

CS 211 Lab Assignment

Instructor: Dan Fleck, Ricci Heishman

Lab: Projectile Position with static methods

Overview

In this lab you will use static Java methods to calculate the next position of a projectile given specific wind and gravity constraints.

The formula for projectile motion in the y direction is:

$$y = \frac{1}{2}at^2 + v_0t + y_0$$

a=acceleration due to gravity

t = time

v_0 is velocity in the y direction at time zero

y_0 is the initial height

We assume the wind is only contributing to the X direction, and thus the formula for the position in the X direction is:

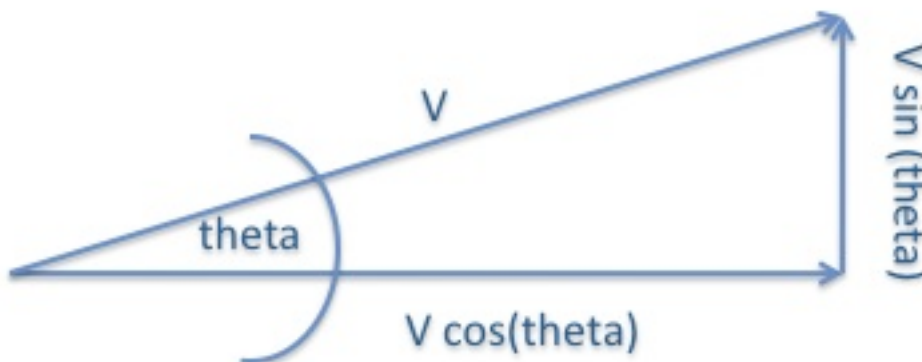
$$x = x_0 + v_0t + wt$$

v_0 is the velocity in the x direction at time zero

x_0 is the initial x position

w is the wind speed

Note: The velocity in the X and Y direction will be different and can be found using geometry from the angle and initial speed:



$$xVel = v * \cos(\theta)$$

$$yVel = v * \sin(\theta)$$

Assignment

You will write a Java class with a static methods with the following signatures:

```
/** This method should return the X position based on the inputs */  
public static double getXPosition(double initialX, double initialY, double gravity,  
double wind, double initialAngle, double initialSpeed, double time)
```

```
/** This method should return the Y position based on the inputs */  
public static double getYPosition(double initialX, double initialY, double gravity,  
double wind, double initialAngle, double initialSpeed, double time)
```

Additionally, you should include a main method to test the static methods for the following inputs:

InitialX	InitialY	Gravity	Wind	initialAngle	initialSpeed	Time
10	1	-9.8	0	30	10	3
20	20	-9.8	5	45	7	5
1	1	-9.8	-2	90	3.5	8

Sample Output

```
initialX:10.0  
initialY:1.0  
wind:0.0  
angle:30.0  
initialSpeed:10.0  
time:3.0
```

```
XPos: 35.98076211353316 YPos:-28.100000000000005
```

```
initialX:20.0  
initialY:20.0  
wind:5.0  
angle:45.0  
initialSpeed:7.0  
time:5.0
```

```
XPos: 69.74873734152916 YPos:-77.75126265847084
```

```
initialX:1.0  
initialY:1.0
```

wind:-2.0
angle:90.0
initialSpeed:3.5
time:8.0

XPos: -14.999999999999998 YPos:-284.6

What to turn in:

1. A Jar file containing your compiled code and source code
2. The output of your program.

Grading Rubric: This assignment is worth 10 points and will be graded based on the following rubric:

Area	Exemplary	Competent	Developing	Points
Class Header	All header components are present, with references and comments that accurately support the state of the file.	All header components are present, but references and comments are incomplete or nonspecific.	Header is missing or only partially present, and references and comments are vague or unmeaningful.	__ / 1
Coding Style	Code implementation utilizes appropriate white space, self-documentation techniques and non-obvious comments.	Code implementation exhibits minor alignment or spacing problems, some comments are missing or redundant.	Significant alignment and spacing problems, comments are generally missing or sporadic.	__ / 2
Method Signatures	The method signatures match the expected signatures exactly.	The method signatures contain the basic components, but are not exactly as expected.	The method signatures are very different from the assignment instructions.	__ / 2
Output X and Y positions	The positions determined by the code for any input are correct.	The position is correctly calculated for the sample inputs, but other inputs generate errors or inaccurate results.	The position calculations do not produce any accurate results.	__ / 3
Main method	The main method is present and correctly executes the required tests.	The main method is present, but only executes some of the required tests or performs inaccurate tests.	The main method is missing or does not work.	__ / 2