

IGDA日本

# 「GDC 2011 ゲームAI分野レポート」



三宅 陽一郎  
(IGDA日本ゲームAI専門部会代表)

@miyayou

<https://www.facebook.com/youichiro.miyake>

[y.m.4160@gmail.com](mailto:y.m.4160@gmail.com)

<http://blogai.igda.jp>

<http://www.linkedin.com/in/miyayou>

2011.4.16

# GDC2011 @miyayou

2.28

Modular  
Component  
AI System

AI PrOn:  
AI Debug

AI  
Navigation

Smart AI  
through  
Influence Map

Staffing the  
Extras

AI  
Developers  
Rant !

3.1

People in  
your pocket

Using  
Randomness

Finding your  
way with  
Havok AI

Kynapse AI

Inside  
Darkspore  
and The Sims:  
Medieval

Scripting  
and AI

3.2

KEYNOTE

The animation  
of Halo:Reach

Multicore Memory  
Management in  
MORTALKOMBAT

Elite  
Postmortem

AI & Animation in  
ASSASIN'S CREED:  
BROTHERHOOD

3.3

Navigation in  
Insomniac Engine

Biofeedback in  
Gameplay

Crysis 2: Getting  
more Interactivity  
out of Animation-  
Data

DOOM  
Postmortem

Animating NPC's  
In UNCHARTED

3.4

Automated Level of  
Detail Generation  
in HALO:REACH

Real-time Helper  
Joint System

Automated Level of  
Detail Generation  
in HALO:REACH

The SIMS3 Create-  
a-Sim User  
Experience

BUNGRING BAY  
Postmortem

# GDC2011 @miyayou

2.28

3.1

3.2

3.3

3.4

Modular  
Component  
AI System

People in  
your pocket

KEYNOTE

Navigation in  
Insomniac Engine

Automated Level of  
Detail Generation  
in HALO:REACH

AI PrOn:  
AI Debug

Using  
Randomness

The animation  
of Halo:Reach

Biofeedback in  
Gameplay

Real-time Helper  
Joint System

AI  
Navigation

Finding your  
way with  
Havok AI

Multicore Memory  
Management in  
MORTALKOMBAT

Crysis 2: Getting  
more Interactivity  
out of Animation-  
Data

Automated Level of  
Detail Generation  
in HALO:REACH

Smart AI  
through  
Influence Map

Kynapse AI

Elite  
Postmortem

DOOM  
Postmortem

The SIMS3 Create-  
a-Sim User  
Experience

Staffing the  
Extras

Inside  
Darkspore  
and The Sims:  
Medieval

AI & Animation in  
ASSASIN'S CREED:  
BROTHERHOOD

Animating NPC's  
In UNCHARTED

BUNGRING BAY  
Postmortem

AI  
Developers  
Rant !

Scripting  
and AI

AI Summit

# GDCのゲームAI全体の傾向

2011年： ゲームAIの大型から小型化 & 大型AI技術の固定化

*有名AI開発者の独立(Halo3, The Sims 3, Killzone 2...)Darkspore,  
The Sims: Medieval, StarCraft 2, Havok AI, Kynapse, Insomiac PathEngine*

2010年： 大型ゲームAIの完成 & キャラクター制御の高度化

*Uncharted 2, Sprinter Cell:Conviction, Killzone2, The Sims 3, Bioshock 2, FIFA*  
[http://igda.sakura.ne.jp/sblo\\_files/ai-igdajp/GDC2010/YMiyake\\_GDC\\_Report\\_2010\\_4\\_3.pdf](http://igda.sakura.ne.jp/sblo_files/ai-igdajp/GDC2010/YMiyake_GDC_Report_2010_4_3.pdf)

2009年： キャラクターAI(アニメーション&パスファインディング) & メタAI

*Morpheme 2.0, Warhammer Online(Kynapse), Halo Wars, Left 4 Dead*

<http://www.digrajapan.org/modules/mydownloads/images/study/20090411.pdf>

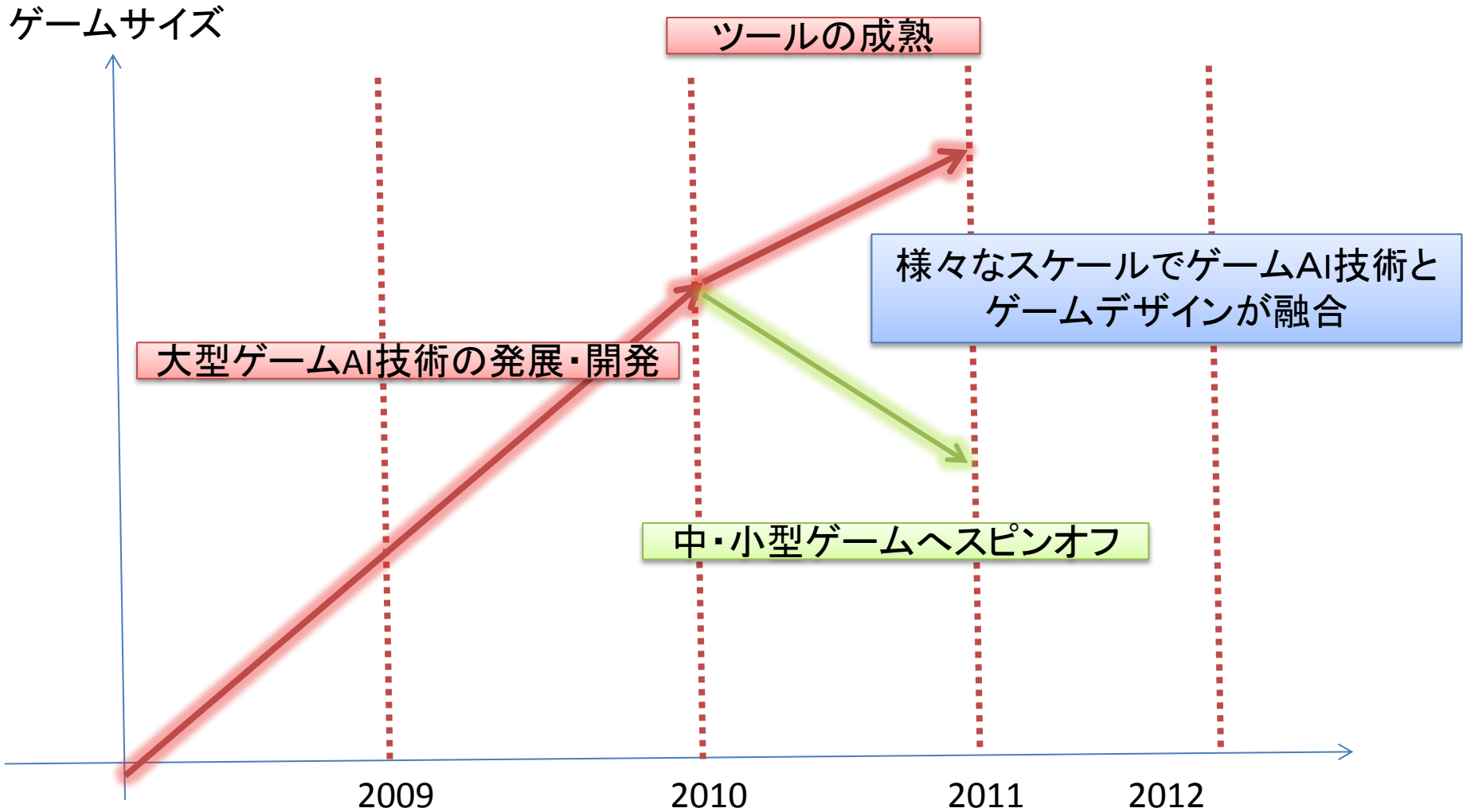
2008年： 大型ゲームAI & プロシージャル

*Spore, Halo3, Assassin's Creed...*

[http://igda.sakura.ne.jp/sblo\\_files/ai-igdajp/AI/IDGA\\_GDC08\\_Miyake\\_public.pdf](http://igda.sakura.ne.jp/sblo_files/ai-igdajp/AI/IDGA_GDC08_Miyake_public.pdf)

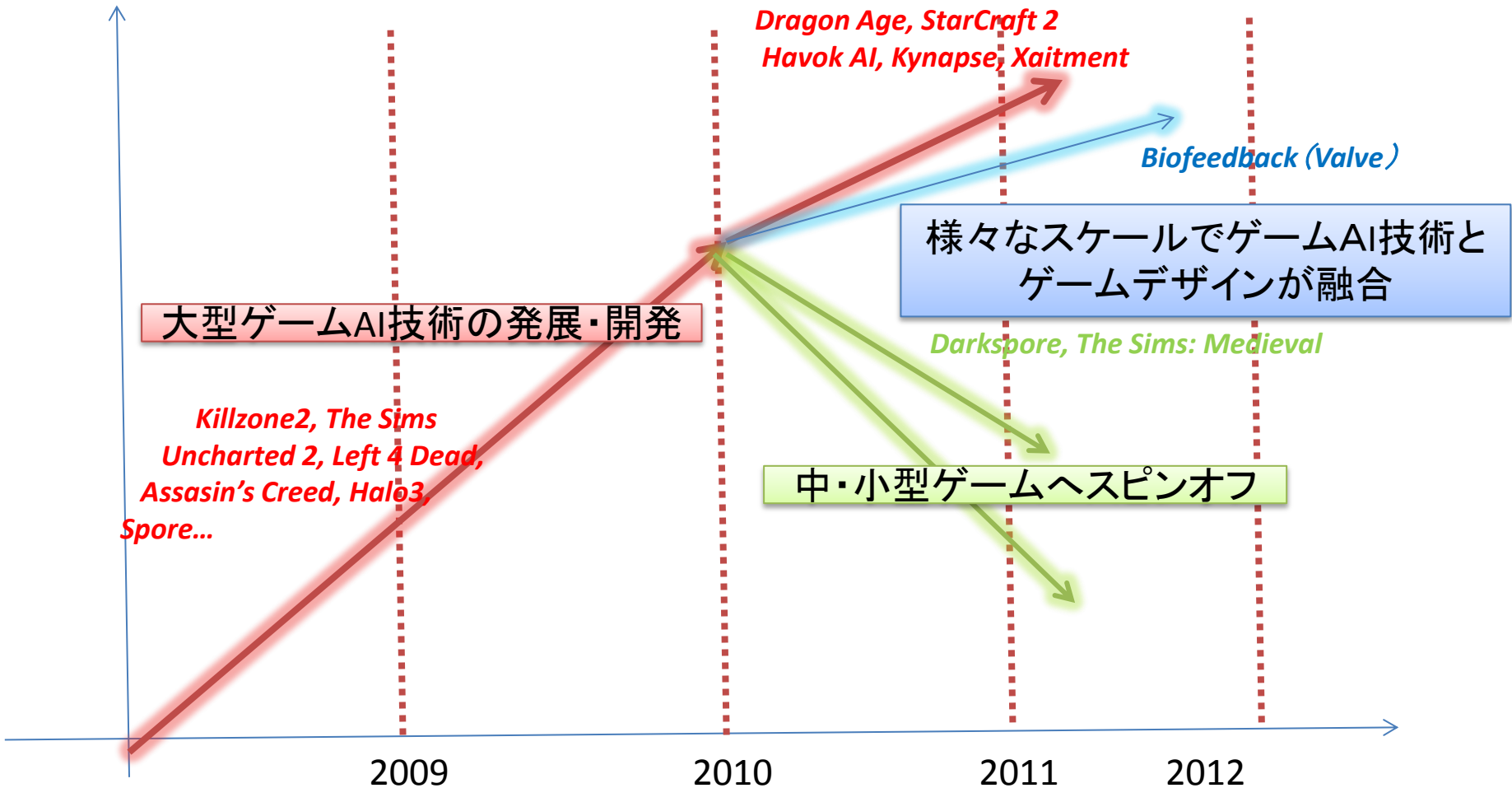


# GDC2011のゲームAI全体の傾向



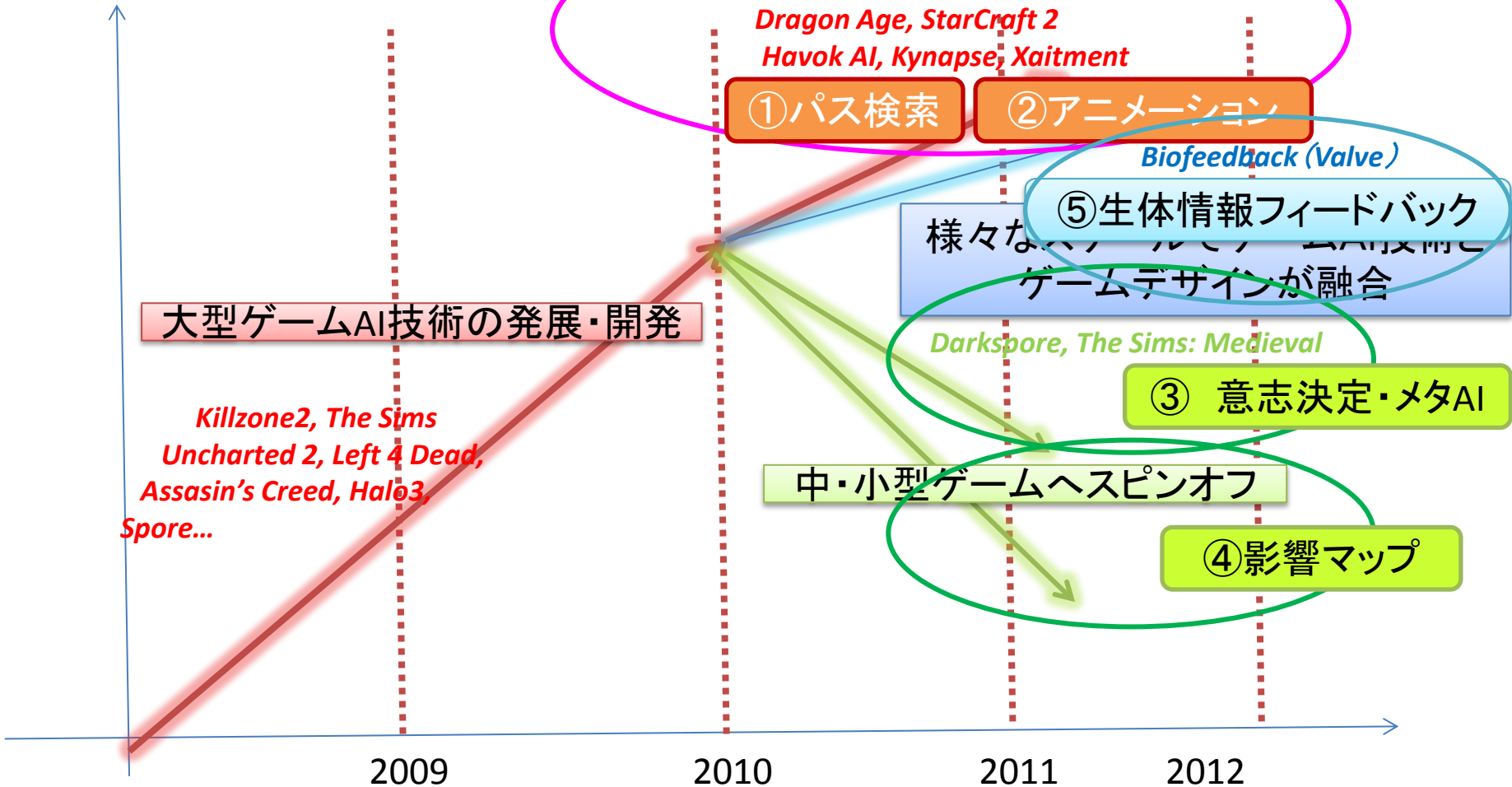
# GDC2011のゲームAI全体の傾向

ゲームサイズ



# GDC2011のゲームAI全体の傾向

ゲームサイズ



# 目次

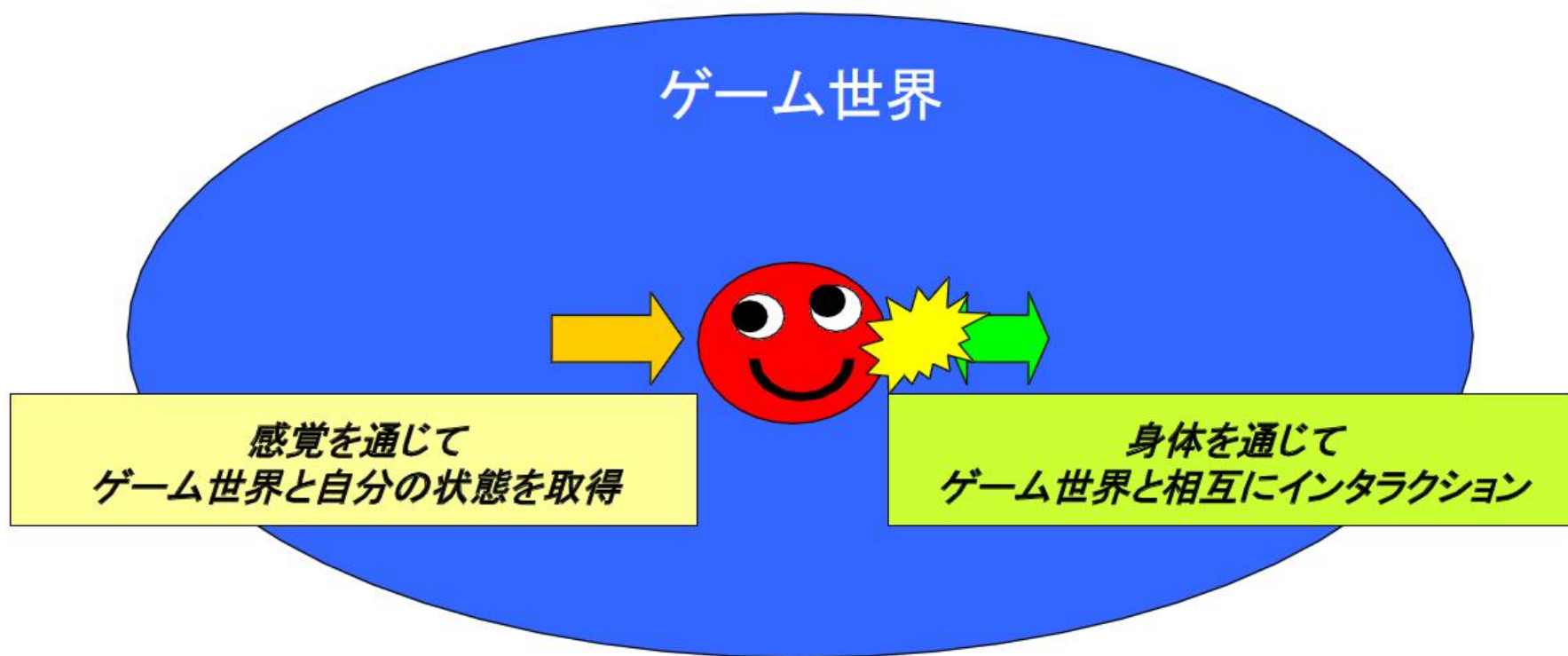
- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims

# エージェント・アーキテクチャ

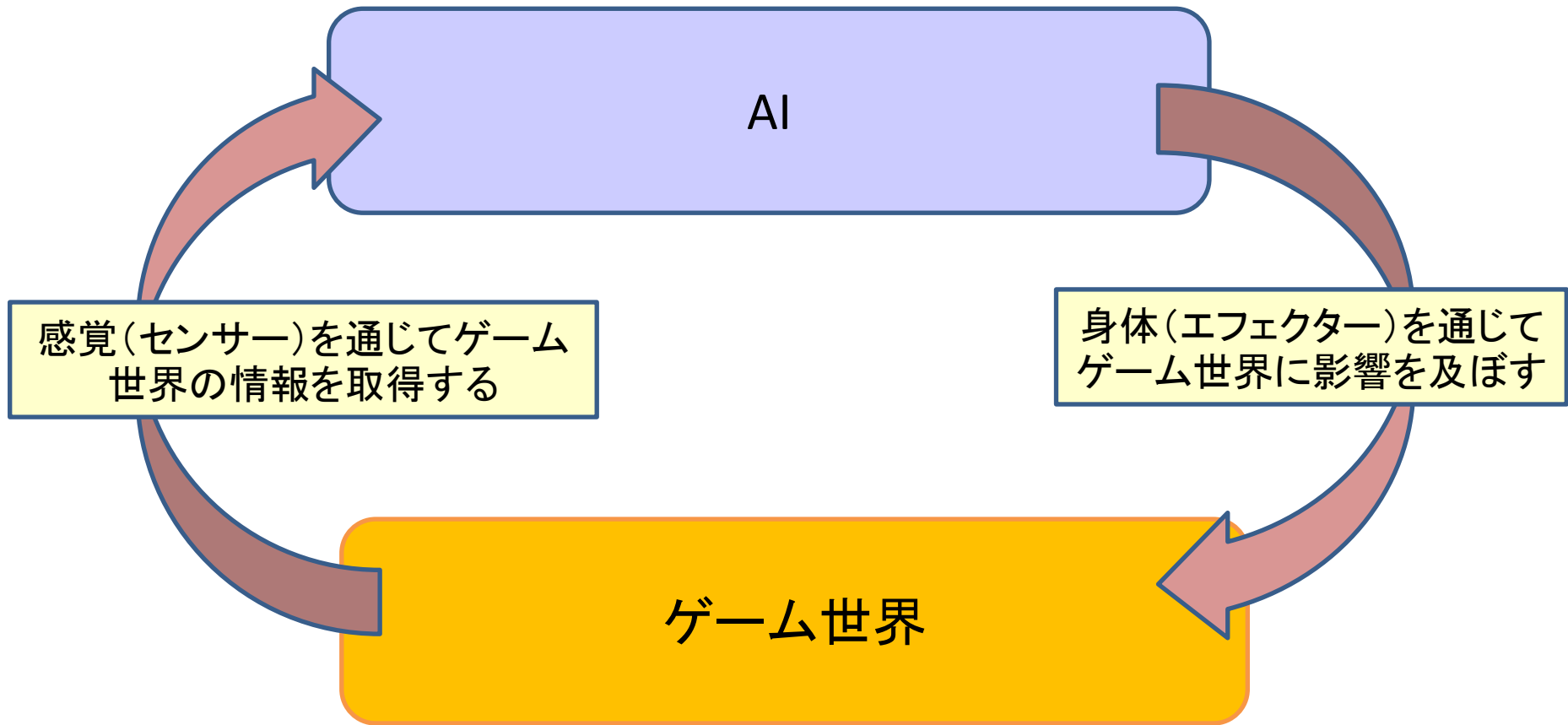
環境内における自律型AIのフレーム

- ① 環境に対して情報を集める感覚(センサー)を持つ
- ② 自ら判断する能力を持つ。
- ③ 環境に対して働きかけることができる能力(エフェクター)を持つ。

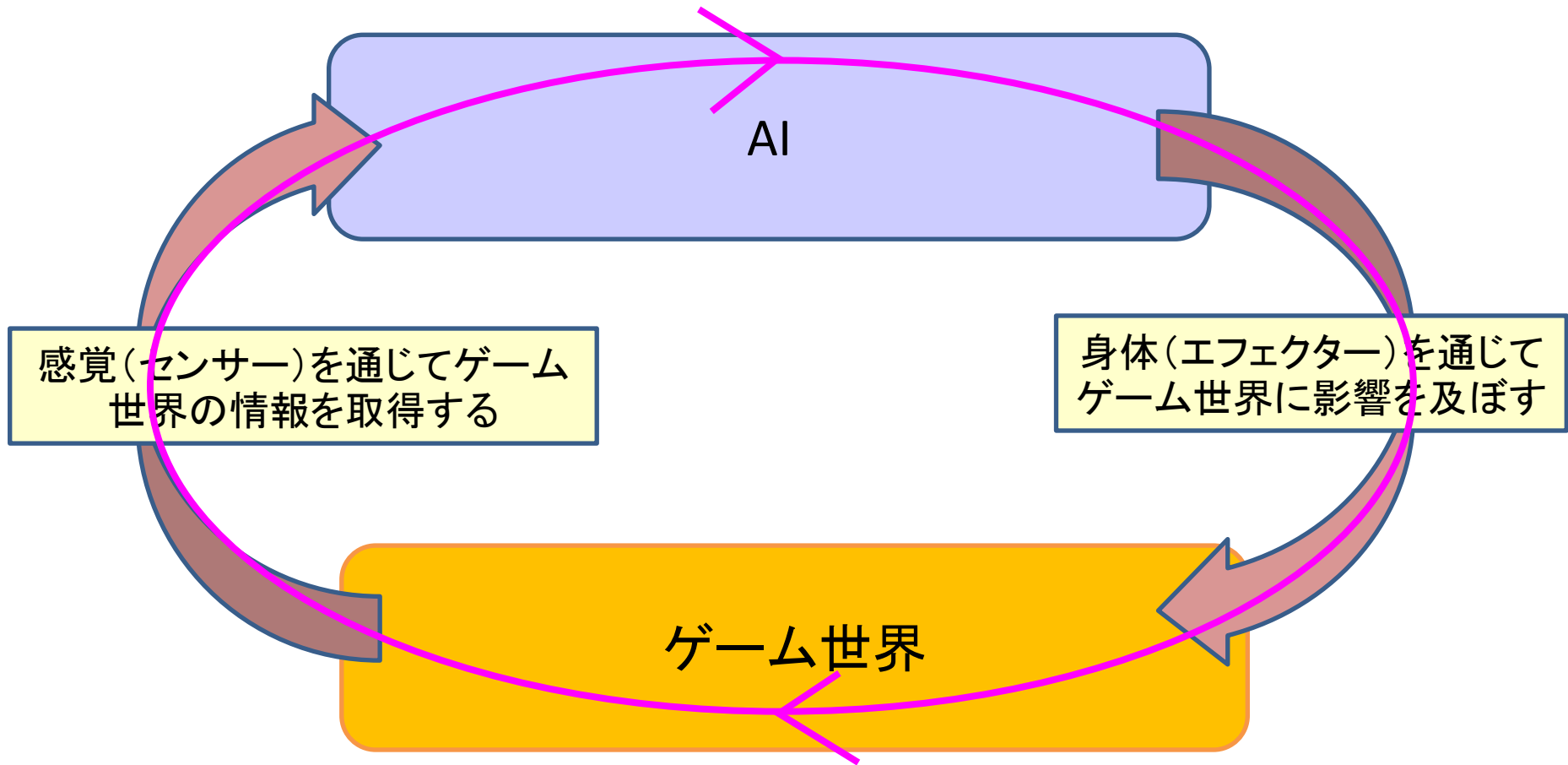
# ゲームにおけるエージェント



# エージェント・アーキテクチャにおける情報の流れ



# エージェント・アーキテクチャにおける情報の流れ



この情報の流れに仕掛けを施す＝知能と世界の相互作用を形成する



# Halo



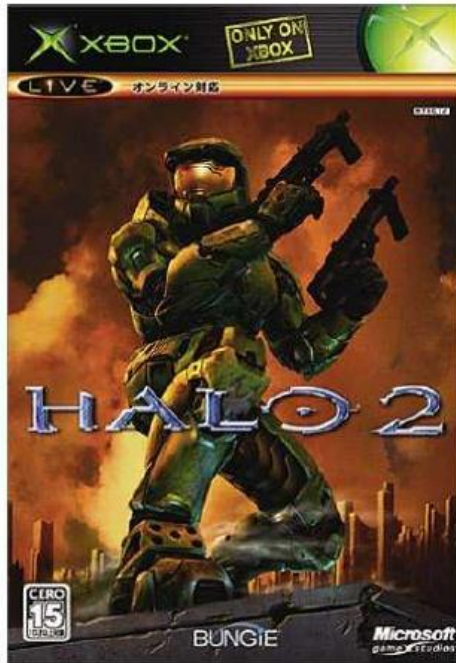
- ◆ 内容: 宇宙船や地表を舞台にしたSFのFPS
- ◆ 開発元: BUNGIE Studio
- ◆ 出版: Microsoft
- ◆ Hardware: Xbox, Windows, Mac
- ◆ 出版年: 2002年

Xbox, 全米、世界を代表するFPSの一つ( Halo 500万本 Halo2 700万 国内10万本)  
「愛嬌のあるNPC」とその演出で、プレイヤーからの定評を得る。

# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine -Frostbite Engine
- (6) 日常系AI
  - The Sims

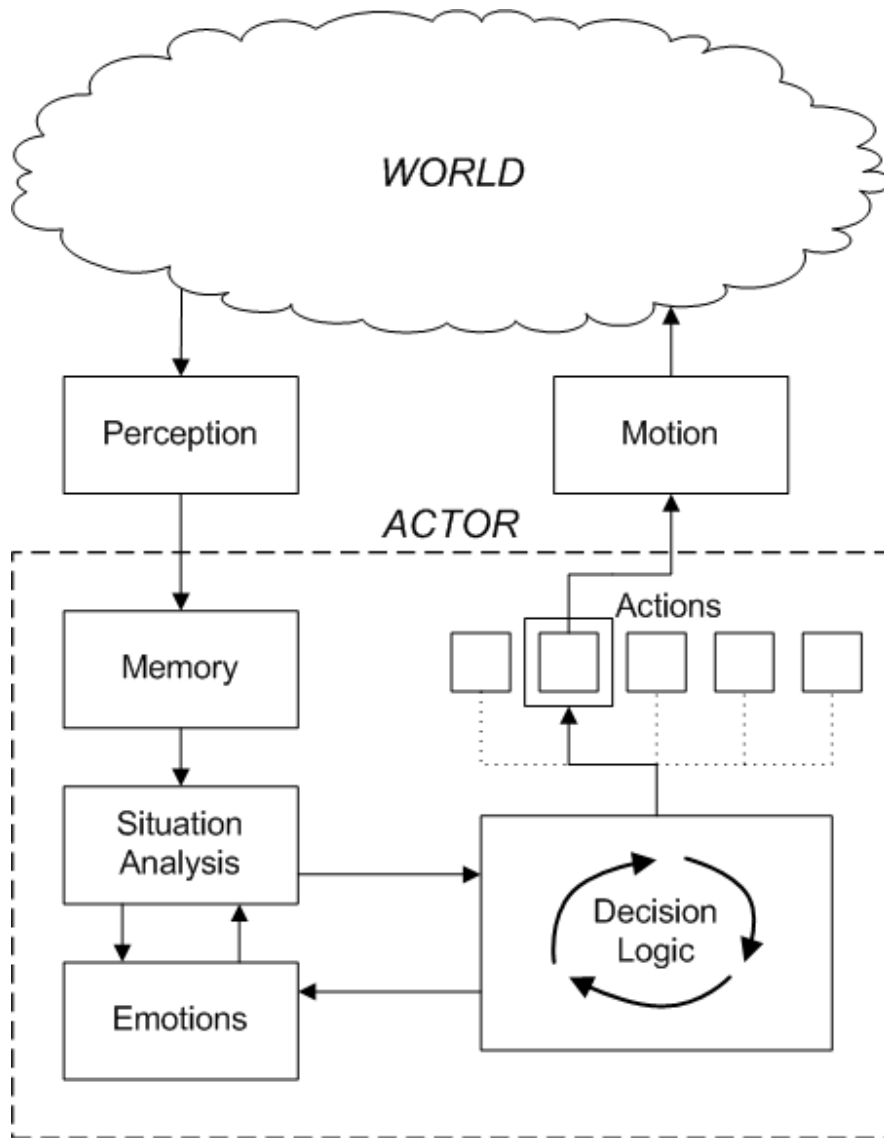
# Halo2



- ◆ 内容: 宇宙船や地表を舞台にしたSFのFPS
- ◆ 開発元: BUNGIE Studio
- ◆ 出版: Microsoft
- ◆ Hardware: Xbox, Windows
- ◆ 出版年: 2004年

Xbox, 全米、世界を代表するFPSの一つ( Halo 500万本 Halo2 700万 国内10万本)  
AIは全てシステムを変更しさらに高度になったが、プレイヤーからはその差がよくわからなかった。

# Halo におけるエージェント・アーキテクチャ



## 特徴:

状況解析(Situation Analysis)によってゲーム内で起こっているイベントが形成（敵が撃って来た、味方がやられた、など）され、それが意志決定機関に渡される。

感情は感情パラメーターなどがあるわけではなく、イベントに応じて決められたセリフを発するようになっている。

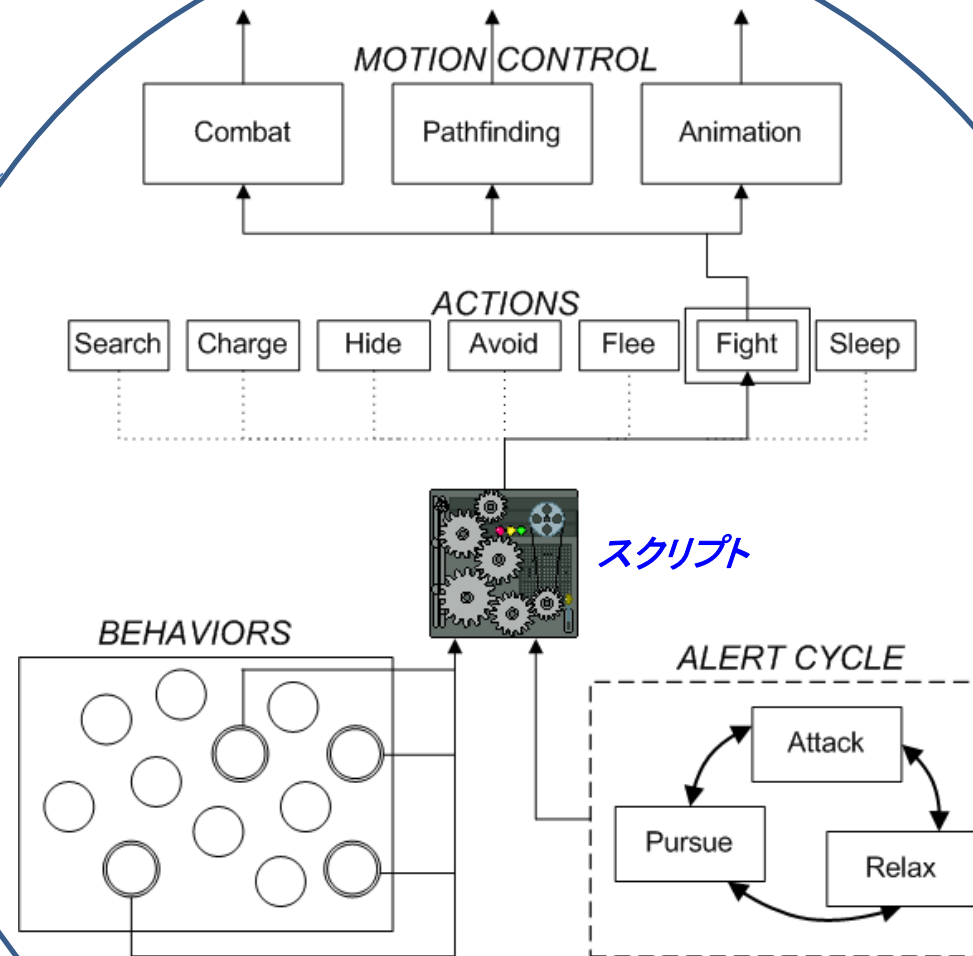
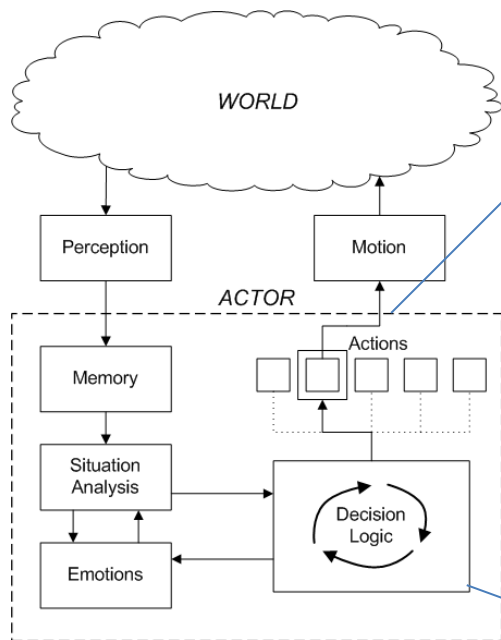
Jaime Griesemer (Design Lead) ,  
Chris Butcher (Engineering Lead)

