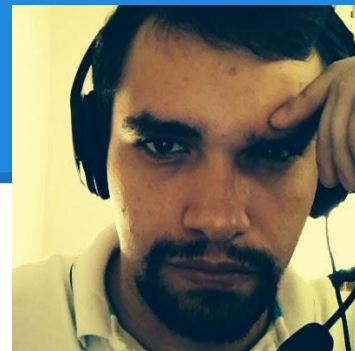


# Android é meu arcade

Daniel Monteiro



# Quem sou eu?

## Daniel Monteiro

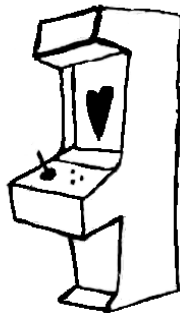
- Bacharel em Ciência da Computação pela UFF.
- ...onde existe um movimento forte de criadores de jogos independentes.
- Participei da pré-história dos jogos para celular.
- Entusiasta de Android desde 2009.
- Engines e bibliotecas são úteis e importantes - mas bem menos divertidas.
- Conhecer o que há por baixo nunca deixará de ser **fundamental**.



# Androids e Arcades?

Jogos para celular não são nenhuma novidade.  
...mas o que o Android tem para oferecer?

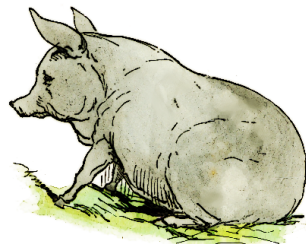
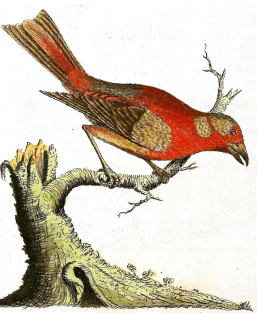
- Afinal, já tivemos (e temos) vários consoles de Video Game baseados em Android - nenhuma novidade!
- Todos carregamos no bolso uma máquina poderosa, que deixará qualquer um feliz de te ver.
- Que tal pensar em experiências hereogêneas? Um mesmo mundo de jogo que pode ser abordado de diferentes formas?
- Diferentes casos de uso para a mesma massa de dados.



# API de controles

Jogar na tela de toque só funciona para atirar pássaros raivosos em porcos.

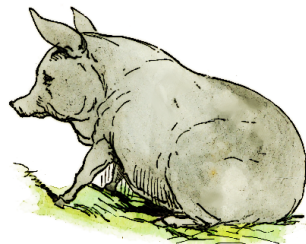
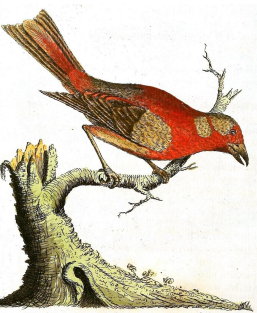
- Suporte para Joysticks USB desde a versão 3.1.
- API para controle avançado de Joysticks.
- Suporte básico muitas vezes não requer qualquer modificação no código!
- Planeje sua tela de jogo para ser navegável ou consuma os eventos de forma responsável.



# API de controles

## Determinando a existência de um Joystick USB

```
private List<Integer> getGameControllerIds() {  
    List<Integer> gameControllerDeviceIds = new ArrayList<Integer>();  
  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.GINGERBREAD) {  
  
        int[] deviceIds = InputDevice.getDeviceIds();  
        for (int deviceId : deviceIds) {  
            InputDevice dev = InputDevice.getDevice(deviceId);  
            int sources = dev.getSources();  
  
            // Verify that the device has gamepad buttons, control sticks, or both.  
            if (((sources & InputDevice.SOURCE_GAMEPAD) == InputDevice.SOURCE_GAMEPAD  
                || ((sources & InputDevice.SOURCE_JOYSTICK)  
                    == InputDevice.SOURCE_JOYSTICK)) {  
                // This device is a game controller. Store its device ID.  
                if (!gameControllerDeviceIds.contains(deviceId)) {  
                    gameControllerDeviceIds.add(deviceId);  
                }  
            }  
        }  
    }  
    return gameControllerDeviceIds;  
}
```



# Exemplo - Giovanni, o explorador

Remake de jogo que escrevi durante o Global Game Jam 2009. O jogo foi rebatizado para homenagear um sobrinho.



