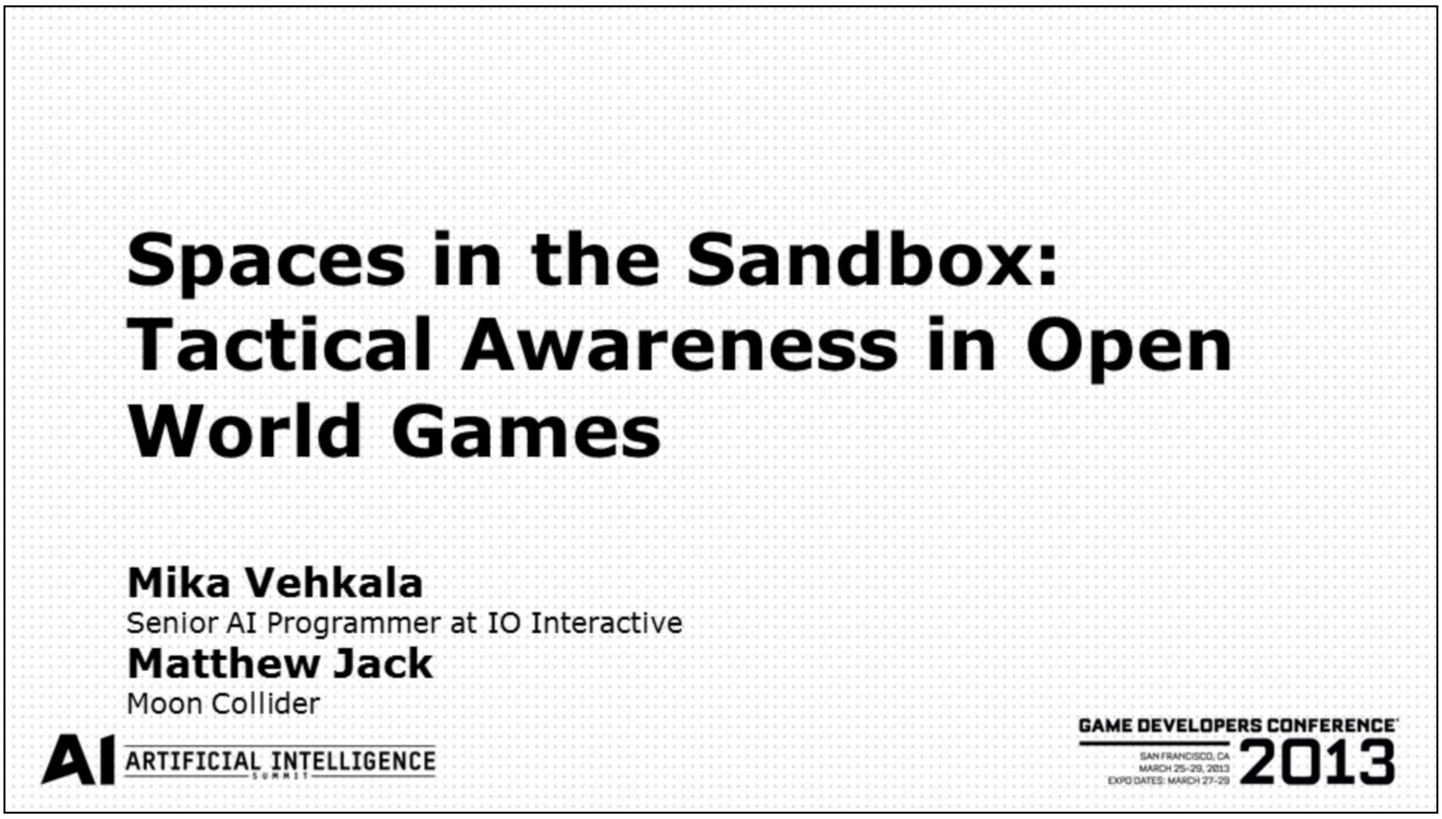


Spaces in the Sandbox: Tactical Awareness in Open World Games

Mika Vehkala
Senior AI Programmer at IO Interactive
Matthew Jack
Moon Collider

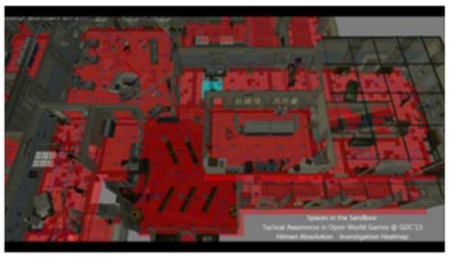
AI ARTIFICIAL INTELLIGENCE SUMMIT

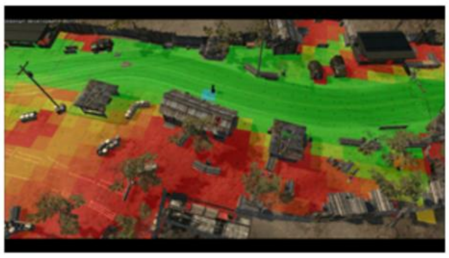
GAME DEVELOPERS CONFERENCE
SAN FRANCISCO, CA
MARCH 25-29, 2013
EXPO DATES: MARCH 27-29
2013

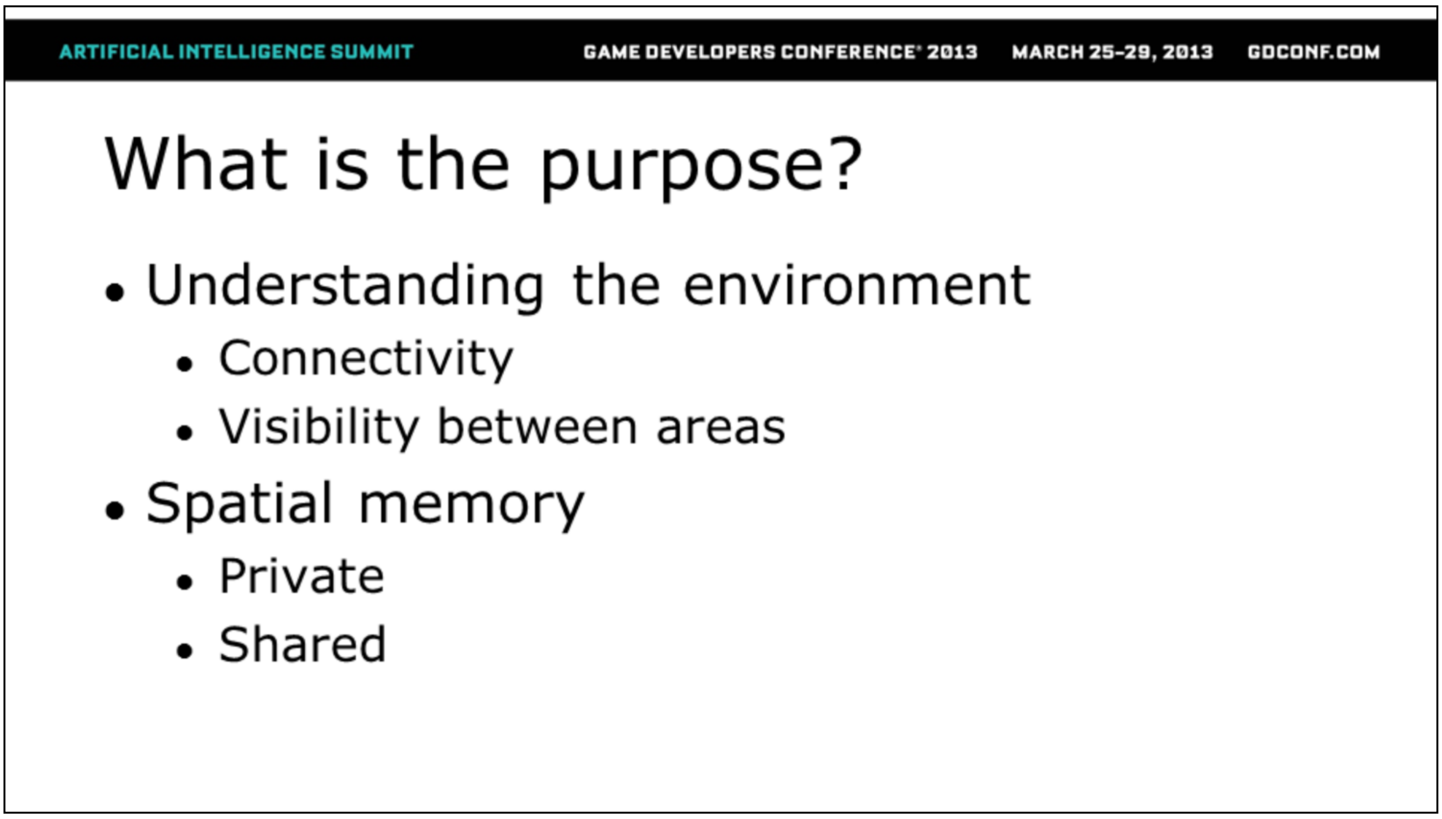
ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25–29, 2013 GDCONF.COM

Overview

- Part I – Offline
- Part II – Runtime
- Part III - Examples


Spaces in the Sandbox
Tactical Awareness in Open World Games @ GDC'13
Hitman Absolution - Investigation Heatmap



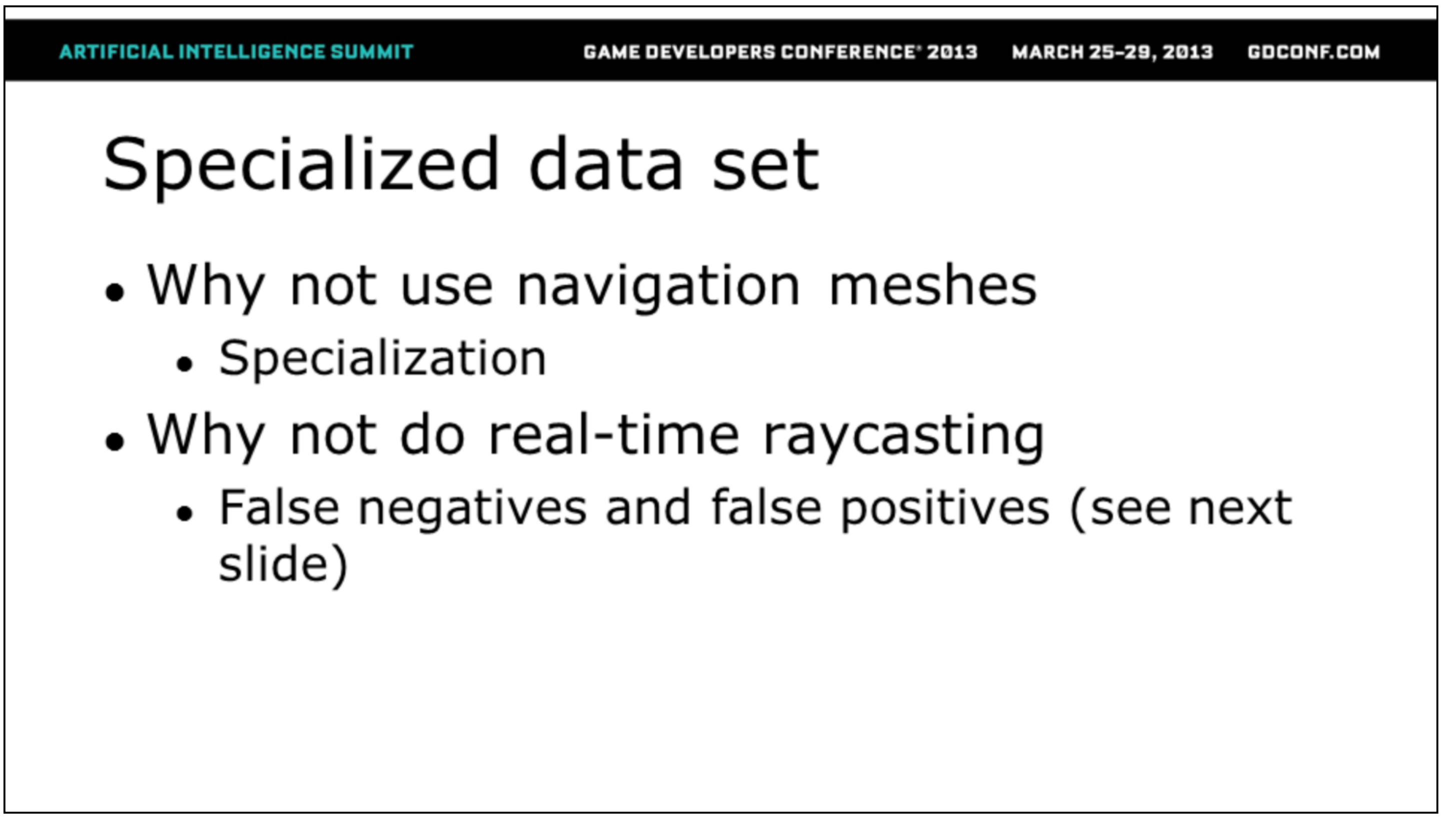

ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

What is the purpose?

- Understanding the environment
  - Connectivity
  - Visibility between areas
- Spatial memory
  - Private
  - Shared

For game AI to be able to figure out what is the best course of action for NPC's in a given situation, and more importantly **where they should be**, some information about the world around the NPCs is needed.

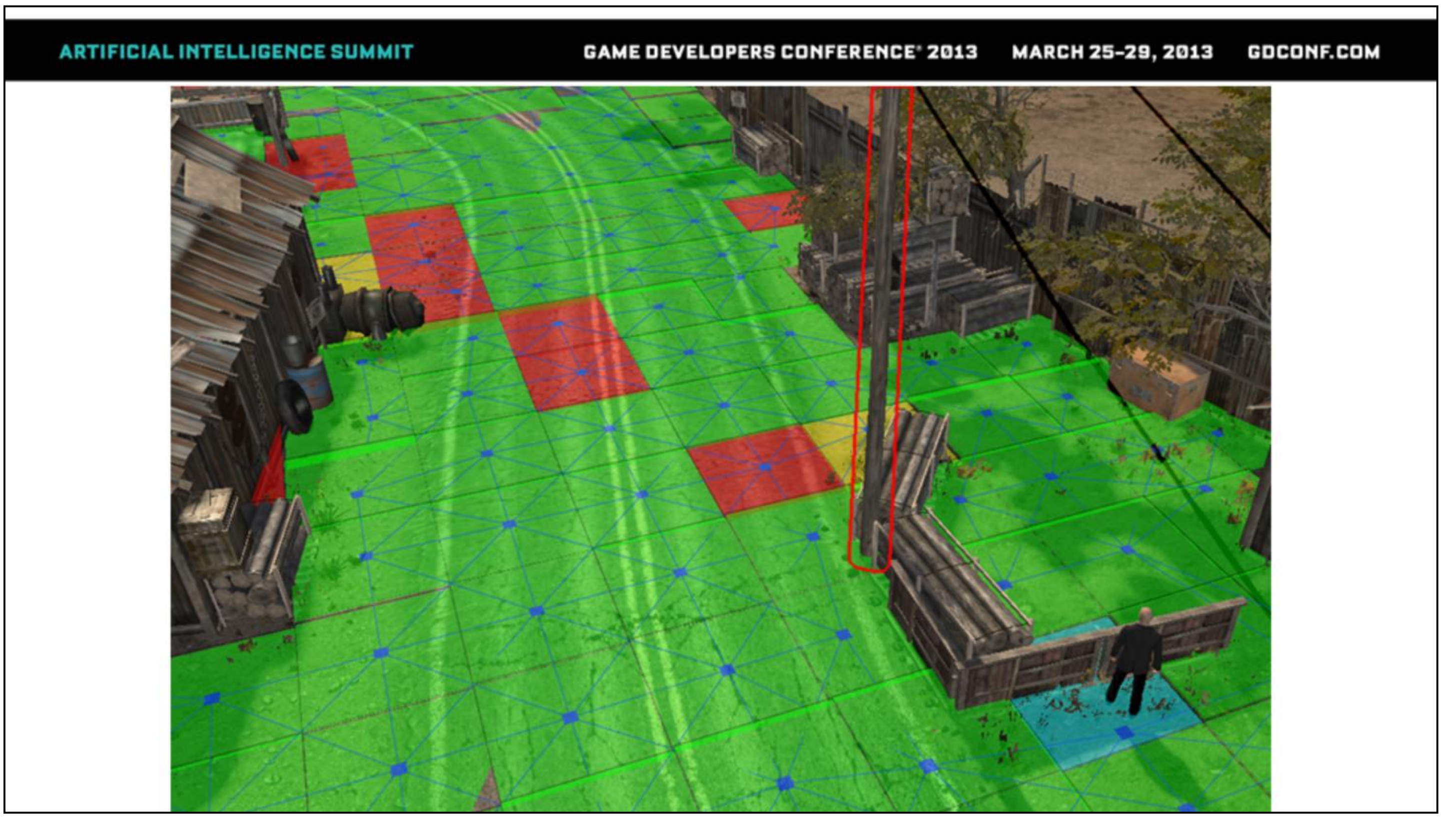
The data set needs to contain **connectivity information**, just like navmesh, about the world that can be traversed by NPCs. It also needs to contain information about **visibility**, what can one see from each 'point' in the world. This is a very important requirement since doing any kind of tactical decisions that can extend potentially tens of seconds into the future one must be able to figure out which locations gives the best and most long lasting probability in either gaining or avoiding line-of-sight and line-of-fire. Selected position might be invalidated before NPC reaches it since player will have the same amount of time to change his position. Obviously for games where the average lifetime of NPCs, from spawning to death, is only few seconds this might not be the case. Finally, it should be possible to access this data very fast and have the possibility of storing arbitrary information to each of the nodes. Storing information acts as spatial memory which is important for coordination.



Specialized data set

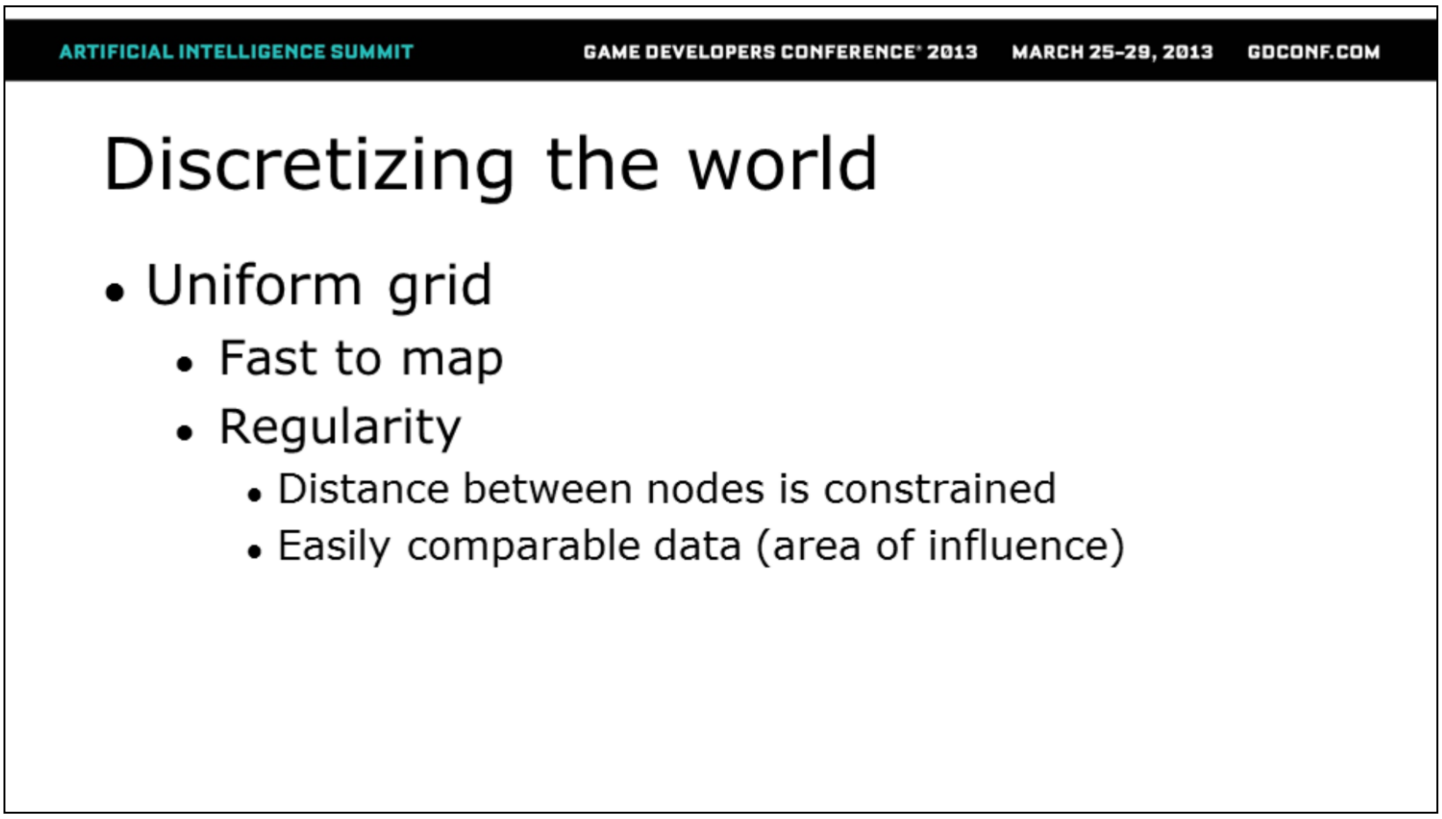
- Why not use navigation meshes
  - Specialization
- Why not do real-time raycasting
  - False negatives and false positives (see next slide)

Specialization is good as it gives good performance and simple API for each use-case.


ARTIFICIAL INTELLIGENCE SUMMIT
GAME DEVELOPERS CONFERENCE® 2013
MARCH 25–29, 2013
GDCONF.COM

Example of one raycast from node-to-node being blocked by rather thin obstacle. This can be avoided by performing multiple raycasts between nodes and having a threshold to decide if there is visibility between nodes. For example 80% threshold would make most of those nodes marked as red become visible (green) until the very few at the distance.

Part I - Offline

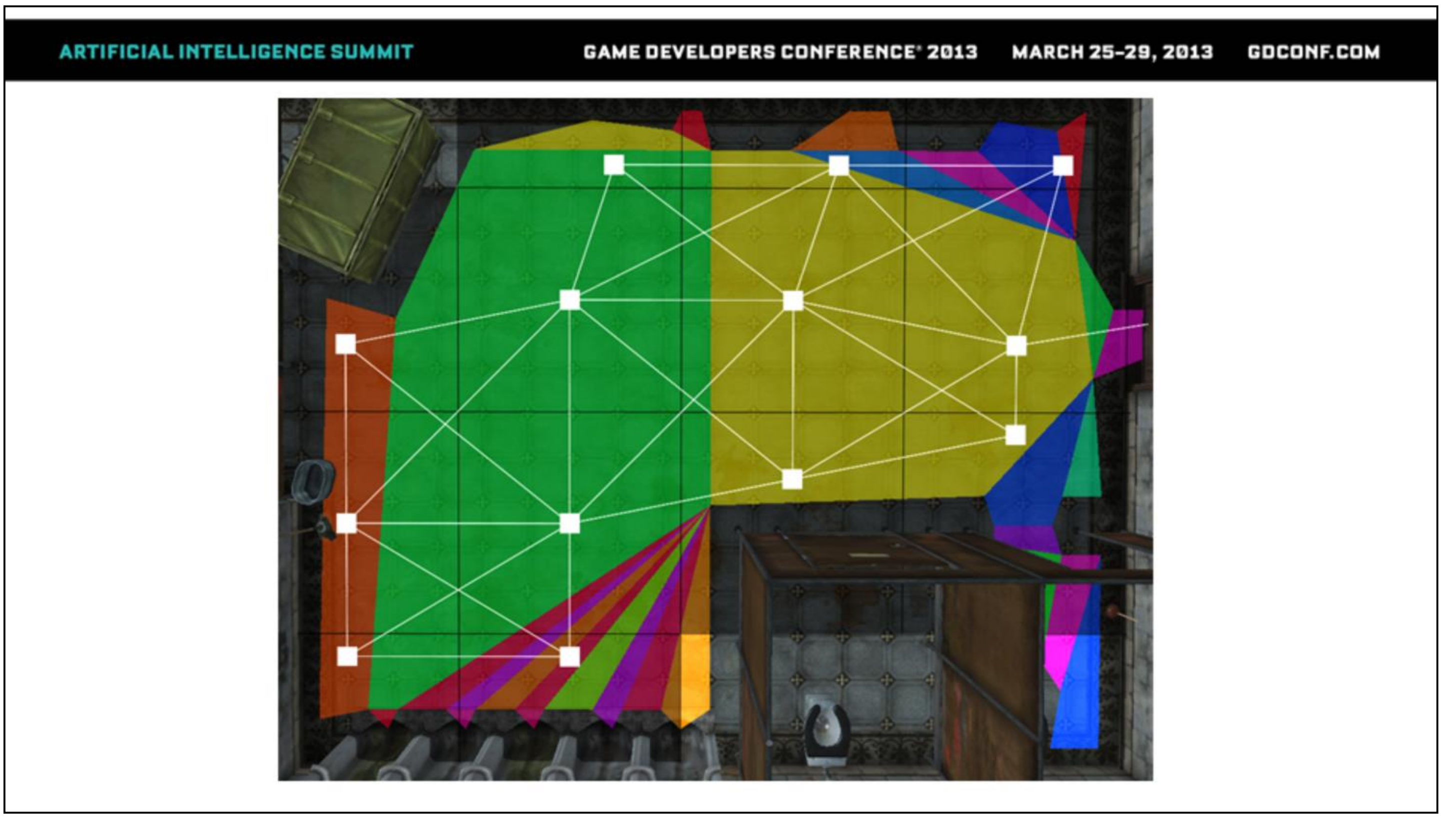

ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Discretizing the world

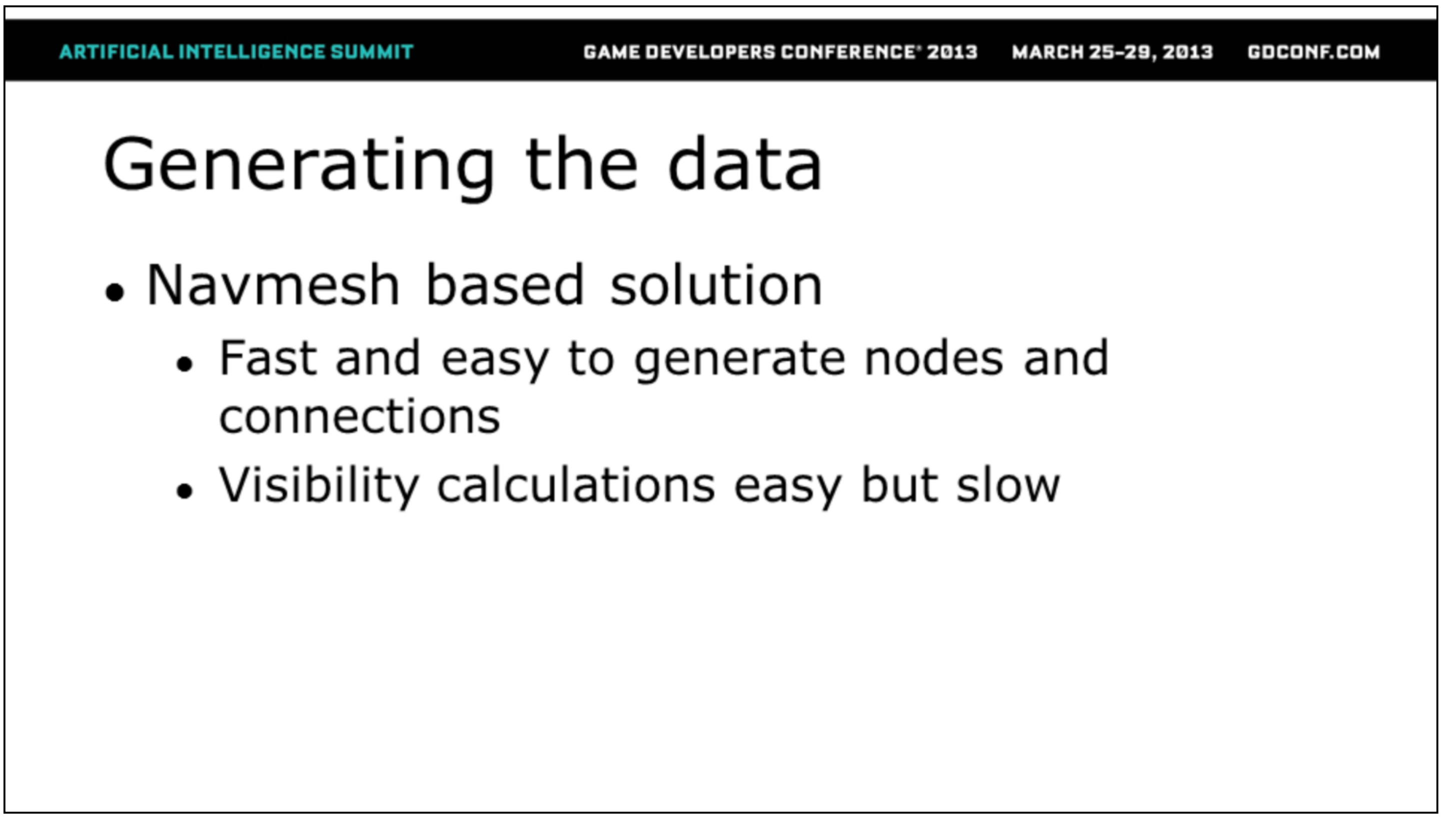
- Uniform grid
  - Fast to map
  - Regularity
    - Distance between nodes is constrained
    - Easily comparable data (area of influence)

Our solution is to discretize the world into cells, uniform grid, since it is fast to map any arbitrary location and it also makes the nodes and any data stored in them easily comparable as they all have roughly the same area of influence. In addition distance between nodes being constrained it means it is possible to use them as pretty good approximation of distance calculation as the nodes and their connections describe the topology closely matching that of the navmesh.

Nodes inside cells can still be freely placed so more detail can be achieved from this.


ARTIFICIAL INTELLIGENCE SUMMIT
GAME DEVELOPERS CONFERENCE® 2013
MARCH 25–29, 2013
GDCONF.COM

Example of a navmesh and grid on top of it. Colored polygons are the navmesh, black lines make the grid and white rectangles with white lines are the nodes and their connections. The width and height of the cells is 2.25m and as can be seen the nodes describe the area of the room closely matching that of the navmesh.

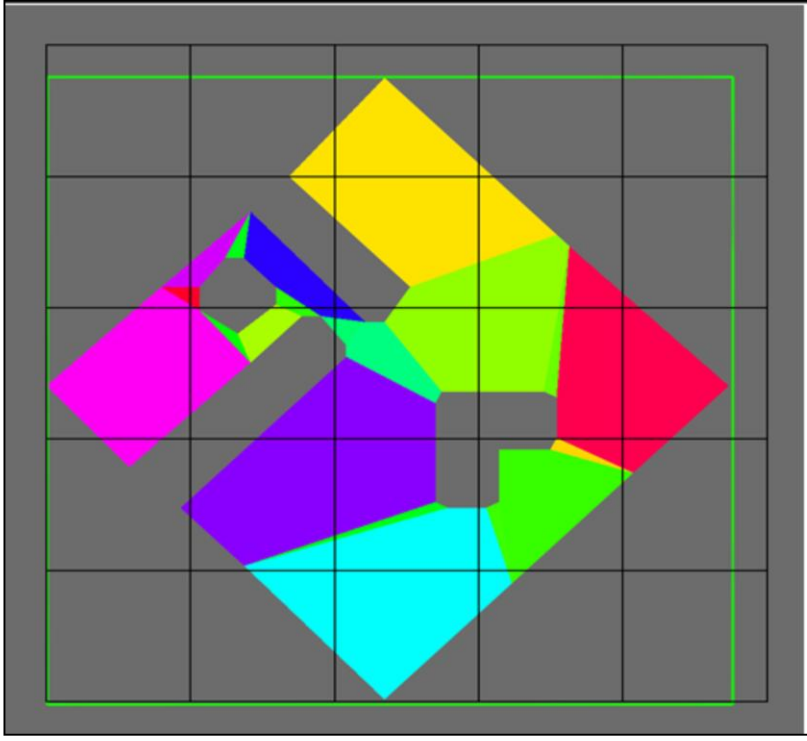

ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Generating the data

- Navmesh based solution
  - Fast and easy to generate nodes and connections
  - Visibility calculations easy but slow

Generating nodes for average level in HM:A takes only few seconds while generating the visibility with good quality takes up to 15minutes with a decent work station utilizing all cores.

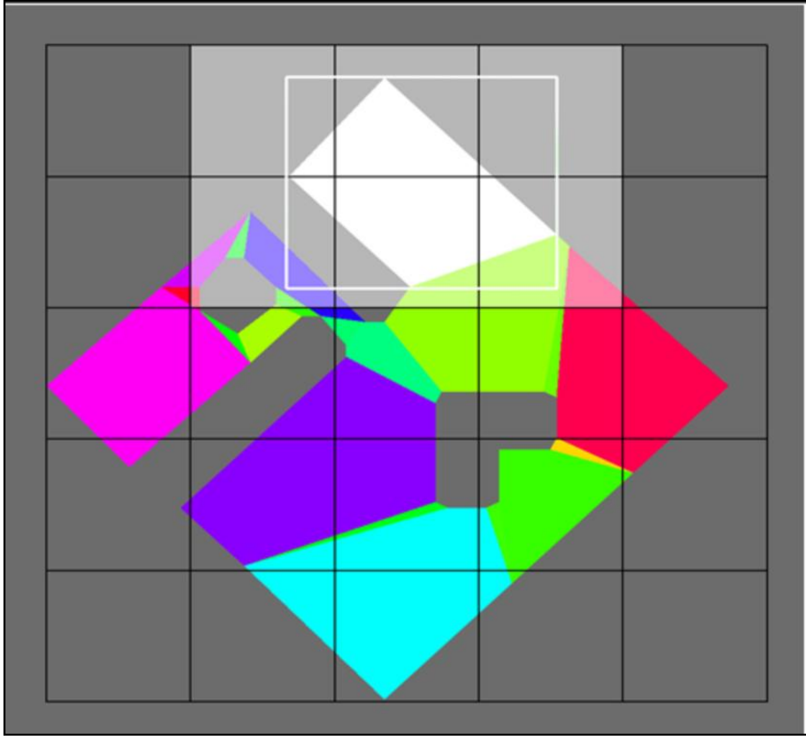
For debugging purposes it is good to have the possibility of skipping the visibility calculation part so that the node placement and creation algorithm is easy to debug.



Find grid dimensions

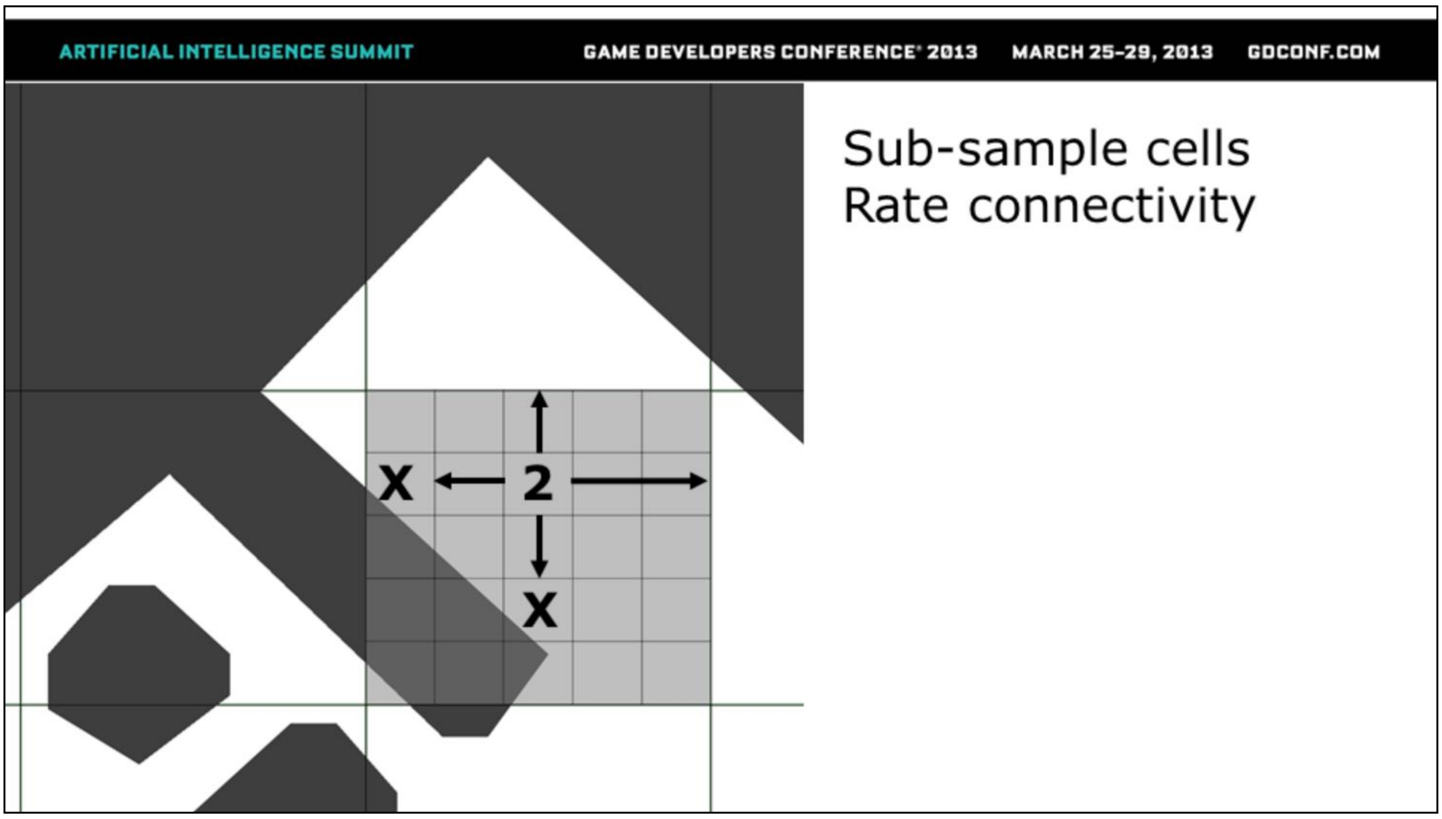
Can be fixed number of nodes which means the cell size will vary or can be fixed cell size to have varying number of nodes.

Find bounding box of navmesh use that to calculate the cell size and grid bounds.



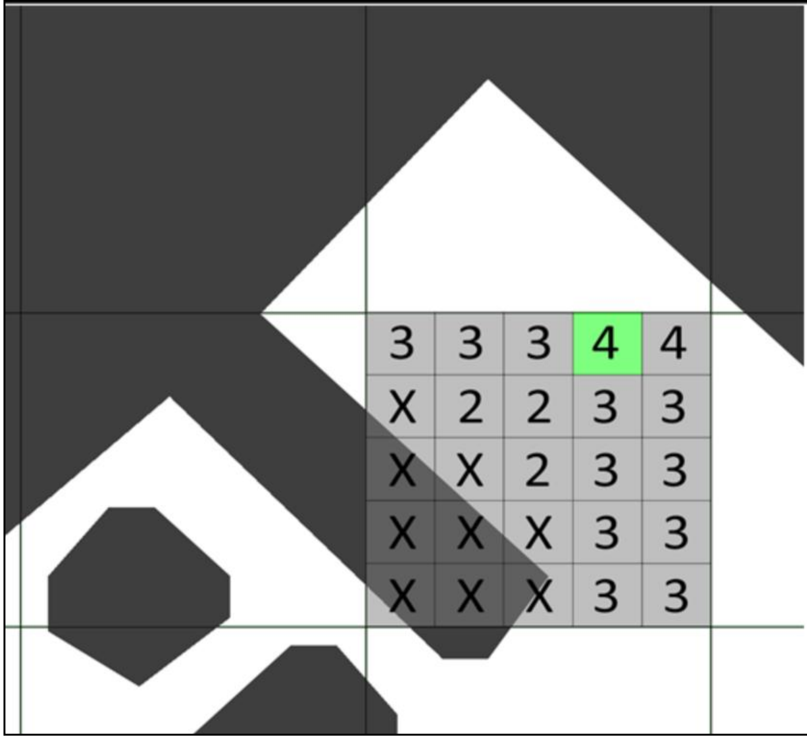
Iterate navmesh polys
Mark 'occupied' cells

Optimization step to quickly find the cells that actually can have a node.

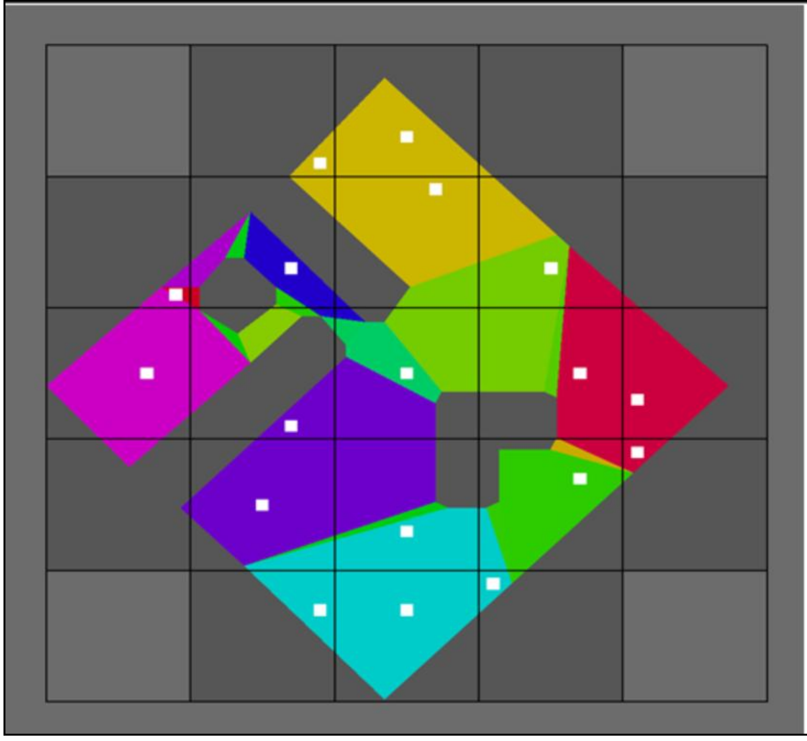


Sub-sample cells
Rate connectivity

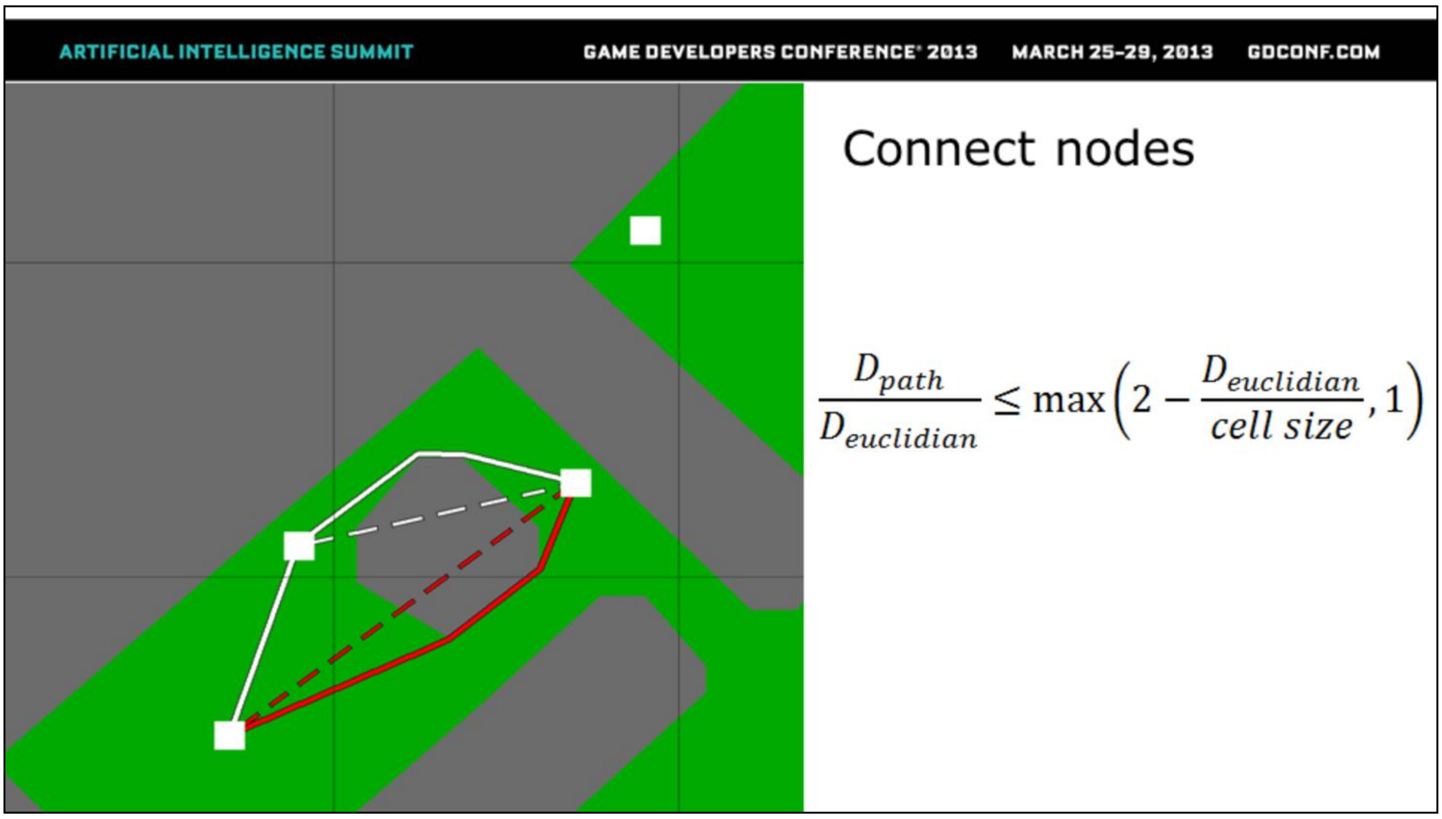
Find the exact node placement, trying to maximize connectivity to neighboring cells.



Pick highest scoring, closest to center



Node placement in cells



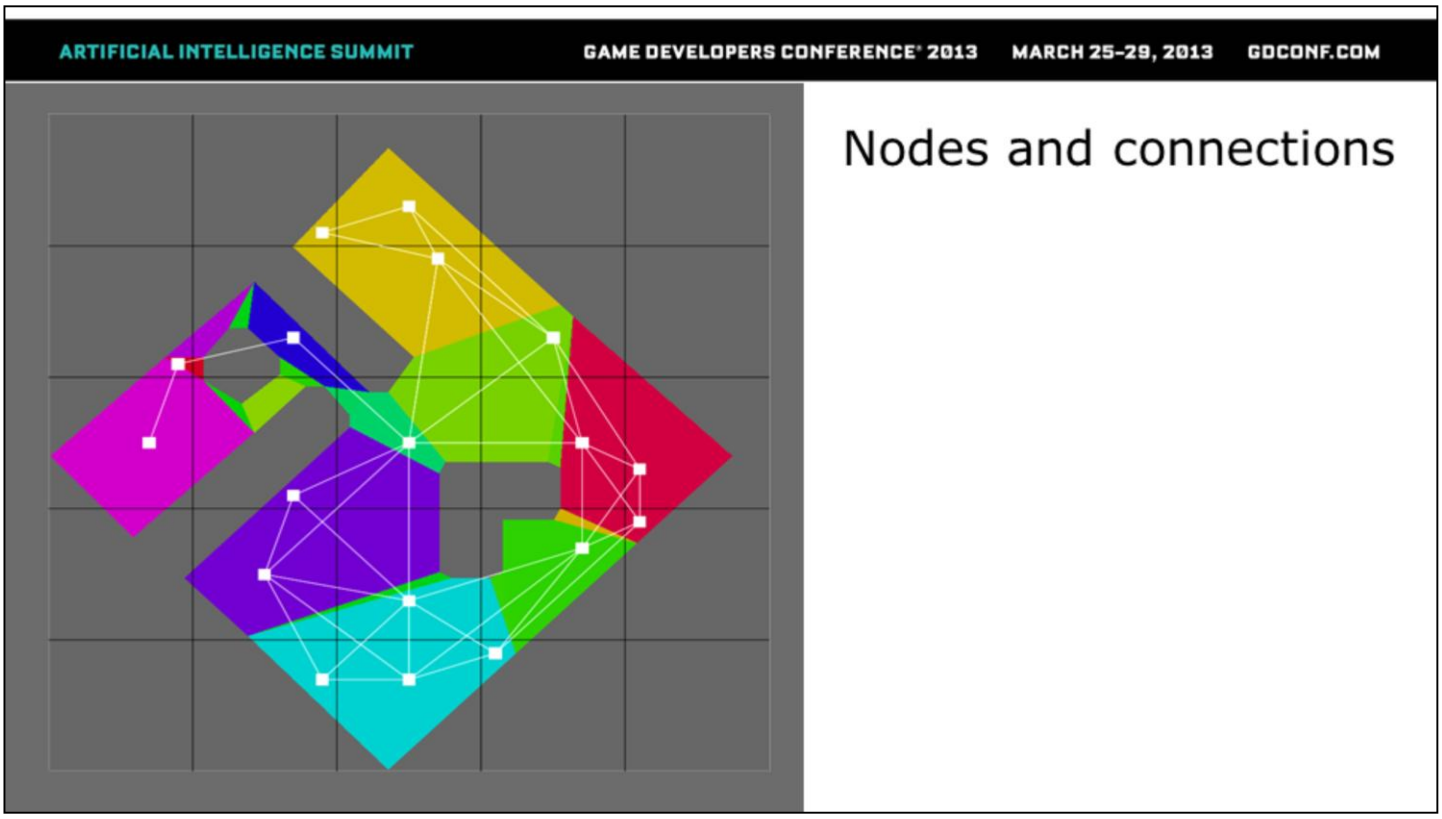
ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Connect nodes

$$\frac{D_{path}}{D_{euclidian}} \leq \max\left(2 - \frac{D_{euclidian}}{cell\ size}, 1\right)$$

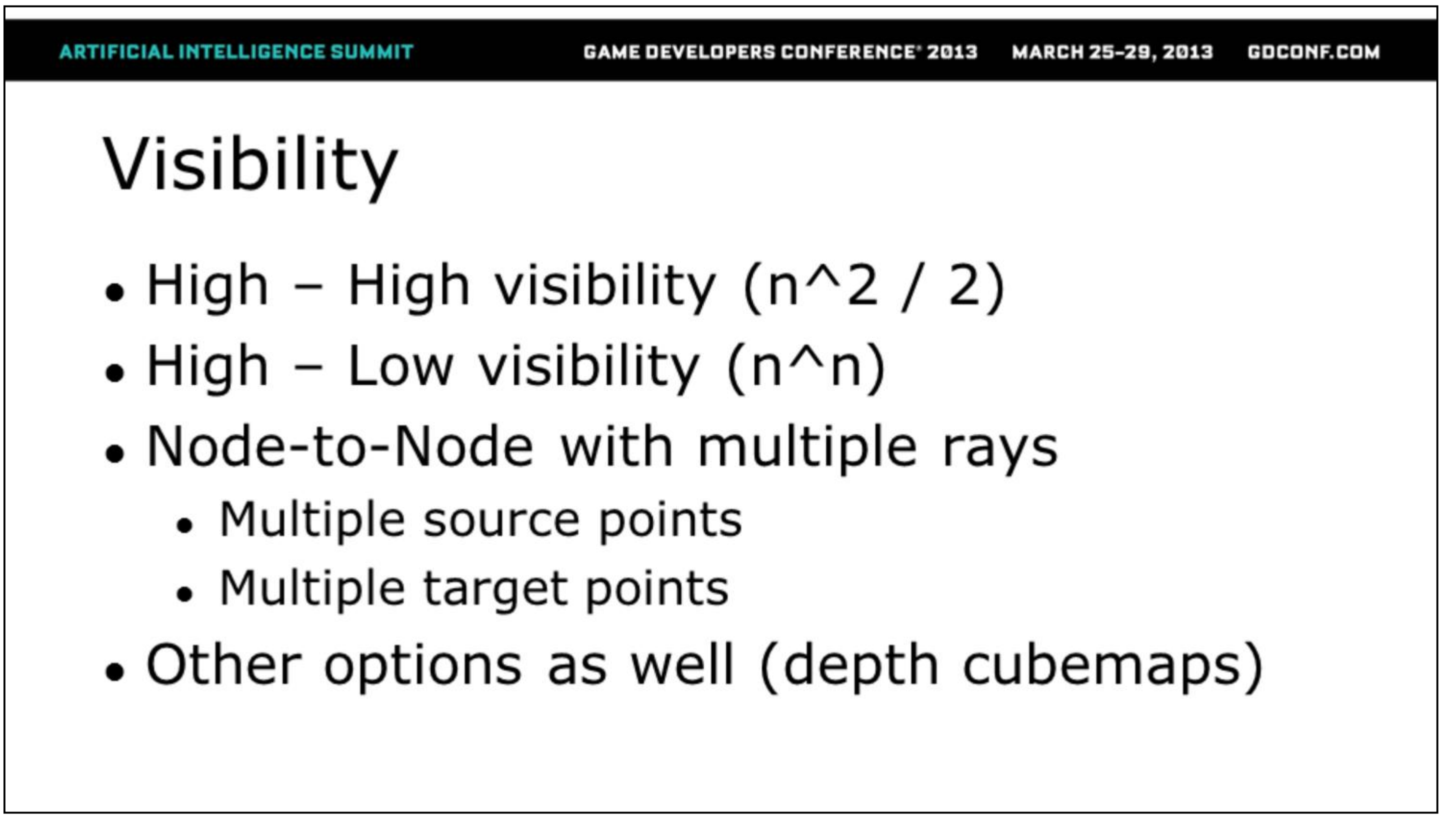
Allow connecting nodes even if no straight line in navmesh to deal with small navmesh corridors. With our 2.25m cell size there were quite a lot of these but most of the time the node placement would solve them.

