Programming Language Laboratory

Programação para a Plataforma Android – Aula 10

Armazenamento Persistente de Dados

- Como armazenar as opções de preferência de uma aplicação?
- Como tocar músicas em uma atividade Android?
- Como armazenar dados usando *Bundles?*
- Como usar o sistema de arquivos do aparelho celular?
- Como manipular arquivos de texto em Java?
- Como armazenar objetos diretamente em arquivos?

Armazenamento Permante

- Dados armazenados permanentemente não desaparecem quando a aplicação termina sua execução.

Em que situações precisamos de armazenamento persistente?

Pense em aplicações Android que usam persistência.

Armazenamento Permante

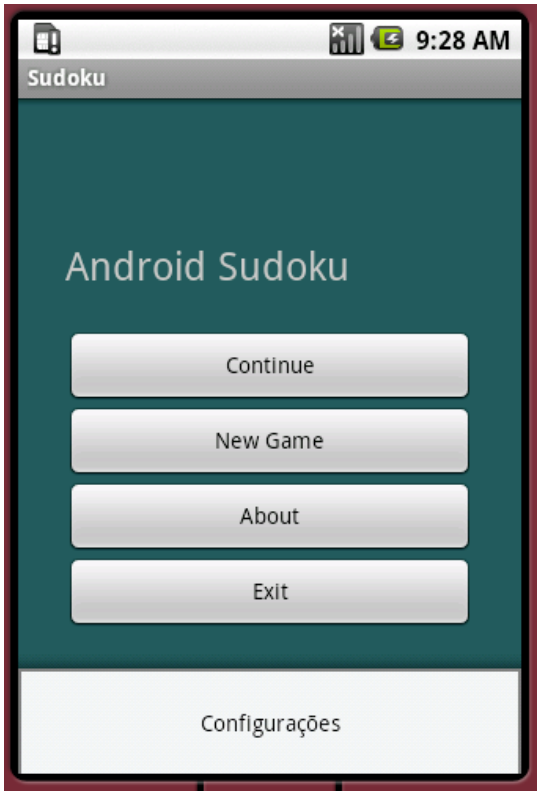
- Dados armazenados permanentemente não desaparecem quando a aplicação termina sua execução.
  - Guardar a lista de compras;
  - Armazenar a lista de endereços;
  - Lembrar a posição das peças de xadrez no tabuleiro;
  - Armazenar as opções escolhidas para uma aplicação;
  - etc

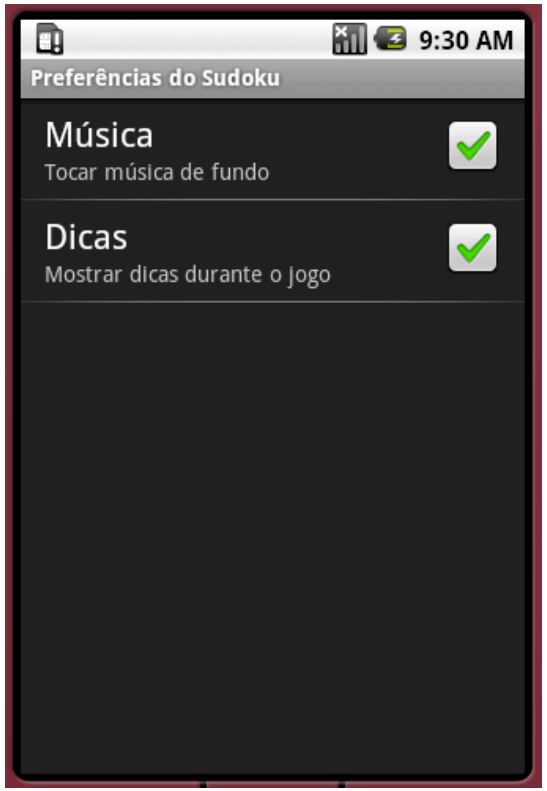
Muitas formas de Armazenamento

- A API de preferências
- Estados de “*Bundles*”
- Arquivos de armazenamento em memória *flesh*
- Etc

Preferências

- Nosso Sudoku possui um menu de opções.
  - Você lembra como isso foi implementado?





