

ゲームAI連続セミナー

第5回

「NERO における学習と進化」

三宅 陽一郎

y_miyake@fromsoftware.co.jp

2007/10/27

November 2007

第5回セミナーの構成

はじめに

ゲームAI連続セミナーのご紹介

Enjoy AI !

第1部 ゲームAI 技術解説 (100分) (講師:三宅)

第1章 C4アーキテクチャー (20分)

第2章 遺伝的アルゴリズム (30分)

第3章 ニューラルネットワーク (30分)

第4章 発展 (10分)

各章の終わりに質疑応答を行います。

(休憩 20分)

第2部 ディスカッション (2時間) (アイデア:IGDAゲームAI運営委員)

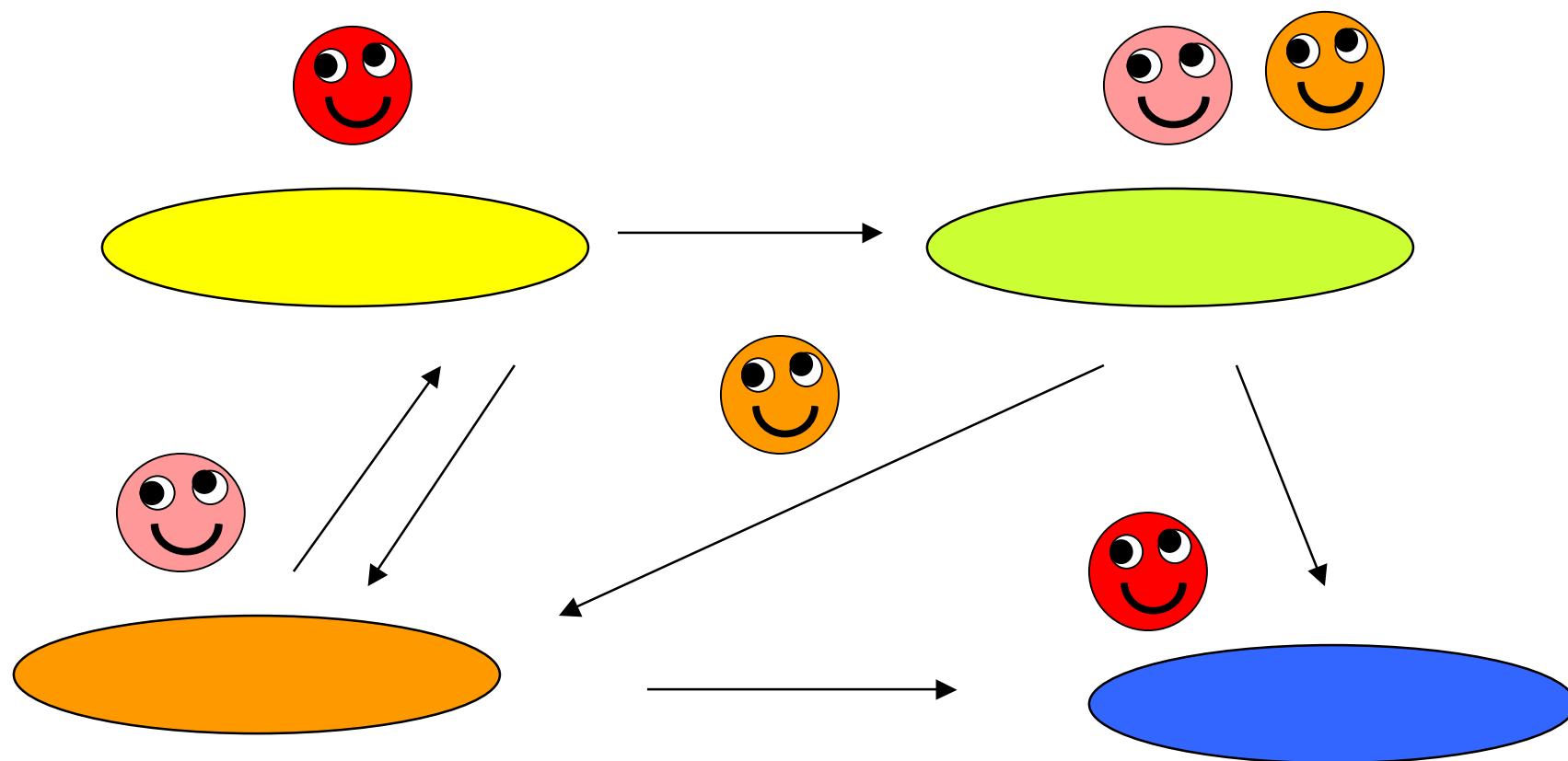
(総合司会 + コーディネート:長久、グループ司会者)

(I) グループ討論 (80分)

(I) まとめ (30分)

はじめに

ゲームAI連続セミナー全体のご紹介



IGDA日本

International Game Developers Association (国際ゲーム開発者協会) Japan

IGDAは全世界のゲーム開発者を対象とした国際NPOです。

igda.jp へアクセス!

igdaとは

IGDA(International Game Developer Association: アイジーディーエー; 国際ゲーム開発者協会)は、ゲーム開発者(個人)を対象とした国際NPOです。開発者の世界的コミュニティを形成し、個々の開発者が成長できるように協力を進めていくことを目的としています。全世界60ヶ国、都市を中心に支部が存在し、全世界12,000人のメンバーを抱え、毎年のように世界各地で開催されるゲーム関連イベントの運営等に携わっています。

代表的なイベントに、毎年春に米国で開催されるGDC(Game Developers Conference; ジョーダンシー; ゲーム開発者会議)があります。また、IGDAのメンバーの投票により決定される、前年の優れたゲームを表彰「ゲームディベロッパーズ・チョイス・アワード」はよく知られています。GDC 2007で行われた表彰では、日本のタイトルとして「ゼルダの伝説 トワイライトプリンセス」(任天堂)の日本版が、(任天堂)がゲームデザイン部門(大賞)「カブコン」がキャラクターデザイン部門で賞を獲っています。さらに、生涯を通じてゲームの発展に寄与した人物に贈られる、生涯功労賞が、宮本茂氏(任天堂)に贈られています。日本人の受賞者としては3人目となります。

IGDAのサイト: www.igda.org
GDCのサイト: www.gdc.com/japan/gdc.com

igda international game developers association Japan chapter

IGDA 日本とは

IGDA (International Game Developer Association) Japan chapter: アイジーディーエー日本・国際ゲーム開発者協会日本、代表(新澤士)は、IGDAの日本支部です。2002年4月に設立されました。日本でのゲーム開発者のコミュニティを開発者、ゲーム開発者向けの情報が流通し、ゲームとゲーム産業を発展させることを目的に活動を行っています。

CEDECや、東京ゲームショー、アジアオンラインゲームカンファレンス(AOGC: BSA主催)、日本デジタルゲーム学会(JGDG: Japan)等への運営協力など、ゲーム開発者を支援する働きかけを行っています。IGDA日本のサブ支部として、IGDA関西があります。

IGDA 日本のサイト: www.igda.jp

2007年 IGDA 日本主催および関連イベント

- ・アジアオンラインゲームカンファレンス 2007 東京 2007/2/22 ~ 23(BSA主催、後援)
- ・Game Developers Conference 07 2007/3/5 ~ 9(SMP主催、日本版サイト運営協力)
- ・Game Developers Conference 07 レポート 2007/3/24
- ・IGDA 横浜ゲーム開発者セミナー 第32回 CAPSULEゲームクリエイターの視点 (IGDA関西) 2007/7/11
- ・シンポジウム『映像制作の法と経緯』(BSA主催、後援) 2007/7/21
- ・ゲームAI開発セミナー: ゲームAIを脱ミステリー
- 第1回 7/16(東京)におけるNECの動的映像制作法 2006/12/16
- 第2回 7/16(東京)におけるゲーム制作のリアルタイムレンダリング 2007/12/16
- 第3回 7/16(東京)におけるゲーム制作のリアルタイムレンダリング 2007/12/16
- 第4回 7/16(東京)におけるFPGA(高度型汎用計算機) 2007/6/30
- 第5回 第6回は10月以降に開催予定。
- ・ゲームCG開発セミナー: 新世代CG技術の事例報告
- 第1回 7/16(東京)におけるCGムービー制作法 2007/4/28
- 第2回 7/16(東京)におけるCGムービー制作法 2007/4/28
- 第3回 7/16(東京)におけるCGムービー制作法 2007/4/28

Local Sponsors

ゲーム開発者の、
ゲーム開発者による、
ゲーム開発者のための国際NPO

ゲーム開発者

= 開発者 + メディア + ... + 先生 + 学生

ゲーム開発にいろいろな形で携わる全ての人

ボランティアとして開発者同士が、
相互にインタラクションする場

CEDEC 2007
パンフレットを見よう!

IGDA日本 ゲームAI連続セミナー(全6回)

技術的地平を「なるべく遠くまで見せる」そして開発へ...

より広い技術的フィールドで
開発者が楽しくゲームのAIを開発できるようにしよう！



第5回 N.E.R.O 進化と学習



第4回 Halo2 階層型有限状態機械

第3回 ChromeHounds マルチエージェント



第2回 F.E.A.R プランニング

第1回 Killzone 知識表現(世界表現)



ゲームAIセミナー群

第1回	Killzone 知識表現 (世界表現)	4/12	殺陣の森	100
第2回	F.E.A.R プランニング	5/13	殺陣の森	100
第3回	ChromeHounds マルチエージェント	6/10	殺陣の森	100
第4回	Halo2 有限状態機械	7/18	殺陣の森	100
第5回	NERO 学習と進化	8/15	殺陣の森	100
第6回	ゲームAIの基礎	9/12	殺陣の森	100
第7回	ゲームAIの応用	10/10	殺陣の森	100
第8回	ゲームAIの発展	11/7	殺陣の森	100
第9回	ゲームAIの未来	12/5	殺陣の森	100
第10回	ゲームAIの総まとめ	1/2	殺陣の森	100
第11回	ゲームAIの総まとめ	1/9	殺陣の森	100
第12回	ゲームAIの総まとめ	1/16	殺陣の森	100
第13回	ゲームAIの総まとめ	1/23	殺陣の森	100
第14回	ゲームAIの総まとめ	1/30	殺陣の森	100
第15回	ゲームAIの総まとめ	2/6	殺陣の森	100
第16回	ゲームAIの総まとめ	2/13	殺陣の森	100
第17回	ゲームAIの総まとめ	2/20	殺陣の森	100
第18回	ゲームAIの総まとめ	2/27	殺陣の森	100
第19回	ゲームAIの総まとめ	3/6	殺陣の森	100
第20回	ゲームAIの総まとめ	3/13	殺陣の森	100
第21回	ゲームAIの総まとめ	3/20	殺陣の森	100
第22回	ゲームAIの総まとめ	3/27	殺陣の森	100
第23回	ゲームAIの総まとめ	4/3	殺陣の森	100
第24回	ゲームAIの総まとめ	4/10	殺陣の森	100
第25回	ゲームAIの総まとめ	4/17	殺陣の森	100
第26回	ゲームAIの総まとめ	4/24	殺陣の森	100
第27回	ゲームAIの総まとめ	5/1	殺陣の森	100
第28回	ゲームAIの総まとめ	5/8	殺陣の森	100
第29回	ゲームAIの総まとめ	5/15	殺陣の森	100
第30回	ゲームAIの総まとめ	5/22	殺陣の森	100
第31回	ゲームAIの総まとめ	5/29	殺陣の森	100
第32回	ゲームAIの総まとめ	6/5	殺陣の森	100
第33回	ゲームAIの総まとめ	6/12	殺陣の森	100
第34回	ゲームAIの総まとめ	6/19	殺陣の森	100
第35回	ゲームAIの総まとめ	6/26	殺陣の森	100
第36回	ゲームAIの総まとめ	7/3	殺陣の森	100
第37回	ゲームAIの総まとめ	7/10	殺陣の森	100
第38回	ゲームAIの総まとめ	7/17	殺陣の森	100
第39回	ゲームAIの総まとめ	7/24	殺陣の森	100
第40回	ゲームAIの総まとめ	7/31	殺陣の森	100
第41回	ゲームAIの総まとめ	8/7	殺陣の森	100
第42回	ゲームAIの総まとめ	8/14	殺陣の森	100
第43回	ゲームAIの総まとめ	8/21	殺陣の森	100
第44回	ゲームAIの総まとめ	8/28	殺陣の森	100
第45回	ゲームAIの総まとめ	9/4	殺陣の森	100
第46回	ゲームAIの総まとめ	9/11	殺陣の森	100
第47回	ゲームAIの総まとめ	9/18	殺陣の森	100
第48回	ゲームAIの総まとめ	9/25	殺陣の森	100
第49回	ゲームAIの総まとめ	10/2	殺陣の森	100
第50回	ゲームAIの総まとめ	10/9	殺陣の森	100
第51回	ゲームAIの総まとめ	10/16	殺陣の森	100
第52回	ゲームAIの総まとめ	10/23	殺陣の森	100
第53回	ゲームAIの総まとめ	10/30	殺陣の森	100
第54回	ゲームAIの総まとめ	11/6	殺陣の森	100
第55回	ゲームAIの総まとめ	11/13	殺陣の森	100
第56回	ゲームAIの総まとめ	11/20	殺陣の森	100
第57回	ゲームAIの総まとめ	11/27	殺陣の森	100
第58回	ゲームAIの総まとめ	12/4	殺陣の森	100
第59回	ゲームAIの総まとめ	12/11	殺陣の森	100
第60回	ゲームAIの総まとめ	12/18	殺陣の森	100
第61回	ゲームAIの総まとめ	12/25	殺陣の森	100
第62回	ゲームAIの総まとめ	1/1	殺陣の森	100
第63回	ゲームAIの総まとめ	1/8	殺陣の森	100
第64回	ゲームAIの総まとめ	1/15	殺陣の森	100
第65回	ゲームAIの総まとめ	1/22	殺陣の森	100
第66回	ゲームAIの総まとめ	1/29	殺陣の森	100
第67回	ゲームAIの総まとめ	2/5	殺陣の森	100
第68回	ゲームAIの総まとめ	2/12	殺陣の森	100
第69回	ゲームAIの総まとめ	2/19	殺陣の森	100
第70回	ゲームAIの総まとめ	2/26	殺陣の森	100
第71回	ゲームAIの総まとめ	3/5	殺陣の森	100
第72回	ゲームAIの総まとめ	3/12	殺陣の森	100
第73回	ゲームAIの総まとめ	3/19	殺陣の森	100
第74回	ゲームAIの総まとめ	3/26	殺陣の森	100
第75回	ゲームAIの総まとめ	4/2	殺陣の森	100
第76回	ゲームAIの総まとめ	4/9	殺陣の森	100
第77回	ゲームAIの総まとめ	4/16	殺陣の森	100
第78回	ゲームAIの総まとめ	4/23	殺陣の森	100
第79回	ゲームAIの総まとめ	4/30	殺陣の森	100
第80回	ゲームAIの総まとめ	5/7	殺陣の森	100
第81回	ゲームAIの総まとめ	5/14	殺陣の森	100
第82回	ゲームAIの総まとめ	5/21	殺陣の森	100
第83回	ゲームAIの総まとめ	5/28	殺陣の森	100
第84回	ゲームAIの総まとめ	6/4	殺陣の森	100
第85回	ゲームAIの総まとめ	6/11	殺陣の森	100
第86回	ゲームAIの総まとめ	6/18	殺陣の森	100
第87回	ゲームAIの総まとめ	6/25	殺陣の森	100
第88回	ゲームAIの総まとめ	7/2	殺陣の森	100
第89回	ゲームAIの総まとめ	7/9	殺陣の森	100
第90回	ゲームAIの総まとめ	7/16	殺陣の森	100
第91回	ゲームAIの総まとめ	7/23	殺陣の森	100
第92回	ゲームAIの総まとめ	7/30	殺陣の森	100
第93回	ゲームAIの総まとめ	8/6	殺陣の森	100
第94回	ゲームAIの総まとめ	8/13	殺陣の森	100
第95回	ゲームAIの総まとめ	8/20	殺陣の森	100
第96回	ゲームAIの総まとめ	8/27	殺陣の森	100
第97回	ゲームAIの総まとめ	9/3	殺陣の森	100
第98回	ゲームAIの総まとめ	9/10	殺陣の森	100
第99回	ゲームAIの総まとめ	9/17	殺陣の森	100
第100回	ゲームAIの総まとめ	9/24	殺陣の森	100

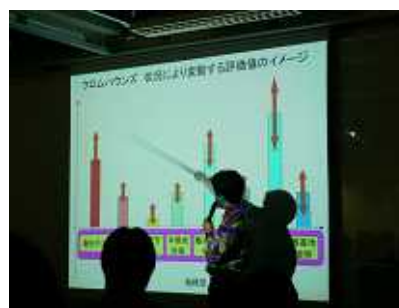
第5回 NERO 学習と進化

第4回 Halo2 有限状態機械

第3回 ChromeHounds マルチエージェント

第2回 F.E.A.R プランニング

第1回 Killzone 知識表現 (世界表現)



社内セミナー
(週1回/～90回)

IGDAゲームAI連続セミナー
(2ヶ月1回/全6回)

FROM SOFTWARE

igda international game developers association Japan chapter

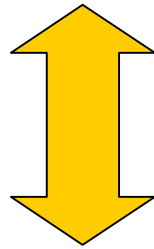
日本最大のゲーム開発者カンファレンス
CEDEC 2007

目的

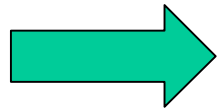
- (1) ゲームAIの技術について紹介、発表できる場を作る。
- (2) ゲームAIに関して討論を行う
オープンな相互インタラクションのできる場を用意する。
- (3) 参加者の意見を取り入れて、
ゲーム業界で本当に必要なセミナーの形を実現する。
(運営やスピーカーなど常時募集する)

ゲームAIの特徴

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



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



ゲームとゲームAI は、
お互いを進化させる力を持つ

実際のゲーム開発へ活用する



これからのゲーム開発におけるゲームAIの
可能性を知る

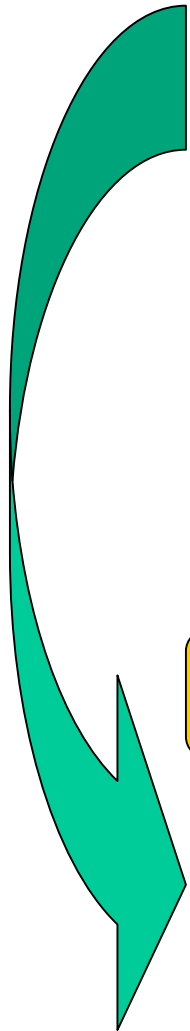


IGDA
ゲームAI連続セミナー

開発者が一同に介して、
ゲームAIについて学び、討論する場

よい循環を形成する

日本におけるゲームAIのセミナー・カンファレンスの場



より広い技術的フィールドで
開発者が楽しくゲームのAIを開発できるようにしよう！

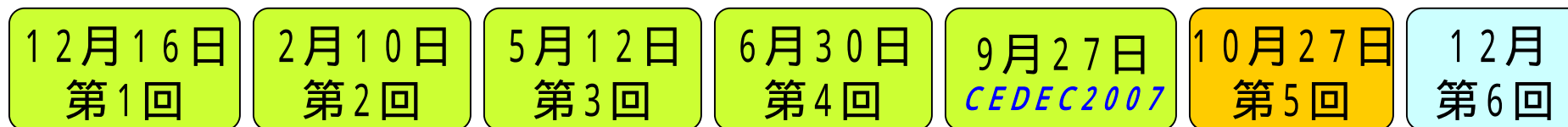
2006年9月

セミナーの現在のプラン

2007年12月



2ヶ月に1回実施



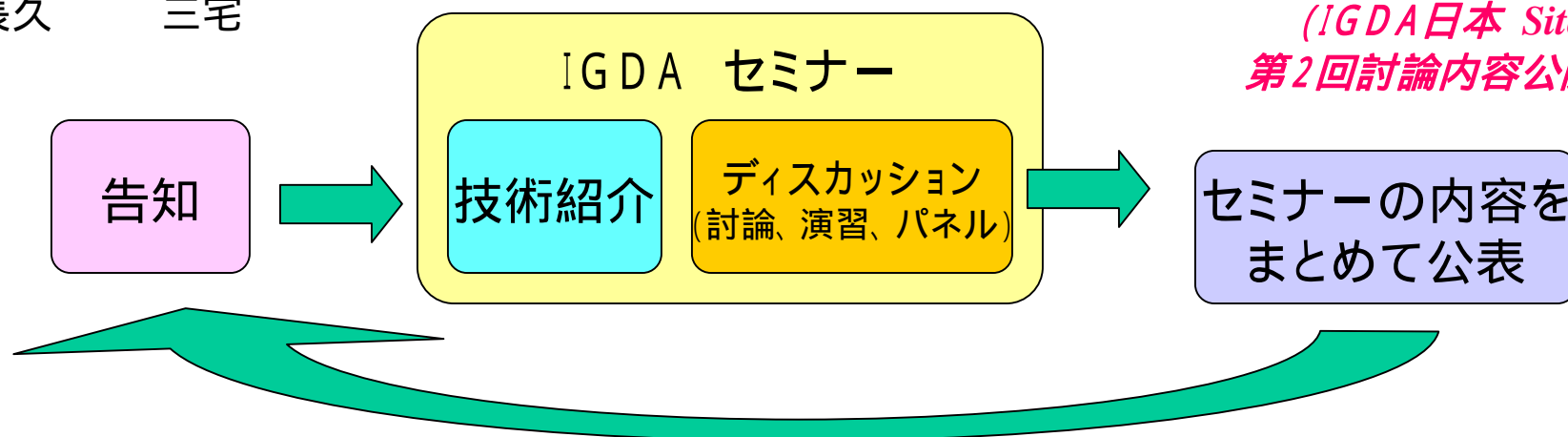
コーディネーター
長久



講師
三宅

全6回(継続を目指す)
会場は大学、企業のホールなど

第1、2回資料公開
(IGDA日本 Site)
第2回討論内容公開



目標へ進みつつ、参加者の意図を汲みながら
徐々に改善して行く

第4回からの反省と改善点

(WEBアンケート協力ありがとうございます！
今回もよろしくお願いします！)

(1) 講演

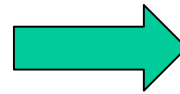
先に具体例を出して
その後に説明してほしい



具体的なデモを提示して、
次に解説をする

(2) グループワーク

うまく議論が進まない



個人ワークがベース。
その上で議論

(3) 運営(継続のために省エネ運営)

IGDA日本に協力してくださる方(ボランティア)を募集！

第5回のチャレンジ

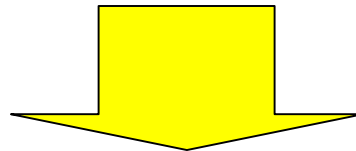
(1) 日本初の企業スペースによる開催

海外では、EAなど、いろいろな企業が会場を提供しているらしい

(2) 企業スペースによる懇親会

いろいろな人と話をしたい

飲み会でなく懇親をしたい



ゲームを囲んでゲームを語ろう！

第6回セミナーへ

CEDEC 2007 からのアンケート

復習だけなら物足りない



「まとめ」ではなく、
新しい技術トピック

第5回セミナーの構成

はじめに

ゲームAI連続セミナーのご紹介

Enjoy AI!

