DH2323 DGI18

INTRODUCTION TO COMPUTER GRAPHICS AND INTERACTION

ANIMATION

Christopher Peters

CST, KTH Royal Institute of Technology, Sweden

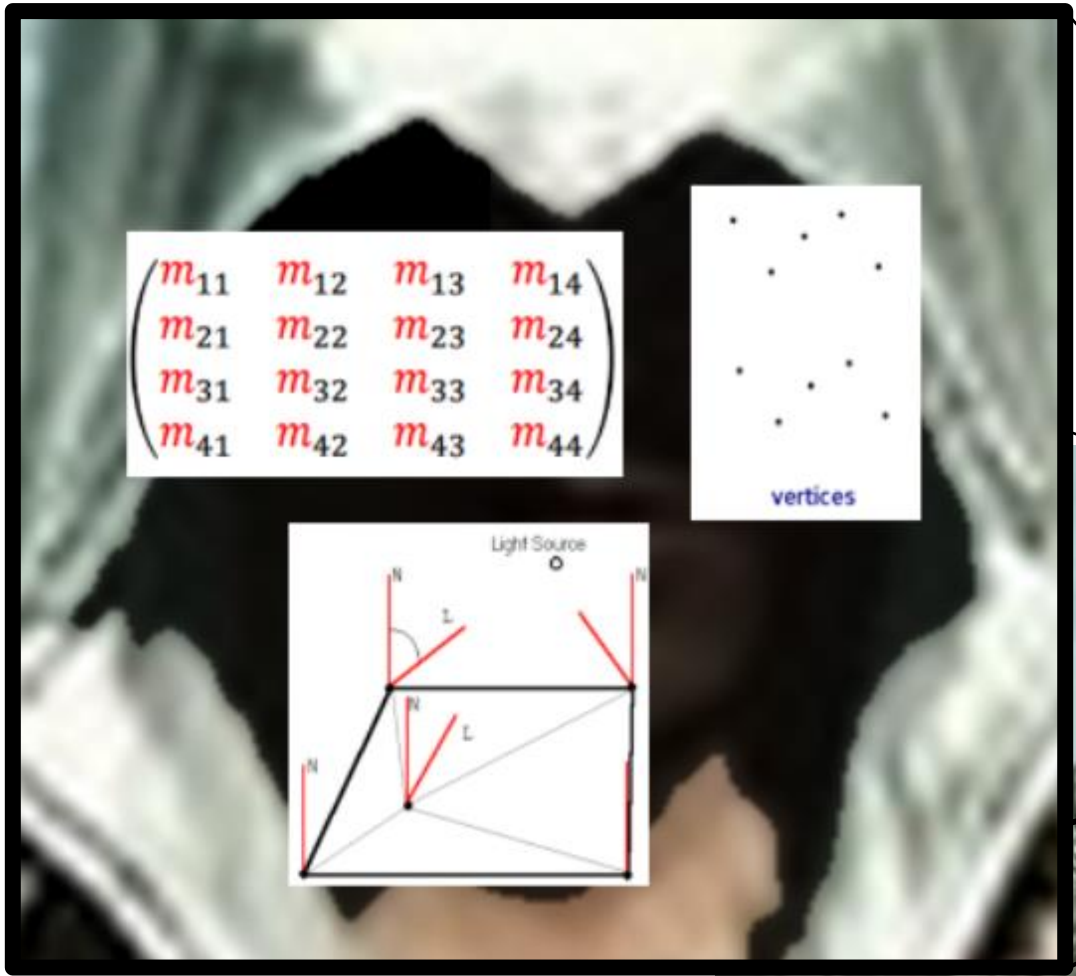
chpeters@kth.se

http://kth.academia.edu/ChristopherEdwardPeters

ROYAL INSTITUTE OF TECHNOLOGY

Underneath the CG hood

Mesh data (vertices, etc)
Positioning (transformations)
Lighting and shading



Assassin's Creed II

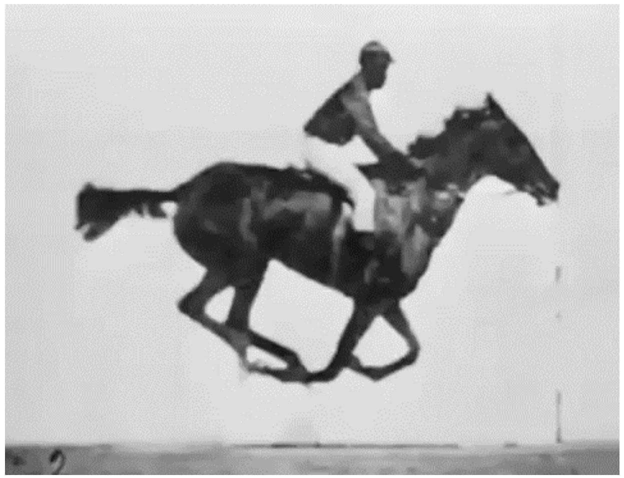
Animation

Showing consecutive related static images one after another produces the perception of a moving image

Eadweard Muybridge

Traditional Animation: master artists draw important **key-frames** in the animation

Apprentices draw the multitude of frames in-between these key-frames (***tweens)***



ROYAL INSTITUTE OF TECHNOLOGY

Computer Animation

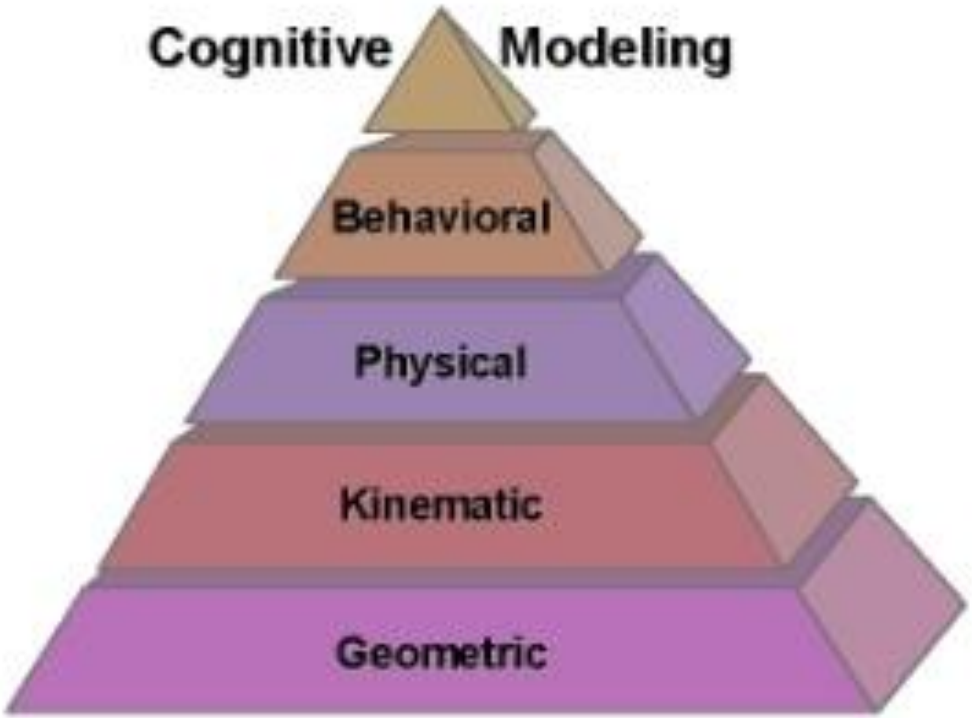


Figure 1: Cognitive modeling is the new apex of the CG modeling hierarchy.

From : Cognitive Modelling: Knowledge Reasoning and Planning for Intelligent Characters, Funge, Tu and Terzopoulos

Computer Animation

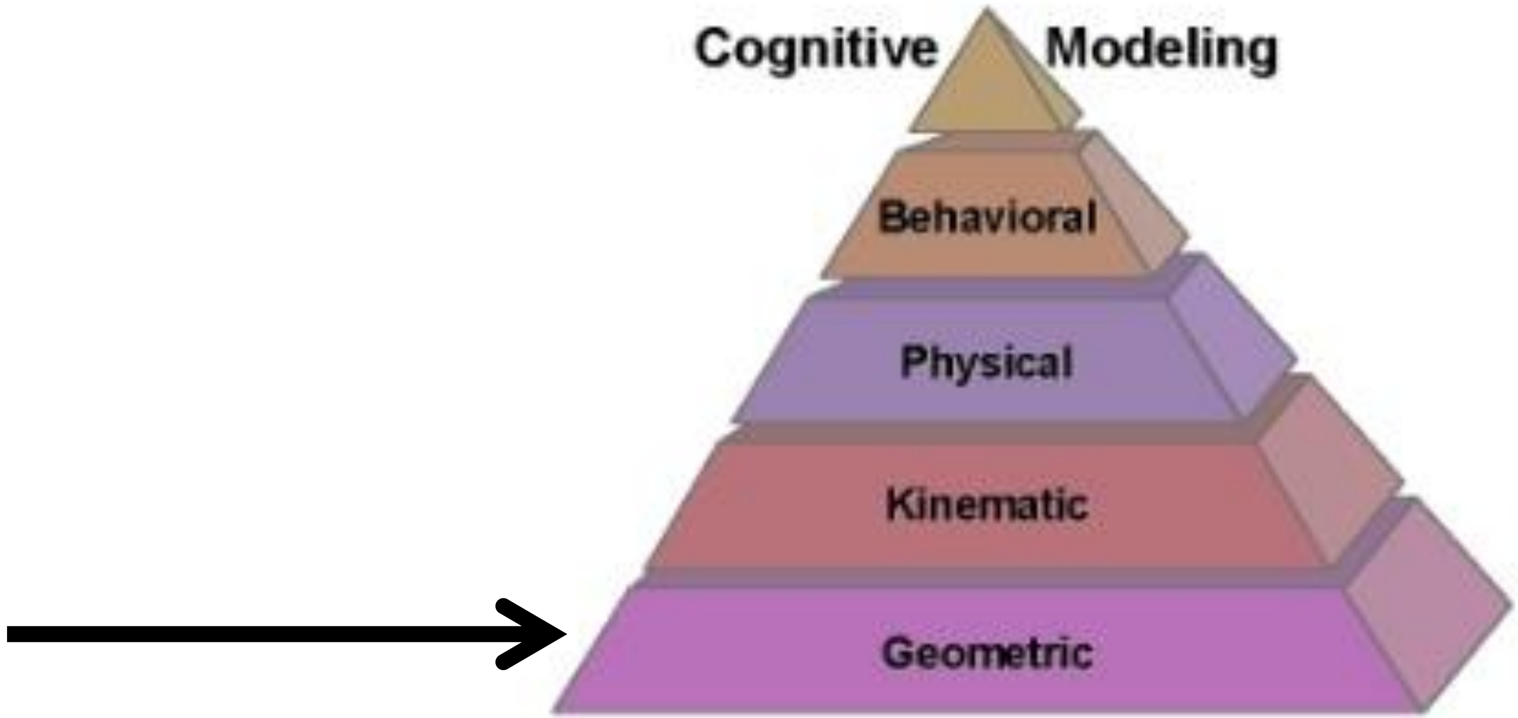
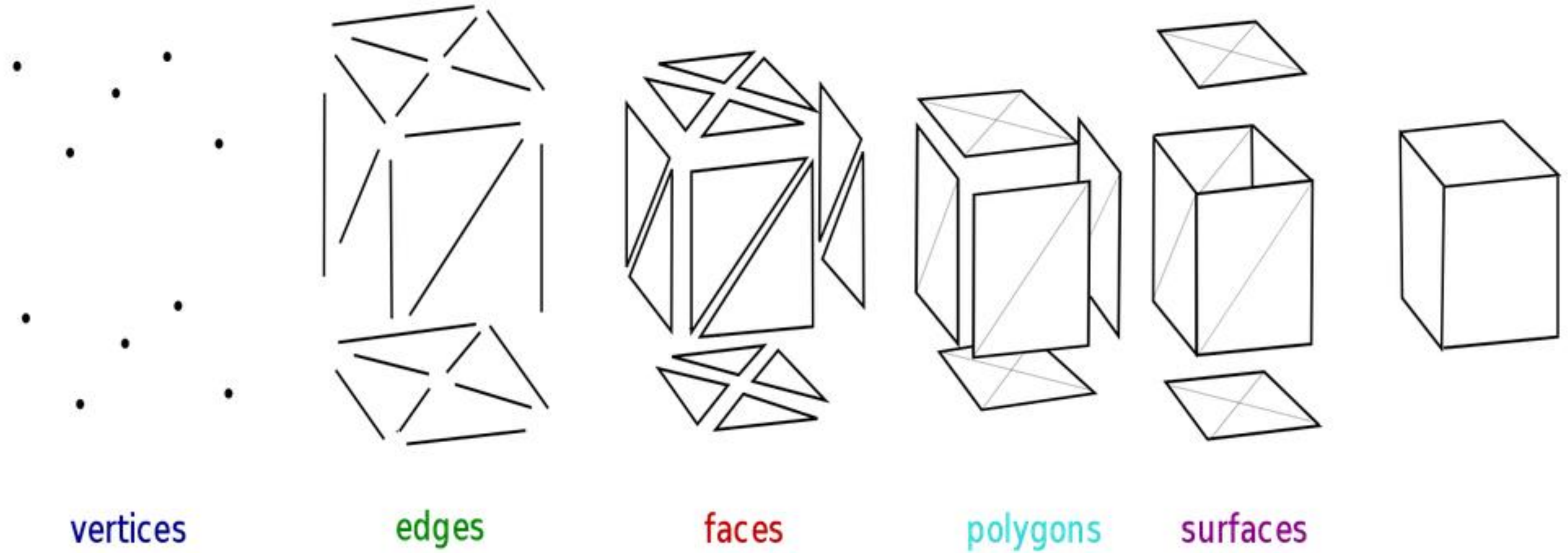


Figure 1: Cognitive modeling is the new apex of the CG modeling hierarchy.

