

こんにちは！



ミケーヤ



DiGRA Japan 第14回 月例研究会

「Spore におけるゲームAI技術とプロシージャル」

Game AI & Procedural Contents Generation in Spore

— ウィル・ライトのゲームAI論 —

三宅 陽一郎
(フロム・ソフトウェア)

y_miyake@fromsoftware.co.jp

7.4.2008

この資料はIGDA日本のサイトにアップされます。



コンテンツ

第1部

ゲームAI技術から見る SimCity – The Sims – Spore

第2部

プロシージャル・コンテンツ・ジェネレーションと Spore

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.ca.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.

References (2) Maxis 開発者たちの情報

- (1) Kenneth D. Forbus
Some notes on programming objects in. The Sims
http://www.qrg.northwestern.edu/papers/Files/Programming_Objects_in_The_Sims.pdf
- (2) Ken Forbus, “Simulation and Modeling: Under the hood of The Sims”
(NorthWestern大学、講義資料)
http://www.cs.northwestern.edu/%7Eforbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/frame.htm
- (3) Alex J.Champandard
“Living with The Sims’ AI: 21 Tricks to Adopt for Your Game”
<http://aigamedev.com/reviews/the-sims-ai>
- (4) Jake Simpson (Lead Simulator Engineer on Sims 2) GDC2005
“Scripting and Sims2: Coding the Psychology of Little people”
<https://www.cmpevents.com/Sessions/GD/ScriptingAndSims2.ppt>
http://www.gamedev.net/columns/events/coverage/feature.asp?feature_id=51

Figures on the pages are from these references.

References (3) Maxis 開発者たちの情報

Don Hopkins のサイト <http://www.donhopkins.com/drupal/>
The Sims のコーナーには、いろいろな情報が載っています。

http://www.donhopkins.com/drupal/taxonomy_menu/4/49/3

The Soul of The Sims, by Will Wright

<http://www.donhopkins.com/drupal/node/148>

には、Will Wright が、1997 年に書いた、Macintosh 用の C-Program が掲載されています。

例(The Sims)の、キャラクターの内面の制御コードのようです。(Motive Engine)
数値は複雑ですが、300行ほどの単純なコードです。

確かに、The Sims の中核になるコードですね。(本物だろうとは思いますが、嘘をつく必要がない！)
確かに、Will Wright は生粋のシム好きですね。僕もこういうコード書くからよくわかります。

見えない力学(街を形成する力、進化の力)を数値上で再現したくなる、という欲求がよく表現されています。

Figures on the pages are from these references.

References (4)

Development of Spore Wiki

http://en.wikipedia.org/wiki/Development_of_Spore

は、特に最後のリファレンスが技術情報の宝庫です。

ただし、Wiki なので、完全に正しい情報ばかりでは、ありません。

Figures on the pages are from these references.

第1部

ゲームAI技術から見る SimCity – The Sims – Spore

Game AI Technology in SimCity, The Sims & Spore

第1章 “ウィル・ライト”論

第2章 ウィル・ライトのゲームAI論

The SimCity Sub AI
The Sims Peer AI
Spore AI

“ウィル・ライト”論

“ウィル・ライト”とは誰か？

Bungeling Bay (1984)

SimCity (1989)

SimEarth (1990,

SimAnt (1991)

SimCity 2000 (1993)

SimCopter (1996)

SimCity 3000 (1999)

The Sims (2000)

SimCity4 (2003)

The Sims 2 (2004)

SimCity Societies (2007)

Spore (2008)

Spore (2000)

The Sims 3 (2008)

http://spore.wikia.com/wiki/Will_Wright

三宅の ゲームデザイナー“ウィル・ライト” 論

街を形成する力、生態系、進化を促す力など、
見えないものの力学 = ダイナミクス を把握する力を持ち、
それをゲームで表現しようとする人

I'm interested in the process and strategies for design.

The architect Christopher Alexander, in his book *Pattern Language* formalized a lot of spatial relationships into a grammar for design.

I'd really like to work toward a grammar for complex systems and present someone with tools for designing complex things.

- Will Wright, from an interview with Wired magazine, 1994.

経済学、生態学、進化、物理、AI、こういったダイナミクスを全体のネットワークとして捉える能力は研究者としては、珍しくないが、ゲーム開発者としては稀。

(Maxis の産学連携の原動力の一つであると予想できる)

三宅の 技術者“ウィル・ライト” 論

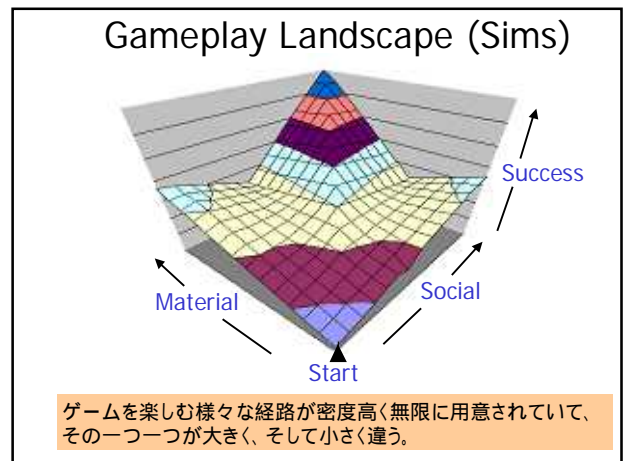
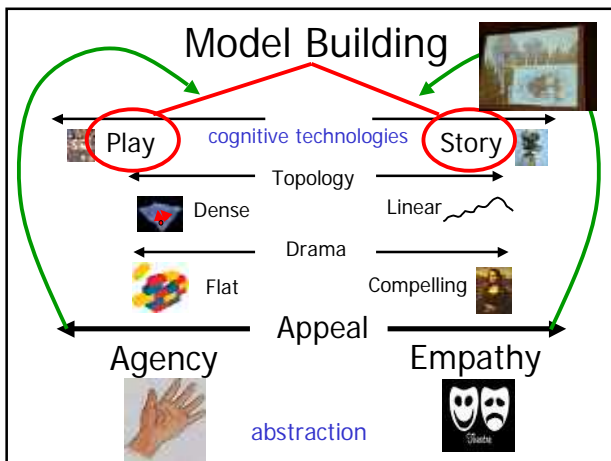
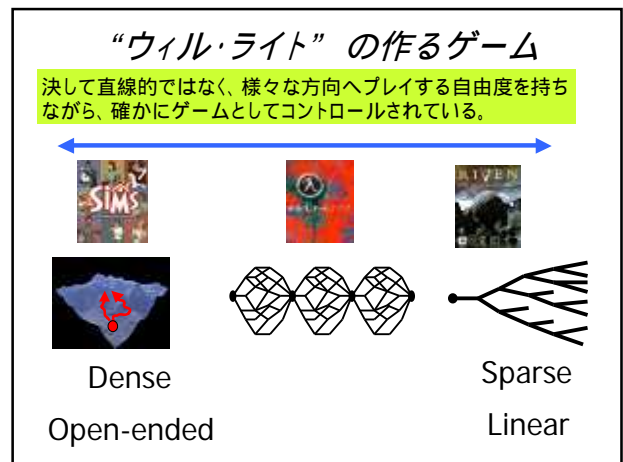
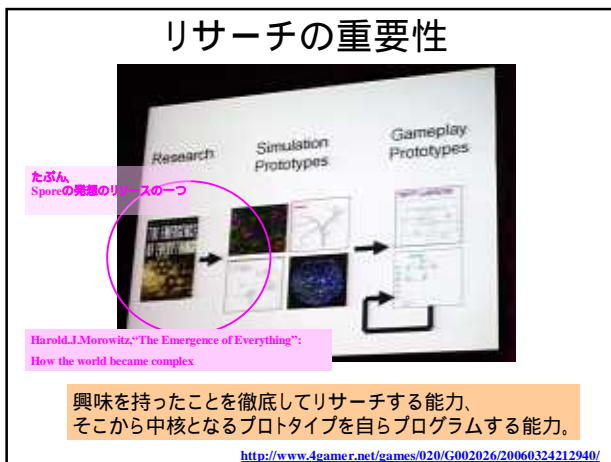
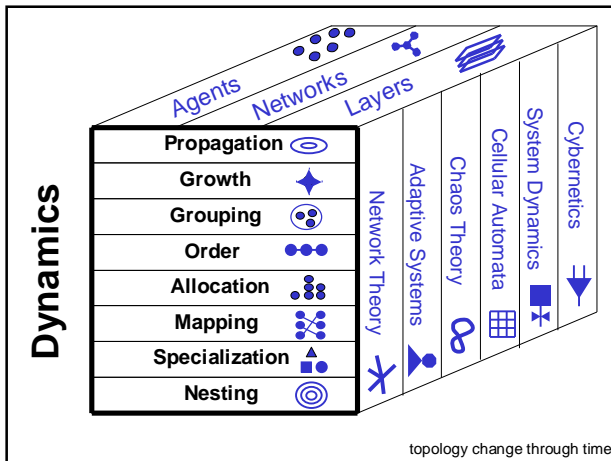
把握した **ダイナミクス** を数値モデルとして実現できる能力を持つ

