

Desenvolvimento de Jogos para Dispositivos Móveis

Bruno Cicanci
gamedeveloper.com.br

Dispositivos Móveis



Quando as Operadoras Dominaram o Mundo



Era Pré iPhone



- Apenas operadoras publicavam jogos e aplicativos
- As empresas eram obrigadas a seguir as exigências das operadoras
- O lucro era baixo, inclusive para grandes empresas
- Milhares de aparelhos com características diferentes e sem padrões

Era Pós iPhone



- Desenvolvedores independentes podem publicar seus aplicativos e jogos
- Operadoras não influenciam nas lojas de aplicativos e jogos, apenas em *embeds*
- Empresas pequenas e grandes lucram muito neste mercado
- Poucas variações nas características dos aparelhos

iOS (iPhone, iPod Touch e iPad)



- Mesma versão para iPod Touch, iPhone e iPad (3.2+)
- HVGA, qHD, XGA
- Objective-C e C/C++ com ambiente de desenvolvimento apenas no MacOSX
- Valor da licença: \$99
- Categoria "Games" da App Store é bloqueada no Brasil
- Necessário registrar aparelho para desenvolvimento
- +425.000 apps disponíveis*

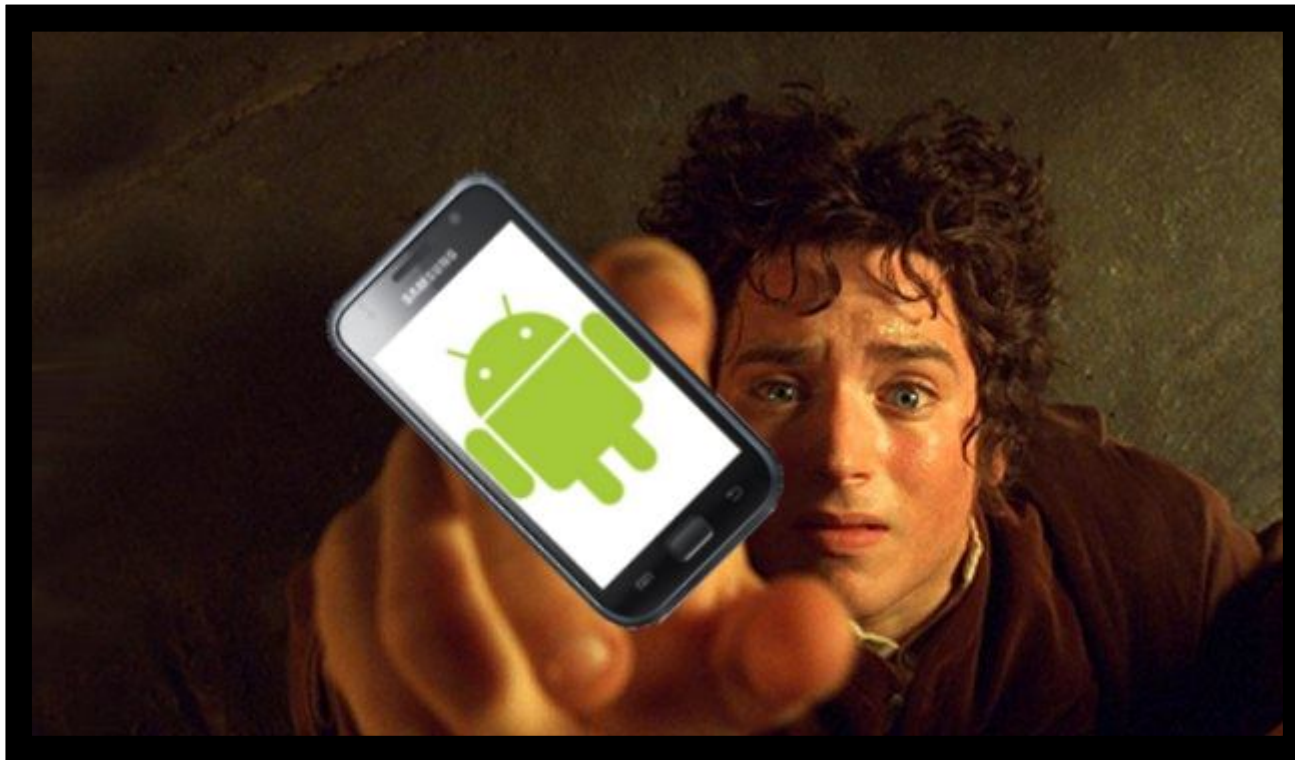
Android OS



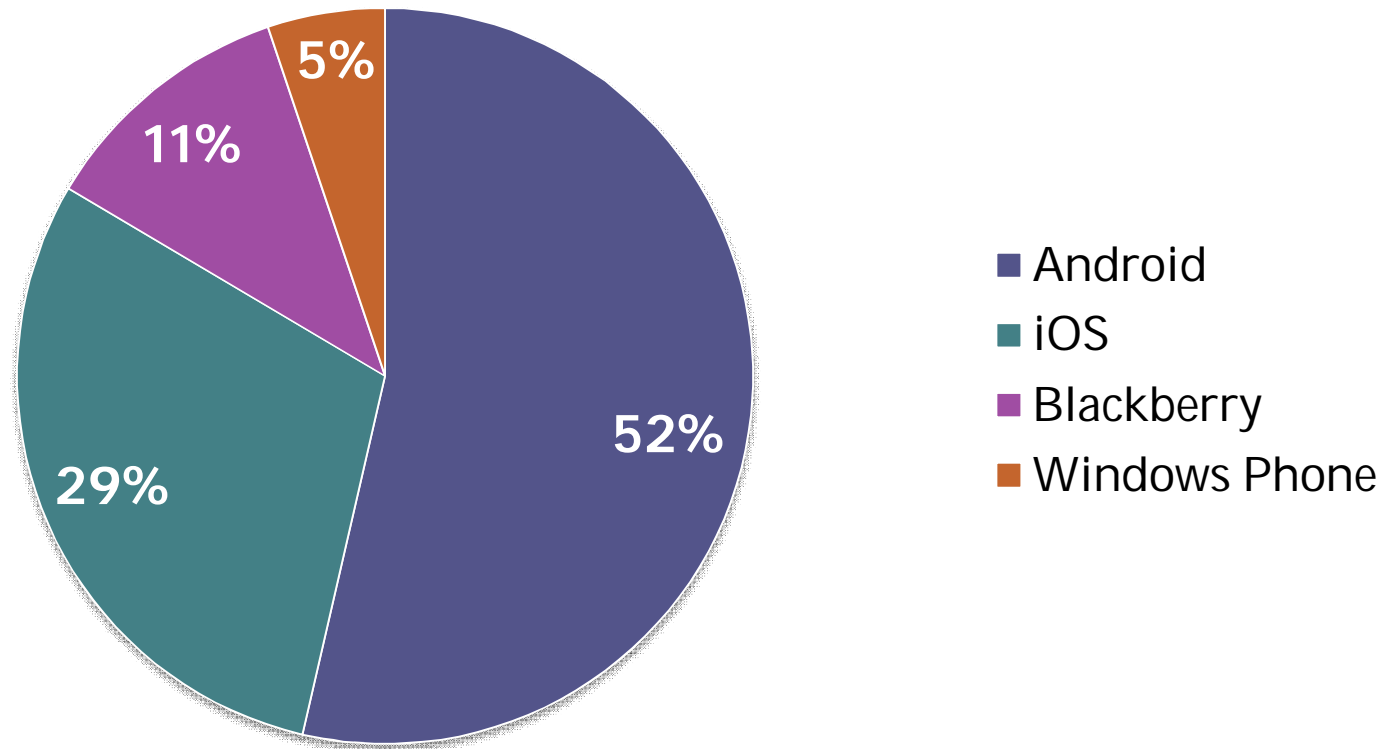
- Versões para smartphones (1.x ~ 2.3.x) diferentes das versões para tablets (3.x)
