

Python 大作业：

三体运动的计算机图像模拟

5130379033 姜霁恒 周一 6-8 节

本程序主要的功能就是模拟三体运动的图像。

三体问题是天体力学中的基本力学模型。它是指三个质量、初始位置和初始速度都任意的可视为质点的天体，在相互之间万有引力的作用下的运动规律问题。现在已知，三体问题不能精确求解，即无法预测所有三体问题的数学情景，只有几种特殊情况已研究。三体问题最简单的一个例子就是太阳系中太阳、地球和月球的运动。在浩瀚的宇宙中，星球的大小可以忽略不计，所以我们可以把它们看成质点。如果不计太阳系其他星球的影响，那么它们的运动就只是在引力的作用下产生的，所以我们可以把它们的运动看成一个三体问题。

由于目前的数学技术并不能达到物理的要求，因此三体问题不能给出一个代数解，只能在精确给定全部初始参量的情况下求出数值解。本程序即利用了三体问题的这一性质，通过用户自定义三个星球的初始状态（质量，位置和速度）来观察在此后三个星体的运动图像。但是由于时间有限，本程序尚未制作关于三个星球碰撞检测的内容。如有需要，我将在以后完善这一部分功能。

制作这一程序的考虑是有多方面因素在内的。我是软件工程系的学生，高中 3+1 选的是物理，对一些物理问题有浓厚的兴趣。另外，我的成绩并不算名列前茅，在年级里大概处于中游水平，因此能力也并不是非常之强，加之本学期我们还有其他的编程大作业，所以选择了制作这个略微简单的程序，没有复杂的算法和数据结构，也希望老师可以理解。

程序中选取所有单位为国际单位。利用的物理公式有：

$$F_1 = F_2 = G \frac{m_1 \times m_2}{r^2} \quad a = \frac{\Delta v}{\Delta t} \quad \vec{F} = m\vec{a}$$

这其中，大 G 的数值为一个常量，在标准单位制下取 6.67428×10^{-11} 。但是在本程序中，为了能在程序中更好的演示运动结果，将大 G 的值增大 1×10^{11} 倍。另外，为了方便演示，我把时间加速了 1.5 倍。不能加的更快的原因在下文会提到。

在目前的逐步尝试中，我发现了三组比较特殊的数据：

第一组：

天体 1：质量 5，初位置(100, 0, 0)，初速度(0, 11, 0)

天体 2: 质量 5, 初位置(-100, 0, 0), 初速度(0, -11, 0)

天体 3: 质量 1900, 初位置(0, 0, 0), 初速度(0, 0, 0)

这是中央大质量星体不动, 旁边两个小星体互不干扰地绕着大星球公转的模型。

第二组:

天体 1: 质量 2500, 初位置(0, 100, 0), 初速度(-10.2, 0, 0)

天体 2: 质量 2500, 初位置(-86.6, -50, 0), 初速度(5.1, -8.8, 0)

天体 3: 质量 2500, 初位置(86.6, -50, 0), 初速度(5.1, 8.8, 0)

这是三个天体绕公共质心公转的模型。

第三组:

