BÁO CÁO

ĐỒ ÁN LẬP TRÌNH ANDROID CĂN BẢN

|  |  |
| --- | --- |
| Sinh viên thực hiện: | ĐỖ HOÀNG TUẤN ANH |
| Số điện thoại: | 01212123231 |
| Email: | [anh.dht3772@gmail.com](mailto:anh.dht3772@gmail.com) |
| Đề tài: | Xây dựng trò chơi Sudoku |
| Ngày khai giảng: | 26/08/2016 |
| Ca học: | 2/4/6 19g30-21g30 |
| Ngày nộp: | 05/12/2016 |

# **MỤC LỤC**

[**MỤC LỤC** 1](#_Toc467941095)

[**1.** **ĐỒ ÁN XÂY DỰNG “TRÒ CHƠI SUDOKU”** 4](#_Toc467941098)

[**1.1.** **Giới thiệu về trò chơi Sudoku** 4](#_Toc467941100)

[**1.2.** **Giao diện ứng dụng “Trò chơi Sudoku”** 4](#_Toc467941101)

[**1.3.** **Các chức năng chính của ứng dụng “Trò chơi Sudoku”** 8](#_Toc467941102)

[**1.4.** **Phân tích thiết kế Database** 24](#_Toc467941103)



## **ĐỒ ÁN XÂY DỰNG “TRÒ CHƠI SUDOKU”**



### **Giới thiệu về trò chơi Sudoku**

Sudoku là một trò chơi trí tuệ nổi tiếng, thu hút nhiều người tham gia thuộc nhiều tầng lớp, độ tuổi khác nhau. Sudoku ra đời ở Nhật và không lâu sau đó đã nhanh chóng lan rộng trên toàn thế giới.

Sudoku là một loại trò chơi lôgic 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.

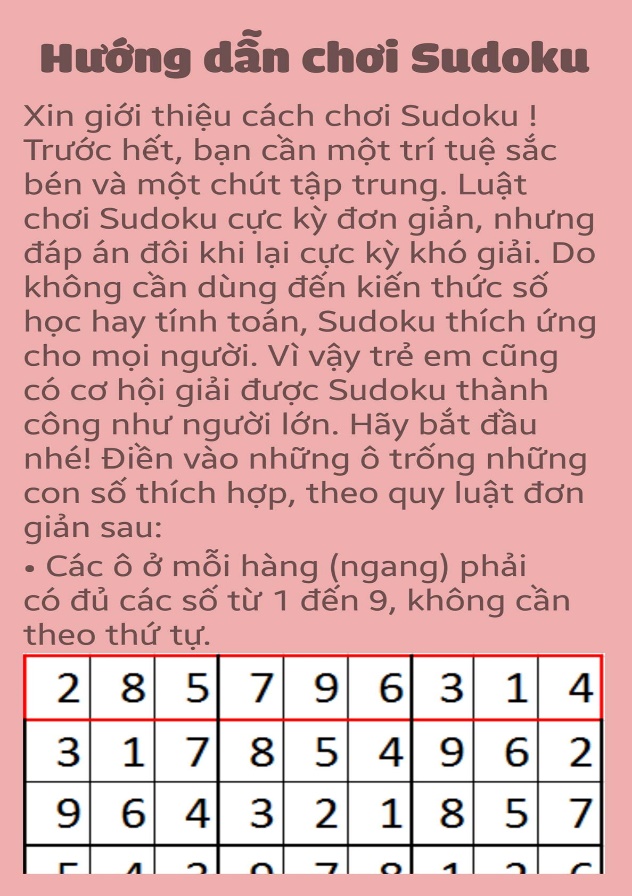
Một đề Sudoku là một hình vuông, mỗi chiều có 9 ô nhỏ, hợp thành 9 cột, 9 hàng và được chia thành 9 ô lớn 3x3. Một vài ô nhỏ được đánh số, đó là những manh mối duy nhất để bạn tìm lời giải. Tùy theo mức độ nhiều hay ít của các manh mối, các câu đố được xếp theo mức độ dễ, trung bình hay khó.

