

## ゲームAI連続セミナー

### 第1回

# KillzoneにおけるNPCの動的な制御方法

三宅 陽一郎

y\_miyake@fromsoftware.co.jp

# Reference of figures

- [1] William van der Sterren (2001), "Terrain Reasoning for 3D Action Games",  
[http://www.cgf-ai.com/docs/gdc2001\\_paper.pdf](http://www.cgf-ai.com/docs/gdc2001_paper.pdf)
- [2] William van der Sterren (2001) , "Terrain Reasoning for 3D Action Games(GDC2001 PPT)", [http://www.cgf-ai.com/docs/gdc2001\\_slides.pdf](http://www.cgf-ai.com/docs/gdc2001_slides.pdf)
- [3] Remco Straatman, Arjen Beij, William van der Sterren (2005) ,"Killzone's AI : Dynamic Procedural Combat Tactics", [http://www.cgf-ai.com/docs/straatman\\_remco\\_killzone\\_ai.pdf](http://www.cgf-ai.com/docs/straatman_remco_killzone_ai.pdf)
- [4] Arjen Beij, William van der Sterren (2005),"Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)", [http://www.cgf-ai.com/docs/killzone\\_ai\\_gdc2005\\_slides.pdf](http://www.cgf-ai.com/docs/killzone_ai_gdc2005_slides.pdf)
- [5] Damian Isla (2005), "Dude, where's my Warthog? From Pathfinding to General Spatial Competence", <http://www.aiide.org/aiide2005/talks/isla.ppt>
- [6] Chris Butcher,Jaime Griesemer (2002),"The Illusion of Intelligence The Integration of AI and Level Design in Halo",  
[http://www.gamasutra.com/features/gdcarchive/2002/jaime\\_griesemer.ppt](http://www.gamasutra.com/features/gdcarchive/2002/jaime_griesemer.ppt)
- [7] 三宅 陽一郎 (2006) , “クロムハウズにおける人工知能開発から見るゲームAIの展望, CEDEC2006 PPT

# 本講演の構成

## はじめに

### ゲームAI連続セミナーのご紹介

#### 第1部 ゲームAI 技術解説 (60分)

第1章 技術テーマとゲーム説明 (15分)

第2章 静的位置検出 (Killzone の先行研究) (15分)

第3章 動的位置検出 (Killzone's AI) (20分)

第4章 発展 (10分)

各章の終わりに質疑応答を行います。

#### 第2部 ディスカッション (65分)

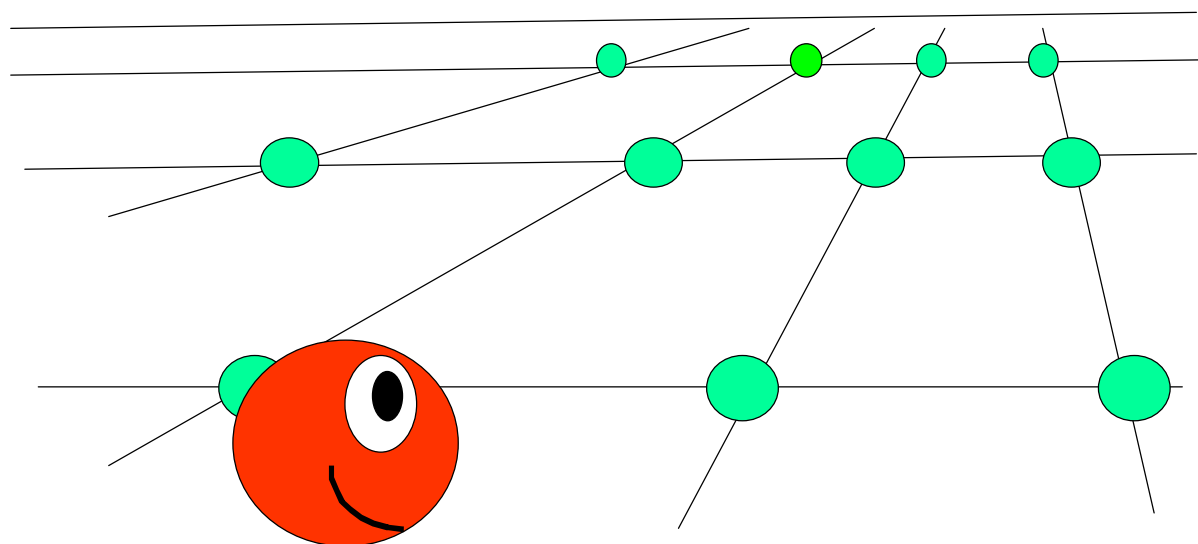
(AI) 一つのテーマについて15～20分を予定。

テーマはこちらが準備したもの、参加者からテーマを募集(60分)

(セミナー) 今日の感想とセミナーの改善点を討論する(5分)

はじめに

# ゲームAI連続セミナー全体のご紹介

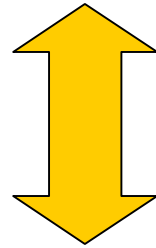


# 目的

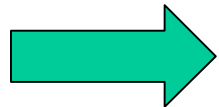
- (1) ゲームAIの技術について紹介、発表できる場を作る。
- (2) ゲームAIに関して討論を行う  
オープンな相互インタラクションのできる場を用意する。
- (3) 参加者の意見を取り入れて、  
ゲーム業界で本当に必要なセミナーの形を実現する。  
(運営やスピーカーなど常時募集する)

# ゲームAIの特徴

- (1) 一つのゲームにAIの技術を組み込んで行くことで、その技術が潜在的に持っていた力を発見、発展することが出来る。

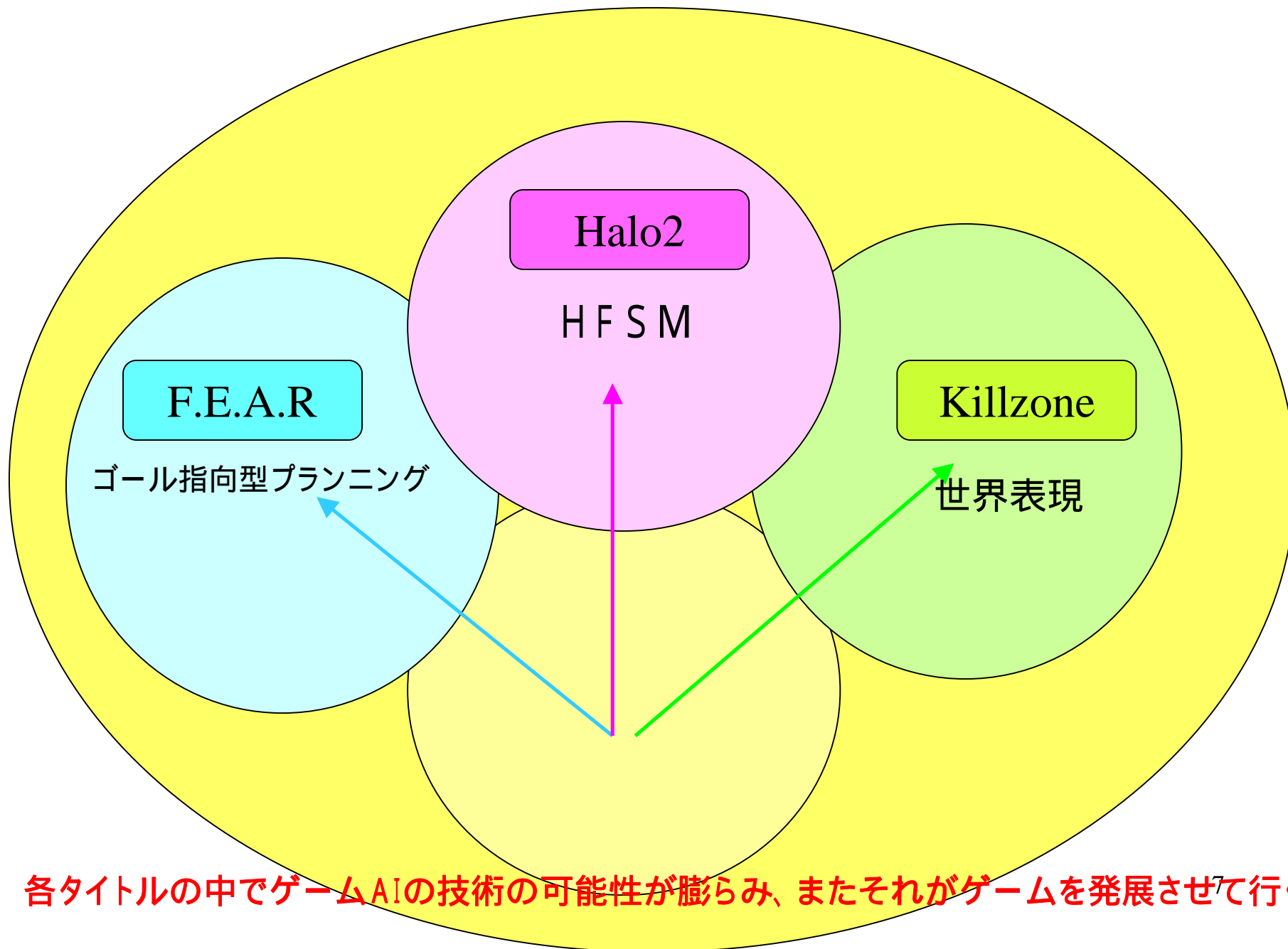


- (2) 一つのAIの技術を追求して行くことで、新しいゲームデザインが見えて来る



ゲームとゲームAI は、  
お互いを進化させる力を持つ

# ゲームとゲームAIの共進化



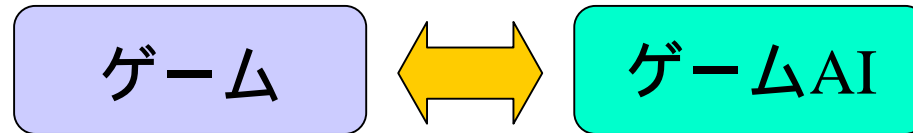
# 欧米と日本におけるゲームAIの開発の比較

	欧米	日本
情報公開	論文、会議で発表	不明
文献	論文や書籍の膨大な蓄積	とても少ない
アカデミズムとの関係	大学でゲームAIを研究。 大学から開発の現場へ参加。	希少
取り組むAIの分野	FPSを中心とするキャラクターメインのゲーム	さまざまなゲーム
業界としての取り組み	標準化への流れがある (IGDA AIISC(人工知能インタフェース標準化委員会))  (AIのミドルウェア市場の形成)	標準化の流れに参加する形 (IGDA JAPAN を通して紹介がされている。CEDECでも RoundTableがある)

(例) Quakeを扱った修士論文は世界的に有名になった。



# ゲームAIにおける日本のチャンス



- (1) ゲームAIは、ゲームのコンテンツに対応して形成される。
- (2) FPS、スポーツゲームだけでなく、いろいろな種類のゲームを作る日本には、いろいろな種類のAIを作れるチャンスがある。 ➡ 技術的チャレンジ
- (3) 日本は、ゲームAIの発展に欧米とは違う方向からも貢献できるはずだ。

➡ そのための発端となる場所が欲しい...

実際のゲーム開発へ活用する



これからのゲーム開発におけるゲームAIの  
可能性を知る

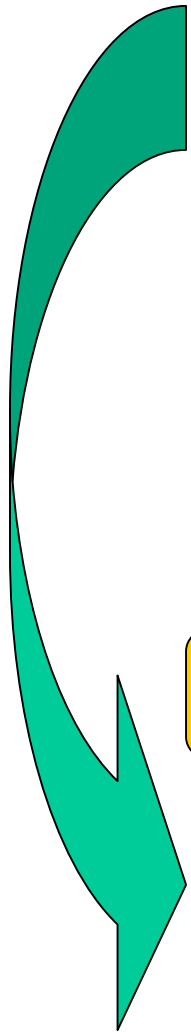


IGDA  
ゲームAI連続セミナー

開発者が一同に介して、  
ゲームAIについて学び、討論する場

よい循環を形成する

日本におけるゲームAIのセミナー・カンファレンスの場



# セミナーの現在のプラン

2006年9月



12月  
第1回

2月  
第2回

2ヶ月に1回実施

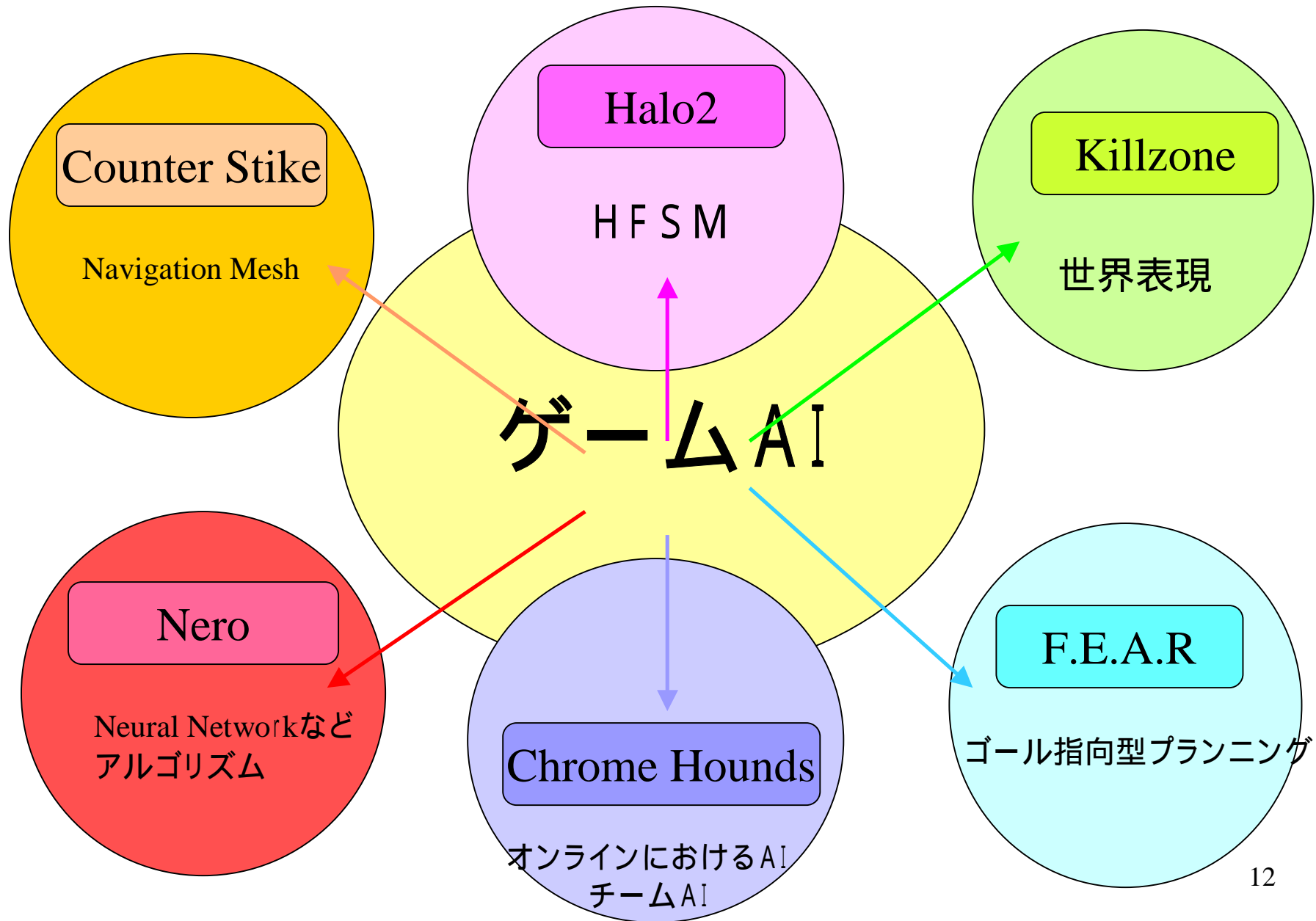
CEDEC2007  
まとめ

全6回(継続を目指す)  
会場は大学、企業のホールなど



目標へ進みつつ、参加者の意図を汲みながら  
徐々に改善して行く

# これからのセミナーの展開



# ゲームAI連続セミナー

(1) 参加者が作るセミナー

意見をフィードバック

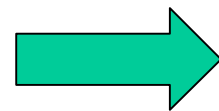
話す側にもなる

議論にできるだけ参加

(2) 参加者が望むものを自分で持ち帰るセミナー

(3) 開発者と出会えるセミナー

学生、開発者、研究者のインタラクションの場



ゲーム開発へ

# 質疑応答

# 第1部 ゲームAI 技術解説(60分)

1. 技術テーマとゲーム説明 (15分)
2. 静的位置検出 (Killzone の先行研究) (15分)
3. 動的位置検出 (Killzone's AI) (20分)
4. 発展 (5分)

各章の終わりに質疑応答を行います。

# 第1章

## 技術テーマとゲーム説明

### ◆ 技術テーマ「世界表現」の説明

#### ◆ Killzoneの説明



# AIの二つのアプローチ

AI に如何に世界を認識させるか？

Neural Network  
自己組織化マップ  
画像認識

AIが自分で世界を  
解釈する

難しい



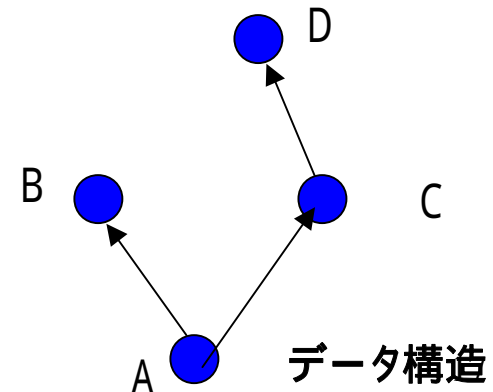
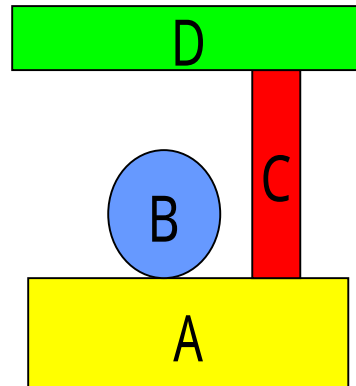
?

知識表現

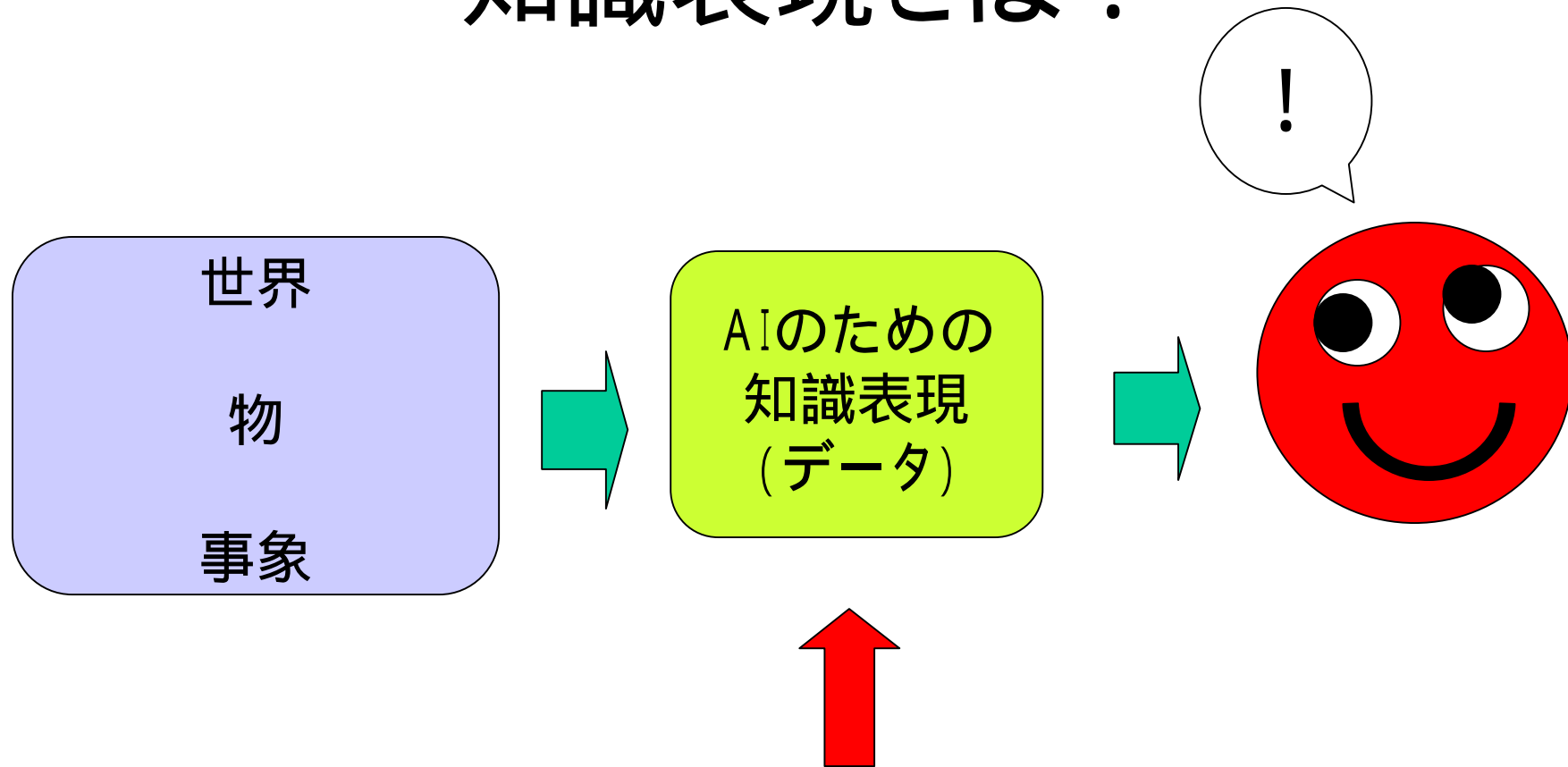
(knowledge representation)

