TRƯỜNG 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 (Tạo bàn phím ảo. Bài báo cáo số 2)

*Người hướng dẫn: (Thầy) Vũ Đình Trung*

*Sinh viên thực hiện: NGUYỄN THỊ TƯƠI*

*Hải phòng, tháng 4 năm 2017*

Mục lục

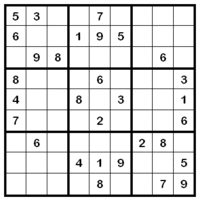
[CHƯƠNG 1. TÌM HIỂU VỀ GAME SODOKU 4](#_Toc477946381)

[CHƯƠNG 2. NHỮNG KIẾN THỨC CẦN CÓ ĐỂ XÂY DỰNG GAME 6](#_Toc477946382)

[CHƯƠNG 3: TẠO BÀN PHÍM ẢO 7](#_Toc477946384)

[CHƯƠNG 4 HOÀN THIỆN GAME 12](#_Toc477946385)

# CHƯƠNG 1. TÌM HIỂU VỀ GAME SODOKU

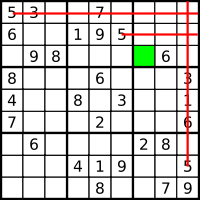


Một Bảng chơi sodoku

 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.

Cách chơi đúng:



Phía trên ở bên phải phải có số 5. Bằng cách hatch hai bên và ở trên và vì có số 5 ở những nơi khác, người chơi bảng này có thể trừ ra các ô trống ở vùng mà không thể có số 5, rồi chỉ còn lại một ô (được tô đậm).

Người chơi chiến thắng nếu 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.



# CHƯƠNG 2. NHỮNG KIẾN THỨC CẦN CÓ ĐỂ XÂY DỰNG GAME

1. Biết tải và cài đặt ứng dụng Eclipse Neon 1 + Cài đặt ADT Plugin cho Eclipse để lập trình Android, Biết cách tạo máy ảo trong Eclipse
2. Có kiến thức về kế thừ,overrid,static, Interface trong java,final hằng số
3. Biết làm việc với relative Layout,Linear Layout, table layout,textview,xử lý button + Edittext + xử lý sự kiện, dialog,canvas, [Intent - truyền dữ liệu giữa các activity](https://www.youtube.com/watch?v=OhO3yKOEmQY)
4. Chơi game sodoku thành thạo

# CHƯƠNG 3: TẠO BÀN PHÍM ẢO

1. **Tạo giao diện keypad**

* KeyPad.xml:

Thiết kế giao diện bàn phím. Bàn phím có 9 số được tạo bằng 9 buttom.

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/keypad"

android:orientation="vertical"

android:stretchColumns="\*"

>

<TableRow >

<Button

android:id="@+id/keypad\_1"

android:text="1"

/>

<Button

android:id="@+id/keypad\_2"

android:text="2"

/>

<Button

android:id="@+id/keypad\_3"

android:text="3"

/>

</TableRow>

<TableRow >

<Button

android:id="@+id/keypad\_4"

android:text="4"

/>

<Button

android:id="@+id/keypad\_5"

android:text="5"

/>

<Button

android:id="@+id/keypad\_6"

android:text="6"

/>

</TableRow>

<TableRow >

<Button

android:id="@+id/keypad\_7"

android:text="7"

/>

<Button

android:id="@+id/keypad\_8"

android:text="8"

/>

<Button

android:id="@+id/keypad\_9"

android:text="9"

/>

</TableRow>

</TableLayout>

1. **Tạo 1 class Keypad để xử lý sự kiện**

* Ở class này chúng ta thực hiện ánh xạ đến giao diện keypad.

public class KeyPad extends Dialog {

protected static final String TAG = "Sudoku";

private final View keys[] = new View[9]; // khởi tạo 1 mảng 9 phần tử có kiểu dữ liệu là view

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);

dismiss();

}

private void findViews() {

// Mapujemy nasza klawiature wraz z kazdym przyciskiem

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);

}

1. **Đây là kết quả:**



Để tăng thêm thử thách cho người dùng, chúng ta sẽ cho hiển thị hết keypad ảo , thay vì ẩn đi những title đã dùng như sau

for (int element : useds) {

if(element != 0) {

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

}

}

Intent(Game.this,MainActivity.class);

startActivity(intent);

}});

builder.create().show();

}

}

* Chèn 1 số vào mảng:

private void setTile(int x, int y, int value) { //chen so vao mang puzzle

puzzle[y\*9+x] = value; }

* Lây 1 số từ mảng:

private int getTile(int x, int y) { // lay so tu mang puzzle

return puzzle[y\*9+x];

}

* Chèn số vào bảng:

public String getTileString(int x, int y) {

int v = getTile(x, y);

if (v==0)

return "";

else

return String.valueOf(v);

}

# CHƯƠNG 4 HOÀN THIỆN GAME

