder (**slide 50**). Then, demonstrate the following steps and have students follow along, step-by-step. (**slides 51-60**)

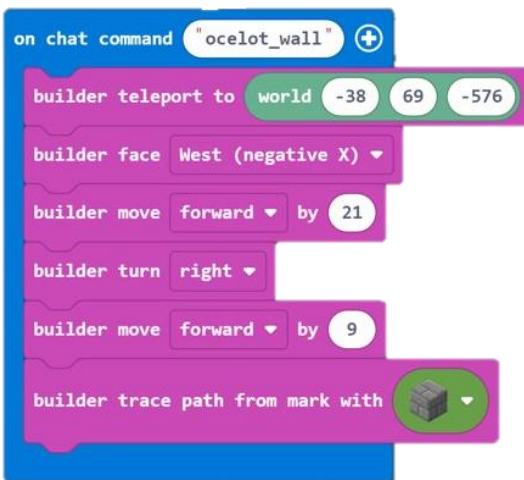


Students will use the BUILDER face block in this coding activity. Minecraft worlds have a North to South and East to West as well as up and down grid. The easiest way to find the direction is to use a compass. The compass is located in the 2<sup>nd</sup> slot of the hotbar. When your player is holding the compass, the red dial will always point North. In the image below, the direction on the compass is West. This is the direction we want the BUILDER to work.



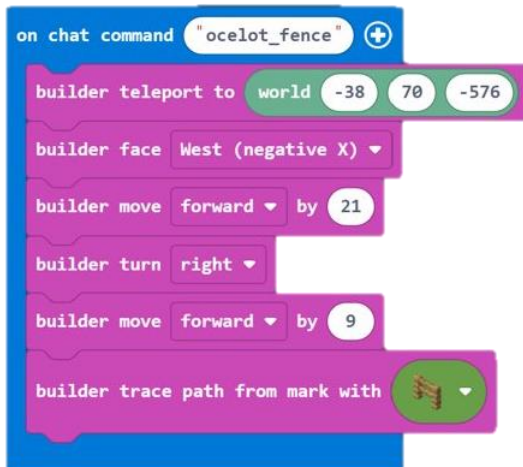
Students should test their code. (**slide 61**)

### Coding Solution:



Students are now going to add the fence. Return to Code Builder. Duplicate the existing code. Rename the on chat command to **ocelot\_fence**. Change the location to one block higher and change the material to oak fence. (**slide 62**) Students should then run and test their code. (**slide 63**)

### Coding Solution:



## Independent Work (Teacher Support; "You Do")

Students should now code an ocelot for their animal enclosure. (slide 64-65)

### Coding Solution:



## LESSON 3 CONCLUSION

Upon completion of this lesson, students should be able to answer the following questions: (Slide 66)

1. What does the **BUILDER** function do and what is its main benefit?  
Answer: It allows us to build using an invisible marker and builds everything much faster.
2. Why did we use the compass in this lesson?  
Answer: To tell the builder in what direction it should build

These questions can also be used as a formative assessment. The formative assessments can be found at the end of this document that can be printed out. Allow students to reflect and share about their experiences. (slide 67)

## EDUCATIONAL STANDARDS

UNITED STATES	AUSTRALIA	UNITED KINGDOM
<p>DEVELOP PROGRAMS WITH SEQUENCES AND SIMPLE LOOPS, TO EXPRESS IDEAS OR ADDRESS A PROBLEM.</p> <p>CSTA 1A-AP-10</p>	<p>WRITING AND ENTERING A SIMPLE SET OF INSTRUCTIONS JOINTLY TO SEQUENCE EVENTS AND INSTRUCTIONS</p> <p>FOUNDATION TO YEAR 2 ACTDIP004</p>	<p>UNDERSTAND WHAT ALGORITHMS ARE; HOW THEY ARE IMPLEMENTED AS PROGRAMS ON DIGITAL DEVICES; AND THAT PROGRAMS EXECUTE BY FOLLOWING PRECISE AND UNAMBIGUOUS INSTRUCTIONS</p> <p>COMPUTING KEY STAGE 1</p>
<p>MODEL THE WAY PROGRAMS STORE AND MANIPULATE DATA BY USING NUMBERS OR OTHER SYMBOLS TO REPRESENT INFORMATION.</p> <p>CSTA 1A-AP-09</p>	<p>PRESENTING A SEQUENCE OF INSTRUCTIONS OR EVENTS IN A SERIES OF SLIDES OR SCREENS WITH TEXT AND PICTURES</p> <p>FOUNDATION TO YEAR 2 ACTDIP004</p>	<p>CREATE AND DEBUG SIMPLE PROGRAMS</p> <p>COMPUTING KEY STAGE 1</p>
<p>STUDENTS BREAK DOWN PROBLEMS INTO COMPONENT PARTS, EXTRACT KEY INFORMATION, AND DEVELOP DESCRIPTIVE MODELS TO UNDERSTAND COMPLEX SYSTEMS OR FACILITATE PROBLEM-SOLVING.</p> <p>ISTE 5C</p>		
<p>STUDENTS UNDERSTAND HOW AUTOMATION WORKS AND USE ALGORITHMIC THINKING TO DEVELOP A SEQUENCE OF STEPS TO CREATE AND TEST AUTOMATED SOLUTIONS.</p> <p>ISTE 5D</p>		

**NAME:** \_\_\_\_\_ **DATE:** \_\_\_\_\_

**A Zoo: LESSON 1 FORMATIVE ASSESSMENT**

<p>What does the <b>LOOPS</b> drawer help you do and when did we use it in this lesson?</p>	
<p>What else could you add to your zoo gate to make it more exciting for your visitors?</p>	

**NAME:** \_\_\_\_\_ **DATE:** \_\_\_\_\_

**A Zoo: LESSON 2 FORMATIVE ASSESSMENT**

Why must you pay close attention to whether each position number is a minus or not?	
What's the difference between absolute world position and relative player position?	



**NAME:** \_\_\_\_\_ **DATE:** \_\_\_\_\_

**A Zoo: LESSON 3 FORMATIVE ASSESSMENT**

<p>What does the <b>BUILDER</b> function do and what is its main benefit?</p>	
<p>Why did we use the compass in this lesson?</p>	