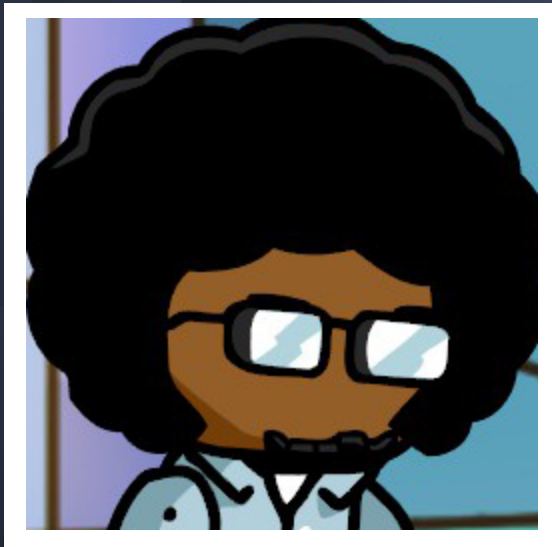


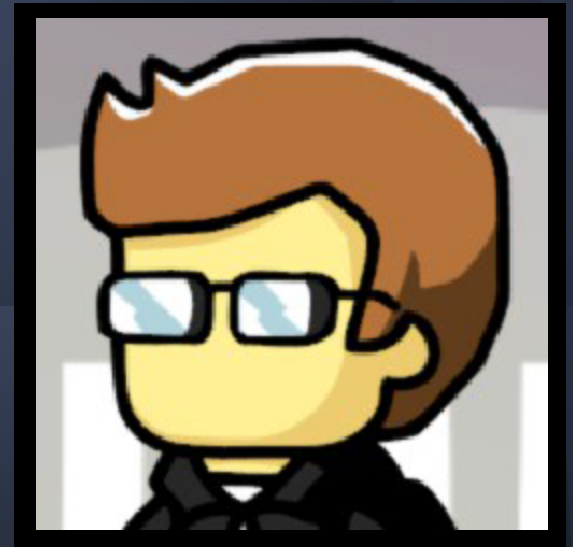
El Equipo YinYang
{Jorge Palacios, Christian Chomiak}
presenta

Desarrollo videojuegos: *No soy gallo, soy chocobo*



Ola k
ase?

Kiere aser
juego o k
ase?



Estamos hoy reunidos porque...



Agenda

- Introducción al desarrollo de videojuegos.
- Retos de la industria.
- Diferencias entre la academia y "el mundo real".
- Breve introducción a Unity.
- Recomendaciones y conclusiones

Introducción al desarrollo de videojuegos

¿Qué es un videojuego?

(Lámina dejada en blanco para que la audiencia de su definición)

Introducción al desarrollo de videojuegos

¿Qué es el desarrollo de juegos?

- También conocido como gamedev.
- Disciplina dedicada al desarrollo de software de entretenimiento (videojuegos).
- Game developer != Gamer

Introducción al desarrollo de videojuegos (Principales) Áreas y disciplinas

- Programación
 - IA
 - Rendering
 - Gameplay
- Diseño de juego
 - Game design
 - Level design
- Arte
 - Concept Art
 - Modelado 3D



Game Designer != Game Programmer

Introducción al desarrollo de videojuegos

Áreas y disciplinas

- Narrativa
- Sonido
- Playtesting



Situación en Venezuela

- Ley de videojuegos y juguetes bélicos.
 - Fundación FILANTROPIA.
- Posibilidades de trabajo
- Agrupaciones
- Eventos
 - GameJams



Caracas GameJam 2013

Introducción al desarrollo de videojuegos

¿Qué se necesita saber?

Conocimientos y habilidades

- Básicas
 - Matemáticas.
 - Trigonometría.
 - Álgebra.
- Deseables
 - Física.
 - Movimiento.
 - Comunicación.
 - Prototipado.

Introducción al desarrollo de videojuegos

¿Qué se necesita saber?

Conocimientos y habilidades ... para Game Designers

- Psicología.
- Lógica.
- Arte.
- Programación (básica).
- Redacción y ortografía
- Comunicación (avanzada).

Introducción al desarrollo de videojuegos

¿Qué se necesita saber?

