READ ME

姓名：周宁
班级：111171
学号：20171004140

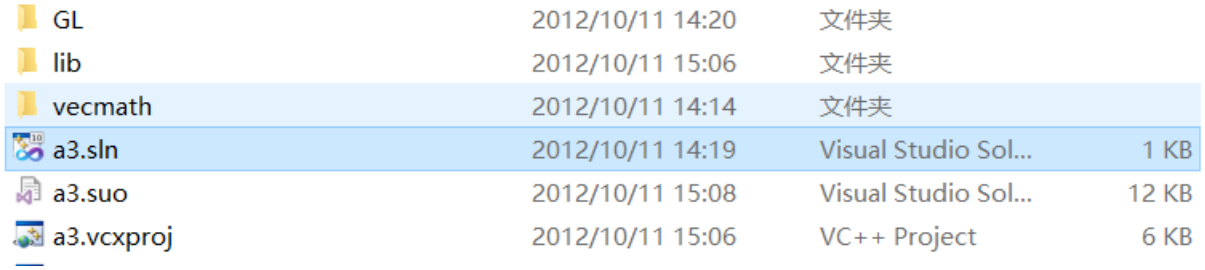
一、实习思路：

1. 编译和运行程序

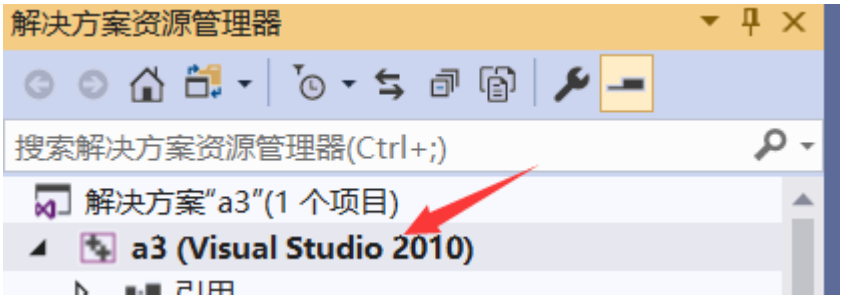
①首先下载 VS 2010 相对应的工具集（最新版的 VS 2019 编译不了）

②将“作业 3”文件夹下的“distrib”文件夹，复制到平时写代码的工作目录下（不复制也可以）

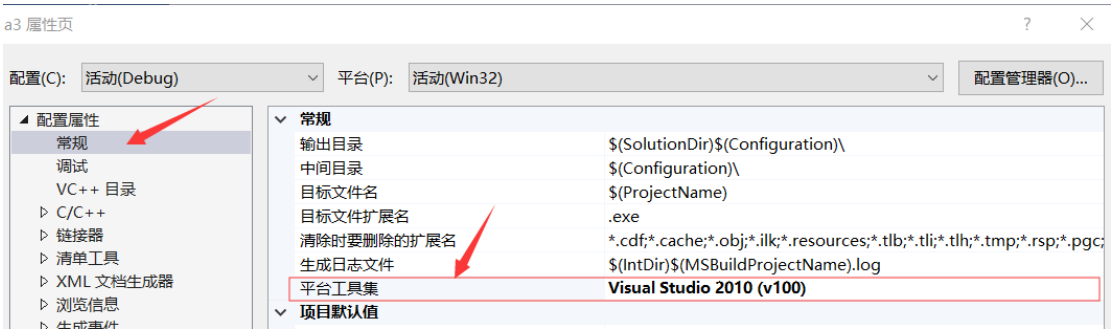
③直接双击“distrib”文件夹下的 a3.sln 文件即可自动打开 Visual Studio 进行代码编写



④在解决方案资源管理器里面选中 a3 然后右键选中属性



⑤选中配置属性->常规，然后选择平台工具集，将其调成“Visual Studio 2010 (v100)”



2. **编写时间积分器(TimeStepper.xpp)**

时间积分器需要我们实现类 ForwardEuler、Trapzoidal 中的 takeStep(ParticleSystem* particleSystem,float stepSize) 函数。这个函数会更新传进来的粒子系统 particleSystem 的状态，从后面的题目看来，可以简单的理解为利用 evalF（）求导得到的速度（当然这是简单一阶 ODE 理解），然后对速度积分得到位移，再结合初始值就得到了最终的位置了。