From : Cognitive Modelling: Knowledge Reasoning and Planning for Intelligent Characters, Funge, Tu and Terzopoulos

Object Decomposition

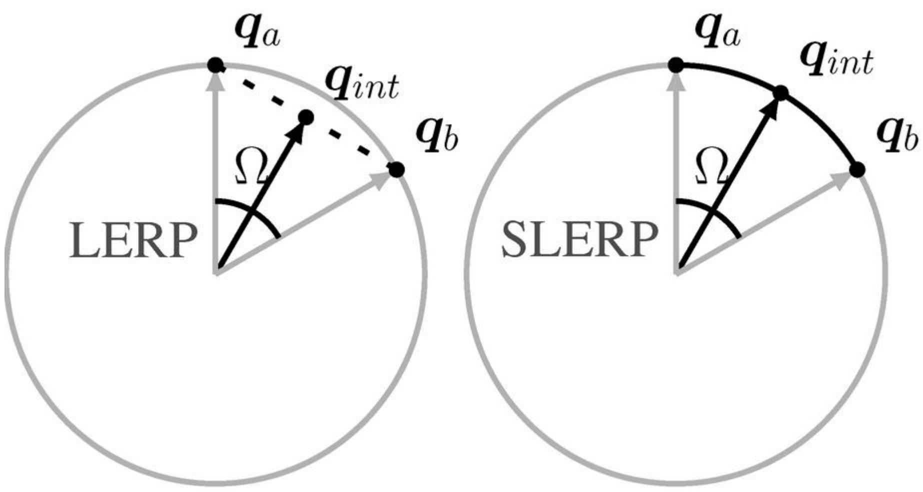


Computer Animation

Object has an initial configuration and a final configuration

Specified by the human **or by an algorithm**

Position and orientation *interpolation*



Valenti et al., Sensors 2015

Computer calculates intermediate positions/orientations

Computer Animation

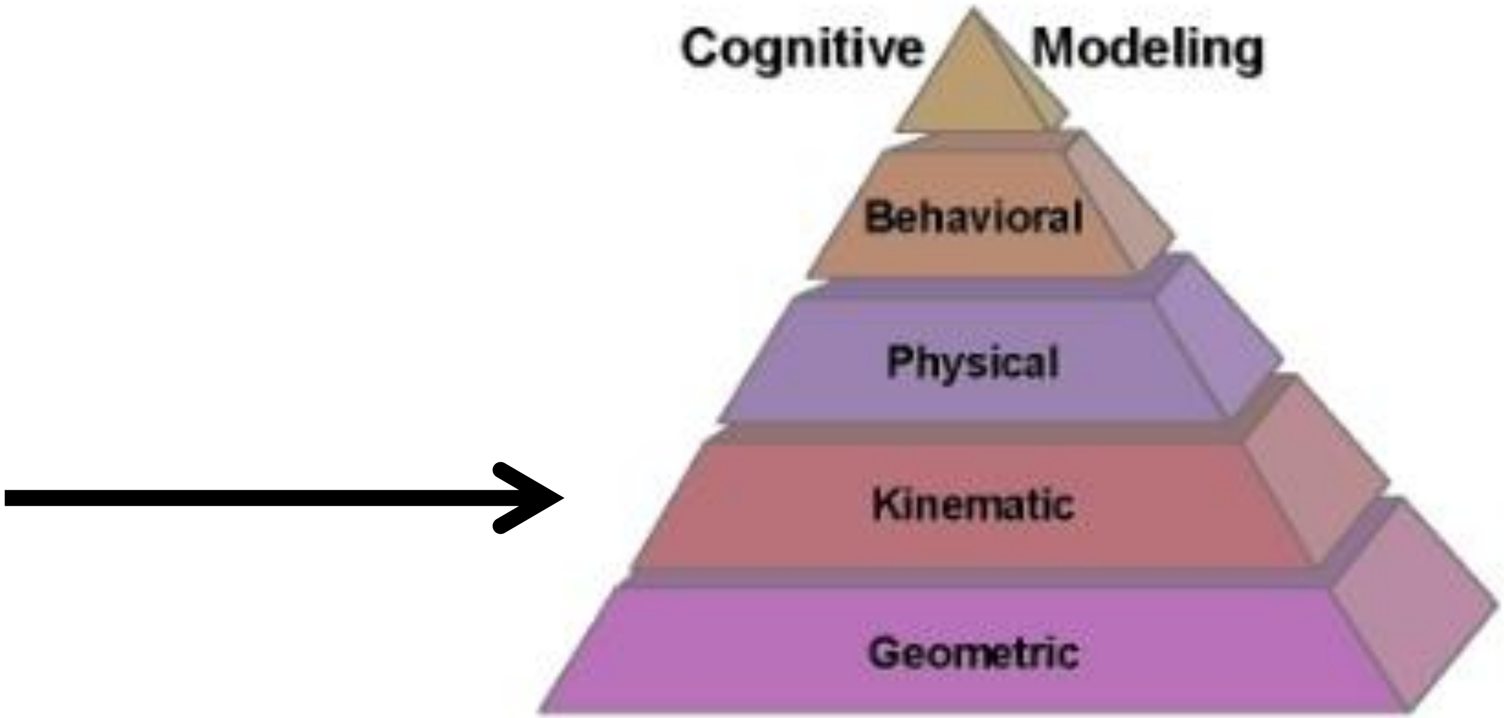
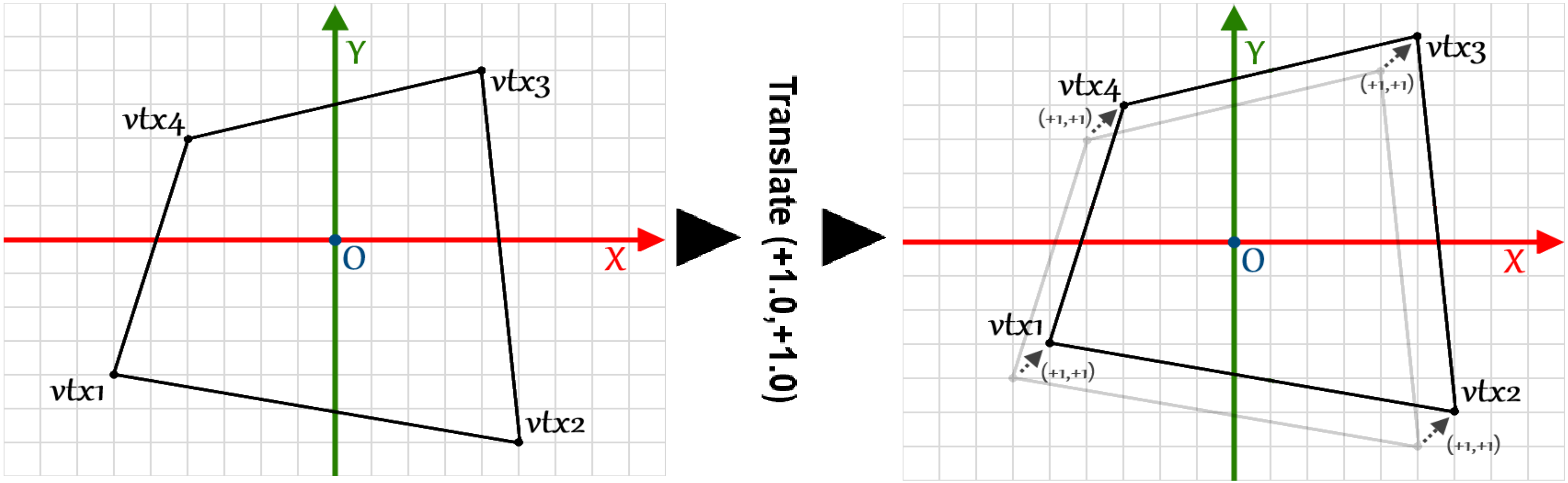


Figure 1: Cognitive modeling is the new apex of the CG modeling hierarchy.

From : Cognitive Modelling: Knowledge Reasoning and Planning for Intelligent Characters, Funge, Tu and Terzopoulos

Translating an object

Translation operation takes place on ***a point (vertex)***
But a geometric object (*mesh*) is a *collection* of vertices
How to translate that?
=> Translate each of its vertices



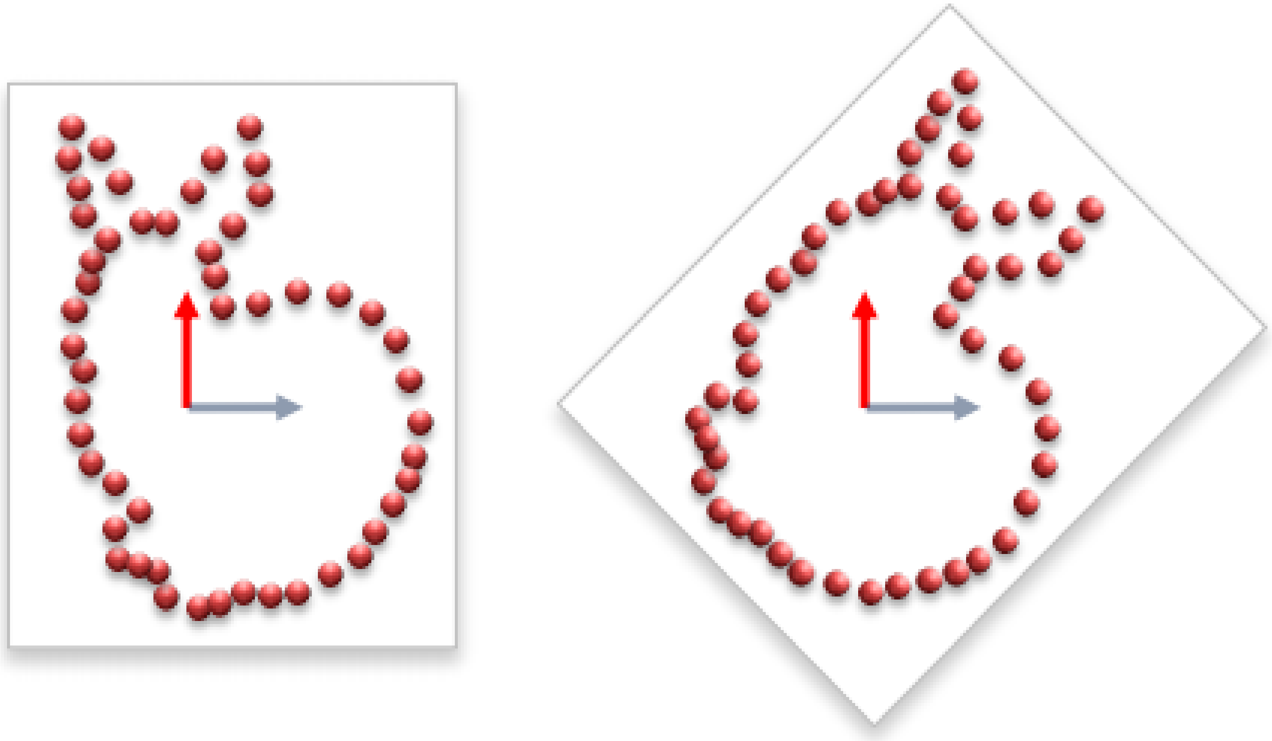
Rotating an object

Rotation operation takes place on a point

How to rotate a object?

The same procedure applies:

Rotate *each vertex* that comprises the object



Representation

Transformations are represented as 4x4 *matrices*

Translation

$$\mathbf{T}(t_x, t_y, t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation around *x*-axis

$$\mathbf{R}_x(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation around *y*-axis

$$\mathbf{R}_y(\phi) = \begin{pmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation around *z*-axis

$$\mathbf{R}_z(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{M} \cdot \mathbf{x} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix}$$

Question

How do we interpolate orientations?

