

STAGE 7 – INTUITION

To make the game a little easier and more compelling, this stage provides players with prompts about what is in adjacent modules, replacing their real-life senses. If there are worker aliens in the next module, an astronaut will say, "I can hear scuttling!" If there is a ventilation shaft, they will say, "I can feel cold air!"



Add the following procedure to your code:

```
def intuition():
    global possible_moves, workers, vent_shafts
    #Check what is in each of the possible moves
    for connected_module in possible_moves:
        if connected_module in workers:
            print("I can hear something scuttling!")
        if connected_module in vent_shafts:
            print("I can feel cold air!")
```







0 1 2 3
tabs tab tabs tabs

In the main program section, underneath `if won == False and alive == True:`, add:

```
intuition()
```



Investigate

<p>ITEM</p> 	<p>Q: Why are <code>print</code> and <code>Print</code> not the same in a Python program?</p>
<p>STRUCTURE</p> 	<p>Q: The <code>intuition()</code> subroutine contains a counter-controlled <code>for</code> loop. Why are <code>for</code> loops considered to be counter-controlled?</p>
<p>PURPOSE</p> 	<p>Q: What are the possible outputs from the <code>intuition()</code> subroutine?</p>
<p>REASON</p> 	<p>Q: Why are two <code>if</code> statements needed instead of <code>elif</code>?</p>
<p>RELATION</p> 	<p>Q: The call to the <code>intuition()</code> subroutine is placed underneath this line of code:</p> <pre>if won == False and alive == True:</pre> <p>Why is it placed here?</p>
<p>APPROACH</p> 	<p>Q: How can we extend the <code>intuition()</code> subroutine so it provides the user with additional information?</p>



Programming project: Telium

Make

1 point

Add a comment above the `intuition()` subroutine that summarises the purpose of the new module.

1 point

Add extra code to check if the queen alien is in an adjacent module and output the message, "Listen! Did you hear that?"

1 point

Add extra code to check if an information panel is in an adjacent module and output the message, "There is a panel near here. We could use it to find lifeforms." Don't forget to include the `global` list too.

2 points

Add a new procedure to handle information panels. When a player enters a module with a panel, they can choose to interact with it. If they do, the space station loses 50 power, and the panel will output the location of the queen alien. The player cannot perform this action if power is less than 51.

3 points

Messages will be repeated if there is more than one of the same NPC or object in an adjacent module. For example, if there were two ventilation shafts in two adjacent modules, the program would output, "I can feel cold air! I can feel cold air!" This helps the player deduce what is in adjacent modules, but the repeating messages feel odd. Change the program so the message is not duplicated but, rather, a different message is output if there are two or three of the same object in adjacent modules. There is a good opportunity to use the scanner to output the message.

3 points

Add a new function to the player scanner – `SCAN`. The player chooses which module to scan, and the station loses 25 power. The scanner tells the player what is in the module. A player can only scan an adjacent module. The player cannot perform this action if the power is less than 26.

Evaluate

1. How effective do you feel the `intuition()` subroutine is in aiding gameplay?
2. Play the game for a while and suggest other ways the `intuition()` subroutine could be enhanced to help the player.

