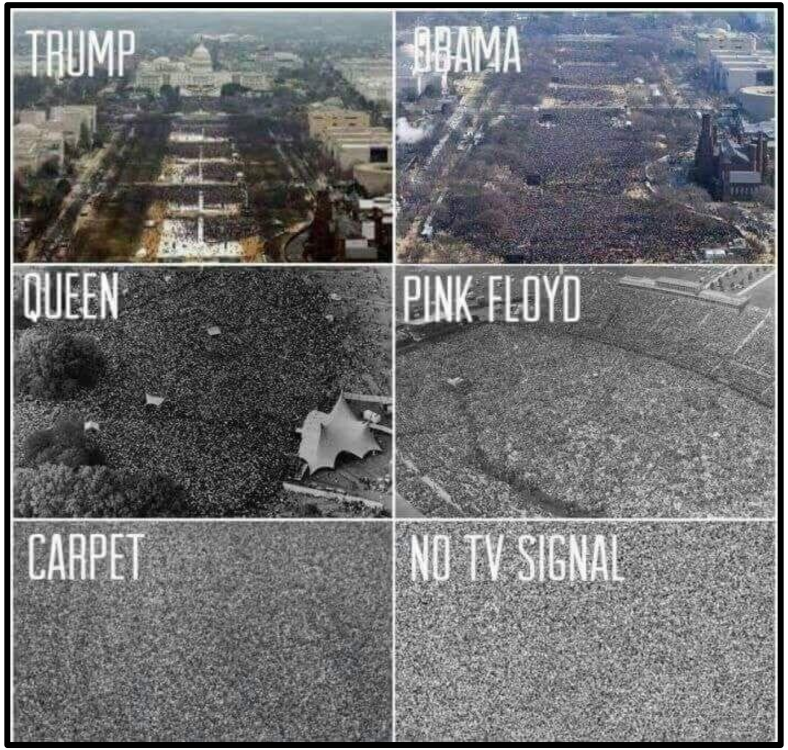
Crowd Simulation

Oihan Abruña - Guillermo Núñez -
Sebastian Paas - Iker Quintana

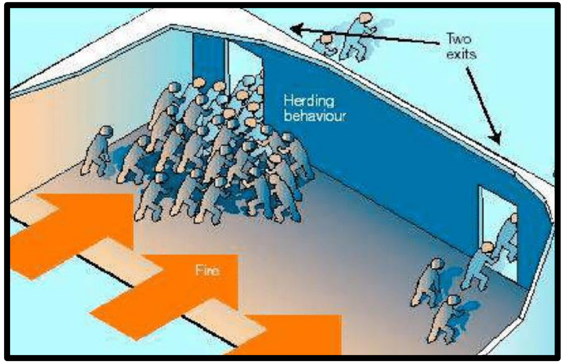
DigiPen Institute of Technology Europe – Bilbao

Crowds



Crowd Simulations

Urban Planning

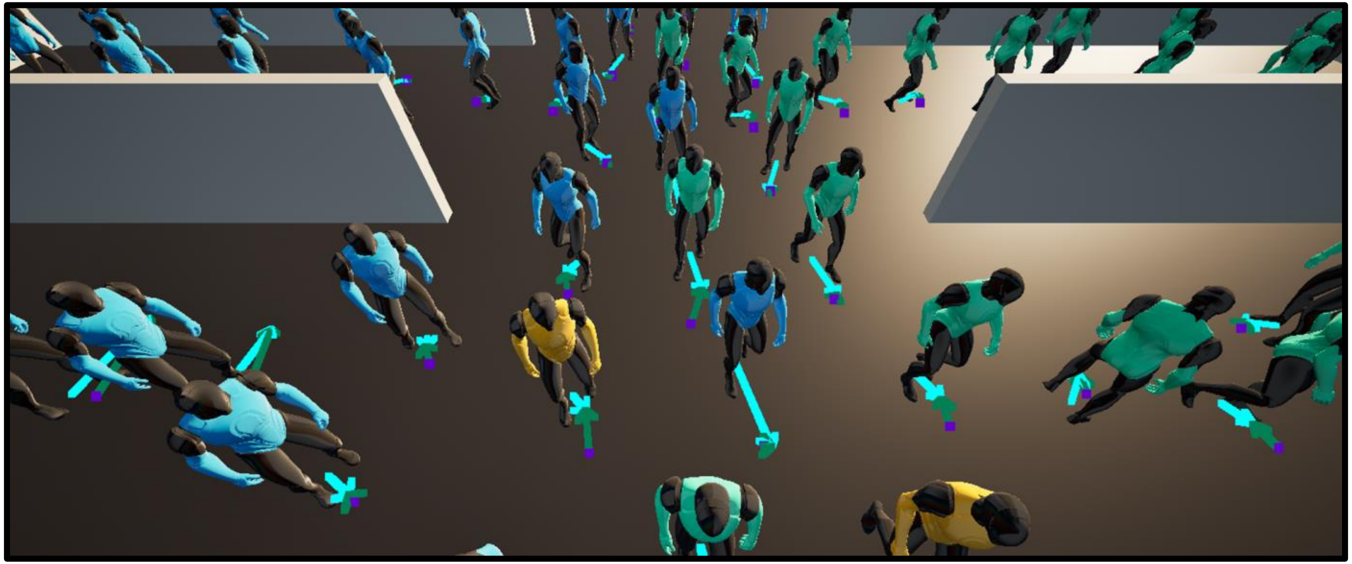


Evacuation simulation

Animation and movies

Games

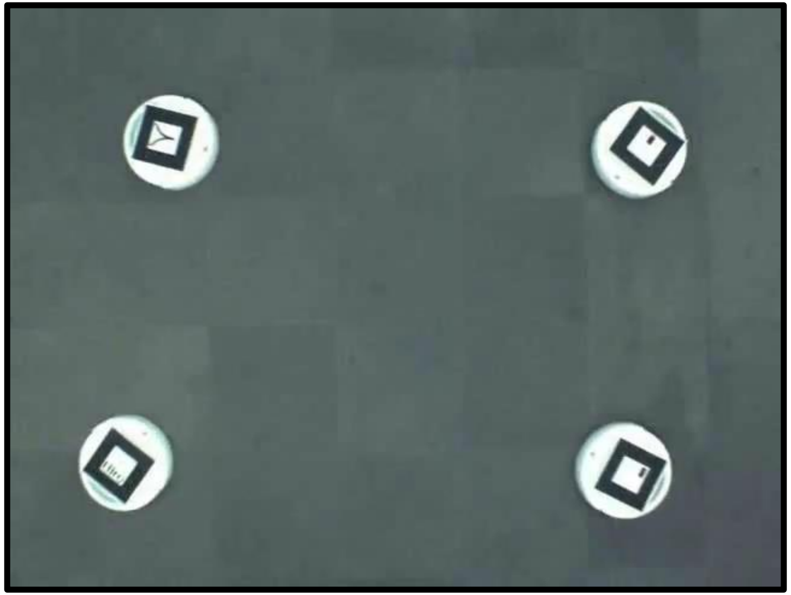
Microscopic paradigm



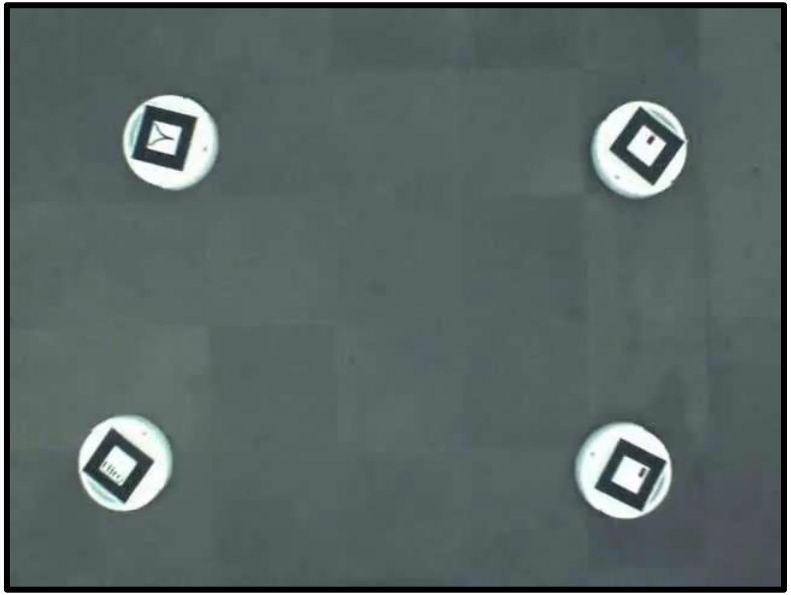
Local navigation

Optimal Reciprocal Collision Avoidance (ORCA)

Optimal Reciprocal Collision Avoidance （ORCA）



Optimal Reciprocal Collision Avoidance (ORCA)



Required data:

1. Radius of circular collider
2. Velocity
3. Position

Steps

Three-step process:

1. Calculate the set of velocities that would result in a collision (VO).

2. Select the optimal velocity outside all velocity obstacles and is closest to preferred velocity.

3. Update the agent’s velocity to this new, safe value.

How do we do this?

Three-step process:

1. Calculate the set of velocities that would result in a collision (VO).

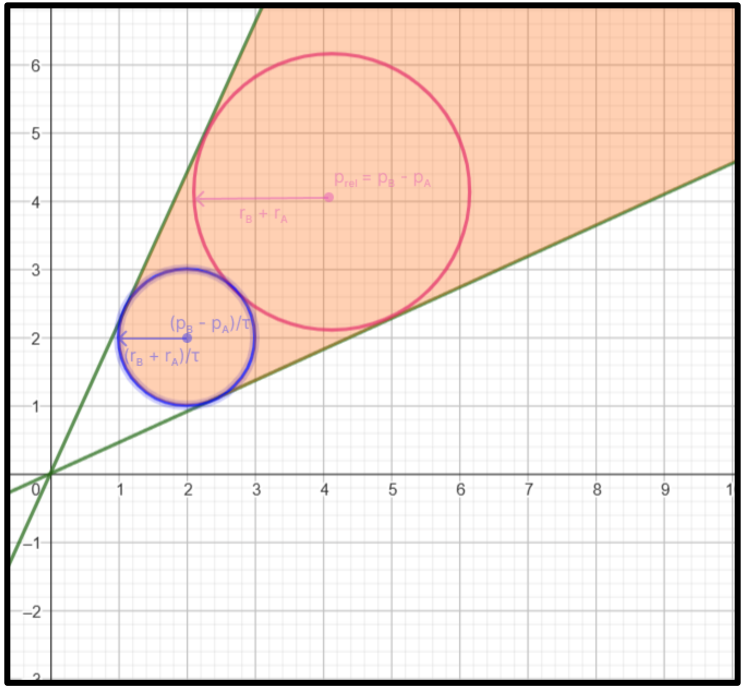
2. Select the optimal velocity outside all velocity obstacles and is closest to preferred velocity.

