

STAGE 5 – LOCKING A MODULE

To win the game, the player has to trap the queen alien, Telium. The queen will always try to escape if a player arrives in the same module. The queen will not go past the player to escape via the module the player has come from. Therefore, the queen can only be trapped in a module that has two or less exits.

When designing the space station, it is important that there are a number of places where this can happen. The player won't know how to trap the queen – or indeed, that this is how the game is won – until they deduce it through playing, adding a puzzle element to the game. By locking certain modules and entering the same module as the queen, it can be coerced into a game-winning position.

The game is won when the queen has no modules to escape to.

Locking a module costs power – this forces the player to think before they act, as the game is lost if the station runs out of power. To make the game harder, only one module's exits can be locked at any one time.

A module cannot be locked if it contains the queen alien.

The currently locked module is stored in a variable called `locked`.



Add the following procedure to your code:

```
def lock():
    global num_modules, power, locked
    new_lock = int(input("Enter module to lock:"))
    if new_lock < 0 or new_lock > num_modules:
        print("Invalid module. Operation failed.")
    elif new_lock == queen:
        print("Operation failed. Unable to lock module.")
    else:
        locked = new_lock
        print("Aliens cannot get into module",locked)
        power_used = 25 + 5 * random.randint(0,5)
```

In the `get_action` procedure, add the code in an appropriate place to handle the user choosing to lock a module when prompted for a move:







```
0         if action == "SCANNER":
1             command = input("Scanner ready. Enter command (LOCK):")
2             if command == "LOCK":
3                 lock()
4
```

tabs tab tabs tabs tabs



Programming project: Telium

Investigate

ITEM 	Q: Identify the main programming construct being used in the <code>lock()</code> subroutine.
STRUCTURE 	Q: Explain the selection statement used in the <code>lock()</code> subroutine.
PURPOSE 	Q: The second piece of code you added to the <code>get_action</code> subroutine contained a tested <code>if</code> statement – why was this needed?
REASON 	Q: The space station should lose some power each time the player uses their scanner to lock a module – this does not happen, but why not?
RELATION 	Q: The <code>if ... elif ... else</code> statement used in the <code>lock()</code> subroutine could have been written as two separate <code>if</code> statements. Why is it better to write it as shown here?
APPROACH 	Q: How would you modify the <code>lock()</code> subroutine to prevent a module from being locked if it contained a worker alien?



Programming project: Telium

Make

1 point

Comment the code so that each part is explained.

1 point

Add a line of code to reduce station power when a module is locked.

2 points

Add a POWER command to the scanner that outputs the space station's current power.

2 points

The procedure does not currently prevent a player from locking a module that is already locked – can you add that to the program?

3 points

The input is not fully exception-handled. Can you prevent the program from crashing if the user enters an invalid input for the module they want to lock?

3 points

Create a simple text parser for the game so the player can enter LOCK and the module in the same line. For example, an input of LOCK 2 would lock module 2. It should work without having to input the word SCANNER first – this makes the game more user-friendly for players who are familiar with the game.

3 points

Further extend the code from the previous task to work with a range of input sanitisation options – for example, L2 and lock2 have the same effect.

3 points

Add another command – LIFEFORMS – that outputs how many aliens (queen and workers) are currently detected in the space station.

Evaluate

1. In what ways is the validation insufficient when entering a module to lock?