- QVGA, HVGA, WVGA, FWVGA, qHD, SVGA, WXGA
- Java e C/C++ com ambiente de desenvolvimento no Windows, Linux e MacOSX
- Valor da licença: \$25
- Qualquer aparelho pode ser usado para desenvolvimento
- +250.000 apps disponíveis*

*Fonte: develop (24/08/2011) - <http://www.develop-online.net/news/38512/Windows-Marketplace-surpasses-30000-apps>

One Build To Rule Them All



Mercado de smartphones nos EUA



Tablets no Brasil e no Mundo



- Podem rodar aplicativos de smartphones sem muitos problemas
- Atualmente quase todo fabricante de celular está produzindo tablets
- Android OS permitiu o crescimento para concorrência com a Apple
- Podem substituir netbooks nos próximos anos
- Governo planeja dar incentivo para produção no Brasil*

*Fonte: G1 (24/05/2011) - <http://g1.globo.com/tecnologia/noticia/2011/05/doze-empresas-querem-produzir-tablets-no-brasil-diz-mercadante.html>

Desenvolvimento de Jogos



Início do Projeto



- Definir um tema/conceito
- Definir características do jogo como um produto:
 - Público
 - Plataforma
 - Mercado
 - Concorrência
 - Diferenciais
- Montar um plano de negócio com foco em produto pode ser útil para organizar esta parte do projeto