人が世界を  
解釈する形式、ヒント、  
或いは関係そのものを  
与えてやる

手間はかかるが確実



# 知識表現とは？

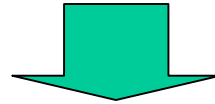


人間が準備してあげる

# よい“AI”を作るには？

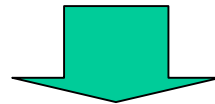
AIに自分で、  
世界を「解釈」させることは難しい。

よいAI = ゲームにフィットしたAI



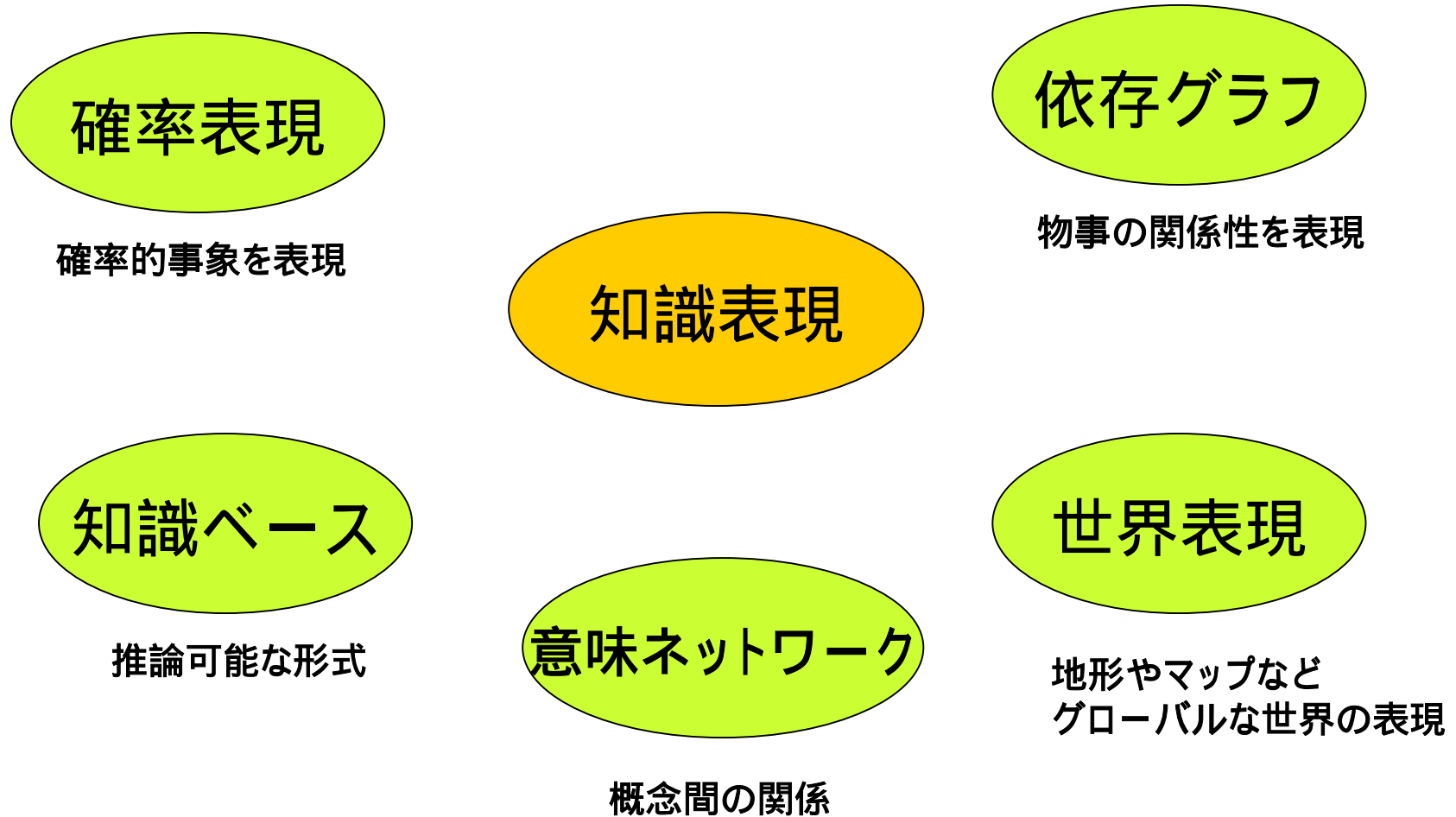
ゲーム世界の諸関係をAIの目的に応じて適切に表現してやる必要がある。

**知識表現(knowledge representation)**



その表現されたデータの上に思考を築く。

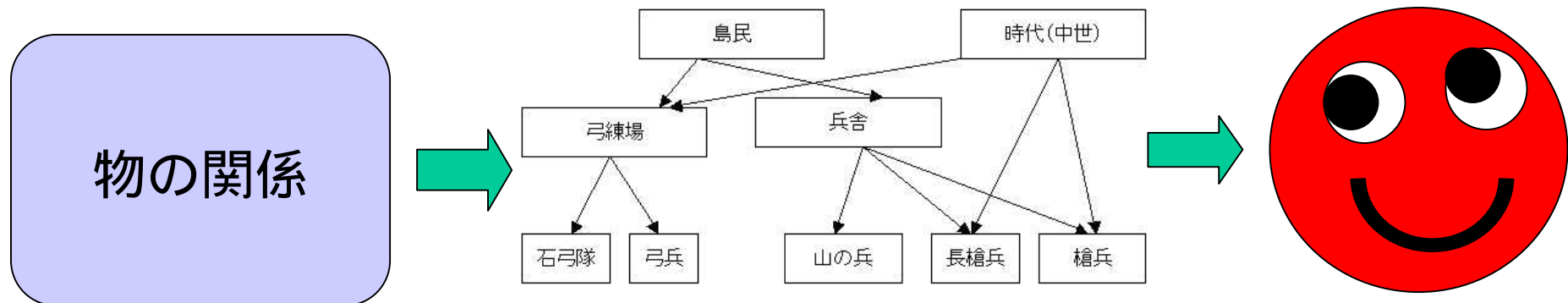
# ゲームに応用可能なおける知識表現の分野



# ゲームに応用可能なおける知識表現の分野

## 依存グラフ

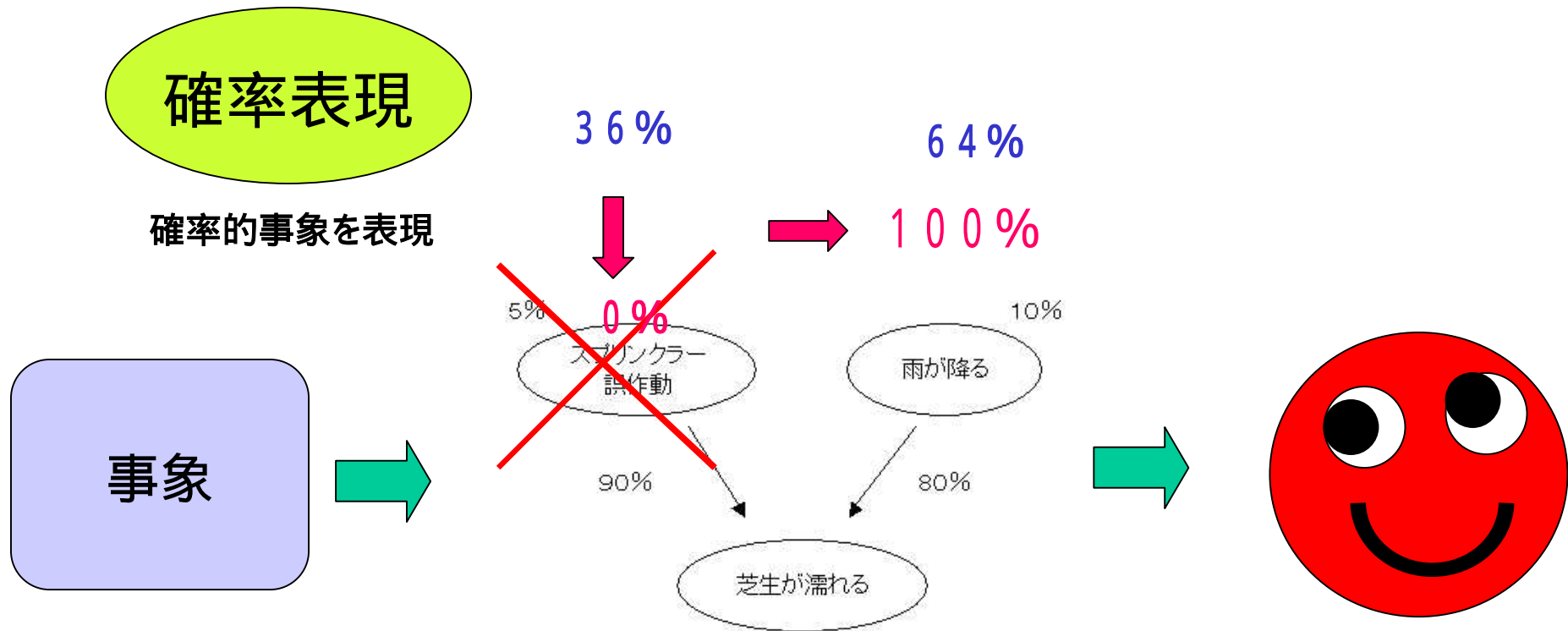
物事の関係性を表現



「あるものを作るには、あるものが必要である」という関係性

 シミュレーションゲーム

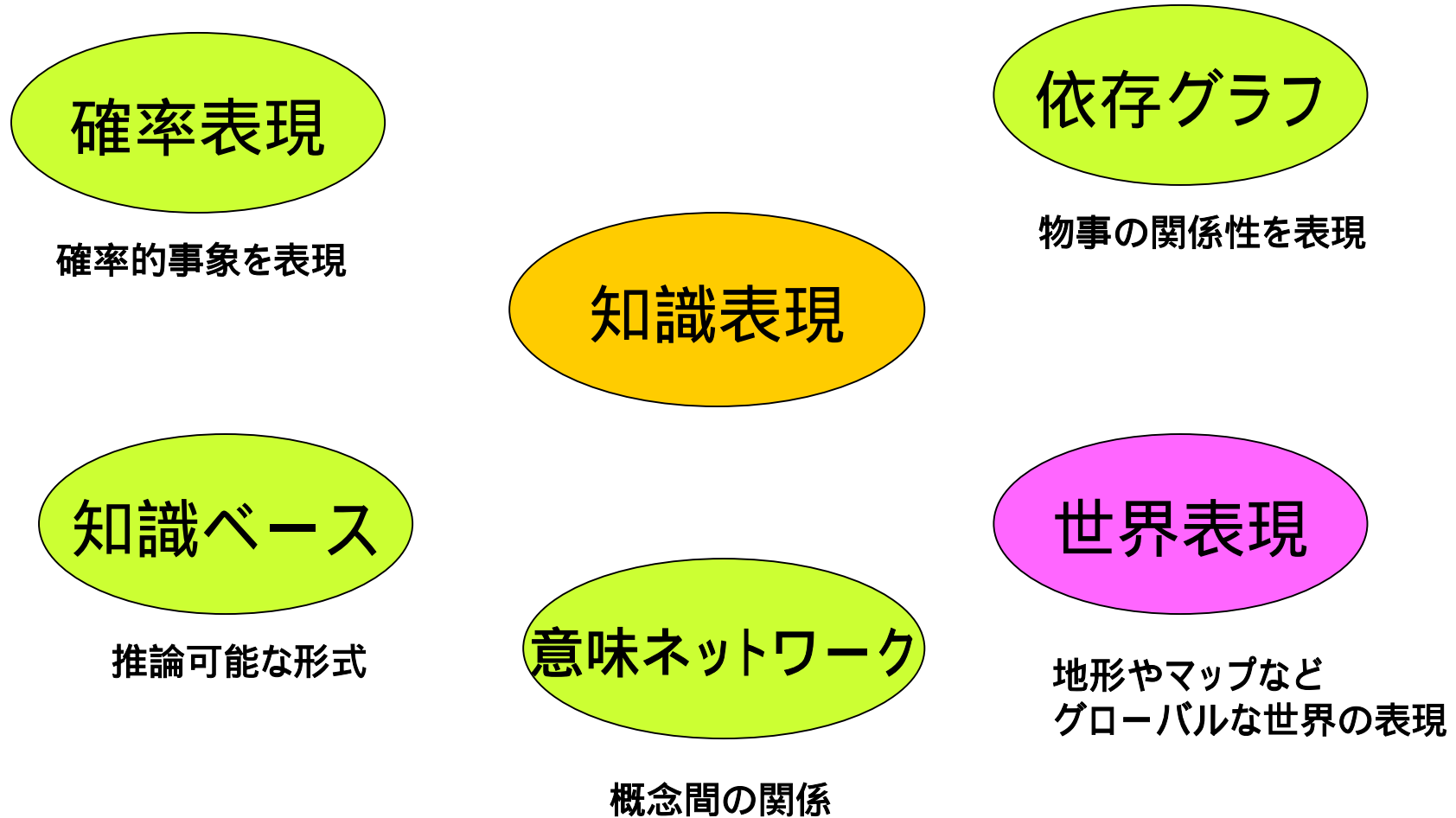
# ゲームに応用可能なおける知識表現の分野



芝生が濡れているのをCOMが発見した場合、  
COMはその原因を確率的に推論する。

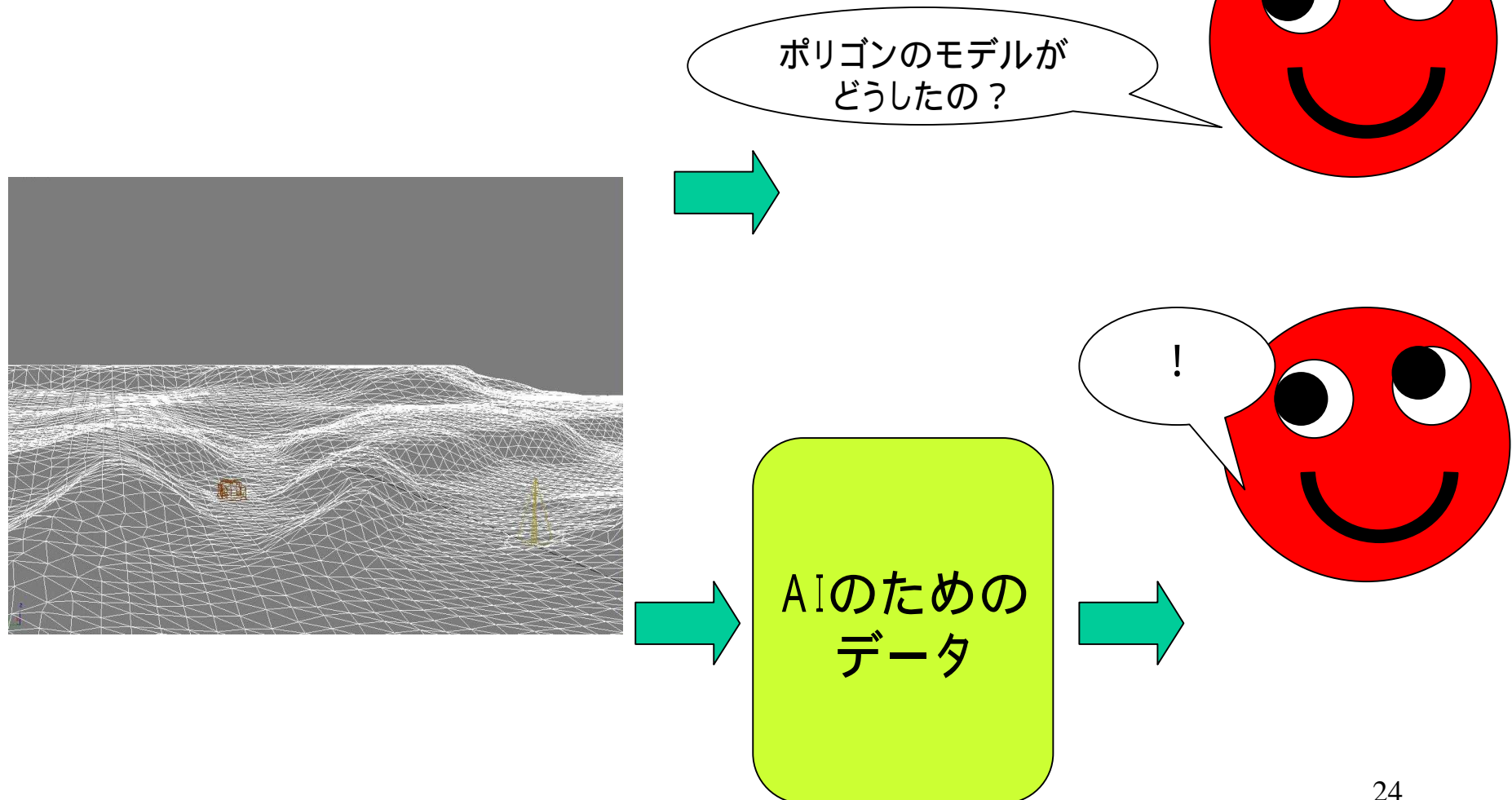
➡ 知っているか知らないかよりも、人間らしい推論

# ゲームに応用可能なおける知識表現の分野



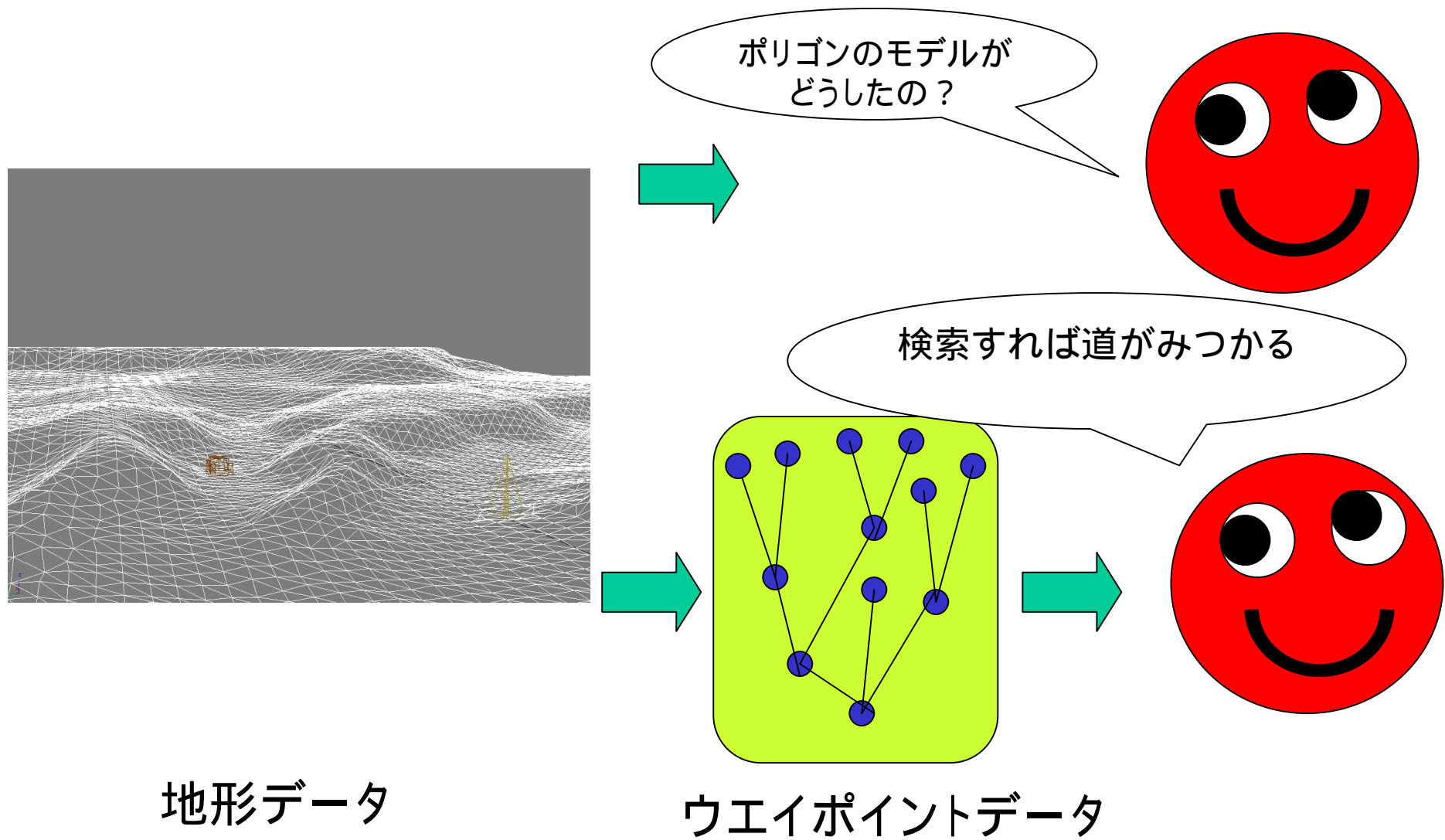
# 世界表現とは？

AIの属する世界の大局的な情報の知識表現



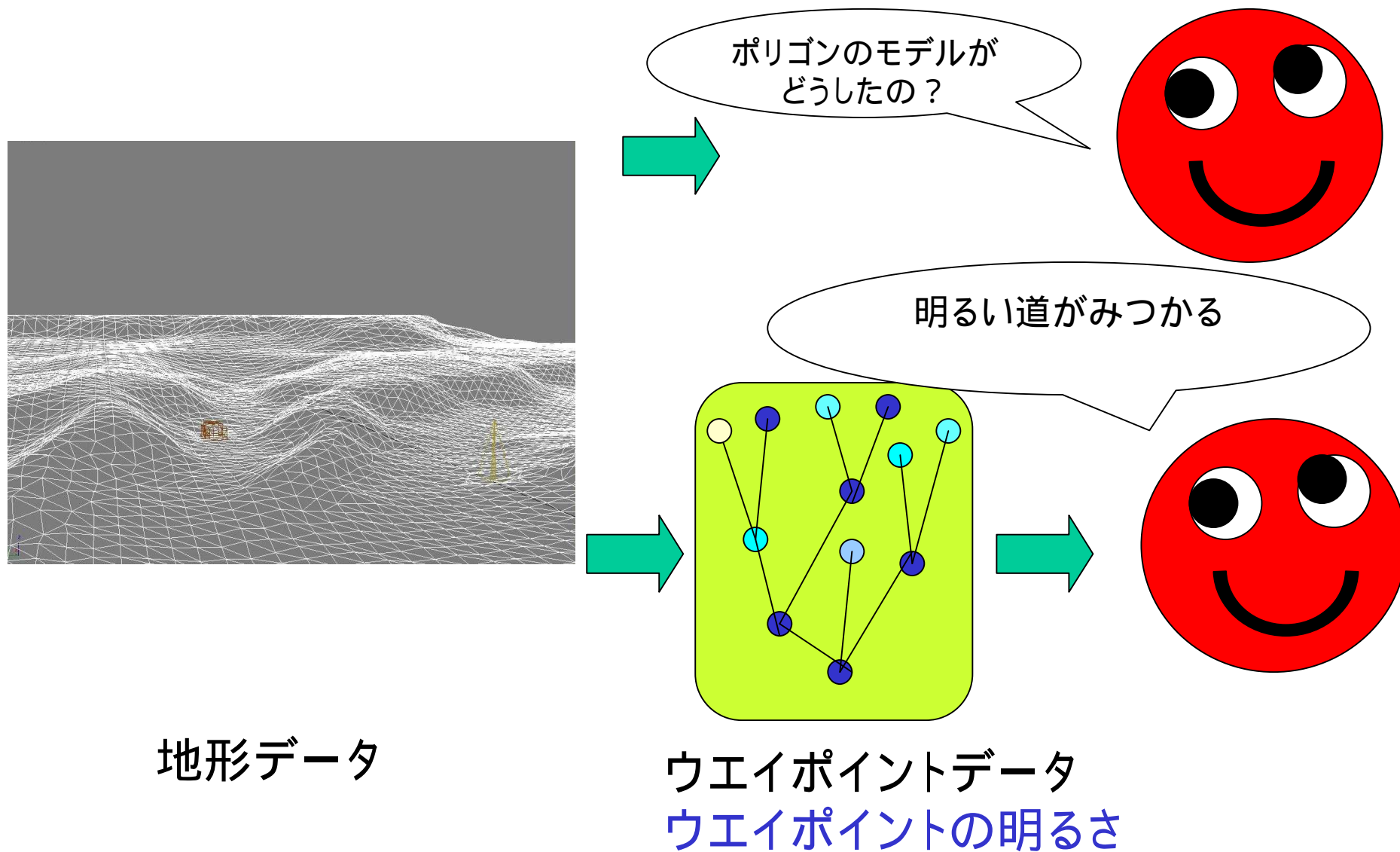


# 世界表現の例 パス検索データ

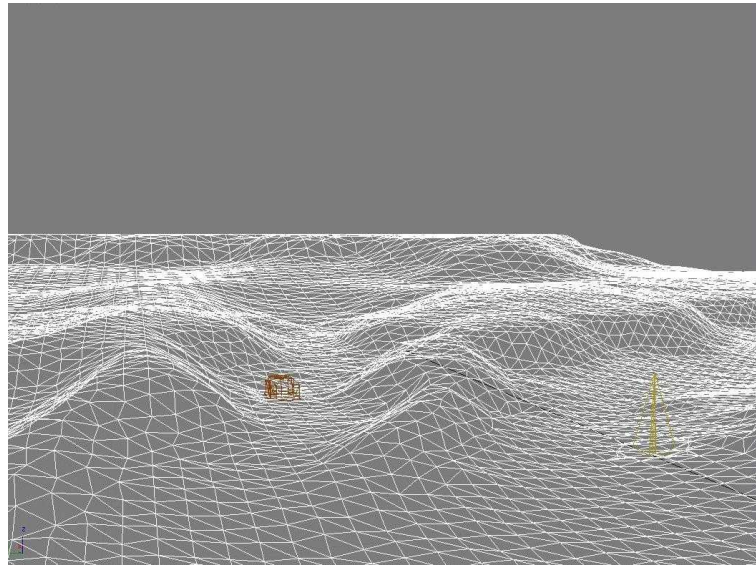


# 世界表現の例

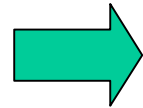
## パス検索データ + ウェイポイントの明るさ



# 世界表現の例 パス検索データ + ウェイポイントの明るさ + 東側に対して視線が通らない(見通しが悪い)



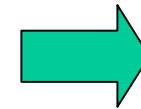
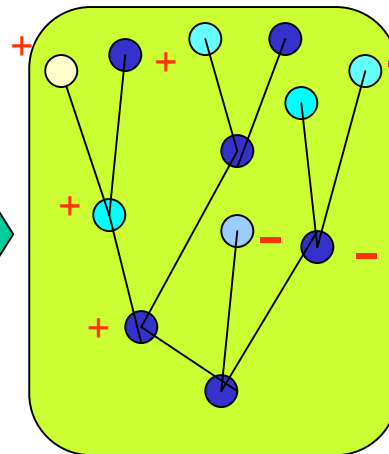
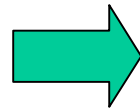
地形データ



ポリゴンのモデルが  
どうしたの？



敵が東側から来るので、  
東側から見えにくくて暗い道を通ろう



ウェイポイントデータ  
ウェイポイントの明るさ  
東側から見えやすいかどうか( <sup>27</sup> +、- )

# 今日のテーマ

テーマ

世界表現を精緻にすればするほど、  
AIができることは増え、また高度になる

どう工夫すればよいのだろうか？

- (1) 自分で考える。
  - (2) 他の人がどうやっているかを研究する。
  - (3) 他の開発者と議論する。
- ⇒ セミナー前半  
⇒ セミナー後半

# 第1章

## 技術テーマとゲーム説明

- ◆ 技術テーマ「世界表現」の説明

- ◆ Killzoneの説明

# Killzoneとは



2004-2005 (ミリオンセラー)

販売 SCEE (欧州)

SEGA (日本)

開発元 Guerrilla

FPS

Online (日本版にはない)

# Killzoneの特徴( 1 )

## 一人称視点のシューティングゲーム( F P S )

Killzone 紹介画像(省略、以下参照)

- <http://sega.jp/ps2/killzone/>
- <http://www.killzoneps2.com/>
- <http://www.gamespot.com/ps2/action/killzone/index.html>

# Killzoneの特徴( 2 )

敵は人間(宇宙から帰って来た)

Killzone 紹介画像(省略、以下参照)

- <http://sega.jp/ps2/killzone/>
- <http://www.killzoneps2.com/>
- <http://www.gamespot.com/ps2/action/killzone/index.html>



# Killzoneの特徴(3)

マップは地上。建物の中、荒野など多様で広い。

Killzone 紹介画像(省略、以下参照)

- <http://sega.jp/ps2/killzone/>
- <http://www.killzoneps2.com/>
- <http://www.gamespot.com/ps2/action/killzone/index.html>

