



IMAGIRE DAY

ゲーム開発のための プロシージャル技術の応用

三宅 陽一郎

y_miyake@fromsoftware.co.jp

9/9/2008

FROM SOFTWARE

本講演は、
AI, プロシージャル分野を扱います。

そこで、CEDEC2008における
この分野のスケジュールを確認しておきましょう。



AI, プロシージャル分野スケジュール

9.9.(火)

<p>IMAGIRE DAY</p> <p>三宅 陽一郎</p> <p>ゲーム開発のための プロシージャル技術の応用</p>
<p>IMAGIRE DAY</p> <p>今給黎 隆</p> <p>プロシージャルグラフィックス - 理論と実践</p>

9.10.(水)

<p>AI DAY</p> <p>大野 功二</p> <p>ニューラルネットワークと ゲームAI</p>
<p>AI DAY</p> <p>芝村 裕吏</p> <p>ゲーム開発のための プロシージャル技術の応用</p>
<p>AI DAY</p> <p>森川 幸人</p> <p>ゲームとAIは ホントに相性がいいのか？</p>
<p>AI DAY</p> <p>大橋 晴行 高橋 義之 鎌田 浩平</p> <p>魔ごころを、きみに ~「勇者のくせになまいきた。」 が目指した自己組織化アルゴリズム~</p>

9.11.(木)

<p>CEDECラボ AI</p> <p>伊藤 毅志 西野 順二</p> <p>ゲームとAI (1) ロボカップサッカーにおけるAI</p>
<p>CEDECラボ AI</p> <p>保木 邦仁</p> <p>ゲームとAI (2) 将棋におけるAI (ボナンザ)</p>
<p>CEDECラボ プロシージャル</p> <p>宮田 一乗</p> <p>プロシージャル技術の動向</p>
<p>AI DAY</p> <p>長久 勝</p> <p>ゲームAIを再び語る</p>

IMAGIRE DAY

三宅 陽一郎

ゲーム開発のための
プロシージャル技術の応用

IMAGIRE DAY

今給黎 隆

プロシージャルグラフィックス
- 理論と実践

CEDECラボ プロシージャル

宮田 一乗

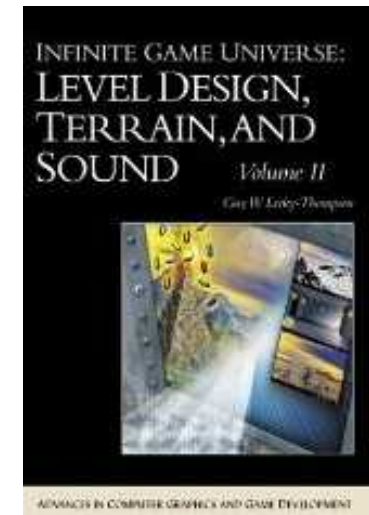
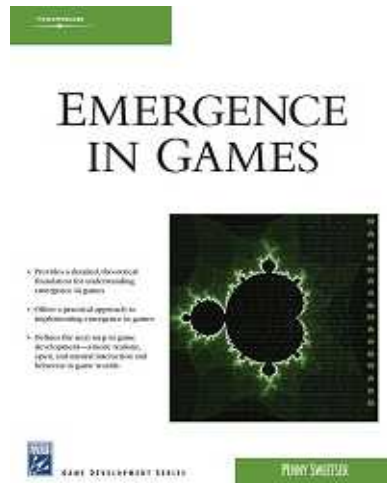
プロシージャル技術の動向

3つの資料を合わせてご覧ください。

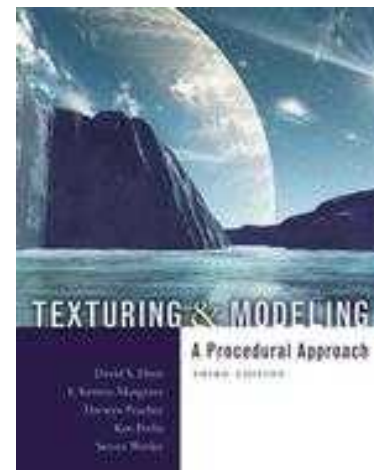
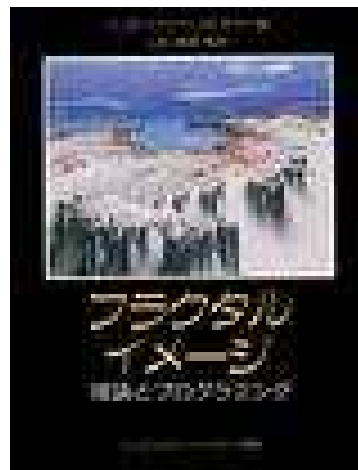
プロシージャルな世界が見えて来ます！

プログラミング・テキスト

最近



古典



フラクタルイメージ 理論とプログラミング (ハードカバー)
ハインツ・オットー パイトゲン(編集), ディートマー ザウペ(編集),
山口 昌哉(翻訳)シュプリンガー・フェアラク東京 (1990/08)

References

本講演は

「プロシージャル技術から、新しく可能なゲームの形を捉える」ための5講演

- (1) IGDA日本 ゲームAI連続セミナー第5回資料 (アストロノカ)
<http://www.igda.jp/modules/mydownloads/visit.php?cid=2&lid=69>
- (2) IGDA日本 ゲームAI連続セミナー第6回資料 (広範なプロシージャル技術の紹介)
<http://www.igda.jp/modules/mydownloads/visit.php?cid=2&lid=70>
- (3) IGDA日本 GDC2008報告会資料 (Far Cry 2)
<http://www.igda.jp/modules/mydownloads/visit.php?cid=2&lid=74>
- (4) DCAJ 2007年度 報告書 (広範なプロシージャル技術の教科書、立命館大学で講演)
「デジタルコンテンツ制作の先端技術応用に関する調査研究報告書」
http://www.dcaj.org/report/2007/ix1_07.html
- (5) DiGRA Japan 2008年6月公開講座 資料 (SimCity, The Sims, Spore)
<http://www.igda.jp/modules/mydownloads/visit.php?cid=2&lid=77>

を素材として、より本格的に、
ゲーム開発におけるプロシージャルの理論
を展開したものです。

目次

第1部 プロシージャル技術とは何か？ (15分)

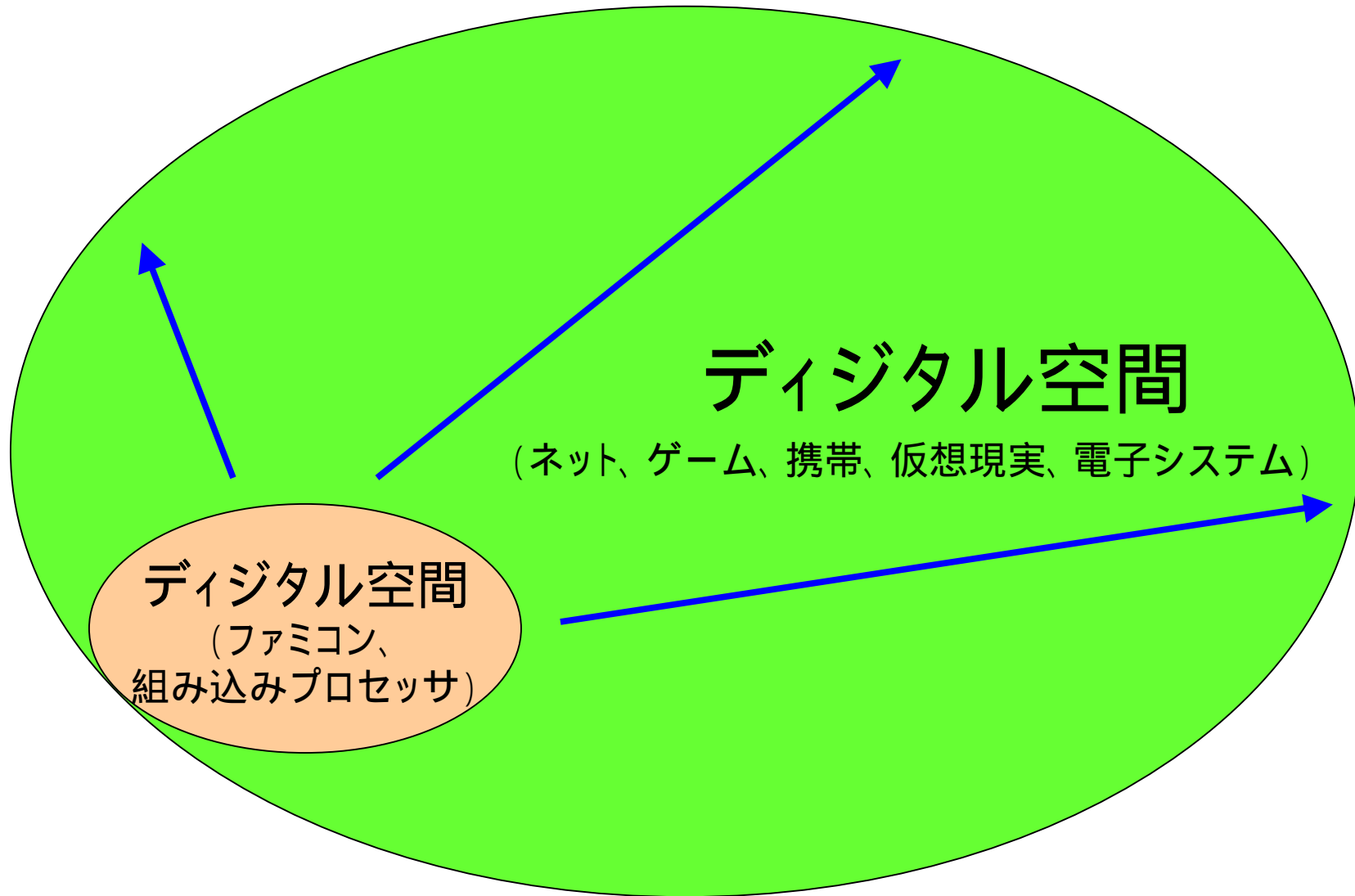
第2部 ゲーム開発のためのプロシージャル技術
(45分)

第3部 まとめ (20分)

第1部

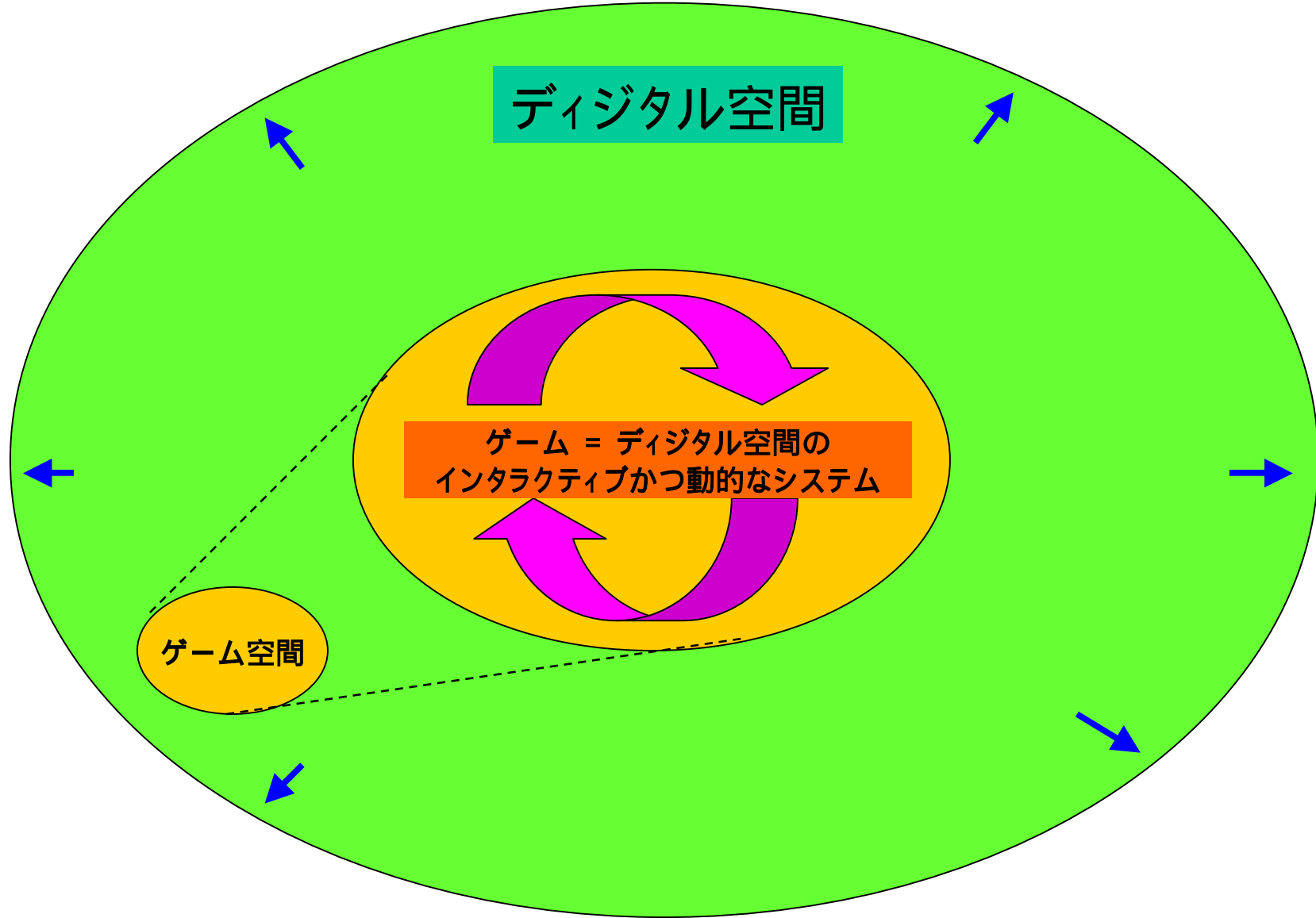
プロシージャルとは何か？

デジタル空間とゲーム



この30年で我々が触れることが出来るデジタル空間は格段に増大した

デジタル空間とゲーム



ゲーム空間も格段に増大した

デジタル空間とゲーム

ツール・プログラマー



デジタル空間

ツール・プログラム

- 3Dツール
- 2Dツール
- スクリプト言語
- サウンドツール
- エクセルなど

- 3Dモデル
- 2Dデータ
- スクリプト
- サウンド
- ゲーム設定データ

ゲーム
プログラム



開発者

ツール・チェイン
(ツール間の連携)

コンテンツ・データ
(データ間の連携)

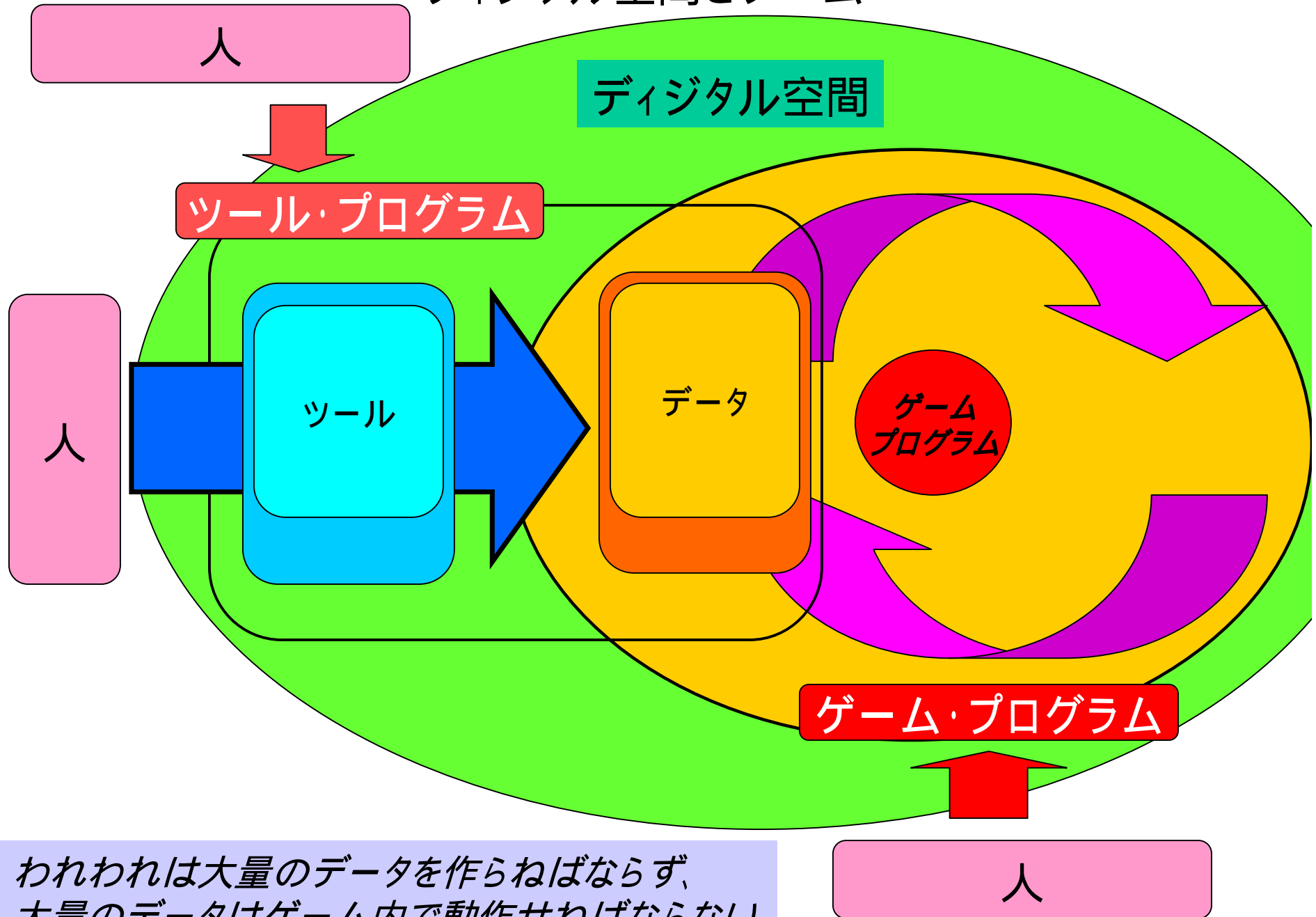
コンテンツ・パイプライン

ゲーム・プログラム



ゲーム・プログラマー

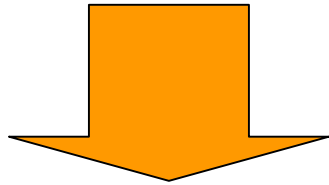
デジタル空間とゲーム



われわれは大量のデータを作らねばならず、
大量のデータはゲーム内で動作せねばならない。

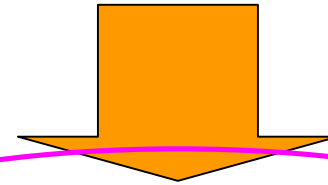
拡大するゲーム開発に対してゲーム開発者は何が欲しいか？

ゲームデータを生成・発展・消滅させる力
ゲームを動作させる力



人の手でデータを生成する
人の手で動作を定義する

マンパワー



計算によってデータを生成する
計算によって動作を定義する

プロシージャル

どちらかを選ぶかはゲームの主に規模による = 次世代機からは需要が急増

導入コスト: 人件費(コンスタント)

問題点 : 開発の規模に比例して
人手が必要
巨額の開発費があれば...

導入コスト: ソフトウェア・システムを
整備するまでの投資(2~3年)

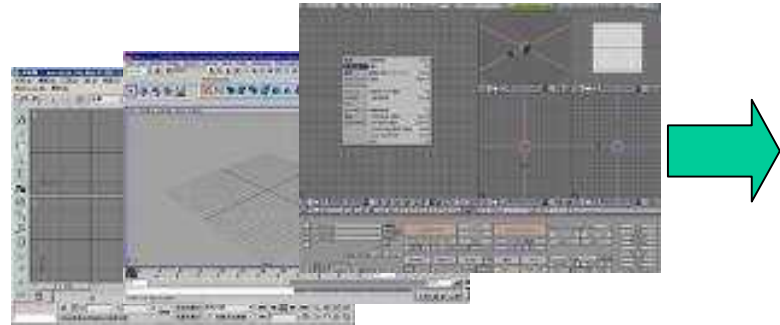
問題点 : 高度な技術レベル(簡単なものもある)
技術を使いこなすノウハウの必要性
中途半端ではケガをする

プロシージャル・コンテンツ・ジェネレーション

デジタル空間で草原を作りなさい

アプローチ1

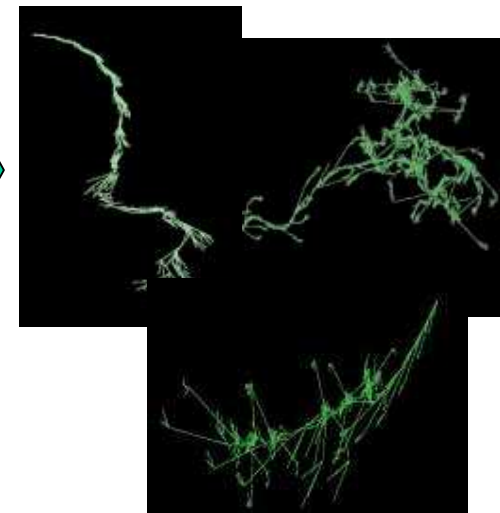
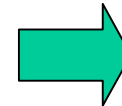
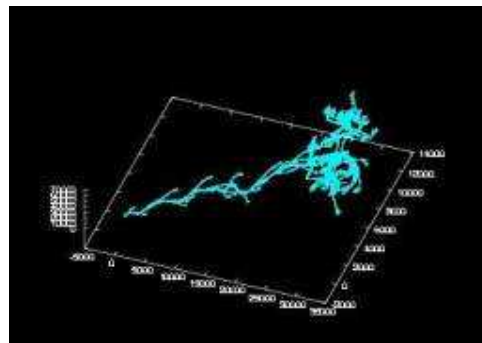
人の手によって
モデリングを行う



アプローチ2

プログラムによって
モデルを自動生成する

アルゴリズム



2003, Youichiro Miyake

プロシージャル

```
void gen(){  
    if(unit_length > MAX_LENGTH) { add_stem(); return; }  
    add_rings(current_stem);  
  
    if(steps > 1000);return;  
    gen();  
}  
  
int make(){ gen(); return; }
```

反復関数法

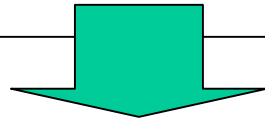
プロシージャル技術 → ゲームデザインへ

たくさんの演算によってモデルを生成する

コンテンツ自動生成 (Procedural Contents Generation)

プロシージャルとは？

プロシージャル(Procedural)
= 計算による、連続した操作による



デジタルゲームでは？

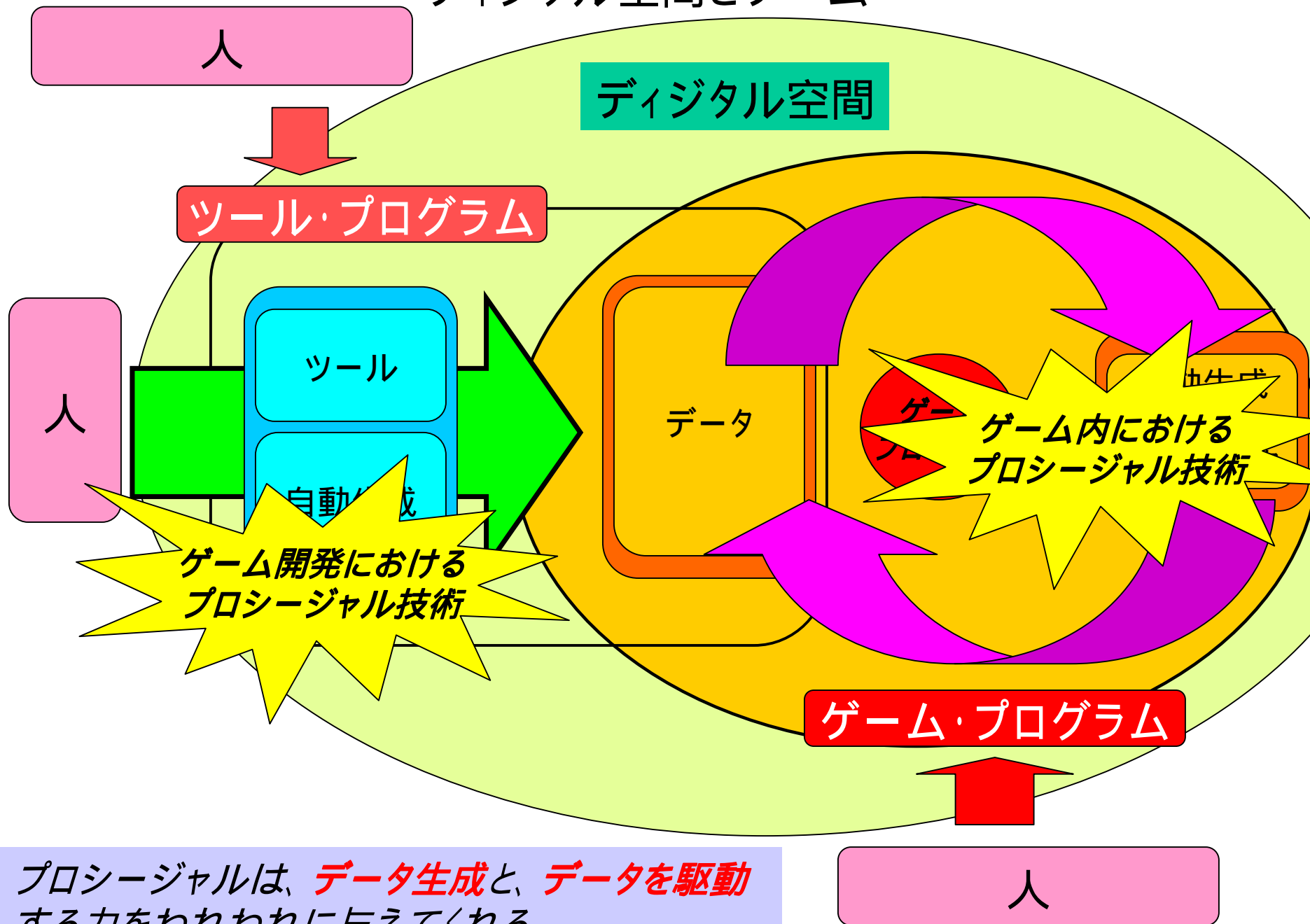
ゲーム空間、デジタル空間において、
自身の連続操作によって自律的な力を持つファクター

(例) 自動生成、自律的なAI

プロシージャル・コンテンツ・ジェネレーション
(Procedural Contents Generation = PCG)

プロシージャル制御
(Procedural AI)

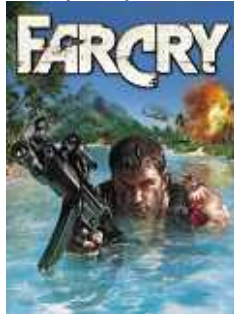
デジタル空間とゲーム



プロシージャルは、**データ生成**と、**データを駆動**する力をわれわれに与えてくれる。

プロシージャルなゲームエンジン

Far Cry(Crytek & Ubisoft)



CryEngine 1.0



Far Cry Instincts

Crysis(Crytek & EA)



CryEngine 2.0

Far Cry 2(Ubisoft Montreal & Ubisoft)



Dunia Engine



Frostbite Engine

次世代Battlefield用エンジン

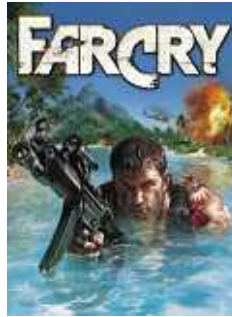


次世代Battlefieldシリーズ用エンジンFrostbite (4gamers)

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

プロシージャルなゲームエンジン

Far Cry(Crytek & Ubisoft)



CryEngine 1.0



*Far Cry Instincts
(Ubisoft Montreal & Ubisoft)*

Crysis(Crytek & EA)



CryEngine 2.0

Far Cry 2(Ubisoft Montreal & Ubisoft)



DuniaEngine

CryEngine 2.0

デモ



**Procedural Vegetation
Animation and Shading**

CRYTEK



Procedural Vegetation Animation in Crysis

tiago_gpuGems3_1280_mpeg

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

DUNIA ENGINE 2.0

FAR CRY 2 におけるプロシージャル・データ生成



FAR CRY 2 におけるプロシージャル・データ生成

Procedural Data Generation in FAR CRY 2

<https://www.cmpevents.com/GD08/a.asp?option=C&V=11&SessID=6240>

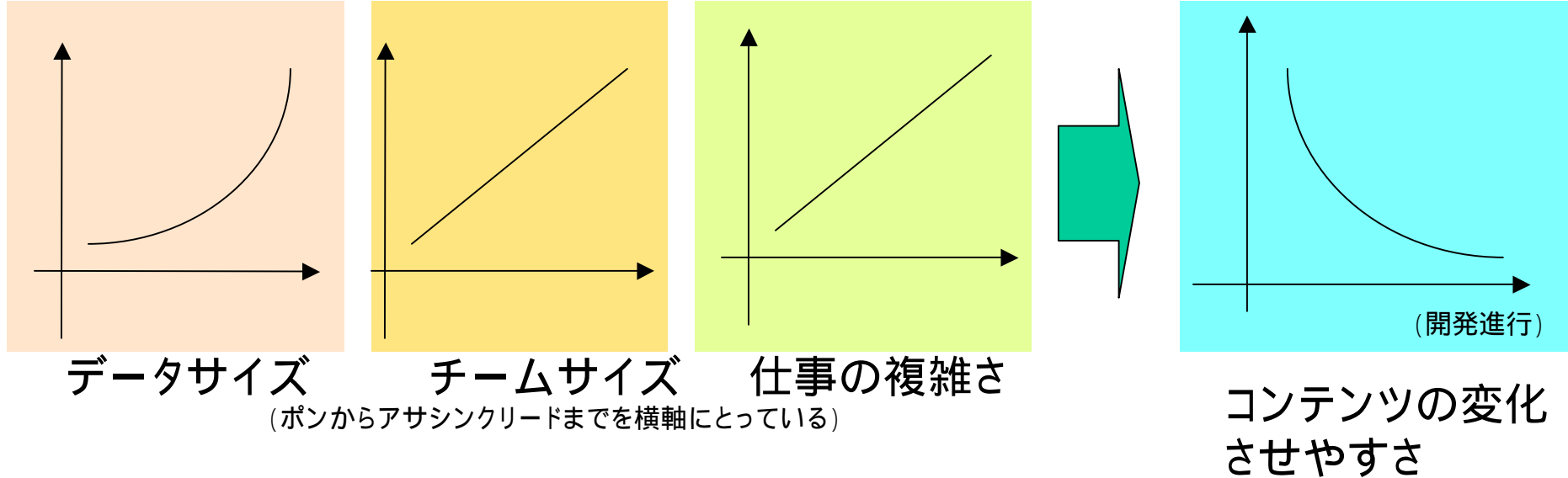
講演者: Dominic Guay (Technical Director, Ubisoft)

日時: 2月21日(木) 9時 ~ 10時

場所: North Hall, 132号室

これからのゲーム開発の問題点

Rushing into a wall (壁にぶつかる)



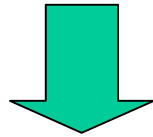
複雑に絡み合ったシステムでは、開発が進めば進むほど一部の变更で何処にバグが生成するかわからないので、データを触れなくなる。

開発が進むにつれて、ゲームコンテンツはますます変更しにくいものになる

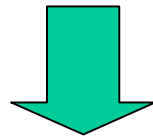
よいツールやプロセスは一つの問題の解決にはなっても、長期的なこの問題の壁を乗り越えることができない

Far Cry 2 におけるモチベーション

50km四方のマップを作る



アニメーション・データの量が膨大で作りきれない
(アニメーション・ディレクター)



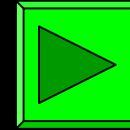
植物に関して

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

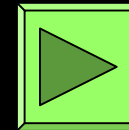
アニメーションや**破壊のされ方**は、簡単な設定をしておけば、
ゲーム内で環境に応じてシミュレートされる。

DUNIA Engine デモ

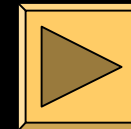
(I) Growing Vegetation



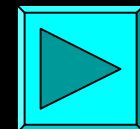
(II) Tree Generation



(III) Day and Night cycle



(IV) Realistic Weather System



[参考]データ自動生成技術を大胆に導入したFPS , 「Far Cry 2」の光と(4gamers)

<http://www.4gamer.net/games/047/G004713/20080222019/>

プロシージャルなゲームエンジン



Frostbite Engine

オブジェクト生成

(We can generate objects with procedural techniques

- Then use rules to deform / destroy / modify / move them
- Better interactivity)

Semi-procedural surface shader

Procedural shader

Procedurally distributed on the fly

Practical Example: Mountains Generation and Realistic Snow Accumulation



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

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

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

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

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

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

ハードウェア (CPU) から見たプロシージャル

ハードウェア・ベンダーからは、次なるコンピューティングの用途としてプロシージャルの方向に舵が取られている

AMD

GDC 07, 08

Intel

次世代Battlefiled用エンジン

Allegorithmic

<http://www.allegorithmic.com/>

Procedural Helps You Avoid the Resolution Problem

- Any stored texture has limited resolution.
 - If you zoom in too closely, you will see a lack of detail
 - Or even signs of the original pixels in the texture
- Procedural patterns can
 - Zooming in: Introduce noise
- Zooming out
 - A prebaked texture will still look good
 - A procedural texture can have areas without seams or artifacts
- No mapping problem
 - Don't have to worry about painful parameterization
 - Solid textures

Procedural Techniques: Now!

- Computers are fast enough so that procedural is real-time now!
 - Flexible shader models allow us to directly translate many of the offline shaders
- Direct3D10® opened new doors for procedural generation in real-time: *flexibility and power*
 - Convenience of geometry shaders and stream out
 - More flexible use of texture / buffer resources
 - Ability to directly render and filter volume textures
 - Integer and bitwise operations

Logos: GIG, frostbite, Game Developers Conference, AMD, ATI

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

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

Coherent procedural normal map

Procedural diffuse map

Complex procedural environment map

This scene is made entirely of procedural textures

Intel GDC 2008 Threading Successes 06 - Allegorithmic

http://softwarecommunity.intel.com/isn/downloads/graphics/GDC/Threading_Successes.zip

プロシージャルなゲームエンジンのポイント

ゲーム製作過程における
プロシージャル技術を用いたステージ生成

ゲーム製作技術

デジタル・ファーム(レンダリング・ファームなど)

+

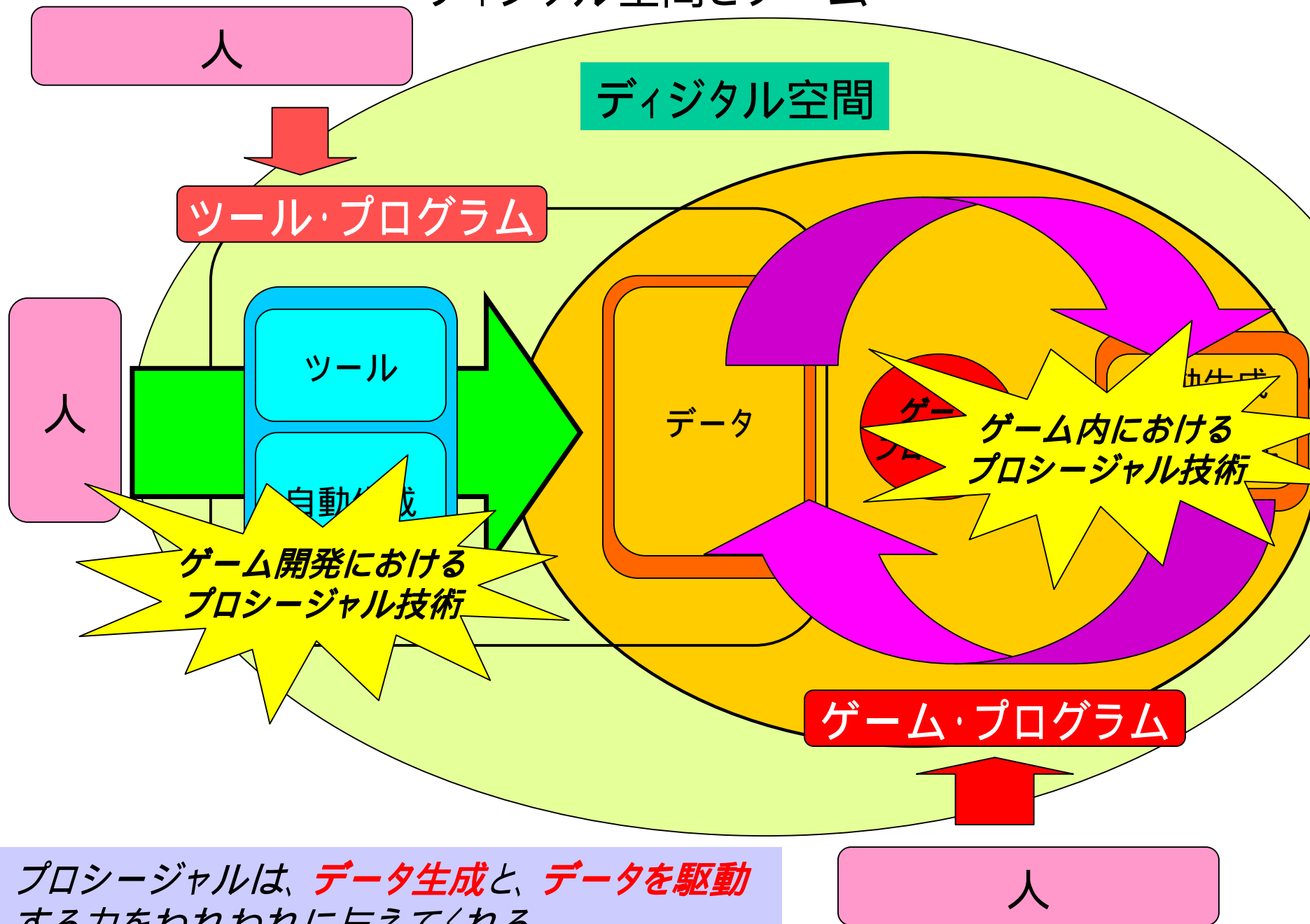
プロシージャル技術による高度で広範な
リアルなインタラクション空間の実現

ゲーム内技術

リアルタイム・コンピューティング

これから大型のゲームエンジンを設計しようという人間はプロシージャル技術を含めて考えなければならない

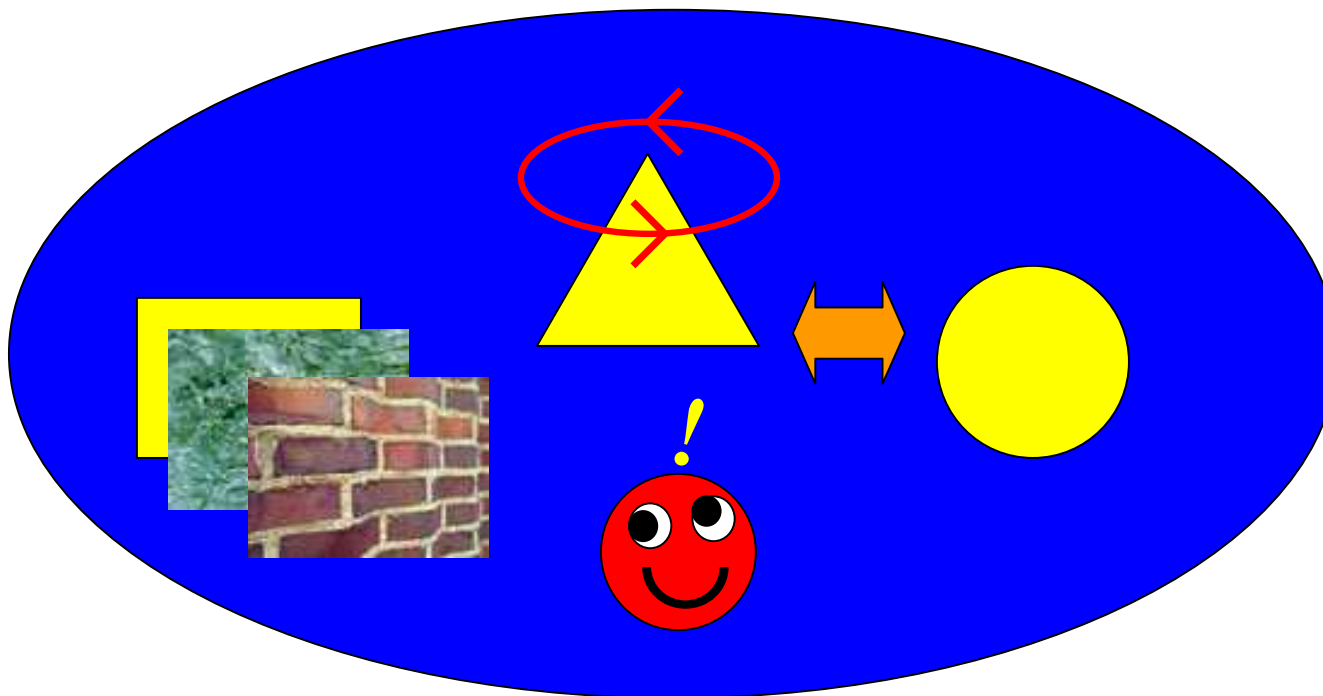
デジタル空間とゲーム



プロシージャルは、**データ生成**と、**データを駆動**する力をわれわれに与えてくれる。

プロシージャル – 固定された世界から変化する世界へ

デジタル・ワールド



固定モデル 固定テクスチャー 固定アニメーション 固定AI 固定インタラクション

プロシージャル・コンテンツ・ジェネレーション

テクスチャー**生成**

Procedural Texture

モデル**自動生成**

Deformable Model

アニメーション**生成**

Animation Synthesis

即応インタラクション

物理

認識・判断するAI

Procedural AI

街生成

City Generation

会話・ストーリー生成

Story Generation

人工市場

Artificial Societies

究極地球シミュレーター

(世界シミュレーターという意味で)

Earth Simulator



プロシージャル・コンテンツ・ジェネレーション

テクスチャー**生成**

Procedural Texture

CEDEC2008 プロシージャル講演 3日目(3コマ目)

モデル**自動生成**

Deformable Model

CEDEC2008 プロシージャル講演 1日目(4コマ目)

