6.3
Big Quiz Project

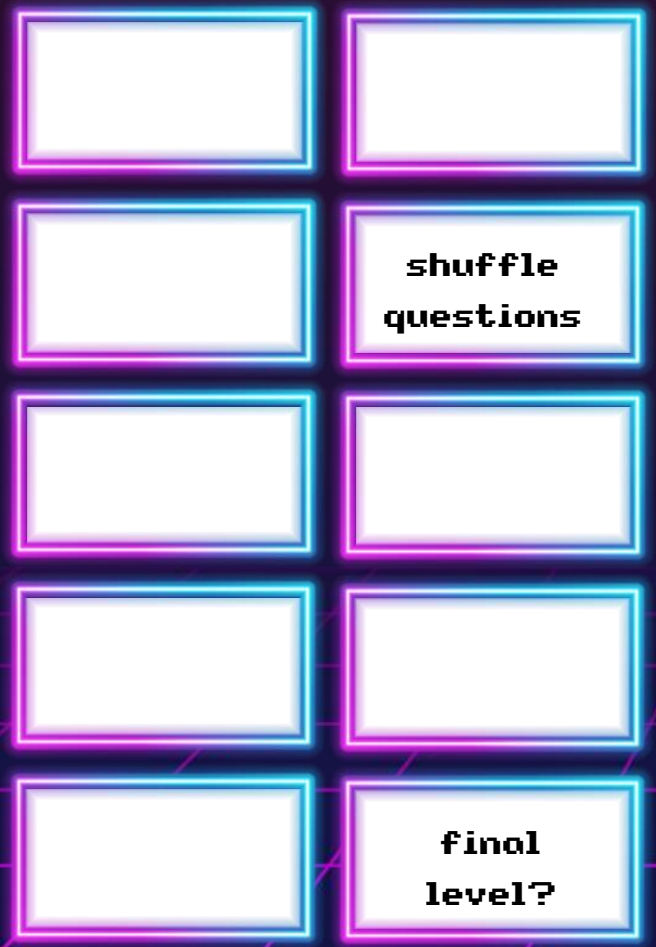
Part 3 of 3

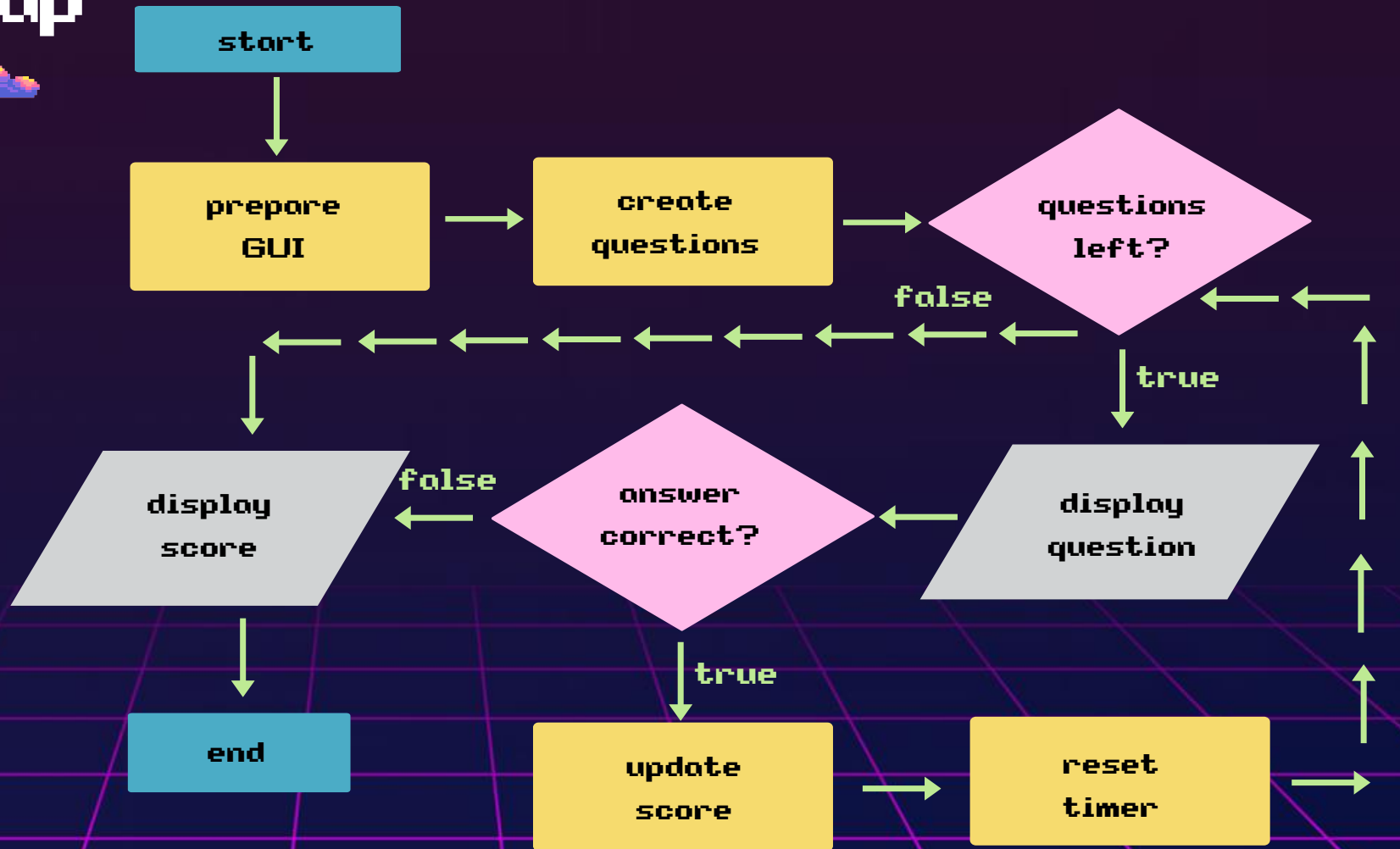
6.3 Big Quiz Project

It does not matter how well you did in Part 1. You all start from the same place!

- **Revise the flowchart:** We will do a quick recap of the flowchart.
- **Follow the recipe:** We will provide you with a clear outline to complete the project.
- **Support is available:** Refer to your notes, previous class code, or search online.
- **Remember:** Your teacher is here to assist you.
- **Sample code:** You will **not** start from scratch this time. You will start from a code sample. You should find it in your project.