3. Update the agent’s velocity to this new, safe value.

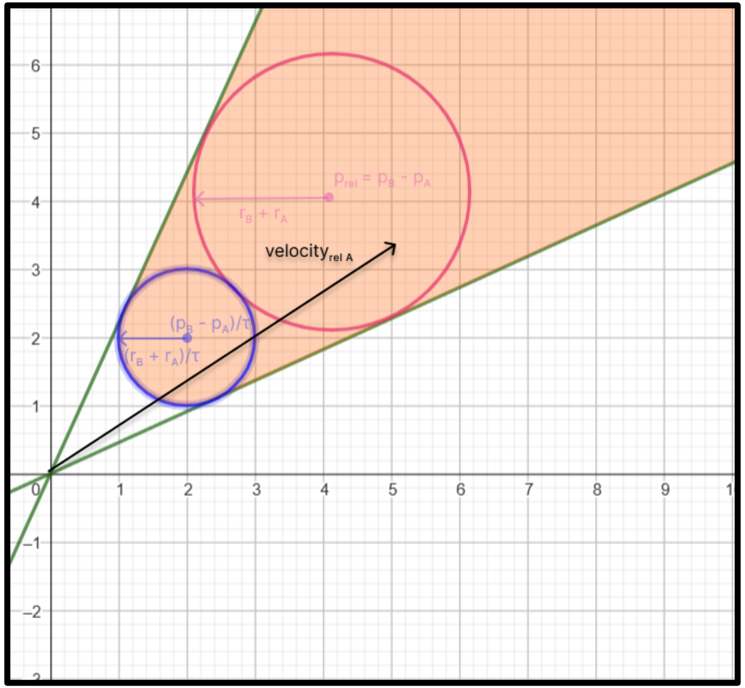
Velocity obstacles

**Do not look at the picture if you are younger than 18.*

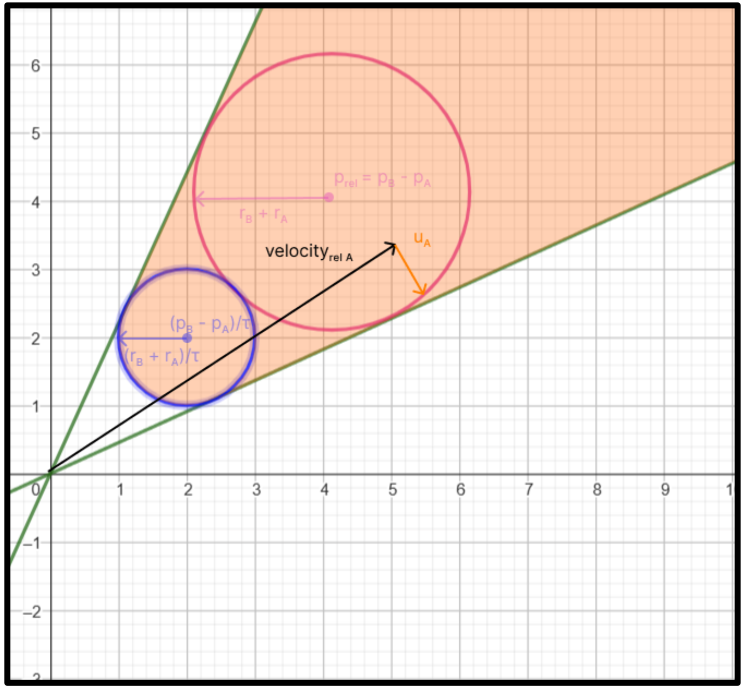
Velocity obstacles



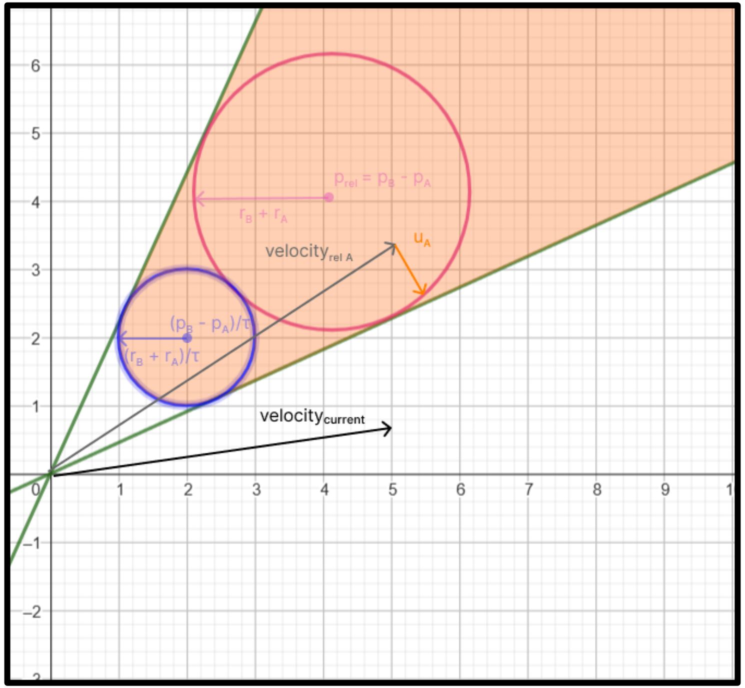
Select optimal velocity



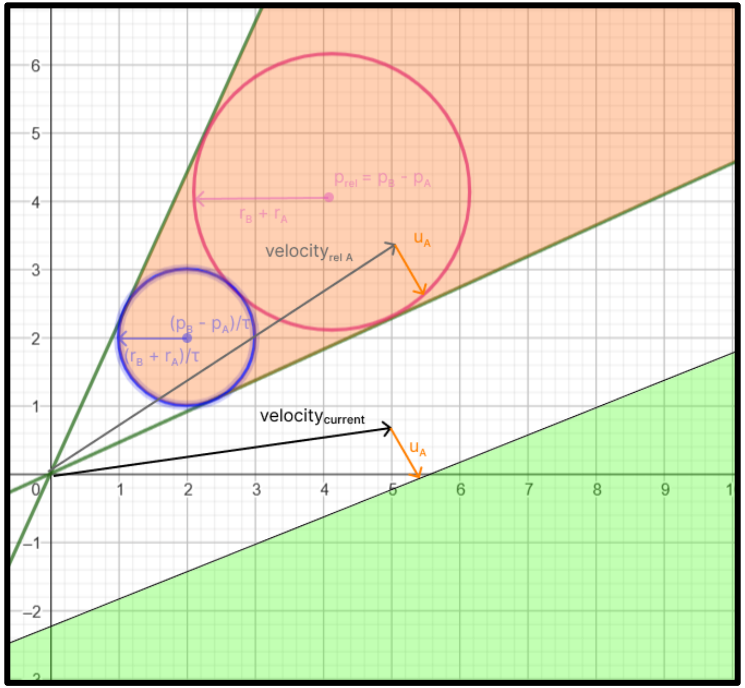
Select optimal velocity



Select optimal velocity

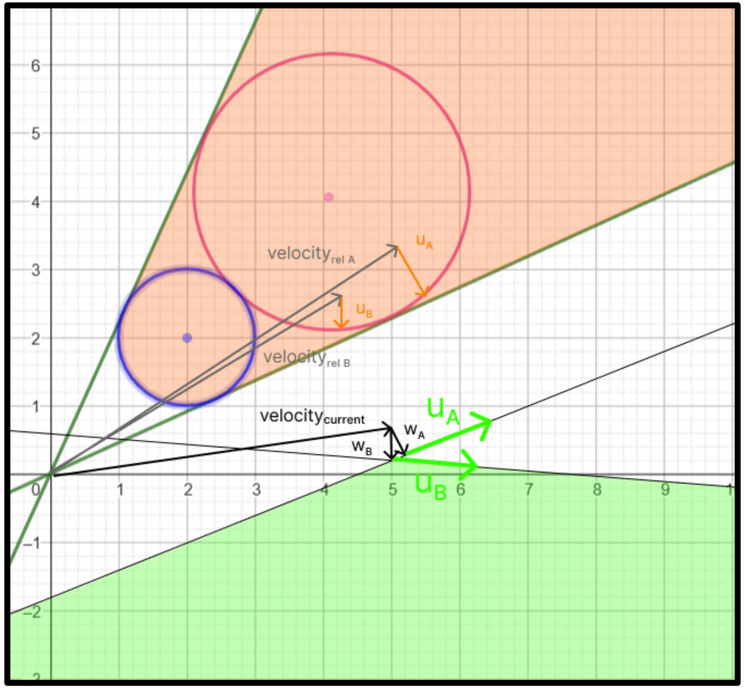


Select optimal velocity

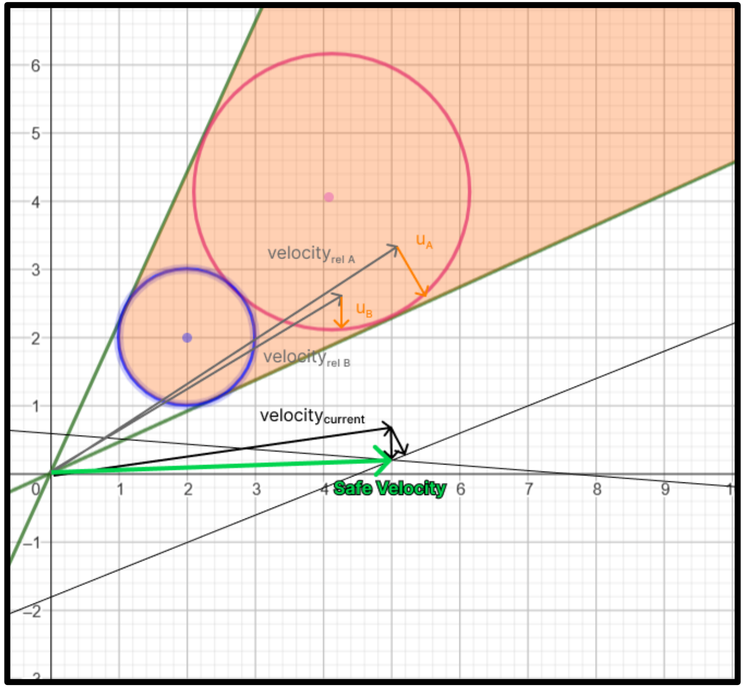


Select optimal velocity

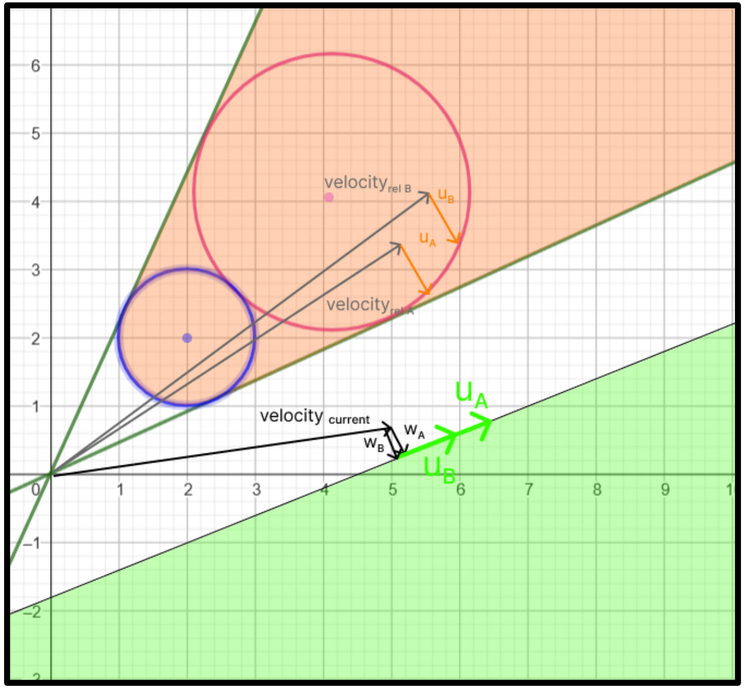
Select optimal velocity



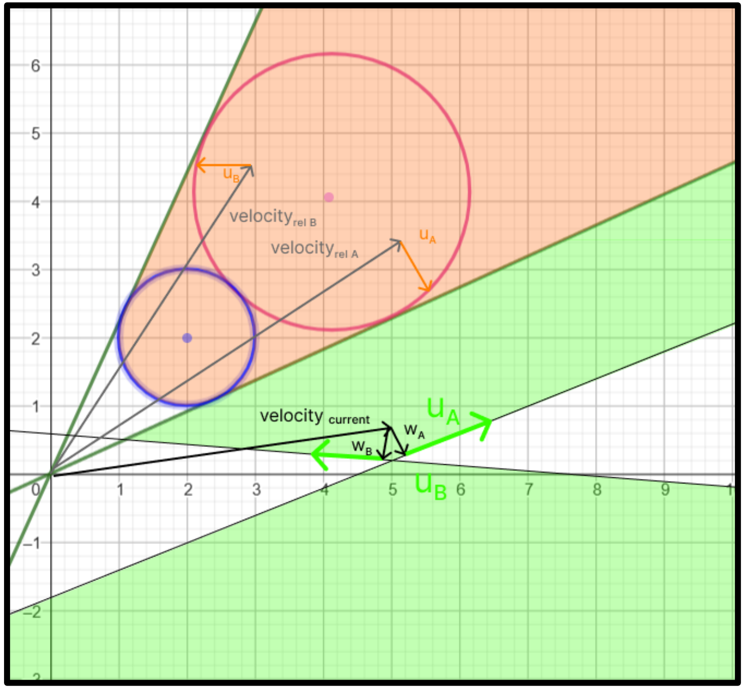
Update velocity



Edge cases



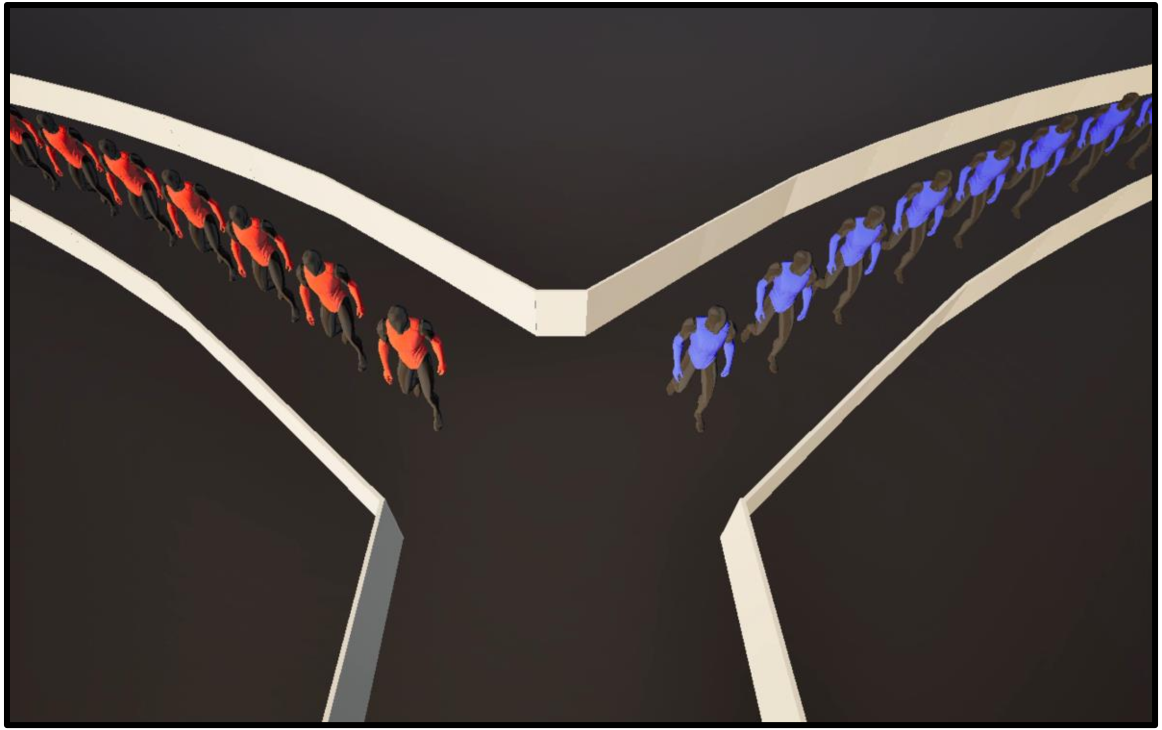
Parallel or the same lines



Opposing lines

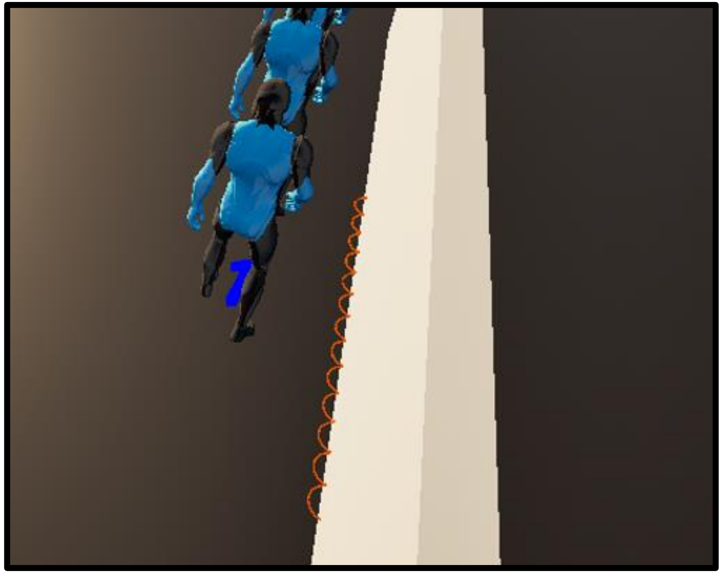
Example

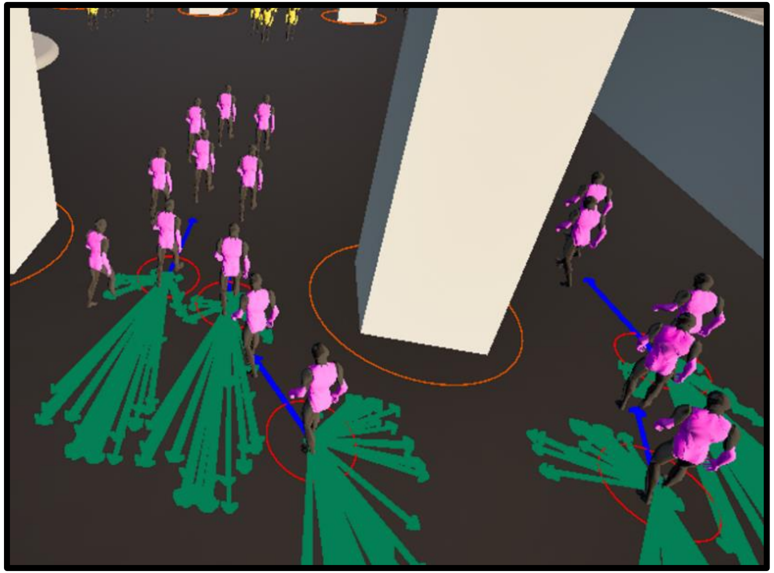
