

ゲームAI入門 「環境と知性の相互作用」

三宅陽一郎

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

Twitter: @miyayou

感想・ご質問は メール twitter やfacebookまで

2014.4.18 @東京工科大学八王子キャンパス

学歴？



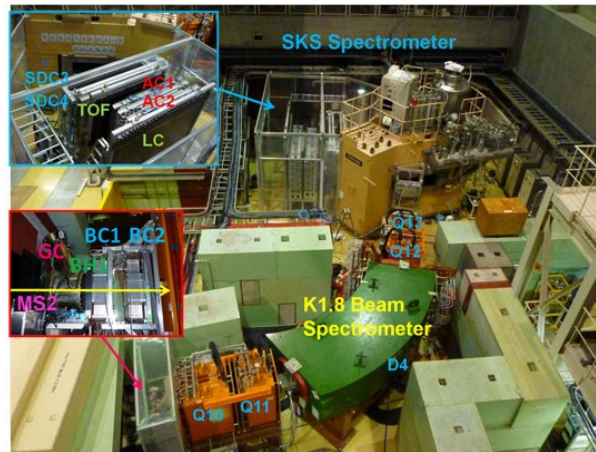
京都大学(数学)



大阪大学(原子核実験物理)

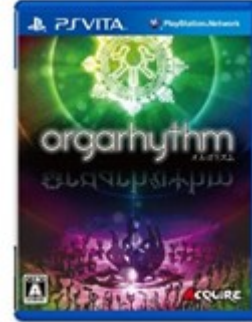
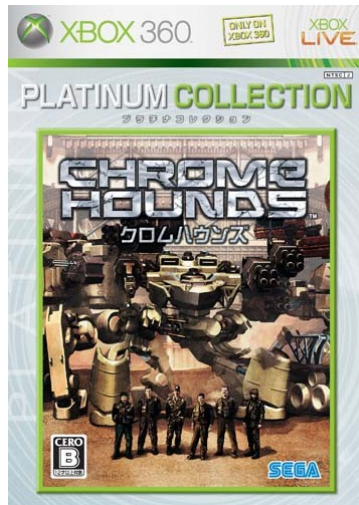


東京大学
(エネルギー工学/人工知能)

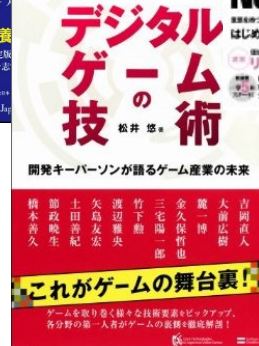
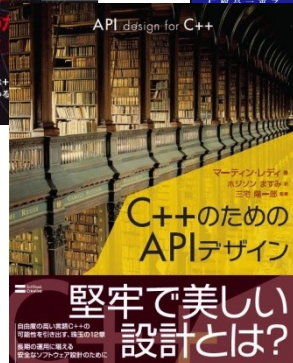
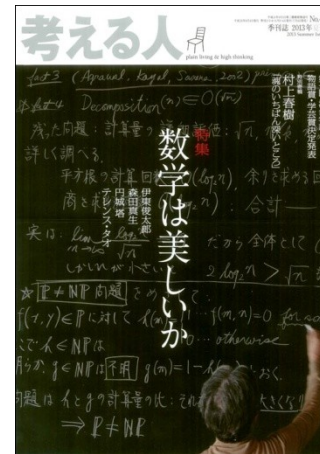
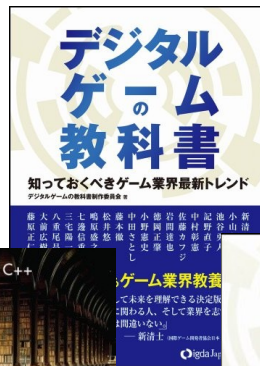


高エネルギー加速器研究所(半年ぐらい。修士論文)

Works (2006-2012)



AI for Game Titles



AI for Books

考える人



plain living & high thinking

No.45
季刊誌 2013年夏号
2013 Summer Issue

fact 3 (Agrawal, Kayal, Saxena, 2002) primality testing

村^{特別賞}上^{特別賞}春^{特別賞}樹^{特別賞} 魂のいちばん深いところ

第二回 河合隼雄 物語賞・学芸賞決定発表

Decomposition $(n) \in O(\sqrt{n})$

残った問題: 計算量の詳細評価: $\sqrt{n}$, $\log_2 n$

詳しく調べる.

平方根の計算回数 $(\log_2 n)$, 余りを求める回数 $(\log_2 n)$: 合計 $2 \log_2 n$

商と剰余 $O(\log_2 n)$: 合計 $3 \log_2 n$

実は: $\lim_{n \rightarrow \infty} \frac{\log_2 n}{\sqrt{n}} = 0$ だから全体として $O(\sqrt{n})$

しかし n が小さいと $2 \log_2 n > \sqrt{n}$ あり

☆ $P \neq NP$ 問題をめぐって:

$f(x, y) \in P$ に対して $h(n) = 1 \dots f(m, n) = 0$ for some m

ここで $h \in NP$ は

月か. $g \in NP$ は $\boxed{\text{不明}}$ $g(m) = 1 - h(m)$ とおく.

問題は h と g の計算量の比: それぞれ n と n^2 くらい

$\Rightarrow P \neq NP$

伊東俊太郎 森田真生 田城塔 テレンス・タオ

考える人



plain living & high thinking

季刊誌 2013年夏号 2013年8月4日発行

特集 数学は美しいか

13

グラフィック	池田亮司 14 杉本博司 16 Orist 18 数学を変えた人々 20 算額 26
イントロダクション	数学の愉悅を味わうために 山本貴光 28
インタビュー	円城塔 天才数学者は、変人とはかぎらない 32
	伊東俊太郎 人は数学に何を求めてきたか 41
	三宅陽一郎 人工知能は数学を理解できるのか 48
	私が世界で一番美しいと思う数式・証明 54
エッセイ	森田真生 数学と情緒 56
ルポルタージュ・エッセイ	筑駒中高数学科学研究会 趣味は数学 64 竹内薫 超難問と天才数学者の 66
	長岡亮介 東大の数学入試問題は美しい 70 深川英俊 算数の数しきり 74
これも数学.	中島さち子 78 緑リュウジ 80 野崎昭弘 84 吉村仁 86 Orist 88
ブックガイド	発見と難問の森に遊ぶ 山本貴光 91
論文	テレンス・タオ 素数の研究——その構造とランダム性について 96

第1回 河合隼雄 物語賞・学芸賞 決定発表

107

選評 受賞のことば 西加奈子 藤原辰史

108

特別寄稿 村上春樹 魂のいちばん深いところ

102

考える人
鳥類学者は
恐竜をかく語りき
川上和人

188

新連載 富士——ある山の伝記 池内紀 134 アメリカン・レガシー 渡辺靖 148

連載 high thinking

ヨーロッパの身体性 童老孟司	126	柳田国男、今いずこ 山折哲雄	204
新地球紀行 大河内直彦	168	琥珀のアーカイブ 今福龍太	214
東北巡礼 吉本直子、中野晴生	176	生命の内と外 永田和宏	224
行ったり来たり マイケル・エメリック	186	チキユズイン 本内達郎	234
ニッポンの馬 内澤旬子	190	考えない 宮沢肇夫	238
月日の残像 (最終回) 山田太一	200		

連載 plain living

日本のすごい味 平松洋子	114	「犬が星見た」をめぐる旅 高山なおみ	156
京都寺町お茶ごよみ 渡辺都	124	娘と私 さげさかのりこ	164
私の暮らし (最終回) 大賀妙子	144	考える手 コシナのレンズ	240

連載 graphic special

ニッポンの里山 今森光彦	8	動物たちの惑星 岩倉光昭	248
--------------	---	--------------	-----

今号、80〜275ページ「ある山」は連載いたします。

Books

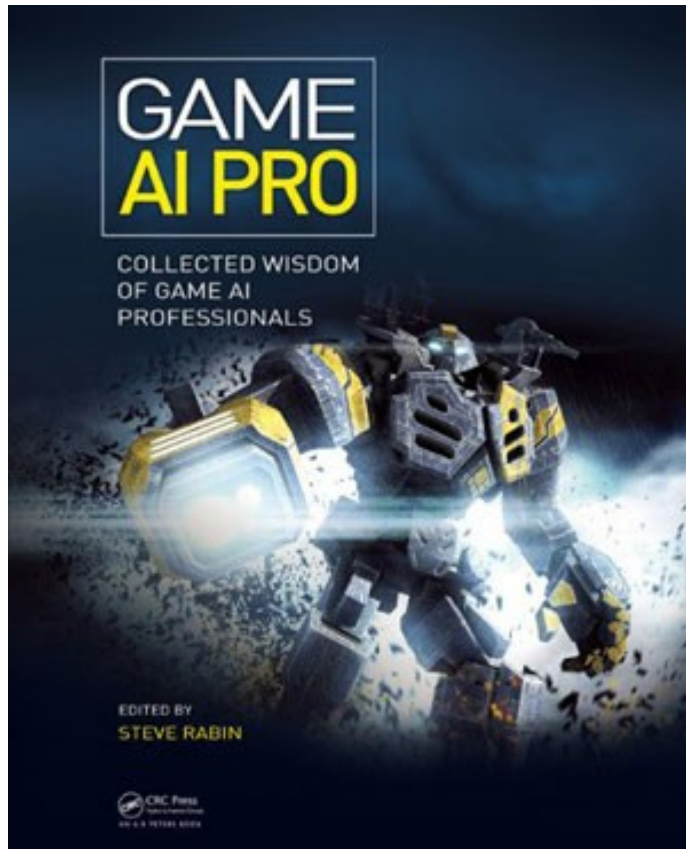


Table of Contents

General Wisdom

~

Movement and Pathfinding

Pathfinding Architecture Optimizations

Steve Rabin and Nathan Sturtevant

Choosing a Search Space Representation

Nathan R. Sturtevant

Creating High-Order Navigation Meshes through Iterative Wavefront Edge Expansions

D. Hunter Hale and G. Michael Youngblood

Precomputed Pathfinding for Large and Detailed Worlds on MMO Servers

Fabien Gravot, Takanori Yokoyama, and Youichiro Miyake

Techniques for Formation Movement using Steering Circles

Stephen Bjore

Collision Avoidance for Preplanned Locomotion

Bobby Anguelov

Crowd Pathfinding and Steering Using Flow Field Tiles

Elijah Emerson

Efficient Crowd Simulation for Mobile Games

Graham Pentheny

Animation-Driven Locomotion with Locomotion Planning

Jarosław Ciupiński

Strategy and Tactics

Tactical Position Selection: An Architecture and Query Language

Matthew Jack

Tactical Pathfinding on a NavMesh

Daniel Brewer

Beyond the Kung-Fu Circle: A Flexible System for Managing NPC Attacks

Michael Dawe

Hierarchical AI for Multiplayer Bots in Killzone 3

Remco Straatman, Tim Verweij, Alex Champandard, Robert Morcus, and Hylke Kleve

Using Neural Networks to Control Agent Threat Response

Michael Robbins

~

Precomputed Pathfinding for Large and Detailed Worlds on MMO Servers
Fabien Gravot, Takanori Yokoyama, and Youichiro Miyake

<http://www.crcpress.com/product/isbn/9781466565968>

FF14 A Realm Reborn Path-finding demo

FFXIVで使われているAI技術～敵NPCはどうやって経路を探索しているのか？

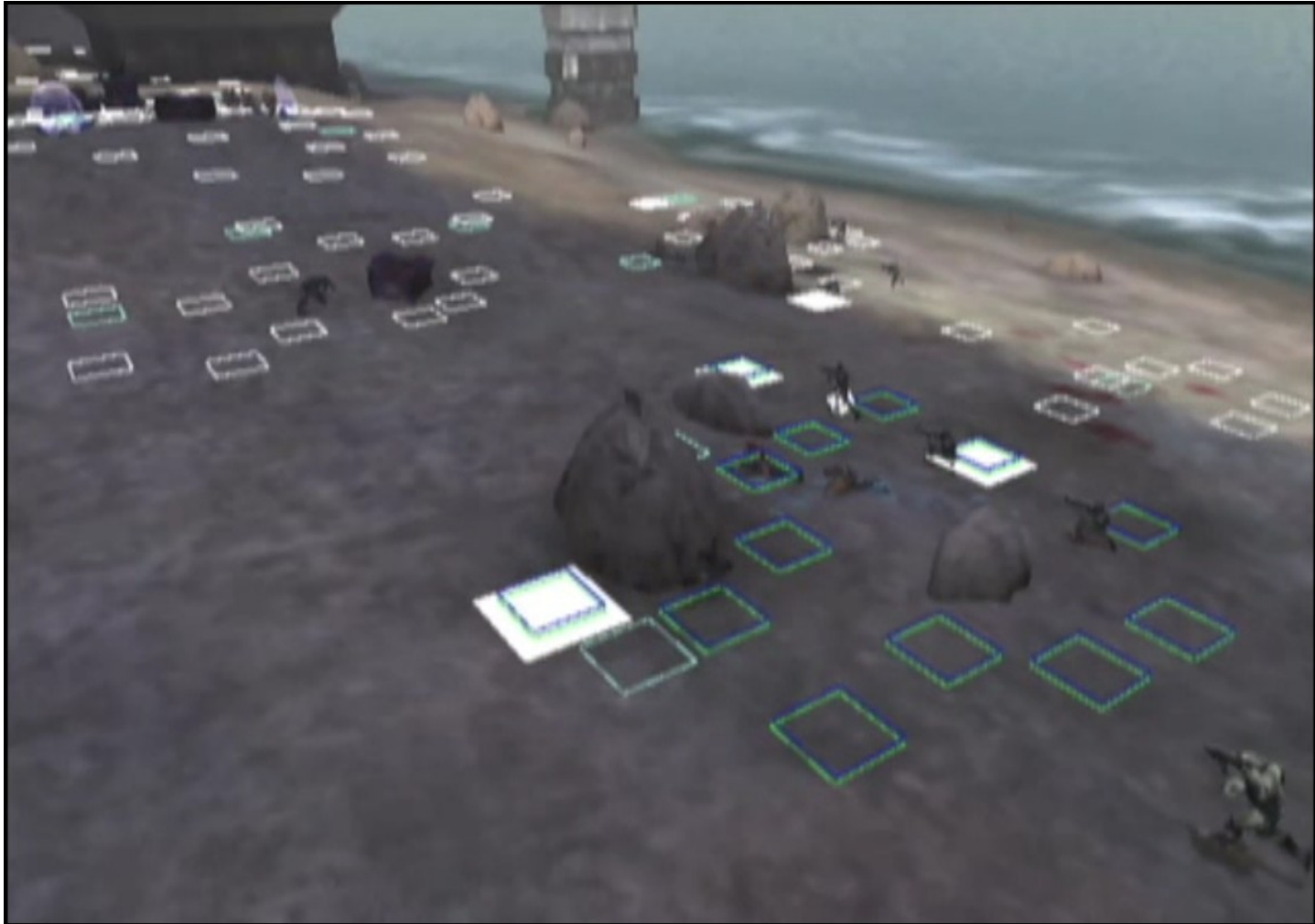
<http://www.4gamer.net/games/032/G003263/20121205079/>

(4gamers)

https://www.youtube.com/watch?v=9_vdpndn4jl

<https://www.youtube.com/watch?v=o3ihN2kSzMs>

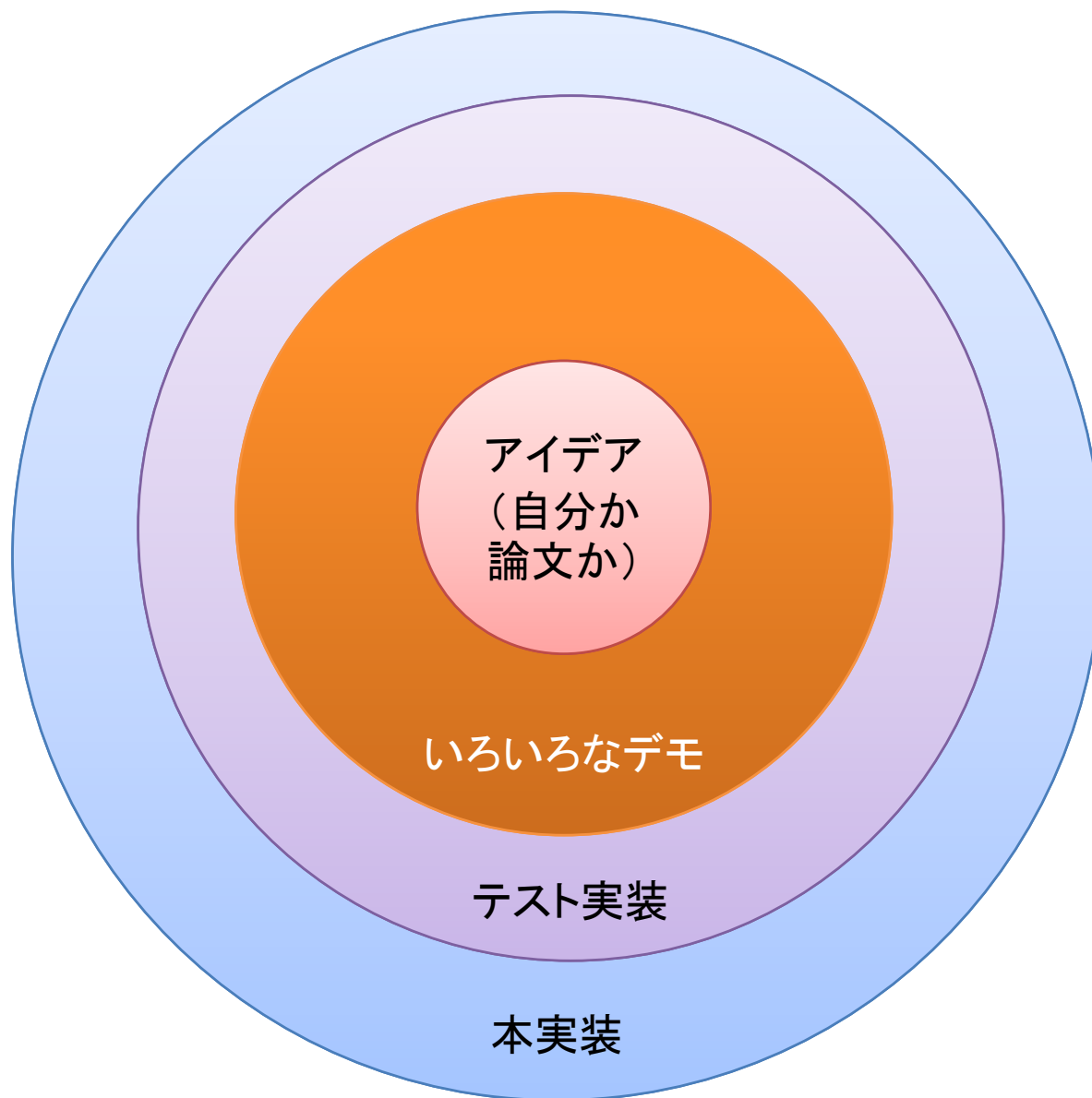
Halo3 multi-Agent demo



Bungie Publications

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

ゲーム技術の成立プロセス



デジタルゲームとは？

= 多様なユーザーの体験
(ユーザー・エクスペリエンス)を作ること。

楽しい
怖い
爽快
悲しい
不快
うっとうしい
心地よい

集めたい
壊したい
積み上げたい
使いたい
逃げたい
突入したい
ぼーとしていたい

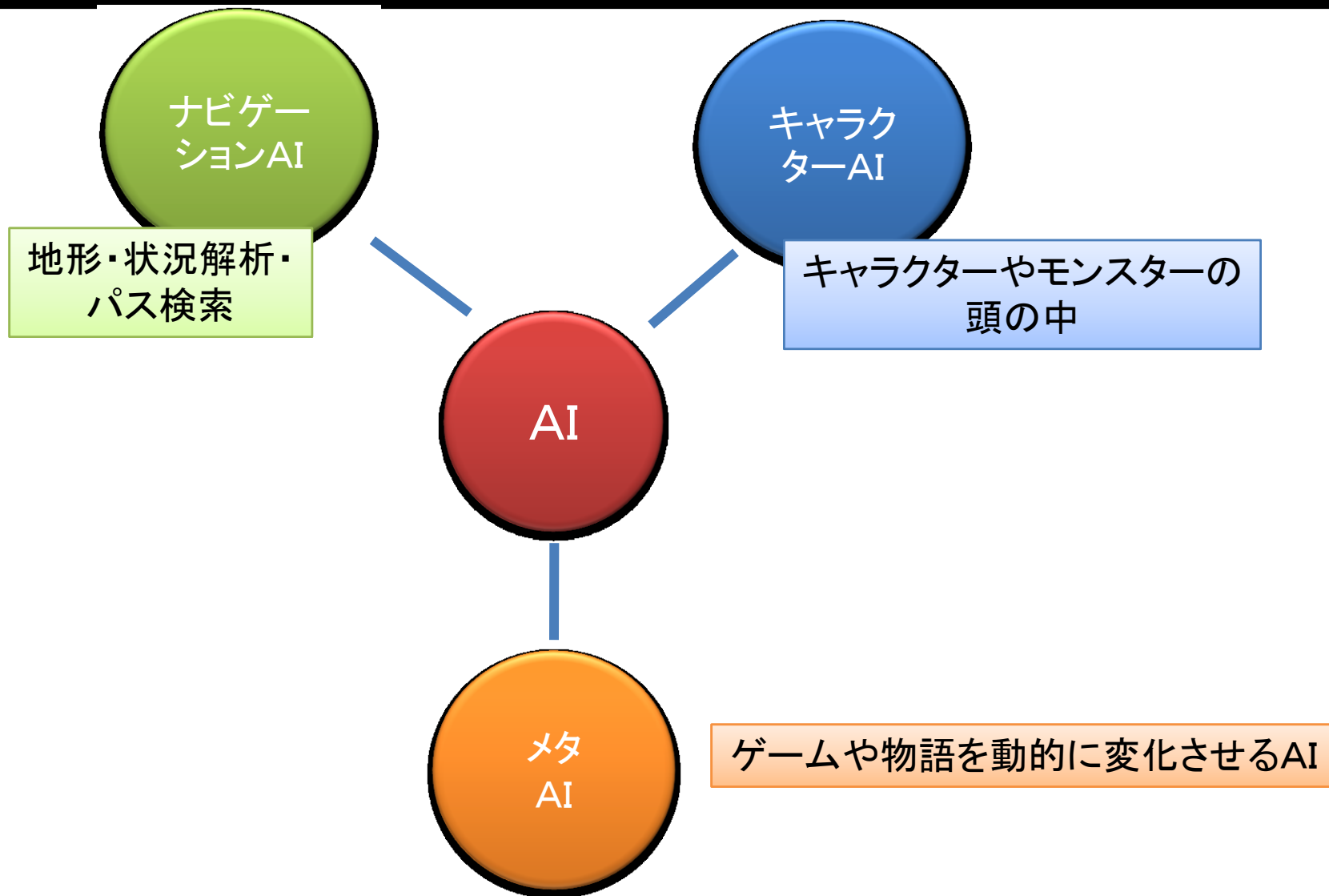
退屈
わくわく
うんざり
あきた
ずっとプレイしていたい
集めたい
消費したい

ユーザー・エクスペリエンスは 何から成り立っているか？

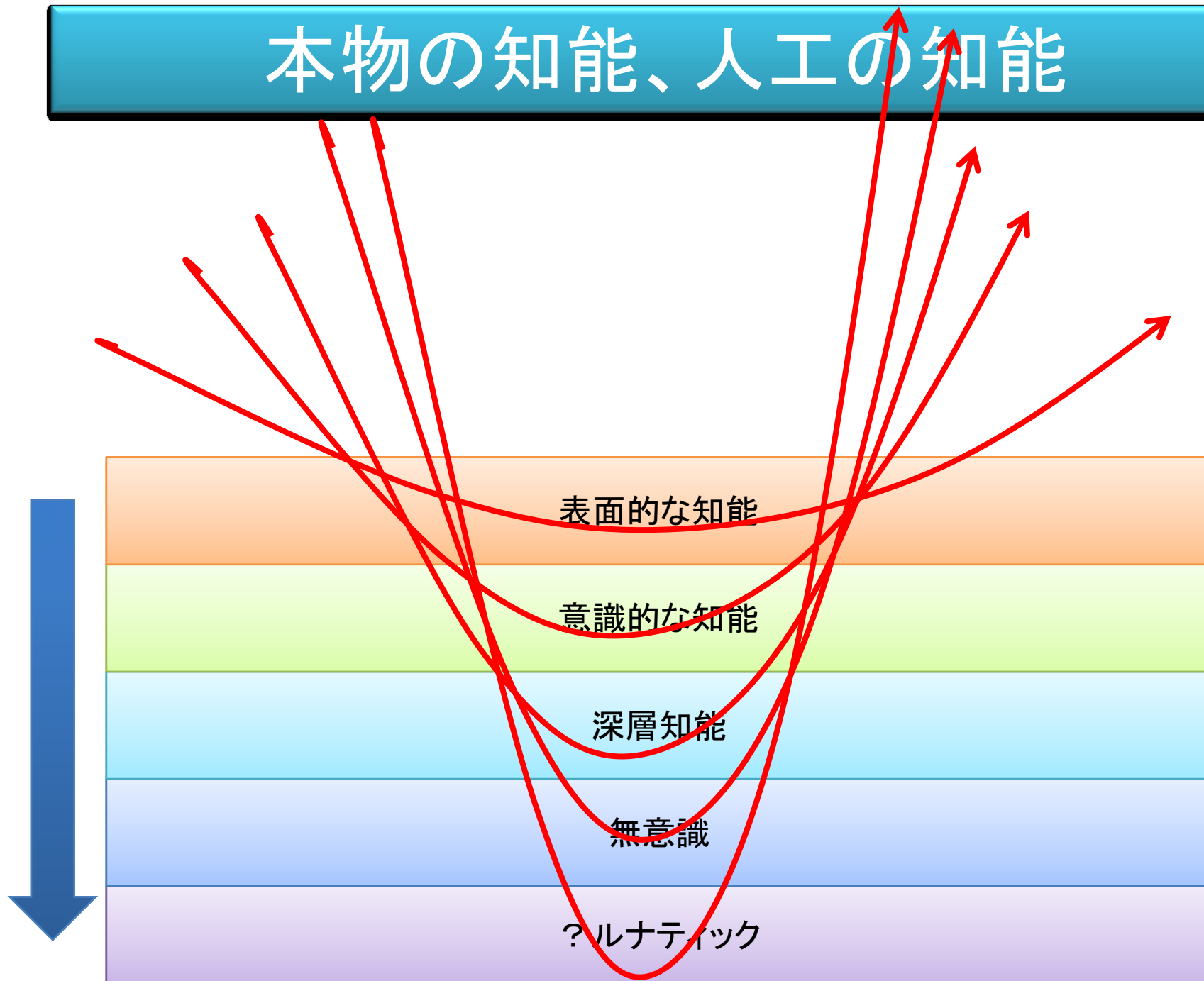
- ビジュアル
- サウンド
- 操作
- インタラクション
- 物語
- AI
-