Conocimientos y habilidades ... para Game Programmers

- Algoritmos (`while(mas){mejor;}`)
- Scripting (deseable).
- Programación orientada a objetos (necesaria).
- C, C++ (indispensable).

Fantasma (y cómo vencerlos)



Fantasmas (y cómo vencerlos)

"No sé programar"

- Aprende.
- Sinfín de libros y tutoriales en Internet.



Fantasma (y cómo vencerlos)

"Soy programador, no artista."

Recursos gratis en Internet (Creative Commons)	Asset Store (Unity)
Artistas Freelance	Panas que estudien diseño

Fantasma (y cómo vencerlos)

"Soy estudiante (no tengo dinero)."

**Herramientas
OpenSource**

**Versiones gratis de
herramientas pagas**

Convenios Universitarios:

- **Descuentos para estudiantes**
- **Software gratis***

* Ciertas condiciones aplican

Fantasma (y cómo vencerlos)

"Nunca he hecho un juego, pero quiero hacer el próximo CoD*."

- Bájate de esa nube.
- Empieza por algo sencillo.
 - Experiencia++



***Aplica también para MMORPGs/Diablo**

Fantasma (y cómo vencerlos)

"<Inserte otra excusa acá>"

iwfraser88
Level 6 Game Developer - 855xp

@iwfraser88 BI Nf Pf Ré 48 dA Ng Ko G+ Fb db Gh Sc Ic Gj

1 2 3 4 5 6 7 8 9 10 11 12

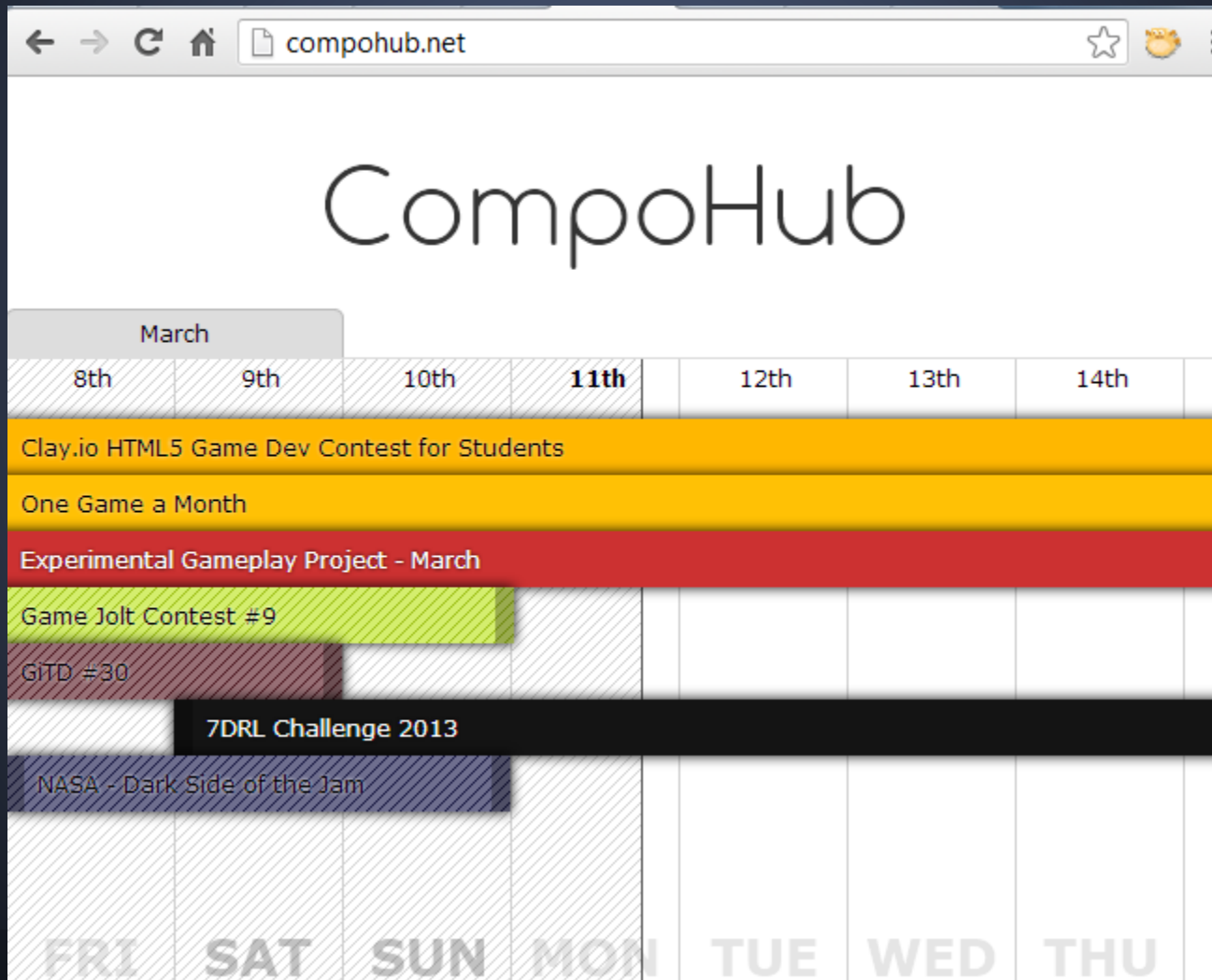
coder

2d art 3d art music sfx design writer biz testers team

Level 6 (800xp) Level 7 (1300xp)

Fantasma (y cómo vencerlos)

"<Inserte otra excusa acá>"



El mundo real