The Illusion of Intelligence

<http://halo.bungie.net/Inside/publications.aspx>

# Halo におけるエージェント・アーキテクチャ

## FSMとスクリプトによる意思決定



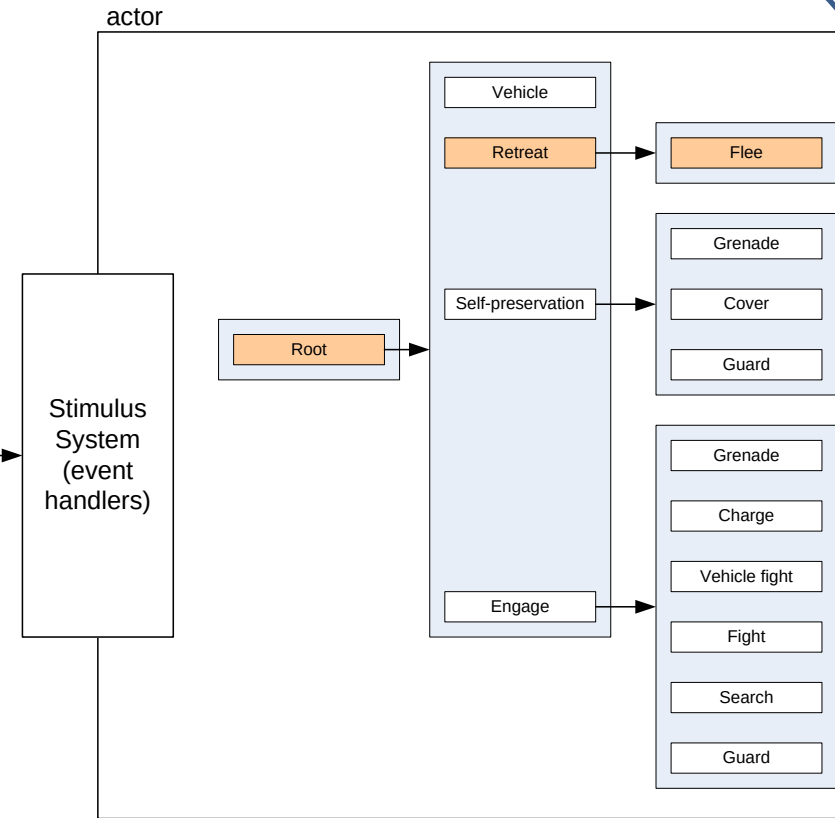
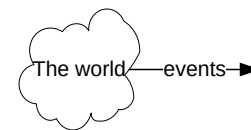
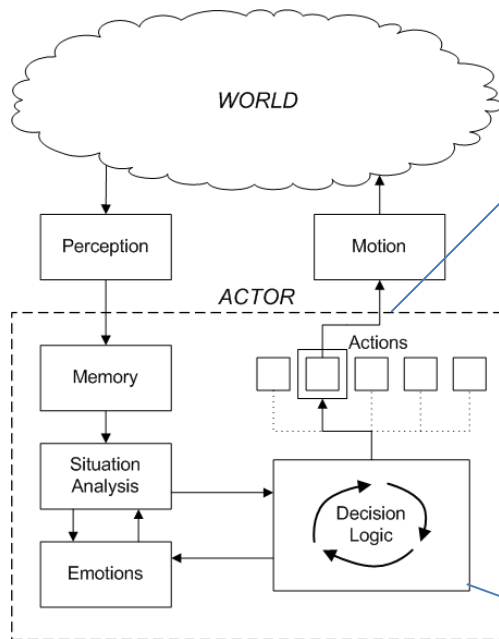
Jaime Griesemer (Design Lead),  
Chris Butcher (Engineering Lead)

The Illusion of Intelligence  
<http://halo.bungie.net/Inside/publications.aspx>



# Halo 2におけるエージェント・アーキテクチャ

## Behavior-treeによる意思決定



Handling Complexity in the Halo 2 AI, D. Isla, GDC 2005  
<http://naimadgames.com/publications.html>

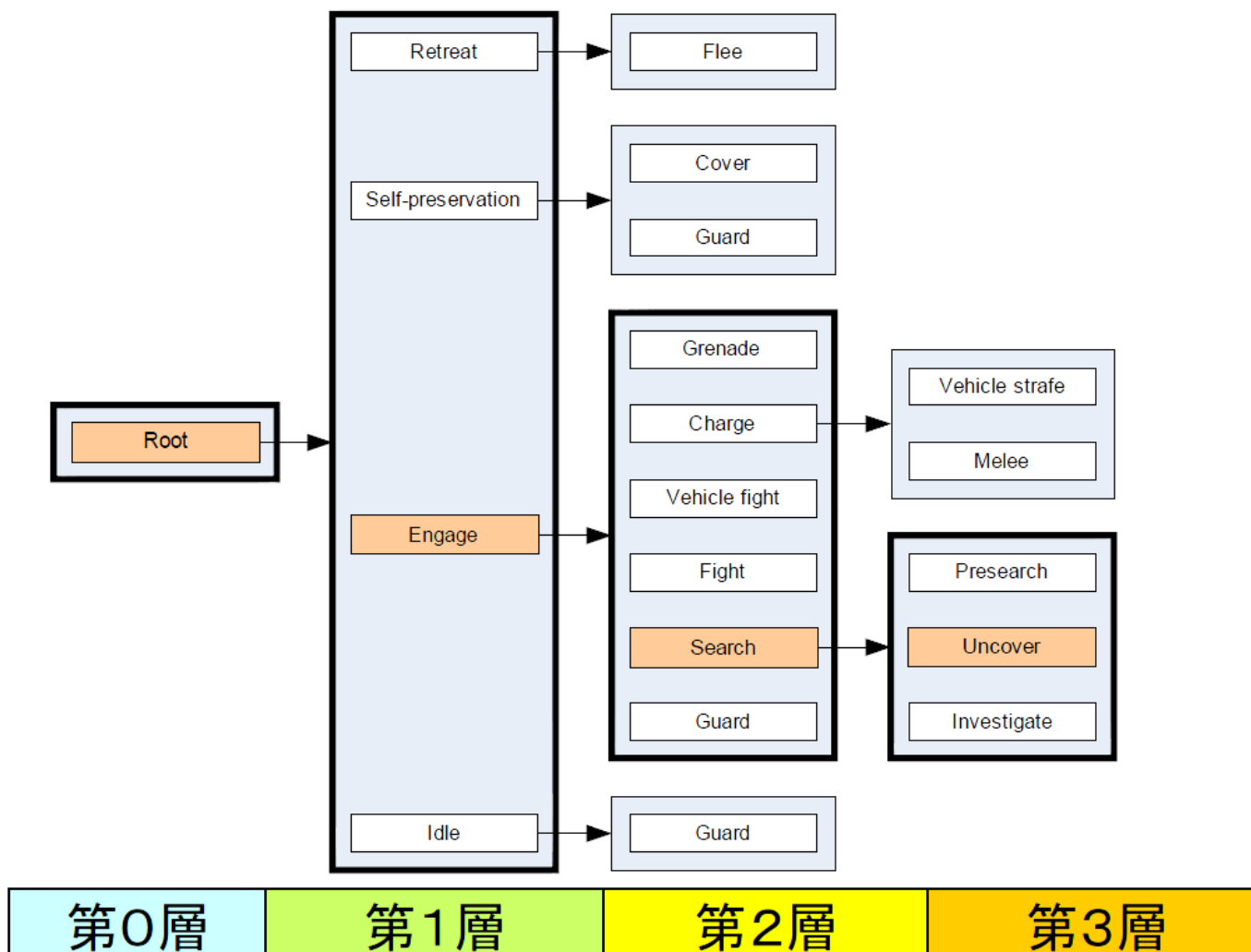
Jaime Griesemer (Design Lead),  
Chris Butcher (Engineering Lead)

The Illusion of Intelligence  
<http://halo.bungie.net/Inside/publications.aspx>

# HFSM(階層型有限状態マシン)による意志決定

DAG (有向非環グラフ) 一方向にノードが伸びて、ノードが環を為すことのないグラフ構造

振る舞いツリー 末端のノードは全てキャラクターの行動を表現する

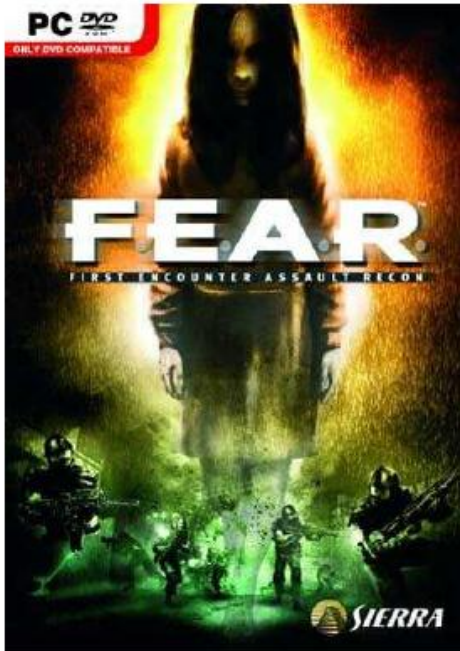


# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine -Frostbite Engine
- (6) 日常系AI
  - The Sims



# F.E.A.R



内容: 閉鎖空間の中のホラーFPS

開発元: Monolith Production

出版: SIERRA

Hardware: Windows, PS3

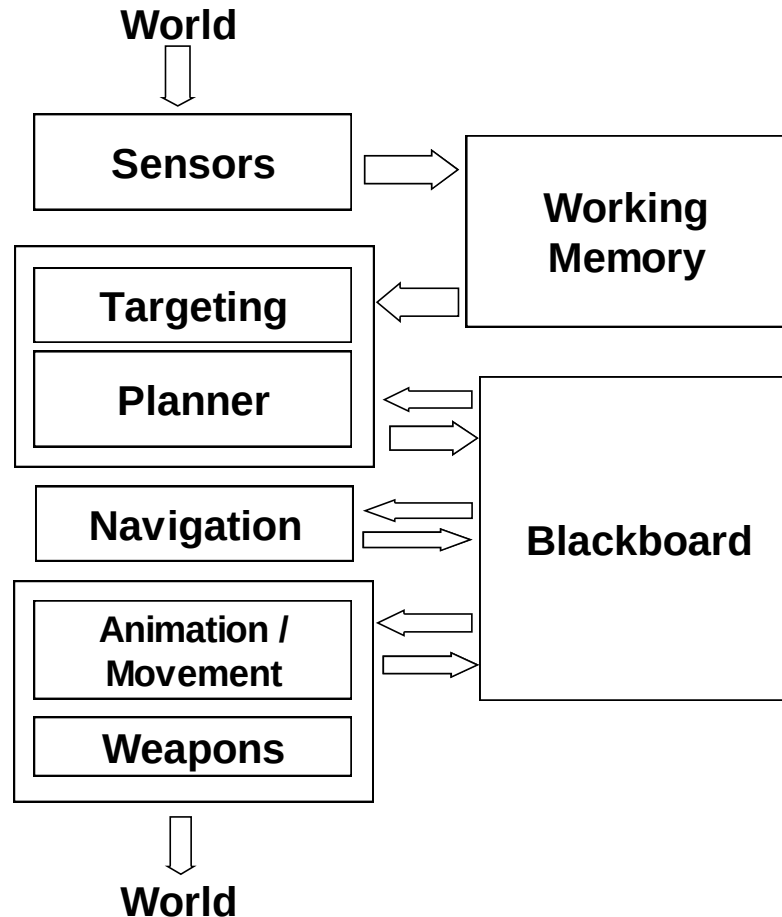
出版年: 2004年



Agent Architecture Considerations for Real-Time Planning in Games (AIIDE 2005)  
[http://web.media.mit.edu/~jorkin/AIIDE05\\_Orkin\\_Planning.ppt](http://web.media.mit.edu/~jorkin/AIIDE05_Orkin_Planning.ppt)

FPSとホラーを、映画的な演出によって結びつけたエポックメイキングな名作。  
長年発展させて来たAI技術の本領が発揮され、開発者、プレイヤーから高い支持を集める。

# F.E.A.R. におけるエージェント・アーキテクチャ



# ①統一事実記述形式

キャラクター

オブジェクト

任務

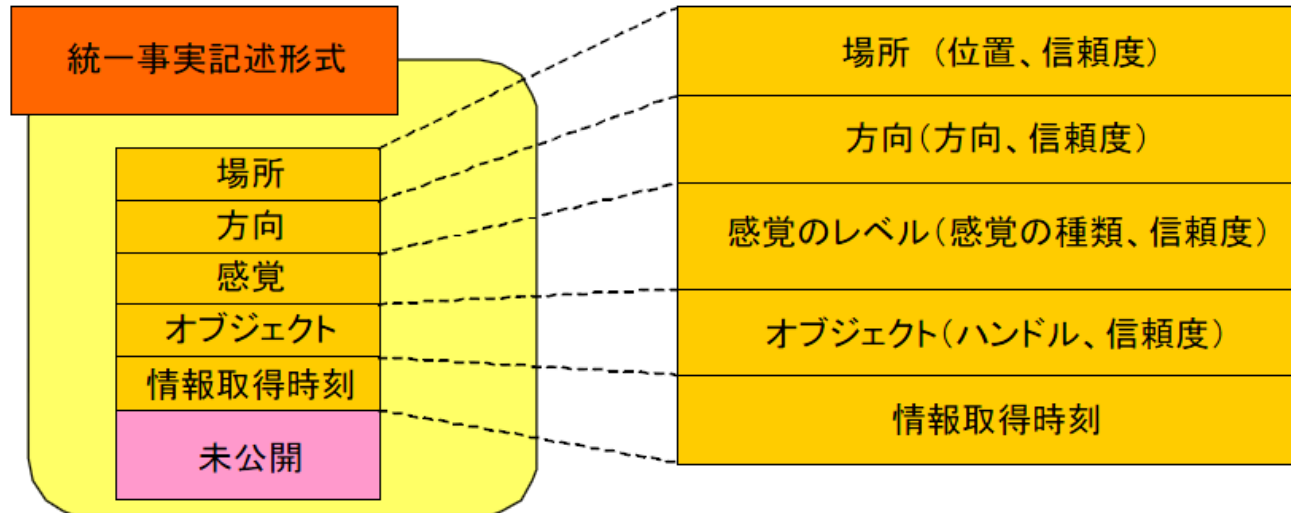
事件

パス

欲求

ノード

全て以下の形式(フォーマット)で記述する。



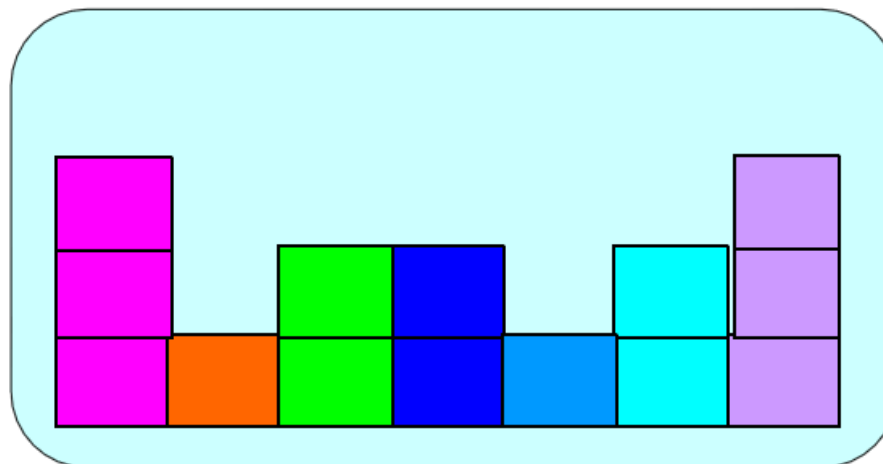
全部で本当は16個の属性がある

```
WorkingMemoryFact
{
    Attribute<Vector3D>    Position
    Attribute<Vector3D>    Direction
    Attribute<StimulusType> Stimulus
    Attribute<Handle>      Object
    Attribute<float>       Desire
    ...
    float                  fUpdateTime
}
```

```
Attribute<Type>
{
    Type Value
    float fConfidence
}
```

# ①統一事実記述形式

記憶領域に認識した事実をWorking Memoryへ蓄積する



Working Memory

敵Cが時刻Mに位置Zにいた  
信頼度 = 0.6

敵Aが時刻Lに位置Xにいた  
信頼度 = 1.0

敵Bが時刻Mに位置Yにいた  
信頼度 = 0.0



ターゲットを決める

Blackboard

ターゲットはAにする

# F.E.A.Rのプランニング① シンボル

**kTargetIsDead = ture**  
この兵士Aは死んだ

**kTargetAtme = ture**  
この兵士Bは自分を狙っている



**kWeaponIsLoaded = false**  
私Cの武器は装填済みでない



# F.E.A.Rのプランニング① シンボル

エージェントの認識する世界をもっとシンプルに表現したい

→ 20個のシンボルで世界を集約して表現する

## シンボル

kSymbol\_AtNode  
どのノードにいるか

kSymbol\_WeaponArmed  
武装しているか

kSymbol\_Target  
IsSuppressed  
威嚇されているか

kSymbol\_TargetIs  
AimingAtMe  
どのノードにいるか

kSymbol\_AtNodeType  
どんなタイプのノードにいるか

kSymbol\_UsingObject  
オブジェクトを使っているか?

kSymbol\_WeaponLoaded  
装填されているか

kSymbol\_RidingVehicle  
乗り物に乗っているか

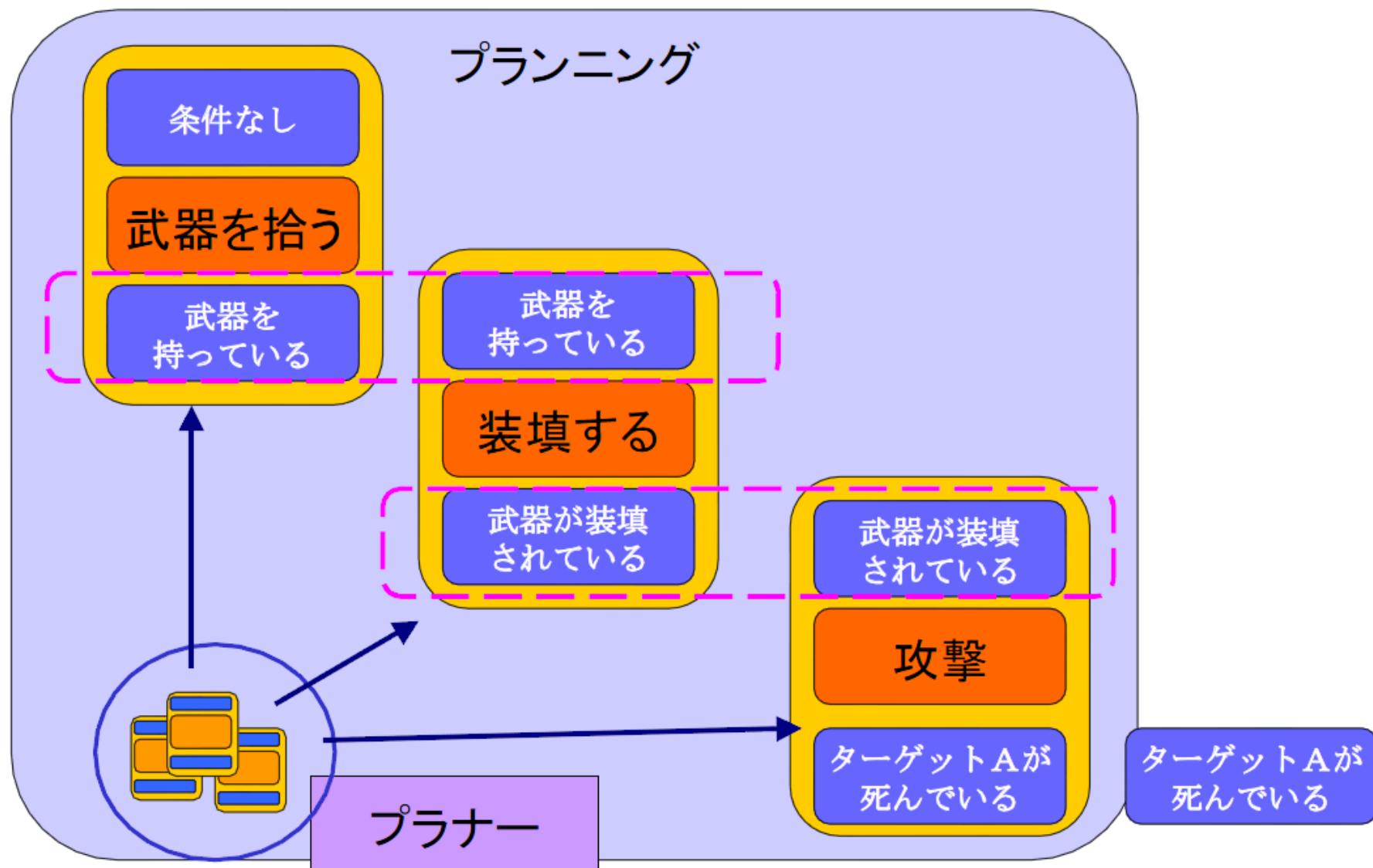
kSymbol\_TargetIsDead  
死んでいるか

各エージェントについて (Agent-centric)

このシンボルをプランニングへ

# F.E.A.R.のプランニング②

## シンボルによる連鎖プランニング



# F.E.A.R. デモ





# F.E.A.R. デモ



Agent Architecture Considerations for Real-Time Planning in Games (AIIDE 2005)  
[http://web.media.mit.edu/~jorkin/AIIDE05\\_Orkin\\_Planning.ppt](http://web.media.mit.edu/~jorkin/AIIDE05_Orkin_Planning.ppt)

# F.E.A.R. デモ

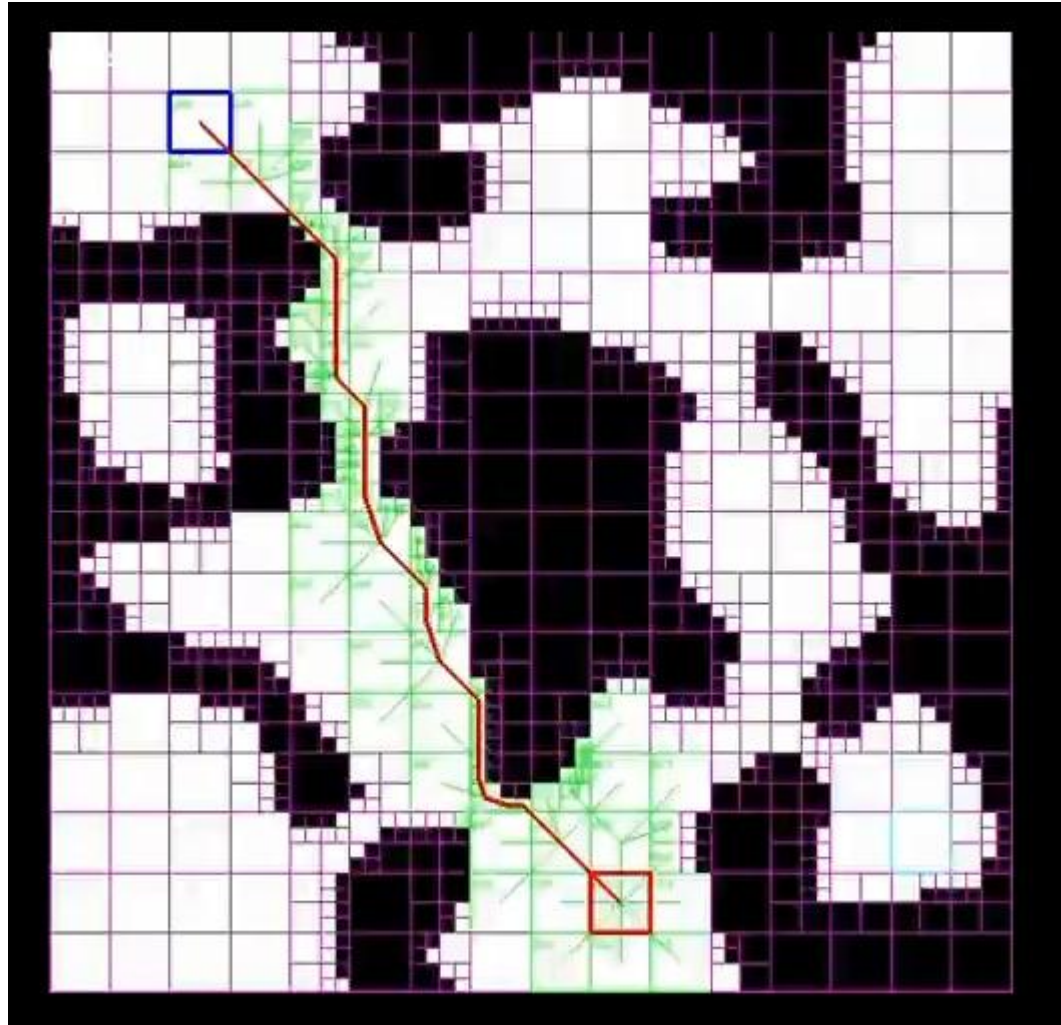


Agent Architecture Considerations for Real-Time Planning in Games (AIIDE 2005)  
[http://web.media.mit.edu/~jorkin/AIIDE05\\_Orkin\\_Planning.ppt](http://web.media.mit.edu/~jorkin/AIIDE05_Orkin_Planning.ppt)

# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims

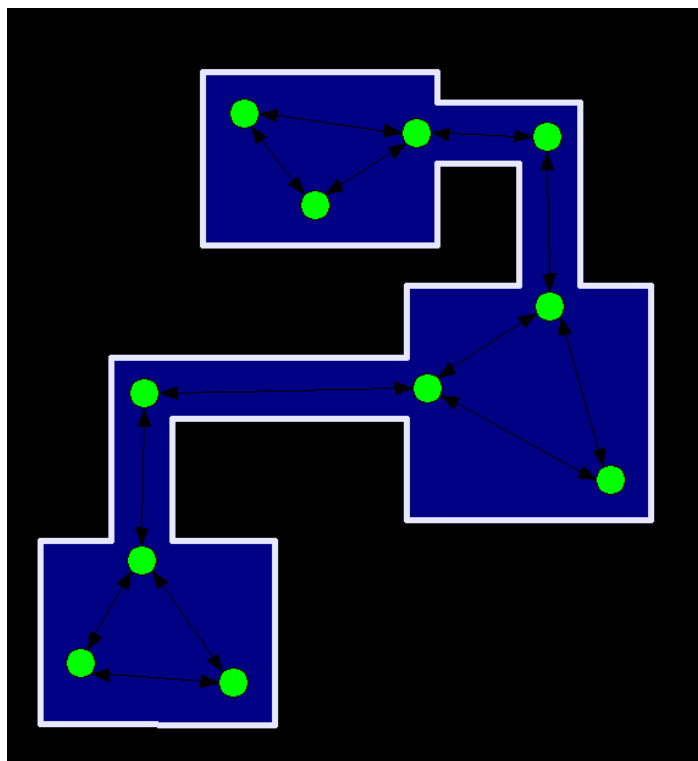
# ナビゲーションメッシュとA\*デモ



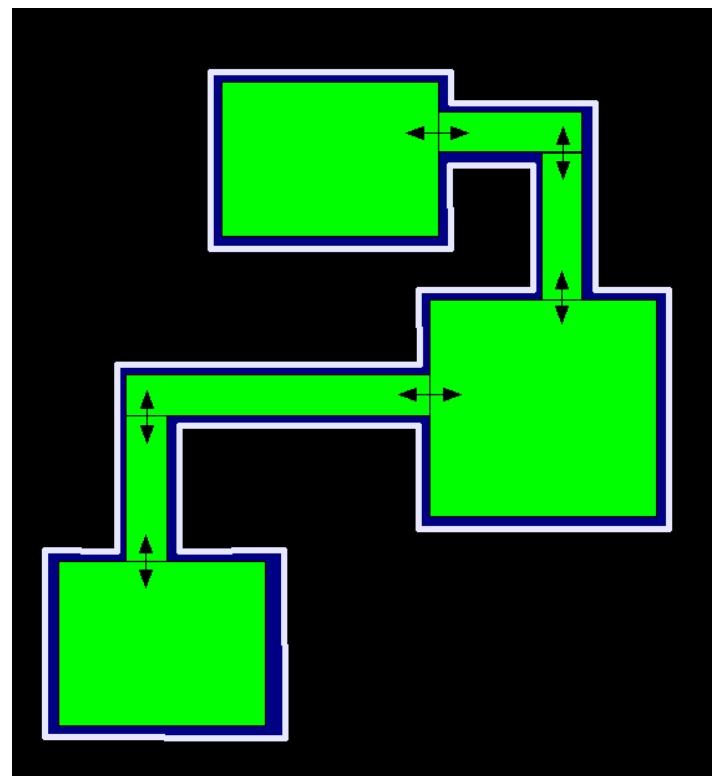
<https://www.youtube.com/watch?v=95aHGzzNCY8>

# CounterStrike におけるパス検索

- Navigation Mesh, ウェイポイントは単なるパスシステムではなくて3D環境をAIが認識するための仕組み(=世界表現、World Representation(WR))

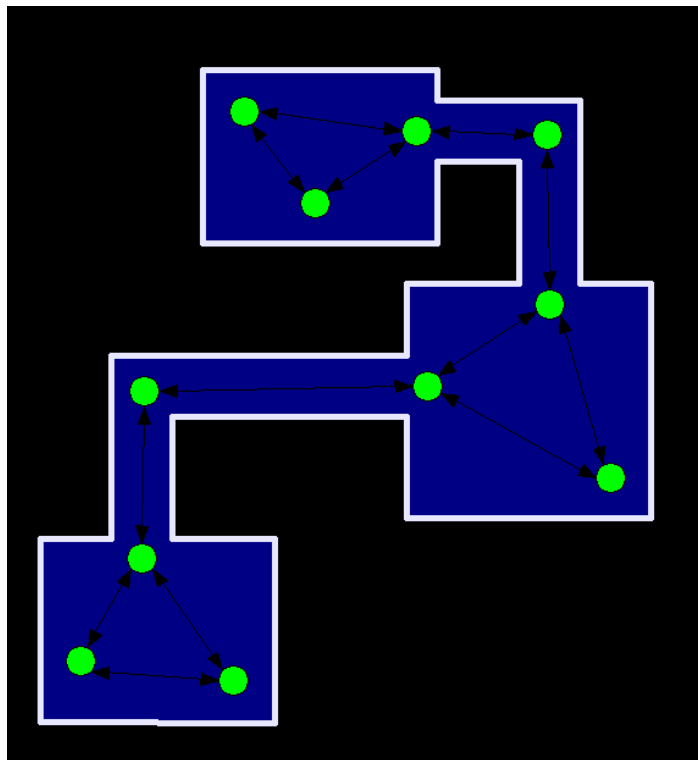


ウェイポイント



ナビゲーション・メッシュ

# CounterStrike におけるパス検索

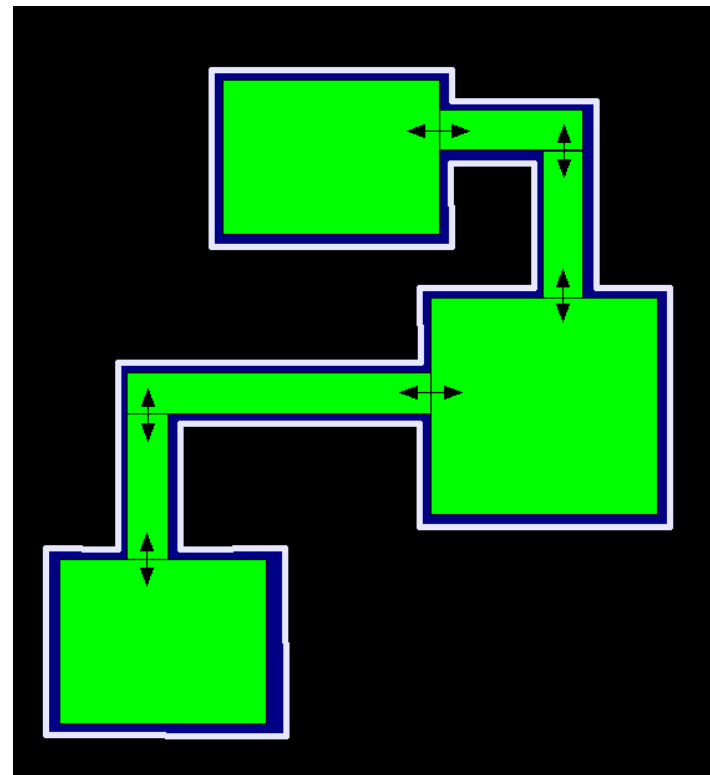


## ウェイポイント

利点: 精密なパス

決定: 地形情報を含まない。

階層化できない。



## ナビゲーション・メッシュ

利点: 地形情報を含んだパス検索  
階層化できる。



採用

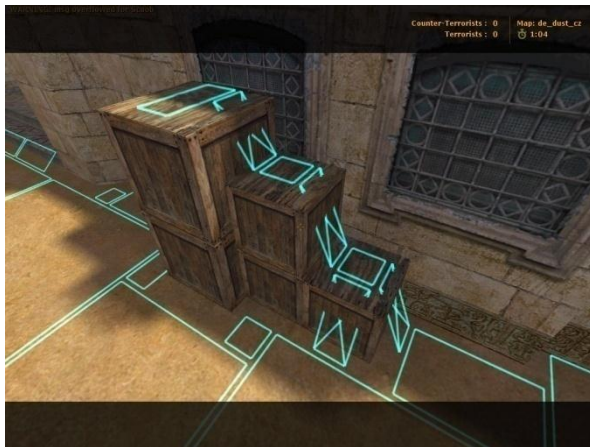
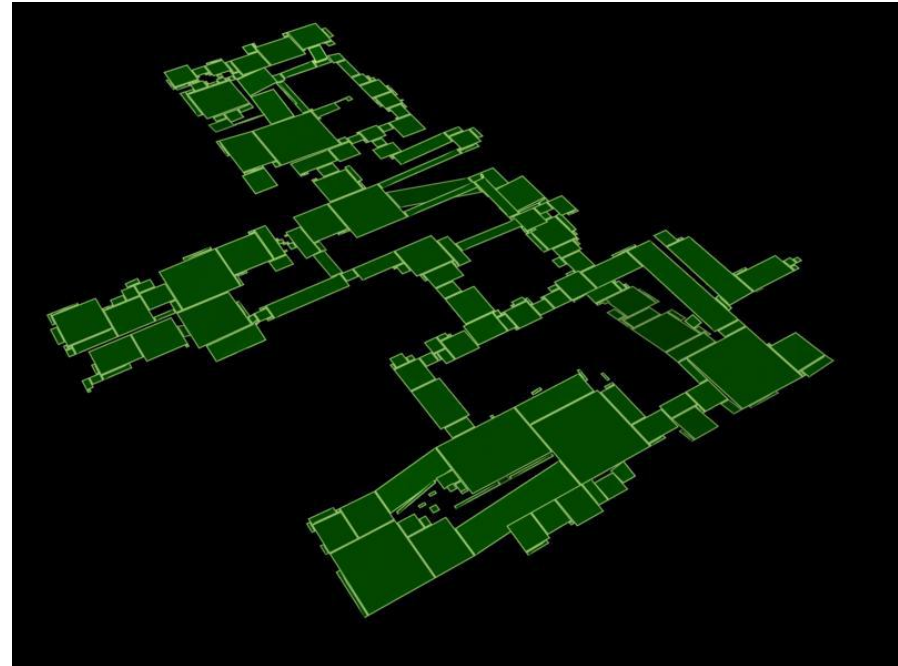


# Counter-Strike におけるパス検索



The Official Counter-Strike Bot

<http://aigamedev.com/insider/article/official-counter-strike-bot/>



- メッシュ = 長方形(その中で障害物に当たらずに歩くことが出来るエリア)
- メッシュの接続の属性(一方向、双方向、はしご、など)
- 各メッシュは属性を持つ。
  - 属性1: キャラクターの動作(ジャンプ、かがむ、など)
  - 属性2: 危険度パラメーター(味方がそこでやられたら危険度アップ)
  - 属性3: イベント用の特別な場所などとしての設定



# CounterStrike におけるパス検索

ボットが扱うべき環境

- コーナー/エッジ/小さなオブジェクト
- 窓際の出っ張りをまたいで行く
- ギャップをジャンプする
- 一方通行の落下
- ドア
- 梯子
- かがむ場所
- 破壊して進む (窓とか)

