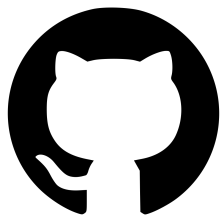


Motor de juegos 3D/2D en C++ y OpenGL

**Realizado por Javier Aguilera Puerta
Tutorizado por Francisco de Asís Rus Mansilla**

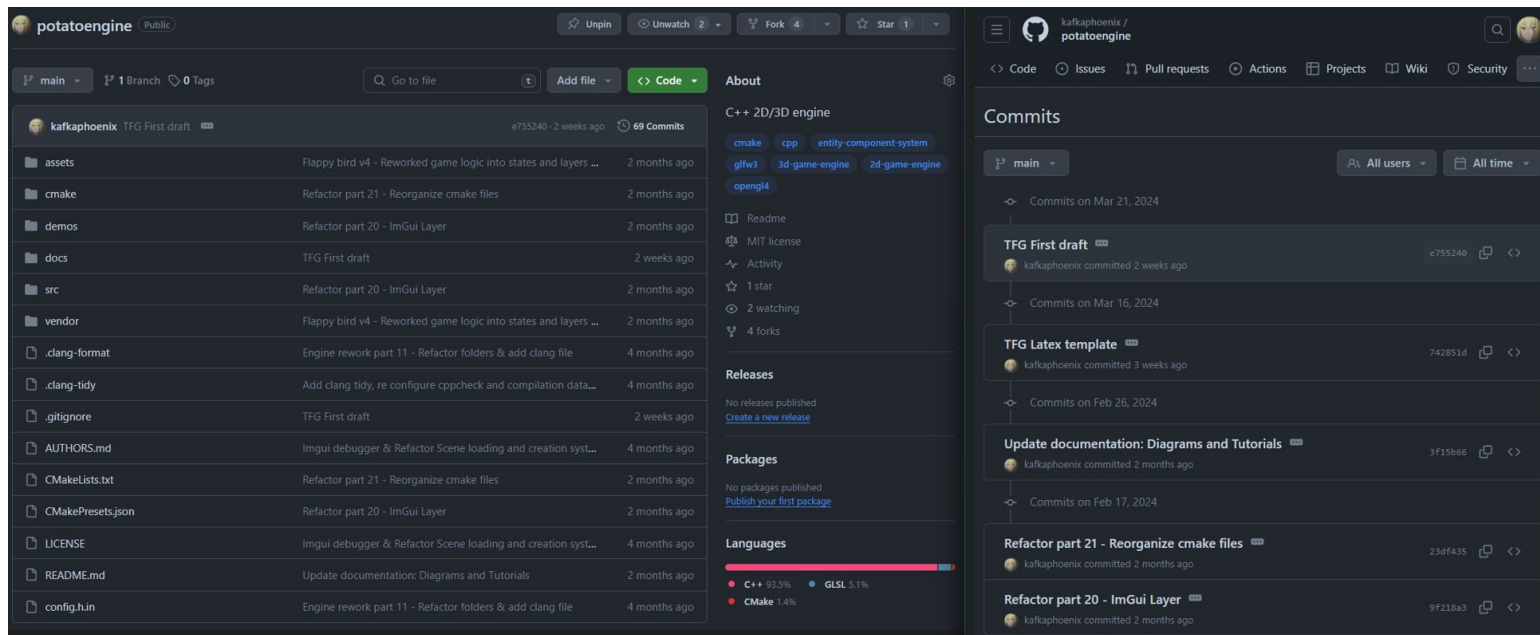
Organización del proyecto



Organización del proyecto



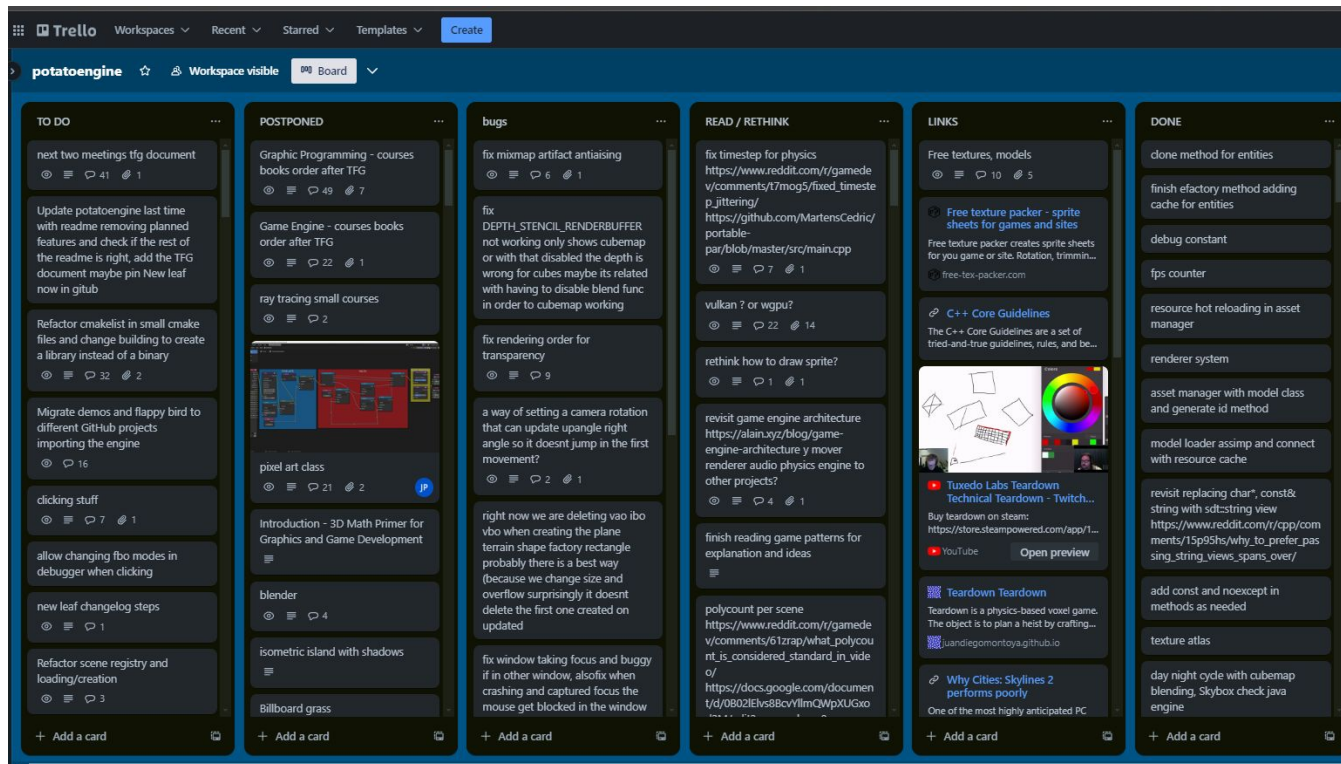
Scan me!



