

Physics Assignment 4 – Week 3.6 Bouncing off an angled line

INTRODUCTION

With this programming assignment, 5 *bonus points* can be gained which count towards the final course grade (see 'Assessment' on Blackboard for grading details). For each assignment part, the points are indicated.

These bonus point can be gained by *signing off your solutions during the labs*.

In addition, these assignments prepare for the digital exam, which will contain similar questions. *More challenging assignment parts are preceded with a star*; these will not be part of the exam.

ASSIGNMENT 4.1 COLLISION DETECTION ON AN ANGLED LINE – 1 POINT

To implement this collision detection, you'll first need to be able to calculate a vector's normal and dot product.

First extend your Vec2 struct with the **Dot(...)** method which returns the dot product between the current and the given vector. Do **not** normalize any vectors within the Dot product. Use the provided sample 001_vector_projection to demonstrate that it works correctly.

Second extend your Vec2 struct with the **Normal()** method which returns a new Vec2 representing the **unit** normal for the current vector. Use the provided sample 002_line_collision_detection to demonstrate that it works correctly (a normal arrow will be shown on the line segment).

Last but not least, implement the basic collision detection mechanism for an angled line by completing/fixing the provided 002_line_collision_detection example. The ball should be green on the side of the line to which the normal is pointing and red when it is (partly) over the line. Study the slides to understand how you have to use your Normal() and Dot() methods to implement this.

ASSIGNMENT 4.2 COLLISION RESOLVING ON AN ANGLED LINE – 1 POINT

Extend 4.1 by making sure the ball cannot cross the line. Do this by resetting the position in the direction of the normal. Using the exact point of impact is allowed, but not required.

ASSIGNMENT 4.3 VELOCITY REFLECTION ON AN ANGLED LINE – 1 POINT

In order for this to work, you first have to implement a **Reflect (...)** method in your Vec2 struct. The Reflect method takes two parameters; a bounciness value (the COR, coefficient of reflection) and the vector in which to reflect the current vector. *Write a simple console test case to prove that your Reflect method works as expected.*

After verifying that your Reflect method works, use it to implement a moving ball that bounces off an angled line. First detect whether a collision occurred using your test from 4.1, then resolve the collision by putting the ball back onto the line using the mechanism from 4.2 and end with reflecting the velocity of the ball using the normal of the line it hit.

*** ASSIGNMENT 4.4 MULTIPLE LINES – 2 POINTS**

Extend 4.3 so that it works correctly with multiple lines and gravity. **Make sure** you can reset your test setup, slow down the framerate and draw the ball trajectory, same as in Assignment 3.