Dandelion环境配置

开发者文档，或者Click Here，其实这个模型就是白嫖卡梅的Scotty3D，我愣是看不出有啥实质性区别

1. 编译与运行

1.1. 环境

1️⃣编译器支持C++17，Cmake版本不低于3.14

```
1.Linux终端输入g++ --version可查看g++版本，从g++8开始C++17被完整地支持
注意g++是C++编译器而gcc是C的编译器
2.同样方式查看Cmake版本
```

此次的版本：

```
dhy@dhy-virtual-machine:~/桌面$ g++ --version
g++ (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

dhy@dhy-virtual-machine:~/桌面$ cmake --version
cmake version 3.22.1
CMake suite maintained and supported by Kitware (kitware.com/cmake).
```

但是保险起见对g++进行一下升级：gcc-12.1.0.tar.gz下载

```
1.解压：
tar zxvf gcc-12.1.0.tar.gz

2.下载GCC文件与依赖：
cd gcc-12.1.0
./contrib/download_prerequisites

3.生成makefile编译文件：在gcc-12.1.0目录内执行以下指令
mkdir gcc-build-12.1.0
cd gcc-build-12.1.0
../configure -enable-checking=release -enable-languages=c,c++ -disable-multilib

4.然后编译：此次虚拟机有8个内核，所以对现有目录输入(编译时间真的久)
make -j8

5.安装：在当前目录下输入
sudo make install

6.结果
```

```
dhy@dhy-virtual-machine:~/桌面$ g++ -v
使用内建 specs。
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=/usr/local/libexec/gcc/x86_64-pc-linux-gnu/12.1.0/lto-wrapper
目标：x86_64-pc-linux-gnu
配置为：../configure -enable-checking=release -enable-languages=c,c++ -disable-multilib
线程模型：posix
Supported LTO compression algorithms: zlib
gcc 版本 12.1.0 (GCC)
```

2️⃣Linux上Dandelion仅对 KDE/GNOM桌面环境支持'

本次的桌面环境为：

```
dhy@dhy-virtual-machine:~/桌面$ echo $XDG_CURRENT_DESKTOP
ubuntu:GNOME
```

3️⃣显卡驱动至少支持 OpenGL 3.3 Core Profile

```
1.安装glxinfo工具: sudo apt install mesa-utils
2.查询 OpenGL版本:
dhy@dhy-virtual-machine:~/桌面$ glxinfo | grep "OpenGL version"
OpenGL version string: 4.1 (Compatibility Profile) Mesa 22.0.5
3.查询OpenGL Profile支持
dhy@dhy-virtual-machine:~/桌面$ glxinfo | grep "OpenGL core profile version"
OpenGL core profile version string: 4.1 (Core Profile) Mesa 22.0.5

显然满足要求
```

4️⃣关于虚拟机：Windows虚拟机无法使用Dandelion框架，Linux目前暂时无问题，本次采用Linux虚拟机

1.2. 依赖

1️⃣源代码、静态库和预编译的可执行文件都在GitHub上开源

2️⃣GNOME桌面环境需要Zennity/matedialog/qarma实现基于portable-file-dialogs的对话框

检查是否安装：Zenity已经安装

```
dhy@dhy-virtual-machine:~/桌面$ which zenity
/usr/bin/zenity
dhy@dhy-virtual-machine:~/桌面$ which matedialog
dhy@dhy-virtual-machine:~/桌面$ which qarma
dhy@dhy-virtual-machine:~/桌面$
```

1.3.项目结构：整个项目的结构

```
dandelion
├── CMakeLists.txt
├── CREDITS
├── deps
│   ├── assimp
│   ├── Eigen
│   ├── glad
│   ├── glfw
│   ├── imgui
│   ├── portable-file-dialogs.h
│   ├── spdlog
│   ├── fmt
│   └── stb
├── docs
├── README.md
├── resources
└── src
├── main.cpp
├── geometry
├── simulation
├── platform
├── render
├── scene
├── ui
└── utils
```

1.4. Linux平台下的编译

直接把这个仓库丢到Linux环境下，根目录改名dandelion

打开终端，进入 dandelion 目录并执行如下命令

```
dhy@dhy-virtual-machine:~/桌面/dandelion$ mkdir build
dhy@dhy-virtual-machine:~/桌面/dandelion$ cd build
```

1.4.1. 执行`$ cmake -S .. -B . -DCMAKE_BUILD_TYPE=Debug`

1️⃣指令含义：

1. `-S ..`: 告诉CMake从哪获取源码，`-S`标志后面的路径是源代码的路径，`..`表示父目录(根目录)
2. `-B .`: 告诉CMake在哪生成构建文件，`-B`后路径是构建目录的路径，`.`表示当前目录
3. `-DCMAKE_BUILD_TYPE=Debug`: 是为CMake设置一个变量方法，此处设置`CMAKE_BUILD_TYPE`变量为`Debug`，意味着项目会以调试模式进行编译

