



LABORATORIO 1

Videojuegos, Diseño

Historia, Concepto y Diseño de Videojuegos

1. Pre-Laboratorio

- Tome algún videojuego que conozca e investigue o analice.
 1. Genero del juego.
 2. Resumen de las mecánicas del juego.
 3. Plataforma en las que esta disponible el juego.

2. ¿Qué es un Videojuego?

Es un juego electrónico que requiere la interacción humana con una interfaz de usuario para generar *feedback* visual en dispositivo de vídeo. Los dispositivos electrónicos utilizados para jugar videojuegos son llamados plataformas, ejemplos de ello son los computadores personales y las consolas de videojuegos.

El dispositivo principal de entrada en un videojuego se llama control de juego o *game controller*, este dispositivo suele variar por genero o plataforma, siendo por ejemplo en los juegos de computador personal mas común el uso del teclado y/o ratón, mientras que en los juegos de consolas de videojuegos es mas común el uso de un mando.

La palabra *video* en videojuegos se refiere a un dispositivo de imágenes, sin embargo los videojuegos como hoy conocemos proveen mas de una fuente de *feedback* tales como audio en su gran mayoría, vibración o *force feedback*, etc.

3. Breve Historia de los Videojuegos - Primera Generación

Cuadro 1: Historia de los Videojuegos

1952●	A. S. Douglas crea el primer juego electrónico documentado para su tesis doctoral, un juego de tic-tac-toe corriendo en el computador <i>EDSAC</i> de la Universidad de Cambridge [7].
1958●	<i>Tennis for Two</i> de William Higinbotham en un osciloscopio [3].
1962●	<i>Spacewar!</i> de Steven Russell en el computador <i>PDP-1</i> [11].
1967●	Ralph Baer y sus compañeros de trabajo crean la primera consola de videojuegos que funciona sobre televisión estándar, la llaman <i>Brown Box</i> [4].
1971●	Nolan Bushnell crea la primera arcade llamada Computer Space [10].
1972●	Sale en venta la primera consola de videojuegos para el hogar llamada Magnavox Odyssey, se venden aproximadamente 330.000 unidades [9].
1972●	Pong publicado por Atari y creado por Nolan Bushnell se convierte en el primer juego arcade exitoso [2].
1974●	Sale Gran Trak 10 primer juego arcade de carreras [1].
1974●	Sale Maze Wars considerado el primer shooter en primera persona [6].
1977●	Atari saca a la venta la consola <i>Video Game Computer System</i> (<i>Atari 2600</i> o <i>VCS</i>).

4. Desarrollo de Videojuegos

Los videojuegos son usualmente creados en grupos de desarrollo conformados por varias personas con distintos roles, una persona puede tener la capacidad de manejar uno o mas roles, algunos roles solo son necesarios durante ciertas etapas del juego (ejemplo testers) mientras que otros son necesarios durante todo el proceso de desarrollo (ejemplo programadores). Los roles necesarios para la creación de un videojuego pueden variar según el objetivo y alcance del videojuego pero usualmente son los siguientes [8, p. 149]:

4.1. Roles

Diseñador

Diseña las mecánicas de juego, reglas, limitantes, objetivos, estructura y alcance del videojuego. Básicamente son los visionarios del juego, en proyectos grandes usualmente este trabajo es dividido en varios roles como diseñadores de mecanicas de juego, diseñadores de interfaces, diseñadores de aventuras (quests), escritores, etc.

Artista

Encargado de producir todo el arte utilizado en el juego, el trabajo del artista puede ser orientado a 3D o 2D. Artistas 2D suelen producir arte conceptual, sprites, texturas, e interfaces de usuario. Artistas 3D suelen producir modelos o mallados, animación, ambientes 3D y cinemáticas.

Programador

Crea la base de código del videojuego, esto incluye:

- Físicas: Programación del motor de físicas, simulaciones físicas, colisiones, movimiento de objetos, etc.
- Inteligencia Artificial: Programación de objetos o agentes interactivos utilizando técnicas de inteligencia artificial para juegos como scripting, planificación, decisiones basadas en reglas, etc.
- Gráficos: Programación del contenido gráfico con importantes consideraciones en memoria y performance, la producción del motor gráfico, la integración de modelos, texturas y demás contenido que debe funcionar junto con el motor de físicas y motor de juego.
- Sonido: Integración de musica, dialogo y efectos de sonido en distintos sitios y situaciones.

