

# Spatial Database

Friday, October 29, 2010  
8:26 AM

This design uses the SQLite database as a backend to the spatial database. The spatial data will provide the following services:

- World view of region
  - o One can display all the items in perspective
- Spatial Triggers
  - o One can register a region to be watched for collisions. If another object enters the region, the a callback will

AutomatonFactory <<singleton>>
Automaton& GetAutomaton(Point)

Point
int X
int Y
Point(X,Y)

SpatialModel
watch_region(const Region&, callback) select_region(const Region&) const update_region(const Region&, Point) //Updates a region create_region(Region&)//Accepts an empty region and initializes it in the db

Automaton
int64_t spatial_id Point current_location Vector heading

Vector
int speed int direction

RegionFactory <<singleton>>
Region& GetRegion(Point, Point, Point, Point)

Tables:

Automatons <<R*-Tree>>
Id Point A

Regions <<R*-Tree>>
Id Point A Point B Point C Point D

Region
Point A Point B Point C Point D protected: int64_t spatial_id