**Name: Session:**

**Programming I**

**Lab Exercise 11.6.2019**

**Projectile Class**

In this project, we will design a Projectile Class. The Projectile class will contain 4 class methods:

* \_\_init\_\_ - initializes data members
* update – recalculates new position of projectile
* getX – retrieves downrange distance
* getY – retrieves altitude of projectile

and 5 data members:

* xpos – downrange distance of projectile
* ypos – altitude of projectile
* xvel – horizontal velocity
* yvel – vertical velocity
* theta – launch angle in radians

This Projectile class will form the basis of a game that we are going to make at some future date.

1. On your desktop, in your MyPython folder create another folder called Projectile Project.
2. Create a new file in IDLE and save as *projectile.py* in the Projectile Project folder
3. Add the code for the Projectile class definition
4. Save the file as projectile.py in the Projectile Project folder
5. Create a new file in IDLE and save it as *projectileTest.py* in the Projectile Project folder
6. Add the code for the Projectile test program and save it as projectileTest.py in the Projectile Project folder.
7. Notice the line

from projectile import Projectile

which imports the Projectile class from *projectile.py*

1. Run the program projectileTest.py
2. When you have your program running satisfactorily, save your program as projectileTest.py
3. Now modify the program to report the maximum altitude the projectile attains in its flight.
4. Now modify projectileTest.py to use pylab to display a plot of your projectile’s path. You should also display the maximum altitude as well as the distance the projectile traveled down range.
5. **Print your source code after making this modification and attach to this sheet.**

**Class Definition of Projectile**

#projectile.py

#Author: nmessa

#Projectile motion project

#Projectile class definition file

from math import \*

class Projectile:

def \_\_init\_\_(self, angle, velocity, height):

self.xpos = 0.0

self.ypos = height

self.initial\_height = height

theta = pi \* angle / 180.0 #convert to radians

self.xvel = velocity \* cos(theta) # x-component of velocity

self.yvel = velocity \* sin(theta) # y-component of velocity

def update(self, time):

self.xpos = time \* self.xvel

self.ypos = self.initial\_height + self.yvel \* time - 4.9 \* time\*\*2

def getX(self):

return self.xpos

def getY(self):

return self.ypos

#end of class definition

Our test program will include two functions: main() and getInputs()

**Projectile Test Program**

## projectileTest.py

## author: nmessa

## Test file for the Projectile class

from projectile import Projectile

def getInputs():

a = float(input("Enter the launch angle (in degrees): "))

v = float(input("Enter the initial velocity (in meters/sec): "))

h = float(input("Enter the initial height (in meters): "))

dt = float(input("Enter the time interval between position calculations: "))

return a, v, h, dt

def main():

time = 0.0

#get input from user

angle, velocity, h0, deltaT = getInputs() #get inputs from user

cBall = Projectile(angle, velocity, h0) #construct Projectile

#prepare for first position calculation

time += deltaT

cBall.update(time)

#simulation loop

while cBall.getY() > 0:

cBall.update(time)

print ("Distance traveled: ", round(cBall.getX(), 3), "meters at time =", time, "seconds")

print ("Altitude: ", round(cBall.getY(), 3), "meters")

print()

time += deltaT

main()

**Questions:**

1. Why do we need to make a first position calculation prior to entering the simulation loop?
2. What is the purpose of deltaT?
3. Why do we need to convert our launch angle from degrees to radians?
4. In the line

yvel1 = self.yvel - 9.8 \* time

why is 9.8 negative?

1. What are the initial coordinates of the Projectile?
2. For a more accurate calculation of the maximum altitude should deltaT be larger of smaller? Why?

**The Mathematics of Projectile Motion**

**Projectile motion** is a form of [motion](http://en.wikipedia.org/wiki/Motion_(physics)) where a particle (called a projectile) is thrown obliquely near the earth's surface, **it moves along a curved path under the action of**[**gravity**](http://en.wikipedia.org/wiki/Gravity). The path followed by a projectile is called its [trajectory](http://en.wikipedia.org/wiki/Trajectory).

