

JVRC

Choreonoidハンズオン

(株)MIDアカデミックプロモーションズ

松坂要佐

# Choreonoid

- 振り付けソフトとして初期の開発が始まる
- フレームワークとしての発展
  - プラグイン機構
  - Boostを使ったモダンなC++コード
  - 軽快なシーングラフ
- ロボット開発に必須の各種機能
  - 逆運動学
  - 物理シミュレーション
  - センサシミュレーション

# Choreonoidの類似ソフト

- OpenHRP
  - 物理シミュレーション、センサシミュレーション
  - Java (Eclipse)によるクロスプラットフォームUI
  - CORBAサービスによる各機能の実装
- Gazebo + rviz
  - ROS標準の組み合わせ
  - 物理シミュレーション、センサシミュレーション(やや重い)
  - 各種ビジュアライゼーション機能
  - ROSスタックと連携させることで逆運動学も使える
- Choreonoidの特徴
  - ハイパフォーマンスかつ軽量
  - 単独でシミュレーション+逆運動学

セットアップ

# インストール

- 必要環境

- Ubuntu Linux (各種バージョンに対応)

- 以下のコマンドを入力

```
$ sudo add-apt-repository ppa:hrg/daily
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install choreonoid libcnoid-dev  
openrtm-aist openrtm-aist-dev doxygen
```

```
$ sudo apt-get install git
```

# サンプルファイルの取得

- プログラムサンプル

- 以下のコマンドを入力

```
$ git clone https://github.com/jvrc/samples
```

- モデルファイル

- 以下のURLからJVRC-1をダウンロード

<https://jvrc.org/download.html>

- githubにも公開されているので以下のコマンドでもよい

```
$ git clone https://github.com/jvrc/model
```

# ハンズオン チュートリアル

<http://jvrc.github.io/tutorials/html-ja/>

# ハンズオン

- サンプル1
  - モデルのロード
  - 物理シミュレーションの適用
- サンプル2
  - RTCの接続
- サンプル3
  - モーションシーケンスの作成
  - モーションシーケンスに基づく制御
- サンプル4
  - センサデータの取得



# Choreonoid利用のポイント

- モデルのロード後にはアイテムのチェックボックスをオンにする
  - 個人的にはデフォルトでオンになってほしい、、、。
- アイテムには親子関係がある
  - 「モデル」は「ワールド」の下に置く
  - 「シミュレータ」は「ワールド」の下に置く
  - 「RTC」は「接続したいモデル」の下に置く

→親子関係を守ることで機能が有効になる

サンプル通りには動くが  
正直、先は長い

、、、なので

# JVRC必勝法

P.S. 私もエントリーしました

# hrpsys-base

- <https://github.com/fkanehiro/hrpsys-base>
- 鍛えられたコンポーネント集
  - 制御系
  - センサー系
  - 表示系
- 充実したコミッター勢
  - <https://github.com/fkanehiro/hrpsys-base/graphs/contributors>
- 実質的なDRC日本陣営共通レポジトリ

# hrpsys-base

- インストール方法

- PPAが登録してあれば以下のコマンドを打つだけ

```
$ sudo apt-get install hrpsys-base
```

- hrpsys-baseは更新頻度が高い(毎日コミットがある)の  
ですが、PPAはdaily buildなので日々自動で最新パッ  
ケージが降ってきます

- お勧めの勉強方法

- issueやpull requestのディスカッションを見るだけでもか  
なり勉強になります

→Githubのアカウントを作ってプロジェクトをウォッチ設定

# hrpsys-baseのコンポーネント

- SequencePlayer
    - キーフレームを補完して再生
    - 逆運動学ベースのモーション生成
  - AutoBalancer
    - IMUセンサを使ったバランス制御
    - 二足歩行(平地、階段)のサンプルあり
- <https://github.com/fkanehiro/hrpsys-base/tree/master/sample/SampleRobot>
- RTMを使っていればそのまま繋がる  
→使わない手はない

# GUIで操作するのはだるい

- ChoreonoidはPythonで操作できます

```
worldItem = WorldItem()
RootItem.instance().addChildItem(worldItem)
timeBar = TimeBar.instance()
timeBar.setFrameRate(500)
timeBar.setTimeRange(0.0, 15.0)
timeBar.setFillLevelSync(False)
robotItem = loadBodyItem(shareDirectory() + "/model/SR1/SR1.yaml")
```

- 各検証用シミュレーションの自動化など

# ルールを深読みする

- 物理シミュレーションについて
  - 精度の高いChoreonoid用シミュレーションエンジンが後日公開されそれが本選で使われるらしい
  - Choreonoid以外のシミュレータは使わない方が良い？
- ROSの使用について
  - Choreonoidの使用は必須だが、RTMが必須であるとは書いてない
  - 海外からのエントリーを推奨する記述(招待枠あり?)
  - 使っても良い前提で心構えした方がよさそう
  - e.g. ブリッジを作って環境認識はROSに投げるなど



# 自由討論