Mostly this happens when the narrow navmesh corridors are not axis-aligned as the basic node placement would only scan in cardinal directions and the grid is always axis-aligned.


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

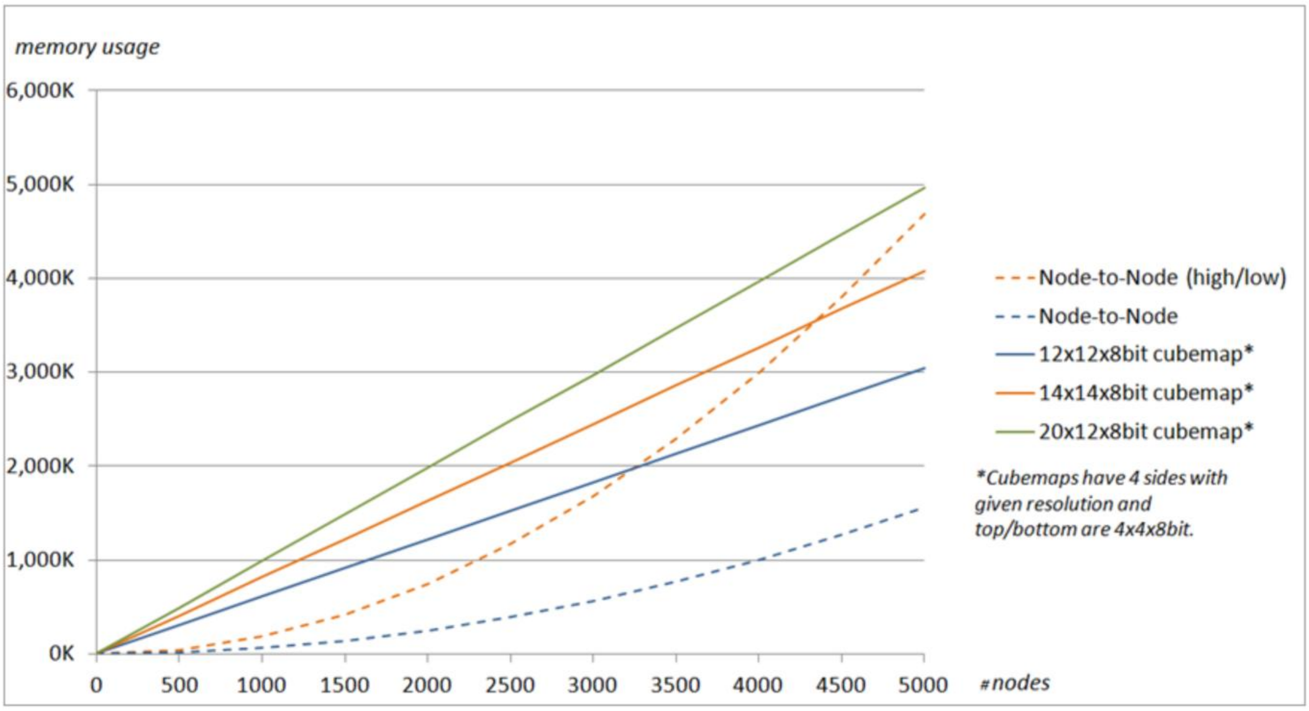
Nodes and connections

After this step you would perform a node-to-node visibility step simply issuing raycasts between the nodes. For ”high” visibility you need only n * n / 2 raycasts as A->B is the same as B->A while performing a ”high” to ”low” raycasts you need n * n as A->B is not the same as B->A. Here you should try to do more than one ray between two nodes as it will result in better description of the visibility in terms of tactical decisions.


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

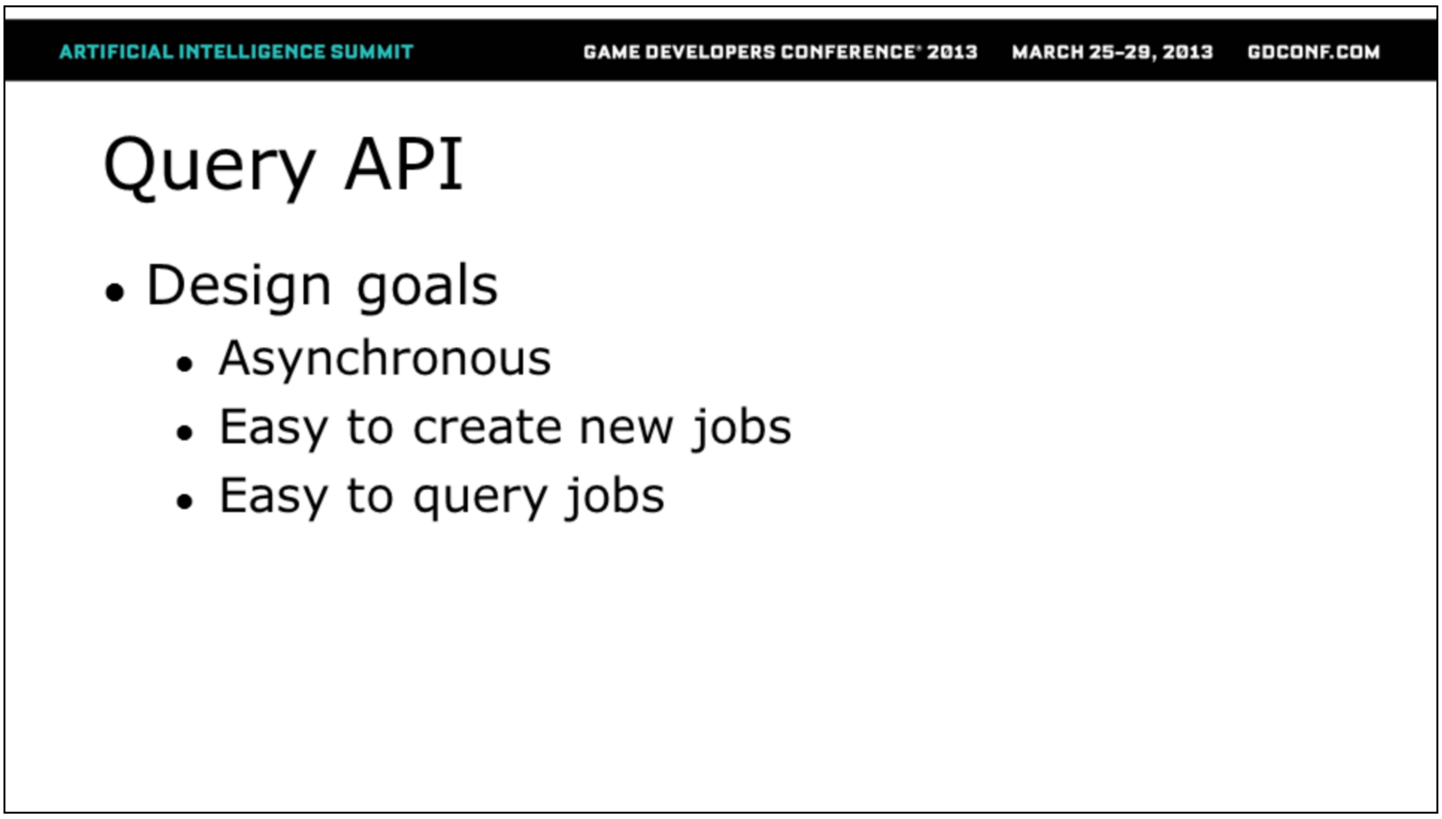
Visibility

- High – High visibility (n^2 / 2)
- High – Low visibility (n^n)
- Node-to-Node with multiple rays
  - Multiple source points
  - Multiple target points
- Other options as well (depth cubemaps)



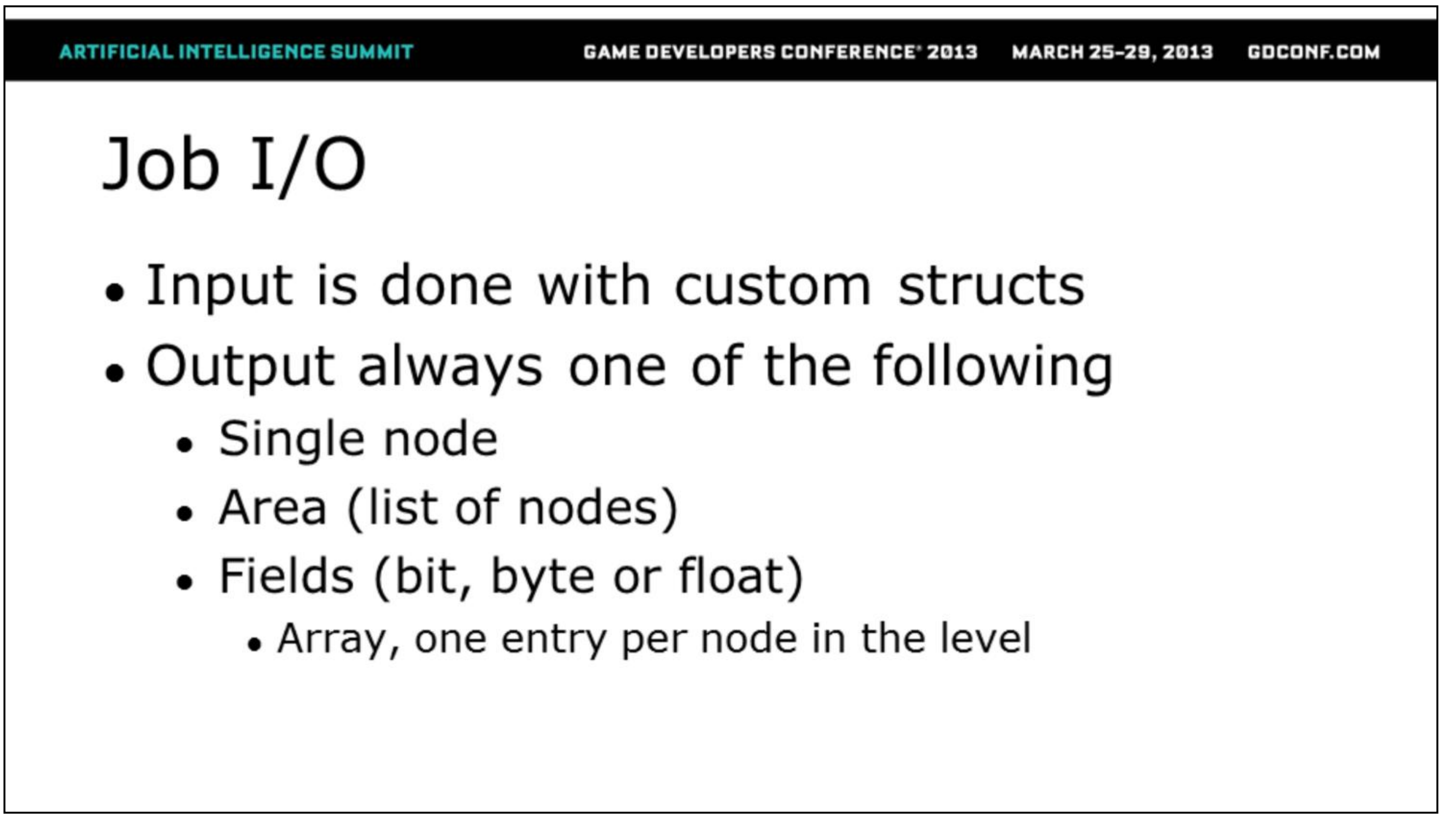
*Cubemaps have 4 sides with given resolution and top/bottom are 4x4x8bit.

Part II - Runtime



Query API

- Design goals
    - Asynchronous
    - Easy to create new jobs
    - Easy to query jobs

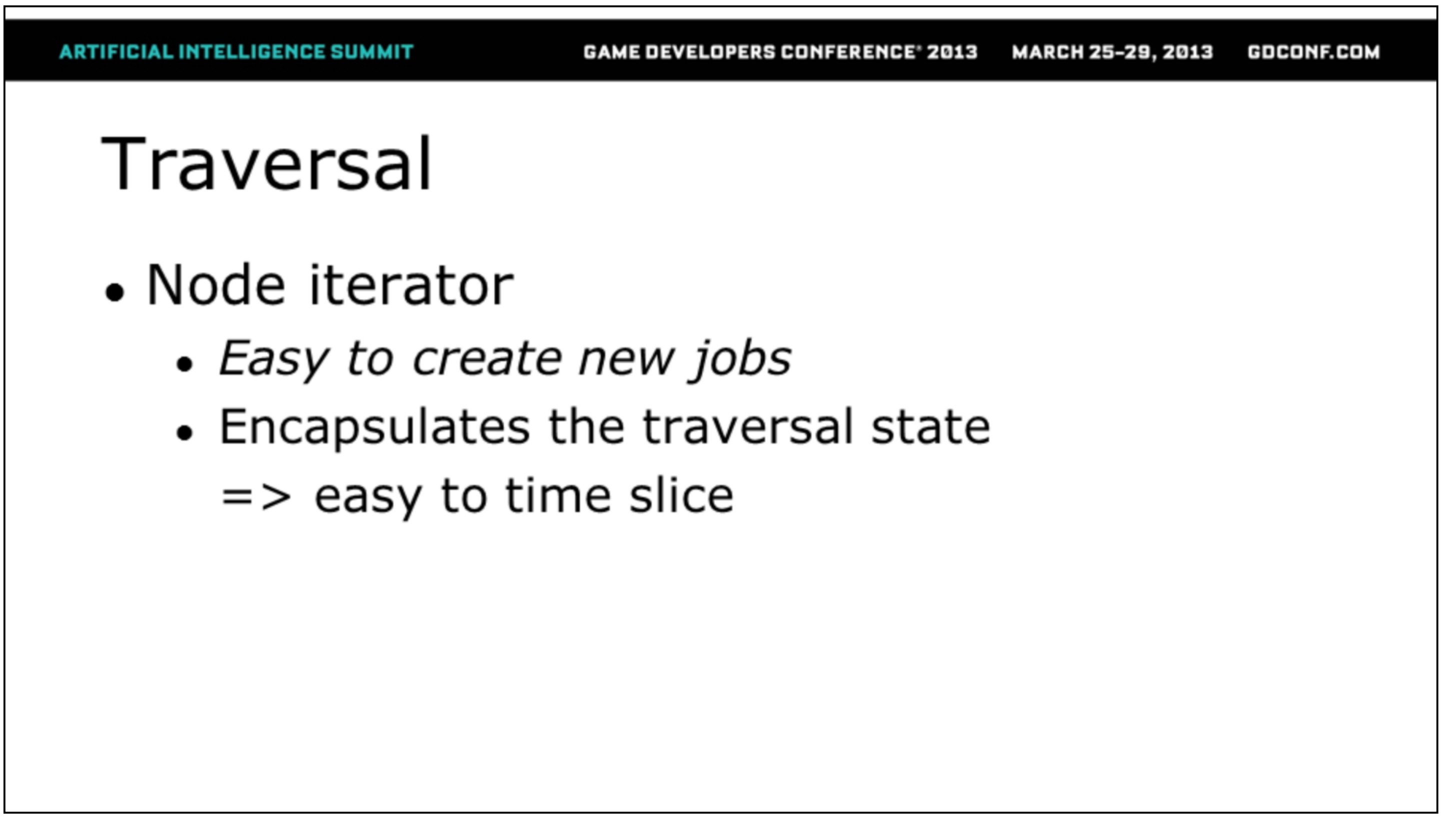

ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Job I/O

- Input is done with custom structs
- Output always one of the following
  - Single node
  - Area (list of nodes)
  - Fields (bit, byte or float)
    - Array, one entry per node in the level

Each job had a custom struct that they take in driving the processing, nothing special about that.

Output would always be one of the three types; fields, area or single node. Most of the jobs were fields and the final selection process was done as a 'node-job' which would pick the best rating, or closest with good enough rating etc. Jobs could take as input something that other jobs would output, this was often the case with the selection jobs.

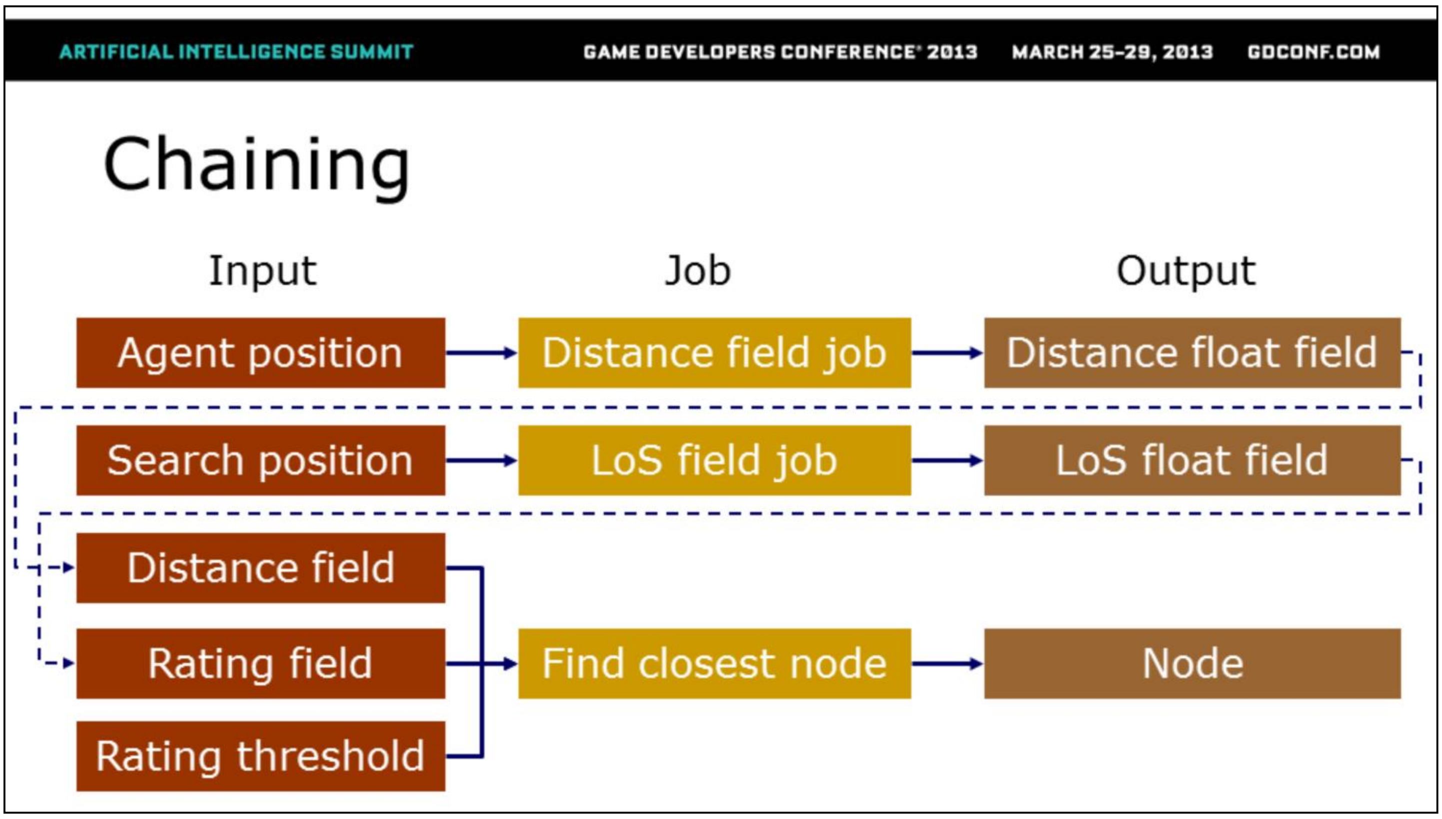


Traversal

- Node iterator
  - *Easy to create new jobs*
  - Encapsulates the traversal state
    => easy to time slice

To make it easy to implement new jobs and have them easily be time sliced a node iterator was done to encapsulate the traversal state and deal with the advancing in the grid.

Breadth-first was used since it proved to be optimal to create distance fields.


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Chaining

Three simple jobs that perform a straightforward task is combined to form a more complex query.

Find the closest node, in path-finding distance to agent, that has line-of-sight to search position. The 'Find closest node' would start from the 'Search position' which means that the node that will be returned has line-of-sight to the search position and is within x meters from the search position and is the one closest to the Agent.

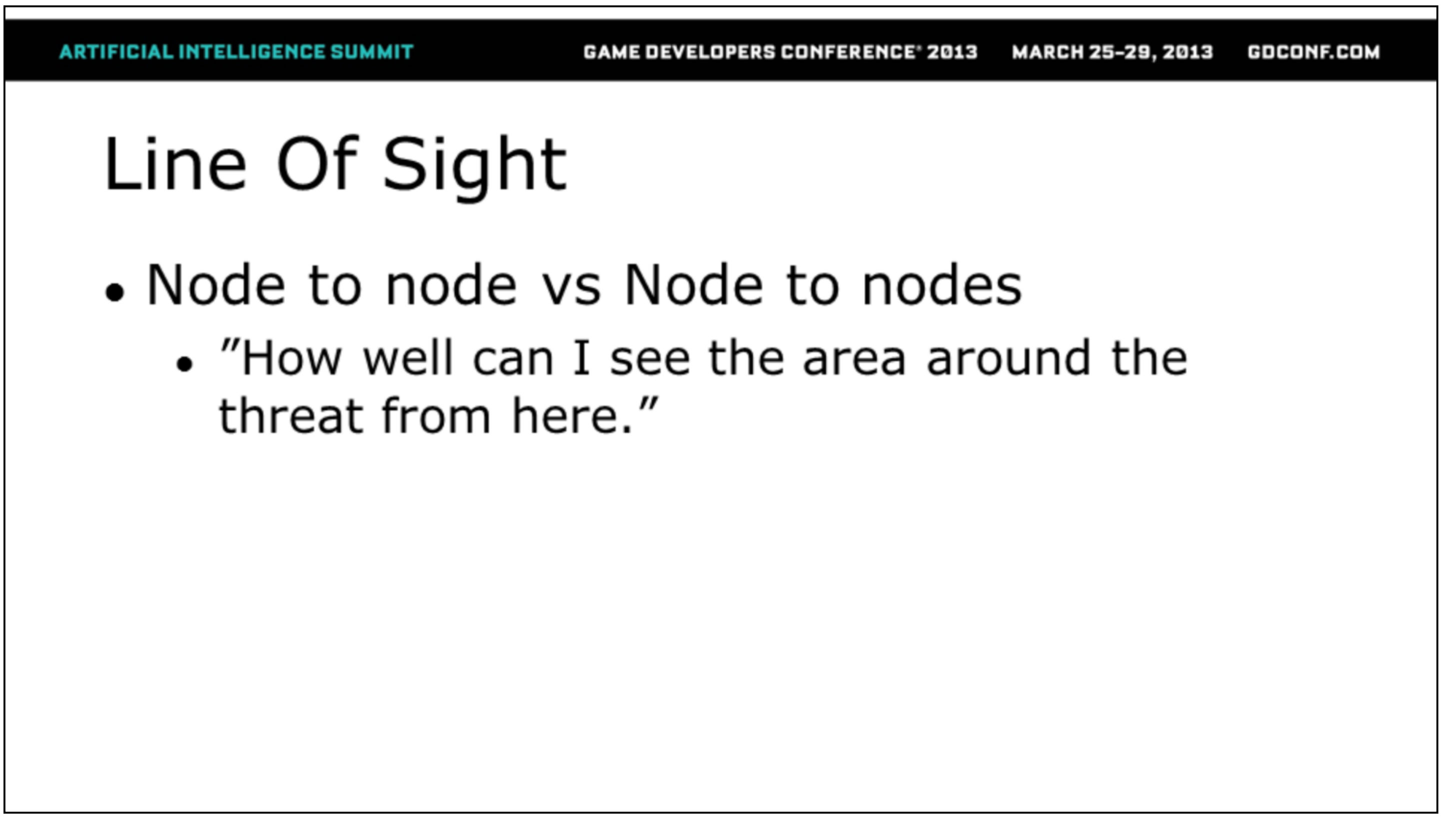
ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25–29, 2013 GDCONF.COM

Part III - Examples

Combat

- Utilizes chaining of jobs a lot
- Most jobs are shared data between NPC's

Some information can be shared and is updated parallel to the NPC specific jobs. Using double buffering of the float fields means that there is always data to be read even when update is in progress.


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

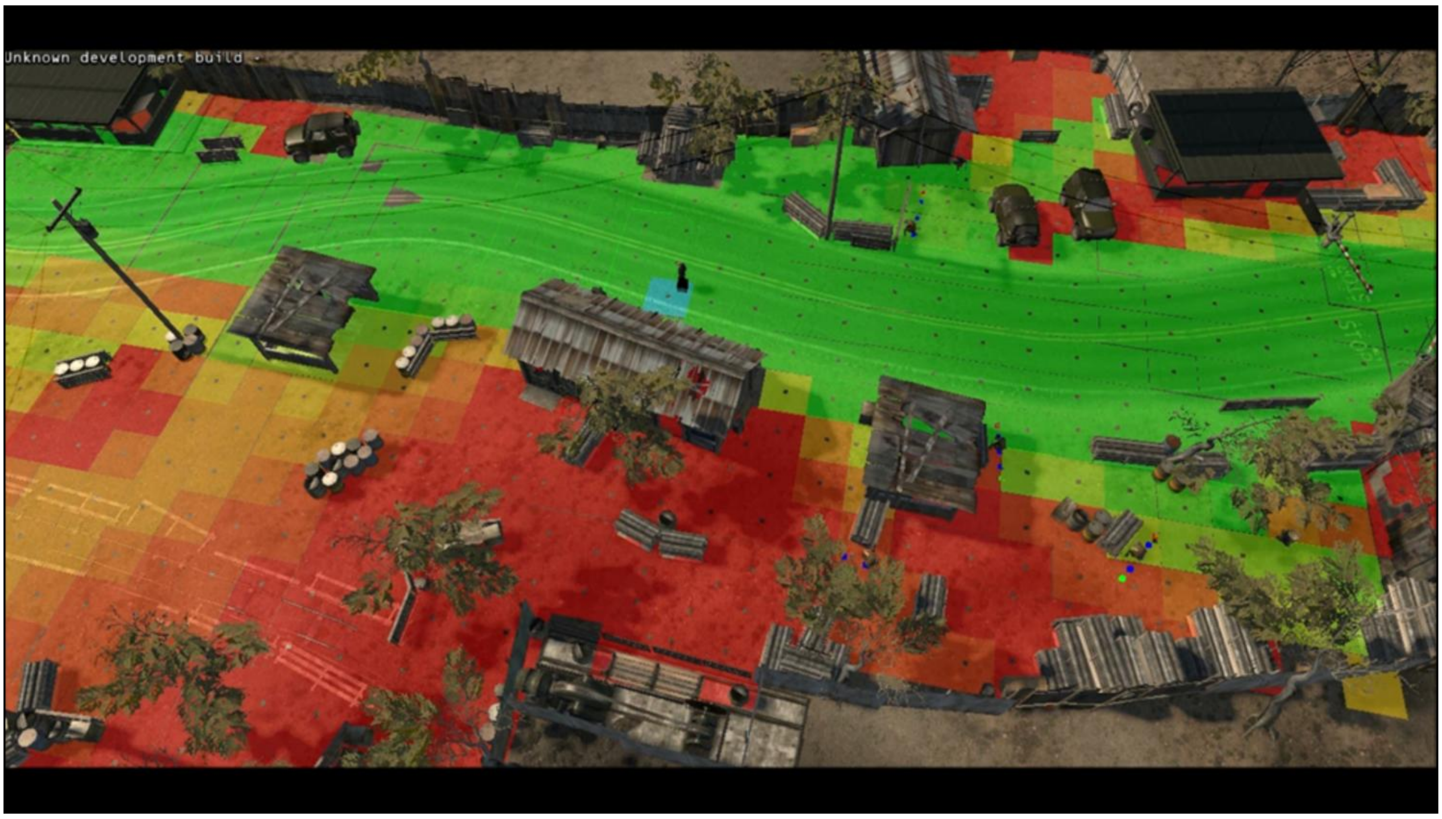
Line Of Sight

- Node to node vs Node to nodes
  - ”How well can I see the area around the threat from here.”

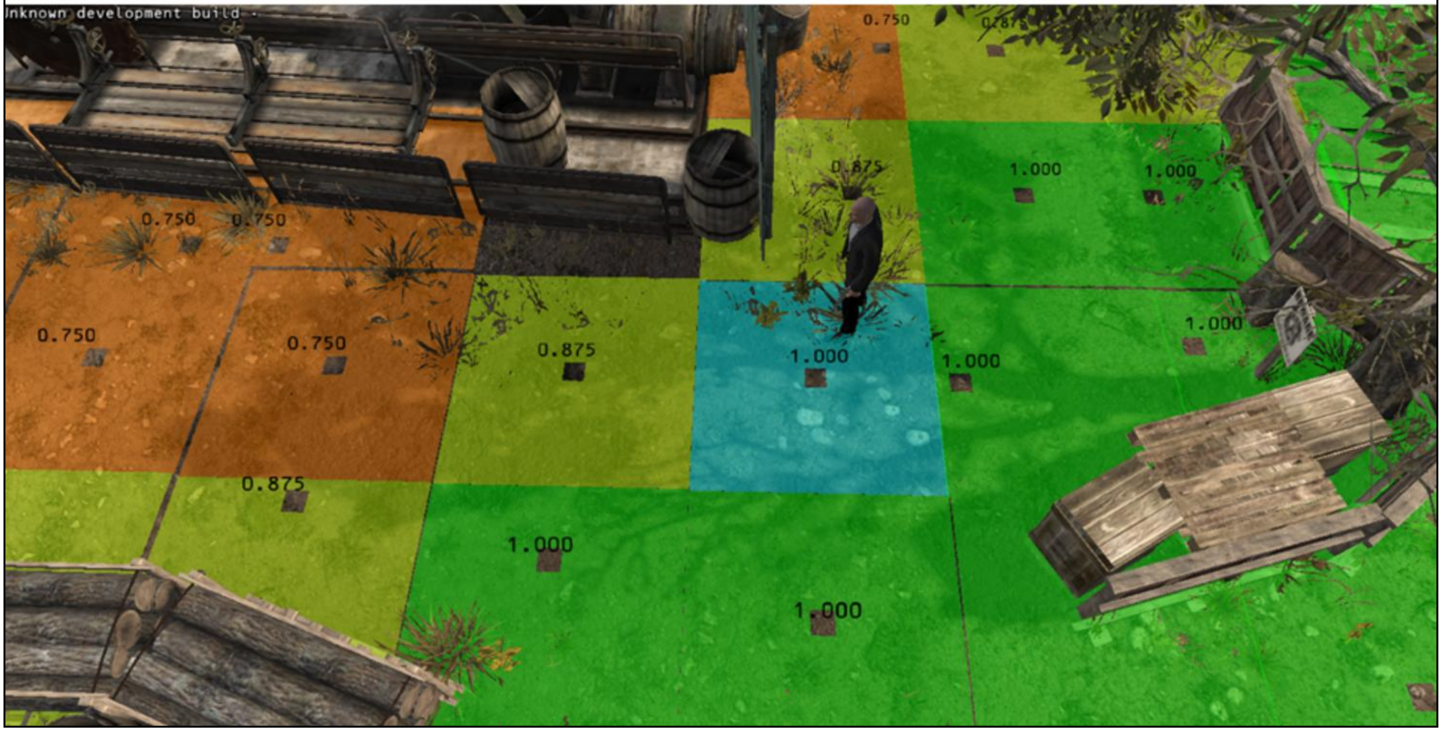
Most of the time NPC is interested to have a float field which describes how well can certain area be seen from each node. Float value of 0 means no visibility where 1 would mean full visibility over the given area.

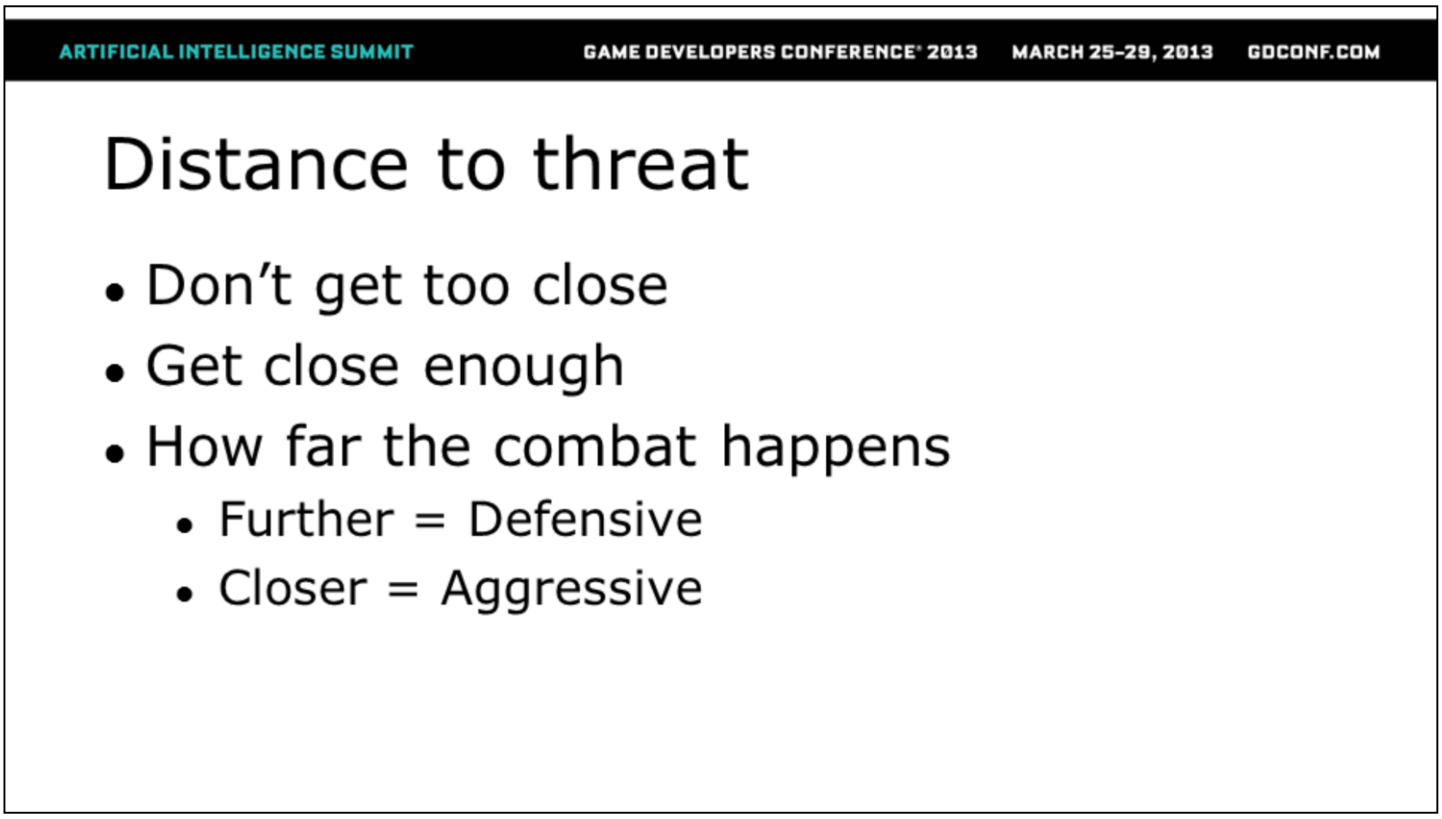
When performing the LoS job area size is given and it means that from each node multiple nodes around each of the target node are being checked. If there are 10 nodes within the target area then seeing 5 of them would give 50% visibility. Using this approach means that it is less likely to end up in a situation where query is being made, NPC goes to location only to see that situation has changed and there is no LoS anymore.

This can be updated constantly without any specific request from individual NPC as it is calculated to threat position.


Unknown development build

ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25–29, 2013 GDCONF.COM



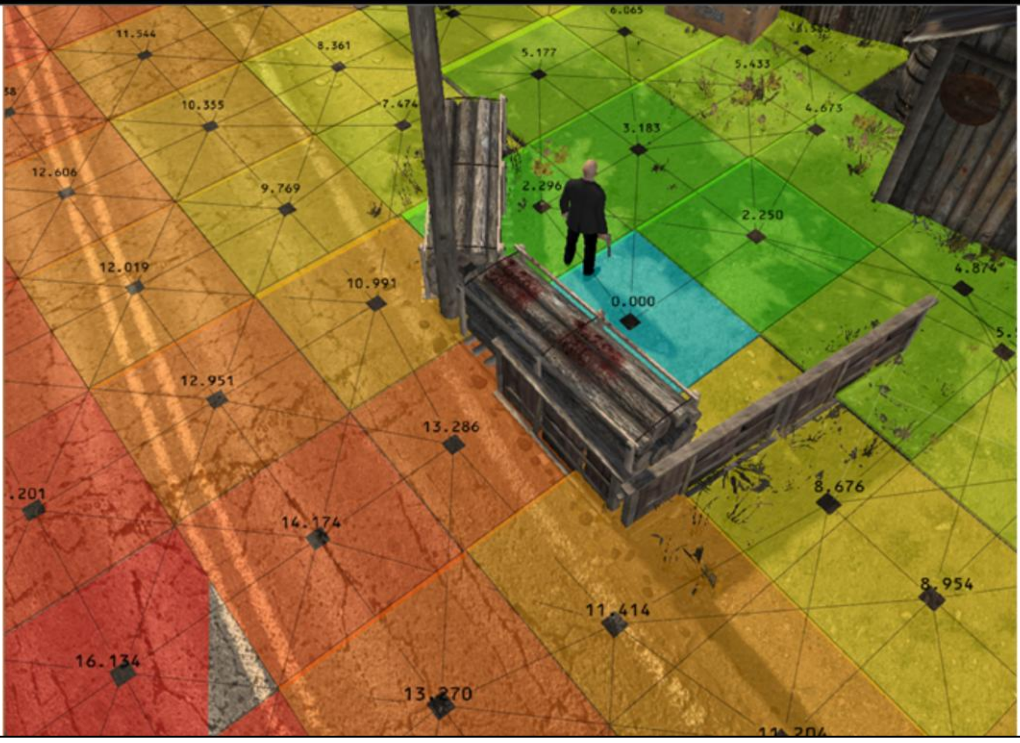


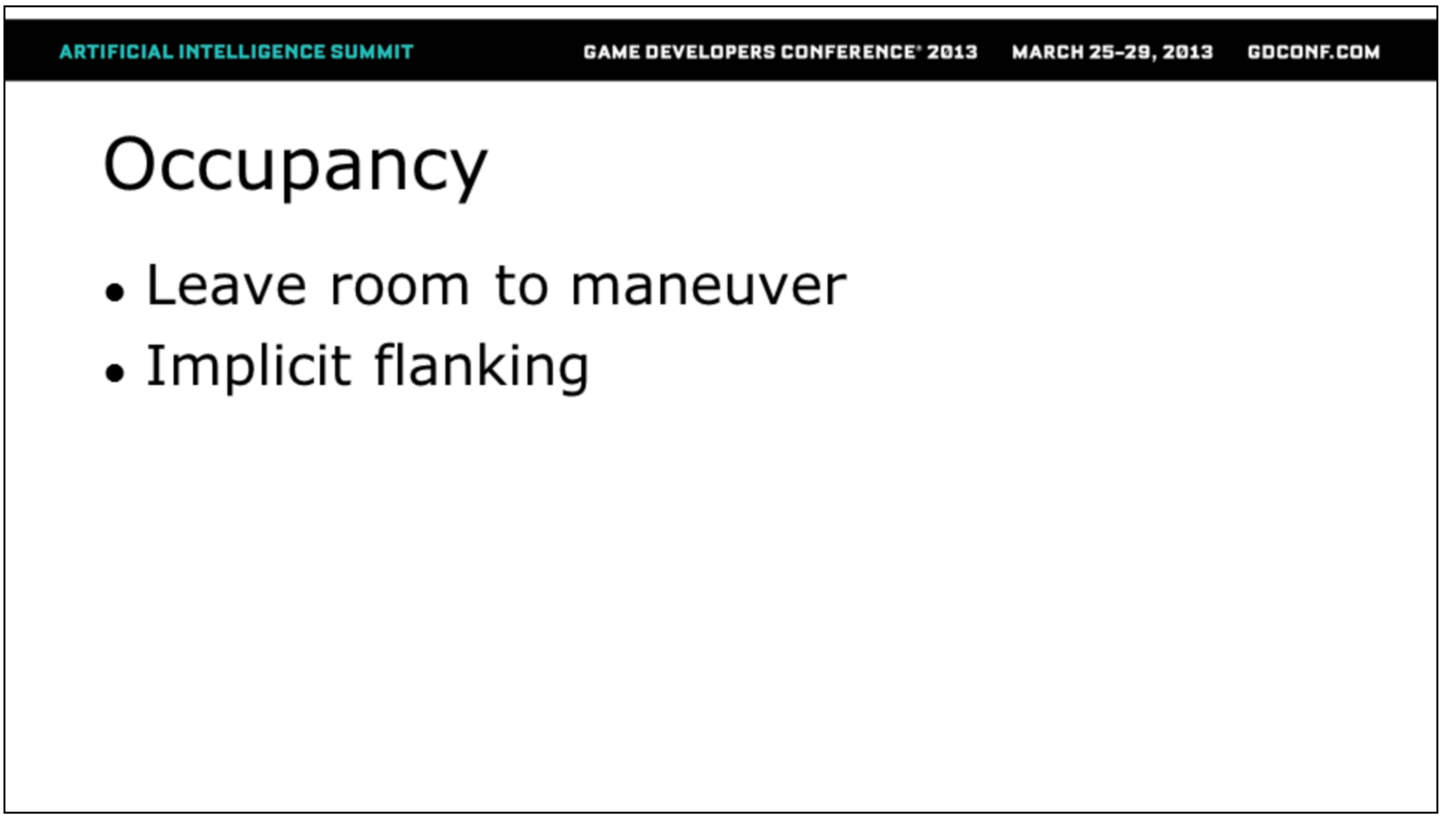
Distance to threat

- Don't get too close
- Get close enough
- How far the combat happens
  - Further = Defensive
  - Closer = Aggressive

Using distance field to threat to seek preferred distance where to fight. Changing this preferred distance is easy way to control the nature of the combat, defensive vs aggressive. This is path-finding distance and not euclidian distance as NPC's seek to be able to run after the threat and keep themselves not too close for target to run past or at them.

This can be updated constantly without any specific request from individual NPC as it is calculated from threat position.


6.065
11.544
8.361
5.177
5.433
10.355
7.474
3.183
4.673
12.606
9.769
2.296
2.250
12.019
10.991
0.000
4.874
12.951
13.286
.201
14.174
8.676
11.414
8.954
16.134
13.270


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

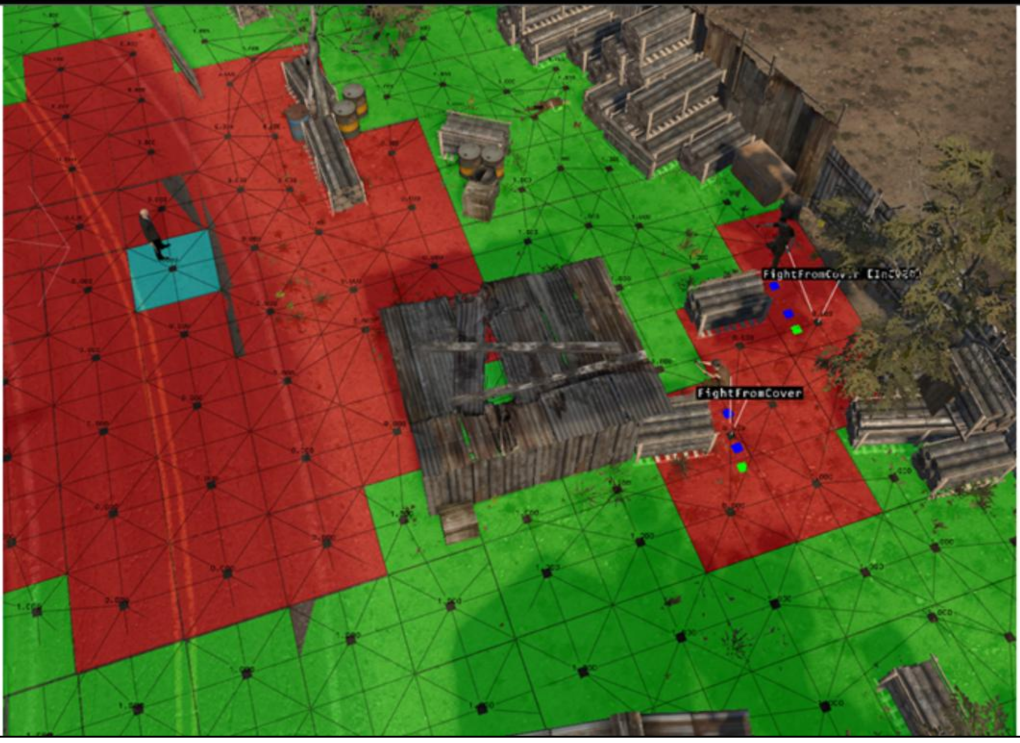
Occupancy

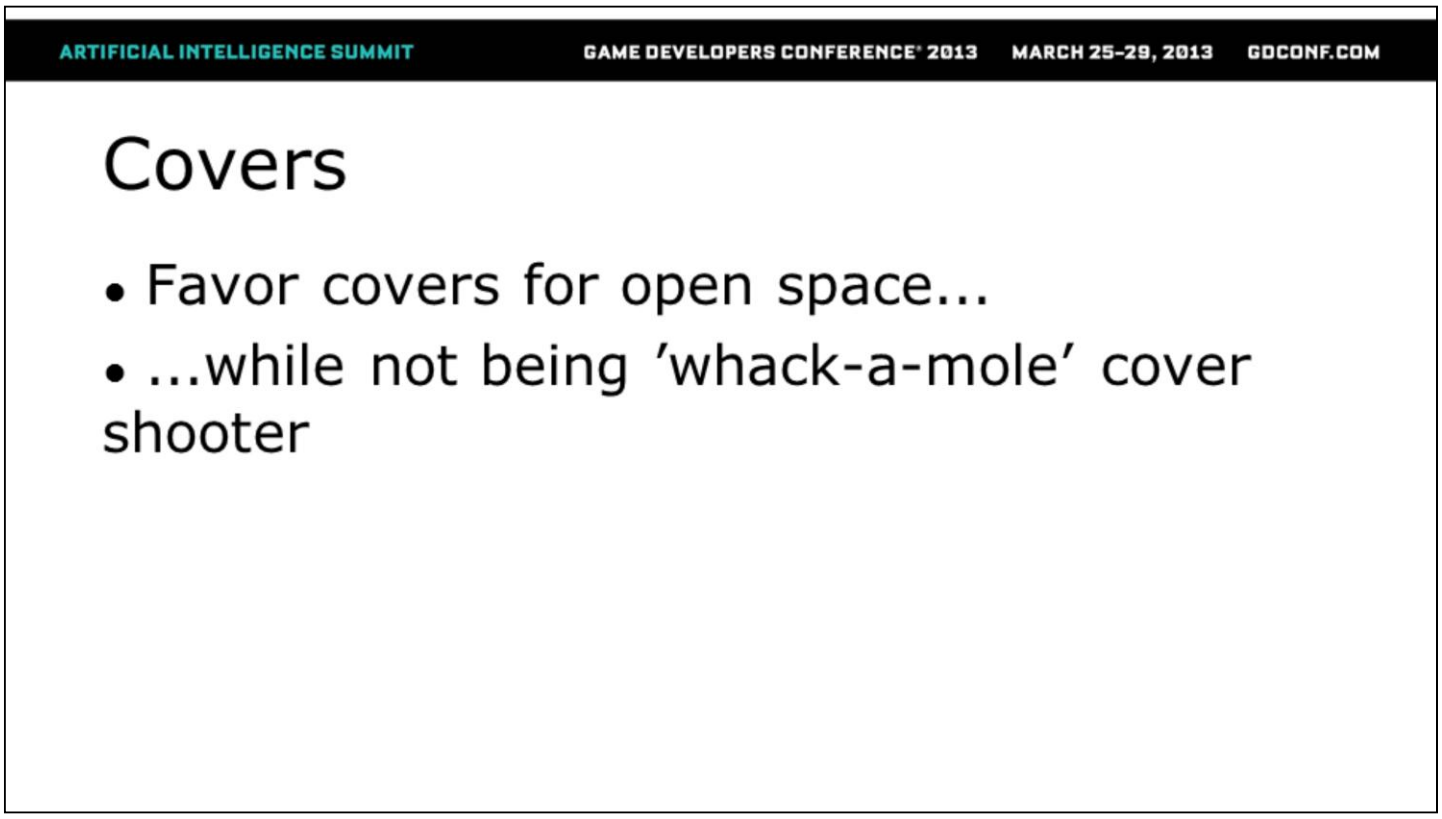
- Leave room to maneuver
- Implicit flanking

This tells which nodes are close to an NPC or close to the threat. Will cause NPC's to spread around and in combination with the preferred distance to fight they will start to flank eventually.

HM:A did not have a specific flanking behavior but it was triggered due to occupancy and hazard zones invalidating nodes in front of the player.

This gets updated regularly and is shared between NPC's since it is not dependent on any one NPC.


