OTRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM

KHOA CÔNG NGHỆ THÔNG TIN

---\*\*\*---



BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN “Phát triển ứng dụng mã nguồn mở”

*Đề tài:*

GAME SODOKU

*Người hướng dẫn: VŨ ĐÌNH TRUNG*

*Sinh viên thực hiện: DƯƠNG TRANG NGUYÊN*

*Hải phòng, tháng 3 năm 2016*

TRƯỜNG ĐẠI HỌC HÀNG HẢI

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN LẬP TRÌNH ANDROID

---\*\*\*---

1. Tên đề tài

BÀI TẬP LỚN

HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG MÃ NGUỒN MỞ

Ứng dụng Canvas trong lập trình game sodoku

2. Mục đích

* Tạo ra được ứng dụng game sodoku chơi được trên thiết bị di động

*Hải Phòng, tháng 4 năm 2017*

NGƯỜI HƯỠNG DẪN

VŨ ĐÌNH TRUNG

# Lời mở đầu:

**Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Android cho phép các nhà phát triển tạo ra các ứng dụng di động hấp hẫn tận dụng tất cả một điện thoại cung cấp. Các nhà phát triển viết ứng dụng android dựa trên ngôn ngữ java. Sự ra mắt android vào ngày 5/11/2007 gắn với sự thành lập của liên minh thiết bị cầm tay mã nguồn mở, bao gồm 78 công ty phần cứng, phần mềm và viễn thông nhằm mục đích tạo nên một chuẩn mở cho điện thoại di động trong tương lai.**

# Chương I. TỔNG QUAN

# Canvas là gì

**Canvas** được xem như là một bề mặt (hình dung như tờ giấy, bảng) mà chúng ta có thể vẽ bất cứ thứ gì lên đó. Ví dụ như vẽ một điểm, đường thằng, hình chữ nhật, đường tròn, elip, văn bản, hay thậm chí là một hình ảnh và các hình ảnh phức tạp khác nữa.

# Canvas có thể làm những gì

Trong Android nếu bạn muốn tạo một view với và tự vẽ lại tất cả các thành phần của view đó thì một điều chắc chắn rằng bạn phải sử dụng canvas.

Canvas trong Android có cung cấp cho chúng ta các method để vẽ tất cả các đối tượng như sau:

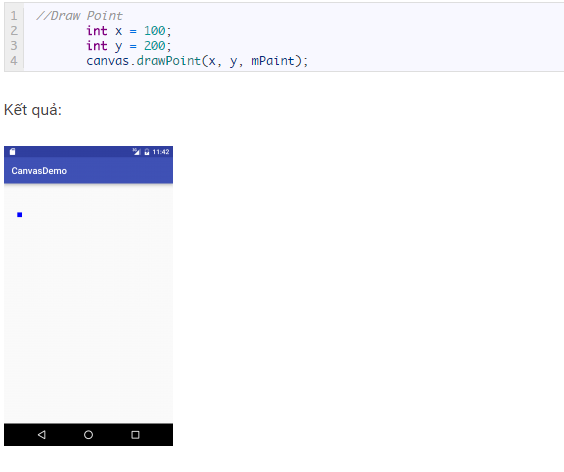
* Các đối tượng hình học cơ bản (point, line, oval, rect..)
* Vẽ hình ảnh (bitmap, drawable)
* Vẽ Path (Tập hợp các điểm)
* Vẽ Text

Những thành phần mà chúng ta vẽ đều được xử lý trong phương thức **onDraw(Canvas canvas)** của class **View**. Và để vẽ những đối tượng, thành phần lên Canvas chúng ta phải cần sử dụng một đối tượng Paint để định dạng Style, color, size cho net vẽ.

