

16-Bits de Historia

La Batalla de Puebla

Manual Técnico

16-Bits de Historia

Haro Calvo, Fernando; Gonzalez Ceseña, Adan

Universidad Autónoma de Baja California
Facultad de Ingeniería, Arquitectura y Diseño
Ingeniero en Software y Tecnologías Emergentes

Programación Estructurada

Núñez Yepiz, Pedro

12 de diciembre de 2023

CONTENIDO

CONTENIDO	2
INTRODUCCIÓN	3
1. DESCRIPCIÓN GENERAL	3
1.1 ALCANCE	3
1.2 OBJETIVOS	4
2. TÉRMINOS Y DEFINICIONES	4
3. MODELO DE ANÁLISIS	5
3.1 MODELO DE DISEÑO	7
4. ASPECTOS TÉCNICOS	8
4.1 HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO	8
4.1.1 Visual Studio Code	8
4.1.2 Raylib	8
4.1.3 Adobe Photoshop	8
4.1.4 GitHub	8
4.2 MODIFICACIÓN LOCAL	9
5. ARQUITECTURA DEL SISTEMA	11
5.1 REQUISITOS MÍNIMOS DEL SISTEMA	11
6. BIBLIOGRAFÍA	12

INTRODUCCIÓN

El presente manual técnico proporciona una visión detallada del código fuente y la arquitectura del juego desarrollado en este proyecto. Se presentan, por lo tanto, los aspectos técnicos e informáticos del videojuego “16-Bits de Historia”, con el objetivo de ayudar a los desarrolladores a entender cómo está organizado el código del juego, cómo se utilizan los recursos y cómo se puede modificar y ampliar el juego.

1. DESCRIPCIÓN GENERAL

El videojuego consiste en una aventura interactiva que transporta a los jugadores a través de diferentes escenarios y desafíos situados en el evento histórico de la Batalla de Puebla ocurrido el 5 de mayo de 1862, con el fin de proporcionar una experiencia educativa y agradable a los jóvenes, con una público objetivo de 10 - 14 años de edad.

Este juego tiene la misión de enseñar de manera innovadora la historia de México, centrándose en el evento histórico de la Batalla de Puebla que tuvo lugar el 5 de mayo de 1862. A través de la inmersión del videojuego, los jugadores tienen la oportunidad de explorar y vivir los sucesos de esta batalla crucial de una manera interactiva y entretenida. El juego busca fomentar el interés por la historia, brindando una perspectiva única que va más allá de los métodos de enseñanza tradicionales.

1.1 ALCANCE

El alcance del videojuego se limita a la Batalla de Puebla, específicamente el evento histórico que tuvo lugar el 5 de mayo de 1862. El juego se enfoca en proporcionar una representación detallada y precisa de este acontecimiento crucial en la historia de México. Se limita la longitud del juego a 3 niveles o escenarios, donde cada uno tiene una cinemática previa explicando el contexto y dando instrucción al jugador.

El juego se desarrolla utilizando una estética visual de 16 bits utilizando una perspectiva de plano cenital, donde el personaje se puede mover en las cuatro direcciones e interactuar con diversos objetos. La programación se realiza considerando las limitaciones y características de plataformas que admiten gráficos de 16 bits. La arquitectura del código se estructura de manera modular para facilitar la comprensión y el mantenimiento. Se busca la escalabilidad para permitir futuras modificaciones y adiciones al juego de manera eficiente.

1.2 OBJETIVOS

La combinación de elementos educativos y lúdicos tiene como objetivo no solo transmitir hechos históricos de manera precisa, sino también estimular el pensamiento crítico y la curiosidad histórica de los jugadores. A través de diálogos, misiones y eventos, el juego proporciona información detallada sobre los personajes, el contexto y las decisiones clave que transcurrieron en la Batalla de Puebla.

2. TÉRMINOS Y DEFINICIONES

Con la finalidad de mostrar el funcionamiento del juego al usuario, haremos uso de algunos términos con los cuales será necesario estar familiarizado.