### **Giao diện ứng dụng “Trò chơi Sudoku”**

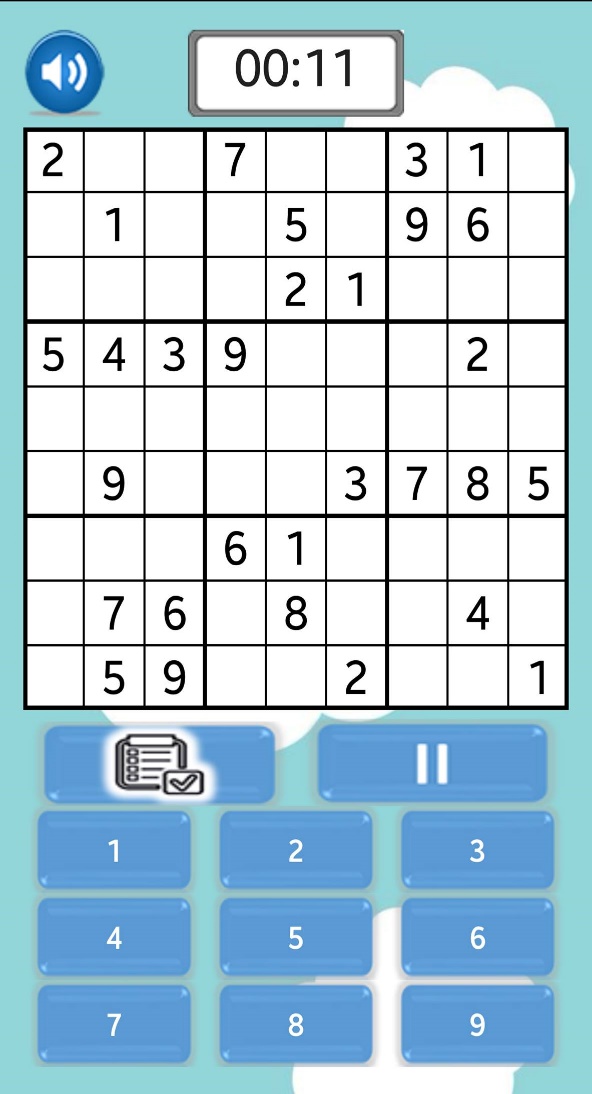
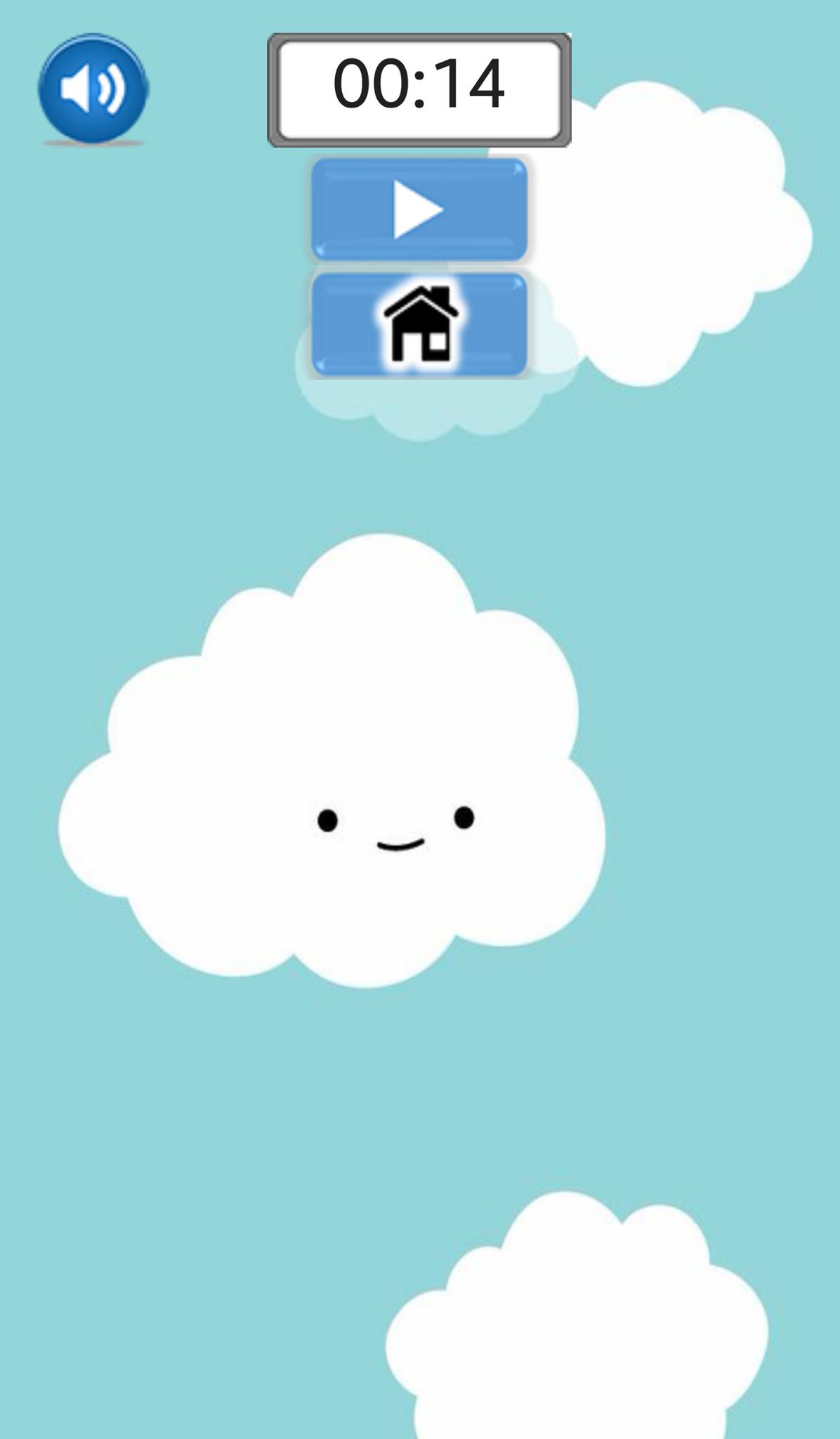
**\* Giao diện chính: \* Giao diện chọn độ khó:**

