LibWiiEsp: Biblioteca libre de desarrollo de videojuegos para Nintendo Wii

Ezequiel Vázquez de la Calle
Codirectores:
Manuel Palomo Duarte
Antonio García Domínguez

Índice

Videoconsolas

Una videoconsola es un sistema electrónico de entretenimiento que ejecuta videojuegos. Pueden tener diversas arquitecturas.

- Nintendo Wii tiene arquitectura *Power PC*.

Videoconsolas

Una videoconsola es un sistema electrónico de entretenimiento que ejecuta videojuegos. Pueden tener diversas arquitecturas.

- Nintendo Wii tiene arquitectura *Power PC*.

Sistemas cerrados

Los fabricantes de videoconsolas controlan desarrollo y ejecución.

- Kits de desarrollo sólo accesibles mediante contratos.
- Soportes con sistemas de ficheros privativos.
- No se permite correr ejecutables sin firmar digitalmente.

El *Homebrew* o software casero

- *Scene*: sacar máximo partido a aparatos electrónicos.
- Herramientas libres para crear y ejecutar código sin firmar.
- Lanzadores de ejecutables.
- *Custom Firmwares*.
- Ampliación de la funcionalidad de una videoconsola.

Introducción

El *Homebrew* o software casero

- *Scene*: sacar máximo partido a aparatos electrónicos.
- Herramientas libres para crear y ejecutar código sin firmar.
- Lanzadores de ejecutables.
- *Custom Firmwares*.
- Ampliación de la funcionalidad de una videoconsola.

Reacciones de los fabricantes

- Actualizaciones bloquean software casero.
- Nintendo 64: cartuchos.
- Playstation 3: demandas judiciales.
- Xbox 360: XNA.

Introducción

¿Por qué este PFC?

- *Libogc*: muy bajo nivel, difícil de usar y centrada en el hardware de Wii.
- Todo escrito en C, existiendo soporte para C++.
- Escasa documentación, difícil de encontrar y en inglés.

Introducción

¿Por qué este PFC?

- *Libogc*: muy bajo nivel, difícil de usar y centrada en el hardware de Wii.
- Todo escrito en C, existiendo soporte para C++.
- Escasa documentación, difícil de encontrar y en inglés.

Objetivos

- Crear una herramienta útil y libre para desarrollar para Wii.
- Generar documentación amplia, práctica y en español.
- Ofrecer una visión general del funcionamiento de Wii.
- Proporcionar juegos de ejemplo.

Índice

Calendario final

Pruebas informales de *Libogc* realizadas durante el curso 09-10.
En noviembre de 2010 se decide construir una biblioteca completa.

- **Fase de planificación**: realizar pruebas de viabilidad, investigación y búsqueda de información necesaria.
- **Fase de desarrollo**: establecer requisitos y construcción de módulos.
  - Por cada módulo:
    - Análisis: identificación de funcionalidad necesaria.
    - Diseño: diseño del componente.
    - Implementación: codificación del módulo.
    - Pruebas: comprobaciones y validaciones.
  - Construcción de juegos de ejemplo.
- **Fase de documentación**: redactar manual, referencia y memoria del proyecto.

