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.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.

References (2) Maxis 開発者たちの情報①

(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"
(NorthWerstern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②

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。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

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

① The SimCity Sub AI
② The Sims Peer AI
③ 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)

The Sims 3 (2008)

デビュー

シムズ
時代

Spore
時代

Maxis

EA, Maxis

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.

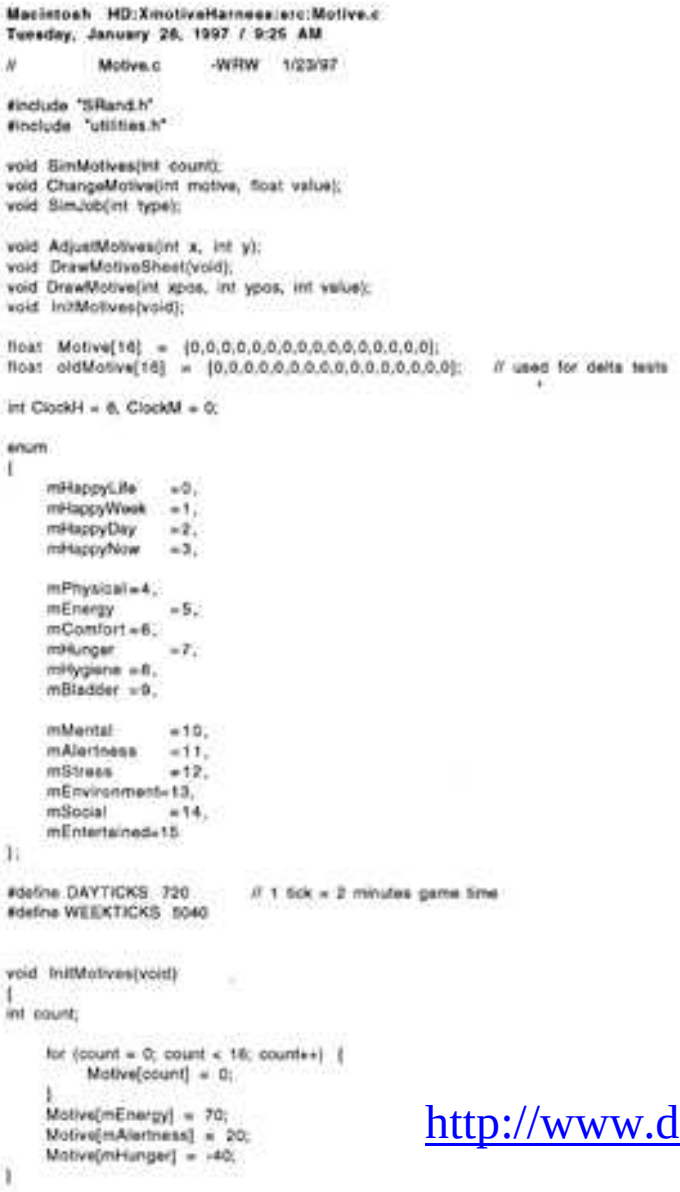
- Will Wright, from an interview with Wired magazine, 1994.

**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



Macintosh HD:XmotiveHarness:src:Motive.c
Tuesday, January 26, 1997 / 9:25 AM

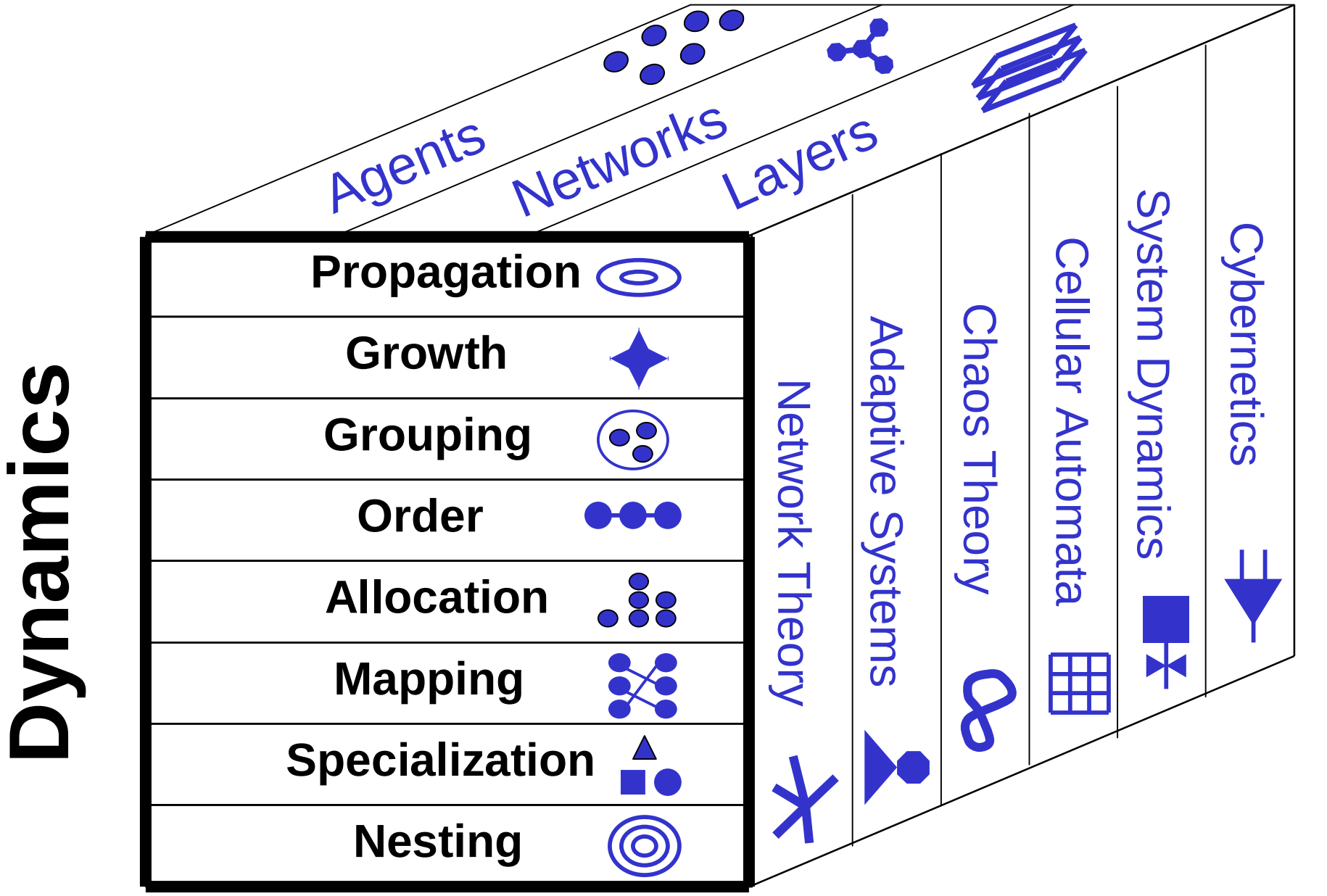
Page: 1

```
//          Motive.c        -WRW   1/23/97

#include "SRand.h"
#include "utilities.h"

void SimMotives(int count);
void ChangeMotive(int motive, float value);
void SimJob(int type);

void AdjustMotives(int x, int y);
void DrawMotiveSheet(void);
void DrawMotive(int xpos, int ypos, int value);
void InitMotives(void);

float Motive[16] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
float oldMotive[16] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};    // used for delta tests

int ClockH = 6, ClockM = 0;

enum
{
    mHappyLife    =0,
    mHappyWeek    =1,
    mHappyDay     =2,
    mHappyNow     =3,

    mPhysical=4,
    mEnergy       =5,
    mComfort=6,
    mHunger       =7,
    mHygiene =8,
    mBladder =9,

    mMental       =10,
    mAlertness    =11,
    mStress       =12,
    mEnvironment=13,
    mSocial       =14,
    mEntertained=15
};

#define DAYTICKS  720       // 1 tick = 2 minutes game time
#define WEEKTICKS  5040


void InitMotives(void)
{
int count;

    for (count = 0; count < 16; count++)  {
        Motive[count] = 0;
    }
    Motive[mEnergy] = 70;
    Motive[mAlertness] = 20;
    Motive[mHunger] = -40;
}
```

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)

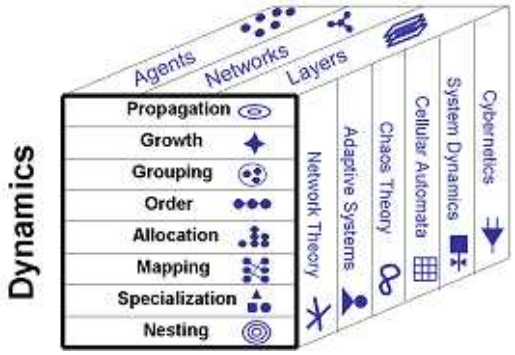
プログラマーらしい、
抽象的な動的な流れを追う能力、
プログラムの可能性をよく知っている。

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


Dynamics
Propagation
Growth
Grouping
Order
Allocation
Mapping
Specialization
Nesting
Agents
Networks
Layers
Network Theory
Adaptive Systems
Chaos Theory
Cellular Automata
System Dynamics
Cybernetics
topology change through time

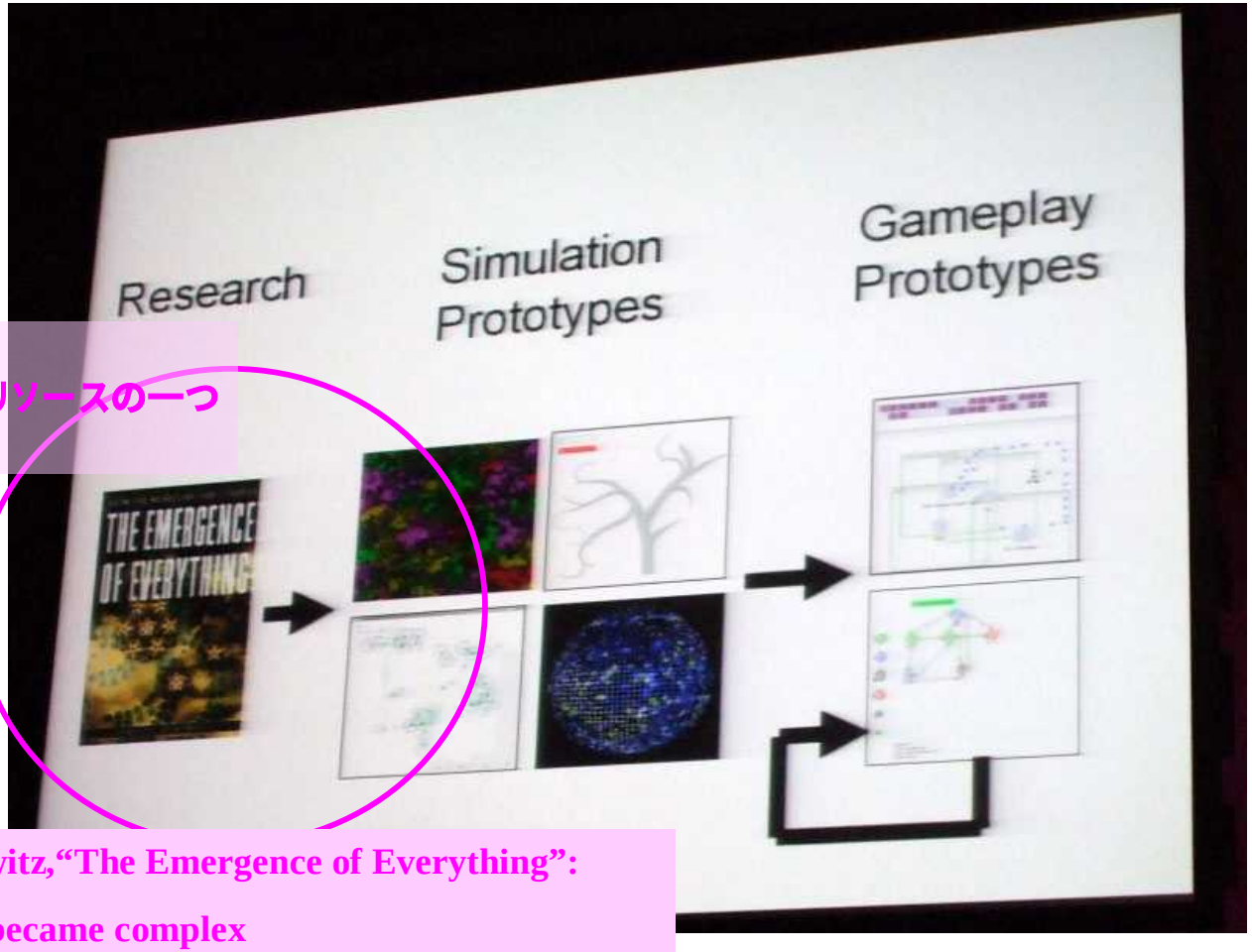
“ウィル・ライト” の知性の特徴

抽象的的な思考を、具体的なイメージへ投影できる力を持つ。
イメージからイメージへメタファーを渡って行く思考が出来る。



抽象的思考から具象事例へ、さらにその逆、というサイクルは実は結構、負荷の高い力のいる思考。彼はそれを膨大な雑多なイメージによって補う。

リサーチの重要性



たぶん、
Sporeの発想のリソースの一つ

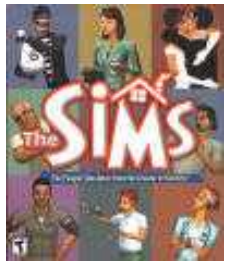
Harold.J.Morowitz,“The Emergence of Everything”:
How the world became complex

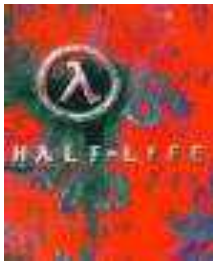
興味を持ったことを徹底してリサーチする能力、
そこから中核となるプロトタイプを自らプログラムする能力。

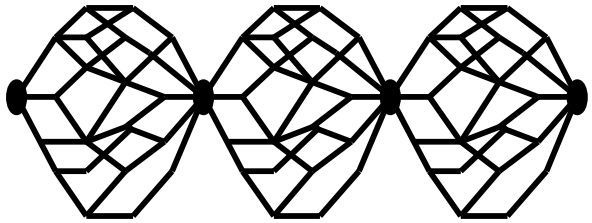
http://www.4gamer.net/games/020/G002026/20060324212940/

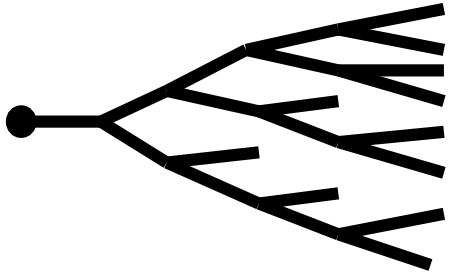
“ウィル・ライト” の作るゲーム

決して直線的ではなく、様々な方向へプレイする自由度を持ちながら、確かにゲームとしてコントロールされている。









Dense

Open-ended

Sparse

Linear

Model Building

Play

cognitive technologies

Story

Topology

Dense

Linear

Drama

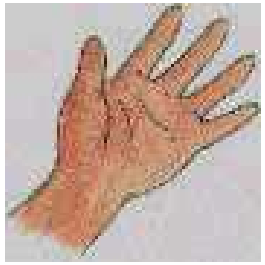
Flat

Compelling

Appeal

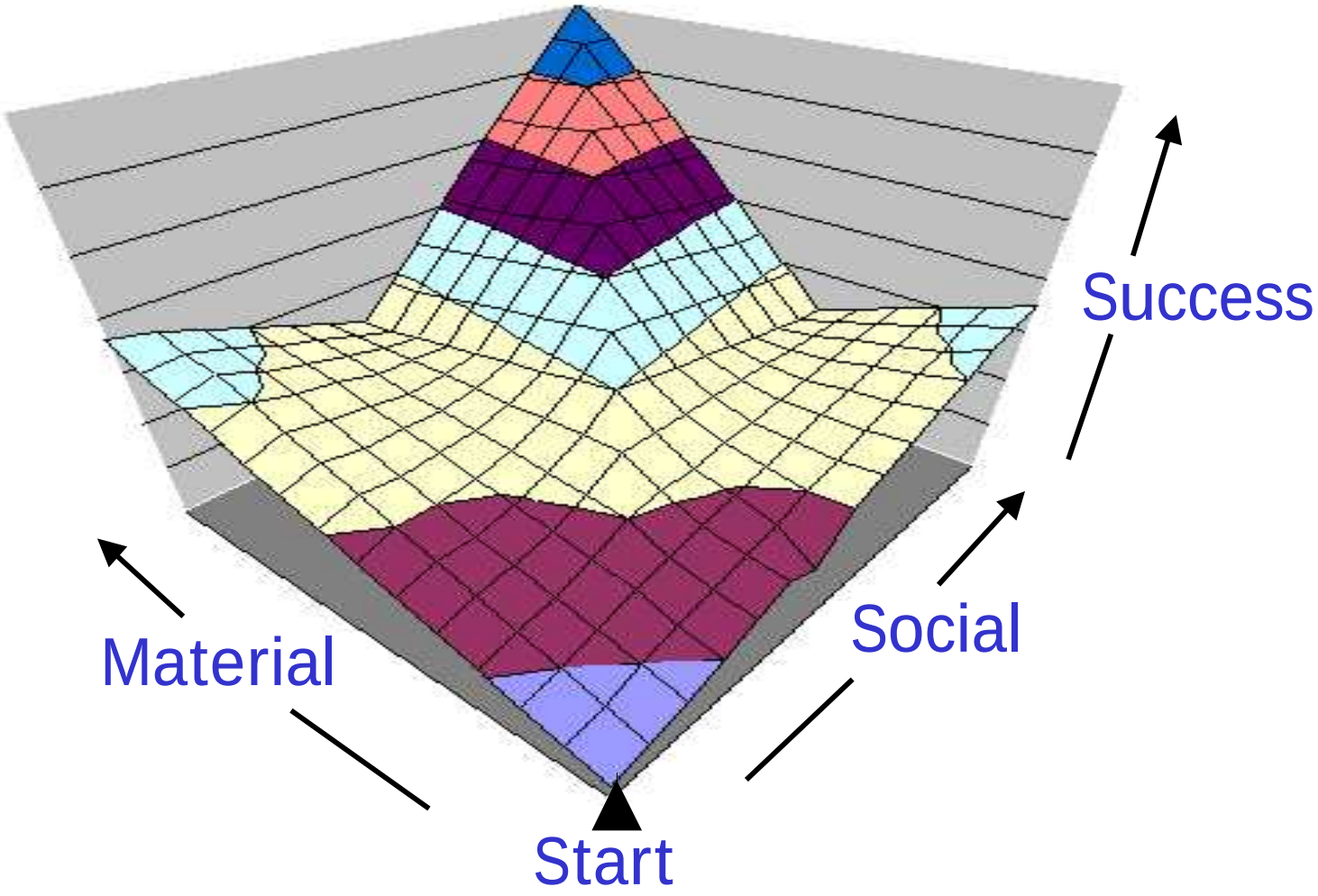
Agency

Empathy

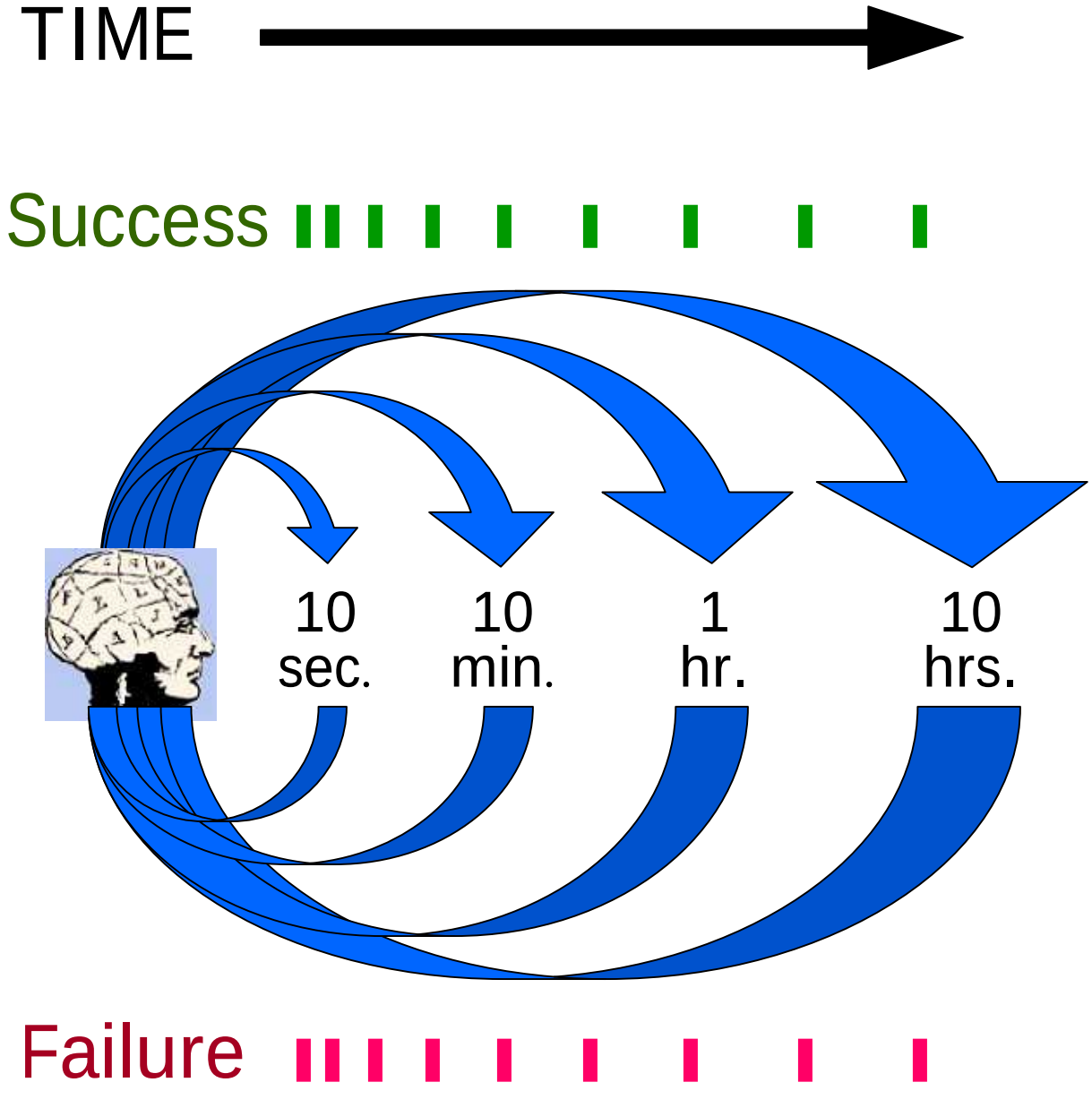


abstraction

Gameplay Landscape (Sims)



ゲームを楽しむ様々な経路が密度高く無限に用意されていて、その一つ一つが大きく、そして小さく違う。



Patterns in time

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

① The SimCity Sub AI
② The Sims Peer AI
③ Spore AI

“ウィル・ライト” のゲームAI論

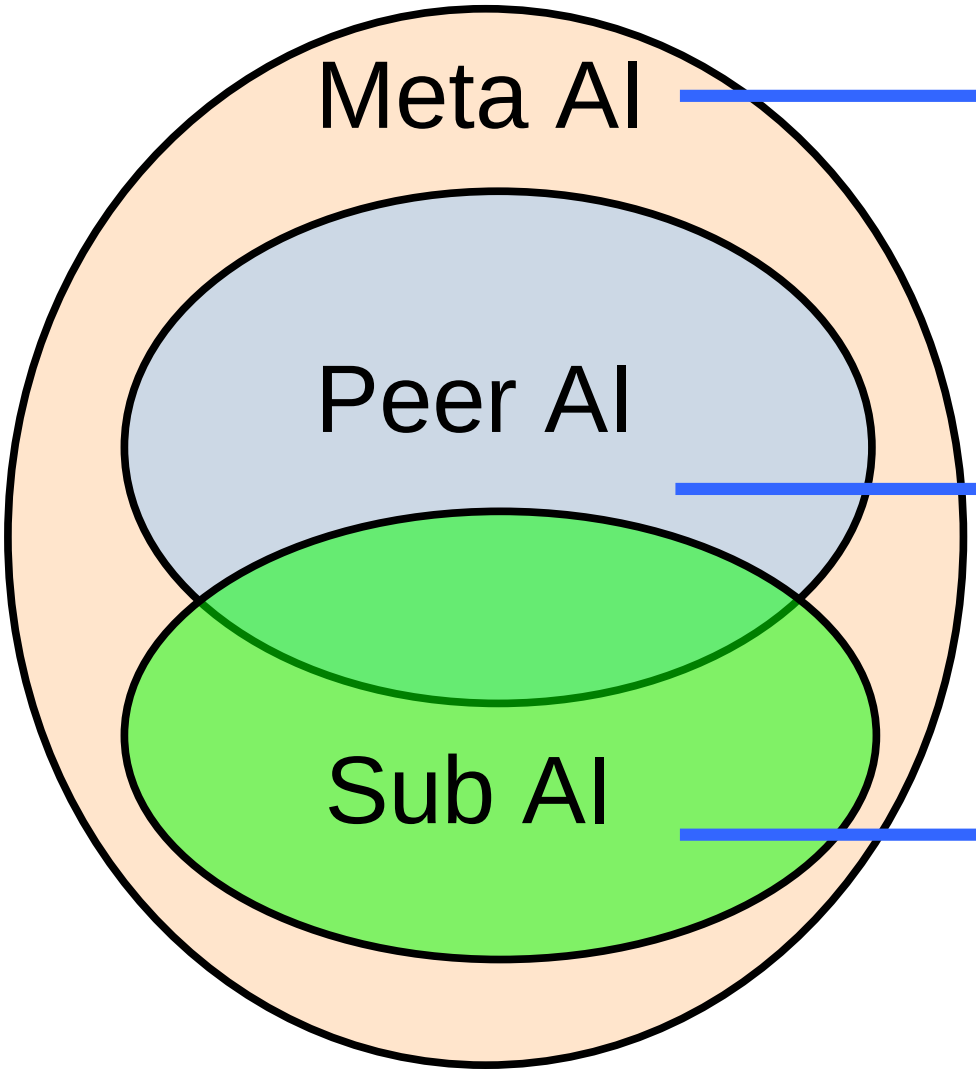
① The SimCity Sub AI
② The Sims Peer AI
③ Spore AI

“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

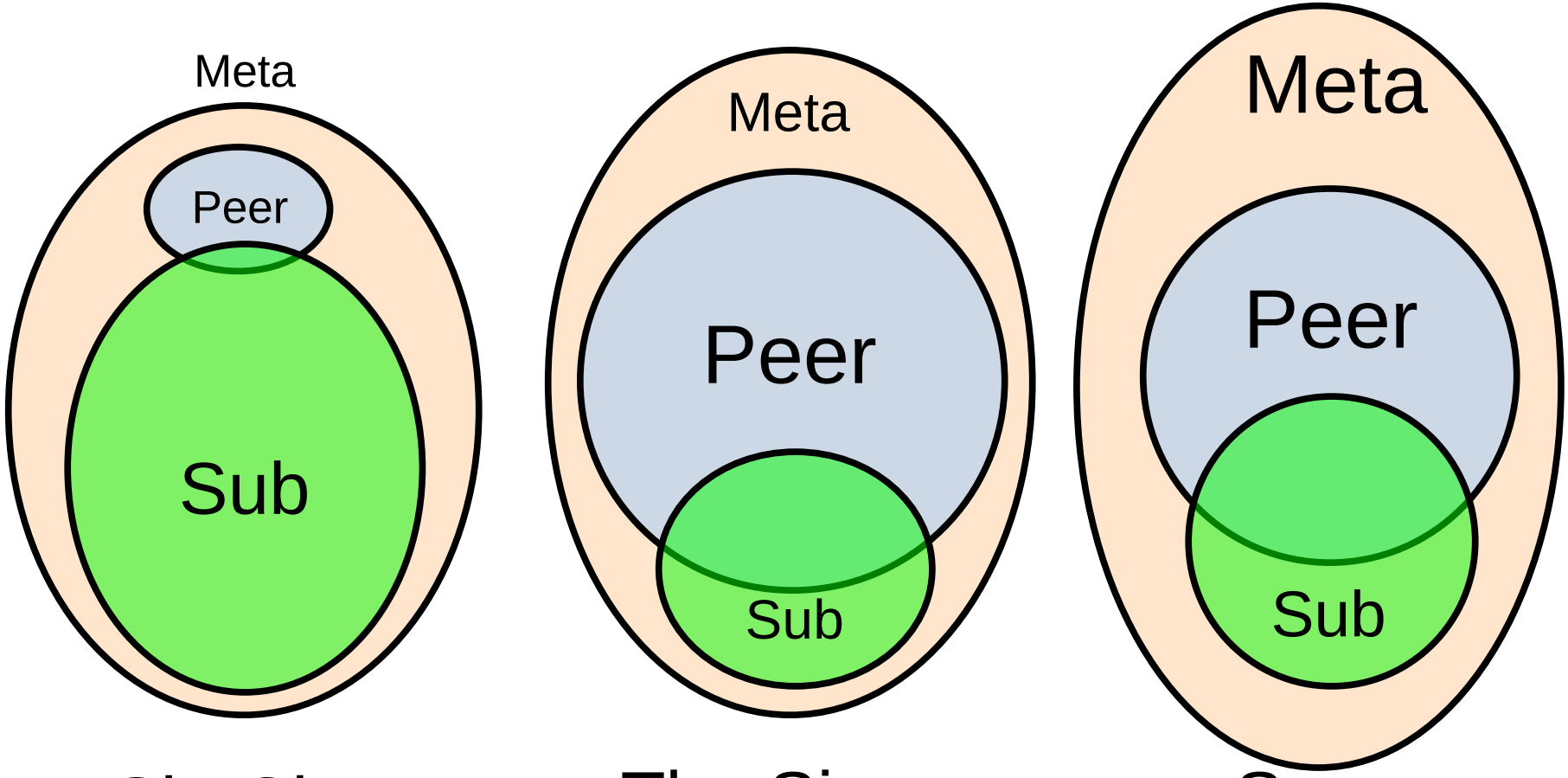


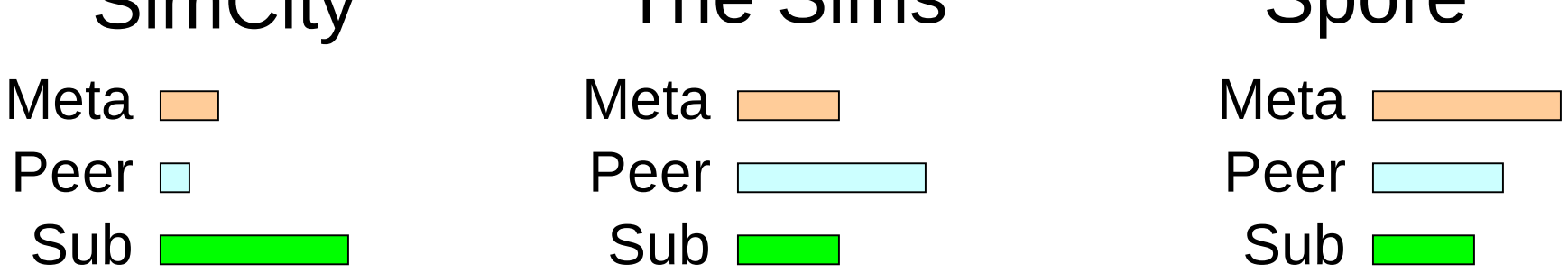
Experience
- Information Flow
- Pacing
- Simple Player Model

Agents
- Behavior
- Opponents/Avatars
- Complex Player Model

Simulation
- Physics
- Tactile
- Intuitive Player Model

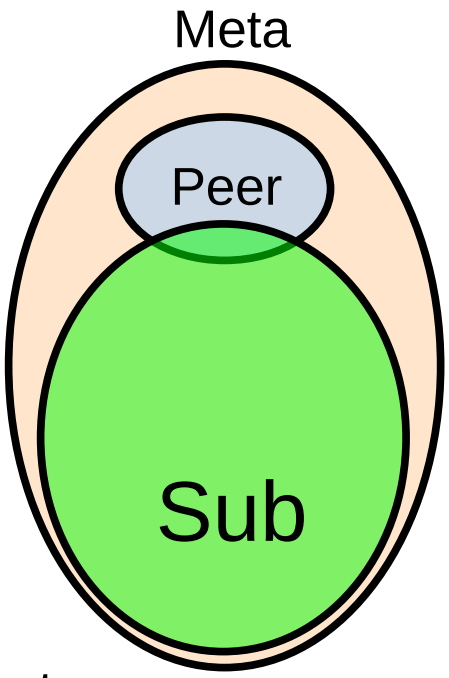
ゲーム毎に占める割合が違う





The SimCity Sub AI

SimCity シリーズのAIの作り方



Meta
Peer
Sub

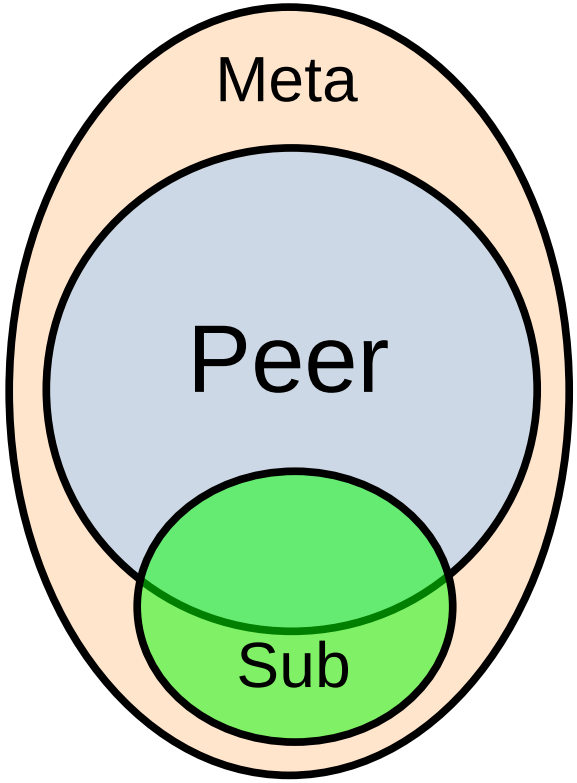
Crime = Pop. Density^2 - Land Value - Police Effect

Land Value = Distance[Zonetype] + Terrain + 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。**

The Sims Peer AI

The Sims シリーズのAIの作り方



Meta
Peer
Sub

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

Sims (not under direct player control) choose what to do by selecting, from all of the possible behaviors in all of the objects, the behavior that maximizes their current happiness.