アニメーション**生成**

Animation Synthesis

即応インタラクション

物理

認識・判断するAI

Procedural AI

街生成

City Generation

参考資料

会話・ストーリー生成

Story Generation

人工市場

Artificial Societies

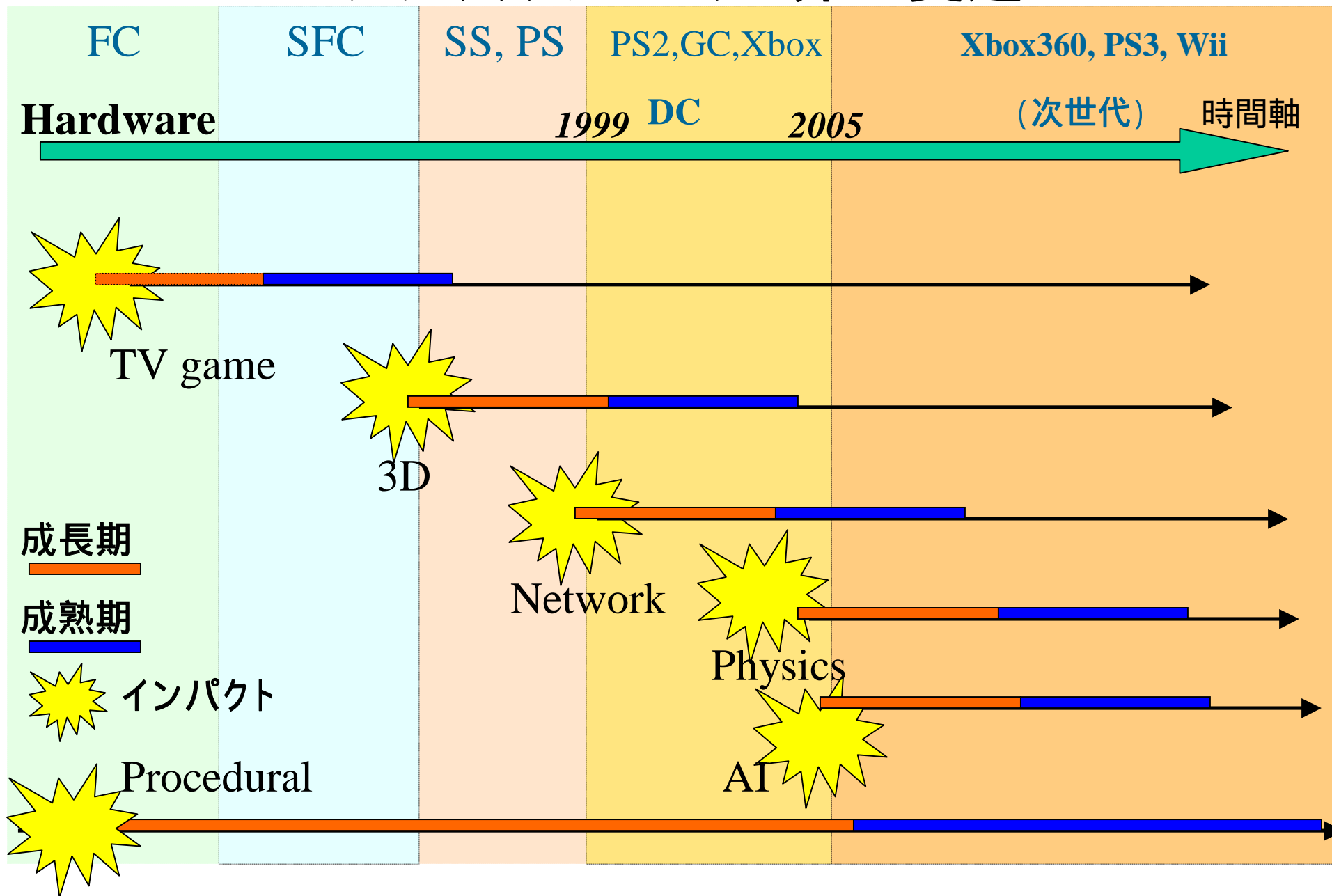
究極地球シミュレーター

(世界シミュレーターという意味で)

Earth Simulator



デジタルゲーム世界の変遷



プロシージャルの歴史: CG

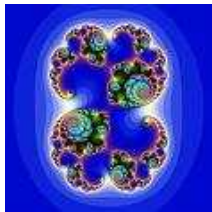
1980

1990

2000

フラクタル
幾何学

1975



フラクタル・イメージ
SIGGRAPH 1987

フラクタル幾何学を基礎とした
プロシージャルなモデル生成

地形自動生成

植物自動生成

雲自動生成

3Dモデル生成

テクスチャ自動生成

2Dモデル生成

Terragen

SpeedTree

natFX MAX

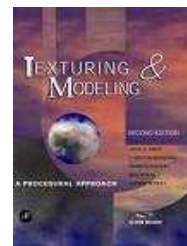
パリン・ノイズ

1997,2002

Pro FX



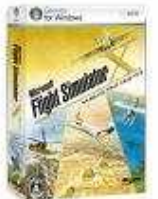
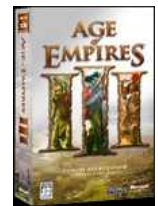
1988



1994

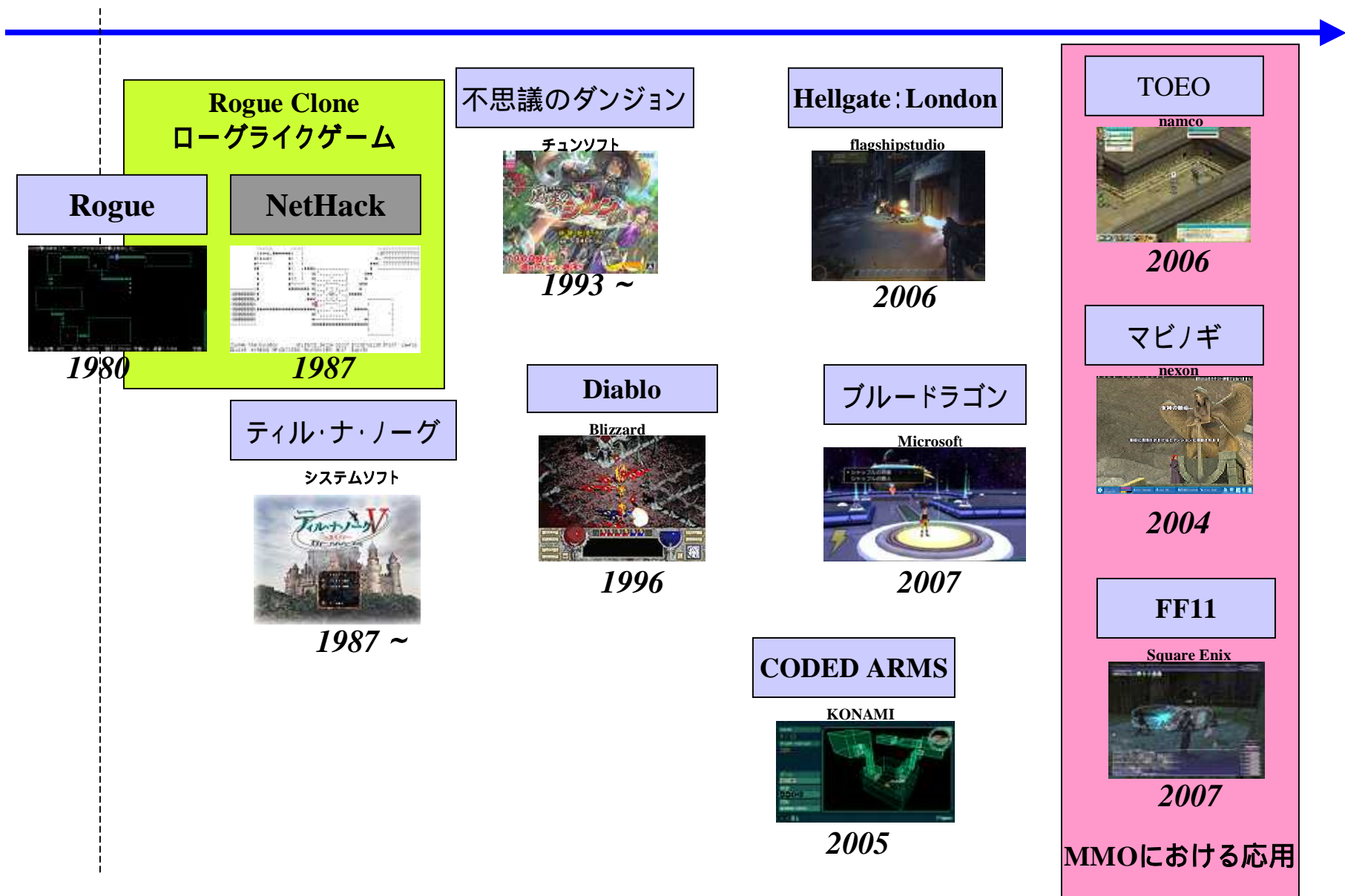


2003



プロシージャルの歴史: ダンジョン自動生成

1980



プロシージャルの歴史: 会話&自然言語

会話

Facade

Michael Mateas, Andrew Stern



2005

Virtual Avater



岡山県立大学 渡辺研究室
<http://hint.cse.oka-pu.ac.jp/>

まだゲームには
十分に応用されていない

自然言語

くまうた

MuuMuu



どこでもいっしょ

SCE



シーマン

ビバリウム



12000行のテキストライン

日本のゲームデザインの
巧み

プロシージャルの歴史: アニメーション編

1980

遺伝的アルゴリズムによる
アニメーション生成
オックスフォード大学



natural motion, <http://www.naturalmotion.com>

Indiana Jones



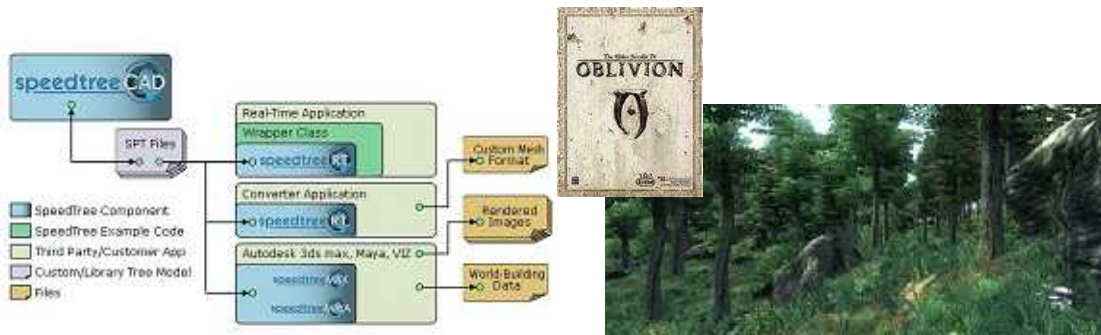
<http://ps3.ign.com/articles/705/705489p1.html>

物理的なインタラクションからアニメーションを自動生成

ワシントン大学



弾性体のアニメーション



植物のアニメーション

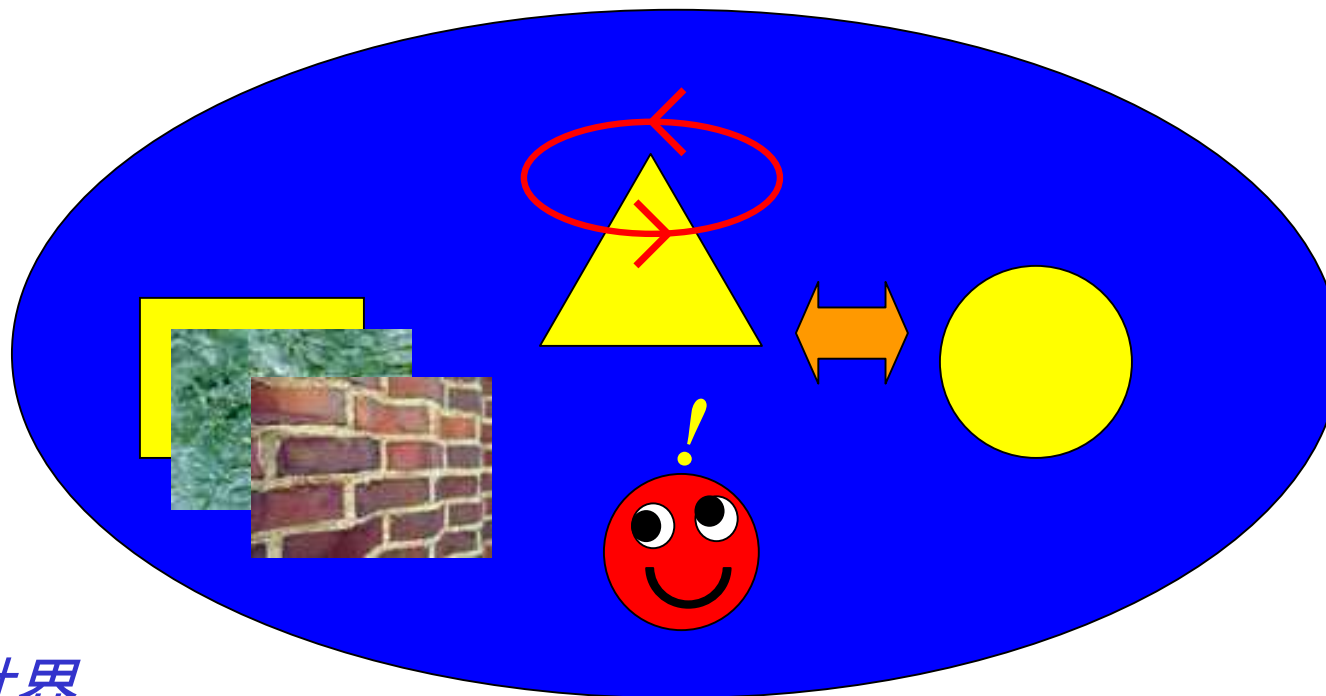
SpeedTree, <http://www.speedtree.com/>

CryEngine2



プロシージャル – 固定された世界から変化する世界へ

デジタル・ワールド



固定した世界

固定モデル 固定テクスチャー 固定アニメーション 固定AI 固定インタラクション

変形するモデル 生成するテクスチャー 生成するアニメーション 認識・判断するAI 即応インタラクション

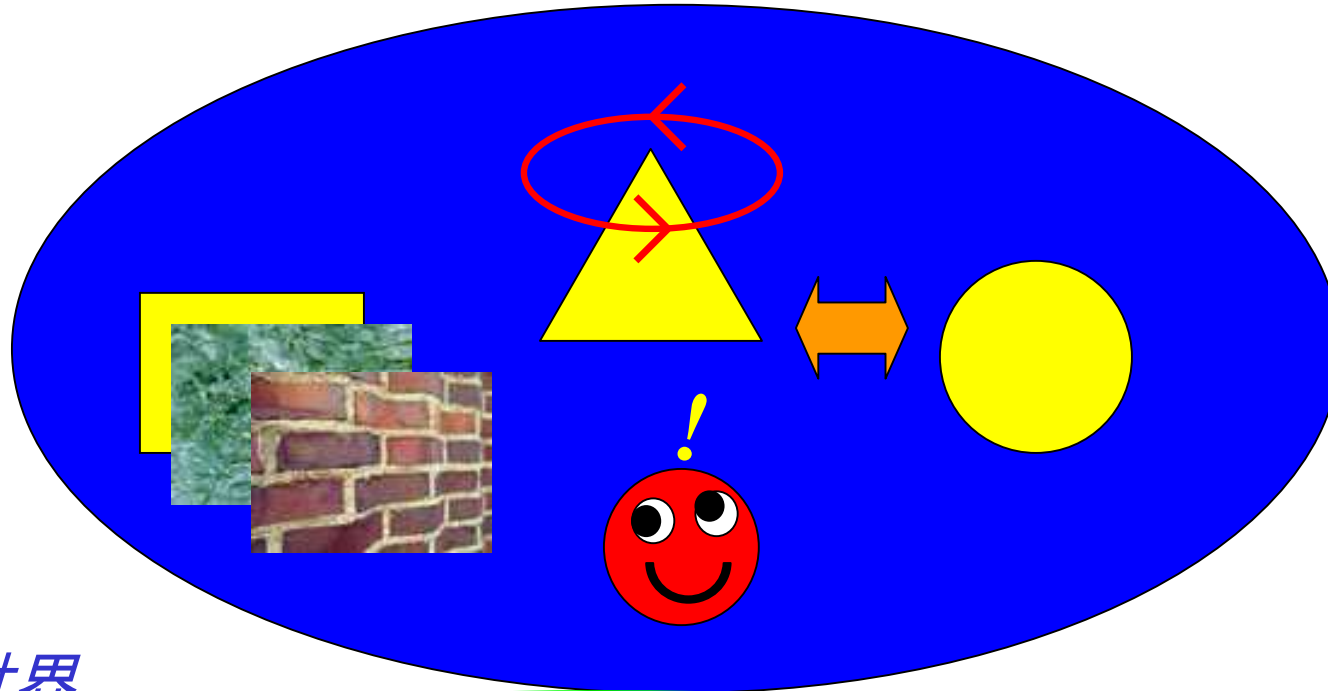
変化する世界

Procedural World

(物理)

プロシージャル – 固定された世界から変化する世界へ

デジタル・ワールド



固定した世界

固定モデル 固定テクスチャ 固定アニメーション 固定AI 固定インタラクション



変形するモデル 生成するテクスチャ アニメーション生成 認識・判断するAI 即応インタラクション

変化する世界

Procedural World (全てが計算される世界)

(物理)

プロシージャルな制御

シューティングゲームで100体の敵キャラクターを動作させなさい

アプローチ1

人の手によって
動作を定義する

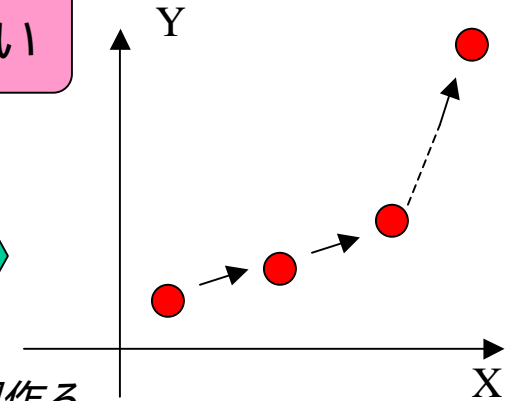
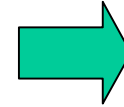
(1.0, 2.3)

(2.0, 3.1)

....

(10.0, 7.4)

...軌道の定義を100個作る



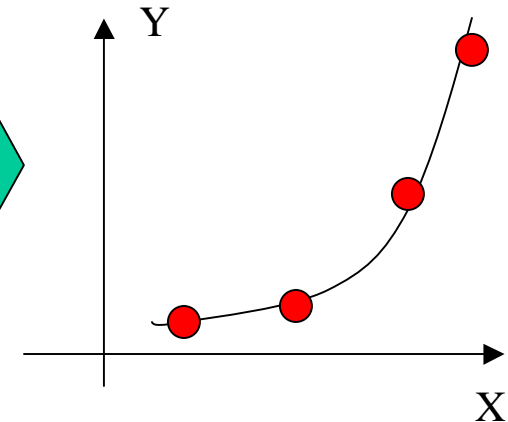
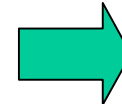
アプローチ2

アルゴリズム

数式によって
動作を生成する

$$Y = b * X * X - a * X$$

a, b のパラメータを変えて
軌道を生成



プロシージャル

プロシージャル技術 → ゲームデザインへ

たくさんの演算によって動作を生成する

プロシージャルな制御(プロシージャルなアプローチ)

プロシージャルとは？

プロシージャル(Procedural)
= 計算による、連続した操作による

デジタルゲームでは？

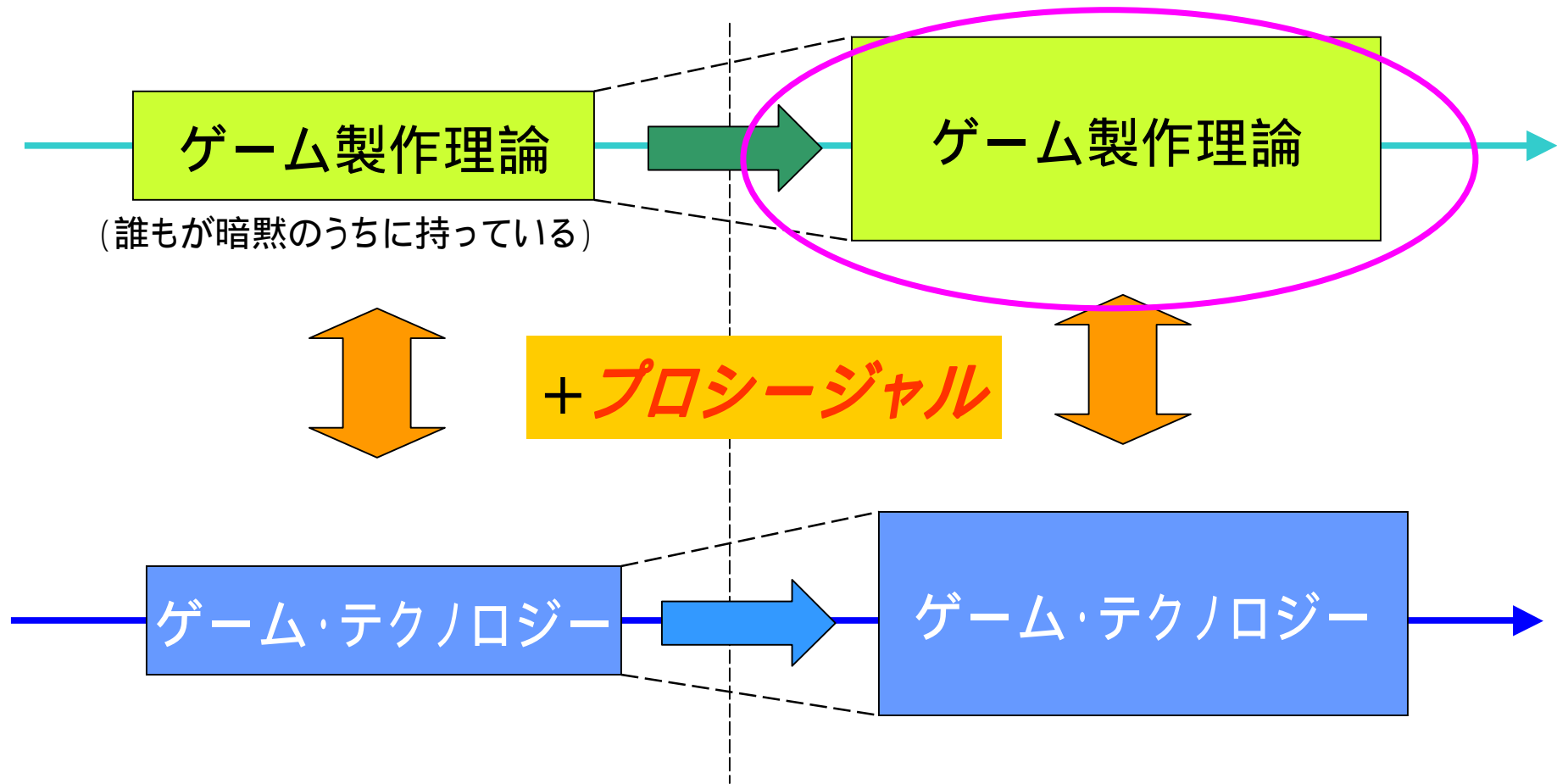
ゲーム空間、デジタル空間において、
自身の連続操作によって自律的な力を持つファクター

プロシージャル・コンテンツ・ジェネレーション
(Procedural Contents Generation = PCG)

プロシージャル制御
(Procedural AI など)

(質はともかく) プロシージャル技術は、デジタル空間で
コンテンツを自動生成、自動制御する力を与えてくれる

ゲーム開発技術と製作セオリー

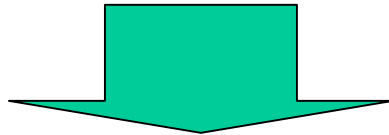


2005

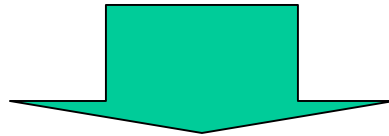
プロシージャルに対応して、新しく(各自の)ゲーム製作理論を拡張する必要がある。

第1章 まとめ

- (1) プロシージャル技術は、ゲーム開発技術、ゲーム内技術に応用できる技術である。



- (2) プロシージャル技術は、ゲーム開発とゲームデザインを根底から変化させて行く技術である。



- (3) 新しいゲーム製作のフレームを構築し、新しいゲームデザインを探求するチャンスがある。

第2部

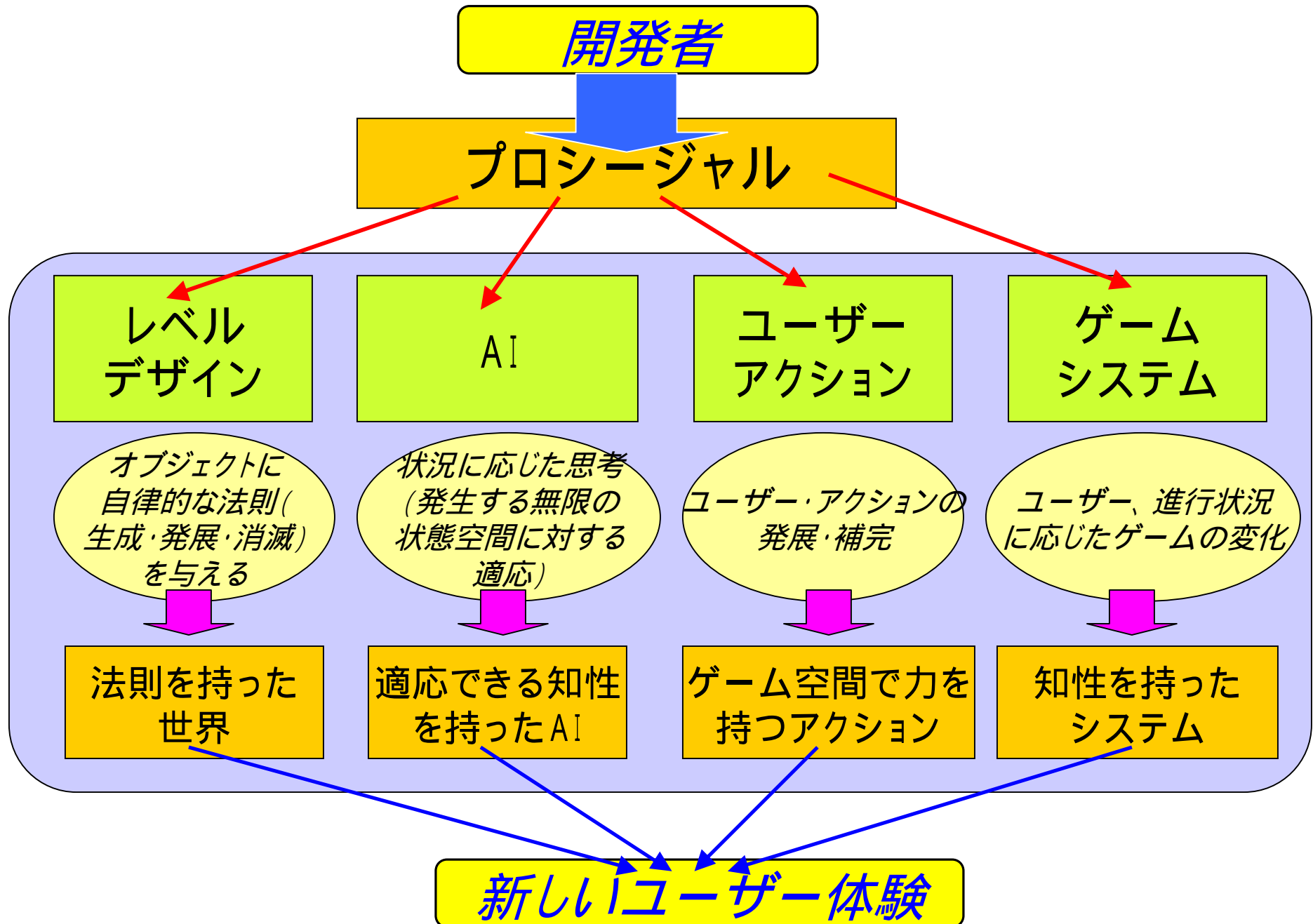
ゲーム開発のためのプロシージャル技術

ゲーム開発におけるプロシージャル技術の使い方

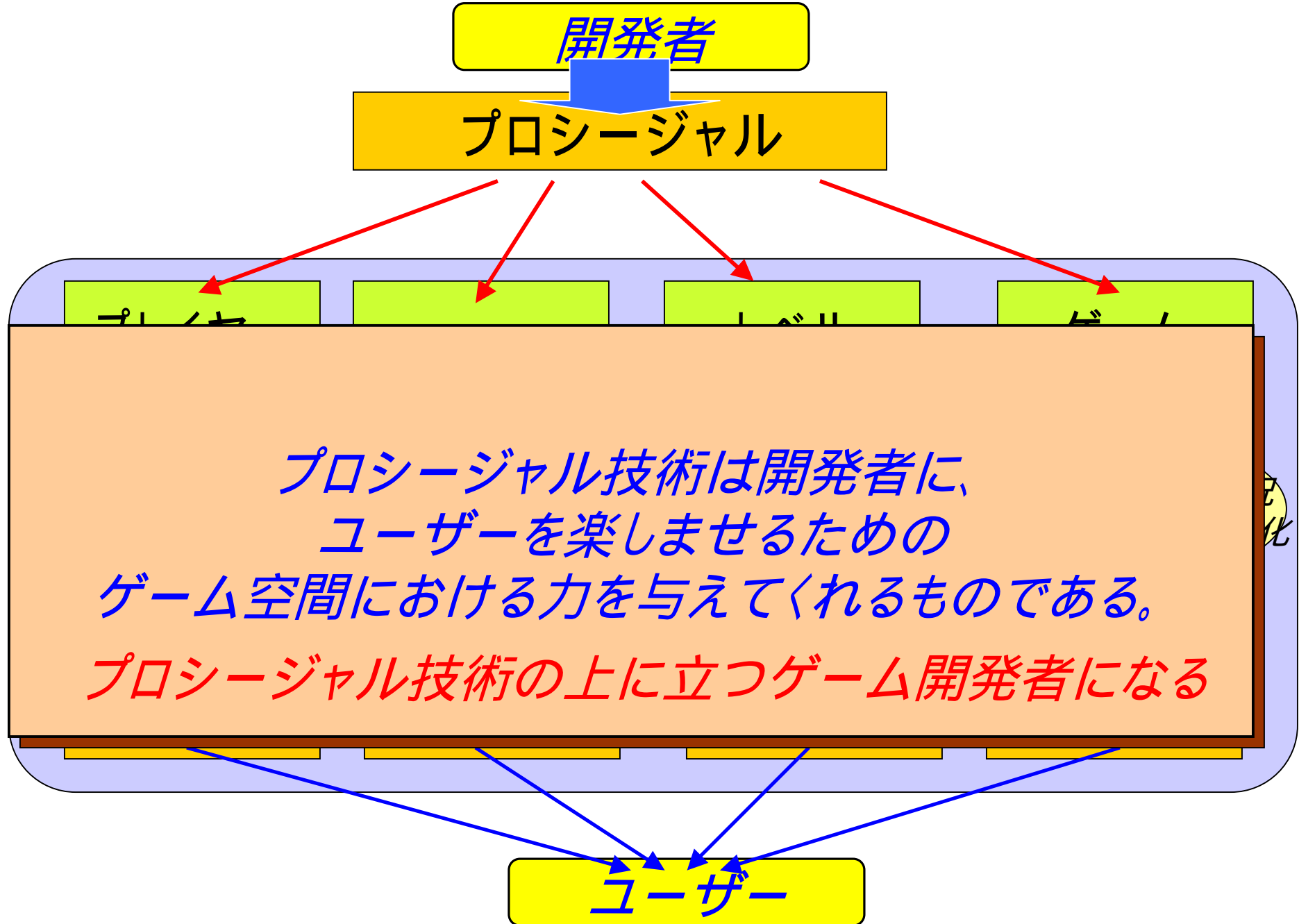
プロシージャル技術を
ゲーム開発の何処に用いるか？

プロシージャル技術の
ゲーム開発における力点は何処か？

ゲームのためのプロシージャル技術の4つのポイント



本講演の主旨「プロシージャル技術の上に立つゲーム開発者になる」



第2部

第1章

ゲーム世界(レベルデザイン)とプロシージャル

第2章

キャラクター(NPC, AI)システムとプロシージャル

第3章

ユーザー・アクションとプロシージャル

第4章

ゲームシステムとプロシージャル

第5章

まとめ

第2部

第1章

ゲーム世界(レベルデザイン)とプロシージャル

第2章

キャラクター(NPC, AI)システムとプロシージャル

第3章

ユーザー・アクションとプロシージャル

第4章

ゲームシステムとプロシージャル

第5章

まとめ

第1章

ゲーム世界(レベルデザイン)と プロシージャル



ゲームとは何か？

ルールには2種類のルールがある。

明示的なルール(explicit rule)

明文化される説明書に載っている) 意識されるルール

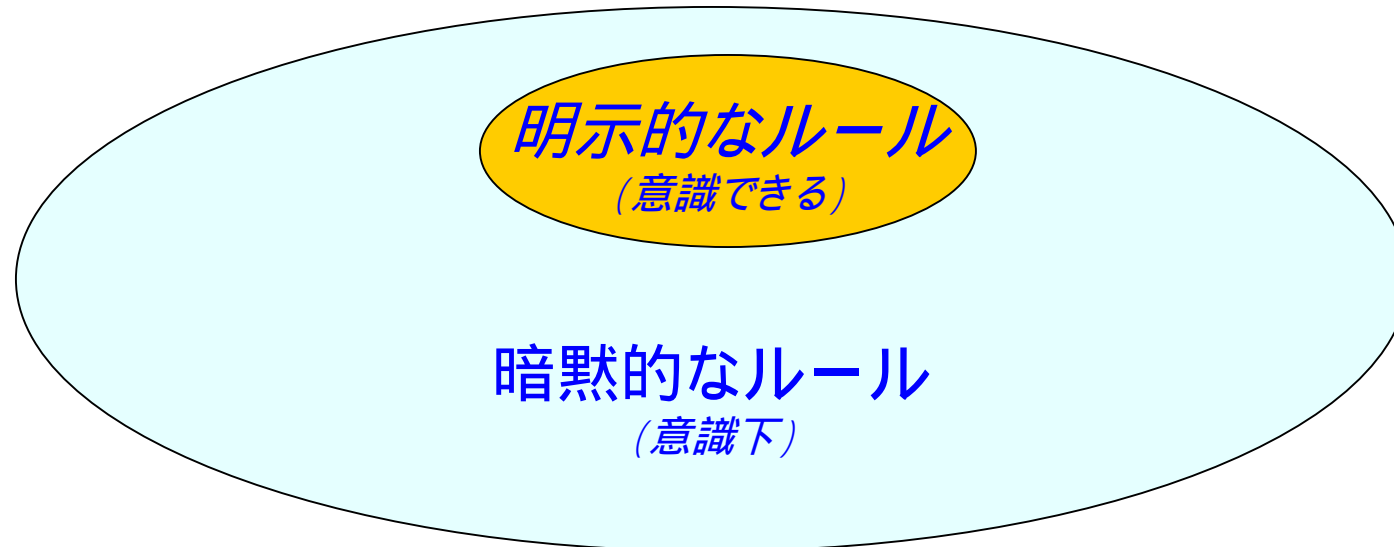
(例) 自明なルール(HPが0になったらプレイ不能)

Aボタンを押せばマリオはジャンプします。Bボタンでダッシュします。

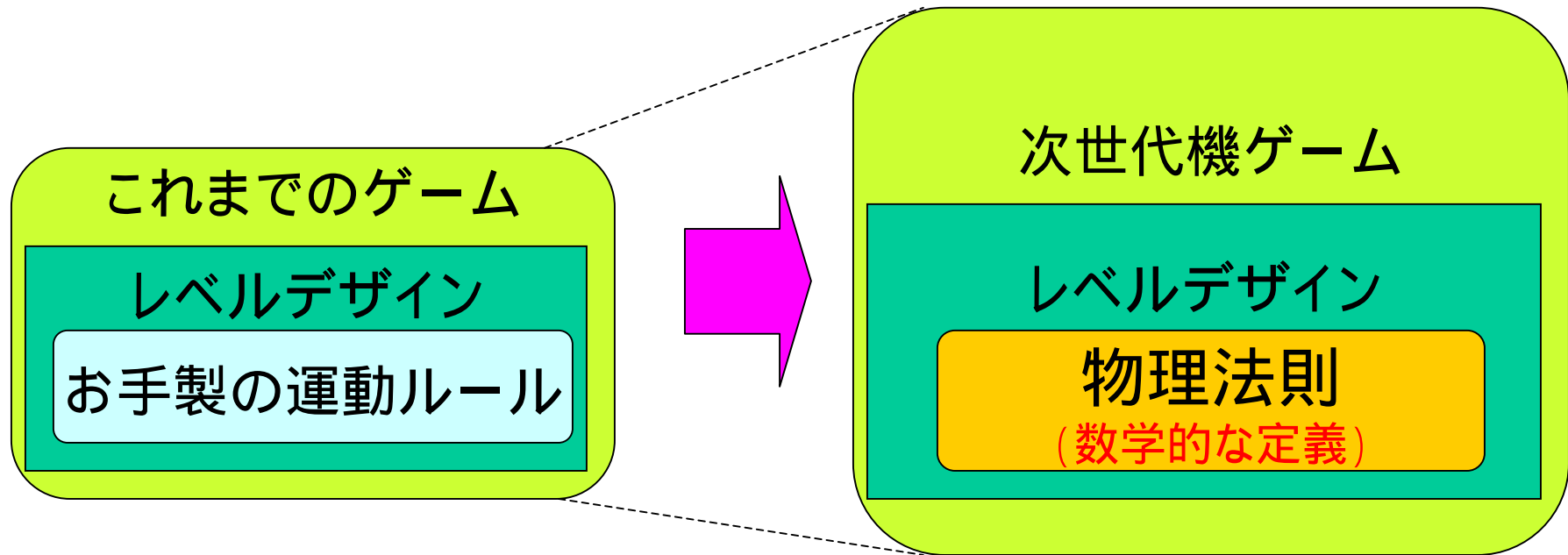
暗黙的なルール(implicit rule)

明確に意識されない背後でゲームを動かすルール

(例) 物理、AIの学習則、経済や政治の法則、弾道の落下曲線、爆発の大きさ
ジャンプしたマリオは落下します。歩けば足音がします。AIは足音に気付きます。



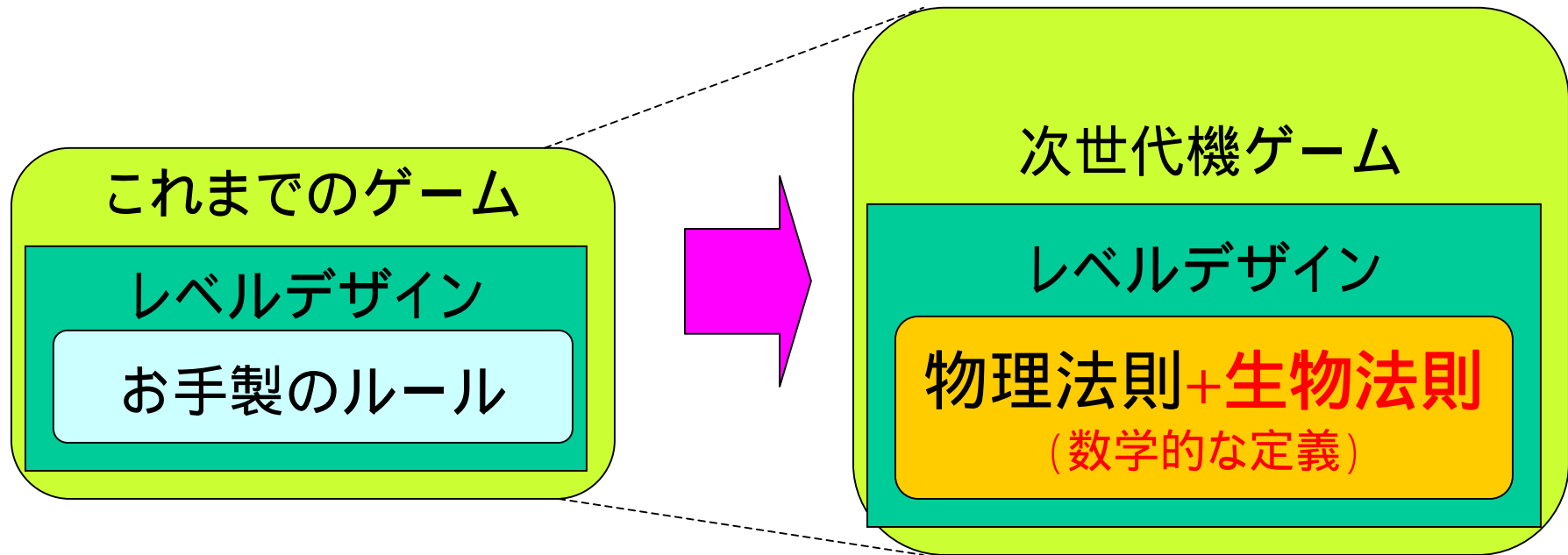
(暗黙的なルール) 物理



ゲームステージにおける物理運動の**プロシージャル化**

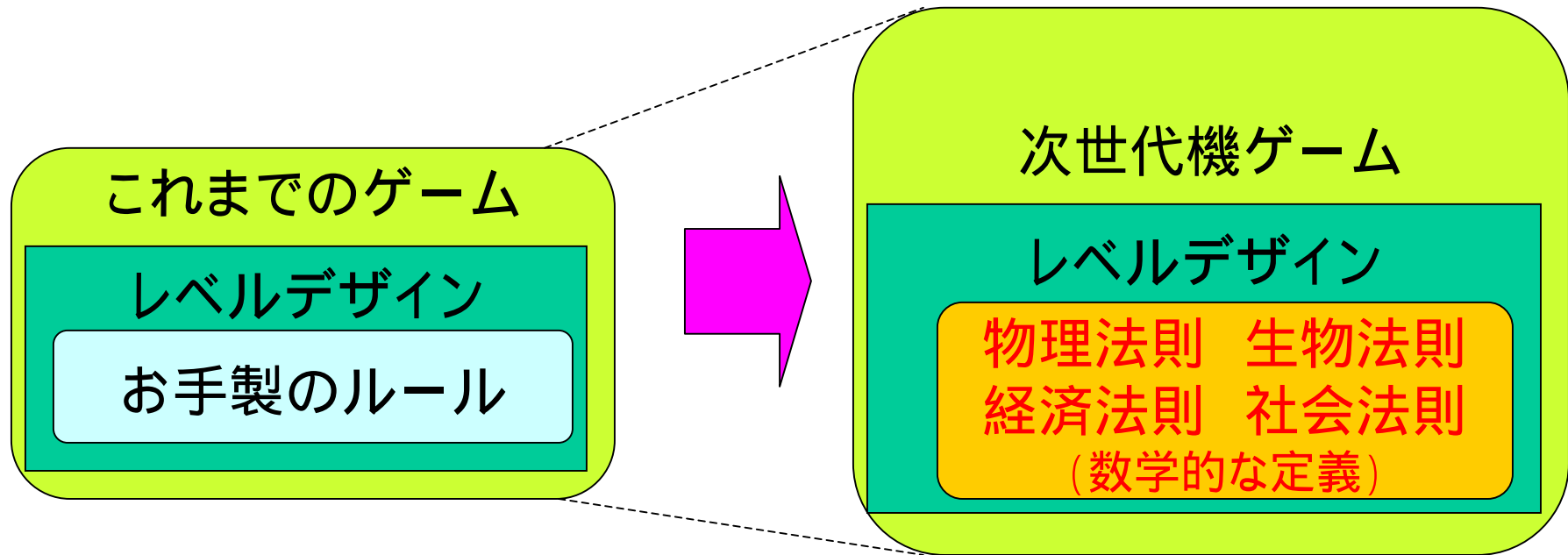
ゲームステージにおけるスケールの拡大に伴う
プロシージャル技術の導入

(暗黙的なルール为例) 物理 + 生物の法則



ゲームステージにおけるあらゆる運動の**プロシージャル化**

(暗黙的なルール) もっとたくさんある

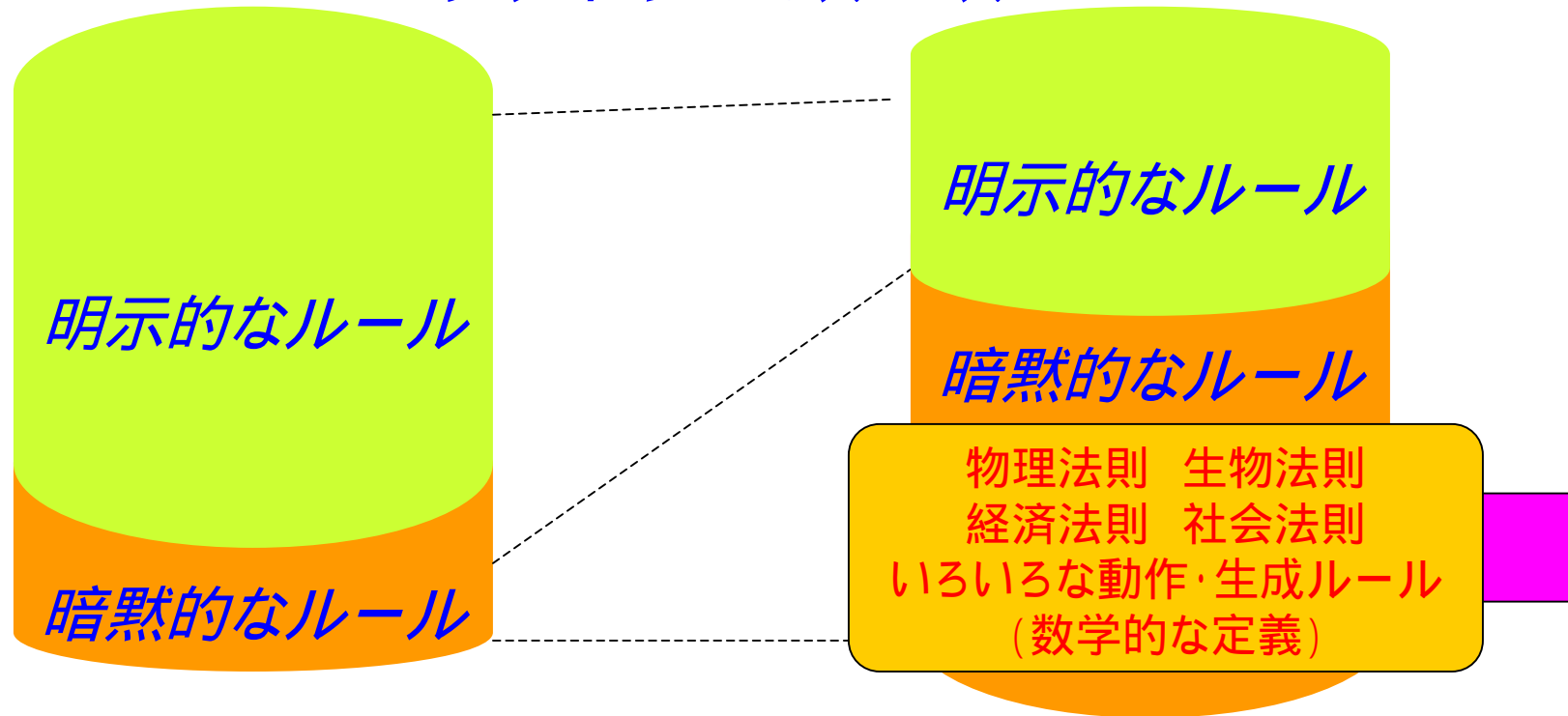


ゲームステージにおけるあらゆる運動の**プロシージャル化**

新しい大局的・局所的なルールを導入することで、
プロシージャル技術はゲームステージを**より豊かに駆動する**

ゲームとは何か？

プレイヤーとルール



非電源系ゲーム

(ボードゲーム・カードゲーム)

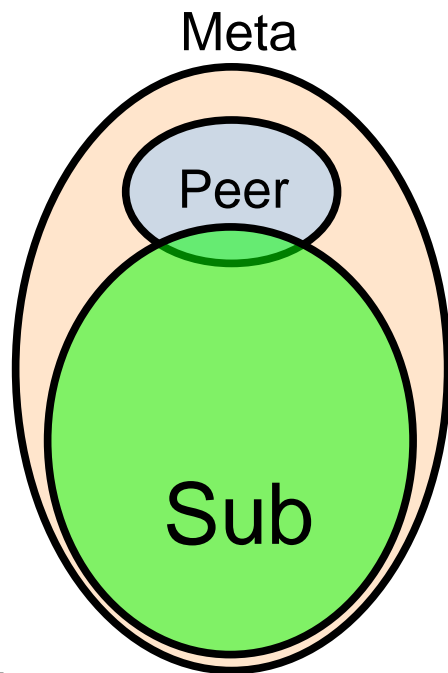
デジタルゲーム

デジタルゲームはボードゲームなどに比べて、暗黙的なルールを多く組み込めるという性質を持つ。

(デジタルゲームは根底から世界を再構成しているから)

The SimCity Sub AI

SimCity シリーズのAIの作り方



Meta 

Peer 

Sub 

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

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

世界をダイナミクス(力学系、動的な数値の仕組み)として動かす。
世界を動かす SubAI(=シミュレーション) を構築。

References

本講演は、主にWill Wright氏の5つのスライドの図を参考に構成されています。

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

に全てのPPTデータがあります。

(1) “AI: A Desing Perspective” AIIDE 2005 (PPTが公開)

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

(2) “MODELS COME ALIVE !” PC Forum 2003 (March 24)

(3) “Dynamics for designers” GDC 2003 (PPT,VIDEOが公開)

http://www.gamasutra.com/features/gdcarchive/2003/Wright_Will.ppt

http://www.gamasutra.com/features/20030403/wright_01.shtml

(4) “Desing Plunder” GDC2001

http://www.gamasutra.com/features/20010323/byrd_01.htm

(5) “The Future of Contents ” GDC 2005 (資料は未公開)

<http://www.4gamer.net/news/history/2005.03/20050314184429detail.html>

(動画サイトなどにも講演ムービーが上がっています)

Figures on the pages are from these references.

