

/*

LScript Layout

Parenter_ds

LW11.6~

V1.0

Daisuke Suzuki

History: 2014/09/01

*/

Parenter_ds は、アイテムを固定させる為のレイアウトプラグインです。主にキャラクターの歩行や物を持つ時などに利用しますが、考え次第で様々な利用方法が見つかると思います。

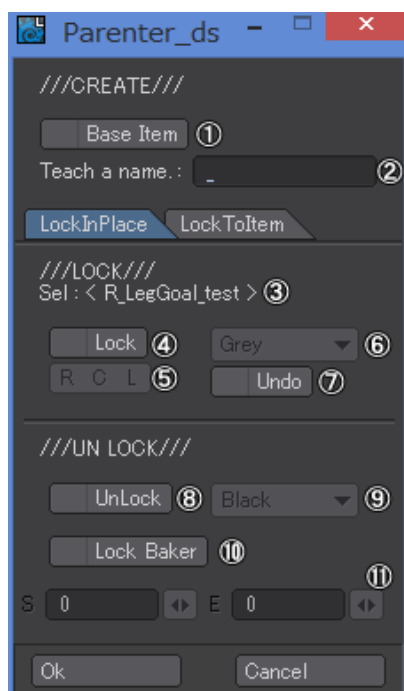
【利用条件】

- ・自動キーをオンにしてください。
- ・同名のアイテム (Null (1) 、Null (2)) には利用しないで下さい。

パネルの説明

◆ LockInPlace (その場で固定) -----

現在選択されているアイテムの位置に LockItem を追加します。



注意 : 同名のアイテムには利用出来ません。

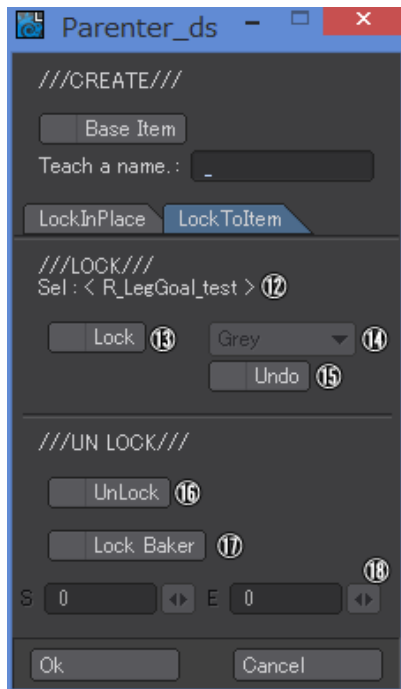
Null (1) 、Null (2) のような同名で自動で
番号がカウントされるアイテムには利用出
来ません。

- ① Base アイテムを追加します。LockInPlace と LockToItem を利用する時に必要なものです。
- ② 接尾辞のキャラ名を入力します。
- ③ 現在選択されているアイテム名が表示されます。
- ④ その場でアイテム位置を固定します。
- ⑤ LockItem をどの Base アイテムの子にするか決めます。(④をオンで有効)
- ⑥ LockItem の色を決めます。(④をオンで有効)

- ⑦誤ってアイテムをロックした時に元の状態へ戻します。
- ⑧アイテム位置の固定を解除します。
- ⑨ LockItem の色を決めます。(⑧をオンで有効)
- ⑩ LockItem の動きを固定されたアイテムへ焼き付けます。
- ⑪始まりと終わりのフレーム範囲を指定してモーションベイクさせます。(⑩をオンで有効)

