A Framework for Designing 3D Virtual Environments

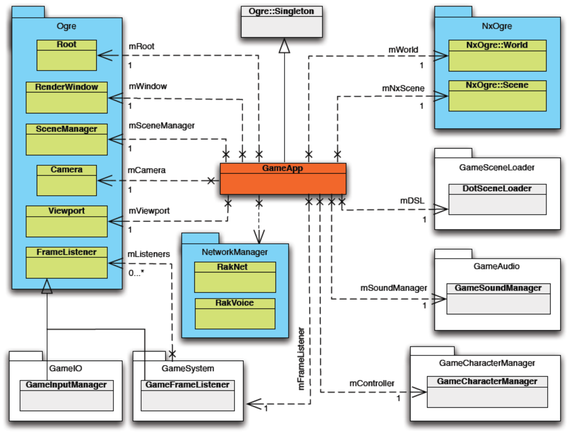
Salvatore Catanese[1] Emilio Ferrara[2] Giacomo Fiumara[1] Francesco Pagano[3]

[1]Dept. of Physics, Informatics Section, University of Messina, Italy [2]Dept. of Mathematics, University of Messina, Italy [3]Dept. of Information Technology, University of Milan, Italy

Goals and Motivations

- Design a framework to support
  - Non-commercial videogames design
  - Academic experimentation
  - Virtual environments simulation
- Exploiting open-source components to:
  - Shorten the design process
    - Facilitating the development
    - Letting developers focus on aspects such as gameplay, game mechanics, etc. instead of technicals
    - Introducing a level of abstraction w.r.t. adopted libraries
  - Develop from scratch required components not already existing as open-source solutions
- With the following characteristics:
  - 3D rendering
  - Physics effects
  - Environmental effects
  - I/O, sound, scene, network management

Framework Architecture



- Rendering Engine
  - Direct3D & OpenGL Rendering Pipelines Implementation
- Scene Manager
  - Bounding Volumes Hierarchies, Binary Spatial Partitioning, Octrees, ...
- Physics Engine
  - Collision Detection
  - Environmental Phenomena
- Character Controller
  - Encapsulation Model, Ragdoll Rigid-body System, ...
- Networking
  - Multiplayer Sessions, VoIP Sessions, ...

... Show-case

Characteristics included

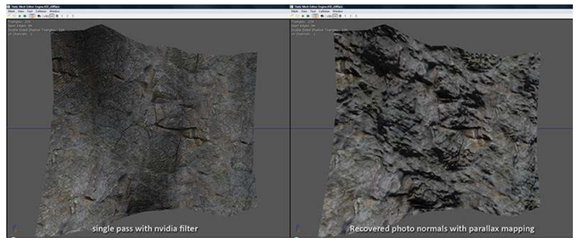
- Terrain and Scene generation and management
  - Maps (heighmpas, density maps, coverage maps, color maps)
  - Outdoor Scenes Generation (via Terrain Generator)
  - Indoor Scenes Design (via Blender)
- Improved rendering techniques
  - Parallax Mapping, Texture Splatting
- Skydome simulation
  - Dynamic sky simulation (clouds, sun/moon, astonomical starfield)
  - Day-night cycle simulation, dynamic lighting, ...
- Particle system simulation
  - Fuzzy phenomena (rain, clouds, water, ...)
- Weather simulation
  - Environmental atmospheric effects
- Fluid dynamics
  - Dynamics of fluids (water, ...) and interactions
- Geometry Simulation
  - Static/Dynamic Level of details management

Adopted Tools

- Blender (Scene Designer) + physics exporter plugin
- ExtendedDotScene (Scene Manager) + physics extension
- PhysX (Physics Engine)
  - Caelum (Sky manager)
  - ParticleUniverse (Particle System Manager)
  - Weather Simulator (Weather Manager)
  - Hydrax (Fluid Dynamics Manager)
  - Paged Geometry (Static/Dynamic Geometry & Lighting Manager)
- OGRE (Rendering Engine)
- OpenAL (3D Audio Library)
- RakNet (Networking Library)
- OpenIS (I/O Manager)

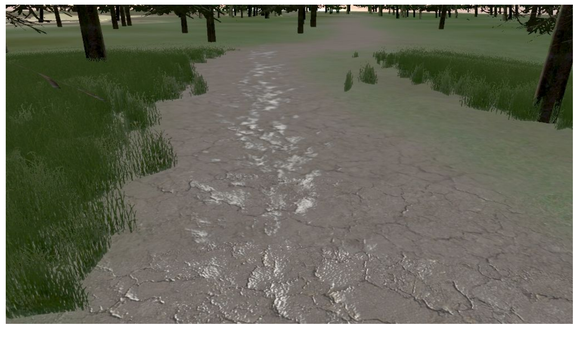
Parallax mapping

This technique improves the realism of texturing flat surfaces, giving the optic illusion of depth. The rendering engine has been extended for supporting the parallax mapping with multiple iterations of the process.



Texture splatting

We extended the graphical features of the rendering engine implementing the texture splatting algorithm; it was adopted (together with *parallax mapping*), for example, to better reproduce the effect of pathways through the forest.



Main Results

- A framework for supporting design and development
- Introduction of levels of abstraction w.r.t. adopted tools and libraries
- Extension of 3D rendering engine with the support of physics and environmental effects
- The modular structure of the framework allows further expansions and personalization
- The platform is released open-source
- A new library from scratch for simulating and managing the weather
- A new scene format "Extended DotScene" DTD
- Integration of six open-source components "coding" new interfaces