デジタルゲームにおける最新の人工知能システム

三宅陽一郎

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

Twitter: @miyayou

感想・ご質問は メイルtwitter やfacebookまで

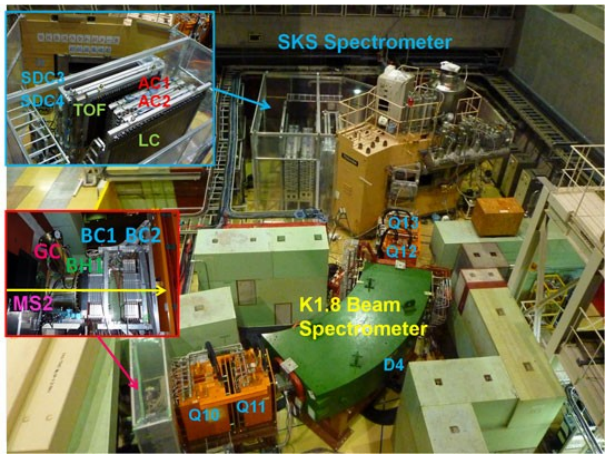
2014.1.10 @東京工芸大学　中野キャンパス

学歴？

京都大学（数学）

大阪大学（原子核実験物理）

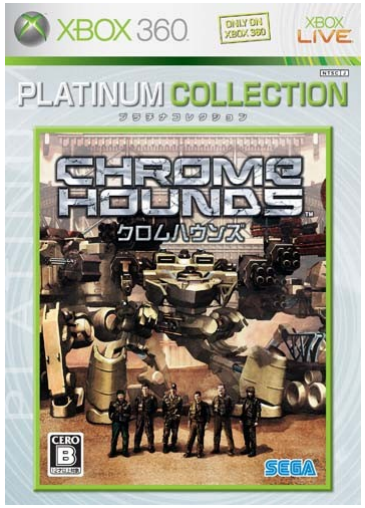
東京大学
（エネルギー工学/人工知能）

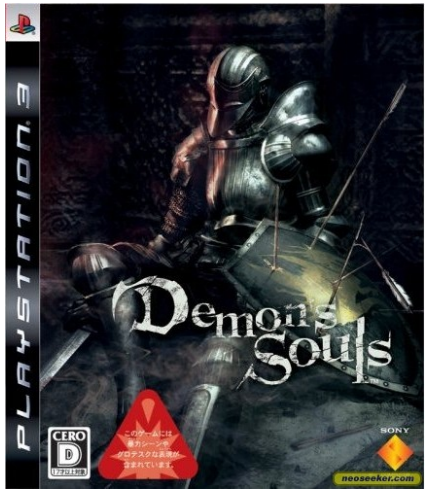


高エネルギー加速器研究所（半年ぐらい。修士論文）

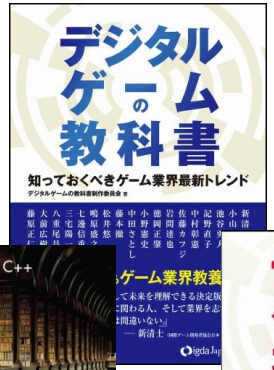
http://www.facebook.com/youichiro.miyake

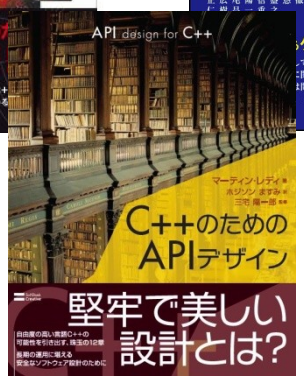
Works (2006-2012)

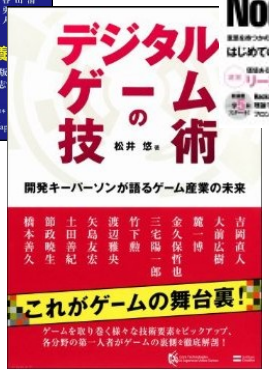


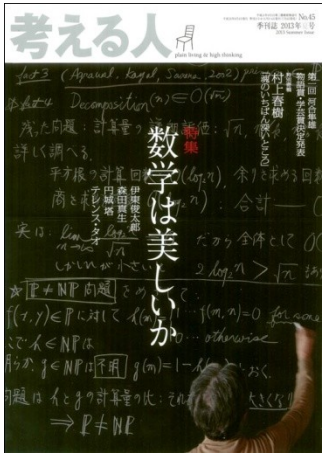


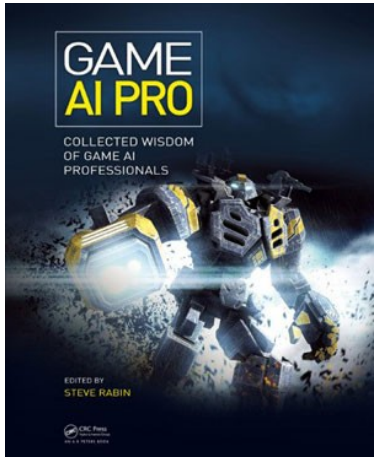
AI for Game Titles

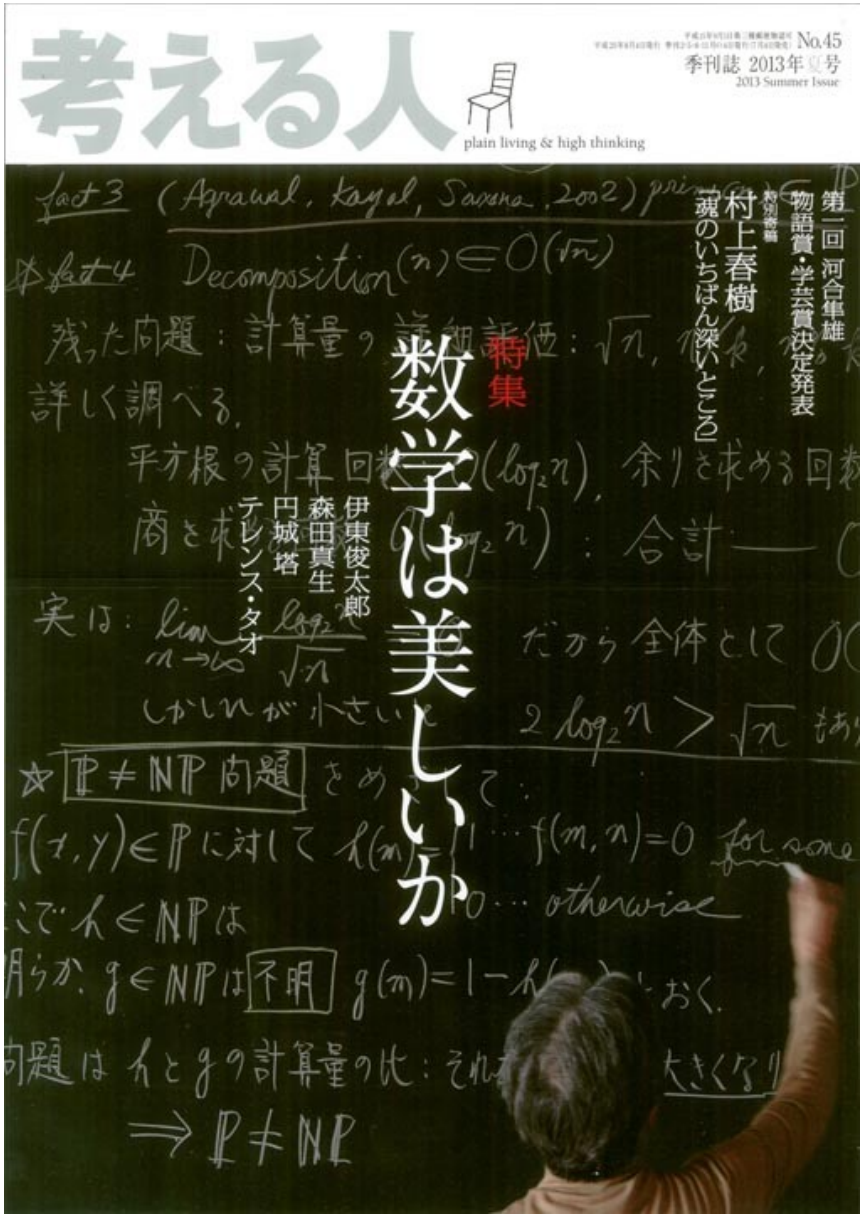












考える人

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 数学の愉悦を味わうために 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世界を探検する 66
長岡亮介 東大の数学入試問題は美しい 70 深川英俊 算額の愉しみ 74

これも数学。 中島さち子 78 綾リュウジ 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

考える夏 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 養老孟司 126
新地球紀行 大河内直彦 168
東北巡礼 吉本直子、中野晴生 176
行ったり来たり マイケル・エメリック 186
ニッポンの馬 内澤旬子 190
月日の残像（最終回） 山田太一 200
柳田国男、今いずこ 山折哲雄 204
琥珀のアーカイヴ 今福龍太 214
生命の内と外 永田和宏 224
チキュウズィン 木内達朗 234
考えない 宮沢章夫 238

連載 plain living

日本のすごい味 平松洋子 114
京都寺町お茶ごよみ 渡辺都 124
私の暮らしかた（最終回） 大貫妙子 144
『犬が星見た』をめぐる旅 高山なおみ 156
娘と私 さげさかのりこ 164
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 岩合光昭 248

今号、ロータリ小子「みちばの絵本」は休載いたします。

表紙写真
「P≠NP」という未証明の難問について、自身の思考プロセスの一端を書く数学者の野崎昭弘氏。
撮影 川上尚見
協力 野崎昭弘、青梅市教育委員会

アートディレクション
島田隆

柱子のトレンディーマーク
坂崎千春

ふんわり広告社
写真 青野健児
デザイン 島田隆

品切で選べられた場合を除き、本誌からのコピーを禁じます。

Books

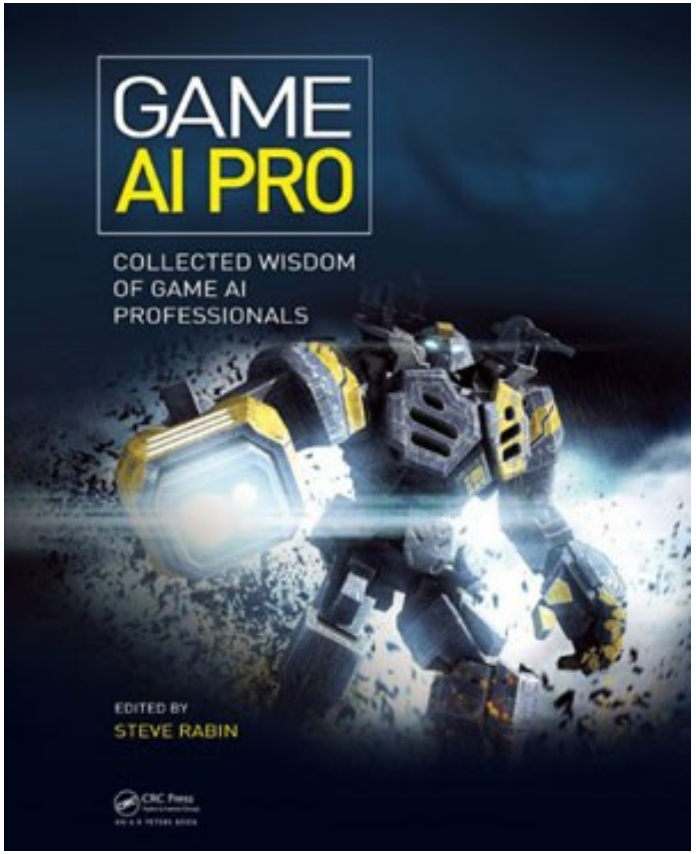


Table of Contents

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

第一部

ゲームAIとは何か？

ゲームにおける人工知能

- 知的な機能をゲームに提供する。
- ゲームデザインの柱となる。
- ダイナミックにゲームコンテンツを変化させる。

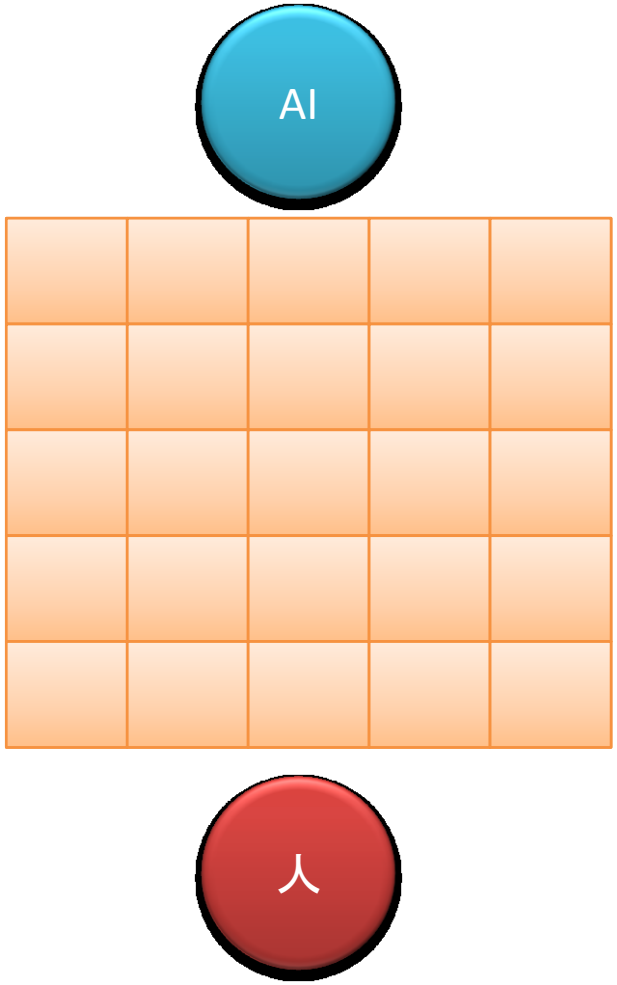
*あらゆるゲーム技術の中で、このような性質を持っているのは**AI**だけです。*

ゲームAIの3つの種類

(1) メタAI　＝　ゲーム全体を監視するAI
(2) キャラクターAI　＝　キャラクターの意思決定をするAI
(3) ナビゲーションAI　＝　地形を認識するAI

それぞれのAIが機能を発揮することで、ゲーム全体がいきいきとする。

ゲームAIを作る過程



チェス、将棋、囲碁など、
ボードゲームの人工知能は、

人間の代わりをするAIであって、
ゲームデザインの一部ではない。

ゲームAIを作る過程

http://www.giantbomb.com/total-war/3025-265/

アクションゲーム、RTS、RPGなど、
デジタルゲームの人工知能は、

ゲームを構成するAIであって、
ゲームデザインの一部である。

ゲームAIは誰のもの？

- *ゲームにおけるAIの仕組みは**エンジニア**が作ります。*
- *しかしスクリプトや調整パラメーターは**ゲームデザイナー**が書くことが多いです。*
- *ゲームAIはゲームの一部だから、**ゲームデザイナー**は人工知能で何ができるかをよく理解してゲームデザインしなければなりません。*

第二部　第一章

ゲームを作ろう1：キャラクターAI編

ゲームを作ろう

- ゲームAIは**ゲームデザインの一部**であると書きました。
- だから、**ゲームAIを理解するには、ゲームを作りながら考える**のが一番です。
- ここでは、**ゲームを作りながら**、ゲームAIについて考えてみましょう。

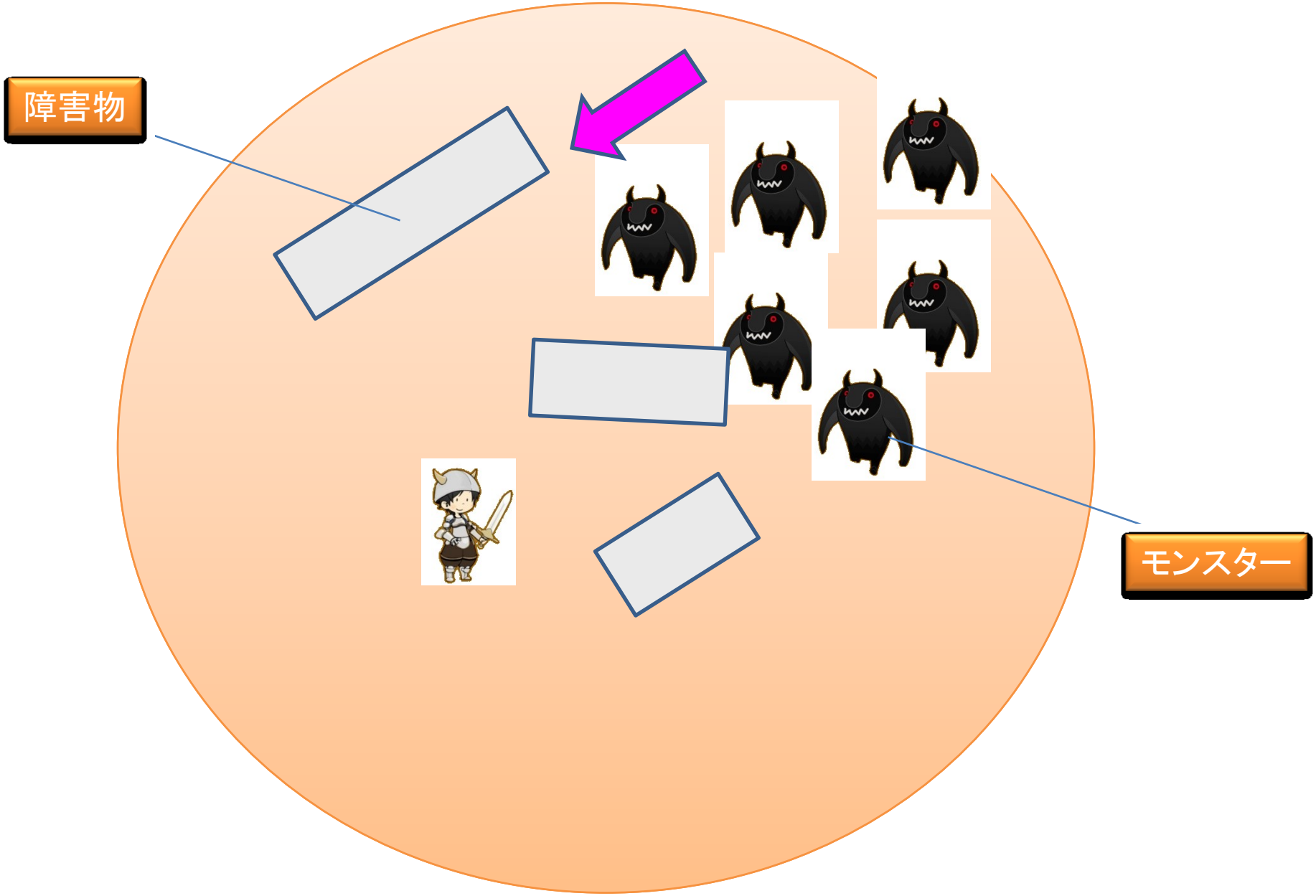
ゲームデザイン

- コンセプト　＝ ゲームの仕組み
- テーマ　＝ ファンタジー？ミリタリー？
- テクニック　＝ ゲームの細かいルール

ゲームデザイン

- コンセプト　… 多vs多でリアルタイムで魔法を打ち合う
- テーマ　… ファンタジー
- テクニック　… 体力があって0になったら戦闘不能。

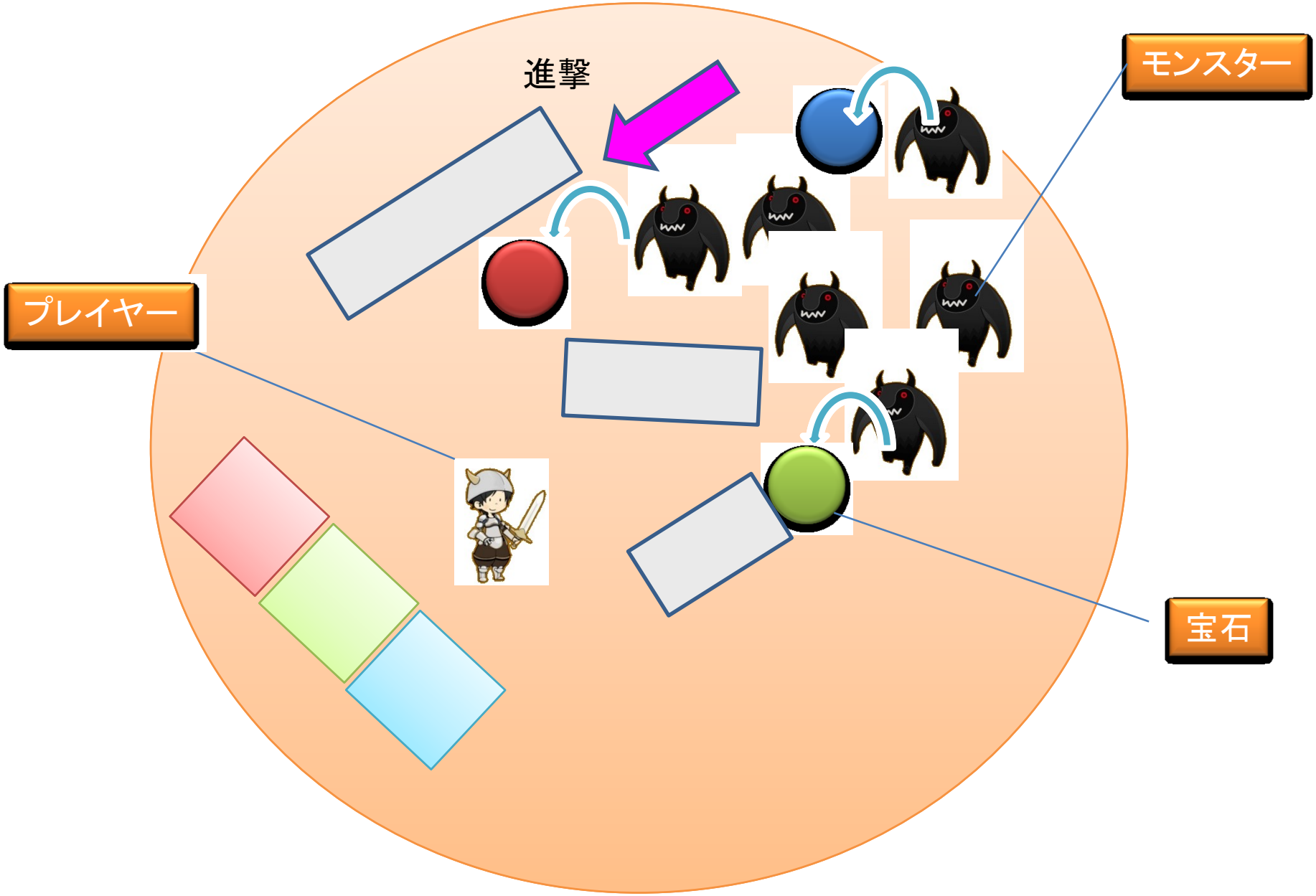
ステップ1：　イメージ



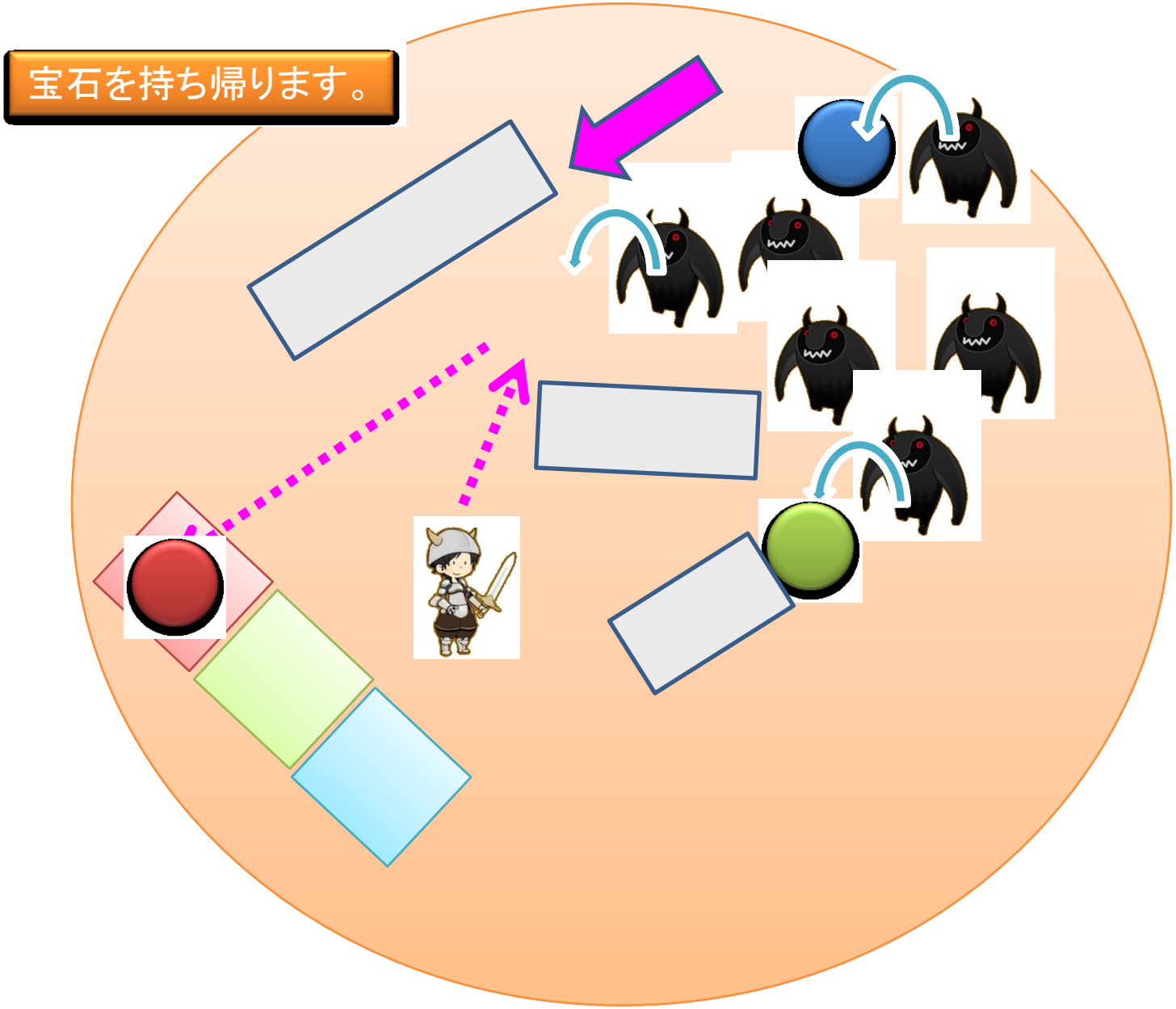
ステップ2：コンセプト

- 地形をうまく使いながら、魔法を撃ち合って戦うゲーム。
- 何のために戦うの？　モンスターが宝石アイテムをドロップする。
- それを指定された場所に持ち帰ると、クリア。