Modelo de Negócio

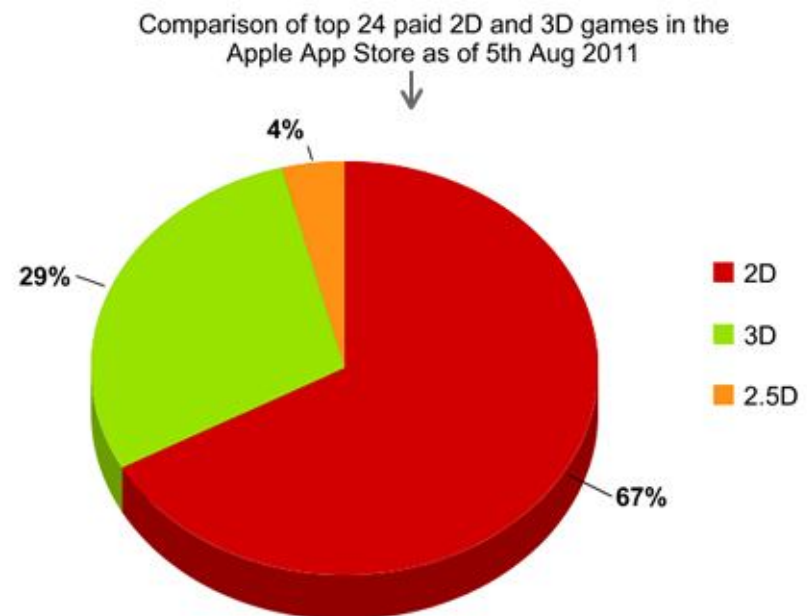
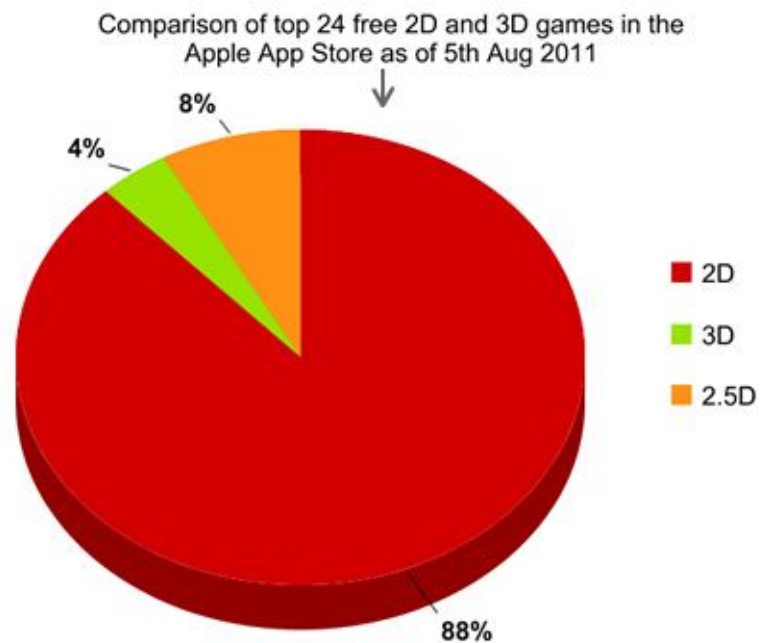
Pago

- Funciona bem com jogos de iOS, porém desde que tenha uma versão de demonstração para o jogador testar
- Jogos indie geralmente tem preços entre \$0.99 e \$2.99
- Não é legal colocar anúncios em versões de demonstração

Gratuito

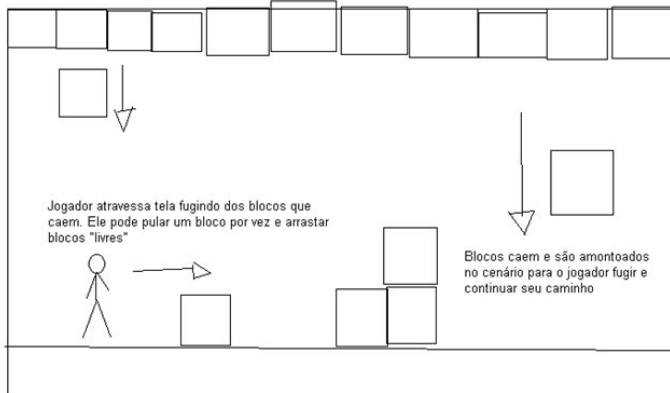
- Funciona bem com jogos de Android, pois a pirataria neste OS é maior
- Anúncios, quando colocados em locais corretos e com uma frequência adequada, podem dar um bom retorno
- Jogos Freemium são muito comuns no Android OS e iOS

2D, 2.5D ou 3D?



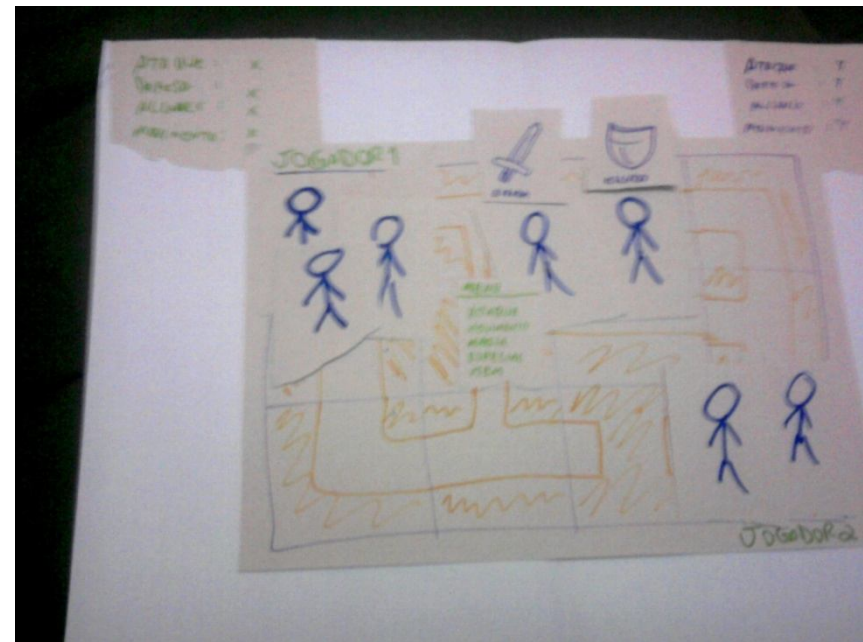
Game Design

No-Name Game: Esboço do cenário e principais ações do jogador e da fase.