Crysis における植物アニメーション自動生成

Procedural Vertex Animation



http://www.crytek.com/fileadmin/user_upload/cryengine2/CryENGINE2Features.pdf

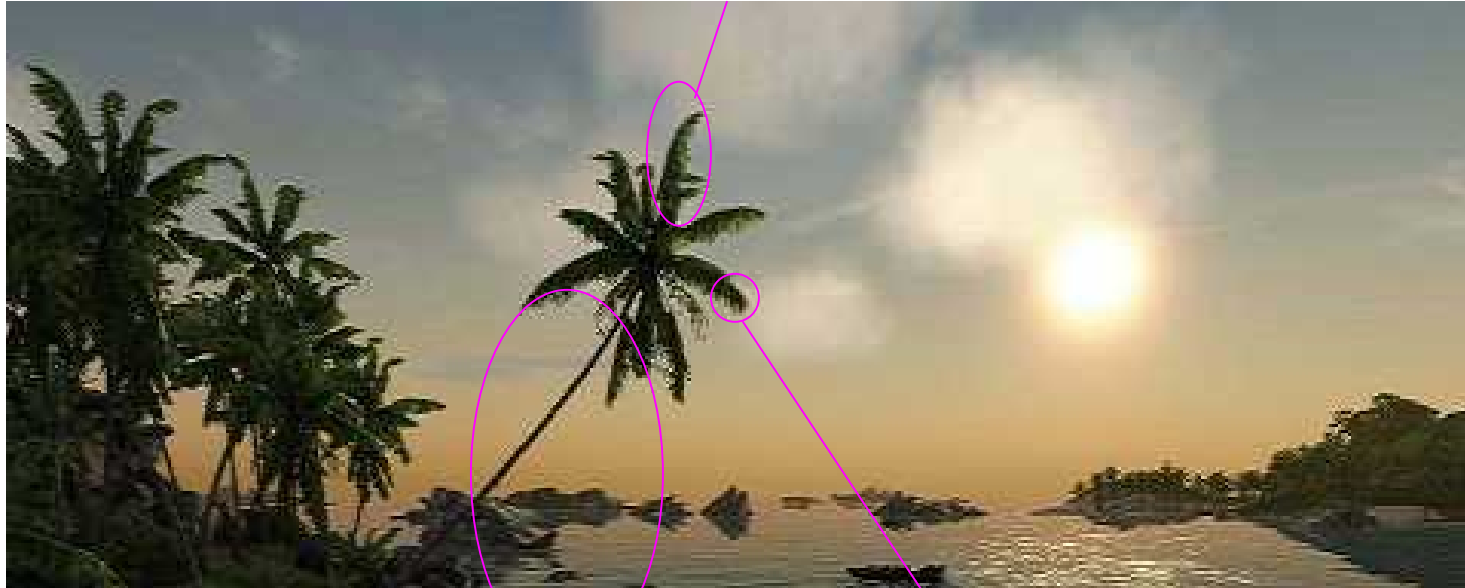
[Tiago Sousa, "Chapter 16: Vegetation Procedural Animation and Shading in Crysis", GPU Gems 3, 2007](#)

Finding Next Gen CryEngine2,

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

風に揺れる森 in Crysis

葉全体(leaves)

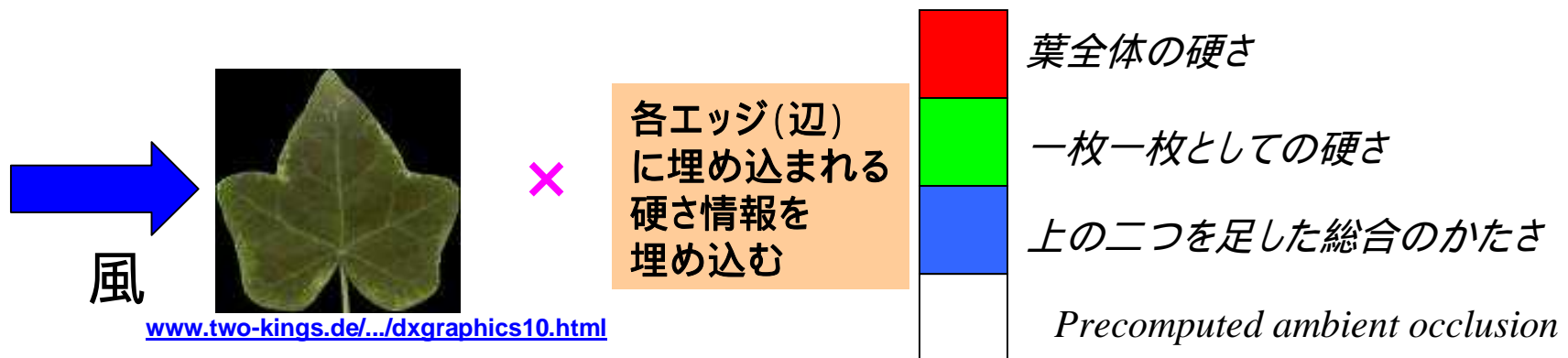


Finding Next Gen CryEngine2, [http://ati.amd.com/developer/gdc/2007/Mittring-Finding_NextGen_CryEngine2\(Siggraph07\).pdf](http://ati.amd.com/developer/gdc/2007/Mittring-Finding_NextGen_CryEngine2(Siggraph07).pdf)

全体(entire)

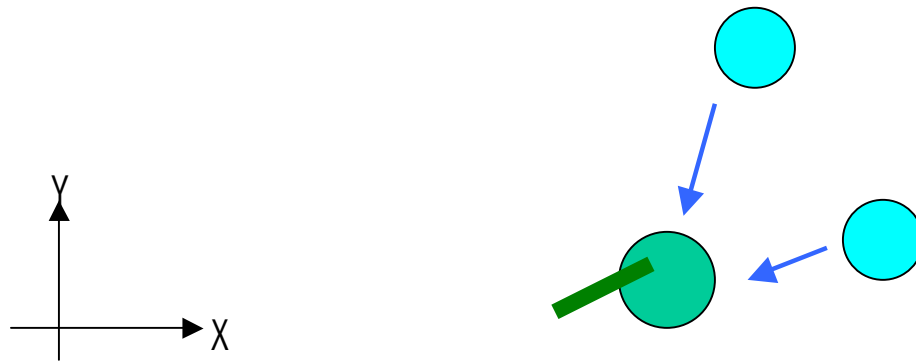
葉(leaf)

風と葉の硬さからモデルの曲がり具合を計算する



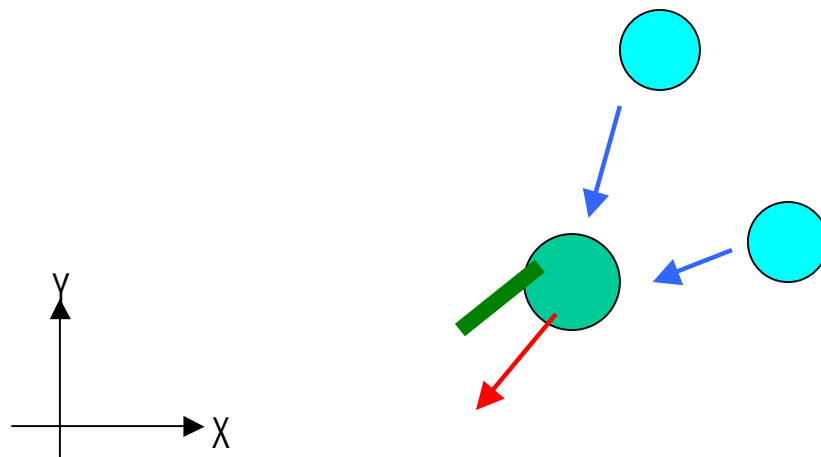
風に揺れる森 in Crysis

その樹木に対する幾つかの風源の効果を足し合わせる



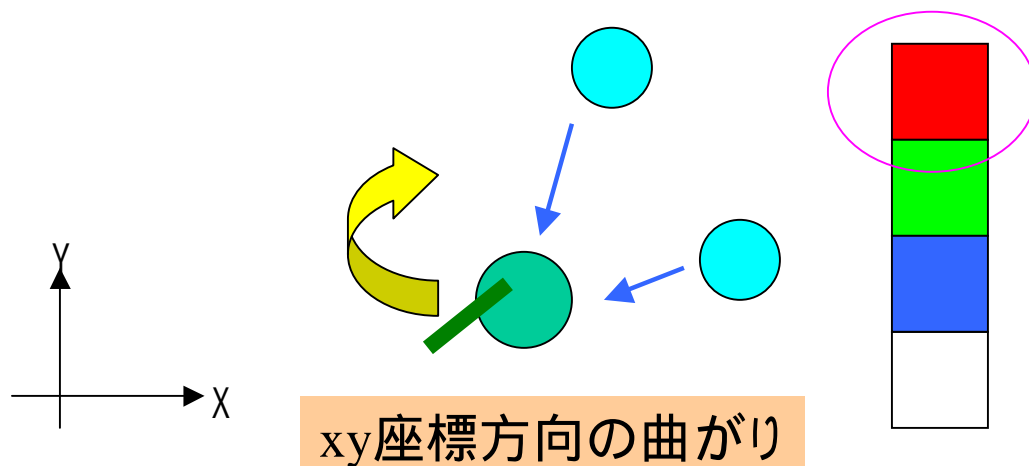
http://www.geimeris.com/wp-content/images/crysis_later.jpg

世界座標(xy)上の風ベクトルから、全体の曲げ方を決める。



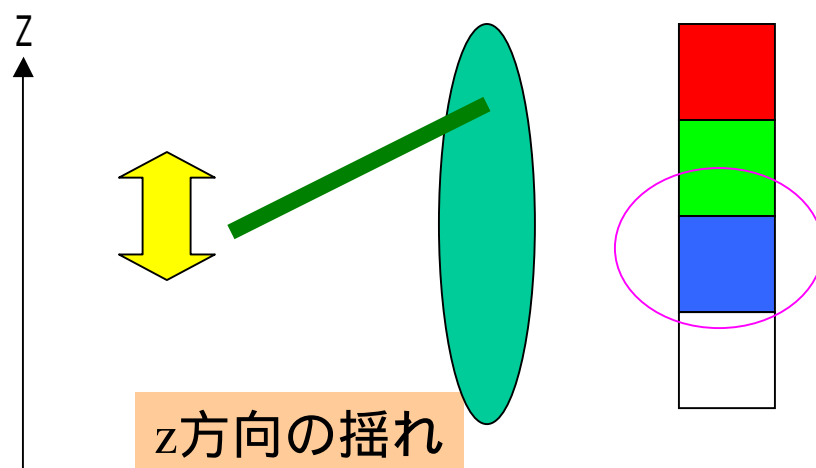
風に揺れる森 in Crysis

葉のまとまりに対する曲がりを、風の強さだけから計算する



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

一枚の葉の揺れ(z-方向)を、風の強さだけから計算する



デモ



**Procedural Vegetation
Animation and Shading**

CRYTEK



Procedural Vegetation Animation in Crysis

tiago_gpuGems3_1280_mpeg

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

References

- (1) SpeedTree
<http://www.speedtree.com/>
- (2) Procedural vertex-animation(OceanWater)
http://b0rken.dk/hornet/shaders/proc_water.html
- (3) [Tiago Sousa, "Chapter 16: Vegetation Procedural Animation and Shading in Crysis", GPU Gems 3, 2007](#)
- (4) natural motion, <http://www.naturalmotion.com>
- (5) E3 2006: Indiana Jones Eyes-on,
<http://ps3.ign.com/articles/705/705489p1.html>
- (6) Finding Next Gen CryEngine2,
[http://ati.amd.com/developer/gdc/2007/Mittring-Finding_NextGen_CryEngine2\(Siggraph07\).pdf](http://ati.amd.com/developer/gdc/2007/Mittring-Finding_NextGen_CryEngine2(Siggraph07).pdf)

Figures on the pages are from these references.

暗黙的なルールが入ったゲーム

物理法則



Havok

経済法則

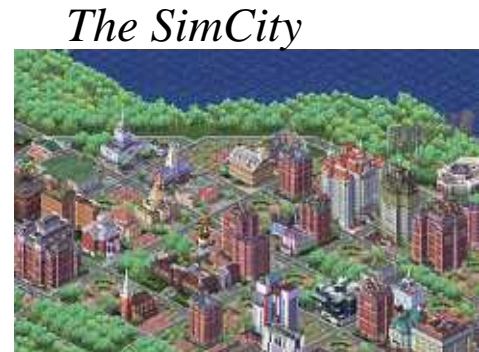


FF11

生物法則



生成ルール



新しい大局的・局所的な**数学的な法則**や**動作・生成のルール**を導入することで、プロシージャル技術はゲームをより豊かに**駆動する**

第2部

第1章

ゲーム世界(レベルデザイン)とプロシージャル

第2章

キャラクター(NPC,AI)システムとプロシージャル

第3章

ユーザー・アクションとプロシージャル

第4章

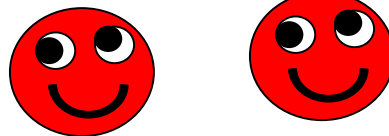
ゲームシステムとプロシージャル

第5章

まとめ

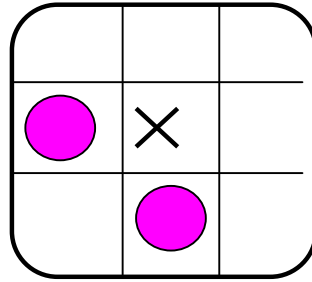
第2章

キャラクター (NPC, AI) システムと プロシージャル



ちよつとここで、
人工知能の用語を一つ覚えましょう！

AIが対象とする状態空間の数



高々、 $9!$ 個

オフタイム
ディスクリート(離散)

数

リアルタイム
コンティニュアス(連続)

スケール



将棋 10 の 38 乗



囲碁 10 の 171 乗

AIラボ講演 3日目(2コマ目)

バーチャル



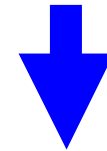
スポーツ(F1)



エンターテインメント

デジタルゲーム

AIラボ講演 3日目(1コマ目)

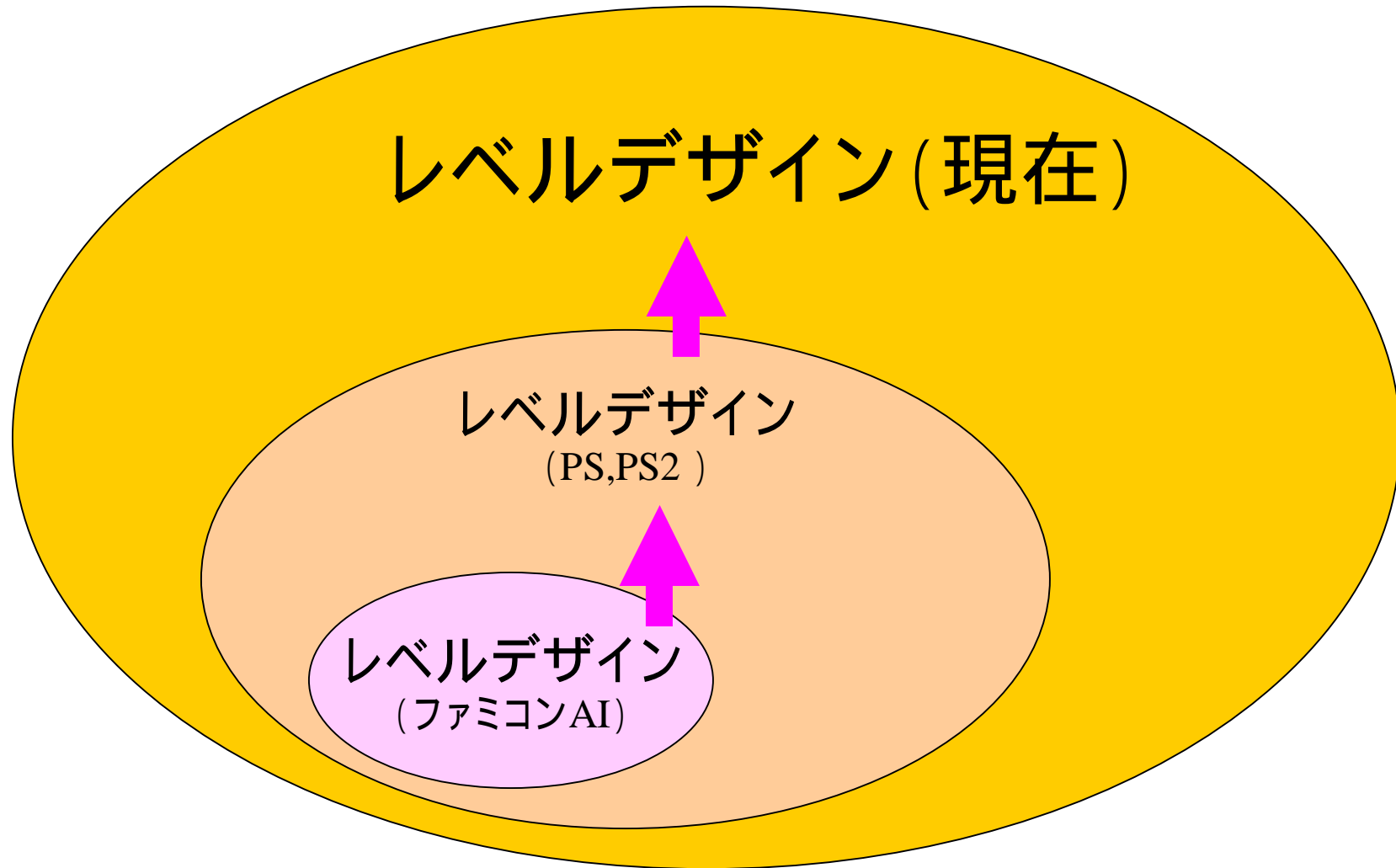


リアル



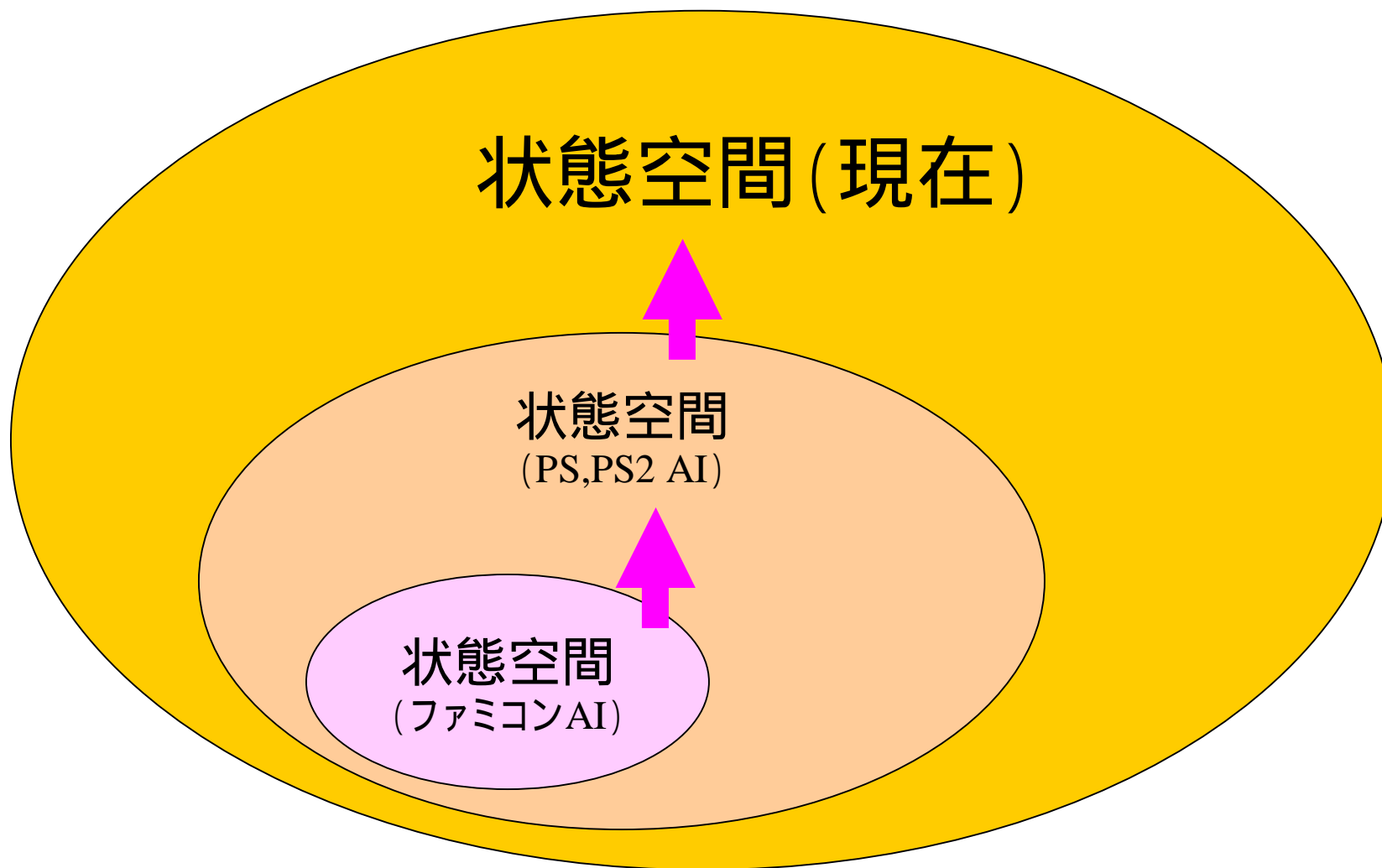
IMAGIRE DAY1日目(3コマ目)

レベルデザインのスケールの拡大と複雑化



この30年で我々が触れることが出来るデジタル空間は格段に増大した

AIが対象とする状態空間の拡大



この30年で我々が触れることが出来るデジタル空間は格段に増大した

AIは想定される全ての状態空間に
対応しなければならないが、

どのようなアプローチが考えられるだろうか？

(ゲームAI)

AIの2つのアプローチ

ゲーム世界に適応したAIを作りなさい

アプローチ1

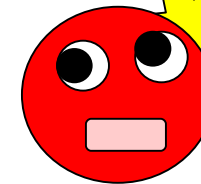
イベントに対して
決められた行動を
する

話しかけられた



こんぼうは
120ゴールド
になります

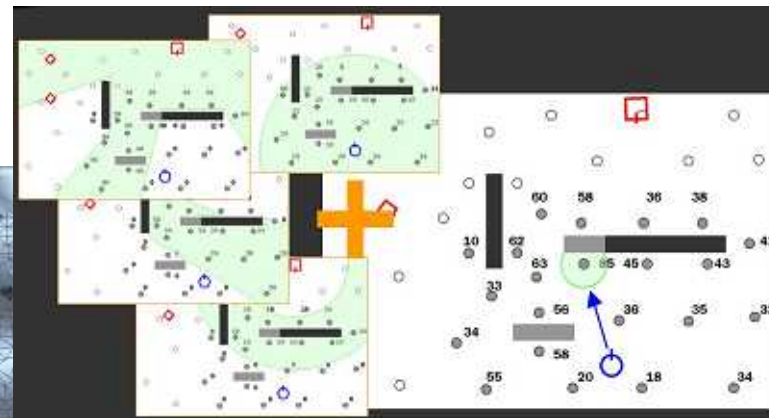
武器を出した!



Attack!

アプローチ2

周囲の情報を
(アルゴリズム、
プログラムによって)
解析して
判断して行動する



Killzone AI: Dynamic Procedural Combat Tactics

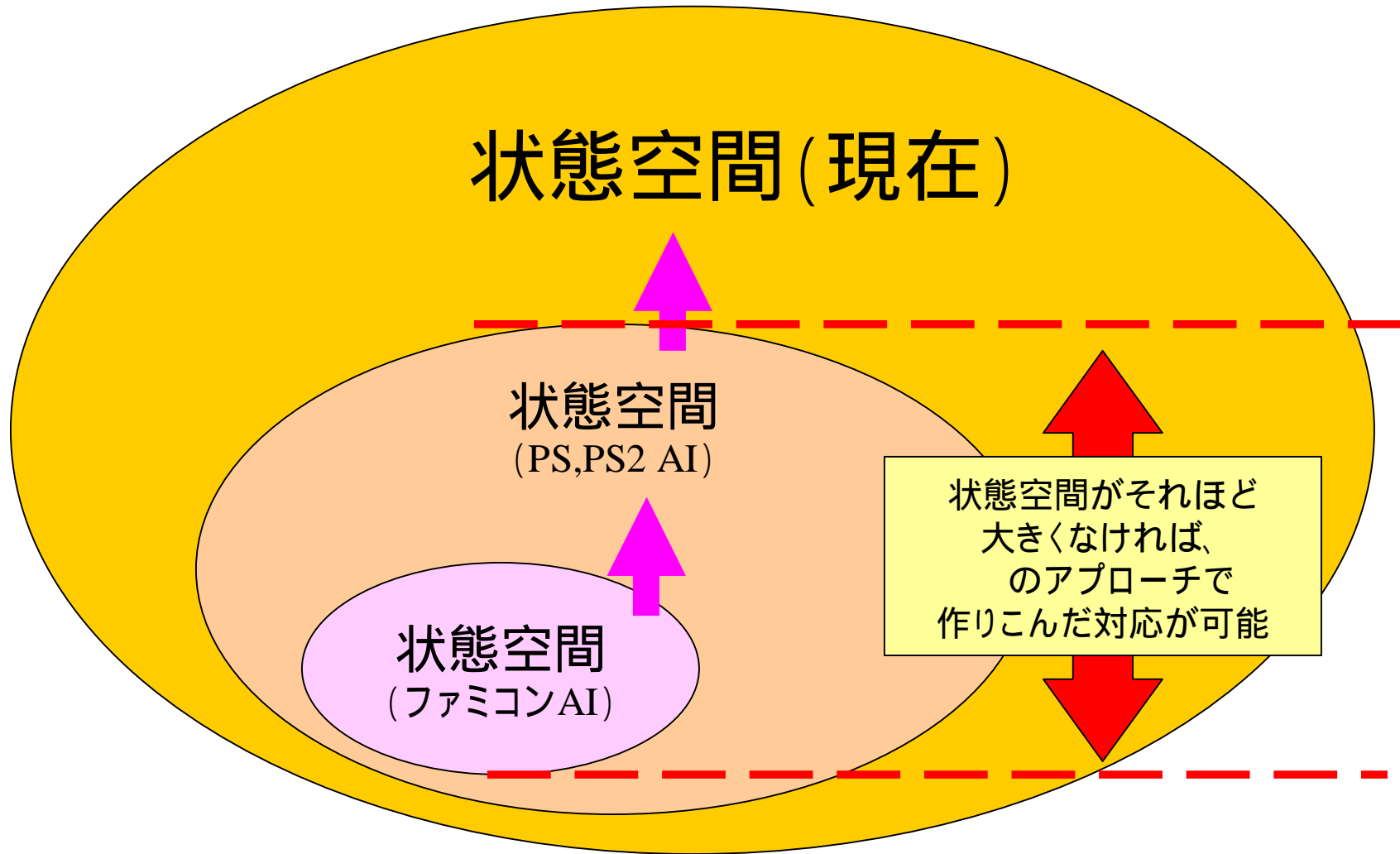
プロシージャル

プロシージャル技術 → ゲームデザインへ

たくさんの演算によってキャラクターの行動を決定する

プロシージャル AI (Procedural AI)

AIが対象とする状態空間の拡大

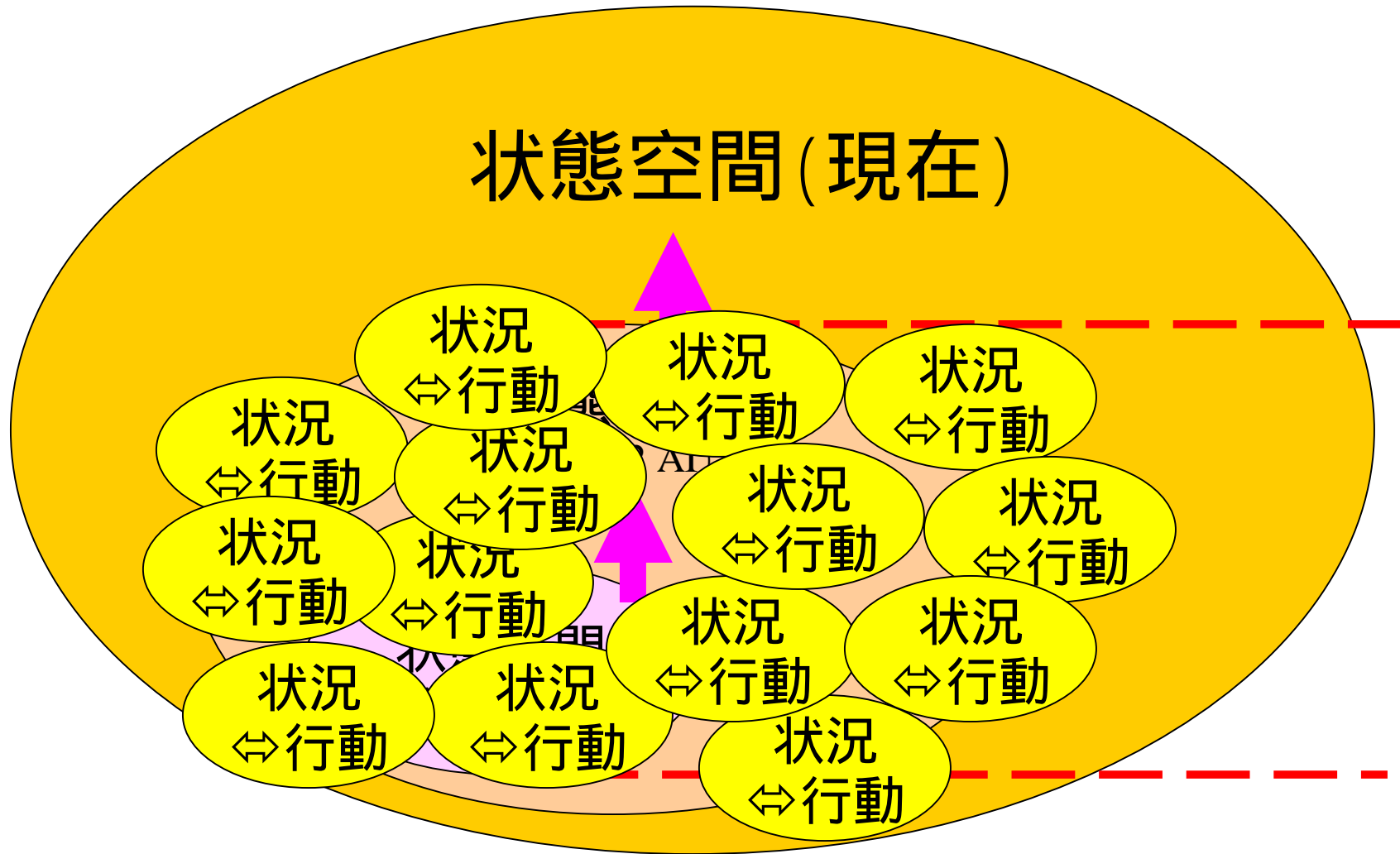


Case 1 ...

Case 2 ...

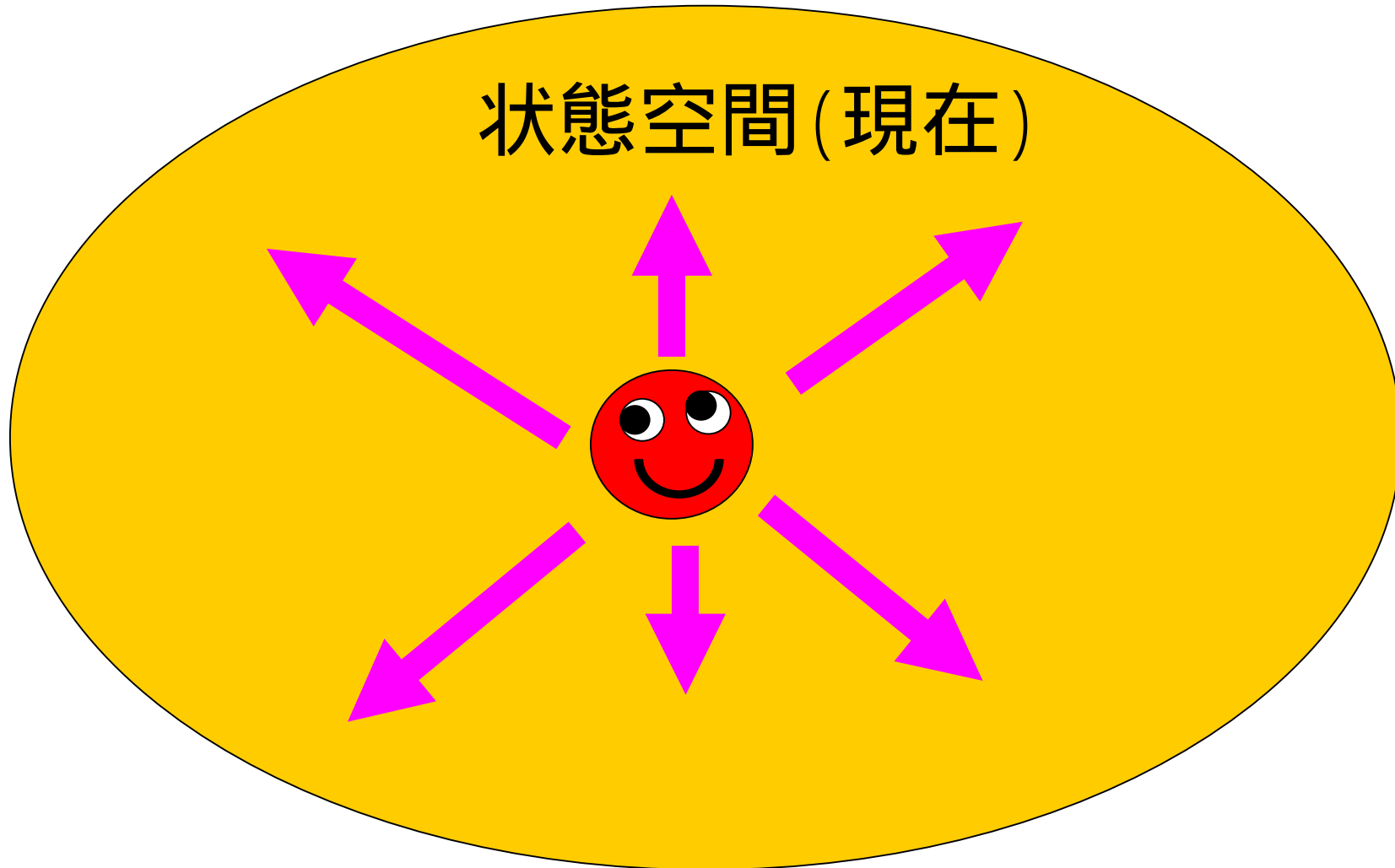
Case 3 ...