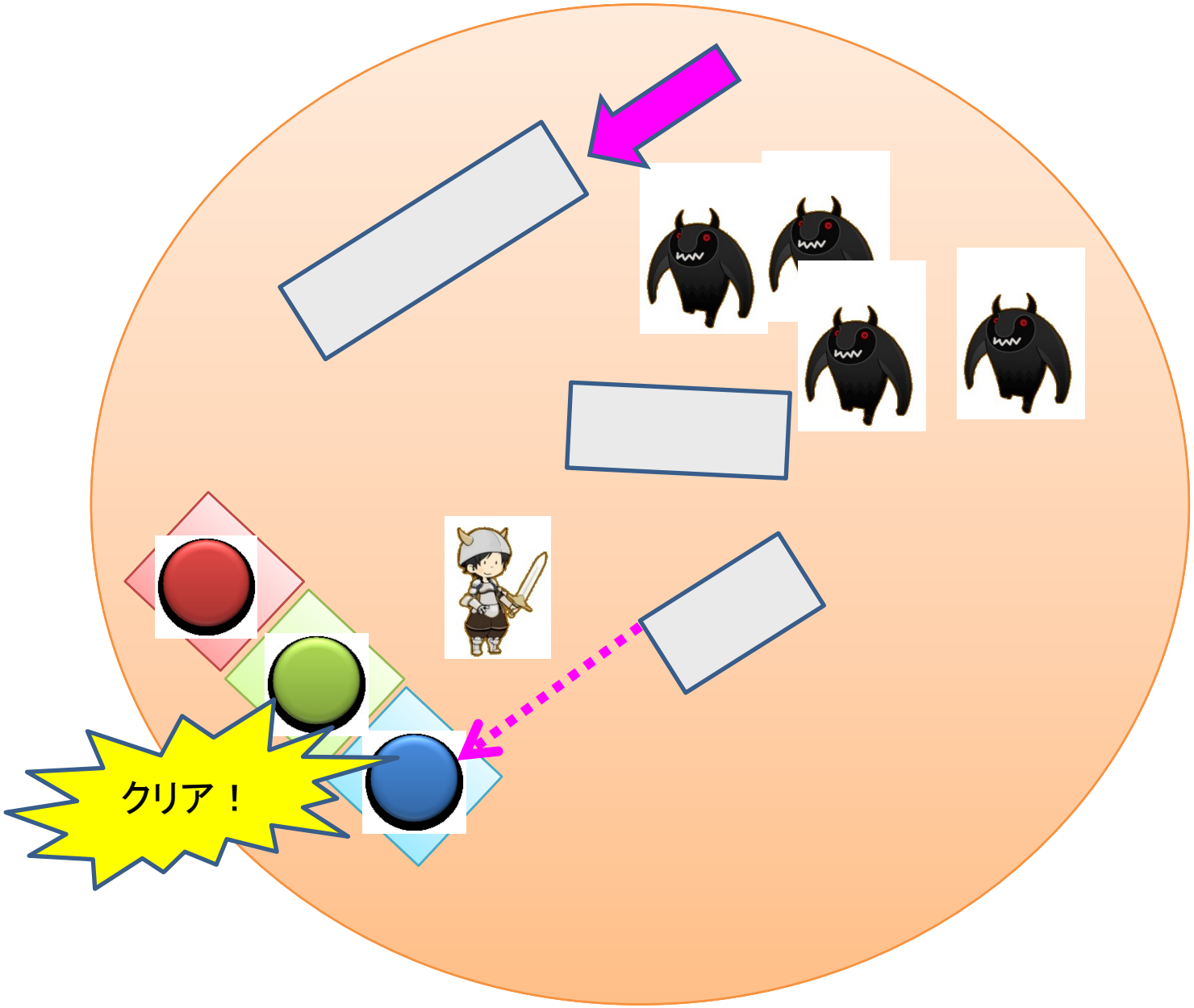
ステップ1：　イメージ



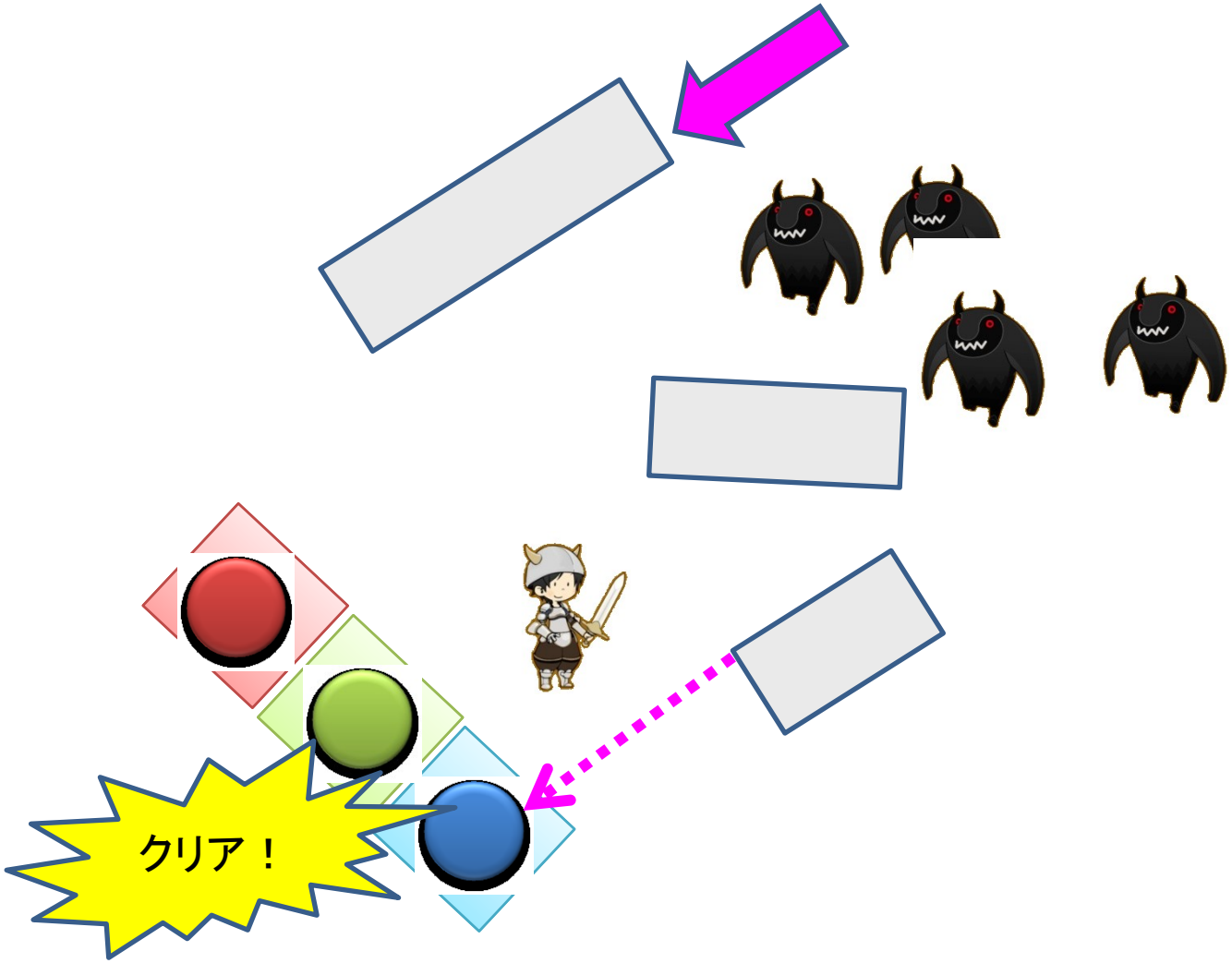
ステップ1：　イメージ



ステップ1：　イメージ



ステップ1：　イメージ



ゲームデザインを再考する。

どうすれば、このゲームが面白くなるか？
＝　どんな体験をユーザーに与えたいか？

ユーザー・エクスペリエンス
(UX ＝ User Experience)

ステップ1：　イメージ

モンスターがどこにいるかわからない不安。
とつぜん、襲ってくる。
ドロップしたアイテムをリスクを冒して持って還る。

ユーザーエクスペリエンスを作る

モンスターがどこにいるかわからない不安。
とつぜん、襲ってくる。
ドロップしたアイテムをリスクを冒して持って還る。

では、ユーザーエクスペリエンスを作る要素とは？

- *レベルデザイン（ステージ）*
- *ゲームルール*
- *デザイン*
- *インタラクティビティ*
- *キャラクター＝AI*

ゲーム＝ユーザー体験を作ること。
すべてはユーザーの体験を作る要素。

エンターテインメント（ゲーム）におけるAI技術を知るとは、
AIによってどのようなユーザーエクスペリエンスを知ることができるかを知ること。

ゲームにおける
ユーザーエクスペリエンスとは何か？

- ユーザーエクスペリエンスを科学する力
  ＝　ゲームデザイナーの力

- スターバックス
- ディズニーランド
- 小説
- テレビ
- 会話
- twitter
- facebook
- …
- ゲーム

いろんなエンターテインメントにおいて、
どうやってユーザーエクスペリエンスが
組み立てられているかを科学しよう！

スターバックス

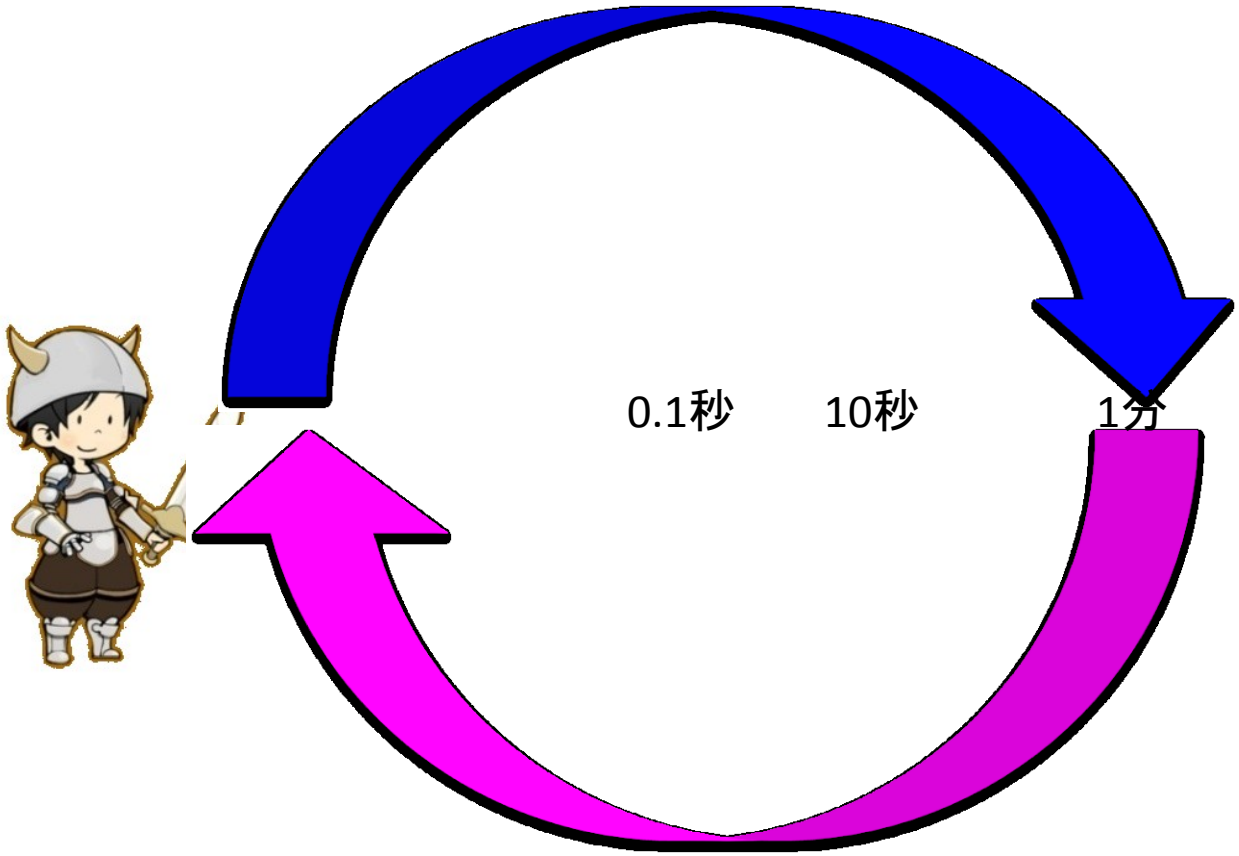
- お店に入る
  - 注文の列に並ぶ
    - 注文する
      - 待つ
        - 座る
          - コーヒーを飲む
            - 本を読む
              - 飲み終わる
                - 読み終わる
                - ぼ～とする
                - 店を出る

スターバックスのUX

- コーヒーがおいしい（瞬時）
- 本を読みながらコーヒーをすするのが楽しい（中期）
- お店にいるのが心地良い（長期）

エンターテインメントは、さまざまな時間スケールの楽しみを提供しなければならない。

複数の時間スケールのゲームデザイン



Will Wright のゲームデザイン論

SporeにおけるゲームAI技術とプロシージャル - DiGRA JAPAN （三宅陽一郎）
http://digrajapan.org/?wpdmact=process&did=Ni5ob3RsaW5r

ゲームのUX

- 敵と戦うのが楽しい（瞬時）
- 後ろに回り込んだり位置取りをするのが楽しい（中期）
- 宝石を集めるのが楽しい（長期）

エンターテインメントは、さまざまな時間スケールの楽しみを提供しなければならない。

ゲームのUXのためのAI

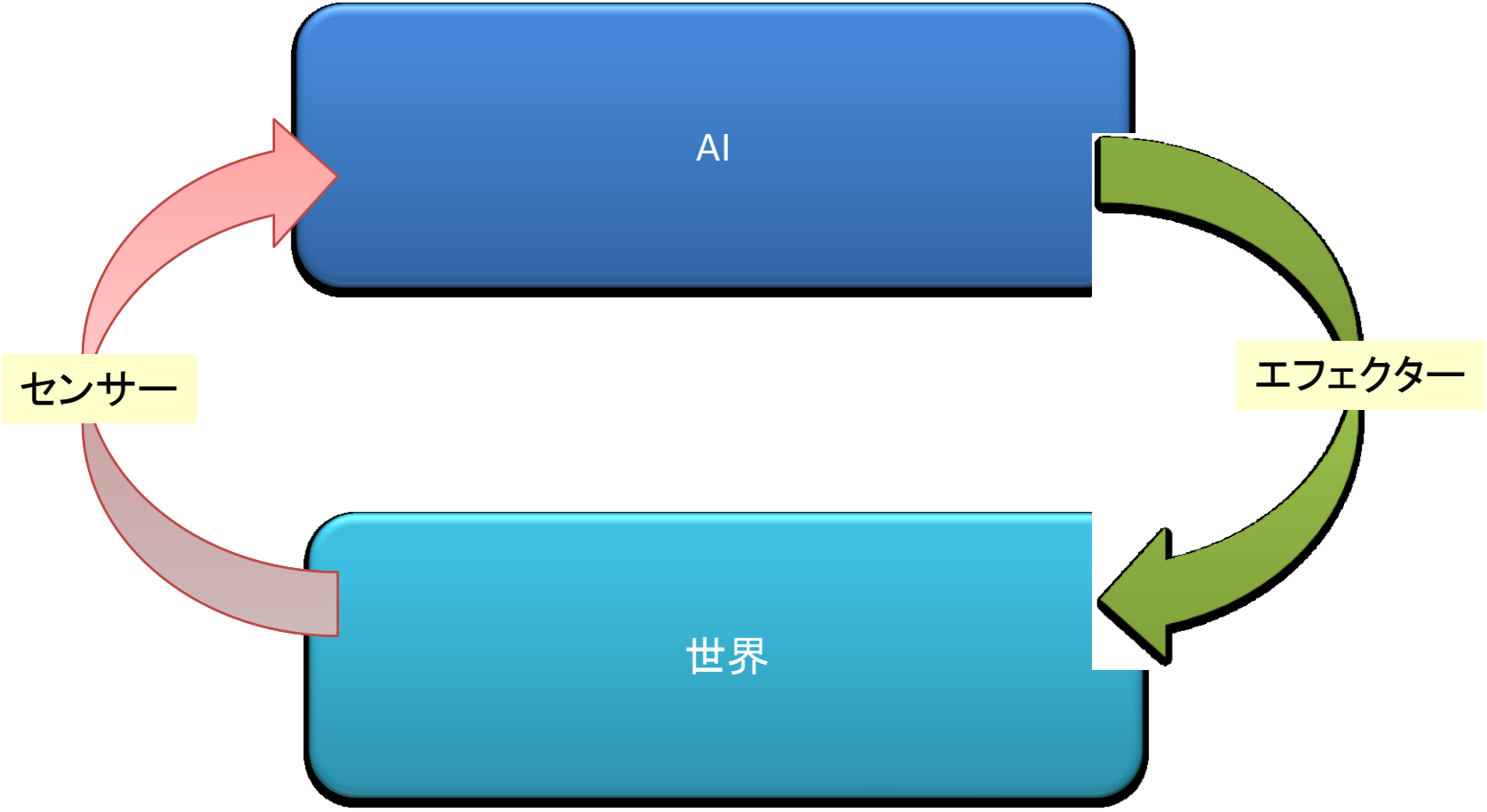
- 敵と戦うのが楽しい（瞬時）
  ＝ AIとユーザーのアクションをどう定義するか？
- 後ろに回り込んだり位置取りをするのが楽しい（中期）
  ＝ AIの位置取り（移動）をどのようにするか？
- 宝石を集めるのが楽しい（長期）
  ＝ どうやって宝石集めを妨害するか？

エンターテインメントは、さまざまな時間スケールの楽しみを提供しなければならない。

AIとユーザーのインタラクション

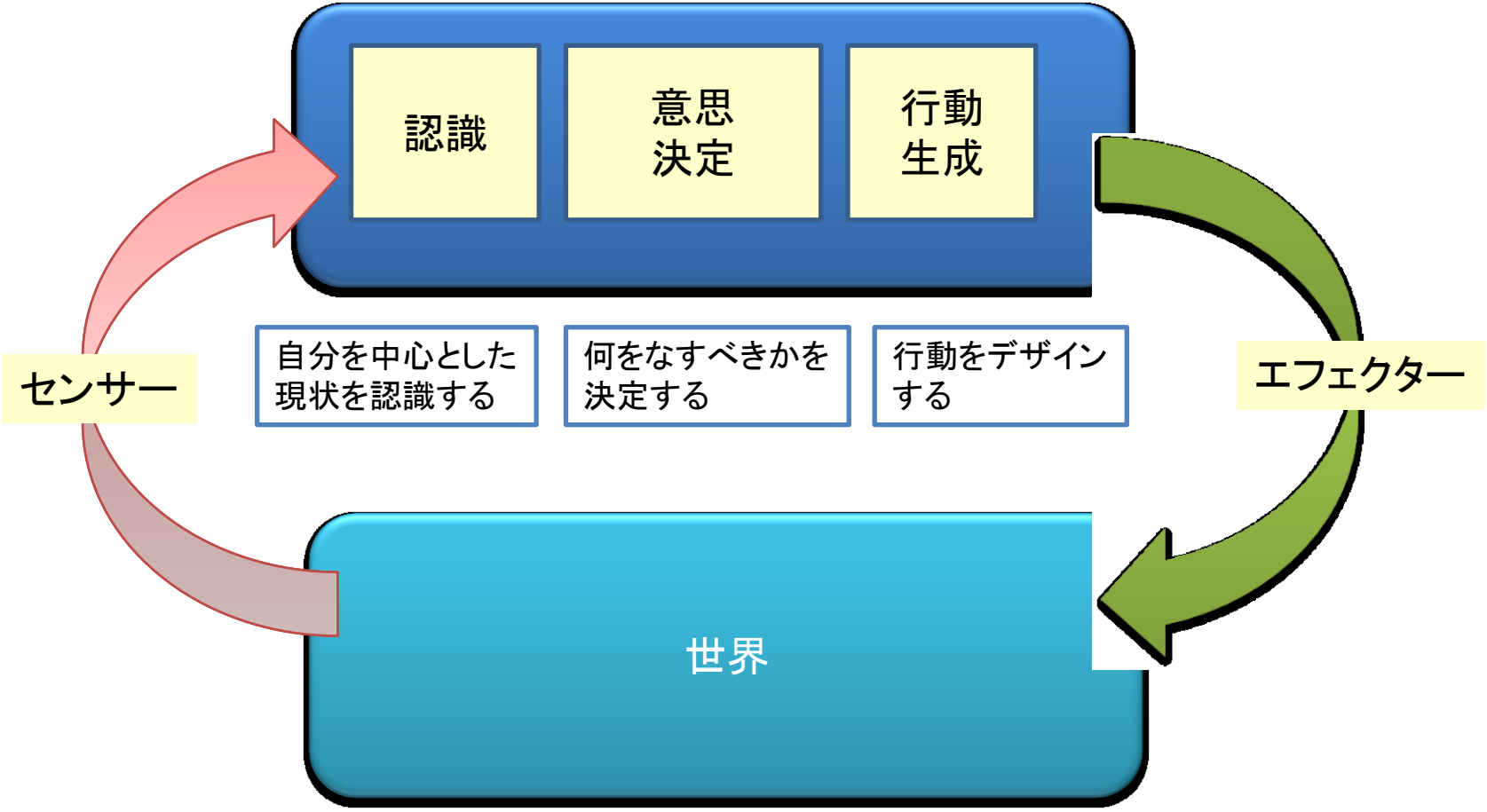
「キャラクターAI」をうまく作ることで、ユーザーとのインタラクションを作成する。

