



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Desarrollo de videojuegos

© Fernando Berzal, berzal@acm.org

Herramientas de desarrollo



Motores de videojuegos [game engines]

http://en.wikipedia.org/wiki/Game_engine

Sistemas diseñados
para la creación y desarrollo de videojuegos

- La misma infraestructura se puede utilizar para distintos juegos
- La misma implementación de un juego se puede portar a distintas plataformas (PC/móvil/consola).



Herramientas de desarrollo



Motores de videojuegos [game engines]

http://en.wikipedia.org/wiki/Game_engine

Funcionalidad típica:

- Motor gráfico: Renderizado de gráficos 2D/3D
- Motor físico (p.ej. detección de colisiones)
- Lenguaje de scripting
- Sonido
- Animaciones
- Redes
- IA
- ...



Herramientas de desarrollo



Alternativas

- Desarrollo nativo usando el SDK estándar para una plataforma (junto con bibliotecas para videojuegos)
- Bibliotecas multiplataforma (para lenguajes de propósito general)
- Lenguajes para el desarrollo de videojuegos
- IDE's para el desarrollo de videojuegos



Desarrollo nativo



- Android

<http://www.android.com/>

- Web: HTML5 & JavaScript

<http://www.w3.org/TR/html5/>

- Consolas

Xbox One Development Kit \$500

PlayStation 4 Development Kit \$2500

Nintendo Development Kit \$2000-\$10000



Desarrollo nativo: Android



<http://developer.android.com/>

- Android SDK (Java):

Eclipse IDE

+ ADT [Android Developer Tools]

<http://developer.android.com/sdk/>



- La biblioteca estándar incluye OpenGL

<http://developer.android.com/guide/topics/graphics/opengl.html>

- Decenas de "game engines"

<http://mobilegameengines.com/android>

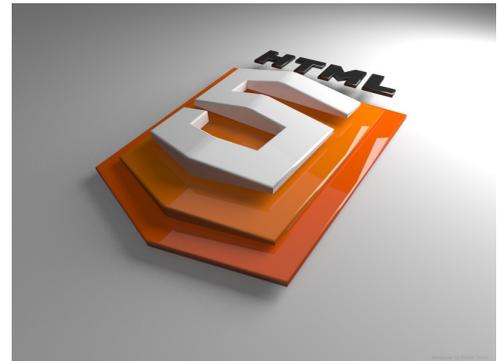
<http://software.intel.com/en-us/blogs/2012/03/13/game-engines-for-android>



Desarrollo nativo: HTML5



- HTML5 + JavaScript



- Decenas de "game engines"
<http://html5gameengine.com/>

- p.ej. **enchant.js** <http://enchantjs.com/>
- Quintus** <http://html5quintus.com/>
- LimeJS** <http://www.limejs.com/>
- EaselJS** <http://createjs.com/>



Bibliotecas multiplataforma



LWJGL Lightweight Java Game Library

<http://lwjgl.org/>

- Java (J2ME)



- Licencia BSD



- Acceso a controladores de juegos (gamepads, joysticks), OpenGL (gráficos), OpenCL (cálculo) y OpenAL (audio)



Bibliotecas multiplataforma



libGDX

<http://libgdx.badlogicgames.com/>

libGDX

- Java
- Licencia Apache
- El mismo código permite generar juegos para Android, Windows, Linux, iOS y HTML5
- Utiliza LWJGL internamente



Bibliotecas multiplataforma



SDL Simple Directmedia Layer

<http://www.libsdl.org/>



- C / C++ / C# / Python
- Licencia zlib
http://www.gzip.org/zlib/zlib_license.html
- Acceso al hardware a bajo nivel (OpenGL & Direct3D)
- Windows, Mac OS X, Linux, iOS & Android



Bibliotecas multiplataforma



Pygame

<http://www.pygame.org/>



- Python
- Conjunto de módulos construidos sobre SDL (no requiere OpenGL)
- Licencia GPL [GNU Lesser General Public License], permite crear juegos gratuitos y comerciales.