AIが対象とする状態空間の拡大



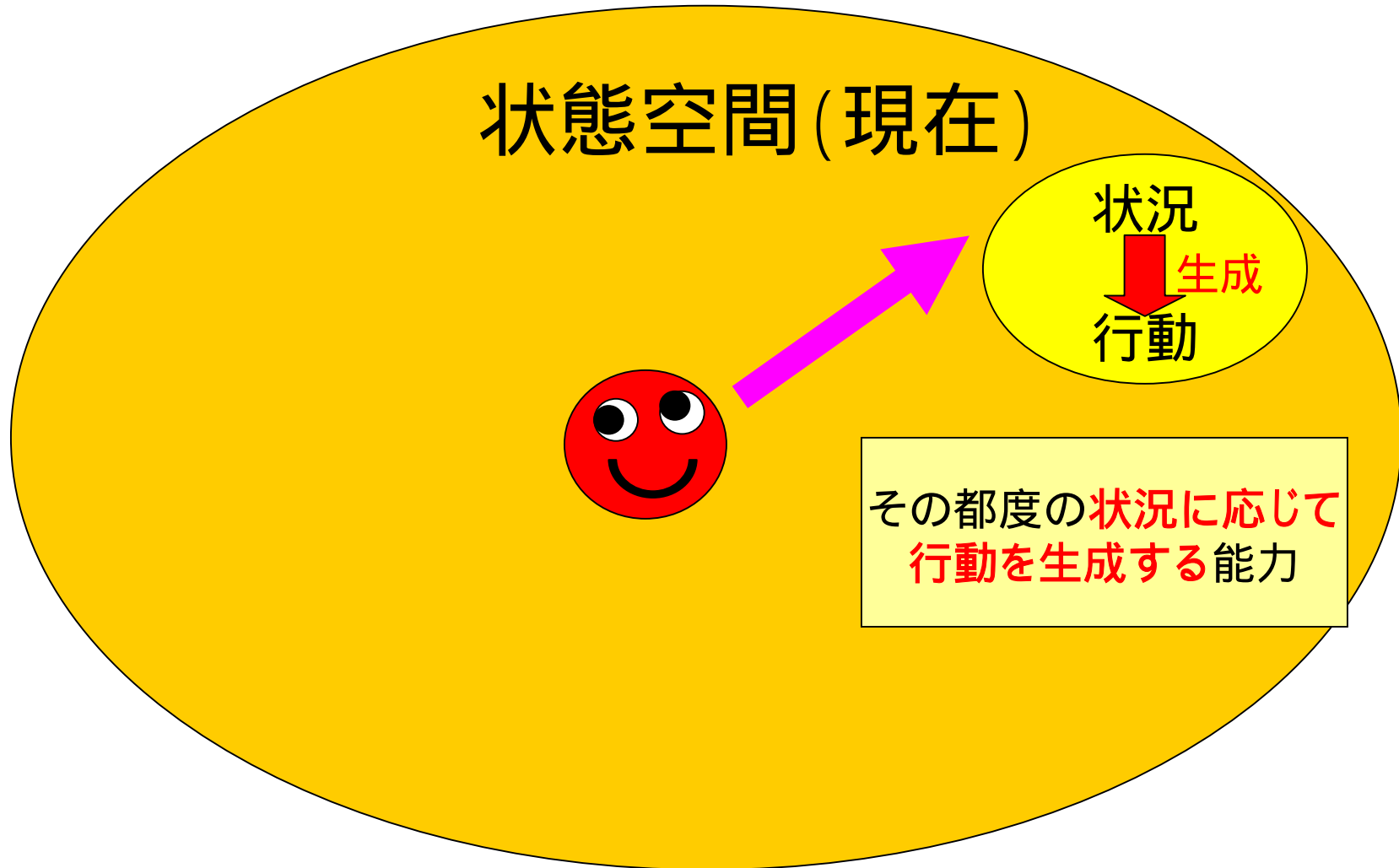
Case 1 ... Case 2 ... Case 3 Case n

AIが対象とする状態空間の拡大

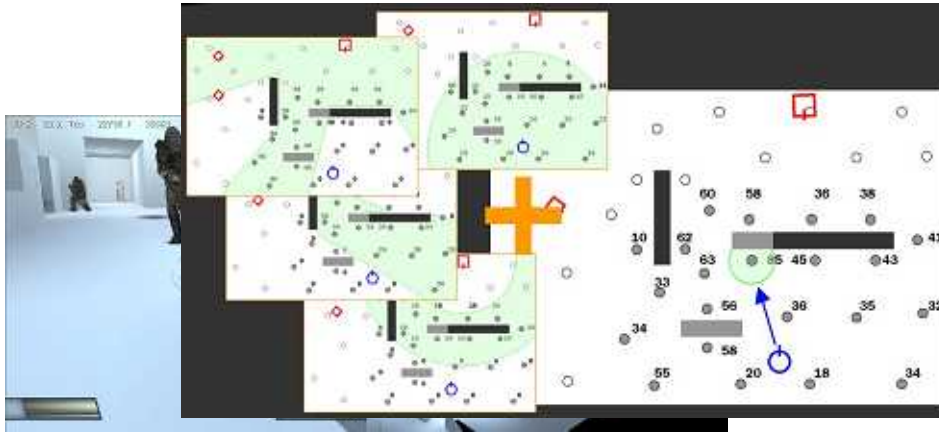


しかし状態空間が複雑かつ大きくなれば、
プログラマーに状態空間に対応する能力がAIに必要とされる

AIが対象とする状態空間の拡大



しかし状態空間が複雑かつ大きくなれば、
プログラマーに状態空間に対応する能力がAIに必要とされる



Killzone

- マップ上の複数の評価値を計算して行動を決定する



Chromehounds

- 状況に応じてプランを生成する



Far cry Instincts

- プレイヤーの行動に応じた行動
 ←→ スクリプト

Killzone におけるプロシージャルAI

広大なマップで戦術的に移動し攻撃するNPC



バトルフィールドでは
最大7vs7(一人はプレイヤー)
で戦う



味方

人間

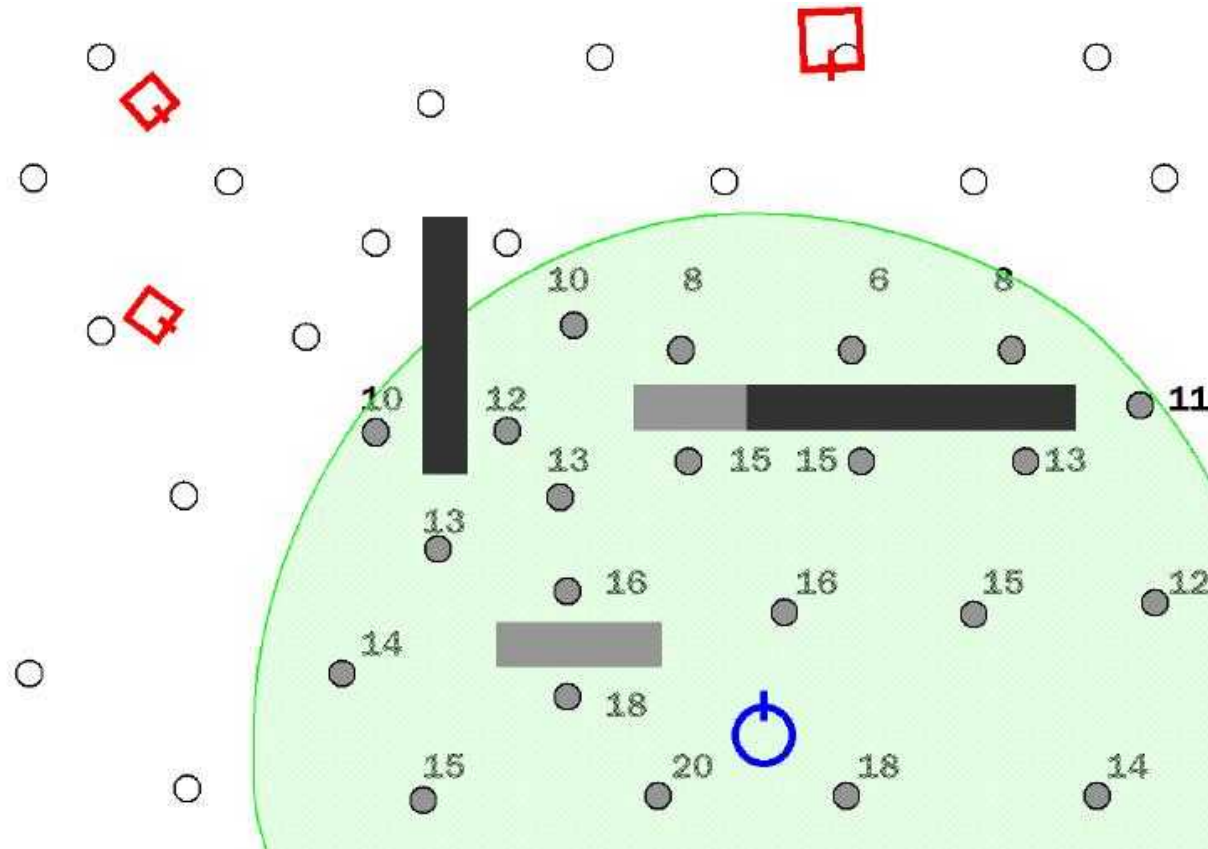
(強化)人間

敵



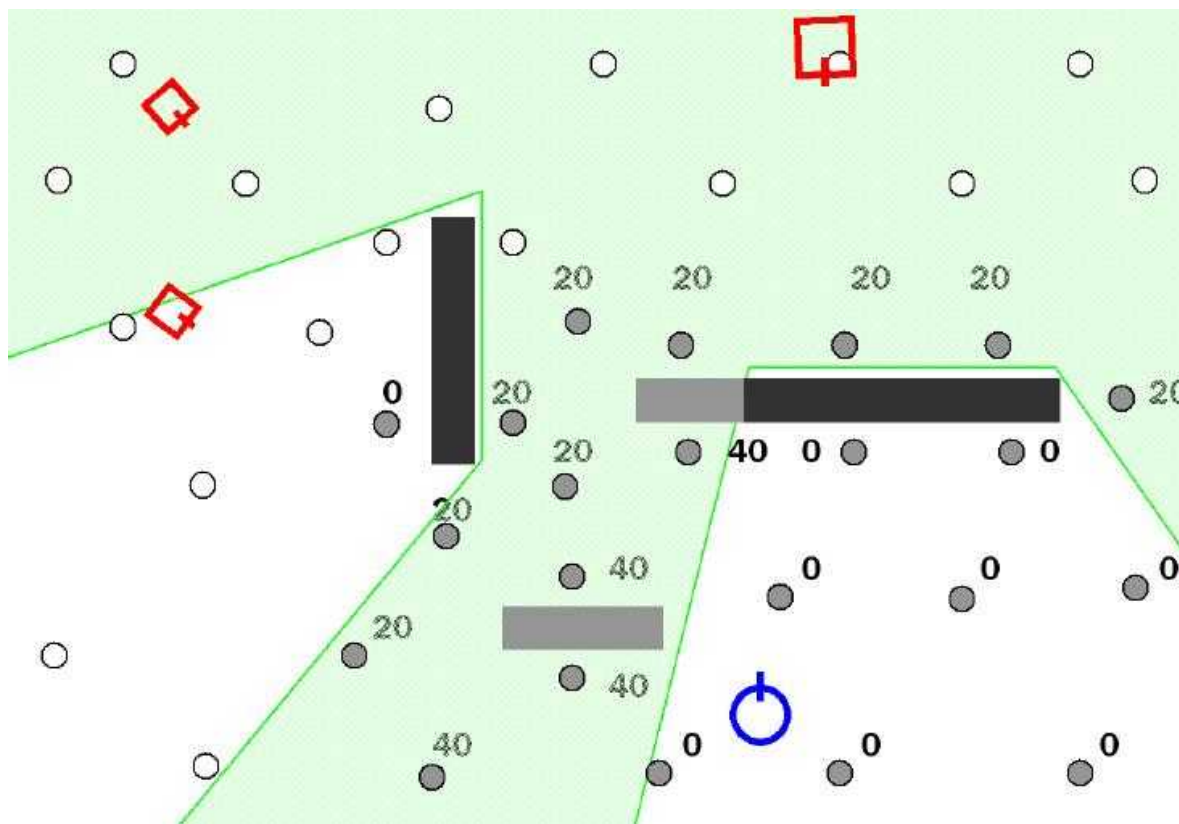
Step1

一定半径内のポイントを候補に上げる



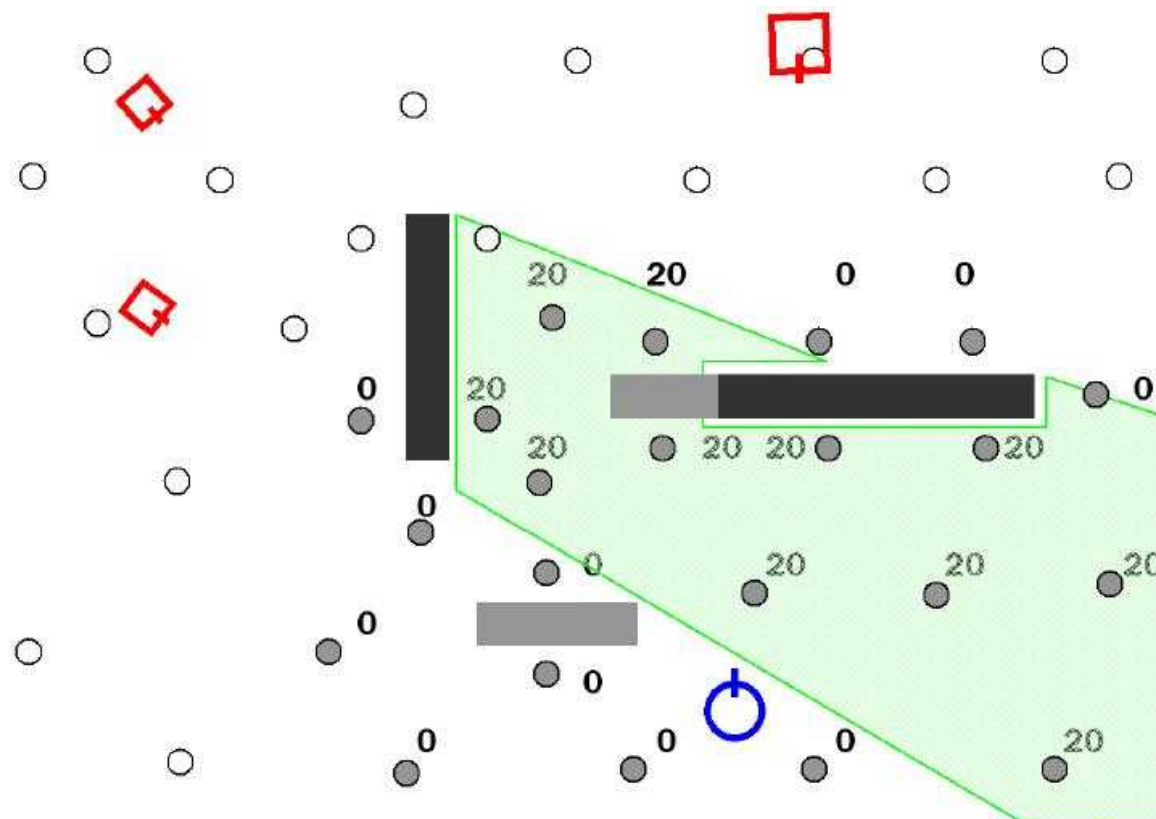
この時点で他のポイントはこれ以降、評価しない → 計算量の軽減

Step2 第一の敵から射線の通るポイント进行评估する

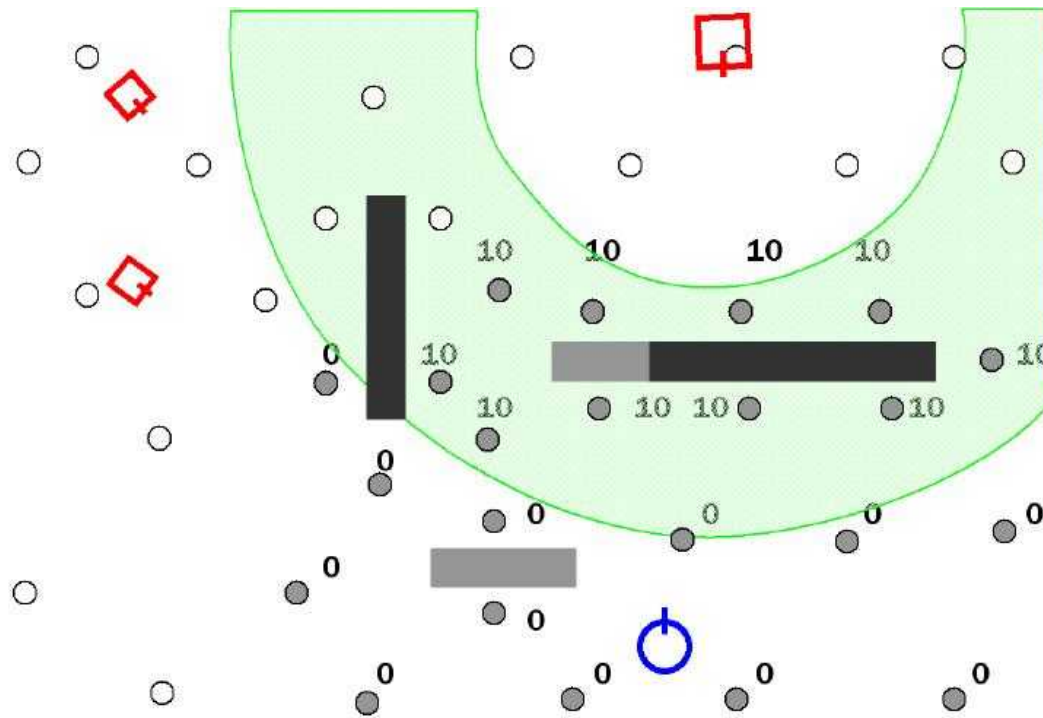


- 40点 ... 立った状態のみ射線が通る
- 20点 ... 座っても立っても射線が通る

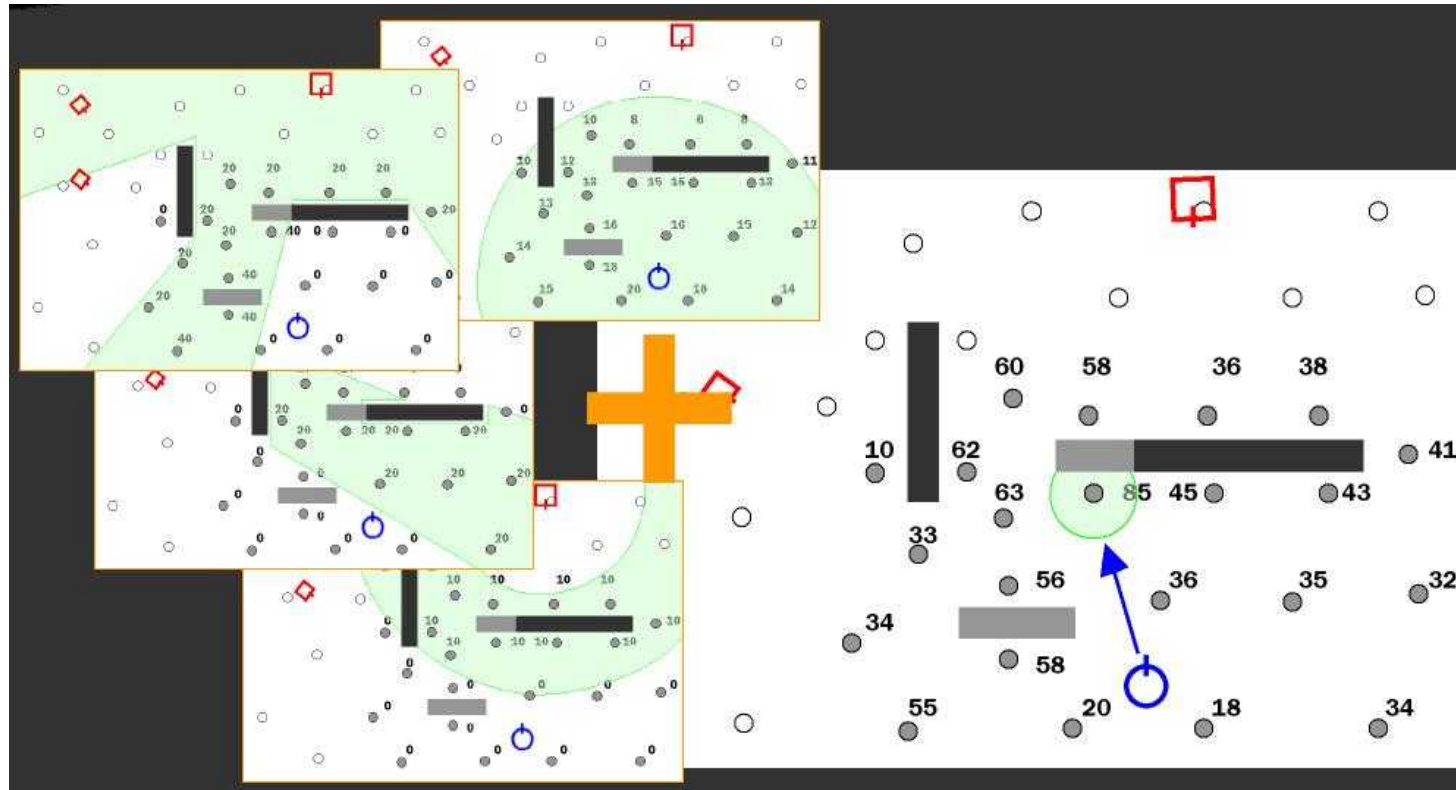
Step3 第二の敵たちから射線の通らないポイント进行评估する



Step4 攻撃最適領域内のポイント进行评估する

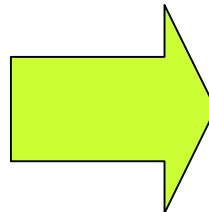


Killzone におけるプロシージャルAI



状況

敵の位置と自分の位置



行動

射線の通り具合から
最適な位置を計算して移動

References for Killzone

- [1] **William van der Sterren** (2001), "Terrain Reasoning for 3D Action Games", http://www.cgf-ai.com/docs/gdc2001_paper.pdf
- [2] **William van der Sterren** (2001) , "Terrain Reasoning for 3D Action Games(GDC2001 PPT)", http://www.cgf-ai.com/docs/gdc2001_slides.pdf
- [3] **Remco Straatman, Arjen Beij, William van der Sterren** (2005) , "Killzone's AI : Dynamic Procedural Combat Tactics", http://www.cgf-ai.com/docs/straatman_remco_killzone_ai.pdf
- [4] **Arjen Beij, William van der Sterren** (2005), "Killzone's AI : Dynamic Procedural Combat Tactics (GDC2005)", http://www.cgf-ai.com/docs/killzone_ai_gdc2005_slides.pdf
- [5] **Damian Isla** (2005), "Dude, where's my Warthog? From Pathfinding to General Spatial Competence", <http://www.aiide.org/aiide2005/talks/isla.ppt>

Figures on following pages are from these references.

ChromeHounds におけるプロシージャルAI

15分間、広大なフィールドで
計画性を持って人間と戦うことができるAI

(マルチエージェント)



プレイヤーチーム(最大6名)

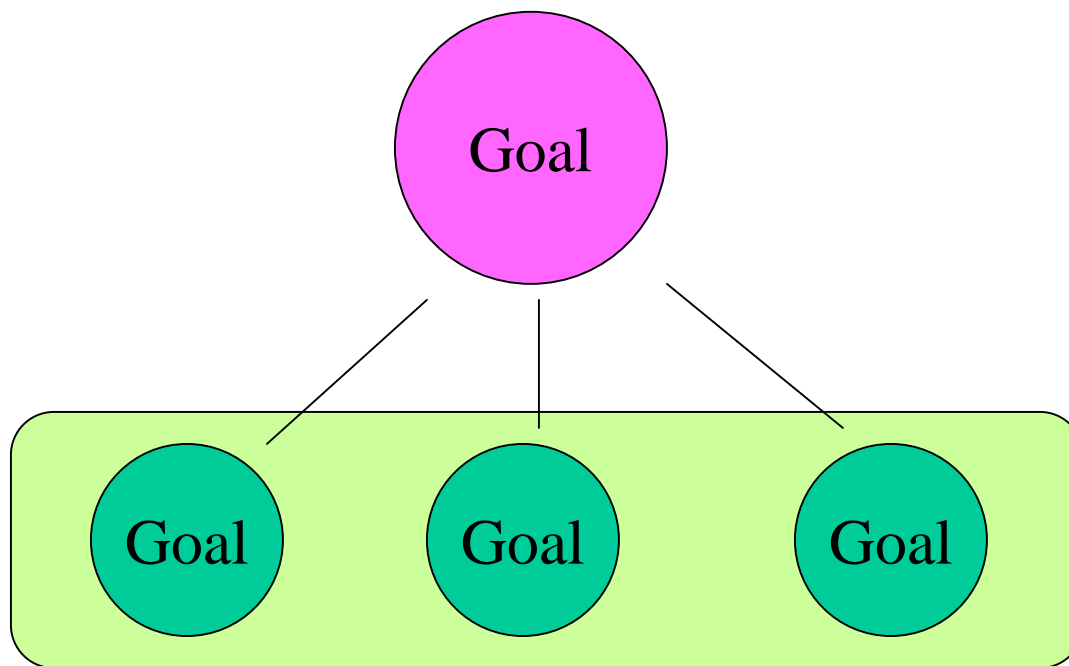


AIチーム(最大6名)

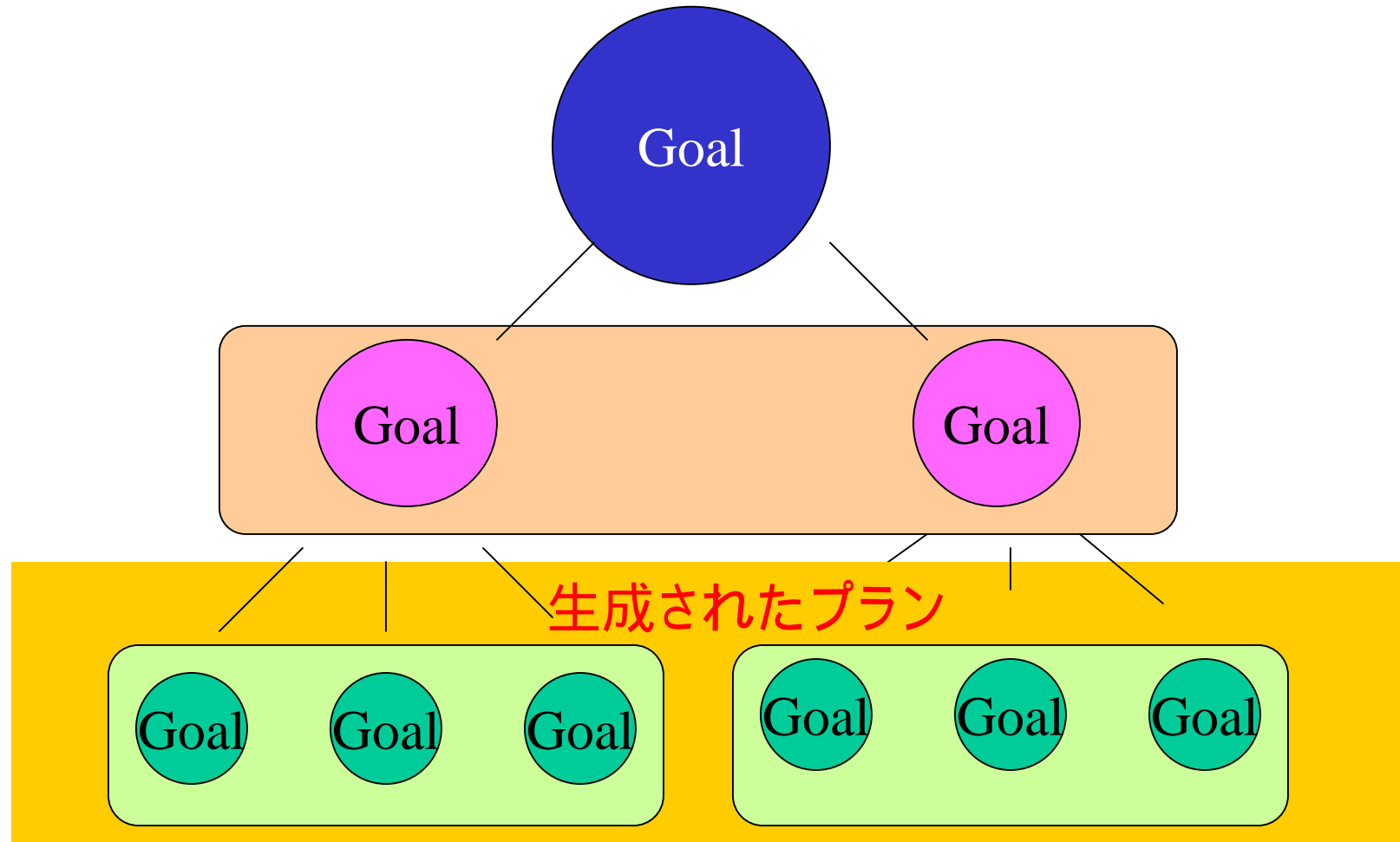


階層型ゴール指向プランニング

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



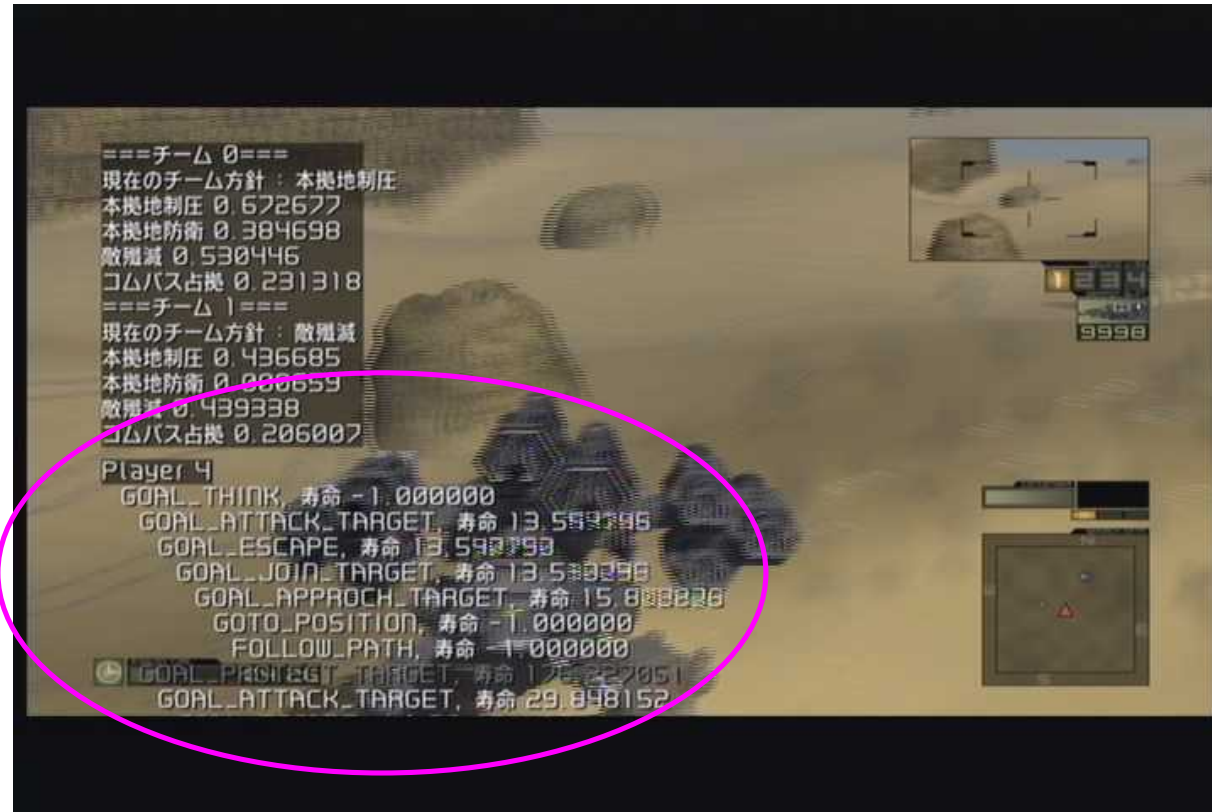
生成されたプランを順番に小ゴールを実行することで、
中間ゴールが達成され、大ゴールが達成される

デモ



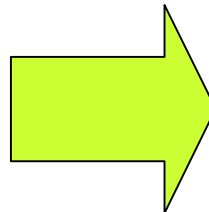
Chromehounds プランニングシステム

Chromehounds におけるプロシージャルAI



状況

戦闘の展開の進み方
敵と味方の位置
自分の位置



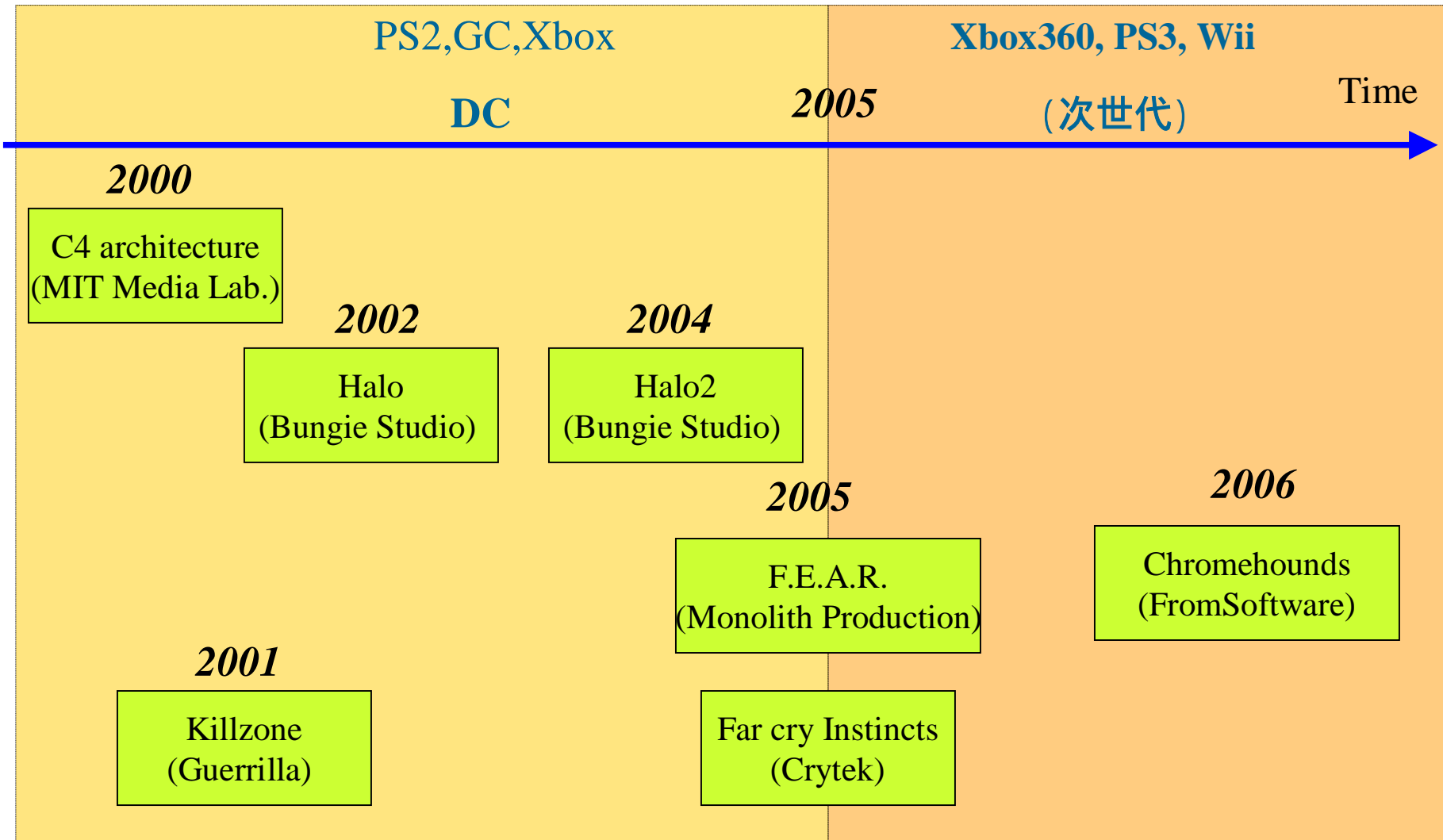
行動プラン

プランニング思考によって
長期的な戦略行動プランを作成

References for Chromehounds

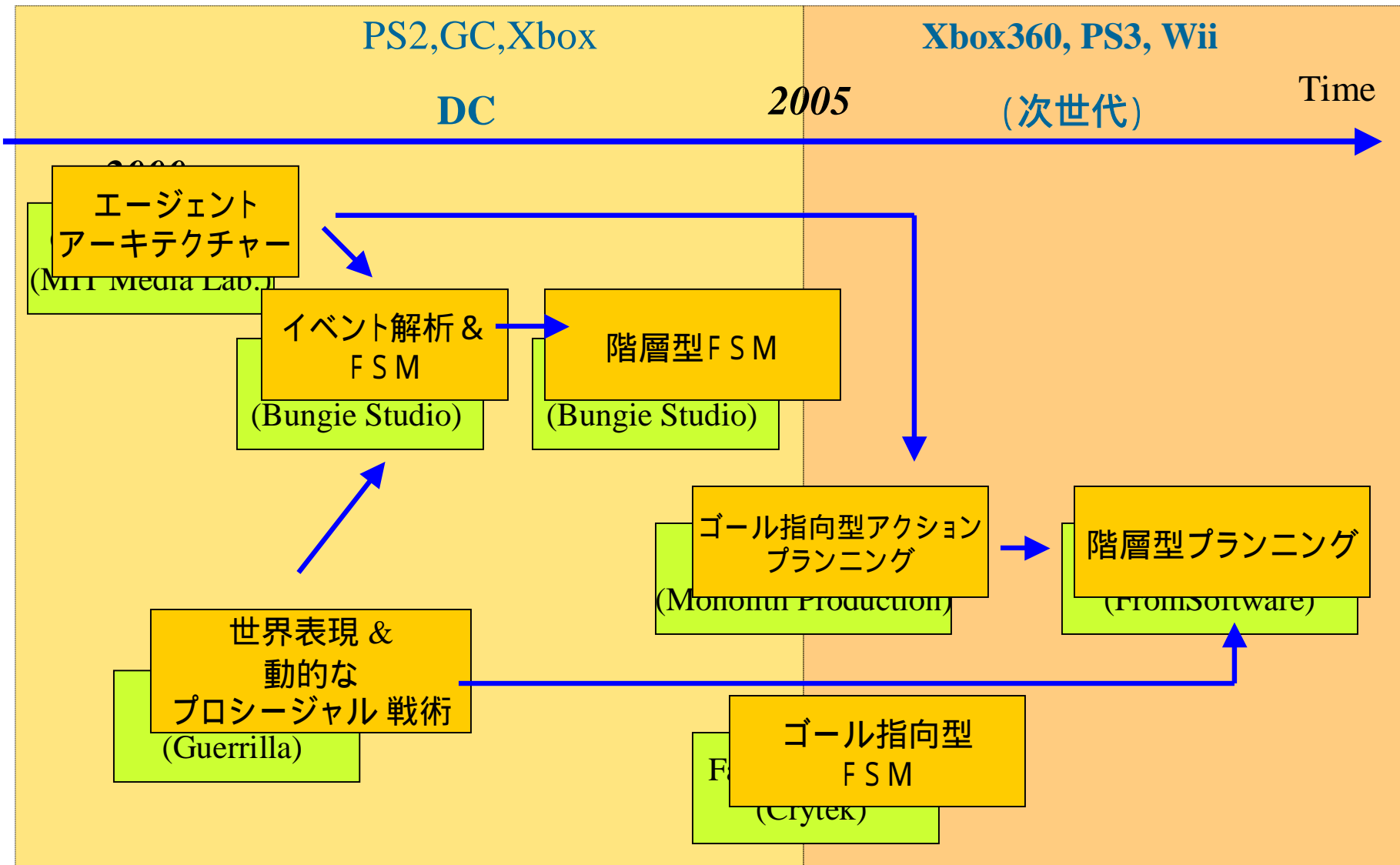
- (1) 三宅陽一郎: クロムハウズにおける人工知能開発から見るゲームAIの展望,
CEDEC2006講演資料 (2006)
<http://www.igda.jp/modules/mydownloads/visit.php?cid=2&lid=63>
- (2) 三宅陽一郎: エージェント・アーキテクチャーから作るキャラクターAI,
CEDEC 2007 講演資料(2007) (**Killzone, Halo, Halo2**)
<http://www.igda.jp/modules/mydownloads/visit.php?cid=2&lid=78>
- (3) 三宅陽一郎, 「人工知能が拓くオンラインゲームの可能性」AOGC2007,
<http://www.bba.or.jp/AOGC2007/2007/03/download.html>

プロシージャルの歴史: AI編



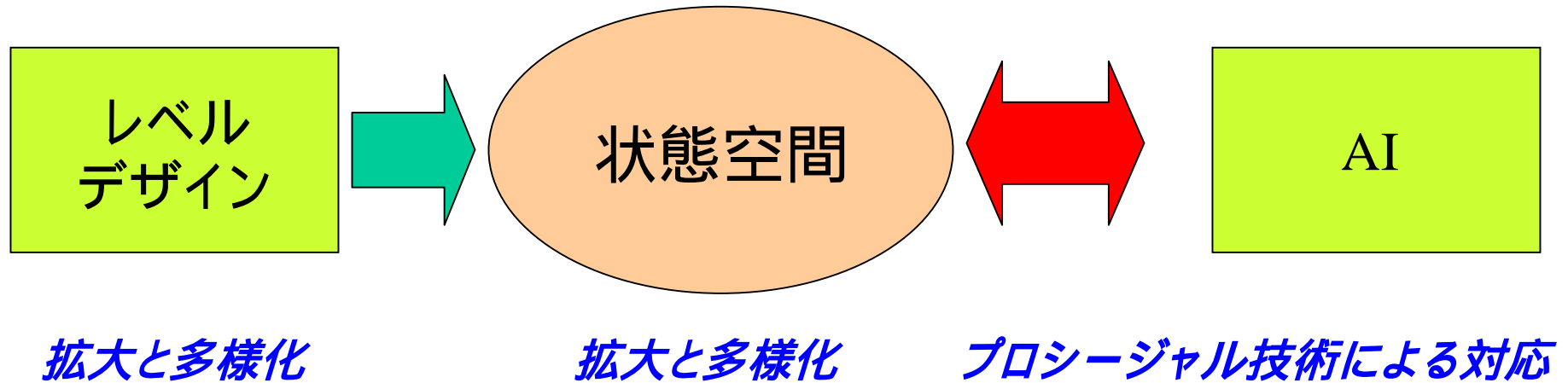
プロシージャルAI = アルゴリズムによるAI

プロシージャルの歴史: AI編



プロシージャルAIにおけるいろいろなアルゴリズム

プロシージャルAI



拡大し多様化するレベルデザインと状態空間を、
数学的に解析してアルゴリズムによって自動的に行動を生成する
のが、プロシージャルAI技術である。

プロシージャルAIによって、ますます拡大し複雑化するレベルデザインに
適応したAIを作成することができる。

第2部

第1章

ゲーム世界(レベルデザイン)とプロシージャル

第2章

キャラクター(NPC, AI)システムとプロシージャル

第3章

ユーザー・アクションとプロシージャル

第4章

ゲームシステムとプロシージャル

第5章

まとめ

第3章

ユーザー・アクションとプロシージャル



ちよつとここで、
ゲームデザインの用語を一つ覚えましょう！

可能性空間 (the space of possibility)

Katie Salen, Eric Zimmerman, "Rules of Play: Game Design Fundamentals", MIT Press, 2003



ルールズ・オブ・プレイ

可能性空間 = ユーザーの行為が定義される空間

But game designers do not directly design play. They only design the structures and contexts in which play takes place, indirectly shaping the actions of the players. We call the space of future action implied by a game design *the space of possibility*. It is the space of all possible actions that might take place in a game, the space of all possible meanings which can emerge from a game design. (P.67, Unit 1: Core Concepts | interactivity)

しかし、ゲームデザイナーは直接プレイをデザインするわけではない。彼らは、プレイが起こる構造やコンテキストをデザインするだけで、そこから間接的に、プレイヤーのアクションを形作るのである。我々はゲームデザインによって暗に定義された実行可能なアクションの空間を可能性空間と呼ぶことにする。可能性空間とはゲーム内で実行可能な全ての行動のことであり、またゲームデザインから導かれる可能な意味の空間である。

可能性空間

リアル

ゲーム空間

ユーザー

可能性空間

(=ユーザーのアクション
がゲーム空間内で
影響・意味のある範囲)

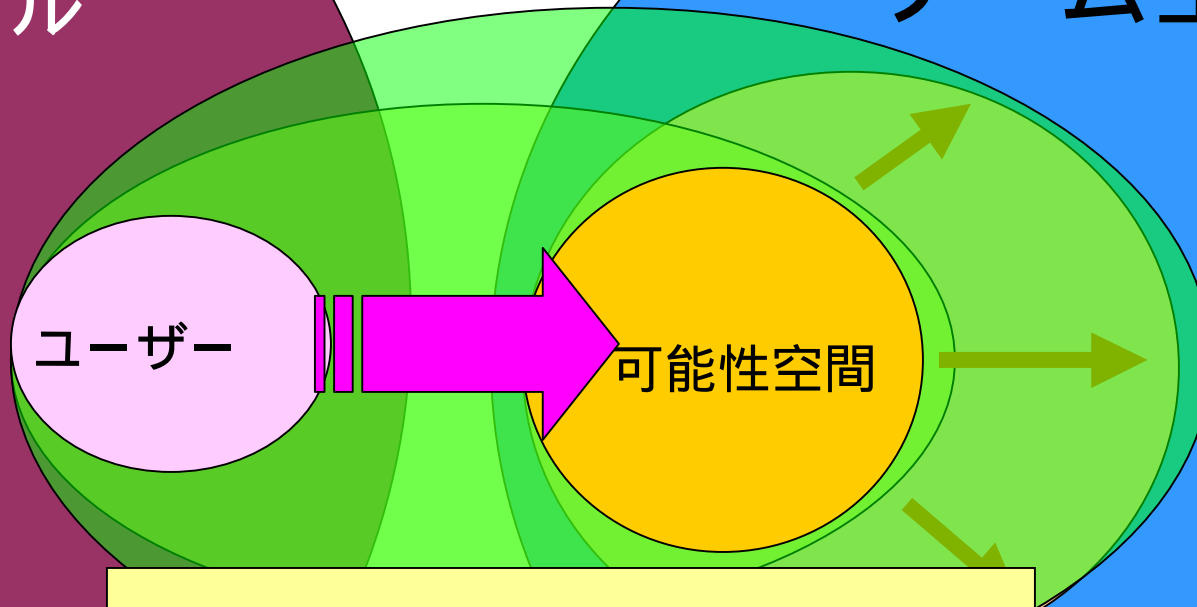
ユーザーの現実感覚のバーチャル空間への拡張
& ユーザーの全能感の実現
(自分がコントロールしているという感覚)



可能性空間

リアル

ゲーム空間



プロシージャル技術によって
可能性空間を拡張することができる



Sporeにおけるプロシージャル・ユーザー・アクション

簡単な操作で3Dクリーチャー・モデルからアニメーションまでを作成する

ユーザー作成コンテンツ (User Generated Contents, **UGC**)



キャラクターを作成するエディター



車を作成するエディター



都市(要塞)を作成するエディター



宇宙船を作成するエディター

生物、建物、惑星、主要なものは全てユーザーが生成可能

シミュレーション
(Sim Everything !)