Flowchart Recap

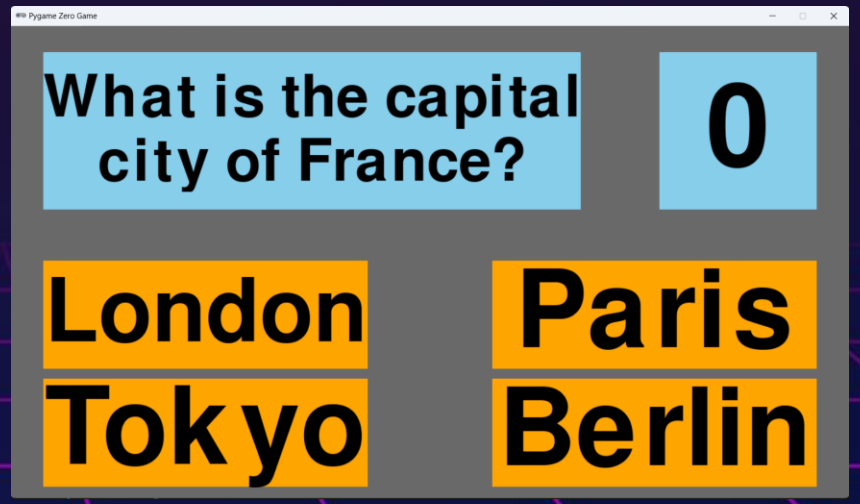




6.2 Big Quiz Project

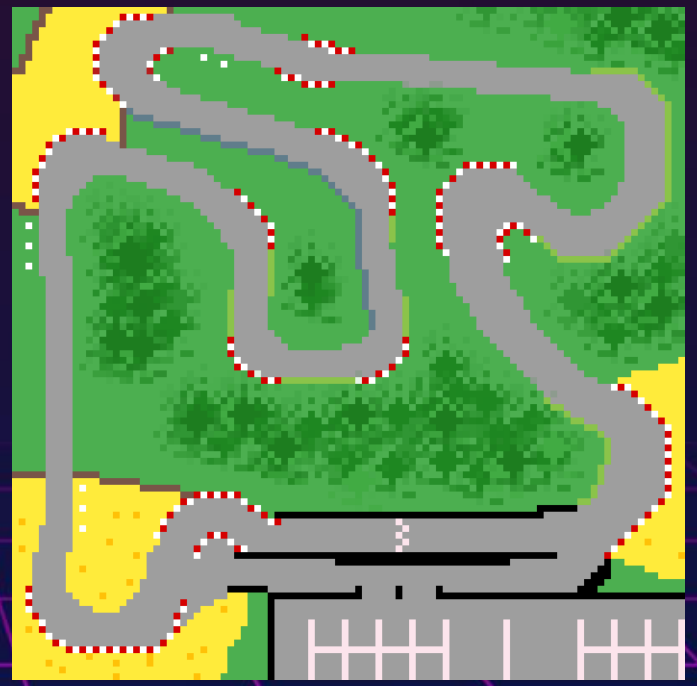
Your starting point works as it should!

- When you run the sample code you will notice that it displays a question, and it decreases the timer!
- This means that if after a change you get an error, it is because of something you added or removed.
- Whenever you make a change, no matter how small always test that you do not have **syntax errors**.



Plan of action

- You must complete two functions to make the game work: **on_mouse_down()** and **correct_answer().**
- Read all the tutorial slides and analyze the sample code, before attempting an exercise.
- You will complete the code that checks whether the player clicked on the right answer box.
- When the player got the right answer, then you need to set the next question, unless there are no questions left, and the game is over.

