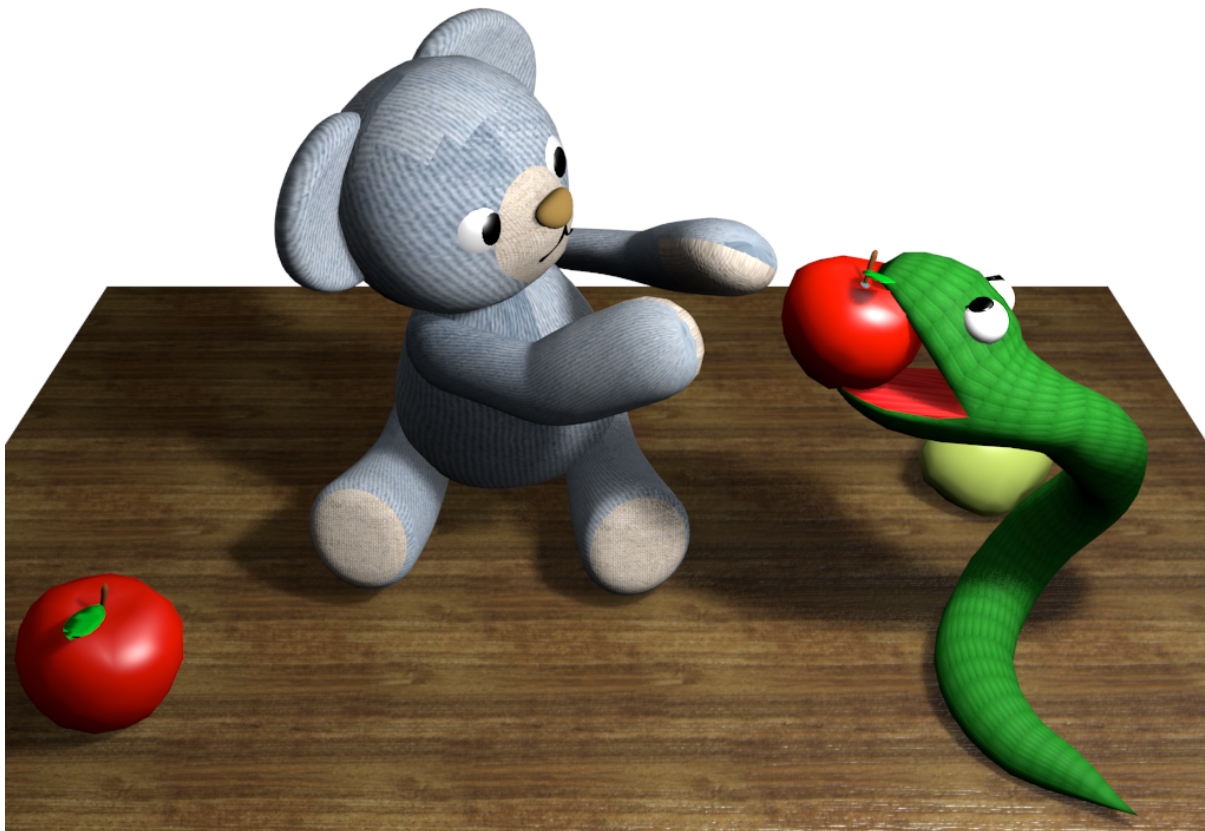


# SprBlender TUTORIAL

---





# はじめに

SprBlender は、Blender 上で Springhead2 による物理シミュレーションを利用可能にする Python モジュールです。

SprBlender を用いると、Springhead2 が持つ多様な関節・IK 制御・キャラクタ動作生成等の機能を利用した物理シーンを、Blender 上で構築・編集・実行することができるようになります。

シミュレーションはリアルタイムで動作し、動作中のシーンにマウスやヒューマンインタフェースを用いてインタラクションすることができるため、アニメーション製作に限らずリアルタイムデモの作成やゲームエンジンとしての利用も可能です。

