Steering Behaviors

CSCI 321
based on *Programming Game AI by Example*, Mat Buckland, 2005

WWU

November 9, 2017

Steering Behaviors

- Good tutorial:
  `https://gamedevelopment.tutsplus.com/series/understanding-steering-behaviors--gamedev-12732`
- The original:
  `https://www.red3d.com/cwr/steer/`

Combining Steering Behaviors

- Weighted Truncated Sum
- Weighted Truncated Running Sum with Prioritization
- Prioritized Dithering

Weighted Truncated Sum

```
SteeringForce.Zero()

SteeringForce.Add( Wander() * dWanderAmount )
SteeringForce.Add( WallAvoid() * dWallAvoidAmount )
SteeringForce.Add( Separation() * dSeparationAmount )

return SteeringForce.Truncate(MAX_STEERING_FORCE)
```

- Problems:
  - Costly: all behaviors computed every step
  - Weights difficult to tweak
  - Conflicting forces: backed into a corner by several others

Weighted Truncated Running Sum with Prioritization

```
SteeringForce.Zero()

SteeringForce.Add( WallAvoid() * dWallAvoidAmount )
if (SteeringForce.Greater(MAX_STEERING_FORCE)):
  return SteeringForce.Truncate(MAX_STEERING_FORCE)

SteeringForce.Add( Separation() * dSeparationAmount )
if (SteeringForce.Greater(MAX_STEERING_FORCE)):
  return SteeringForce.Truncate(MAX_STEERING_FORCE)

SteeringForce.Add( Wander() * dWanderAmount )
if (SteeringForce.Greater(MAX_STEERING_FORCE)):
  return SteeringForce.Truncate(MAX_STEERING_FORCE)

return SteeringForce.Truncate(MAX_STEERING_FORCE)
```

Weighted Truncated Running Sum with Prioritization

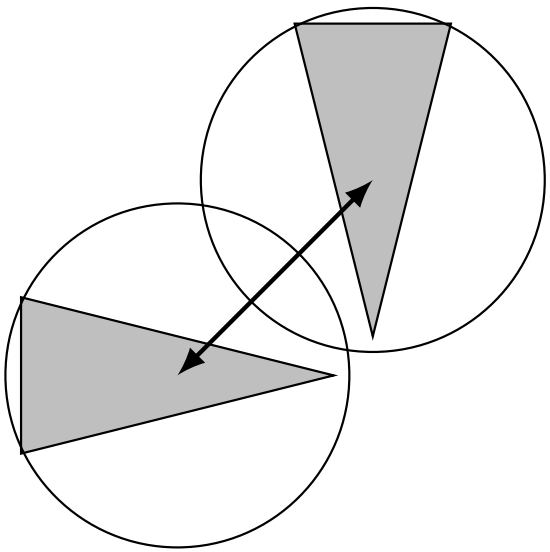
- Wall avoidance more important than vehicle alignment.
- Separation more important than align.
- If any one force becomes large, the lower priority forces are not even considered.

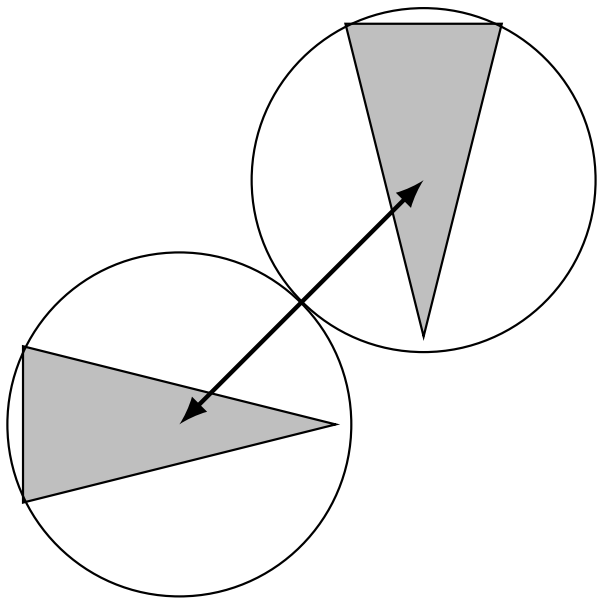
Prioritized Dithering

```
prWallAvoid = 0.9
prSeparation = 0.8
prWander = 0.5

if random.uniform() > prWallAvoid:
  SteeringForce.Add( WallAvoid() * dWallAvoid / prWallAvoid )
  return SteeringForce.Truncate(MAX_STEERING_FORCE)

if random.uniform() > prSeparation:
  SteeringForce.Add( Separation() * dSeparation / prSeparation )
  return SteeringForce.Truncate(MAX_STEERING_FORCE)

if random.uniform() > prWander:
  SteeringForce.Add( Wander() * dWander / prWander )
  return SteeringForce.Truncate(MAX_STEERING_FORCE)
```

Ensuring Zero Overlap

- Add simple collision detection using bounding circles.
- Move colliding objects along line between centers until not colliding
- No other physics (bouncing, *etc.*) added.





Spatial Partitioning

- Partition space into $O(n)$ cells.
- Each object updates its cell location.
- Only check for collisions in cells which overlap collision bound circle.
- Reduces $O(n^2)$ to $O(n)$.
- `Another Big Shoal.exe`

Smoothing

- Occasionally bots appear to shudder.
- Different steering takes place every other frame:
  - Toward object, away from obstacle, toward object, away from obstacle, ...
- Simple solution: decouple bot heading from actual velocity.
- Bot heading is average velocity of last few steps.