

# Vẽ nhiều hình tại nhiều vị trí chạm

## Phát triển tiếp bài 3.

Từ bài 3 ta phát triển tiếp để ta chạm vào nhiều lần sẽ vẽ ra nhiều hình (các hình sẽ được giữ lại mỗi lần chạm).

1. Đầu tiên ta phải xây dựng một lớp mới để đóng gói việc vẽ lại. Class này gồm tọa độ mX,mY, bitmap và hàm doDraw để tự vẽ. Tạo 1 class mới tên “element.java” và viết lệnh như sau:

```
public class element {
    private int mX;
    private int mY;
    private Bitmap mybitmap;

    public element (Resources res,int x,int y)
    {
        mybitmap=BitmapFactory.decodeResource(res,R.drawable.ic_launcher);
        mX= x-mybitmap.getWidth()/2;
        mY=y-mybitmap.getHeight()/2;
    }

    public void doDraw(Canvas canvas)
    {
        canvas.drawBitmap(mybitmap,mX,mY,null);
    }
}
```

2. Hàm Draw2d.java cũng cần phải thay đổi lại. Hàm này chỉ còn quản lý việc vẽ chứ không tự vẽ, nhiệm vụ vẽ sẽ giao cho class “element”. Đầu tiên bỏ biến mX, mY, mybitmap và thay bằng ArrayList như sau:

```
private ArrayList<element> taphop=new ArrayList<element>();
```

3. Sửa hàm onTouchEvent lại để mỗi lần chạm sẽ tạo ra một phần tử (element) mới , và đưa nó vào tập hợp:

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    // TODO Auto-generated method stub

    element e=new element(getResources(),(int)event.getX(),(int)event.getY());
    taphop.add(e);

    return super.onTouchEvent(event);
}
```

4. Sửa lại hàm onDraw, bỏ việc vẽ mà dùng for để duyệt tất cả các phần tử rồi gọi hàm vẽ để vẽ lại.

```
protected void doDraw(Canvas canvas) {
    // TODO Auto-generated method stub
    super.onDraw(canvas);
    canvas.drawColor(Color.BLACK);

    for(element phantu : taphop)
        phantu.doDraw(canvas);
}
```

```
}
```

5. Chạy chương trình và thấy rằng nếu chạm chậm chậm sẽ chạy tốt nhưng khi chạm nhanh và liên tục sẽ phát sinh ra lỗi. Lỗi này là do trong khi Viewthread gọi hàm doDraw() và chạy nó trong vòng lặp thì ta lại chạy việc thêm 1 phần tử vào ArrayList làm thay đổi ArrayList. Do đó hàm doDraw và hàm onTouchEvent đụng nhau chỗ thẳng thì đang dùng for để duyệt list thì thẳng onTouchEvent lại thêm phần tử mới vào list. Sửa lỗi này bằng cách dùng hàm synchronized() để đồng bộ hóa việc đọc và ghi trên cùng đối tượng. Khi đó hàm add() sẽ đợi đến khi vẽ xong và đối tượng “taphop” được giải phóng và sẵn sàng được sửa. Ta sẽ sửa 2 hàm onDraw và onTouchEvent lại như sau:

```
protected void doDraw(Canvas canvas) {  
    // TODO Auto-generated method stub  
    super.onDraw(canvas);  
  
    canvas.drawColor(Color.BLACK);  
  
    synchronized(taphop)  
    {  
        for(element phantu : taphop)  
            phantu.doDraw(canvas);  
    }  
}
```

```
@Override  
public boolean onTouchEvent(MotionEvent event) {  
    // TODO Auto-generated method stub  
    synchronized(taphop)  
    {  
        element e=new element(getResources(),(int)event.getX(),(int)event.getY());  
        taphop.add(e);  
    }  
    return super.onTouchEvent(event);  
}
```

6. Chạy và kiểm tra kết quả, chạm liên tục nhiều lần ở nhiều vị trí khác nhau để thấy không phát sinh lỗi.

