**Sinh viên thực hiện:**

**1. Nguyễn Thị Hương - 52039**

**Bài kiểm tra số 2**

**Môn: Phát Triển Hệ Điều Hành Mã Nguồn Mở**

**Đề tài nhóm : Lập trình game Sodoku trên Android**

**Phần tìm hiểu**:**Xử lí chạm cho game Sodoku lập trình trên Android**.

1. Xử lí chạm (Touch Event ) là gì ?

* Touch Event là sự kiện quan trọng nhất trong game di động, nó dễ tạo ra và cung cấp nhiều chức năng.
* Ví dụ: Khi bạn chạm vào một màn hình(cảm ứng) của một điện thoại di động, lúc này điện thoại sẽ nhận sự kiện này và xử lý nó. Có thể thao tác của bạn sẽ được trả lời hay không được đáp ứng, nhưng ở bên dưới vẫn có một cái gì đó xảy ra.
* Khi người dùng chạm vào một widget trên một thiết bị màn hình cảm ứng, một sự kiện cảm ứng xảy ra trên các widget. Hầu hết thời gian, bạn không cần phải biết làm thế nào để xử lý sự kiện này. Thay vào đó, bạn có thể sử dụng sự kiện khác như một sự kiện Click để có được ứng dụng của bạn để làm việc theo cách bạn muốn.
* Thông thường để khởi tạo và xử lí một Touch event của người chơi tại một điểm thì hầu hết ta phải sử dụng tới 3 sự kiện như sau:
* **OnTouchBegin**  : Sự kiện được kích hoạt khi bạn nhấn xuống.
* **OnTouchMoved** : Sự kiện được kích hoạt nếu bạn di chuyển các đối tượng và vẫn nhấn xuống.
* **OnTouchEnded** : Sự kiện được kích hoạt khi bạn thả tay ra khỏi màn hình cảm ứng.

***2: Code mô tả sự kiện chạm nút:***

public boolean onTouchEvent(MotionEvent event)

1

{

if(event.getAction()!=MotionEvent.ACTION\_DOWN)

return super.onTouchEvent(event);

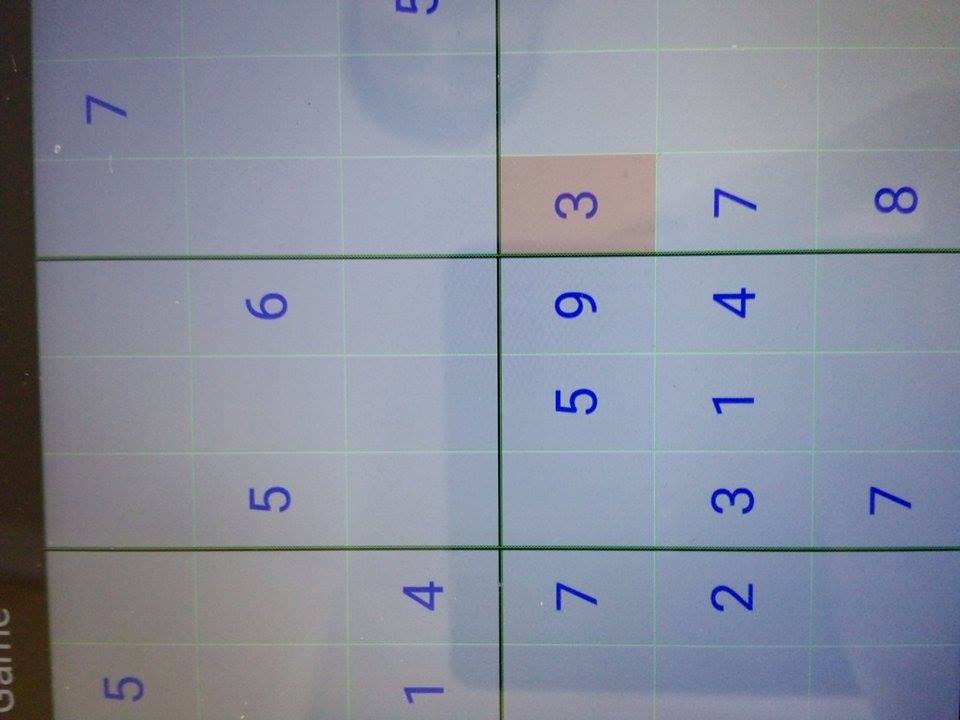
select ((int)(event.getX()/width ),(int)(event.getY()/height));

game.showKeypadOrError(selectX, selectY);

return true;