Plano Cenital: Perspectiva visual donde la cámara se encuentra directamente encima del personaje y la escena, ofreciendo una vista aérea.

16-bits: Referencia a la capacidad de procesamiento de la unidad central de procesamiento (CPU) y la representación gráfica del juego. Los gráficos de 16 bits son conocidos por su estilo pixelado y su estética retro.

Cinemática: Breve secuencia animada que precede o sigue a un nivel del juego, utilizada para proporcionar contexto narrativo.

Niveles o Escenarios: Segmentos jugables del juego que representan momentos específicos de la Batalla de Puebla.

Modularidad: Diseño del código que favorece la división en módulos o bloques independientes para facilitar el desarrollo y mantenimiento.

NPC: Significa "Non-Playable Character" (Personaje No Jugable). Estos personajes son controlados por el sistema del juego en lugar de por el jugador.

3. MODELO DE ANALISIS

El modelo de análisis se centra en la representación y la lógica de juego. Aquí se plantean las entidades clave y sus interacciones. Este análisis proporciona una base sólida para el diseño y la implementación del juego. Las entidades principales que se deben tener en cuenta son las siguientes:

Jugador: Representa al usuario dentro del juego, capaz de moverse y realizar acciones como interactuar con objetos y con NPCs.

Escenarios: Áreas específicas que representan momentos históricos de la Batalla de Puebla en las que el jugador puede explorar e interactuar.

Objetos Interactivos y NPCs: Elementos en el juego con los que el jugador puede interactuar para obtener información adicional o avanzar en la trama.

Cinemáticas: Se utilizan para presentar información histórica y proporcionar contexto narrativo.

Recursos gráficos: Las texturas e imágenes de cada elemento visual, escenarios, personajes y fondos.

Ajustes del juego: Pantalla completa, activar y desactivar música, subir y bajar volumen.