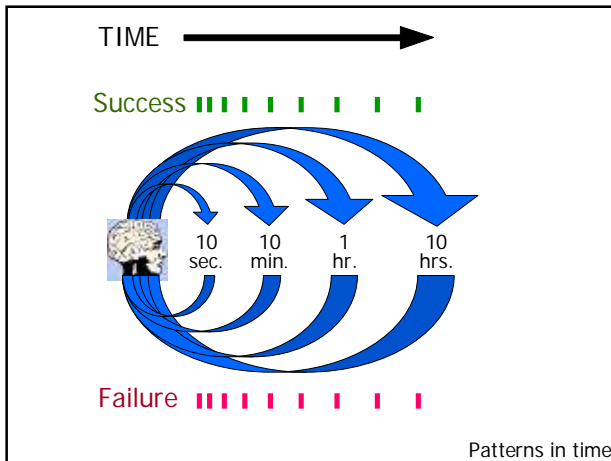
ある朝持って来た、
ウィル・ライトのThe Sims の
ダイナミクスを表現した
C のコード
(The Soul of The Sims, by Will Wright)

(The Soul of The Sims, by Will Wright)

プログラマーらしい、
抽象的な動的な流れを追う能力、
プログラムの可能性をよく知っている。

<http://www.donhopkins.com/drupal/node/148>





第1部

ゲームAI技術から見る SimCity - The Sims - Spore

Game AI Technology in SimCity, The Sims & Spore

第1章 “ウィル・ライト”論

第2章 “ウィル・ライト”のゲームAI論

The SimCity Sub AI
The Sims Peer AI
Spore AI

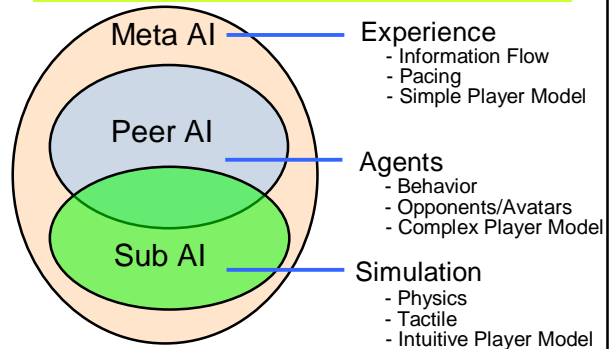
“ウィル・ライト”のゲームAI論

The SimCity Sub AI
The Sims Peer AI
Spore AI

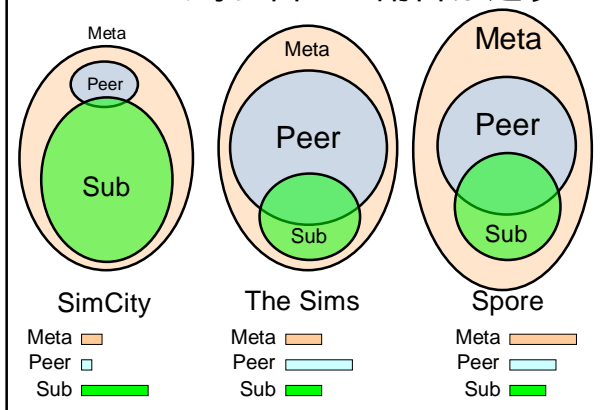
“AI: A Desing Perspective” AIIDE 2005 (PPTが公開) に準拠
<http://www.aiide.org/aiide2005/talks/index.html>

“ウィル・ライト”のゲームAI論

ゲーム体験をコントロールするAI、キャラクターAI、
ゲーム世界を動かすAI、の3つの分類してゲームAIを捉える

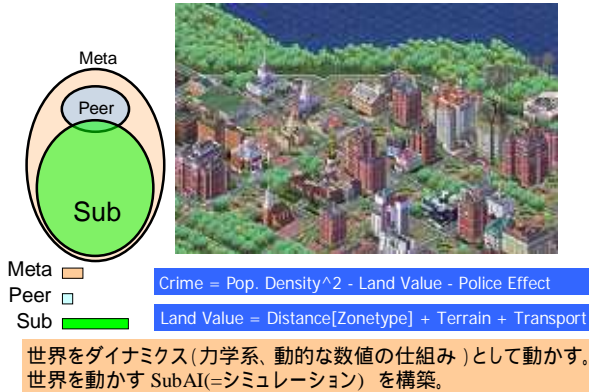


ゲーム毎に占める割合が違う



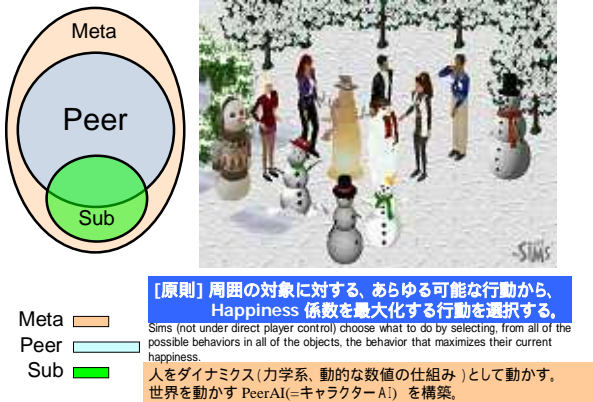
The SimCity Sub AI

SimCity シリーズのAIの作り方



The Sims Peer AI

The Sims シリーズのAIの作り方



オブジェクトに仕込むデータ構造

Data (Class, Sate)

Graphics (sprites, z-
Animations (skeletal)

Sound Effects

Code (Edith)

-Main (object thread)
-External 1
-External 2
-External 3

パラメーター

グラフィックス
アニメーション

サウンド

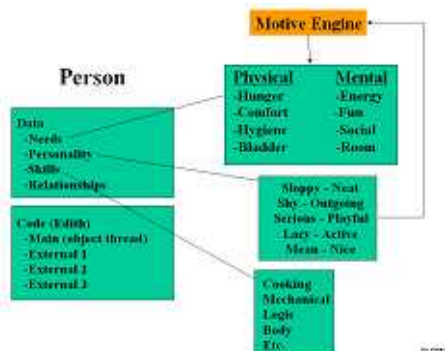
メインスレッド

いろいろなインタラクションの仕方



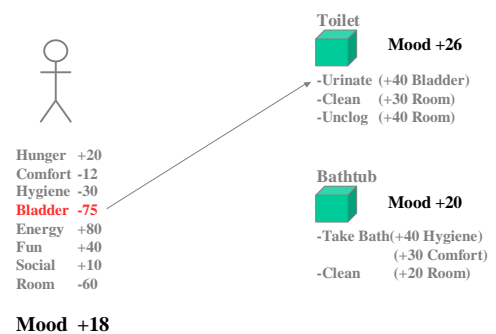
Ken Forbus, "Simulation and Modeling: Under the hood of The Sims" (NorthWestern大学、講義資料)
http://www.cs.northwestern.edu/%7Eforbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/frame.htm

NPCに仕込むデータ構造



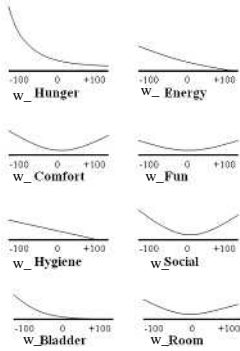
Ken Forbus, "Simulation and Modeling: Under the hood of The Sims" (NorthWestern大学、講義資料)
http://www.cs.northwestern.edu/%7Eforbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/frame.htm

最適な行動を選択する



[原則] 周囲の対象に対する、あらゆる可能な行動から、
Happiness (ここではMood) 係数を最大化する行動を選択する。

Happiness 係数の計算のためのウェイト



= Mood (-100..+100)

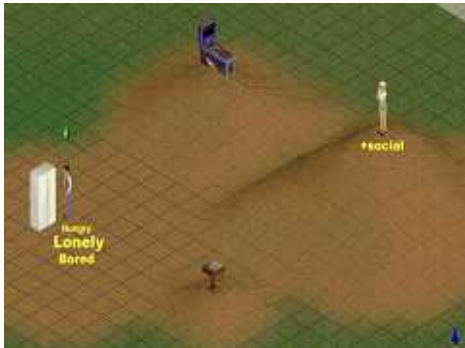
Happiness = $W_Hunger * Hunger + W_Energy * Energy + \dots$

Happiness を最大化



冷蔵庫が最も総合的にHappinessを上昇させるから

Happiness を最大化



冷蔵庫へ行きます。

Happiness を最大化



お腹が膨れたので、ちょっと退屈だから、女の子と話します。

[余談] Halo3

上記の手法は、

Declarative Method として Halo3 へ

Halo3 9講演のPPT資料 <http://www.bungie.net/Inside/publications.aspx>

Sims の話へ戻って...



