**Lê Đức Nam – 52047**

**1.Cách để xác định màu sắc**

Nếu bạn muốn làm việc với màu sắc, nó thường được coi là một thực hành tốt nhất để bắt đầu bằng cách thêm vào một file colors.xml vào thư mục res/value. Sau đó, bạn có thể thêm một yếu tố màu sắc cho mỗi màu sắc mà bạn muốn xác định như trong ví dụ duoi day. Trong các yếu tố màu sắc, bạn có thể sử dụng các thuộc tính tên để chỉ định tên cho màu sắc. Trong hình này, ví dụ, tập tin colors.xml tạo ra màu sắc có tên là tiểu học, trung học, đại học, nền, bóng tối và ánh sáng. Cùng với những màu sắc mang một màu sắc mà bạn có thể sử dụng cho các ứng dụng của bạn.

**File colors.xml thư muc res/values**

*<?xml version="1.0" encoding="utf-8"?>*

*<resources>*

*<color name="primary">#141315</color>*

*<color name="secondary">#736C6B</color>*

*<color name-"tertiary">#DDEOCE</color>*

*<color name="background" >#A6D39D</color>*

*<color name-"dark”>#000000</color>*

*<color name="light">#FFFFFF</color>*

*</resources>*

Trong cơ thể của một yếu tố màu sắc, bạn có thể sử dụng một hệ thập lục phân, hoặc hex, giá trị để xác định một giá trị RGB cho màu sắc. Điều này làm việc rất giống với việc sử dụng các giá trị hex để xác định màu sắc RGB với HTML và CSS. Nếu bạn không quen với cách làm việc này, bạn có thể tìm kiếm trên Internet để tìm hiểu thêm về nó.

**2. Áp dụng màu sắc trong Android**

Ví dụ thứ hai cho thấy ba cách để áp dụng một màu cho tài sản của một widget Format. Đầu tiên, bạn có thể sử dụng một giá trị hex để chỉ định một màu sắc. Thứ hai, bạn có thể sử dụng một tên cho một màu sắc tùy chỉnh. Thứ ba, bạn có thể sử dụng một tên cho một màu sắc được xây dựng trong **Android**. Những kỹ thuật này, sử dụng một màu sắc tùy chỉnh có thể làm cho nó dễ dàng hơn để áp dụng màu sắc một cách nhất quán và nó có thể làm cho ứng dụng của bạn dễ dàng hơn để duy trì nếu bạn quyết định để thay đổi màu sắc sau này.

Áp dụng vào các widget sử dụng giá trị hexadecimal

*android:textColor-"#141315"*

Sử dụng tên từ file colors.xml

*android: textColor- " Ocolor/primary"*

Sử dụng tên của một màu **Android** color

*android: textColor- " @android: color/darker\_gray "*

Ví dụ thứ ba cho thấy làm thế nào để sử dụng một styles để áp dụng một màu. Để làm điều đó, bạn sử dụng một phần tử mục trong một styles để thiết lập màu sắc cho một tài sản. Khi bạn sử dụng kỹ thuật này, bạn có thể sử dụng các giá trị hex hoặc kiểu được đặt tên, mặc dù nó thường thích hợp hơn để sử dụng kiểu được đặt tên bất cứ khi nào có thể.

Áp dụng vào các style

*<style name-"Textview" parent-”@android:style/Widget.Textview">*

*<item name-"android:textColor">®color/primary</item>*

*</style>*

Ví dụ thứ tư cho thấy làm thế nào để áp dụng một màu bằng cách ghi đè các màu sắc mặc định của một chủ đề. Để làm điều đó, bạn sử dụng một phần tử mục trong một themes để ghi đè lên một trong những thuộc tính màu sắc cho chủ đề. Khi bạn sử dụng kỹ thuật này, bạn phải sử dụng một styles đặt tên, không phải là một giá trị hex.

**Áp dụng vào  theme**

*<style name-"AppTheme" parent="AppBaseTheme">*

*<!— Set new colors for the theme —>*

*<item name-"android:windowBackground">9color/background</item> <item name-"android: textColorPrimary">@color/primary</item>*

*<item name-"android:textColorSecondary">(?color/secondary</item> </style>*

File  colors.xml cho  **Android** 4.2 (AP117)

*\android-sdks\platforms\android-17\data\res\values\colors.xml*

Nếu bạn muốn tìm hiểu thêm về các màu sắc mà **Android** sử dụng trong các themes tích hợp của nó, bạn có thể xem các attrs.xmlvàcolors.xml tập tin định nghĩa các thuộc tính mặc định và màu sắc cho **Android**. Để tìm những tập tin này, bạn có thể tìm trong các data\res\value thư mục cho các **API Android** trên máy tính của bạn.

**Xây dựng lớp chướng ngại phía dưới.**

public class BotBorder extends GameObject{  
 private Bitmap image;  
 public BotBorder(Bitmap res, int x, int y)  
 {  
 height = 200;  
 width = 20;  
 this.x = x;  
 this.y = y;  
 dx = GamePanel.MOVESPEED;  
 image = Bitmap.createBitmap(res, 0, 0, width, height);  
  
 }  
 public void update()  
 {  
 x +=dx;  
  
 }  
 public void draw(Canvas canvas)  
 {  
 canvas.drawBitmap(image, x, y, null);  
}}