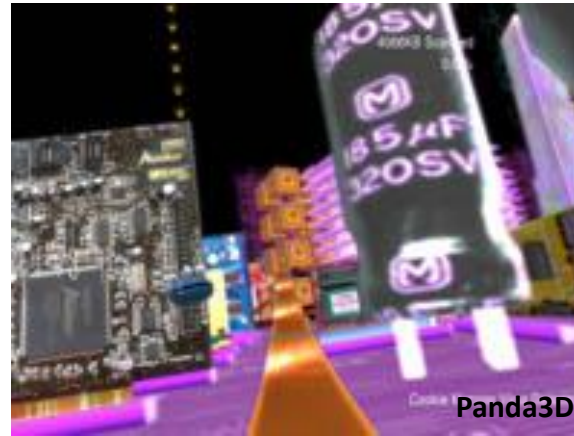


Desenvolvimento de Jogos de Computador

Indústria, design e programação de jogos



Seminário de Tecnologias de Programação de Jogos

José Braga de Vasconcelos
Nuno Magalhães Ribeiro

Instituto Superior de Tecnologias Avançadas
ISTEC Porto

Dezembro 2013

Seminário de Tecnologias de Programação de Jogos

Objetivos

- Apresentar a área de design e programação de jogos digitais
- Interpretar um jogo como uma aplicação de software e multimédia
- Apresentar ambientes integrados de desenvolvimento de jogos
- Integrar a programação de jogos na área da engenharia multimédia.

Seminário de Tecnologias de Programação de Jogos

Agenda

- Indústria de jogos de computador
- Processo de criação e design de um jogo
- Engenharia de software para o desenvolvimento de jogos
- Ambientes de desenvolvimento de Jogos
- Programação de jogos com Python e Pygame
- Programação de jogos para a Web em HTML5

Indústria de jogos

História recente da programação de jogos

1958 - os jogos passaram à era digital e entraram no mundo dos computadores. William Higinbotham desenvolveu o jogo ***Tennis for two***.

1962 - um aluno do *Massachusetts Institute of Technology* (MIT) desenvolveu o jogo ***Space Wars***.

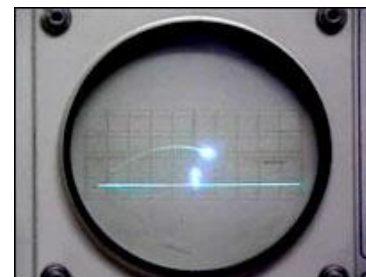
1971 - primeira máquina de jogos a moedas. A partir daqui surgiram companhias como a **Atari** e outras que conceberam **consolas para ligar aos televisores**.

1982 - ZX Spectrum

1986 - Nintendo Game Boy

1994 – Sony Play Station

2006 – Nintendo Wii



Mercados Web de jogos

Games Marketplaces

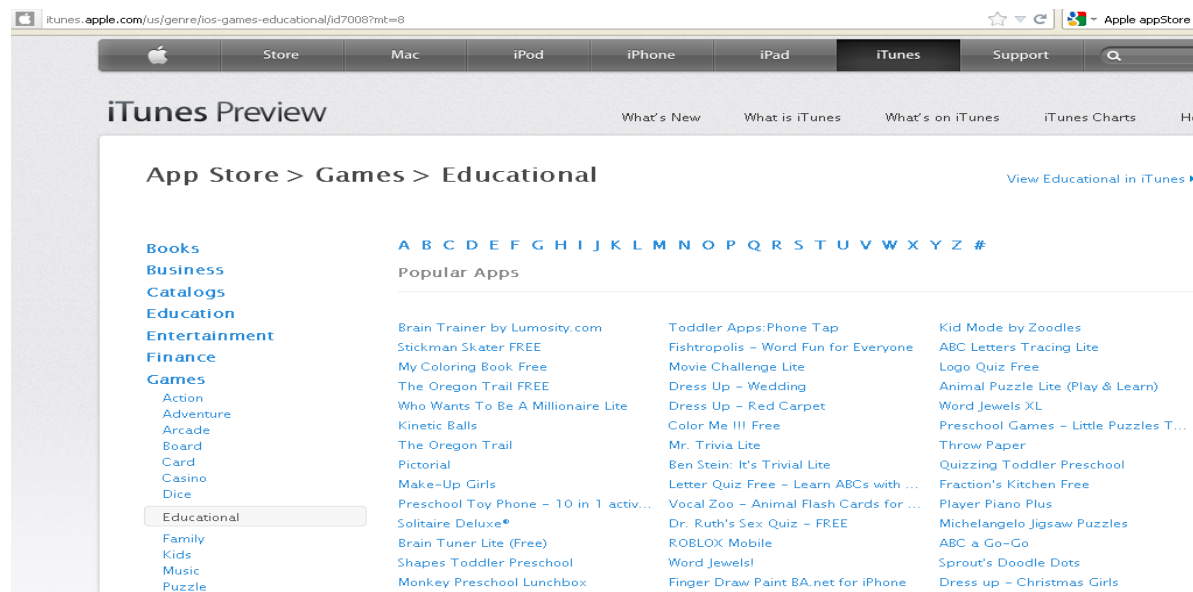
Actualmente a concorrência e a diversidade de opções na Web para a aquisição de jogos e outras aplicações é muito intensa e dinâmica.

Diariamente são lançados novos produtos nos mercados web (web *marketplaces*) mais conhecidos, nomeadamente a **AppStore** (Apple), o **Android store** (Google Play Store), o **Windows Mobile Market** e XBOX (consola de jogos) *marketplace*.

Existem ainda os *marketplaces* das consolas de jogos Nintendo e Sony, também muito activas no mercado global.

Mercados Web de jogos

Apple AppStore



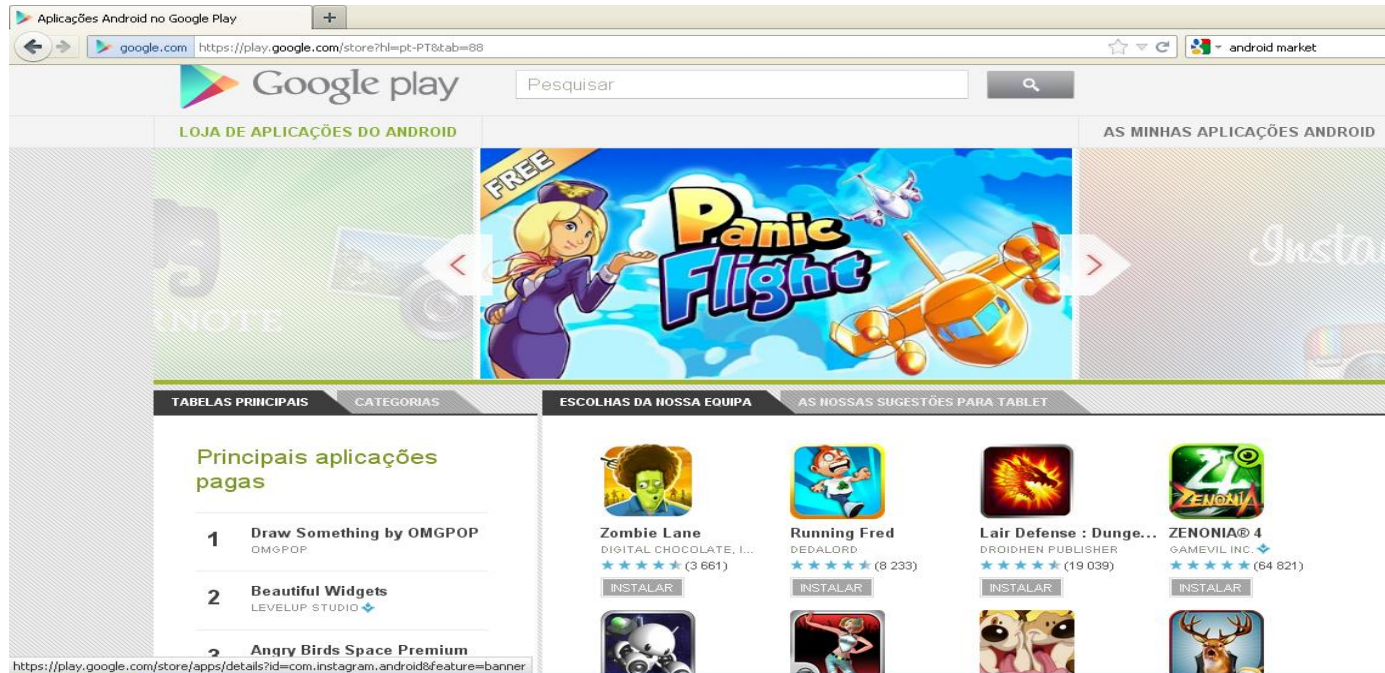
O sítio Web *AppStore* da Apple para a promoção e comercialização de aplicações foi o **primeiro mercado** (*marketplace*) web que obteve **visibilidade global**, assim como um sucesso crescente ao longo dos anos.

A loja web da Apple tem milhares de **jogos** disponíveis, assim como outras **aplicações**, serviços e produtos em constante actualização.

A loja web da Apple ainda é considerada o **marketplace global** mais activo e dinâmico tendo em conta o número e diversidade de transacções efectuadas em todo o globo.