Diagrama de Gantt

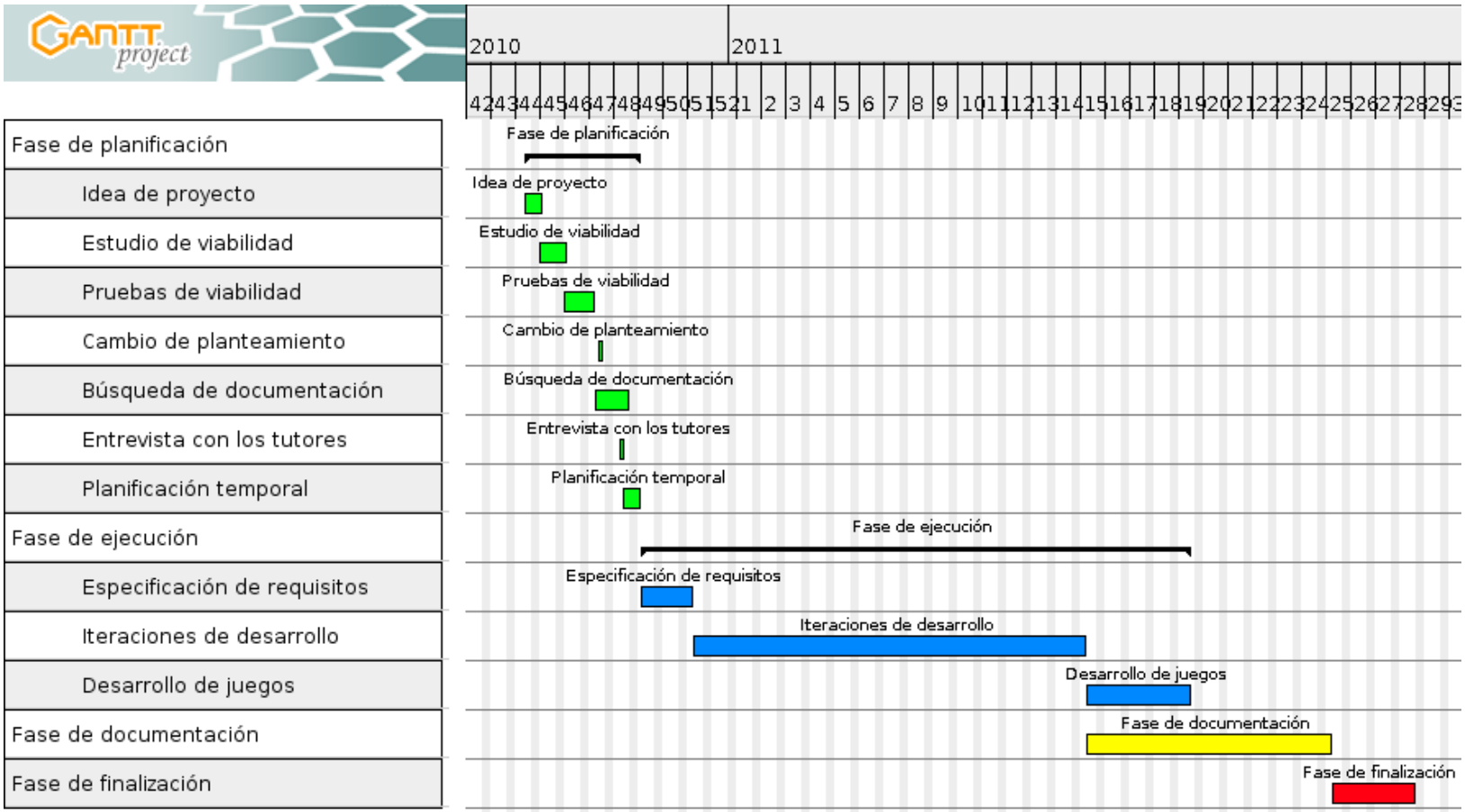


Figura: Diagrama de Gantt

Índice

Big Endian

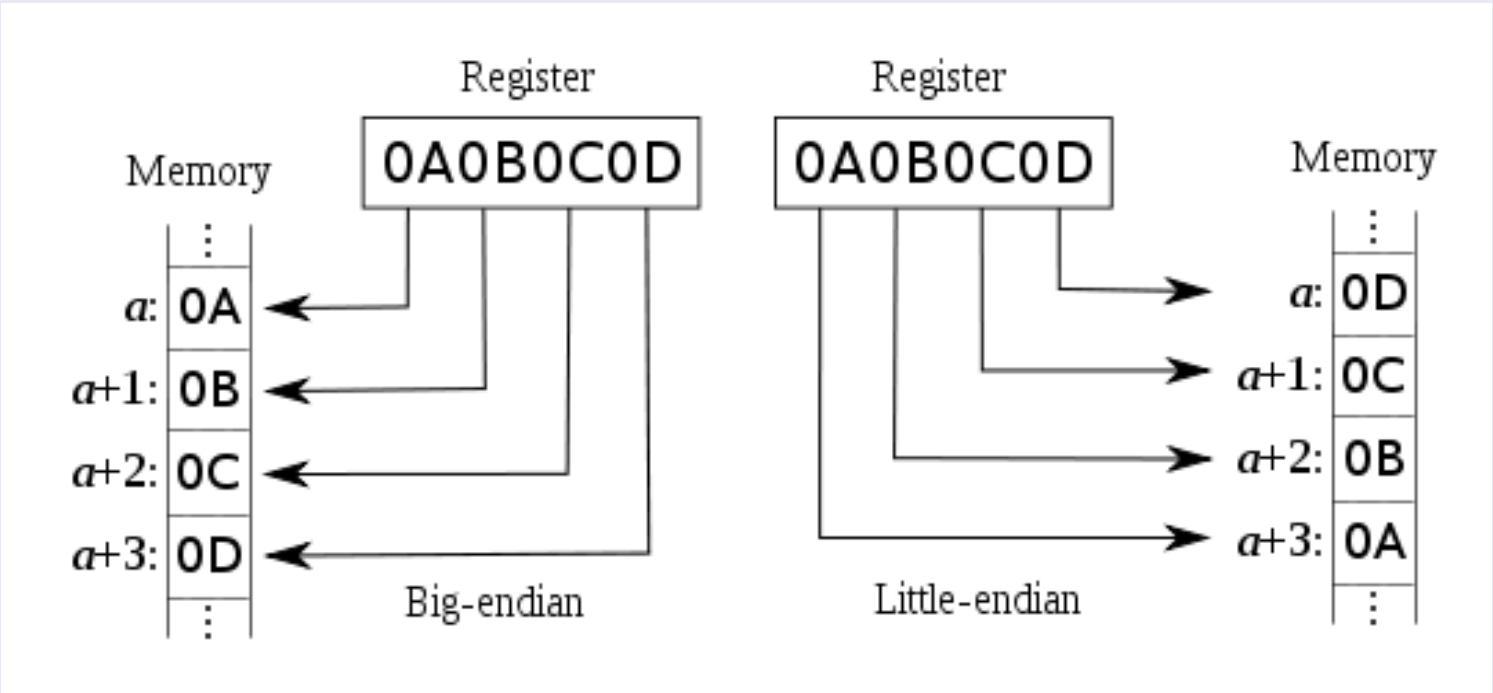


Figura: Diferencias entre *Big Endian* y *Little Endian*

Alineación de los datos

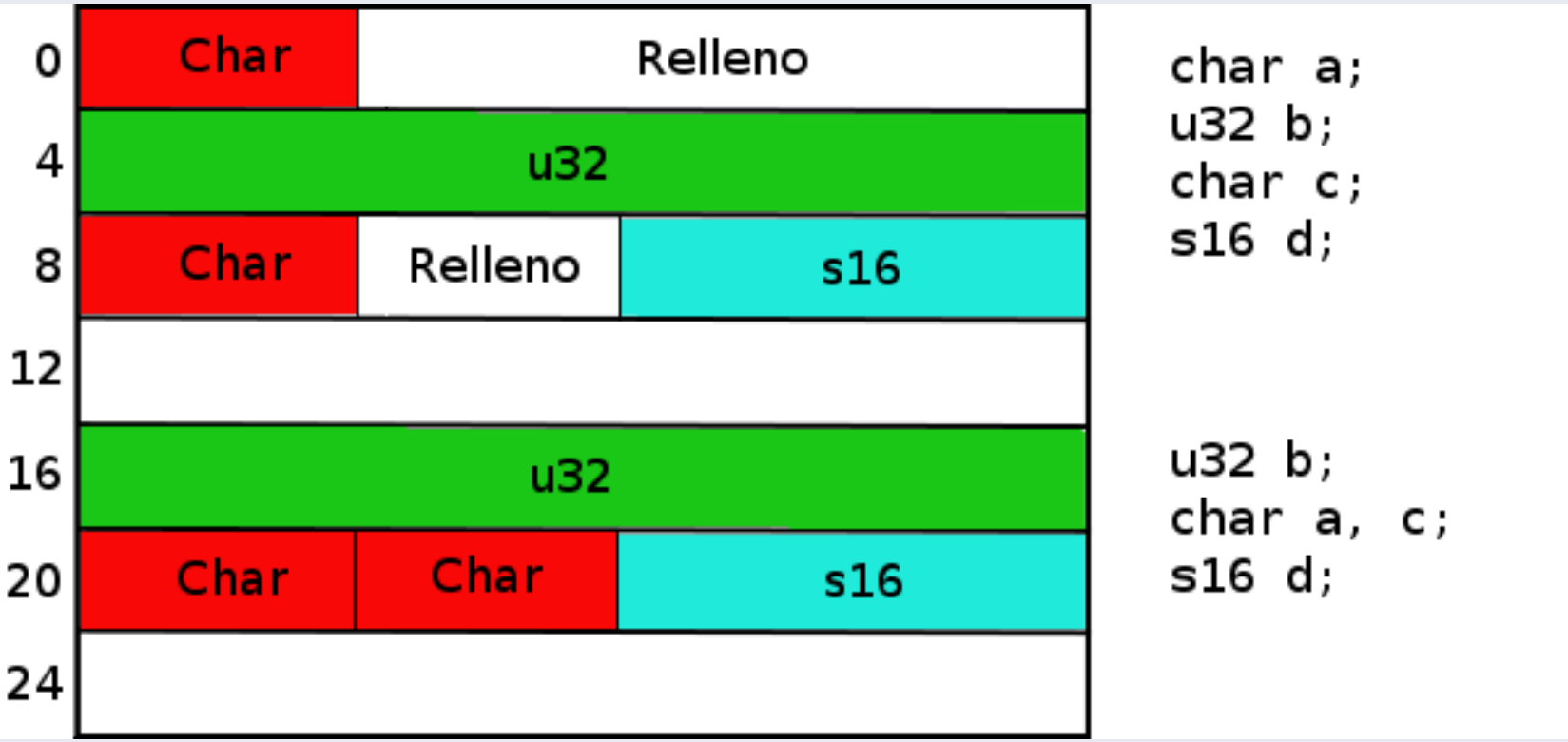


Figura: Estado de la memoria, alineando o no los datos

Otras consideraciones

- Alineación y relleno al leer desde un periférico: 32 bytes.
- Cantidad de memoria: Wii tiene 64 MB de RAM.
- Lanzar un programa: Homebrew Channel.
- Tipos de datos.

Tipo de dato	Descripción	Rango
u8	Entero de 8 bits sin signo	0 a 255
s8	Entero de 8 bits con signo	-127 a 128
u16	Entero de 16 bits sin signo	rango 0 a 65535
s16	Entero de 16 bits con signo	-32768 a 32767
u32	Entero de 32 bits sin signo	0 a 0xffffffff
s32	Entero de 32 bits con signo	-0x80000000 a 0x7fffffff
u64	Entero de 64 bits sin signo	0 a 0xffffffffffffffff
s64	Entero de 64 bits con signo	-0x8000000000000000 a 0x7fffffffffffffff
f32	Flotante de 32 bits	-
f64	Flotante de 64 bits	-

Figura: Tipos de datos utilizados en Nintendo Wii

Necesidades detectadas

Controlar sistemas de la videoconsola

- Sistema gráfico.
  - Dibujo de texturas.
  - Escritura con fuentes de texto.
- Sistema de audio: música y efectos de sonido.
- Controladores *Wii Remote*.
- Medios de almacenamiento: tarjeta SD.