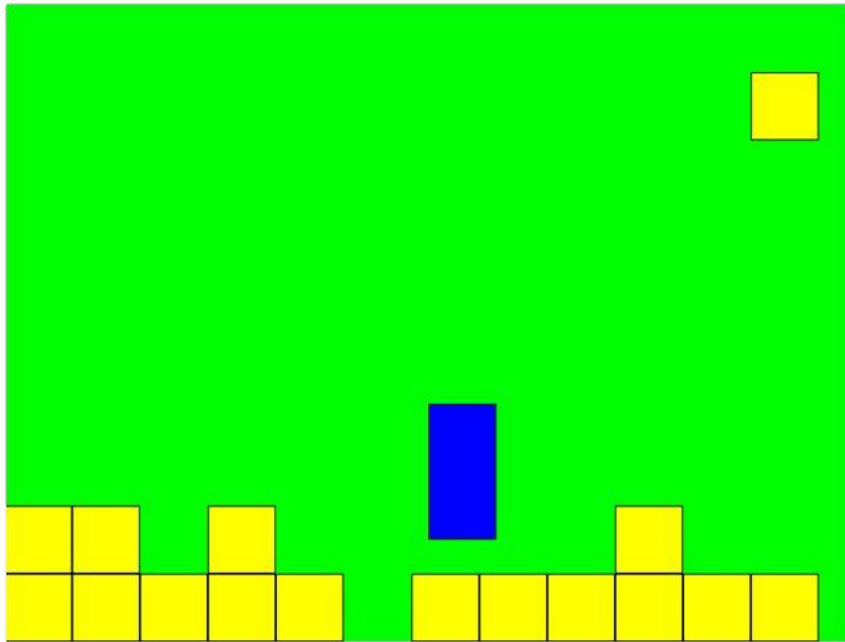


- Principais pontos:
 - Objetivos
 - Desafios
 - Recompensas
- Testar conceito e validar o gameplay utilizando protótipos antes de começar o desenvolvimento

Protótipo de Papel



Protótipo Digital



Início da Produção



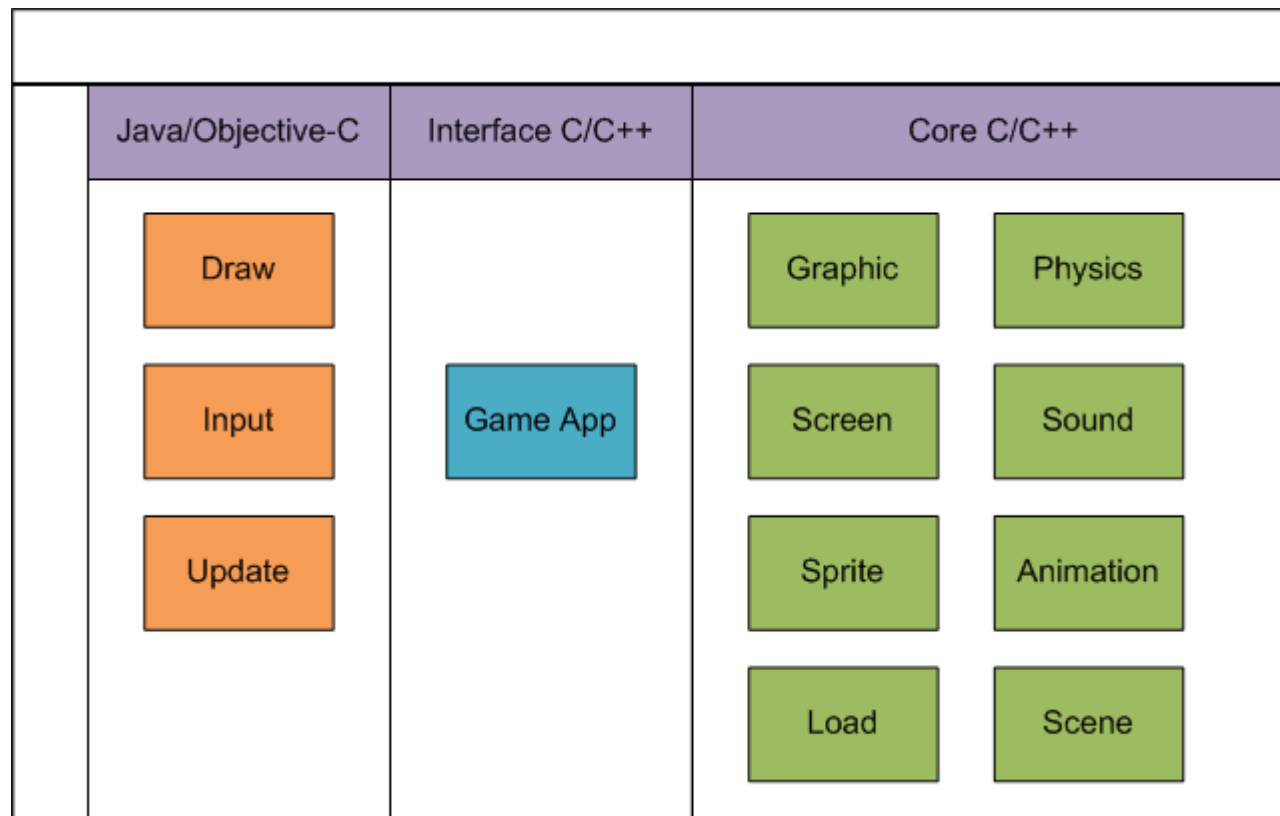
- Jogar fora o protótipo!
- Utilize um repositório para o código do jogo com controle de versão
- Mantenha documentado todos softwares utilizados e suas respectivas versões
- Crie um cronograma com foco em tarefas de cada área
- Compartilhe os documentos
- Defina linguagens e engines quer serão utilizadas