◆ LockToItem (アイテムへ固定) -----

現在選択されているアイテムの位置に LockItem を追加します。



注意 : 同名のアイテムには利用出来ません。

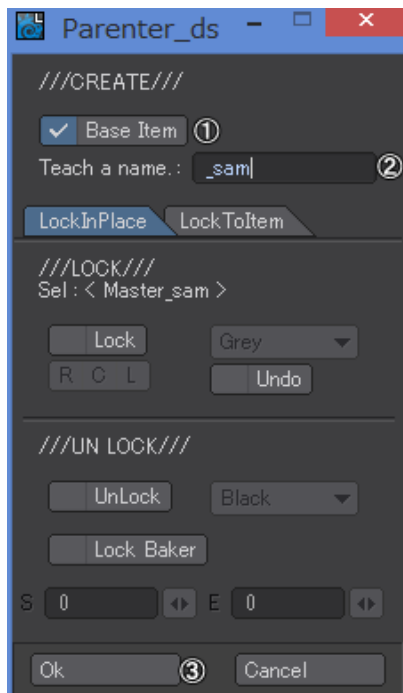
Null (1) 、Null (2)のような同名で自動で番号がカウントされるアイテムには利用出来ません。

- ⑫ 現在複数選択されている2番目のアイテム名が表示されます。

- ⑬ 1 番目に選択したアイテムを 2 番目に選択したアイテムへ固定します。
- ⑭ LockItem の色を決めます。LockItem (Null) 自体は非表示になっています。
(⑬をオンで有効)
- ⑮ 誤ってアイテムをロックした時に元の状態へ戻します。
- ⑯ アイテム位置の固定を解除します。
- ⑰ LockItem の動きを固定されたアイテムへ焼き付けます。
- ⑱ 始まりと終わりのフレーム範囲を指定してモーションベイクさせます。
(⑰をオンで有効)

使用方法 (早見表)

◆ LockInPlace の使用手順 -----



[1] : Base Item

固定したいアイテムにベースアイテムを追加します。

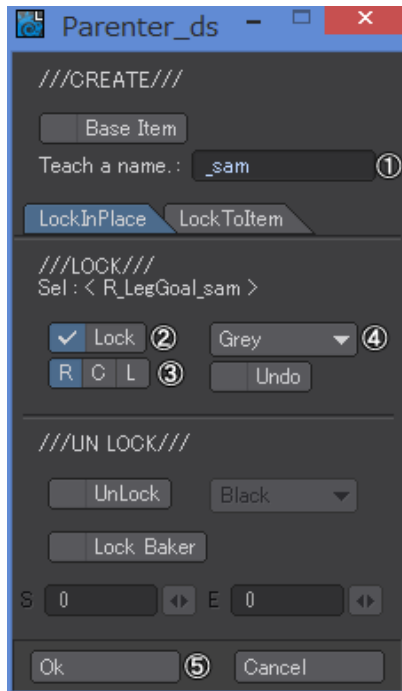
[準備]

一番親に当たるアイテムを選択します。

① Base Item オン。

② 名称(接尾辞) 入力。
例: L_GoalItem_sam → 接尾辞の「_sam」を入力。

③ スクリプト 実行。



[2] : LockInPlace (Lock)

その場でアイテムを固定します。

[準備]

固定したいアイテムを 1 つ選択して固定開始フレームへ移動します。

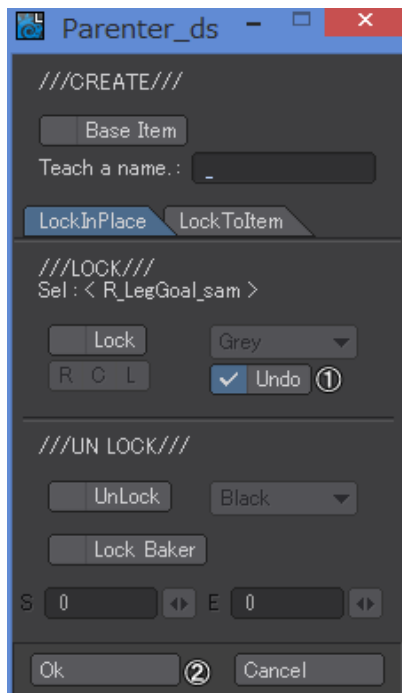
① 名称 入力。

② Lock オン。

③ どの Base アイテムの子にするか。
(Right / Center /Left)

