

# クロムハウন্ズにおける 人工知能開発から見るゲームAIの展望

CEDEC 2006 講演

三宅 陽一郎  
(FromSoftware, Inc.)

2006.8

y\_miyake@fromsoftware.co.jp



# クロムハウズにおける 人工知能開発から見るゲームAIの展望

CEDEC 2006 講演

三宅 陽一郎  
(FromSoftware, Inc.)

2006.8

y\_miyake@fromsoftware.co.jp

# クロムハウন্ズにおける 人工知能開発から見るゲームAIの展望

CEDEC 2006 講演

三宅 陽一郎

(FromSoftware, Inc.)

y\_miyake@fromsoftware.co.jp

- (1) ムービーの文字が小さいですので、なるべく前でご覧ください。
- (2) 配布資料は、読んで理解して頂けるように作っており、  
パワーポイントと対応していません。
- (3) パワーポイントは、CEDEC の サイトで公開される予定です。

# この講演で感じて頂きたいもの

「人工知能って、今こういうふうな状況なんだ」

「人工知能って、こういうことができるんだ」



持って返って頂きたいもの

「自分のゲームにも、人工知能技術を取り入れてみようかなあ」

# 講演のための用語の準備

AI = Artificial Intelligence、人工知能

- COM ... プレイヤー以外のキャラクター
- AI ... 人工知能(技術)のことを指す。

今日は、人工知能が、ゲームに対してどのような可能性を拓くのか、を説明したいと思います。

# 本講演の構成

## 第1部

ゲームへの人工知能技術の導入のあり方 (15分)  
質疑応答 (5分)

## 第2部

クロムハウズにおける人工知能の開発 (45分)  
質疑応答 (5 + 5分)

## 第3部

これからのゲームとゲームAIのための展望 (10分)  
質疑応答 (5分)

# 第1部

ゲームへの人工知能技術の導入のありかた  
(20分)

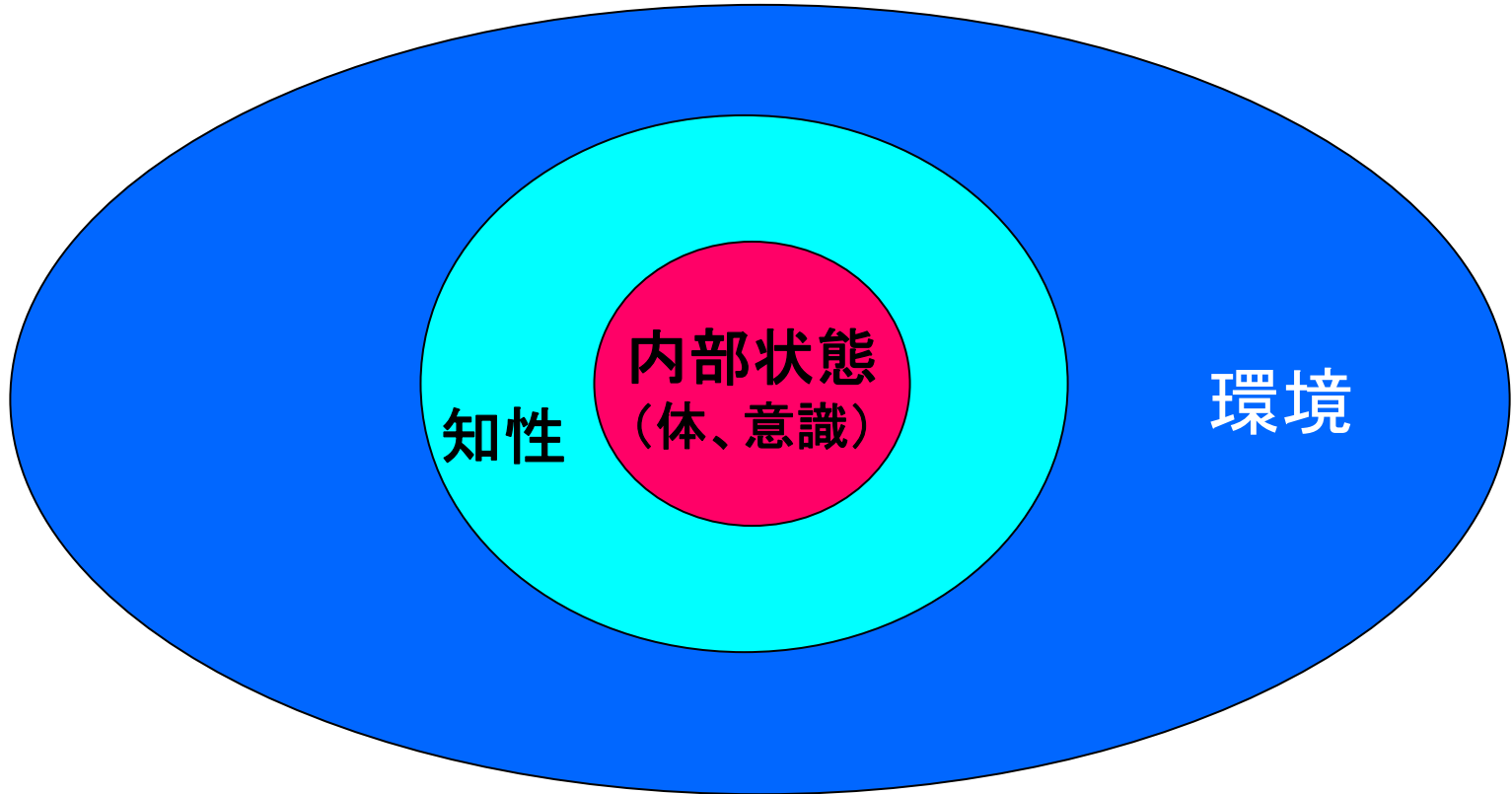
人工知能技術とゲームとの関係を概観する



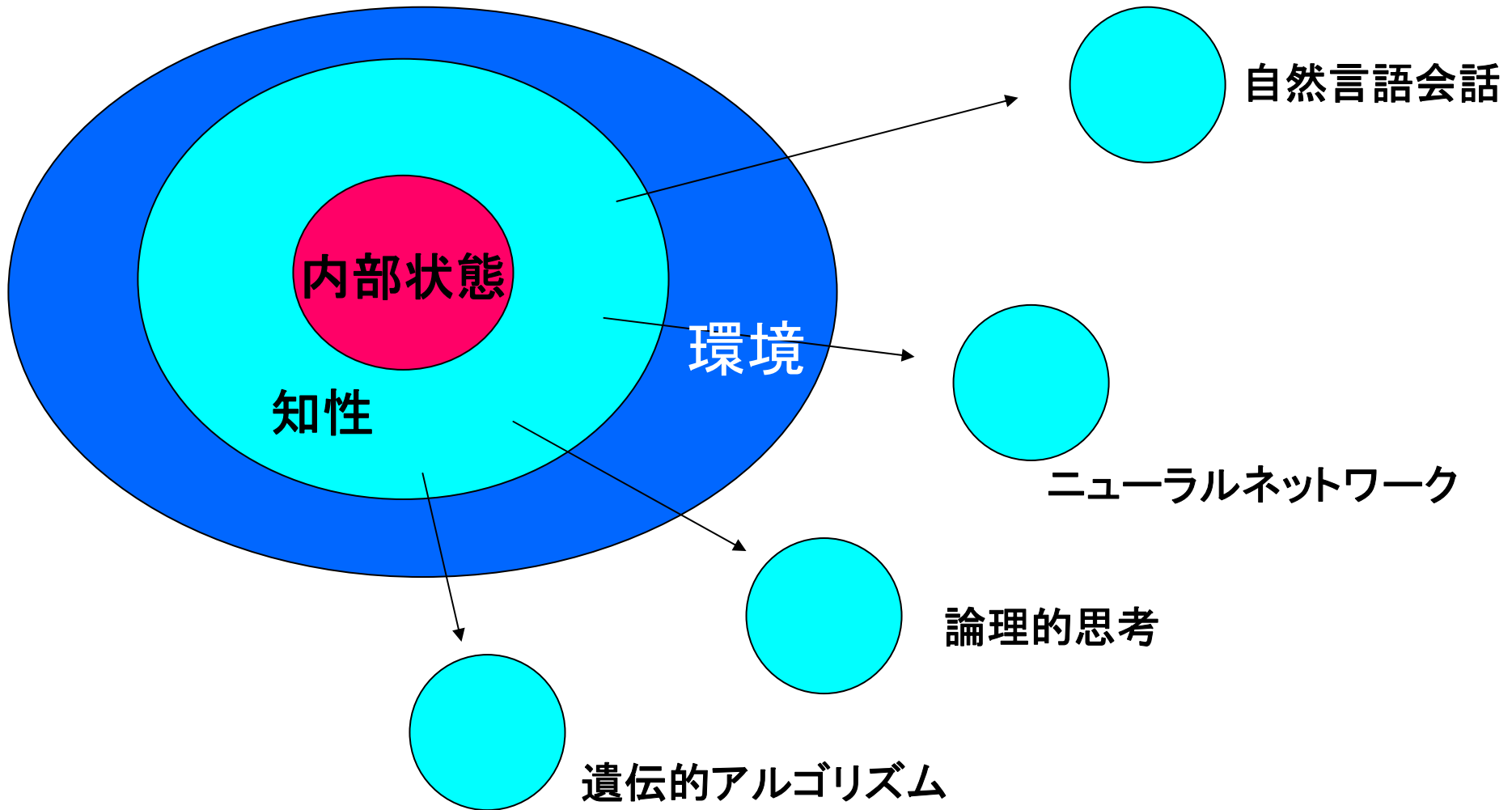
人工知能とは何か？

# そもそも知性とは？

- 生き物が、外部の環境に対して、適応するために持ったシステム



# 人工知能とは？



生命の知能を機能ごとに、取り出して発展させたもの。

# 人工知能の研究の歴史（若い！発展 中）

人工知能学会ホームページより

[http://www.nistep.go.jp/achiev/ftx/jpn/stfc/stt038j/0405\\_03\\_feature\\_articles/200405\\_fa03/200405\\_fa03.html](http://www.nistep.go.jp/achiev/ftx/jpn/stfc/stt038j/0405_03_feature_articles/200405_fa03/200405_fa03.html)

# この50年で発展して来た人工知能技術たち

(1) エージェント

(8) Finite State Machine

(2) ニューラルネットワーク

(9) Automaton

(3) Rule-Base A.I.

(10) Subsumption Architecture

(4) ベイジアンネットワーク

(11) Competitive evolution

(5) 自己組織化マップ

(12) A\* Path Finding

(6) 遺伝的アルゴリズム

....

(7) 遺伝的プログラミング

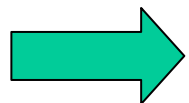
アカデミックに学問として蓄積されて来た  
人工知能技術には、たくさんの技術がある。

どれぐらいゲームに応用されているの？  一部だけ

(赤色...よく応用 青色...少しだけ応用)

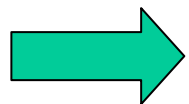
# 何故、今まで応用されて来なかったか？

- (1) 特にグラフィック偏重でゲームが発展して来た。
- (2) メモリ容量や計算時間が必要なものが多い。
- (3) 一般的な基礎がなく、逆に、  
基礎を探求して行く学問である。



それぞれの分野で断片的である(入りやすいが奥が深い)

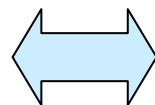
- (4) 数学、哲学、科学の上に組み上げられている分野が多く、応用の仕方が難しく見えにくい。



数学 ... 偏微分方程式、多変数ダイナミクス

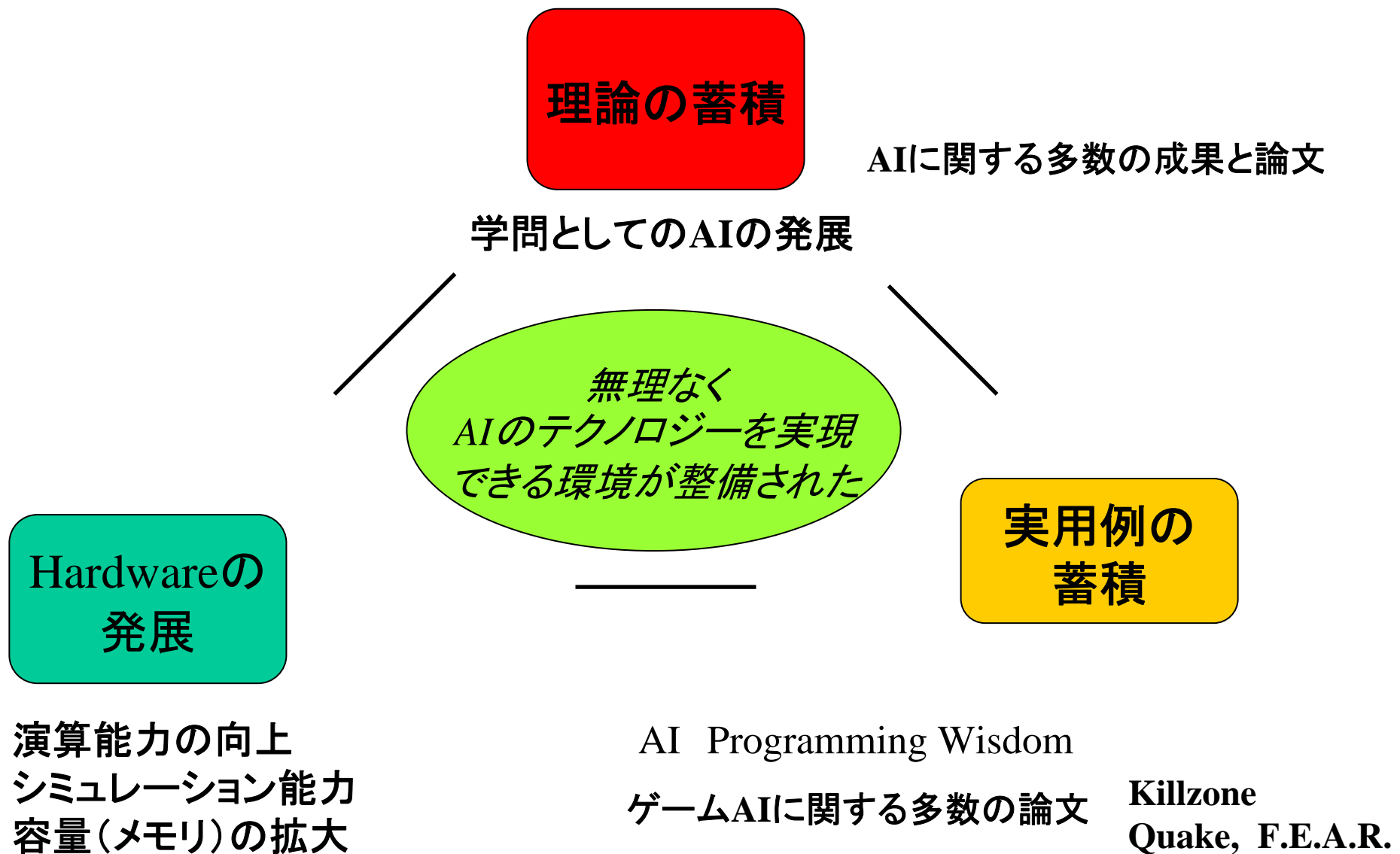
科学 ... 進化論

哲学 .... 現象学

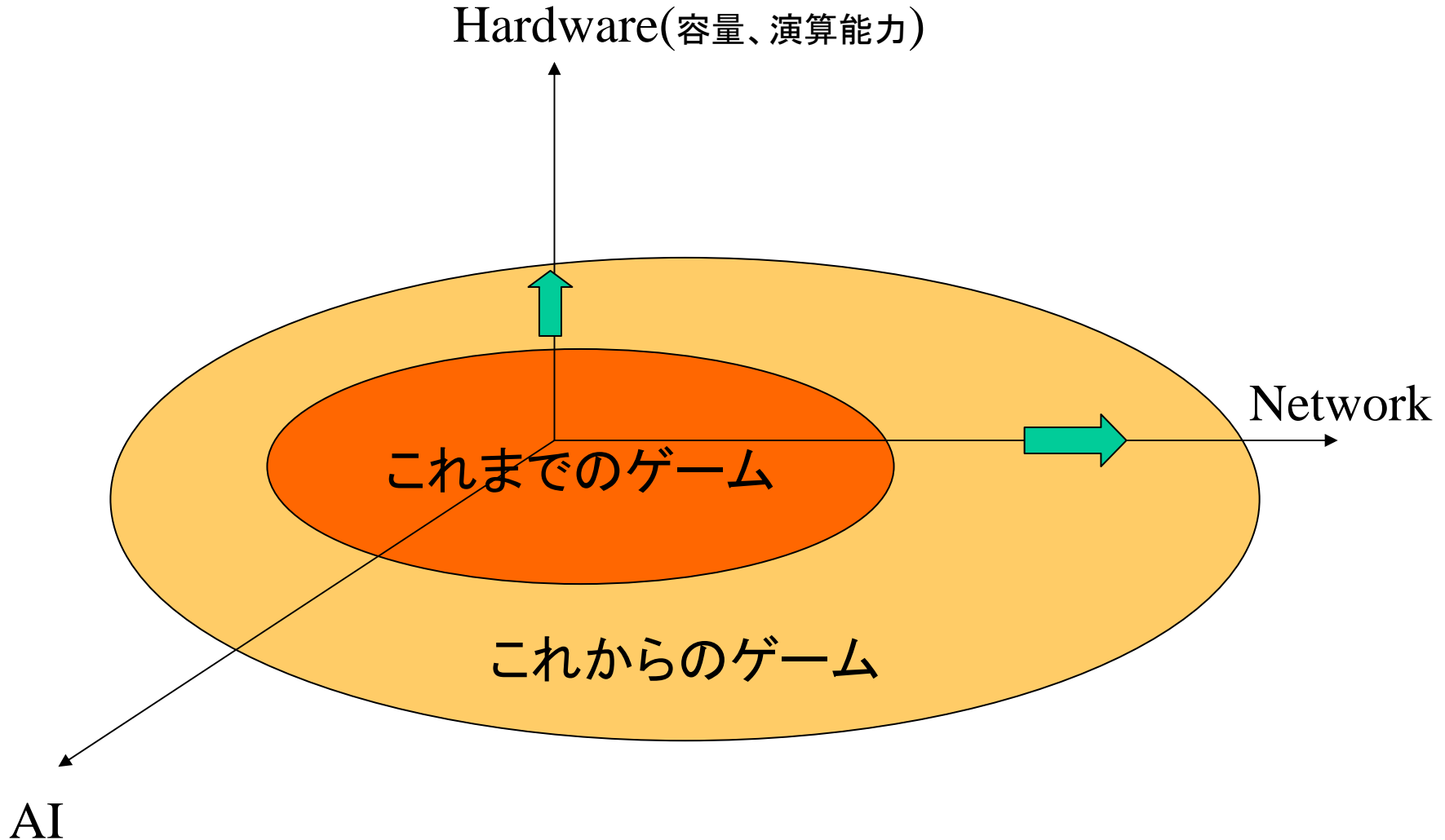


CG (四次元の数学)