第1部 ゲームAI 技術解説 (100分) (講師: 三宅)

第1章 概論 (10分)

第2章 C4アーキテクチャー (20分)

第3章 遺伝的アルゴリズム (30分)

第4章 ニューラルネットワーク (30分)

第5章 発展 (10分)

各章の終わりに質疑応答を行います。

(休憩 20分)

第2部 ディスカッション (2時間) (アイデア: IGDA ゲームAI運営委員)

(総合司会 + コーディネート: 長久、グループ司会者)

(I) グループ討論 (80分)

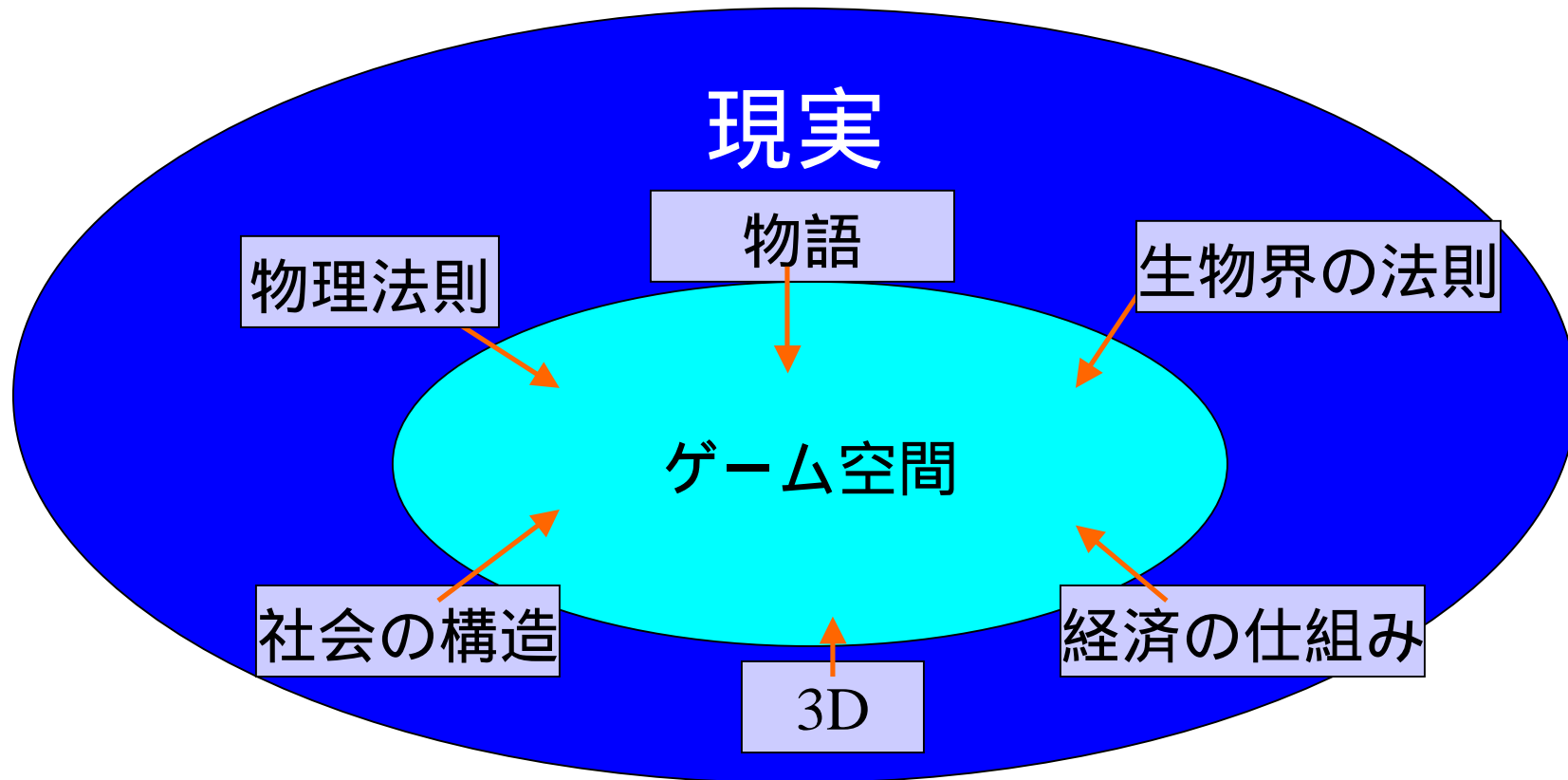
(I) まとめ (30分)

第1章 概論

「生物界の法則の力をゲーム空間へ導く」

第5回のテーマ

ゲーム世界は現実を模倣する。政治、経済、3D空間、そして、物理法則によってゲーム世界を豊かにしようとする



物理法則や経済の仕組みは既にゲームへ導入されているが、
「生物界の法則」は、まだ十分に導入されていない

(テーマ) 「生物界の法則」をゲーム世界に導入すること

学習・進化・遺伝

生物界の法則

生物の生殖・遺伝原理

遺伝的アルゴリズム

生物の学習原理

個体を対象とする

[原始的な学習原理] ニューラルネットワーク

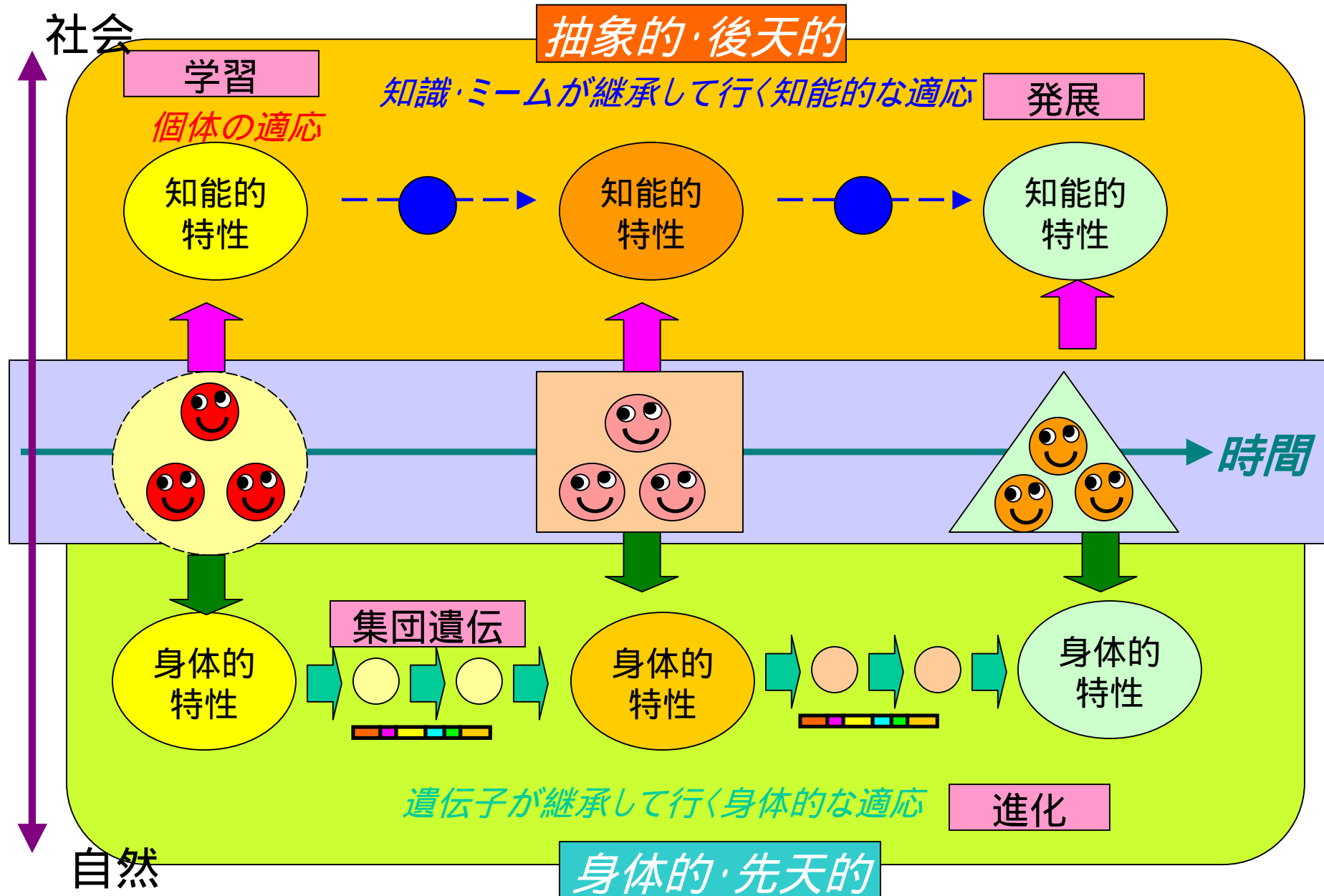
[高度な学習原理] 認識過程・思考・記憶における学習

生物の進化

個体集団を対象とする

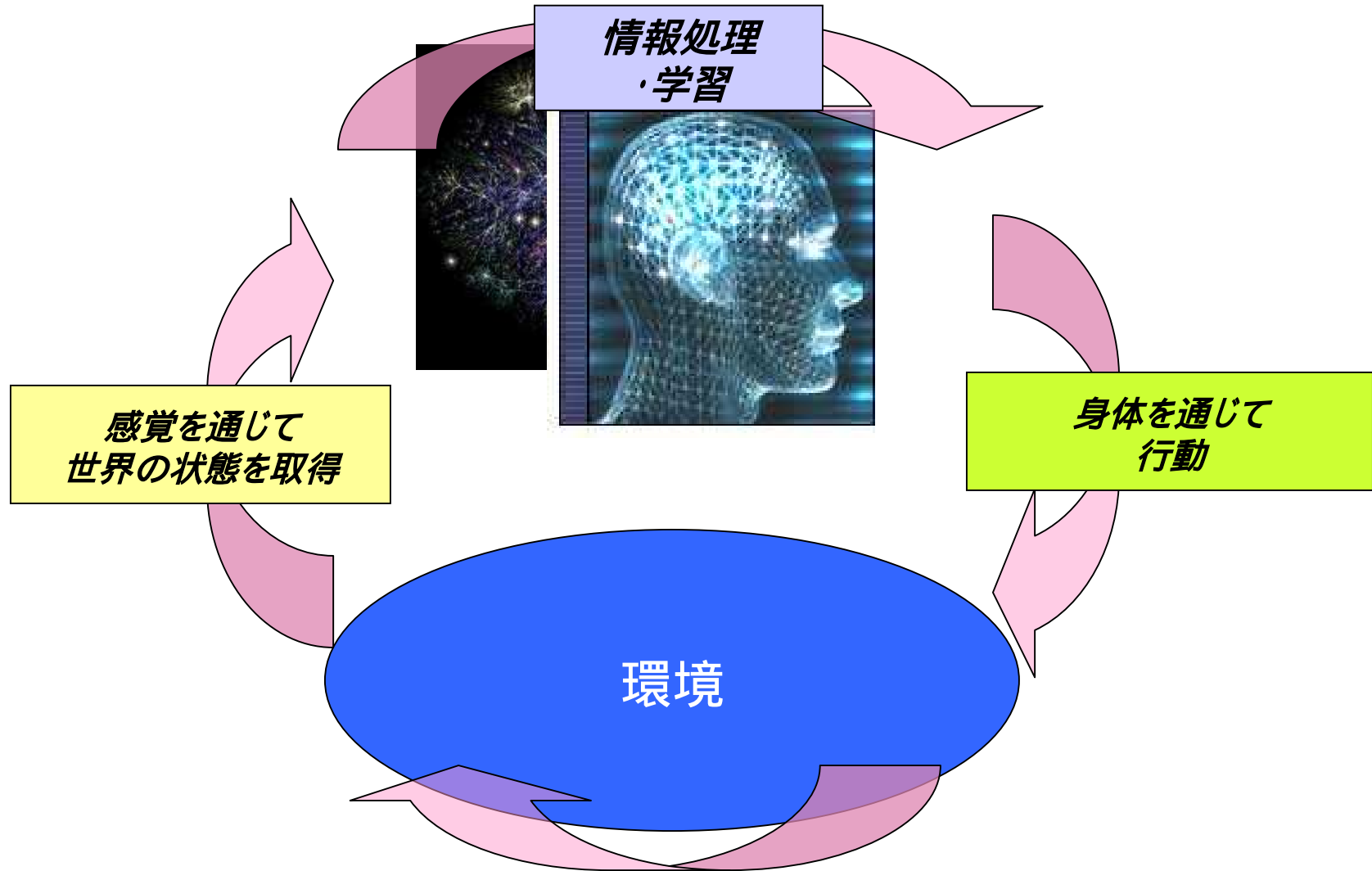
ニューラルネットワーク × 遺伝的アルゴリズム

学習・遺伝・進化



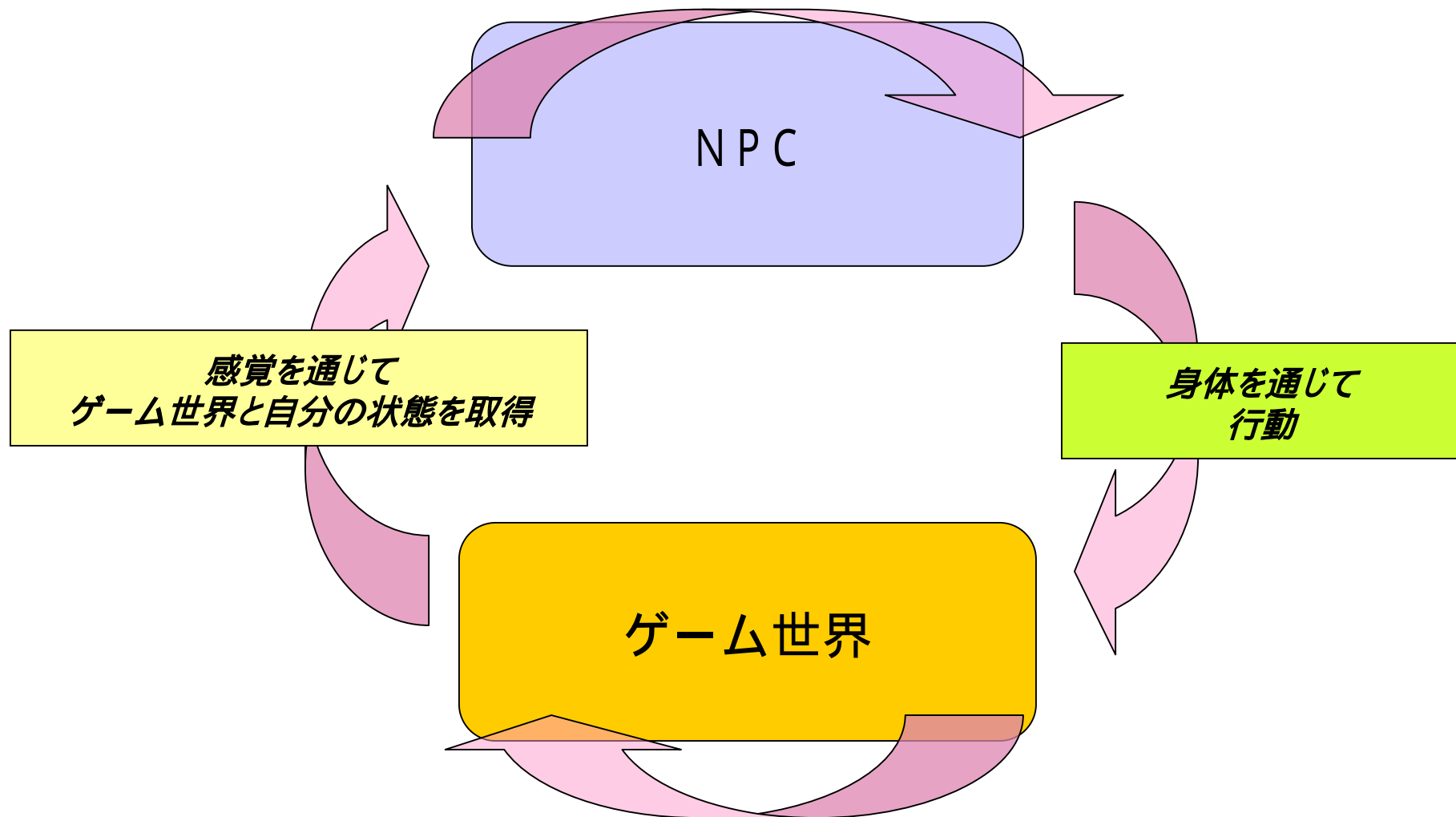
個体の学習

人間の身体・脳と学習



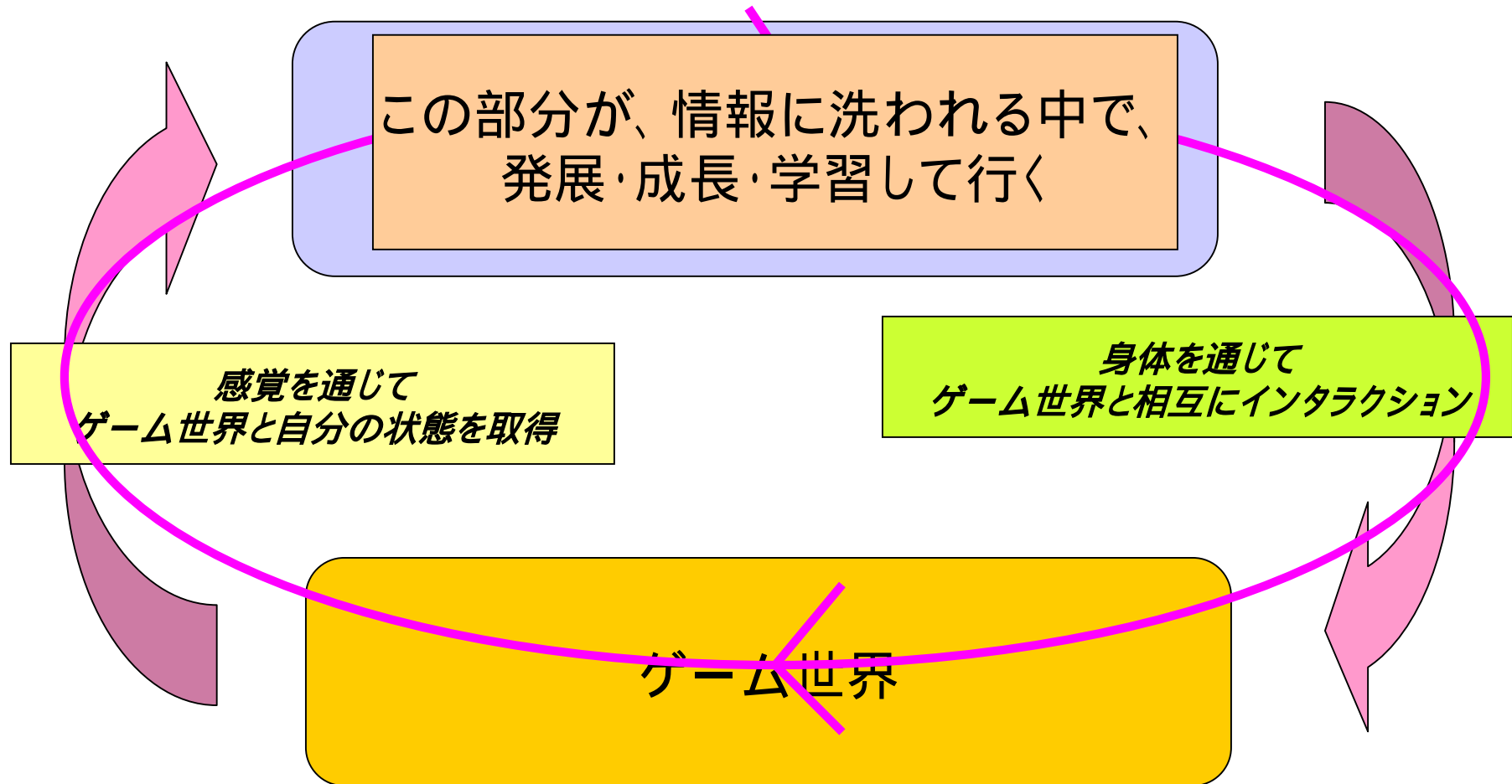
人間の脳(100億のニューロンの結合)は、産まれたときから、周囲の環境から情報を取得し、行動し、学習する無数のサイクルの中で発展する。

ゲームAI



人間の脳(100億のニューロンの結合)は、産まれたときから、
周囲の環境から情報を取得し、行動し、学習する無数のサイクルの中で発展する。

エージェントと学習



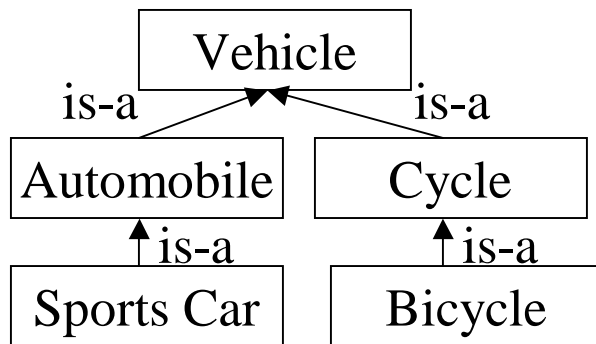
人工知能を作る二つのアプローチ

知能

シンボリズム (記号主義)

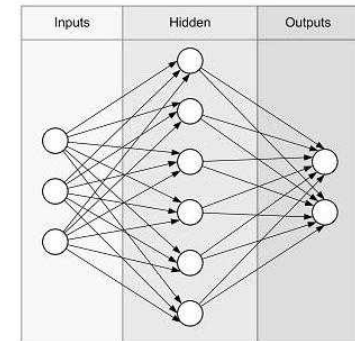
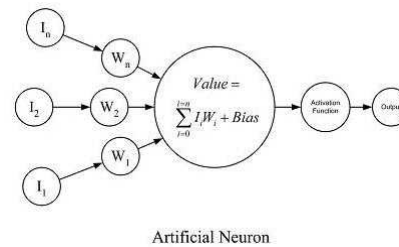
シンボル(数値、文字列)による知識表現
によって人工知能を組み立てる方法

If (x < 100) Jump else Walk.



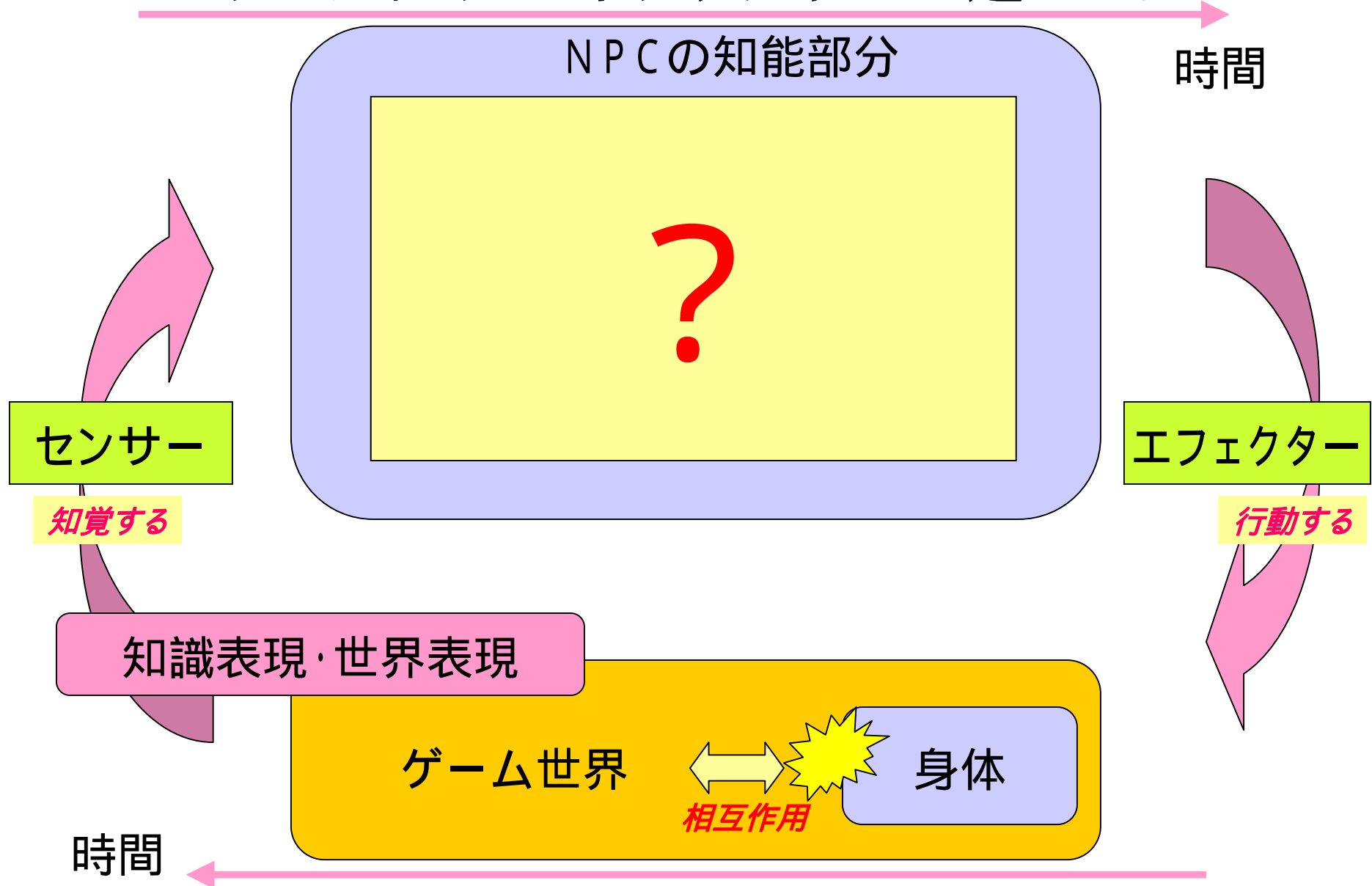
コネクショニズム (結合主義)

神経素子(ニューロン)の
結合回路(ニューラルネットワーク)
によって人工知能を組み立てる方法



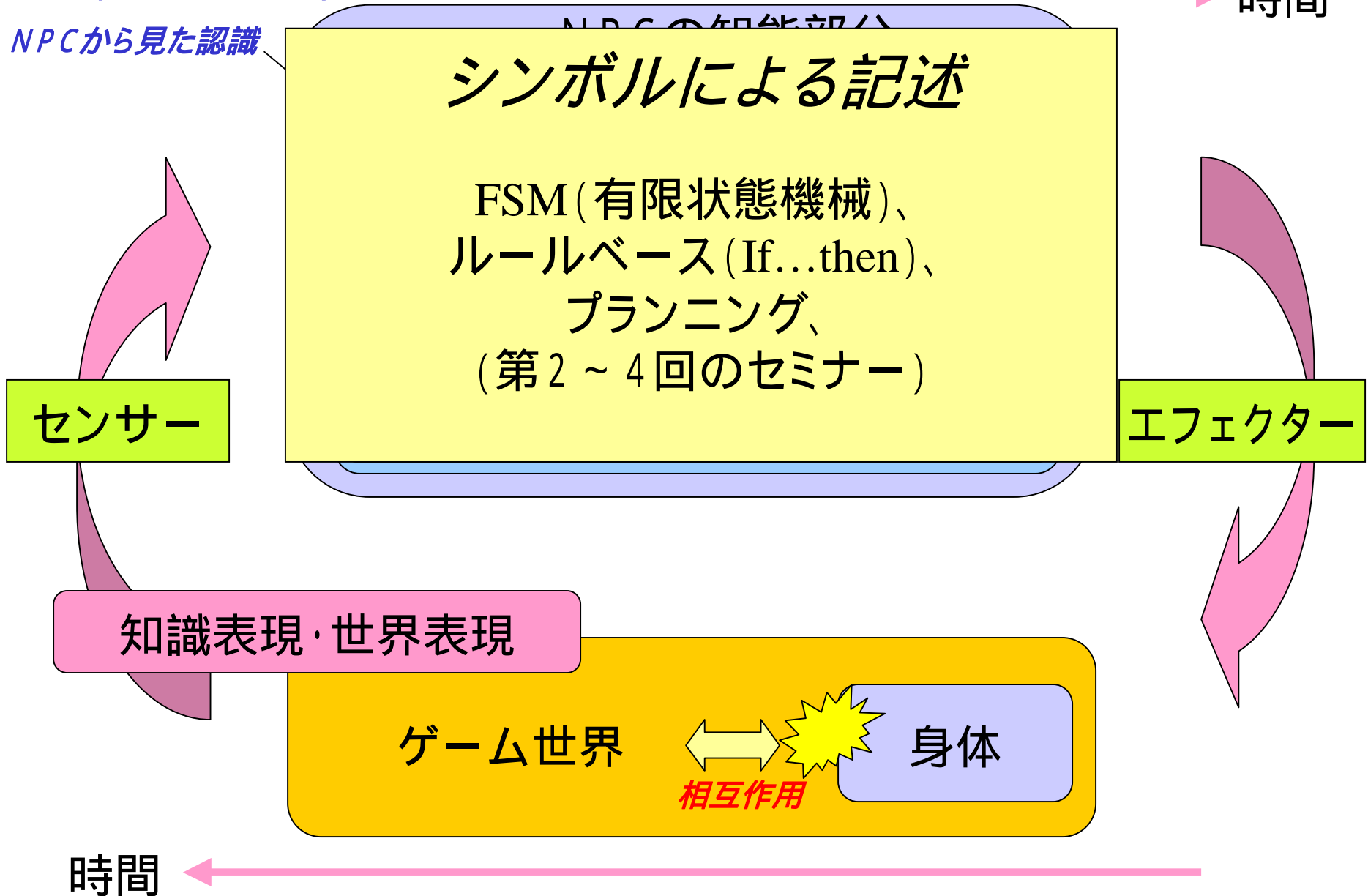
二つの方法論として、両者は相容れることなく、
徹底的に別の道をたどって人工知能を形造ろうとする