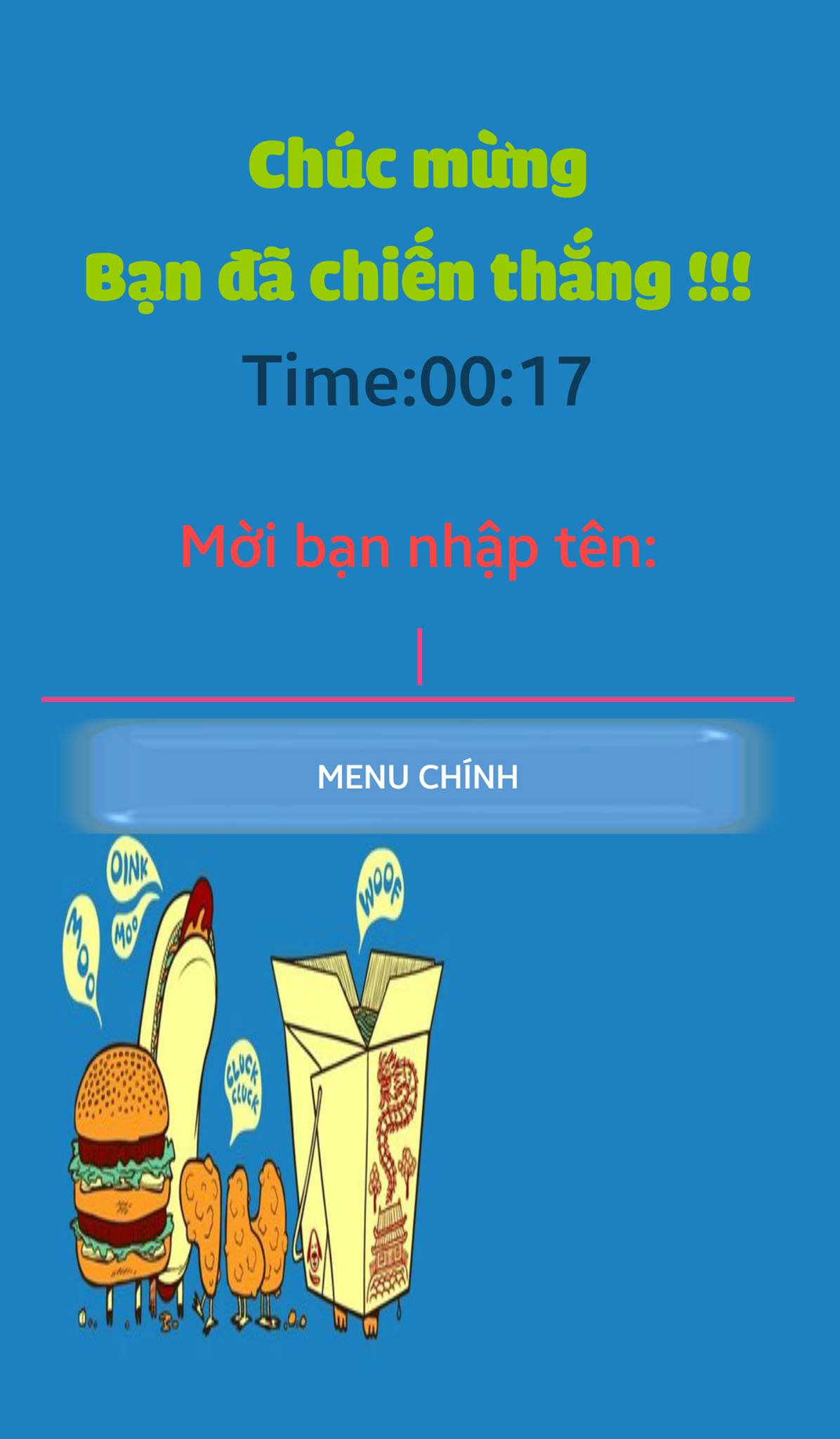
****

**\*Giao diện Điểm cao: \*Giao diện Hướng dẫn:**

****

**\*Giao diện Game: \*Giao diện Pause Game:**

****

**\* Giao diện Win Game:**

### **Các chức năng chính của ứng dụng “Trò chơi Sudoku”**

* + 1. *Chức năng New Game*Ở chức năng này gồm các chức năng nhỏ khác như: Chuyển Intent sang Activity chọn độ khó và sang Activity Game, Vẽ bàn chơi, sử dụng thuật toán check logic kiểm tra trùng trong game Sudoku, check kết quả, Pause game, Resume game,…

***\* Chuyển Intent sang Activity chọn độ khó, Continue game, Tutorial, High Score:***

Ở lớp Main Activity khai báo các Button cần gán sự kiện

Button**buttonnew**,**buttoncontinue**,**buttonintruction**,**buttonhighscore**;

Hàm ánh xạ các Button:

**protected void** addControl()*//Ham gan su kien cho cac button*{  
 **buttonnew**=(Button) findViewById(R.id.***newgame***);  
 **buttonnew**.setOnClickListener(**this**);  
 **buttoncontinue**=(Button) findViewById(R.id.***continuegame***);  
 **buttoncontinue**.setOnClickListener(**this**);  
 **buttonintruction**=(Button) findViewById(R.id.***intruction***);  
 **buttonintruction**.setOnClickListener(**this**);  
 **imgbtnsound**=(ImageButton)findViewById(R.id.***imageButtonSound***);  
 **imgbtnsound**.setOnClickListener(**this**);  
 **buttonhighscore** =(Button) findViewById(R.id.***ButtonHighscore***);  
 **buttonhighscore**.setOnClickListener(**this**);  
 **imgbtnvie**=(ImageButton) findViewById(R.id.***imgbuttonvie***);  
 **imgbtnvie**.setOnClickListener(**this**);  
 **imgbtneng**=(ImageButton) findViewById(R.id.***imgbuttoneng***);  
 **imgbtneng**.setOnClickListener(**this**);  
}

Hàm chuyển Intent chọn độ khó:

**private void** chooseDifficult() *//Ham chuyen qua menu chon do kho*{  
 *decontinue* =**null**;  
 *dapancontinue* =**null**;  
 *manchoicontinue* =**null**;  
 *timecontinue* =**null**;  
 Intent intent =**new** Intent(MainActivity.**this**,Difficult.**class**);  
 **this**.startActivity(intent);  
}

Hàm gán sự kiện cho các Button:

String **language**;*//bien luu trang thai ngon ngu*

@Override  
**public void** onClick(View view) {  
 **switch** (view.getId()){  
 **case** R.id.***continuegame***: {  
 {  
 Intent intent = **new** Intent(MainActivity.**this**, Game.**class**);  
 **this**.startActivity(intent);  
 }  
 };**break**;  
 **case** R.id.***intruction***:{  
 Intent intent =**new** Intent(MainActivity.**this**,Tutorial.**class**);  
 **this**.startActivity(intent);  
 };**break**;  
 **case** R.id.***newgame***: chooseDifficult();**break**;  
 **case** R.id.***ButtonHighscore***:  
 {  
 Intent intent =**new** Intent(MainActivity.**this**,HighScore.**class**);  
 **this**.startActivity(intent);  
 };**break**;  
 **case** R.id.***imageButtonSound***:{  
 **if**(*sound*==**true**)  
 {  
 **imgbtnsound**.setBackgroundResource(R.drawable.***soundofficon***);  
 *sudokusound*.pause();  
 *sound*=**false**;  
 }  
 **else** {  
 **imgbtnsound**.setBackgroundResource(R.drawable.***soundonicon***);  
 *sudokusound*.start();  
 *sound*=**true**;  
 }  
 }**break**;  
  
 **case** R.id.***imgbuttoneng***: **language**=**"en"**;  
 changeLanguage();setContinue();**break**;  
 **case** R.id.***imgbuttonvie***: **language**=**"vi"**;  
 changeLanguage();setContinue();**break**;  
 }  
}

***\* Vẽ bàn chơi, vẽ số đề chơi, vẽ số người chơi nhập:***

Ở class PuzzleView, khai báo các đối tượng, thuộc tính để vẽ bàn chơi Sudoku extends từ View:

**private final** Game **game**;  
**private final** Rect **selRect** = **new** Rect(); *//hinh chu nhat the hien o dang chon***private float width**;  
**private float height**;  
**protected int selX**=-1; *//stt cot***protected int selY**=-1; *// stt hang*

Các hàm sự kiện phục vụ cho việc Vẽ bàn chơi, vẽ số đề chơi, vẽ số người chơi nhập:

**public** PuzzleView(Context context) {  
 **super**(context);  
 **this**.**game** = (Game) context;  
 setFocusable(**true**);  
 setFocusableInTouchMode(**true**);  
}  
  
@Override  
**protected void** onSizeChanged(**int** w, **int** h, **int** oldw, **int** oldh) { *//chia man hinh thanh 9 cot 9 hang* **width** = w / 9f;  
 **height** = h / 9f;  
 **super**.onSizeChanged(w, h, oldw, oldh);  
}  
  
@Override  
**protected void** onDraw(Canvas canvas) { *// Ve man choi Sudoku* **super**.onDraw(canvas);  
 *//chon mau khung choi* Paint day = **new** Paint();  
 day.setColor(Color.***BLACK***);  
 day.setStrokeWidth(10f);  
 Paint mong = **new** Paint();  
 mong.setColor(Color.***BLACK***);  
 mong.setStrokeWidth(5f);  
 *// dinh dang mau sac va kich co so de* Paint sode = **new** Paint(Paint.***ANTI\_ALIAS\_FLAG***);  
 sode.setColor(Color.***BLACK***);  
 sode.setStyle(Paint.Style.***FILL***);  
 sode.setTextSize(**height** \* 0.75f);  
 sode.setTextScaleX(**width** / **height**);  
 sode.setTextAlign(Paint.Align.***CENTER***);  
 *// dinh dang mau sac va kich co so dien sai* Paint sosai = **new** Paint(Paint.***ANTI\_ALIAS\_FLAG***);  
 sosai.setColor(Color.***RED***);  
 sosai.setStyle(Paint.Style.***FILL***);  
 sosai.setTextSize(**height** \* 0.75f);  
 sosai.setTextScaleX(**width** / **height**);  
 sosai.setTextAlign(Paint.Align.***CENTER***);  
 *//dinh dang mau sac va kich co so dien vao* Paint sonhap = **new** Paint(Paint.***ANTI\_ALIAS\_FLAG***);  
 sonhap.setColor(Color.***BLUE***);  
 sonhap.setStyle(Paint.Style.***FILL***);  
 sonhap.setTextSize(**height** \* 0.75f);  
 sonhap.setTextScaleX(**width** / **height**);  
 sonhap.setTextAlign(Paint.Align.***CENTER***);  
 *//tham so de ve so vao giua o choi* Paint.FontMetrics fm = sode.getFontMetrics();  
 **float** x = **width** / 2;  
 **float** y = **height** / 2 - (fm.**ascent** + fm.**descent**) / 2;  
 *// chon mau o dang duoc chon* Paint selected = **new** Paint();  
 selected.setColor(Color.***GREEN***);  
 *//ve so nguoi choi dien vao* {  
 **if** (**selX** >= 0 && **selY** >= 0 &&**this**.**game**.laygiatride(**selX**, **selY**) == **""**) {  
 canvas.drawRect(**selRect**, selected);  
 **if** (**this**.**game**.**so** != 0) {  
 **this**.**game**.setValue(**selX**, **selY**, **this**.**game**.**so**);  
 **if** (**this**.**game**.laygiatringuoichoi(**selX**, **selY**) != **""**)  
 canvas.drawRect(**selRect**, selected);  
 **else** canvas.drawText(**this**.**game**.laygiatringuoichoi(**selX**, **selY**), **selX** \* **width** + x, **selY** \* **height** + y, sonhap);  
 **this**.**game**.**so** = 0;  
 }  
 }  
 **else this**.**game**.**so**=0;  
 *//ve lai ban choi sau khi nhap* **for** (**int** i = 0; i < 9; i++) {  
 **for** (**int** j = 0; j < 9; j++) {  
 **if** (**this**.**game**.laygiatride(i, j) != **""**)  
 canvas.drawText(**this**.**game**.laygiatringuoichoi(i, j), i \* **width** + x, j \* **height** + y, sode);  
 **else** canvas.drawText(**this**.**game**.laygiatringuoichoi(i, j), i \* **width** + x, j \* **height** + y, sonhap);  
 }  
 }