Remember how they are represented

1) Rotations around axes

2) Rotation matrices

Possible solutions: Euler angles and rotation matrix interpolation

ROYAL INSTITUTE OF TECHNOLOGY

#1: Euler Angles

- An Euler angle is a rotation around a single axis
- Any orientation can be specified by composing three rotations
  - Each rotation is around one of the **principle axes**
  - i.e. (x, y, z) – first rotate around x, then y, then z
  - Think of roll, pitch and yaw of a flight simulator
- When rotating about a single axis, is possible to interpolate a single value
  - However, for more than one axis, interpolating individual angles will not work well
  - Unpredictable intermediate rotations
  - Gimbal lock

#2: Rotation Matrices

Interpolating between two rotation matrices does not result in a rotation matrix

- Does not preserve rigidity of angles and lengths
- This result of an interpolation of 0.5 between the identity matrix and 90 degrees around the x-axis does not produce a valid rotation matrix:

$$\text{Interpolate}\left(\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}, \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{vmatrix} \right) \equiv \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & -0.5 & 0.5 \end{vmatrix}$$

ROYAL INSTITUTE OF TECHNOLOGY

Solution

Use *quaternion interpolation*

Quaternions don't suffer from Gimbal lock

Can be represented as 4 numbers instead of 9 of a 3x3 matrix

Trivial conversion between angle/axis representation

Interpolation between two quaternions is easy (once you know how)

Quaternion looks like this:

$q[w,(x,y,z)]$ also written $q[w,v]$ where $v = (x,y,z)$

$q = w + xi + yj + zk$

ROYAL INSTITUTE OF TECHNOLOGY

Representation

For a right-hand rotation of θ **radians** about unit vector *v*, quaternion is:

$q = (\cos(\theta/2); \mathbf{v} \sin(\theta/2))$

- Note how the 3 imaginary coordinates are noted as a vector
- Only **unit quaternions** represent rotations
  - Such a quaternion describes a point on the 4D unit hyper-sphere
- Important note: q and –q represent the **exact same** orientation
- Different methods for doing quaternion interpolation:

LERP, SLERP (Spherical linear interpolation)

In Practice

- Not always the best choice
  - Quaternions are (as you will have noticed) hard to visualise and think about
  - If another method will do and is simpler, it will be a more appropriate choice
- But…
  - Extremely useful in many situations where other representations are awkward
  - Easy to use in your own programs once you have a quaternion class
  - See Animation track labs and GLM library

Computer Animation

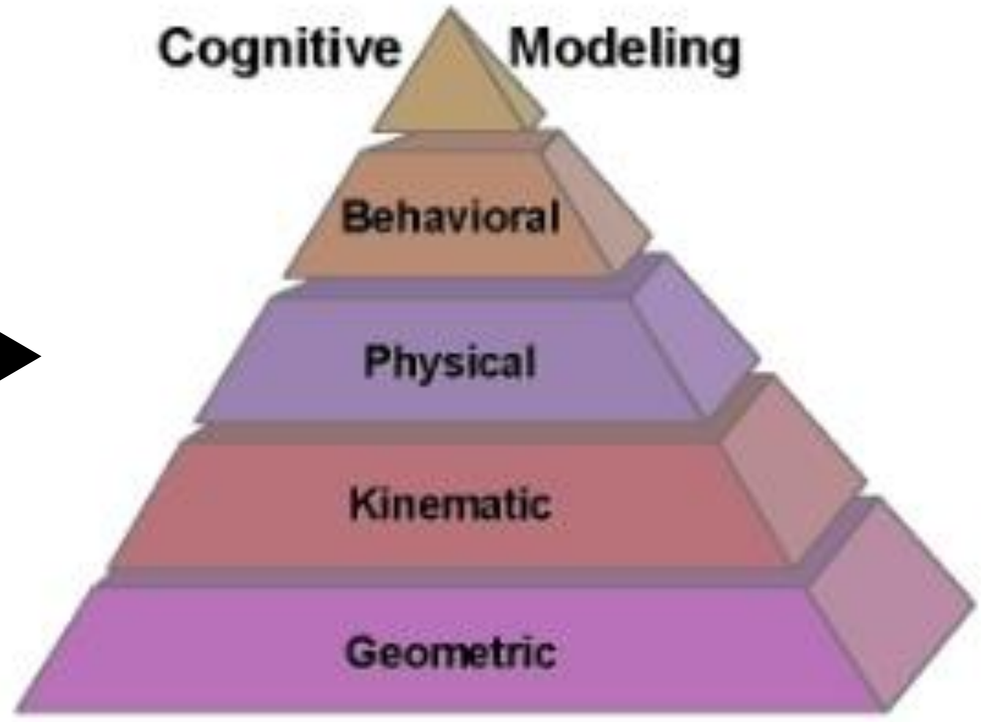


Figure 1: Cognitive modeling is the new apex of the CG modeling hierarchy.

From : Cognitive Modelling: Knowledge Reasoning and Planning for Intelligent Characters, Funge, Tu and Terzopoulos

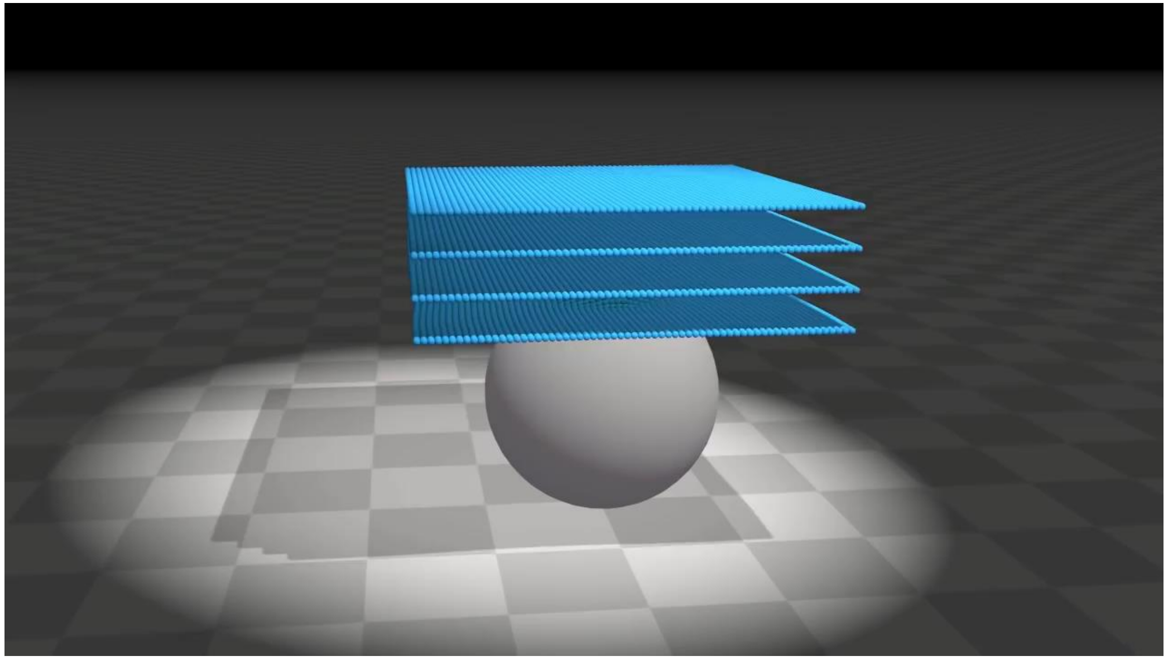
Physically-based Simulation

Manually specifying the positions and orientations of objects is cumbersome

Does not produce realistic results

Instead:

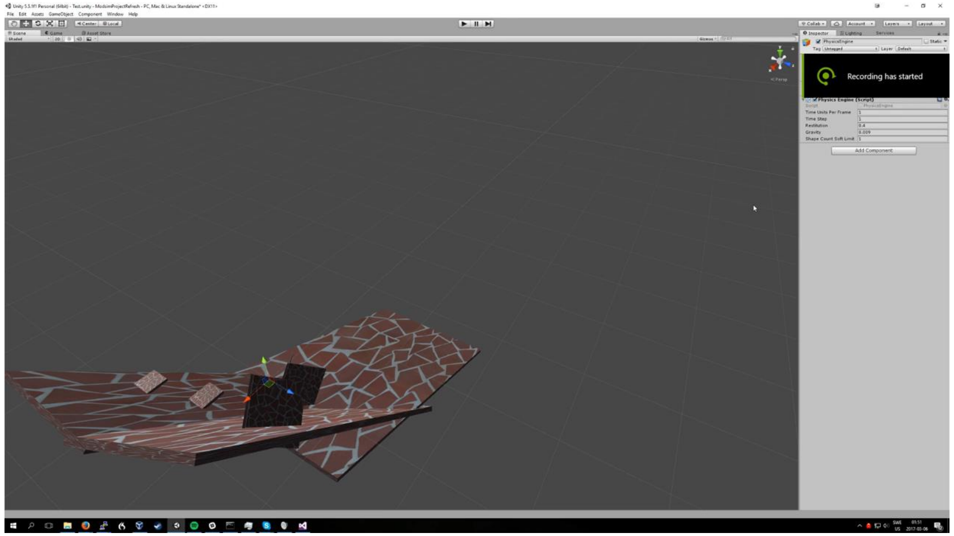
1) Define a physical simulation model
2) Give virtual object physical characteristics
3) Allow the computer to simulate their animation according to the physical laws

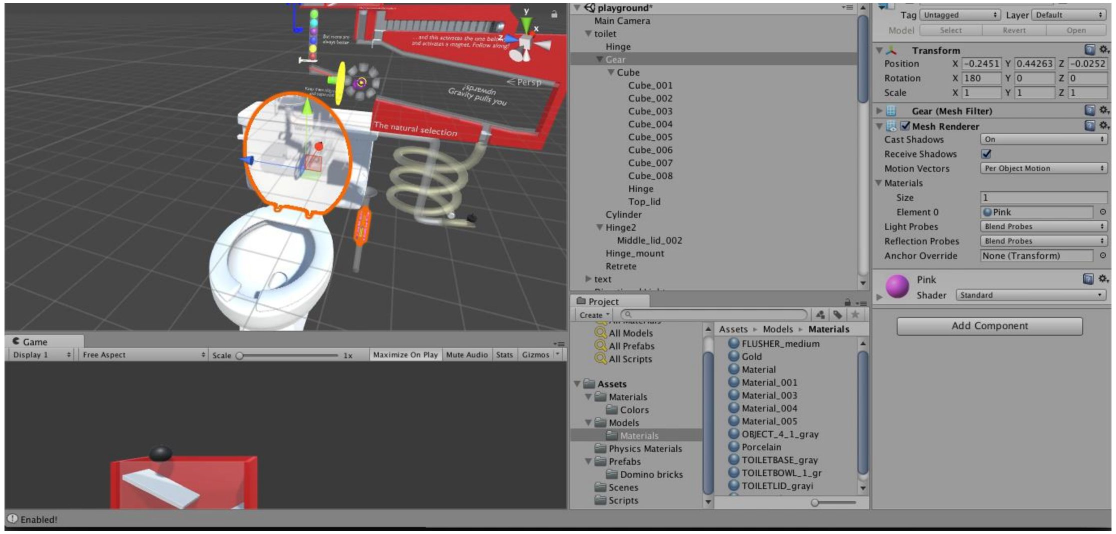


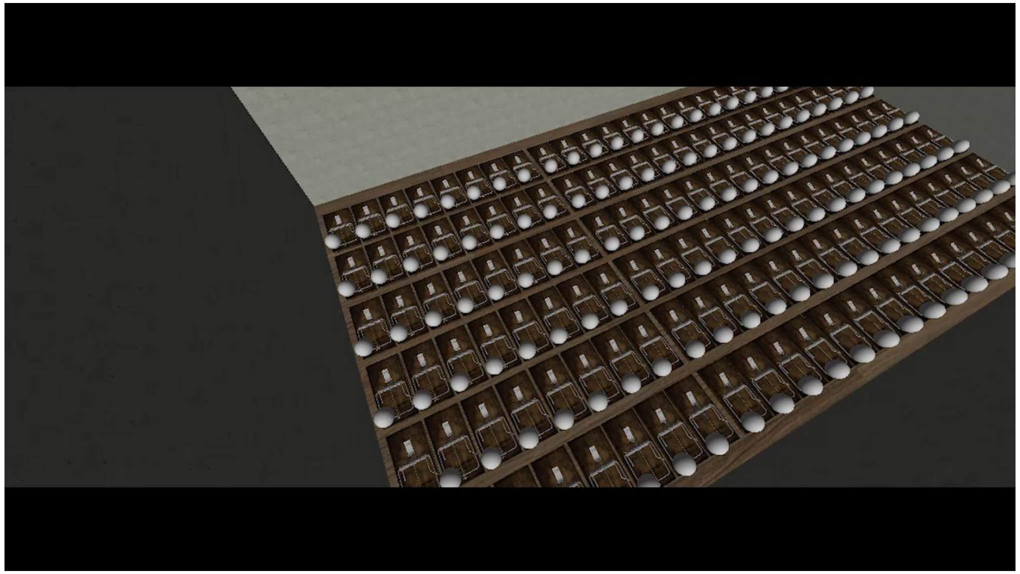
Nvidia GameWorks Unreal Engine 4

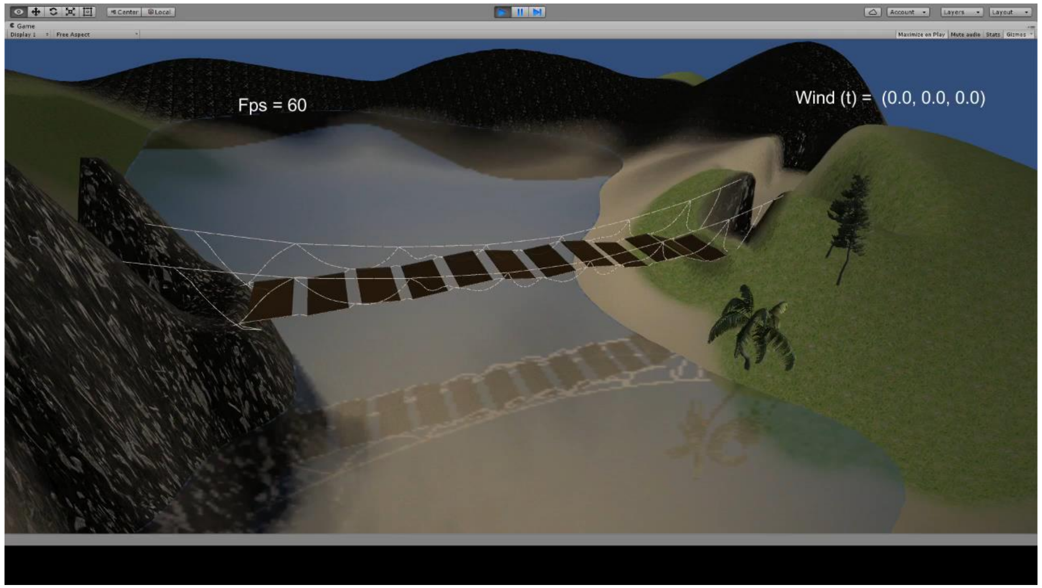
KTH
VETENSKAP OCH KONST
ROYAL INSTITUTE OF TECHNOLOGY