La realidad de los desarrolladores

Los desarrolladores AAA...

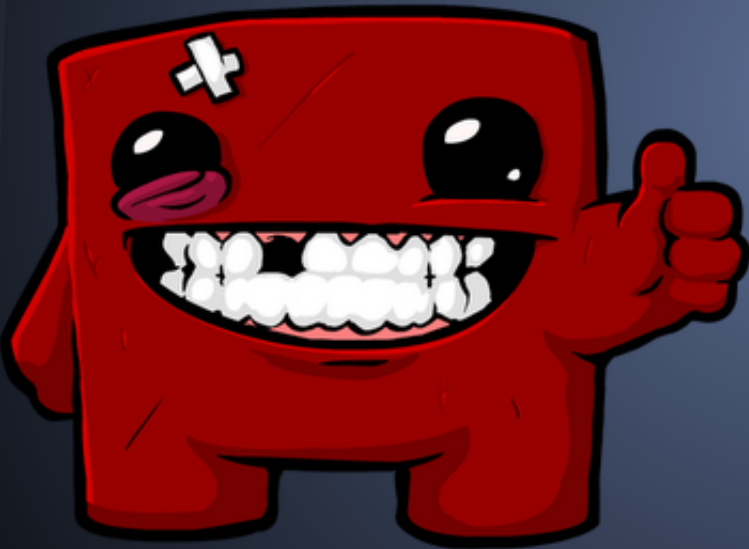


- Ven poco beneficio económico aún cuando gran parte del juego depende de ellos. i.e. Cobran salario.
- Están restringidos a proyectos con fórmulas exitosas. Poco espacio para la innovación.

El mundo real

La realidad de los desarrolladores

Los desarrolladores Indie...



- Trabajan en equipos de pequeño y mediano tamaño.
- Tienen poco presupuesto.
 - Kickstarter.
- No tienen quince y último.*

*Ciertas condiciones aplican



El Boom Indie

INDIE GAME

THE MOVIE



Indie Game: The Movie

El mundo real

Trade-off con la academia

Cosas que no se aprenden en la academia

- Resolver.
 - Muchas veces, las soluciones simples sirven igual o mejor que las "correctas".
- Colaborar.
 - Comunidad en Internet
 - Foros.
 - Stack Overflow
 - No reinventar la rueda.

Cosas que no se aprenden en el mundo real

- Estudio de algoritmos.
 - Algoritmos avanzados.
 - Estado del arte.
- "Only the best is good enough." -LEGO

Desarrollo

Niveles y herramientas

- **Bajo nivel** - Sangre, sudor y lágrimas
 - Librerías como SDL y GLUT.
- **Nivel medio** - Sudor y lágrimas
 - Frameworks como XNA, PyGame, Enchant.js, Flixel.
- **Alto nivel** - Sudor
 - Motores como Unity, UDK y Source.
- **Nivel Nyancat** - Flores silvestres, helado y ponicornios
 - Herramientas como GameMaker y RPGMaker.

