Environmentally Conscious AI
Improving Spatial Analysis and Reasoning

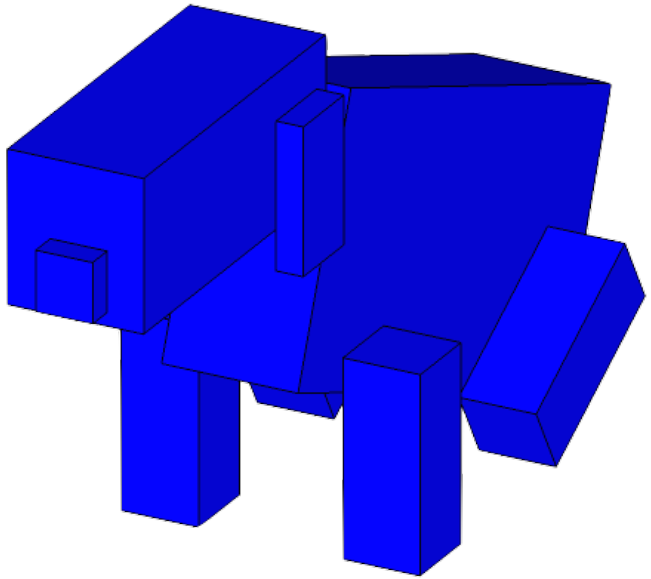
Clodéric Mars
MASA LIFE Lead
cloderic.mars@masagroup.net
@clodericmars

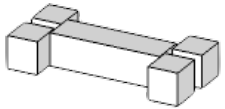
AI ARTIFICIAL INTELLIGENCE SUMMIT
GAME DEVELOPERS CONFERENCE 2014
SAN FRANCISCO, CA
MARCH 17–21, 2014
EXPO DATES: MARCH 19–21

What is Spatial Reasoning?

- Spatial objects
  - POIs
  - Areas/zones
  - Influence map
  - ...
- Relationships
  - Distance
  - Visibility
  - Accessibility
  - ...

"What is the **closest** bone that is **not too old**, out of the **cat likely position** and that I can chew while **observing** the street?"





How?

- List of points of interests
- Query several sources
  - "closest" ➡ *pathfinding*
  - "not too old" ➡ *metadata*
  - "likely position" ➡ *influence map*
  - "observing" ➡ *visibility mesh, pvs*
- Add some custom logic

➥ **A little complex for "just" a dog!**

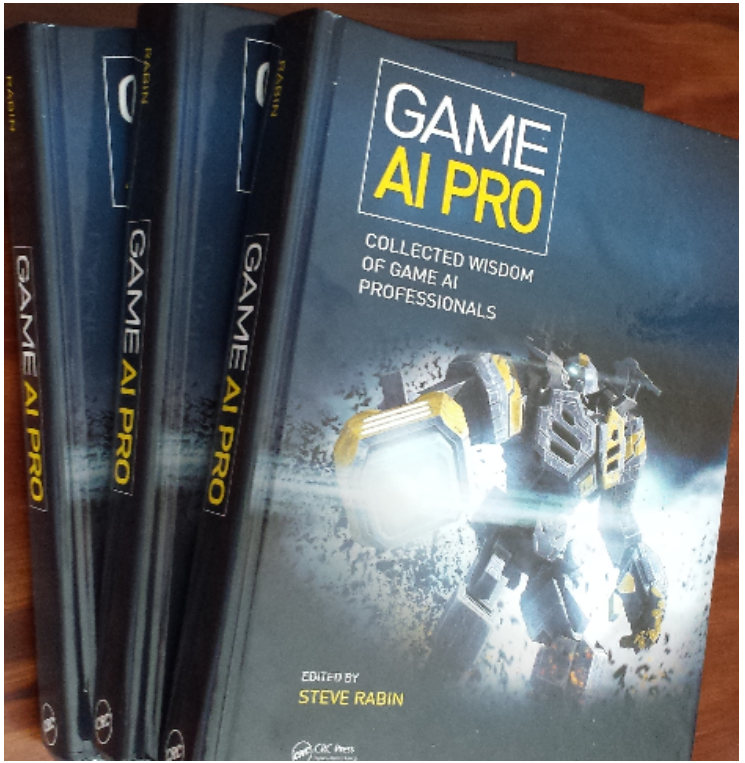
Wouldn’t it be cool if...