# Killzoneの特徴( 4 )

## 建築物は殆ど破壊できない。

Killzone 紹介画像(省略、以下参照)

- <http://sega.jp/ps2/killzone/>
- <http://www.killzoneps2.com/>
- <http://www.gamespot.com/ps2/action/killzone/index.html>

# プレイモード

## (1) キャンペーン

ストーリーモード

## (2) バトルフィールド

C O Mとの対戦(同等の立場で戦闘)

- デスマッチ .... 最大13のC O Mを相手にデスマッチ
- チームデスマッチ .... 最大7vs7(一人はプレイヤー)で戦う
- ドミネーション ... 5つの占拠場所を奪い合う
- サプライドロップ .... 5つのコンテナを自分の陣地へ持ち帰る
- アサルト ... プレイヤー側が2つの重要物資を敵から守る
- ディフェンド&デストロイ ... お互いに2つの重要物資を敵から守る

C O Mの人数と難易度は設定できる

# Killzone のプレイ画面デモ

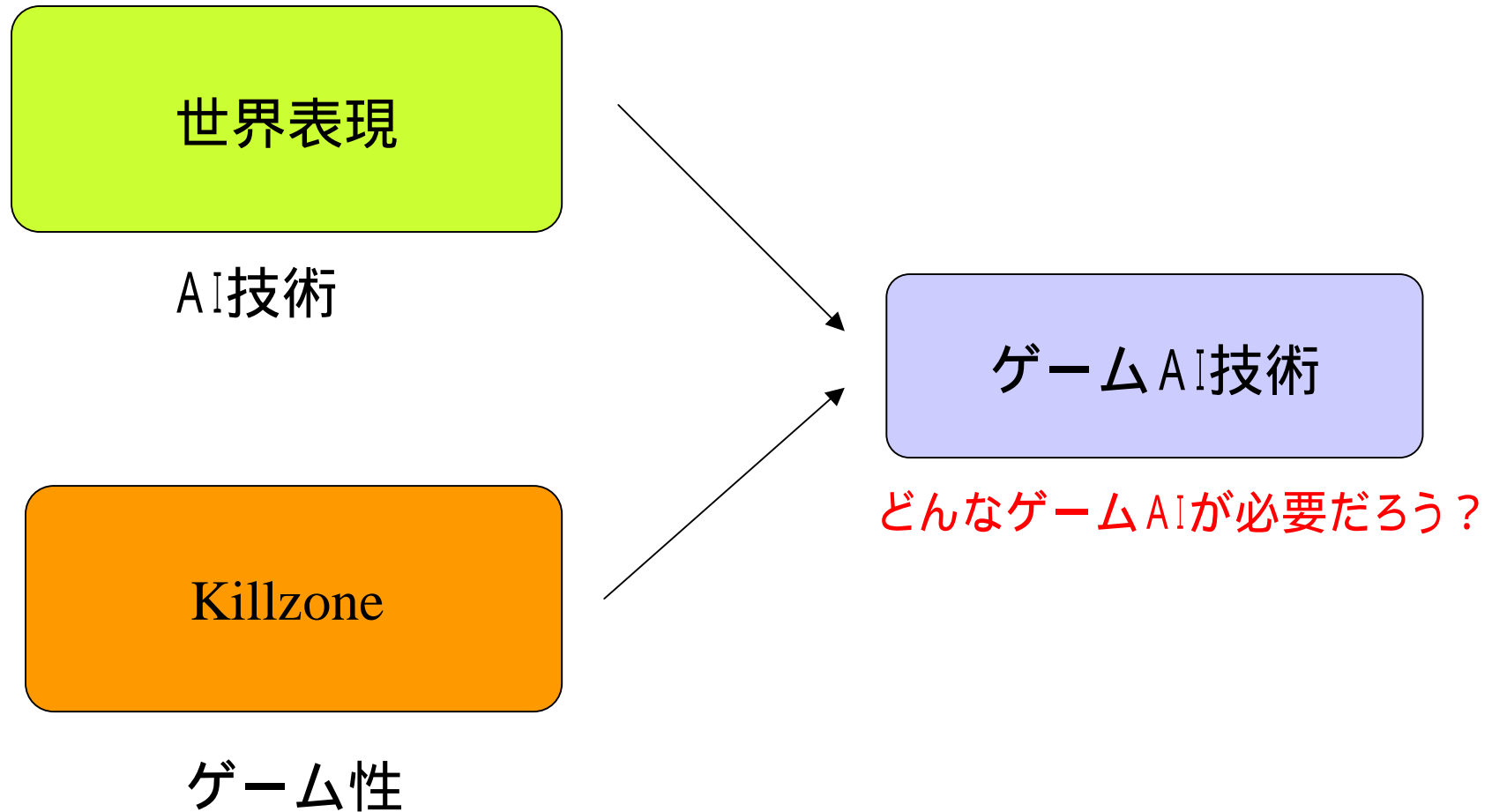
Killzone 紹介映像(省略、以下参照)

<http://sega.jp/ps2/killzone/>

<http://www.killzoneps2.com/>

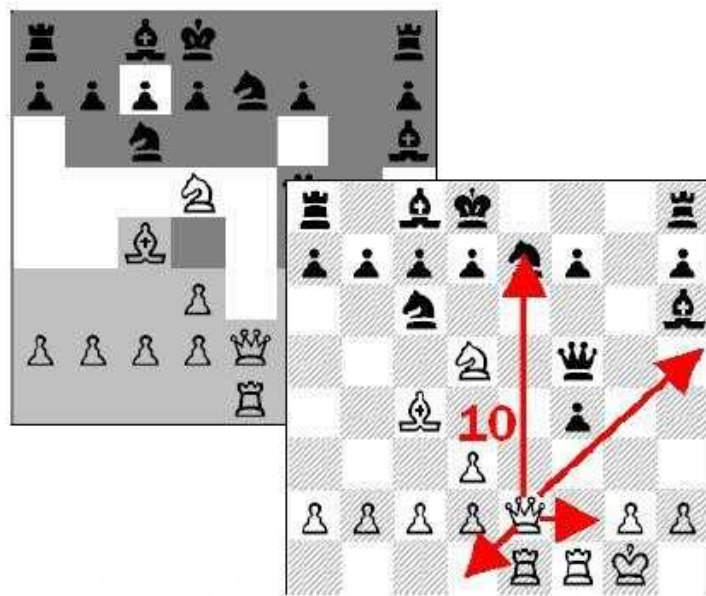
<http://www.gamespot.com/ps2/action/killzone/index.html>

# Killzone と世界表現

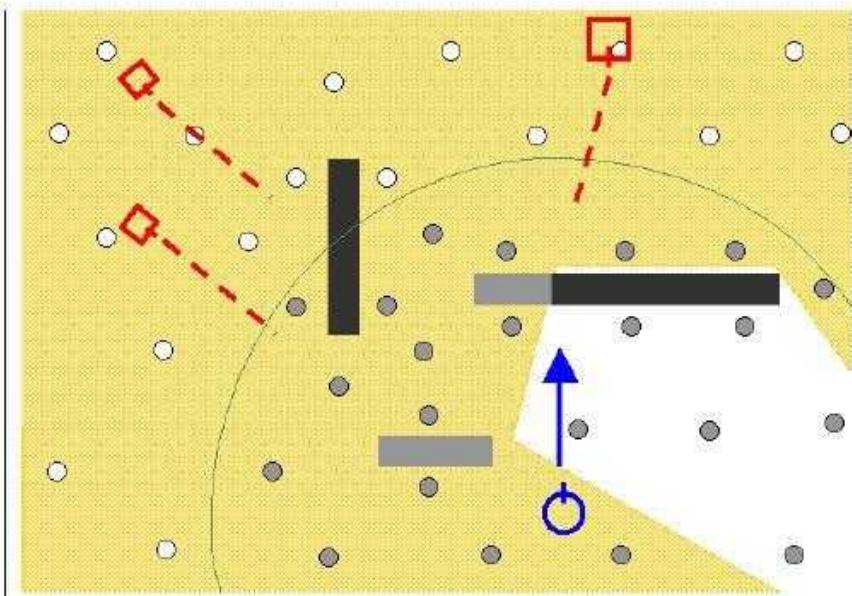


# Killzone のAI が目指すもの

- (1) 複雑で広いマップの中を
- (2) 目的に応じて敵に対して有利な位置を取りつつ戦う



**a. Board control and piece mobility in chess.**



**b. Lines-of-fire and mobility in a tactical shooter.**

Figure from pages 4 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics" [3]

# Killzone のAI が目指したもの

- (1) 複雑で広いマップの中を
- (2) 目的に応じて敵に対して有利な位置を取りつつ戦う

戦闘最適距離に近付く  
自分の射線の通る場所へ行く  
敵の射線の通る場所を避ける  
遮蔽物の側にいる  
屈めば遮蔽されるような場所へ行く  
壁に沿って歩く  
危険地帯から遠ざかる



どのようなAIの能力が必要か？

# 一般的なアプローチ

- (1) レベルデザイナーによるヒントの埋め込み
- (2) スクリプトによる条件反射

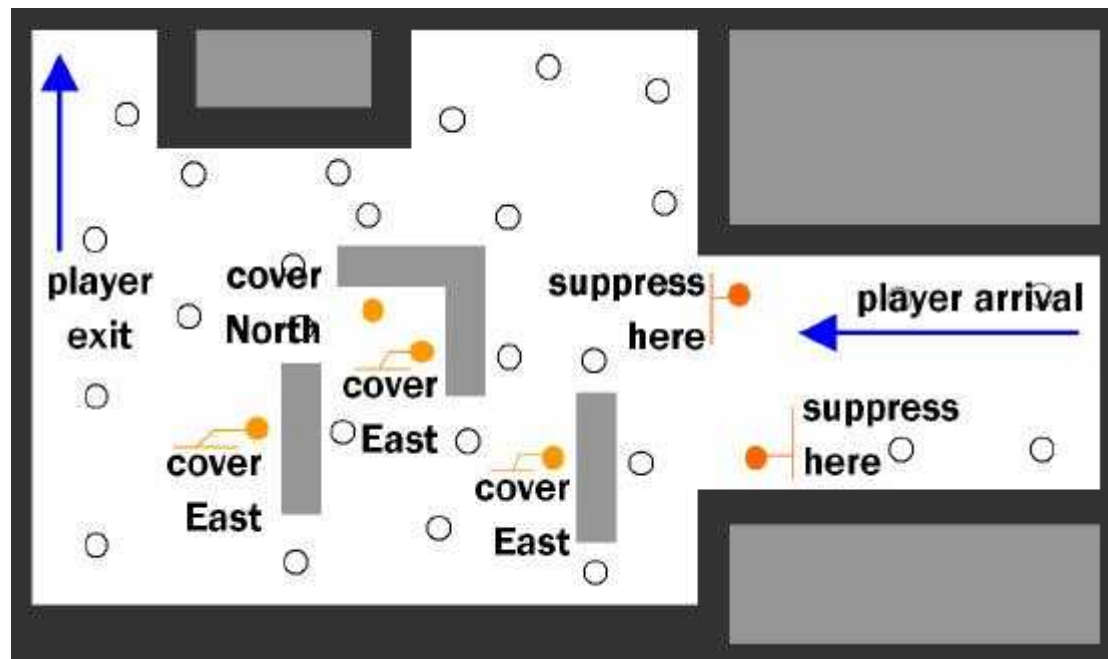


Figure from pages 5 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

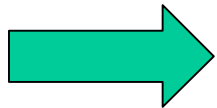
レベルデザイナーが想定できる範囲のことをする。  
手間がかかる。



# Killzone のアプローチ

戦闘最適距離に近づく  
自分の射線の通る場所へ行く  
敵の射線の通る場所を避ける  
遮蔽物の側にいる  
屈めば遮蔽されるような場所へ行く  
壁に沿って歩く  
危険地帯から遠ざかる

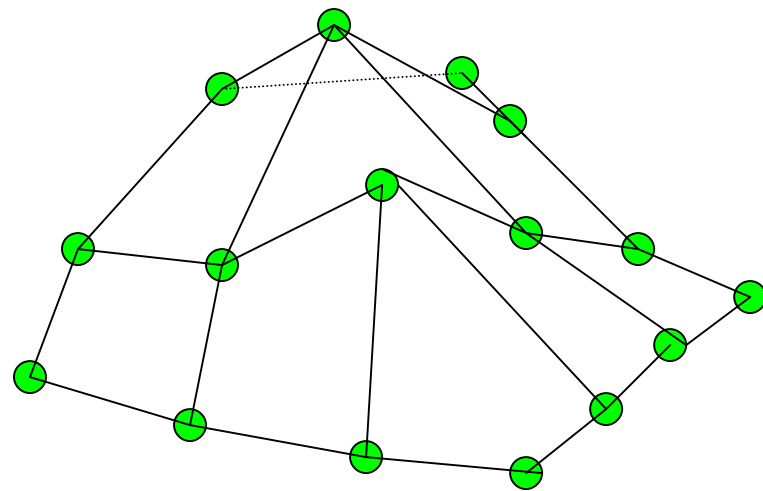
目的と状況に応じて最適な位置を正確に割り出すことができる能力  
をCOMに持たせたい



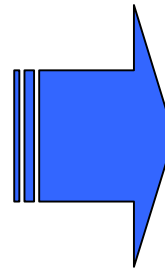
戦術的位置検出 (Tactical position picking)

# 位置検出(position picking)とは？

候補となる位置ポイントの中から、最適なポイントを見つけること



地形モデル



## 用意されたデータ

5 10	7 9	2 8	1 3
4 9	10 8	3 6	2 2
5 3	1 8	6 10	2 9
1 1	3 7	5 3	8 1

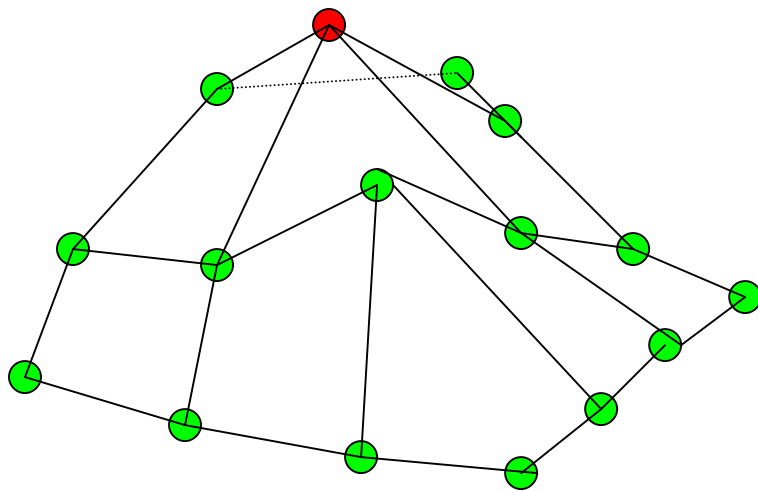
俯瞰した図

赤 高さ  
青 視界の広さ

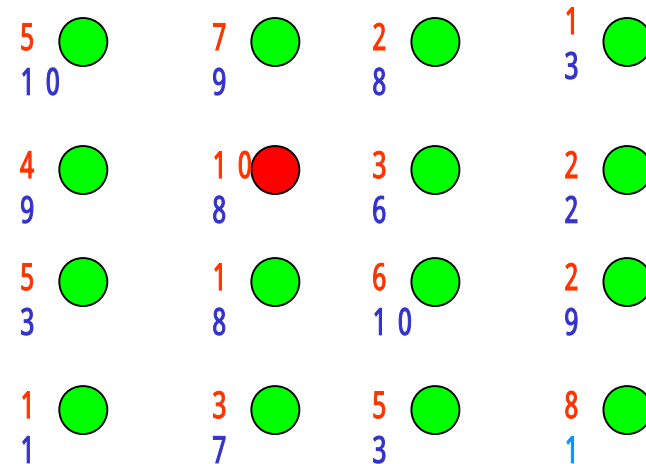
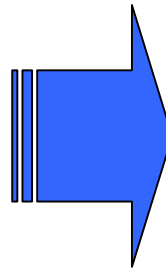
# 位置検出(position picking)とは？

最も高い場所は？ = 高さの最大値を持つ点

用意されたデータ



地形モデル

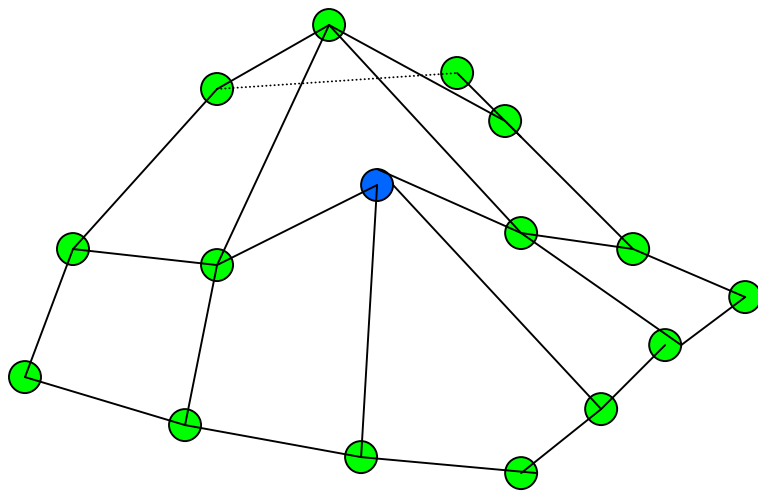


俯瞰した図

赤 高さ  
青 視界の広さ

# 位置検出(position picking)とは？

最も見晴らしのよい場所は？ = 視界の広さの最大値を持つ点



地形モデル

## 用意されたデータ

5 10	7 9	2 8	1 3
4 9	10 8	3 6	2 2
5 3	1 8	6 10	2 9
1 1	3 7	5 3	8 1

俯瞰した図

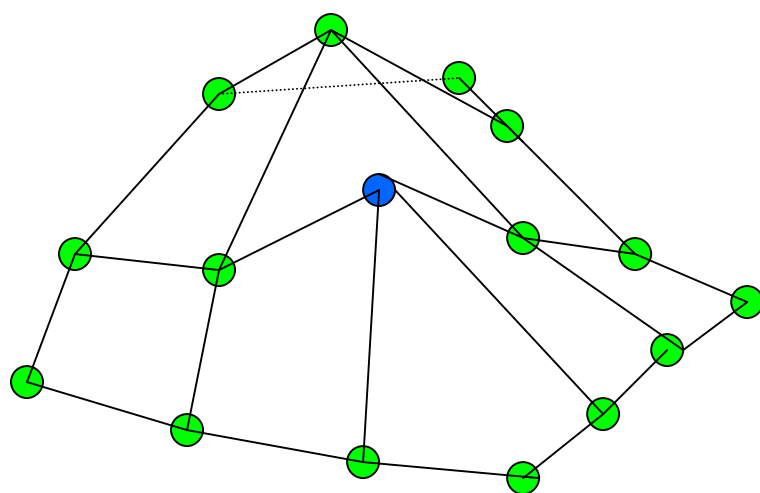
赤 高さ  
青 視界の広さ

# 位置検出(position picking)とは？

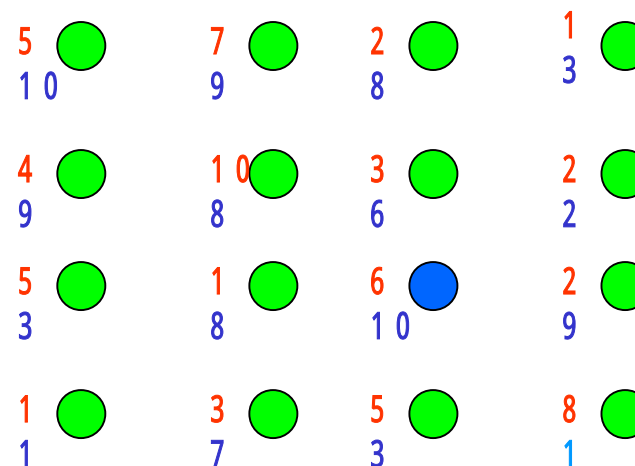
最も高くて見晴らしがよい場所は？ = 高さと視界の広さを持つ点

位置評価関数の導入  $f(h, v) = 0.7 * h + 0.3 * v$

用意されたデータ



地形モデル



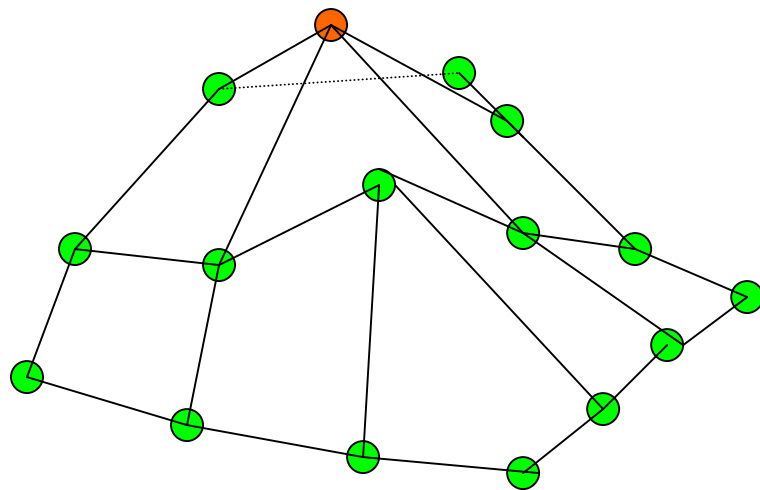
俯瞰した図

赤 高さ  
青 視界の広さ

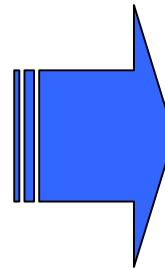
# 位置検出(position picking)とは？

最も高くて見晴らしがよい場所は？ = 高さと視界の広さを持つ点

ウェイトの付いた評価関数の導入  $f(h, v) = 0.7 * h + 0.3 * v$



地形モデル



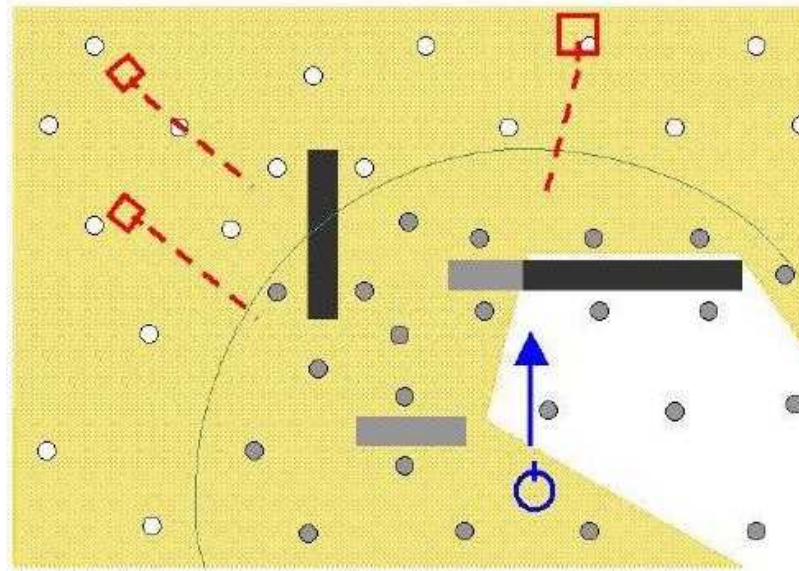
用意されたデータから  
評価関数によって計算したデータ

6.5	7.6	3.8	1 3
5.5	9.4	3.9	2 2
4.4	3.1	7.2	2 9
1.0	4.2	4.4	8 1

俯瞰した図

赤 高さ  
青 視界の広さ

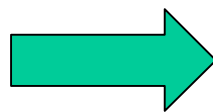
## 戦術的位置検出(tactical position picking)



b. Lines-of-fire and mobility in a tactical shooter.