④ LockItem の色。

⑤ スクリプト 実行。



[3] : LockInPlace (Undo)

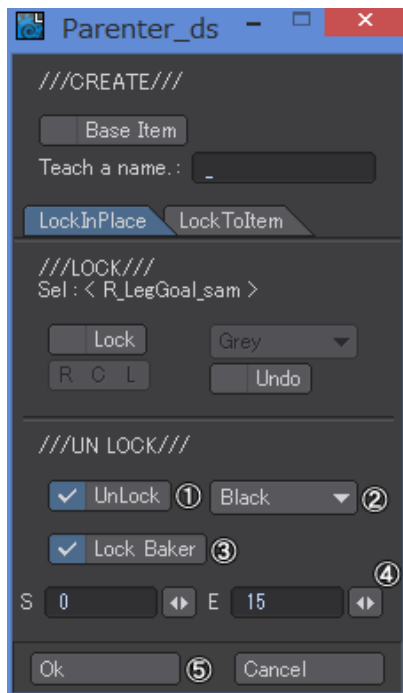
誤って【Lock】を実行した場合はアンドゥで元の状態に戻します。

[準備]

固定された/したアイテムのどちらか 1 つを選択します。

① Undo オン。

② スクリプト 実行。



[4] : LockInPlace (UnLock)

Lockしたアイテムを解除します。

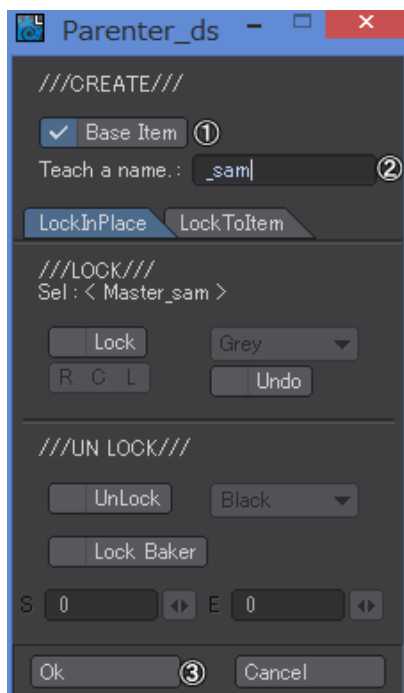
[準備]

固定された/したアイテムのどちらか 1 つを選択して
固定終了フレームへ移動します。

- ① UnLock オン。
- ② UnLockItem(目安アイテム)の色。
- ③ Lock Baker オン。
(固定されたアイテムに動きを焼き込む。)
- ④ UnLock Baker のフレームの範囲 指定。
(S はスタート。 E はエンド。)
- ⑤ スクリプト 実行。

後は [2] ~ [4] の繰り返しです。

◆ LockToItem の使用手順 -----



[1] : Base Item

固定したいアイテムにベースアイテムを追加します。

[準備]

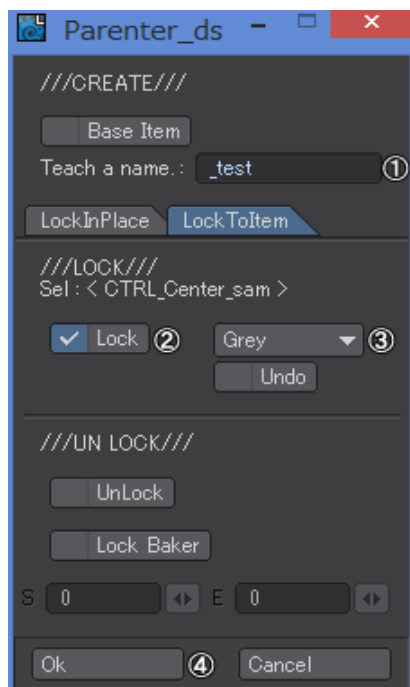
一番親に当たるアイテムを選択します。

① Base Item オン。

② 名称 入力。

例: L_GoalItem_sam → 接尾辞の「_sam」を入力。

③ スクリプト 実行。



[2] : LockToItem (Lock)