Static obstacles



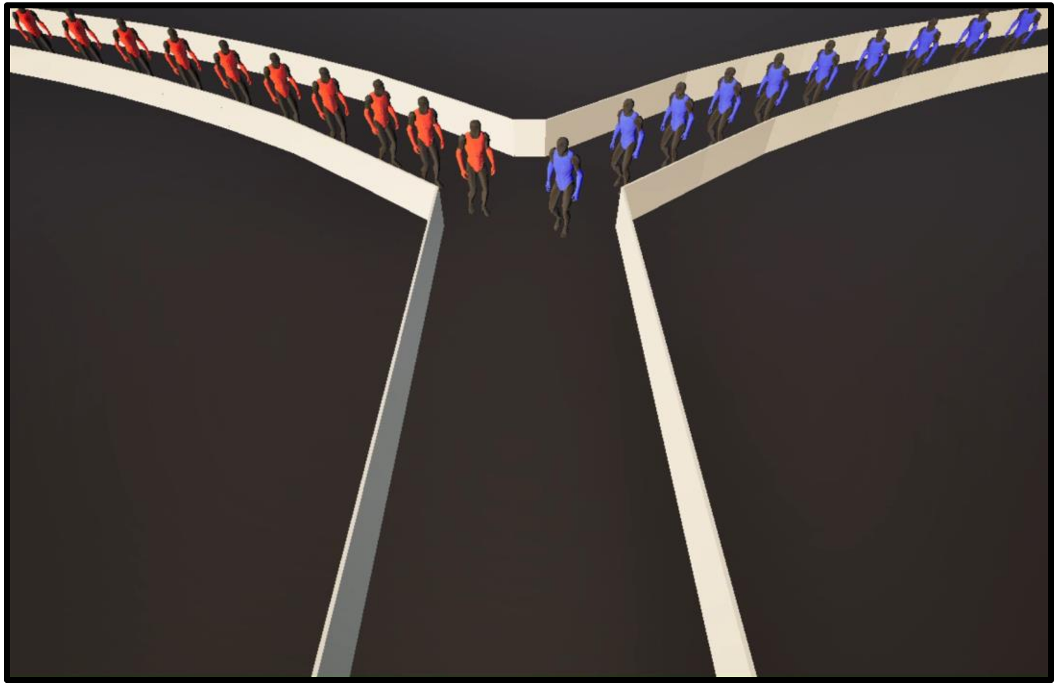
Static obstacles

Split into circles





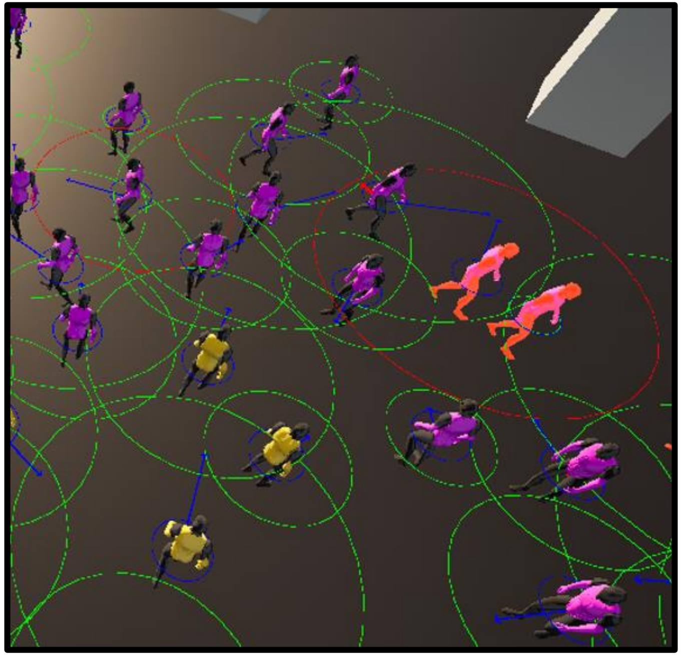
Static obstacles



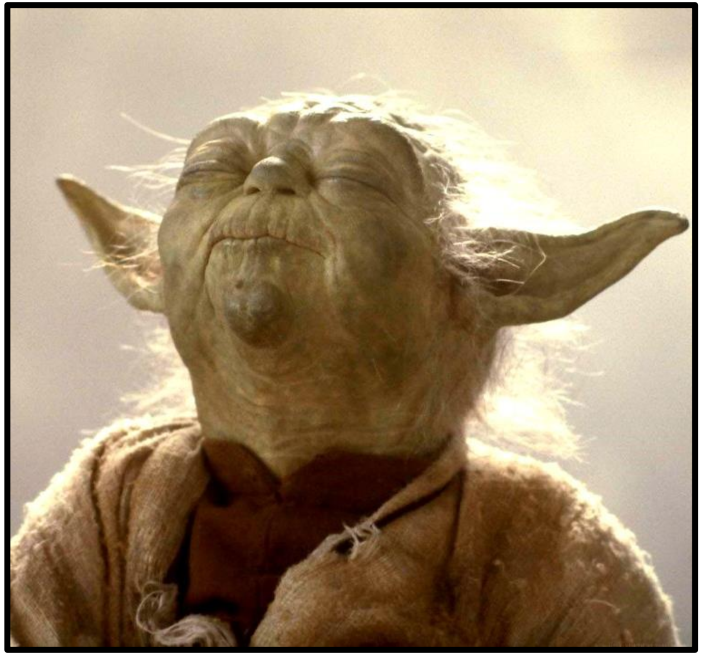
Universal Power Law

Universal Power Law

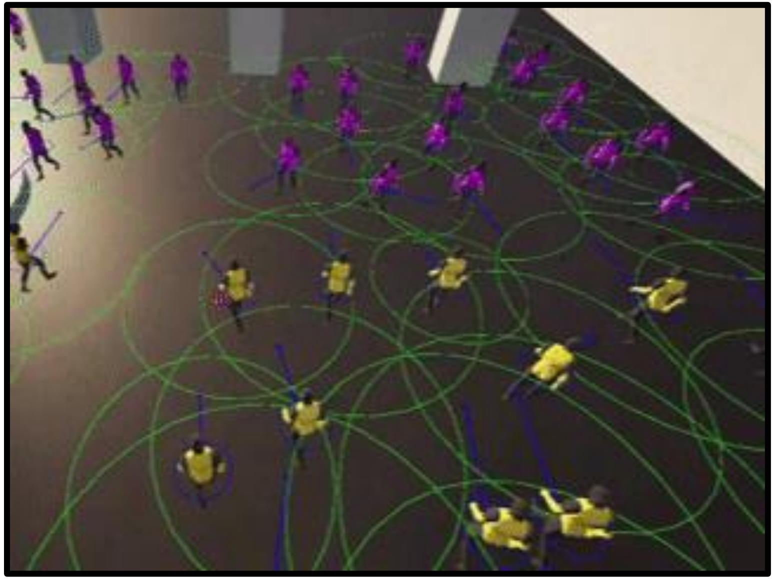
Universal Power Law



=



Steps



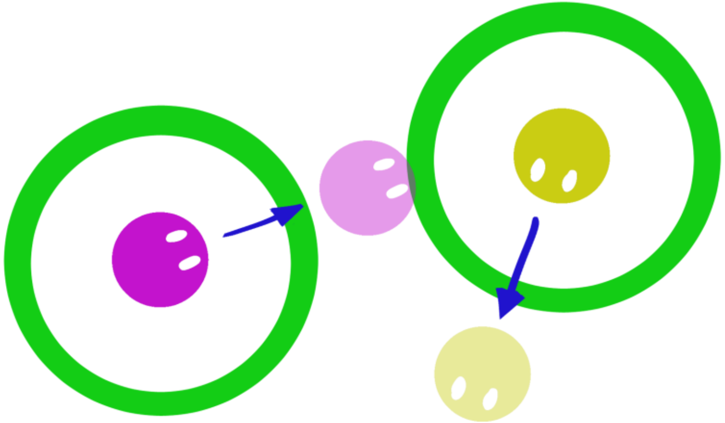
Main steps:

1. Calculate τ (time-to-collision)
2. Check Collision
3. Calculate Interaction Energy
4. Calculate Avoidance Force
5. Pray to evade

Time-to-collision

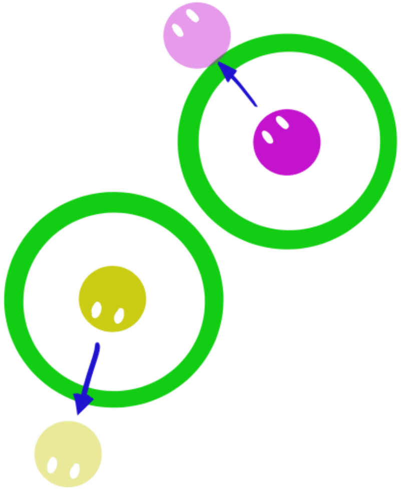
Tau(T) = time-to-collision

$$\tau = \frac{(-dp \cdot dv)}{|dv|^2}$$

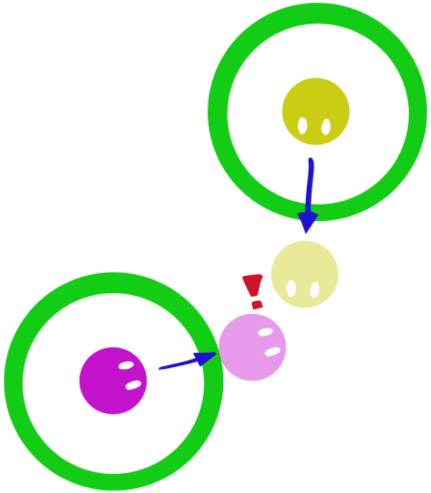


Time-to-collision

Tau(T) = time-to-collision



T is more than 3s or
negative ==> No collision



T is less than 3s ==> Collision!

Interaction Energy