- Mecánicas de Juego: Implementación de varias reglas, objetivos y respuestas del juego.
- Scripting: Desarrollo y mantenimiento de un sistema de comando en alto nivel para la interacción con varios elementos del juego.
- Interfaces de Usuario Desarrollo de elementos de interfaz, menús y sistemas de *feedback*.
- Procesamiento de Input: Establece correlación de distintas acciones, eventos y sistemas de respuestas con variados dispositivos de entrada.
- Networking: Administración de data recibida de forma local o a través de internet.
- Herramientas de juego: Producción de herramientas de desarrollo para el videojuego, especialmente para diseñadores y scripting.

Diseñador de niveles

Crea los niveles, misiones y retos del videojuego utilizando programas específicos. Los diseñadores de niveles trabajan con versiones completas e incompletas del juego e interactúan directamente con editores de niveles usualmente desarrollados por los programadores del videojuego, esto para eliminar la necesidad de los diseñadores tener que interactuar directamente con el código del videojuego.

Ingeniero de Sonido

Se encarga de los efectos de sonido y su debido posicionamiento en tiempo y espacio sea 2D o 3D.

Testers

El control de calidad y *QA* o *quality assurance* (aseguramiento de calidad) es llevado por los testers, estos se encargan de analizar un videojuego y documentar todo defecto de software encontrado además de analizar si el juego cumple o no con el diseño propuesto.

5. Lenguajes y Herramientas

5.1. HTML5 y Javascript

Phaser

Descripción: *Phaser is a fun, free and fast 2D game framework for making HTML5 games for desktop and mobile web browsers, supporting Canvas and WebGL rendering.*

URL: <https://phaser.io/>

PlayCanvas

Descripción: *PlayCanvas is the world's easiest to use WebGL Game Engine. It's free, it's open source and it's backed by amazing developer tools.*

URL: <https://playcanvas.com/>

Unity3D

Descripción: *Unity is a flexible and powerful development platform for creating multiplatform 3D and 2D games and interactive experiences. It's a complete ecosystem for anyone who aims to build a business on creating high-end content and connecting to their most loyal and enthusiastic players and customers.*

URL: <https://unity3d.com/unity>

COCOS2D-JS

Descripción: *Cocos2d-X is a suite of open-source, cross-platform, game-development tools used by thousands of developers all over the world.*

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

5.2. Java

Slick2D

Descripción: *PSlick2D is an easy to use set of tools and utilites wrapped around LWJGL OpenGL bindings to make 2D Java game development easier.*

URL: <http://slick.ninjacave.com/>

jMonkeyEngine

Descripción: *It's a free, open source game engine, made especially for Java game developers who want to create 3D games using modern technology. The software is programmed entirely in Java, intended for wide accessibility and quick deployment.*

URL: <http://jmonkeyengine.org/>

5.3. C++

COCOS2D-X

Descripción: *Cocos2d-X is a suite of open-source, cross-platform, game-development tools used by thousands of developers all over the world.*

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

Torque2D

Descripción: *Torque 2D is an extremely powerful, flexible, and fast open source engine dedicated to 2D game development.*

URL: <https://www.garagegames.com/products/torque-2d/features>

Godot Engine

Descripción: *Godot is an advanced, feature packed, multi-platform 2D and 3D game engine. It provides a huge set of common tools, so you can just focus on making your game without reinventing the wheel.*

URL: <http://www.godotengine.org/wp/>

5.4. C#

Paradox

Descripción: *Paradox is a versatile and engaging game engine, that will empower you to make stunning games that better fit your vision!*

URL: <http://paradox3d.net/>

Monogame

Descripción: *MonoGame is an Open Source implementation of the Microsoft XNA 4 Framework. Our goal is to allow XNA developers on Xbox 360, Windows & Windows Phone to port their games to the iOS, Android, Mac OS X, Linux and Windows 8 Metro. PlayStation Mobile, Raspberry PI, and PlayStation 4 platforms are currently in progress.*

URL: <http://www.monogame.net/>

Wave Engine

Descripción: *Component Based Game Engine architecture, 2D and 3D physics engines, beautiful visuals effects, cross-platform support, advanced layout system and much more.*

URL: <http://waveengine.net/>

Unity3D

Descripción: *Unity is a flexible and powerful development platform for creating multiplatform 3D and 2D games and interactive experiences. It's a complete ecosystem for anyone who aims to build a business on creating high-end content and connecting to their most loyal and enthusiastic players and customers.*

URL: <https://unity3d.com/unity>

5.5. Python

Pygame

Descripción: *Pygame is a cross-platfrom library designed to make it easy to write multimedia software, such as games, in Python. Pygame requires the Python language and SDL multimedia library. It can also make use of several other popular libraries.*

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

6. Diseño de Videojuegos

El diseño de un juego generalmente inicia con una idea, esta usualmente es una modificación o re-implementación de algún concepto ya existente. Esta idea de juego puede caer entre uno o distintos géneros de juego, es usual la mezcla de géneros en el diseño de un videojuego. El diseñador usualmente produce una propuesta de juego documentada en la que se incluye concepto, mecánicas de juego, características del juego, escenario o ambiente del juego e historia y requerimientos para jugar el juego (en cuanto a capacidad cognitiva), este documento también suele incluir los requerimientos para el desarrollo del juego, las personas encargadas y roles asignados para el desarrollo además de una de costos [5, p. 101].

