

Game Programming Gems 2

par Matthieu Brucher (<http://matthieu-brucher.developpez.com/>) (Blog)

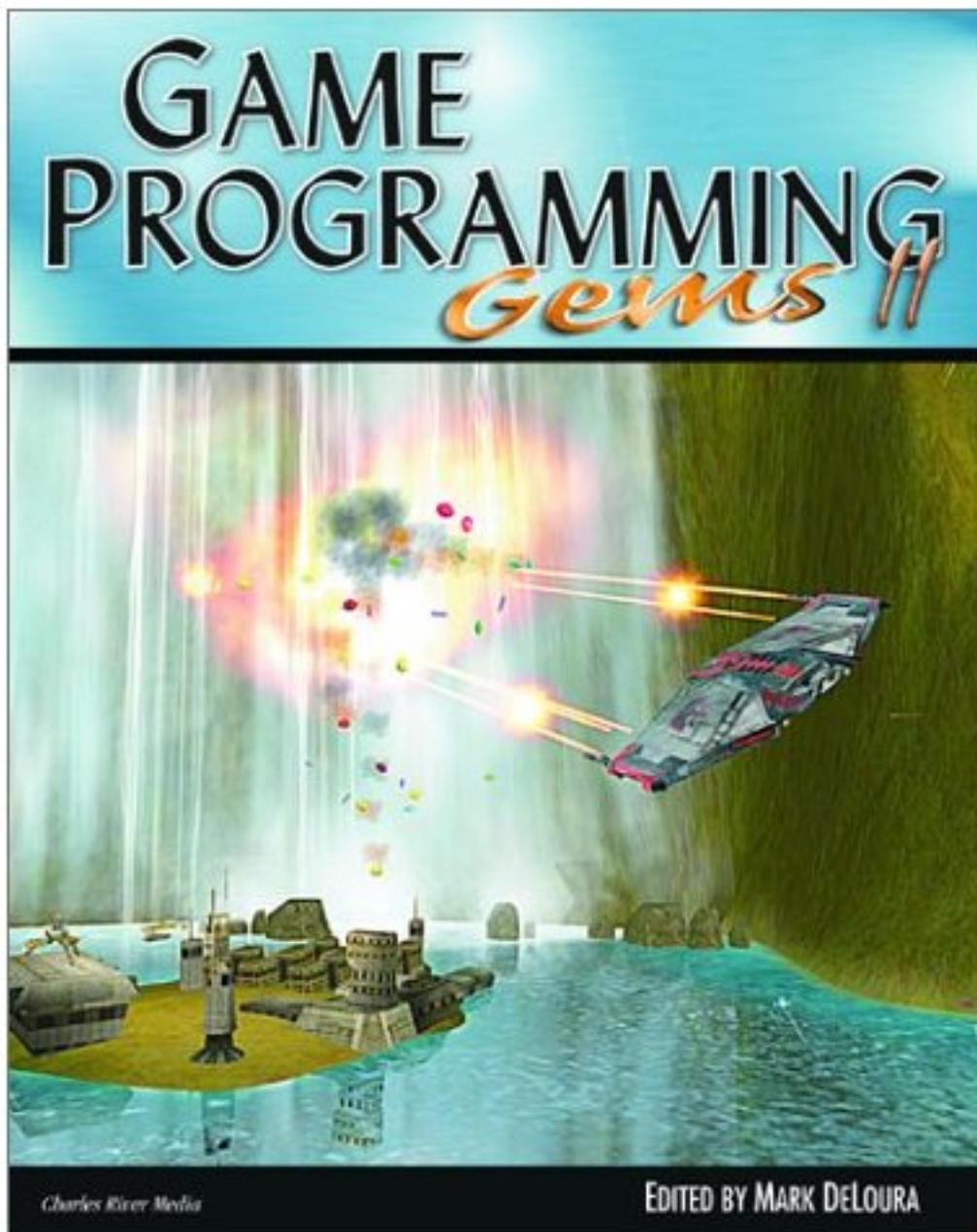
Date de publication : 18/07/2006

Dernière mise à jour : 27/09/2006

Critique de Game Programming Gems 2 édité par *Mark DeLoura*

- I - Description
- II - Table des matières
- III - Critique : Paradoxal
- IV - Liens annexes

I - Description



Game Programming Gems 2, le second volume dans cette série à succès, contient un ensemble nouveau de plus de 70 articles sur divers topics qui peuvent être utilisés dans vos propres projets de jeux. Ecrits par des experts de la programmation de jeux, chaque contribution propose une solution clé en main aux problèmes de programmation, ou une méthode originale pour réduire le temps de programmation et les redondances. Edité par *Mark DeLoura*, ancien éditeur en chef de **Game Developer Magazine**, et un groupe d'éditeurs spécialisés, cette collection couvre les topics majeurs utiles pour développer un moteur de jeu au top. Six sections complètes, incluant une nouvelle section sur la programmation audio, fournissent des techniques que les experts peuvent implémenter immédiatement, tandis que les programmeurs intermédiaires et débutants peuvent l'utiliser pour améliorer leurs talents et leur connaissance. Ceci est une référence à avoir qui contient la connaissance et l'expérience de plus de 40 développeurs de jeux.