# 何故、これからは応用が可能か？

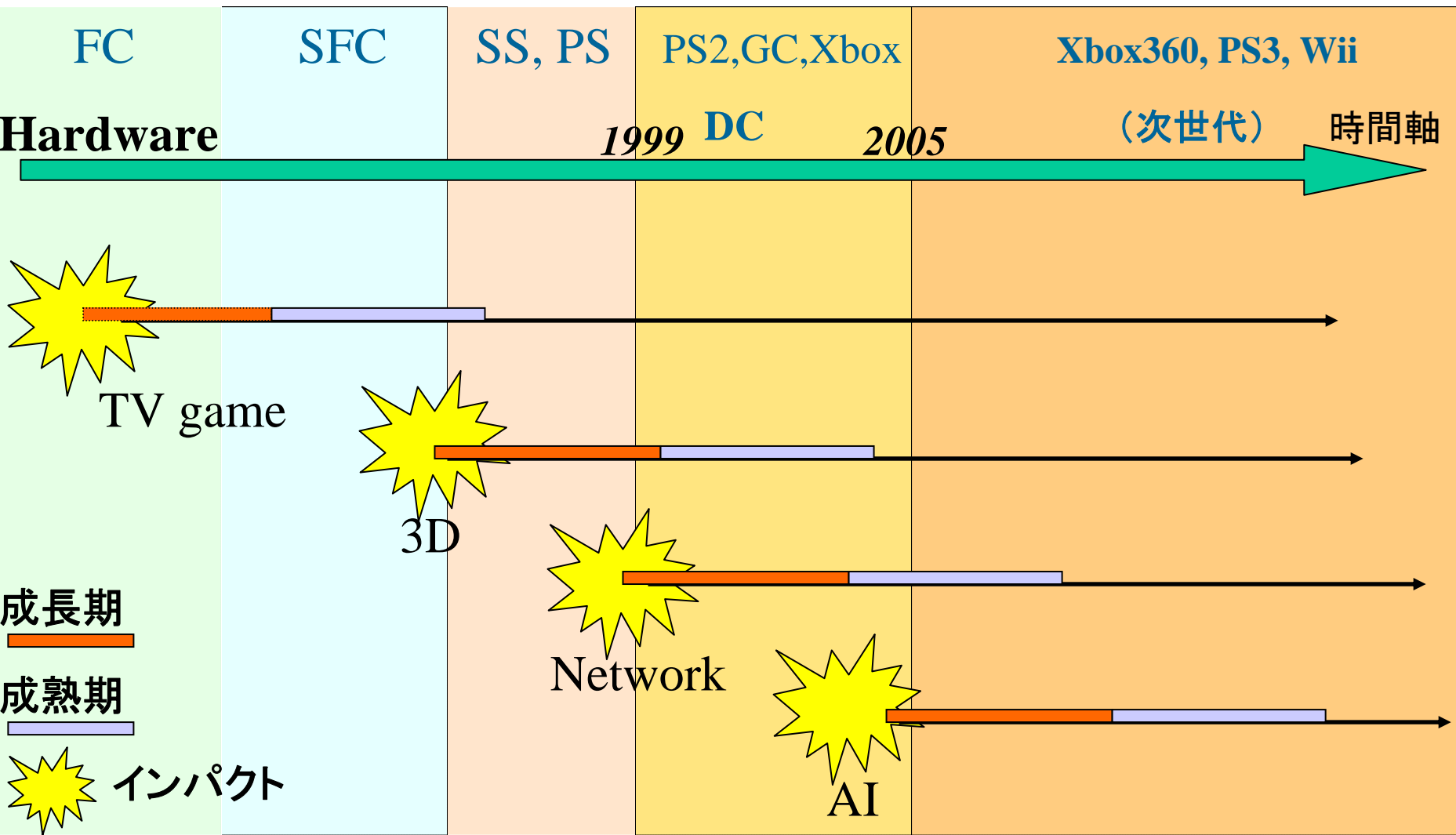


# 次世代ゲームの第3の軸としてAI





# 人工知能技術の導入の適切なタイミングはいつか？



技術の歴史的な流れから見て、人工知能技術のゲームへの応用は、次世代で成長し、次々世代で成熟するだろう。

# 人工知能を導入の仕方と問題点

# 人工知能技術の導入の仕方

(Step 1) 人工知能技術を調査、研究しておく。



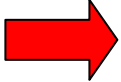
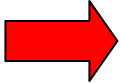
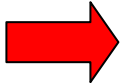
(Step 2) ゲームの要求に対して、どのような技術が必要であるかを見抜き、ゲーム設計の中に組み込む。



(Step 3) 製作ラインの中にワークフローを実現する。  
(新しい技術には、新しいデバッグ方法)

この具体例を、クロムハウズを題材に説明いたします

# 人工知能導入の問題点と注意点

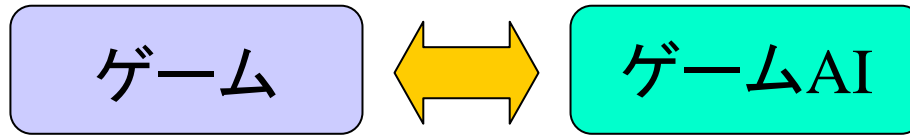
- (1) 日本では開発の情報を公開することが少ない。  
ゲームへの人工知能の応用の情報は、圧倒的に欧米が多い。  
 **文献は英語で探す。海外の方が総合的に質と量が高い。**
- (2) 応用の仕方は、抽象的な思考を通して行う必要がある。  
既存の具体例をそのままそのゲームへ応用できる例は多くない。  
 **基本的(簡単)な知識からクリエイティブに発想しよう。**
- (3) 数学的原理や、アルゴリズムは、特殊なケースで動かなかったり、方程式が発散したりする場合がある。  
 **研究ならばある程度動けばいいが、製品には十分なデバッグと安全装置が必要！**

# 欧米と日本におけるゲームAIの開発の比較

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

(例) Quakeを扱った修士論文は世界的に有名になった。

# ゲームAIにおける日本のチャンス



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

# 第1部 まとめ

- (1) 人工知能のゲームへの応用は、これまでは制約があって難しかったが、次世代から発展の段階に入り、次々世代で成熟すると予想できる。
- (2) 現在のところ、欧米がゲームAIの分野で、ゲームとしても開発環境としても整備を発展させつつある。
- (3) しかし、日本にも、独自のチャンスがある。

# 第1部 質疑応答



# 本講演の構成

## 第1部

ゲームを発展させるための人工知能技術の  
導入のあり方の概論 (20分)

質疑応答 (5分)

## 第2部

クロムハウズにおける人工知能の開発 (40分)

質疑応答 (10分)

## 第3部

これからのゲームとゲームAIのための展望 (10分)

質疑応答 (5分)

## 第2部

# クロムハウズにおける 人工知能の開発

人工知能の「発想の仕方」と「技術導入の仕方」

# Contents

クロムハウズAIの開発を題材に説明を行う

- I. クロムハウズ紹介
- II. クロムハウズのCOMを作る  
(中間で質疑応答)
- III. まとめ

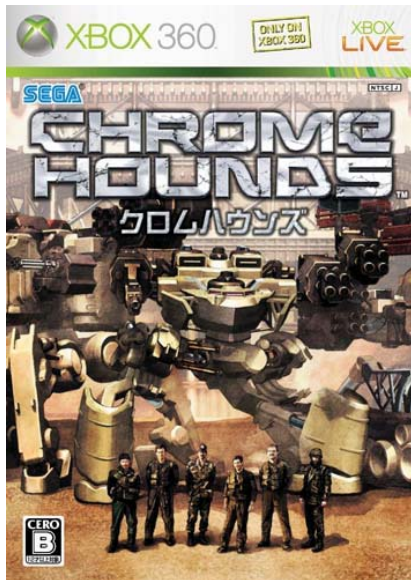
# I. クロムハウন্ズ紹介

紹介ムービーをご覧ください



# クロムハウন্ズ

発売元 : セガ  
デザイン : フロムネットワークス  
開発元 : フロムソフトウェア



2006年6月、世界7カ国で、同時発売

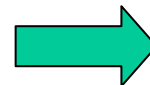
# クロムハウنزとはどんなゲームか？

## 基本設定:

- (1) オンライン
- (2) アクション
- (3) 勝利条件
- (4) スカッドと呼ばれるチームをベース
- (5) 80を超える複雑なマップ
- (6) コミュニケーションのシステム
- (7) 戦略の重要性

Multiplayer Online Battles on Xbox 360

詳細





# 基本設定

## (1) オンライン



オンラインでチーム同士で戦う

## (2) アクション



基本はメカ対メカのアクションゲーム

## 基本設定

### (3) 勝利条件



15分以内に、敵全員を殲滅するか、敵本拠地を破壊！

時間切れの場合は、コムバス占拠数がより多い側が勝者！



## (4) スカッドと呼ばれるチームをベース



最大6人のプレイヤーがスカッドとなって、一度に同じ戦場に出ることができる。

## (5) 80を超える複雑なマップ



市街、山岳、さばく、河、湖、海など、  
多岐に渡る80以上 のマップ  
ハウズは15mのスケール、マップは3km四方

## (6) コミュニケーションのシステム



コムバス（通信塔）を占拠してネットワーク領域を作り、  
初めてチームメンバーと通信可能！  
コムバスをどう取るか、  
敵、味方の情報をどう伝えるか、の情報戦だ！





各プレイヤーは、広大なマップを15分間、  
敵を予測して、戦略を立てなければならない。  
接近戦だけで、勝敗はつかない

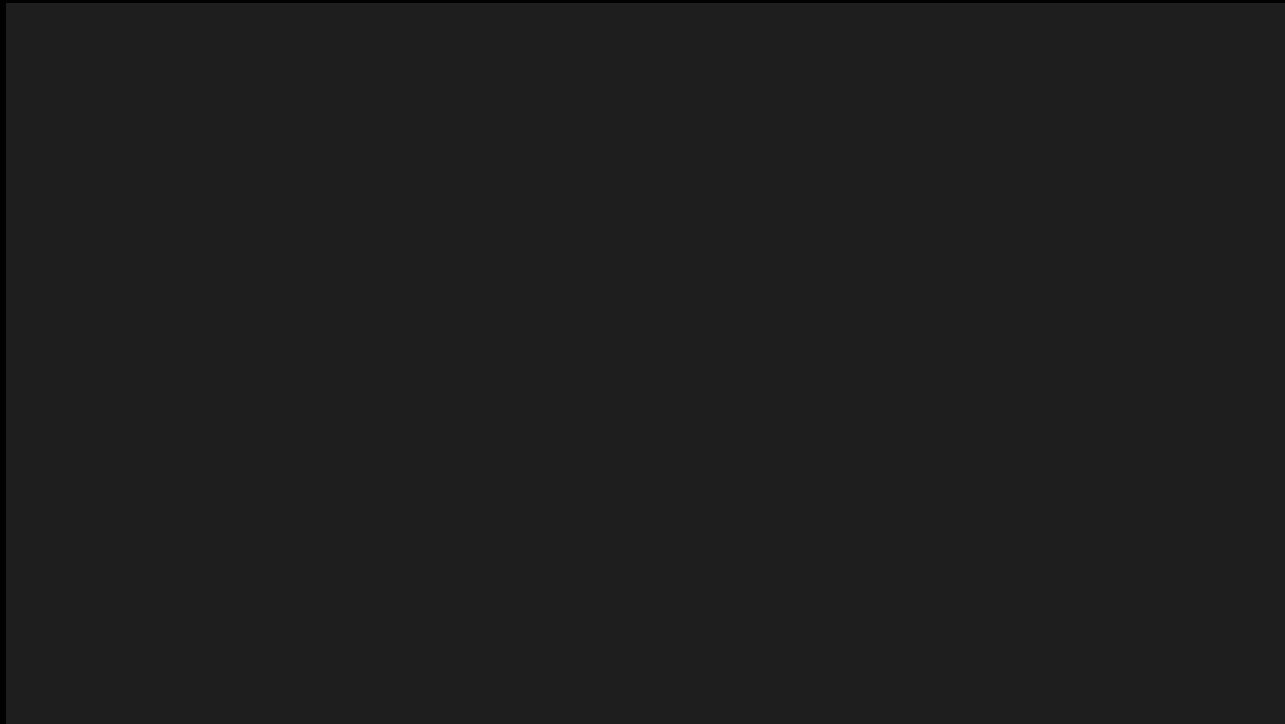
# COM への要求

オンライン上で対戦チームが見つからないときに、  
人間の代わりに、プレイヤーチームと戦う  
COM のチームを作ってくれ



申し訳ありませんが、  
30秒だけこの問題を考えてみましょう。

クロムハウズで、人間の代わりに、  
プレイヤーチームと戦う COM のチームを作る



## II. クロムハウন্ズのCOMを作る

アイデアから実際の製作、デバッグまでを追体験しよう

## より詳細なクロムハウズAIへの要求の考察

- プレイヤーと同じ条件(チームメンバー数、平等な情報)のもとで、同等に渡り合える能力。

COMは15分間の間、戦略、戦術、情報戦においてプレイヤーと同等に渡り合わねばならない。



自律的に思考し、自分で移動する  
「**自律型エージェント**」(autonomous agent)  
を作ろう。

# 「自律型エージェント」とは

## (1) 自律型

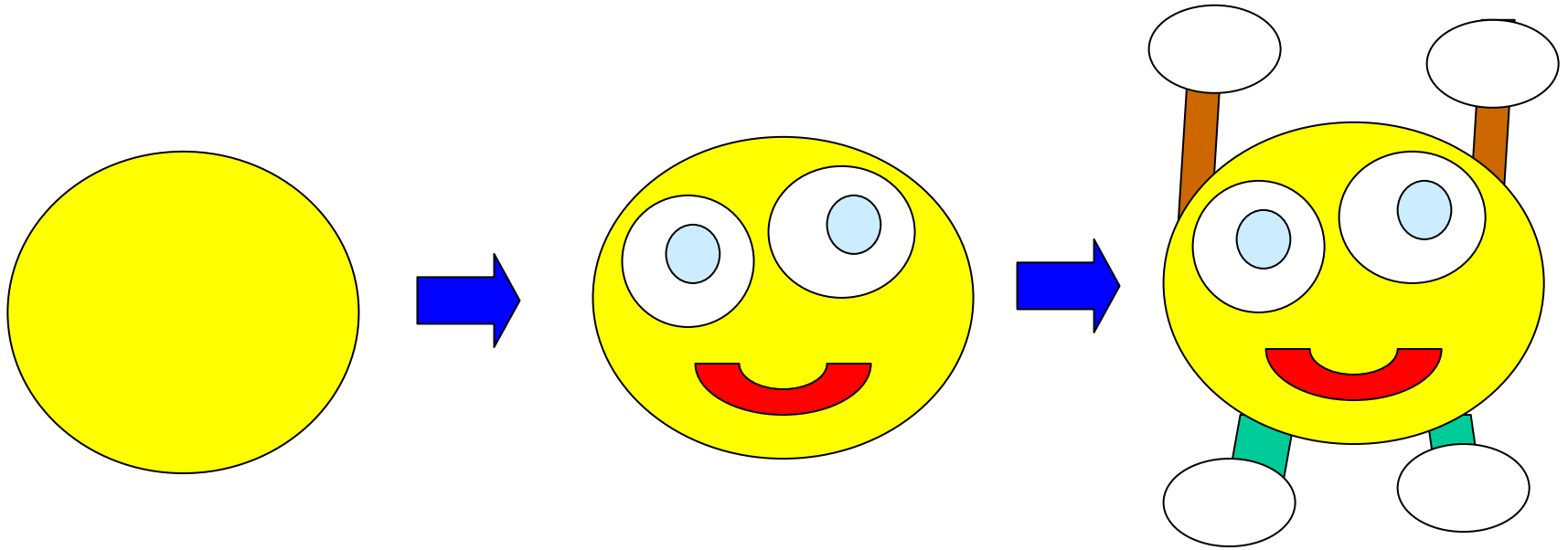
あらかじめプログラムされた行動だけでなく、自ら状況に応じた行動をすることができる。

## (例) 自律型システム

... システムに不具合があると、自分自身で修復する。

## (2) エージェント

環境に対して情報を集める感覚(センサー)を持ち、  
また、環境に対して働きかけることができる能力を持つ。



体(オブジェクト)を持ち

感覚を持ち

世界に働きかける  
能力を持つ

# 世の中にあるエージェントの具体例

## サッカーエージェント

フィールドの情報を集める能力があり、  
かつ、サッカープレイヤーとして参加する。  
⇒ ロボカップ



## ネットワークエージェント

ネットワーク上で、情報を検索し、データベースに情報を収集する。  
⇒ ロボット、ウイルス

お掃除エージェント ... 部屋の情報を確認して、塵を集める

⇒ お掃除ロボット

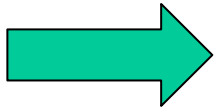
# 「自律型エージェント」とは

## (1) 自律型

あらかじめプログラムされた行動だけでなく、自ら状況に応じた行動をすることができる。

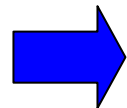
## (2) エージェント

環境に対して情報を集める感覚(センサー)を持ち、環境に対して働きかけることができる能力を持つ。



## 自律型エージェント

周囲の環境に対して、自ら情報を収集し、それに基づいて判断し、行動する能力を持つ実体。



人間のような知能のモデル

# クロムハウズで 自律的エージェントを実現することを考える

自分で判断して、自分で行動して、  
プレイヤーと戦う

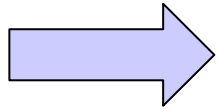
(1) 考える - どんな技術が必要だろうか？

(2) 歩く - どんな技術が必要だろうか？



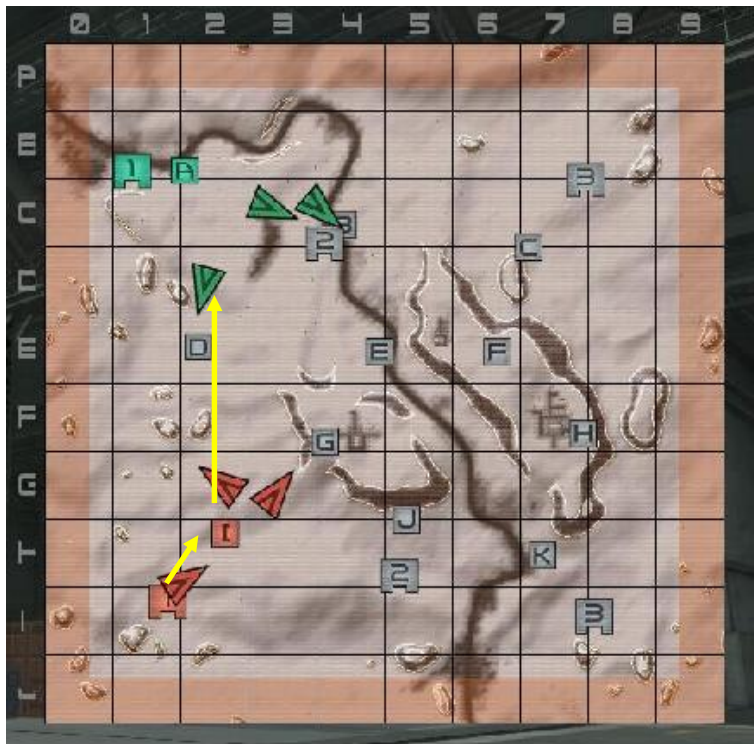
# COMの思考

# COMは、どういうふうを考えれば、うまく行動できるだろうか？



人間同士の戦闘がどうなっているかパターンを集めてみよう

例 1



本拠地から出発



最も近くの通信塔へ行って占拠

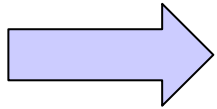


敵本拠地へ前進する。



敵と戦闘

# COMは、どういうふうを考えれば、うまく行動できるだろうか？



実際の戦闘がどうなっているかパターンを集めてみよう

例 2



本拠地から出発

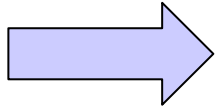


近いコンバスを巡回してパトロール



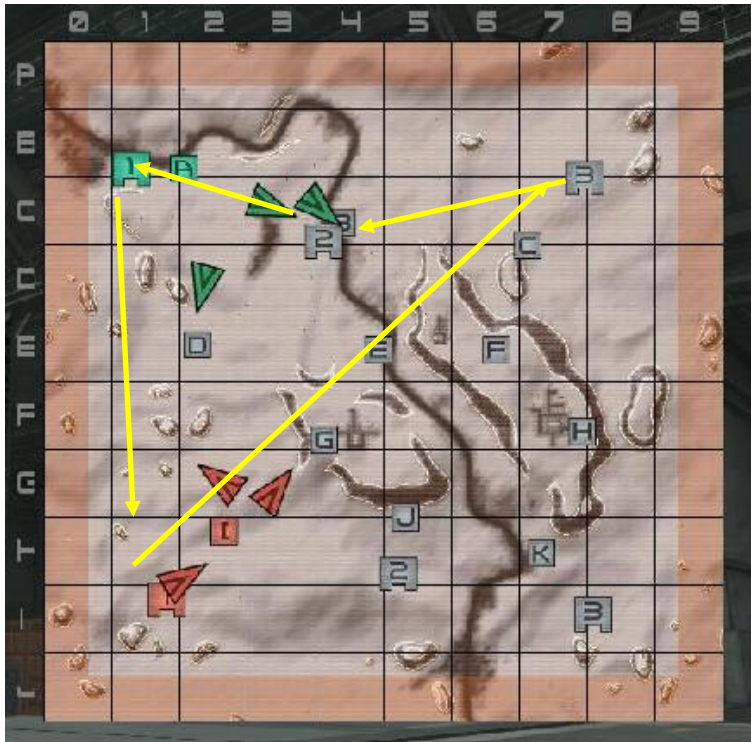
自分の本拠地に戻って敵が来るのを待つ

# COMは、どういうふうを考えれば、うまく行動できるだろうか？



実際の戦闘がどうなっているかパターンを集めてみよう

例 3



本拠地から出発



敵の本物の本拠地がどれであるかを偵察



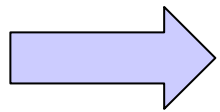
本拠地発見



味方本拠地へ戻る

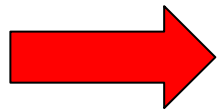
# わかったこと その(1)