FightFromCover [InCover]
FightFromCover


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

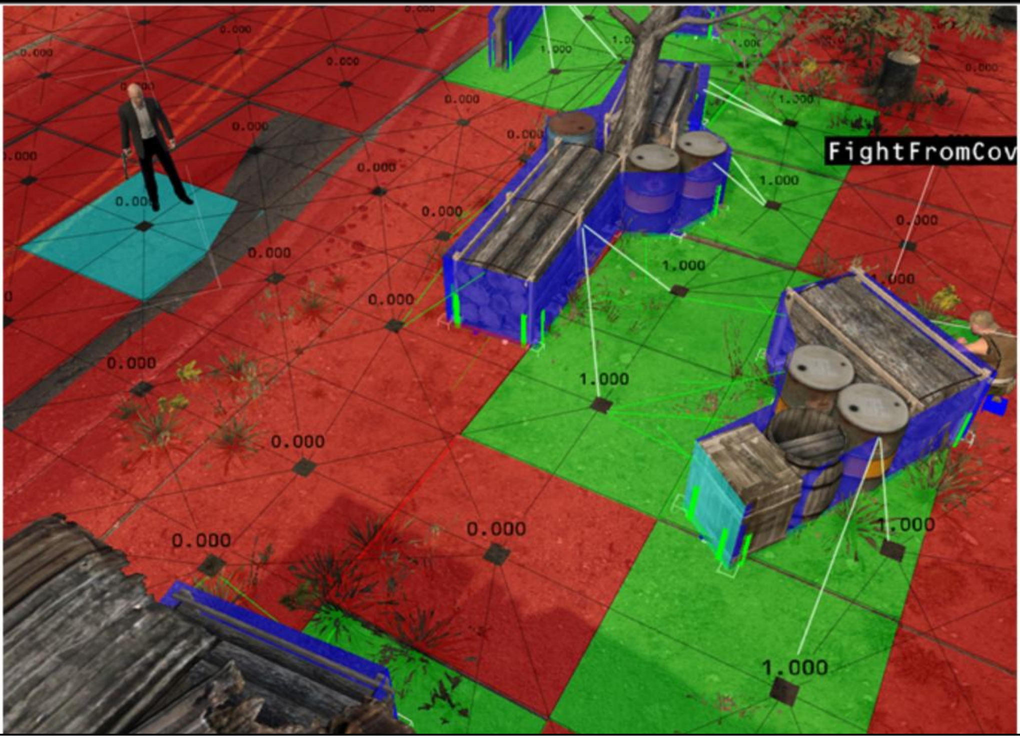
Covers

- Favor covers for open space...
- ...while not being 'whack-a-mole' cover shooter

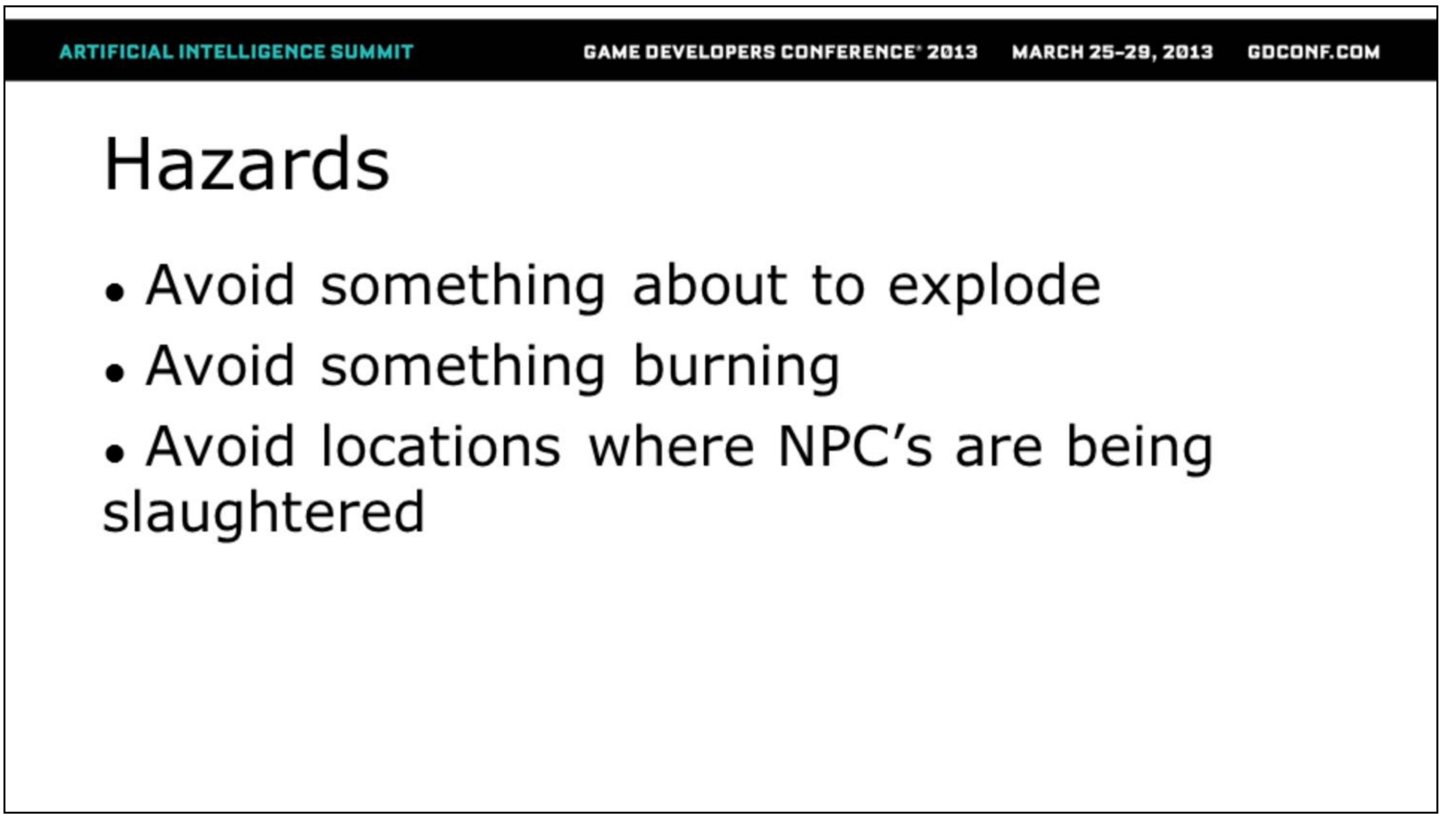
Cover map tells which covers would currently, given the threat position, give cover from the threat. Any node being close enough to a cover which is valid at the moment will have value of 1 where others will have value of 0. Since this is just one value amongst all the others it will not mean that NPC always takes cover even though they clearly favor them.

It is possible to have a node without valid nearby cover that is the best node given the current situation which leads into NPC fighting from open.

This is also shared between NPC's and is not dependant on individual NPC request as only threat position will affect how this field changes.



While lines show associated covers to the nodes while green means connection to a valid cover.

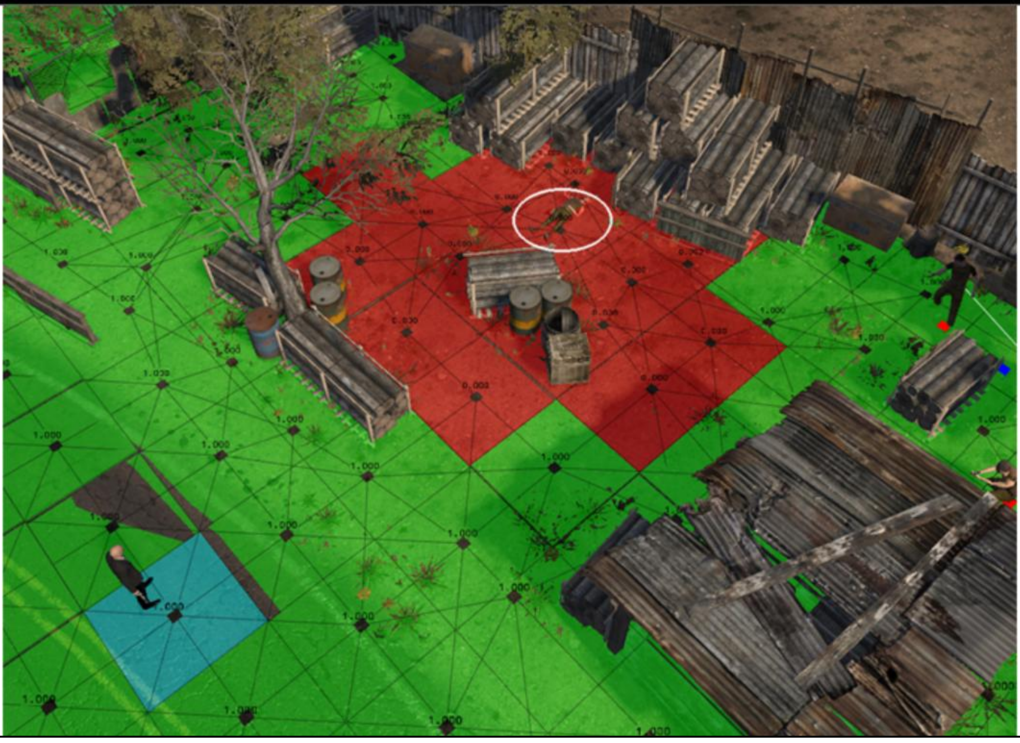


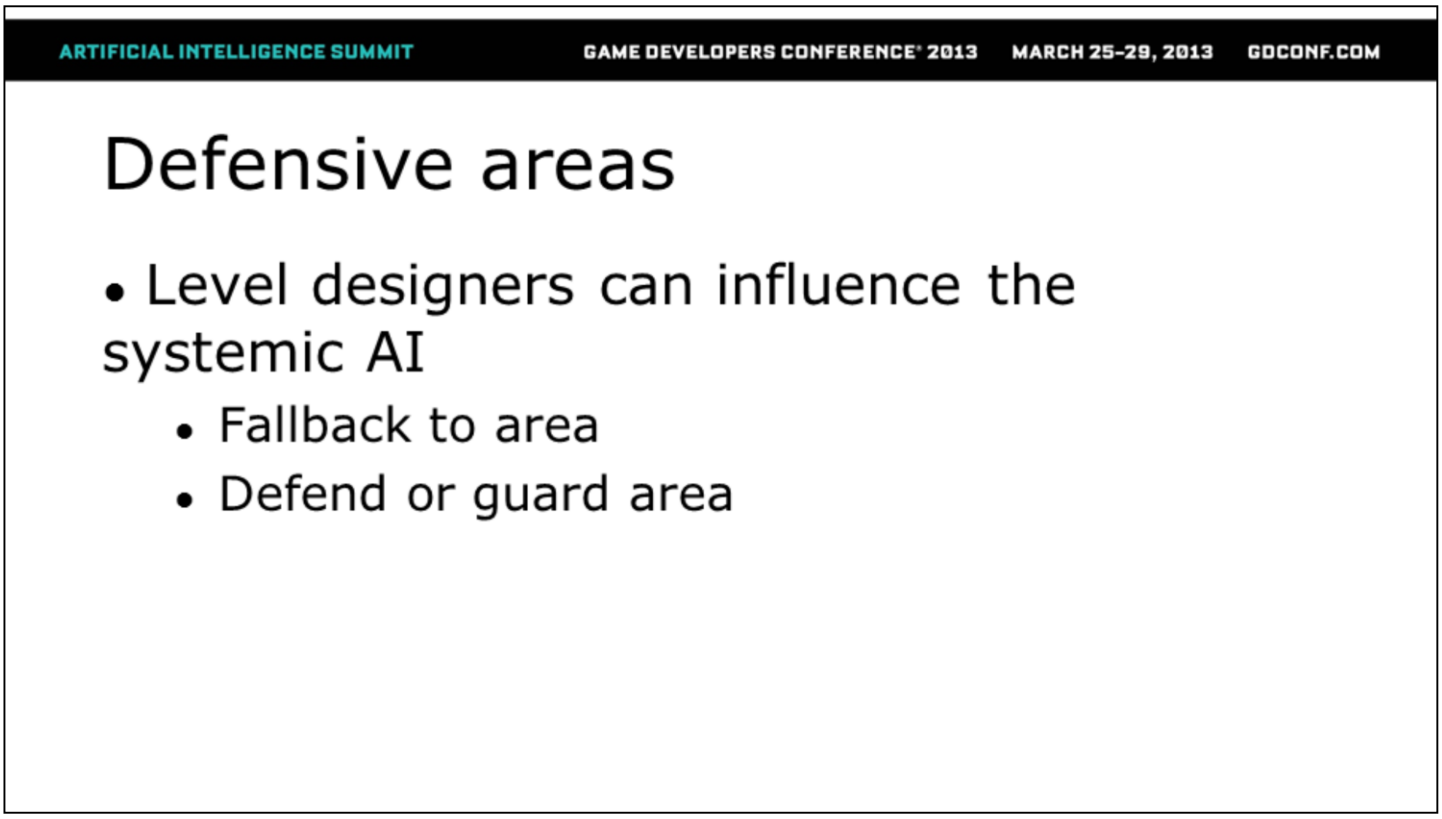
Hazards

- Avoid something about to explode
- Avoid something burning
- Avoid locations where NPC's are being slaughtered

This is a byte field keeping count of hazard zones currently active. Hazard zones are simply a point with a radius and a timer. Everytime a new zone gets activated it will increase the number on each of the nodes within its influence by 1 and when it gets deactivated it will decrease the number on each of the nodes within its influence.

This means that when hazard field has a number higher than 0 the node is currently associated with at least one active hazard zone. Easy way to keep information about spatial influence without needing to iterate through all hazard zones everytime a node is being evaluated.



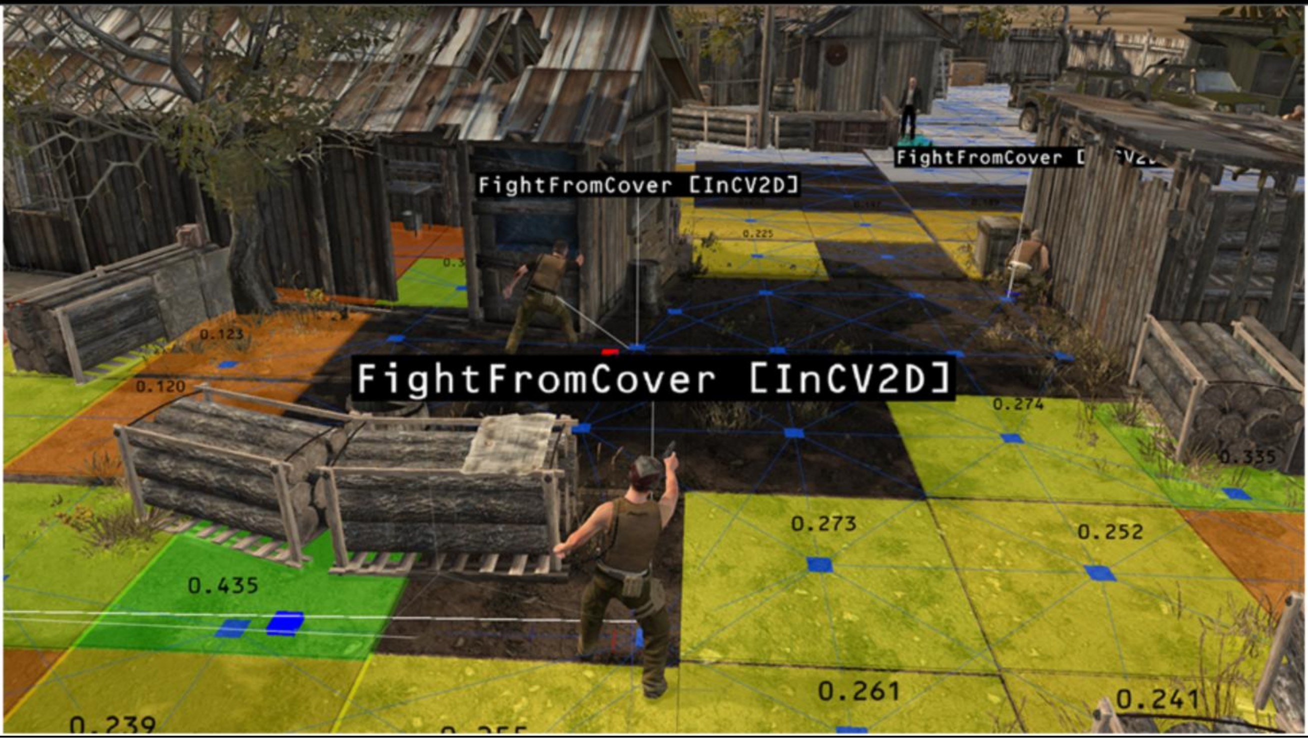

ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Defensive areas

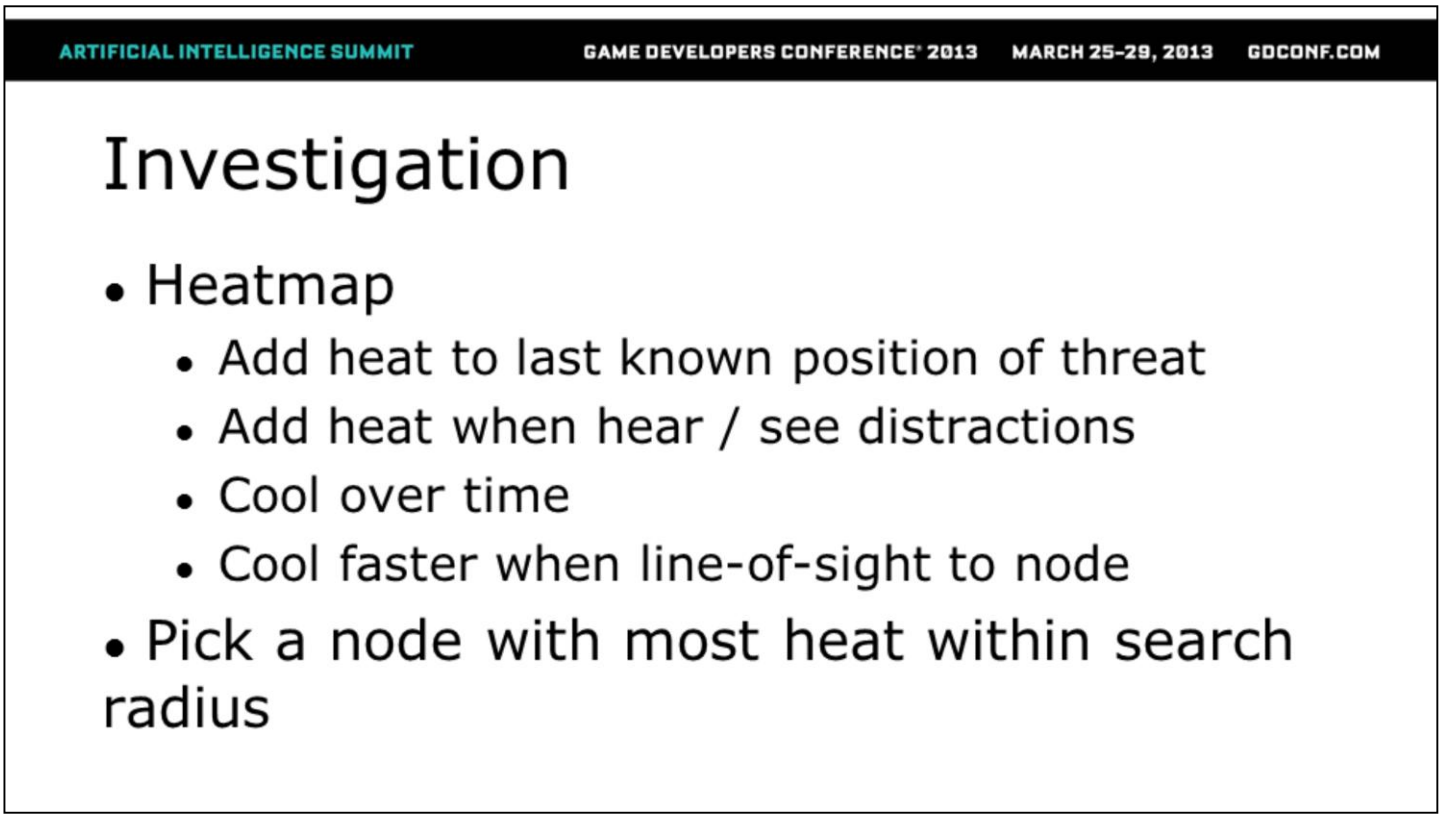
- Level designers can influence the systemic AI
  - Fallback to area
  - Defend or guard area

Simple control for designers to create guidance for positioning for NPC's operated by systemic AI.

ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

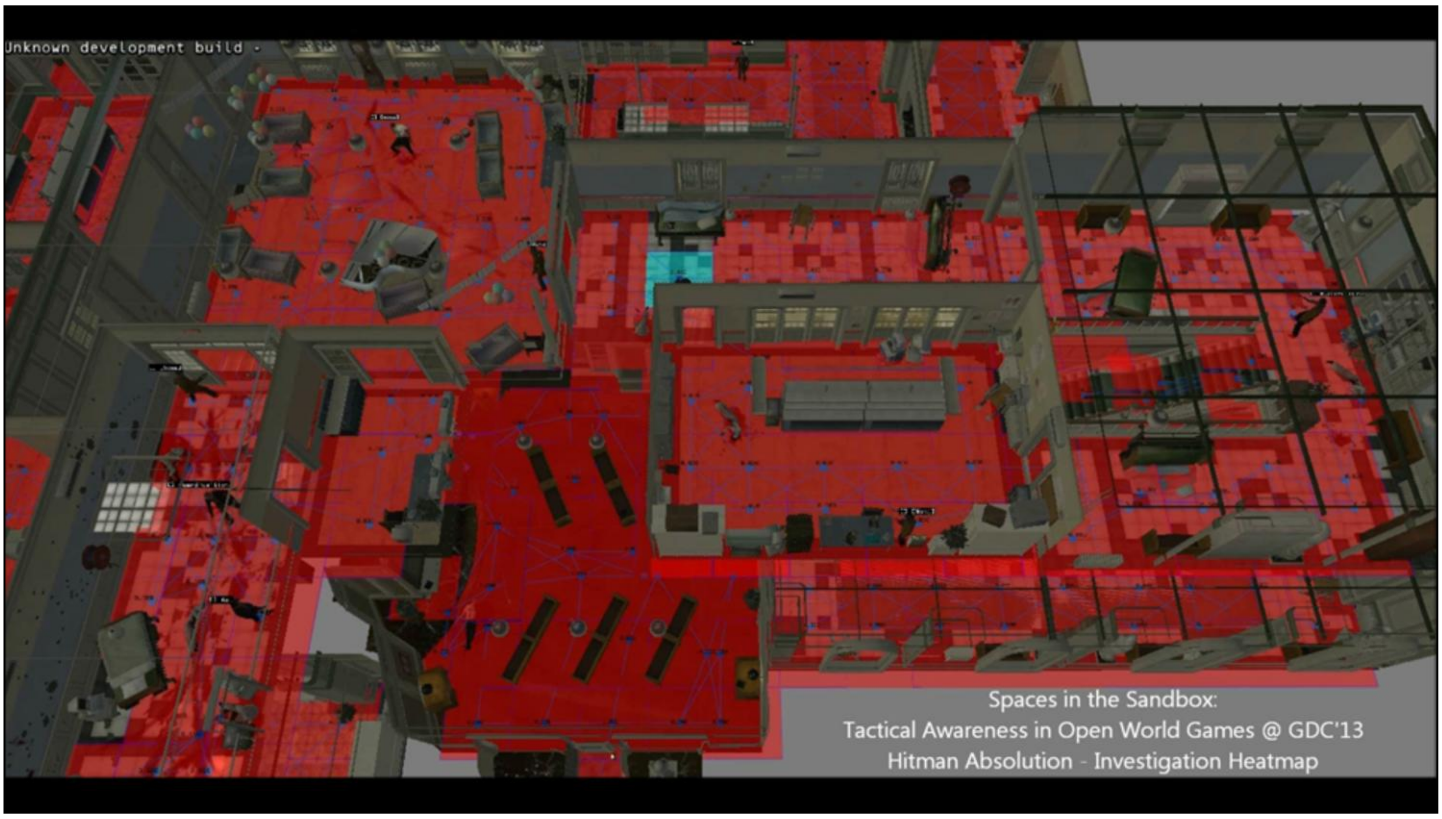


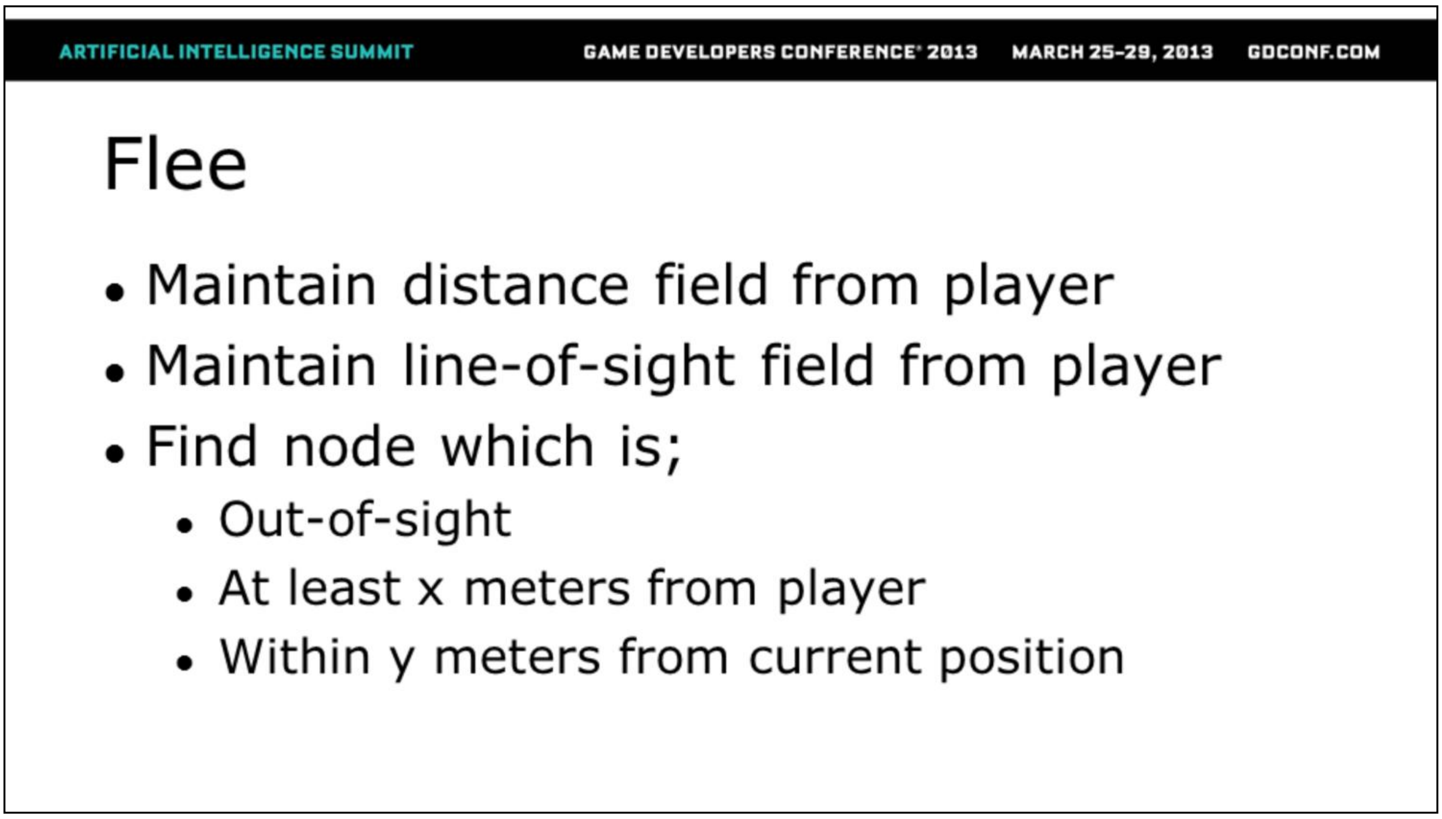
Composite field from NPC’s point-of-view. Threat further away in top right section of the image.



Investigation

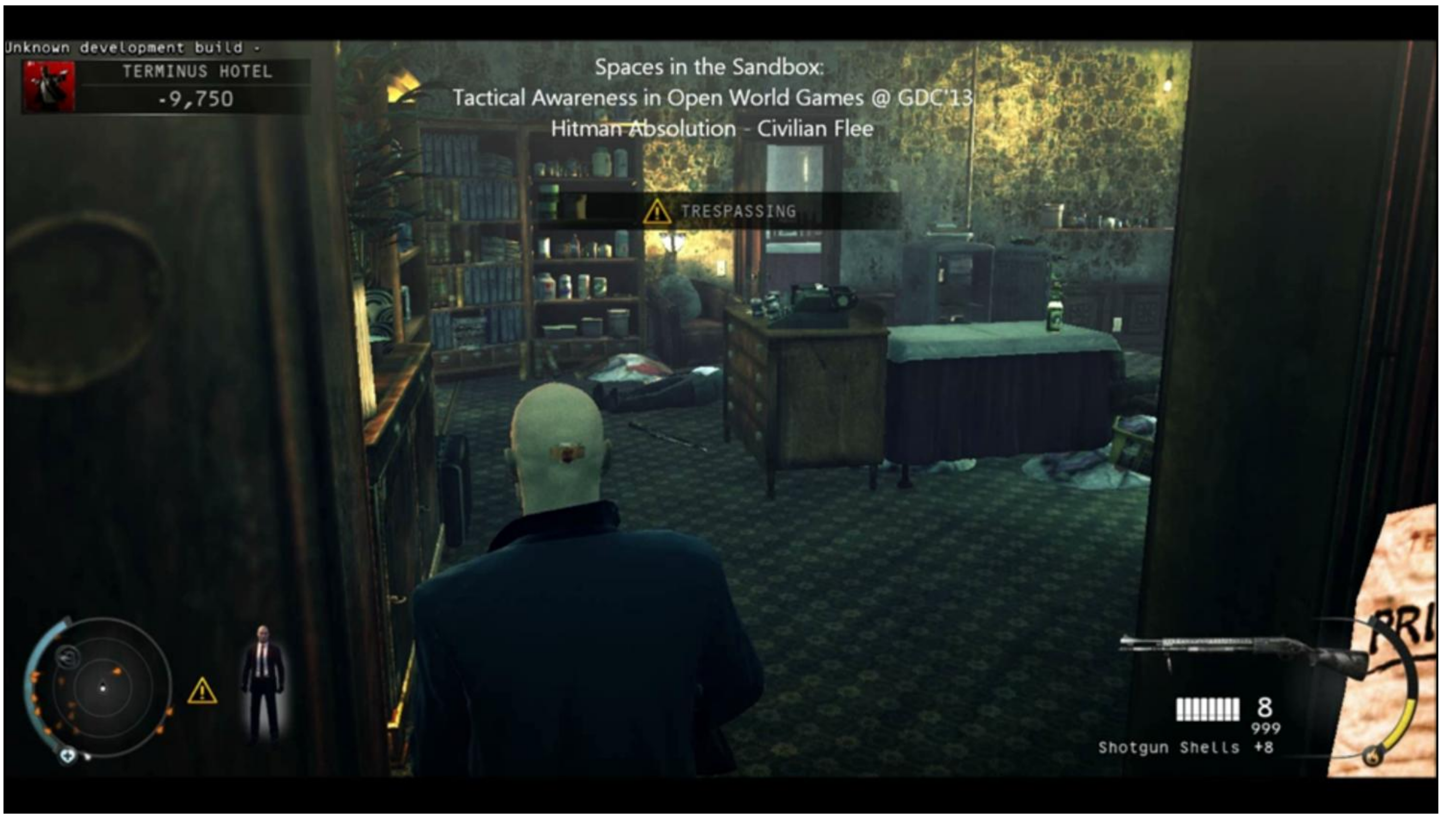
- Heatmap
  - Add heat to last known position of threat
  - Add heat when hear / see distractions
  - Cool over time
  - Cool faster when line-of-sight to node
- Pick a node with most heat within search radius

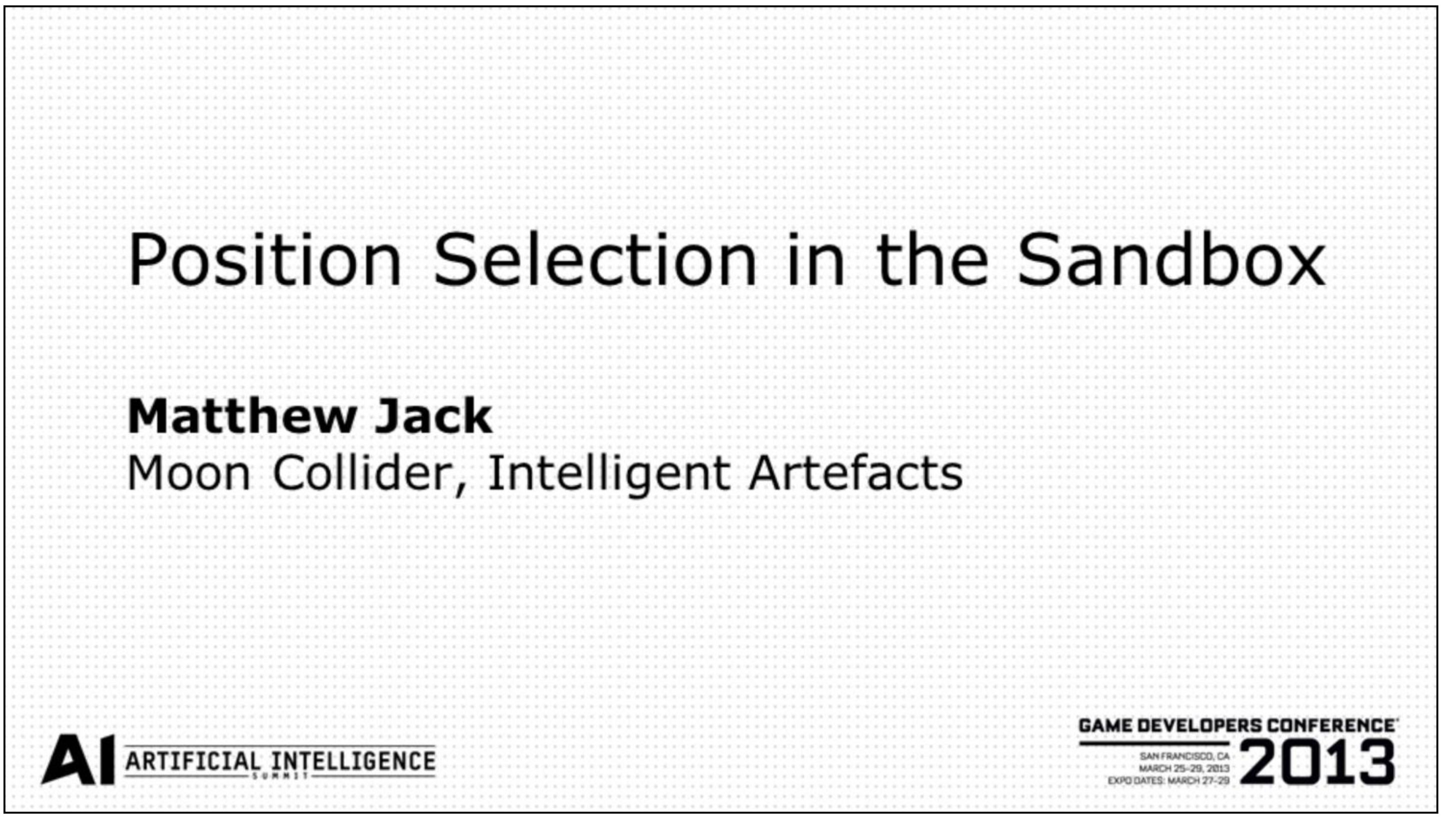

Unknown development build -
Spaces in the Sandbox:
Tactical Awareness in Open World Games @ GDC'13
Hitman Absolution - Investigation Heatmap



Flee

- Maintain distance field from player
- Maintain line-of-sight field from player
- Find node which is;
  - Out-of-sight
  - At least x meters from player
  - Within y meters from current position


Spaces in the Sandbox:
Tactical Awareness in Open World Games @ GDC'13
Hitman Absolution - Civilian Flee



Position Selection in the Sandbox

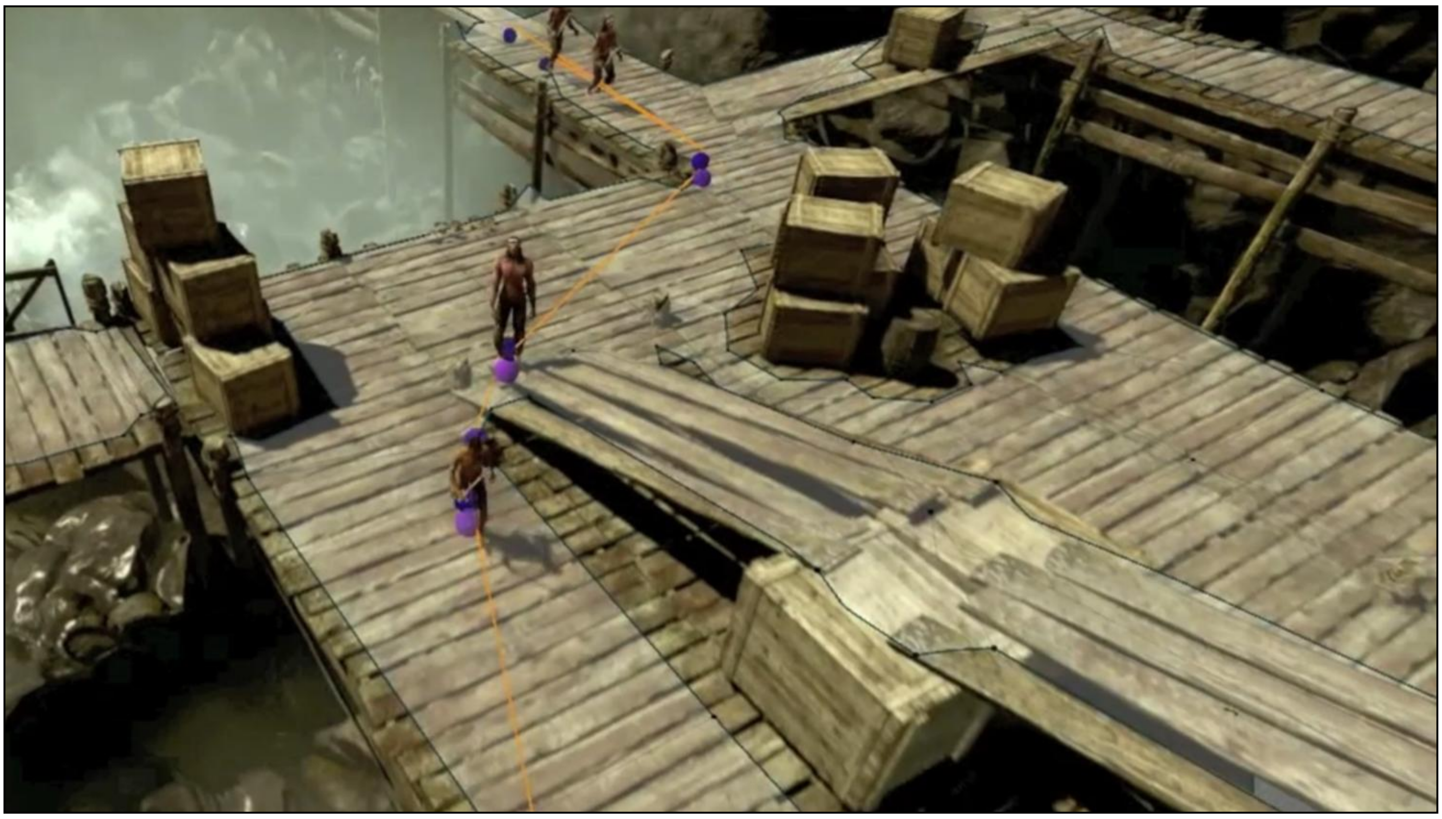
Matthew Jack
Moon Collider, Intelligent Artefacts

ARTIFICIAL INTELLIGENCE SUMMIT

GAME DEVELOPERS CONFERENCE
SAN FRANCISCO, CA
MARCH 25-29, 2013
EXPO DATES: MARCH 27-29
2013

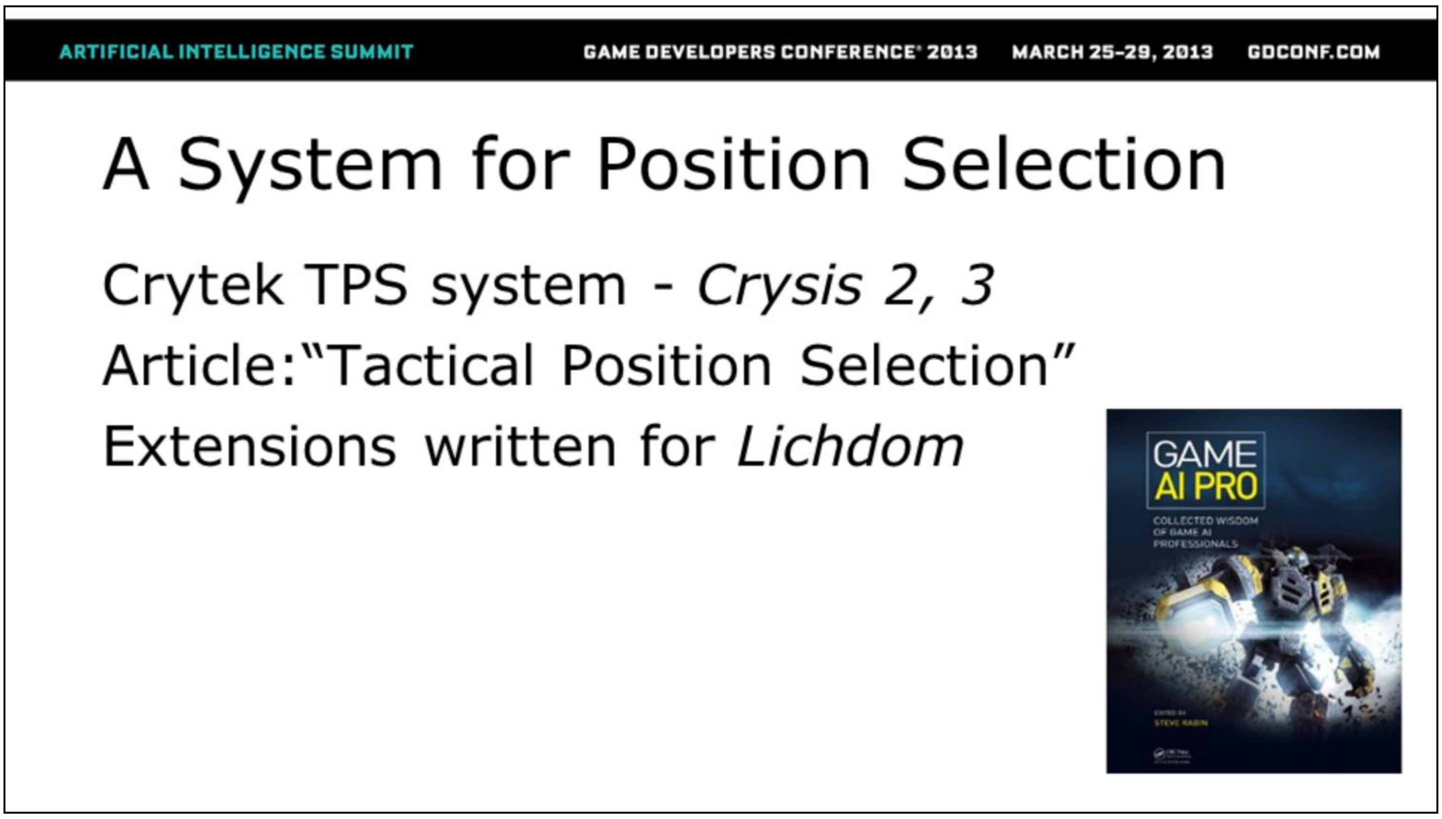
www.mooncollider.com
www.intelligentartefacts.com

Quick video clips of my experience



Dynamic Navigation under Kythera in Umbra
www.umbragame.com

XAVIANT
Pre-Alpha Code



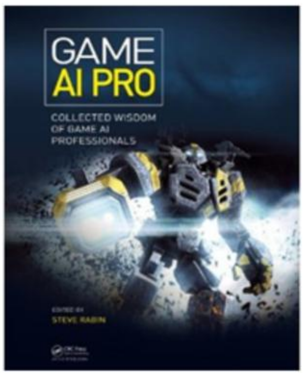
ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25–29, 2013 GDCONF.COM

A System for Position Selection

Crytek TPS system - *Crysis 2, 3*

Article:"Tactical Position Selection"

Extensions written for *Lichdom*

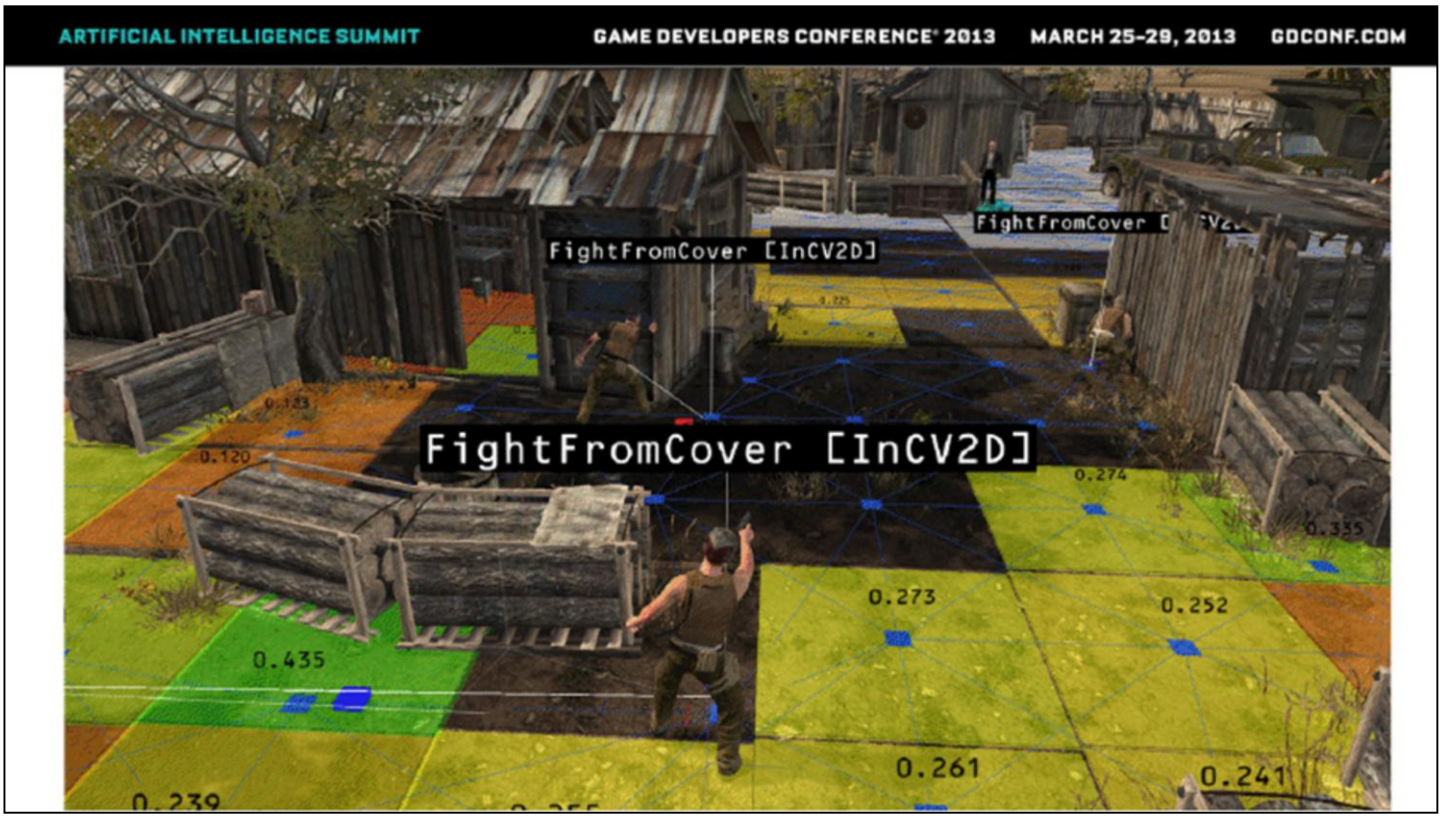


This talk is based on my work at Crytek on the Tactical Point Selection system.