キャラクターAIの作り方



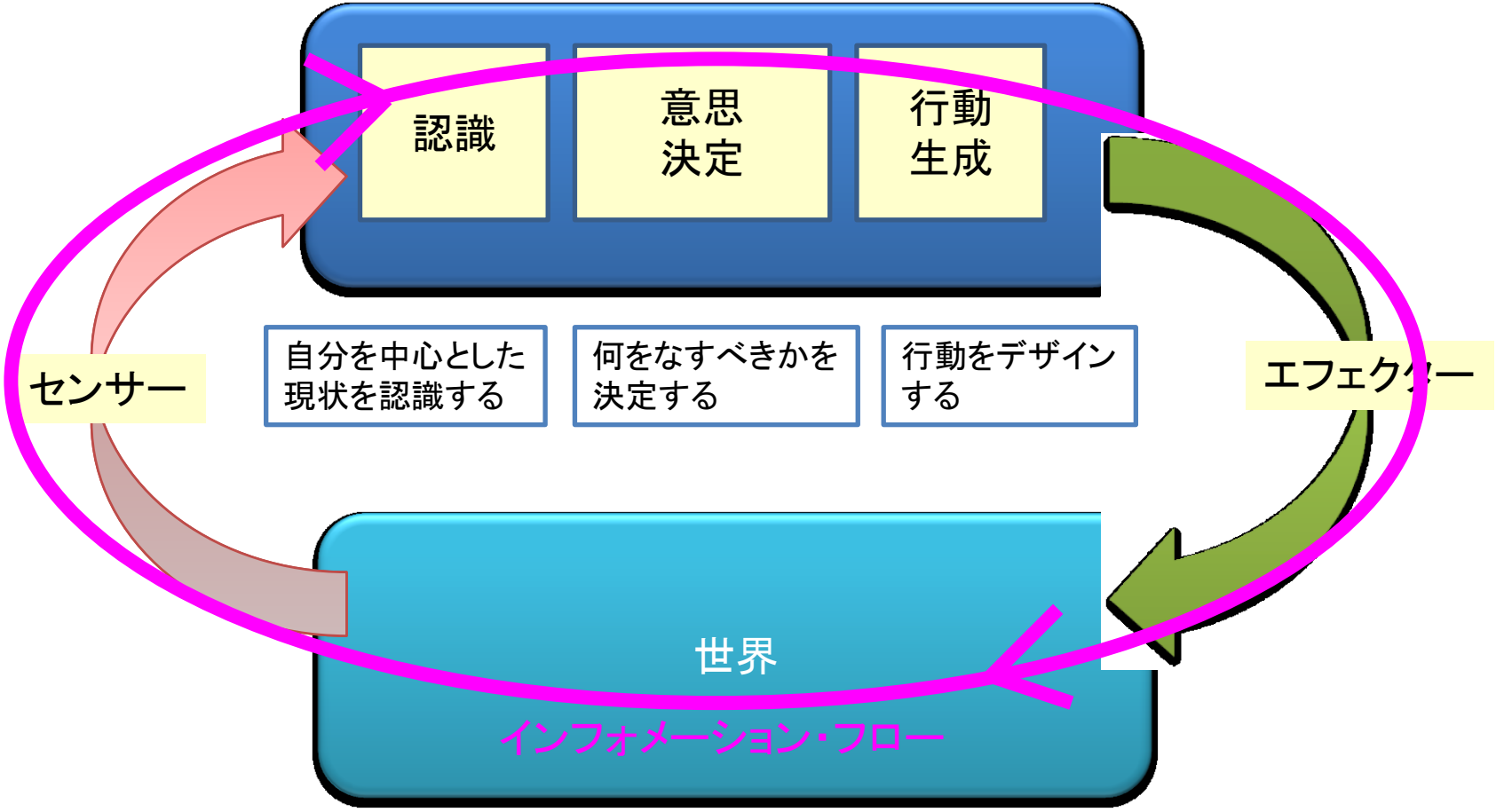
エージェントアーキテクチャ

キャラクターAIの作り方



エージェントアーキテクチャ

キャラクターAIの作り方



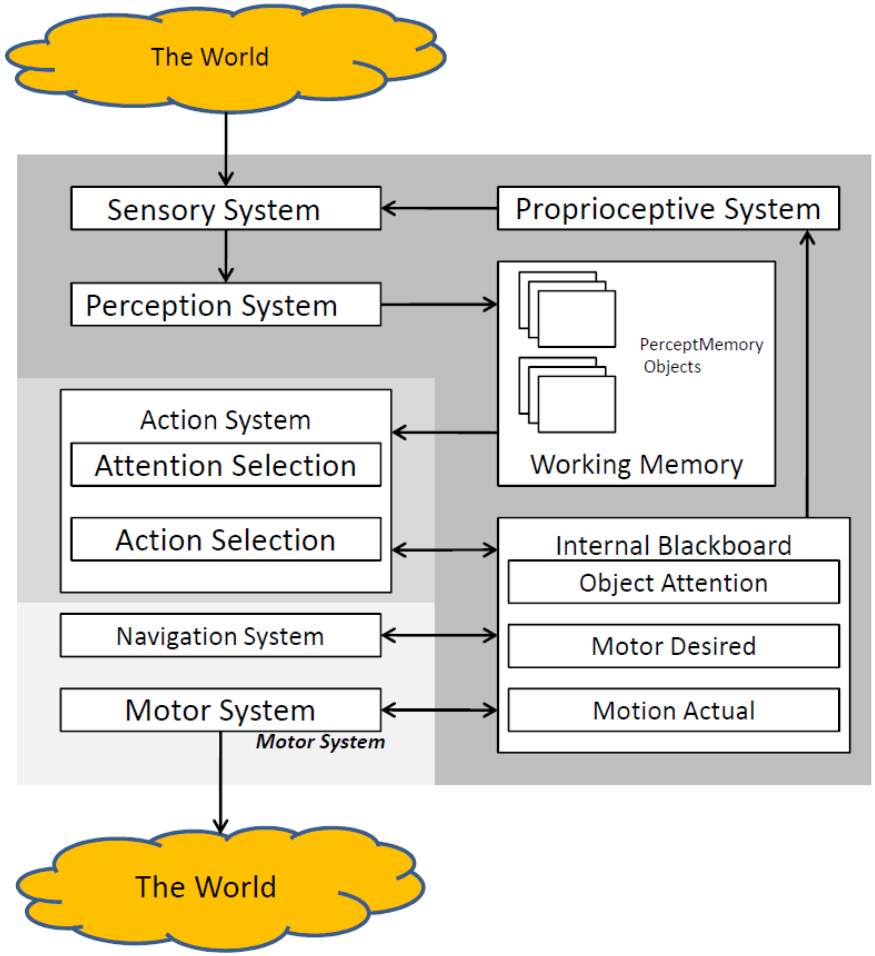
エージェントアーキテクチャ

C4 Architecture

MIT Media Lab.

Synthetic Characters Group

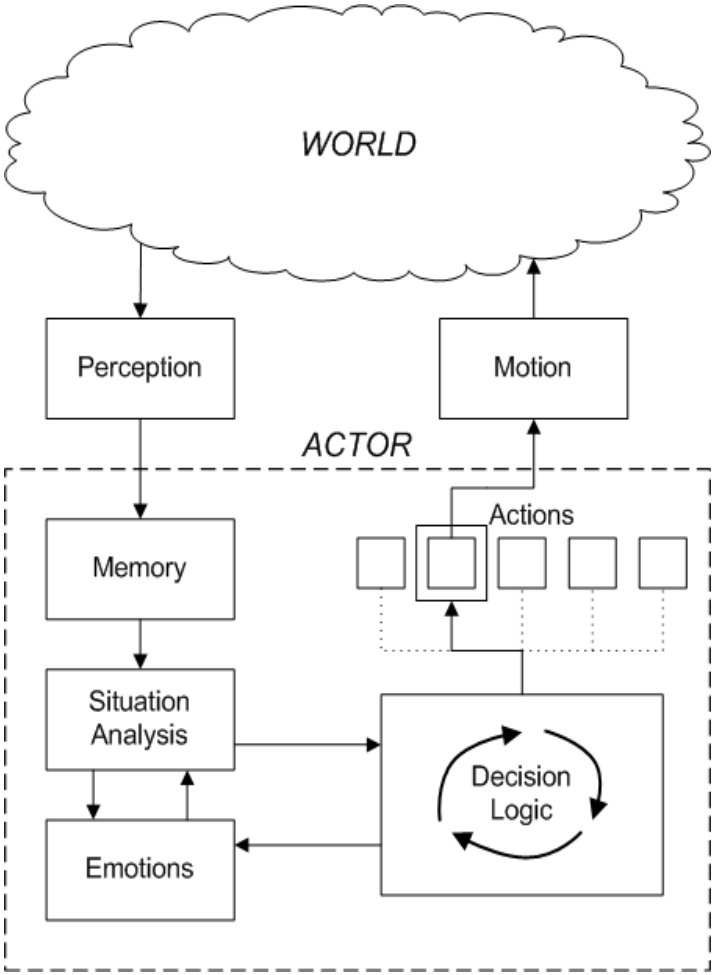
Researching Virtual Pet in Digital World.



D. Isla, R. Burke, M. Downie, B. Blumberg (2001).,
"A Layered Brain Architecture for Synthetic Creatures", http://characters.media.mit.edu/Papers/ijcai01.pdf
http://naimadgames.com/publications.html

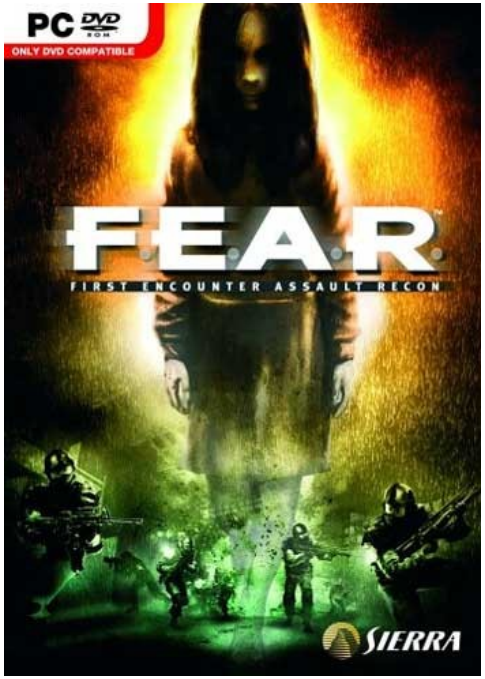
Halo Agent Architecture

- Genre : SciFi-FPS
- Developer : BUNGIE Studio
- Publisher : Microsoft
- Hardware: Xbox, Windows, Mac
- Year : 2002

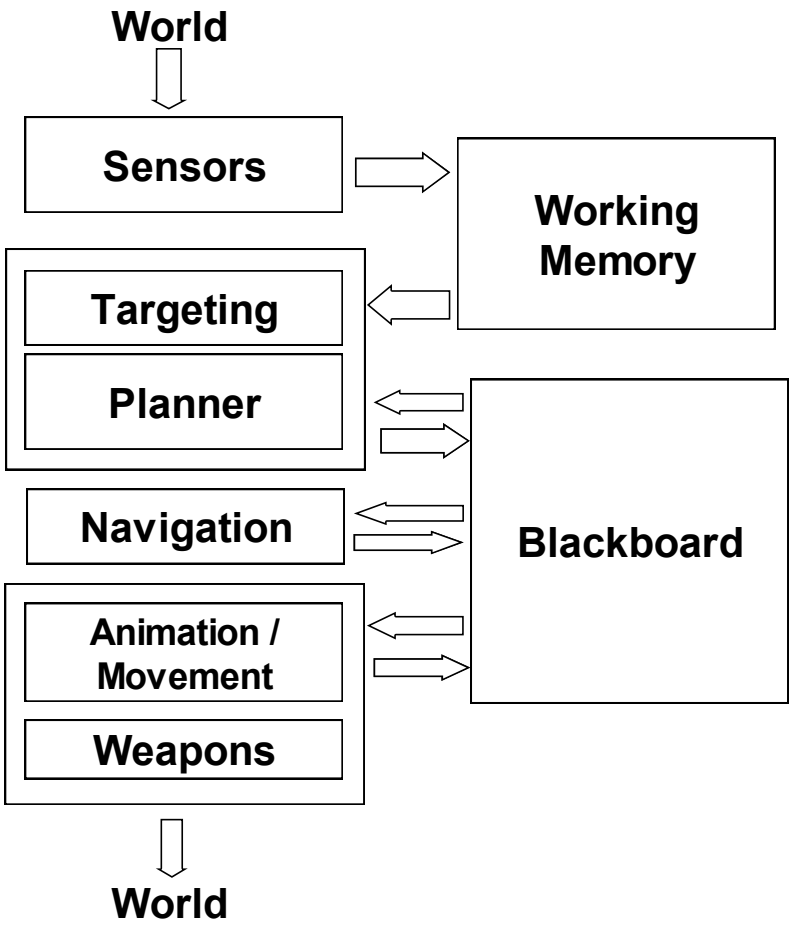


Jaime Griesemer(GDC 2002),The Illusion of Intelligence: The Integration of AI and Level Design in Halo
http://halo.bungie.net/inside/publications.aspx

F.E.A.R Agent Architecture



- Genre : Horror FPS
- Developer : Monolith Production
- Publisher : SIERRA
- Hardware: Windows
- Year : 2004

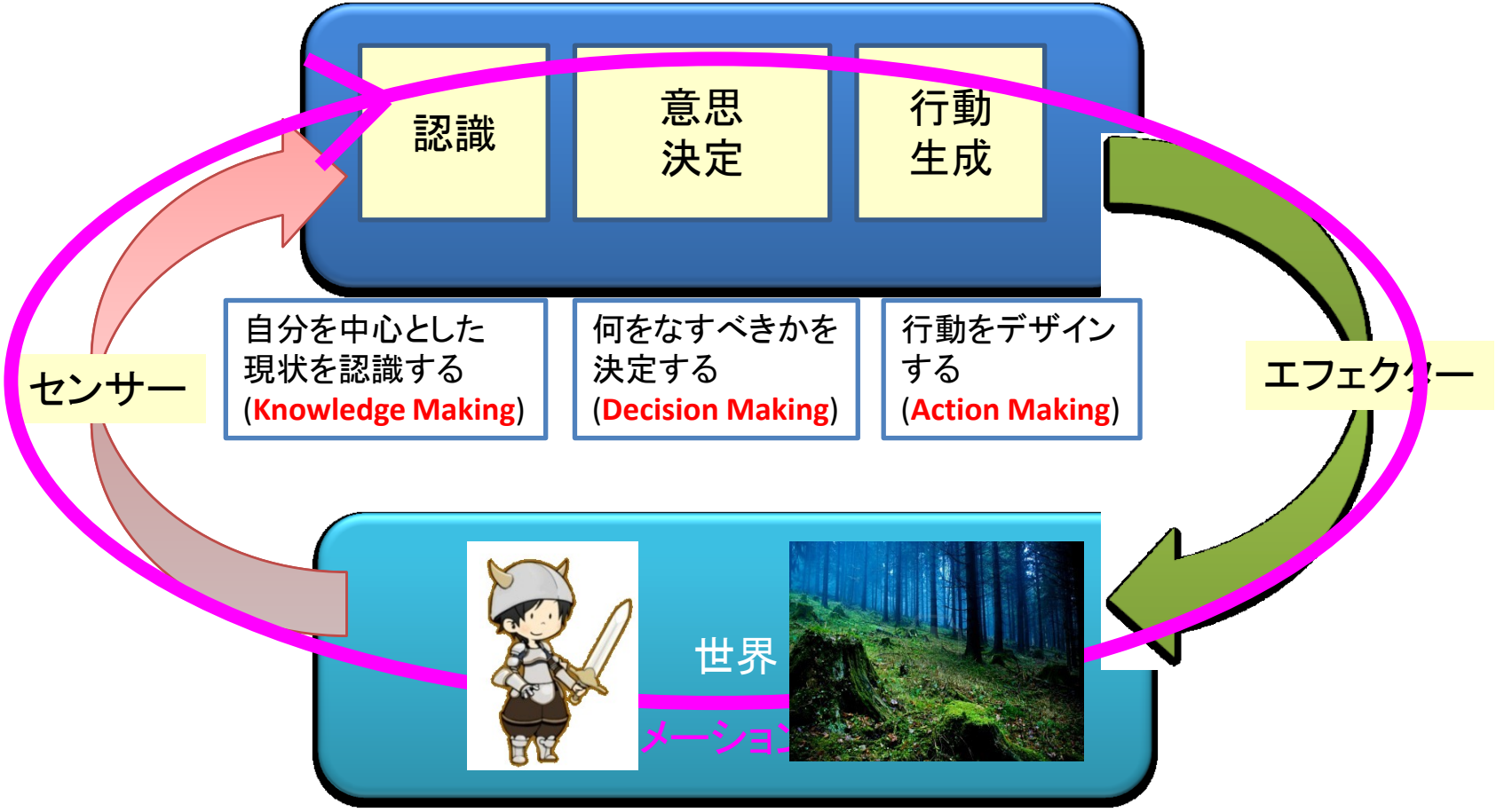


Agent Architecture Considerations for Real-Time Planning in Games (AIIDE 2005)
http://web.media.mit.edu/~jorkin/AIIDE05_Orkin_Planning.ppt