Preferências

Sudoku.java

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case R.id.settings:
        startActivity(new Intent(this, Prefs.class));
        return true;
    }
    return false;
}
```

Modifique a classe Prefs para que ela armazene e retorne as preferências escolhidas pelo usuário.

O que **esse** método está fazendo?

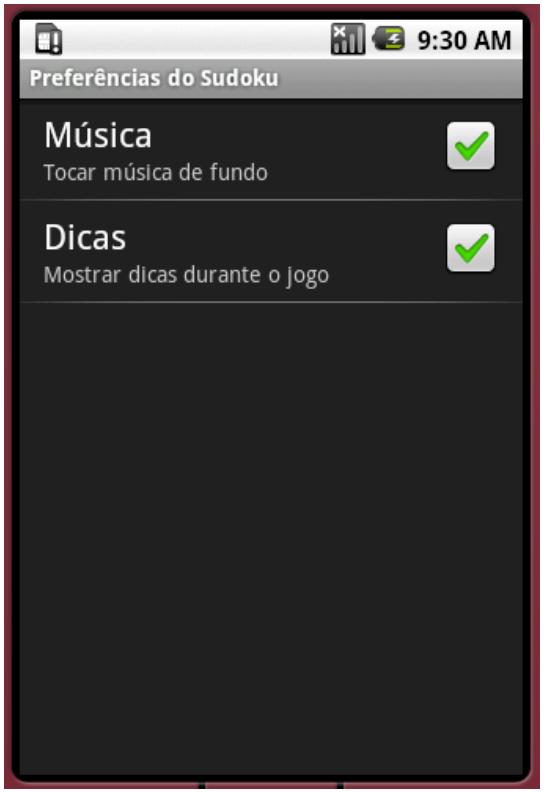
```
public class Prefs extends PreferenceActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    addPreferencesFromResource(R.layout.settings);
  }
}
```

Prefs.java

Preferências

settings.java

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen
 xmlns:android="http://schemas.android.com/apk/res/android">
 <CheckBoxPreference
  android:key="music"
  android:title="@string/music_title"
  android:summary="@string/music_summary"
  android:defaultValue="true" />
 <CheckBoxPreference
  android:key="hints"
  android:title="@string/hints_title"
  android:summary="@string/hints_summary"
  android:defaultValue="true" />
</PreferenceScreen>
```




```
public class Prefs extends PreferenceActivity {
 @Override
 protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  addPreferencesFromResource(R.layout.settings);
 }
}
```

Prefs.java

Armazenamento de Preferências

```
public class Prefs extends PreferenceActivity {
  private static final String OPT_MUSIC = "music";
  private static final boolean OPT_MUSIC_DEF = true;
  private static final String OPT_HINTS = "hints";
  private static final boolean OPT_HINTS_DEF = true;
  ...
  public static boolean getMusic(Context context) {
    return PreferenceManager.getDefaultSharedPreferences
      (context).getBoolean(OPT_MUSIC, OPT_MUSIC_DEF);
  }
  public static boolean getHints(Context context) {
    return PreferenceManager.getDefaultSharedPreferences
      (context).getBoolean(OPT_HINTS, OPT_HINTS_DEF);
  }
}
```

Como dá-se a comunicação entre o que "entra" nesse banco de dados e o que "sai" dele?

Como "aplicar" essas preferências?

Prefs.java

Music.java

Tocar ou não tocar uma música?

```
public static void play(Context context, int resource) {
    stop(context);
    mp = MediaPlayer.create(context, resource);
    mp.setLooping(true);
    mp.start();
}
```

A class Prefs comporta-se como um singleton.

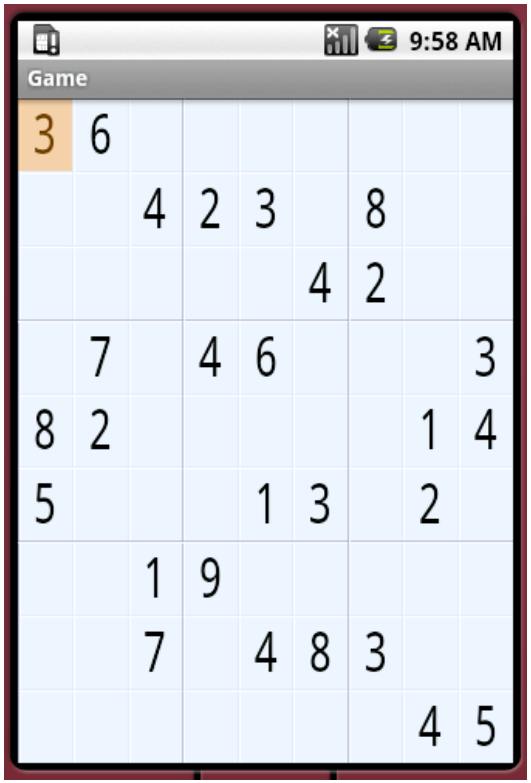
```
public static void play(Context context, int resource) {
    stop(context);
    if (Prefs.getMusic(context)) {
        mp = MediaPlayer.create(context, resource);
        mp.setLooping(true);
        mp.start();
    }
}
```

E como habilitar ou desabilitar as "dicas"?

Removendo as dicas

PuzzleView.java

```
if (Prefs.getHints(getContext())) {
    Paint hint = new Paint();
    int c[] = { getResources().getColor(R.color.puzzle_hint_0),
          getResources().getColor(R.color.puzzle_hint_1),
          getResources().getColor(R.color.puzzle_hint_2), };
    Rect r = new Rect();
    for (int i = 0; i < 9; i++) {
        for (int j = 0; j < 9; j++) {
            int movesleft = 9 - game.getUsedTiles(i, j).length;
            if (movesleft < c.length) {
                getRect(i, j, r);
                hint.setColor(c[movesleft]);
                canvas.drawRect(r, hint);
            }
        }
    }
}
```