Programação

MacOSX		Windows	Linux
Objective-C	Java		
C/C++ & OpenGL ES			
iOS	Android OS		

- Desde o início do projeto, preocupe-se com a flexibilidade das classes
- Modularizar as classes do jogo também é recomendado a para reusabilidade do código
- Design Pattern Singleton é muito utilizado em jogos
- Separe o código do jogo do código específico de um OS
- Utilize Máquina de Estados para controlar o fluxo do jogo
- Torne seu jogo graficamente escalável

Arquitetura

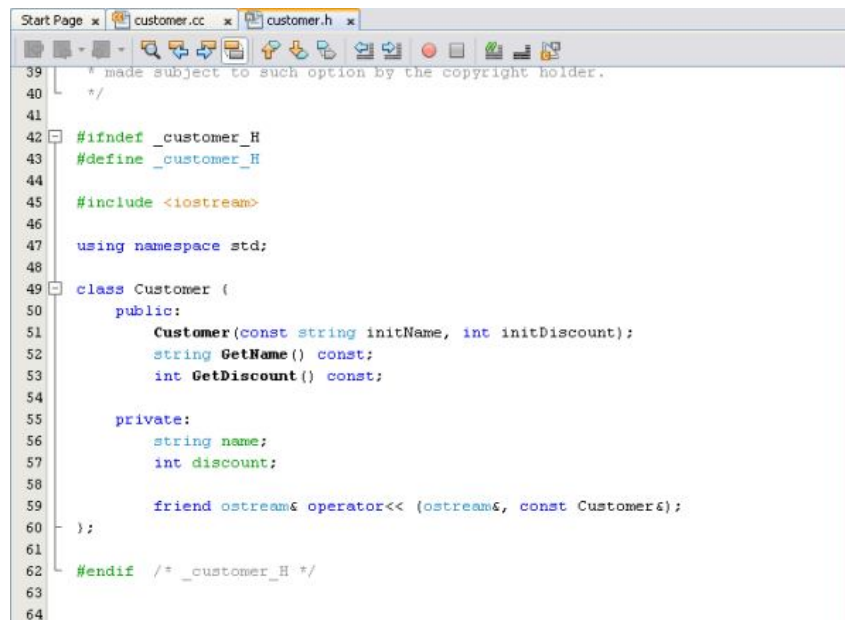


Game Engines

Engine	Prós	Contras	Indie	Pro
Unity3D	Editor 3D fácil de usar, software bem completo.	Apenas para jogos em 3D e 2.5D.	\$400*	\$1500*
ShiVa 3D	Programação em C++, mais barata que a Unity3D.	Apenas para jogos em 3D e 2.5D.	€ 169	€ 1499
Corona SDK	Fácil de usar e rápida para criar jogos em 2D.	Apenas jogos em 2D. Linguagem script própria.	\$199*	\$199*
Marmalade SDK	SDK muito completo, programação em C++.	Não possui editor visual.	\$ 149	\$ 499
Cocos2Dx	Opensource e programação em C++.	Apenas jogos em 2D. Não possui editor visual.	\$ 0	\$ 0

*Por plataforma

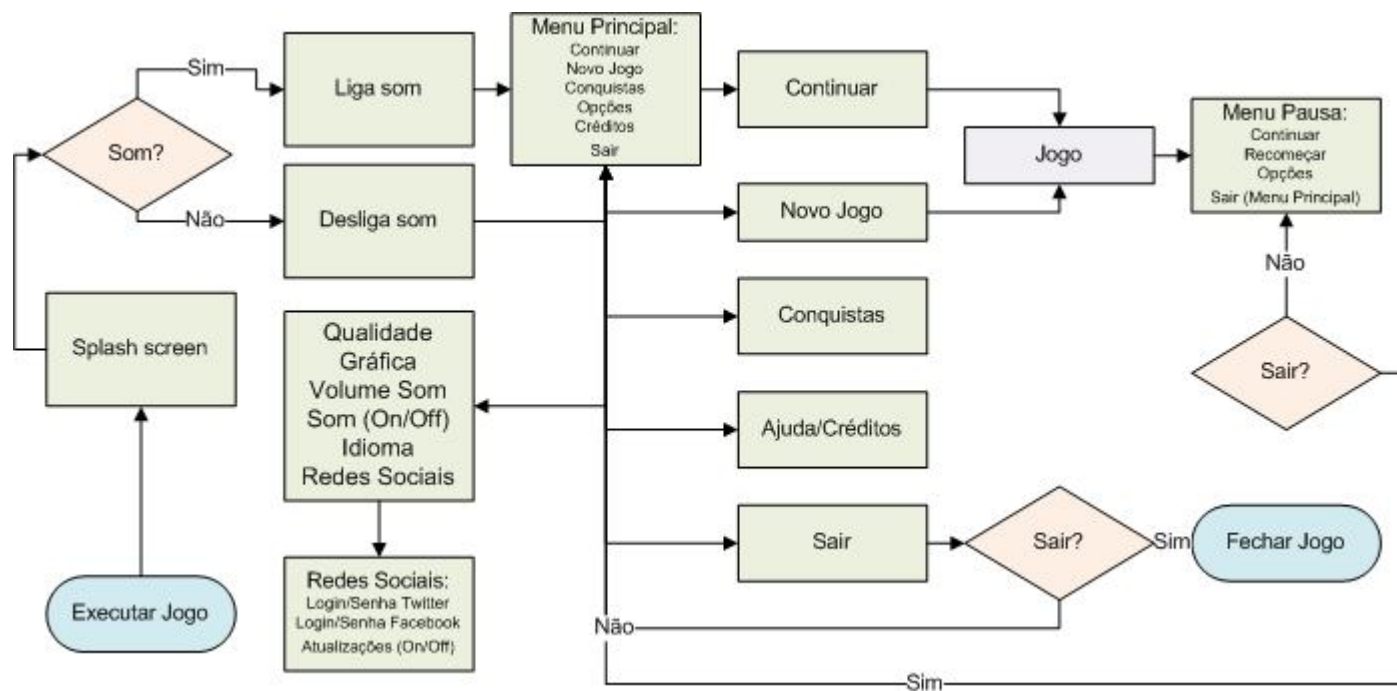
Frameworks



```
39  /* made subject to such option by the copyright holder.
40  */
41
42  #ifndef _customer_H
43  #define _customer_H
44
45  #include <iostream>
46
47  using namespace std;
48
49  class Customer {
50  public:
51      Customer(const string initName, int initDiscount);
52      string GetName() const;
53      int GetDiscount() const;
54
55  private:
56      string name;
57      int discount;
58
59      friend ostream& operator<< (ostream&, const Customer&);
60  };
61
62  #endif /* _customer_H */
63
64
```