記号主義と結合主義による エージェント・アーキテクチャーの違いとは？



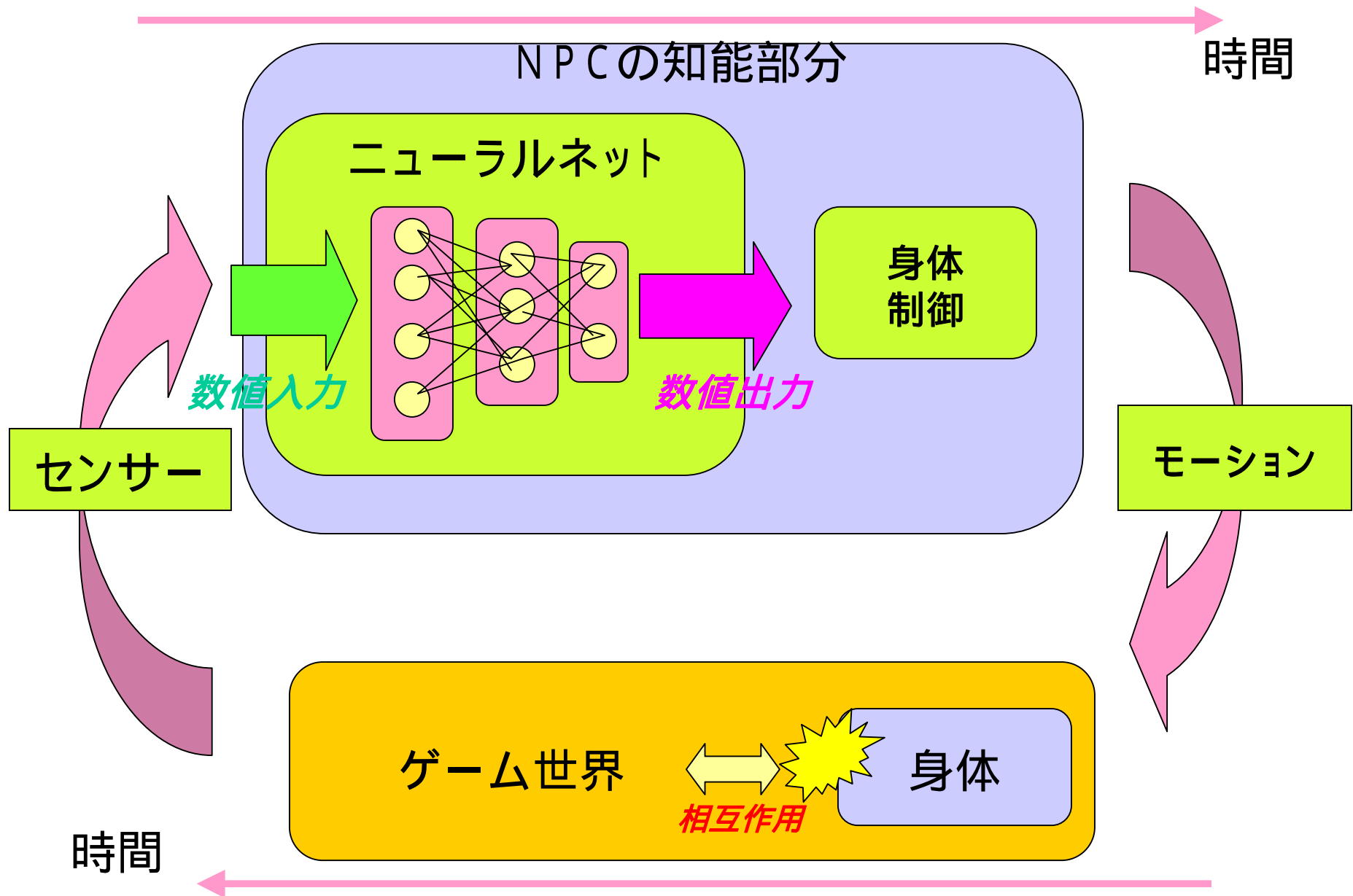
記号主義によるエージェント・アーキテクチャ

(シンボリズム)



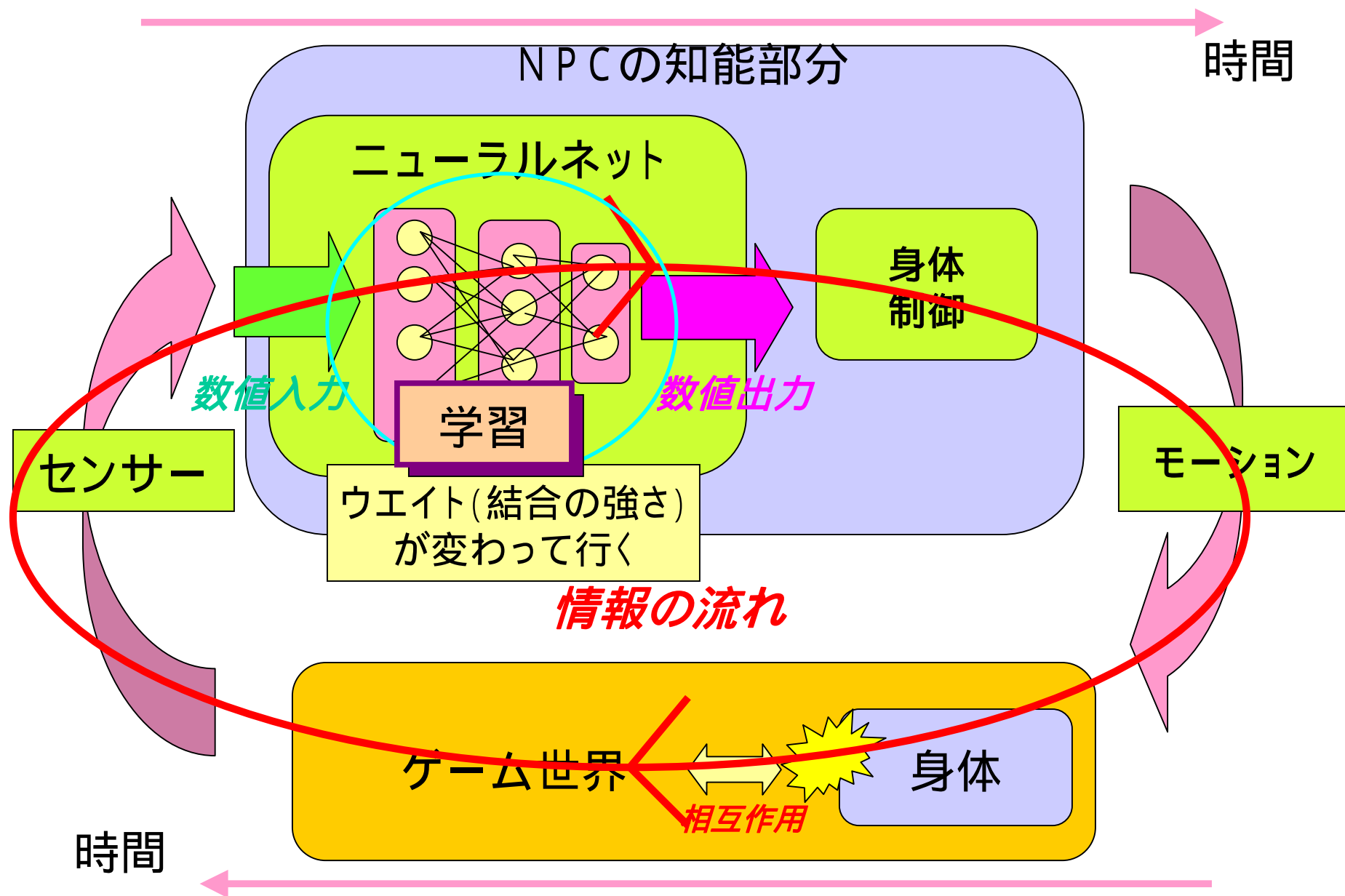
結合主義によるエージェント・アーキテクチャー

(コネクショニズム)



結合主義によるエージェント・アーキテクチャー

(コネクショニズム)



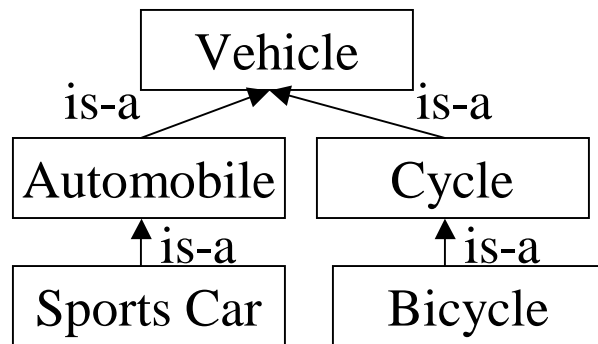
人工知能を作る二つのアプローチ

知能

シンボリズム (記号主義)

シンボル(数列、文字列)による知識表現
によって人工知能を組み立てる方法

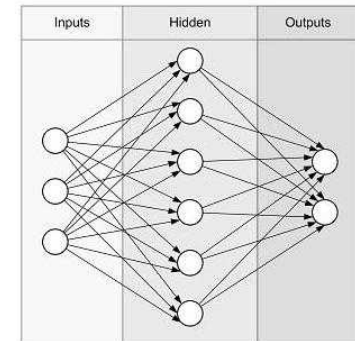
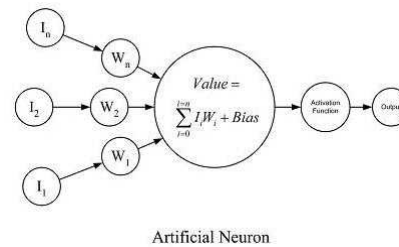
If ($x < 100$) Jump else Walk.



C4アーキテクチャー

コネクショニズム (結合主義)

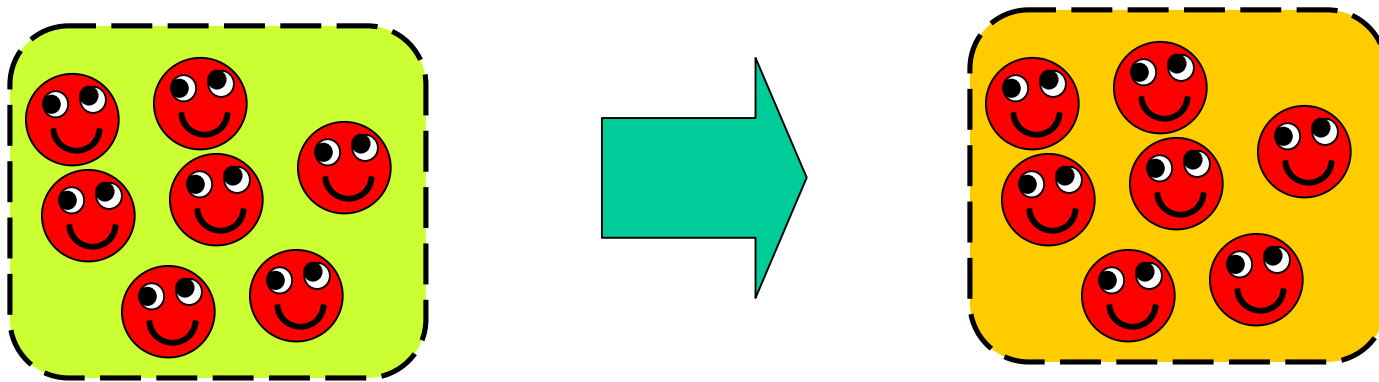
神経素子(ニューロン)の
結合回路(ニューラルネットワーク)
によって人工知能を組み立てる方法



ニューラルネット

集合の進化

進化論的計算方法



遺伝的アルゴリズム

第5回セミナーの構成

はじめに

ゲームAI連続セミナーのご紹介

Enjoy AI!

第1部 ゲームAI 技術解説 (100分) (講師: 三宅)

第1章 概論 (10分)

第2章 C4アーキテクチャー (20分)

第3章 遺伝的アルゴリズム (30分)

第4章 ニューラルネットワーク (30分)

第5章 発展 (10分)

各章の終わりに質疑応答を行います。

(休憩 20分)

第2部 ディスカッション (2時間) (アイデア: IGDA ゲームAI運営委員)

(総合司会 + コーディネート: 長久、グループ司会者)

(I) グループ討論 (80分)

(I) まとめ (30分)

C4 アーキテクチャー

デジタル空間で生きる生物の知性のために提案されたアーキテクチャー



MIT Media Lab.

Synthetic Characters Group

論文： D. Isla, R. Burke, M. Downie, B. Blumberg (2001).,
“A Layered Brain Architecture for Synthetic Creatures”,
<http://characters.media.mit.edu/Papers/ijcai01.pdf>
(ゲームAIにおける最重要論文の1つ)

生き物の知性をエージェント・アーキテクチャーに写し取る

バーチャルな空間に生きる犬が実現され、後にF.E.A.R. のNPCのAIに応用される。

Demo Movie Trial-medium

Duncan は、どんなバーチャル牧羊犬
なのか見てみよう！

<http://web.media.mit.edu/~bruce/whatsnew.html>
(MIT Media Lab Bruce Blumberg's Web Page)

C4 アーキテクチャーの課題

仮想空間でリアルに生きる動物のためのアーキテクチャー

(応用) バーチャル牧羊犬“Duncan”



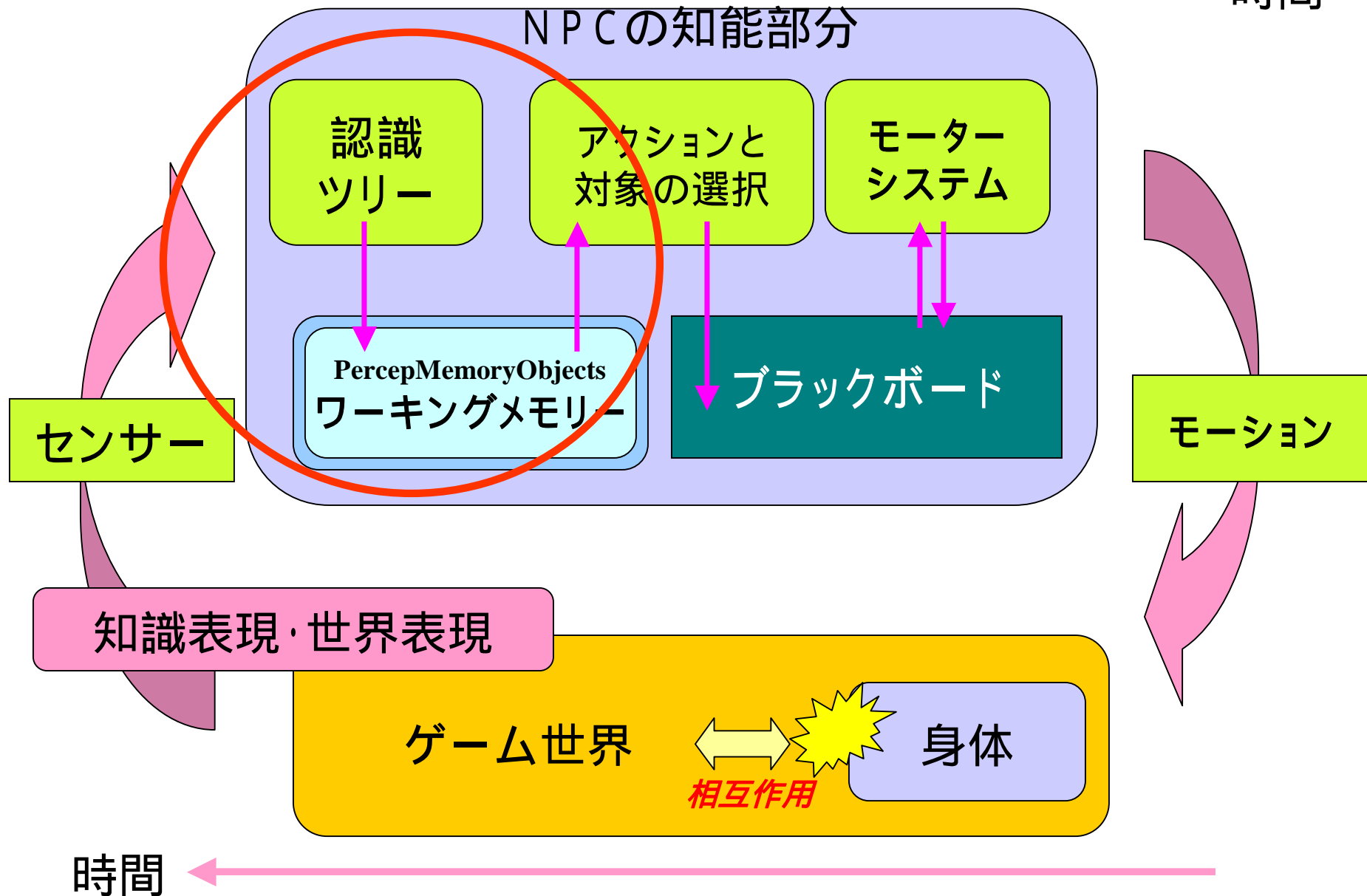
プレイヤーは羊飼いとなってボイスで指示したり、訓練(学習)したりする。

(注) C4 アーキテクチャーが大切なのであって、
この犬のデモは「何が出来るか」が大切であって、
デモの出来はたいして重要ではない。(研究と開発の違い)

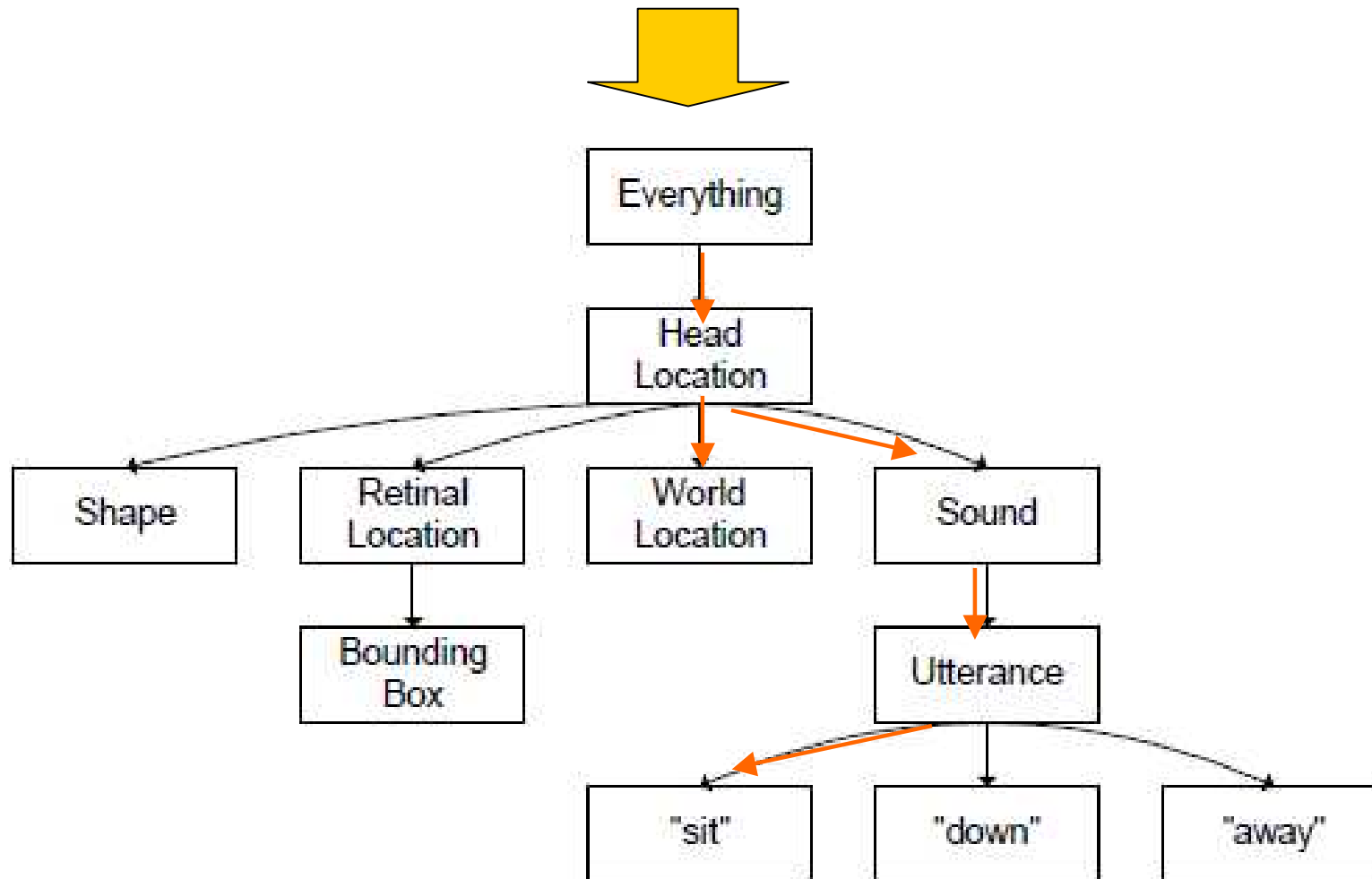
研究結果を調べるときは、その本質だけを見抜こう！

MIT メディアラボ C4 アーキテクチャー

時間 →



認識ツリーとは？

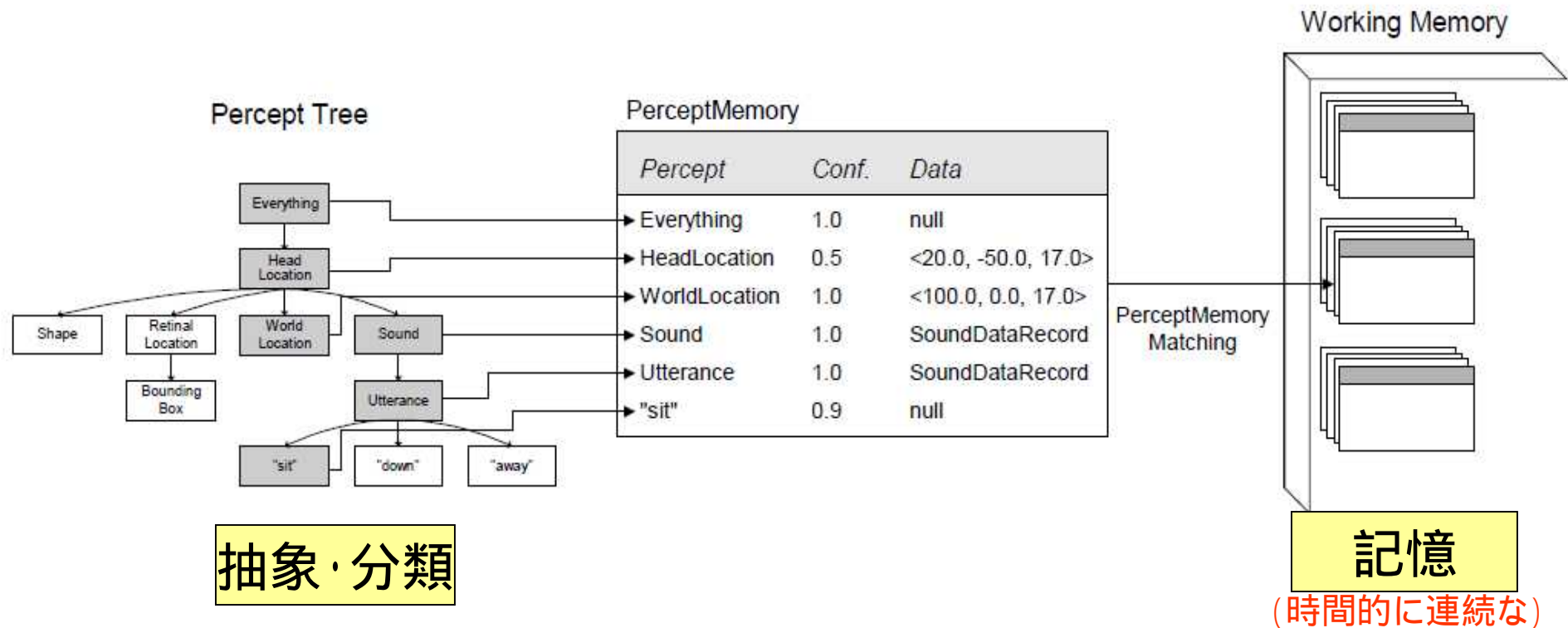


センサーから得た情報が、一体何の情報なのか、
分類(抽出)する条件分岐ツリー。

認識ツリーとワーキングメモリー

認識ツリー

PerceptMemoryObjects
ワーキングメモリー



認識ツリーで分類した情報を、PerceptMemoryObjectとしてワーキングメモリーへ蓄積して行く。

行動の形式

アクション・タプル

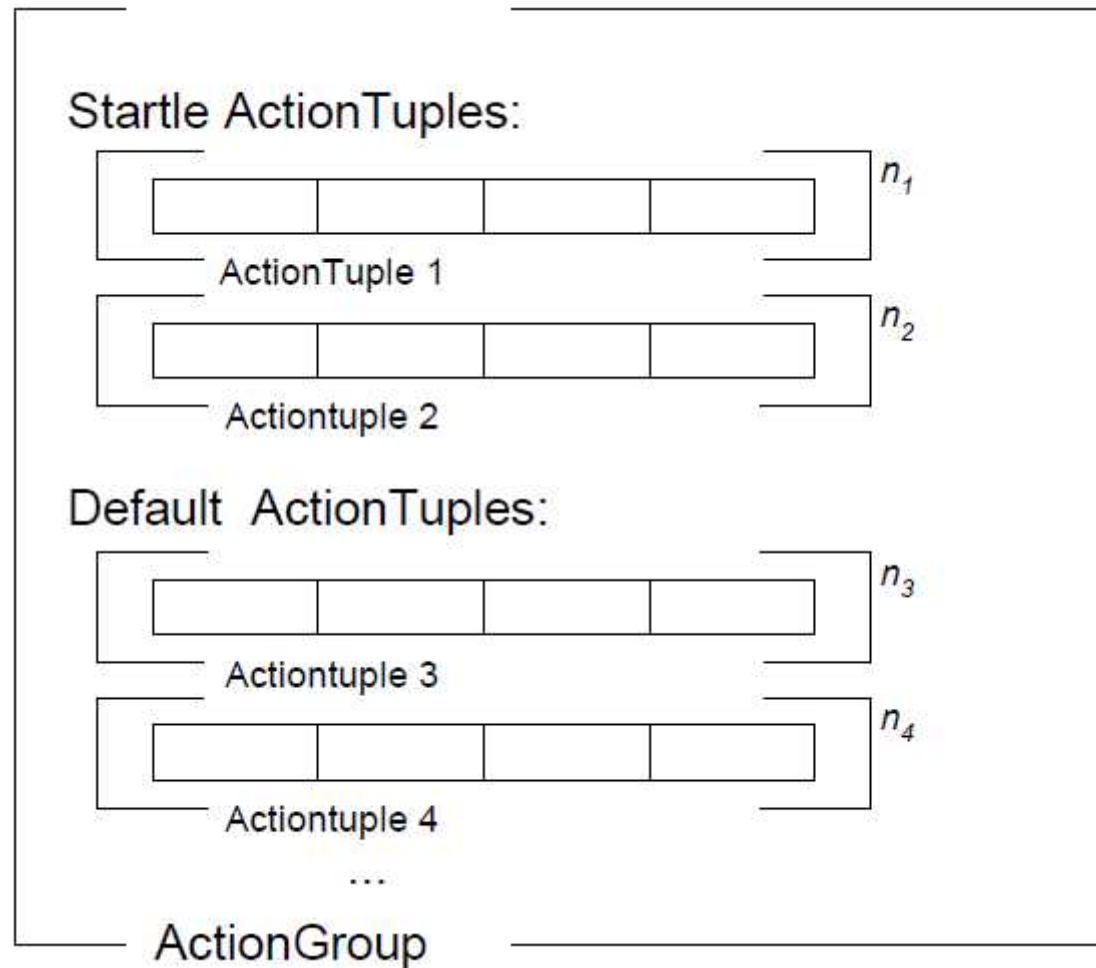


=	トリガー条件 (開始)	行動	オブジェクト情報	終了条件	重要度
---	----------------	----	----------	------	-----