Figure from pages 4 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics" [3]

戦術(接近して攻撃、隠れる、逃げるなど)に対して戦況に応じて最適な位置を見つける能力



判断に必要な情報を位置(ポイント)に対して用意する

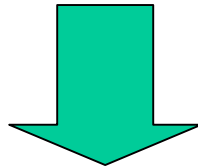
# 静的位置検出と動的な位置検出

## (1) 静的な位置検出

時間と共に状況が変化しないステージで  
最適なポイントを見出す

### 第2章

William van der Sterren,  
“Terrain Reasoning for 3D Action Games” (GDC 2001)



## (2) 動的な位置検出

時間と共に状況が変化する(移動する敵など)ステージで  
リアルタイムに最適なポイントを見出す

### 第3章

Remco Straatman, Arjen Beij, William van der Sterren,  
“Killzone's AI : Dynamic Procedural Combat Tactics” (GDC 2005)



# 第1章まとめ

- (1) ゲームAIを知識表現という視点から見ると、  
AIのために何をすればよいかが見えて来る。
- (2) 知識表現の一つで、AIの属する世界の大局的な情報  
を表現する世界表現は、これまでもゲームで多用されて来た。
- (3) Killzone は戦術的に移動して戦うAIを目指し、  
位置検出技術を基本としている。

# 質疑応答

# 第1部 ゲームAI 技術解説(60分)

1. 技術テーマとゲーム説明 (15分)
2. 静的位置検出 (Killzone の先行研究) (15分)
3. 動的位置検出 (Killzone's AI) (20分)
4. 発展 (5分)

各章の終わりに質疑応答を行います。

## 第2章

## 静的位置検出

(Killzone の先行研究)

## 第2章

## 静的位置検出

狙撃ポイントとして最適な場所は何処か？



Figure from pages 6 of William van der Sterren(2001) ,"Terrain Reasoning for 3D Action Games" [1]

問題： 東向きに進行する敵に対する  
狙撃ポイントとして最適な場所は何処か？

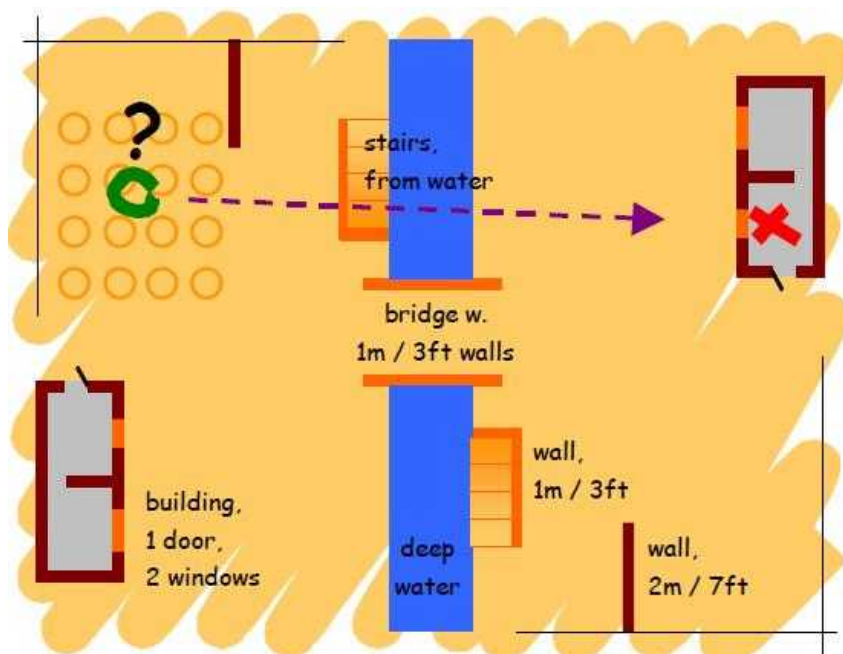


Figure from pages 30-31 of William van der Sterren(2001),"Terrain Reasoning for 3D Action Games(GDC2001 PPT)" [2]



# 静的な戦術的位置検出(tactical position picking) 工程

Step1 ウェイポイントを等間隔に敷き詰める

Step2 各ポイントに地形の持つ基本情報を蓄積する

Step3 各ポイントの基本情報から抽象的な情報を抽出し評価する

Step4 総合的に情報を計算する

これからは全て開発中に事前解析としてデータを作っておく



## Step1 ウェイポイントを等間隔に敷き詰める



Figure from pages 6 of William van der Sterren (2001), "Terrain Reasoning for 3D Action Games" [1]

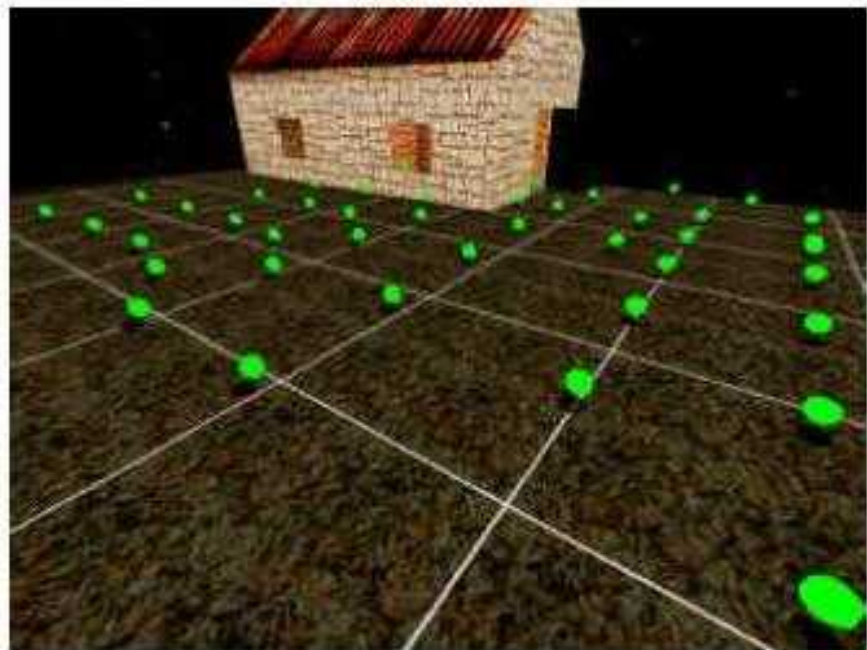


Figure from pages 31 of William van der Sterren(2001), "Terrain Reasoning for 3D Action Games(GDC2001 PPT)" [2]



## Step2 各ポイントに地形の持つ基本情報を蓄積する

### 各ポイントの性質のカテゴリー

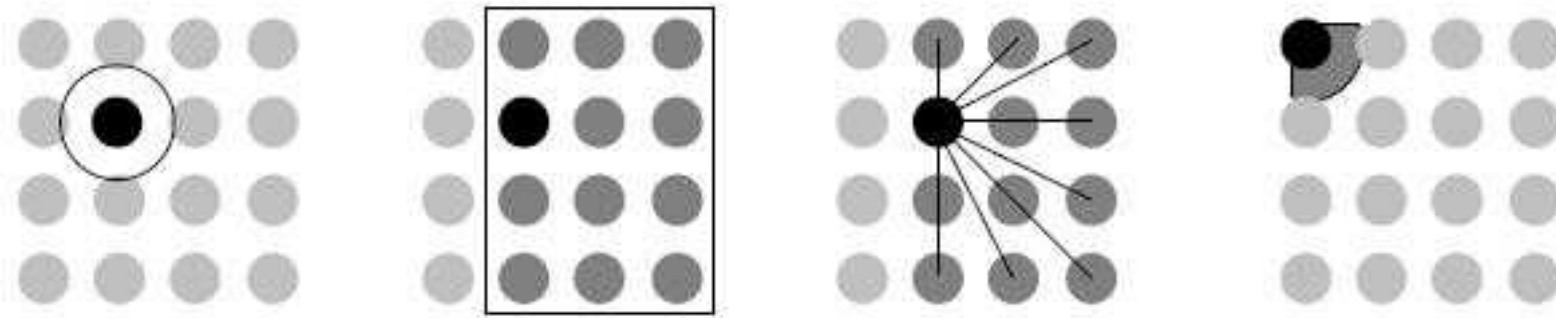


Figure from pages 5 of William van der Sterren (2001), "Terrain Reasoning for 3D Action Games" [1]

局所的な性質

➡ 明るさ、障害物があるか

グループとしての性質

➡ 部屋、河などでウェイポイントが形成するグループ

他のウェイポイントとの関係

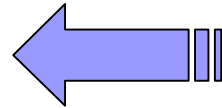
➡ 視線、射線、距離。

フォーカス

➡ ある性質がどれくらい偏っているか。

### Step3 各ポイントの基本情報から抽象的な情報を抽出し評価する

抽象的情報を評価するのに  
必要な基本情報



狙撃ポイントに必要な抽象的情報

- 明るさ
- 障害物があるか
- 特殊な地形かどうか
- 見渡せる距離
- ターゲットまでの到達時間
- 視線
- 射線
- ...

- 見つかりにくい
- 視界が開けている
- 敵から攻撃されにくい
- 遮蔽物が近くにある
- 特に東向きに視界が開けている
- 確実に攻撃ポイントがある
- 背中に対して遮蔽物がある
- 自由に動けるだけのスペースがある
- 敵は遮蔽物が近くでない



位置評価関数

「狙撃ポイントに必要な抽象的情報」(行)と  
「抽象的情報を評価するのに必要な基本情報」(列)の対応表

description	terrain property	light level	obstacles	special terrain	line distance	travel time	line of sight	line of fire	portal member	game activity		property	directional	higher level	focus
a	inconspicuous	✓										✓			
b	overlook distant spots				✓		✓	✓					✓		✓
c	hard to reach					✓							✓		✓
d	cover from targets					✓	✓	✓					✓		✓
e	target without cover						✓	✓					✓		✓
f	overlook key spots			✓			✓	✓	✓				✓	✓	✓
g	observed approaches					✓	✓						✓		
h	focus														✓
i	local movement		✓	✓								✓			
j	overlook traffic			✓					✓				✓		✓

Figure from pages 8 of William van der Sterren (2001), "Terrain Reasoning for 3D Action Games" [1]

## (例) 「敵から隠れているか」(cover from target)の評価

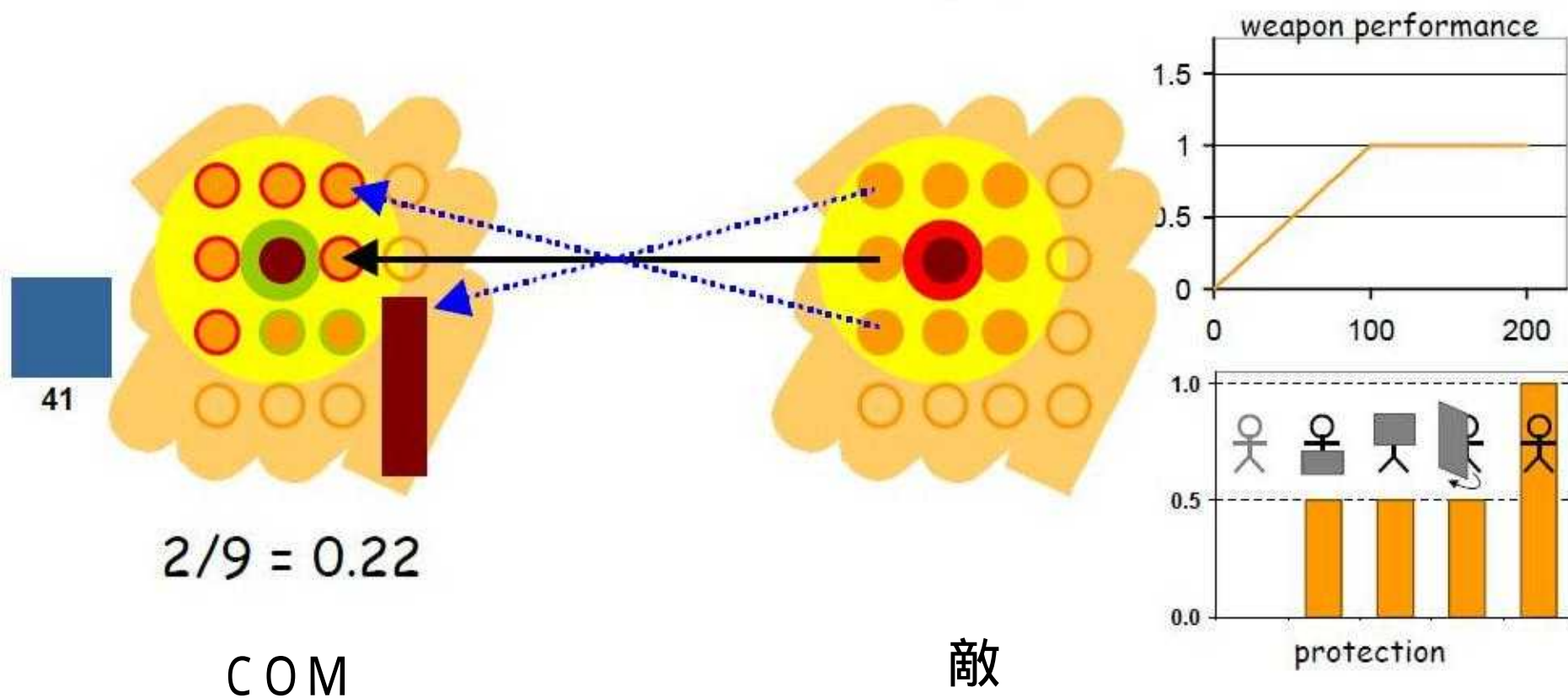


Figure from pages 41 of William van der Sterren(2001),"Terrain Reasoning for 3D Action Games(GDC2001 PPT)" [2]

敵から自分の周囲のウェイポイントに視線が通るかどうかで

## (例) 「敵から隠れているか」(cover from target)の評価

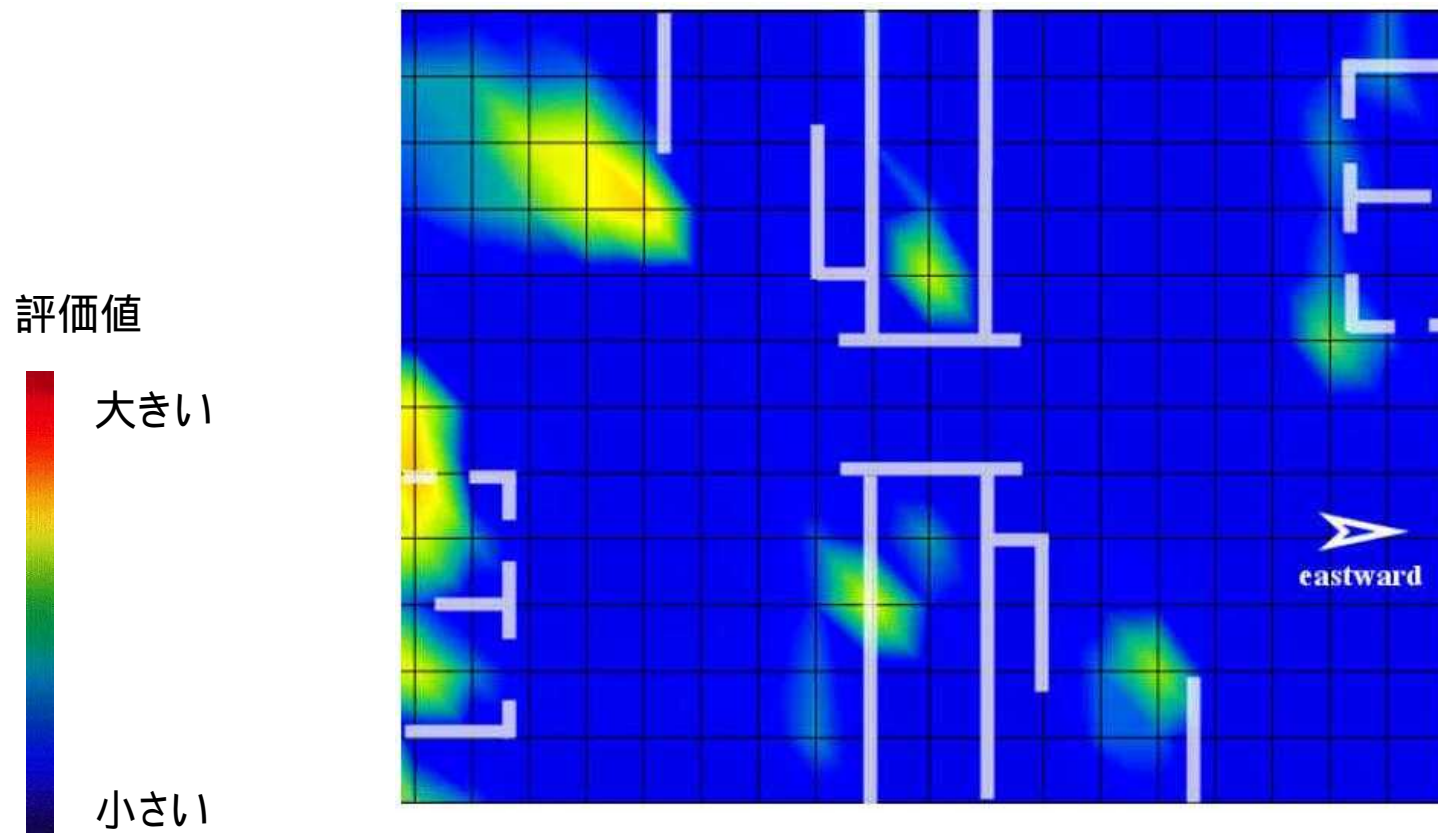


Figure from pages 42 of William van der Sterren(2001),"Terrain Reasoning for 3D Action Games(GDC2001 PPT)" [2]

## 「敵から隠れているか」(cover from target)の評価値の分布

例えば...「東向きだけに開けているか」(focus)

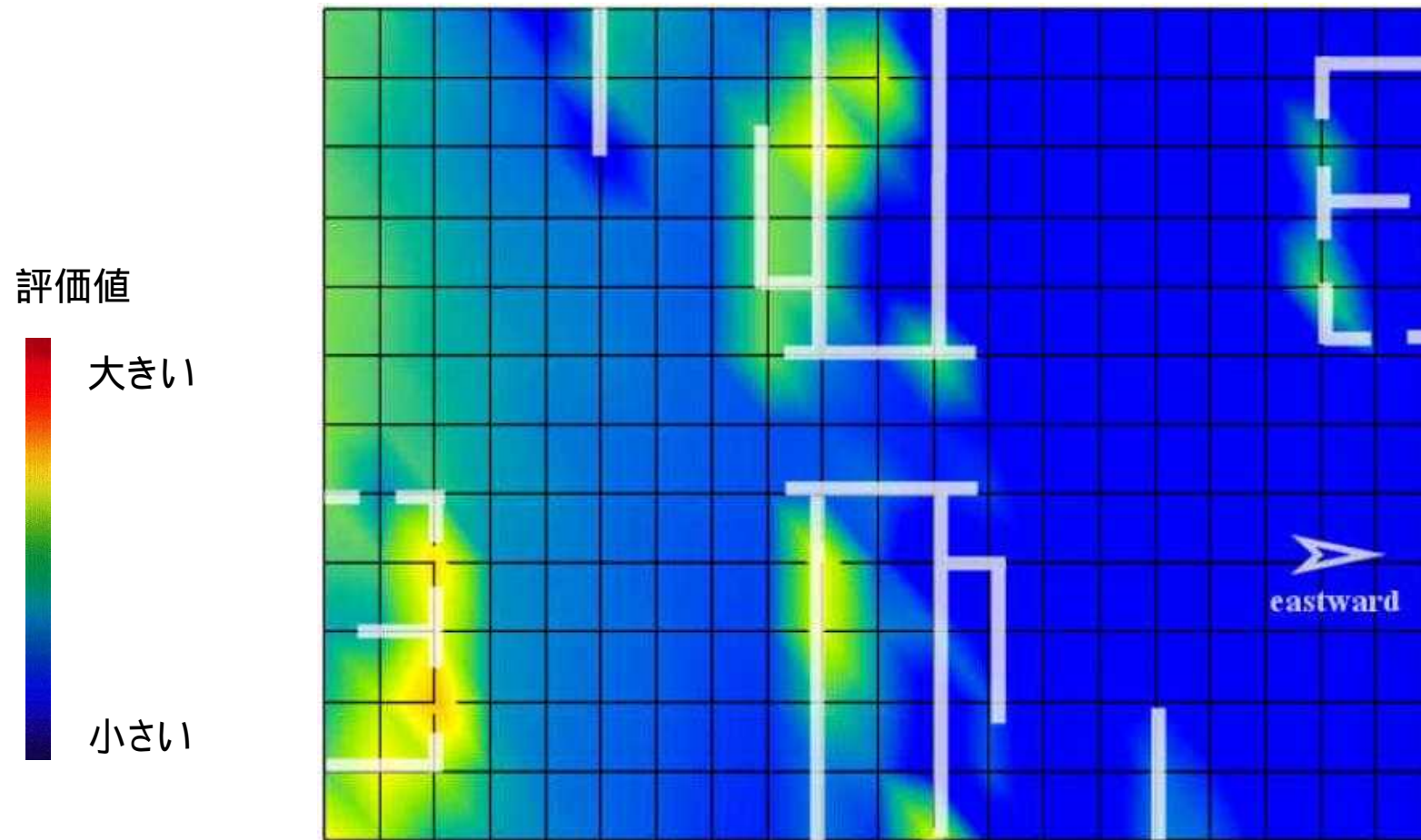
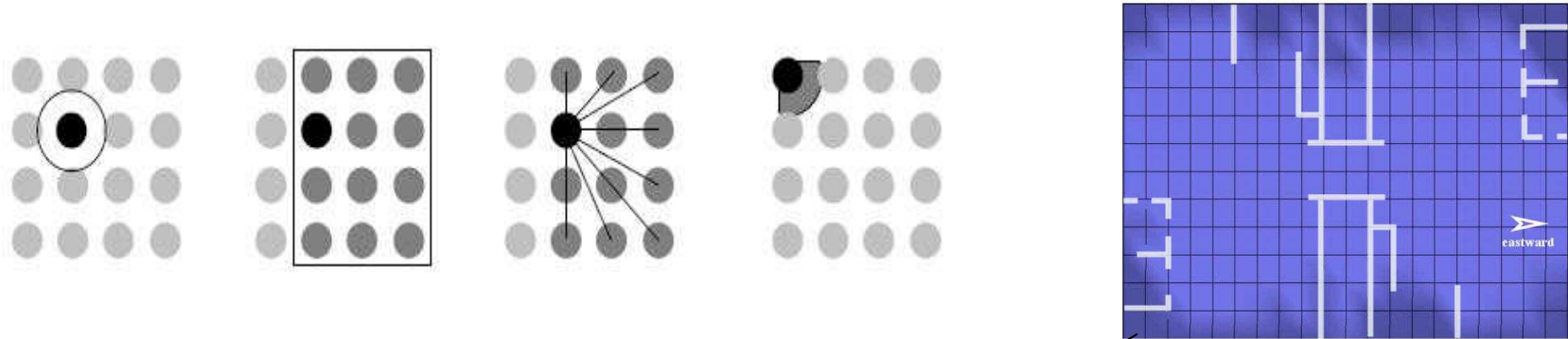


Figure from pages 45 of William van der Sterren(2001),"Terrain Reasoning for 3D Action Games(GDC2001 PPT)" [2]

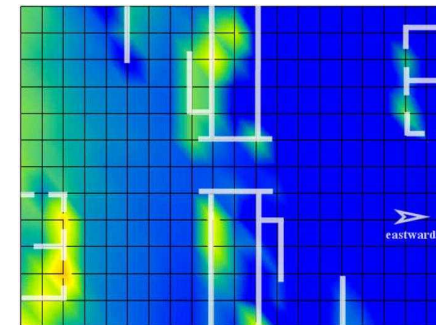
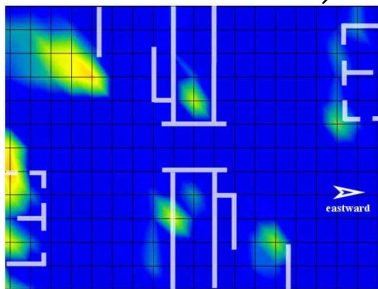


## Step4 総合的に情報を計算する



位置wのd方向に対する狙撃ポイントの評価値

$$\begin{aligned} &= \text{(局所的な性質を評価した点数)} \\ &+ \text{(グループとしての性質を評価した点数)} \\ &+ \text{(フォーカス)} \times \text{(他のウェイポイントとの関係性を評価した点数)} \end{aligned}$$



## Step4 総合的に情報を計算する

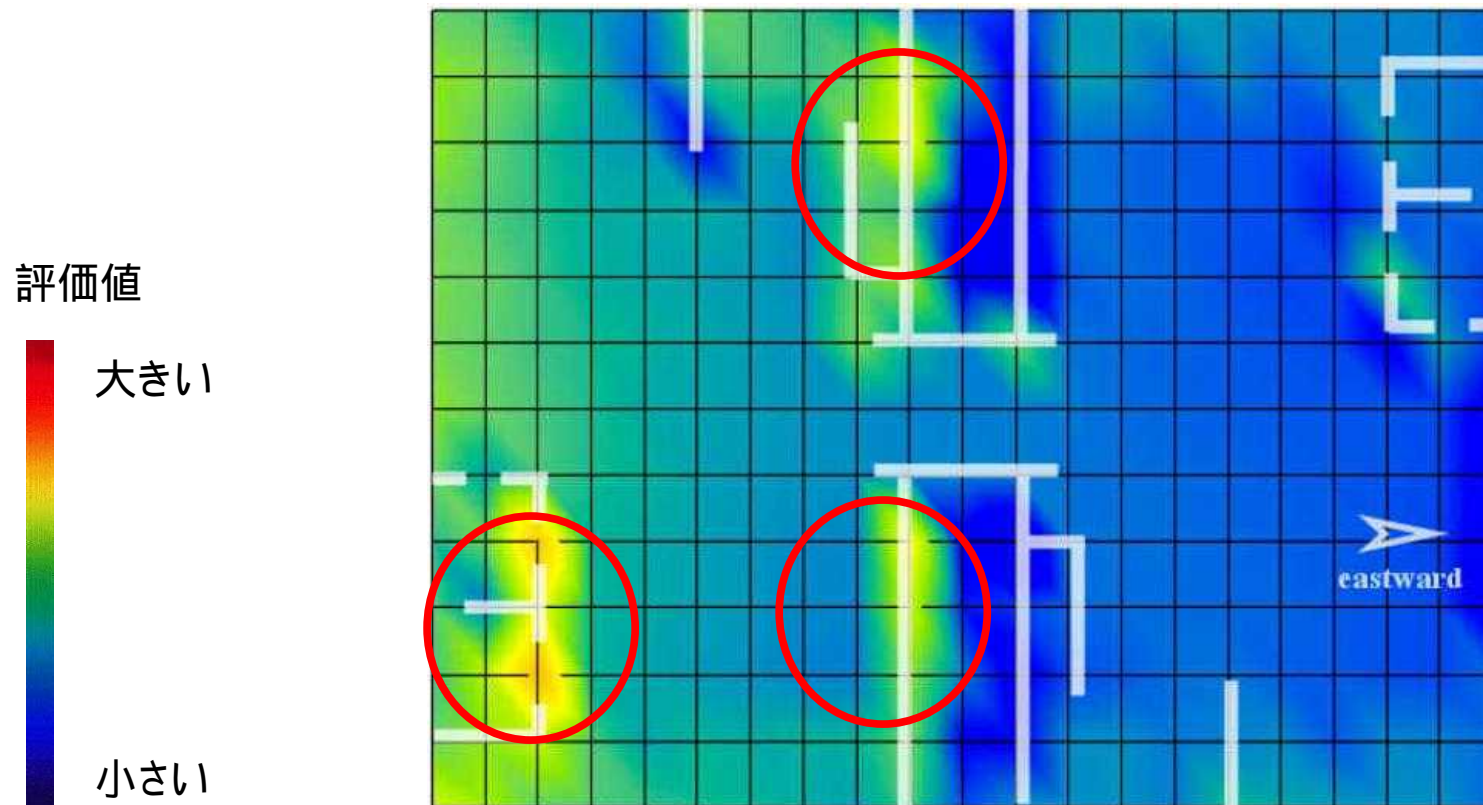


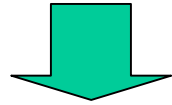
Figure from pages 9 of William van der Sterren (2001), "Terrain Reasoning for 3D Action Games" [1]

