

Mitaka Plus スクリプト入門

株式会社オリハルコンテクノロジーズ

高幣俊之

Mitaka Plus 1.5.4 + http://orihalcon.jp/mitakaplus/mitakaplus_154_jpa.zip



地上から宇宙の果てまでをリアルタイムに映像化 するインタラクティブな宇宙映像表示ツール

対象:

太陽系内の天文現象 宇宙の階層構造の俯瞰 特徴:

- インタラクティブな描画
- ・各要素を正しく映像化
- 階層構造を見ていくのに
 特化した視点移動操作



Mitaka Plus



Mitakaのソースコードをベースに、利便性や 拡張性を向上させた派生ソフトウェア。



Mitaka Plus

Mitaka

Mitaka Plus



Mitakaのソースコードをベースに、利便性や 拡張性を向上させた派生ソフトウェア。 (中身は既にMitaka由来部分が2割、新規開発が8割)

地上絵の表示・GUIの強化





88星座の星座絵表示





88星座の立体表示





日食の本影・半影表示





小惑星探査機はやぶさの追加





金星探査機あかつきの追加







- スクリプトによるカスタマイズと自動化が可能
- 外部データの編集による天体の追加が可能
- 88星座絵の表示、地上絵の表示、スムーズな離着陸と天体間移動、オブジェクトのフェードによる表示・非表示切り替え、任意の天体比表示、日食のコーン表示
- ターゲット選択、日時・観測地設定GUIの追加
- 天体や星座のGUIによる個別表示・非表示切り替え
- 彗星・探査機(はやぶさなど)の追加
- 描画品質の向上と動作速度の高速化

GUIの追加:ターゲットブラウザ





- Mitaka Plus内の天体を階層表示
- チェックボタンで個別の表示・非表示
- 左クリックで選択して天体の情報表示
- ダブルクリックで天体まで飛行
- 右クリックメニューから
 - 天体への飛行
 - ・天体へのジャンプ
 - ・天体のセンタリング
 - ・天体の追跡(地上で)





時刻設定ダイアログ

- カレンダーや数値入力での日時設定
- 指定時刻へのアニメーション
- 手軽な時刻速度スライダー



観測地設定ダイアログ

- ・「観測地」の追加と削除
- 観測地ごとの風景設定
- マウスクリックでの緯度経度移動
- 地球俯瞰でのラベル設定

プレゼンテーションのための機能



- 2 画面モード: プロジェクタ表示と操作画面を分離
 - 1. [表示]→[投影画面]で別ウィンドウ
 - 2. ウィンドウをプロジェクタ側に移動
 - 3. [Alt]+[Enter]キーで全画面表示
 - 4. [ツール]→[終了時のウィンドウ配置を保存]に チェックが入っていれば次回起動時までそのまま
 - 観客には宇宙映像だけを見せられる
 - 手元ではGUIをフル活用できる
 - スクリーンメニューを見せないで操作できる





- フェードイン・フェードアウト
 - → 場面切り替えや補助投影としての利用に便利 2画面モードでは、手元画面は薄く表示
- バーチャルカーソル
 [Tab]キーで表示切り替え
- GUIスタイルの変更
 [ツール]→[GUIスタイル]→[黒背景]

状態の保存と復帰



- 現在の状態をそのまま記憶
 [編集]→[状態の保存]
- 記憶した状態に戻す

 [編集]→[状態の復帰]
 → 手軽な「元に戻す」機能として活用
- 現在の状態をそのままファイルに保存 [ファイル]→[状態を保存]
- ファイルに保存した状態に戻す
 [ファイル]→[状態を読み込み]

状態ファイルの活用



- ・現在の状態をファイルに保存 [ファイル]→[状態を保存]
- ファイルに保存した状態を読み込み
 [ファイル]→[状態を読み込み]
 → あとで使いたいシーンをいくつも用意しておける
 - メニューからシーンを選択したい
 - ショートカットキーでシーンを呼び出したい
 - フェードインから始めたい
 - 今夜の〇〇時の状態にしたい

Mitaka Plusの映像調整



描画設定のカスタマイズ(mitaka.ini)

- 視野範囲の変更 [Configuration] / ScreenW = 180.0
- 日時表示高さの変更
 [Date] / PosY = 0.0
- 天の川銀河の描画品質(より軽く/より精細に)
 [Performance] / GalaxyRayNum,
 GalaxyRayStepLength, GalaxyStarNum
- ・惑星の陰影輝度を変更
 [Lighting] / Ambient = 0.1
- 太陽の発光サイズを変更
 [Sun] / GlareSize = 5