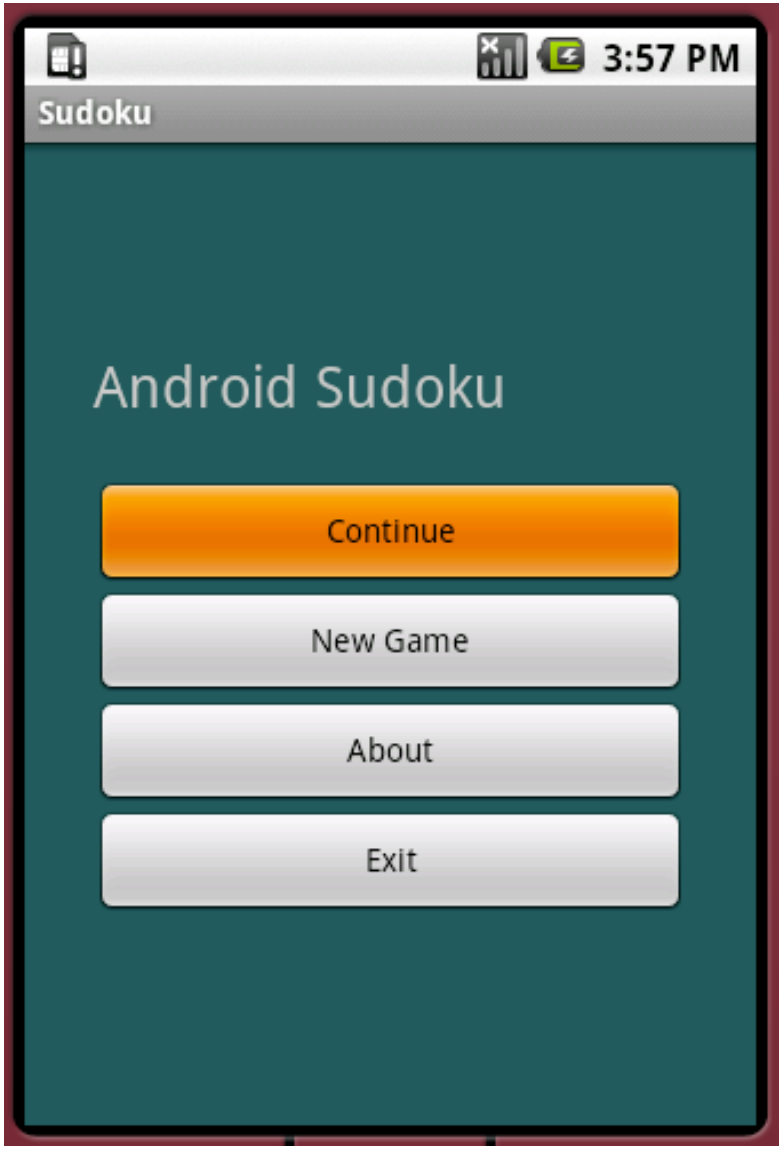
Continuando um Jogo

A opção continue

Modifique a nossa implementação de Sudoku para que seja possível interromper e continuar um jogo interrompido.

Seria possível usar a API de preferências?

No fundo, trata-se de uma tabela hash persistente!



Game.java

Como um jogo novo é obtido

```
private int[] getPuzzle(int diff) {
  String puz;
  switch (diff) {
  case DIFFICULTY_HARD:
    puz = hardPuzzle;
    break;
  case DIFFICULTY_MEDIUM:
    puz = mediumPuzzle;
    break;
  case DIFFICULTY_EASY:
  default:
    puz = easyPuzzle;
    break;
  }
  return fromPuzzleString(puz);
}
```

Podemos adicionar uma nova opção, para que seja possível escolhermos o puzzle a partir da tabela de preferências!

Como fazer isso?

```
private int[] getPuzzle(int diff) {
  String puz;
  switch (diff) {
 case DIFFICULTY_CONTINUE:
    puz = getPreferences(MODE_PRIVATE).getString(PREF_PUZZLE,
    easyPuzzle);
    break;
  case DIFFICULTY_HARD:
    puz = hardPuzzle;
    break;
  case DIFFICULTY_MEDIUM:
    puz = mediumPuzzle;
    break;
  case DIFFICULTY_EASY:
  default:
    puz = easyPuzzle;
    break;
  }
  return fromPuzzleString(puz);
}
```

Game.java

Que constante é essa?

E como colocar o estado do tabuleiro na tabela de preferências?