AIだけでゲームができるわけではない。しかし、
AIなしではなし得ないユーザー・エクスペリエンスがある。

AIの定義

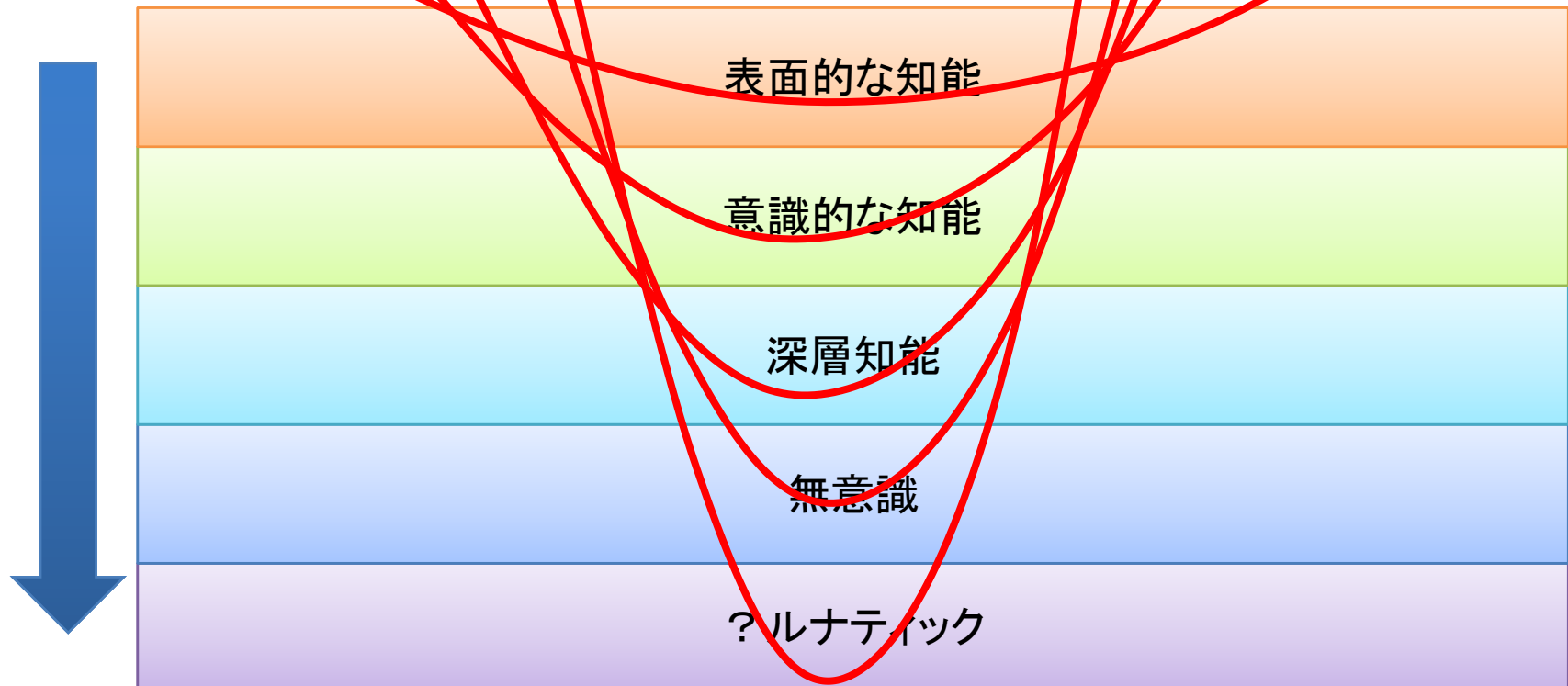


本物の知能、人工の知能

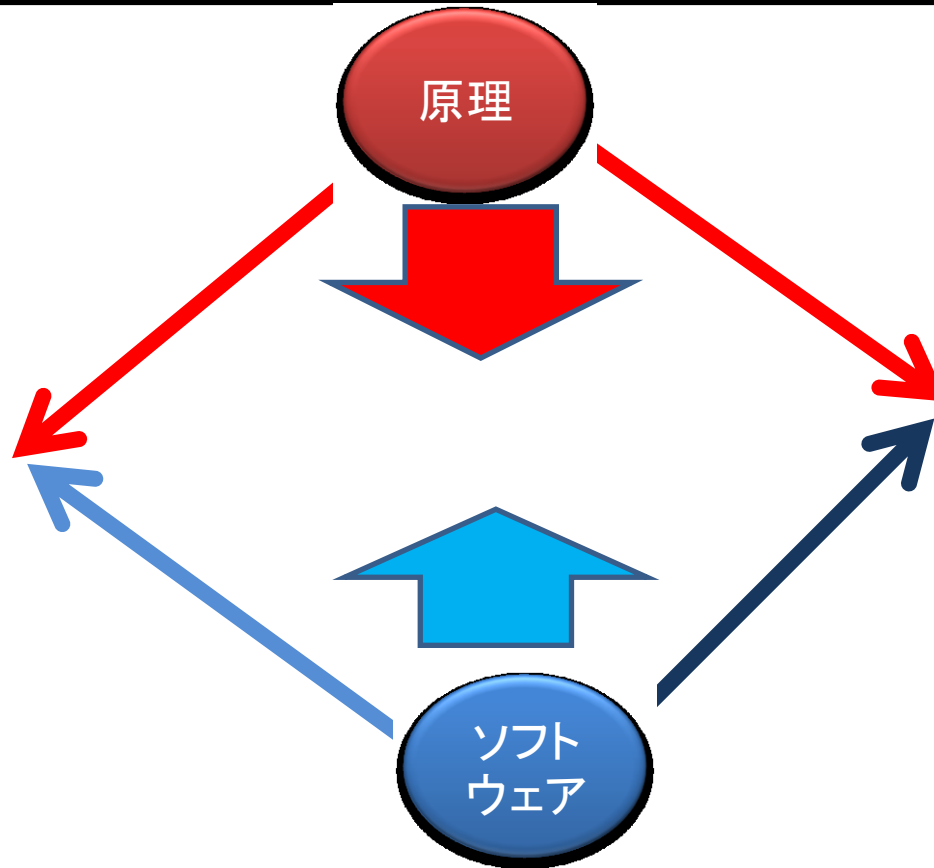


本物の知能、人工の知能

さまざまな階層の知能があるけれど、ゲームではユーザーが知能として感じられることが大切で、必ずしも本当の知能を作る必要はないが、ある程度の知能を作る必要がある。



二つのアプローチ



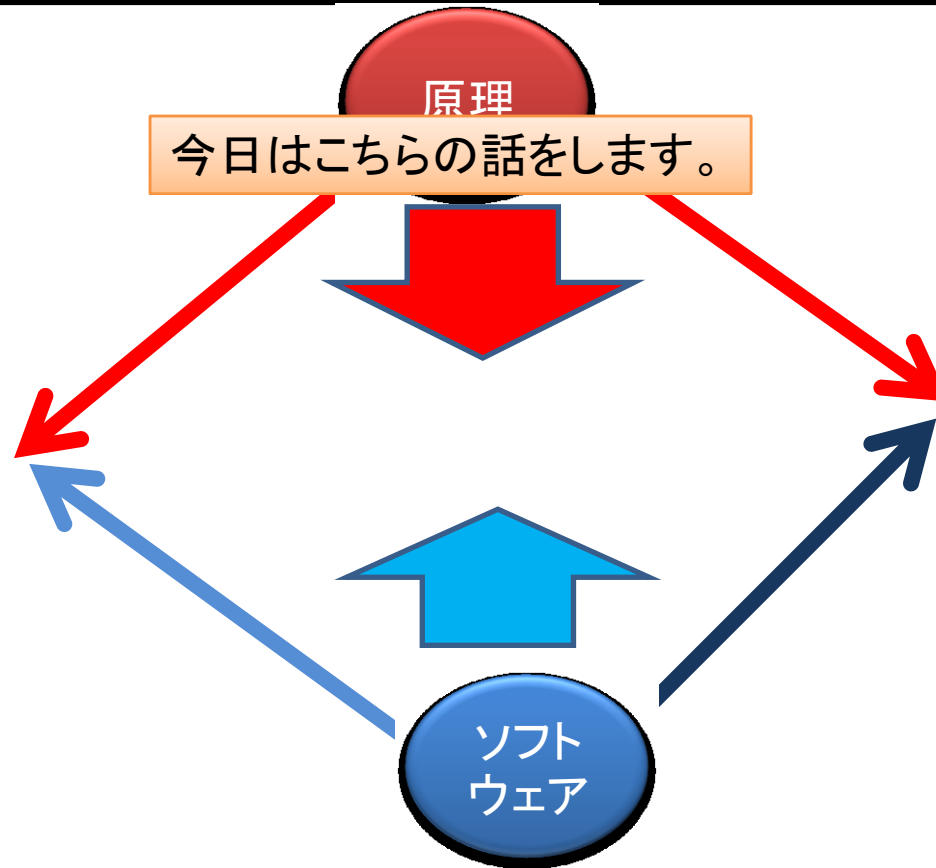
ソフトウェアからのアプローチ

= 自分の書いてるプログラムが「知能」となるように頑張る。(ボトムアップ)

原理からのアプローチ

= 人工知能かくあるべし、という理論から人工知能を作る。(トップダウン)

二つのアプローチ



ソフトウェアからのアプローチ

= 自分の書いてるプログラムが「知能」となるように頑張る。(ボトムアップ)

原理からのアプローチ

= 人工知能かくあるべし、という理論から人工知能を作る。(トップダウン)

知能の誕生と進化

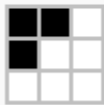
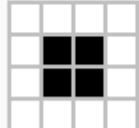
生物は地球上のいろいろな環境で進化して来た。

- 生物は環境と共に進化して来た。
- 生物は地球という環境にフィットした存在。
- 生物は地球という環境が産み出した知性。
- 生物は絶対的な知性ではない。
- 絶対的な知性は存在しない。

LIFE GAME は知能だろうか？



ライフゲームの基本ルール

誕生	生存(維持)	死(過疎)	死(過密)
			

ライフゲームは自己完結型細胞。

＝常に自分自身と周囲の状態だけで次の状態が決まる。

＝内部状態が物理状態と同じ。(あまりに抽象的)

＝自分自身への認識がない。(センサーもない)

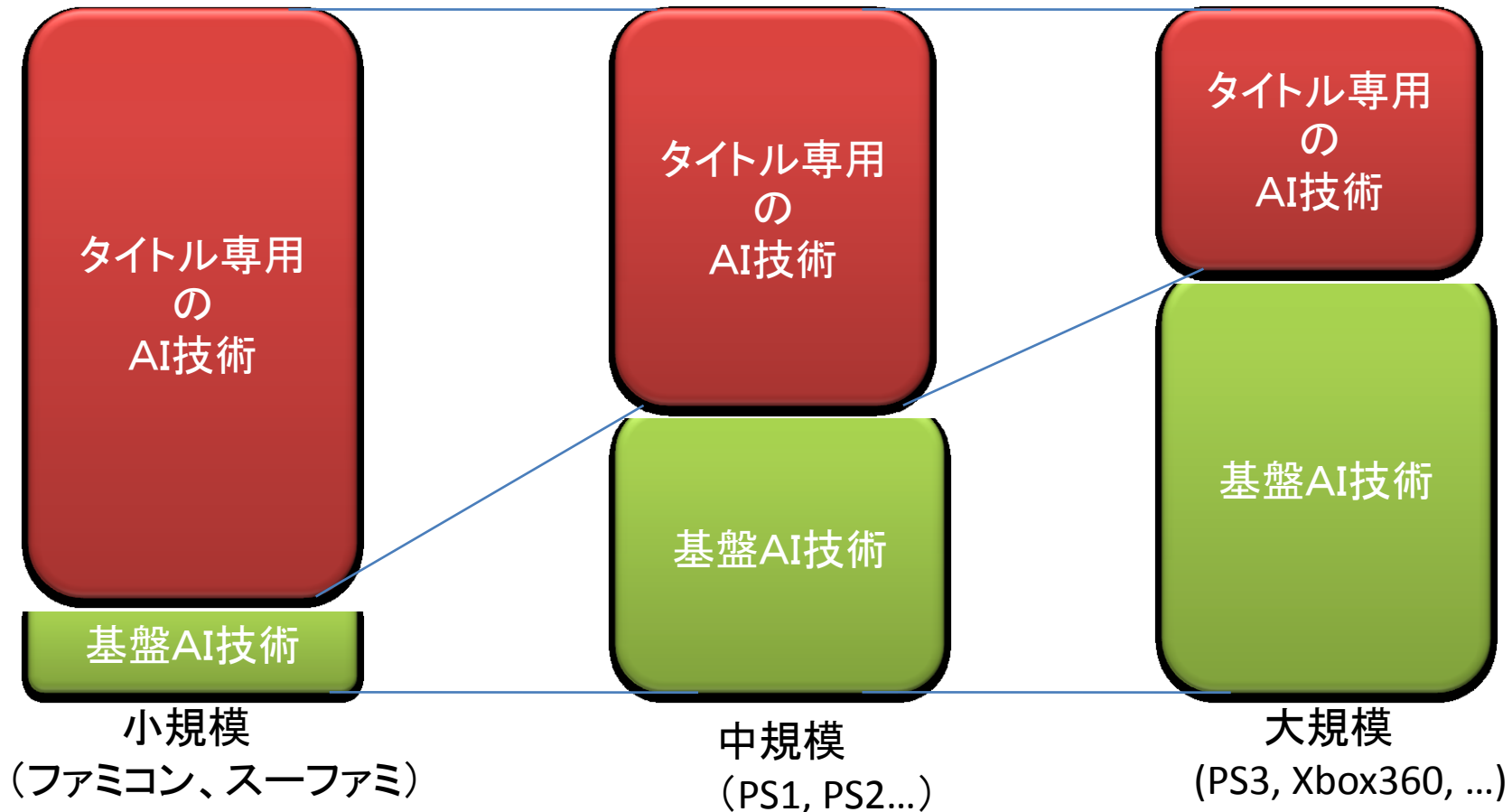
＝現象

<http://archive.globalgamejam.org/2011/life-game-shooter>

知能は相対的なものである。

- 絶対はない。
- 環境に対して知性がある。逆に、知性に対して環境がある。
- 知性のコア、環境のダイナミクス。

AI技術の流れ



一般的に、小規模から大規模ゲームになるほど、基盤技術の占める割合が大きくなる。なぜなら、大規模ほどアセットが大きくデータドリブンの仕組み(=汎用)が必要だから。

ここでいきなり問題。

- $X^4 + A \cdot X^3 + B \cdot X^2 + C \cdot X + D = 0$

(一般解を求めよ)

とても難しい。

- $X^4 + 2 \cdot X^2 + 1 = 0$

(特殊解を求めよ)

簡単。

ここでいきなり問題。

- $X^4 + A \cdot X^3 + B \cdot X^2 + C \cdot X + D = 0$

(一般解を求めよ)

とても難しい。

- $X^4 + 2 \cdot X^2 + 1 = 0$

(特殊解を求めよ)

簡単。

実際のゲーム開発では、技術的に難しい問題につきあたった場合、真正面から解く前に、**そのゲーム固有の条件で問題が簡単にならないか、或いは、仕様を確認して簡単にできないか、**を検討する。

一方、学問では正面突破しなければ意味がない。
しかし、その解法が冗長だった場合、実際の開発では使えない。

それがアカデミズムと産業応用の違い。

ゲーム開発は特殊解で良い。 ゲーム研究は一般解を。

- アカデミックの研究は一般的な研究でなければならない。
- しかし、ゲーム開発の現場は特殊な条件(そのゲームの条件)で解決できれば良い。
= 難しければ、ゲームデザインに支障のない範囲でもっと簡単になるように条件を変える。
- ゲーム開発の基礎技術(ゲームエンジン開発など)は、その中間領域。判断はケースバイケース。だが、できるだけ一般的に解く。

第一部

エージェントAI

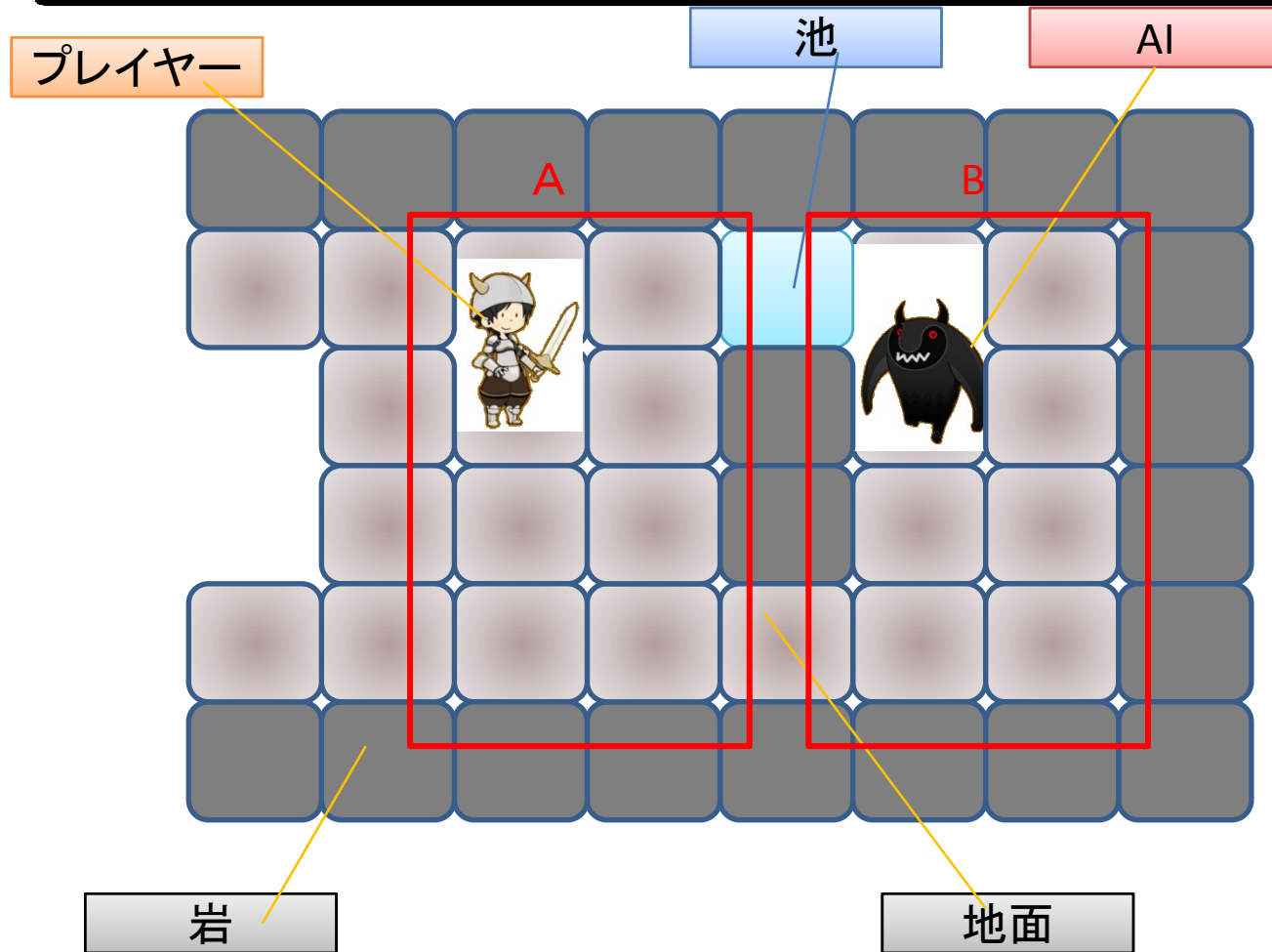
このAIはどう組むか？



このAIはどう組むか？



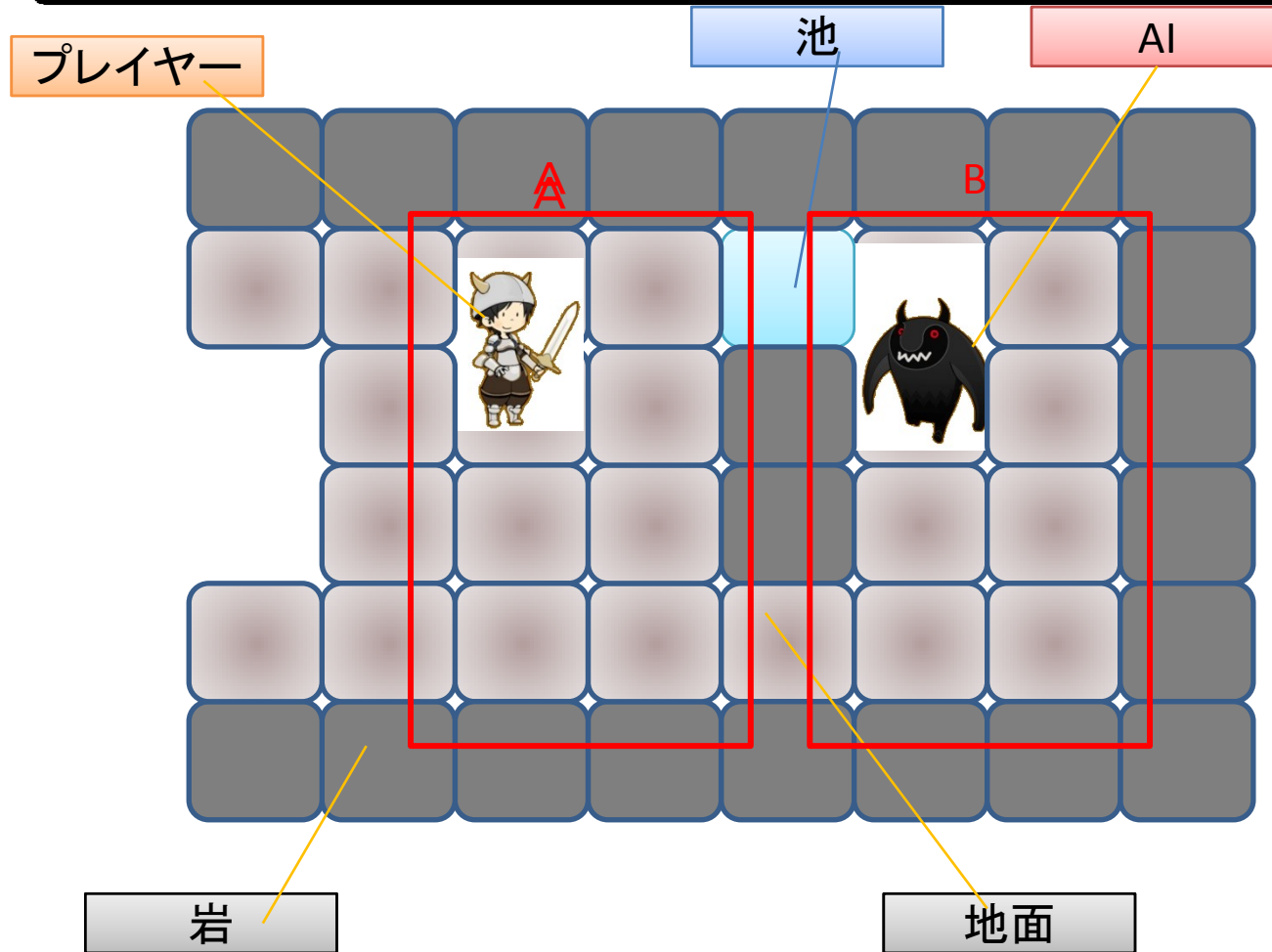
このAIはどう組むか？



(例)
プレイヤーがAにいれば
Bをうろうろする。

プレイヤーがBにいれば
近づいて攻撃する。

このAIはどう組むか？

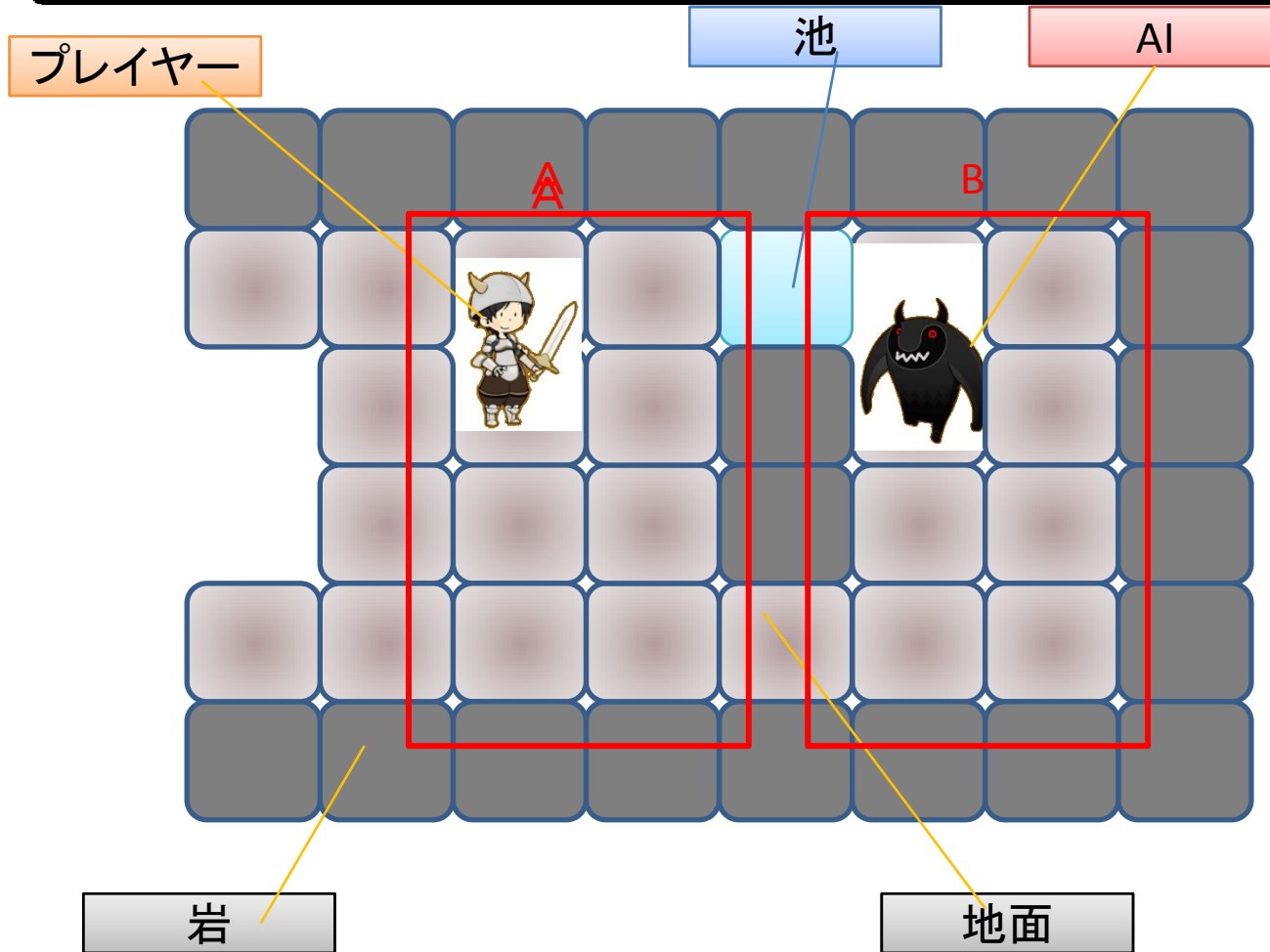


(例)
プレイヤーがAにいれば
Bをうろうろする。

プレイヤーがBにいれば
近づいて攻撃する。

これは、まず書く人の思考がまずあり、その結果を書いている。

このAIはどう組むか？



(例)
プレイヤーがAにいれば
Bをうろうろする。

プレイヤーがBにいれば
近づいて攻撃する。

これは、まず書く人の思考がまずあり、その結果を書いている。

つまりAI自身は何も知らない操り人形。



Scripted AI から自律型AIへの変化

ゲームデザイナーの頭の中



スクリプト



Scripted AI

ゲームデザイナーの頭の中



知識



思考



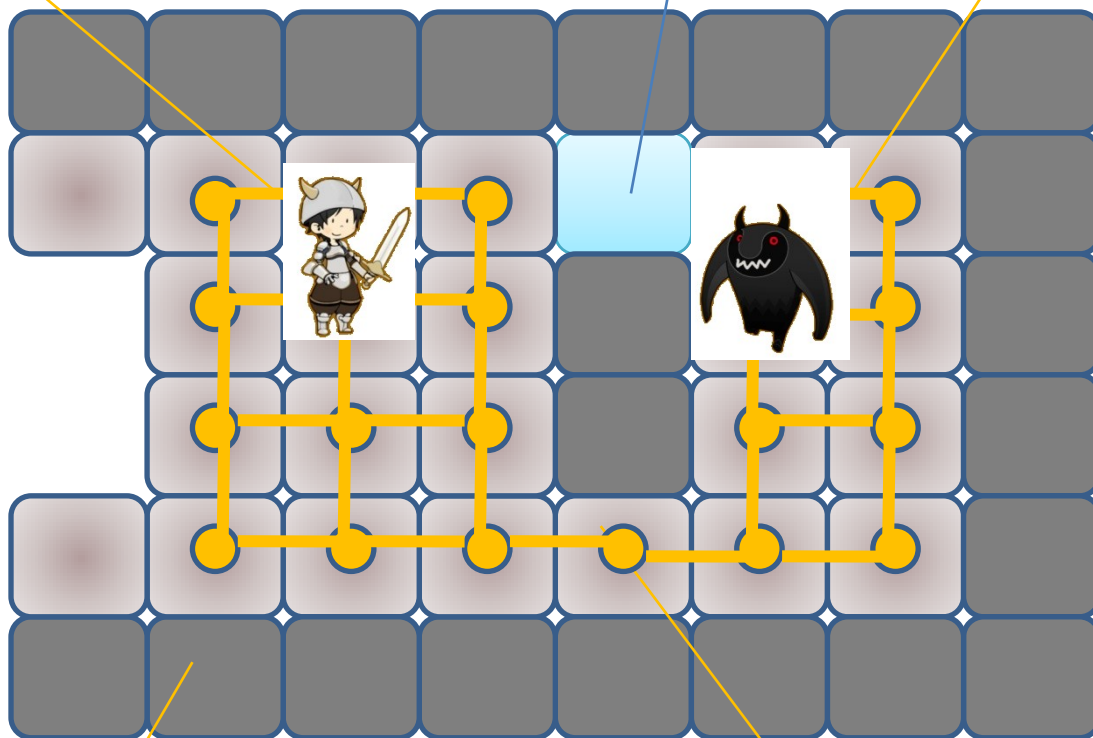
自律型 AI (Autonomous AI)

このAIはどう組むか？

プレイヤー

池

AI



地形データ
(Way
Points)

地形デー
タを用いた
ロジック

敵へのパス検索

岩

地面

AIを作る

人間

地形データ
(Way
Points)

オブジェク
トデータ

自分の
身体データ

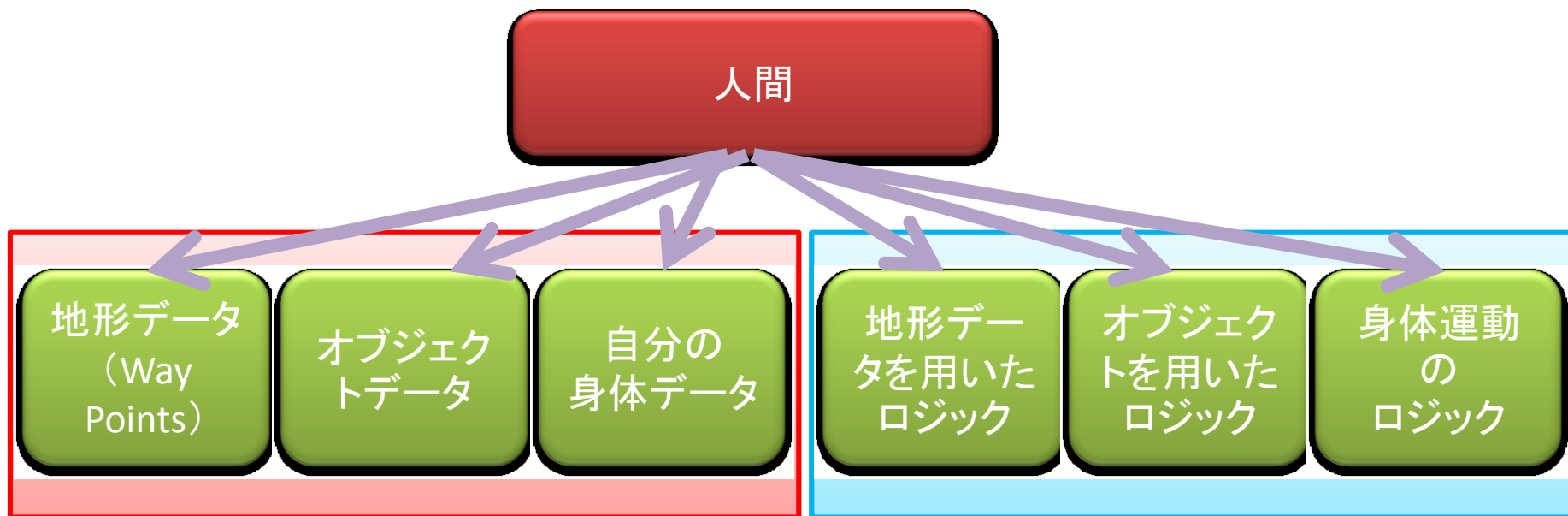
地形デー
タを用いた
ロジック

オブジェク
トを用いた
ロジック

身体運動
の
ロジック



AIを作る

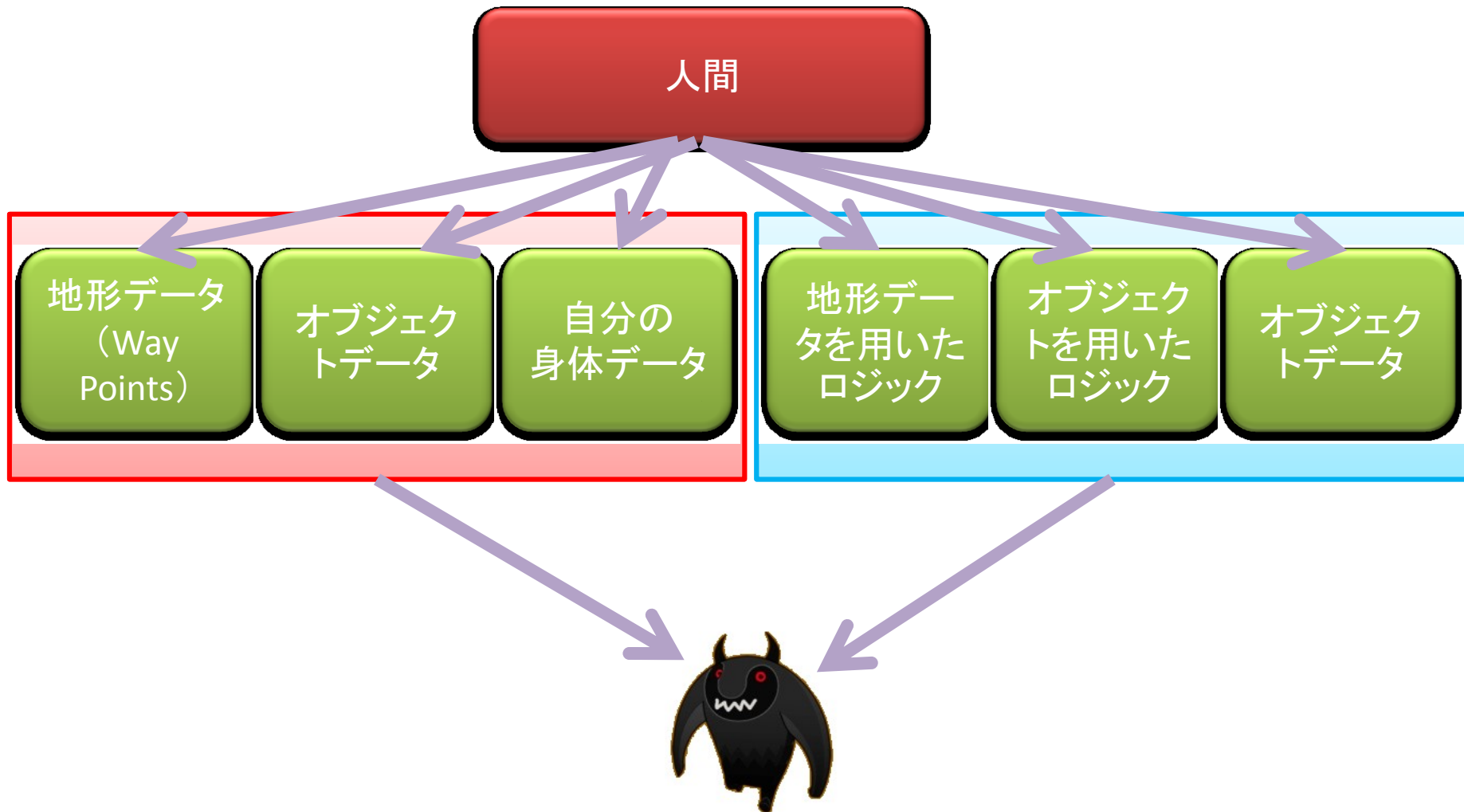


知識を(データ)表現する。(知識表現)