キャラクターAIの作り方

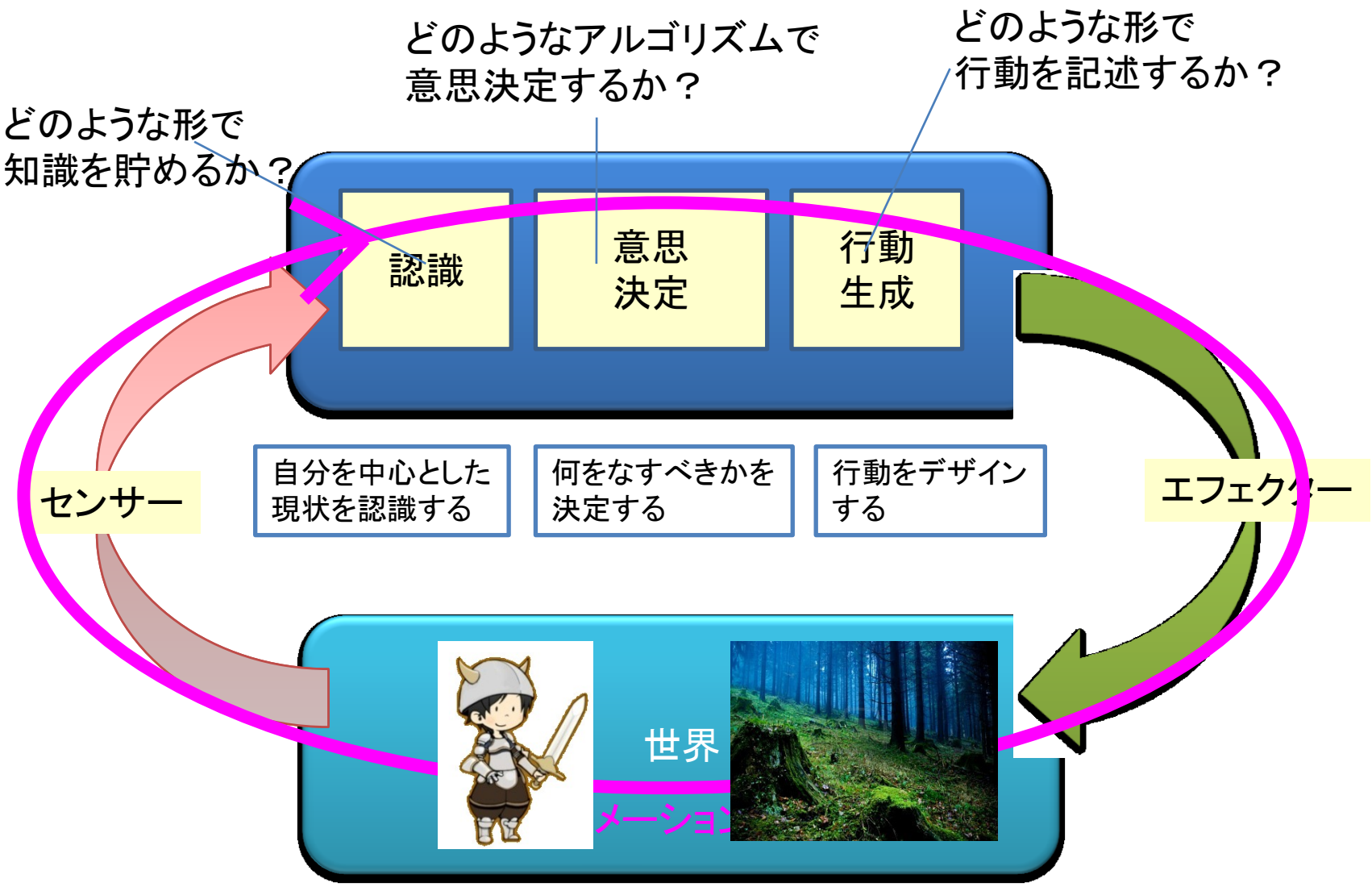
エージェントアーキテクチャ

キャラクターAIの作り方



エージェントアーキテクチャ

キャラクターAIの作り方



エージェントアーキテクチャ

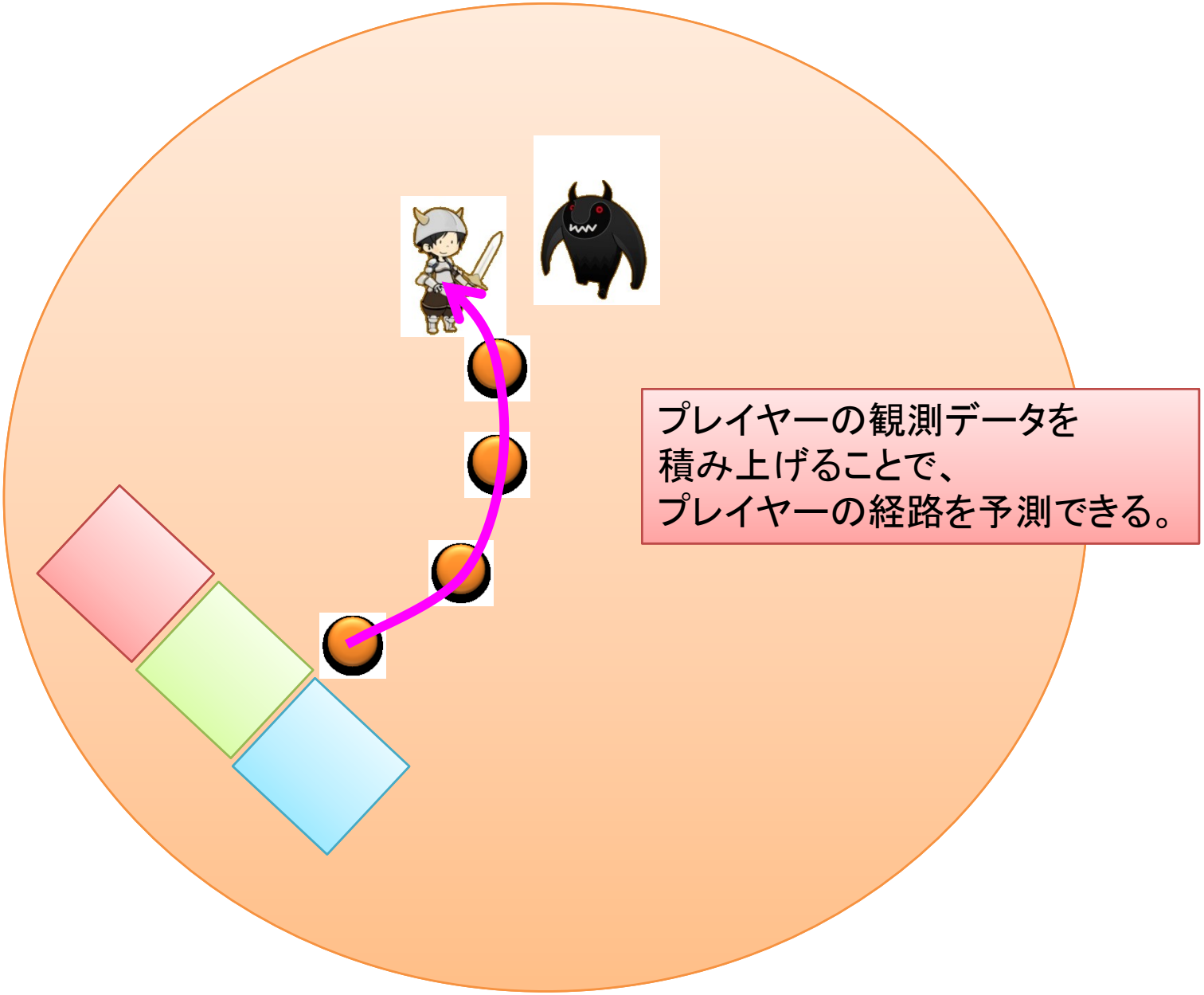
プレイヤーを認識しよう（中距離）。

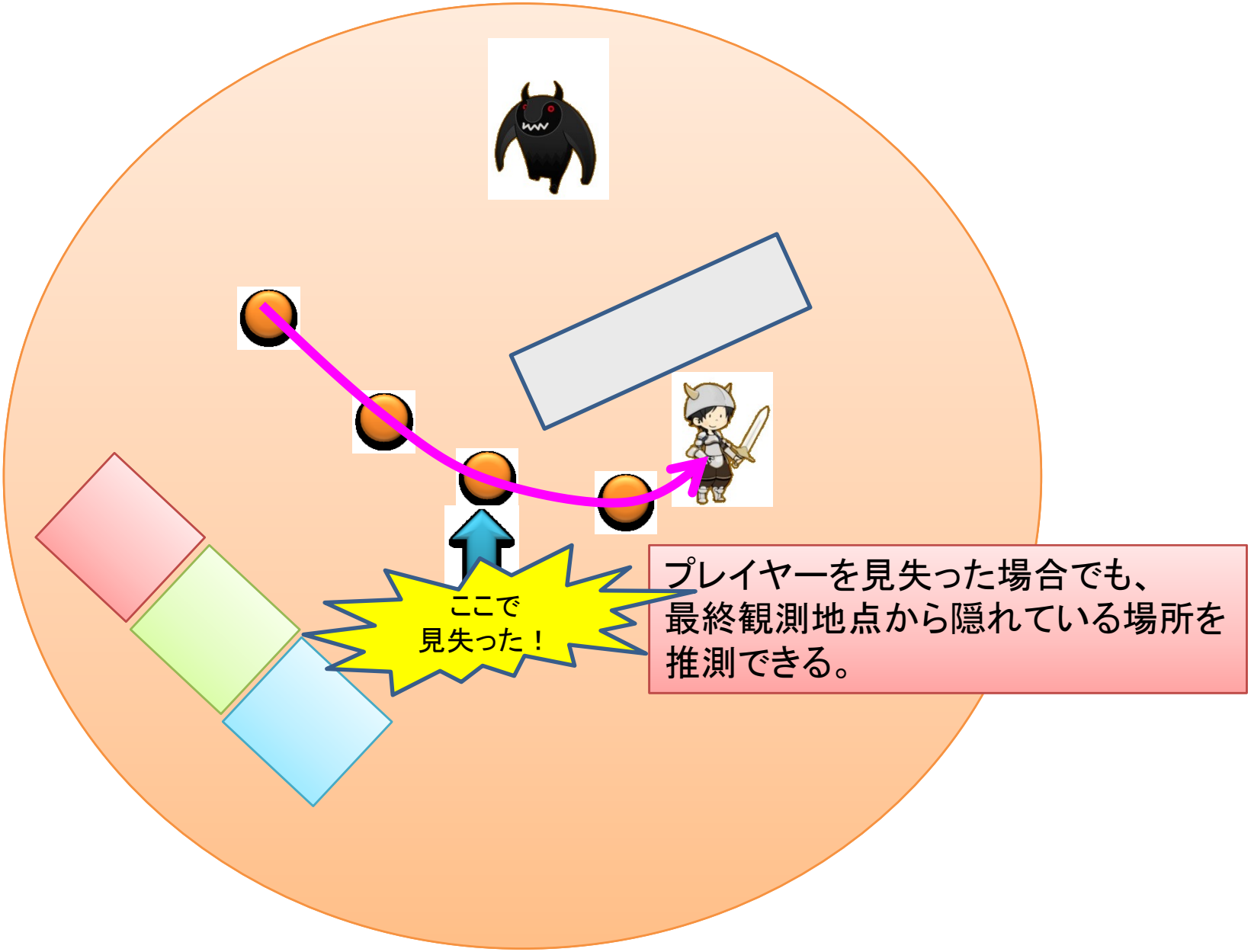
- ここでプレイヤーに関する知識の形を決定しましょう。
- こういった知識の形のことをキャラクターに関する知識表現（Knowledge Representation）と言います。

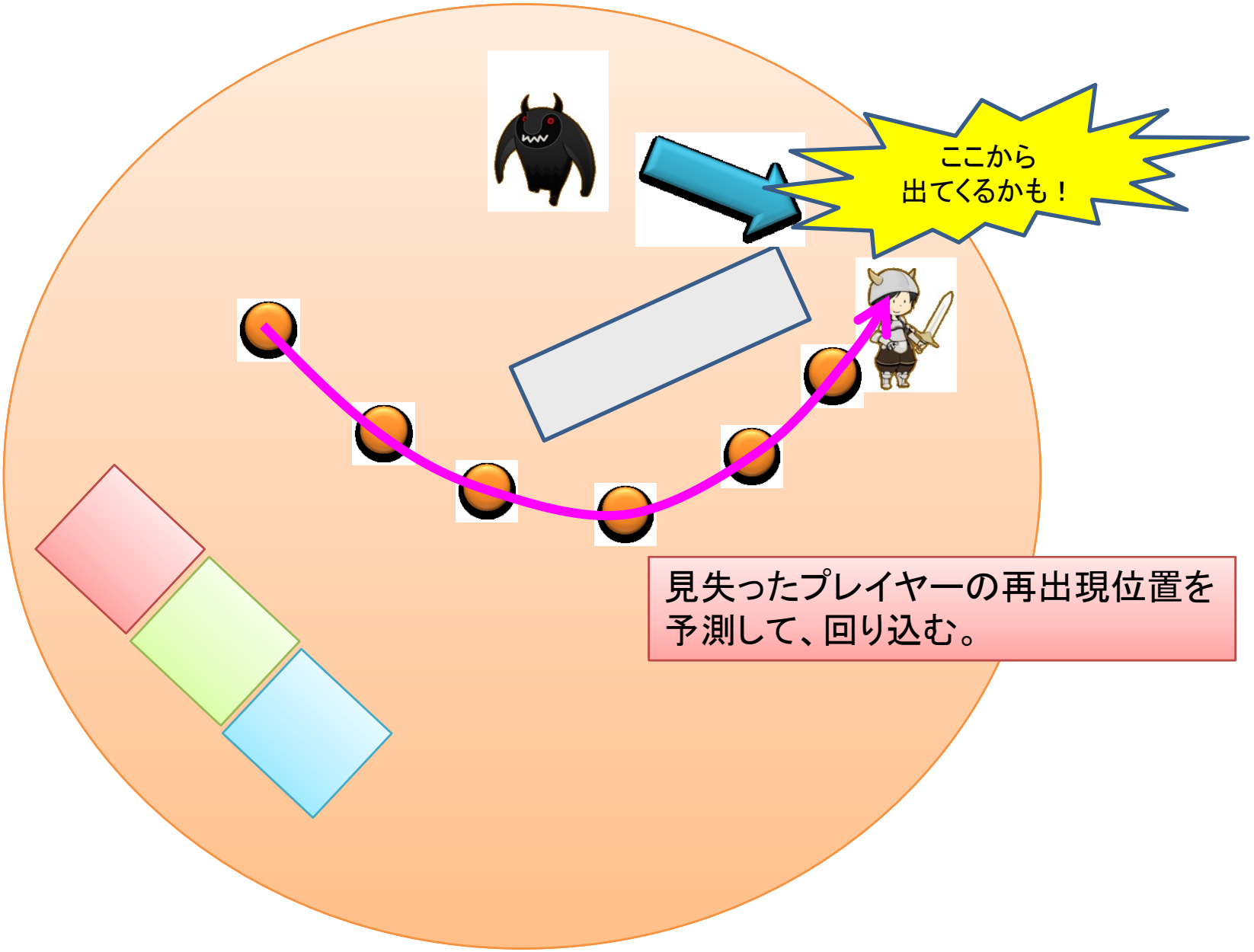
(例)

```
Struct Player_info
{
float _time: // 観測時刻
float[4]  _loc= (4.5,31.4,343.f,1.f);   //位置
float _velocity = 3.2f;  //速度
bool  b_armed = true: //武装してた？
bool  b_injured = falase ; // 負傷してた？
…
};
```

こういった情報を毎フレームか数フレームごとに取る。


プレイヤーの観測データを
積み上げることで、
プレイヤーの経路を予測できる。


ここで
見失った！
プレイヤーを見失った場合でも、
最終観測地点から隠れている場所を
推測できる。

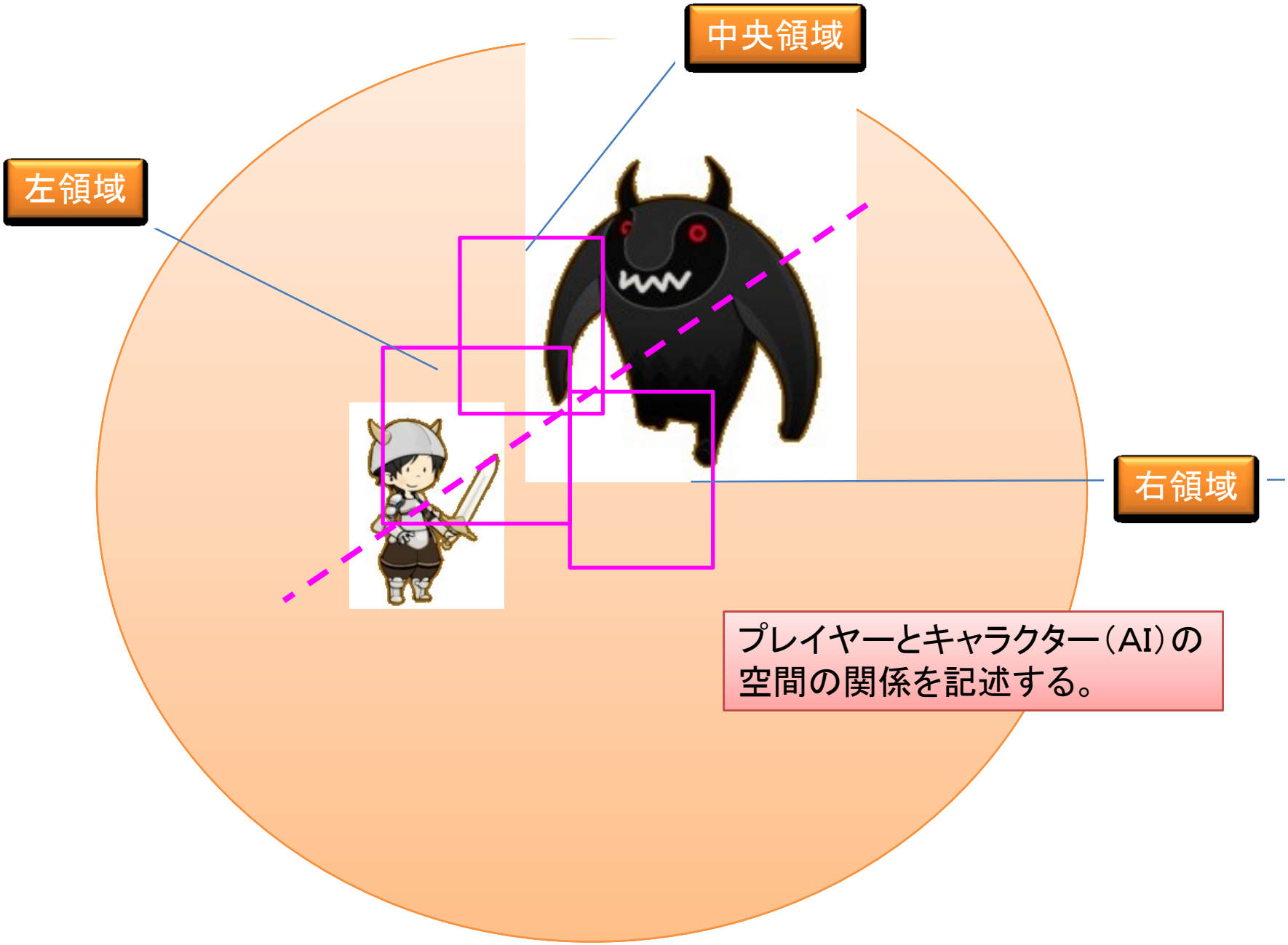

ここから
出てくるかも！
見失ったプレイヤーの再出現位置を
予測して、回り込む。

プレイヤーを認識しよう（近距離）。

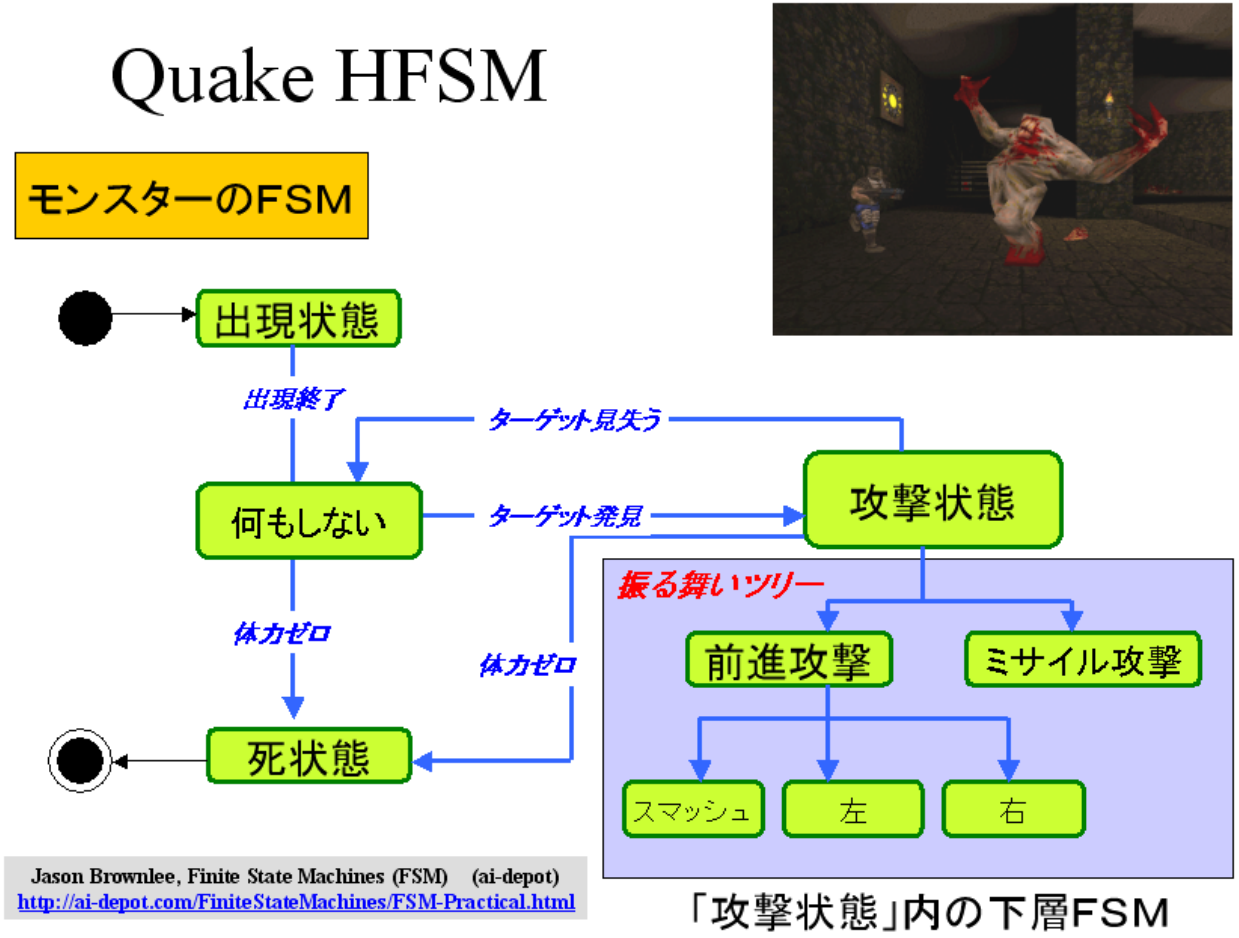
- 接近戦闘は短時間のエンターテインメント。
- 面白い「ちゃんばら」を実現する。
- そのためにどういう知識表現が必要か？

```
(例)
Struct Player_info
{
//近距離
Float  angle = 17.f;  // 自分と相手の軸線からの角度
bool  b_right_attack =false; //右パンチの射程内か？
bool  b_left_attack =true; //左パンチ射程内か？
Bool  b_magic_distance = false ; //魔法届くか？
flaot  _hp = 46;   //相手に与えたダメージ
float   rhp = 72:  //相手から与えられたダメージ
Unsigned int num_kill = 3;  //プレイヤーに倒された数
…

//中距離
float _time: // 観測時刻
float[4]  _loc= (4.5,31.4,343.f,1.f);   //位置
float _velocity = 3.2f;  //速度
bool  b_armed = true: //武装してた？
bool  b_injured = falase ; // 負傷してた？
…
} ;
```

