

Jorge Bello

✉ jbopp.dev@gmail.com | 📞 (+34) 669-842-188 | 💻 jorge-bello-martin | 🐦 @jbopp | 📺 jorgbell | 🏠 jorgbell.github.io/portfolio | 🍷 itch

Projects

Game User Research Tool for Unity - C - ⚙️

September 2022 - Present

- Tool currently under development for Unity that facilitates the implementation of user tests (questionnaires and/or telemetry) in projects.
- Based on extensive research in the form of a final degree project (TFG) about the execution and current state of user tests in the gaming industry.

Quack Engine - C++ - ⚙️ - 🖥️

January - May 2021

- Participation in the design and development of an engine with a data-oriented structure following the Entity-Component-System design pattern with factories, with support for creating data files (levels, options, etc.) in Lua.
- Libraries used: Ogre3D (rendering), CEGUI (UI), FMOD (sound), Bullet (physics), and LuaBridge (Lua scripting support for C++).
- Demo game: The Hallway. Participation in the design and implementation of mechanics (flashlight, random maze generation) and integration of assets from Blender3D to Ogre3D.

Wizara - C - 🎮 - 🖥️

January - May 2019

- Efficient room and entity management system.
- Numerous movement and action mechanics for the player and enemies.
- Lead level designer and responsible for integration in Unity.

FacunDOOM - C - 🎮 - 🖥️

March 2023

- Retro-style FPS with sprite-stacking rendering system for enemies.
- Main involvement in designing and implementing the weapon management system, shaders, audio management, point streaks, and score-board.

Mortis Causa - C++ - 🎮 - 🖥️

January - May 2020

- Developed using the SDL2 API with an abstraction system to separate it from the game logic, following an ECS pattern.
- Main involvement in audio systems and asset integration, design and implementation of main mechanics (mini-games and investigation board), as well as creating sprites and composing the soundtrack.

Nonograma - 🎮 - 🎮 - 🖥️

September - December 2022

- Implemented a cross-platform abstraction (PC and Android) to separate API calls from the logic using Android Studio modules.
- Enhanced Android version with more features compared to the PC version, such as predefined level selection (not just random) and visual palette progression.
- Primary designer of classes to handle different engine sections on PC and Android: app lifecycle loop, rendering and screen drawing functionalities, input handling, and sound.

Duck Hunt Clone - C++ - 🎮 - 🖥️

September - December 2022

- Implemented a cross-platform abstraction (PC and PS4) to separate API calls from the logic using separate Visual Studio projects for each platform.
- Abstraction of the PC platform using SDL2, with independence from sprite rendering functionalities, core SDL features, and input (keyboard, mouse, and gamepad).
- Game implemented using synchronized threads for rendering and logic.

Skills

Programming Languages C++, C#, C, Python, Java, Javascript, Lua, GLSL

Libraries and Software Unity, SDL2, Nvidia Physx, Vuforia AR, SteamVR, OpenGL, Ogre3D, Unreal Engine 4, Blender 3D

Platforms Windows, Native PS4 (C++), Android

Also experience working with Scrum, Github, Agile

Languages Spanish (native), English (B2)

Education

Game Development Degree - Universidad Complutense de Madrid

Madrid, Spain
Sep. 2018 - Present

- Expected completion in September 2024, currently working on final degree project (TFG)

Honors & Awards

2019 **Ganador**, Aurigae Cashware Gamejam

Madrid

2023 **4th place and honorary mention**, ComJamOn 2023

Madrid

Jorge Bello

✉ jbopp.dev@gmail.com | 📞 (+34) 669-842-188 | 💻 jorge-bello-martin | 🐦 @jbopp | 📺 jorgbell | 🏠 jorgbell.github.io/portfolio | 🎮 itch

Proyectos

Game User Research Tool for Unity - C# 🚀 - ⚙️

Septiembre 2022 - Actualidad

- Herramienta para Unity actualmente en desarrollo que permite facilitar la implementación de pruebas de usuario (cuestionarios y/o telemetría) en los proyectos.
- Apoiado en una amplia investigación en forma de TFG acerca de la realización y estado actual de las pruebas de usuario en la industria del videojuego.