Mercados Web de jogos

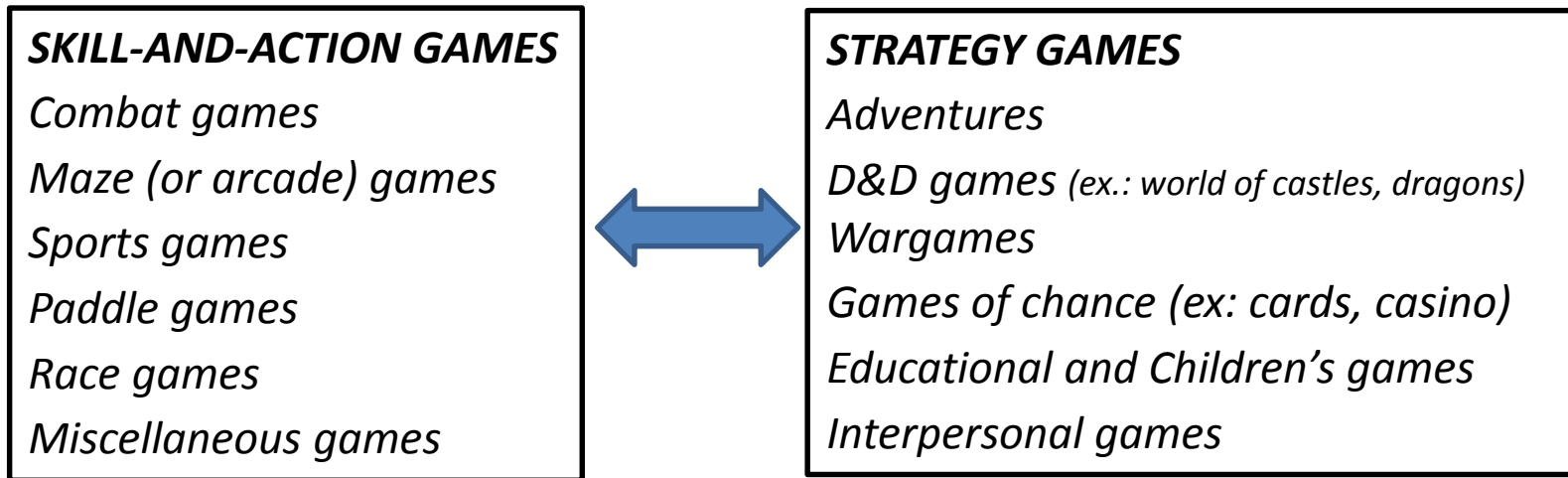
Google Play Store



Este mercado tem crescido paralelamente ao desenvolvimento do sistema operativo **Android** que se especializou na gestão de software e serviços para dispositivos (computadores) móveis, incluindo *smartphones*. Desde o seu lançamento, o número de transacções e downloads cresceu diariamente até ultrapassar (em 2011) os cem milhões de downloads.

Taxionomia de jogos de computador

A forma mais comum de um jogo de computador é designada por *skill and action* (S & A) que aplica essencialmente os movimentos de coordenação visio-motora, incluindo a coordenação manual.



Gameplay

Desafios e acções do jogo

Coordenação físico – motora

precisão

sincronização

combinação de movimentos

Memória e conhecimento

Reconhecimento de padrões

Conhecimento do senso comum

Cultura geral

Desafios lógicos

Dedução

Descodificação

Raciocínio

Gameplay

Desafios e acções do jogo

Exploração

- Relacionamentos espaciais

- Labirintos

- Puzzles

- Espaços ilógicos

Criação e construção

- Edifícios e equipamento

- Design e estética

Conflito

- Estratégia e tática

- Gestão logística

- Sobrevivência

Desafios económicos

- Gestão de recursos

- Gestão de sistemas de produção

Modelo de design e desenvolvimento de um jogo

Engenharia de software

1. Concepção

Concepção da ideia, dos objectivos e da arquitetura geral do jogo

2. Desenvolvimento

Desenvolvimento dos detalhes, prototipagem e testes

3. Manutenção

Ajustamentos e refinamentos

Engenharia de software para o desenvolvimento de jogos

O processo de **Engenharia de Software** contém vários passos antes de entrar efectivamente na parte de codificação e implementação.

No caso do desenvolvimento de jogos, estes passos são introduzidos na criação do documento de desenvolvimento do jogo (***Game Design Document - GDD***).

O GDD é um documento que deverá conter os detalhes necessários para a criação e desenvolvimento de um jogo de computador.

A primeira fase de um projecto para o desenvolvimento de um jogo é a definição da **ideia inicial do jogo**. O resultado desta fase com a análise dos requisitos de software e consequente design do software, resultará no documento de desenvolvimento do jogo (GDD).

Este documento deverá permitir converter os requisitos e o **design do software** numa construção efectiva do **jogo (código)** que determina a fase de **codificação, implementação e integração** do projecto de software.

Engenharia de software para o desenvolvimento de jogos

Funções e competências

Os jogos e os referentes programas de software estão cada vez mais **sofisticados** e complexos.

As **equipas de desenvolvimento** (design, programação, integração e produção) para a construção de jogos têm vindo a crescer, assim como as **funções e competências** dos **recursos humanos**, cada vez mais **especializados**.

- *Design roles*
- *Programmimg and engineering roles*
- *Production Roles*
- *Art roles*
- *Audio roles*
- *Writing roles*
- *Test roles*

Engenharia de software para o desenvolvimento de jogos

Ciências da computação

O desenvolvimento de jogos de computador requer a compreensão e experiência de programação de computadores. Entre outras áreas das ciências da computação, o desenvolvimento de jogos incorpora e integra conhecimento das seguintes áreas:

- Algoritmia e Programação de Computadores
- Engenharia de software (OOD e OOP)
- Computação gráfica e multimédia
- Interacção Homem - Máquina
- Inteligência Artificial

Engenharia de software para o desenvolvimento de jogos

Ciências da computação

Competências a desenvolver:

Computação gráfica

- Para colocar o jogo no ecrã do computador ou dispositivo móvel
- Animação gráfica: modelização e simulação de objectos físicos virtuais

Arte (design) computacional

- Imagens e ambiente gráfico apelativo

Interface design

- Jogos com interfaces intuitivos e funcionais

Áudio computacional

- Efeitos de som durante a execução do jogo

Inteligência artificial

- Para tornar o jogo e os adversários “inteligentes”

Algoritmos e programação eficiente

- Optimização de código para incrementar a performance (*run time efficiency*)

Requisitos principais para o desenvolvimento de jogos

- Interface apelativo e funcional
 - Acções do jogo simples e com efeitos visuais perceptíveis
 - Introdução de efeitos sonoros para alguns eventos
 - Representação dos resultados das acções (ex.: guias, pontuação)
 - Representação do estado (geral) do jogo
- Definição da “inteligência” do jogo
 - Definição clara dos objectivos e do objectivo final
 - Representação de avanços e recuos
 - Definição de possíveis estratégias
 - Utilização de ferramentas para algumas acções
 - Ritmo do jogo adequado
 - Aplicação de algoritmos de IA
 - Definição da “inteligência” dos oponentes

Definição de regras de um jogo

As **regras** de um jogo deverão definir:

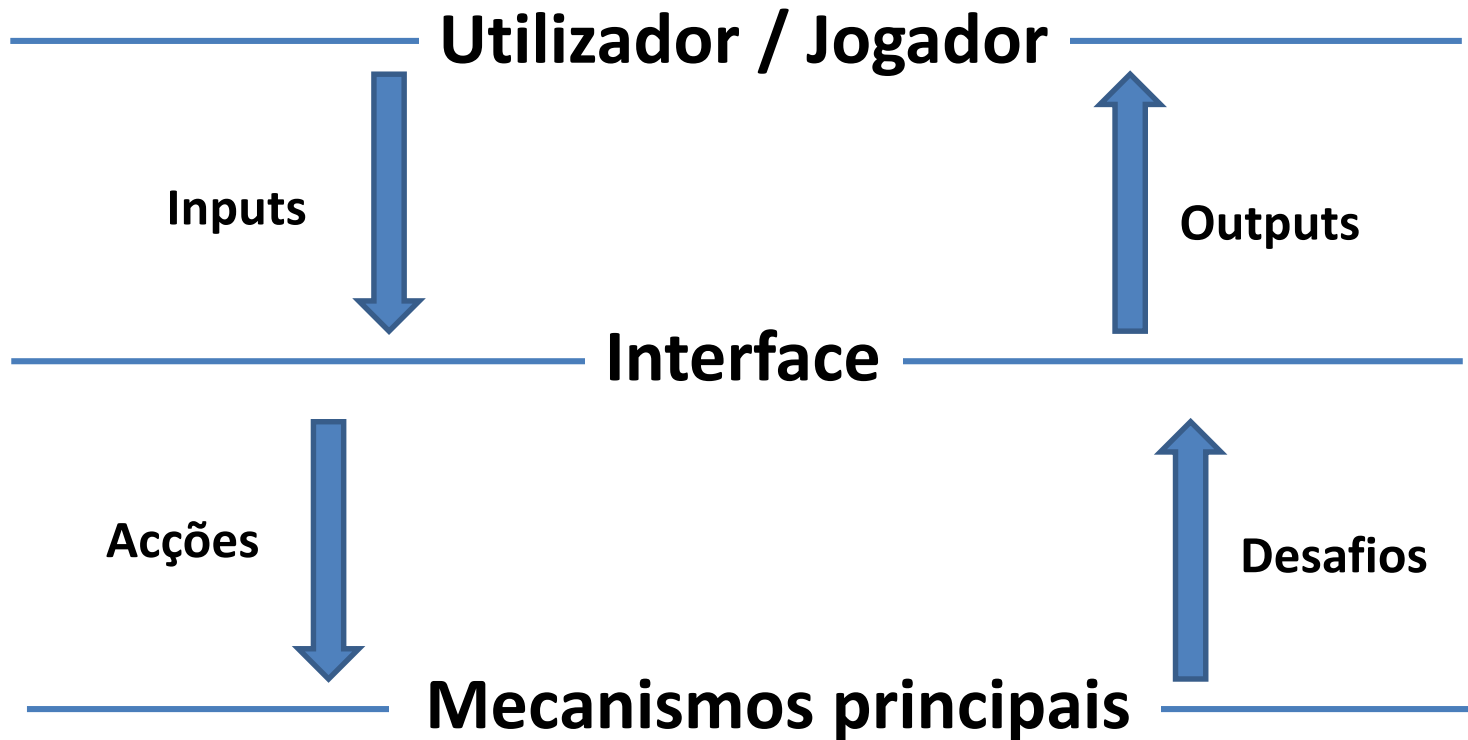
- A **semiótica do jogo**: o significado, os relacionamentos, e o comportamento das entidades do jogo.
- Os **objetivos do jogo**, incluindo (se existir) a definição do objetivo final, ou seja, a forma como o jogo termina com sucesso.
- Os desafios e as acções do jogo (**gameplay**).
- A **sequência do jogo**: a progressão das actividades que definem o jogo.
- **Meta-regras**: regras sobre as regras de modo a definir as circunstâncias e excepções que permitem alterar as regras .