ハウズは、計画を立てて行動しなければいけない。



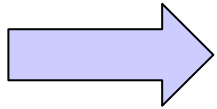
ハウズが長期的に計画を立てて行動できるように思考を作ろう。

(プランニングの技術が必要だ)



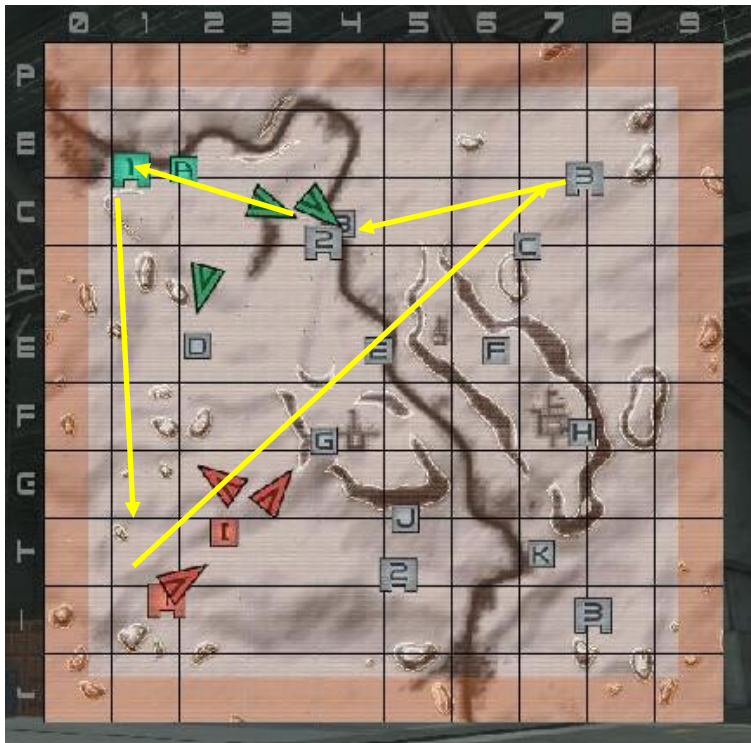
でも、それだけで十分だろうか？

# COMは、どういうふうを考えれば、うまく行動できるだろうか？



実際の戦闘がどうなっているかパターンを集めてみよう

例 1



本拠地から出発



敵の本物の本拠地がどれであるかを偵察



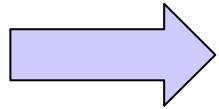
本拠地発見



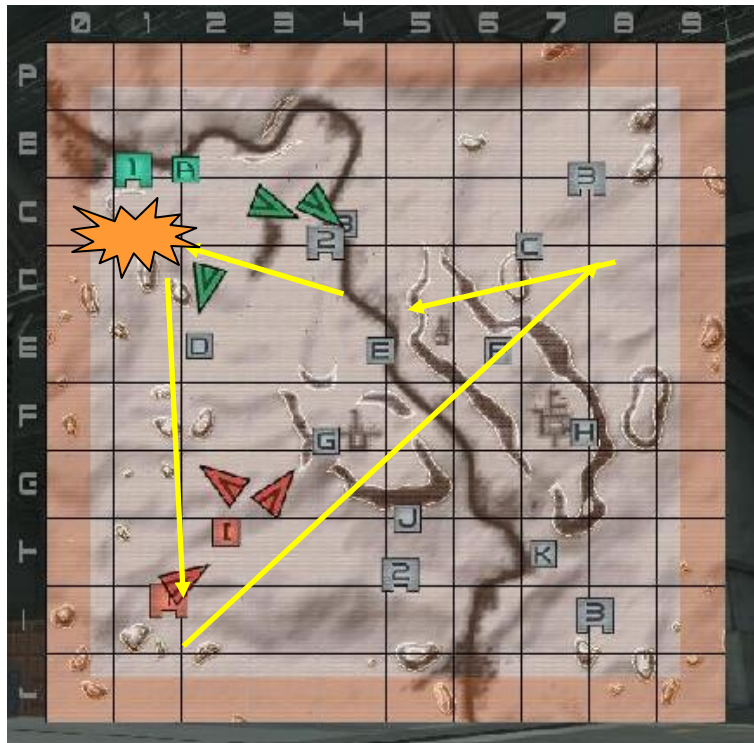
味方本拠地へ戻る



# COMは、どういうふうを考えれば、うまく行動できるだろうか？



実際の戦闘がどうなっているかパターンを集めてみよう



例 2

本拠地から出発



敵の本物の本拠地がどれであることを  
偵察



本拠地発見 **しかし、敵に発見される！**

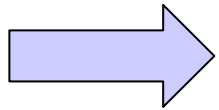


**戦闘に入る**

味方本拠地へ逃げる

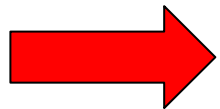
## わかったこと その(2)

ハウズは、その場の状況に応じて、  
計画を修正、変更しなければならない。



常に、周囲の状況を察知して、柔軟に計画  
を修正、変更しなければならない。

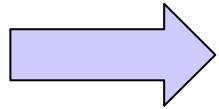
(リアルタイムプランニングの技術が必要だ)



何をどう計画しているんだろう？

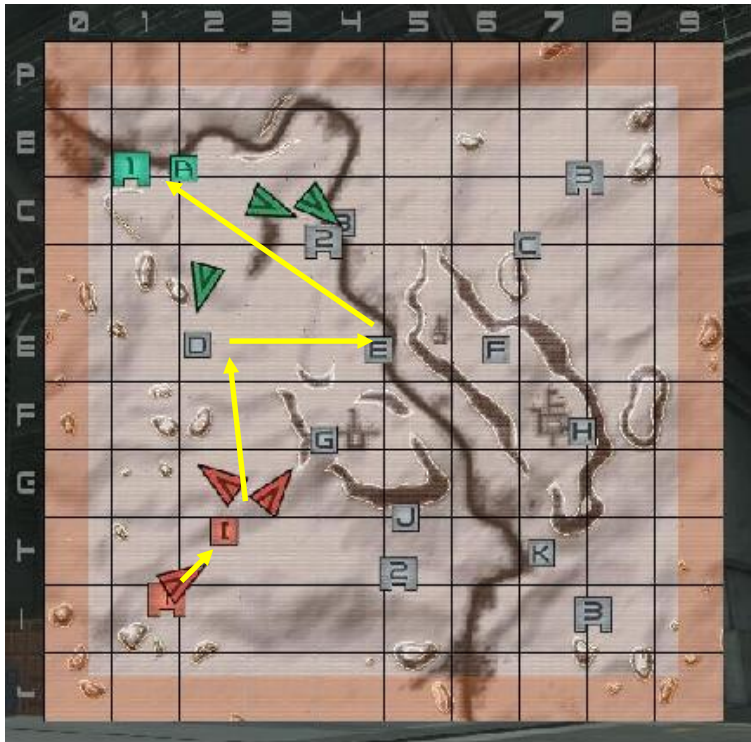


# COMは、どういうふうを考えれば、うまく行動できるだろうか？



実際の戦闘がどうなっているかパターンを集めてみよう

例



本拠地から出発



本拠地に近い通信塔を取る



2番目に近い通信塔を取る

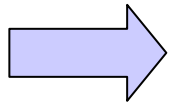


敵本拠地を占拠する

目標を一つずつ達成している！

# わかったこと その(3)

ハウズは、一つ一つゴールを達成している。



(1)ハウズは、幾つかのゴールを順番に達成するプランを立てて行動する。

(2)常に、周囲の状況を察知して、柔軟に計画を修正、変更する。

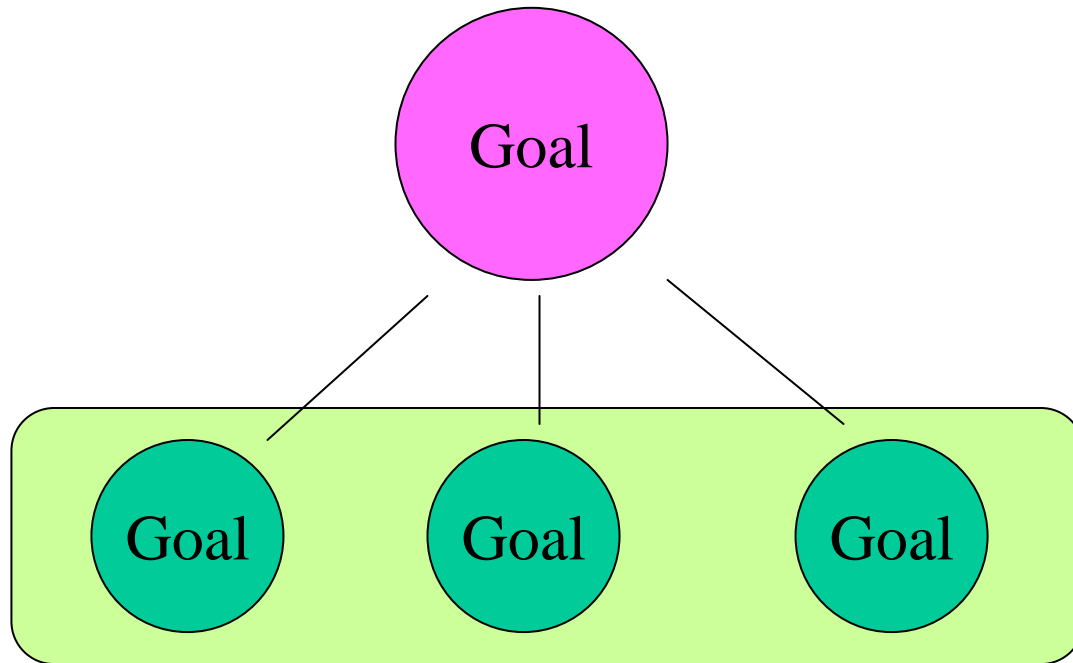
(必要な要件を抽出することが出来た。)