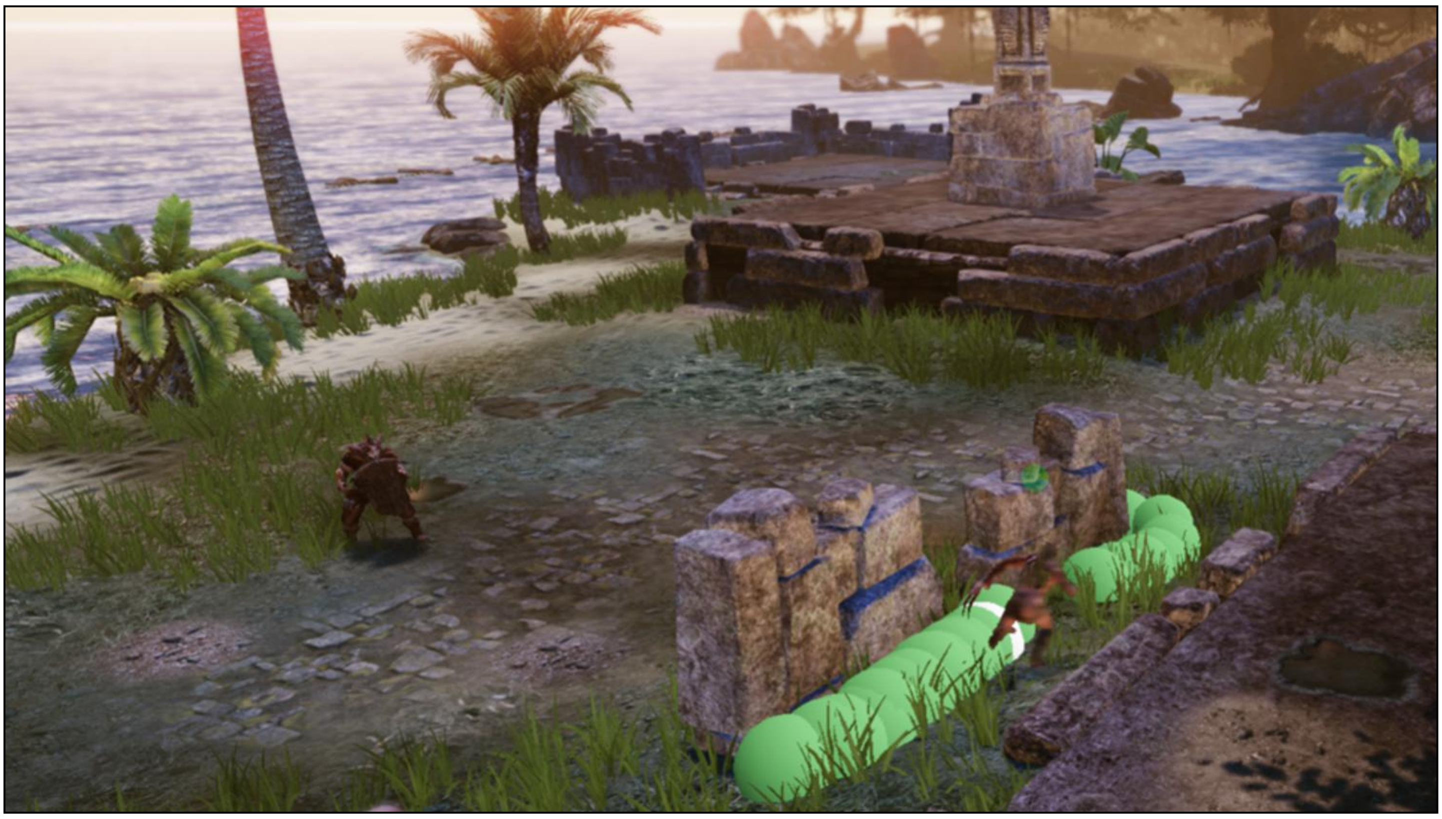
Much more details on many parts of this discussion in the book article. Due to be published April 2013.

www.Xaviant.com

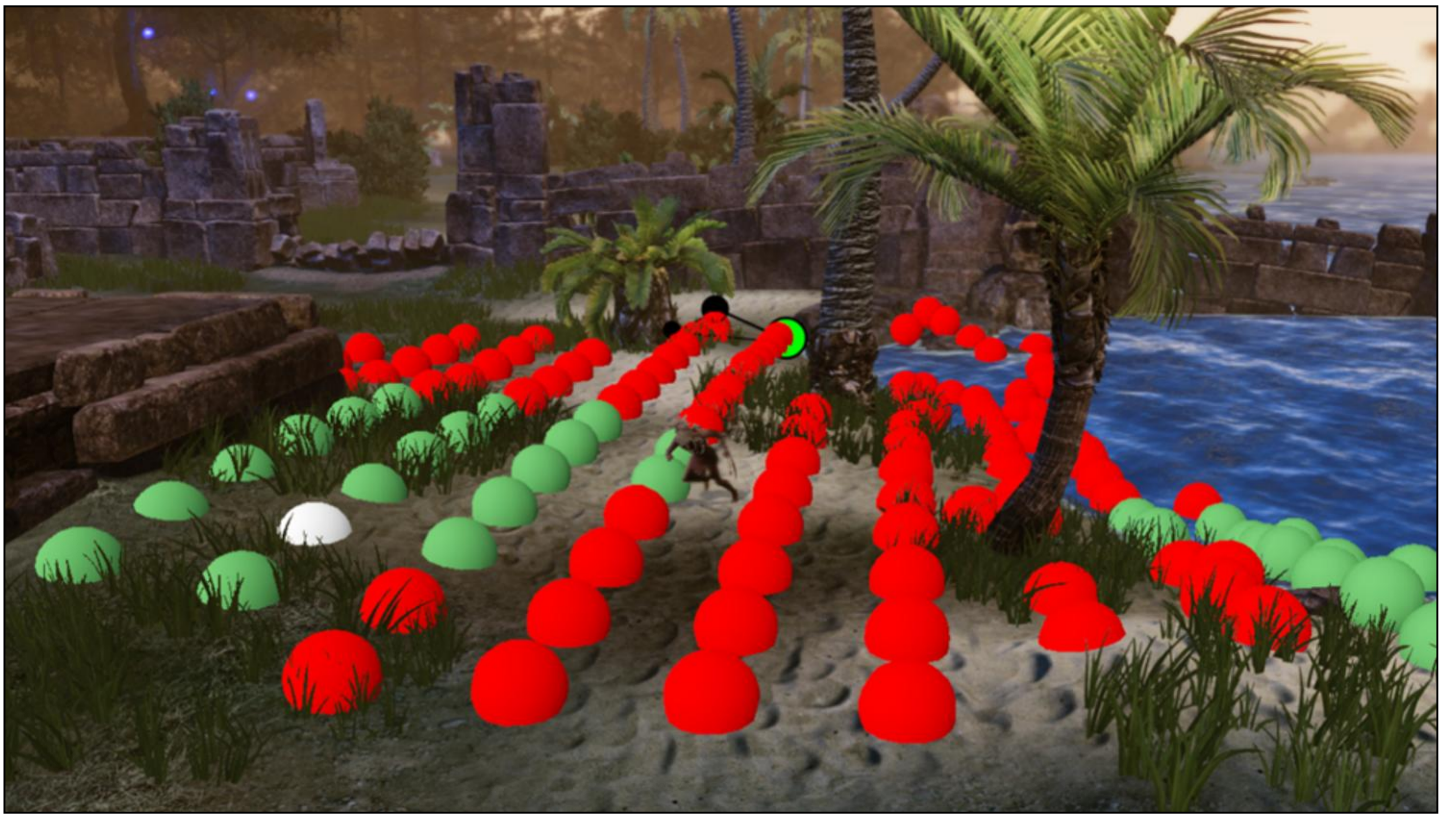
www.lichdom.com



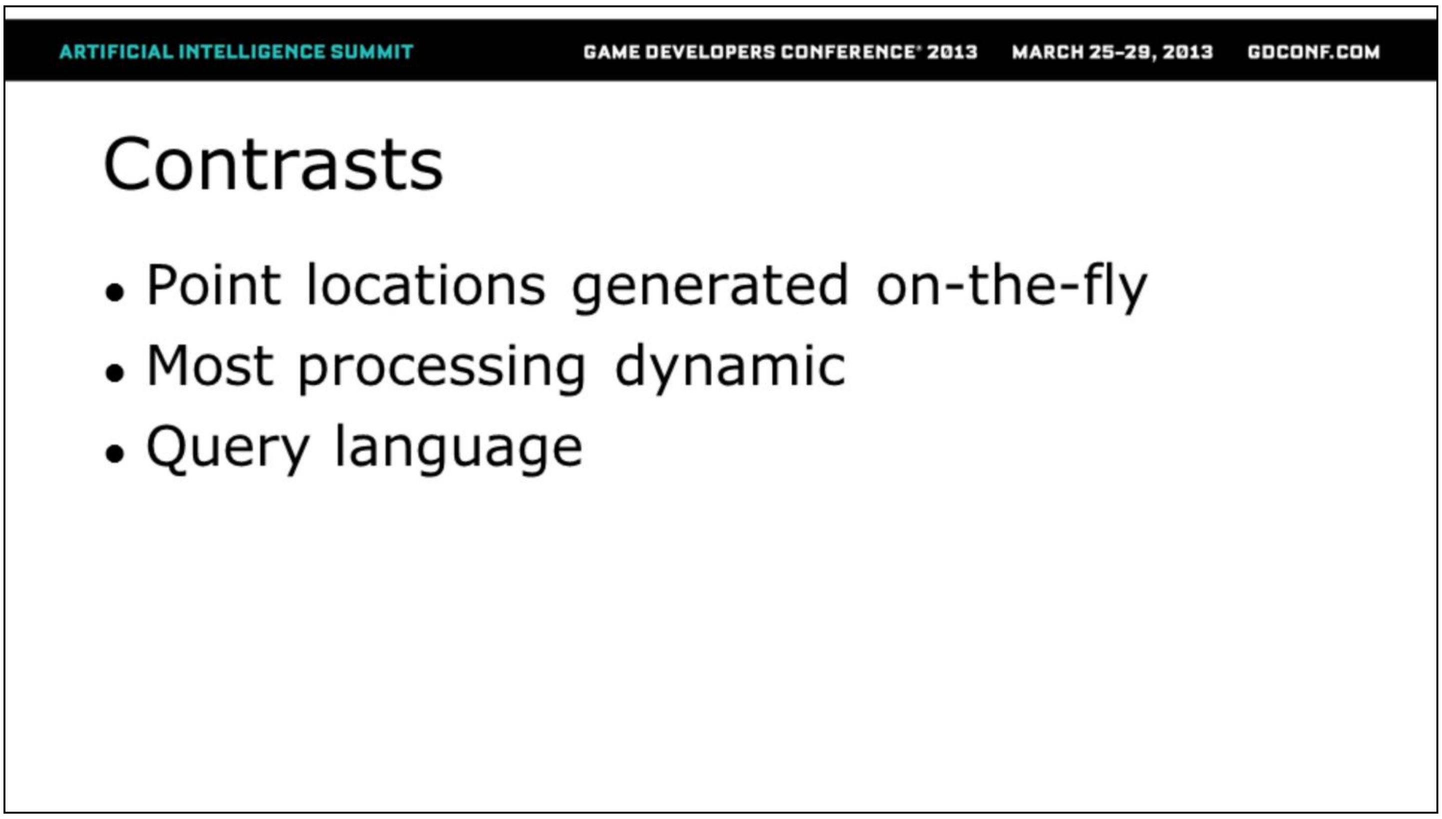
Reminder from Mika's slides. Note stages of generating a point in each time, filtering and then scoring the remainder.



Generating points around a cover object from a target, in Lichdom.



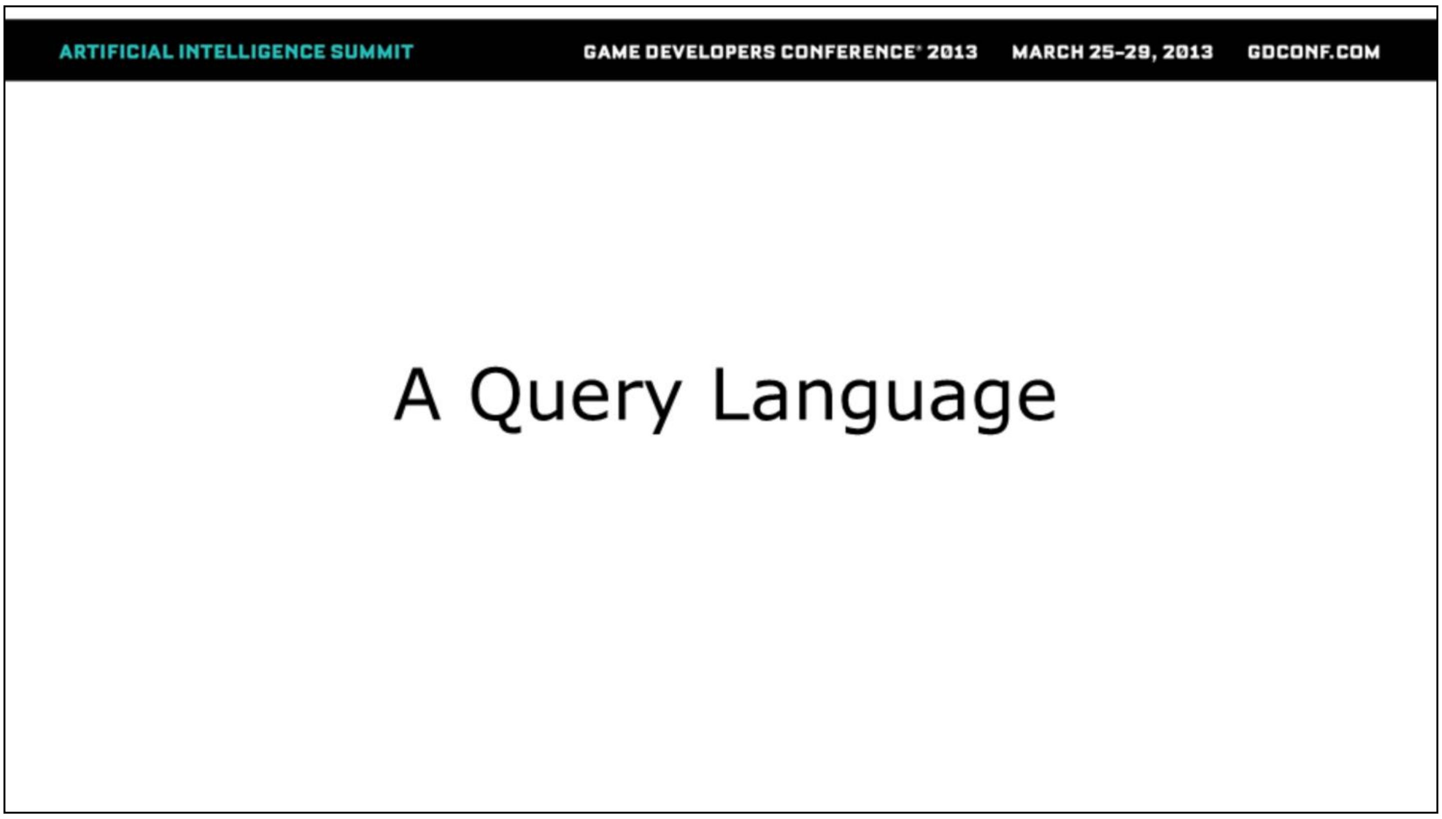
Generating points on open terrain


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25–29, 2013 GDCONF.COM

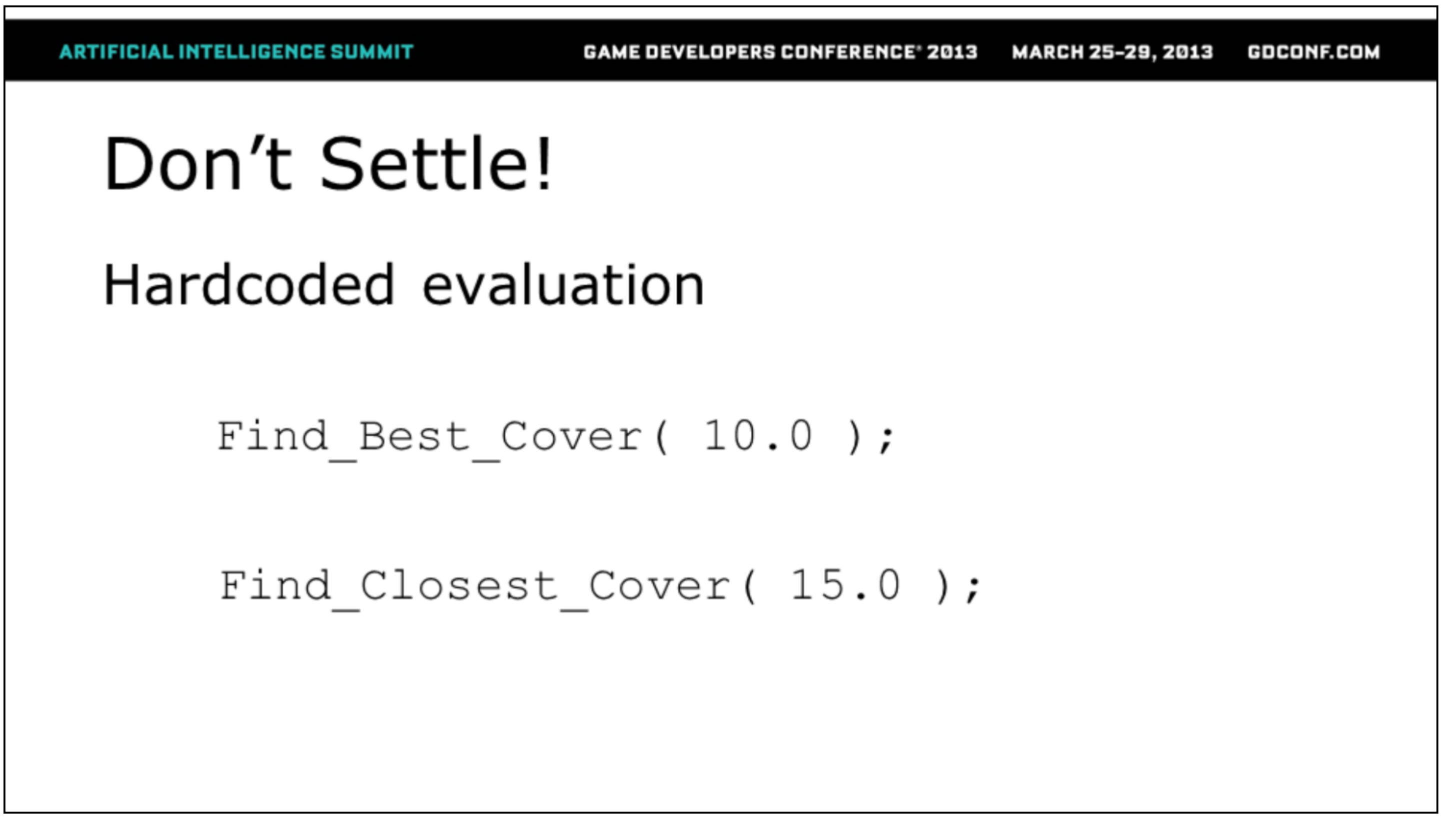
Contrasts

- Point locations generated on-the-fly
- Most processing dynamic
- Query language

What advantages?



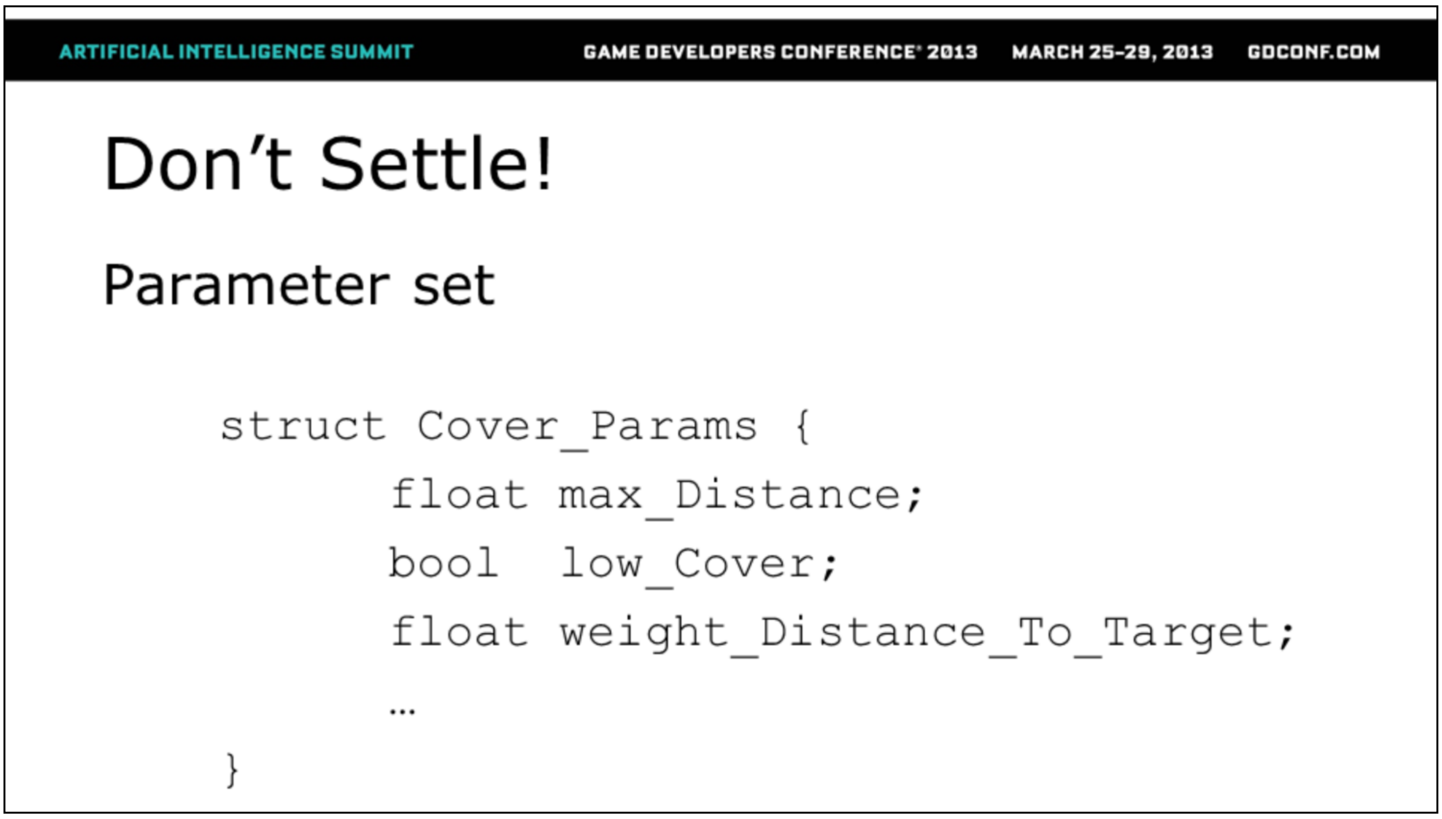
A Query Language



Don't Settle!

Hardcoded evaluation

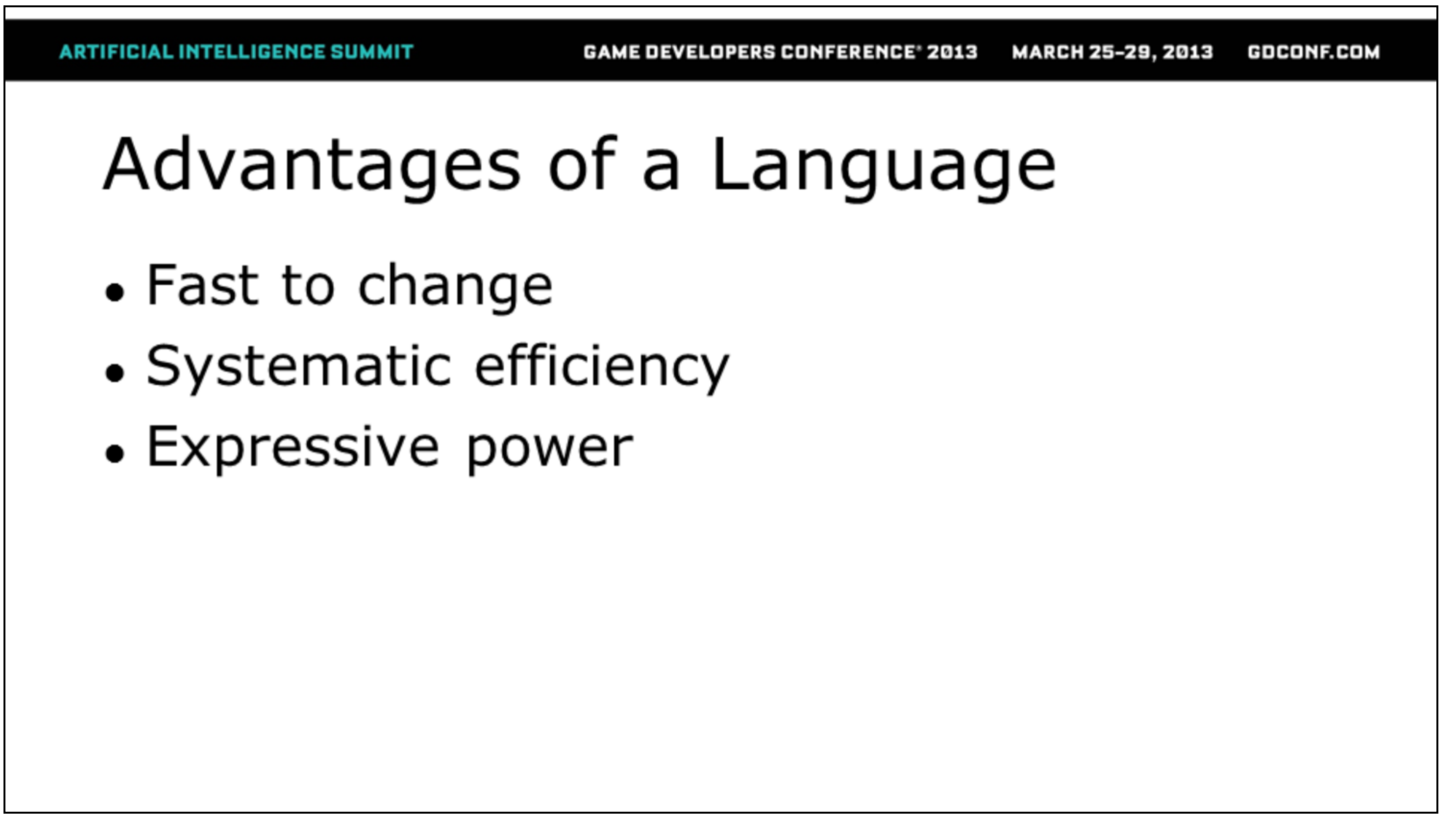
```
Find_Best_Cover( 10.0 );

Find_Closest_Cover( 15.0 );
```



Don’t Settle!

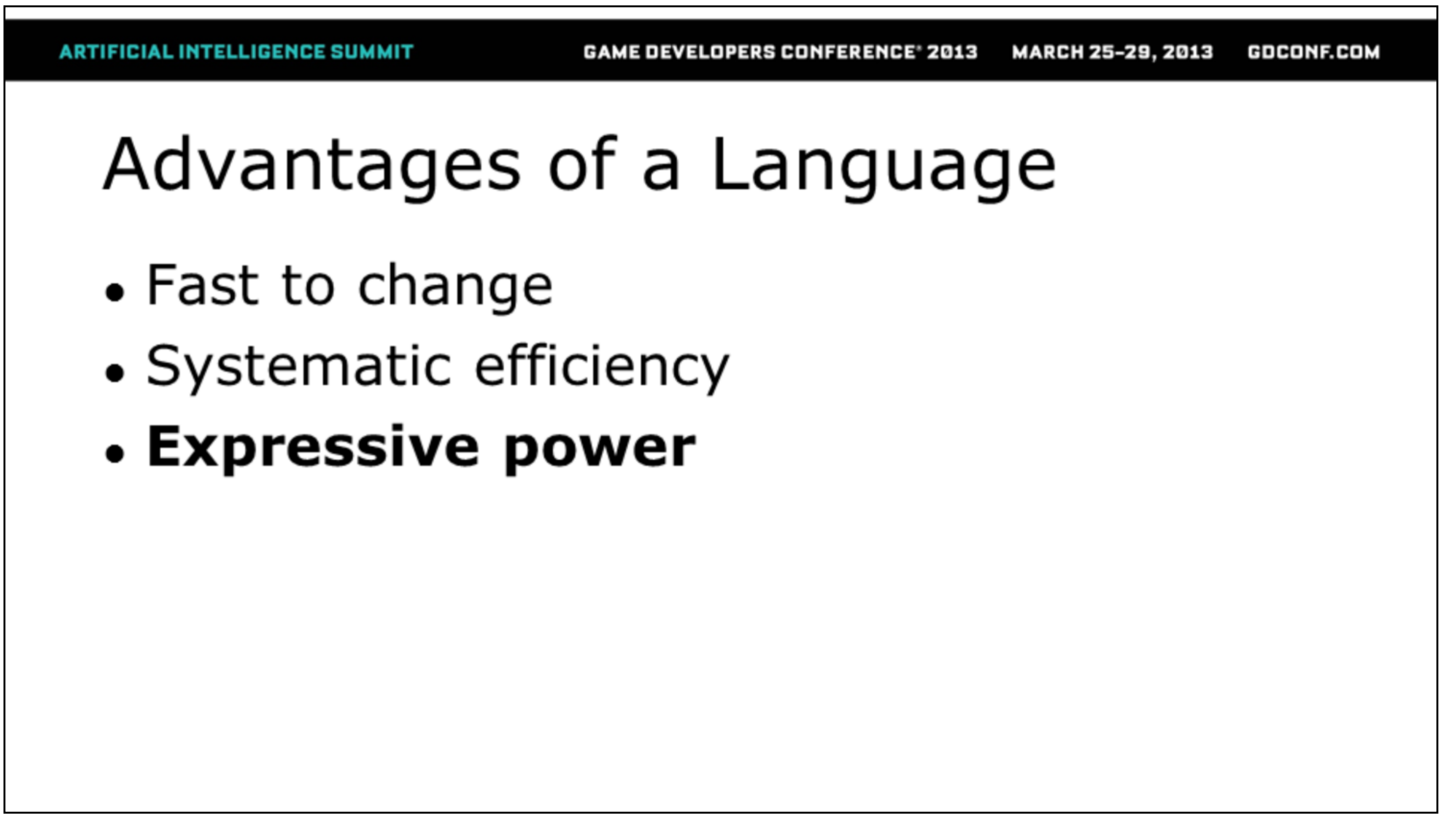
Parameter set

```
struct Cover_Params {
        float max_Distance;
        bool  low_Cover;
        float weight_Distance_To_Target;
        …
}
```



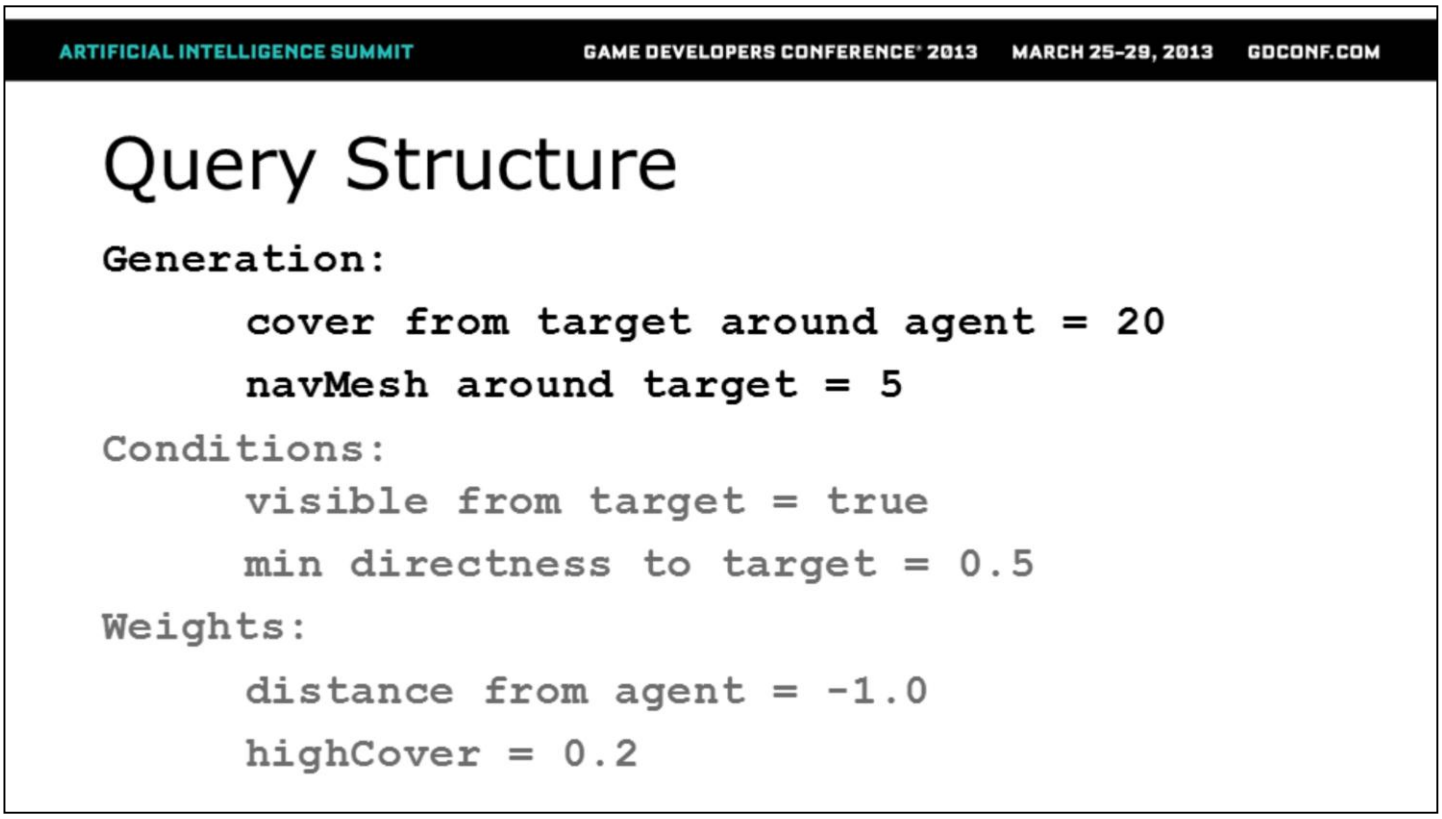
Advantages of a Language

- Fast to change
- Systematic efficiency
- Expressive power



Advantages of a Language

- Fast to change
- Systematic efficiency
- **Expressive power**


ARTIFICIAL INTELLIGENCE SUMMIT
GAME DEVELOPERS CONFERENCE® 2013
MARCH 25-29, 2013
GDCONF.COM

Query Structure

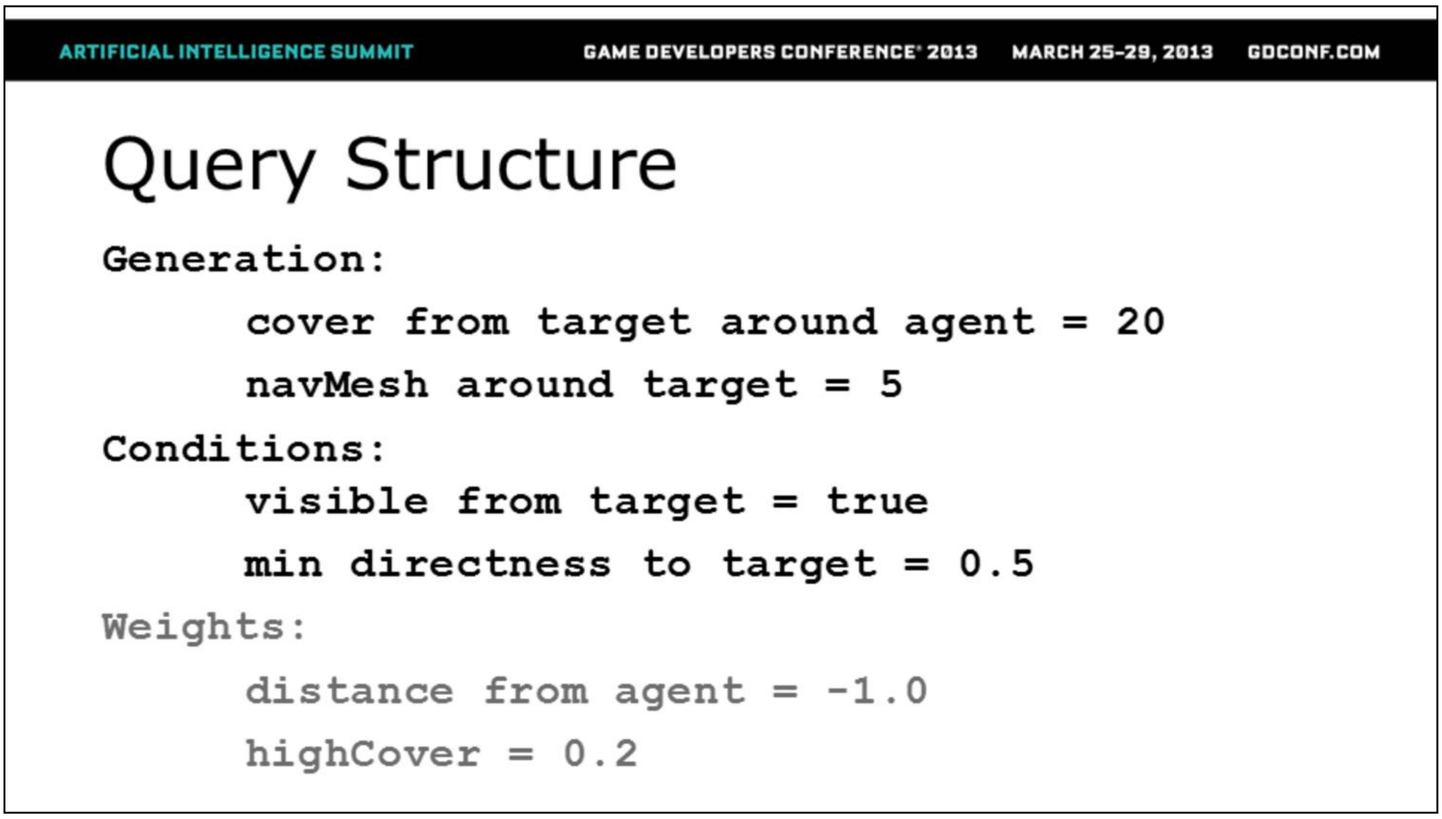
```
Generation:
     cover from target around agent = 20
     navMesh around target = 5
Conditions:
     visible from target = true
     min directness to target = 0.5
Weights:
     distance from agent = -1.0
     highCover = 0.2
```

Simplified query syntax from CryEngine TPS system

- What's the grammar of the language?

The grammar is described here - http://freesdk.crydev.net/display/SDKDOC4/Tactical+Point+System#TacticalPointSystem-QueryLanguageSyntax

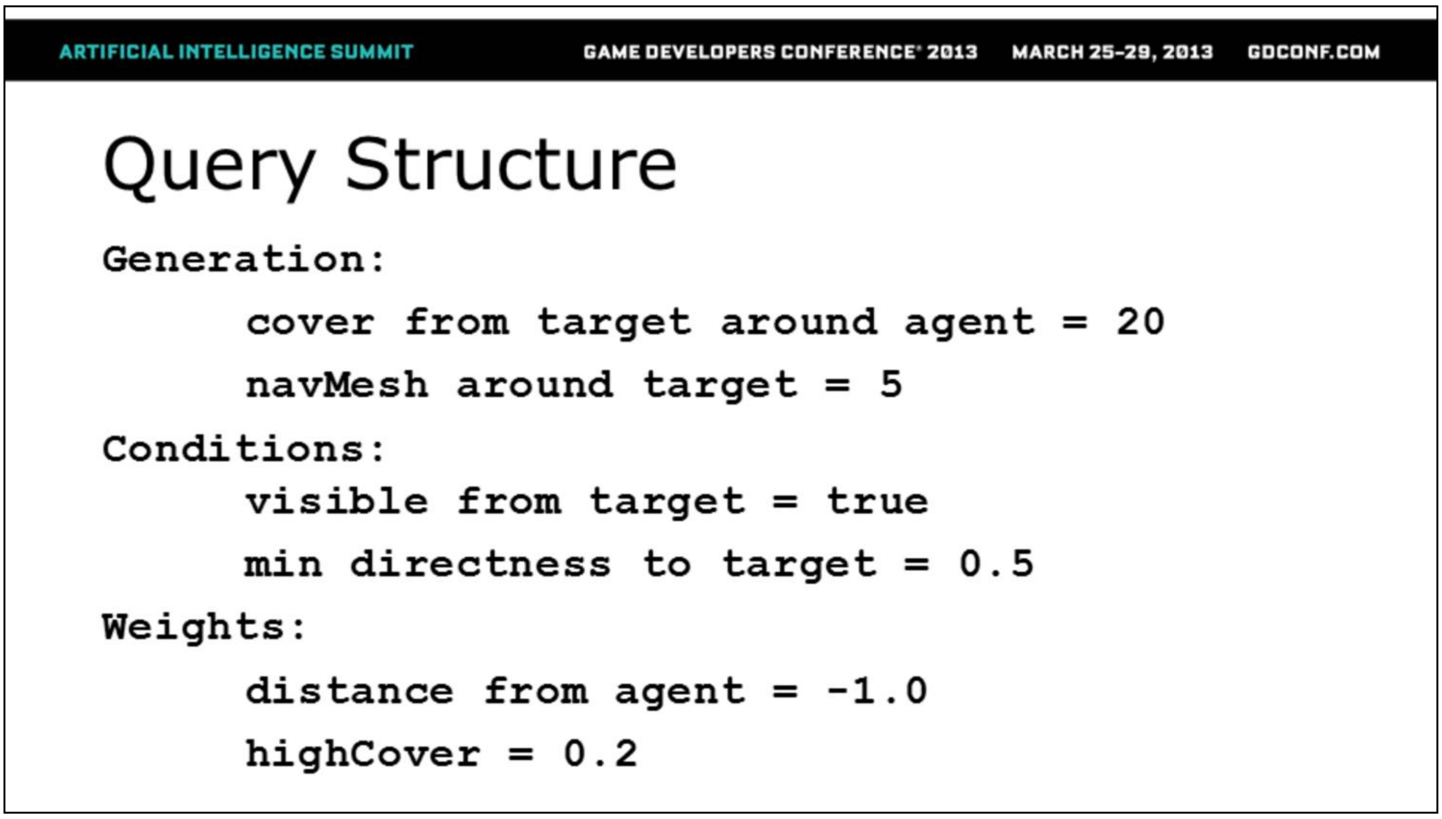
It now has far more keywords than that - this doc really just describes the first design.



ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

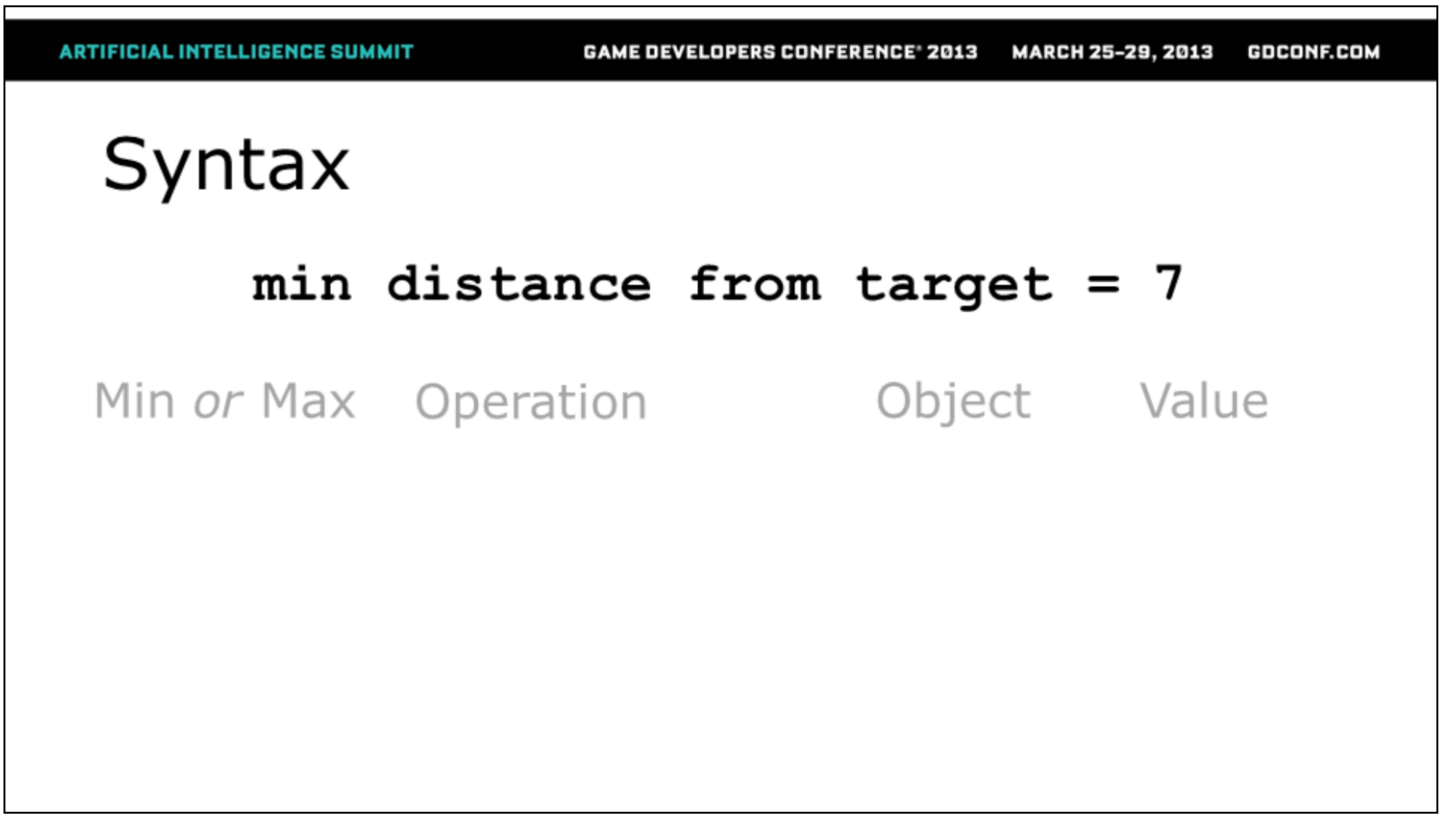
Query Structure

```
Generation:
      cover from target around agent = 20
      navMesh around target = 5
Conditions:
      visible from target = true
      min directness to target = 0.5
Weights:
      distance from agent = -1.0
      highCover = 0.2
```



Query Structure

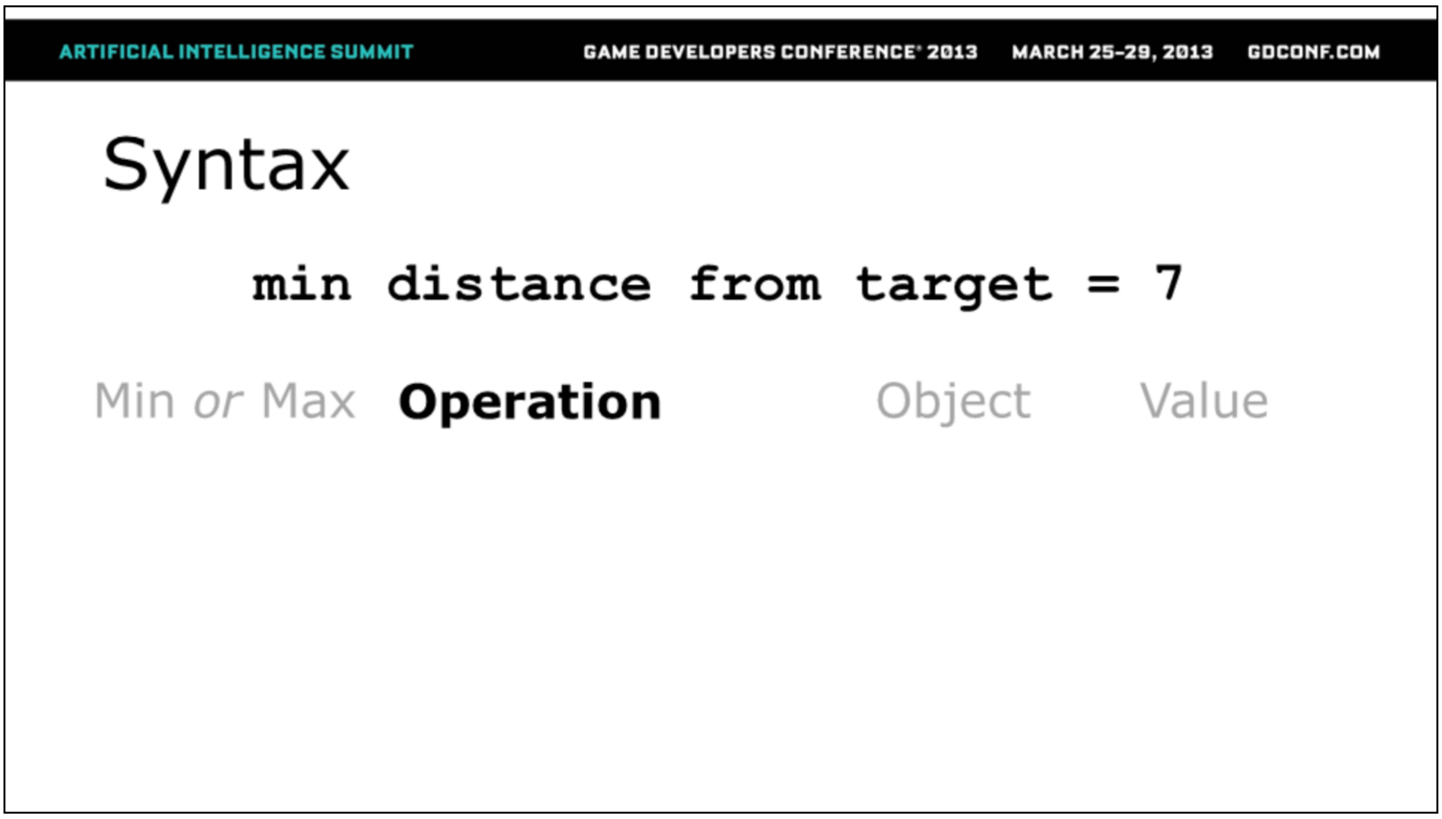
```
Generation:
      cover from target around agent = 20
      navMesh around target = 5
Conditions:
      visible from target = true
      min directness to target = 0.5
Weights:
      distance from agent = -1.0
      highCover = 0.2
```



Syntax

```
min distance from target = 7
```

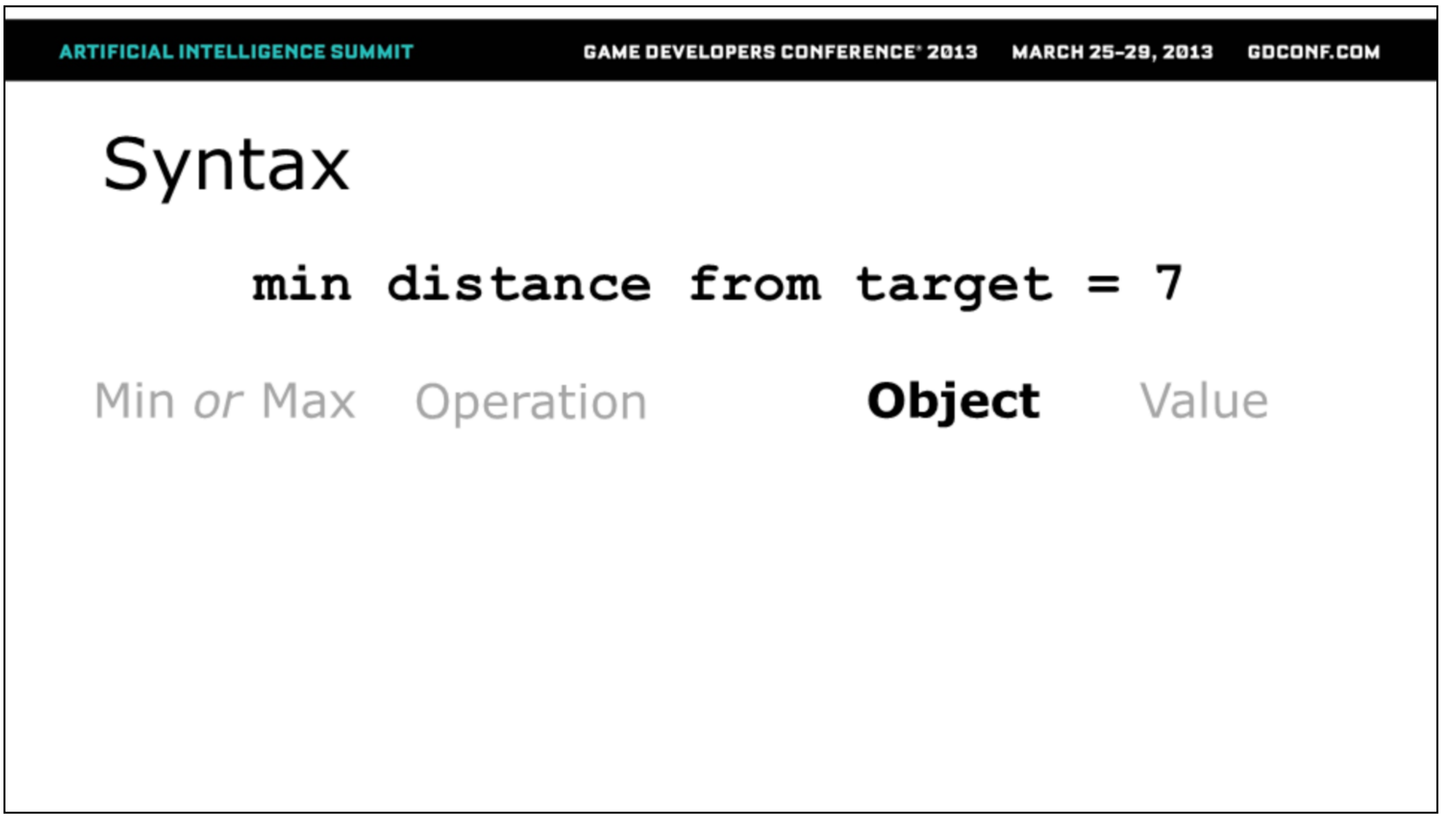
Min *or* Max Operation Object Value



Syntax

```
min distance from target = 7
```

Min *or* Max **Operation** Object Value



ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Syntax

```
min distance from target = 7
```