こういった情報を毎フレームか数フレームごとに取る。


中央領域
左領域
右領域
プレイヤーとキャラクター（AI）の
空間の関係を記述する。

(例) Quake HFSM



http://ai-depot.com/FiniteStateMachines/FSM-Practical.html

状態遷移図を用いる

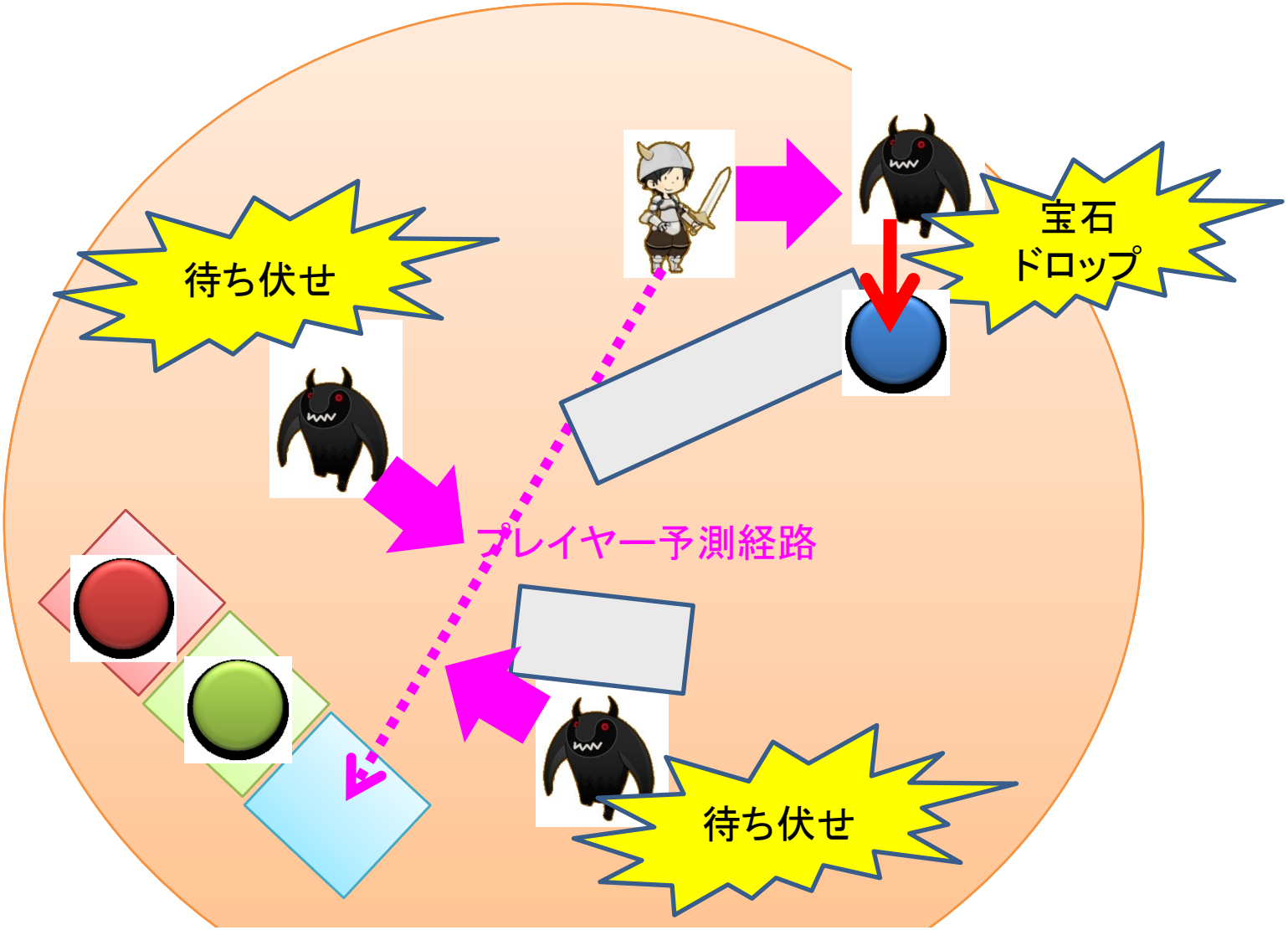
プレイヤーを認識しよう（遠距離）。

- 宝石ドロップに関して。
- あまり頻度多く、宝石をドロップしたくない。
- 現在の宝石の収集がどれぐらいか？

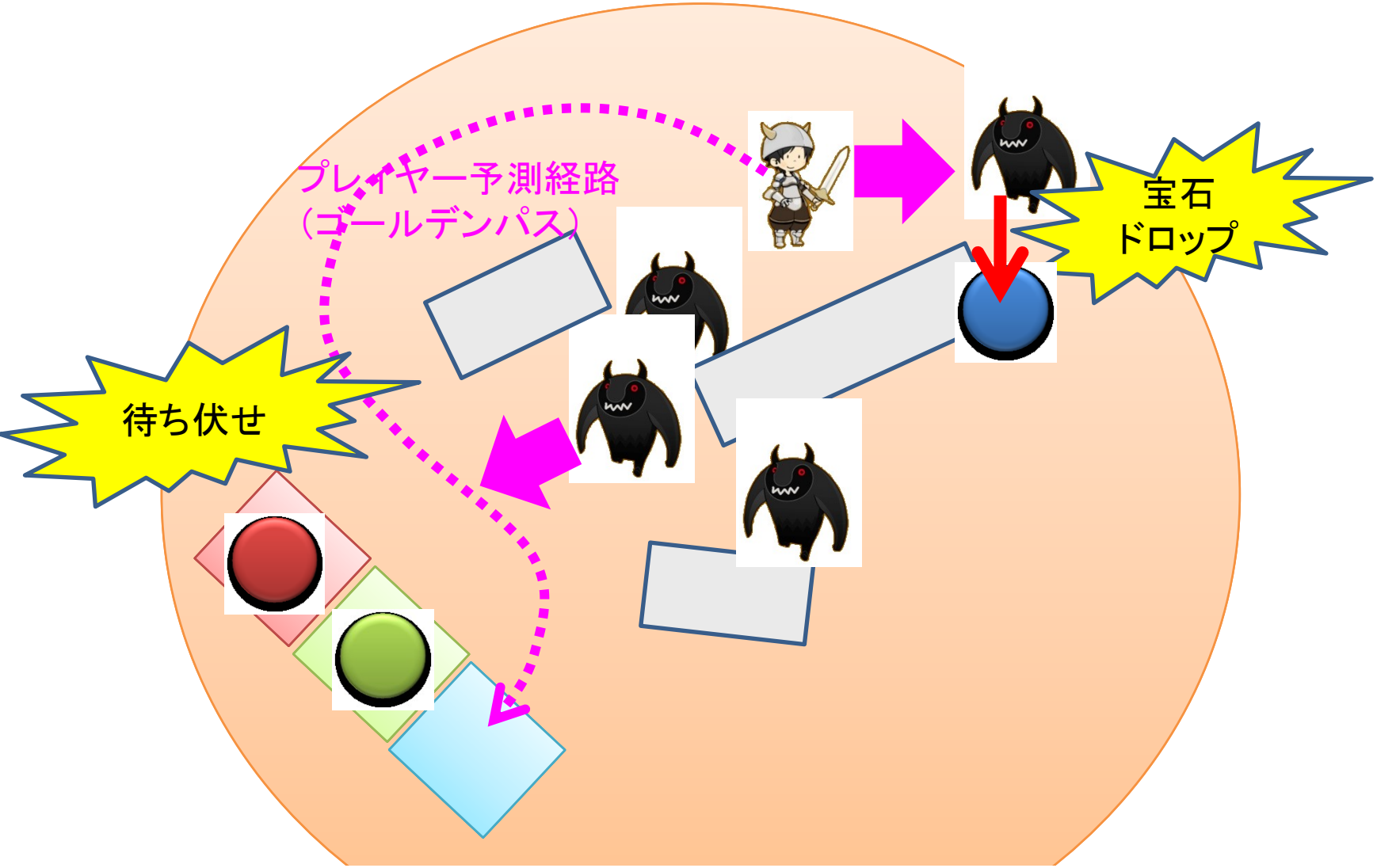
- 長い間、宝石をゲットしていなかったら、倒されたら宝石を落とす
- 宝石を持っていたら、待ち伏せして取り返す。

```
(例)
Struct Player_info
{
//近距離
foat  angle = 17.f;  // 自分と相手の軸線からの角度
bool  b_right_attack =false; //右パンチの射程内か？
bool  b_left_attack =true; //左パンチ射程内か？
Bool  b_magic_distance = false ; //魔法届くか？
flaot  _hp = 46;   //相手に与えたダメージ
float   rhp = 72:  //相手から与えられたダメージ
Unsigned int num_kill = 3;  //プレイヤーに倒された数
…

//中距離
float _time: // 観測時刻
float[4]  _loc= (4.5,31.4,343.f,1.f);   //位置
float _velocity = 3.2f;  //速度
bool  b_armed = true: //武装してた？
bool  b_injured = falase ; // 負傷してた？
…

//遠距離
int  _last_time_get_jewels  =  133 //最後にアイテムをゲットした時刻
Int   num_empty_jewels  = 1; //残りアイテムスロット数
Bool b_has_jewel = true: // 宝石を運んでいる？   運んでいない？
….
};
```

こういった情報を毎フレームか数フレーム取る。



プレイヤーと本拠の間をパス検索して予測経路を計算する(ゴールデンパス)。
ゴールデンパスの近くにAIを集合させる。



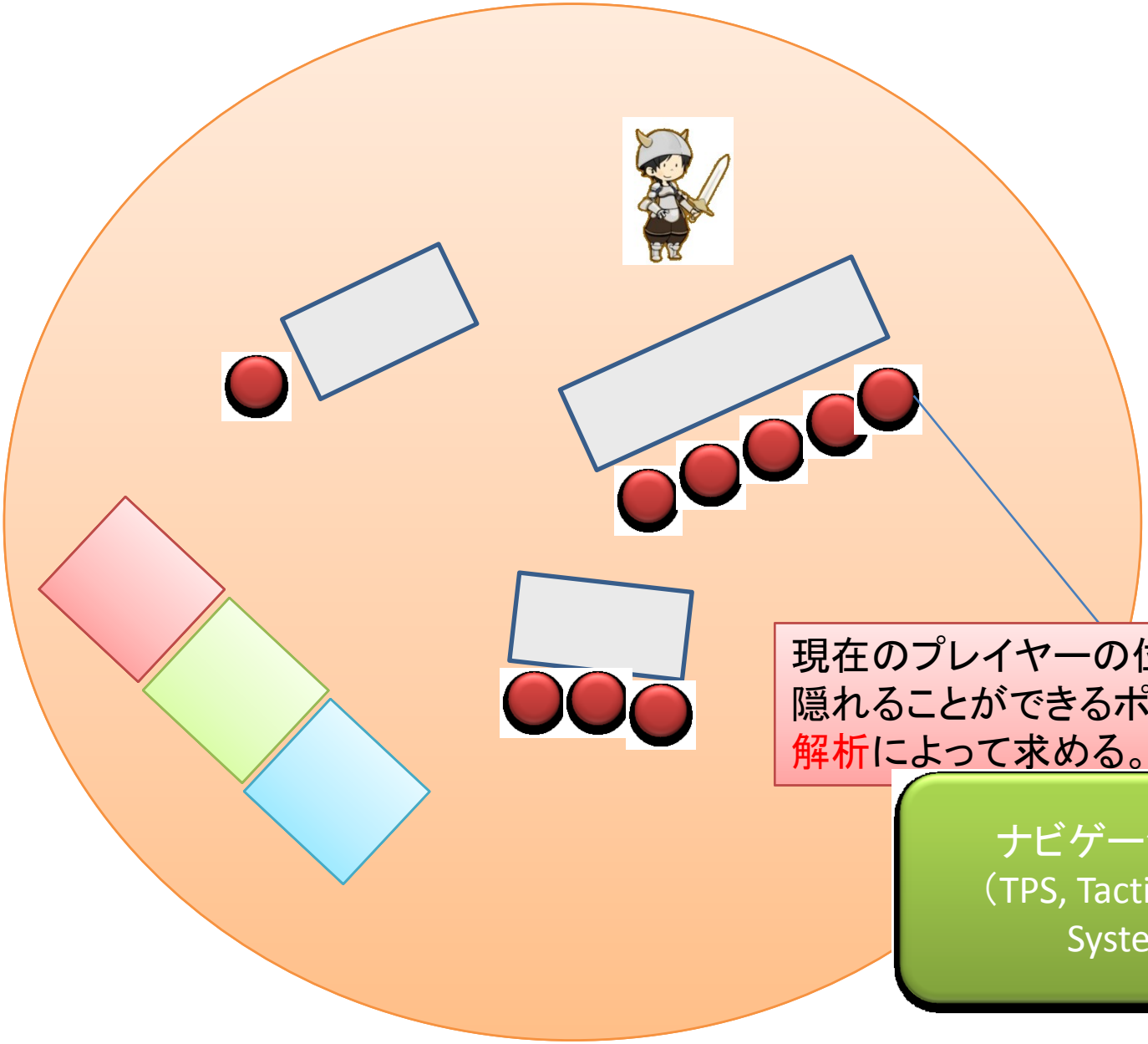
プレイヤーと本拠の間をパス検索して予測経路を計算する（ゴールデンパス）。
ゴールデンパスの近くにAIを集合させる。

第二部　第二章

ゲームを作ろう：ナビゲーションAI編

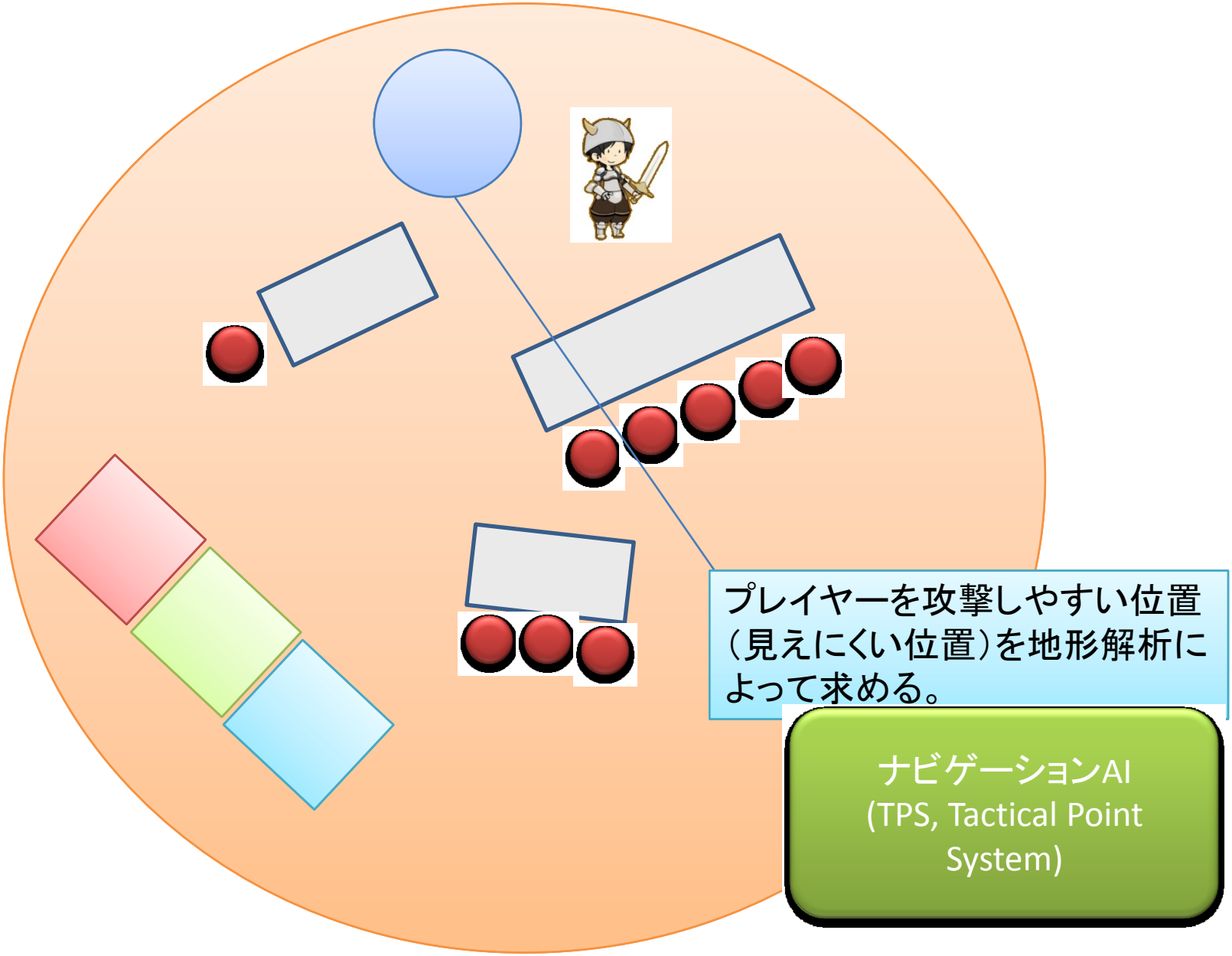
ナビゲーションAI

- ゲームステージを解析して、キャラクターに必要な位置情報を与える。



現在のプレイヤーの位置に対して隠れることができるポイントを地形解析によって求める。

ナビゲーションAI
（TPS, Tactical Point System）


プレイヤーを攻撃しやすい位置（見えにくい位置）を地形解析によって求める。
ナビゲーションAI
(TPS, Tactical Point System)

TPS(Tactical Point System)
= 動的に必要なポイントを計算する

“Lichdom”

Tactical Point System (CryEngine)
http://docs.cryengine.com/display/SDKDOC4/Tactical+Point+System

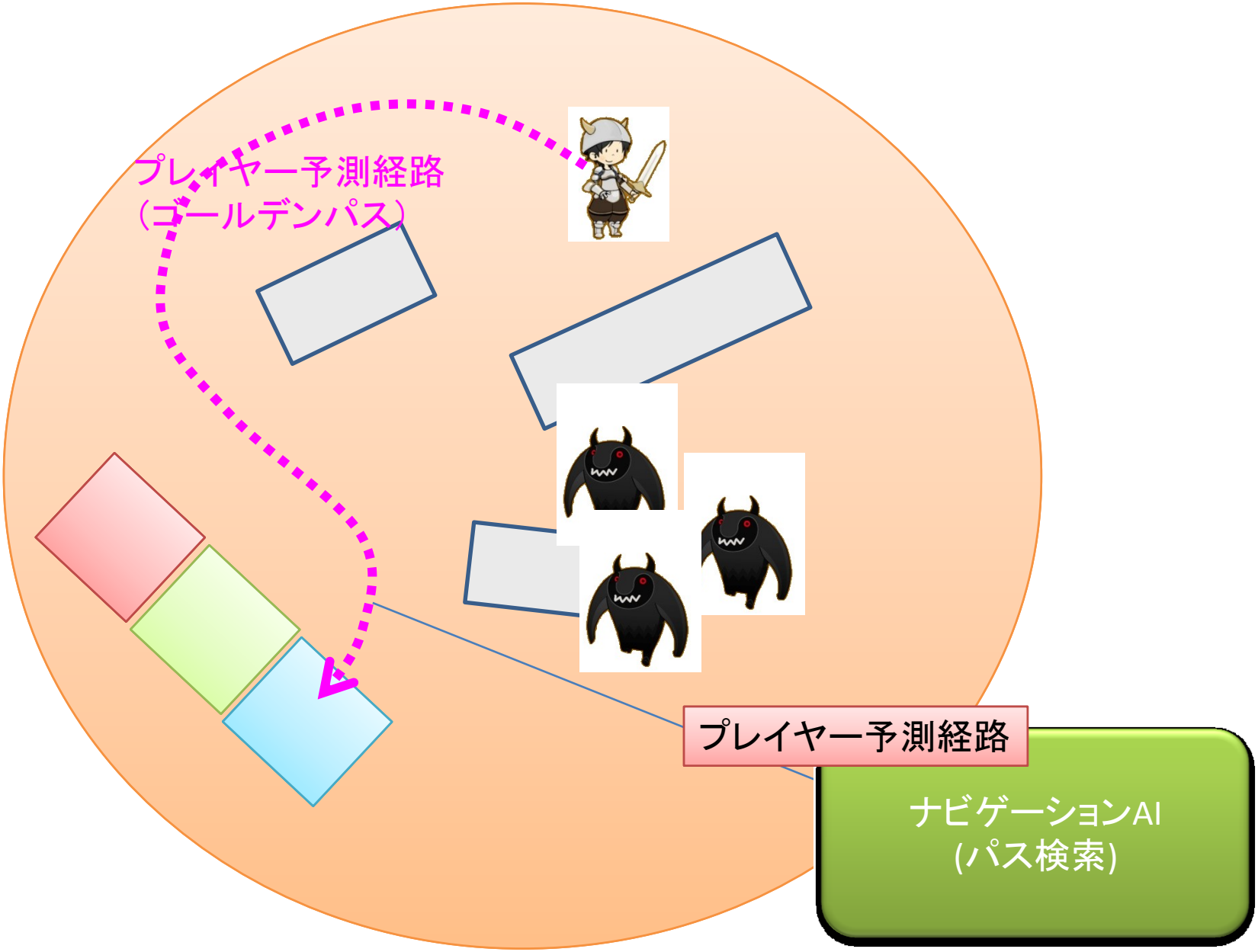
Spaces in the Sandbox: Tactical Awareness in Open World Games
Matthew Jack, Mika Vehkala
Moon Collider, IO Interactive A/S
http://gdcvault.com/play/1018136/Spaces-in-the-Sandbox-Tactical

TPS(Tactical Point System)
= 動的に必要なポイントを計算する

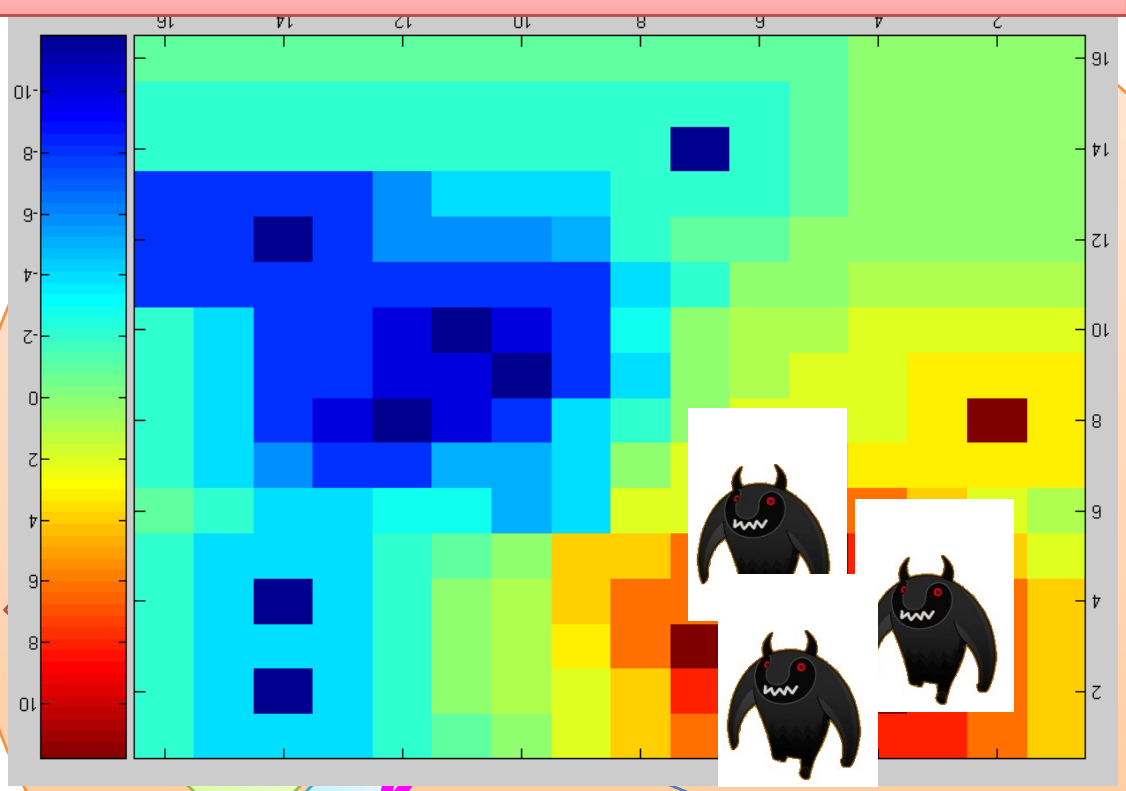
“Lichdom”

Tactical Point System (CryEngine)
http://docs.cryengine.com/display/SDKDOC4/Tactical+Point+System

Spaces in the Sandbox: Tactical Awareness in Open World Games
Matthew Jack, Mika Vehkala
Moon Collider, IO Interactive A/S
http://gdcvault.com/play/1018136/Spaces-in-the-Sandbox-Tactical


プレイヤー予測経路
(ゴールデンパス)
プレイヤー予測経路
ナビゲーションAI
(パス検索)

