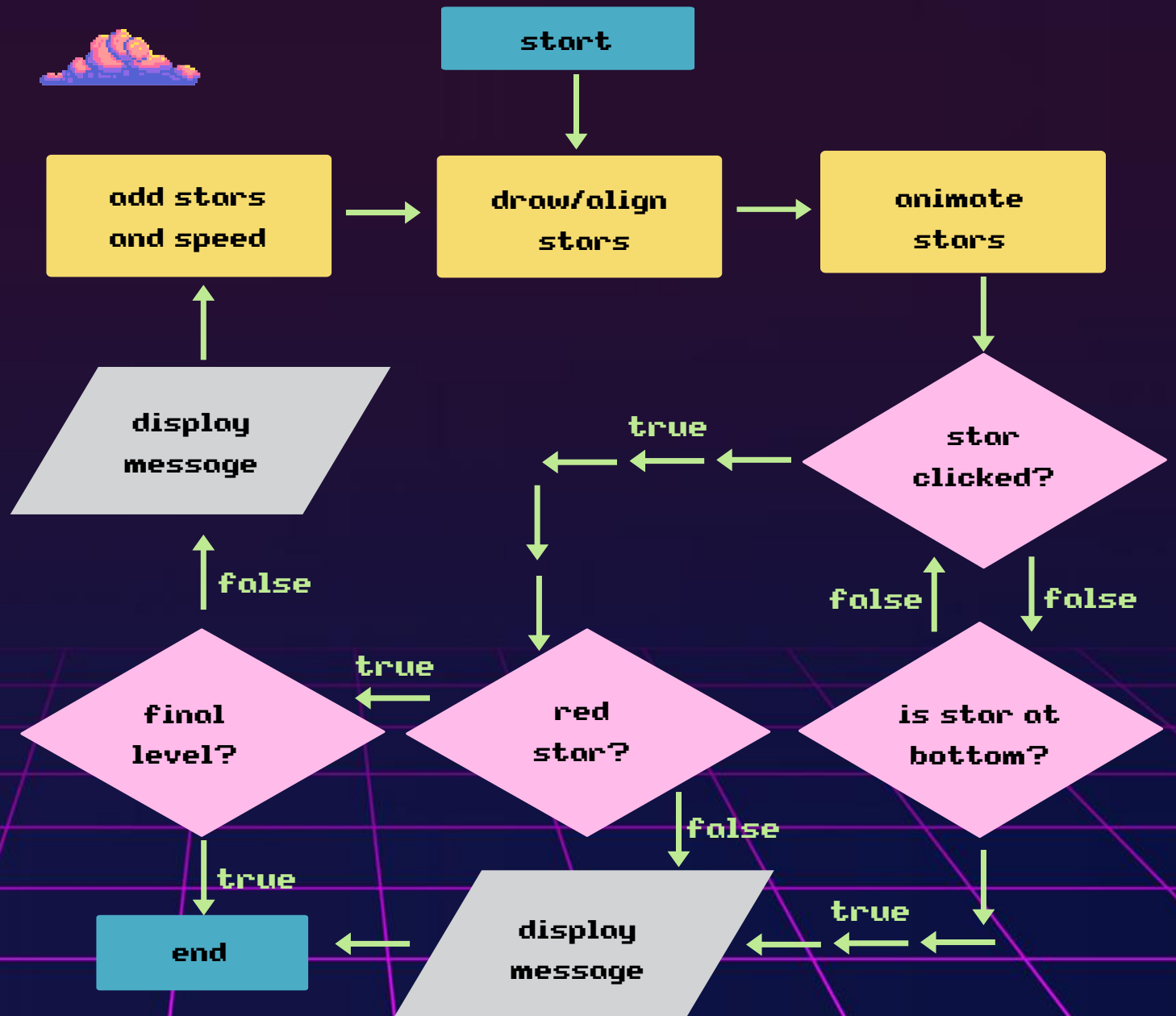# 5.3
Red Alert Project

5.3 Red Alert Project

- Last lesson we worked a lot on the star elements that are displayed and animated on screen.

- Up until now you worked on this game under the leadership of your teacher.

- Have you gained enough confidence to take over the spaceship?