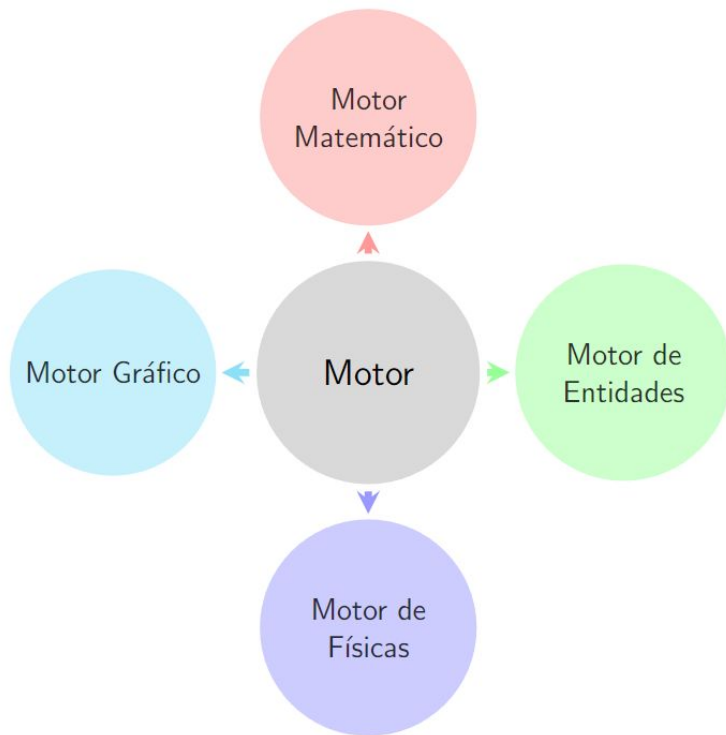
The screenshot displays the GitHub repository for 'potatoengine' by user 'kaphoenix'. The repository is public and has 4 forks and 1 star. The main branch is 'main'. The repository contains a file tree with folders like 'assets', 'cmake', 'demos', 'docs', 'src', and 'vendor', and files like '.clang-format', '.clang-tidy', '.gitignore', 'AUTHORS.md', 'CMakeLists.txt', 'CMakePresets.json', 'LICENSE', 'README.md', and 'config.h.in'. The commit history shows recent commits by 'kaphoenix' with titles like 'Flappy bird v4 - Reworked game logic into states and layers ...', 'Refactor part 21 - Reorganize cmake files', 'Refactor part 20 - ImGui Layer', 'TFG First draft', 'Engine rework part 11 - Refactor folders & add clang file', 'Add clang tidy, re configure cppcheck and compilation data...', 'ImGui debugger & Refactor Scene loading and creation syst...', 'Update documentation: Diagrams and Tutorials', and 'Engine rework part 11 - Refactor folders & add clang file'. The right sidebar shows the 'Commits' section with a list of commits, including 'TFG First draft', 'TFG Latex template', 'Update documentation: Diagrams and Tutorials', 'Refactor part 21 - Reorganize cmake files', and 'Refactor part 20 - ImGui Layer'.