- Física
 - [Box2D](#)
 - [Bullet Physics Library](#)
- Rede
 - [RakNet](#)
- Compressão
 - [Zlib](#)
- Propagandas (Ads)
 - [Admob](#)
 - [Flurry](#)
- Script
 - [Lua](#)

Fluxo de menus e telas



Sistema de telas



- Pop-ups devem ser usadas para confirmar escolhas do jogador, pois dão mais ênfase a pergunta
- Transições entre telas com efeitos tornam o jogo mais profissional
- Interface não deve ser poluída, simplifique opções e use ícones sempre que possível
- Splash com logos devem ser dinâmicas, assim será simples adicionar novas imagens

Sistema gráfico



- Melhore o desempenho do load de imagens/texturas reutilizando recursos que já estão na memória
- Matenha as animações curtas e objetivas
- Para jogos em 2D, utilize Sprite Sheets
- Faça a parte gráfica ser escalável para que seja automaticamente redimensionado, independente da resolução

Sistema de som



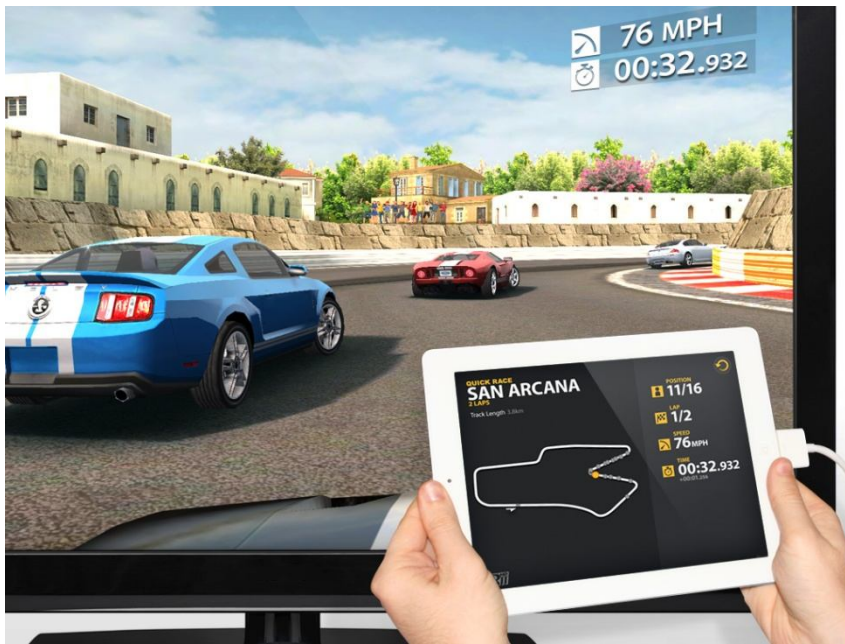
- Nem todo formato de audio tem um bom desempenho em dispositivos móveis
- A concorrência de sons é um grande problema em aparelhos com hardware fraco
- Utilize formatos de audio comprimidos sempre que possível para músicas
- Efeitos sonoros devem ser curtos para não causar problemas de concorrência
- Utilize loop na música para que seu tamanho fique menor