Physically-based Simulation



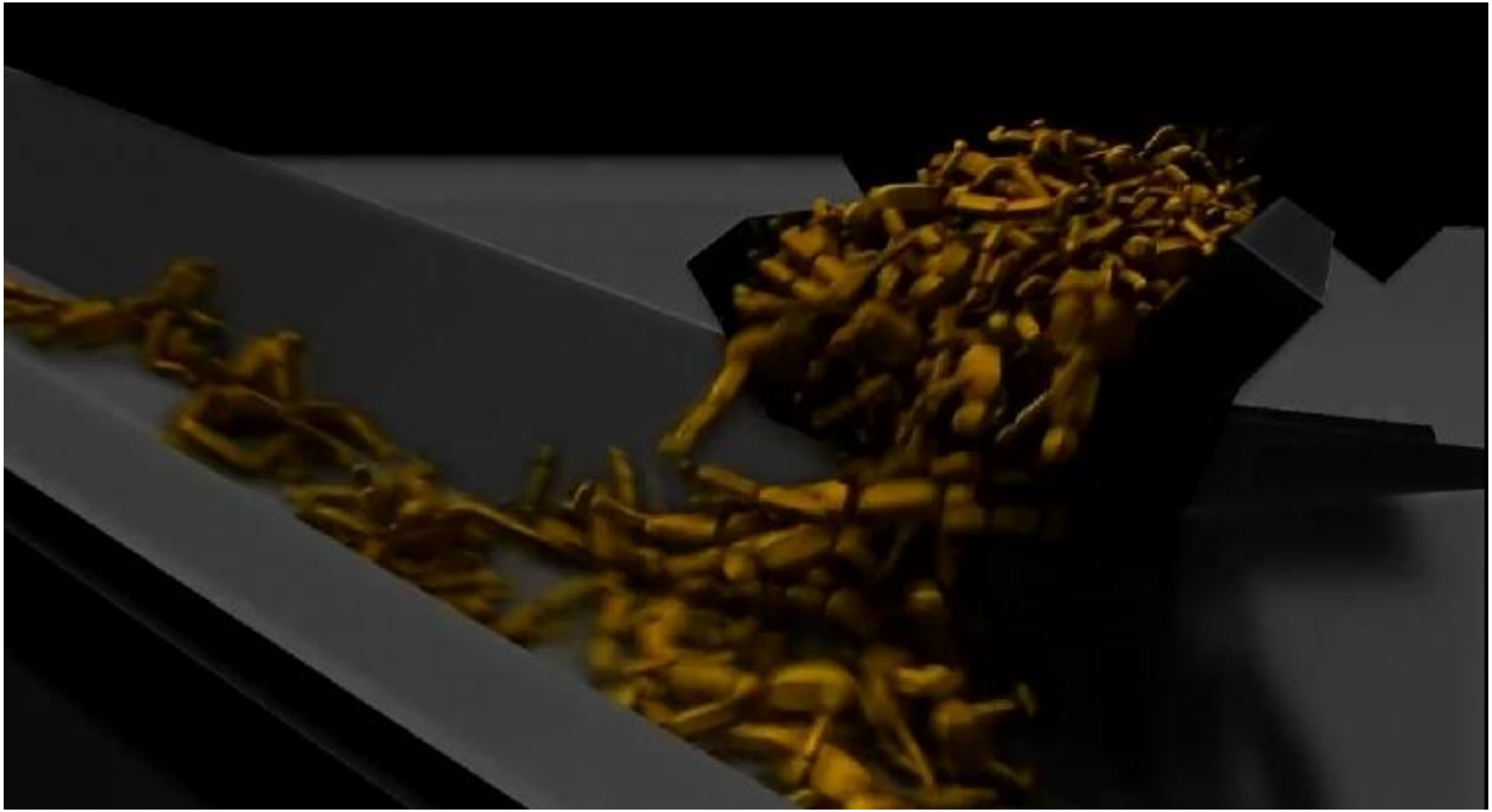






Destructible Environments

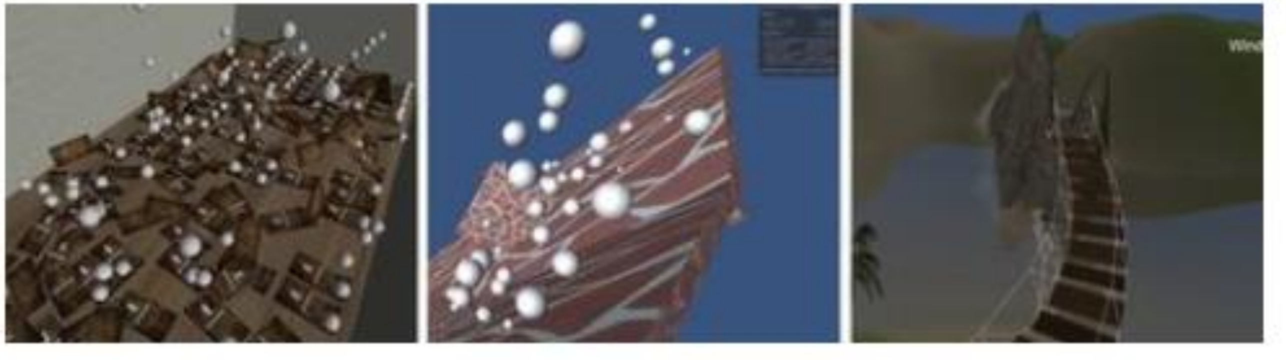
Ragdoll Physics



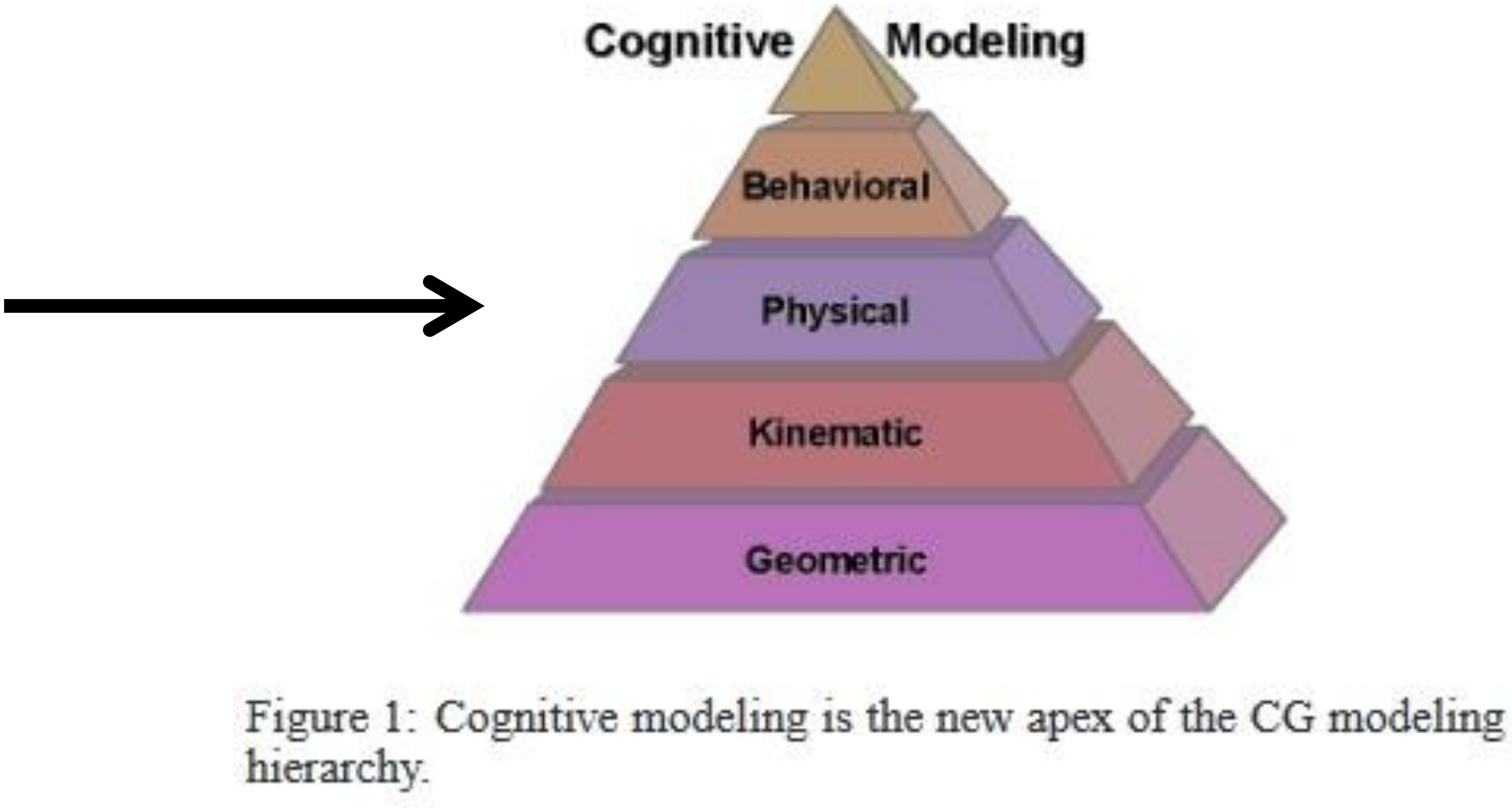
DD1354 on KTH Social

Models and Simulation

This is a course that deals with mathematical models, and numerical methods and algorithms for computer simulation. Modelling and simulation is increasingly important in science and technology, and is also used in entertainment such as physics engines for animation and computer games. Basic mathematical models as particle systems and mass-spring system are presented in the form of ordinary differential equations. The course focuses on practical aspects of methods and algorithms, and implementation of these in computer programs. The course includes a project where the methods are used to model a problem from reality, a scene, or to build a computer game.



Computer Animation



From : Cognitive Modelling: Knowledge Reasoning and Planning for Intelligent Characters, Funge, Tu and Terzopoulos

Behavioural Animation

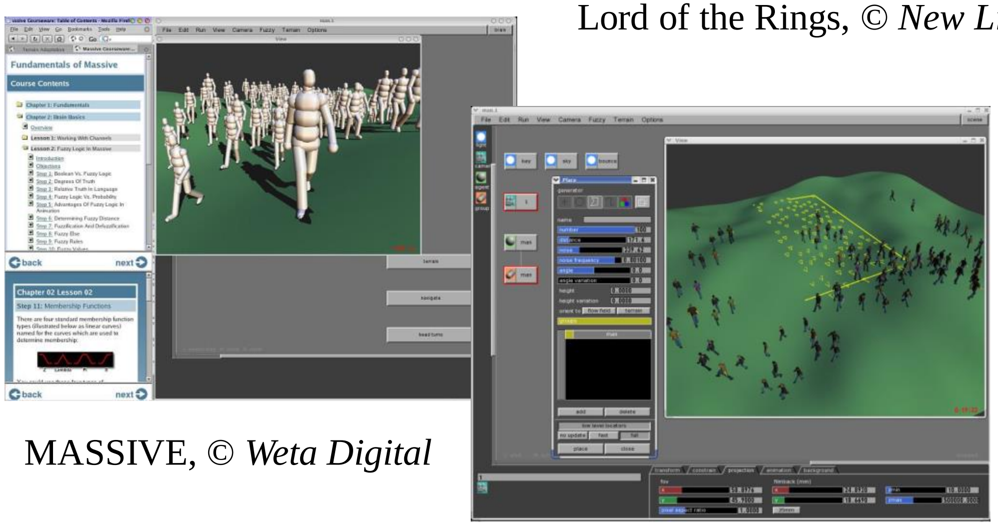
- **Control**
  - A method of controlling virtual entities
  - High-level control (director’s seat)
  - No specification of rotations and translations
- So far, similar to physical animation
  - Physical animation:
    - Objects have physical attributes and are animated according to the physical simulation
  - Behavioural animation:
    - Agents take their own actions depending on the circumstances

ROYAL INSTITUTE OF TECHNOLOGY

In Movies

- MASSIVE
  - Multiple Agent Simulation System In Virtual Environment
  - Agents decide on reaction based on the situation, using a 'brain'
  - Reactions activate key-framed or motion captured animation clips

Lord of the Rings, © *New Line Cinema*



MASSIVE, © *Weta Digital*

Important Factors

Degree of Autonomy

- Selection of potential actions that can be made in certain situations
- Agent decides *itself* which to select given the current situation

Reactive Behaviour and Sensing

- Light decision-making mechanisms
- Agent capable of sensing the external environment
- Varying degrees of sensor sophistication
  - E.g. Ray casting -> synthetic vision

Sensing

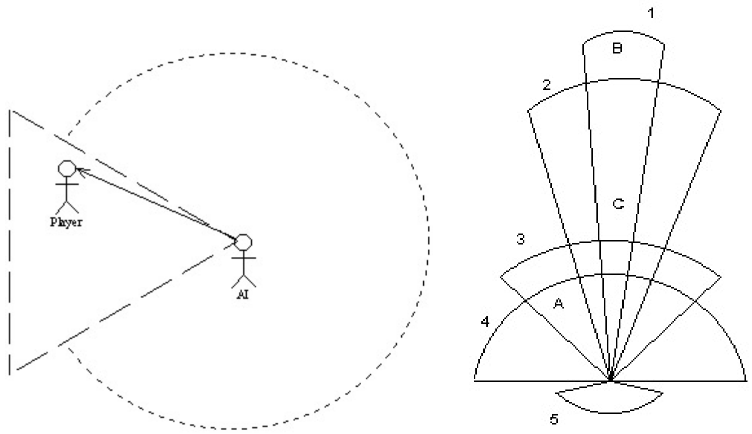
- Environment database
  - –Agents have *unrestricted* access
    - Unrealistic abilities
    - Behaviours are more similar
  - –Simplified senses