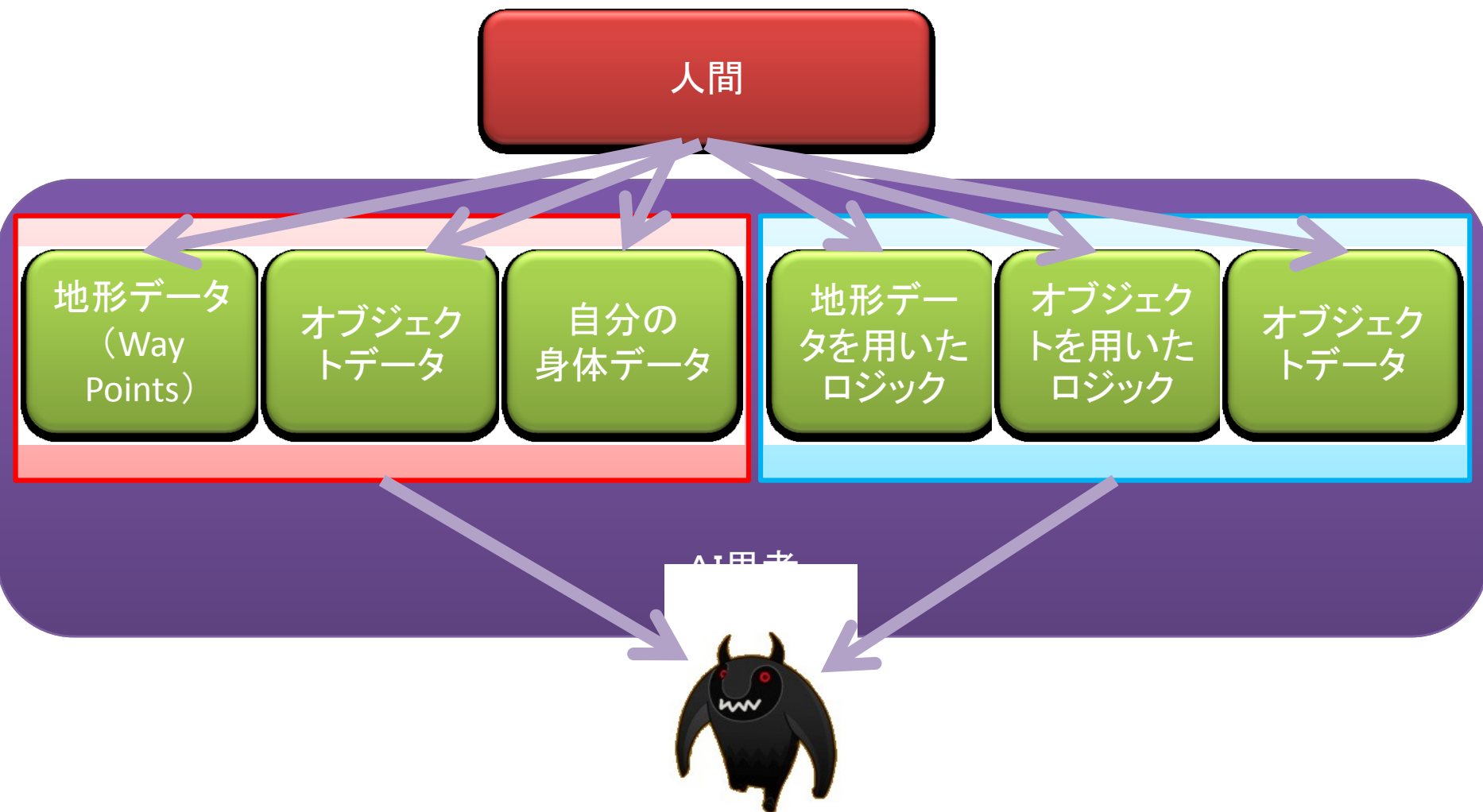
知識に基づいて思考する。(AI思考)



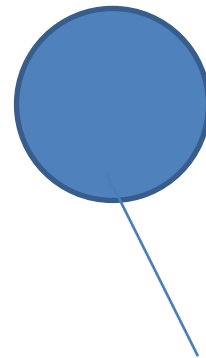
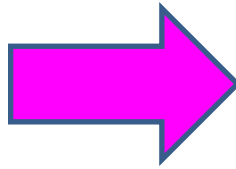
AIを作る＝AIに知識と思考を与えること。



AIを作る＝AIに知識と思考を与えること。



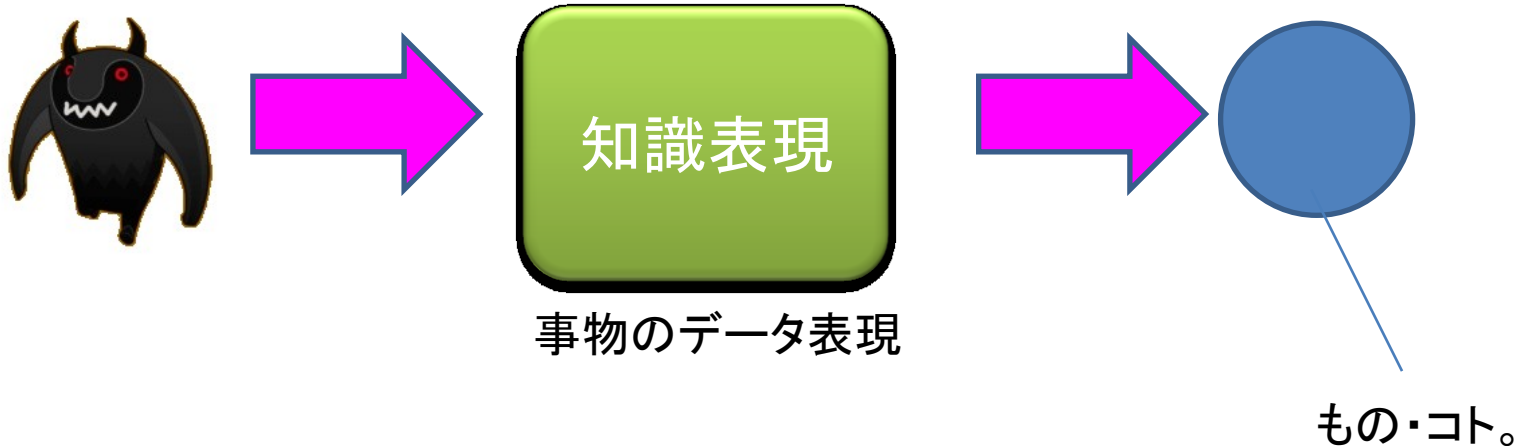
知識表現とは



もの・コト。

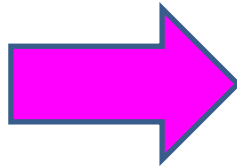
AIは直接、事物(物・事)を理解することはできない。

知識表現とは



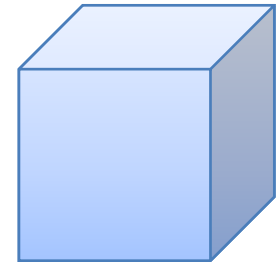
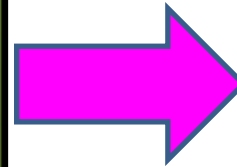
AIは直接、事物(物・事)のデータ表現を通じて世界にアクセスする。

知識表現とは



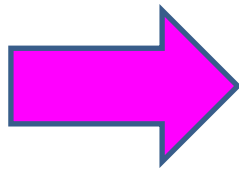
- オブジェクト
- サイズは1m
- 硬い
- 動かすことができる
- 壊すことができる
- 中にアイテムがある
-

事物のデータ表現

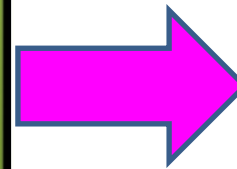


AIは直接、事物(物・事)のデータ表現を通じて世界にアクセスする。

知識表現とは



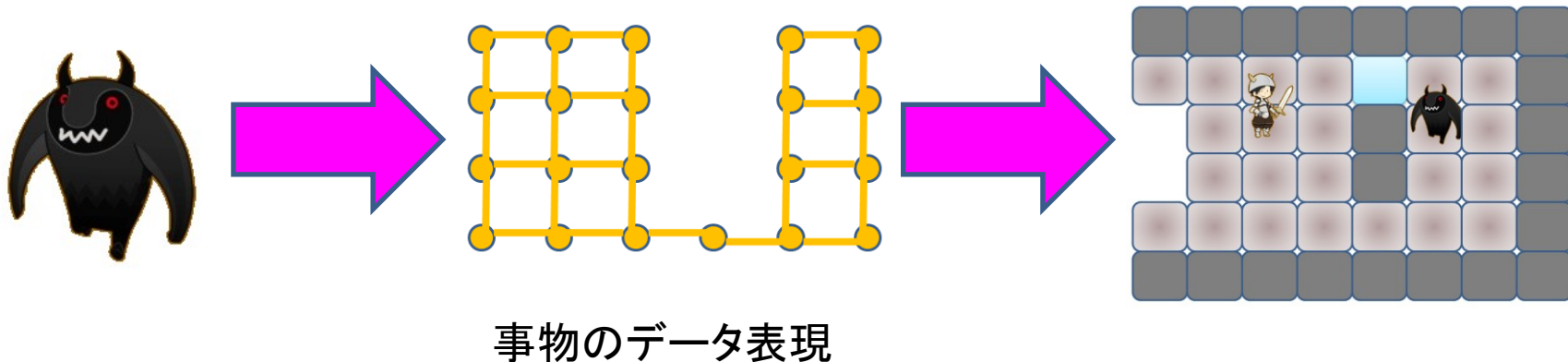
- プレイヤー
- サイズは1.8m
- 敵
- 剣を持っている
- 最大速度は10m/s
- 最大HPは100
- 現在座標(1,20,8)
- HPが半減すると回復魔法を使う
-



事物のデータ表現

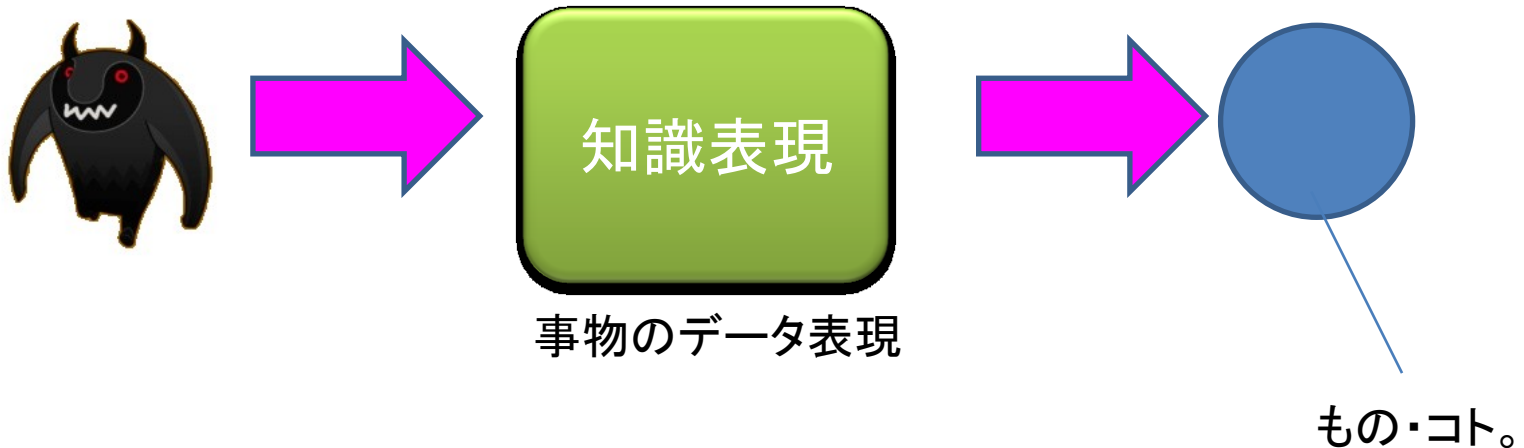
AIは直接、事物(物・事)のデータ表現を通じて世界にアクセスする。

知識表現とは



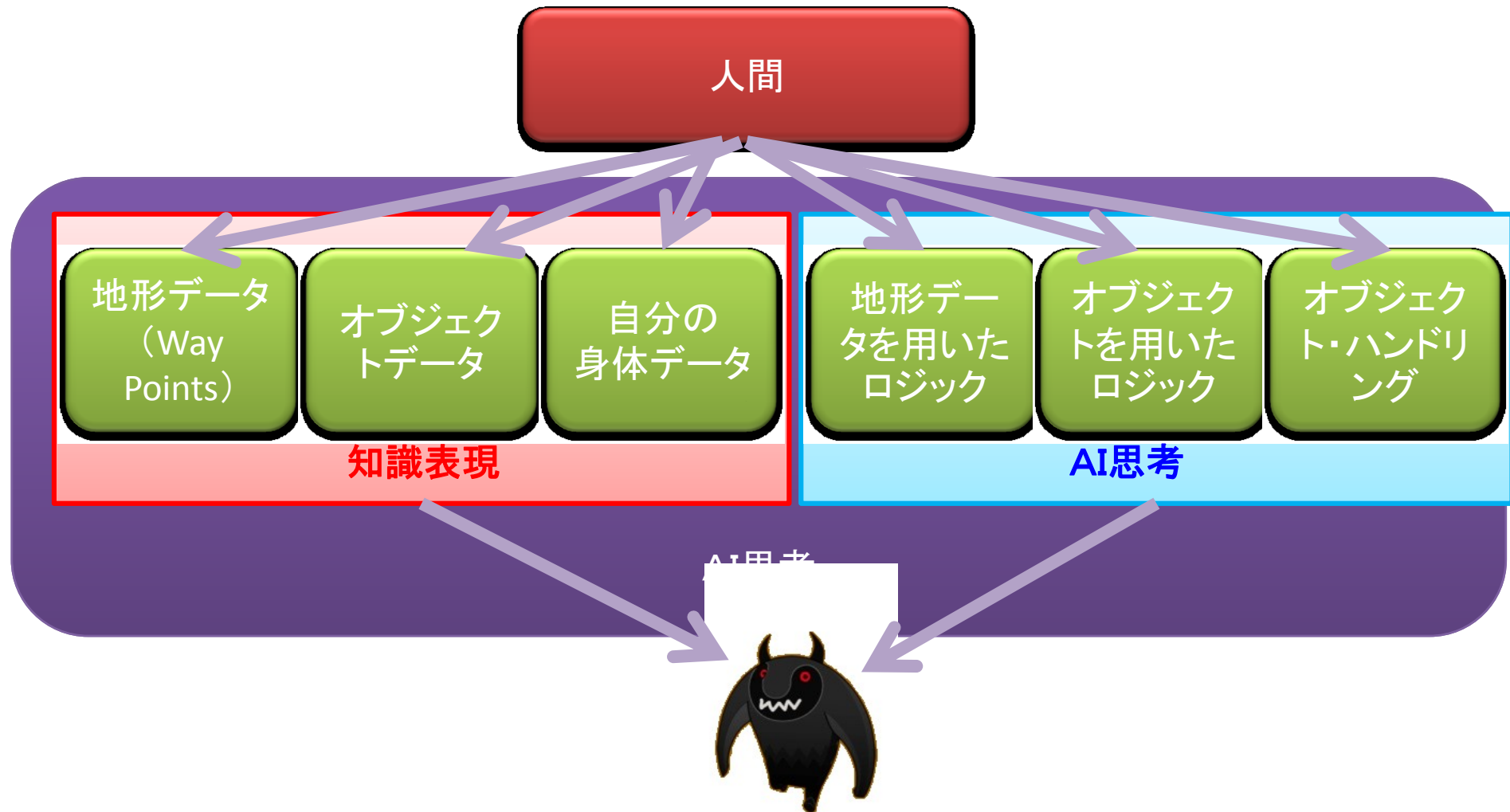
AIは直接、事物（物・事）のデータ表現を通じて世界にアクセスする。

知識表現とは



良い知識表現を作成することが、AIの良い理解を形成する。

AIを作る＝AIに知識と思考を与えること。



AIにとって世界とは？

世界（事物そのもの）

地形データ
(Way
Points)

オブジェク
トデータ

自分の
身体データ

知識表現



AIにとって世界とは？

世界(事物そのもの)

地形データ
(Way
Points)

オブジェク
トデータ

自分の
身体データ

知識表現



AIにとって世界とは？

世界（事物そのもの）

AIは知識表現を通じて世界を観る。＝ 知識表現のインスタンスたちが世界そのもの。

地形データ
(Way
Points)

オブジェク
トデータ

自分の
身体データ

知識表現



AIにとって世界とは？

世界（事物そのもの）

世界において事物は関係している。
その関係性も知識表現の中に含まれていなければならない。

地形データ
(Way
Points)

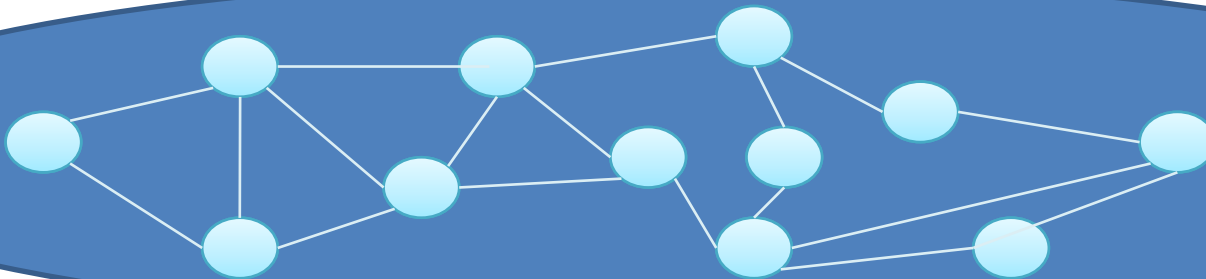
オブジェク
トデータ

自分の
身体データ

知識表現



AIにとって世界とは？



世界において事物は関係している。
その関係性も知識表現の中に含まれていなければならない。

地形データ
(Way
Points)

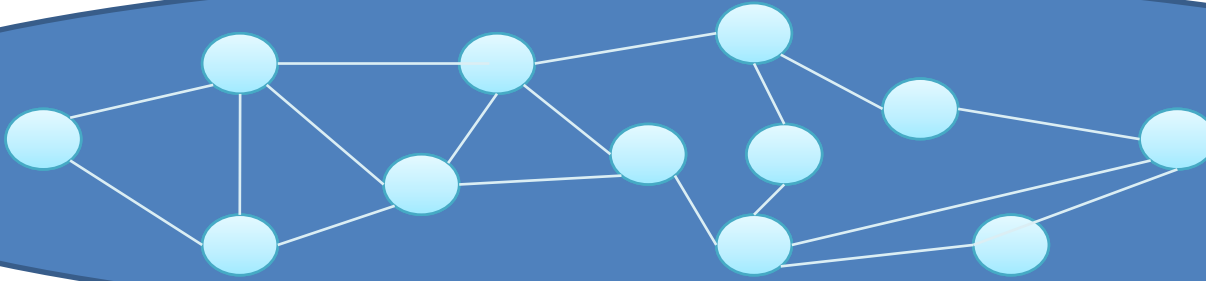
オブジェク
トデータ

自分の
身体データ

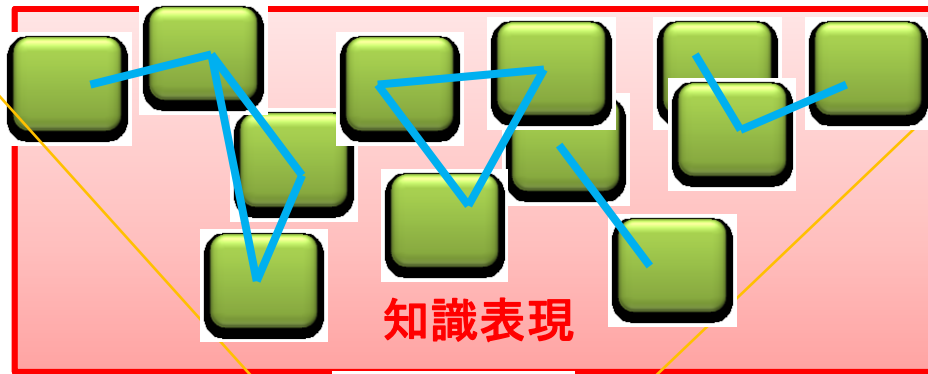
知識表現



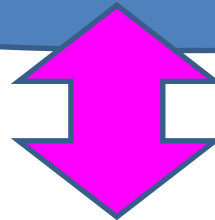
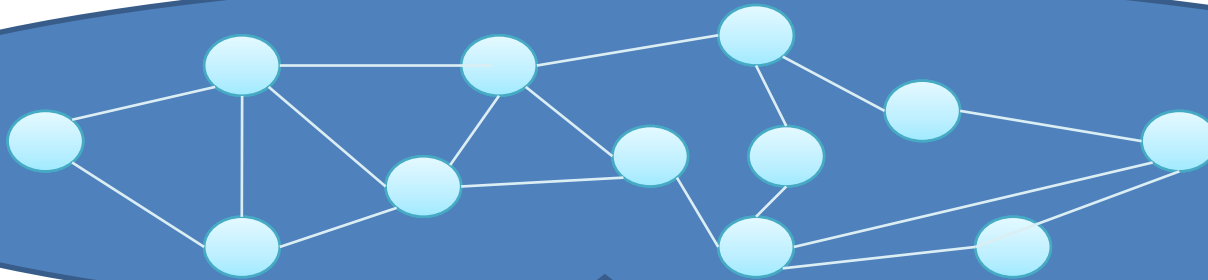
AIにとって世界とは？



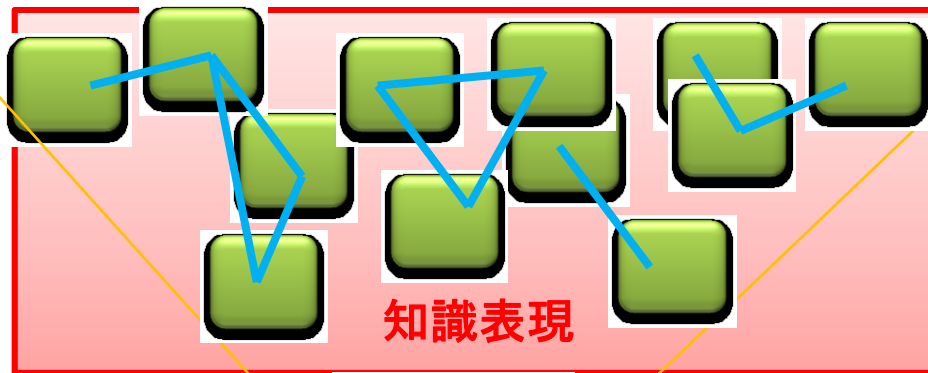
世界において事物は関係している。
その関係性も知識表現の中に含まれていなければならない。



