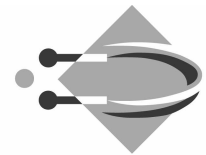


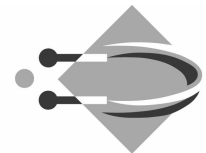
Physics in panda & AI



The **IT** University
of Copenhagen

Today

- 1 Forces in Panda
- 2 AI
- 3 Chase
- 4 Path finding
- 5 A* path finding



Forces in panda

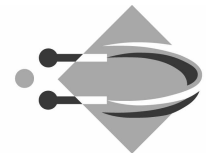
RotationTest.py, Spin+gravitation

Test.py , gravitation

CollisionExample.py

Gravityexample.py

You might want to control the car with force also.



The **IT** University
of Copenhagen

Artificial Intelligence

Mimic intelligent behavior!

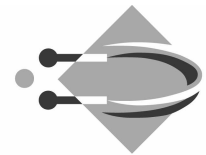
Strong AI : Solves intelligently, Adapts, Learns, Emotion, conscience

Weak AI : Specialised intelligent qualities

Deterministic and Non deterministic

Cheat, State Machines, Path finding, Script, Rule-based, Fuzzy, A-life

In the future? More Adaptivness/Learning



The **IT** University
of Copenhagen

Some topics

Chasing

Pattern movements (motion paths)

Flocking

Potential

Pathfinding

Scripting

Finite State Machines

Fuzzy logic

Rule-Based AI

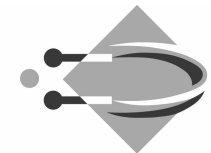
Probability

Bayesian

Neural network

Genetic algorithm

Learning algorithms



The **IT** University
of Copenhagen

Chasing

- Line-Of-Sight

- Car chase

 - Thrust force+left& right steering force

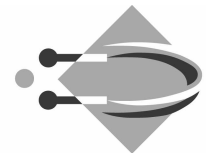
 - Known positions of prey & predator

 - Use difference in position, local orientation

- Global to local

$$x = X \cos a + Y \sin a$$

$$y = -X \sin a + Y \cos a$$



Intercepting

Need estimate of position and velocity of prey

Closing velocity = $V_{prey} - V_{predator}$

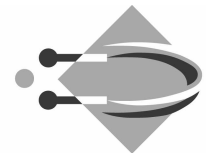
Range to close = $S_{prey} - S_{predator}$

Time to close = $|Range\ to\ close| / |Closing\ velocity|$

Predicted position:

$S_{prey}(T_c) = S_{prey} + V_{prey} T_c$

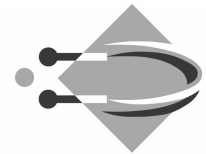
Prey should be in front which can be done by switching to line-of-sight chasing



Pattern movement (motion paths)

Physical Simulated Environments do usually not coexists well with position pattern movements.

Instead control patterns can be created but take care of possible drifting in a looped pattern



Flocking

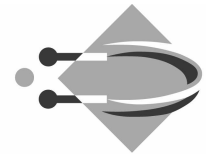
Cohesion (move towards average)

Alignment (head towards average)

Separation (Avoid hitting)

Field of view changes flocking behavior severely.

Adding a pointman/leader



Pathfinding

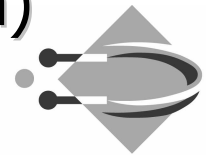
Line-Of-Sight alternating with random movement.

Obstacle avoidance by tracing the outskirts

Wall tracing

"Breadcrumbs"

Waypoint Navigation (Carrace and in general a graph)



The **IT** University
of Copenhagen

A*

Search area/domain

Algorithm

Store parent relation to make path

Score = cost from start + heuristic

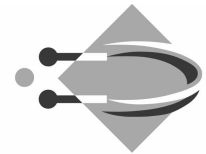
Is this optimal??

Dead end = empty open list and no goal

Terrain cost/Metrix/distance measure

General graph for continous enviroment

Influence maps (Line of sight of enemy, "memory of opponent strategies")



A* "Shortest path in graph"

Add start to open list

While open list is not empty

 cur = lowest cost node from open

 if cur = goal the done

 else

 move cur to closed list

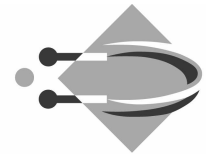
 for each adjacent node

 not on the open list and

 not on the closed list and

 not an obstacle

 move adjacent to open list and calculate cost.



Assignment

Include some physics like gravity or car control.

Use simple AI to include an opponent car.

Deadline for handing in is Friday the 1st of april