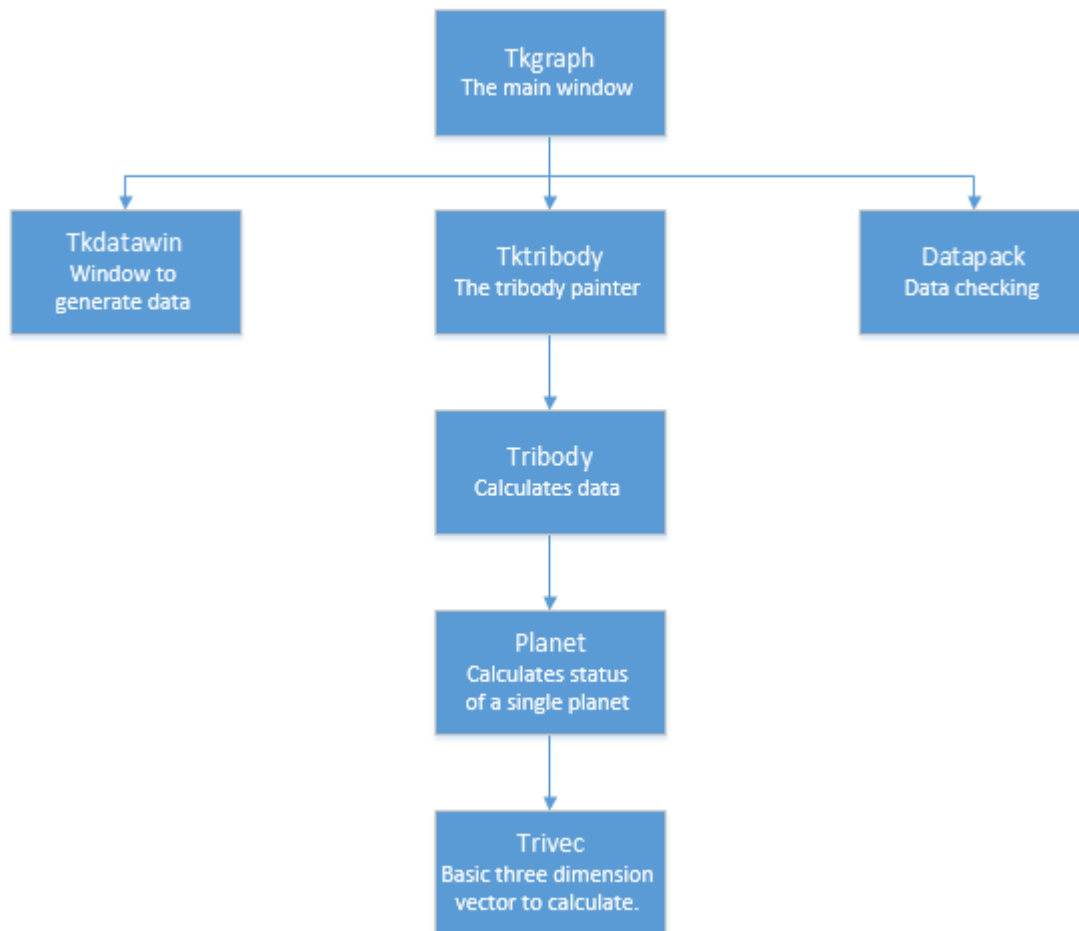
天体 1: 质量 5, 初位置(100, 10, 0), 初速度(0.4, 11, 0)

天体 2: 质量 5, 初位置(100, -10, 0), 初速度(0.4, 11, 0)

天体 3: 质量 1900, 初位置(0, 0, 0), 初速度(0, 0, 0)

这是日-地-月模型在地球和月亮质量相等时的特殊情形, 即质量较小的两个天体(地球和月亮)绕着它们的公共质心公转, 而同时此公共质心又绕着太阳公转。这一组数据不是很好, 因为出现了中间大质量体移动的状况, 这是由于中间的物体质量不够大。

本程序中, 定义的类非常多, 下图给出了各个类的一些基本关系:



其中，Tktribody 分支负责计算和绘制，Tkdatawin 类负责接收，而 Datapack 负责检查，分工明确。而另一种分工方式是由 Tkdatawin 分支负责接收和检查，即将 Datapack 类至于 Tkdatawin 中，以保证 Tkgraph 中只有绘图类，但这里没有采用，因为这种类的模式可以让检查数据独立于接收数据之外，从而避免不必要的接收数据的调用。如果需要多次检查数据，或需要即时检查数据，效率会比之前提到的方式要高一些。

各个类的方法繁多，在此不一一列举，代码中已经提供了必要的说明。**另外，特别注明，此份代码为我在书上及网上查阅多线程、tkinter 各部件用法以及类重载后独立完成的代码。**尽管上课老师并没有详细讲类的方法重载，但我认为这为我的类的编写提供了相当大的便利，尤其是对于三维向量的加法和数乘运算，省去了繁琐的步骤和大量的时间。

要说明的是，本程序依然存在少数 bug 未能解决，尤其是在绘图时使用了多线程运算后，新线程有时会崩溃，而我的从代码本身并没有看出问题。而每次睡眠的时间不能间隔太久，不然崩溃的几率会增加，这也是我为什么最后设置的时间只加快了 1.5 倍的原因。

最后祝老师新年快乐，身体健康。

2015 年 1 月 2 日