

GALA 2007 submission document

**Title:** Graphics for the Masses

**Track:** Student

**Category:** Tool

**Authors:** Nicolas Schulz  
Final year Master Student  
Institute of Computer Science, Augsburg University, Germany

**Contact data:**  
**e-mail:** nicolas@nextgen-engine.net  
**URL:** same as below  
**Phone:** +49 821 598 - 2341 (Elisabeth André)  
**Postal address:** Institute of Computer Science, Augsburg University,  
Eichleitnerstr. 30, 86135 Augsburg, Germany

**URL:** <http://www.nextgen-engine.net>

**Movie file submitted:** NicolasSchulz.avi

**Reference teacher:** Prof. Elisabeth André  
**Affiliation:** Institute of Computer Science, Augsburg University, Germany  
**e-mail:** Elisabeth.Andre@informatik.uni-augsburg.de  
**Phone:** +49 821 598 - 2341

**Process of development:** The crowd simulation demo was accomplished between Jan–April 2007 by Nicolas Schulz in his position as a student researcher. The 3d models were created by Oliver Gerl for several other projects. The graphics engine Horde3D which is used for rendering was created by Nicolas Schulz while working in an independent non-commercial game developers group formed by some students of different study courses.

**Resources used:**  
- Microsoft Visual Studio 2005 for programming  
- Autodesk 3D Studio Max for modelling

**Resources required:** The project was developed and tested on a workstation with Windows XP Prof. OS, 2GB RAM, nVidia Quadro FX 4500 graphics card, on which the application runs with an average of 25 fps. The minimal requirement is a graphics-card with OpenGL 2.0 driver on a system with 1GB RAM.

**Title:** Graphics for the Masses

**1. The application and context of the work**

The application shows downtown Augsburg as a three dimensional interactive model through which the user can navigate in real-time. The streets of the city are populated with 500 characters walking around each with his own destination and independent walk animation. The behavior of the characters can be specified by using LUA, a scripting language which is also used in commercial games like FarCry. All characters have a path planning algorithm based on a visibility graph which allows them to navigate to any location in the city. Furthermore they have special particle based routines to avoid collisions with other characters in a pretty robust way. By having such a high number of characters in a confined environment some interesting emergent collective phenomenas like jamming or organized flows can be observed. The crowd demo is rendered with the open source graphics engine Horde3D which is also developed and maintained by the author of this demo.

**2. Novelty**

In some of the latest computer games developed for next-generation consoles like the Sony Playstation 3 (Assassin's Creed or Kane & Lynch to name two), one can see a clear trend to gameplay which takes advantage of crowd scenes to increase the player's immersion. Nevertheless the number of characters seen on the screen is still quite limited and doesn't usually exceed a limit of 50 to 100. There are already current games in the realtime strategy genre which render hundreds of characters but the quality of rendering and the flexibility of the animation system are rather limited. Usually they use static prebaked animations which don't allow dynamic adjustments like animation blending or mixing. Our goal with this demonstration is to show that it is well possible to use a larger number of fully articulated characters with superior quality without requiring a million dollar budget for development. The key to this is simplicity. Very clean and elaborated concepts are required so that the complexity of developing such an application can still be managed by one student in his spare time. The open source graphics engine Horde3D shows a plenty of these concepts and uses state-of-the-art and beyond techniques to achieve interactive framerates with several hundred visible characters.

**3. The architecture**

The crowd simulation is realized with a system which is a mixture of Craig Reynold's Boids model and other particle based algorithms. Each agent is controlled by a LUA script which is responsible for planning higher level actions. Navigation through the whole scene becomes possible by the application of a visibility graph. The graphics engine Horde3D used for visualization applies a plenty of concepts to enable rendering several hundred animated characters. The scene graph structure and animation system are designed with this requirement in mind. Skinning of character models is done with a vertex shader directly on the graphics card and geometry is optimized for maximum cache efficiency. Furthermore Horde3D supports some special rendering techniques like deferred shading which help to reduce the number of draw passes required for doing lighting calculations. One of the main design goals of the engine is simplicity and clearness of concepts. This makes it possible to maintain the project by one student in his spare time and makes it easier for the community to use and extend it.

**4. Performance**

Since Horde3D is an open source engine, there has been quite a plenty of feedback so far. Most people like the simplicity of the API which enables them to port it to any programming language (C# and Euphoria ports are already available) but has nevertheless the power and flexibility to compete with other graphics engines in the open source or even commercial sector. With new versions released on a regular schedule the user base is growing now and there has already been strong interest from the games and automobile industry to use Horde3D for some demanding projects.