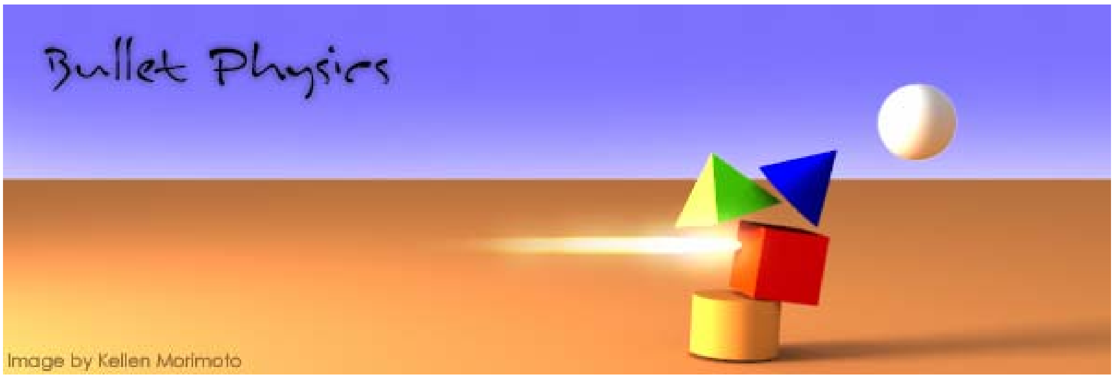


Bullet Physics User Manual

Bullet version: .1-**caliber, Sunday, 31 July 2005**

Introduction

Bullet Physics is an open source collision detection and physics library, related tools, demos, applications and a forum. It is created as a skunk project by Erwin Coumans, ex-Havok employee, free for commercial use. Primary location of the software and the forum is http://www.continuousphysics.com/Bullet/phpBB2/index.php.

Target audience for this work are physics enthusiasts who want to play with collision detection and rigidbody dynamics concepts. Furthermore it can be used by end users in 3D animation and gaming as part of Blender integration (http://www.blender.org). Intention is also to provide a sample integration with Collada Physics importer, which can load physics worlds exported by most 3D authoring packages. See http://www.collada.org for more details.

Quickstart

For end users who want to experiment with Blender should refer to section Blender Bullet Integration. There is a step by step introduction to create and simulate worlds using Bullet. It is recommended to see the provided demos first. This gives an idea of the concepts in this manual.
If you want to use Bullet in your own 3D application, it is best to follow the steps in the CcdPhysicsDemo. In a nutshell:

Create a CcdPhysicsEnvironment object. For each Rigidbody, create a CcdPhysicsController , and add this to the CcdPhysicsEnvironment. Then every frame, you need to update the CcdPhysicsEnvironment by calling the Proceed method, passing the elapsed time.

To construct a CcdPhysicsController, you need to provide:

- a CollisionShape, like a Box, Sphere, Cone, Convex Hull or Triangle Mesh
- Mass and Material properties like friction and restitution
- a link to your Graphics object by deriving from MotionState

Bullet will update the position of every Graphics Objects, by simulating the Rigidbodies, performing Collision Detection, and synchronizing the transform by calling your provided MotionState. There is performance functionality like auto deactivation for non moving objects, and debugging feedback that can be linked to your renderer. See Debugging section for this.

