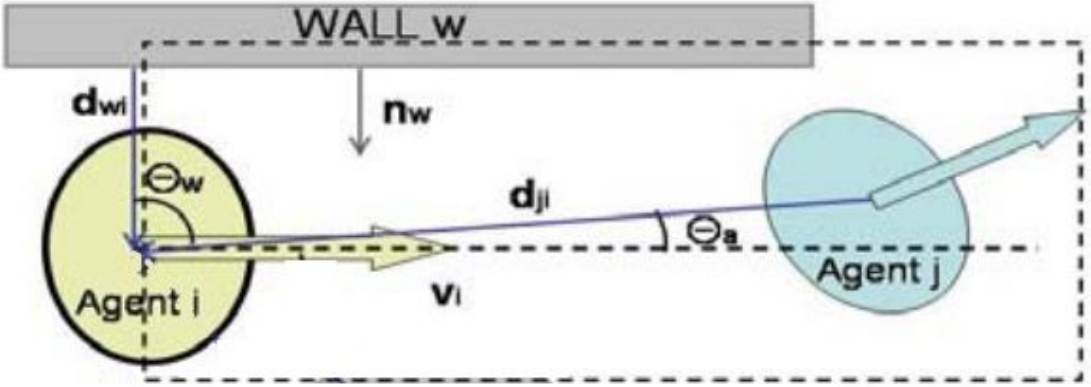
Crowd simulation

- simulation of large group of characters with respect to their individuality

- simulate crowd behaviour (in an emergency, specific situation)
- movie (battle scenes, virtual cities)
- computer game (agents, NPC)

Crowd simulation

- complex psychological model
  - *high-level contro*l - pathfinding, decision making, communications with other agents
  - *low-level control* - motion, collision avoidance, social forces, perception
    - controls basic movement
      - social forces
        - **attractive**
        - **repulsive**



$$\mathbf{F}_i^{To}[n] = \mathbf{F}_i^{To}[n-1] + \mathbf{F}_i^{At}[n]w_i^{At} + \sum_w \mathbf{F}_{wi}^{Wa} + \sum_k \mathbf{F}_{ki}^{Ob}[n]w_i^{Ob} + \sum_{j \neq i} \mathbf{F}_{ji}^{Ot}[n]w_i^{Ot}$$

new direction

force from the previous frame

direction toward a certain goal

walls

obstacles

other agents

forces from surroundings

$$\mathbf{p}_i[n+1] = \mathbf{p}_i[n] + \alpha_i[n]v_i[n]((1-\beta_i[n])\mathbf{f}_i^{To} + \beta_i[n]\mathbf{F}_i^{Fa}[n])T + \mathbf{r}_i[n]$$

$$\mathbf{F}_i^{To}[n] = \mathbf{F}_i^{To}[n-1] + \mathbf{F}_i^{At}[n]w_i^{At} + \sum_w \mathbf{F}_{wi}^{Wa} + \sum_k \mathbf{F}_{ki}^{Ob}[n]w_i^{Ob} + \sum_{j \neq i} \mathbf{F}_{ji}^{Ot}[n]w_i^{Ot}$$

new direction

force from the previous frame

direction toward a certain goal

walls

obstacles

other agents

forces from surroundings

current velocity of agent

Beta is 0 or 0.5 (falled agent will be stepped over or avoided)

$$\mathbf{p}_i[n+1] = \mathbf{p}_i[n] + \alpha_i[n]v_i[n]((1-\beta_i[n])\mathbf{f}_i^{To} + \beta_i[n]\mathbf{F}_i^{Fa}[n])T + \mathbf{r}_i[n]$$

new position

position from the previous frame

Alpha is 0 or 1 (if agent is moving)

fallen agent avoidance force

repulsion force (overlapping position)

avoidance force
for agent *i* and wall *w*

$$\mathbf{F}_{wi}^{Wa} = \frac{(\mathbf{n}_w \times \mathbf{v}_i) \times \mathbf{n}_w}{|(\mathbf{n}_w \times \mathbf{v}_i) \times \mathbf{n}_w|}$$