最終的な位置  $w$  の  $d$  方向に対する狙撃ポイントの評価値の分布

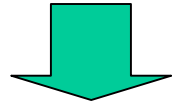


# 第2章まとめ

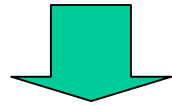
狙撃ポイントという戦術的位置をAIに求めさせることができた



コンピューターは一つの知的な機能を獲得した  
(人間にはあたりまえのことをコンピューターにさせることは難しい)



例えば、方向を変えれば瞬時に別の狙撃ポイントをみつけてくれる



このままでは、即座にゲームに使えるそうにはない

どうすればよいだろうか？

使えない。あきらめる

少し変えれば使えるそうだ。  
Killzone へ<sup>65</sup>

# 質疑応答

# 第1部 ゲームAI 技術解説(60分)

1. 技術テーマとゲーム説明 (15分)
2. 静的位置検出 (Killzone の先行研究) (15分)
3. 動的位置検出 (Killzone's AI) (20分)
4. 発展 (5分)

各章の終わりに質疑応答を行います。

## 第3章

## 動的的位置検出

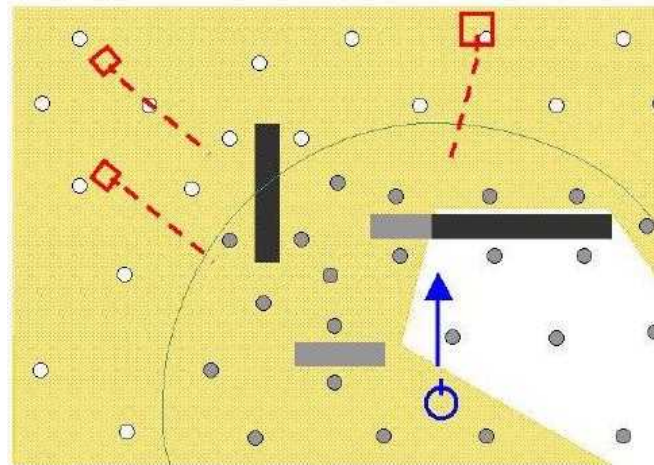
Killzone のAI解説

# 第3章

# 動的的位置検出

リアルタイムに変化する敵、味方の位置と地形から最適な位置を計算によって求める

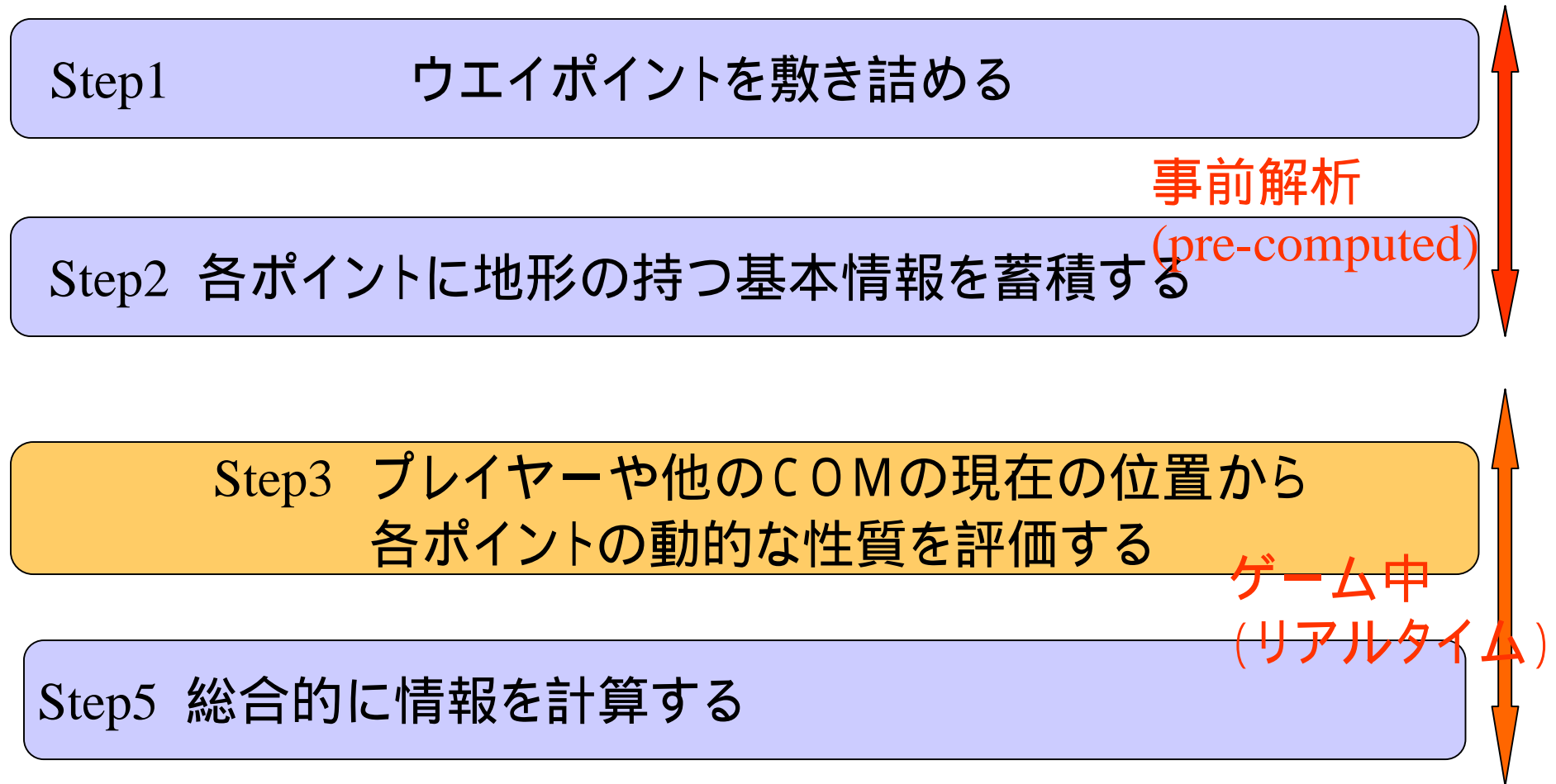
- 
- (1) 敵の射線になるべくさらされずに、  
屈めば敵から隠れることができるポイントへ移動する
  - (2) 敵が隠れている場所から移動しそうな場所に手榴弾を投げる
  - (3) 敵が出現しそうな場所に対して威嚇射撃を行う



b. Lines-of-fire and mobility in a tactical shooter.

Figure from pages 4 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics"

# 動的な戦術的位置検出(tactical position picking) 工程



## Step1 ウェイポイントを敷き詰める

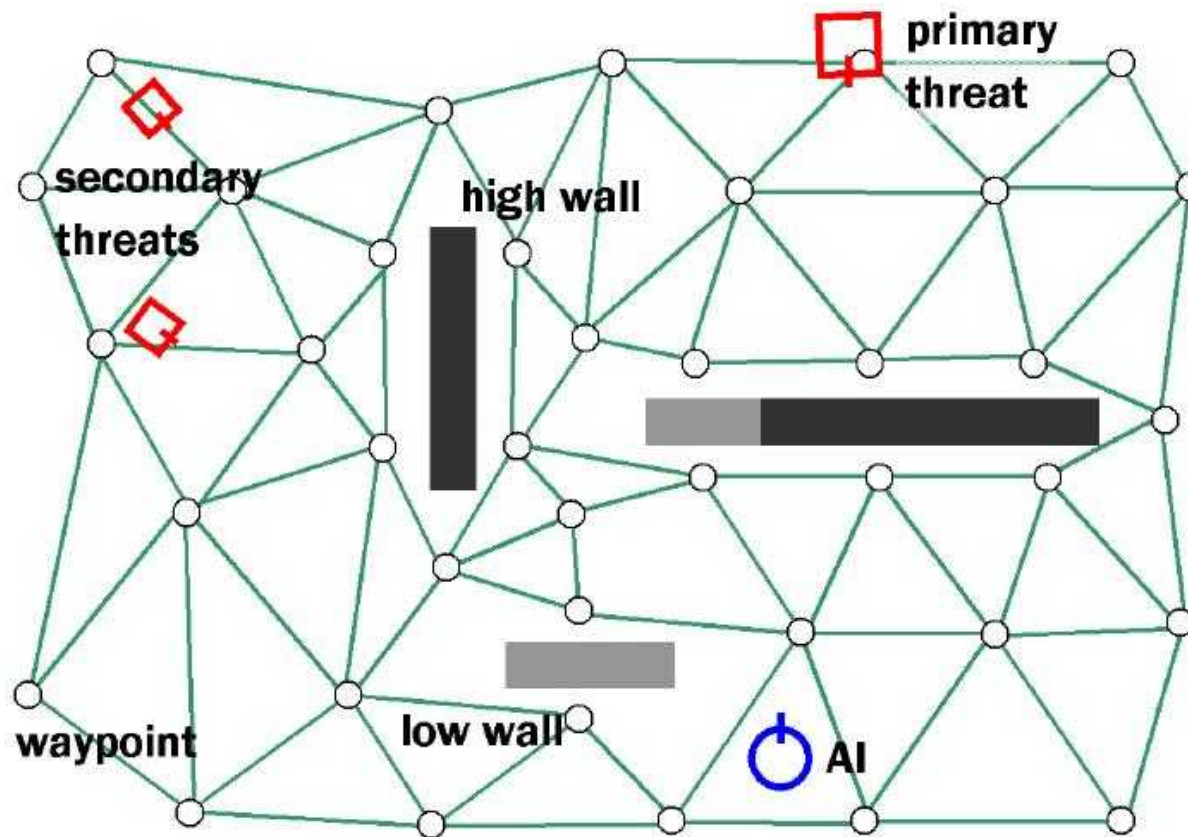


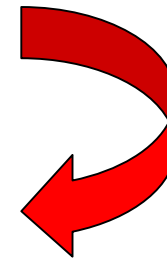
Figure from pages 11 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

## Step2 各ポイントに地形の持つ基本情報を蓄積する

事前に用意しておく。

静的な情報(ゲーム内で変化しない)  
オブジェクトの位置など

動的な情報(ゲーム内で変化する)  
キャラクターの位置に関するもの



ポイント毎のデータ

事前計算されたデータと  
ゲーム状況から  
ゲーム内で計算する。<sup>72</sup>



## Step2 各ポイントに地形の持つ基本情報を蓄積する

静的な情報(ゲーム内で変化しない) = オブジェクトの位置など

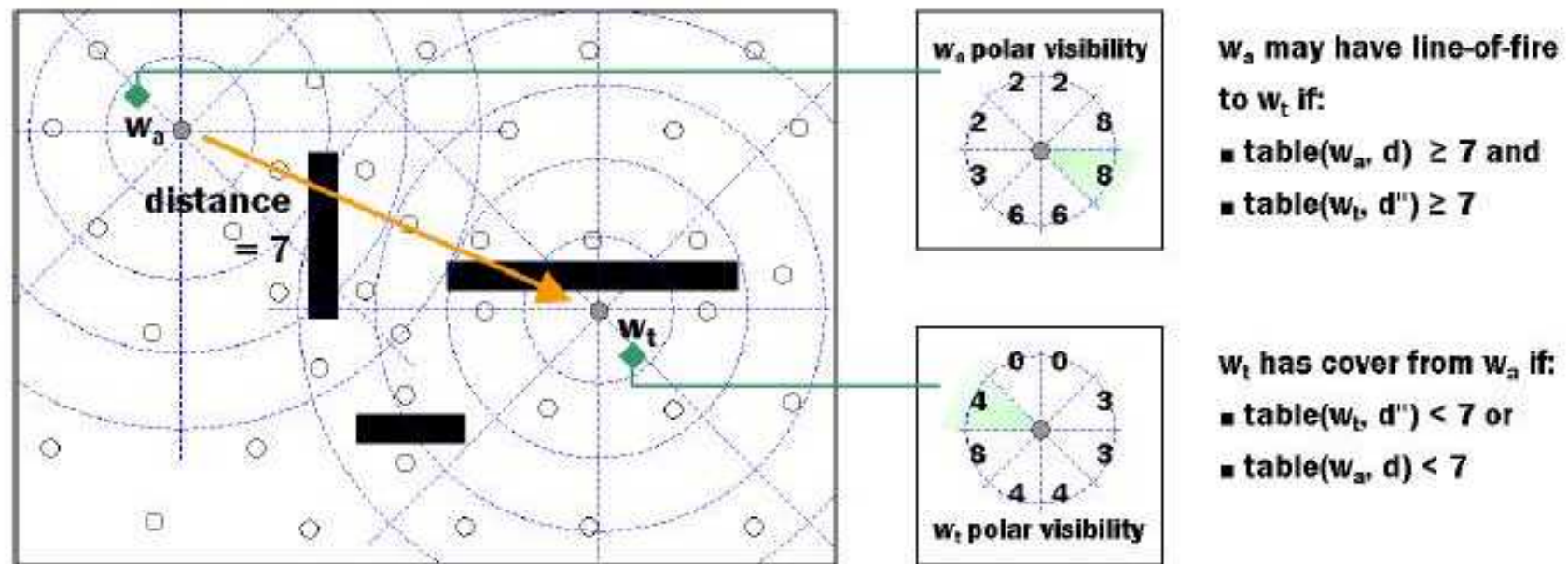


Figure from pages 9 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics" [3]

8方向に対して、可視距離(キャラクターの体が見える限界距離)を計算する(参照テーブルとして持つ)

# 射線判定、遮蔽判定

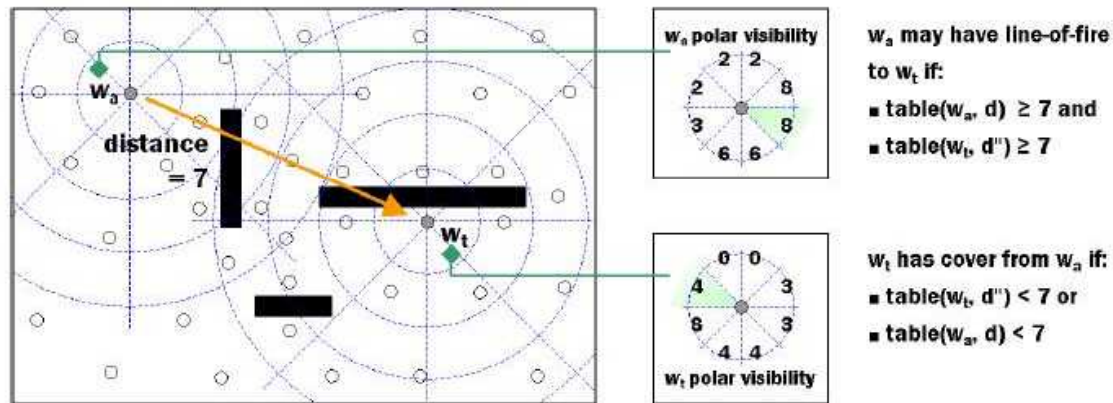
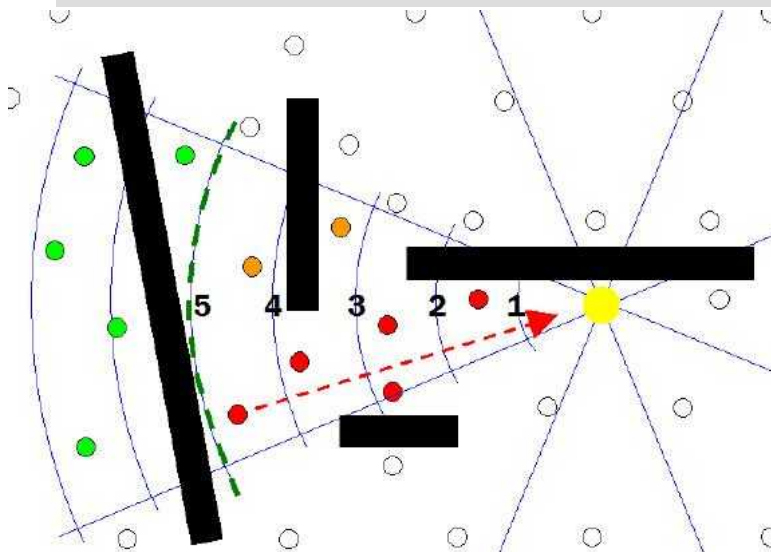


Figure from pages 9 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics" [3]



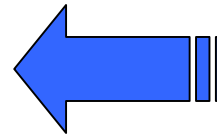
その方向の扇形(セクター)内の全ての点に対して ray cast 計算を行い、その中の最大値を可視距離とする。

Figure from pages 38 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

### Step3 プレイヤーや他のCOMの現在の位置から 各ポイントの動的な性質を評価する

動的な情報(ゲーム内で変化する) = キャラクターの位置に関するもの

抽象的情報を評価するのに  
必要な基本情報



抽象的情報

- ◆ 第一脅威への距離
- ◆ 第一脅威への射線があるか
- ◆ 第一脅威から遮蔽されているか
- ◆ 第二脅威から遮蔽されているか
- ◆ 自分の射線が敵へ通っているか

現在の位置からどれだけ近いか

- 敵から遮蔽されているか
- 敵の射程からどれくらい離れているか
- 味方の射程からどれだけ離れているか
- 味方からどれだけ離れているか
- 

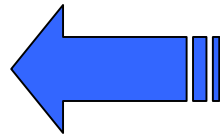
Position evaluation input	Description: (The evaluation function awards a higher score for ...)
Proximity to current position	being closer to or more quickly reached from the current position
Proximity from specific location	being closer to or more quickly reached from a specific location
Cover from primary threat	offering cover from the primary threat (given a stance)
Line-of-fire to primary threat	offering cover from the primary threat (given a stance)
Distance to primary threat	being preferred fighting distance from the primary threat
Outside danger zone	being outside the blast range of (expected) projectile
Cover from secondary threats	offering cover from one or more specified threats other than the primary threat
Outside friendly line-of-fire	being outside the line-of-fire of specified friendly units
Distance from friendly positions	being some distance away from friendly positions
Wall hugging	being close to a wall or obstacle in a specified direction
Nearby cover	being near positions offering cover from the primary threat
Player line-of-fire	not being a position across the player's line-of-fire from the current position
Preferred fighting range	being inside the preferred fighting range

Figure from pages 5 of Remco Straatman,  
Arjen Beij, William van der Sterren  
(2005), "Killzone's AI :  
Dynamic Procedural Combat  
Tactics" [3]

Step3 プレイヤーや他のCOMの現在の位置から  
各ポイントの動的な性質を評価する

動的な情報(ゲーム内で変化する) = キャラクターの位置に関するもの

抽象的情報を評価するのに  
必要な基本情報

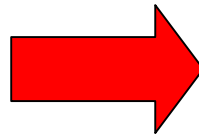


抽象的情報

- ◆ 第一脅威への距離
- ◆ 第一脅威への射線があるか
- ◆ 第一脅威から遮蔽されているか
- ◆ 第二脅威から遮蔽されているか
- ◆ 自分の射線が敵へ通っているか

現在の位置からどれだけ近いか

- 敵から遮蔽されているか
- 敵の射程からどれくらい離れているか
- 味方の射程からどれだけ離れているか
- 味方からどれだけ離れているか
- ....



評価関数

Step3 プレイヤーや他のCOMの現在の位置から  
各ポイントの動的な性質を評価する

何ができるか？

## 動作例

- (1) 第二敵の射線にさらされない、  
第一敵の敵を攻撃できるポイントを見つける
- (2) 敵が隠れている場所から移動しそうな場所に手榴弾を投げる
- (3) 敵が出現しそうな場所に対して威嚇射撃を行う

