Computer Garaphics

A1: Moving Circles

Affiliation: BioMechatronics

Student ID: 2015313483

Name: Kim Hyesung

Data Structure / Algorithms

In professor's solution, I modified circle.h file and main.cpp. In circle structure, I added new variable called moveDir in vec2 and id in uint. moveDir is for storing each circle's velocity and id is to check whether two circles are unique in collision checking.

```
struct circle_t
{
    vec2    center = vec2(0);
    float   radius = 1.0f;
    float   theta = 0.0f;
    vec4    color;
    vec2    moveDir;
    uint    id;
    mat4    model_matrix;
    // public functions
    void    update(float t);
};
```

In create_circles function, I modified to get float n variable to get number of circles to make. And I used random function to make instance to have random attributes.

In circle structure's update function, I got float t variable to get time difference between frames prior to current frame. And I moved center of circle by multiplying circle's moveDir and t. I checked whether the destination of move is in area of window.

In main.cpp, I changed keyboard function to get + and – key and change float Num_Circles variable when they are pressed. I set initial number of circles to be 100. I made update_DeltaTime function to get time gap between frames and called it in update function.

```
for (auto& c : circles)
{
    // per-circle update
    c.update(dt);
    for (auto& target : circles)
    {
        if (c.id != target.id)
        {
            if (checkOverlap(c.center.x, c.center.y, c.radius, target.center.x, target.center.y, target.radius))
            {
                // collision 있음
                float dist_between_centers = sqrtf((c.center.x - target.center.x) * (c.center.x - target.center.x) + (c.center.y - target.center.y) * (c.center.y - target.center.y));
                float fOverlap = 0.5f * (dist_between_centers - c.radius - target.radius);
                //calculating normal vector
                float normal_vecx = (target.center.x - c.center.x) / dist_between_centers;
                float normal_vecy = (target.center.y - c.center.y) / dist_between_centers;

                //calculating tangent vector
                float tangent_x = -normal_vecy;
                float tangent_y = normal_vecx;

                //dot product between tangent vector and velocity
                float dot_tan_c = c.moveDir.x * tangent_x + c.moveDir.y * tangent_y;
                float dot_tan_target = target.moveDir.x * tangent_x + target.moveDir.y * tangent_y;

                //dot product between normal vector and velocity
                float dot_norm_c = c.moveDir.x * normal_vecx + c.moveDir.y * normal_vecy;
                float dot_norm_target = target.moveDir.x * normal_vecx + target.moveDir.y * normal_vecy;

                c.moveDir.x = tangent_x * dot_tan_c + normal_vecx * dot_norm_target;
                c.moveDir.y = tangent_y * dot_tan_target + normal_vecy * dot_norm_target;
                target.moveDir.x = tangent_x * dot_tan_c + normal_vecx * dot_norm_c;
                target.moveDir.y = tangent_y * dot_tan_target + normal_vecy * dot_norm_c;
                c.center.x -= fOverlap * (c.center.x - target.center.x) / dist_between_centers;
                c.center.y -= fOverlap * (c.center.y - target.center.y) / dist_between_centers;
                target.center.x += fOverlap * (c.center.x - target.center.x) / dist_between_centers;
                target.center.y += fOverlap * (c.center.y - target.center.y) / dist_between_centers;
            }
        }
    }
```

In render function, I made checked collision by looping every pair of circles, comparing sum of two circle's radius and distance between their centers. If sum of radius is bigger, I considered they are colliding and calculated normal vector, tangent vector, dot product between tangent vector and velocity, normal vector and velocity. Using those figures, the function put new velocity of each circles and arrange center to avoid overlapping.

Discussions

There can be some afterimages of circle if the speed is too high or if there are too many circles to calculate. In my desktop about 250 circles caused afterimages.