    Volume tests –anything falling within the volume is ‘sensed’

    Ray-casting – shoot rays out and check for collisions

    Synthetic vision – render scene from agent’s viewpoint

    - Not machine vision
- Active vision – orient senses



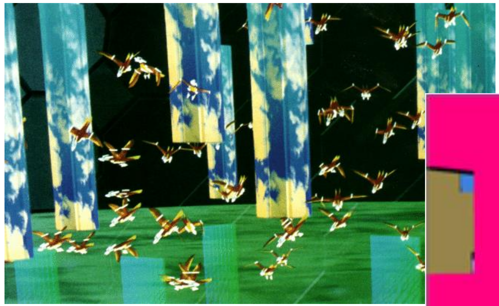
ROYAL INSTITUTE OF TECHNOLOGY

Emergent Behaviours

- Advantages
    - Simple set of rules can lead to behaviours
  - Useful for animating large crowds
    - Do not have to manually adjust the animation of each agent individually…
    - …but important to provide differentiation in the behaviour of agents so they do not all act in the exact same way
- Disadvantages
  - Hard to achieve specific desired effects
    - Agents animated only through indirect means
    - Reacting to each other and to environment
    - Many variables to consider so outcomes can be hard to predict

Examples

- Boids
- Artificial fish
- ALIVE: Silas T. Dog
- Improv
- Cognitive modelling



© Reynolds, 1987

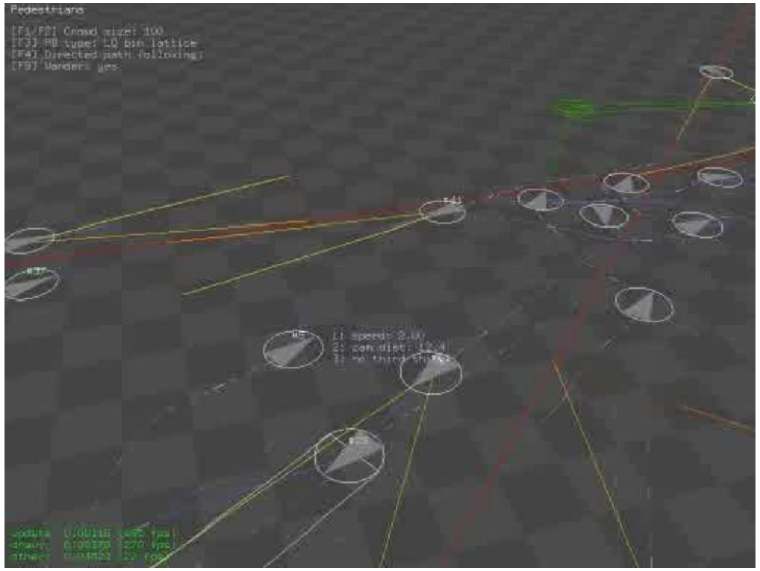
© Blumberg and Galyean, 1995

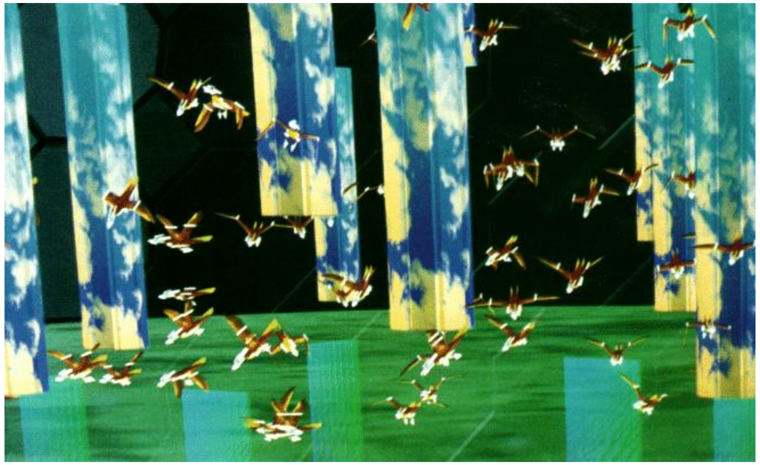
© Tu and Terzopoulos, 1994

© Funge et al. 1999

© Perlin and Goldberg, 1996

Boids

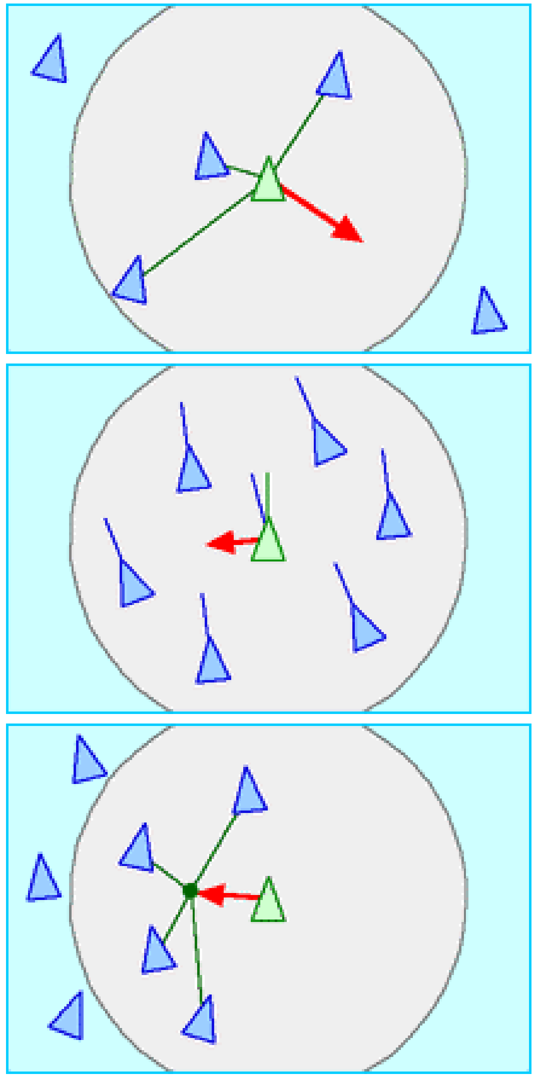




- Bird-oids
- Craig Reynolds, SIGGRAPH 1987
- Useful for modelling animal behaviour
  – Flocks, herds and so on
  - *Steering* approaches
  - See red3d.com

Boids

- Maneuver relative to positions and velocities of nearby flock-mates
- Three types of *steering* behaviours
  - Separation: avoid crowding
  - Alignment: steer towards average heading
  - Cohesion: move towards average position
- Sense only within local spherical neighbourhood



© Reynolds

Boids

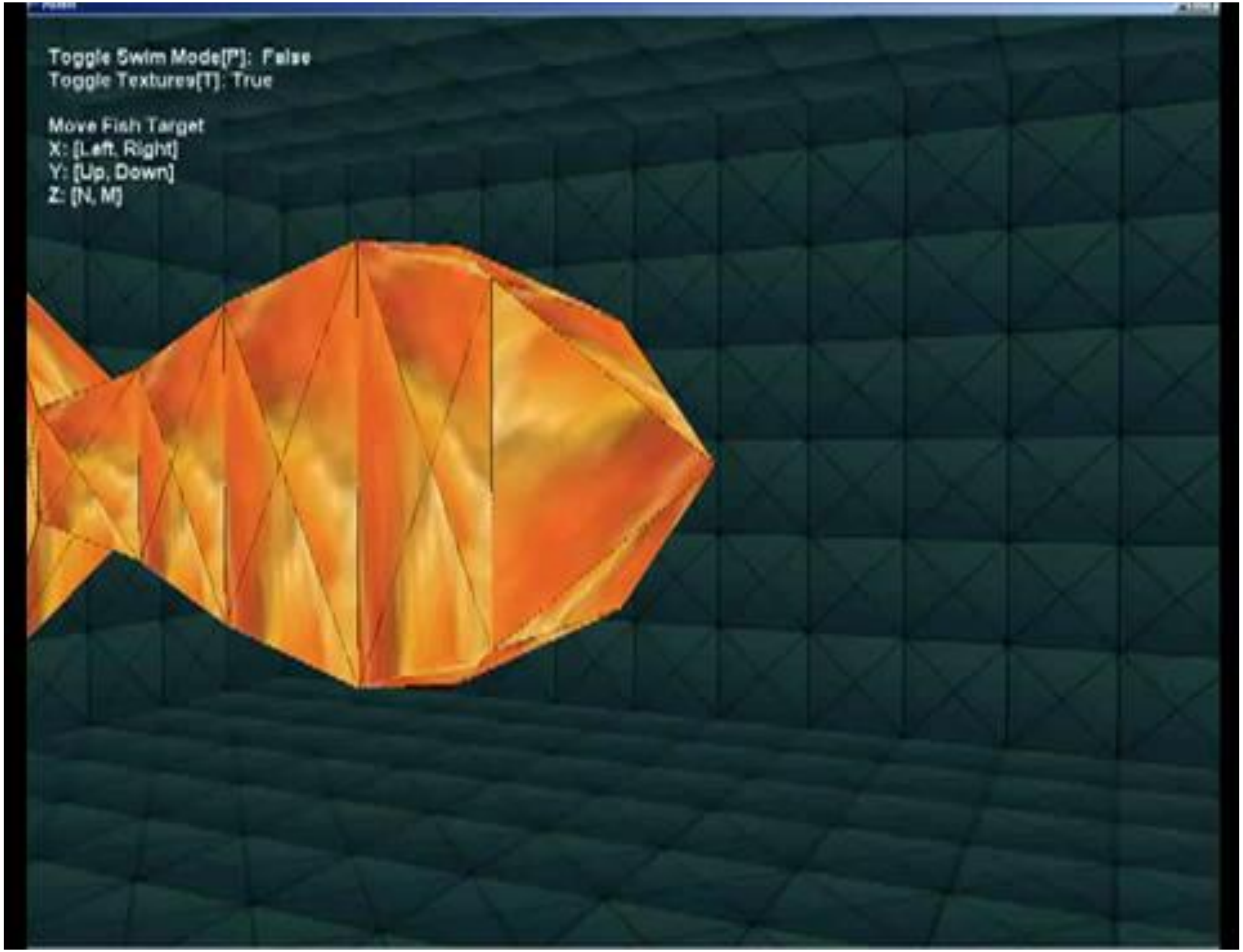
Craig Reynolds

Boids

Artificial Fishes

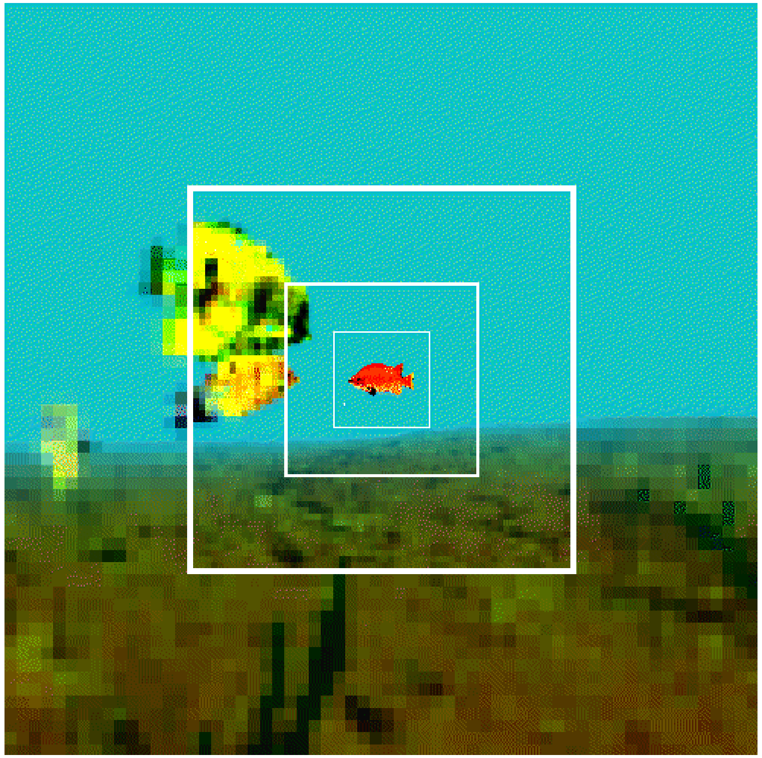
- Tu and Terzopoulos, SIGGRAPH 1994
- Perception, motor control, physically-based fish model