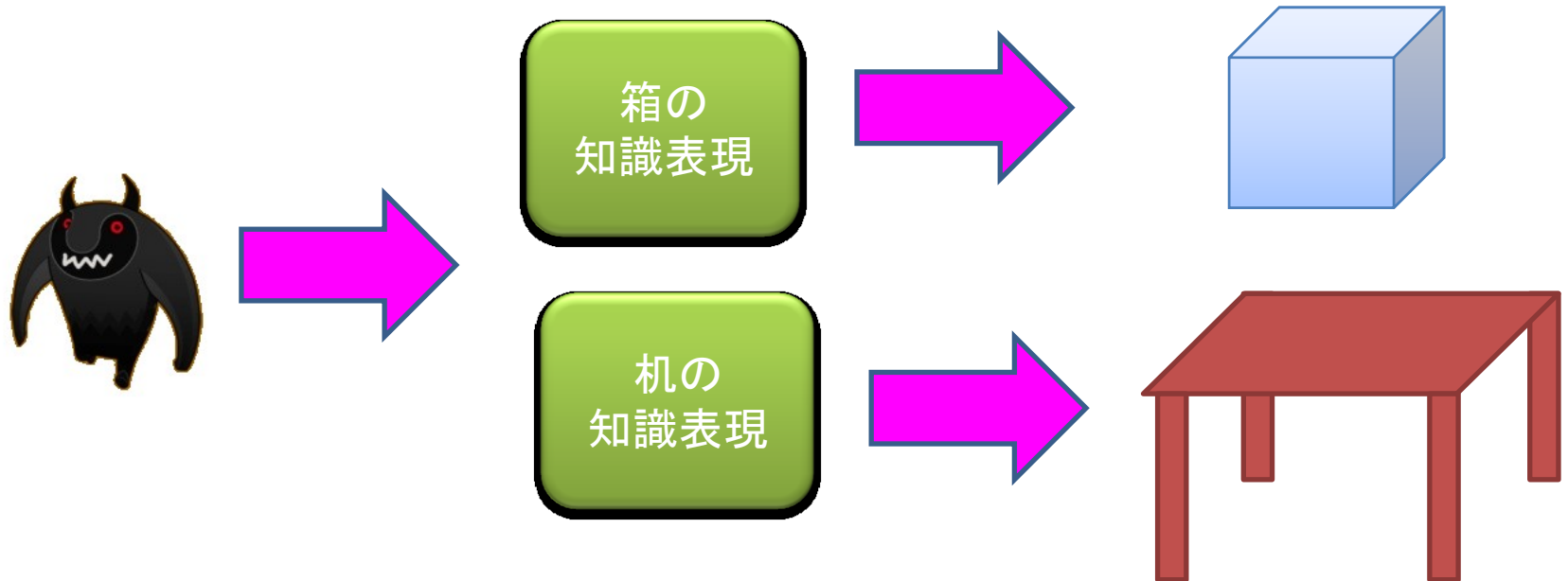
AIにとって世界とは？



関係性の比喻

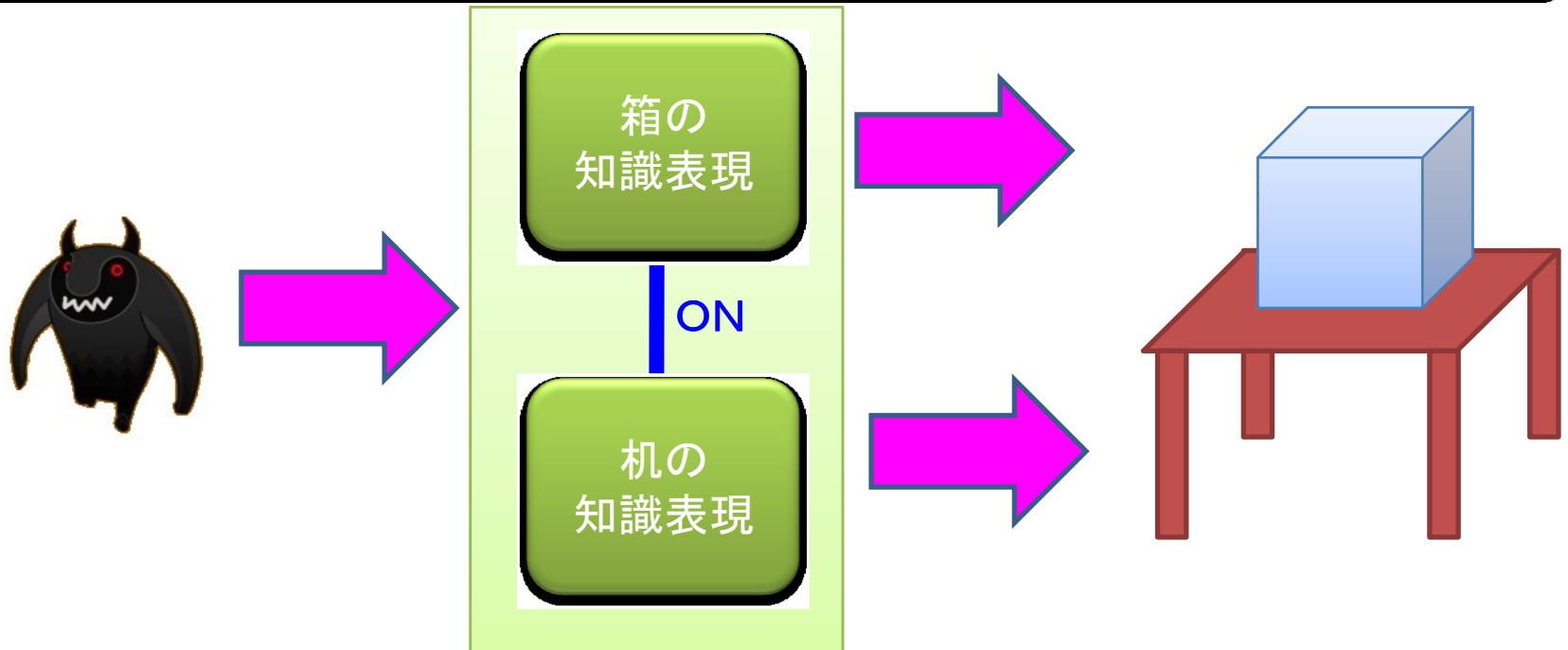


知識表現とは



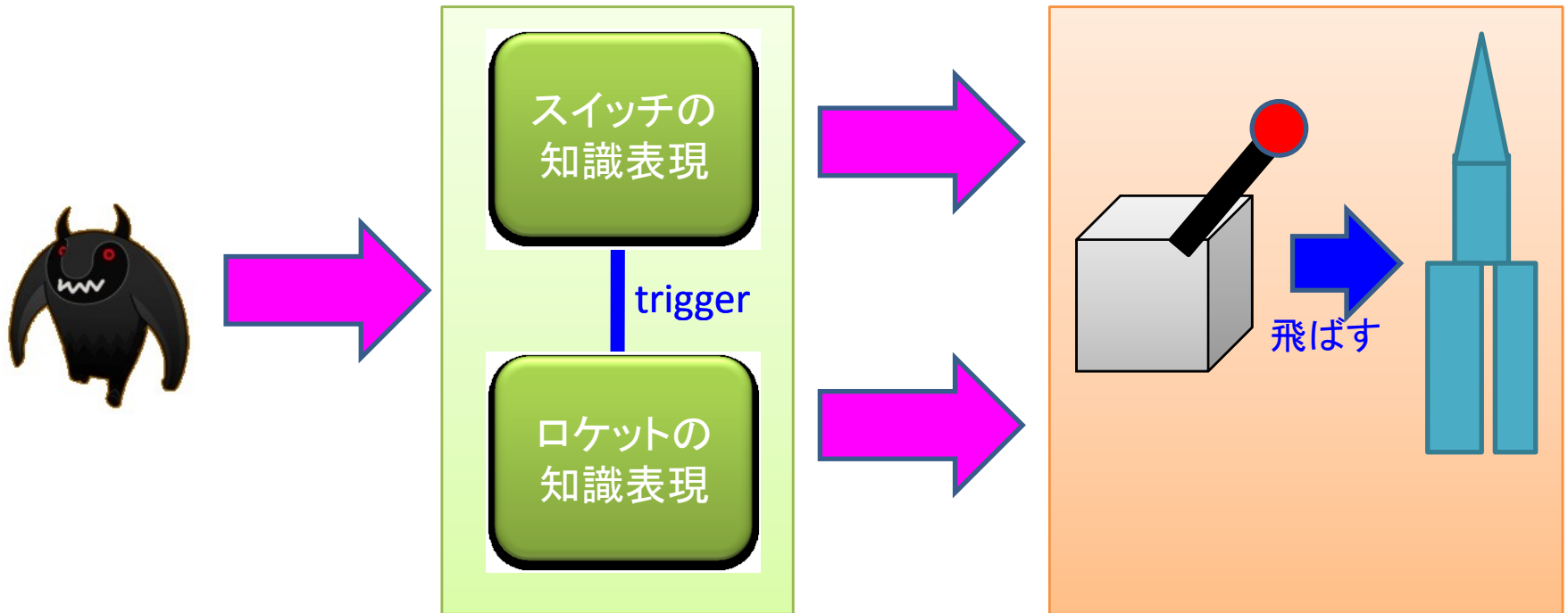
AIは直接、事物(物・事)のデータ表現を通じて世界にアクセスする。

関係性表現とは



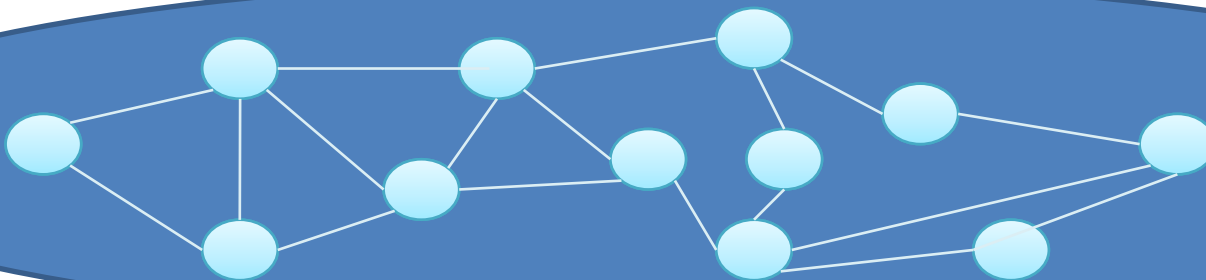
事物の関係性は知識表現の関係性にも反映されねばならない。(＝関係性表現)

関係性表現とは

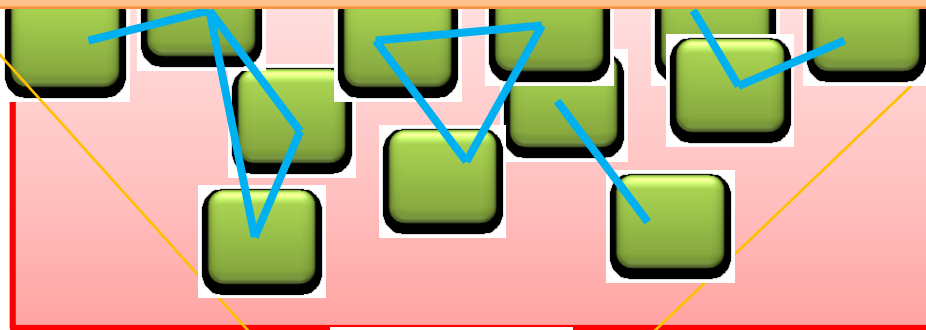


事物の関係性は知識表現の関係性にも反映されねばならない。(＝関係性表現)

知識表現のネットワーク



AIは知識表現のネットワークで世界を理解する。
ゲームAIを作るときは、まず前提として、
知識表現のネットワークを作る必要がある。



知識表現のネットワーク

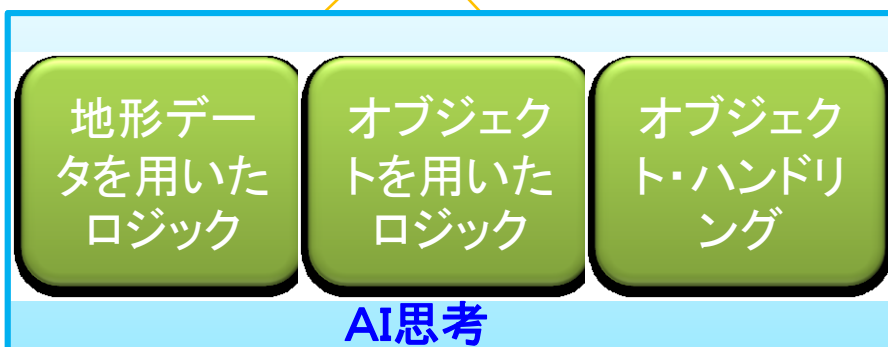
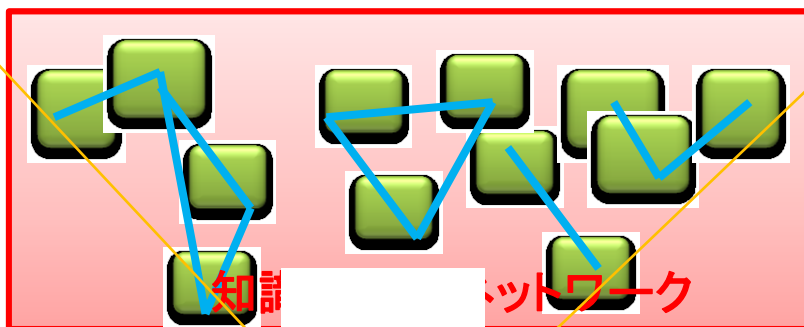


研究テーマ：知識表現

- (1) **新しい知識表現の探求** どのような知識表現、知識表現ネットワークを作れば、AIに世界を高度に認識させることができるだろうか？
- (2) **知識表現を作るための技術** ゲーム内にあるオブジェクト、地形、それらの関係性、膨大な知識表現のデータを自動的に作ることができるだろうか？

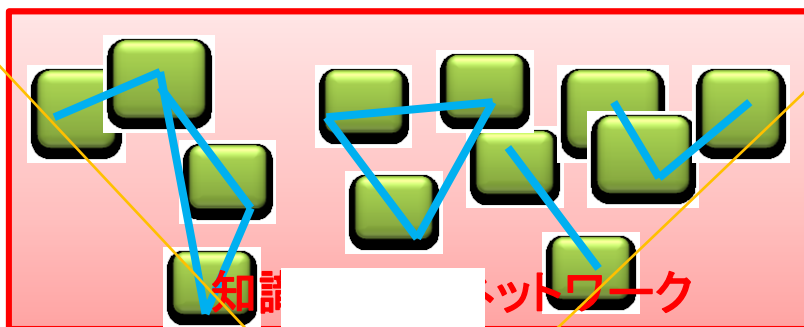
知識表現のネットワーク

世界(事物そのもの)



知識表現のネットワーク

世界(事物そのもの)

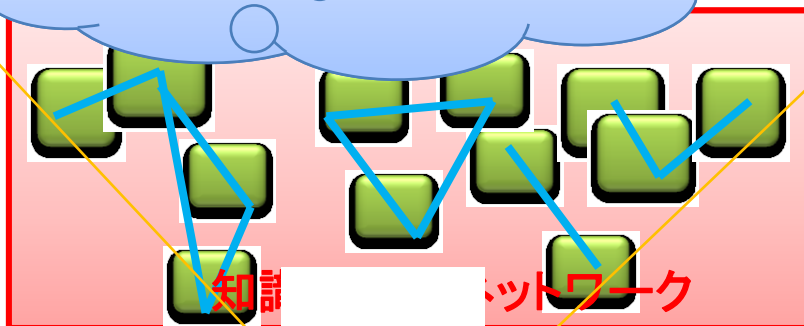


認識・思考・意思決定

知識表現のネットワーク

世界(事物そのもの)

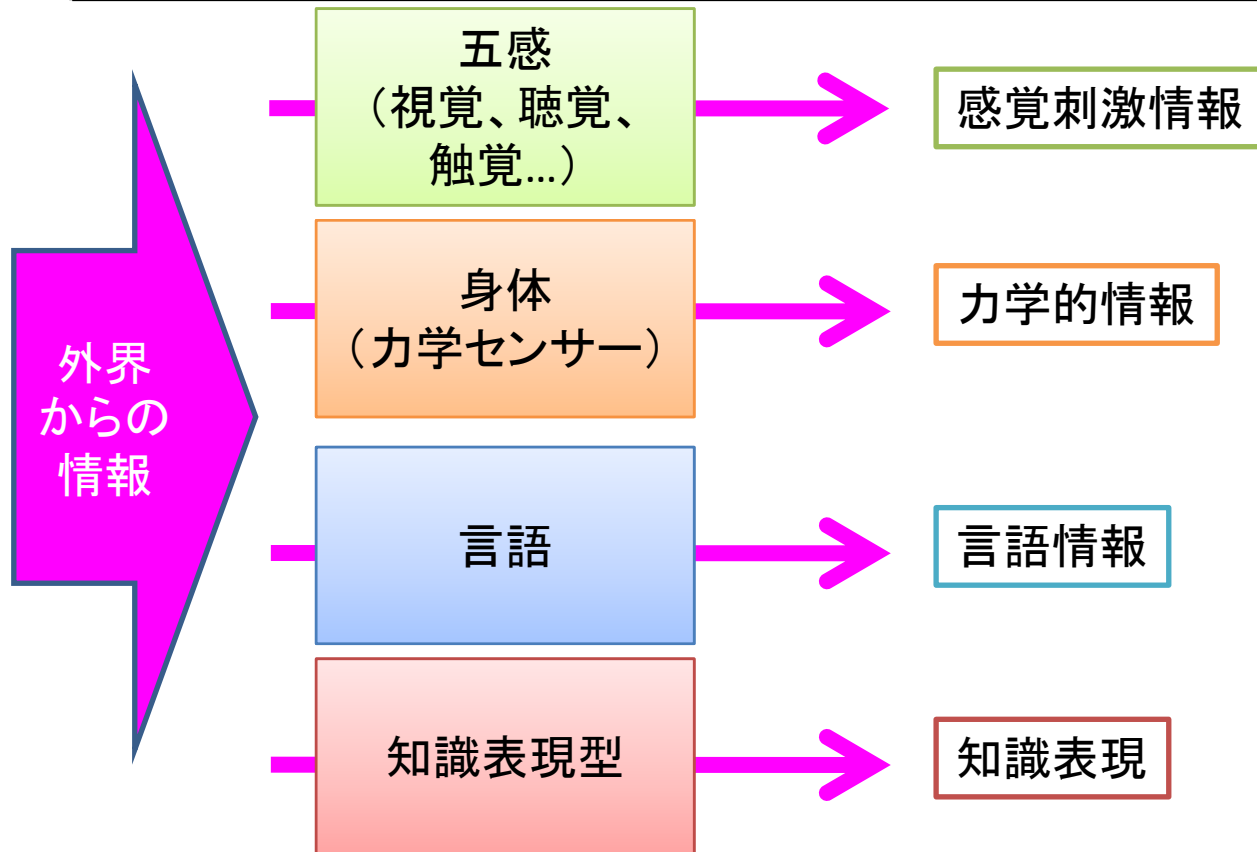
センサー群



思考・意思決定

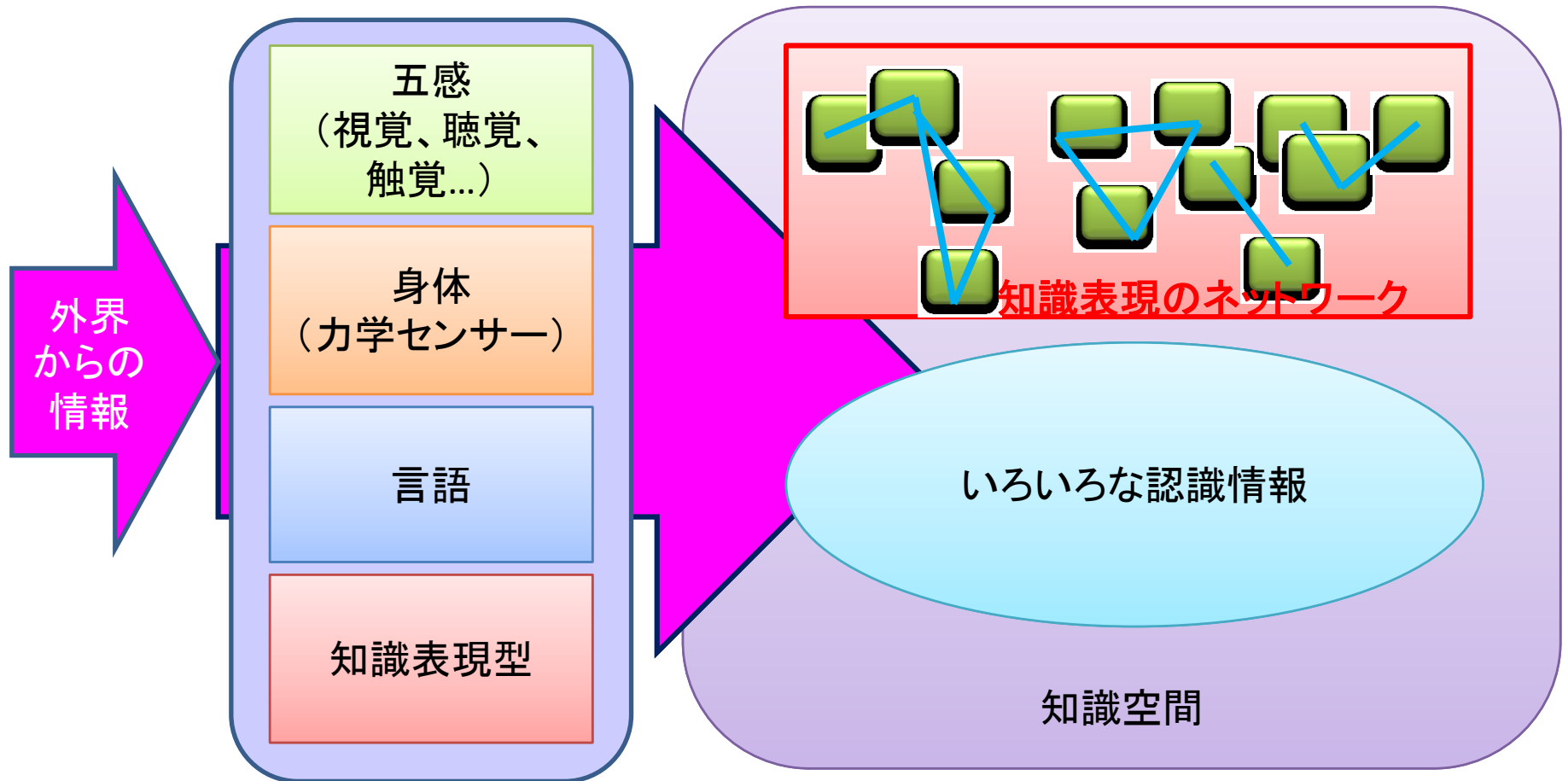
センサー群と知識表現

- (1) センサーとは世界から情報を得る窓口。たとえば、五感（視覚、聴覚....）。
- (2) しかし、それだけではなく、身体そのものも（物理的な力に対する）センサーである。
- (3) センサーの拡張として言語そのものもセンサーたり得る。
- (4) 知識表現そのものも、また高度なセンサーの型として機能する。



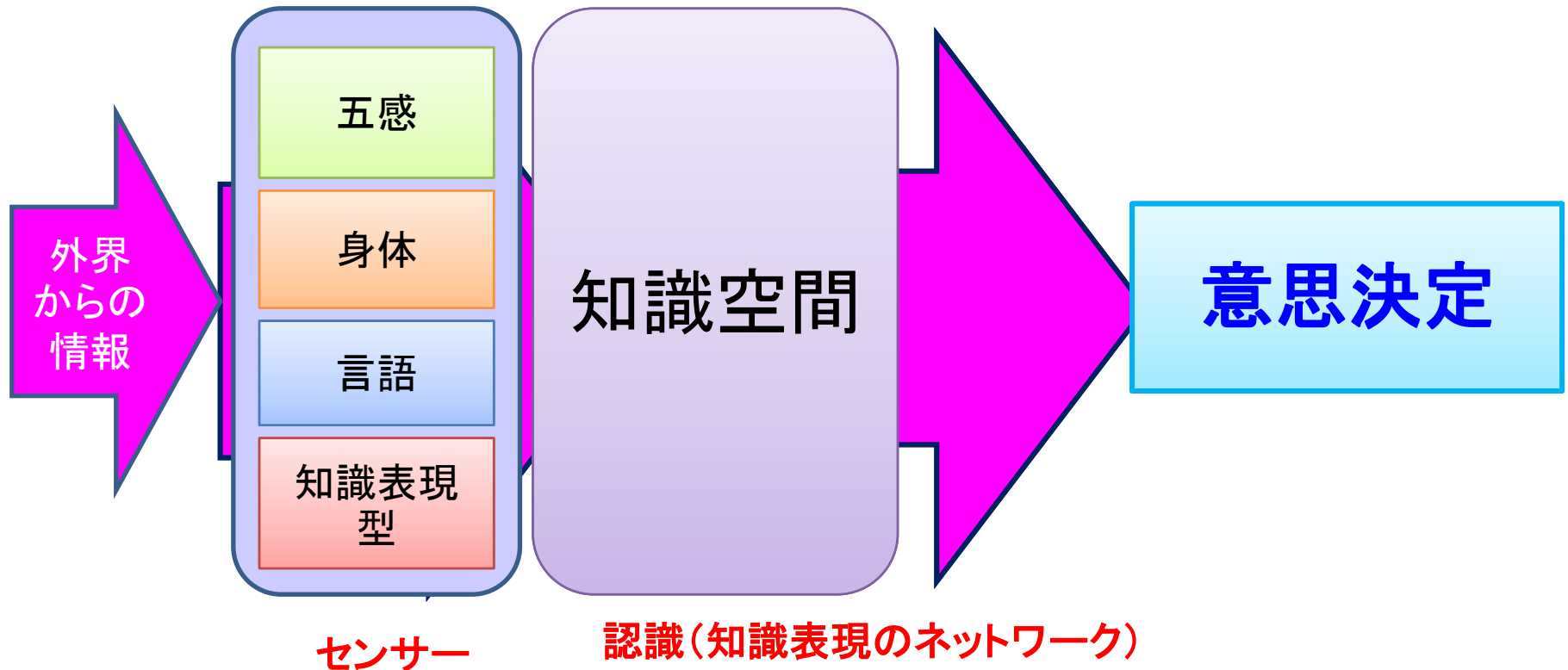
センサー群と知識表現

- (1) センサーとは世界から情報を得る窓口。たとえば、五感（視覚、聴覚...）。
- (2) しかし、それだけではなく、身体そのものも（物理的な力に対する）センサーである。
- (3) センサーの拡張として言語そのものもセンサーたり得る。
- (4) 知識表現そのものも、また高度なセンサーの型として機能する。



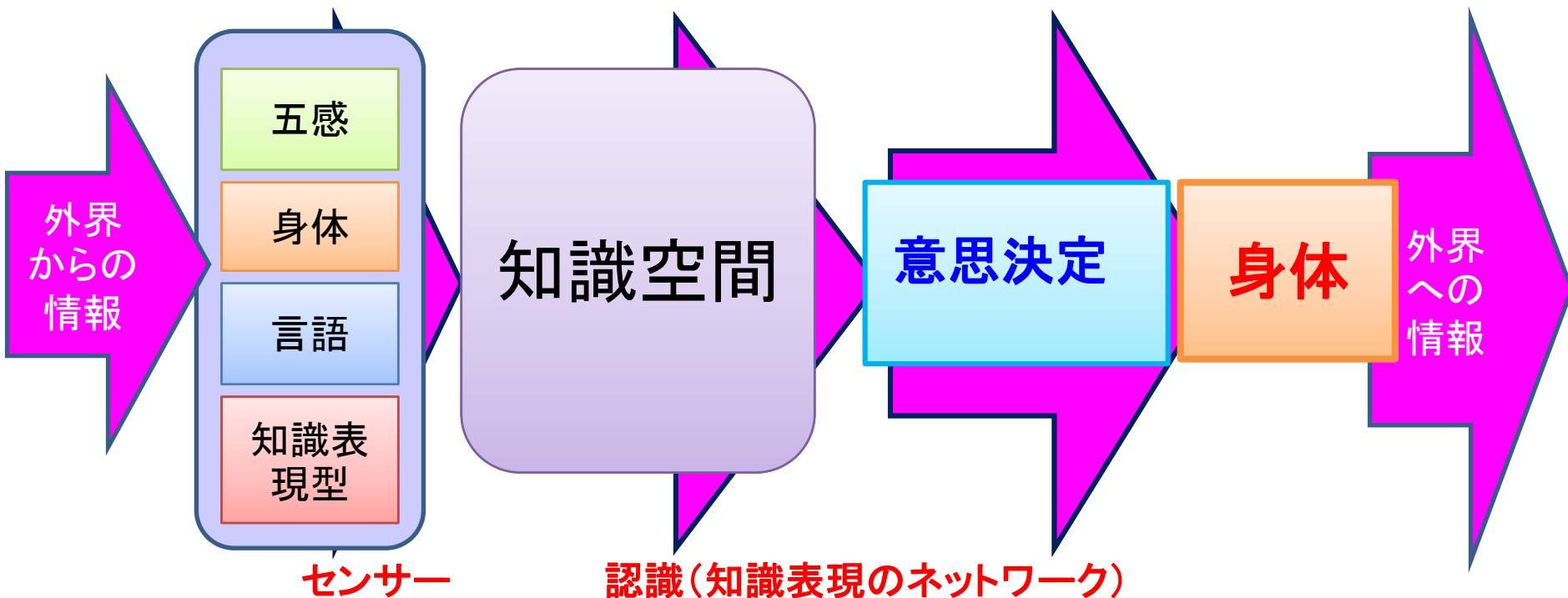
センサー群と知識表現

- (1) センサーとは世界から情報を得る窓口。たとえば、五感（視覚、聴覚....）。
- (2) しかし、それだけではなく、身体そのものも（物理的な力に対する）センサーである。
- (3) センサーの拡張として言語そのものもセンサーたり得る。
- (4) 知識表現そのものも、また高度なセンサーの型として機能する。



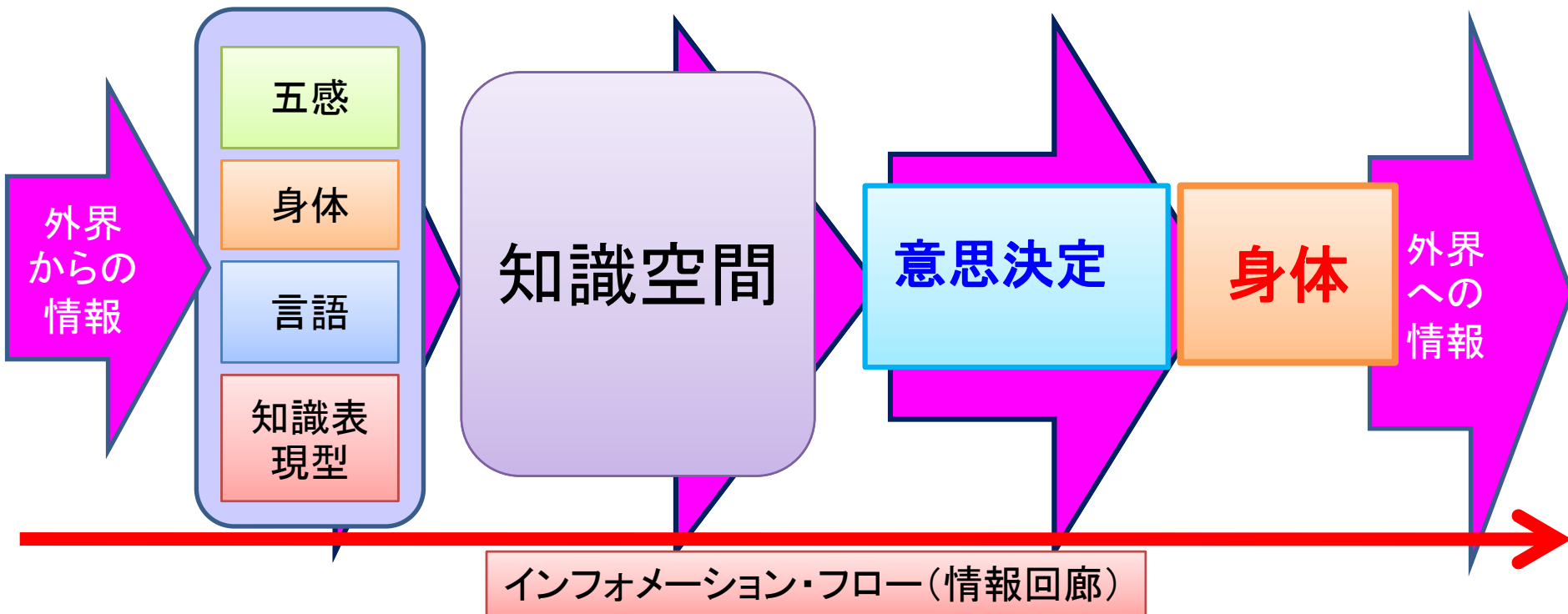
センサー群と知識表現

- (1) センサーとは世界から情報を得る窓口。たとえば、五感（視覚、聴覚....）。
- (2) しかし、それだけではなく、身体そのものも（物理的な力に対する）センサーである。
- (3) センサーの拡張として言語そのものもセンサーたり得る。
- (4) 知識表現そのものも、また高度なセンサーの型として機能する。



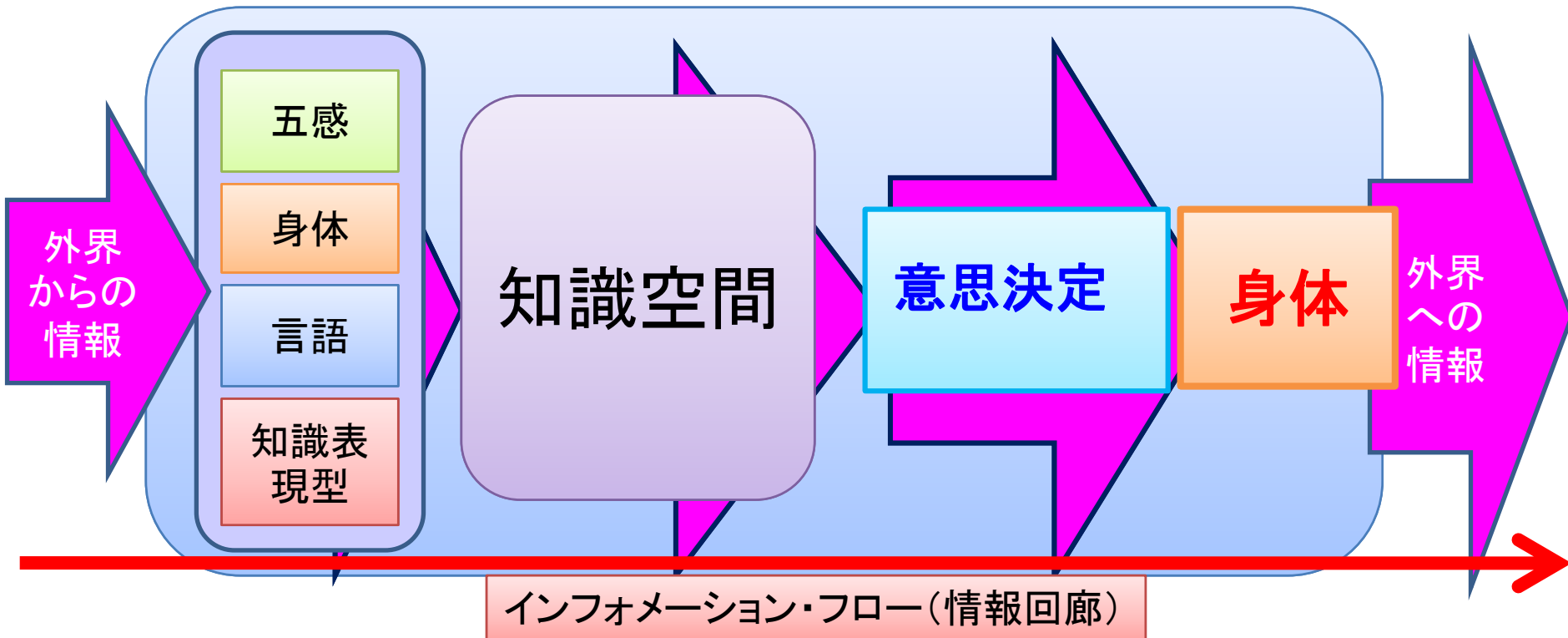
センサー群と知識表現

- (1) センサーとは世界から情報を得る窓口。たとえば、五感（視覚、聴覚....）。
- (2) しかし、それだけではなく、身体そのものも（物理的な力に対する）センサーである。
- (3) センサーの拡張として言語そのものもセンサーたり得る。
- (4) 知識表現そのものも、また高度なセンサーの型として機能する。