①对于类 ForwardEuler，根据文档当中的内容为 X(t + h)=X + hf(X,t)，可以简单易懂的得到代码为 “state[i] = state[i] + stepSize * f[i]”，其中数组 f 为 evalF（）的返回值。从数学角度分析就是 $y = y_0 + h*x$，其中 x 为 y 的导数。

②对于类 Trapzoidal，根据文档中的内容为 X(t + h)=X +(f0 + f1) h/2，也可以简单移动的得到代码为“state[i] = state[i] + stepSize * (f0[i]+f1[i])/2”。从数学的角度分析就是 $y = y_0 + h*(\frac{X_0+X_1}{2})$，其中 x 为 y 的导数。

3. **实现简单例子(simpleSystem.cpp)**

需要实现类 simpleSystem，主要实现类中的函数 vector<Vector3f> evalF(vector<Vector3f> state) 和 void draw()

①实现函数 evalF(vector<Vector3f> state)，该函数的主要功能就是对传进来的 state 进行求导，然后返回对应的求导后的数组。在 simpleSystem 系统当中，已经写明了导数为 $f(x,t) = \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix}$（上课也讲过，这其实是一个圆，当然文档也有写）。代码就很简单，直接就是对数组 state 进行遍历，然后 state[i]对应的导数为 Vector3f v(-state[i].y(), state[i].x(), 0.)。

②实现函数函数 draw()，这个函数的组要作用是将粒子画出来。比较简单直接对 state 进行遍历，然后将 state[i]作为位置画出来即可。

③在 SimpleSystem 中的构造函数中给 m_vVecState，添加元素，我选择添加了一个

Vector3f（1，0，0）元素（半径为 1 的圆）。

实现后运行程序可以看到一个粒子在那里转转转，如果将系统调成 ForwardEuler 会越转半径越大，然后就飞出去，而 Trapzoidal 和 RK4 都不会飞。

4. **实现物理仿真**（pendulumSystem.cpp）

需要实现类 pendulumSystem，主要实现类中的函数 vector<Vector3f> evalF(vector<Vector3f> state) 和 void draw()

①实现数据结构 particle，在粒子里面存储其所连接的粒子（m_link_particles）、弹性系数（m_Elastic_coefficient）、静态长度（r）、粒子的摩檫力系数（m_Capacitance_coefficient）、质量（m_mass）以及是否一直保持静止（isStill）

②实现函数 evalF(vector<Vector3f> state)，该函数的组要内容是对传进来的 state 进行求导，但是和 SimpleSystem 系统求导不太一样，是对二阶常微分方程进行求导，但是将二阶常微分方程转化为一阶常微分方程然后进行求导。简单的说就是 state 数组当中偶数下标当中存储位置坐标，奇数下标当中存储速度坐标，然后进行求导后求导返回的数组 f 当中偶数下标存储速度，奇数下标存储加速度，速度等于 state[i]奇数下标的值，加速度可以通过求解这个粒子受到的合力，然后除以质量进行求解。所以需要对粒子受到的弹力、阻力（摩檫力）、重力进行计算，并且计算出来的力均为向量，所以直接相加即可，不需要考虑这么多。

③计算重力，重力可以直接由质量乘以 10 简单的得出数值，然后直接 Vector3f gravity(0, -g, 0)转化为向量即可。

④计算弹力，采取对每个粒子连接的粒子（m_link_particles）进行遍历，然后采用公式 $F(x_i, x_i^{'}, m_i) = -k(\| d \| - r)\frac{d}{\| d \|}$,其中$d = x_i - x_j$ 进行计算，这个公式非常巧妙，计算出来的弹力直接就是向量，所以不需要考虑弹力的方向，不论是拉伸还是压缩方向都自动求好了。但是对于两个连接的粒子由要求，两者不能在同一个位置，即 d 不能等于 0，如果 d 等于将无法进行计算，也不清楚弹力的方向。

⑤计算摩檫力，计算摩檫力可以采用公式 $F(x_i, x_i^{'}, m_i) = -kx_i^{'}$，求出来的也是一个向量，方向和速度方向相反（高中物理题目做过很多）。