## 動作例 1

第二敵の射線にさらされない、  
第一敵の敵を攻撃できるポイントを見つける

Step1

一定半径内のポイントを候補に上げる

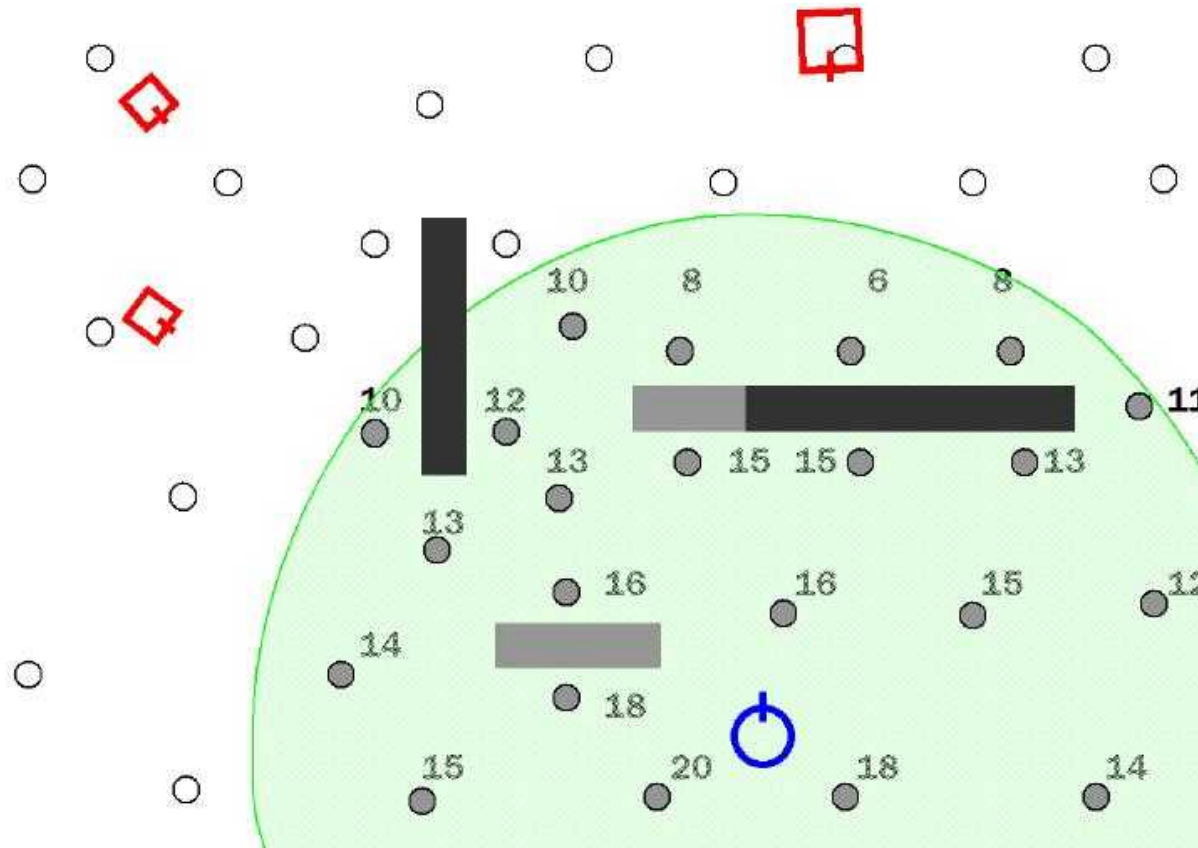


Figure from pages 12 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

この時点で他のポイントはこれ以降、評価しない ➡ 計算量の軽減



## Step2 第一の敵から射線の通るポイント进行评估する

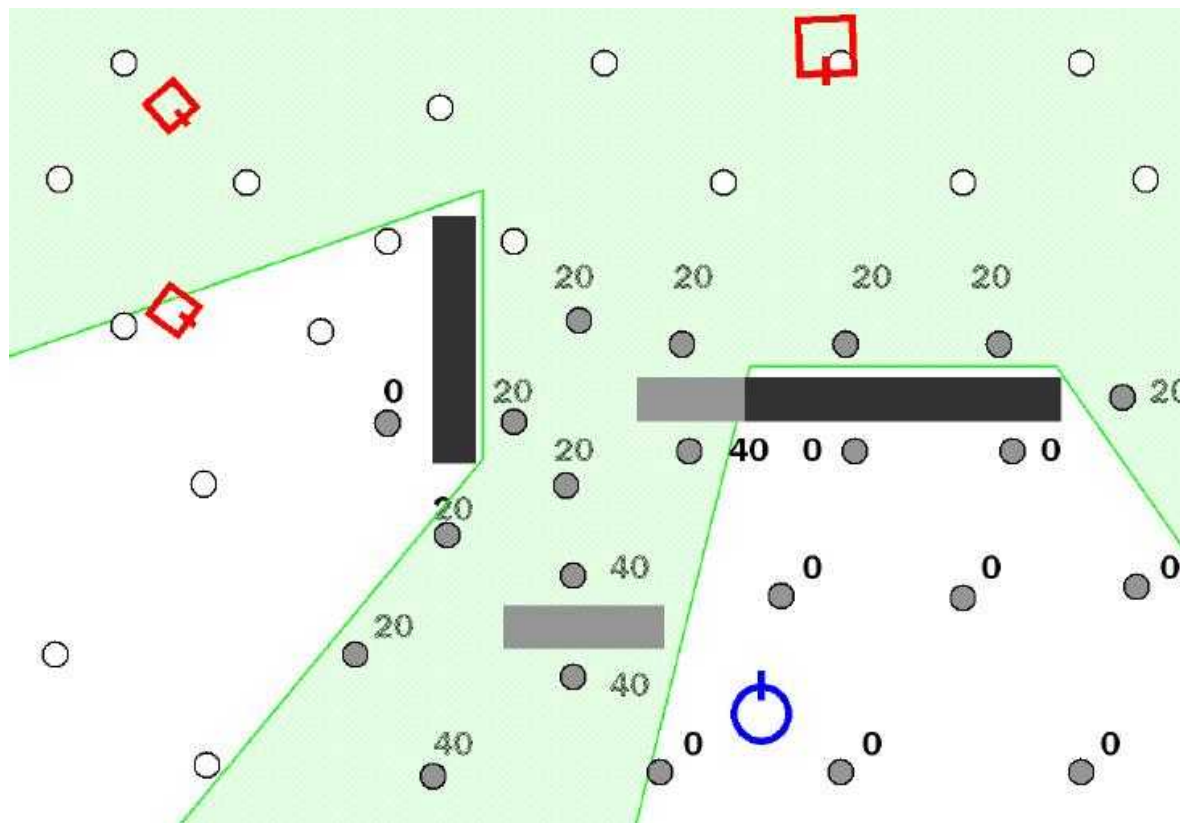


Figure from pages 13 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

40点 ... 立った状態のみ射線が通る  
20点 ... 座っても立っても射線が通る



### Step3 第二の敵たちから射線の通らないポイント进行评估する

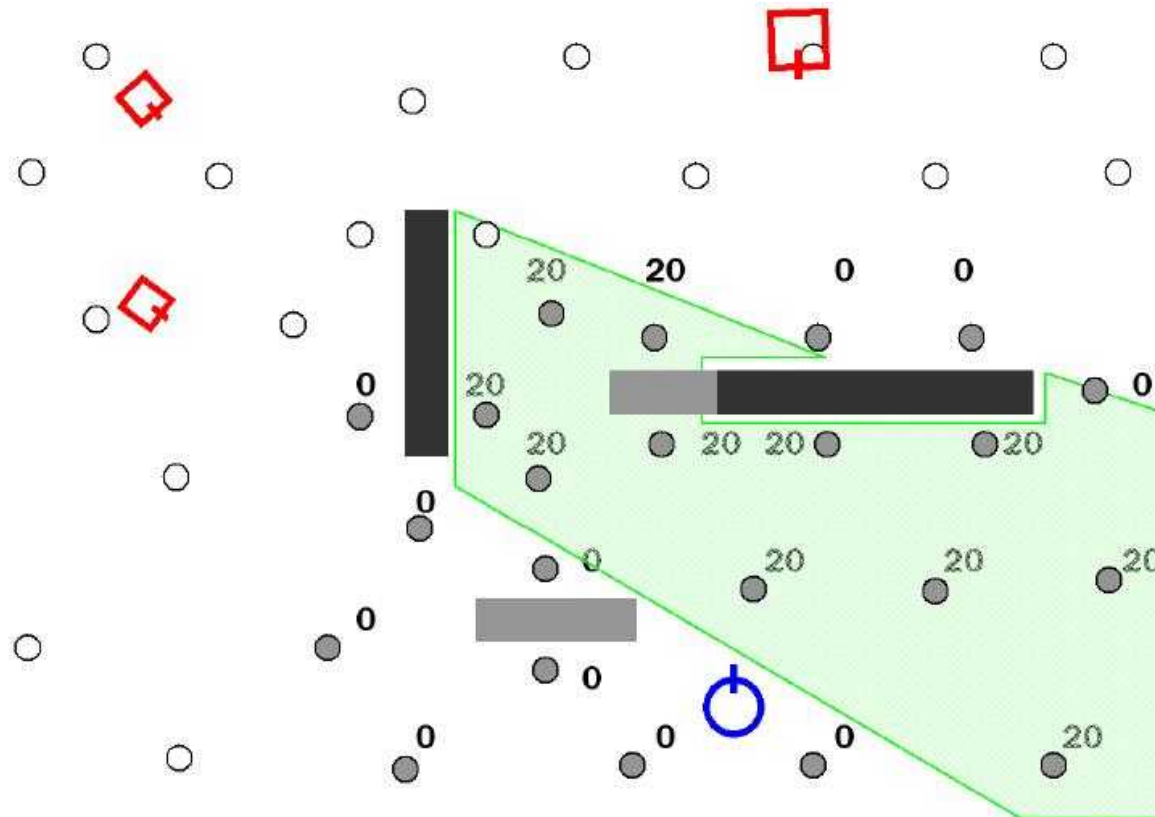


Figure from pages 14 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

## Step4 攻撃最適領域内のポイント进行评估する

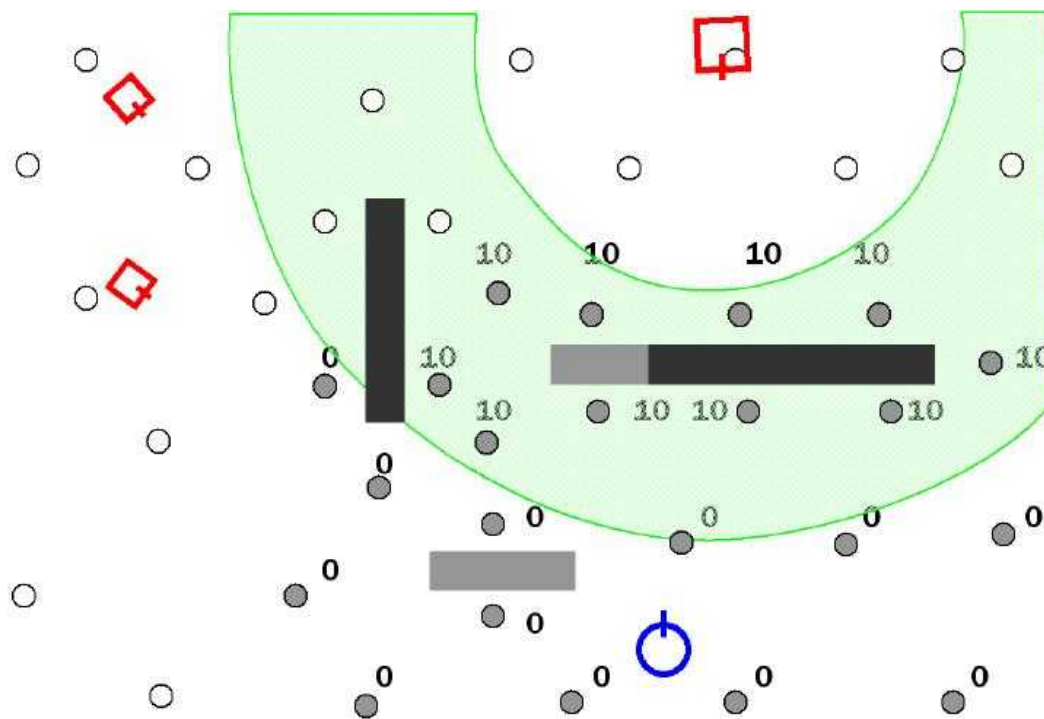


Figure from pages 15 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

## Step5 Step1~4の評価値を重みをかけて足し合わせる

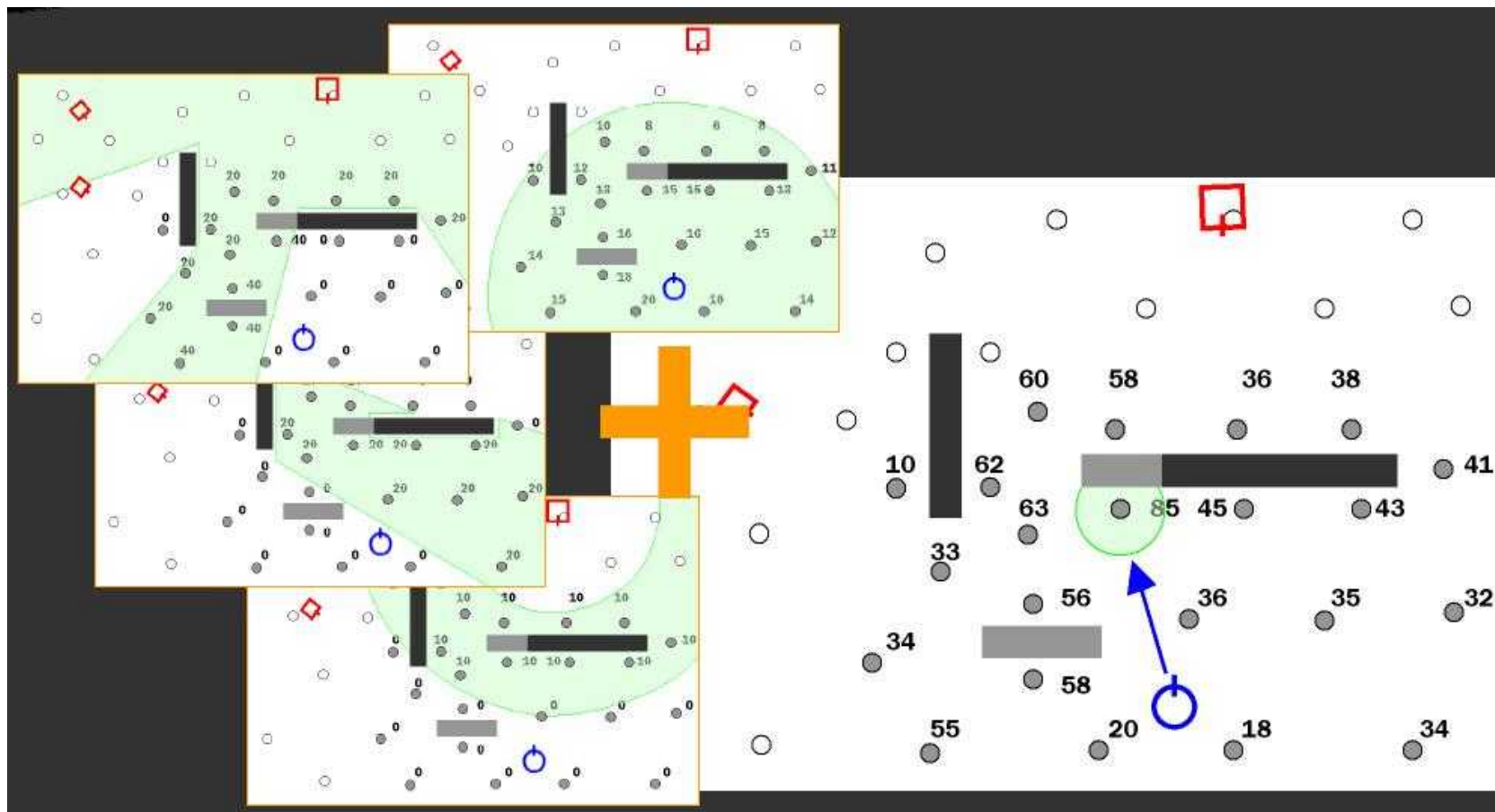


Figure from pages 16 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

## 動作例 2

敵が隠れている場所から移動しそうな場所  
に手榴弾を投げる

## (2) 敵が隠れている場所から移動しそうな場所に手榴弾を投げる

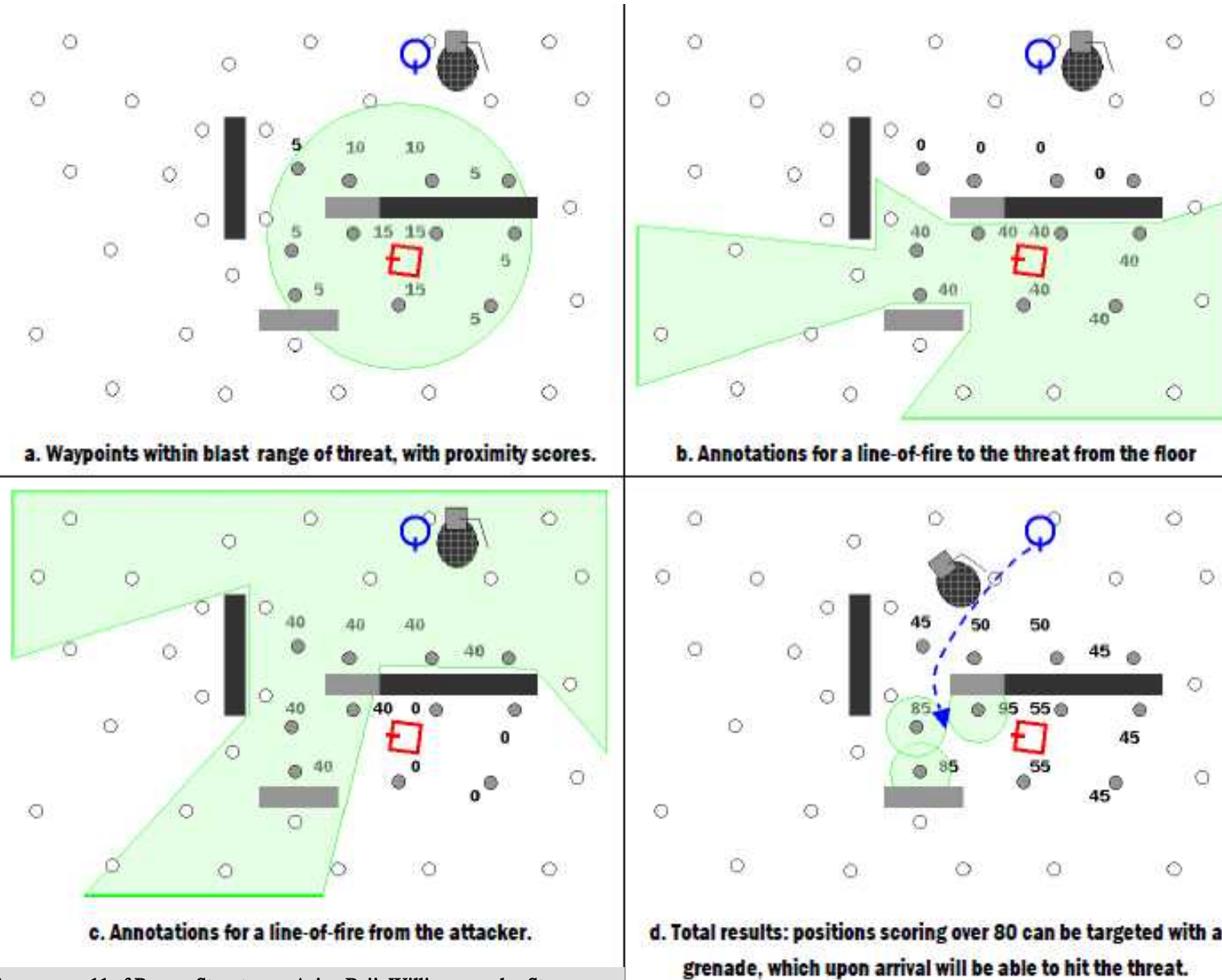


Figure from pages 11 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics" [3]

## 動作例 3

隠れている敵に対して威嚇射撃を行う

### (3) 隠れている敵に対して威嚇射撃を行う

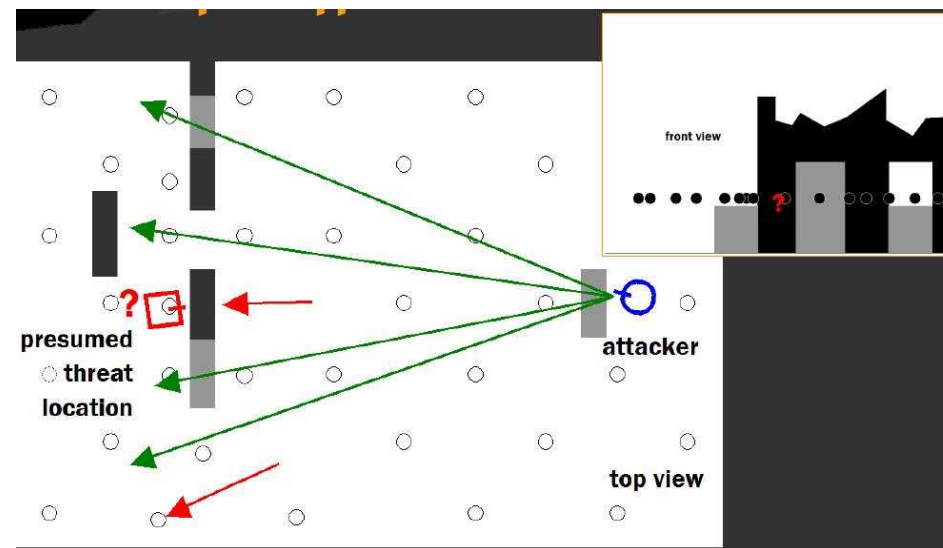
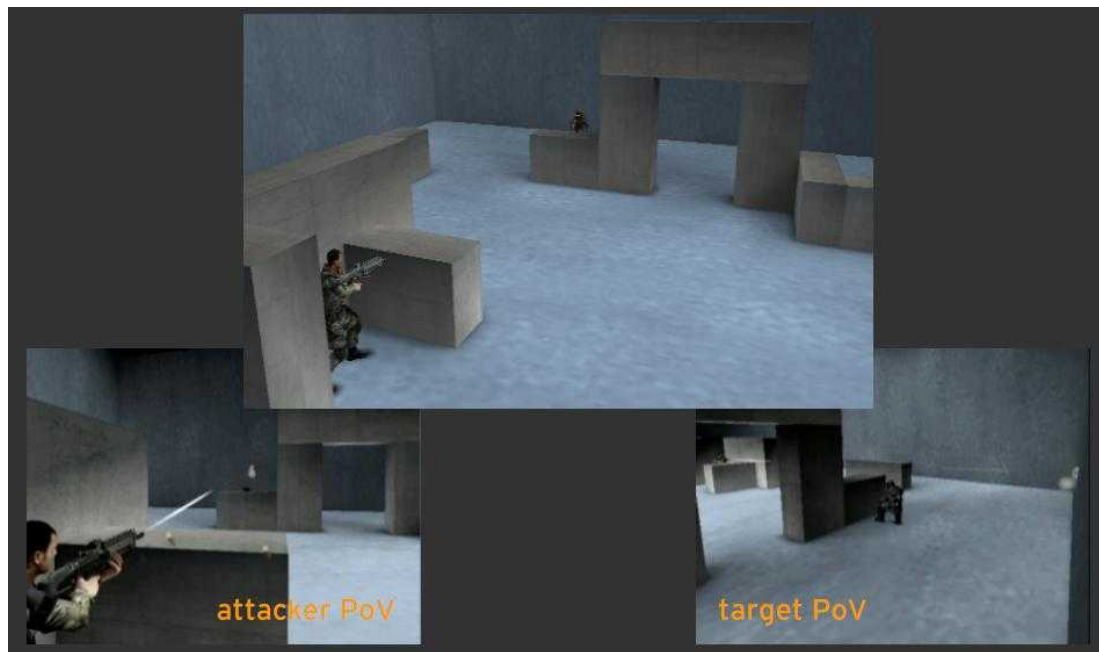


Figure from pages 27,34 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

## Step1 威嚇する敵一定半径内のポイントを候補に上げる

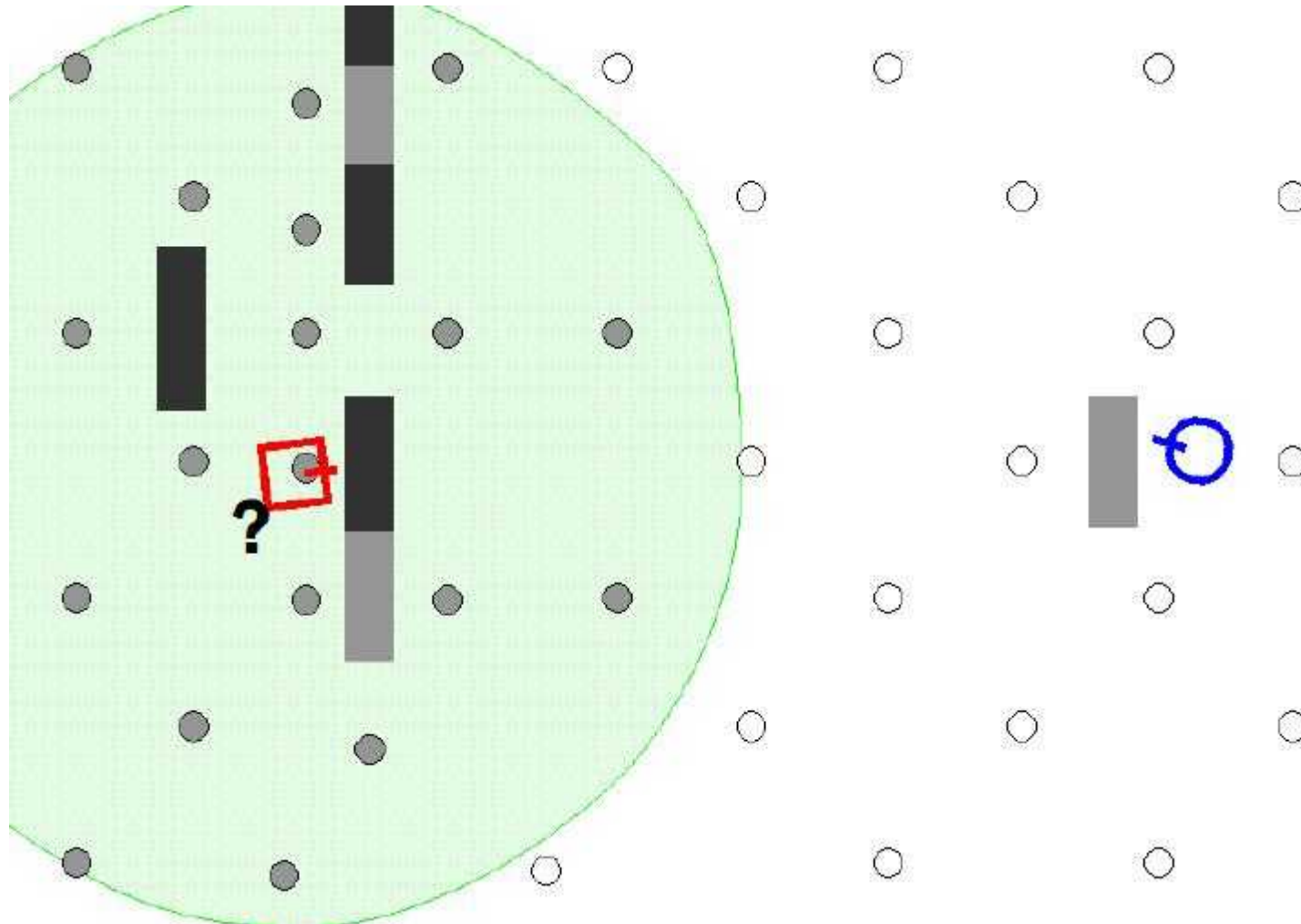


Figure from pages 28 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]