センサー群と知識表現

- (1) センサーとは世界から情報を得る窓口。たとえば、五感（視覚、聴覚....）。
- (2) しかし、それだけではなく、身体そのものも（物理的な力に対する）センサーである。
- (3) センサーの拡張として言語そのものもセンサーたり得る。
- (4) 知識表現そのものも、また高度なセンサーの型として機能する。



エージェント・アーキテクチャ

人工知能

センサー

エフェクタ

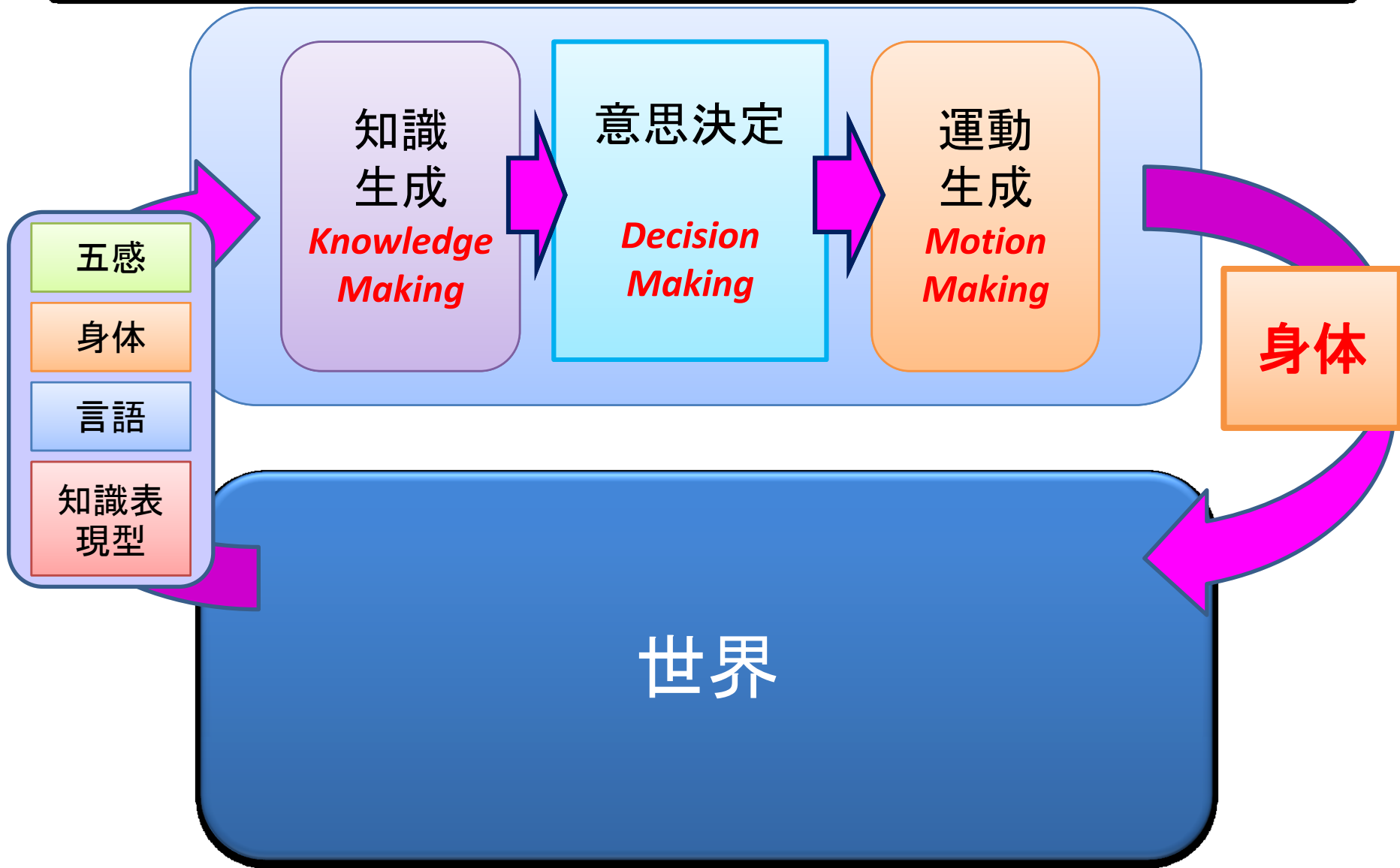
世界



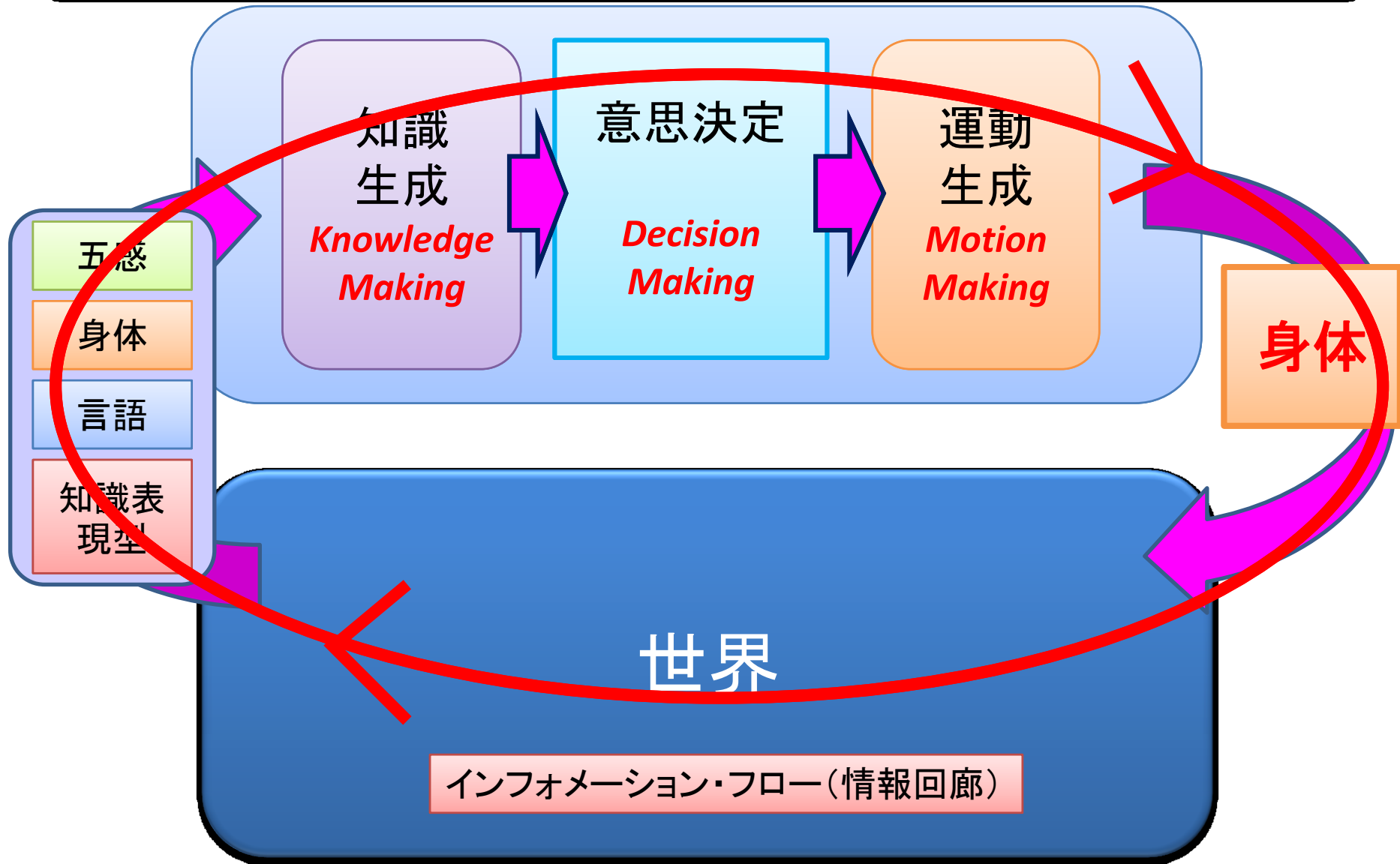
エージェント・アーキテクチャ



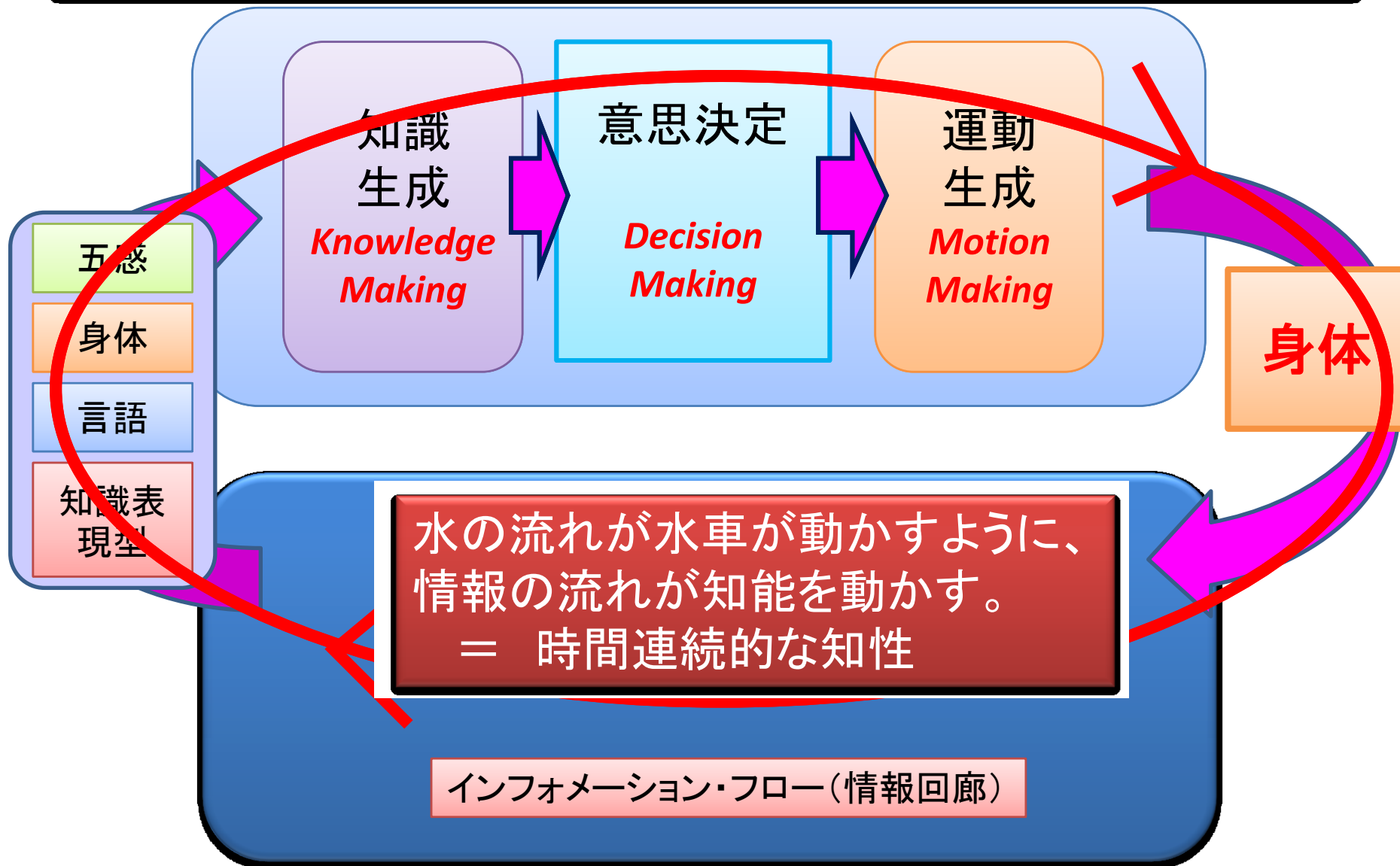
エージェント・アーキテクチャ



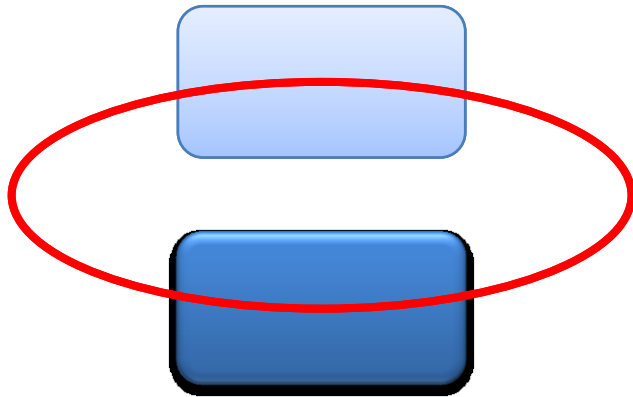
エージェント・アーキテクチャ



エージェント・アーキテクチャ



アカデミックなAIとの違い



リアルタイムAI
連続時間、連続空間



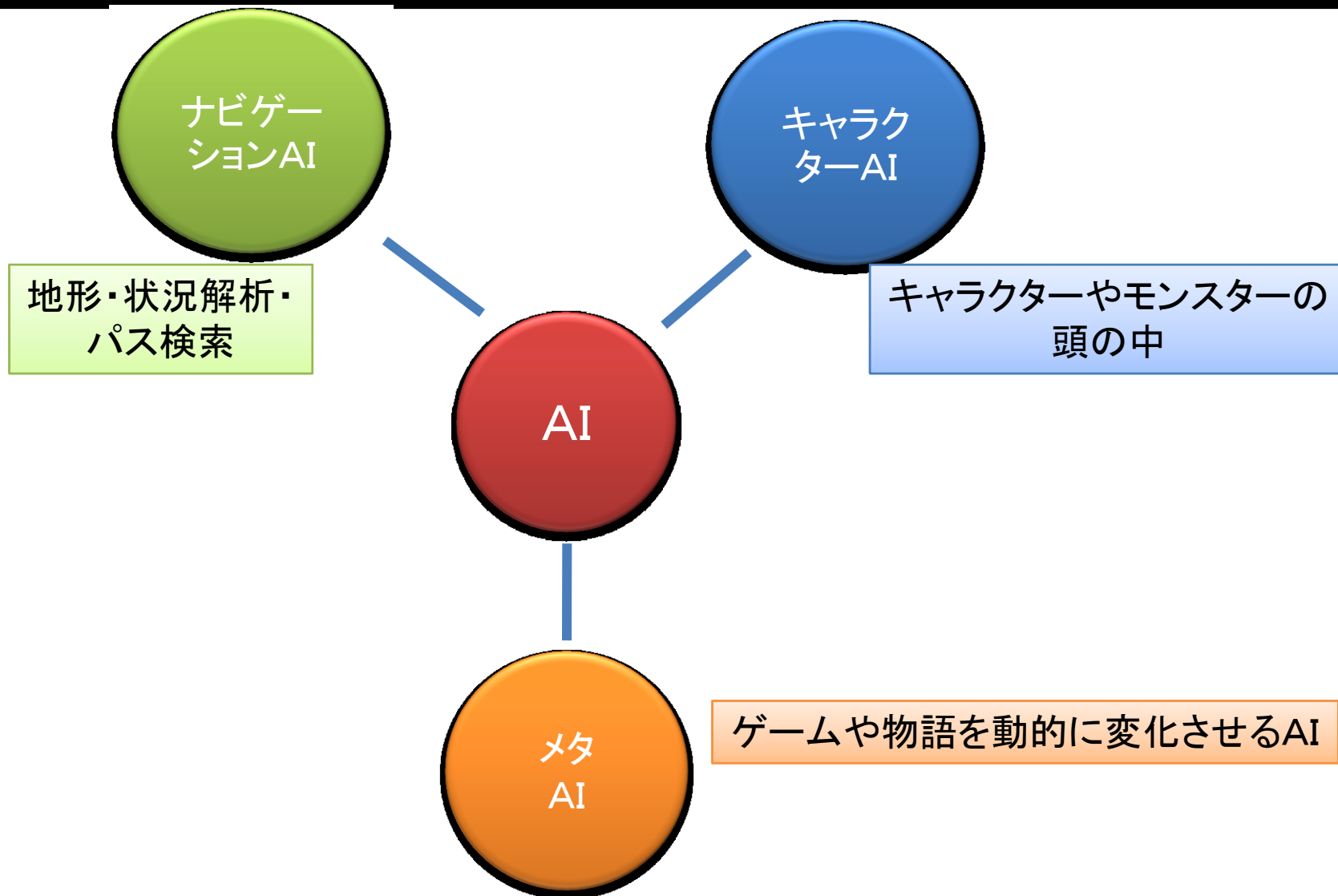
<http://game.e-frontier.co.jp/ai/shogi16/>

ターンベースAI
離散時間、離散空間

将棋(テーブルゲーム)AIと アクションゲームAIの違い

	アクションゲーム	将棋
ゲーム	開発当初はゲームがない (ゲームと一緒にAIを作る =AIはゲームの一部)	既にある。 (AIはゲームの外にある。 将棋の中にAIは含まれない)
ステージ(空間)	三次元地形・連続	盤の目・離散的
時間	連続時間(リアルタイム)	ターン
登場人物	キャラクター/モンスター	駒
AI	キャラクターのブレイン/ ゲーム全体の操作するAI(メタAI)	将棋の差し手としてのAI
ゲーム表現	?	ゲームツリー
状況	連続的变化	離散的变化
目的	楽しませる。ステージを成立させる。	勝利
ゲームのつながり	一定時間、一定区間で区切れる。ランダムな要素も。(1~5秒間の判断)	厳密にすべての手がつながっていない(ツリー検索が有効)
何を作るか?	ユーザーの主観的体験(UX) AIそのものが目的ではない。	賢いAI, 面白いAI

AIの定義



研究テーマ:アーキテクチャ

- (1) **アーキテクチャの探求** どのようなエージェント・アーキテクチャの仕組みが、より高度なより人間的なAIを実現できるか？ゲームの中で色々な仕組みを実験し検証せよ。
- (2) **インフォメーション・フロー** エージェント・アーキテクチャの中で、どのような情報形態の変化、情報の流れの形成を実現すべきかを探求せよ。いろいろなゲームAIの中のインフォメーション・フローの流れの形態を分類し、研究せよ。

第二部

ナビゲーションAI

ナビゲーションAI

ナビゲーションAIとは、
ゲームの環境全体から必要な情報を提示する
AIのことを言う。

ここで言う「環境」とは、
地形、オブジェクトの配置、敵の配置、勢力図など、
レベル空間にまつわる全ての情報のことです。

※レベル＝ゲームステージのこと。

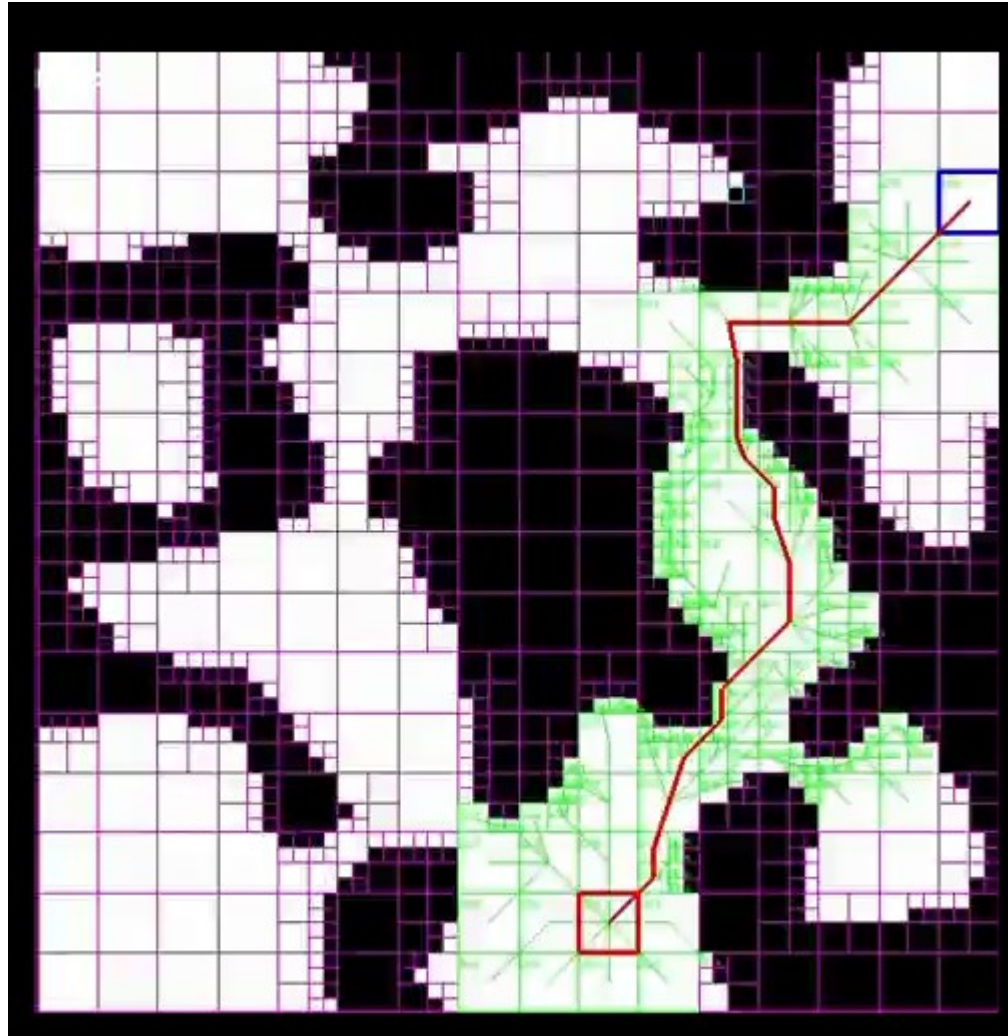
ナビゲーションAIの代表例

- ナビゲーション (Navigation、パス検索)
= 指定したポイントまでのパスを見つける。
- 影響マップ (Influence Map)
= ゲームの全体的な状況をリアルタイムに捉える。
- 戦術ポイント検索システム (Tactical Point System)
= 地形から目的に必要なポイントを見つける。

パス検索

＝ゲーム内で任意の2点間の経路をゲーム進行中に計算する技術

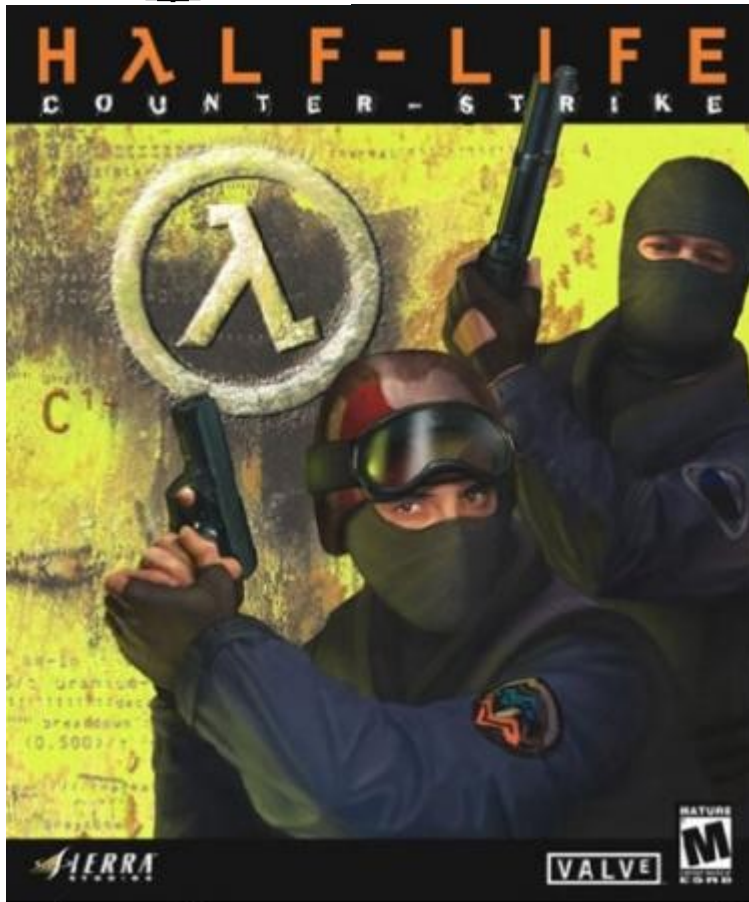
ナビゲー
ションAI



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

Counter Strike (2000)

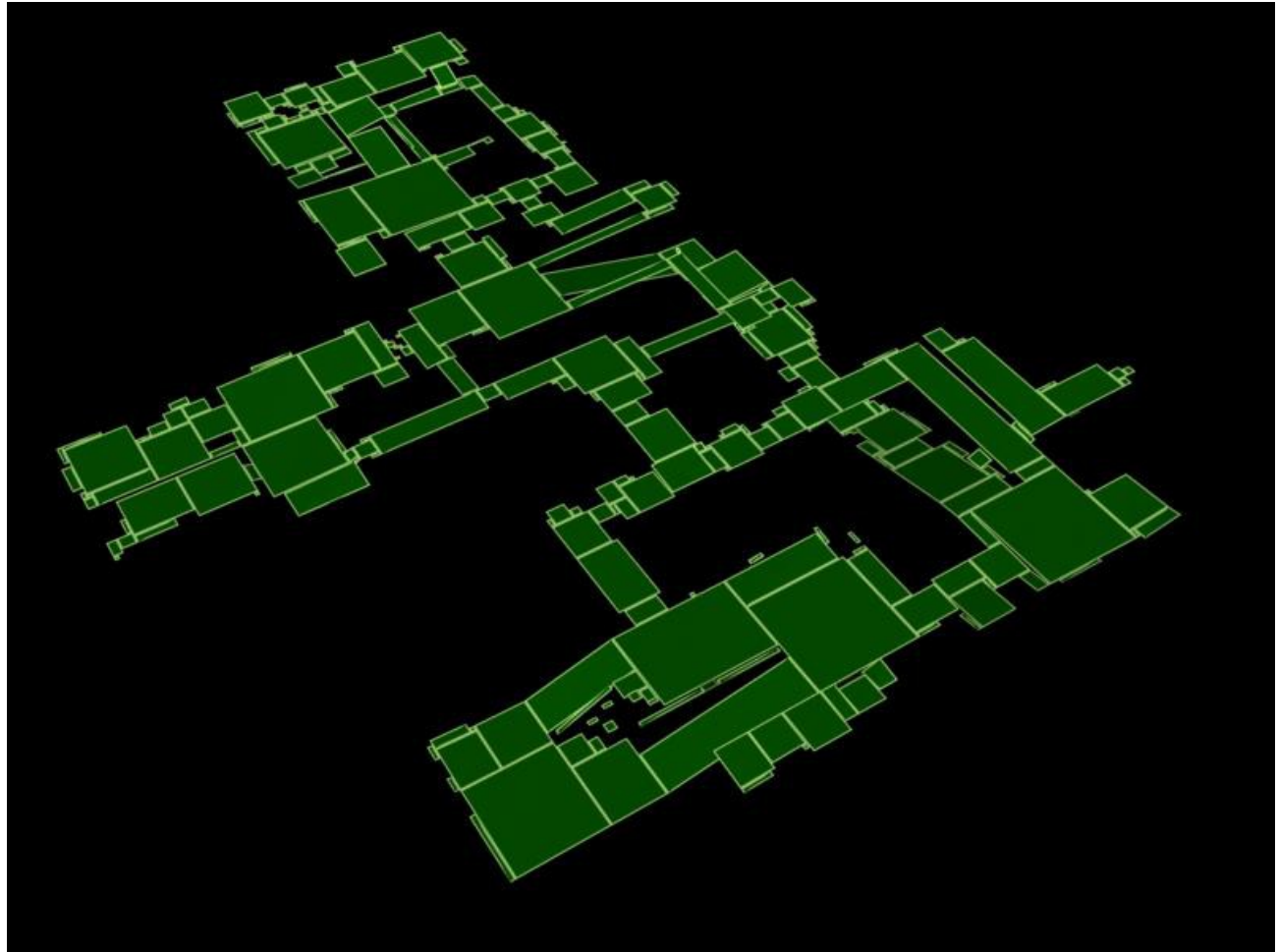
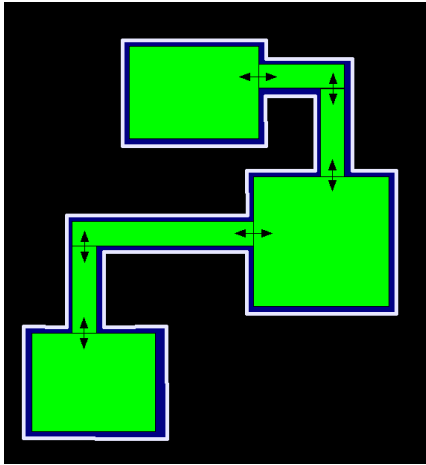
ナビゲー
ションAI



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



レベル内をキャラクターが自由に移動できるようにする



The Official Counter-Strike Bot by Michael Booth

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

ナビゲー
ションAI

Path Following (デモ)



The Official Counter-Strike Bot by Michael Booth

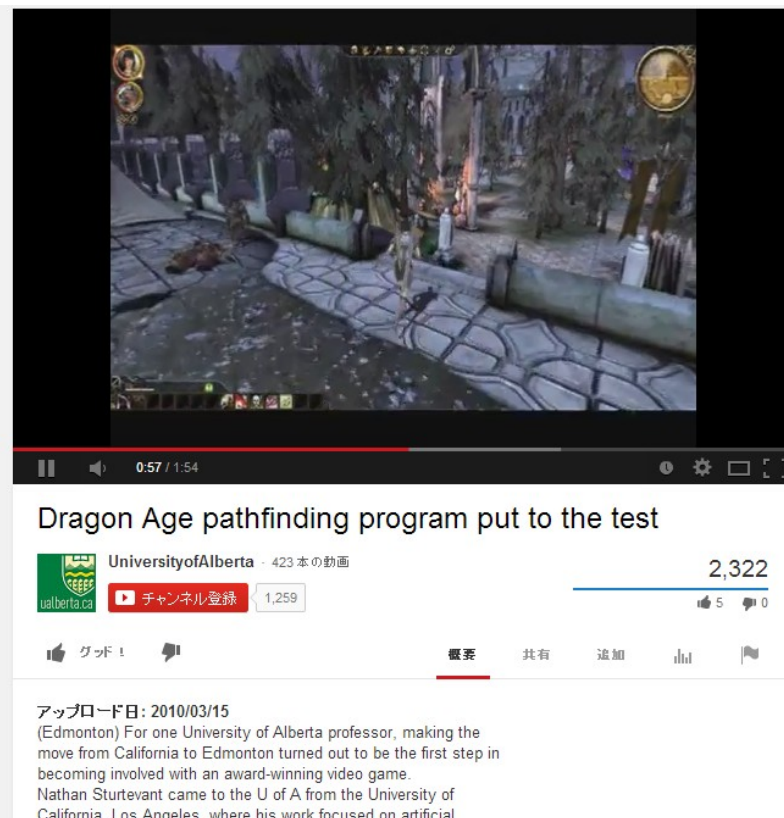
<http://aigamedev.com/insider/presentation/official-counter-strike-bot/>