(例)

“Away”と聞く	半時計 周りに回る	羊の形を認識する	15秒、或いは 群れを囲う	50
-----------	--------------	----------	------------------	----

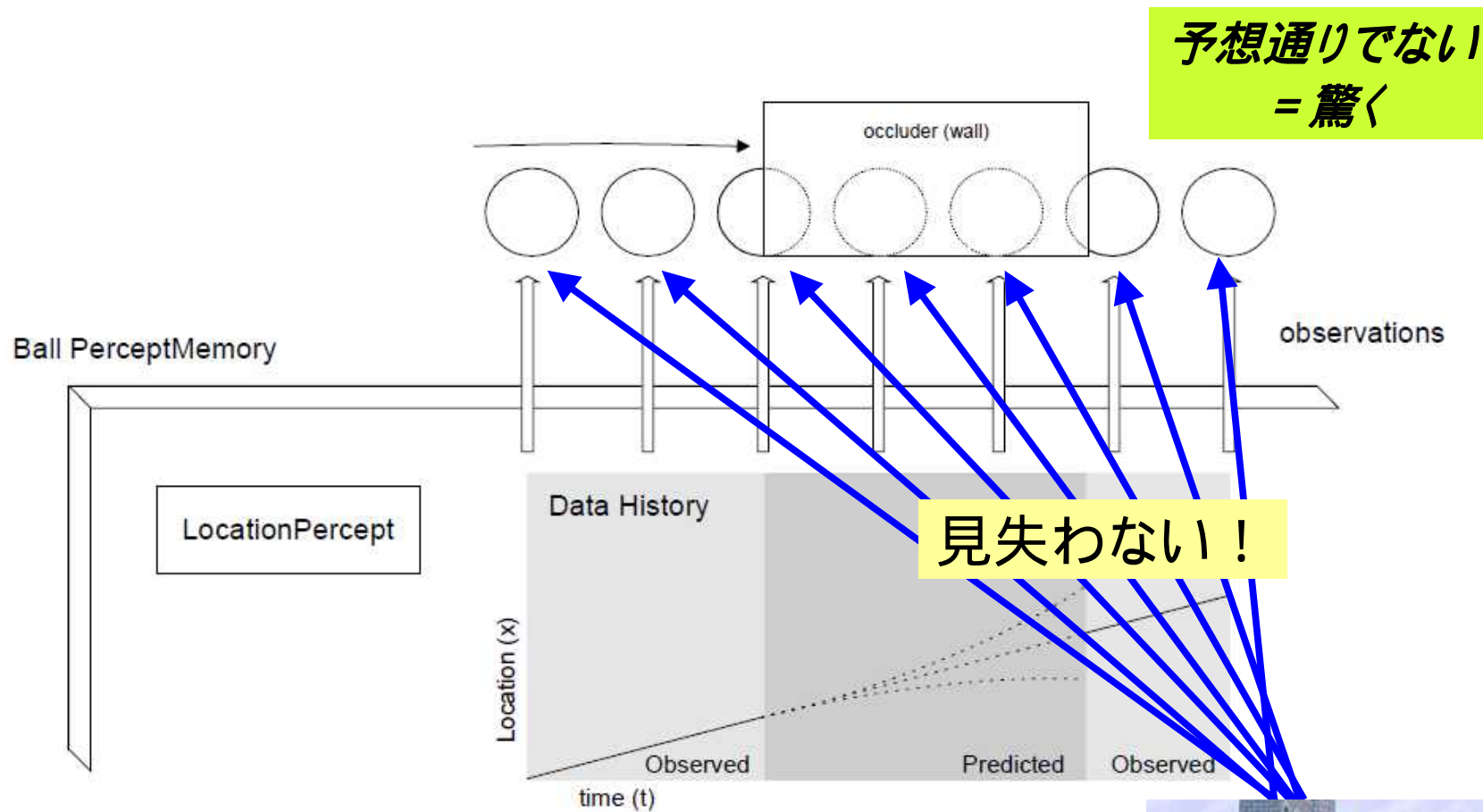
行動の選択



基本的に最も高い評価値を獲得した行動が選択される

$$\text{評価値} = (\text{トリガーコンテキストの出力する値}) \times (\text{重要度})$$

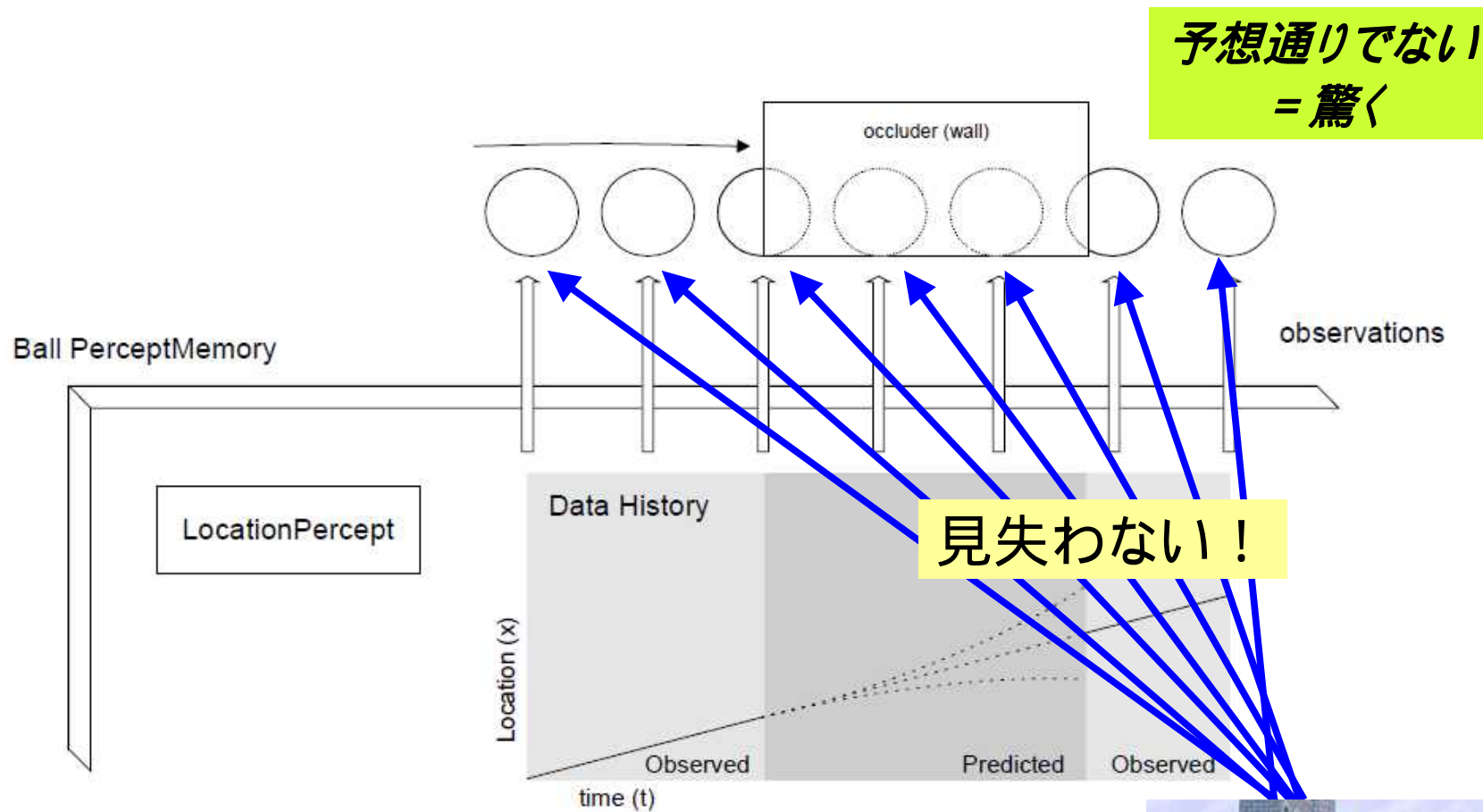
記憶によって可能になること = 予測と驚き



対象(ボール)が遮蔽物に隠れても、
履歴の記憶から動きを予測することが出来る。



記憶によって可能になること = 予測と驚き

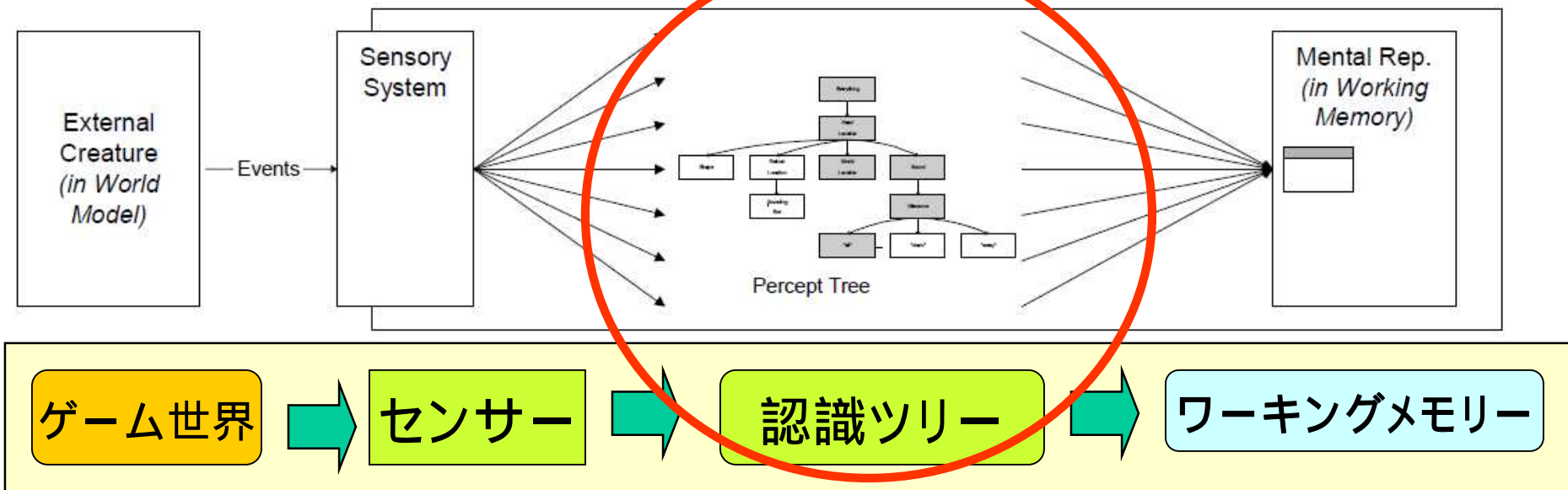


対象(ボール)が遮蔽物に隠れても、
履歴の記憶から動きを予測することが出来る。



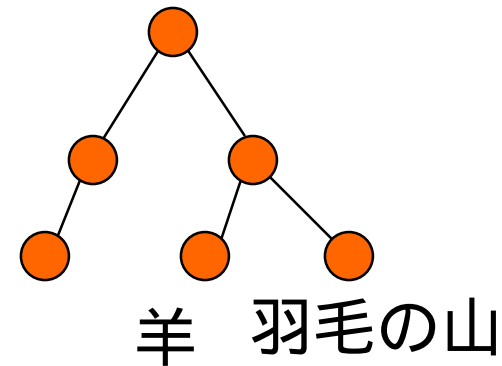
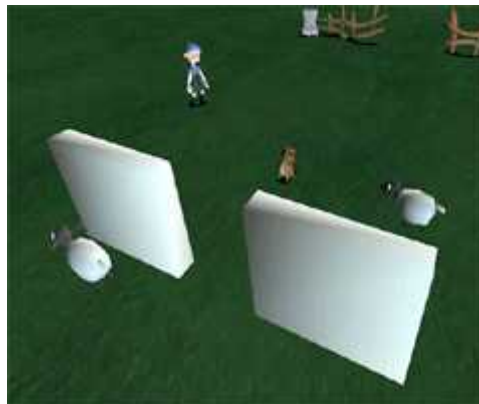
認識ツリーによって可能になること

判断間違い



(感覚から世界を再構成している)

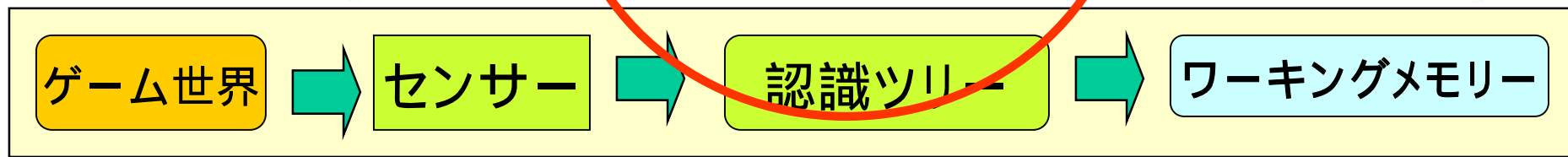
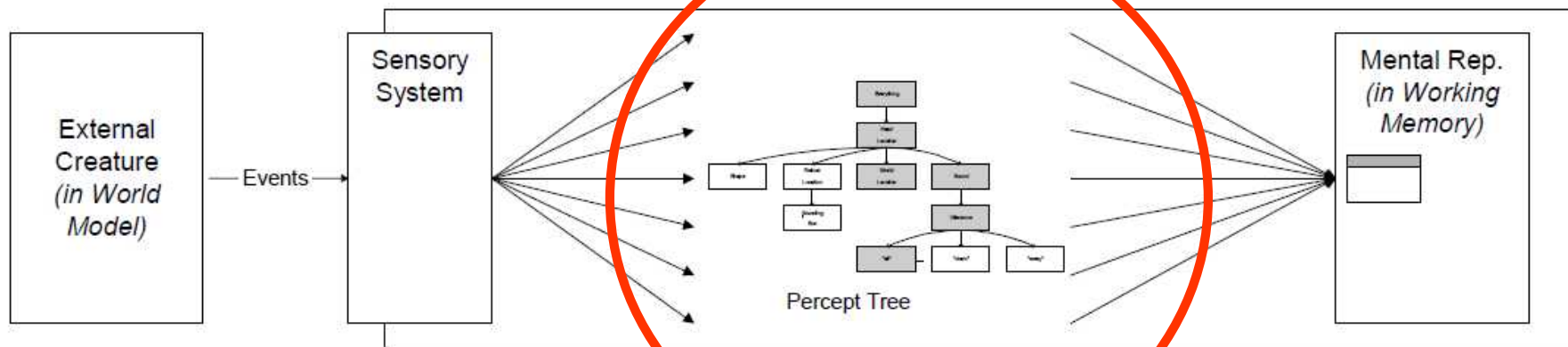
認識ツリーは、間違いを犯すこともある。



人間らしい間違い

認識ツリーによって可能になること

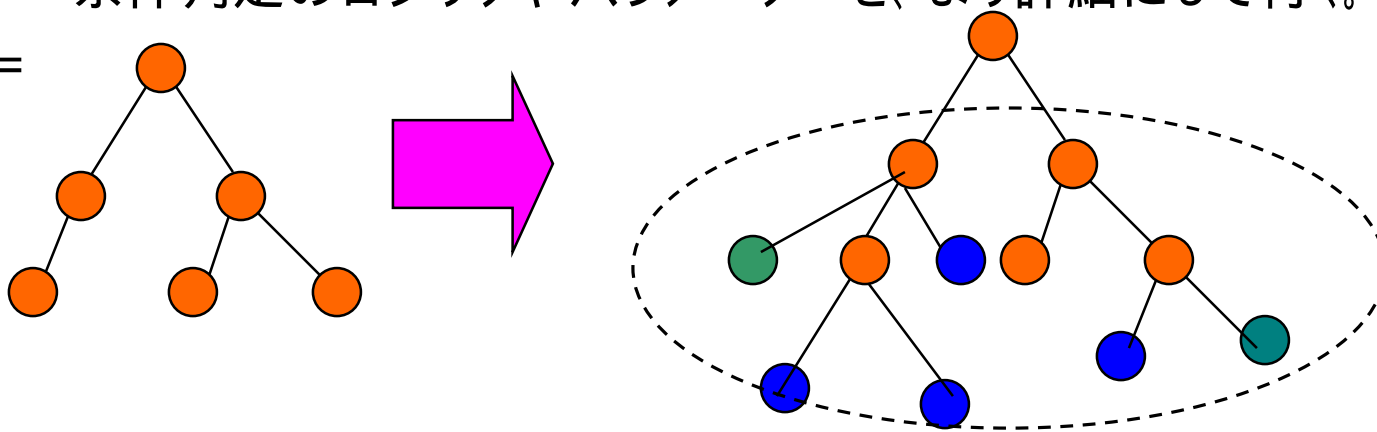
学習



(感覚から世界を再構成している)

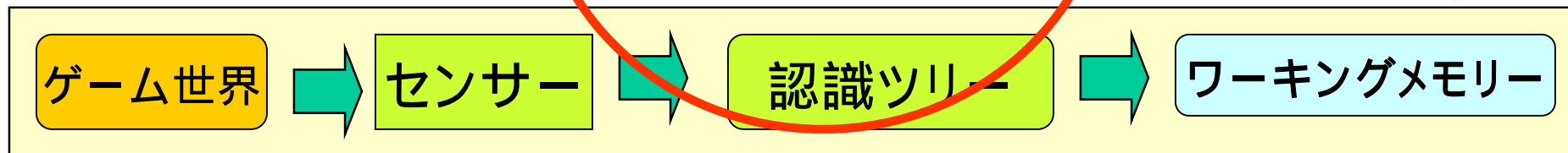
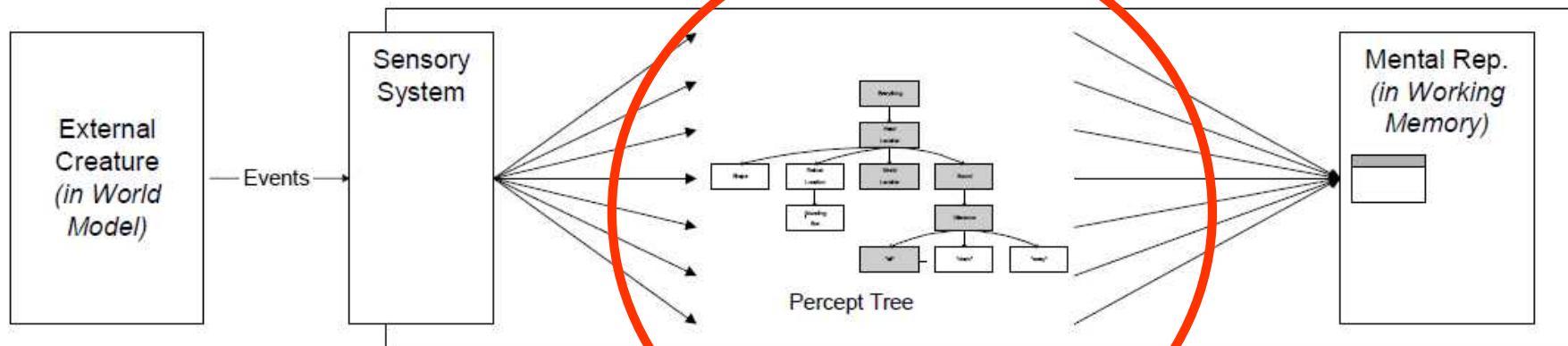
(I) 精緻化 = 条件判定のロジックやパラメーターを、より詳細にして行く。

(II) 発展 =



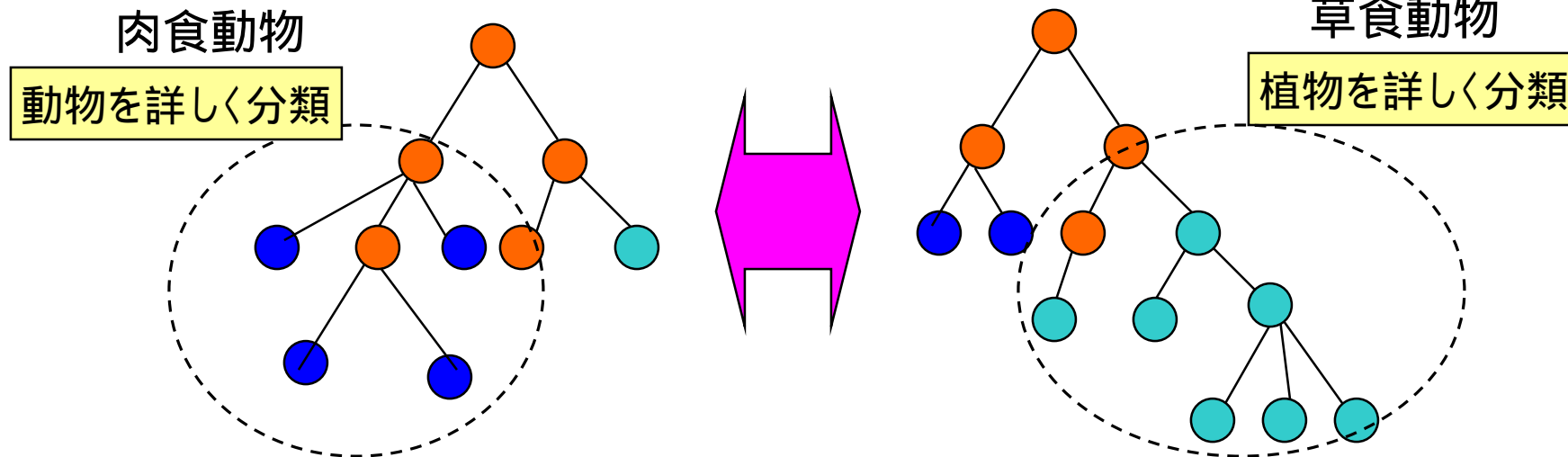
一般的な概念の学習

認識ツリーによって可能になること 個性を付け



(感覚から世界を再構成している)

生物ごとに必要な認識の仕方は違う。



認識ツリーによるNPCの個性化

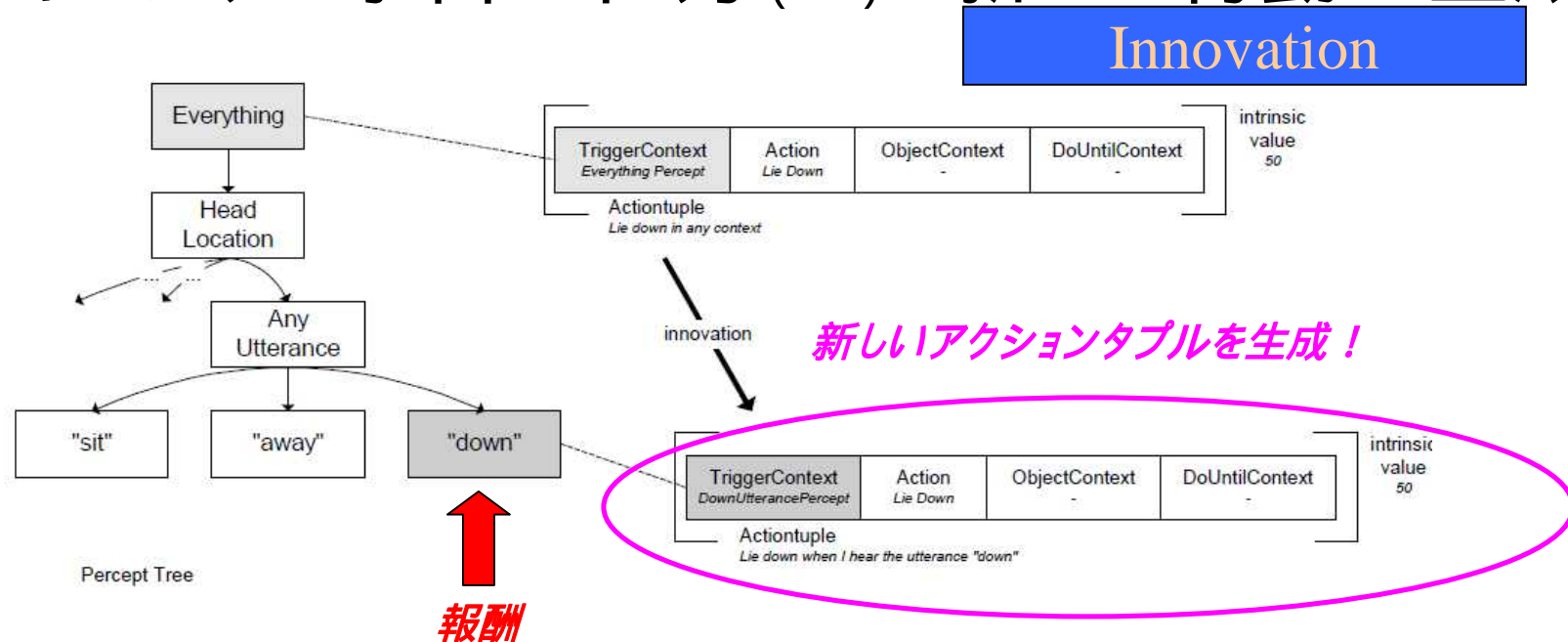
C 4 における学習

強化学習

(reinforcement learning)

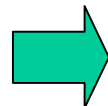
(報酬を与えたことはより高い確率で起こり、罰を与えたものは行動しないようにする)

C 4 における学習の仕方 (1) 新しい行動の生成



プレイヤーがマイクを通して命令を出す。
Duncan は、それを認識して行動する。
プレイヤーはそれが正解なら報酬を与える。
もしある行動の評価値が高く、トリガーの相関が強ければ新しく行動を生成する。

