Parallax scrolling – nền chuyển động cho game

# Parallax scrolling

1982 Irem giới thiệu kĩ thuật Parallax scrolling trong game Moon Patrol. Nó là là kĩ 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, kĩ 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 du:

Với 3 tấm hình bên dưới.

|  |  |  |
| --- | --- | --- |
| [File:Sky back layer.png](http://upload.wikimedia.org/wikipedia/commons/9/9f/Sky_back_layer.png) | [File:Vegetation (middle layer).png](http://upload.wikimedia.org/wikipedia/commons/8/8b/Vegetation_(middle_layer).png) | [File:Ground (front layer).png](http://upload.wikimedia.org/wikipedia/commons/b/be/Ground_(front_layer).png) |

Các bước tiến hành sẽ như sau:

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



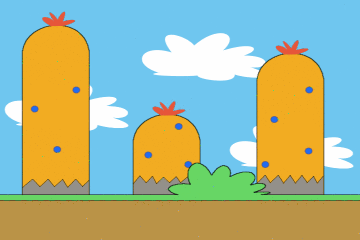
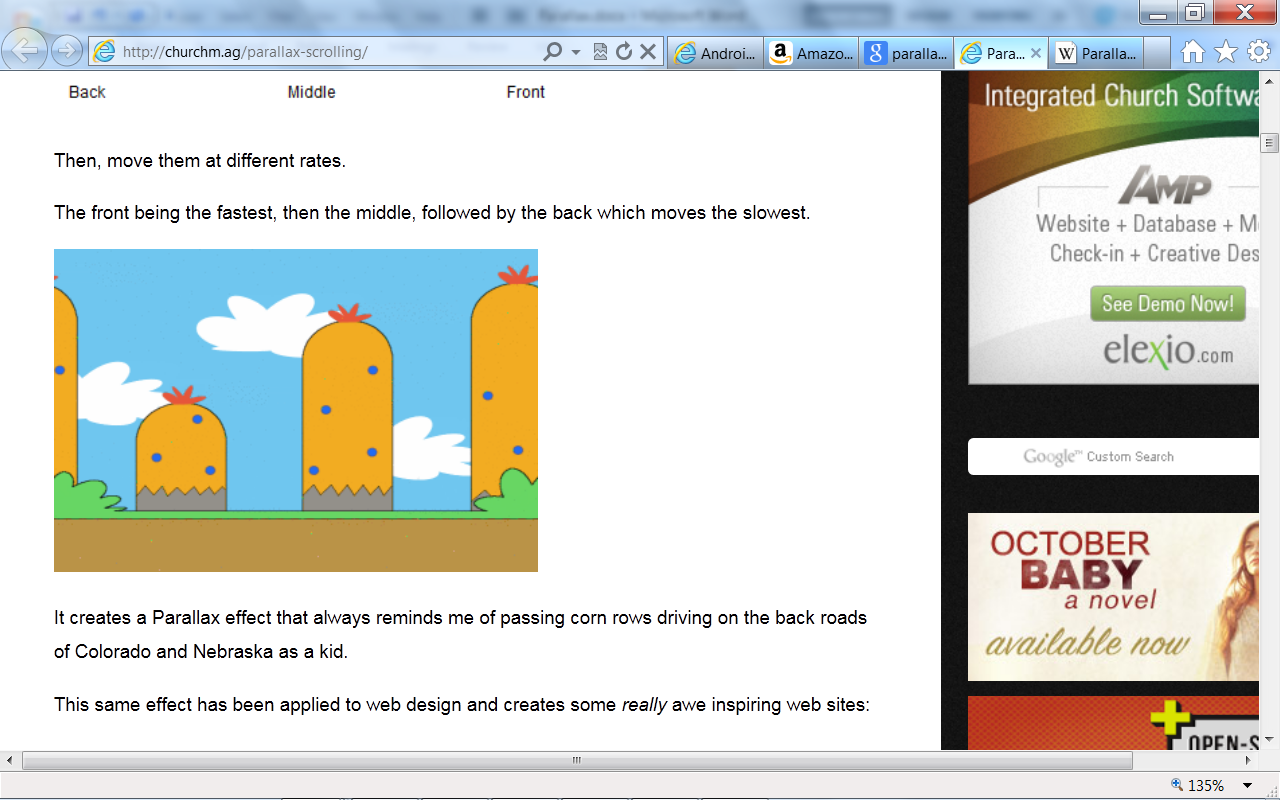
1. Khi kéo như thế sẽ nếu hình vượt khỏi khung nhìn thì sẽ bị mất (nhìn hình: sẽ xuất hiện vùng trắng phía sau). Khi đó ta lại vẽ một hình tương tự vậy nối đuôi theo sau. Và cứ thế ta đã tạo ra chuyển động cho 1 lớp nền.