<https://github.com/kaphoenix/potatoengine>

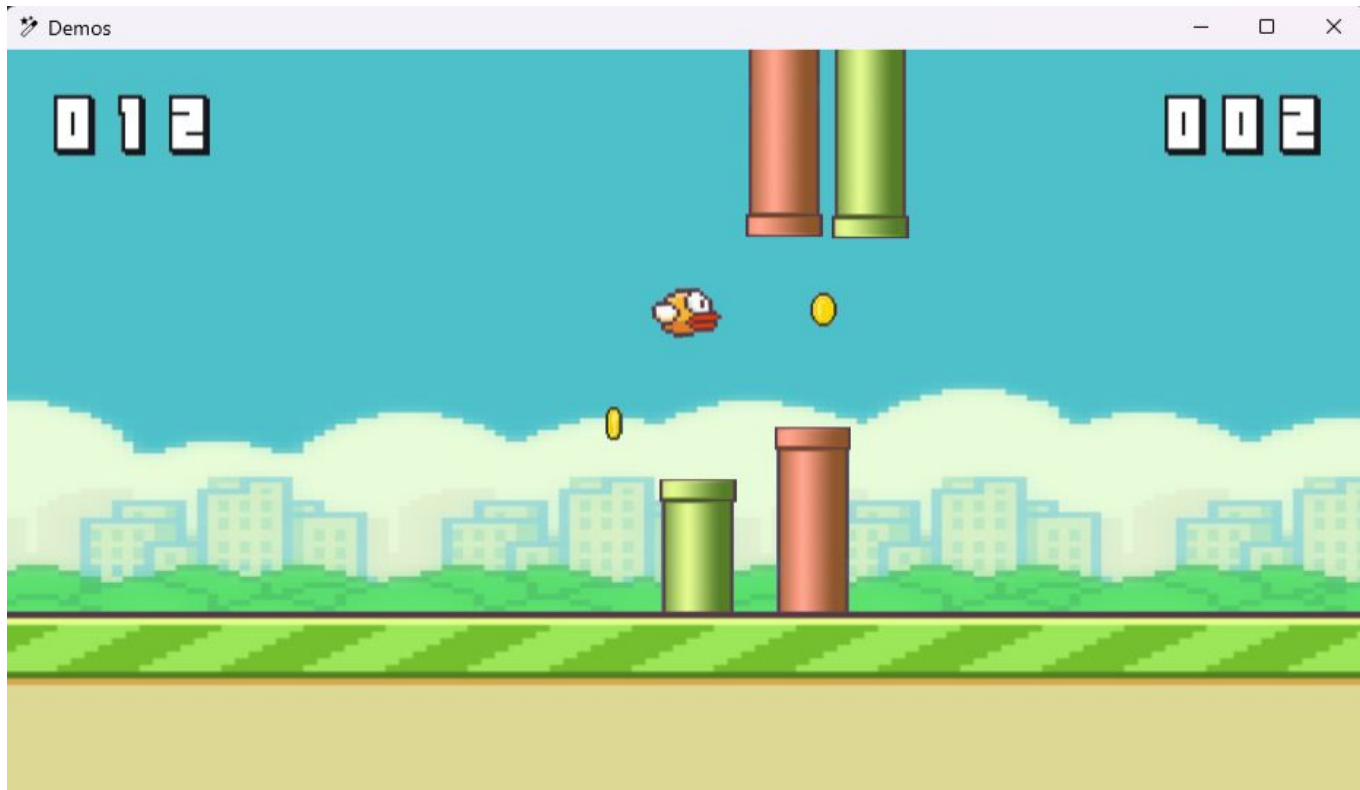
Organización del proyecto



¿Qué es un motor de juegos?



¿Qué es un motor de juegos?



¿Por qué hacer uno?

“Scientia potentia est” - Thomas Hobbes

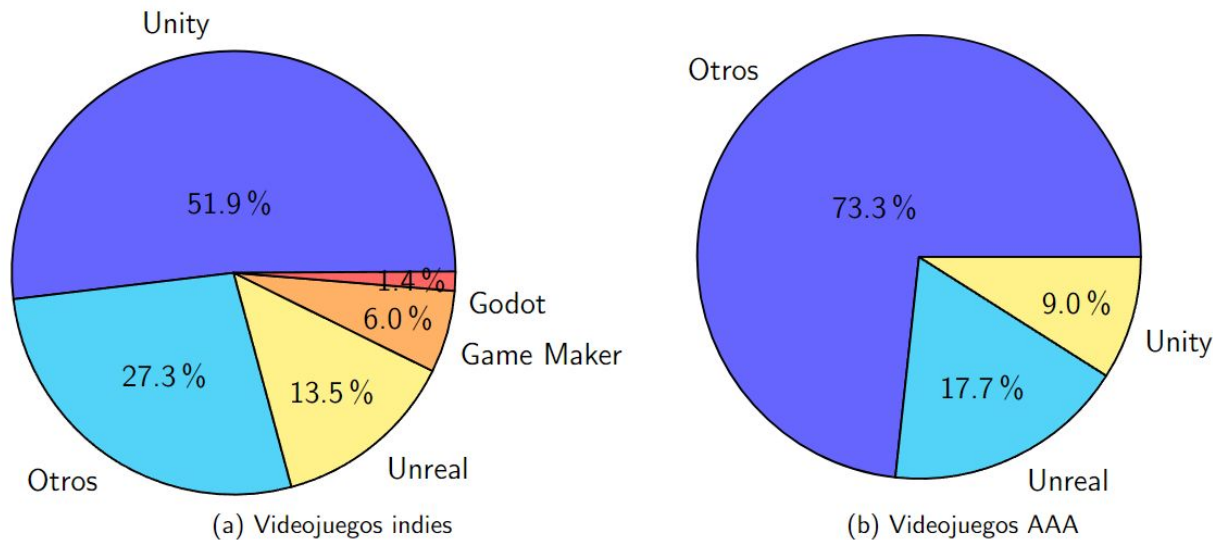
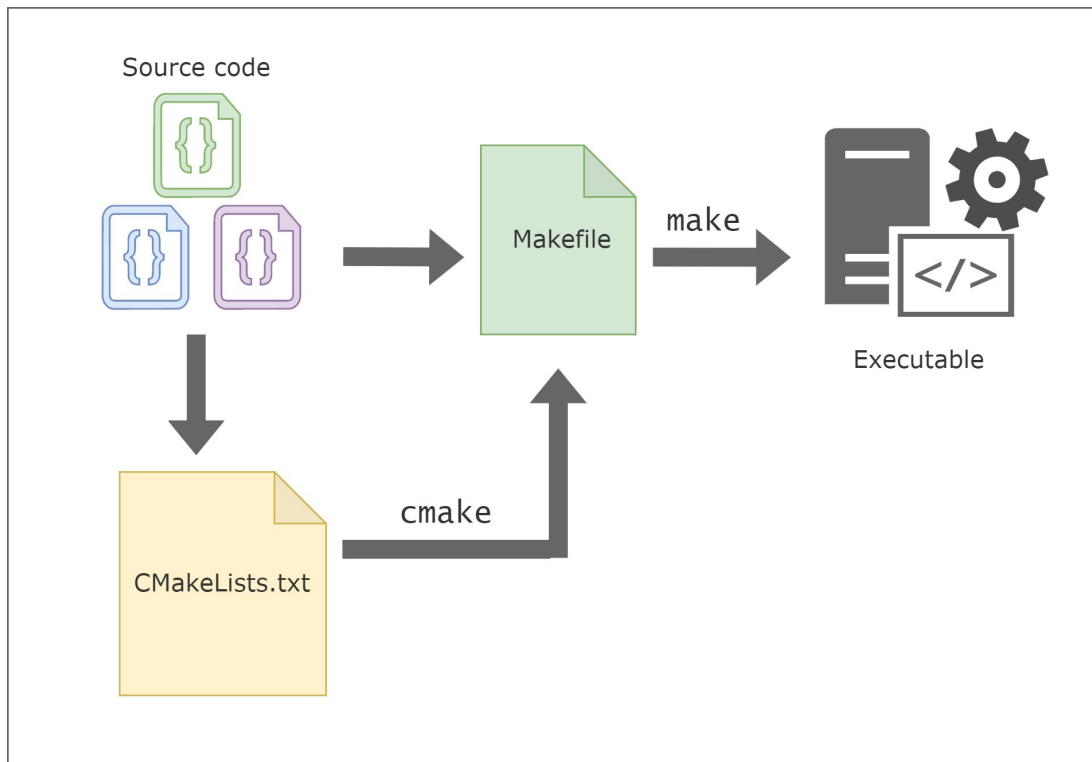


