Slab sje @1-536

—-

- セミナー
- books
- 計算機設定
- sje2009

関連サイト

- ODE公式ページ

= :Open Dynamics Engine - Home : http://www.ode.org/ode.html

```
= :ODE Wiki : http://opende.sourceforge.net/wiki/index.php/Main_Page
= :ODE API Reference : http://opende.sourceforge.net/docs/
* ODE本
=:森北出版「簡単！実践！ロボットシミュレーション : 
http://www.morikita.co.jp/shoshi/ISBN978-4-627-84691-3.html
=:「簡単！実践！ロボットシミュレーション」サポートサイト : 
http://demura.net/9ode/robotsimu
= :ODE簡易API集： http://demura.net/tutorials/api
*その他
= :Ubuntu Linux : http://www.ubuntulinux.jp/
```

//:MiniGW - Home|http://www.mingw.org/

予定とお知らせ

9/30(木) slabテーマ発表会

10/7(木) ODEイントロ、開発環境の準備、次回以降の担当 （出席：濱口、萩原、中井）

1. Automake
2. libtool
3. g++
4. freeglut3-dev

5. **ODE**
6. Env.
    - Fontsize
    - Printer Setting

10/14(木) pp.4-21 (hello.cpp, bounce.cpp) さの （出席：濱口、萩原、(中井)）

- robosimu090614.zip

10/21(木) pp.22-33 (monoBot.cpp) 濱口 + 中井発表（出席：濱口、中井、萩原(遅刻)）

10/28(木) pp.34-48 萩原 （出席：萩原、濱口）

11/4(木) pp.48-54 萩原/濱口 + LaTeX 入門その1 + 中井発表（出席：萩原、濱口、中井）

- Upgrade to Ubuntu 10.10
- Ubuntuの日本語環境

11/11(木) LaTeX 入門その2 + 中井発表（出席：中井、濱口、萩原）

- example1.tex
- example1.pdf

11/18(木) これ移行のどこかで制作案発表+中井発表

- 12/16 制作案発表&中井発表

11月中旬以降

- 各自で必要に応じて第3章以降を独習してください。自分の作品イメージを作りながら進めてください。その意味では、必ずしも全てを網羅的に勉強する必要はないはずです。
- 基本的に（制作案の発表が無い限り）数理情報演習の時間に集まる必要はありません。ただし、原則として木曜の4講時は、さのが 1-536/538 に必ず待機するようにします。わからないところや制作案の策定など、さのを使いたい場合はこの時間を使ってください。（人が来なさそうなときは5講時終了時まで在室しませんけど。）また、オフィスアワー（木と金の3講時）にも基本的に在室しています。1-536 については、いつでも(24h/365days)自由に利用してください。

制作案の作成と発表について

- ODEを使ったプログラムの制作案を **LaTeX を使って** A4 2ページ以内にまとめてください。その内容を全員の前で発表してもらいます。人数分の配布プリント（制作案）を用意しておいて下さい。およそ、以下のような内容を含むことを期待します。

1. //What// 何を作るか（目的）
2. //Why// なぜそれを作るか（動機）
3. //How// どのように作るか（方法）
4. //You// あなた自身があるか（特徴、工夫）

- 発表会の開催は、各々が作成案をまとめた時点で数理情報演習の時間に行います。発表準備が整った人は、前の週末までに「来週、制作案を発表するので集合。」とかいうメイルを **sje@slab.math.ryukoku.ac.jp** に流してください。このメイルが流れた次回の数理情報演習の時間は「全員集合」です。メイルは @mail.ryukoku.ac.jp のアドレスに送られるので、毎週末メイルをチェックして発表会の有無を確認してください。
- 制作案は教科書のサンプルファイルなどを参考にして構いませんが、必ず自分のアイデアを盛り込んでください。
- 制作案の発表を終えた人は、自分のプログラムの作成に取りかかってください。

1/13(木) 最終作品の発表会 + 中井発表&レポート提出

- 制作したプログラム作品のお披露目会です。
- 完成した作品が制作案の内容から変わっていたとしても構わないです。

ODE 0.11 のインストール

- 0.11.1 も同様。
- ~~OSX 10.6 snow leopard 用の patch がありました（ここ）。; ちょっと変だったのを ode-0.11.1 用に修正したものが以下です//patch -p1 < ode-0.11.1-OSX10.6-GLUT.patc// してください。~~
  - ode-0.11.1-osx10.6-glut.patch
- このへんから、開発版スナップショットを持ってくるのが良さそうです。上記他、幾つか採用された patch が含まれています。

