[**Android**](http://phocode.com/java/android-lap-trinh-android/android-luu-tru-du-lieu-tren-file/)**–**[**Lưu trữ dữ liệu trên file**](http://phocode.com/java/android-lap-trinh-android/android-luu-tru-du-lieu-tren-file/)

Hầu hết khi làm ứng dụng chúng ta đều phải lưu lại dữ liệu để dùng cho các lần làm việc sau này, chẳng hạn như lưu thông tin của người dùng, lưu trạng thái của ứng dụng… Có 3 cách lưu trữ dữ liệu đó là:

* Lưu dưới dạng từ điển trong các file ưu tiên
* Lưu file trong thẻ nhớ
* Lưu trong cơ sở dữ liệu SQLite

# Trong bài này chúng ta sẽ lưu trữ cách thứ nhất: Lưu dưới dạng từ điển trong các file ưu tiên

Các file ưu tiên ở đây là các file nằm trong thư mục shared-refs, thư mục này nằm trong thư mục cài đặt ứng dụng trên máy của bạn.

Nếu dữ liệu bạn lưu lại không lớn lắm thì bạn nên sử dụng cách này. Ở đây chúng ta sẽ dùng lớp SharedPreferences để lưu trữ dữ liệu, đối tượng SharedPreferences trỏ tới một file XML chứa dữ liệu là các cặp khóa-giá trị và cung cấp các phương thức để đọc và ghi dữ liệu trên file này.

#### Tạo đối tượng SharedPreferences

Để tạo một đối tượng SharedPreferences thì chúng ta dùng một trong 2 phương thức:

* getSharedPreferences() — phương thức này sẽ trả về một đối tượng SharedPreferences trỏ tới một file do chúng ta chỉ định.
* getPreferences() — mặc định ứng dụng đã có sẵn một file riêng để lưu dữ liệu, phương thức này sẽ trả về đối tượng SharedPreferences trỏ tới file đó.

Chúng ta sẽ dùng phương thức getSharedPreferences() khi muốn lưu dữ liệu trong nhiều file khác nhau, phương thức này nhận vào tên file và chế độ đọc. Ví dụ:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | package com.phocode;  import android.content.SharedPreferences;  import android.app.Activity;  import android.os.Bundle;  public class MainActivity extends Activity  {  @Override  public void onCreate(Bundle savedInstanceState)  {  super.onCreate(savedInstanceState);  setContentView(R.layout.main);  SharedPreferences sharedPref = getSharedPreferences("save",  Context.MODE\_PRIVATE);  }  } |

Chế độ MODE\_PRIVATE tức là file này chỉ có thể mở được từ ứng dụng củachúng ta. File save.xml sẽ được tạo ra nếu chưa có.

Trong bài nhóm em dùng phương thức SharedPreferences

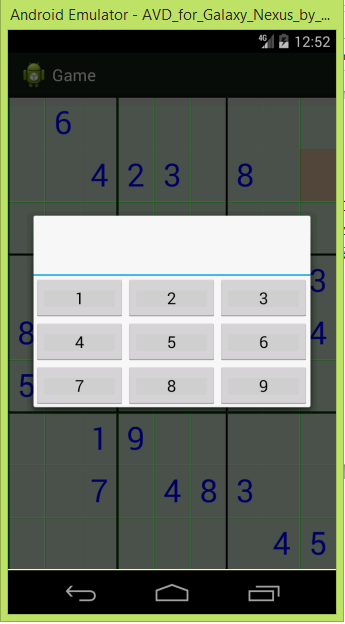
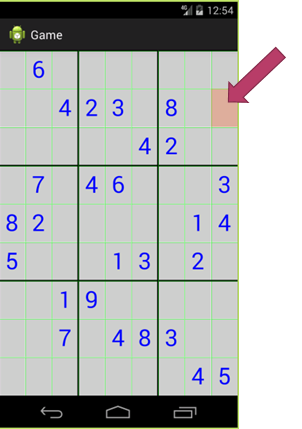
#### 

#### Ghi dữ liệu lên đối tượng SharedPreferences

Để ghi dữ liệu lên các file ưu tiên này thì chúng ta tạo một đối tượng SharedPreferences.Editor từ phương thức SharedPreferences.edit().

Lớp SharedPreferences cung cấp các phương thức ghi các kiểu dữ liệu thông thường như putInt() để ghi một giá trị số nguyên, putFloat() ghi số thực, putString() ghi chuỗi... các phương thức này nhận vào khóa kèm theo giá trị. Sau khi gọi các phương thức này chúng ta gọi phương thức commit() thì dữ liệu mới thực sự được ghi vào file.