Sistema de E/S



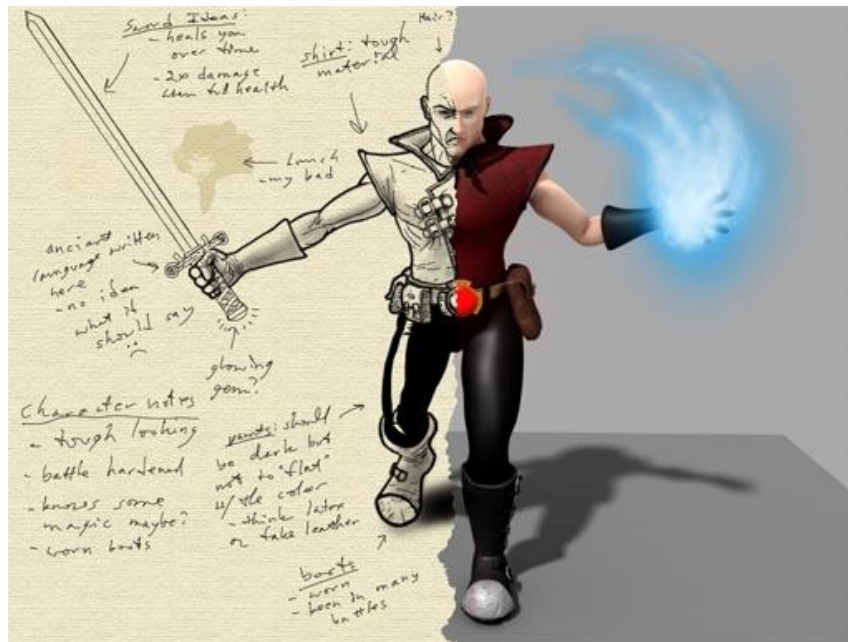
- Trate separadamente todos eventos de input que um device pode receber:
 - Toque na tela
 - Acelerômetro
 - Botões de ação
- Caso seja possível rodar seu jogo em landscape e portrait, implemente a troca de orientação

Outros sistemas



- Física
- Save game
- Logs
- Achievements
- Redes Sociais
- Partida em rede
- Publicidade
- Venda de itens

Arte



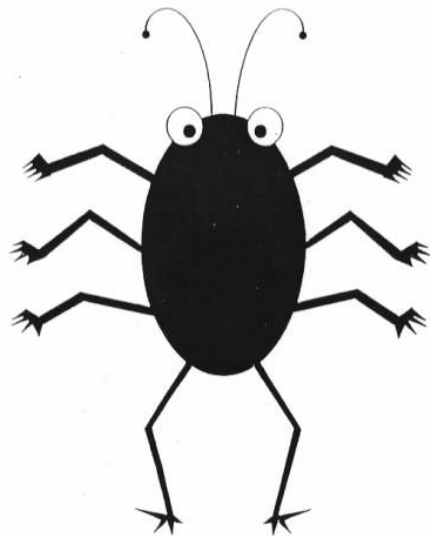
- Nem todos aparelhos suportam compressão de textura (ATITC, PVRTC, ETC)
- Recursos descomprimidos são maiores, porém garantem o funcionamento do jogo em qualquer aparelho
- Em alguns aparelhos, o OpenGL ES requer que os arquivos sejam POT
- Utilize fontes TTF ou mapa de caracteres (adicione as fontes no projeto também)

Testes

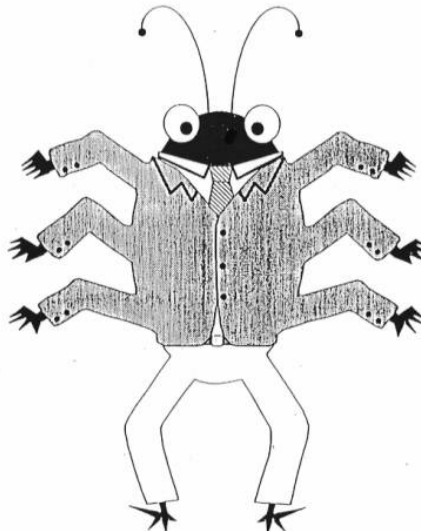


- Peça para outras pessoas testarem seu jogo em aparelhos diferentes
- Caso reportem erros, peça para descreverem como causaram o erro
- Utilize logs para identificar cada detalhe do jogo
- Público ideal para testar:
 - Seu público-alvo
 - Quem não costuma jogar

Possíveis problemas



BUG



FEATURE

- Não confunda um bug com uma nova feature no jogo
- Flexibilidade do código ajuda em casos de grandes alterações
- Alterações em imagem são bem comuns, mas geralmente não causam impacto na programação
- Em alguns casos uma correção pode ser a remoção ou alteração de uma feature
- Saiba ouvir sugestões, principalmente para corrigir problemas



Publicação

Apple App Store

- Registrar-se em <http://developer.apple.com/>
- Pagar a taxa anual de \$99
- Enviar jogo para homologação da Apple
- Se aprovado, seu jogo estará disponível pelo preço que você definiu

Android Market

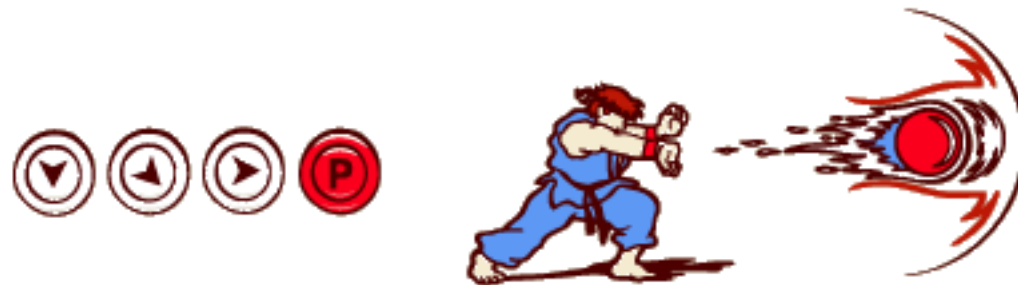
- Registrar-se em <http://developer.android.com>
- Pagar a taxa anual de \$25
- Não requer homologação do Google, seu jogo fica disponível imediatamente

Pós Produção



- Mantenha seu jogo vivo com DLCs, updates e promoções
- Utilize o jogo criado para fazer uma sequência ou reskin
- Se o jogo criar uma marca, tente explorar novos gêneros com os mesmos recursos
- Pense no que deu certo e errado para melhorar nos próximos projetos

Dicas sobre Desenvolvimento de Jogos





Estude!