選択したアイテムへ固定します。

[準備]

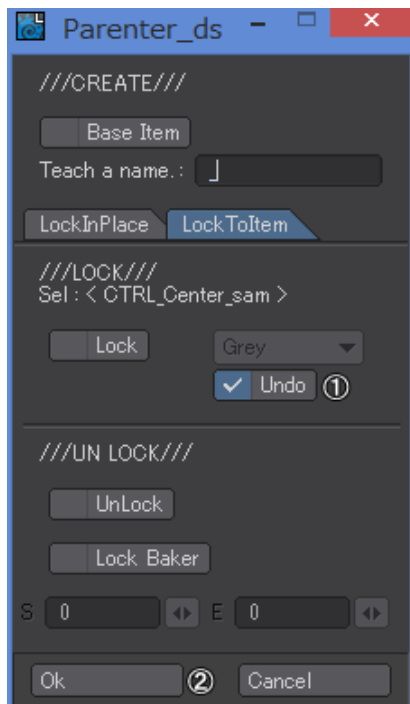
アイテムを2つ選択して、1番目に選択したアイテムを2番目に選択したアイテムへ固定します。

① 名称 入力。

② Lock オン。

③ LockItem の色。

④ スクリプト 実行。



[3] : LockToItem (Undo)

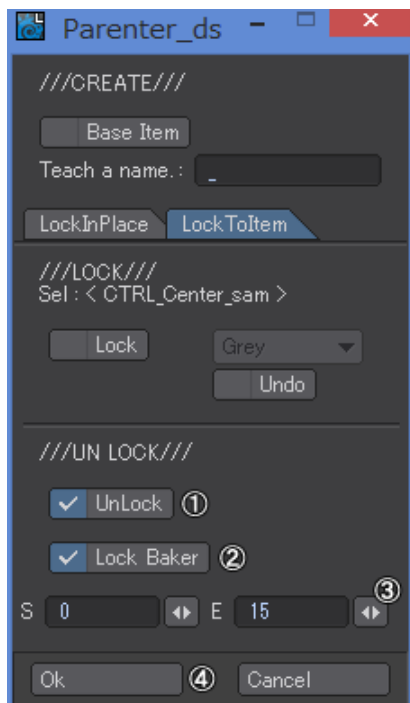
誤って【Lock】を実行した場合はアンドゥで元の状態に戻します。

[準備]

固定された/したアイテムの両方を複数選択します。
(Lockした時と同様の選択順にしてください。)

① Undo オン。

② スクリプト 実行。



[4] : LockToItem (UnLock)

Lockしたアイテムを解除します。

[準備]

固定された/したアイテムの両方を複数選択します。
(Lockした時と同様の選択順にしてください。)

① UnLock オン。

② UnLock Baker オン。
(固定されたアイテムに動きを焼き込む。)

③ Lock Baker のフレームの範囲 指定。
(S はスタート。 E はエンド。)