人をダイナミクス（力学系、動的な数値の仕組み）として動かす。
世界を動かす PeerAI(=キャラクターAI) **を構築。**

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 (obiect 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

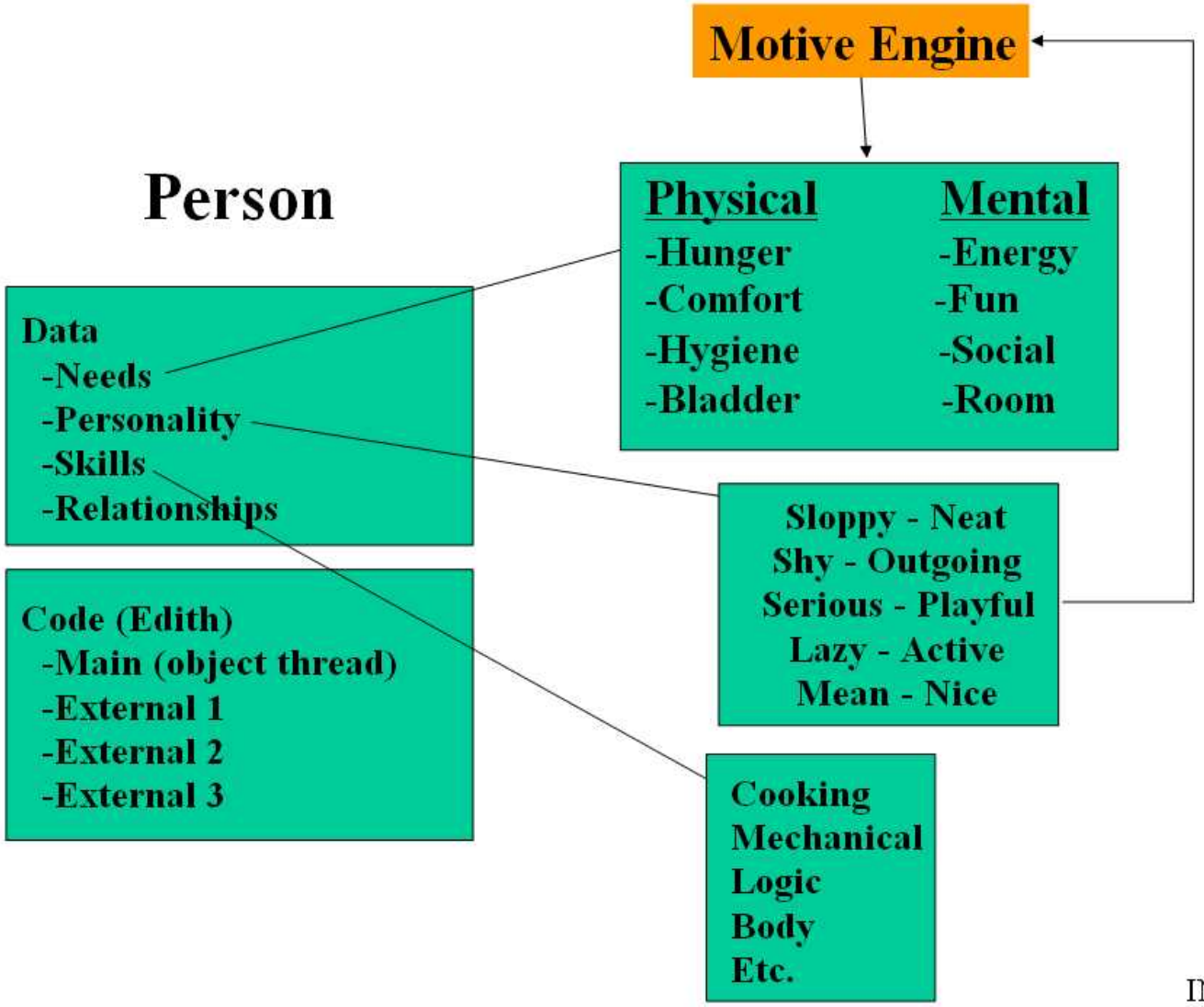
サウンド

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" (NorthWerstern**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



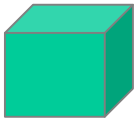
Ken Forbus, “Simulation and Modeling: Under the hood of The Sims” (NorthWerstern**大学、講義資料**)
http://www.cs.northwestern.edu/%7Eforbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/frame.htm

最適な行動を選択する

Hunger +20
Comfort -12
Hygiene -30
Bladder -75
Energy +80
Fun +40
Social +10
Room -60

Mood +18

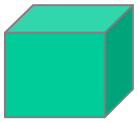
Toilet



Mood +26

-Urinate (+40 Bladder)
-Clean (+30 Room)
-Unclog (+40 Room)

Bathtub

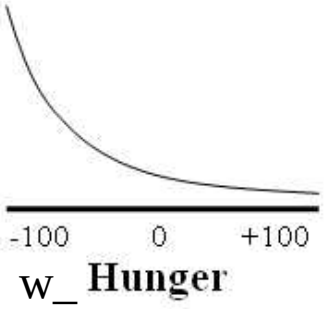


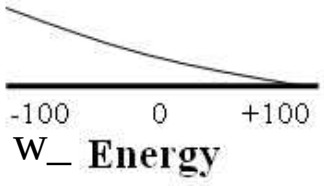
Mood +20

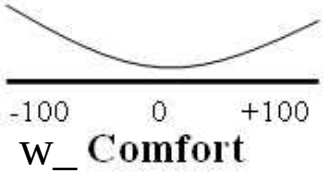
-Take Bath(+40 Hygiene)
(+30 Comfort)
-Clean (+20 Room)

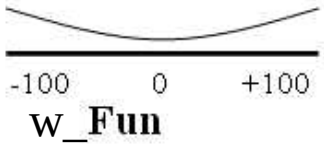
[原則] 周囲の対象に対する、あらゆる可能な行動から、Happiness（ここではMood）係数を最大化する行動を選択する。

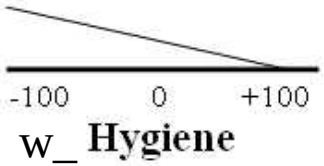
Happiness 係数の計算のためのウエイト

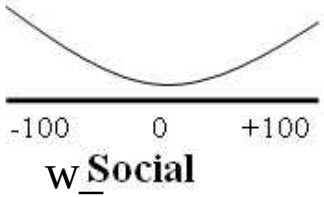


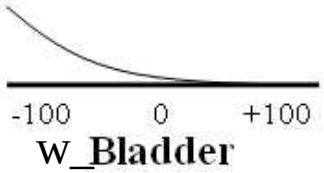


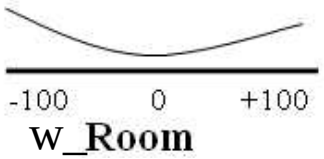








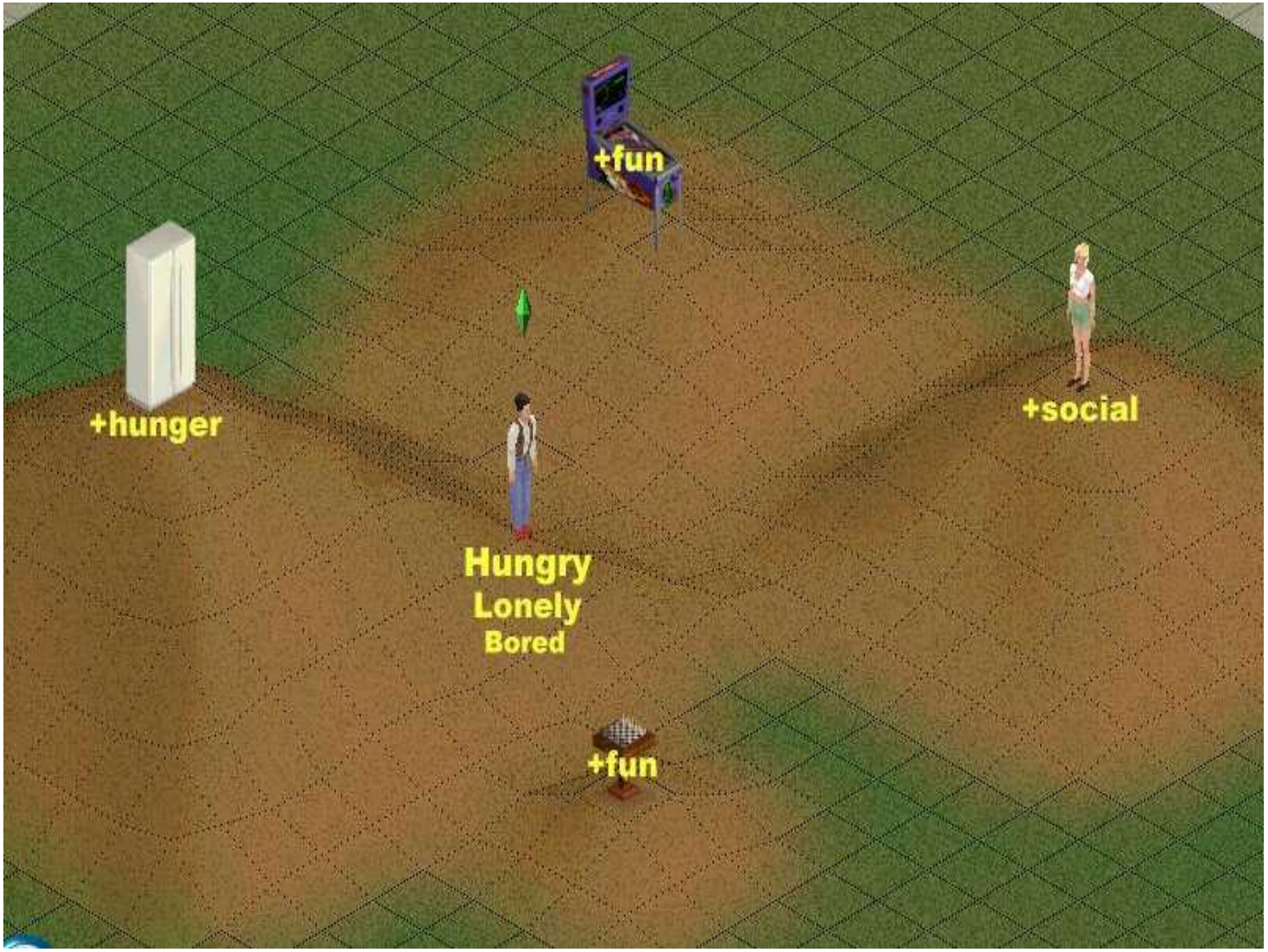




Mood (-100..+100)

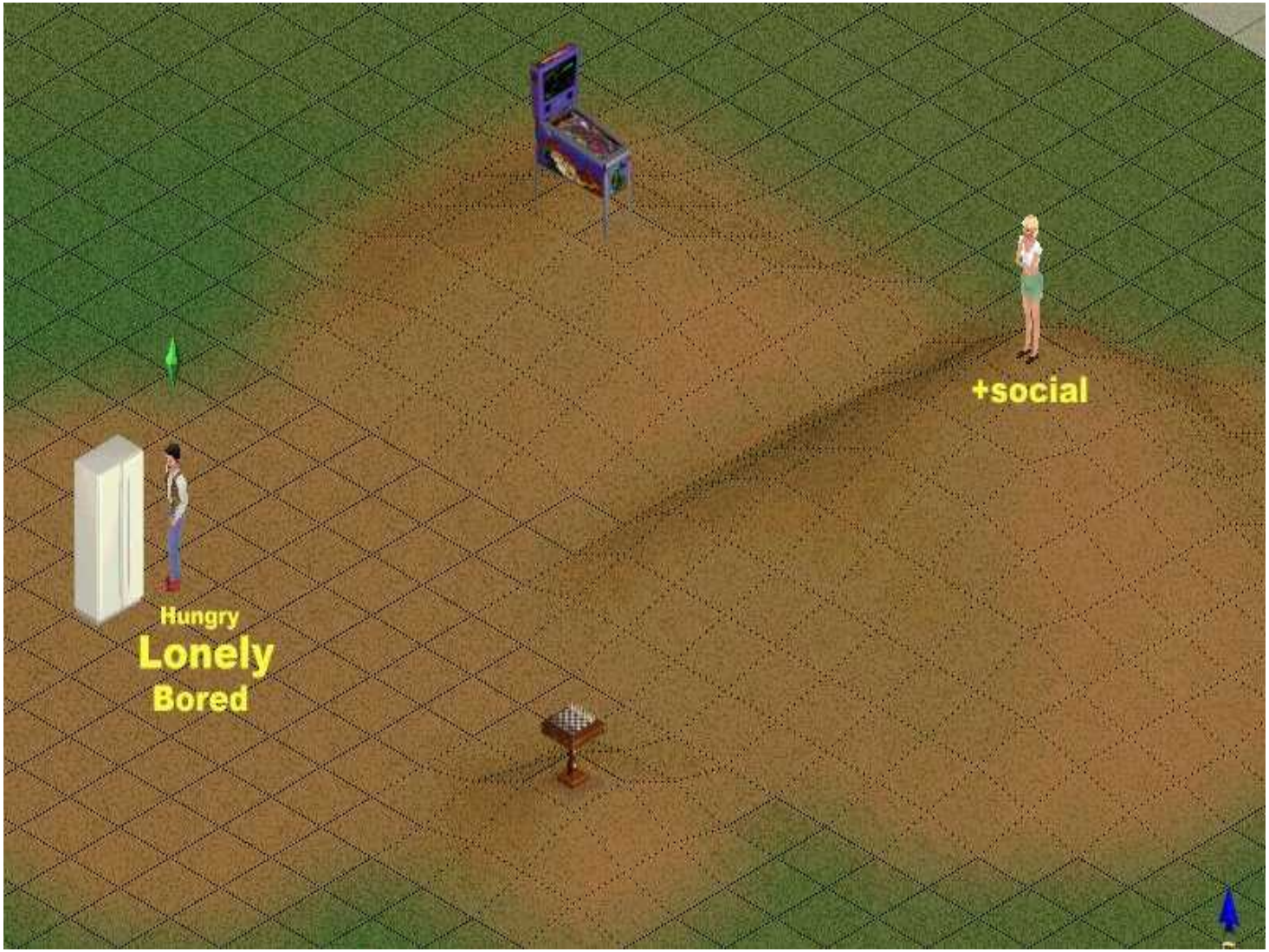
Hapiness = W_Hunger * Hunger + W_Engergy * Energy + ...

Happiness を最大化



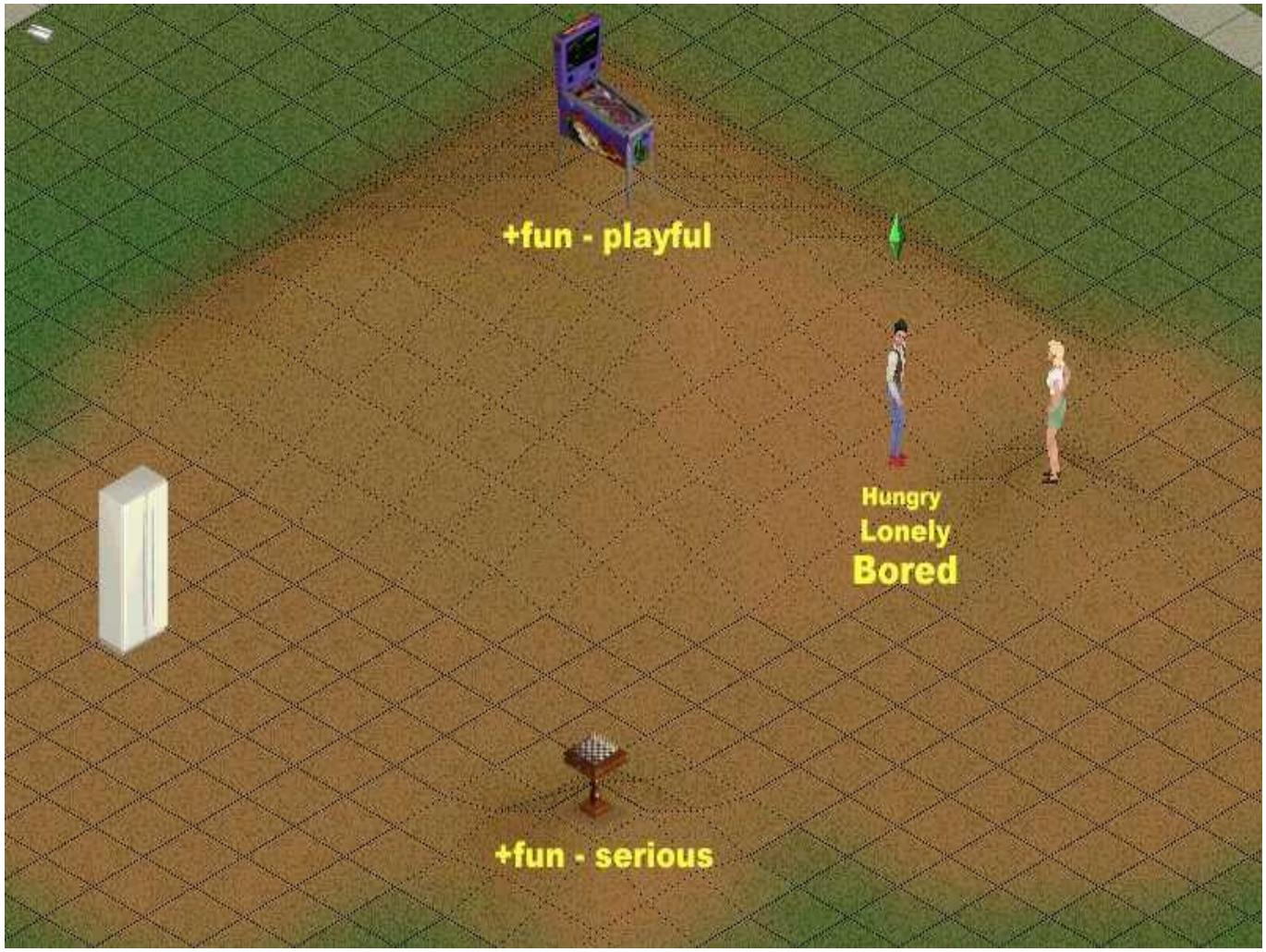
冷蔵庫が最も総合的にHapinessを上昇させるから

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。

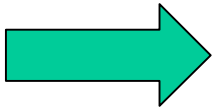
[余談] Halo3

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

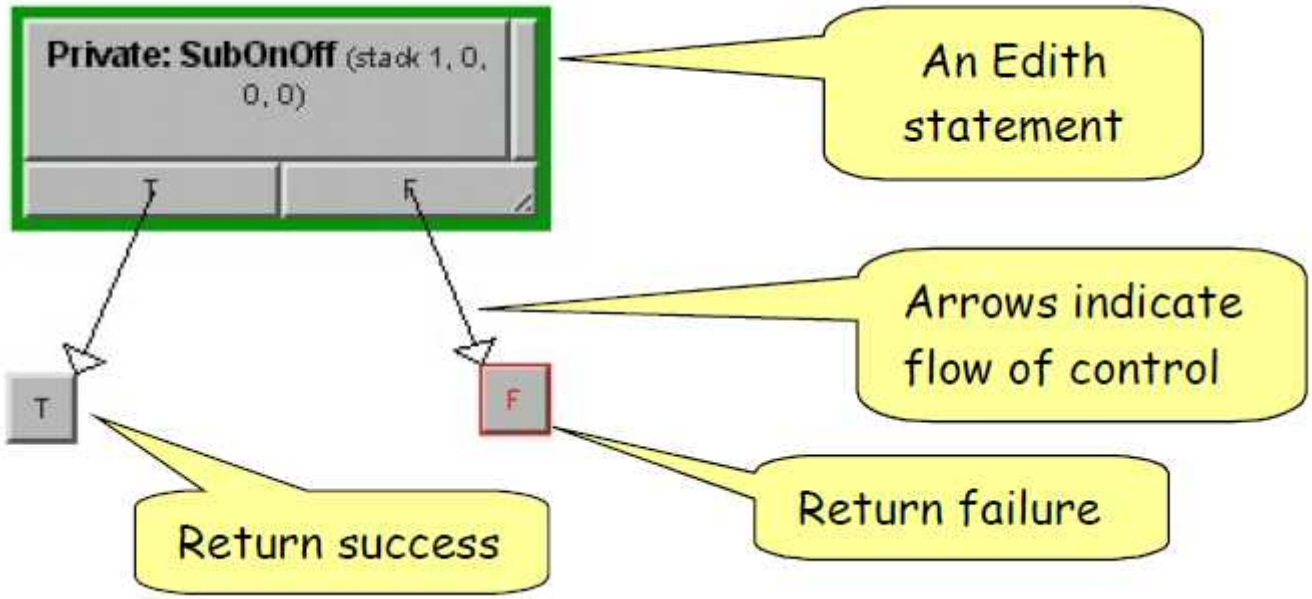
これだけだと、
原始的で単発の行動しかできないけれど…

実際は、
一連の行動のシークエンスを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

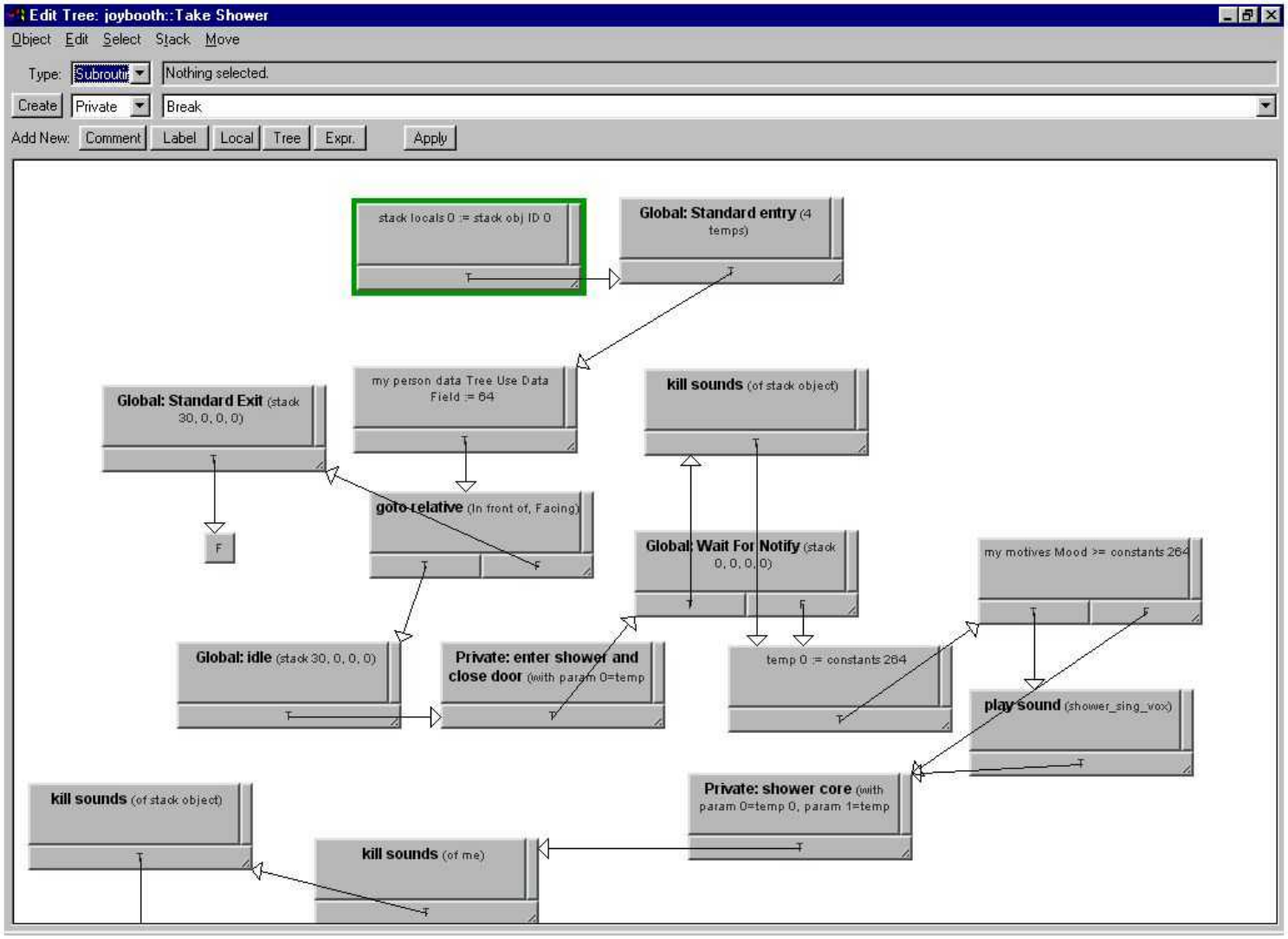


Kenneth D. Forbus “Some notes on programming objects in. The Sims”