Min *or* Max Operation **Object** Value

Syntax

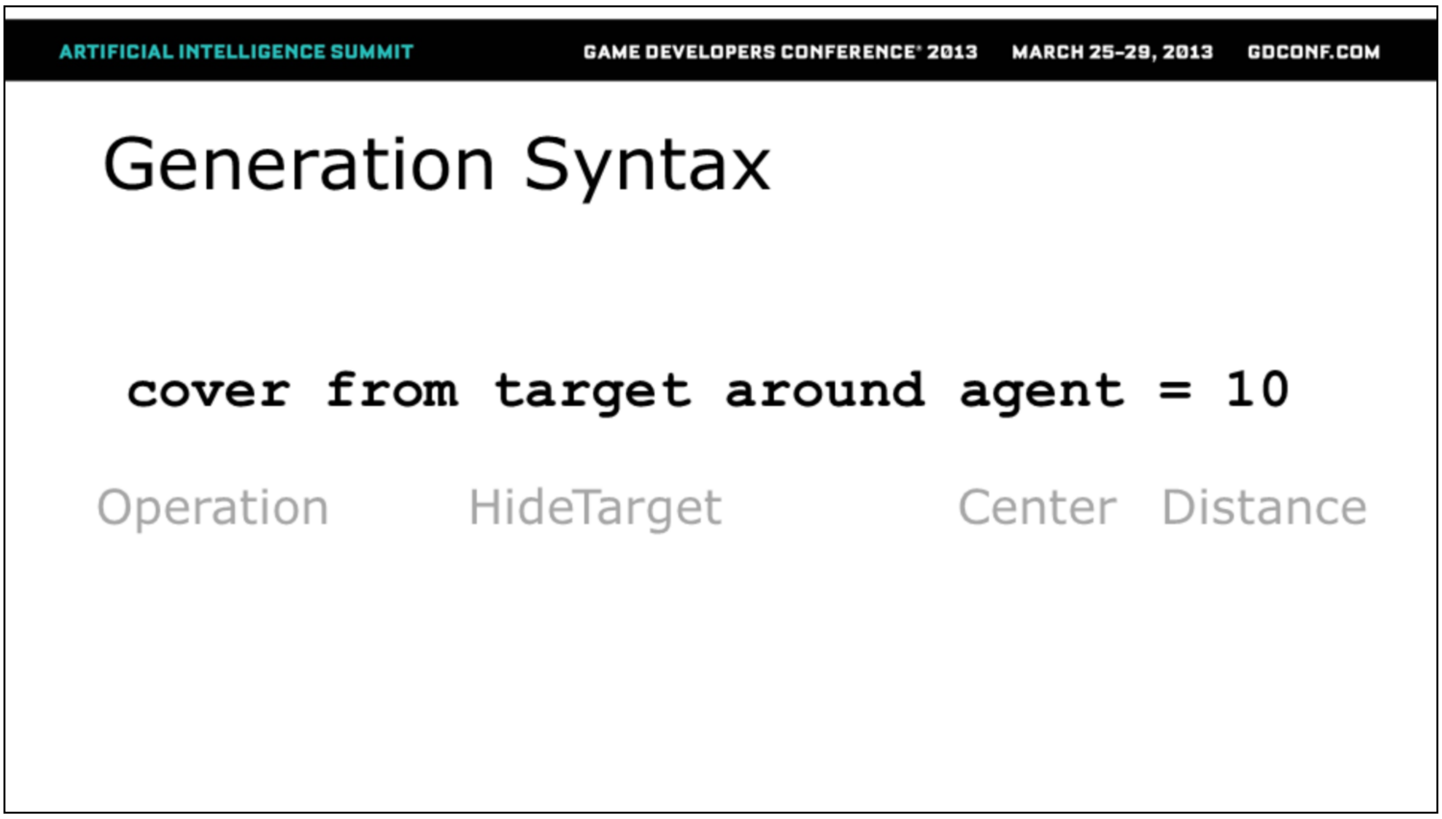
```
min distance from target = 7
```

Min *or* Max Operation Object Value

Syntax

```
min distance from target = 7
```

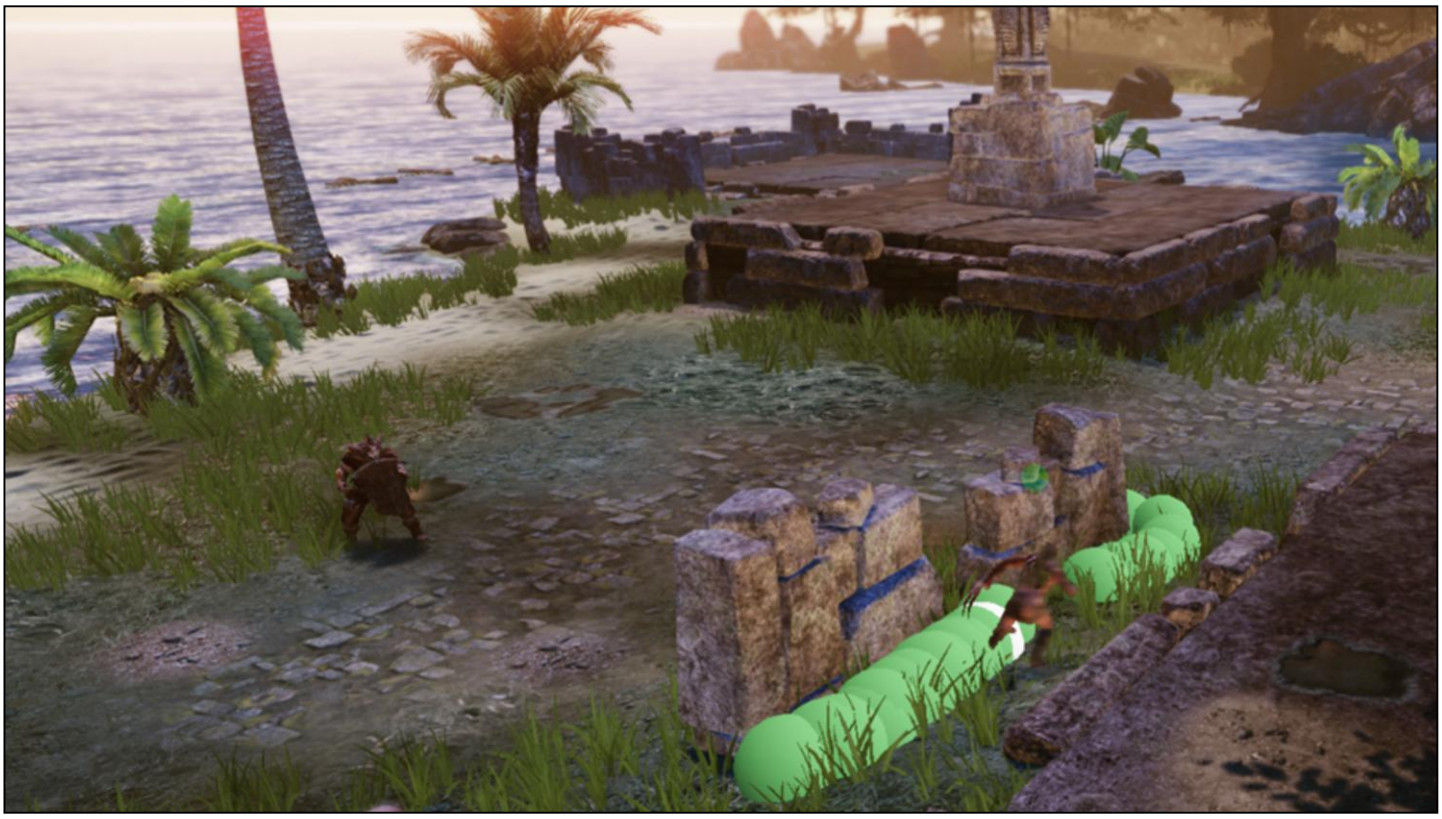
Min *or* Max Operation Object **Value**

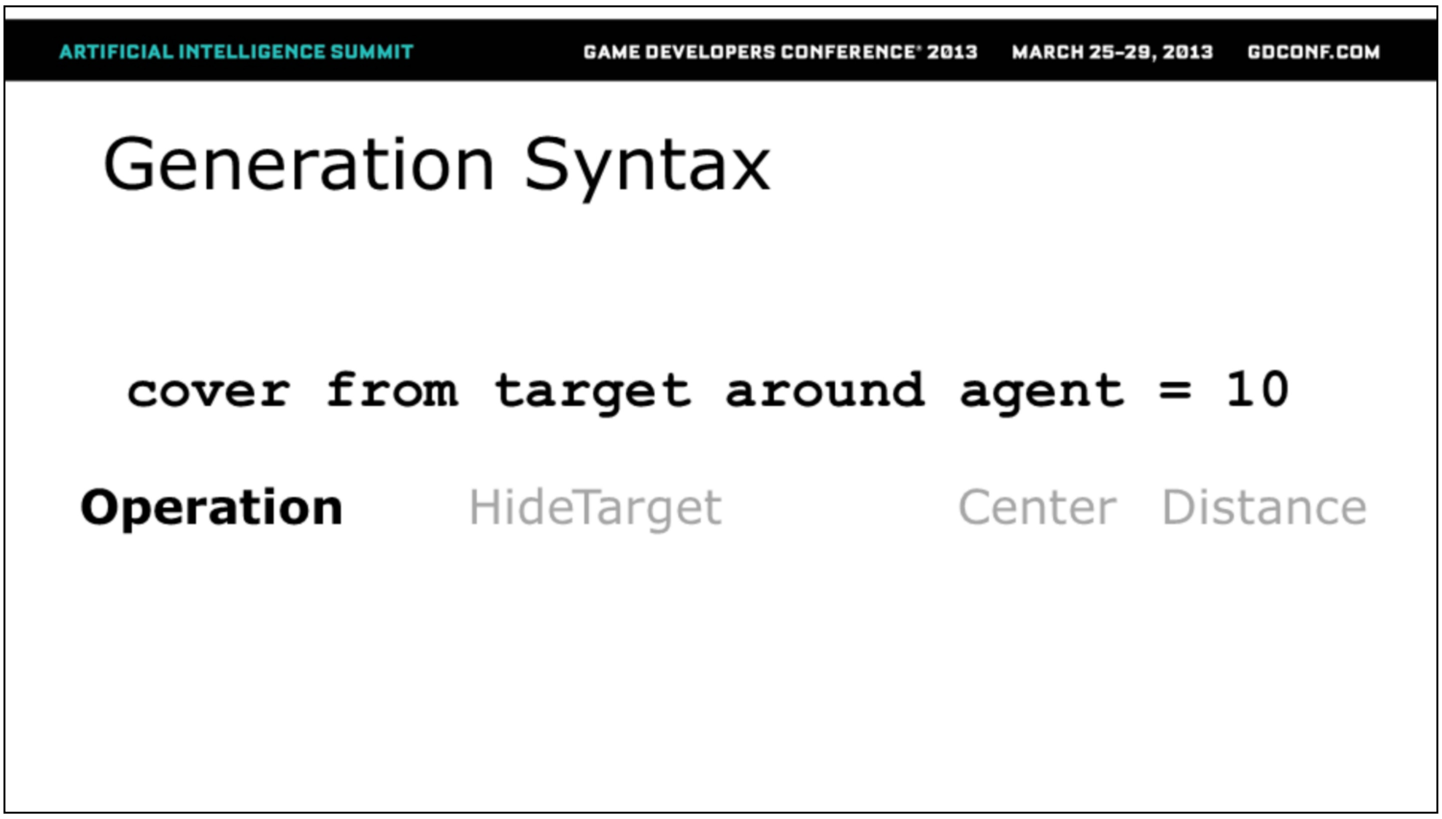


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Generation Syntax

```
cover     from target around agent  = 10
Operation      HideTarget      Center  Distance
```

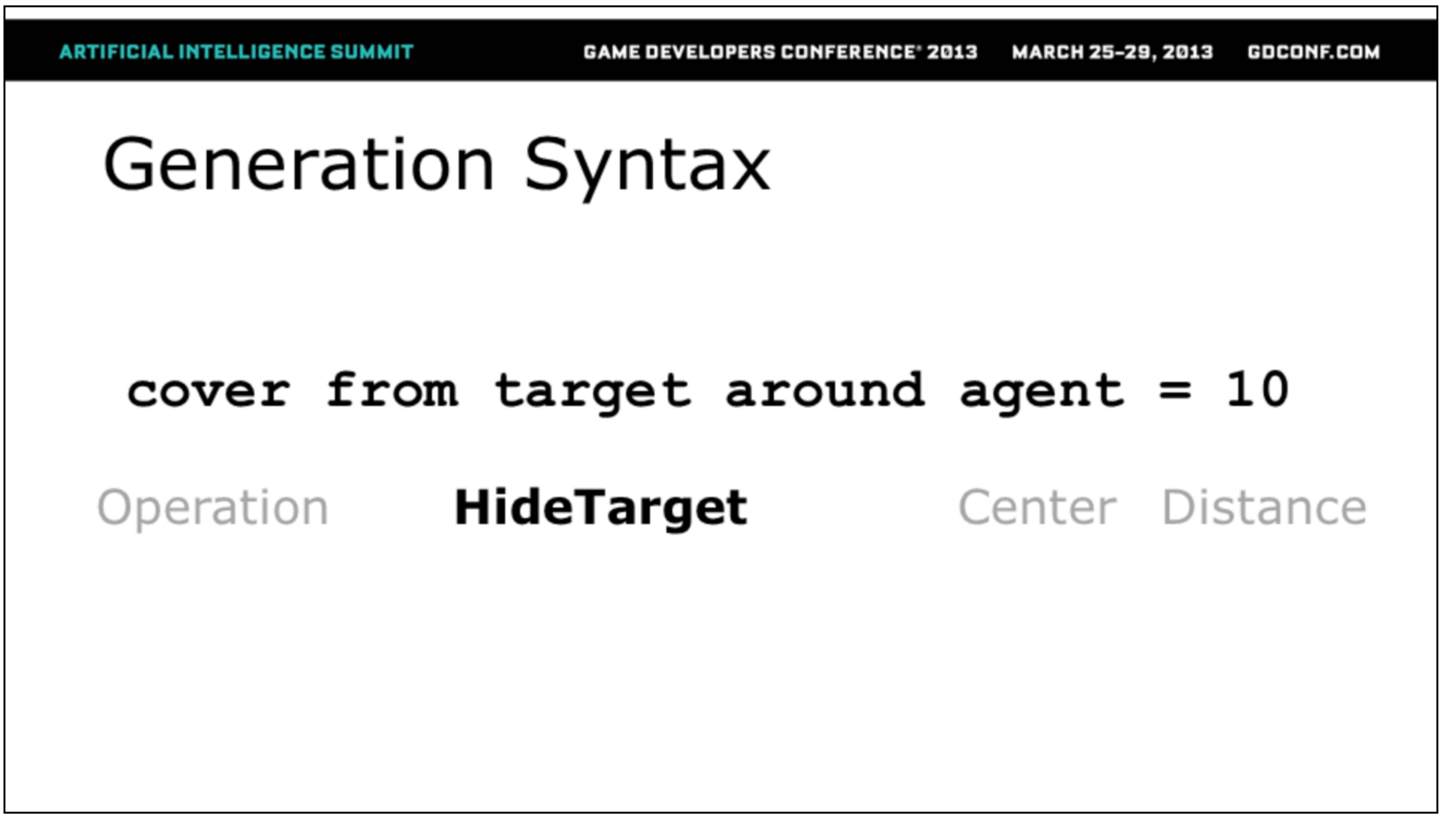




Generation Syntax

```
cover from target around agent = 10
```

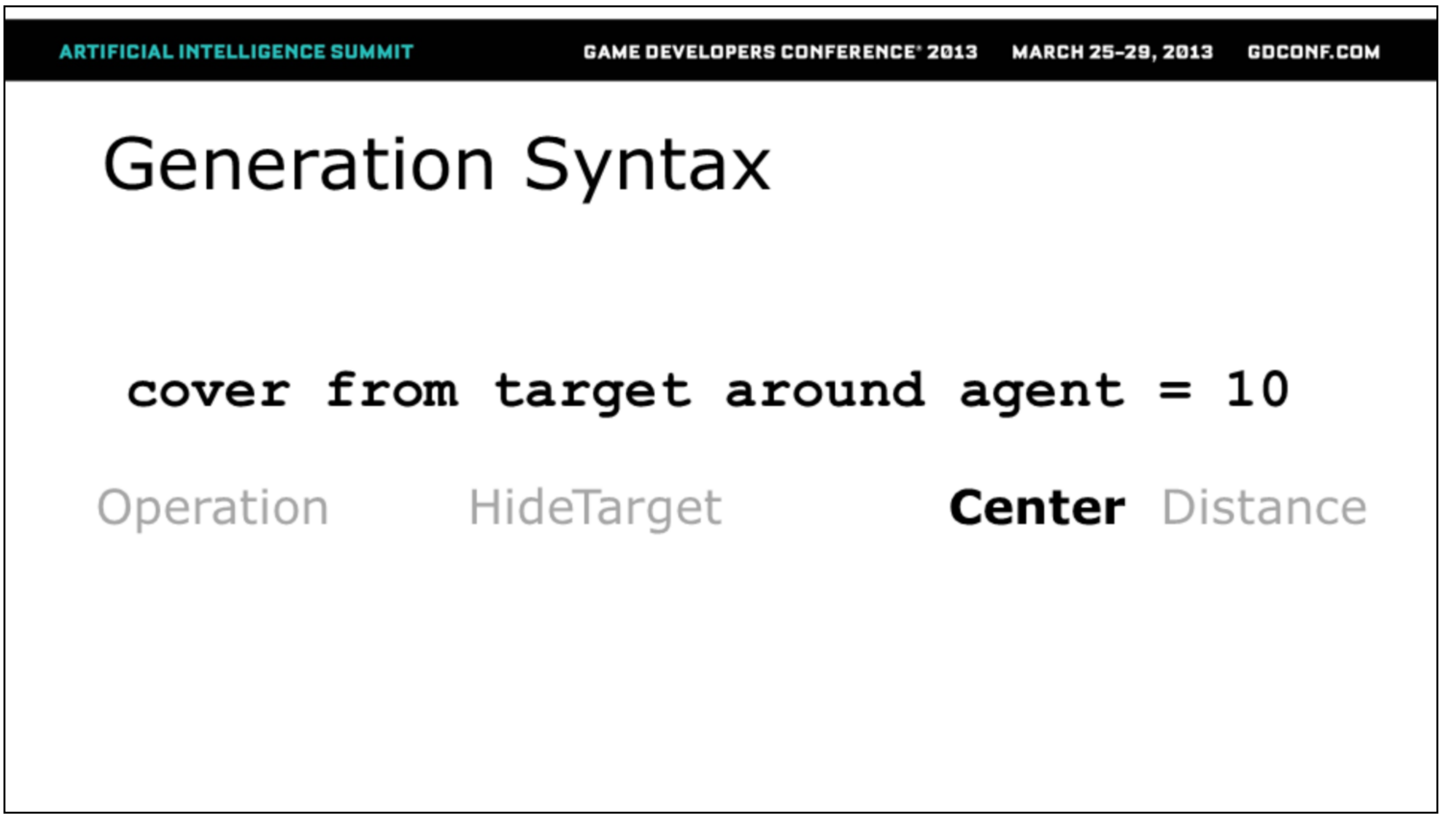
Operation HideTarget Center Distance



Generation Syntax

```
cover from target around agent = 10
```

Operation **HideTarget** Center Distance

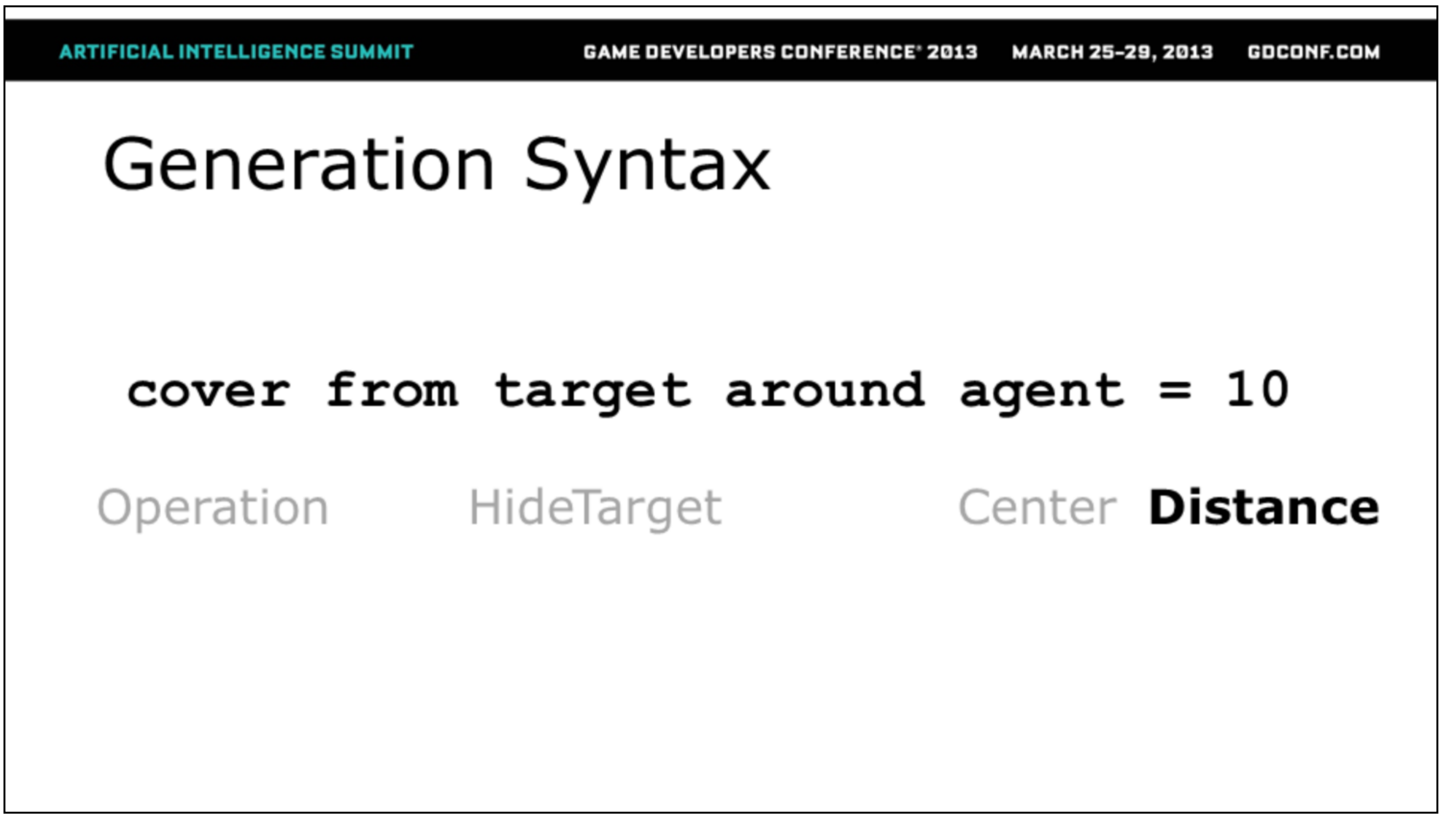


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Generation Syntax

```
cover from target around agent = 10
```

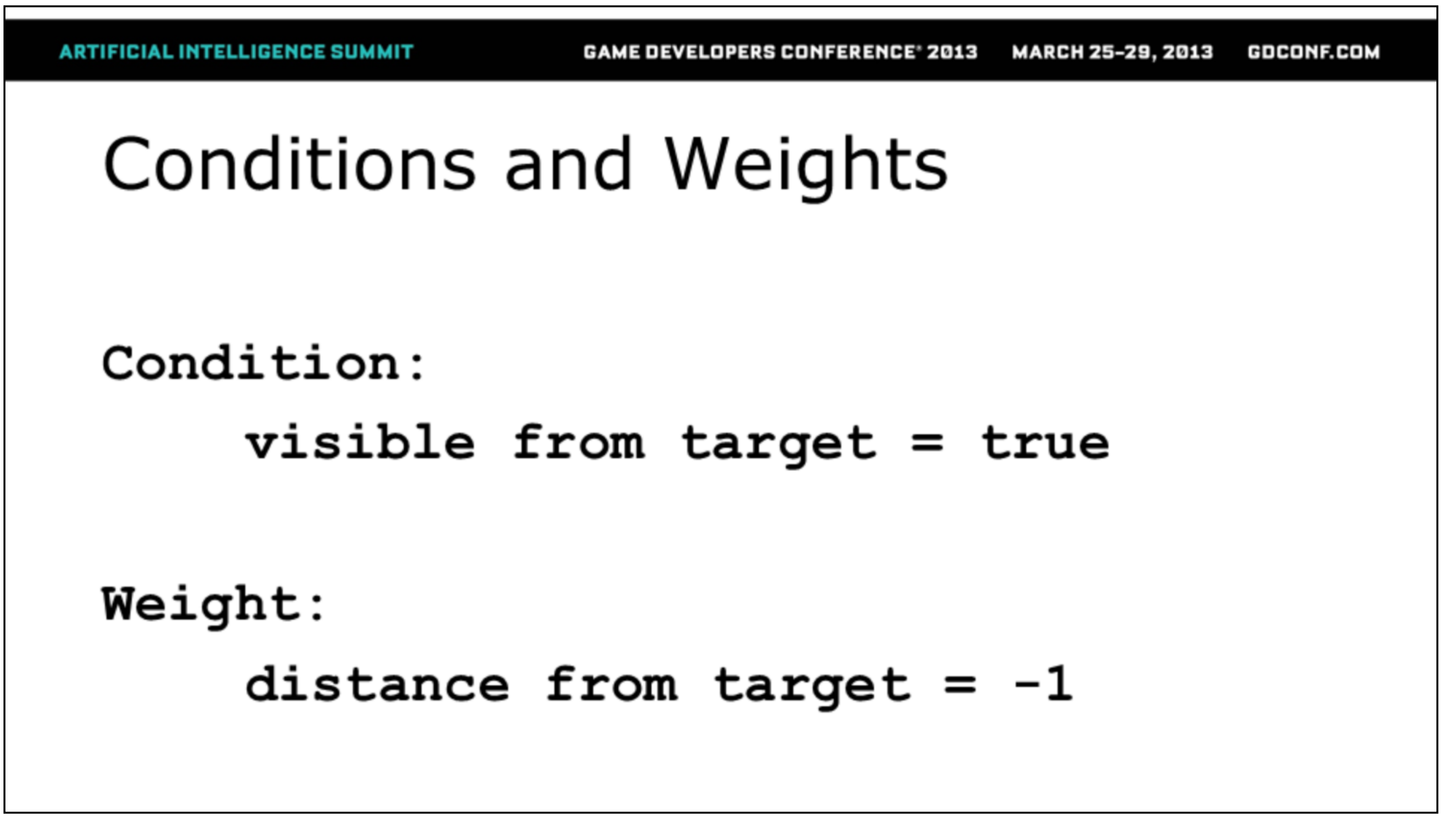
Operation HideTarget **Center** Distance



Generation Syntax

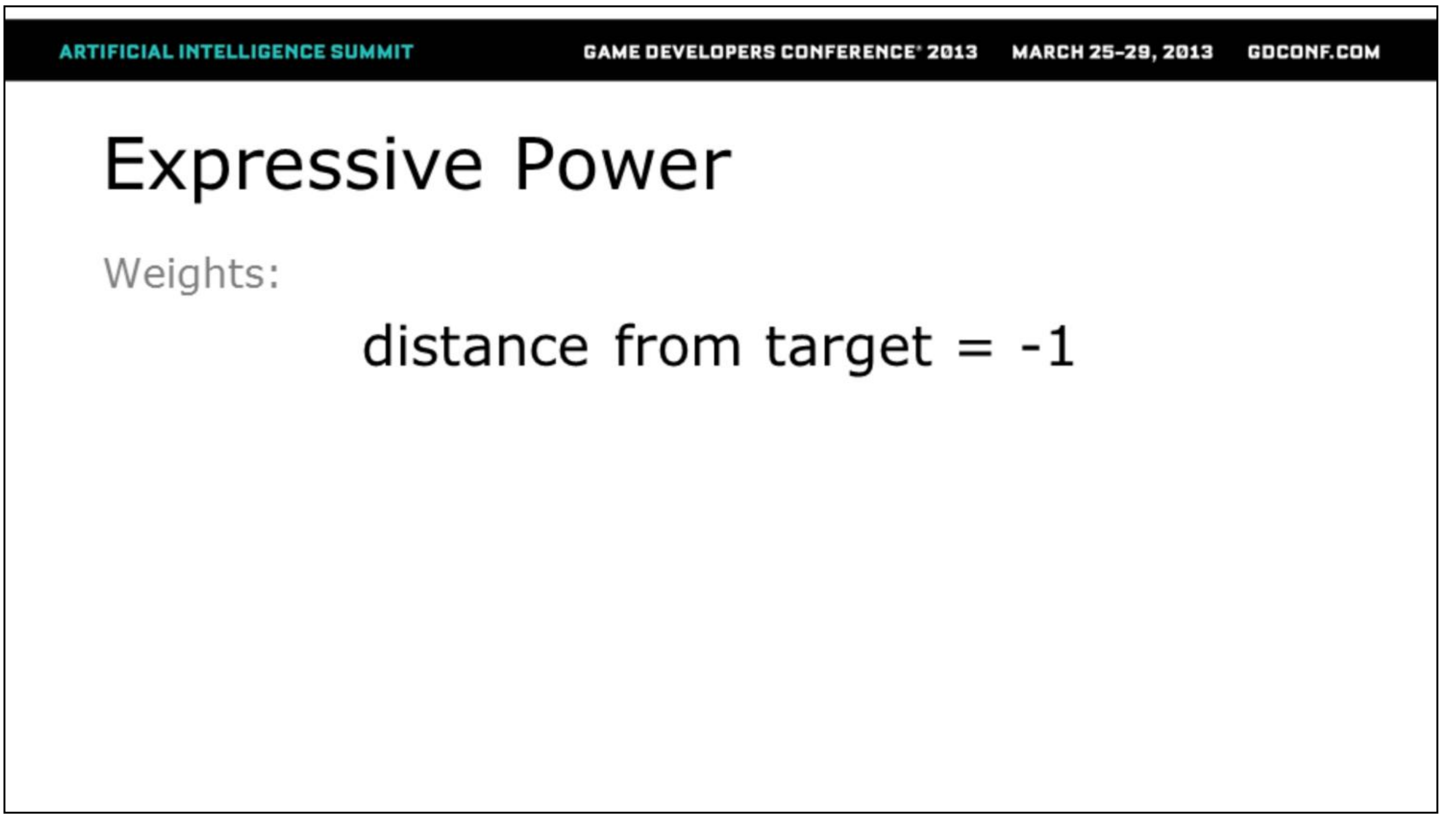
```
cover from target around agent = 10
```

Operation HideTarget Center **Distance**



Conditions and Weights

```
Condition:
    visible from target = true

Weight:
    distance from target = -1
```



Expressive Power

Weights:

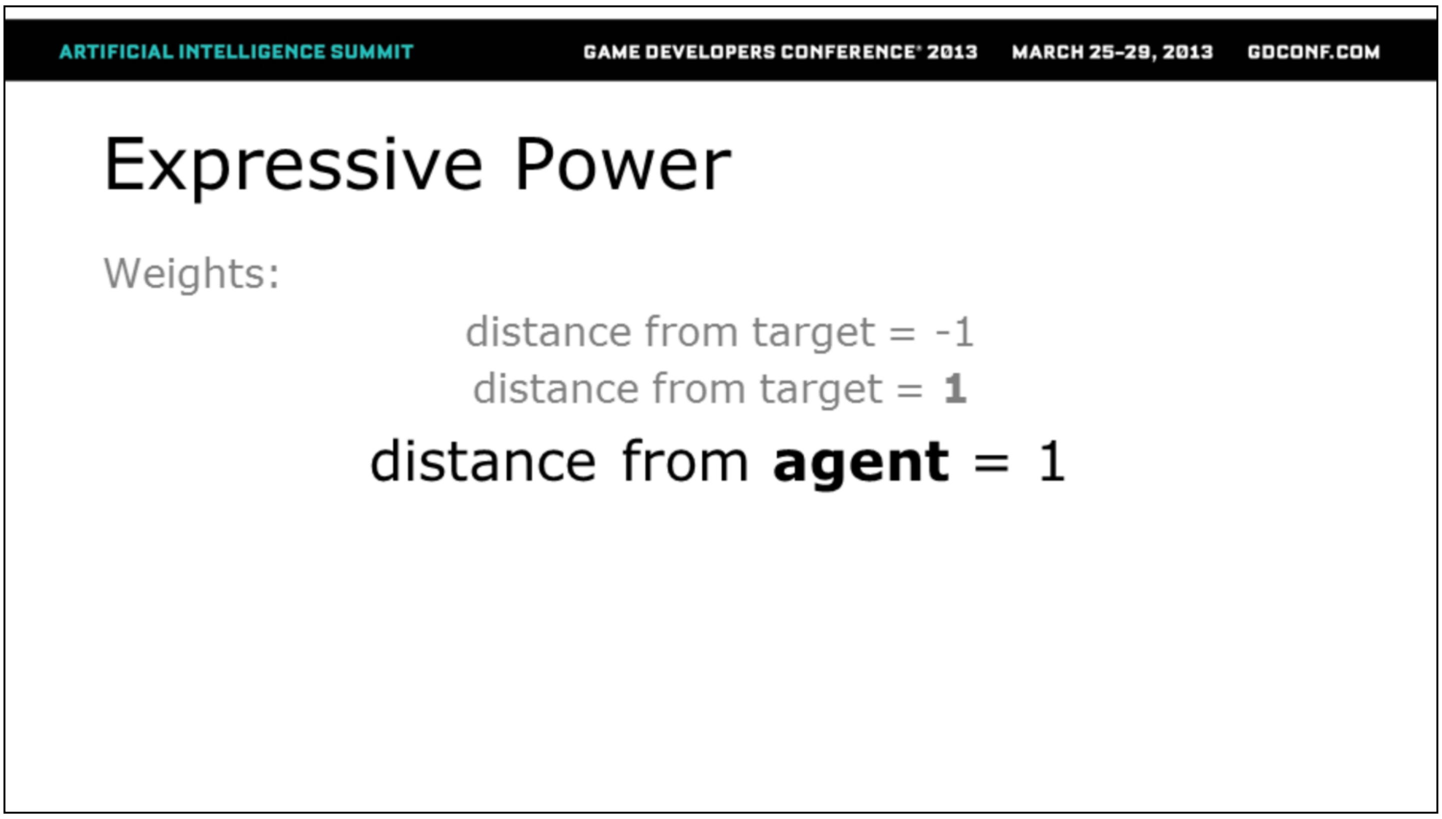
distance from target = -1

Expressive Power

Weights:

distance from target = -1

distance from target = **1**



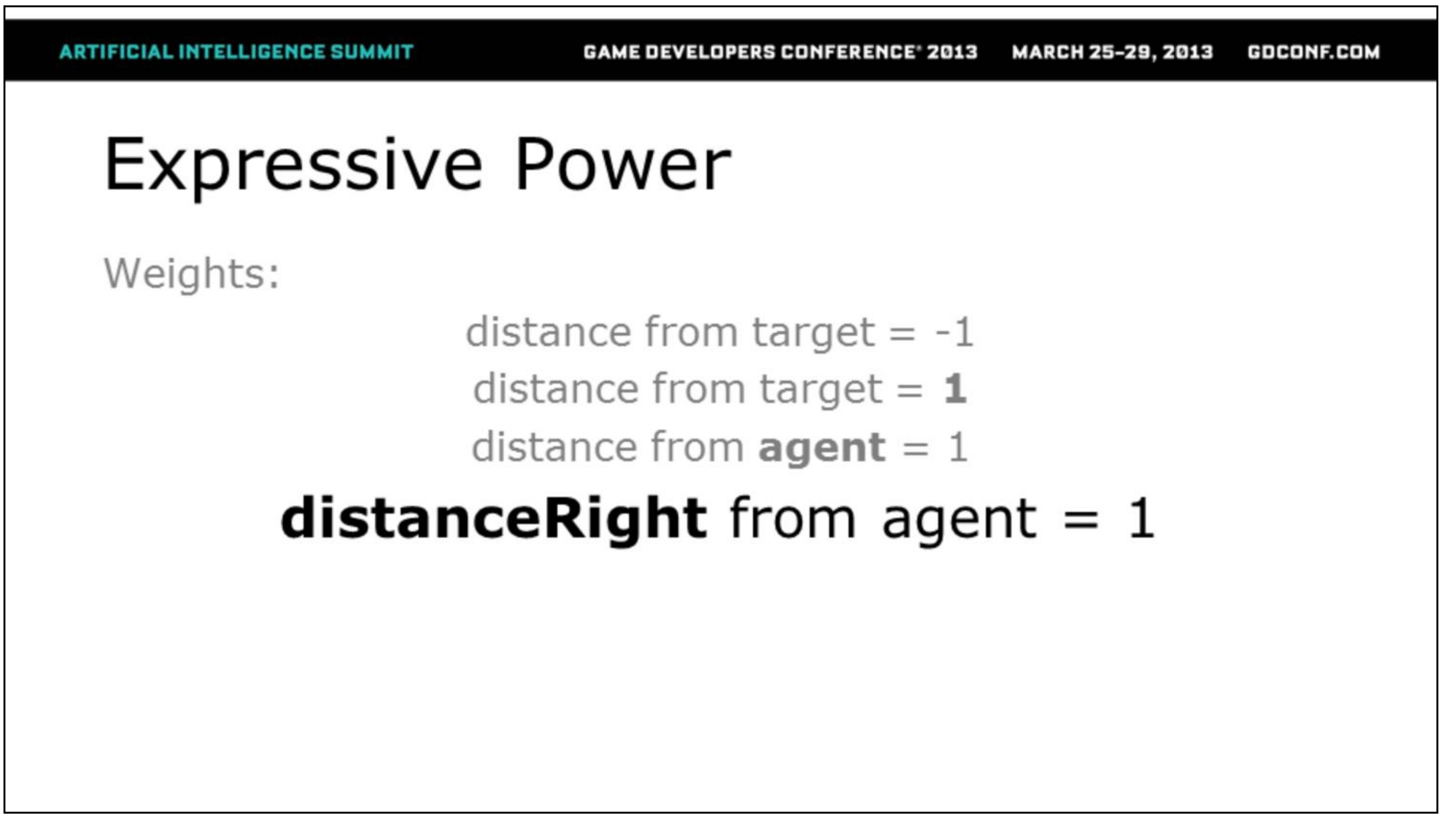
Expressive Power

Weights:

distance from target = -1

distance from target = **1**

distance from **agent** = 1



Expressive Power

Weights:

distance from target = -1

distance from target = **1**

distance from **agent** = 1

distanceRight from agent = 1

Expressive Power

Weights:

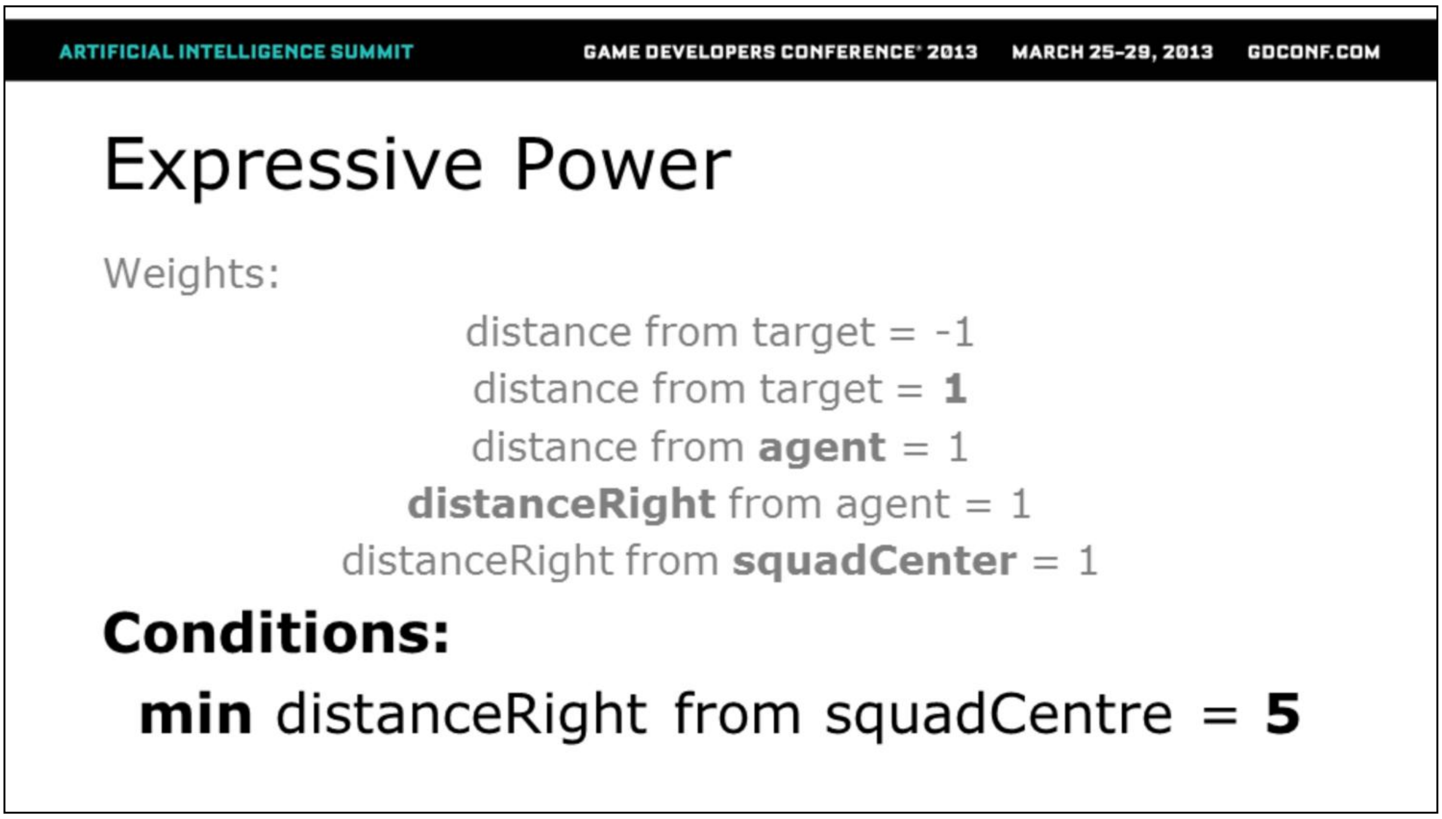
distance from target = -1

distance from target = **1**

distance from **agent** = 1

distanceRight from agent = 1

distanceRight from **squadCenter** = 1



Expressive Power

Weights:

distance from target = -1

distance from target = **1**

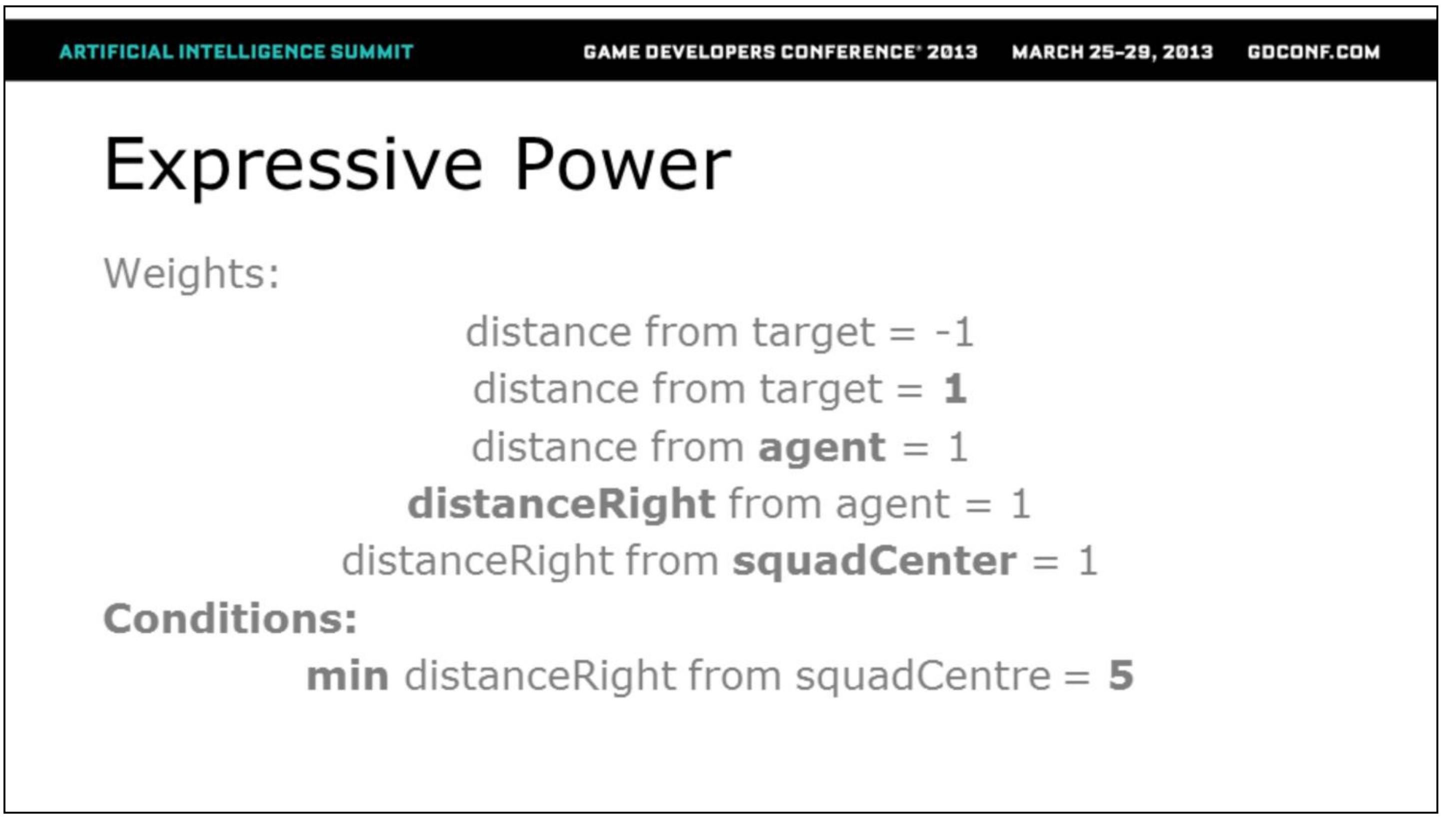
distance from **agent** = 1

distanceRight from agent = 1

distanceRight from **squadCenter** = 1

Conditions:

min distanceRight from squadCentre = **5**



Expressive Power

Weights:

distance from target = -1

distance from target = **1**

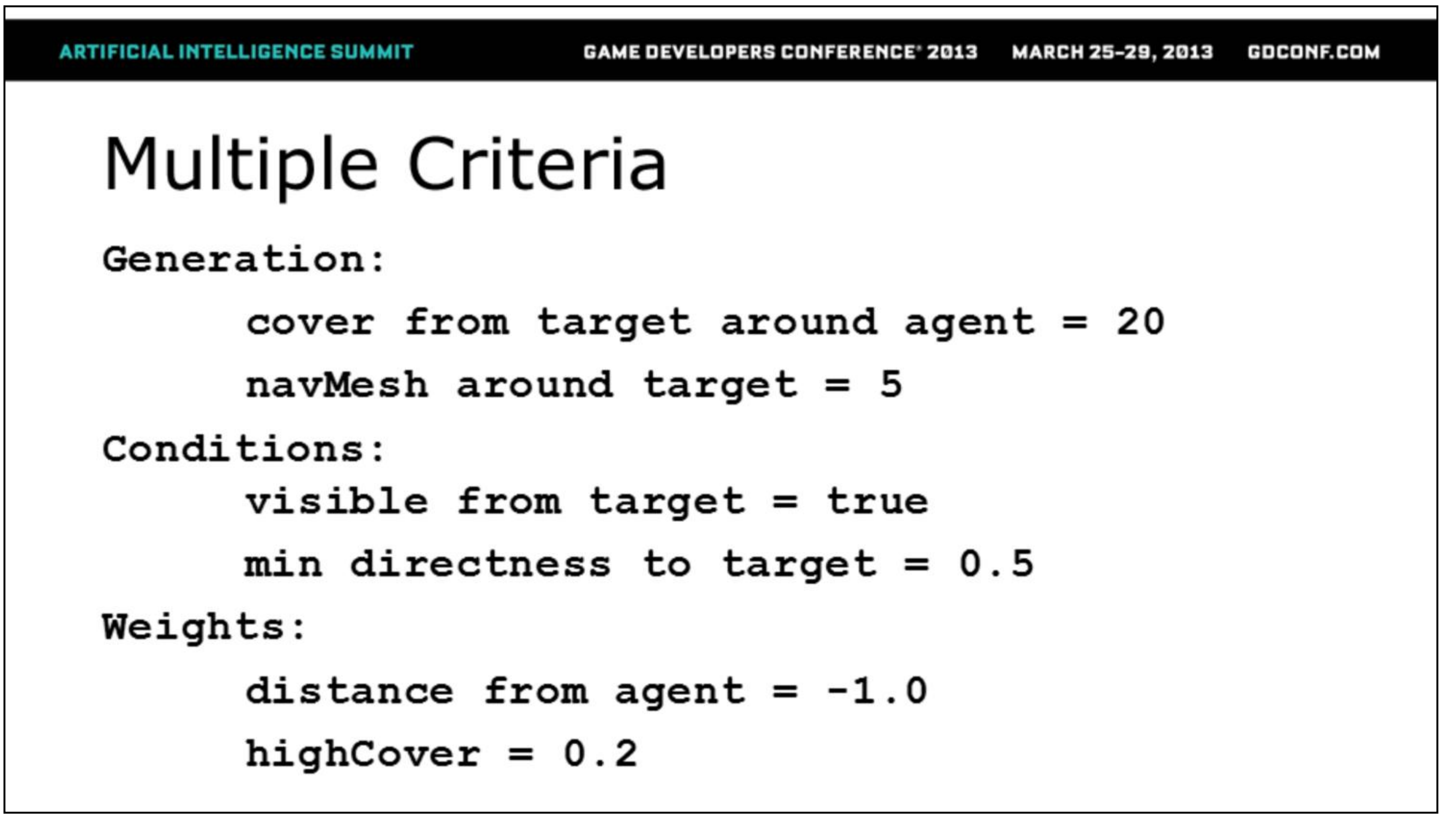
distance from **agent** = 1

distanceRight from agent = 1

distanceRight from **squadCenter** = 1

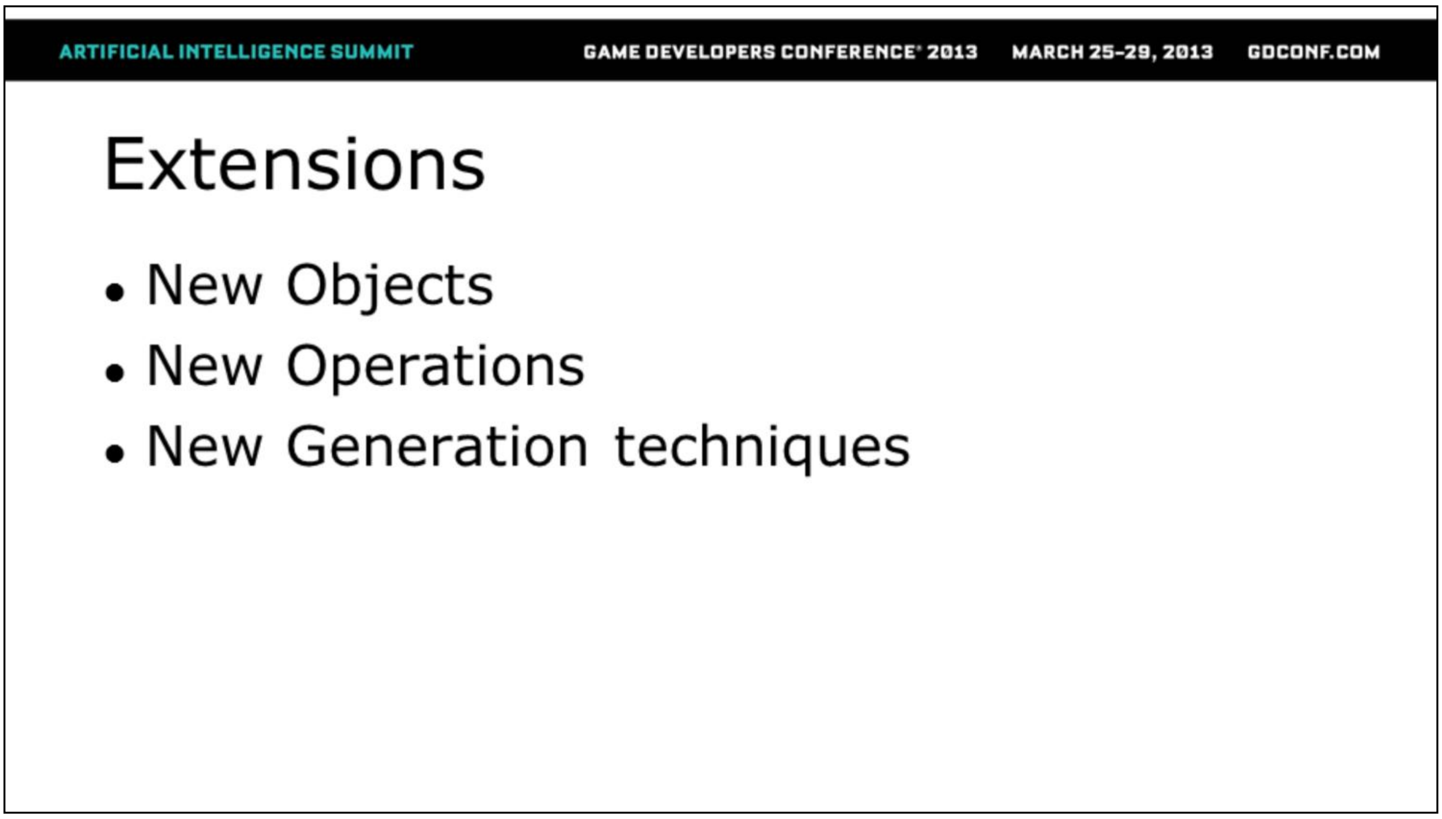
Conditions:

min distanceRight from squadCentre = **5**



Multiple Criteria

```
Generation:
      cover from target around agent = 20
      navMesh around target = 5
Conditions:
      visible from target = true
      min directness to target = 0.5
Weights:
      distance from agent = -1.0
      highCover = 0.2
```


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Extensions

- New Objects
- New Operations
- New Generation techniques

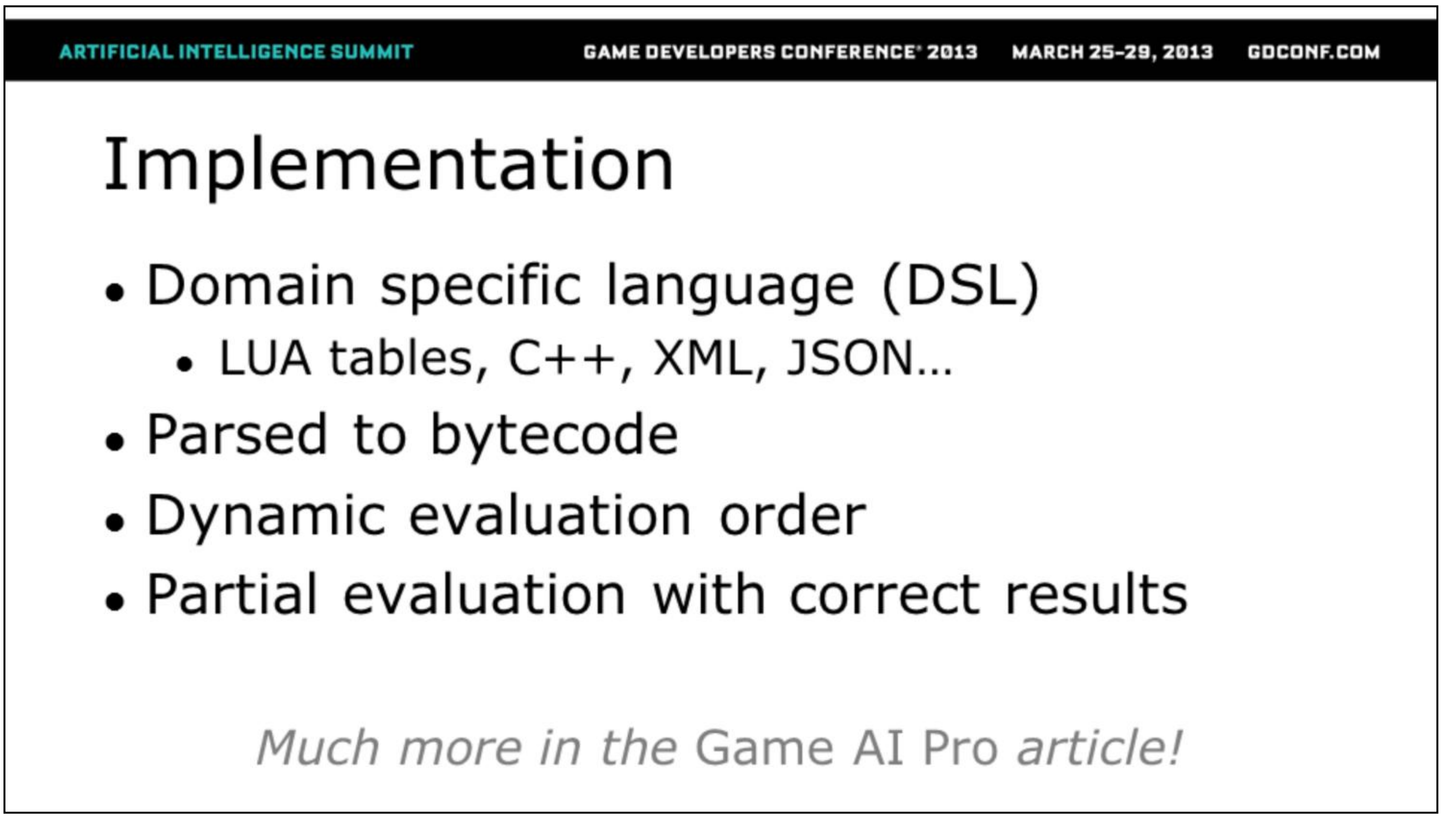
- Did you have other keywords in the query language?