- Behavior designers can focus on **using**
  - “One stop” gateway
  - Expressive query
- Developers can **extend** and **optimize**

➥ **Handle the “technical” complexity for the benefit of both**

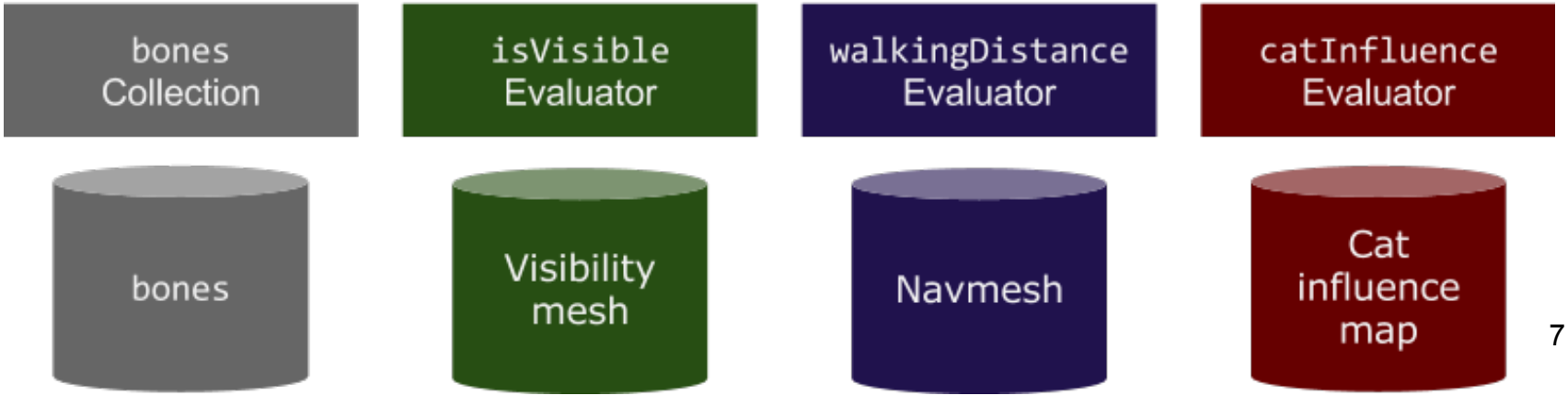
Inspirations

- Game industry
  - Environment Tactical Querying (ETQ) [Zielinsky 2013]
  - Tactical Position Selection (TPS) [Jack 2013]
- GIS
  - PostGIS (postgis.net)



```
SELECT b
FROM bones
WHERE b.metadata.age < 20
     AND catInfluence(b.position) < 0.1
     AND isVisible($.street,b.position)
ORDERED BY walkingDistance(b, $.position) LIMIT 1
```