$$\mathbf{F}_i^{To}[n] = \mathbf{F}_i^{To}[n-1] + \mathbf{F}_i^{At}[n] w_i^{At} + \sum_w \mathbf{F}_{wi}^{Wa} + \cancel{\sum_k \mathbf{F}_i^{Ob}[n] w_i^{Ob}} + \sum_{j \neq i} \mathbf{F}_{ji}^{Ot}[n] w_i^{Ot}$$

new direction

force from the previous frame

direction toward a certain goal

walls

obstacles

other agents

forces from surroundings

current velocity of agent

Beta is 0 or 0.5 (falled agent will be stepped over or avoided)

$$\mathbf{p}_i[n+1] = \mathbf{p}_i[n] + \alpha_i[n] v_i[n] ((1 - \cancel{\beta_i[n]}) \mathbf{f}_i^{To} + \cancel{\beta_i[n] \mathbf{F}_i^{Fa}[n]}) T + \mathbf{r}_i[n]$$

new position

position from the previous frame

Alpha is 0 or 1 (agent is moving or is pushed away by other)

new direction force

fallen agent avoidance force

repulsion force (overlapping position)

avoidance force
for agent *i* and agent *j*

avoidance force
for agent *i* and wall *w*

$$\mathbf{F}_i^{To}[n] = \mathbf{F}_i^{To}[n-1] + \mathbf{F}_i^{At}[n]w_i^{At} + \sum_w \mathbf{F}_{wi}^{Wa} + \sum_{j \neq i} \mathbf{F}_{ji}^{Ot}[n]w_i^{Ot}$$

$$\mathbf{p}_i[n+1] = \mathbf{p}_i[n] + \alpha_i[n]v_i[n]\mathbf{f}_i^{To}T + \mathbf{r}_i[n].$$

$$\mathbf{r}_i[n] = \sum_w \mathbf{F}_{wi}^{R\_Wa} + \sum_k \mathbf{F}_{ki}^{R\_Ob}[n] + \lambda \sum_{j \neq i} \mathbf{F}_{ji}^{R\_Ot}[n]$$