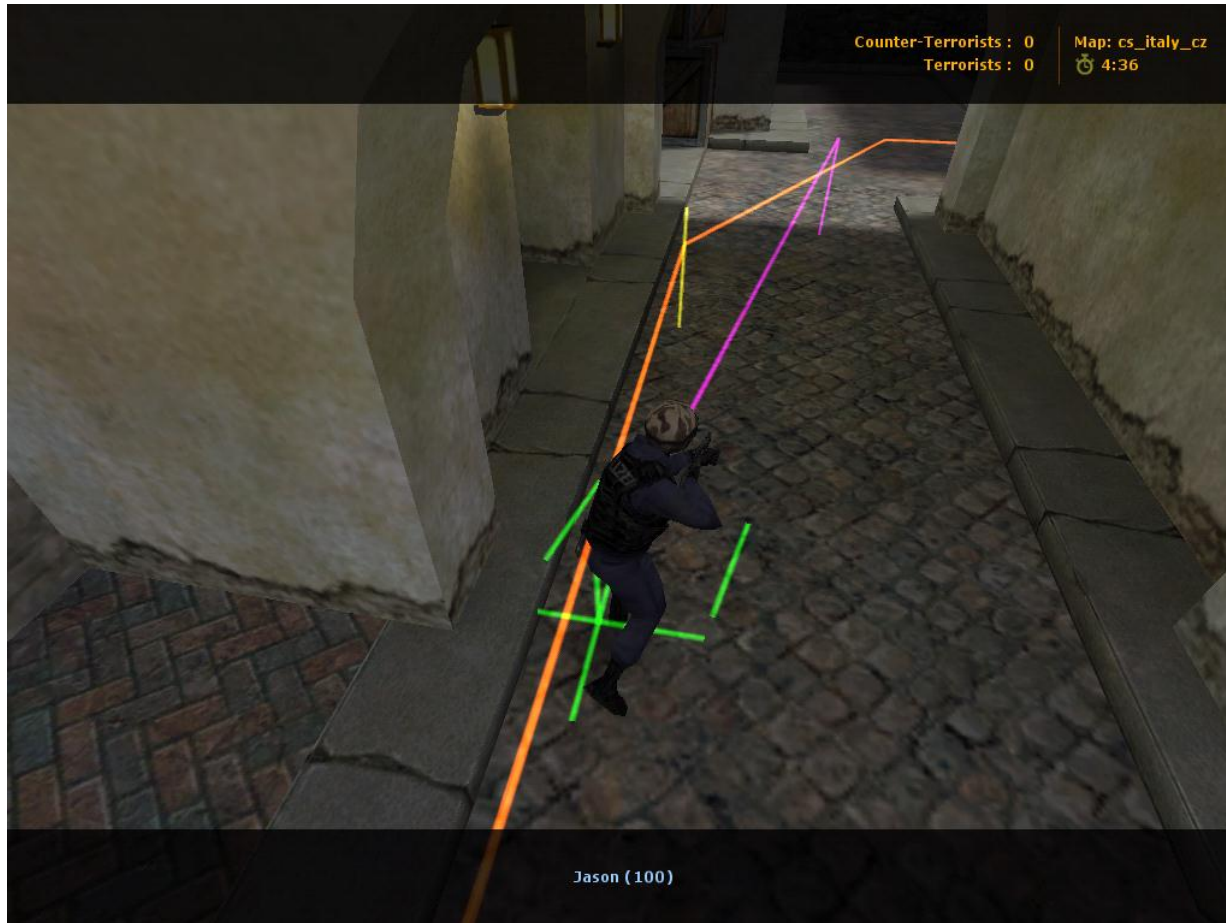
これらをパスデータに埋め込んで動作させる。

- 自動的に隠れることができるポイントを学習
- チーム間でコミュニケーション

## Movement Hierarchy



# Path Following (デモ)



The Official Counter-Strike Bot by Michael Booth, Michael Booth on October 19, 2008  
<http://aigamedev.com/insider/presentation/official-counter-strike-bot/>

# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims

# StarCraft 2 のパス検索

Evolution of RTS AI

Bob Fitch, Blizzard

<http://movingai.com/aiide11/speakers.html>

<http://movingai.com/aiide11/BlizzardAI.zip>

特徴:

- 要素: ナビゲーションメッシュ
- 生成: ドロネー三角形分割(Delaunay Triangulation)
- スペック: 64000メッシュ 2000体が使用
- 動的に変化しない
- 図形タイプ(メッシュに付けるの種類): 大地、崖、ビル、何ものなし

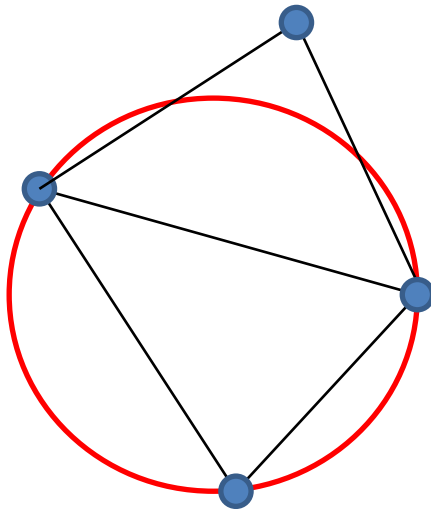
AI Navigation: It's Not a Solved Problem - Yet

James Anhalt, Alexander Krings, Nathan Sturtevant

Blizzard Entertainment, Nihilistic, University of Denver

Track / Format AI Summit

<http://www.gdcvault.com/play/1014514/AI-Navigation-It-s-Not>



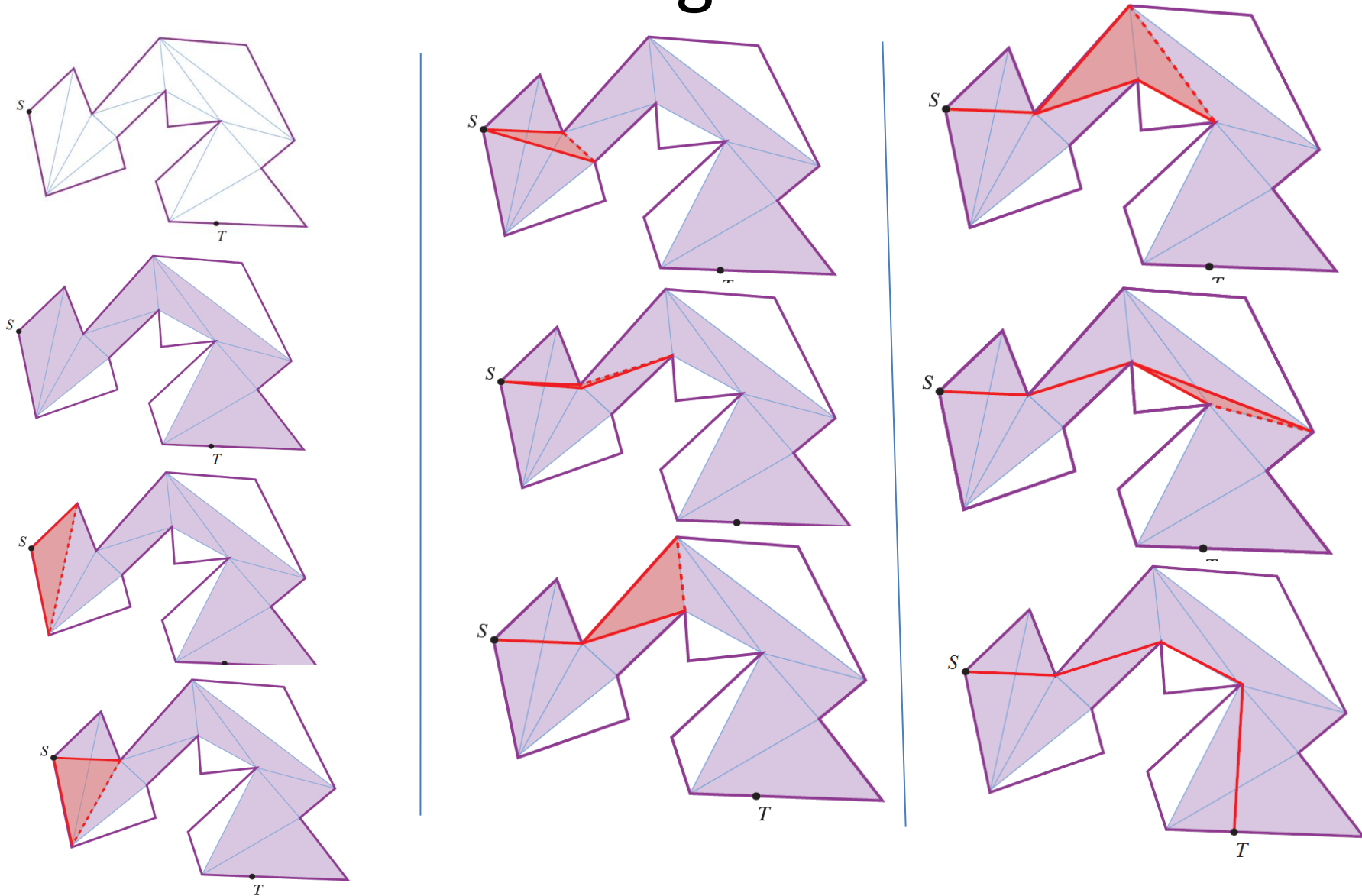
ドロネー三角形分割

三角形の外接円に他の三角形の頂点が含まれない分割の仕方



※パス検索の分野では計算幾何学や数学のトポロジーといった専門分野がこれからますます必要となる

# Funnel Algorithm



<http://www.cs.uwaterloo.ca/~alubiw/shortest-path-lecture.pdf>

[http://skatgame.net/mburo/ps/thesis\\_demyen\\_2006.pdf](http://skatgame.net/mburo/ps/thesis_demyen_2006.pdf)

# 目次

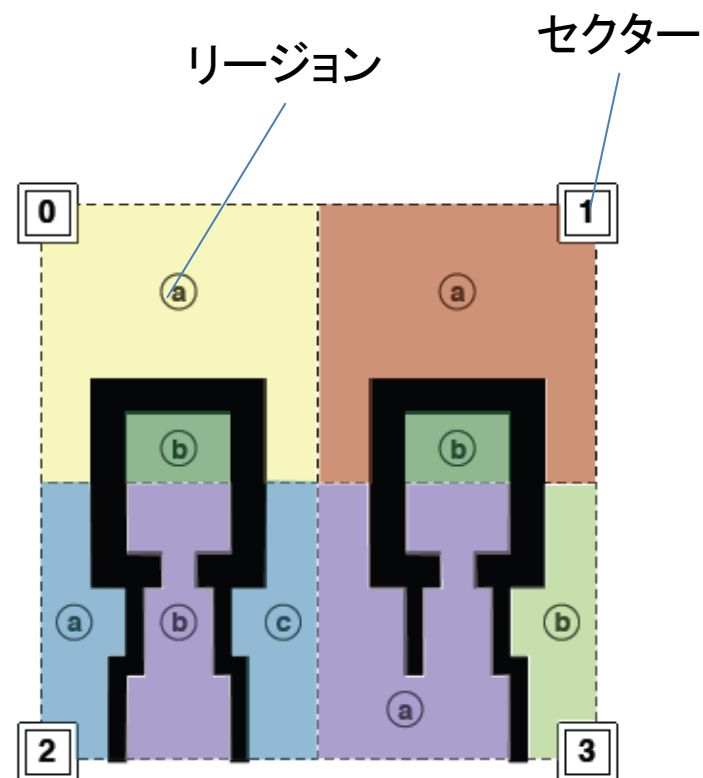
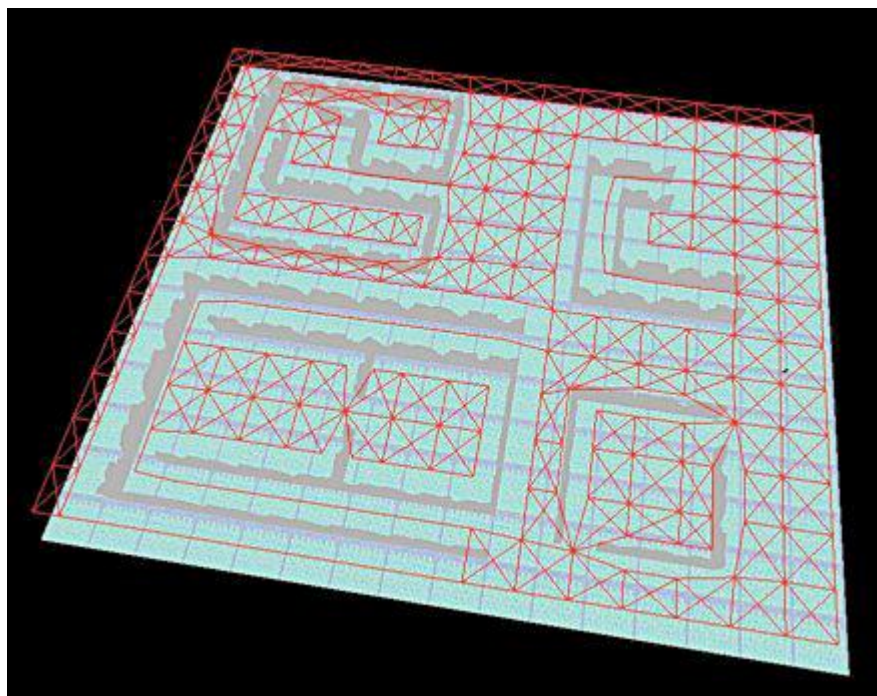
- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine -Frostbite Engine
- (6) 日常系AI
  - The Sims



# Dragon Age のパス検索

特徴:

- 要素: グリッド + アブストラクト・グラフ (2階層)
- 生成: グリッド
- 省メモリ、高速
- グリッド < リージョン < セクター  
(階層型パス検索システム)



「Dragon Age」(Bioware)におけるパス検索システム (E3) (y\_miyake のゲームAI千夜一夜)

<http://blogai.igda.jp/article/29628420.html>



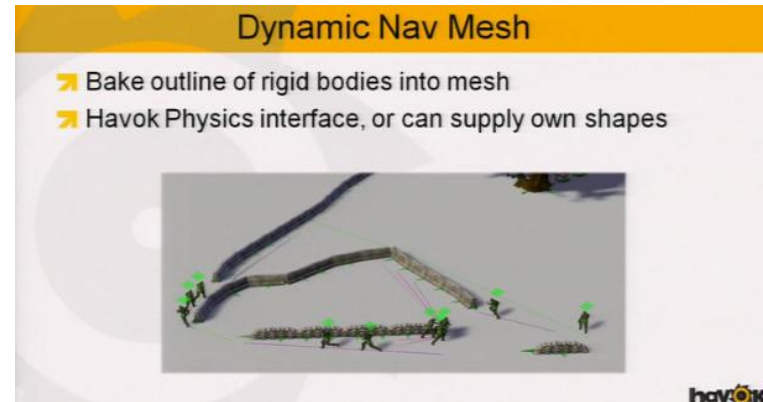
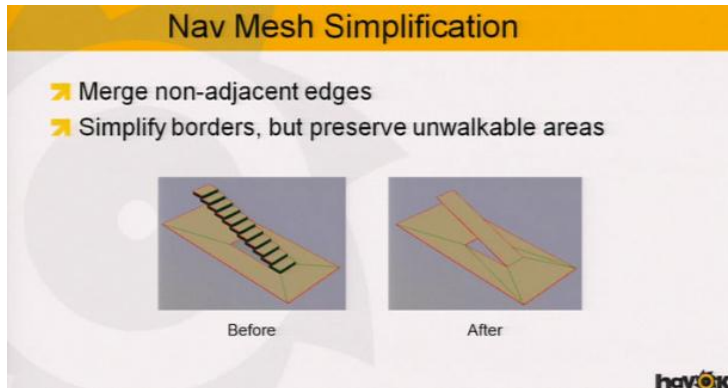
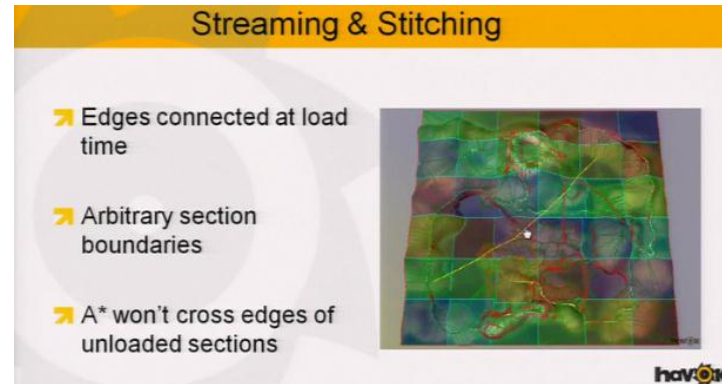
# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine -Frostbite Engine
- (6) 日常系AI
  - The Sims

# Havok AI

## Havok AI

- Navmesh 自動生成
- 自動シンプリケーション
- 自動動頂点連結
- 領域分割(階層化)
- 障害物の焼き込
- 物理との連携
- A\*



Finding Your Way with Havok AI (Presented by Havok),  
Chris Elion, Brett Vickers (Havok, ArenaNet)

<http://www.gdcvault.com/play/1014700/Finding-Your-Way-with-Havok>

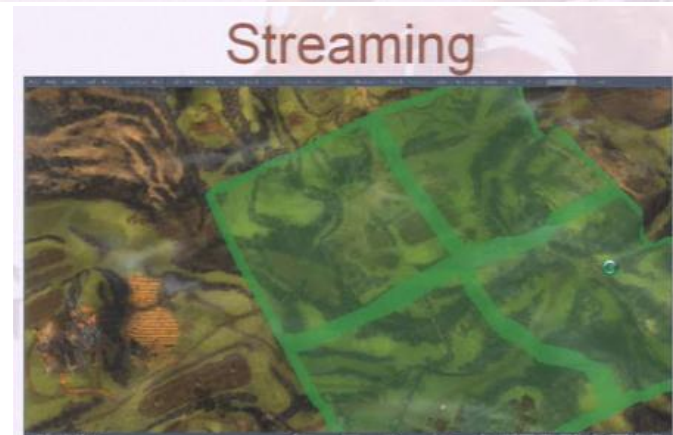
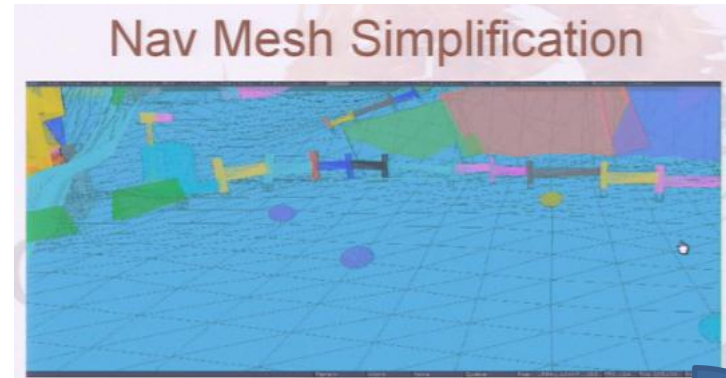
# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - **Guild War 2(Havok AI)**
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims

# Guild War 2(Havok AI)

## Guild War 2

- シードを置いてNavmesh 自動生成
- 自動シンプリケーション
- 領域分割(ストリーミング)
- 動的メッシュ生成
- 動的コスト変更 ( $A^*$ )
- Raycast に使用



# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims

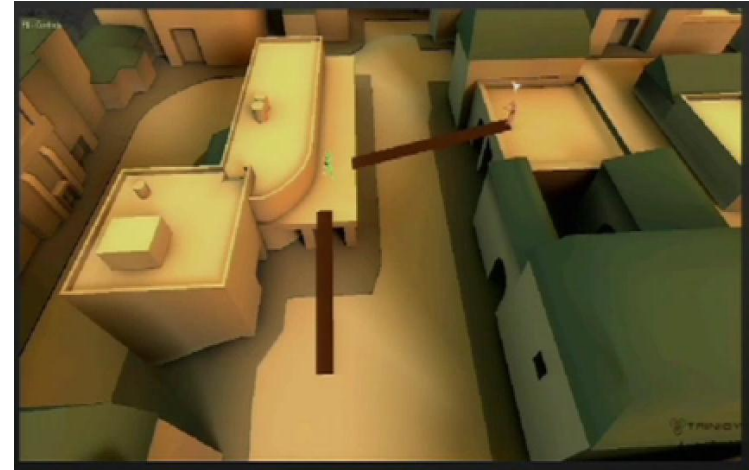
# Autodesk® Kynapse®

## Kynapse 特徴

- キャラクター制御全般
- - dynamic avoidance,  
dynamic reaction など
- データ並列 (SPUなどに対応)
- NavMesh がポイントグラフで覆っ  
ている
- レイキャストに使用

(Future)

データ分割。並列データ単位 CELL



## Pathdata generation and pathdata basics Graph Density

Triangulated polygon  
with holes  
no altitude approx

Delaunay  
triangulation

Greedy point  
insertion.  
NavMesh  
edge length  
heuristic.

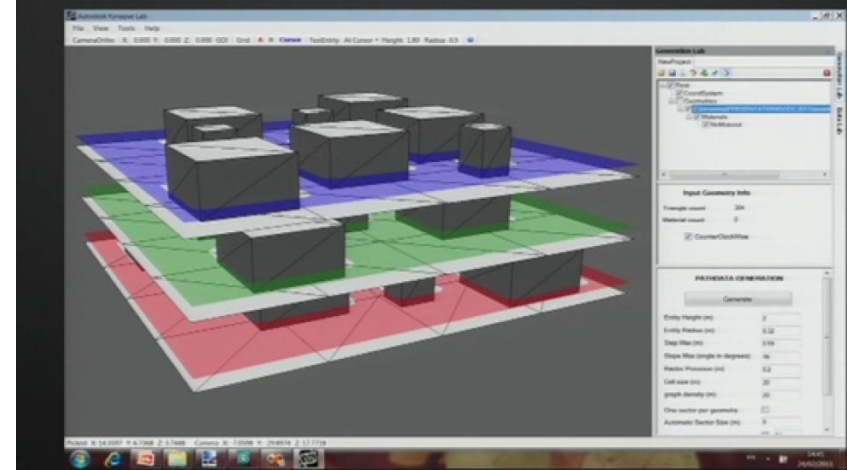
Dense  
delaunay  
triangulation

PointGraph  
(in floor)

Greedy edge  
collapse and  
much more.

Dense  
PointGraph

## Data Parallelism The Cell partition - Floors

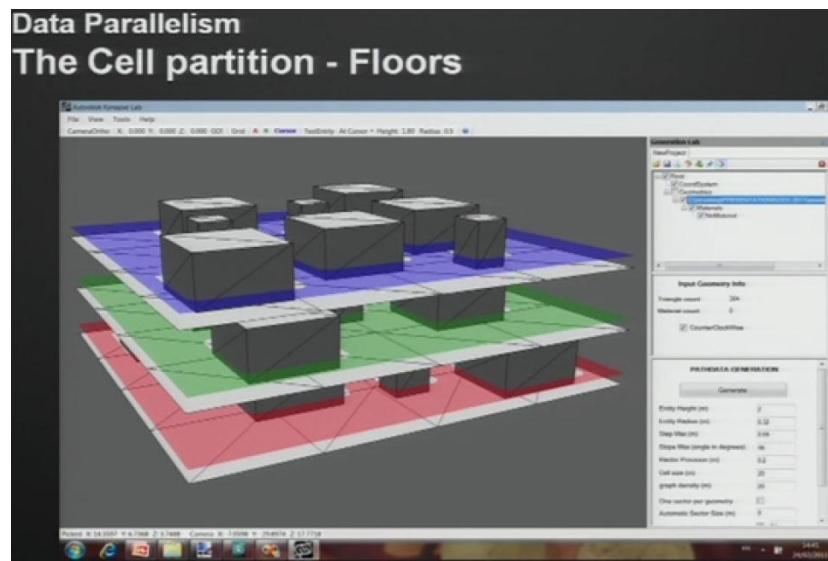




# Autodesk® Kynapse®

## Kynapse 特徴

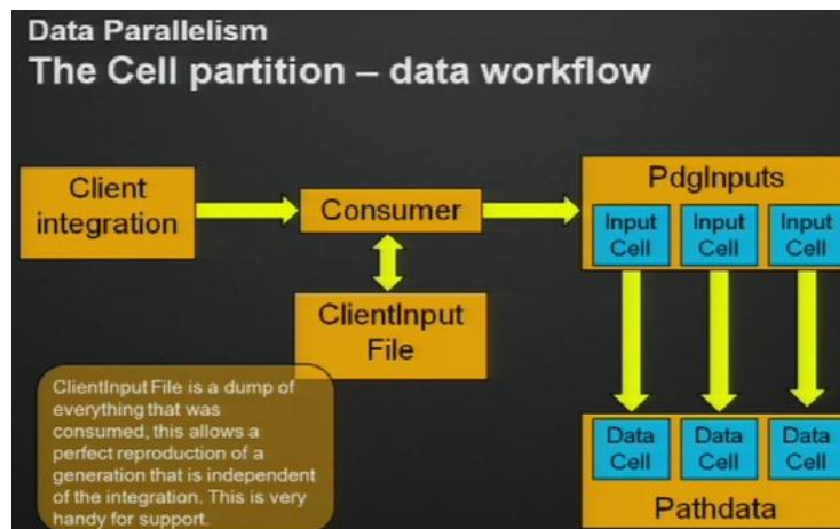
- キャラクター制御全般
- - dynamic avoidance, dynamic reaction など
- データ並列 (SPUなどに対応)
- NavMesh がポイントグラフで覆っている
- レイキャストに使用 (Future)
- データ分割。並列データ単位 CELL



Autodesk® Kynapse®

Kynapse Middleware: Pathfinding Data Structures, from Multi-Core Generation to SPU Usage

<http://area.autodesk.com/gdc2011>





# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims

# DOOM , Quake シリーズ (BSP, Binary Space Partitoning)

BSP (Binary Space Partioning)

- キャラクター移動
- キャラクターの位置把握  
(Area Awareness System)

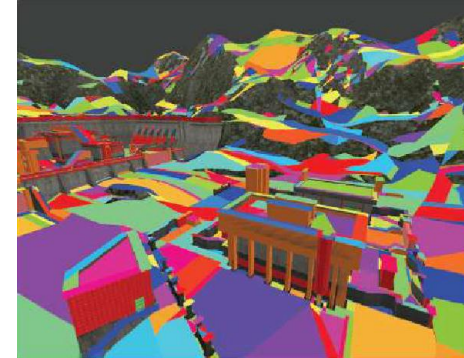
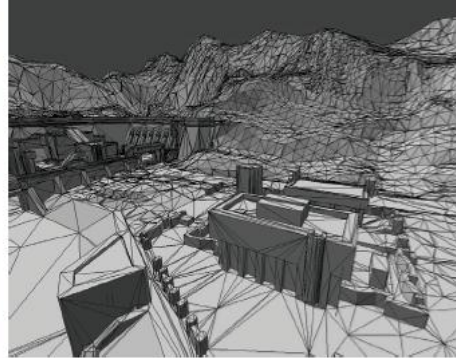
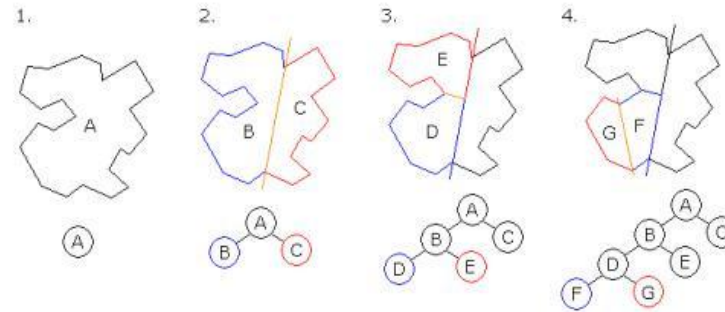


Figure 2. Boundary representation of C-Space for an outdoor environment with buildings

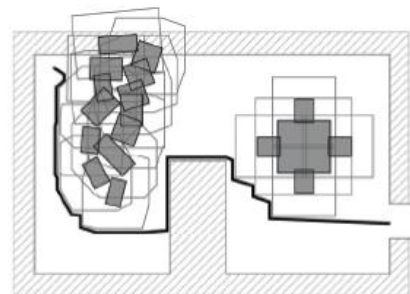


Figure 7. Top down view of one of the paths around the obstacles

Motion Planning

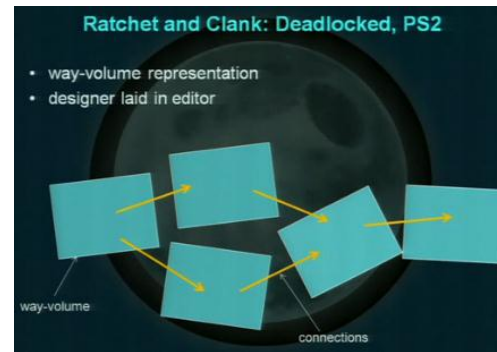
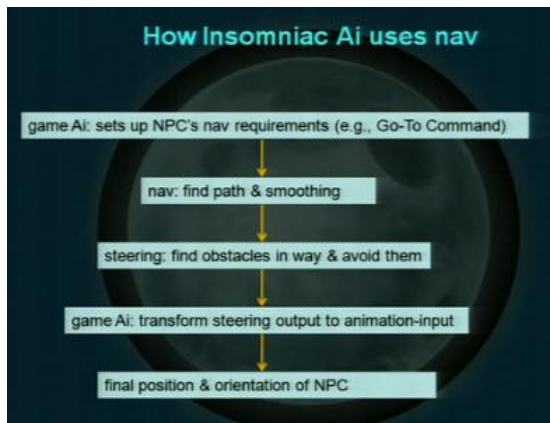
<http://intechweb.org/book.php?id=74&content=title&sid=1>

22 Automated Static and Dynamic Obstacle Avoidance in Arbitrary 3D Polygonal Worlds J.M.P. van Waveren and L.J.M. Rothkrantz

# Insomniac Navigation Engine

## Insomniac PathEngine 特徴

- キャラクター制御まで含めたパスシステム
- - Steering & Behavior レベル
- ボリューム表現(Ratchet and Clank:Deadlocked,PS2)
- PS3 から平面ナビメッシュ
- - Resistance: Fall of man Maya で  
デザイナー作成 × A\*アルゴリズム
- 階層型検索は破棄(パスキャッシュを嫌)



Navigation in Insomniac Engine

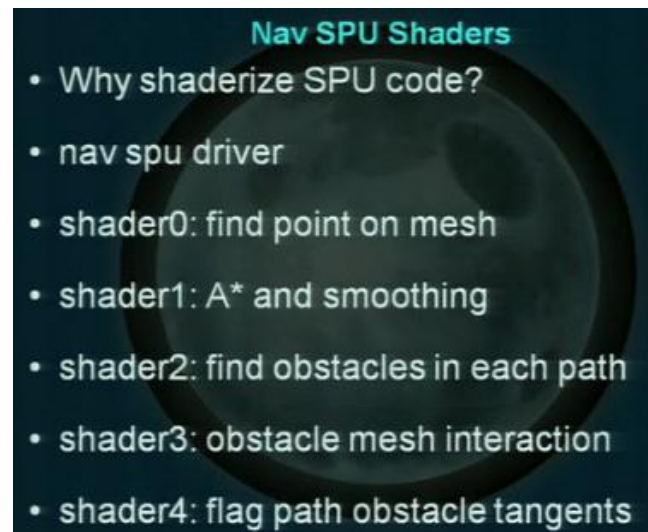
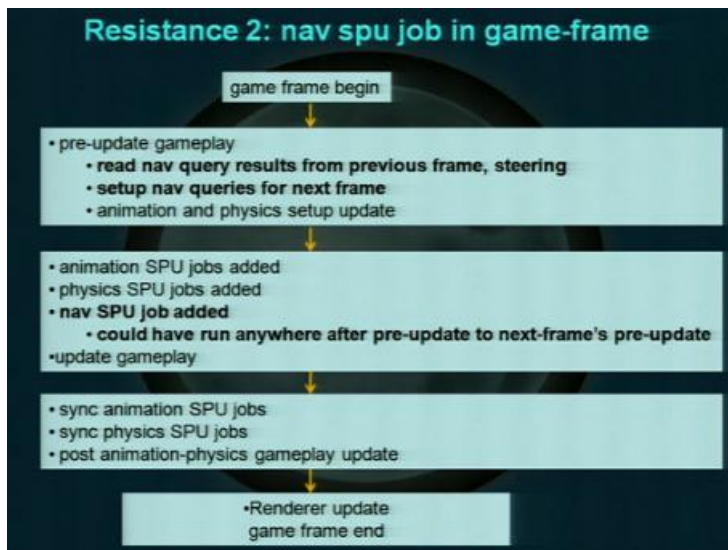
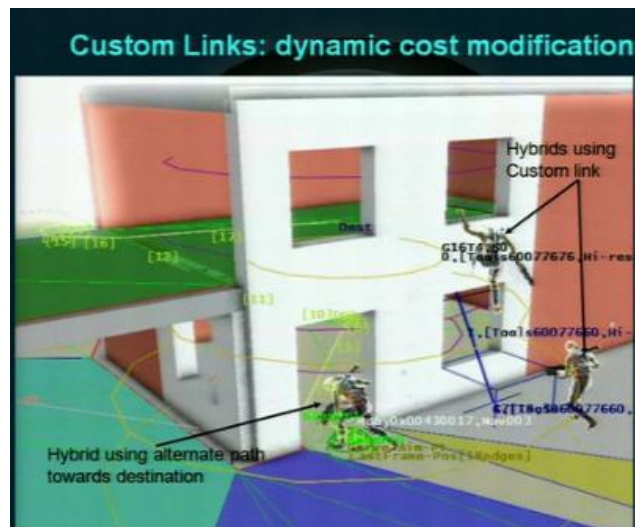
<http://www.insomniacgames.com/navigation-gdc11/>

# Insomniac Navigation Engine

## Insomniac PathEngine 特徴

- Resistance 2
- PS3 (SPU)に最適化
- カスタムリンク(手製リンク作成)
- Shader を使ってパスシステム構築

Maya手製から自動生成、さらにランタイム生成へ開発中

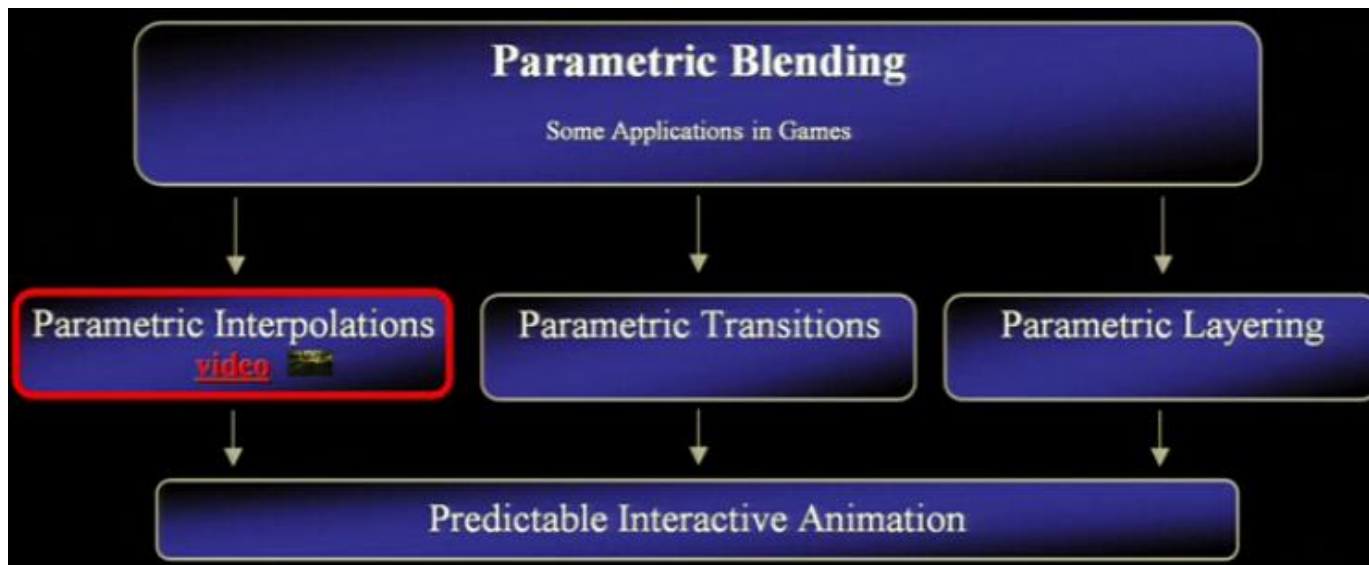


# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysic 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine -Frostbite Engine
- (6) 日常系AI
  - The Sims



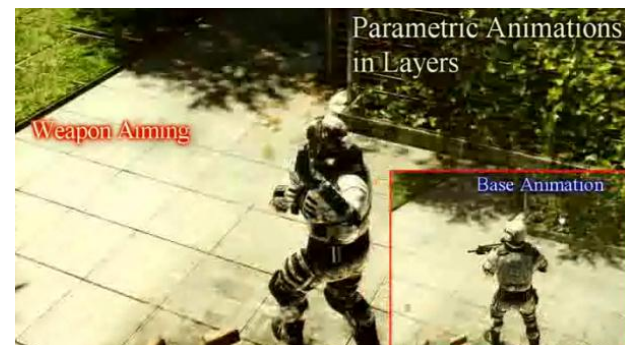
# Crysis 2 のアニメーション



ビヘイビア間の補完  
(スローラン、ファストラン)