Requisitos principais para o desenvolvimento de jogos

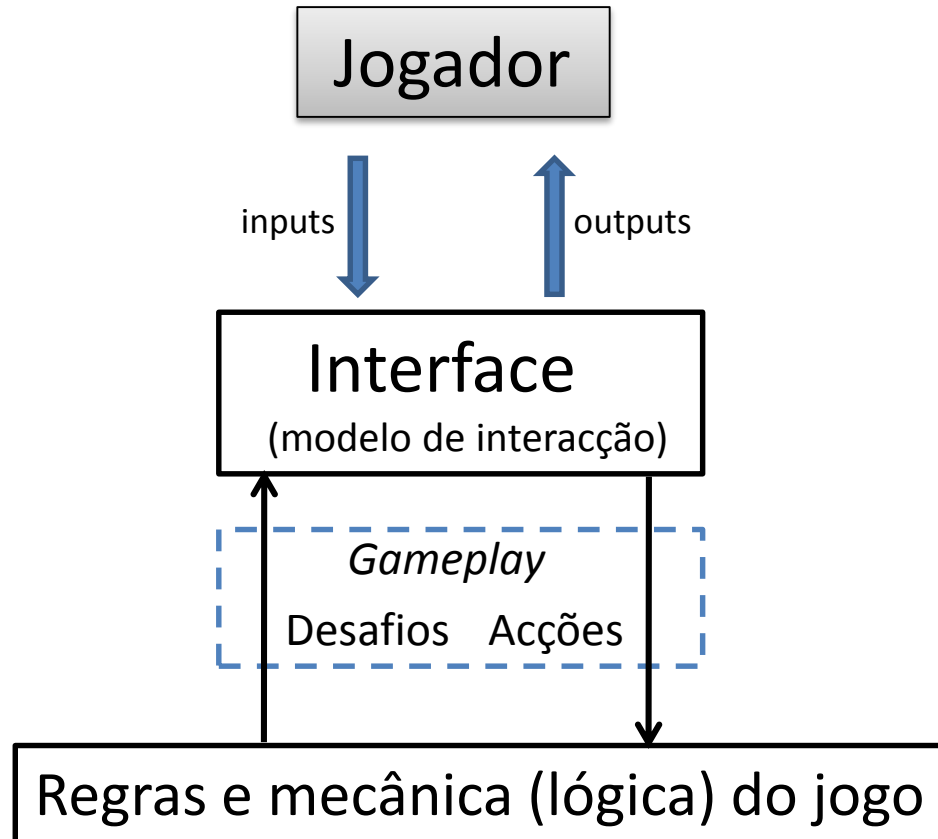
- Definição das regras do jogo
 - Instruções que os jogadores aceitam
- *Gameplay* (acção do jogo)
 - Desafios e acções do jogo
- Condição que determina o fim (objectivo final) do jogo
- Meta-regras
 - Situações que determinam a alteração de regras
 - Excepções às regras predefinidas

Gameplay

Componentes e mecanismos de um jogo



Componentes e relacionamentos principais de um jogo



Sistema de desenvolvimento de um jogo

Componentes e serviços do sistema de jogo

Um jogo de computador incorpora os seguintes componentes e serviços principais de forma modular.

- Objectos de visualização
- Objectos de áudio
- Entrada e saída de dados (*input-output*)
- Persistência (memória secundária)
- Comunicações e *middleware*

Sistema de desenvolvimento de um jogo (*OOD View*)

Objectos principais de um jogo

Entidades do jogo

(game entities)

Representa o que existe no jogo, ou seja, os personagens, os obstáculos, ou qualquer outro objecto que faça parte do jogo.

Acção do jogo

(Gameplay)

Esta classe de objectos do jogo representa os mecanismos e a lógica do jogo. Estes objectos representam as regras e as restrições do jogo através da definição de eventos, tais como correr, saltar, atirar, acelerar, explodir, desaparecer, diminuir, aumentar, etc.

Estes objectos são normalmente invocados pelos objectos anteriores (entidades do jogo), e podem afectar outros tipos de objectos.

Sistema de desenvolvimento de um jogo

Objectos principais de um jogo

Entidades do jogo em tempo de execução

(runtime game objects)

Esta classe de objectos não deve ser confundida com a primeira (entidades do jogo). Esta classe representa as diferentes formas como aparecem as entidades do jogo durante a execução do jogo.

Estado do jogo

(game state)

Objectos para a representação dos diferentes estados de um jogo.

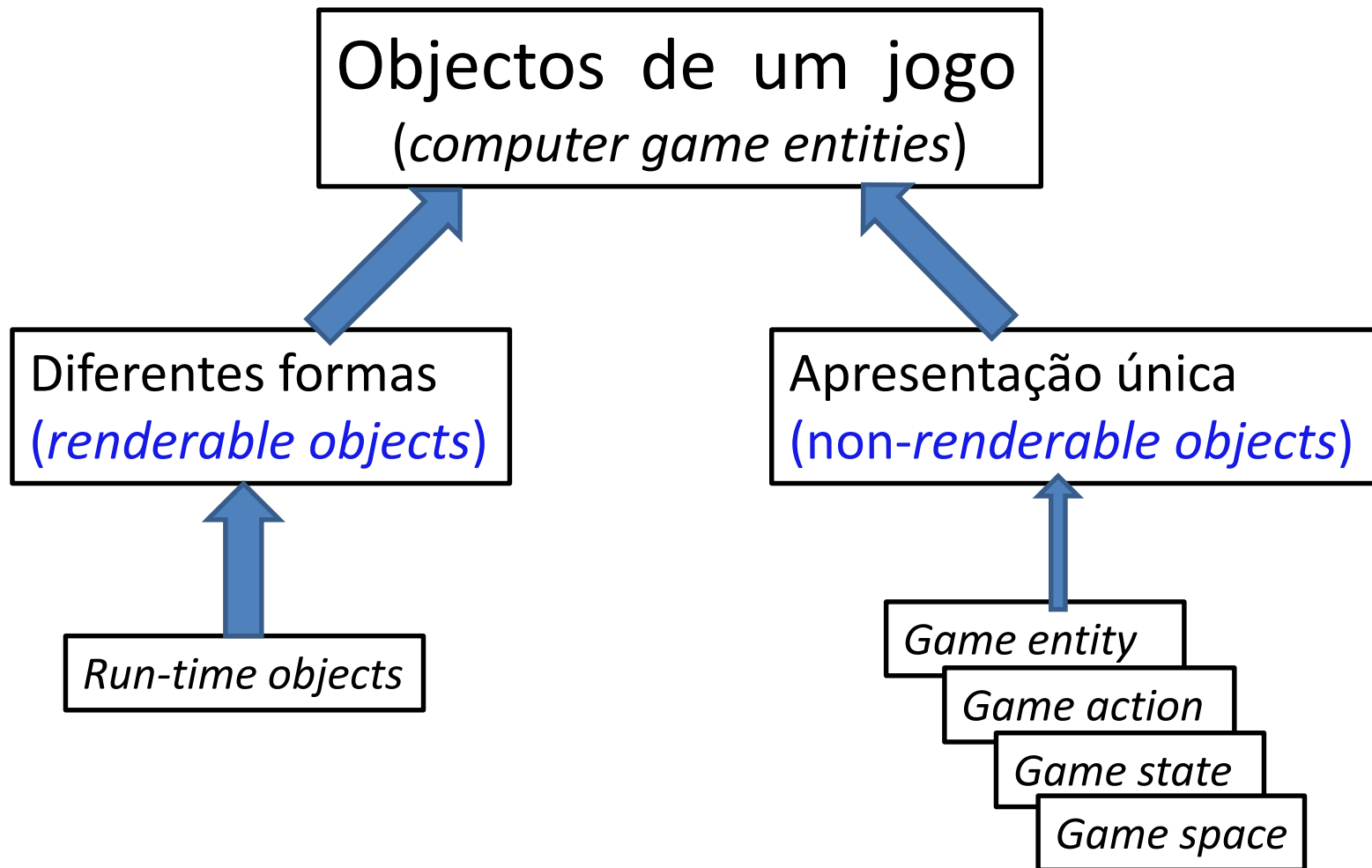
Espaço do jogo

(game space)

Colecções (conjuntos) de objectos reutilizáveis que representam diferentes realidades, tipos e padrões de jogos. Estes objectos normalmente representam processos de optimização do jogo, incluindo algoritmos de rendering, assim como mecanismos de IA.