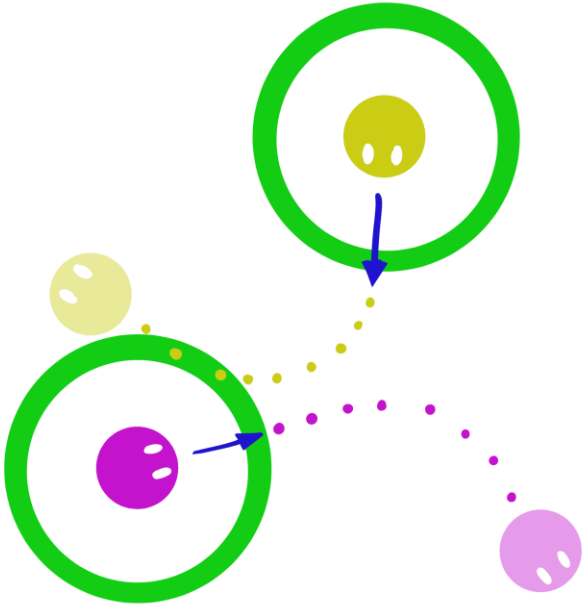
$$E(\tau) = \frac{k}{\tau^2} \cdot e^{-\tau/\tau_0}$$

When E increases ==> T decreases

Avoidance Force

Gradient of
Energy function

$$F_{ij} = -\nabla_{p_{ij}} E(\tau)$$



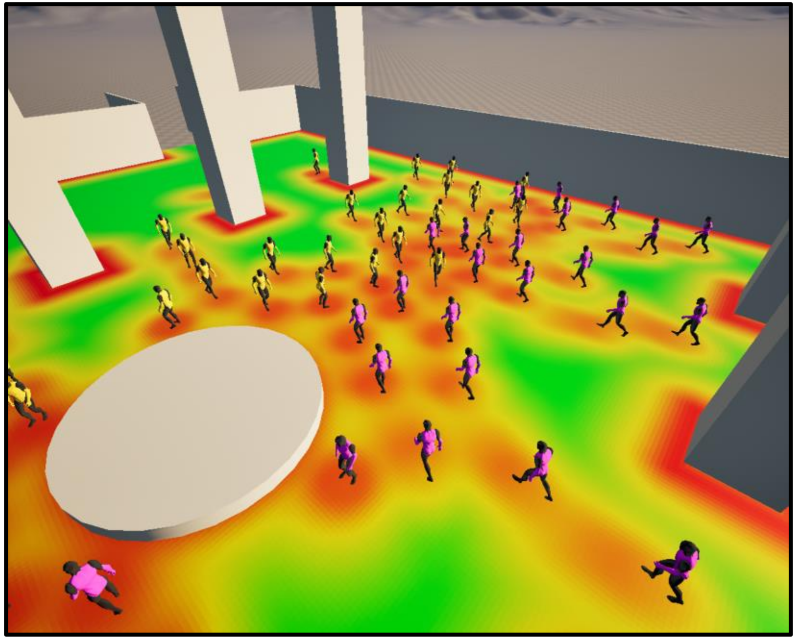
Gradient respect to the position

$$F_{ij} = -\left[\frac{ke^{-\tau/\tau_0}}{\left|v_{ij}\right|^2\ \tau^2}\left(\frac{2}{\tau}+\frac{1}{r_0}\right)\right]\left[v_{ij} - \frac{\left|v_{ij}\right|^2\ r_{ij}\ -(p_{ij}\cdot v_{ij})\ v_{ij}}{\sqrt{(p_{ij}\cdot v_{ij})^2 - \left|v_{ij}\right|^2\left(\left|p_{ij}\right|^2 - (R_i+R_j)^2\right)}}\right]$$

Credits to Karamouzas

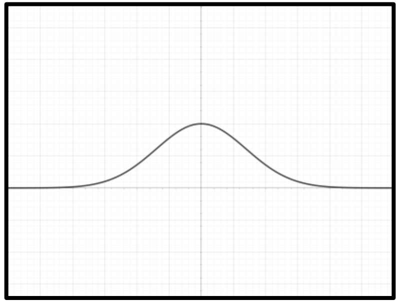
WarpDriver

Collision probability fields



Probability function

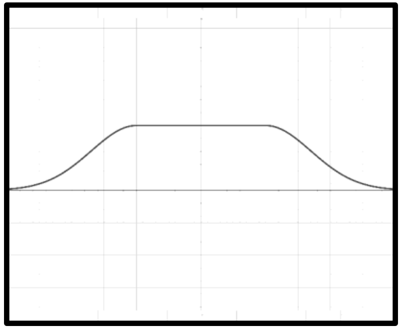
$$P(d) = e^{\left(-\frac{d^2}{2\sigma^2}\right)}$$



