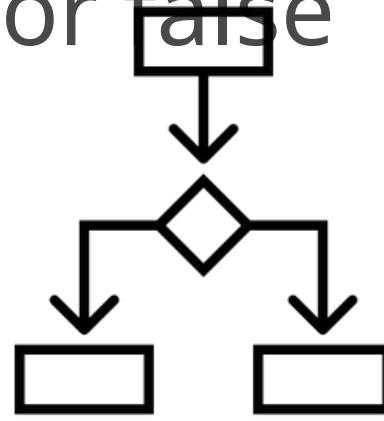
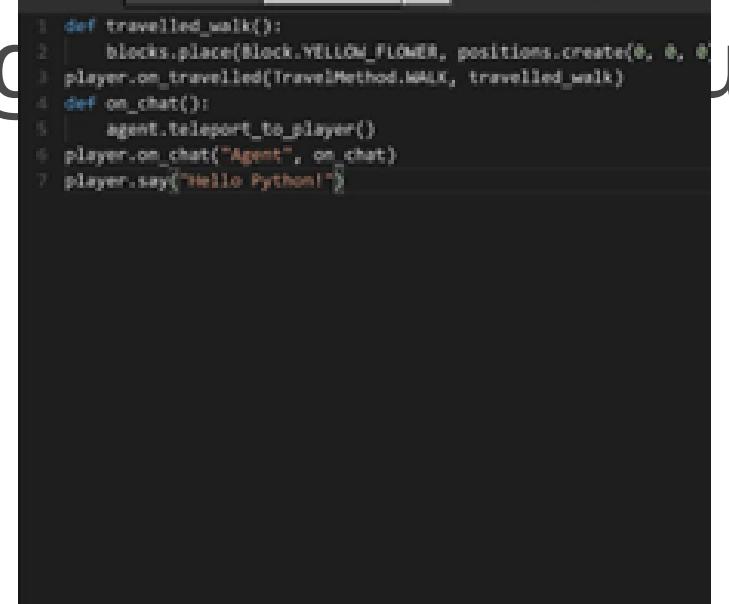


WHAT WE WILL LEARN TODAY?

- I will learn and apply the coding concept of conditionals.
- I will create, test, and debug my Python code.
- I will embrace and demonstrate a coding mindset.

IMPORTANT VOCABULARY

There are some important things for us to understand before we begin playing- let's review some concepts first!

Conditionals	Python	Syntax
<p>expressions that evaluate to either true or false</p>  <pre>graph TD; Start(()) --> Decision{ }; Decision -- True --> Path1[]; Decision -- False --> Path2[];</pre>	<p>a text-based computer program language</p>  <pre>1 def travelled_walk(): 2 blocks.place(Block.YELLOW_FLOWER, positions.create(0, 0, 0)) 3 player.on_traveled(TravelMethod.WALK, travelled_walk) 4 def on_chat(): 5 agent.teleport_to_player() 6 player.on_chat("Agent", on_chat) 7 player.say("Hello Python!")</pre>	<p>a set of rules that are used to create the programming language structure</p> <p>player.say("hi")</p>

GOAL FOR THE DAY



Welcome!

Today, you will continue to develop the Agent. CodingMine wants to code the Agent to drive a car by itself. Before testing the Agent on the actual road, you will code the Agent in different simulated traffic situations.

You will use conditionals and Boolean logic in your code for the Agent.

CODING CONCEPTS

Boolean Logic

Boolean logic is one of the most important concepts in computer science. Boolean logic uses the numbers **1** and **0**. These numbers represent if something is **true (1)** or **false (0)**. When programming with Boolean logic, you can use the following operators:

AND operator: This operator looks at Boolean statements and if they are both true (1), the operator will give us back a 1. However, if either one of those statements is a 0, then the operator will give us back a 0.