Mitaka Plusの映像調整-2



- 地球の海面反射の変更
 [Earth] / Specular = 6.0 海面反射の強さ
 [Earth] / Shininess = 60.0 ハイライトの大きさ
 [Earth] / AtmosphereIntensity = 6.0 大気の輝度
- 文字描画フォントの変更
 FaceNameSB = "Arial"
 FaceNameMB = "MS ゴシック"

- 半角文字のフォント 全角文字のフォント
- 各種描画色の変更
 [Color] / * = #CCBFF
 <u>16進数で赤、青、長を0-255(FF</u>)の順に記述

変更をmitaka.iniに保存してMitaka Plusを起動





ラベルの変更

- 惑星名•衛星名 data/ssystem_jp.dat
- 恒星名

data/star_names_jp.dat

• 星座名

data/constellation_names_jp.dat

- 近傍銀河名 data/nearby_galaxy_names_jp.dat
- 天体表面上の地名 data/defaultplaces_jp.dat

Mitaka Plusのデータ調整



太陽系内天体の追加

- ・ 衛星:data/orbits/*.dat(data/ssystem_jpから参照)
- 小惑星:data/orbits/asteroids/*.dat
- 彗星:data/orbits/comets/*.dat

以下を記述したテキストファイルをフォルダに 置くと、小惑星・衛星として追加される。

- Name: スクリプトで参照する名前(空白不可) DisplayName: 表示される名前
- q:近日点距離(天文単位)
- e:離心率
- i: 軌道傾斜角(度)
- w:近日点引数(度)
- node:昇交点黄経(度)

http://neo.jpl.nasa.gov/orbits/

Hale-Bopp(1995O1)

```
Name: Hale-Bopp(1995O1)
DisplayName: ヘール・ボップ
q: 0.91413637
e: 0.99508608
i: 89.43019
w: 130.58921
node: 282.47089
Tp: 2450539.63709
Size: 500
Visible: 1
```



Mitaka Plusのスクリプト

Mitaka Plusのスクリプト



- Mitaka Plusは「Lua」というスクリプト言語を使う
- 簡単な文法で一行のコマンド実行から 複雑なプログラムまで幅広く対応できる
- Lua言語は市販の書籍で詳しく学習できる
- いろいろ覚えなきゃいけないの?



Mitaka Plusのすべての動作は スクリプトによって実行されている

- Mitaka Plusの操作の様子を記録すれば、
 その操作を実行するためのスクリプトがわかる
 - |1. [ツール]→[スクリプト編集]で編集画面を開く
 - 2. この画面の[ツール]→[記録]で記録開始
 - [時刻]→[時刻の表示]で時刻表示の切り替え Time:toggleTimeInfo()が記録される
 - 4. [ツール]→[記録]で記録終了
 - 5. [ツール]→[実行]で動作確認

スクリプトリファレンスで確認



- 1. $[\land \nu \nu] \rightarrow [\land \rho \nu] \land \rho \nu = 1$ doc/script_jp/index.htmlをウェブブラウザで開く
- 2. 「スクリプトリファレンス/Time」のページで Time:toggleTimeInfo()の内容を確認

Time:toggleTimeInfo() 時刻情報の表示・非表示を切り替えます。このメソッドに よってTime.timeInfoプロパティが変化します。 例: Time.toggleTimeInfo(); 関連: Time.timeInfo

ショートカットキーを追加



- スクリプトマニュアルの「ショートカット」参照
- 1. テキストエディタ(メモ帳など)で scripts/shortcuts_jp.luaを開く
- Input:addShortcut("T", false, "Time:toggleTimeInfo()", "時刻の表示・非表示"); を一番最後に追記して保存。
 - カッコ内の意味は順番に、
 - ショートカットキー(""で囲む)
 - ・ 押しっぱなしで実行するかどうか(true/false)
 - ・実行したいスクリプト(""で囲む)
 - 説明(""で囲む。MitakaPlusのヘルプ表示で使用)
- 3. Mitaka Plusを再起動

もうひとつショートカットキーを追加

Input:addShortcut("O", false, "Planets:toggleOrbits()", "惑星軌道線の表示・非表示"); Input:addShortcut("JSBUTTON4", false, "Planets:toggleOrbits()", "惑星軌道線の表示・非表示");

