1 Circle–Circle Collision

Definitions:

Position circle A: $\vec{p}_A$
Position circle B: $\vec{p}_B$
$\vec{p}_{A0} = \vec{p}_A(t = 0)$
$\vec{p}_{A1} = \vec{p}_A(t = 1)$
$\vec{p}_{B0} = \vec{p}_B(t = 0)$
$\vec{p}_{B1} = \vec{p}_B(t = 1)$
Radius circle 1: r_1
Radius circle 2: r_2

For Circle–Circle collsion detection two cases are need to be evaluated:

1. Circles do not cover a common area
2. One circle is inside the other

For the calculation, this only changes the way the radii are handled. For case 1, the following definition is used

$$r = r_1 + r_2 \tag{1}$$

Case 2 implies

$$r = |r_1 - r_2| \tag{2}$$

With t being the time of collision factor ($t \in [0, 1]]$) the following equation must be solved:

$$\|\vec{p}_{A0} + (\vec{p}_{A1} - \vec{p}_{A0}) \cdot t - (\vec{p}_{B0} + (\vec{p}_{B1} - \vec{p}_{B0}) \cdot t)\| = r \tag{3}$$

$$\|\vec{p}_{A0} + \vec{p}_{\Delta A} \cdot t - (\vec{p}_{B0} + \vec{p}_{\Delta B} \cdot t)\| = r \tag{4}$$

$$\|(\vec{p}_{A0} - \vec{p}_{B0}) + (\vec{p}_{\Delta A} - \vec{p}_{\Delta B}) \cdot t\| = r \tag{5}$$

$$[(\vec{p}_{A0} - \vec{p}_{B0}) + (\vec{p}_{\Delta A} - \vec{p}_{\Delta B}) \cdot t]^2 = r^2 \tag{6}$$

$$[U + V \cdot t]^2 = r^2 \tag{7}$$

$$U^2 + 2UVt + V^2t^2 = r^2 \tag{8}$$

$$t^2 + 2\frac{UV}{V^2}t + \frac{U^2 - r^2}{V^2} = 0 \tag{9}$$

$$t_{1/2} = -\frac{p}{2} \quad \stackrel{+}{-} \quad \sqrt{\left(\frac{p}{2}\right)^2 - q} \tag{10}$$

with

$$\frac{p}{2} = \frac{UV}{V^2} \tag{11}$$

and

$$q = \frac{U^2 - r^2}{V^2} \tag{12}$$

If the square root is complex or t is negative, there is no collsion. If the square root is zero, there is one collision. Otherwise, there are two solutions for t, first collision appears for the smaller t.

2 Polygon–Polygon Collision

For polygon–polygon collision every vertex v_n of polygon B is tested for collision with every line between vertex v_n and v_{n+1} of polygon A.

Definitions:

Point on line between vertex v_n and v_{n+1} of polygon A for each dimension: p_A
Vertex v_n of polygon B for each dimension: p_B
$p_{A0} = p_A(t = 0)$
$p_{A1} = p_A(t = 1)$
$p_{B0} = p_B(t = 0)$
$p_{B1} = p_B(t = 1)$

2.1 Version 1 – Point–Line

The line between vertices of polygon A is defined by the accordant vertices $p_{A00} = p_{A_0}(t = 0)$ and $p_{A10} = p_{A_1}(t = 0)$

$$p_{A0} = p_{A00} + \alpha(p_{A10} - p_{A00}) \tag{13}$$

$$p_{A1} = p_{A01} + \alpha(p_{A11} - p_{A01}) \tag{14}$$

$$p_{A0} + (p_{A1} - p_{A0}) \cdot t = p_{B0} + (p_{B1} - p_{B0}) \cdot t \tag{15}$$

$$\frac{p_{A0} - p_{B0}}{(p_{B1} - p_{B0}) - (p_{A1} - p_{A0})} = t \tag{16}$$

$$\frac{p_{A00} + \alpha(p_{A10} - p_{A00}) - p_{B0}}{(p_{B1} - p_{B0}) - [(p_{A01} + \alpha(p_{A11} - p_{A01})) - (p_{A00} + \alpha(p_{A10} - p_{A00}))]} = t \tag{17}$$

$$\frac{p_{A00} - p_{B0} + \alpha(p_{A10} - p_{A00})}{(p_{B1} - p_{B0}) - (p_{A01} - p_{A00}) + \alpha[(p_{A10} - p_{A00}) - (p_{A11} - p_{A01})]} = t \tag{18}$$

