

Bogdan Predescu

Software Engineer and Game Developer

Contact

Email: bogdan.predescu.ba@gmail.com

Phone: (+44) 7455 596 074

I am a long time programmer and in the past few years I've also been significantly involved in Game Design. I've worked with different platforms (PC, mobile, VR, AR, cross-platform applications, etc.), in recent years primarily focusing on VR experiences.

I've been involved in many different aspects of game development (Gameplay mechanics, AI, Multiplayer, Game Design, Project Architecture, Performance Optimization, etc.) and I have 10 years of experience with the Unity engine.

I've worked with different team sizes, from solo projects to teams of 50+, including experience in leadership roles. I can handle team structures that require being self-guided, and I can take ownership and personal responsibility of entire features or the core project architecture.

Skills

Advanced Level

C#
Unity Engine
VR and AR
Mobile Games (iOS, Android)
Gameplay Programming
Game Design
AI Systems and AI Design
Physics and Procedural Animations
Game Networking and Multiplayer
Particle Systems and VFX
UI and UX
Project Architecture

Intermediate Level

Unity DOTS
C/C++
SQL
Shaders
Compute Shaders
Nvidia CUDA
Animation Systems
Cross-Platform applications
Project Management

Project Experience

"VR Spellcasting Game" (Meta Quest, PC VR, others) – In Development

Role: Programmer and Gameplay Design

Languages and Tools: C# (programming language), Unity (engine)

Description: VR Action Game about casting spells through hand gestures, in a psychedelic environment where gameplay interactions and visual effects are synchronized to the beat of the music.

Tasks:

- Designed and implemented a Spell Casting system that uses hand gestures with the VR hand controllers
- Designed and implemented a Music System that dynamically controls music layers and sound effects based on gameplay events and the Player's progress.
- Designed and implemented gameplay mechanics (AI enemies, spell effects, interactive objects), many of which synchronize to the beat of the music
- Designed levels and combat encounters
- Designed and implemented a custom Particles System, using Unity's Multithreaded Job System, for effects synchronized to the beat of the music, and that can apply a sequence of different effects to the same group of particles.

"Untitled Strategy Game" (PC)**Role:** Programmer and Gameplay Design**Languages and Tools:** C# (programming language), Unity (engine)**Description:** Strategy Game Prototype, with a focus away from micromanagement and more on high level strategic choices.**Tasks:**

- Design and implemented Player input, Camera controls, Unit Selection and Unit Commands system
- Implemented a Hex Tilemap System and Editor Tool
- Implemented Pathfinding and Navigation System for Hex Tilemaps
- Designed and implemented combat mechanics for units with different attack types and target priorities
- Designed and implemented various construction entities, with in-game build menu system and upgrades tech tree.
- Designed and implemented gameplay mechanics for different types of units (infantry, vehicles, buildings)
- Implemented AI Player system and designed behaviors for a few different test levels.
- Designed and implemented various menus and in-game UI elements (build options and build orders queue, resources, unit stats, selection indicators, pathfind indicators, etc.)

"Chicken Chase" (Android, iOS)**Role:** Programmer**Languages and Tools:** C# (programming language), Unity (engine)**Description:** Turn-Based Puzzle Game for mobile platforms**Tasks:**

- Implemented a turn-based system using Unity's new multithreaded Job System, with the feature of separated visual and computational turn sequences, which maintains high runtime performance irrespective of how complex and computationally intensive the gameplay mechanics would get.
- Implemented gameplay mechanics
- Implemented system and behaviors for AI characters
- Implemented Player input and camera controls

- Implemented Menu and UI System
- Implemented In-Game Store using Unity IAP

“Jeff Wayne’s The War of the Worlds: The Immersive Experience” (HTC Vive)

Role: Programmer, Gameplay and UX Design

Languages and Tools: C# (programming language), Unity (engine)

Description: Location-Based Immersive Experience featuring immersive theater, multisensory effects, VR, motion simulators and holograms, based on Jeff Wayne’s The War of the Worlds musical.

Links: <https://thewaroftheworldsimmersive.com/>
<https://www.youtube.com/watch?v=KMVT0zWmuNk>

Tasks:

- Implemented VR player controls and other external inputs used by the immersive theater staff
- Level Design and various VFX
- Coordinated with many other disciplines (3D Art, Animation, Audio, Theater Design, Physical Props, IT, Networking, etc.) to implement the sequence of events and interactive elements of the experience. Scheduled and synchronized all elements using Unity’s Timeline system.
- Implemented VR interactive elements and mechanics, connected to motion tracked real-world props, rigs and motion simulators.
- Implemented various optimization techniques for some of the more performance intensive features.

“GPU Particle System” (PC, high-end mobile, VR, others)

Role: Programmer (Solo Project)

Languages and Tools: C# + HLSL (programming languages), Unity (engine)

Description: GPU Particle System using DirectX 11 Compute Shaders

Links: <https://youtu.be/mnoOyyZ2nxE>
<https://youtu.be/Ar1HrjEhbso>
<https://youtu.be/RuGtUxnOxRY>

Tasks:

- Implemented a custom Particle System that runs on the GPU using Compute Shaders. It can scale to a much higher number of particles (several millions on a high-end GPU) and supports more complex effects.