⑥实现函数函数 draw()，这个函数的组要作用是将粒子画出来。比较简单直接对 state 进行遍历，然后将偶数的下标（代表位置）画出来即可。

⑦实现画弹簧函数 void drawSpring(int i)，只需要将两个粒子之间的线画出来即可。

⑧实现类的构造函数 PendulumSystem(int numParticles,vector<particle> particles):ParticleSystem(numParticles)，传进来的参数进行直接赋值即可，其中

实现之后，先在 main.cpp 当中更改 system，然后可以静止一个粒子（求导时先判断 isStill，如果为真，合力为 0），然后加入另外一个粒子，设置好相应的变量就可实现单摆。也可以加入四个粒子，静止其中的一个粒子，就可以形成多粒子链。

5. **实现粒子系统布料（ClothSystem.cpp）**

需要实现类 ClothSystem，我将类 ClothSystem 改为继承自 PendulumSystem，然后主要实现其中的函数 vector<Vector3f> evalF(vector<Vector3f> state) 和 void draw()

①实现函数 vector< particle> createParticles(int s,float r)，生成对应的粒子，并且会给每个粒子生成相对应的弹簧（如：结构弹簧、抗剪弹簧、抗弯弹簧），其中 s 为粒子数量，r 为粒子之间的间距。

② 实现 evalF(vector<Vector3f> state) 函数，在函数当中会调用 PendulumSystem::evalF(state)，除此之外，会进行有无风力的判断，有风力就会加上风力。

③实现 draw()函数，在这个函数当中会判断是画格网还是画结构弹簧，如果是画格网就采用三角形建模的方法，画出一个个三角形即可，如果是画结构弹簧，就画出结构弹簧即可。

④进行调参，需要调试各种弹簧的静态长度、弹簧的弹性系数、物体质量、阻力系数以及步长，这个是最难，需要调试到合适才能够使得布料看起来更像布料。

6. **附加分**

附加分只是实现了加风，在 evalF 的时候给特定的位置多添加一个方向的力就类似于加风，比较简单。球体碰撞也想做想的是在距离球心快到半径距离（距离球心距离 d-r<0.1）的时候就停止运动即可，但是这样无法实现那种晃来晃去的碰撞比较失败。

二、实习中遇到的困难和解决方法

1. 理解时间积分器

解决方案：查看 ppt 物理动画，然后看上面的尚未分方程 ODE，慢慢理解解决问题。

$$\frac{d\,\mathbf{X}(t)}{dt} = f(\mathbf{X}(t), t)$$

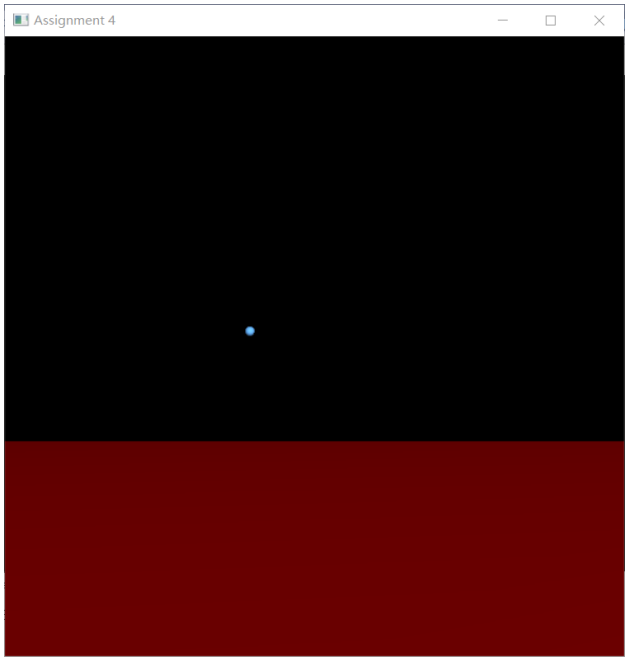
- 给定函数 *f(X,t)* 求 X(t)
- 典型的，初值问题：
  - 给定 $X(t_0)=X_0$
  - 求 $X(t)$ 对于 $t > t_0$