# Telas pequenas, grandes propositos

## Suporte a segunda tela:

- Existente de forma não oficial e limitada a bastante tempo.
- Suporte oficial, nativo desde Jellybean 4.2.
- Compatível com saídas HDMI e com Miracast.
- Suporte para telas com composição heterogenea (cada tela exibindo um conteúdo diferente).
- Fácil implementação no stock Android, não tão fácil com Support Library (mas com modelo mais robusto!).



# Telas pequenas, grandes propósitos

## A seleção de uma segunda tela

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN_MR1) {  
    mMediaRouter = (MediaRouter) getSystemService(Context.MEDIA_ROUTER_SERVICE);  
  
    mRouteInfo = mMediaRouter.getSelectedRoute( MediaRouter.ROUTE_TYPE_LIVE_VIDEO );  
  
    if ( mRouteInfo != null ) {  
  
        Display presentationDisplay = mRouteInfo.getPresentationDisplay();  
  
        if ( presentationDisplay != null ) {  
            ((ViewManager) view.getParent()).removeView( view );  
            Presentation presentation = new GamePresentation( this, presentationDisplay, view );  
            presentation.show();  
        }  
    }  
}
```



# Telas pequenas, grandes propositos

## A montagem da apresentação da segunda tela

```
@TargetApi(Build.VERSION_CODES.JELLY_BEAN_MR1)
private final static class GamePresentation extends Presentation {

    final GameView canvas;

    public GamePresentation(Context context, Display display, GameView gameView ) {
        super(context, display);

        this.canvas = gameView;
    }

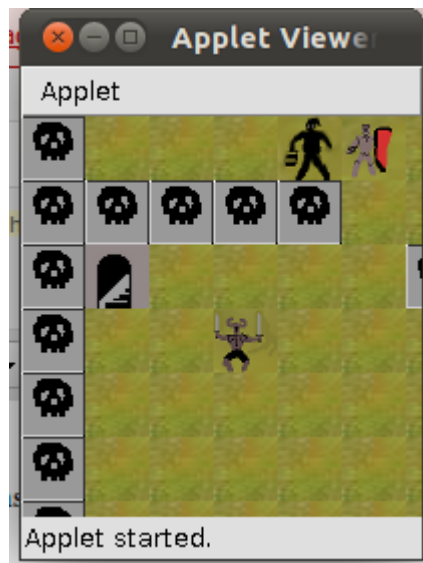
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(canvas );
    }
}
```

# Exemplo - Os Cavaleiros do Alentejo

Jogo inspirado em minhas raízes portuguesas, originalmente escrito para um GameJam em 2011, como um Java Applet.

- Ideal para API de segunda tela, por ser um jogo por turnos.
- Suporte a Joystick, que também altera o layout da tela.



# Casting - Miracast

## HDMI sem-fio:

- What's in a name? that which we call a rose/by any other name would smell as sweet.
- Planeje seu jogo para lidar com latência.
- Compatibilidade ampla com SmartTVs.
- Configuração complicada - que pode estar escondida!
- Suporte a Chromecast usando o recurso experimental de Cast Screen.
- Bom suporte para APIs gráficas, como OpenGL ES.



# Casting - Exemplo

## Demo 3D

- Demo simples de Engine de estudo, usada para estudos de algoritmos de visibilidade e streaming de dados para GPU.
- Miracast apenas transmite o fluxo de frames comprimido.
- Renderização acontece no dispositivo emissor.



# Casting - Chromecast

## Uma nova ~~esperança~~ API

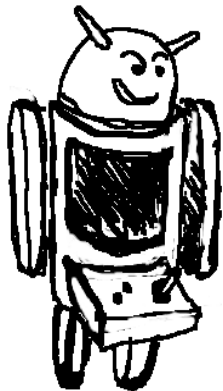
- Programa de desenvolvedores pago a parte!
- Até então, apenas HTML5!
- Canvas e WebGL são ótimos, mas quem quer escrever um novo render? E como manter a consistência visual?
- Anunciada no IO 2015, o suporte ainda é altamente experimental
- ...passou debaixo do radar da mídia? (vídeo ausente do Youtube!)
- Documentação ainda é conflitante e samples ainda tem problemas.

**Não consegui criar exemplos para a API =-(**



# Game Over

Vamos conversar!



Daniel Monteiro  
[danielmonteiro@id.uff.br](mailto:danielmonteiro@id.uff.br)  
<http://about.me/danielmonteiro>

(todas a imagens desta apresentação são de domínio público ou de divulgação de seus respectivos donos de copyright. Exceto quando elas são ruins. Aí elas são de minha autoria)

