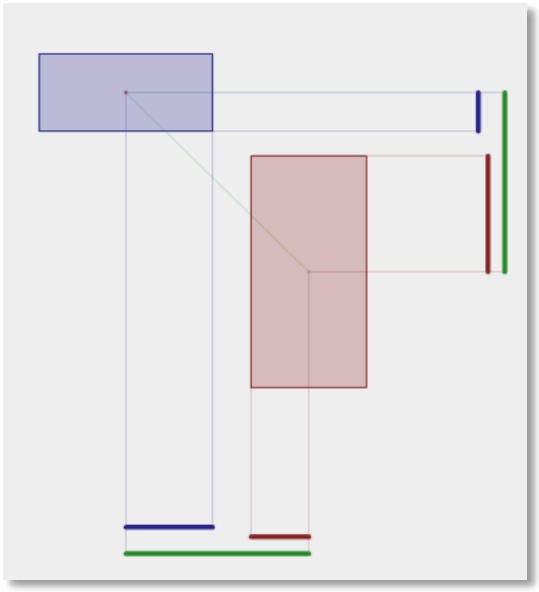
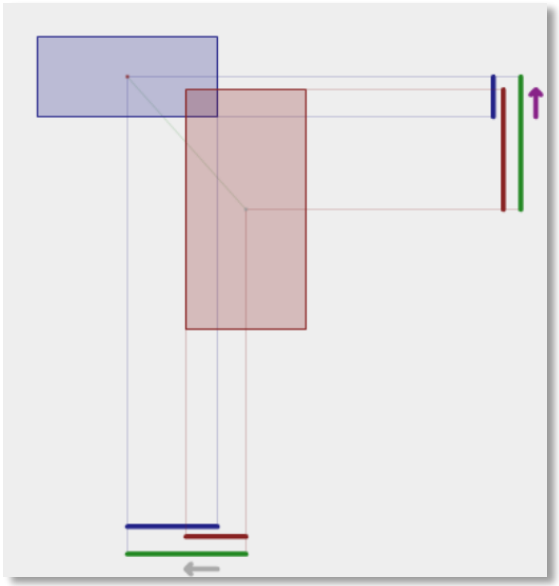
Xét va chạm

Thực hiện:

Phát triển tiếp từ bài 5:

Phần này ta sẽ xét va chạm giữa các đối tượng.





Theo thuật toán ta thấy rằng 2 hình chữ nhật chạm nhau khi thỏa 2 điều kiện (cho 2 trục x,y là).

- Kích thước từ tâm của HCN_A đến tâm của HCN_B theo trục x <= nửa rộng HCN_A + nửa rộng HCN_B
- Kích thước từ tâm của HCN_A đến tâm của HCN_B theo trục y <= nửa cao HCN_A + nửa cao HCN_B

Các đại lượng này ta đều có thể tính được vì ta có tọa độ x,y của từng HCN và có kích thước của từng HCN.

Phần 1: Sửa class Enemies và Bullet

Giờ ta phải bổ sung cho 2 class là Bullet và Enemies các hàm để lấy được chiều rộng và chiều cao của đối tượng, và thêm hàm để lấy tâm theo trục X, tâm theo trục Y.

1. Mở file Bullet và thêm 4 hàm như sau:

```
public int getWidth()
{
	return bitmap.getWidth();
}
public int getHeight()
{
	return bitmap.getHeight();
}
public int gettamX()
{
	//tam x=toa do x cong voi nua rong
	return x+(bitmap.getWidth()/2);
}
```

```
public int gettamY()
{
	//tam y=toa do y cong nua cao
	return y+(bitmap.getHeight()/2);
}
```

2. Copy đoạn 4 hàm vừa tạo trong Bullet, mở class Enemies và paste 4 hàm này vào.

Phần 2: Xây dựng hàm va chạm trong GamePanel

1. Mở file GamePanel.java ta tạo ra một hàm mới vc_b_e, hàm này dùng để xét va chạm giữa bullet và enemies. Nó nhận vào 1 bullet và một enemies sau đó tính ra các thông tin cần thiết rồi xét điều kiện, nếu thỏa điều kiện tức là có va chạm nó trả về true, ngược lại không có va chạm thì trả về false, (có thể viết ngắn hơn nhưng chẻ ra cho nó dễ hiểu ;-))

```
public boolean vc_b_e(Bullet bullet,Enemies enemies)
{

	float nuarong_b=(float)bullet.getWidth()/2;
	int nuacao_b=bullet.getHeight()/2;
	float nuarong_e=(float)enemies.getWidth()/2;
	int nuacao_e=enemies.getHeight()/2;
	//khoang cach 2 tam theo x
	int kc_ht_x=Math.abs(bullet.gettamX()-enemies.gettamX());
	//khoang cach 2 tam theo y
	int kc_ht_y=Math.abs(bullet.gettamY()-enemies.gettamY());
	if(kc_ht_x<=nuarong_b+nuarong_e && kc_ht_y<=nuacao_b+nuacao_e)
		return true;
	else
		return false;
}
```

2. Tiếp theo ta xây một hàm mới tên “xetvacham” hàm này sẽ duyệt tất cả bullets và tương ứng với mỗi bullets sẽ duyệt tất cả enemies để xét xem nó có va chạm không, nếu có va chạm giữa một bullets và một enemies nào đó sẽ xóa 2 thằng đó ra khỏi ArrayList. Chỗ này nên để trong try catch.

```
public  void xetvacham(Canvas canvas)
{
	try{
		for(int i=0;i<bullets.size();i++)
			for(int j=0;j<enemies.size();j++)
			{
				if(vc_b_e(bullets.get(i), enemies.get(j))==true)
				{
					bullets.remove(i);
					enemies.remove(j);
				}
			}
	}catch(Exception e)
	{
		Log.d("loi",e.toString());
	}

}
```

3. Cuối cùng trong hàm onDraw, sau khi vẽ bullet và enemies xong ta gọi hàm xetvacham để bỏ các đối tượng đã va chạm với nhau.

```
            if(myelement!=null)
            {
                  myelement.doDraw(canvas);//ve may bay
                  this.doDrawBullet(canvas); //ve tap hop vien dan
                  this.doDrawEnemies(canvas);//ve tap hop Enemies
                  xetvacham(canvas);//xet va cham
            }
```