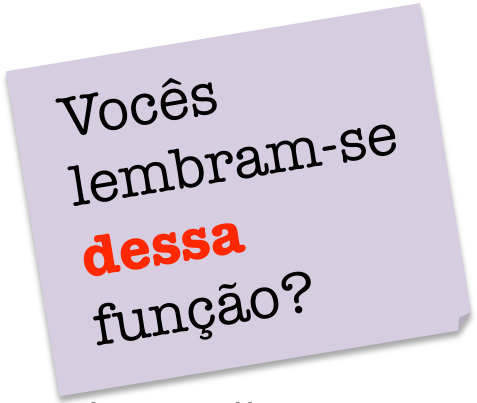
Procure pensar nos eventos necessários para essa tarefa

Quando o puzzle deve ser salvo?

onPause

Game.java

```
public class Game extends Activity {
  private static final String PREF_PUZZLE = "puzzle" ;
  …
  @Override
  protected void onPause() {
    super.onPause();
    Log.d(TAG, "onPause");
    Music.stop(this);
    getPreferences(MODE_PRIVATE).edit().putString(PREF_PUZZLE,
      toPuzzleString(puzzle)).commit();
  }
  …
}
```



toPuzzleString

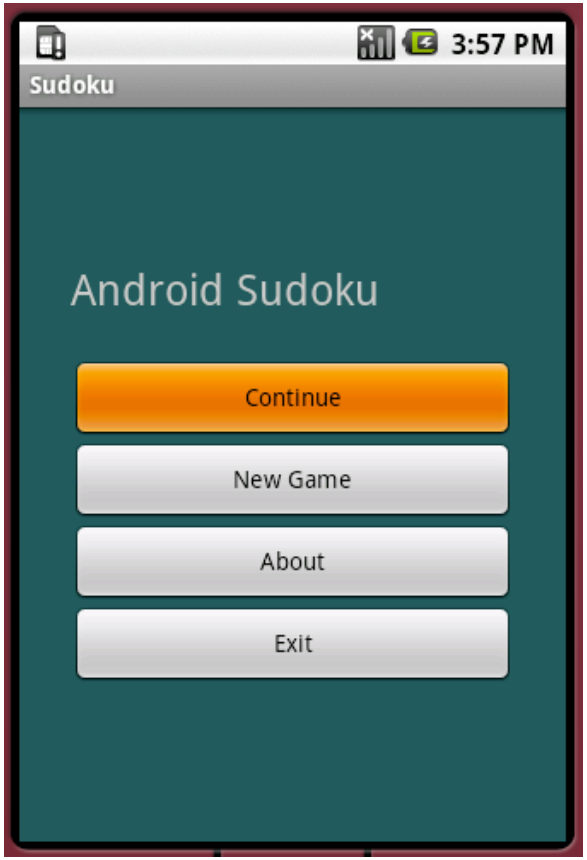
Game.java

```
static private String toPuzzleString(int[] puz) {
  StringBuilder buf = new StringBuilder();
  for (int element : puz) {
    buf.append(element);
  }
  return buf.toString();
}
```

E como podemos começar um jogo que havia sido interrompido?

Em outras palavras, como tratar o botão continue?

Onde está o código que lida com eventos de clique dessa atividade?

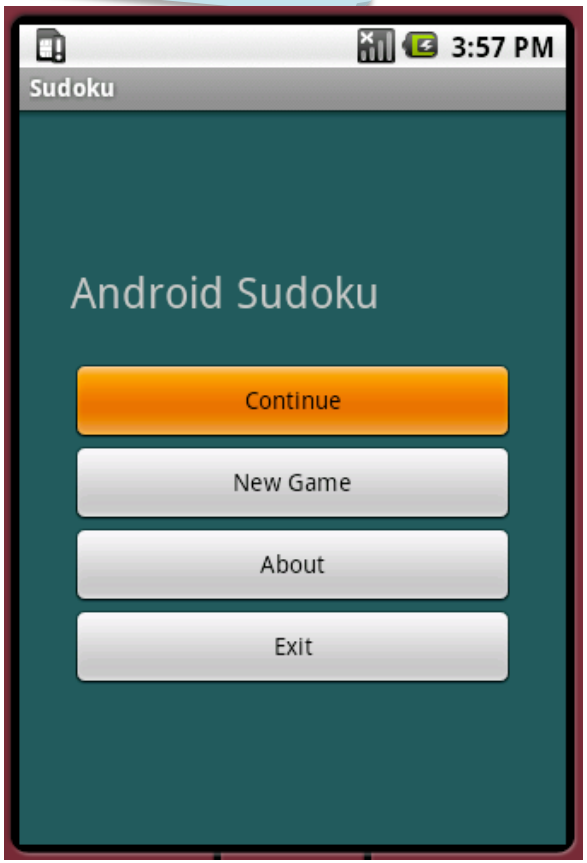


Eventos de Clique

Sudoku.java

```
public void onClick(View v) {
  switch (v.getId()) {
  case R.id.about_button:
    Intent i = new Intent(this, About.class);
    startActivity(i);
    break;
  case R.id.new_button:
    openNewGameDialog();
    break;
  case R.id.exit_button:
    finish();
    break;
  }
}
```

E como alterar essa classe para que o botão continue possa funcionar?