Hell yes :-) The version that ships with the Crytek SDK has perhaps 70 keywords. A few notables...

- Generators: Grid (mentioned in talk), Entities (point at every entity of given type), CurrentPos (generate just one point at position of given Object), Cover (as talk). NavMesh (in talk) is currently Xaviant-only.

- Objects: the player, the agent querying, his reference point

- Operations: various cover quality keywords, Reachable, Directness, CoverDensity, CameraCenter (for position on screen), distance to nearest friend, distance to nearest foe, Random, HeightRelative, ElevationAngle, CanReachBefore (am I closer to this than my enemy is?) I hope those names give you the rough idea.


ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Implementation

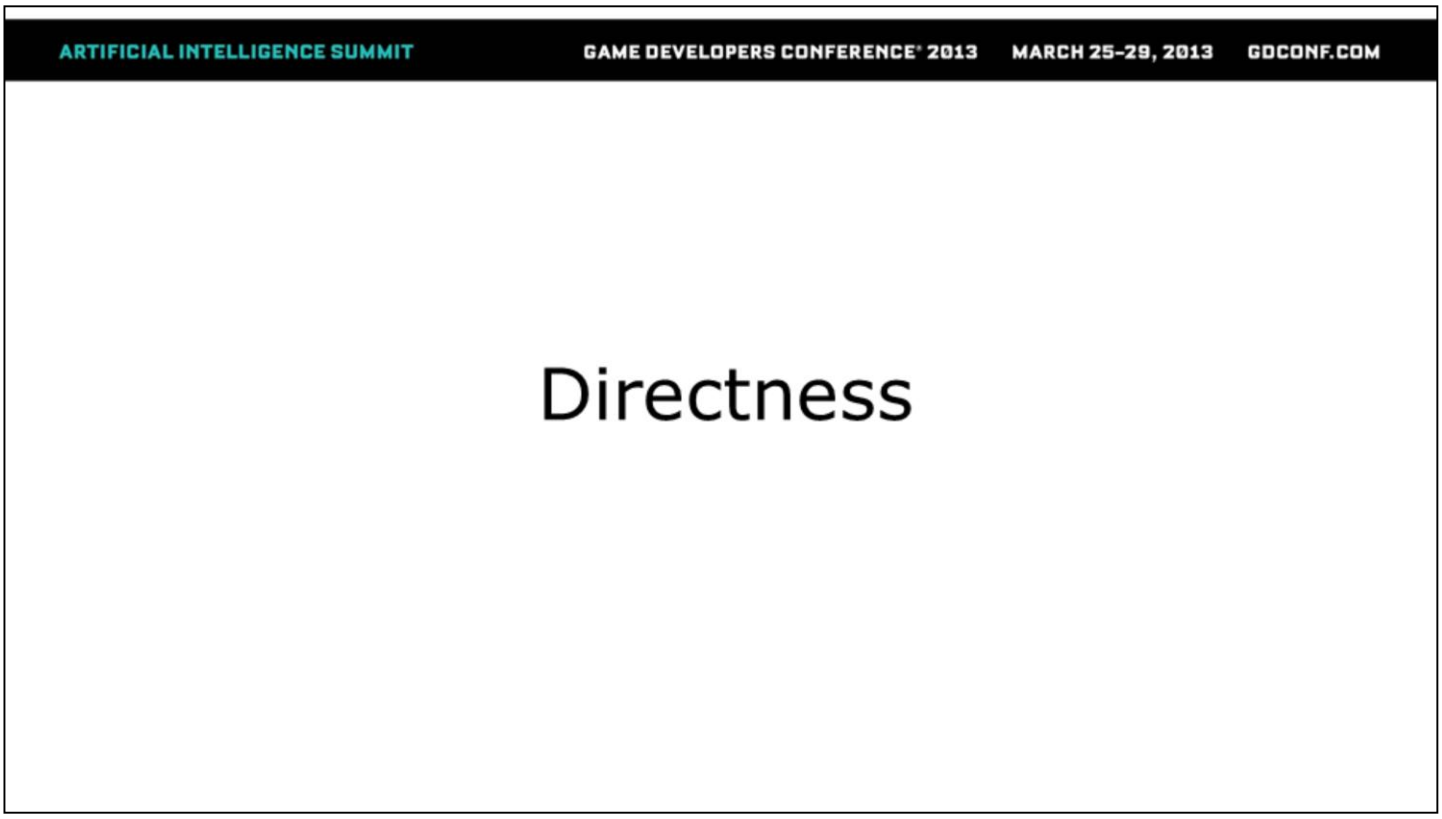
- Domain specific language (DSL)
  - LUA tables, C++, XML, JSON...
- Parsed to bytecode
- Dynamic evaluation order
- Partial evaluation with correct results

Much more in the Game AI Pro *article!*

- How much work was it to implement the parser etc?

A few days I think. All the queries are built up as Lua tables, so the Lua parser does most of the work, then we query that using existing APIs. Each element of the table consists of a string and a value, so something like: min_distance_from_player = 7 . The element names are tokenized, using the underscores as separators. The keywords are looked up in a string map to get bytecode values and syntax errors are generated if it doesn't fit the grammar. A couple of hundred lines for the parser itself? The supporting code - runtime API, scheduling, etc - is on top of that.

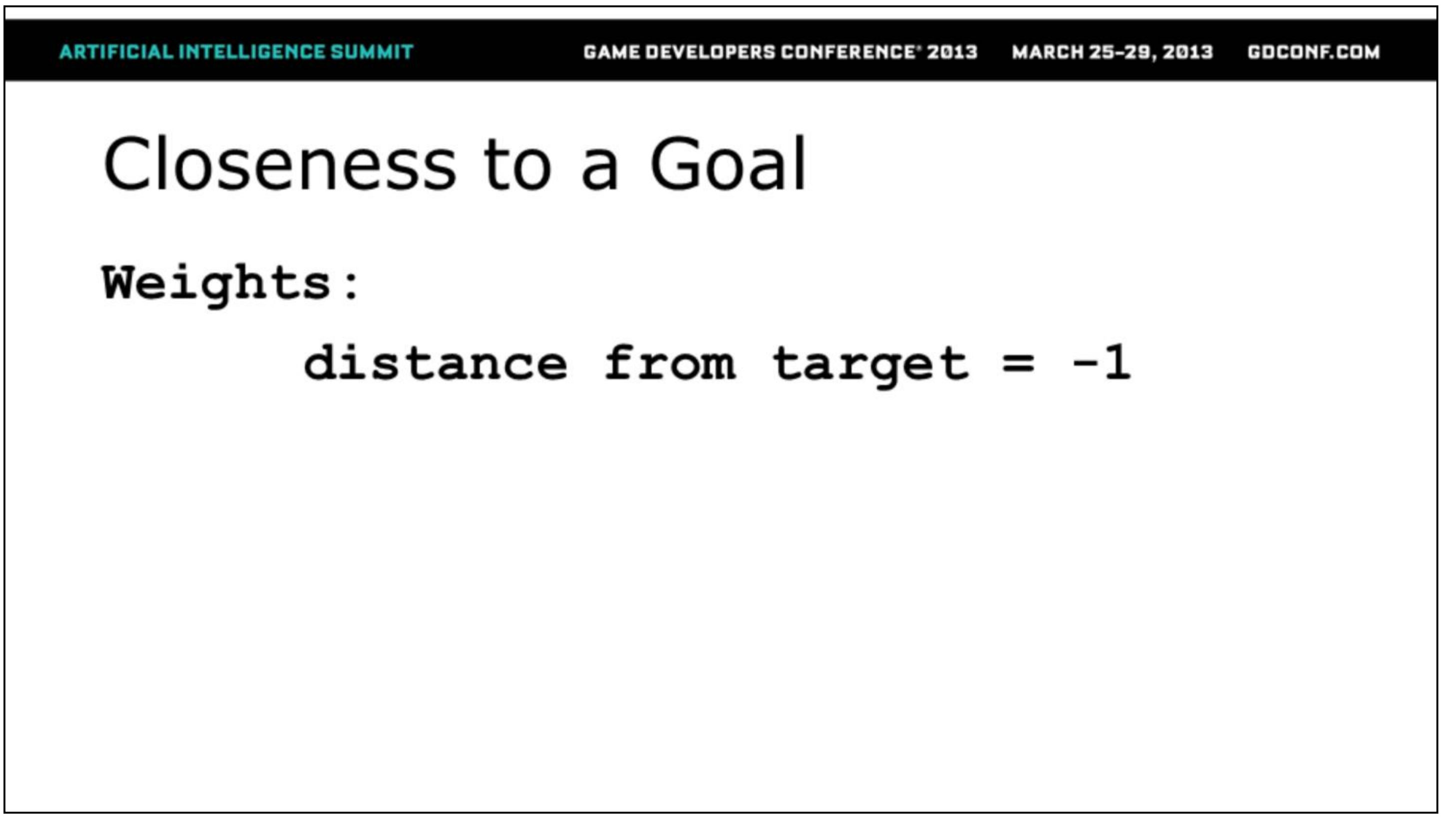
The dynamic evaluation scheme (i.e. optimising) is the most complex part. Not everyone would need that.



ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

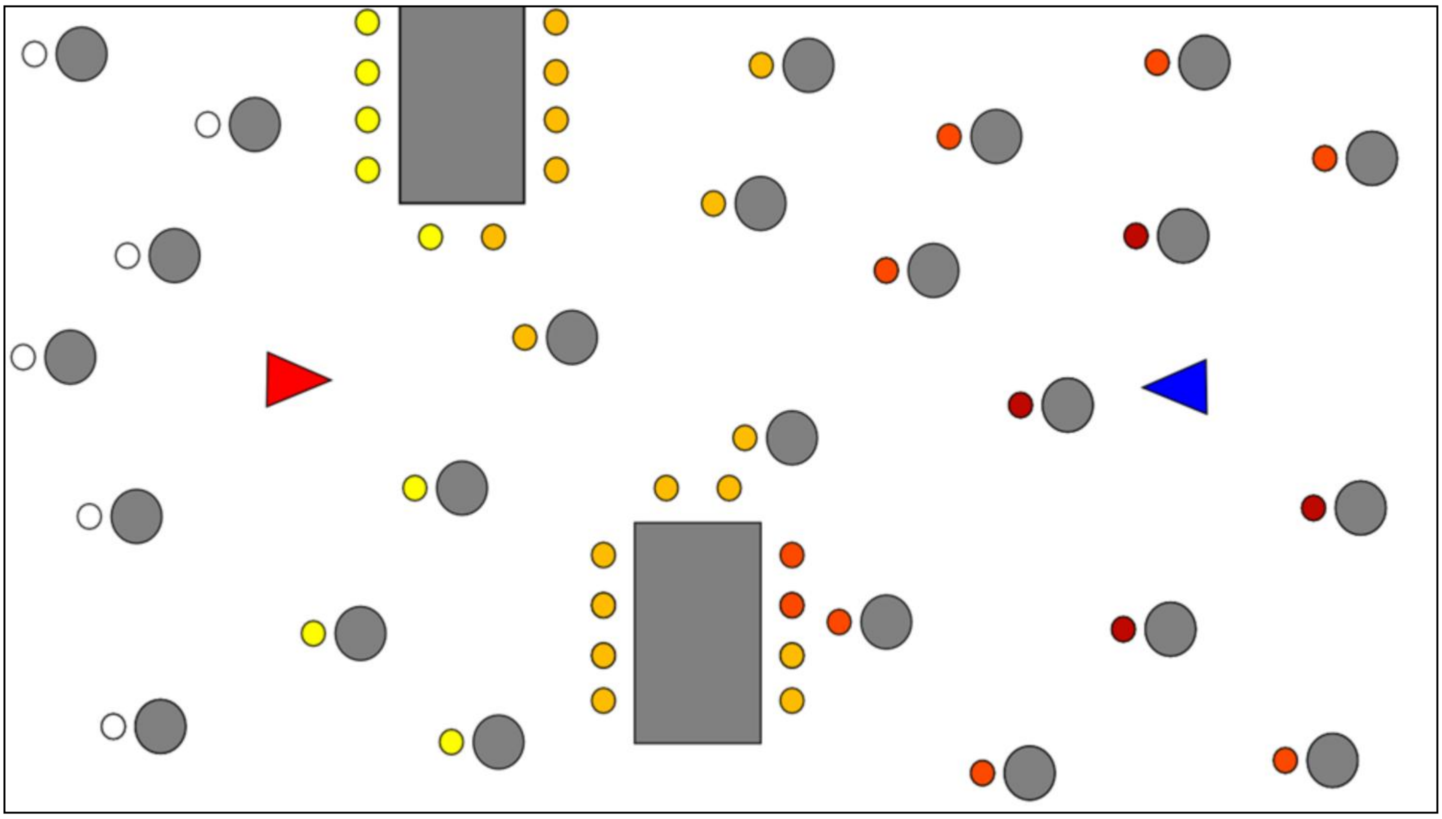
Directness

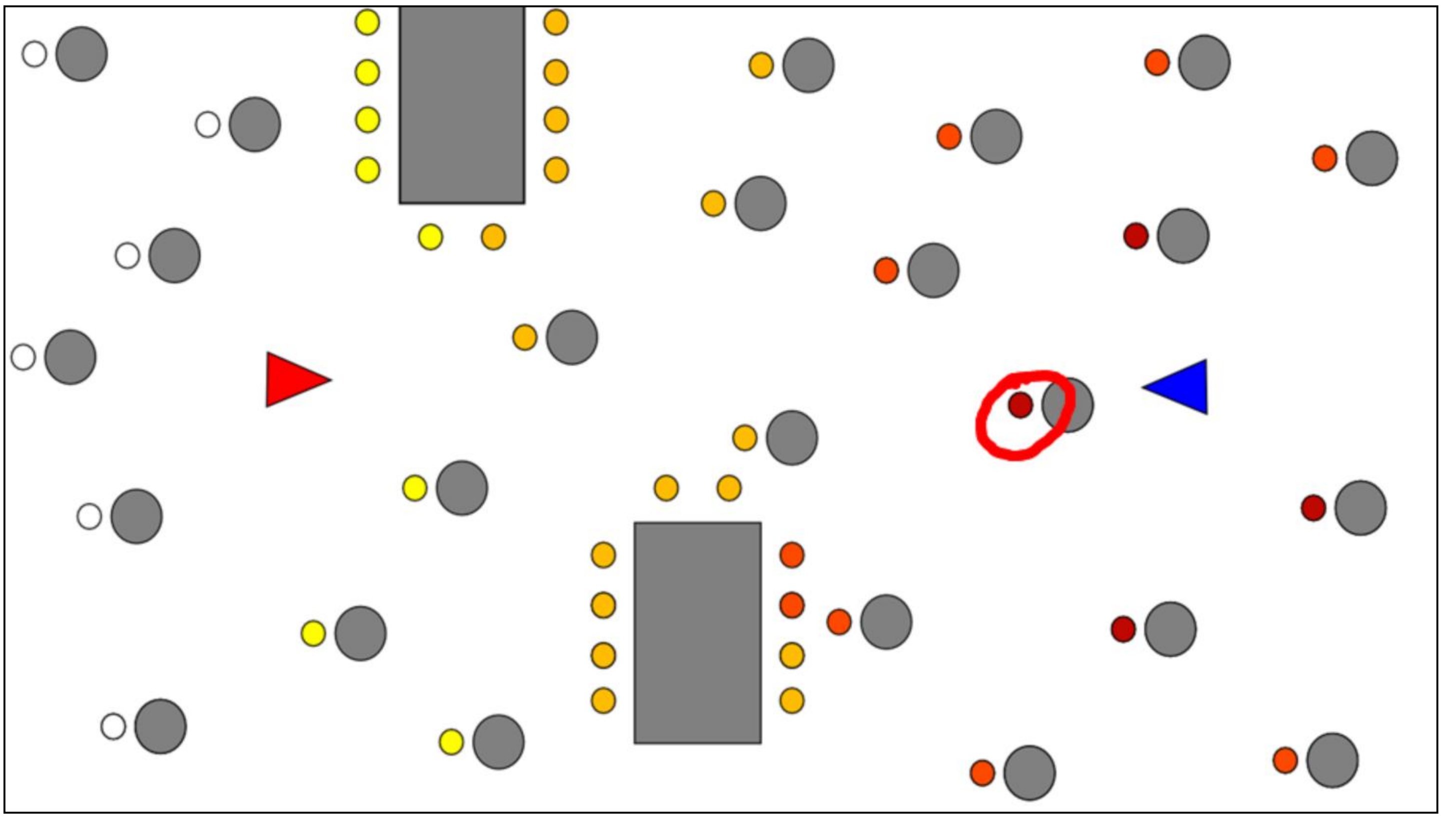
A simple operation most people could apply in their games

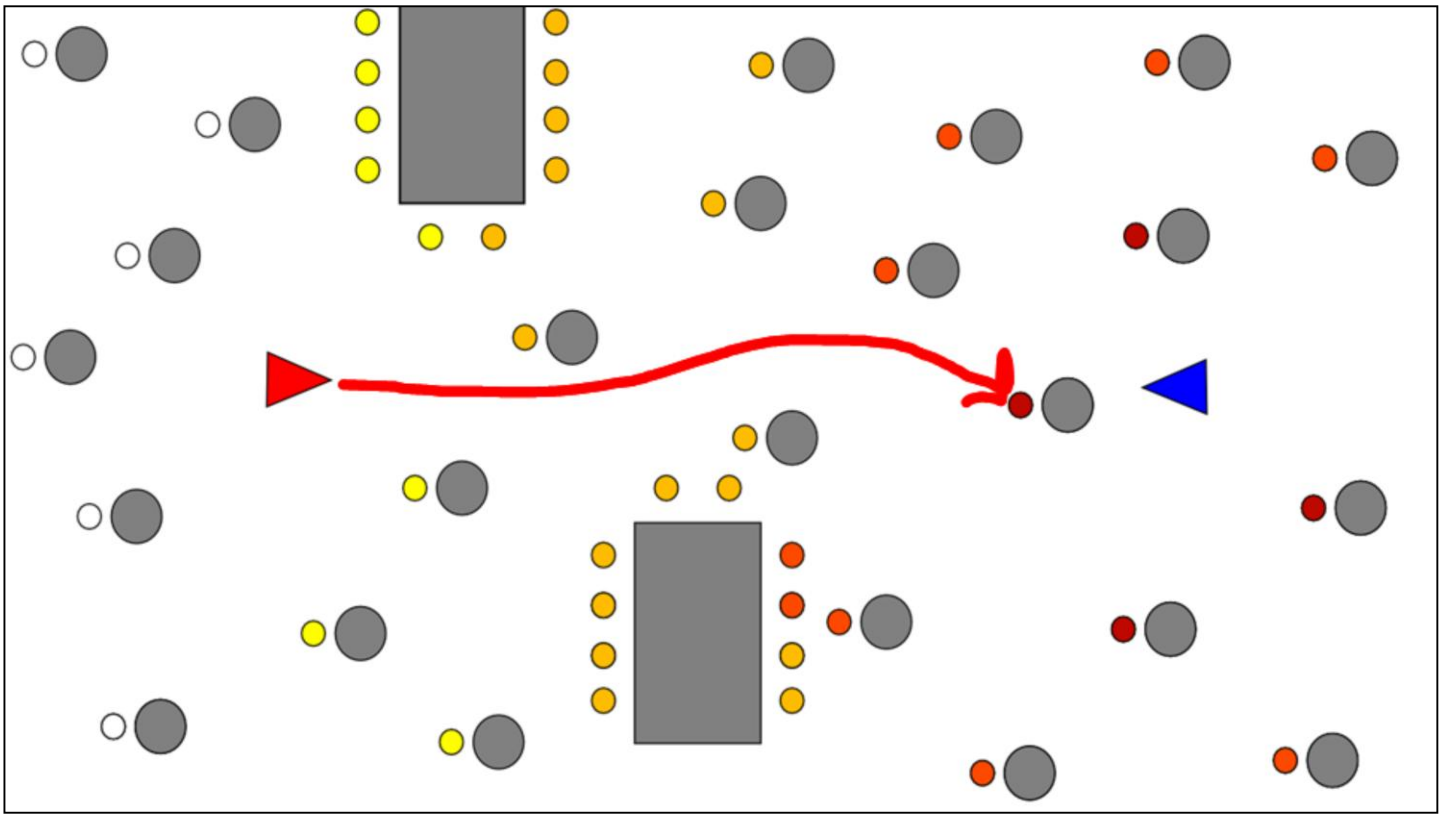


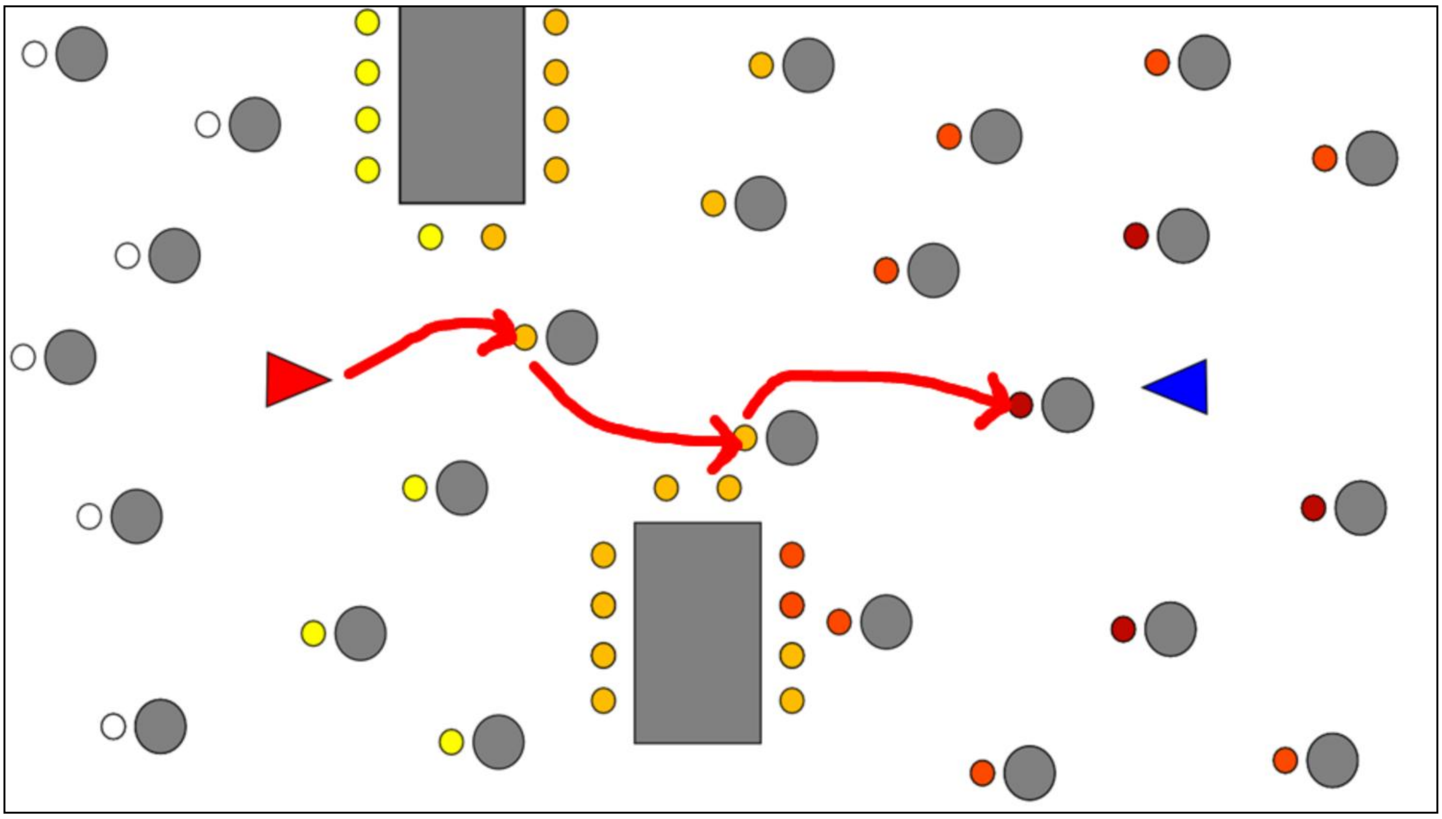
Closeness to a Goal

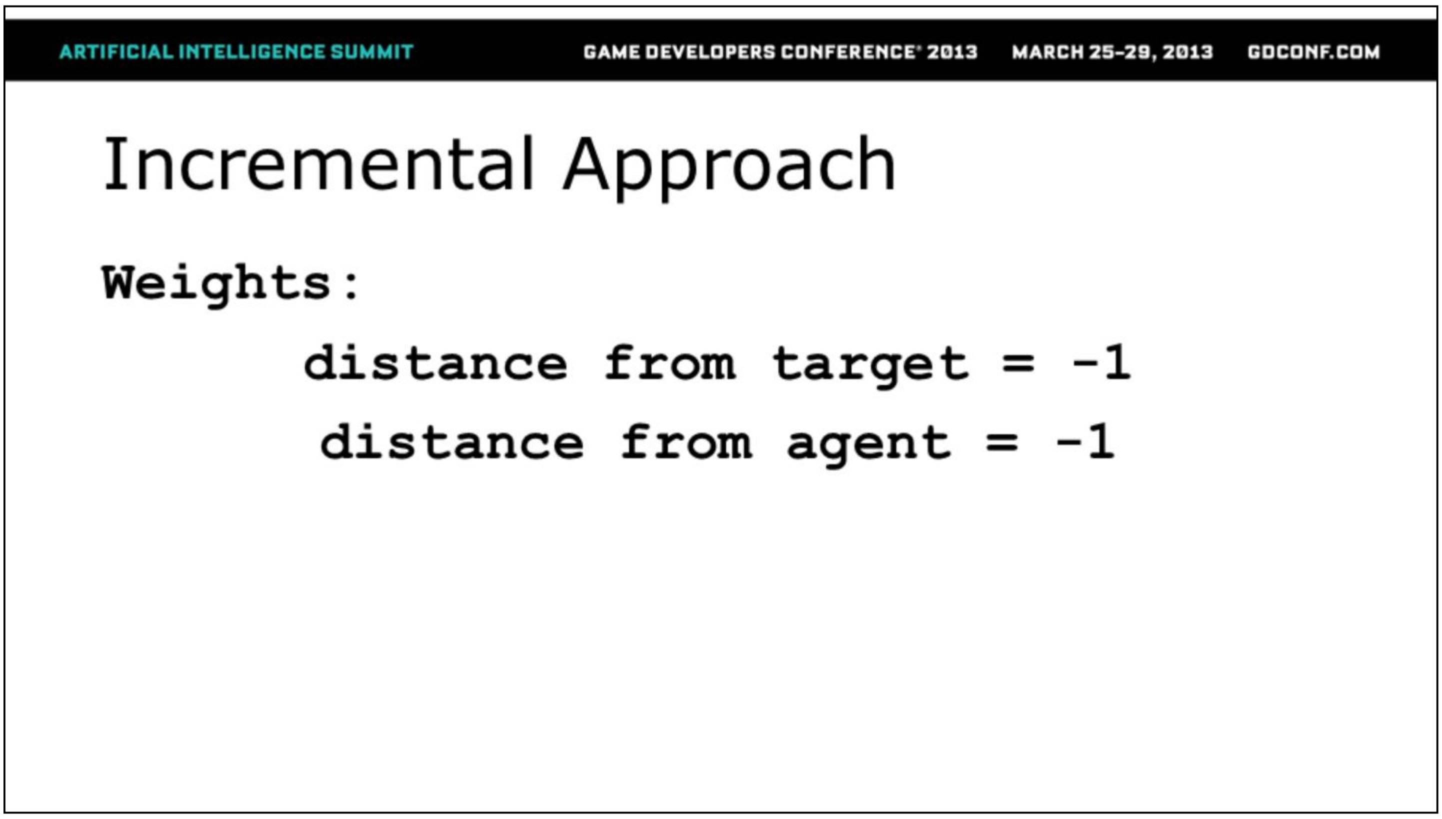
```
Weights:
        distance from target = -1
```









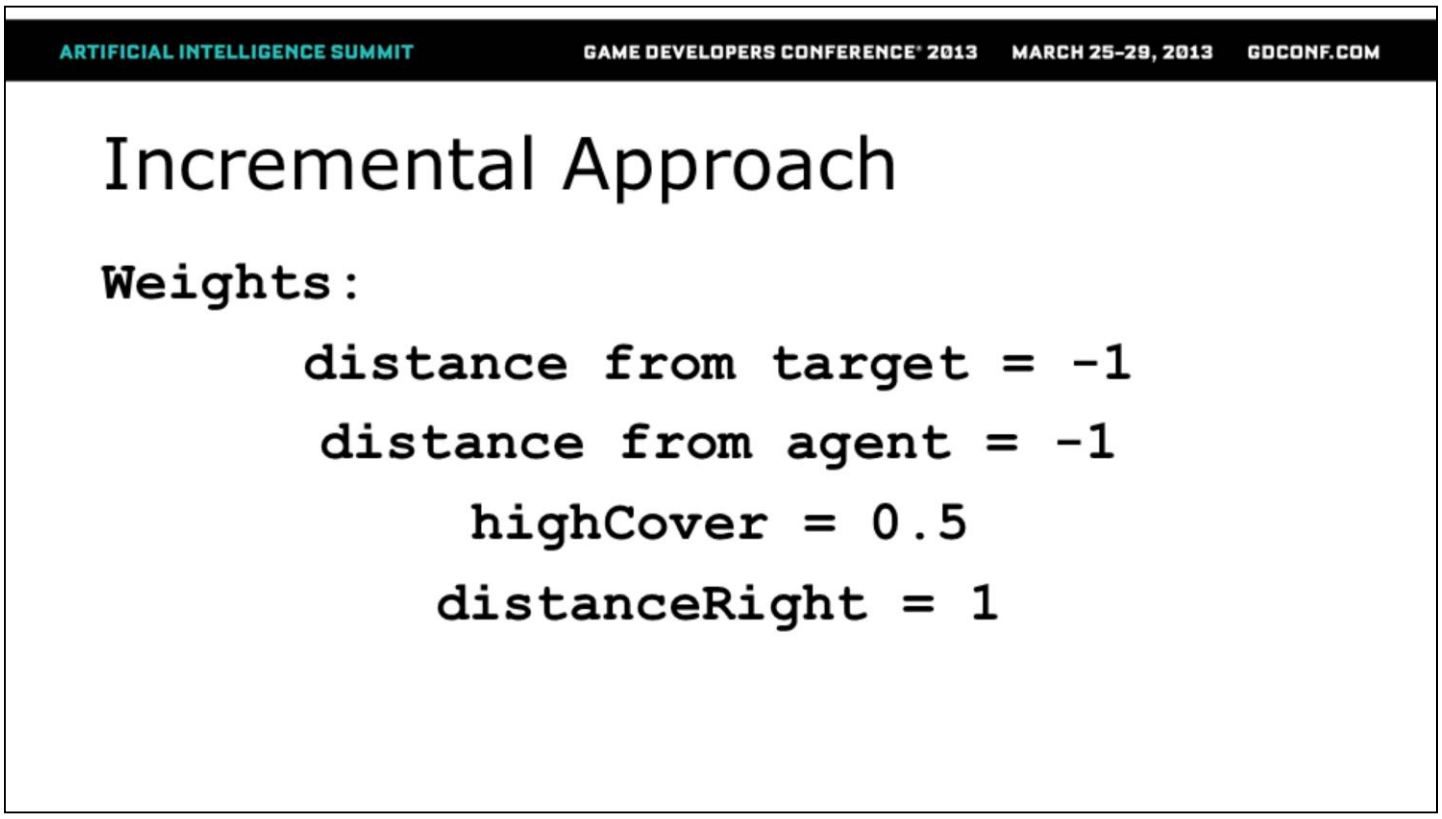

ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25-29, 2013 GDCONF.COM

Incremental Approach

```
Weights:
        distance from target = -1
         distance from agent = -1
```

As I mentioned, there is a problem with this “balancing”. If these two distances are combined linearly, intuitively you expect to get a midpoint. If you were to change the weightings, you’d expect to bias the midpoint one way or the other.

In fact, all the points on the line between the two objects score equally. For every metre you move in one direction or the other, the loss of score on one weighting is equal to that gained in the other. Hence the result will usually be more or less random selection between points according to which is closest to this line between the objects. This isn’t necessarily going to be obvious from results and I think this is a common mistake. If you don’t bias equally, points as close as possible to the higher biased point will be selected, rather than just a different fractional midpoint.



Incremental Approach

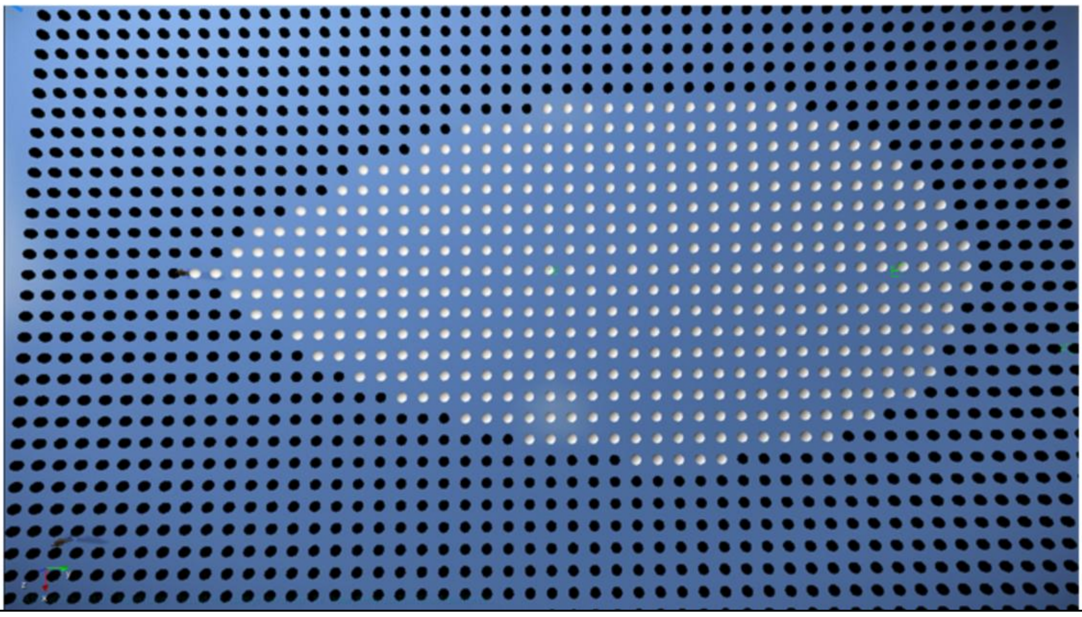
```
Weights:
        distance from target = -1
         distance from agent = -1
               highCover = 0.5
             distanceRight = 1
```

Reframing the Problem

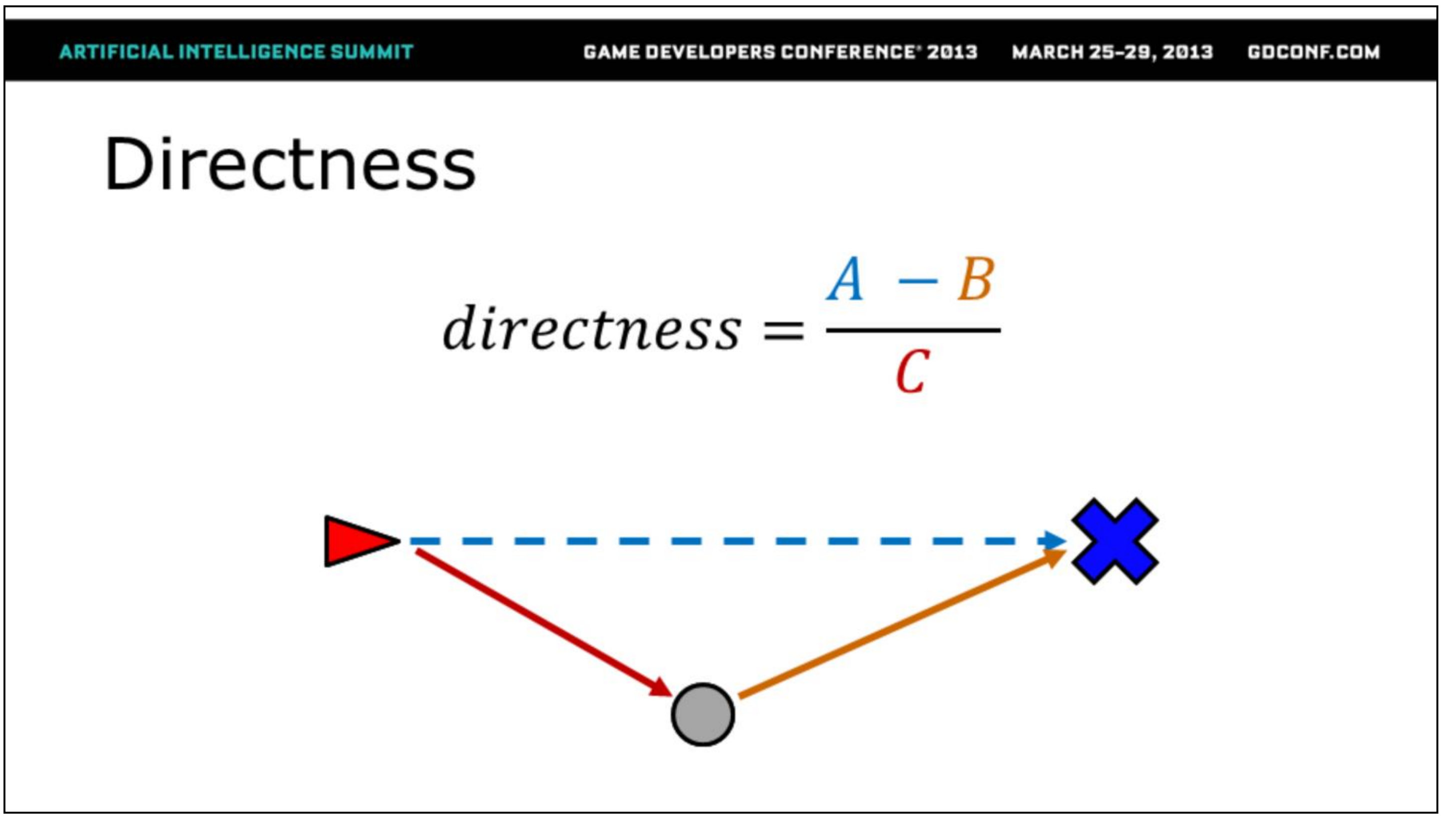
Required:

- Convergence towards a goal
- Control within that constraint

min directness = 0.8



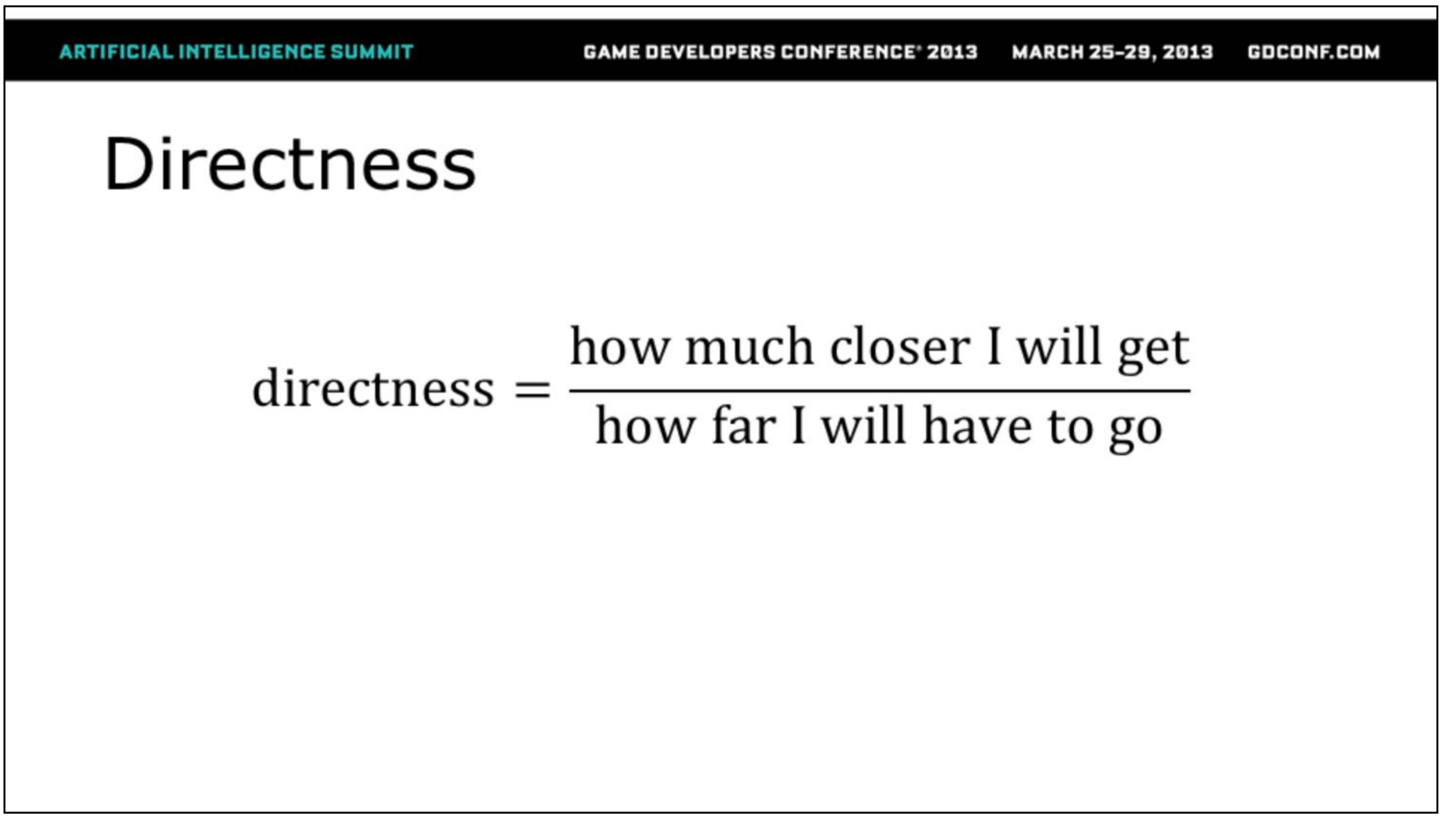
In all these diagrams, the agent is on the left, the object we are using directness to converge upon is on the right.


ARTIFICIAL INTELLIGENCE SUMMIT
GAME DEVELOPERS CONFERENCE® 2013
MARCH 25–29, 2013
GDCONF.COM

Directness

$$directness = \frac{A - B}{C}$$

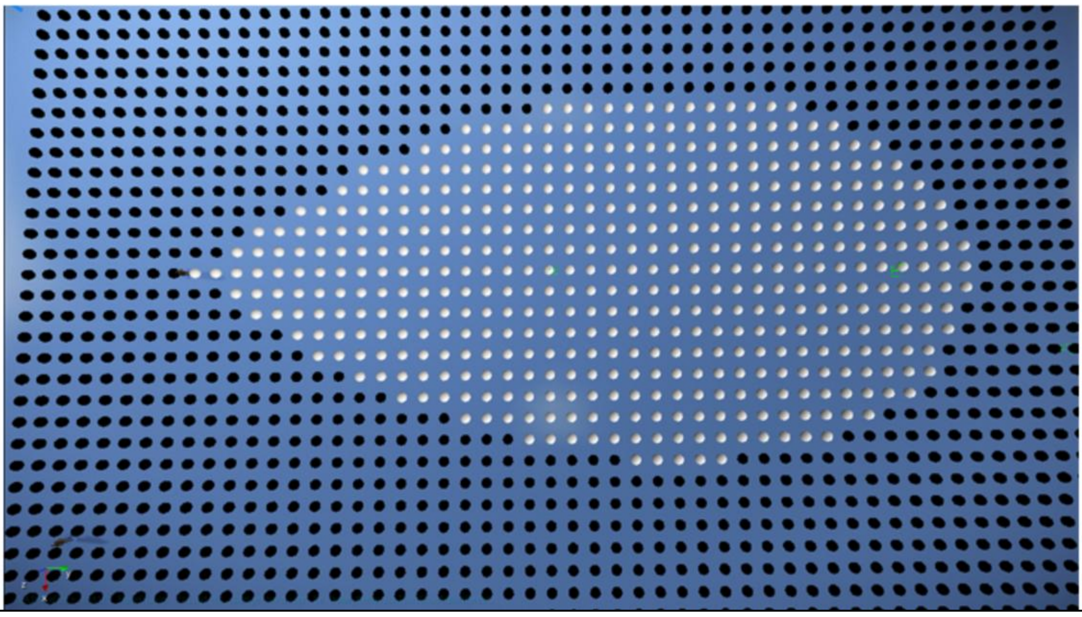
Triangle is agent, grey circle is a point we are considering, the cross is the goal object we are converging on.