これに対応する、人工知能の技術は何か？

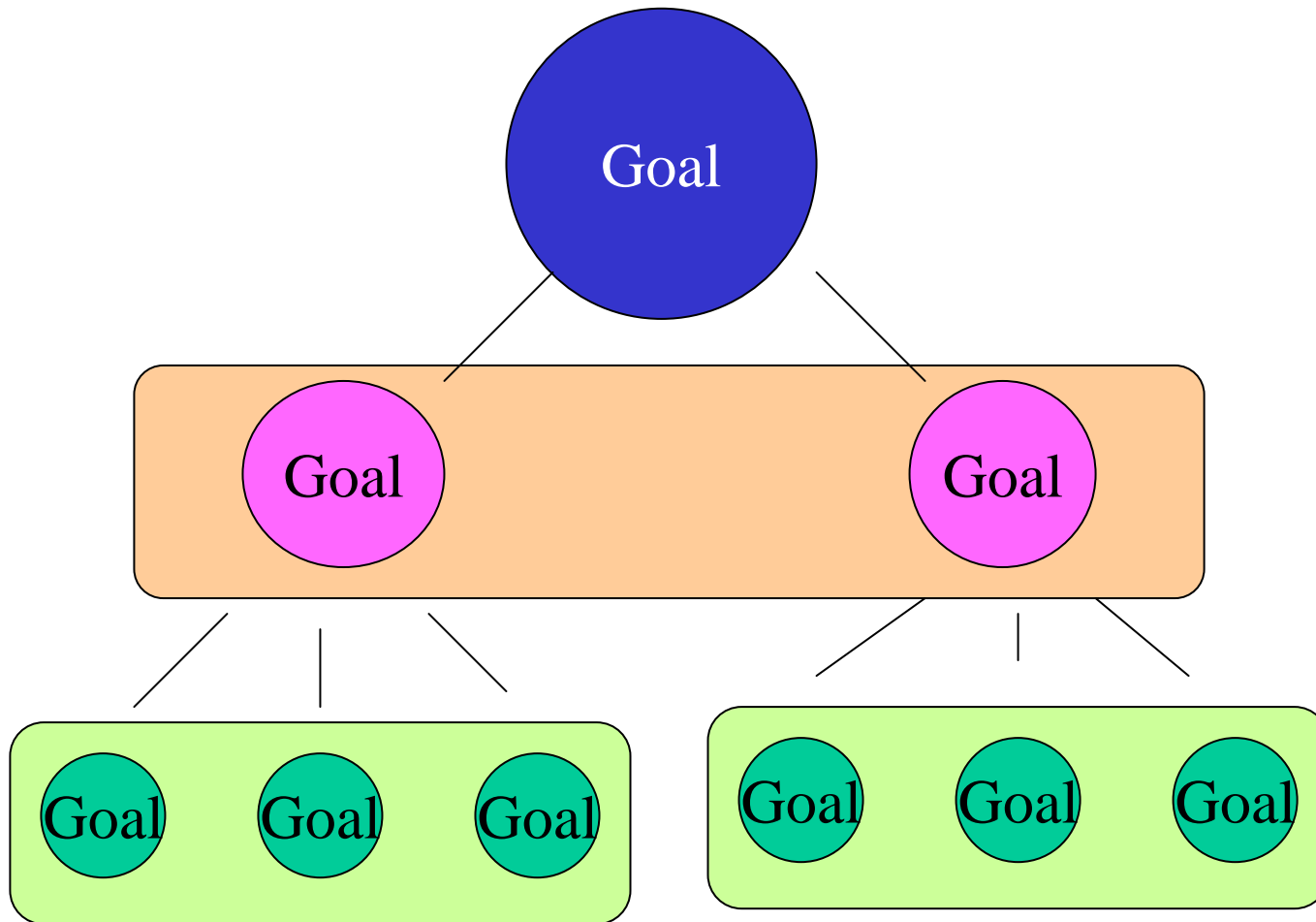
ゴール指向型プランニング

**ゴール指向型プランニングとは？**

一つのゴールはより小さなゴールから組み立てられる



ゴールはより小さなゴールから組み立てられる



# ゴール指向型プランニングの考え方

映画を見たい

映画館に行く

映画館は新宿だ

新宿駅に行く

晴れなら

新宿駅へ歩く

雨なら

電車で新宿へ

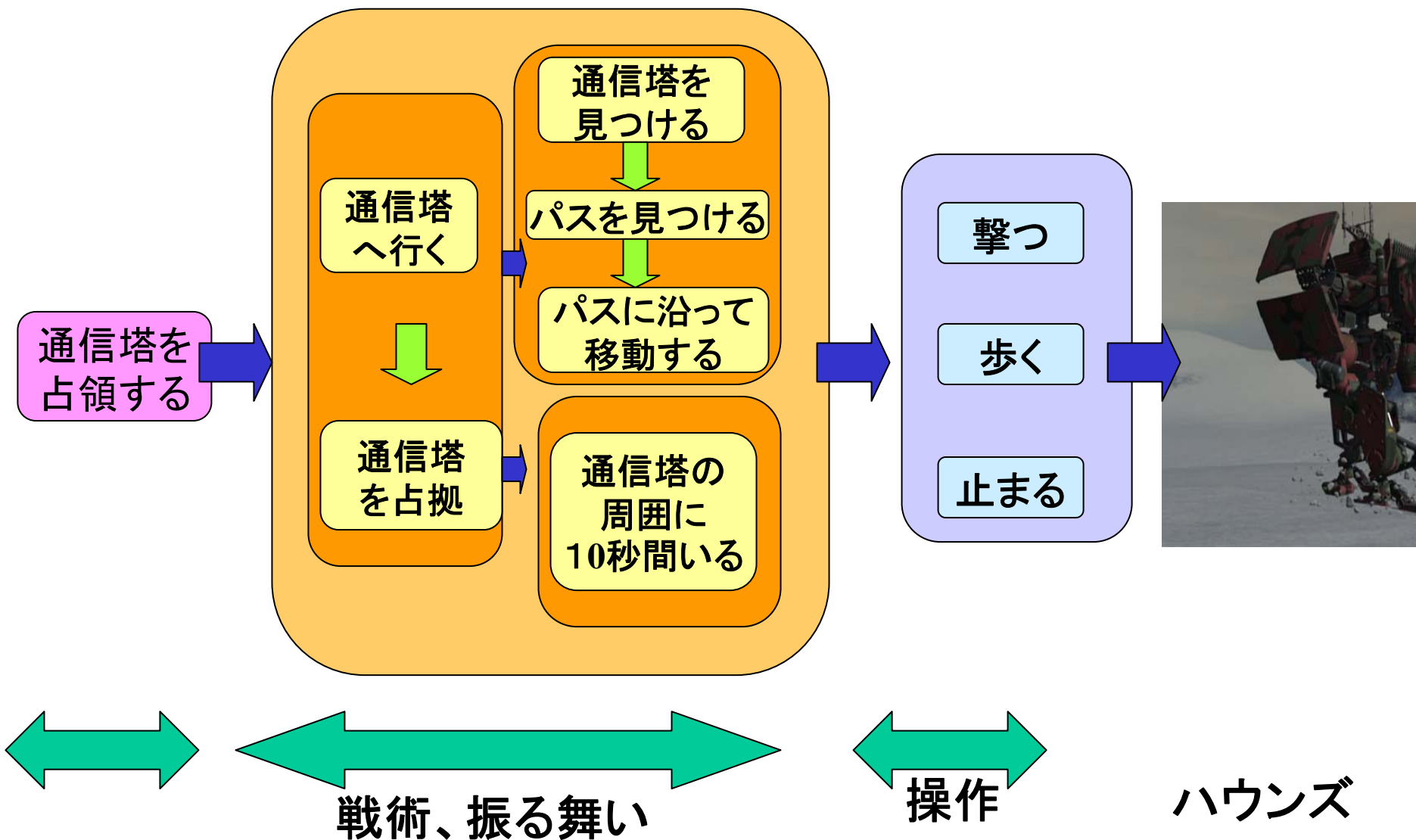
駅まで歩く

電車に乗る

映画館まで歩く

映画を見る

# クロムハウズにおける ゴール指向型プランニング



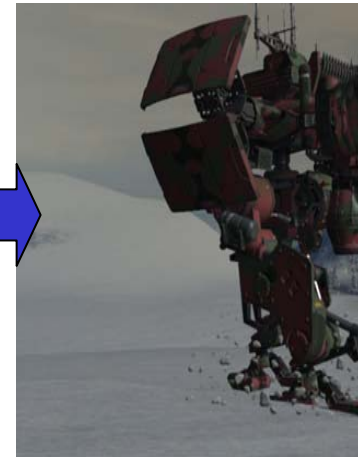
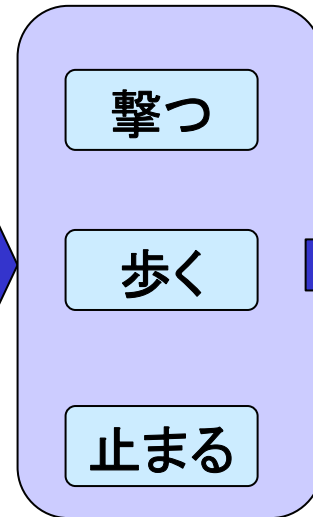
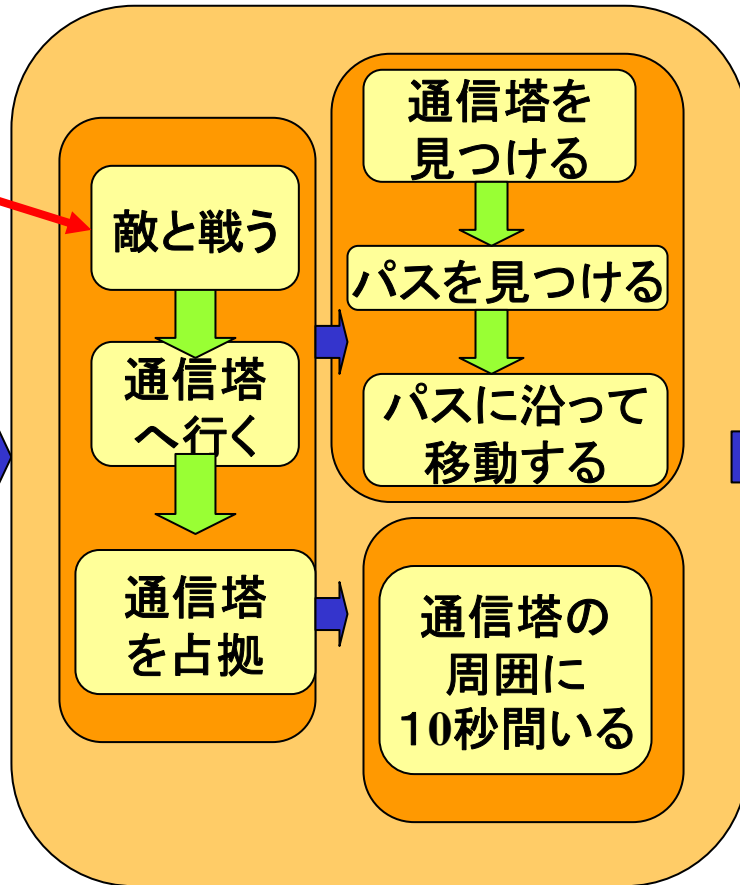
# クロムハウنزにおける

## リアルタイムゴール指向型プランニング



敵が来た!

通信塔を占領する



戦略



戦術、振る舞い

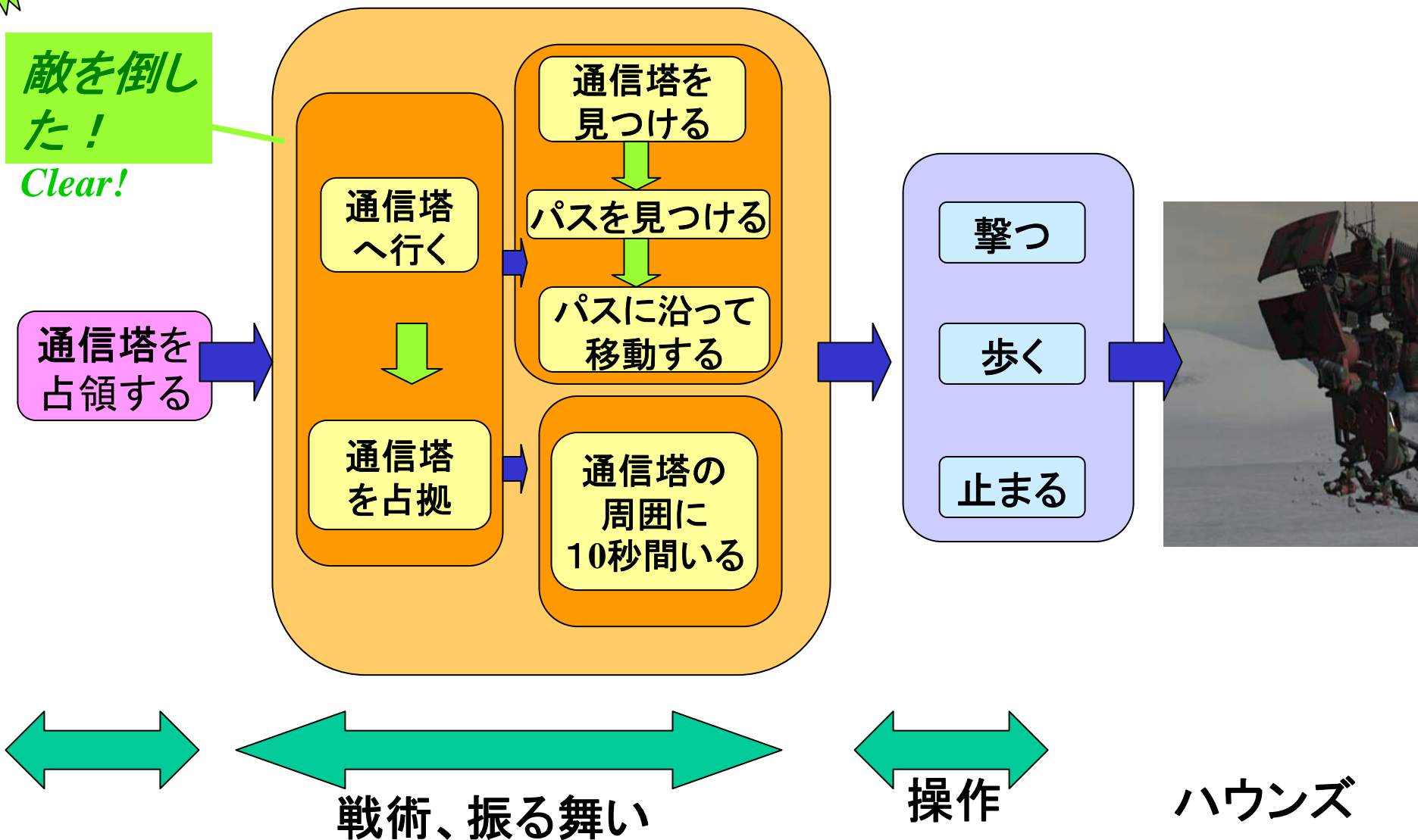


操作

ハウنز



# クロムハウنزにおける リアルタイムゴール指向型プランニング



ゴール指向型プランニングによって  
敵本拠地へ進行しつつ敵の攻撃に対処する  
様子をご覧ください。



思考がその場で状況に応じて生成(インスタンス)される

Player 1

GOAL\_THINK, 寿命 -1.000000

GOAL\_CONQUER\_COMBAS, 寿命 43.450000

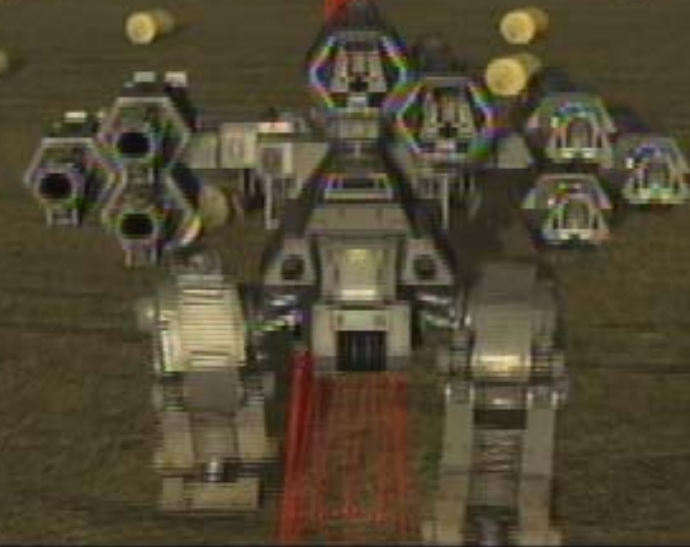
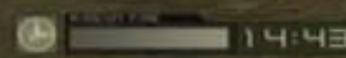
ATTACK\_AROUND\_COMBAS, 寿命 13.500000

GOAL\_STAY, 寿命 3.500000

GOTO\_POSITION, 寿命 -1.000000

FOLLOW\_PATH, 寿命 -1.000000

STAY\_HERE, 寿命 30.000000



# リアルタイムゴール指向型プランニングによって 状況に即応しつつ長期的戦略を実行するデモ



Player 1

GOAL\_THINK, 寿命 -1.000000

GOAL\_CONQUER\_BASE, 寿命 140.710783

GOAL\_WALK\_AROUND, 寿命 140.600851

GOTO\_POSITION, 寿命 -1.000000

FOLLOW\_PATH, 寿命 -1.000000

ATTACK\_AROUND\_BASE, 寿命 141.757599



1234  
9998



12:23



# リアルタイムゴール指向型プランニングによって 状況に即応しつつ長期的戦略を実行するデモ

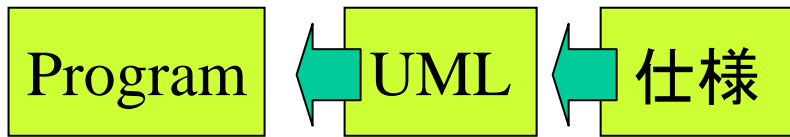


このようにして、一つ一つの戦略を実装して行く。

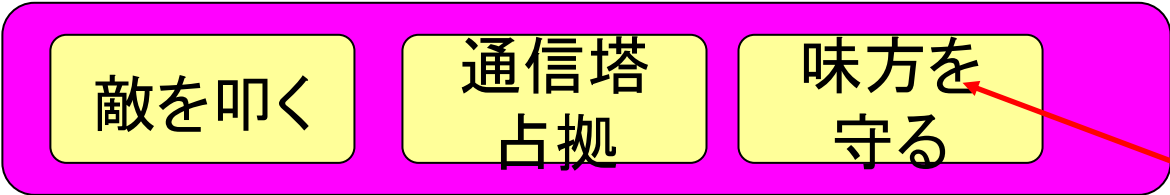
では、実際に、誰が、どのようにして作って行くのか？

# ゴール指向型AIの開発工程

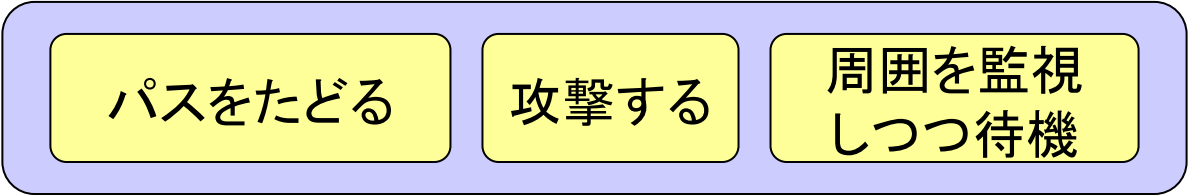
(パターンランゲージによる漸近的成長)



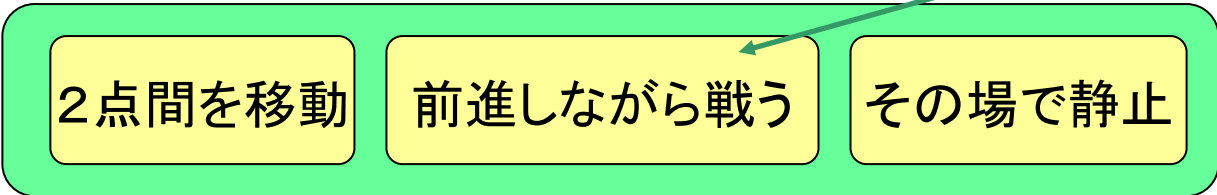
戦略層



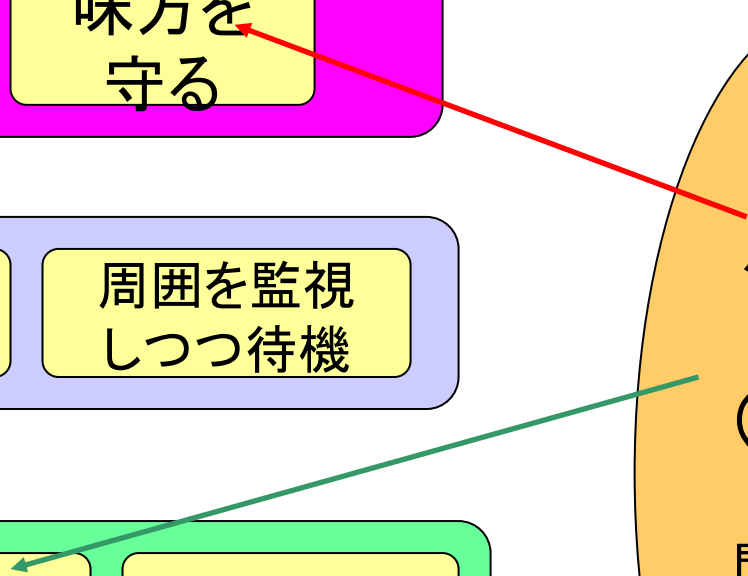
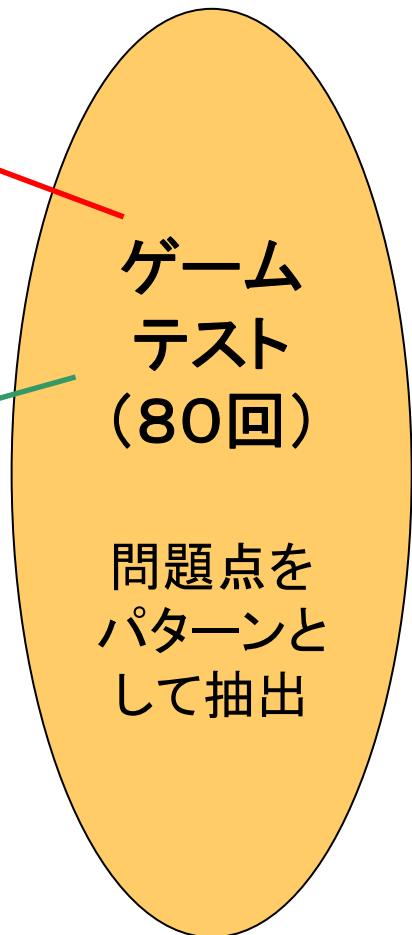
戦術層



振る舞い層



操作層



# プログラマーのための実装工程

## ゴール指向型プログラム構造

(入れ子構造)



- Activate ... 初期セッティング
- Process ... アクティブな間の行動
- Terminate ... 終了処理

(関数の内容を全てスクリプトで記述)

インスタンス



# 最終的なゴール総合図

## 戦略層

敵を叩く

通信塔  
占拠

味方を  
守る

本拠地  
防衛

敵本拠地  
破壊

味方を  
助ける

巡回  
する

敵基地  
偵察

## 戦術層

パスを  
たどる

近付く

攻撃  
する

ある地点へ  
行く

合流  
する

巡回  
する

逃げる

## 振る舞い層

2点間を  
移動

歩く、一度  
止まる、歩く

静止  
する

後退  
する

前進  
する

敵側面  
へ移動

## 操作層

歩く

撃つ

止まる

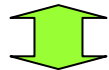
# ゴール指向型プランニングのまとめ

## リアルタイム ゴール指向型プランニングとは？

- (1) 「ゴール」と呼ばれる目標を、戦略に沿って順番に組み上げることがをプランニングという。
- (2) 周囲の状況の変化に応じてプランニングをすることをリアルタイムプランニングという。

## 何が可能になったか？

- (1) 長い時間に渡る一連の動作の制御



- (2) 短い時間における状況の変化に適応

# リアルタイム ゴール指向型プランニング の解説に対する

## 質疑応答

# クロムハウズで 自律的エージェントを実現することを考える

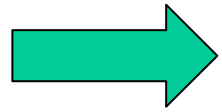
自分で判断して、自分で行動して、  
プレイヤーと戦え

(1) 考える - リアルタイムのゴール指向型プランニング

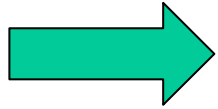
(2) 歩く - **どんな技術が必要だろう？**

COMを歩かす

# 自律型エージェントに必要な移動能力



フィールドを自由に歩き回ることが出来る



フィールドの、どんな2点の間も、  
自分でパスを見つけて移動することができる

リアルタイムパス検索



マップをパス検索することが出来る  
データの形にしておく必要がある

# パス検索することが出来るデータの形

## (1) ウェイポイント

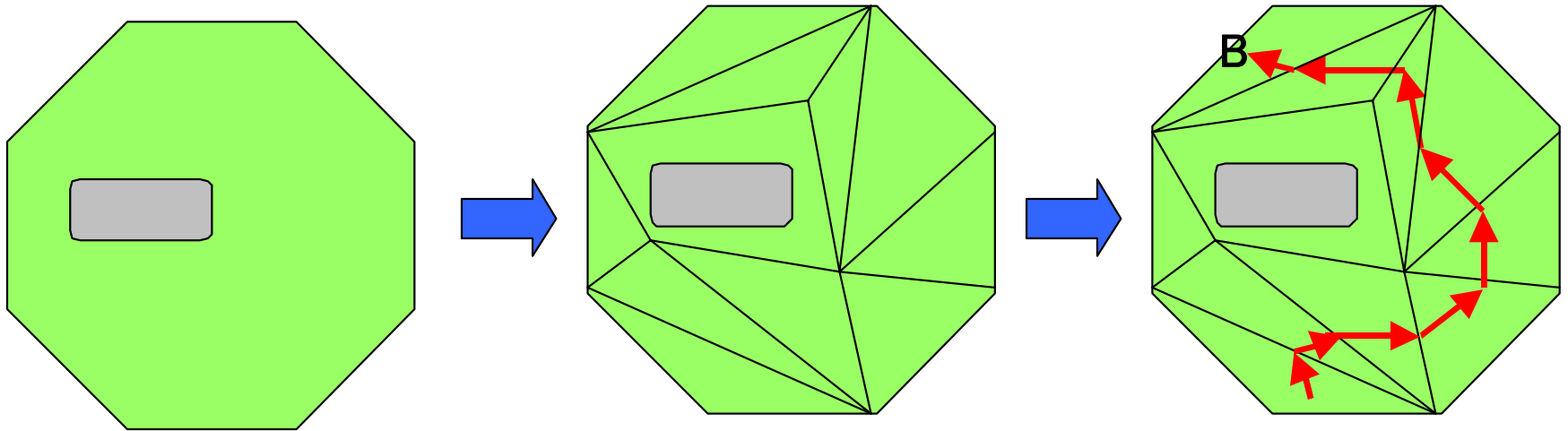
- ➡ ウェイポイントの数が膨大。  
どこへ置くかで、ハウズズの行動が極端に制限される。