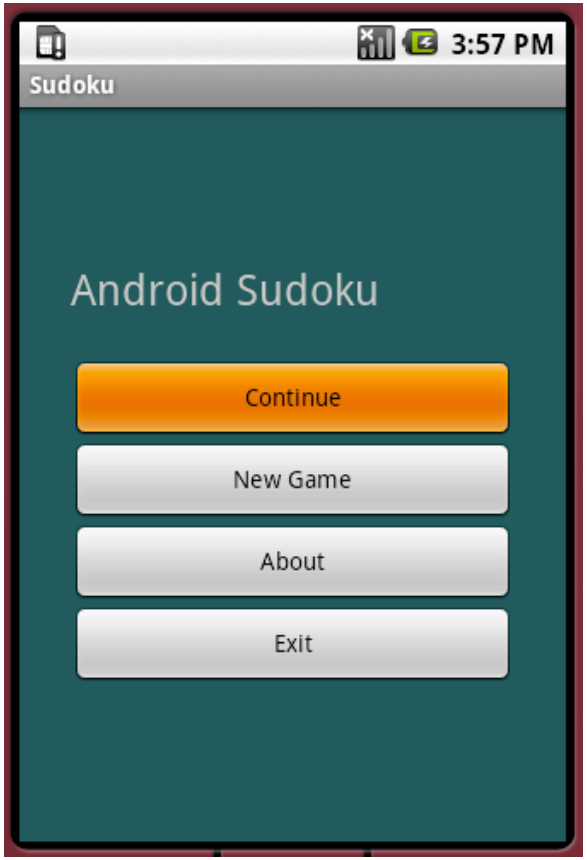


Eventos de Clique

Sudoku.java

```
public void onClick(View v) {
  switch (v.getId()) {
  case R.id.continue_button:
    startGame(Game.DIFFICULTY_CONTINUE);
    break;
  case R.id.about_button:
    Intent i = new Intent(this, About.class);
    startActivity(i);
    break;
  case R.id.new_button:
    openNewGameDialog();
    break;
  case R.id.exit_button:
    finish();
    break;
  }
}
```

Ei, mas está faltando associar o evento ao botão!



Botões e Eventos

```
@Override
 public void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.main);
  View aboutButton = findViewById(R.id.about_button);
  aboutButton.setOnClickListener(this);
  View newButton = findViewById(R.id.new_button);
  newButton.setOnClickListener(this);
  View exitButton = findViewById(R.id.exit_button);
  exitButton.setOnClickListener(this);
  View continueButton = findViewById(R.id.continue_button);
  continueButton.setOnClickListener(this);
 }
```

Orientação da Tela

- Se virarmos a tela em modo paisagem, nossa aplicação perde a informação sobre onde estava o cursor:

Como resolver esse problema?

Estado da Visão

- Toda visão possui um estado: posição do cursor, números, etc.
- Quando trocamos a visão, o método onDraw da visão é invocado.

Por que a posição dos números é salva entre modo retrato e paisagem?

E como podemos agora salvar a posição do cursor?

Bundles

- “Bundles” são tabelas usadas para passar dados entre Atividades.
  - Podem ser atividades diferentes
  - Ou a mesma atividade, no passado e no futuro.
    - Nesse caso, toda atividade usa dois métodos:
    - `protected Parcelable onSaveInstanceState()`
    - `protected void onRestoreInstanceState (Parcelable state)`

Salvando Estados

PuzzleView.java

```
@Override
protected Parcelable onSaveInstanceState() {
  Parcelable p = super.onSaveInstanceState();
  Bundle bundle = new Bundle();
  bundle.putInt(SELX, selX);
  bundle.putInt(SELY, selY);
  bundle.putParcelable(VIEW_STATE, p);
  return bundle;
}
```

É claro: convém não esquecer as **constantes**...

E como seria o método que restaura o estado da atividade?

Restaurando Estados

PuzzleView.java

```
@Override
protected void onRestoreInstanceState(Parcelable state) {
  Log.d(TAG, "onRestoreInstanceState");
  Bundle bundle = (Bundle) state;
  select(bundle.getInt(SELX), bundle.getInt(SELY));
  super.onRestoreInstanceState(bundle.getParcelable(VIEW_STATE));
  return;
}
```

Como foi amarrada essa ligação entre o estado salvo e o estado restaurado?

Falta, agora, encontrar um identificador para a visão.

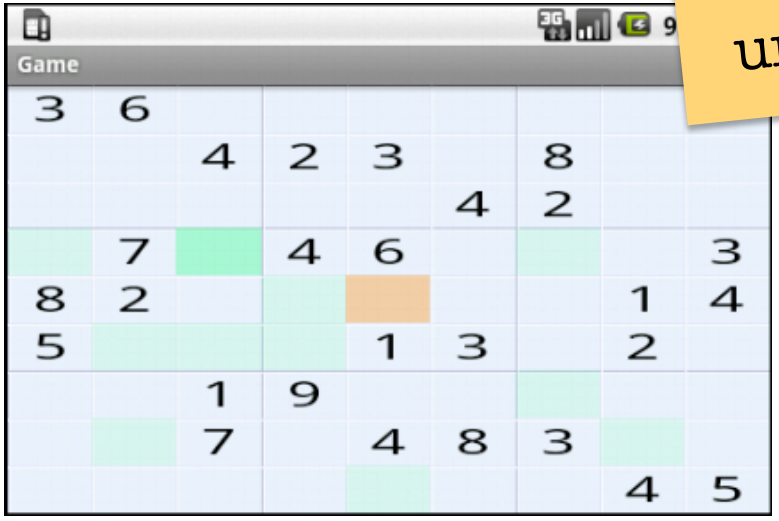
Identificadores

```
public class PuzzleView extends View {
  private static final int ID = 42;
  ...
  public PuzzleView(Context context) {
    super(context);
    this.game = (Game) context;
    setFocusable(true);
    setId(ID);
    setFocusableInTouchMode(true);
  }
  ...
}
```

E porque não colocamos o identificador de PuzzleView em um arquivo XML?

