## 2022年暑期社会实践活动优秀调查报告推荐表

|  |  |  |  |
| --- | --- | --- | --- |
| 调查报告名称 | Python计算机模拟，验证物理开普勒第一、第二定律、以及第二宇宙速度 | | |
| 实践队名称 | 20xx级xxx班实践队1 | 作者 | xxx |
| 报告内容 | （1500 字以上，可另附纸） | | |
| 所在单位党组织意见 | （盖章）  年 月 日 | | |
| 市教育局团委意见 | （盖章）  年 月 日 | | |

**Python计算机模拟，验证物理开普勒第一、第二定律、以及第二宇宙速度**

xxx中学 xxx班

**负责人：**xxx **指导老师：**xxx

**分类：**学科研究类 **关键词：**Python 物理 引力 计算机模拟

**引言：**学习了物理《必修二》后，自己虽然明白了万有引力定律与第一宇宙速度、天体圆轨道之间的关系。但万有引力定律与第二宇宙速度，以及与开普勒第一、第二定律的关系在物理书中没有直接的推理过程。为此，编写了本程序, 研究万有引力定律与天体轨道行为之间的关系，以及验证开普勒第一、第二定律。

**正文：**

**一、实践目标**: 研究万有引力定律与天体轨道行为之间的关系; 验证开普勒第一、第二定律; 验证第二宇宙速度。。

**二、实践过程**:

**1.工具**: PyCharm 2022.2 编辑器、Python 3.7.8、pygame库。

**2.创建引力的计算机模拟**

首先, 创建CelestialBody类。打开PyCharm编辑器, 新建一个py文件, 输入以下代码:

G = 1 *# 引力常量***class** CelestialBody: *# 天体类* **def** \_\_init\_\_(self, mass, position, velocity,radius,color):  
 self.m=mass  
 self.x,self.y=position  
 self.v\_x, self.v\_y=velocity  
 self.radius=radius *# 在屏幕上的显示大小* self.color=color

根据公式 、 、 、 ，可编写以下代码:   
 **def** acceleration(self):  
 *# 计算行星的引力及加速度* a\_x=0; a\_y=0  
 **for** other **in** stars:  
 **if** other == self:**continue** diff\_x=other.x-self.x *# 另一个天体与自身的距离差* diff\_y=other.y-self.y  
 r = math.sqrt(diff\_x \*\* 2 + diff\_y \*\* 2) *# 距离* F = G \* self.m \* other.m / r\*\*2  
 *# 分解为水平、竖直方向* a\_x += F / self.m \* diff\_x / r  
 a\_y += F / self.m \* diff\_y / r  
 **return** a\_x, a\_y  
 **def** step\_once(self): *# 单次计算  
 # 计算行星位置* a\_x, a\_y = self.acceleration()  
 self.v\_x += a\_x \* t *# t为单次计算经过的时间，越小计算越精确* self.v\_y += a\_y \* t  
  
 self.x+= self.v\_x \* t  
 self.y+= self.v\_y \* t

注意, 其中的t为单次计算经过的时间，这个值越小, 计算机模拟的误差越小。这个值如果过大, 行星可能会飞出去。

图形界面:

**def** step(lst):  
 surf = pygame.Surface(size)  
 surf.fill(color=(200,200,200))  
 **for** body **in** lst:  
 body.step\_once()  
 pygame.draw.circle(surf,body.color,

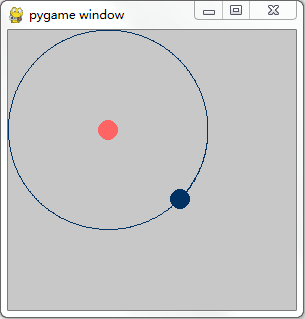
(int(body.x),int(body.y)),body.radius) *# 画出各个天体*

**for** i **in** range(len(path)-1): *# 画出轨迹* pygame.draw.line(surf,(0,50,100),path[i],path[i+1],1)  
 path.append((stars[1].x,stars[1].y))

**return** surf  
  
t = 0.1  
size = width,height = 400, 400  
screen = pygame.display.set\_mode(size,pygame.RESIZABLE)  
screen.fill((255, 255, 255))  
stars = [CelestialBody(10000,(100,100),(0,0),10,(255,100,100)),  
 CelestialBody(10,(200,100),(0,10),10,(0,50,100))]  
path = [] *# 行星经过的路径(轨迹)*

clock = pygame.time.Clock()  
running = **True  
while** running:  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 running=**False  
 elif** event.type == pygame.VIDEORESIZE:  
 size = width,height = event.size  
 screen = pygame.display.set\_mode(event.size, pygame.RESIZABLE)  
 screen.blit(step(stars),(0,0))  
 pygame.display.flip()  
 clock.tick(60)  
pygame.quit()

