

# Collision Detection in Math

**Note 1.** Before we start, lets make some simplifying assumptions:

- i. We are only dealing with circles
- ii. Circles have unit mass
- iii. Collisions are elastic
- iv. We don't handle  $n$ -body collisions

Assumptions ii., iii. and iv. reduce the collision equations to that of swapping velocities:

$$\begin{aligned} v_{a_f} &= \frac{v_{a_i}(m_a - m_b) + 2m_b v_{b_i}}{m_a + m_b} = \frac{v_{a_i}(1 - 1) + 2 \cdot 1 \cdot v_{b_i}}{1 + 1} = \frac{v_{a_i} \cdot 0 + 2v_{b_i}}{2} = v_{b_i} \\ v_{b_f} &= \frac{v_{b_i}(m_b - m_a) + 2m_a v_{a_i}}{m_a + m_b} = \frac{v_{b_i}(1 - 1) + 2 \cdot 1 \cdot v_{a_i}}{1 + 1} = \frac{v_{b_i} \cdot 0 + 2v_{a_i}}{2} = v_{a_i} \end{aligned}$$

First we define our data model.

**Let**  $\mathbf{Circles} = \{(p, v, r) | p, v \in \mathbb{R}^2 \text{ and } r \in \mathbb{R}^+\}$

The set **Circles** is all circles on the 2 dimensional plane with position  $p$ , velocity  $v$  and radius  $r$ .

We now recursively construct the sequence of worlds

**let**  $\mathbf{Worlds} = \{\mathbf{world}_n\}_0^\infty$  where  $\mathbf{world}_n \in \mathcal{P}(\mathbf{Circles})$

$$\mathbf{world}_n = \begin{cases} \exists \mathbf{world}_0 \in \mathcal{P}(\mathbf{Circles}) & \text{if } n = 0 \\ \text{stepWorld}(\mathbf{world}_{n-1}) & \text{if } n > 0 \end{cases} \quad (1)$$

**Note 2.** This sequence  $\mathbf{Worlds} = \{\mathbf{world}_0, \mathbf{world}_1, \dots\}$  can be played back by transforming it with a render function that maps each world to an image ie.  $r: \mathcal{P}(\mathbf{Circles}) \rightarrow M_{640 \times 480}$

As we can see from the definition of **Worlds**, every world  $\mathbf{world}_i$  is fully dependent on the previous world  $\mathbf{world}_{i-1}$  except for the very first world  $\mathbf{world}_0$ . Thus, once the first world is chosen, all future worlds are determined.

Alright, lets look at the **stepWorld** function

**let**  $\text{stepWorld}: \mathcal{P}(\mathbf{Circles}) \rightarrow \mathcal{P}(\mathbf{Circles})$

$$\text{stepWorld}(\mathbf{world}_n) = \{c_{n+1} \in \mathbf{Circles} | c_{n+1} = \text{stepCircle}(\text{circle}, \mathbf{world}_n) \text{ for } \text{circle} \in \mathbf{world}_n\} \quad (2)$$

We will define **stepCircle** in a bit, first we need to some definitions.

**Definition 3.** We define the relation **intersect** over *Circles*

Let  $a, b \in \mathbf{Circles}$  where  $a = (p_a, v_a, r_a)$  and  $b = (p_b, v_b, r_b)$

$$a \text{ intersect } b \iff |p_b - p_a| \leq r_a + r_b \quad (3)$$

where  $|p_b - p_a|$  is the euclidean norm on  $\mathbb{R}^2$

Now we define the function **colliding** which gives us all circles in a world that intersect a circle

**let colliding: Circles  $\times$   $\mathcal{P}$ (Circles)  $\rightarrow$   $\mathcal{P}$ (Circles)**

$$\text{colliding}(\text{circle}, \text{world}) = \{x \in \text{world} \mid x \neq \text{circle} \text{ and } (\text{circle}, x) \in \text{intersect}\} \quad (4)$$

Great, now lets define **stepCircle: Circles  $\times$   $\mathcal{P}$ (Circles)  $\rightarrow$  Circle**

$$\text{stepCircle}(\text{circle}, \text{world}) = \begin{cases} \text{physics}(\text{collide}(c, \text{world})) & \text{if } \text{colliding}(c, \text{world}) \neq \emptyset \\ \text{physics}(c) & \text{if } \text{colliding}(c, \text{world}) = \emptyset \end{cases} \quad (5)$$

where **physics** is the function that takes a circle and integrates the position forward one time step.

**let physics: Circles  $\rightarrow$  Circles**

**let  $\Delta t = 1/60$**

$$\text{physics}((p, v, r)) = (p + \Delta t v, v, r) \quad (6)$$

**let collide: Circles  $\times$   $\mathcal{P}$ (C)  $\rightarrow$  Circles**

$$\text{collide}(\text{circle}, \text{world}) = \text{bounce}(\text{circle}, \text{chooseColliding}(\text{circle}, \text{world})) \quad (7)$$

**let bounce: Collide  $\times$  Collide  $\rightarrow$  Collide** be the function that updates the first circle when it collides with the second circle

$$\text{bounce}(((p_a, v_a, r_a), (p_b, v_b, r_b)))) = (p_a, v_b, r_a) \quad (8)$$

Lets look at **chooseColliding: C  $\times$   $\mathcal{P}$ (C)  $\rightarrow$  C**. This function is a bit of a hack, its purpose is to keep the simulation deterministic when multiple collisions happen at the same time. The way it does this is it finds the ‘minimum’ circle out of the set of circles that intersect with the circle we care about.

Ok lets do it. First a definition

**Definition 4.** We define the relation  $(\leq)$  on  $\mathbb{R}^2$

let  $a, b, c, d \in \mathbb{R}$

$$(a, b) \leq (c, d) \iff (a \leq c) \wedge (b \leq d) \quad (9)$$

**let m: Circles  $\times$  Circles  $\rightarrow$  Circles** be a function that chooses the minimum circle of two given circles

$$m(((p_a, v_a, r_a), (p_b, v_b, r_b)))) = \begin{cases} (p_a, v_a, r_a) & \text{if } (p_a \leq p_b) \wedge (v_a \leq v_b) \wedge (r_a \leq r_b) \\ (p_b, v_b, r_b) & \text{if } \neg((p_a \leq p_b) \wedge (v_a \leq v_b) \wedge (r_a \leq r_b)) \end{cases} \quad (10)$$

$$\text{chooseColliding}(c, w) = x \text{ for } x \in \text{colliding}(c, w) \text{ s.t. } \forall a \in \text{colliding}(c, w), m(x, a) = x \quad (11)$$

**Theorem 5.**  $\forall c \in \text{Circles}, \forall w \in \mathcal{P}(\text{Circles}), \exists! x \in \text{Circles s.t. } x = \text{chooseColliding}(c, w)$

*not going to prove this here, cause i don't want to have to prove it for the python version. But just have to show that  $(p_a \leq p_b) \wedge (v_a \leq v_b) \wedge (r_a \leq r_b)$  defines a total ordering on Circles.*