- C/C++ e OpenGL
- Orientação a Objetos
- Arquitetura/Engenharia de Software
- Design Patterns
- Boas práticas de programação
- Inglês



Não confie em emuladores!

- Testar em aparelhos reais pode ajudar a identificar problemas que não ocorrem em emuladores, como:
 - Erros gráficos
 - Estouro de memória
 - Baixo desempenho
- Peça para outras pessoas testarem em seus aparelhos



Cuide bem do seu código!

- Mantenha sempre atualizado e organizado
- Adicione comentários úteis no código
- Utilize repositórios de código para controle de versão e alteração (SVN, CVS, Perforce, Mercurial, SourceSafe, etc) - BitBucket
- Documentação, mesmo mínima, ajuda a lembrar o que foi feito



Publique seu jogo!

- Pesquise detalhes de como publicar em cada plataforma
- Se for vender seu jogo, faça também uma versão gratuita limitada
- Mantenha seu jogo atualizado com conteúdo adicional
- Sempre que possível, publique em mais de uma plataforma



Jogue!

- Conheça mais sobre outros jogos, não apenas os que você gosta
- Tente entender como foi feita a programação de seus elementos (interface, gameplay, etc)
- Aprenda a analisar tecnicamente os jogos e entender a sua complexidade



Mantenha-se atualizado!

- Saiba o que está acontecendo na indústria
- Entenda as tendências relacionadas a jogos (redes sociais, smartphones, tablets, TV digital, html5)
- Leia sobre desenvolvimento de outros jogos e post-mortem dos projetos
- Converse com profissionais da indústria
- Tente participar de eventos, concursos e fóruns
- Acompanhe sites e blogs, monte um portfólio

Obrigado!

bruno@gamedeveloper.com.br

Links

- Sites:
 - **Gamasutra** - <http://www.gamasutra.com/>
 - **Gamesindustry** - <http://www.gamesindustry.biz/>
 - AppData - <http://www.appdata.com/>
 - Ars Technica - <http://arstechnica.com/gaming/>
 - MCV - <http://www.mcvuk.com/>
 - **GameCareerGuide** - <http://www.gamecareerguide.com/>
- Fóruns
 - GameDev - <http://www.gamedev.com.br/>
 - PDJ - <http://www.programadoresdejogos.com/>
 - UniDev - <http://www.unidev.com.br/>
- Blogs:
 - **AltDevBlogADay** - <http://altdevblogaday.com/>
 - **Game Developer** – <http://gamedeveloper.com.br>
 - Game Producer - <http://www.gameproducer.net/>
 - Indie Games - <http://indiegames.com/index.html>
- Revistas:
 - **Casual Connect** - <http://casualconnect.org/magazine-archive/>
 - **Develop** - <http://www.develop-online.net/>
 - EDGE - <http://www.next-gen.biz/>
 - Game Developer Magazine - <http://www.gdmag.com/homepage.htm>
 - **Game Sauce** - <http://gamesauce.org/news/>
 - Wired - <http://www.wired.com/gaming>

Livros

- Desenvolvimento de Jogos e programação:
 - **Code Complete** - <http://amzn.to/oqrBeP>
 - **More Effective C++** - <http://amzn.to/qQSksw>
 - Desenvolvimento de Jogos Eletrônicos - <http://novatec.com.br/livros/jogos2/>
 - Software Engineering for Game Developers - <http://amzn.to/qrY571>
 - Game Engine Architecture - <http://amzn.to/pgnZFR>
 - Game Coding Complete - <http://amzn.to/qAONz4>
 - Programming Game AI by Example - <http://amzn.to/nQ5SdY>
 - Real-time Rendering - <http://amzn.to/pbHDTp>
 - Game Programming All in One - <http://amzn.to/p2S05M>
 - Beginning C++ through Game Programming - <http://amzn.to/q0X1fV>
 - XNA Game Studio Creators Guide - <http://amzn.to/nmSIFf>
 - iPhone Programming The Big Nerd Ranch Guide - <http://amzn.to/p6kUkJ>
 - Head First iPhone and iPad Development - <http://amzn.to/p0qXOo>
 - Agile Game Development with Scrum - <http://amzn.to/nfp3T3>
- Game Design:
 - Game Design Workshop - <http://amzn.to/pVvJE0>
 - Rules of Play: Game Design Fundamentals - <http://amzn.to/oHmOZ8>
 - The Game Design Reader: A Rules of Play Anthology - <http://amzn.to/pTjOd7>
 - Half-Real: Video Games between Real Rules and Fictional Worlds - <http://amzn.to/qA7yd6>
 - A Theory of Fun for Game Design - <http://amzn.to/nmR1Ya>
 - The Art of Game Design: A book of lenses - <http://amzn.to/p0Z5xr>