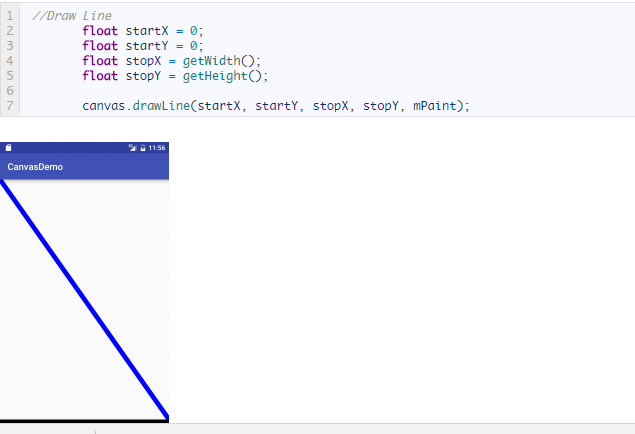
# Vẽ các đối tượng cơ bản bằng canvas

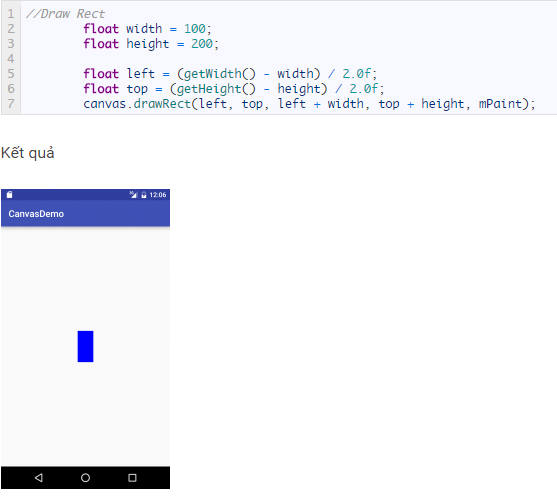
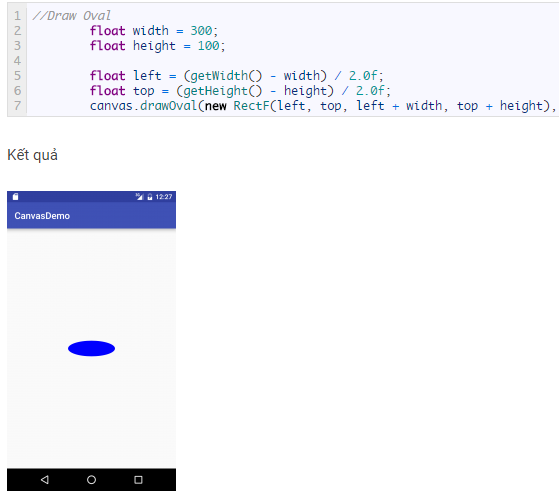
Canvas có cung cấp cho chúng ta rất nhiều các phương thức để chúng ta vẽ các đối tượng hình học bắt đầu bởi phương thức **cavas.draw…**

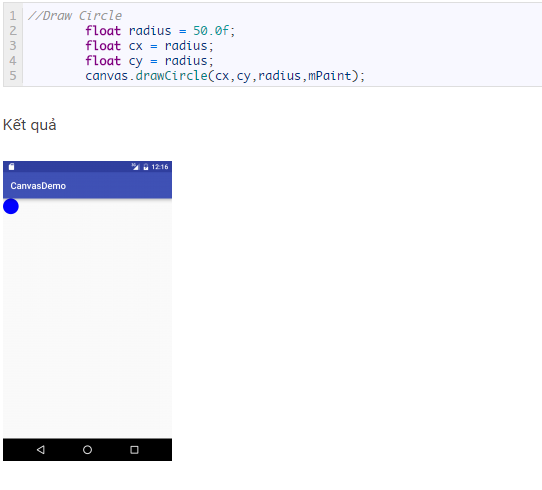
* Draw Point



Draw Line



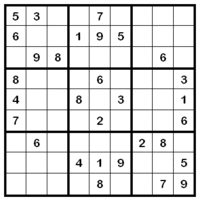
* Draw Rect
* Draw Circle
* Draw Ovals



* Draw Arc

# Chương II. ỨNG DỤNG CANVAS TRONG GAME SUDUKU

# Giới thiệu trò chơi



Một Bảng chơi sodoku

 SODOKU là một loại [trò chơi lôgic](https://vi.wikipedia.org/w/index.php?title=Tr%C3%B2_ch%C6%A1i_l%C3%B4gic&action=edit&redlink=1) và cách chơi là điền số từ 1 đến 9 vào những ô trống sao cho mỗi cột dọc, mỗi hàng ngang, mỗi phân vùng nhỏ (ô 3x3) có đủ các số từ 1 đến 9 mà không được lặp lại.