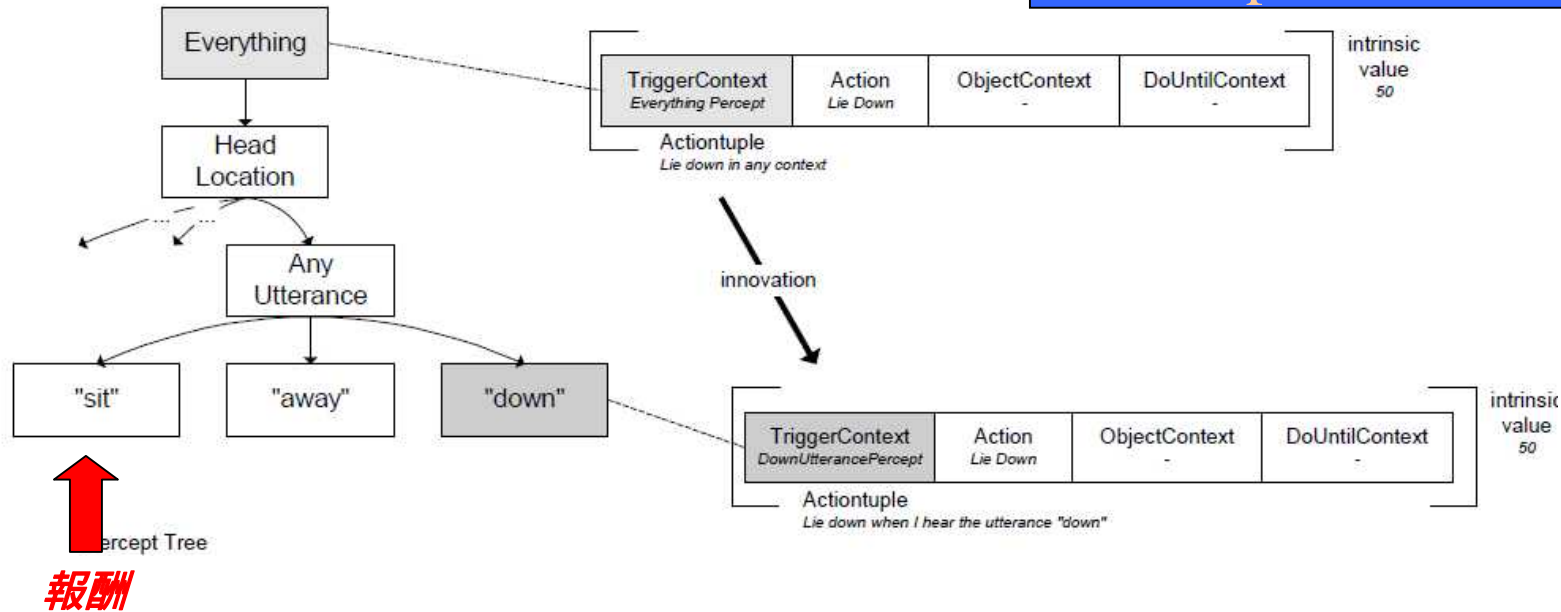
ここではdownに十分に高い報酬が与えられる場合に、
Everything PerceptとDownUtterancePerceptの相関が強いので、
トリガーを置き換えた新しい行動を生成する。



新しい行動の生成

C 4 における学習の仕方 (2) 判別能力の向上

State-space discovery



プレイヤーがマイクを通して命令を出す。
Duncan は、それを認識して行動する。
プレイヤーはそれが正解なら報酬を与える。

よい報酬が得られた情報は、よい例として蓄積する。
例えば「SIT」という発音だと判別してよい報酬が得られれば、
その「SIT」という発音データはよい「SIT」の発音のサンプルとしてモデル化する。

➡ 判別能力の精度の向上

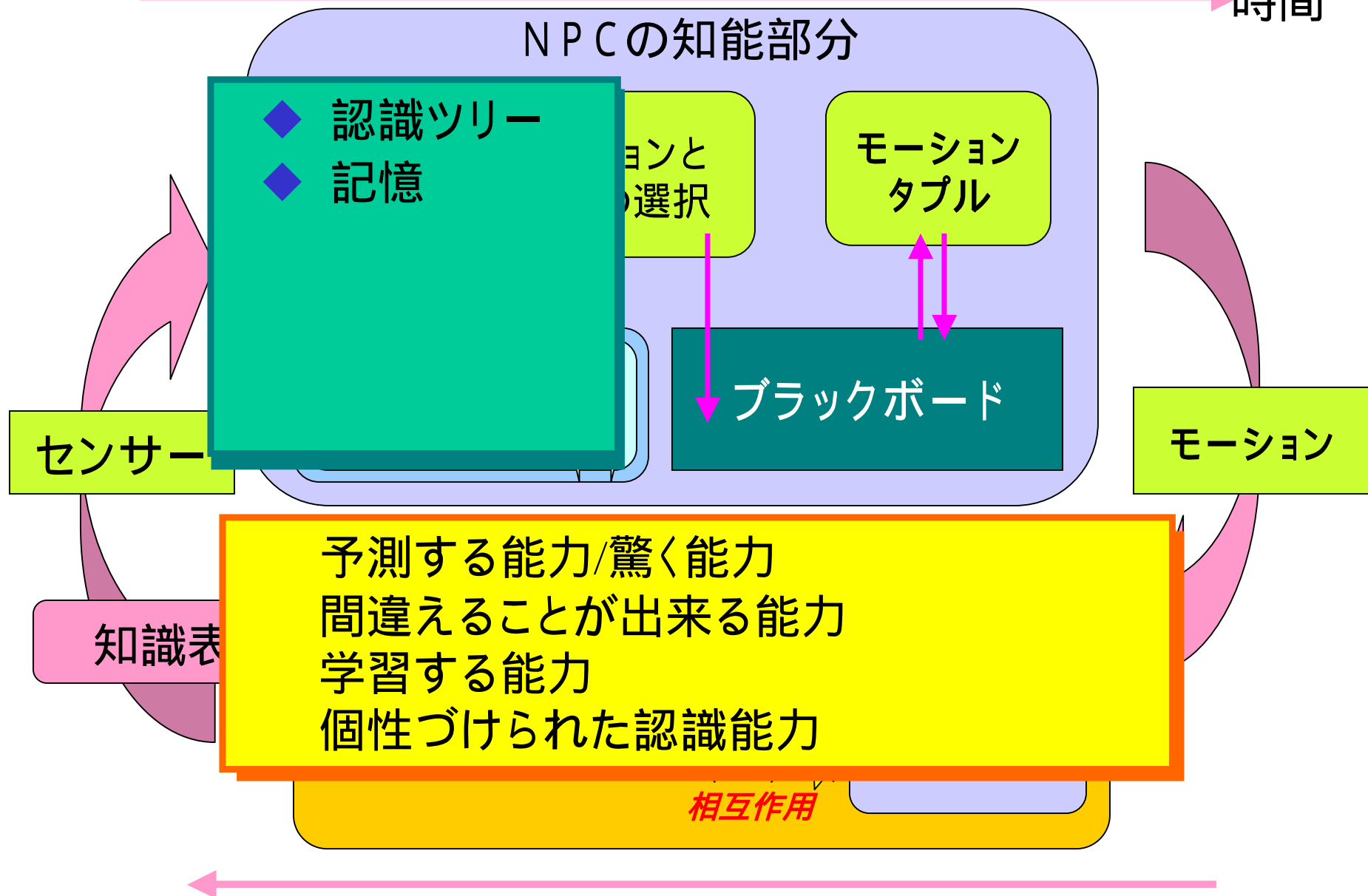
Demo Movie Persistence

バーチャル牧羊犬*Duncan*は、記憶を持つ。
一体それが、行動にどう現れるか、見てみよう！

<http://web.media.mit.edu/~bruce/whatsnew.html>
(MIT Media Lab Bruce Blumberg's Web Page)

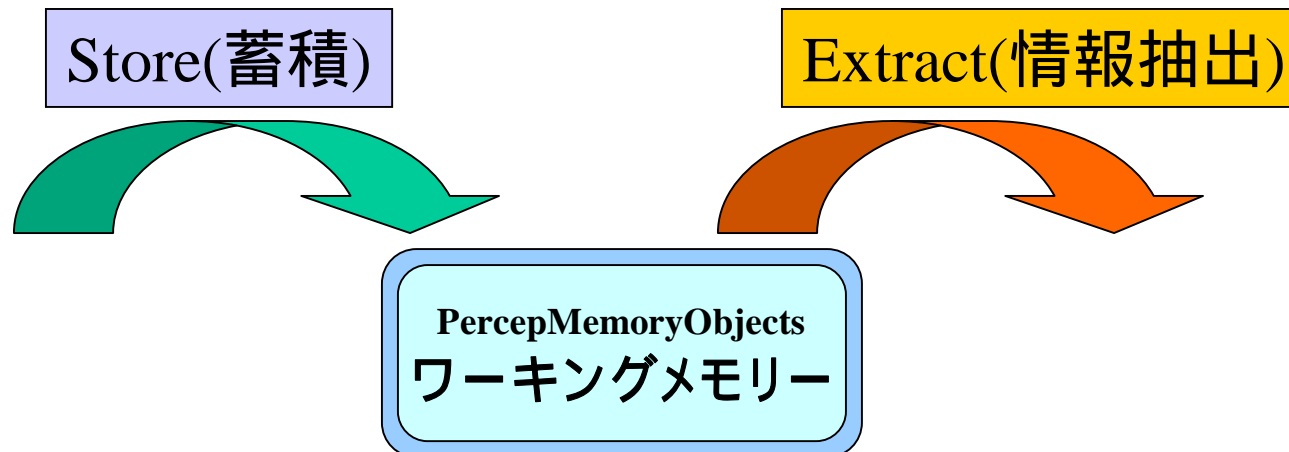
MIT メディアラボ C4 アーキテクチャー

時間



第2章 まとめ

- (1) C4アーキテクチャーは、FPS-AI の基礎となったアーキテクチャーである。
- (2) 構成的なアーキテクチャーにおいては、モジュールを入れ替えることでAIに個性を与えたり、発展させることができる。
- (3) 情報の入り口である認識過程を変更することで、記憶の形を決定することができる。
- (4) 記憶を蓄積し、そこから情報を抽出することで、環境を学習した行動を取らせることができる。



第5回セミナーの構成

はじめに

ゲームAI連続セミナーのご紹介

Enjoy AI!

第1部 ゲームAI 技術解説 (100分) (講師: 三宅)

第1章 概論 (10分)

第2章 C4アーキテクチャー (20分)

第3章 遺伝的アルゴリズム (30分)

第4章 ニューラルネットワーク (30分)

第5章 発展 (10分)

各章の終わりに質疑応答を行います。

(休憩 20分)

第2部 ディスカッション (2時間) (アイデア: IGDAゲームAI運営委員)

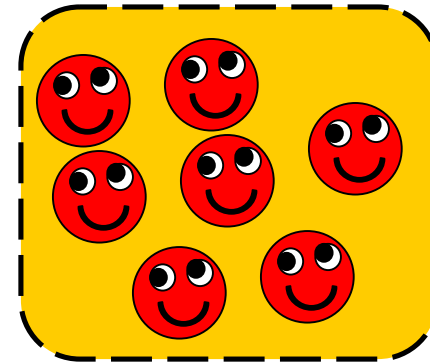
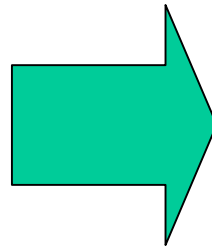
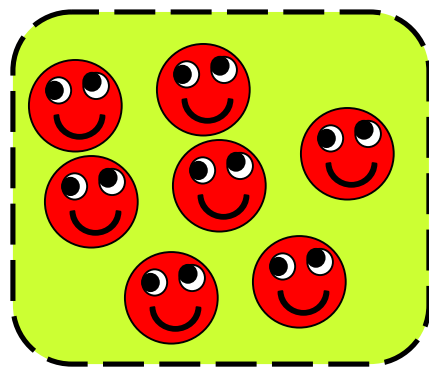
(総合司会 + コーディネート: 長久、グループ司会者)

(I) グループ討論 (80分)

(I) まとめ (30分)

第2章 遺伝的アルゴリズム

1. 遺伝的アルゴリズムのイメージ
2. 遺伝的アルゴリズムの原理
3. 遺伝的アルゴリズムのゲームへの応用
4. 遺伝的アルゴリズムの展望

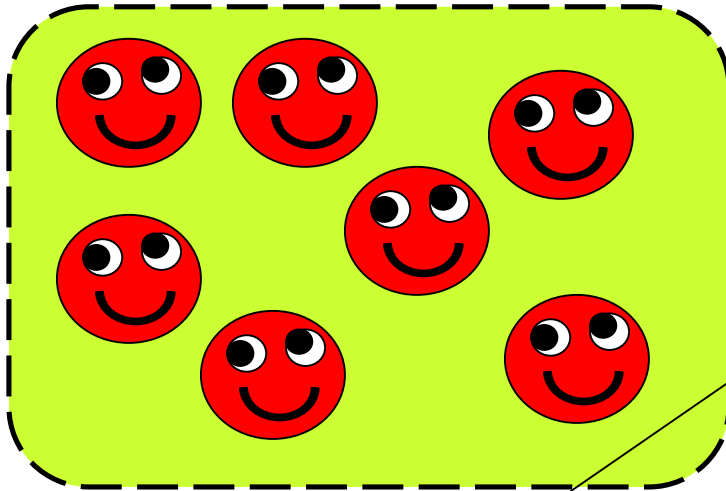


1. 遺伝的アルゴリズムのイメージ

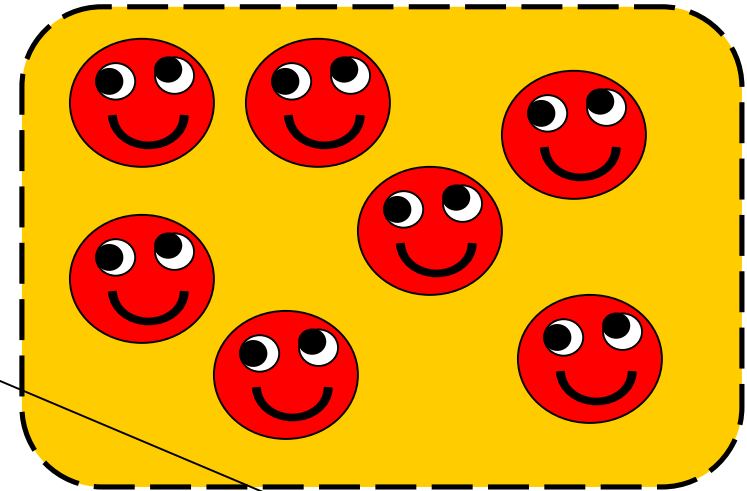
遺伝的アルゴリズム

集団を一定の方向に進化させる方法

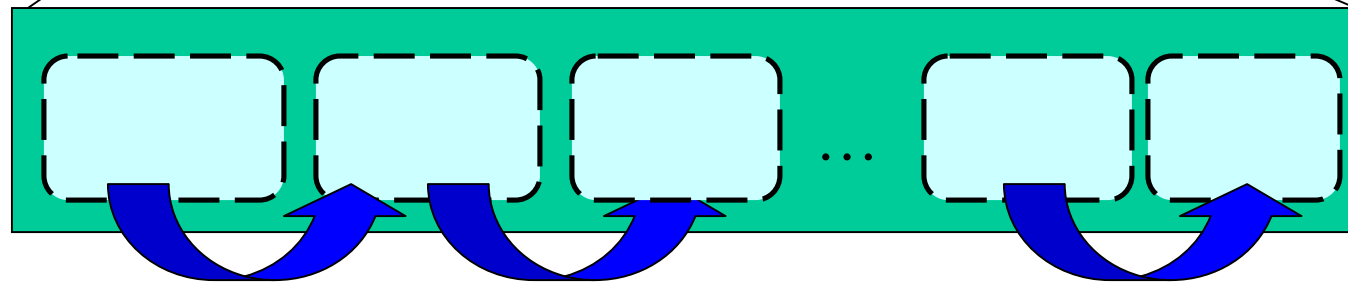
最初の世代



新世代(100～世代後)

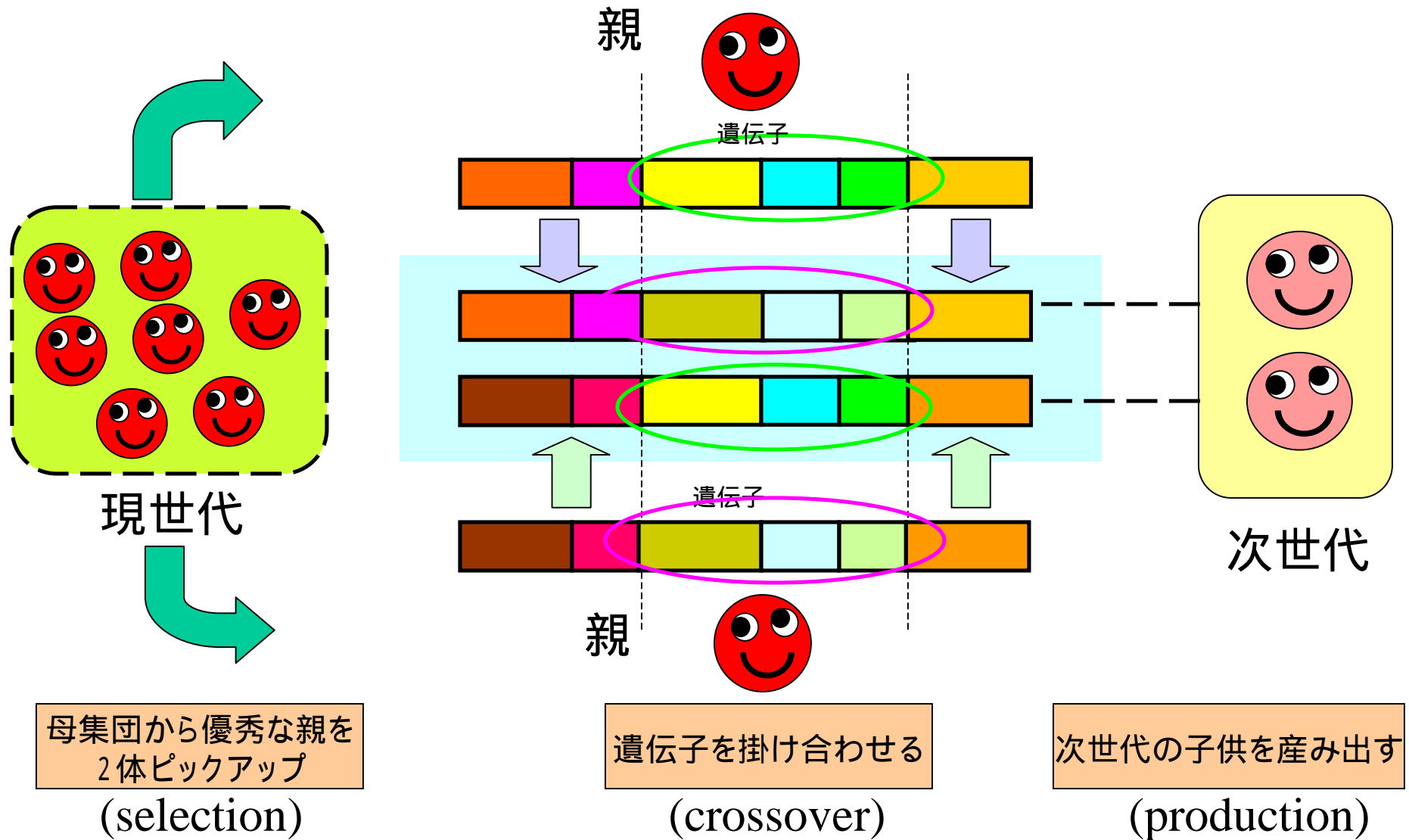


世代を経て進化させる



一つの世代が次の世代を交配によって産み出す

遺伝的アルゴリズムの仕組み

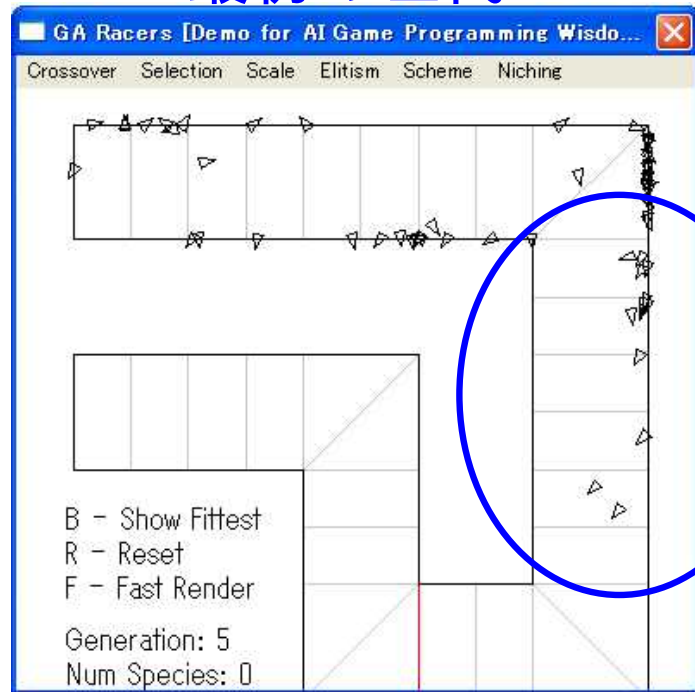


このサイクルをくり返すことで世代を進めて望ましい集団を産み出す

(例) GA Racer

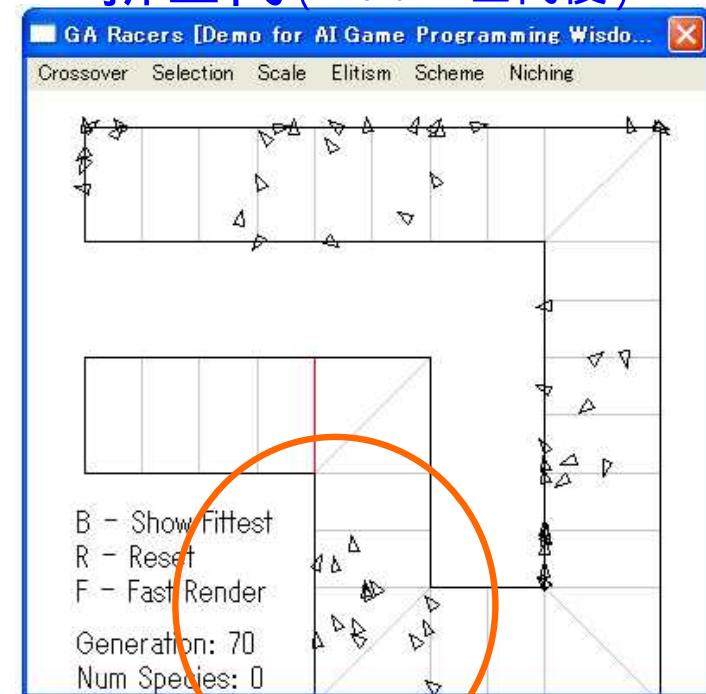
遺伝的アルゴリズムによって、遠くまで到達できるレーサーを作成する。

最初の世代



最初はここまでしか
たどり着けないけど...

新世代(100~世代後)



だんだんと遠くまで、
たどりつけるようになる。

Mat Buckland, "Building Better Genetic Algorithm", 11.4., AI Game Programming Wisdom 2
(CD-ROMにソースコードと実行ファイルがあります)

(例) アステロイド

最初の世代



最初、船()は隕石()に
ぶつかってばかり...

