Pygame - Sprites

Imagine you're making a video game, and you want to create characters, enemies, or objects that can move around the screen. In Pygame, we use something called "sprites" to make this happen.

Sprites in Pygame are like the actors or objects in your game. You load images, create sprite objects from them, and then control their movements and interactions to bring your game to life. It's a fundamental concept in game development that helps you create dynamic and interactive games.

Gathering sprites

Before we can start drawing sprites in Pygame, we need to find/create the images we want to use as sprites. For this tutorial, you will need three sprites. Ideally, the images should be in the PNG file format. This file format allows for transparency, which is great for characters and objects in video games.

If you cannot think of what sprites you want to use, you can find some in the tutorial's folder on GitHub.

Make sure that the images are saved in your project folder. It is a good idea to organise them into a special folder.

Loading images into Pygame

To load an image into your game, you need to load it using the image's local path. So for example, if you project folder structure looks like this:

```
├─── main.py
├─── sprites
        ├─── spaceship.png
        ├─── balloon.png
        ├─── logo.png
```

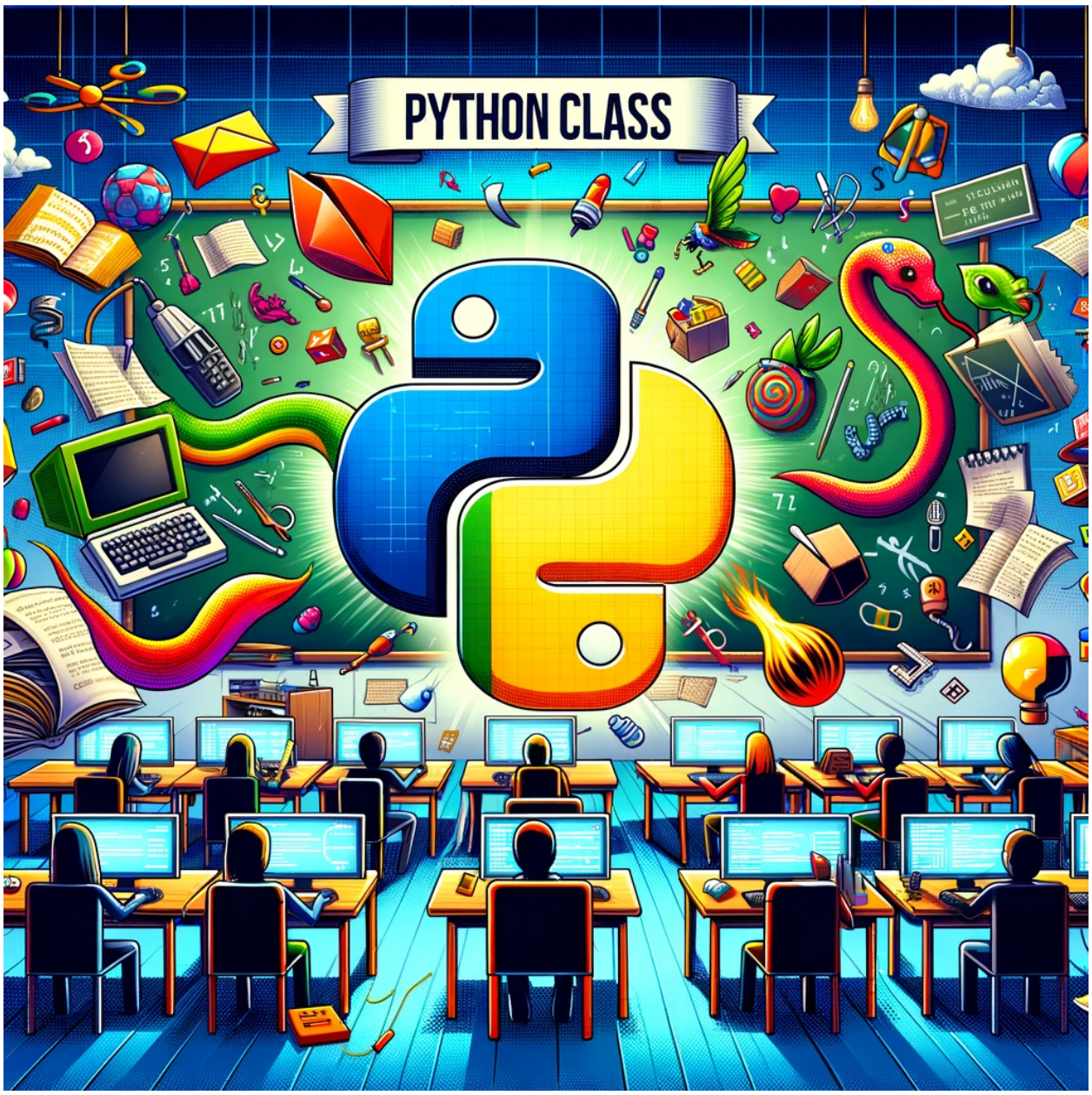
The local path to the logo image would be:

```
./sprites/logo.png
```

To load the image in your game, write the following line of code:

```
logo_image = pygame.image.load("./sprites/logo.png")
```

Classes



From here on out, when we want to add a player, actor, enemy or any object to our game, we will use classes. A class in Python is like a blueprint. It defines how something should be structured and what it should do. Imagine you're creating a recipe for a type of cake; that's your class. We can then make objects from these blueprints called instances.