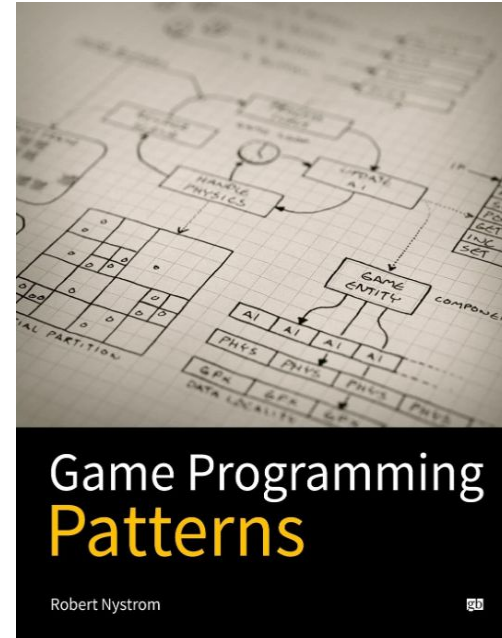
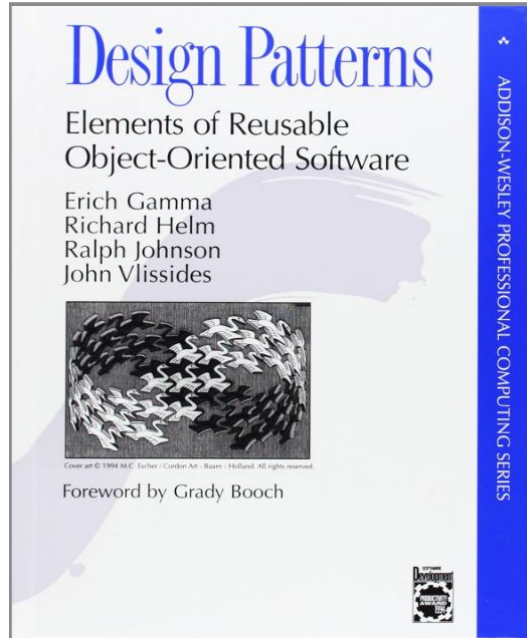
Figura 1: Distribución de motores usados para videojuegos (septiembre de 2023)

¹ Gamalytic.com, Motores de juegos más populares en la industria de los videojuegos, <https://gamalytic.com/blog/exploring-the-pc-engine-landscape>

Arquitectura del motor - Compilación



Arquitectura del motor - Patrones de diseño

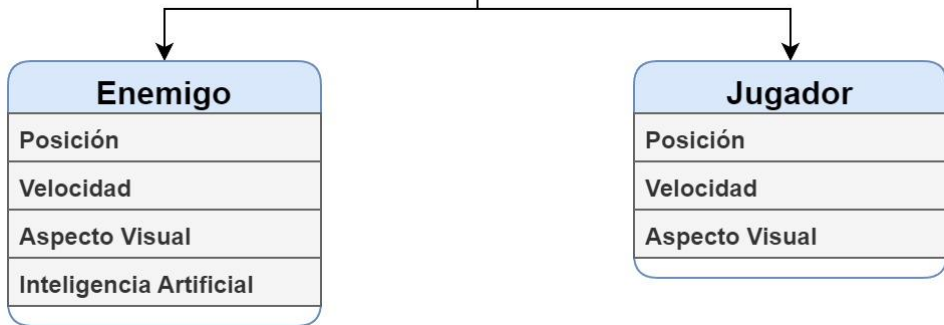


Arquitectura del motor - ECS vs OO

Orientada a
objetos (OO).