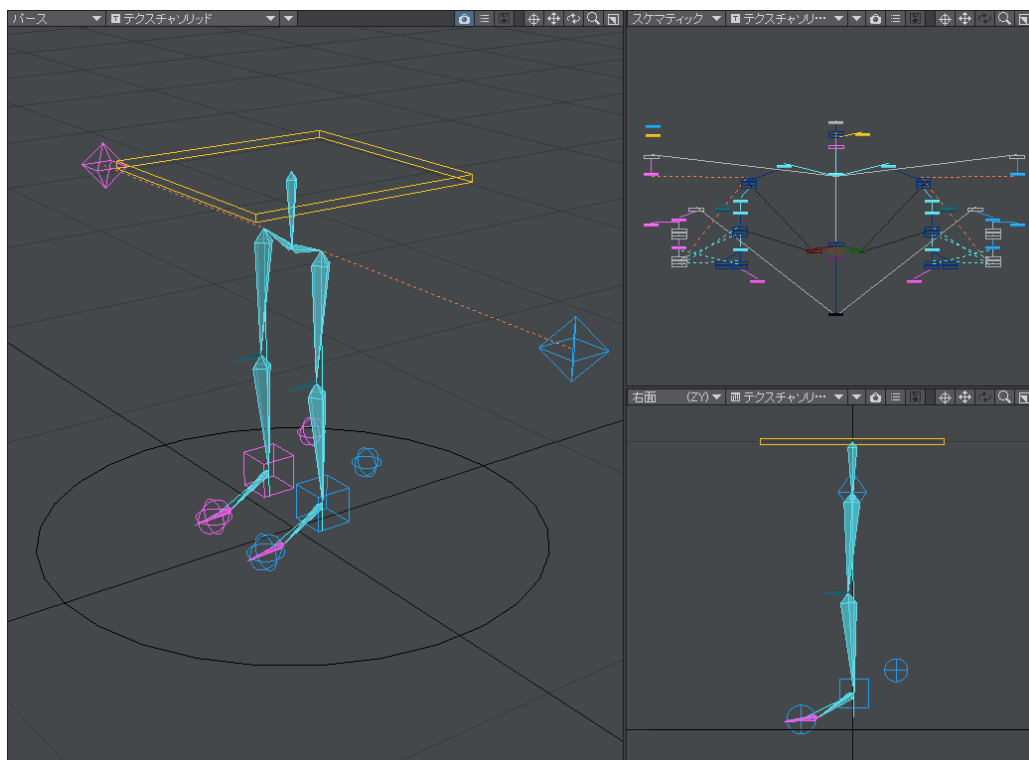
④ スクリプト 実行。

後は[2] ~ [4] の繰り返しです。

使用方法（実例）

◆ サンプル -----

ダウンロードされたファイルには下半身リグのシーンファイルが含まれています。このサンプルを例に **Parenter_ds** の使用方法をご説明します。



(0001_Sample.lws)

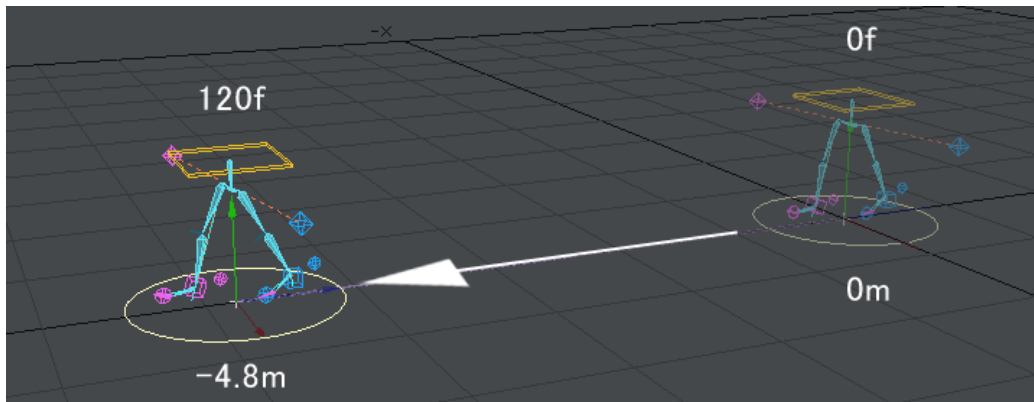
(0002_Sample_Walk.lws)

◆ 歩行 -----

LockInPlace を利用して歩行モーションを作成します。

[準備]