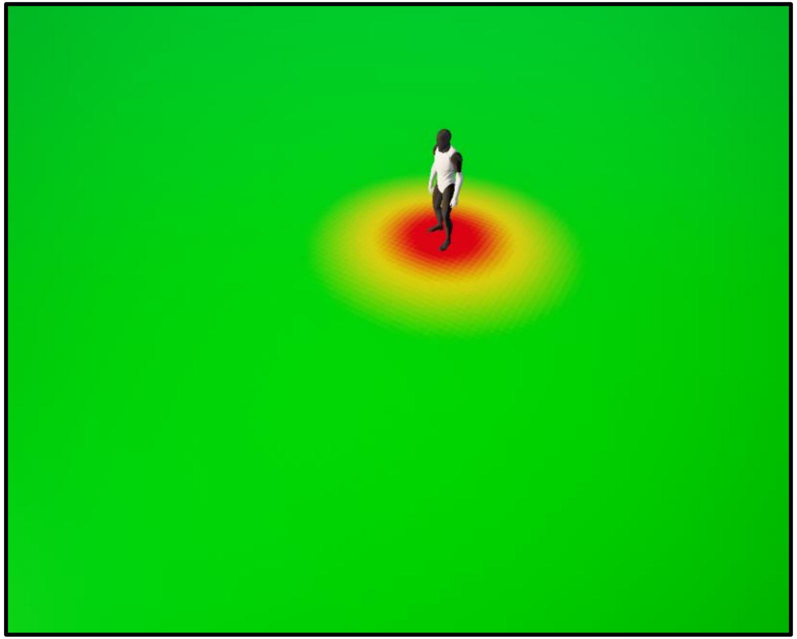
Position

Probability function

$$P(d) = e^{\left(-\frac{(d-R)^2}{2\sigma^2}\right)}$$

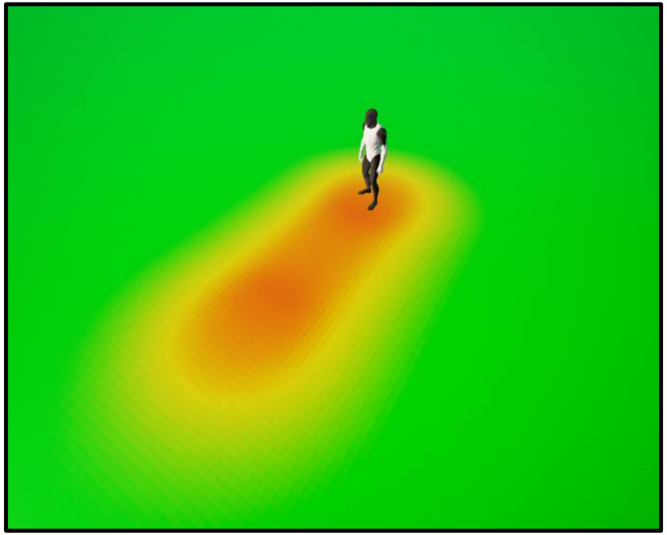


Size



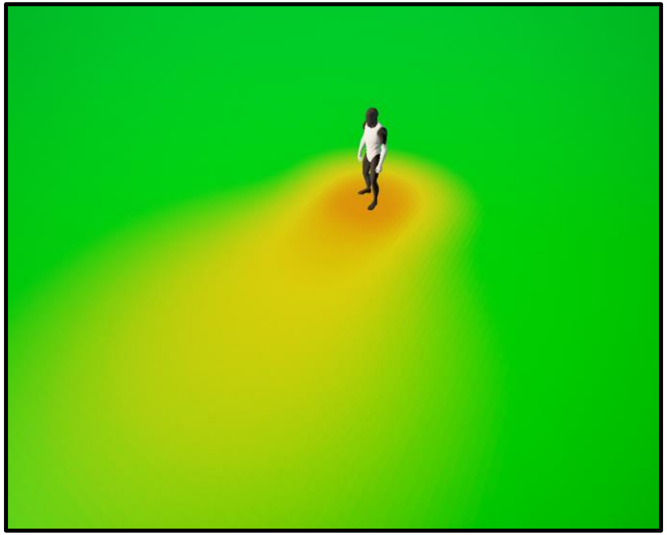
Velocity and Time horizon

$$\Delta t = \frac{T}{N}$$

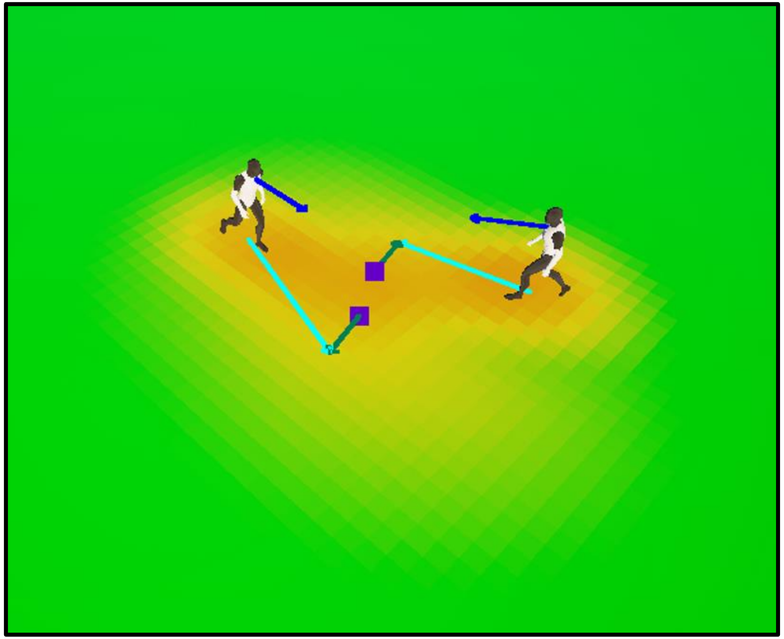


Time uncertainty

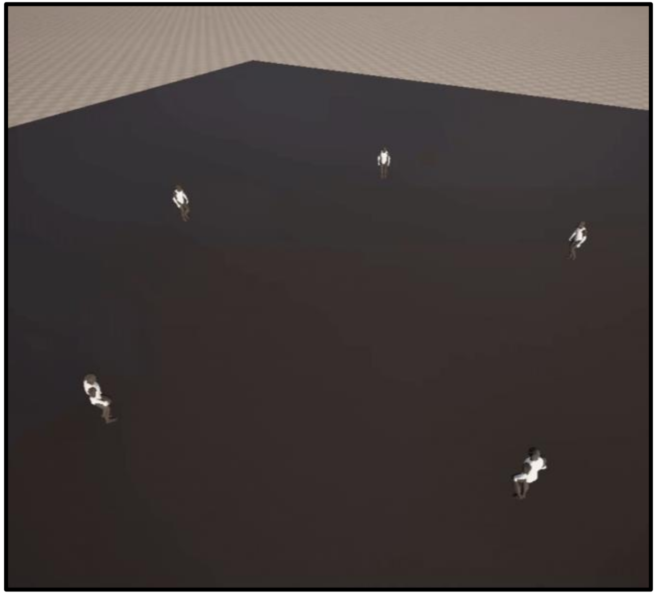
- Reduced Confidence Near the Agent: $\alpha(t) = \frac{1}{1 + \lambda t}$ $P' = \alpha(t) \cdot P$
- Increased Spread: $\sigma(t) = \sigma + \gamma \cdot t$

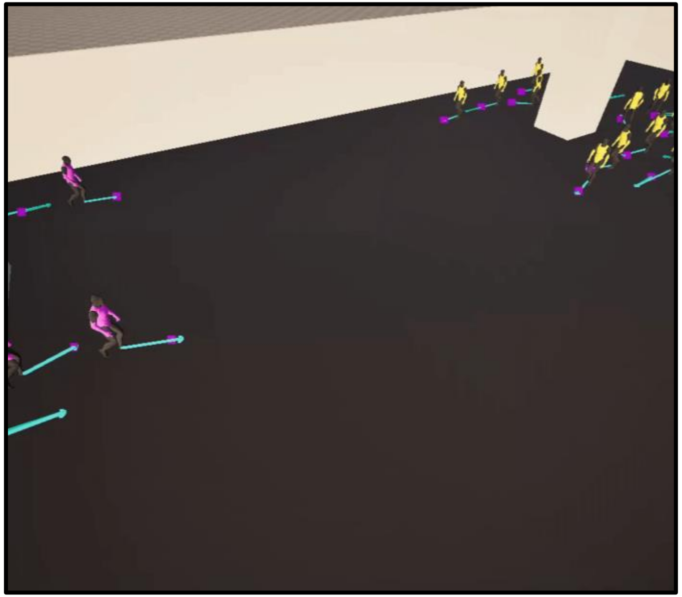


Collision avoidance

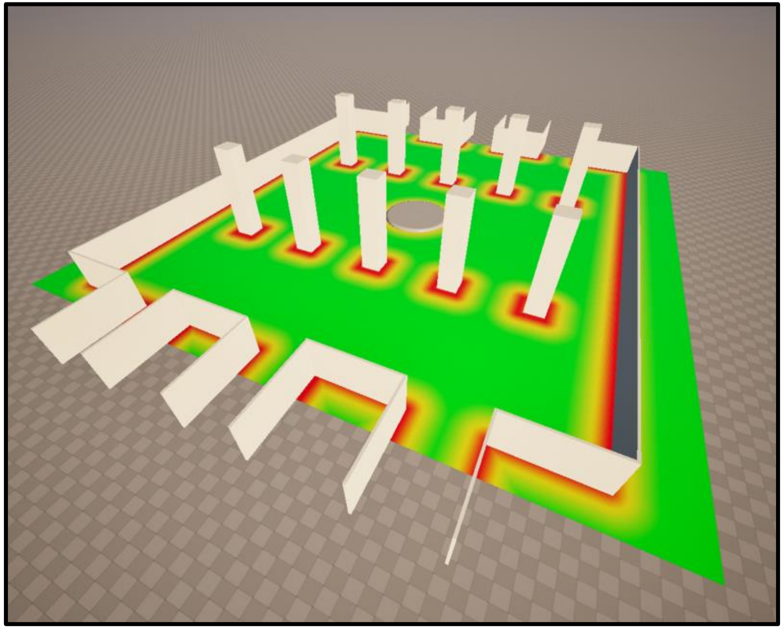


Examples



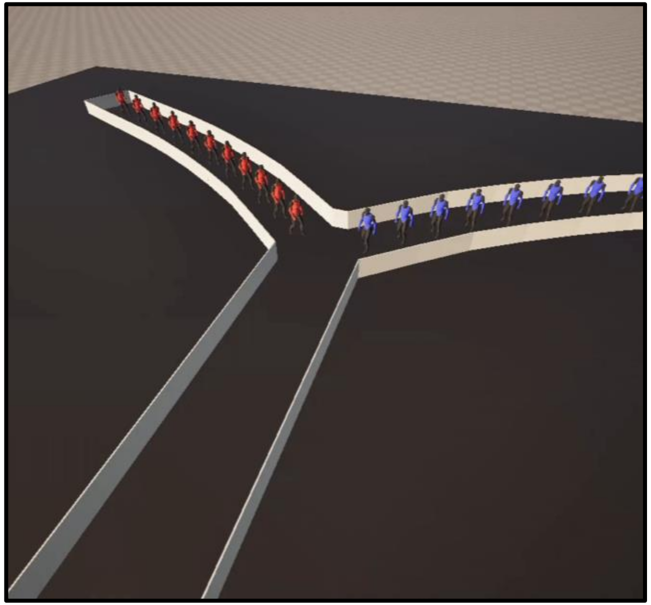


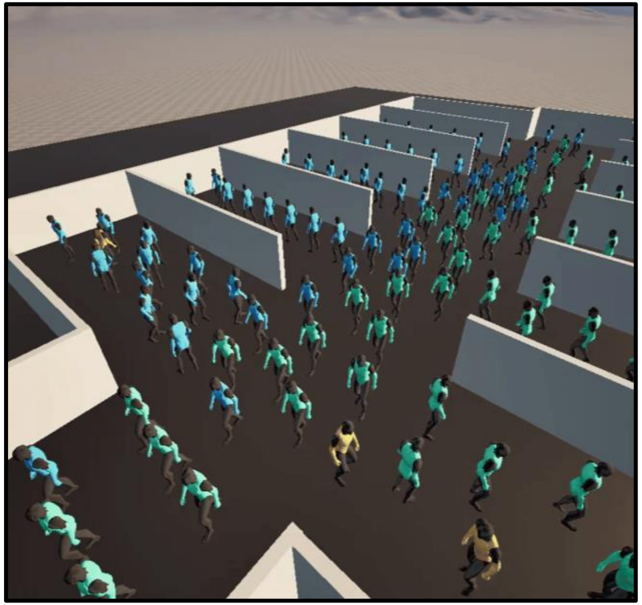
Static obstacles with non-cylindrical shapes