Desarrollo
Unity3D



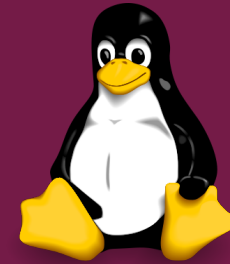
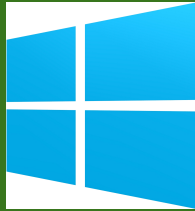
Desarrollo - Unity3D

Plataformas

Creación

Licencia Pro
especial

Deployment



PS3

Wii™

Versiones

Unity

- Gratis
- Muestra splash screen y watermarks.

Unity Pro

- Licencia de \$1500
- Permite:
 - Render-to-texture
 - Iluminación global
 - Efecto de post-procesamiento
 - Occlusion culling
 - Plug-ins

Programación

- Scripting

- Motor creado a partir de Mono (implementación OpenSource del Framework .NET).
- Lenguajes
 - UnityScript (Javascript en esteroides)
 - C#
 - Boo (sintaxis inspirada en Python)

- IDE

- MonoDevelop (incluído en el instalador)
- Soporte para Visual Studio

Desarrollo - Unity3D

Interfaz

(Lámina dejada en blanco. Se mostró la interfaz de Unity en vivo y directo)

Plug-ins

- Librerías escritas en código nativo (C, C++, Objective-C).
- Permiten integración entre Unity y aplicaciones de terceros.
- Por ejemplo:
 - ARToolKit for Unity: Realidad Aumentada.
 - FingerGestures: detección de input via touch.
 - OpenCV: procesamiento de imágenes.

Desarrollo - Unity3D Asset Store

Playmaker

Editor Extensions/Visual Scripting
Hutong Games LLC
★★★★★

50% OFF SALE - NEW 1.5 UPDATE

Realize your creative vision without coding!

NEW! Fsm Template Instances.
NEW! Use FSM Templates as Actions.
NEW! Full Photon Networking support.
NEW! More powerful Get/Set Property actions.
NEW! Graph View zoom and minimap.
NEW! Editor localized in Chinese, Japanese, French...

Quickly make gameplay prototypes, A.I. behaviors, animation graphs, interactive objects, cut-scenes, walkthroughs and more...

[More Info >>](#)



Search

Categories

- Home
- ▶ 3D Models
- ▶ Textures & Materials
- ▶ Audio
- ▶ Complete Projects
- ▶ Editor Extensions
- ▶ Scripting
- ▶ Particle Systems
- ▶ Services
- ▶ Animation

1.3
PIXEL ART EDITOR

TEXTURE TOOLS

SWF2PNG SOLIDIFY RESIZE
FILTERS UNPACK BATCH PROCESSING

Texture Tools - Pixel ...
Editor Extensions/Utilities
Cats in the Sky
★★★★★

uniSWF

FI

uniSWF - Flash to Unity
Editor Extensions/GUI
Flaming Pumpkin Ltd
★★★★★

<XML> <JSON> Serialization

XML-JSON Serialization
Scripting/Input - Output
A World For Us
★★★★★

SUIMONO

INTERACTIVE WATER SYSTEM



SUIMONO Water System
Textures & Materials/Water
Tanuki Digital
★★★★★

Most Popular

<p>Bootcamp ★★★★★ Free</p>	<p>NGUI: Next-Gen UI ★★★★★ \$95.00</p>	<p>2D Toolkit ★★★★★ \$65.00</p>	
<p>Playmaker ★★★★★ \$45.00</p>	<p>SPACE for Unity - Space S... ★★★★★ \$85.00</p>	<p>Car Tutorial ★★★★★ Free</p>	
<p>Ultimate FPS Camera ★★★★★ \$15.00</p>	<p>Medieval Environment Pack ★★★★★ \$70.00</p>	<p>Terrain Assets ★★★★★ Free</p>	

Hot Deals!

[See complete list](#)



1. **Unlimited+ for Mecanim ...**
Editor Extensions/Animation

- Top Paid
- Top Free
- Top Crossing
- Latest
- My Stuff

Desarrollo - Unity3D

Unity+Git

¿Por qué usar un Sistema de Control de Versiones?

