



**Neo** : Why am I here?  
**The Architect** : Your life is the sum of a remainder of an unbalanced equation inherent to the programming of the matrix. You are the eventuality of an anomaly, which despite my sincerest efforts I have been unable to eliminate from what is otherwise a harmony of mathematical precision.

**InIn**  
IntersectionInstance

**intersect()** -> dict  
evaluates the intersection.  
An algorithm with mathematical calculations.

**World**

**objects:** list[Object]

**constructor**(map: Map)  
**evolve**(delta-t)  
calls evolve() of all objects which have no owner, takes care of offspring objects, and finally kills objects which ask for it  
**run()**  
manages delta-t (how small it should be), manages intersections using self.intersect(), and calls self.evolve()  
**intersect()** -> **intersection-result:** list[InIn]  
instantiates InIn for each pair of objects and calls InIn.intersect() to evaluate the intersection

**Map**



**Smith to Neo:** Because Of You, I'm No Longer An Agent Of This System. Because Of You, I've Changed. I'm Unplugged. A New Man, So To Speak.

**Sensor**  
inherently an Object, so needs shape, position, evolve(), etc.

**sense()**

**BumperSensor**  
detects at a hit upon intersection, boolean result

**Robot**

**evolve()** is the heart of Robot actions.

**VaccumCleaner**  
in version 0, a cylinder with no wheels and stuff.

**Object**

**objectID**  
every object needs an ID to be tracable in map and in state  
**shape: Shape**  
which could be empty if the object is an owner  
**position: Position**  
position and orientation of an anchor point of the object  
**evolve**(delta-t, **intersection-result:** InIn) -> list[Object]:  
**offspring-objects**  
changes the state (position, internal attributes, etc) of the object  
trivial evolution: when the object never changes state  
offsprings are the possibly non-physical objects required to accomplish something.  
**visualize()**  
returns the information required for visualization  
**bounding-box()** -> Box  
returns a box which contains the whole object. used to optimize intersection evaluation  
**get-required-delta-t()**  
calculates the delta-t it requires to operate  
**time-to-die()** -> bool  
tells the World if it wants to be eliminated. This might be where Agent Smith cheated the matrix!

**Position**  
position and orientation

**x,y,z**  
**phi,theta**

**Shape**

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

**Cylinder**

**Box**