## (2) ナビゲーションメッシュ

- ➡ 容量が大きい。  
メッシュによって地形の情報と形状を  
全て取り込むことが出来る。

# Navigation Mesh 法とは？

マップを凸多角形で埋めてキャラクターを移動させる方法



## 手順

- (1) マップを障害物を含まない三角形に分割する(データを用意する)
- (2) その三角形の情報から、パスを検索する(ゲーム内リアルタイム)。
- (3) 検索したパスに従って移動する(ゲーム内リアルタイム)。



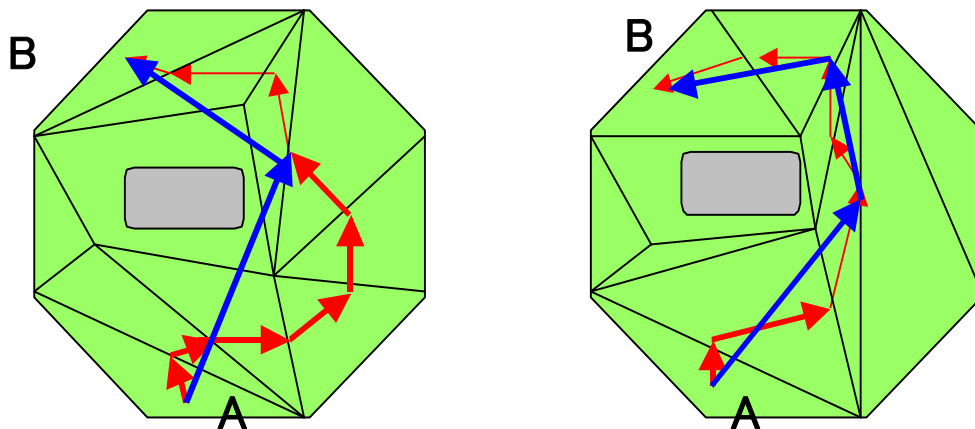
# ナビゲーションメッシュの利点

## 適用できる場合

キャラクターが床から離れない場合にのみ有効な方法

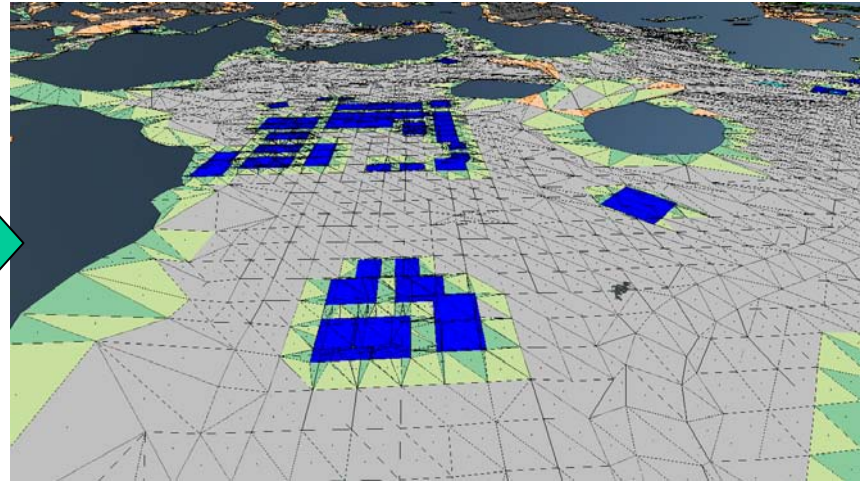
## 利点

- (1) マップ上のどの点からどの点でも、ゲーム中にリアルタイムに経路を探索し移動できる。
- (2) 平面的につながっていれば、どんな地形にも対応できる。
- (3) マップの情報をデータに埋め込みやすい。



# クロムハウズにおけるナビゲーションメッシュ

- (1) 30000 – 80000 メッシュ
- (2) 複雑な地形にも対応
- (3) メッシュを当りモデルから **自動生成**
- (4) マルチ分解能
- (5) 地形表面の性質(雪、砂など)の情報が埋め込まれたメッシュ



山岳、街、湖など、80に及ぶバリエーションに富んだマップに対し、**単一のデータ形式、アルゴリズム**で対応することが出来た(汎用性)。

# ナビゲーシヨンメツシュ工工程

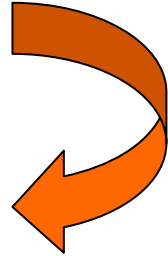
あたりモデル

第1次中間データ

第2次中間データ

第3次中間データ

ナビゲーシヨン  
メツシュ



ポリゴンを削減する (3D Studio Max)

# ナビゲーションメッシュ工程

あたりモデル

第1次中間データ

第2次中間データ

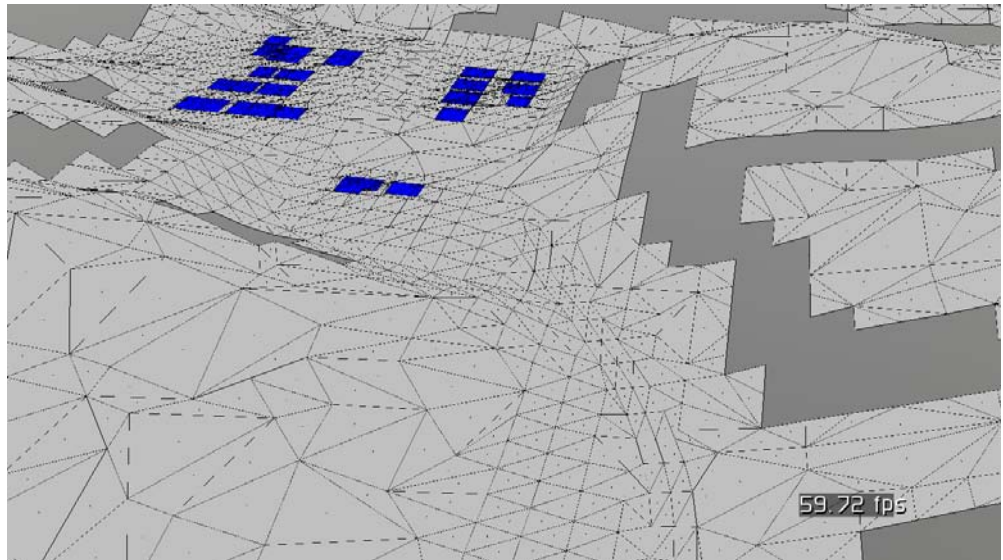
第3次中間データ

ナビゲーション  
メッシュ

デバッグ



配置オブジェクトとのあたりを取る(3D Studio Max plugin)



# ナビゲーションメッシュ工程

あたりモデル

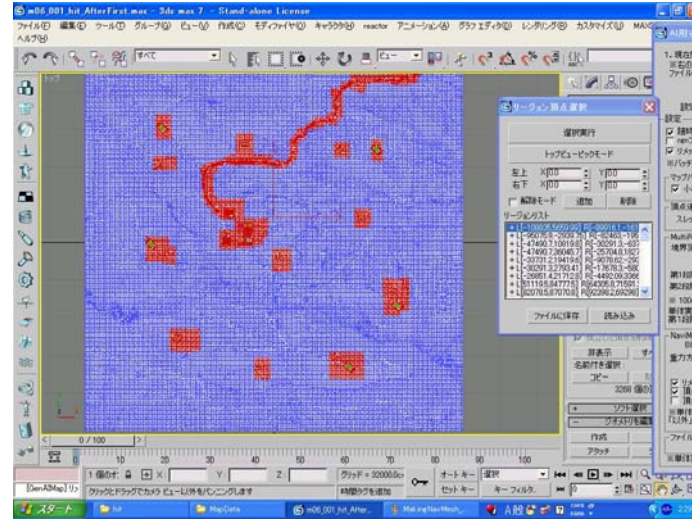
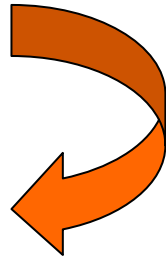
第1次中間データ

第2次中間データ

第3次中間データ

ナビゲーション  
メッシュ

デバッグ



ポリゴンを削減する(3D Studio Max)

# ナビゲーションメッシュ工程

あたりモデル

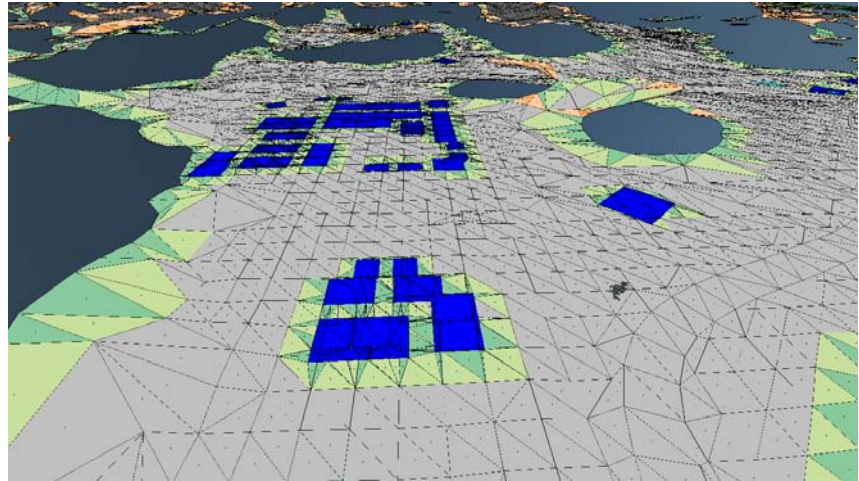
第1次中間データ

第2次中間データ

第3次中間データ

ナビゲーション  
メッシュ

デバッグ



プログラムで地形と表面の情報を埋め込む



# ナビゲーションメッシュ工程

(1) あたりモデルと比較する。

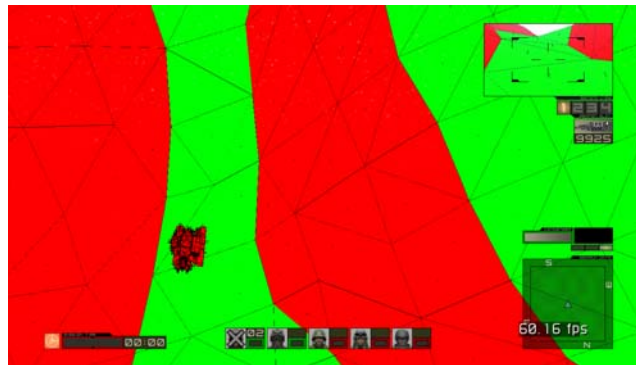
あたりモデル



第1次中間データ

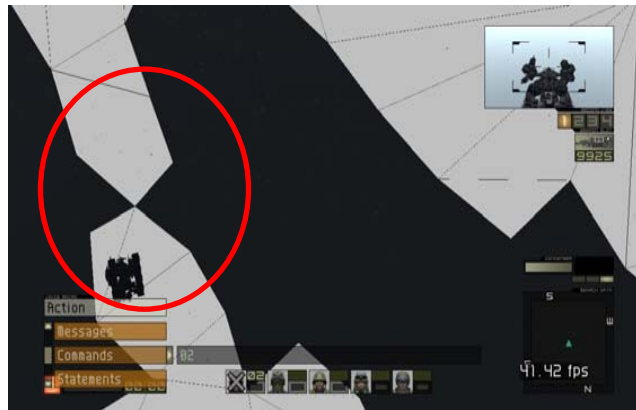
第2次中間データ

第3次中間データ



ナビゲーション  
メッシュ

デバッグ



# ナビゲーションメッシュ工程

あたりモデル

第1次中間データ

第2次中間データ

第3次中間データ

ナビゲーション  
メッシュ

デバッグ

(2) 実機でテストを行う。

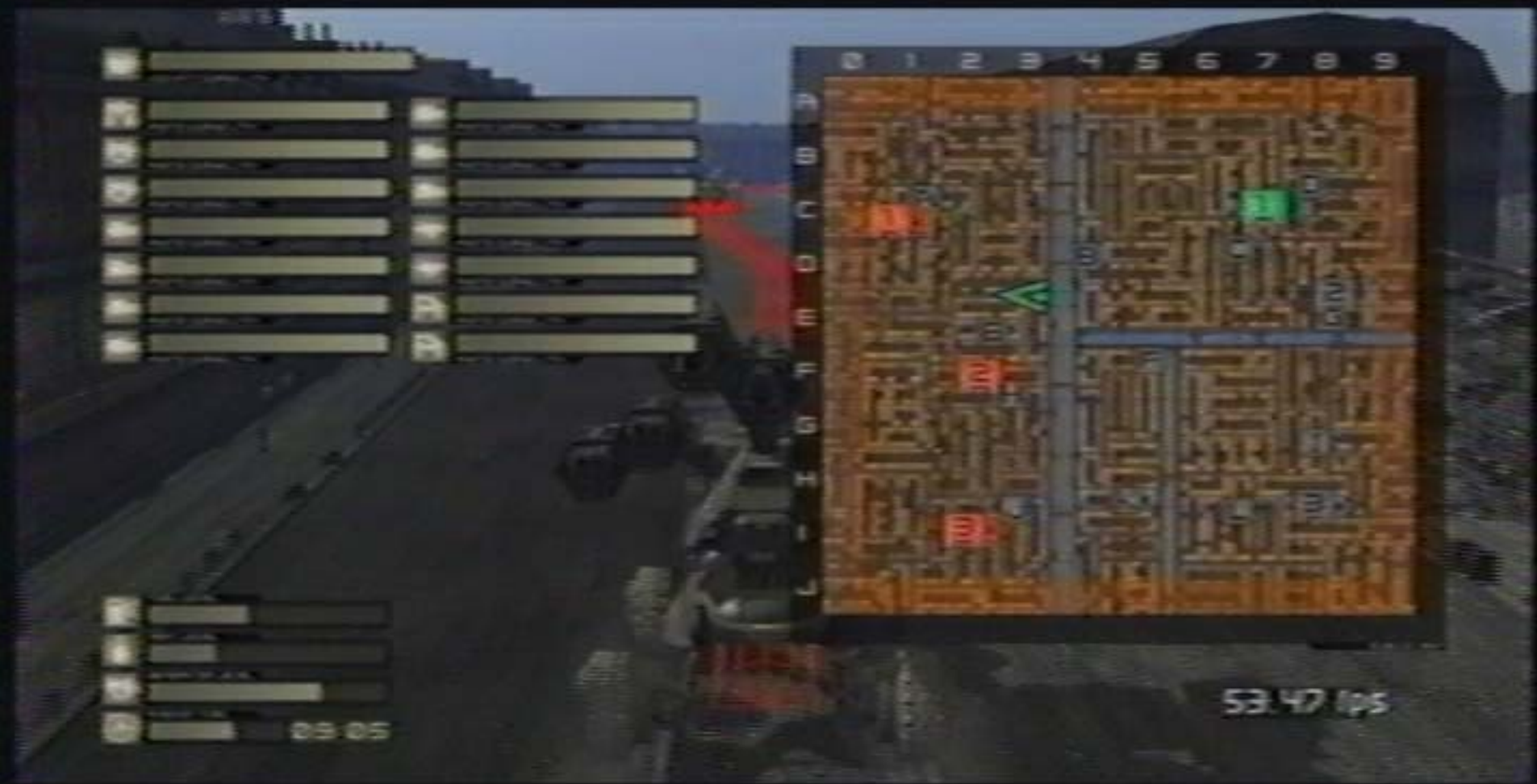


各、本拠地同士を結ぶパスと、  
コンパスを巡回するパスをチェック

デバッガーさん

80 x 20 x 2 = 3200 以上のチェック

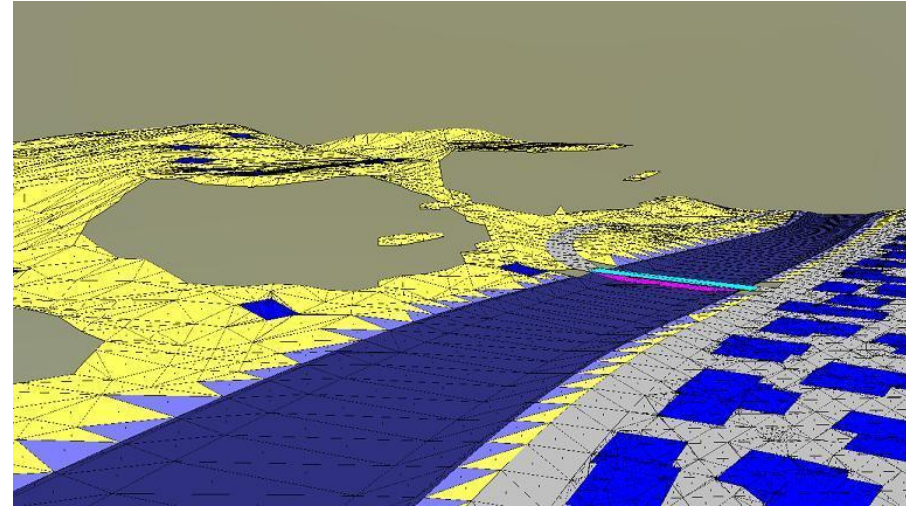




53.47 fps



# 情報が埋め込まれたナビゲーションメッシュ



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

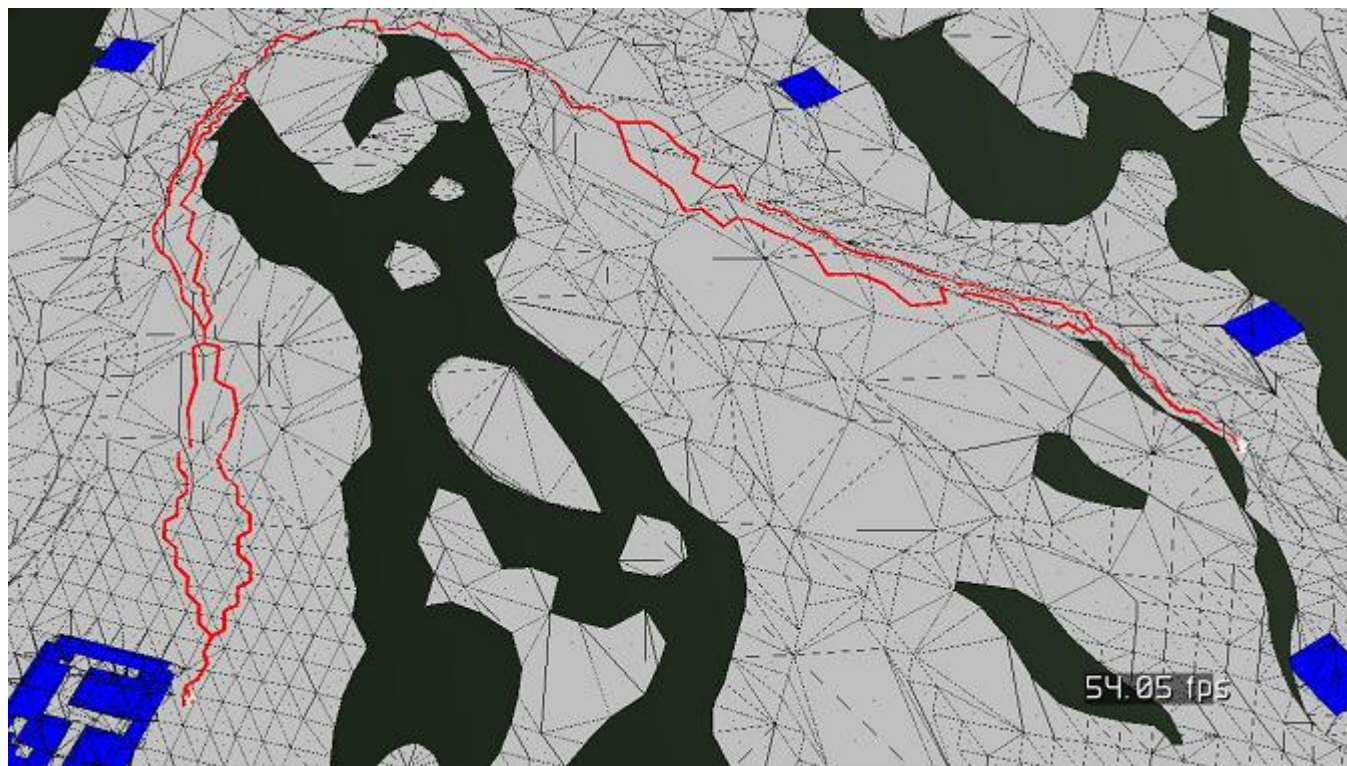
➡ 最短時間の経路を導く

ハウন্ズが地表効果を考慮して移動する

(2) 障害物が破壊されたら、メッシュのデータを更新する

➡ ハウন্ズが状況の変化に対応して移動する

ナビゲーションメッシュ上を  
A\*アルゴリズムによって  
ハウズが移動するデモをご覧ください。







# クロムハウズにおける リアルタイムパス検索 のまとめ

## ナビゲーションメッシュ法とは？

- (1) 地形を三角形に分けてパスを検索して移動する方法
- (2) ハウズが床から離れない場合に有効

## 何が可能になったか？

- (1) フィールド内の完全に自由な移動
- (2) 地表の効果や、オブジェクトの破壊など、  
環境の変化を考慮に入れて賢く行動する。
- (3) (1)(2)を全てのマップの形状に対応。

# クロムハウズで 自律的エージェントを実現する

自分で判断して、自分で行動して、  
プレイヤーを倒す

(1) 考える - リアルタイムのゴール指向型プランニング

(2) 歩く - リアルタイムのナビゲーションメッシュ上の  
A\* アルゴリズムによるパス検索

# 自律型エージェントへ

リアルタイム ゴール指向型プランニング

×

リアルタイム パス検索



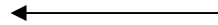
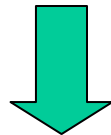
自律型エージェント ?