“Project Short Bridge” (Gear VR, PC VR)

Role: Game Designer and Programmer (Solo Project)

Languages and Tools: C# (programming language), Unity (engine)

Description: VR Puzzle Adventure about jumping between alternate dimensions in which the world has various specific differences.

Links: https://youtu.be/l_sAlutaL94
<https://youtu.be/03SJvycr2HY>

Tasks:

- Designed Puzzles and implemented various gameplay mechanics and interactive items
- Designed and implemented physics-based VR hand interactions
- Designed and implemented VR locomotion mechanics for navigation in a 3D physics-based world
- Implemented transition mechanic between parallel dimensions (different versions of a Level)
- Implemented various performance optimization techniques for low-end GearVR devices (Samsung S6), including a semi-custom lightmap system to be able to dynamically transition between different lighting versions for the same Level.

“Project Foundation” (PC)

Role: Gameplay Programmer and Game Designer (team of 6)

Languages and Tools: C# (programming language), Unity (engine), SQL

Description: Top-Down Multiplayer Shooter with a Sci-Fi/Cyberpunk setting.

Links: <https://youtu.be/ILvNblesJJ4>

Tasks:

- Implemented Player Character Controller (movement, animations, physics, input and camera controls)
- Designed and implemented weapons, special abilities and physics based effects
- Designed and implemented AI system and behaviors for neutral AI characters
- Implemented game attributes database and basic Lobby system for multiplayer matches
- Implemented in-game networking functionality including synced player movement, server-authoritative AI characters, weapons, abilities and other gameplay mechanics.
- Involved in all design aspects of the project, with a lead role in the Core Game Concept and various Gameplay mechanics
- Handled a significant part of the project management (not my main role)

Undergraduate Project: “Evolving Enemy AI System in Games” (PC)

Role: Designer and Programmer (solo project)

Languages and Tools: C# (programming language), Unity (engine)

Description: Research paper and demo application of an AI System for NPCs in an open world game environment.

The project combines traditional approaches like Finite State Machines with more advanced techniques like Dynamic Scripting and Affective Behaviour Modeling, to generate gameplay diversity.

Tasks:

- Implemented AI character animations using Unity’s Mecanim system.
- Implemented AI character navigation using Unity’s NavMesh system.
- Designed and implemented RPG combat mechanics and character evolution
- Designed and implemented static AI behaviors for passive open world activities (using HFSMs)
- Designed and implemented AI system that dynamically generates combat behaviors, for characters with evolving stats and attributes.

“Kraken Attack” (Android, iOS)

Role: Project Manager and Gameplay Programmer (team of 3)

Languages and Tools: Lua (programming language), Corona SDK (engine)

Description: 2D Multi-Touch Physics Arcade game in which you control a giant octopus that destroys and eats waves of enemies. Our team won First Place with this project in a university competition to design and develop a mobile game.

Tasks:

- Planned the development and delegated tasks
- Designed and implemented the basic game architecture
- Designed and implemented the Player Character Controller and physics interactions
- Implemented the AI system and basic behaviors
- Implemented Multi-Touch controls and gameplay events

OTHER PROJECTS:

- VR Multiplayer Prototype from Unity’s “Tanks!” Demo
- Quest System Node Editor Tool for RPG/Adventure Games, using Unity’s new Editor UI System
- “METALIFE” - Narrative Puzzle Game for mobile, which simulates a realistic phone
- “Bro Bots” - VR Experience - <https://www.breakingfourth.com/bro-bots>
<https://www.oculus.com/experiences/rift/1379869002120921>
- “Utopia 6” - VR Experience - <https://www.breakingfourth.com/utopia6>
- “Ctrl” - VR Experience - <https://www.breakingfourth.com/ctrl>
<https://www.youtube.com/watch?v=5-qTqUofdHA>
- “Castle Tilt” - Tower Siege game for Mobile. Our team won second place in a multi-university Game Jam.
- Small Augmented Reality application (Vuforia, Unity)
- Student Virtual Campus using PRINCE2 project management methodology (Unity)
- “D-Day” - Top-Down Shooter Technical Demo for a basic custom C++ engine and its integration with other APIs (C++, OpenGL, CUDA)
- Basic GPU Particles System using DirectX 11 Compute Shaders
- Interactive Digital Postcard (C++, OpenGL)
- Image Processing application (OpenGL)
- Space Invaders clone (C++, OpenGL)
- Image Effects using CUDA
- Turn-based Strategy Flash Game with Facebook integration
- Small game using XNA
- A few small Flash Games
- Managing software for gym trainers (C#, Visual Studio, SQL)
- Small physics simulation (C#, DirectX 9)
- C# .Net Text Messenger

Education

Computer Games Development – BSc (Hons)

University of Bedfordshire (2011 – 2014)

Informatics

Faculty of Mathematics and Computer Science - University of Bucharest (2009 – 2011)

Diploma in Advanced C#, ASP.NET and Web Services

Training Center in Informatics Bucharest (2007 – 2009)