**The initial velocity**

If the projectile is launched with an initial velocity **v**0, then it can be written as

 \mathbf{v}_0 = v_{0x}\mathbf{i} + v_{0y}\mathbf{j}.

The components *v*0*x* and *v*0*y* can be found if the angle, α is known:

 v_{0x} = v_0\cos\alpha,

 v_{0y} = v_0\sin\alpha.

If the projectile's range, launch angle, and drop height are known, launch velocity can be found by

 V_0 = \sqrt{{R^2 g} \over {R \sin 2\theta + 2h \cos^2\theta}} .

## Kinematic quantities of projectile motion

In projectile motion, the horizontal motion and the vertical motion are independent of each other; that is, neither motion affects the other.

### Acceleration

Since there is no acceleration in the horizontal direction, the velocity in horizontal direction is constant which is equal to V0cos(α). The vertical motion of the projectile is the motion of a particle during its free fall. Here the acceleration is constant, equal to *g*. The components of the acceleration:

 a_x = 0 ,

 a_y = -g .

### Velocity

The horizontal component of the velocity remains unchanged throughout the motion. The vertical component of the velocity increases linearly, because the acceleration is constant. At any time *t*, the components of the velocity:

 v_x=v_0 \cos(\alpha) ,

 v_y=v_0 \sin(\alpha) - gt .

The magnitude of the velocity (under the Pythagorean theorem):

 v=\sqrt{v_x^2 + v_y^2 \ } .

### Displacement

Displacement and coordinates of parabolic throwing

At any time *t*, the projectile's horizontal and vertical displacement:

 x = v_0 t \cos(\alpha) ,

 y = v_0 t \sin(\alpha) - \frac{1}{2}gt^2 .

The magnitude of the displacement:

 \Delta r=\sqrt{x^2 + y^2 \ } .

## Parabolic trajectory

Consider the equations,

 x = v_0 t \cos(\alpha) ,

 y = v_0 t \sin(\alpha) - \frac{1}{2}gt^2 .

If we eliminate *t* between these two equations we will obtain the following:

y=\tan(\alpha) \cdot x-\frac{g}{2v^2_{0}\cos^2 \alpha} \cdot x^2,

This equation is the equation of. Since *g*, α, and *v*0 are constants, the above equation is of the form

y=ax+bx^2,

in which *a* and *b* are constants. This is the equation of a parabola, so the path is parabolic. The axis of the parabola is vertical.

## The maximum height of projectile

**Maximum height of projectile**

The highest height which the object will reach is known as the peak of the object's motion. The increase of the height will last, until *v*y = 0.

 0=v_0 \sin(\alpha) - gt_h ,

Time to reach the maximum height:

 t_h = {v_0 \sin(\alpha) \over g} .

From the vertical displacement the maximum height of projectile:

 h = v_0 t_h \sin(\alpha) - \frac{1}{2}gt_h^2,

so

 h = {v_0^2 \sin^2\alpha \over {2g}}  .

|  |  |  |
| --- | --- | --- |
| [http://upload.wikimedia.org/wikipedia/commons/thumb/4/4e/ParabolicWaterTrajectory.jpg/250px-ParabolicWaterTrajectory.jpg](http://en.wikipedia.org/wiki/File:ParabolicWaterTrajectory.jpg) | http://upload.wikimedia.org/wikipedia/commons/thumb/f/f2/Ferde_hajitas1.svg/250px-Ferde_hajitas1.svg.png | http://upload.wikimedia.org/wikipedia/commons/thumb/3/3e/Ferde_hajitas2.svg/250px-Ferde_hajitas2.svg.png |
| Parabolic water trajectory | Initial velocity of parabolic throwing | Components of initial velocity of parabolic throwing |

|  |  |
| --- | --- |
| http://upload.wikimedia.org/wikipedia/commons/thumb/0/00/Ferde_hajitas3.svg/250px-Ferde_hajitas3.svg.png | http://upload.wikimedia.org/wikipedia/commons/thumb/8/86/Ferde_hajitas4.svg/250px-Ferde_hajitas4.svg.png |
| Displacement and coordinates of parabolic throwing | Maximum height of projectile |