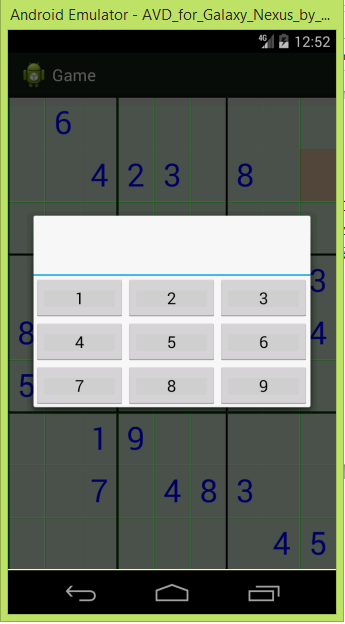
Khi click vào 1 ô bất kỳ để nhập số 1 activity làm chức năng dialog hiện lên Acivity mới lên làm che khuất một phần Activity cũ (vẫn nhìn thấy Activity cũ) hoặc thoát activity cũ sẽ : Sảy ra sự kiện onPause với Activity cũ



Khi click vào 1 ô trống

Activity KeyPad

Hay khi thoát ứng dụng hay ấn nút qoay lại



sự kiện onPause  xảy ra

Trong Sự kiện onPause ta sẽ lưu dữ liệu ta đang chơi

Code đã thay đổi:

protected void onPause()

{

super.onPause();

getSharedPreferences("mydata", MODE\_PRIVATE).edit().putString(PREF\_PUZZLE, toPuzzleString(puzzle)).commit();

getSharedPreferences("mydata1", MODE\_PRIVATE).edit().putString(Ori, toPuzzleString(originpuzzle)).commit();

}

Trong đó:

* mydata/mydata1 là tên file lưu nếu chưa có file mydata.xml sẽ được tạo
* MODE\_PRIVATE là chế độ lưu tức là file này chỉ có thể mở được từ ứng dụng củachúng ta
* PREF\_PUZZLE/ Ori là khóa
* toPuzzleString(puzzle): puzzle là dữ liệu cần lưu (mảng thao tác trực tiếp- thêm, sửa,xóa số). Sở dĩ dùng hàm này để chuyển chuỗi puzzle đang ở dạng mảng số (puzzle đang ở dạng mảng số để vẽ lên bàn chơi bằng class PuzzleView) sang chuỗi ký tự thì mới lưu được.
* toPuzzleString(originpuzzle): originpuzzle là mảng dữ liệu cần lưu (mảng này là mảng gốc dùng để lưu các gợi ý-các giá trị không thể thay đổi)
* .edit(). Là phương thức khởi tạo
* .putString Đưa dữ liệu
* .commit() là phương thức cho phép bắt đầu lưu vào file có phương thức này thì dữ liệu mới được lưu vào file
* Để tránh trong quá trình chơi có thể thay đổi số gợi ý (số mặc định) chúng em đưa ra giải pháp : có 2 mảng 81 phần tử. 1 mảng để thao tác dữ liệu 1 mảng để lưu số mặc định. Khi thay đổi 1 số trong ô nào đó sẽ kiểm tra xem số thay đổi có phải số gợi ý mặc định hay không( puzzle[i] có bằng originpuzzle[i] hay không ) nếu bằng sẽ ko thay đổi được

Code cũ:

protected void onPause()

{

super.onPause();

getPreferences(MODE\_PRIVATE).edit().putString(PREF\_PUZZLE, toPuzzleString(puzzle)).commit();

}

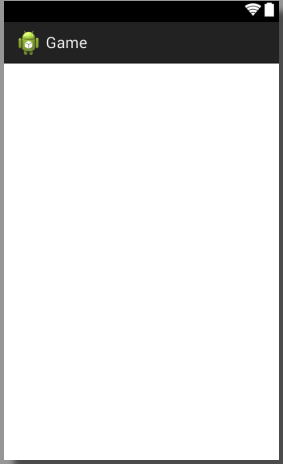
#### Đọc dữ liệu từ đối tượng SharedPreferences

Để đọc dữ liệu thì chúng ta gọi các phương thức như getInt(), getFloat(), getString()... từ đối tượng SharedPreferences.

Khi vào lại ứng dụng sau khi đã thoát game:



Hay Click button CONTINUE để có thể chơi tiếp:



Dùng **Intent** để mở Activity game

continuebutton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

intent = new Intent(MainActivity.this, Game.class);

}

});

Đến đây ta mới đọc dữ liệu từ file lưu trữ PREF\_PUZZLE bằng phương thức getString()... để sau đó là các công việc vẽ bảng đang chơi dở của class **PuzzleView.java**

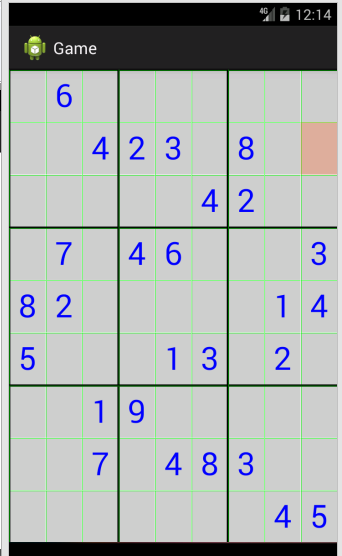
Code đã thay đổi:

String puz = getSharedPreferences("mydata", MODE\_PRIVATE).getString(PREF\_PUZZLE, easyPuzzle);

String ori = getSharedPreferences("mydata1", MODE\_PRIVATE).getString(Ori, easyPuzzle);

Code cũ:

getPreferences(*MODE\_PRIVATE).getString(PREF\_PUZZLE, easyPuzzle);*

Trong đó:

* Mydata/mydata1 là file cần lấy dữ liệu
* MODE\_PRIVATE tức là file này chỉ có thể mở được từ ứng dụng củachúng ta
* PREF\_PUZZLE/ Ori là khóa cần lấy dữ liệu
* easyPuzzle:nếu không có dữ liệu thì trả về giá trị mặc định là mảng easyPuzzle(mảng này được em cho trước, được khai báo trong class game-activity game)

Kết quả