2️⃣报错与解决：

```
CMake Error at /usr/share/cmake-3.22/Modules/FindPackageHandleStandardArgs.cmake:230 (message):
  Could NOT find X11 (missing: X11_X11_INCLUDE_PATH X11_X11_LIB)
Call Stack (most recent call first):
  /usr/share/cmake-3.22/Modules/FindPackageHandleStandardArgs.cmake:594 (_FPHSA_FAILURE_MESSAGE)
  /usr/share/cmake-3.22/Modules/FindX11.cmake:457 (find_package_handle_standard_args)
  deps/glfw/CMakeLists.txt:208 (find_package)
```

这是因为在CMake的时候，试图找到X11库，但是没能找到，所以要安装X11

此外一并安装X11的很多拓展与开发头文件，量大管饱以防报错

```
sudo apt install libx11-dev libxrandr-dev libxinerama-dev libxcursor-dev libxi-dev libxext-dev libxfixes-dev libxft-dev libxkbcommon-dev libxmu-dev libxt-dev libxv-dev libxxf86vm-dev libgl1-mesa-dev libglu1-mesa-dev
```

1.4.2. 执行`$ cmake --build . --parallel 8`

1️⃣指令含义：

1. `--build .`: 告诉CMake在当前目录`.`中查找CMake生成的构建文件，并使用这些文件来构建项目，`--build`指示CMake执行构建过程
2. `--parallel 8`: 它告诉CMake并行构建项目，并使用最多8个线程进行构建。这可以加速构建过程，特别是在多核心的机器上

2️⃣报错与解决：

```
/usr/bin/ld: 找不到 -ldandelion-ray-debug: 没有那个文件或目录
/usr/bin/ld: 找不到 -ldandelion-bvh-debug: 没有那个文件或目录
collect2: error: ld returned 1 exit status
gmake[2]: *** [CMakeFiles/dandelion.dir/build.make:695：dandelion] 错误 1
gmake[1]: *** [CMakeFiles/Makefile2:169：CMakeFiles/dandelion.dir/all] 错误 2
gmake: *** [Makefile:136：all] 错误 2
```

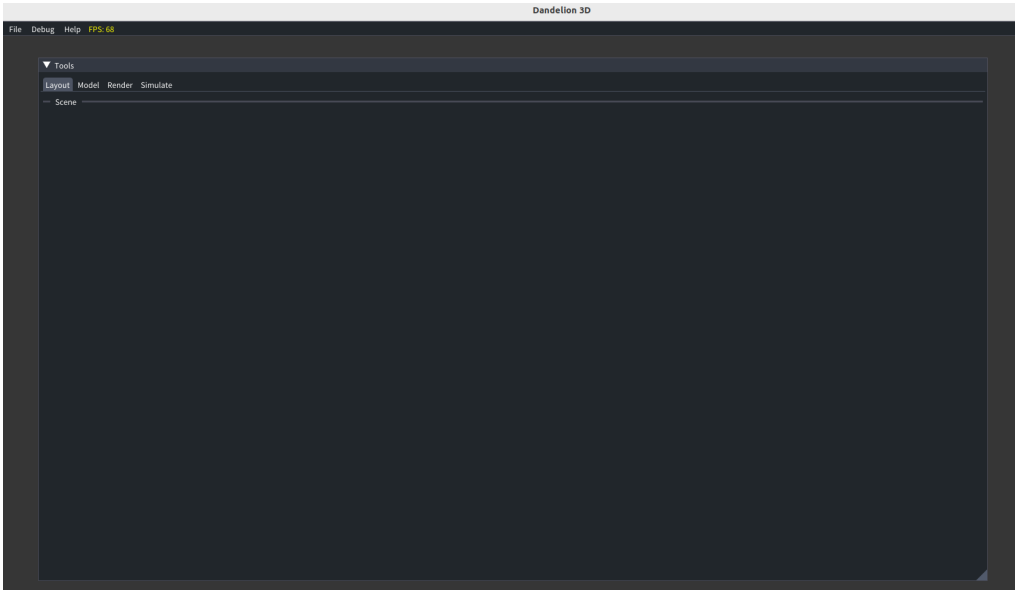
其实这是对于群文档的内容存在遗漏，可以看到这个仓库的右边有一个Release

在本次实验的环境中下载这个dandelion-lib-1.0.1-linux-x64-gcc.tar.gz，顺带避雷这个dandelion-1.0.1-linux-x64.tar.gz 不行