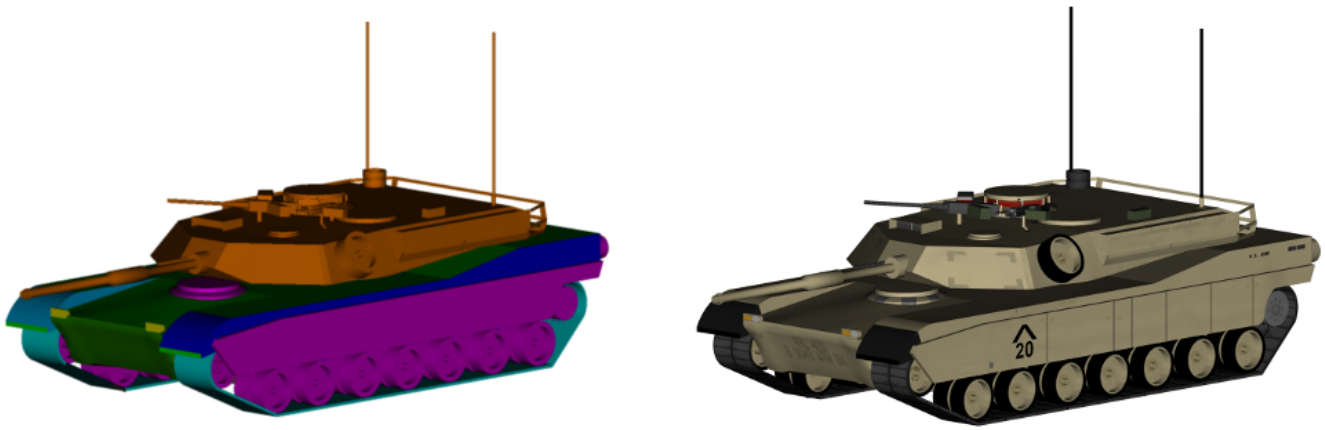


Figura: Figura 3D sin texturas y con texturas

Necesidades detectadas

Otros módulos con funcionalidad necesaria

- Analizador XML.
- Organizar los recursos multimedia.
- Reproducción de animaciones.
- Soporte de internacionalización.
- Detección de colisiones.
- Registro de mensajes del sistema.

Necesidades detectadas

Elementos de un juego 2D

- Actor: elemento con entidad propia dentro del juego.
- Escenario: mundo en el que los actores interactúan.

Plantillas

- Abstraen de los detalles comunes de estos elementos.
- Plantilla para actores: base para definir comportamiento.
- Plantilla para escenarios: crear niveles de forma sencilla.
- Plantilla para clase principal: inicialización de la videoconsola.

Necesidades detectadas

Documentación amplia y útil

Facilitar el aprendizaje por medio de documentación.

- Manual de instalación y uso. (34 páginas)
- Manual de referencia completo. (Generado con Doxygen)
- Memoria del proyecto. (243 páginas)

Necesidades detectadas

Documentación amplia y útil

Facilitar el aprendizaje por medio de documentación.

- Manual de instalación y uso. (34 páginas)
- Manual de referencia completo. (Generado con Doxygen)
- Memoria del proyecto. (243 páginas)

Ejemplos didácticos

Ilustrar el funcionamiento de la biblioteca.

- **Arkanoid Wii**: romper ladrillos con una bola.
- **Duck Hunt Wii**: cazar patos que aparecen en la pantalla.
- **Wii Pang**: personaje rompe pompas con ganchos verticales.

Metodología

- Metodología iterativa, basada en *Rational Unified Proccess*.
- Por cada etapa se construye un módulo completo.

Metodología

- Metodología iterativa, basada en *Rational Unified Proccess*.
- Por cada etapa se construye un módulo completo.

Características del sistema

- Totalmente orientado a objetos.
- Separación completa de código fuente y datos: XML.
- Fase de diseño:
  - Patrón *Singleton*: para los módulos con una única instancia.
  - Patrón Visitante: para la implementación de *Double Dispatch* en el módulo de colisiones.

Proyección ortográfica

- Foco de luz en el infinito.
- Proyectar perpendicularmente a la pantalla.
- No hay cambios de tamaño por distancia al plano.

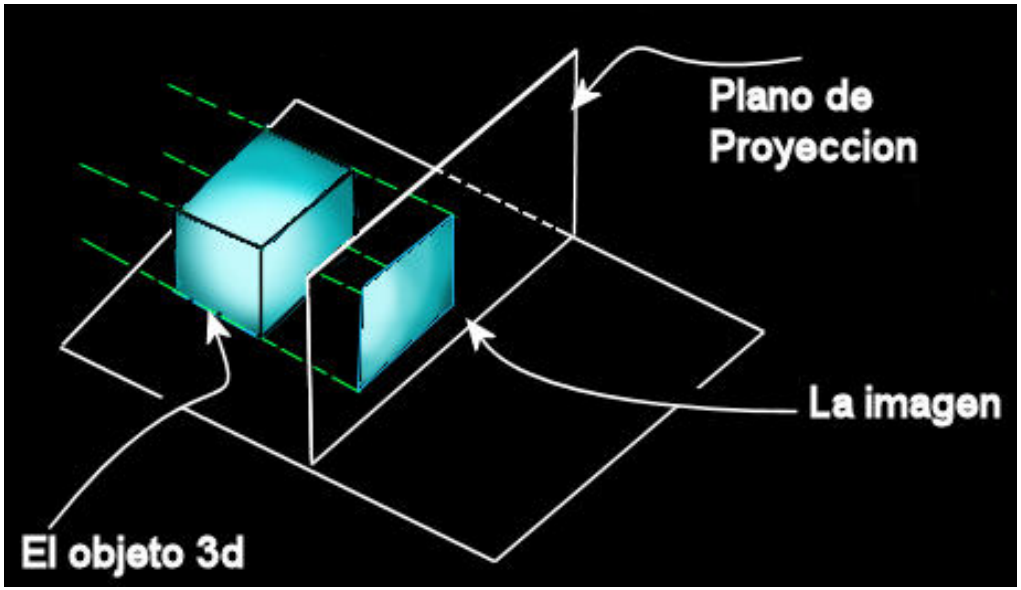


Figura: Proyección ortográfica

Doble búfer

- Dos zonas de memoria para dibujar fotogramas.
- Se consigue evitar parpadeos al mostrar animaciones.

