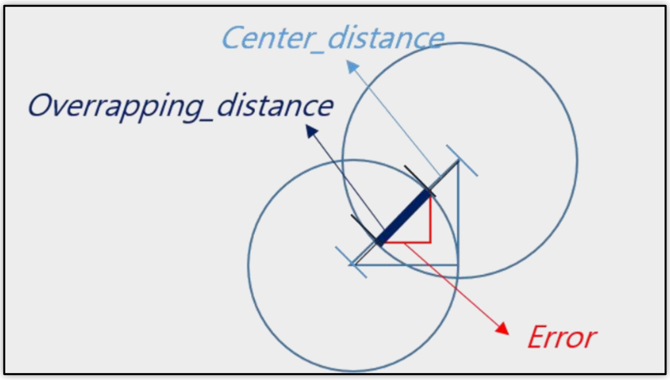
Moving Circles

2013314587
독어독문학과
AnHyeonJang, 안현장

1. Objective
    1) Implementing as if the collision occurs between the circles and walls.
2. Data Structure
    1) *Circle_t* contains the information that is center, radius, model matrix, and update. Additionally it contains velocity, mass and index.
    2) *Wall_t* defines the virtual limitation of the window. It contains the coordinate information, left, down, right, up, update. And its information should be updated following window size.
    3) *Circles* is the list that contains *Circle_t* type circles.
3. Algorithm
    1) To create the initialized 20 circles use 3 loop brutal-force algorithm.
    2) In *reshape()* the instance of the type wall_t is updated following window size.
    3) In *render()* function
        i. the position of a circle is updated,
        ii. check the collision and update the velocity
        iii. and render



    4) For checking the collision, use total 2 loop.
        i. To calculate the collision use Elastic Collision formula, it is in the function *calculate()*
        ii. *obj_collsion()* checks the collision between the two circles, and calculates new velocity as the mentioned Formula.
            a. Add the value for the circles to fix the overlapping error, and the correction value is calculated as follows.
            b. center c_1, c_2 of the circles and its radius r_1, r_2

$$Error = \frac{overlapping\_dis}{centers\_dis} * (centers\_diff) = ((r_1 + r_2) - \|c_1 - c_2\|)/\|c_1 - c_2\| * (c_1 - \ c_2)$$

        iii. *wall_collision()* changes the direction when a circle arrives at the end of the window. It also has the correction value as follows.

$$Error = c_1 \pm (r_1 - c_1 + wall)$$

4. Discussion
    1) The sticking problem that the overlapping circles move together occurs when the circles invade the mutual area so that the velocity cannot be update appropriately. To avoid the circles overlapping, it is the simple solution

to give the correction value. However, because of the value to avoid the problem, there is unexpected occasion that the circles seem like change their velocity shortly before the collision. Although this is not big problem, become also new error. This can be reduced by minimizing the correction value. However, if the correction value is smaller than the actually calculated, the overlapping can occur whenever.

2) But don’t need to consider controlling the correction values if the velocity of the circles is so appropriate that people are hard to see the error that the circles change their velocity shortly before the collision. To do this, the program should have restriction of the velocity, by giving the maximum and minimum speed scale.