ナビゲー
ションAI

Dragon Age



<http://shop.newtrino.jp/eshopdo/refer/vid13655.html>



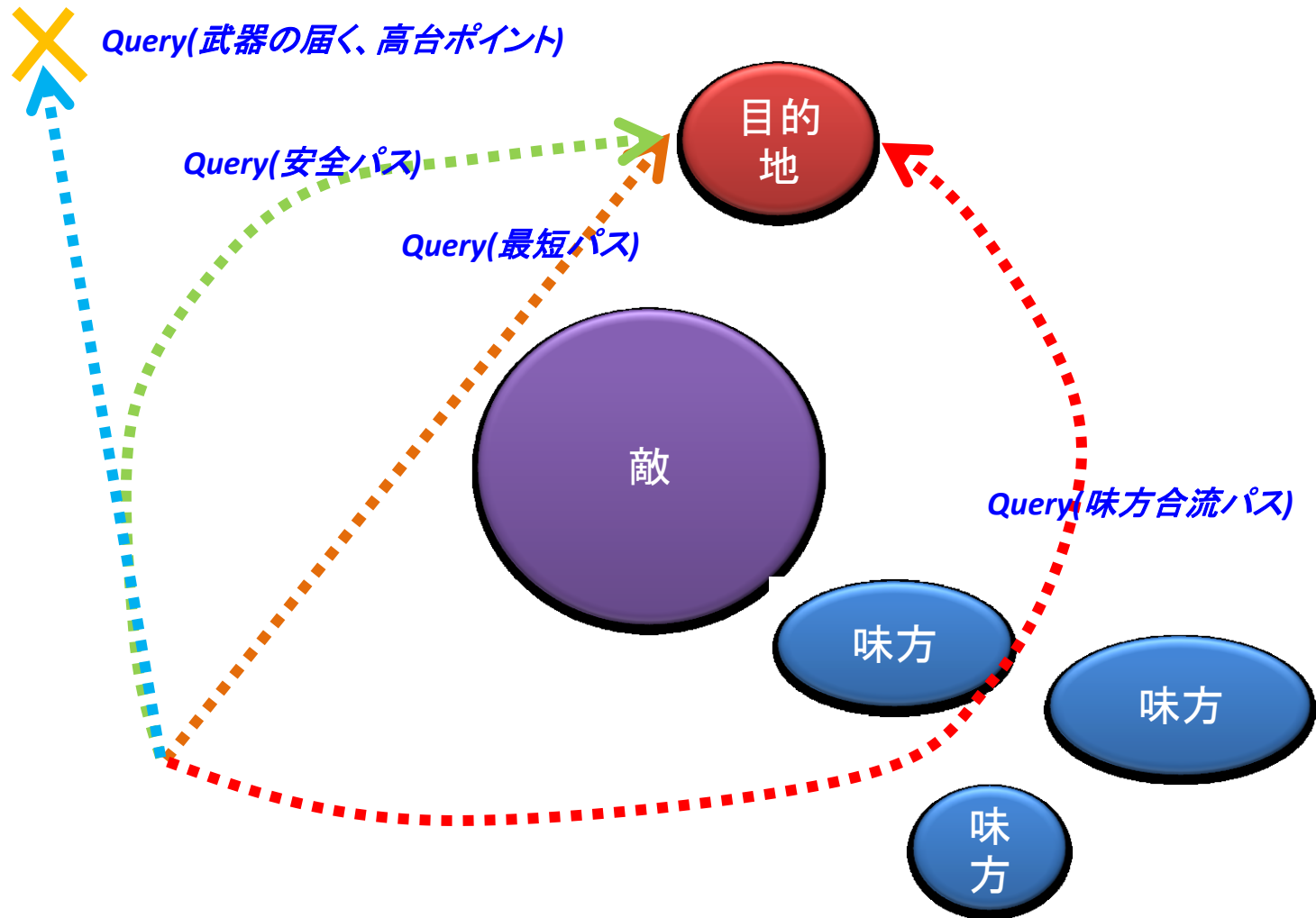
Dragon Age pathfinding program put to the test (University of Alberta)

https://www.youtube.com/watch?v=l7YQ5_Nbifo

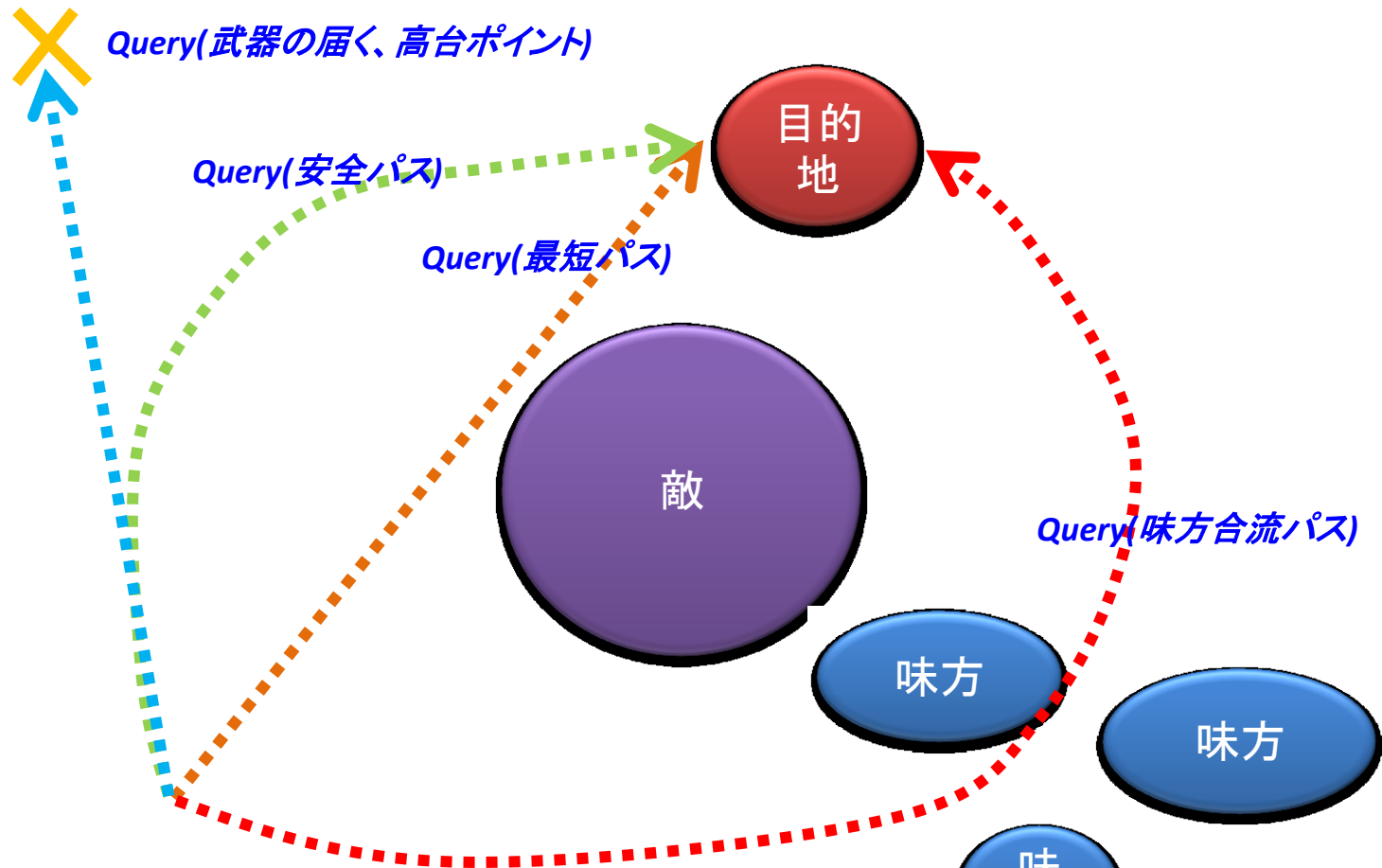
ナビゲーションAI

- ナビゲーション (Navigation、パス検索)
= 指定したポイントまでのパスを見つける。
- 影響マップ (Influence Map)
= ゲームの全体的な状況をリアルタイムに捉える。
- 戦術ポイント検索システム (Tactical Point System)
= 地形から目的に必要なポイントを見つける。

(例) クエリー(Query)システム ＝問い合わせシステム



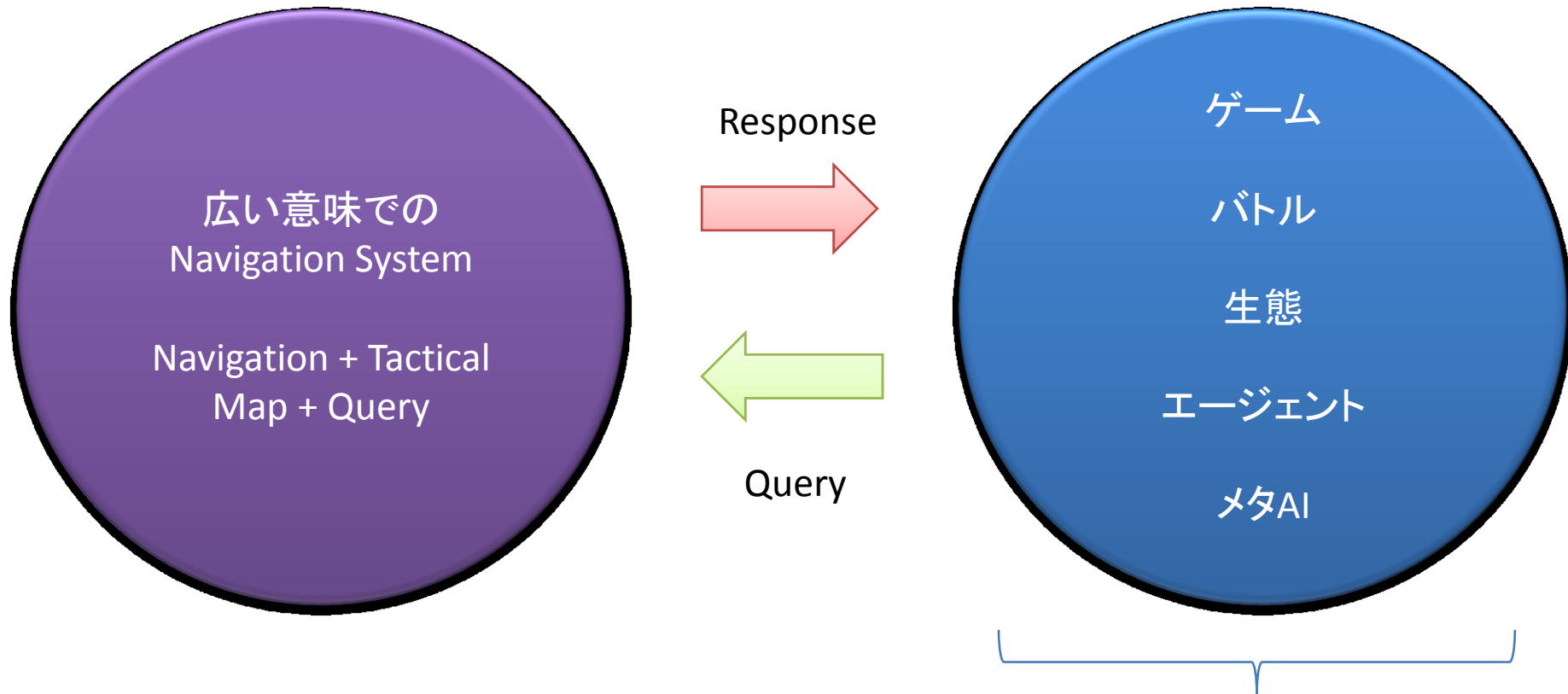
(例)クエリー(Query)システム ＝問い合わせシステム



状況に応じて、必要な経路(パス)、必要なポイントを、
「こういう特徴の経路を探してください」「こういう特徴のポイントを探してください」
という注文(キュー)を投げると、返してくれるシステム。

TPS

= クエリー(問い合わせ)システム

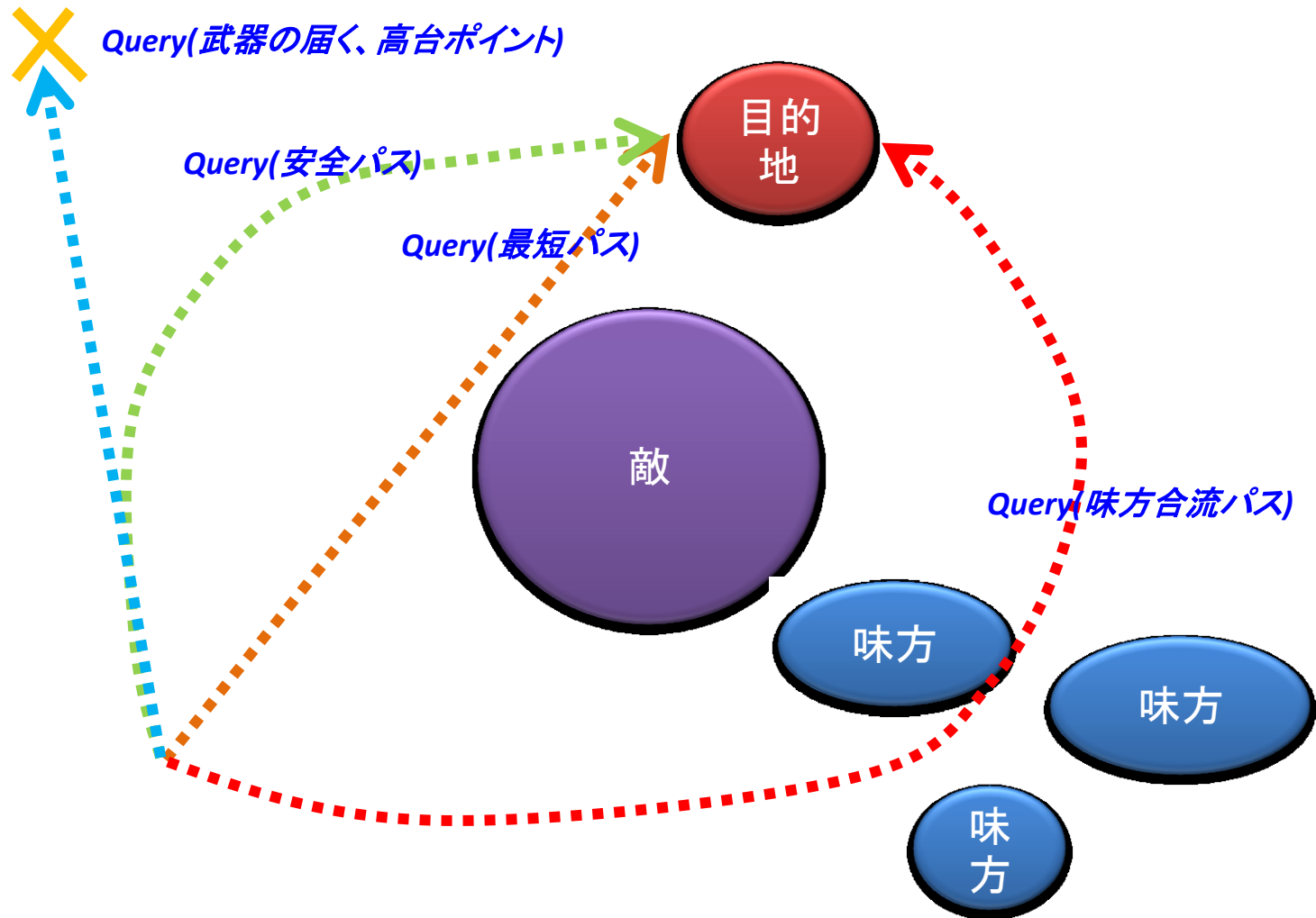


必要な要件を聞き込む
+ 想定する
+ インパクトのある応用例を考える。

(例)TPSクエリー(Query)システム ＝問い合わせシステム

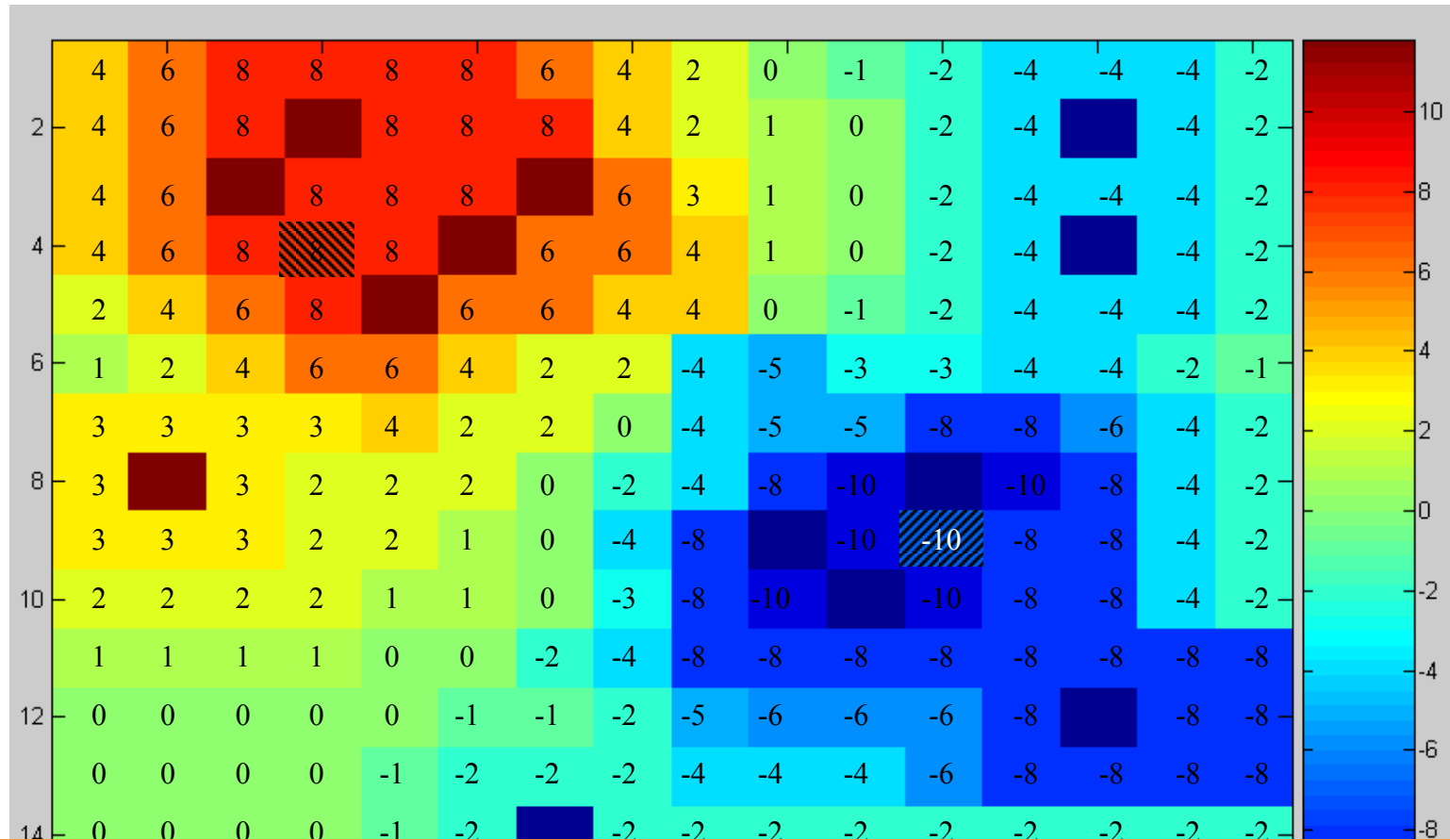
- Query: 自分の今のポイントから座標(10,20,5)までの経路を検索せよ。
(通常の経路検索)
- Response: $\rightarrow (1,19,2) \rightarrow (3,22,15) \rightarrow \dots \rightarrow (10,20,5)$
- Query: 自分の今のポイントから座標(30,5,72)までの安全な経路を検索せよ。(条件付き検索)
- Response: $\rightarrow (4,1,3) \rightarrow (6,4,8) \rightarrow \dots \rightarrow (30,5,72)$
- Query: 自分の半径20m以内で、敵の身長より高くて、射線が通っているポイントを検索せよ。(戦術ポイント検索)
- Response: 座標(15,20,17) (8,2,40)... など。
- Query: 自分の半径10m以内で敵から見えない場所を検索せよ。
- Response: 座標(11,7,55)。
- Query: プレイヤーの予測経路上の見えない場所で、日陰になっているポイントを検索せよ。(スポーニングポイント検索)
- Response: 座標(66,100,22)。

(例)TPSクエリー(Query)システム ＝問い合わせシステム



Tactical Map の例 (影響マップ)

(例) 敵と自分の勢力をリアルタイムに計算する。

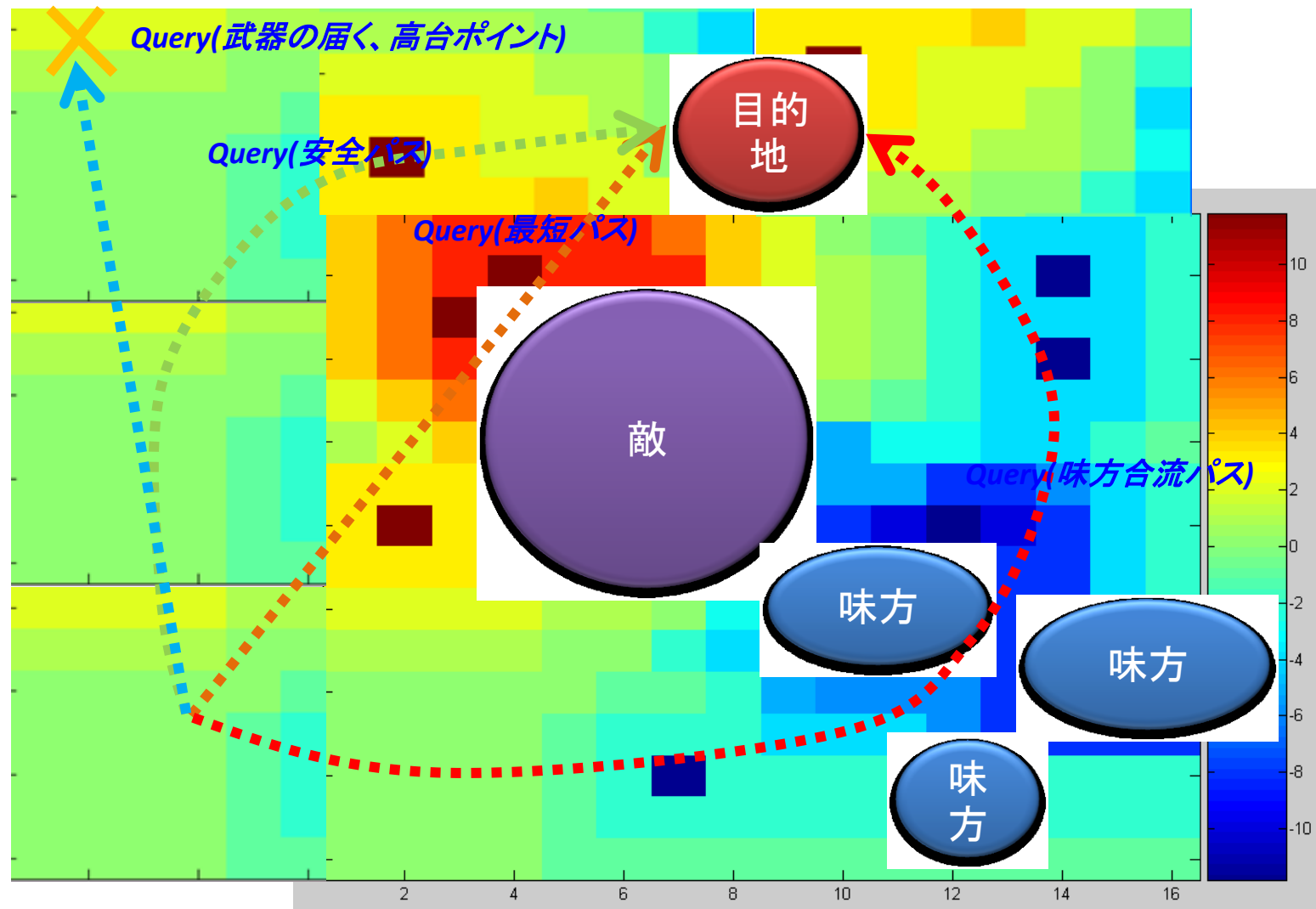


影響マップ＝敵、味方勢力図をマス目ごとの数値として表現したもの。

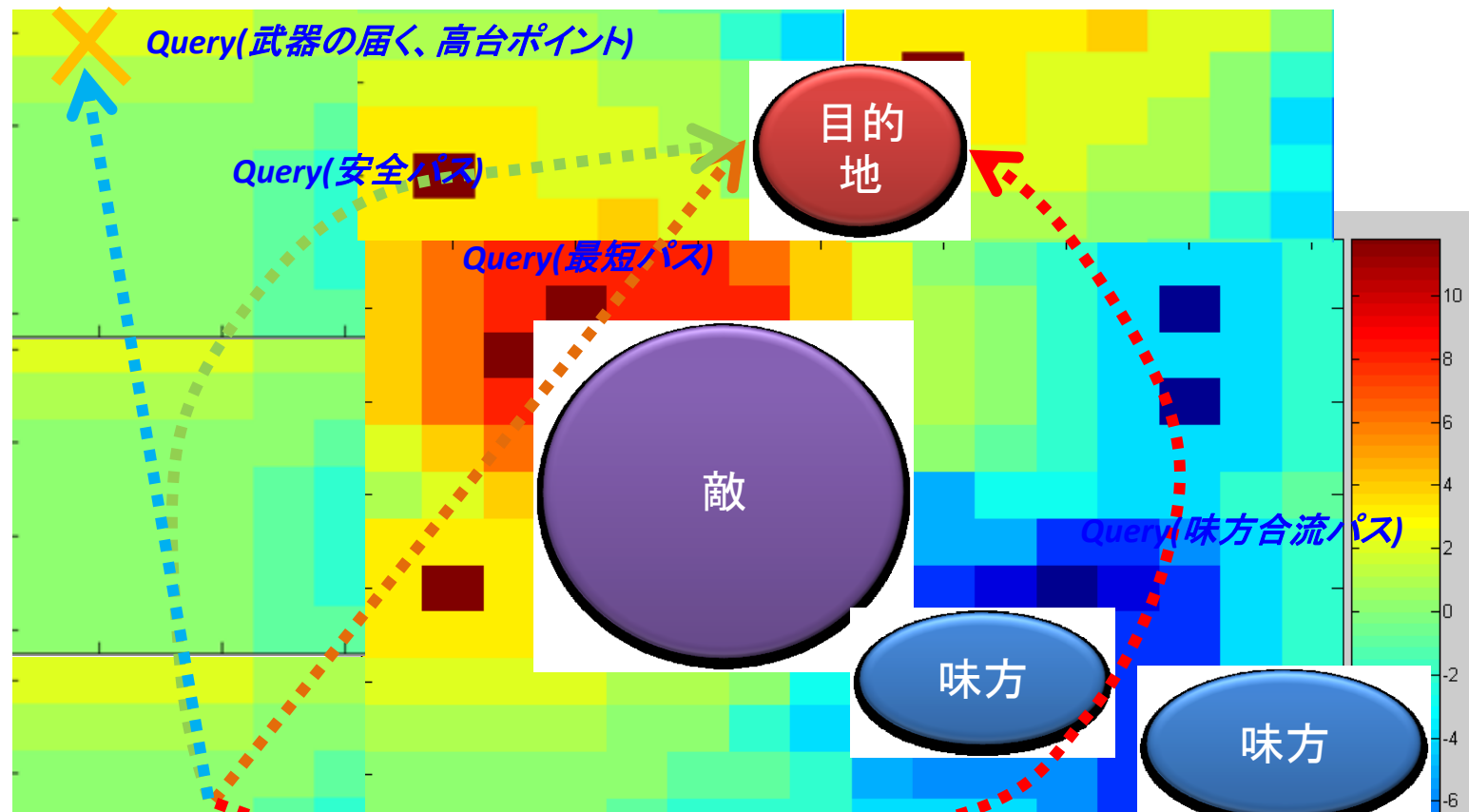
＝赤が敵勢力の影響が高い、青が味方勢力が高い。

＝これをパス検索に加えることで注文に応じたパス検索を行えるようになる。

(例)TPSクエリー(Query)システム ＝問い合わせシステム



(例)クエリー(Query)システム ＝問い合わせシステム



影響マップ＝敵、味方勢力図をマス目ごとの数値として表現したもの。
＝赤が敵勢力の影響が高い、青が味方勢力が高い。
＝これをパス検索に加えることで注文に応じたパス検索を行えるようになる。

TPSクエリーシステム ＝ポイントを動的に配置して評価する



Spaces in the Sandbox: Tactical Awareness in Open World Games (GDC2012) Matthew Jack, Mika Vehkala
<http://www.gdcvault.com/play/1018136/Spaces-in-the-Sandbox-Tactical>

TPSクエリーシステム ＝ポイントを動的に配置して評価する



- ① ポイントをレベル上に生成する。
- ② 各ポイントに目的に応じた評価値を計算する。
- ③ 評価値から目的に応じているか、いないかを分類する。(赤＝駄目、緑＝良い)