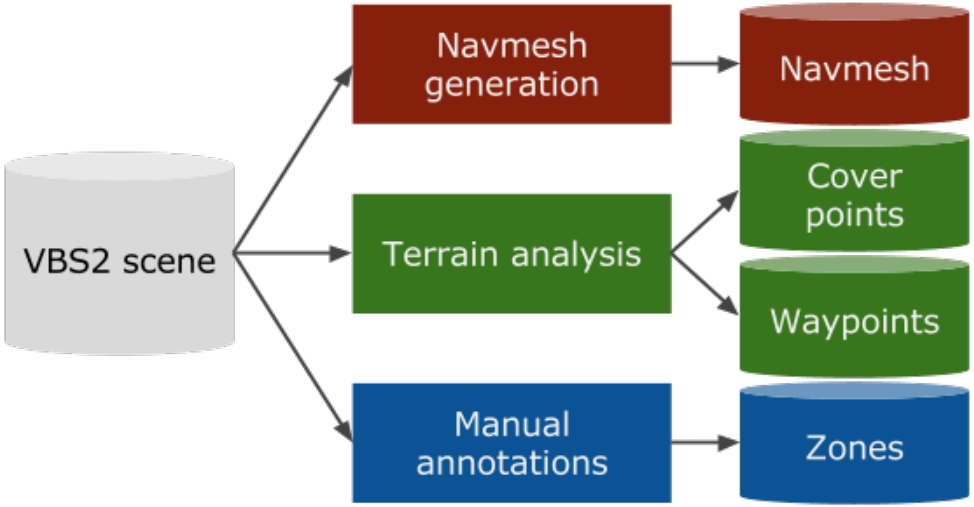
MASA LIFE's SpatialDB

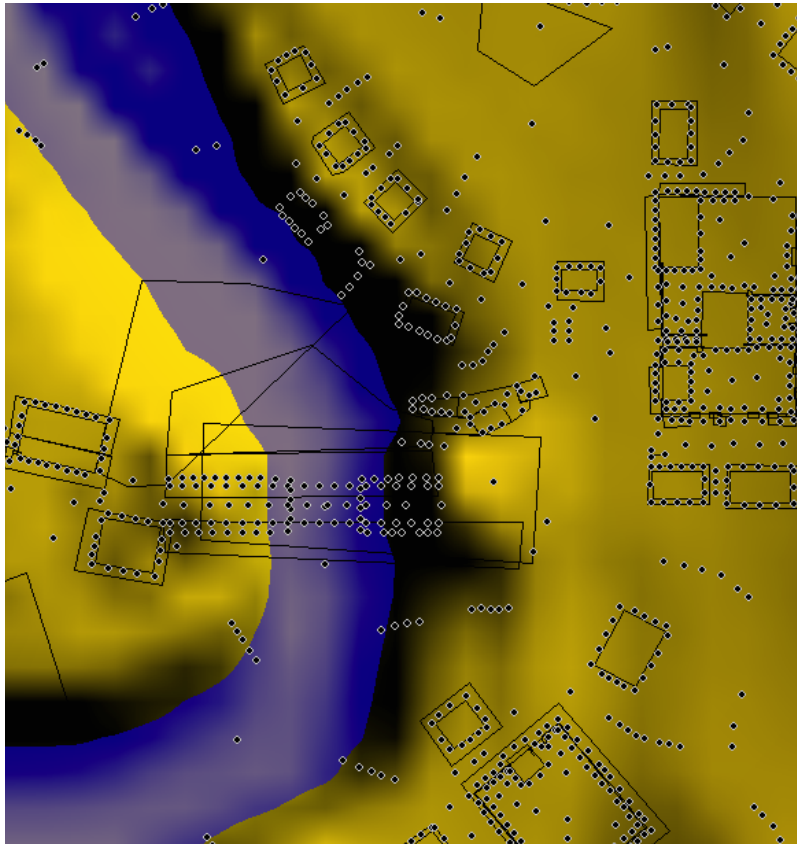


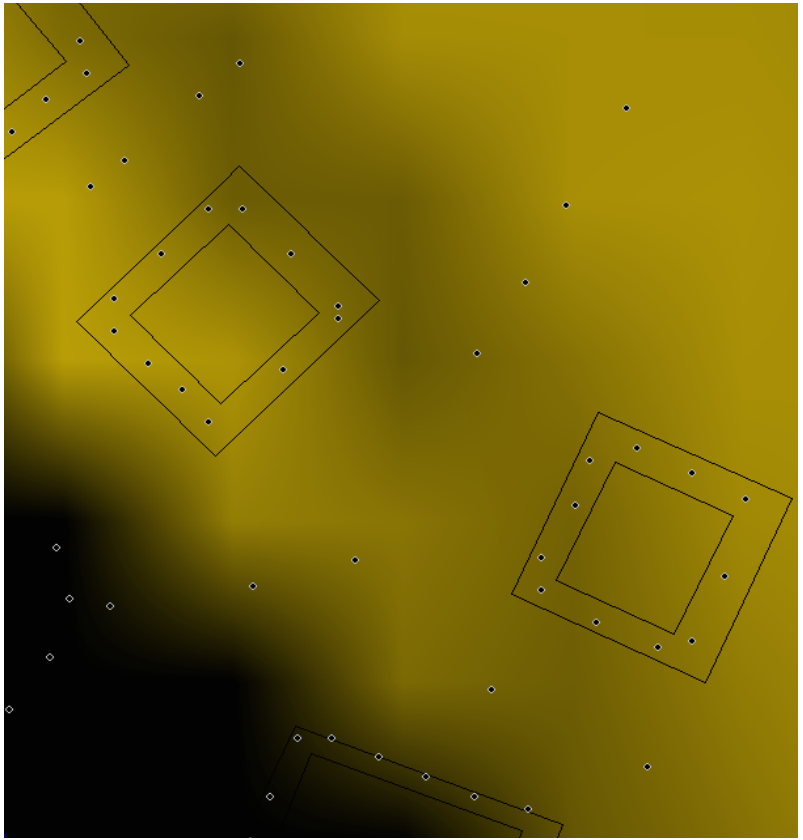
Let’s study a practical example!