Edith

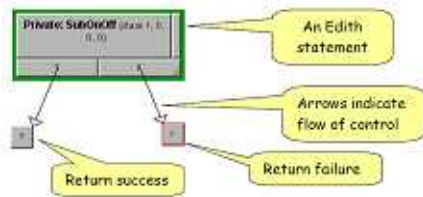
これだけだと、
原始的で単発の行動しかできないけれど...

実際は、
一連の行動のシーケンスをAIに行わせたい。

そのためのビジュアル・プログラミング環境が、
Edith

Edith

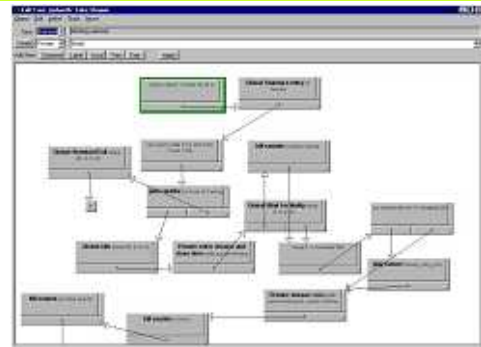
プログラミング・オブジェクトを繋げて行く
ビジュアル・プログラミング環境



Kenneth D. Forbus "Some notes on programming objects in The Sims"
http://www.qrg.northwestern.edu/papers/Files/Programming_Objects_in_The_Sims.pdf

Edith

プログラミング・オブジェクトを繋げて行くビジュアル・プログラミング環境



Kenneth D. Forbus "Some notes on programming objects in The Sims"
http://www.qrg.northwestern.edu/papers/Files/Programming_Objects_in_The_Sims.pdf

Edith Demo

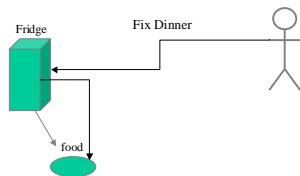


<http://www.DonHopkins.com/home/movies/TheSimsPieMenus.mov>

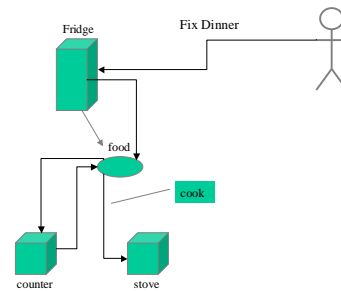
実例 「調理して食べる」



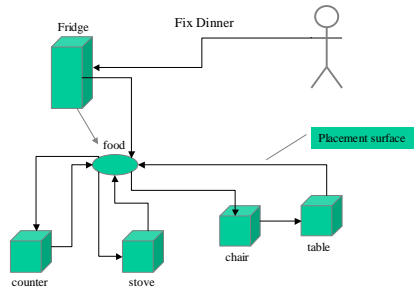
実例 「調理して食べる」



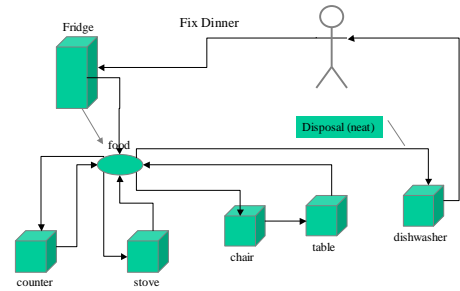
実例 「調理して食べる」



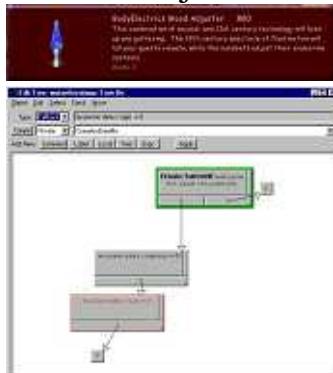
実例 「調理して食べる」



実例 「調理して食べる」



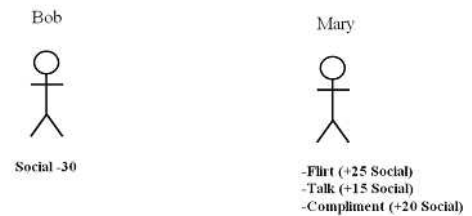
Edith for Mood Adjuster (パーティ盛り上げグッズ)



Kenneth D. Forbus "Some notes on programming objects in The Sims"

http://www.grg.northwestern.edu/papers/Files/Programming_Objects_in_The_Sims.pdf

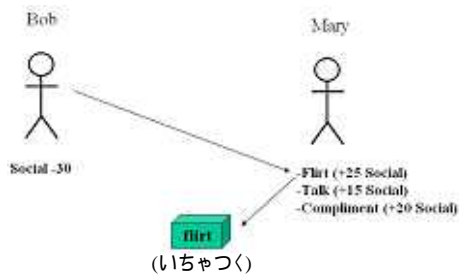
人同士のコミュニケーション



Kenneth D. Forbus "Some notes on programming objects in The Sims"

http://www.grg.northwestern.edu/papers/Files/Programming_Objects_in_The_Sims.pdf

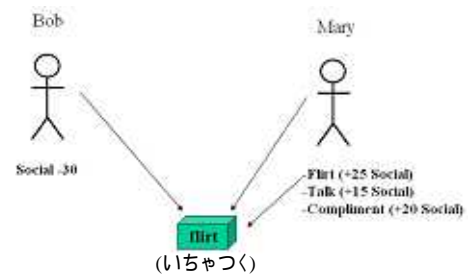
人同士のコミュニケーション



Kenneth D. Forbus "Some notes on programming objects in The Sims"

http://www.grg.northwestern.edu/papers/Files/Programming_Objects_in_The_Sims.pdf

人同士のコミュニケーション



Kenneth D. Forbus "Some notes on programming objects in The Sims"

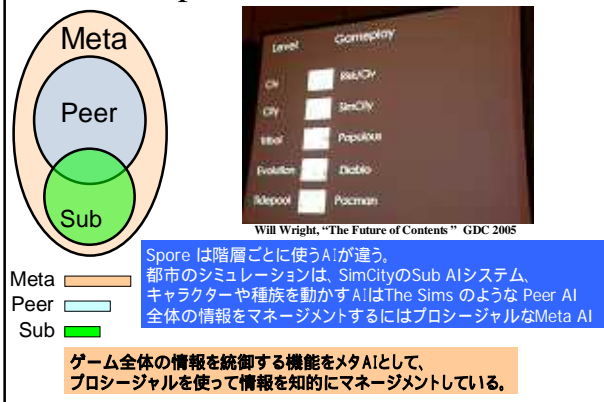
http://www.grg.northwestern.edu/papers/Files/Programming_Objects_in_The_Sims.pdf



Spore AI

- Spore Peer AI
- Spore Meta AI

SporeのAIの作り方



7層に分かれたステージと作成コンテンツ

<http://www.gamespot.com/pc/strategy/spore/news.html?sid=6165667>

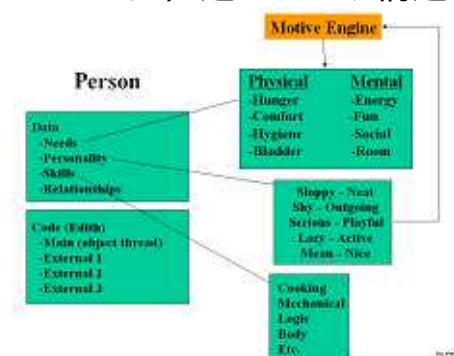


Spore Peer AI

(予想)

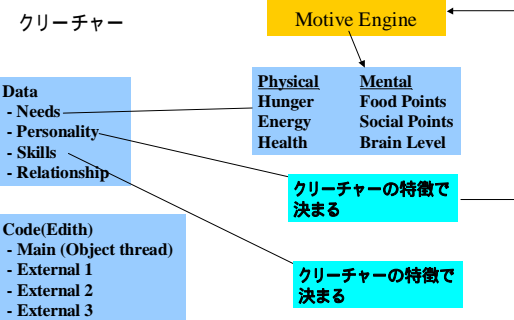
Spore のキャラクター制御

NPCに仕込むデータ構造



Ken Forbus, "Simulation and Modeling: Under the hood of The Sims" (NorthWestern大学、講義資料)
http://www.cs.northwestern.edu/%7Eforbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/frame.htm