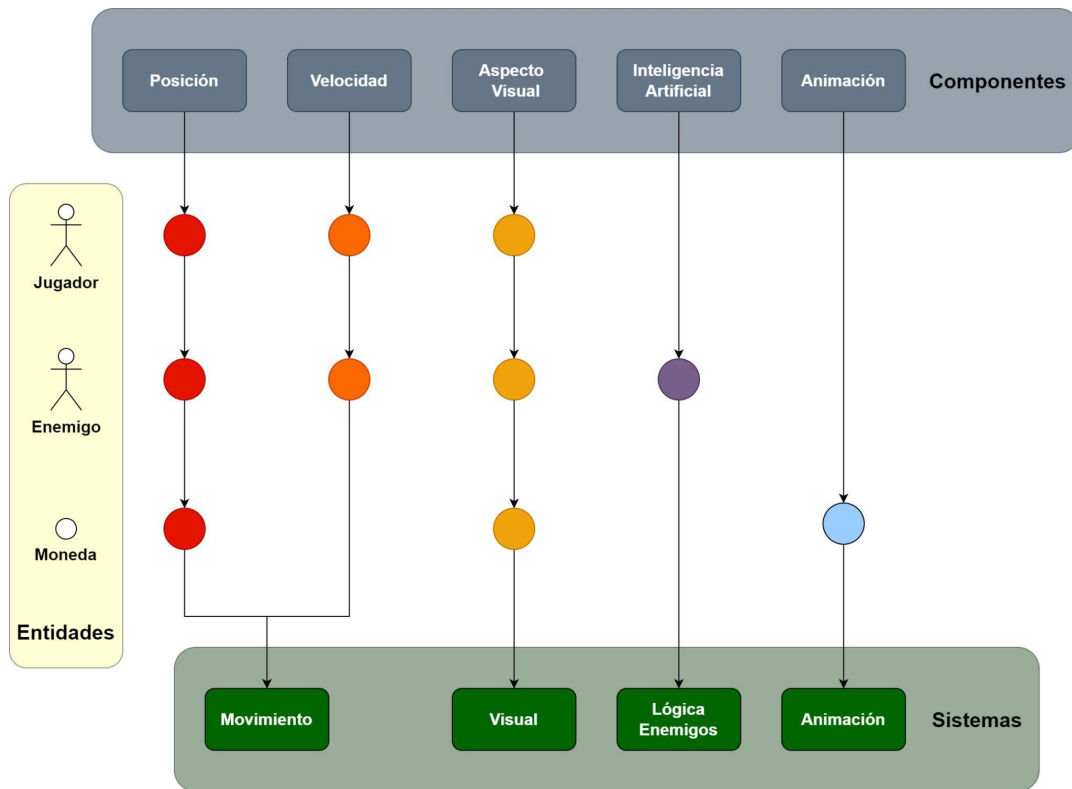
Moneda
Posición
Aspecto Visual
Animación

Entidad
Posición
Velocidad
Aspecto Visual



Arquitectura del motor - ECS vs OO

Entidad Componente
Sistema(ECS).



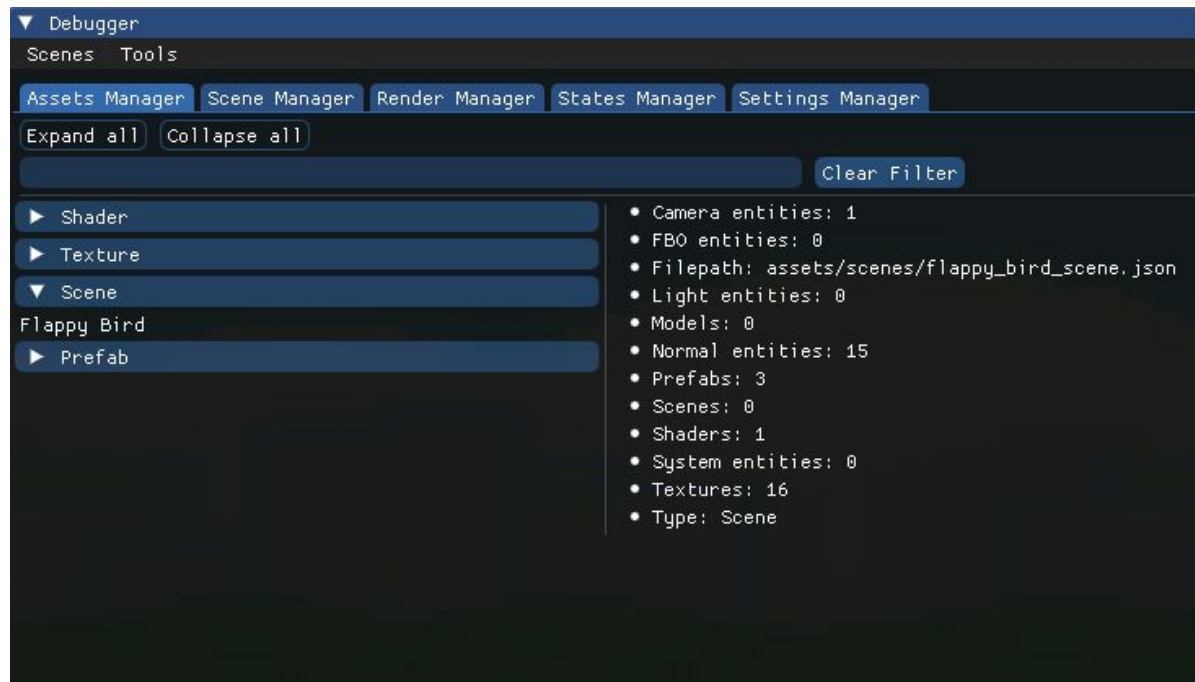
Arquitectura del motor - Otros componentes del motor

¿Qué otros componentes básicos necesita un motor?

- Configuración: resolución, idioma, etc.
- Assets / Recursos: imágenes, modelos 3D, sonidos, etc.
- Logger: traza de lo que ocurre en la aplicación.
- Máquina de estados: diferentes niveles del juego.
- Eventos: eventos de teclado, ratón, lógica, etc.