Following behavior

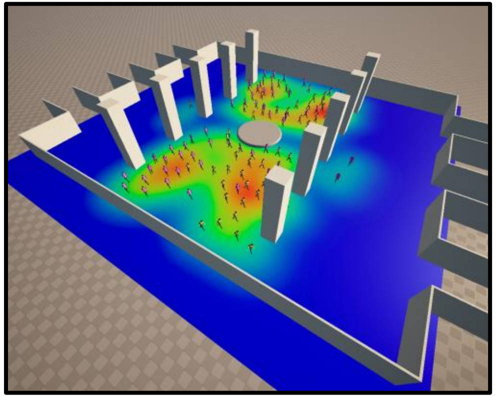
Examples

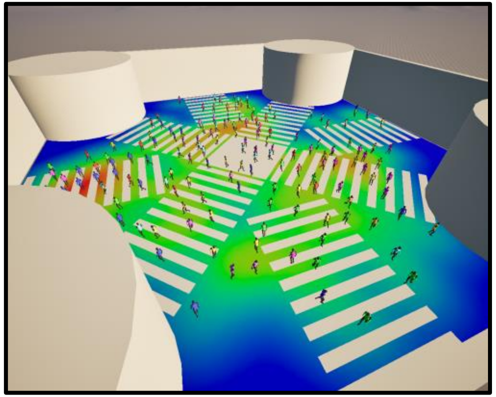




DenseSense

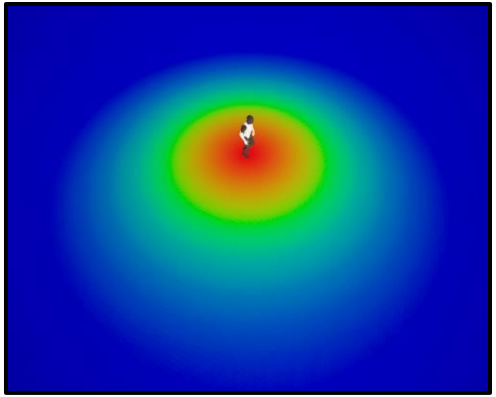
Density awareness





Crowd density function

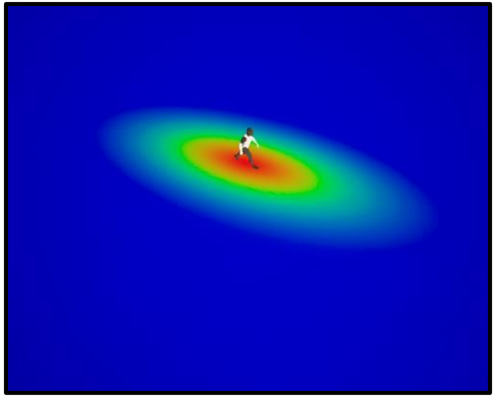
$$w(d) = \frac{1}{\sqrt{2\pi\sigma}}\, e^{\left(-\frac{d^2}{2\sigma^2}\right)}$$



Elliptical personal space

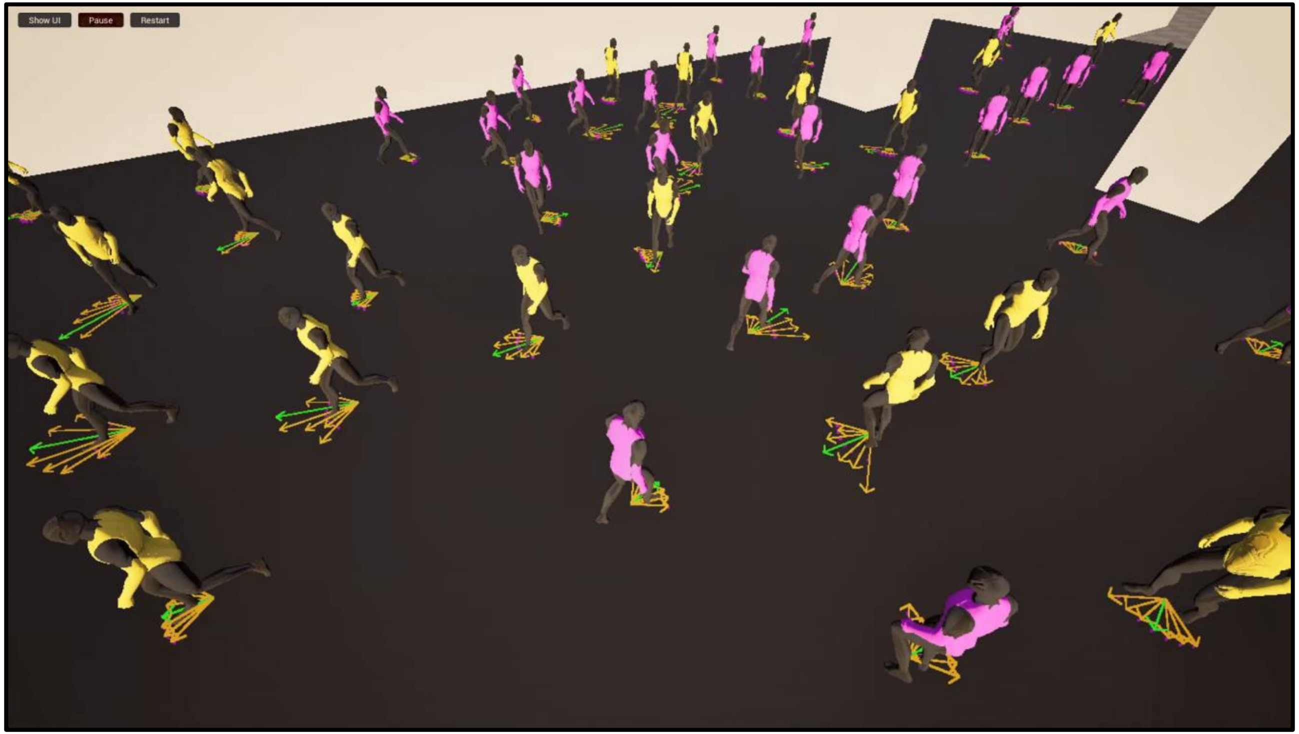
$$d' = 2.5(d - D)$$

$$D = (d \cdot v_\theta) \cdot v_\theta$$



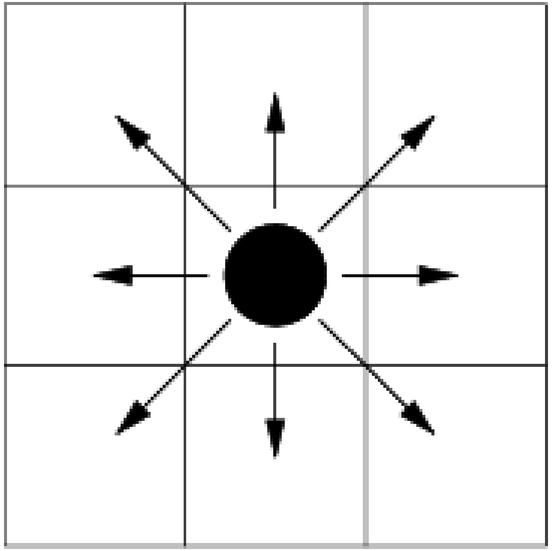
Selection of preferred velocity

Example: Hall



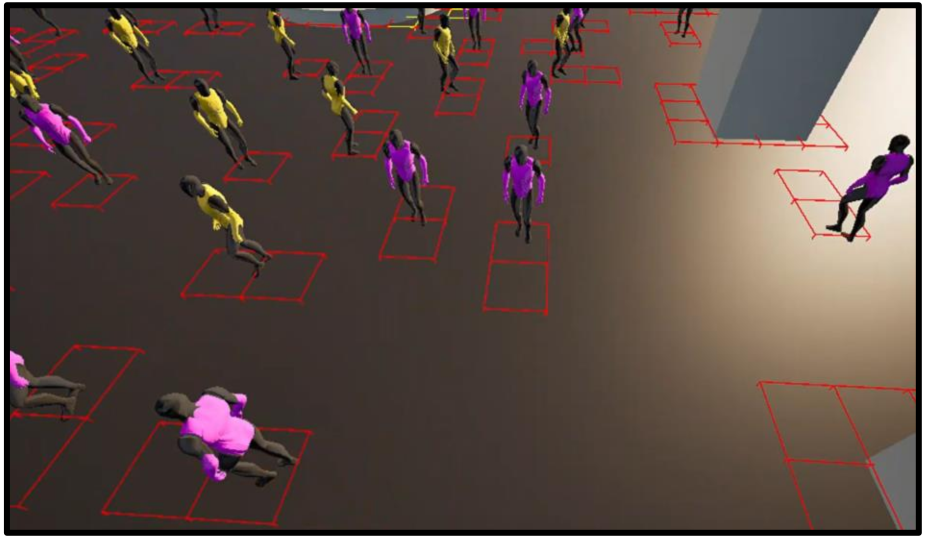
Cellular Automata

Steps



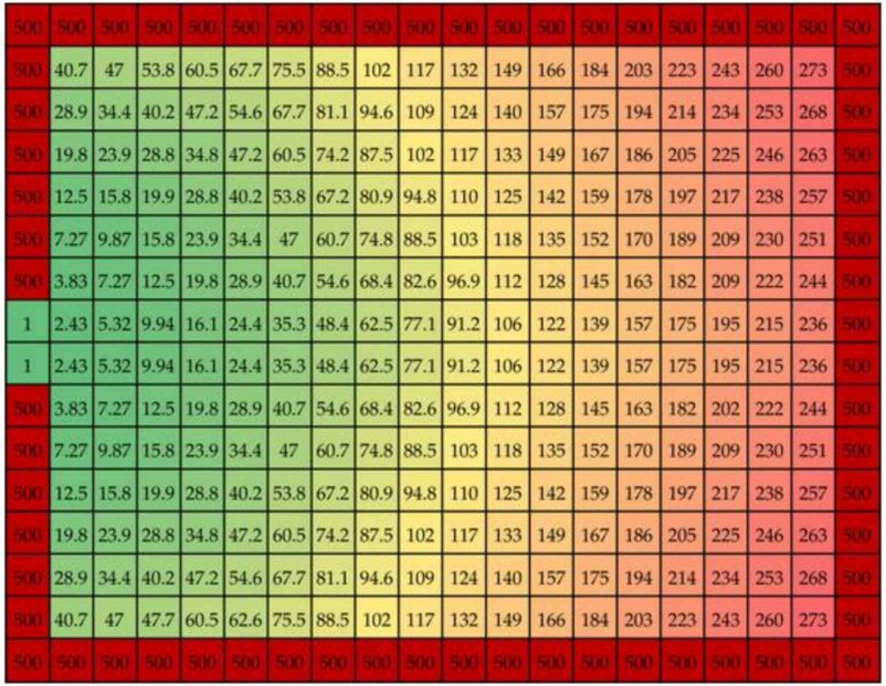
1. Check the state of the current cell 8 neighbors.
2. Cell is occupied, then skip it.
3. Cell is not occupied, then calculate distance to the goal.
4. If the distance is shorter than the previous neighbor cell, then keep it as the best result.

