Physics Assignment 2 – Week 2.4 Trigonometry

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, at the latest during the fourth week* (week 3.6).

In addition, these assignments prepare for the final assignment, which should include most of these aspects.

All assignment parts are optional, but part 2.1 is strongly recommended, and serves as a basis for future assignments. More challenging assignment parts are preceded with a star.

Assignment 2.1 Vec2 implementation – 2 points

Extend the Vec2 class from Assignment 1 with the following methods:

- static Deg2Rad - converts the given degrees to radians
- static Rad2Deg - converts the given radians to degrees
- static GetUnitVectorDeg - returns a new vector pointing in the given direction in degrees
- static GetUnitVectorRad - returns a new vector pointing in the given direction in radians
- static RandomUnitVector - returns a new unit vector pointing in a random direction (all angles should have the same probability (density)).
- SetAngleDegrees - set vector angle to the given direction in degrees (length doesn't change)
- SetAngleRadians - set vector angle to the given direction in radians (length doesn't change)
- GetAngleRadians - gets the vector angle in radians
- GetAngleDegrees - gets the vector angle in degrees
- RotateDegrees - rotate the vector over the given angle in degrees
- RotateRadians - rotate the vector over the given angle in radians
- RotateAroundDegrees - rotate the vector around the given point over the given angle in degrees
- RotateAroundRadians - rotate the vector around the given point over the given angle in radians

*Reuse as much code as possible. Write a small test case for **every** operation that you implement, just like you did in Assignment 1.* (You need to show these test cases again when you sign off this assignment part.)

Assignment 2.2 Building a drivable tank

During this assignment you must use your completed Vec2 class and the information from the lecture to construct a drivable tank. Assets for the tank and a very basic code setup have been provided in the download for this assignment. There are assets for different tank bodies and barrels, bullets, backgrounds and engine sounds.

Requirements:

- For this assignment **you are not allowed to use the Move and TransformPoint methods** from the GXPEngine. All movements and transformations have to be calculated using your Vec2 class, but you are allowed to use a GameObject's x/y/rotation properties.
- The structure of your implementation must include a Tank class with a nested child object for the barrel (so MyGame has a Tank child which has a Barrel child).

- Rotation centers and placement of the barrel need to be reasonable (so this is the first thing that needs to be fixed in the given code setup).

2.2.1 Driving the tank – 1 point

Make it possible for the user to drive the tank with the following requirements:

- Forward/back accelerates/decelerates the tank in the direction the tank is facing, up to a maximum speed.
- If you are not accelerating/decelerating, the tank's speed slowly returns back to 0.
- Left/right rotates the tank, relative to the current speed of the tank, so a speed of 0 means you can't rotate.

2.2.2 Shooting bullets – 1 point

Make it possible for the user to shoot bullets with the tank with the following requirements:

- The tank barrel should always point at the mouse pointer (no easing), no matter how the tank is rotated.
- On mouse down a bullet is shot **from the nozzle** of the barrel in the direction of the barrel (a bullet image is available in the given assets, and a basic bullet class is given).

* 2.2.3 Barrel easing – 1 point

This is an update of assignment 2.2.2: instead of always pointing the barrel directly at the mouse pointer, the rotation of the barrel should slowly ease towards the mouse pointer, using the **shortest** angle. Make sure to test this behavior all around, with a rotated tank as well.