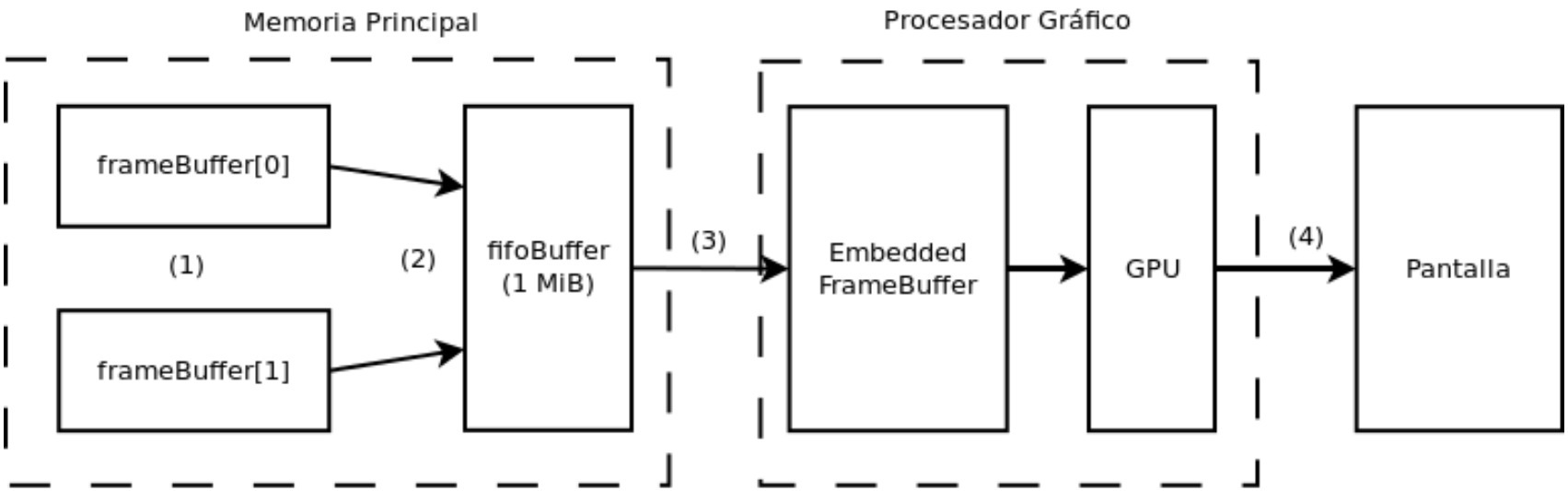


Figura: Esquema del sistema gráfico con doble búfer

Formato de vídeo RGB5A3

- Formato nativo con el que trabaja el procesador gráfico.
- Permite trabajar cómodamente con transparencias.

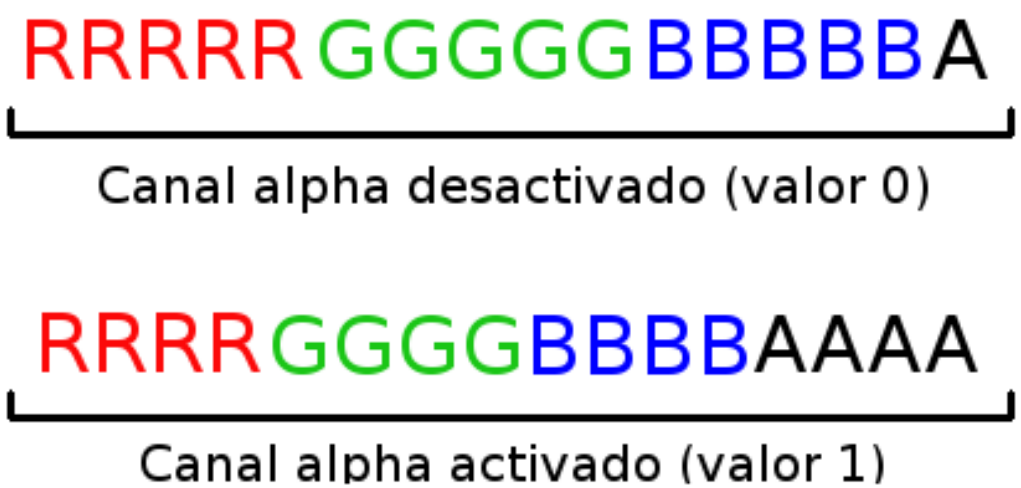
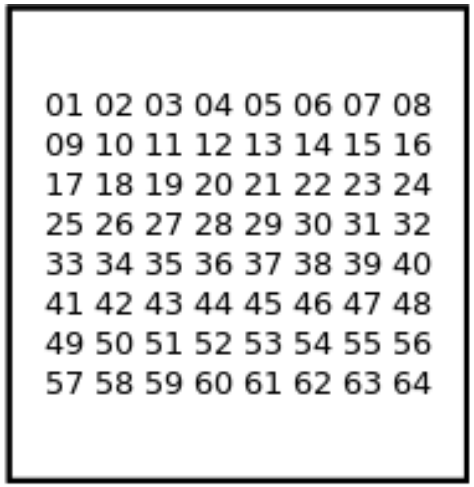


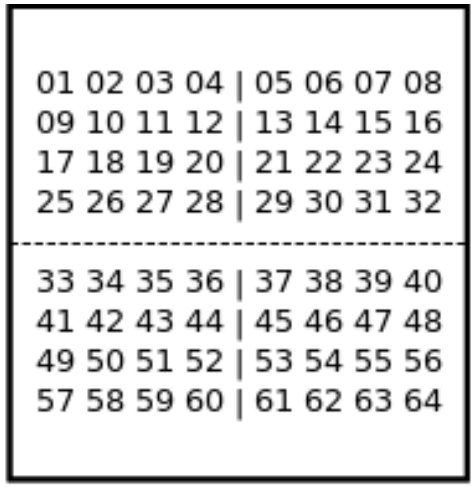
Figura: Formato de píxel RGB5A3

Texturas organizadas en *tiles*

- Píxeles adyacentes en la imagen, también en memoria.
- GPU usa texturas en tiles para trabajar de forma optimizada.