7. Actividad

El alumno debe realizar un documento de propuesta de juego, el documento contiene información extensa y concisa sobre el proyecto de juego.

- Primeramente debe conceptualizar un videojuego y sus mecánicas.
 1. Escriba un resumen de su "idea" de juego en al menos un párrafo.
- Luego debe realizar el '*Documento de Propuesta de Juego*' donde se extiende y explica en detalle la base conceptual del juego (es común que algunas de las cosas escritas acá cambien durante el desarrollo pero la ideal es mantener el concepto original). El documento debe contener lo siguiente:
 1. **Concepto**, describa la visión del juego en una o dos oraciones.
 2. **Genero**, describa con una sencilla oración donde encaja el juego dentro de un genero definido.
 3. **Gameplay (Mecánicas de Juego)**, en esta sección resuma que hace el jugador mientras esta jugando, también debe indicar como el juego forma parte del genero nombrado y que características lo diferencian de la base conceptual de este genero.
 4. **Features (Características o Distintivos)**, aca debe listar cuales son los principales puntos de atractivo de su juego y que significan estos puntos para el juego.
 5. **Setting (Ambiente o Escenario)**, resuma en pocos párrafos que hace el mundo de su juego y sus ocupantes interesantes.
 6. **Historia**, en caso de tener historia en el juego, haga una pequeña sinopsis.

7. **Target Market (Mercado de Destino)**, explique para quienes esta siendo desarrollada su idea de juego (nichos, géneros de juego específicos, adultos, niños, publico general, fans de algún deporte, etc) y las razones por la que escoge a este publico como destino.
- **Análisis Competitivo o de Mercado**, investigue juegos similares al menos en genero de juego, sus números en venta y popularidad, como su juego saldrá mejor o peor que la competencia y que soluciones tiene para competir ¹.
 - **Escoger un framework, motor o conjunto de bibliotecas para el desarrollo de juego en PC**, debe escoger un framework, motor de juego o conjunto de bibliotecas para el desarrollo de su juego, es de importancia para evitar complicaciones en el desarrollo que lo que escoja incluya o sea fácil de integrar con bibliotecas o frameworks que provean ²:
 1. Lógica de juego directamente o a través de scripting.
 2. Editor o manejo de niveles o escenas.
 3. Editor o manejo de objetos de escena.
 4. Manejo de inputs, dispositivos de entrada.
 5. Fácil manejo y organización de assets (arte, texturas, audio, modelos, etc)
 6. Detección de colisiones.
 7. Simulaciones físicas y sistemas de partículas.
 8. Inclusión de animaciones.
 9. Networking

Referencias

- [1] Gran Trak 10. http://www.arcade-museum.com/game_detail.php?game_id=7992.
- [2] ALCORN, A. Atari History, Timeline 1972-1984, November-30-1972. <https://www.atari.com/history/1972-1984-0>.
- [3] ANDERSON, J. Tennis for Two, The story of an early computer game. <http://www.pong-story.com/1958.htm>.
- [4] BAER, R. H. The Brown Box, 1967-68. http://americanhistory.si.edu/collections/search/object/nmah_1301997.
- [5] BETHKE, E. *Game Development and Production*. Wordware Game Developer's Library, 2003.

¹Es usual incluir el resultado y resumen de esta investigación en el '*Documento de Propuesta de Juego*'

²Es usual agregar en el '*Documento de Propuesta de Juego*' plataforma de destino, herramientas que se van usar y justificación de estas elecciones

- [6] COLLEY, S. Stories from the Maze War 30 Year Retrospective. <http://www.digibarn.com/history/04-VCF7-MazeWar/stories/colley.html>.
- [7] DOUGLAS, A. Noughts and Crosses - The oldest graphical computer game. <http://www.pong-story.com/1952.htm>.
- [8] GATES, B. *Game Design (2nd Edition)*. Thomson Course Technology, 2004.
- [9] LOWOOD, H. Magnavox Odyssey, First home video game console. <http://www.pong-story.com/odyssey.htm>.
- [10] LOWOOD, H. Videogames in Computer Space: The Complex History of Pong. <http://lmc.gatech.edu/~bogost/courses/spring10/lcc8823/lowood.pdf>, 2010.
- [11] MARKOFF, J. Alan Kotok, 64, a Pioneer In Computer Video Games. *The New York Times* (2006).