このチュートリアルでは、SprBlender の使用法を具体例に基づいて基本から詳細まで解説していきます。

本文書は

- Blender 2.65a
- SprBlender 2.0pre

に基づいています。



# 目次

はじめに	i
第 1 章 インストール	1
1.1 必要なもの	1
1.2 ダウンロード	1
1.3 インストール	2
1.4 サンプルを実行してみる	4
1.5 (開発者向け) Subversion から最新版を利用するには	5
第 2 章 Blender の基本操作	9
2.1 ファイル操作	10
2.2 画面分割	10
2.3 3D View の操作	10
2.4 物体の操作	10
2.5 “Outliner” ウィンドウ	10
2.6 “Properties” ウィンドウ	10
第 3 章 剛体で遊ぶ	11
3.1 物体を剛体にする	11
第 4 章 キーフレーム録画	13
4.1 録画してみる	13
第 5 章 関節と IK	15
5.1 関節を作ってみる	15
第 6 章 スキンメッシュ	17
6.1 Springhead-Armature-Skinmesh 連動の仕組み	17
第 7 章 スクリプティング	19
7.1 スクリプトを置く場所	19

---

7.2	SprBlender の内部構造 . . . . .	19
第 8 章	クリーチャ	21
8.1	クリーチャの構成 . . . . .	21
第 9 章	UI リファレンス	23
9.1	Scene パネル . . . . .	23
第 10 章	トラブルシューティング	25
10.1	概要 . . . . .	25
10.2	プログラムのビルド . . . . .	25
10.3	物理シミュレーション関連 . . . . .	26
10.4	ファイル関連 . . . . .	26
10.5	その他のトラブル . . . . .	26

## 第1章

# インストール

---

SprBlender の入手・インストールから、起動・サンプルの実行までを解説します。

SprBlender は Blender 上で動作する Python モジュールです。そのため、Blender のインストールフォルダ内の所定の位置に配置する必要があります。今のところ、インストーラは用意されていません（将来的には用意するべきだと考えています）。

### 1.1 必要なもの

SprBlender の動作に必要なものは以下のとおりです。

1. Windows XP 以上 (7 を推奨, 32bit/64bit は問わない)
2. Blender 2.5 以上 (2.65a を推奨)
3. SprBlender
4. Springhead DLL for SprBlender(Spr.pyd)
5. Springhead が依存するライブラリの DLL(glut 等)

SprBlender zip パッケージをダウンロードすると、3～5 をまとめて入手することができます。

### 1.2 ダウンロード

Blender の公式配布ページは

<http://www.blender.org/download/get-blender/>

です。

2013 年 2 月末現在、Blender の最新版は 2.66 となっています。そのため、2.65a を入手するには旧バージョン配布ページ (<http://download.blender.org/release/Blender2.65/>) へ行って

```
blender-2.65a-windows64.exe  
(* 32bit windows の場合は blender-2.65a-windows32.exe)
```

をダウンロードします<sup>\*1</sup>。

---

<sup>\*1</sup> 2.66 での動作テストを行っていないため 2.65 をおすすめしていますが、動作する可能性は十分あります。テストしたらドキュメ

次に, SprBlender をダウンロードします。配布ページ

[http://springhead.info/wiki/SprBlender\\_download](http://springhead.info/wiki/SprBlender_download)

へ行き, 最新版の zip パッケージをダウンロードします。

## 1.3 インストール

### Blender のインストール

インストーラ版の Blender は, ダウンロードした exe ファイルを実行するとインストールが始まります。指示に従い, インストールを行なってください。

Blender の標準的なインストールフォルダは

```
C:\Program Files\Blender Foundation\Blender
```

です。本文書では, 以後 Blender をこのフォルダにインストールしたものとして話を進めます (異なるフォルダにインストールした場合は適宜読み替えてください)。

### SprBlender のインストール

次に, SprBlender のインストールを行います。

ダウンロードした SprBlender zip パッケージを展開すると, Fig 1.1 のようなフォルダ構成になります。

#### 1. SprBlender 本体を Blender インストールフォルダの modules にコピーします。

Blender がインストールされたフォルダ内に

```
C:\Program Files\Blender Foundation\Blender\2.65\scripts\modules
```

というフォルダがあります。ここに, 解凍してできた `scripts\modules\spb` フォルダをそのままコピーします (Blender のインストールフォルダ内にも `scripts\modules\spb` フォルダ ができることになります)。