Substitution of constants:

$$f(\alpha) = \frac{a + \alpha b}{c + \alpha d} = t \tag{19}$$

This is true for coordinates x, y. For both dimensions, time t must be equal. Hence, α can be calculated by solving the following equation:

$$\frac{a_x + \alpha b_x}{c_x + \alpha d_x} = \frac{a_y + \alpha b_y}{c_y + \alpha d_y} \tag{20}$$

After some rearranging we get

$$\alpha^2(b_x d_y - b_y d_x) + \alpha(a_x d_y + b_x c_y - a_y d_x - b_y c_x) + (a_x c_y - a_y c_x) = 0 \tag{21}$$

which is a simple quadratic equation. All valid $\alpha \in [0, 1]$ result in a time t, that is also only valid for $t \in [0, 1]$.

2.2 Version 1 – Point–Line/Circle–Line

Testing the end points

The first part is to test the end points of the line for collsion when dealing with a circle.

$$\|(\vec{p}_{A0} + t \cdot (\vec{p}_{A1} - \vec{p}_{A0})) - (\vec{p}_{B0} + t \cdot (\vec{p}_{B1} - \vec{p}_{B0}))\| = r \tag{22}$$

$$\|(\vec{p}_{A0} - \vec{p}_{B0} + t \cdot (\vec{p}_{A1} - \vec{p}_{A0} - \vec{p}_{B1} + \vec{p}_{B0})\| = r \tag{23}$$

$$\|a + t \cdot b\| = r \tag{24}$$

$$b^2t^2 + 2abt + a^2 - r^2 = 0 \tag{25}$$

This is a simple quadratic equation which must be solved for t.

Testing the lines between end points

In this second part, collision is handled by the distance d_i of a point to the line between two points. For point–line collision distance $d_i = 0$ is sufficient. In case of a circle, $d_i = r$ with r being the radius of the circle.
The distance to the line is calculated similar to using a plane in Hessian Normal Form in 3D.
One approximation is very important: Since a simple interpolation is used, lines do not neccessarily keep their length. Hence, the norm of a vector between to points might change for this model, while in reality, it would stay the same. For easier calculation, the unchanged length is used for calculation, which slightly disagrees with the model.

$$\|\vec{p}_{A10} - \vec{p}_{A00}\| \approx \|(\vec{p}_{A10} + t \cdot (\vec{p}_{A11} - \vec{p}_{A10})) - (\vec{p}_{A00} + t \cdot (\vec{p}_{A01} - \vec{p}_{A00}))\| \tag{26}$$

With this, the distance is calculated as follows:

$$\frac{(\vec{p}_{A10} + t \cdot (\vec{p}_{A11} - \vec{p}_{A10})) - (\vec{p}_{A00} + t \cdot (\vec{p}_{A01} - \vec{p}_{A00}))}{\|\vec{p}_{A10} - \vec{p}_{A00}\|} \quad \times \tag{27}$$

$$\frac{(\vec{p}_{B0} + t \cdot (\vec{p}_{B1} - \vec{p}_{B0})) - (\vec{p}_{A00} + t \cdot (\vec{p}_{A01} - \vec{p}_{A00}))}{\|\vec{p}_{A10} - \vec{p}_{A00}\|} \quad = \quad d_i \tag{28}$$

$$[(\vec{p}_{A10} - \vec{p}_{A00}) + t \cdot (\vec{p}_{A11} - \vec{p}_{A10} - \vec{p}_{A01} + \vec{p}_{A00})] \quad \times \tag{29}$$

$$[(\vec{p}_{B0} - \vec{p}_{A00}) + t \cdot (\vec{p}_{B1} - \vec{p}_{B0} - \vec{p}_{A01} + \vec{p}_{A00})] \quad = \quad d_i \, \|\vec{p}_{A10} - \vec{p}_{A00}\| \tag{30}$$

$$(a + t \cdot b) \times (c + t \cdot d) = d_i e \tag{31}$$

$$(b \times d)t^2 + (b \times c + a \times d)t + (a \times c - d_i e) = 0 \tag{32}$$

This is a simple quadratic equation which must be solved for t. There is a special case: If the line does not move in time, then $b = 0$. Hence, the resulting equation is simplified to:

$$(a \times d)t + (a \times c - d_i e) = 0 \tag{33}$$

$$t = \frac{d_i e - a \times c}{a \times d} \tag{34}$$

In both cases, the order of operands of the cross product must also be changed, so there are several solutions for time t to test for the smallest value. If a time is found, it has to be validated. By using the time and the dot product, it is tested, if the point of collsion lies within the line.