Artificial Fishes



Artificial Fishes

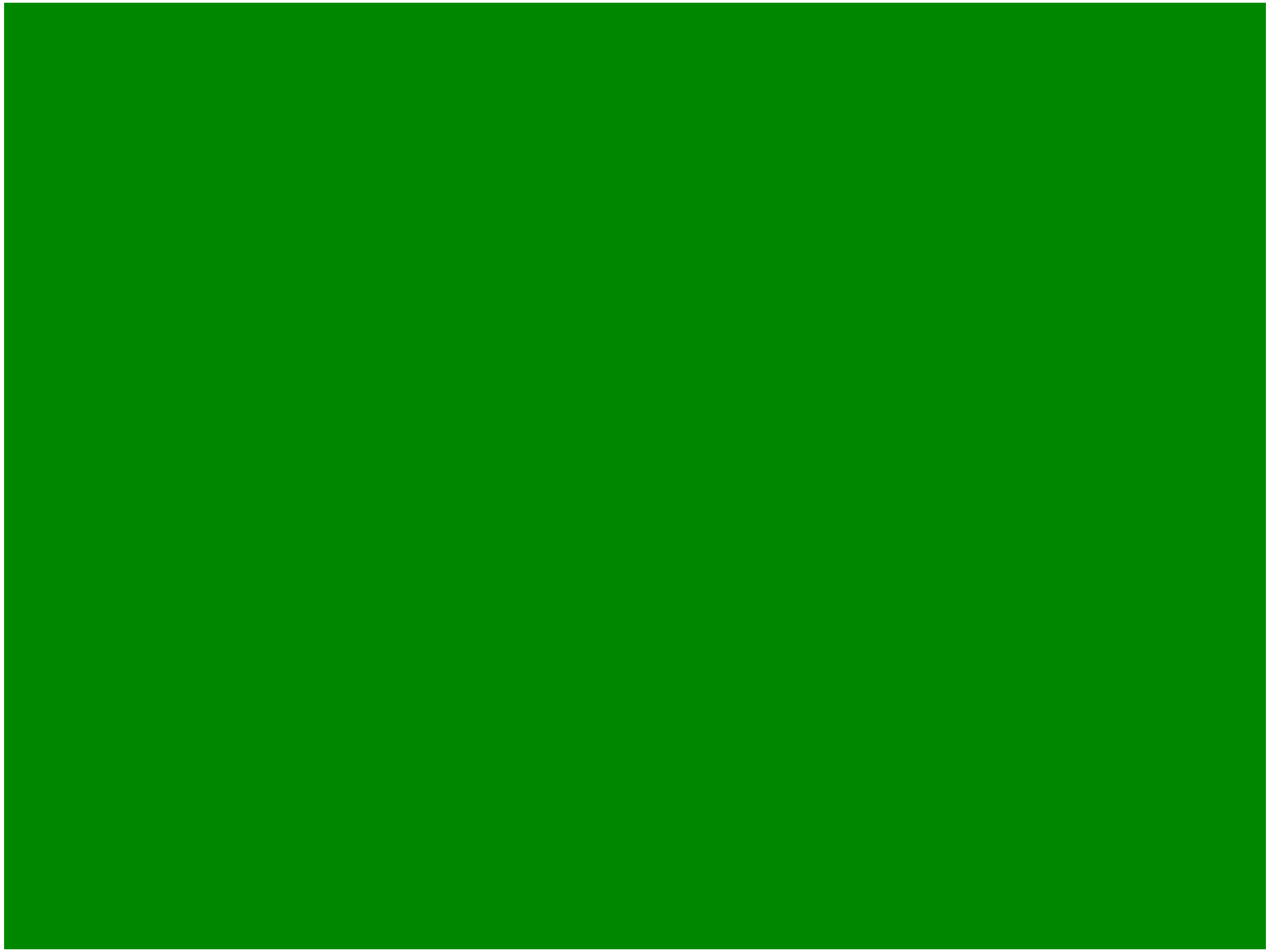
- Intentions generated
  - Habits, mental state and sensory perception
  - Chooses behaviour routine
  - Motor controllers
- Sensory perception
  - 300 degree F.O.V.
  - Selective attention mechanism and foveated vision
- Physics-based fish model
  - Spring-mass system



© Tu and Terzopoulos, 1994

Artificial Fishes

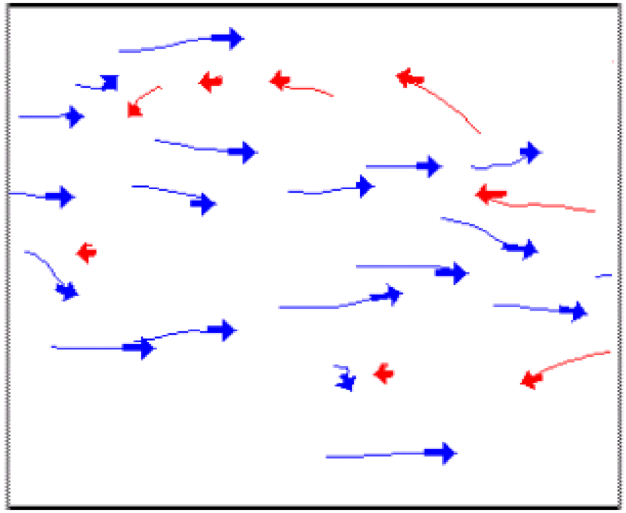
Artificial Life

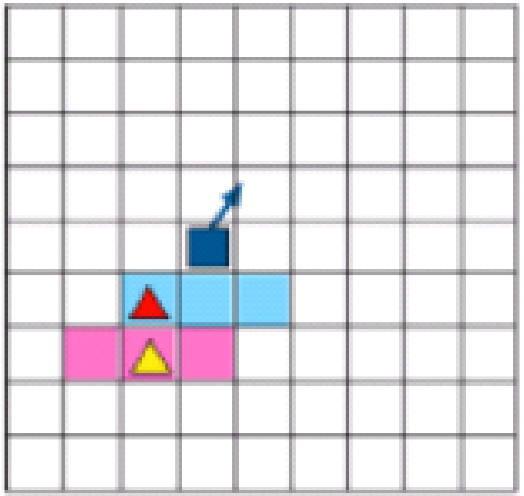


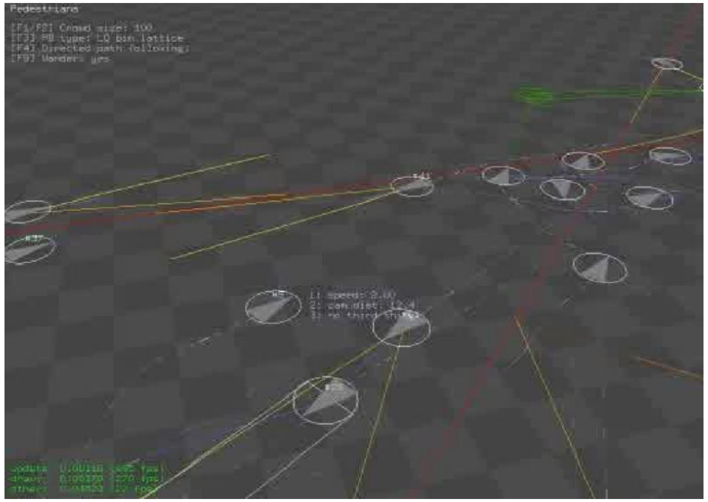
Karl Sims and more recent variations

Polyworld: http://shinyverse.org/larryy/Polyworld.html

Approaches







Social forces

- \+ Realistic pushing behaviour, lane formation
- \- Behaviour more like particles than humans

Cellular automata

- \+ Fast, easy to implement
- \- Underlying checkerboard pattern
- E.g. Loscos et al

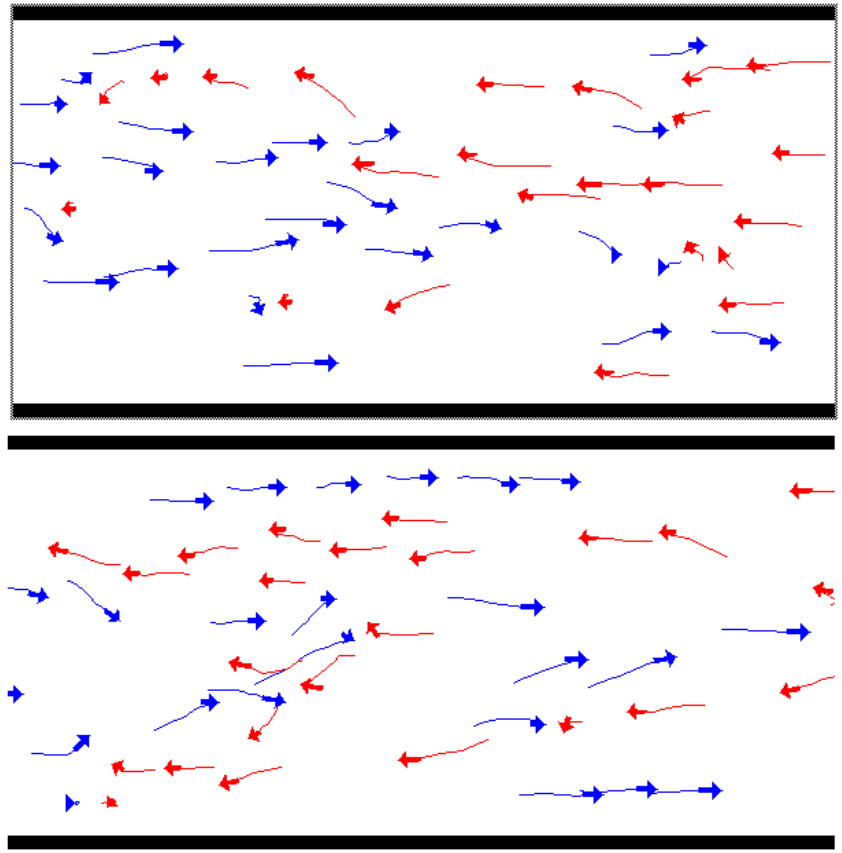
Rule-based

- \+ Realistic for low and medium densities
- \- Collision detection and repulsion not considered
- E.g. Reynolds

ROYAL INSTITUTE OF TECHNOLOGY

Social Forces

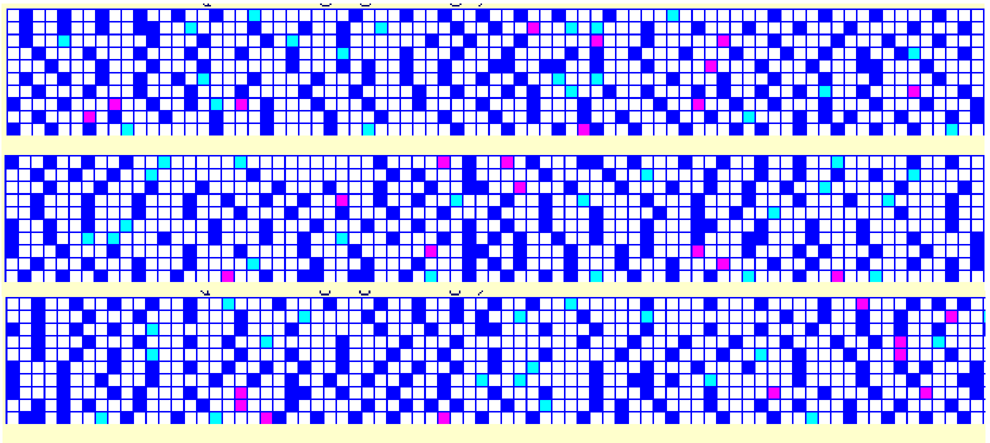
- Helbing social force model
  - Physical and socio-psychological factors
  - Particle system
  - Applies repulsion and tangential forces
  - Interactions between people and obstacles; flow
    - Realistic pushing behaviours, lane formation
    - Behave more like particles than humans, vibrations



http://rcswww.urz.tu-dresden.de/%7Ehelbing/Pedestrians/Corridor.html

Cellular Automata

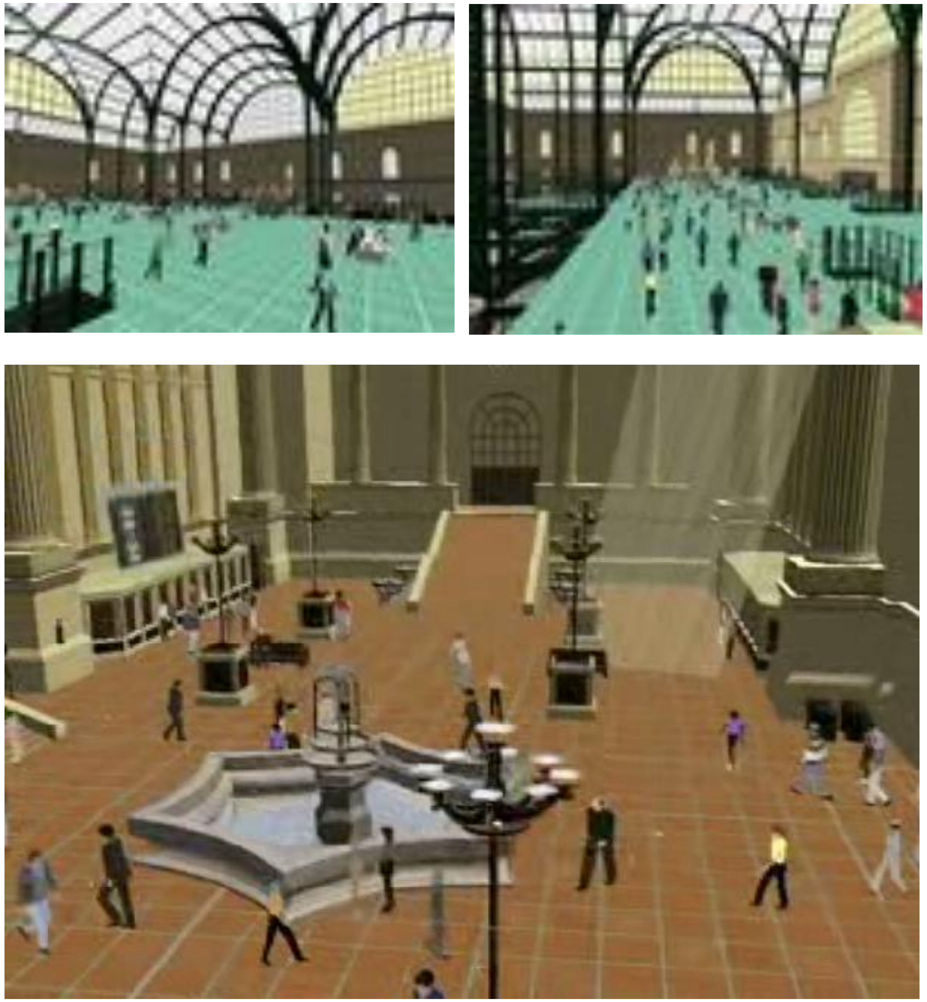
- Pedestrians simulated as particles in a grid of cells
  - Floor space is discretized
- Pedestrian picks the most beneficial neighbouring cell to proceed toward
  - Based on local neighbouring cells