Preservando o Estado da Atividade

- Notem que apenas a posição do cursor está sendo salva. Os números no tabuleiro não estão sendo salvos.



Consertar essa lacuna fica como um exercício.

Mas isso não é fácil: a visão não manipula puzzles!

O Sistema de Arquivos

- Android OS é Linux.
  - E portanto, possui um sistema de arquivos.
- Arquivos podem ser manipulados pelas classes na biblioteca `java.io`.
- Cada aplicação possui seu próprio espaço.
  - Normalmente `data/data/nome_pacote`
- E a classe Context possui métodos para manipular os arquivos armazenados nesse espaço.

Manipulação de Arquivos

- A classe `Context` possui diversos métodos para manipular arquivos:
  - `deleteFile`: apaga um arquivo e retorna verdadeiro caso a deleção tenha acontecido.
  - `fileList`: retorna um arranjo de strings com o nome dos arquivos no espaço da aplicação.
  - `openFileInput`: abre um arquivo para leitura.
  - `openFileOutput`: abre um arquivo para escrita.

Arquivos de Texto

- Podemos armazenar qualquer tipo de arquivo como “recursos crus”, em res/raw/

Exibição de texto

Implemente uma classe que abra, leia e exiba o conteúdo de um arquivo de texto armazenado em res/raw/file1.txt

Como fazer isso em Java, versão desktop?

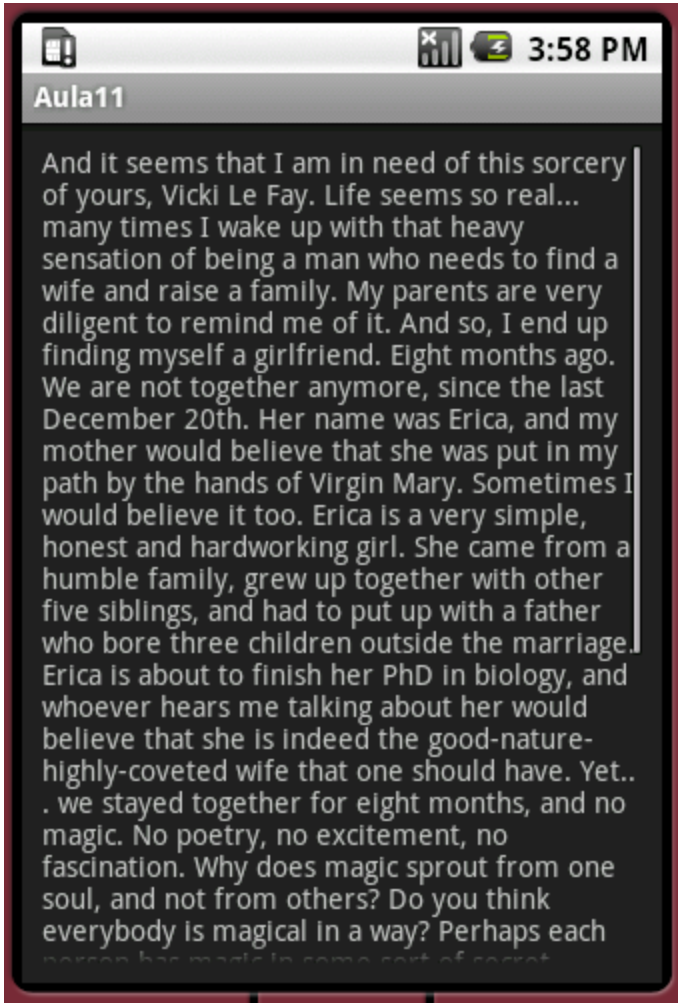
Como deveria ser o layout dessa aplicação?

E haveria alguma diferença em Android?

ScrollLayout

Como ler o conteúdo do arquivo?