- In this lesson you will complete this game without your teacher.

- Don't worry, we will give you an easy-to-follow recipe.

- Remember, you are never too far away from your coding buddies.

Flowchart Recap

start

add stars and speed → draw/align stars → animate stars

animate stars → star clicked?

star clicked? — **true** →

star clicked? — **false** → is star at bottom?

is star at bottom? — **false** → (up to star clicked?)

true → red star?

red star? — **true** → final level?

red star? — **false** → display message

is star at bottom? — **true** → display message

display message → end

final level? — **false** → display message → add stars and speed

final level? — **true** → end

Over to you from here

- When you run the sample, the screen loads Level 1 and animates the stars until they reach the bottom ending the game.

- Complete the **on_mouse_down()** function to react to the player's mouse clicks.

- Write code for the **click_red_star()** function that stops the animations and updates the current level if the game is not complete.

- Write code for the **stop_animations()** function which accepts the list of stars as a parameter. This is a simple function that stops the animation of each star.
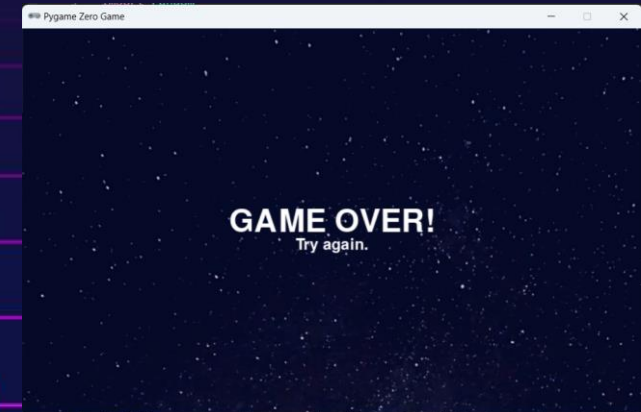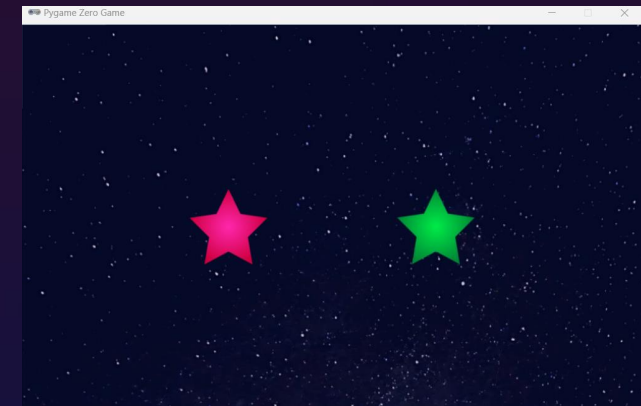
Assessment Criteria

1. Application of Python Skills in a Scenario

Appropriate use of variables	1
Appropriate use of lists	2
Appropriate use of arithmetic statements	3
Appropriate use of nested decision/iteration statements	6
Appropriate use of user-defined functions	3
Construction of user-defined functions	3
Appropriate use of the Pygame Zero module	2

2. Game Functionality

Detection and handling of a mouse click on a star	3
Game progression when player clicks on a red star	3
Stop star animations when required	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**

L-Università ta' Malta

MATSEC Examinations Board

SEC 09 Syllabus
Computing

2025
Updated March 2023

Time to test whether you did gain the learning outcomes.

Follow the recipe

- We have given you instructions on how to complete the game.

- You have three tasks to complete in this exercise.

- Read the instructions carefully and implement the TODO exercises in the sample code as best as you can.

- You may always refer to code we wrote in class and search a little bit on the Internet.

5.3a Handle Mouse Clicks

- Work on the function named **on_mouse_down()** which is **invoked automatically** every time the player clicks on the screen.

- Pygame Zero provides the **pos**ition of the click through this function.

- Write the code that identifies whether the position is on any one of the stars.