（参考的 ppt）

2. 调试粒子布料系统
解决方法：这个真的是比较考验经验，只能先设置好一个变量然后再慢慢的进行调试最后得出比较满意的结论。

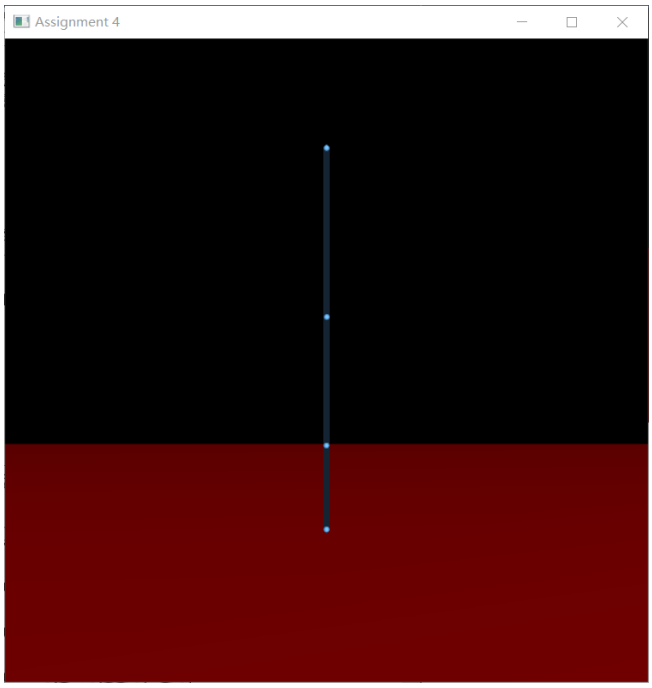
三、实习结果展示

simpleSystem 系统



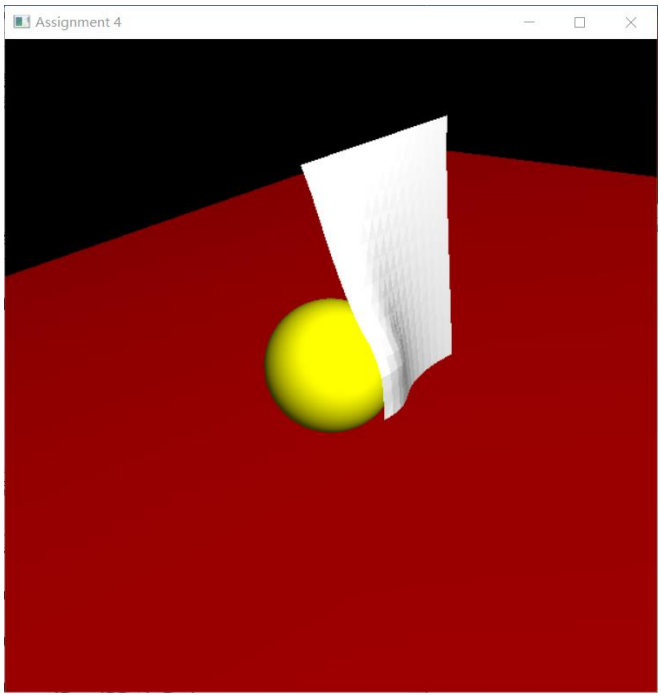
会看见一个粒子转转转

pendulumSystem



其中第一个粒子固定，看见三个粒子弹弹

ClothSystem



四、实习总结

本次实习是对物理动画的一次练习，实现了基本的三个粒子系统，实现了基本的力场。在这个系统里面体现物理的常识（让我们知道了大学学习物理还是有用的），然后我在里面还玩了一些学习的物理实验，比如：单摆等，感觉特别有趣，总体来说根据实习又学习了 ppt 上的很多东西，收获不少。

不足之处有很多，首先是球体的碰撞，我是将粒子碰撞球后就停止了运动让其处于球面保持静止（类似粘在表面），但是如果想要在球表面滑来滑去没有办法实现，然后是 ClothSystem 中参数设置并不好，稳定性比较差，感觉还是没有调试出最完美的参数，最后是附加分也没有完成的很多，如果有时间，应该继续钻研，将这些都做出来。