NOT operator: This operator looks at a Boolean logic statement and flips it. If the statement is 0 using the NOT operator, it will flip it to a 1. Vice versa, if the statement is 1 using the NOT operator, it will flip it to a 0.

Conditionals

There are different commands for conditionals:
if, **if else**, and **elif** commands.

- The **if** command is a conditional and will only run the code inside it if a defined condition is met (true). The outcome condition of a condition can only be true (1) or false (0).
- The **if else** command works the same way; however, if the if conditional is not met (0), the else conditional will run another piece of code.
- The **elif** command is a command that is placed after an if command and has its own defined condition. If the condition is true (1), the piece of code inside it will run.

SYNTAX FOR PYTHON

Compare ==	Indentation
<p>Double equals or compare is used when we want to compare two statements (Boolean, numerical, or string). The double equals are used with conditionals, to compare the defined condition against the Minecraft world outcome.</p>	<p>Indentation in Python are used when we want to declare that a piece of code belongs to the command above. To ident a piece of code, use the tab <code>for i in range(2):</code> <code> agent.move(FORWARD,1)</code>.</p>

WELCOME



This is your spawn point,
the location where you
begin game play.

TALK TO THE CEO



TALK TO THE CEO



This is the pop-up screen we will see on our screen.

After you have read the message, click on the “X” in the top right corner to continue game play.

WALK INTO THE ROOM TO BEGIN



ACTIVITY #1



Our first activity requires us to write code to help the Agent understand the meaning of the different traffic lights: green, red, and yellow.

TALK TO THE DEVELOPER



This is the pop-up screen we will see on our screen.

After you have read the message, click on the “X” in the top right corner to continue game play.

CREATE YOUR CODE

We will write 3 different sections of code; one for each of the traffic lights.



ACTIVITY #1

Part 1

You need to make the Agent move when there is a block to its left. When you run the code, the Agent will move forward.

(HINT: Replace the word “true” in the conditional and place your conditional instead).

Part 2

Now, you need to make the Agent stop when the light is red. When you run your code, the Agent will move when there is NOT a block on its left side.

Part 3

You will need to make the Agent wait for 2 seconds once it is at the yellow light. (HINT: 1000 milliseconds is equivalent to 1 second)

When you run your code, the Agent will move forward until there is a block on its left side. Then, it will wait for 2 seconds and continue.



The screenshot shows the Microsoft Code Builder interface. The title bar says "Code Builder". The top menu has "Blocks" and "Python" tabs, with "Python" selected. On the left is a sidebar with categories: BASIC (orange), PLAYER (blue), BLOCKS (green), MOBS (purple), AGENT (red), GAMEPLAY (brown), POSITIONS (teal), LOOPS (light green), LOGIC (light blue), VARIABLES (pink), MATH (light purple), and ADVANCED (dark grey). The main area contains the following Python code:

```
1 # Replace the lines below with your code #
2 # for loop set to 7
3 # Add the operator NOT to the condition below
4 # if conditional with an Agent detect condition
5 # Make the Agent move forward
6 # if conditional with an Agent detect condition
7 # Pause for 2000 ms
8 # Make the Agent move forward
9 # End of loop
```

To the right of the code, there are three vertical lines with labels: "Part 1" under line 1, "Part 2" under line 3, and "Part 3" under lines 4 through 9. In the bottom right corner of the interface, there are icons for back, forward, and search, followed by a large green "Run" button.

TEST YOUR CODE



You need to run your code after each part. This activity will be completed in 3 parts. The activity is complete when the Agent reaches the gold block.