ALCON

複数のスクリプトを一度に実行したい場合は ; (セミコロン)で区切って並べる。 忘れないよういつも;を末尾に付けておくと良い

Input:addShortcut("L", false, "Planets:toggleOrbits(); Planets:toggleLabels()", "惑星の軌道線とラベルの表示・非表示");

スクリーンメニューを追加



- スクリプトマニュアルの「スクリーンメニュー」参照
- 1. テキストエディタ(メモ帳など)で scripts/screenmenu_jp.luaを開く 2. 一番最後の2行を参考に menuTest = ScreenMenu:addMenu("テスト"); menuTest:addAction("軌道線の表示・非表示", "Planets:toggleOrbits();"); を追記 (menuTest は好きな英語名でOK) テスト 3. Mitaka Plusを再起動 軌道線の表示・非表示



スクリーンメニューで状態を切り替え



dofile('*mitakaフォルダからの相対ファイルパス*'); 状態ファイルやスクリプトファイルを実行

- 1. [ファイル]→[状態を保存]でいくつかの状態を state/ファイル名.lua として保存
- 2. テキストエディタ(メモ帳など)で scripts/screenmenu_jp.luaを開く
- 3. 一番最後に menuTest:addAction("*状態の説明*", "dofile ('state/ファイル名.lua');"); を追記
- 4. Mitaka Plusを再起動



GUIを自由にレイアウトしたい: [ツール]→[UIブラウザ]でHTMLによるGUIを表示

- Ul/index.htmlや、ここからリンクされる HTMLファイルでGUIを作成
- Mitaka Plusで実行したい内容を "mtkp://スクリプト"としてハイパーリンク 例:

< a href="mtkp://dofile('ssystem.lua')"

- GUIにはテキスト、画像、Flash、 複雑な仕組みもJavaScriptで作成可能
- オンラインでGUIの共有も



少しスクリプトの文法を…



Time:toggleTimeInfo();

Time ←「オブジェクト」 関連する機能をまとめたもの toggleTimeInfo() ←「メソッド(命令)」 実行したい命令。命令によっては カッコ内に,(カンマ)で区切って 命令の詳細を記述する

命令の実行では、オブジェクトとメソッドの間を (: (コロン) で区切る

スクリプトの簡単な実行



・ [ツール]→[スクリプト出力]で 「スクリプト出力」ウィンドウを表示 下端の入力行でスクリプトを入力して[Enter] 正しく実行されれば、実行したスクリプトが ulletウィンドウの上部に履歴表示される • 書き間違いがあれば、入力内容とともに error: のメッセージが上部に表示される [↑][↓]キーで過去に入力したスクリプトを 入力行に呼び出せる

オブジェクトとプロパティ



Time.timeInfo

Time ←「オブジェクト」 関連する機能をまとめたもの timeInfo ←「プロパティ」

状態を表す値。プロパティによって 整数や実数、真偽(true/false)や文字列など 値の種類も決まっている

オブジェクト.プロパティのようにピリオドで区切る

プロパティに値を設定すれば状態が変わる プロパティの値を調べれば状態がわかる

プロパティの使い方



- 「スクリプト出力」ウィンドウで
 プロパティの設定
 Time.timeInfo = true;
 で日時の表示。文法は
 オブジェクト.プロパティ = 値;
- プロパティの取得 print(Time.timeInfo) で「スクリプト出力」ウィンドウに値が表示される

利用例

Time.JD = Time.JD + 0.99726; 1恒星日進める。地上で惑星の移動を見せるのに便利

スクリプトの例



- 今日の夜21時の太陽系を表示
- 1. 太陽系全景が見える視点に移動
- 2. [ツール]→[スクリプト編集]で 「スクリプト編集」ウィンドウを開く
- 3. [ツール]→[状態を記録] 現在の状態がプロパティ設定の羅列として記録される
- 4. 最後に以下を追記
 Time:setCurrent(); -- 現在時刻に設定
 Time.localHour = 21; -- 日本時間の21時に設定
 Time.localMinute = 0;
 Time.localSecond = 0;
- 5. [ファイル]→[スクリプトとして保存]でファイルに保存 (以上、状態をファイルに保存して編集するのと同じ)

便利なスクリプト



 フェードアウト Scene.Fader:fadeOut(1.5); -- 1.5秒でフェードアウト
 フェードイン

Scene.Fader:fadeIn(1.5); -- 1.5秒でフェードイン

3. フェードアウト→スクリプト実行→フェードイン Scene.Fader:fadeOutIn(秒数, "*スクリプト*");