*//cac dkien vẽ màu để ktra cac thuat toan giai Sudoku***if** (**selX** >= 0 && **selY** >= 0) {  
 *//check theo logic* {  
 *//lay gia tri x,y cac o trung theo cot,hang,o3x3* List<Integer> cot = **this**.**game**.checkcot(**selX**, **selY**);  
 List<Integer> hang = **this**.**game**.checkhang(**selX**, **selY**);  
 List<Integer> o3x3 = **this**.**game**.check3x3(**selX**, **selY**);  
 **int** saix = 0;  
 **int** saiy = 0;  
 **if** (cot.size() > 0) {  
 canvas.drawText(**this**.**game**.laygiatringuoichoi(**selX**, **selY**), **selX** \* **width** + x, **selY** \* **height** + y, sosai);  
 **for** (**int** i = 0; i < cot.size(); i++) {  
 saix = cot.get(i) % 9;  
 saiy = cot.get(i) / 9;  
 canvas.drawText(**this**.**game**.laygiatringuoichoi(saix, saiy), saix \* **width** + x, saiy \* **height** + y, sosai);  
 }  
 }  
 **if** (hang.size() > 0) {  
 canvas.drawText(**this**.**game**.laygiatringuoichoi(**selX**, **selY**), **selX** \* **width** + x, **selY** \* **height** + y, sosai);  
 **for** (**int** i = 0; i < hang.size(); i++) {  
 saix = hang.get(i) % 9;  
 saiy = hang.get(i) / 9;  
 canvas.drawText(**this**.**game**.laygiatringuoichoi(saix, saiy), saix \* **width** + x, saiy \* **height** + y, sosai);  
 }  
 }  
 **if** (o3x3.size() > 0) {  
 canvas.drawText(**this**.**game**.laygiatringuoichoi(**selX**, **selY**), **selX** \* **width** + x, **selY** \* **height** + y, sosai);  
 **for** (**int** i = 0; i < o3x3.size(); i++) {  
 saix = o3x3.get(i) % 9;  
 saiy = o3x3.get(i) / 9;  
 canvas.drawText(**this**.**game**.laygiatringuoichoi(saix, saiy), saix \* **width** + x, saiy \* **height** + y, sosai);  
 }  
 }  
  
 }  
 *//check theo dap an* **if** (**this**.**game**.**checkdapan**) {  
 **if** (**this**.**game**.laygiatride(**selX**, **selY**) == **""**) {  
 **if** (**this**.**game**.laygiatringuoichoi(**selX**, **selY**) != **this**.**game**.laygiatridapan(**selX**, **selY**) &&  
 **this**.**game**.laygiatringuoichoi(**selX**, **selY**) != **""**) {  
 Paint mausai = **new** Paint();  
 mausai.setColor(Color.***RED***);  
 canvas.drawRect(**selX** \* **width**, **selY** \* **height**, **selX** \* **width** + **width**, **selY** \* **height** + **height**, mausai);  
 }  
 canvas.drawText(**this**.**game**.laygiatringuoichoi(**selX**, **selY**), **selX** \* **width** + x, **selY** \* **height** + y, sonhap);  
 }  
  
 }  
}  
**this**.**game**.**checkdapan** = **false**;

*//ve ban choi* **for** (**int** i = 0; i < 10; i++) {  
 canvas.drawLine(0, i \* **height**, getWidth(), i \* **height**,mong);  
 canvas.drawLine(i \* **width**, 0, i \* **width**, getHeight(), mong);  
 }  
 **for** (**int** i = 0; i < 10; i++) {  
 **if** (i % 3 != 0) **continue**;  
 {  
 canvas.drawLine(0, i \* **height**, getWidth(), i \* **height**, day);  
 canvas.drawLine(i \* **width**, 0, i \* **width**, getHeight(), day);  
 }  
 }  
 {  
 canvas.drawLine(0, 0 \* **height**+5, getWidth(), 0 \* **height**+5, day);  
 canvas.drawLine(0 \* **width**+5, 0, 0 \* **width**+5, getHeight(), day);  
 canvas.drawLine(0, 9 \* **height**-5, getWidth(), 9 \* **height**-5, day);  
 canvas.drawLine(9 \* **width**-5, 0, 9 \* **width**-5, getHeight(), day);  
 }  
 }  
}  
  
**private void** getRect(**int** x, **int** y, Rect rect) {  
 *//ham xu ly chieu dai chieu rong de set kich thuoc hinh chu nhat* rect.set((**int**) (x \* **width**), (**int**) (y \* **height**), (**int**) (x \* **width** + **width**), (**int**) (y \* **height** + **height**));  
}  
  
**private void** select(**int** x, **int** y) {  
 invalidate(**selRect**);  
 *//xu ly toa do nguoi dung chon tren man hinh thanh gia tri theo hang cot* **selX** = Math.*min*(Math.*max*(x, 0), 8);  
 **selY** = Math.*min*(Math.*max*(y, 0), 8);  
 getRect(**selX**, **selY**, **selRect**);  
 invalidate(**selRect**);  
}  
  
@Override  
**public boolean** onTouchEvent(MotionEvent event) {  
 *//ham set su kien touch tren man hinh* **if** (event.getAction() != MotionEvent.***ACTION\_DOWN***)  
 **return super**.onTouchEvent(event);  
 select((**int**) (event.getX() / **width**), (**int**) (event.getY() / **height**));  
 Log.*d*(***TAG***, **"onTouchEvent: x "** + **selX** + **", y "** + **selY**);  
 **return true**;  
}