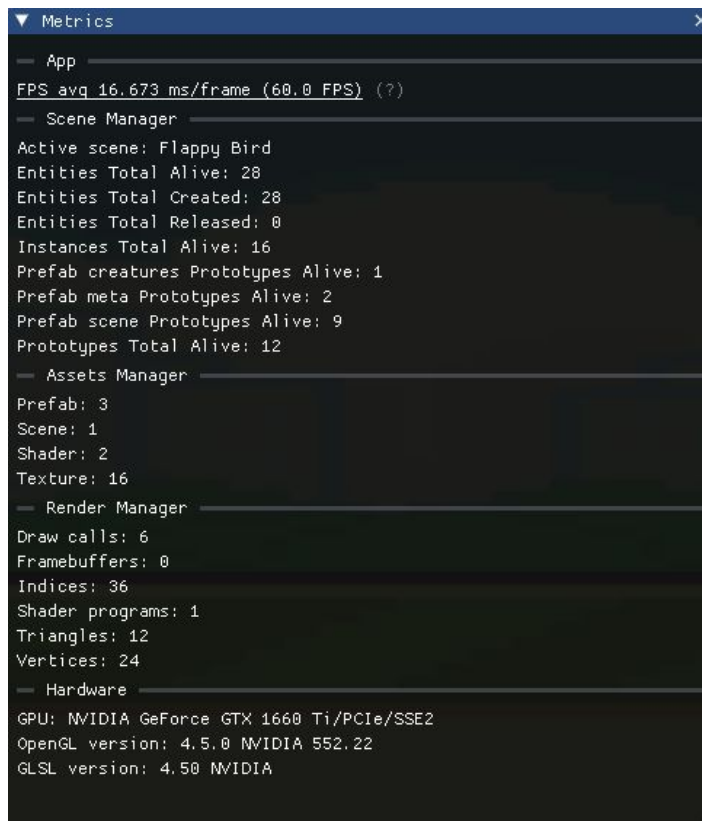
Herramienta debugger - Inspector

Permite analizar los diferentes componentes del motor.



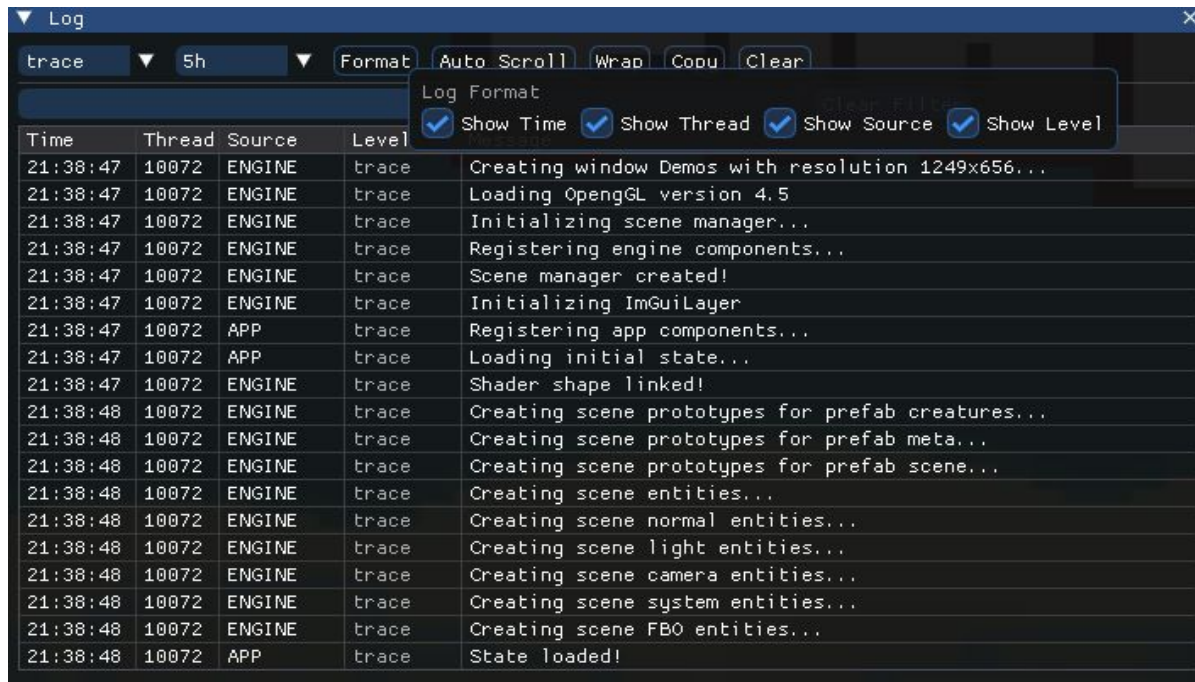
Herramienta debugger - Métricas

Permite visualizar el rendimiento de la aplicación.



Herramienta debugger - Logger

Permite analizar las trazas de la aplicación.



Arquitectura del motor - Managers API

Abstraen la lógica interna del motor.

- Logger Manager: Configuración de logs.
- Settings Manager: Configuración general.
- States Manager: Estado de la aplicación.
- Windows Manager: Ventana de la aplicación y eventos.
- Render Manager: Dibujo o renderizado de la escena usando el motor gráfico.
- Scene Manager: Controla el motor de entidades y la escena.
- Assets Manager: Toda operación relacionada con assets o recursos.