MOVE TO THE NEXT AREA



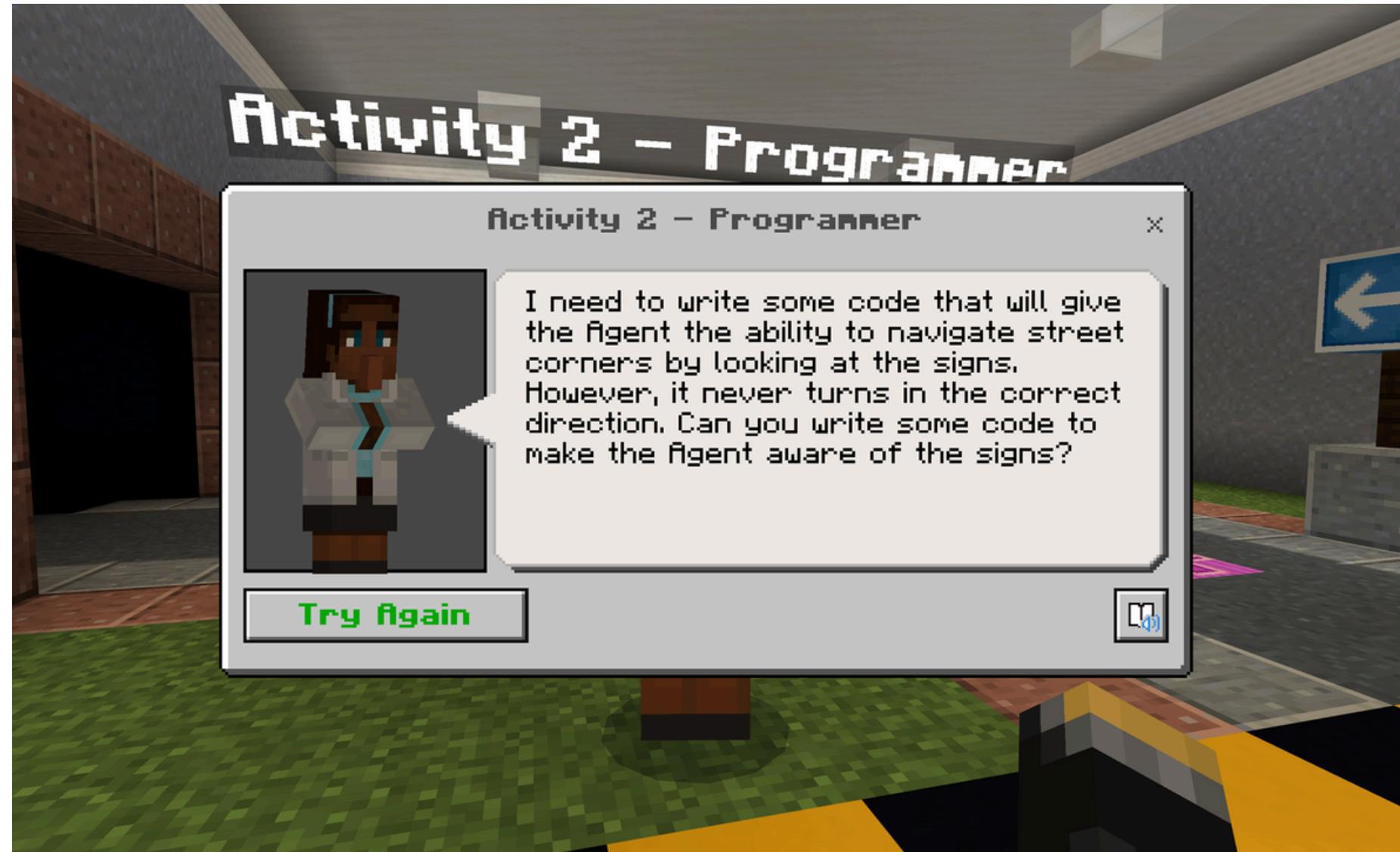
Continue on to the next area and begin Activity #2.

ACTIVITY #2



Move towards the programmer to find out about your next activity.

TALK TO THE PROGRAMMER



This is the pop-up screen we will see on our screen.

After you have read the message, click on the “X” in the top right corner to continue game play.

ACTIVITY #2: PART 1



We are going to need to write some code to help the Agent navigate the busy streets by reading the road signs.