新世代(4世代後)



次第に隕石を避けることが
できるようになる。

Brian Schwab, "Genetic algorithm", Chapter.20., AI Game Engine Programming,
Charles River Media (2004) (CD-ROMにソースコードと実行ファイルがあります)

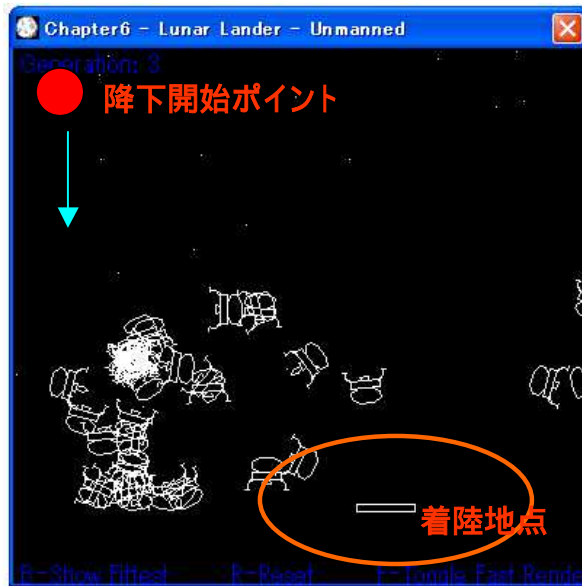
(例) ムーンランディング(月着陸)

落下法則(物理法則)のもとで降下ポイントから台座に垂直に着陸する操作を見出す

3 世代

20 世代

35 世代



最初は全く見当違い。

だんだんと近付いて来る。

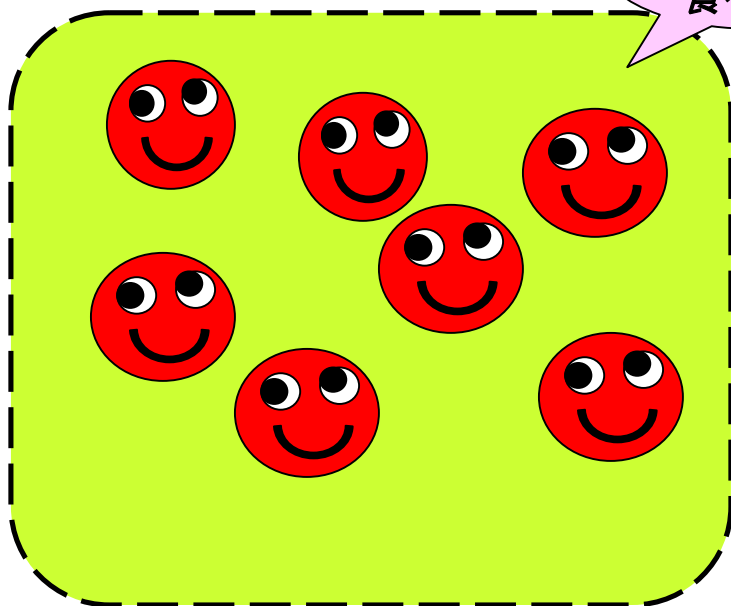
Mat Buckland, Andre Lamothe, “Moon Landings Made Easy”, chapter.6.,
AI techniques for game programming, Premier Press (2002)
(CD-ROMにソースコードと実行ファイルがあります)



(例) アストロノーカ

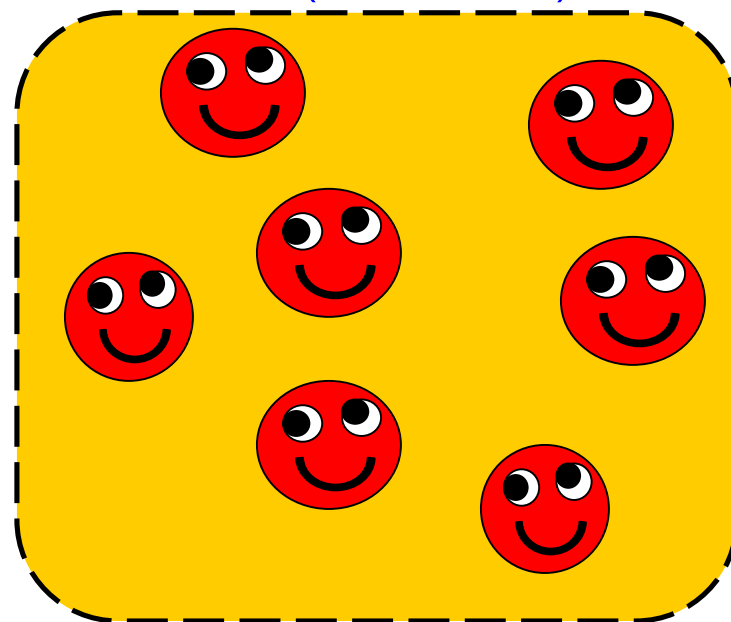
最初の世代

野菜
食べたい



最初はすぐに罠にかかるけど

新世代(5～世代後)



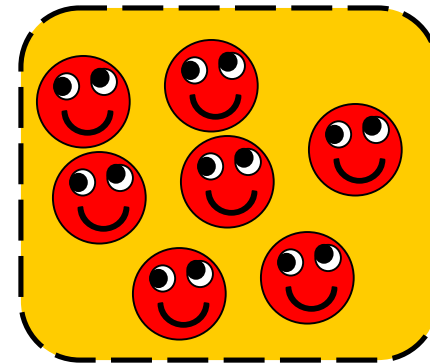
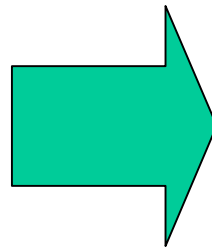
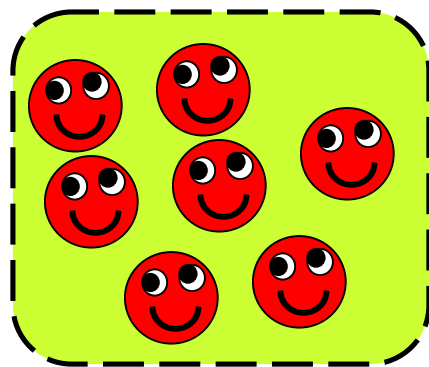
だんだんと罠にかからないようになる

MuuMuu, プレイステーション用ソフト「アストロノーカ」(Enix, 1998)

<http://www.muumu.com/games/astro/>

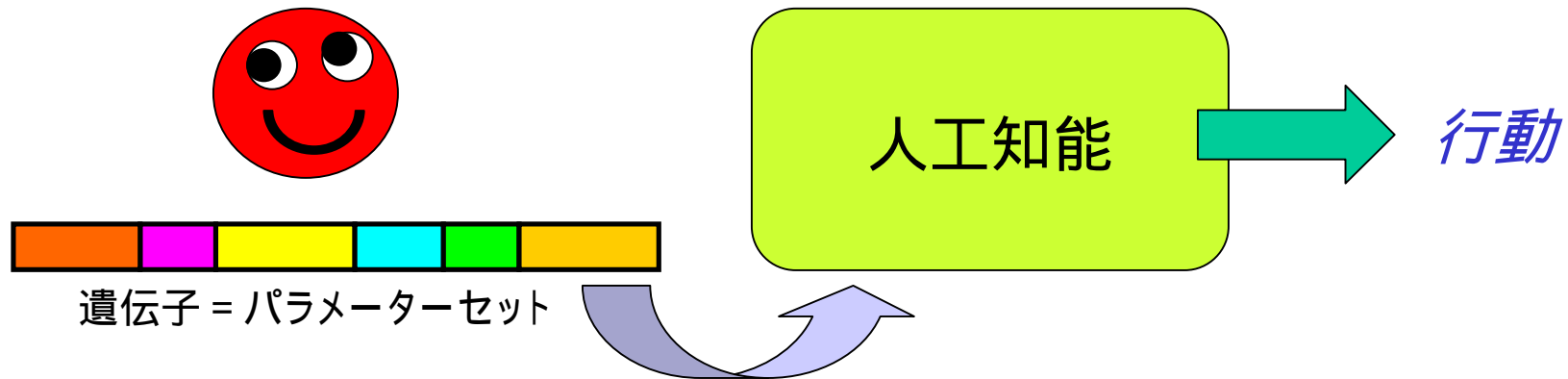
第2章 遺伝的アルゴリズム

1. 遺伝的アルゴリズムのイメージ
2. 遺伝的アルゴリズムの原理
3. 遺伝的アルゴリズムのゲームへの応用
4. 遺伝的アルゴリズムの展望



2 . 遺伝的アルゴリズムの原理

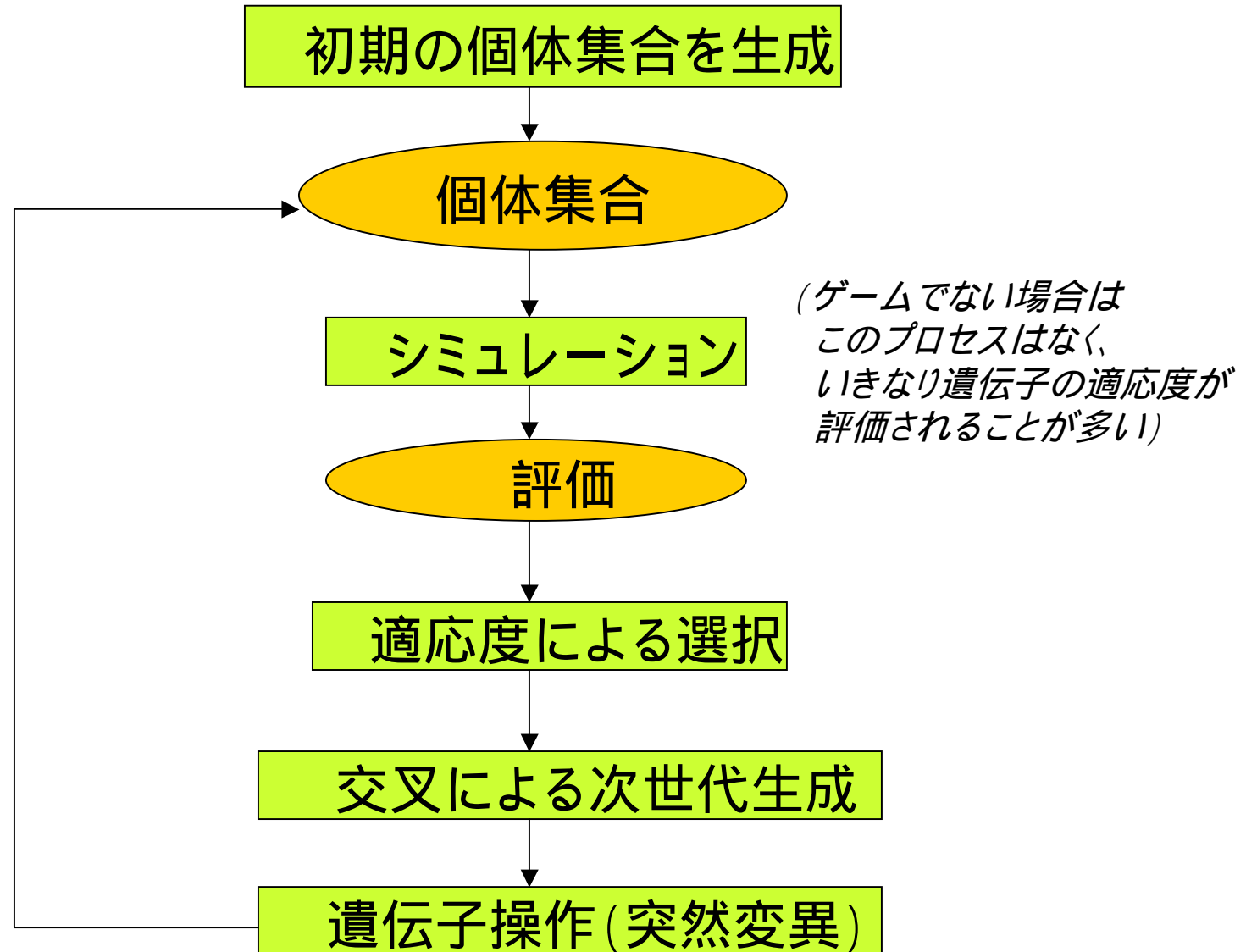
遺伝的アルゴリズムが使える時



パラメーターセットが身体的特性・知能を特徴付ける

パラメーターのセットによって、NPCを定義する。

遺伝的アルゴリズムによる個体集団進化のシーケンス



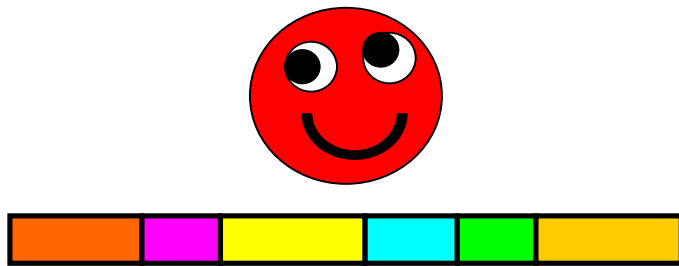
以下、それぞれのステップの
技術を詳細に解説します。

Mat Buckland, "Building Better Genetic Algorithm", 11.4., AI Game Programming Wisdom 2
(CD-ROMにソースコードと実行ファイルがあります)

以下の解説は、上記の解説に即します。
いろいろ教科書や解説があるけれど、ゲーム開発に関しては
上記の解説が最も優れているし、
ソースコードもあるので利用しやすいと思います。

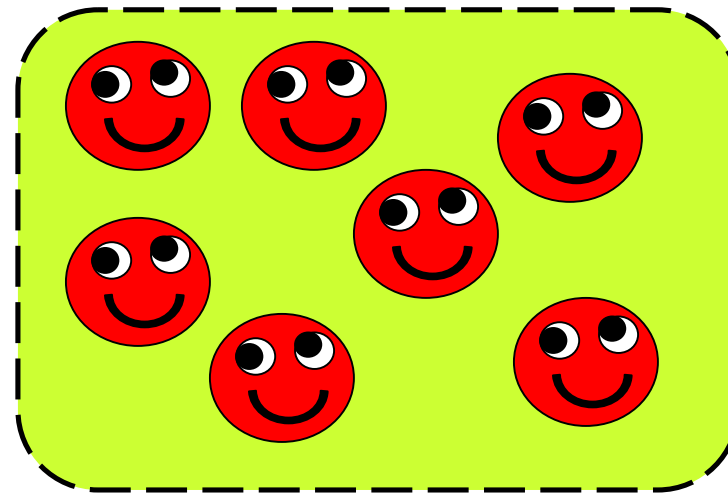
初期の個体集合を生成

個体を多数 (GAにはある程度の母数が必要) 用意し、各NPCに遺伝子コードを設定し、初期値を設定する。



数値、或いは、記号を要素とする。

数値表現としては、
各要素を実数で表す「実数表現」と
2進数(1, 0)で表す「バイナリー表現」がある。



なるべく、パラメーターの初期値は、まばらになるようにしよう！
(普通は乱数などを使って分布が均一になるようにする)

【遺伝的アルゴリズムの思想】さまざまな遺伝子による
集団の多様性を維持しながら、一つの方角に進化させる。

(解説)「バイナリー表現」と「実数表現」

遺伝子コードには、いろいろな表現がある

実数表現

12.3	3.1	56.7	5.4	2.0	23.9
------	-----	------	-----	-----	------

バイナリー表現

0101010	0110101	10100101	010100010	101010	1010101
---------	---------	----------	-----------	--------	---------

記号表現

B	C	A	D	F	E
---	---	---	---	---	---

ベクトル表現

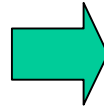
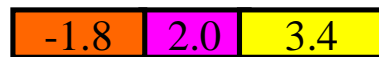
...

(解説) 遺伝子型 (遺伝子の上の表現の仕方)

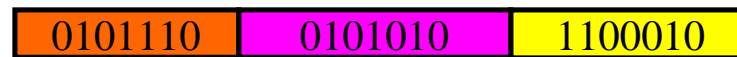
実際のパラメータと違う表現でもよい

(例) $-6.4 < x, y, z < 6.3$

実数表現



バイナリー表現



パラメーター (実数) を
そのまま使う

2進数に表現しなおす

$$(x + 6.4) \times 10 = x'$$

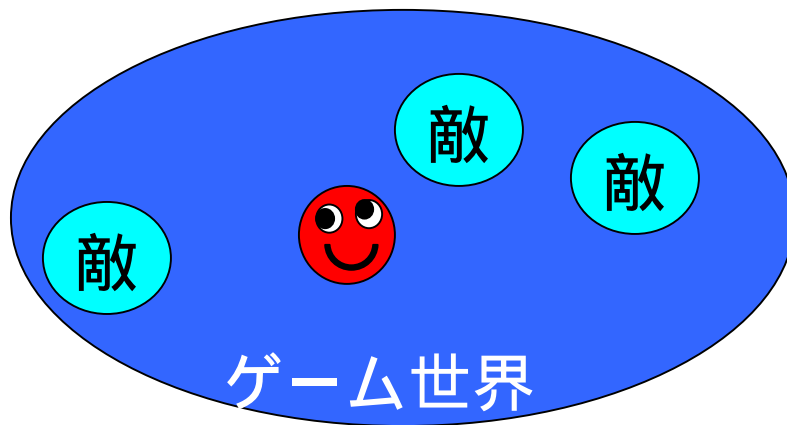
表現の仕方によって突然変異のさせ方が違って来ます(後述)。

シミュレーションとNPCの評価

NPCが生きるゲーム世界の中で、実際に一定時間動作させるなどして、製作者がNPCに望む目標に対する評価値(達成値)をつける。

ゲーム製作者の意図を反映する評価関数を作る

(例) 強いNPCを作りたければ、 $\text{評価値} = 0.7 * \text{撃破数} + 0.3 * \text{残りHP}$
取り合えず生き延びることできるNPCなら、 $\text{評価値} = \text{生存時間}$











順位	評価値	
1位	86.3	
2位	78.4	
3位	75.3	
....		
...		
100位	38.2	

君はこの世界でどれだけ僕が求めるにふさわしいのだ？

遺伝子を評価するのではなく、その遺伝子を持つ個体が、世界でどれだけ優秀であるかを測る。

評価値から適応度を計算する

評価値から、その個体の環境に対する適応度を計算する。
評価値が大きいほど、適応度は大きくなるようにしておく。

順位	評価値		順位	適応度	
1位	86.3		1位	9.32	
2位	78.4		2位	8.83	
3位	75.3		3位	7.81	
....				
...			...		
100位	38.2		100位	0.02	

評価値とは、その環境で達成した行為の点数のこと。
適応度とは、環境にどれくらい対応しているかを表す。

両者の対応関係は、比例関係にあるならどう作ってもよい。

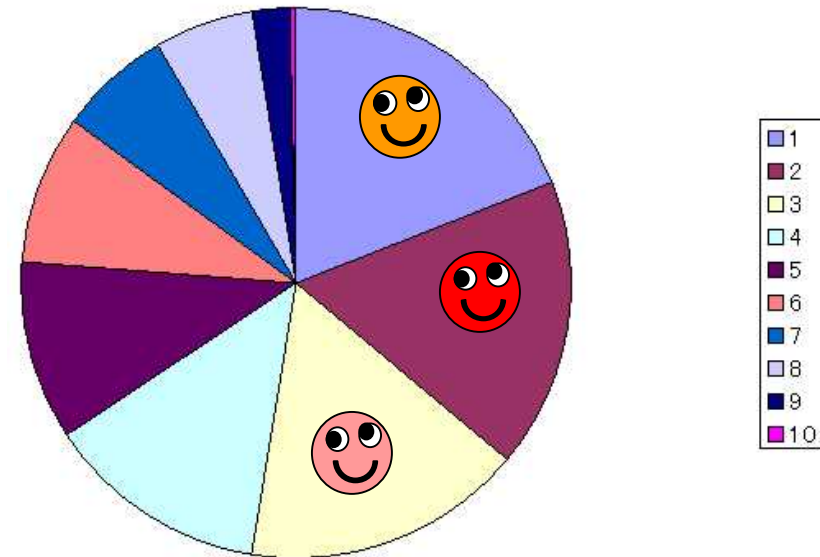
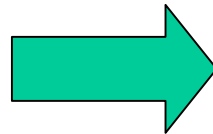
(例) 同じでもいいや。 $\text{適応度} = \text{評価値}$

上位の点数は、差に意味がないから $\text{適応度} = \log(\text{評価値}/100)$ など。

選択

生き延びて子孫(offspring)を残せる個体を決定する

順位	適応度	
1位	0.93	😊
2位	0.81	😊
3位	0.79	😊
4位	0.63	😊
5位	0.51	
6位	0.44	
7位	0.32	
8位	0.28	
9位	0.10	
10位	0.02	



適応度比例方式(ルーレット選択)

... 適応度の大きさに比例した確率で生き延びて親になれる。

(無作為にダーツを投げて親を決めるイメージ。)

大きな適応度の領域ほどあたりやすい。プログラムでは勿論、乱数を使う)

(解説) スケーリング

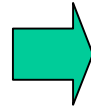
(問題) 適応度比例方式では、あまりに小さい適応度のNPCにチャンスがない。

適応度の分布を少し緩和してからルーレットしよう！

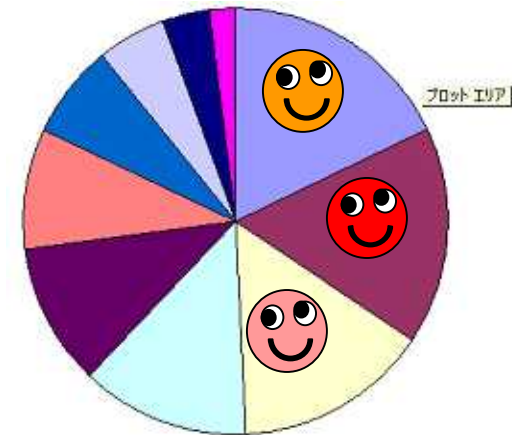
いろいろなスケーリング法(1)

順位スケーリング

順位	適応度	
1位	0.93	😊
2位	0.81	😊
3位	0.79	😊
4位	0.63	😊
5位	0.51	😊
6位	0.44	😊
7位	0.32	😊
8位	0.28	😊
9位	0.10	😊
10位	0.02	😊



順位	適応度	
1位	10	😊
2位	9	😊
3位	8	😊
4位	7	😊
5位	6	😊
6位	5	😊
7位	4	😊
8位	3	😊
9位	2	😊
10位	1	😊



(解説) スケーリング

(問題) 適応度比例方式では、あまりに小さい適応度のNPCにチャンスがない。

適応度の分布を少し緩和してからルーレットしよう！

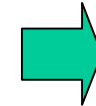
いろいろなスケーリング法(2)

シグマスケーリング

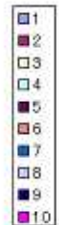
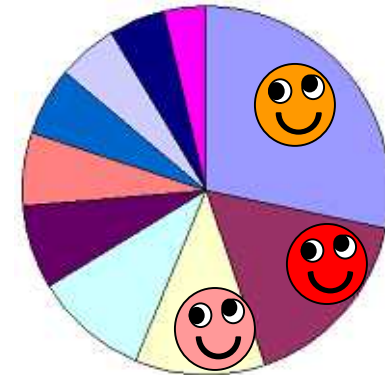
順位	適応度	
1位	0.93	😊
2位	0.81	😊
3位	0.79	😊
4位	0.63	😊
5位	0.51	😊
6位	0.44	😊
7位	0.32	😊
8位	0.28	😊
9位	0.10	😊
10位	0.02	😊



順位	適応度	
1位	1.45	😊
2位	0.43	😊
3位	-0.02	😊
4位	-0.12	😊
5位	-0.34	😊
6位	-0.45	😊
7位	-0.48	😊
8位	-0.54	😊
9位	-0.57	😊
10位	-0.68	😊



+ 1



新しい適応度 = (適応度 - 適応度の平均値) / 2
(: 標準偏差、定数)

(解説) スケーリング

(問題) 適応度比例方式では、あまりに小さい適応度のNPCにチャンスがない。

適応度の分布を少し緩和してからルーレットしよう！

いろいろなスケーリング法(3)

ボルツマンスケーリング

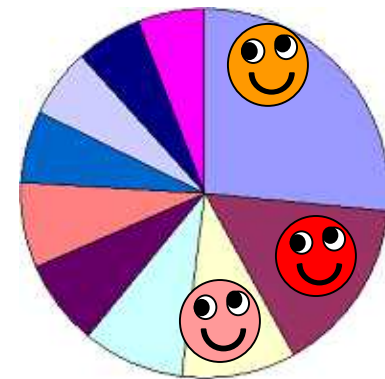
順位	適応度	
1位	0.93	😊
2位	0.81	😊
3位	0.79	😊
4位	0.63	😊
5位	0.51	😊
6位	0.44	😊
7位	0.32	😊
8位	0.28	😊
9位	0.10	😊
10位	0.02	😊



順位	適応度	
1位	1.65	😊
2位	0.53	😊
3位	-0.02	😊
4位	-0.12	😊
5位	-0.24	😊
6位	-0.25	😊
7位	-0.38	😊
8位	-0.40	😊
9位	-0.47	😊
10位	-0.49	😊



+ 1

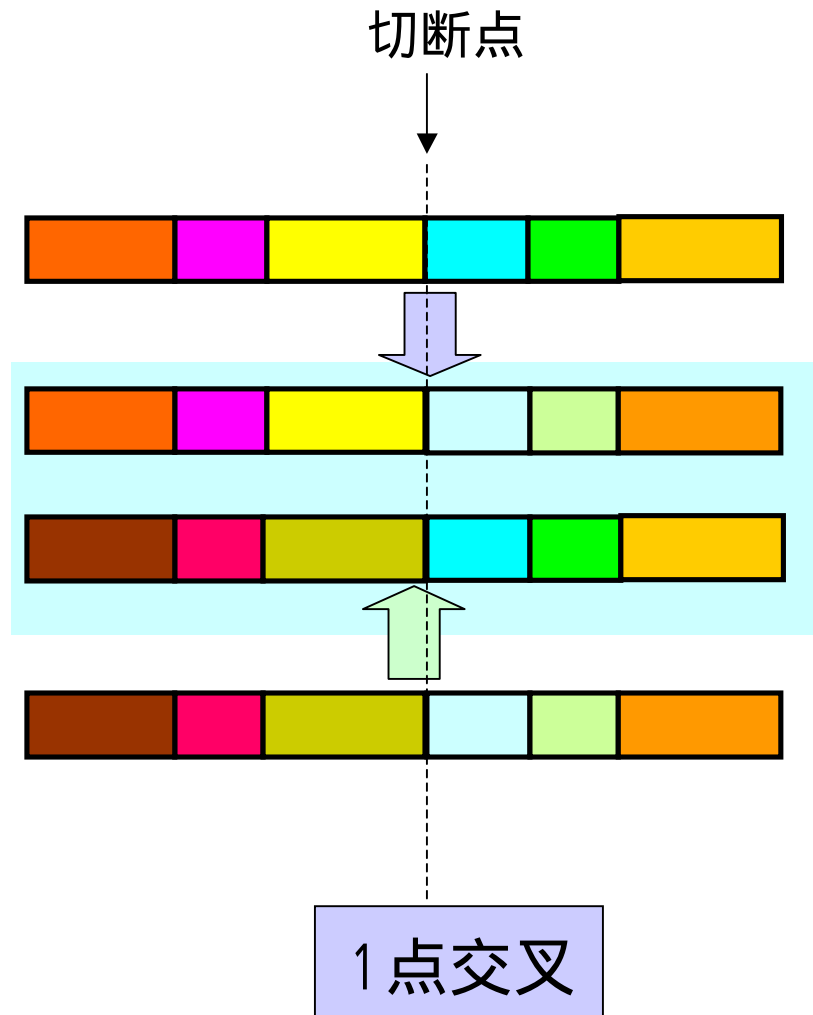


$$\text{新しい適応度} = \frac{\exp(\text{適応度}/T)}{\sum \exp(\text{適応度}/T)}$$

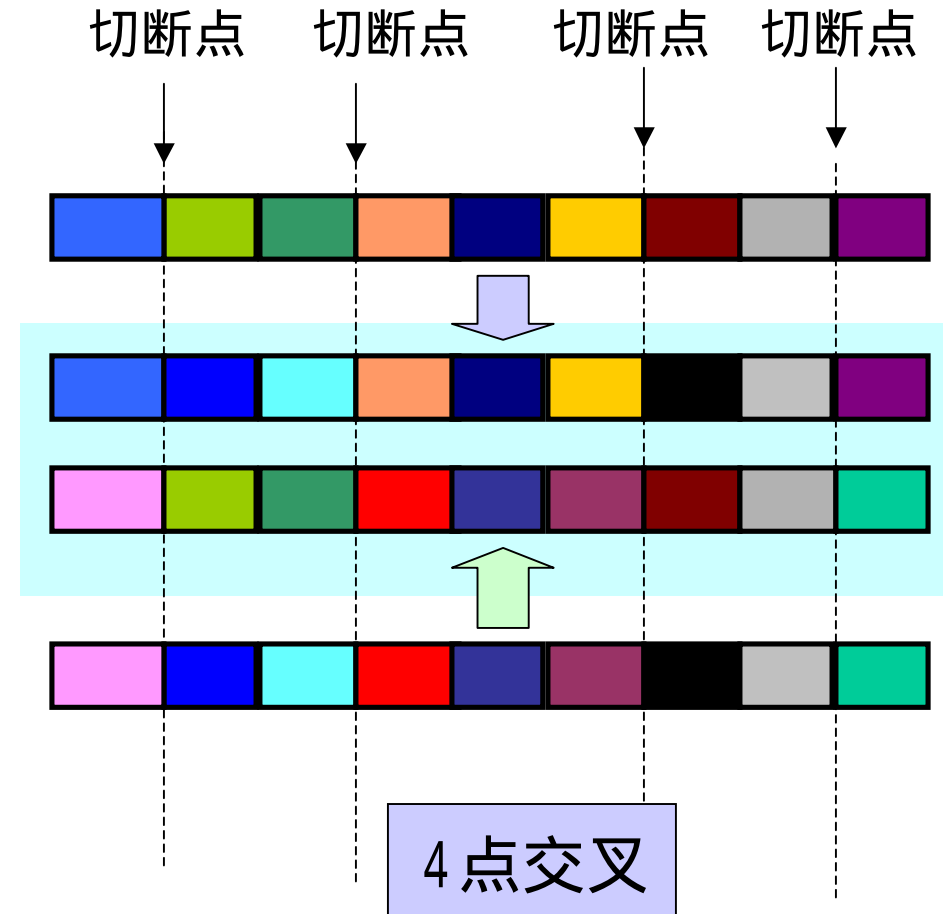
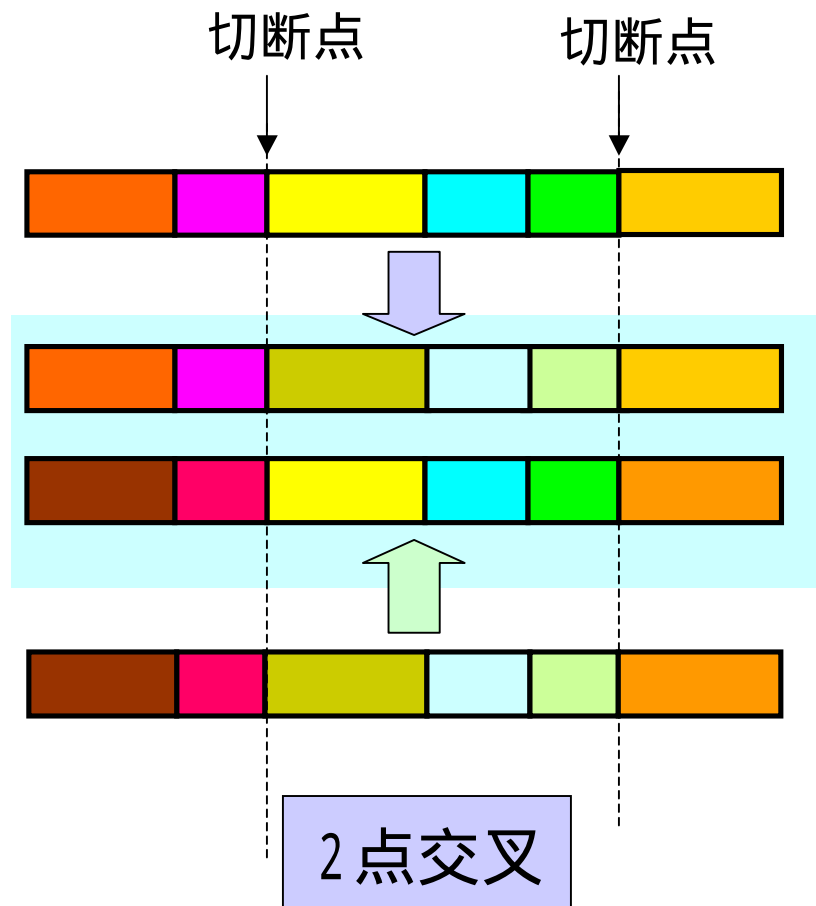
$T = 3 \times \text{NPC の数}$