Datos de textura organizados de forma lineal.
Se reciben en el orden:
01 02 03 04 05 06 07 08 09 10 11 12 ...



Datos de textura organizados en tiles.
Se reciben en el orden:
01 02 03 04 09 10 11 12 17 18 19 20 ...

Figura: Textura lineal y textura organizada en *tiles* de 4x4

Detalles de implementación

Sistema de audio

- Procesador DSP dedicado únicamente al control de audio.
- Reproduce hasta 16 flujos de sonido (voces) simultáneamente.
- Una voz reservada para pistas de música.
- Formato nativo de Wii:
  - 48000 Hz.
  - Estéreo (dos canales).
  - *Samples* de 16 bits con signo.

Detalles de implementación

Fuentes de texto

- *FreeType2* genera una imagen *bitmap* para cada carácter.
- Mapa de bits monocromo: un píxel se dibuja si tiene valor 1.
- En pantalla se dibuja cada carácter píxel a píxel.

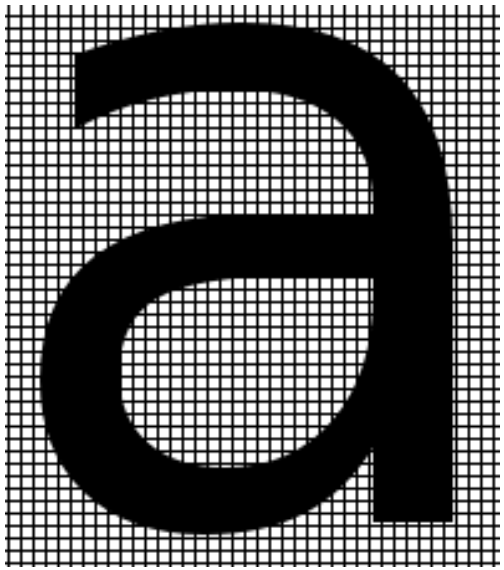
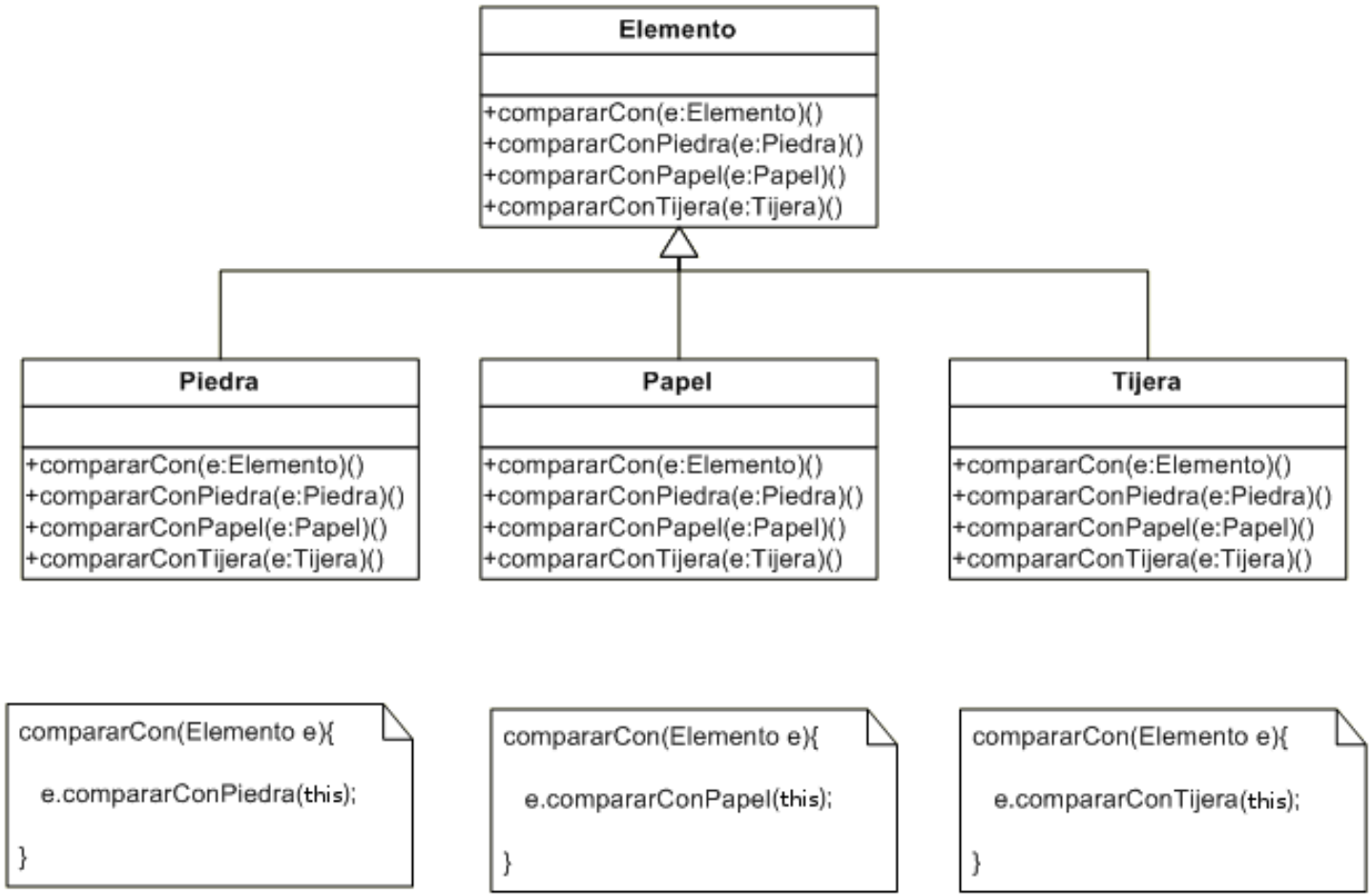