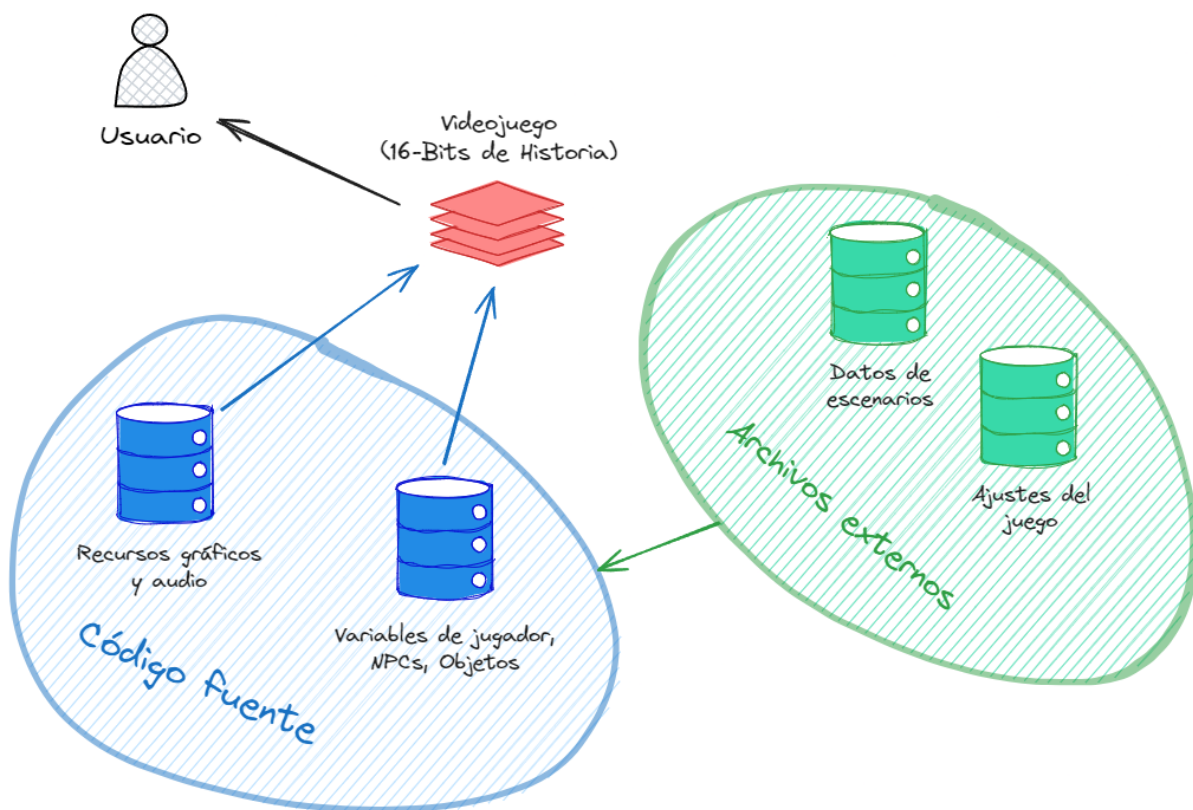


Figura 1.- Elementos involucrados en el análisis del sistema

3.1 MODELO DE DISEÑO

El siguiente diagrama de flujo tiene como objetivo presentar de manera general el algoritmo de jugabilidad principal.

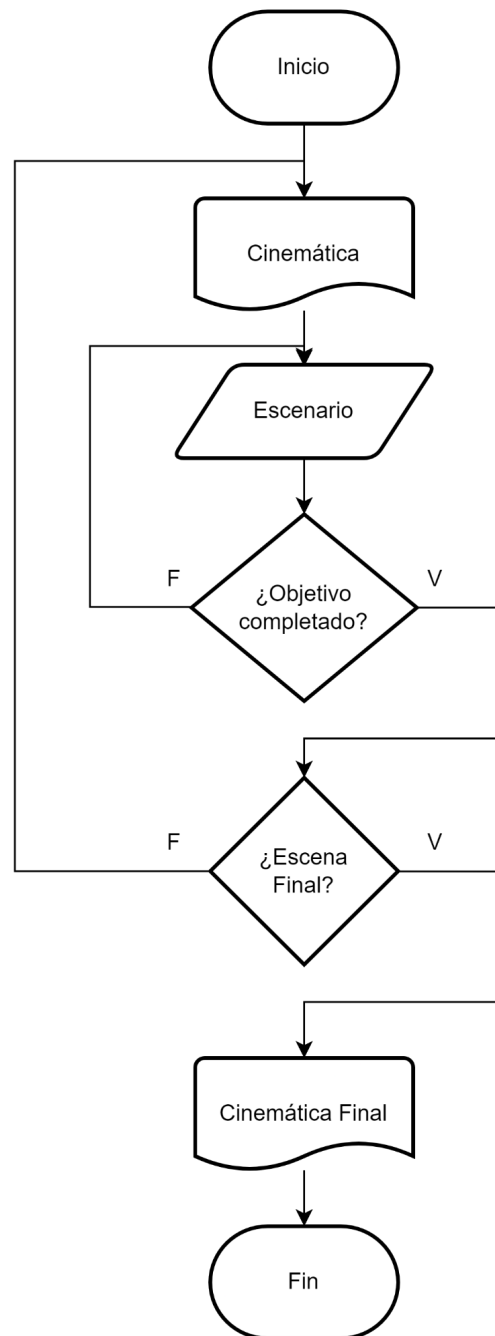


Figura 2.- Diagrama de flujo del algoritmo principal

4. ASPECTOS TÉCNICOS

En esta sección, se detallan los aspectos técnicos clave relacionados con el desarrollo del juego "16-Bits de Historia". Estos aspectos incluyen la plataforma de desarrollo, las librerías utilizadas y las herramientas de diseño.