単細胞から宇宙までの進化シミュレーションゲーム



単細胞フェーズ



部族フェーズ

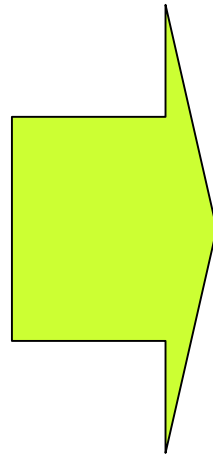


文明化フェーズ



ギャラクティックフェーズ

ユーザーから見たSpore エディター

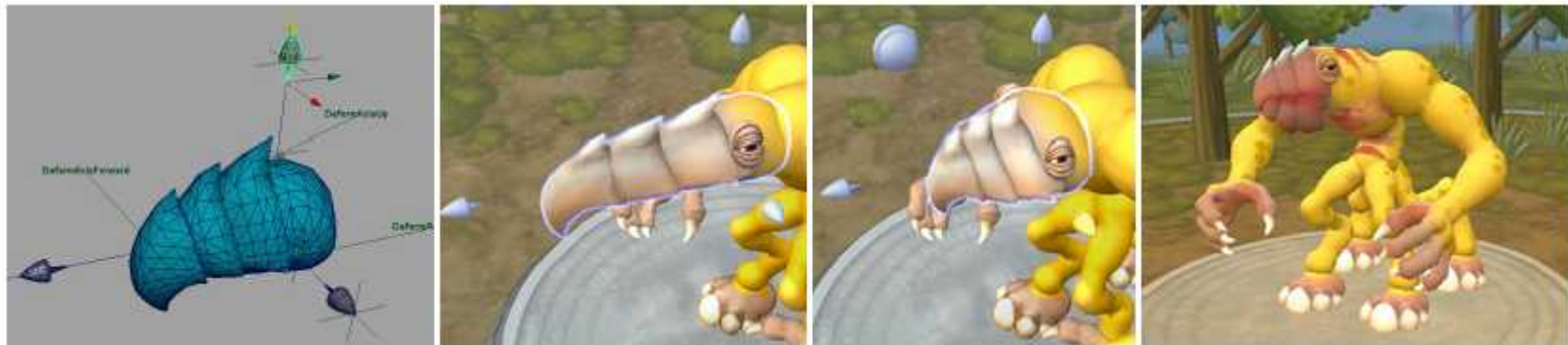


パーツをドラッグアンドドロップして、
パーツをモーフィング変形させて、
テクスチャーを選んで
モデルを作る

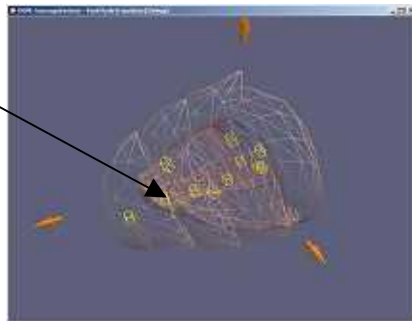
できたモデルはすぐにアニメーション！

(1) デフォーマブル・モデル

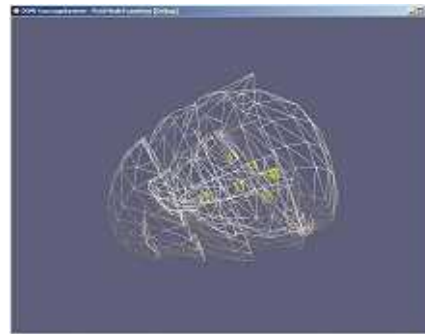
モデルは「リゴブロック」と呼ばれるパーツからなる
「リゴブロック」はマウスによって制御ポイントを動かして
変形することができる



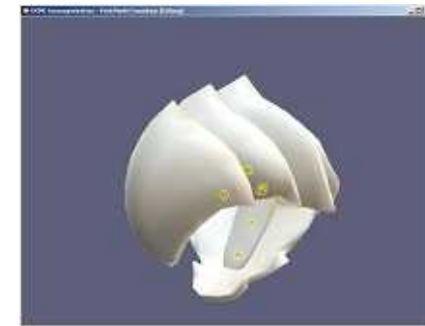
ハンドル



正式なモデル



ランタイム・モデル
パラメーターに沿った
アニメーションを製作



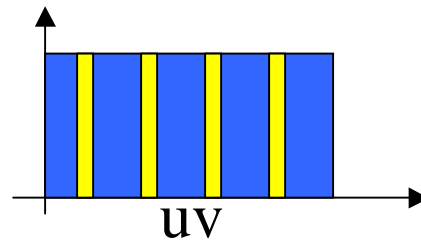
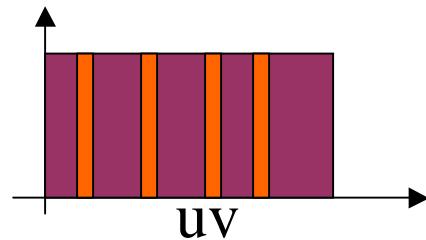
ユーザーはマウスで
コントロールポイントを動かすことで、
モデルを変形する(頂点を直接操作
しているわけではない)

(2) プロシージャル・テクスチャ生成

細かすぎない シンプルすぎない



ユーザーが選ぶ



Henry Goffin, Grue, Chris Hecker, Ocean Quigley, Shalin Shodhan, Andrew Willmott, Player-Driven Procedural Texturing, <http://www.cs.cmu.edu/~ajw/s2007>

Spore
最難関

(3) プロシージャル・アニメーション



モデルを作る

自動化
(プロシージャル・
アニメーション)



自由に動く

ユーザーが作ったどんなモデルに対してもアニメーションさせたい

どうすればできるだろう？

普通は、3Dアニメーターがツール上でキーフレームでアニメーションを作っていくよなー。

その作業手順のデータをそのままデータ化すれば、
アニメーションを割り当てることのできるのでは？

Spore アニメーション生成技術

キャラクターが作成したクリーチャーの形状にあわせて、アニメーションデータを再作成(re-targeting)する。

アニメーターがGUIで部位を指定して、動きを指定した操作のデータを蓄積する
(足が何本で、どれくらい地面から離れていて...)

Generalize(一般化) (データ化)

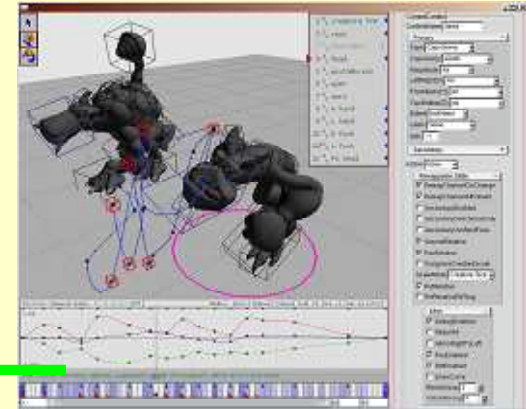
“G関数”

キャラクターの形状に依存しない一般空間のデータ

“S関数”

Specialize(再作成) (ゲーム内リアルタイム)

そのクリーチャーの持つ条件と似た条件を「アニメーターがアニメーションを作成したときの条件」から検索して、マッチした場合のアニメーションを少し変形してクリーチャーにフィットさせる



SIGGRAPH2008 “Real-time Motion Retargeting to Highly Varied User-Created Morphologies”

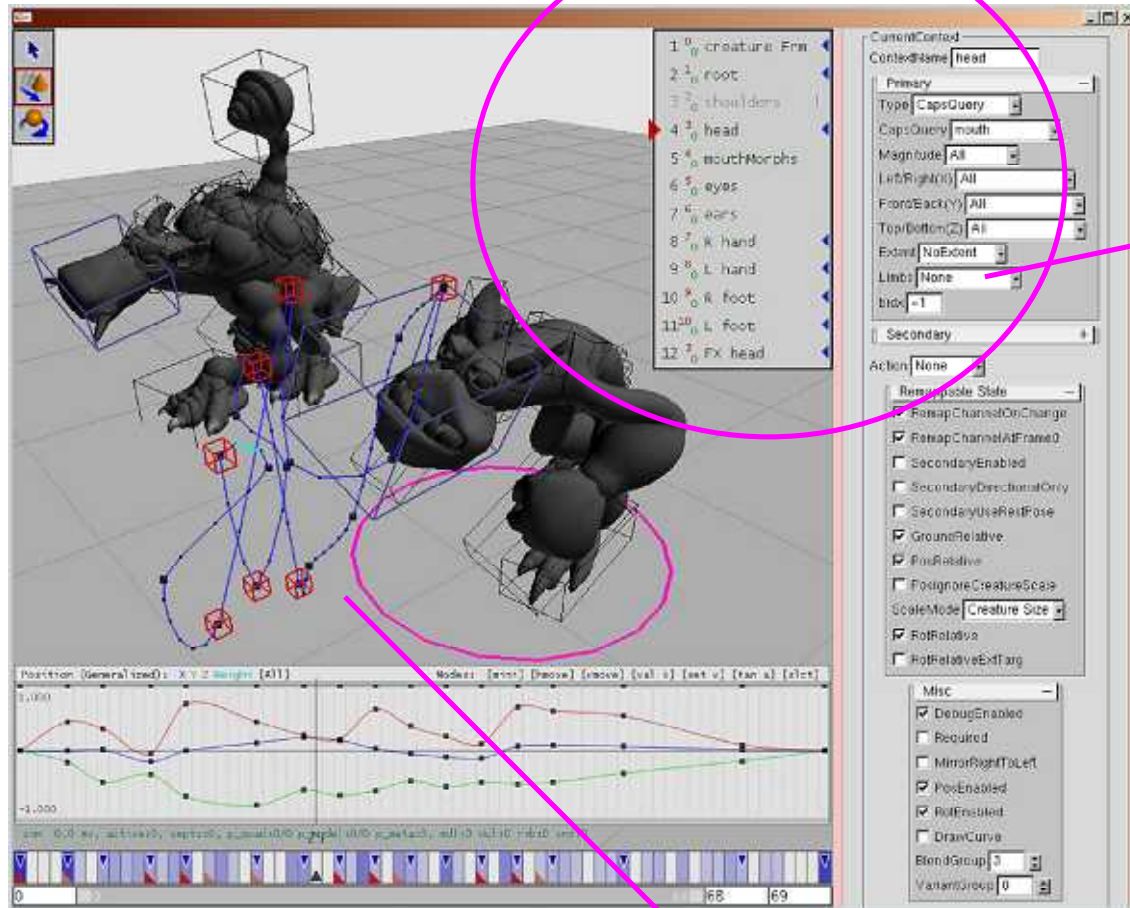
http://chrishecker.com/Real-time_Motion_Retargeting_to_Highly_Varied_User-Created_Morphologies

GDC2007

“How To Animate a Character You've Never Seen Before”

http://chrishecker.com/How_To_Animate_a_Character_You%27ve_Never_Seen_Before

Spore アニメーション・データ作成



部位を指定

アニメーションを指定

G関数生成のための情報をインプット

(3) プロシージャル・アニメーション



モデルを作る

自動化
(プロシージャル・
アニメーション)



自由に動く

キャラクターが作成したクリーチャーの形状にあわせて、アニメーションデータを再作成(re-targeting)する。

ユーザーから見たSpore エディター



たくさんの高度な
プロシージャル技術
(1)(2)(3)

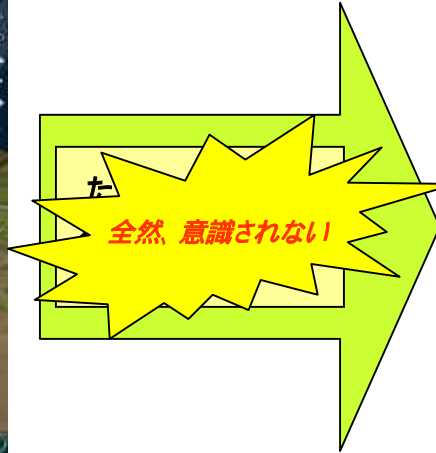


パーツをドラッグアンドドロップして、
パーツをモーフィング変形させて、
テクスチャーを選んで
モデルを作る

できたモデルはすぐにアニメーション！

プロシージャル技術によって、ユーザーに少ない労力で、
高度で大きなコンテンツを作成する力を与えることができる。

ユーザーから見たSpore エディター



パーツをドラッグアンドドロップして、
パーツをモーフィング変形させて、
テクスチャーを選んで
モデルを作る

できたモデルはすぐにアニメーション！

プロシージャル技術によって、ユーザーに少ない労力で、
高度で大きなコンテンツを作成する力を与えることができる。

これ以外にも、
実は *Spore* はプロシージャル技術の結晶！

「Spore」はプロシージャル技術の結晶

グラフィック、オブジェクト、サウンド、あらゆる分野においてプロシージャル技術を応用する次世代のゲーム



テクスチャー (SIGGRAPH2007)

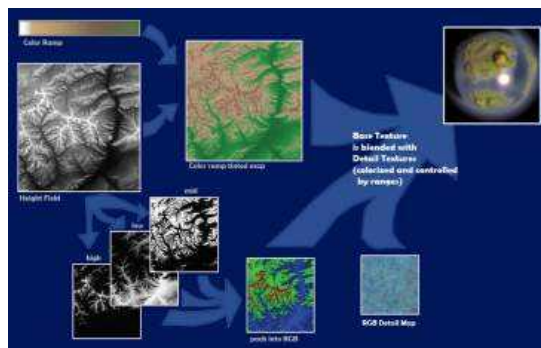
オブジェクト配置 (SIGGRAPH2007)

モデル生成 (SIGGRAPH2007)

モデル変形 (SIGGRAPH2007)

アニメーション生成 (SIGGRAPH2008)

音楽プロシージャル (GDC2008初公開)



Spore におけるプロシージャル技術

UGC (User Generated Contents, ユーザー作成コンテンツ) に
PCG (Procedural Contents Generation, 自動生成技術) を
応用することで、ユーザーにコンテンツを産出す力を与え、
小さな労力で、大きなコンテンツを作り出す面白さを実現した。

References

- (1) カーネギー・メロン大学の [Andrew J. Willmot](http://www.cs.cmu.edu/~ajw/s2007/) 博士のHPページのSIIGRAPH2007のコーナーにまとめられています。
<http://www.cs.cmu.edu/~ajw/s2007/>
- (2) [E3 2006 # 011] E3 2006最大の話題作？ ウィル・ライトの「Spore」
<http://www.4gamer.net/news/history/2006.05/20060511195155detail.html>
- (3) Steve Capell et al., Interactive Skeleton-Driven Dynamic Deformations
<http://grail.cs.washington.edu/projects/deformation>
- (4) Development of Spore
(このサイトによると、デモシーナーたち、著名なフラクタル関係の技術者を Maxis は集めているようです)
http://en.wikipedia.org/wiki/Development_of_Spore
- (5) Spore の Procedural Music については、WEB に動画などがアップされているので、[Spore Procedural Music](#) などで検索してみよう！

Figures on the pages are from these references.

可能性空間

リアル

ゲーム空間

ユーザー

可能性空間

プロシージャル技術は、大規模、複雑化するゲームにおいて、スケールと複雑さにあったユーザー・アクションをユーザーに与えることができる。

可能性空間を拡張することができる



第2部

第1章

ゲーム世界(レベルデザイン)とプロシージャル

第2章

キャラクター(NPC, AI)システムとプロシージャル

第3章

ユーザー・アクションとプロシージャル

第4章

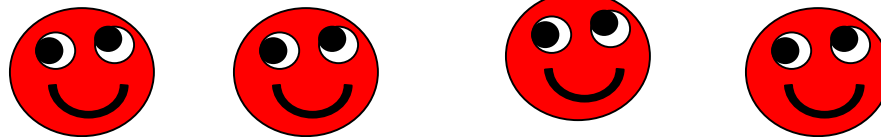
ゲームシステムとプロシージャル

第5章

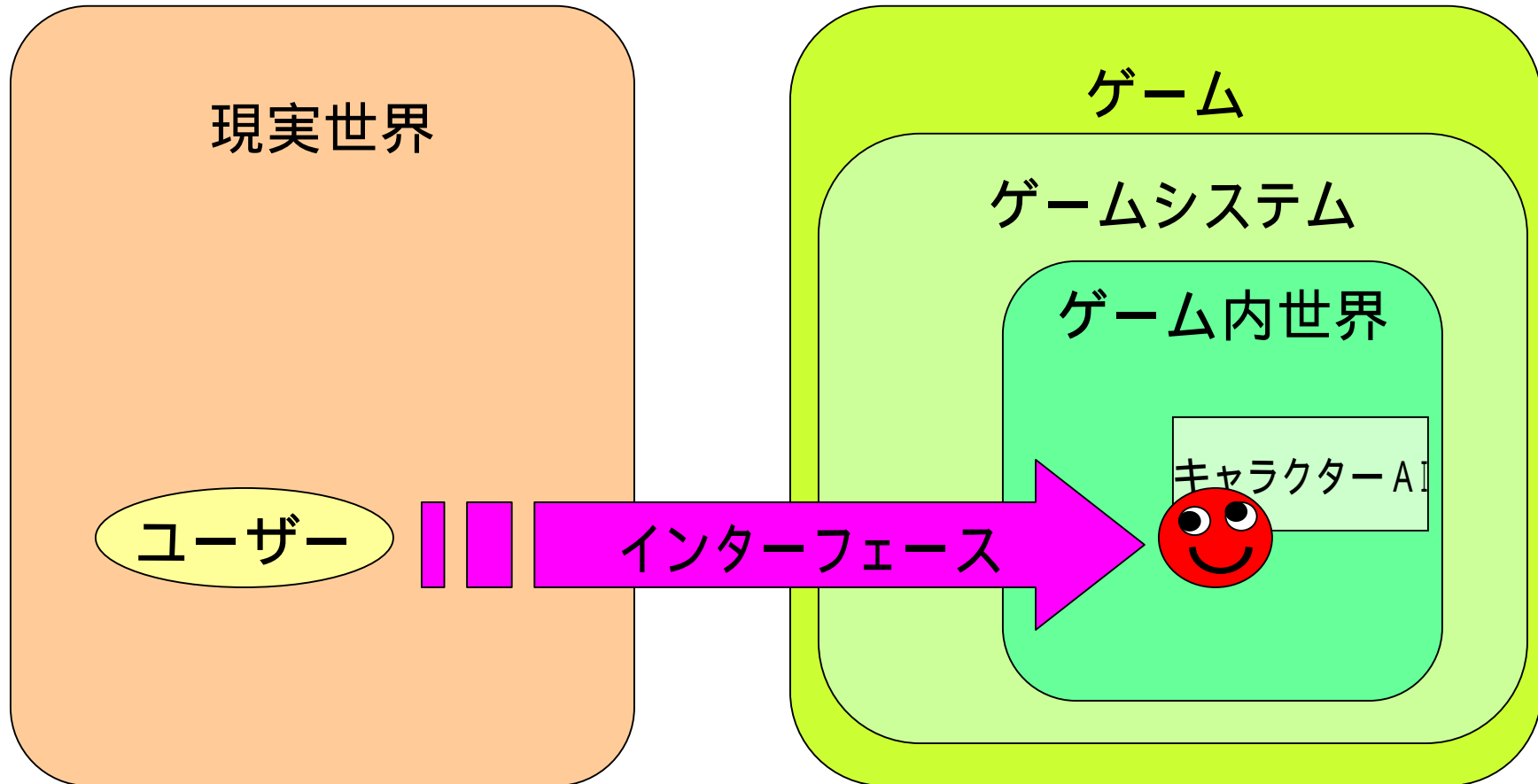
まとめ

第4章

ゲームシステムとプロシージャル

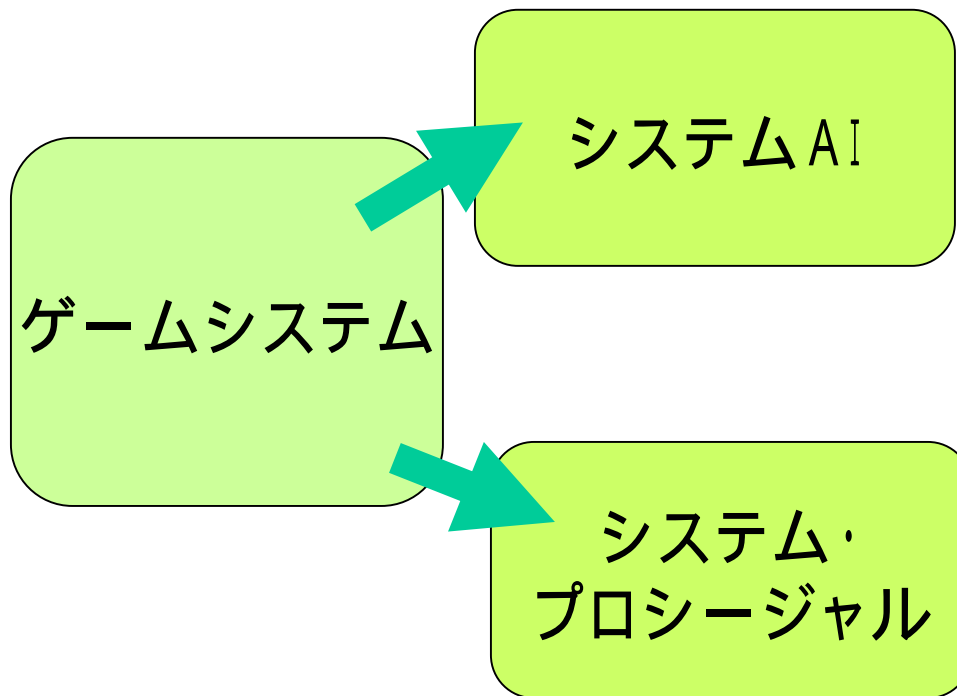


ゲームシステム



ゲームシステム = ユーザーとゲーム世界の間で、ゲームを進行・調整する。

ゲームシステムAI, プロシージャル



インターフェースコントロール
レベルコントロール
ユーザー解析

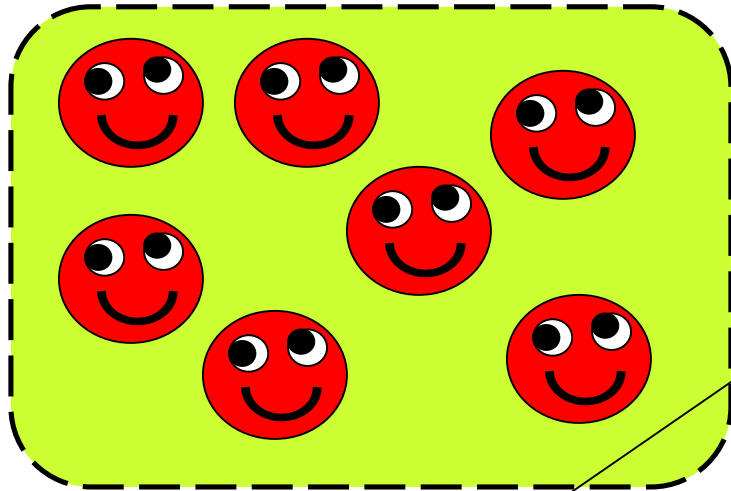
ゲームの進行を監視しながら、
ユーザーを楽しませるために、
ゲームの進行をコントロールする
システム

前置きとして、
遺伝的アルゴリズムを解説する

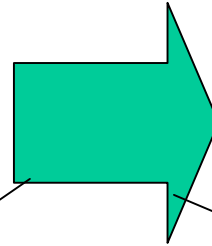
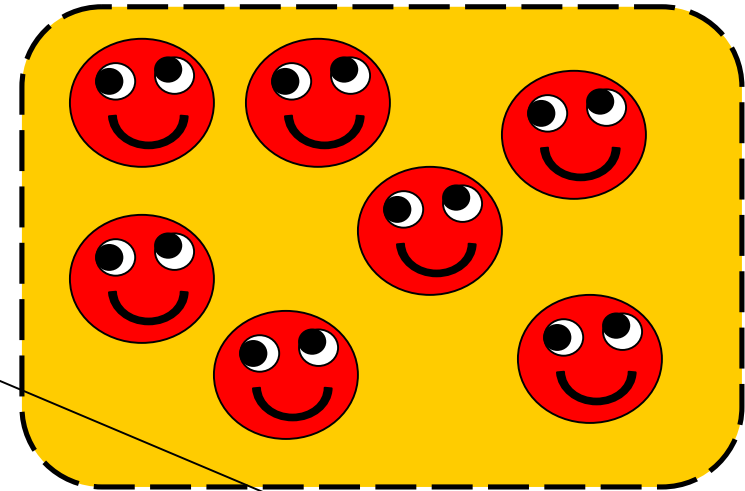
遺伝的アルゴリズム

集団を一定の方向に進化させる方法

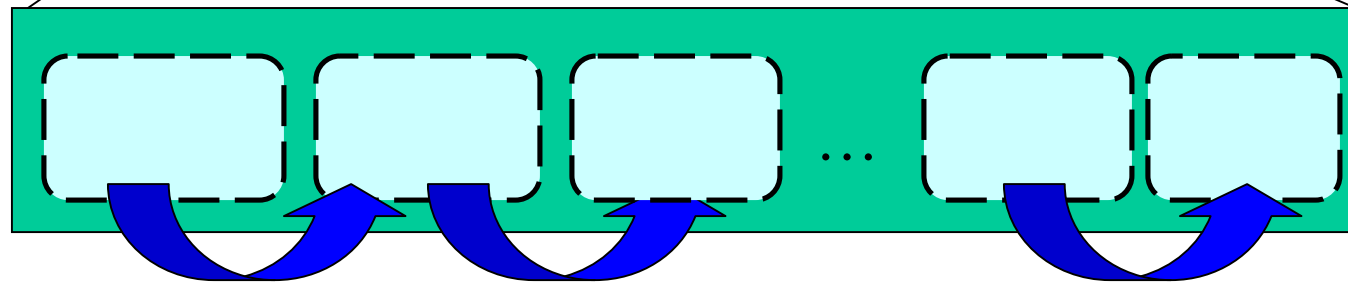
最初の世代



新世代(100~世代後)

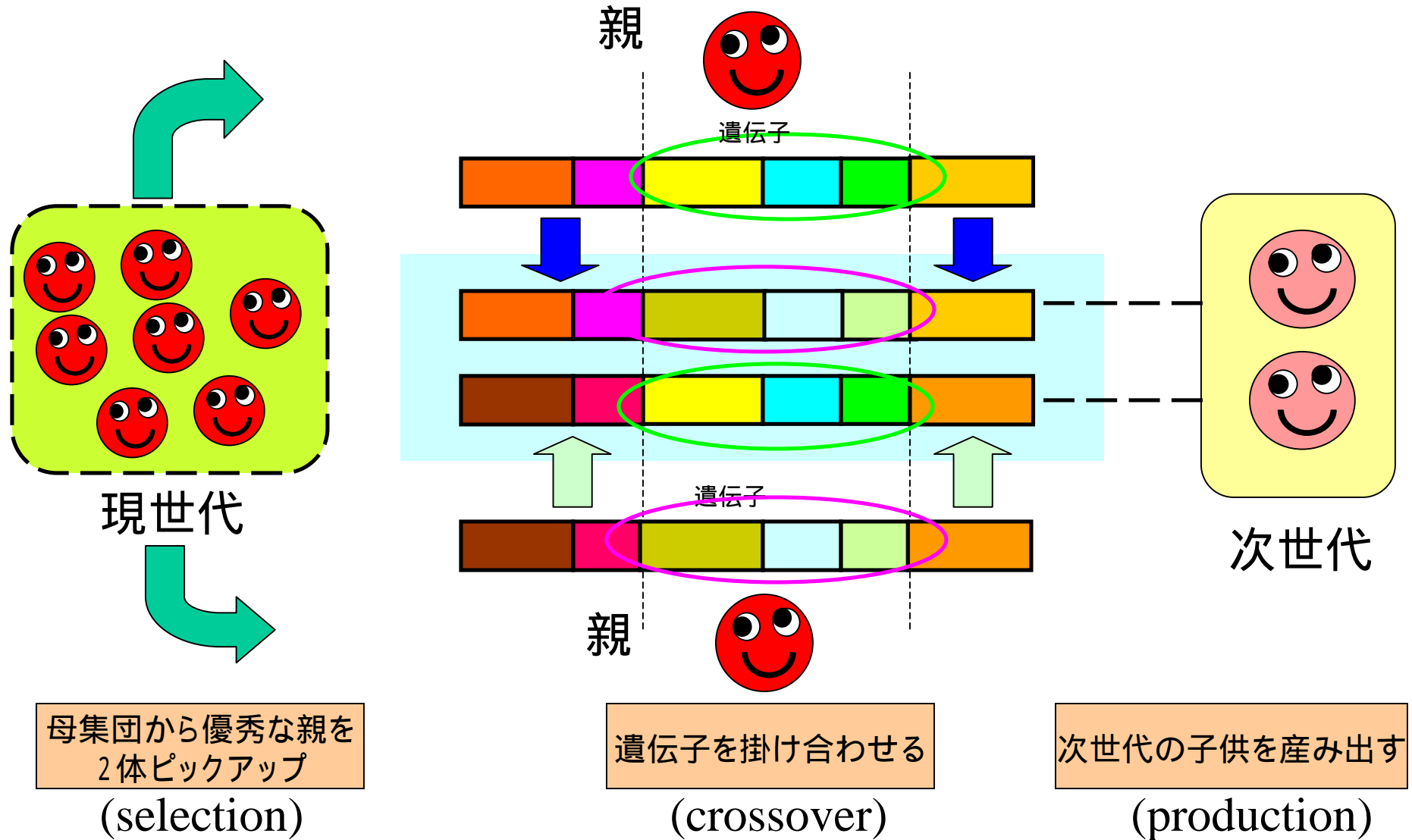


世代を経て進化させる



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

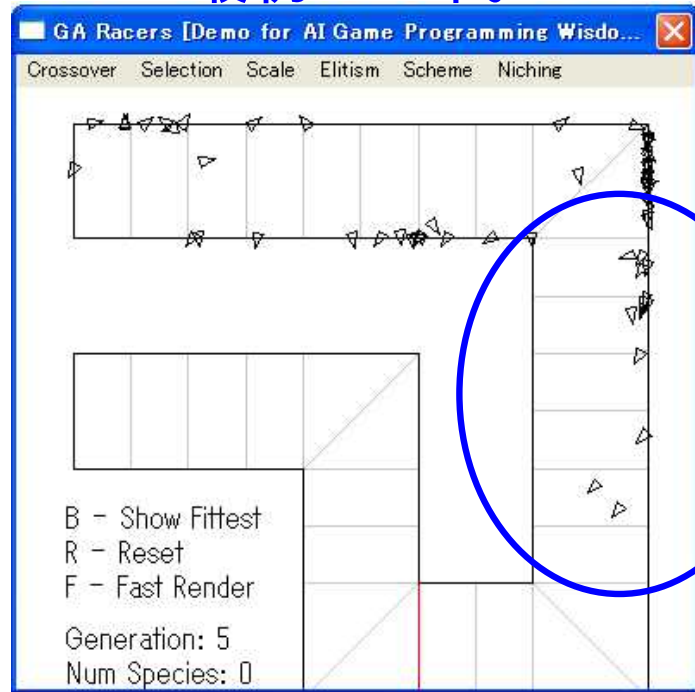


このサイクルをくり返すことで世代を進めて望ましい集団を産み出す

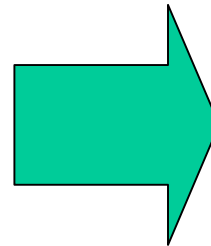
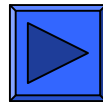
(例) GA Racer

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。

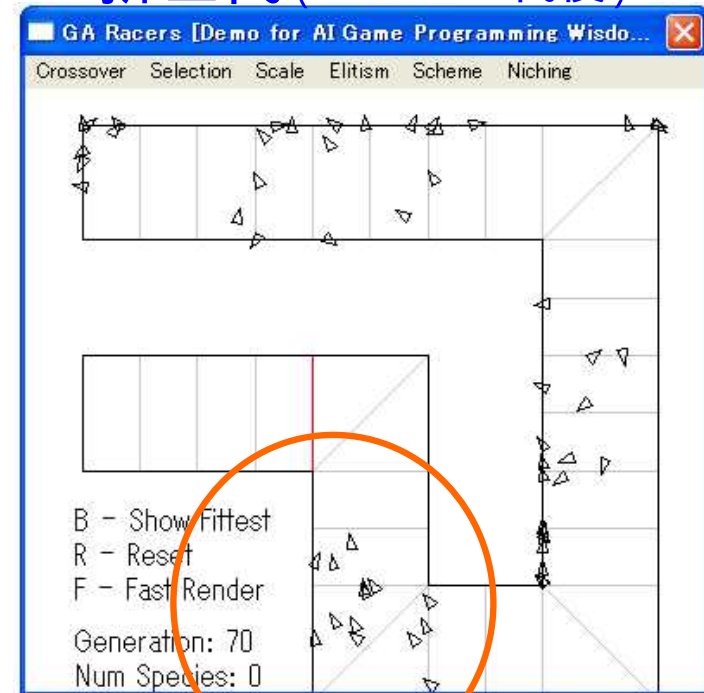
最初の世代



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。

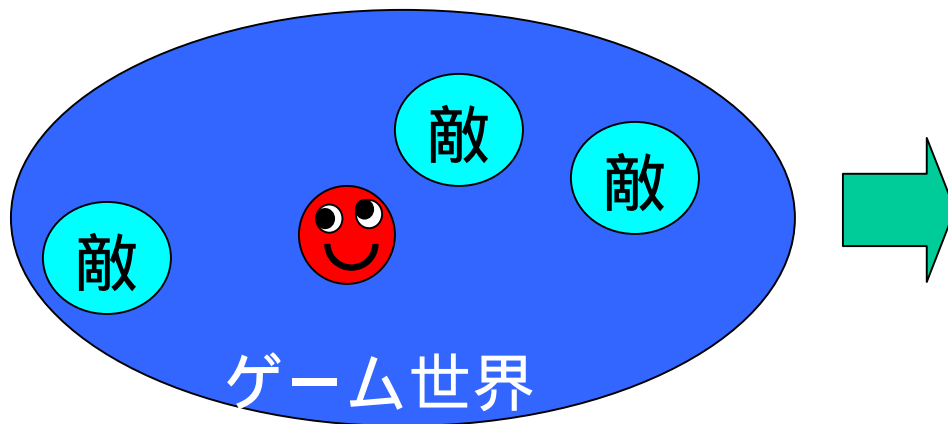
Mat Buckland, "Building Better Genetic Algorithm", 11.4., AI Game Programming Wisdom 2
(CD-ROMにソースコードと実行ファイルがあります)

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。

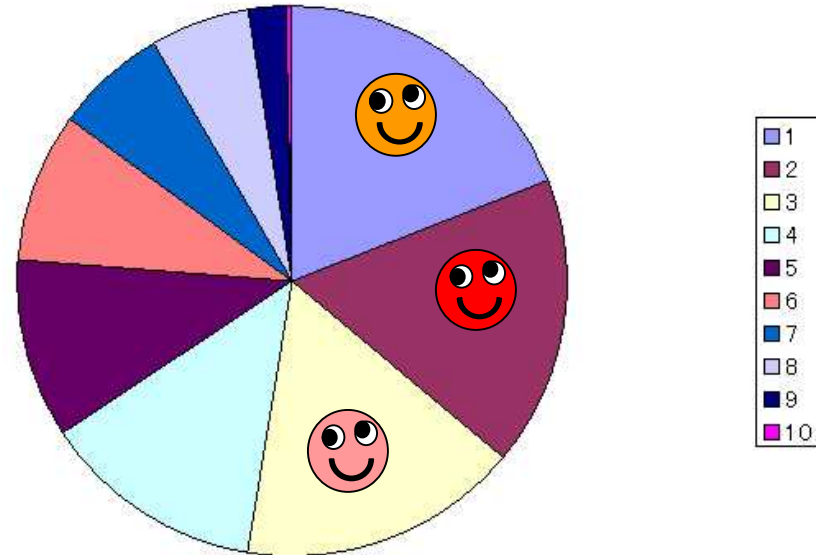
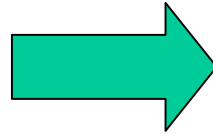
(例) 同じでいいや。 $適応度 = 評価値$

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

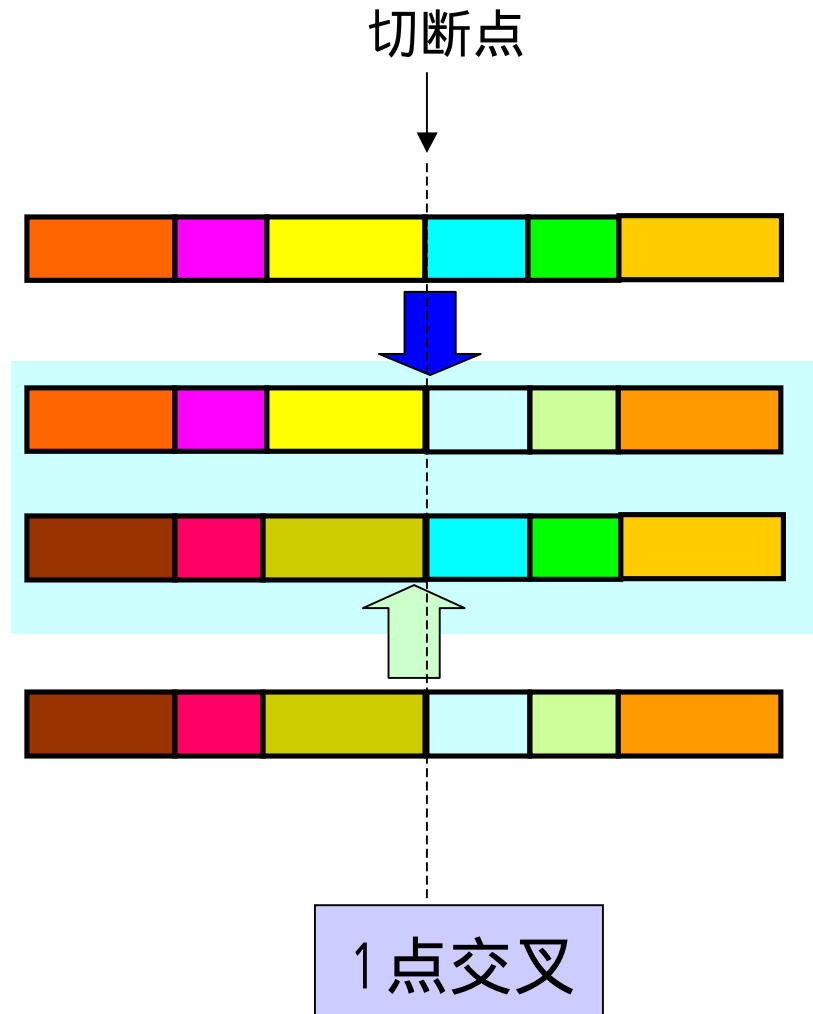
... 適応度の大きさに比例した確率で生き延びて親になれる。

(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)

交叉による次世代生成

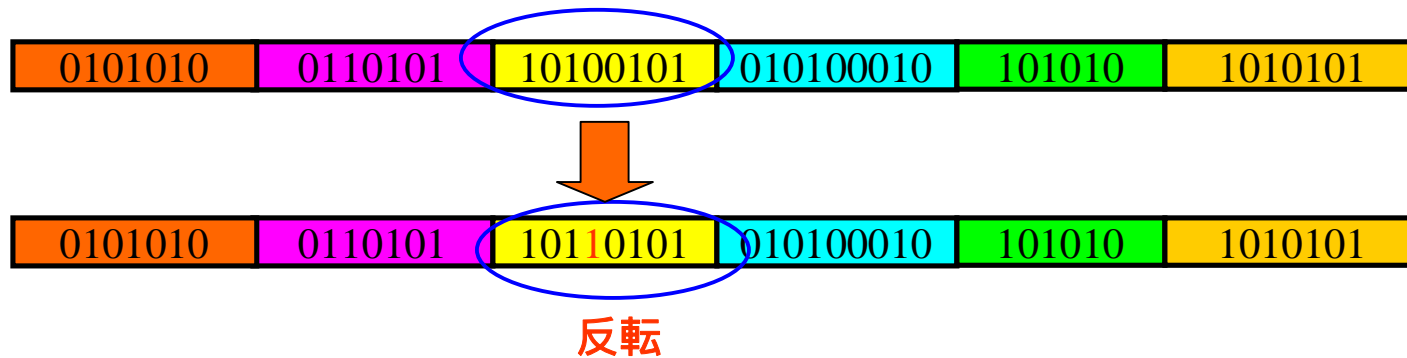
選んだ2つの親の遺伝子を交叉(crossover)させる。



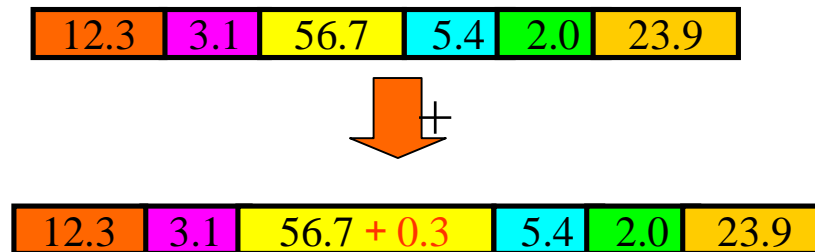
遺伝子操作 (突然変異)