Figura: Mapa de bits monocromo asociado al caracter *a*

Colisiones: Técnica de *Double Dispatch*

Evita tener que identificar el tipo de dos objetos derivados de la misma clase base.



Escenarios: Mapas de *tiles*

Tile: una imagen cuadrada, rectangular o hexagonal, utilizada para generar imágenes de mayor complejidad.

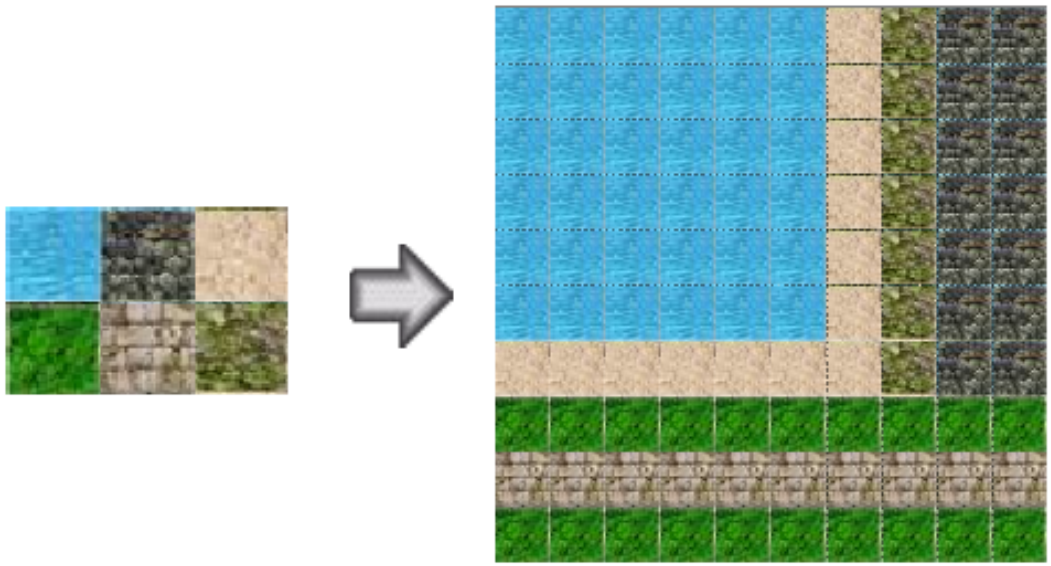


Figura: *Tileset* (conjunto de *tiles*) y mapa de *tiles*

Herramientas utilizadas

Tecnología

- C++, con GNU Make.
- XML

Desarrollo

- **Gimp**: edición de imágenes.
- **SoX**: edición de sonido.
- **Doxygen**: documentación automática de código.
- **LaTeX**: sistema de composición de textos.
- **Gantt Project**: diagramas de planificación.
- **Dia**: editor de diagramas UML.
- **Cppcheck**: analizador estático de código fuente C++.
- **Subversion**: control de revisiones.
- **Tiled**: editor de mapas de *tiles*.

Herramientas utilizadas

DevKitPPC

- Generación de ejecutables para sistemas *Power PC*.
- Reglas de compilación específicas para Wii.
- Fácilmente ampliable con bibliotecas externas.
- **Wiiload**: lanzar ejecutables mediante red local.

Bibliotecas externas

Previamente adaptadas para trabajar con *DevKitPPC*.

- **Libogc**: permite acceder al hardware de Nintendo Wii.
- **Libfat**: para trabajar con particiones FAT/FAT32
- **FreeType2**: necesaria para utilizar fuentes de texto.
- **TinyXml**: proporciona una interfaz para trabajar con XML.

Pruebas y validación

Especificación de las pruebas

- Garantizar el buen funcionamiento de la herramienta.
- Se han realizado durante y después del desarrollo.
- Han consistido en varios grupos de pruebas:
  - Pruebas de módulo.
  - Pruebas de sistema.
  - Pruebas de *makefile*.
  - Pruebas de juegos.
  - Análisis estático del código con *Cppcheck*.
- Todas las pruebas se han realizado sobre la videoconsola.

Pruebas de módulo

Prueba exhaustiva de cada módulo, tras finalizar su desarrollo:

- Caja blanca: comprobar los distintos caminos que toma el flujo de ejecución en el módulo.
- Caja negra: partiendo de conjuntos de datos de entrada y comprobando la salida que producen.

Pruebas y validación

Pruebas de módulo

Prueba exhaustiva de cada módulo, tras finalizar su desarrollo:

- Caja blanca: comprobar los distintos caminos que toma el flujo de ejecución en el módulo.
- Caja negra: partiendo de conjuntos de datos de entrada y comprobando la salida que producen.