Sistema de desenvolvimento de um jogo

Objectos principais de um jogo



Modelo de design e desenvolvimento de um jogo

Engenharia de software

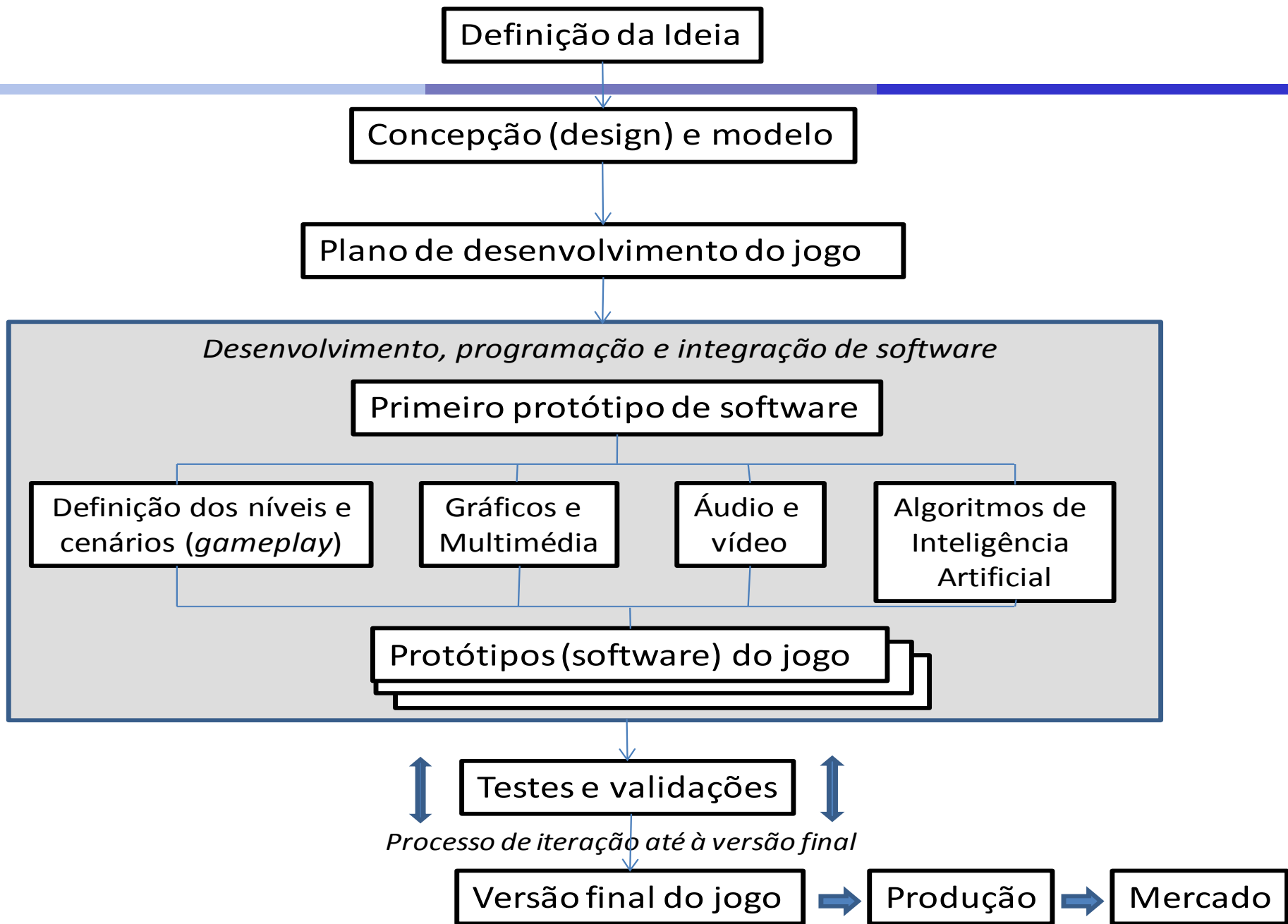
1. Concepção

- Definição do conceito e da ideia do jogo
- Objectivos principais
- Definição de audiência (público alvo do jogo)
- Principais funções e papéis do jogador
- *Storytelling* (história do jogo)

2. Desenvolvimento

- Definição do *gamelay*
- Entidades e protagonista(s) do jogo
- Definição do *game world* (ambiente)
- Mecanismos e regras principais
- Definição dos diferentes níveis do jogo (progressão)
- Definição do ambiente de desenvolvimento de software
 - Linguagem, motor do jogo (*game engine*), bibliotecas
- Construir (codificar e integrar)
- Testar e iterar

Engenharia de software para o desenvolvimento de jogos



Engenharia de software para o desenvolvimento de jogos

Áreas relacionadas

- *Overview of computer games (game industry)*
- *Game software project management*
- *Analysis of the game-engine framework code*
- *Object-oriented methods for games*
- *Software design patterns for games*
- *Animation techniques*
- *Simulating and creating virtual environments*
- *Game worlds and game Intelligence*
- *Computer graphics and user controls (HCI)*

Ambientes integrados para o desenvolvimento de jogos

Game IDE

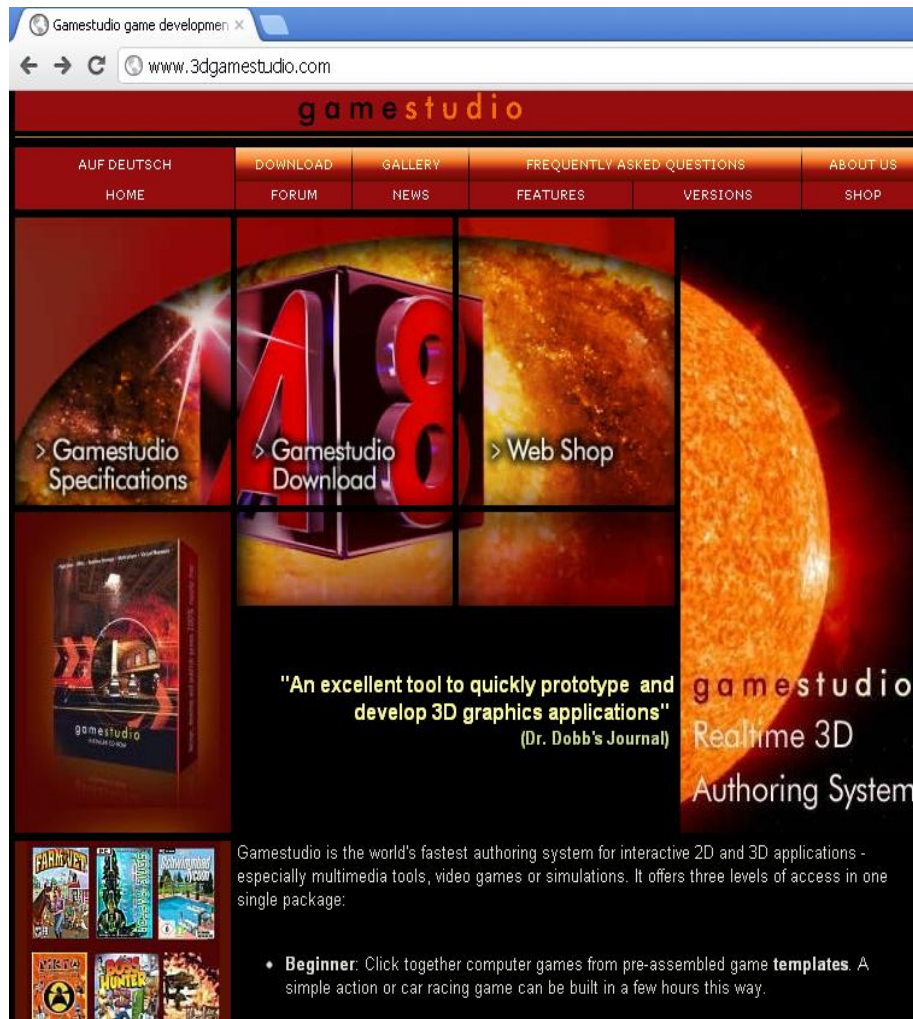
Existem diferentes **ambientes integrados de desenvolvimento (IDE) de jogos** que permitem facilitar e organizar a construção dos diversos componentes de software de um jogo.

Estes ambientes têm normalmente por base uma **linguagem de programação**, razão pela qual exige alguma experiência de programação de computadores no sentido de maximizar a utilização destas ferramentas.

No entanto, alguns ambientes de desenvolvimento, como é o caso do **Unity 3D** e do **MIT Scratch**, permitem o desenvolvimento de jogos de uma forma simples e integrada, através do seu interface intuitivo, e sem a especificação de longas linhas de código fonte numa determinada linguagem de programação.

Ambientes integrados de desenvolvimento de jogos

3D Game Studio



The screenshot shows the homepage of the Gamestudio website. The browser address bar displays "www.3dgamestudio.com". The website has a dark red header with the "gamestudio" logo. Below the header is a navigation menu with links for "AUF DEUTSCH", "DOWNLOAD", "GALLERY", "FREQUENTLY ASKED QUESTIONS", and "ABOUT US". Underneath, there are sub-links for "HOME", "FORUM", "NEWS", "FEATURES", "VERSIONS", and "SHOP". The main content area features a large graphic of the "A8" logo in a 3D cube, set against a background of a planet. Navigation arrows point to "Gamestudio Specifications", "Gamestudio Download", and "Web Shop". A quote from Dr. Dobb's Journal describes Gamestudio as "An excellent tool to quickly prototype and develop 3D graphics applications". Below this, the text "gamestudio Realtime 3D Authoring System" is displayed. At the bottom, there is a section titled "Gamestudio is the world's fastest authoring system for interactive 2D and 3D applications - especially multimedia tools, video games or simulations. It offers three levels of access in one single package:" followed by a list of game templates and a description of the "Beginner" level.

gamestudio

AUF DEUTSCH DOWNLOAD GALLERY FREQUENTLY ASKED QUESTIONS ABOUT US
HOME FORUM NEWS FEATURES VERSIONS SHOP

> Gamestudio Specifications > Gamestudio Download > Web Shop

"An excellent tool to quickly prototype and develop 3D graphics applications"
(Dr. Dobb's Journal)

gamestudio
Realtime 3D
Authoring System

Gamestudio is the world's fastest authoring system for interactive 2D and 3D applications - especially multimedia tools, video games or simulations. It offers three levels of access in one single package:

- **Beginner:** Click together computer games from pre-assembled game templates. A simple action or car racing game can be built in a few hours this way.

Game Studio é um ambiente de desenvolvimento de software comercial, contudo contém uma versão (Gamestudio A8) sem custos associados de modo a iniciar a aprendizagem desta ferramenta.

O 3D Game Studio contém os componentes e mecanismos essenciais para o desenvolvimento de um jogo 3D: programação e implementação de objectos multimédia (áudio, imagens, vídeos) e outros elementos gráficos (GUI), gestão de colisões, modelos 2D e 3D, e implementação de algoritmos de *rendering*.

Ambientes integrados de desenvolvimento de jogos

Panda 3D



Panda 3D é um ambiente de desenvolvimento de jogos 3D que tem por base as **linguagens de programação Python e C++**.

O Panda 3D tem uma comunidade associada muito activa no desenvolvimento (e suporte) de jogos.

A **documentação** está muito bem estruturada e existe um conjunto vasto de tutoriais e exemplos (com código) de jogos.

O Panda 3D é um projecto desenvolvido e mantido pela Universidade de Carnegie Mellon (EUA).