Tactical behaviors in VBS2

- Tactical groups...
- ...navigating in a "hostile" environment
- Behaviors that don't depend on the environment







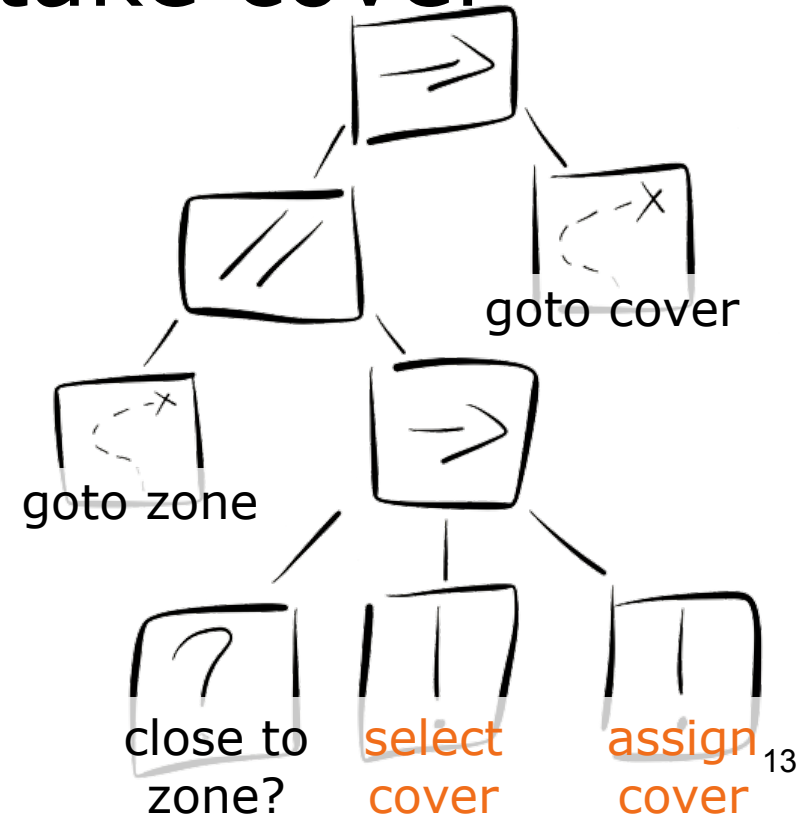
"select cover" query

```
SELECT p FROM interest_points
WHERE p.metadata.type == 2
    AND (p.metadata.assignee == NULL
        OR p.metadata.assignee == $.id)
    AND p.geometry.x > $.position.x - 10
    AND p.geometry.x < $.position.x + 10
    AND p.geometry.y > $.position.y - 10
    AND p.geometry.y < $.position.y + 10
ORDERED BY distance(p.geometry, $.position)
LIMIT 10
```

- collection
- type (2 is cover points)
- free or already assigned to me
- no farther than 10 m
- order by distance to me
- take the first 10

