While true InIn IntersectionInstance map and in state shapeClass shape positionClass position Sensor motionClass motion sense() I guess pointer to all world objects is required offspringObjects=evolve(delta-t) + world intersect method How these possibly imaginary objects interact LidarSensor with the world and how long they live is not an BumperSensor Robot essentially a light beam rotating at an architectural concern Brain brain detects at a hit upon intersection, does x-y plane and measuring distance with evolve(delta-t) not involved in shape object requires information. For instance, insurrounding objects Call sensors if any tersection with other objects. What to pass to Call brain.stateTranform() evolve()? visualize() boundingBox() VaccumCleaner Brain brain needs shape, we can omit this anyway Sensor laser dumpState() Sensor bumper Wheel wheelRight of states. Maybe with variables names and val-Wheel wheelLeft ues. All textual get-required-delta-t() Wheel model a wheel much. Just an object which evolves in its position angularSpeed evolve(delta-t)

World Object objects[] WorldState lastWorldState constructor(map, initial state) parsing and interpreting a chosen map format done in this function? evolve(delta-t) For object i in objects: offspringObjects=objects[i].evolve(delta-t, intersectionResult[i])) self.addObject(offspringObjects) determine delta-t from all Object.get-required-delta-t() intersectionResult=worldClass.intersect() t=t+delta_t worldClass.evolve(intersectionResult) lastWorldState.recordState(objects) InIn intersectionResult[]=intersect() intersectionResult is an array of array of InIn, where IntersectionResult[i] is an array of InIn pertaining to object i for every object i: for every other object j: if not cubesIntersect(objects[i].boundingBox() ,objects[j].boundingBox()): intersectionResult[i]=empty intersectionResult[i][j]=no idea what to do, an instance of InIn #they still might have no intersection. #But if they do we need some details Position position and orientation Object x,y,z objectID every object needs an ID to be tracable in

position of an anchor point regardless of shape

offsprings are the possibly created objects.

pass for a trivial evolution. Otherwise, the

returns the information required for graphics

simply calls that of the shape? TBD. if world

return value must have the object ID + a list

calculates the delta-t it requires to operate

note that we could but don't have to

changes the state

WorldState

use: e.g. debug

the simulator works in delta-t steps. A worldState can be recorded either before or after evolution of all the objects. A partial worldState is invalid unless every object state is flagged as completed or not. The reason is that the world cannot be initialized with a snapshot of a partially completed worldState without knowing which objects have already evolved in that snapshot.

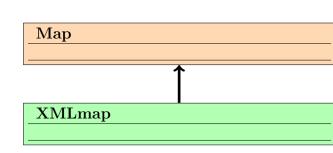
A data structure which keeps a set of states for every objectID

load(file) dump(file)

extractState(objectID) recordState(objects[])

For object in objects:

Get and record object.dumpState()



phi, theta

Motion

Velocity and acceleration

vel-x, vel-y, vel-z acc-x,acc-y,acc-z

Shape

some uniform definition? See the comment on

boundingBox()

returns a x-y plane bounding box. Can be done using a generalized algorithm, no implemented only in the parent class.

DiscretizedShape

takes a mesh as input. Mesh given in a file or something?

DiscretizedShape2D

es bitmap and if required extends it to a mesh of infinitely long pieces along z-axis

DiningTable

Cylinder legs[] Cube topSurface Disk axis1

> xis2 dinate center

Cylinder Disk disk1

Disk disk2