1. 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.

[](http://upload.wikimedia.org/wikipedia/commons/3/34/Parallax_scrolling_example_scene.gif) 

# 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"*>

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

1. 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;

1. 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*);

}

1. Xây dựng tiếp hàm ten 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 hinh 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** {

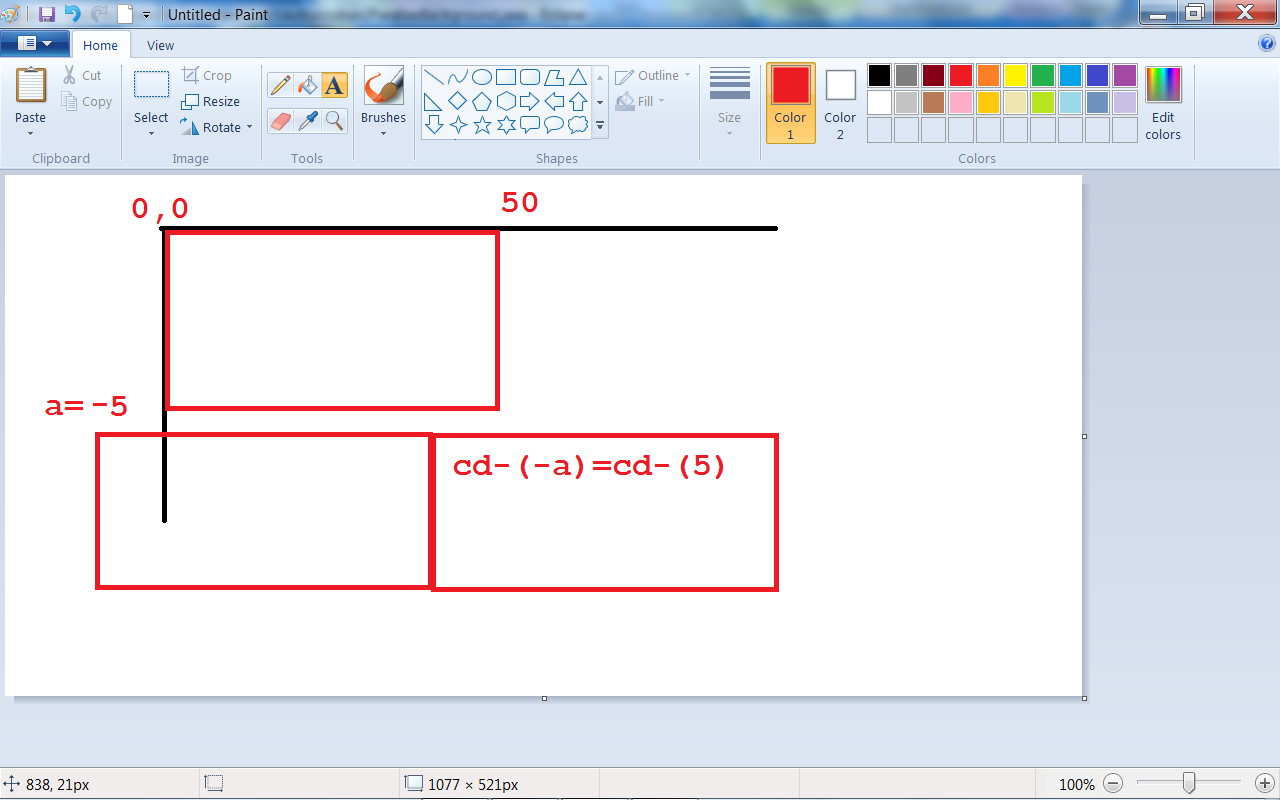
canvas.drawBitmap(hinhnen2, toadonen2\_X, 0, **null**);

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.



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

ParallaxBackground background; //bien hinh nen chuyen dong

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

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

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

**protected** **void** onDraw(Canvas canvas) {

**super**.onDraw(canvas);

canvas.drawColor(Color.*BLACK*);

background.doDrawRunning(canvas);

**if**(myelement!=**null**)

myelement.doDraw(canvas);

}

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