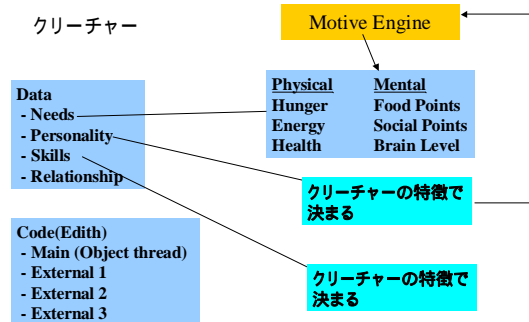
NPCに仕込むデータ構造(予想)



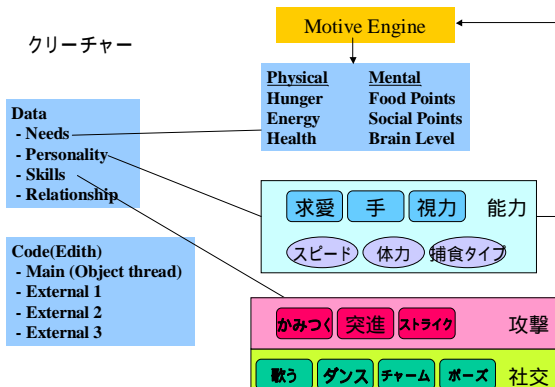
Sporeクリーチャークリエーターから自動決定されるパラメータ



NPCに仕込むデータ構造(予想)



NPCに仕込むデータ構造(予想)



オブジェクトに仕込むデータ構造

Data (Class, State)

Graphics (sprites, z-Animations (skeletal))

Sound Effects

Code (Edith)
-Main (object thread)
-External 1
-External 2
-External 3

パラメーター

グラフィックス
アニメーション

サウンド

メインスレッド

いろいろなインタラクションの仕方



<http://www.4gamer.net/games/020/G002026/20080610033>

クリーチャー同士のコミュニケーション



<http://www.4gamer.net/games/020/G002026/20080610033/>

クリーチャー同士のコミュニケーション



<http://www.4gamer.net/games/020/G002026/20080610033/>

クリーチャー同士のコミュニケーション



<http://www.4gamer.net/games/020/G002026/20080610033/>

Spore Tribal Phase Demo



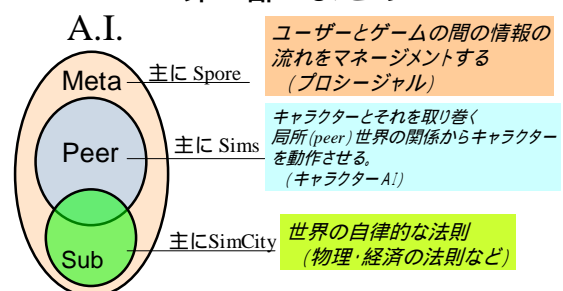
<http://www.4gamer.net/games/020/G002026/20080610033/>

Spore Meta AI (プロシージャルなど)

...は、まとめて第2部へ。

とりあえず、第1部まとめへ...

第1部 まとめ



ウィル・ライトはゲーム内で独立したファクターとして実装した状況に応じて運動する法則、機能を「A.I.」と定義する。

第2部

プロシージャル・コンテンツ・ジェネレーションと Spore

PCG (Procedural Contents Generation) & Spore

Spore Meta AI

はじめに

プロシージャル・コンテンツ・ジェネレーションは、
ゲーム開発の黎明期からあった話である。

しかし、あくまで亜流であり、
ハードウェアの制約からゲームの
限られた部分に使われて来たただだった。
(ダンジョンとかメロディー生成とか...)

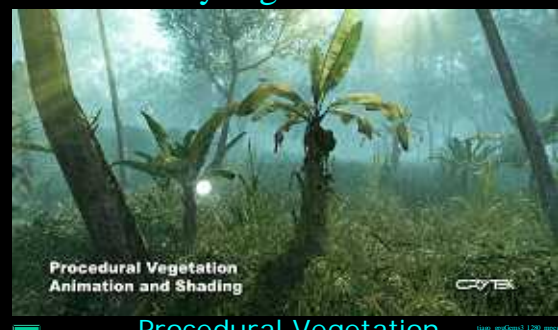
現代のゲーム開発において、この技術に
新しい意味付けをして導入のインパクトを作ったのは、

CryEngine (Crytek) と *Spore (Maxis,EA)*

現在のプロシージャル2大エンジン



CryEngine デモ



Procedural Vegetation
Animation in Crysis

Tiago Sousa, "Chapter 16: Vegetation Procedural Animation and Shading in Crysis", GPU Gems 3, 2007

DUNIA Engine デモ

(I) Growing Vegetation

(II) Tree Generation

(III) Day and Night cycle

(IV) Realistic Weather System

Far Cry 2 Dunia Engine Tech Videos and Screens
<http://www.n4g.com/pc/News-89415.aspx>

はじめに

CryEngine (Crytek) はドイツの企業らしく、
綿密な積み上げのもとに開発された
プロシージャル・ゲームエンジンだったが、

Spore (Maxis,EA) におけるプロシージャル技術は、
ウィル・ライト (Maxis) 氏が、
個人的な思想と熱狂のもとに
彼の作家性を起点として、
起こしたものであった。

ここでは、ウィル・ライト (Maxis) の思想を通じて、
Spore のプロシージャル技術を解説して行きたい。

第1章 プロシージャルとは何か？

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

プロシージャル・コンテンツ・ジェネレーションとは？

ゲーム空間、デジタル空間において、
計算によってあらゆるコンテンツを自動生成すること

(注) プロシージャル・コンテンツ・ジェネレーションは頭文字を取って、
PCG と略される場合があります。

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

アルゴリズム

プログラムによって
モデルを自動生成する



プロシージャル

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)

最も身近なプロシージャル

...それは、3DCG！



<http://www.tekapon.net/fsm/>

ドッターがドットを打つ
= 一枚の絵

3Dデータ x カメラ
= 無限にたくさんの絵
CGアニメーションによる
低コスト化

3DCGは自動的(Procedural)に絵を作ってくれるシステム

Demoscene (.farbrausch)



- 64kbから映像を生成する

<http://www.farb-rausch.com>



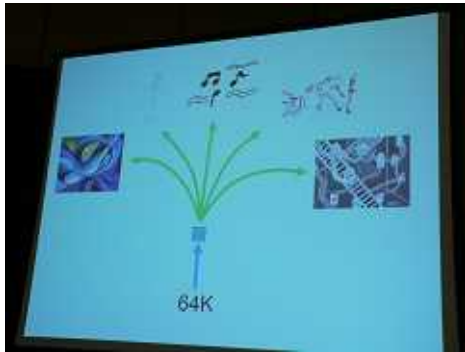
デモ

<http://www.farb-rausch.com>



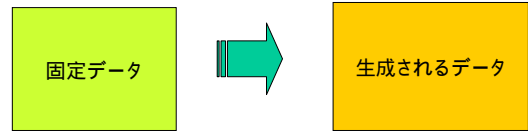
fr-08: .the .product theDemoscene/fr08_final
.farbrausch demos

デモシーナー が教えてくれること



Will Wright, "The Future of Contents" GDC 2005

デモシーナー が教えてくれること



特徴 作成コスト
完全にコントロール
ロード時間
容量が大きい

特徴 容量が小さい
アルゴリズムで自動生成
計算量が多い(ロードではなく計算)



(1)「コンテンツ=データ」ではない。それは生成できるものだ。
(2)大きな(巨艦大砲的な)ゲームも**プロシージャル**に作成すれば
手間もコストもデータ量も抑えて作ることができる

新しいゲーム製作のコンセプト

デモシーナーからSporeへ

デモシーナー

64k

プロシージャル
による展開



Spore

100k (png形式)