[Blue and Adler, 2000]

ROYAL INSTITUTE
OF TECHNOLOGY

Autonomous Pedestrians



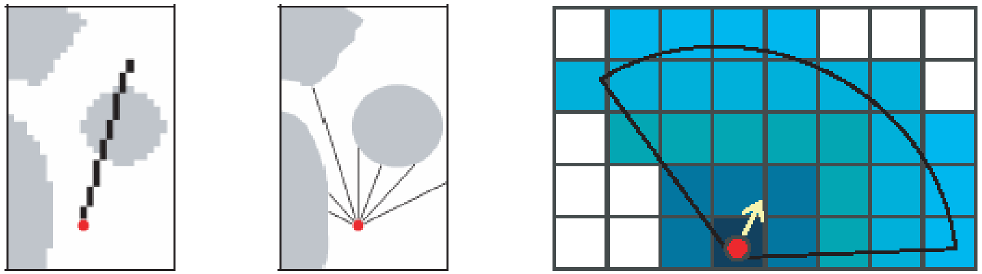
Shao and Terzopoulos 2005

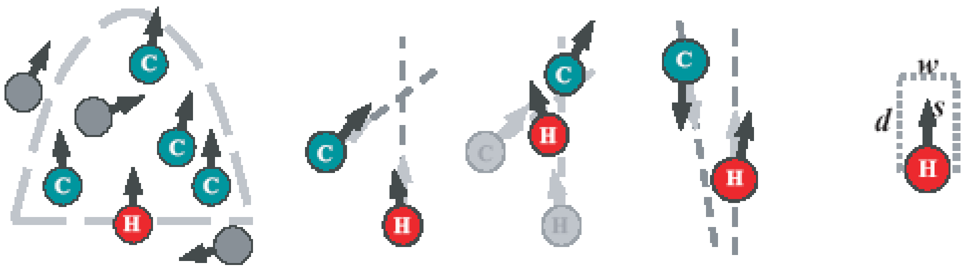
- Shao and Terzopoulos
  - Model individuals
  - Cognitive modelling
  - Deliberative as well as reactive behaviours
  - Virtual environment represented by hierarchical collection of maps

KTH VETENSKAP OCH KONST

ROYAL INSTITUTE OF TECHNOLOGY

Autonomous Pedestrians





Shao and Terzopoulos 2005

- Perception
  - Stationary *(left, middle)* and mobile objects *(right)*
- Behavioural control
  - Primitive reactive behaviours
  - Building blocks to support more complex behaviours
  - Controlled by action selection mechanism
- Cognitive control
  - Global path planning

Autonomous Pedestrians

Environment Hints

Adding *smarts* to the environment

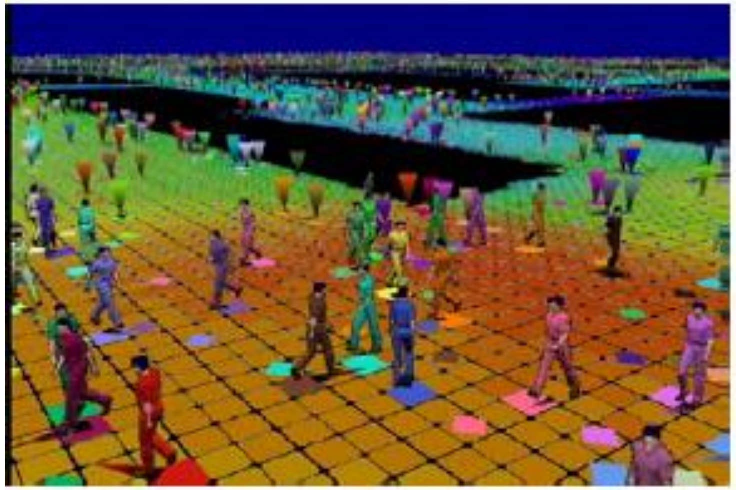
- Support agent behaviour by tagging objects and zones
- Describe what objects can be used for and how to use them
- Differentiate between different types of zones in the environment

Path-finding

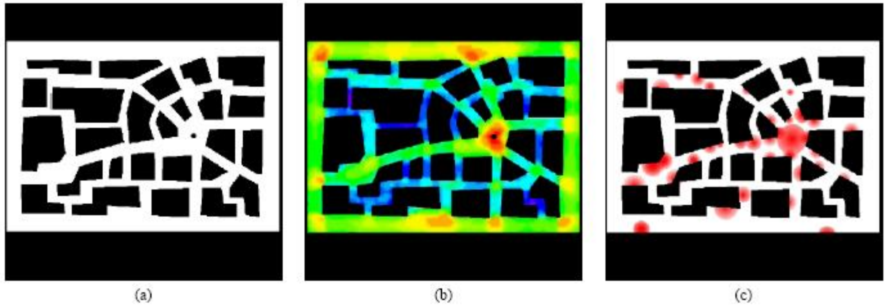
- Compute a path between a current position and a destination, avoiding obstacles
- Usually combined with *steering*

Examples

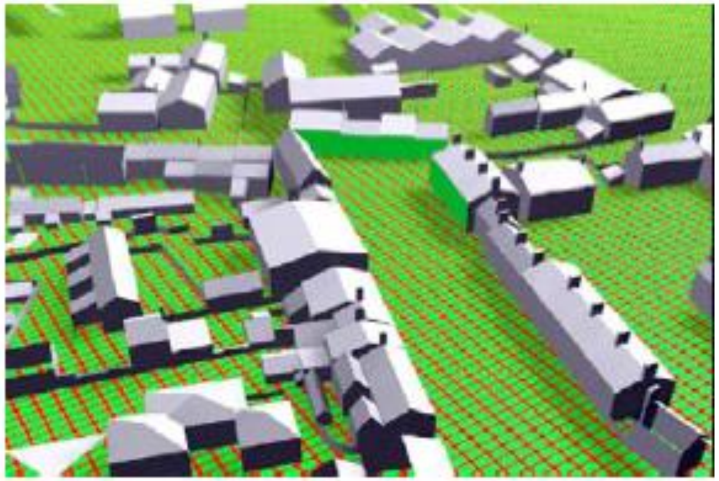
- Behavioural Maps
  - Collision, visibility and attraction
- Collision map
  - Indicate zones where agents cannot traverse
- Other info: *Thief the Dark Project*



© Tecchia et al. 2002



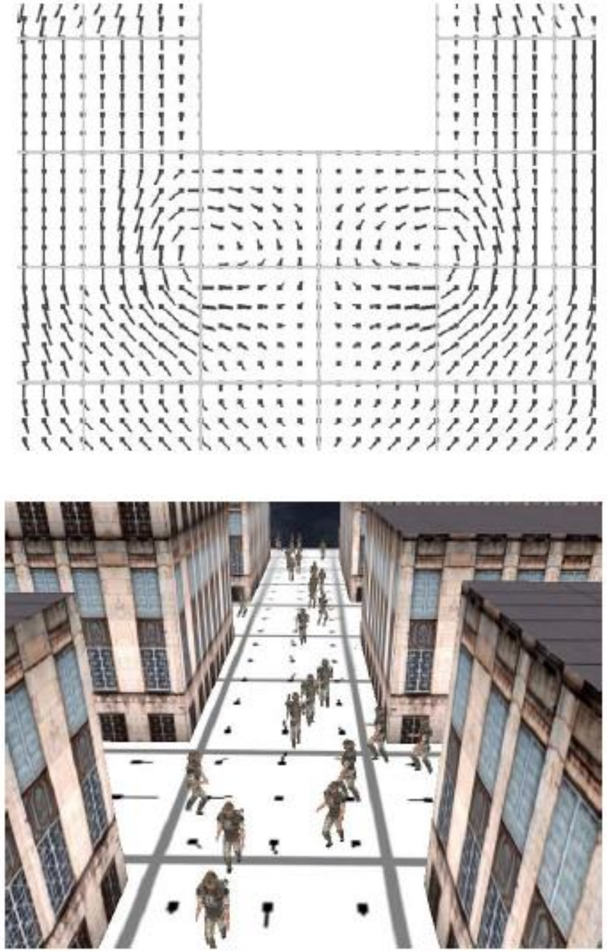
© Tecchia et al. 2002



© Tecchia et al. 2002

KTH VETENSKAP OCH KONST
ROYAL INSTITUTE OF TECHNOLOGY

Examples

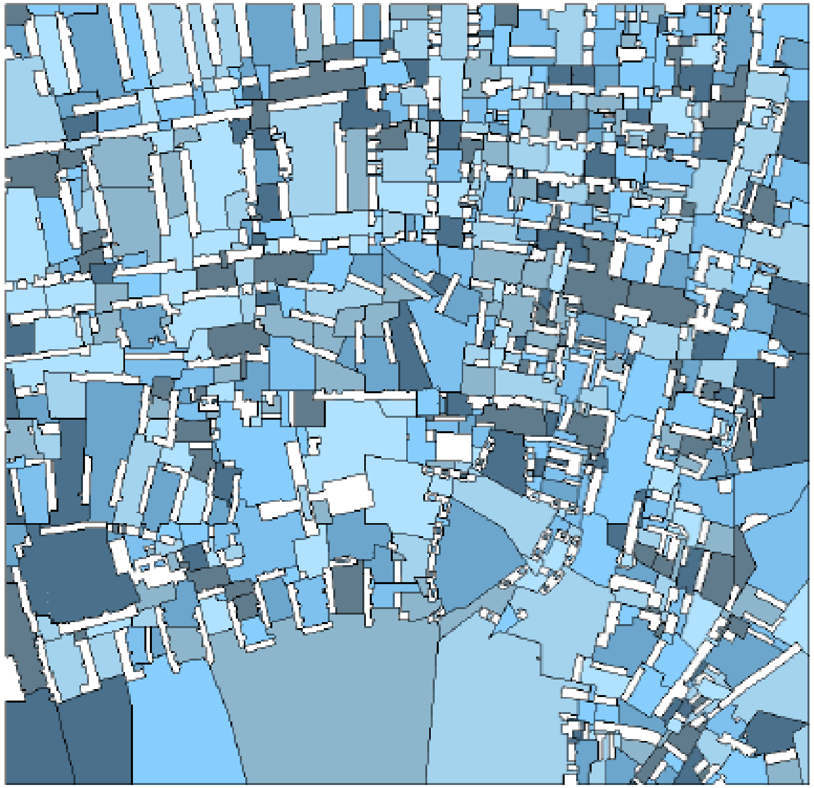


[Chenney 2004]

- Potential fields
  - Velocity fields
  - E.g. “Flow Tiles”
  - Each tile is a small stationary region of velocity field
  - Tiles can be pieced together to form larger velocity fields
  - Easy to design

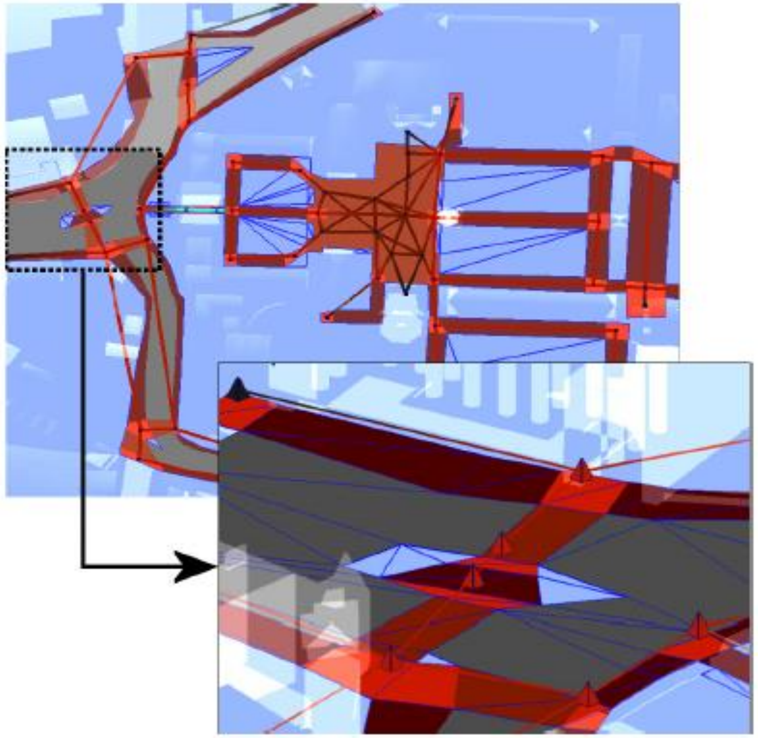
Examples

- Spatial partitioning
- *Cells* and *Portals*
- Automatically partition a city-like environment into zones
- Portals are connections between those zones
  - Think of them as windows that connect two zones together



