

Educator Guide

Python 101 - Lesson 10

60+ minutes

Creative Coding

2 PLAYER CHALLENGE

EDUCATION.MINECRAFT.NET

THEME OVERVIEW

In this lesson, the developers at CodingMine want to roll out the launch of the Agent to the public. However, it must be stress tested. The developers have created a testing area where the Agent must complete up to ten challenges, within a time limit. These challenges are a compilation of all the use cases the Agent has been coded to carry out throughout the unit.

The challenges will be carried out in one large coding arena. In this arena they will be given a coding challenge to complete, they will then have 3.5 minutes (210 seconds) to finish it.

This lesson has been designed for either one or two players. The students have the ability to play with other students, competing to see who can complete a challenge in the fastest time.

OBJECTIVE

By the end of the challenge, students will:

- Use all of their accumulated coding knowledge to accomplish a set number of tasks

CODING CONCEPTS

- Creative Coding
  During these tasks, they will have every command used throughout this course available in the coding window. This means that for each challenge there are several different ways to complete it. Tell the students that they can use whichever way they think is the most efficient to complete the challenge. The students can try this lesson and the coding challenges multiple times to see if they can get a better score each time.

CODING CHALLENGES

For each of the challenges, only one solution is given. As long as the code achieves the intended result, any other code is valid. Students will automatically be moved along to next challenge once 210 seconds have gone by.

Students will begin game play by talking to the CEO and then selecting one or two players. If they select two players, each student needs to go into a separate pod. Each student has a reset button in their coding arena; this button will only reset their challenge.

At the end of the lesson, the students will be able to see how many challenges they have completed.

Throughout the lesson, where applicable, the Agent has already been given all of the required block or items needed for that challenge.

Blinding Lights: **Nested Loops**
Objective: Tell the students that they need to code the Agent to place the Redstone lamps on top of the Redstone blocks whilst walking forwards along the walkway. Once the code is run the Agent will move forward while placing Redstone lamps on the Redstone blocks. When the Agent reaches the gold block the challenge is complete.

Code snippets:

```
for index in range(3):
    agent.place(RIGHT)
    for index2 in range(2):
        agent.move(FORWARD, 1)
    agent.place(LEFT)
    for index3 in range(2):
        agent.move(FORWARD, 1)
```

Row by Row: **Nested Loops**
Objective: Tell the students that they have to code the Agent to move through the area sideways, row by row, without it turning. Once the code is run the Agent will move through the area left and right, row by row, until it reaches the gold block, thus completing the challenge.

Code snippets:

```
for index in range(4):
    for index2 in range(8):
        agent.move(RIGHT, 1)
    agent.move(FORWARD, 1)
    for index3 in range(8):
        agent.move(LEFT, 1)
    agent.move(FORWARD, 1)
```

Pumpkin or Melon?: **Conditionals**
Objective: Tell the students that they need to code the Agent to move towards the gold block while breaking the melons but leaving the pumpkins alone. Once the

code is run the Agent will move forward, only breaking the melons. When the Agent reaches the gold block, the challenge is complete.

Code snippets:

```
for index in range(10):
    if agent.inspect(AgentInspection.BLOCK, RIGHT) == MELON_BLOCK:
        agent.destroy(RIGHT)
    agent.move(FORWARD, 1)
```

Diamond Rush: **Variables**

Objective: Tell the students that they need to finish the code so that the Agent moves towards the gold block and counts each diamond block it passes over. Once the code is run the Agent will move forward while counting the diamond blocks it passes over. At the end, when the Agent reaches the gold block, it will stack the number of diamonds passed over in a pillar and the challenge will be complete.

Code snippets:

Before:

```
diamond = 0
```

```
for index in range(11):
    agent.move(FORWARD, 1)
for index2 in range(diamond):
    agent.place(FORWARD)
    loops.pause(500)
```

After:

```
diamond = 0
for index in range(11):
    if agent.inspect(AgentInspection.BLOCK, DOWN) == DIAMOND_BLOCK:
        diamond += 1
    agent.move(FORWARD, 1)
for index2 in range(diamond):
    agent.place(FORWARD)
    loops.pause(500)
```

What's That Syntax!?: **Syntax**

Objective: Tell the students that they will be given a piece of code that does not run, as it has all of the syntax missing. They need to add the syntax in, to make the code run. Then they need to place the matching syntax blocks in the world.
(Hint: The syntax blocks for this challenge are in the chest)

Once the last syntax block has been placed and the Agent is standing on the gold block, the challenge is complete.

Code snippets:

Before:

```
for i in range 10
    agent move FORWARD 1
player say  Finished
```

After:

```
for i in range(10):
    agent.move(FORWARD, 1)
player.say("Finished")
```

Spring Arrival: **Debugging**

Objective: Tell the students that they have been given a code that does not work. This code is supposed to plant flowers on each block of grass by making the Agent move through the area row by row. Tell the students that they need to find the bugs in the code and fix them. Once the code is run the Agent will move through the area row by row, planting flowers on the grass blocks until it reaches the gold block thus completing the challenge.