#### 2. Spr.pyd も同様に modules フォルダにコピーします。

この時, コピー元の `Spr.pyd` は Blender のバージョンと 32bit/64bit の違いに応じて適切なフォルダから選択してください。Blender2.65 以上の場合は `py33` から, Blender2.64 以下の場合は `py32` から選びます<sup>\*2</sup>。32bit/64bit の違いはインストールしてある Blender に合わせてください (例えば, もしも

ントも書き換える予定です。なお, テストしてくれる人柱を募集中。

<sup>\*2</sup> これは, Blender に組み込まれている Python のバージョンが 2.65 から Python3.3 に上がったためです。Springhead にも Python が組み込まれており, 両方のバージョンを合わせる必要があります。



```

+- samples/
| +- blend/ ..... サンプルの.blend ファイル
| +- textures/
+- scripts/
| +- modules/
|   +- spb/ ..... SprBlender 本体
+- spbapi_cpp/
| +- bin/ ..... Springhead DLL 格納フォルダ
|   +- py32/
|     | +- win32/
|     |   | +- Spr.pyd ..... Blender2.64 以下, 32bit 用
|     |   +- win64/
|     |     +- Spr.pyd ..... Blender2.64 以下, 64bit 用
|   +- py33/
|     +- win32/
|       | +- Spr.pyd ..... Blender2.65 以上, 32bit 用
|       +- win64/
|         +- Spr.pyd ..... Blender2.65 以上, 64bit 用
|
+- spr_depend/
  +- win32/ ..... Springhead 依存ライブラリの DLL (32bit 用)
  +- win64/ ..... Springhead 依存ライブラリの DLL (64bit 用)

```

Fig. 1.1 SprBlender パッケージ展開後のフォルダ構成

64bit Windows に 32bit 版 Blender をインストールした場合は 32bit を選んでください).  
コピー先は同じく,

C:\Program Files\Blender Foundation\Blender\2.65\scripts\modules

です.

### 3. spr\_depend フォルダを移動し、環境変数 PATH に登録します.

解凍してできた spr\_depend フォルダをどこか適切な場所<sup>\*3</sup>に移動します. ここでは,

C:\Program Files\Blender Foundation\Blender\spr\_depend

に置いてみます.

次に, スタートメニューから

- (a) 「コンピューター」を右クリック
- (b) 「プロパティ」を選択
- (c) 「システムの詳細設定」をクリック
- (d) 「環境変数」ボタンをクリック

<sup>\*3</sup> このフォルダを移動すると環境変数を設定し直すことになりますので場所選びは慎重に.

で環境変数ダイアログを出します。

ユーザー環境変数の一覧に「Path」変数があればそれを選択して「編集」を、なければ「新規」をクリックします。新規の場合は変数名を「Path」としてください。  
変数値に

C:\Program Files\Blender Foundation\Blender\spr\_depend\win64

(\* 32bit の場合は \win32)

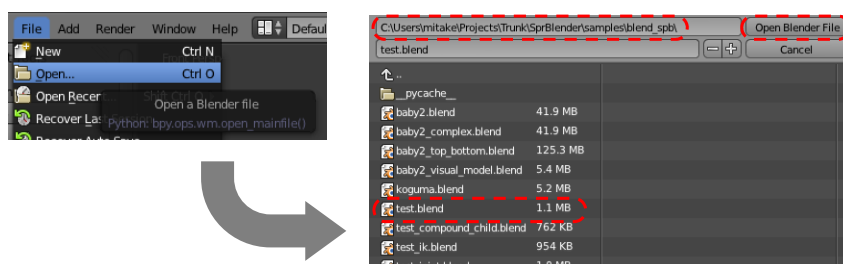
を追加します。新規の場合はそのまま入力し、編集の場合は既存の内容の最後にセミコロンをつけてから上記の内容を入力します。