II - Table des matières

- Section 1 - General Programming
 - 1.1 Optimizations for C++ games
 - 1.2 Inline Functions Versus Macros
 - 1.3 Programming with Abstract Interfaces
 - 1.4 Exporting C++ Classes from DLLs
 - 1.5 Protect yourself from DLL Hell and Missing OS Functions
 - 1.6 Dynamic Type Information
 - 1.7 A Property Class for Generic C++ Member Access
 - 1.8 A Game Entity Factory
 - 1.9 Adding Deprecation Facilities to C++
 - 1.10 A Drop-in Debug Memory Manager
 - 1.11 A Built-in Game Profiling Module
 - 1.12 Linear Programming Model for Windows-based Games
 - 1.13 Stack Winding
 - 1.14 Self-Modifying Code
 - 1.15 File Management using Resource Files
 - 1.16 Game Input Recording and Playback
 - 1.17 A Flexible Text Parsing System
 - 1.18 A Generic Tweaker
 - 1.19 Genuine Random Number Generation
 - 1.20 Using Bloom Filters to Improve Computational Performance
 - 1.21 3ds Max Skin exporter and animation toolkit
 - 1.22 Using Web Cameras in Video Games
- Section 2 - Mathematics
 - 2.1 Floating-Point Tricks: Improving Performance with IEEE Floating Point
 - 2.2 Vector and Plane Tricks
 - 2.3 Fast, Robust Intersection of 3D Line Segments
 - 2.4 Inverse Trajectory Determination
 - 2.5 The Parallel Transport Frame
 - 2.6 Smooth C2 Quaternion-based Flythrough Paths
 - 2.7 Recursive Dimensional Clustering: A Fast Algorithm for Collision Detection
 - 2.8 Programming Fractals
- Section 3 - Artificial Intelligence
 - 3.1 Strategies for Optimizing AI
 - 3.2 Micro-Threads for Game Object AI
 - 3.3 Managing AI with Micro-Threads
 - 3.4 An Architecture for RTS Command Queuing
 - 3.5 A High-Performance Tile-based Line-of-Sight and Search System
 - 3.6 Influence Mapping
 - 3.7 Strategic Assessment Techniques

- 3.8 Terrain Reasoning for 3D Action Games
- 3.9 Expanded Geometry for Points-of-Visibility Pathfinding
- 3.10 Optimizing Points-of-Visibility Pathfinding
- 3.11 Flocking with Teeth: Predators and Prey
- 3.12 A Generic Fuzzy State Machine in C++
- 3.13 Imploding Combinatorial Explosion in a Fuzzy System
- 3.14 Using a neural network in a game: A concrete example
- Section 4 - Geometry Management
 - 4.1 Comparison of VIPM Methods
 - 4.2 Simplified Terrain Using Interlocking Tiles
 - 4.3 Sphere Trees for Fast Visibility Culling, Ray Tracing, and Range Searching
 - 4.4 Compressed Axis-Aligned Bounding Box Trees
 - 4.5 Direct Access Quadtree Lookup
 - 4.6 Approximating Fish Tank Refractions
 - 4.7 Rendering Print Resolution Screenshots
 - 4.8 Applying Decals to Arbitrary Surfaces
 - 4.9 Rendering Distant Scenery with Skyboxes
 - 4.10 Self-Shadowing Characters
 - 4.11 Classic Super Mario 64 Third-Person Control and Animation
- Section 5 - Graphics Display
 - 5.1 Cartoon Rendering: Real-time Silhouette Edge Detection and Rendering
 - 5.2 Cartoon Rendering Using Texture Mapping and Programmable Vertex Shaders
 - 5.3 Dynamic Per-Pixel Lighting Techniques
 - 5.4 Generating Procedural Clouds Using 3D Hardware
 - 5.5 Texture Masking for Faster Lens Flare
 - 5.6 Practical Priority Buffer Shadows
 - 5.7 Impostors: Adding Clutter
 - 5.8 Operations for Hardware-Accelerated Procedural Texture Animation
- Section 6 - Audio Programming
 - 6.1 Game Audio Design Patterns
 - 6.2 A Technique to Instantaneously Reuse Voices in a Sampler-based Synthesizer
 - 6.3 Software-based DSP Effects
 - 6.4 Interactive Processing Pipeline for Digital Audio
 - 6.5 A Basic Music Sequencer for Games
 - 6.6 An Interactive Music Sequencer for Games
 - 6.7 A Low-Level Sound API

III - Critique : Paradoxal

Il y a du très bon dans ce livre, mais aussi du très moyen voir du fortement déconseillé.

Pour autant, je ne peux pas juger de la partie concernant le graphisme, la première concernant la programmation générale mélange les bons conseils - macros, fonctions inline - et les très mauvais - code auto-modifiant - quand certaines parties sont plutôt inutiles - les générateurs de nombres aléatoires -. La partie sur l'audio, nouvelle, commence par des évidences pour qui connaît les patterns, mais la suite est pertinente.

La partie mathématique commence quant à elle par une optimisation des calculs flottants. Actuellement, elle ne sert sans doute plus aussi utile, mais la suite est intéressante, par exemple calculer les paramètres pour une trajectoire bien précise. La partie sur l'intelligence artificielle est la suite de celle de GPG en partie - les machines à états -, parle de micro-threads implémentés en assembleur - à utiliser avec parcimonie, ce n'est pas portable -.

Dans l'ensemble, ce livre est intéressant, mais à prendre avec des pincettes.

IV - Liens annexes

 ***Critique sur la page de livres Jeux***

 ***Achat sur Amazon.fr***

 ***Lien vers le site de l'éditeur***