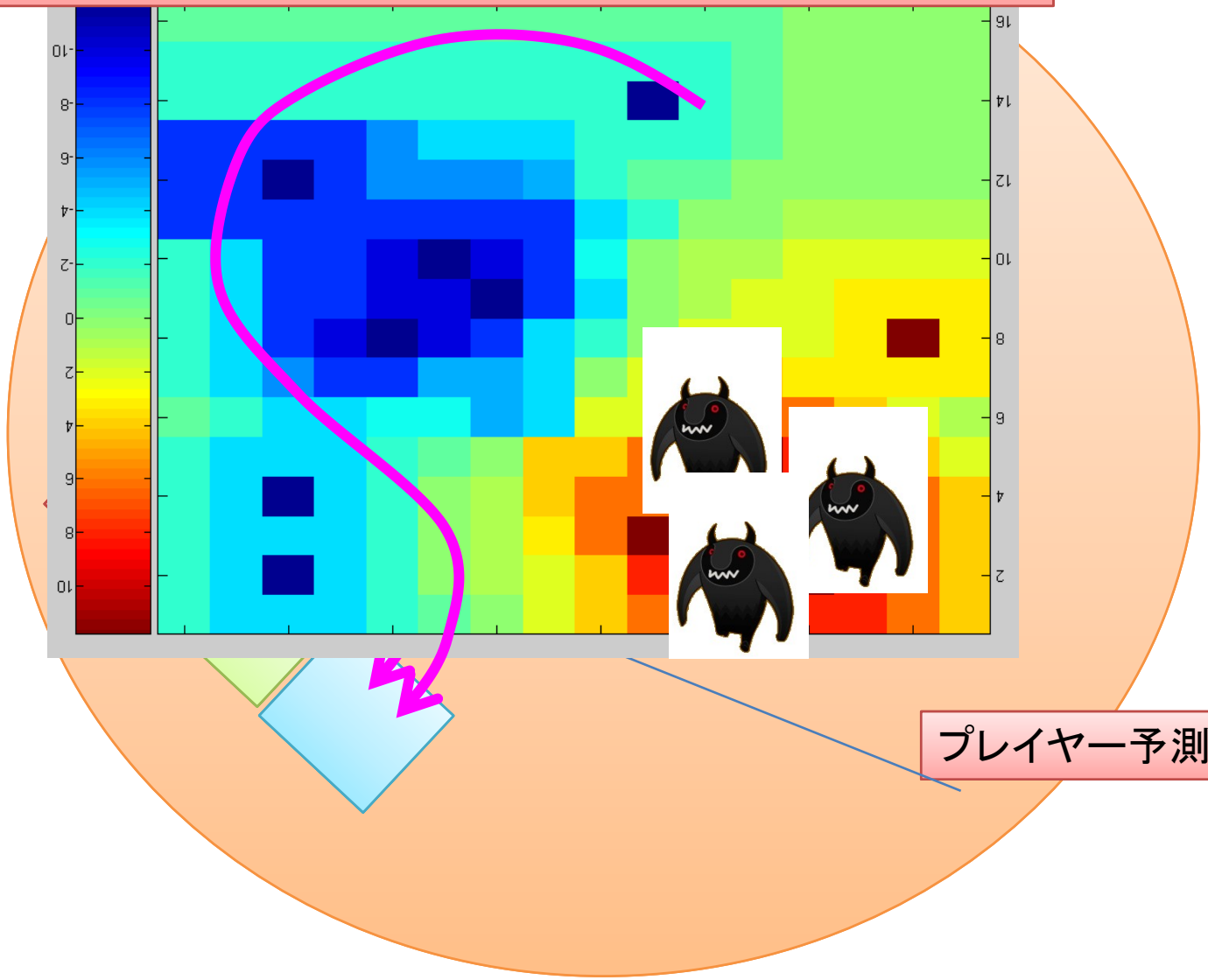
影響マップ＝敵の勢力を数値化したもの。赤＝強い。青＝弱い。



プレイヤー予測経路

影響マップ＝敵の勢力を数値化したもの。赤＝強い。青＝弱い。

敵勢力をパス計算のコストに載せると、敵勢力を避けるような
パスが計算される。

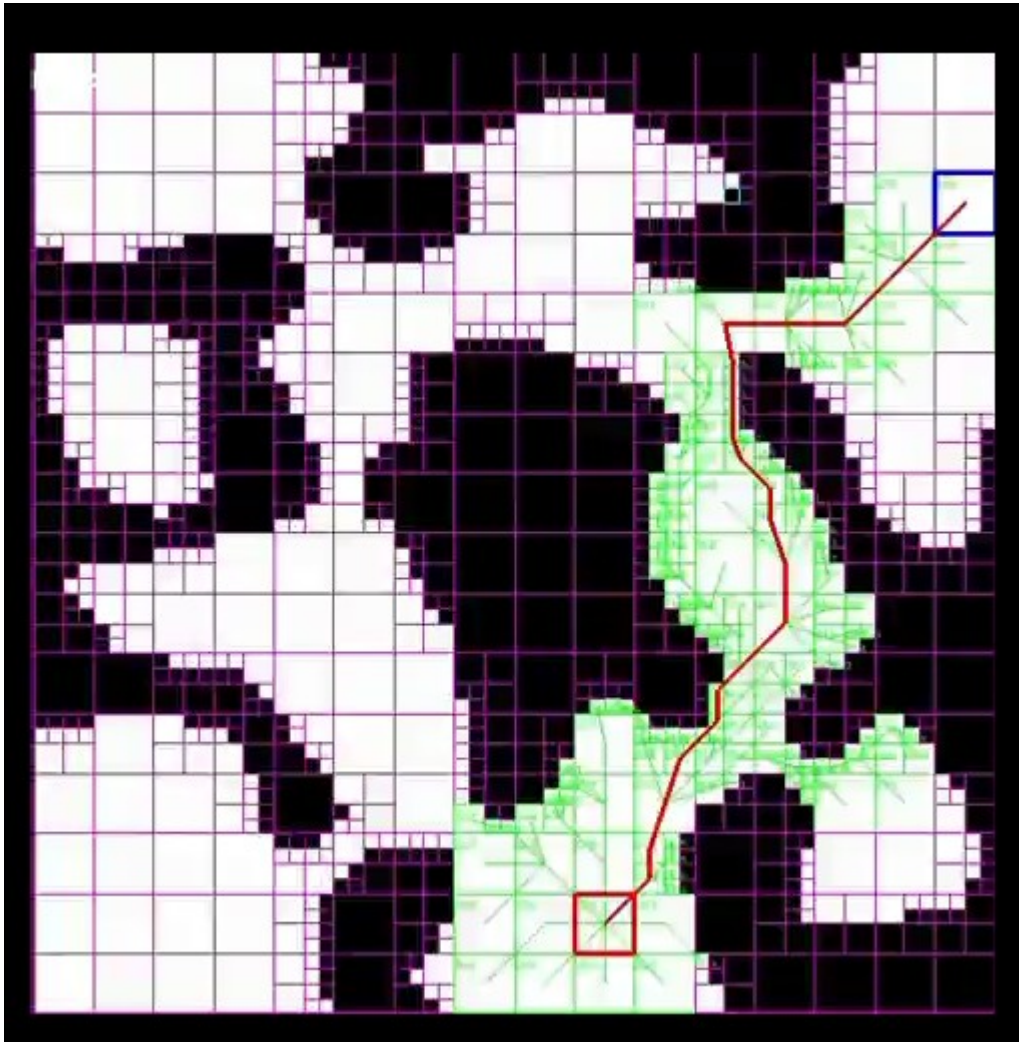


On the AI Strategy for KILLZONE 2's Multiplayer Bots
http://aigamedev.com/open/coverage/killzone2/

エリア（ウエイポイントをクラスター化したもの）ごとの
敵勢力のパラメーター表示。赤＝強い（危険）。青＝弱い（安全）。

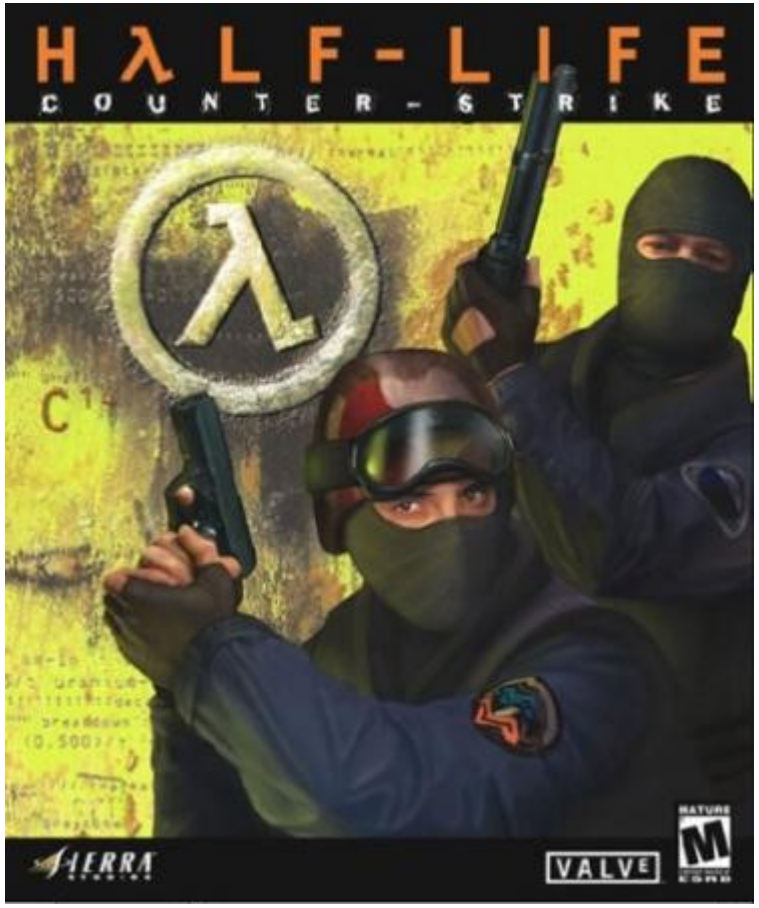
パス検索

＝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



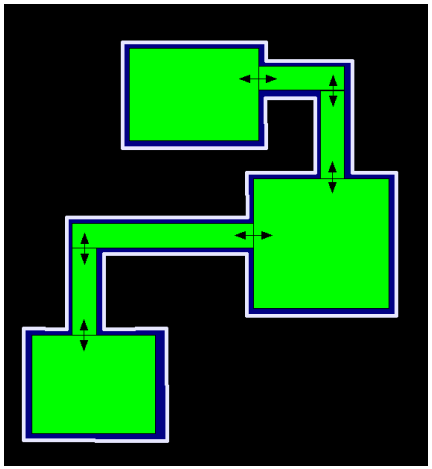
https://www.youtube.com/watch?v=95aHGzzNCY8

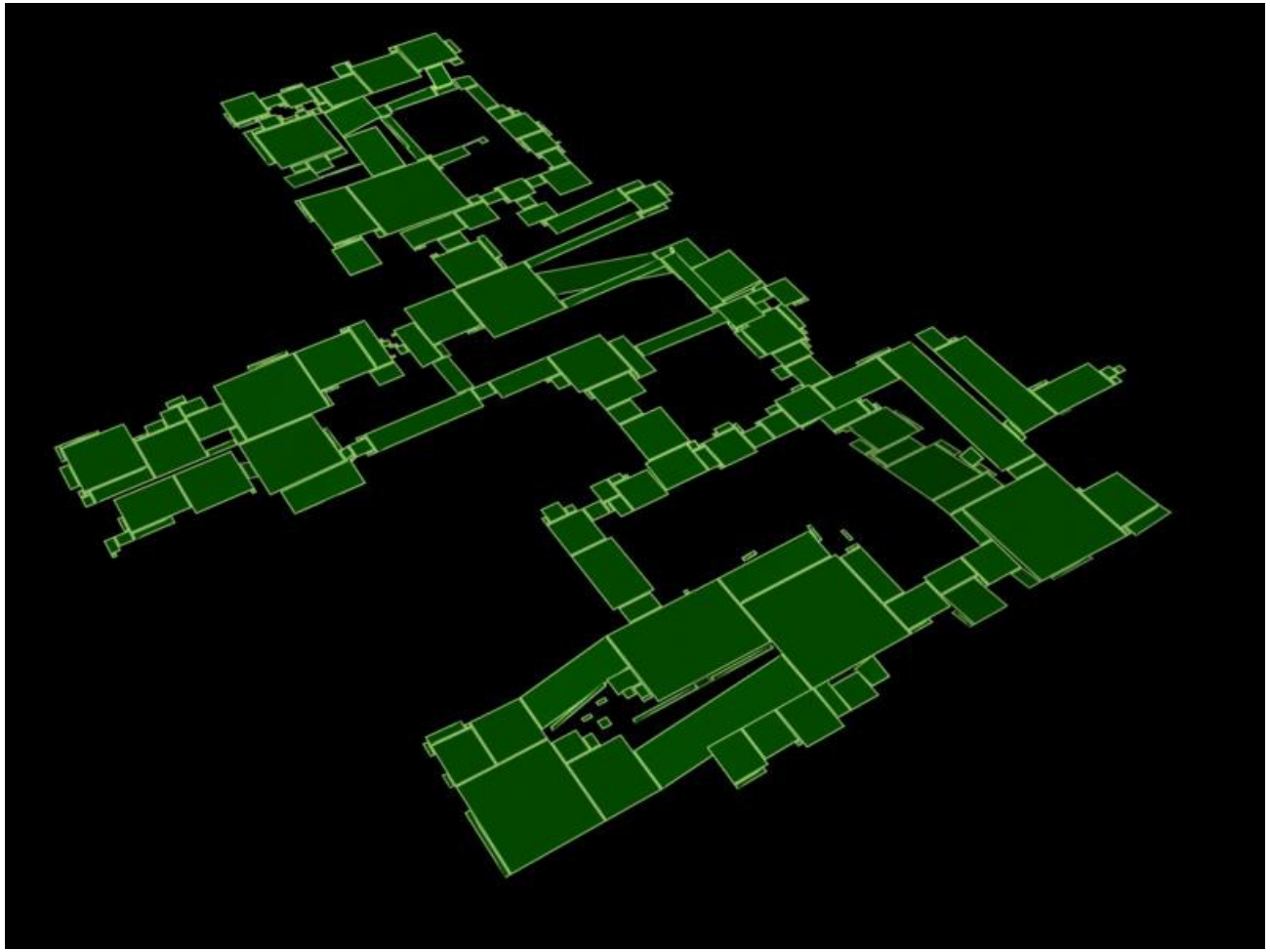
Counter Strike (2000)



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

自由に移動できる





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

Path Following (デモ)

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

Dragon Age

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

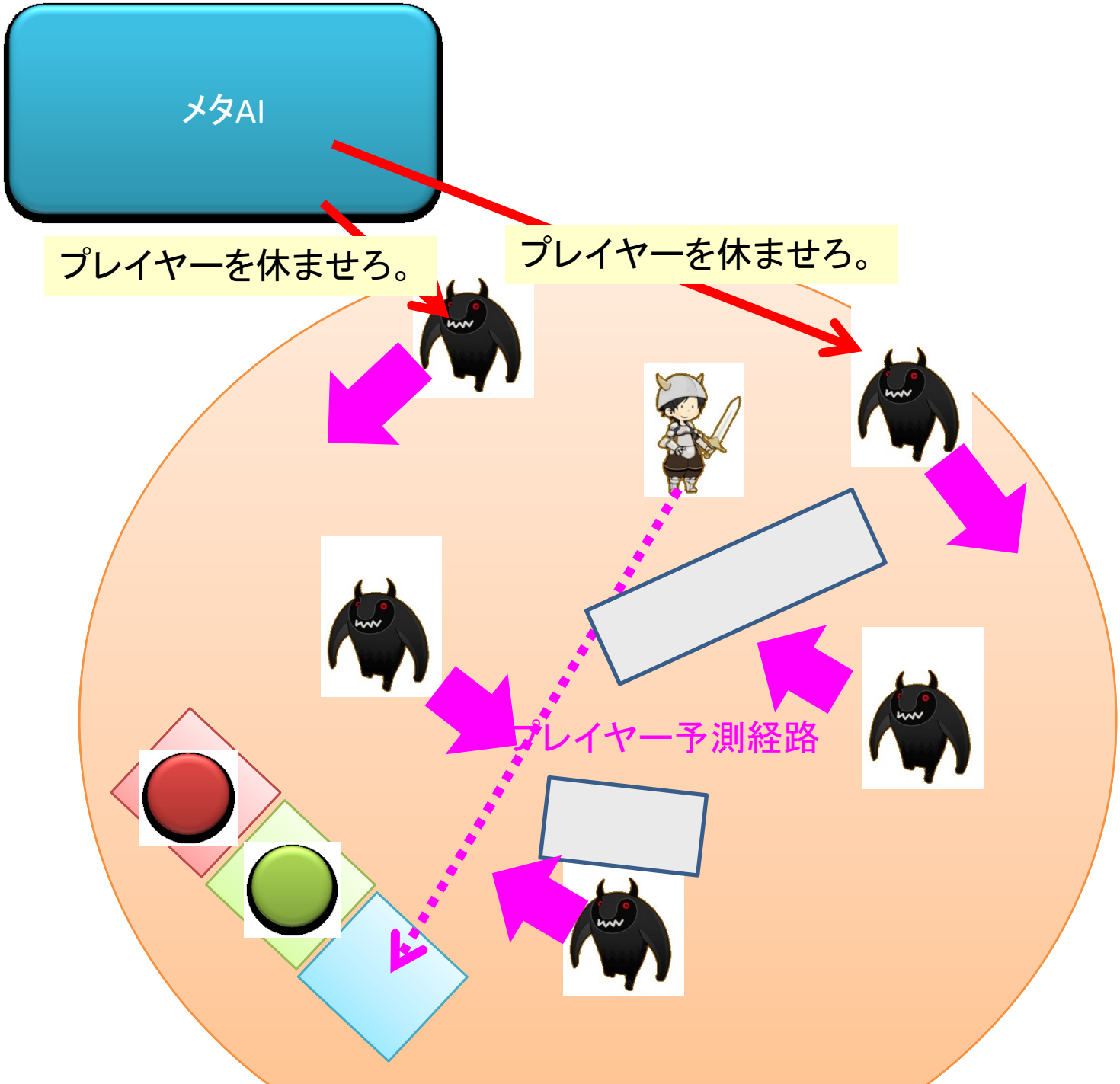
Dragon Age pathfinding program put to the test（University of Alberta）
https://www.youtube.com/watch?v=I7YQ5_Nbifo