TPSクエリーシステム 隠れるポイントを見つける。



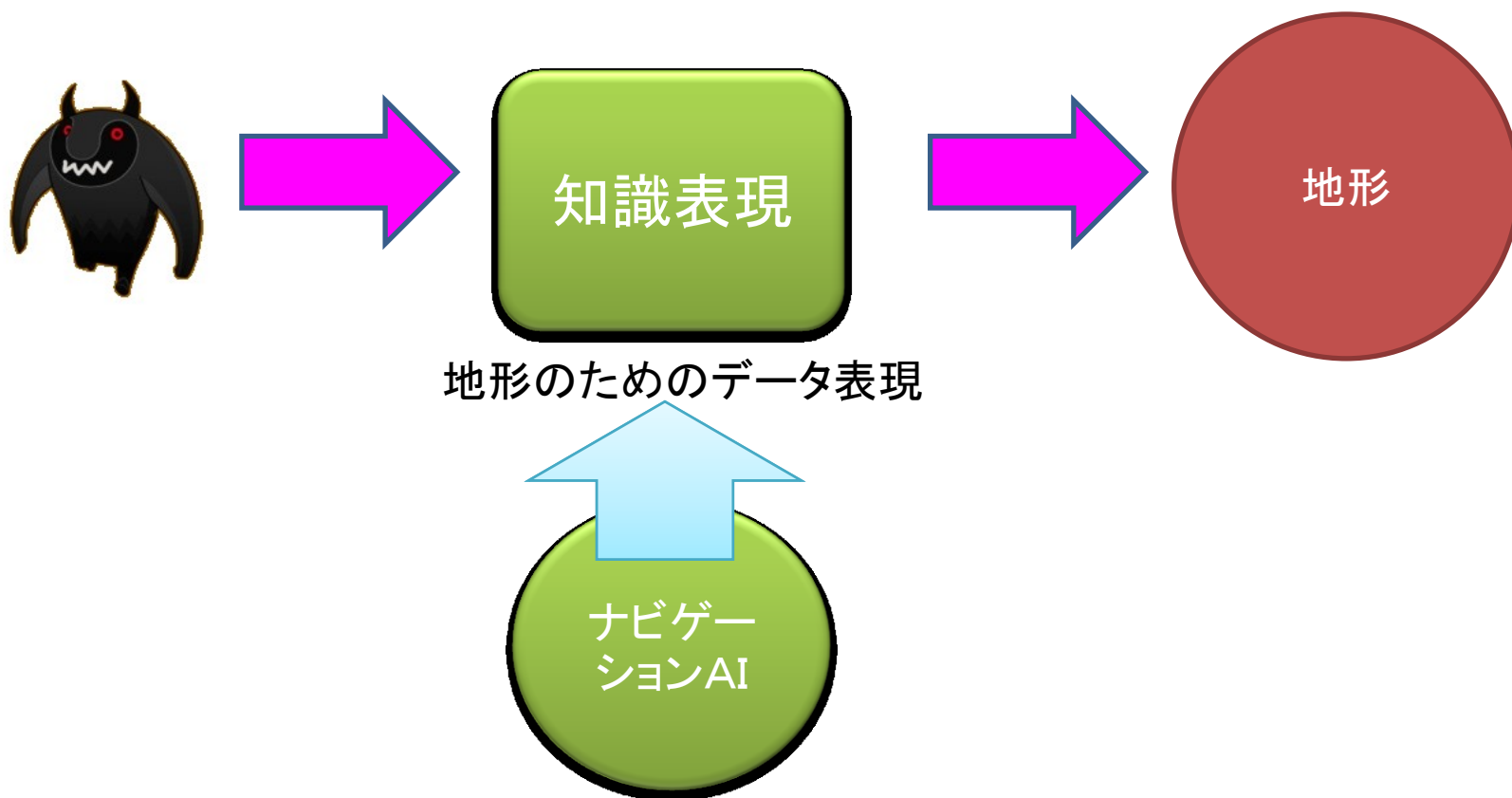
Spaces in the Sandbox: Tactical Awareness in Open World Games (GDC2012) Matthew Jack, Mika Vehkala
<http://www.gdcvault.com/play/1018136/Spaces-in-the-Sandbox-Tactical>

TPSクエリーシステム 隠れるポイントを見つけてくれる。



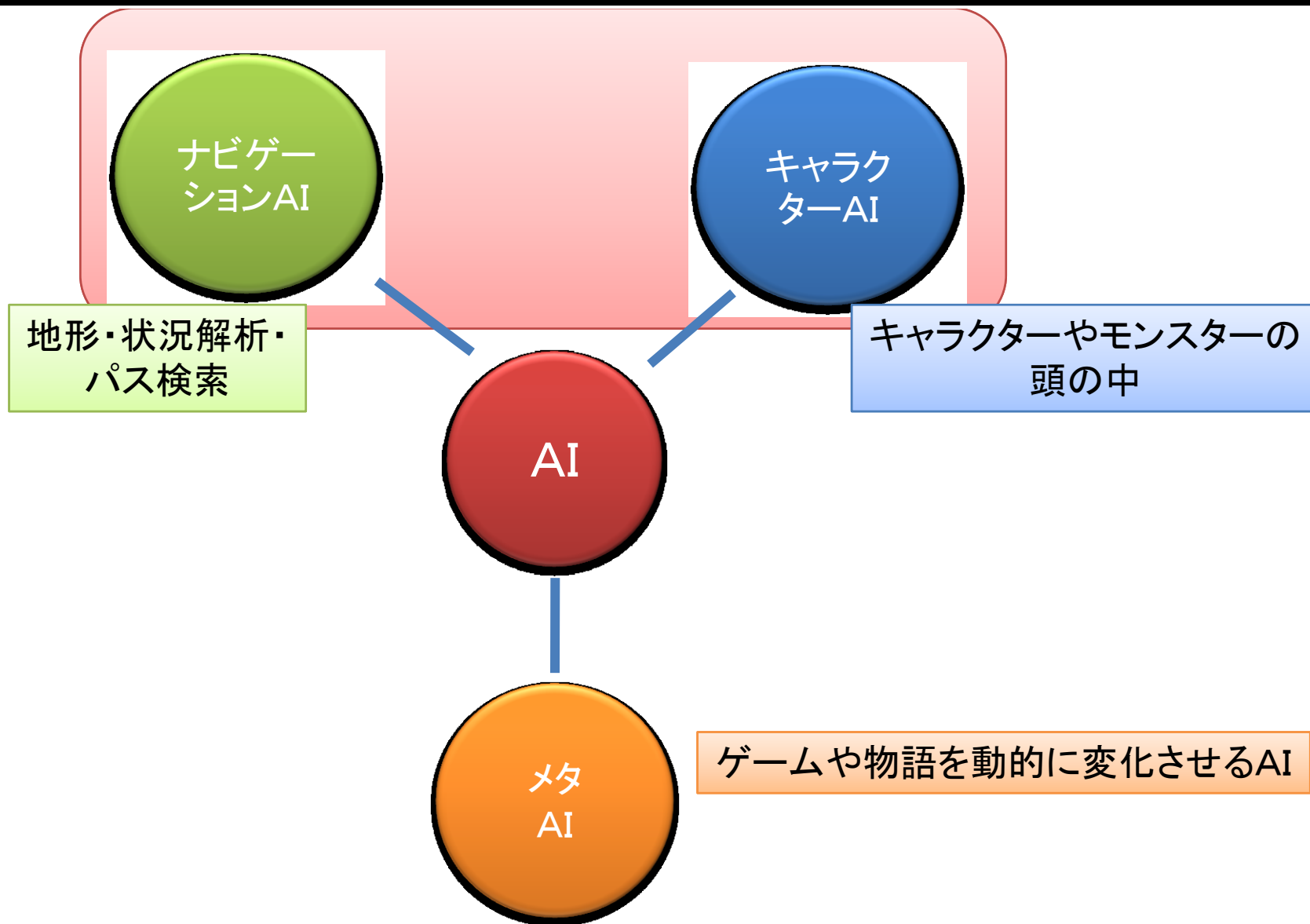
- ① ポイントをレベル上に生成する。
- ② 各ポイントが敵から見えているか、見えていないかを計算する。
- ③ 評価値から目的に応じているか、いないかを分類する。(赤＝駄目、緑＝良い)

ナビゲーションAIの役割

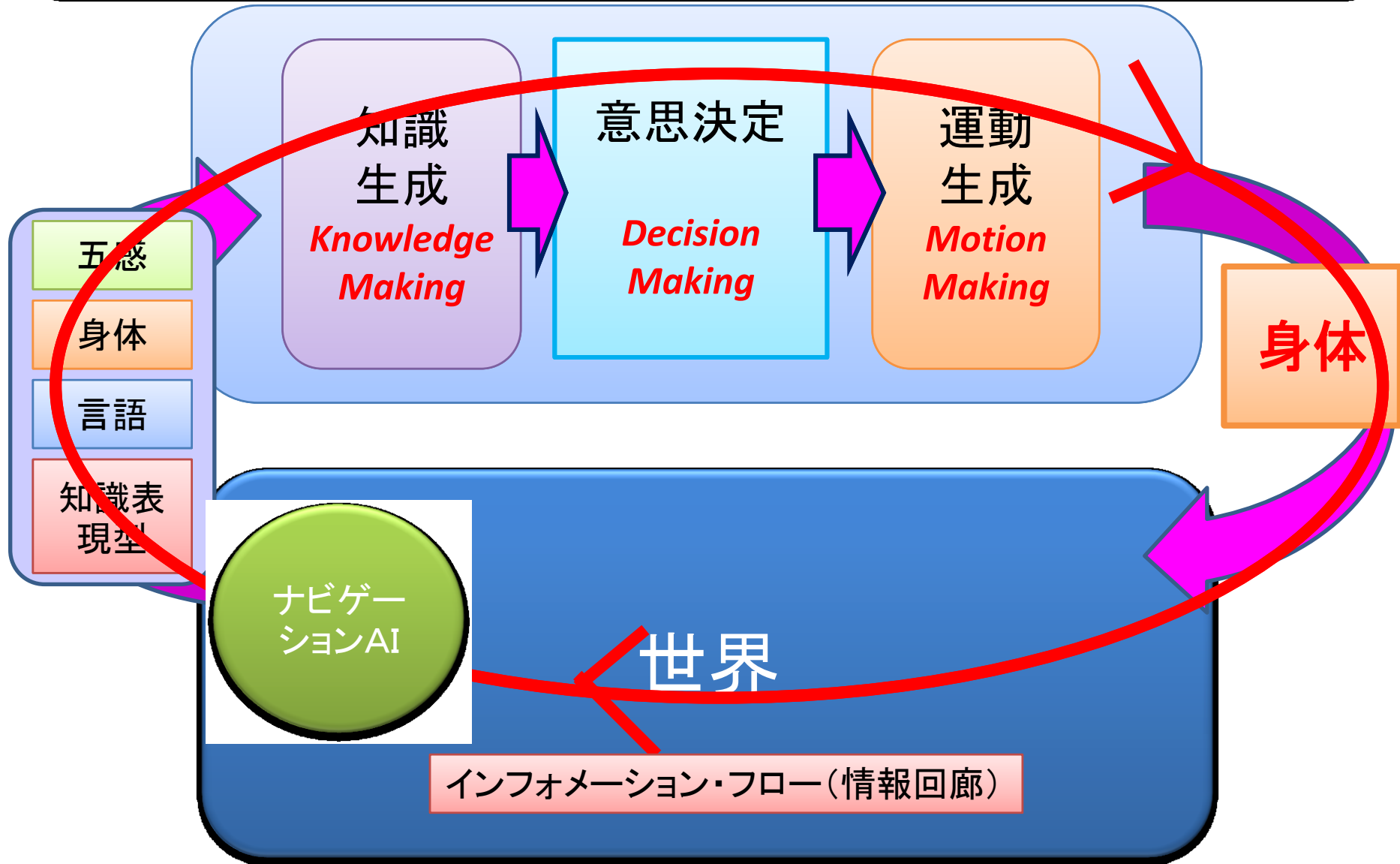


ナビゲーションAIの役割は地形のための知識表現を定義し、リアルタイムにその変化をアップデートすること。

AIの定義



エージェント・アーキテクチャ



AIの定義



ナビゲーションAI

地形・状況解析・
パス検索

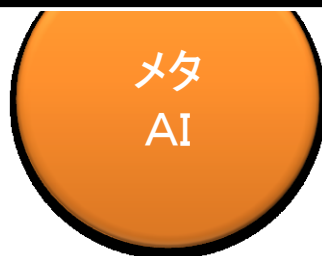


キャラクターAI

キャラクターやモンスターの
頭の中

ナビゲーションAIはキャラクターAIを
レベル(ゲームステージ)に「なじませる」役割を持ちます。

ここで「なじませる」という意味は環境にフィットした、
という意味です。



メタ
AI

ゲームや物語を動的に変化させるAI

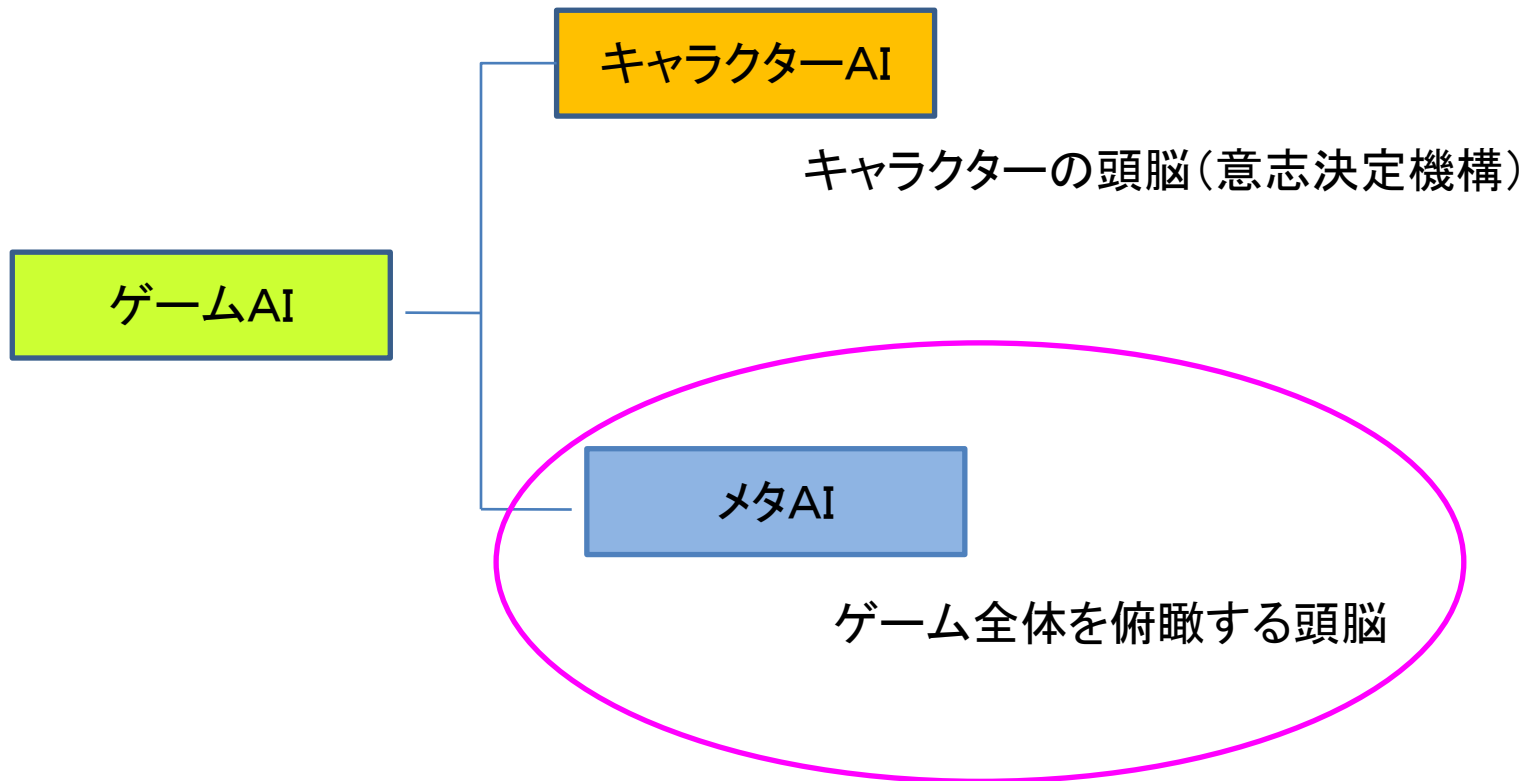
研究テーマ: ナビゲーション

- (1) ナビゲーションの問題 地形やレベルから、知能に必要なより抽象的な、人間的な情報を抽出する技術「地形解析」「状況解析」(terrain analysis, situation analysis)の技術を探求せよ。
- (2) TPS 戦術ポイントシステム (Tactical Point System) はこれからパス検索を含みながら超えて行く技術である。この技術を明確に基礎付けながら発展させよ。

第三部

メタAI

メタAI



メタAIの歴史

1980

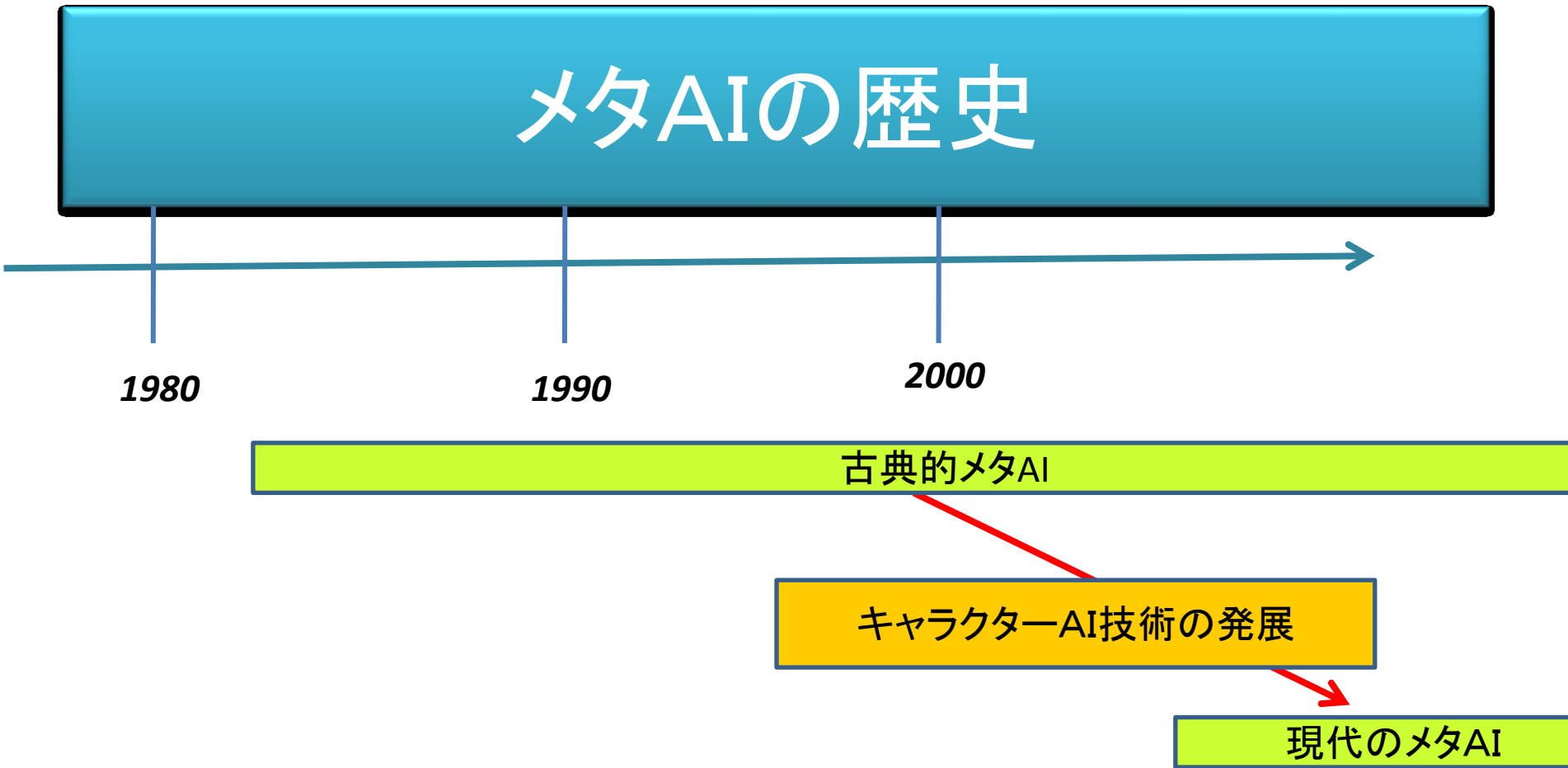
1990

2000

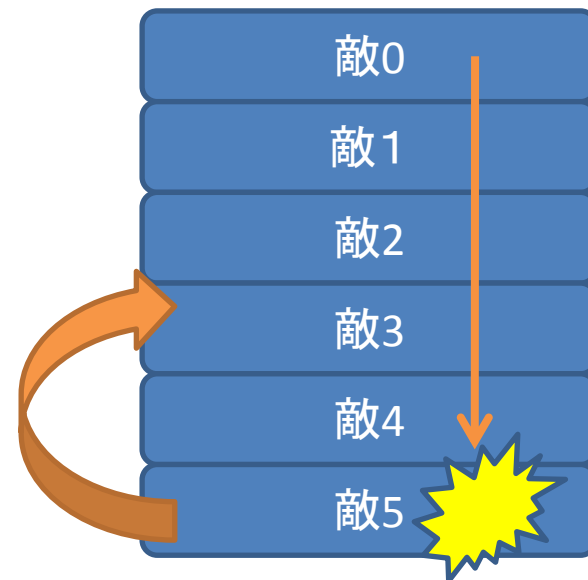
古典的メタAI

キャラクターAI技術の発展

現代のメタAI



(例)「ゼビウス」 敵出現テーブル巻き戻し



遠藤雅伸氏 あと面白い機能なんですけれど、ゼビウスには非常に簡単なAIが組み込まれています。**「プレイヤーがどれくらいの腕か」というのを判断して、出てくる敵が強くなるんです。**

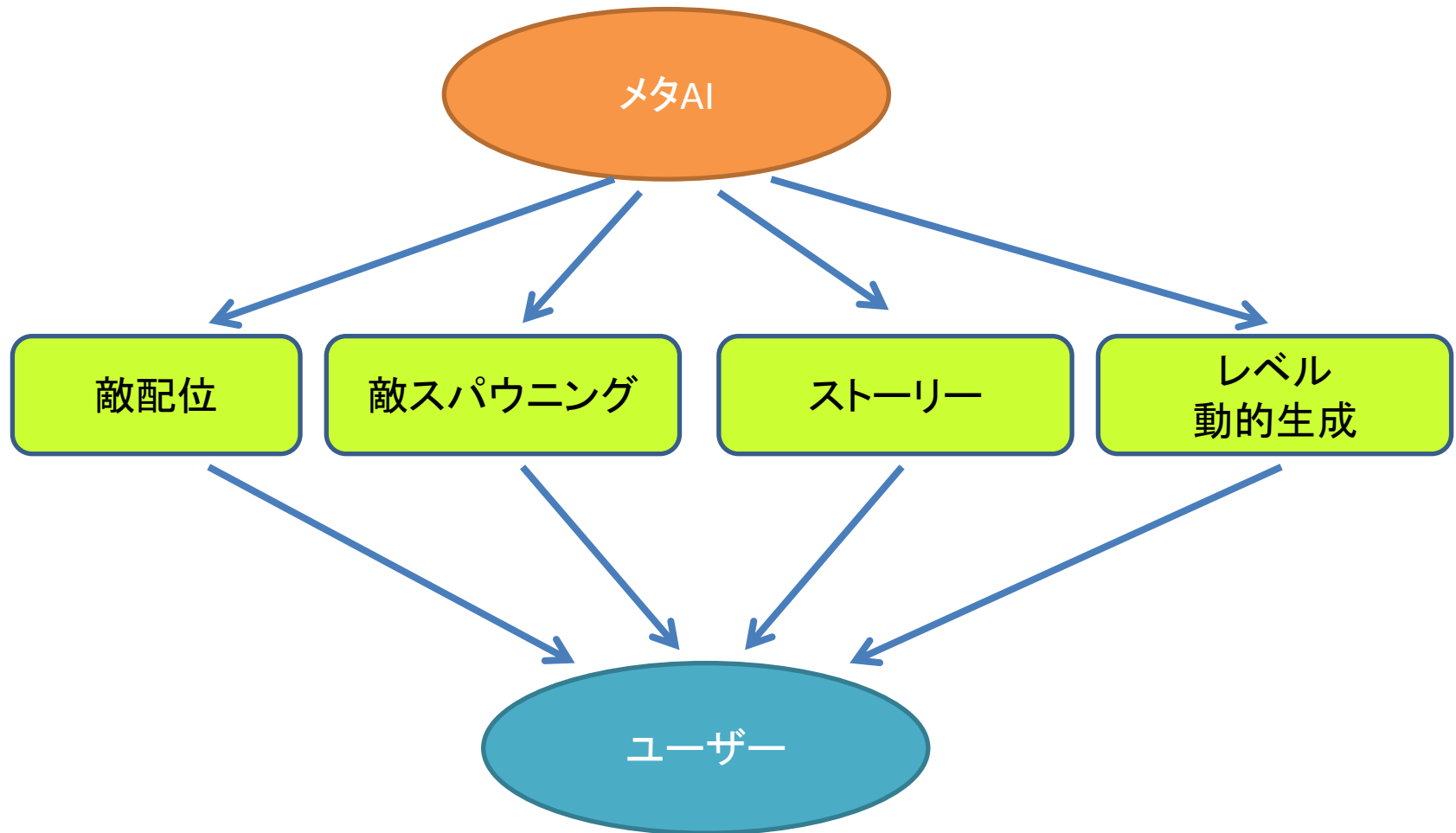
強いと思った相手には強い敵が出てきて、弱いと思った相手には弱い敵が出てきます。そういったプログラムが組み込まれています。ゲームの難易度というのは「初心者には難しくて、上級者には簡単だ」ということが、ひとつの難易度で(調整を)やっていくと起きてしまうので、その辺を何とか改善したいな、ということでそういったことを始めてみたのですけれど、お陰で割合にあまり上手くない人でも比較的長くプレイできる、うまい人でも最後のほうに行くまで結構ドラマチックに楽しめる、そういった感じになっています。 — ゼビウスセミナー —

<http://spitfire.client.jp/shooting/xevious2.html>

現代のメタAI



より積極的にゲームに干渉する。



Michael Booth, "**The AI Systems of Left 4 Dead**," Artificial Intelligence and Interactive Digital Entertainment Conference at Stanford.

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

適応型動的ペーシング

[基本的発想]

- (1) ユーザーがリラックスしている時に、ユーザーの緊張度が一定の敷居を超えるまで敵をぶつけ続ける。
- (2) ユーザーの緊張度が一定の緊張度を超えると敵を引き上げる。
- (3) リラックスすると敵を出現し始める((1)へ)。

Michael Booth, "The AI Systems of Left 4 Dead," Artificial Intelligence and Interactive Digital Entertainment Conference at Stanford.

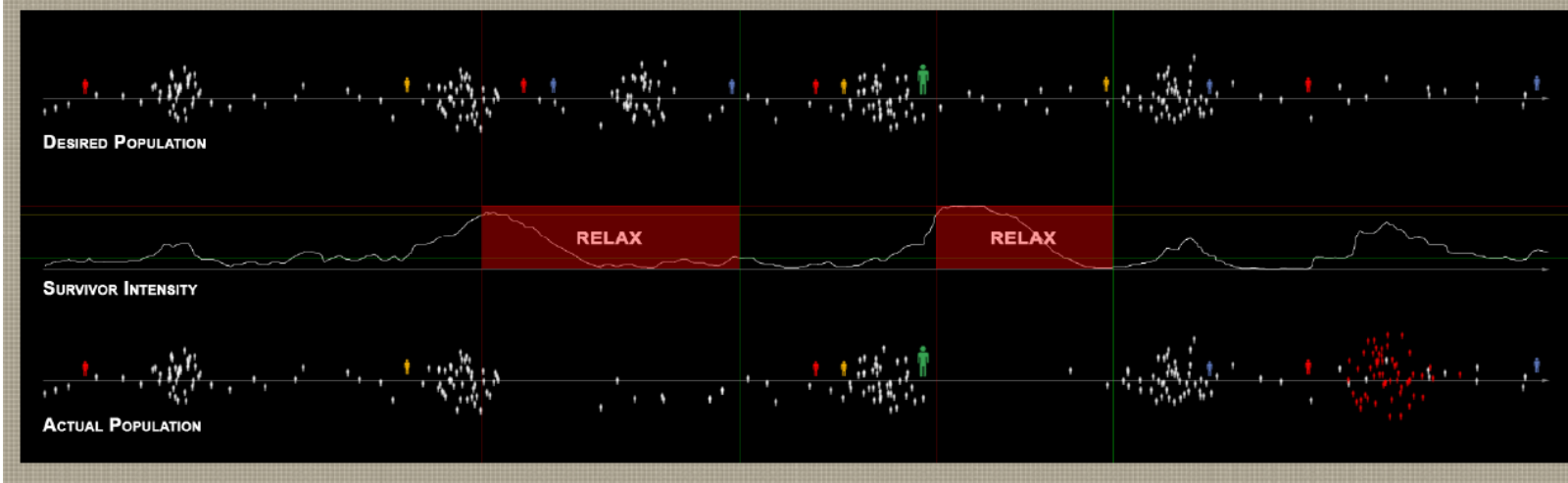
<http://www.valvesoftware.com/publications.html>