Occupied State



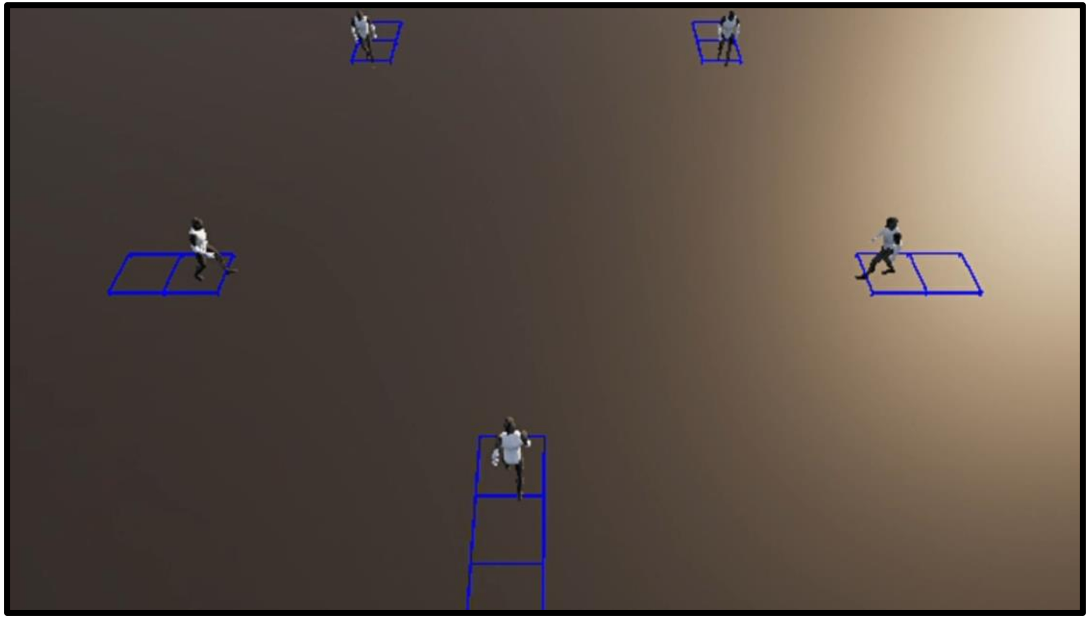
- Static Obstacles or current Agent cell (Red).
- The next cell an Agent will move to (Red).
- Neighbor cell has at least 4 occupied neighbors (Yellow).

Static Floor Field

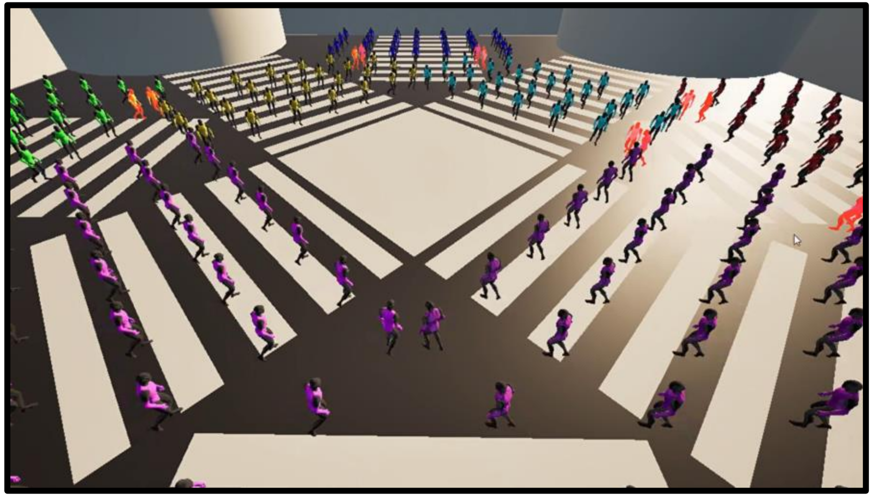


500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500
500	40.7	47	53.8	60.5	67.7	75.5	88.5	102	117	132	149	166	184	203	223	243	260	273	500
500	28.9	34.4	40.2	47.2	54.6	67.7	81.1	94.6	109	124	140	157	175	194	214	234	253	268	500
500	19.8	23.9	28.8	34.8	47.2	60.5	74.2	87.5	102	117	133	149	167	186	205	225	246	263	500
500	12.5	15.8	19.9	28.8	40.2	53.8	67.2	80.9	94.8	110	125	142	159	178	197	217	238	257	500
500	7.27	9.87	15.8	23.9	34.4	47	60.7	74.8	88.5	103	118	135	152	170	189	209	230	251	500
500	3.83	7.27	12.5	19.8	28.9	40.7	54.6	68.4	82.6	96.9	112	128	145	163	182	209	222	244	500
1	2.43	5.32	9.94	16.1	24.4	35.3	48.4	62.5	77.1	91.2	106	122	139	157	175	195	215	236	500
1	2.43	5.32	9.94	16.1	24.4	35.3	48.4	62.5	77.1	91.2	106	122	139	157	175	195	215	236	500
500	3.83	7.27	12.5	19.8	28.9	40.7	54.6	68.4	82.6	96.9	112	128	145	163	182	202	222	244	500
500	7.27	9.87	15.8	23.9	34.4	47	60.7	74.8	88.5	103	118	135	152	170	189	209	230	251	500
500	12.5	15.8	19.9	28.8	40.2	53.8	67.2	80.9	94.8	110	125	142	159	178	197	217	238	257	500
500	19.8	23.9	28.8	34.8	47.2	60.5	74.2	87.5	102	117	133	149	167	186	205	225	246	263	500
500	28.9	34.4	40.2	47.2	54.6	67.7	81.1	94.6	109	124	140	157	175	194	214	234	253	268	500
500	40.7	47	47.7	60.5	62.6	75.5	88.5	102	117	132	149	166	184	203	223	243	260	273	500
500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500

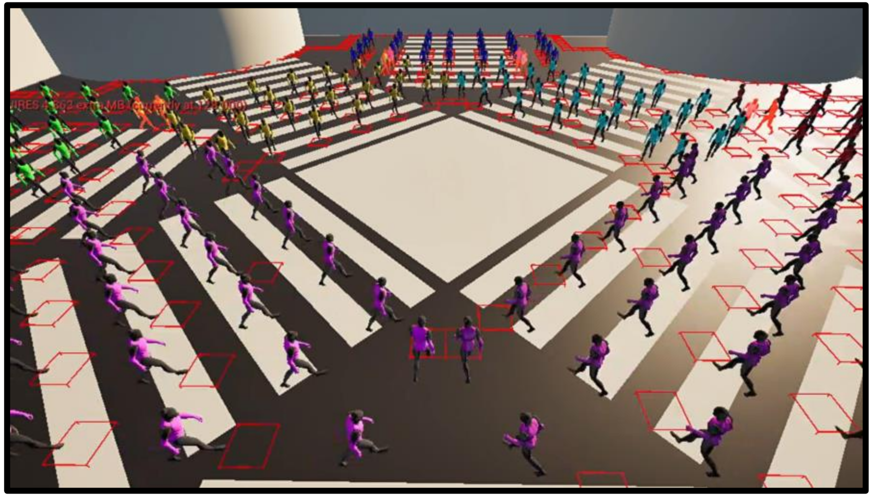
Dynamic Floor Field



Comparision



Without Floor Field



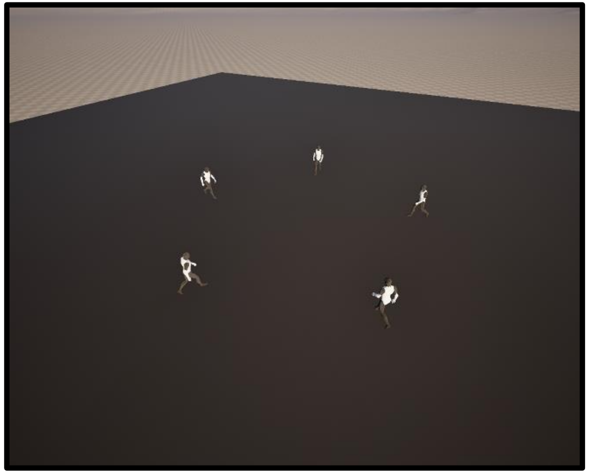
With Floor Field

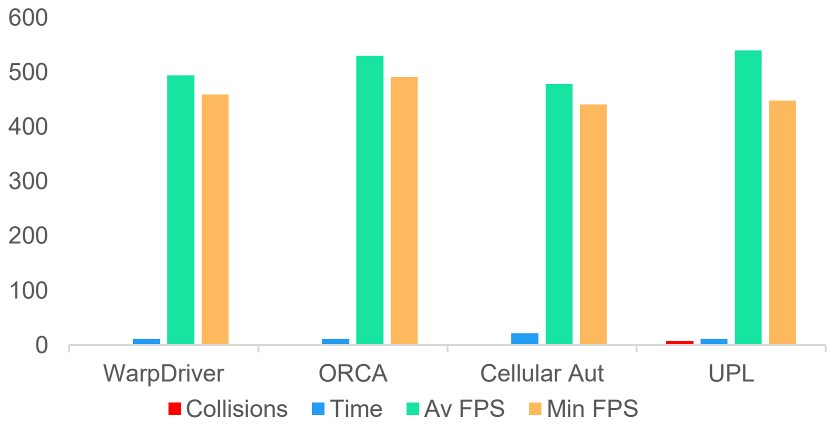
Algorithm evaluation

Evaluation Metrics

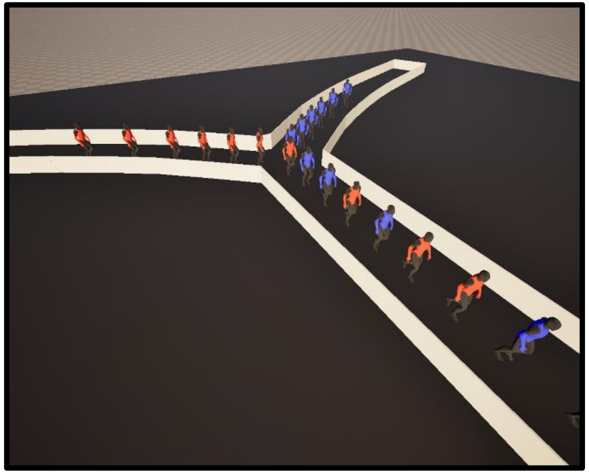
- Collision Count
- Simulation Completion Time
- Computational Cost