Bibliotecas multiplataforma



OpenFL

<http://www.openfl.org/>

OpenFL



- Haxe (<https://en.wikipedia.org/wiki/Haxe>)
- Diseñada copiando el API de Flash, permite crear juegos Flash, HTML5, Windows, OS X, Linux, iOS, Android, BlackBerry, Firefox OS y Tizen.
- Licencia MIT, permite crear juegos gratuitos y comerciales.



Otras bibliotecas



Box2D

<http://box2d.org/>

- C++
- Licencia zlib
- 2D Physics Engine



Cocos2D-x

<http://www.cocos2d-x.org/>

- C++ / Lua / JavaScript
- Licencia MIT
- 2D Physics Engine (p.ej. Box2D)



Lenguajes



Lua

[luna en portugués]

<http://www.lua.org/>

- Lenguaje interpretado
- Portable (funciona en cualquier plataforma que disponga de un compilador de C).
- Licencia MIT



Juegos que usan Lua:

- World of Warcraft
- Angry Birds



Otras bibliotecas



Corona SDK

<http://coronalabs.com/products/corona-sdk/>

- Lua
- Multiplataforma para dispositivos móviles (iOS, Android, Kindle & Windows Phone 8)
- Soporte para estándares de la industria: OpenGL, OpenAL, Box2D, Lua...



IDEs para videojuegos

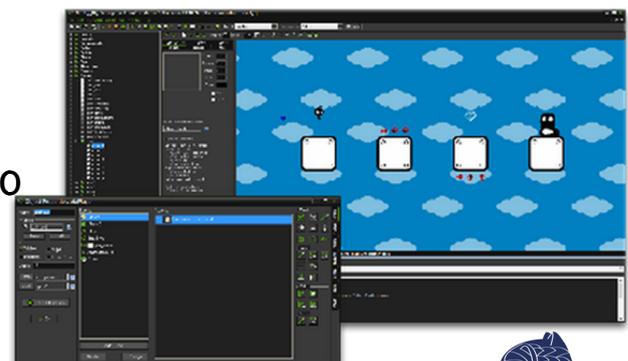


GameMaker: Studio

<http://www.yoyogames.com/studio>

YoYo Games

- Multiplataforma
- Programación visual
- Sprites, objetos, eventos...
- Lenguaje de scripting propio
- Versión gratuita de prueba



IDEs para videojuegos

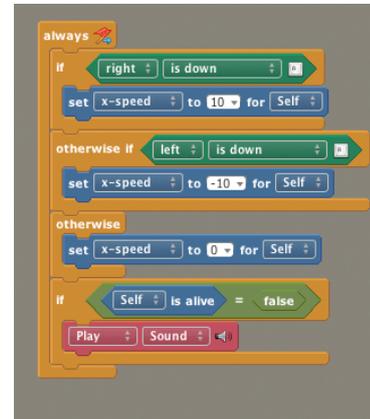


Stencyl

<http://www.stencyl.com/>

stencyl

- Programación visual con Scratch (aunque también se puede programar usando Haxe).
- Física y colisiones: Box2D
- Gratis para juegos web, \$99 al año para juegos de escritorio, \$199 al año para juegos móviles (Android & iOS)