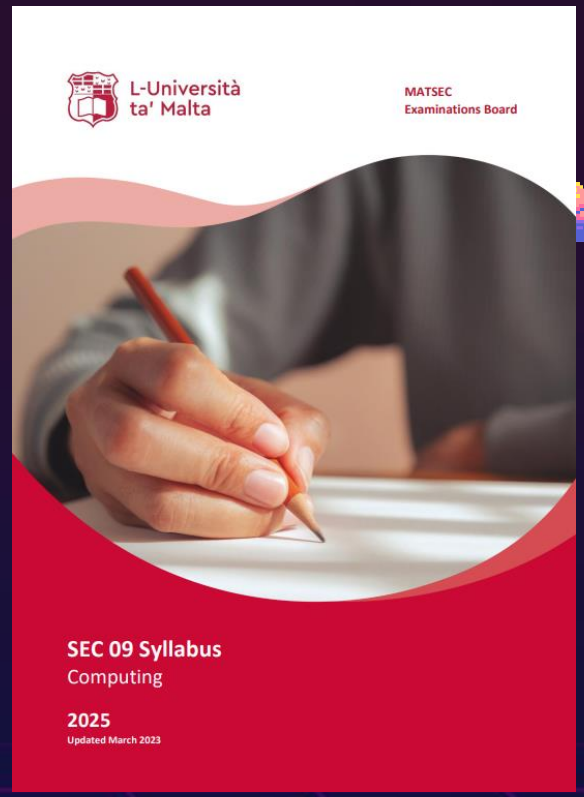


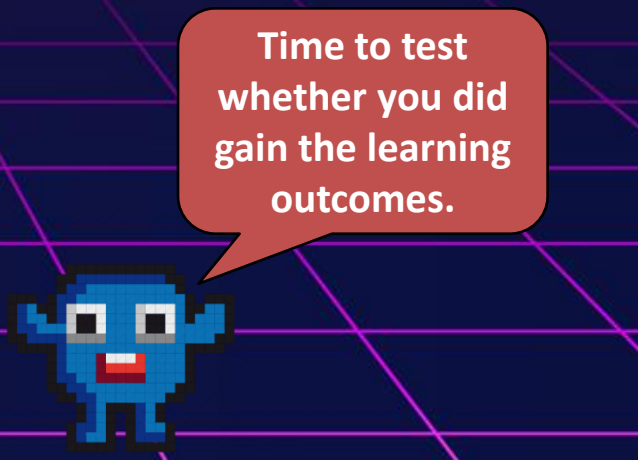
Over to you from here

- **Read the slides:** Go through the slides with instructions and explanations.
- **Review sample:** For each exercise, review the sample code and be familiar with the comments marked as **TODO**.
- **Easy tasks:** Identify the parts that you think you can do easily.
- **Just write code:** It is alright if your code does not work as you wish. You can still get a lot of marks.

```
def on_mouse_down(pos):
    # TODO: create an index variable and set it to 1
    # ... your code ...

    # TODO: loop through each answer_box
    #           if answer_box.collidepoint(pos)
    #                 if answer is correct
    #                 then correct_answer()
    #                 else handle_game_over()
    #           increase index by 1
    # ... your code ...
    pass

# TODO: Define a function correct_answer(), it does not take in parameters, and it does
#       Inside, write code to increase the score by 1

# TODO: After (inside correct_answer() still):-
#       if there are any questions left in the list,
#             then pop() the next_question like in line 63 and reset the timer to 10
#             else handle_game_over()
```

Assessment Criteria	
1. Application of Python Skills in a Scenario	
Appropriate use of variables including arithmetic statements.	3
Appropriate use of lists/dictionaries including access to them.	3
Use of user-defined functions	3
Construction of user-defined functions	3
Appropriate use of nested decision/iteration statements	6
Appropriate use of the Pygame Zero module	2
2. Game Functionality	
Detecting a mouse click on an answer box	3
Update the score when the answer is correct	3
Game over when no questions left or incorrect answer.	3
3. Programming Practices	
Abide by Python programming style conventions	1
Descriptive commenting	1
Identify and fix errors in the program independently	3
Total	**34**





Follow the recipe

- We have given you instructions on how to complete the game.
- You have **5 hairy tasks** to complete in this part.
- Read the instructions carefully and implement them as best as you can.
- You may always refer to code we wrote in class and search a little bit on the Internet.

6.1a Correct Answer Function

```
# TODO: Define a function correct_answer(), it does not take in parameters,
#       Inside, write code to increase the score by 1
```

- Define a function for **correct_answer().**
- It does not take in any parameters, and it does not return a value.
- For this exercise, the function will simply increase the score variable by one.
- However, to access the score variable you need to use the global keyword! The first line of this function should be:

  `global score, questions, next_question, time_left`.

Many students find defining a function difficult. But you got a lot of practice. Learn from your previous mistakes!

Are you stuck?

If you are finding it difficult to proceed, please use a help token and your teacher will give you more guidance.

6.1b Game Progression

- You will continue to write code inside the **correct_answer()** function.
- You will write code that checks whether the questions list has elements. If so, then you will **pop()** the **next_question** from the list. You need to use list methods to complete this code. It might be worth practicing a bit here first. There is also the **len()** function.
- Do not forget that when the player has a new question, the timer is reset. And if there are no questions, well we think you know the drill.

List Methods

Python has a set of built-in methods that you can use on lists.

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

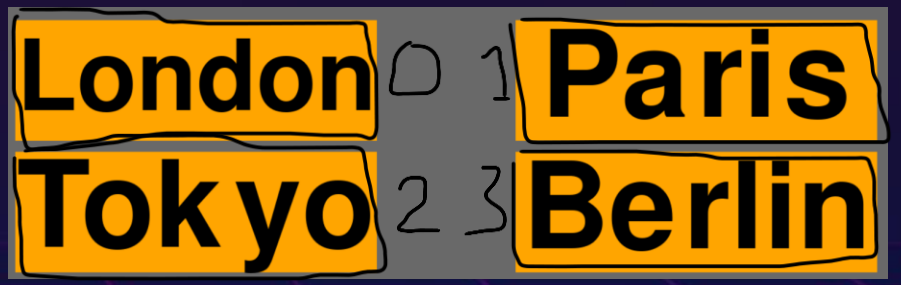
This is a hard skill. You might want to go through examples slowly first. You have time if you keep calm.