Pruebas de sistema

Comprobar funcionamiento de cada módulo junto con los demás.

- Animación e Imagen.
- Galería y recursos multimedia.
- Actor y figuras de colisión.
- Etc.

Pruebas y validación

Pruebas de *makefile*

- Compilación.
- Generación de documentación.
- Empaquetado.
- Instalación y desinstalación.

Pruebas de juegos

- *Playtesting*: varias personas juegan y reportan errores.

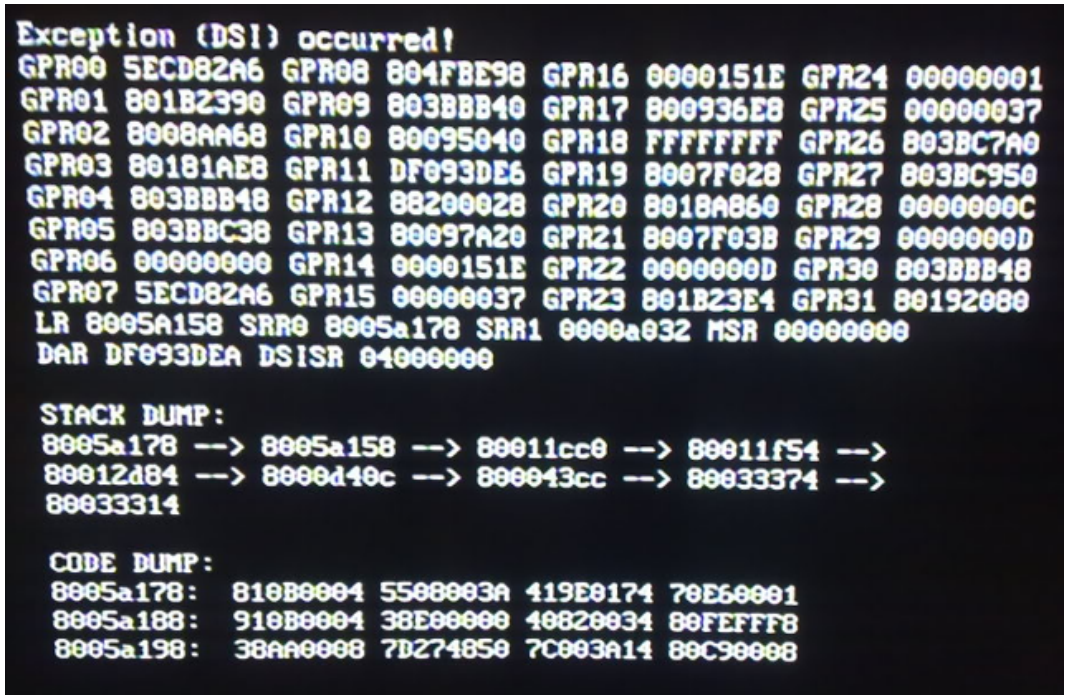
Análisis estático de código fuente

- Se utilizó Cppcheck.
- `> cppcheck --enable=all include src`

Pruebas y validación

Problemas surgidos durante el desarrollo

- Poca documentación existente: tutoriales muy incompletos.
- Control de la eficiencia: hardware limitado.
- Trabajo a bajo nivel con recursos: imágenes y fuentes.
- Difícil depuración.



Índice

Conclusiones

Cumplimiento de objetivos

- Se da acceso a los subsistemas de Nintendo Wii.
- Se cubren los puntos básicos de videojuegos 2D:
  - Gestión de recursos multimedia.
  - Animaciones.
  - Detección de colisiones.
  - Soporte de internacionalización.
  - Diseño de escenarios.
  - Registro de mensajes del sistema.
- Documentación completa, útil y en español.
- Tres juegos de ejemplo.
  - Mecánicas diferentes.
  - Código fuente totalmente comentado.

Conclusiones

Objetivos personales

- Desarrollo de biblioteca partiendo de una base pequeña.
- Aprendizaje de varias herramientas libres:
  - Profundización en GNU Make.
  - Bibliotecas: Libfat, FreeType2 y TinyXML.
  - Doxygen.
  - LaTeX.
  - Subversion.
- Adquisición de conocimientos sobre Nintendo Wii:
  - Trabajo con formatos multimedia.
  - Control de los mandos de la videoconsola.
  - Comprender cómo funciona Wii internamente.
- Puesta en práctica de los conocimientos adquiridos.
- Contribución al mundo del Software Libre y el *Homebrew*.

Conclusiones

Posibles mejoras

- Sistema de sonido 3D.
- Soporte para otros periféricos de Nintendo Wii.
- Puertos USB traseros.

Futuro del proyecto

- Desarrollo de juegos más complejos.
- Creación de comunidad de desarrolladores.

Índice

Bibliografía recomendada


Página oficial del proyecto DevKitPro.
http://www.devkitpro.org


Wiki sobre *homebrew* para Nintendo Wii.
http://www.wiibrew.org


Tutorial de programación para Nintendo Wii (Hermes).
http://www.elotrolado.net/wiki/Curso_de_programacion


Tutoriales de programación para Nintendo Wii de Scene Beta.
http://wii.scenebeta.com/tutoriales/wii


Aburruzaga García, Medina Bulo, Palomo Lozano
Fundamentos de C++. Servicio de Publicaciones UCA, 2001.
ISBN: 84-7786-734-8.

Demostración de los juegos de ejemplo

Arkanoid Wii

Wii Pang

Duck Hunt Wii

Esto es todo...

Gracias por su atención

¿Preguntas?

http://libwiiesp.forja.rediris.es/