

IRON BALL FIGHT

REPORT, MANUAL AND DOCUMENTATION

Yihan Zhang (Uni:yz2567)

February 25, 2014

INTRODUCTION

Iron ball fight is a 3D shooting game which is still in continuous developing. At the time of submission, the game only implemented basic game logics. That is, you can shoot off geometric objects in the game space. You can follow further changes in my github repository .

INSTALLATION

1. For Linux user, if you have bash installed in your machine, you can just move to the direction and type:

```
./install
```

and every thing will be fine. If you can't run this script, or feel unsafe about what it is doing, you could follow general installation described in part 2.

2. The general installation method should work for all Linux and Mac systems. Firstly, you need to build up all the directories that is essential for cmake to run. At root directory, type:

```
mkdir ./bin ./bin/shader ./bin/obj ./build
```

Secondly, run cmake:

```
cd build
```

```
cmake ..
```

Finally, you could find the executable at:

```
./bin/IronBallFight
```

Just run it and have fun!

ABOUT THE GAME

CODE DESCRIPTION

1. Implemented features:

- Full 3D camera movement. Using a quaternion based method, you can move the camera freely in a 3D space, including any kind of translation and rotation. Thus view from any position at any angle is always approachable.
- Shader based basic light effect. The light effect inside the game are fully implemented by shaders using GLSL. It makes the game looks more realistic and fantastic. Although the light source is currently invisible, you could just imagine it is emitting ultraviolet rays, and all geometries are made of some kind of material that changing this ray into visible lights.
- Collision detection with simple geometry. The game could successfully detect the collision between your bullet (which is, sadly, invisible) and geometries in the space. All shooting events are based on this simple algorithm.

- External geometry loader. This function comes from a slightly modified external codes (you could find the source in **Reference and External Sources** part of this section). This makes it possible to load the geometry I created in Blender.

2. **Reference and External Sources:**