ACTIVITY #2: PART 1

Part 1

You need to make the Agent turn left once it reaches the sign and the continue on to the gold block. When you run your code, the Agent should move forward until it reaches the turn sign. Then, the Agent will move in the direction of that sign and then finally stopping on the gold block.

The screenshot shows the Microsoft Code Builder application window titled "Code Builder". The interface includes a toolbar with "Share", "Blocks", "Python" (selected), and "Microsoft" branding. A sidebar on the left lists categories: BASIC, PLAYER, BLOCKS, MOBS, AGENT, GAMEPLAY, POSITIONS, LOOPS, LOGIC, VARIABLES, MATH, and ADVANCED. The main area displays the following Python code:

```
1 left = BLUE_GLAZED_TERRACOTTA
2 right = PINK_GLAZED_TERRACOTTA
3 # Replace the lines below with your code #
4 # Change value of loop below from 9 to 21
5 # for loop set to 9 | Part 2
6 # if else conditional with an Agent inspect condition | Part 1
7 # Make the agent turn left | Part 1
8 # elif conditional with an Agent inspect condition | Part 2
9 # Make the agent turn right | Part 2
10 # else part of the if else conditional | Part 1
11 # Make the agent move forward | Part 1
12 # End of loop | Part 1
```

At the bottom right are navigation icons for back, forward, and search, along with a large green "Run" button.