OK を押すと環境変数の登録が完了します。環境変数を設定したあとは **PC を再起動** してください。

## 1.4 サンプルを実行してみる

SprBlender にはいくつかのサンプルが付属しています。ここでは基本的なサンプルである test.blend を実行してみることにします。

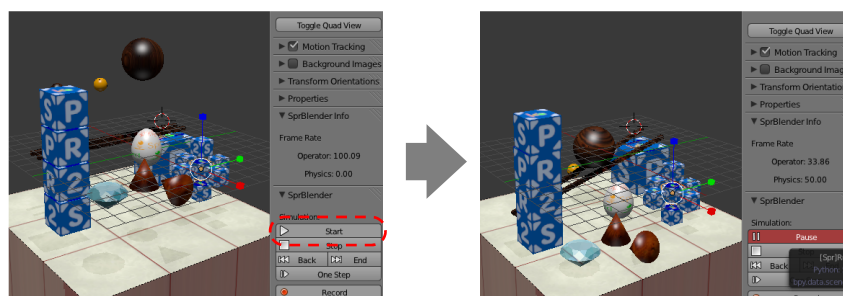
Blender を起動し、File メニューの Open を選ぶとファイルダイアログが表示されます。



SprBlender zip パッケージを解凍したものの中に samples というフォルダが含まれているはずです。samples フォルダへのパスをファイルダイアログのフォルダパスのボックスに入力します。

出てきたファイル一覧から test.blend を選択して Open Blender File をクリックするとファイルが開きます。

test.blend がロードされると Fig 1.2 のような画面になります。画面中央下寄りにある、右三角の付いた「Start」ボタンを押すことで、SprBlender のシミュレーションを開始することができます。



シミュレーションの停止は「Pause」または「Stop」です（これらの違いについては第3章で詳しく解説し

ます).

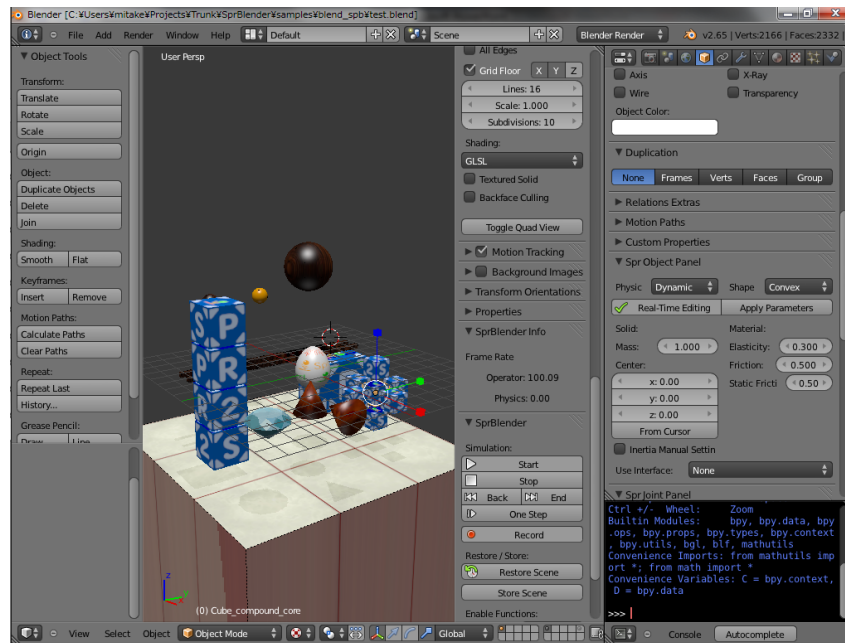


Fig.1.2 test.blend をロードしたところ

## 1.5 (開発者向け) Subversion から最新版を利用するには

### Subversion レポジトリからのチェックアウト

以下の URL で SprBlender の Subversion レポジトリにアクセスできます。

- SSH 経由 (要アカウント申請)  
`svn+ssh://springhead.info/home/svn/spr2/SprBlender/trunk`
- HTTP 経由 (コミット不可)  
`http://springhead.info/svn/spr2/SprBlender/trunk/`

http 経由でのアクセスはアカウント登録が要りませんが、コミットすることができません。最新版を使用した  
いだけの場合に利用してください。

TortoiseSVN 等を使ってチェックアウトしてください。ここでは

C:\users\[ユーザ名]\SprBlender

にチェックアウトしたとします。

## Springhead の依存 DLL のチェックアウト

SprBlender の動作には、Springhead に含まれている依存ライブラリの DLL (glut 等) が必要です。これはパッケージ版には含まれていますが、Subversion レポジトリには含まれていないため Springhead の Subversion レポジトリから別途チェックアウトする必要があります。

