SIMONE BISATO

Game developer

About Me

I'm a gameplay programmer with three years of specialized training at AIV– Italian Videogame Academy, where I honed my skills in crafting engaging and responsive player experiences. I specialize in gameplay mechanics, AI systems, and technical implementation using Unity and Unreal Engine, ensuring smooth and scalable solutions with a keen eye on performance and maintainability.

Padova, Italy

+39 3911787005

work.simonebisato@gmail.com

Portfolio: simonebisato.it

GitHub: github.com/Graaced

Education

2021 - 2024

Diploma in Game Programming
AIV - Italian Videogame Academy, Rome

Languages

Italian: Native

English: B2 (Technical reading, team communication)

Skills

- Handling of graphic assets
- Performance-aware development and maintainability strategies
- Cross-functional collaboration with art, design teams

Projects

Full portfolio available at : simonebisato.it

Nebula Ranger

A retro-style space shooter. Focused on simple, engaging gameplay while strengthening programming and teamwork skills.
Key Features: Asteroid Spawn System, Power-Up System, State Machine, UI, Audio, VFX

Skyfall

An endless survival game where the player avoids falling asteroids while driving a car on a small planet. Developed independently with a focus on time management and movement mechanics.
Key Features: Movement & Camera System, Missions, Audio, VFX

PourMeSomePuree

A top-down 2D game created with a custom engine built from scratch using AivFast2D.
Key Features: Custom Physics Engine Code, Animations, UI, Enemy and Player Mechanics

Technical skills

Programming Languages

- C# – Advanced
- C++ – Intermediate
- Python – Basic

Game Engines

Unity

- Development 3D/2D
- UI systems, physics simulation
- Scriptable Objects
- Asset Management

Unreal Engine

- Visual scripting with Blueprints: Used in academic settings to create interactive gameplay features, including environmental triggers, AI behaviors using Behavior Tree and Blackboard systems, and environmental interaction systems (levers, buttons, collectible objects) through Blueprint Interfaces.

- Native C++ scripting

Gameplay Development

- Programming core gameplay mechanics
- Custom tools & editor extensions
- AI development with Behavior Trees and Finite State Machines
- Gameplay systems integration
- UI/UX implementation
- Animation & audio integration
- Level design workflows

Tools & Workflow

- Version control: Git, GitHub, Fork
- Project/task management: Trello
- Debugging, optimization, rapid prototyping