1. ここ から ode-0.11.tar.bz2 をダウンロード。
2. ホームディレクトリなどで展開。

```
tar xvjf ode-0.11.tar.bz2
```

3. ディレクトリ ode-0.11 に移動□INSTALL.txt を眺める//2. Building with Autotools (Linux, OS X, ext.// に従ってコンパイルとインストール。

```
% sh autogen.sh
% ./configure --enable-double-precision --enable-debug
% make
% sudo make install
```

   - 0.11 はリリースビルドだと衝突判定に不具合があるらしい。デバッグモードでビルド□(–enable-debug)
   - MacOSX なら、描画にCarbonが利用される□X11を利用したければ以下のように指定する。

```
% ./configure --enable-double-precision --enable-debug --with-drawstuff=X11
```

1. 適当なデモで動作を確認。

```
% ./ode/demo/demo_<なんちゃら>
```

2. 3D描画ライブラリ drawstuff のファイルをコピーしておく。

```
% sudo mkdir -p /usr/local/include/drawstuff
% sudo cp include/drawstuff/*.h /usr/local/include/drawstuff/.
% sudo cp drawstuff/src/libdrawstuff.la /usr/local/lib/.
% sudo cp drawstuff/src/.libs/libdrawstuff.a /usr/local/lib/.
```

3. drawstuff で利用するテキスチャーも適当な場所へコピーしておく。

```
% sudo mkdir -p /usr/local/share/drawstuff
% sudo cp -r drawstuff/textures /usr/local/share/drawstuff/.
```

    - あとで、テキスチャー画像を追加したい場合は、ユーザー領域に置いておくと追加変更が容易。

作業環境を作る

1. 適当な作業ディレクトリを作成する□ODEのデモを例にして、この作業ディレクトリでODEを使ったプログラムをコンパイルできるようにする。

```
% mkdir ~/myODE
```

2. デモのソースを適当にコピー。

```
% cp ~/ode-0.11/ode/demo/demo_buggy.cpp .
```

3. テキスチャの場所を指定しているインクルードファイルをコピーして編集。

```
% cp ~/ode-0.11/ode/demo/texturepath.h .
```

    - インストールの手順 6. でテキスチャを置いたディレクトリを指定しとく。

```
#define DRAWSTUFF_TEXTURE_PATH "/usr/local/share/drawstuff/textures"
```

- **0.11.1 では -texturepath での指定も可（dsSimulationLoop()の引数指定）。**

1. 以下の内容を Makefile として作業ディレクトリに。TARGET は適宜変更。

- Linux 用です□OSX の場合はコメントに従って修正。

CC = g++ -I -O2 -Wall -g TARGET = demo_buggy OBJS = $(TARGET).o SOURCE = $(TARGET).cpp HEADER = texturepath.h LIBS = -L/usr/X11R6/lib INDS = -I. -I/usr/X11R6/include

ifeq ($(shell uname),Linux) OPTS = -lICE -lSM -lm -ldrawstuff -lX11 -lGL -lGLU -lpthread endif

ifeq ($(shell uname),Darwin) XVER = $(shell uname -r | sed -e “s/\.[0-9]*\.[0-9]//”)

ifeq ($(XVER),9) OPTS = -lm -ldrawstuff -lX11 -framework OpenGL -framework Carbon -framework AGL -lpthread endif

ifeq ($(XVER),10) OPTS = -lm -ldrawstuff -lX11 -framework OpenGL -framework GLUT -lpthread endif

endif

ODE_LIBS = $(shell ode-config –libs) ODE_INCLUDE = $(shell ode-config –cflags) -DdTRIMESH_ENABLED

all:$(TARGET)

$(TARGET):$(OBJS) $(HEADER)

```
    $(CC) -o $@ $(OBJS) $(LIBS) $(INDS) $(OPTS) $(ODE_LIBS)
```

$(OBJS):$(SOURCE) $(HEADER)

```
    $(CC) -c $(SOURCE) $(LIBS) $(INDS) $(ODE_INCLUDE)
```

.PHONY: clean clean:

```
    rm $(TARGET) $(OBJS) *.*~ *~
```

}}

1. 作業ディレクトリで make して、デモプログラムが動けば準備完了。

RIGHT:今日□&counter(today); 昨日□&counter(yesterday); 累計□&counter(total);