Quack Engine - C++ - ⚙️ - 🖥️

Enero - Mayo 2021

- Participación en el diseño y desarrollo de un motor con una estructura a datos siguiendo un patrón de diseño Entity-Component-System con factorías, con soporte para realizar los archivos de datos (niveles, opciones, etc) en Lua.
- Librerías utilizadas: Ogre3D (renderizado), CEGUI (UI), FMOD (sonido), Bullet (físicas) y LuaBridge (soporte de scripting en Lua para C++).
- Videojuego de demostración: The Hallway. Participación en el diseño de mecánicas e implementación de las mismas (linterna, generación aleatoria del laberinto) y en la integración de assets desde Blender3D a Ogre3D.

Wizara - C# 🚀 - 🎮 - 🖥️

Enero - Mayo 2019

- Sistema eficiente de gestión de salas y entidades.
- Multitud de mecánicas de movimiento y acción para el jugador y enemigos.
- Diseñador principal de niveles y encargado de su integración en Unity.

FacunDOOM - C# 🚀 - 🎮 - 🖥️

Marzo 2023

- FPS con estilo retro y sistema de renderizado de sprite-stacking para los enemigos.
- Participación principal en el diseño e implementación del sistema de gestión de armas, shaders, gestión de audio, rachas de puntos y score-board.

Mortis Causa - C++ - 🎮 - 🖥️

Enero - Mayo 2020

- Desarrollado utilizando la API de SDL2 con un sistema de abstracción para separarla de la lógica del videojuego, siguiendo un patrón ECS.
- Participación principal en los sistemas de audio e integración de assets, diseño e implementación de mecánicas principales (minijuegos y tablero de investigación), además de la creación de sprites y composición musical de la banda sonora.

Nonograma - 🎮 - 🎮 - 🖥️

Septiembre - Diciembre 2022

- Implementada una abstracción multiplataforma (PC y Android) para separar llamadas a la API de la lógica mediante módulos de Android Studio.
- Versión de Android mejorada con más funcionalidades que en PC: selección de niveles predefinidos (no sólo aleatorios), progresión con obtención de paletas visuales.
- Principal diseñador de las clases para gestionar los diferentes apartados del motor en PC y Android: bucle para el ciclo de vida de la app, funcionalidades de renderizado y dibujado en pantalla, recepción de entrada, sonido.

Duck Hunt Clone - C++ - 🎮 - 🖥️

Septiembre - Diciembre 2022

- Implementada una abstracción multiplataforma (PC y PS4) para separar llamadas a la API de la lógica utilizando un proyecto en Visual Studio para cada una de las plataformas.
- Abstracción de la plataforma PC utilizando SDL2, con independencia de las funcionalidades para renderizado de sprites, funcionalidades núcleo de SDL y entrada (teclado, ratón y gamepad).
- Videojuego implementado utilizando hebras de ejecución diferentes sincronizadas para renderizado y lógica.

Competencias

Programación	C++, C#, C, Python, Java, Javascript, Lua, GLSL
Librerías y Software	Unity, SDL2, Nvidia Physx, Vuforia AR, SteamVR, OpenGL, Ogre3D, Unreal Engine 4, Blender 3D
Plataformas	Windows, Native PS4 (C++), Android
Experiencia trabajando con	Scrum, Github, Agile
Idiomas	Español (nativo), Inglés (B2)

Educación

Grado en Desarrollo de Videojuegos - Universidad Complutense de Madrid

Madrid, España
Sep. 2018 - Actualidad

- Prevista finalización en septiembre de 2024, realizando actualmente TFG

Reconocimientos

- 2019 **Ganador**, Aurigae Cashware Gamejam
- 2023 **4º puesto y mención honorífica**, ComJamOn 2023

Madrid
Madrid