## Step2 自分 (COM) の射線が通るポイント进行评估する

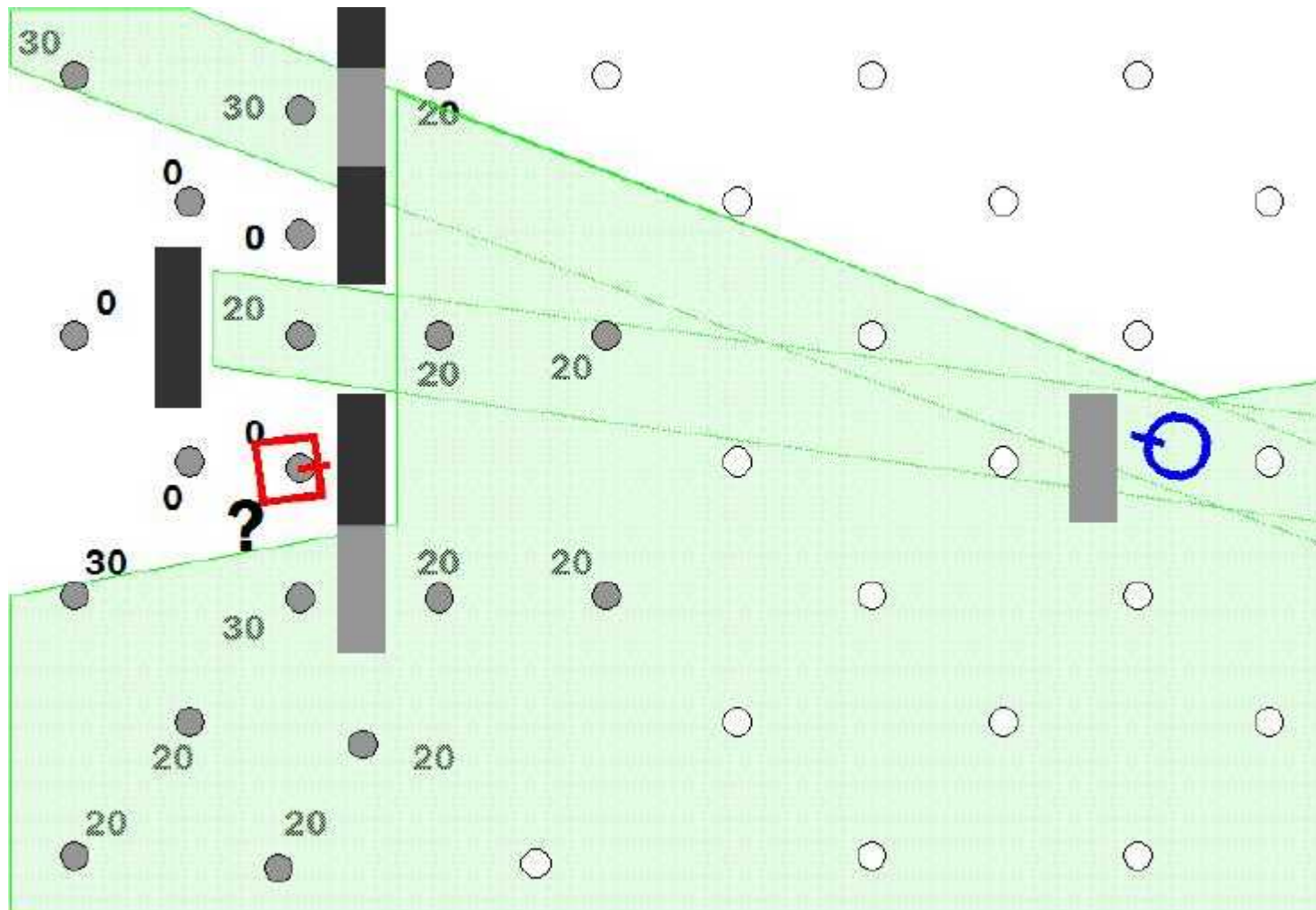


Figure from pages 29 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

## Step3 COMの方向と逆に敵が遮蔽物が側にあるポイント进行评估する

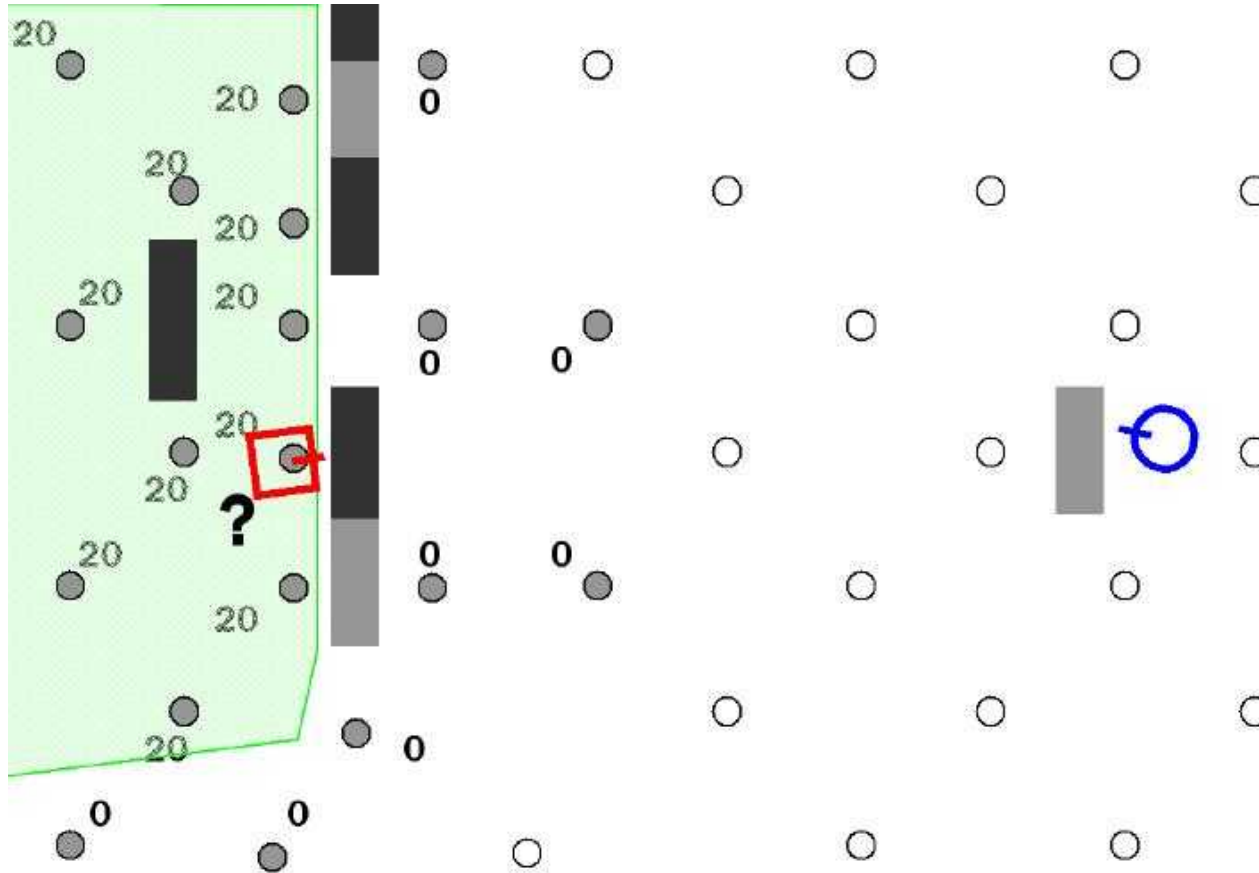


Figure from pages 30 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

Step4 Step1~Step3 を総合して、威嚇ポイントの候補を上げ、  
COMから見て同じ立体角にあるものはまとめる

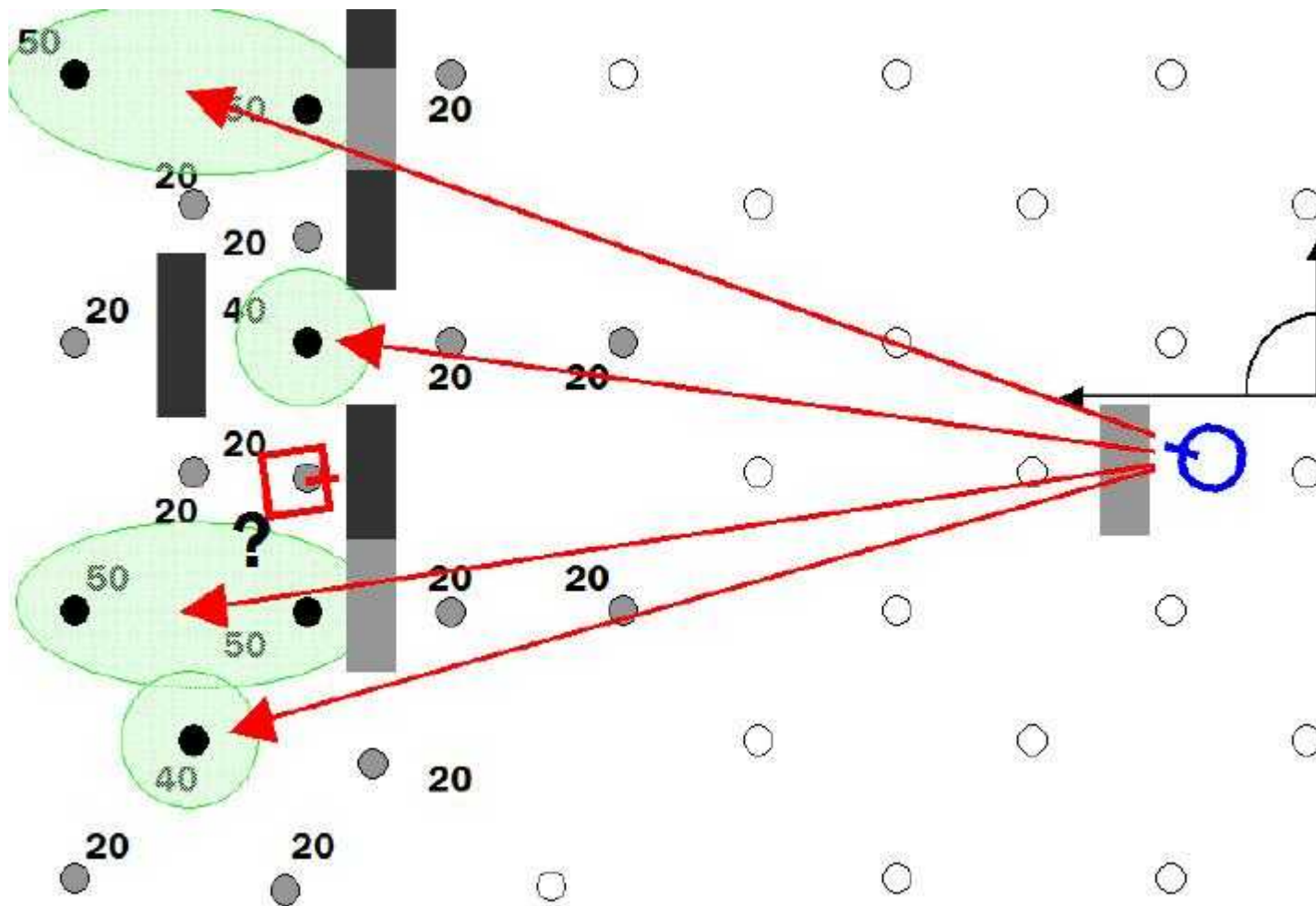


Figure from pages 32 of Arjen Beij, William van der Sterren, "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)" [4]

## 第3章 まとめ

- (1) Killzone は、事前計算によって8方向に対する可視距離を計算し、ゲーム内における視線、射線計算を簡単な計算に変換した。
- (2) 視線、射線計算の負荷を軽くしたおかげで、戦術的位置検出のシステムの実現に成功した。
- (3) 結果として、Killzone は、広大で多様なマップに適応したAIを実現し、COMに高度な行動を与えることに成功した。

# 質疑応答

# 第4章 発展

# 第1部 ゲームAI 技術解説(60分)

1. 技術テーマとゲーム説明 (15分)
2. 静的位置検出 (Killzone の先行研究) (15分)
3. 動的位置検出 (Killzone's AI) (20分)
4. 発展 (10分)

各章の終わりに質疑応答を行います。

# 第4章 発展

ゲーム世界をAIの目的に応じて適切に表現してあげる必要がある。  
( 知識表現(knowledge representation)、世界表現(World representation) )



Killzone



8方向に対する可視距離



他のゲーム



?



自分の開発するゲーム



?



# 第4章 発展

ゲーム世界をAIの目的に応じて適切に表現してあげる必要がある。  
( 知識表現(knowledge representation)、世界表現(World representation) )



Killzone



8方向に対する可視距離



Halo 2



戦術的位置



自分の開発するゲーム



?

# Halo2 における世界表現

The Halo place representation:

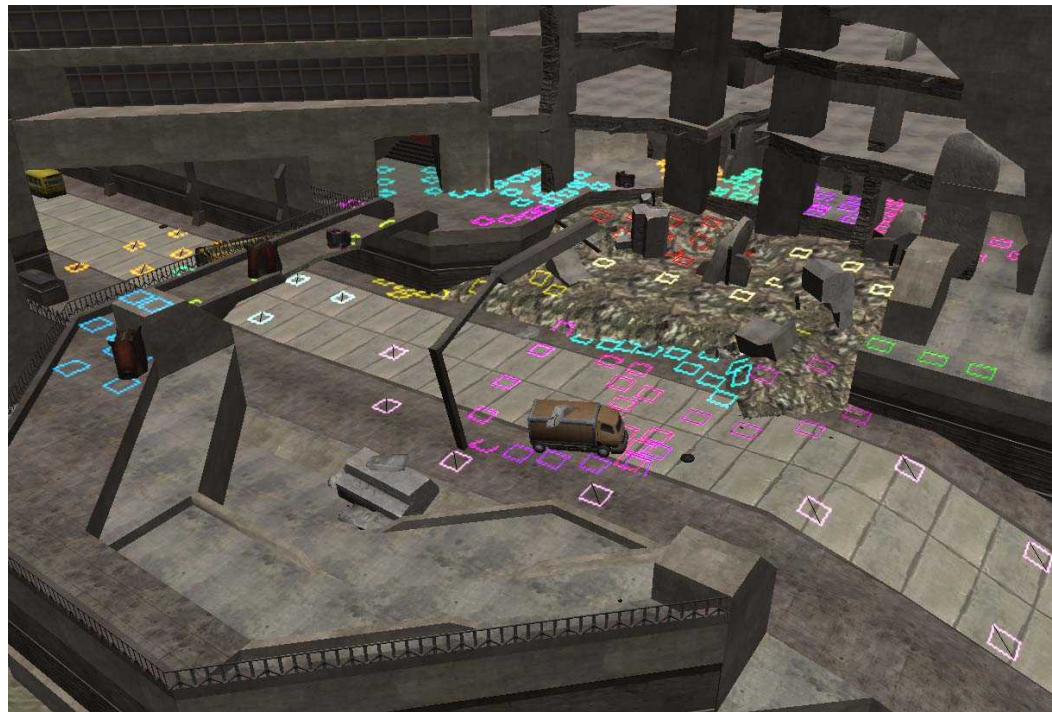


Figure from pages 15 of Damian Isla, "Dude, where's my Warthog? From Pathfinding to General Spatial Competence" [5]

Zones → Areas → Positions

# Halo2 における世界表現

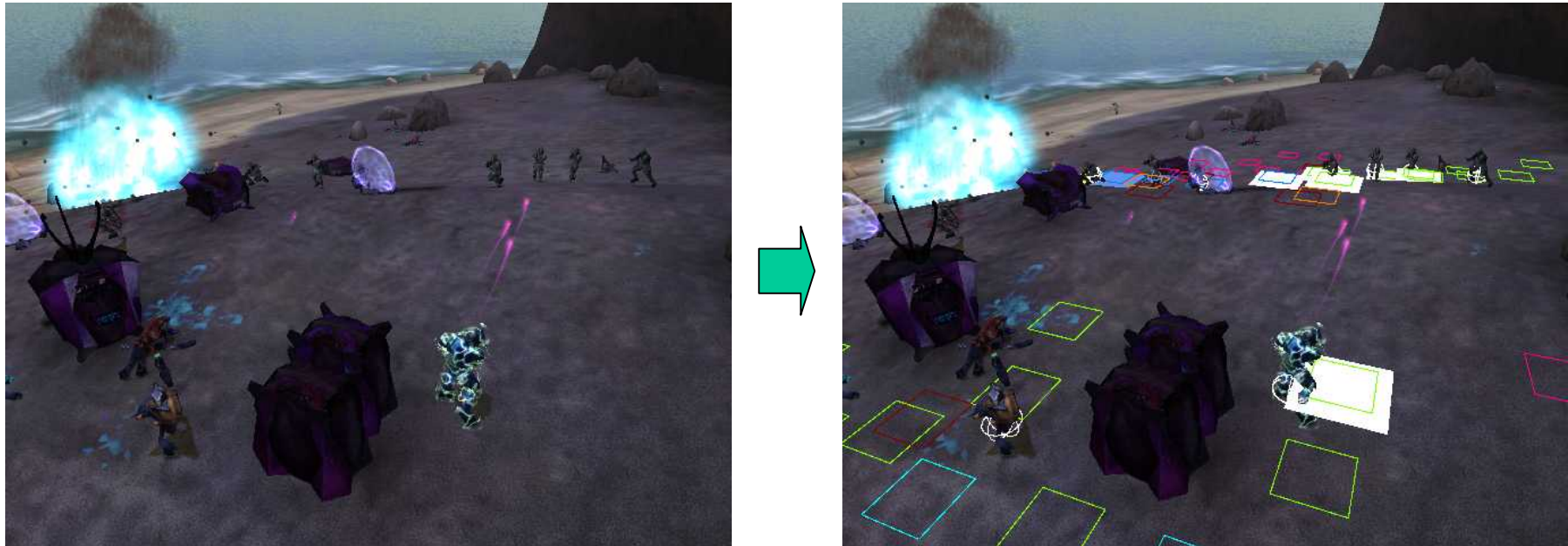


Figure from pages 13,14 of Chris Butcher,Jaime Griesemer,"The Illusion of Intelligence The Integration of AI and Level Design in Halo" [6]

Halo2 (FPS)のCOMは、それほど広い範囲は動かないが、  
限定されたエリアの中でユーザーから見て賢いと思える動きをする



# Halo2 における世界表現



Figure from pages 15 of Damian Isla, "Dude, where's my Warthog? From Pathfinding to General Spatial Competence" [5]

# Halo 2における世界表現



Figure from pages 15 of Damian Isla, "Dude, where's my Warthog? From Pathfinding to General Spatial Competence" [5]

# 第4章 発展

ゲーム世界をAIの目的に応じて適切に表現してあげる必要がある。  
( 知識表現(knowledge representation)、世界表現(World representation) )



Killzone



8方向に対する可視距離



Halo 2



戦術的位置



Chrome Hounds



地表状態  
トポロジー情報



# Chrome Houndsにおける世界表現



Figure from pages of 三宅 陽一郎(2006), "クロムハウズにおける人工知能開発から見るゲームAIの展望 [7]"

# 情報が埋め込まれたナビゲーションメッシュ

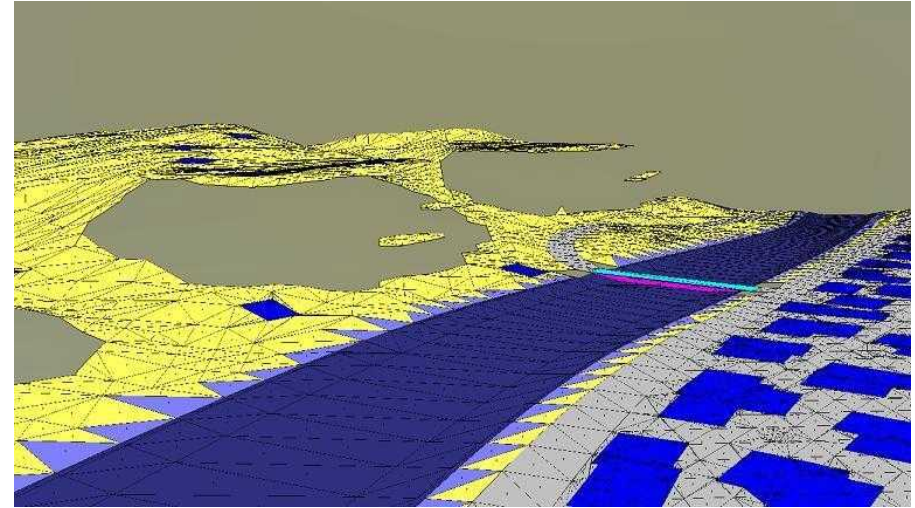


Figure from pages 90 of 三宅 陽一郎(2006), "クロムハウズにおける人工知能開発から見るゲームAIの展望 [7]"

- (1) 水や砂地は、ハウズのスピードを減速させるので、メッシュに表面の性質を埋め込んでおく



最短時間の経路を導く

ハウズが地表効果を考慮して移動する

- (2) 障害物が破壊されたら、メッシュのデータを更新する



ハウズが状況の変化に対応して移動する



# まとめ

- (1) 世界表現という視点からゲームを見ると、ゲームタイトルにおけるゲームAIの工夫が見えて来る。
- (2) そのような工夫を学んで行くことで、自分自身の開発するゲームにも応用することができる。

# 質疑応答

# 本講演の構成

はじめに

ゲームAI連続セミナーのご紹介

## 第1部 ゲームAI 技術解説 (60分)

第1章 技術テーマとゲーム説明 (15分)

第2章 静的位置検出 (Killzone の先行研究) (15分)

第3章 動的位置検出 (Killzone's AI) (20分)

第4章 発展 (10分)

各章の終わりに質疑応答を行います。

## 第2部 ディスカッション (65分)

(AI) 一つのテーマについて15～20分を予定。

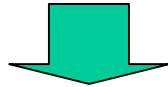
テーマはこちらが準備したもの、参加者からテーマを募集(60分)

(セミナー) 今日の感想とセミナーの改善点を討論する(5分)

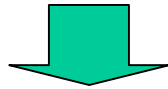
# ディスカッション

# ディスカッションの狙い

今日のゲーム開発期間 1, 2年 ~ 4, 5年



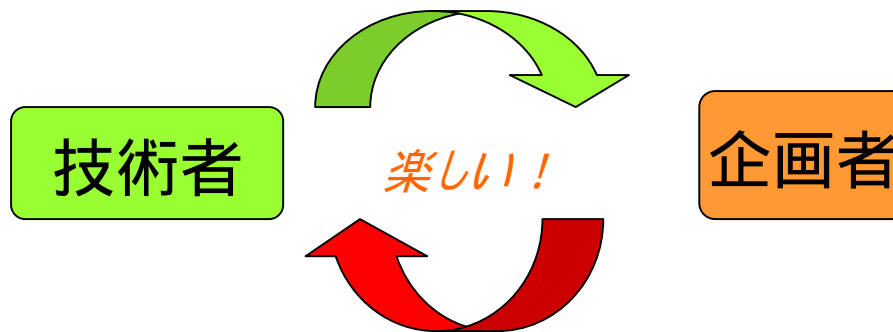
ゲームAIの基本設計をする機会 = 1/1 year ~ 1/5 year



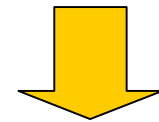
ディスカッション = ゲームAIの導入を想定して模擬的にゲームを作る

新しい技術的  
挑戦

こういことができますよ！



新しい技術込みの  
ゲームデザイン



WEBで公開

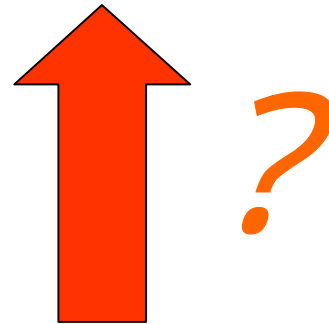
こういうアイデアがあるけどできるか？

閉塞したゲームスタイルを打ち破る新しいゲームデザインの創造

# ディスカッションの道筋

紹介されたゲームAI技術を理解し、それを応用、発展させ、  
これからのゲーム開発へつなげる。

自分がこれから  
開発するゲーム



Killzone の  
AIシステム

# そのためには何をすればいいか？

(1) そのままでは使えない

抽象化(パターンを抜き出す)

(2) 使うための制限がある

制限を取り払う

(3) Killzoneとは条件が違う(日本と外国、PS2とPS3、チームプレイ)

発展させる

(4) どうやって作ればいいのか？

工程を考える

# (1) そのままでは使えない

→ **抽象化** (パターンを抜き出す。パターンランゲージの方法)

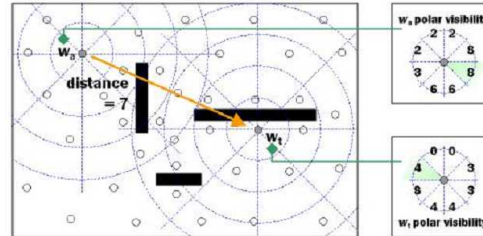


Figure from pages 9 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics" [3]

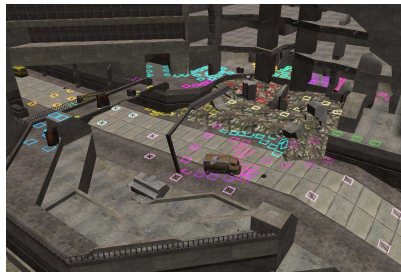


Figure from pages 15 of Damian Isla, "Dude, where's my Warthog? From Pathfinding to General Spatial Competence" [5]

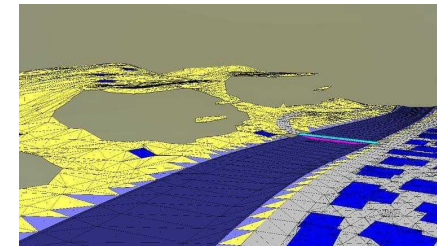


Figure from pages 90 of 三宅 陽一郎(2006), "クロムハウন্ズにおける人工知能開発から見るゲームAIの展望" [7]

## パターン1

AIに能力を持たせるには、たとえそれが同じ対象であっても、その問題の種類に応じた表現のデータを作成しなければならない。

自分の作るCOMにさせたい行動、目的、問題に応じてゲーム世界を表現してやる



## (例) Halo2

アフォーダンス表現 = COMにその世界で行うことが出来る行動



Figure from pages 22 of Damian Isla, "Dude, where's my Warthog? From Pathfinding to General Spatial Competence" [5]

Halo 2ではオブジェクトに対して「できること」の情報が埋め込まれている

# (1) そのままでは使えない

→ **抽象化** (パターンを抜き出す。パターンランゲージの方法)