IDEs para videojuegos

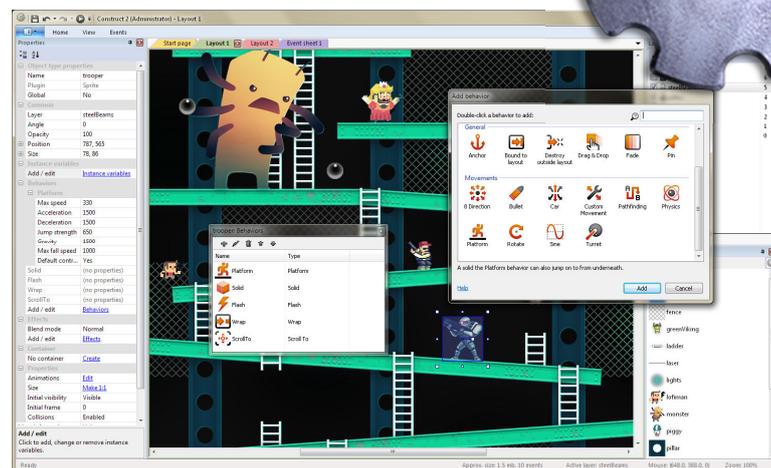


Construct2

<https://www.scirra.com/construct2>

Scirra

- HTML5
- Juegos 2D
- Prototipado
- Python (opcional)
- Versión de prueba gratuita



IDEs para videojuegos



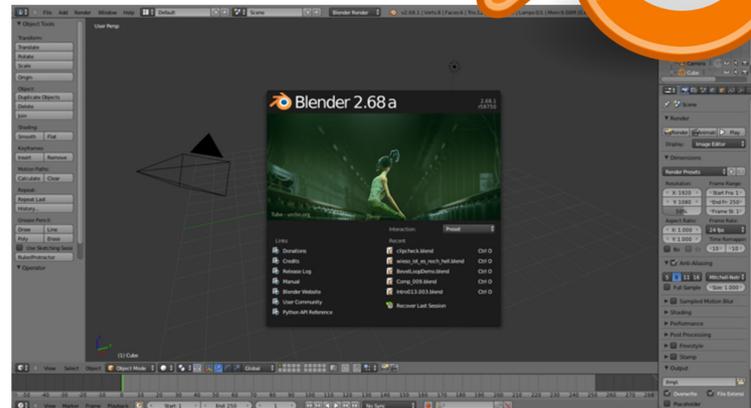
Blender

<http://www.blender.org/>

Blender Foundation



- 3D
- Python
- OpenGL
- OpenAL
- Licencia GNU



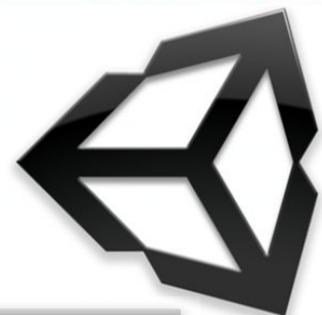
IDEs para videojuegos



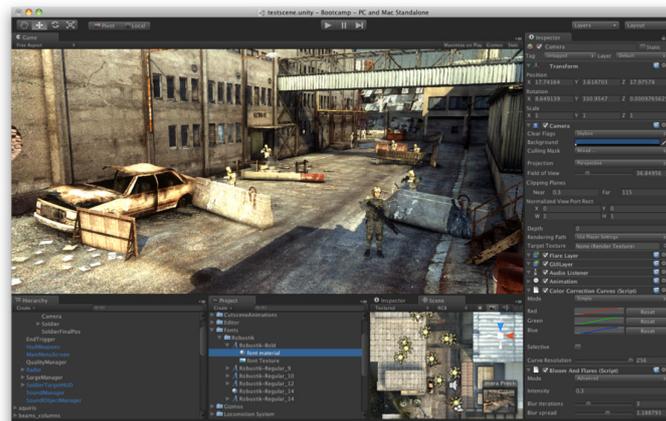
Unity

<http://unity3d.com/>

Unity Technologies



- Multiplataforma
- C# (Mono), JavaScript & Boo
- 2D & 3D
- Versión gratuita de prueba
- \$1500 o bien \$75 al mes