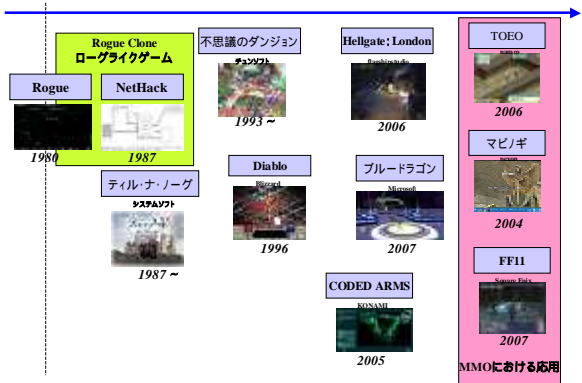
プロシージャル
による展開



ゲームにとってプロシージャルとは何か？

プロシージャルの歴史: ダンジョン自動生成

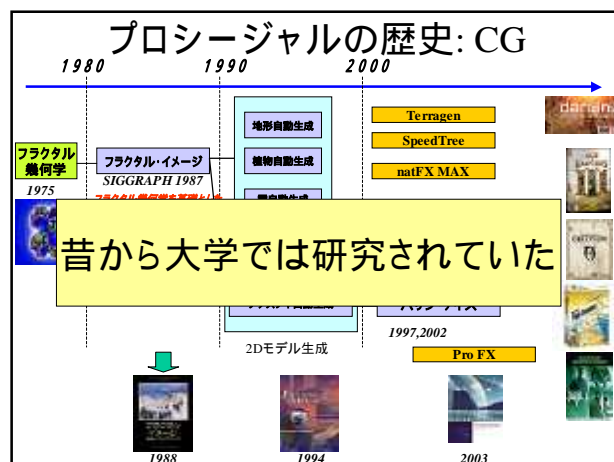
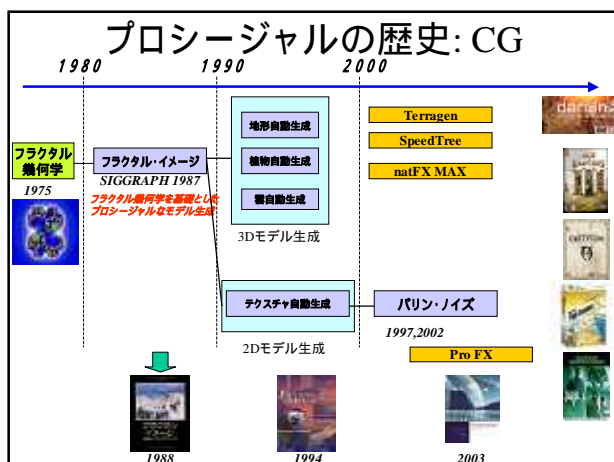
1980



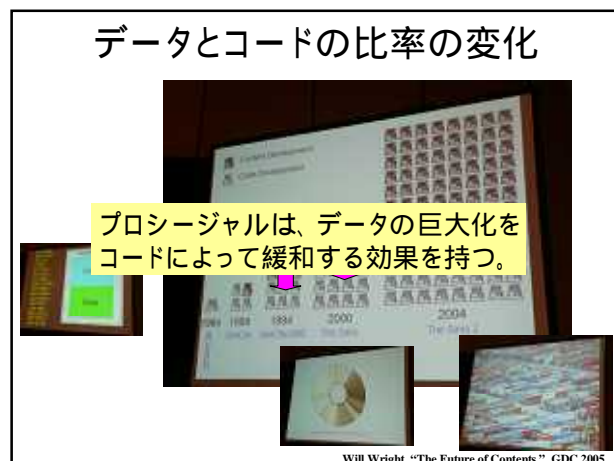
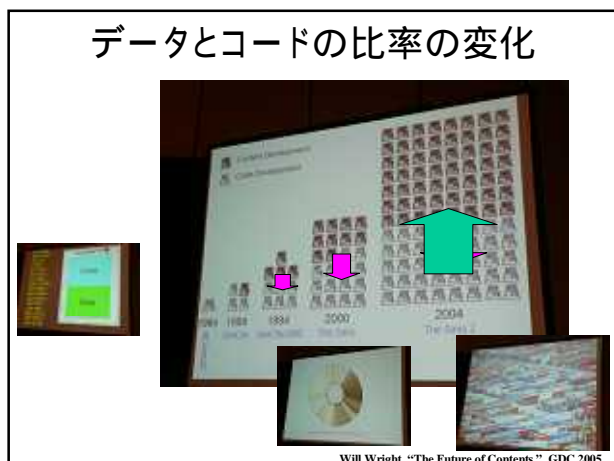
プロシージャルの歴史: ダンジョン自動生成

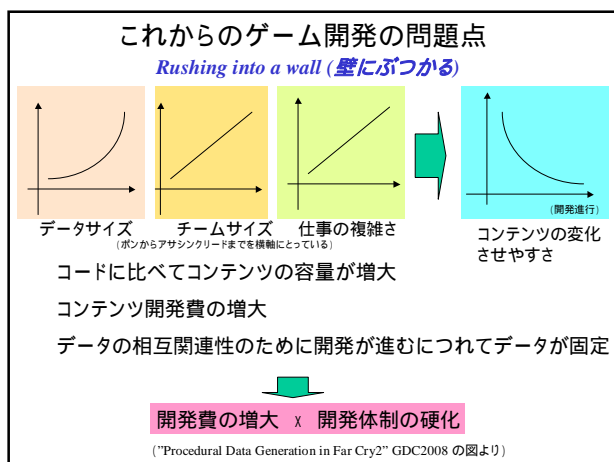
1980





今、ゲームにとってプロシージャルとは何か？





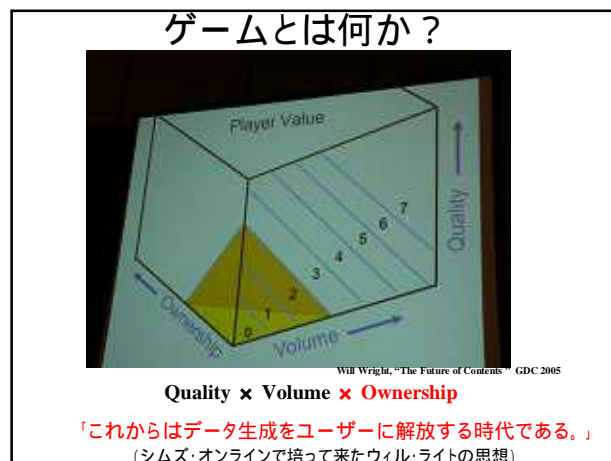
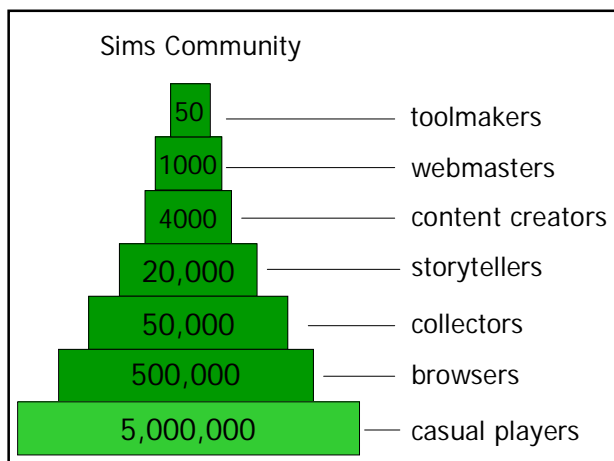
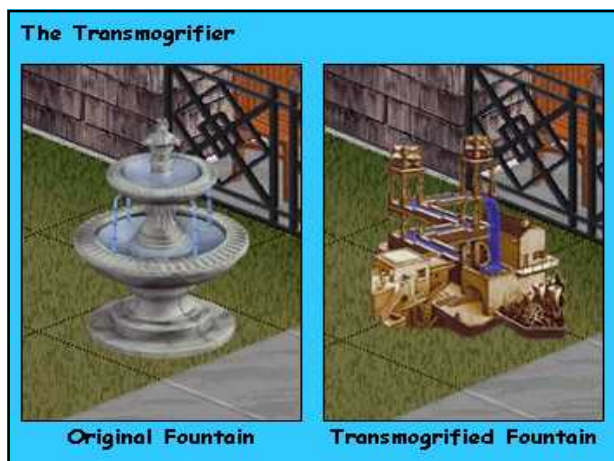
ウィル・ライトの思想を追う

コンテンツが変化させやすく、
 大量のデータを作って、
 それが手の込んだ高品質で、
 開発費のかさまない

そんな開発を可能にする技術は何か？

プロシージャル・コンテンツ・ジェネレーション

...だけでは足りない。



ウィル・ライトの思想を追う

コンテンツが変化させやすく、
大量のデータを作って、
それが手の込んだ高品質で、
開発費のかさまない

そんな開発を可能にする技術は何か？

プロシージャル・コンテンツ・ジェネレーション
×
ユーザー・ジェネレイティッド・コンテンツ

ウィル・ライトの思想を追う

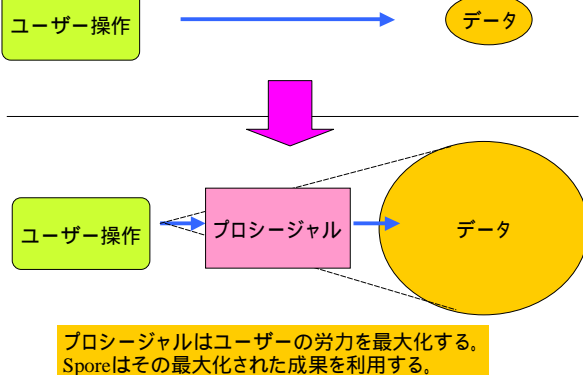
コンテンツが変化させやすく、
大量のデータを作って、
それが手の込んだ高品質で、
開発費のかさまない

