

所在情報

= 居室: 1号館5階 538 & ときどき 536 (学生実験室)
= email: [[sano@math.ryukoku.ac.jp]]
= オフィスアワー: 木曜日3講時/6講時 (会議等で不在のときもあるので事前にメールでも下さい。)
= オフィスアワー以外の時間でも構いません。説明会以前の面会を歓迎します。:

担当教員: 佐野彰

表のテーマ1: 物理計算エンジンを使ってロボットシミュレータをつくる

フリーの物理計算エンジン **ODE(Open Dynamics Engine)** を使ってロボットシミュレータ (でなくてもいいけど) をつくります。

= 物理計算エンジン (ODE) とはなんですか? :

計算機上の仮想空間で、物理世界における物体の動き (落下 回転 衝突など) を計算してくれるプログラムモジュール (C言語ライブラリ) です。3次元空間上で、個々の物体の動き (動力学計算) とそれらの干渉 (衝突検出計算) を計算します。計算機上の物理シミュレータを作成する際、これらの計算を物理エンジンに任せることで、コントロール部などのプログラムに注力できます。

= 物体のコントロールはどうするのですか? :

個々の物体の接合部にモータ設定し、これらのモータをキーボード、マウスを通して、あるいはプログラムで直接駆動することでシミュレータ空間のロボット筐体などをコントロールします。物体に直に力を与えたり、摩擦係数などをコントロールすることもできます。

= 利用するプログラミング言語は何ですか? :

CあるいはC++を利用します。

表のテーマ2: iOS アプリをつくる

iPhone や iPad 上で動くiOSアプリケーションをつくりたい。

= これまでの演習でやっていたプログラミングと何が違いますか? :

Xcodeというプログラミングの統合環境を利用します。ボタンなどの画面配置をGUIで設計することもできます。

= 利用するプログラミング言語は何ですか? :

Swift を利用します。2014年にAppleが発表した新しい言語です。オープンソース化してSwift2になっています。

表のテーマに共通する話題

= 複数テーマの位置づけは？ :

一つ以上のテーマを選択して下さい。みなさんが選択したテーマの分布に応じて、順番に、あるいは同時並行的に進めようと思います。したがって配属を希望する場合は、**事前にどのテーマに取り組みたいのかを必ずさのに伝えて下さい。**

= どんなふうに演習を進めていきますか？ :

まずは、各自で ODE¹あるいは Xcode のプログラム環境を整備するところから始めます。前半は教科書などに沿って輪講形式（順番に担当箇所をみんなに説明する）で ODE や Xcode の基礎を勉強します。ただし、時間的にも基本的な内容しかできないので、作りたいものに応じて各自で調べたり勉強したりが必要です。最終的には、それぞれがシミュレータやiOSアプリなど、自分の作品を提案 設計して全員の前で発表します。合間に LaTeX をやったりしています。

裏のテーマ

1. 持ち込みテーマ

独自のテーマ持ち込みを歓迎します。勝手に自由にやりたいという方は相談してみてください。下記URLにある過去の演習テーマなども参考になるかもしれません。

2. インテリジェンスを創る

インテリジェントなシステムを創ることに興味がある人は是非。人間の認知過程（とくに、学習や発達）に興味のある人は楽しめるかもしれません。

現在、お掃除ロボットRoombaをプログラムでコントロールしたいという人も募集しています。

連絡先など

- %配属希望者は必ず面会に来て「希望テーマをさのに伝えて」ください。%希望理由なども尋ねます。希望しなくても来室は歓迎します。
- 佐野は 1-538 に居ることが多いです。電子メール(sano@math.ryukoku.ac.jp)でアポイントメントを取ると確実です。
- これらの情報 + α が以下のURLの **Slab**数理情報演習配属2015 にあります。
<http://www-slab.math.ryukoku.ac.jp/>

過去の説明会資料

- [2014](#)
- [2013](#)
- [2012](#)
- [2011](#)

- [2010](#)
- [2009](#)
- [2008](#)
- [slab数理情報演習配属2007](#)
- [slab数理情報演習配属2006](#)
- [sje2004.pdf](#)
- [sje2003.pdf](#)
- [sje2002.pdf](#)
- [sje2001.pdf](#)
- [sje2000.pdf](#)

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

From:

<https://www-slab.math.ryukoku.ac.jp/> - **www-slab.math**

Permanent link:

<https://www-slab.math.ryukoku.ac.jp/lecture/sje/guide/2015?rev=1522212886>



Last update: **2018/03/28 13:54**