EL2310 Project 3 – C++
SCR15
Martin Isaksson, misakss@kth.se 10/27/2015

Project 3

Introduction

This project is supposed to make sure that the given program can use problems with multiple obstacles instead of just one circle as an obstacle, i.e. the main problem is to replace the already existing class SingleCircleWorld with classes/sub classes that can read in multiple obstacles, including both circles and rectangles, and check if the path is colliding with any obstacle and also make sure that MATLAB can print the problem out.

To do all this, the structure in Figure 1 is used, where the class World is given.

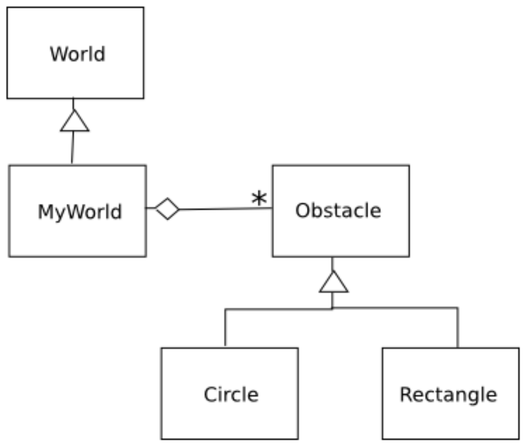

Figure 1 The used structure

MyWorld

MyWorld is a sub class of the class World that aggregates obstacles. World is a base class and an abstract class because there is a pure virtual function in the class.

In MyWorld the functions are readObstacle, collidesWith and writeMatlabDisplayCode. In readObstacle the program reads in the given text-file containing the type of obstacle and the coordinates. Depending on which type the obstacle is, the function creates the obstacle with help of the base class Obstacle by generating a list of obstacles (their coordinates) and with the sub classes Circle and Rectangle. The function returns true is reading the file was successful.

In the function collidesWith it checks the list of obstacles to see if the path collides with the obstacle, which is done in Circle and Rectangle. The function returns true if collision.

Also in the function writeMatlabDisplayCode it checks the list of obstacles to draw the obstacles in MATLAB. Also here it is done in Circle and Rectangle.

Obstacle

This is an abstract base class which is used to make it easier to have more different obstacles. If there is a new problem with octagon as obstacles, then only a sub class called Octagon needs to be created. Because the class Obstacle is written so that the children of it can use its functions and it is the same functions for all of its children.

Circle

In this sub class of Obstacle, the collision check is being done and also it is here that MATLAB gets the code for printing out the circles.

If the distance between the center of the circle and the coordinates of the path is equal to or smaller than the radius, there is a collision and the function collidesWith returns true.

Rectangle

In this sub class of Obstacle, the collision check is being done and MATLAB gets the code to print the Rectangles.

To print out a rectangle with a rotation, the following equations is being used:

$$x' = x\cos(\theta) - y\sin(\theta)$$
$$y' = y\cos(\theta) + x\sin(\theta)$$

, where x' and y' are the new corner coordinate.

To check if the coordinates of the path collides with any rectangle, a method using the areas of the triangles created by two corners and the point/coordinates of the path (See Figure 2), is being used.

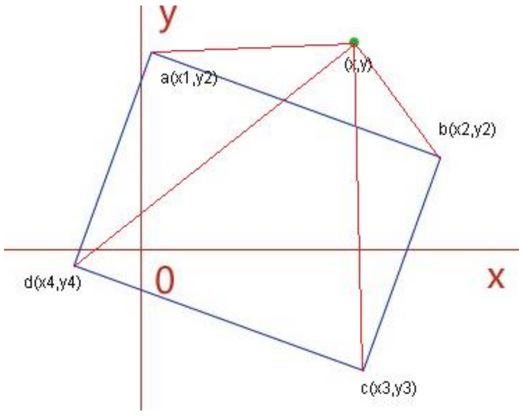

Figure 2 Triangles between the corners and the path-point

If then the area of one of the 4 triangles is not greater than the area of the rectangle, then the point is inside the rectangle and there is a collision and collidesWith will return true.