Linux环境下解压，里面四个全部放到deps目录下，然后终于构建成功

1.4.3. 运行程序`$ ./dandelion`

1️⃣生成开始界面：



2️⃣终端输出日志：

```
dhy@dhy-virtual-machine:~/桌面/dandelion-main/build$ ./dandelion
[Default] [info] Dandelion 3D, started at 2023-09-19 23:05:59+0800
[Platform] [debug] Try to create OpenGL context 4.6
[Platform] [info] Failed to create OpenGL context 4.6, drop back to 4.3
[Platform] [debug] Try to create OpenGL context 4.3
[Platform] [warning] Failed to create OpenGL context 4.3, drop back to 3.3
[Platform] [debug] Try to create OpenGL context 3.3
[Platform] [info] runtime OpenGL context: 4.1 (Core Profile) Mesa 22.0.5
[Platform] [info] Physical screen size: 508x285 mm, diagonal: 22.93 in
[Platform] [info] screen DPI: 96.06, scale factor: 1.0
[Platform] [info] The loaded vertex shader: resources/shaders/vertex.glsl
[Platform] [info] The loaded fragment shader: resources/shaders/fragment.glsl
[Platform] [debug] Vertex shader 1 compiled successfully.
[Platform] [debug] Fragment shader 2 compiled successfully.
[Platform] [info] Shader program 3 link succeeded
```

2. 模式与功能

2.1. 四个模式

```
布局模式：这是Dandelion启动时的模式，用于放置、移动、旋转和缩放物体
建模模式：对物体进行形变和各种几何处理操作
渲染模式：调整光源和相机参数，将场景渲染成图像
物理模拟模式：设置物体的动力学属性，通过求解运动方程生成动画
```

2.2. 功能

1️⃣帮助窗口：Help -> Usage

2️⃣加载物体：File -> Load File as a Group，选取cube.obj文件

.obj/.dae文件中有多个mesh，每个文件被加载成一个组，每个mesh被加载成一个物体

3️⃣之后四个模式的操作详见文档

2.3. Problems

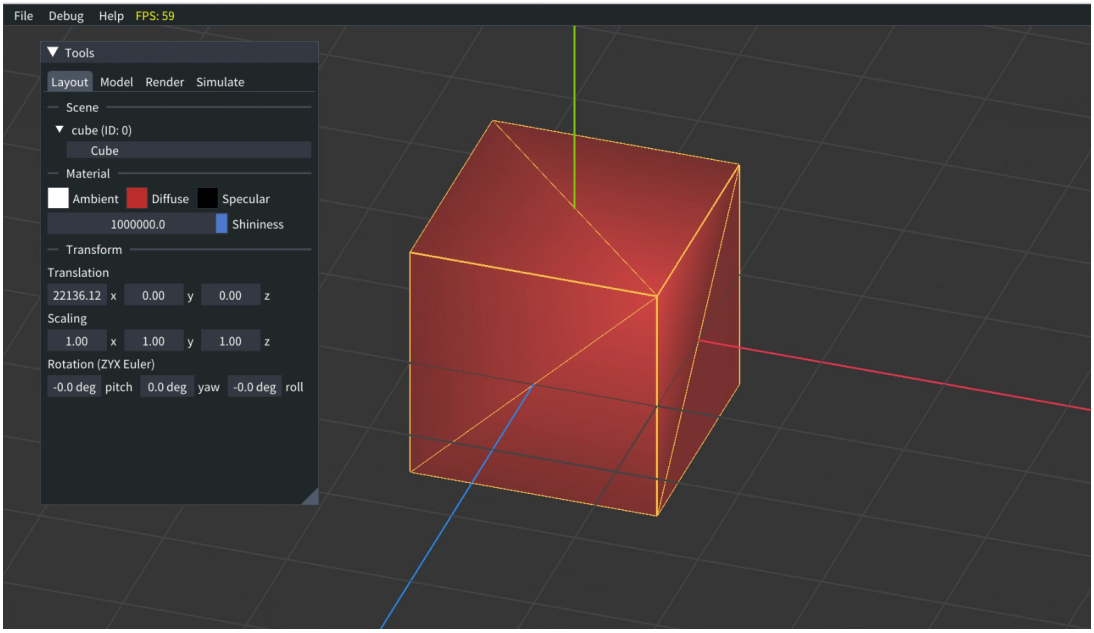
不出意外的出意外了，虚拟机上运行还是出问题了，VMware的OpenGL的bug全责，解决方案就是买了台Ubuntu主机

3. 提交

加载 cube.obj 文件后，在布局模式下将它的 x 坐标(Translation属性的x项)设为22136.11582

由于 GUI 上只会显示两位小数，设置完成后为 22136.11

然后提交这个图片


File
Debug
Help
FPS: 59
Tools
Layout
Model
Render
Simulate
Scene
cube (ID: 0)
Cube
Material
Ambient
Diffuse
Specular
1000000.0
Shininess
Transform
Translation
22136.12 x
0.00 y
0.00 z
Scaling
1.00 x
1.00 y
1.00 z
Rotation (ZYX Euler)
-0.0 deg pitch
0.0 deg yaw
-0.0 deg roll