Information

Gallery

Download

Documentation

Community

Manual

Python Reference

C++ Reference

Cheat Sheets

Videos

Sample Applications

Panda3D Manual: Installing Panda3D in Windows

← [Prev](#)

↑ [Top](#)

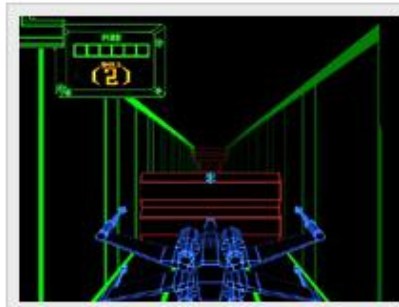
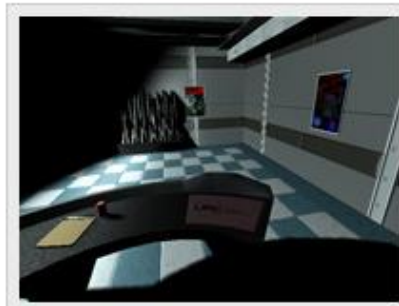
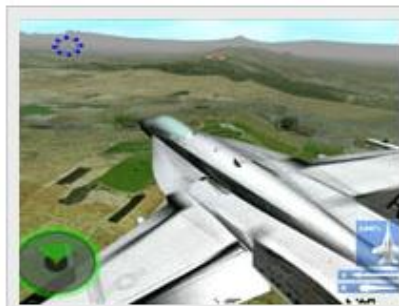
[Next](#) →

Python / [C++](#)

If you have already installed panda previously, you should uninstall it before installing a new version. Once your machine is clean of all previous versions of panda, you should download the windows installer from the [download](#) page. Run the installer, and follow the prompts:



RANDOM USER PROJECTS



Ambientes integrados de desenvolvimento de jogos

Panda 3D

www.panda3d.org/download.php?sdk&version=1.8.1



Carnegie Mellon
Entertainment
Technology
Center



[Information](#) [Gallery](#) [Download](#) [Documentation](#) [Community](#)

[Runtime](#) [SDK](#)

Download Panda3D SDK



Panda3D SDK 1.8.1
Free One-Click Download
Microsoft Windows

Download Panda3D SDK 1.8.1

Bugfix release, fixing many of the issues found in 1.8.0.

See the [Release Notes](#).

[Other versions](#)



[Microsoft Windows](#) (Detected)
All versions



[Mac OS X](#)
Leopard, Snow Leopard



[Ubuntu](#)
DEB repository



[Debian](#)
DEB repository



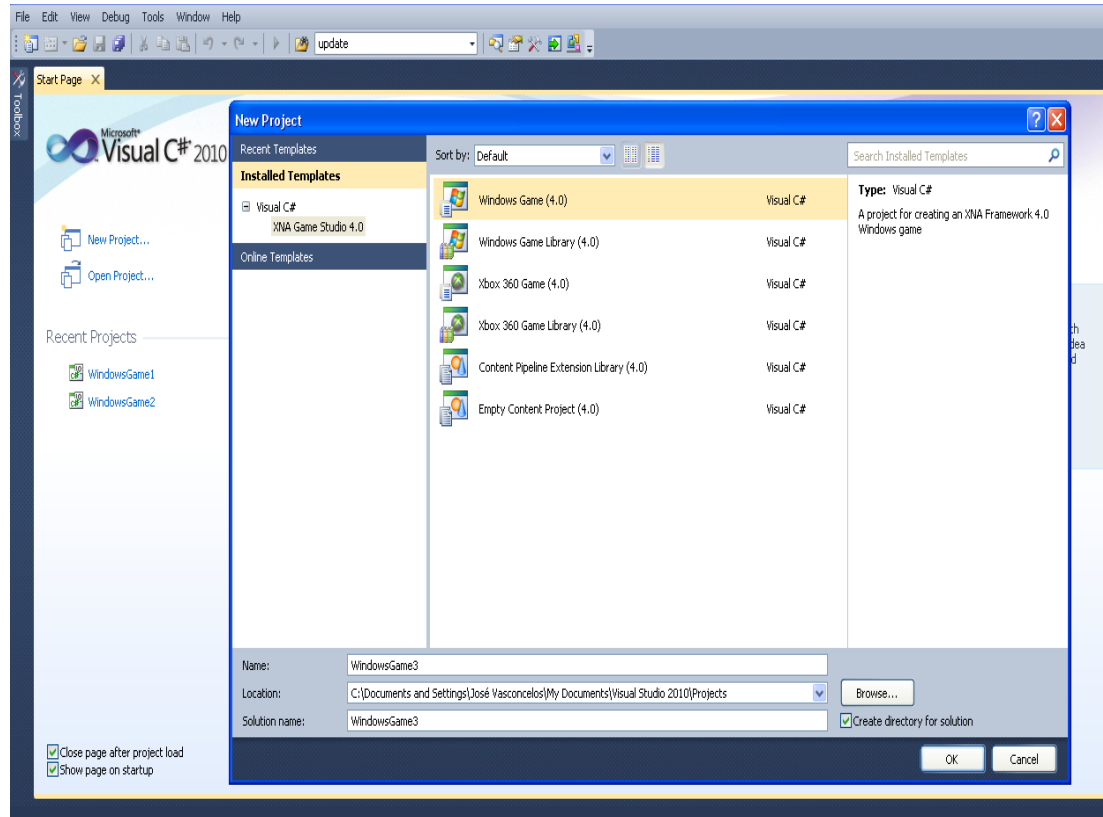
[Fedora](#)
RPM packages



[Other](#)
Source Archives

Ambientes integrados de desenvolvimento de jogos

Microsoft XNA



O **XNA Game Studio** é um ambiente integrado de desenvolvimento de software dedicado à construção e programação de jogos.

O Microsoft XNA é incorporado depois de instalado no ambiente d **Microsoft Visual Studio** e exige o conhecimento da **linguagem de programação C#**.

As **classes** necessárias para a execução de um jogo XNA funciona em **.NET para Windows** ou em **.NET Compact Framework** para a consola XBOX ou Zune.

Ambientes integrados de desenvolvimento de jogos

Microsoft XNA

www.microsoft.com/download/en/details.aspx?id=23714

Download Center

[Windows](#) [Office](#) [Product downloads](#) [Download categories](#) [Security](#) [Resources](#)



Microsoft XNA Game Studio 4.0



Quick links

- [↓ Overview](#)
- [↓ System requirements](#)
- [↓ Instructions](#)

Looking for support?



[Visit the Microsoft Support site now >](#)

Windows Server System Center

Choose a private cloud solution from Microsoft.

Microsoft XNA Game Studio 4.0 makes it easier than ever to create great video games for Windows-based PCs, Xbox 360 consoles, and Windows Phone

Quick details

Version:	4.0	Date published:	9/16/2010
Language:	English		
File name	Size		
XNAGS40_setup.exe	48.8 MB	DOWNLOAD	

Overview

XNA Game Studio 4.0 is a programming environment that allows you to use Visual Studio 2010 to create games for Windows Phone, the Xbox 360 console, and Windows-based computers. XNA Game Studio 4.0 includes the XNA Framework 4.0, which is a set of managed libraries designed for game development based on Microsoft .NET Framework 4.

[↑ Top of page](#)

Ambientes integrados de desenvolvimento de jogos

MIT Scratch

The screenshot shows the MIT Scratch website interface. At the top, there's a navigation bar with links for 'casa', 'projectos', 'galerias', 'suporte', 'fóruns', 'acerca', and 'Lingua'. Below this is the Scratch logo and a search bar. The main content area features a large blue banner with the text 'Create and share your own interactive stories, games, music, and art'. To the left of the banner is the Scratch cat logo and a 'Download Scratch' button. To the right is a preview of a Scratch script. Below the banner are three sections: 'Projectos em Destaque' with three project thumbnails, 'Collab Camp' with a 'Learn more' link, and 'Projectos Seleccionados por Collab-Camp' with three more project thumbnails. A 'Dia do Scratch' section is also visible at the bottom right of the screenshot.

Este projecto é um exemplo ilustrativo da importância da área do design de software, principalmente na área de **aprendizagem da algoritmia** e consequente lógica da matemática e do raciocínio.

Este software oferece uma **perspectiva simples de construção de jogos** e simulações com base na configuração de blocos de dados, assim permitindo que os programadores possam ter idades de dez anos (aproximadamente) ou superiores, mas conseguirem criar os seus próprios **projectos de jogos e simulações interactivas**.

Ambientes de Desenvolvimento de Jogos

IDE *Unity*

3D Game authoring tool



Ambientes integrados de desenvolvimento de jogos

Unity IDE



<http://unity3d.com/>



Ambientes integrados de desenvolvimento de jogos

Unity IDE

Editor

Integrated Editor

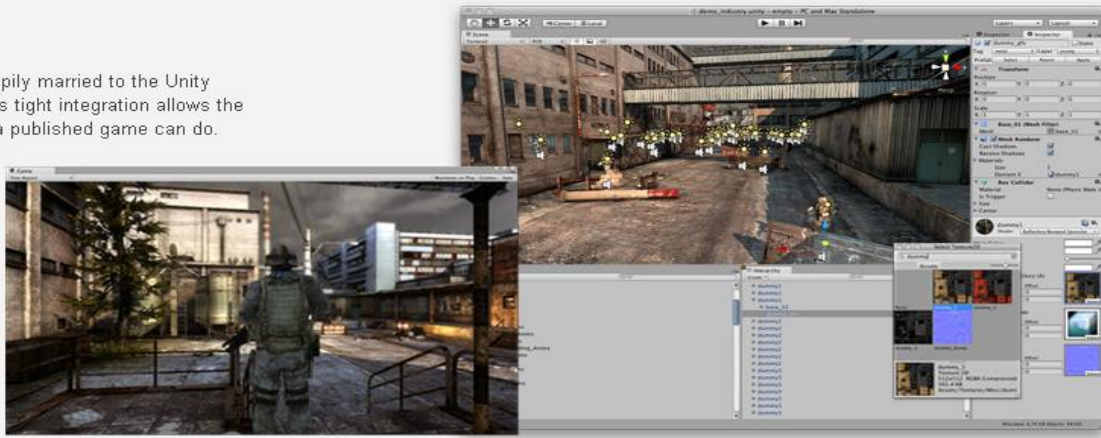
Scene Construction

Asset Pipeline

Shuriken

Integrated Editor

The Unity game engine is happily married to the Unity development environment. This tight integration allows the Unity Editor to do everything a published game can do.