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

Edith

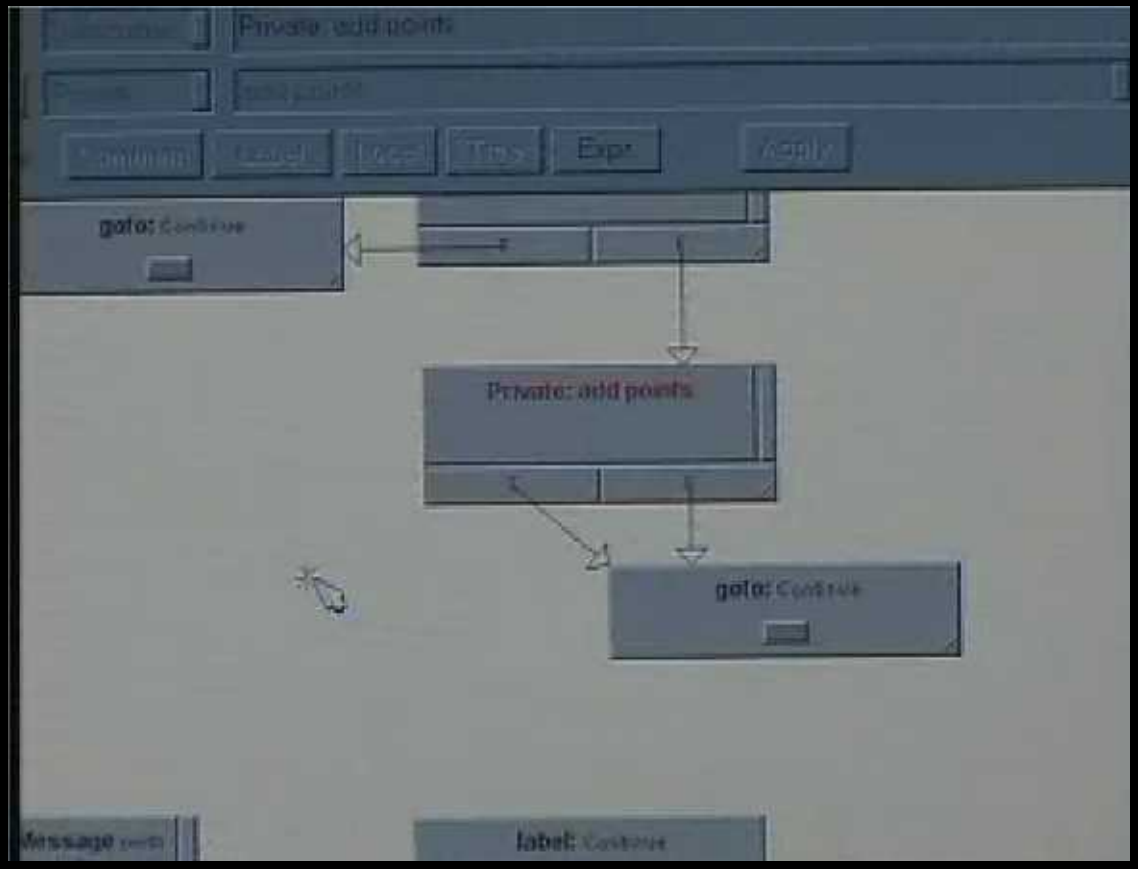
プログラミング・オブジェクトを繋げて行くビジュアル・プログラミング環境



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

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

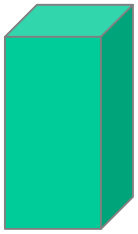
Edith Demo



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

実例①　「調理して食べる」

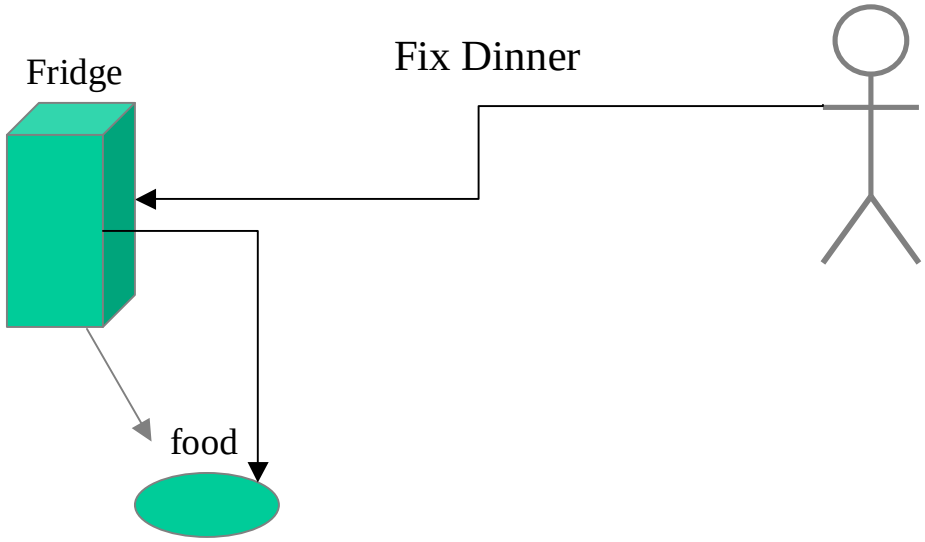
Fridge



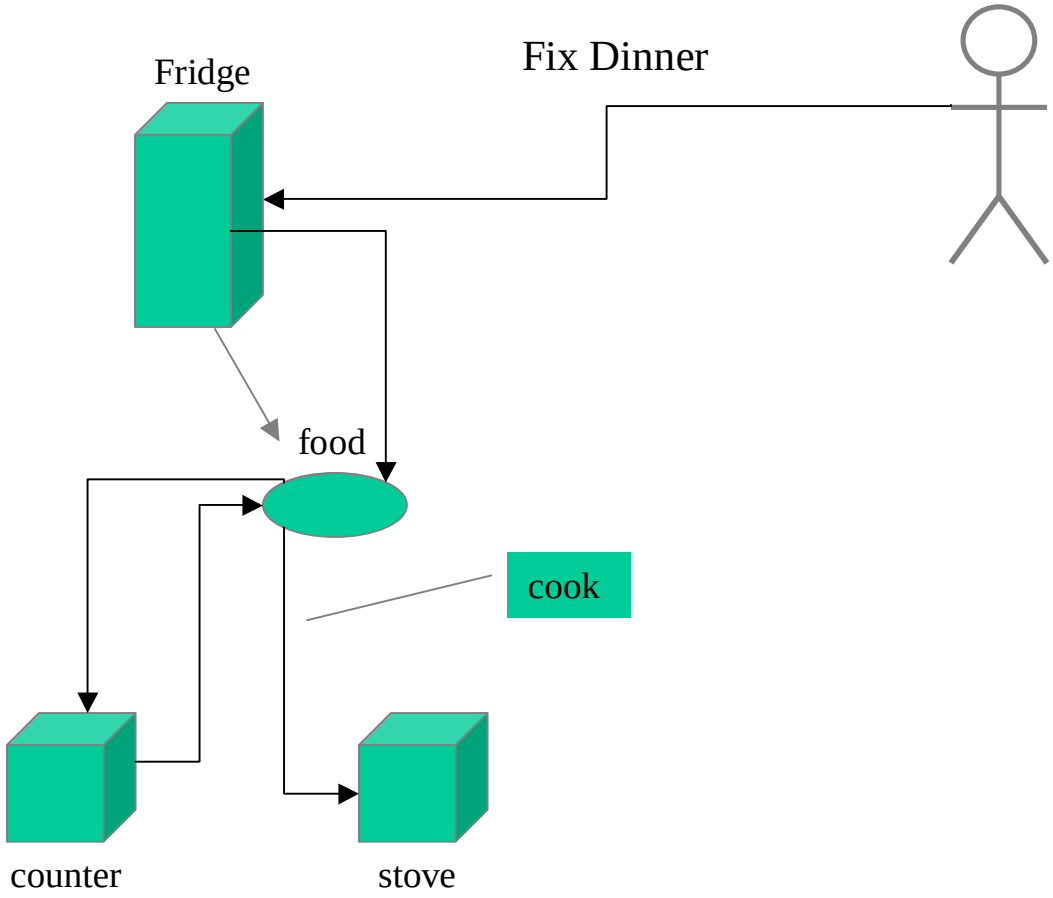
Hunger +30

Hungry

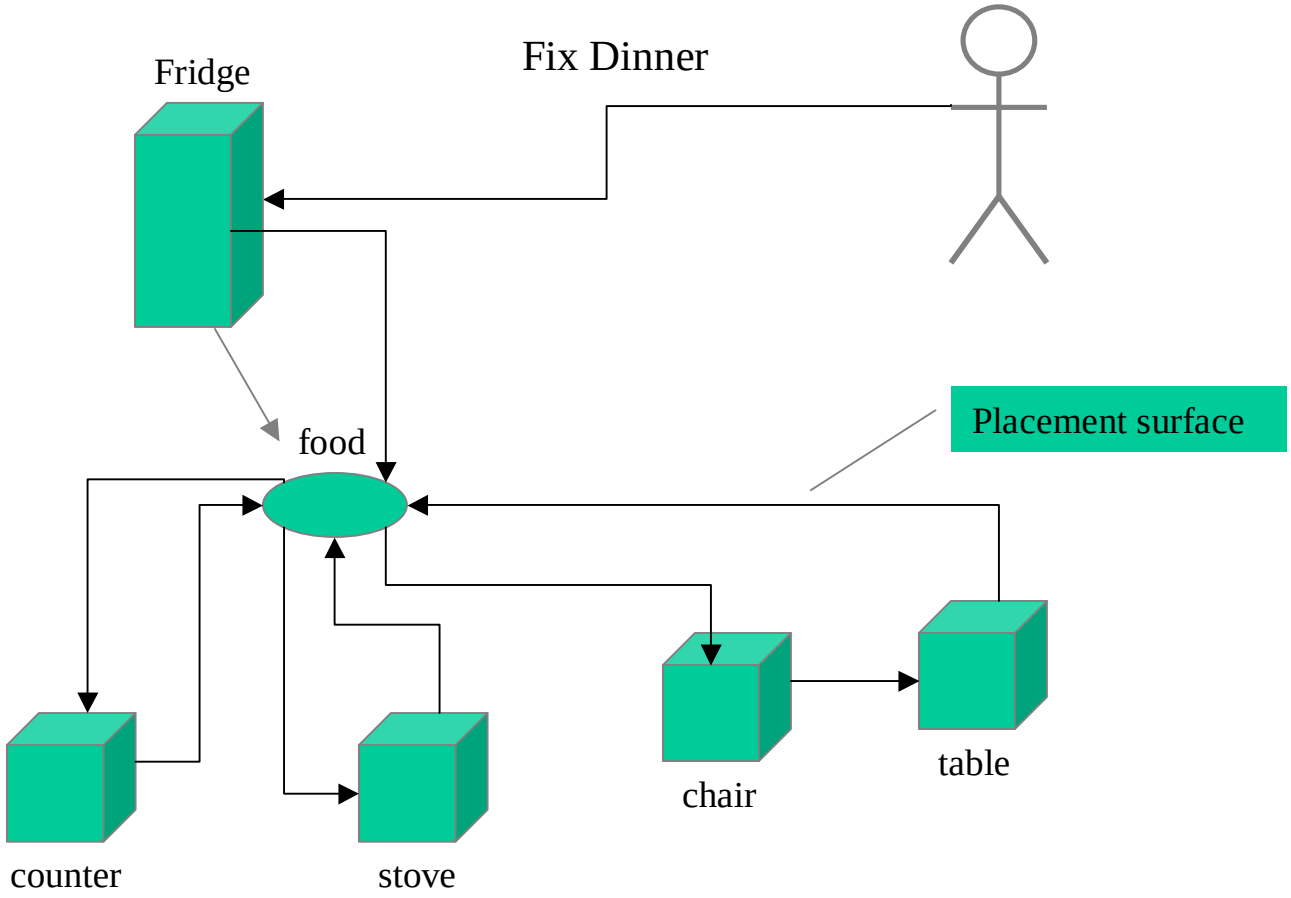
実例①　「調理して食べる」



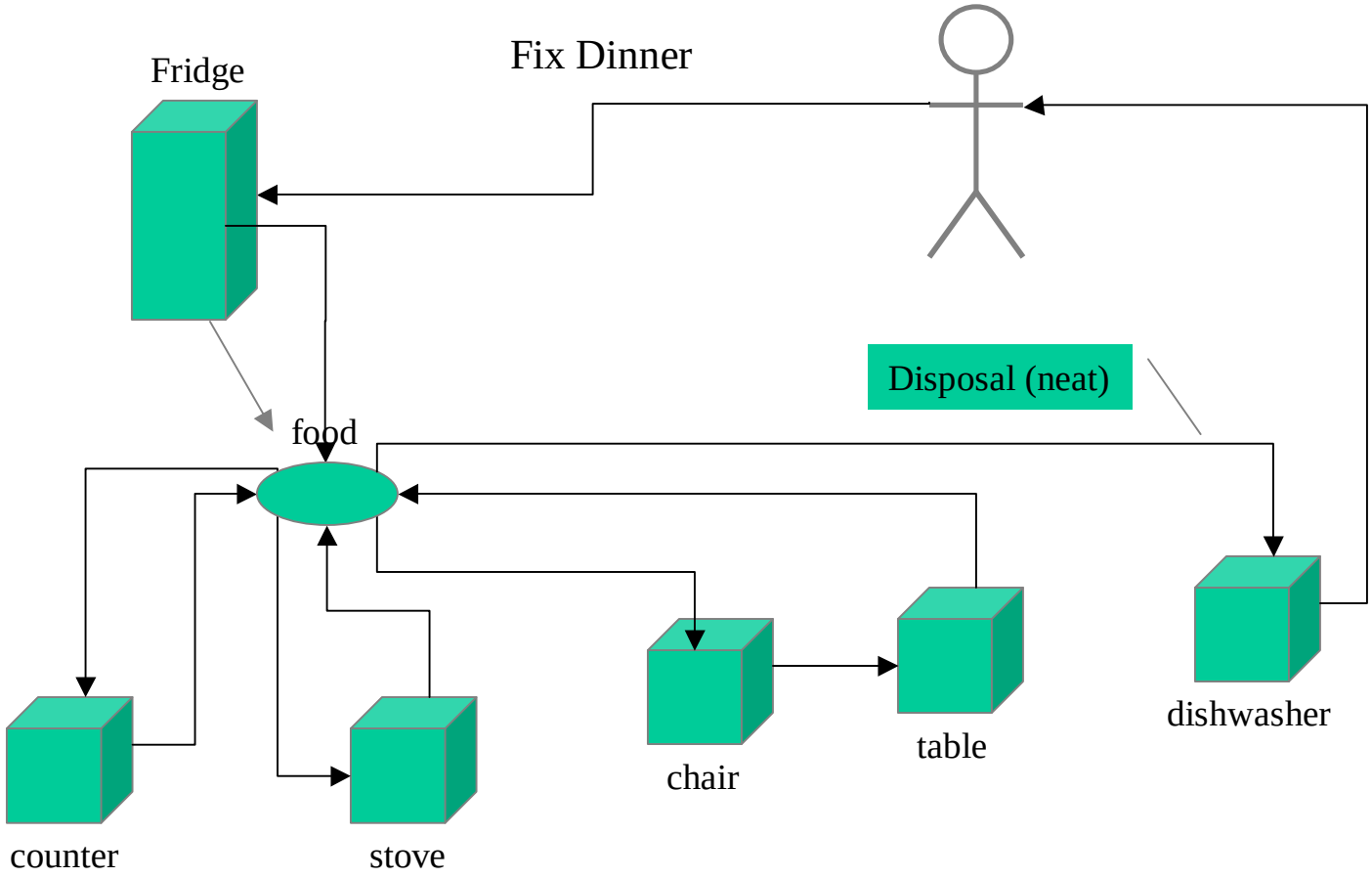
実例①　「調理して食べる」



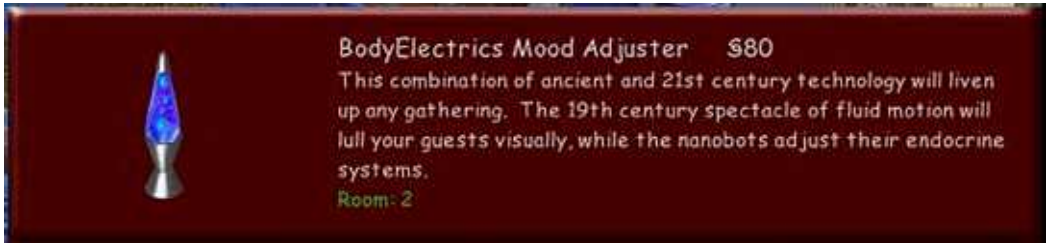
実例①　「調理して食べる」

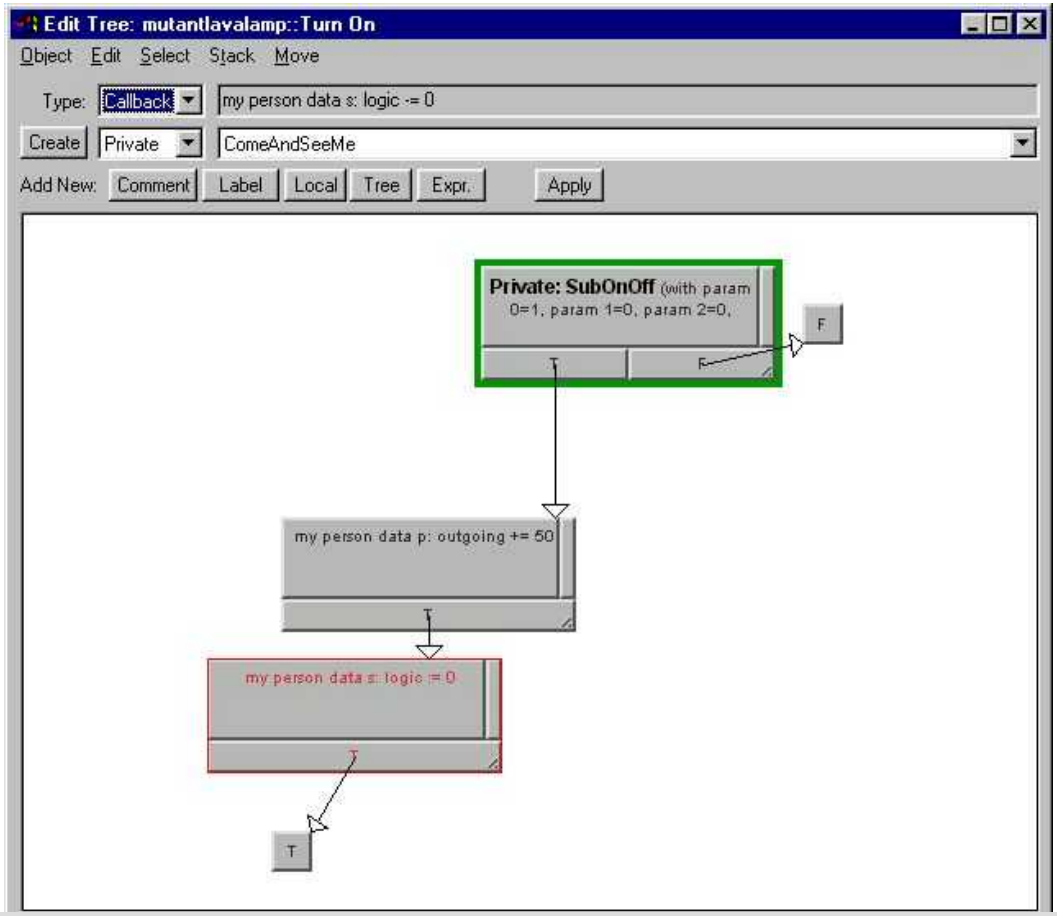

Fridge
Fix Dinner
food
Placement surface
counter
stove
chair
table

実例①　「調理して食べる」


Fix Dinner
Fridge
Disposal (neat)
food
counter
stove
chair
table
dishwasher

Edith for Mood Adjuster (パーティ盛り上げグッズ)

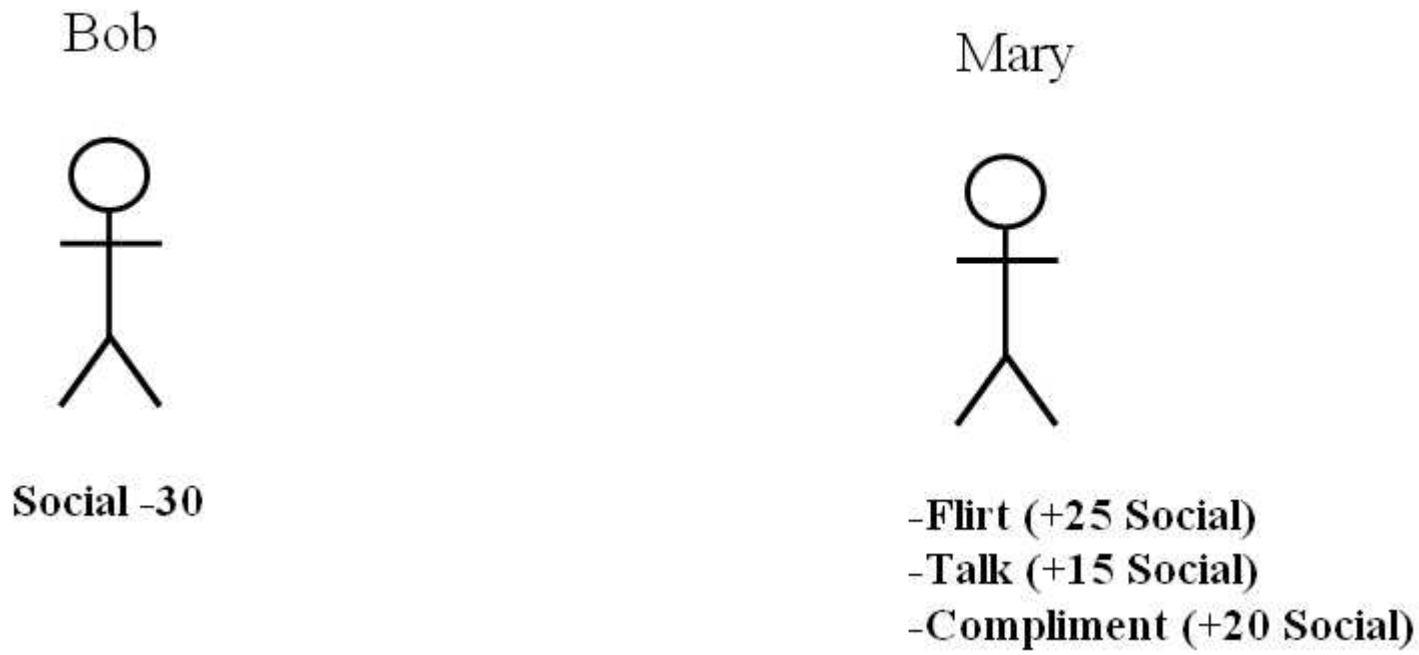




Kenneth D. Forbus “Some notes on programming objects in. The Sims”

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

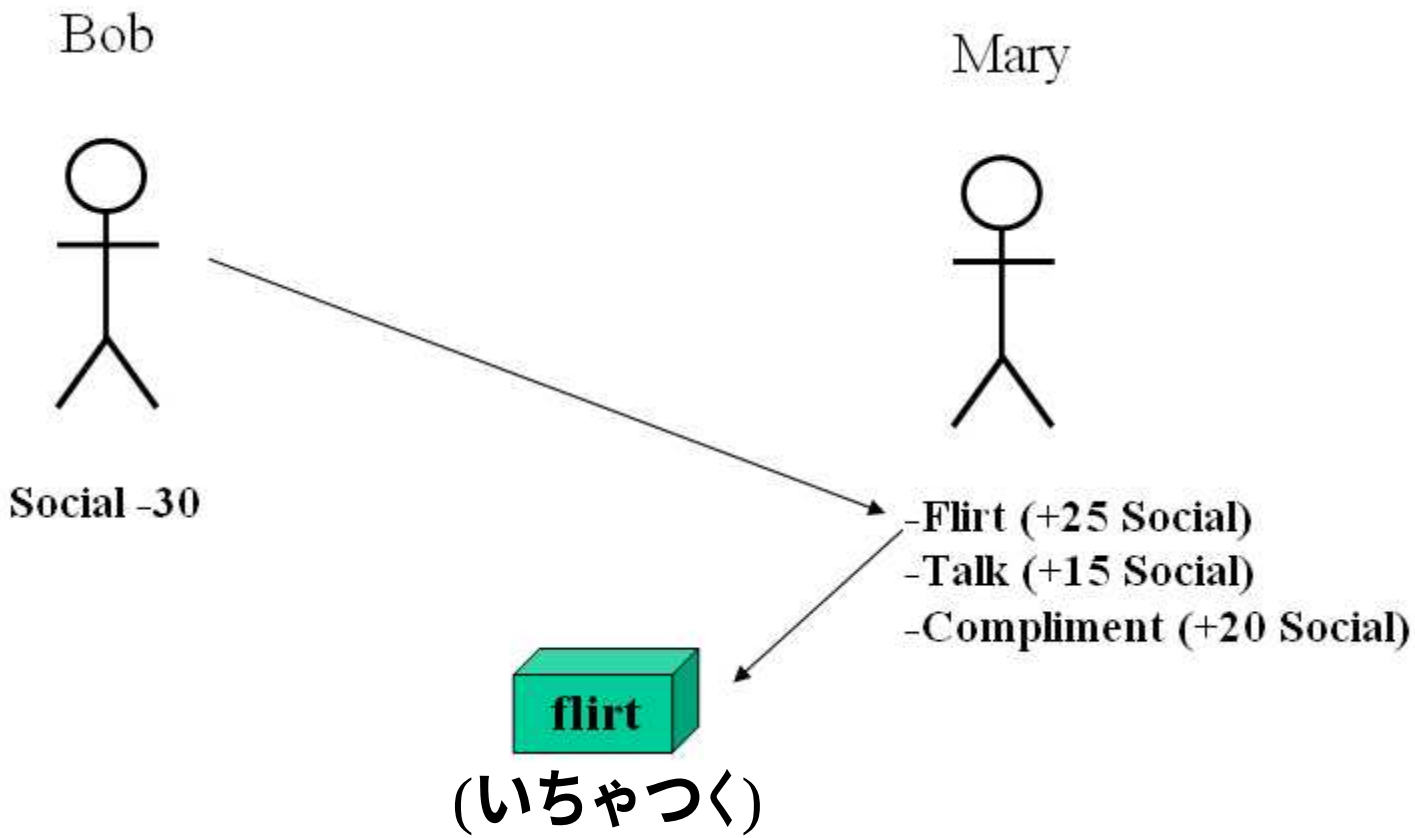
人同士のコミュニケーション



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

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

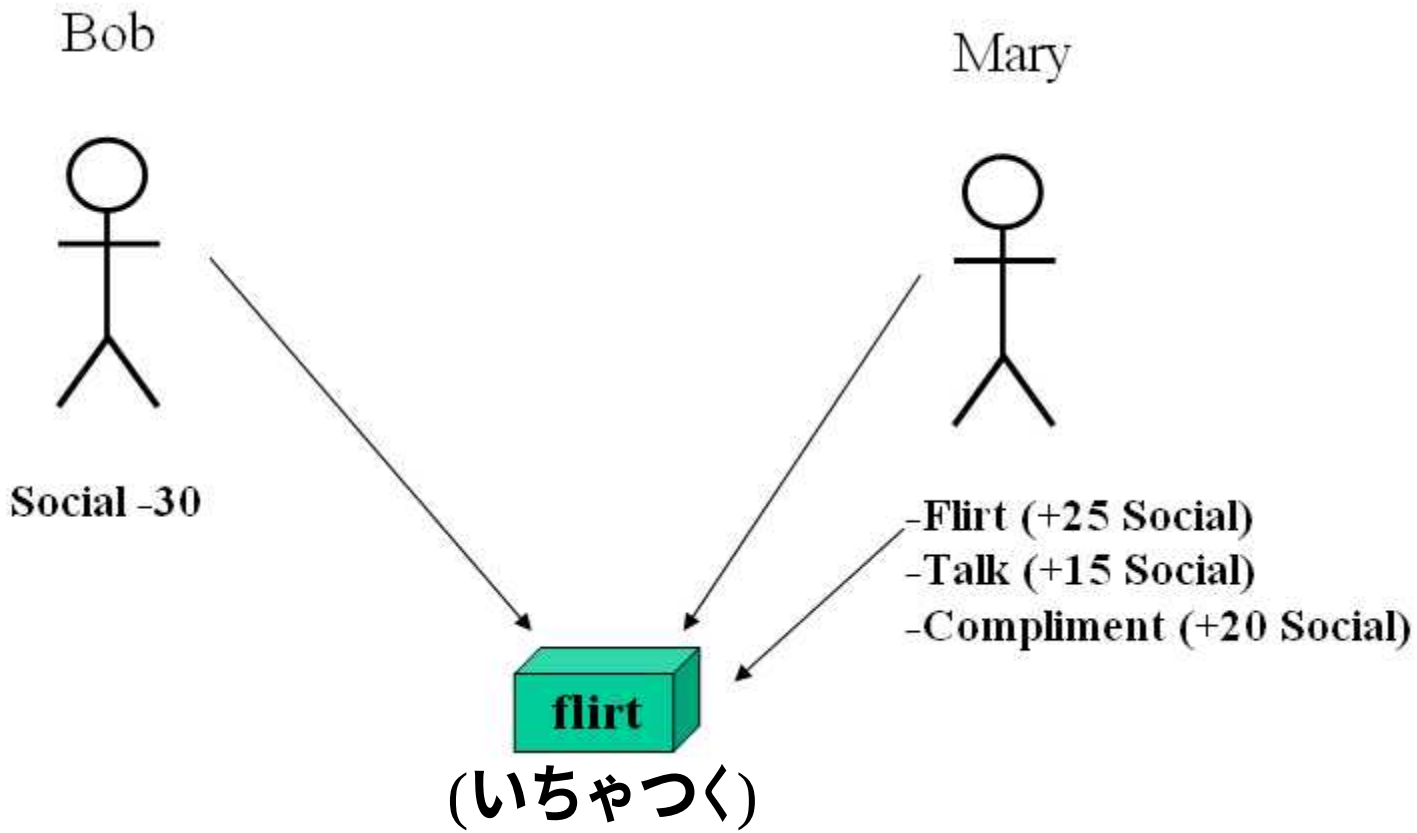
人同士のコミュニケーション



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

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

人同士のコミュニケーション



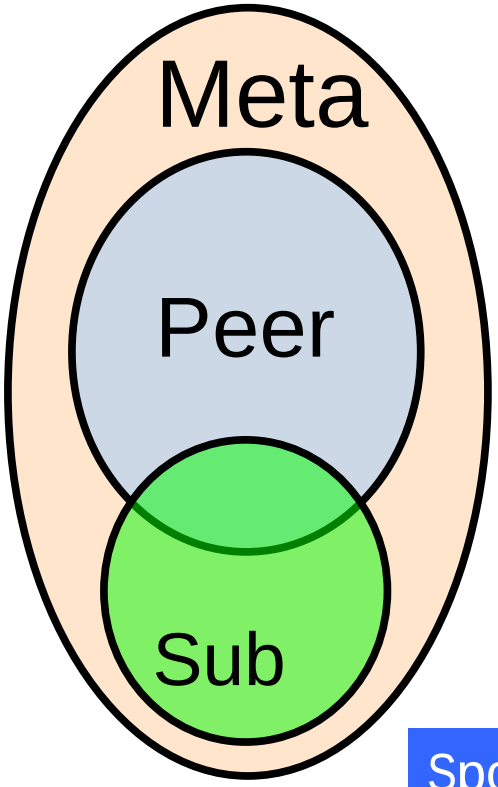
Kenneth D. Forbus “Some notes on programming objects in. The Sims”

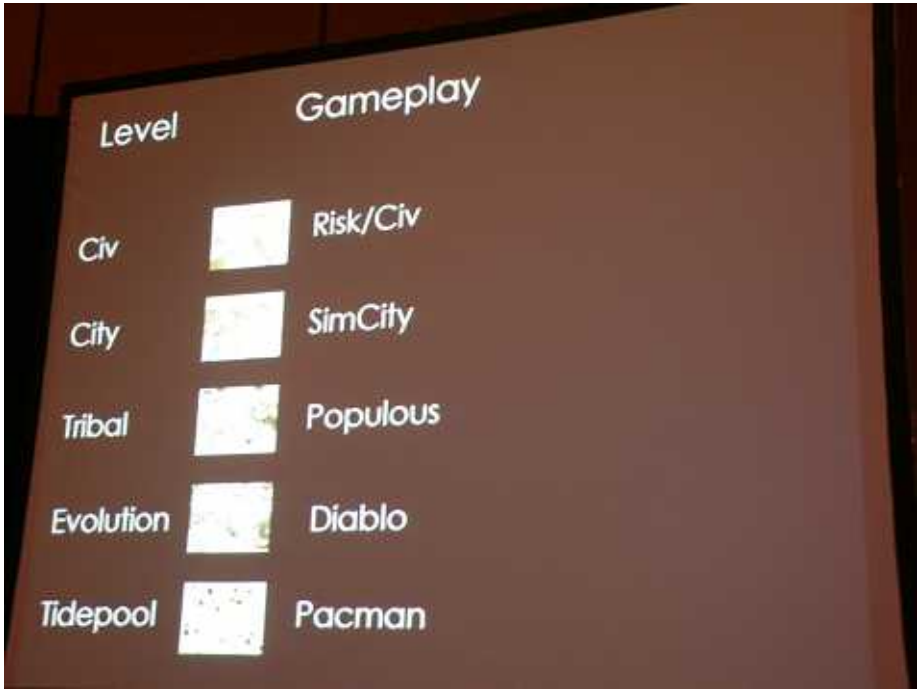
http://www.qrg.northwestern.edu/papers/Files/Programming_Objects_in_The_Sims.pdf

Spore AI

- Spore Peer AI
- Spore Meta AI

SporeのAIの作り方





Will Wright, “The Future of Contents ” GDC 2005

Meta
Peer
Sub

Spore は階層ごとに使うAIが違う。
都市のシミュレーションは、SimCityのSub AIシステム、
キャラクターや種族を動かすAIはThe Sims のような Peer AI
全体の情報をマネージメントするにはプロシージャルなMeta AI