Code snippets:

Before:

```
for index in range(4):
    for index2 in range(8):
        if agent.inspect(AgentInspection.BLOCK, DOWN) == GRASS:
            agent.place(DOWN)
        agent.move(FORWARD, 1)
    agent.turn(RIGHT_TURN)
    agent.move(FORWARD, 1)
    agent.turn(RIGHT_TURN)
    for index3 in range(8):
        if True:
            agent.place(DOWN)
        agent.move(FORWARD, 1)
    agent.turn(LEFT_TURN)
    agent.move(FORWARD, 1)
    agent.turn(LEFT_TURN)
```

After:

```
for index in range(5):
    for index2 in range(8):
        if agent.inspect(AgentInspection.BLOCK, DOWN) == GRASS:
            agent.place(DOWN)
        agent.move(FORWARD, 1)
    agent.turn(RIGHT_TURN)
    agent.move(FORWARD, 1)
    agent.turn(RIGHT_TURN)
    for index3 in range(8):
        if agent.inspect(AgentInspection.BLOCK, DOWN) == GRASS:
            agent.place(DOWN)
        agent.move(FORWARD, 1)
    agent.turn(LEFT_TURN)
    agent.move(FORWARD, 1)
    agent.turn(LEFT_TURN)
```

Depths of the Earth: Loops

Objective: Tell the students that they need to make the Agent mine the diamond ore behind it and the drop the mined diamond into the collections area at the other end. Make the Agent return, mine the emerald ore that will appear and drop it in the collection area. Once the code is run the Agent will mine the diamonds and emeralds behind itself and then drop them in the collection area at the other end.

Code snippets:

```
for index in range(2):
    agent.destroy(BACK)
    agent.collect_all()
    agent.move(FORWARD, 8)
    agent.drop_all(FORWARD)
    agent.move(BACK, 8)
```

Diamond or Dirt: Calculation and Boolean

Objective: Tell the students that they have been given four expressions to which they need to find the answer, so that they can move the Agent over to the gold block. Explain to the students that if the answer to the expression is 1 they should place a diamond block (True), and if the answer is 0 they should place a dirt block (False). The blocks need to be placed from left to right.
(Hint: The diamond and dirt blocks for this challenge are in the chest)

Once the code is run, the combination 1 0 1 0 will show up in the chat. This means that a diamond block (1) needs to be placed, followed by a dirt block (0) and then another diamond block and another dirt block. When the Agent reaches the gold block, the challenge is complete.

Code snippets:

```
player.say(10000 / 10000 + 64.64 + 64.64 - 72 - 57.28)
player.say(64 / 4 + 64 / 64 - 128 / 8 - 1)
player.say(19283746 / 19283746 - 1 + 1000 / 100 - 9)
player.say(8 - 9 + 7 + 32 * 2 - 64 / 2 - 38)
```

Do I Need to List it Out?: Lists

Objective: Tell the students that they have been given a list of the blocks. They then need to find out which block type the Agent and the player should stand on to complete the challenge. To find which block type the Agent should stand on they should sort the list alphabetically and choose the second block form the list. To find which block type the player should stand on they should reverse the list and pop the fourth block in the list, and then choose the sixth block in the list. Once the code is run, they will see that the answer for the Agent is emerald. The students should then press the button next to the emerald blocks to make the Agent stand there. They will also see that the answer for their player is diamond. They should then stand on the diamond blocks to complete the challenge.

Code snippets:

Before:

```
block_list = ["DIAMOND", "ICE", "EMERALD", "STONE", "WOOD", "GOLD", "Q
UARTZ"]
```

After:

```
block_list = ["DIAMOND", "ICE", "EMERALD", "STONE", "WOOD", "GOLD",
"QUARTZ"]

block_list.sort()
player.say("The Agents position is:" + block_list[1])
block_list.pop(3)
block_list.reverse()
player.say("The players position is:" + block_list[5])
```

The Agent Labyrinth: While Loops

Objective: Tell the students that they need to write some code to make the Agent go through the maze. They should use the colored blocks on the floor as forward, left, and right control directions in which the Agent will move. Once the code is run the students can control the Agent by standing on the different colored blocks. When the Agent reaches the gold block, the challenge is complete.

Code snippets:

```
while True:
    while blocks.test_for_block(LIGHT_BLUE_CONCRETE, pos(0, -1, 0)):
        agent.move(FORWARD, 1)
```

```
    while blocks.test_for_block(LIME_CONCRETE, pos(0, -1, 0)):
        agent.move(RIGHT, 1)
    while blocks.test_for_block(YELLOW_CONCRETE, pos(0, -1, 0)):
        agent.move(LEFT, 1)
```