O **Unity 3D** tem servido de ambiente a um conjunto de jogos com sucesso globalmente, incluindo jogos online e para as diferentes plataformas existentes. As suas funcionalidades e potencialidades de representação de modelos têm vindo a facilitar o **desenvolvimento de software para jogos em diferentes níveis de complexidade e referentes orçamentos.**

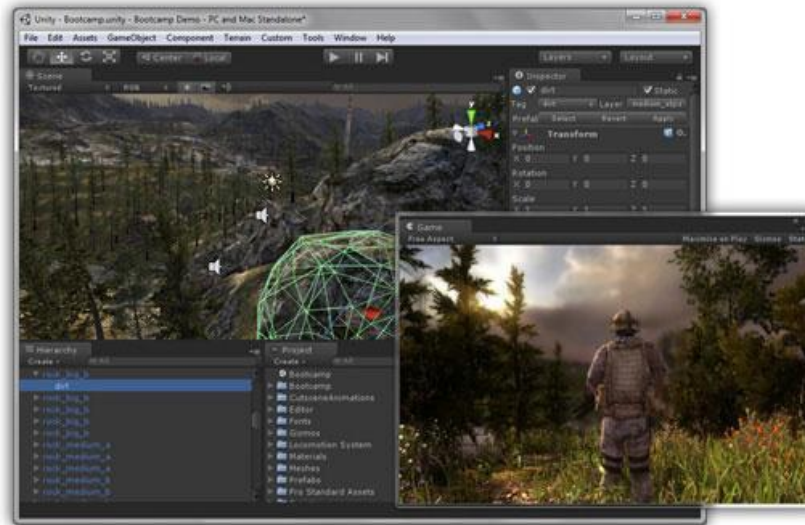
Programadores a título pessoal, **designers** com determinadas competências de programação, assim como grandes **empresas** multinacionais na área do desenvolvimento de software para jogos, têm vindo a utilizar este ambiente.

As funcionalidades gráficas, de design e de integração de componentes multimédia, **permitem reduzir significativamente o tempo e o esforço de desenvolvimento de um jogo.**

Unity 3D

Ambiente integrado para o desenvolvimento de jogos

O **Unity** é um ambiente integrado de desenvolvimento de jogos (IDE), também designado como **game authoring tool** no sentido de incluir uma ferramenta de edição para o design e construção de jogos.



O Unity incorpora um motor (**game engine**) que toma decisões e automatiza operações, como os métodos de **rendering** que permitem manipular e visualizar gráficos 3D, assim como as regras e o **gameplay**, pelas quais o jogo é regido.

3D Unity

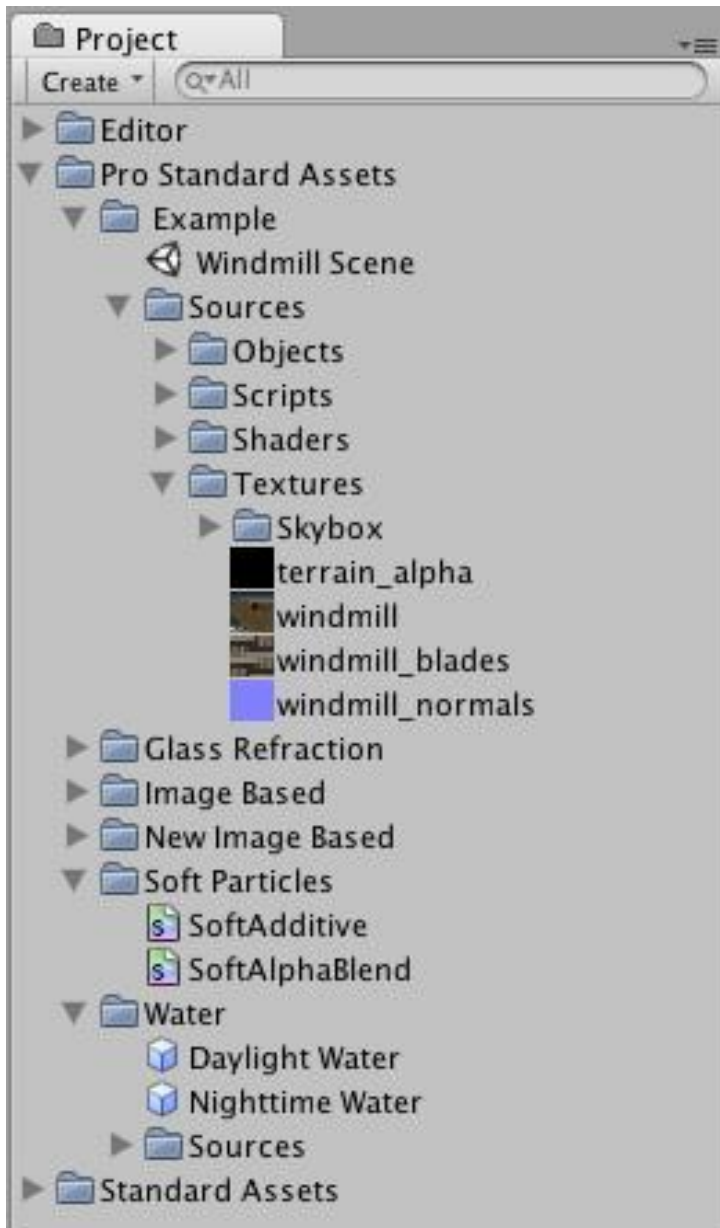
User Interface



IDE 3D Unity

User Interface

Unity project



- **Assets** folder
- **Project View**
- **Game assets:**
Scenes, scripts, 3D models, textures, audio files, and Prefabs.

Prefab is a specific type of asset
(reusable **GameObject** stored in **Project View**)

Linguagem de Programação Python

Python Programming Language - Official Website

www.python.org

python™

search

Advanced Search

- ABOUT >>
- NEWS >>
- DOCUMENTATION >>
- DOWNLOAD >>
- 下載 >>
- COMMUNITY >>
- FOUNDATION >>
- CORE DEVELOPMENT >>

Help

Package Index

Quick Links (2.7.6)

- >> Documentation
- >> Windows Installer
- >> Source Distribution

Quick Links (3.3.3)

- >> Documentation
- >> Windows Installer
- >> Source Distribution

Python Jobs

Python Merchandise

Python Wiki

Python Insider Blog

Python Programming Language – Official Website

Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.

Python is free to use, even for commercial products, because of its OSI-approved [open source license](#).

New to Python or choosing between Python 2 and Python 3? Read [Python 2 or Python 3](#).

The [Python Software Foundation](#) holds the intellectual property rights behind Python, underwrites the [PyCon conference](#), and funds many other projects in the Python community.

[Read more](#), -or- [download Python now](#)

- » **Python 3.4.0 beta 1 has been released**
The first beta for Python 3.4, [Python 3.4.0b1](#), has been released.
Published: Sun, 24 November 2013, 14:00 -0800
- » **Python 3.3.3 released**
[Python 3.3.3](#) is now available.
Published: Tue, 19 November 2013, 08:00 +0100
- » **Python 3.3.3 release candidate 2 has been released**
The [second release candidate for Python 3.3.3](#) has been released.
Published: Tue, 12 October 2013, 08:00 +0100

Support the Python Community

Help the Python community by becoming an associate member or making a one-time donation.

Python 3 Poll

I wish there was Python 3 support in

(enter PyPI package name)

Vote Results

Simulate biomolecules with Python...

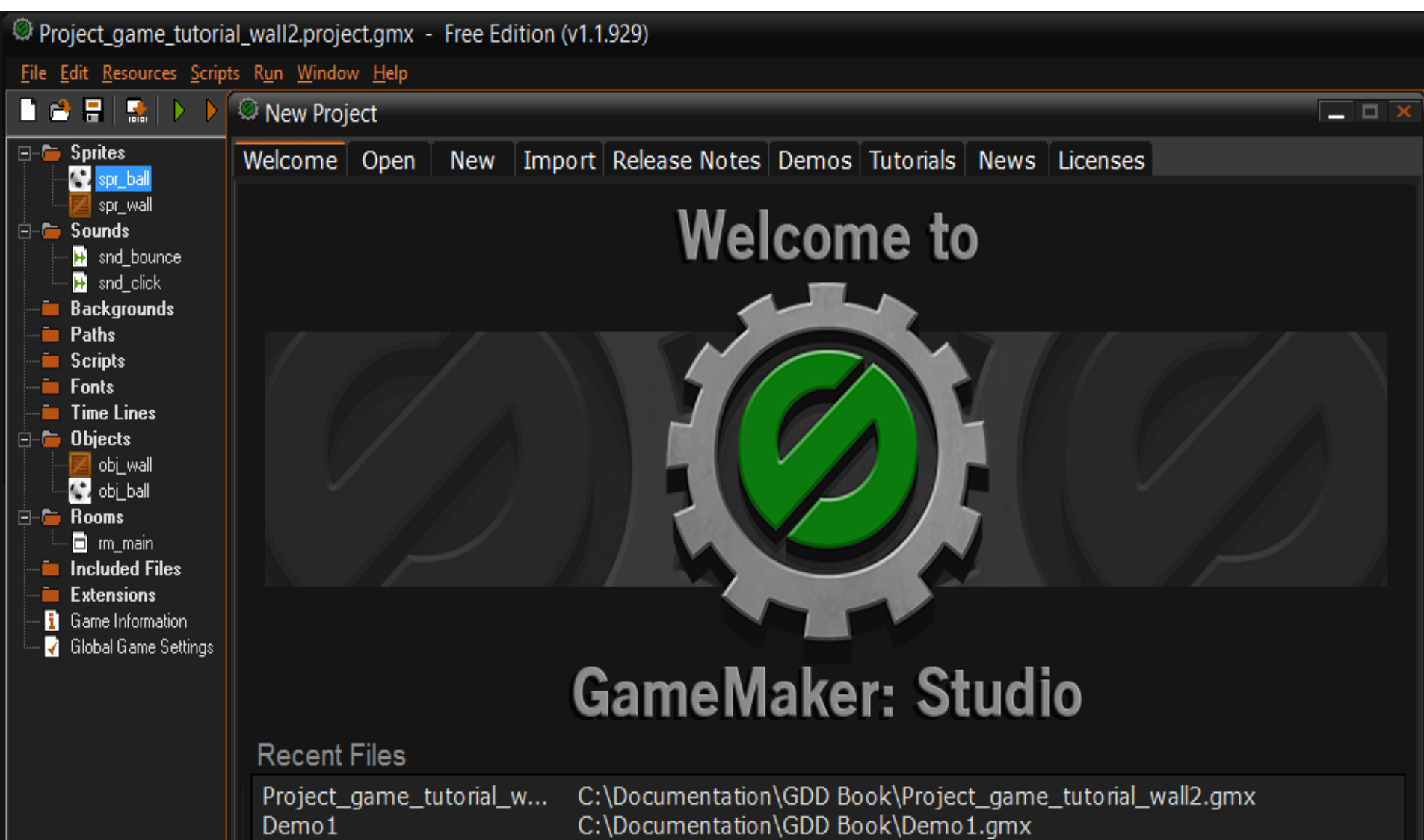


... joining users such as Rackspace, Industrial Light and Magic, AstraZeneca, Honeywell

