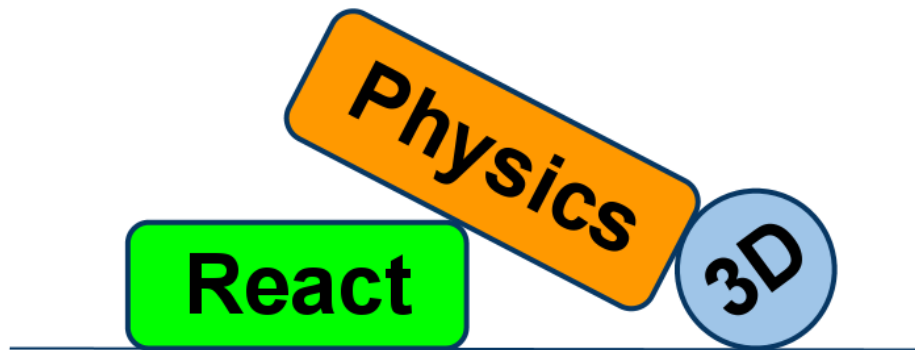


ReactPhysics3D library

User Manual

Daniel Chappuis



<http://code.google.com/p/reactphysics3d/>

January 17, 2011

Contents

1	Introduction	3
2	Information about this version	3
3	License	3
4	Compilation	3
5	Use ReactPhysics3D in your application	4
6	Examples	4
7	Bugs	4

1 Introduction

ReactPhysics3D is a free physics C++ library. ReactPhysics3D is released under the MIT open source license. The project is at its beginning. The goal of the project is to implement a 3D physics framework in C++ that could be used by someone who needs to deal with a physics simulation in its application.

2 Information about this version

The current version is version 0.01. This is the beginning of this project and therefore, the library is not very complete yet. You can use it to try the examples or improve it.

This first release contains the following features :

- Rigid body dynamics
- Discrete collision detection
- OBB bounding volumes
- Broadphase collision detection (SAP algorithm using AABB)
- Narrowphase collision detection (SAT algorithm using OBB)
- Collision response and friction

3 License

The ReactPhysics3D library is under MIT license. For more information, please read the "LICENSE" file.

4 Compilation

You can use the "cmake" program to generate the makefiles or the project files for your IDE. Then, you will be able to compile the library to create the static library file in the /lib folder. Then you can link your program with this static library to use ReactPhysics3D in your application.

To build the ReactPhysics3D library, run the cmake command in the main folder of the library (where you can find the CMakeLists.txt file). For instance to generate the makefiles on a Unix platform, you can use the command :

```
cmake . -G "Unix Makefiles"
```

Notice that the previous command generates the makefiles to compile in DEBUG mode. If you want to generate the makefiles to compile in RELEASE mode,

you should use the following command :

```
cmake . -G "Unix Makefiles" -DCMAKE_BUILD_TYPE:STRING=Release
```

then, you can compile the code to build the static library in the `/lib` folder with the following command in the main directory :

```
make
```

5 Use ReactPhysics3D in your application

First build the static library of ReactPhysics3d as described above to get the static library file in the `/lib` folder. Then, in your code, you have to include the ReactPhysics3D header file with the line :

```
#include "reactphysics3d.h"
```

Note that the `"reactphysics3d.h"` header file can be found in the `/src` folder.

Don't forget to link your application with the ReactPhysics3D static library. Then, you should be able to compile your application using the ReactPhysics3D library. You should also take a look at the examples to get a better idea of how to use the ReactPhysics3D library.

6 Examples

You can find some OpenGL demos in the `/examples` folder of the library. Before compiling the examples, you have to compile the `reactphysics3d` library as described above. To compile the examples, you will need to have the `"freeglut"` library. Note that you can also use GLUT instead but you need to change the `"freeglut"` include header file in the `main.cpp` by the GLUT header. When you compile the examples, don't forget to link with :

```
-lGL -lGLU -lglut -lreactphysics3d
```

7 Bugs

If you find some bugs, don't hesitate to report them on the issue tracker of the ReactPhysics3D website at :

```
http://code.google.com/p/reactphysics3d/issues/list
```

Thanks a lot for reporting the bugs that you find. It will help us to correct and improve the library.