avoidance force
for agent *i* and agent *j*

avoidance force
for agent *i* and wall *w*

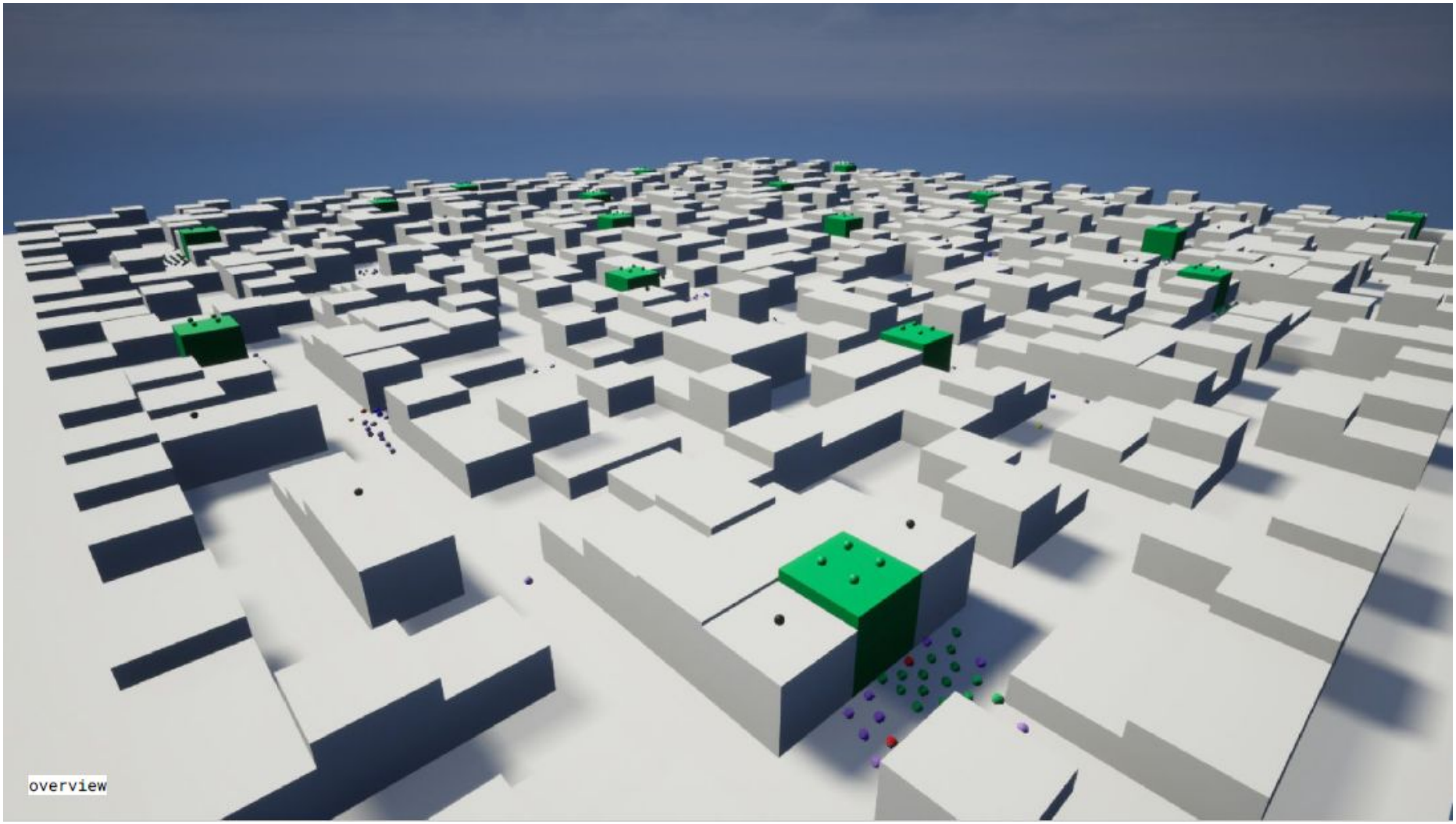
$$\mathbf{F}_i^{To}[n] = \mathbf{F}_i^{To}[n-1] + \mathbf{F}_i^{At}[n]w_i^{At} + \sum_w \mathbf{F}_{wi}^{Wa} + \sum_{j \neq i} \mathbf{F}_{ji}^{Ot}[n]w_i^{Ot}$$