ビヘイビアのチェンジ



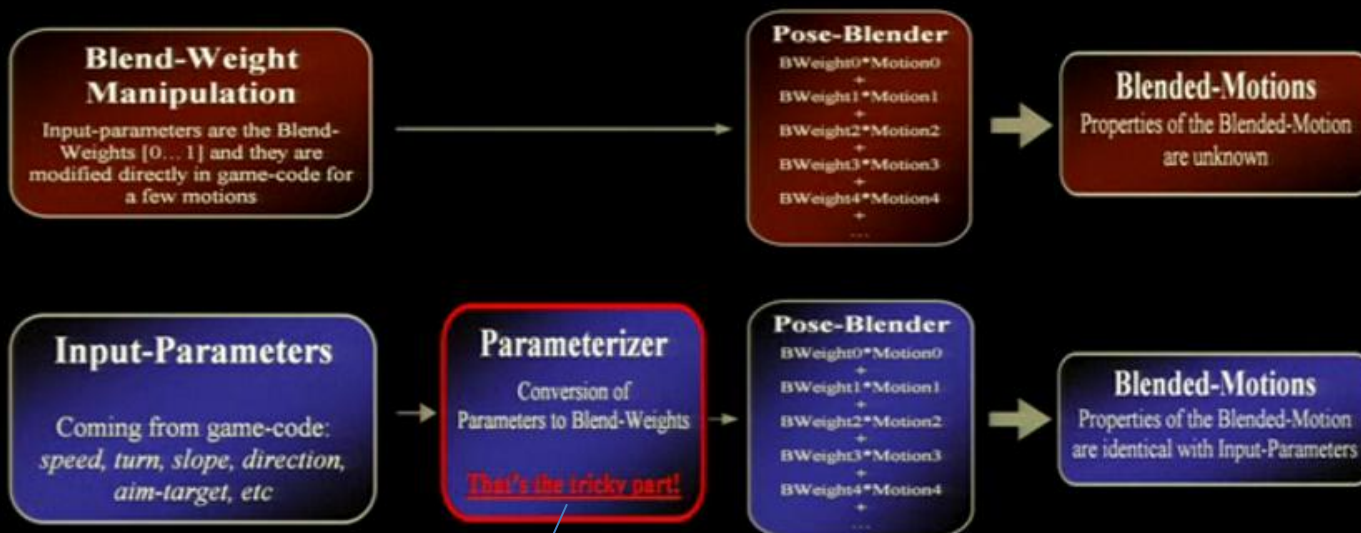
アニメーションの足し算

Ivo Herzeg (Crytek), CRYISIS 2: Getting More Interactivity out of Animation-Data  
<http://www.gdcvault.com/play/1014528/CRYISIS-2-Getting-More-Interactivity>



# Crysis 2 のアニメーション

## Animation vs Parametric Blending



この部分が違う。  
パラメタライザーが満たすべき要件。

### The 5 Features a Parameterizer must have!

1. Accurate Parameter Mapping
2. Artist-Directed Blending
3. Continuous-Control
4. Runtime Efficient
5. Memory Efficient

*Conclusion: if only one of these features is missing, then it's very hard to use it in game-productions.*

# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims

# Killzone 2



内容:クローズド、オープンワールドが混在する舞台での戦闘  
開発元: Guerrilla Games  
出版: SCE  
Hardware: PS3  
出版年: 2009年

# Killzone 2 の AI

**Based on:** Alex Champandard, Tim Verweij, Remco Straatman, "Killzone 2 Multiplayer Bots",  
[http://files.aigamedev.com/coverage/GAIC09\\_Killzone2Bots\\_StraatmanChampandard.pdf](http://files.aigamedev.com/coverage/GAIC09_Killzone2Bots_StraatmanChampandard.pdf)

# Killzone 2 AI (マルチプレイヤーモード)

(1) Killzone 2 のAI思考

(2) Killzone 2 のマップ自動解析





# Killzone 2 Screen





# Killzone 2 AI (マルチプレイヤーモード)

## (1) Killzone 2 のAI思考

## (2) Killzone 2 のマップ自動解析

### Scope

Killzone 2 / PS3

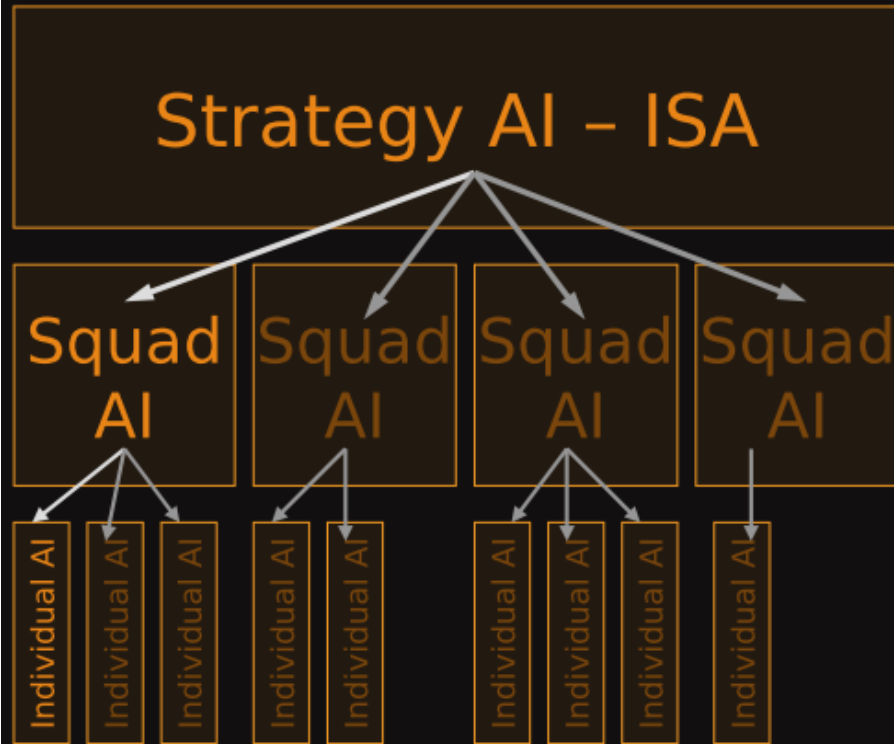
- Max 32 players
- Team-based game modes
- Multiple game modes on one map
- Players unlock / mix “badge abilities”
- Offline (1 human player & bots)
- Online (human players & bots)



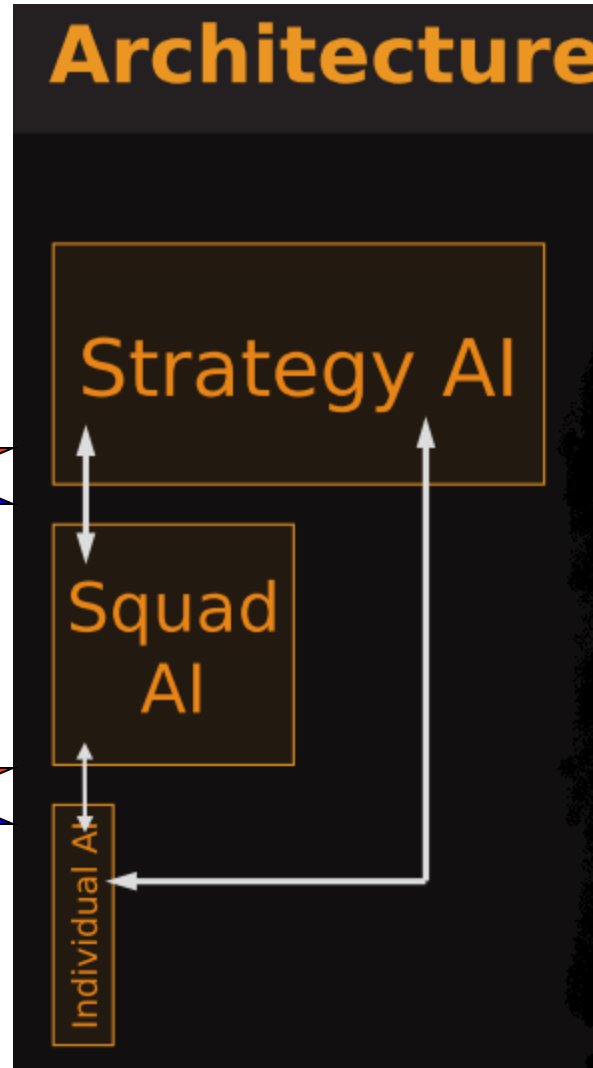
# Killzone2 AI

集団のAIのコントロールの仕方を考える

## Architecture



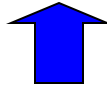
# Killzone2 AI



防衛、前進など戦術を指示



戦術の成功・失敗を報告  
(フィードバック)



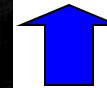
移動地点を指示



戦況を報告(フィードバック)



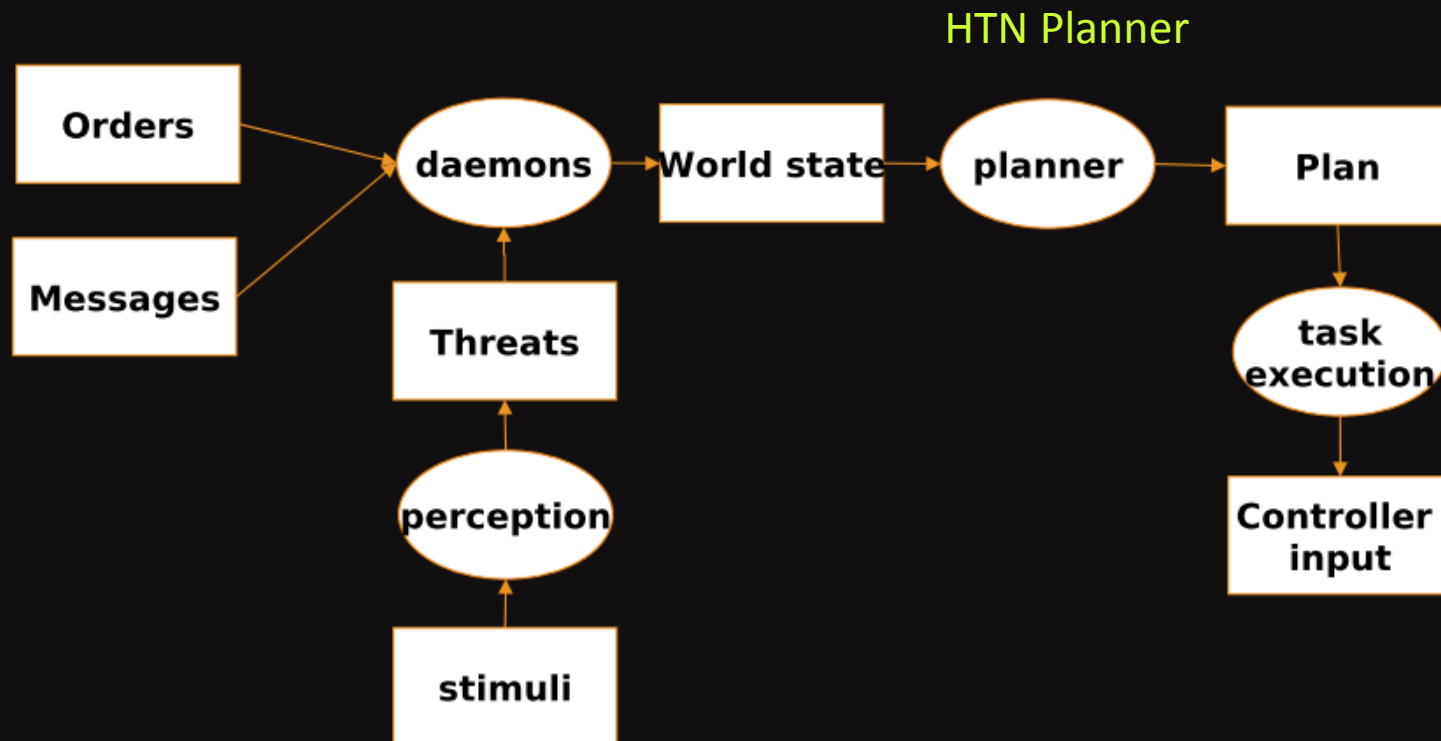
ターゲット指示



指示の再発行要求

# 各AIのアーキテクチャ

## Individual AI



# HTN (Hierarchical Task Network)

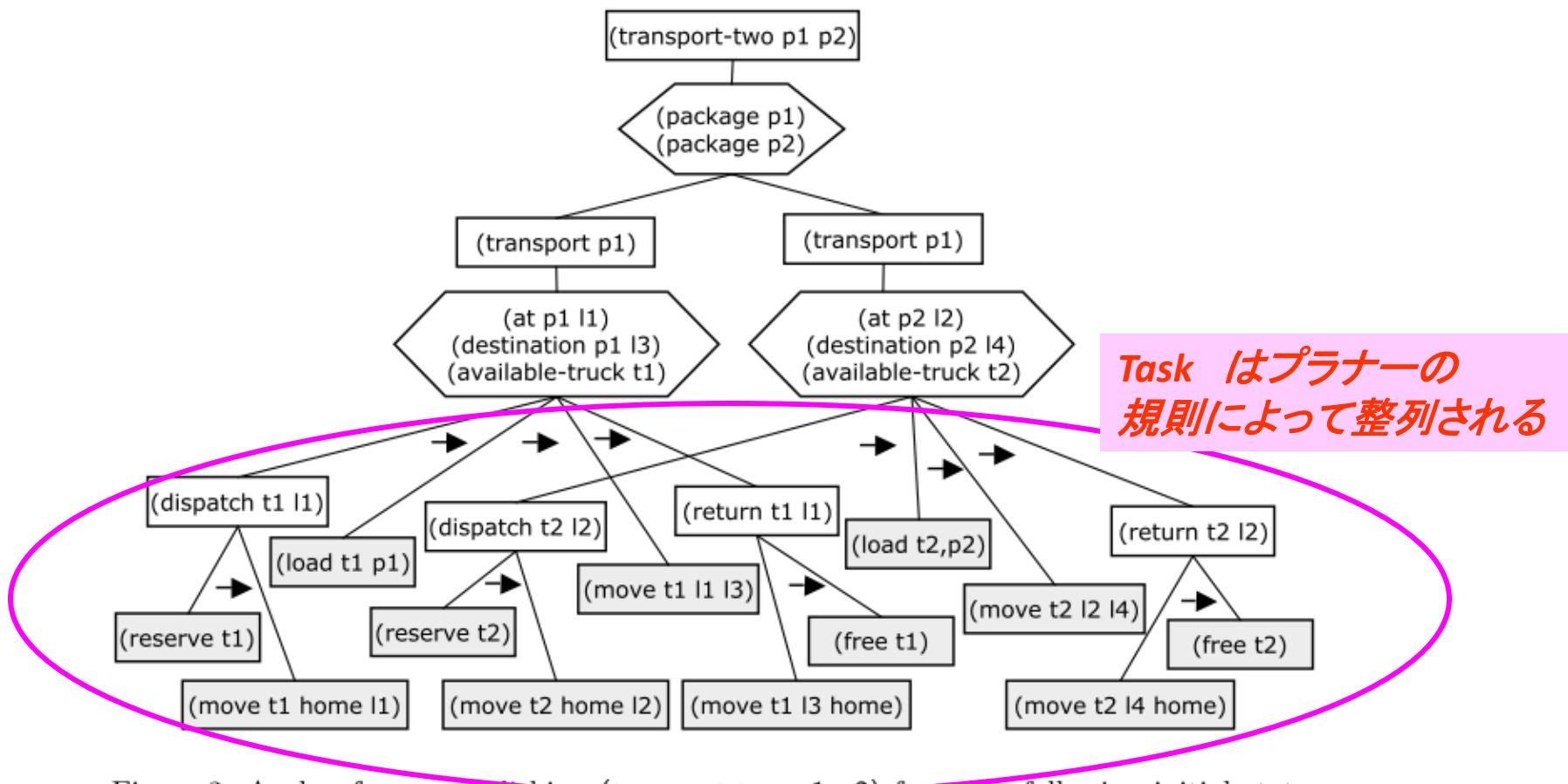
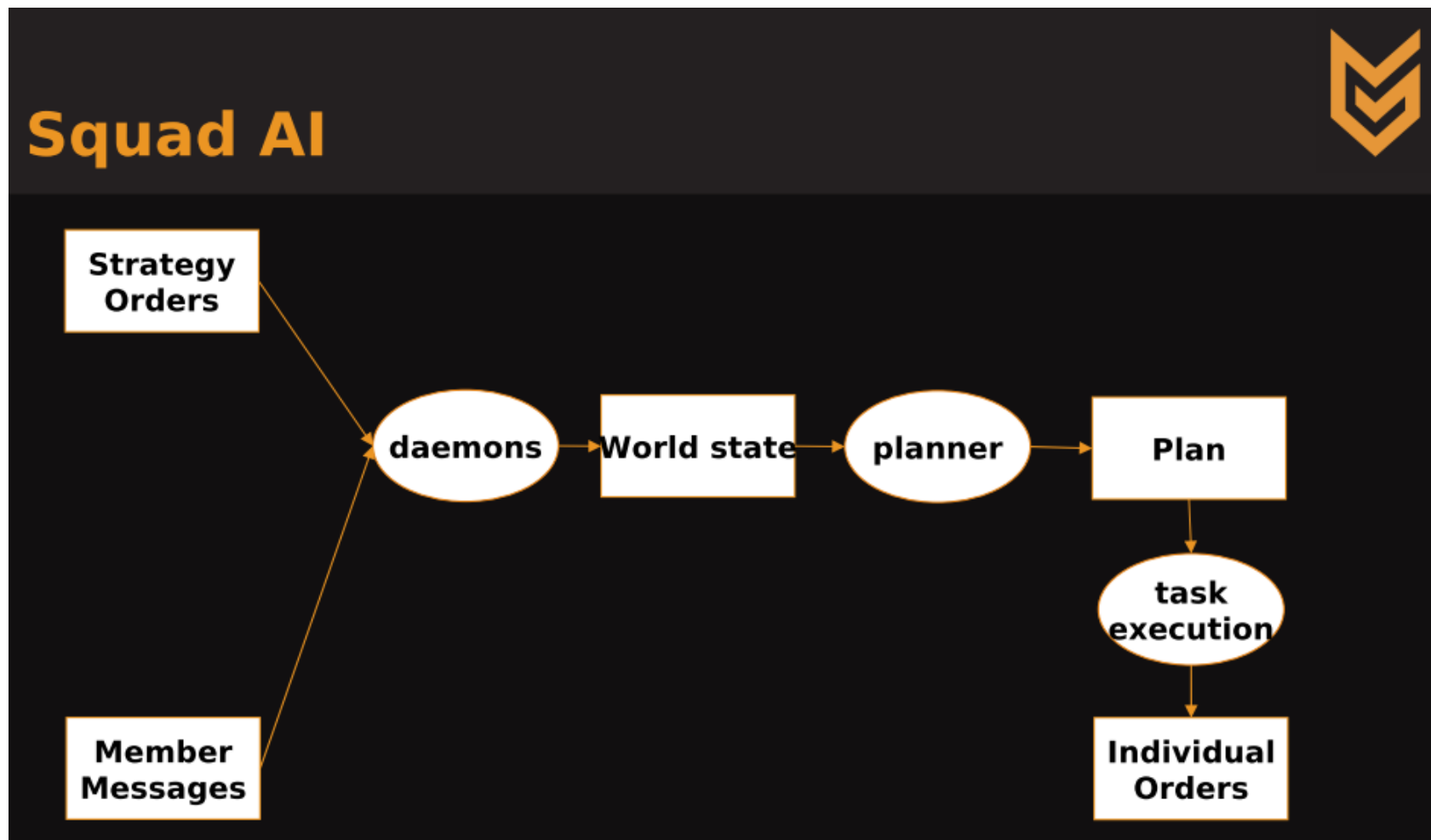


Figure 2: A plan for accomplishing (transport two p1 p2) from the following initial state:  
{(package p1), (at p1 l1), (destination p1 l3), (available-truck t1), (at t1 home),  
(package p2), (at p2 l2), (destination p2 l4), (available-truck t2), (at t2 home)}.

# Squad AI アーキテクチャ





# Killzone2におけるHTN統計

## Individual AI : Random Numbers



Individual bot domain

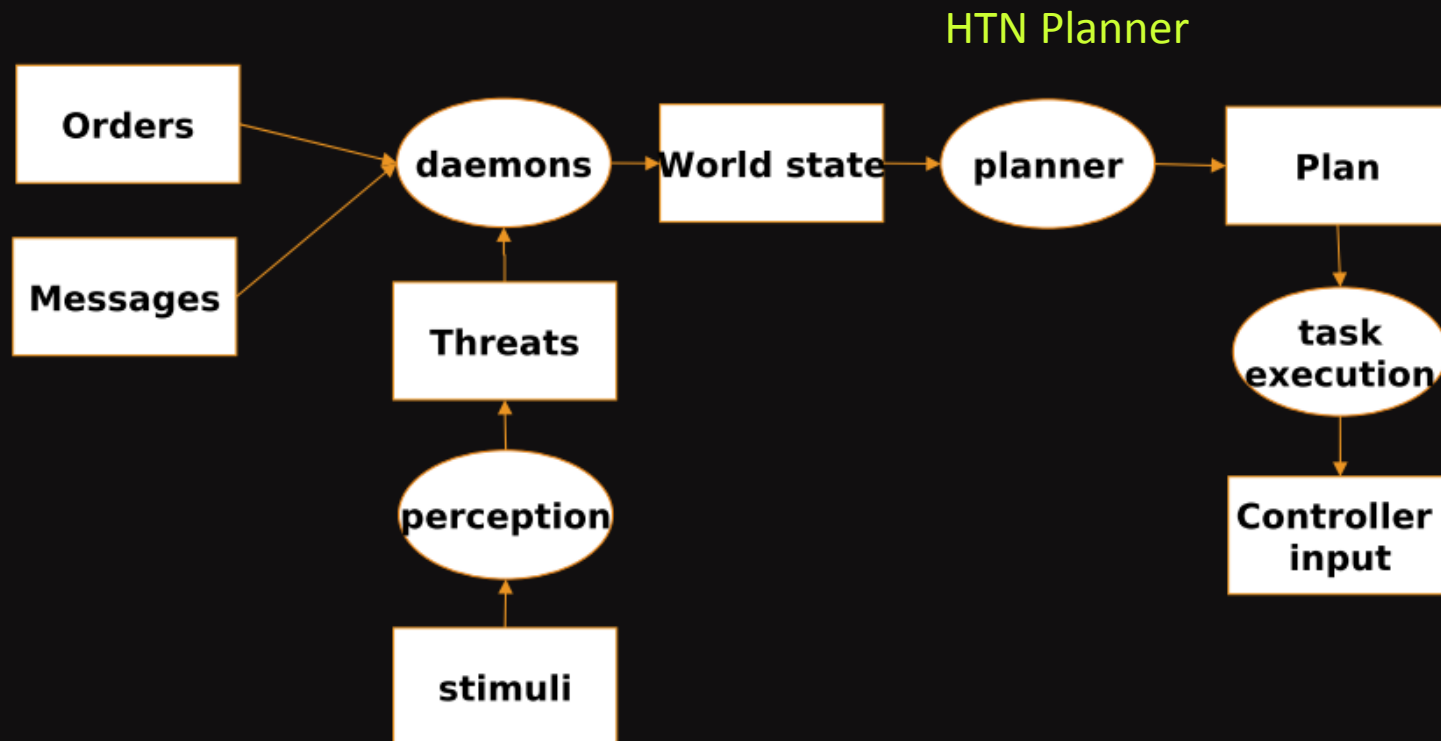
- 360 methods
- 1048 branches
- 138 behaviors
- 147 continue branches

During multiplayer game (14 bots / max. 10 turrets / max. 6 drones / squads)

- Approx. 500 plans generated per second
- Approx. 8000 decompositions per second
- Avg. 15 decompositions per planning.
- Approx 24000 branch evaluations per second.

# 各AIのアーキテクチャ

## Individual AI



# Squad AI



```
(:method (order_member_defend ?inp_member ?inp_id ?inp_level ?inp_marker ?inp_context_hint)
  ...
  (branch_advance
    ()
    (
      (!forget member_status ?inp_member **)
      (!remember - member_status ?inp_member ordered_to_defend ?inp_id)
      (!start_command_sequence ?inp_member ?inp_level 1)
      (do_announce_destination_waypoint_to_member ?inp_member)
      (!order ?inp_member clear_area_filter)
      (!order ?inp_member
        set_area_restrictions (call find_areas_to_wp ?inp_member (call get_entity_wp ?
inp_marker)))
      (!order_custom ?inp_member move_to_defend ?inp_marker ?inp_context_hint)
      (!order_custom ?inp_member send_member_message_custom completed_defend ?
inp_id)
      (order_set_defend_area_restriction ?inp_member (call get_entity_area ?inp_marker))
      (!order_custom ?inp_member defend_marker ?inp_marker ?inp_context_hint)
      (!end_command_sequence ?inp_member)
    )
  )
)
```

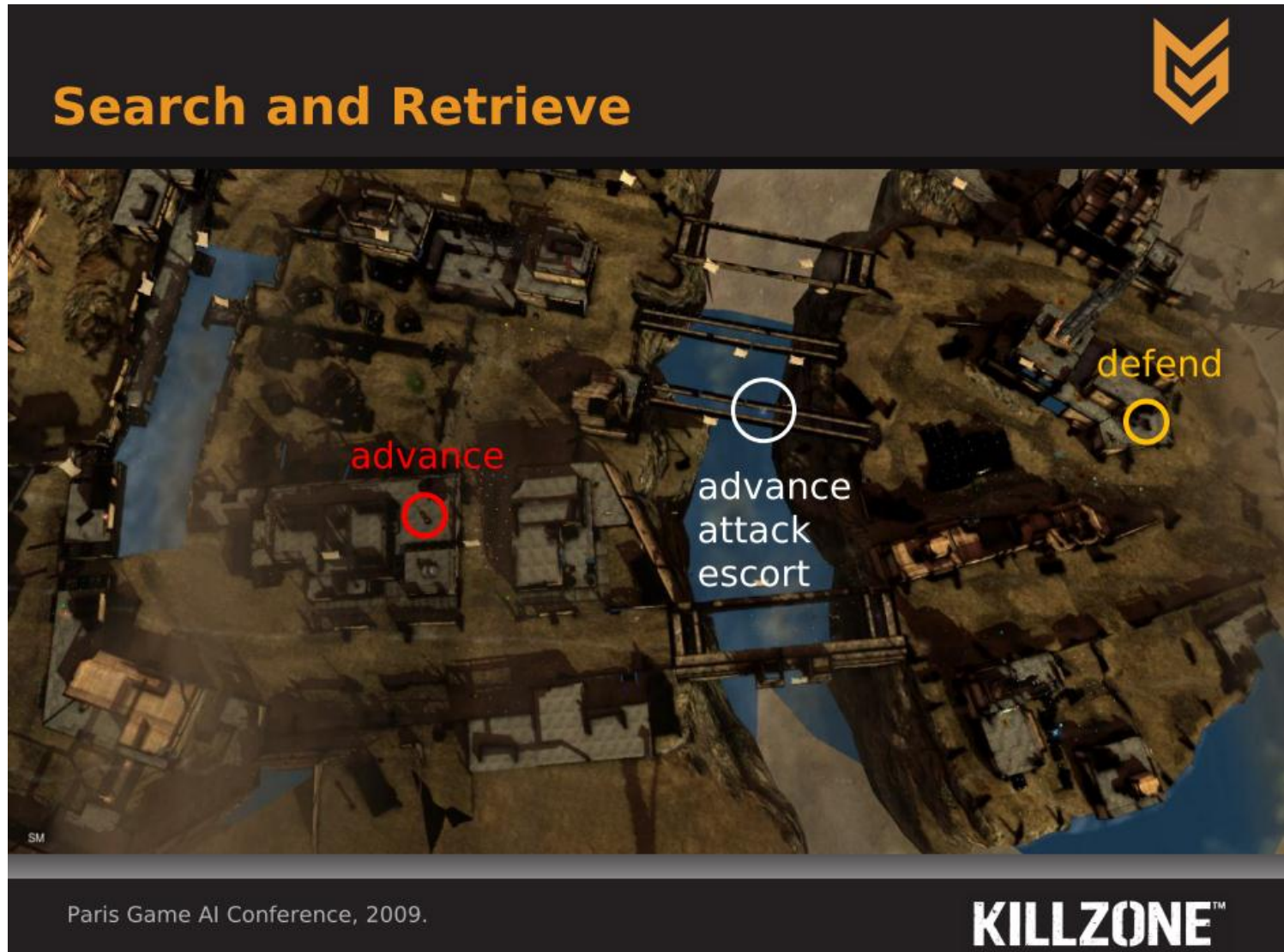
# Squad AI



```
(:method (order_member_defend ?inp_member ?inp_id ?inp_level ?inp_marker ?inp_context_hint)
  ...
  (branch_advance
    ()
    (
      (!forget member_status ?inp_member **)
      (!remember - member_status ?inp_member ordered_to_defend ?inp_id)
      (!start_command_sequence ?inp_member ?inp_level 1)
      (do_announce_destination_waypoint_to_member ?inp_member)
      (!order ?inp_member clear_area_filter)
      (!order ?inp_member
        set_area_restrictions (call find_areas_to_wp ?inp_member (call get_entity_wp ?
inp_marker)))
      (!order_custom ?inp_member move_to_defend ?inp_marker ?inp_context_hint)
      (!order_custom ?inp_member send_member_message_custom completed_defend ?
inp_id)
      (order_set_defend_area_restriction ?inp_member (call get_entity_area ?inp_marker))
      (!order_custom ?inp_member defend_marker ?inp_marker ?inp_context_hint)
      (!end_command_sequence ?inp_member)
    )
  )
)
```

兵士へ命令

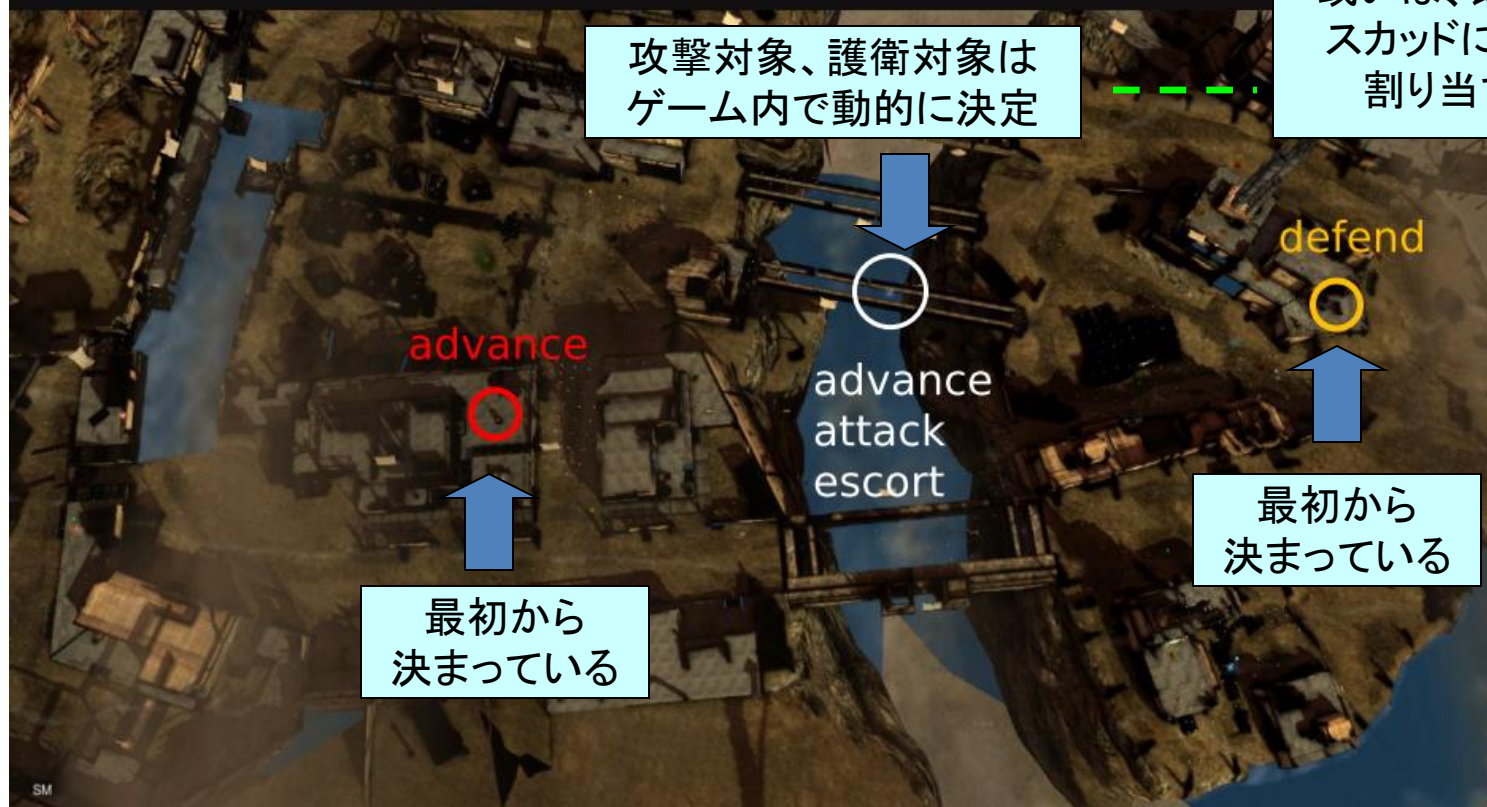
# スカッドを目的に応じて動的に構成





# スカッドを目的に応じて動的に構成

## Search and Retrieve



対象が決まれば  
対象からの距離など  
からスカッドを形成。  
或いは、既にある  
スカッドに対象を  
割り当てる。



# Killzone 2 AI (マルチプレイヤーモード)

(1) Killzone 2 の AI思考

(2) Killzone 2 のマップ自動解析

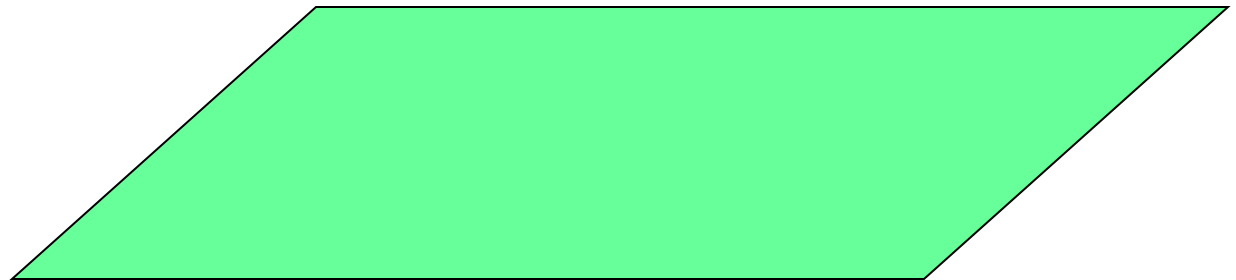
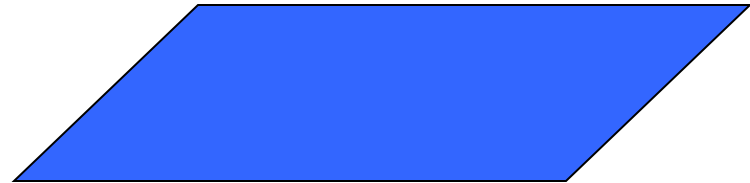
# マップ

AIの基本原則:異なる目的に、異なるデータ表現を作成する  
たとえば、同じ対象物であっても、AIの行動用途に応じてデータ表現を作れ。

戦術思考

戦略思考

ウェイポイント



特に、マップに関しては「データ階層化」しておくこと(世界表現)

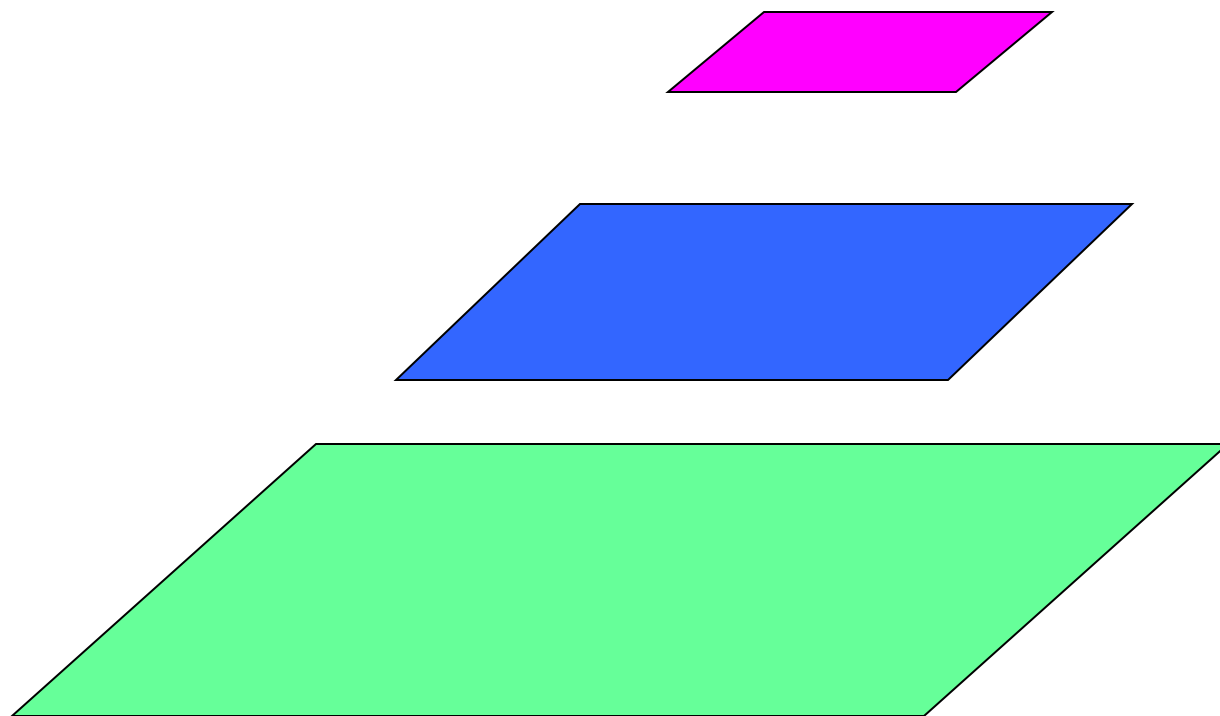
# Killzoneにおけるマップ

AIの基本原則:異なる目的に、異なるデータ表現を作成する  
たとえば、同じ対象物であっても、AIの行動用途に応じてデータ表現を作れ。

戦術思考

戦略思考

ウェイポイント



特に、マップに関しては「データ階層化」しておくこと(世界表現)

# 戦略グラフを作りたい

**Strategic Graph**

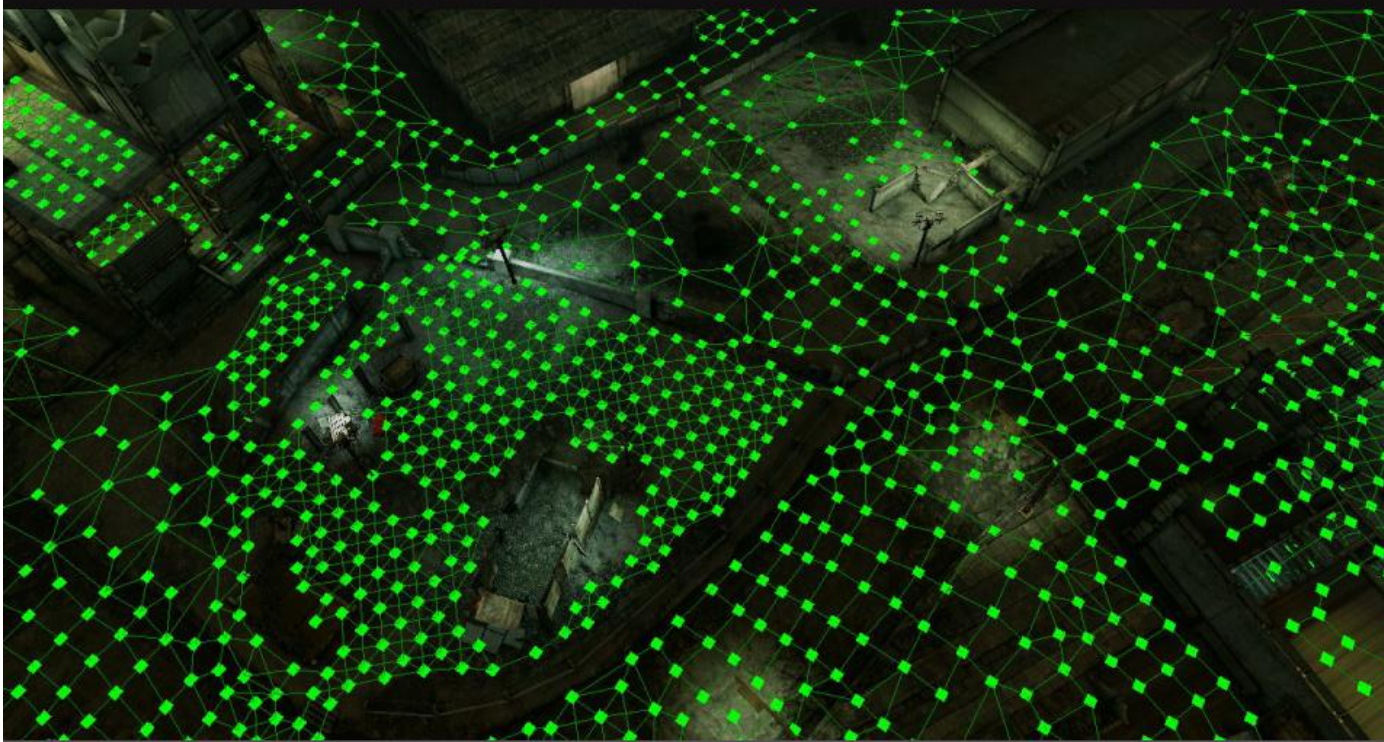


Paris Game AI Conference, 2009.