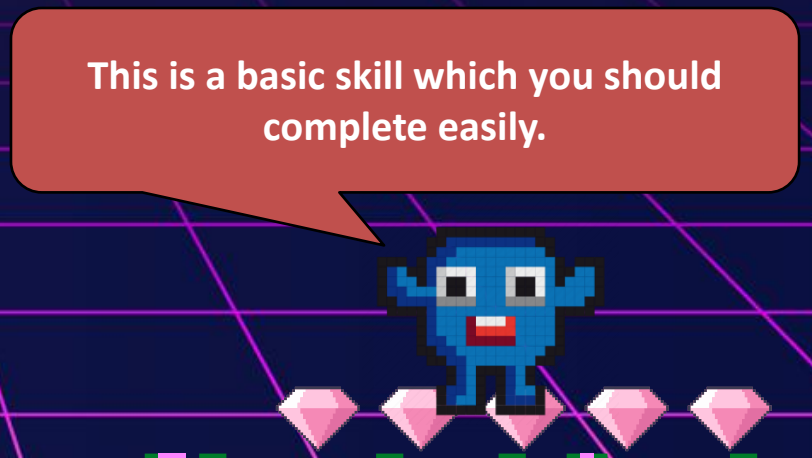
Are you stuck?

If you are finding it difficult to proceed, please use a help token and your teacher will give you more guidance.

6.1c Create an index variable

- You will now complete the code of the **on_mouse_down()** function. Remember, you have done something similar *before*.
- For this exercise, all you need to do is create a number variable called **answer_index** and set it to 0.
- The **answer_index** will store the position the **answer_box** the user clicked on! For example, if the player clicked on “London” then answer_index will store 0.





6.1d Player clicked on answer

- You will write code that checks whether the **pos**ition of the mouse click collides with an **answer_box**. You have done something of the sort *before*.
- Remember, to check whether an element collides with a position we use the **collidepoint(pos)** function.
- On the right is an example of how the function can be used. Please remember, we do not have *alien* in our game, we have **answer_box** from a list of **answer_boxes**. So, think!!

```
def on_mouse_down(pos):
    if alien.collidepoint(pos):
        print("Eek!")
    else:
        print("You missed me!")
```

Learn from your previous SBAs and lessons. Do you know which part we are talking about?

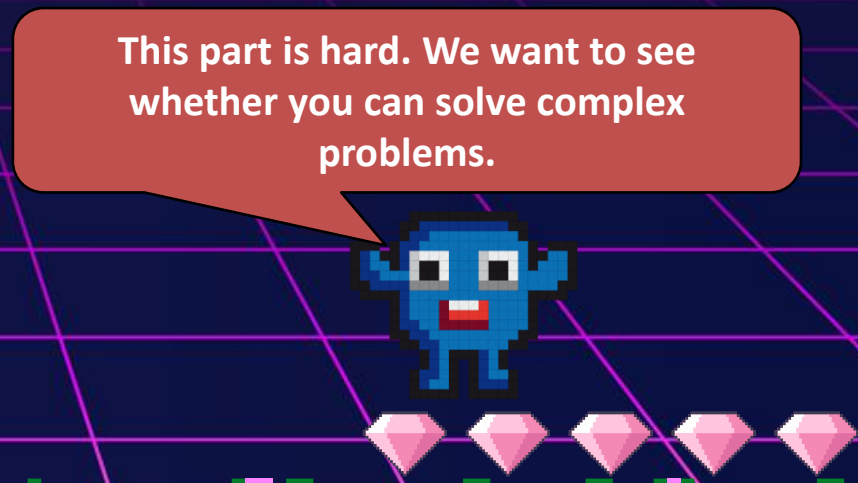
Are you stuck?

If you are finding it difficult to proceed, please use a help token and your teacher will give you more guidance.

