

# Utilizando Pure Data em softwares para Android

Antonio Deusany de Carvalho Junior  
dj@ime.usp.br

# Motivação

- Android
  - Mobilidade
  - Desempenho computacional
- Pure Data
  - Diversos tipos de usuários
  - Ampla comunidade (ativa)
- Música Ubíqua
  - “música feita por múltiplos usuários, usando uma variedade de dispositivos, tanto fixos quanto portáteis”

# Pure Data

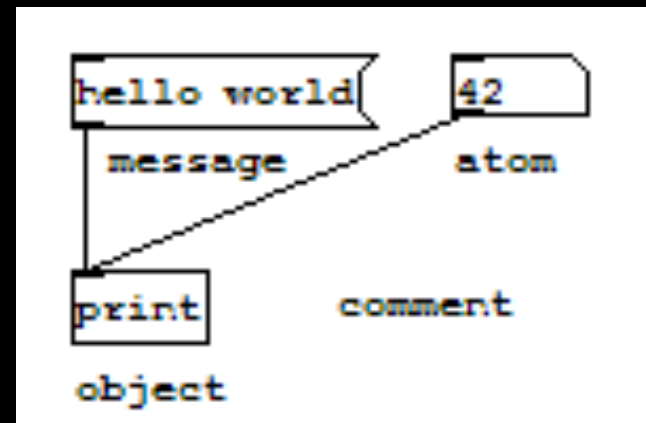
- PD
- Linguagem de programação voltada para multimídia
- Criada por Miller Puckette
- <http://puredata.info/>
- Multiplataforma

# Pure Data

- Possibilidade de criação e manipulação de:
  - Áudio
  - Vídeo
  - Gráficos OpenGL
  - Imagens
- Detalhes DSP
  - 44100 Hz
  - 1 bloco a cada 64 samples

# Pure Data

- Entidades da linguagem
  - átomos: float, símbolo, ponteiro para estruturas de dados
  - mensagens: compostas de um ou mais átomos a serem enviados para objetos
  - objetos: matemáticos, lógicos, bitwise, funções DSP, códigos externos
  - comentários



# Pure Data

- Distribuições
  - PD Vanilla
  - PD Extended
- Aulas online:
  - <http://pd-la.info/pd-media/miller-puckette-mus171-videos/>

# Pure Data

- Prática!
  - Download
    - <http://puredata.info/downloads>
  - Criar patch
  - Ativar computação de áudio



# Android+PD

- Facilidade para processamento de áudio
- Mobilidade para performances
- Alto desempenho em certos aparelhos
  - Asus PadFone 2 is a quad-core (1.5GHz Snapdragon S4) Android 4.0 Ice Cream Sandwich
- Possibilidade de interação com sensores



# Android+PD

- Como integrar:
  - libpd
    - PdAudio
    - PdService
  - Pd DroidParty

# libpd

- Biblioteca desenvolvida para integração com linguagens de programação que suportam código nativo como C, Java, Objective-C, Python (incluindo PyGame), Lua
- Utiliza Pd Vanilla (é possível utilizar *externals*)
- <http://libpd.cc/>
- <https://github.com/libpd>

# libpd

- Passos para uso:
  - Initialize the audio components.
  - Create a dispatcher and register it with PdBase.
  - Add listeners, if any.
  - Load your patch or patches.
  - Start the audio components.

# libpd

- Métodos básicos (send/receive):
  - static int sendBang(String receiver);
  - static int sendFloat(String receiver, float value);
  - static int sendSymbol(String receiver, String symbol);
  - static int sendList(String receiver, Object... list);
  - static int sendMessage(String receiver, String message, Object... list);

