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

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

We now recursively construct the sequence of worlds

let $\mathbf{Worlds} = \{\mathbf{world}_{\boldsymbol{n}}\}_{\mathbf{0}}^{\infty}$ where $\mathbf{world}_{\boldsymbol{n}} \in \boldsymbol{\mathcal{P}}(\mathbf{Circles})$

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

Note 2. This sequence $\mathbf{Worlds} = \{\mathbf{world_0}, \mathbf{world_1}, \ldots\}$ can be played back by tranforming it with a render function that maps each world to an image ie. $\boldsymbol{r} : \boldsymbol{\mathcal{P}}(\mathbf{Circles}) \longrightarrow \boldsymbol{M}_{\mathbf{640 \times 480}}$

As we can see from the definition of **Worlds**, every world $\mathbf{world}_{\boldsymbol{i}}$ is fully dependent on the previous world $\mathbf{world}_{\boldsymbol{i-1}}$ 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 stepWorld: $\boldsymbol{\mathcal{P}}(\mathbf{Circles}) \longrightarrow \boldsymbol{\mathcal{P}}(\mathbf{Circles})$

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

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

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

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

$$a \,\text{intersect}\, b \iff |p_b - p_a| \leqslant r_a + r_b \tag{3}$$

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 × $\boldsymbol{\mathcal{P}}$(Circles) → $\boldsymbol{\mathcal{P}}$(Circles)

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

Great, now lets define **stepCircle: Circles × $\boldsymbol{\mathcal{P}}$(Circles) → Circle**

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

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

let physics: Circles → Circles

let $\boldsymbol{\Delta t = 1/60}$

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

let collide: Circles × $\boldsymbol{\mathcal{P}(C)}$ → Circles

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

let bounce: Collide × Collide → Collide be the function that updates the first circle when it collides with the second circle

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

Lets look at **chooseColliding: $\boldsymbol{C \times \mathcal{P}(C) \to C}$**. 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 $(\leqslant)$ on $\mathbb{R}^2$*

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

$$(a, b) \leqslant (c, d) \Longleftrightarrow (a \leqslant c) \wedge (b \leqslant d) \tag{9}$$

let $\boldsymbol{m}$: Circles × Circles → Circles be a function that chooses the minimum circle of two given circles

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

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

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 $(p_a \leqslant p_b) \wedge (v_a \leqslant v_b) \wedge (r_a \leqslant r_b)$ defines a total ordering on Circles.