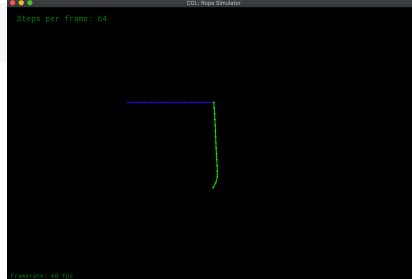
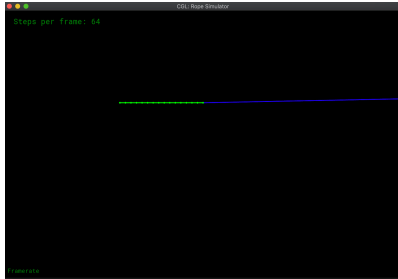



```
F=ma
v(t+1) = v(t) + a(t) * dt
x(t+1) = x(t) + v(t) * dt // For explicit method
x(t+1) = x(t) + v(t+1) * dt // For semi-implicit method
```

运行./ropesim。仿真应该就开始运行了，但是只有 3 个结点，看起来不够多。在 application.cpp 文件的最上方，你应该可以看到欧拉绳子和 Verlet 绳子的定义。改变两个绳子结点个数（默认为 3 个），比如 16 或者更多。

运行 ./ropesim -s 32 来设置仿真中每帧不同的仿真步数。尝试设置较小的值和较大的值（默认值为 64）。



1.3 显式 Verlet

Verlet 是另一种精确求解所有约束的方法。这种方法的优点是只处理仿真中顶点的位置并且保证四阶精度。和欧拉法不同，Verlet 积分按如下的方式来更新下一步位置：

$$x(t+1) = x(t) + [x(t) - x(t-1)] + a(t) * dt * dt$$

利用这种方法，我们可以仿真弹簧系数无限大的弹簧。

在本次作业中，上述公式的 $a(t)$ 特指重力加速度，不包含弹簧力加速度。具体实现时，首先利用上述公式计算重力影响下，质点的位置变化。然后用解约束的方法来进行进一步更新质点位置：简单地移动每个质点的位置使得弹簧的长度保持原长。假设弹簧当前长度和稳定状态长度的差值 L ，每个质点的修正向量长度应为 $L/2$ ，方向为一个质点指向另一质点。

只要对每个弹簧执行这样的操作，我们就可以得到稳定的仿真。为了使运动更加平滑，每一帧可能需要更多的仿真次数。

1.4 阻尼

向显式 Verlet 方法积分的胡克定律中加入阻尼。现实中的弹簧不会永远跳动-因为动能会因摩擦而减小。阻尼系数设置为 0.00005, 加入阻尼之后质点位置更新如下：

$$x(t+1) = x(t) + (1 - \text{damping_factor}) * [x(t) - x(t-1)] + a(t) * dt * dt$$

1.5 说明

你应该修改的函数是：

- rope.cpp 中的 Rope::rope(...)
- rope.cpp 中的 void Rope::simulateEuler(...)
- rope.cpp 中的 void Rope::simulateVerlet(...)

程序实现

2.1 安装依赖

本次作业需要预先安装 OpenGL, Freetype 还有 RandR 这三个库。可以通过以下命令进行安装：

```
$sudo apt install libglu1-mesa-dev freeglut3-dev mesa-common-dev  
$sudo apt install xorg-dev
```

2.1 编译、运行程序

请下载工程的代码框架并通过下面的命令创建工程：

```
$ mkdir build  
$ cd build  
$ cmake ..  
$ make
```

之后，你应该可以使用命令 `./ropesim` 来运行仿真。

作业描述与提交

3.1 作业提交

将PDF报告文件和代码压缩打包提交到zhangzk3@mail2.sysu.edu.cn邮箱，邮件及压缩包命名格式为：hw7_姓名_学号。

3.2 作业描述

Task1：完善rope.cpp 中的 Rope::rope(...)。（20分）

完善该构造函数，运行程序并截图。

Task2：完善rope.cpp 中的 Rope::simulateEuler(...)（40分）

- 在application.cpp文件内将欧拉绳子的质点数量修改为16。
- 分别实现显式欧拉法、半隐式欧拉法，附上代码，运行程序并截图。
- 比较在不同的步数下（如16,64,256,1024），欧拉绳子的摆动情况。
- 比较阻尼系数为0, 0.001,0.01,0.1下，欧拉绳子的摆动情况。

注意：此处，阻尼力使用简化的做法， k_d 为阻尼系数：

```
f_damping = - k_d * velocity
```

Task3：完善rope.cpp 中的 Rope::simulateVerlet(...)（40分）

- 在application.cpp文件内将Verlet绳子的质点数量修改为16。
- 实现显式Verlet法，附上代码，运行程序并截图。
- 比较在不同的步数下（如16,64,256,1024），Verlet绳子的摆动情况。
- 比较阻尼系数为0, 0.00005,0.0005,0.005下，Verlet绳子的摆动情况。