- The game will continue if the position is on the **red** star, otherwise the game is over.
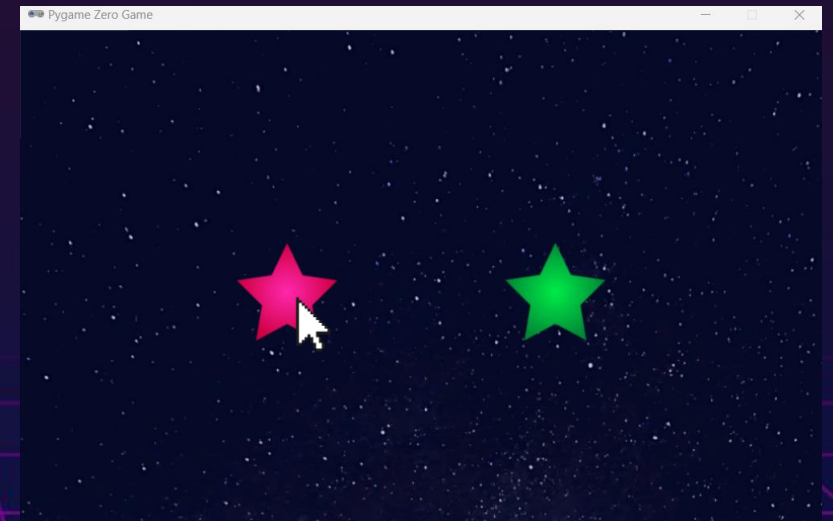
5.3a Handle Mouse Clicks

- Actors in Pygame Zero are complex data structures that group together **variables** and **functions**.

- The Pygame Zero docs explain how you can check whether the position of the mouse click is indeed within a star actor <u>here</u>.

- Actors have a function named **collidepoint()** which you must use.

- The example provided should help but you need to adapt it slightly to make it work in your game.

- Find the relevant TODO in the sample and complete the code.

```
1    def on_mouse_down(pos):
2        if alien.collidepoint(pos):
3            print("Eek!")
4        else:
5            print("You missed me!")
```

"Eek!"

Do not forget that our Actors are stars, not aliens.

Are you stuck?
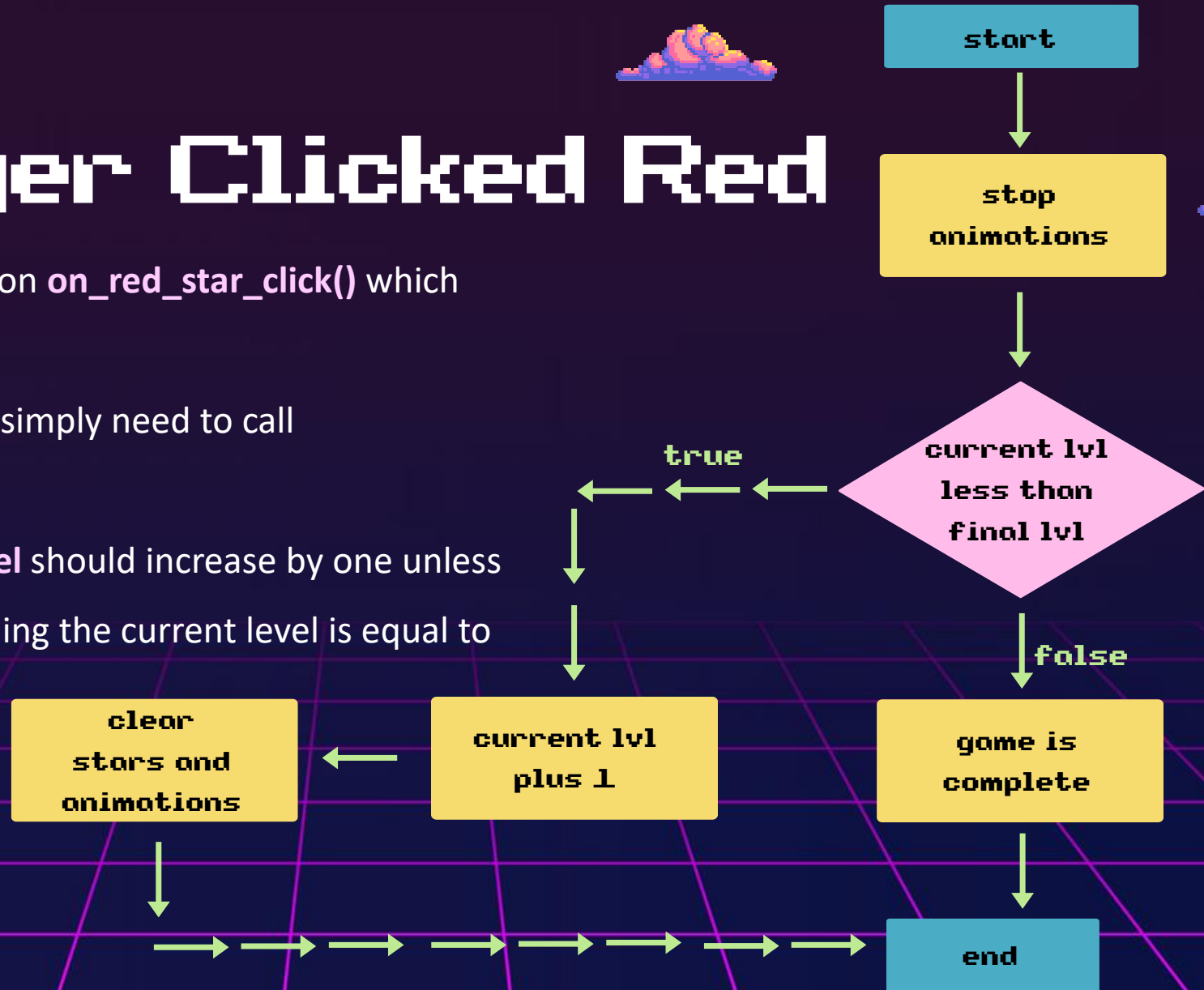
If you are finding it difficult to proceed,