ゲーム全体の情報を統御する機能をメタAIとして、
プロシージャルを使って情報を知的にマネージメントしている。

7層に分かれたステージと作成コンテンツ

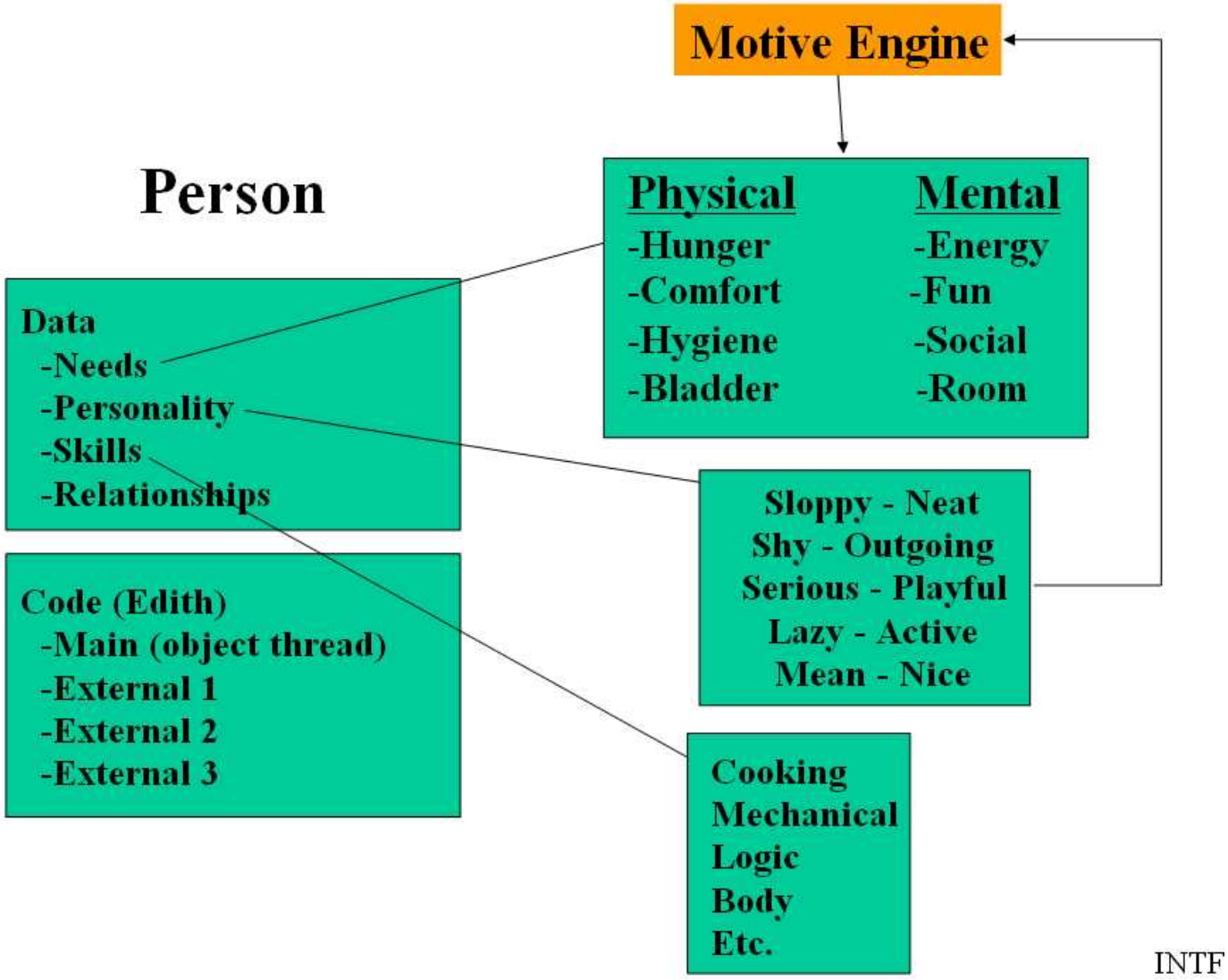
http://www.gamespot.com/pc/strategy/spore/news.html?sid=6165667

Spore Peer AI

(予想)

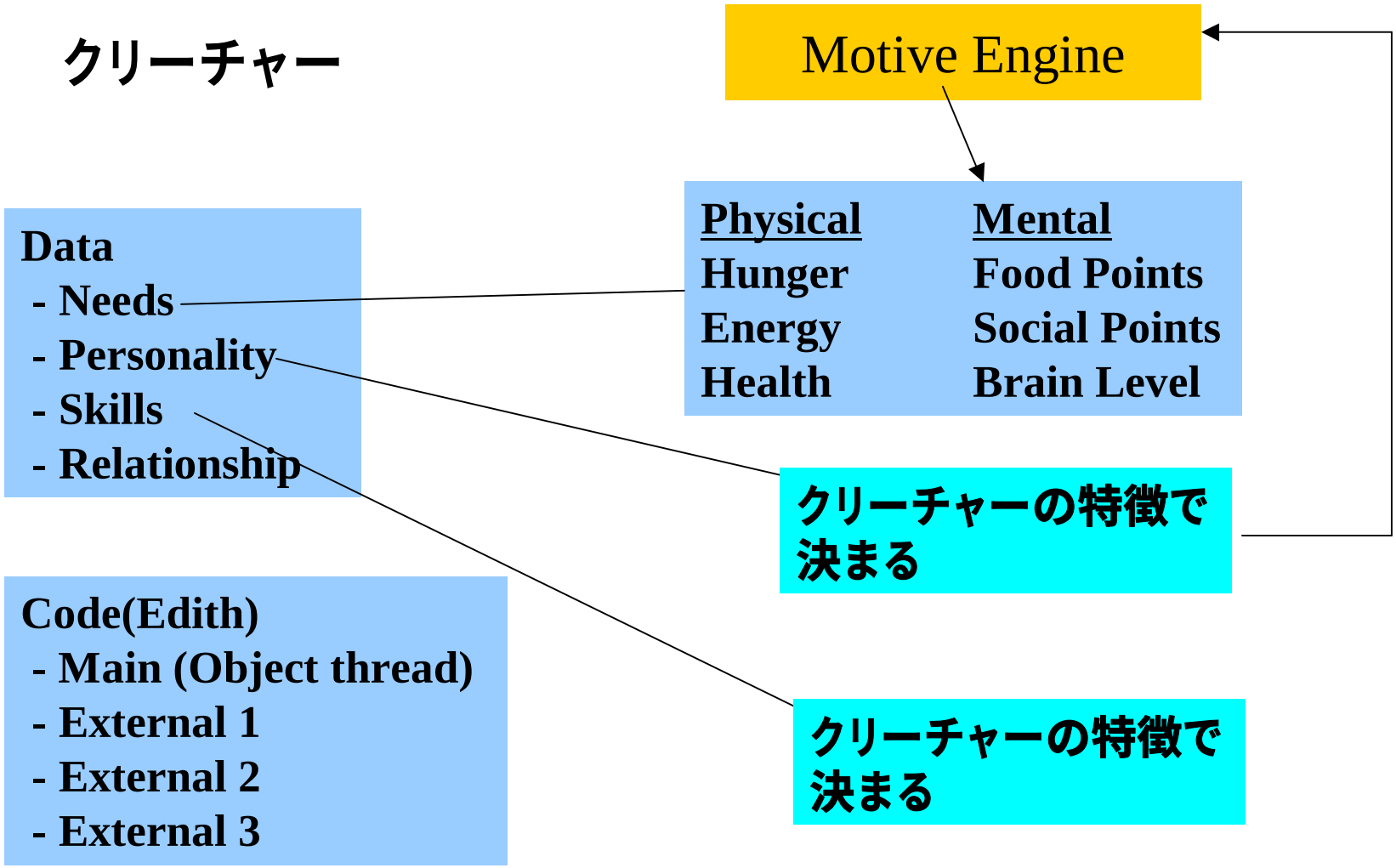
Spore **のキャラクター制御**

ＮＰＣに仕込むデータ構造



Ken Forbus, “Simulation and Modeling: Under the hood of The Sims” (NorthWerstern**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）

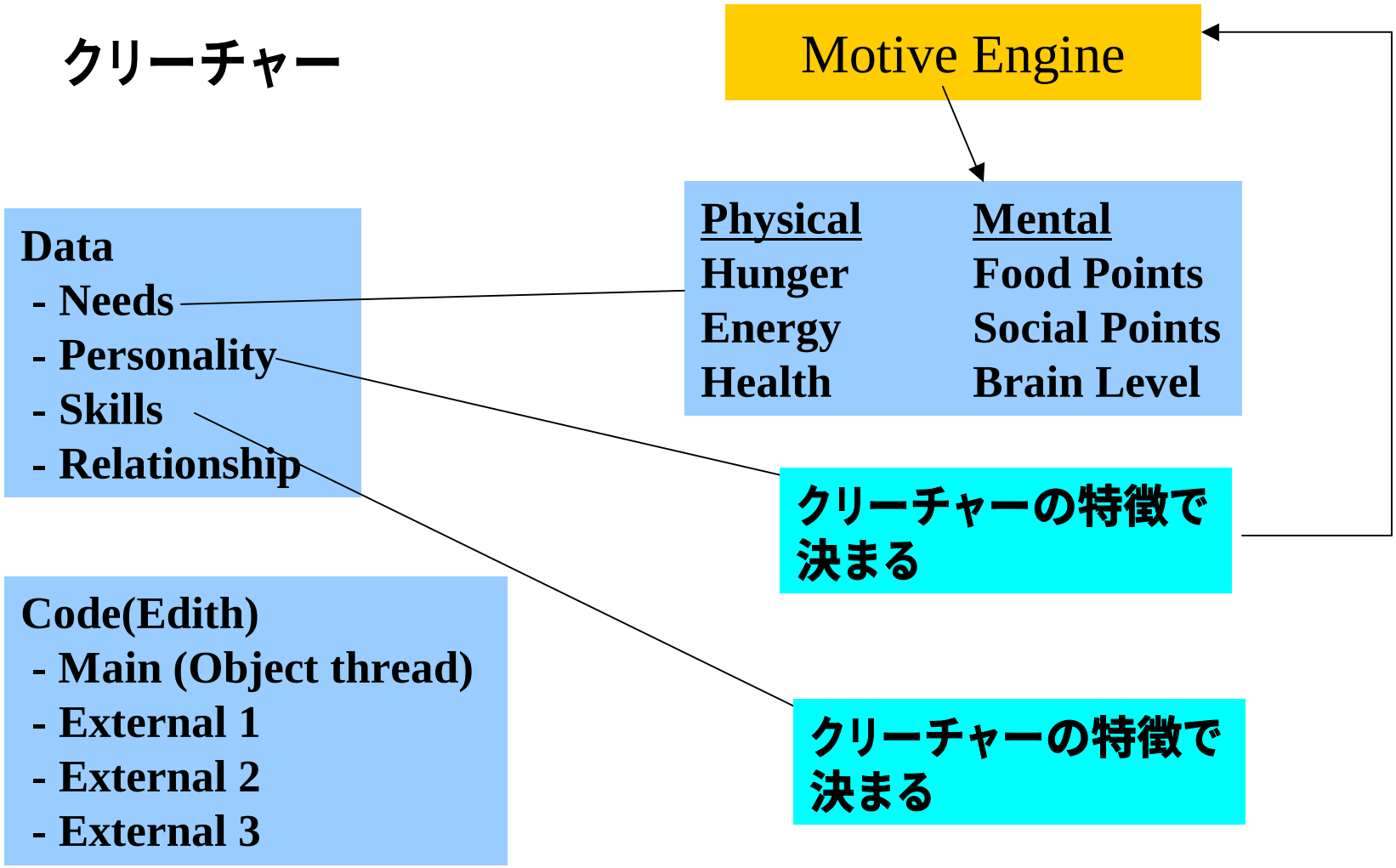



SPORE(TM)クリーチャークリエイター トライアル版
ボタンでひとつ前の作業を取り消すことができます

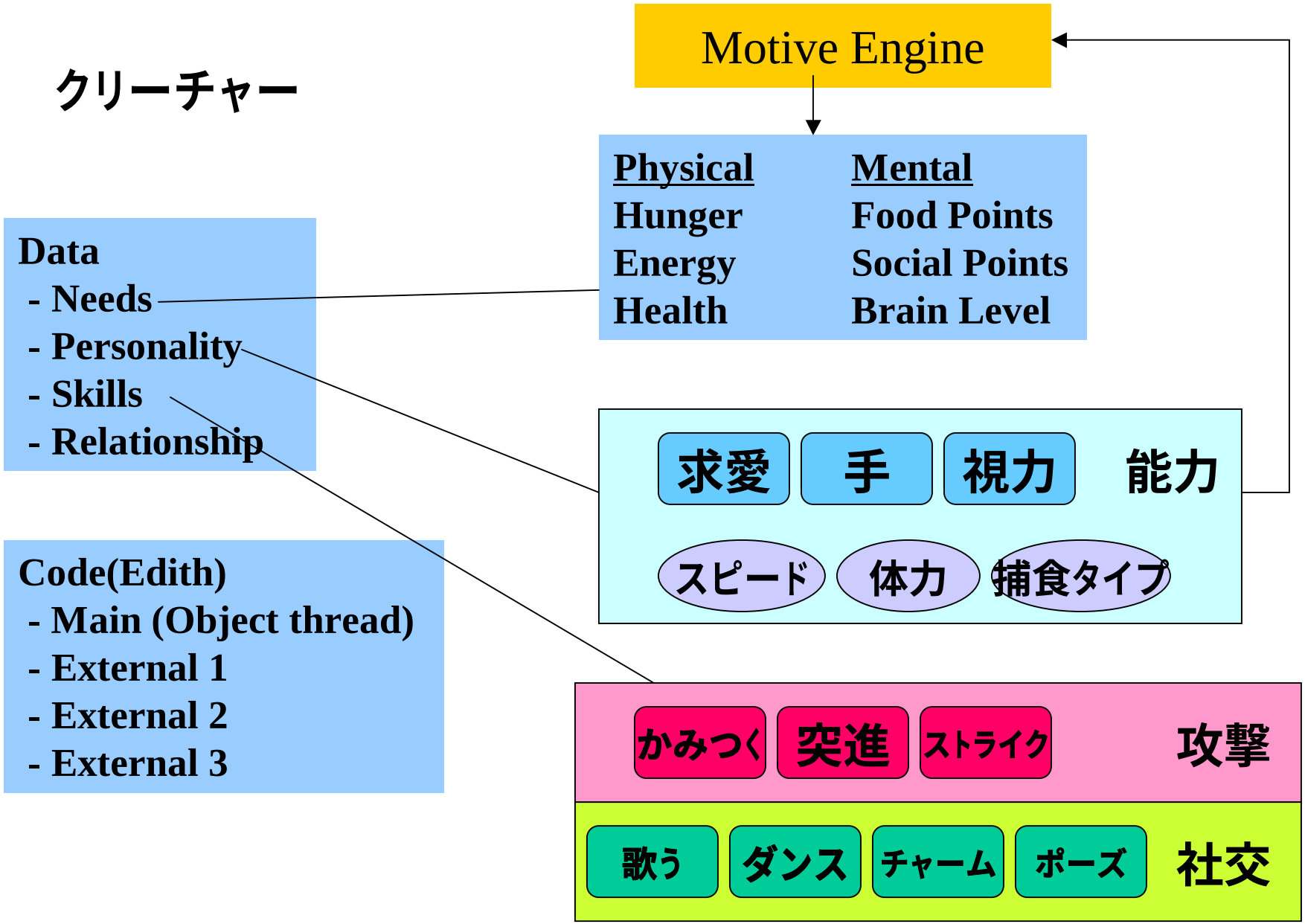
Sporeクリーチャークリエーターから
自動決定されるパラメータ

- 能力
  - 求愛
  - 手
  - 視力
- 攻撃
  - かみつく
  - 突進
  - ストライク
- 社交
  - 歌う
  - ダンス
  - チャーム
  - ポーズ
- スピード
- 体力
- 捕食タイプ

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, 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

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

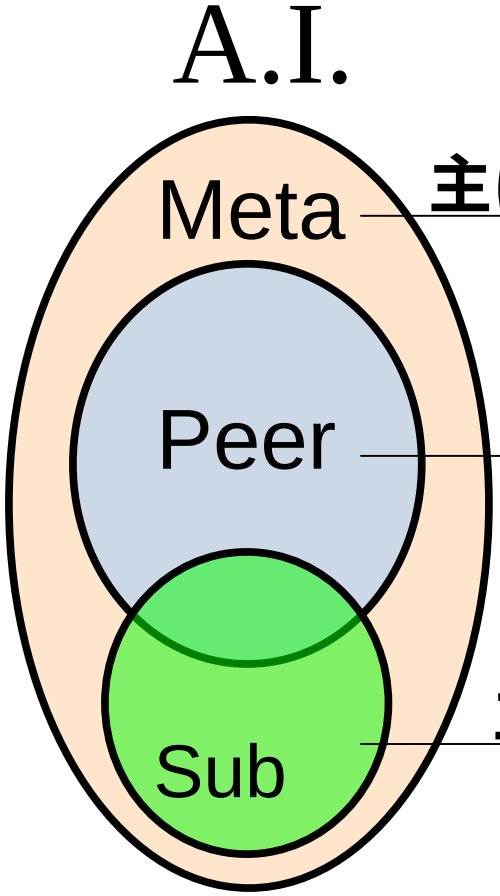
http://www.4gamer.net/games/020/G002026/20080610033/

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



主に Spore

主に Sims

主にSimCity

ユーザーとゲームの間の情報の流れをマネージメントする（プロシージャル）

キャラクターとそれを取り巻く局所（peer）世界の関係からキャラクターを動作させる。（キャラクターAI）

世界の自律的な法則（物理・経済の法則など）

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。

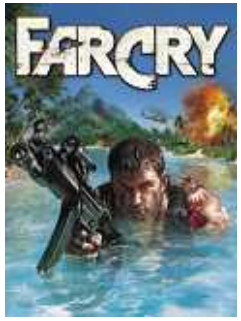
しかし、あくまで亜流であり、
ハードウエアの制約からゲームの
限られた部分に使われて来ただけだった。
（ダンジョンとかメロディー生成とか…）

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

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

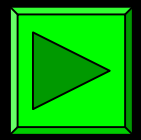
CryEngine デモ

tiago_gpuGems3_1280_mpeg

Procedural Vegetation Animation in Crysis

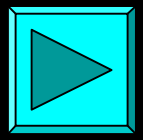
Tiago Sousa, "Chapter 16: Vegetation Procedural Animation and Shading in Crysis", GPU Gems 3, 2007

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、

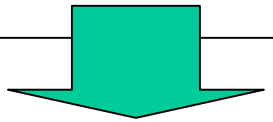
Spore(Maxis,EA)におけるプロシージャル技術は、
ウィル・ライト(Maxis)氏が、
個人的な思想と熱狂のもとに
彼の作家性を起点として、
起こしたものであった。

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

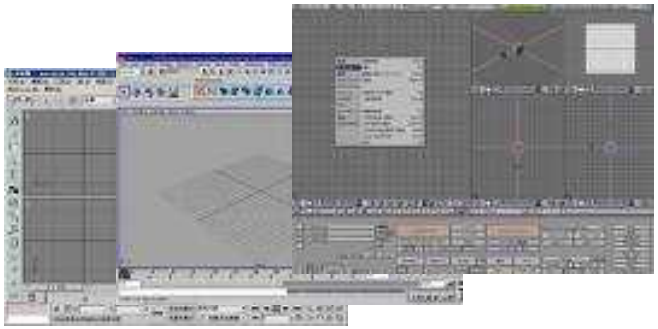
(注) プロシージャル･コンテンツ･ジェネレーションは頭文字を取って、
PCG と略される場合があります。