以下の URL で Springhead の Subversion レポジトリ内にアクセスできます。

- SSH 経由 (要アカウント申請)

`svn+ssh://springhead.info/home/svn/spr2/Springhead2/trunk/bin`

- HTTP 経由 (コミット不可)

`http://springhead.info/svn/spr2/Springhead2/trunk/bin/`

SprBlender と同じく、http 経由でのアクセスはアカウント登録が要りませんが、コミットすることができません。

ここでは、

`C:\users\(ユーザ名)\Springhead2\bin`

にチェックアウトしたとします。

### tips

SprBlender の Subversion レポジトリには、Spr.pyd のソースコードも含まれています。Spr.pyd をビルドする場合、Springhead2 のソースツリー全体が必要となります。これは

- `http://springhead.info/svn/spr2/Springhead2/trunk`

から取得することができます。

なお、Spr.pyd のビルドのために Springhead2 をチェックアウトする場合は、SprBlender と同じ場所に置く必要があります。

## シンボリックリンクによるインストール

SprBlender の開発は Subversion レポジトリで進められており、レポジトリにアクセスすれば開発者向けの最新版が利用可能です (当然ですが動作の保証はありません)。

通常は spb フォルダを Blender のインストールフォルダにコピーして使いますが、Subversion からチェックアウトして開発を行う場合、ワーキングコピーを Program Files 内に置くのは何かと面倒です。

ワーキングコピーをユーザーフォルダに置いたまま、編集や svn up を行った時に (再度 Blender フォルダにコピーすることなく) Blender の動作に反映させるには、**シンボリックリンク**を利用すると便利です。

```
- Blender/2.65/                - C:/users/(ユーザ名)/SprBlender
+- scripts/                    +- scripts/
| +- modules/                  | +- modules/
|   +- [spb] <-----[Link]--- | -- +- spb/
```

```

+- spbapi_cpp/          +- spbapi_cpp/
+- bin/                 +- bin/
+- py33/                +- py33/
+- win64/               +- win64/
+- [Spr.pyd] <-----[Link]----- +- Spr.pyd

```

シンボリックリンクを利用すると、このように同じファイルやフォルダが複数の場所にあるような状態を作ることができます。開発はユーザーフォルダで行いつつ、動作チェックも瞬時に行うことができます。

シンボリックリンクを張るにはコマンドプロンプトを**管理者権限で起動**し、

```

> cd C:\Program Files\Blender Foundation\Blender\2.65\scripts\modules
> mklink /j spb C:\users\[ユーザー名]\SprBlender\scripts\modules\spb
> mklink Spr.pyd
    C:\users\[ユーザー名]\SprBlender\spbapi_cpp\bin\py33\win64\Spr.pyd

```

のようにします。

## 環境変数の設定

Blender が 32bit 版か 64bit 版かで設定する内容が異なります。

32bit 版 Blender を利用する場合

```
c:\users\[ユーザー名]\Springhead2\bin\win32
```

64bit 版 Blender を利用する場合

```

c:\users\[ユーザー名]\Springhead2\bin\win32
c:\users\[ユーザー名]\Springhead2\bin\win64

```

に環境変数 PATH を設定してください（環境変数の設定の仕方は § 1.3 を参照）。**64bit 版の場合は win32 と win64 の両方に PATH を通す必要がある**ことに注意してください<sup>\*4</sup>。

<sup>\*4</sup> と書きましたが本当に必要なか自信が持てなくなってきました。確認してもし必要ないようならこの記述は修正します。



## 第2章

# Blender の基本操作

---

ここでは, SprBlender を使うにあたって最低限知っておくとい Blender の機能について簡単にまとめます.

Blender の詳しい使い方については, すでにわかりやすく解説された書籍や Web サイト等が多数ありますのでそちらを参考にされることをおすすめします.