そんな開発を可能にする技術は何か？

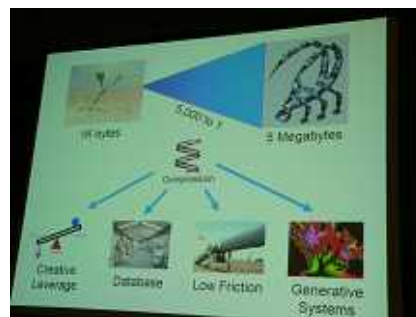
Spore

プロシージャル・コンテンツ・ジェネレーション
×
ユーザー・ジェネレイティッド・コンテンツ

Spore のゲームデザインの本質

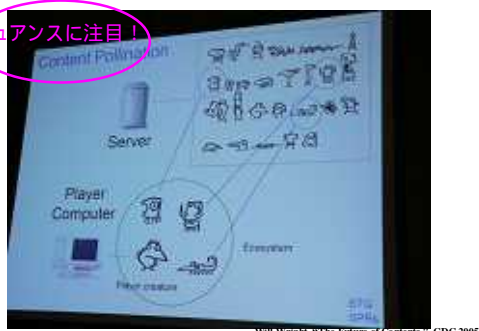


Spore のゲームデザインの本質



Spore のゲームデザインの本質

このニュアンスに注目！



Pollination = 配布 (もとの意味は花粉を飛ばす)

技術詳細

Spore とはどんなゲーム？

自分だけの宇宙を作って行くゲーム

生物、建物、惑星、主要なものは全てユーザーが生成可能

単細胞フェーズ

集団行動フェーズ

ギャラクティックフェーズ

<http://www.4player.net/news/feature/2006.05/20060511105155denji.html>

「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 (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)

(1) モデル生成

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

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)

Lydia Choy, Ryan Ingram, Ocean Quigley, Brian Sharp, Andrew Willmott
Rigblocks: Player-Deformable Objects, <http://www.cs.cmu.edu/~ajw/s2007>

(2) 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

UV

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

(3) 星を創る

Andrew Willmott, Creating Spherical Worlds, <http://www.cs.cmu.edu/~ajw/s2007>

表面テクスチャ、法線マップ

カラーマップ

射影 (ヤコビ行列)

法線マップ

プロシージャル・ブラッシング

惑星の表面をブラシで凹凸をつけることができる

Andrew Willmott, Creating Spherical Worlds, <http://www.cs.cmu.edu/~ajw/s2007>

Procedural Brushing Demo



3rd spore demonstration at SIGGRAPH 07 (YouTube)

(4) 惑星表面に植物を植える

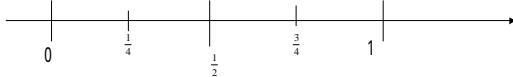


Andrew Willmott, Creating Spherical Worlds, <http://www.cs.cmu.edu/~ajw/s2007>

ハルトン・シーケンス

ハルトン・シーケンス

ベースとなる素数を一つ選ぶ。例えば2
その素数で、 $(0, 1)$ 区間を選んだ素数の数に分割する
分けた区間をさらに素数の数に分割する。
ここでは $(0, 1/2)$ $(1/2, 1)$ をさらに2分割する



$1/2, 1/4, 3/4, 1/8, 3/8, 5/8, 7/8, \dots$

1次元ハルトン・シーケンス

$1/3, 1/9, 2/9, 4/9, 5/9, 2/3, 7/9, 8/9, \dots$

$(1/2, 1/3), (1/4,)$

ハルトン・シーケンス

ハルトン・シーケンス

1次元ハルトン・シーケンス

$1/2, 1/4, 3/4, 1/8, 3/8, 5/8, 7/8, \dots$

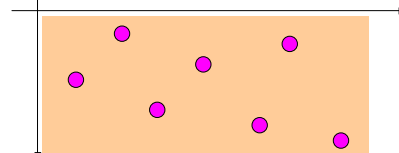
$1/3, 1/9, 2/9, 4/9, 5/9, 2/3, 7/9, 8/9, \dots$



$(1/2, 1/3), (1/4, 1/9), (3/4, 2/9), (1/8, 4/9) \dots$

2次元ハルトン・シーケンス

大きい素数を使わなければだいたい均等に分布する



実際は、割り算をビットシフト演算に変えた
高速な Incremental Halton Sequence を使う



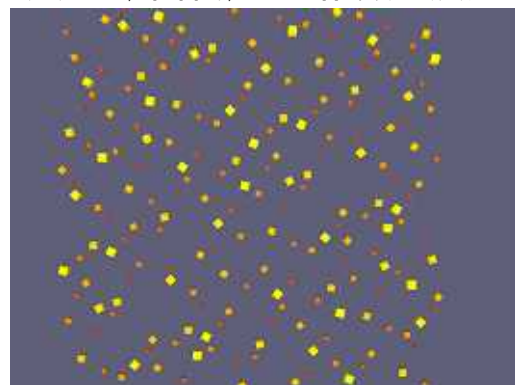
20

100

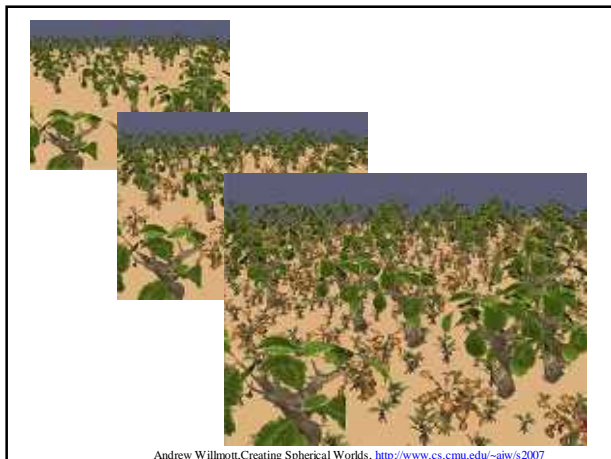
500

Andrew Willmott, Creating Spherical Worlds, <http://www.cs.cmu.edu/~ajw/s2007>

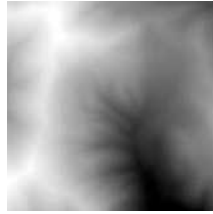
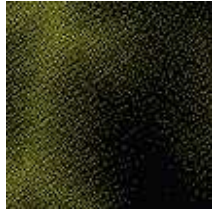
大きさ、回転、色を乱数で決定



Andrew Willmott, Creating Spherical Worlds, <http://www.cs.cmu.edu/~ajw/s2007>



密度マップと分布

密度マップ
(惑星表面全体に渡る植物の密度)

分布結果

全体の分布を、各位置(小エリア)の密度を越えないように、植林して行く

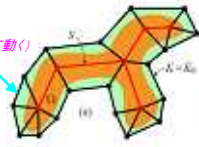
Andrew Willmott, Creating Spherical Worlds, <http://www.cs.cmu.edu/~ajw/s2007>

(5) アニメーション (Development of Spore の情報による)

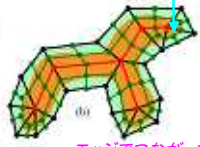
スケルトン・ドリブン・アニメーション

骨を動かすことで、自動的に計算で、その周りに肉の部分も動かすテクニック

コントロールラティス
(骨と同期して動く)



骨を動かすと



エッジでつながった肉も運動して動くのだ

有限要素法

頂点をエッジ(辺)でつないで、頂点同士の運動関係を定義してシミュレーションする技術

Steve Capell et al., Interactive Skeleton-Driven Dynamic Deformations
<http://grail.cs.washington.edu/projects/deformation/>

スケルトンとコントロールラティスの同期



色ごとに
局所座標系が
入っています



スケルトン

● コントロールラティス

有限要素法の非線型方程式(複雑)を解いてシミュレーション

Steve Capell et al., Interactive Skeleton-Driven Dynamic Deformations
<http://grail.cs.washington.edu/projects/deformation/>

デモ



Capell-2002-ISD-dvxx

Steve Capell et al., Interactive Skeleton-Driven Dynamic Deformations
<http://grail.cs.washington.edu/projects/deformation/>