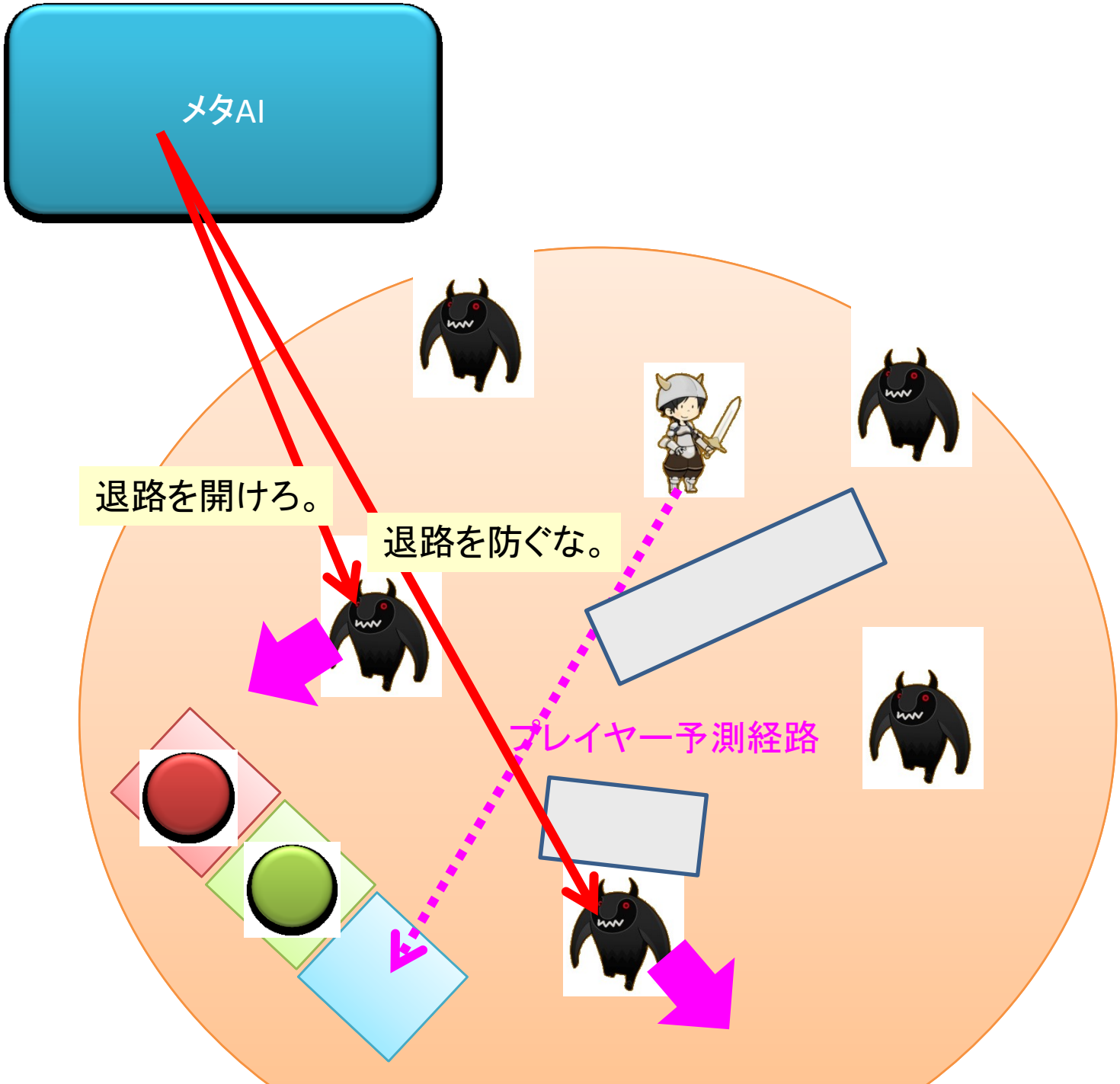
第二部　第三章

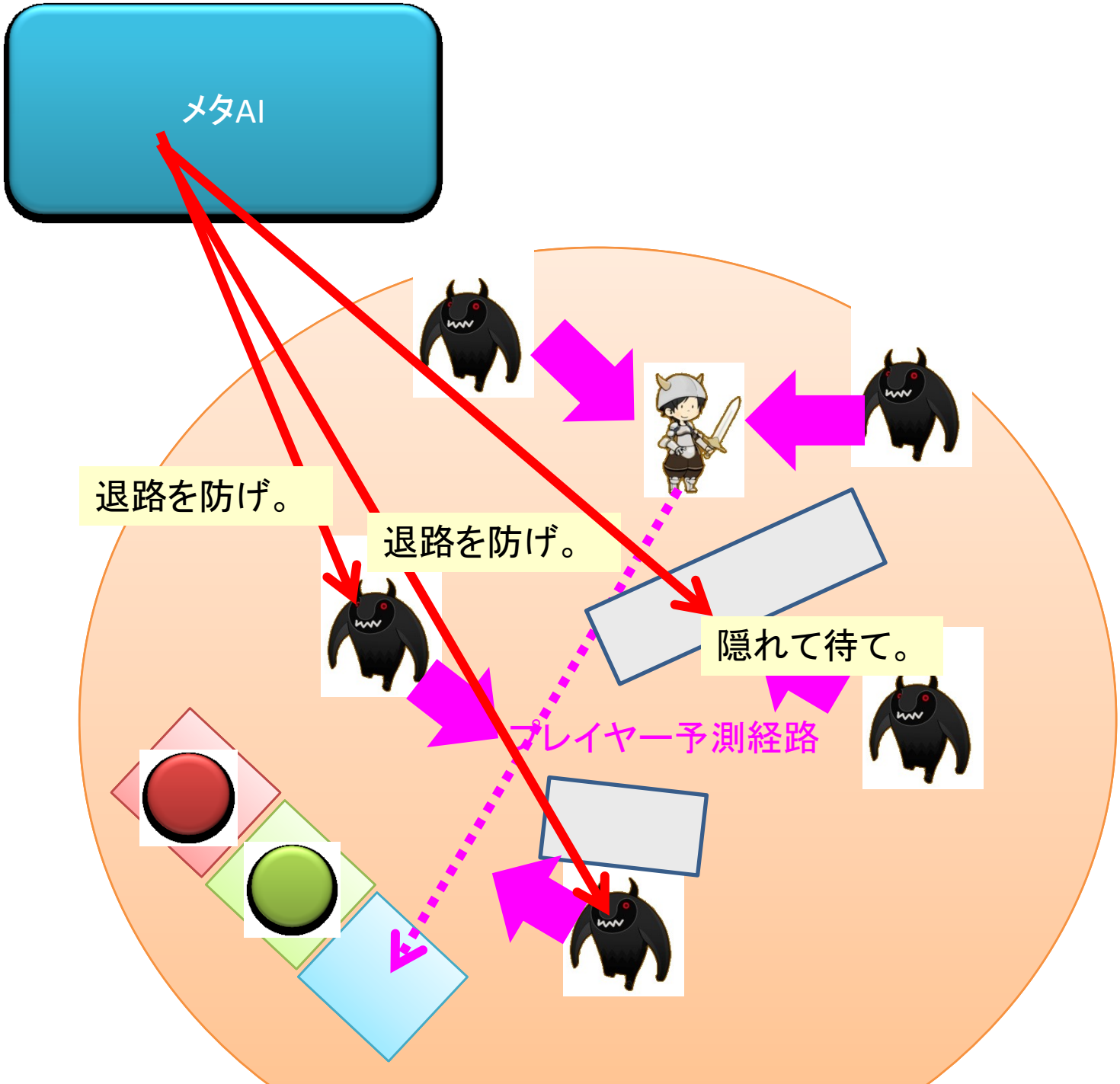
ゲームを作ろう：メタAI編

メタAI

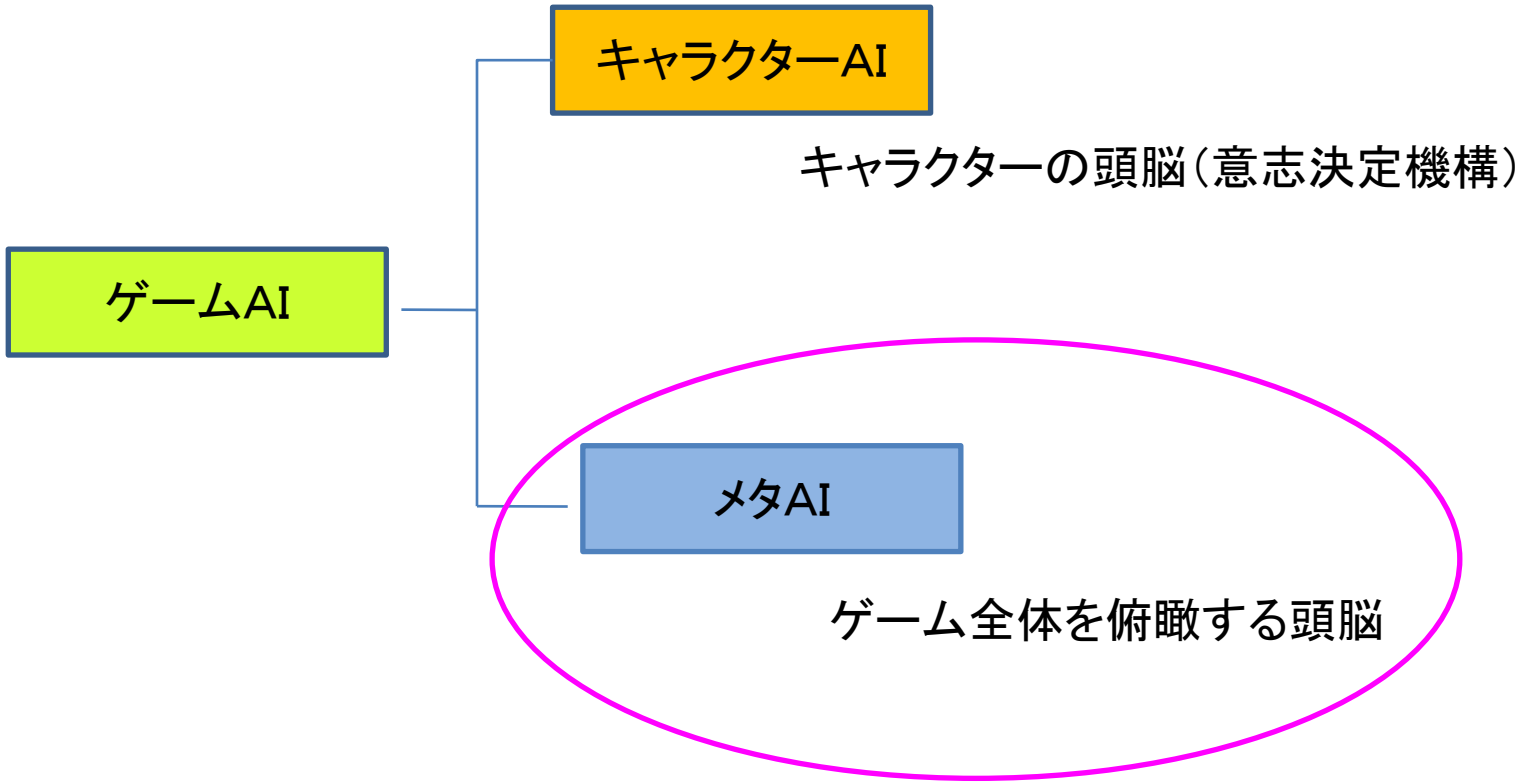
- ゲーム状況を監視し、ゲームの流れを作る人工知能
- ゲーム状況とはここでは、ステージの状態やユーザーの進行具合、スキルなどを言う。


メタAI
プレイヤーを休ませろ。
プレイヤーを休ませろ。
プレイヤー予測経路

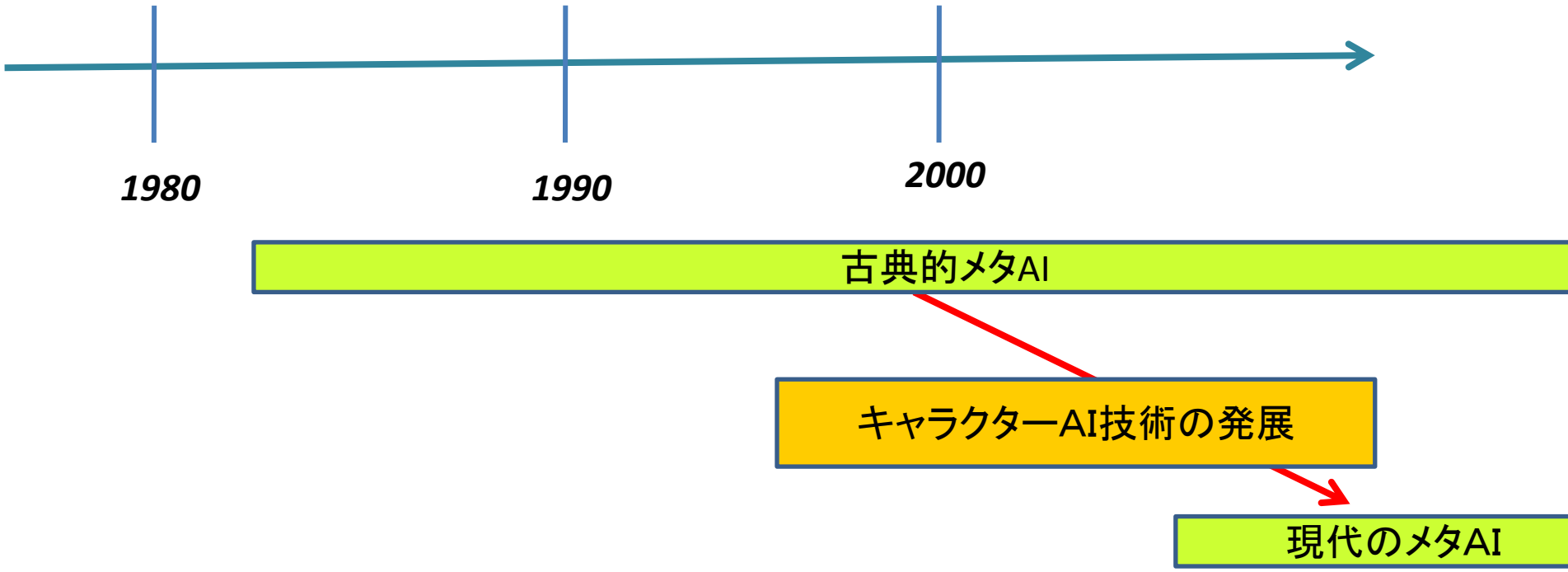

メタAI
退路を開けろ。
退路を防ぐな。
プレイヤー予測経路


メタAI
退路を防げ。
退路を防げ。
隠れて待て。
プレイヤー予測経路

メタAI



メタAIの歴史



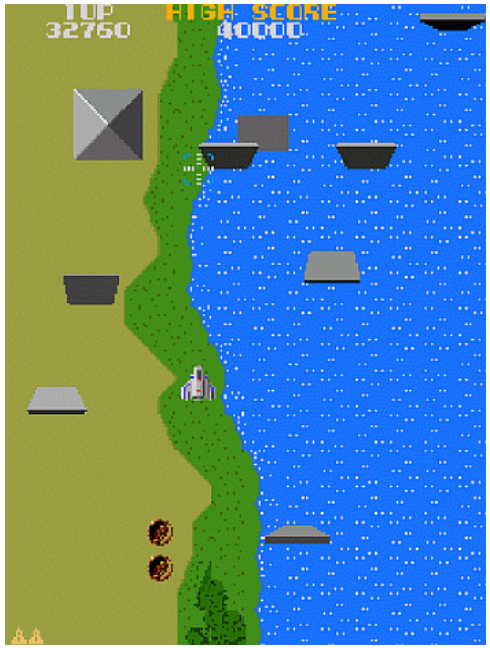
古典的メタAI = レベル調整AI

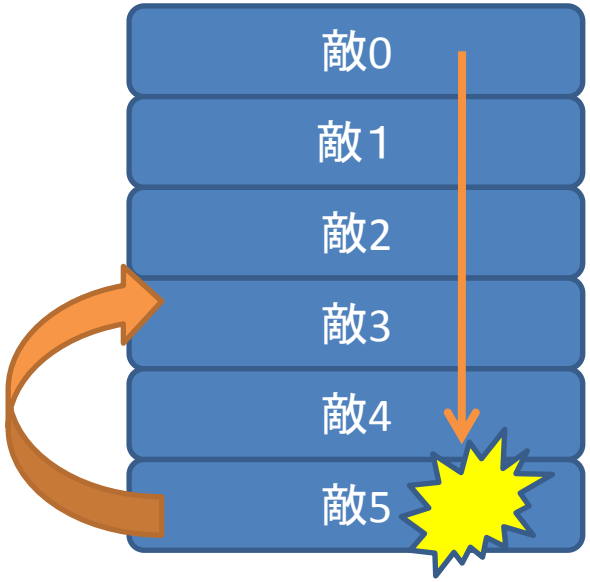
*岩谷徹氏：ファンと一般のユーザーを満足させる方法のひとつは人工知能AIのような考え方です。**プレーヤースキルをプログラム側から判断して、難易度を調整していくというものです。これを私はセルフゲームコントロールシステムと呼んで10年以上前から開発に使っています。***

ー International Game Designers Panel ー

http://game.watch.impress.co.jp/docs/20050312/gdc_int.htm

（例）「ゼビウス」　敵出現テーブル巻き戻し





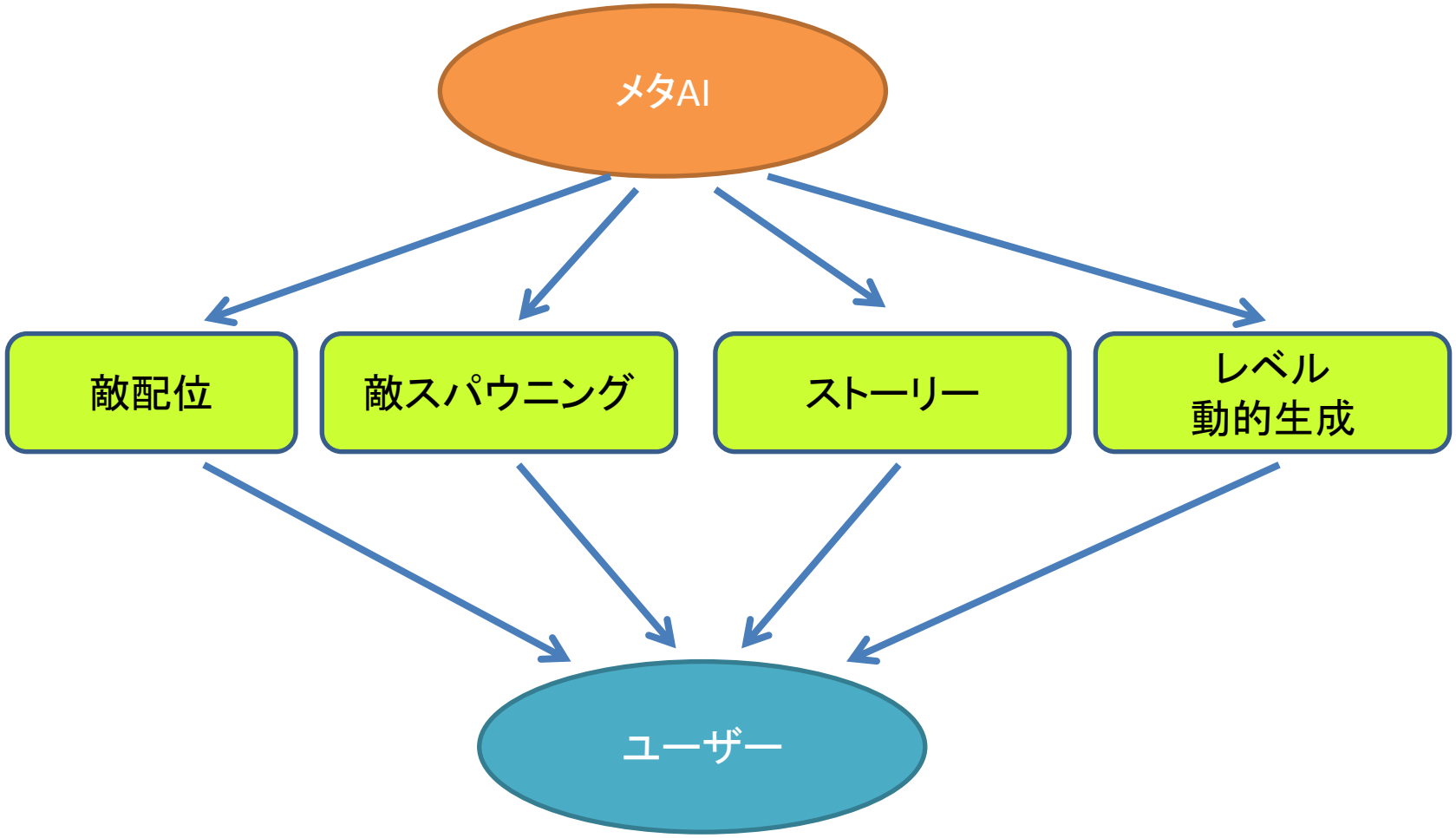
*遠藤雅伸氏　あと面白い機能なんですけれど、ゼビウスには非常に簡単なAIが組み込まれています。**「プレイヤーがどれくらいの腕か」というのを判断して、出てくる敵が強くなるんです。***

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いている時に、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)。

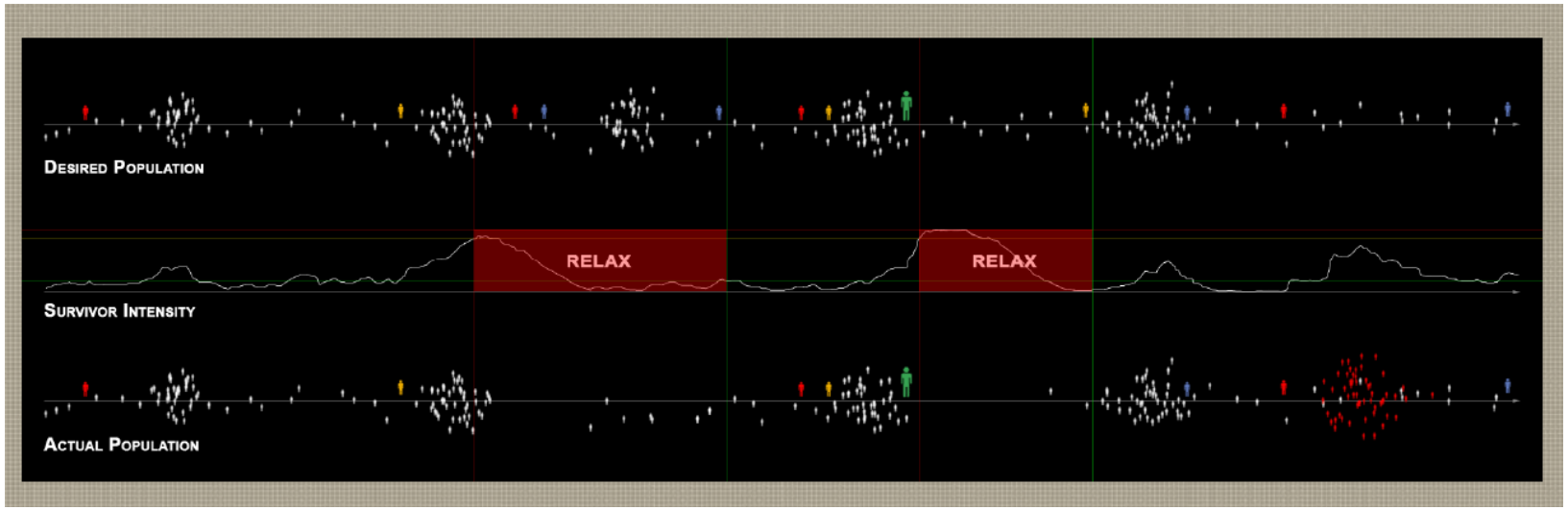
Michael Booth, "**The AI Systems of Left 4 Dead**," Artificial Intelligence and Interactive Digital Entertainment Conference at Stanford.
http://www.valvesoftware.com/publications.html

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

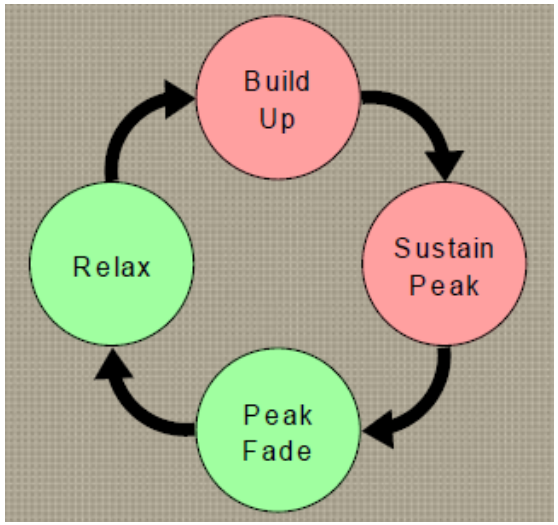
計算によって
求められた
理想的な敵出現数

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。

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

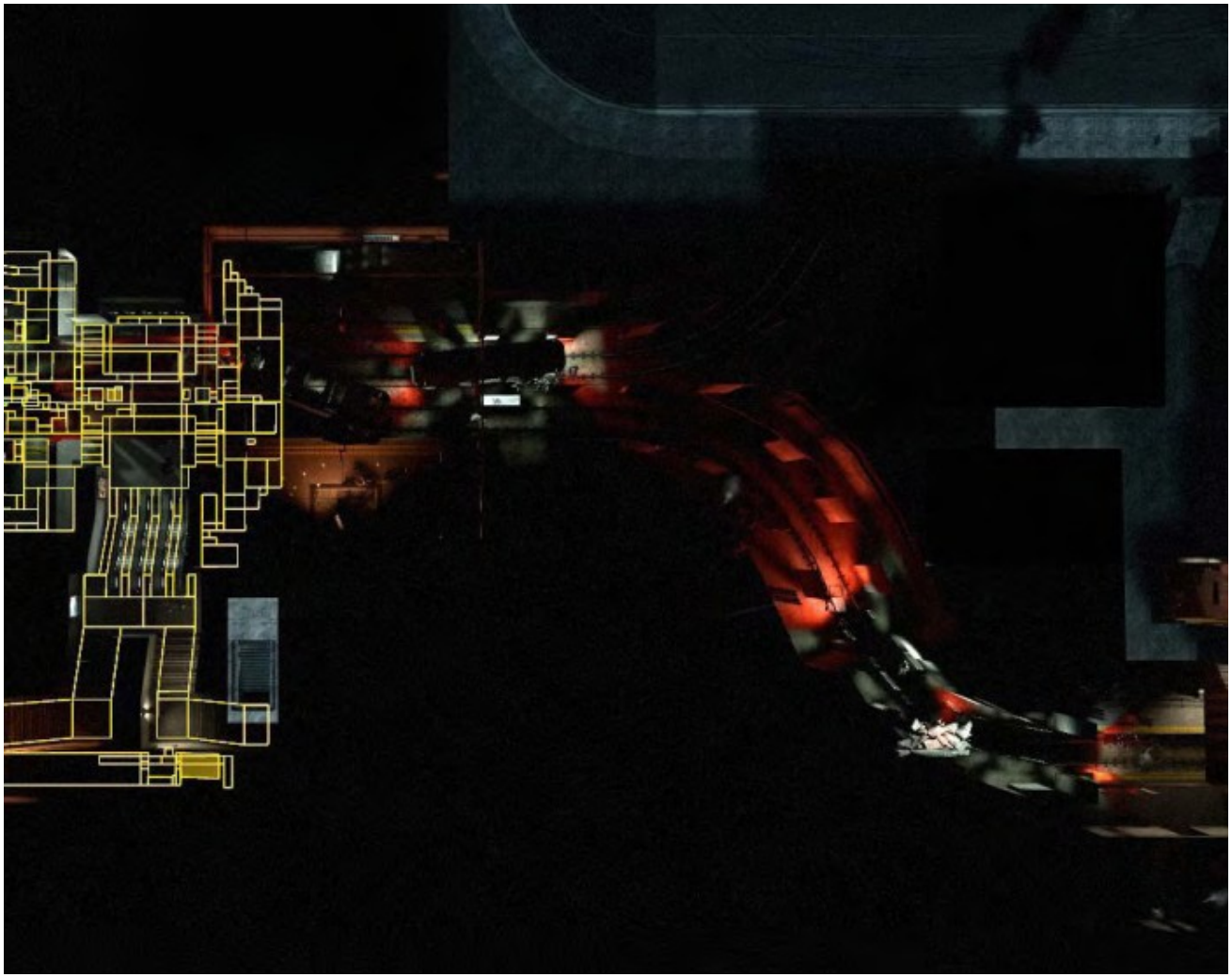
＝　キャラクターAIが環境を認識する方法

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。

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)



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)

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

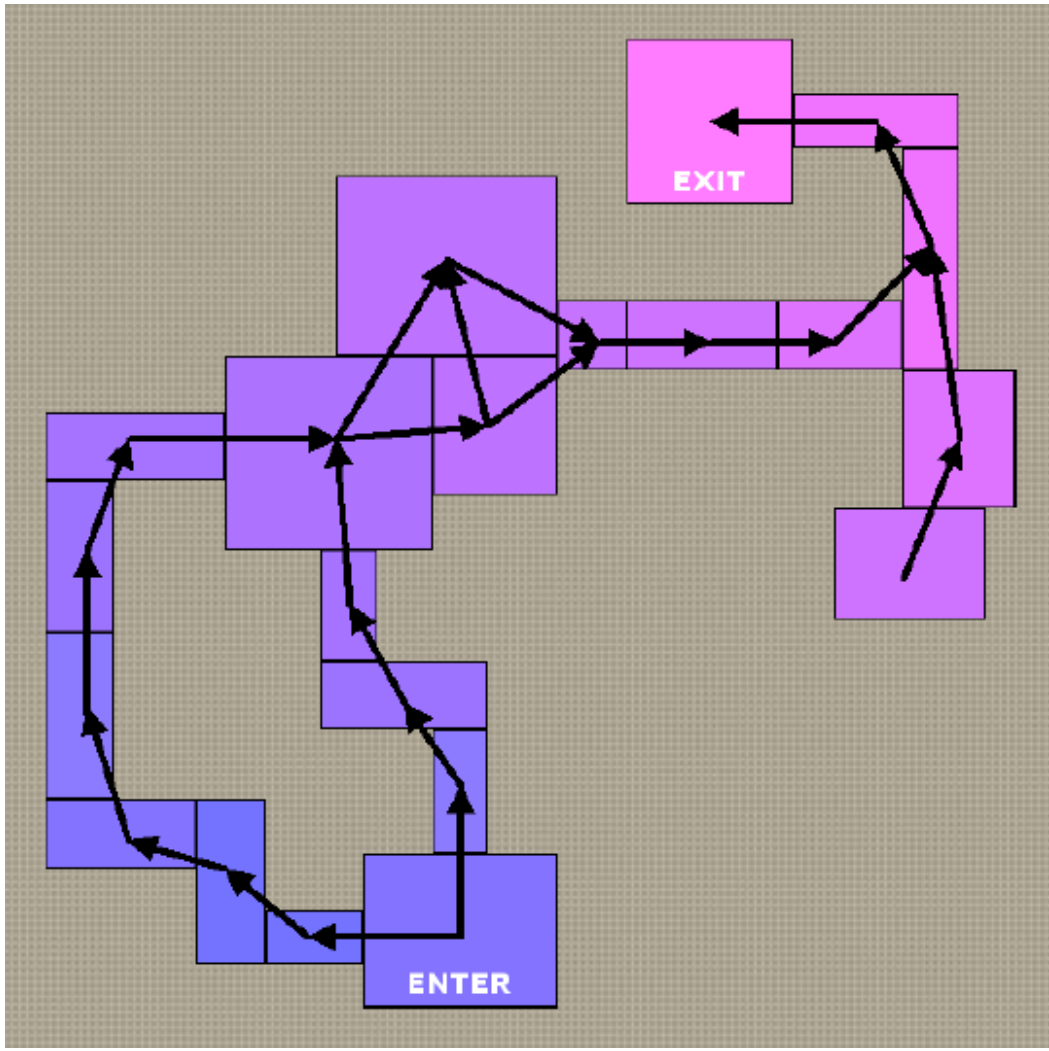
メタAIが作用を行う領域

(AAS=Active Area Set)