4.1 HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

4.1.1 Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente altamente configurable y de código abierto desarrollado por Microsoft. Fue usado para la codificación y depuración del juego, al igual que su compilación.

4.1.2 Raylib

Raylib es una biblioteca de código abierto y fácil de usar diseñada para el desarrollo de videojuegos y usos gráficos en general. Es particularmente útil para proyectos que requieren un rendimiento eficiente y gráficos en 2D.

4.1.3 Adobe Photoshop

Adobe Photoshop es una herramienta de edición de imágenes ampliamente utilizada en diversas disciplinas creativas. En el proyecto fue utilizado para la creación y edición de gráficos y creación de texturas.

4.1.4 GitHub

GitHub es una plataforma de desarrollo colaborativo que utiliza el sistema de control de versiones Git. Aunque no es una herramienta de desarrollo en sí misma, fue esencial para el manejo del código fuente y la colaboración entre desarrolladores, al igual que el manejo de versiones y la distribución.

4.1.5 Tiled

Tiled es un editor de mapas sencillo, pero potente que permite diseñar niveles usando mapas de texturas. Funcionó como una herramienta gratuita para permitir la creación sencilla de diseños de mapas.

4.2 MODIFICACIÓN LOCAL

Para modificar localmente este proyecto, necesitarás instalar algunas herramientas y seguir los siguientes pasos:

1. **Instalar VS Code:** Para poder modificar el código necesitarás de un editor de desarrollo, el que fue utilizado en el desarrollo fue Visual Studio Code, que se puede conseguir de la página oficial de Microsoft: <https://code.visualstudio.com/>
2. **Instalar CMake:** CMake es una herramienta de código abierto que permite generar archivos de compilación nativos. Se puede descargar desde aquí: <https://cmake.org/download/> (elegir la versión correcta según el sistema operativo).
3. **Instalar el compilador de C:** Dependiendo de tu sistema operativo, necesitarás instalar un compilador de C. En Windows, se puede utilizar MinGW que se consigue de: <https://www.mingw-w64.org/>. En Linux puedes usar gcc, que viene incluido en la mayoría de distribuciones.
4. **Instalar la extensión de C/C++ para VS Code:** Esta extensión proporciona características de edición avanzadas para los lenguajes C y C++. Puedes instalarla desde el marketplace de VSCode.