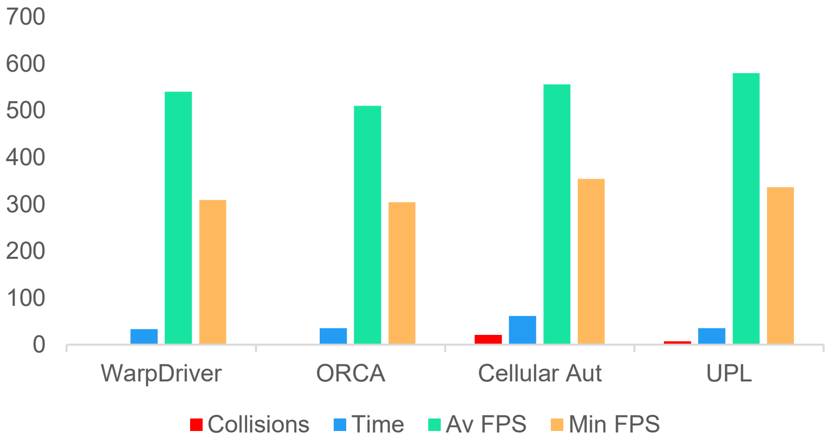
Circle



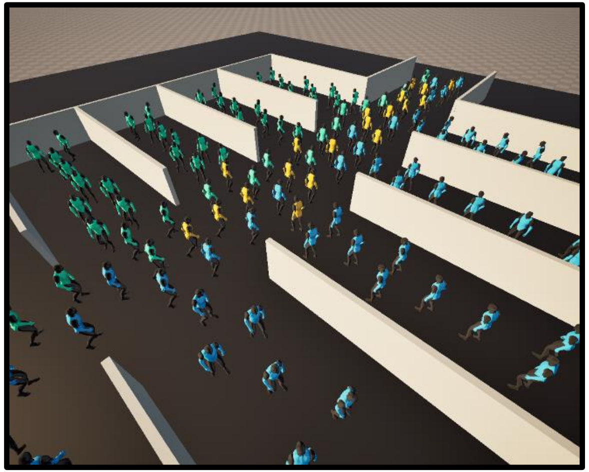

600
500
400
300
200
100
0
WarpDriver
ORCA
Cellular Aut
UPL
🟥 Collisions
🟦 Time
🟩 Av FPS
🟧 Min FPS

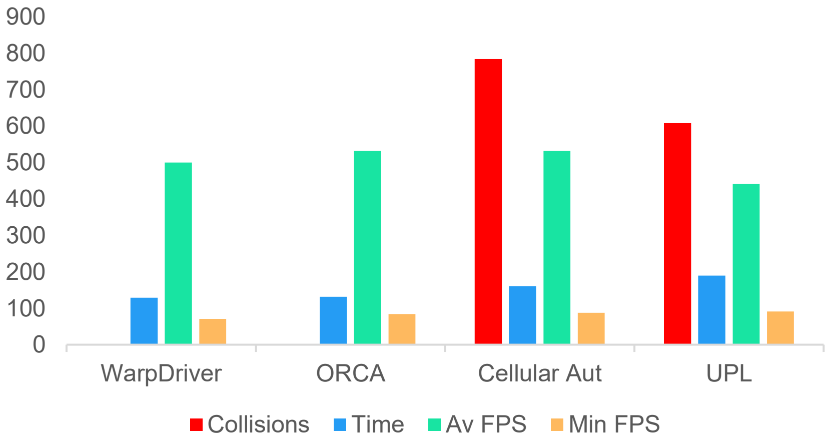
Fork Path



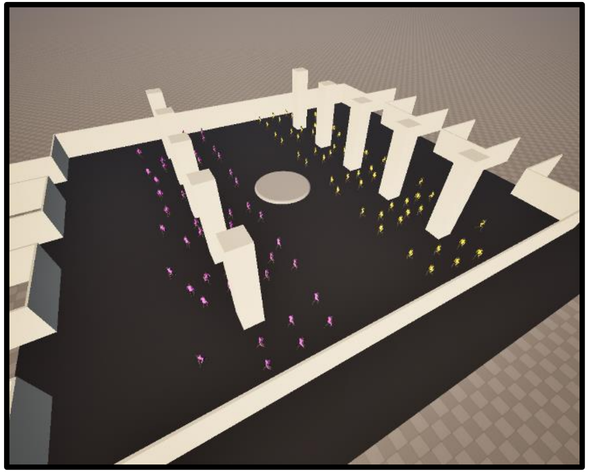

700
600
500
400
300
200
100
0
WarpDriver
ORCA
Cellular Aut
UPL
🟥 Collisions
🟦 Time
🟩 Av FPS
🟧 Min FPS

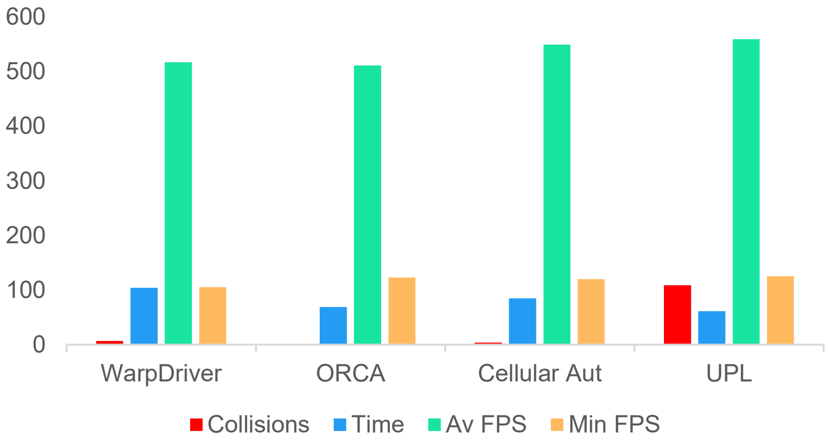
Queue



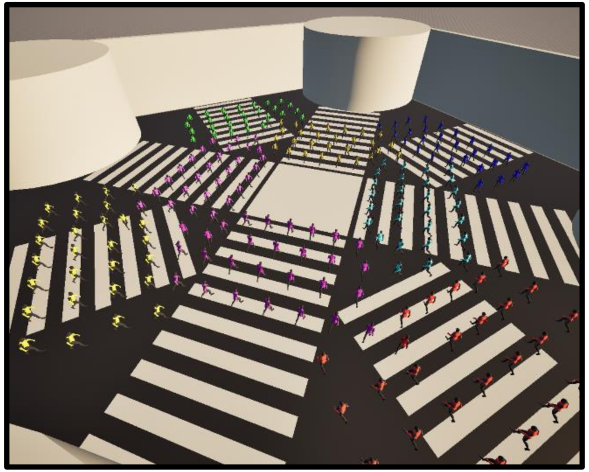

900
800
700
600
500
400
300
200
100
0
WarpDriver
ORCA
Cellular Aut
UPL
🟥 Collisions
🟦 Time
🟩 Av FPS
🟧 Min FPS

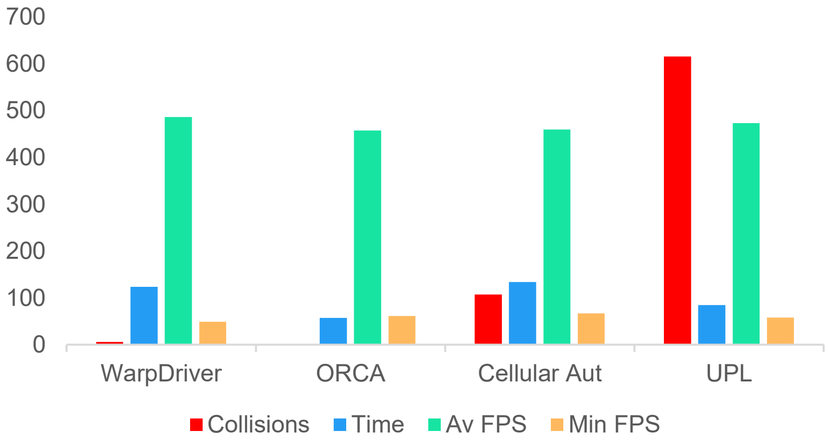
Hall



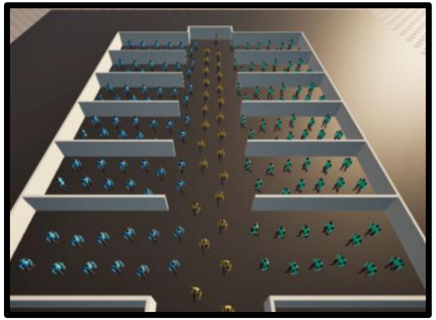

600
500
400
300
200
100
0
WarpDriver
ORCA
Cellular Aut
UPL
🟥 Collisions
🟦 Time
🟩 Av FPS
🟧 Min FPS

Shibuya crossing



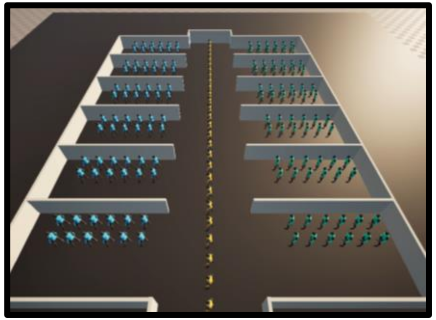


Conclusions



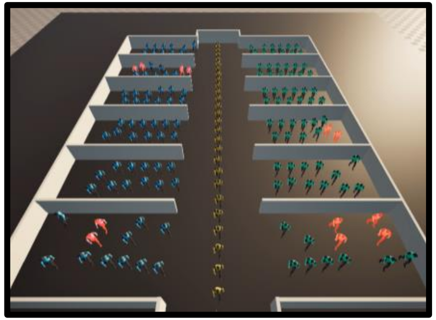
Warpdriver

- ⊕ Natural movement
- ⊕ Few collisions



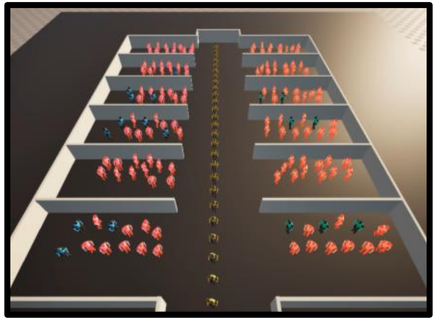
Orca

- ⊕ No collisions
- ⊕ Agents arrive fast at destination



Universal Power Law

- ⊕ Cheap to compute
- ⊖ Many collisions in dense areas



Cellular Automata

- ⊕ Cheap to compute
- ⊖ Many collisions in small areas

Thank you!