ACTIVITY #2: PART 2

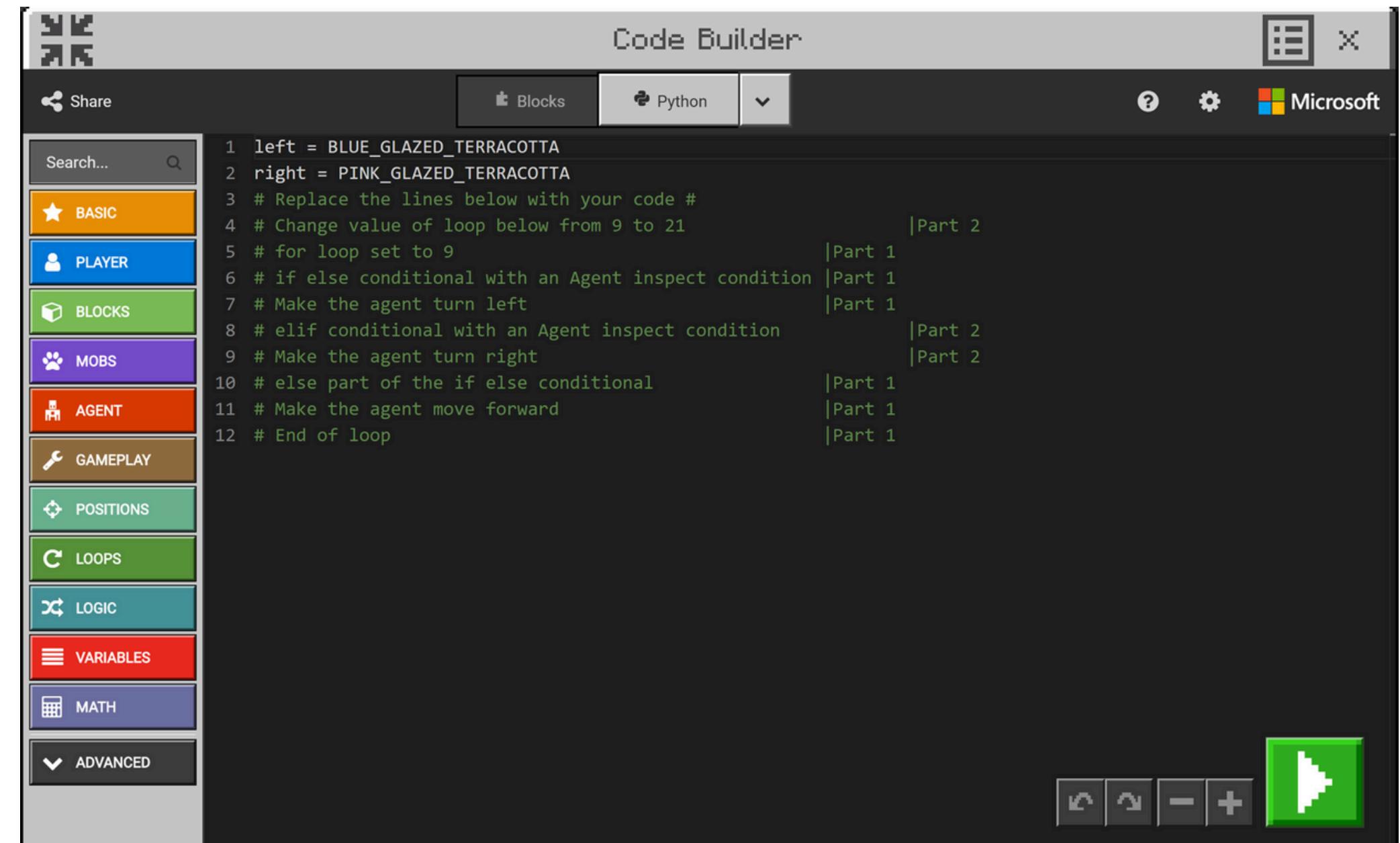


We are going to need to write some code to help the Agent navigate the busy streets by reading the road signs.

ACTIVITY #2: PART 2

Part 2

You need to make the Agent move forward, turning left and right until it reaches the gold block. When you run your code, the Agent will move forward until it reaches a turn sign. Then it will turn in that direction (as shown on the sign). When the Agent reaches the gold block, Activity #2 is complete!



The screenshot shows the Microsoft Code Builder application window titled "Code Builder". The interface includes a toolbar with "Share", "Blocks", "Python" (selected), and "Microsoft" branding. A sidebar on the left contains a search bar and categories: BASIC (orange), PLAYER (blue), BLOCKS (green), MOBS (purple), AGENT (red), GAMEPLAY (brown), POSITIONS (teal), LOOPS (light green), LOGIC (light blue), VARIABLES (red), MATH (purple), and ADVANCED (grey). The main area displays the following Python code:

```
1 left = BLUE_GLAZED_TERRACOTTA
2 right = PINK_GLAZED_TERRACOTTA
3 # Replace the lines below with your code #
4 # Change value of loop below from 9 to 21 | Part 2
5 # for loop set to 9 | Part 1
6 # if else conditional with an Agent inspect condition | Part 1
7 # Make the agent turn left | Part 1
8 # elif conditional with an Agent inspect condition | Part 2
9 # Make the agent turn right | Part 2
10 # else part of the if else conditional | Part 1
11 # Make the agent move forward | Part 1
12 # End of loop | Part 1
```