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

①アプローチ1

人の手によって
モデリングを行う

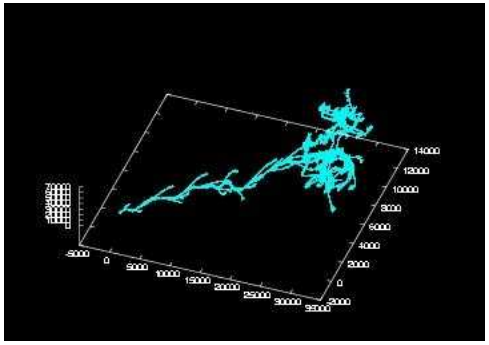


②アプローチ2

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

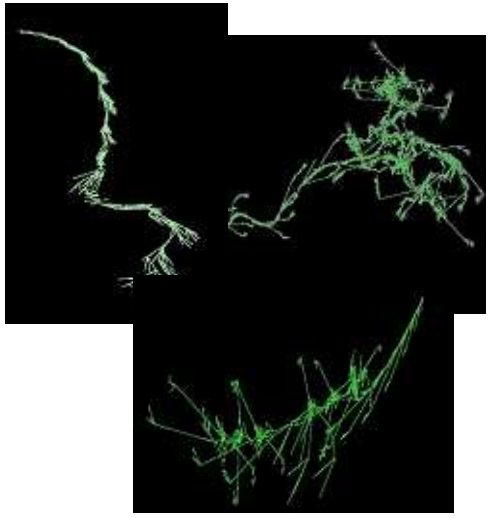
プロシージャル



```
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



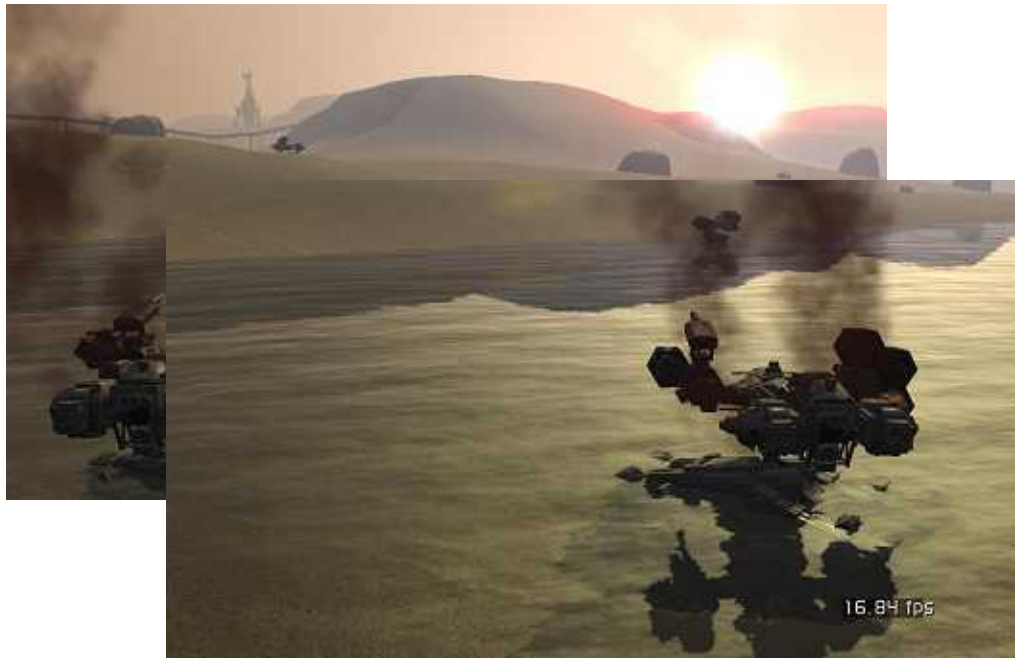
2003, Youichiro Miyake

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.tekepon.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)

http://212.202.219.162/home

- 64kb**から映像を生成する**

http://www.farb-rausch.com

デモ

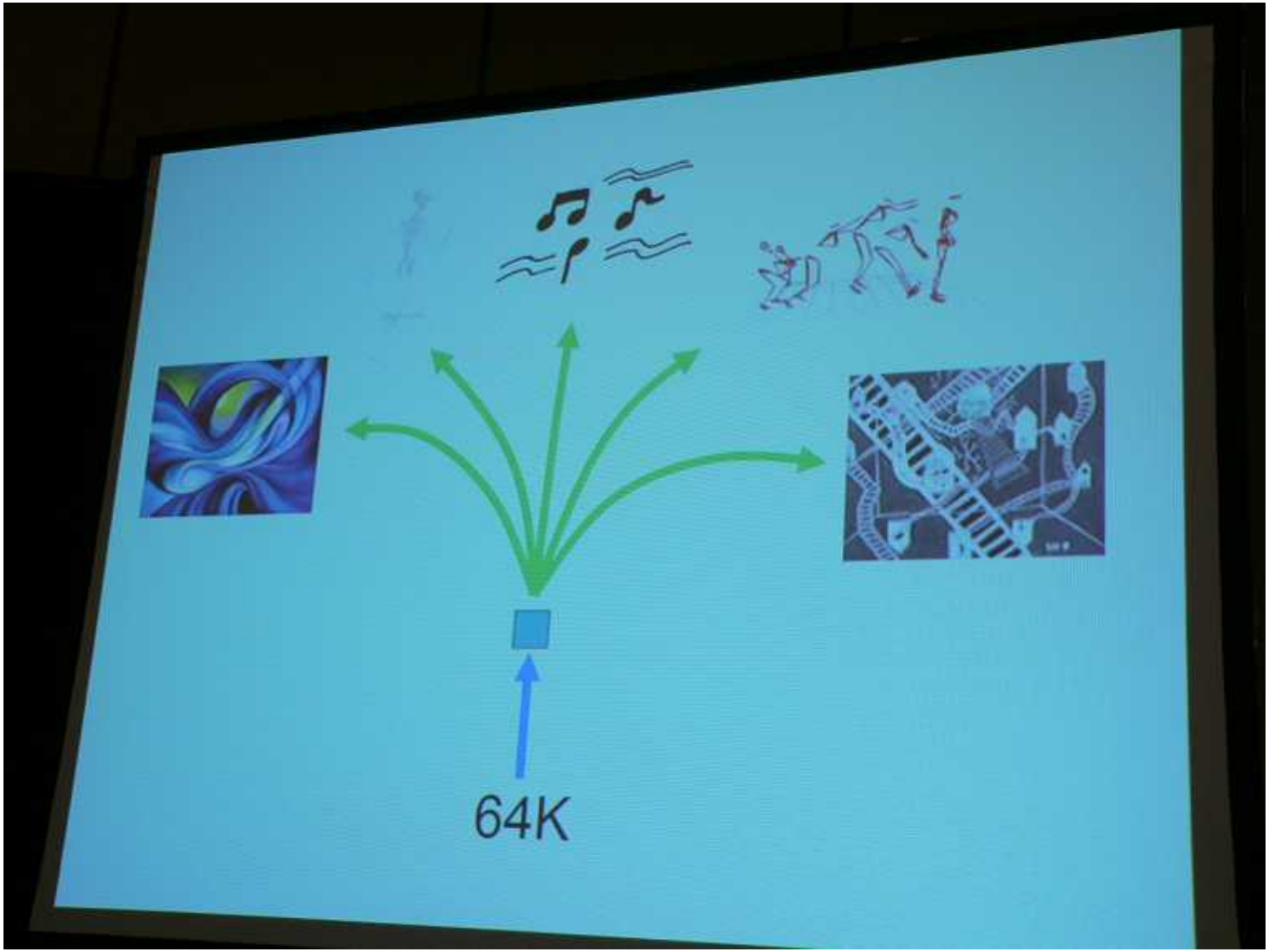
http://www.farb-rausch.com

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

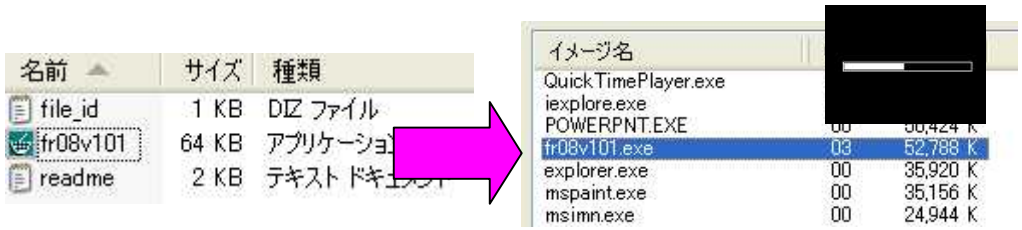
デモシーナー が教えてくれること

固定データ

生成されるデータ

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）



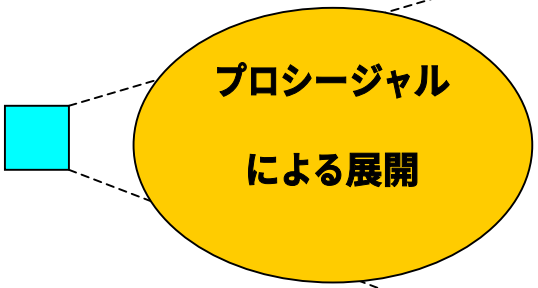
(1)「コンテンツ＝データ」ではない。　それは生成できるものだ。
(2) 大きな（巨艦大砲的な）ゲームもプロシージャルに作成すれば
手間もコストもデータ量も抑えて作ることができる

新しいゲーム製作のコンセプト

デモシーナーからSporeへ

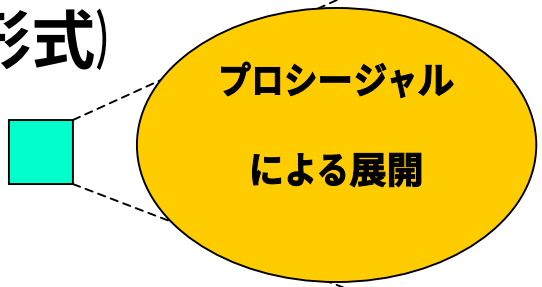
デモシーナー

６４ｋ



Spore

１００ｋ　(png形式)



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

Rogue Clone
ローグライクゲーム

Rogue

1980

NetHack

1987

不思議のダンジョン

チュンソフト

1993～

Hellgate：London

flagshipstudio

2006

ティル・ナ・ノーグ

システムソフト

1987～

Diablo

Blizzard

1996

ブルードラゴン

Microsoft

2007

CODED ARMS

KONAMI

2005

TOEO

namco

2006

マビノギ

nexon

2004

FF11

Square Enix

2007

MMOにおける応用

プロシージャルの歴史: ダンジョン自動生成

1980

Rogue Clone
ローグライクゲーム

Rogue

NetHack

1987～

不思議のダンジョン

チュンソフト

1993～

1996

Hellgate：London

flagshipstudio

2007

TOEO

namco

2006

FF11

Square Enix

2007

CODED ARMS

KONAMI

2005

MMOにおける応用

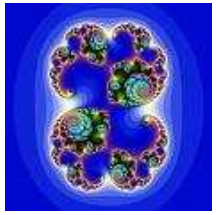
昔から部分的に応用されていた

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

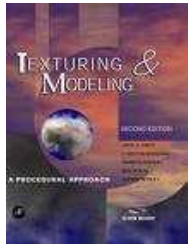
natFX MAX

パリン・ノイズ

1997,2002

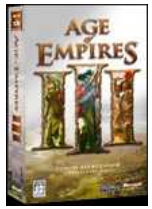
Pro FX

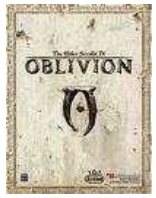
1988



1994

2003





プロシージャルの歴史: CG

1980 1990 2000

フラクタル幾何学

1975

フラクタル・イメージ

SIGGRAPH 1987

地形自動生成

植物自動生成

2Dモデル生成

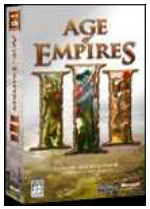
Terragen

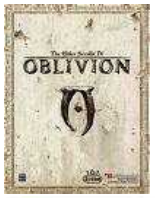
SpeedTree

natFX MAX

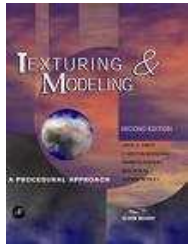
1997,2002

Pro FX





1988



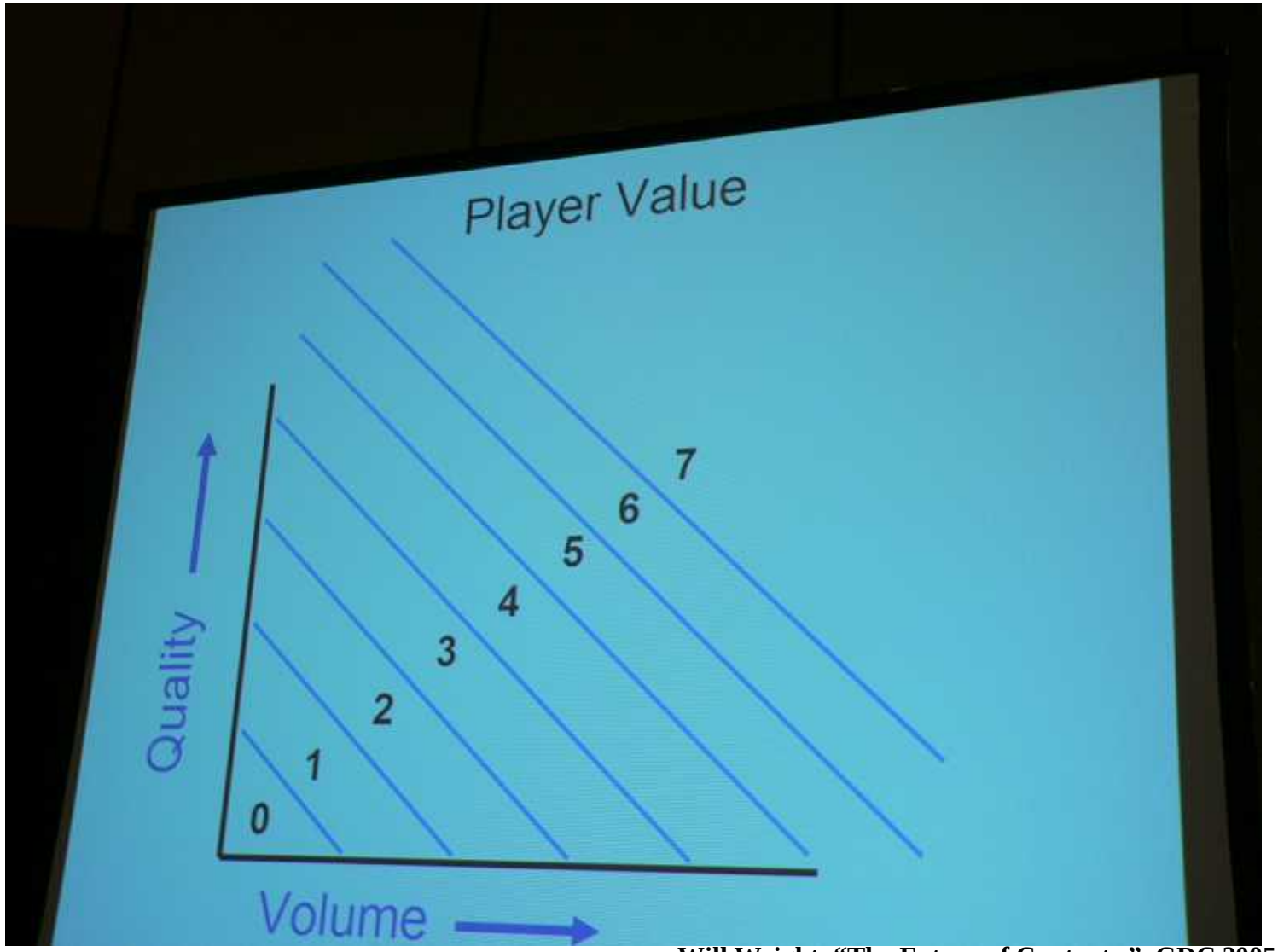
1994

2003

昔から大学では研究されていた

今、ゲームにとってプロシージャルとは何か？

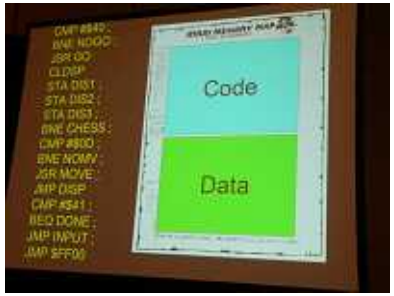
ゲームとは何か？

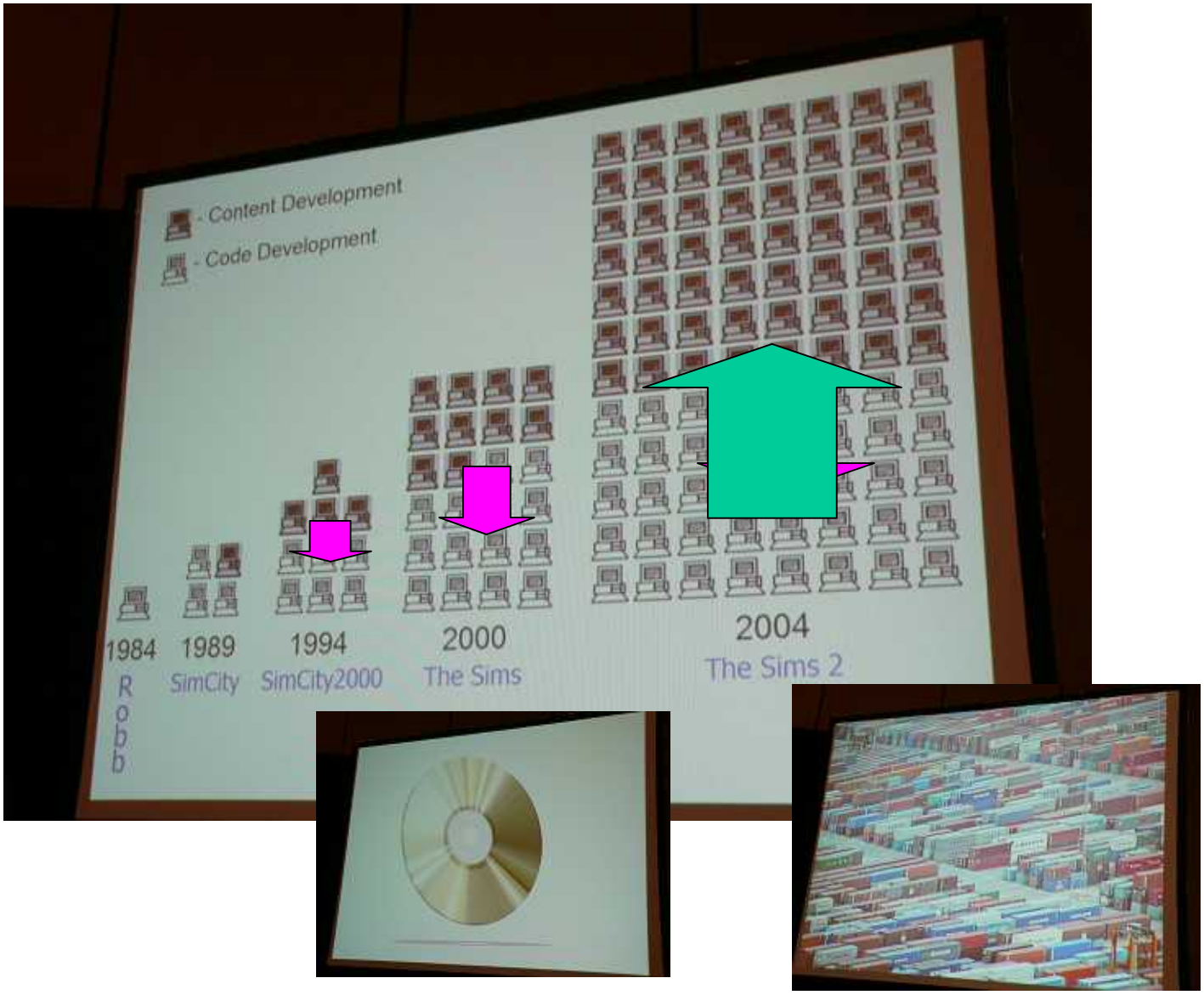


Will Wright, “The Future of Contents ” GDC 2005

Quality × Volume

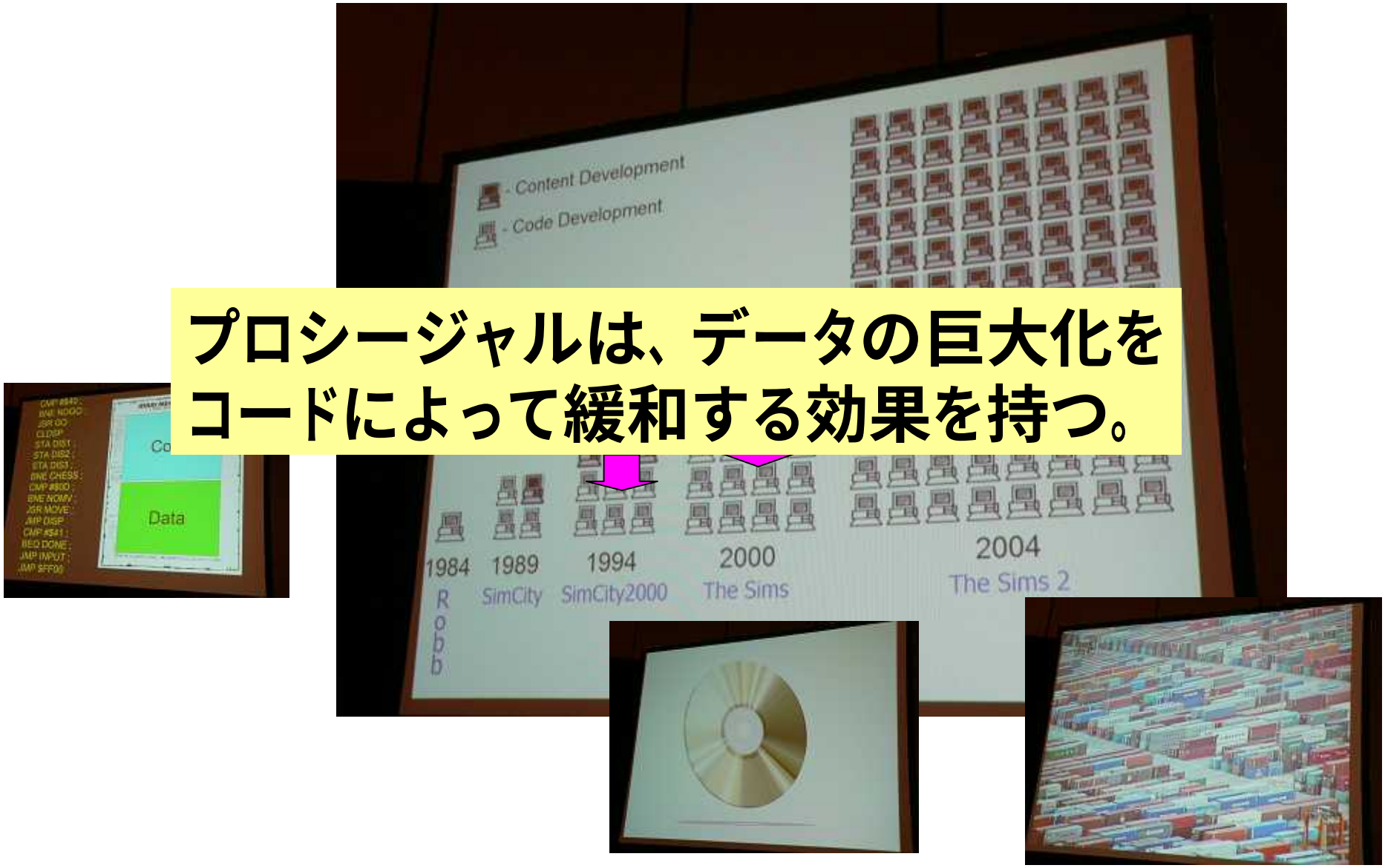
データとコードの比率の変化





Will Wright, “The Future of Contents ” GDC 2005

データとコードの比率の変化

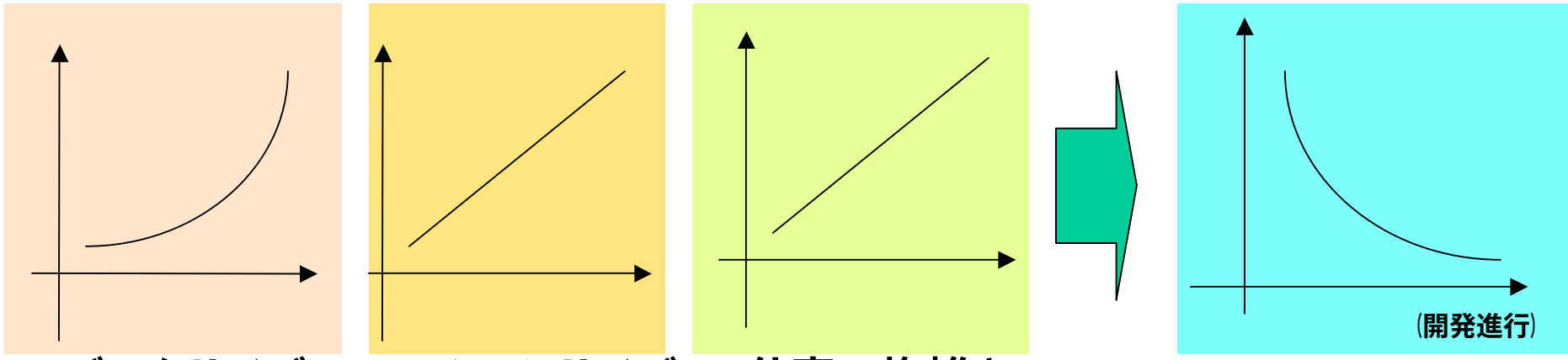


プロシージャルは、データの巨大化をコードによって緩和する効果を持つ。

Will Wright, “The Future of Contents ” GDC 2005

これからのゲーム開発の問題点

Rushing into a wall (壁にぶつかる)