Bảng câu đố hình vuông, mỗi chiều có 9 ô nhỏ, hợp thành 9 cột, 9 hàng và được chia thành 9 ô lớn 3x3. Một vài ô nhỏ được đánh số, đó là những manh mối duy nhất để bạn tìm lời giải. Tuỳ theo mức độ nhiều hay ít của các manh mối, các câu đố được xếp loại dễ, trung bình, khó hay cực khó. Ngoài ra, còn những bảng như 16x16, 25x25 hay thậm chí 100x100.

VẼ BÀN CHƠI BẰNG CANVAS

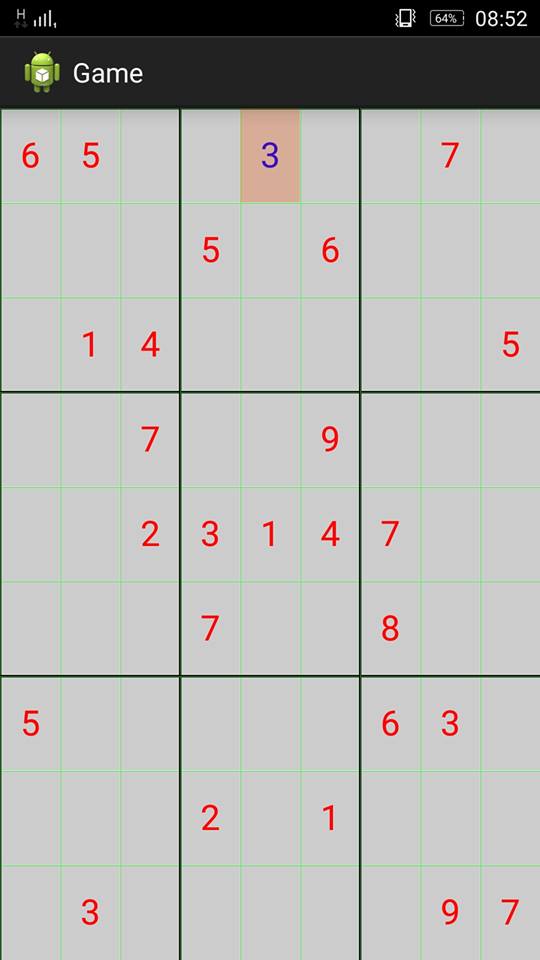
Thêm hàm  onDraw vào PuzzleView.java như sau :

**protected void** onDraw(Canvas canvas) {  
 **new** Paint();  
 Paint background = **new** Paint(); vẽ phân vùng  
 Paint pen1 = **new** Paint();  
 Paint pen2 = **new** Paint(); vẽ bảng kiểu đan lưới  
 Paint pen3 = **new** Paint();  
 background.setColor(-3355444);  
 canvas.drawRect(0.0F, 0.0F, (**float**)**this**.getWidth(), (**float**)**this**.getHeight(), background);  
 pen1.setColor(-16777216);  
 pen2.setColor(-16711936);  
 pen3.setColor(-1);

**int** fm;

**for**(fm = 0; fm < 9; ++fm) {  
 canvas.drawLine(0.0F, (**float**)fm \* **this**.height, (**float**)**this**.getWidth(), (**float**)fm \* **this**.height, pen3);  
 canvas.drawLine(0.0F, (**float**)fm \* **this**.height + 1.0F, (**float**)**this**.getWidth(), (**float**)fm \* **this**.height + 1.0F, pen2);  
 canvas.drawLine((**float**)fm \* **this**.width, 0.0F, (**float**)fm \* **this**.width, (**float**)**this**.getHeight(), pen3);  
 canvas.drawLine((**float**)fm \* **this**.width + 1.0F, 0.0F, (**float**)fm \* **this**.width + 1.0F, (**float**)**this**.getHeight(), pen2);  
 }  
  
 pen1.setStrokeWidth(5.0F);  
  