交叉による次世代生成

選んだ2つの親の遺伝子を交叉(crossover)させる。



交叉による次世代生成



基本的に、1点交叉が基本です。

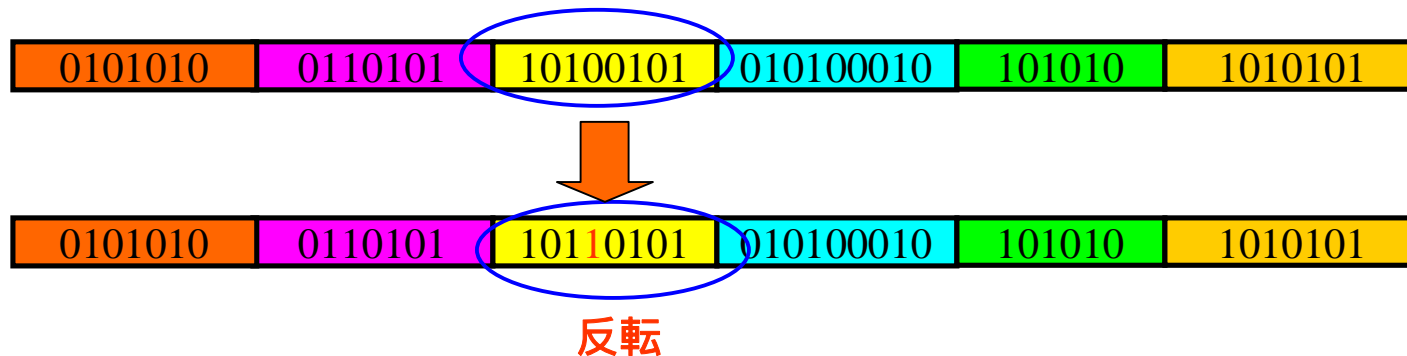
切断点が多いということは「性質が保存されにくく」次世代に分岐の多い進化を促すことです。
より多様なNPCを産み出して、ベストフィットするNPCを探すことを意味します。

1点交叉から始めて結果を見て、適度な交叉を見出すことをお勧めします。

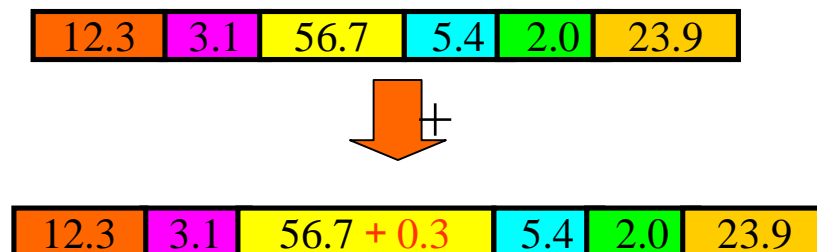
遺伝子操作 (突然変異)

ある確率 (突然変異率) で、遺伝子コード上の遺伝子 (内容) をランダムに対立遺伝子に書き換える。

バイナリー表現



実数表現

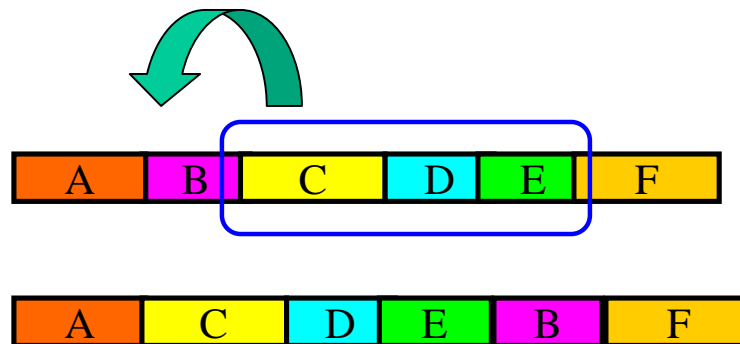
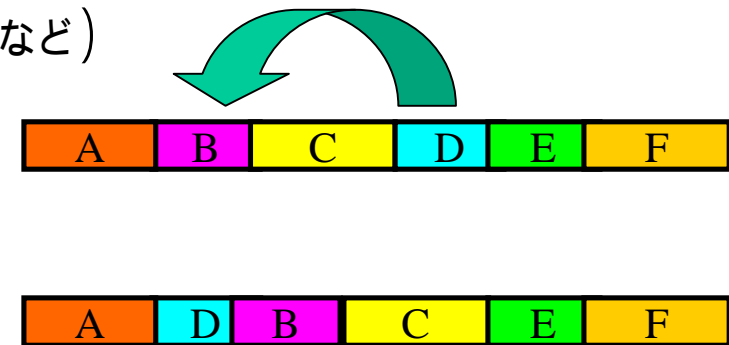
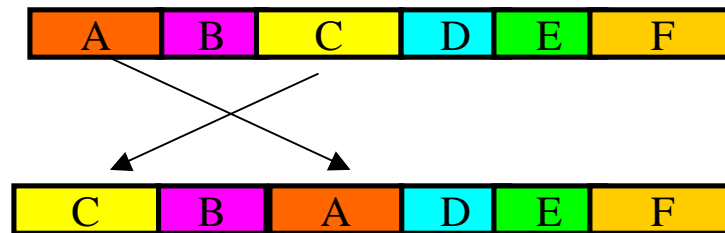


遺伝子に多様性を与える

遺伝子操作 (突然変異)

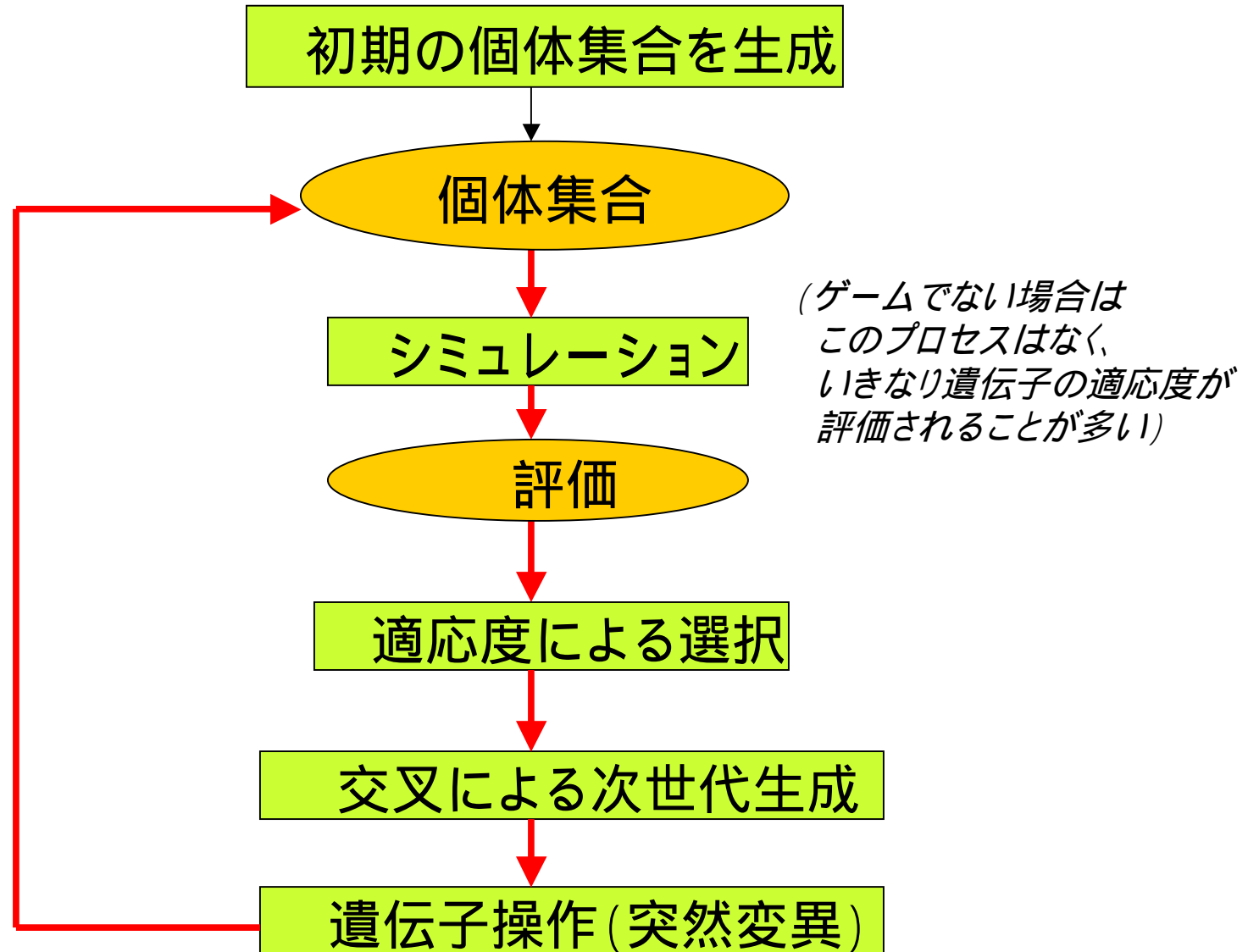
ある確率 (突然変異率) で、遺伝子コード上の遺伝子 (内容) をランダムに対立遺伝子に書き換える。

記号順列表現 (順序に意味がある場合。ルート検索など)



遺伝子に多様性を与える

このプロセスを何度もくり返すことでNPCの集合は進化します



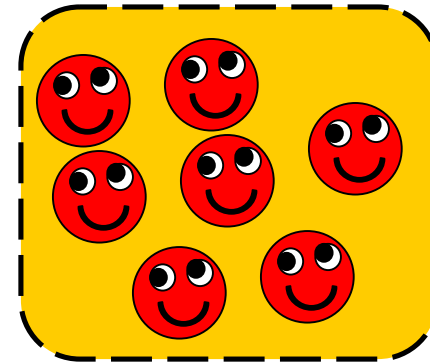
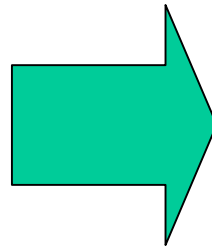
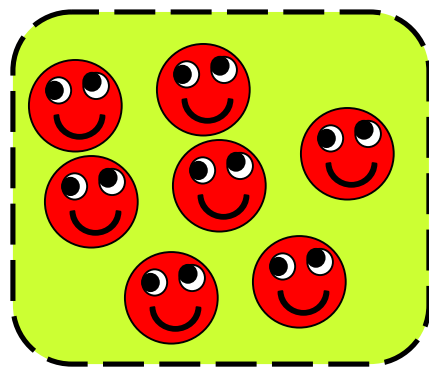
以下、詳細の説明終了です。

Mat Buckland, "Building Better Genetic Algorithm", 11.4., AI Game Programming Wisdom 2
(CD-ROMにソースコードと実行ファイルがあります)

より深い理解のためには、
上記の文書やソースコードを研究してみましょう！

第2章 遺伝的アルゴリズム

1. 遺伝的アルゴリズムのイメージ
2. 遺伝的アルゴリズムの原理
3. 遺伝的アルゴリズムのゲームへの応用
4. 遺伝的アルゴリズムの展望



3 . 遺传的アルゴリズムのゲームへ応用

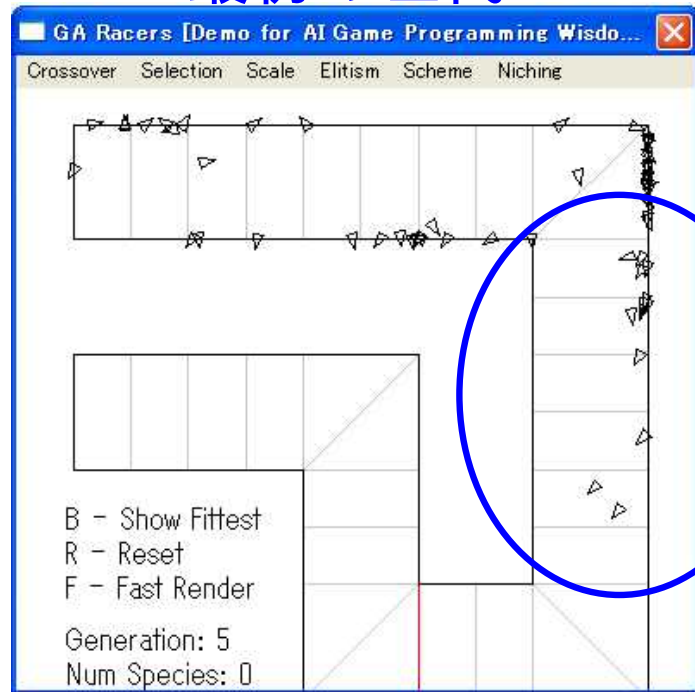
デモ解説

デモの組み立て方を見ながら、
ゲームへの応用へのアイデアを考えよう！

(例) GA Racer

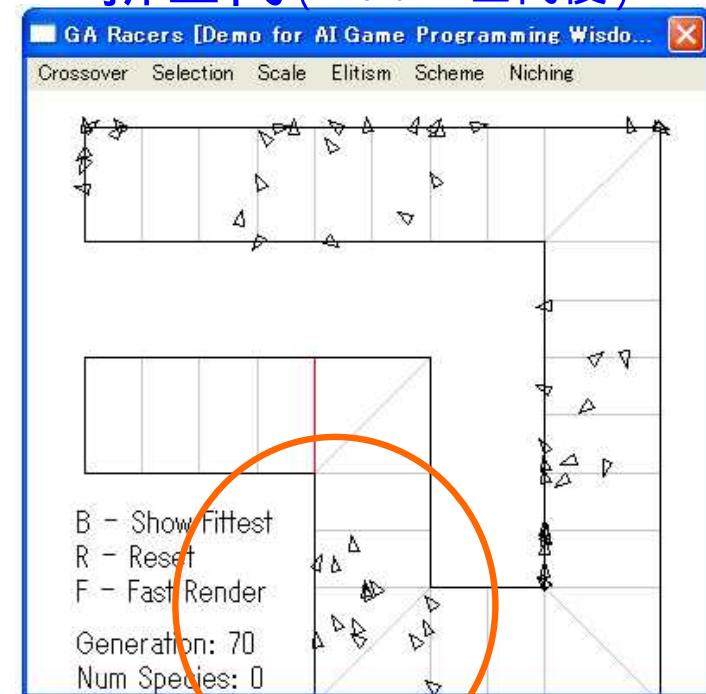
遺伝的アルゴリズムによって、遠くまで到達できるレーサーを作成する。

最初の世代



最初はここまでしか
たどり着けないけど...

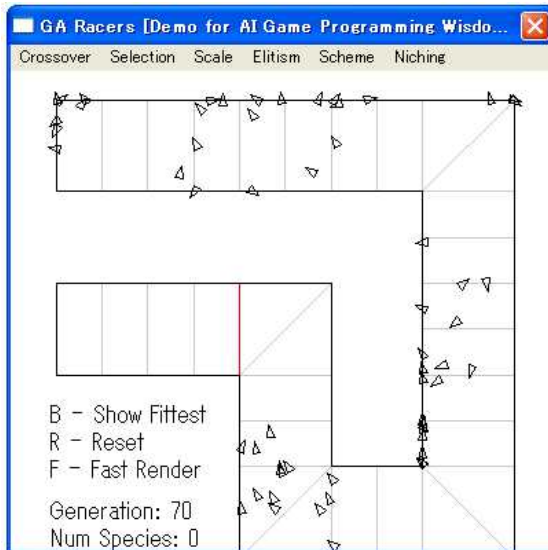
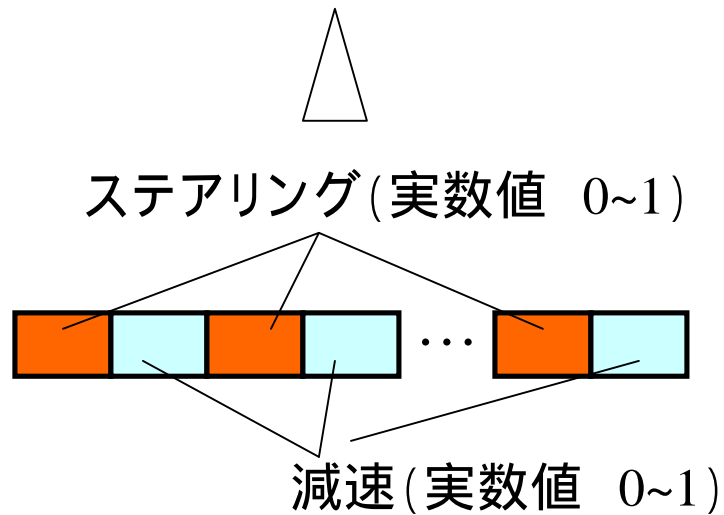
新世代(100~世代後)



だんだんと遠くまで、
たどりつけるようにする。

Mat Buckland, "Building Better Genetic Algorithm", 11.4., AI Game Programming Wisdom 2
(CD-ROMにソースコードと実行ファイルがあります)

全体の流れ



初期の個体集合を生成

個体集合

シミュレーション

評価

どれだけ遠くまで行けたか？

適応度による選択

交叉による次世代生成

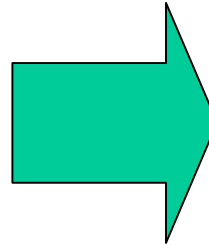
遺伝子操作 (突然変異)

(例) アステロイド

最初の世代



最初、船()は隕石()に
ぶつかってばかり...



新世代(4世代後)



次第に隕石を避けることが
できるようになる。

Brian Schwab, “Genetic algorithm”, Chapter.20., AI Game Engine Programming, Charles River Media (2004) (CD-ROMにソースコードと実行ファイルがあります)

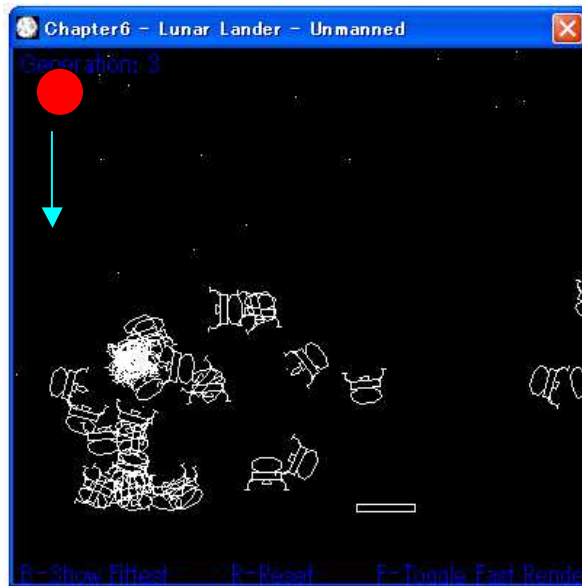
(例) ムーンランディング(月着陸)

降下ポイントから台座に垂直に着陸する操作を見出す

3 世代

20 世代

35 世代

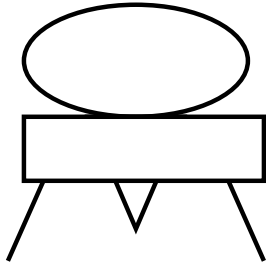


最初は全く見当違い。

だんだんと近付いて来る。

Mat Buckland, Andre Lamothe, “Moon Landings Made Easy”, chapter.6.,
AI techniques for game programming, Premier Press (2002)
(CD-ROMにソースコードと実行ファイルがあります)

全体の流れ



A

アクションタイプ

0 ... 右回転

1 ... 左回転

2 ... 原則

3 ... 何もしない

D

時間間隔

初期の個体集合を生成

個体集合

シミュレーション

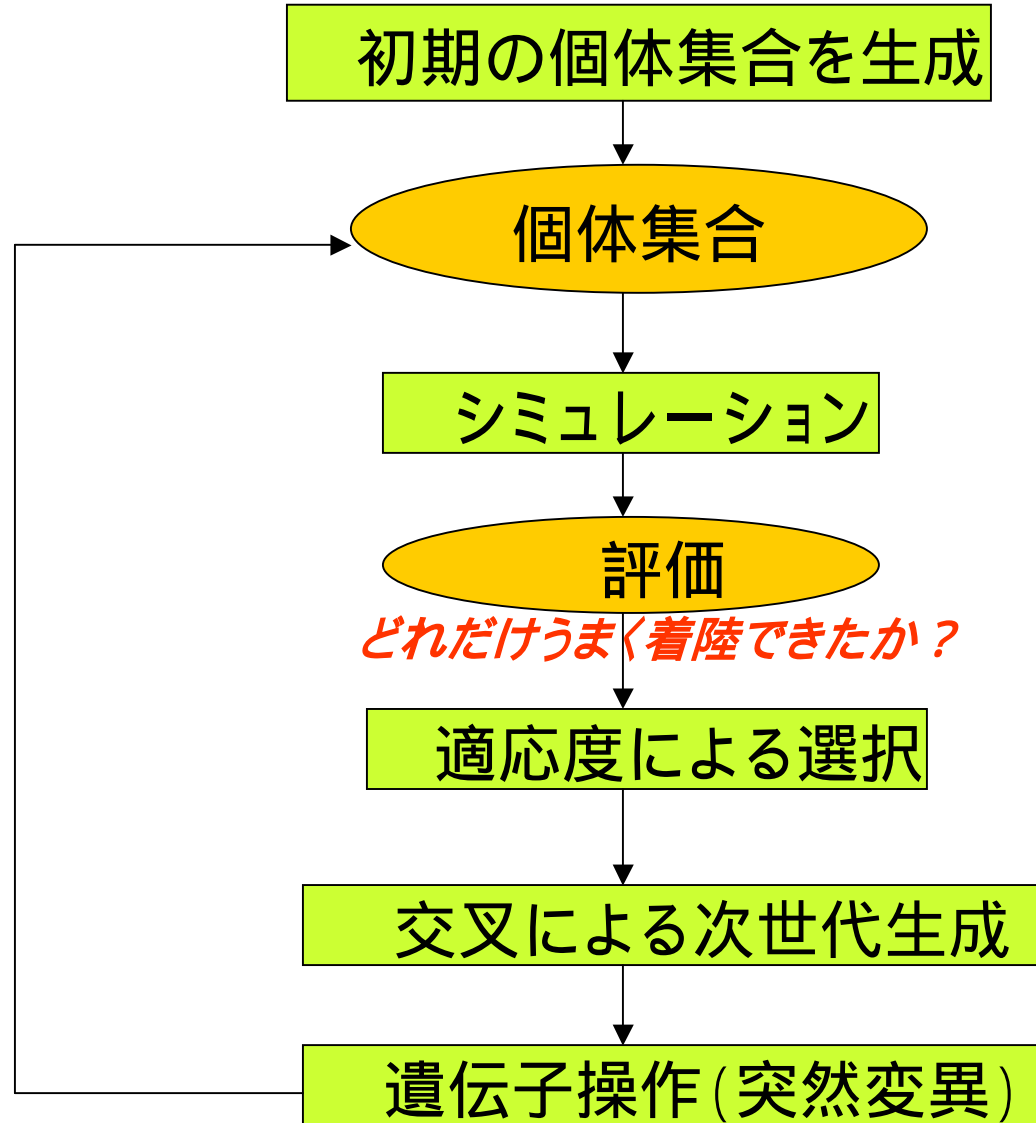
評価

どれだけうまく着陸できたか？

適応度による選択

交叉による次世代生成

遺伝子操作 (突然変異)