***\* Thuật toán check logic Sudoku:***

Check logic sao cho các số hàng ngang đủ từ 1-9 không trùng, các số hàng dọc đủ từ 1-9 không trùng, các số trong vùng ô 3x3 đủ từ 1-9 không trùng:

*/\* Cac buoc kiem tra Logic bai toan Sudoku \*/***protected** List<Integer> checkcot(**int** x,**int** y)  
{  
 *//ham kiem tra trung theo cot* List<Integer> mangsai=**new** ArrayList<Integer>();  
 **for** (**int** k = x; k < x + 1; k++)  
 **for** (**int** j = 0; j < 9; j++) {  
 **if** (x == k && y == j) ;  
 **else** {  
 **if** (laygiatringuoichoi(x, y) == laygiatringuoichoi(k,j)) {  
 mangsai.add(j\*9+k);  
 }  
 }  
 }  
 **return** mangsai;  
}  
  
**protected** List<Integer> checkhang(**int** x,**int** y)  
{  
 *//ham kiem tra trung theo hang* List<Integer> mangsai=**new** ArrayList<Integer>();  
 **for** (**int** k = y; k < y + 1; k++)  
 **for** (**int** j = 0; j < 9; j++) {  
 **if** (x == j && y == k) ;  
 **else** {  
 **if** (laygiatringuoichoi(x, y) == laygiatringuoichoi(j,k)) {  
 mangsai.add( k\*9+j);  
 }  
 }  
 }  
 **return** mangsai;  
}  
  
**protected** List<Integer> check3x3(**int** x, **int** y) {  
 *//ham kiem tra trung theo o 3x3* List<Integer> mangsai=**new** ArrayList<Integer>();  
 **int** k = 0;  
 **int** h = 0;  
 **if** (x < 3) k = 3;  
 **else if** (x < 6) k = 6;  
 **else if** (x < 9) k = 9;  
 **if** (y < 3) h = 3;  
 **else if** (y < 6) h = 6;  
 **else if** (y < 9) h = 9;  
 **for** (**int** i = k - 3; i < k; i++)  
 **for** (**int** j = h - 3; j < h; j++)  
 **if** (i == x && j == y) ;  
 **else** {  
 **if** (laygiatringuoichoi(x, y) == laygiatringuoichoi(i, j)) {  
 mangsai.add(j\*9+i);  
 }  
 }  
 **return** mangsai;  
}  
*/\* ------------------------------------------- \*/*

***\* Check đáp án so với kết quả và chuyển Intent sang Activity Win***

Hàm sự kiện này được xảy ra ở class Game:

**protected void** winGame() *//ham so sanh dap an voi man choi, neu dung auto win*{  
 **int** checkwin=0;  
 **for** (**int** i = 0; i < 9; i++)  
 **for** (**int** j = 0; j < 9; j++) {  
 **if**(laygiatringuoichoi(i,j)==laygiatridapan(i,j))  
 checkwin+=1;  
 }  
 **if**(checkwin==80) *{* **de**=**""**;  
 **dapan**=**""**;  
 **manchoi**=**""**;  
 updateData();  
 **time**.stop();  
 finish();  
 Intent intent =**new** Intent(Game.**this**,Win.**class**);  
 intent.putExtra(**"WIN"**,**time**.getText().toString());  
 **this**.startActivity(intent);  
 *}*}

***\*Các hàm sự kiện xử lý cho hoạt động của chương trình Pause, Resume, Destroy (class Game):***

@Override  
**protected void** onResume() { *//chuong trinh dang chay thi nhac chay va dong ho chay* **super**.onResume();  
 **if**(MainActivity.*sound*)MainActivity.*sudokusound*.start();  
 **time**.setBase((SystemClock.*elapsedRealtime*() + **timecontinue**));  
 **if**(**isChronometerRunnig**) {  
 **time**.start();  
 }  
 setIconSound();  
}  
@Override  
**protected void** onPause() { *//chuong trinh dang ngung thi nhac ngung va dong ho ngung* **super**.onPause();  
 **time**.stop();  
 MainActivity.*sudokusound*.pause();  
 MainActivity.*saveSoundStatus*(MainActivity.*sound*);  
 **if**(**isChronometerRunnig**)  
 updateData();  
}  
@Override  
**protected void** onDestroy() {  
 **super**.onDestroy();  
}

* + 1. *Chức năng Tutorial*

Khai báo Button:

Button **buttonintruction;**

Ánh xa:

**buttonintruction**=(Button) findViewById(R.id.***intruction***);  
**buttonintruction**.setOnClickListener(**this**);

Sự kiện chuyển Intent qua Tutorial Activity:

**case** R.id.***intruction***:{  
 Intent intent =**new** Intent(MainActivity.**this**,Tutorial.**class**);  
 **this**.startActivity(intent);  
};**break**;

* + 1. *Chức năng High Score*

Được thực hiện ở class HighScore

Tạo các đối tượng phục vụ cho chức năng High Score:

TextView **tv\_highscores**,**time**;  
String **thoigian** = **""**;

Hàm xử lý lấy dữ liệu High Score:

**protected void** getHighScore() *//ham lay du lieu nguoi choi tu data*{  
 Cursor c;  
 ArrayList<String> danhsach= **new** ArrayList<>();  
 c = MainActivity.*DatabaseSudoku*.layDataHighScore();  
 **if** (c.moveToFirst() == **true**) {  
 **do** {  
 danhsach.add(c.getString(0));  
 } **while** (c.moveToNext());  
 }  
 Collections.*sort*(danhsach);  
 **for**(**int** i=1;i<danhsach.size();i++) {  
 **if**(i>10) **break**;  
 **thoigian** += String.*valueOf*(i)+**" "**+ danhsach.get(i) + **"\n"**;  
 }  
}