**KILLZONE™**

# Killzone 2 のウェイポイント

## Waypoint Network



Paris Game AI Conference, 2009.

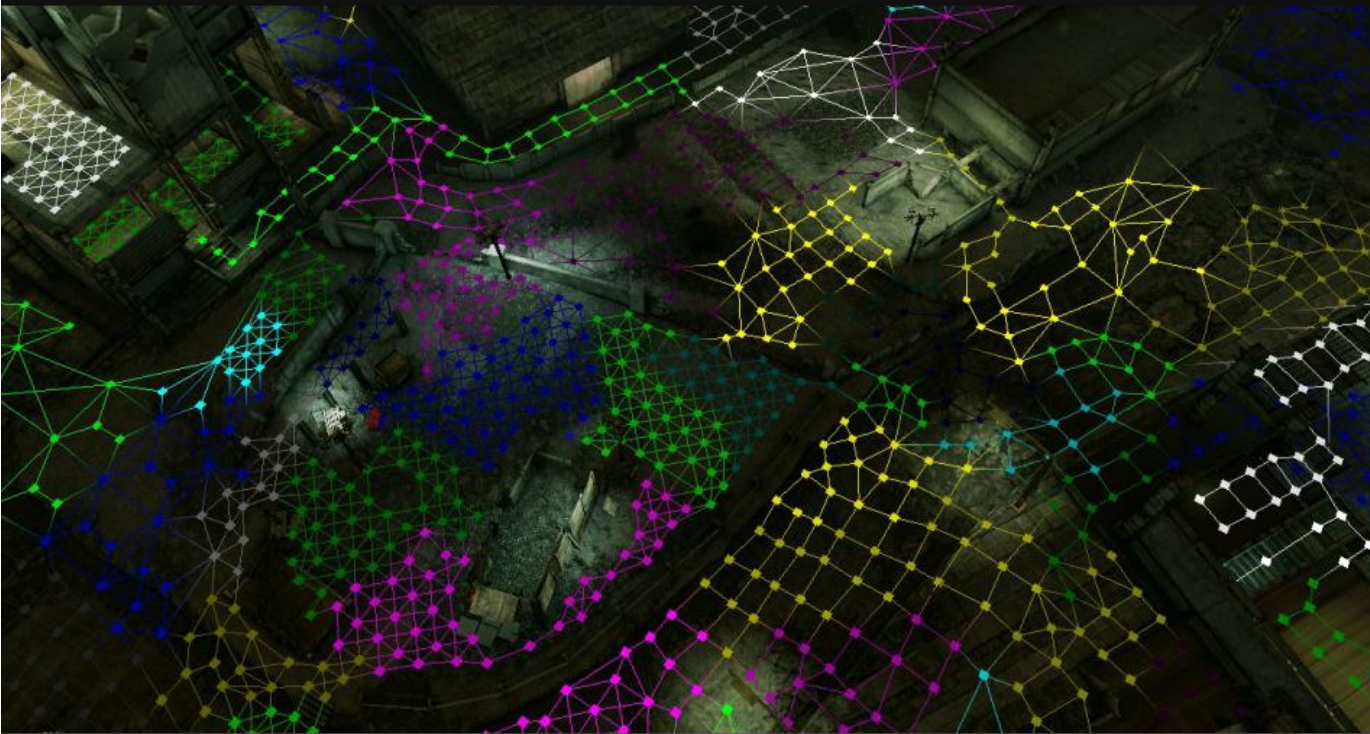
**KILLZONE™**



# ウェイポイントをクラスター化

サイズと含むウェイポイント数を決めて  
なるべく正方形になるように形成

## Waypoint Areas



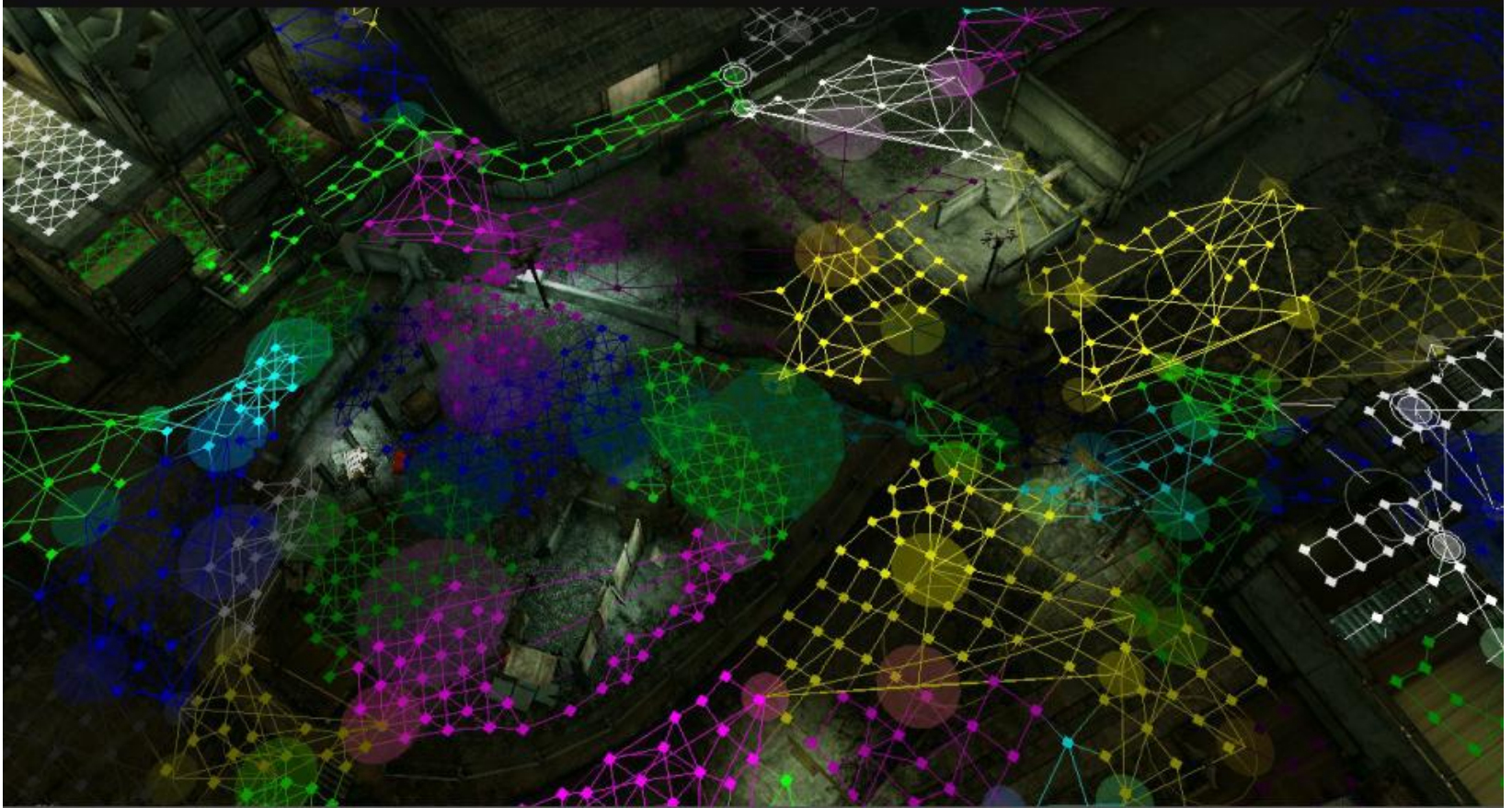
Paris Game AI Conference, 2009.

**KILLZONE™**



# 戦略グラフ

## Strategic Graph



Paris Game AI Conference, 2009.

**KILLZONE™**

# 戦略グラフ

各エリア間で接するポイントをピックアップして結ぶ

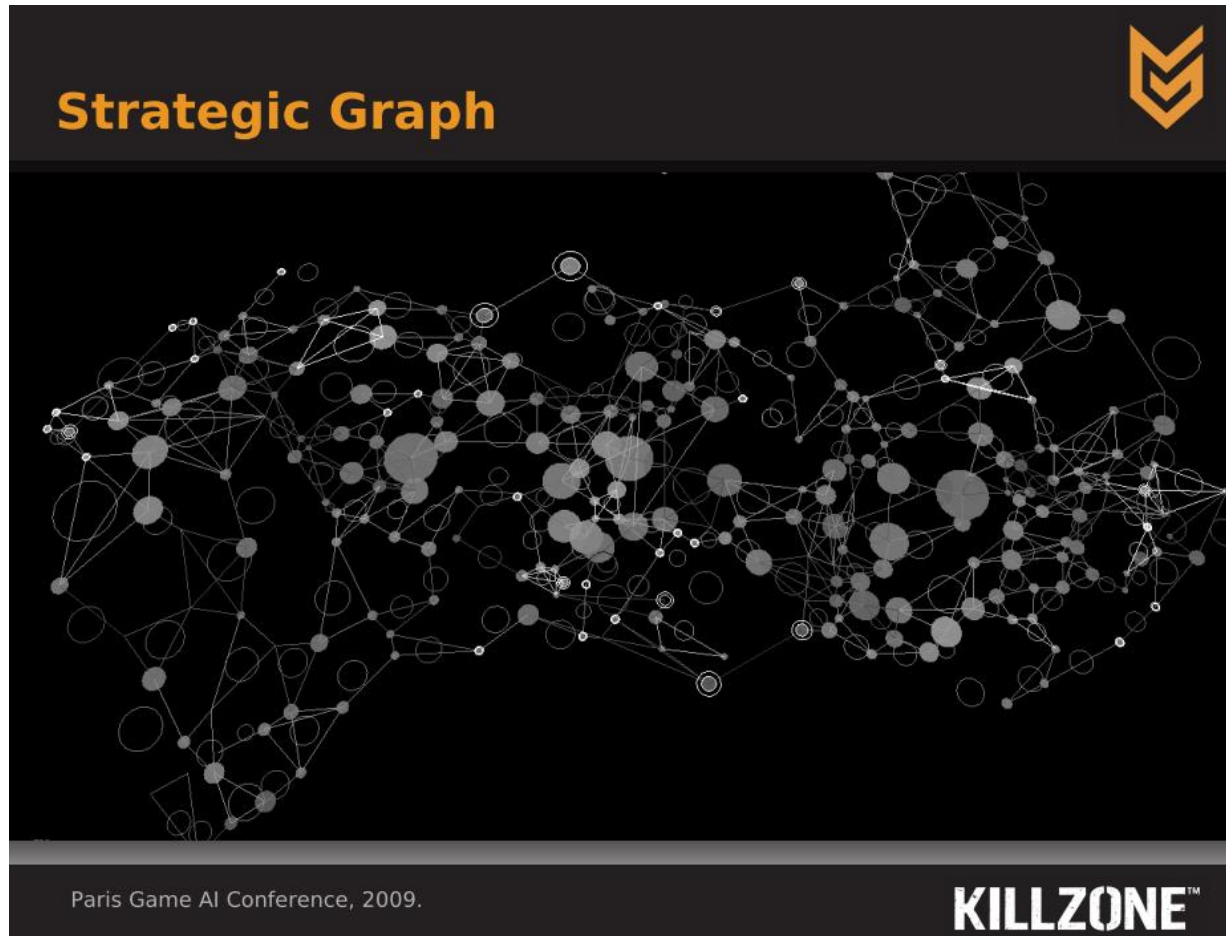
## Strategic Graph



Paris Game AI Conference, 2009.

**KILLZONE™**

# 戦略グラフ



何に使うのか？ グローバルな戦況判断

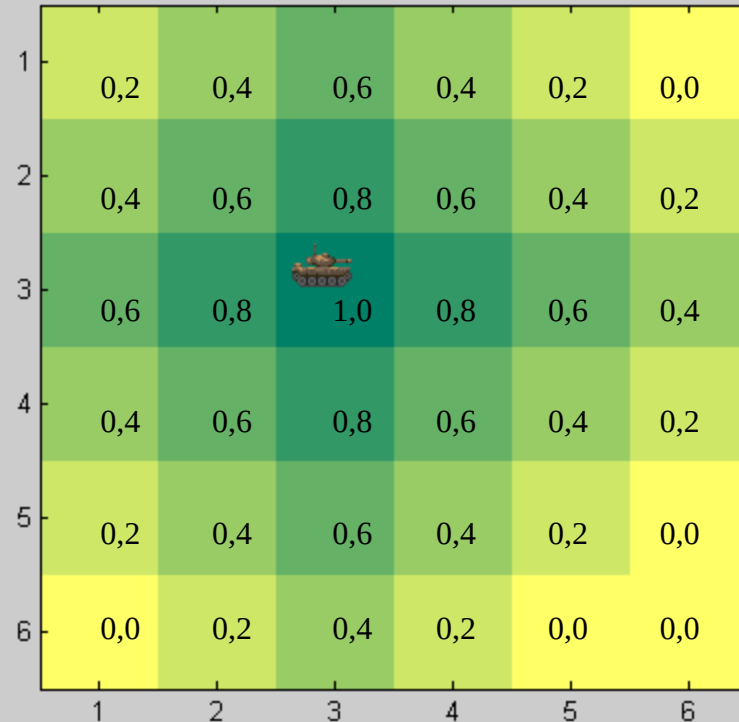


# 影響マップの方法 <解説>

# Influence Map(影響マップ)

セル分割されたマップに、問題とする性質の評価値を記録して行く方法

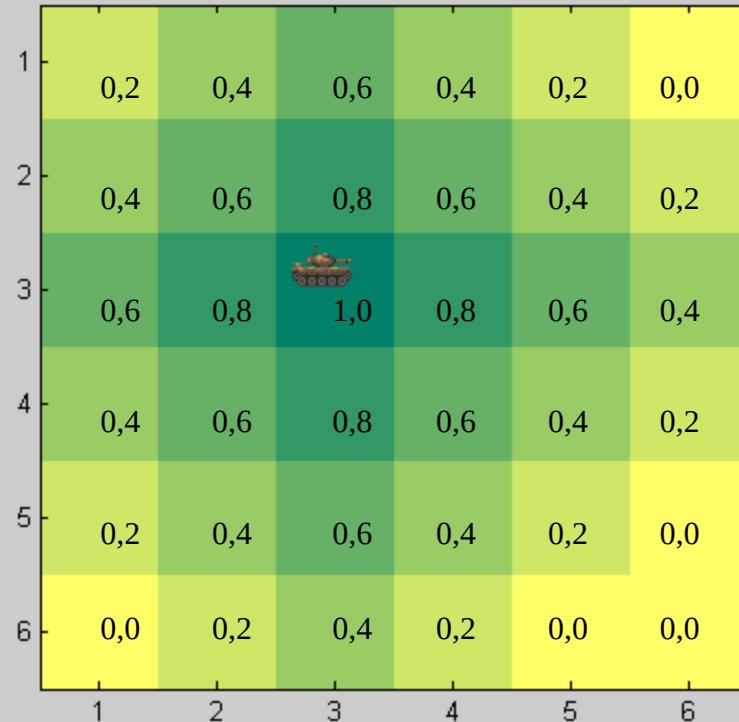
(例)①占有度



# Influence Map(影響マップ)

セル分割されたマップに、問題とする性質の評価値を記録して行く方法

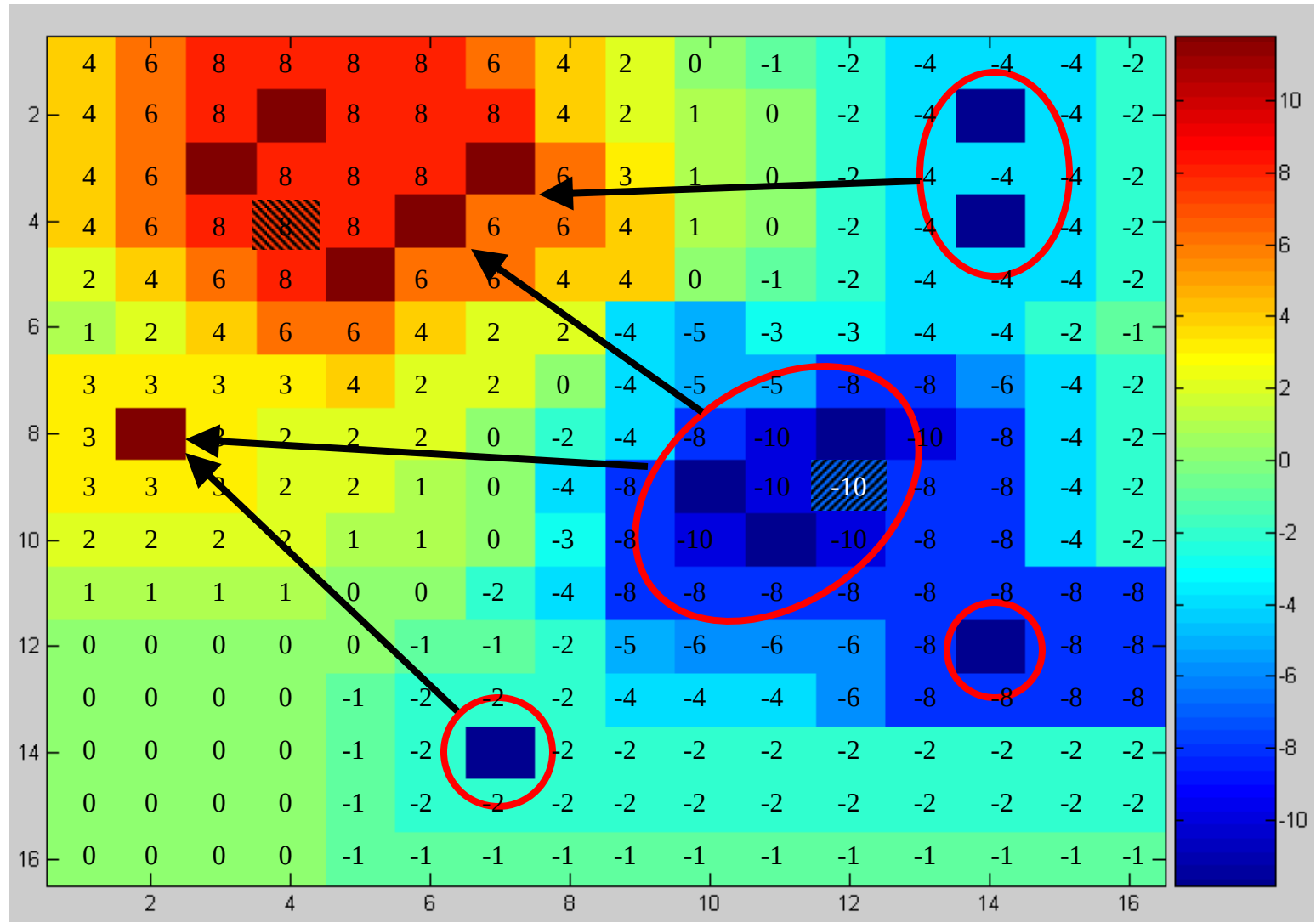
- (例) ① 占有度  
② 通過可能確率





# Influence Map(影響マップ)

(例) 様々なIMから計算して戦略的な位置取りを計算する



# Age of Empire(AOE)における利用例



AOEはタイルベースの  
RTS(リアルタイムストラテジー)

マップは毎回、自動生成

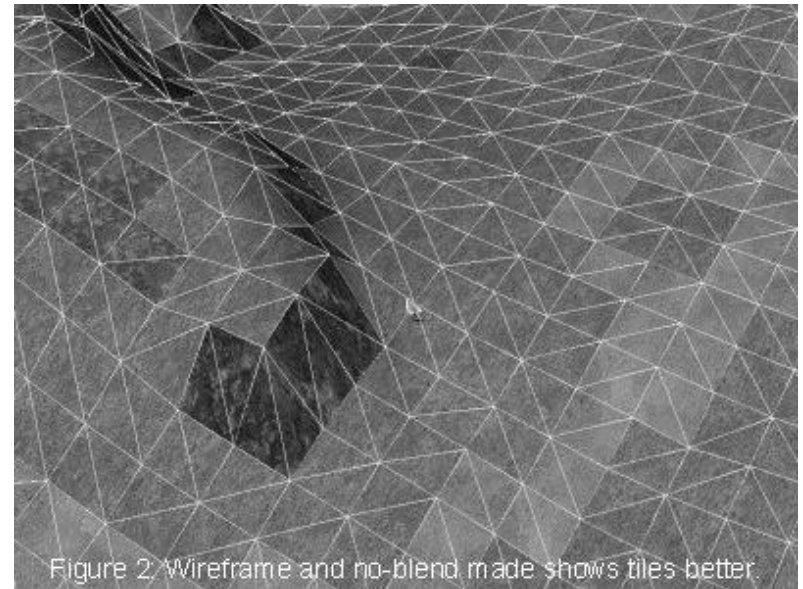


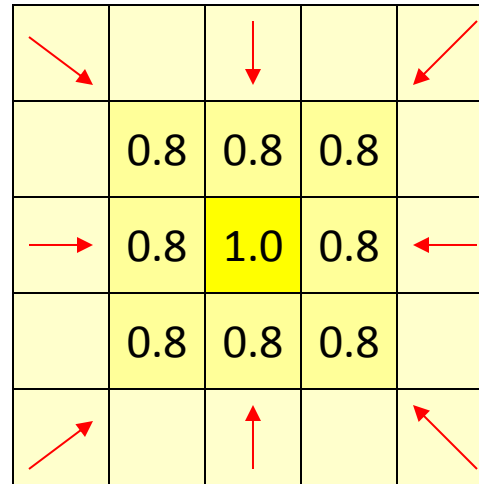
Figure 2: Wireframe and no-blend mode shows tiles better.

# Age of Empire(AOE)における利用例①

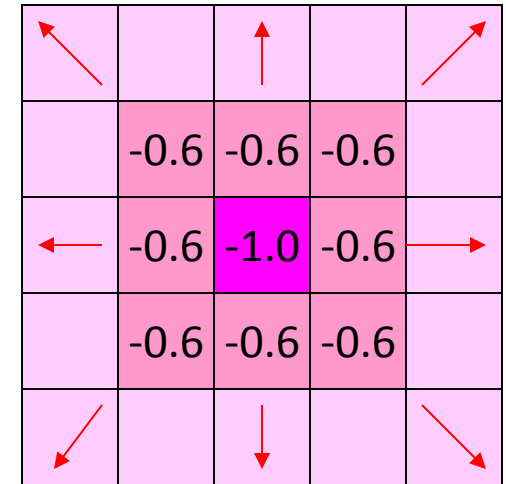
## 金鉱発掘小屋の最適位置自動検出



Figure 3: Influence map around gold mines.



金鉱＝アトラクター  
(吸引源)



防壁＝デトラクター  
(排斥源)

AIに金鉱の発掘小屋を適切な位置に置かせるには？

＝ 金鉱に近いが、近すぎではいけないところに置く

金鉱の回りのタイル(仮想的)防壁を置いて、  
近すぎも遠すぎもしない点に最高点のタイルが来るようにする

＝ 自動位置検出

# Age of Empire(AOE)における利用例②

## 兵隊の配置の最適位置自動検出



|  |     |     |     |  |
|--|-----|-----|-----|--|
|  |     |     |     |  |
|  | 0.8 | 0.8 | 0.8 |  |
|  | 0.8 | 1.0 | 0.8 |  |
|  | 0.8 | 0.8 | 0.8 |  |
|  |     |     |     |  |

高い場所＝アトラクター  
(吸引源)

|  |      |      |      |  |
|--|------|------|------|--|
|  |      |      |      |  |
|  | -0.6 | -0.6 | -0.6 |  |
|  | -0.6 | -1.0 | -0.6 |  |
|  | -0.6 | -0.6 | -0.6 |  |
|  |      |      |      |  |

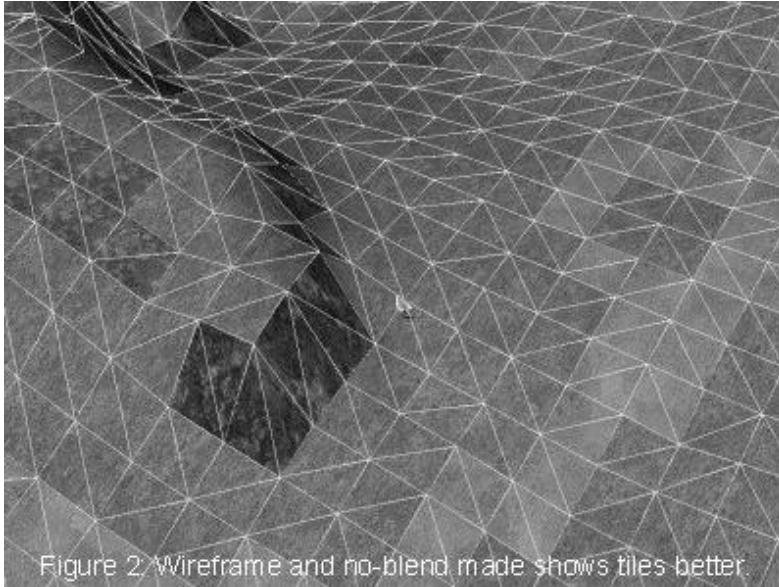
家がある場所＝デトラクター  
(排斥源)

兵士を街からなるべく遠く、高台の場所に置かせたい  
＝ 高い場所(アトラクター)、家がある(デトラクター)とする



# Age of Empire(AOE)における利用例③

## 兵隊の配置の最適位置自動検出

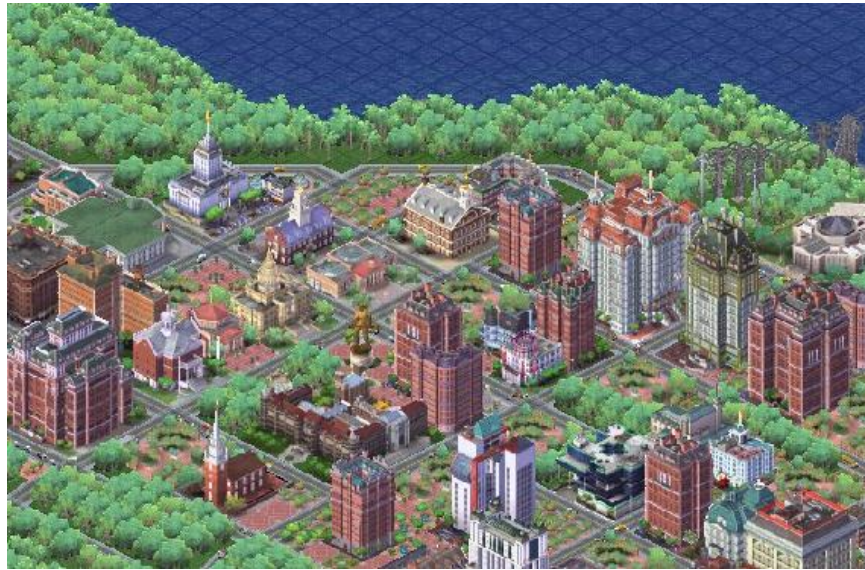


|                                                                                     |      |                                                                                     |      |                                                                                     |
|-------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------|
|  |      |  |      |  |
|                                                                                     | -0.6 | -0.6                                                                                | -0.6 |                                                                                     |
|  | -0.6 | -1.0                                                                                | -0.6 |  |
|                                                                                     | -0.6 | -0.6                                                                                | -0.6 |                                                                                     |
|  |      |  |      |  |

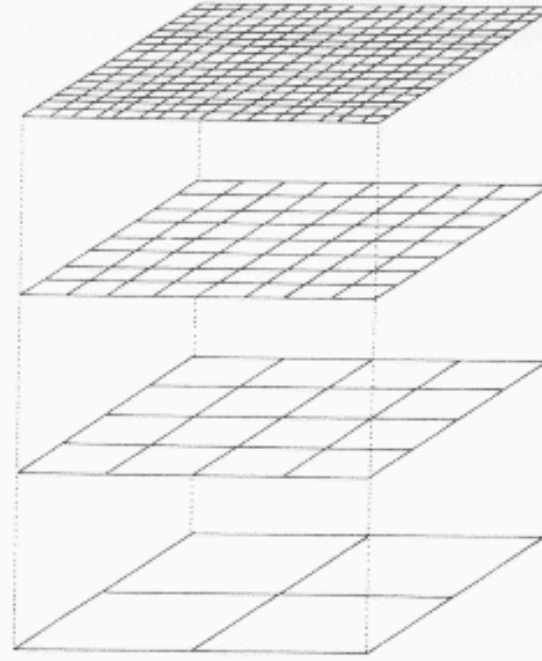
既知の敵のビル、以前通ったルート = デトラクタ  
(排斥源)

この排斥源が与えるコストをパス検索のコストとすると、  
「なるべく遠ったことのない敵の基地からみづかりにくいパス」  
を見つけてることが出来る。

# SimCity における影響マップの利用



シムシティーの多層構造モデル



第一層  
道路や鉄道、要素の大きさ要素間の関係をシミュレーション

第二層  
「人口密度」「交通渋滞」「環境汚染度」「ランドバリュー」「犯罪発生率」をシミュレーション

第三層  
地形の影響をシミュレーション

第四層  
「人口増加率」「消防署」「警察署」「消防署の影響」「警察署の影響」をシミュレーション

$$\text{Crime} = \text{Pop. Density}^2 - \text{Land Value} - \text{Police Effect}$$

$$\text{Land Value} = \text{Distance}[\text{Zonetype}] + \text{Terrain} + \text{Transport}$$

“AI: A Desing Perspective” AIIDE 2005

<http://www.aiide.org/aiide2005/talks/index.htm>

!

