

일반인을 위한
물리 코딩

3주차. 만유인력과 천체의 운동

$$\vec{F} = -\frac{Gm_1m_2}{|r|^2}\hat{r}$$

볼꽃놀이

$$\vec{F} = -\frac{Gm_1m_2}{|r^2|}\hat{r}$$

불꽃놀이 만들기

등가속운동 응용

흩어지면서 밑으로 떨어질 때,
포물체 운동을 하는 불꽃



여러 개의 객체를 한꺼번에 다루는 Python



불꽃놀이 만들기

등가속운동 응용

리스트
자료형 활용반복문 형태의
for 문 활용

출처 : 게티이미지뱅크



GlowScript 2.7 V

실제 불꽃 하나씩
위치 리스트로 만들

rList = list()

objList = list()

위치 좌표에 따라 하나씩
object도 리스트로 만들

#Creating Objects

ground = box(pos = vec(0,-5,0), size = vec(15,
0.01,15))



```
for i in range(0,100):
```

0에서 100까지

```
    rList.append(vec(0,-4,0))
```



for문을 통해 동일한 위치 값을 100번 반복하여 설정



```
for i in range(0,100):  
    rList.append(vec(0,-4,0))  
  
for r in rList: 위치리스트에서 하나씩 값을 가져옴  
    objList.append(sphere(pos = r, radius = 0.1,  
color = vec(random(), random(), random()),  
make_trail=True, retain = 30))
```




```
for i in range(0,100):  
    rList.append(vec(0,-4,0))
```

```
for r in rList:  
    objList.append(sphere(pos = r, radius = 0.1,  
color = vec(random(), random(), random()),  
make_trail=True, retain = 30))
```

각 불꽃 하나하나를 구 형태로 만들 것




```
for i in range(0,100):  
    rList.append(vec(0,-4,0))
```

```
for r in rList:  
    objList.append(sphere(pos = r, radius = 0.1,  
color = vec(random(), random(), random()),  
make_trail=True, retain = 30)) #Explain it
```

position은 r

구의 반지름은 0.1



```
for i in range(0,100):  
    rList.append(vec(0,-4,0))  
  
for r in rList:  
    objList.append(sphere(pos = r, radius = 0.1,  
color = vec(random(), random(), random()),  
make_trail=True, retain = 30)) #Explain it
```

임의의 컬러 값이 지정됨

 랜덤 함수는 0~1 사이에서 임의의 값을 리턴함

```
for i in range(0,100):  
    rList.append(vec(0,-4,0))  
  
for r in rList:  
    objList.append(sphere(pos = r, radius = 0.1,  
color = vec(random(), random(), random()),  
make_trail=True, retain = 30)) #Explair...
```

각각의 다른 색 공이 만들어짐



```
for i in range(0, N):  
    rList.append(rList[i])  
  
for r in rList:  
    objList.append(r)  
    color = vec([0.5, 0.5, 0.5])  
    make_trajectory(r, color, explain = 0.1,
```

각각의 다른 색 공이 만들어짐

RGB 3원색 (빨간색, 녹색, 파란색)



```
for i in range(0,100):  
    rList.append(vec(0,-4,0))
```

불꽃 자취 만들기

클립 보드에 복사되었습니다.

```
objList.append(sphere(pos = r, radius = 0.1,  
    color = vec(random(), random(), random()),  
    make_trail = True, retain = 30)) #Explain it
```

물체에 자취 그리기(꼬리그리기)

어느정도 지나면 사라지도록 함



```
#Initial Condition  
vi = vec(0,5.0,0)  
a = vec(0,-3,0)  
explosion = False
```

```
for obj in objList:  
    obj.v = vi
```

폭발 관련 변수



```
#Initial Condition  
vi = vec(0,5.0,0)  
a = vec(0,-3,0)  
explosion = False
```

```
for obj in objList:  
    obj.v = vi
```

100개의 물체에 대해 모두 설정

obj.v는 초기 속도 vi로 모두 설정



```
#time Setting  
t = 0  
dt = 0.01
```

시간이 1초 지나면 폭발

```
#Simulation Loop  
while t < 12:  
    rate(1/dt)  
    #Explosion  
    if t > 1 and explosion == False:  
        print("explosion!")
```