Caso de uso - Diseño de Flappy bird

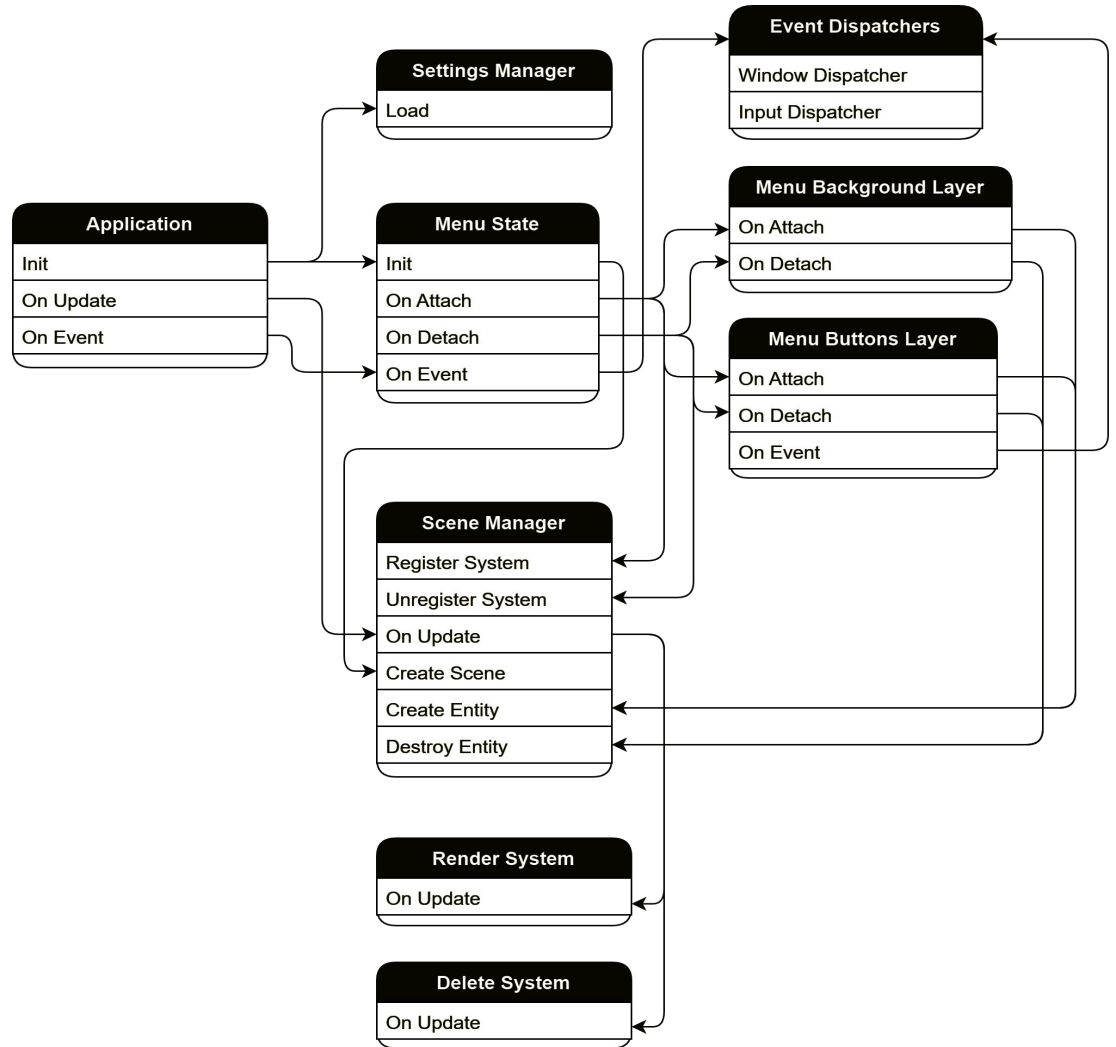


Variación de un juego de móvil que se hizo famoso añadiendo niveles aleatorios.

FLAPPY_BIRD

- assets
 - models
 - prefabs
 - scenes
 - shaders
 - textures
- build
- engine
- src
 - components
 - dispatchers
- layers
 - game
 - menu
- states
 - gameState.cpp
 - gameState.h
 - menuState.cpp
 - menuState.h
- systems
- app.cpp
- utils.h
- vendor
- CMakeLists.txt