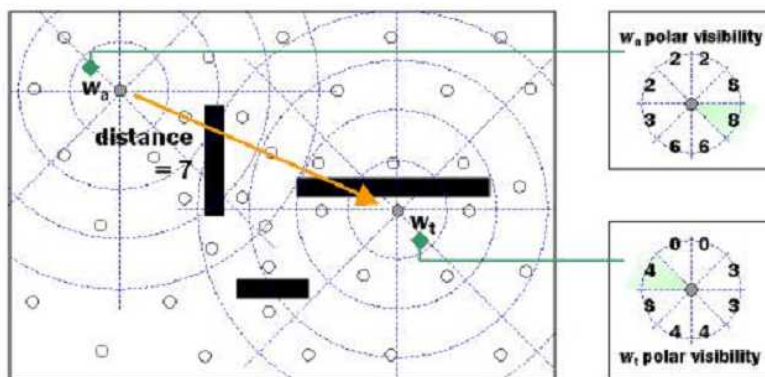


Table 32 Kbyte  
For 2000 Waypoints

Figure from pages 9 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics" [3]

## パターン2

地形解析など計算量の多い解析は、開発中に事前にデータとして用意しておくことで、ゲーム中の負荷を軽減することが出来る。

レベルデザインの段階からCOMにさせたいことを考えて事前にデータを用意しよう

(例) Halo2, Gears of war

Gears of war (Xbox360) 紹介画像(省略、以下参照)

<http://www.xbox.com/ja-JP/games/g/gearsofwar/>

レベルデザインとAIが最初から統一されてデザインされている例

Gears of war (Xbox360)<sub>15</sub>

(1) そのままでは使えない

→ 抽象化 (パターンを抜き出す。パターンランゲージの方法)

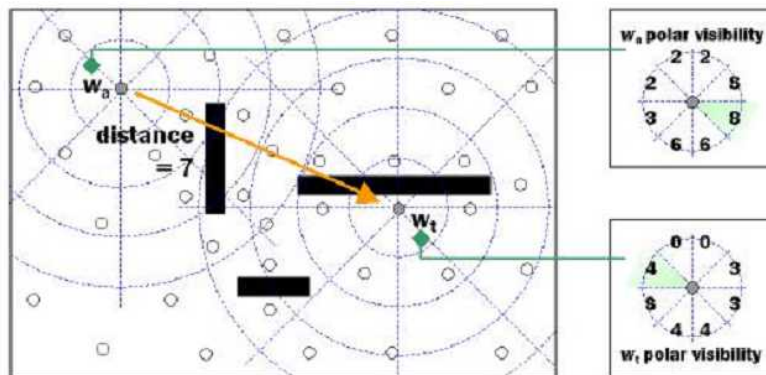


Table 32 Kbyte  
For 2000 Waypoints

Figure from pages 9 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics" [3]

### パターン3

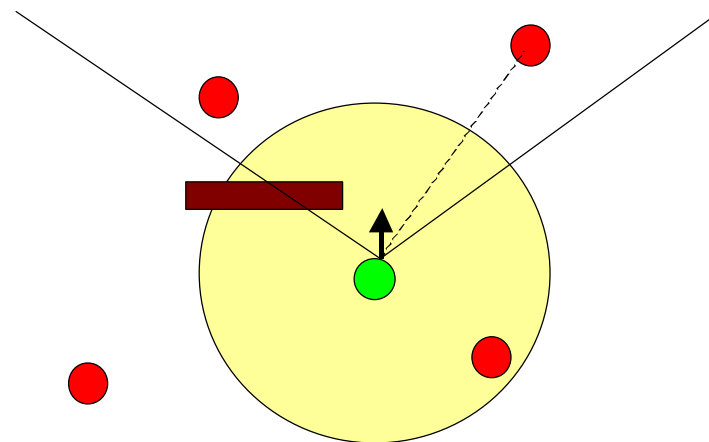
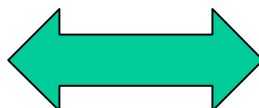
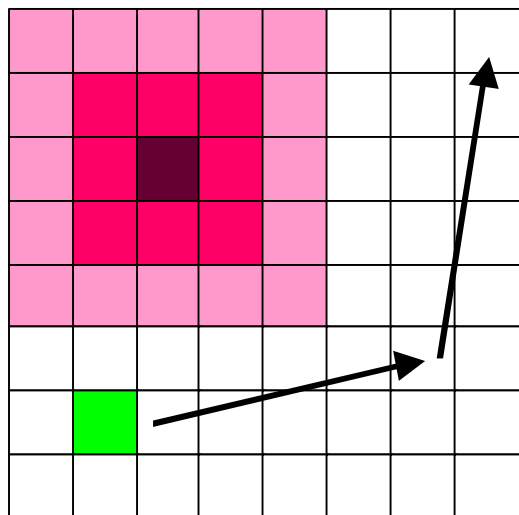
地形解析を行う場合に、マップモデルから行うのではなく、ウェイポイントを置いて、ウェイポイントの間だけで解析を行い、必要な分解能で地形の性質を抽出する。

地形解析において、ウェイポイントの間隔の精度で情報を抽出する

Killzone 特有

# (1) そのままでは使えない

→ **抽象化** (パターンを抜き出す。パターンランゲージの方法)



## パターン4

キャラクター同士のインタラクションを場を介して行うことで、  
地形情報とキャラクターたちの関係から行動を決定する。

認識のモデルをどう設計するか考えよう。

(例) インフルエンス・マップ

# 議題その(1)

- (1) 今までプレイして来たゲームのAIでこういった世界表現を取り入れているゲームは何か？
- (2) 自分の作って来たゲームで、この方向で開発したAIがあるか？
- (3) このシステムを応用して、これからどんなゲームが作れるだろうか？

# そのためには何をすればいいか？

(1) そのままでは使えない

抽象化(パターンを抜き出す)

(2) 使うための制限がある

制限を取り払う

(3) Killzoneとは条件が違う(日本と外国、PS2とPS3、チームプレイ)

発展させる

(4) どうやって作ればいいのか？

工程を考える

(2) 使うための制限がある

→ 制限を取り払う

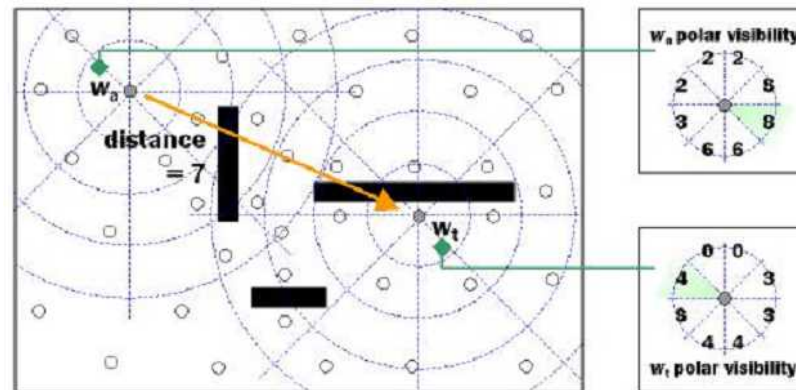


Figure from pages 9 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics" [3]

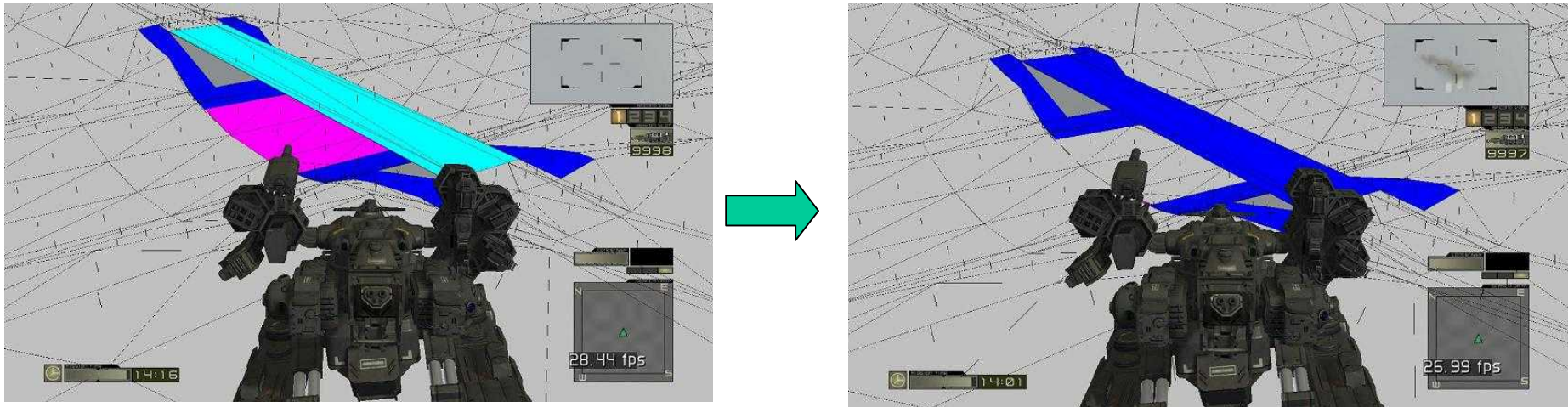
Killzone では、遮蔽に使えるようなオブジェクトは壊れることはない。

→ リアルタイムに情報を更新する

(例) クロムハウنز



# (例) クロムハウنز



## リアルタイムに変更

橋のメッシュは水色、橋に頭を打って通れない橋の下のメッシュはピンク色。  
青色は通れないメッシュ(ここでは左図では橋げたが青色になっている)

ハウنزが橋を壊すと、橋は通れないので青色になり、橋の下は橋がなくなって通れるようになるので、白(通行可能)になる。

## 議題その(2)

「遮蔽に使えるようなオブジェクトは壊れることはない」  
という条件はこれからのゲームでは大きなディスアドバンテージだと考えられる。



では、この条件を取り除くことは技術的に可能だろうか？

可能であるとして、それをゲームデザインにどう活かすことができるだろうか？

# そのためには何をすればいいか？

(1) そのままでは使えない

抽象化(パターンを抜き出す)

(2) 使うための制限がある

制限を取り払う

(3) Killzoneとは条件が違う(日本と外国、PS2とPS3、チームプレイ)

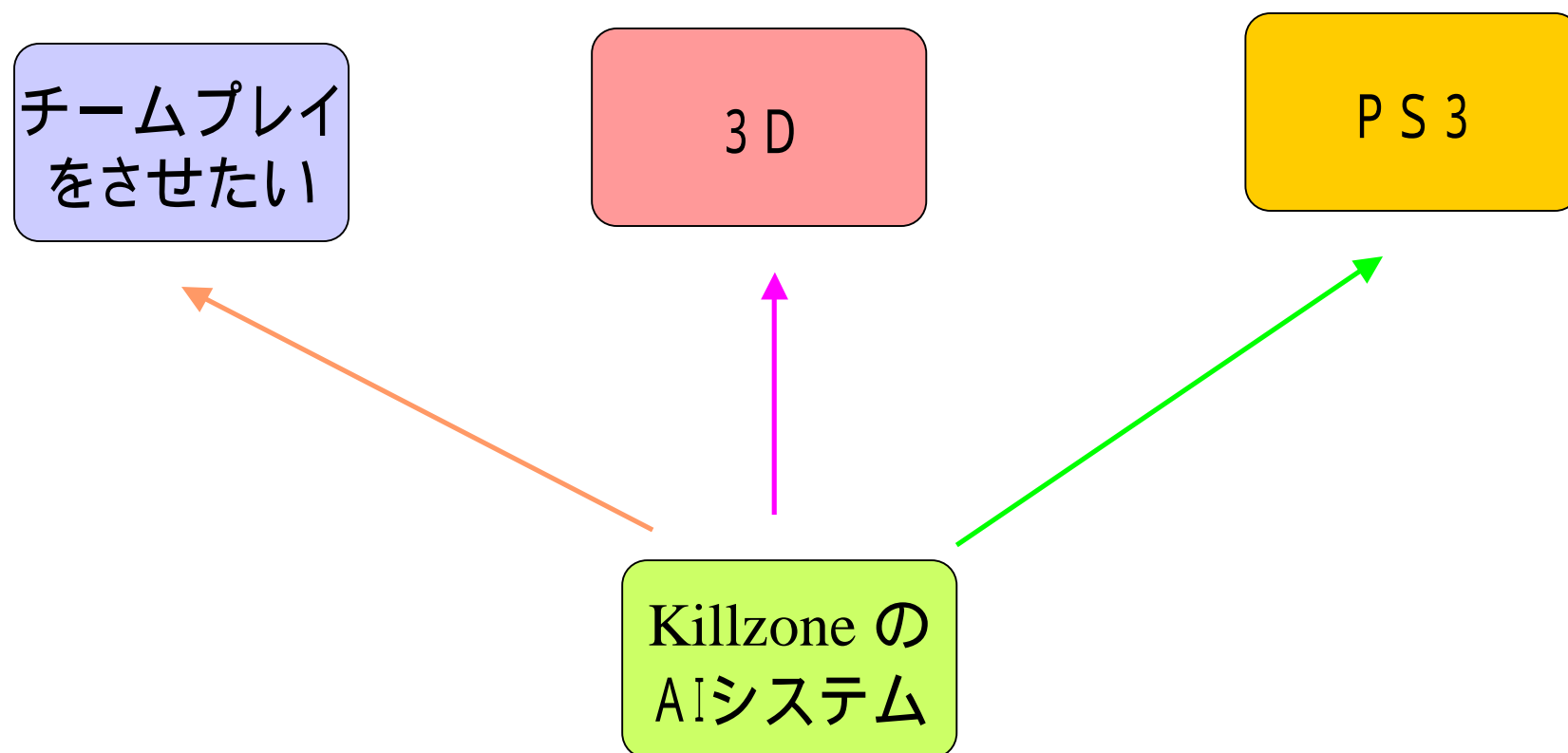
発展させる

(4) どうやって作ればいいのか？

工程を考える

(3) 自分が作りたいゲームはKillzoneとは条件が違

→ 発展させる



議題3 Killzone からどのような方向への発展が可能だろうか？

# そのためには何をすればいいか？

(1) そのままでは使えない

抽象化(パターンを抜き出す)

(2) 使うための制限がある

制限を取り払う

(3) Killzoneとは条件が違う(日本と外国、PS2とPS3、チームプレイ)

発展させる

(4) どうやって作ればいいのか？

作業工程とデバッグ方法を考える

(4) どうやって作ればいいのか？

→ 作業工程とデバッグ方法を考える

(1) ウェイポイントはレベルデザイナーが置き、  
テストステージでAIをデバッグ調整する。

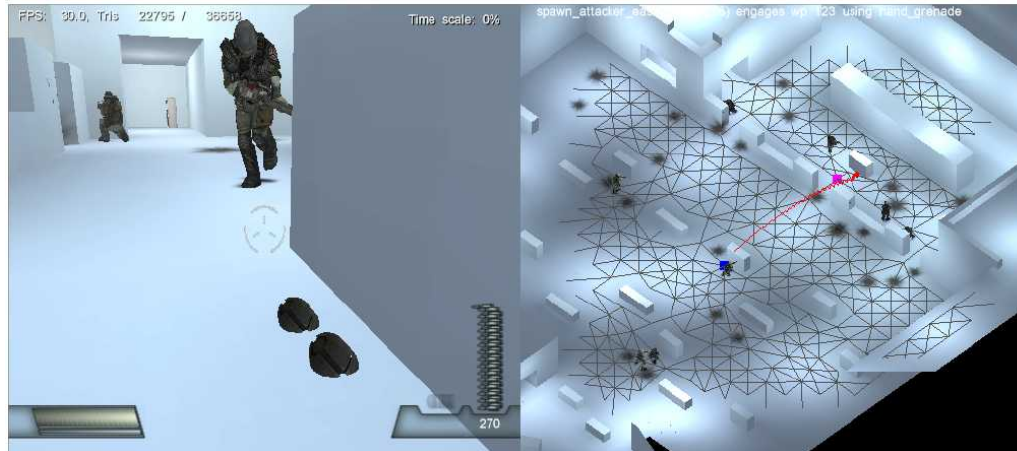


Figure from pages 14 of Remco Straatman, Arjen Beij, William van der Sterren (2005), "Killzone's AI : Dynamic Procedural Combat Tactics" [3]

(2) 新しい技術を導入するためには、新しいデバッグ方法が必要。

(例) パス検索

# (例) パス検索

固定パス ... そのパスに一度COMをあるか歩かせればよい

パス検索システム... マップ上のどの点でもパス検索の結果を得られれば移動できる

パス検索の結果が返って来ることと、  
実際に歩くことができることは違う。

- (1) 崖から落ちる
- (2) 速度がつくと急に止まらない
- (3) 機体の一部がオブジェクトに引っ掛かる

## (例) パス検索

Killzone ... 崖に対しては、機体は落ちることができないように、壁が作ってある。(物理的に不自然)

Chrome Hounds (ゲーム仕様)

... いつ如何なるときも、崖に落ちることができる。

- 解決策
- (1) 主要ルートは徹底的にチェックする。
  - (2) 崖、壁に隣接するメッシュにはフラグをつけて、パス検索では、そこに近寄らないパスを導く。
  - (3) 加速度から落下判定をし、落下した場合は、その地点から再パス検索を行う。

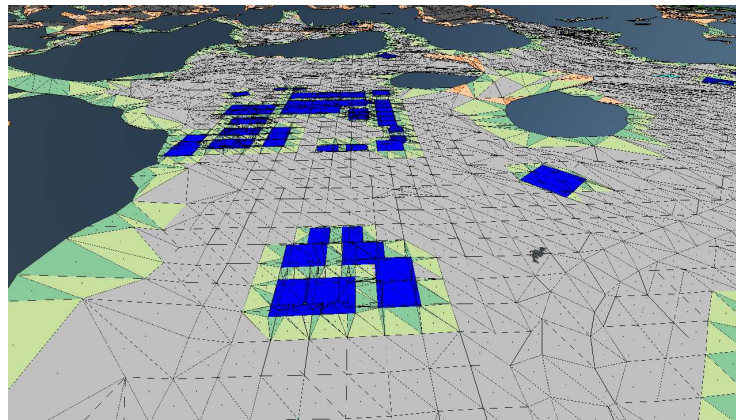


Figure from pages 90 of 三宅 陽一郎(2006),  
"クロムハウন্ズにおける人工知能開発から見るゲームAIの展望 [7]"



## 議題その(4)

Killzone の AIは、どのように調整をし、デバッグをしているか？

# そのためには何をすればいいか？

(1) そのままでは使えない

抽象化(パターンを抜き出す)

(2) 使うための制限がある

制限を取り払う

(3) Killzoneとは条件が違う(日本と外国、PS2とPS3、チームプレイ)

発展させる

(4) どうやって作ればいいのか？

作業工程とデバッグ方法を考える

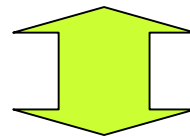
(5) フリーディスカッション

# 欧米と日本のゲームにおけるAIの役割の違い

欧米    リアル指向

フィールド単位

ステージがあって、その中で自由に動き回る



日本    演出型

シーン単位

物語の中の役割を果たす役割

# 本講演の構成

はじめに

ゲームAI連続セミナーのご紹介

## 第1部 ゲームAI 技術解説 (60分)

第1章 技術テーマとゲーム説明 (15分)

第2章 静的位置検出 (Killzone の先行研究) (15分)

第3章 動的位置検出 (Killzone's AI) (20分)

第4章 発展 (10分)

各章の終わりに質疑応答を行います。

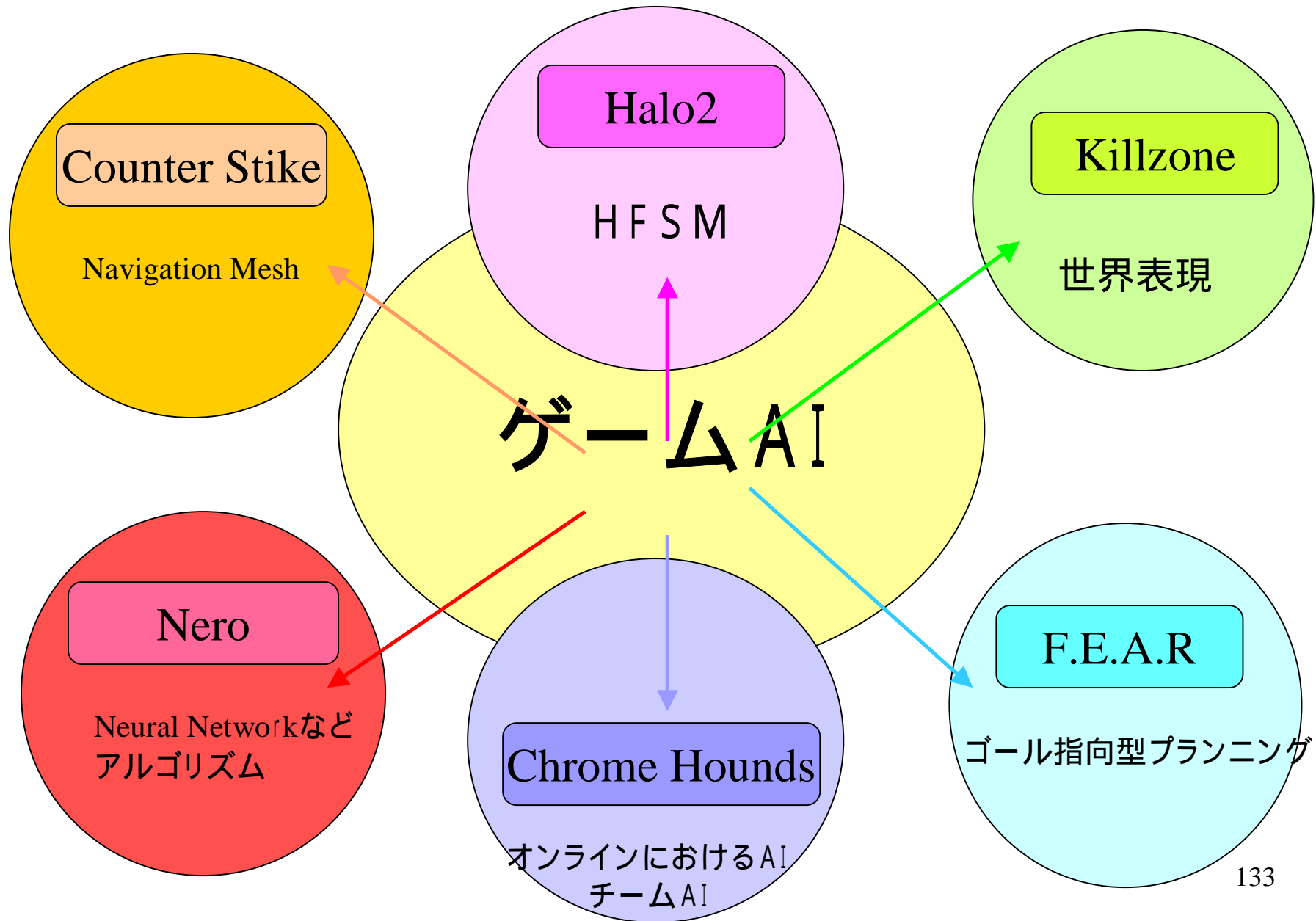
## 第2部 ディスカッション (65分)

(AI)一つのテーマについて15～20分を予定。

テーマはこちらが準備したもの、参加者からテーマを募集(60分)

(セミナー)今日の感想とセミナーの改善点を討論する(5分)

# これからのセミナーの展開



ご清聴ありがとうございました。

これ以外に、意見や質問があれば、  
メールかアンケートへ

**y\_miyake@fromsoftware.co.jp**

(IGDA Japan登録アドレス yoichi-m@pk9.so-net.ne.jp )

WEB上の意見交換にはIGDA Japanのサイトをご利用ください

<http://www.igda.jp>

# ゲームAI連続セミナー 第2回

日程 2月10日

場所 東京大学本郷キャンパス工学部新2号館

テーマ 「F.E.A.R. における  
ゴール指向型アクションプランニング」

F.E.A.R. 紹介 画像(省略、以下参照)

- <http://www.4gamer.net/news.php?url=/DataContents/game/1610.html>
- <http://www.lith.com>

長い時間に渡って自分の行動をプランニング(計画)するCOM

# ゲームAI連続セミナー 第2回 参考文献

- 論文 Orkin, J. (2006), [3 States & a Plan: The AI of F.E.A.R.](#), Game Developer's Conference Proceedings.
- 参考文献(I) Mat Buckland, “Programming Game AI by Example”, Chapter 9, WORDWARE publishing  
(第9章とそのサンプルコードはゴール指向型プランニングの優れた解説です。  
教科書をお探しの方は、ゲームAIについて最良の書の一つです。推薦します。)
- 参考文献(II) Jeff Orkin' HP  
<http://web.media.mit.edu/~jorkin/>  
(Jeff Orkin は、米におけるゲームAIにおけるゴール指向型プランニングの推進者の一人。著者のサイトに豊富な情報があります。)
- 参考文献(III) 星野 瑠美子, “F.E.A.R.のAI - 3つの状態とゴール指向プランニングシステム”  
[http://www.igda.jp/modules/xeblog/?action\\_xeblog\\_details=1&blog\\_id=62](http://www.igda.jp/modules/xeblog/?action_xeblog_details=1&blog_id=62)  
(IGDA Japan サイト内、GDC2006講演の紹介)