运行效果截图:



**3.** **验证开普勒第一定律**

程序验证行星的公转轨道是否为椭圆, 先获取椭圆的长、短轴， 再使用椭圆的方程计算。如果行星位置符合椭圆的方程, 即输出为一个定值, 则验证通过。

打开PyCharm编辑器, 新建一个py文件, 输入之前CelestialBody类的代码, 以及以下代码:

**def** get\_orbit\_shape(): *# 获取椭圆轨道的长、短轴长度, 及中点x坐标* max\_x=max(x\_lst)  
 min\_x=min(x\_lst)  
 **return** max\_x-min\_x,max(y\_lst)-min(y\_lst),(max\_x + min\_x)/2  
  
t = 0.08  
size = width,height = 400, 400  
screen = pygame.display.set\_mode(size,pygame.RESIZABLE)  
screen.fill((255, 255, 255))  
stars = [CelestialBody(10000,(100,100),(0,0),10,(255,100,100)),  
 CelestialBody(10,(200,100),(0,7),10,(0,50,100))]  
x\_lst=[];y\_lst=[]  
time = 0

path = [] *# 行星经过的路径(轨迹)*  
  
clock = pygame.time.Clock()  
running = **True  
while** running:  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 running=**False  
 elif** event.type == pygame.VIDEORESIZE:  
 size = width,height = event.size  
 screen = pygame.display.set\_mode(event.size, pygame.RESIZABLE)  
 screen.blit(step(stars),(0,0))  
 pygame.display.flip()  
 clock.tick(120)  
  
 time += t  
 **if** time>30:  
 w,h,centerx = get\_orbit\_shape()  
 print((stars[1].x-centerx)\*\*2 + \  
 (w/h \* (stars[1].y-stars[0].y))\*\*2) *# 椭圆的方程, 输出结果为一个定值*

x\_lst.append(stars[1].x)  
 y\_lst.append(stars[1].y)  
  
pygame.quit()

**4.** **验证开普勒第二定律**

原理: 程序先将行星轨道扫过的部分分割成一个个三角形, 再计算行星轨道扫过的面积, 并除以经过的时间。如果得到一个定值, 则验证通过。

打开PyCharm编辑器, 新建一个py文件, 输入之前CelestialBody类的代码, 以及以下代码:

**def** distance(p1,p2):  
 dx=p1[0]-p2[0];dy=p1[1]-p2[1]  
 **return** math.sqrt(dx\*\*2 + dy\*\*2)  
**def** calc\_square(t):  
 a=distance(t[0],t[1])  
 b=distance(t[1],t[2])  
 c=distance(t[0],t[2])  
 p = (a+b+c)/2  
 **return** math.sqrt(p\*(p-a)\*(p-b)\*(p-c))  
**def** step(lst):  
 **global** S\_total  
 surf = pygame.Surface(size)  
 surf.fill(color=(200,200,200))  
 previous = stars[1].x, stars[1].y *# 行星的上一个位置* **for** body **in** lst:  
 body.step\_once()  
 *# 开普勒第二定律* triangle = [center, previous, (stars[1].x, stars[1].y)]  
 area.append(triangle)  
 S\_total += calc\_square(triangle)  
 **for** triangle **in** area:  
 pygame.draw.polygon(surf, (0, 100, 255),  
 triangle, width=0)  
 previous = stars[1].x, stars[1].y  
  
 **for** body **in** lst:  
 pygame.draw.circle(surf, body.color, (int(body.x), int(body.y)), body.radius) *# 画出各个天体*  
 **return** surf  
  
t = 0.05  
size = width,height = 400, 400  
center = 100,100  
  
screen = pygame.display.set\_mode(size,pygame.RESIZABLE)  
screen.fill((255, 255, 255))  
stars = [CelestialBody(10000,center,(0,0),10,(255,100,100)),  
 CelestialBody(10,(200,100),(0,7),10,(0,50,100))]  
area = [] *# 行星扫过的区域*S\_total = 0 *# 行星扫过区域的总面积*time = 0  
  
clock = pygame.time.Clock()  
running = **True  
while** running:  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 running=**False  
 elif** event.type == pygame.VIDEORESIZE:  
 size = width,height = event.size  
 screen = pygame.display.set\_mode(event.size, pygame.RESIZABLE)  
 **elif** event.type == pygame.MOUSEBUTTONDOWN:  
 *# 清零* time = S\_total = 0  
 area = []  
  
 time += t  
 print(**"时间:"**,time,**"行星扫过面积:"**,S\_total,  
 **"行星扫过面积÷时间 ="**,S\_total/time)  
  
 screen.blit(step(stars),(0,0))  
 pygame.display.flip()  
 clock.tick(120)  
  
pygame.quit()