Caso de uso - estado Menú



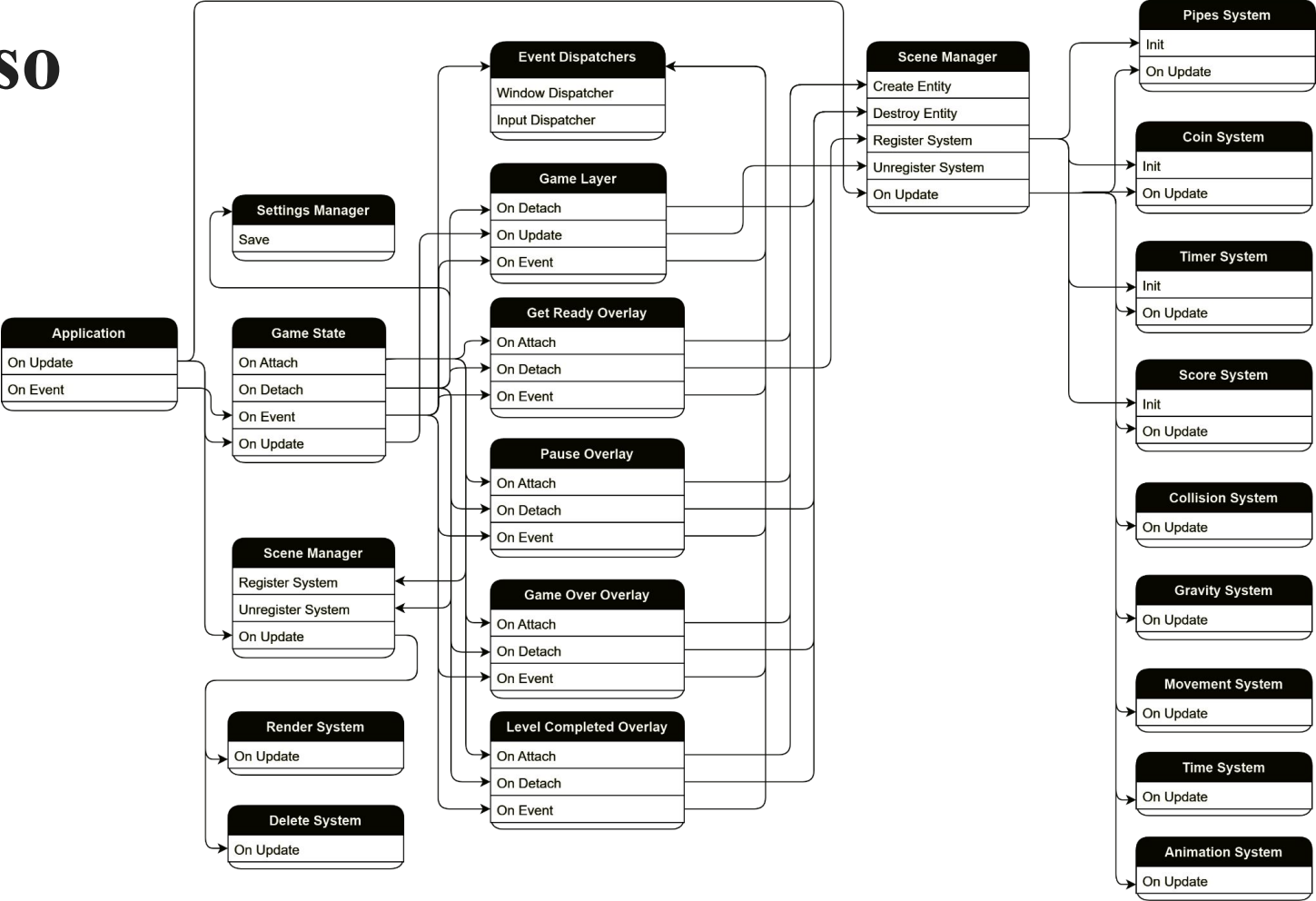
Caso de uso - Estado Menú



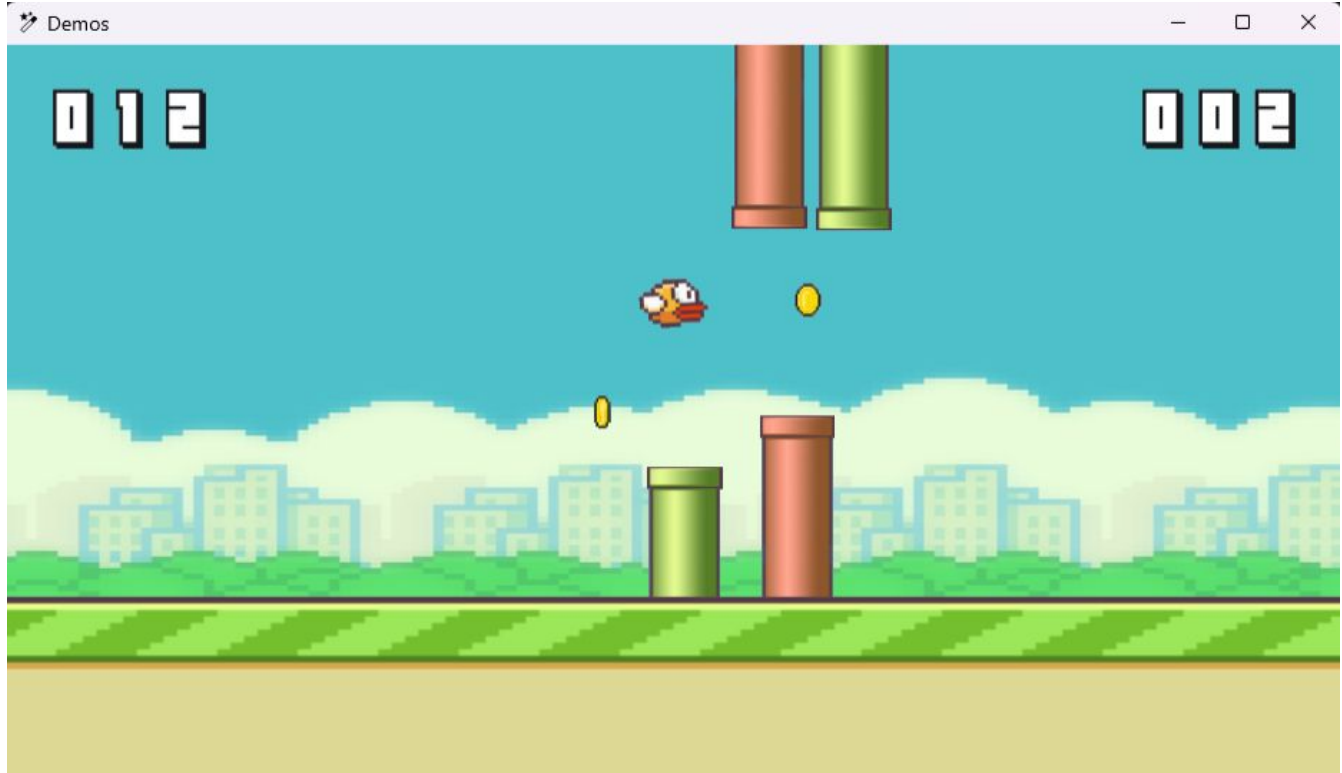
Caso de uso

- Estado

Juego



Caso de uso - Estado Juego



Caso de uso - Flappy bird demostración



Caso de uso - escenas en 3D demostración



Documentación y Tutorial

El proyecto se completa con una serie de vídeos tutoriales cubriendo todas las partes del motor, así como mostrando la creación del juego 2D, todo esto acompañado por diagramas. Pueden encontrarse en el repositorio de Github.

Conclusiones

- Aprendizaje personal.
- Herramienta totalmente funcional contribuyendo al código abierto ofreciendo un motor preparado para crear juegos.
- Objetivo didáctico: 12 vídeos tutoriales del motor y 5 vídeos creando el juego 2D Flappy bird.
- Uso real posible: demostraciones tanto en juegos 2D como en 3D de su utilidad para la comunidad online.

Posibles mejoras

- Importar el proyecto como librería auto-instalable.
- Soporte para más APIs gráficas: Vulkan, Metal y DirectX.
- Curso formación juegos 3D.

Preguntas