## 2.1 ファイル操作

### 2.1.1 .blend ファイルを開く

---

## 2.2 画面分割

### 2.2.1 画面を分割する

---

### 2.2.2 画面の分割を解除する

---

## 2.3 3D View の操作

### 2.3.1 視野の回転・ズーム・視野中心の移動

---

### 2.3.2 パース表示・直交投影表示

---

### 2.3.3 マウス・キーボードの設定

---

## 2.4 物体の操作

### 2.4.1 追加

---

### 2.4.2 選択・削除

---

### 2.4.3 レイヤー

---

### 2.4.4 変形・回転

---

物体の中心について,

## 2.5 “Outliner” ウィンドウ

## 2.6 “Properties” ウィンドウ

### 2.6.1 “Object” パネル

---



## 第3章

# 剛体で遊ぶ

---

まずは剛体を作って遊びます。

### 3.1 物体を剛体にする



## 第4章

# キーフレーム録画

---

シミュレーションの様子をキーフレームアニメーションに記録することができます。

### 4.1 録画してみる



## 第5章

# 関節と IK

---

関節の作り方と，IK の使い方についてです．

### 5.1 関節を作ってみる



## 第6章

# スキンメッシュ

---

Blender では、Armature の機能を使ってスキンメッシュを動かすことができます。Armature と Springhead の剛体・関節モデルを連動させれば、物理に従って動作するスキンメッシュを実現できます。

曲線・曲面状に変形する物体を作成したり、キャラクタの身体をシミュレーションしたりする事に利用できます。

### 6.1 Springhead-Armature-Skinmesh 連動の仕組み





## 第7章

# スクリプティング

---

シミュレーション実行中に Python スクリプトを実行して、物理モデルに加わる力やトルク、位置や速度等を設定することができます。これにより、シミュレーション内の物理モデルをリアルタイムで制御することができます。

制御がなければただ単に物がぶら下がったり転がったりするだけですが、スクリプトを書いて制御を行えば機械やロボットのように能動的に動くものをシミュレートできます。

### 7.1 スクリプトを置く場所

### 7.2 SprBlender の内部構造



## 第8章

# クリーチャ

---

Springhead には、キャラクタ・いきもの・ロボットといった「クリーチャ」の動作・行動生成を実現するための機能が搭載されています。SprBlender のクリーチャ機能では、Springhead のもつクリーチャ機能の利用に加え、パラメータ編集 UI や AI 数値の可視化など独自のクリーチャ作成補助機能が備わっています。

### 8.1 クリーチャの構成



## 第9章

# UI リファレンス

---

SprBlender で利用可能な UI の一覧を示します.

### 9.1 Scene パネル



## 第10章

# トラブルシューティング

---

### 10.1 概要

本章ではよくあるトラブルとその対処方法を紹介します。

### 10.2 プログラムのビルド

主に Visual Studio 上で Springhead をビルド際によく遭遇するトラブルと対処法について述べます。

**現象** 「include ファイルを開けません」というコンパイルエラーが出る

**解決策** プロジェクト設定で Springhead のインクルードパスが設定されているか確認してください。

**現象** 「未解決の外部シンボル〜が〜で参照されました」というリンカエラーが出る

**解決策** プロジェクト設定で Springhead のライブラリパスが設定されているか確認してください。

**現象** 「〜は既に〜で定義されています」というリンカエラーが出る

**解決策** プロジェクト設定で、ランタイムライブラリがユーザプログラムと Springhead ライブラリとで同一か確認してください。

## 10.3 物理シミュレーション関連

## 10.4 ファイル関連

現象 ファイルがロードできない

解決策 1 プログラムの実行ディレクトリを確認し、ファイルのパスが適切か確認してください。

解決策 2 バイナリ形式の X ファイルは読めません。テキスト形式で保存してください。

解決策 3 特殊な方言が使われていて読めないこともあります。開発者に問い合わせてください。

## 10.5 その他のトラブル

現象 なぜか動かない

解決策 バージョンが古い可能性があります。まず `svn update` を実行してコードを最新にし、次に Springhead ライブラリのリビルドを行ってください。

現象 それでも動かない

解決策 たまに (?)Springhead のバグであることがあります。開発者に問い合わせてください。