Pygame provides a helpful class called Sprite. This is a built-in class in Pygame that helps in managing game objects. It's like a general recipe for creating any sprite (game character or object) in your game. We will use this as a base class for our own classes. This essentially just means that we want to have the features and

functionalities of the Sprite class and then add to it with our own code. You can make a class like this:

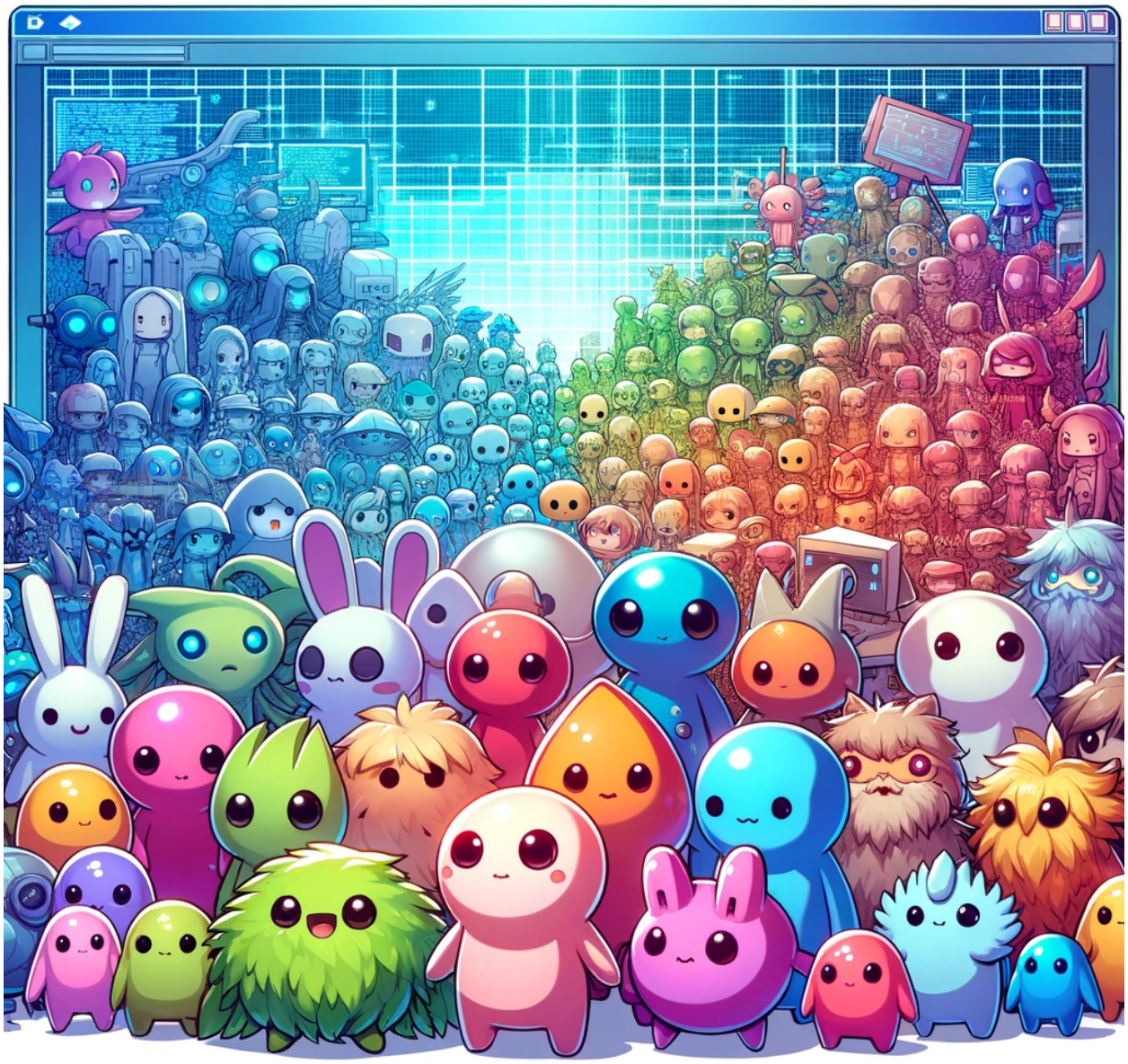
```
class Example(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()

    def update(self):
        print("hello, world!")
```

You can see that the Example class inherits from the pygame.sprite.Sprite class. The __init__ function is what will get called when we create an instance of this class. Within this function, we will call any setup code the class needs. The super().__init__ function call makes sure that the base classes init function is also called. Most of our classes will also have an update function. This will be called every frame. This is where we will call most of our gameplay related to this class.

To make an instance of a class, use the line:

```
example = Example()
```

Sprite Groups



A sprite group is a collection of sprites (characters or objects) that you can manage together. This is how we will be drawing our sprites to the screen. Sprite groups allow us to easily draw a large number of sprites at the same time. This avoids us having to draw sprites one by one, which would be a lot of work and would mean a lot of repeating code.

You can create a sprite group and add sprites to it like this:

```
example_group = pygame.sprite.Group()
```

```
example_group.add(spaceship)
```

You can then draw and update all sprites in the group:

```
example_group.update()
example_group.draw(screen)
```

Challenge

1. Create a fresh Pygame project and set up the basics in main.py. If you want, you can grab the starter code from the starter-project folder in the Github repository
2. Create two new classes, both of which take an image as an init parameter
3. Create an instance of each class
4. Create a sprite group and add both instances to it
5. Then draw all sprites in the sprite group in every frame