

## Practical Session 3: Dynamic backgrounds in Pygame

### Overview

In this session you will be looking at different ways of making backgrounds more dynamic in nature, rather than a plain coloured background or a static image. This will be done by introducing the concept of parallax scrolling and scrolling a background that is larger than the screen. The homework tasks for this week will involve you considering the coursework and what game demo you would like to make for it.

You will need to download the “Scratch/Pygame Resources” and “Pygame Scripts” from the class materials wiki page.

### Task 3.1

Your first task is to create a parallax scrolling starfield that could be used for a spaceship game. A parallax scrolling background is a scrolling background that is split into layers, with each of the layers moving at different speeds. This gives the impression of depth as it appears that the slower moving layers are further away than the faster moving layers. The simplest demonstration of this is to use a scrolling starfield where there really is only one layer, but the different speeds of the stars gives the illusion of distance.

The way to achieve the effect of a parallax scrolling starfield with Pygame is quite straight-forward. You need to creating a sprite object for a star and then adding multiple instances of it to a list depending on how many stars you want ... meaning if you wanted more stars, you could just change a single variable (e.g. *numStars*).

The helper script *Star.py* includes a basic star class that needs to be completed. Rather than use a sprite for the star object I have added code to draw a single white pixel at the star's position. Feel free to use a sprite or your own star class.

Using the framework provided for last week's practical, or one of your own creation, follow these steps (adjust the instructions to fit the class objects you are using):

1. Import the Star class from the Star module – look at how this was achieved with the Sprite class
2. Initialise global variables
  - 2.1. *numStars* – set to the number of stars you want (100 works quite well)
  - 2.2. *stars = []* – this creates an empty list to add the star objects to
3. Initialise the stars
  - 3.1. For each star 'for *i* in range(0, *numStars*-1)'
    - 3.1.1. Append a new star to the stars list '*stars.append(Star())*'

- 3.2. Set the star to a random position on the screen
- 3.3. Set a random speed for the star (from 0.0 to 2.0 should do)
4. Tick all the stars
  - 4.1. For each star
    - 4.1.1. `stars[i].tick()`
    - 4.1.2. If the y position is at the bottom of the screen
      - 4.1.2.1. Move to the top of the screen, but at a random x position (this makes it look like a new star, not the same star coming back into the screen)
      - 4.1.2.2. Give the star a random speed, again between 0.0 and 2.0 should do
5. Render all the stars
  - 5.1. For each star
    - 5.1.1. Render the star, passing `screen` as the argument

Once you have this code implemented, have tested it out, and are happy with it, adjust the number of stars to give the best effect. If you get stuck, ask a tutor for help.

### **Task 3.2**

What happens when you have a background larger than your screen, and you want it to scroll from side to side as your character moves? Rather than set the background up as an actual background, you could set it up as a large sprite, only part of which is being shown on screen at any one time. This is fairly trivial in Pygame, so I will leave this task completely up to you.

Using the `large_background.png` file, create a large sprite so that pressing the left or right cursor key will scroll the “background” appropriately. You should also make sure that the screen is always filled with the background i.e. you can’t scroll past the left or right edges. You can ignore the other background files, these were created as a workaround for Scratch.

### **Homework Task 3.1**

For next week you should have an idea of the game you wish to develop for the coursework. You may already have an idea, but you need to get it down on paper in such a way that a tutor will be able to read the description and have a fair idea of whether it will be possible to complete within the given time, and conversely, if there is going to be enough content in it. As you will be revising the game idea as you progress in the actual game’s development, don’t be worried if there are a few things that you don’t quite have sorted out for next week.

**Homework Task 3.2**

A few suggestions for how the above tasks could be extended are given below, at varying degrees of difficulty. This is just to give you an idea of how games and demos can evolve, and you may have different ideas of your own. For next week, have a go at expanding on the tasks you have been given, and use your imagination!

1. The parallax scrolling starfield could be added to the spaceship demo from last week, rather than having the static starfield background.
2. A larger area of space could be simulated by the spaceship staying in the centre of the screen until it got to the edge of the area, at which point it would be moving towards the edge of the screen. This may be a tricky concept to grasp, but basically you would have a play area larger than the screen itself, and scrolling in both the x and y axes.
3. The movement of the stars could be changed so that they are stationary while the spaceship is, and while the spaceship is moving, the stars move in the opposite direction.
4. You could add in a character to the side-scrolling demo so he moves appropriately through the level.
5. The side-scrolling demo could be extended with a much larger background that needs scrolling in both horizontal and vertical directions.
6. A similar background to the one used in the side-scrolling demo could be created with separate sprite objects for clouds, hills, trees etc. Each of these objects could scroll at different speeds depending on the distance they are to be perceived at, essentially combining the side-scrolling and parallax scrolling demos.