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   >
  <ScrollView
   xmlns:android=
     "http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:padding="10dip">
   <TextView
     android:id="@+id/textView"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content" />
  </ScrollView>
</LinearLayout>
```

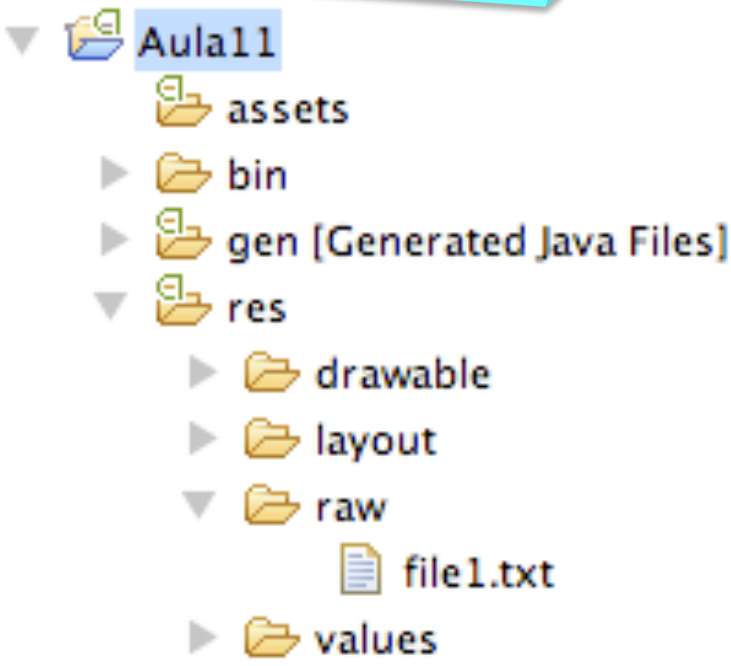


Lendo um Arquivo

AulaActivity10.java

```
private String readText() {
  InputStream inputStream = getResources().openRawResource(R.raw.file1);
  ByteArrayOutputStream byteArrayOutputStream =
      new ByteArrayOutputStream();
  int i;
  try {
    i = inputStream.read();
    while (i != -1) {
      byteArrayOutputStream.write(i);
      i = inputStream.read();
    }
    inputStream.close();
  } catch (IOException e) {
    e.printStackTrace();
  }
  return byteArrayOutputStream.toString();
}
```

Algum problema com esse método?



Bufferização

AulaActivity10.java

```
private String getTextFromFile() {
    StringBuffer contents = new StringBuffer();
    try {
        InputStream rawRes = getResources().openRawResource(R.raw.file1);
        BufferedReader input =
            new BufferedReader(new InputStreamReader(rawRes));
        String line = null;
        while ((line = input.readLine()) != null) {
            contents.append(line + '\n');
        }
    } catch (IOException ex) {
        ex.printStackTrace();
    }
    return contents.toString();
}
```

Arquivos podem ser lidos ou escritos de várias formas diferentes.

Como usar esse método?

Decoradores

```
private String getTextFromFile() {
   StringBuffer contents = new StringBuffer();
   try {
      InputStream rawRes = getResources().openRawResource(R.raw.file1);
      BufferedReader input =
         new BufferedReader(new InputStreamReader(rawRes));
      String line = null;
      while ((line = input.r
         contents.append(
      }
   } catch (IOException ex
      ex.printStackTrace()
   }
   return contents.toString();
}
```

BufferedReader intercepta todos os métodos de InputStream, impondo sobre esses métodos novos comportamentos. Esse padrão de projetos chama-se *decorador*.

Em que outras situações Decoradores são usados?

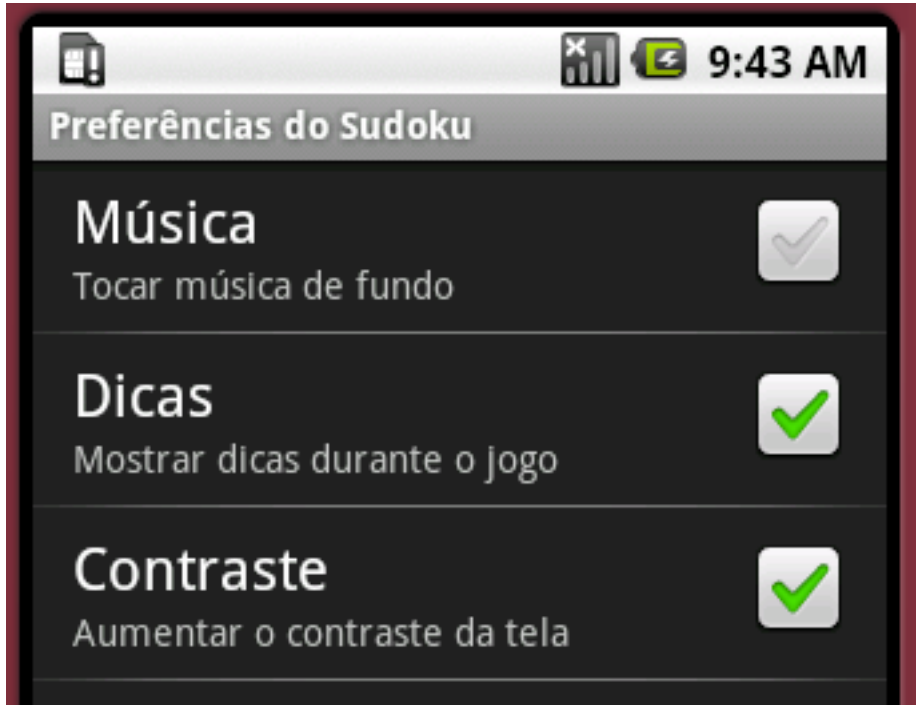
Lendo e exibindo arquivos

AulaActivity10.java

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    TextView text = (TextView) findViewById(R.id.textView);
    text.setText(getTextFromFile());
}
```

Exercícios: Contraste

- Modifique a aplicação Sudoku, para que a visão do jogo tenha dois modos: “contraste” e “sem contraste”. A visão com contraste deve usar um fundo branco, e linhas maiores negras.
- Essa opção também deve ser definida no layout de preferências, via um botão de seleção.

