

Parallax scrolling – nền chuyển động cho game

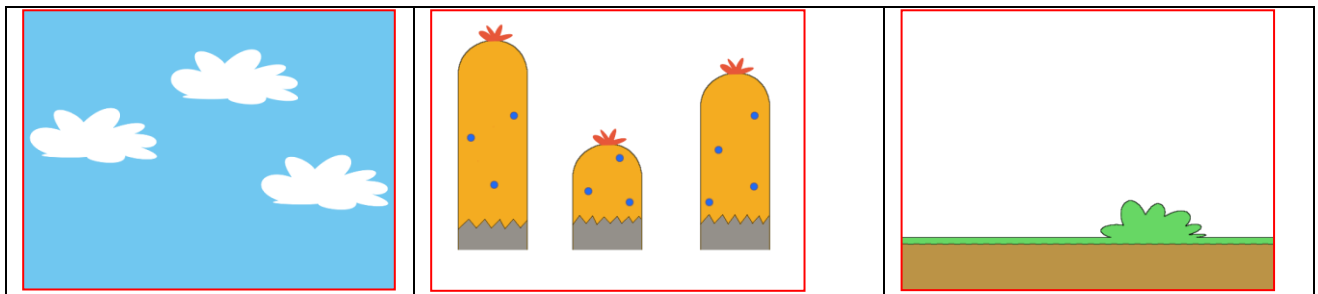
Parallax scrolling

1982 Irem giới thiệu kỹ thuật Parallax scrolling trong game Moon Patrol. Nó là kỹ thuật tạo ra ảo giác của chuyển động. Nó không ngừng phát triển và đến nay nó còn được xây dựng trên web với sự hỗ trợ của HTML5 và CSS3.

Trong game và các ứng dụng 2D, kỹ thuật parallax scrolling dùng nhiều lớp đặt chồng lên nhau và từ từ duy chuyển qua vùng nhìn thấy. Mỗi lớp sẽ di chuyển với vận tốc khác nhau tạo thành hiệu ứng chiều sâu.

Ví dụ:

Với 3 tấm hình bên dưới.



Các bước tiến hành sẽ như sau:

1. Hình đám mây sẽ được kéo từ từ qua vùng hiển thị trên thiết bị.



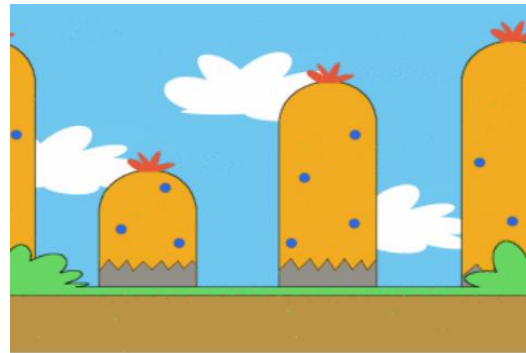
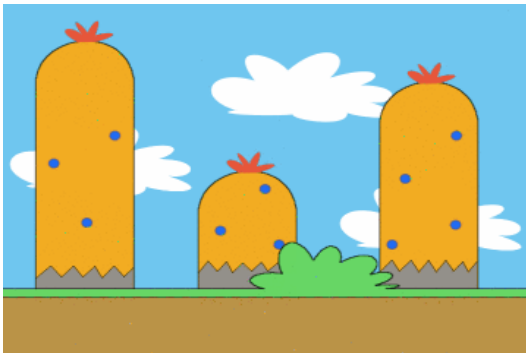
2. Khi kéo như thế sẽ nếu hình vượt khỏi khung nhìn thì sẽ bị mất (nhìn hình: sẽ xuất hiện vùng trắng phía sau). Khi đó ta lại vẽ một hình tương tự vậy nối đuôi theo sau. Và cứ thế ta đã tạo ra chuyển động cho 1 lớp nền.



3. Tiếp theo ta tạo ra các lớp khác và cũng làm tương tự khi kéo. Mấu chốt là các lớp có tốc độ kéo khác nhau sẽ tạo thành hiệu ứng nền chuyển động.



Khi đó kết quả là tuy các lớp cứ chạy lặp đi lặp lại nhưng do có nhiều lớp và tốc độ chạy khác nhau nên thứ nhất ta thấy như có chiều sâu, thứ hai ta không thấy rõ sự lặp lại.