Hàm onCreate ở class High Score:

@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_highscores***);  
  
 addStyleTextView();  
  
 **time**=(TextView) findViewById(R.id.***diem***);  
 getHighScore();  
 **time**.setText(**thoigian**);  
}

Ở class MainActivity, tạo button chuyển Intent sang Activity HighScore:

Button **buttonhighscore**;

Ánh xạ:

buttonhighscore =(Button) findViewById(R.id.***ButtonHighscore***);  
buttonhighscore.setOnClickListener(**this**);

Gán sự kiện cho button highscore:

**case** R.id.***ButtonHighscore***:  
{  
 Intent intent =**new** Intent(MainActivity.**this**,HighScore.**class**);  
 **this**.startActivity(intent);  
};**break**;

* + 1. *Chức năng bật tắt âm thanh*

Khai báo Image Button âm thanh:

ImageButton **imgbtnsound;**

Các hàm xử lý âm thanh:

**static** MediaPlayer *sudokusound*; *//am thanh nhac nen*

**static boolean** *sound*;*//bien luu trang thai am thanh trong va sau khi thoat game*

**protected void** createSound()*//Ham khoi tao am thanh nhac nen*{  
 *sudokusound*= MediaPlayer.*create*(**this**,R.raw.***sudoku\_soundtrack***);  
 *sudokusound*.setLooping(**true**);  
}  
  
**public void** getDataSound()*//Ham lay trang thai am thanh da luu*{  
 *sharedPreferences* =getSharedPreferences(**"sound"**,***MODE\_PRIVATE***);  
 *sound*=*sharedPreferences*.getBoolean(**"sound"**,**true**);  
}  
  
**static void** saveSoundStatus(**boolean** sound)*//Ham luu lai trang thai am thanh*{  
 SharedPreferences.Editor editor=*sharedPreferences*.edit();  
 editor.putBoolean(**"sound"**,sound);  
 editor.commit();  
}  
  
**protected void** setIconSound()*//Ham set hinh anh on/off cua button am thanh*{  
 **if**(*sound*) {  
 **imgbtnsound**.setBackgroundResource(R.drawable.***soundonicon***);  
 *sudokusound*.start();  
 }  
 **else** {  
 *sudokusound*.pause();  
 **imgbtnsound**.setBackgroundResource(R.drawable.***soundofficon***);  
 }  
}

* + 1. *Chức năng Continue*

*Tạo các đối tượng phục vụ cho việc Continue Game:*

**static** DataSudoku *data*;*//doi tuong lay du lieu tu class DataSudoku***static** Database *DatabaseSudoku*;*//contructor DataSudoku***static** String *decontinue*, *dapancontinue*, *manchoicontinue*, *timecontinue*;*//cac bien luu man choi,dap an, thoi gian continue***static** Cursor *c*;*// con tro de doc du lieu***int back**=0;  
**static** SharedPreferences *sharedPreferences*;*// doi tuong luu tru du lieu tren sharedpreferences*

*Các hàm sự kiện phục vụ cho Continue Game:*

**protected void** setContinue()*//Ham an/hien continue*{  
 **if**(*decontinue* ==**null**|| *decontinue*.compareTo(**""**)==0) **buttoncontinue**.setVisibility(View.***INVISIBLE***);  
 **else buttoncontinue**.setVisibility(View.***VISIBLE***);  
}  
  
**protected void** getData()*//Ham lay du lieu man choi cu tu data*{  
 *DatabaseSudoku* = **new** Database(MainActivity.**this**);  
 *DatabaseSudoku*.open();  
 *c* = *DatabaseSudoku*.layData();  
 Random r = **new** Random();  
 **if** (*c*.moveToPosition(r.nextInt(4)) == **true**) {  
 String de\_easy = *c*.getString(1);  
 String de\_medium = *c*.getString(2);  
 String de\_hard = *c*.getString(3);  
 String da\_easy = *c*.getString(4);  
 String da\_medium = *c*.getString(5);  
 String da\_hard = *c*.getString(6);  
 *data* = **new** DataSudoku(de\_easy, de\_medium, de\_hard, da\_easy, da\_medium, da\_hard);  
 }  
 **if**(*c*.moveToPosition(0)==**true**)  
 {  
 *decontinue* =*c*.getString(7);  
 }  
 **if**(*c*.moveToPosition(1)==**true**)  
 {  
 *manchoicontinue* =*c*.getString(7);  
 }  
 **if**(*c*.moveToPosition(2)==**true**)  
 {  
 *dapancontinue* =*c*.getString(7);  
 }  
 **if**(*c*.moveToPosition(4)==**true**)  
 {  
 *timecontinue* =*c*.getString(7);  
 }  
}

* + 1. *Chức năng thay đổi ngôn ngữ bằng cách khai thác tài nguyên chuỗi*

Khai báo hai Image Button:

ImageButton **imgbtnsound**,**imgbtnvie**,**imgbtneng**;

Ánh xạ:

**imgbtnvie**=(ImageButton) findViewById(R.id.***imgbuttonvie***);  
**imgbtnvie**.setOnClickListener(**this**);  
**imgbtneng**=(ImageButton) findViewById(R.id.***imgbuttoneng***);  
**imgbtneng**.setOnClickListener(**this**);

Các hàm chuyển ngôn ngữ:

**protected void** getDataLanguage()*//Ham lay trang thai ngon ngu da luu*{  
 **language**=*sharedPreferences*.getString(**"language"**,**"en"**);  
}  
  
**static void** saveLanguageStatus(String language)*//Ham luu lai trang thai ngon ngu*{  
 SharedPreferences.Editor editor=*sharedPreferences*.edit();  
 editor.putString(**"language"**,language);  
 editor.commit();  
}  
  
