Assignment 1

moving-circles

Affiliation: Mathematics, CS

student ID: 2017313260

Name: Lee Jae Min

This code is based on gl-04-circle.

1. Data Structure

1.1. circle.h

First, structure of circle_t, I added elements velocity, vector cir_list and some function.

It has update, check_collision, check_collision_circle function.

velocity was given random value at create_circles function. vector cir_list is saves information about circles which has not been updated about speed after the collision. It prevents speed changes to superposition, So prevent the law of conservation of momentum be hold.

The update function is called from the render() function, which updates the information for each circle.

The check_collision function is check about collision with circle and walls, then change velocity.

The check_collision_circle function is check about collision between circles then change velocity.

1.2 main.cpp

Change NUM_TESS to 30, for object looks circle.

Add number of circle(CIR_NUM), and it can be changed, set minimum(CIR_MIN) and maximum(CIR_MAX).

Initial window size set 1280*720.

It has current simulation parameter t, I add passed_t. which is time of last call render() function, before call render() at time t. This parameter can make speed proportional to time.

2. Algorithm

2.1 create_circles

Make circles which has random value of center, radius, velocity. So import random, use random_device. Set rmin, rmax inverse proportional to number of circle.

I want to initial circles are don't collision. so, before emplace back, check distance between center and sum of radius, if it has collision, remake circle.

2.2 check_collision

Think about the place where the circle arrives after time. So calculate (t - passed_t)*velocity, add this and reverse the speed if the circle touches or crosses the wall.

2.3 check_collision_circle

let mass is proportional to size of circle. So, it will be radius^2 * pi.

Check collision, don't just compare with sum of radius and distance between centers.

For collision, two circles need to be close. So, Consider the distance approached by speed.

Projection each velocity to the line connecting the two centers.

So, compare with distance + projection speed and sum of radius.

If it has collision, take this method (by wiki.)

$$\mathbf{v}_1' = \mathbf{v}_1 - \frac{2m_2}{m_1+m_2}\,\frac{\langle \mathbf{v}_1-\mathbf{v}_2,\,\mathbf{x}_1-\mathbf{x}_2\rangle}{\|\mathbf{x}_1-\mathbf{x}_2\|^2}\,(\mathbf{x}_1-\mathbf{x}_2),$$
$$\mathbf{v}_2' = \mathbf{v}_2 - \frac{2m_1}{m_1+m_2}\,\frac{\langle \mathbf{v}_2-\mathbf{v}_1,\,\mathbf{x}_2-\mathbf{x}_1\rangle}{\|\mathbf{x}_2-\mathbf{x}_1\|^2}\,(\mathbf{x}_2-\mathbf{x}_1)$$

2.4 update

Change center value is calculated by velocity * time.

Circle explained by translate matrix and scale matrix.

2.5 etc

Press keyboard, each have function.

3. Discussions

To prevent collision, additional condition is only speed projection, it has a collision when it is not touched or overlapped. A more accurate collision detective system is needed.