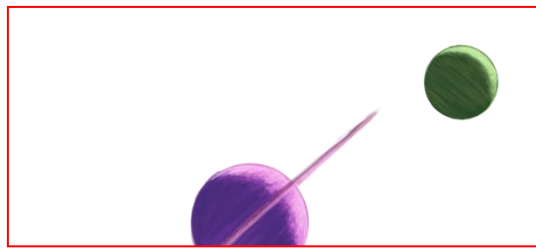
Thực hiện:

Phát triển tiếp từ bài 1.

1. Mở file Manifest, trong the activity thêm thuộc tính screenorientation chỉ là landscape để chương trình chuyển thành chế độ nằm ngang.

```
<activity
    android:name=".CautruccobanActivity"
    android:label="@string/app_name"
    android:screenOrientation="landscape">
```

2. Tạo thư mục drawable trong thư mục res. Chép 2 hình bên dưới vào (hoặc 2 hình của bạn thiết kế)



3. Tạo 1 class mới để quản lý việc vẽ nền có tên “ParallaxBackground”. Trong class vừa tạo khai báo các biến:

```
private int toadonen1_X = 0;  
private int toadonen2_X = 0;  
private Bitmap hinhnen1;  
private Bitmap hinhnen2;
```

4. Xây tiếp hàm tạo nhận vào resource và khởi tạo 2 hinhnen ánh xạ đến 2 tấm hình.

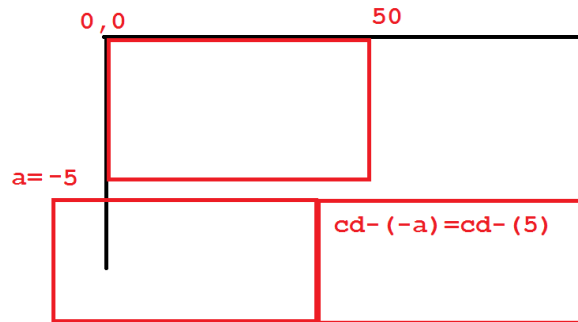
```
public ParallaxBackground(Resources c)  
{  
    hinhnen1 = BitmapFactory.decodeResource(c, R.drawable.background_a);  
    hinhnen2 = BitmapFactory.decodeResource(c, R.drawable.background_b);  
}
```

5. Xây dựng tiếp hàm tên doDrawRunning, hàm này dùng để vẽ 2 nền chạy. Mỗi nền khi chạy sẽ vẽ nối đuôi thêm một hình nữa để tạo Parallax scrolling.

```
public void doDrawRunning(Canvas canvas) {  
    //giam toa do de dich chuyen cho nen1  
    toadonen1_X = toadonen1_X - 1;  
  
    //giam toa do de dich chuyen cho nen2  
    toadonen2_X = toadonen2_X - 4;  
  
    // tinh do lech cho hinh 2 (xem hình minh hoa)  
    int toadonen1_phu_X = hinhnen1.getWidth() - (-toadonen1_X);  
  
    //da di chuyen het thi quay lai tu dau  
    if (toadonen1_phu_X <= 0) {  
        toadonen1_X = 0;  
        // chi can ve 1 tam  
        canvas.drawBitmap(hinhnen1, 0, 0, null);  
    } else {  
        // ve 1 tam lech va tam 2 noi duoi theo  
        canvas.drawBitmap(hinhnen1, toadonen1_X, 0, null);  
        canvas.drawBitmap(hinhnen1, toadonen1_phu_X, 0, null);  
    }  
  
    int toadonen2_phu_X = hinhnen2.getWidth() - (-toadonen2_X);  
  
    if (toadonen2_phu_X <= 0) {  
        toadonen2_X = 0;  
        canvas.drawBitmap(hinhnen2, toadonen2_X, 0, null);  
    } else {  
        canvas.drawBitmap(hinhnen2, toadonen2_X, 0, null);  
        canvas.drawBitmap(hinhnen2, toadonen2_phu_X, 0, null);  
    }  
}
```

```
}
}
```

Trong hàm khi tính toán tọa độ cho hình 2(hình nối đuôi) xem hình để hiểu rõ hơn.



6. Quay về file “GamePanel.java” khai báo biến toàn cục ParallaxBackground.

```
ParallaxBackground background; //biến hình nền chuyển động
```

7. Trong hàm tạo khởi tạo đối tượng “background” và truyền vào Resources.

```
background=new ParallaxBackground(this.getResources());
```

8. Trong hàm onDraw sau khi vẽ nền đen cho canvas ta gọi hàm doDrawRunning của background để vẽ nền.

```
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    canvas.drawColor(Color.BLACK);
    background.doDrawRunning(canvas);

    if(myelement!=null)
        myelement.doDraw(canvas);
}
```

9. Chạy chương trình và xem sự chuyển động của nền.