- 1 : 「0002_Sample_Walk.lws」のシーンを開きます。
このシーンファイルには移動しない状態の歩行サイクルが作成されています。



- 2 : フレームスライダを 120 フレームヘドラッグ、Master アイテムを選択して Z 方向に -4.8m 移動します。プレビュー再生すると歩く動きが出来上がったと思います。しかし、この状態では足が滑っているように移動しているので、足のゴールアイテムを地面にしっかり固定させる必要があります。
ここから Parenter_ds の出番です。

[BaseItem を追加]

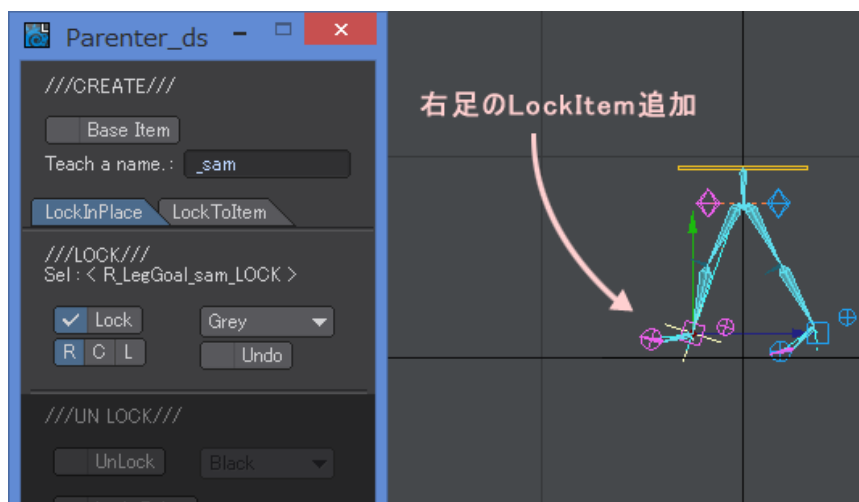
- 3 : LockItem を追加する為の準備をします。
一番親である Master が選択されている事を確認して Parenter_ds のコマンドボタンを押します。「Base Item」のチェックボックスをオン、「Teach a name.」にキャラ名を入力します。今回は適当に sample の「_sam」をキャラ名として入力しましょう。
(下の画像の Master アイテム以下に接尾辞で「_sam」が付けられていますが、これは付けても付けなくてもどちらでも構いません。Base アイテムに名前さえ付けてあれば Parenter_ds は問題無く利用出来ます。)



- 4 : OK ボタンを押すと、4 つのアイテムが追加されてリグの一番親のアイテムが「Base_ (キャラ名) 」になりました。

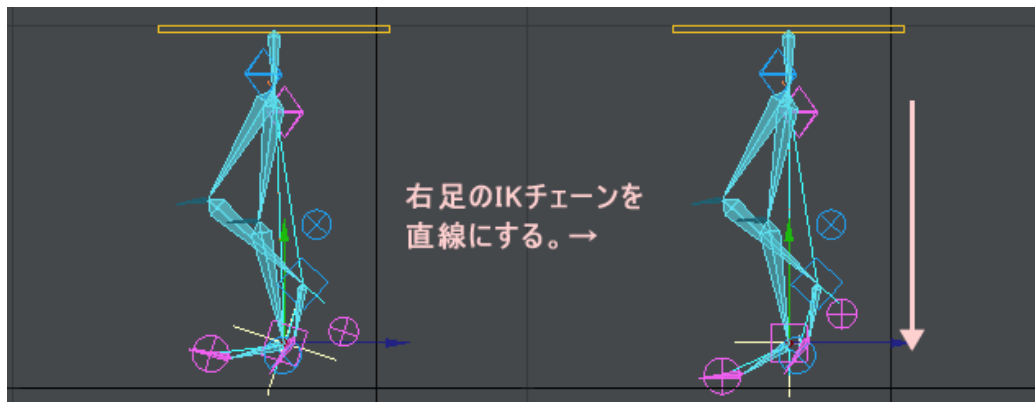
[LockItem を追加]