조건에 따라 실행되는 if 문



```
#scene.waitFor('click')
for obj in objList:
    obj.v = obj.v + vec(random()-0.5,random()
-0.5,random()-0.5)
    explosion = True
```



속도의 변화를 -0.5 에서 0.5로 줌



```
#scene.waitFor('click')  
for obj in objList:  
    obj.v = obj.v + vec(random()-0.5,random()  
-0.5,random()-0.5)  
    explosion = True
```



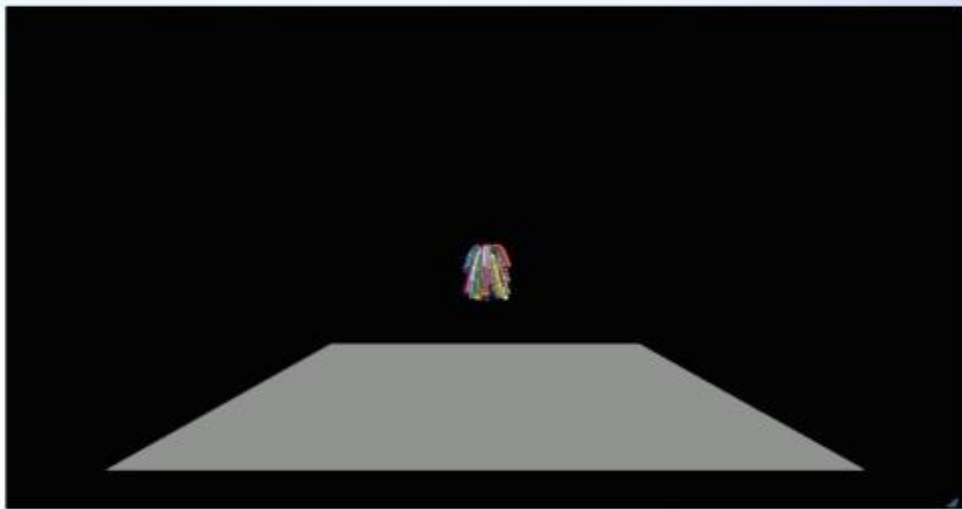
한 번만 수행: **explosion = True**



```
#Velocity and Position Update  
for obj in objList:  
    obj.v = obj.v + a*dt  
    obj.pos = obj.pos + obj.v*dt
```



물체를 하나씩 가져와 속도부터 순차적으로 업데이트





ground에서 다시 튀어 오르도록 구성



```
#Collision Handling
if obj.pos.y < ground.pos.y:
    obj.pos.y = ground.pos.y
    obj.v.y = -0.8*obj.v.y
    obj.color = vec(random(), random(),
random())
#Time Update
t = t + dt
```

땅과의 충돌 처리 부분

y좌표가 땅(ground)의
y좌표보다 작을 때



공이 튀어 오르게 하려면?

obj.pos.y가
ground.pos.y보다 작을 때

```
#Step 2
#Collision Handling
    if obj.pos.y < ground.pos.y:
        obj.pos.y = ground.pos.y
        obj.v.y = -0.8*obj.v.y
        obj.color = vec(random(), random(),
random())
#
#Time Update
t = t + dt
```



공이 튀어 오를 때 색깔 바꾸기

```
#Step 2
#Collision Handling
    if obj.pos.y < ground.pos.y:
        obj.pos.y = ground.pos.y
        obj.v.y = -0.8*obj.v.y
        obj.color = vec(random(), random(),
random())
#
#Time Update
t = t + dt
```

RGB값 변경

공이 튀어 오를 때 색깔 바꾸기





리스트와 for 문 학습 권장

쓰로 코딩하는 연습 필요



뉴턴의 제3법칙

작용 반작용의 법칙

물체에 힘이 작용하는 동시에,
항상 크기가 같고 방향이 반대인
힘이 작용함

자료출처

* 게티이미지뱅크
(<https://www.gettyimagesbank.com/>)
- 468710646



세종대학교
SEJONG UNIVERSITY