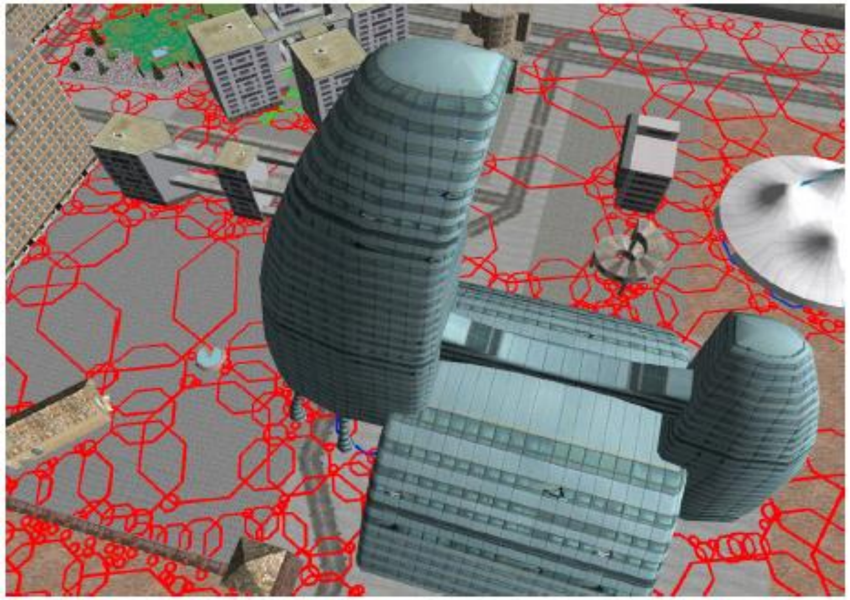
[Lerner et al 2006]

Examples



MetroPed

- Peters and O' Sullivan
- Path-planning and zoning
- Manual scene partitioning and behaviour tagging

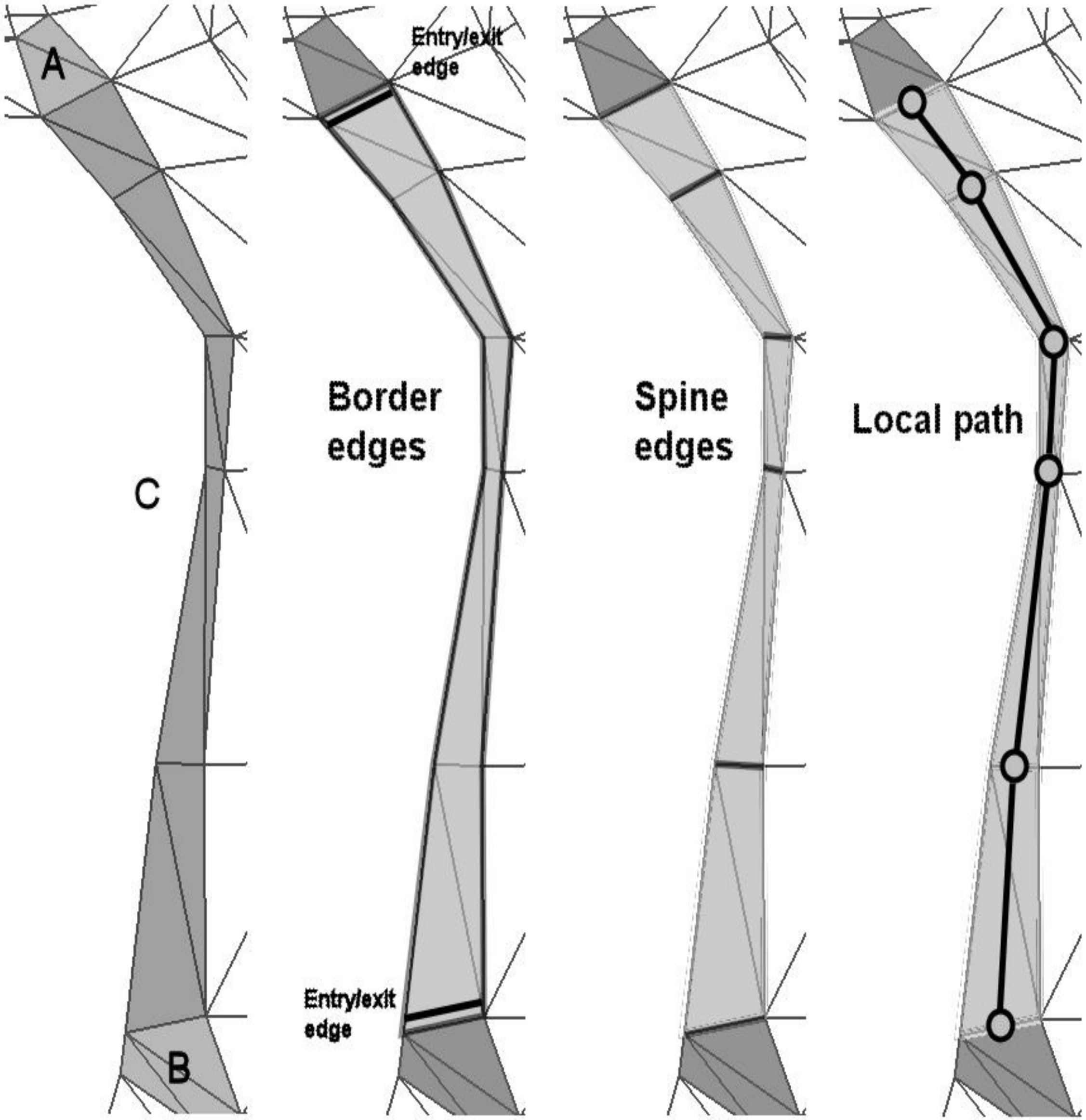


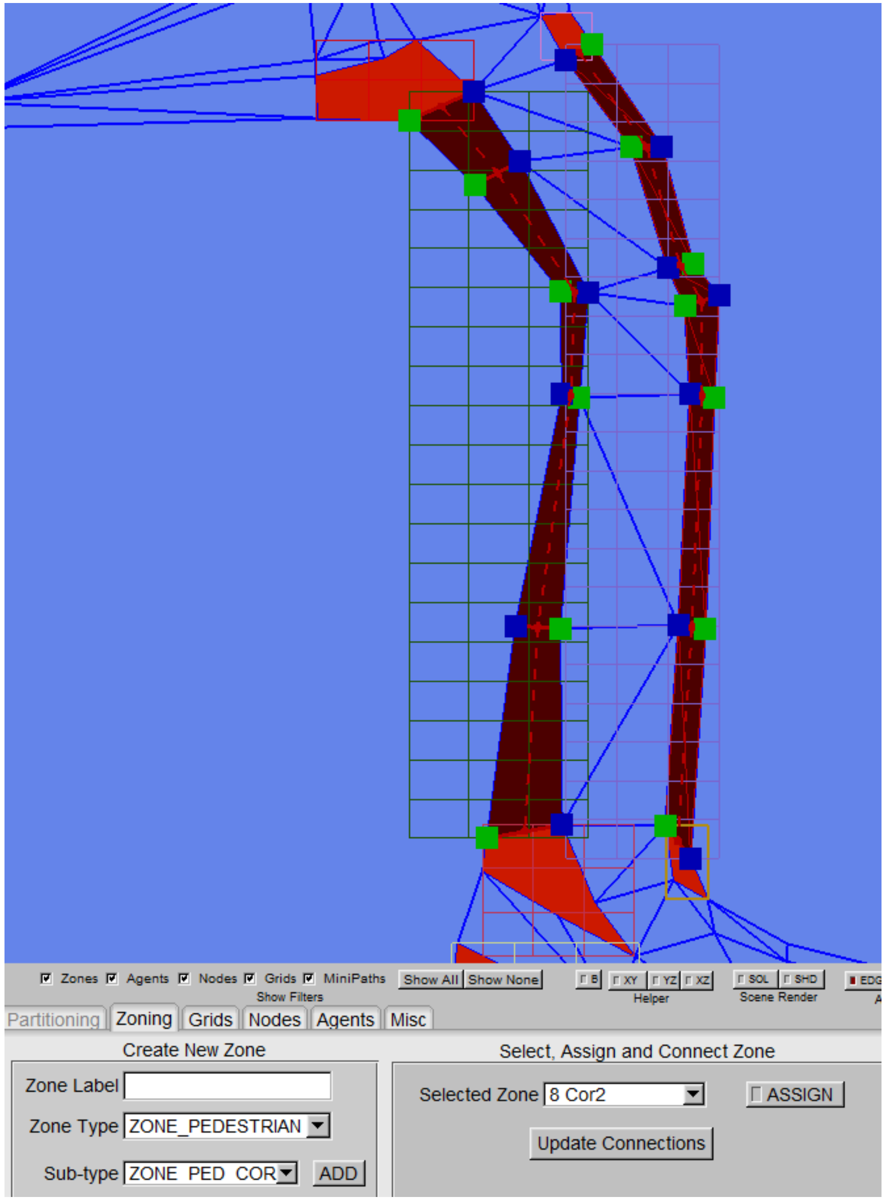
Bounding cylinders

- Pettre, Thalmann et al
- Scalable path-planning
- Fully automatic scene partitioning

ROYAL INSTITUTE OF TECHNOLOGY

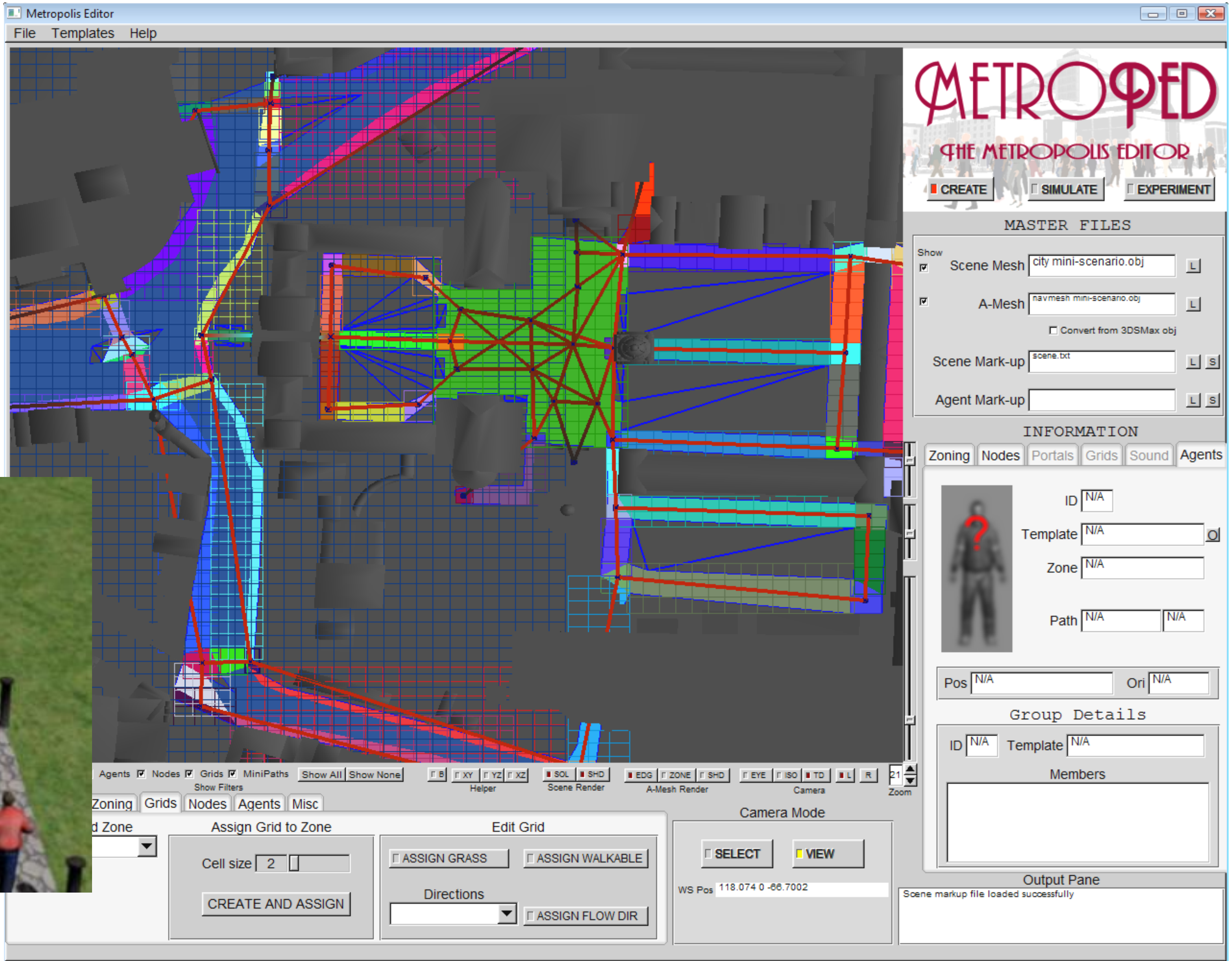
Example: Local Paths





KTH VETENSKAP OCH KONST
ROYAL INSTITUTE OF TECHNOLOGY

Smart Environment Editing



Reminders

- You should be working on Lab 3
- Assignments due on 25th May
  - DH2323 submission will open very soon

- Upcoming lab sessions:

May 16th, 10:00-12.00, VIC Studio

May 25th, 13:00-15:00, VIC Studio

ROYAL INSTITUTE OF TECHNOLOGY

Next lecture

- User studies and perception
- VIC Studio
- Monday 14th May
- 10:00 – 12:00

VICSTHLM
VISUALISATION INTERACTION COLLABORATION