# 自律型エージェントへ

リアルタイム ゴール指向型プランニング

×

リアルタイム パス検索



意思決定

自律型エージェント ?



# COM の意志決定

# COMの意思決定過程

周囲の状況を反映して意思決定する

評価関数法

どれぐらい状況に適しているか、点数をつけて比較する方法

意志決定機構

選択

戦略

本拠地  
占拠

敵を叩け

通信塔  
占拠

本拠地  
防衛

味方機を  
助けよ

プランニング

# エージェントが意思決定をする仕組み

その戦略を達成することで得られる **見返り(S; 重要度)** と、それを達成するための **リスク(R; 危険度)** の兼ね合い

$$\text{実行評価値(E)} = S * (1 - R)$$



本拠地  
破壊

通信塔  
占拠

1 E = 24

C E = 20

2 E = 12

D E = 32

3 E = 6

F E = 3

J E = 21

# 通信塔の「重要度 S」

3つのファクターから決まる。

- (1) 味方司令部との関係 (通信塔- 敵司令部)
- (2) 敵司令部との関係 (通信塔- 味方司令部)
- (3) 通信塔同士の関係 (通信塔- Combustion)

$$S = W\_EBase * (\sum F(L\_EBase, L\_MapScale)) * Est\_Base\_NonConstFactor \\ + W\_SBase * F(L\_SBase, L\_MapScale) \\ + W\_InComNet * Est\_InComNet$$

$$W\_EBase + W\_SBase + W\_InComNet = 1$$

$$Est\_InComNet = W\_static * Est\_Static\_Combustion + W\_dynamic * Est\_Dynamic\_Combustion$$

$$Est\_Static\_Combustion = (Connectable\_Number - Connected\_Number) / Max\_Connectable\_Number$$

$$Est\_Dynamic\_Combustion = Connected\_Number / Max\_Connectable\_Number$$

$$W\_static + W\_dynamic = 1$$

...

# 通信塔の「重要度 S」

3つのファクターから決まる。

- (1) 味方司令部との関係 (通信塔- 敵司令部)
- (2) 敵司令部との関係 (通信塔- 味方司令部)
- (3) 通信塔同士の関係 (通信塔- Combus)

$$S = W_1 * \text{味方司令部との距離による関数} + \\ W_2 * \text{敵司令部との距離による関数} + \\ W_3 * \text{隣の通信塔の占拠状態からなる関数}$$

W ... 重み

# 通信塔の「危険度 R」

3つのファクターによる。

- (1) 敵ハウズが通信塔からどれぐらいの距離にいるか。
- (2) ザコ敵がどれぐらいの距離にいるか。
- (3) 味方ハウズが通信塔からどれぐらいの距離にいる

$$R = W_1 * \text{敵ハウズの通信塔との距離による関数} \\ + W_2 * \text{ザコ敵と通信塔の距離による関数} \\ + W_3 * \text{味方ハウズ通信塔との距離の関数}$$

W ... 重み

パラメーターと関数の形を調整する

意志決定の形やハウズの個性が決定

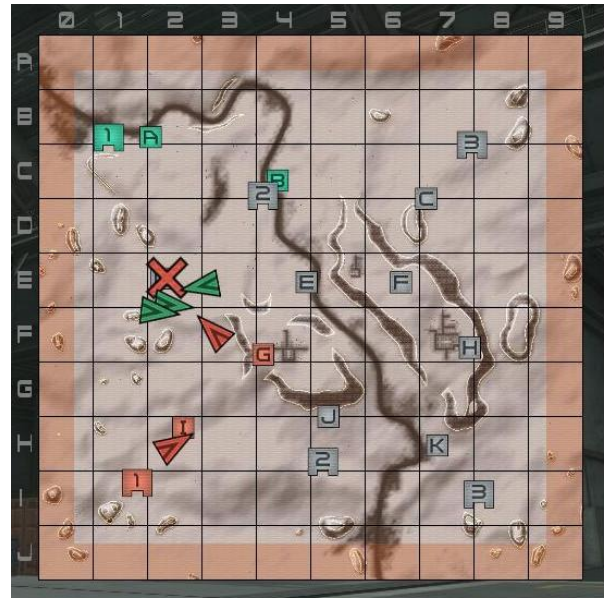
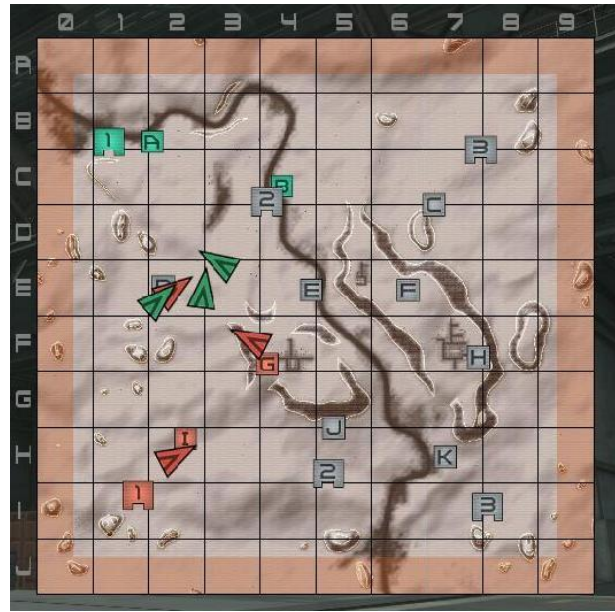
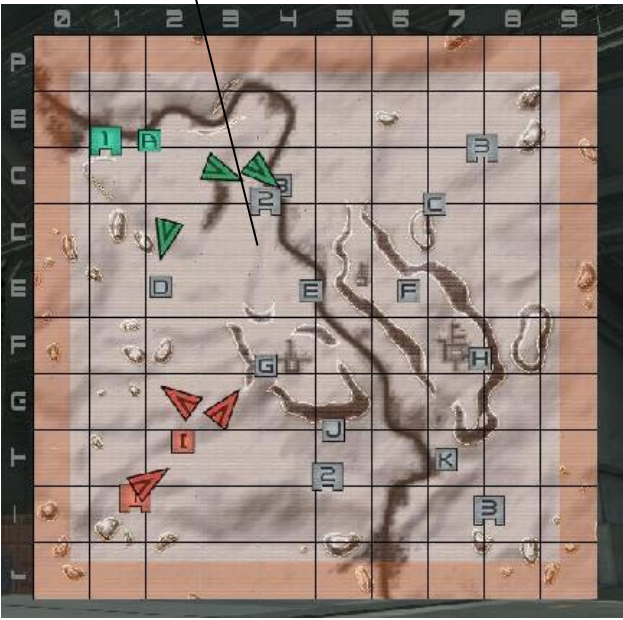
テストをくり返しなら

計 100 近くのパラメーターを調整

# 戦略の流れ

戦場の俯瞰図。緑色がCOM、赤がプレイヤー。

COM1

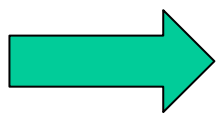


2人のエージェントが、通信塔Bを取りに向かい、残りの一体はDへ向かう。

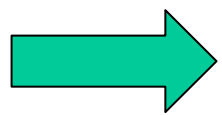
通信塔Bを占拠した後、残りの1体を援護するため、Dへ向かう。

プレイヤーを攻撃して、破壊した。

通信塔  
占拠



味方機を  
助ける



敵を叩け

Strategy of COM1

Strategy of COM1

Strategy of COM1

# 意思決定機構のデモ

COMが自分で判断をする様子  
をご覧ください。



Player 3

EVAL\_ATTACK\_TARGET, 0.000000  
EVAL\_CONQUER\_COMBAS, 0.186761  
EVAL\_CONQUER\_BASE, 0.169044  
EVAL\_PROTECT\_TARGET, 0.148441  
EVAL\_RESQUE\_FRIEND, 0.000000  
EVAL\_DEFEND\_MY\_BASE, 0.144628  
EVAL\_SCOUT\_BASE, 0.115344  
EVAL\_TRAVEL, 0.000000  
EVAL\_CONQUER\_BASE\_WITH\_COMBAS, 0.000000  
EVAL\_CONQUER\_MORE\_COMBAS, 0.000000  
EVAL\_DEFEND\_MY\_BASE\_WITH\_COMBAS, 0.000000  
EVAL\_SCOUT\_BASE\_WITH\_COMBAS, 0.000000  
EVAL\_CONQUER\_BASE\_WITH\_COMBAS\_TO, 0.000000

Player 3

GOAL\_THINK, 寿命 -1.000000  
GOAL\_CONQUER\_COMBAS, 寿命 41.062820  
ATTACK\_AROUND\_COMBAS, 寿命 11.122226  
GOAL\_STAY, 寿命 1.129863  
GOTO\_POSITION, 寿命 -1.000000  
FOLLOW\_PATH, 寿命 -1.000000



54.31 fps

# 自律型エージェントの実現

リアルタイム ゴール指向型プランニング

×

リアルタイム パス検索

×

意思決定機構



自律型エージェント 完成

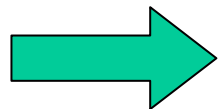
# クロムハウズにおける 自律型エージェントの実現まとめ

## 自律型エージェントの実現

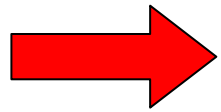
- (1) クロムハウズでは、自律型エージェントを実現するために、3つの技術が必要であることがわかった。

リアルタイム ゴール指向型プランニング  
リアルタイム パス検索  
意志決定

- (2) 3つの技術をクロムハウズのゲームに応用し、自律型エージェントを作った。
- (3) それは、当初の目的どおり、「自分で考え、自分で移動する」エージェントとして動作した



これで十分だろうか？



問題： 複数の機体を  
動かしてみるとまとまりがない

人間のチームのように、互いの連携と取って  
チームとして行動させたい

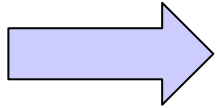


チーム全体を統制して行動をさせる、  
司令官のようなAIが必要

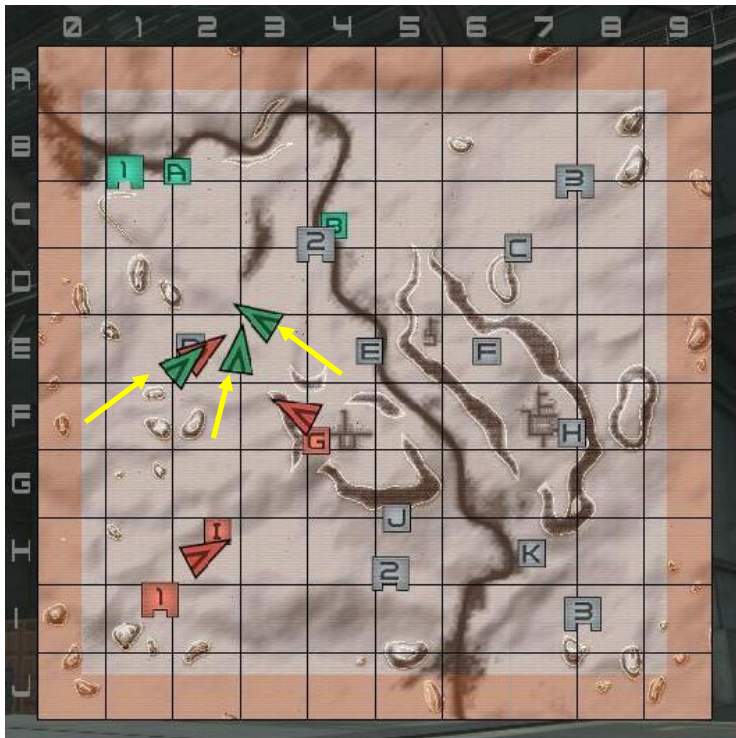
チームAIを導入する

チームAIを作る

# 例えば、チームとして こういうことをさせたい



実際の戦闘がどうなっているかパターンを集めてみよう



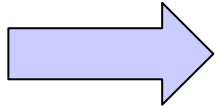
一機だけの戦闘で戦局が  
変わることはない。



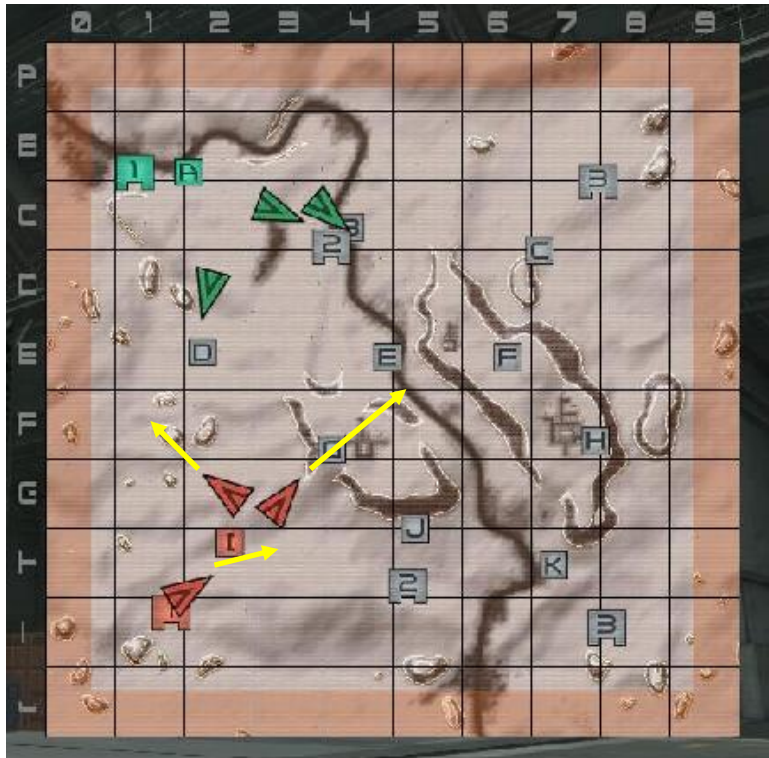
敵をやっつける時は、なるべく  
多 vs 1 になるようにする

戦力を集中させたい

# 例えば、チームとして こういうことをさせたい



実際の戦闘がどうなっているかパターンを集めてみよう



勝負が決まり始める後半では、  
勝つための方針がばらばらに  
ならないようにしたい

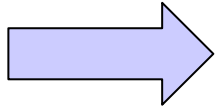


ゲーム後半では、チームAIが  
方針を決定する

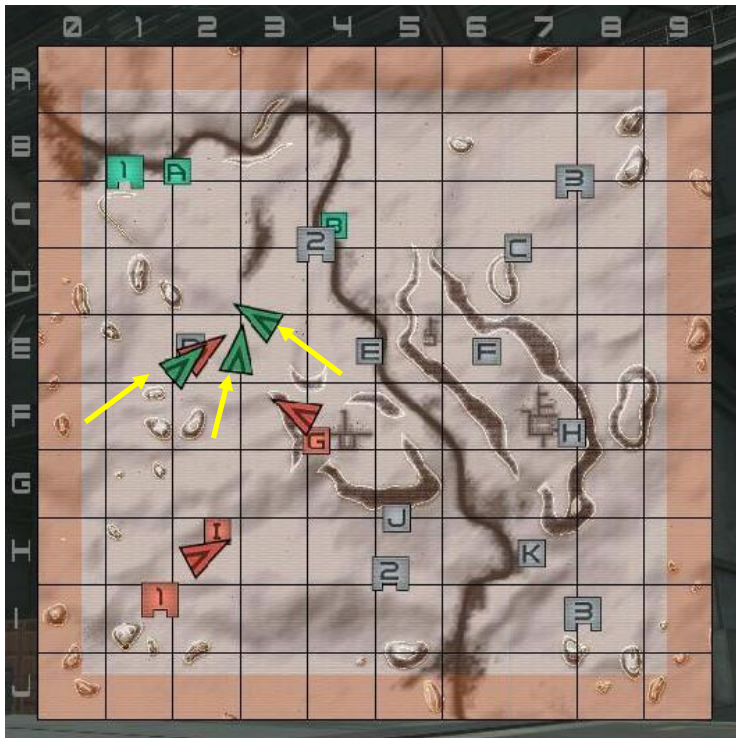
ゲーム後半では、方針を統一



# 例えば、チームとして こういうことをさせたい



実際の戦闘がどうなっているかパターンを集めてみよう



敵基地を落とすのは、  
火力が必要



1体で行っても、火力が足りない上に  
敵が防衛している



本拠地は多数の機体で攻めたい

勝負をかけるタイミングを  
あわせたい



# チームAIの構造

4つの戦略を持ち、ゲーム全体の状況を反映する  
評価関数によって、一つの戦略を決定する。

(評価関数による意思決定 = 個体の意志決定と同じ方法)