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

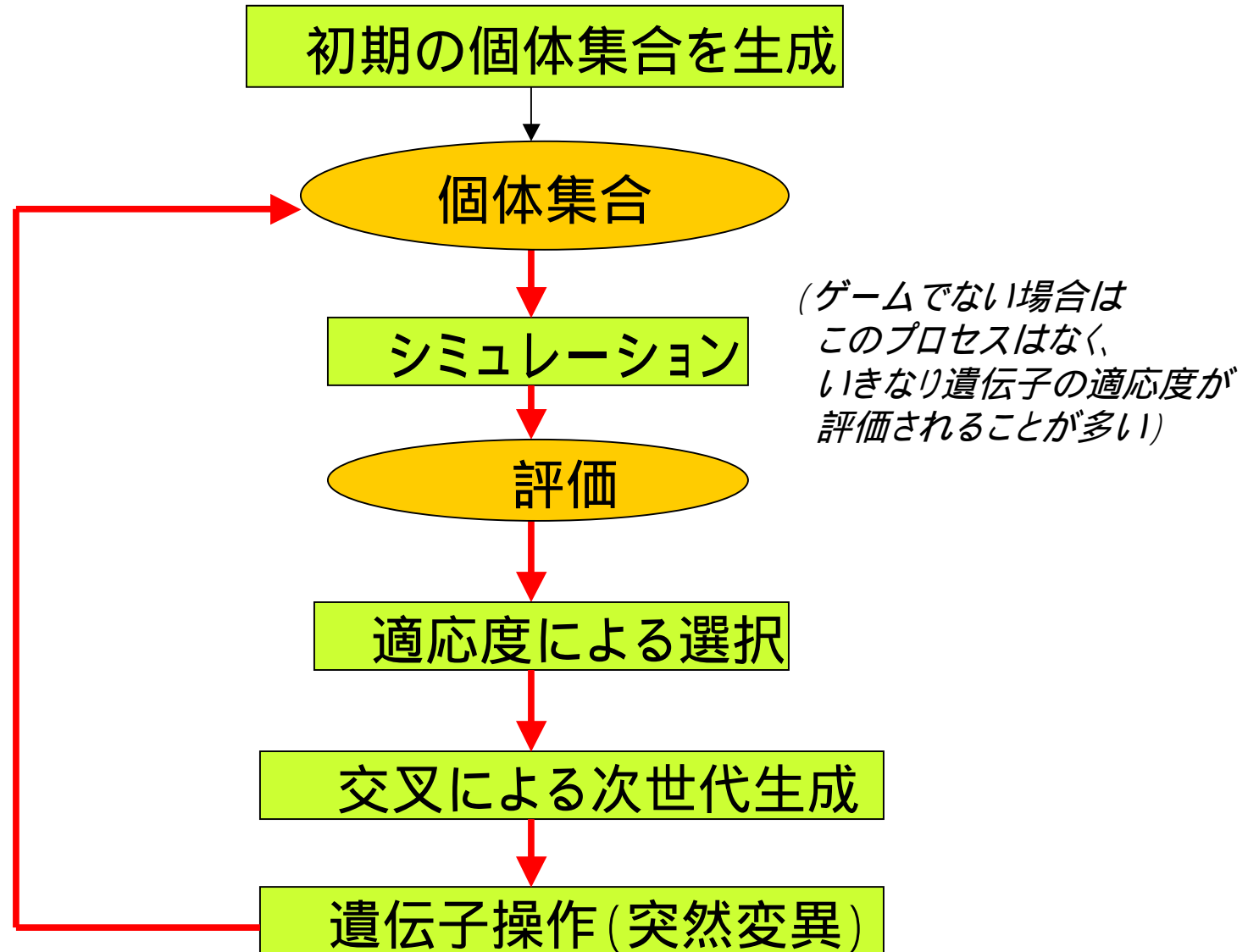


実数表現



遺伝子に多様性を与える

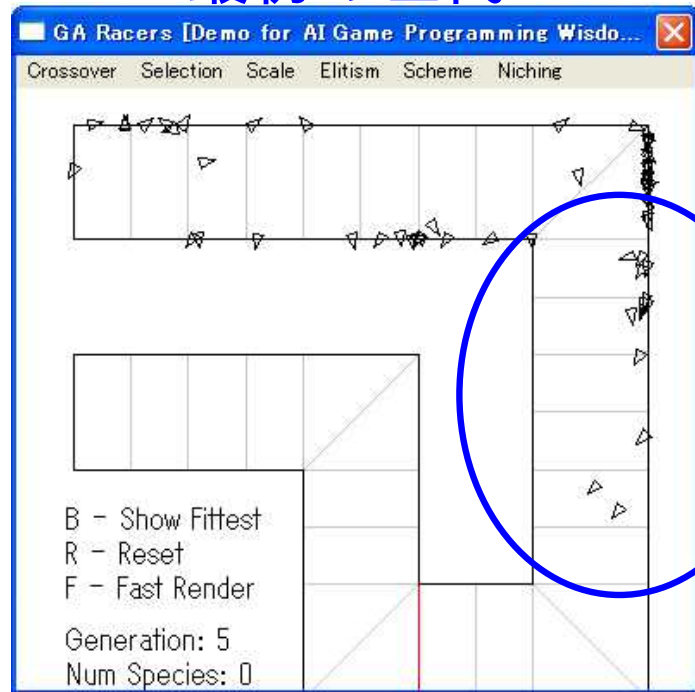
このプロセスを何度もくり返すことでNPCの集合は進化します



(例) GA Racer

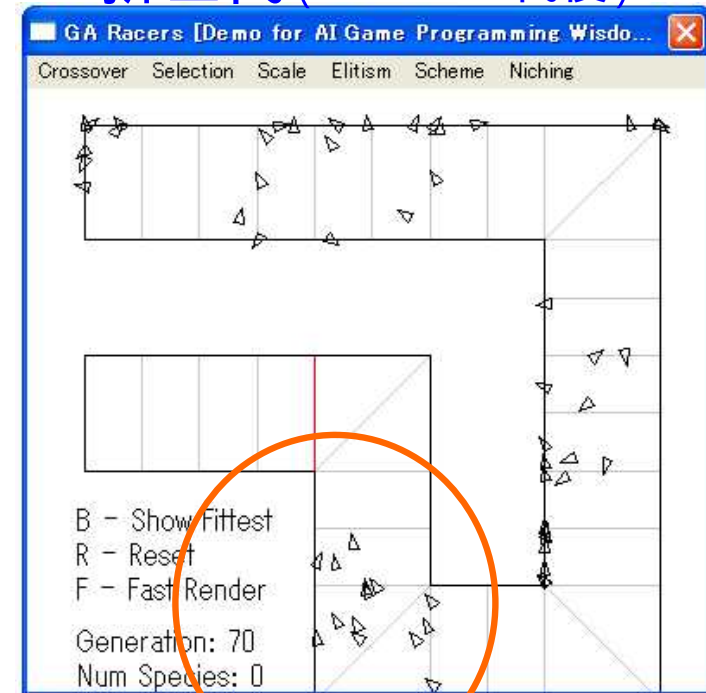
遺伝的アルゴリズムによって、遠くまで到達できるレーサーを作成する。

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)

しかし、これはNPCの集団に
遺伝的アルゴリズムを組み込んだだけ

AIとしての技術
ゲームシステムではない。

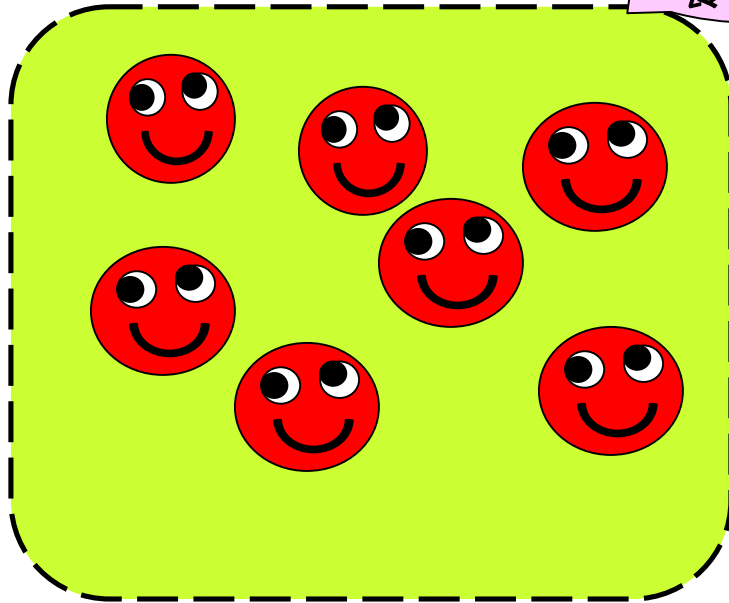


世界最高峰の遺伝的アルゴリズムを使ったゲーム
(AIをどうゲームに使うか、という手本のようなゲーム)

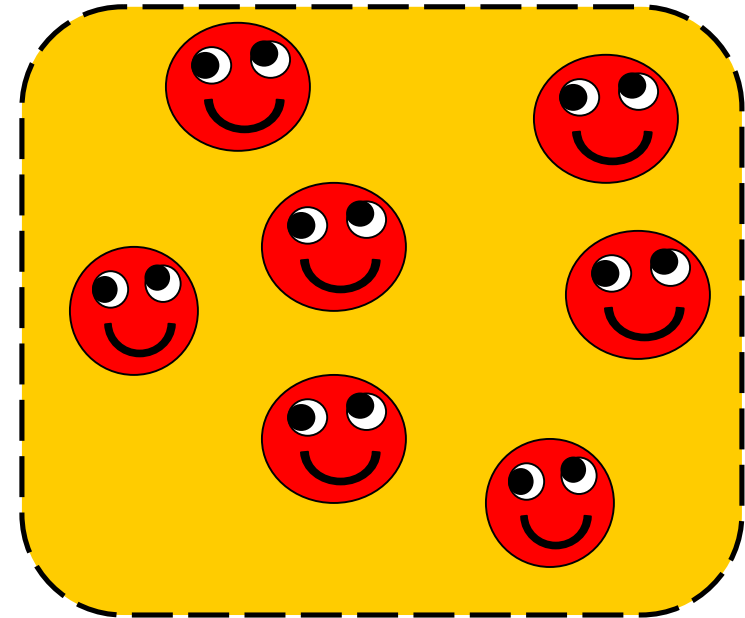
(例) アストロノーカ

最初の世代

野菜
食べたい



新世代 (5 ~ 世代後)



最初はすぐに罠にかかるけど

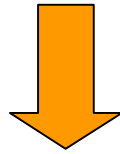
だんだんと罠にかからないようになる

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

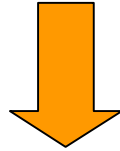
<http://www.muumu.com/games/astro/>

どういうゲーム？

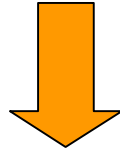
珍しい野菜を育てる



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/>

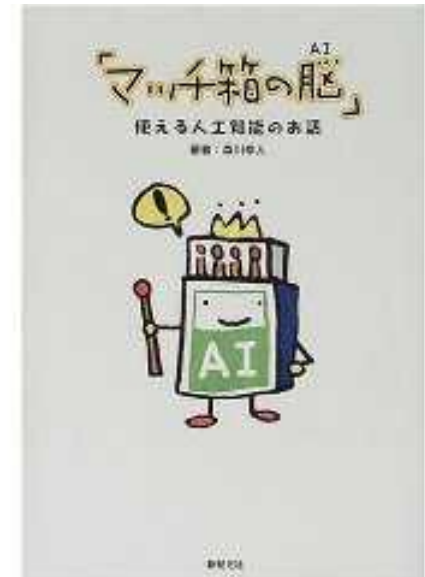
以下の解説は

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>

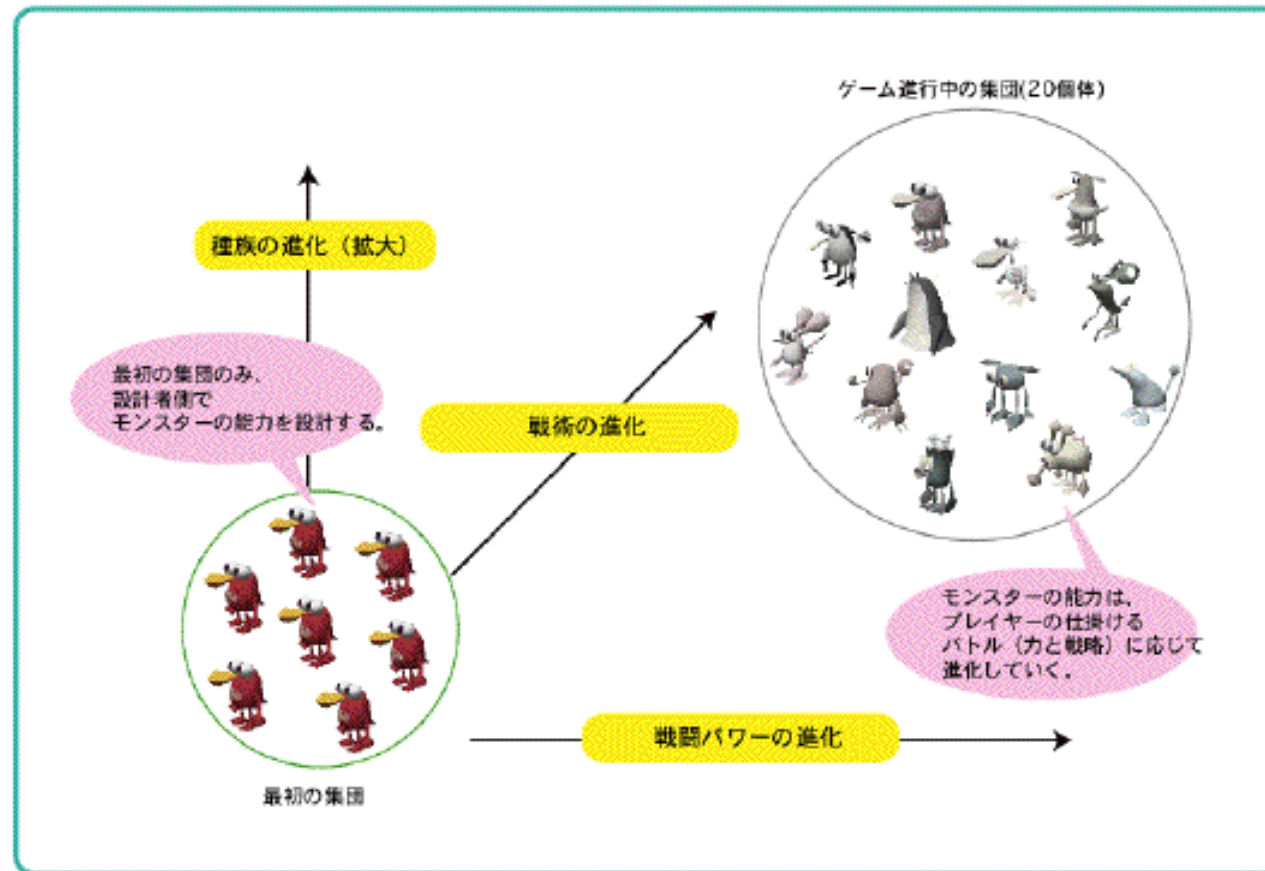
に準拠します。

詳細は以下の資料へ

CEDEC2008 AI day 講演 2日目(3コマ目)

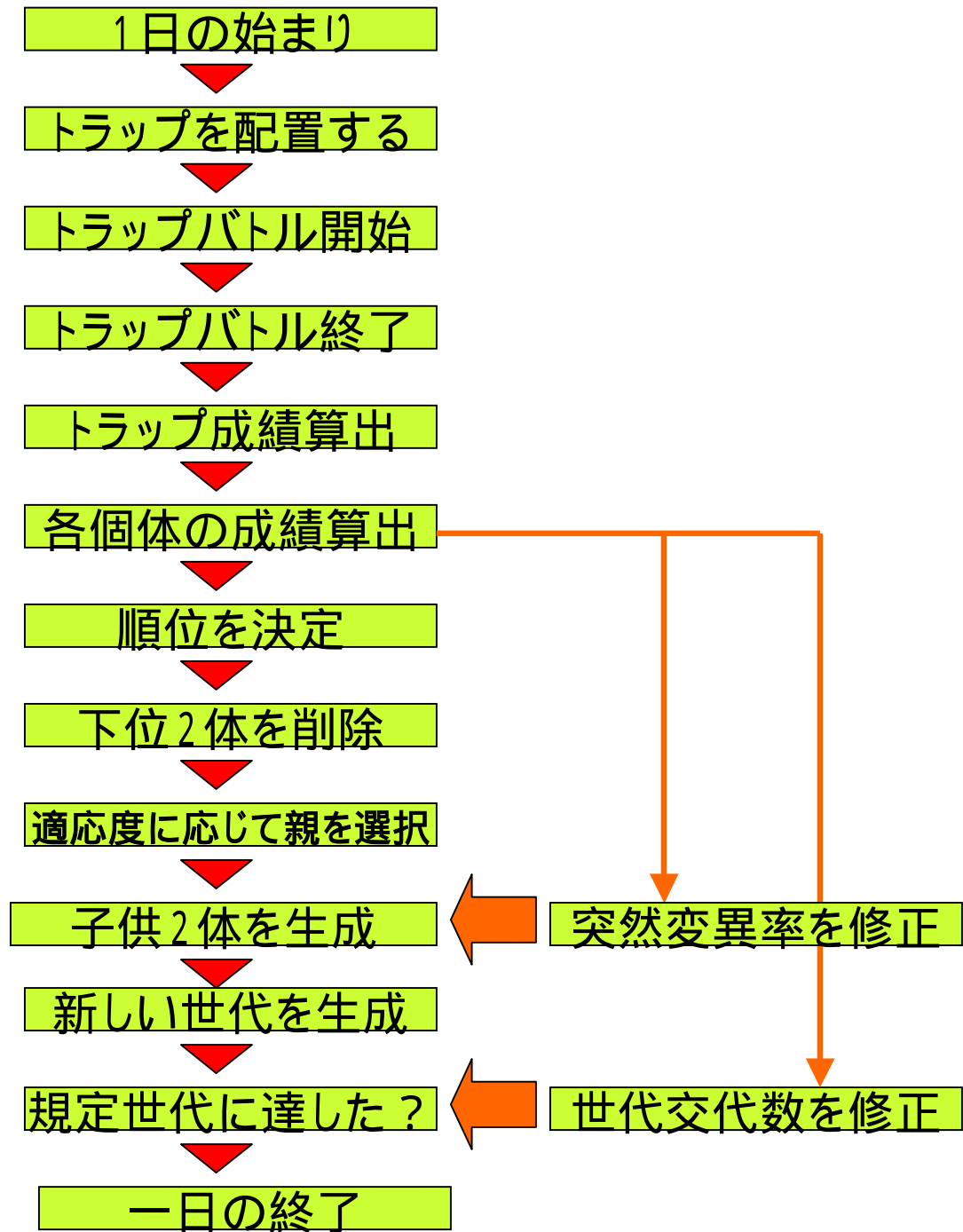


全体の流れ



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.muumuu.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]

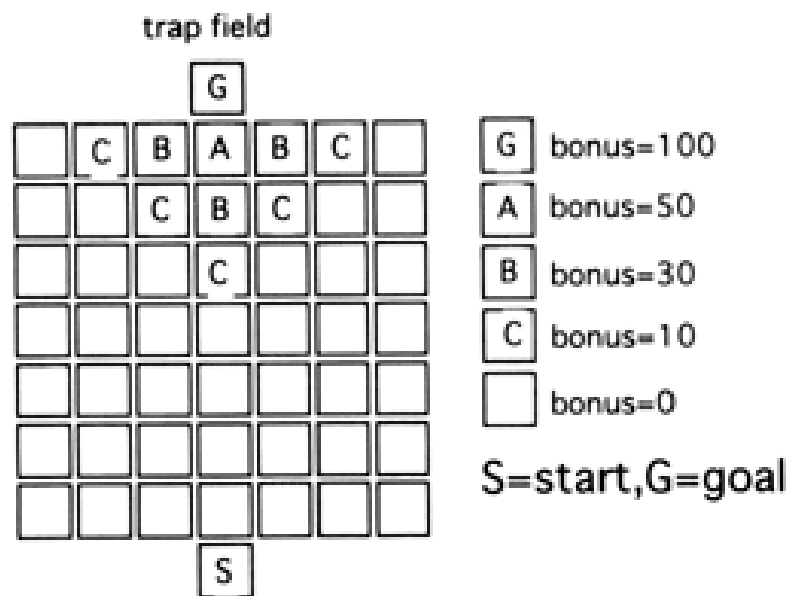
1.87	6.85	16.25	25.03	25.03	16.25	6.85	1.87
------	------	-------	-------	-------	-------	------	------

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。



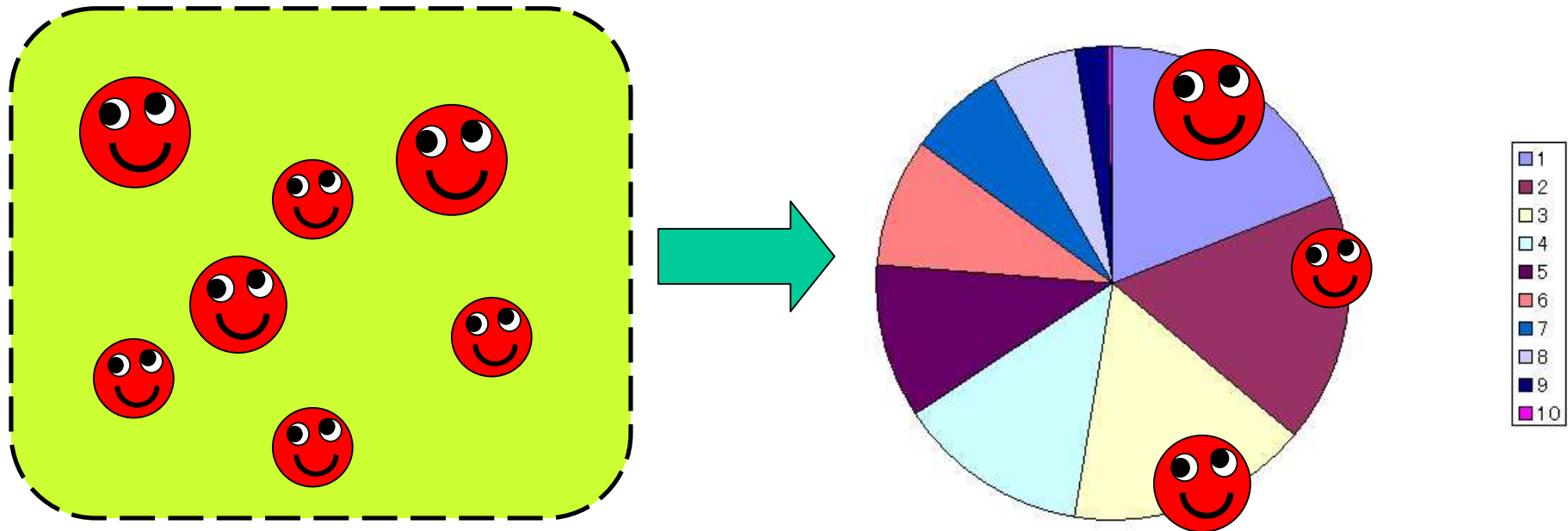
適応度 = 成績 + TB時間*0.3 + エンジョイ*0.5 + トラップ点 + 安全点 + HP*0.5

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

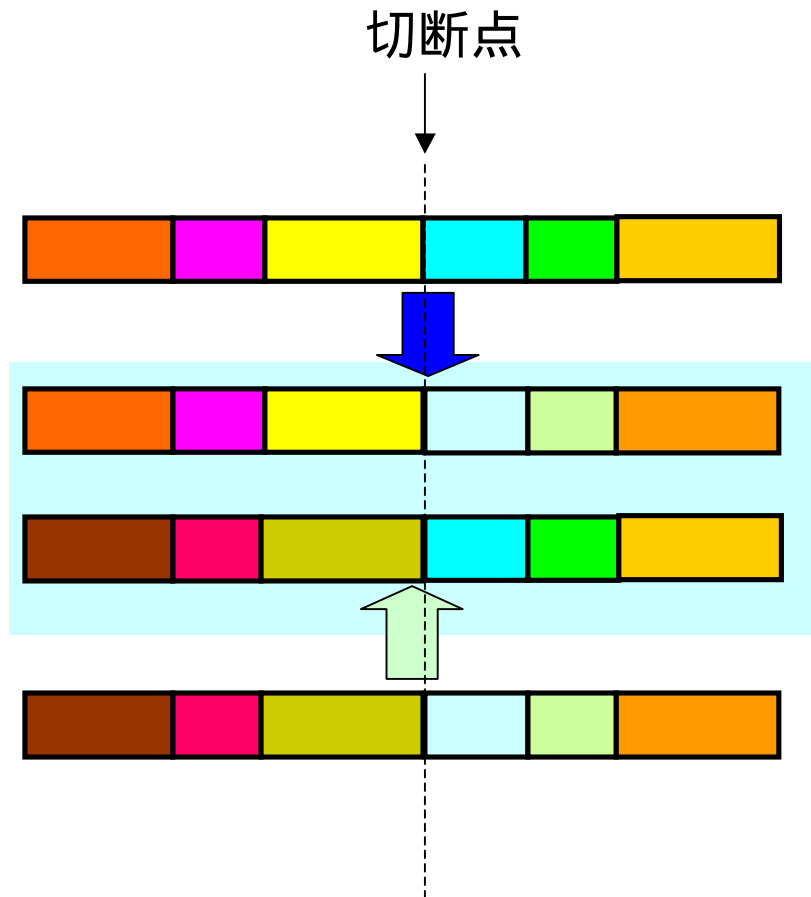


適応度比例方式 (ルーレット選択)

... 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。



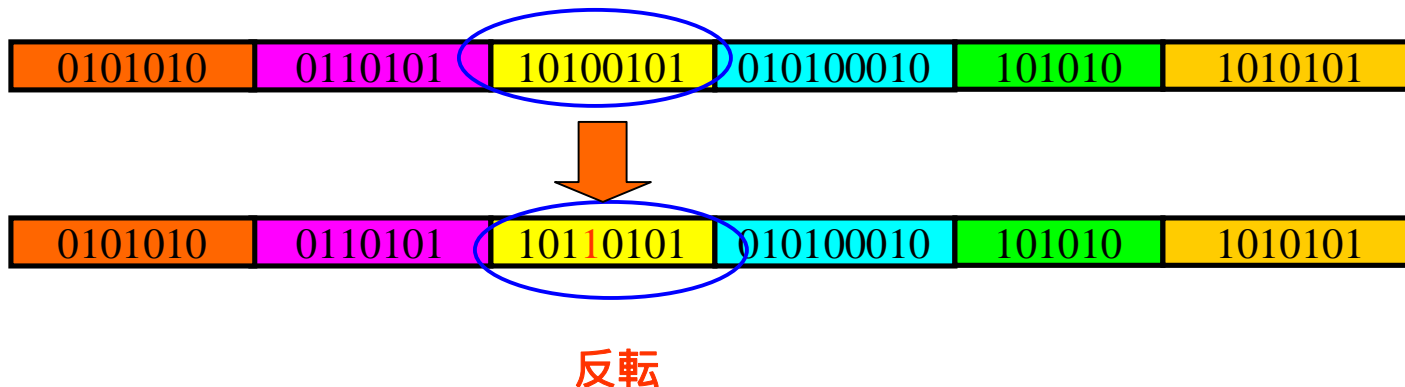
(詳しい交叉の情報はわかりません)

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

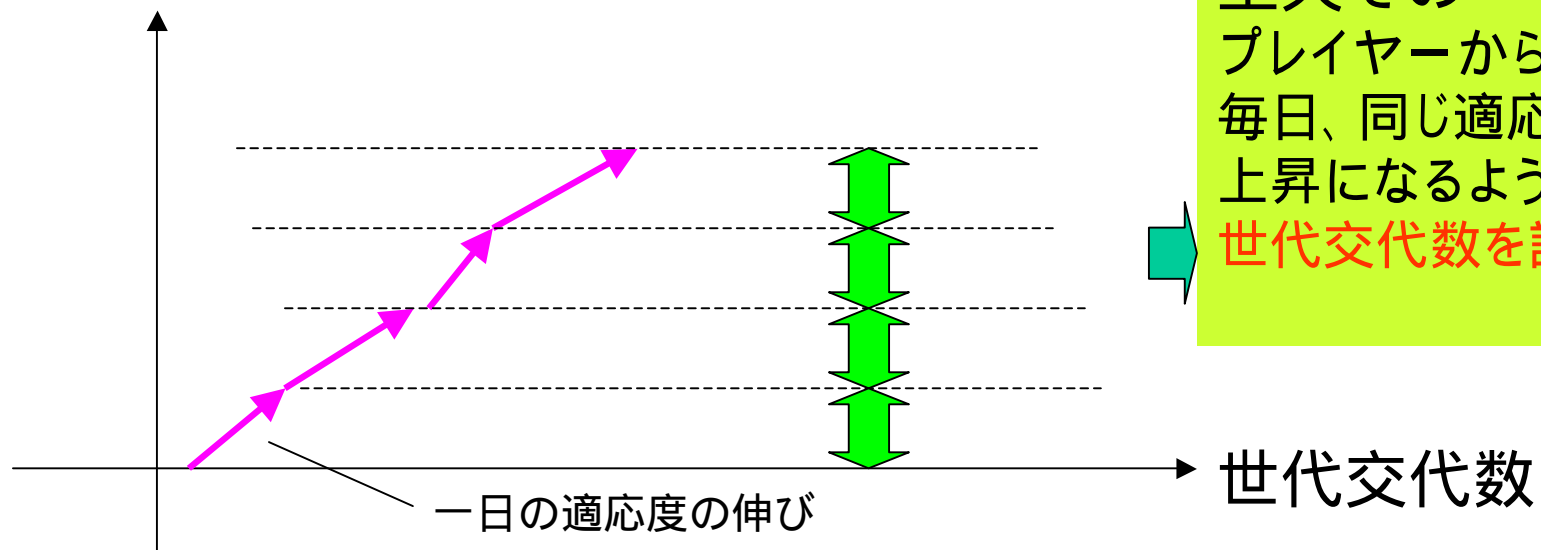
距離が小さい = 遺伝子が似ている
距離が大きい = 遺伝子が異なる

ゲームシステムとしての工夫

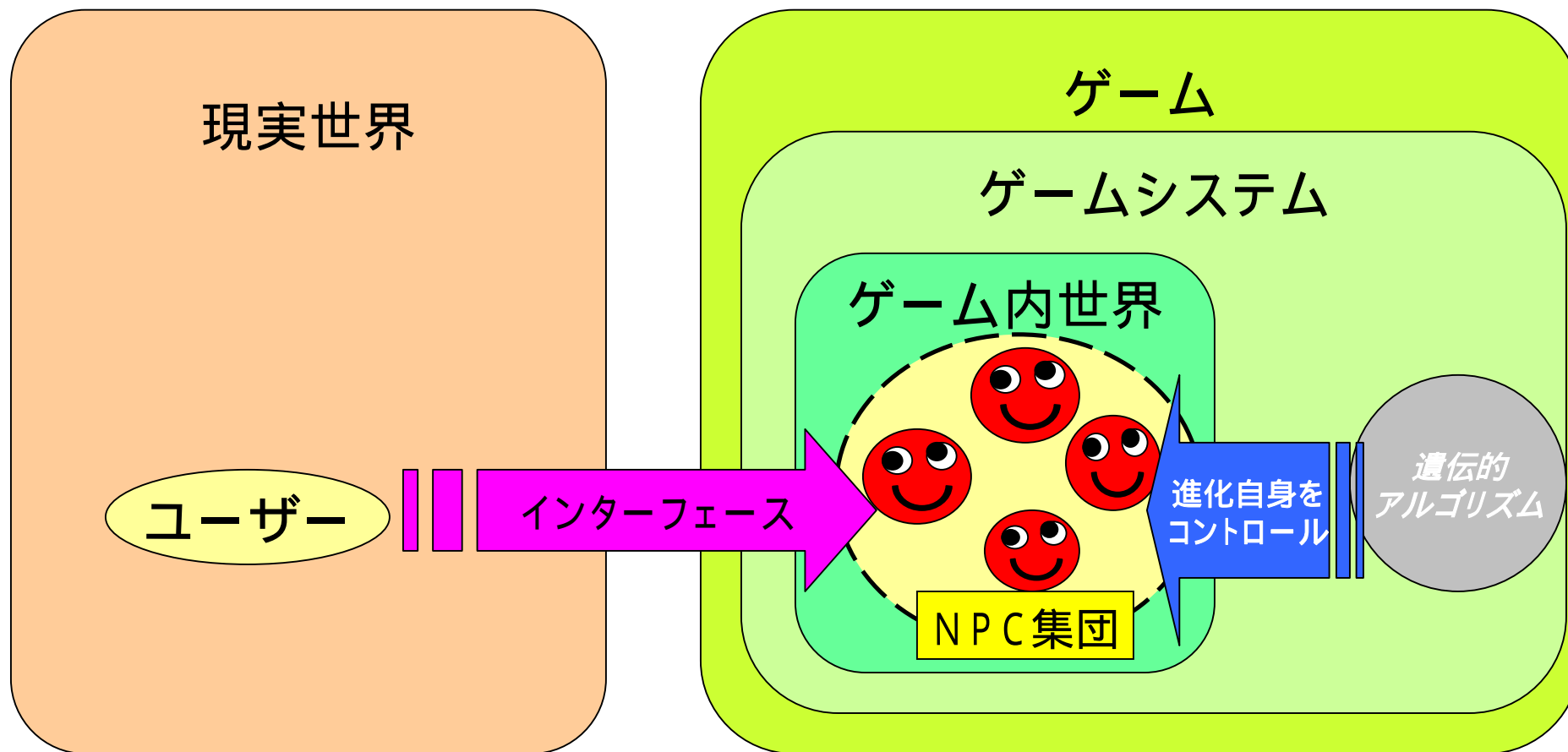
工夫その 遺伝的アルゴリズムは集団に対するアルゴリズム
→ 一体のトラップバトルの裏で他の20体も同じトラップバトルをして、
全体として世代交代をさせている。

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



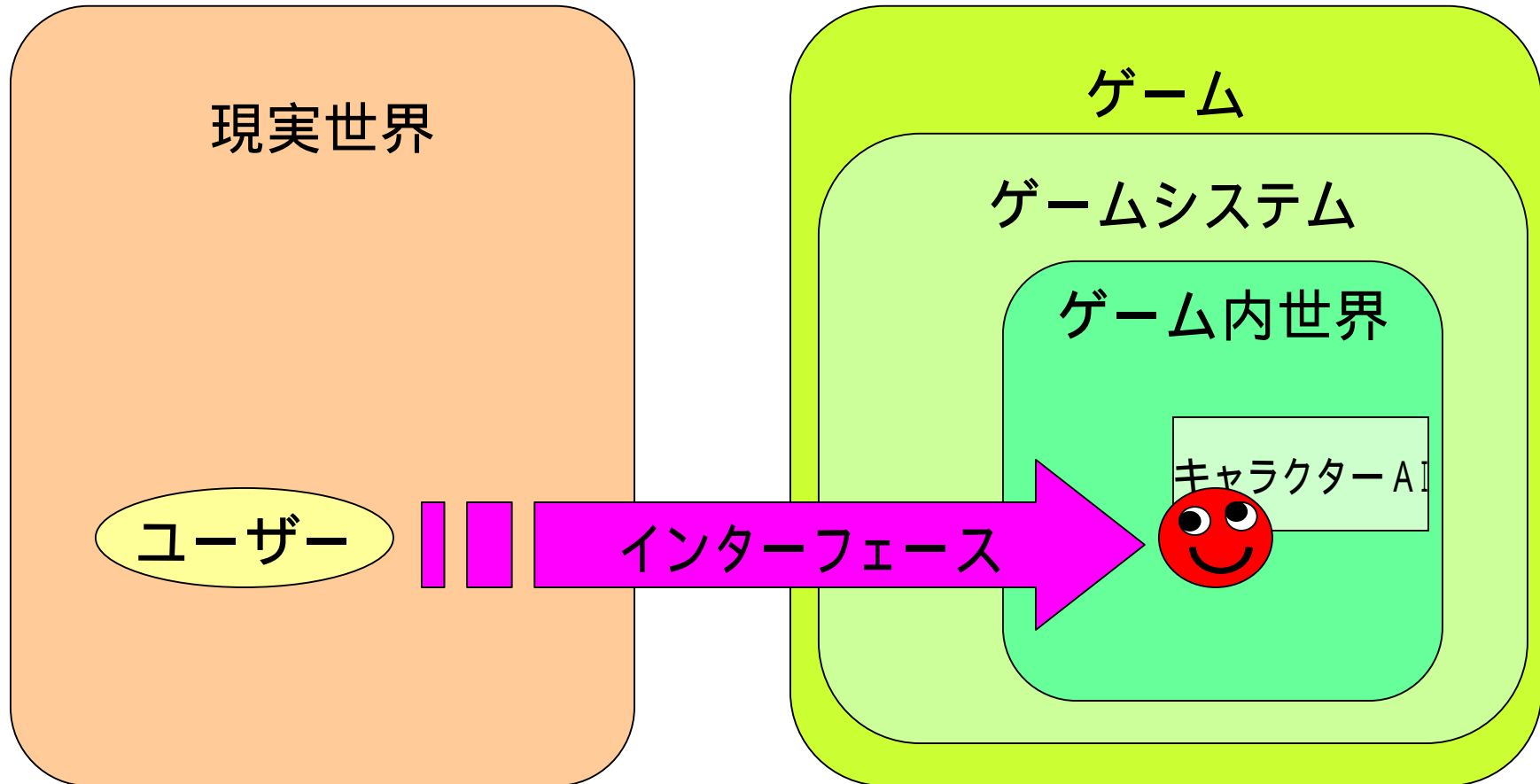
アストロノーカにおけるプロシージャル技術



アストロノーカは、NPCの集団に遺伝的アルゴリズムを組み込み、ユーザーのアクションに応じて進化する仕組みを作った上で、ゲームデザイン的な要請から、進化の速度が一定になるように、遺伝的アルゴリズムを調整して走らせている。

高度に調整された数学的ダイナミクスの上にゲーム性を成り立たせている
今でもなお先駆的なゲーム。そして傑作。

ゲームシステム・プロシージャル



ゲームシステムにおけるプロシージャル技術こそは、
様々なユーザーごとにゲームをコントロールすることを可能にする。
= ユーザー自身の性質を含んだゲームデザインを可能にする技術である

第2部

第1章

ゲーム世界(レベルデザイン)とプロシージャル

第2章

キャラクター(NPC, AI)システムとプロシージャル

第3章

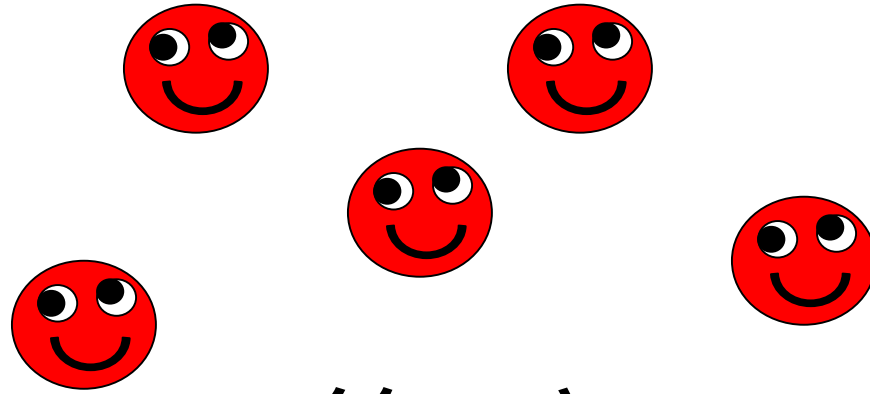
ユーザー・アクションとプロシージャル

第4章

ゲームシステムとプロシージャル

第5章

まとめ

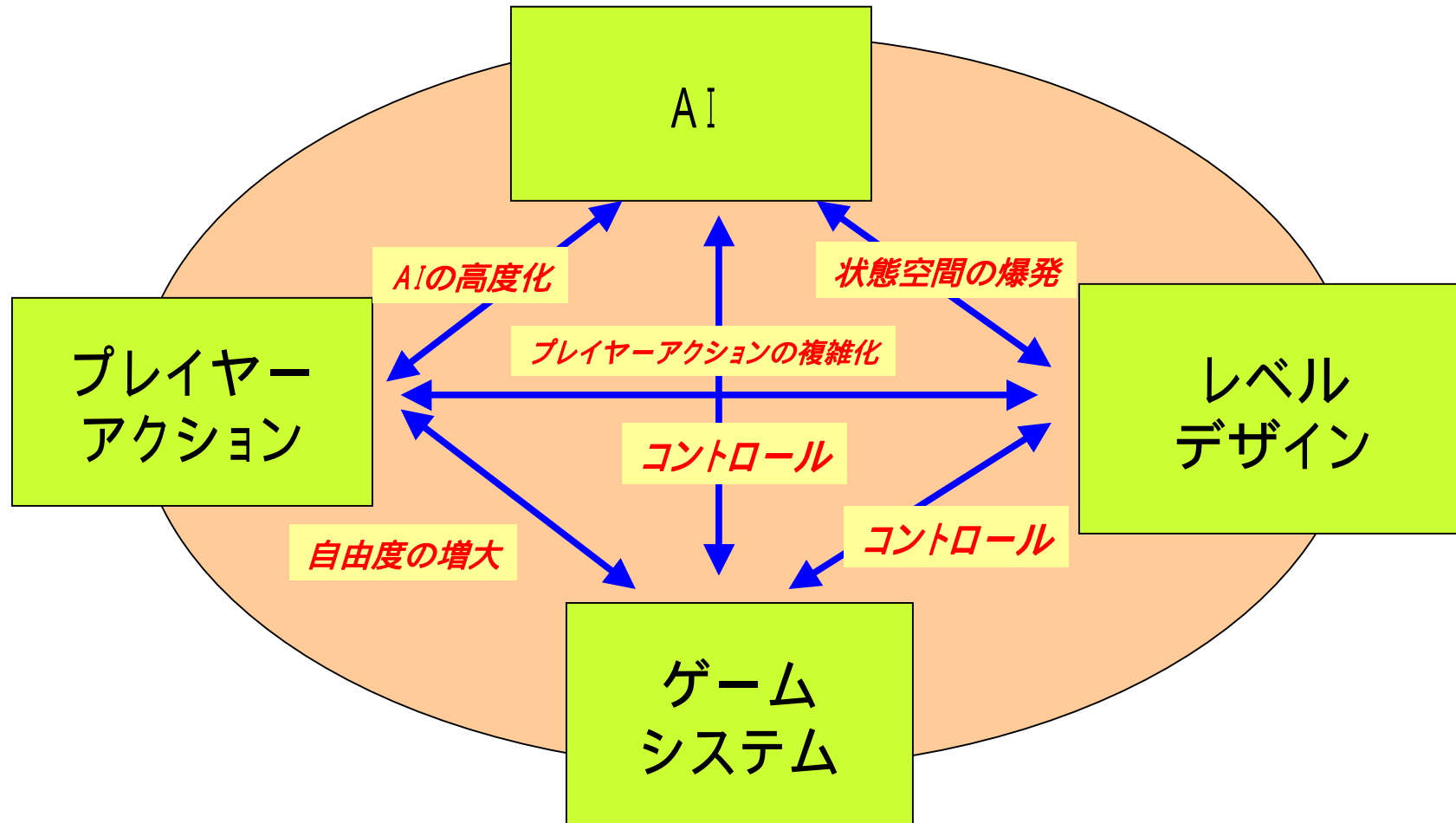


第5章

まとめ

ゲームデザインとプロシージャル

デジタルゲームの現状



プレイヤー、AI、レベルデザイン、ゲームシステム...
4つが互いに関連して(有機性)ゲームを成立させる

ちょっとここで、

ゲームデザインの用語をさらにもう一つ覚えましょう！

マジックサークル (Magic Circle)

Katie Salen, Eric Zimmerman, "Rules of Play: Game Design Fundamentals", MIT Press, 2003



ルールズ・オブ・プレイ

マジックサークル = ゲームという行為の空間

(例) パックマンのマジックサークル

... 「食べる、逃げる」という行為の総体とそれが可能な空間

スーパーマリオ

... 「走る、壊す、踏む」の組み合わせの行為の連続とそれが可能な空間

Although the **magic circle** is merely one of the example in Huuzinga's list of "play-grounds", the term is used here as shorthand for the idea of a special place in time and space created by a game.