- 5 : 右足のゴールアイテムから固定していきましょう。
0 フレームであることを確認して右足のゴールアイテムを選択、Parenter_ds を起動します。
- 6 : 「Teach a name.」にキャラ名を入力して、LockInPlace タブの「Lock」のチェックボックスをオンにします。



- 7 : 「R | C | L」のボタンと LockItem の色のプルダウンが有効になりました。「R | C | L」は、どの Base アイテムの階層に LockItem を追加するかを決めます。
右足のゴールアイテムを選択してるので「R」を選択しましょう。

- 8 : OK ボタンを押すと、右足に **LockItem** が追加されました。
フレームスライダを動かすと足が固定されているのが分かります。



- 9 : 現在選択されている **Null** が **LockItem** なので早速動きを付けていきます。7 フレームへ移動して横面から見て **IK** チェーンが真っ直ぐになるように移動して回転の **P** 値を 0° にします。
フレームスライダを動かして足の動きが自然になっているか確認してください。理想的な足の接地が出来ていると思います。

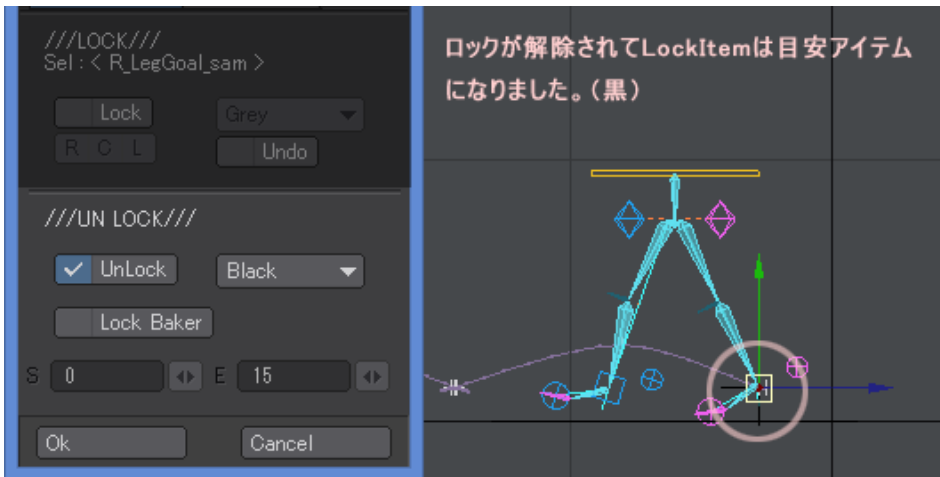
[**LockItem** を解除]

- 10 : 7 フレームから 15 フレームへ移動してアイテムの固定を解除しましょう。この時、

- ・「**R_LegGoal_LOCK**」
- ・「**R_LegGoal**」

どちらかを選択していることを確認してください。

- 11 : **Parenter_ds** を起動します。
LockInPlace タブの「**UnLock**」のチェックボックスをオンにします。**UnLock** 実行後、**LockItem** は何の意味を成さない足跡の目安アイテムになるので、右足の **LockItem** だと分かり易いように **Black** から他の色を指定して構いません。



- 12 : OK ボタンを押すと、右足のアイテムが **LockItem** から解放されて 0～15 フレームの範囲で **LockItem** の位置に固定されました。
- 13 : しかし、これは始めのフレームから固定を行ったものなので、**UnLock** をオンにするだけで上手く出来ました、それ以降の動きを固定する場合には「**LockBaker**」を利用してモーションベイクする必要があります。言っている意味がよく分からないかも知れないので、今度はあえて失敗してみます。

[2 歩目の **LockItem** 追加]