スケルトン・ドリブン・アニメーション

一応、この紹介した論文が、*Spore* に使われているらしいと、

Development of Spore Wiki
http://en.wikipedia.org/wiki/Development_of_Spore

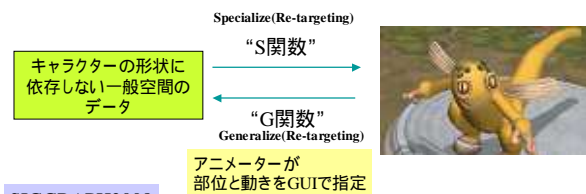
に記載されていた。さらに、最近、以下のPreceding

“Real-time Motion Retargeting to Highly Varied User-Created Morphologies”
http://chrishecker.com/Real-time_Motion_Retargeting_to_Highly_Varied_User-Created_Morphologies

の内容がSIGGRAPH2008 で発表されるとのことです。

Spore アニメーション要件

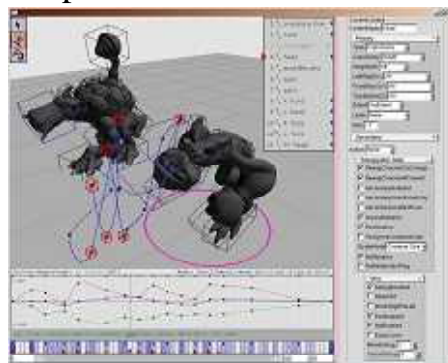
キャラクターが作成したクリーチャーの形状にあわせて、アニメーションデータを再作成(re-targeting)する。



SIGGRAPH2008

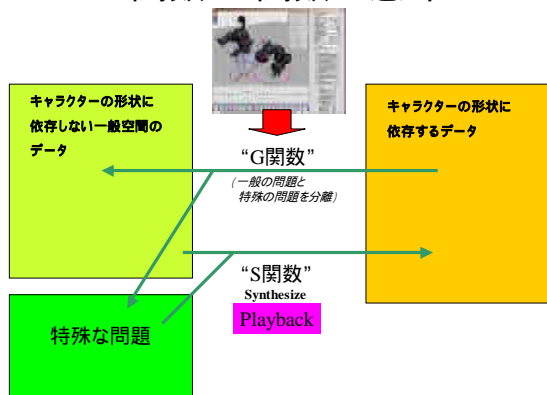
"Real-time Motion Retargeting to Highly Varied User-Created Morphologies"
http://chrishecker.com/Real-time_Motion_Retargeting_to_Highly_Varied_User-Created_Morphologies

Spore アニメーション要件

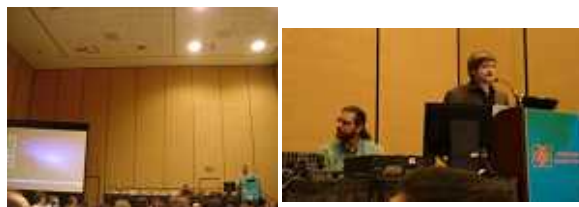


G関数生成のための情報をインプット

G関数とS関数の意味



(6) SPORE におけるプロシージャル・ミュージック



SPORE におけるプロシージャル・ミュージック (GDC2008)
 Procedural Music in SPORE
<https://www.cmcpevents.com/GD08/a.asp?option=C&V=11&SessID=6426>
 講演者: Kent Jolly (Audio Director, Electronic Arts), Aaron McLeran (composer, n/a)
 日時: 2月20日(水) 12時~13時
 場所: West Hall, 2024号室

プロシージャル・ミュージック

- 音楽の自動生成
- 環境音楽 (Ambient Music)

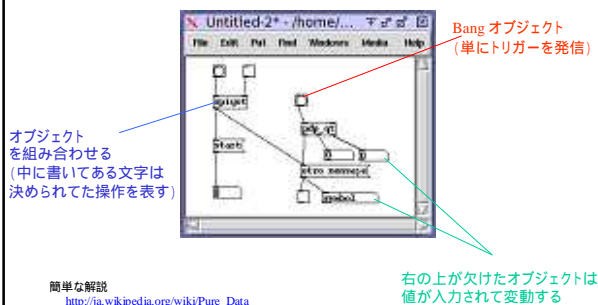
くり返しのない(never repeat)音楽を目指す

Procedural Music
 (その場で生成される音楽)

そのステージでユーザーとインタラクションしながら生成される音楽

Pd とは？

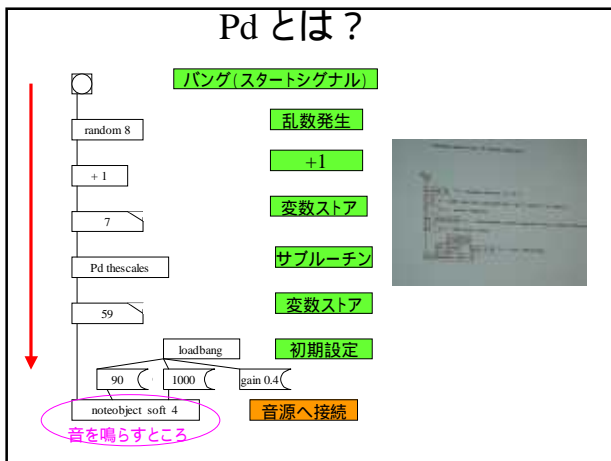
Pdとは、グラフィカルな環境で関数や変数を結線することで制御フローや(音)信号の流れを定義するビジュアルプログラミング言語の一つ
 Miller Puckette によって1990年代に開発されたフリーの言語



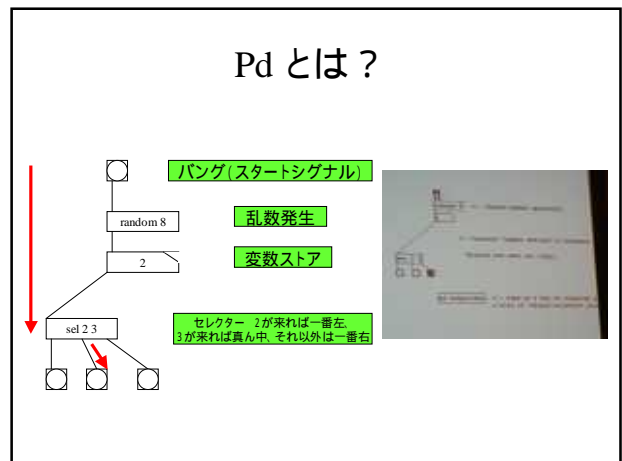
簡単な解説
http://ja.wikipedia.org/wiki/Pure_Data