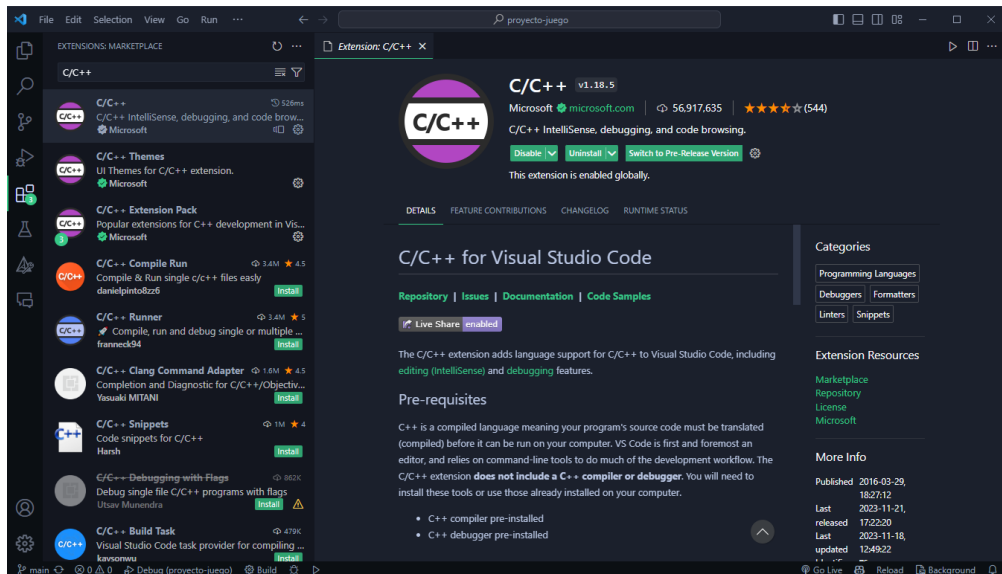


Figura 3. Extensión C/C++ dentro de Visual Studio Code Extensions Marketplace

5. **Clonar el repositorio:** Accediendo al sitio de GitHub, podrá encontrar el código fuente, este se puede clonar utilizando git o descargar directamente como un archivo comprimido.
6. **Abrir el proyecto en VS Code:** Abre VS Code y dirígete a File > Open Folder para abrir el directorio del proyecto.
7. **Configurar CMake:** Abre el archivo CMakeLists.txt y asegúrate de que todas las rutas y configuraciones son correctas para tu sistema.
8. **Compilar el proyecto:** Usa la tarea de compilación de VSCode para compilar el proyecto. Puedes acceder a esto a través de Terminal > Run Build Task.
9. **Ejecutar el proyecto:** Una vez compilado el proyecto, puedes ejecutarlo usando la tarea de ejecución de VS Code (Terminal > Run Without Debugging).

5. ARQUITECTURA DEL SISTEMA

Los componentes que conforman el sistema se integran por el usuario, siendo el individuo que interactúa con la computadora a nivel de aplicación. Interactúa con el videojuego disponiendo del teclado para realizar los movimientos dentro del videojuego así como la interacción con personajes y objetos del entorno.

5.1 REQUISITOS MÍNIMOS DEL SISTEMA

- **Sistema Operativo:** El juego es compatible con sistemas operativos Windows y Linux.
- **Procesador:** Se recomienda un procesador con capacidad para ejecutar gráficos de 16 bits de manera eficiente.
- **Memoria RAM:** Se sugiere al menos 2 GB de RAM para un rendimiento óptimo.
- **Almacenamiento:** Se requiere un espacio de almacenamiento mínimo de 100 MB para la instalación del juego.
- **Teclado:** Se utiliza el teclado como dispositivo de entrada principal para que el jugador realice movimientos y acciones dentro del juego.

6. BIBLIOGRAFÍA

Alegsa, L. (2023, junio 12). *Definición de 16 bits*. Alegsa.com.ar.

https://www.alegsa.com.ar/Dic/16_bits.php

Download CMake. (s/f). Cmake.org. Recuperado el 12 de diciembre de 2023, de

<https://cmake.org/download/>

Extensions, L. M. A. (s/f). *Visual Studio Code - code editing. Redefined*.

Visualstudio.com. Recuperado el 12 de diciembre de 2023, de

<https://code.visualstudio.com/>

Gustavo, B. (2019, abril 12). *Qué es GitHub y cómo empezar a usarlo*. Tutoriales

Hostinger. <https://www.hostinger.es/tutoriales/que-es-github>

Llamas, L. (2022, julio 27). *Raylib, librería Open Source para hacer videojuegos*.

Luis Llamas.

<https://www.luisllamas.es/raylib-libreria-open-source-para-hacer-videojuegos/>

MinGW-w64. (s/f). Mingw-w64.org. Recuperado el 12 de diciembre de 2023, de

<https://www.mingw-w64.org/>

Online, C. P. A. (2020, julio 13). *Planos según la posición de la cámara: cenital y*

nadir. Blog de CPA Online.

<https://www.cpaonline.es/blog/cine-y-tv/planos-segun-la-posicion-de-la-camara-cenital-y-nadir/>

Tiled. (s/f). Tiled. Recuperado el 12 de diciembre de 2023, de

<https://www.mapeditor.org/>

(S/f). Platzi.com. Recuperado el 12 de diciembre de 2023, de

<https://platzi.com/clases/1509-personajes-npcs/17580-que-son-los-npcs-y-por-que-son-importantes/>