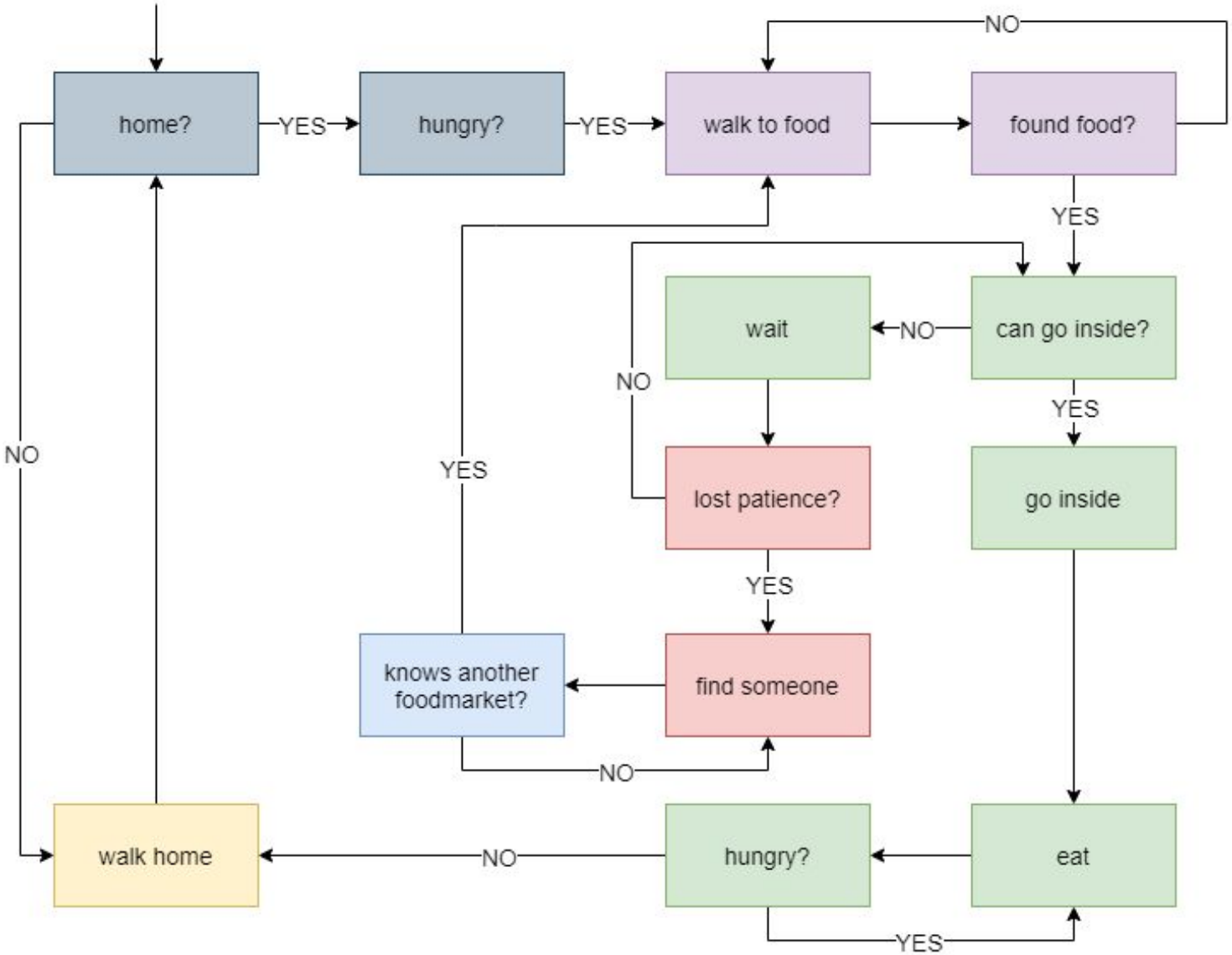
$$\mathbf{p}_i[n+1] = \mathbf{p}_i[n] + \alpha_i[n]v_i[n]\mathbf{f}_i^{To}T + \mathbf{r}_i[n].$$

$$\mathbf{r}_i[n] = \sum_w \mathbf{F}_{wi}^{R\_Wa} + \cancel{\sum_k \mathbf{F}_{ki}^{R\_Ot}[n]} + \lambda \sum_{j \neq i} \mathbf{F}_{ji}^{R\_Ot}[n]$$

ČVUT

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE




home?
YES
hungry?
YES
walk to food
found food?
NO
YES
can go inside?
NO
wait
YES
go inside
NO
lost patience?
YES
find someone
knows another foodmarket?
NO
YES
NO
walk home
NO
hungry?
YES
eat

breadth-first search (BFS) - O(V+E), finding the shortest path in unweighted graph

at home
going to food market
waiting to get in
eating
lost patience, searching for others
discussing alternatives
going home

```
FVector Agent::otherAgentsRepulsionForce(std::vector<Agent*>& others) {

        return force;
}
```

