Boids

Tuesday, November 30, 2021 1:15 PM

Recall: 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.

Simple vehicle model parameters

- 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.

Boid Algorithm (high level):

update (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

Boid Behaviors (aka "steering behaviors")

Two kinds:

- individual : each agent acts on its own
  + neighbors

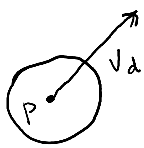
group: agents take into account ...

Individual Behaviors:

Seek: Vehicle chases a given target

$\cdot P_d$

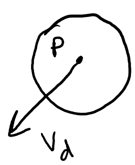
$$V_d = \left(\frac{P_d - P}{\| P_d - P \|} \right) maxSpeed$$



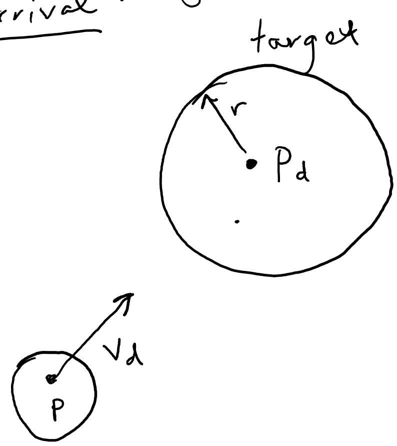
Flee: Vehicle runs away from a target

$\cdot P_d$

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



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



P_d = desired pos

r = arrival radius

targetOffset = $P_d - P$

distance = $\| P_d - P \|$

if distance $\leq r$

  speed = (distance / r) maxSpeed

else

  speed = maxSpeed

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

Departure: opposite of Arrival, gradually accelerate

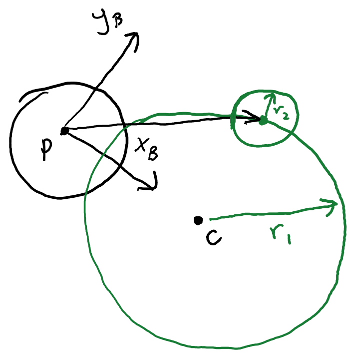
Wander: aimless, smooth movement

note: straight forward approach of choosing a random V_d [illegible] frame is twitchy

Wander

note: straight forward approach of choosing ... each frame 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_2 = magnitude of random displacements ("wander rate")

c = center of circle (local coordinates)

jitter velocity = $(r_2 * \text{random}(0,1),\ r_2 * \text{random}(0,1))$

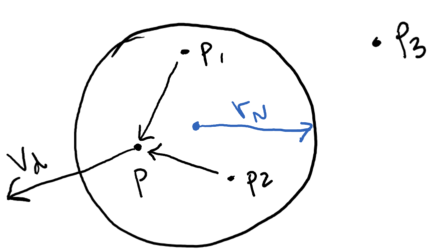
$$V_{jitter} = r_1 * \frac{\text{jitter velocity}}{\|\text{jitter velocity}\|}$$

$$V_d = V_d + V_{jitter}$$

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

Group Steering Behaviors:

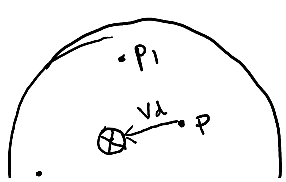
Separation: rule for maintaining distance between vehicles



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

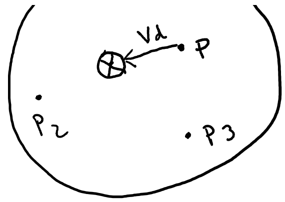
where N is the set of all vehicles within radius r_N

Cohesion: rule for keeping vehicles together



$$V_d = \underbrace{\sum_{i \in N}}_{} \frac{p_i}{|N|} - p$$

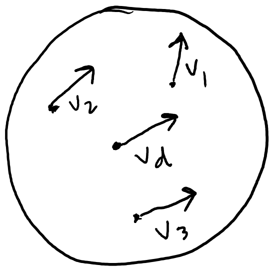
#neighbors



$\underbrace{\quad i \in N \quad}$ #neighbors

average position of all neighbors

<u>Alignment</u>: rule for pointing all vehicles in the same direction



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

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

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

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

$$V_{follow} = K_{arr} V_{arrival} + k_{flock} V_{flocking}$$

uses leader as target

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

① Given V_d, we compute F & τ

$$V_d = \underbrace{\|v_d\| > \text{max Speed}}_{\text{...dition}} \; ? \; \underbrace{\text{maxSpeed} * \frac{V_d}{\|V_d\|}}_{} \; : \; \underbrace{V_d}_{}$$

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

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

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

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

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

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

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