データサイズ　チームサイズ　仕事の複雑さ

(ポンからアサシンクリードまでを横軸にとっている)

コンテンツの変化させやすさ

① コードに比べてコンテンツの容量が増大

② コンテンツ開発費の増大

③ データの相互関連性のために開発が進むにつれてデータが固定

開発費の増大　x　開発体制の硬化

(”Procedural Data Generation in Far Cry2” GDC2008 の図より)

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て、
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？

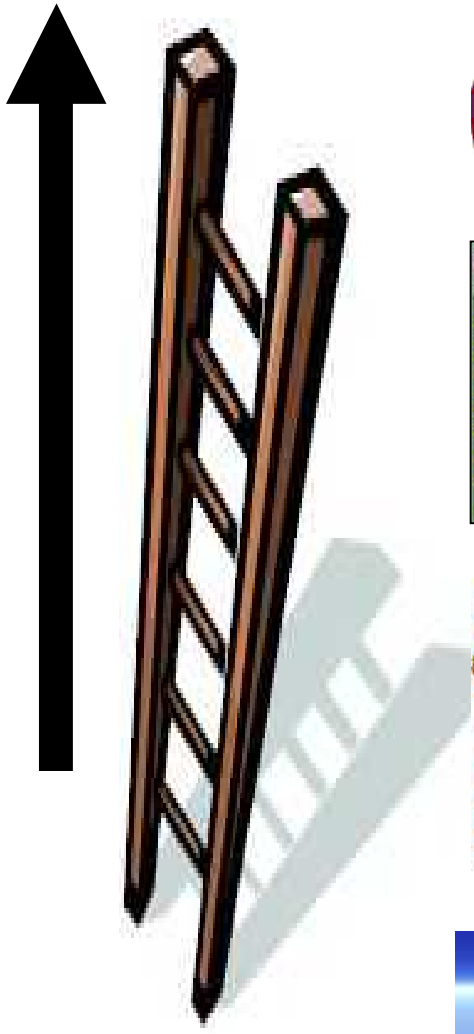
プロシージャル・コンテンツ・ジェネレーション

…だけでは足りない。

The Transmogrifier

Original Fountain

Transmogrified Fountain



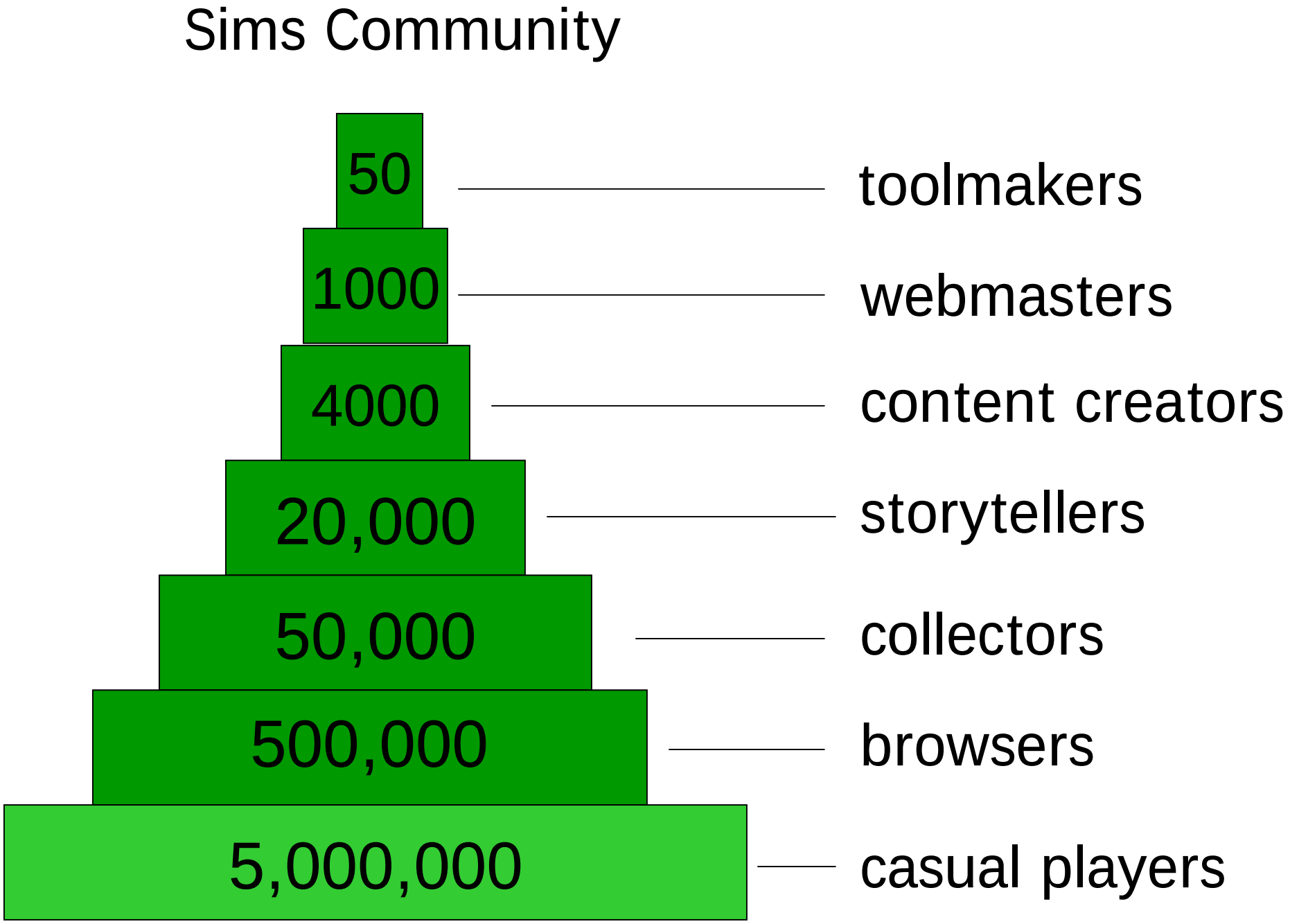
Creators

Webmasters

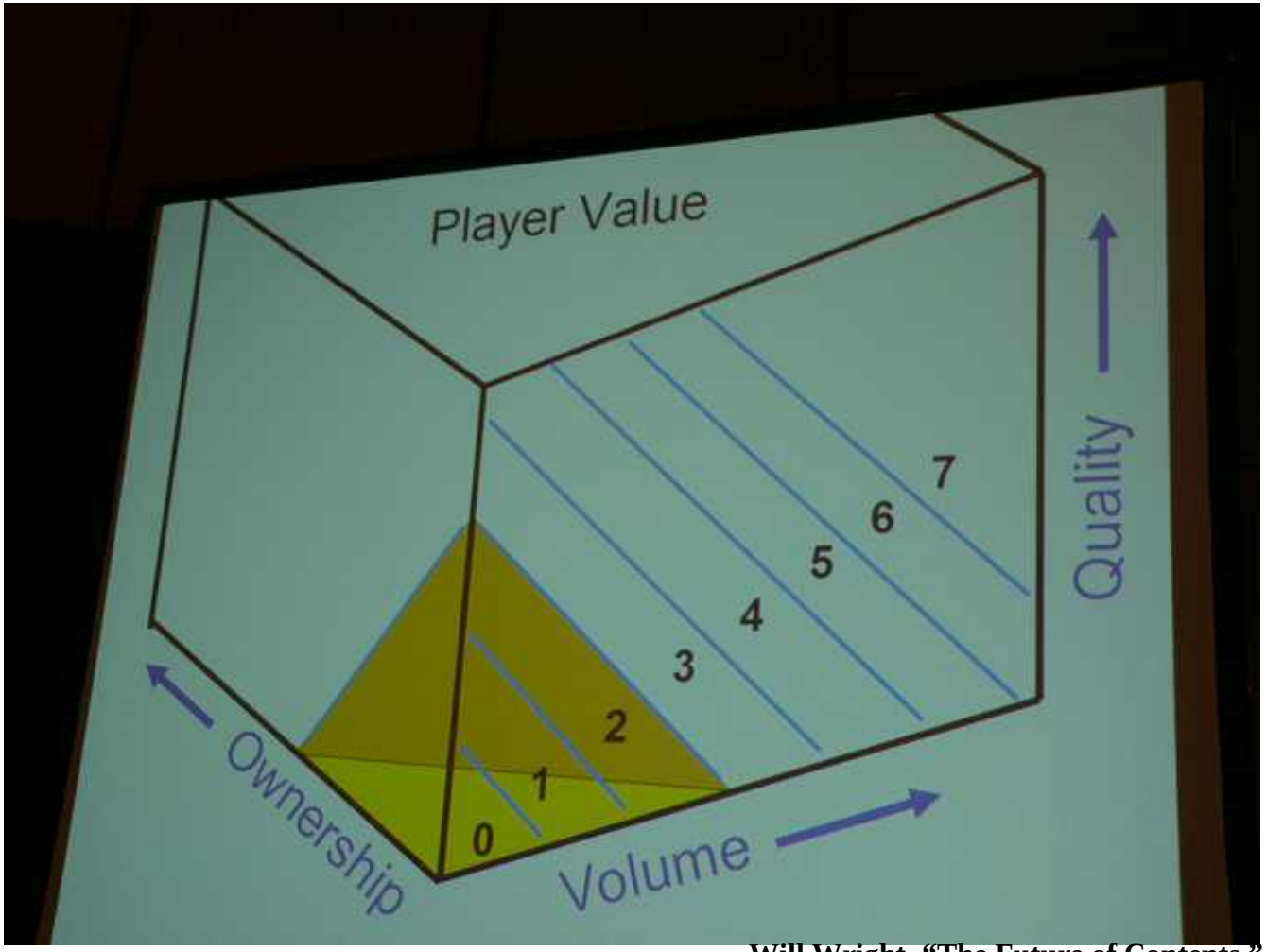
Collectors

Casual Players

rung spacing


Sims Community
50
toolmakers
1000
webmasters
4000
content creators
20,000
storytellers
50,000
collectors
500,000
browsers
5,000,000
casual players

ゲームとは何か？



Will Wright, “The Future of Contents ” GDC 2005

Quality × Volume × Ownership

「これからはデータ生成をユーザーに解放する時代である。」

（シムズ・オンラインで培って来たウィル・ライトの思想）

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て、
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て、

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？

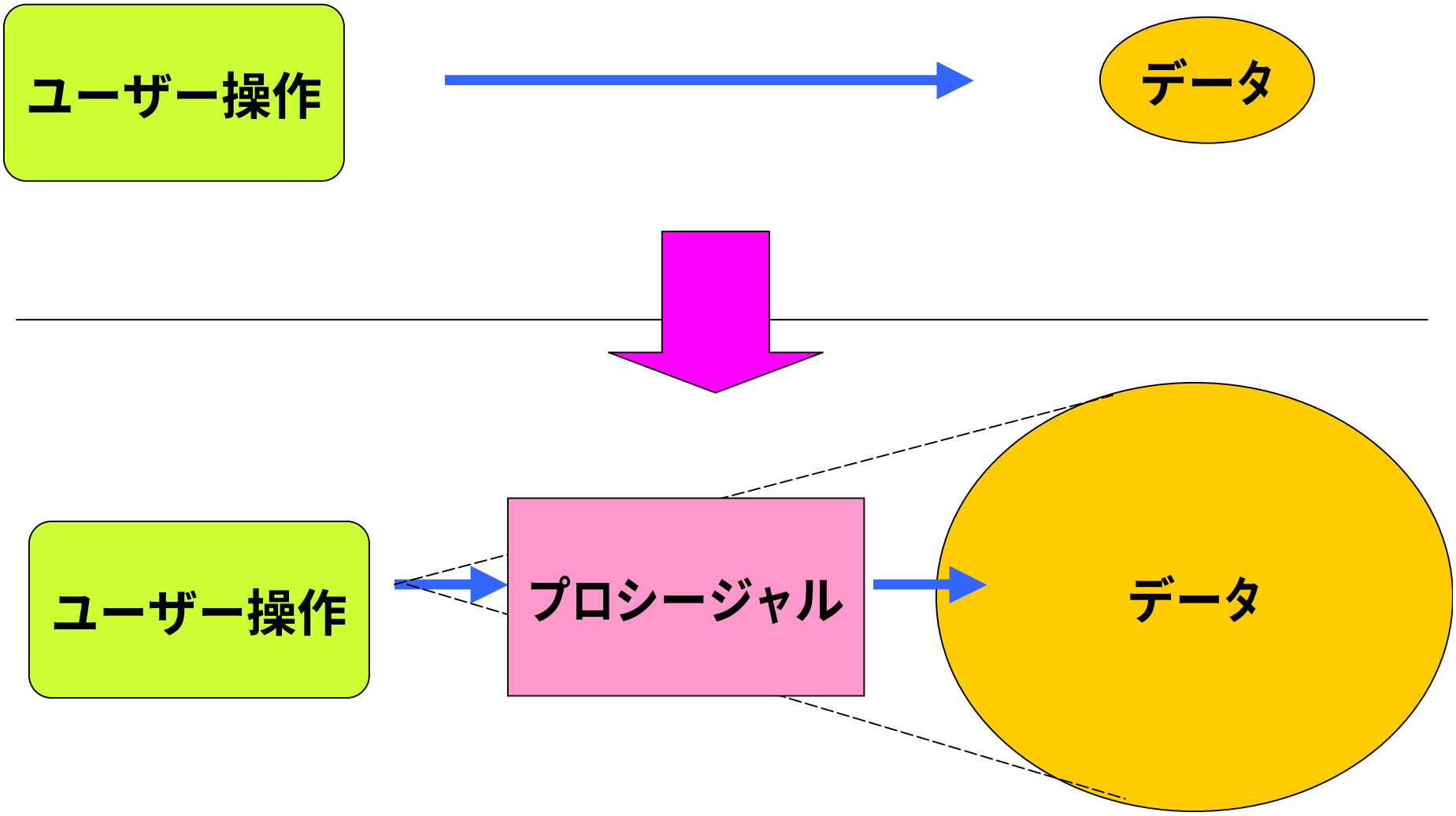
Spore

プロシージャル・コンテンツ・ジェネレーション

×

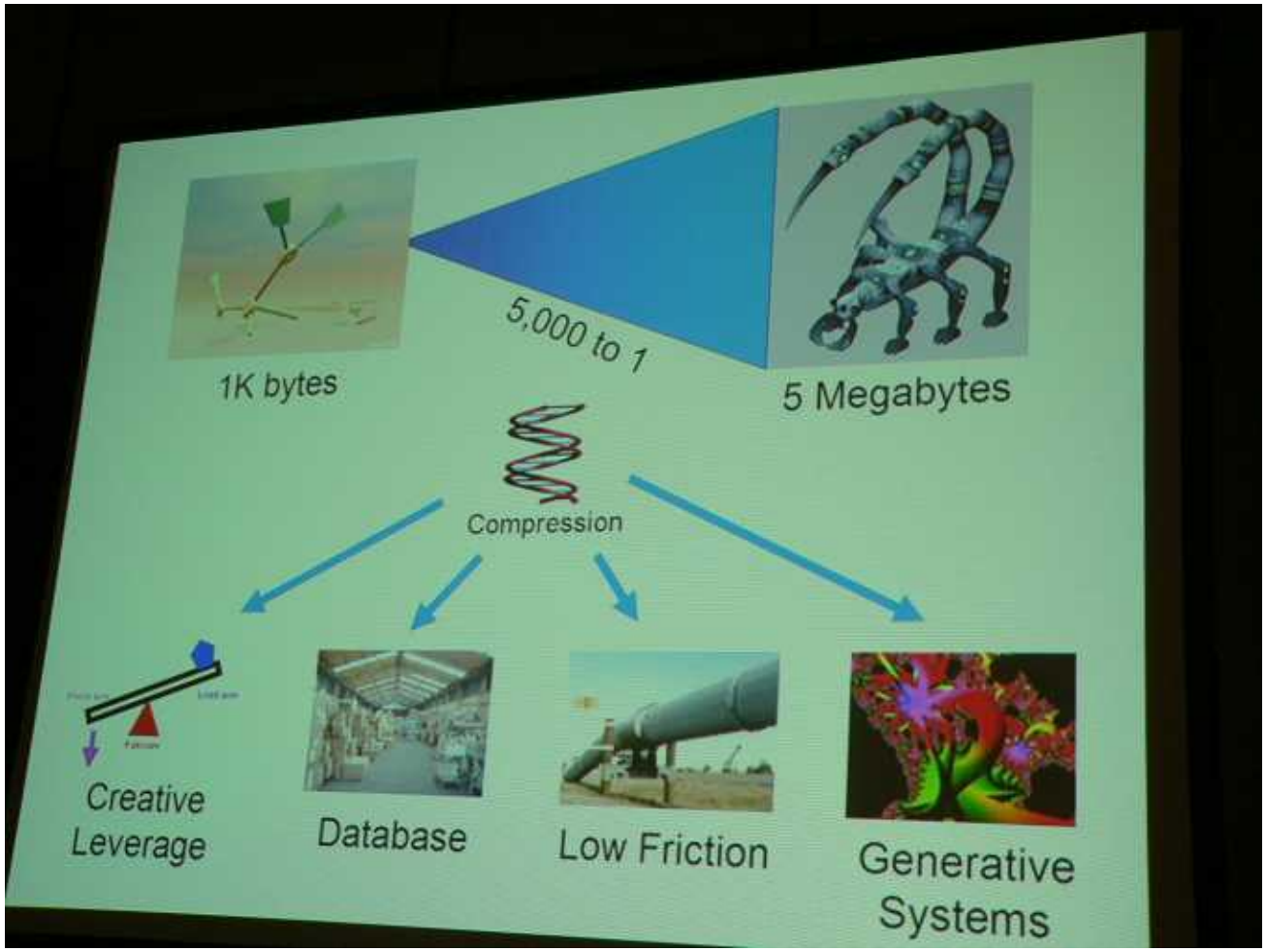
ユーザー・ジェネレイティッド・コンテンツ

Spore のゲームデザインの本質



プロシージャルはユーザーの労力を最大化する。
Sporeはその最大化された成果を利用する。

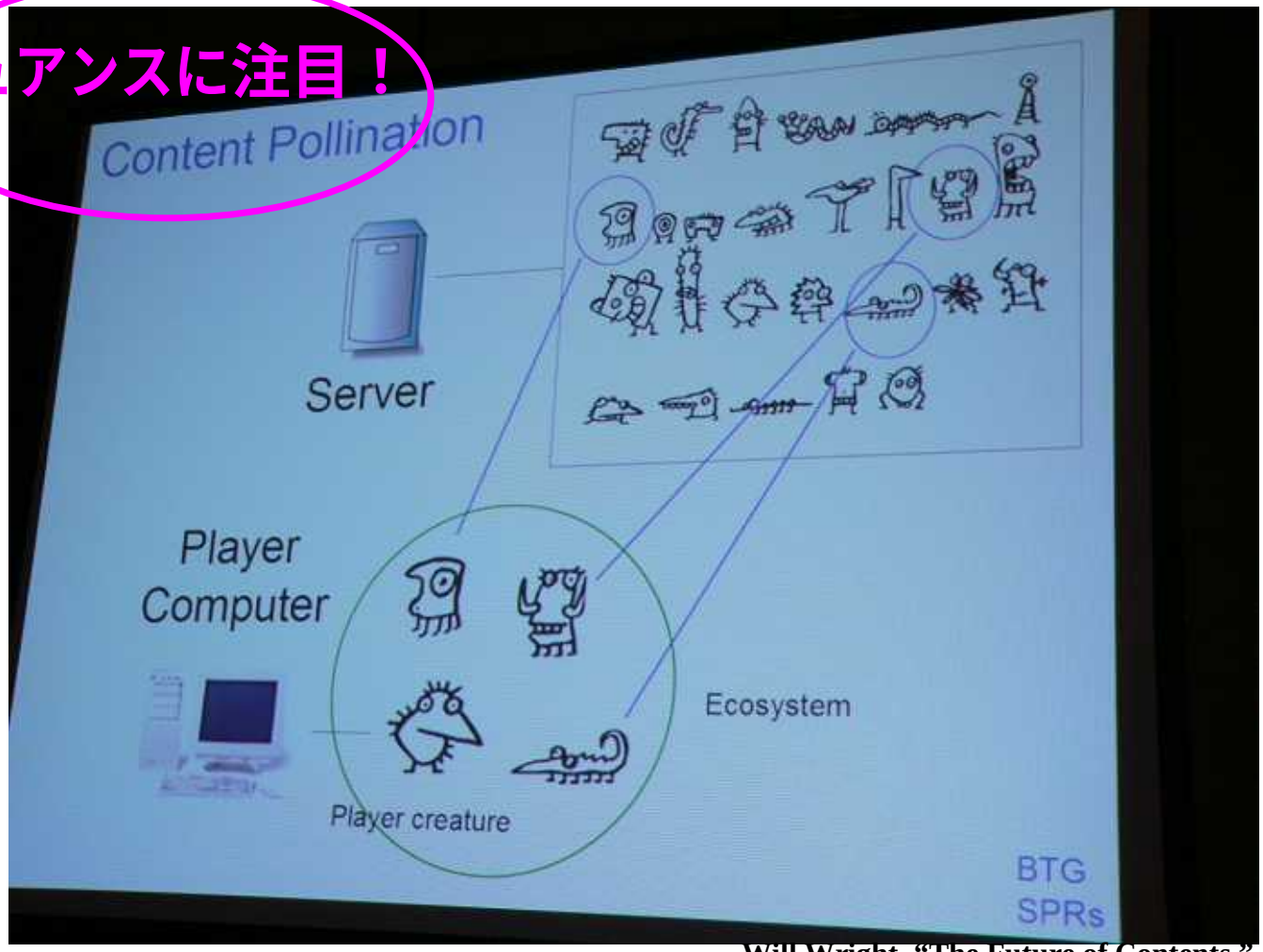
Spore のゲームデザインの本質



Will Wright, “The Future of Contents ” GDC 2005

Spore のゲームデザインの本質

このニュアンスに注目！



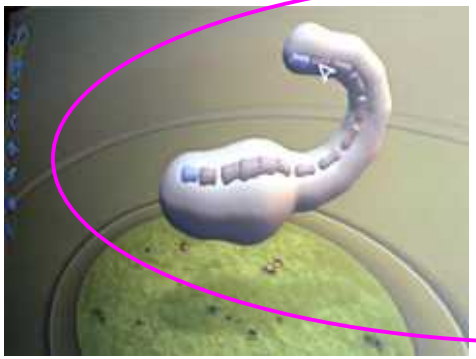
Will Wright, “The Future of Contents” GDC 2005

Pollination = **配布（もとの意味は花粉を飛ばす）**

技術詳細

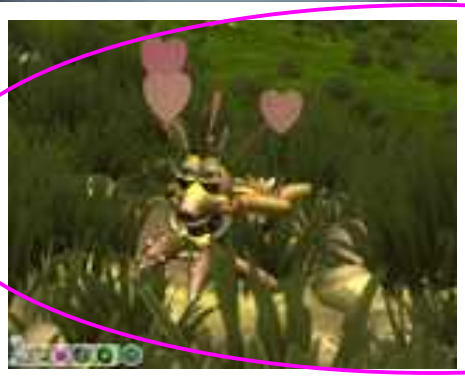
Spore とはどんなゲーム？

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



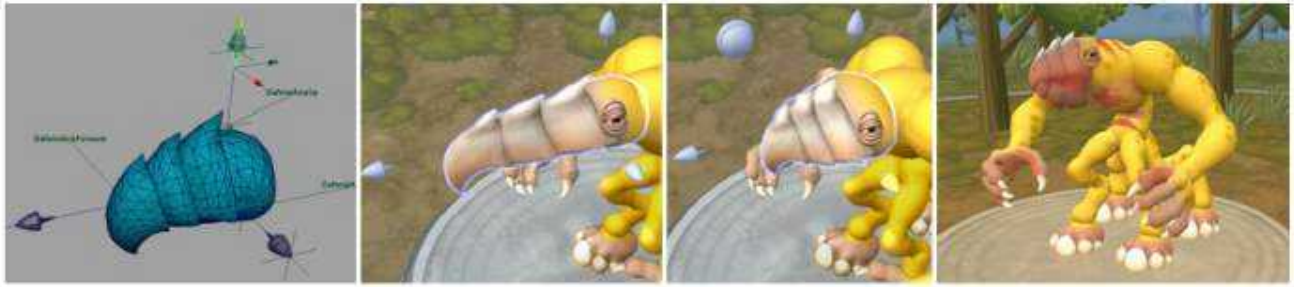
集団行動フェーズ

ギャラクティックフェーズ

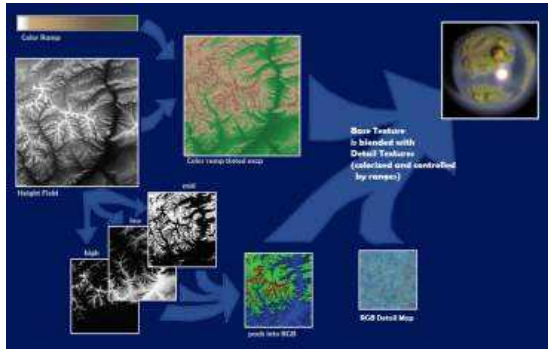
http://www.4gamer.net/news/history/2006.05/20060511195155detail.html

「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

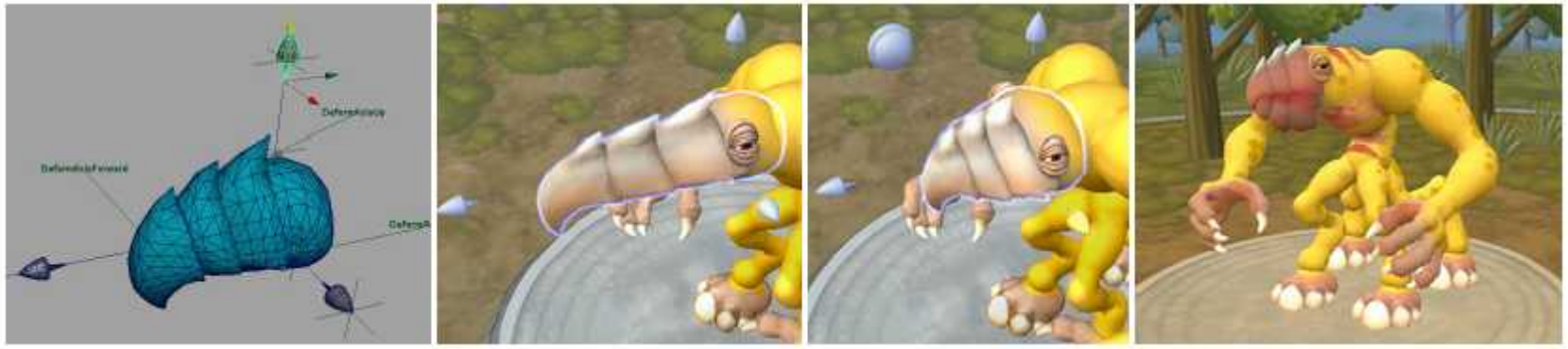


① テクスチャー（SIGGRAPH2007）
② オブジェクト配置（SIGGRAPH2007）
③ モデル生成（SIGGRAPH2007）
④ モデル変形（SIGGRAPH2007）
⑤ アニメーション生成 (SIGGRAPH2008)
＋⑥ 音楽プロシージャル（GDC２００８初公開）

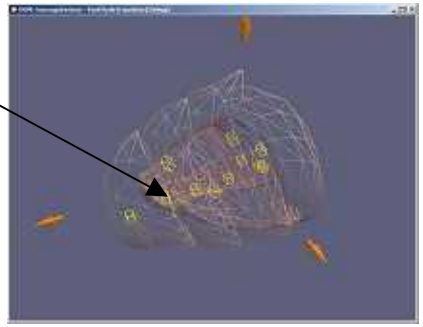


(1) モデル生成

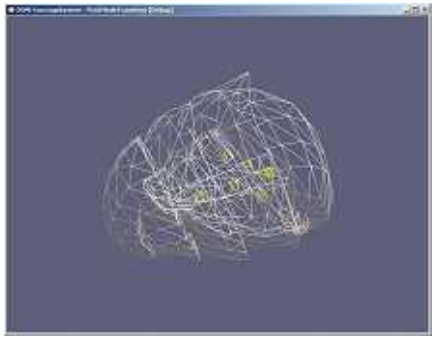
① モデルは「リゴブロック」と呼ばれるパーツからなる
② 「リゴブロック」はマウスによって制御ポイントを動かして
変形することができる



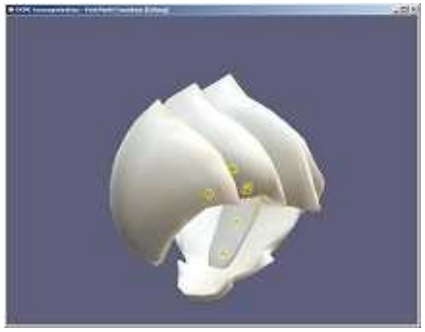
ハンドル



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）

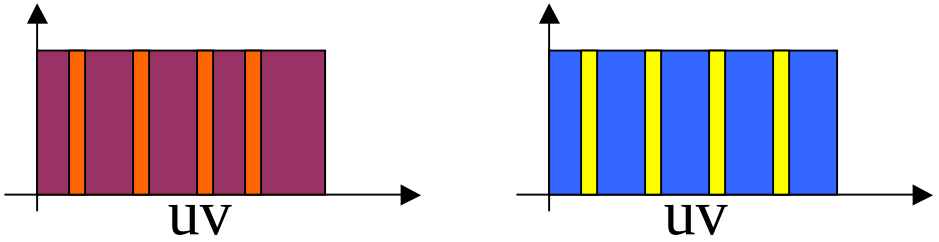
Lydia Choy, Ryan Ingram, Ocean Quigley, Brian Sharp, Andrew Willmott
Rigblocks: Player-Deformable Objects, http://www.cs.cmu.edu/~ajw/s2007

(2)テクスチャ生成

細かすぎない　シンプルすぎない

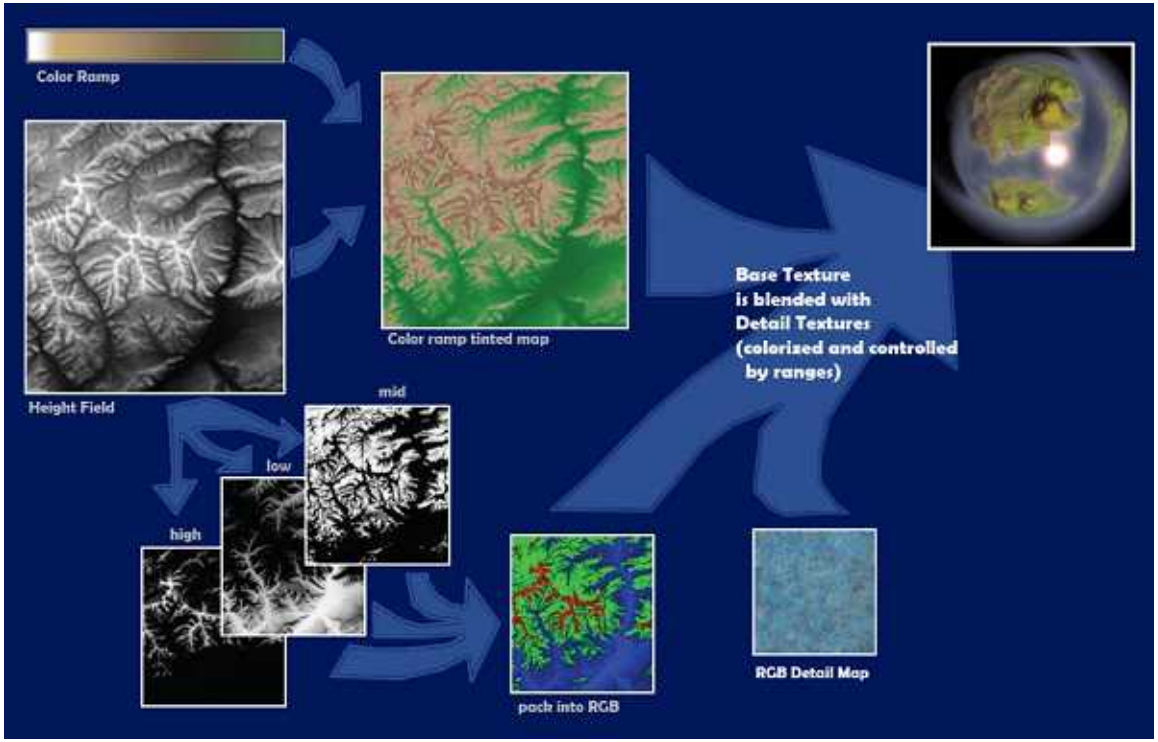


ユーザーが選ぶ



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

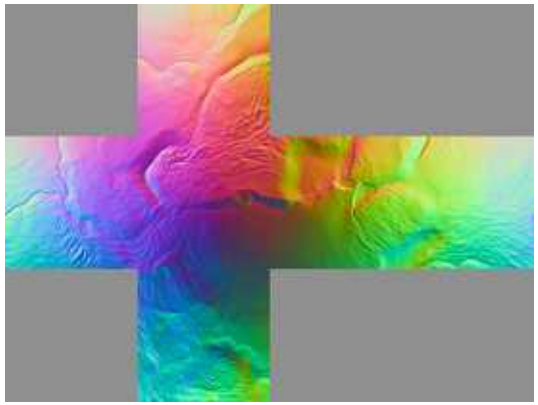
(3) 星を創る



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