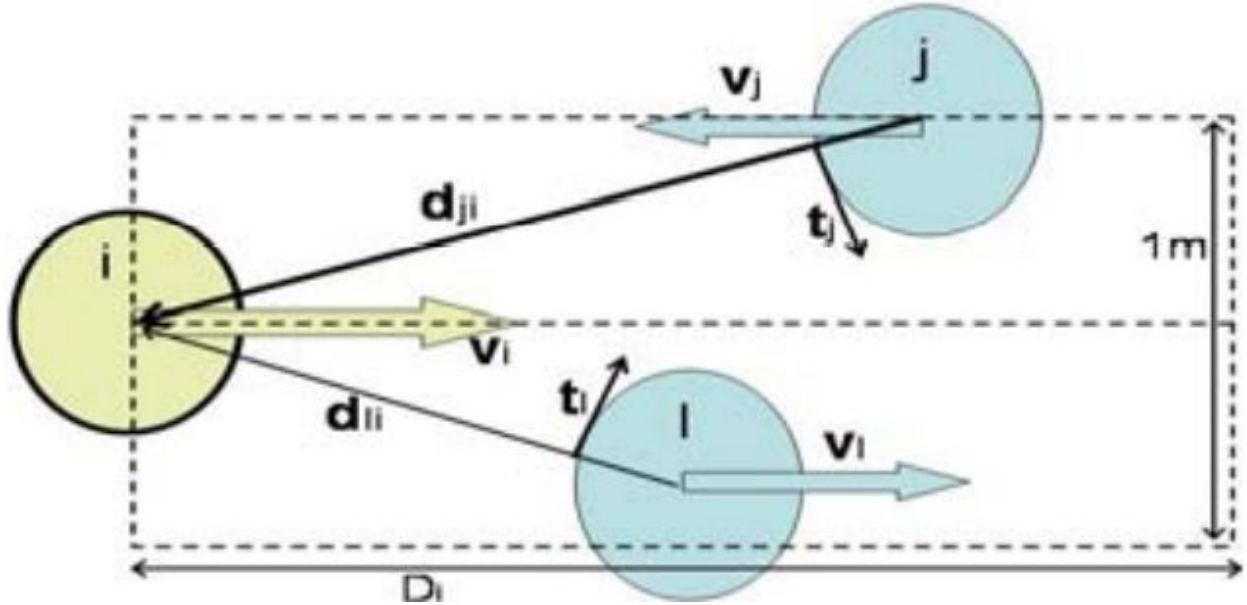
$$\mathbf{F}_{ji}^{R\_Ot}[n] = \frac{\left(\mathbf{p}_i[n] - \mathbf{p}_j[n]\right)\left(r_i + \varepsilon_i + r_j - d_{ji}[n]\right)}{d_{ji}[n]}$$

https://repository.upenn.edu/cgi/viewcontent.cgi?article=1223&context=hms

Controlling Individual Agents in High-Density Crowd Simulation *Nuria Pelechano*

```
FVector Agent::otherAgentsAvoidanceForce(std::vector<Agent*>& others) {

        return force;
}
```



$$\mathbf{t}_j = \frac{(\mathbf{d}_{ji} \times \mathbf{v}_i) \times \mathbf{d}_{ji}}{\left|(\mathbf{d}_{ji} \times \mathbf{v}_i) \times \mathbf{d}_{ji}\right|}$$

$$\mathbf{F}_{ji}^{Ot} = \mathbf{t}_j w_i^d w_i^o$$

distance weight

$$w_i^d = (d_{ji} - D_i)^2$$

orientation difference weight

$$w_i^o = \begin{cases} 1.2 & if\,(\mathbf{v}_i \cdot \mathbf{v}_j) > 0 \\ 2.4 & otherwise \end{cases}$$

```
FVector Agent::otherAgentsAvoidanceForce(std::vector<Agent*>& others) {

    return force;
}
```

Agent.h
- radius
- personalSpaceEpsilon
- vel

$$\mathbf{t}_j = \frac{(\mathbf{d}_{ji} \times \mathbf{v}_i) \times \mathbf{d}_{ji}}{\left|(\mathbf{d}_{ji} \times \mathbf{v}_i) \times \mathbf{d}_{ji}\right|}$$

GetSafeNormal()

$$\mathbf{F}_{ji}^{Ot} = \mathbf{t}_j w_i^d w_i^o$$

orientation difference weight

distance weight

$$w_i^d = (d_{ji} - D_i)^2$$

$$w_i^o = \begin{cases} 1.2 & if\,(\mathbf{v}_i \cdot \mathbf{v}_j) > 0 \\ 2.4 & otherwise \end{cases}$$

- N. Pelechano, J. M. Allbeck, and N. I. Badler. Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’07, page 99–108, Goslar, DEU, 2007. Eurographics Association.

- Emese Szabó, Vojtěch Tomas *Crowd simulation during a pandemic*, MMA semester project December 2020