チームAI 意志決定機構



敵殲滅

本拠地  
防衛

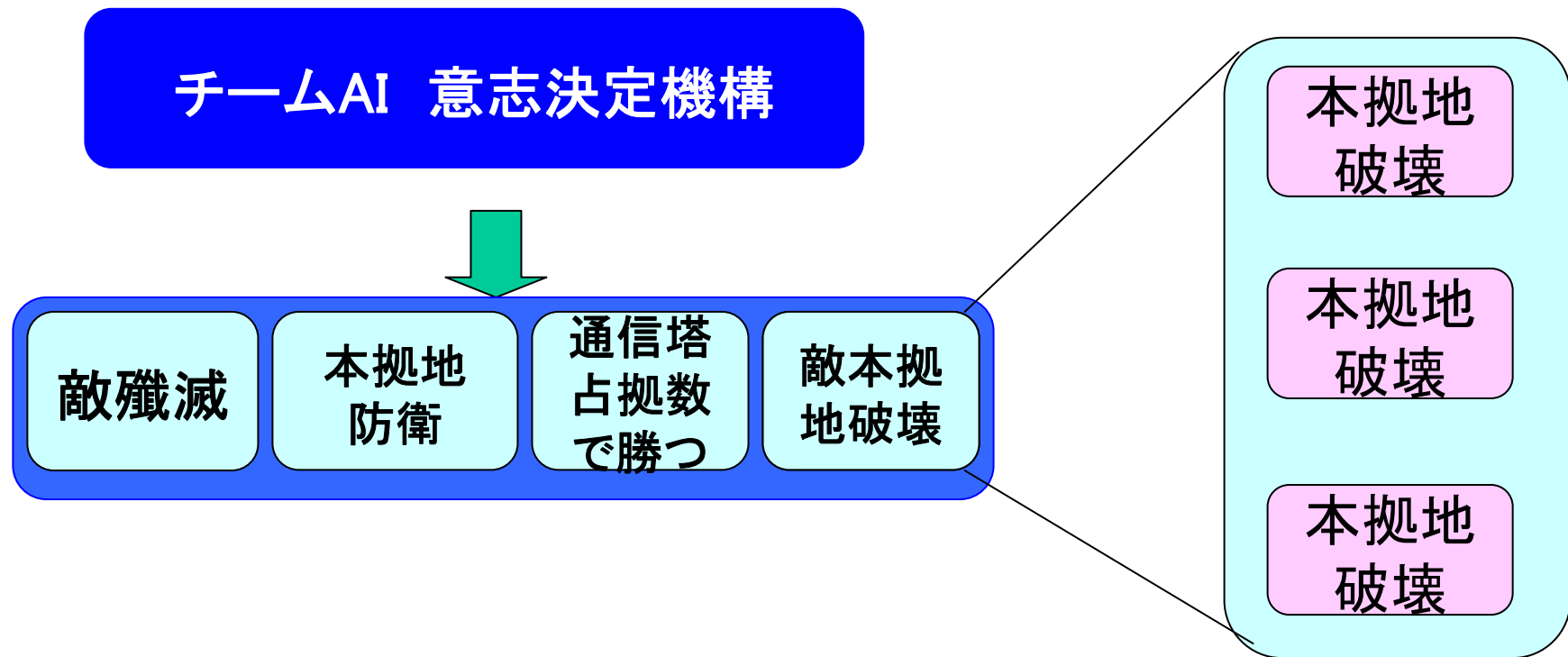
通信塔  
占拠数  
で勝つ

敵本拠地  
破壊

チームとしての戦略  
(=勝利条件と同じ)

# チームAIの構造

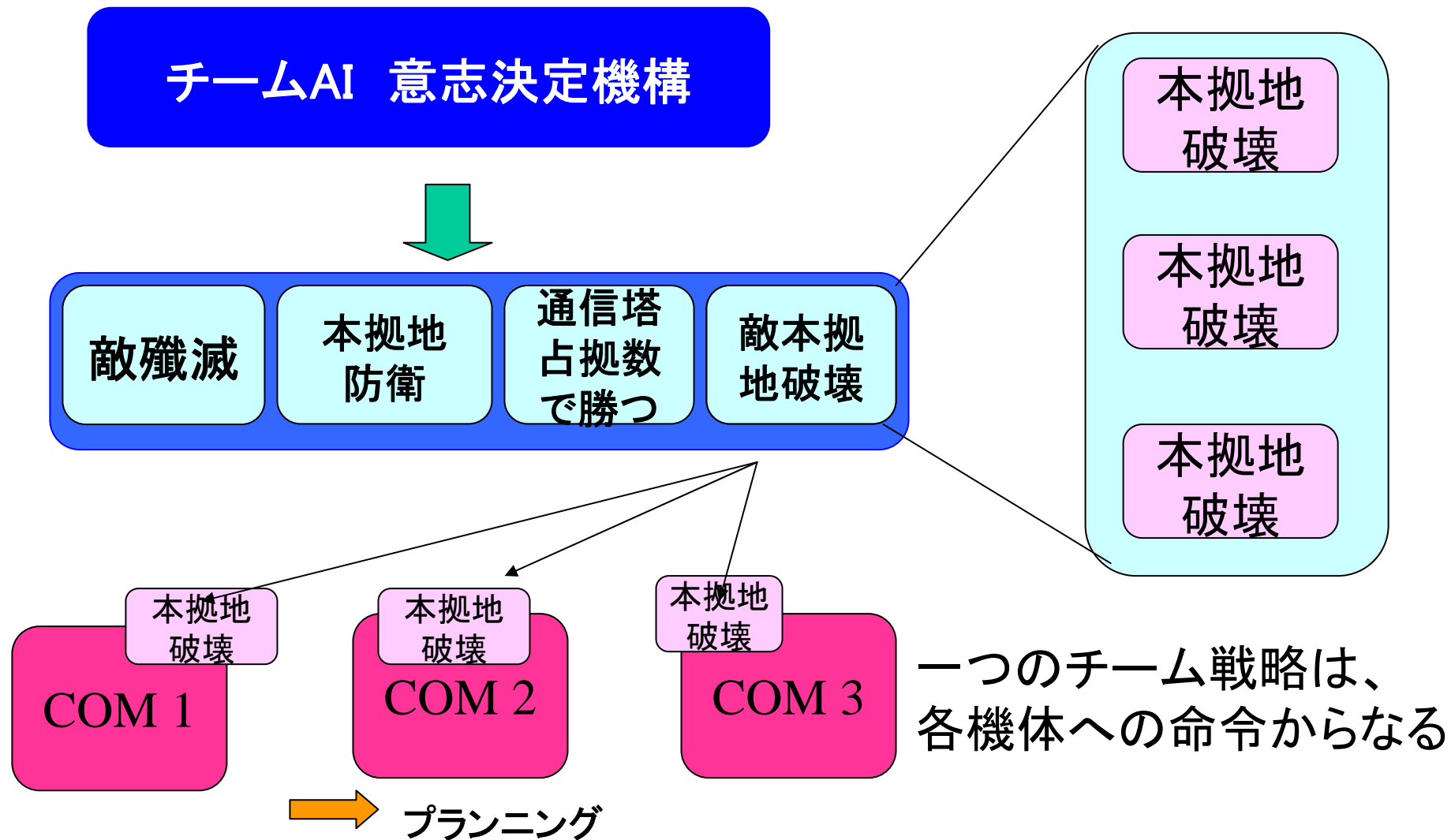
4つの戦略を持ち、評価関数によって、一つの戦略を決定する。  
(評価関数による意思決定 = 個体の意志決定と同じ方法)



一つのチーム戦略は、  
各機体への命令からなる

# チームAIの構造 = ゴール指向型の拡張

COMのゴール指向プランニングの上に、チームAIを積み上げる

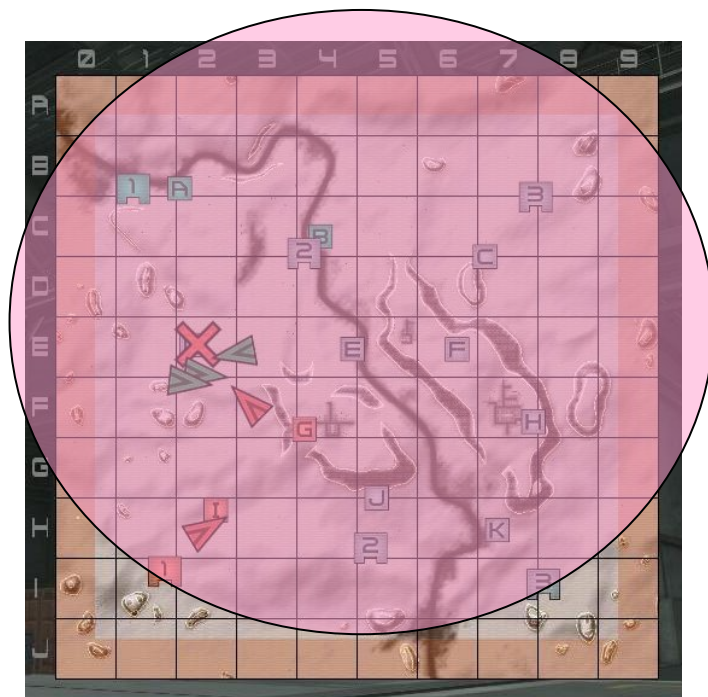
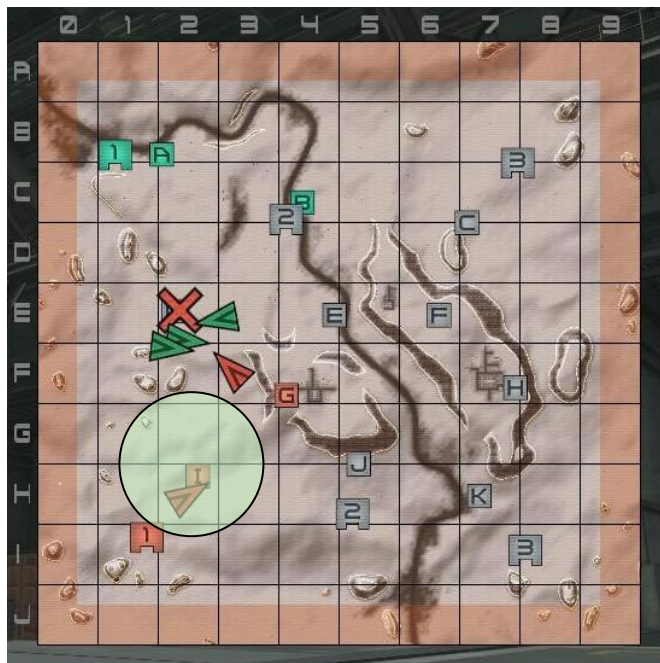


観察すると、チームAIの判断がある時は正しく、  
個体AIの方の判断がある時は正しい

何故か？

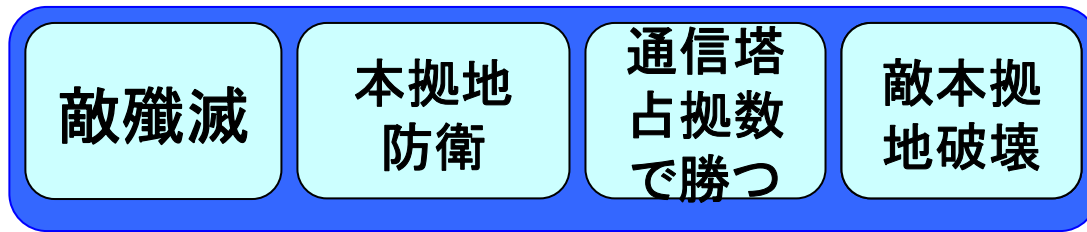
個体はハウズは、自分の廻りの局所的な情報を元に判断  
チームAIは、戦局全体の情報を元に判断

➡ 両方の判断を比較して正しい方を選択するべきだ



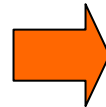
# チームAI の意思決定と COMの判断 を比較して、最終的に決定する

チームAI 意志決定機構



本拠地破壊

COM 2

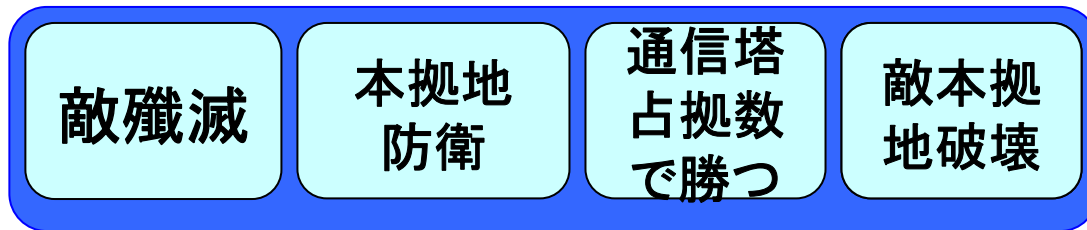


通信塔占拠

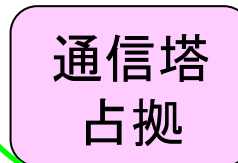
# チームAI の意思決定と COMの判断 を比較して、最終的に決定する

チームAI 意志決定機構

チームAIとCOMの  
ゴールの評価値を  
比較して高い方を選択する。



実行評価値 : 76

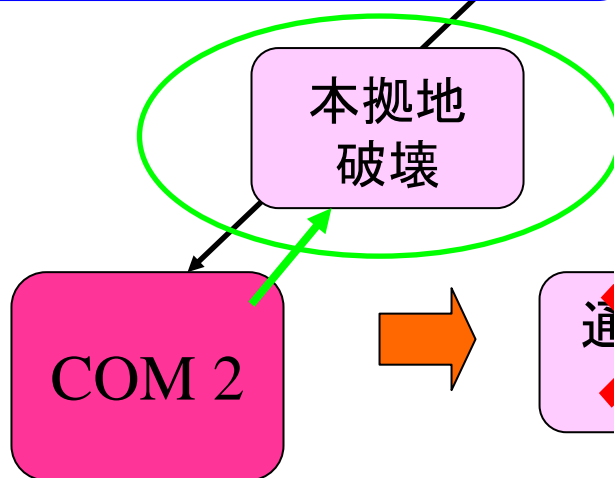
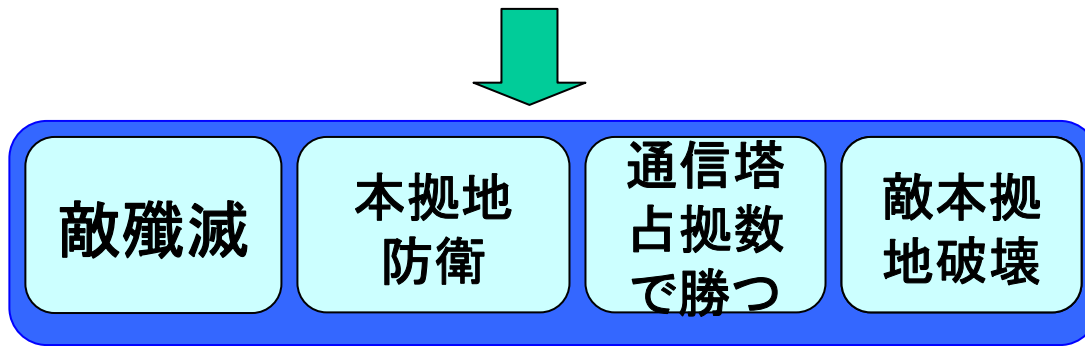


実行評価値 : 88

# チームAI の意思決定と COMの判断 を比較して、最終的に決定する

チームAI 意志決定機構

COMが二つの評価値を  
比較して高い方を選択する。



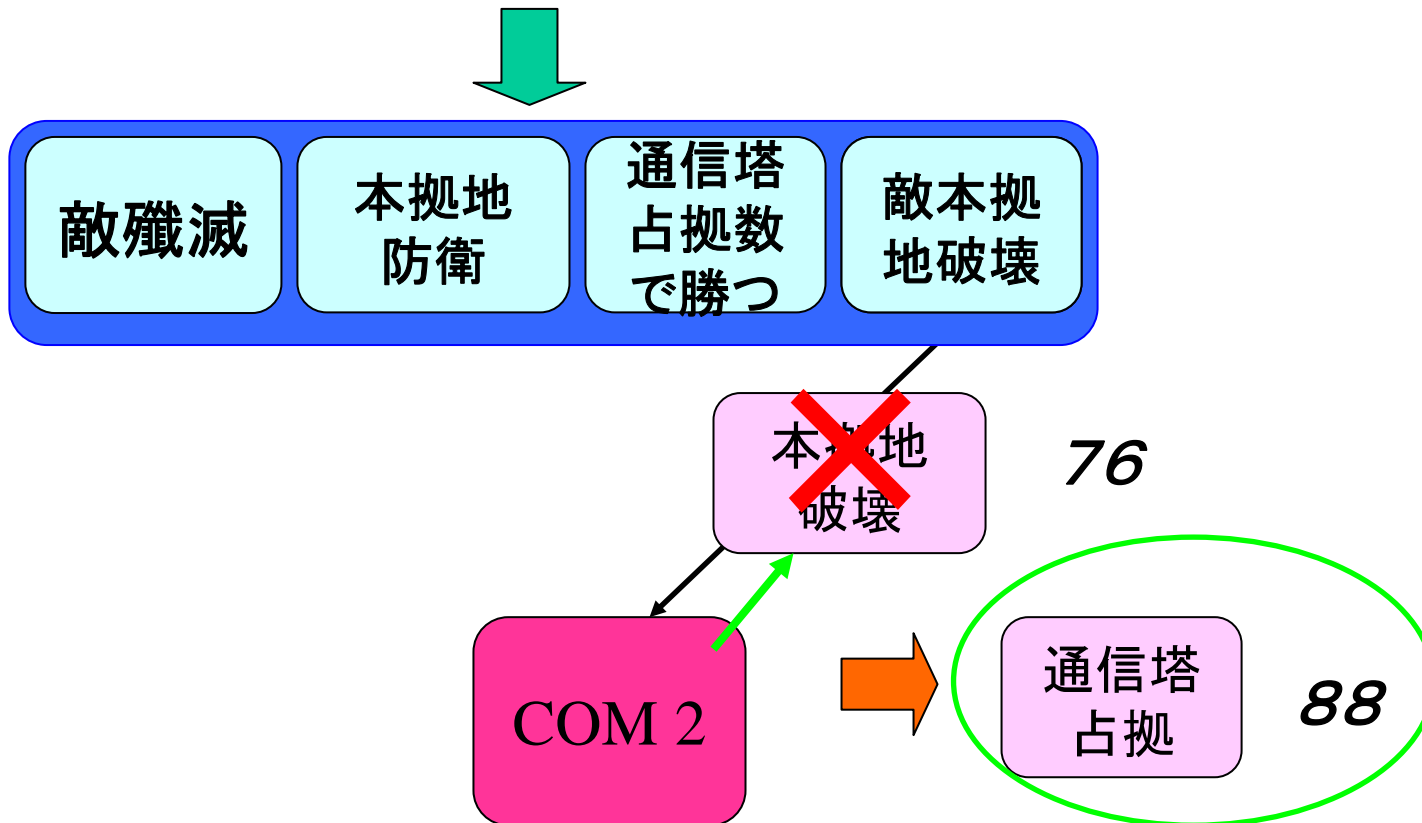
実行評価値：64

実行評価値：53

# チームAI の意思決定と COMの判断 を比較して、最終的に決定する

チームAI 意志決定機構

COMが二つの評価値を  
比較して高い方を選択する。





# チームAIのデモをご覧ください



Player 1  
GOAL\_THINK, 寿命 -1.000000  
GOAL\_ATTACK\_TARGET, 寿命 73.012772  
GOAL\_ESCAPE, 寿命 72.995522  
GOAL\_JOIN\_TARGET, 寿命 72.978271  
GOAL\_APPROACH\_TARGET, 寿命 13.980742  
GOTO\_POSITION, 寿命 -1.000000  
FOLLOW\_PATH, 寿命 -1.000000

A larger screenshot of the game environment showing the AI goal list for Player 1. The list is the same as in the smaller screenshot, but the top three goals (GOAL\_THINK, GOAL\_ATTACK\_TARGET, GOAL\_ESCAPE) are highlighted with a green circle. A green arrow points from the top of the list in the smaller screenshot to the 'GOAL\_ATTACK\_TARGET' goal in this larger screenshot.