IDEs para videojuegos



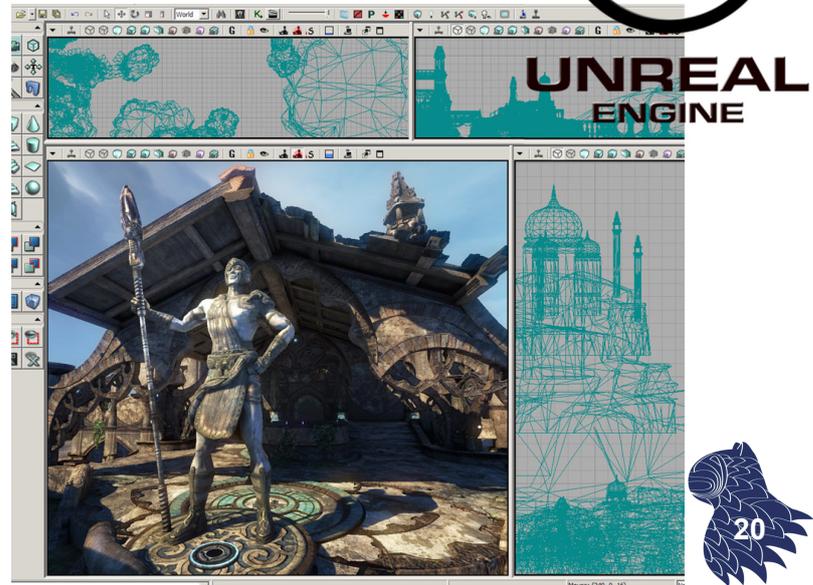
Unreal Engine

<http://www.unrealengine.com/udk/>

Epic Games



- Multiplataforma
- C++
- 2D & 3D
- Gratis para uso no comercial
- Royalties (5%) a partir de \$3000 por trimestre



IDEs para videojuegos



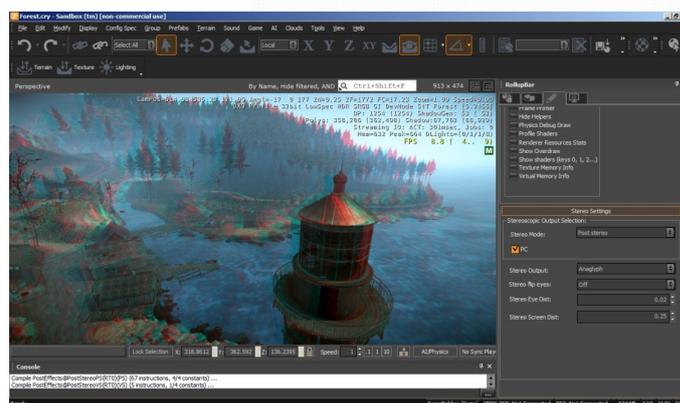
CryENGINE

<http://mycryengine.com/>

Crytek



- Multiplataforma
- C++/Lua
- 3D
- Gratis para uso no comercial
- Royalties (20%) o suscripción...



Más alternativas



- http://en.wikipedia.org/wiki/List_of_game_engines
- <http://www.indiedb.com/engines?sort=ranktoday-asc>



Proyectos previos...



#	Herramienta	Lenguaje	URL
12	Unity	C# / JavaScript / Boo	http://unity3d.com/
4	Cocos2d-x	C++	http://www.cocos2d-x.org/
3	Phaser.io	JavaScript	http://phaser.io/
3	Pygame	Python	http://pygame.org/
2	Unreal Engine	C++	http://www.unrealengine.com/
2	Game Maker	GML	https://www.yoyogames.com/studio
2	LibGDX	Java	http://libgdx.badlogicgames.com/
2	jMonkeyEngine	Java	http://jmonkeyengine.org/
1	Corona SDK	Lua	http://coronalabs.com/
1	Stencyl	Scratch / Haxe	http://www.stencyl.com/
1	SFML	C++	http://www.sfml-dev.org/





Selección de herramientas de desarrollo

Elegir al menos tres formas alternativas de implementar el videojuego propuesto y realizar un análisis de las ventajas y desventajas que ofrece cada una de ellas.

Para cada alternativa analizada:

- Nombre
- Página web
- Categoría (biblioteca, lenguaje de programación, IDE...)
- Coste (open source vs. licencia comercial)
- 3 características destacadas que pueden resultar útiles
- 3 limitaciones observadas de cara al videojuego propuesto

Decisión final (tabla resumen y justificación razonada)

