Code Reading Report V2

2015011308 唐适之

1 Introduction

1.1 Purpose of the Report

This report describes the architecture and system design of *LiquidFun*, including its components, workflow, advanced techniques, and key design principles. This is not a API reference, and will not cover all the detailed functions.

1.2 Scope of the Project

LiquidFun is a 2D rigid-body and fluid simulation C++ library for games based upon *Box2D*. It provides support for procedural animation of physical bodies to make objects move and interact in realistic ways[1]. It brings particles to *Box2D* so as to simulate fluid. *LiquidFun* is not a physical simulator for scientific analysing. It provides an approximate but efficient way of calculation. *LiquidFun* is either not a game framework. It only provides an interface to calculate the physics, but does not involves in displaying and controling.

1.3 Reference Material

- *LiquidFun Programmer's Guide.*

  http://google.github.io/liquidfun/Programmers-Guide/html/index.html

- *LiquidFun API Documentation.*

  http://google.github.io/liquidfun/API-Ref/html/index.html

2 System Overview

2.1 Architectural Design

There are three major modules inherited from *Box2D*: Common, Collision and Dynamics. The Common module is an infrastructure, which provides memory allocation, math, settings and data structures. The Collision module takes charge of static geometry, which defines

[1] http://google.github.io/liquidfun/ Overview

shapes and handles geometric queries. The Dynamics module simulates the physics using the two module above.

The Dynamics module can only do with rigid bodies, and therefore a Particle moudle is added in *LiquidFun*, which provides simulation of particles. There is also a Rope module which seems to be uncompleted however.

The figure below is the relations among these modles, extended from the graph in *Box2D* document, which describes the three *Box2D* modules. In this figure, the modules below make use of the modules above.

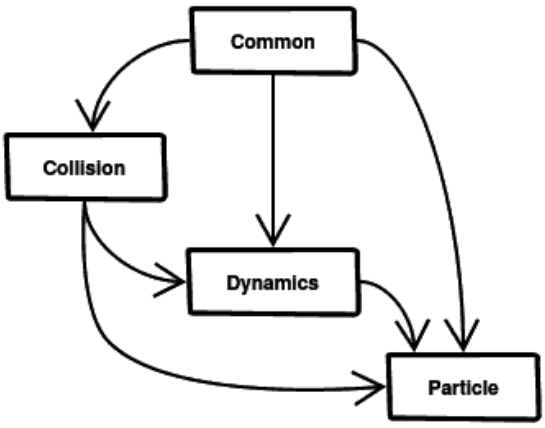

Figure 1: Modules