===チーム 0===

現在のチーム方針：本拠地制圧

本拠地制圧 0.672677

本拠地防衛 0.384698

敵殲滅 0.530446

コムバス占拠 0.231318

===チーム 1===

現在のチーム方針：敵殲滅

本拠地制圧 0.436685

本拠地防衛 0.000659

敵殲滅 0.439338

コムバス占拠 0.206007

Player 4

GOAL\_THINK, 寿命 -1.000000

GOAL\_ATTACK\_TARGET, 寿命 13.553795

GOAL\_ESCAPE, 寿命 13.593799

GOAL\_JOIN\_TARGET, 寿命 13.533298

GOAL\_APPROCH\_TARGET, 寿命 15.808828

GOTO\_POSITION, 寿命 -1.000000

FOLLOW\_PATH, 寿命 -1.000000

GOAL\_PROSITIT\_TARGET, 寿命 175.227051

GOAL\_ATTACK\_TARGET, 寿命 29.848152



234  
9998



# チームAIまとめ

## この方法の利点と欠点

- (1) 完全に秩序のある行動はしなくなった
- (2) 得たもの  
局所的な状況に対応しながら、  
グローバルな情報を元にした  
チームの行動を取る能力

# 第2部まとめ

# 第2部 質疑応答

# 本講演の構成

## 第1部

ゲームを発展させるための人工知能技術の  
導入のあり方の概論

## 第2部

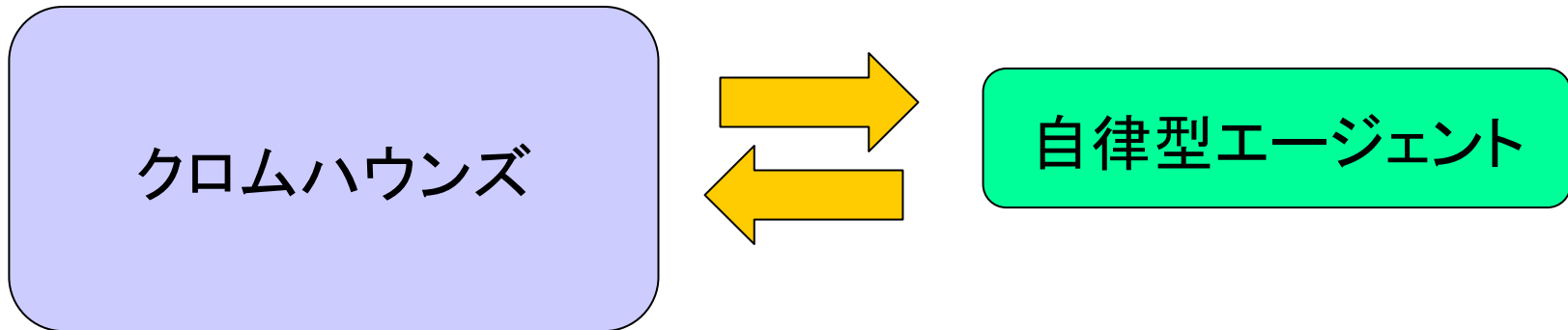
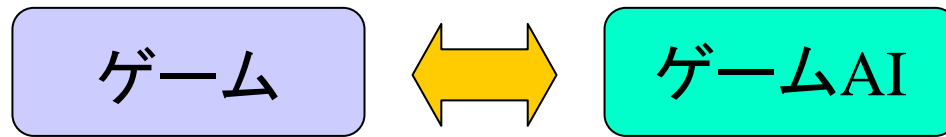
クロムハウズにおける人工知能の開発

## 第3部

これからのゲームとゲームAIのための展望(10分)  
質疑応答(5分)

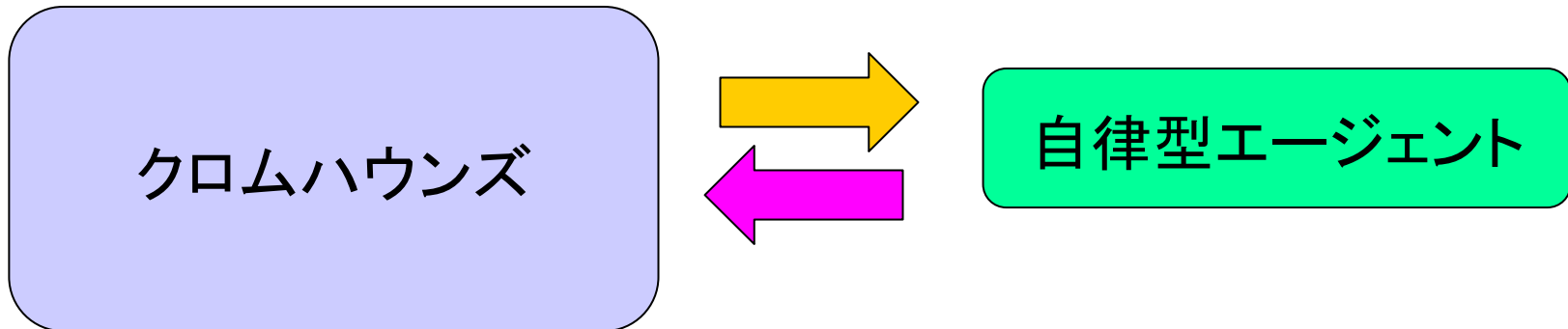
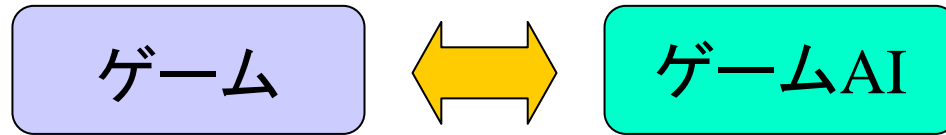
# 第3部 これからのゲームと ゲームAIのための展望

# ゲームとゲームAI





# ゲームとゲームAI



# ゲームAIからゲームを見る

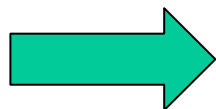
- (1) クロムハウنزを作ることで見えて来たもの
- (2) クロムハウنزを超えて

クロムハウズを作ることで  
見えて来たもの

# クロムハウズで実現したかったこと

(と気付いた)

コンピューターの中に一つの世界を作り、  
その世界で自律的に生きることが出来る生命を作り、  
プレイヤーと対決させてみたい。



新しいゲーム体験を創造する

クロムハウন্ズ以外でこれから実現したいこと  
コンピューターの中に一つの世界を作り、  
その世界で自律的に生きることが出来る生命を作り、  
プレイヤーをその世界に放り込んでみたい。



新しいゲーム体験を創造する

クロムハウন্ズを超えて

# ➡ 自律型エージェントを使って もっといろいろなゲームデザインができる

(例1) クロムハウنزに限らず、  
仲間としてのエージェントを考える。

- (1) 少し命令するだけで、後は状況に応じて自ら判断してくれる
- (2) プレイヤーの情報を取り込んで意思決定をすることで、プレイヤーにアドバイスをすることが出来る。
- (3) いつでも、何処へでも移動する能力を持つ。



# ➡ 自律型エージェントを使って もっといろいろなゲームデザインができる

(例2) クロムハウズに限らず、  
敵としてのエージェントをより進化させる。

- (1) プレイヤーの情報を取り込んで意思決定をすることで、プレイヤーの行動を予測する。
- (2) プレイヤーの癖を見抜く。
- (3) いつでも、何処へでも移動する能力を持つ



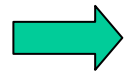




# 自律型エージェントを使って もっといろいろなゲームデザインができる

(例3)

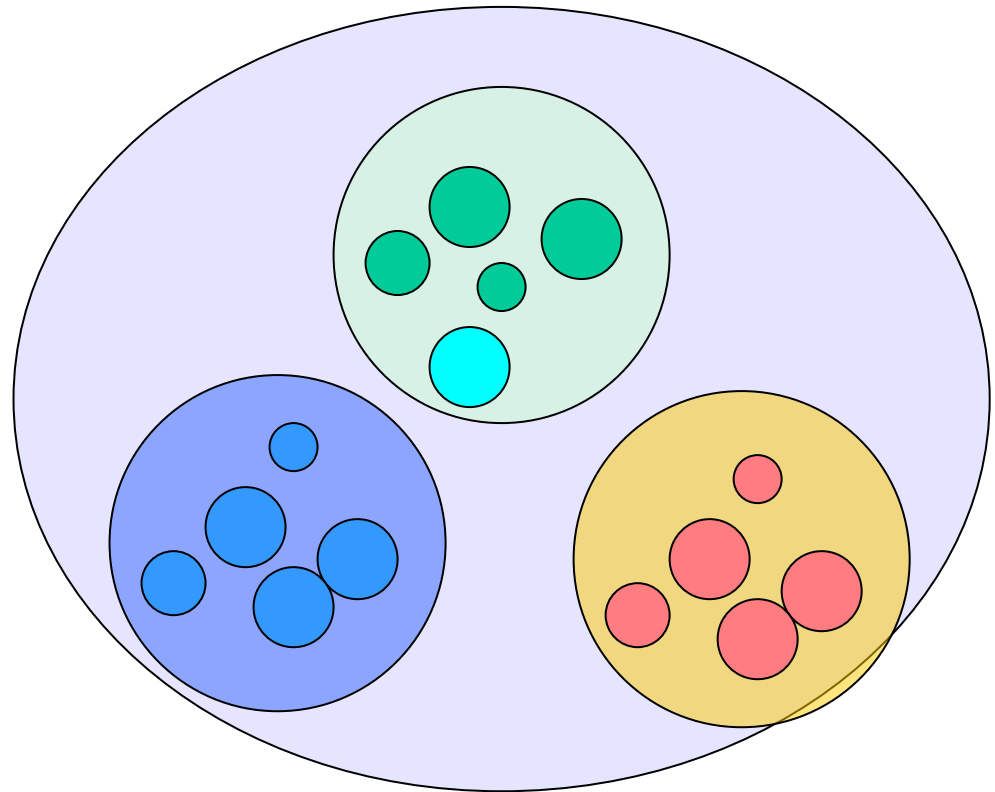
エージェントは、戦闘する敵でなくてもよい。



自律型エージェントたちで社会全体を作って、  
その中にプレイヤーを放り込んでみる。

今までと何が違うか？

- (1) それぞれのエージェントが高度な目的を持ち、自律的に行動する。
- (2) 社会全体としても、集団に、目的を持たせることができる。  
(チームAIの応用)
- (3) いつでも、何処へでも移動する能力を持つ





# 自律型エージェントを使って

## もっといろいろなゲームデザインができる

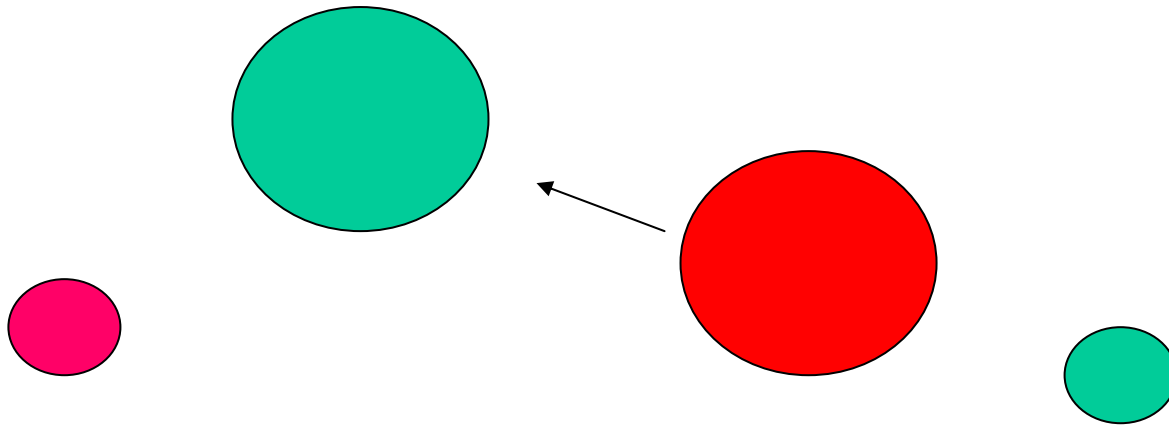
(例4)


人間とエージェントのペアが一組であるようなゲームを作る

ロボットゲームで小型のロボットのエージェントに学習機能を持たせて、成長させて、エージェントと共に戦う。

今までと何が違うか？

ある程度、命令を聞き、ある程度、自律的に戦う。



 自律型エージェントを使って  
もっといろいろなゲームデザインができる

(例5)

エージェントのチーム同士で戦闘をさせてみる

エージェントだけからなるチーム同士の戦闘の  
デモをご覧ください。

全てのCOMはAIで動いています。



14:46



234  
9998





1234  
9998

11:15





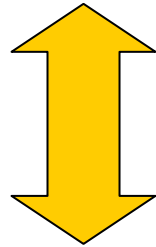
1234  
9998



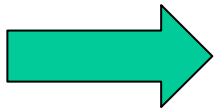
08:07

# 今回のゲームAIの開発で気付いたこと

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



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



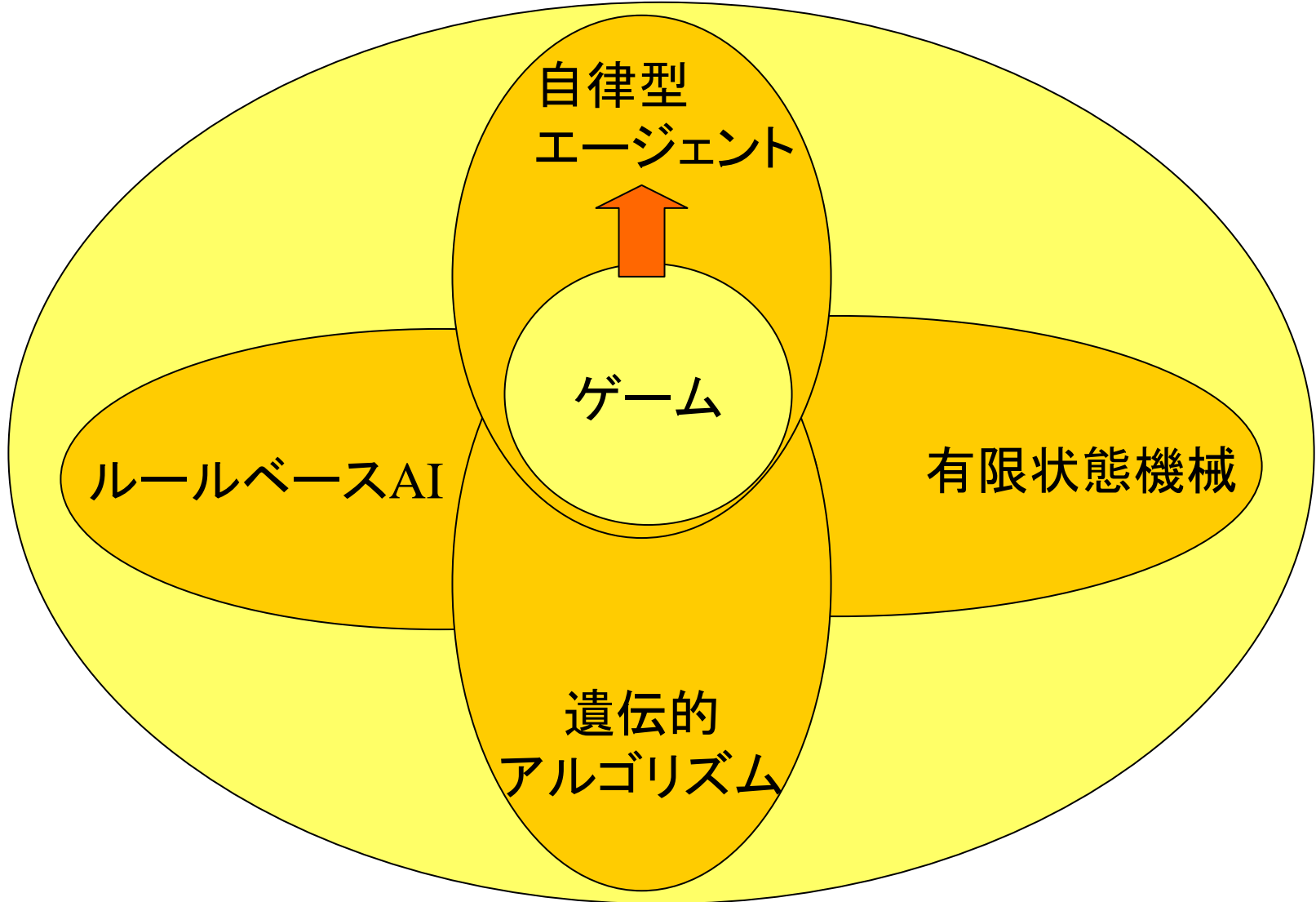
ゲームとゲームAI は、  
お互いを進化させる力を持つ

# ゲームとゲームAIの共進化

それぞれの開発がいろいろな方向を探求する

⇒ ゲームの可能性が広がる

⇒ 共有できれば、なお素晴らしい





# 本講演のまとめ

- (1) ゲームAIには、  
ゲームの未来を切り拓く力がある。
- (2) ゲームとゲームAIは、お互いがお互いを進化させる。
- (3) ゲーム業界全体で、AIの技術の応用に取り組むことで、ゲームの新しい時代を築くことができる。

ご清聴ありがとうございました。

## 質疑応答

これ以外に、意見や質問があれば、メールへ

[y\\_miyake@fromsoftware.co.jp](mailto:y_miyake@fromsoftware.co.jp)

リファレンス

# ゴール指向型AIの参考文献

- Mat Buckland, “Programming Game AI by Example”, WORDWARE publishing
- Jeff Orkin, “[3 States & a Plan: The AI of F.E.A.R.](http://web.media.mit.edu/~jorkin/F.E.A.R.)”, <http://web.media.mit.edu/~jorkin/>
- F.E.A.R.のAI - 3つの状態とゴール指向プランニングシステム (IGDA Japan ホームページ)

[http://www.igda.jp/modules/xeblog/?action\\_xeblog\\_details=1&blog\\_id=62](http://www.igda.jp/modules/xeblog/?action_xeblog_details=1&blog_id=62)

# ナビゲーションメッシュの参考文献

## (1) カウンターストライク

[http://developer.valvesoftware.com/wiki/Bot\\_Navigation\\_for\\_Counter-Strike:Source#Introduction](http://developer.valvesoftware.com/wiki/Bot_Navigation_for_Counter-Strike:Source#Introduction)

# パターンランゲージの参考文献

- (1) C.アレグザンダー  
オレゴン大学の実験  
(鹿島出版会)
- (2) C.アレグザンダー  
まちづくりの新しい理論  
(鹿島出版会)

# 評価値システム

## (1) Killzone' AI

[http://www.cgf-ai.com/docs/straatman\\_remco\\_killzone\\_ai.pdf](http://www.cgf-ai.com/docs/straatman_remco_killzone_ai.pdf)