6.1e Is the answer correct?

- The question data is stored as a dictionary as shown on the right. It has a particular key which points to the answer!
- Write code that checks whether the **answer_index** also points to the same answer.
- For example, for the question on the right, **answer_index** must have a value of 1.
- The comments in the sample code should guide you enough to complete the rest.

```
q5 = {
    "q": "What is the capital city of France?",
    "o": ["London", "Paris", "Tokyo", "Berlin"],
    "a": 1
}
```

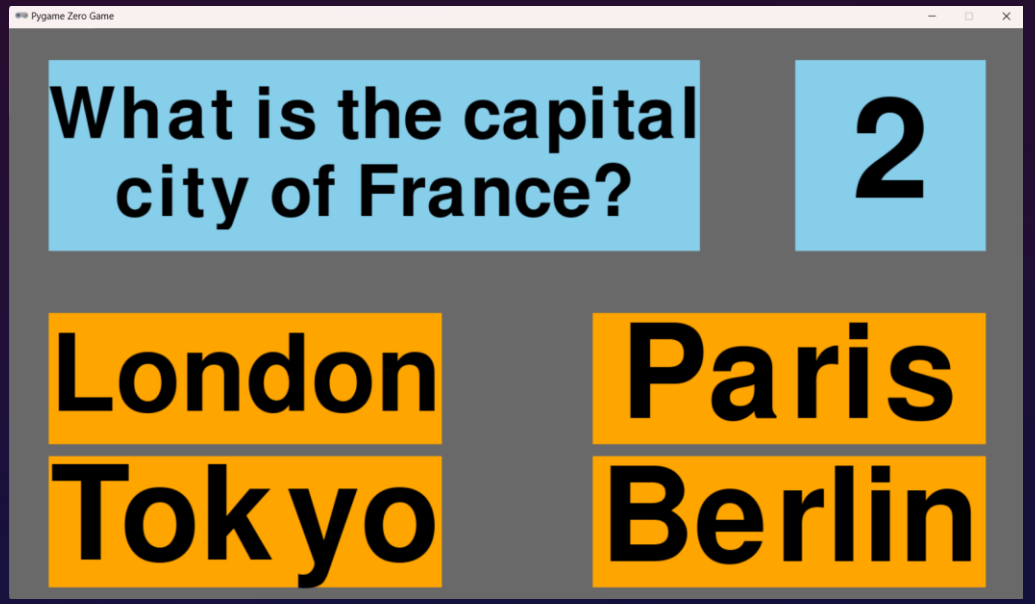


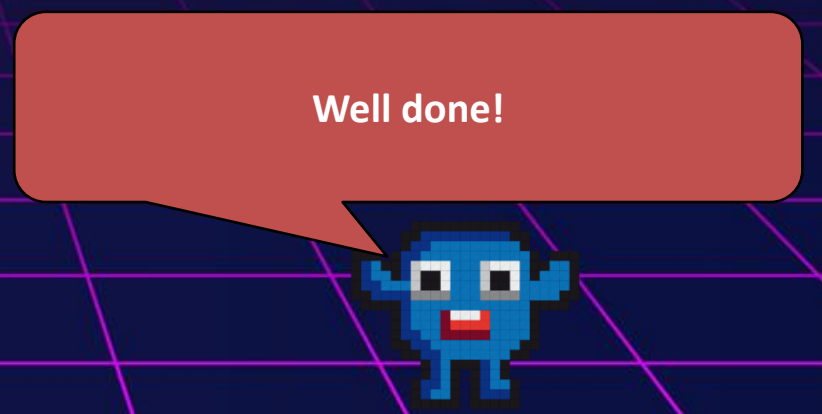
Are you stuck?

If you are finding it difficult to proceed, please use a help token and your teacher will give you more guidance.

Time to play!

- If you have completed all the exercises as instruct, then it is time to test the **final version of your game**.
- Run your code and ensure that there are no **syntax errors**.
- It is also a good idea to review the assessment criteria and make sure that your code is easy to correct by a teacher.





You have leveled up!

You are making great strides. Well done.