法線マップ

射影
（ヤコビ行列）

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
つけることが出来る

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

Procedural Brushing Demo

3rd spore demonstration at SIGGRAPH 07 (YouTube)

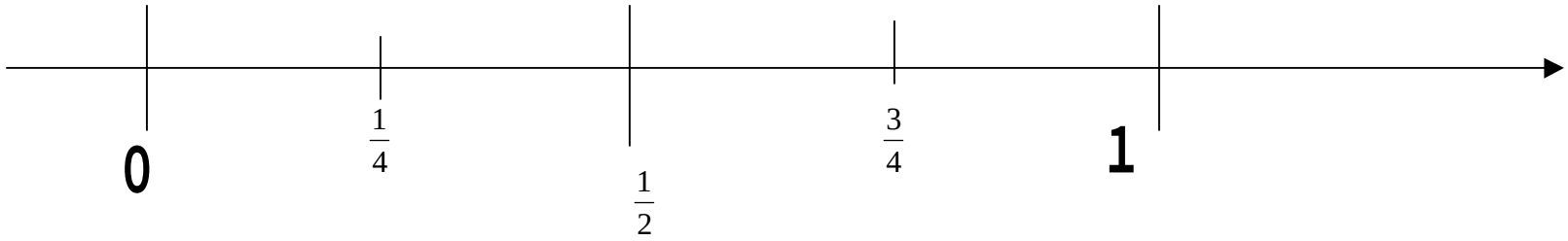
(4)　惑星表面に植物を植える

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

ハルトン･シークエンス

ハルトン･シークエンス

① ベースとなる素数を一つ選ぶ。例えば2
② その素数で、(0，1) 区間を選んだ素数の数に分割する
③ 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, …..

1次元ハルトン･シークエンス

1/3 , 1/9, 2/9, 4/9, 5/9, 2/3, 7/9, 8/9, …..

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

ハルトン･シークエンス

ハルトン･シークエンス

1次元ハルトン･シークエンス

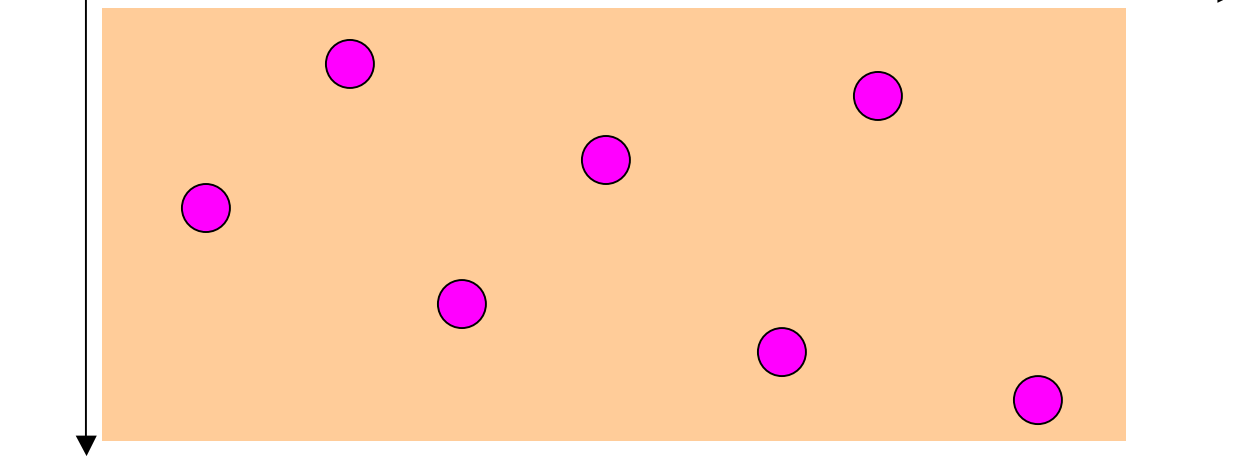
1/2 , 1/4, 3/4, 1/8, 3/8, 5/8, 7/8, …..

1/3 , 1/9, 2/9, 4/9, 5/9, 2/3, 7/9, 8/9, …..

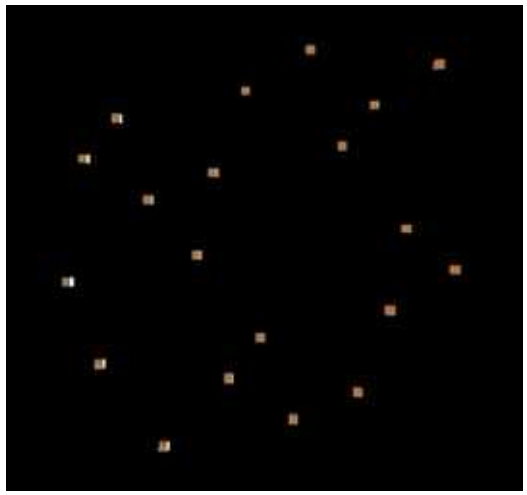
(1/2,1/3), (1/4,1/9), (3/4,2/9), (1/8,4/9) ….

2次元ハルトン･シークエンス

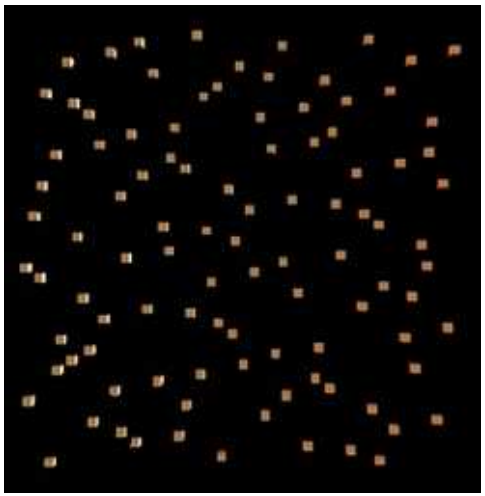
大きい素数を使わなければだいたい均等に分布する



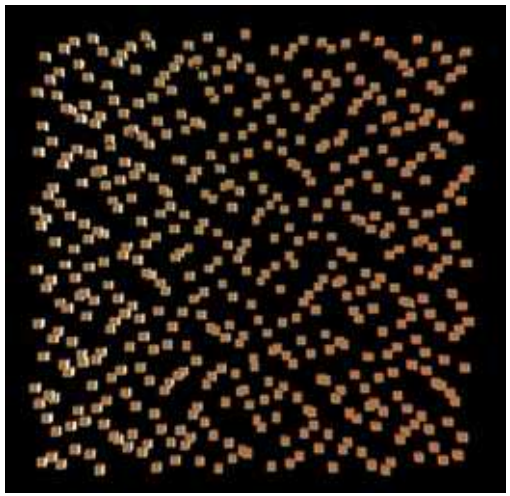
実際は、割り算をビットシフト演算に変えた
高速な *Incremental Halton Sequence* ***を使う***



20



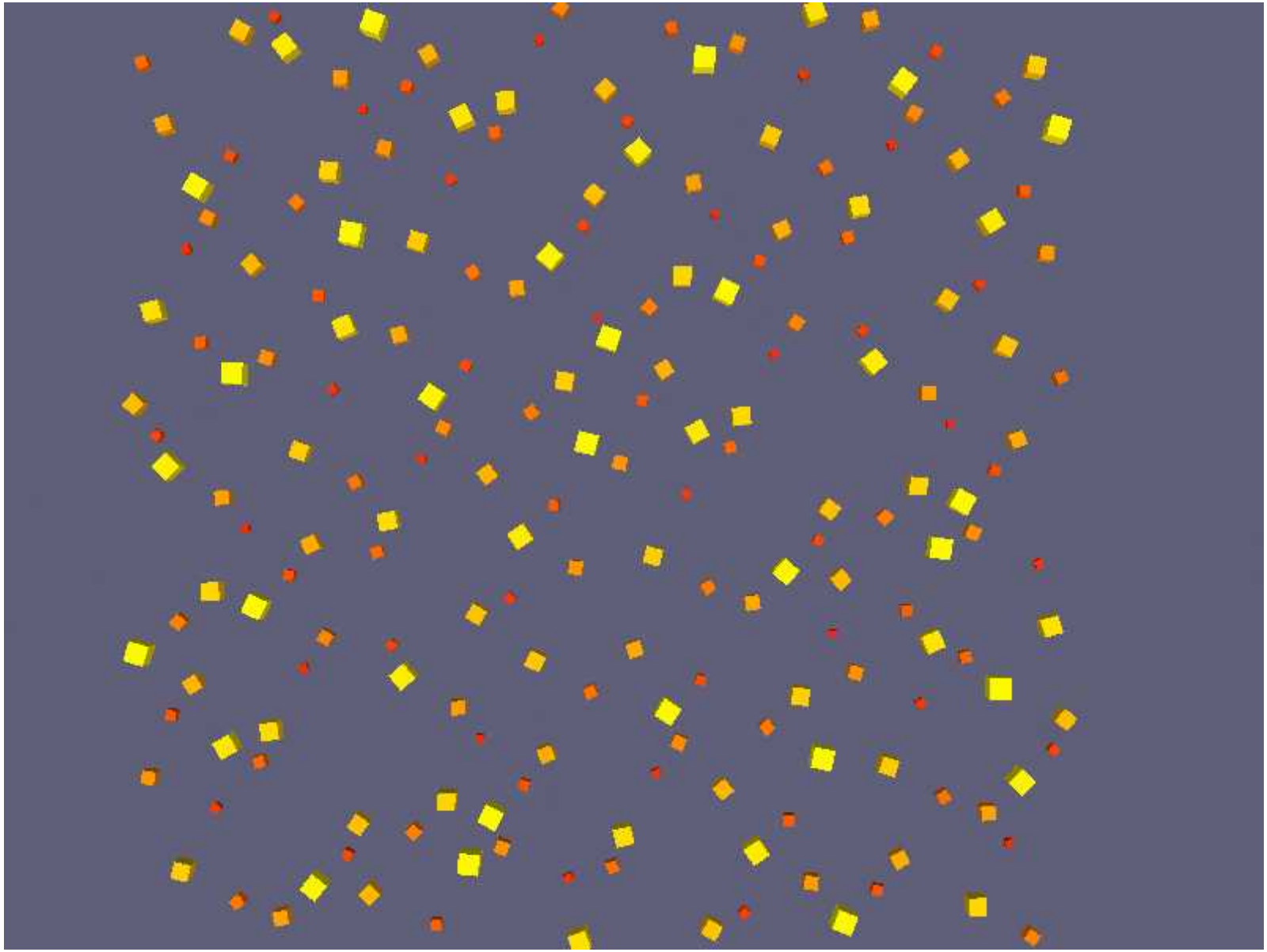
100



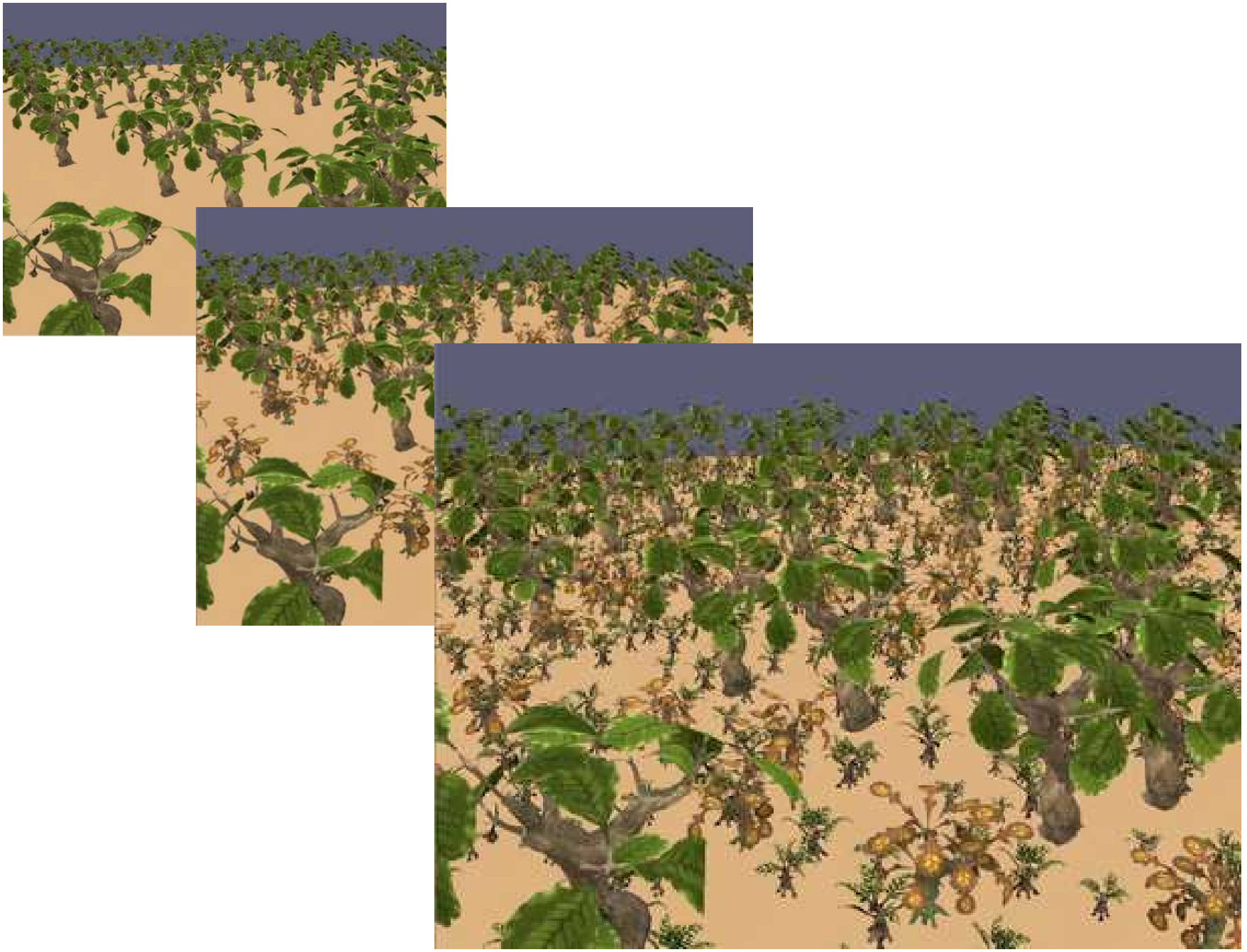
500

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

大きさ、回転、色を乱数で決定

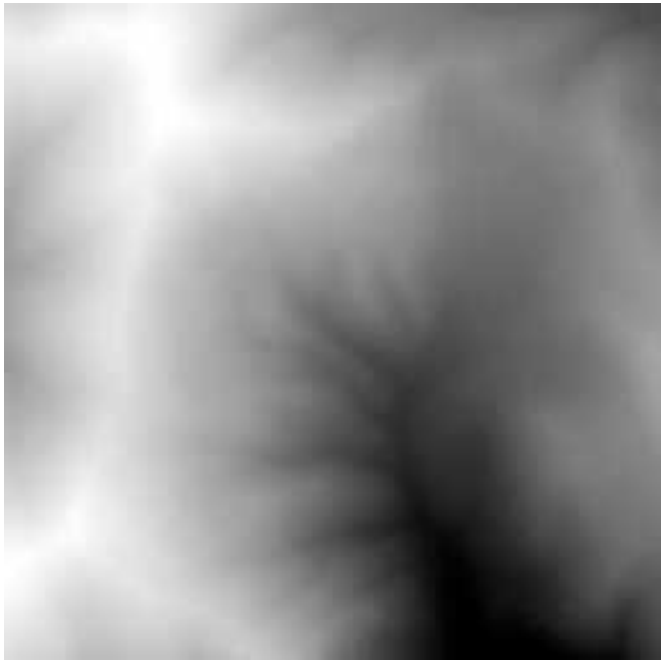


Andrew Willmott,Creating Spherical Worlds, http://www.cs.cmu.edu/~ajw/s2007

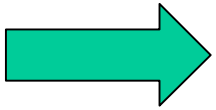


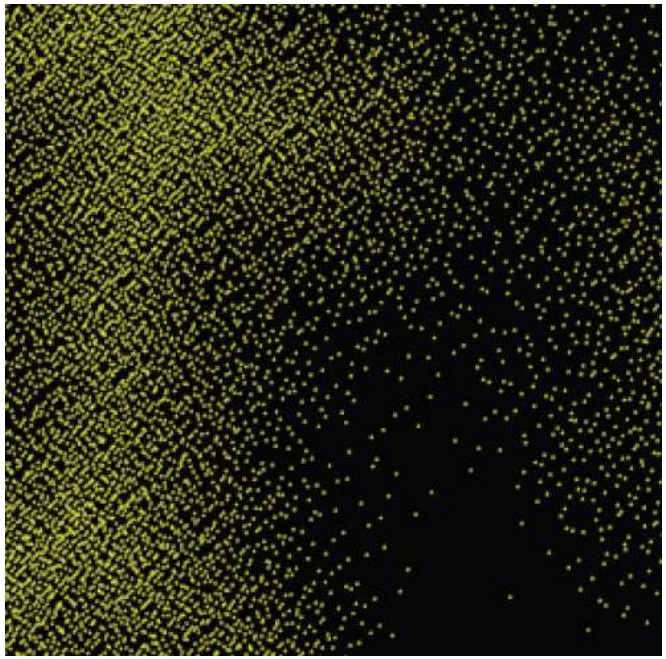
Andrew Willmott,Creating Spherical Worlds, http://www.cs.cmu.edu/~ajw/s2007

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

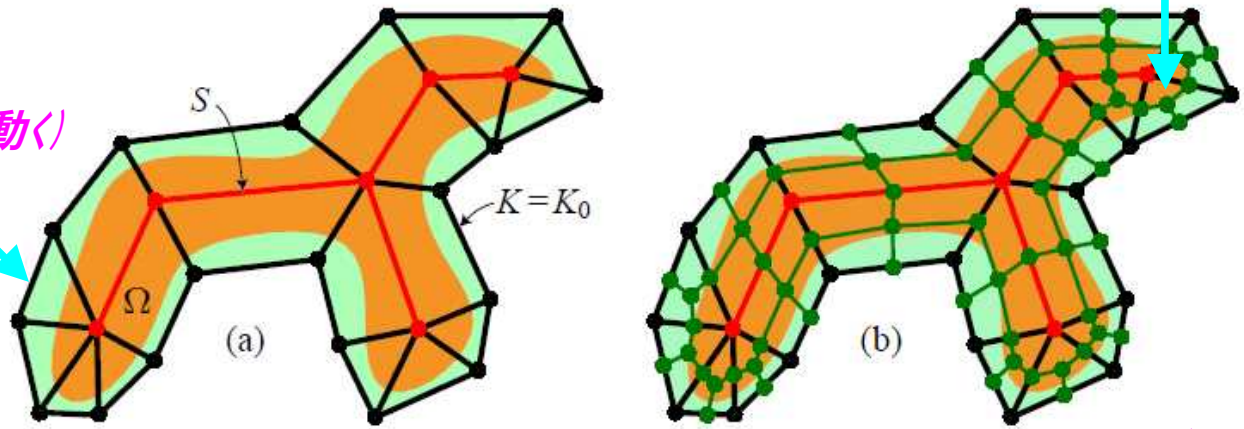
(5) アニメーション（Development of Sporeの情報による）

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テクニック

骨を動かすと

コントロール ラティス（骨と同期して動く）



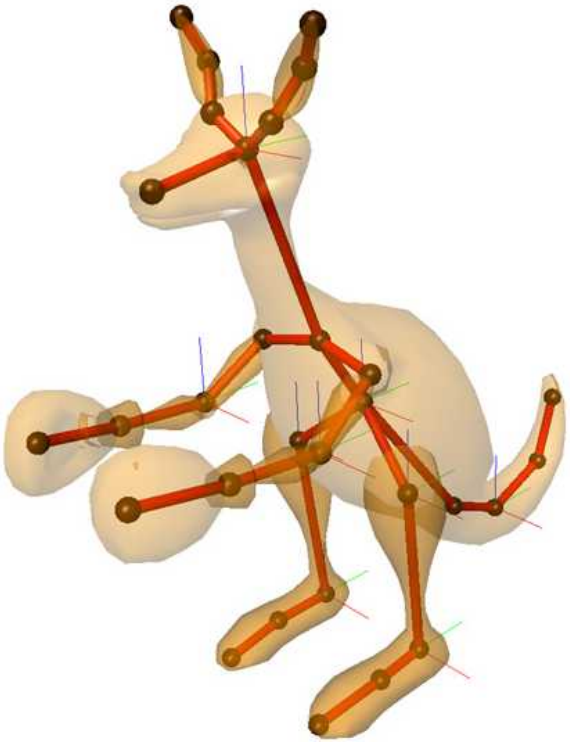
エッジでつながった肉も連動して動くのだ

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



スケルトン

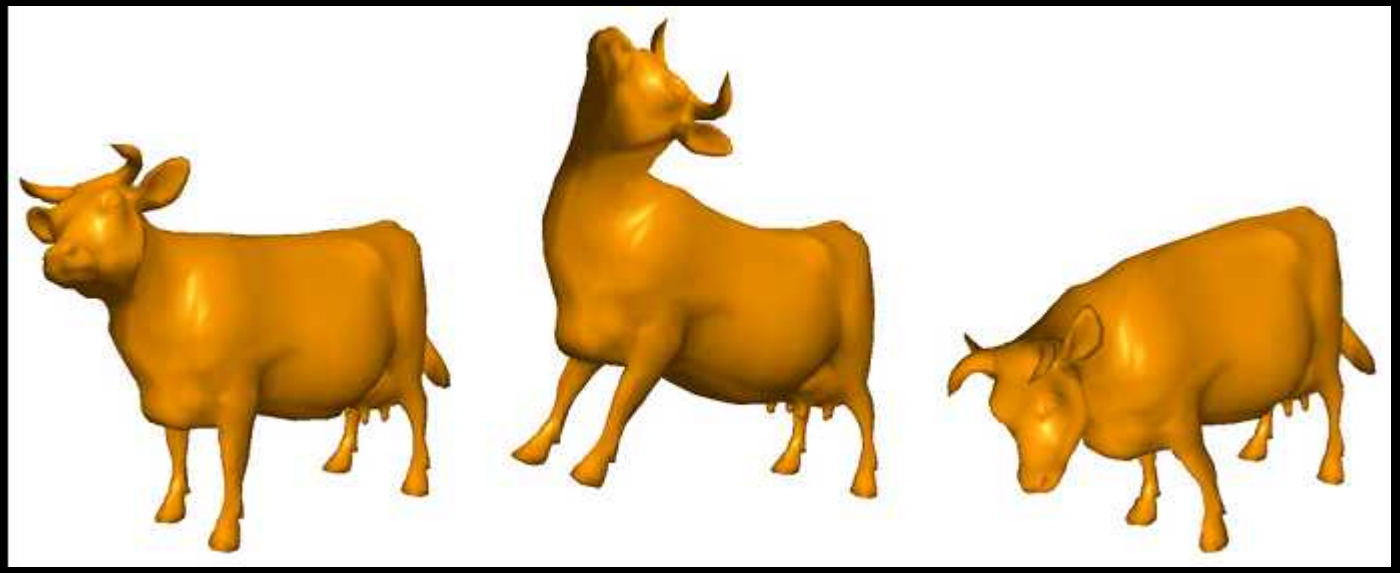
色ごとに
局所座標系が
入っています

- 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-divx

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。**

キャラクターの形状に
依存しない一般空間の
データ

Specialize(Re-targeting)

“S関数”

“G関数”

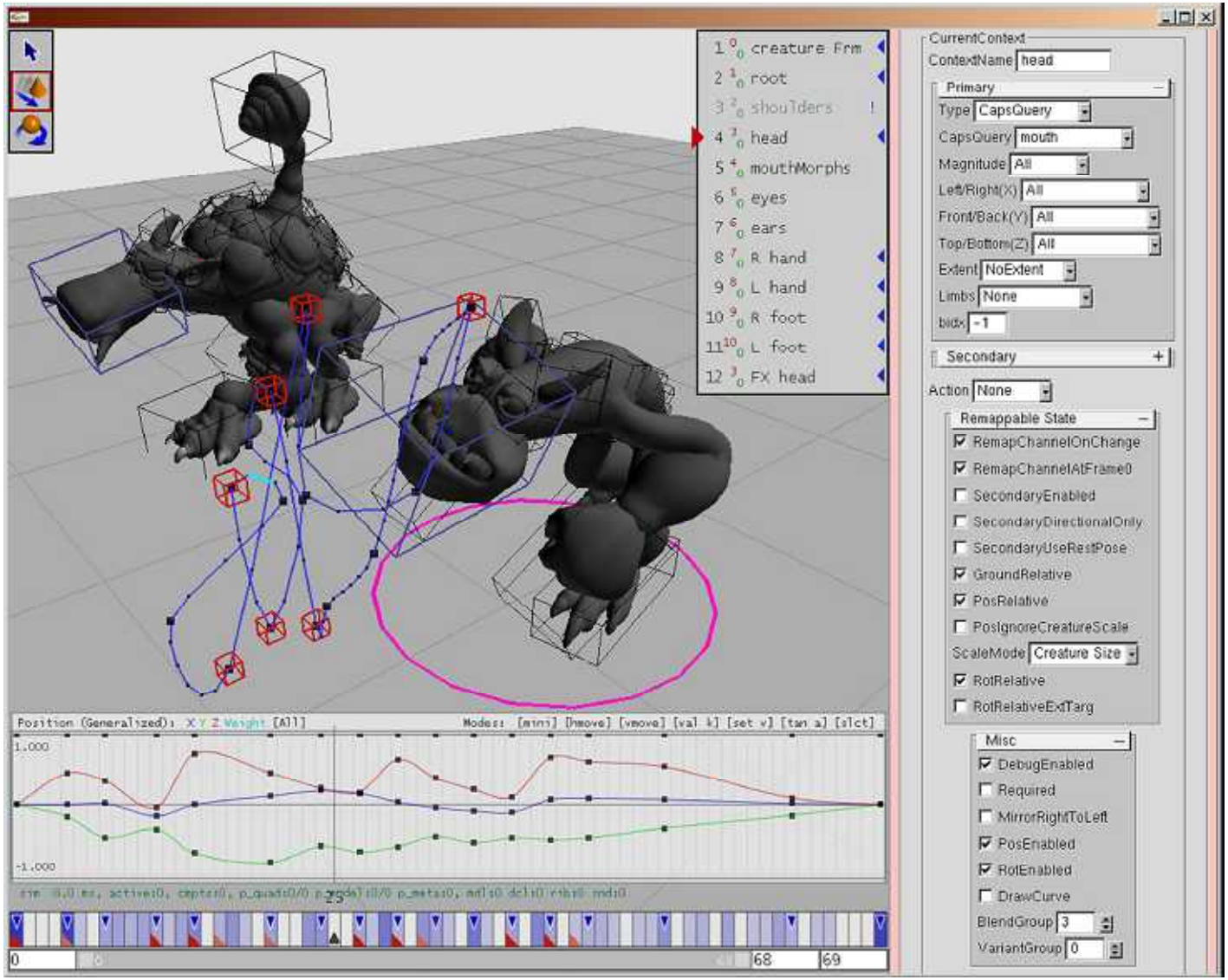
Generalize(Re-targeting)

アニメーターが
部位と動きをGUIで指定

SIGGRAPH2008

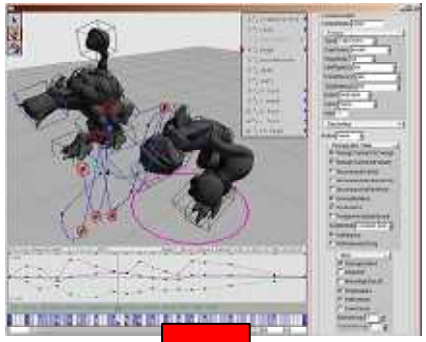
“Real-time Motion Retargeting to Highly Varied User-Created Morphologies”
http://chrishecker.com/Real-time_Motion_Retargeting_to_Highly_Varied_User-Created_Morphologies

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



キャラクターの形状に
依存しない一般空間の
データ

キャラクターの形状に
依存するデータ

“G関数”

*(一般の問題と
特殊の問題を分離)*

“S関数”

Synthesize

Playback

特殊な問題

(6) SPORE におけるプロシージャル･ミュージック

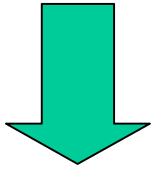
http://pc.gamespy.com/pc/spore/853810p1.html

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.cmpevents.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



