

UnityMol: Visualization with a game engine

Z. Lv^{1,2}, A. Tek², M. Chavent³ and M. Baaden²

¹Ocean University of China, QingDao, China. ²Institut de Biologie Physico-Chimique, Laboratoire de Biochimie Théorique, CNRS UPR 9080, 13, rue Pierre et Marie Curie, F-75005 Paris, France. ³CEA, DAM, DIF, 91297 Arpajon, France.

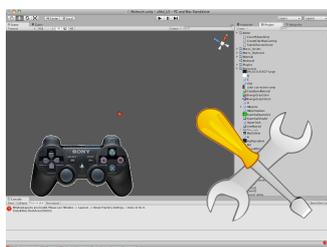


Introduction

In every biology fields technological progress continuously increases the amount of experimental and theoretical data. Hence, building visualization application to retrieve relevant information is more and more complex and requires higher and higher programming skills. This is not achievable by most of non-computer science teams, which then struggle to analyze new data.

In the other hand, the video game market is full of high-quality and visually appealing games playable by everyone and created by small independant companies with low resources. The needs in terms of graphics and interactivity are often comparable to science visualization application.

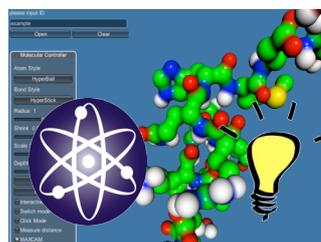
Can the scientific community use game development methods and tools to ease the creation and diffusion of visual applications?



In this work, we explore wether a game engine can be used as a tool to develop and prototype molecular visualization applications for subsequent use in research or education.

We chose the Unity3D game engine (<http://unity3d.com/>) as it proposes a free license and many useful features :

- ✓ Multi-platform (Windows, MacOS, web, mobile devices) applications with minimal programming effort
- ✓ User-friendly interface for creation of 3D application using object oriented concepts
- ✓ Well featured and documented API available in C#, JavaScript and Boo/Python including physics and 3D interaction routines
- ✓ Shader Lab to create and debug shader programs
- ✓ Active and helpful web community

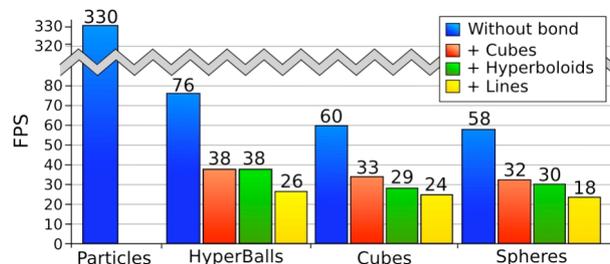


We created UnityMol, a molecular viewer prototype, using the basics features of Unity and see how it is reliable to build science applications

UnityMol

Features	Implementation	Quality	Performances	
Inputs (loading pdb files)	😊	😞	😊	No direct web upload
Molecular representation with built-in methods	😊	😊	😞	Not as fast as pure OpenGL
Point-Sprites representation	😞	😊	😊	Uses particles system
Camera	😊	😊	😊	
GUI	😊	😞	😞	No skins / Greatly decreases FPS
Basic physics	😊	😊	😊	
Interaction	😊	😊	😊	Selection / Translation / Measure distances
Implementation of HyperBalls [1]	😞	😊	😊	Slightly different from CG / Not as fast as pure OpenGL
Visual effects	😊	😊	😊	Pro version only / Scripts from the community
Adding network visualization	😊	😊	😊	

Performances



Benchmark of graphical methods implemented in Frame Per Second (FPS) for a 1280x1024 viewport, Nvidia GTX 285, Mac OS X 10.6, PDB id : 1KX2, 1249 atoms, 1113 bonds.

Multi-Platforms ?

- Linux** : not supported by Unity3D, Wine may be an option
- Mac OS X** : everything ok
- Windows** : HyperBalls does not work with DirectX, Unity bug
- iOS** : not working yet, pro version only
- Web** : no direct upload, can be solved with an additional file server, works better with Firefox

Conclusion

- ⇒ Very good for **prototyping** without advanced programming skills.
- ⇒ **Multi-platform** not perfect but works for the main features.
- ⇒ Useful for web-based viewers but need a web server to do a real web-app.

Perspectives

- ⇒ Optimizations
- ⇒ Explore the mobile features (not free and need work)
- ⇒ Serious gaming

References:

[1] M. Chavent, A. Vanel, A. Tek, B. Levy, B. Raffin, S. Robert, M. Baaden, GPU-accelerated atom and dynamic bond visualization using HyperBalls, a unified algorithm for balls, sticks and hyperboloids, accepted in Journal of Computational Chemistry.