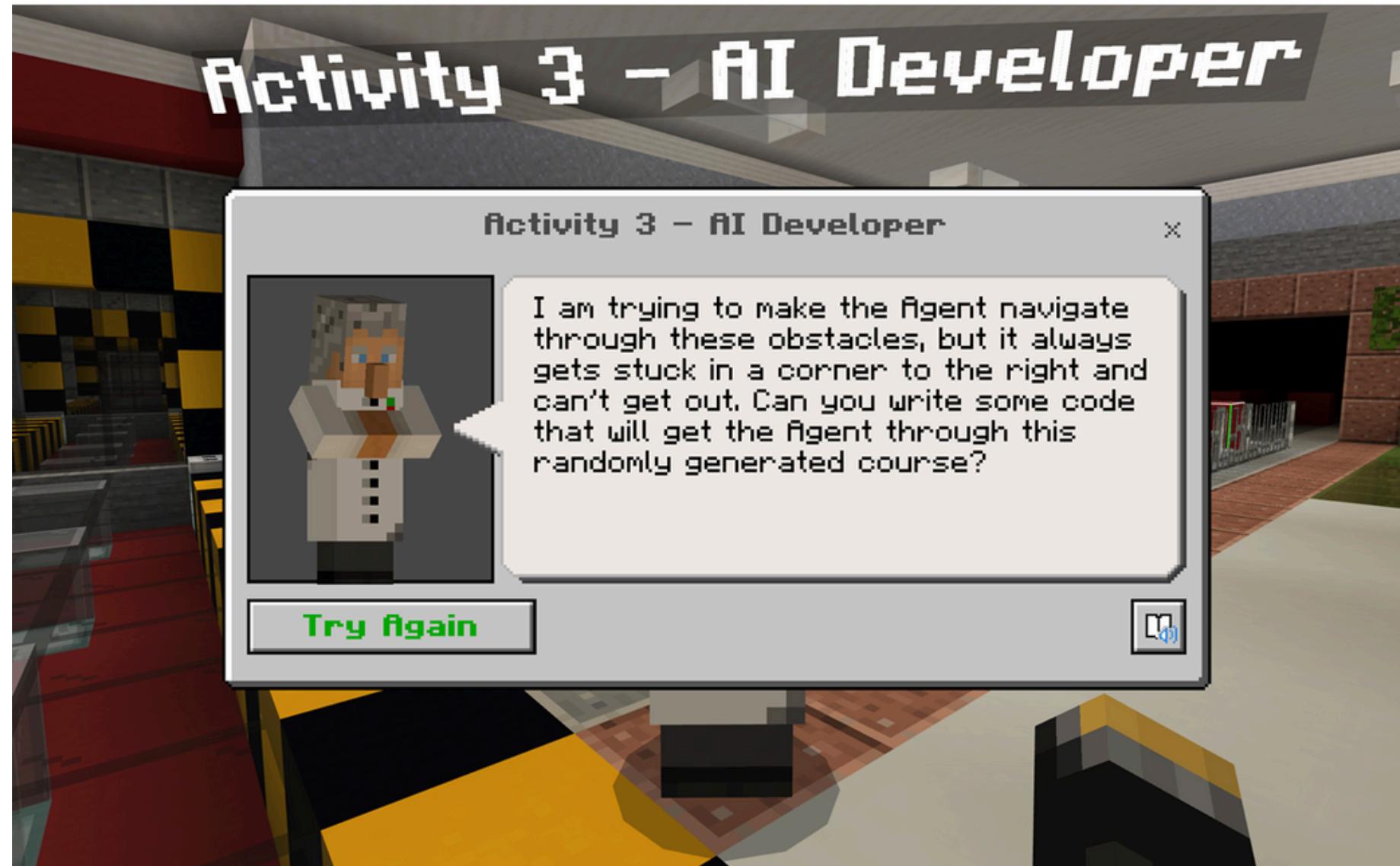
At the bottom right of the code editor are standard window control buttons (minimize, maximize, close) and a green "Run" button.

ACTIVITY #3



Walk over to the next area and talk to the AI Developer!

TALK TO THE AI DEVELOPER



This is the pop-up screen we will see on our screen.

After you have read the message, click on the “X” in the top right corner to continue game play.

ACTIVITY #3



In this activity, you need to write code using conditionals to help the Agent to detect obstacles. The Agent should be able to navigate through the end, no matter what blocks are placed. Once the Agent detects a block, it will be highlighted.

SUCCESS!



Recap

What you've done today:

- Learned and applied the coding concept of conditionals.
- Created, tested, and debugged my Python code.
- Embraced a coding mindset.



REFLECTION

- When talking about Boolean logic, what does 1 and 0 represent?
- What are conditionals?
- What does the **AND** operator do?
- What does the syntax == do?