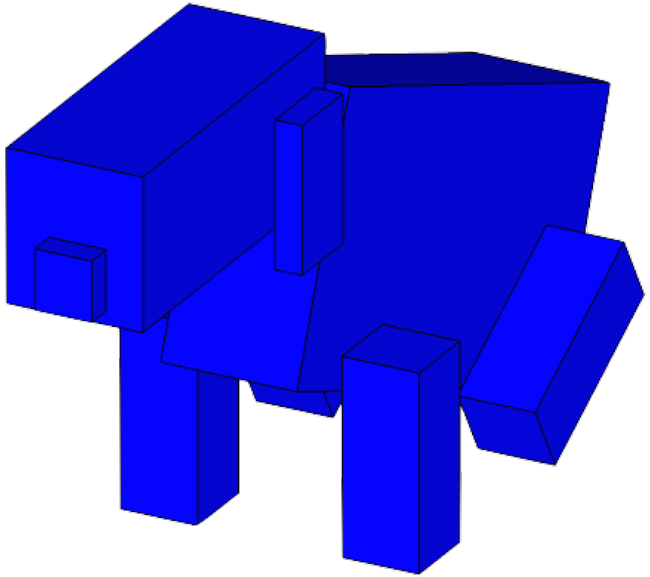
Reach a zone and take cover

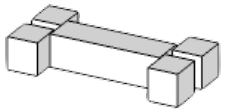
- "select cover"
  - SpatialDB query
  - Distance
  - Protected direction
  - Availability
- "assign cover"
  - SpatialDB update
  - Transactional



Technical stuffs

- SQLite (www.sqlite.org)
  - Easy to integrate
  - Public Domain
  - Mature
  - Fast
  - Geospatial index (r-tree)
  - Functions
  - In-Memory
- peg/leg (piumarta.com/software/peg/)





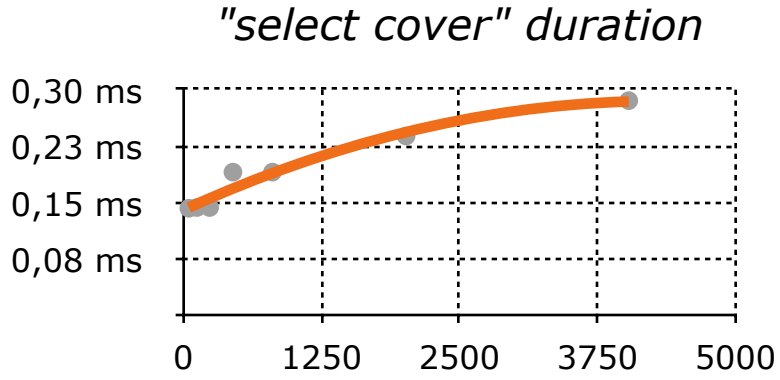
How? … with this implementation

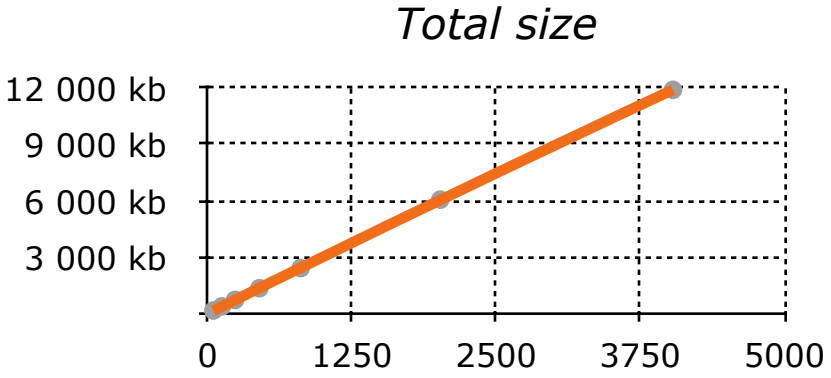
- *List of points of interests (bones)*
- *Query several sources (navmesh, visibility, …)*
- *Add some custom logic*

- SQLite Tables
  - metadata
  - spatial index
- SQLite Functions
  - Signature
  - Callback to subsystem
  - Metadata access/index
- SQLite ‘arithmetic’ and custom parser

What worked

- Fast implementation of first version
- Data driven behaviors
- Familiar query language
- OK performances





Core i7 @ 2Ghz // 8Go of RAM

What needs to be improved

- Easy-to-use assignment
- Tools
  - inspection
  - sandbox
- Performances
  - grouping/ordering evaluators
  - time slicing
  - multithreading

Takeaways

1. **Decouple** (designer/developer)
2. Use **existing** tech (eg. SQLite)
3. Create a **familiar** language

Meet us at
booth #1743
@masalife_ai