¿Qué es Git?

¿Cómo lograr Unity+Git?

Sistema de Control de Versiones

Es cualquier práctica que permita llevar un control de los cambios en un proyecto.

- git
- mercurial
- subversion
- perforce
- visual sourcesafe (DON'T YOU DARE!)

Desarrollo - Unity3D

Sistema de Control de Versiones

- github
- bitbucket
- google code
- source forge
- assembla

y hay más...



Desarrollo - Unity3D
¿Qué es Git?



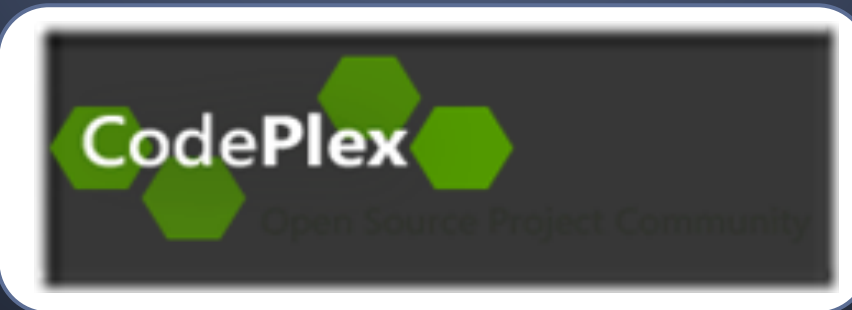
Es un sistema de control de versiones (duh).

Diseñado por Linus Torvalds.

¿Por qué es mejor que otras soluciones, como Mercurial y SVN?

Desarrollo - Unity3D

¿Cómo usar Git?



Unity+Git

- Definir el archivo .gitignore
- Configurar el proyecto en Unity:
 - Edit > Project Settings > Editor
 - Seleccionar “Meta files” abajo de Version Control Mode

Recomendaciones y conclusiones



Hey, listen!

Conclusiones y

Recomendaciones

- **Inicia un blog!**
 - Compartes experiencias.
 - Llevas registro de lo que has aprendido.
- **Trabaja en un portafolio**
 - Un blog puede ser un primer trampolín.
 - Es una muestra tangible de tus capacidades.
- **Participa en una comunidad.**
 - Aprendes de la experiencia de los demás.
 - Un lugar para la sana competición y colaboración.

Conclusiones y

Más recomendaciones

- **Mantente:**
 - Organizado.
 - Usa herramientas de control de proyectos, como Trello.
 - Actualizado.
 - La computación es un área en constante evolución.
 - Curioso.
 - Ponte retos.

Conclusiones y

Muchas más recomendaciones

- Haz brainstorming.
 - Muchas soluciones > una única solución.
- Programa
 - La práctica hace al maestro.
- Descansa.
 - *"All work and no play makes Jack a dull boy."*
 - En otras palabras, ten una vida.

Y hay más!

Recuerda que...

...hay gente que está o estuvo en tu misma situación.

...el desarrollo de software es un proceso iterativo.

...lo que no se comienza hoy no se termina mañana.

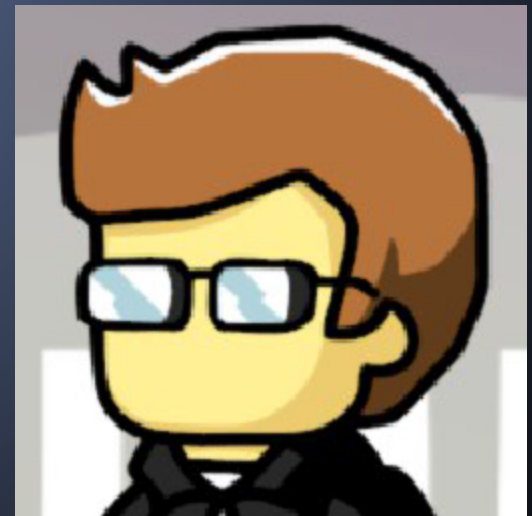
PU-BLI-CI-DAD



www.pctroll.me
[@pctroll](#)

bonusdisc.wordpress.com
[@cchomiak23](#)

www.darkrecon.com
[@darkrecongame](#)



Muchas gracias!



Esto es un chocobo