メタAI(=AI Director)によるユーザーのリラックス度に応じた敵出現度

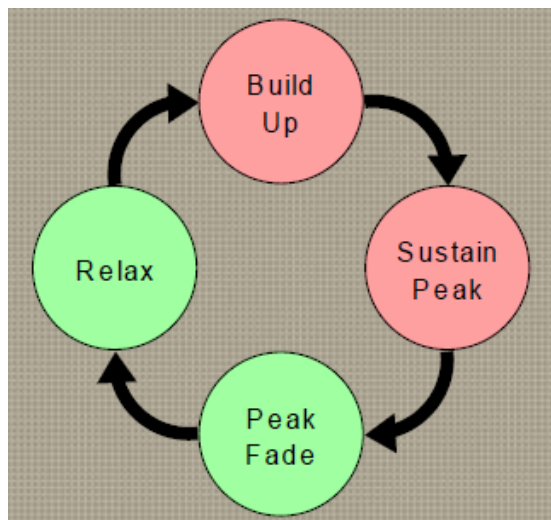
計算によって
求められた
理想的な敵出現数

ユーザーの緊張度

実際の敵出現数



(1) リラックスを破るように敵を出現させる。



Build Up ... プレイヤーの緊張度が目標値を超えるまで敵を出現させ続ける。

Sustain Peak ... 緊張度のピークを3-5秒維持するために、敵の数を維持する。

Peak Fade ... 敵の数を最小限へ減少して行く。

Relax ... プレイヤーたちが安全な領域へ行くまで、30-45秒間、敵の出現を最小限に維持する。

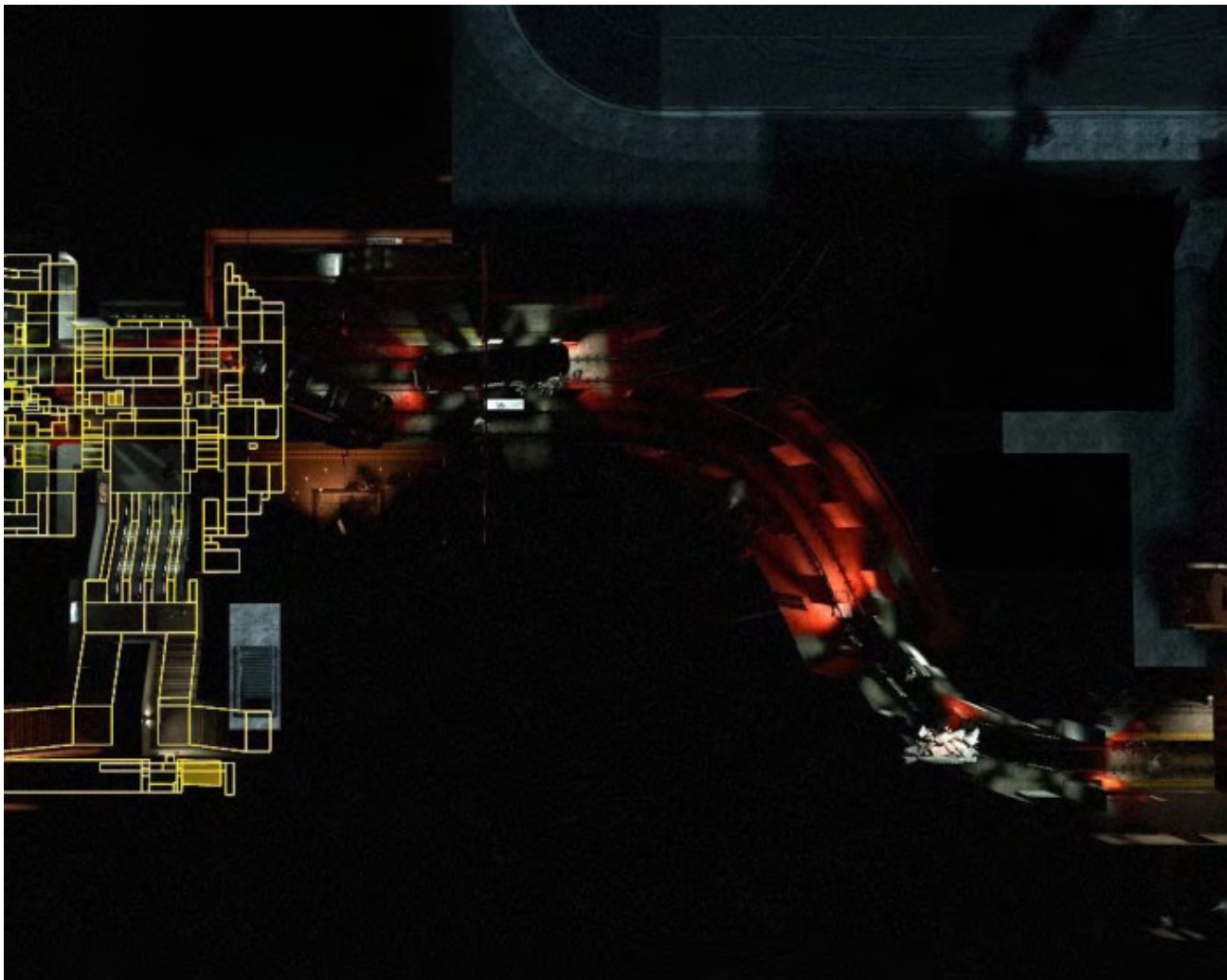
メタAIがゲームを認識する方法

= キャラクターAIが環境を認識する方法

キャラクター用に作成された
ナビゲーションメッシュを
メタAIがゲームの進行を認識する
ために使用する。



メタAIが作用を行う領域 (AAS=Active Area Set)



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

メタAIが作用を行う領域 (AAS=Active Area Set)



メタAIが作用を行う領域 (AAS=Active Area Set)

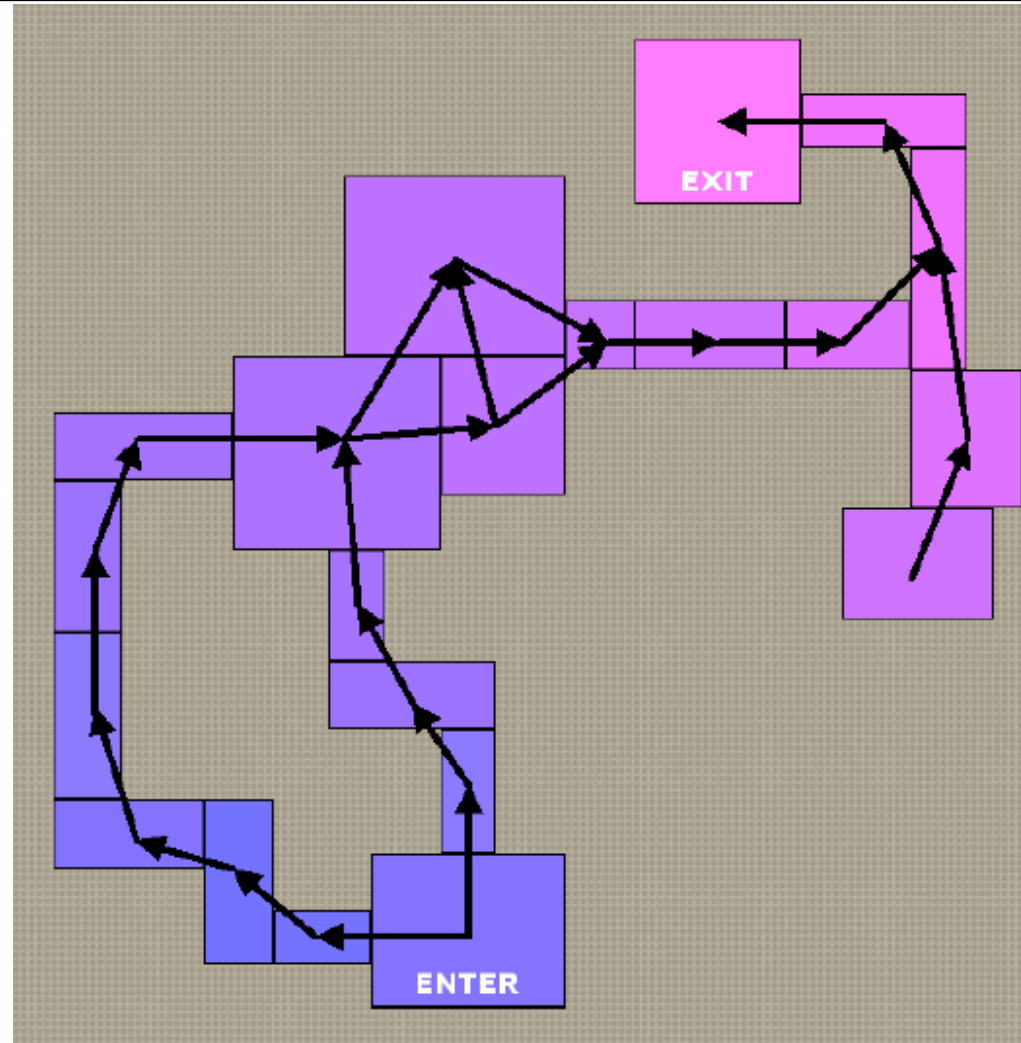


Michael Booth, "The AI Systems of Left 4 Dead," Artificial Intelligence and Interactive Digital Entertainment Conference at Stanford.
<http://www.valvesoftware.com/publications.html>

安全な領域までの道のり(Flow Distance)

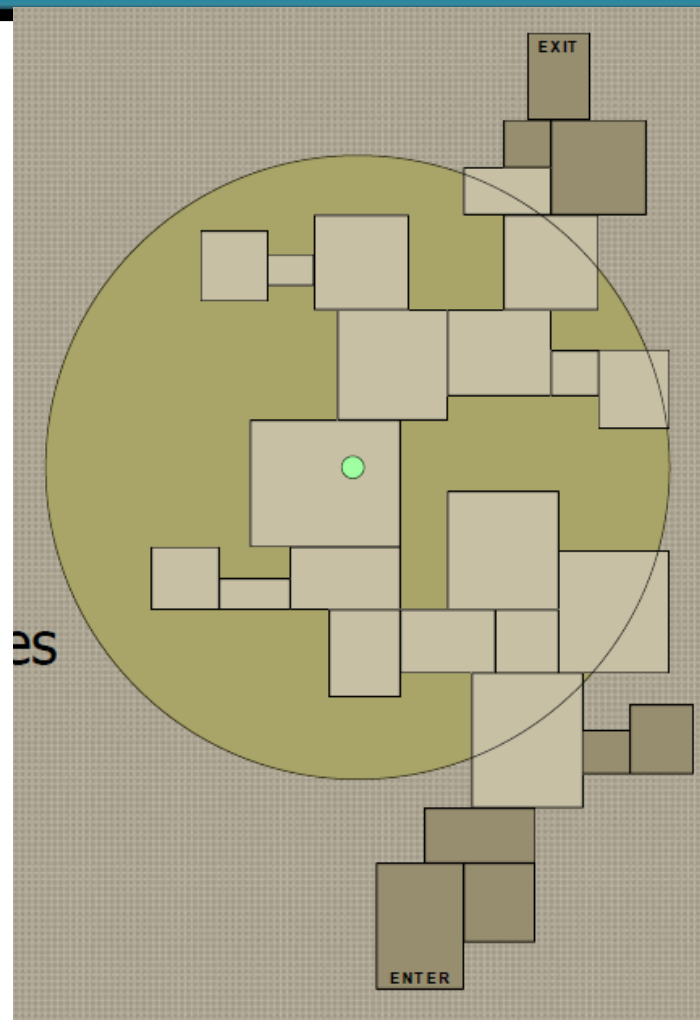
プレイヤー群の経路をトレース・予測

- どこへ来るか
- どこが背面になるか



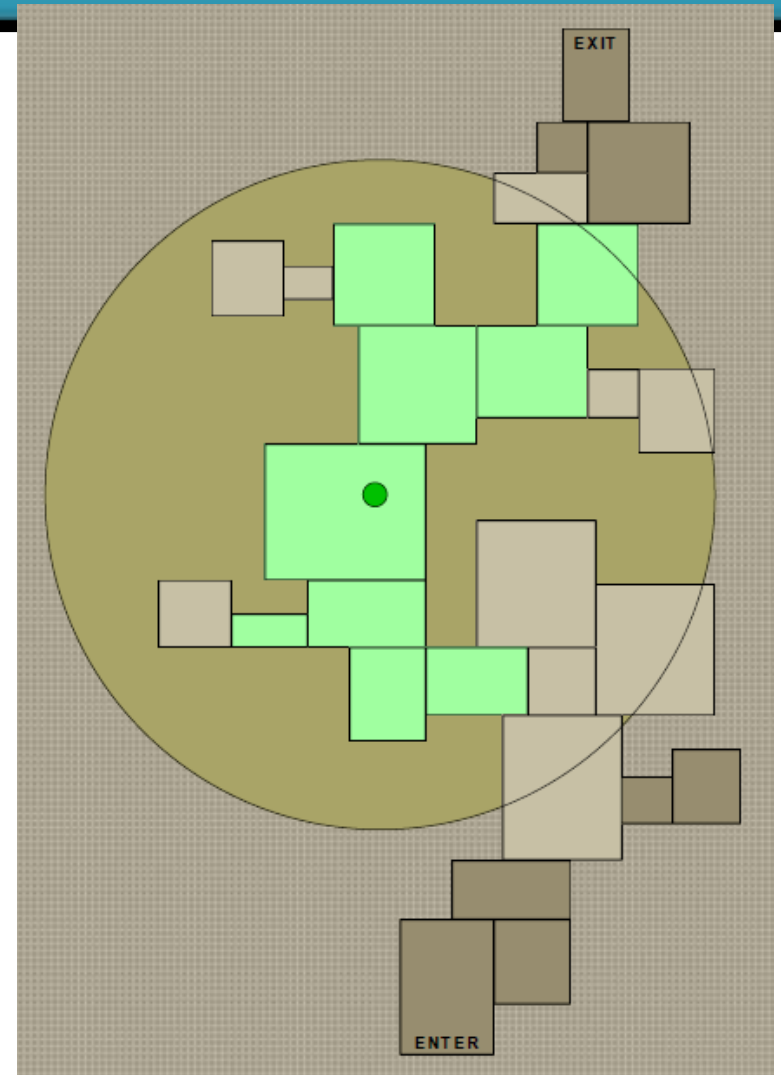
AAS に対して行うこと。

メタAIは AAS に移動に伴い、
敵の群れを生成・消滅させたりする。

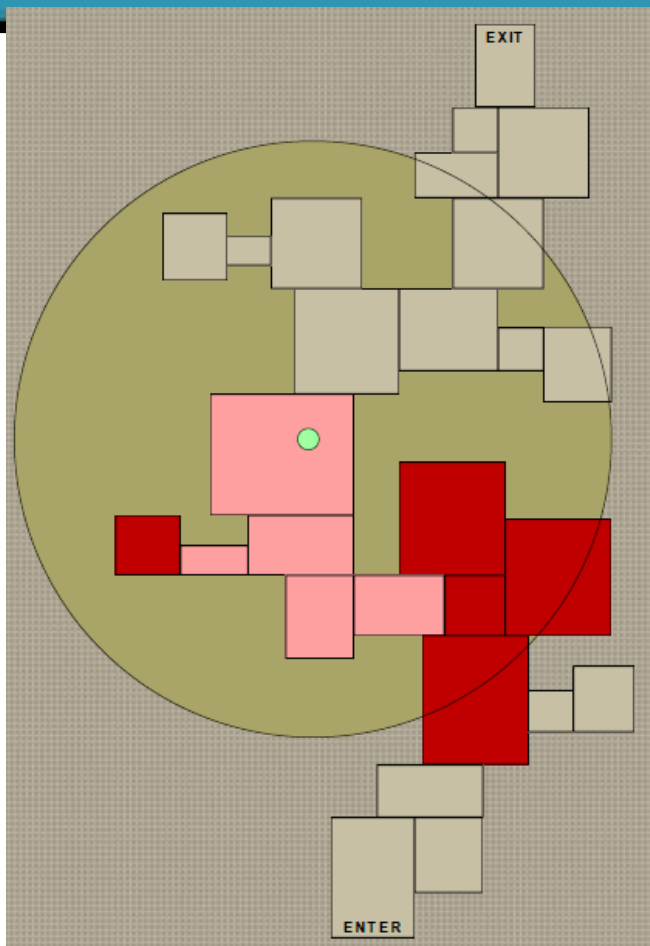


プレイヤーからの可視領域

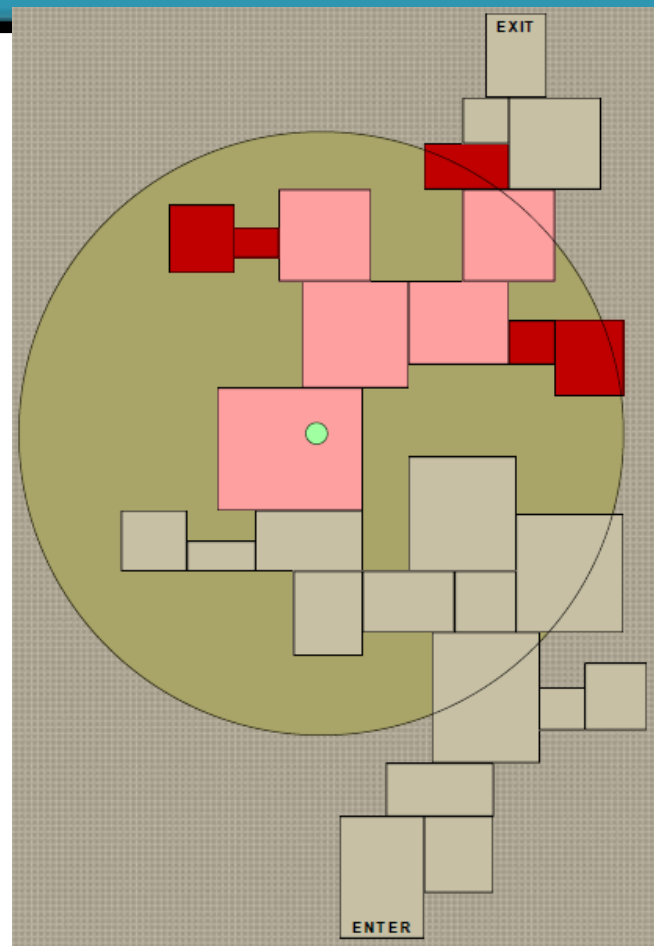
可視領域では、例えば、敵の
スパウニング(発生)などはできない。



敵出現領域



背後



前方

高頻度

Goal: Promote Replayability Procedurally Populated Environments

➤ Structured Unpredictability in Left 4 Dead

- Wanderers (high frequency)
 - Common Infected that wander around in a daze until alerted by a Survivor
- Mobs (medium frequency)
 - A large group (20-30) of enraged Common Infected that periodically rush the Survivors
- Special Infected (medium frequency)
 - Individual Infected with special abilities that harass the Survivor team
- Bosses (low frequency)
 - Powerful Infected that force the Survivors to change their strategy
- Weapon Caches (low frequency)
 - Collections of more powerful weapons
- Scavenge Items (medium frequency)
 - Pipe bombs, Molotovs, Pain Pills, Extra Pistols

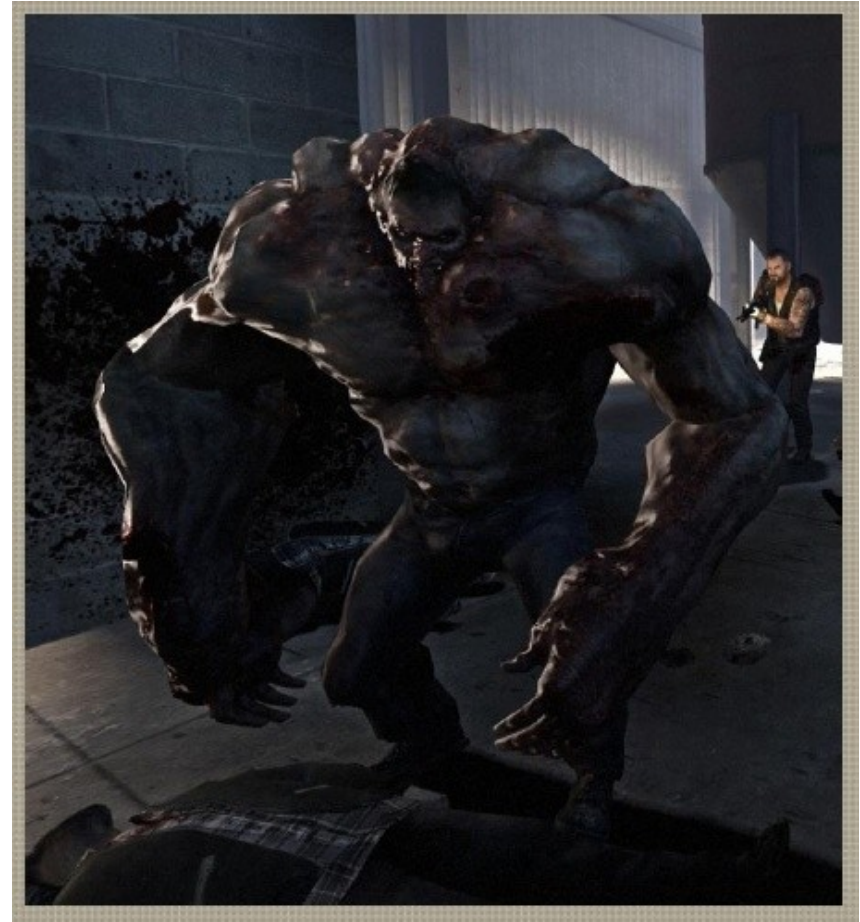
敵の種類、アイテムの種類ごとに出現頻度が違う。

低頻度

ボス出現アルゴリズム

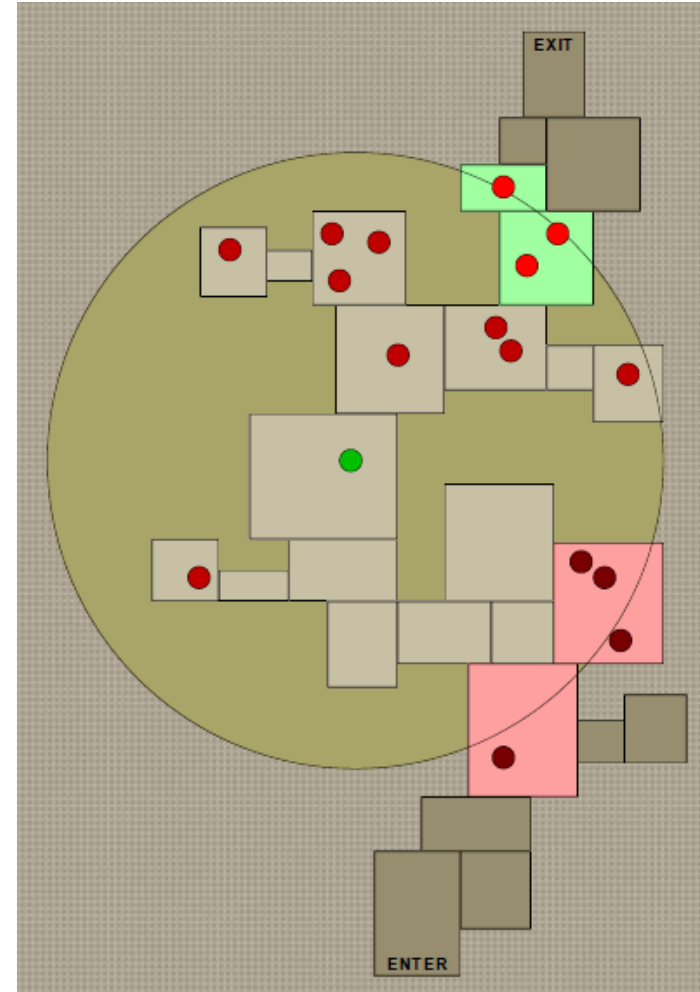
- (1) N体を予想される逃走経路上に配置
- (2) 3つのイベントパターン
(何もいない、を含む)
(例) Tank, Witch, 何もいない
- (3) 同じパターンのくり返しは禁止
(例) Witch, 何もいない、Witch はOK。
Witch, Witch はだめ。

何もいない



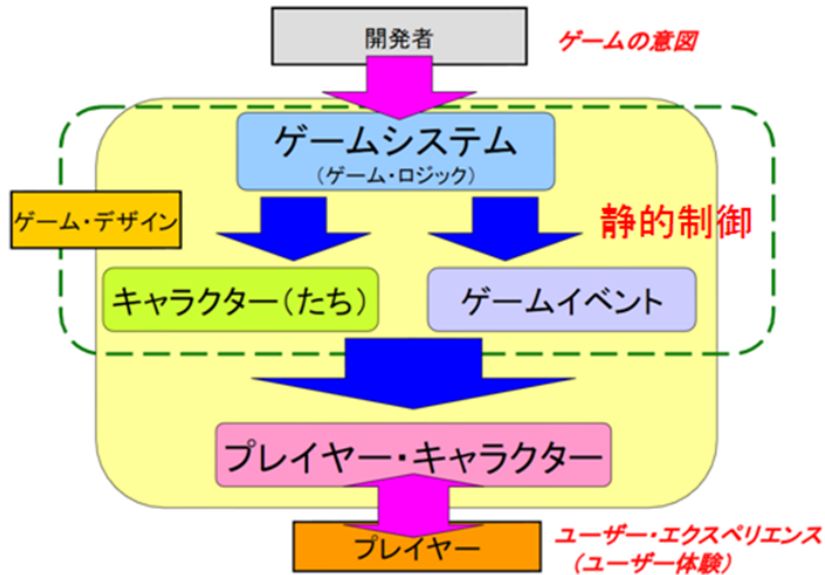
具体的なアルゴリズム

- (1) 各エリアに、出現数 N を決定する
- (2) 出現数 N は予想される逃走経路の長さと要求される密度によって計算される。
- (3) あるエリアがAAS の中に入るとクリーチャーが N 体生成される
- (4) そのエリアがAAS の外に出ると生成が中止され、クリーチャーは消滅される。
- (5) N はそのエリアがプレイヤーから見える場合、或いは、プレイヤーがリラックスモードの場合には、強制的に0になる。



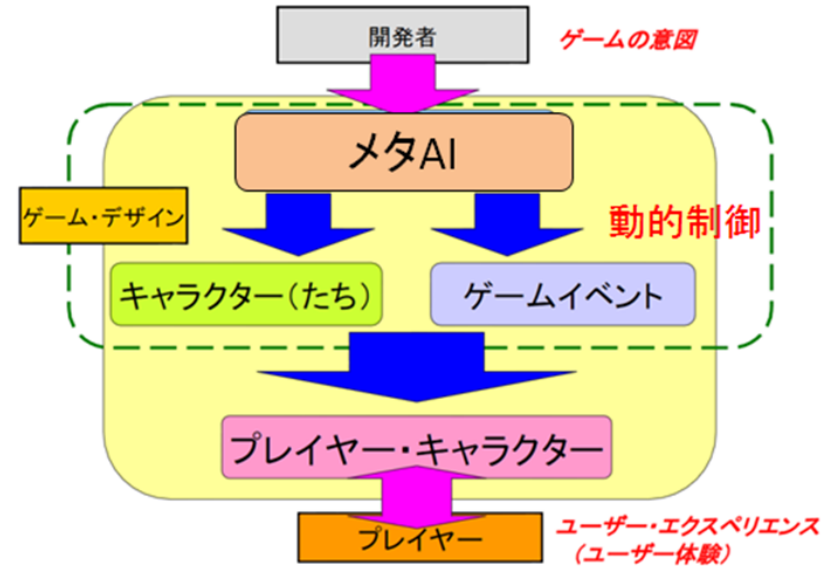
まとめ

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。

メタAIを含んだ 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。

理想的にはプロシージャルに無限のコンテンツが生成される。

メタAIを入れ替えるだけで、ゲームコンテンツが入れ替わる。メタAIという軽い部分だけを配信することでコンテンツを入れ替えることができる。

参考文献

- (1) Michael Booth, "**Replayable Cooperative Game Design: Left 4 Dead**," Game Developer's Conference, March 2009.
- (2) Michael Booth, "**The AI Systems of Left 4 Dead**," Artificial Intelligence and Interactive Digital Entertainment Conference at Stanford.

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

- (3) 三宅 陽一郎, “メタAI”, 「デジタルゲームの技術」
P.186-190, ソフトバンククリエイティブ

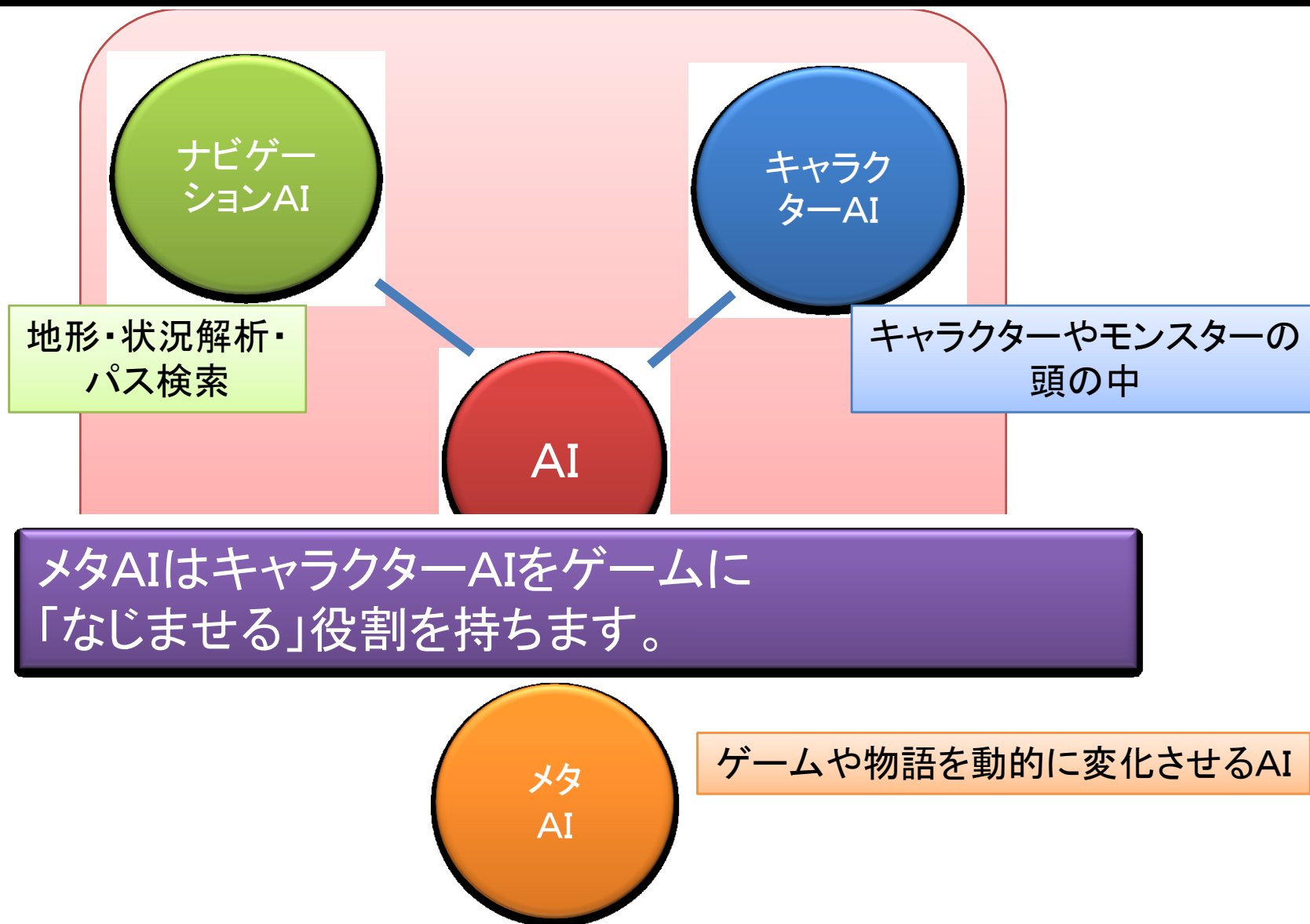
研究テーマ: メタAI

- (1) **メタAI** メタAIは、何をゲームから観測し、何をユーザーから観測し、どのような調整やダイナミクスをゲームの中で産み出したいか、で定義される。まだまだ事例が少ない。応用例を考案し実験せよ。
- (2) **メタAIと新しいゲーム配信** メタAIの仕組みでゲームを実装すれば、メタAIの部分を変えるだけで新しいゲーム性の実現できる。このような新しいゲーム配信の仕組みを明確に定義付けよ。

第四部

まとめ

AIの定義



まとめ

- ゲームAIには、**3つのAI**があります。
- それぞれが違う部分のAIを担当します。ゲームという問題の各部分を分散して解いています。
(**分散人工知能**)
- かつ、それぞれが**協調**することで、新しい機能生み出しています。
- それぞれのAIには別々の研究テーマがあります。
- その**テーマを一つ一つ克服して行く**ことが、結果としてゲームAIを前進させて行くことになります。

ゲームAI入門 「環境と知性の相互作用」

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

Twitter: @miyayou

感想・ご質問は メールtwitter やfacebookまで

ゲームAI千夜一夜

<http://blogai.igda.jp/>

三宅陽一郎 論文・講演資料集(4)

<http://blogai.igda.jp/article/92899318.html>