 **for**(fm = 0; fm < 9; ++fm) {  
 **if**(fm % 3 == 0) {  
 canvas.drawLine(0.0F, (**float**)fm \* **this**.height, (**float**)**this**.getWidth(), (**float**)fm \* **this**.height, pen1);  
 canvas.drawLine(0.0F, (**float**)fm \* **this**.height + 1.0F, (**float**)**this**.getWidth(), (**float**)fm \* **this**.height + 1.0F, pen2);  
 canvas.drawLine((**float**)fm \* **this**.width, 0.0F, (**float**)fm \* **this**.width, (**float**)**this**.getHeight(), pen1);  
 canvas.drawLine((**float**)fm \* **this**.width + 1.0F, 0.0F, (**float**)fm \* **this**.width + 1.0F, (**float**)**this**.getHeight(), pen2);  
 }  
 }

Paint foreground = **new** Paint(1);  
 foreground.setTextSize(70.0F);  
 foreground.setStyle(Style.FILL);  
 foreground.setTextAlign(Align.CENTER);  
 FontMetrics var12 = foreground.getFontMetrics();  
 **float** x = **this**.width / 2.0F;  
 **float** y = **this**.height / 2.0F - (var12.ascent + var12.descent) / 2.0F;  
  
 **for**(**int** select = 0; select < 9; ++select) {  
 **for**(**int** j = 0; j < 9; ++j) {  
 **if**(**this**.game.getOriTile(select, j) == **this**.game.getTile(select, j)) {  
 foreground.setColor(-65536);  
 } **else** {  
 foreground.setColor(-16776961);  
 }  
  
 canvas.drawText(**this**.game.getTileString(select, j), (**float**)select \* **this**.width + x, (**float**)j \* **this**.height + y, foreground);  
 }  
 }  
  
 Paint var13 = **new** Paint();  
 var13.setColor(Color.argb(64, 255, 80, 0));  
 canvas.drawRect(**this**.selectRect, var13);  
 }  
}



background.setColor(Color.LTGRAY)

pen1.setStrokeWidth(5.0F);  
  
 **for**(fm = 0; fm < 9; ++fm) {  
 **if**(fm % 3 == 0) {  
 canvas.drawLine(0.0F, (**float**)fm \* **this**.height, (**float**)**this**.getWidth(), (**float**)fm \* **this**.height, pen1);  
 canvas.drawLine(0.0F, (**float**)fm \* **this**.height + 1.0F, (**float**)**this**.getWidth(), (**float**)fm \* **this**.height + 1.0F, pen2);  
 canvas.drawLine((**float**)fm \* **this**.width, 0.0F, (**float**)fm \* **this**.width, (**float**)**this**.getHeight(), pen1);  
 canvas.drawLine((**float**)fm \* **this**.width + 1.0F, 0.0F, (**float**)fm \* **this**.width + 1.0F, (**float**)**this**.getHeight(), pen2);  
 }  
 }  
**vẽ các vùng 3x3**

Vẽ phần vùng dạng lưới

Paint background = new Paint

Paint pen1 = new Paint();

Paint pen2 = new Paint();

Paint pen3 = new Paint();

background.setColor(-3355444);

canvas.drawRect(0.0F, 0.0F, (float)this.getWidth(), (float)this.getHeight(), background);

pen1.setColor(-16777216);

pen2.setColor(-16711936);

pen3.setColor(-1);

int fm;

for(fm = 0; fm < 9; ++fm) {

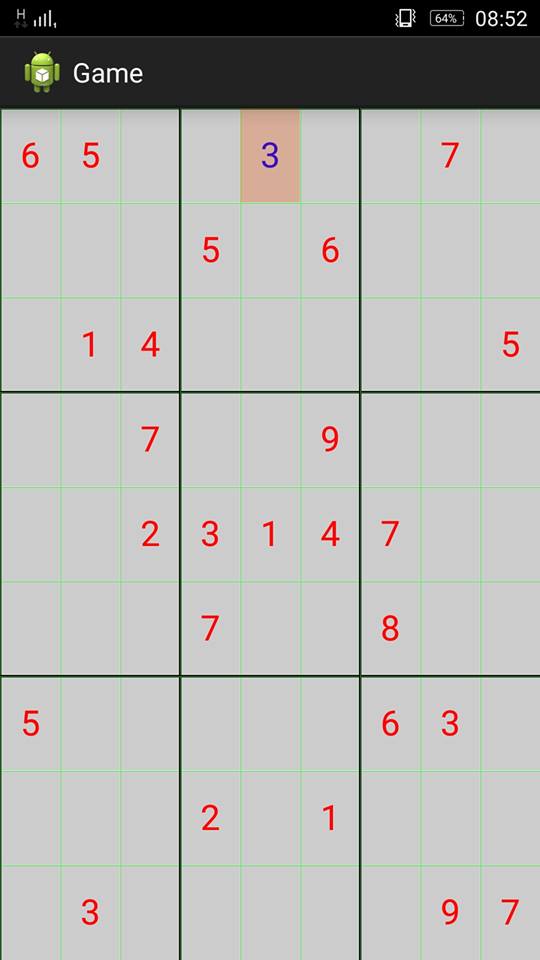
canvas.drawLine(0.0F, (float)fm \* this.height, (float)this.getWidth(), (float)fm \* this.height, pen3);