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)

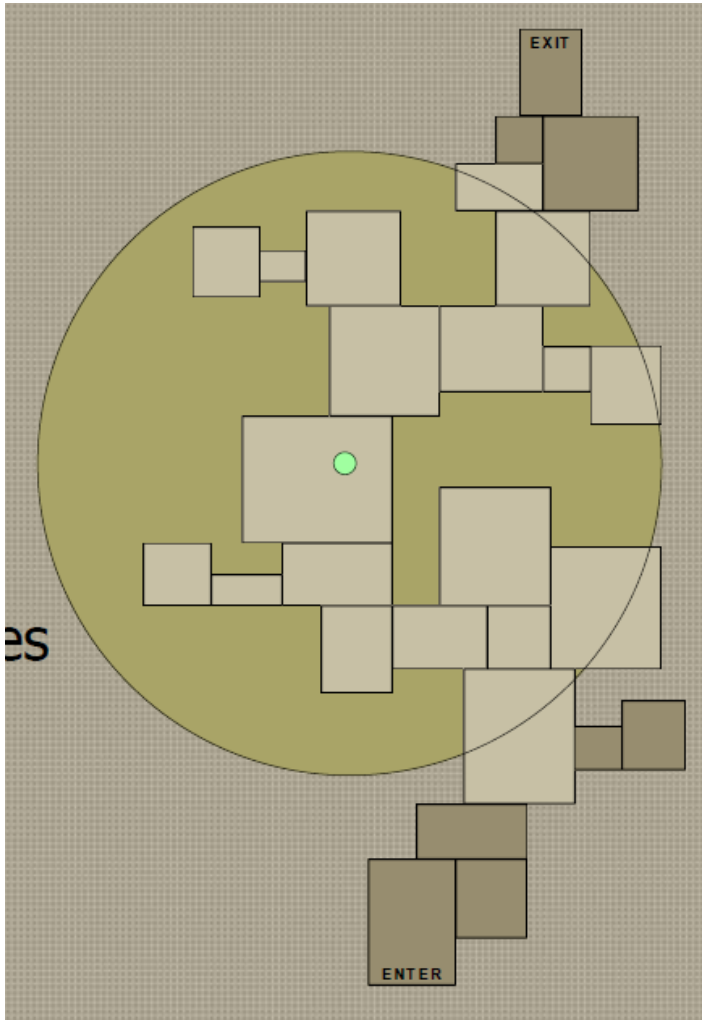
プレイヤー群の経路をトレース・予測
- どこへ来るか
- どこが背面になるか



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

AAS に対して行うこと。

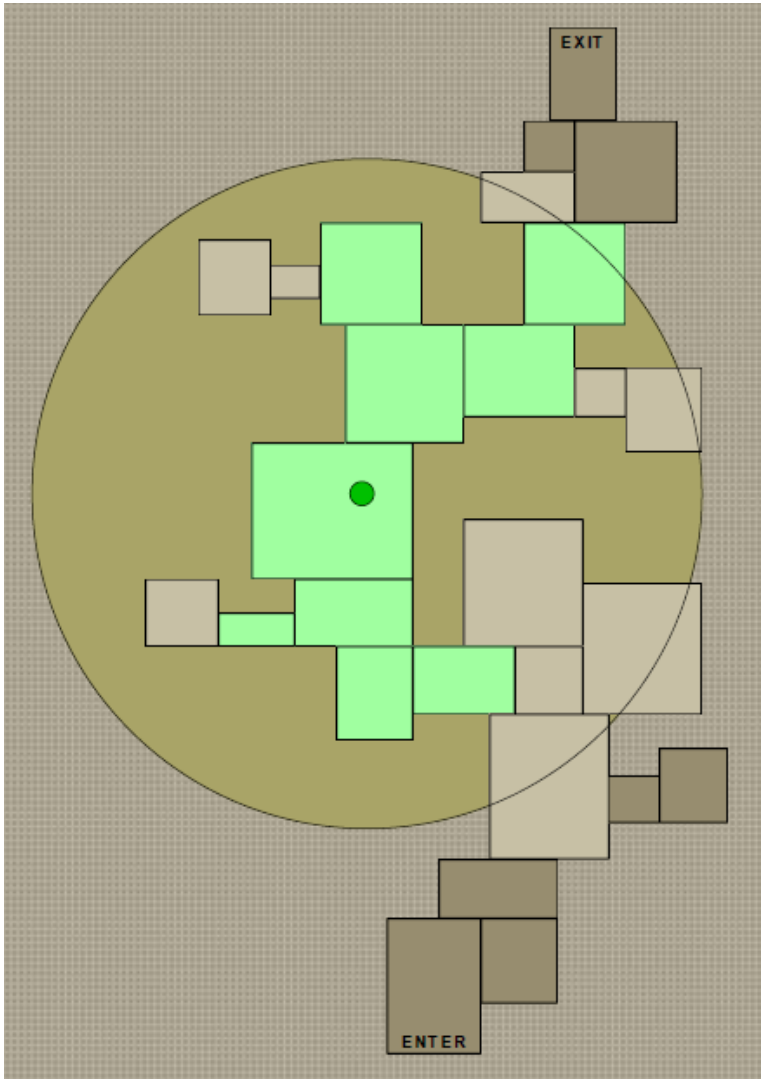
メタAIは AAS に移動に伴い、
敵の群れを生成・消滅させたりする。



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

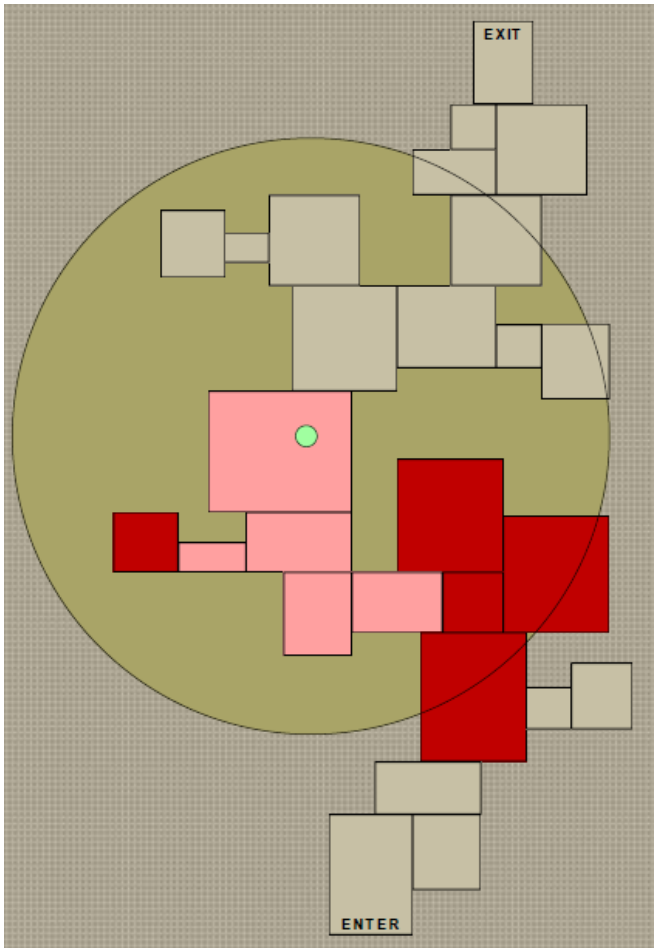
プレイヤーからの可視領域

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。

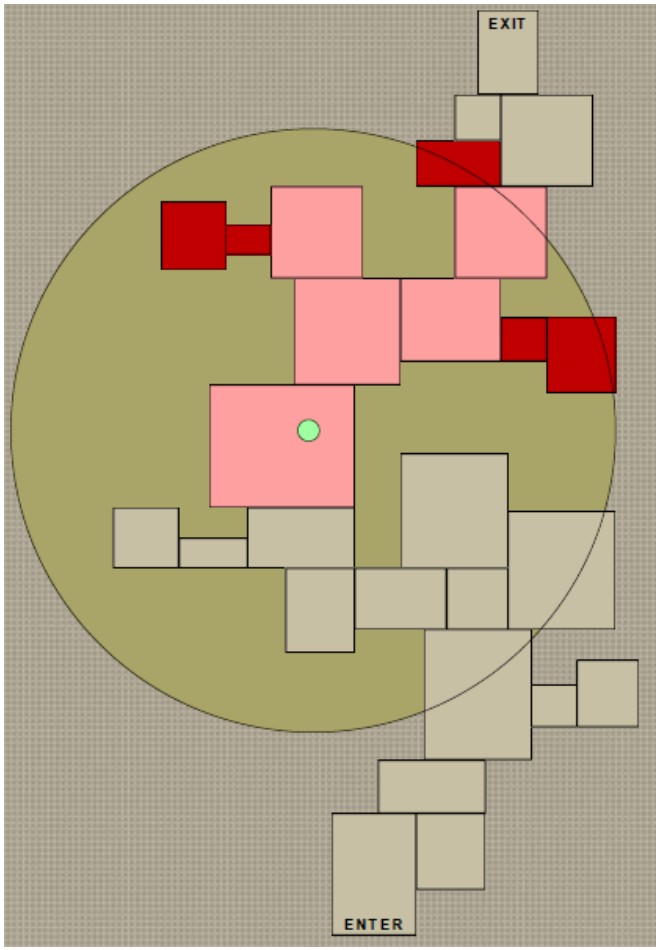


Michael Booth, "**The AI Systems of Left 4 Dead**," Artificial Intelligence and Interactive Digital Entertainment Conference at Stanford.
http://www.valvesoftware.com/publications.html

敵出現領域



背後



前方

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

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

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

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

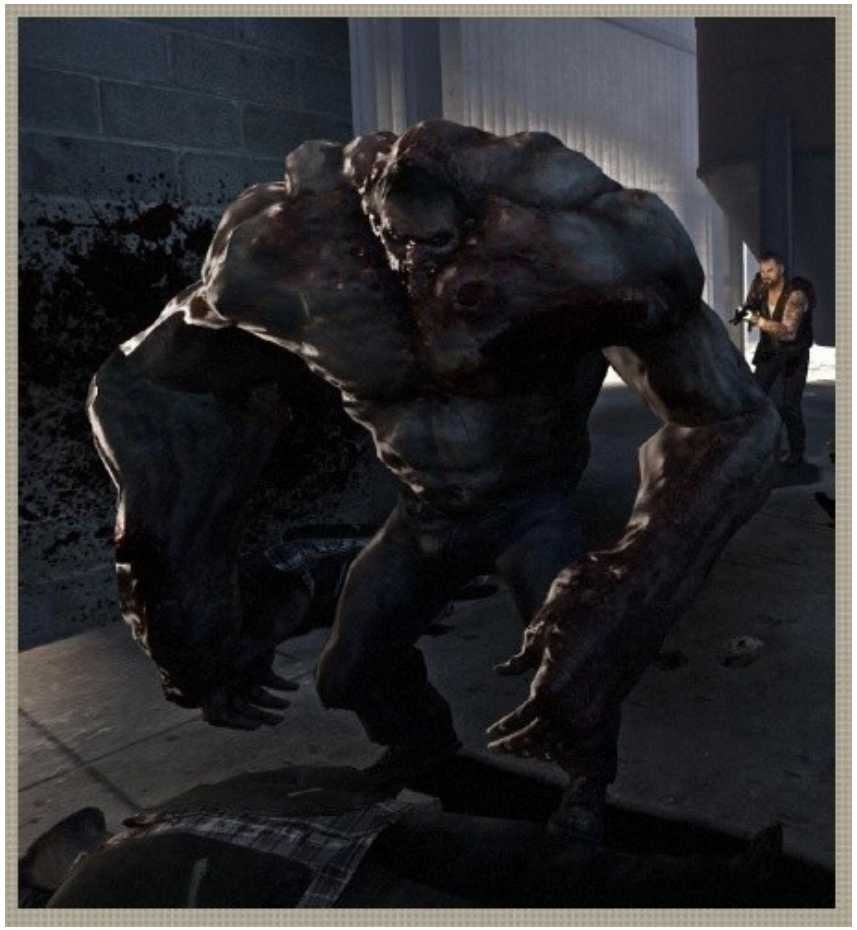
ボス出現アルゴリズム

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

(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

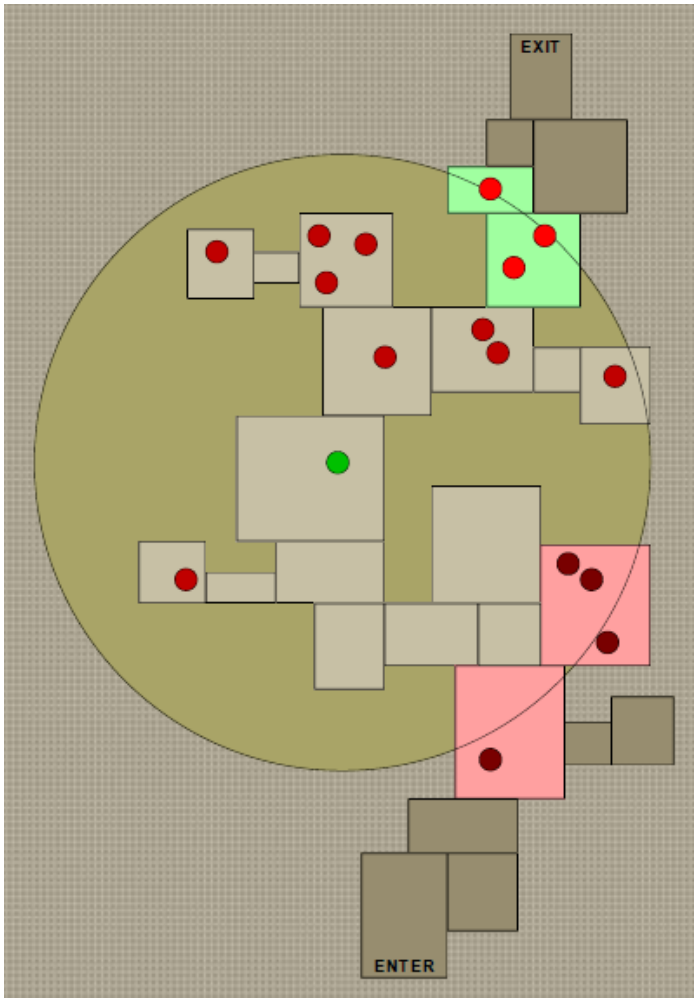
(1) 各エリアに、出現数 N を決定する

(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。

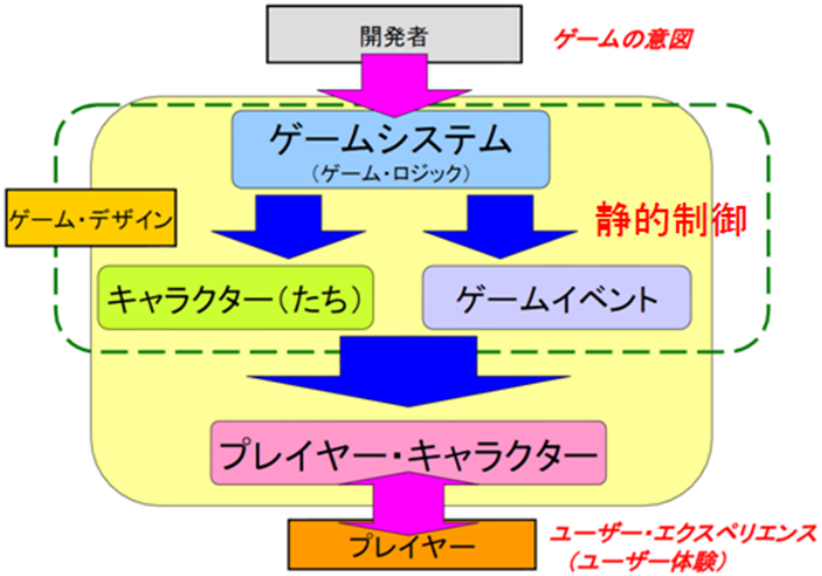
(5) Nはそのエリアがプレイヤーから見えている場合、或いは、プレイヤーがリラックスモードの場合には、強制的に0になる。



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

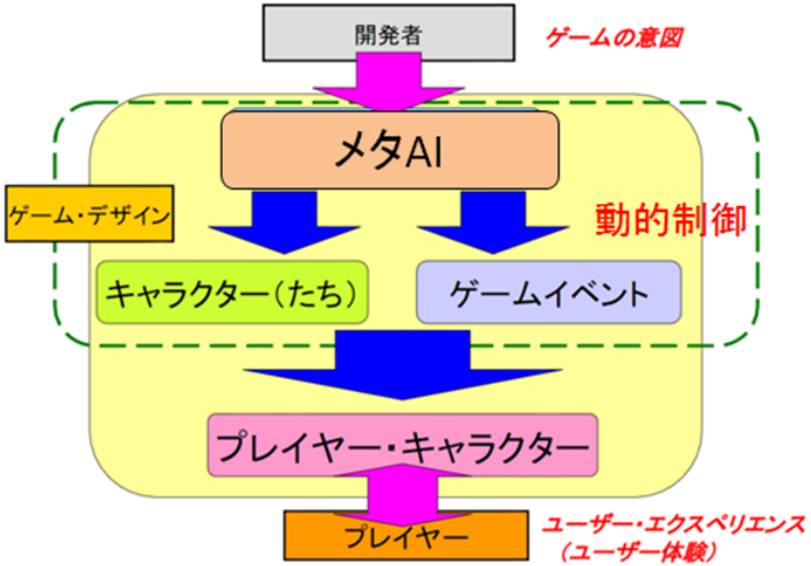
まとめ

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



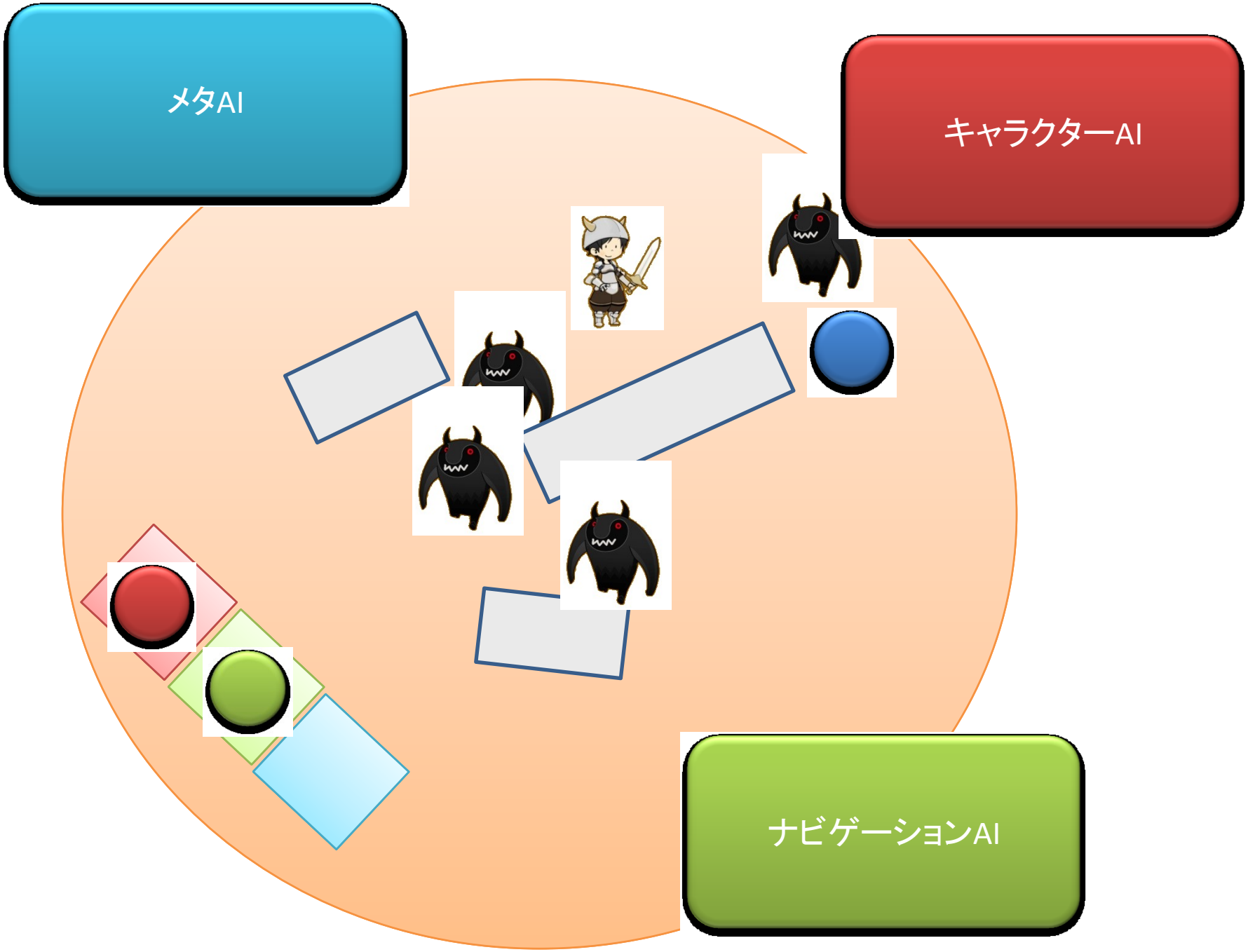
メタAIがゲームとプレイヤーの状況、性質を認識しながら、動的にゲームを変化させる。

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。

第三部

まとめ


メタAI
キャラクターAI
ナビゲーションAI

現代のゲームAI

- メタAI、キャラクターAI、ナビゲーションAIの三者が協調して、ゲーム全体を作り上げる。
- 動的にゲームを制御することで、固定されたデータが少なくし、ユーザーに応じて柔軟にゲームを変形させる。
- ゲームコンテンツが生成されるので、コンパクト（ミニマル）なシステムで大きなコンテンツを作ることができる。

デジタルゲームにおける最新の人工知能システム

三宅陽一郎

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

Twitter: @miyayou

感想・ご質問は メイルtwitter やfacebookまで

ご清聴ありがとうございました。

2014.1.10 @東京工芸大学　中野キャンパス