<http://thesims.ea.com/us/will/>

多摩 豊, 角川書店, 1990

「ウィル・ライトが明かすシムシティーのすべて」,

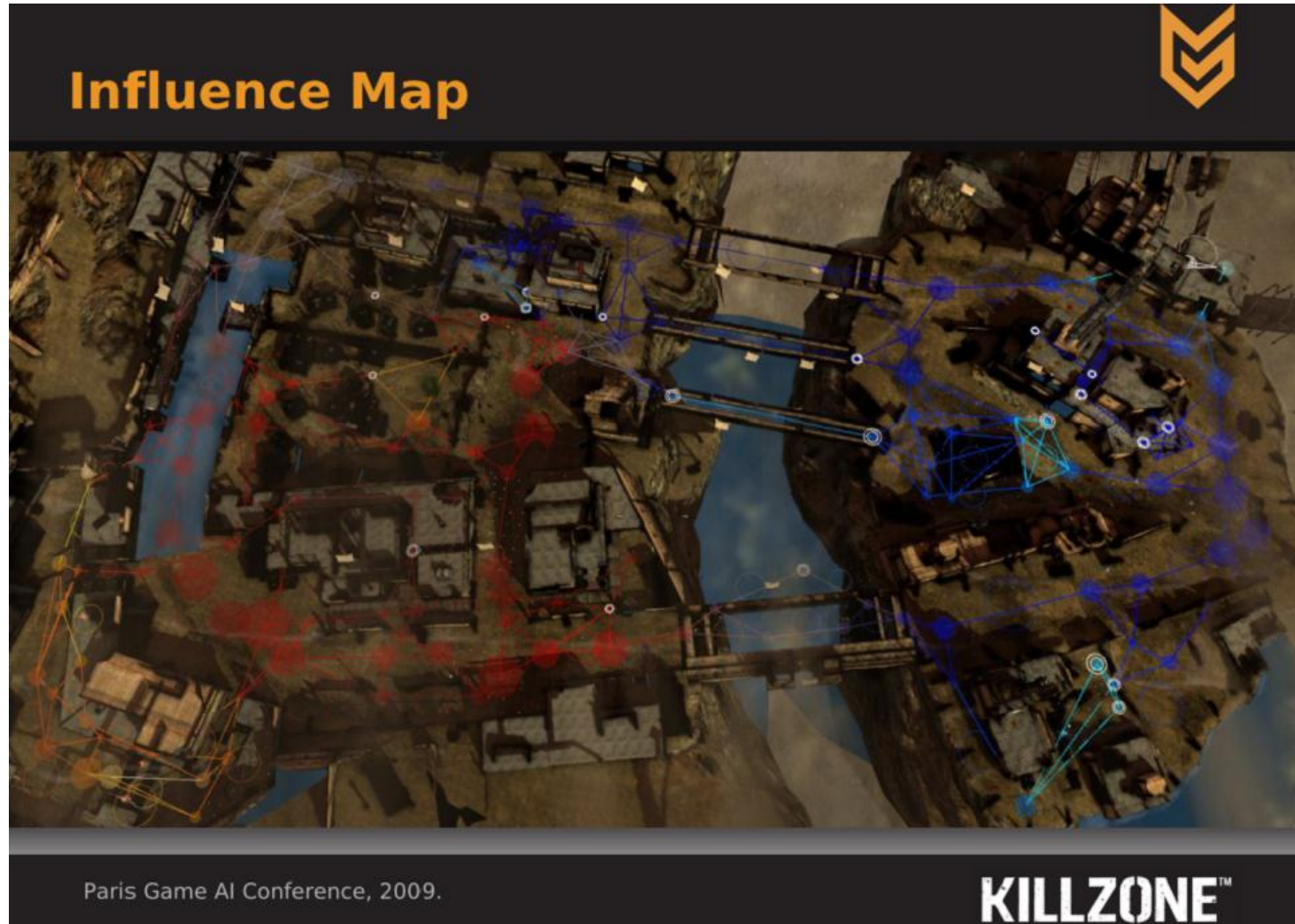


影響マップの方法 <解説>終わり

# Killzone 2 における影響マップ

ボット、砲塔、死亡ポイントの情報を反映させる

さまざまな意思決定に用いる



# Killzone 2 における戦略的パス検索

スカッド毎にコストと影響マップを使って戦略パスを見出す

## Strategic Pathfinding Costs





# Killzone 2 における戦略的パス検索

スカッド毎にコストと影響マップを使って戦略パスを見出す

## Strategic Pathfinder



# Killzone 2 における戦略的パス検索

スカッド毎にコストと影響マップを使って戦略パスを見出す

## Strategic Pathfinder

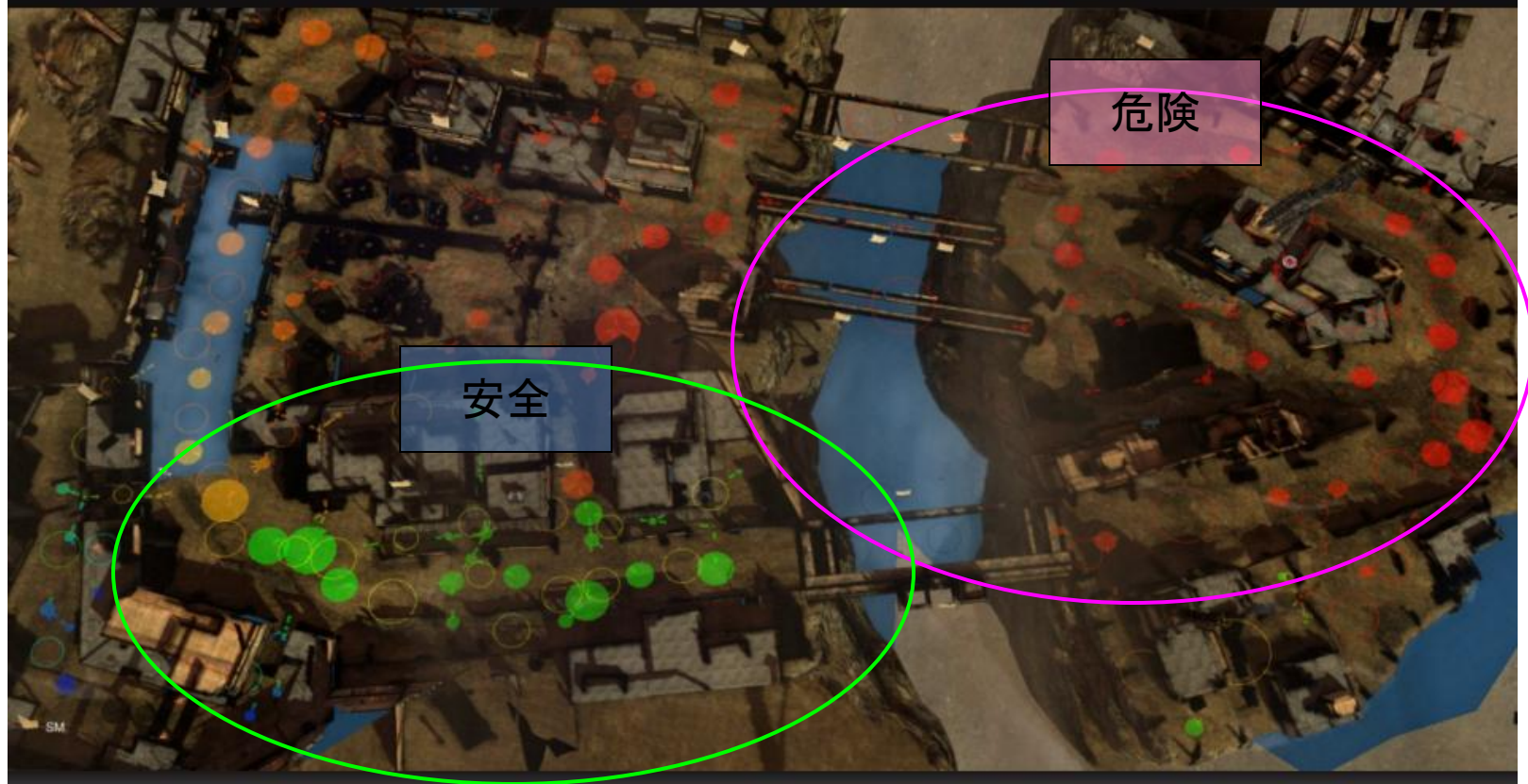




# Killzone 2 における戦略的パス検索

スカッド毎にコストと影響マップを使って戦略パスを見出す

## Strategic Pathfinder





# Killzone 2 における戦略的パス検索

スカッド毎にコストと影響マップを使って戦略パスを見出す

## Strategic Pathfinder



# Killzone 2 AIのまとめ

- (1) 各AI、スカッドAIに階層型タスクネットワークプランナー（HTN Planner）を導入した。
- (2) 戦略マップを自動生成した（マップを階層化した）。
- (3) 戦略マップ上に影響マップの方法を導入して、スカッドの戦略思考、戦略的パス検索を導入した。

# 目次

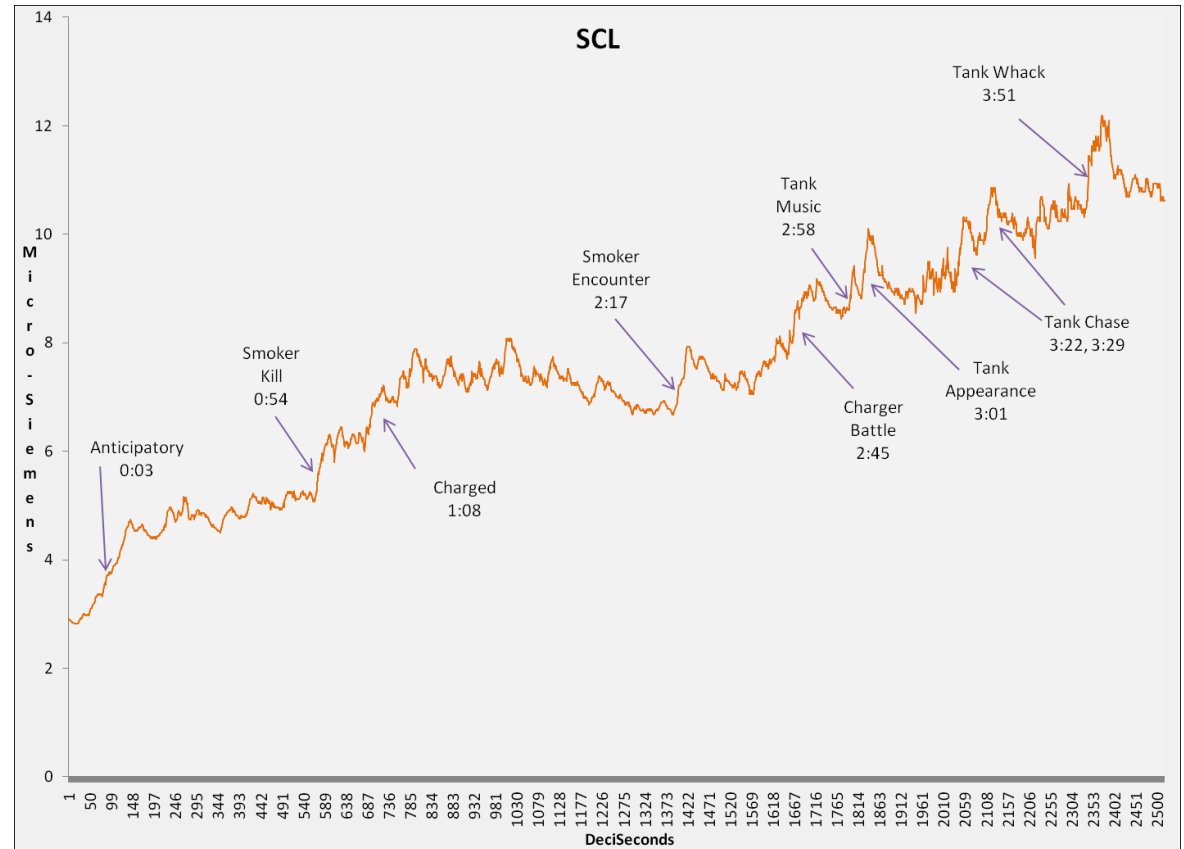
- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims

# Valve – Left 4 Dead 2における生体情報フィードバック

## 実例① Left 4 Dead 2 における状況とSCLの相関

### 生体情報

- 心拍数
- 皮膚電流
- 表情
- 眼球の動き
- など

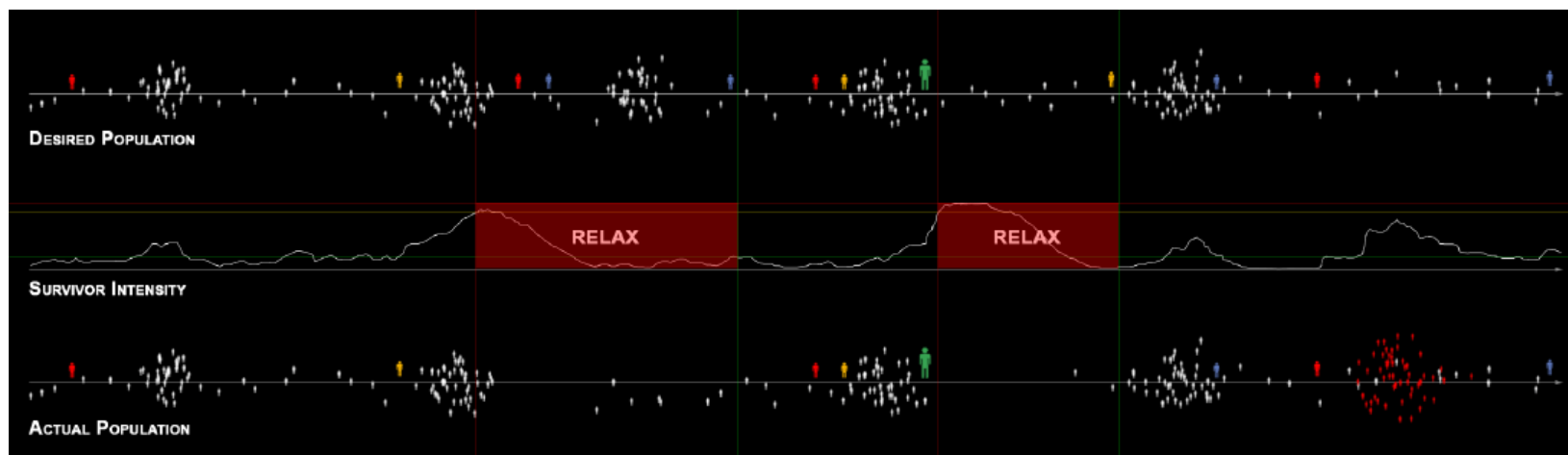


## Left 4 Dead 2 における状況とSCLの相関

SCL(皮膚電流を測定)

Mike Ambinder. March 2011. "Biofeedback in Gameplay: How Valve Measures Physiology to Enhance Gaming Experience." Game Developers Conference.  
<http://www.valvesoftware.com/company/publications.html>

# Left 4 Dead のメタAIが取得する情報



Michael Booth, "Replayable Cooperative Game Design: Left 4 Dead," Game Developer's Conference, March 2009

<http://www.valvesoftware.com/publications.html>



# Valve – Left 4 Dead 2における生体情報フィードバック

実例② Alien Swarm におけるSCLデータを制限タイムに反映

## 生体情報

- 心拍数
  - 皮膚電流
  - 表情
  - 眼球の動き
- など



SCL アップ  
SCLダウン



タイマースピードアップ  
タイマースピーダウン

熱中度をコントロール

SCL(皮膚電流を測定)

Mike Ambinder. March 2011. "Biofeedback in Gameplay: How Valve Measures Physiology to Enhance Gaming Experience." Game Developers Conference.  
<http://www.valvesoftware.com/company/publications.html>

# Valve – Left 4 Dead 2における生体情報フィードバック

## 実例③ Portal における Eye Control

### 生体情報

- 心拍数
  - 皮膚電流
  - 表情
  - 眼球の動き
- など



視線を測定



視線でターゲティング

Mike Ambinder. March 2011. "Biofeedback in Gameplay: How Valve Measures Physiology to Enhance Gaming Experience." Game Developers Conference.  
<http://www.valvesoftware.com/company/publications.html>

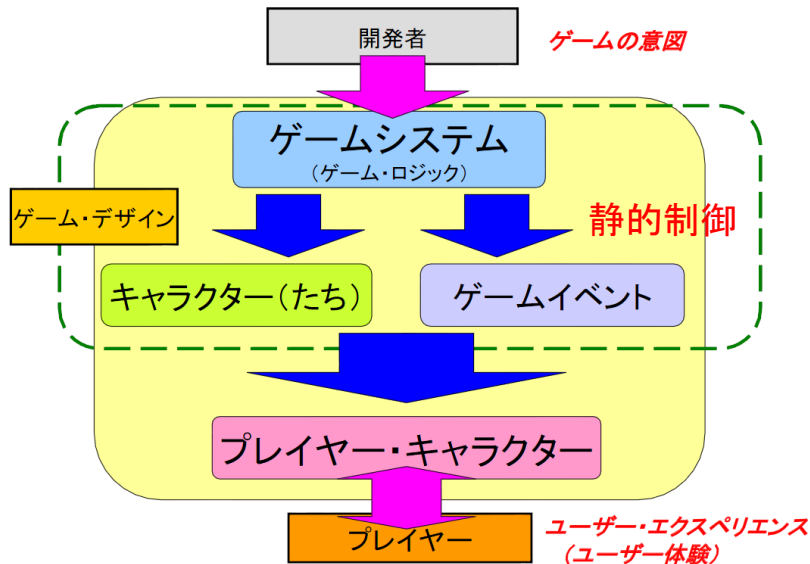
# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine -Frostbite Engine
- (6) 日常系AI
  - The Sims

# メタAI技術

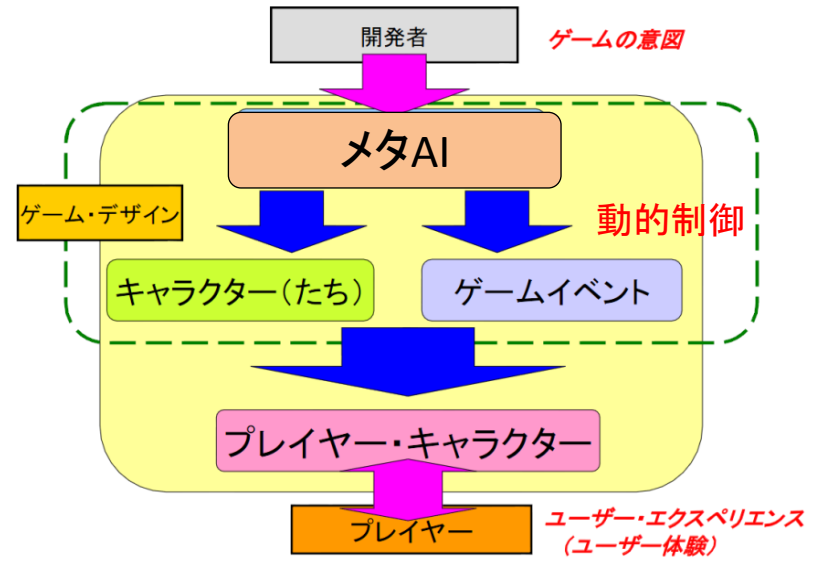
- メタAIとは、ゲーム状態を監視(認識)し、ゲームを動的に変化させるシステムである。
- 古くは、動的にレベルを変化させるシステムであったが、近年は、より積極的にプレイヤーのスキルや経路を予測して適切な敵を適切な場所に配置するなどの例がある。

デジタルゲームの簡易モデル



ゲームシステム(ゲーム・ロジック)を決めるとコンテンツが固定化される。ゲーム・ロジックを増やす=コンテンツが増える。

デジタルゲームの簡易モデル

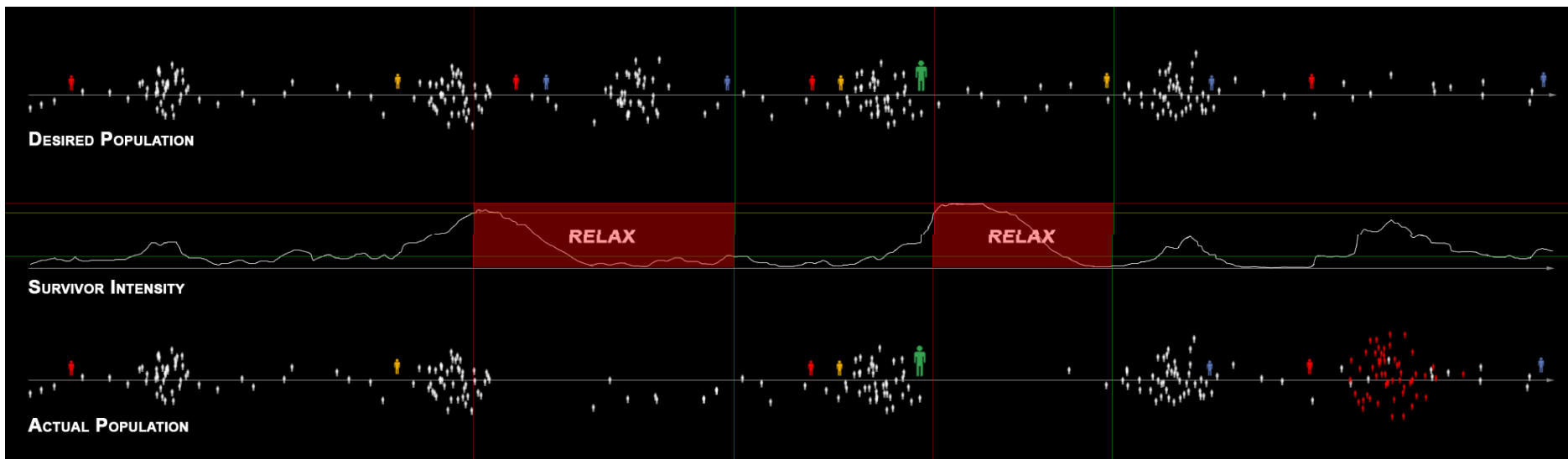


メタAIがゲームとプレイヤーの状況、性質を認識しながら、動的にゲームを変化させる。

いわばゲームデザイナーの分身をゲームの中に実装する



# Left 4 Dead におけるメタAI



プレイヤーチームの緊張度を(ある計算式: 非公開)で計算する。  
緊張度が一定値を超えるまで、敵を出現させ続ける。  
緊張度が一定値を超えると、敵を減少させる。

緩急をつける＝**ペーシング・アルゴリズム** (Pacing Algorithm)

Michael Booth, "Replayable Cooperative Game Design: Left 4 Dead," Game Developer's Conference, March 2009

<http://www.valvesoftware.com/publications.html>



# Left 4 Dead におけるメタAI

メタAIにゲーム状況を認識させる



L4D のナビゲーション・メッシュ

通常、ナビゲーションメッシュは、  
キャラクターをナビゲーションするのと、  
キャラクターが環境を認識するために使う。



L4Dでは、メタAIがプレイヤーの場所を追跡  
するために使う。

# Left 4 Dead におけるメタAI

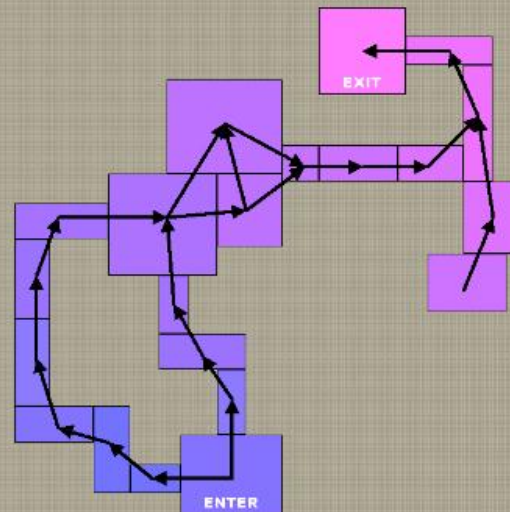
メタAIにゲーム状況を認識させる



AAS(Active Area Set = プレイヤーたちがいる領域周辺)を  
認識しプレイヤーの経路を予測する

## ➤ "Flow Distance"

- Travel distance from the starting safe room to each area in the navigation mesh
- Following increasing flow gradient always takes you to the exit room
- "Escape Route" = shortest path from start safe room to exit
- Used as a metric for populating enemies and loot, and for answering questions such as "is this spot ahead or behind the Survivor group"



# Left 4 Dead におけるメタAI

メタAIにゲーム状況を認識させる

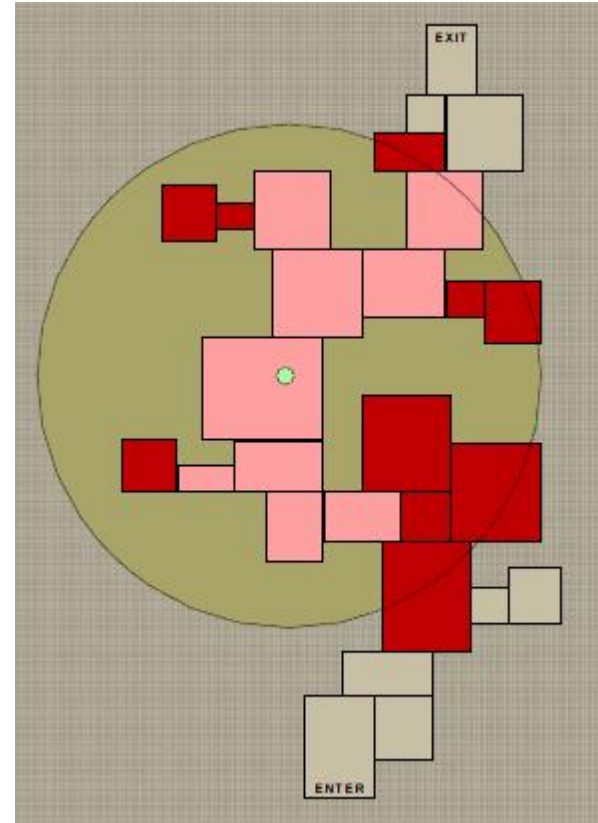
プレイヤー経路を予測し、  
敵の特性に応じて、生成・配置する  
(Spawning & Distribution)



プレイヤーのスキル、メンバー、進行に応じて  
敵の出現数、出現ポイント、タイミングが変化する



ログイン、パーティの度に違った面白さ  
＝何度でもプレイしたくなるゲームをメタAIで実現

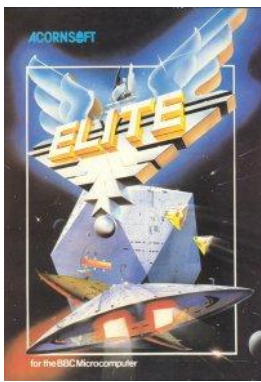


円 = AAS  
ピンク = プレイヤー可視領域  
赤 = AAS かつ不可視領域  
(例えば、赤の領域に投げる武器の  
ある敵を配置するなど)

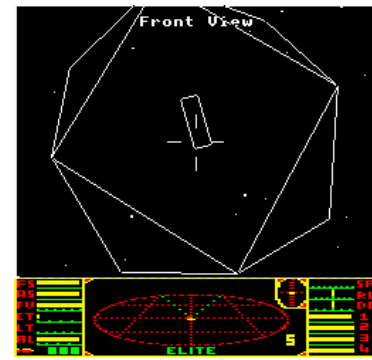
Michael Booth, "The AI Systems of Left 4 Dead,"  
Artificial Intelligence and Interactive Digital Entertainment Conference at Stanford  
<http://www.valvesoftware.com/publications.html>

# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims



# Elite (ArconSoft,1984)



- BBC Mirco (2Mhz 6502, 32K RAM, 22K for game)用ゲーム
- 初の家庭用3Dゲーム(?)
- ワイアフレーム3D, プロシージャル技術
- Cambridge 大学のDavid Braben(1964-、現Frontier Soft創設者、代表), Ian Bell(1962-) の2人が個人的に製作(10代)
- 1983年に完成。パブリッシャー ArconSoft と契約
- 100万本の大ヒット
- Commodore 64, Spetrum, Amstrad, AppleII, など 17 のプラットフォームに移植
- プロシージャル技術を導入したゲームの祖としての位置づけ



# BBC Micro



- Acorn Computer
- 1981-1986 イギリスでホビー用パソコン
- 学校教材として成功を収める
- 2Mhz MOS 6502 (32K RAM )
- BBC BASIC (アセンブラはあまり使わなかったとのこと)
- BASIC インタプリタが優秀
- アメリカに参入失敗
- 文字数 最大  $80 \times 32$  解像度 (最大)  $640 \times 256$
- 色数 最大 8

[http://ja.wikipedia.org/wiki/BBC\\_Micro](http://ja.wikipedia.org/wiki/BBC_Micro)

# Elite の技術

## 星系の自動生成

- それぞれの銀河を自動生成(Procedural Generation)
  - 各惑星のパラメーターは多様かつ決定されている
    - フィボナッチ数列を使う (くり返し使えて、かつ数列としてある程度ランダム)
- 12 34 46 80 126 06 32 38 70 108 78 86 164 150 (2桁のみ使用)
- 各惑星の位置、名前、政府、経済... に使用 (ステージとなる宇宙を自動生成)



The History of Elite: Space, the Endless Frontier (Gamasutra)

[http://www.gamasutra.com/view/feature/3983/the\\_history\\_of\\_elite\\_space\\_the\\_.php](http://www.gamasutra.com/view/feature/3983/the_history_of_elite_space_the_.php)

# Elite の技術

## 3D演算

- 8bit Accumulator と8bit インデックス・レジスタ
- 対数計算は他の演算の10倍の速度(2つの256byteテーブル)
- 回転は一定角度で行列演算を足し算に変換して行う

## モデリング

- スペースシップと認識されるには20ライン強が必要
- 見えない頂点の計算は省く(今で言うカリング)
- 対称性のある図形  
(行列演算を使い回すことが出来る。20%速度アップ)



## レンダリング

- クリアスクリーンが遅い。ダブルバッファ分のメモリもない。
- 描いた線を打ち消すために(ピクセル反転して)描く。
- 角度ごとに違った描線アルゴリズム(今で言うエイリアシング)



## テキスト

- レターペアをトークンとして保持して、惑星の名前を形成

# プロシージャルの歴史: CG

1980

1990

2000

フラクタル  
幾何学

1975



フラクタル・イメージ  
SIGGRAPH 1987

フラクタル幾何学を基礎とした  
プロシージャルなモデル生成

地形自動生成

植物自動生成

雲自動生成

3Dモデル生成

テクスチャ自動生成

2Dモデル生成

Terragen

SpeedTree

natFX MAX

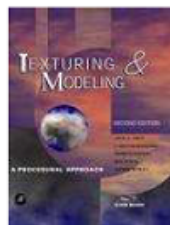
パリン・ノイズ

1997, 2002

Pro FX



1988



1994



2003



# プロシージャルの歴史: ダンジョン自動生成

1980

Rogue



1980

Rogue Clone  
ローグライクゲーム

NetHack



1987

ティル・ナ・ノーグ

システムソフト



1987~

不思議のダンジョン

チュンソフト



1993~

Diablo

Blizzard



1996

Hellgate: London

flagshipstudio



2006

ブルードラゴン

Microsoft



2007

CODED ARMS

KONAMI



2005

TOEO

namco



2006

マビノギ

nexon



2004

FF11

Square Enix



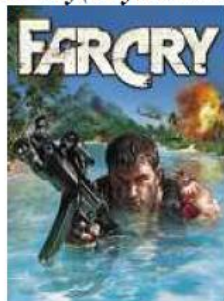
2007

MMOにおける応用



# プロシージャルなゲームエンジン

*Far Cry(Crytek & Ubisoft)*



CryEngine 1.0



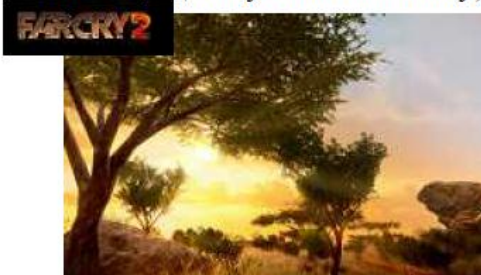
*Far Cry Instincts*

*Crysis(Crytek & EA)*



CryEngine 2.0

*Far Cry 2(Ubisoft Montreal & Ubisoft)*



Dunia Engine



Frostbite Engine

次世代Battlefield用エンジン

Practical Example: Mountains Generation and Realistic Snow Accumulation



次世代Battlefieldシリーズ用エンジンFrostbite (4gamers)

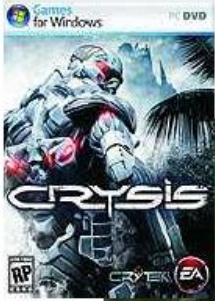
<http://www.4gamer.net/news/history/2007.03/20070308211803detail.html>

# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims

# Crysis における植物アニメーション自動生成

## Procedural Vertex Animation



[http://www.crytek.com/fileadmin/user\\_upload/cryengine2/CryENGINE2Features.pdf](http://www.crytek.com/fileadmin/user_upload/cryengine2/CryENGINE2Features.pdf)

[Tiago Sousa, "Chapter 16: Vegetation Procedural Animation and Shading in Crysis", GPU Gems 3, 2007](#)

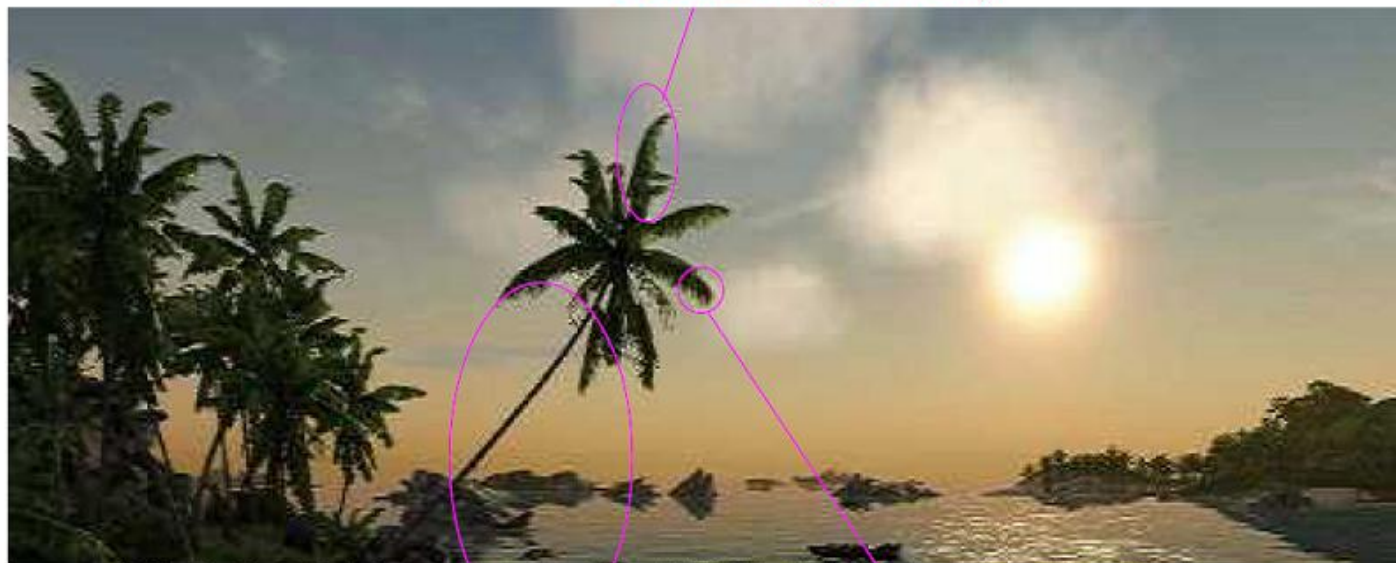
**Finding Next Gen CryEngine2,**

[http://ati.amd.com/developer/gdc/2007/Mittring-Finding\\_NextGen\\_CryEngine2\(Siggraph07\).pdf](http://ati.amd.com/developer/gdc/2007/Mittring-Finding_NextGen_CryEngine2(Siggraph07).pdf)



# 風に揺れる森 in Crysis

葉全体(leaves)

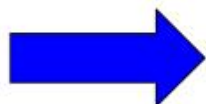


Finding Next Gen CryEngine2, [http://ati.amd.com/developer/gdc/2007/Mittring-Finding\\_NextGen\\_CryEngine2\(Siggraph07\).pdf](http://ati.amd.com/developer/gdc/2007/Mittring-Finding_NextGen_CryEngine2(Siggraph07).pdf)

全体(entire)

葉(leaf)

風と葉の硬さからモデルの曲がり具合を計算する



風



×

各エッジ(辺)  
に埋め込まれる  
硬さ情報を  
埋め込む

[www.two-kings.de/.../dxgraphics10.html](http://www.two-kings.de/.../dxgraphics10.html)



葉全体の硬さ

一枚一枚としての硬さ

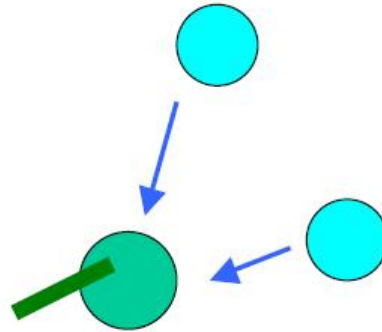
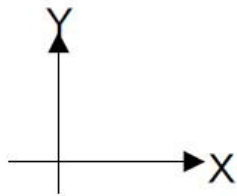
上の二つを足した総合のかたさ

$\alpha$

Precomputed ambient occlusion

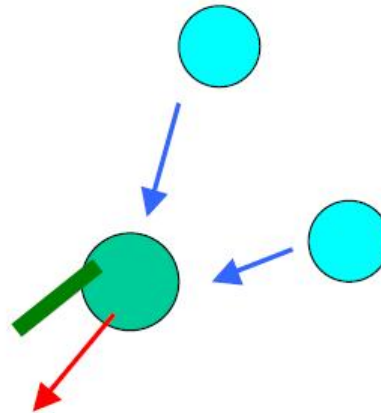
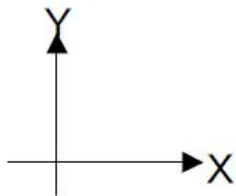
# 風に揺れる森 in Crysis

① その樹木に対する幾つかの風源の効果を足し合わせる



[http://www.geimeris.com/wp-content/images/crysis\\_later.jpg](http://www.geimeris.com/wp-content/images/crysis_later.jpg)

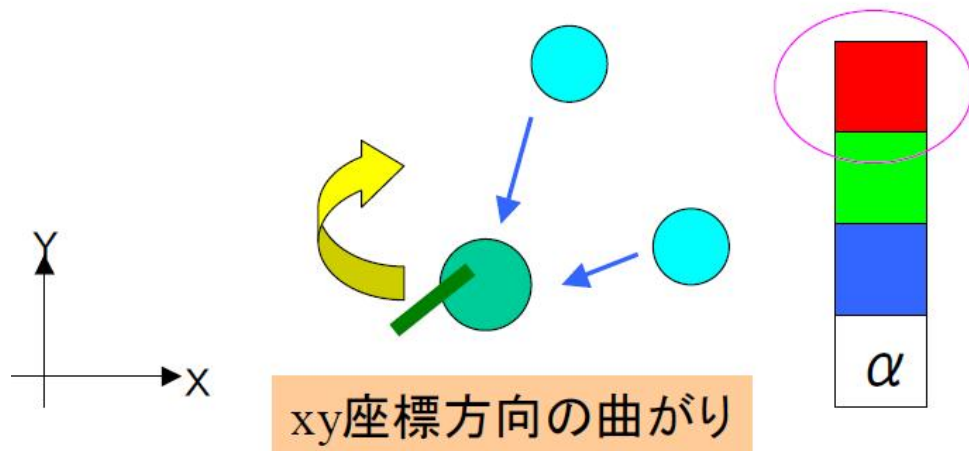
② 世界座標(xy)上の風ベクトルから、全体の曲げ方を決める。





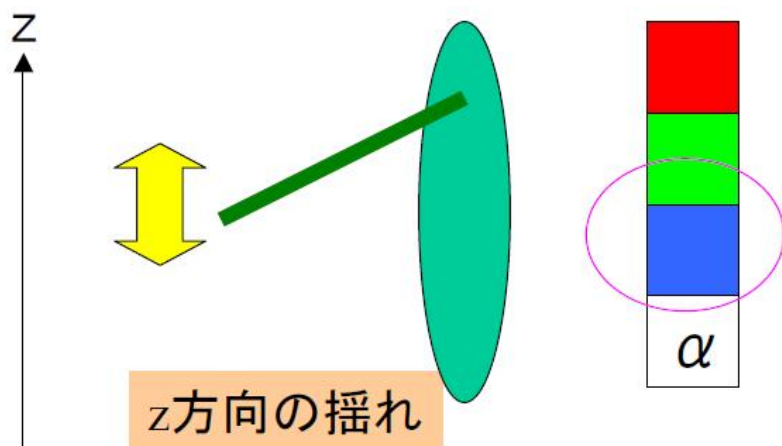
# 風に揺れる森 in Crysis

- ③ 葉のまとまりに対する曲がりを、風の強さだけから計算する



<http://www.crytek.com/technology/cryengine-2/specifications/>

- ④ 一枚の葉の揺れ(z-方向)を、風の強さだけから計算する





# デモ



**Procedural Vegetation  
Animation and Shading**



## Procedural Vegetation Animation in Crysis

tiago\_gpuGems3\_1280\_mpeg



# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine - Frostbite Engine
- (6) 日常系AI
  - The Sims