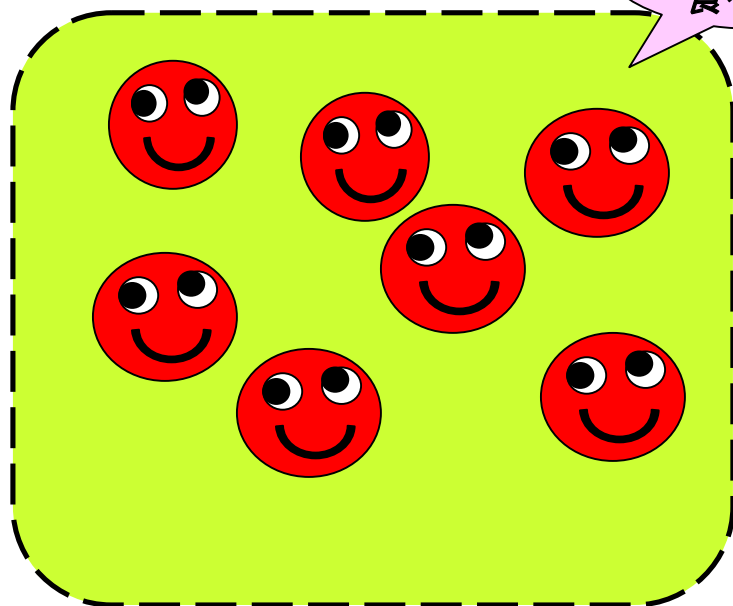




(例) アストロノーカ

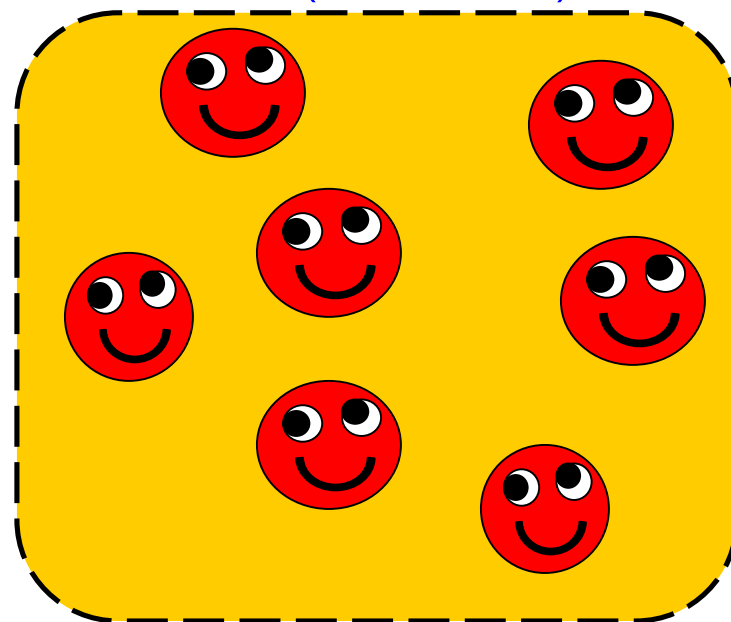
最初の世代

野菜
食べたい



最初はすぐに罠にかかるけど

新世代(5～世代後)



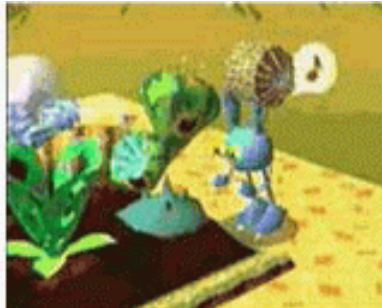
だんだんと罠にかからないようになる

MuuMuu, プレイステーション用ソフト「アストロノーカ」(Enix, 1998)

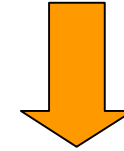
<http://www.muumu.com/games/astro/>

どういうゲーム？

珍しい野菜を育てる



しかしバブーが野菜を食べに来る



トラップを仕掛けて野菜を守れ！



高値で取引、そして野菜コンテストで優勝！



MuuMuu, プレイステーション用ソフト「アストロノーカ」(Enix, 1998)

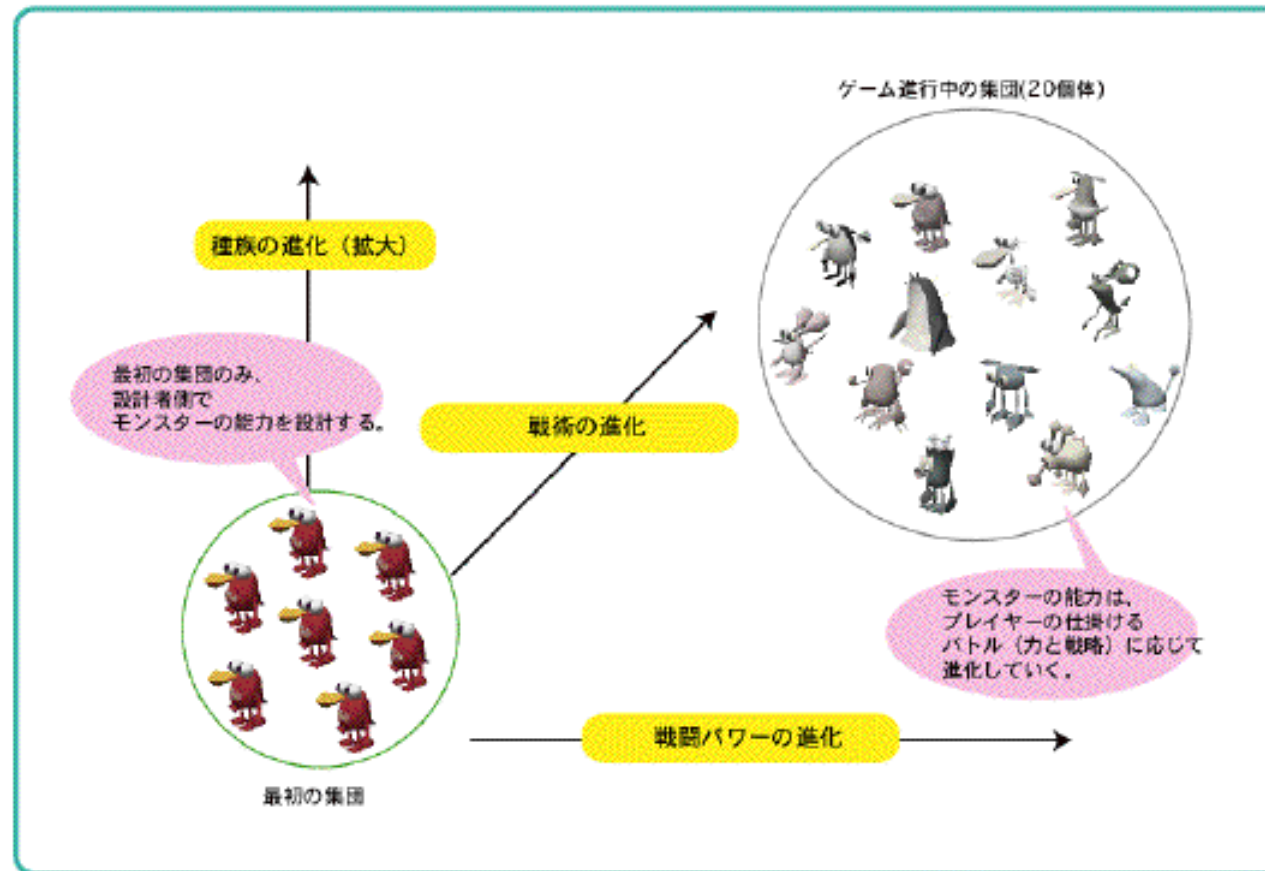
<http://www.muumu.com/games/astro/>

以下の解説は

森川幸人,
「テレビゲームへの人工知能技術の利用」,
人工知能学会誌vol.14 No.2 1999-3
<http://www.1101.com/morikawa/1999-04-10.html>

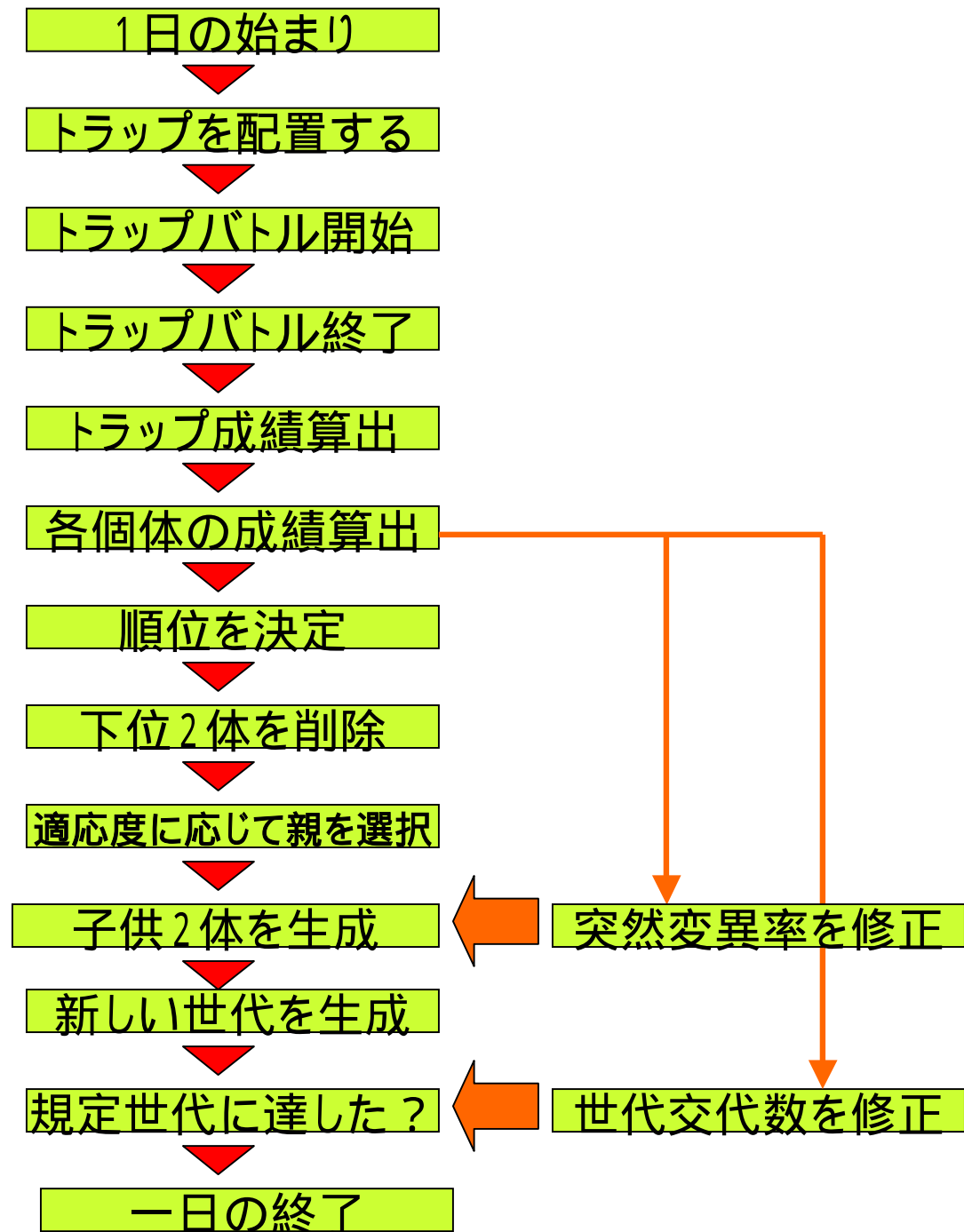
に準拠します。

全体の流れ



森川幸人, 赤尾容子, 「アリの知恵はゲームを救えるか?」, CEDEC2003
http://www.muumu.com/CEDEC2003_ants/CEDEC2003_ants.htm

全体の流れ



4 - 初期の個体集合を生成

個体を多数 (GAにはある程度の母数が必要) 用意し、各NPCに遺伝子コードを設定し、初期値を設定する。

[バブーの属性 (総計 56)]

体重	身長	腕力	脚力	耐性_かかし	耐性_快光線
----	----	----	----	--------	--------

[各ビットの重み]

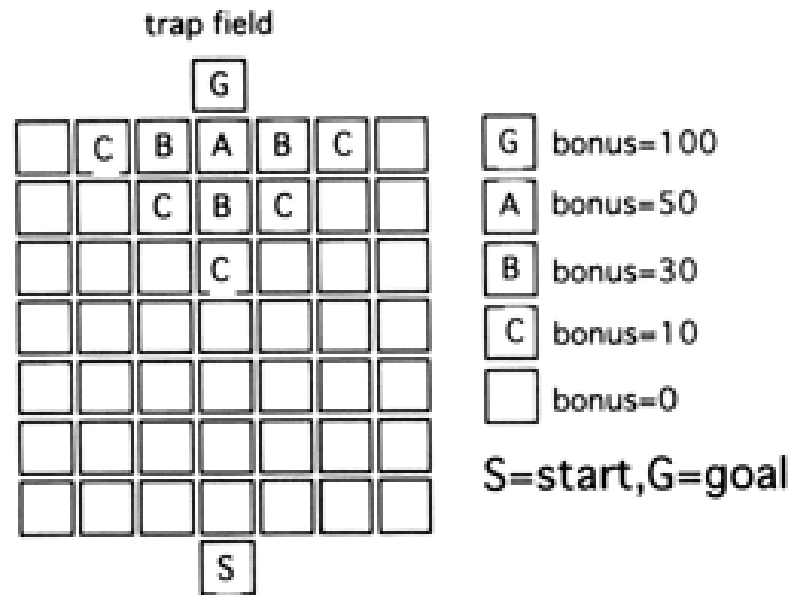
1.87	6.85	16.25	25.03	25.03	16.25	6.85	1.87
------	------	-------	-------	-------	-------	------	------

0 1 2 3 4 5 6 7

$$56 \times 8 = 448 \text{ ビット}$$

4 - シミュレーションとNPCの評価

トラップを奥へと通り抜けることができるほど、評価点が高くなる。



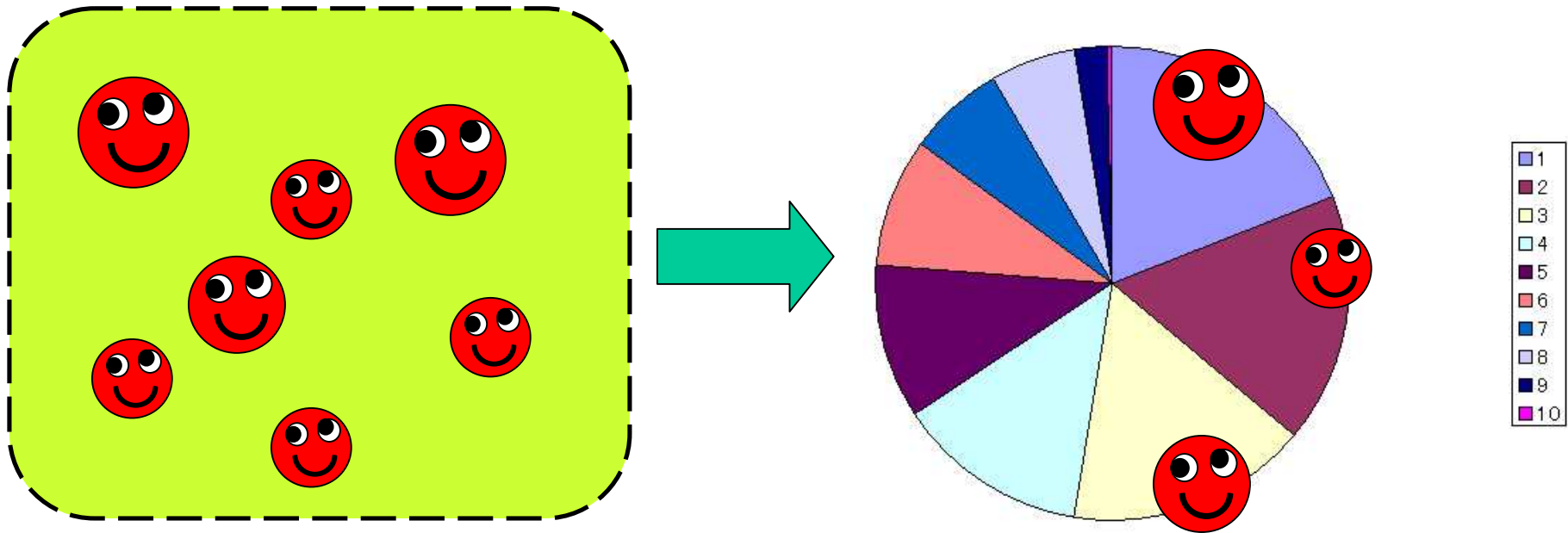
適応度 = 成績 + T B時間*0.3 + エンジョイ*0.5 + トラップ点 + 安全点 + H P*0.5

要した時間

トラップに対する耐性

4 - 選択

生き延びて子孫(offspring)を残せる個体を決定する

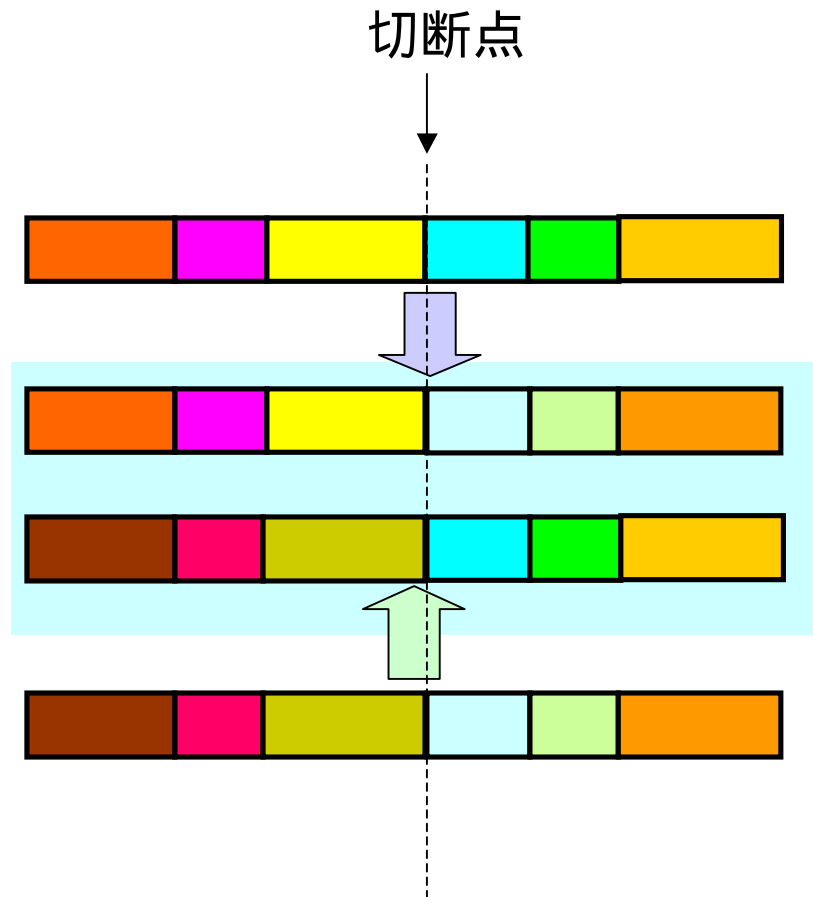


適応度比例方式(ルーレット選択)

... 適応度の大きさに比例した確率で生き延びて親になれる。

4 - 交叉による次世代生成

選んだ2つの親の遺伝子を交叉(crossover)させる。



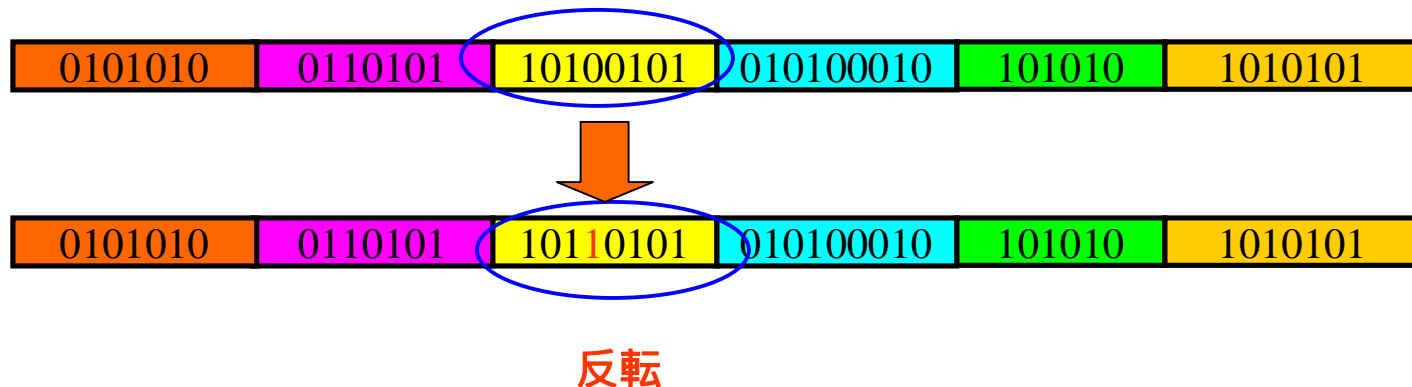
(詳しい交叉の情報はわかりません)

4 - 遺伝子操作 (突然変異)

ある確率 (突然変異率) で、遺伝子コード上の遺伝子 (内容) をランダムに対立遺伝子に書き換える。

突然変異率 3% に設定

バイナリー表現



- (1) 親の遺伝子が似ているほど (ハミング距離が小さい) 突然変異しやすい。
- (2) 突然変異が起こる場所は、トラップの置き方によってある程度限定される。

(解説) ハミング距離とは？

二つの記号列の間の異なる要素の数

1	0	1	0	1	0	0	0	0	1
1	0	1	0	0	0	1	0	1	0

4箇所違う = 距離 4

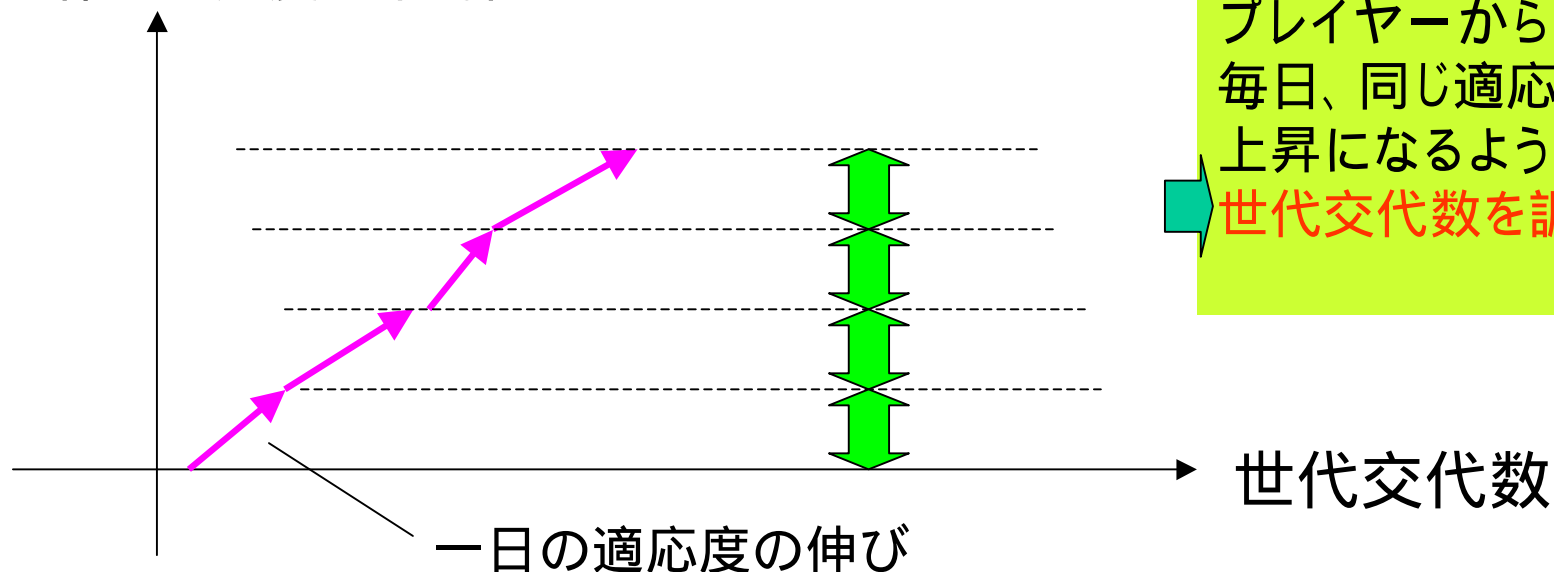
距離が小さい = 遺伝子が似ている
距離が大きい = 遺伝子が異なる

ゲームデザインにおける工夫

工夫その 遺伝的アルゴリズムは集団に対するアルゴリズム
→ 一体のトラップバトルの裏で他の20体も同じトラップバトルをして、
全体として世代交代をさせている。

工夫その 遺伝的アルゴリズムは進化のスピードがプレイヤーに体感させるには遅い
→ プレイヤーには「1世代の変化」と言っているが、
実はだいたい1日5世代分進化させている。

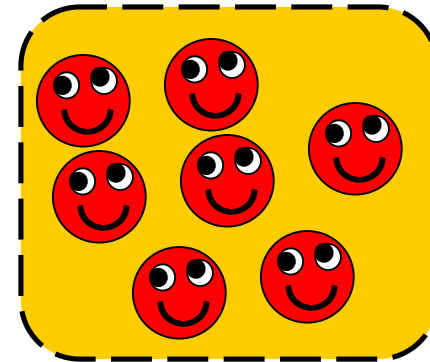
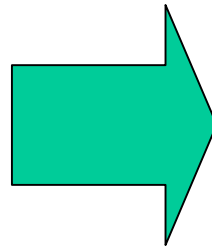
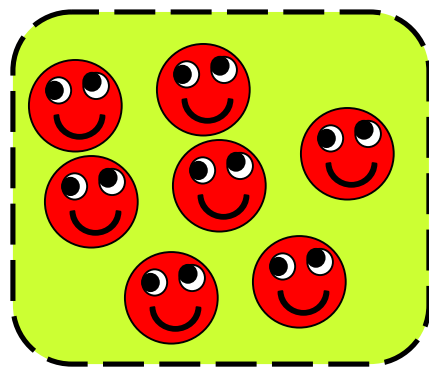
全体の適応度の平均値



工夫その プレイヤーから見て
毎日、同じ適応度の
上昇になるように、
→ 世代交代数を調整している

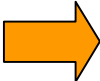

第2章 遺伝的アルゴリズム

1. 遺伝的アルゴリズムのイメージ
2. 遺伝的アルゴリズムの原理
3. 遺伝的アルゴリズムのゲームへの応用
4. 遺伝的アルゴリズムの展望



4 . 遺伝的アルゴリズムの展望

遺伝的アルゴリズムの展望

- (1) 遺伝的アルゴリズムとは、集団の環境への適応である。
環境が明確に定義されればされるほど、アルゴリズムを適用しやすい。
 物理シミュレーション環境の中でNPCを評価してやる。
- (2) 遺伝的アルゴリズムは、既に幾つかのゲームで利用されており、
原理を理解すれば実装は難しくない。
 企画、設計、技術が揃えば導入が可能だ。
- (3) ニューラルネットと合わせた方法が、いろいろと提案されている。
(詳しくは、もう一つのニューラルネットの事前資料へ)
- (4) 多くのデモや、そのソースコードが公開されている。
デモ自体はたいしたことはないが、そこから、
ゲームにイメージへ発展させることが大切である。
- (5) 遺伝的アルゴリズムは、ゲームにおける生物の法則を導入することの
効果を教えてくれる。