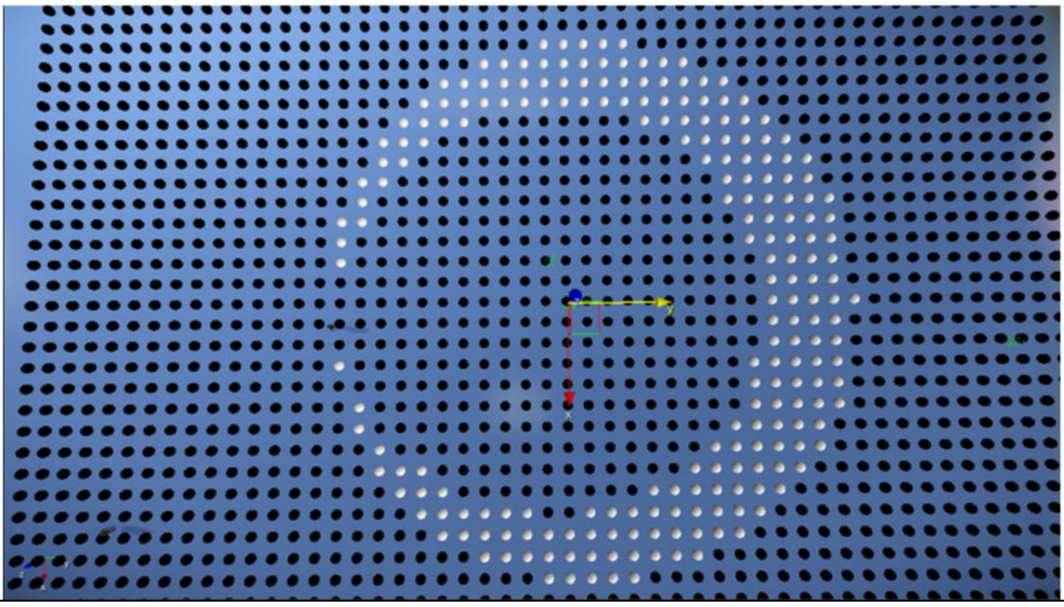
Directness

$$\text{directness} = \frac{\text{how much closer I will get}}{\text{how far I will have to go}}$$

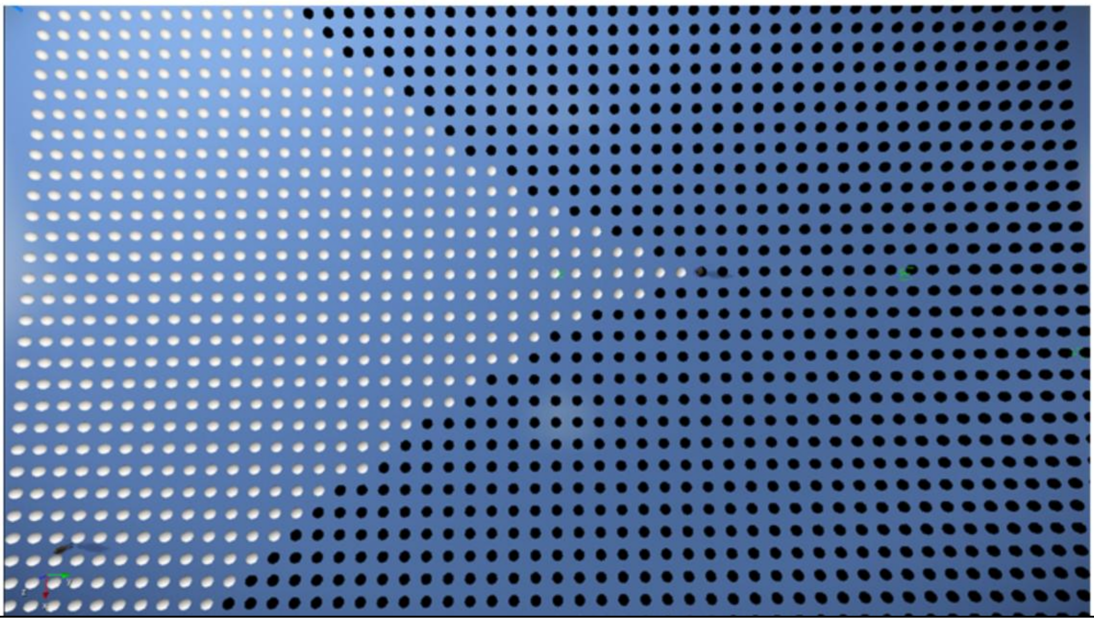
min directness = 0.8

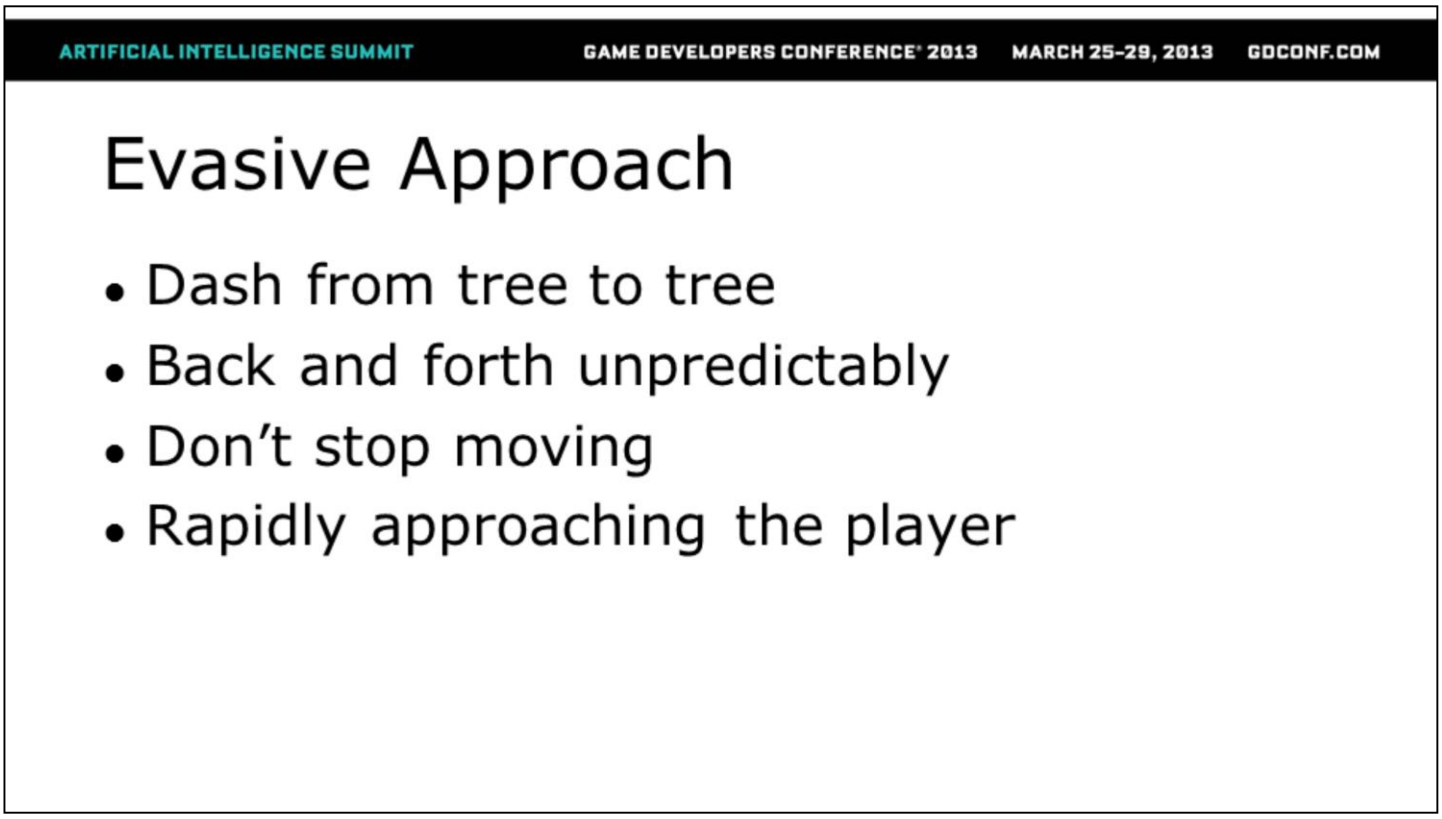


directness = 0



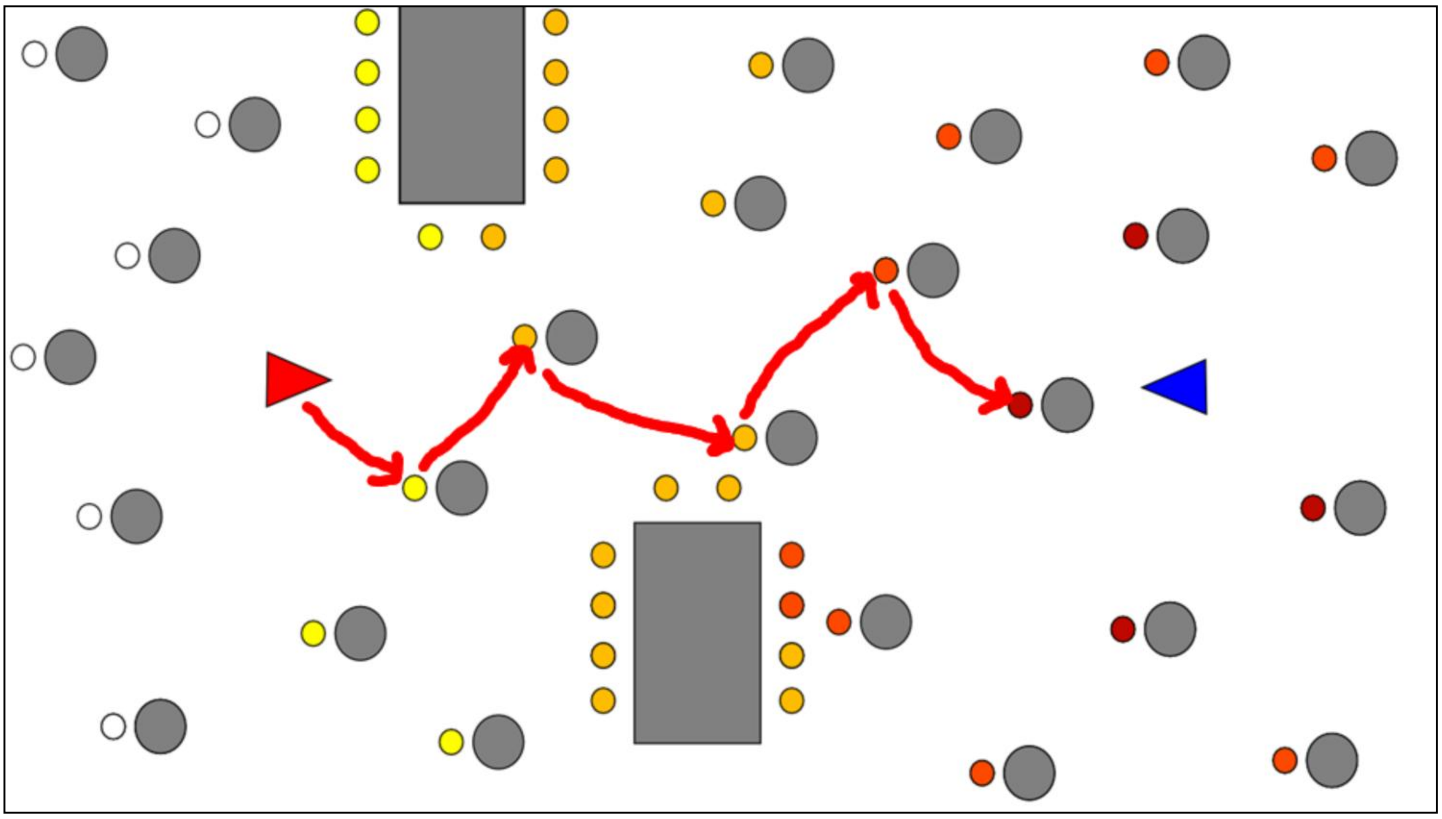
max directness = -0.95

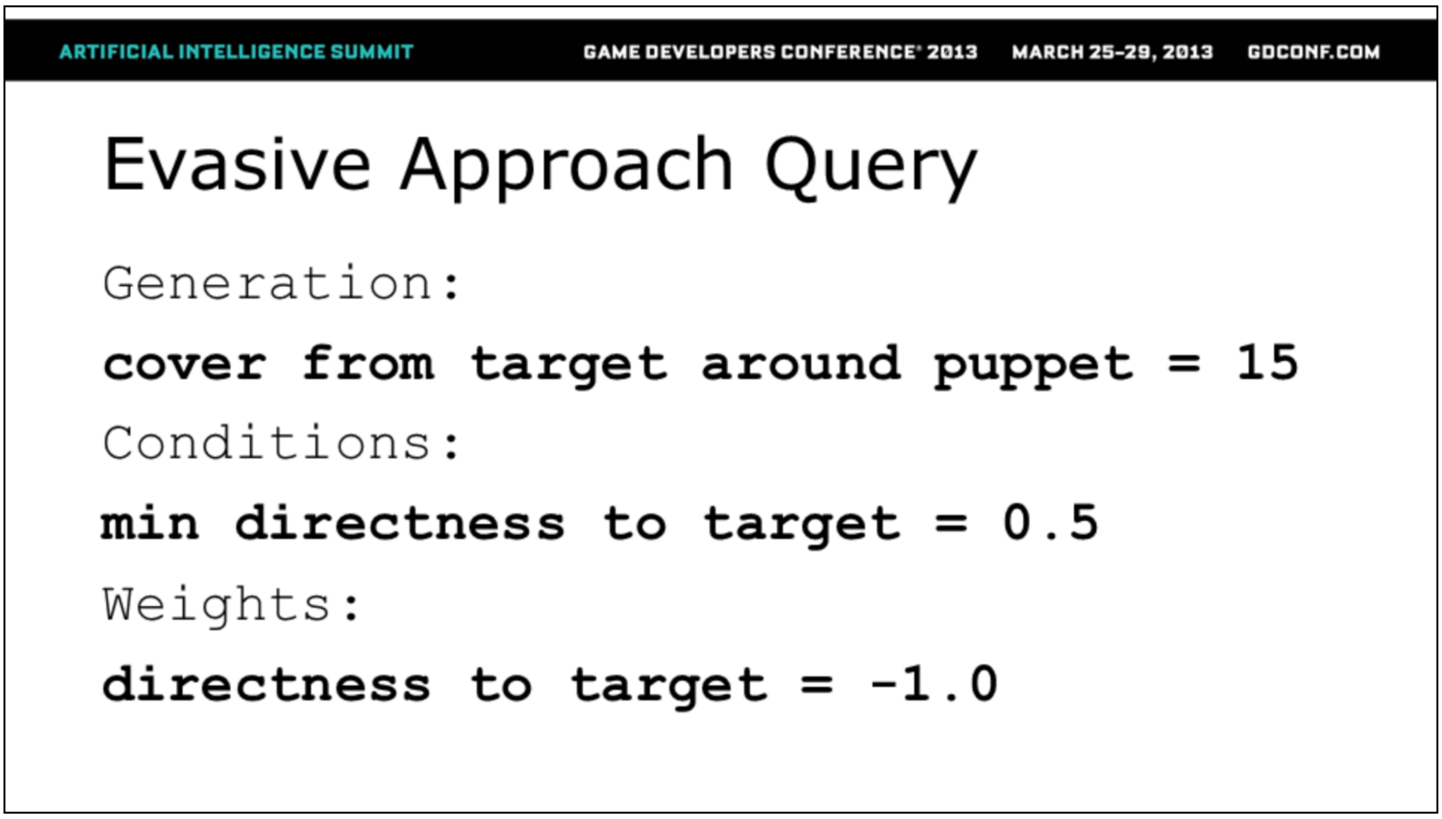




Evasive Approach

- Dash from tree to tree
- Back and forth unpredictably
- Don’t stop moving
- Rapidly approaching the player

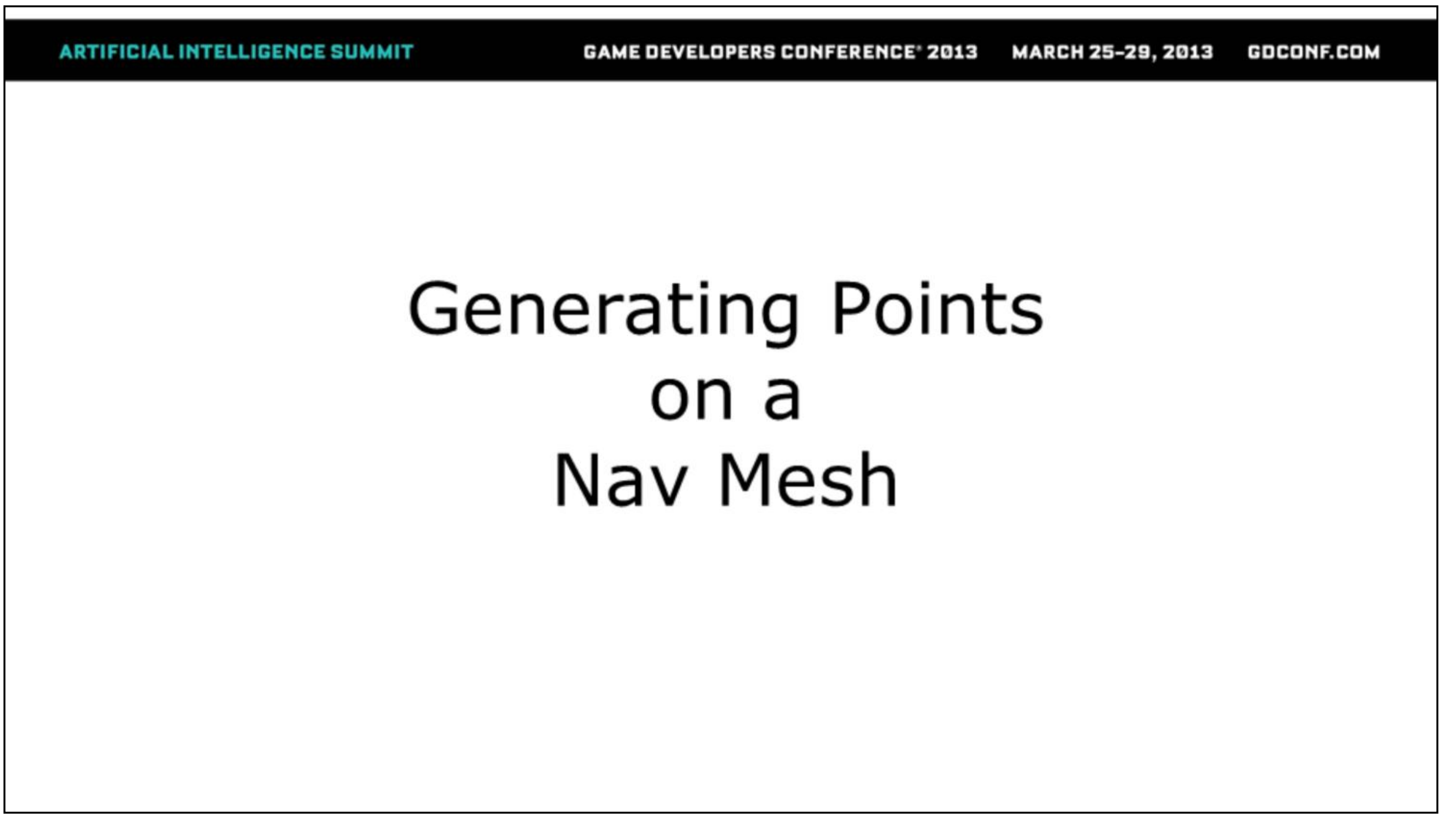




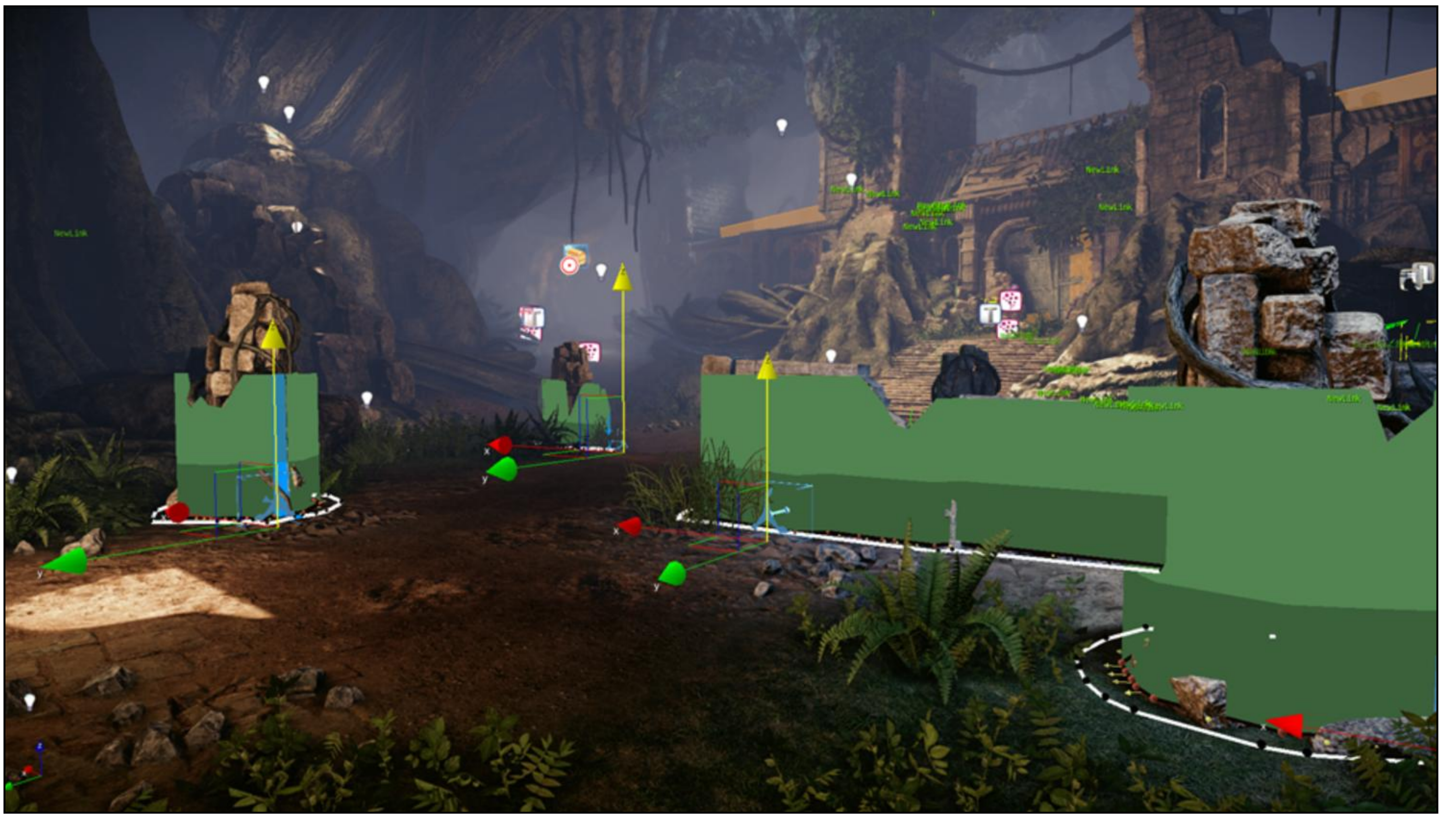
Evasive Approach Query

```
Generation:
cover from target around puppet = 15
Conditions:
min directness to target = 0.5
Weights:
directness to target = -1.0
```

This is all that is required to achieve the zigzag pattern.



Generating Points on a Nav Mesh

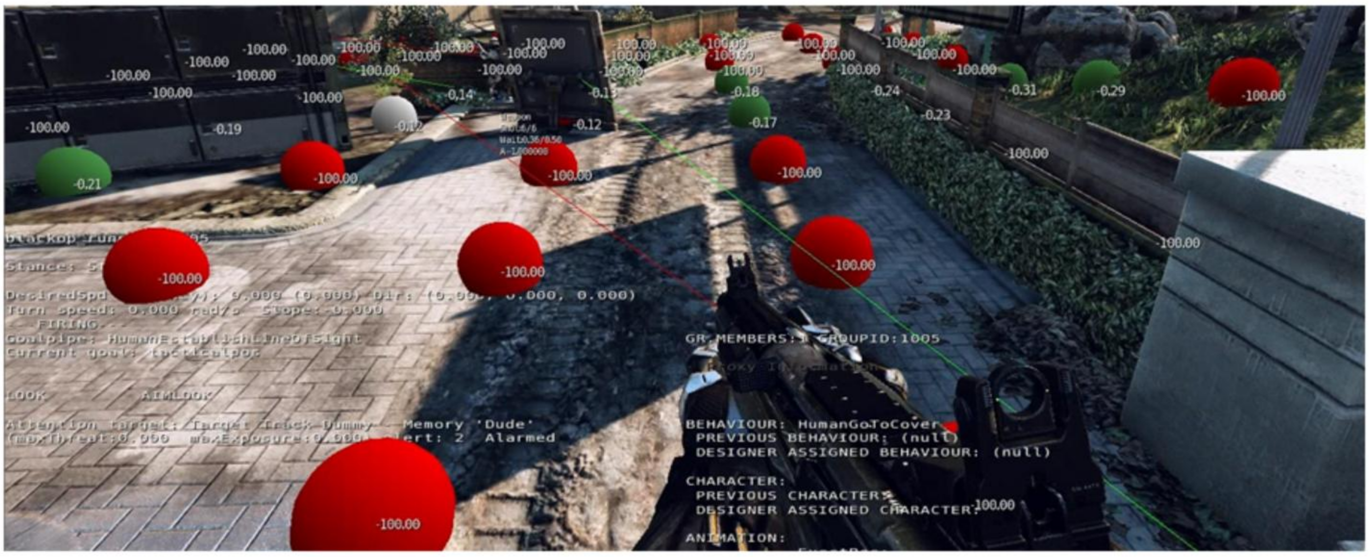


CryEngine semi-automatic cover mapping.

Screenshot from Lichdom, for which we developed extensions such as generation of points on a navigation grid.

ARTIFICIAL INTELLIGENCE SUMMIT GAME DEVELOPERS CONFERENCE® 2013 MARCH 25–29, 2013 GDCONF.COM

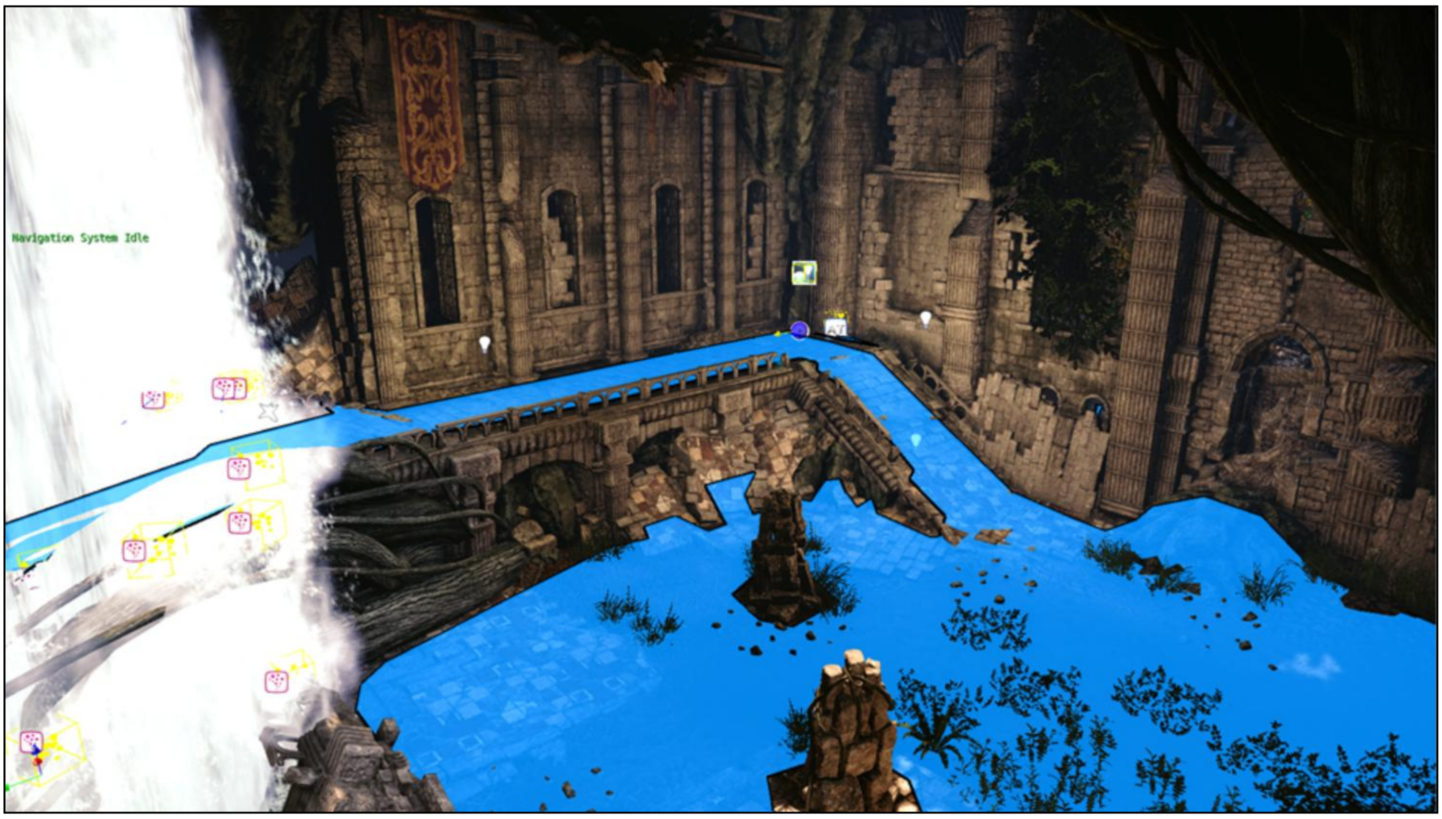
Generation in an Open Grid



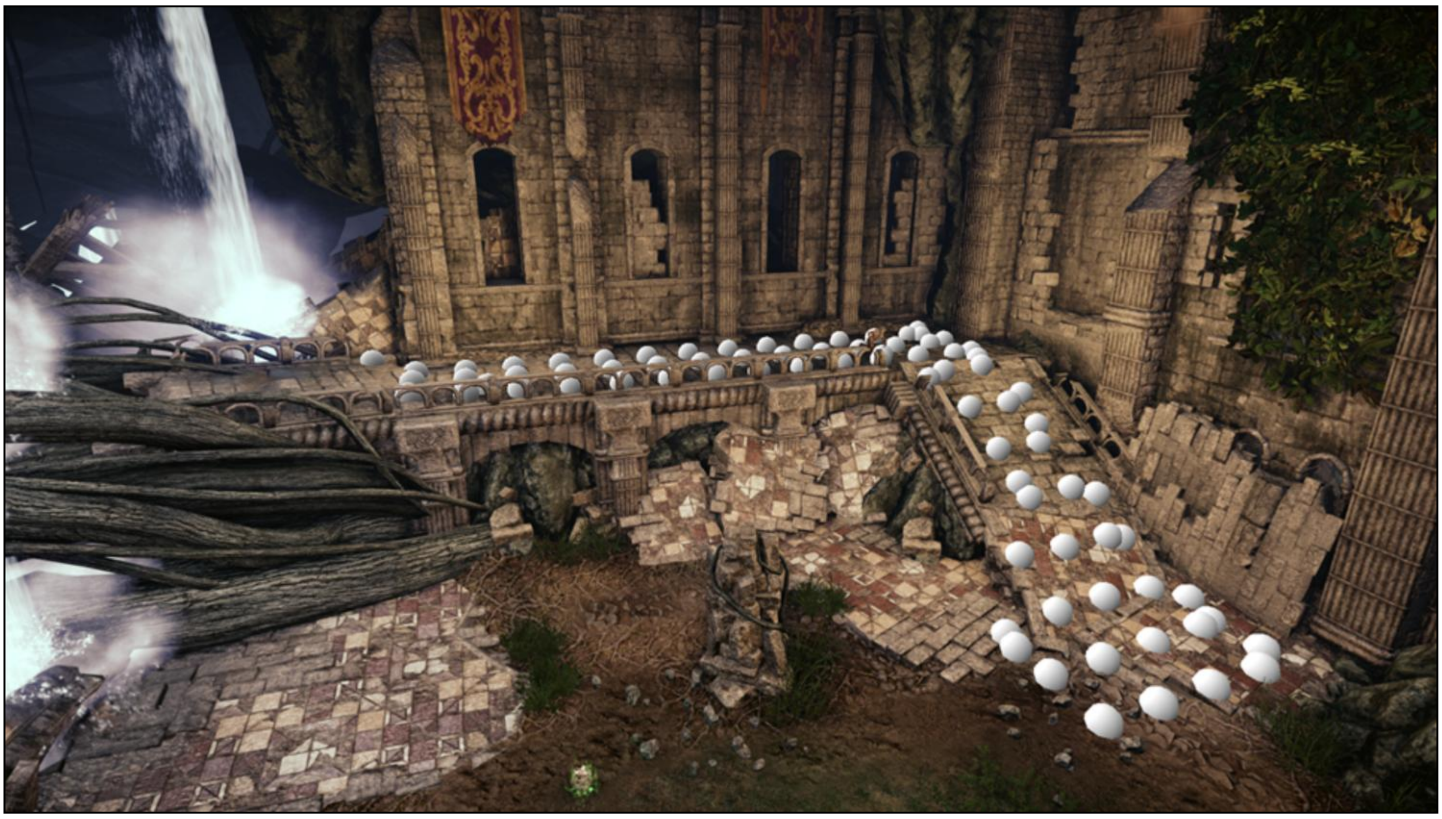
```
DesiredSpd              : 0.000 (0.000) Dir: (0.000, 0.000, 0.000)
Turn speed: 0.000 rad/s  Slope: 0.000
   FIRING
Goalpipe: HumanEstablishLineOfSight
Current goal: tacticalpos

LOOK          AIMLOOK

Attention target: Target Track Dummy   Memory 'Dude'
(maxThreat:0.000  maxExposure:0.000)  Alert: 2  Alarmed
```

```
GR MEMBERS:1 GROUPID:1005

BEHAVIOUR: HumanGoToCover
 PREVIOUS BEHAVIOUR: (null)
 DESIGNER ASSIGNED BEHAVIOUR: (null)

CHARACTER:
 PREVIOUS CHARACTER:
 DESIGNER ASSIGNED CHARACTER:

ANIMATION:
```

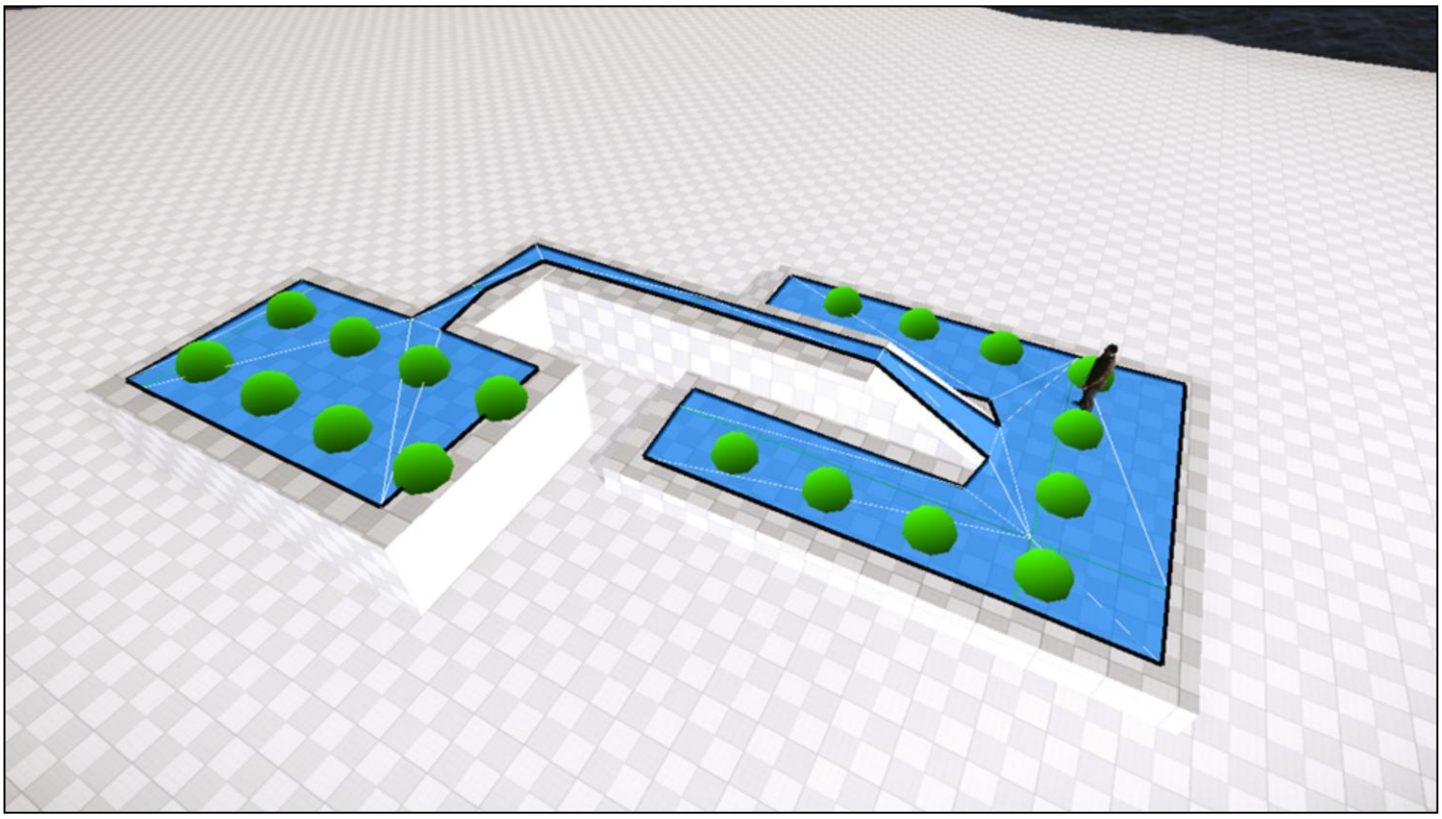
Debugging screenshot from Crysis 2 released product.

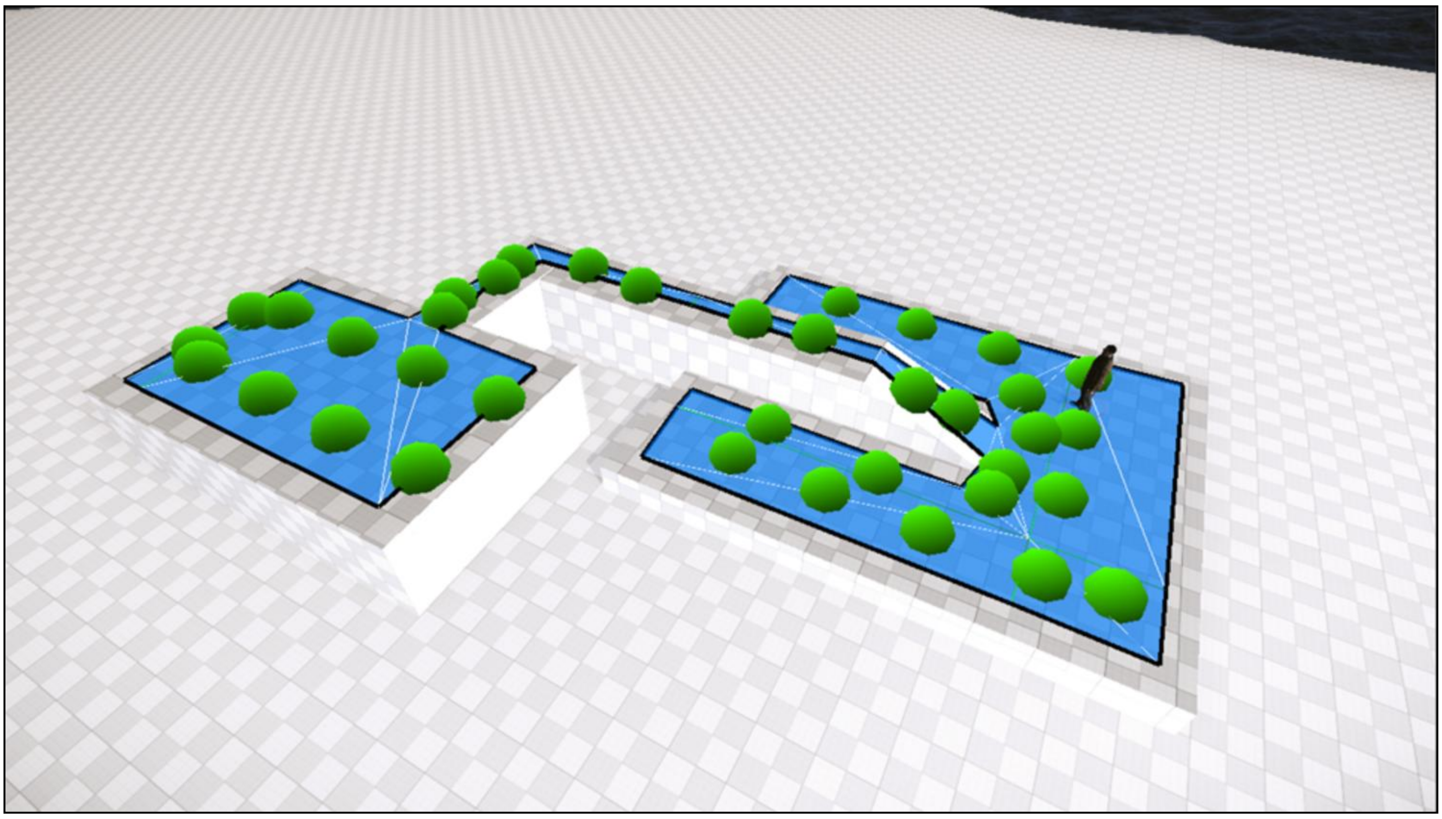


A navigation mesh in Lichdom.

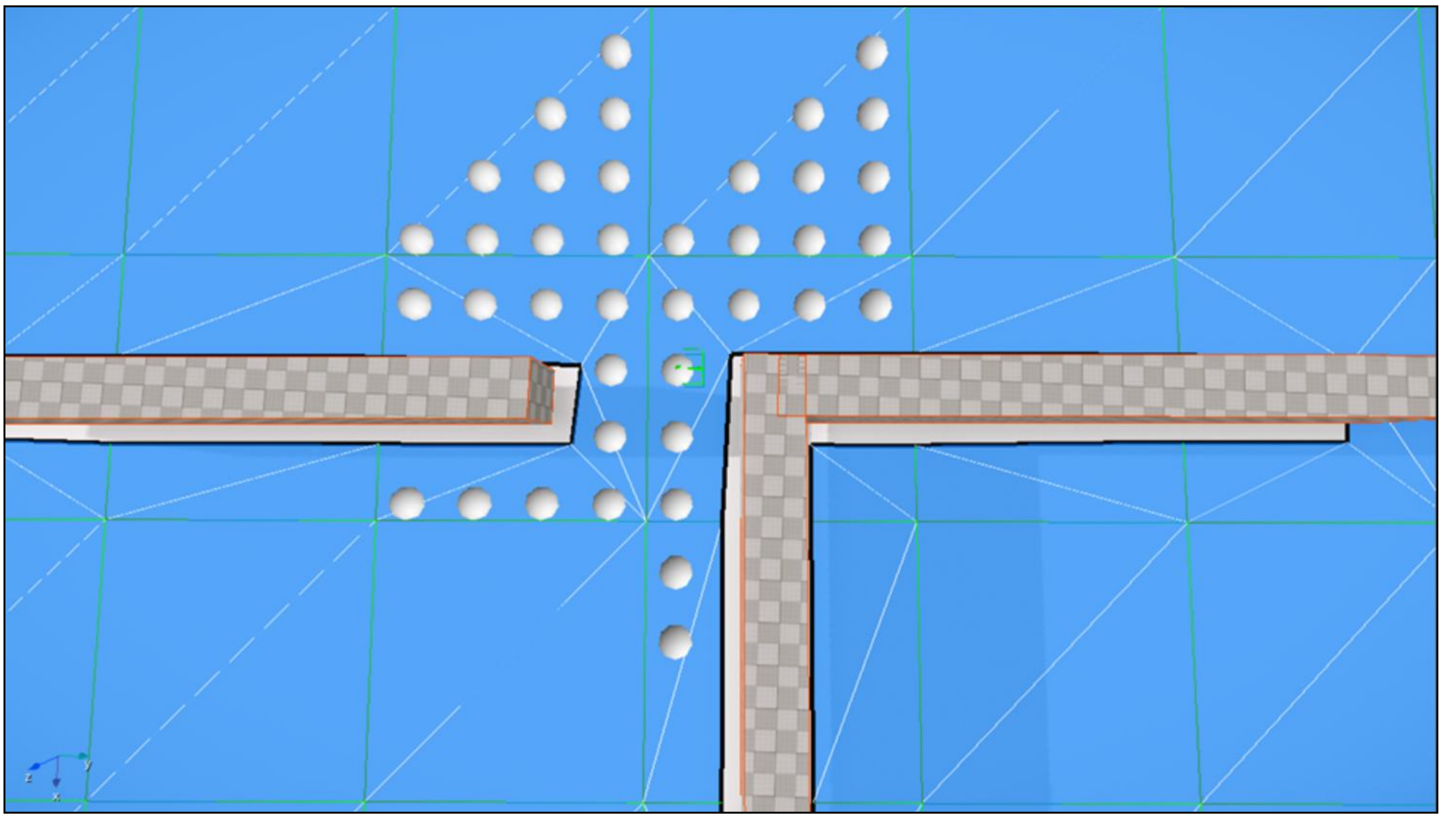


Same area, generating points for TPS selection based on that navmesh, centred on the far corner.

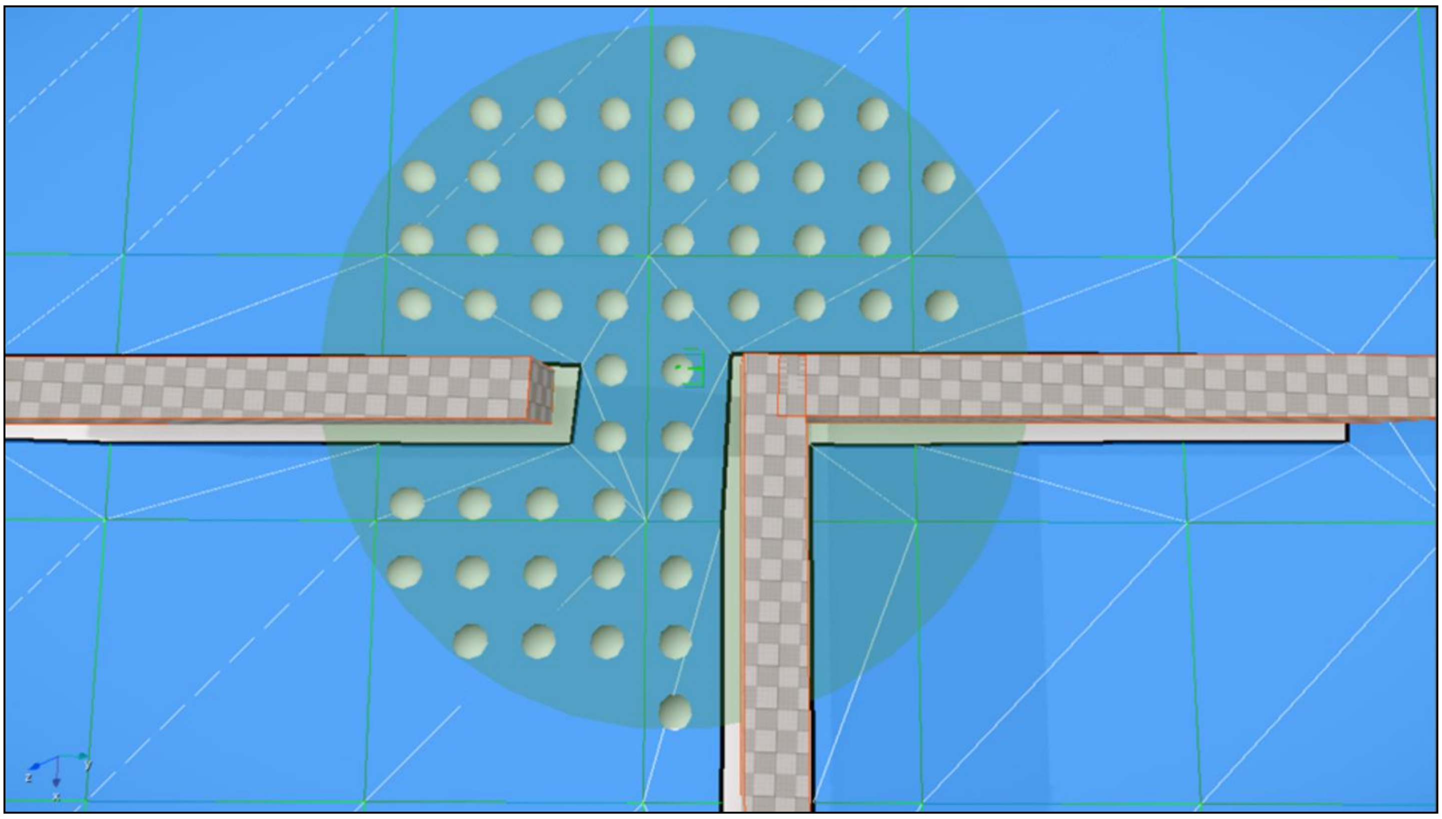




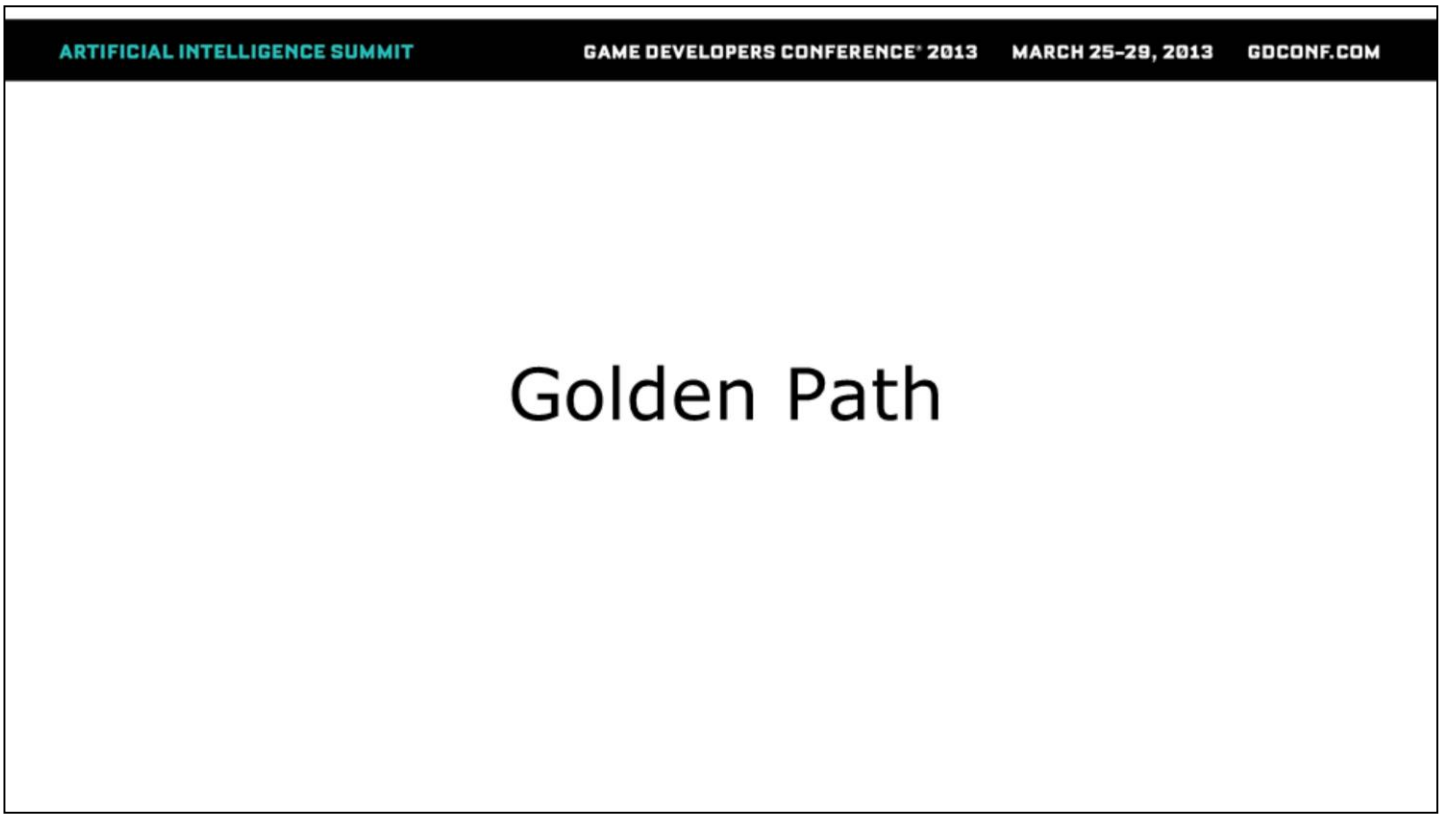
Adding in points for polys that received none to get a minimum even coverage.



