

## Practical Session 3: Dynamic backgrounds in Scratch

### 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 “Scratch Projects” 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 a parallax scrolling starfield would be achieved with most programming languages is actually quite difficult in Scratch; it can be done, but performs really slowly. This would be creating a sprite object for a star and then adding multiple instances of it to a list or array depending on how many stars you want ... meaning if you wanted more stars, you could just change a single variable (e.g. *numStars*).

To achieve this effect in Scratch, you need to create a star sprite, add the necessary code to it, and then duplicate this sprite for how many stars you want in your starfield. While this may sound similar to the above method, it's not as easy to change the number of stars at runtime (say from one level to the next). To adjust the number of stars, you would need to either delete some of the sprites, or create more duplicates.

First of all you need to create a star sprite. This is a simple case of choosing to paint a new sprite, drawing a single white pixel. You could create a fancier looking star if you want, but a single white pixel gives quite a nice effect.

Next, add the appropriate code to the star, following this algorithm:

1. When green flag clicked
2. Go to a random position on the screen

3. Set a random speed for the star (you will need a variable, random range should be about -3 to -1)
4. Repeat forever
  - 4.1. Change the y position by the current speed
  - 4.2. If the y position is at the bottom of the screen
    - 4.2.1. 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)
    - 4.2.2. Set a random speed, again this should be between -3 and -1

Once you have this code implemented and are happy with it (test it out with the one star to make sure it looks ok), create duplicates of this star and run the program to see the results. With only 10 stars, you should be able to see a nice parallax scrolling starfield effect.

### **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. Unfortunately, in Scratch, you can't have a sprite that has a width or height greater than that of the screen (480 \* 360), and if you try to load such an image in you will see that it is scaled down so the whole thing can fit on the screen at the same time.

There is a way around this though, and that is to split your large background into smaller pieces and move them around the screen. The second problem is that in Scratch, no sprite can go completely off the screen. To get around this, you have to write some clever code that moves the pieces around and hides or shows the pieces as appropriate. The pieces also have to have overlapping sections so that when one is hidden but still partially on the screen, there is no blank area left.

The code for achieving this isn't particularly complicated, but it is fairly tricky to get it right. For this reason, rather than asking you create a Scratch project from the beginning, you will use the partially completed *SideScroll\_Scratch.sb* project that has been built to scroll around the *large\_background.png* file which was split into parts A-F. Sprites have been created so that parts A, B, E and F are working correctly, and you must fill in the blanks with parts C and D.

Take your time analysing the code in the project (there is code in each sprite, plus the stage). If you get really stuck, ask a tutor for help.

### **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.