# FarCry2 (2008)におけるプロシージャル技術



50km四方のマップを作る

オブジェクト(草木) & アニメーションデータを  
自動生成

植物に関してプロシージャルのアルゴリズムの助けを借りてデザイナーが生成  
(まとまった単位土地内の**分布**、**成長率**などを指定して生成)

アニメーションや破壊のされ方は、簡単な設定をしておけば、ゲーム内で環境に  
応じてシミュレートされる。 (例) 草原で炎が燃え広がる



# FarCry2 (Dunia Engine) デモ

<http://www.farcry2-hq.com/downloads,18,dunia-engine-nr1.htm>



草原自動生成



時間システム



樹木自動生成



動的天候システム



動的天候システム



# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine -Frostbite Engine
- (6) 日常系AI
  - The Sims



## Frostbite Engine

- ① オブジェクト生成  
(We can generate objects with procedural techniques  
-Then use rules to deform / destroy / modify / move them  
-Better interactivity)
- ② Semi-procedural surface shader
- ③ Procedural shader
- ④ Procedurally distributed on the fly

Practical Example: Mountains Generation and Realistic Snow Accumulation



GDC 2007 Frostbite “Rendering Architecture and Real-time Procedural Shading & Texturing Techniques”

[http://developer.amd.com/assets/Andersson-Tatarchuk-FrostbiteRenderingArchitecture\(GDC07\\_AMD\\_Session\).pdf](http://developer.amd.com/assets/Andersson-Tatarchuk-FrostbiteRenderingArchitecture(GDC07_AMD_Session).pdf)

GDC 2007 “The Importance of Being Noisy: Fast, High Quality Noise”, N. Tatarchuk

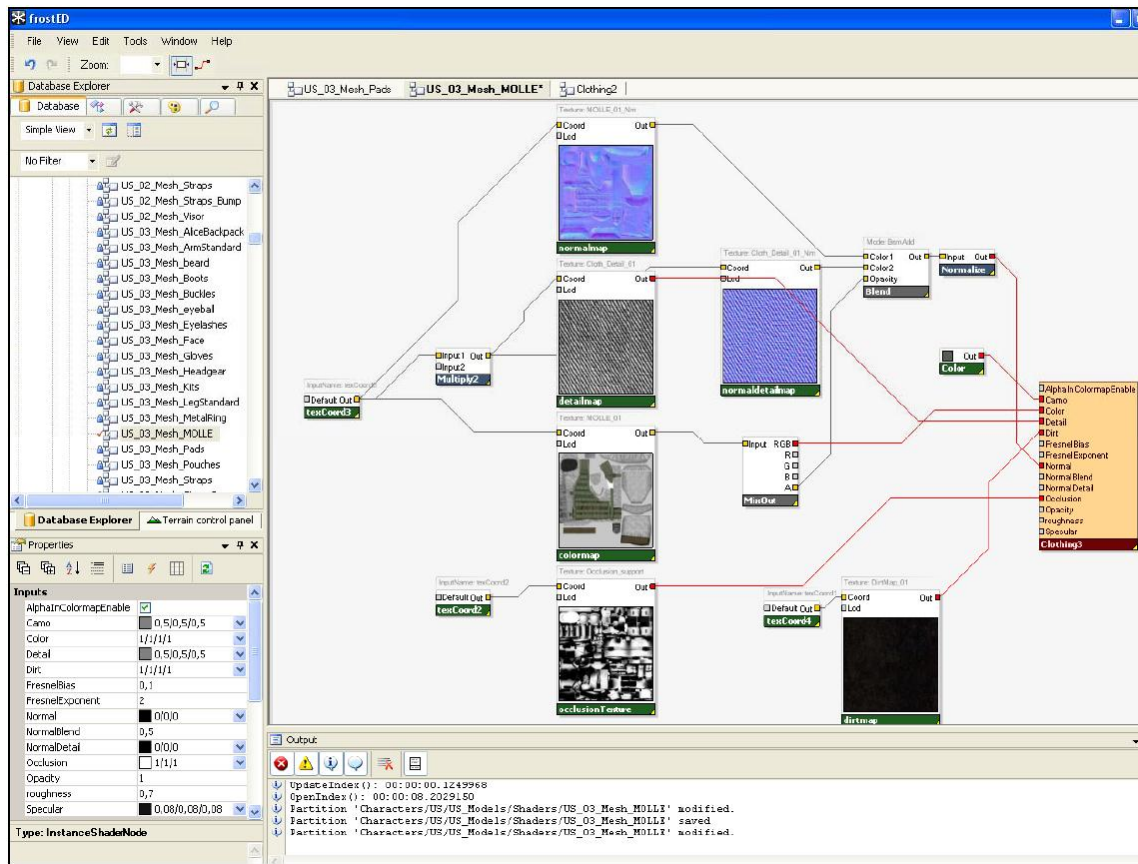
[http://developer.amd.com/Assets/Tatarchuk-Noise\(GDC07-D3D\\_Day\).pdf](http://developer.amd.com/Assets/Tatarchuk-Noise(GDC07-D3D_Day).pdf)

SIGGRAPH 2007 Johan Andersson “Terrain Rendering in Frostbite using Procedural Shader Splatting”

[http://ati.amd.com/developer/gdc/2007/Andersson-TerrainRendering\(Siggraph07\).pdf](http://ati.amd.com/developer/gdc/2007/Andersson-TerrainRendering(Siggraph07).pdf)

# Semi-procedural Surface Shader

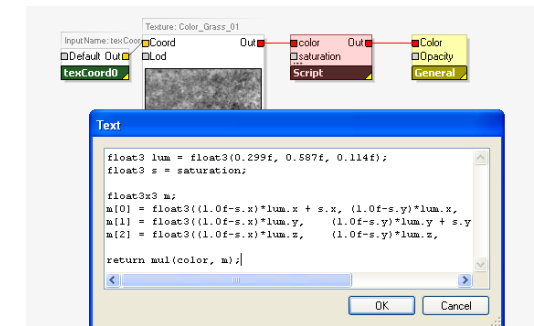
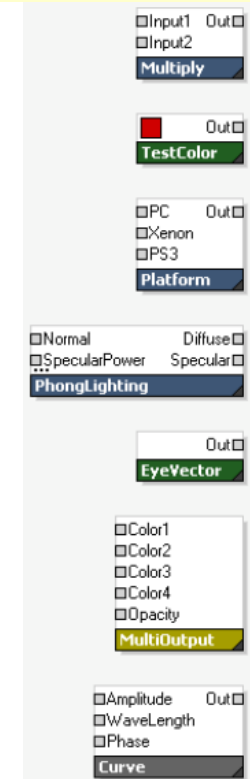
表面状態などに応じてシェーダーが形成される(グラフベースでアーティストも作成可能)



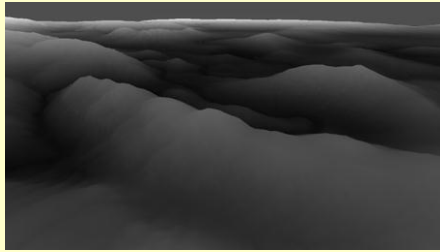
## 組み込み可能ノード

- 基本演算 (mul, add, div)
- 地形条件 (Fresnel, refraction)
- ロジカル (プラットフォームなど)
- パラメータ (スカラ、ベクトル、ブール)

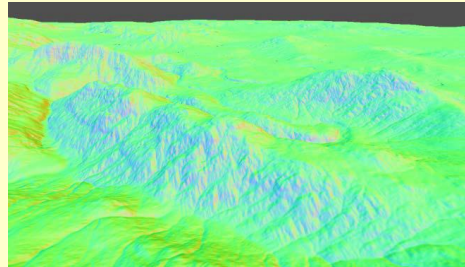
- パラメータ (Z, Normal, eye-vector)
- ライトニング (phong, sub-surface)
- ルート (general, offset, multi-output)
- その他 (script, curve)



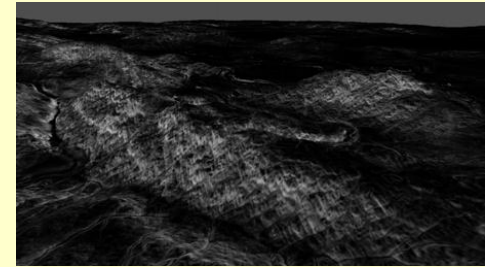
# プロシージャルに生成するシェーダー用のマップ・マスク



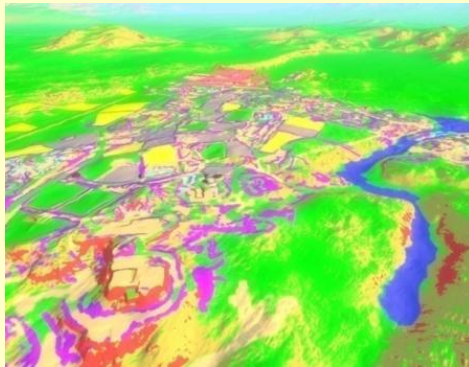
高さマップ



ノーマルマップ



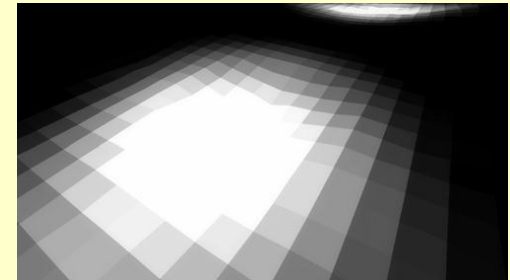
傾斜マップ



マテリアル・マスク



デストラクション・マスク

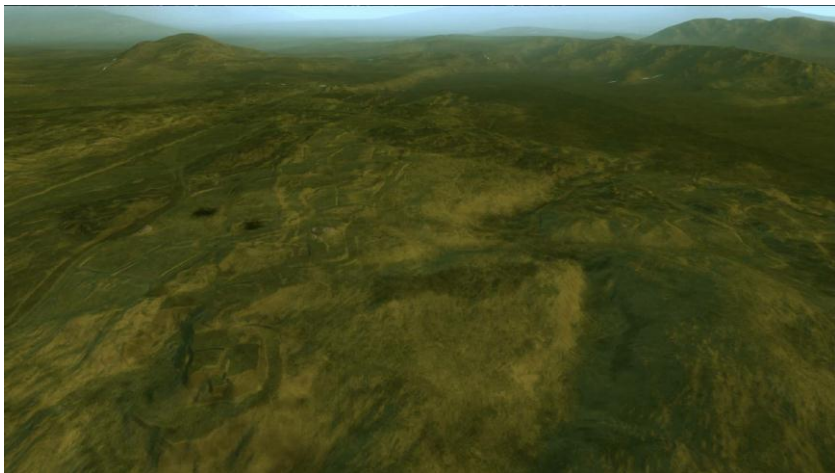


クレーター・マスク

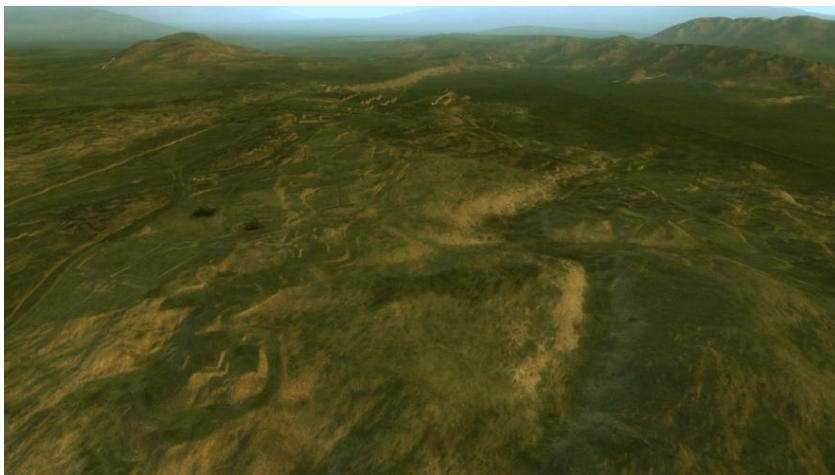




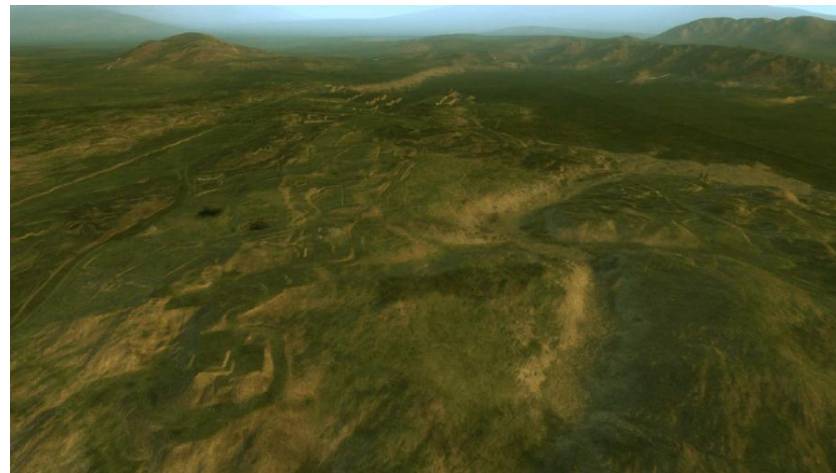
# Semi-procedural Surface Shader 実例



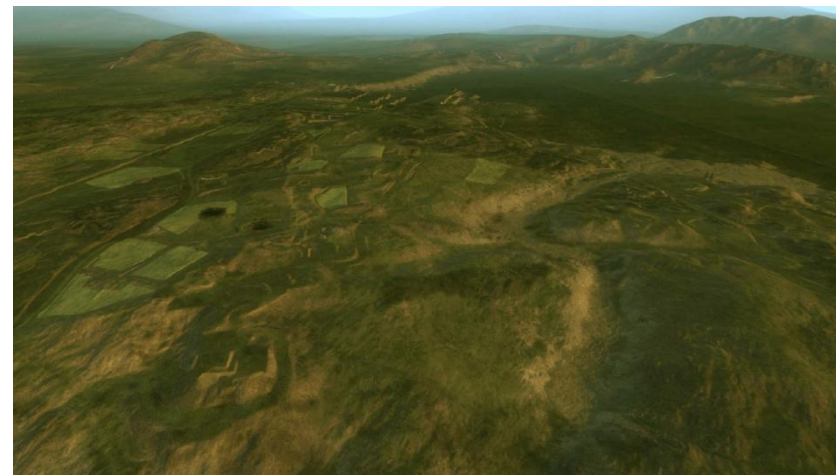
グラス(草)マテリアルのみ



+スロープにおける泥



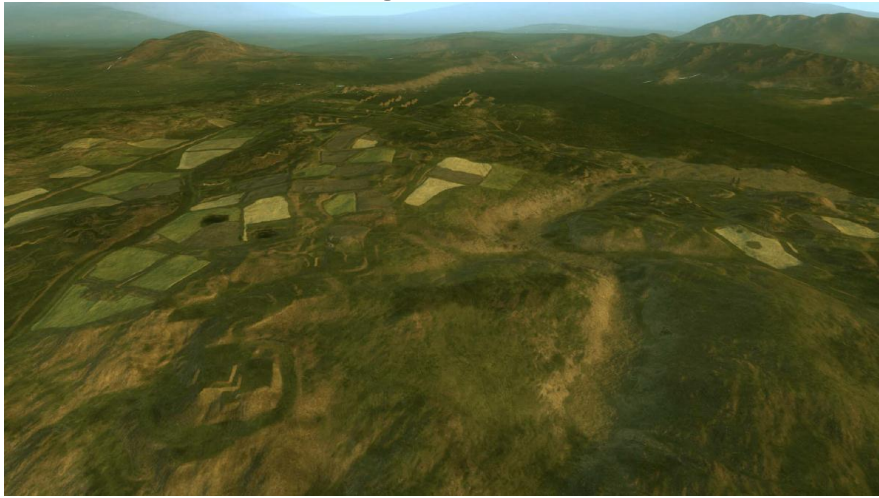
+海面/川面マテリアル



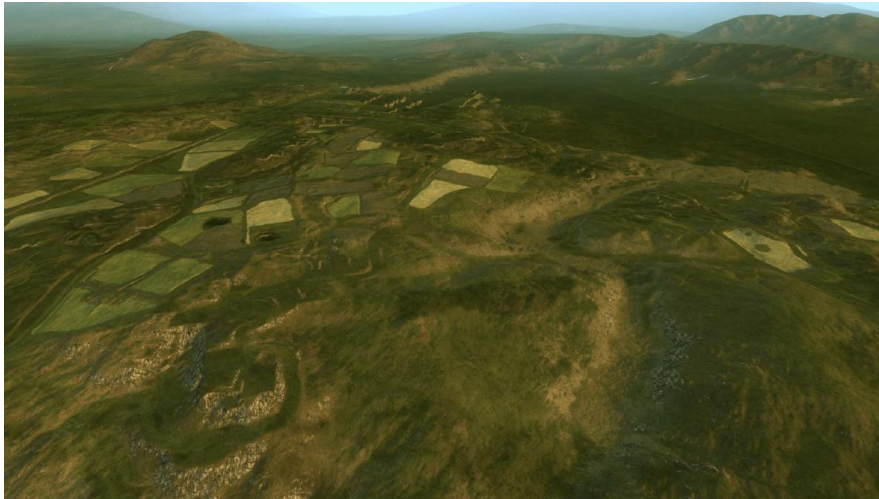
+草原マテリアル



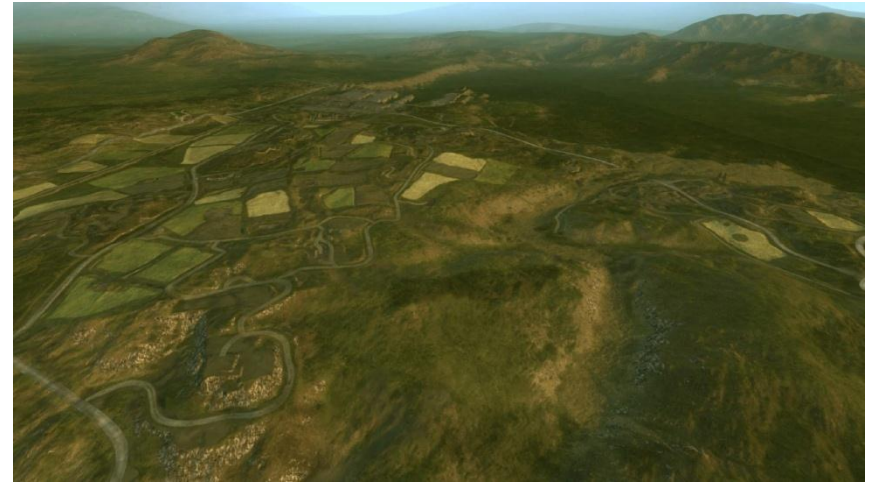
# Semi-procedural Surface Shader 実例



+2つの別タイプ草原マテリアル



+スロープ検出→崖テクスチャ



+道マテリアルにデカールを貼る

# やぶ自動分布&生成



Terrain Rendering in Frostbite Using Procedural Shader Splatting, Johan Andersson

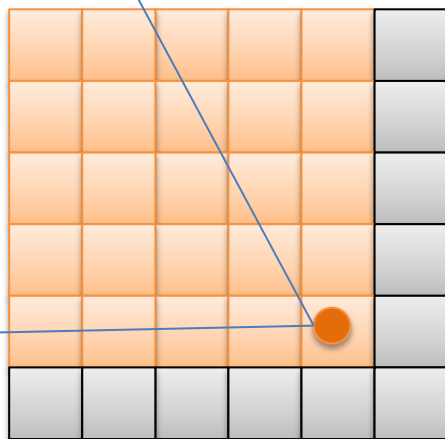
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=E186596C09CC8E696BD79A02DBA080CC?doi=10.1.1.161.8979&rep=rep1&type=pdf>



# やぶ自動分布&生成

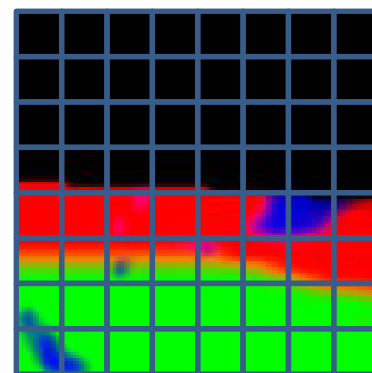
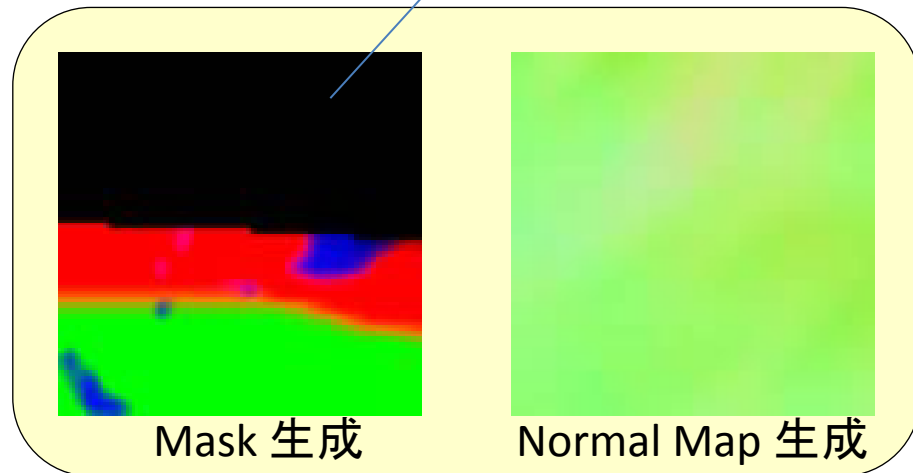
黒=やぶ生成なし

16x16m Cells



ユーザーの周囲の 16x16m Cells 群を  
アクティブにして、プロシージャルに  
やぶを生成する。  
(このCell群自体はあらかじめプールされている)

各Cellは、もともと、やぶのタイプを持っている。  
各Cellはその頂点のVertex Bufferをインスタンスに  
持つ。  
(例えば、破壊された地形の頂点データは変化する)



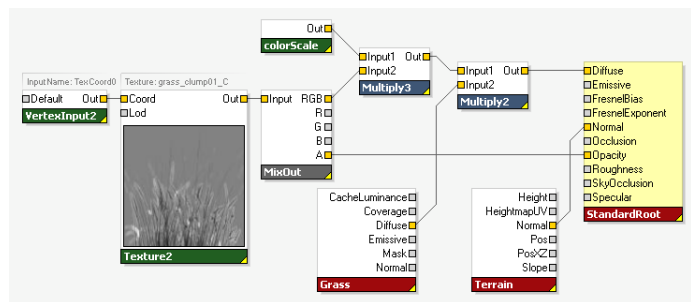
再グリッド化

(グリッドの大きさはその場の  
マテリアルの密度に依存)



やぶ生成

(各グリッドの値と乱数で、やぶを置くか  
決定。Normal Map はやぶモデルの方  
向を決定する。乱数シードはネットワー  
クで共有するためマルチでも同じやぶが見  
える)



やぶシェーダー

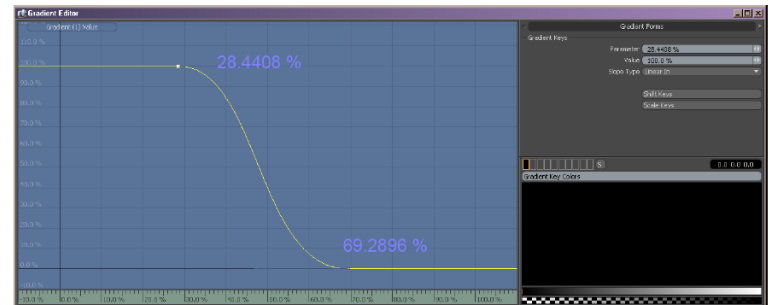
# 地形自動生成 & 自動雪シェーダー



- シェーダーでマルチ分解能の Displacement map で山脈生成
- 乱数は fBM (fractal Brownian Motion)

## Snow Coverage

- ④ Snow coverage determined procedurally
- ④ Artists paint slope's control points as vertex colors
- ④ Slope used for smoothstep to define snow masks
  - ④ Mask 1: smoothstep based on geometric normal's y component
  - ④ Mask 2: smoothstep based on normal map's y component
  - ④ Final snow coverage mask = Mask1 \* Mask2

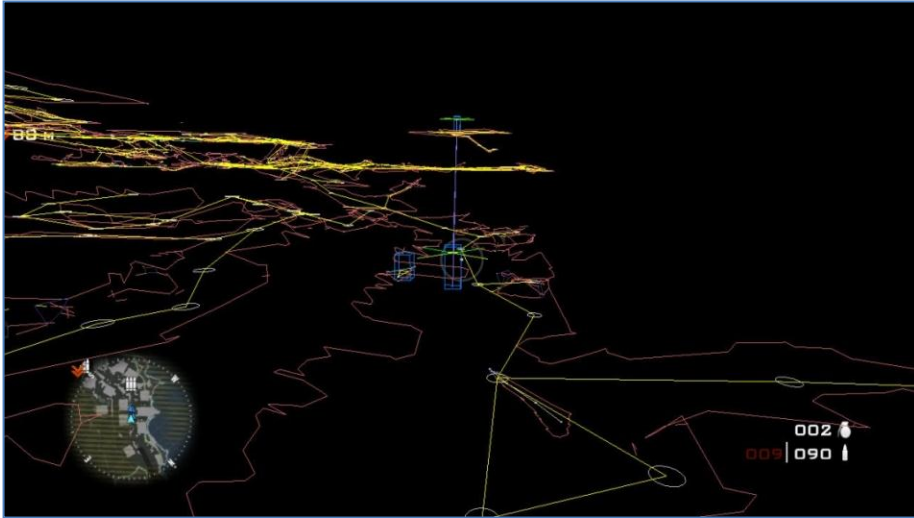


- 地形マップのノーマルマップ
- バンプマップのノーマルマップから Snow Coverage を計算

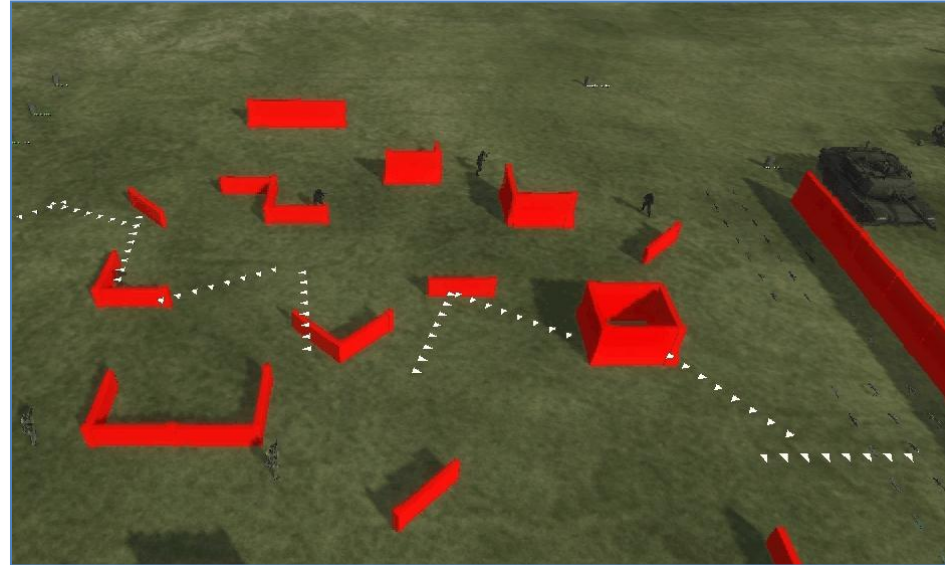
The Importance of Being Noisy: Fast, High Quality Noise, Natalya Tatarchuk  
[http://developer.amd.com/assets/Tatarchuk-Noise\(GDC07-D3D\\_Day\).pdf](http://developer.amd.com/assets/Tatarchuk-Noise(GDC07-D3D_Day).pdf)



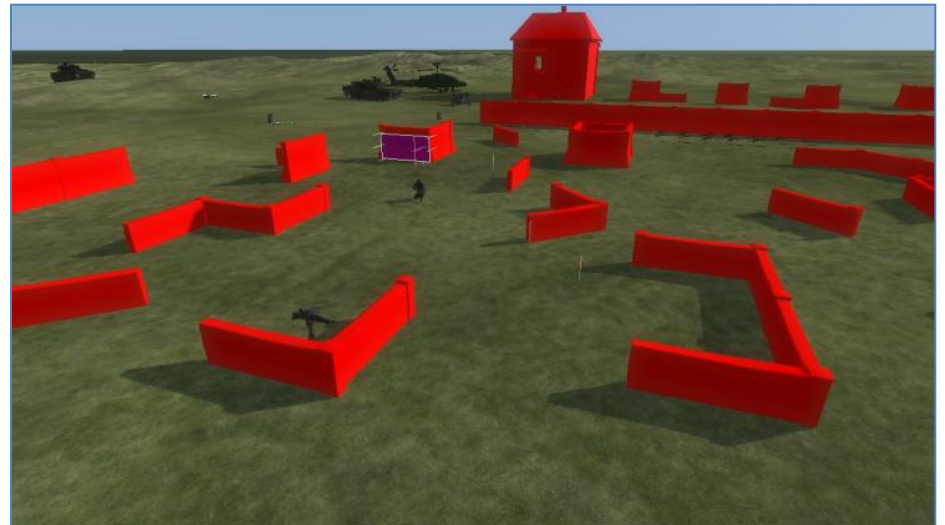
# FrostBite AI (Battlefield: BC2)



AIがグローバルマップを認識するための世界表現



敵、味方のメンバーからの影響度を計算して、  
前線(Frontline, 勢力の分かれ目を検出)



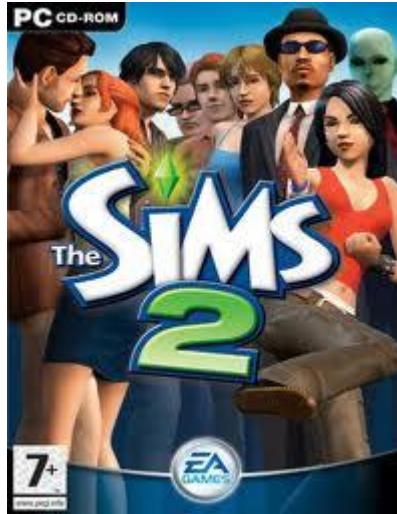
パス検索と物理テストを組み合わせで隠れること  
ができるポイントを検出



# 目次

- (1) エージェント・アーキテクチャ
  - HALO1,2 - F.E.A.R.
- (2) パス検索 & アニメーション
  - CounterStrike のパス検索
  - StarCraft 2 のパス検索
  - Dragon Age のパス検索 (階層化モデルとファンネルアルゴリズム)
  - Havok AI
  - Guild War 2(Havok AI)
  - Autodesk® Kynapse
  - DOOM, Quake シリーズの BSP(Binary Space Partitoning)システム
  - Insomniac 社 Navigation Engine
  - Crysis 2 のパラメトリック・アニメーション・システム
  - Killzone 2
- (3) 生体情報フィードバック(Valve研究事例)
  - Left 4 Dead - Alien Swarm - Portal
- (4) メタAI (Left 4 Dead)
- (5) プロシージャル技術
  - Elite - CryEngine - Dunia Engine -Frostbite Engine
- (6) 日常系AI
  - The Sims

# The Sims AI



内容: 日常箱庭系、人間の社会シミュレーションゲーム

開発元: Maxis

出版: EA

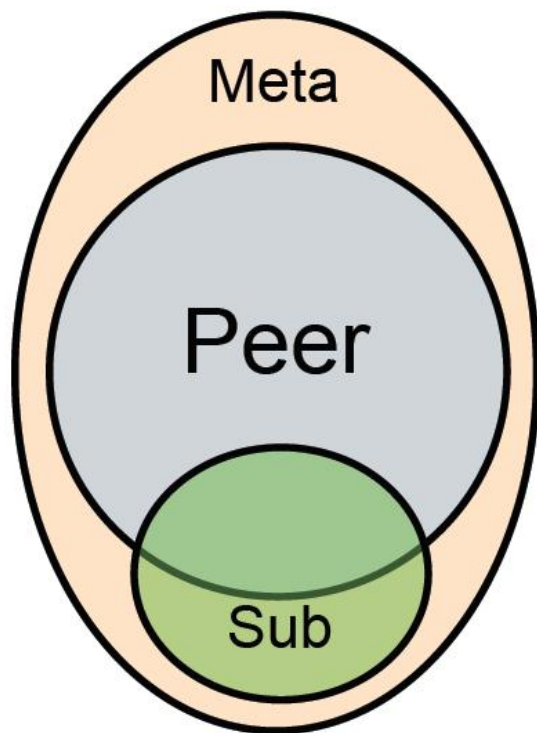
Hardware: PC など

出版年: 2000年～ (2004年に発売された The Sims 2 は 1300万本)

10周年を迎えた The Sims を振り返る (4gamers)

<http://www.4gamer.net/games/036/G003691/20100205014/>

# The Sims シリーズのAIの作り方



Meta 

Peer 

Sub 

**【原則】** 周囲の対象に対する、あらゆる可能な行動から、**Happiness** 係数を最大化する行動を選択する。

Sims (not under direct player control) choose what to do by selecting, from all of the possible behaviors in all of the objects, the behavior that maximizes their current happiness.