# libpd

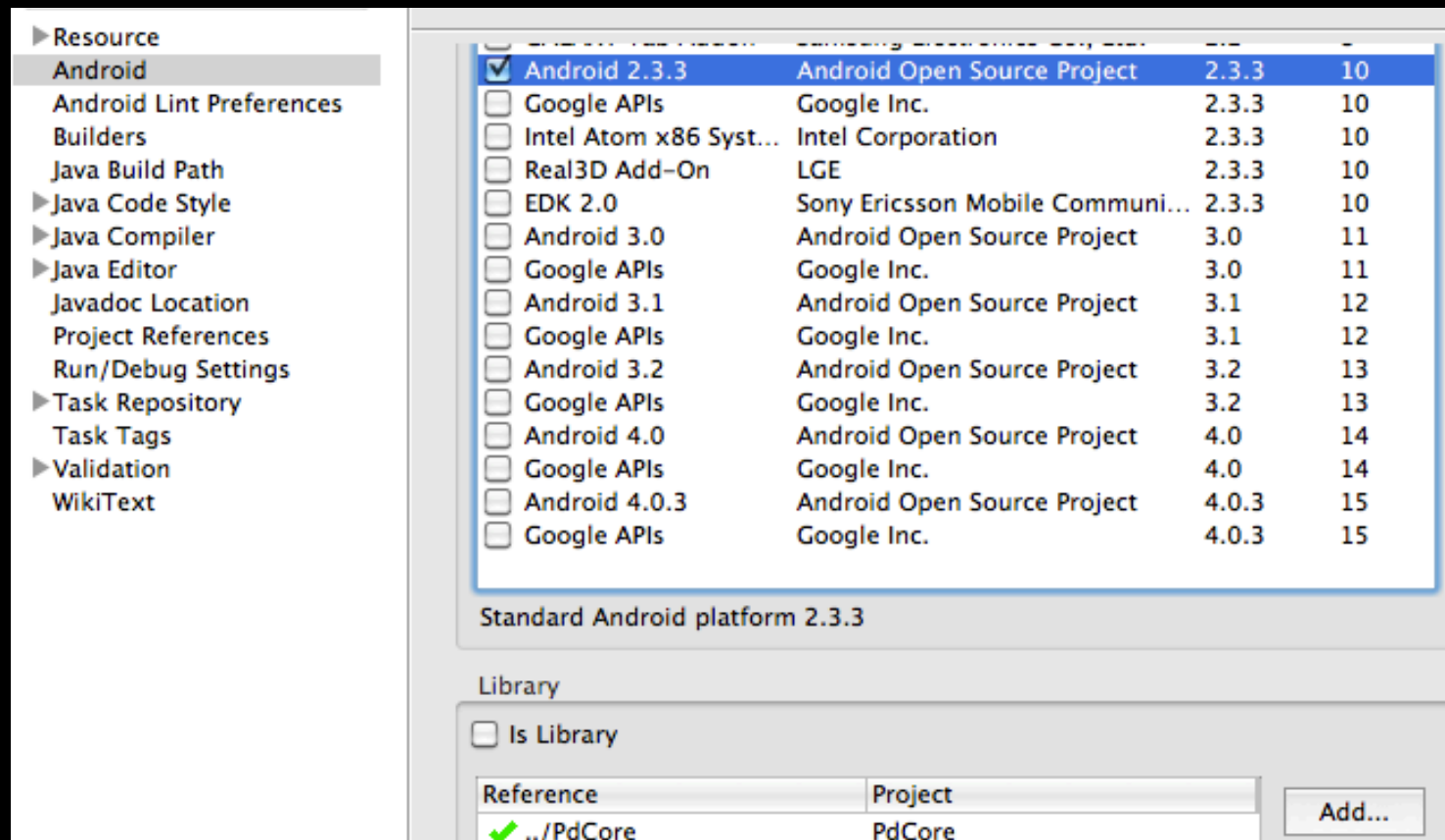
- Como integrar:
  - `git clone git://github.com/libpd/pd-for-android.git`
  - `$ cd pd-for-android`
  - `$ git submodule init`
  - `$ git submodule update`

# libpd

- Como integrar:
  - PdAudio
    - Condiciona a thread de audio à activity
  - PdService
    - Controla a thread de audio em background como um serviço, compartilhando entre *activities* e funcionando quando o aplicativo é pausado ou quando o aplicativo é colocado em background

# libpd

- Configurando projeto no Eclipse



# libpd

- PdAudio

```
private PdUiDispatcher dispatcher;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    initGui();
    try {
        initPd();
        loadPatch();
    } catch (IOException e) {
        Log.e(TAG, e.toString());
        finish();
    }
}
```



# libpd

- PdAudio

```
private void initPd() throws IOException {  
    // Configure the audio glue  
    int sampleRate = AudioParameters.suggestSampleRate();  
    PdAudio.initAudio(sampleRate, 0, 2, 8, true);  
    // Create and install the dispatcher  
    dispatcher = new PdUiDispatcher();  
    PdBase.setReceiver(dispatcher);  
}
```

# libpd

- PdAudio
  - Obs: Adicionar patch zipado na pasta /res/raw

```
private void loadPatch() throws IOException {  
    File dir = getFilesDir();  
    IoUtils.extractZipResource(  
        getResources().openRawResource(R.raw.patch), dir, true);  
    File patchFile = new File(dir, "patch.pd");  
    PdBase.openPatch(patchFile.getAbsolutePath());  
}
```