please use a help token and your teacher

will give you more guidance.

5.3a Handle Mouse Clicks

- Actors in Pygame Zero store a **variable** called **image** which stores the name of the image used.

- This variable stores a string e.g., **'red_star'**.

- When a star is clicked on you must check whether the image of the star is indeed the red one.

- If so, then the game proceeds to execute the function called **on_red_star_click()**.

- Otherwise, the game is over!

- Find the relevant TODO in the sample and complete the code.

`alien.image`

Do not forget that our Actors are stars, not aliens.

Are you stuck?

If you are finding it difficult to proceed,

please use a help token and your teacher

will give you more guidance.

5.3b Player Clicked Red

- Define a user-defined function **on_red_star_click()** which

  stops the animations.

- To stop the animations, you simply need to call

  stop_animations() function.

- The value of the **current level** should increase by one unless

 the game is complete, meaning the current level is equal to

  the **final level**.

```
start
  |
  v
stop
animations
  |
  v
current lvl
less than    --true-->    current lvl
final lvl                   plus 1    -->   clear
  |                                         stars and
  |false                                    animations
  v                                            |
game is                                        v
complete                                      end
  |
  v
end
```

5.3b Player Clicked Red

- Are you up for a small detour to get more marks and reach the highest skill level?

- Click here to learn about **assignment operators** in Python. You can apply this skill to increase the **current level** by one.

- Click here to learn about the **clear()** list function to clear stars and animations.

- You **do not** have to learn new things to complete the task. You can **skip this detour slide** if you want to stick to your Python skill level.

5.3b Player Clicked Red

- When completing the task, keep in mind that this function needs access to **global variables**.

- A lot of the variables we created are at the top of our program outside functions. For our functions to amend these we must use the **global keyword** as we have seen in other functions.

- The **first line of code inside the function** should be:

```
global current_level, stars, animations, game_complete
```

- You know more than enough Python to complete this task.

- Find the relevant TODO in the sample and complete the code.
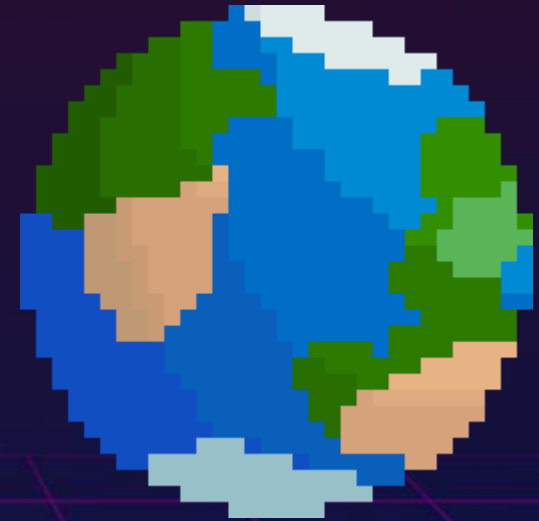
You got this!