人をダイナミクス(力学系、動的な数値の仕組み)として動かす。  
世界を動かす PeerAI(=キャラクターAI) を構築。



# オブジェクトに仕込むデータ構造

**Data (Class, State)**

**Graphics (sprites, z-  
Animations (skeletal)**

**Sound Effects**

**Code (Script)**

- Main (object thread)
- External 1
- External 2
- External 3

パラメーター

グラフィックス  
アニメーション

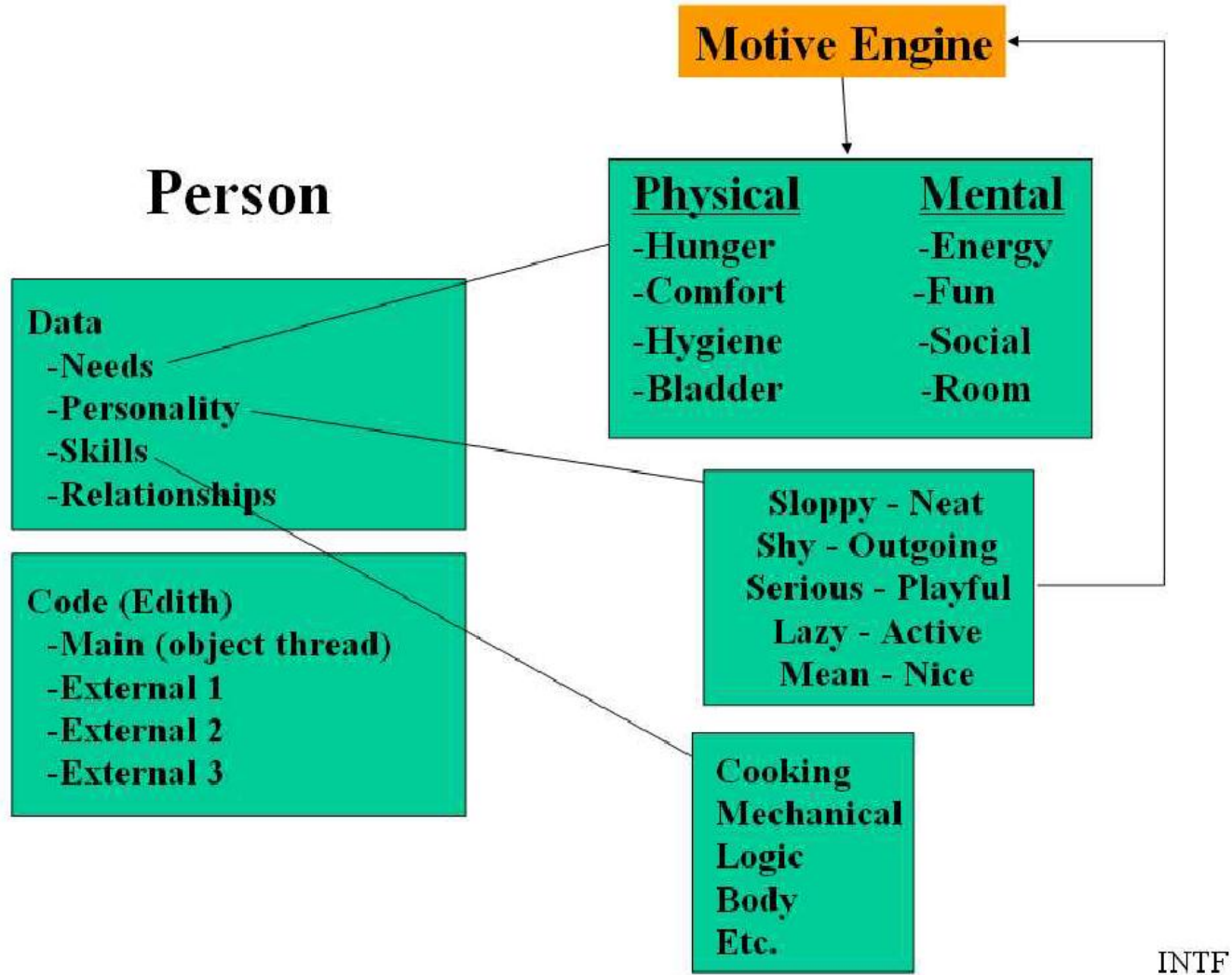
サウンド

メインスレッド

いろいろなインタラクションの仕方



# NPCに仕込むデータ構造



INTF



# 最適な行動を選択する



Hunger +20  
Comfort -12  
Hygiene -30  
**Bladder -75**  
Energy +80  
Fun +40  
Social +10  
Room -60

Toilet



Mood +26

-Urinate (+40 Bladder)  
-Clean (+30 Room)  
-Unclog (+40 Room)

Bathtub



Mood +20

-Take Bath(+40 Hygiene)  
(+30 Comfort)  
-Clean (+20 Room)

Mood +18

[原則] 周囲の対象に対する、あらゆる可能な行動から、  
**Happiness** (ここでは**Mood**) 係数を最大化する行動を選択する。



# Auto-Satisfy Curves

Menu Add Motive Remove Motive Copy Motive

| Motive              | Intensity | InitialMin | InitialMax | HasDef... | Trigger... | TriggerValu... | AddBuff       | RemoveBuff           | CustomClass | Trigger... | TriggerValu... | AddBuff    |
|---------------------|-----------|------------|------------|-----------|------------|----------------|---------------|----------------------|-------------|------------|----------------|------------|
| Bladder             | 4000      | 75         | 100        | True      | -100       | -91            | ReallyHasT... |                      |             | -90        | -61            | HasToPee   |
| Bladder age: Tod... | 4000      | 75         | 100        | True      | -100       | -91            | ReallyHasT... |                      |             | -90        | -61            | HasToPee   |
| Bladder age: Baby   | 4000      | 95         | 100        | True      | -100       | -91            | ReallyHasT... |                      |             | -90        | -61            | HasToPee   |
| Bladder age: Elder  | 4000      | 90         | 100        | True      | -100       | -91            | ReallyHasT... |                      |             | -90        | -61            | HasToPee   |
| Hunger              | 10000     | 0          | 0          | False     | -100       | -81            | Starving      |                      |             | -80        | -61            | VeryHungry |
| Hunger age: Tod...  | 10000     | 0          | 0          | False     | -100       | -81            | Starving      |                      |             | -80        | -61            | VeryHungry |
| Hunger age: Baby    | 10000     | 0          | 0          | False     | -100       | -81            | Starving      |                      |             | -80        | -61            | VeryHungry |
| Energy              | 4000      | 83         | 83         | False     | -100       | -100           | Exhausted     |                      |             | -99        | -80            | Tired      |
| Energy age: Baby    | 4000      | 83         | 83         | False     | -100       | -100           | Exhausted     |                      |             | -99        | -80            | Tired      |
| Energy age: Tod...  | 4000      | 83         | 83         | False     | -100       | -100           | Exhausted     |                      |             | -99        | -80            | Tired      |
| Fun                 | 1250      | 0          | 0          | True      | 0          | 100            |               | Overworked           |             | -100       | -51            | Stressed   |
| Hygiene             | 700       | 80         | 95         | True      | -100       | -81            | Smelly        | SqueakyClean, Grungy |             | -80        | -61            | Grungy     |
| Hygiene age: Baby   | 5000      | 80         | 95         | True      | -100       | -81            | Smelly        | SqueakyClean, Grungy |             | -80        | -61            | Grungy     |
| Hygiene age: To...  | 5000      | 80         | 95         | True      | -100       | -81            | Smelly        | SqueakyClean, Grungy |             | -80        | -61            | Grungy     |
| Work                | 1000      | 0          | 0          | True      |            |                |               |                      |             |            |                |            |
| GoHome              | 0         | 0          | 0          | False     |            |                |               |                      |             |            |                |            |
| Social              | 400       | 95         | 100        | True      | -100       | -99            | Desolate      | Lonely               |             | -98        | -61            | Lonely     |
| Social age: Baby    | 2000      | 95         | 100        | True      | -100       | -89            | Desolate      | Lonely               |             | -88        | -61            | Lonely     |
| Social age: Toddler | 2000      | 95         | 100        | True      | -100       | -89            | Desolate      | Lonely               |             | -88        | -61            | Lonely     |

☒ Universal ☐ Insatiable

Initial Value

☒ From Auto-Satisfy Curve

☐ Fixed

Time Randomness: 0.5

Decay

☐ No Decay

☐ From Auto-Satisfy Curve

☐ Decay to Zero

☐ Fixed Decay

☒ Variable Decay By Moodlet

Configure Moodlet Decay

☐ Show Decay

Specificity

Age:

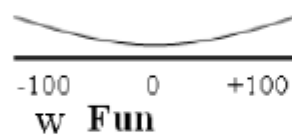
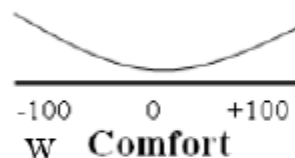
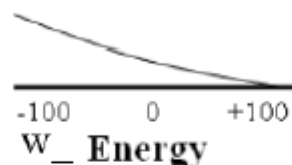
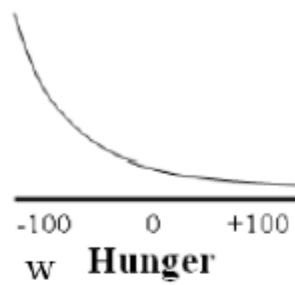
Trait:

Hunger Auto Satisfy

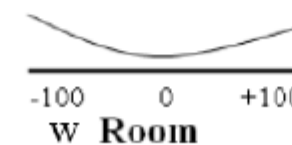
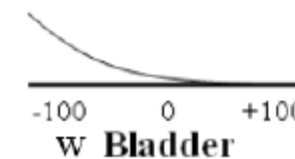
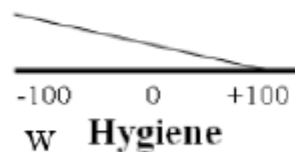
Hunger Desire

Hunger Mood Contribution

# Moodの計算のためのウェイト

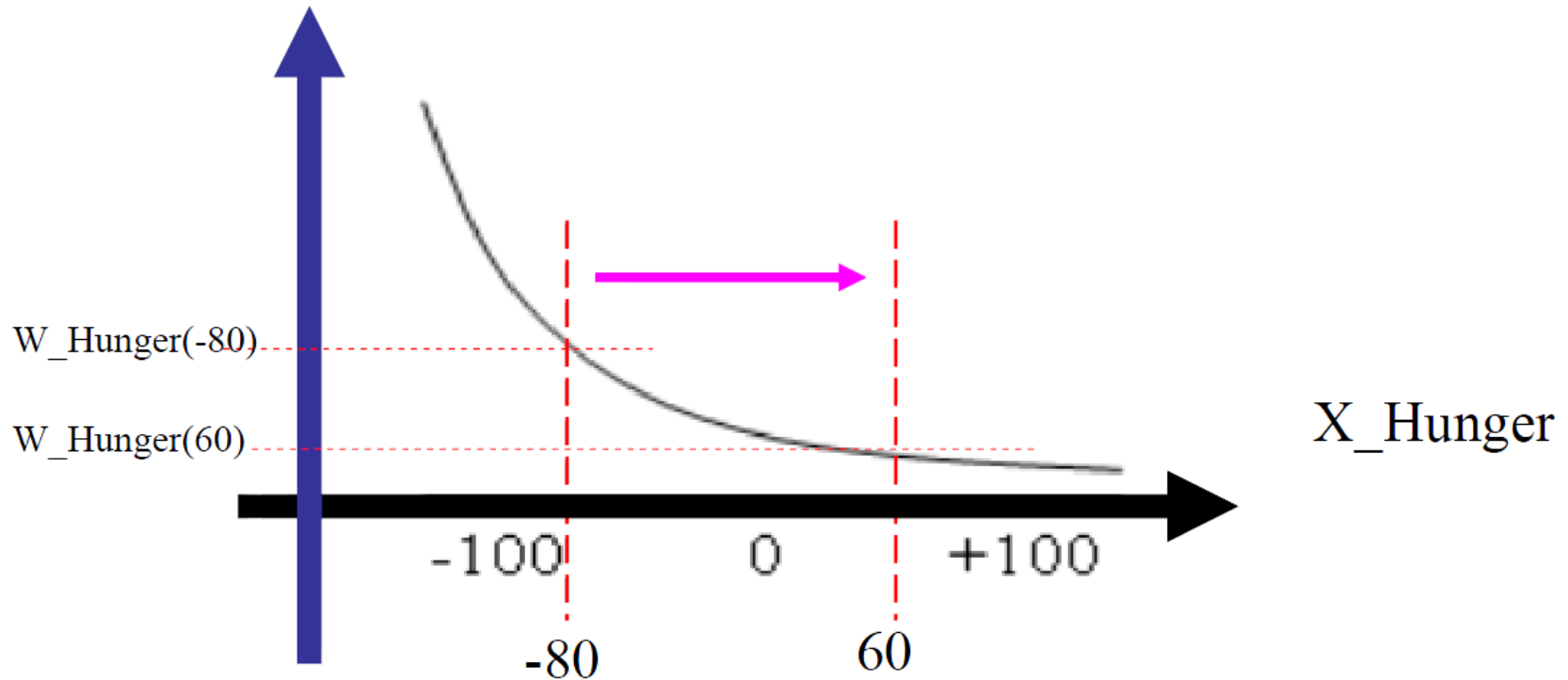


**=** Mood (-100..+100)



$$\text{Mood} = w\_Hunger * Hunger + w\_Energy * Energy + \dots$$

# 計算の仕方



$W\_Hunger$ 関数

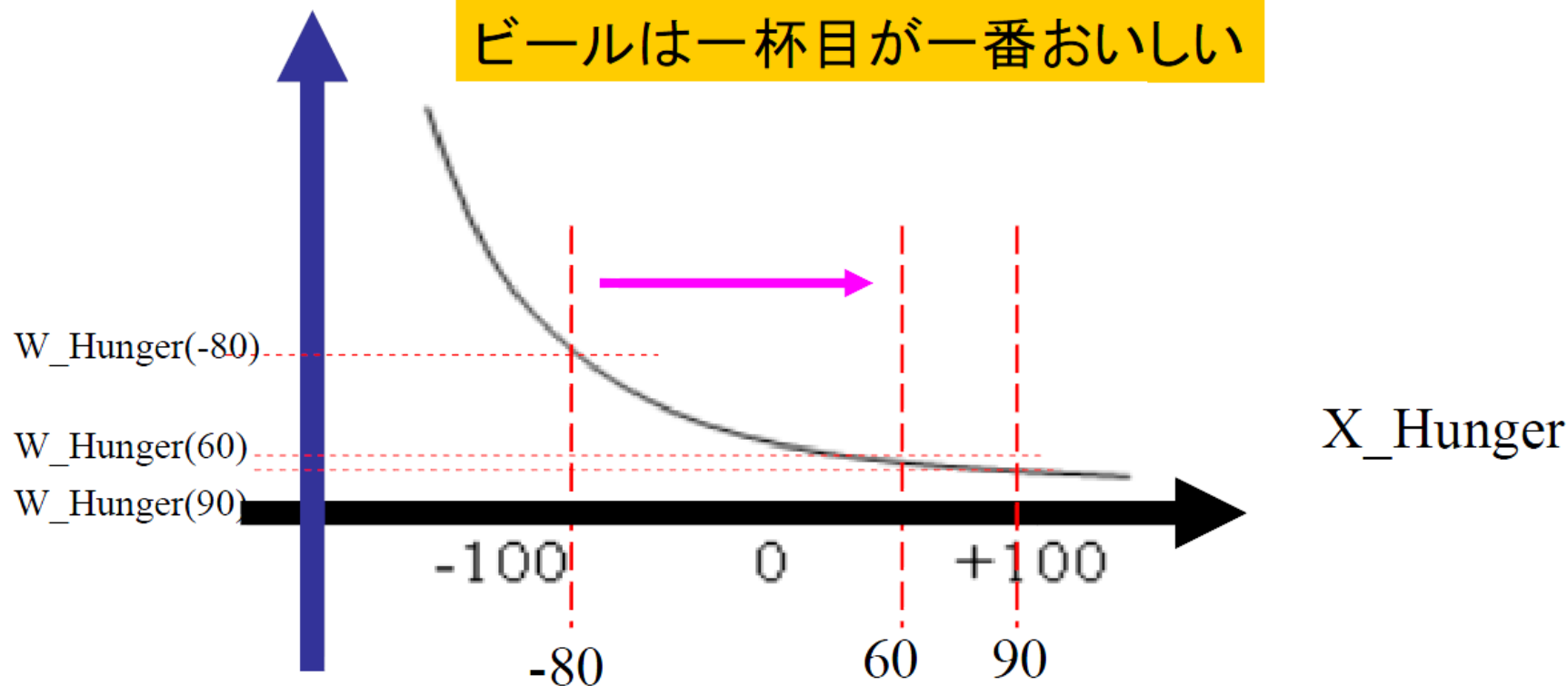
-80の時の食事欲求 =  $W\_Hunger(-80) * (-80)$

60の時の食事欲求 =  $W\_Hunger(60) * (60)$

$\Delta$  (変化分) =  $W\_Hunger(60) * (60) - W\_Hunger(-80) * (-80)$

# 限界効用逓減の法則

ビールは一杯目が一番おいしい



$$\Delta (-80 \rightarrow 60) = W\_Hunger(60) * (60) - W\_Hunger(-80) * (-80)$$

$$\Delta (60 \rightarrow 90) = W\_Hunger(90) * (80) - W\_Hunger(90) * (60)$$

$\Delta (-80 \rightarrow 60)$  の方が  $\Delta (60 \rightarrow 90)$  より圧倒的に大きい

ある程度満たされたものを満たすより、  
満たされないものをある程度満たす方が大きな満足をもたらす



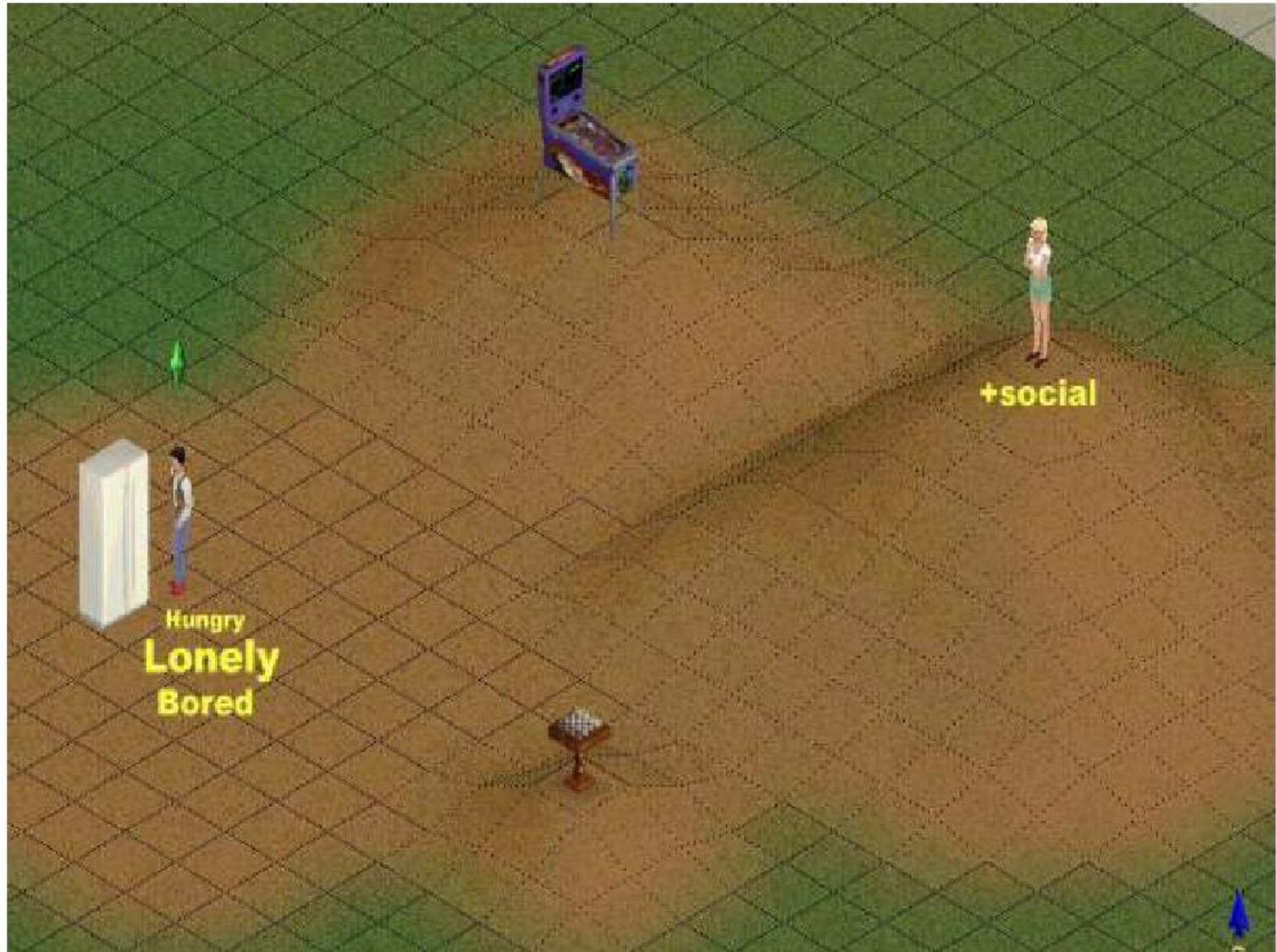
# Mood を最大化



冷蔵庫が最も総合的にMoodを上昇させるから



# Mood を最大化



冷蔵庫へ行きます。

# Mood を最大化



お腹が膨れたので、ちょっと退屈だから、女の子と話します。





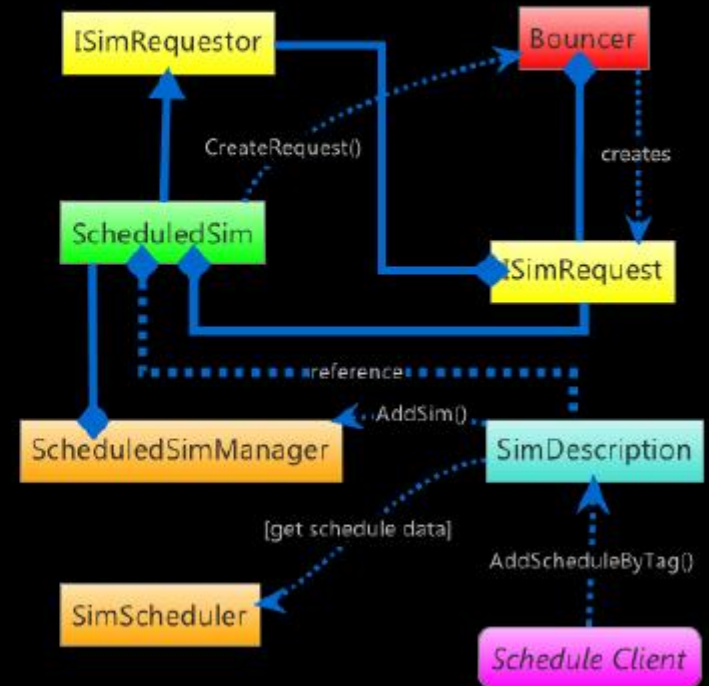
10周年を迎えた The Sims を振り返る (4gamers)

<http://www.4gamer.net/games/036/G003691/20100205014/>

# The Sims: Medieval

## Implementation: Schedules

1. Schedule client calls `AddScheduleByTag()` on `SimDescription`
2. `SimDescription` caches the schedule data from the `SimScheduler` singleton
3. `SimDescription` calls `AddSim()` on `ScheduledSimManager` singleton
4. `ScheduledSimManager` creates a `ScheduledSim` object
5. Each tick, `ScheduledSimManager` iterates over all `ScheduledSim`'s and submits low-priority bouncer requests for any who are on a schedule





# The Sims: Medieval

## Commodities Autonomy Window

The screenshot displays the 'Autonomy Window' for a Sim named Nurse Leah. On the left, there are sliders for various needs: Bladder (-25.50), Hunger (100.00), Energy (100.00), Hygiene (-47.34), Fun (-69.04), Decoration (0.00), Dirtiness (0.00), and Fatigue (100.00). The main area contains a table of actions with columns for Score, Interaction, Object, Output, Blocked, and Outputs. The table lists various interactions such as 'CraftItemAutonomously', 'CheckStock', 'Research', 'bookshelf\_ReadSomething', 'SweepFloor', 'ScrubFloor', 'Socialize With', 'CollectObject', 'GroupChat', and 'Socialize With'. The bottom of the window features a 'History' tab and a 'Refresh' button.

| Score   | Interaction             | Object                | Output   | Blocked            | Outputs                                                                          |
|---------|-------------------------|-----------------------|----------|--------------------|----------------------------------------------------------------------------------|
| 30.8978 | CraftItemAutonomously   | CraftingStationHerbal | 31.2...  | GenericUnknown     | RoleNurse: 31.25 Physician: 0.00 Wizard: 0.00 Spy: 0.00                          |
| 30.8978 | CheckStock              | CraftingStationHerbal | 31.25... | GenericUnknown     | RoleNurse: 31.25 Physician: 0.00                                                 |
| 29.9301 | Research                | Archives              | 31.25... | GenericUnknown     | RoleNurse: 31.25 Wizard: 0.00 Physician: 0.00                                    |
| 22.0138 | bookshelf_ReadSomething | bookshelfPhysician2x1 | 22.73... | GenericUnknown     | RoleNurse: 15.63 Read: 5.00 Fun: 2.11 BeBlacksmithCustomer: 0.00 Physician: 0.00 |
| 15.4703 | SweepFloor              | Leah Blackburn        | 15.62... | GenericUnknown     | RoleNurse: 15.63 Blacksmith: 0.00 Physician: 0.00                                |
| 15.4703 | ScrubFloor              | Leah Blackburn        | 15.62... | GenericUnknown     | RoleNurse: 15.63 Blacksmith: 0.00 Physician: 0.00                                |
| 1.7245  | Socialize With          | Lukas Pritchard       | 1.7245   | GenericUnknown     | Socialize With Lukas Pritchard                                                   |
| 1.7245  | Socialize With          | Yolande Hart          | 1.7245   | GenericUnknown     | Socialize With Yolande Hart                                                      |
| 0.0000  | Socialize With          | Susannah Crockett     | 0.0000   | GenericUnknown     | Socialize With Susannah Crockett                                                 |
| Invalid | CollectObject           | HerbPlantSpawner02    | 0.0000   | Tuning_SkillTooLow |                                                                                  |
| Invalid | CollectObject           | HerbPlantSpawner02    | 0.0000   | Tuning_SkillTooLow |                                                                                  |
| Invalid | CollectObject           | HerbPlantSpawner02    | 0.0000   | Tuning_SkillTooLow |                                                                                  |
| Invalid | GroupChat               | Leah Blackburn        | 0.0000   | Def_TestFailed     |                                                                                  |
| Invalid | SocializeWith           | Leah Blackburn        | 0.0000   | Def_TestFailed     |                                                                                  |
| Invalid | GroupChat               | Susannah Crockett     | 0.0000   | Def_TestFailed     |                                                                                  |
| Invalid | GroupChat               | Lukas Pritchard       | 0.0000   | Def_TestFailed     |                                                                                  |
| Invalid | GroupChat               | Yolande Hart          | 0.0000   | Def_TestFailed     |                                                                                  |

Autonomy Tool Tabs

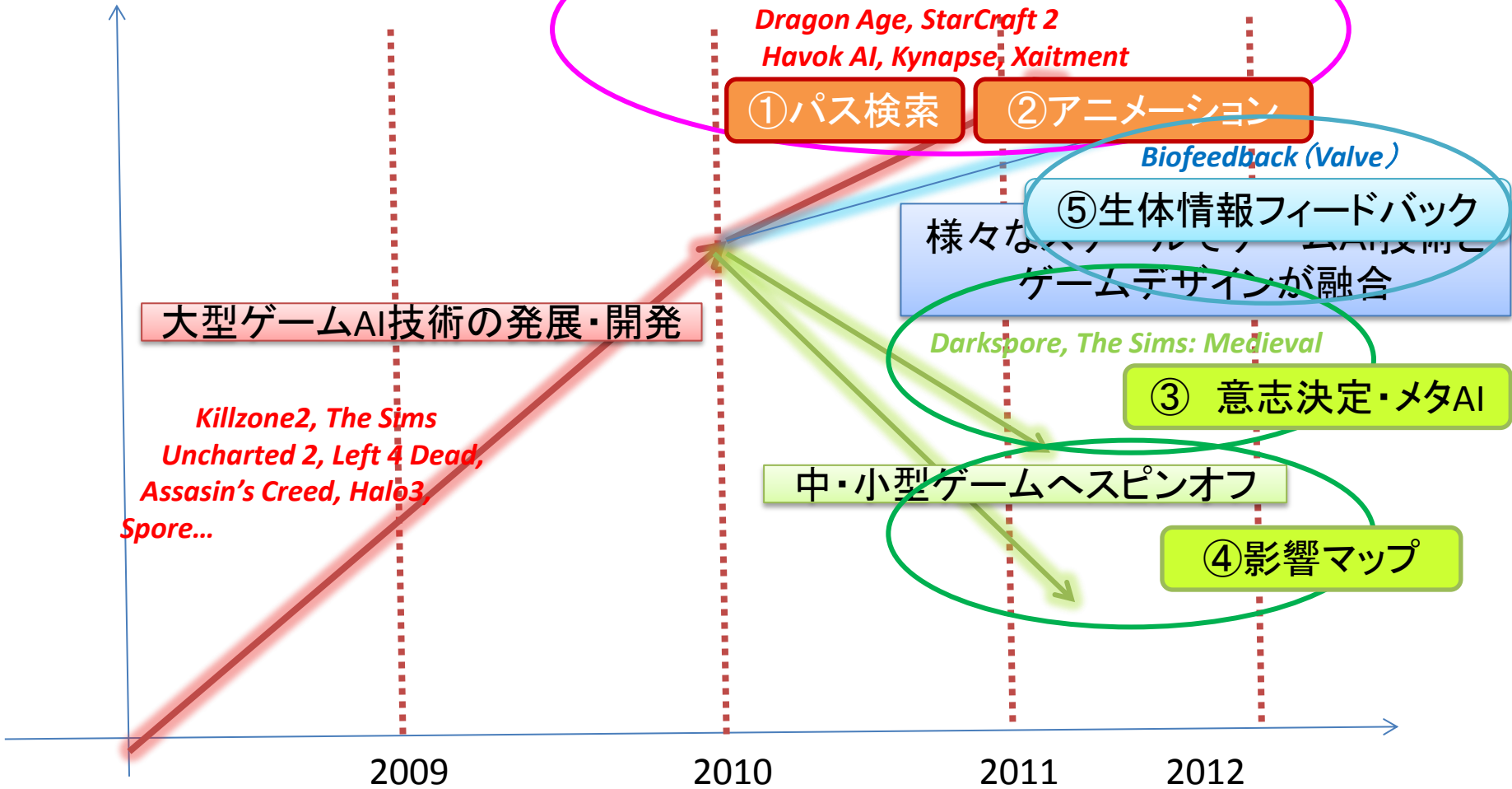
History

Refresh

小型 The Sims のアルゴリズム

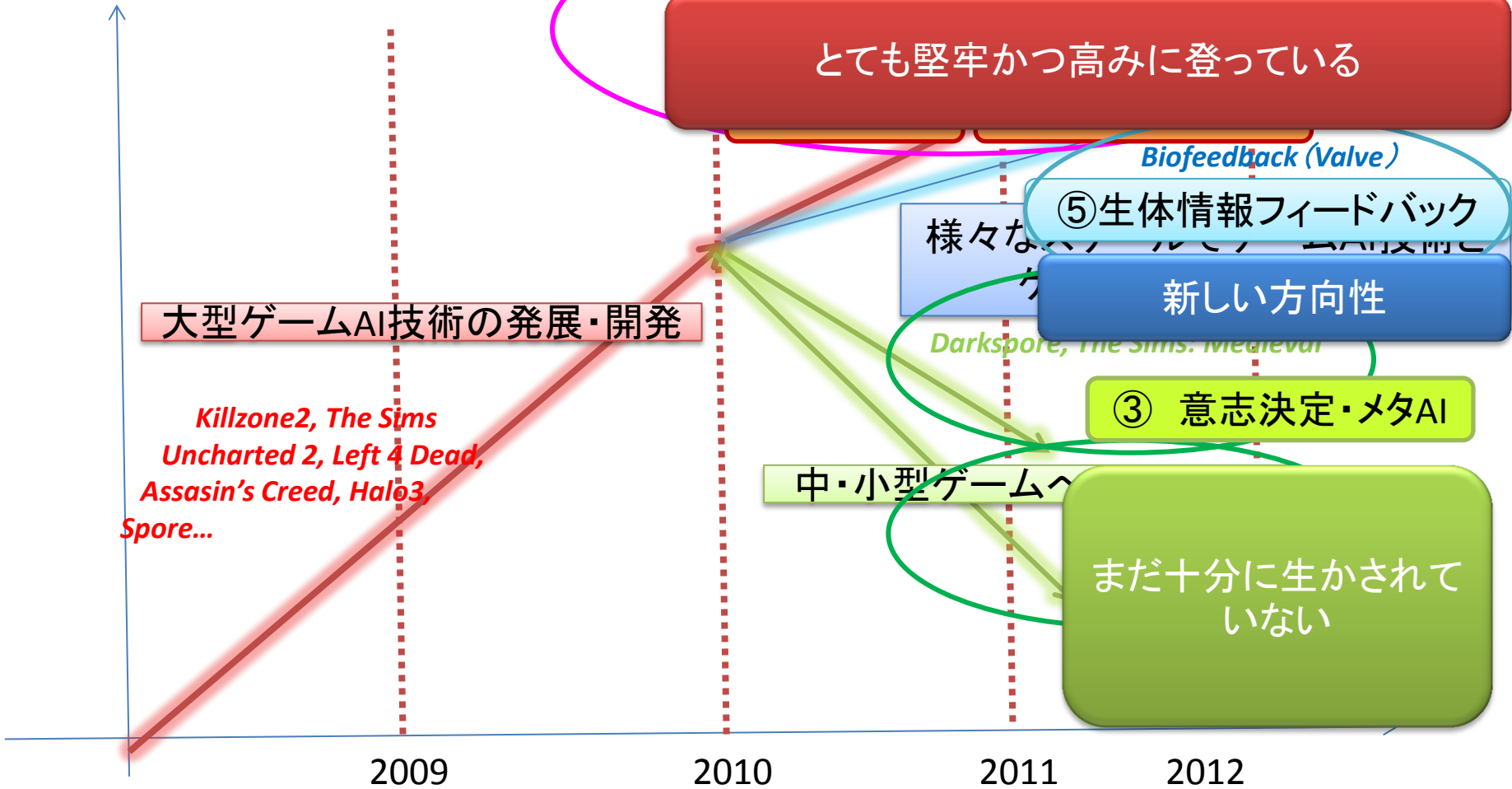
# まとめ

ゲームサイズ



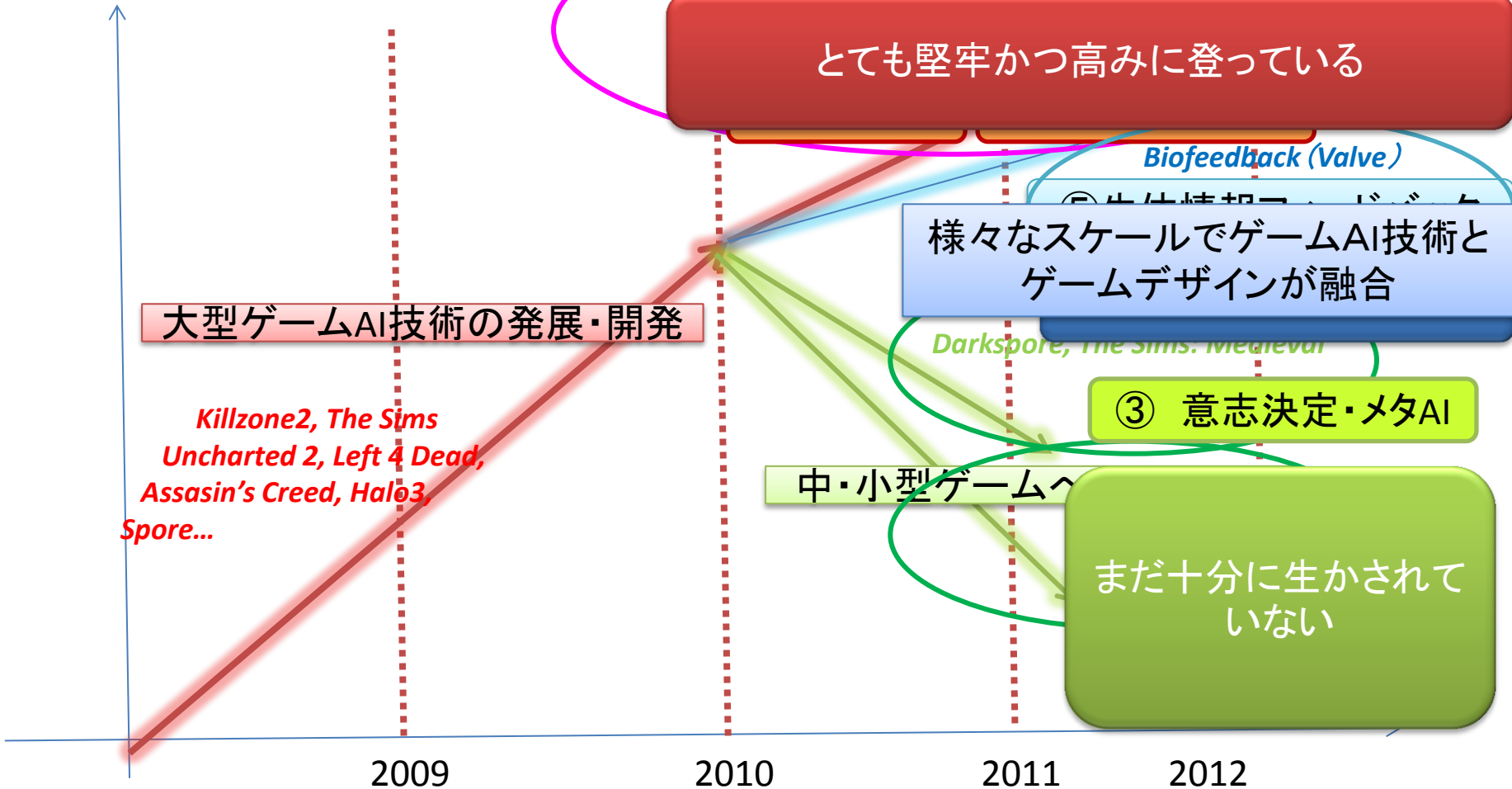
# まとめ

ゲームサイズ



# これからの課題

ゲームサイズ





# これからの課題

- パス検索・アニメーションの部門は、高度に完成されつつある。
- 意志決定の部分はまだまだだ。



大型のAIアルゴリズムが細分化されて、小さなゲームから大きなゲームまでに、AI技術の浸透が始まる。

デザイナーが真に活躍する時が来る。

デザイナーとプログラマーが協調して取り組む時代が来ている。