**5.** **验证第二宇宙速度**

首先通过公式 , 获取行星的第一宇宙速度, 然后乘以 得到第二宇宙速度(推导过程在课本上没有说明)。运行程序, 观察行星初速度略高于或低于第二宇宙速度时, 行星是否会飞出。

打开PyCharm编辑器, 新建一个py文件, 输入之前CelestialBody类的代码, 以及以下代码:

**class** CelestialBody: *# 天体类* **def** \_\_init\_\_(self, mass, position, velocity,radius,color):  
 # 此处省略  
 **def** acceleration(self):  
 # 此处省略  
 **def** step\_once(self): *# 单次计算* # 此处省略  
 **def** get\_v1(self, r):  
 *# 获取半径为r的轨道的环绕速度(第一宇宙速度)* g = G \* self.m / r\*\*2  
 **return** math.sqrt(g \* r)  
 **def** get\_distance(self,other):  
 diff\_x = other.x - self.x *# 另一个天体与自身的距离差* diff\_y = other.y - self.y  
 **return** math.sqrt(diff\_x\*\*2 + diff\_y\*\*2)  
  
**def** step(lst):  
 surf = pygame.Surface(size)  
 surf.fill(color=(200,200,200))  
 **for** body **in** lst:  
 body.step\_once()  
 pygame.draw.circle(surf,body.color,(int(body.x),int(body.y)),

body.radius) *# 画出各个天体*

**for** i **in** range(len(path)-1): *# 画出轨迹* pygame.draw.line(surf,(0,50,100),path[i],path[i+1],1)  
 path.append((stars[1].x,stars[1].y))

**return** surf  
  
t = 0.04  
size = width,height = 400, 400  
screen = pygame.display.set\_mode(size,pygame.RESIZABLE)  
screen.fill((255, 255, 255))  
  
center=CelestialBody(10000,(100,100),(0,0),10,(255,100,100))  
v1 = center.get\_v1(100)  
planet=CelestialBody(10,(200,100),(0,v1\*math.sqrt(2)),  
 10,(0,50,100))  
stars = [center,planet]

path = [] *# 行星经过的路径(轨迹)*  
  
clock = pygame.time.Clock()  
running = **True  
while** running:  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 running=**False  
 elif** event.type == pygame.VIDEORESIZE:  
 size = width,height = event.size  
 screen = pygame.display.set\_mode(event.size, pygame.RESIZABLE)  
 screen.blit(step(stars),(0,0))  
 pygame.display.flip()  
 print(center.get\_distance(planet))  
 *#clock.tick(120)*pygame.quit()

**三、感想与总结**

使用Python 模拟引力，不仅仅是编写代码，更是编程语言与物理学、数学的综合。最难能可贵的是，在我们开展研究的过程中学习到了许多在课本上所看不到的东西。

首先，这次研究，让我感受到计算机的独特用处。在现代社会，计算机作为一种必备的生产力工具，在各个领域都得到了广泛的应用，如计算机模拟。项目中，使用计算机模拟，可以很好地模拟真实世界的引力。但计算机模拟也有不足, 如在计算机算力有限的情况下，模拟会有误差。

其次，这次研究让我感受到数学、物理的独特魅力。实践经验告诉我们: 数学是物理的皇后, 研究物理离不开数学。从毫无头绪到一个个清晰的变量、公式的呈现，其中是我们思维逻辑的提高，也是对研究方法的一种积累。

同时，这次研究，还促进了我编程能力的提升。第一次写出来的程序有各种BUG(漏洞)，后来我通过调试程序，修复了大部分的bug，程序运行顺畅，并且代码有良好的可重用性，也就是同一个CelestialBody类可以用于多项研究。编程的本质是解决问题，这进一步促进了我将编程知识运用到解决问题的实践中去。

总而言之，这一次的研究在无形之中使我们收获到了许多。这是一次对课本上的知识的实际应用。将纸上谈兵的我们带向现实的探究之中。本次研究亦使我从中体会到物理、信息技术这两门学科的独特魅力，从而更深入地理解物理、信息技术学科。