Exemplo de um Programa em Python e Pygame

```
7% anima_bola.py - C:\Documentation\ISTEC\2013_14\Seminarío 2013\Pygame\ani... - □ ×
File Edit Format Run Options Windows Help
# bola em movimento
# bibliotecas
import pygame, sys, os
from pygame.locals import *
# variáveis e constantes
size = (640,480)
speed = [7,7] # velocidade horizontal e vertical
speed2 = [5,5]
black = (0,0,0)
white = (255,255,255)
grey = (128,128,128)
# inicializa o pygame e define ambiente e entidades
pygame.init()
clock = pygame.time.Clock()
screen = pygame.display.set_mode(size)
bola = pygame.image.load(os.path.join('imagens', 'ball3.png'))
balao = pygame.image.load(os.path.join('imagens', 'ball3.gif'))
brect = bola.get_rect()
brect2 = balao.get_rect()
# ciclo do jogo / animação
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    brect = brect.move(speed)
    # verifica limites da janela e simula colisões
    if brect.left < 0 or brect.right > size[0]:
        speed[0] = -speed[0] # inverte movimento horizontal
    if brect.top < 0 or brect.bottom > size[1]:
        speed[1] = -speed[1] # inverte movimento vertical
    brect2 = brect2.move(speed2)
    # verifica limites da janela e simula colisões
    if brect2.left < 0 or brect2.right > size[0]:
        speed2[0] = -speed2[0] # inverte movimento horizontal
    if brect2.top < 0 or brect2.bottom > size[1]:
        speed2[1] = -speed2[1] # inverte movimento vertical
    # display objects
    screen.fill(white)
    screen.blit(bola, brect)
```

GameMaker Studio



Introdução à Programação de Jogos para a Web em HTML5

Seminário de Programação de Jogos

ISTEC

2 de dezembro de 2013

Nuno Magalhães Ribeiro

(baseado no Cap. 6 da obra

“Tecnologias de Programação de Jogos”)

Sumário

1. Objetivos e introdução
2. Linguagens de programação de jogos web
3. Novas funcionalidades HTML5 e CSS3 úteis para jogos interativos na web
4. Ferramentas de desenvolvimento
5. Exemplo 1: jogos baseados no DOM
6. Exemplo 2: jogos baseados no Canvas
7. Conclusão e referências online

Objetivos desta palestra

- Introduzir os conceitos básicos sobre a programação de jogos HTML5 para a web
- Destacar os aspetos mais importantes que são aprofundados no cap. 6 do livro
- Mostrar os 2 exemplos básicos que são trabalhados passo-a-passo no livro
- Referências para continuar a aprendizagem



Introdução

- A linguagem HTML5 proporciona potencialidades que permitem encará-la como uma “nova” plataforma para os jogos *online*
- Os jogos desenvolvidos em HTML5 podem ser jogados em computadores *desktop*, telemóveis, *smartphones* e *tablets* (ex: Android, *iPhone*, *iPad*)

Introdução

Dois tipos distintos de jogos para a *web*:

- Jogos desenvolvidos com base no Modelo de Objetos do Documento (*DOM*), CSS e *jQuery* – Exemplo: <http://playbiolab.com/>
<http://www.mutantzombiemonsters.com/>
- Jogos desenvolvidos com base no novo elemento *Canvas* da linguagem HTML5 e respetiva API de desenho – Exemplo: <http://moh97.us/flow/>

Linguagens programação web

- Linguagens HTML5, CSS3 e *JavaScript*, para desenvolver jogos interativos, 2D e 3D, para a *web*
- Utilização dos novos elementos (*tags*) da linguagem HTML5, as propriedades CSS e o desenvolvimento de *scripts* em *JavaScript* de forma a desenvolver jogos que são executados diretamente num *browser*

Linguagens programação web: para que fim são usadas?

- HTML5: estruturar conteúdos (elementos do jogo) e criar hiperligações
- CSS3: propriedades que definem a apresentação visual no ecrã através da atribuição de estilos aos elementos HTML
- *JavaScript*: lógica de controlo da interação do utilizador com o jogo, tornando-o dinâmico, ao permitir o acesso aos elementos HTML através do *DOM*

Linguagens programação web

HTML

- Linguagem de marcas: baseia-se na inserção de identificadores especiais (marcas ou *tags*) no texto de um documento, de modo a delimitar porções de texto para especificar propriedades
- A utilização de marcas permite separar o processo de criação do conteúdo do processo utilizado para apresentar esse conteúdo
- Do ponto de vista sintático, as marcas HTML são identificadores definidos pela linguagem e são delimitados por parêntesis angulares

Exemplo de uma página HTML

```
<!DOCTYPE html>
<html lang="pt-pt">
<head>
    <meta charset="utf-8">
    <title>Ping Pong</title>
</head>
<body>
    <header>
        <h1>Ping Pong</h1>
    </header>
    <footer>
        Jogo Ping Pong para dois jogadores.
    </footer>
<script src="js/jquery-1.9.0.min.js"></script>
<script>
$(function(){
    alert("Bem-vindo ao jogo Ping Pong.");
});
</script>
</body>
</html>
```

Linguagens programação web

CSS

- As folhas de estilos permitem definir o aspeto das páginas HTML de um modo totalmente independente dos elementos (do conteúdo)
- A folha de estilos contém um conjunto de regras que indicam ao *browser* a forma como deve apresentar no ecrã os elementos da página
- Uma regra é constituída por duas partes: um selector (determina a marca HTML à qual se aplica a regra de estilo) e a declaração do estilo (propriedades de formatação a aplicar à marca)

Linguagens programação web

CSS

- As regras de estilo podem ser embebidas na página HTML a que se aplicam por intermédio da marca `<STYLE>`.
- Em alternativa, as regras podem ser armazenadas num ficheiro separado, com a extensão “.css”, desde que se estabeleça uma ligação entre o ficheiro das regras e a página HTML a que se aplicam, colocando a marca `<LINK>` no cabeçalho do documento HTML

Linguagens programação web

JavaScript

- As instruções *JavaScript* embebidas numa página HTML dotam o *browser* com a capacidade de responder a eventos gerados pelo utilizador, tais como cliques do rato, pressão de teclas e navegação entre páginas
- O uso de *scripts* permite alterar a página HTML enquanto está ser apresentada no *browser*, proporcionando conteúdos mais interativos ao utilizador através da manipulação, no lado do cliente, dos elementos existentes na página

Linguagens programação web

JavaScript

Uso de código *JavaScript* em páginas HTML:

- Sob a forma de instruções e funções contidas numa marca `<SCRIPT>` dentro da página
- Através da especificação de um ficheiro externo contendo o código *JavaScript* (extensão “.js”)
- Através da especificação de uma expressão *JavaScript* como um valor de um atributo HTML
- Sob a forma de *Event Handlers* contidos em marcas HTML para lidar com eventos que resultam de ações do utilizador

Funcionalidades HTML5 para jogos

Para suportar a execução de jogos diretamente no *browser* a especificação HTML5 introduziu novas funcionalidades, incluindo:

- Novos elementos e atributos, tais como secções que conferem às páginas estrutura adicional e um novo significado
- Um conjunto de API que suportam a implementação de aplicações *web* (muito populares desde o surgimento da *web 2.0*)
- Suporte nativo para vídeo e áudio digital

Funcionalidades HTML5 para jogos

Rendering gráfico e bitmaps com a *tag Canvas*

- O elemento `<canvas>` da especificação HTML5 proporciona funções de baixo nível para desenhar formas geométricas e manipular imagens *bitmap*
- Funciona como uma marca `` dinâmica do lado do cliente, permitindo carregar imagens para o seu interior, desenhar formas geométricas no respetivo local e interagir com a área de desenho através de código *JavaScript*

Funcionalidades HTML5 para jogos

API de desenho SVG e WebGL

- Para além do elemento *Canvas* a linguagem HTML5 inclui ainda duas API adicionais para suportar o desenho de objetos gráficos:
 - A linguagem *Scalable VectorGraphics* (*SVG*) para desenho vetorial bidimensional (2D): é possível inserir código *SVG inline* sem recorrer a um objeto específico ou embeber uma marca específica
 - A linguagem *WebGL* para desenho vetorial tridimensional (3D): uma API *JavaScript* com uma sintaxe idêntica à sintaxe do *OpenGL 2.0 ES*

Funcionalidades HTML5 para jogos

Elementos multimédia para áudio e vídeo

- Suporte nativo de áudio e vídeo digital para reproduzir *clips* de áudio e vídeo diretamente no *browser* sem recorrer a *plugins* externos
- `<audio src="UmaMusica.mp3" autoplay />`
- `<video src="UmVideo.avi" controls />`
- As marcas de áudio e vídeo proporcionam vários métodos que permitem lidar com o suporte dos vários *codecs* de áudio e vídeo

Funcionalidades HTML5 para jogos

Geolocalização

- A API **Geolocation** da especificação HTML5 permite que uma página *web* obtenha os valores de latitude e longitude nos quais se localiza o dispositivo do utilizador
- Esta funcionalidade é útil, por exemplo, nos casos em que o utilizador executa o jogo num telemóvel ou num *tablet*, e o programador pretende tirar partido da localização atual do utilizador dentro do jogo

Funcionalidades HTML5 para jogos

Armazenamento local persistente *WebStorage*

- A especificação HTML5 proporciona uma solução para o armazenamento persistente de dados por parte dos navegadores *web*
- Para os jogos HTML5, o armazenamento local permite guardar o estado atual de um jogo: nível atual, items já obtidos pelo jogador e a tabela de classificações, entre outros dados que sejam necessários para que o jogador possa retomar o jogo no ponto exato em que o deixou