Are you stuck?

If you are finding it difficult to proceed,

please use a help token and your teacher

will give you more guidance.
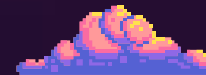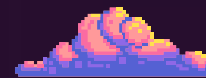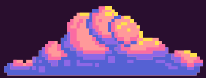
5.3c Stop Animations

- Define a user-defined function **stop_animations()** which stops the animations.

- Like Actors, Animations are also complex data structures.

- Animations in Pygame Zero have a Boolean variable **running** which stores whether the animation is being played.

- If the animation is being played, then you should make it stop by using the **stop()** function.

- Find the relevant TODO in the sample and complete the code.

```
# get info about animation
is_played = my_animation.running

# stop animation
my_animation.stop()
```
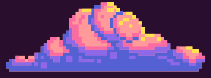
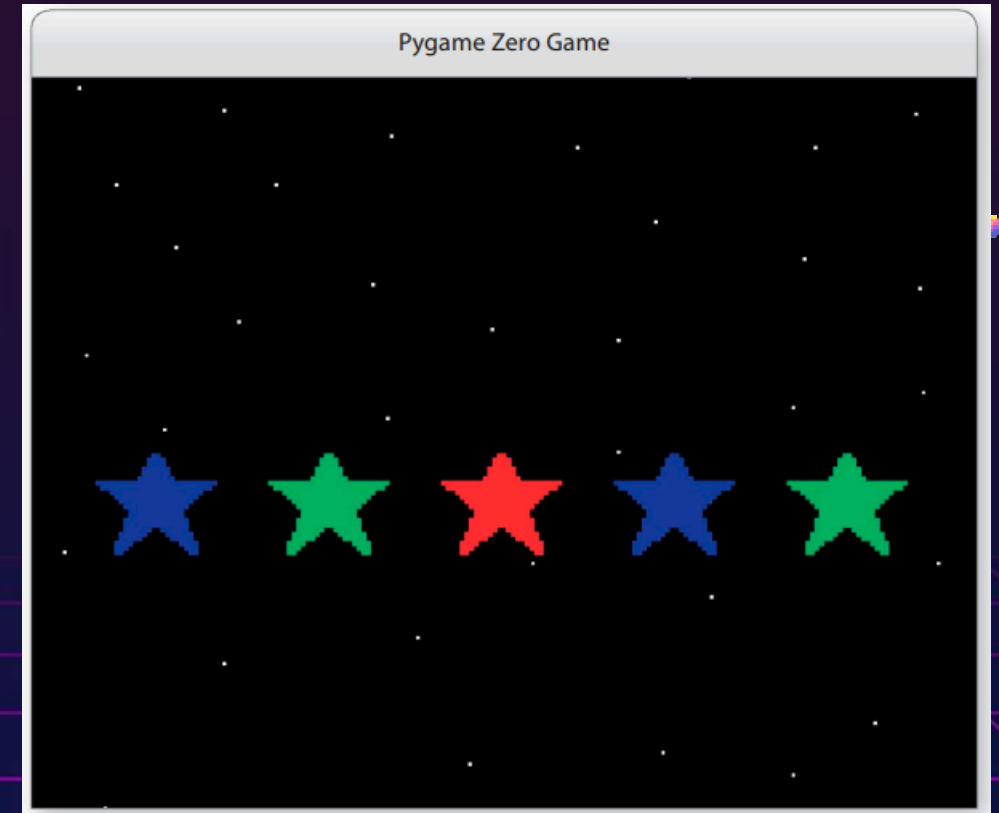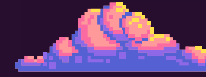**You are reaching the end now.**

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 your code.

- Run your code and try to play the game to see if it works.

- It is also a good idea to review the assessment criteria and make sure that your code is easy to correct by a teacher.

Pygame Zero Game