**private void** changeLanguage()*//Ham doi ngon ngu*{  
 Locale myLocale = **new** Locale(**language**);  
 Resources res = getResources();  
 DisplayMetrics dm = res.getDisplayMetrics();  
 Configuration conf = res.getConfiguration();  
 conf.**locale** = myLocale;  
 res.updateConfiguration(conf, dm);  
 **this**.setContentView(R.layout.***activity\_main***);  
 getData();  
 addControl();  
 addStyleButton();  
 setIconSound();  
}

*Gán sự kiện cho hai Image Button:*

**case** R.id.***imgbuttoneng***: **language**=**"en"**;  
 changeLanguage();setContinue();**break**;  
**case** R.id.***imgbuttonvie***: **language**=**"vi"**;  
 changeLanguage();setContinue();**break**;

* 1. **Phân tích thiết kế Database**

Ứng dụng sử dụng SQLite làm cơ sở dữ liệu chính cho ứng dụng.

Cở sở dữ liệu gồm có những dữ liệu sau

\_id: khóa chính

deEasy: 5 đề Easy

deMedium: 5 đề Medium

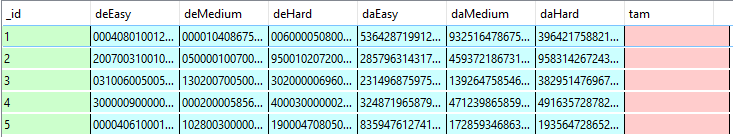
deHard: 5 đề Hard

daEasy: 5 đáp án đề Easy

daMedium: 5 đáp án đề Medium

daHard: 5 đáp án đề Hard

tam: dùng để lưu dữ liệu người chơi phục vụ cho Continue, High Score

****

**Các xử lý trong class Database:**

**public class** Database **extends** SQLiteOpenHelper {  
 **private** Context **context**;  
  
 **private** String **DB\_PATH** = **"data/data/com.example.toan.sudoku/"**;  
 **private static** String *DB\_NAME* = **"database.sqlite"**;  
 **private** SQLiteDatabase **myDatabase**;  
  
 **public** Database(Context context) *//contructor khoi tao database* {  
 **super**(context, *DB\_NAME*, **null**, 1);  
  
 **this**.**context** = context;  
 **boolean** dbexist = checkDatabase();  
  
 **if**(dbexist)  
 {  
 *//Neu Database ton tai thi ko lam gi ca* }  
 **else** {  
 System.***out***.println(**"Database doesn't exist!"**);  
  
 createDatabse();  
 }  
 }  
  
 **public void** createDatabse() *//ham khoi tao database* {  
  
 **this**.getReadableDatabase();  
  
 **try** {  
 copyDatabase();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 **private void** copyDatabase() **throws** IOException *//ham copy database vao thu muc data cua game* {  
  
 InputStream myinput = **context**.getAssets().open(*DB\_NAME*);  
  
 String outFileName = **DB\_PATH** + *DB\_NAME*;  
  
 OutputStream myOutput = **new** FileOutputStream(outFileName);  
  
 **byte**[] buffer = **new byte**[1024];  
 **int** length;  
  
 **while** ((length = myinput.read(buffer)) > 0) *//Coppy tung Mb, doc Database tu tu* {  
 myOutput.write(buffer, 0, length);  
 }  
  
 myOutput.flush();  
 myOutput.close();  
 myinput.close();  
 }  
  
 **public void** open()*//ham mo database* {  
 String myPath = **DB\_PATH** + *DB\_NAME*;  
 **myDatabase** = SQLiteDatabase.*openDatabase*(myPath, **null**, SQLiteDatabase.***OPEN\_READWRITE***);  
 }  
  
 **public synchronized void** close() *//ham dong database (co dong bo hoa)* {  
 **myDatabase**.close();  
 **super**.close();  
 }  
  
 **private boolean** checkDatabase() *//ham kiem tra database ton tai* {  
  
 **boolean** checkdb = **false**;  
  
 **try** {  
 String myPath = **DB\_PATH** + *DB\_NAME*;  
 File dbfile = **new** File(myPath);  
 checkdb = dbfile.exists();  
 }  
 **catch** (SQLiteException e)  
 {  
 System.***out***.println(**"Databse doesn't exist!"**);  
 }  
  
 **return** checkdb;  
 }  
  
 **public** Database(Context context, String name, SQLiteDatabase.CursorFactory factory, **int** version) {  
 **super**(context, name, factory, version);  
 } *//contructor khoi tao database* @Override  
 **public void** onCreate(SQLiteDatabase db) {  
  
 }  
 @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 }  
  
 **public void** update(String update,String hang)*//ham update du lieu moi vao database* {  
 ContentValues up = **new** ContentValues();  
 up.put(**"tam"**,update);  
 **myDatabase**.update(**"DBSudoku"**,up,**"\_id=?"**, **new** String[] {hang});  
 }  
  
 **public void** insertHighScore(String time) *//ham luu lai ket qua sau khi choi xong* {  
 ContentValues values=**new** ContentValues();  
 values.put(**"Time"**,time);  
 **myDatabase**.insert(**"HighScore"**,**null**,values);  
  
 }  
  
 **public** Cursor layData() *//ham get du lieu man choi tu database* {  
 Cursor contro=**myDatabase**.rawQuery(**"select \* from DBSudoku"**, **null**);  
 **return** contro;  
 }  
  
 **public** Cursor layDataHighScore()*//ham get du lieu ket qua choi tu database* {  
 Cursor contro=**myDatabase**.rawQuery(**"select \* from HighScore"**, **null**);  
 **return** contro;  
 }  
  
}