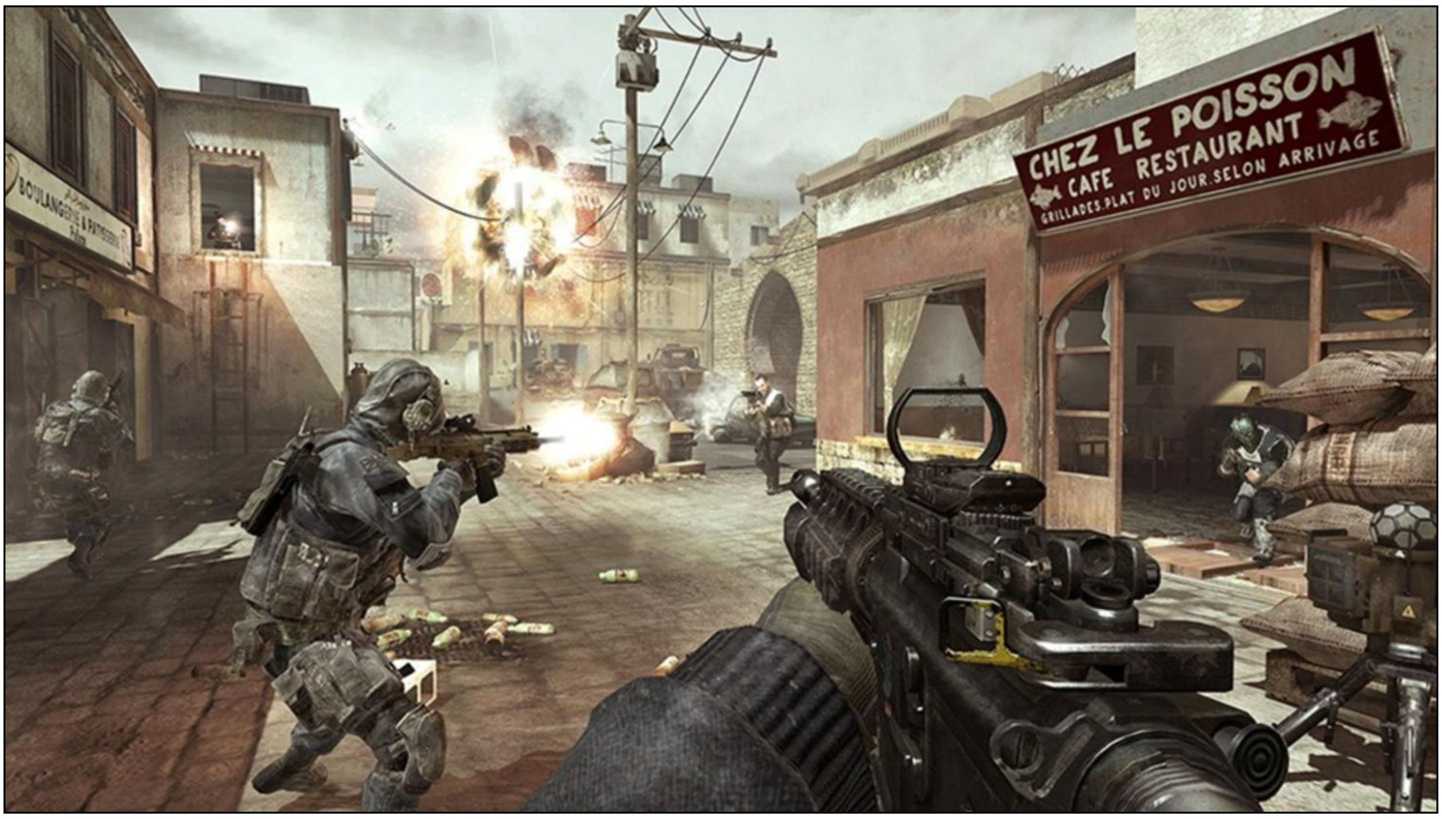
Inaccurate search distance due to pathfinding mesh poly size variation



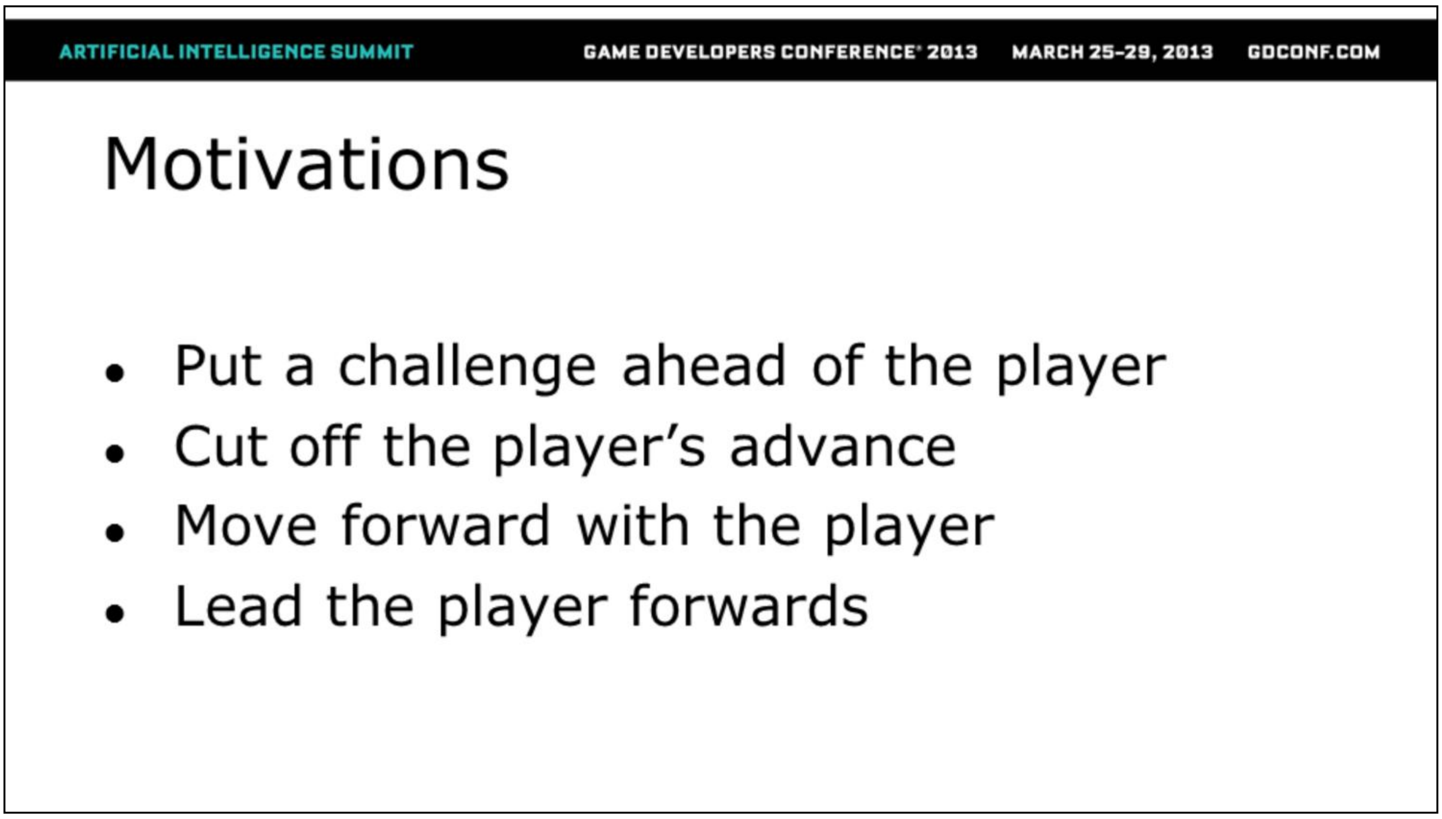
Our correction compromise.



Golden Path

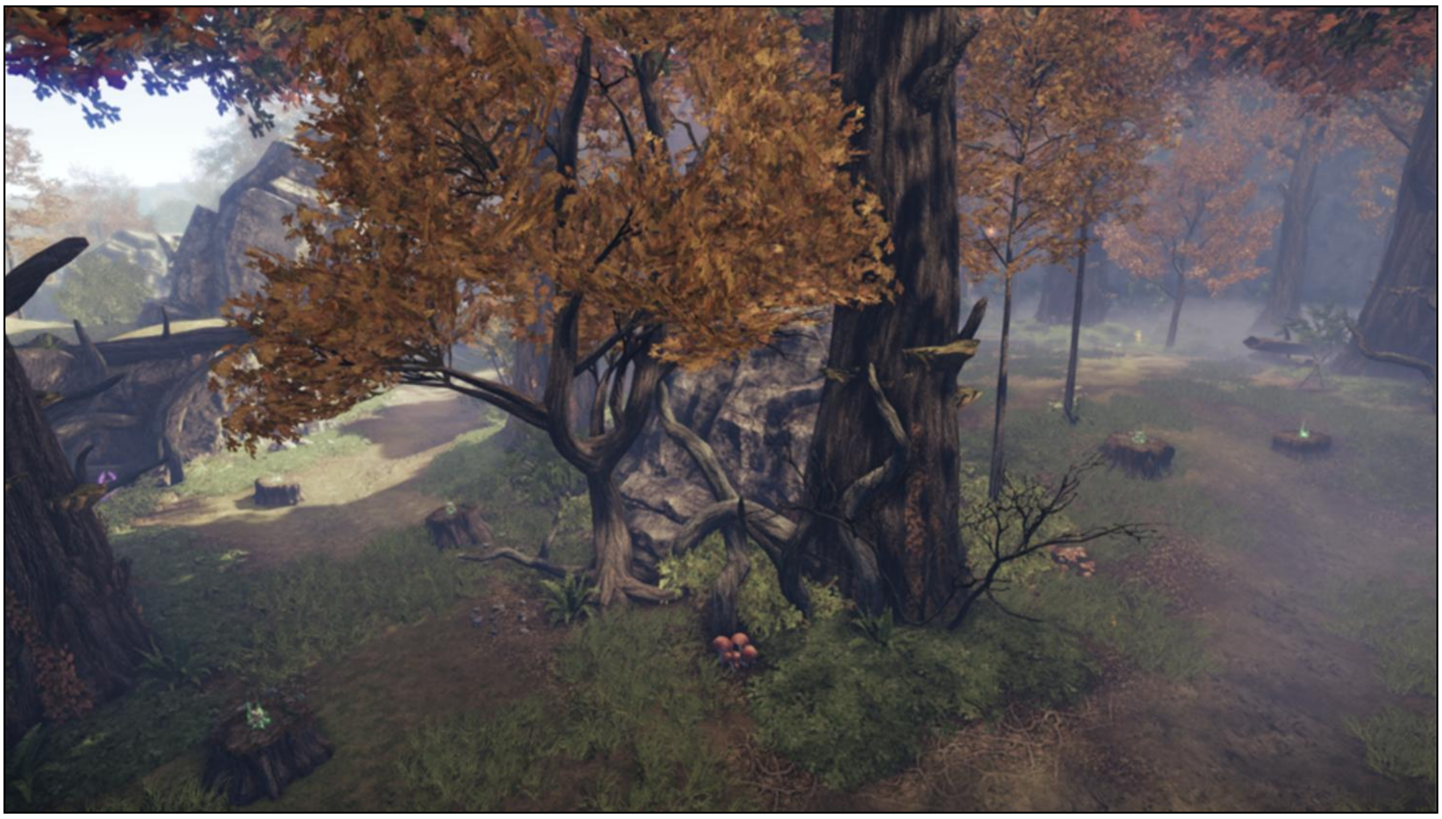


Shot from Modern Warfare 3

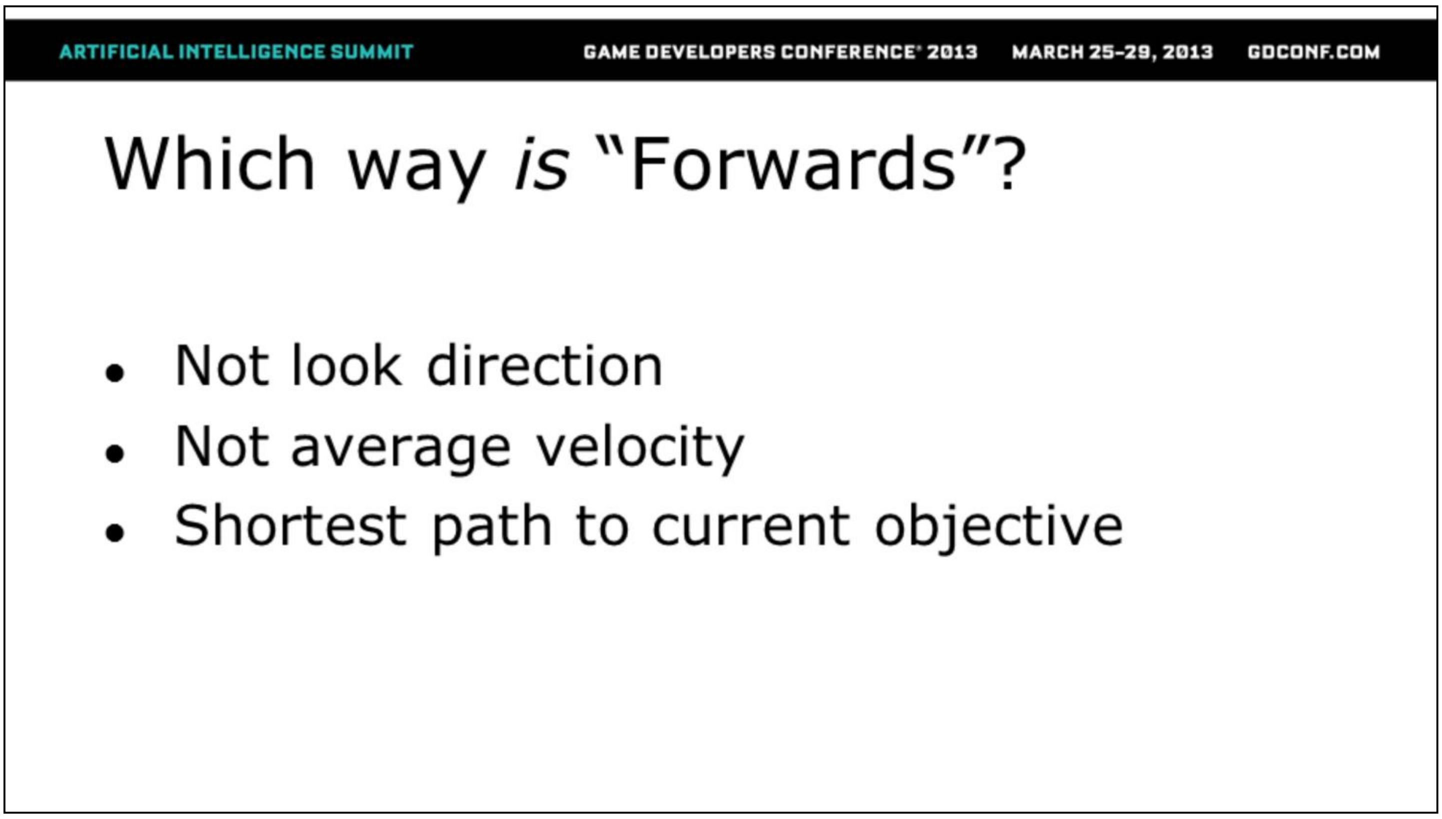


Motivations

- Put a challenge ahead of the player
- Cut off the player's advance
- Move forward with the player
- Lead the player forwards



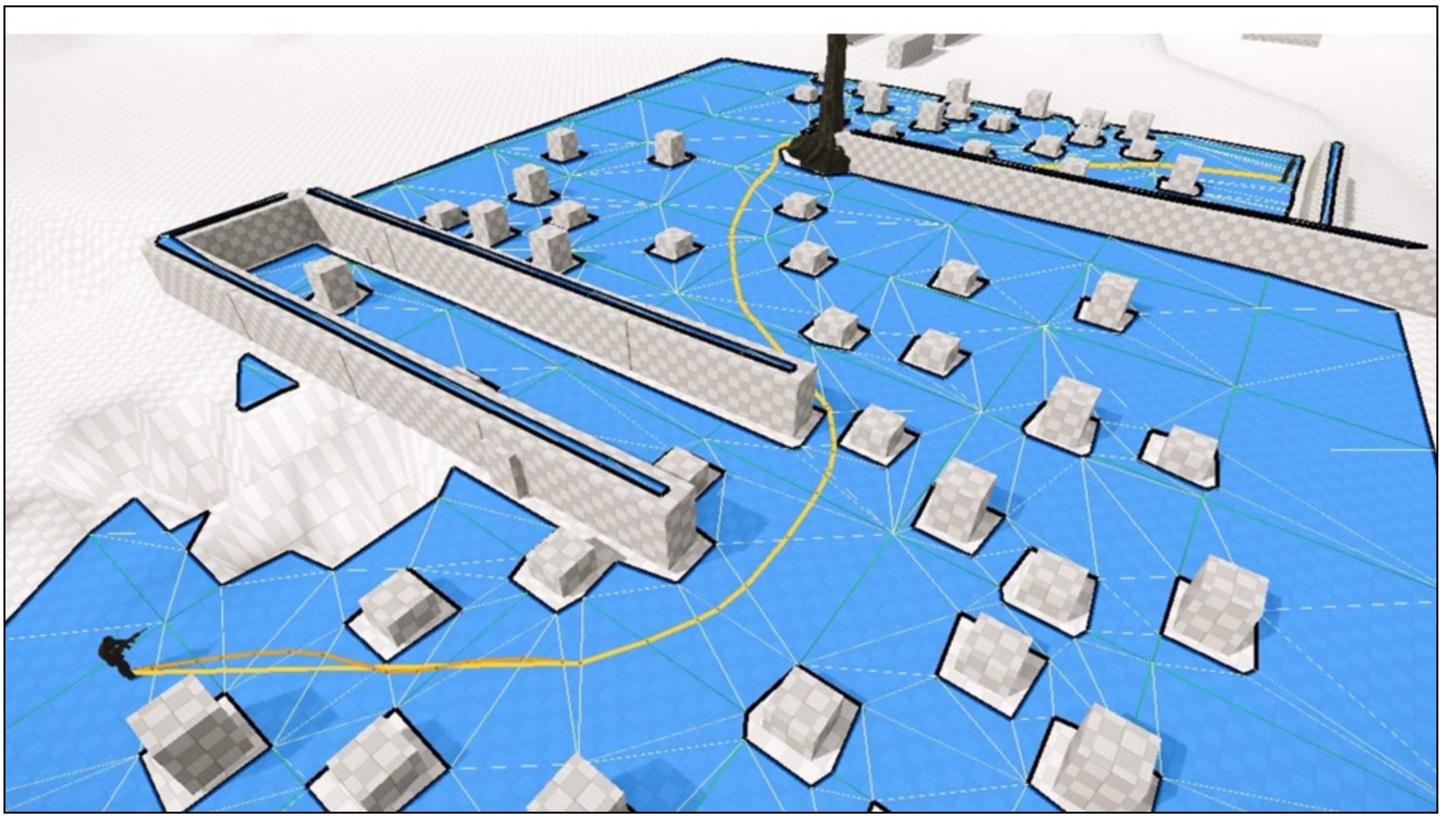
Shot from Lichdom. A circuitous path. Treestumps with green glow are “turrets”



Which way *is* “Forwards”?

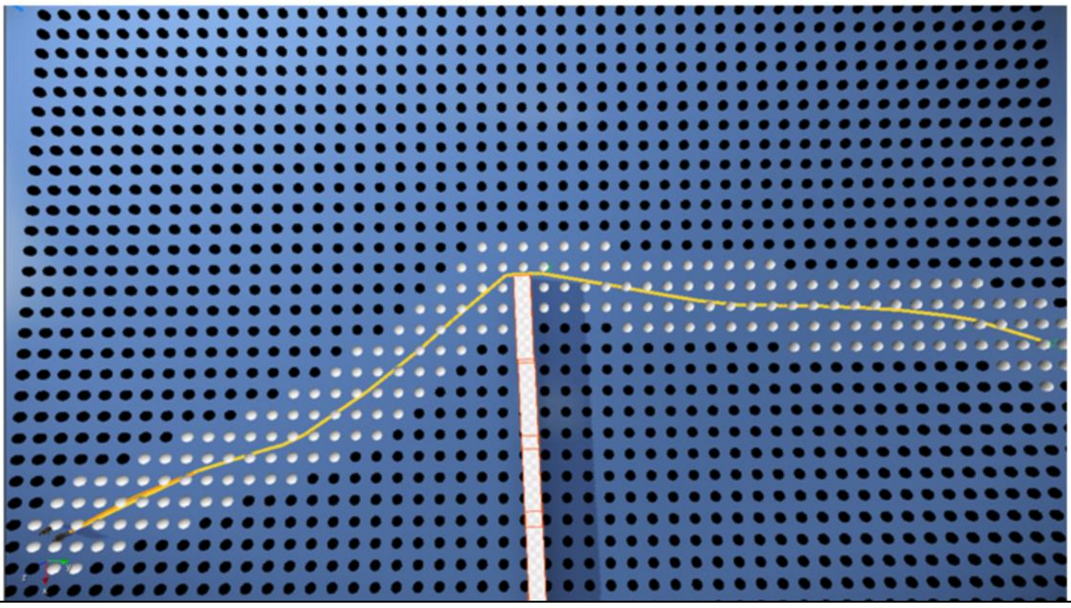
- Not look direction
- Not average velocity
- Shortest path to current objective

A Gameplay Compass

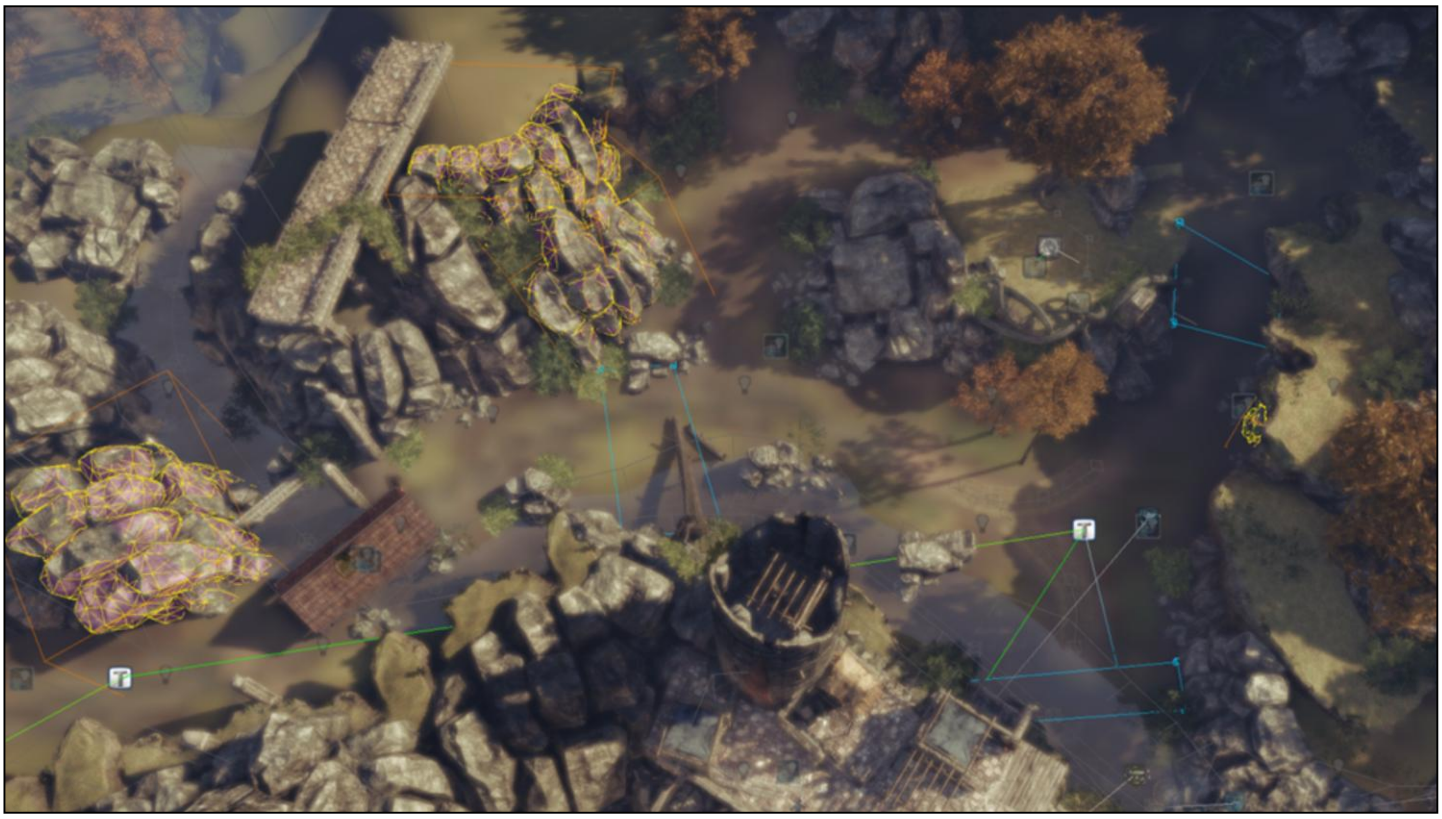


Show from CE3 Sandbox.

min distanceOff = 4

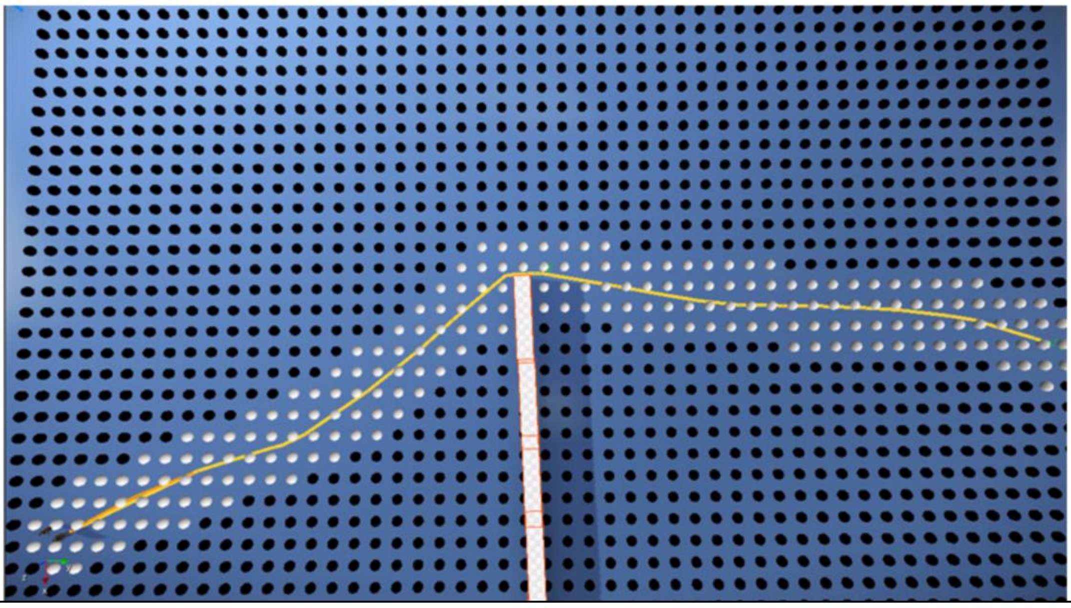


One way to use a golden path in your TPS queries

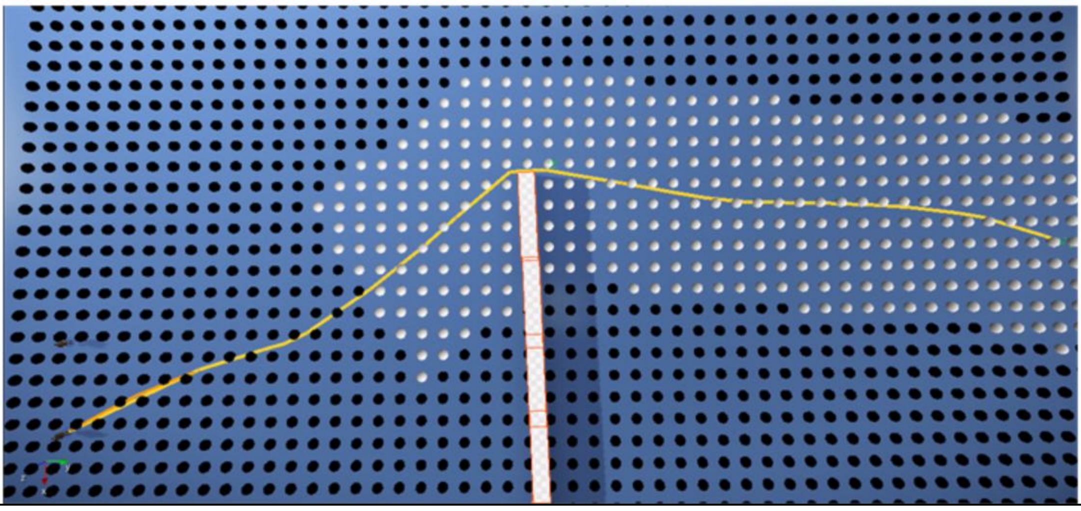


[redacted]

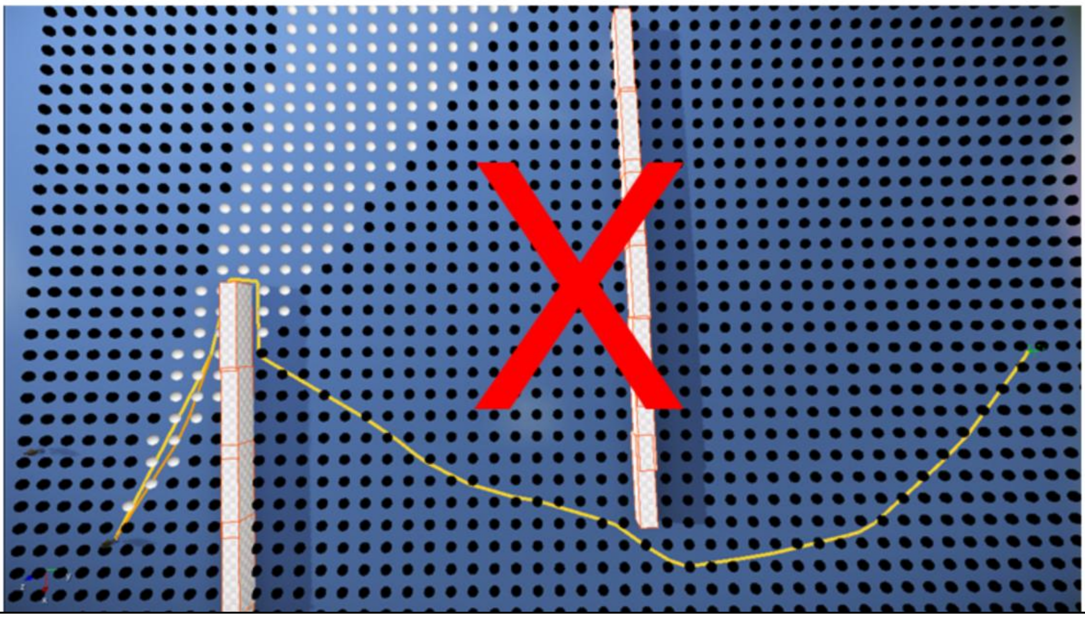
min distanceOff = 4



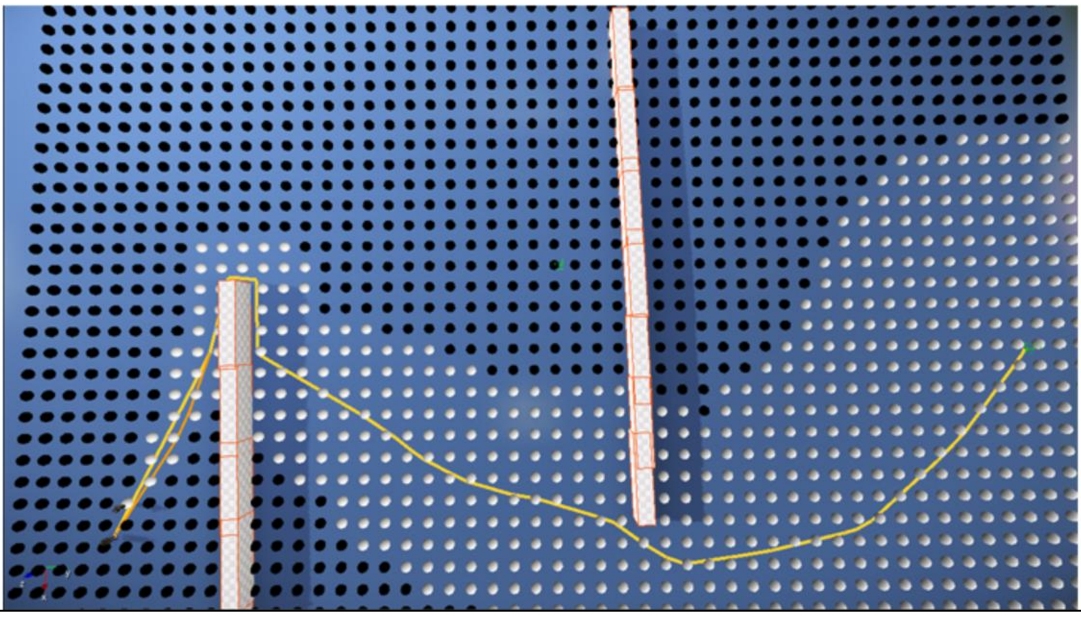
```
min distanceAlong = 20
max distanceOff = 8
```

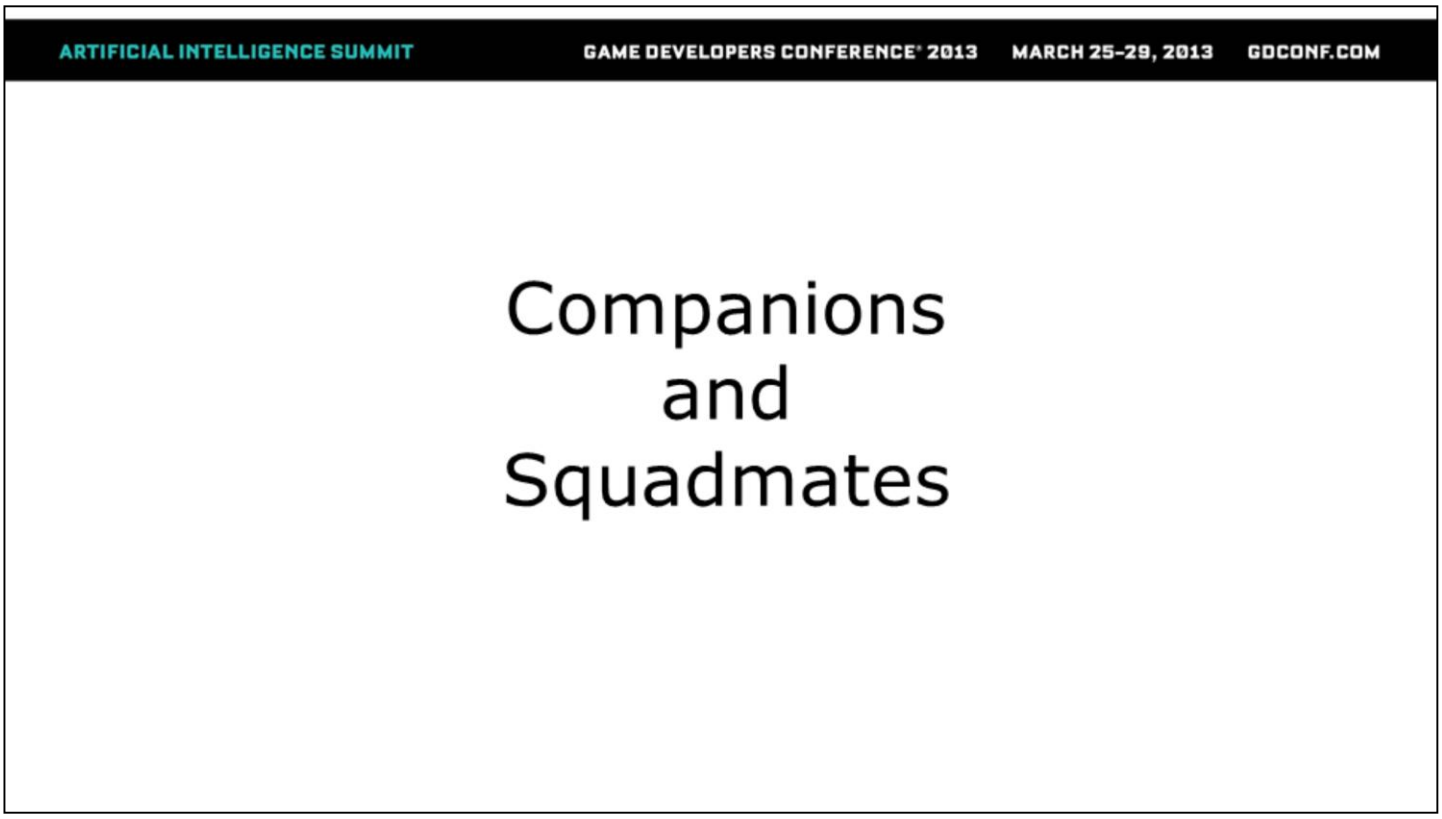


max goldenAngle = 10

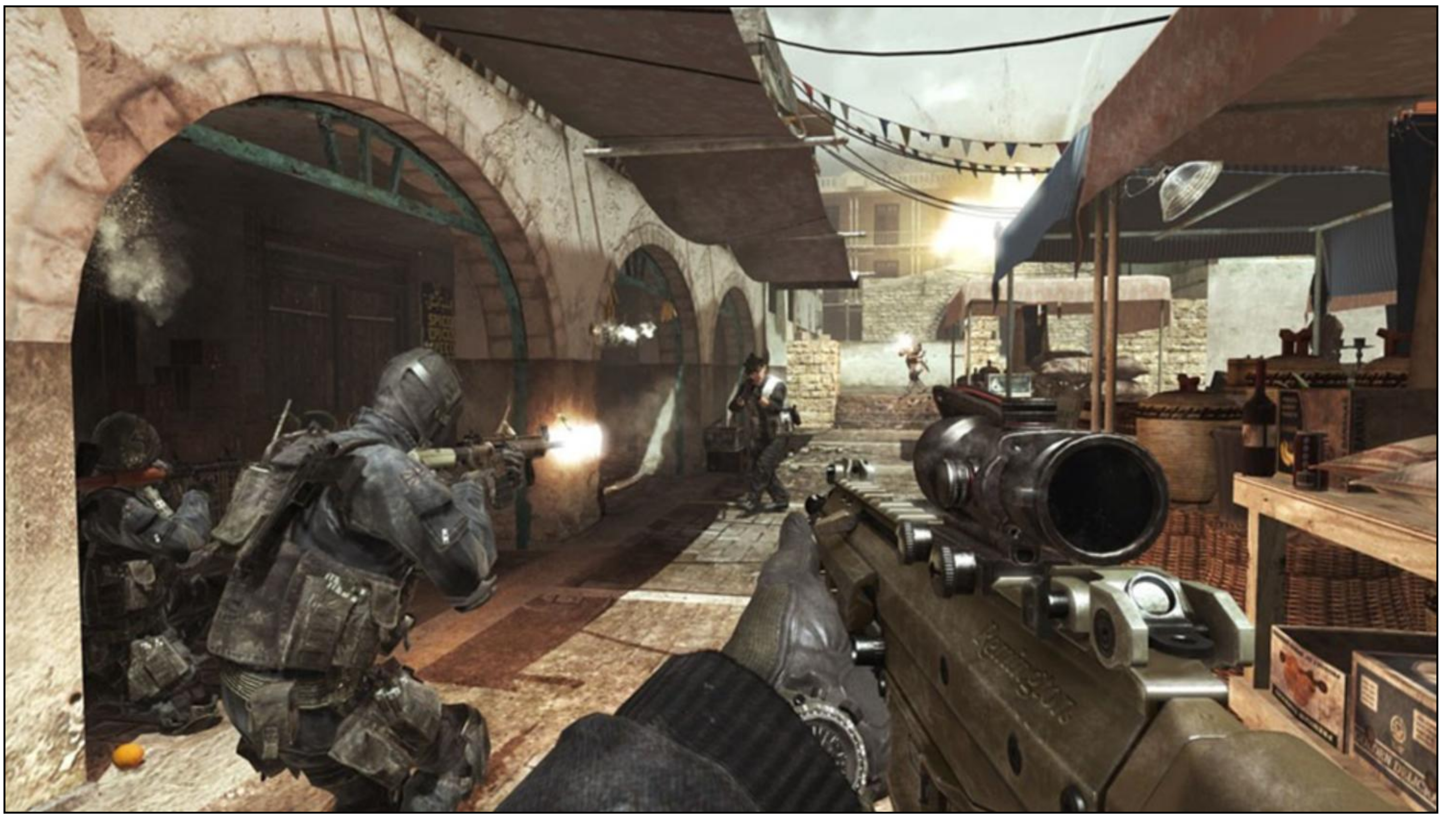


max funnelAngle = 10

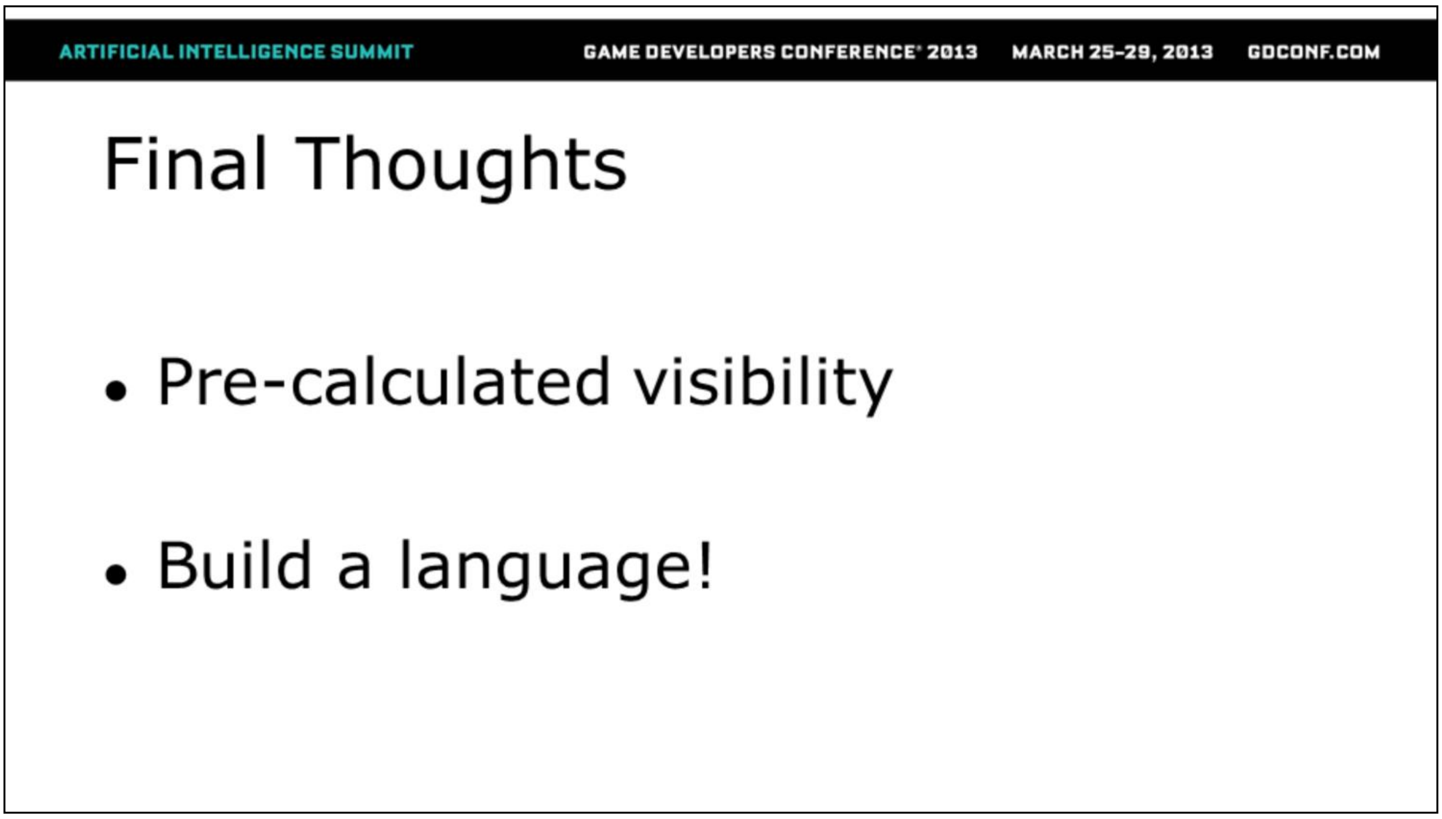




Companions and Squadmates

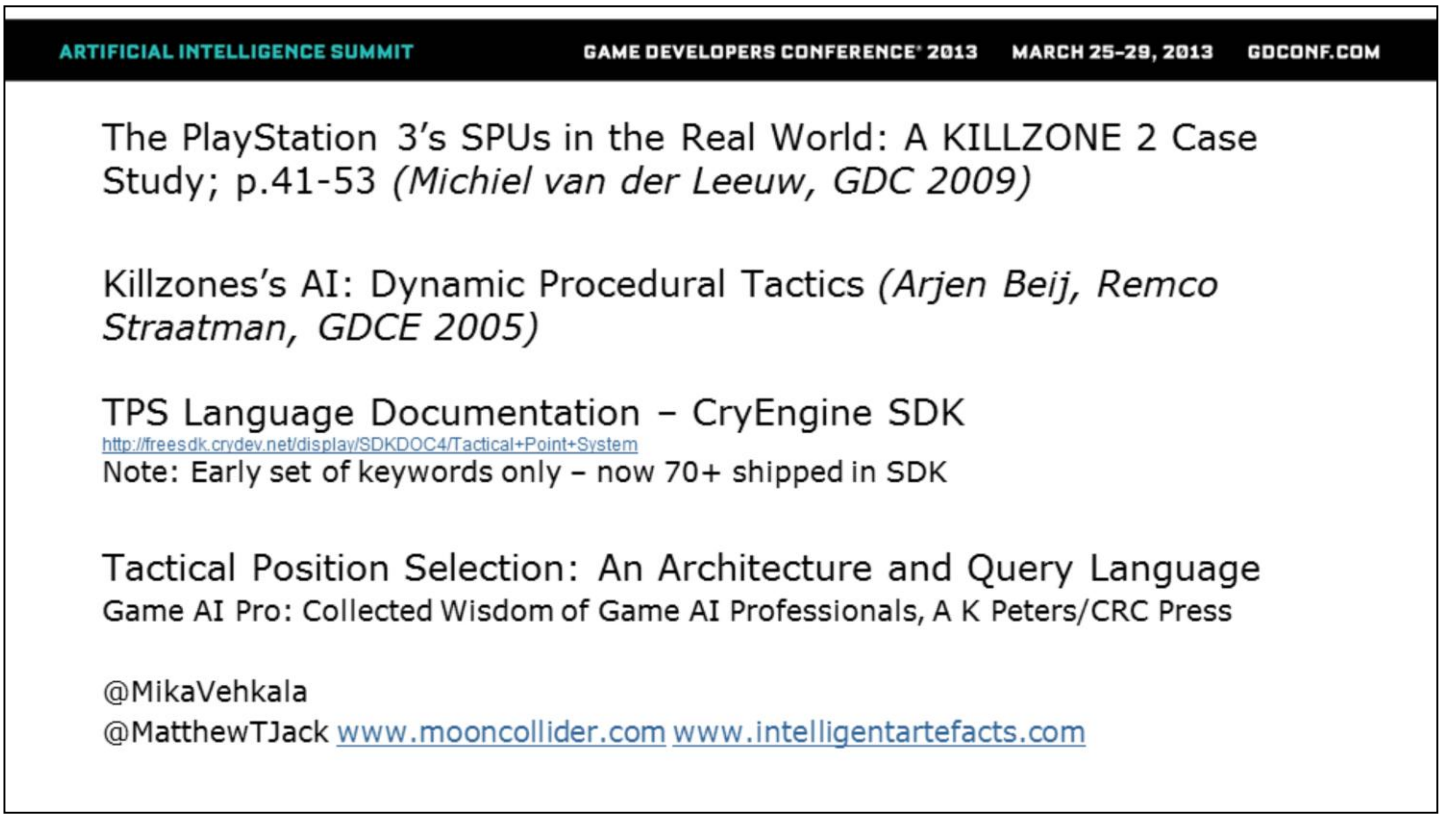


Modern Warfare 3. How could we make squadmates like this fully systematic?



Final Thoughts

- Pre-calculated visibility
- Build a language!



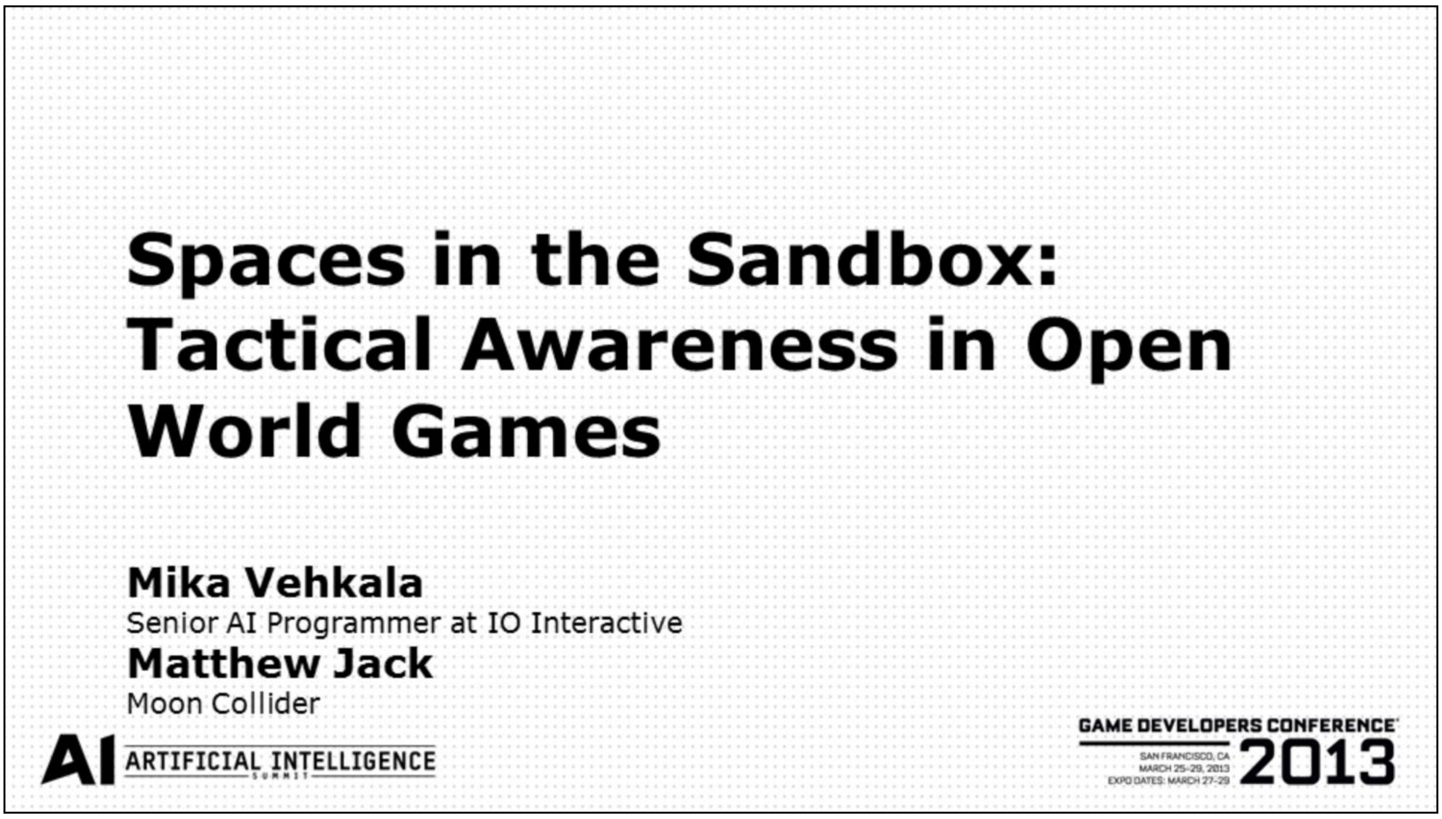
The PlayStation 3's SPUs in the Real World: A KILLZONE 2 Case Study; p.41-53 *(Michiel van der Leeuw, GDC 2009)*

Killzones's AI: Dynamic Procedural Tactics *(Arjen Beij, Remco Straatman, GDCE 2005)*

TPS Language Documentation – CryEngine SDK
http://freesdk.crydev.net/display/SDKDOC4/Tactical+Point+System
Note: Early set of keywords only – now 70+ shipped in SDK

Tactical Position Selection: An Architecture and Query Language
Game AI Pro: Collected Wisdom of Game AI Professionals, A K Peters/CRC Press

@MikaVehkala
@MatthewTJack www.mooncollider.com www.intelligentartefacts.com



Spaces in the Sandbox: Tactical Awareness in Open World Games

Mika Vehkala
Senior AI Programmer at IO Interactive
Matthew Jack
Moon Collider

AI ARTIFICIAL INTELLIGENCE SUMMIT

GAME DEVELOPERS CONFERENCE
SAN FRANCISCO, CA
MARCH 25-29, 2013
EXPO DATES: MARCH 27-29
2013