...

In a vert basic sence, **the magic** circle of a game is where the game takes place. To play a game means entering into a magic circle, or perhaps creating one as a game begins.

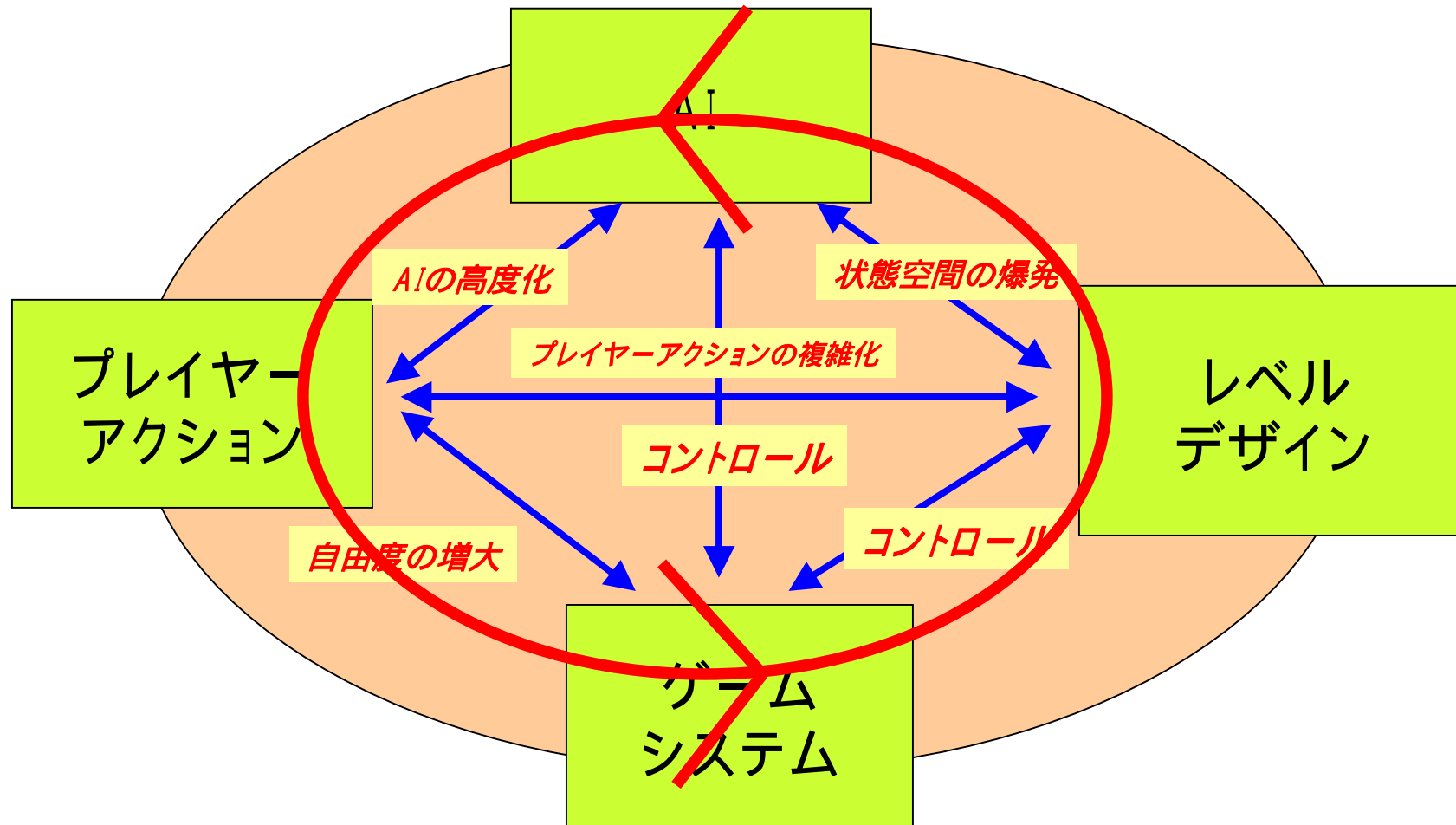
(Unit 1: Core Concepts | The magic circle)

マジックサークルはホイジンガーのプレイグラウンドのリストに含まれる単なる実例ではなく、この言葉ゲームによって作り出される**特別な時間と空間**という概念として端的に用いることにする。

...

基本的な意味として、ゲームにおける**マジックサークル**とはゲームが起こり得る場のことである。ゲームをするということは、マジックサークルに入ることであり、また、一つのゲームが始まるようにそれを作り出すことである。

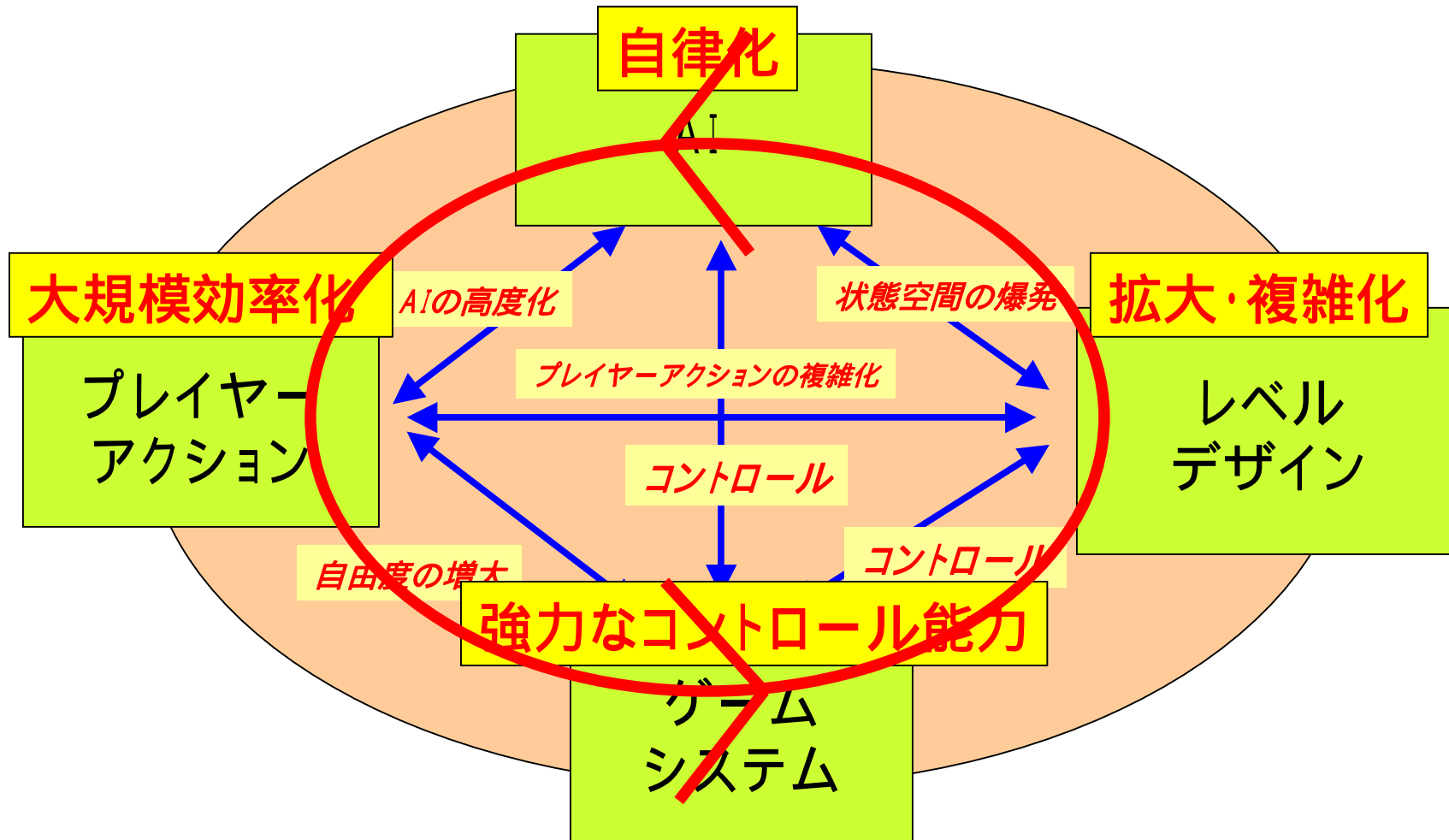
デジタルゲームの現状



マジックサークル = ゲーム(の面白さ)が成立する閉じた行為の連続)

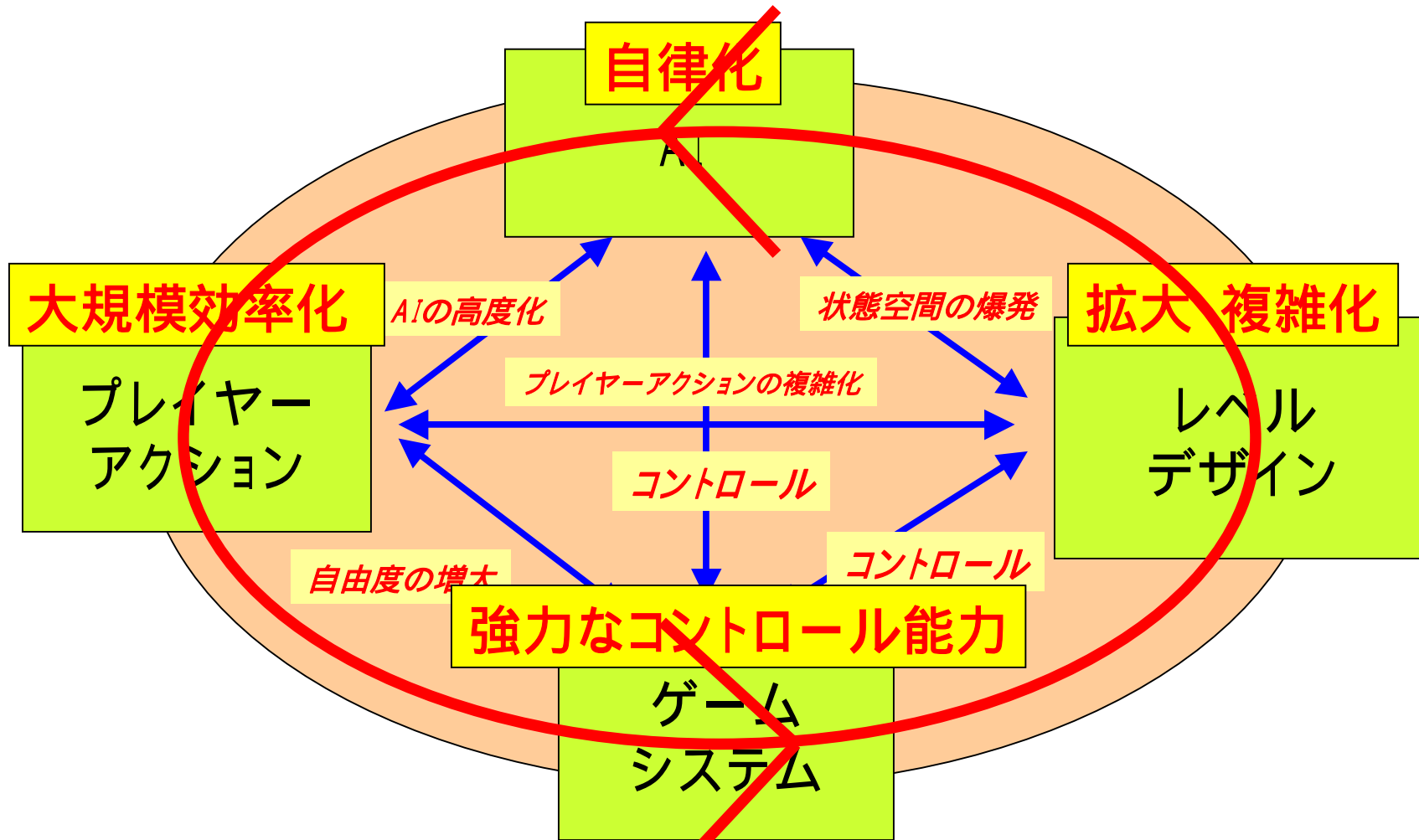
プレイヤー、AI、レベルデザイン、ゲームシステムによってマジックサークルが形成される。

デジタルゲームの最近の変化



もはや、ゲームが拡大してもなお相互的な有機性を保つためには、**プロシージャル技術**が必要である。

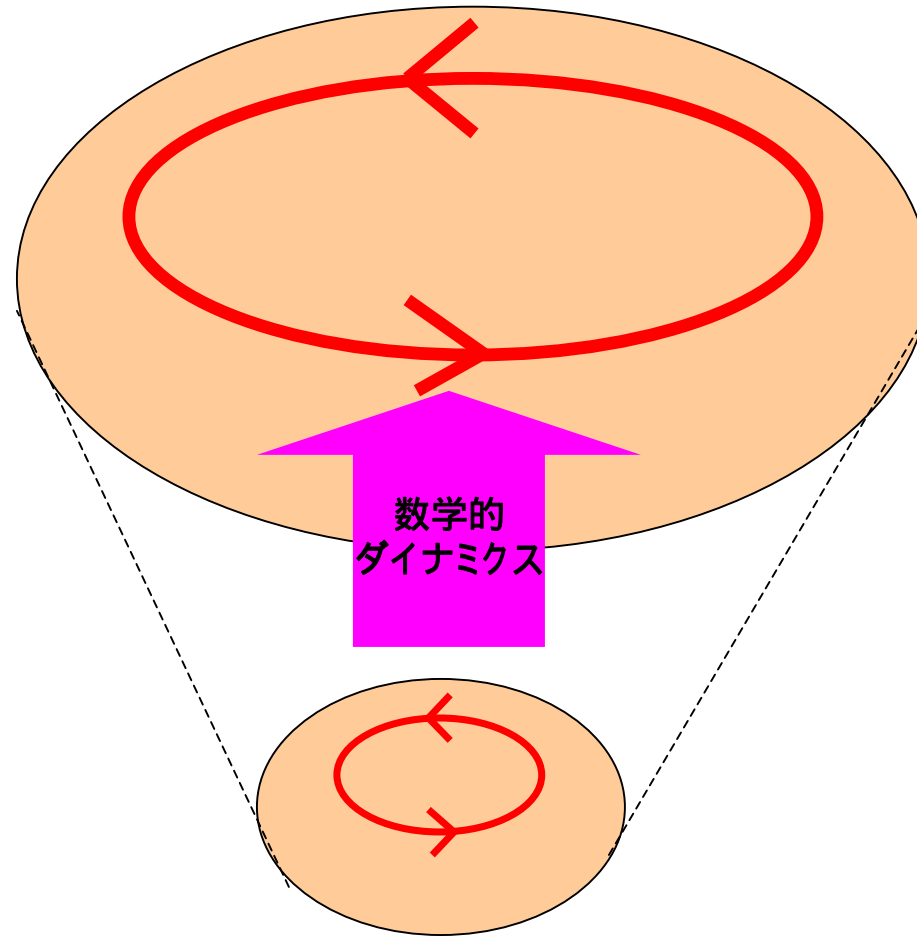
デジタルゲームの最近の変化



ゲームが拡大すれば、マジックサークルも拡大する。

拡大するゲーム空間においてマジックサークルを維持するには
プロシージャル技術が必要である。

マジックサークルとプロシージャル



プロシージャル技術は、拡大するゲーム空間において、マジックサークルの維持しつつ発展させる。

プロシージャル技術はゲームを次の段階へ引き上げる。

これからのゲーム開発者のために...

2D

マリオ FF6 ぷよぷよ
ボンバーマン ドラゴンクエスト
ゼビウス マッピー メトロイド
スーパーマリオ ゼルダの伝説
スーパースターフォース

3D

FF7 オメガブースト
塊魂 斑鳩 ソニックライダーズ
バーチャファイター4
サイキックフォース アーマードコア
激走マウンテンバイク
旋光の輪舞

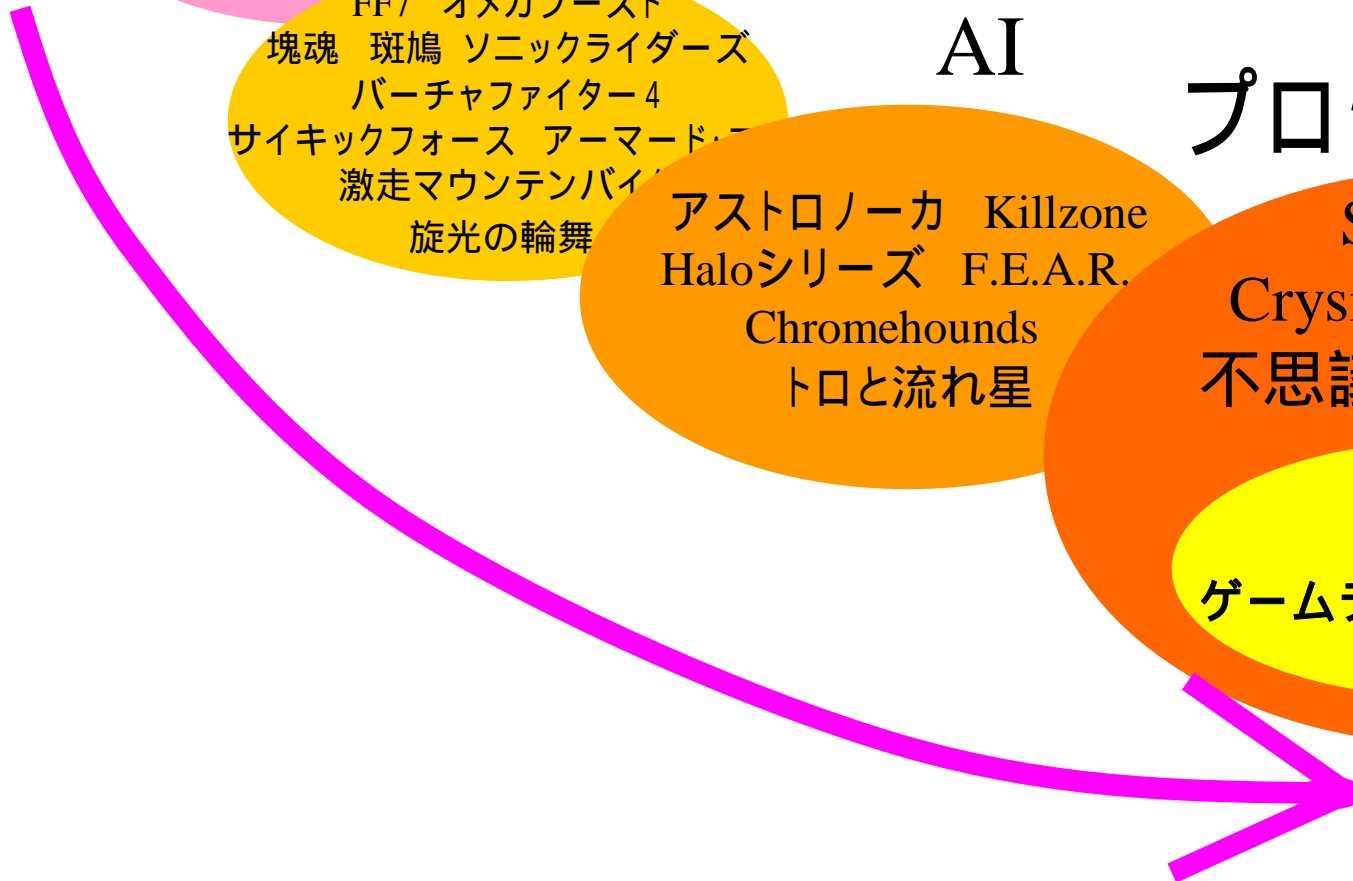
AI

アストロノーカ Killzone
Haloシリーズ F.E.A.R.
Chromehounds
トロと流れ星

プロシージャル

Spore
Crysis Far Cry
不思議のダンジョン

新しい
ゲームデザインの可能性



第3部

より深い領域へ

Programming Tips for Game Programmer

プロシージャルを使って
簡単なゲームを作ってみよう！

生成系を利用したシューティングゲームへの応用

L-system

要素 [A, B]

文法 A AB B A

B

A

AB

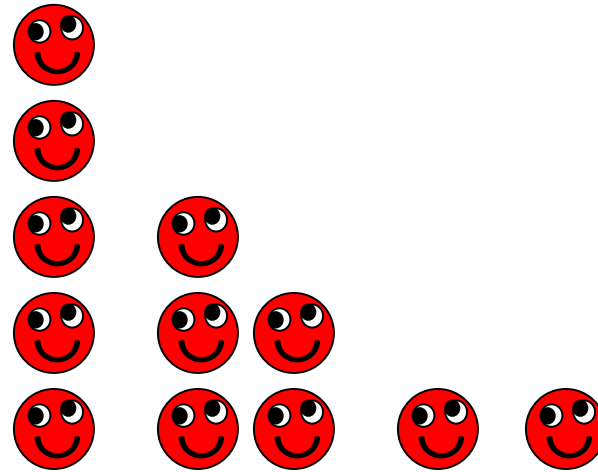
ABA

ABAAB

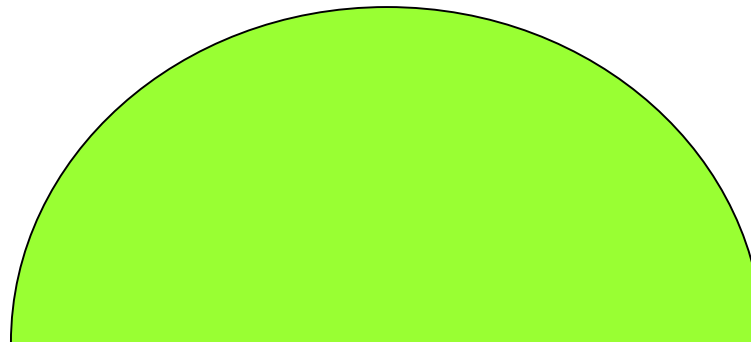
ABAABABA

...

敵キャラクター生成ポイントで利用



敵の配列パターン生成として利用

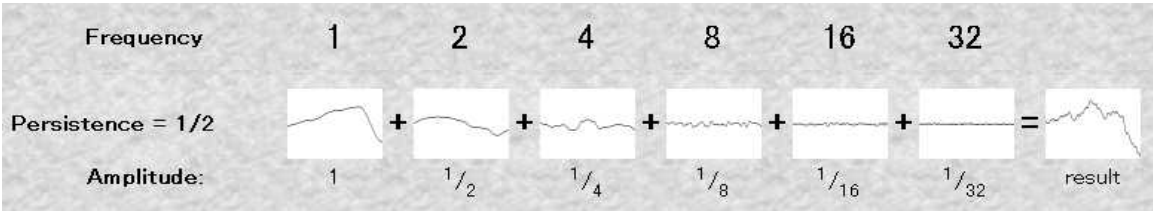


Perlin Noise

パリン・ノイズ

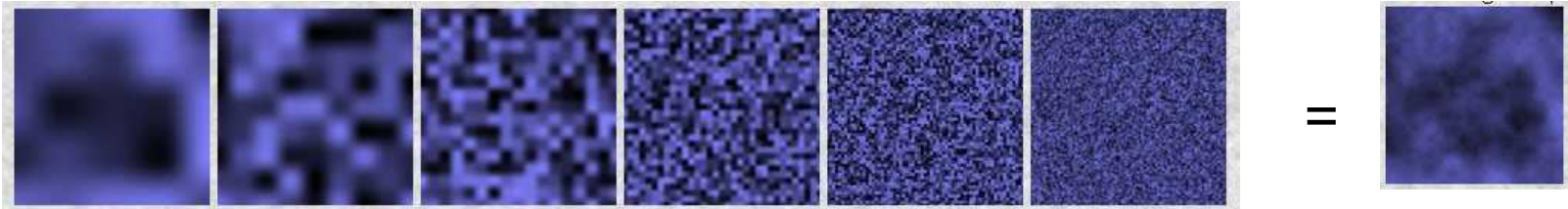
Perlin Noise, http://freespace.virgin.net/hugo.elias/models/m_perlin.htm

1D



階層的な分解能の波長を足し合わせて作るノイズ

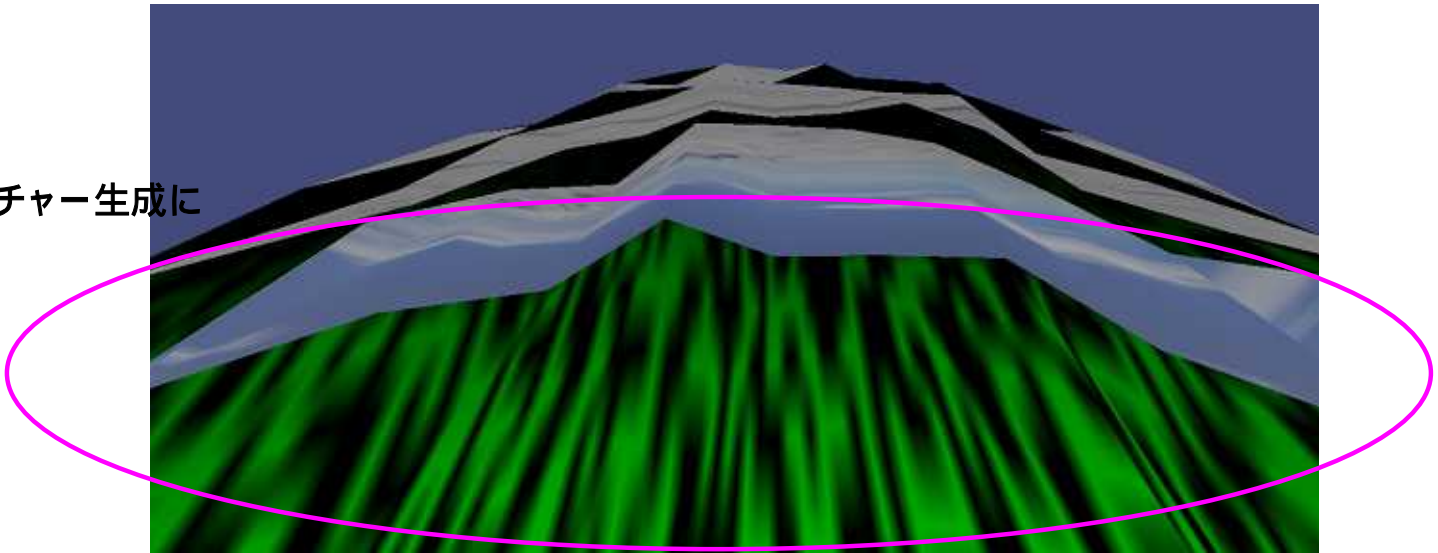
2D



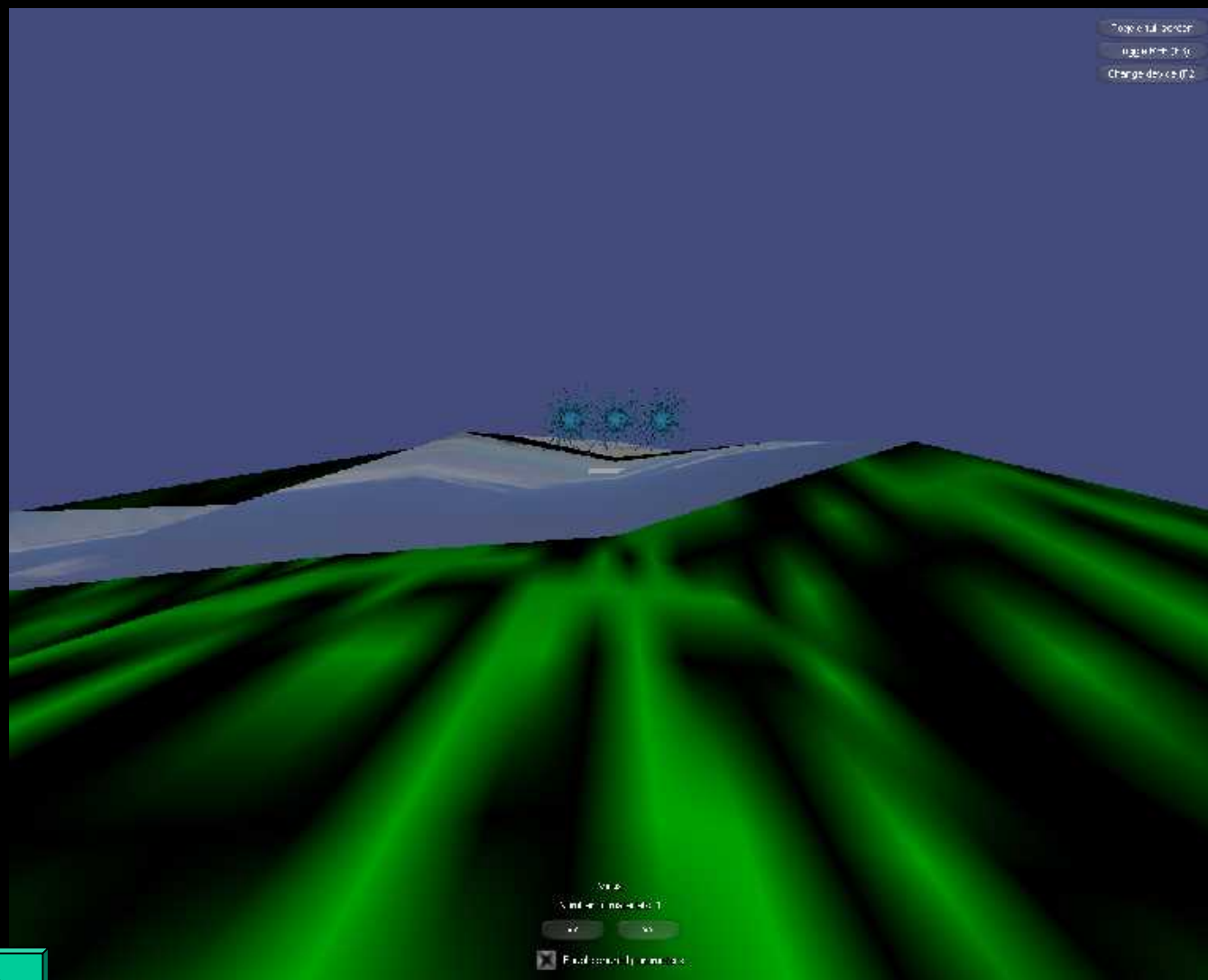
2倍ずつ細かくなっています

result

テクスチャー生成に
利用



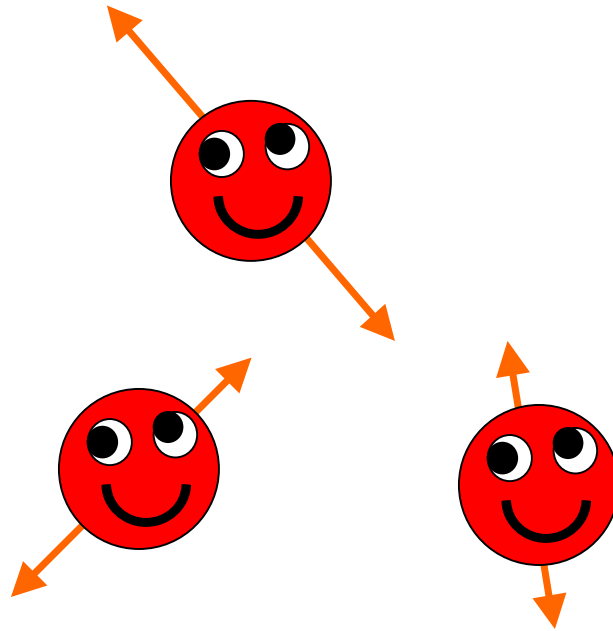
デモ



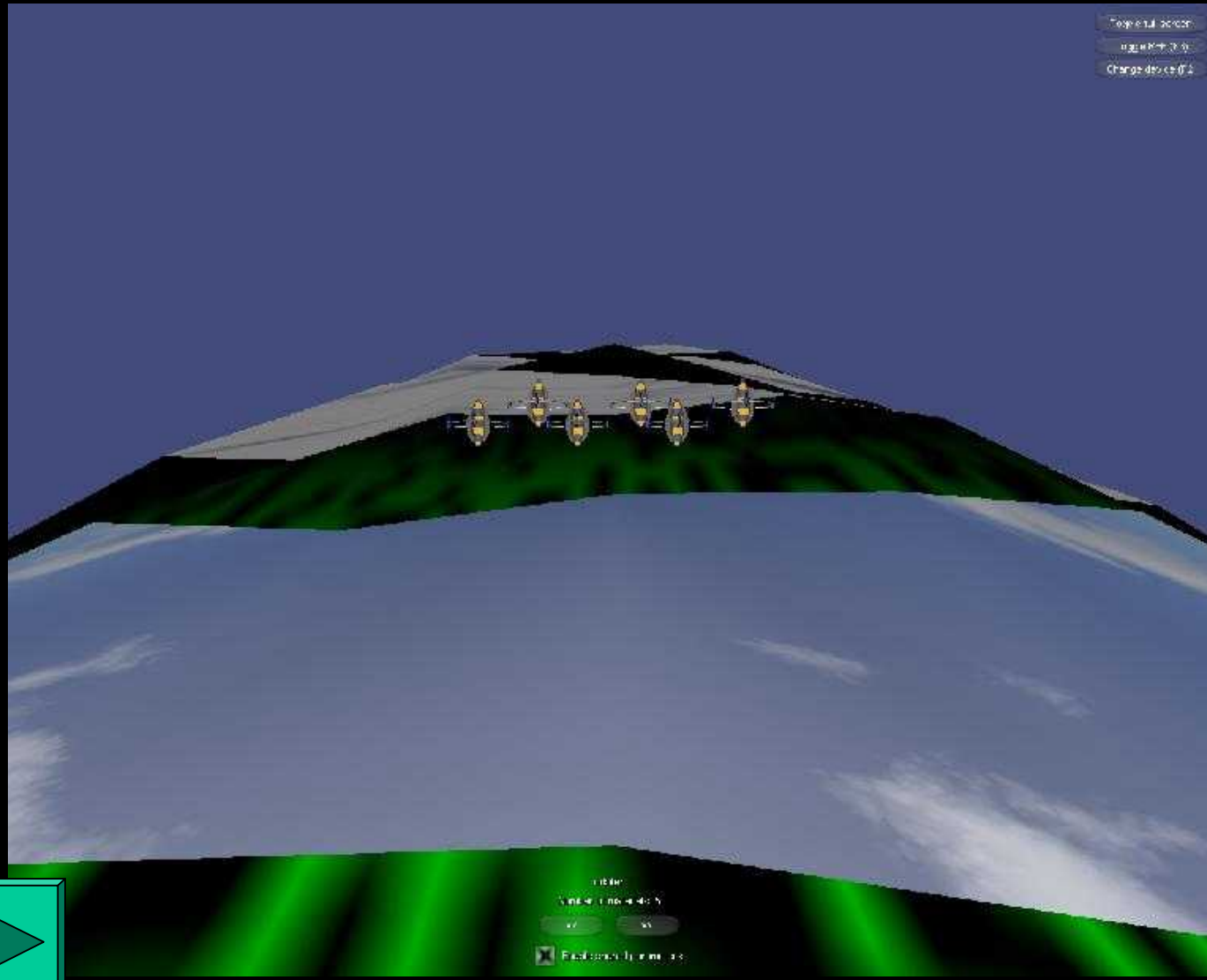
ステージとキャラクター出現を定義する

01

関数によってキャラクターに動きを与える

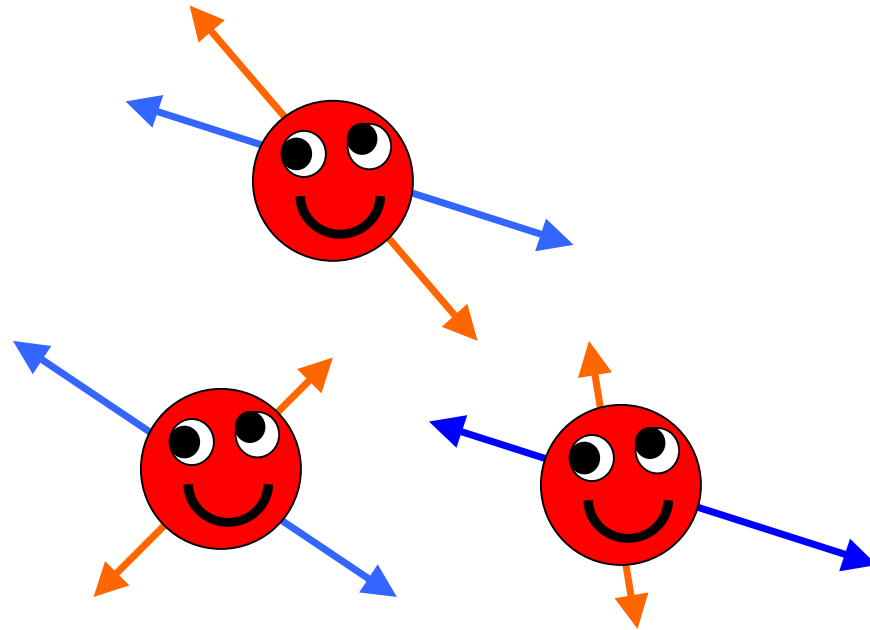


デモ



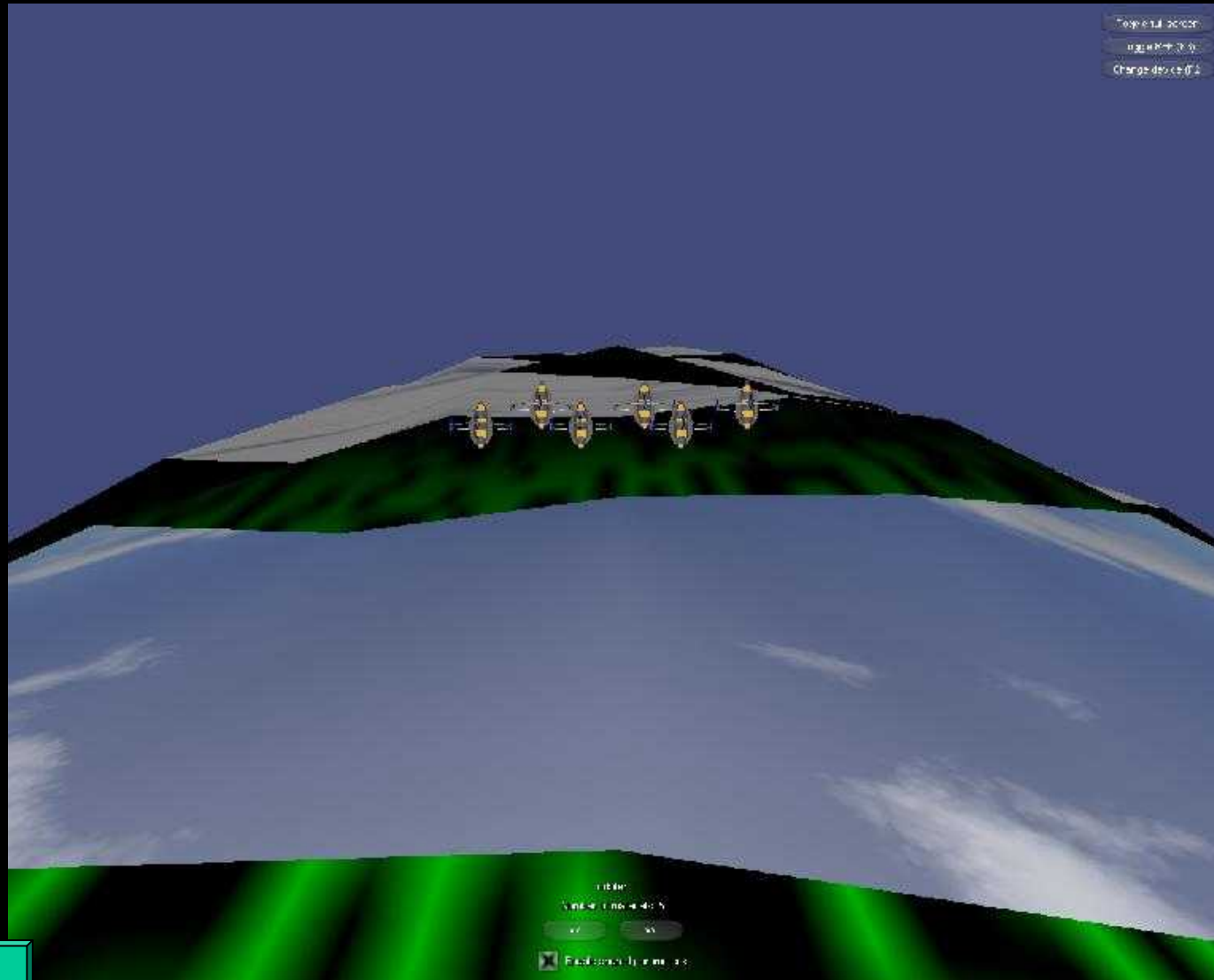
関数によって動きを与える

動作の関数を重ね合わせる



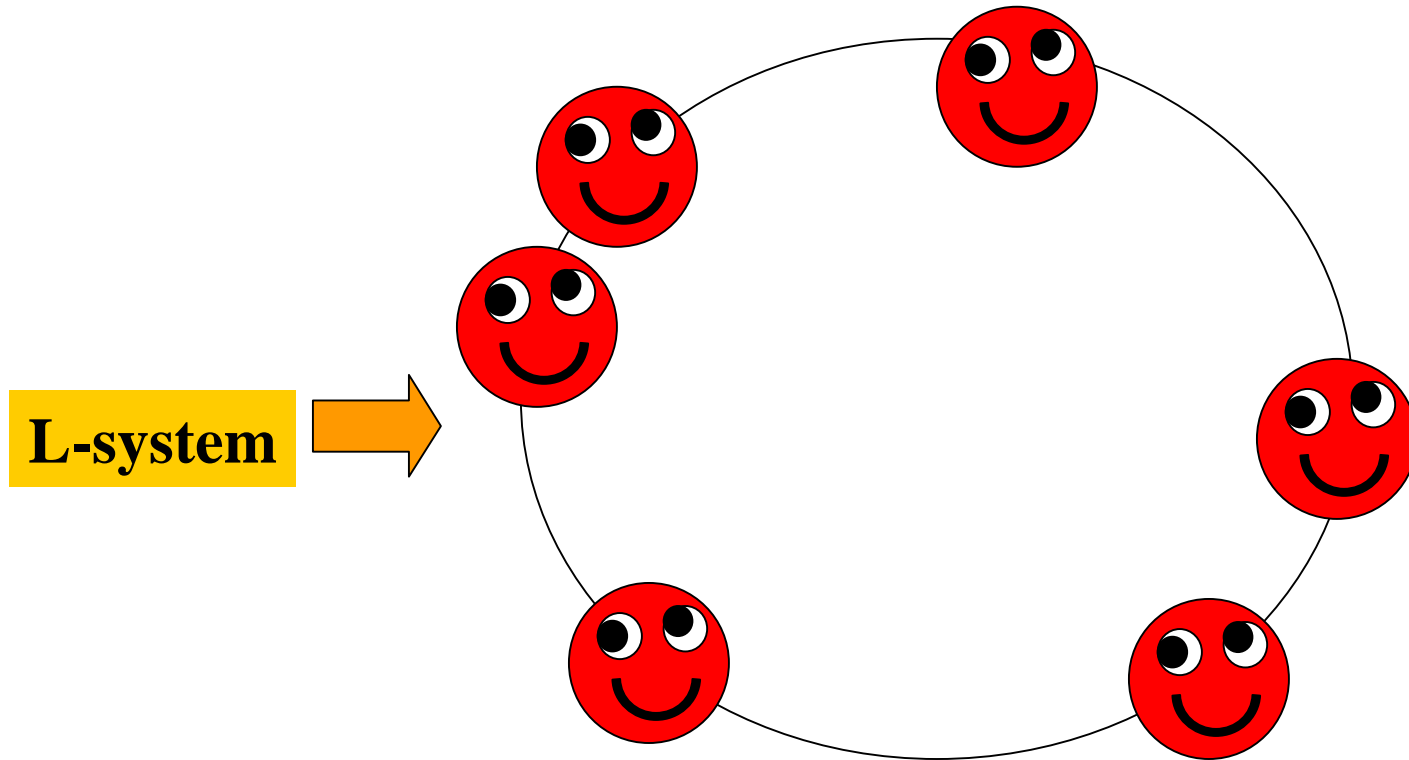
より多様な動き

デモ

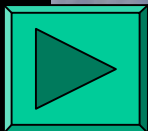
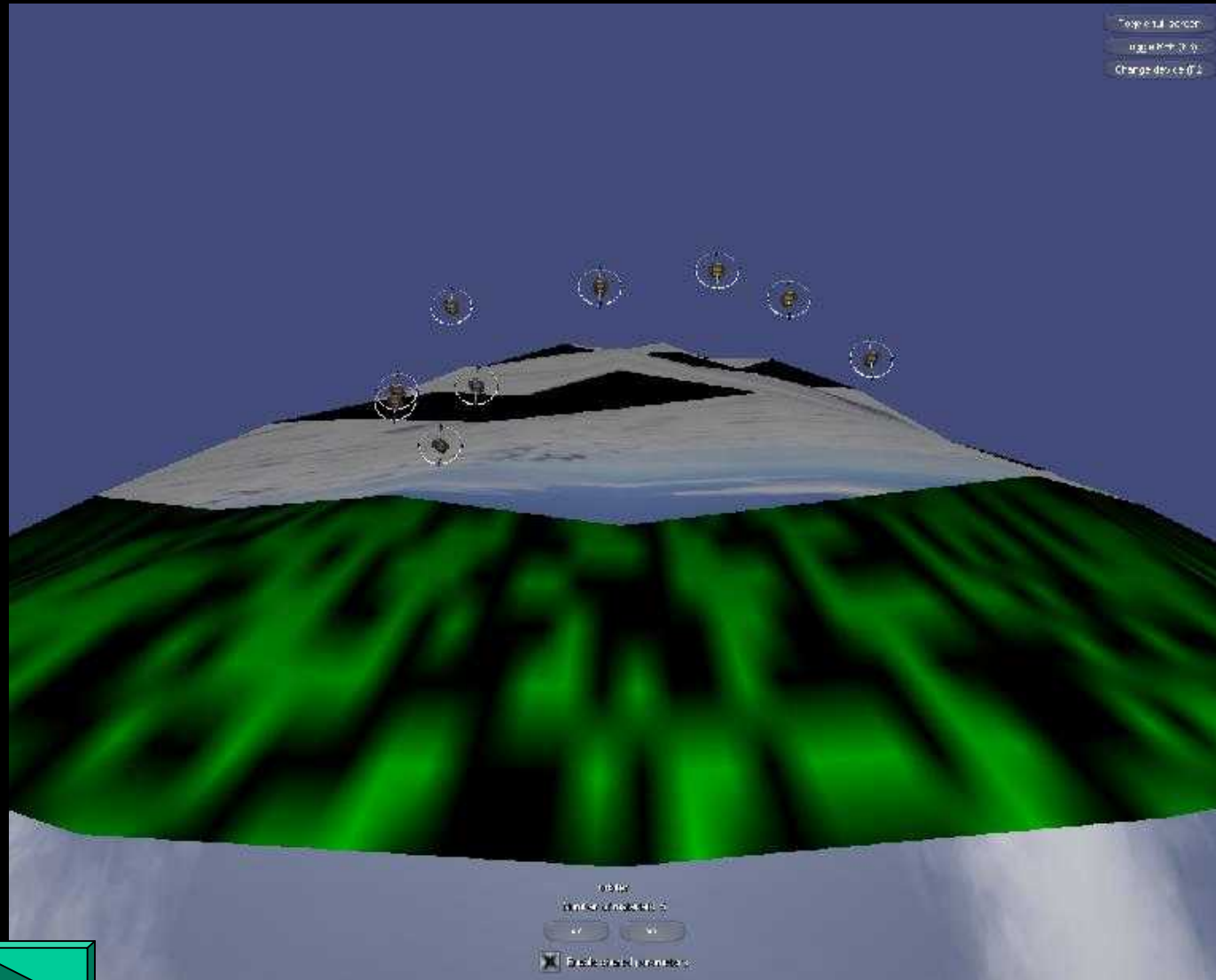