# libpd

- PdAudio

```
protected void onResume() {  
    super.onResume();  
    PdAudio.startAudio(this);  
}  
protected void onPause() {  
    super.onPause();  
    PdAudio.stopAudio();  
}  
public void onDestroy() {  
    super.onDestroy();  
    PdAudio.release();  
    PdBase.release();  
}
```

# libpd

- PdAudio
  - Prática!
    - Baixar Theremin

# libpd

- PdService

```
private PdUiDispatcher dispatcher;
private PdService pdService = null;
private final ServiceConnection pdConnection = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {
        pdService = ((PdService.PdBinder)service).getService();
        try {
            initPd();
            loadPatch();
        } catch (IOException e) { Log.e(TAG, e.toString());
            finish();
        }
    }
    @Override
    public void onServiceDisconnected(ComponentName name) { }
};
```

# libpd

- PdService

```
/** Called when the activity is first created. */  
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    bindService(new Intent(this, PdService.class),  
                pdConnection,  
                BIND_AUTO_CREATE);  
    initSystemServices();  
}
```

# libpd

- PdService

```
private void initPd() throws IOException {  
    // Configure the audio glue  
    int sampleRate = AudioParameters.suggestSampleRate();  
    pdService.initAudio(sampleRate, 1, 2, 10.0f);  
    pdService.startAudio();  
    start();  
    // Create and install the dispatcher  
    dispatcher = new PdUiDispatcher();  
    PdBase.setReceiver(dispatcher);  
}
```

# libpd

- PdService

```
private void start() {  
    if (!pdService.isRunning()) {  
        Intent intent = new Intent(PdActivity.this,  
                                   PdActivity.class);  
        pdService.startAudio(intent,  
                              R.drawable.icon,  
                              "PdActivity",  
                              "Return to PdActivity.");  
    }  
}
```



# libpd

- PdService

```
private void initSystemServices() {
    TelephonyManager telephonyManager =
        (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);
    telephonyManager.listen(new PhoneStateListener() {
        @Override
        public void onCallStateChanged(int state, String incomingNumber) {
            if(pdService == null) return;
            if(state == TelephonyManager.CALL_STATE_IDLE) {
                start();
            } else {
                pdService.stopAudio();
            }
        }
    }, PhoneStateListener.LISTEN_CALL_STATE);
}
```

# libpd

- PdService

```
private void loadPatch() throws IOException {
    // Hear the sound
    File dir = getFilesDir();
    File patchFile;
    IoUtils.extractZipResource(
        getResources().openRawResource(R.raw.patch), dir, true);
    patchFile = new File(dir, "patch.pd");
    PdBase.openPatch(patchFile.getAbsolutePath());
}
@Override
public void onDestroy() {
    super.onDestroy();
    unbindService(pdConnection);
}
```

# libpd

- PdService
  - Prática!
    - Baixar PDaIFx

# DroidParty

- Integra o aplicativo com seu patch em PD
- <http://droidparty.net/>
- Android apk que interpreta gui's de PD
- Com abstrações para melhorar a interface
- Pode ser utilizado para desenvolvimento de app
- Código no bazaar:
  - `bzr co http://droidparty.net/PdDroidParty`

# DroidParty

- Abstrações
  - [wordbutton] - It's like a bang, but on the device it looks like a box with a word in it.
  - [numberbox] - It's like a numberbox, but on the device it stretches to fit a rectangle. (easier layout).
  - [taplist] - Lets the user tap through a list of symbols/floats.
  - [touch] - Kaospad style 2d touch surface.
  - [menubang] - Puts an entry into the menu which sends a bang when the user selects it. Optional icon is called menubang-xxxx.png where xxxx is the name of your menubang.
  - [loadsave] - Wraps and openpanel/savepanel type of interface into a single abstraction that lets the user specify or choose filenames using the Android interface.

# DroidParty

- Como utilizar:
  - Baixar Apk
  - Adicionar patch no sdcard com o nome:
    - droidparty\_main.pd
  - Rodar o DroidParty
    - Obs.: o nome da pasta do patch será o nome dele

# DroidParty

- Prática!
  - Baixar Droid Party
  - Baixar Drum-machine

# Referências

- Aulas de PD
  - <http://pd-la.info/pd-media/miller-puckette-mus171-videos/>
- Livro básico
  - <http://libpd.cc/read-the-book/>



# Método extra

**boolean android.app.[ActivityManager.isUserAMonkey\(\)](#)**

**public static boolean isUserAMonkey ()**

Since: [API Level 8](#)

Returns "true" if the user interface is currently being messed with by a monkey.

Obrigado!

= )

# Utilizando Pure Data em softwares para Android

Antonio Deusany de Carvalho Junior  
dj@ime.usp.br