例:フェードでシーン(state/ssystem.lua)に切り替え Scene.Fader:fadeOutIn(1.5,"dofile('state/ssystem.lua');");

視点アニメーションの作成



- 1. [ツール]→[スクリプト編集]で 「スクリプト編集」ウィンドウを開く
- 2. [ツール]→[状態を記録]で現在の状態を記録
- 3. 視点を操作して移動先の位置・方向へ
- (ツール)→[視点を記録]で視点設定スクリプトを記録
 5秒で移動するアニメーションスクリプトに書き換え
 - Camera.azimuth = 30; \rightarrow
 - Camera.altitude = 40; \rightarrow
 - Camera.fov = 60; \rightarrow
 - Camera:setOrient(0,0,0,1); \rightarrow
 - Camera.zoom = 0.5; \rightarrow

Camera.abc = 0;
$$\rightarrow$$

Camera:setAzimuthTo(30, 5); Camera:setAltitudeTo(40, 5); Camera:setFovTo(60, 5); Camera:setOrientTo(0,0,0,1,5); Camera:setZoomTo(0.5,5);

Camera:setAbcTo(0,移動秒数);

アニメーションの時間制御



指定秒数だけ待つ: waitTime(秒数);

使用例:

Camera.zoom = 0.5;-- 最初の距離Camera:setZoomTo(10, 5);-- 10倍の距離へwaitTime(5);-- 移動が終わるまで待つwaitTime(2);-- もう少し待つCamera:setZoomTo(20, 5);-- さらに倍の距離へ

タイムラインでの編集



「タイムライン」

動画編集ソフトのように、動きや表示の変化を インタラクティブなグラフを使って直感的に編集

タイムライン編集 *		×
ファイル(E) 編集(<u> </u>	E)	
🗋 🖕 💥		
名前	94777	
 ▶ ラベル ⑤ スクリプト ♪ BGM ♪ 効果音 		1
フレーム Root ⊡Scene 戸image ✓ position ✓ rotation		
V alpha V visibility 100% V	o 	9



「タイムライン」は Mitaka Pro の機能ですが、 実はコマンドを知っていればMitaka Plusでも使えます...

GUI:toggleTimelineEditor(); --「タイムライン編集」ウィンドウの表示

「フレーム」はフィルムのコマのようなもの 1フレーム=1/30秒

1秒=30フレーム

「タイムライン編集」ウィンドウ右下で 作成したいショートクリップの長さを設定

タイムラインでの視点移動



- 1. 宇宙空間で最初の視点位置を決める
- タイムラインで■が一番左(フレーム0)の位置で ツールバーの「
 視点を記録」をクリック
- 3. 移動先の視点位置を決める
- 4. ■をフレーム100(3秒ほど後)にドラッグ
- 5. 再び「🞥 視点を記録」をクリック
- 6. 視点移動アニメーション完成! ■をドラッグするか、下の再生ボタンで再生

フレームスクリプト



いつも同じ内容を再生させるために、初期状態を記録 1. タイムラインで■が一番左(フレームO)の位置で ツールバーの「 □ 視点を記録」をクリック

特定のタイミングで表示を切り替える

- タイムラインで■がフレーム30(1秒後)の位置で 「スクリプト」の列をダブルクリック 「スクリプト編集」の[フレームスクリプト」タブが現れる
- ここで変化させたい内容を記述

 (「記録」ボタンで操作を記録するなどして。
 例: Constels:toggleLines();)

タイムライン関係のスクリプト



Timeline:play(); -- タイムラインの再生 Timeline:pause();-- タイムラインの一時停止

タイムラインをファイルに保存 「タイムライン編集」ウィンドウで[ファイル]→[保存] ファイルの拡張子は.ltlまたは.lua (これもスクリプトなので) 保存されたタイムラインの実行 dofile("state/test.ltl");

タイムラインで作成した映像の<u>書き出し!</u> GUI:saveMovieDialog(); -- 宇宙飛行のムービー化など





- Mitakaの醍醐味は、宇宙とMitakaに詳しい語り手が 観客と対話しながら見せるインタラクティブな上映
- 語り手が使いやすいツールとして洗練したものが Mitaka Plus
- スクリプトを勉強する!と構えずに、普段の使い勝手を
 向上する方法はないかと試してみるくらいの気持ちで
- 追加データ、状態ファイル、GUIのカスタマイズ、
 スクリプトなどを通して、見せ方や宇宙の語り方、
 演出ノウハウの共有に役立ててください