幾つかの関数を組み合わせによって動きを与える

初期配置を円状に分布させる

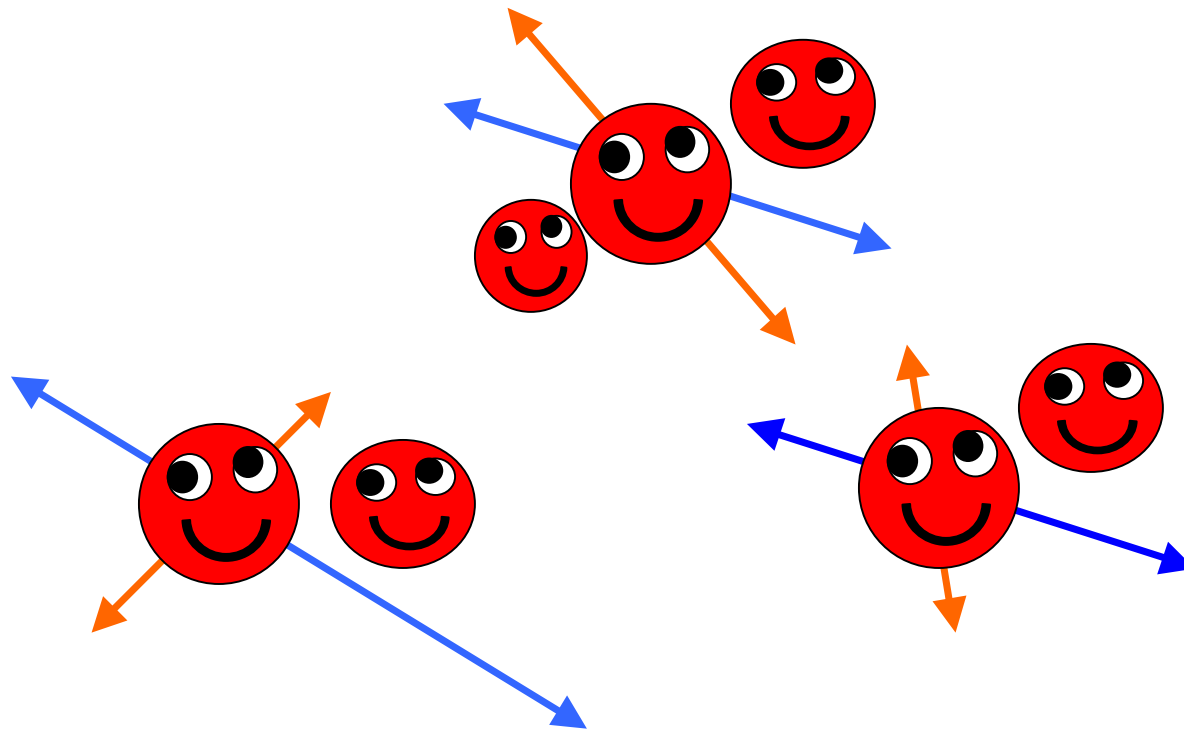


デモ

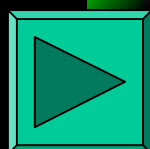
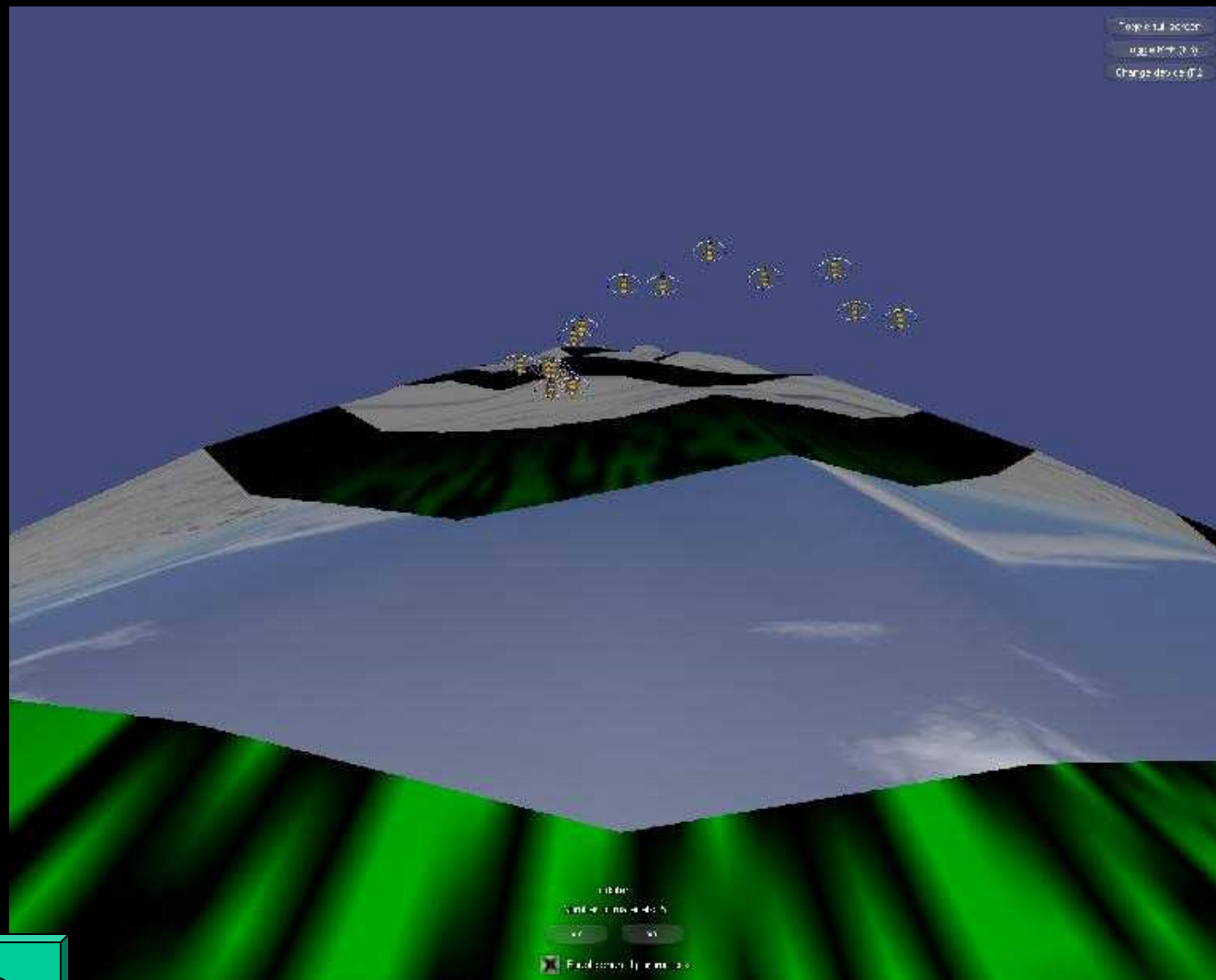


初期配置を円状に分布させる

振幅を変化させる



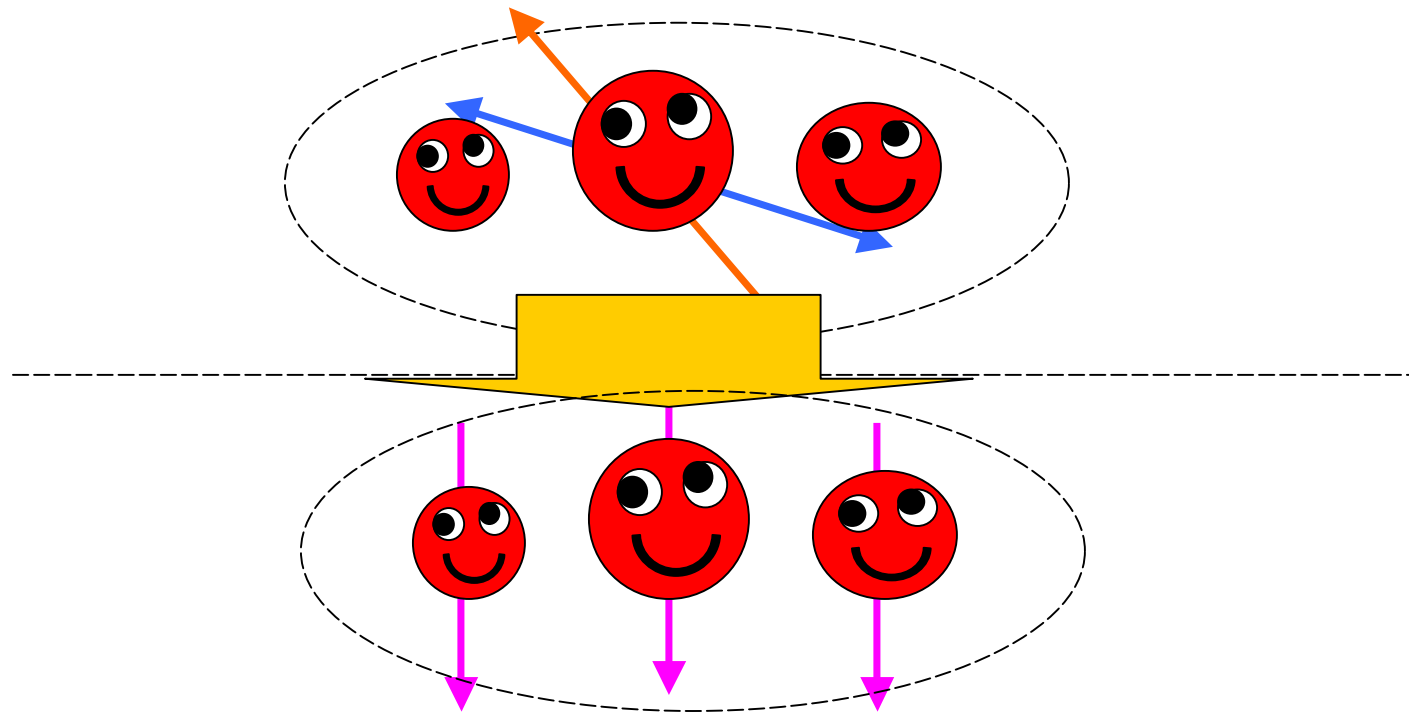
デモ



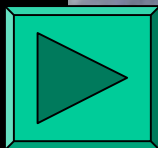
16

振幅を変化させる

時間によって制御を変える



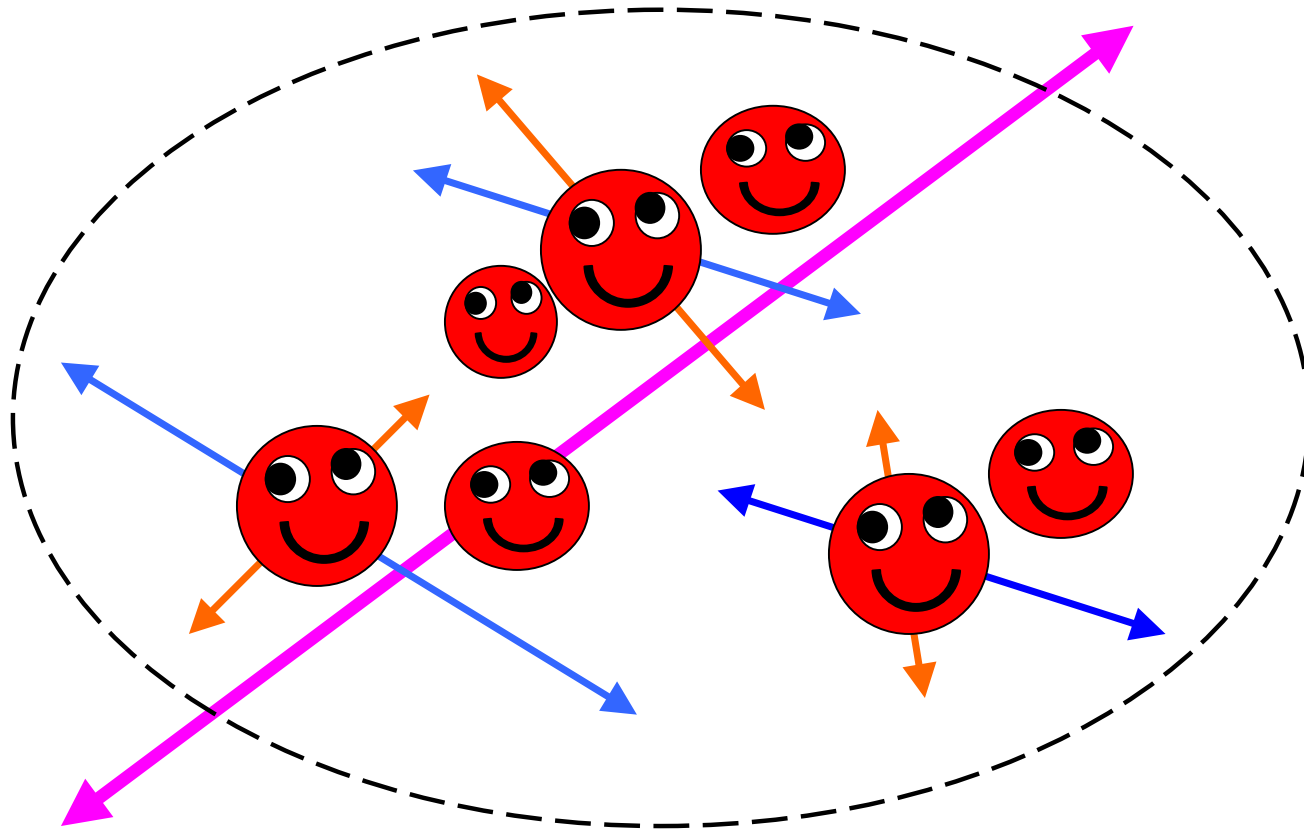
デモ



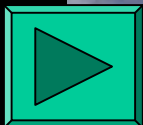
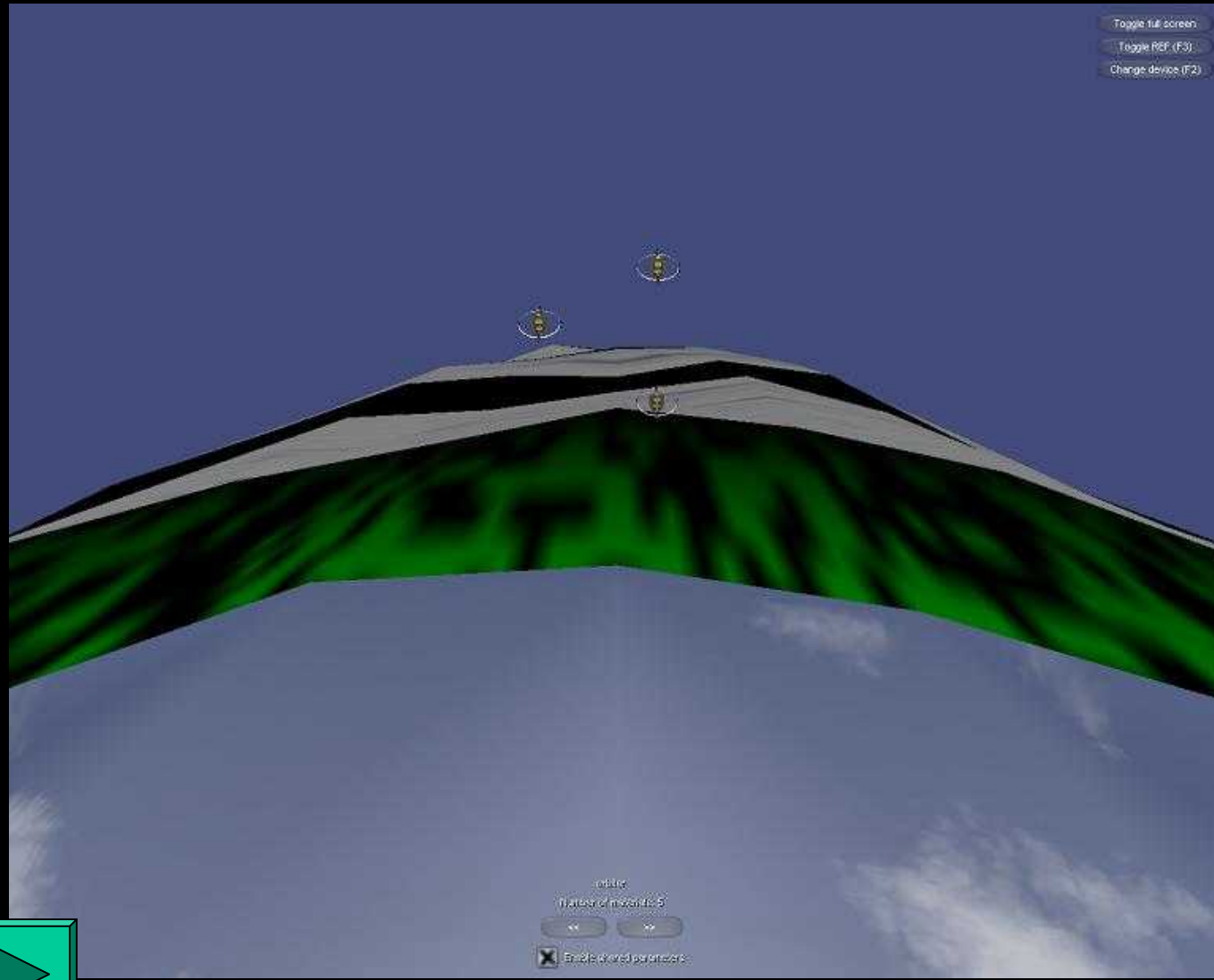
20

時間によって制御を変える

全体に制動を与える

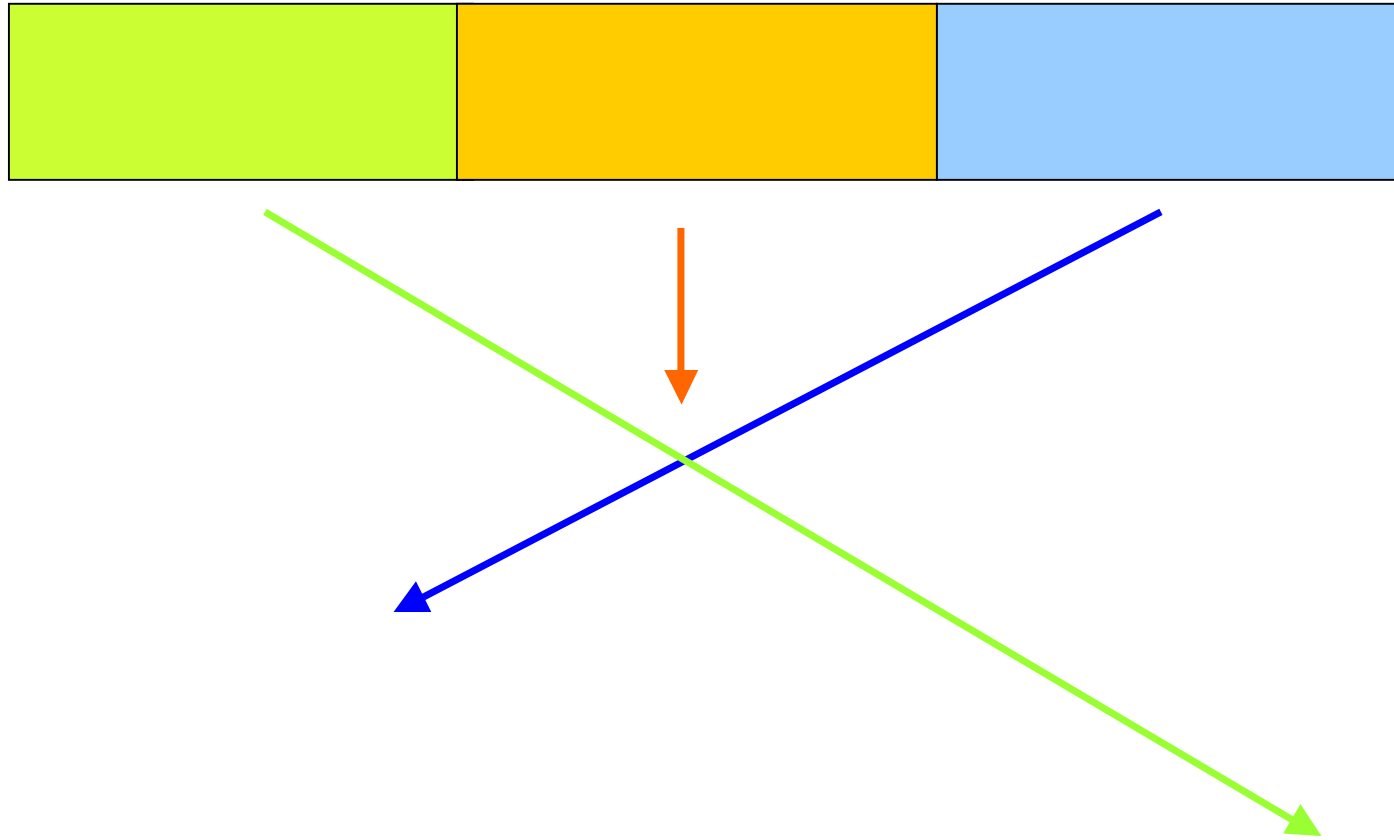


デモ

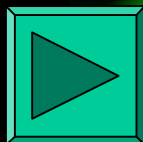
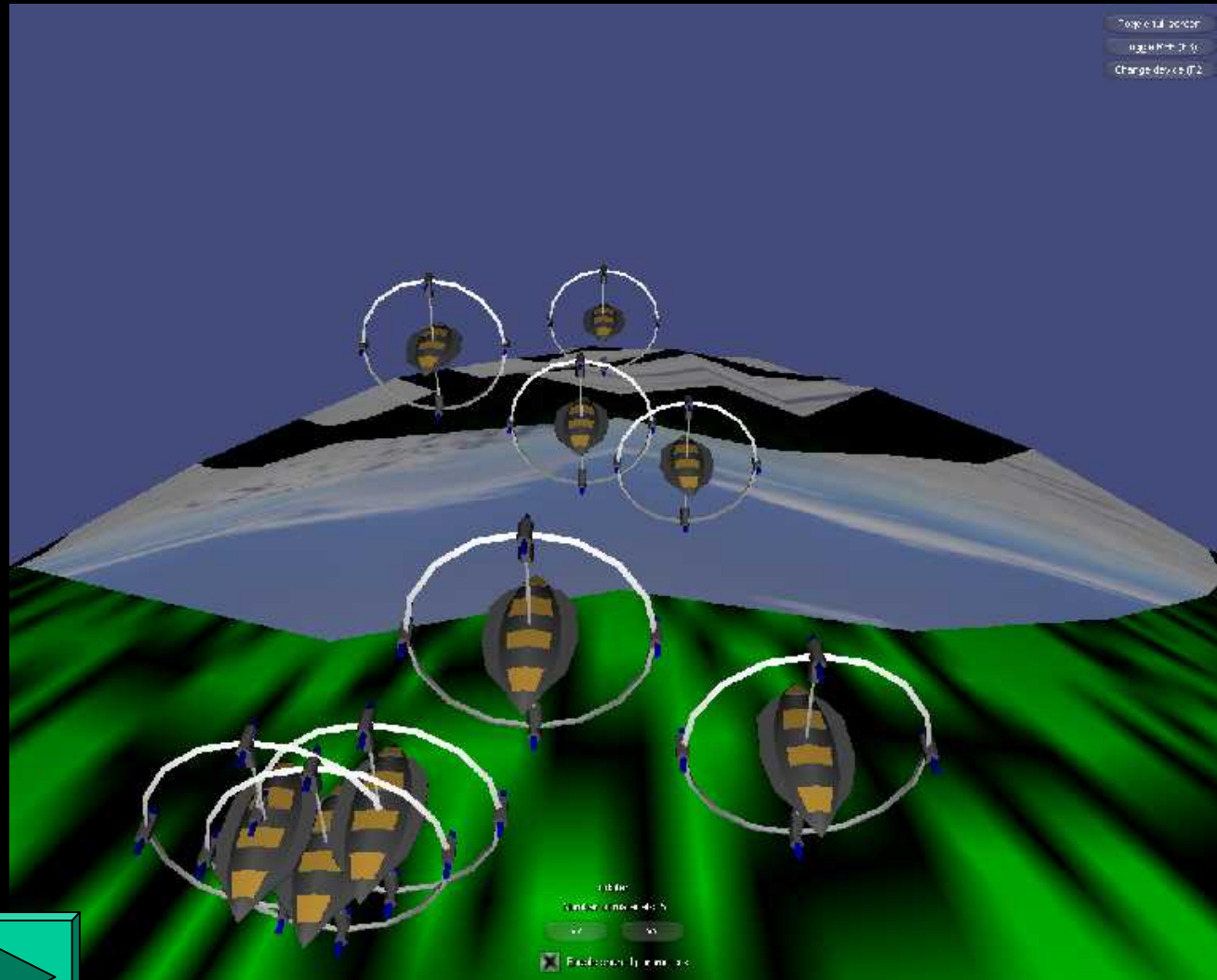


全体に制動を与える

群れに分けて制御を変える



デモ



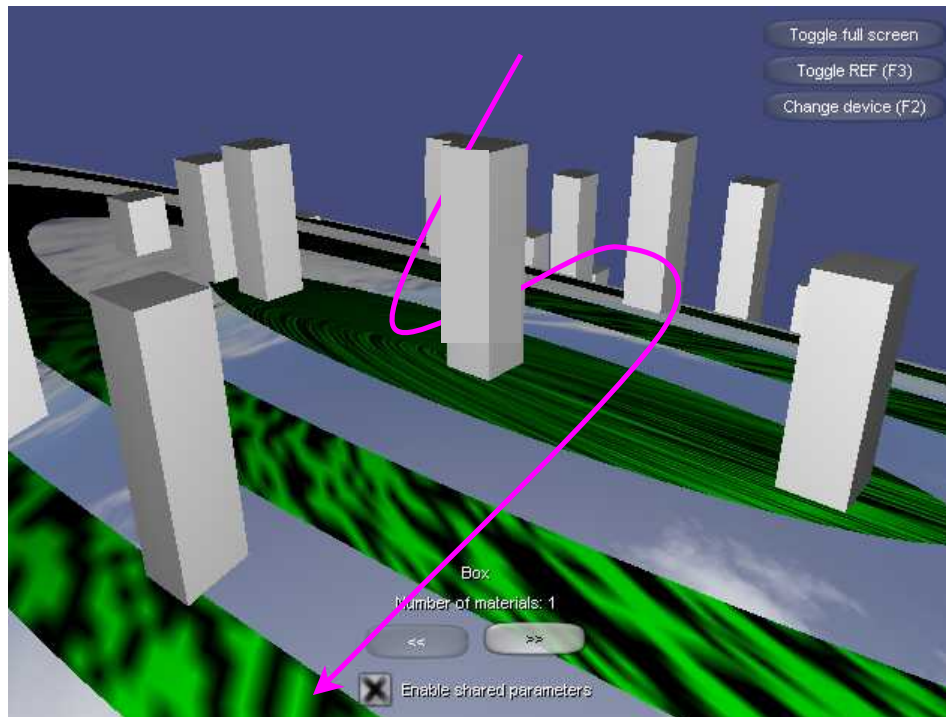
23

群れに分けて制御を変える

アドバンスト(展望)

パス検索

これまでの2D, 3Dシューティングゲームの地平を超えるためには、パス検索システムを導入することが必要

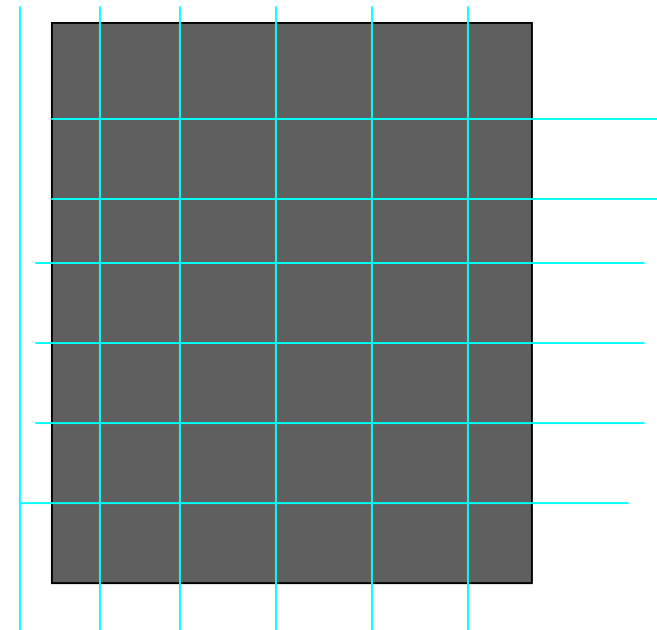


3Dシューティング

3Dパス検索(データ生成が課題、難)

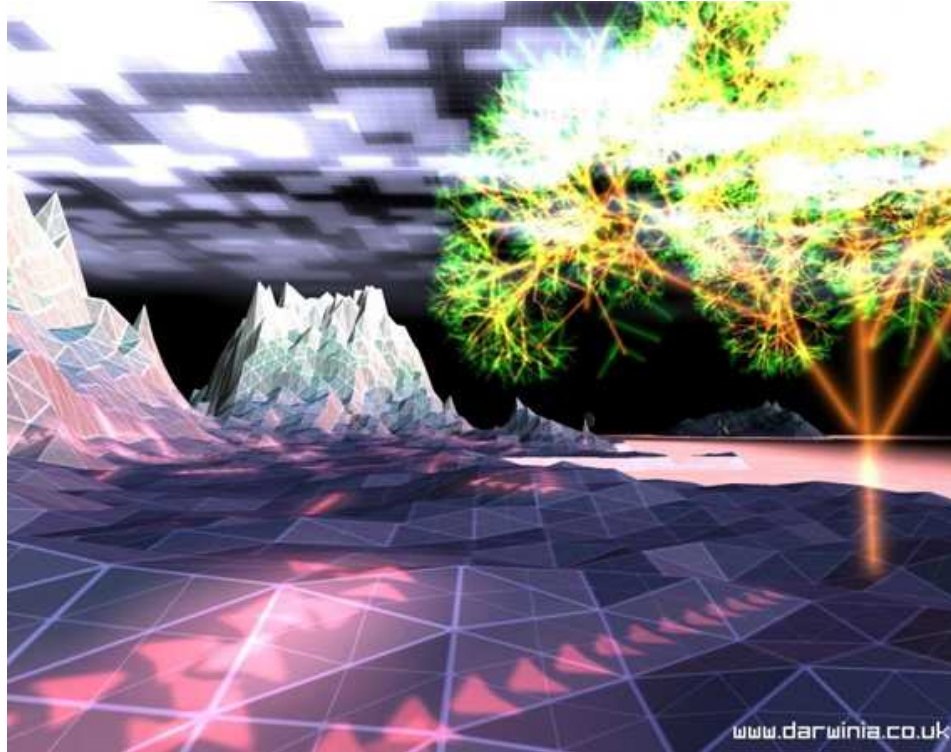
2Dシューティング

2Dパス検索(導入可能)



文字列からの地形自動生成

Darwinia(Introversion Software) 技術情報 未公開



小さなプロダクションでも
プロシージャルを使うことで
質の高い大きなゲームを
作ることができる可能性を
知らしめて、英語圏の
ゲーム関係者に衝撃と注目を
集めている

- (1) 4人で製作
- (2) ベッドルーム・プログラマー 2人
- (3) グラフィックアー 0人
- (4) 2006年 Independent game festival 大賞

[Introversion Software, "Procedural Content Generation", GameCareerGuide.com, 2007](#)

4gamers(体験版):<http://www.4gamer.net/patch/demo/darwinia/darwinia.shtml>

文字列からの地形自動生成

Darwinia(Introversion Software) 技術情報 未公開



小さなプロダクションでも
プロシージャルを使うことで

プロシージャル技術は、
大企業から小企業、そして
インディーズ、同人まで
広い範囲で多様に応用される。

を
を
注目

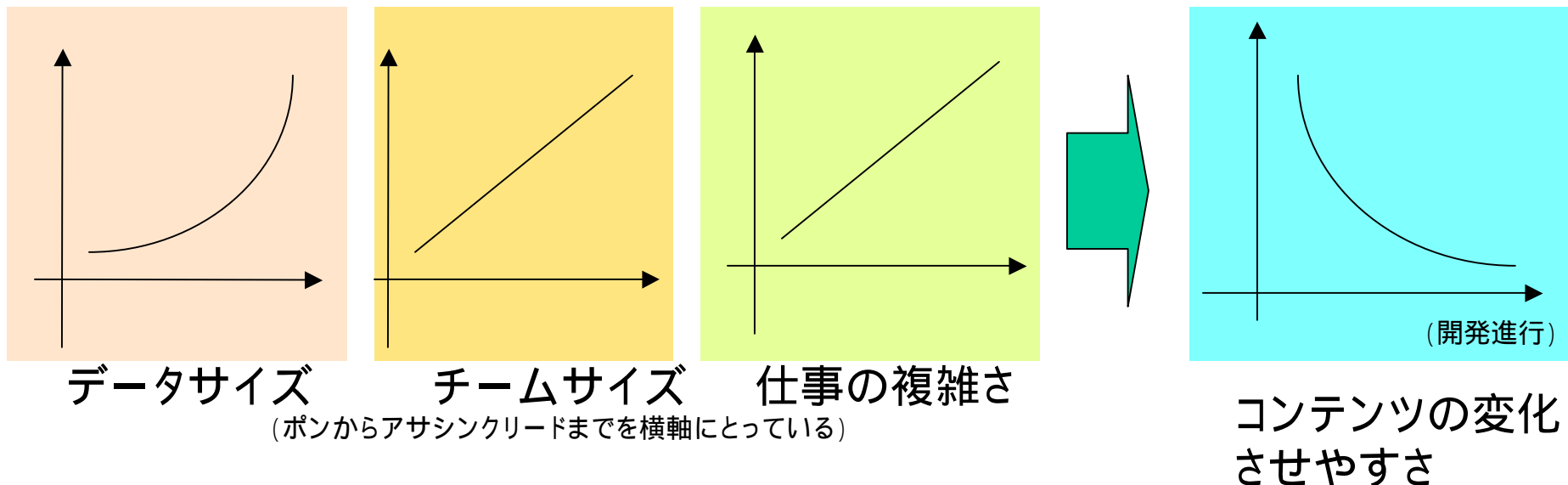
- (2) ベッドルーム・プログラマー 2人
- (3) グラフィックアー 0人
- (4) 2006年 Independent game festival 大賞

[Introversion Software, "Procedural Content Generation", GameCareerGuide.com, 2007](http://www.introversion.com/)

4gamers(体験版):<http://www.4gamer.net/patch/demo/darwinia/darwinia.shtml>

現在のゲーム開発の問題点

Rushing into a wall (壁にぶつかる)



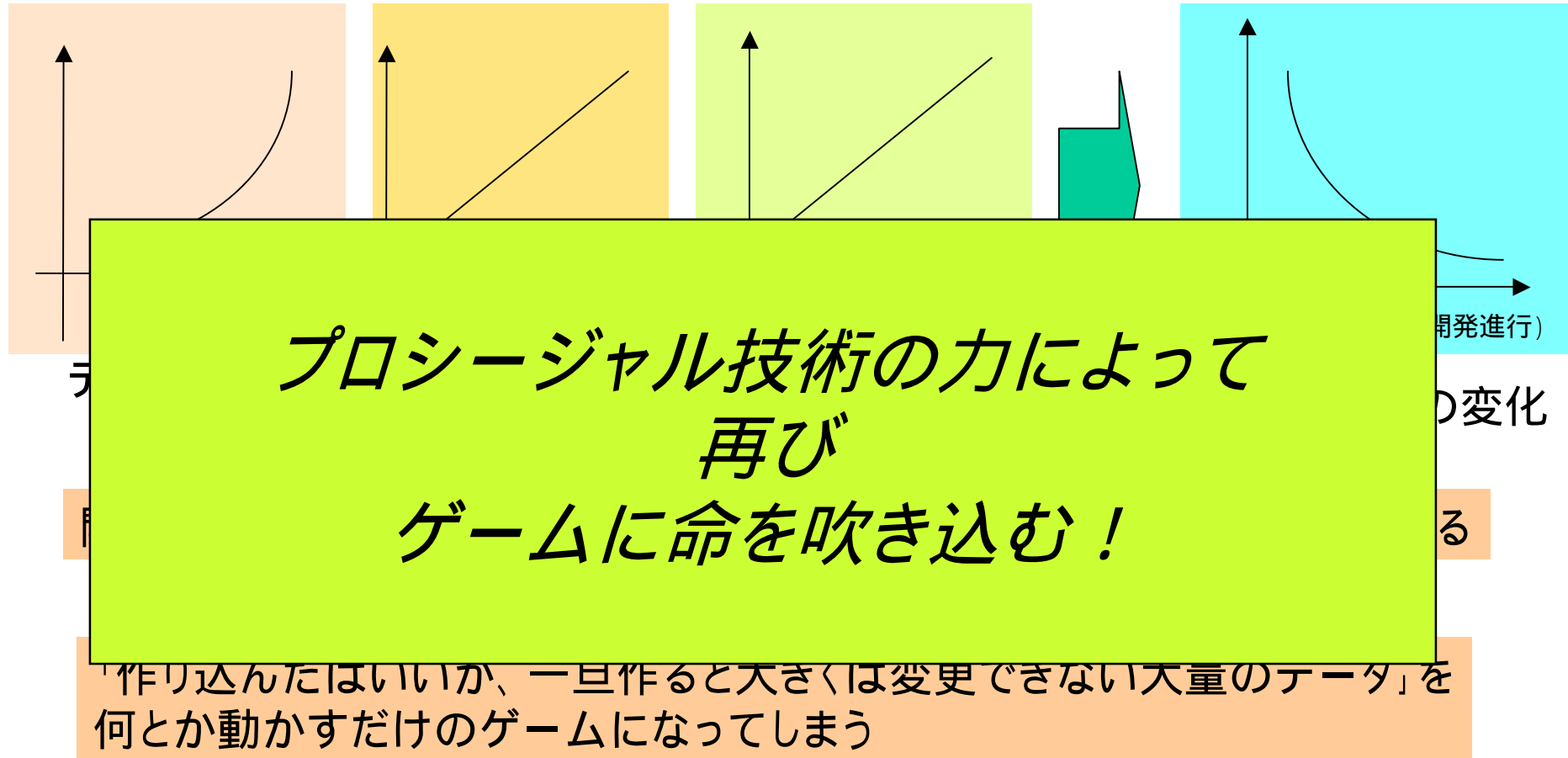
開発が進むにつれて、ゲームコンテンツはますます変更しにくいものになる

「作り込んだはいいが、一旦作ると大きくは変更できない大量のデータ」を何とか動かすだけのゲームになってしまう

それはゲームか？

現在のゲーム開発の問題点とブレイクスルー

Rushing into a wall (壁にぶつかる)



それはゲームか？

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

ご清聴ありがとうございました。



Photo from <http://www.cyberleaf.com/>

これ以外に、意見や質問があれば、メールかアンケートへ

y_miyake@fromsoftware.co.jp

(IGDA Japan登録アドレス yoichi-m@pk9.so-net.ne.jp)

<http://www.igda.jp>