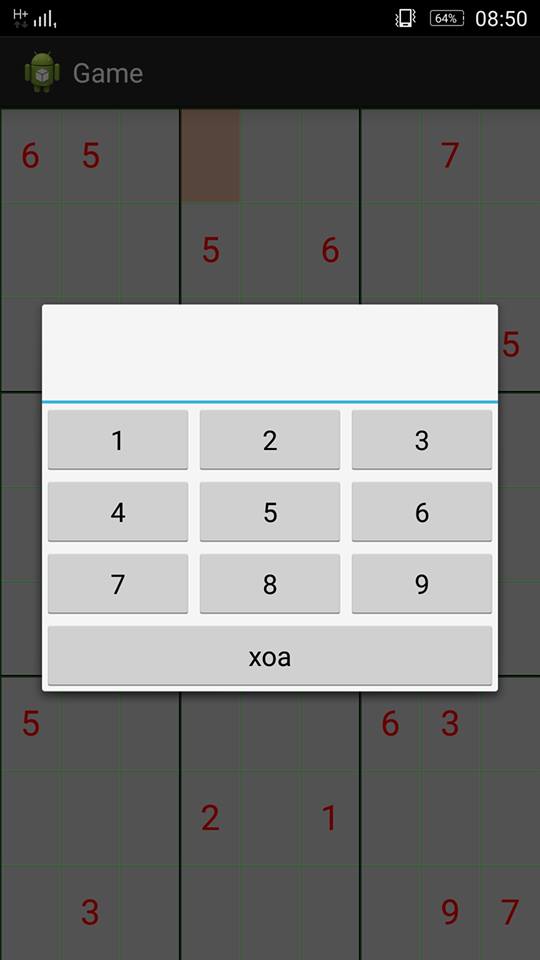
canvas.drawLine(0.0F, (float)fm \* this.height + 1.0F, (float)this.getWidth(), (float)fm \* this.height + 1.0F, pen2);

canvas.drawLine((float)fm \* this.width, 0.0F, (float)fm \* this.width, (float)this.getHeight(), pen3);

canvas.drawLine((float)fm \* this.width + 1.0F, 0.0F, (float)fm \* this.width + 1.0F, (float)this.getHeight(), pen2);

}





Chọn số

vẽ chữ được điền vào

canvas.drawText(this.game.getTileString(select, j), (float)select \* this.width + x, (float)j \* this.height + y, foreground);

}

}

Paint var13 = new Paint();

var13.setColor(Color.argb(64, 255, 80, 0));

canvas.drawRect(this.selectRect, var13);

}

}

vẽ chữ được cố định trên màn chơi

Paint foreground = new Paint(1);

foreground.setTextSize(70.0F);

foreground.setStyle(Style.FILL);

foreground.setTextAlign(Align.CENTER);

FontMetrics var12 = foreground.getFontMetrics();

float x = this.width / 2.0F;

float y = this.height / 2.0F - (var12.ascent + var12.descent) / 2.0F;

for(int select = 0; select < 9; ++select) {

for(int j = 0; j < 9; ++j) {

if(this.game.getOriTile(select, j) == this.game.getTile(select, j)) {

foreground.setColor(-65536);

} else {

foreground.setColor(-16776961);

}

Vẽ số vào ô đã chọn