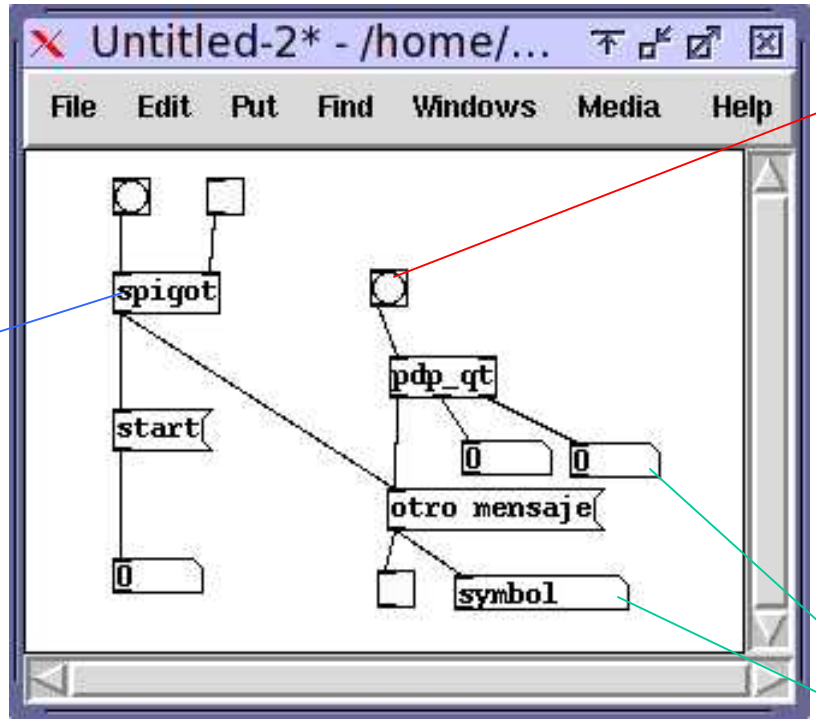
Procedural Music
(その場で生成される音楽)

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



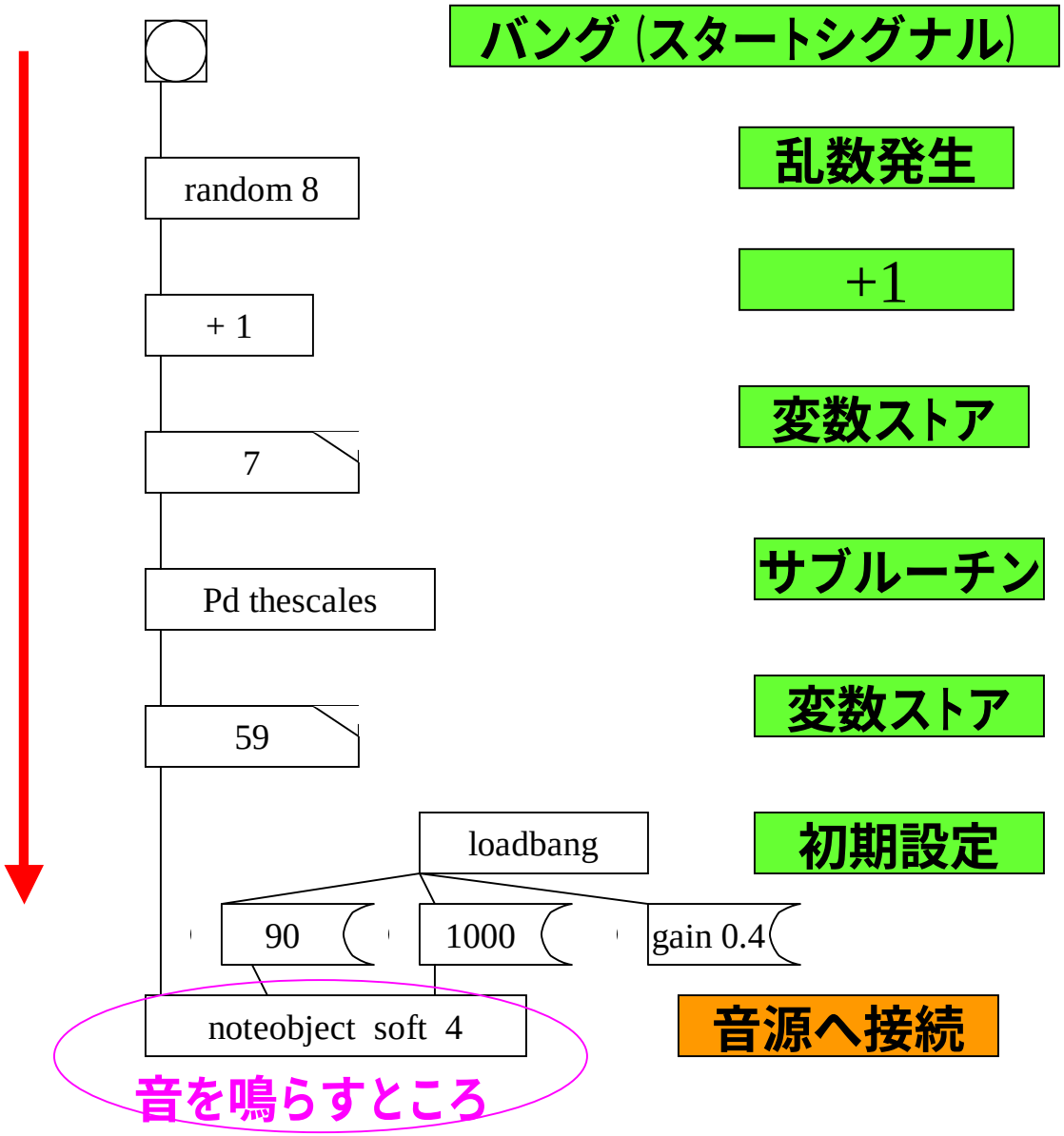
Bang オブジェクト
(単にトリガーを発信)

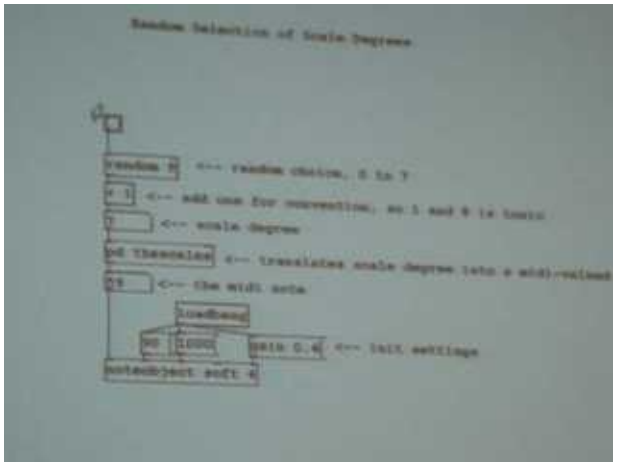
オブジェクト
を組み合わせる
(中に書いてある文字は
決められてた操作を表す)

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

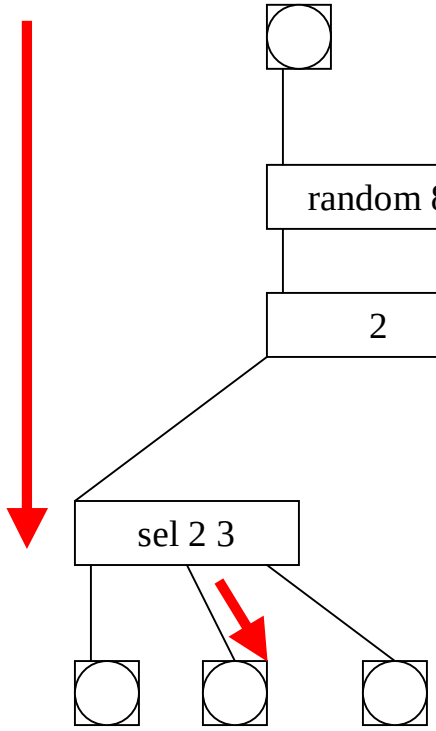
簡単な解説
http://ja.wikipedia.org/wiki/Pure_Data

Pd とは？


バング（スタートシグナル）
乱数発生
+1
変数ストア
サブルーチン
変数ストア
初期設定
音源へ接続
random 8
+ 1
7
Pd thescales
59
loadbang
90
1000
gain 0.4
noteobject soft 4
音を鳴らすところ



Pd とは？

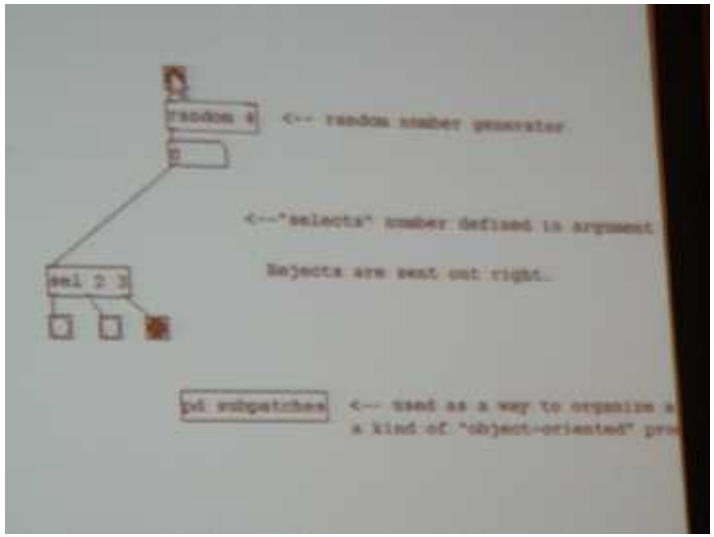


バング（スタートシグナル）

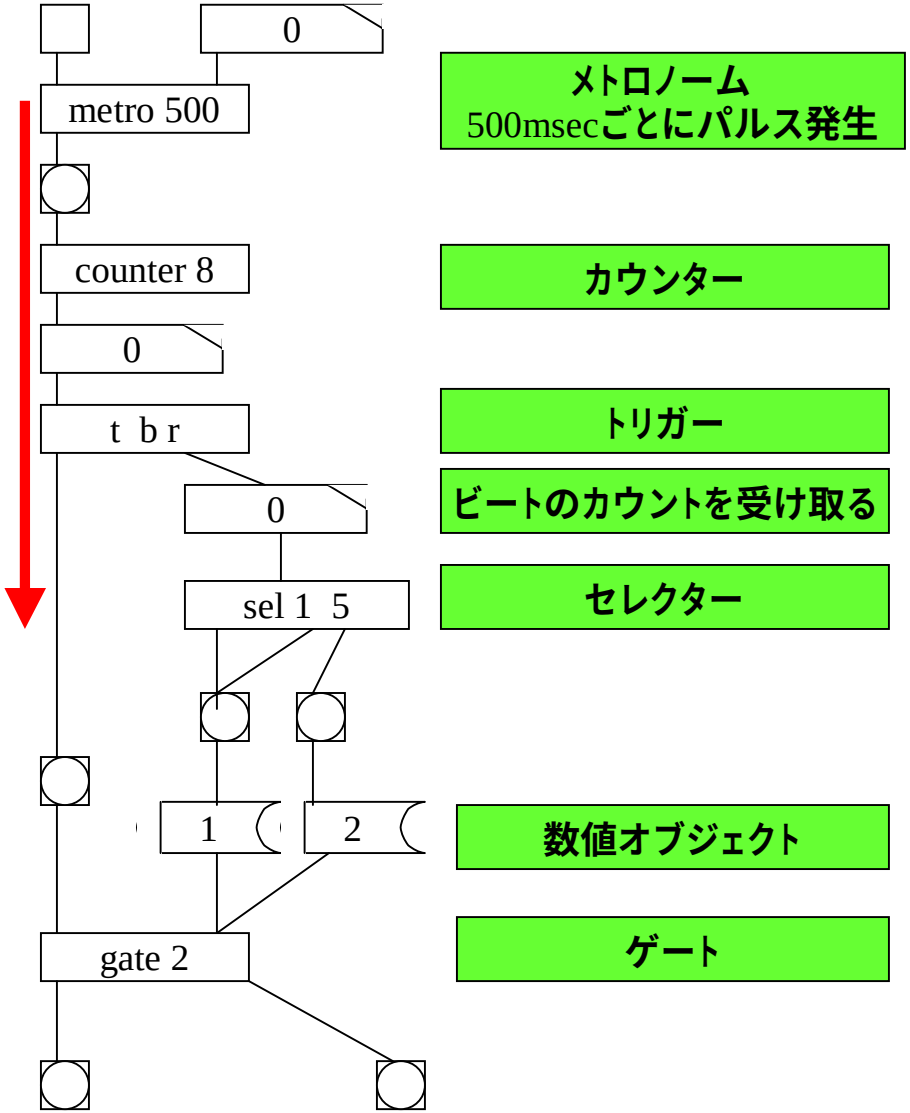
乱数発生

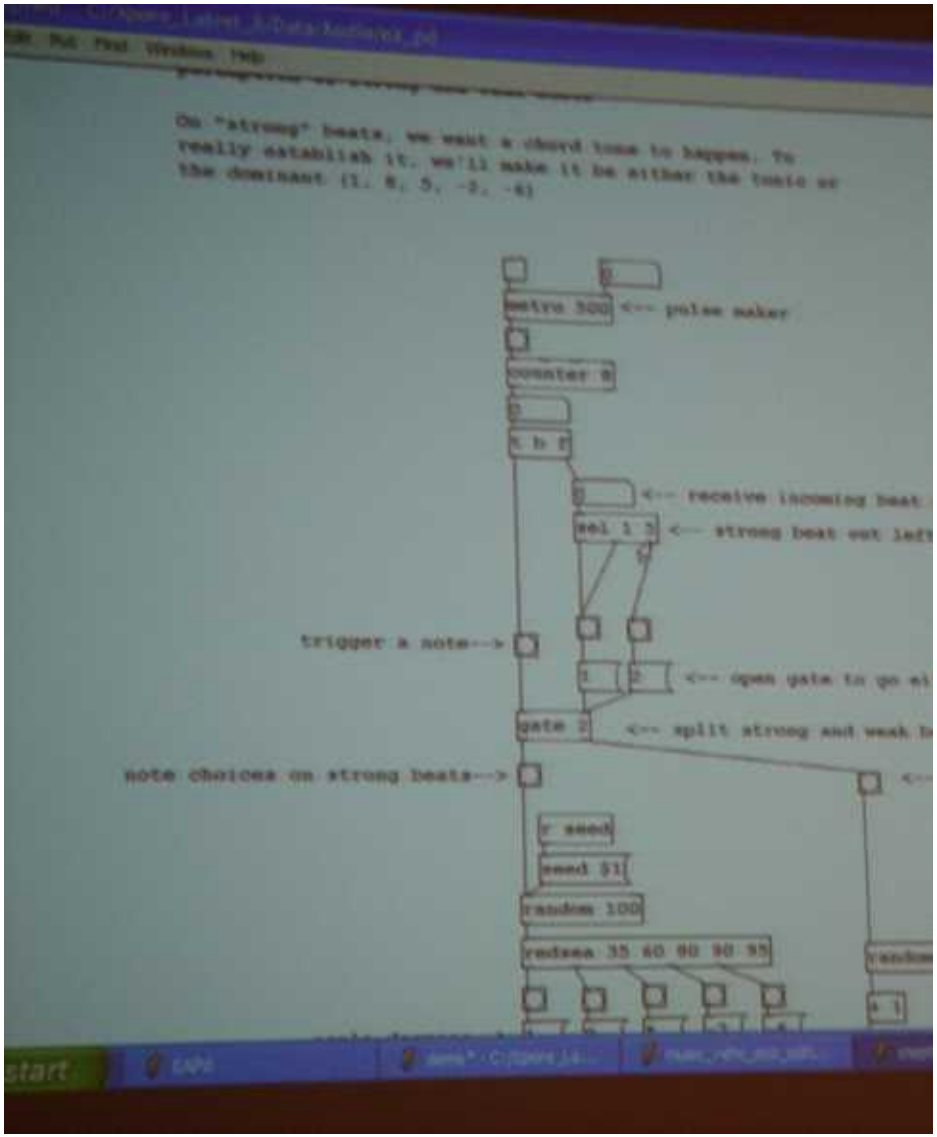
変数ストア

セレクター　2が来れば一番左、
3が来れば真ん中、それ以外は一番右

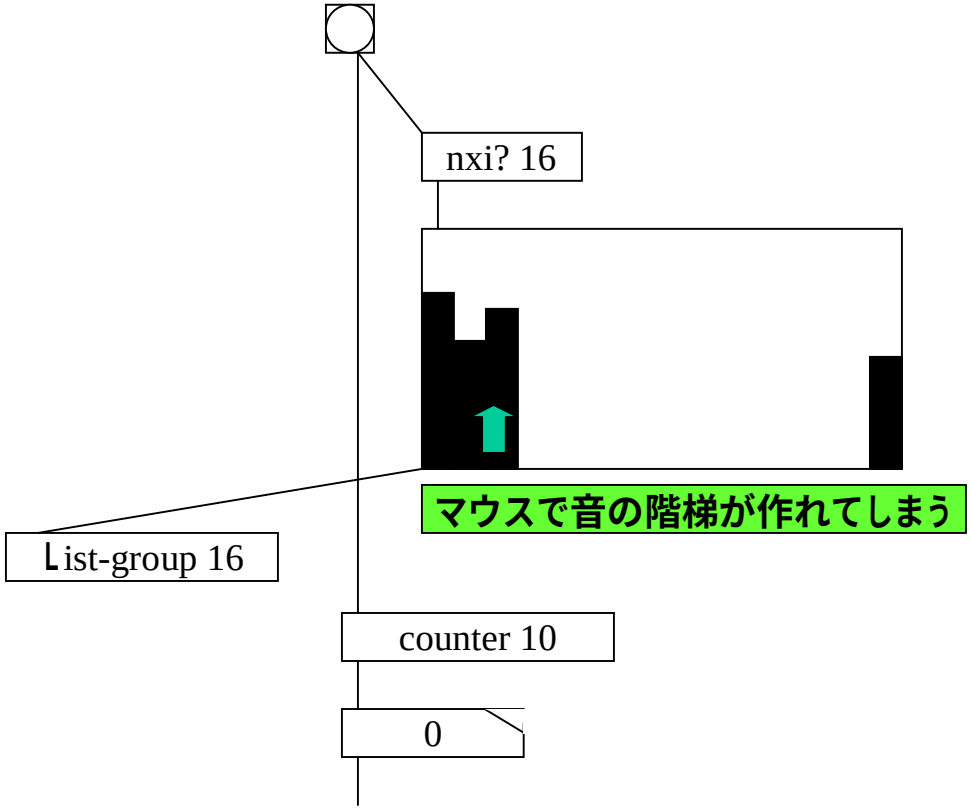


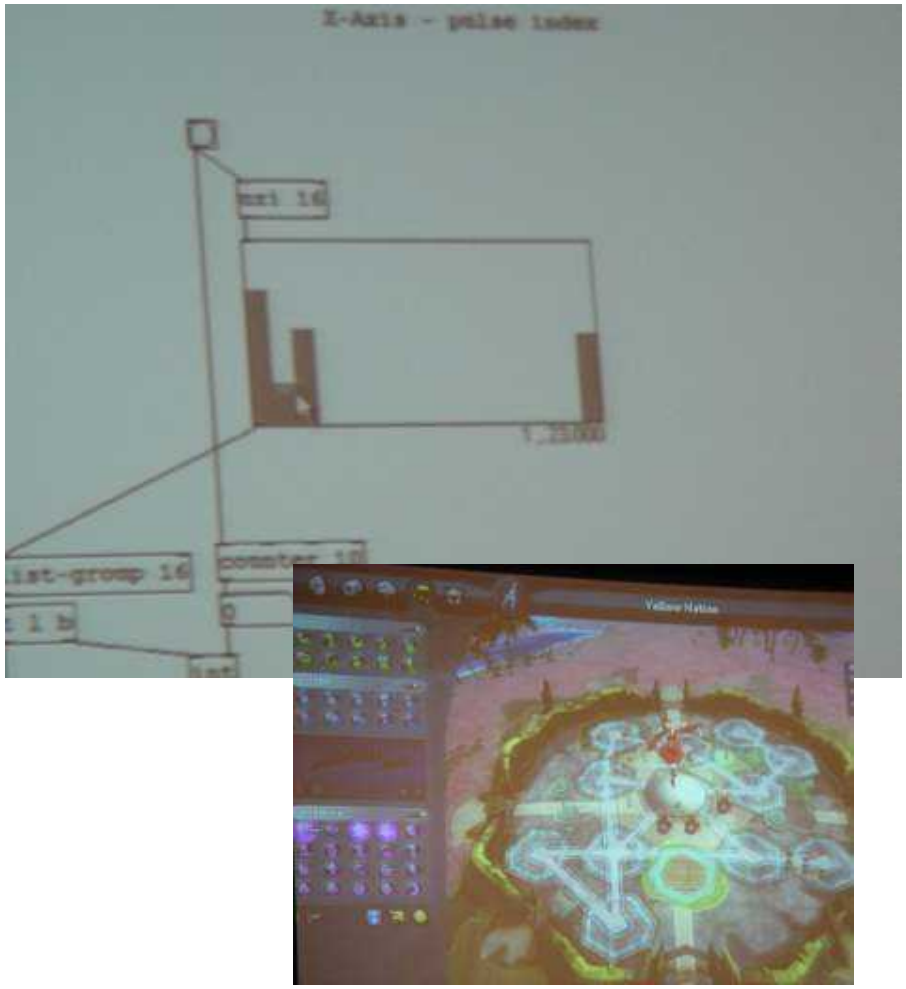
Pd とは？


0
metro 500
counter 8
0
t b r
0
sel 1 5
1
2
gate 2
メトロノーム
500msecごとにパルス発生
カウンター
トリガー
ビートのカウントを受け取る
セレクター
数値オブジェクト
ゲート

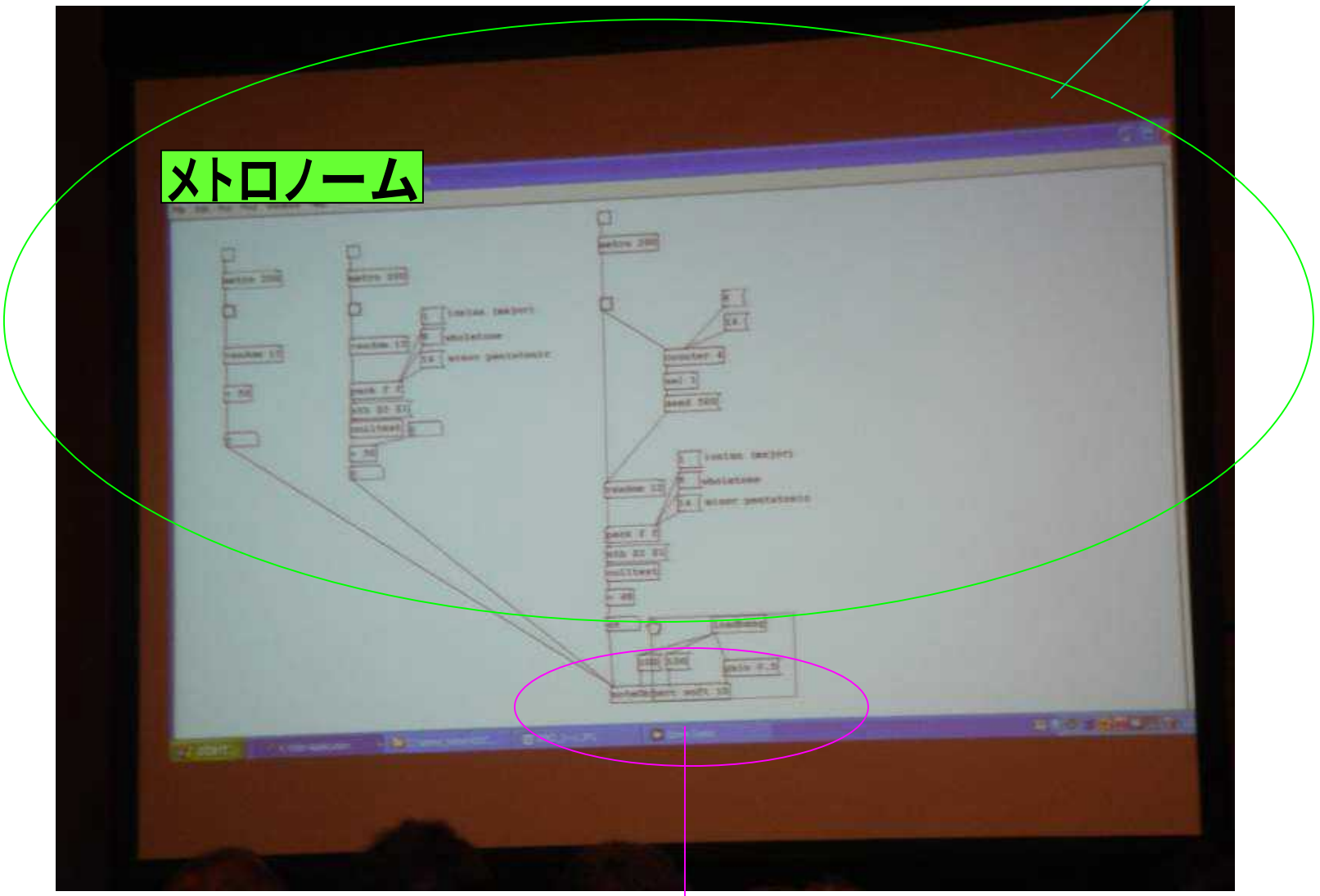

metro 500
counter 8
t b f
sel 1 5
trigger a note-->
gate 2
note choices on strong beats-->

Pd とは？

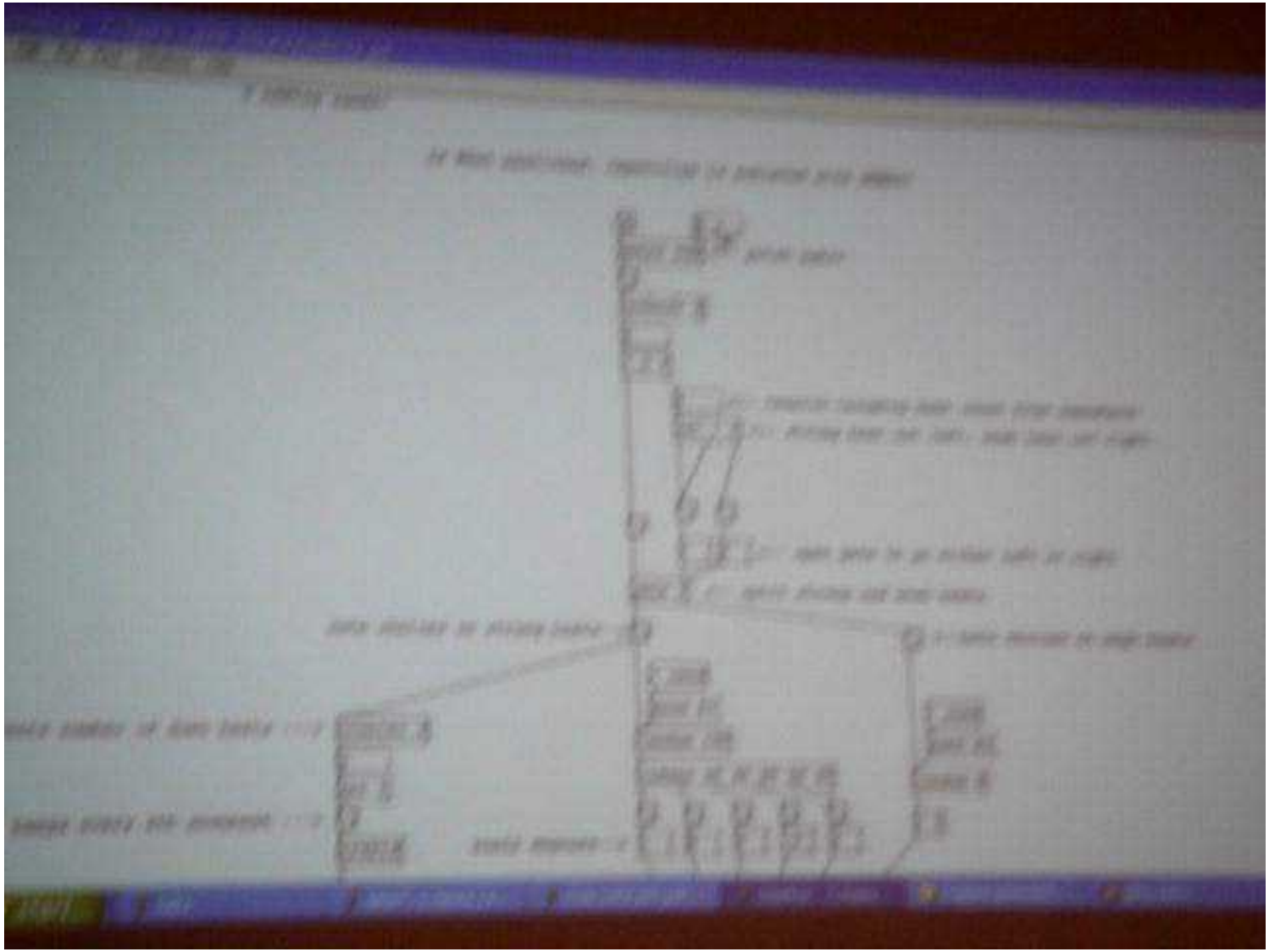




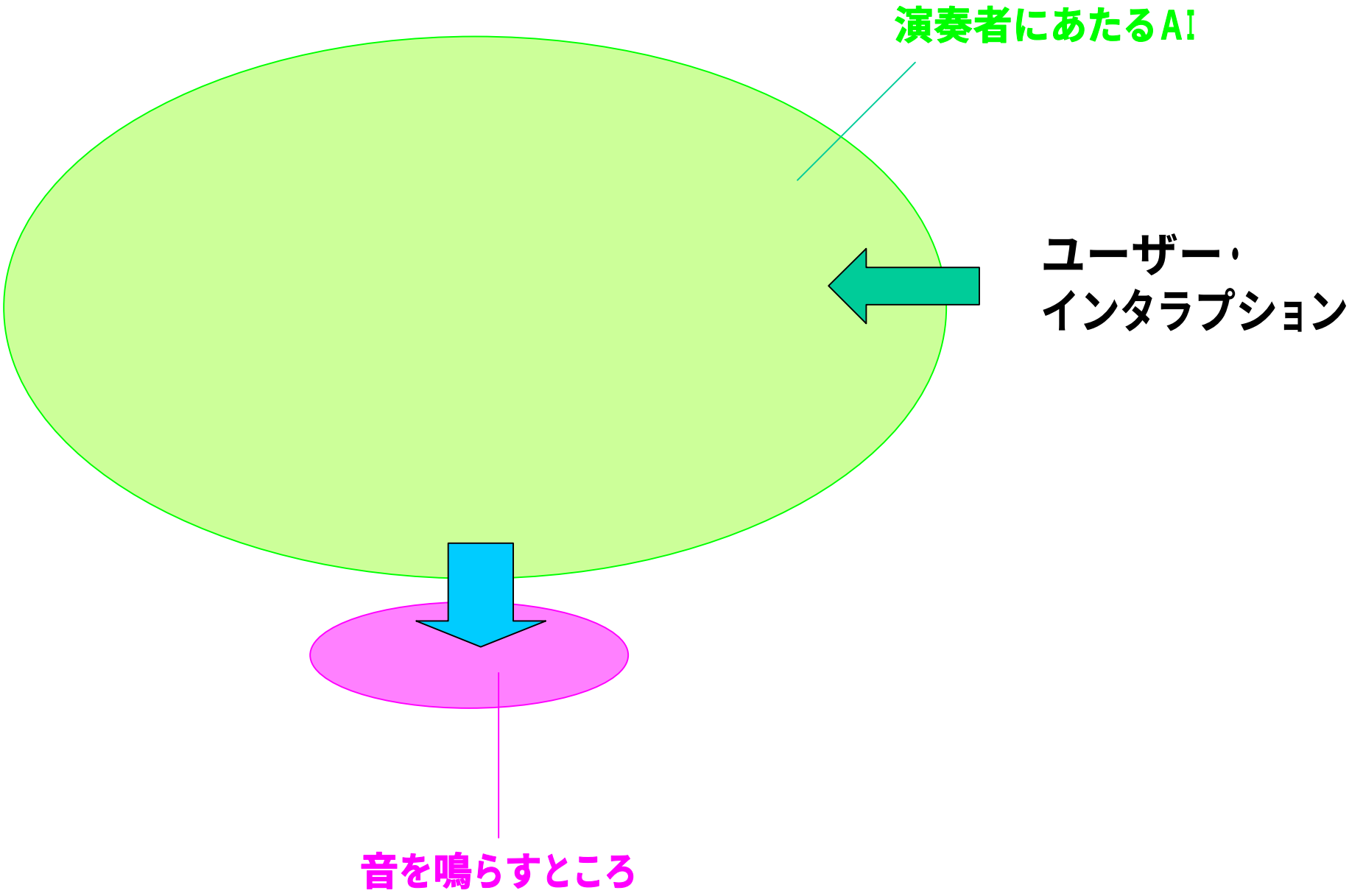
Pd とは？



EAPd



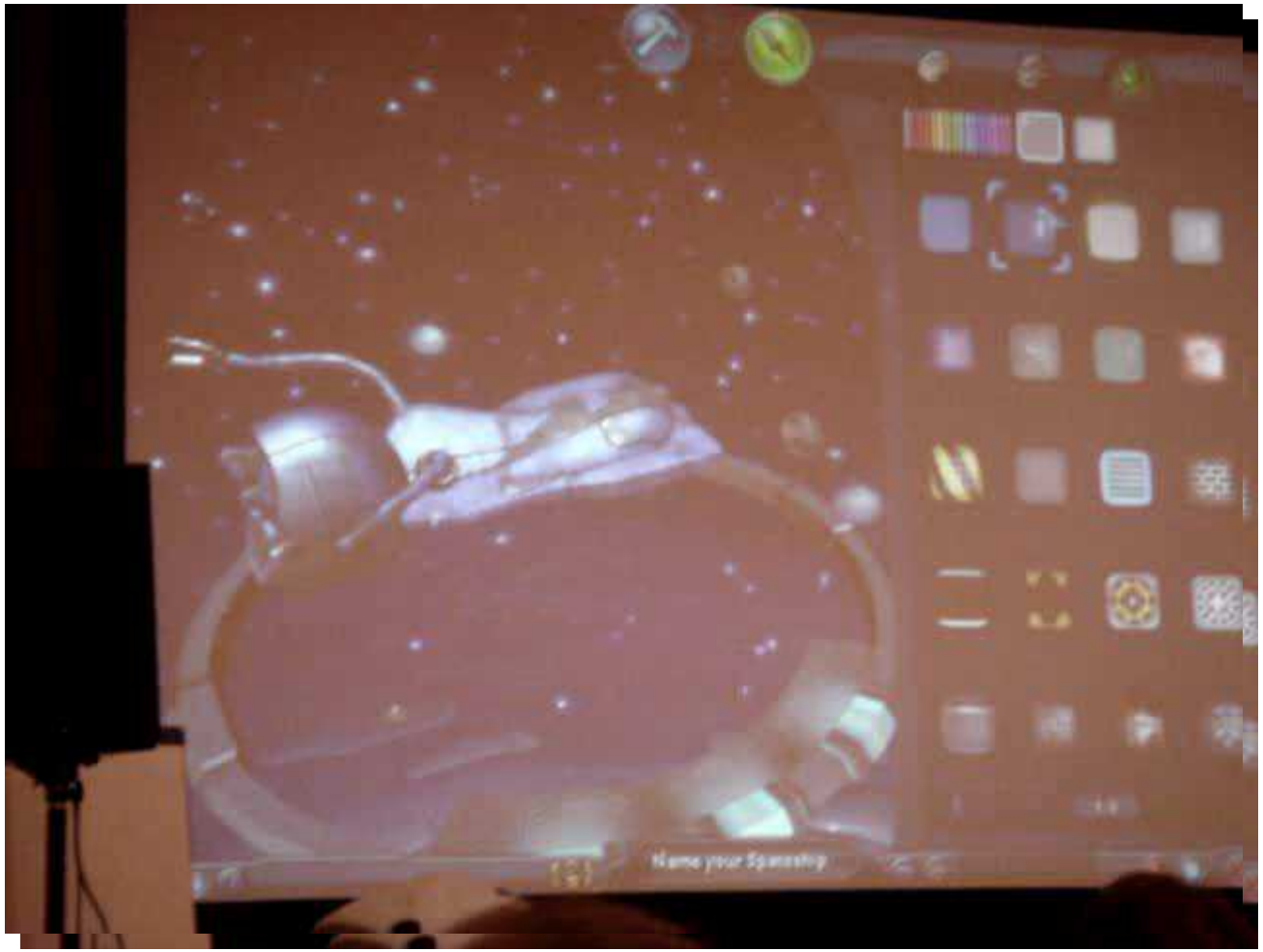
Pd とゲームサウンド



Spore における実現例①

ユーザーのマウス･アクションに応じて音が加わって行く
（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加わる）

Spore における実現例②



なんで今までなかったのだろう？　ゲームにとても馴染んだ技術

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

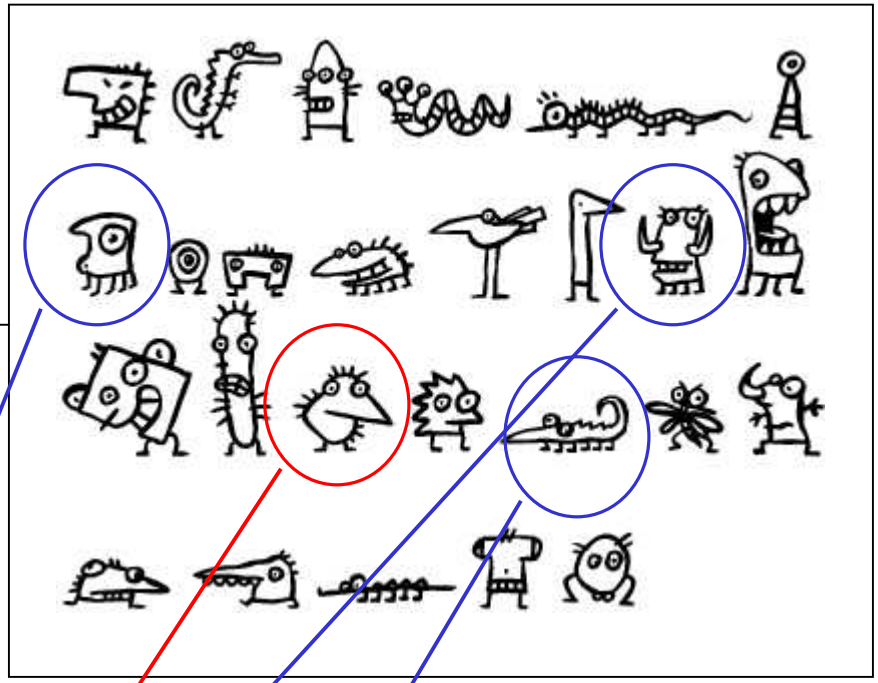
Scale	Player Creation	Technology
Galactic	Colonies	?
Teraform	Planet	地形自動生成、自動分布
Civilization	Vehicles	リグブロック、自動テクスチャ、自動アニメーション
City	Buildings	リグブロック、自動テクスチャ、自動アニメーション
Tribal	Tools	Peer AI, 繁殖
Creature	Animal	リグブロック、自動テクスチャ、自動アニメーション
Cellular	Cell	?
Molecular	Molecule	?

＋ 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

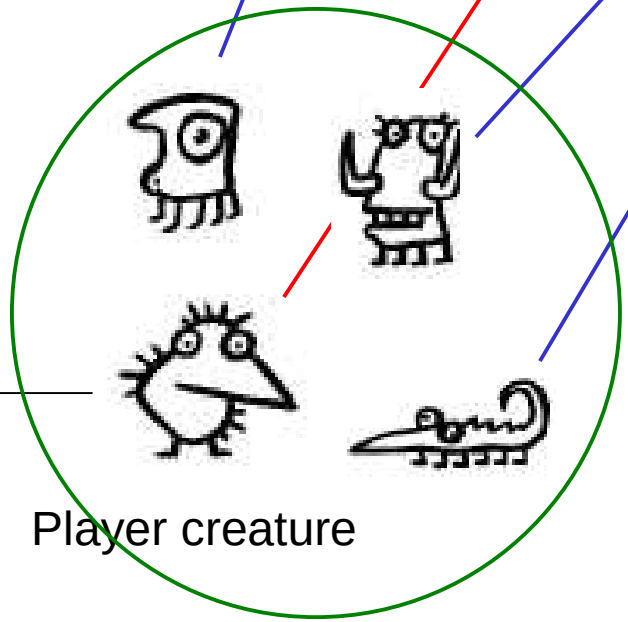
http://www.gamespot.com/pc/strategy/spore/news.html?sid=6165667

Content Pollination

Server



Player
Computer



Player creature

Ecosystem

Pollination #2

Server

Player
Computer

Buy Mode

++

Pollination #3

References

(1)**カーネギー･メロン大学の**Andrew J. Willmot**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。
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）

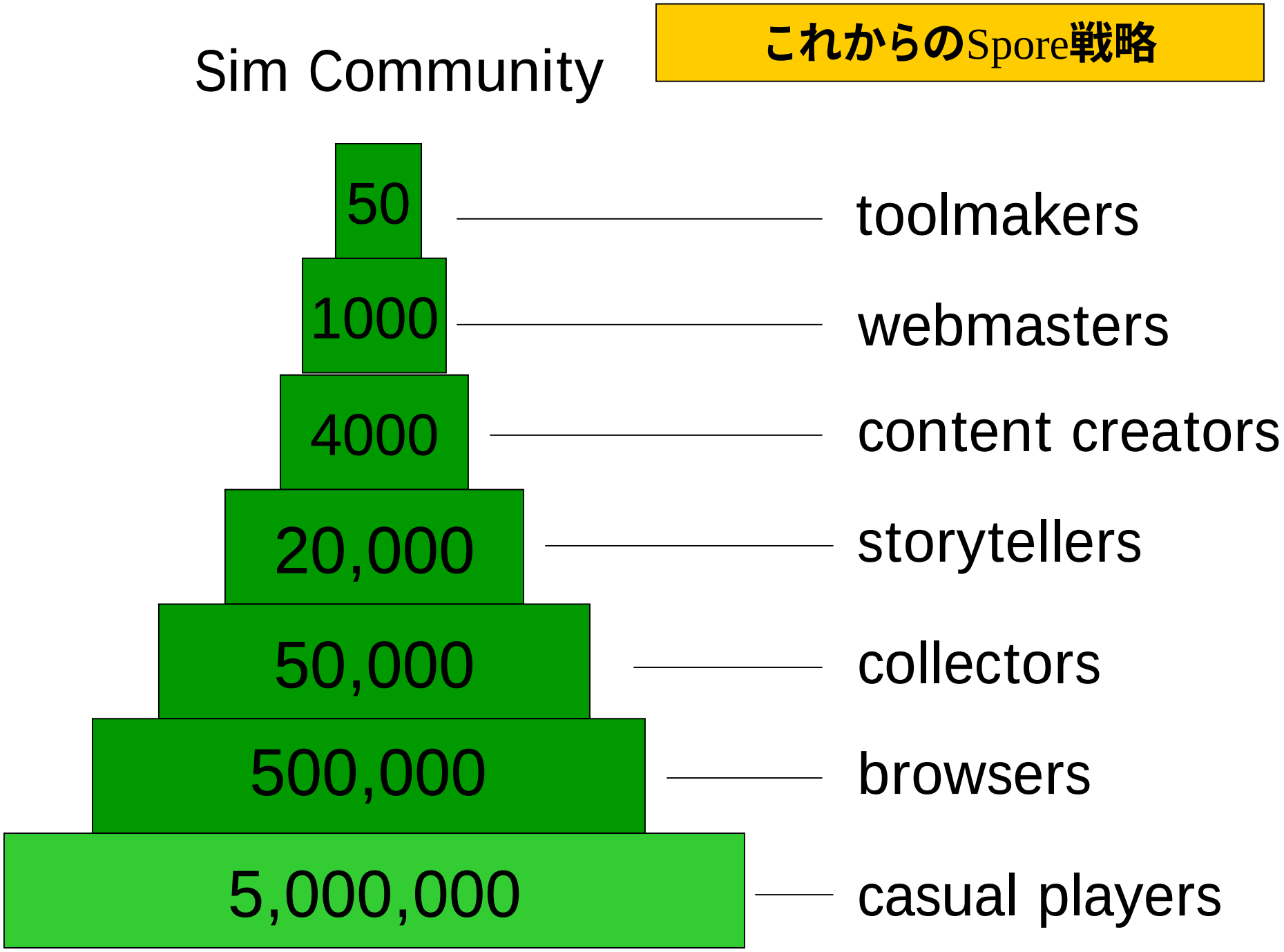
http://www.igda.jp/modules/mydownloads/

＋

カーネギー・メロン大学のAndrew J. Willmot博士のＨＰページのSIIGRAPH2007のコーナー

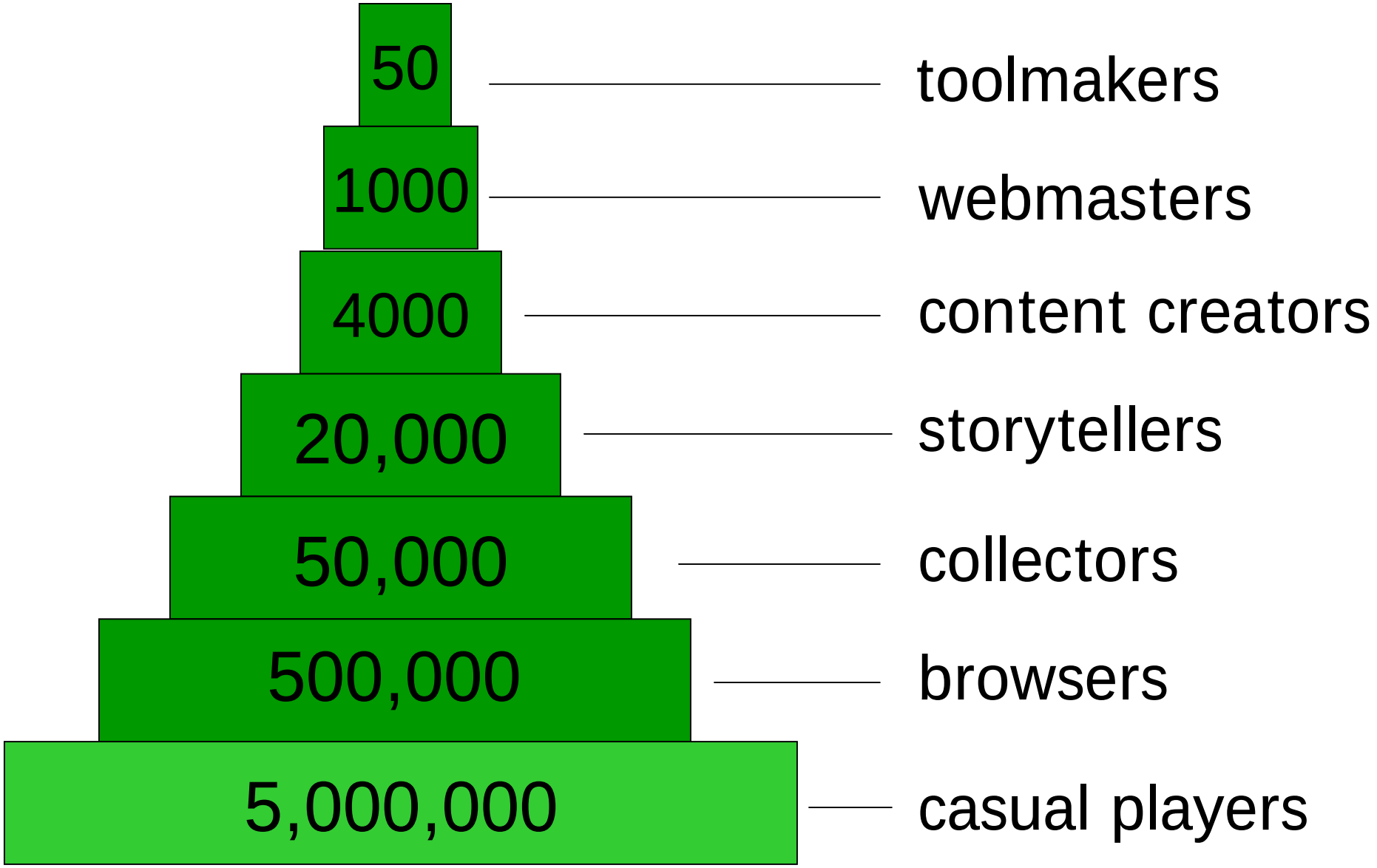
http://www.cs.cmu.edu/~ajw/s2007/

これからのSpore戦略



Spore Community

これからのSpore戦略（予想）



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、メイルへ

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、メイルかアンケートへ

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