Funcionalidades HTML5 para jogos

Comunicações e *WebSockets*

- A API ***WebSockets*** permite estabelecer uma ligação orientada a eventos entre uma aplicação *web* e um servidor de *sockets*
- O servidor envia automaticamente os dados atualizados (tipo *push*) aos *browsers*
- Útil para a criação de jogos *multiplayer*, criando a possibilidade de os jogadores poderem interagir uns com os outros de uma forma praticamente imediata

Funcionalidades HTML5 para jogos

Jogos offline com *Application Cache*

- Possibilita declarar o manifesto de uma *cache*, isto é, a lista de recursos (ficheiros) que ficam em disco para serem acedidos posteriormente quando não existir uma ligação ativa à Internet
- Passa a ser possível armazenar localmente todos os conteúdos manipulados pelo jogo, tais como objetos gráficos e *bitmaps*, os ficheiros *JavaScript* de controlo do jogo, as folhas de estilo CSS e os ficheiros HTML

Funcionalidades CSS3 para jogos

- O CSS3 é uma tecnologia muito relevante no contexto da programação de jogos HTML5 baseados no *DOM* (1º tipo de jogos)
- Em particular, há três propriedades do CSS3 especialmente úteis para a programação de jogos: transições, transformações e animações
- As **transições** permitem aplicar efeitos de interpolação durante as mudanças de estilo dos elementos HTML (ex: transição gradual de cor)

Funcionalidades CSS3 para jogos

- A propriedade de **transformação** permite aplicar fatores de escala, rotações e translações aos elementos HTML: estas transformações consistem de efeitos 2D e 3D
- A propriedade de **animação** permite definir *keyframes* para uma animação completa (cada *keyframe* contém as propriedades que devem ser modificadas no instante em que essa *keyframe* é apresentada pelo *browser*)

Ferramentas de desenvolvimento

Navegadores com suporte HTML5

As versões mais recentes dos seguintes *browsers* proporcionam o suporte que é necessário para executar jogos HTML5:

- **Google Chrome** (<http://www.google.com/chrome/>)
- **Mozilla Firefox** (<http://mozilla.com/firefox/>)
- **Apple Safari** (<http://apple.com/safari/>)
- **Opera** (<http://opera.com>)

Ferramentas de desenvolvimento

Google Chrome

- Ferramentas > Ferramentas do Programador
- As ferramentas de depuração (*debugger*) permitem inspecionar dinamicamente o DOM de uma página *web*, visualizar os tempos de carregamento dos recursos e executar excertos arbitrários de código *JavaScript*
- separador *Elements*: visualizar a árvore dos objetos *DOM* da página atual incluindo os estilos associados a cada elemento da página

Ferramentas de desenvolvimento

Google Chrome

- separador *Resources*: permite identificar as propriedades de cada recurso individualmente
- separador *Console*: deixa executar código *JavaScript*; quando utilizado em conjunto com o comando `console.log()`, liberta o programador da necessidade de ter que utilizar comandos `alert` (ou `println`) para depurar o código
- Extensões para o Chrome: *JSONView*, *YSlow*

Ferramentas de desenvolvimento

Bibliotecas JavaScript para programação

- Bibliotecas e ferramentas que facilitam a tarefa de desenvolvimento de código *JavaScript* e de conteúdos para jogos
- Biblioteca ***jQuery***:
 - *Simplifica a escrita do código e diminui a sua extensão, tornando-o mais legível*
 - *Facilita a navegação pelos elementos do DOM, a sua seleção e respetiva manipulação no código*
 - *Permite lidar com eventos que ocorrem na página*

Ferramentas de desenvolvimento

- A utilização de *jQuery* corresponde a seleccionar elementos da árvore do *DOM* para executar ações sobre eles; os seletores derivam do CSS e proporcionam um conjunto de expressões regulares (*patterns*) para fazer corresponder aos elementos que se pretende seleccionar
- `$(".caixa").css({"top":"100px","left":"200px"});`

Selecciona todos os elementos com a classe caixa e define as respectivas propriedades CSS top e left com os valores indicados

Jogos baseados no DOM

- Jogo Ping Pong (“Hello World” dos jogos)



Jogos baseados no DOM

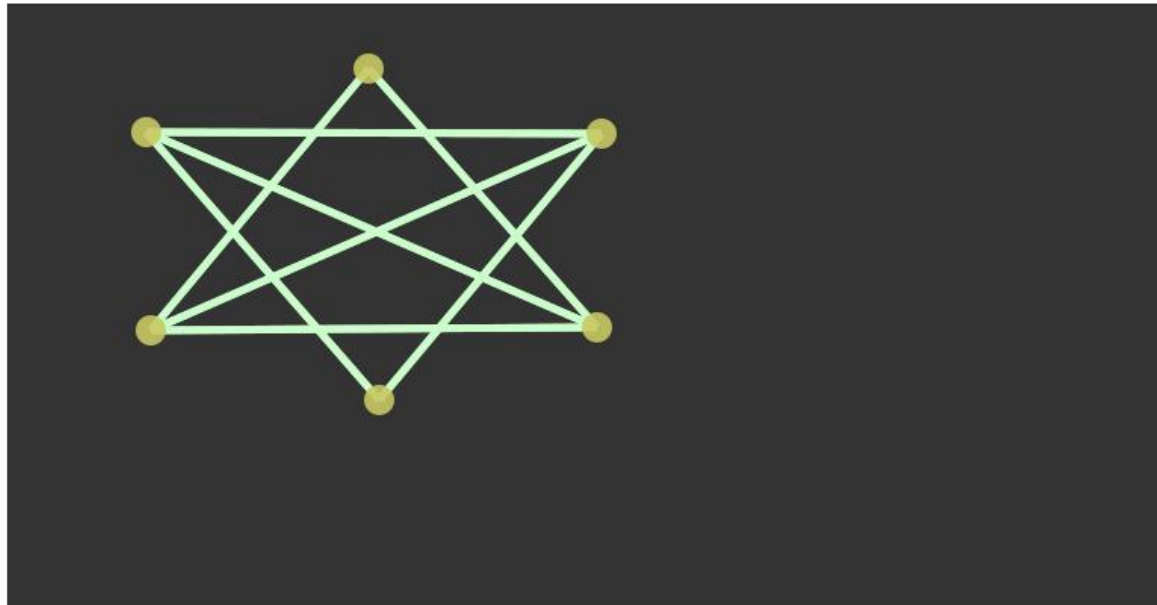
A programação do Ping Pong permite aprender:

- Como preparar um jogo baseado no DOM
- Inserção e manipulação dos elementos do jogo
- Mover objetos do DOM de acordo com teclas pressionadas pelo utilizador
- Suportar múltiplas teclas em simultâneo
- Movimentar objetos do DOM em JavaScript
- Detecção de colisões entre objetos do DOM
- Apresentar pontuações dos jogadores com texto dinâmico em HTML

Jogos baseados no CANVAS

- Jogo Desenredar (Puzzle básico)

Jogo Desenredar baseado no Canvas



Nível 2, Resolucao: 0%

Exemplo de um jogo baseado no Canvas.

Jogos baseados no CANVAS

A programação do Desenredar permite aprender:

- Como preparar um jogo baseado no Canvas
- Desenho 2D no Canvas: círculos e formas geométricas
- Desenho de paths e linhas no Canvas
- Como gerar números aleatórios em JavaScript
- Como guardar a posição de objetos geométricos
- Suportar a interação de utilizador com objetos desenhados através de eventos do rato
- Ciclo de um jogo baseado no Canvas
- Detecção de interseções no Canvas
- Criação de níveis de um jogo no Canvas

Conclusão e referências online

- Página do livro no site da editora FCA:
http://www.fca.pt/cgi-bin/fca_main.cgi/?op=2&isbn=978-972-722-782-2
- Look inside: http://issuu.com/lidel/docs/pdf_novo
- Página facebook do livro:
<https://www.facebook.com/progjogos>

Para saber mais, o que se deve consultar:

- Especificação HTML5: <http://dev.w3.org/html5/spec>
- Williams, J. L. (2012). Learning HTML5 Game Programming: A Hands-on Guide to Building Online Games Using Canvas, SVG, and WebGL. Upper Saddle River, NJ: Addison-Wesley.
- Leituras recomendadas no fim do cap. 6

Conclusão e referências online

- **HTML5 Games** (<http://html5games.com>)
- **HTML5Gamer.net** (<http://html5gamer.net/>)
- **Canvas Demos** (<http://www.Canvasdemos.com>) – disponibiliza o acesso a um conjunto de aplicações e jogos HTML5 que utilizam a marca Canvas, juntamente com um conjunto de tutoriais, ferramentas de programação (bibliotecas *JavaScript* e excertos de código) e outros recursos que auxiliam a aprendizagem da programação de jogos HTML5 baseados no *Canvas*

Conclusão e referências online

- **Script Tutorials** (<http://www.script-tutorials.com/CSS3-games-collection/>) – exemplos de jogos HTML5 desenvolvidos apenas com base no *DOM* (HTML5, CSS3 e *JavaScript*)
- **GamesHTML5** (<http://html5gameonline.com/>) – coleção de jogos HTML5 incluindo jogos *multiplayer* e jogos desenvolvidos para dispositivos móveis;
- **Moby Games** (<http://www.mobygames.com/>) – portal da base de dados mais completa sobre jogos interativos desenvolvidos desde 1979 para várias plataformas, incluindo jogos HTML5 na secção dedicada à plataforma “*browser*”;

Conclusão e referências online

- **HTML5 Game Development** (<http://www.html5gamedevelopment.com/>) – proporciona uma vasta gama de recursos incluindo exemplos de jogos na secção de demonstrações, para além de tutoriais, motores, ferramentas e notícias atualizadas para a comunidade de programação de jogos HTML5;
- **HTML5 Game Dev News** (<http://www.html5game devs.com/>) – *blog* que contém as novidades sobre a programação de jogos em HTML5;
- **Comunidade HTML5 – Portugal** (<http://html5pt.org/>) – O portal da comunidade portuguesa de desenvolvimento em HTML5 que disponibiliza notícias e recursos.

Livros sobre Multimédia

(do mesmo autor)

Multimédia e Tecnologias Interativas, 5ª ed., 2013



Tecnologias de Compressão Multimédia, 1ª ed., 2009