Demos

OpenGL

sample text

CcdPhysicsDemo

sample text

Raytracer

sample text

GjkConvexCastDemo

sample text

ConvexHullDistance

sample text

ConcaveDemo

sample text

CollisionDemo

sample text

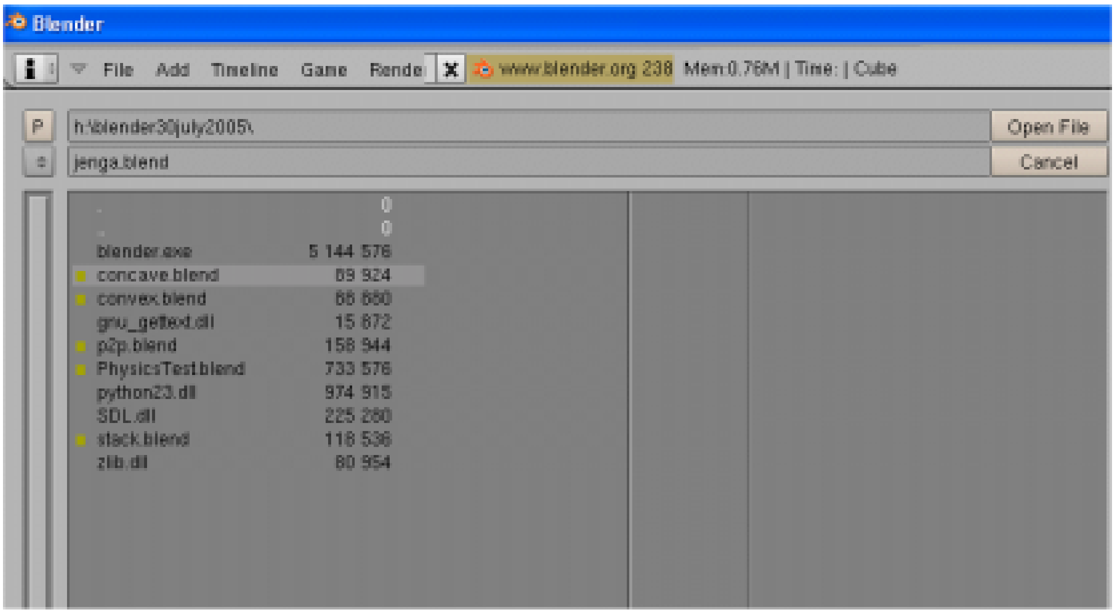
SimplexDemo

sample text

Blender Bullet Integration

Download Blender 2.38 or later, for Windows, Linux or other platforms from http://www.blender.org. A preview can be downloaded in the Blender section of the Bullet forum at http://www.continuousphysics.com/Bullet/phpBB2/index.php.

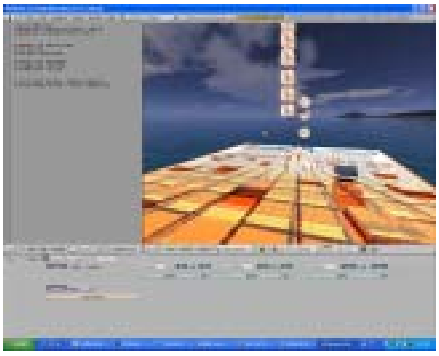
There are some ready made Blender demos that are ready to play. Get them at the Bullet Forum, in the Blender section. Start Blender, and make sure the mouse pointer is in the middle of the 3D window. Then select menubar File/Open or press F1 to open a file:



```
h:\blender30july2005\
jenga.blend

.                          0
..                         0
blender.exe        5 144 576
concave.blend         69 924
convex.blend          66 660
gnu_gettext.dll       15 672
p2p.blend            158 944
PhysicsTest.blend    733 576
python23.dll         974 915
SDL.dll              225 280
stack.blend          118 536
zlib.dll              80 954
```

Left click a file, or use the slider and left click a foldername or .. to navigate, and folder. Then make sure the filename is shown, and left click Open File button, or middleclick on the filename. Another quick option is, instead of pressing F1, press <CTRL>+O, to open the last saved file. Blender remembers this even when you quit the program.



When the file is loaded in Blender, you can play the simulation by moving the mouse cursor in the 3D view, and press ‘P’. Alternatively use menu Game/Start Game. To stop the playback you can press <ESCAPE>.

Todo: Show how to bake keyframes for blender animation. For now there is a script called Game2Ipo, which can be found in http://www.elysiun.com It is not compatible with the Bullet Blender build yet. Watch the Bullet Forum for news on this. Other option will be a better physics - animation integration which is done for ODE, as a Google Summer Project. Bullet and ODE will be supported: http://wiki.blender.org/bin/view.pl/Blenderdev/OdePhysicsIntegration.

Collada Physics Importer

Todo:

ColladaMaya exporter
http://colladamaya.sourceforge.net/

Collada Physics Schema

Collada Importer / Viewer

Bullet SDK Collision Detection and Physics Basics

Latest Bullet Version from CVS:

cvs –z3 –d:pserver:anonymouse@cvs.blender.org:/cvsroot/bf-blender co blender/extern/bullet

Download zipped sources of Bullet and the demos in Bullet Forum "**Bullet Continuous Collision Detection and Physics Library** " in http://www.continuousphysics.com/Bullet/phpBB2/index.php

Browse Bullet On-line:

http://projects.blender.org/viewcvs/viewcvs.cgi/blender/extern/bullet/?cvsroot=bf-blender

Bullet Doxygen Documentation On-line
http://www.continuousphysics.com/Bullet/BulletFull/html/annotated.html
Please not this is not always up to date, it has to be uploaded manually.

It is best to generate the documentation from the sources: Download doxygen, and run 'doxygen' in the Bullet folder. There is a Doxyfile provided, which recursively traverses all sources. For best result also install graphviz for the graphical class diagram and graphical collaboration diagram.

Todo:

CollisionShapes

Box, Sphere, Cylinder, Minkowski Sum, Convex Hull, (Concave) Triangle Mesh, TriangleMeshInterface

BroadphaseCollision

Axis Aligned Bounding Box optimization to prevent expensive calculations between every pair of Rigidbodies.
Simple Broadphase

NarrowPhaseCollision

GetClosestPoint query, GJK

Raycast,

Convex Cast (sweep),

GetTimeOfImpact (linear + angular motion)

CollisionDispatch

todo: find the appropriate collision detection algorithm for a pair of CollisionShapes

Dynamics

ConstraintSolver

Debugging

IDebugDraw

todo:
Explain usage of the IDebugDraw interface. See the CcdPhysicsDemo which implements a callback. Also Blender has a callback for in-game debugging.