右の上が欠けたオブジェクトは
 値が入力されて変動する

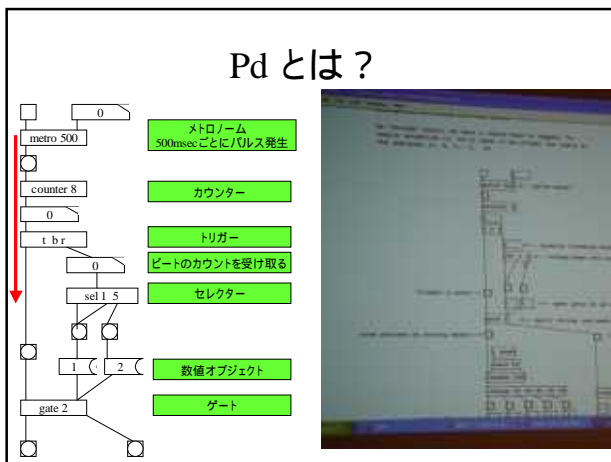
Pd とは？



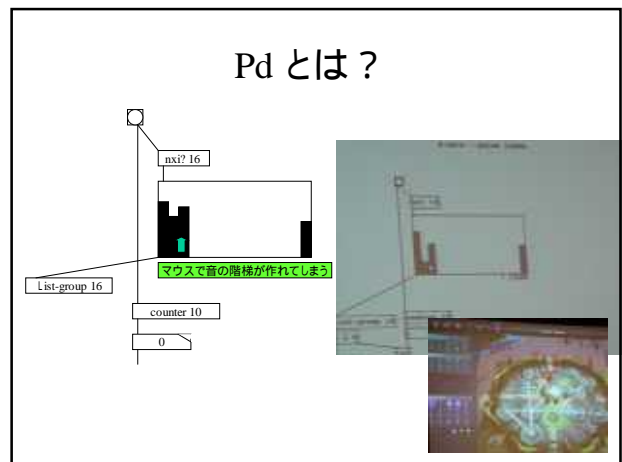
Pd とは？



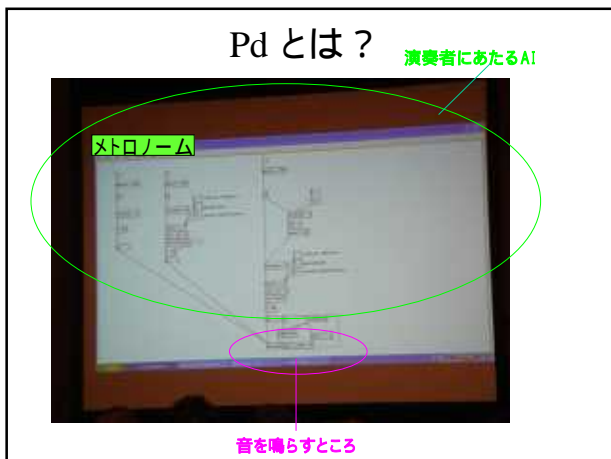
Pd とは？



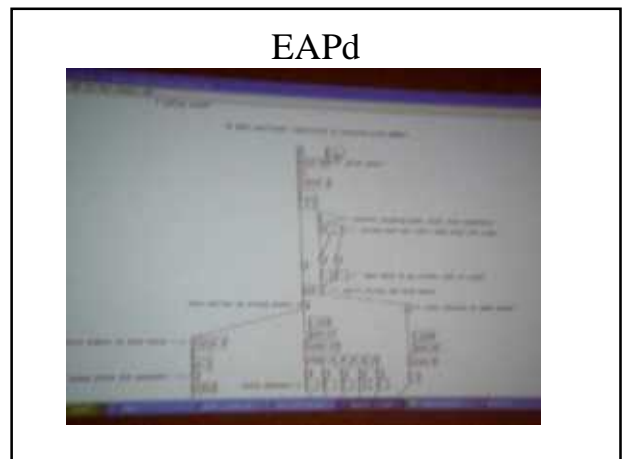
Pd とは？



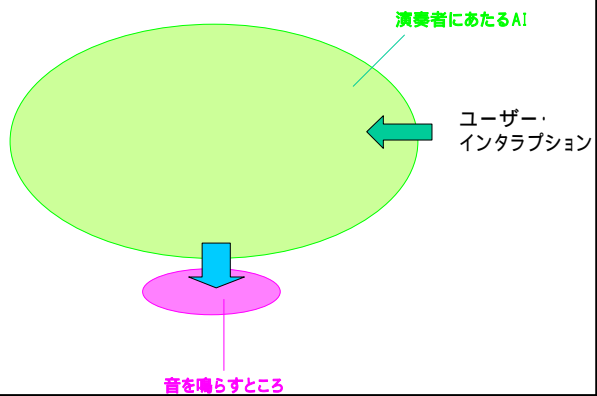
Pd とは？



EAPd



Pd とゲームサウンド



Spore における実現例



ユーザーのマウス・アクションに応じて音が変わって行く
(アイコンの上をマウスがクロスオーバーするだけで音が変わる)

Spore における実現例



なんで今までなかったのだろう？ ゲームにとっても馴染んだ技術

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！

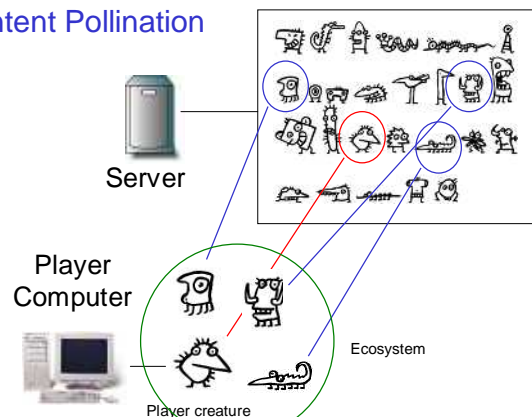
7ステージにおけるプロシージャル技術

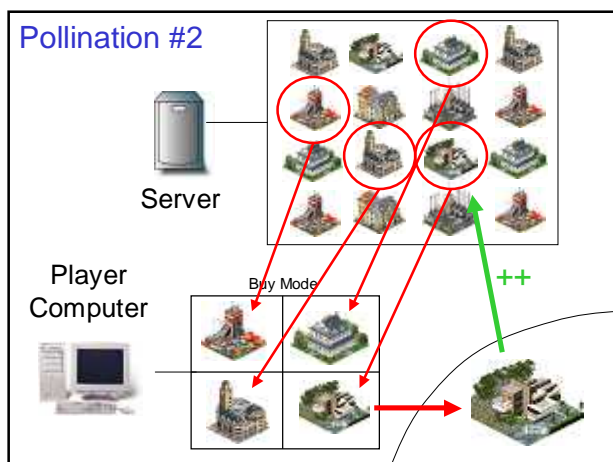
Scale	Player Creation	Technology
Galactic	Colonies	?
Teraform	Planet	地形自動生成、自動分布
Civilization	Vehicles	リグブロック、自動テクスチャ、自動アニメーション
City	Buildings	リグブロック、自動テクスチャ、自動アニメーション
Tribal	Tools	Peer AI、繁殖
Creature	Animal	リグブロック、自動テクスチャ、自動アニメーション
Cellular	Cell	?
Molecular	Molecule	?

+ 各ステージのエディターにおけるプロシージャル・ミュージック

<http://www.gamespot.com/pc/strategy/spore/news.html?sid=6165667>

Content Pollination





References

- (1)カーネギー・メロン大学の [Andrew J. Willmot博士のHPページ](http://www.cs.cmu.edu/~ajw/s2007/) のSIGGRAPH2007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/2006.05/20060511195155_detail.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。
またリファレンスから、多くの情報を得ることができます)
http://en.wikipedia.org/wiki/Development_of_Spore

Figures on the pages are from these references.

Spore の技術情報は

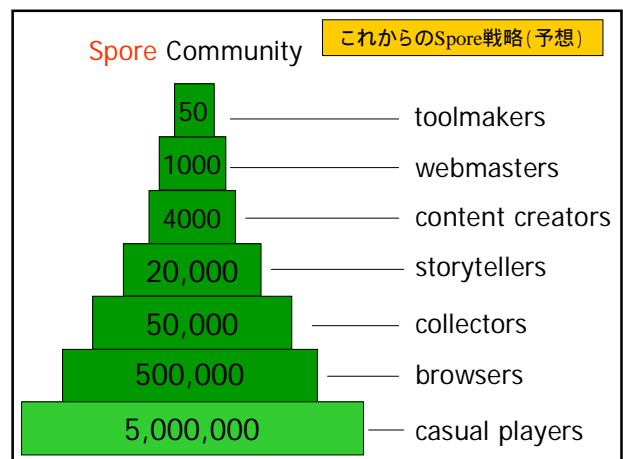
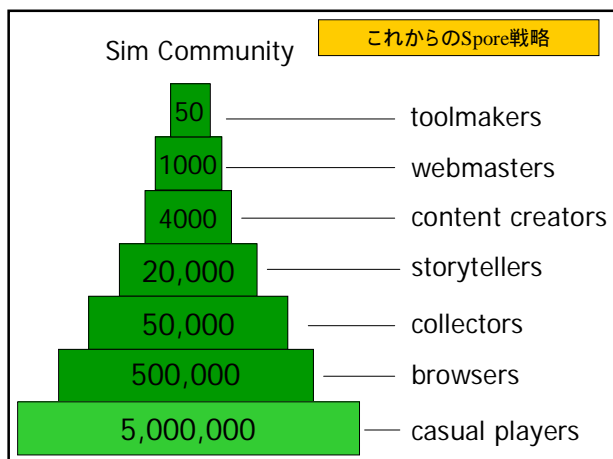
詳しくはゲームAI連続セミナー 第6回資料へ

(IGDA日本の「ダウンロード」にテキストがあります)

<http://www.igda.jp/modules/mydownloads/>

+

カーネギー・メロン大学の Andrew J. Willmot博士のHPページのSIGGRAPH2007のコーナー
<http://www.cs.cmu.edu/~ajw/s2007/>



これからのSpore戦略

(1) Spore のブランド化 (GDC2008で発表)

- Sim シリーズのように連作。

<http://www.4gamer.net/games/047/G004713/20080222067/>

(2) マルチプラットフォーム化

DS など。

「Spore」から見るこれからのEA戦略

プロシージャル技術を自社技術として取り入れて行き、
各開発段階を効率化、高品質化、自動化を行う。



知的機能を利用したゲーム開発工程

ハイ・インテリジェントな開発工程



次世代ゲームエンジン

プロシージャル・ゲームエンジン

(CryEngine, DUNIA Engine...)

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

この資料はIGDA日本のサイトにアップされます。

ご清聴ありがとうございました。



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)

WEB上の意見交換にはIGDA Japanのサイトをご利用ください

<http://www.igda.jp>