Boids

Tuesday, November 30, 2021 1:15 PM

<u>Recall</u>: Our simple 2D vehicle model allows us to move the vehicle to a desired position, P_d.

How can we choose P_d to produce interesting behaviors.

<u>Simple vehicle model parameters</u>

- mass m, inertia I
- position p, heading θ
- velocity v, heading rate $\dot{\theta}$ (angular velocity)
- max force (scalar magnitude)
- max speed (scalar magnitude)
- orientation R

Image we have a character (aka agent) whose movement is controlled by our vehicle model. Where the character moves is determined by our "boid" algorithm.

<u>Boid Algorithm (high level)</u>:

<u>update</u> (dt) // called each frame, dt is the time since the last frame, usu. 0.01s

sense (dt) // Compute desired velocity + orientation, v_d + θ_d, on the current "behaviour"

control (dt) // Compute force + torque, F and τ "tau", based on v_d + θ_d

act (dt) // Compute derivatives + update character state

<u>Boid Behaviors (aka "steering behaviors")</u>

<u>Two kinds</u>:

individual: each agent acts on its own

group: agents take into account neighbors

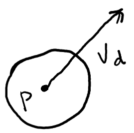
<u>Individual Behaviors</u>:

Seek: Vehicle chases a given target

Seek : Vehicle chases a given target

$\cdot P_d$

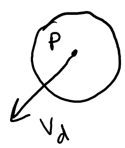
$$V_d = \left(\frac{P_d - P}{\| P_d - P \|} \right) maxSpeed$$



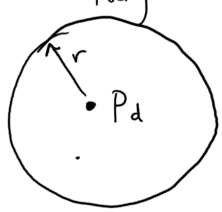
Flee : Vehicle runs away from a target

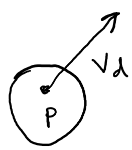
$\cdot P_d$

$$V_d = -\left(\frac{P_d - P}{\| P_d - P \|} \right) maxSpeed$$



Arrival : Similar to seek, except vehicle slows down near the target





P_d = desired pos

r = arrival radius

target Offset $= P_d - P$

distance $= \| P_d - P \|$

if distance $\leq r$

speed $= (\text{distance} / r)$ maxSpeed

else

speed = maxSpeed

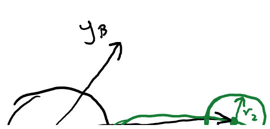
$$V_d = \text{speed} \left(\frac{P_d - P}{\| P_d - P \|} \right)$$

Departure : opposite of Arrival, gradually accelerate

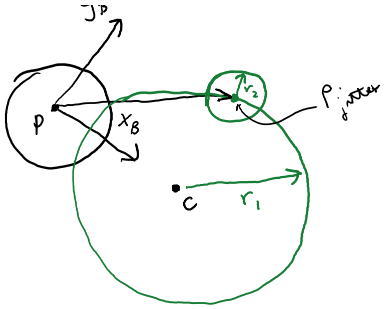
Wander : aimless, smooth movement

note : Straight forward approach of choosing a random V_d each frames is twitchy

Approach : Compute a random offset to the current V_d. Constrain the new velocity to a circle in front of the character



r_1 = wander strength; how large are curves in the wander trajectory



r_1 = wander strength ... wander trajectory
curves in the

r_2 = magnitude of random displacements ("wander rate")

c = center of circle (local coordinates)

$$\theta_{jitter} \mathrel{+}= \text{random}(-r_2, r_2)$$

$$p_{jitter} = c + r_1 \begin{pmatrix} \cos(\theta_{jitter}) \\ 0 \\ \sin(\theta_{jitter}) \end{pmatrix}$$

$$V_d = \left[(p_{jitter} - p) / \| p_{jitter} - p \| \right] * \text{MaxSpeed}$$

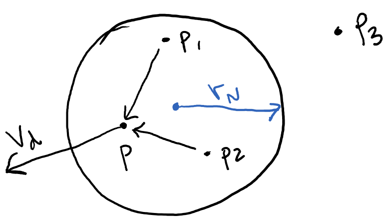
Alternate Approach

$$\theta \mathrel{+}= dt$$
$$x = fn(\theta)$$
$$z = fn(\theta)$$
$$V_d = \text{normalize}\begin{pmatrix} x \\ 0 \\ z \end{pmatrix} * \text{Max Speed}$$

Aside: Other behaviors are "obstacle avoidance", & "path following"

Group Steering Behaviors:

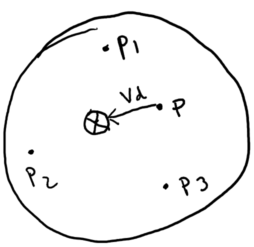
Separation: rule for maintaining distance between vehicles



$$V_d = \sum_{i \in N} \frac{p - p_i}{\| p - p_i \|^2}$$

where N is the set of all vehicles within radius r_N

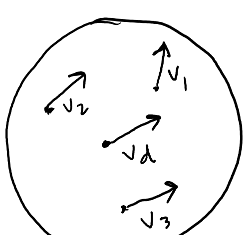
Cohesion: rule for keeping vehicles together



$$V_d = \underbrace{\sum_{i \in N} \frac{p_i}{|N|}}_{\text{average position of all neighbors}} - p$$

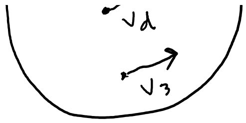
$|N|$: # neighbors

Alignment: rule for pointing all vehicles in the same direction



$$V_{alignment} = \sum_{i \in N} V_i$$

$$\text{ave Speed} = \frac{1}{|N|} \sum_{i \in N} \| V_i \|$$



ave speed $\frac{}{|N|} \underbrace{}_{i \in N}$

$$V_d = \left(\frac{V_{alignment}}{\| V_{alignment} \|} \right) \text{ave Speed}$$

Flocking: combine alignment, cohesion, & separation

$$V_{flocking} = k_{sep} V_{separation} + k_{coh} V_{cohesion} + k_{align} V_{alignment}$$

Leader Following:

→ designate one character as the leader
  → leader has arrival behavior
→ all others perform the following behavior

$$V_{follow} = K_{arr} \underbrace{V_{arrival}}_{\uparrow \text{ uses leader as target}} + k_{flock} V_{flocking}$$

Once we have V_d, we plug it into our steering controller

① Given V_d, we compute F & τ

$$V_d = \underbrace{\| V_d \| > maxSpeed}_{\text{condition}} \; ? \; \underbrace{maxSpeed * \frac{V_d}{\| V_d \|}}_{\text{cond is true}} \; : \; \underbrace{V_d}_{\text{cond is false}}$$

$$F = m K_v (V_d - V)$$

$$F = \| F \| > max\ force \; ? \; maxforce * \frac{F}{\| F \|} \; : \; F$$

$$\theta_d = atan2(V_{dz}, V_{dx})$$

$$\tau = I (k_p \theta_d - K_v \dot{\theta} - K_p \theta)$$

$$\tau = \| \tau \| > maxTorque \; ? \; maxTorque * \frac{\tau}{\| \tau \|} \; : \; \tau$$

② Compute derivatives & update our state
(same as before)