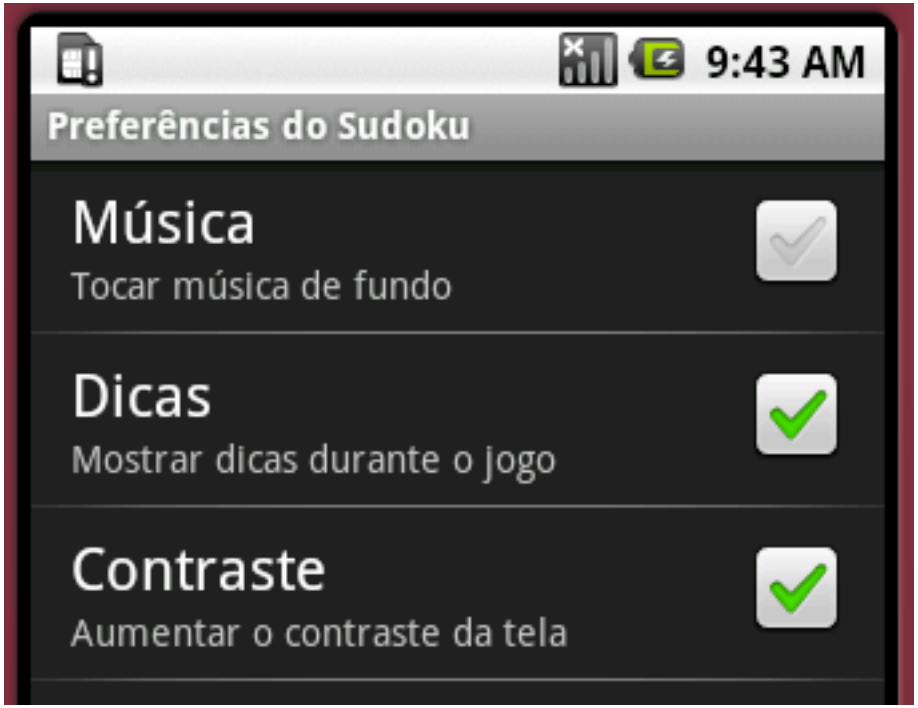


Exercícios: Contraste

- Modifique a aplicação Sudoku, para que a visão do jogo tenha dois modo[illegible] “sem contraste”. A visão com c[illegible] usar um fundo branco, e linhas [illegible] negras.

Como deve ser o novo layout de preferências?

- Essa opção também deve ser definida no layout de preferências, via um botão de seleção.



settings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
    <CheckBoxPreference
      android:key="music"
        android:title="@string/music_title"
        android:summary="@string/music_summary"
        android:defaultValue="true" />
    <CheckBoxPreference
      android:key="hints"
        android:title="@string/hints_title"
        android:summary="@string/hints_summary"
        android:defaultValue="true" />
    <CheckBoxPreference
      android:key="contrast"
        android:title="@string/constrast_title"
        android:summary="@string/constrast_summary"
        android:defaultValue="true" />
</PreferenceScreen>
```

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  ...
  <string name="contrast_title">Contraste</string>
  <string name="contrast_summary">
     Aumentar o contraste da tela
  </string>
  ...
</resources>
```

E como fica a classe Prefs.java?

Prefs.java

```
public class Prefs extends PreferenceActivity {
  // Option names and default values
  private static final String OPT_CONTRAST = "contrast";
  private static final boolean OPT_CONTRAST_DEF = true;
  ...

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    addPreferencesFromResource(R.layout.settings);
  }

  public static boolean getContrast(Context context) {
    return PreferenceManager.getDefaultSharedPreferences(context).getBoolean(
        OPT_CONTRAST, OPT_CONTRAST_DEF);
  }
}
```

Precisamos agora aplicar o contraste.

Escolhendo novas cores

- O modo de contraste deve usar linhas grandes negras, e fundo branco:

Escolhendo novas cores

- O modo de contraste deve usar linhas grandes negras, e fundo branco:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  ...
  <color name="puzzle_big_lines">#ff000000</color>
  <color name="puzzle_light_background">#ffffffff</color>
</resources>
```

E como, agora, usar as preferências?

PuzzleView.java

```
if (Prefs.getContrast(getContext())) {
    background.setColor(getResources().getColor(R.color.puzzle_light_background));
} else {
    background.setColor(getResources().getColor(R.color.puzzle_background));
}

…

if (Prefs.getContrast(getContext())) {
    dark.setColor(getResources().getColor(R.color.puzzle_big_lines));
} else {
    dark.setColor(getResources().getColor(R.color.puzzle_dark));
}
```