}

* 1: Trong phương pháp onTouch, sự kiện đầu tiên là cho các giá trị X,Y khi người dùng ấn vào. Nếu kiểm tra thấy sự kiện không trùng với sự kiện nhấn xuống (ACTION\_DOWN ) thì nó trả về giá trị return super.onTouchEvent(event).
* Ngược lại ta xác định được chiều cao, chiều rộng của sự kiện và từ đó tìm ra được vị trí X,Y để tiếp tục xử lí sự kiện game.showKeypadOrError(selectX, selectY);



Hình trên : Mô tả sự kiên khi ta chạm vào một nút

* **Code phần mô tả 1 class Keypad để xử lý sự kiện**
* Phần vẽ hình theo Canvas ta hiện thị được hình ảnh bàn cho game sodoku với bàn cờ gồm 81 ô, tạo thành 1 ma trận 9 hàng 9 cột.
* Khi ta chạm vào một ô thì sẽ hiện lên bàn phím ảo có các số từ 1->9 để ta lựa chọn.



**public class KeyPad extends Dialog {**

protected static final String TAG = "Sudoku";

private final View keys[] = new View[9];

private View keypad;

private final int useds[];

private final PuzzleView puzzleView;

public KeyPad(Context context, int useds[], PuzzleView puzzleView) {

super(context);

// TODO Auto-generated constructor stub

this.useds = useds;

this.puzzleView = puzzleView;}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_key\_pad);

findViews();

for (int element : useds) {

if(element != 0) {

keys[element - 1].setVisibility(View.INVISIBLE);

}}

setListeners();}

**private void setListeners() {**

for (int i = 0; i < keys.length; i++) {

final int t = i + 1;

keys[i].setOnClickListener(new View.OnClickListener() {

@Override

**public void onClick(View v) {**

returnResult(t);

}});}

keypad.setOnClickListener(new View.OnClickListener() {

@Override

**public void onClick(View v) {**

returnResult(0);

}});}

protected void returnResult(int tile) {

puzzleView.setSelectedTile(tile);

Ánh xạ key[0] bằng button có id là key\_pad1

dismiss();}

**private void findViews() {**

keypad = findViewById(R.id.keypad);

keys[0] = findViewById(R.id.keypad\_1);

keys[1] = findViewById(R.id.keypad\_2);

keys[2] = findViewById(R.id.keypad\_3);

keys[3] = findViewById(R.id.keypad\_4);

keys[4] = findViewById(R.id.keypad\_5);

keys[5] = findViewById(R.id.keypad\_6);

keys[6] = findViewById(R.id.keypad\_7);

keys[7] = findViewById(R.id.keypad\_8);

keys[8] = findViewById(R.id.keypad\_9); }

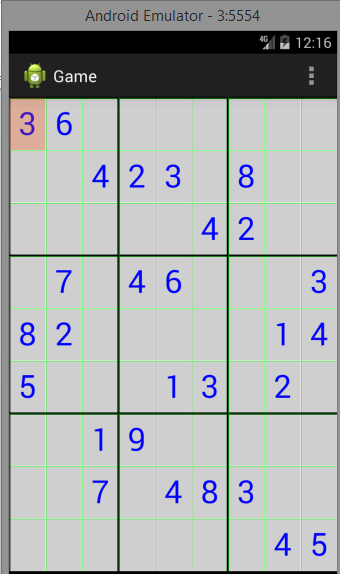
**3: Hằng số của lớp MotionEvent được sử dụng trong bài:**

* **ACTION\_DOWN***:*  Sự bắt đầu của sự kiện cảm ứng.

**4: Phương pháp của lớp MotionEvent**

|  |  |
| --- | --- |
| ***getX:*** | Lấy một giá trị float cho vị trí của trục X của cảm ứng. |
| ***getY:*** | Lấy một giá trị float cho vị trí của trục Y của cảm ứng. |





* **Cuối cùng là kết quả của quá trình xử lí sự kiện chạm nút và quá trình hiện bàn phím ảo.**