- 14 : 左足のゴールアイテムを選択して 2 歩目の **LockItem** を追加しましょう。15 フレームへ移動して 5～12 と同じ作業を行います。今度は左足なので「**R | C | L**」のボタンの選択は「**L**」にしてください。
- 15 : 無事に出来たように見えますが、よく見ると 15 フレーム～22 フレームの間が固定されていません。これは前の動きがある為に上手く固定出来ていないのです。ここもしっかり固定させたい場合には「**LockBaker**」を利用するのでもう一度アイテムを固定し直しましょう。
- 16 : **LockInPlace** は再度アイテムを固定することは出来ない、こういった修正には **LockToItem** を利用していきます。

[LockToItem で固定の修正]

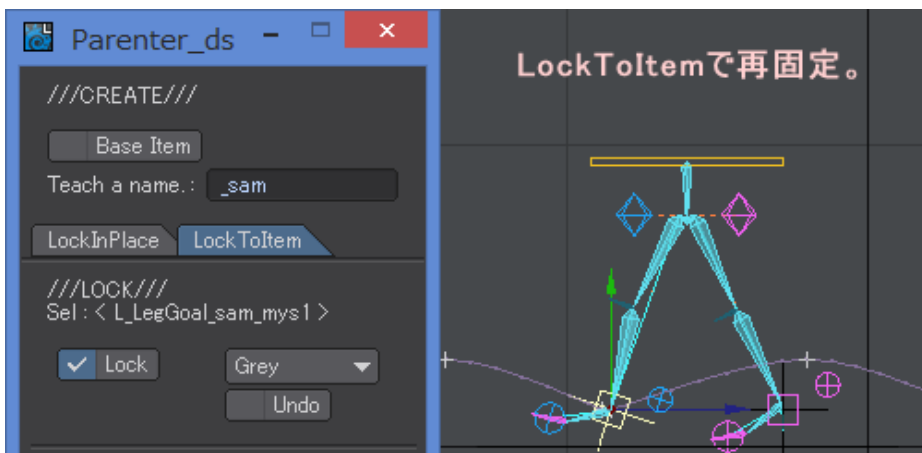
- 17 : LockToItem は本来、物を持ったり放したりする時に利用しますが、LockInPlace で残った目安アイテムを利用して再度固定修正を行うことができます。

LockToItem の利用条件は、

- ・ 一番目に選択したアイテム = 固定されるアイテム
- ・ 二番目に選択したアイテム = 固定するアイテム

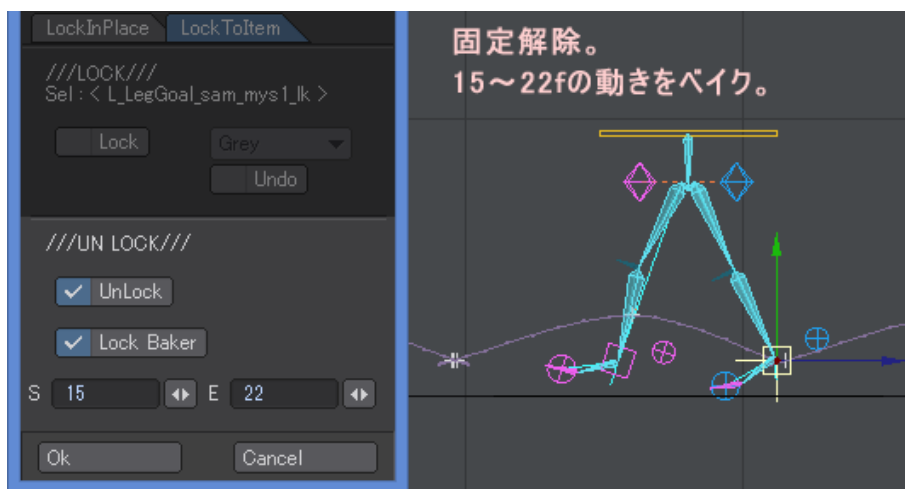
このように 2 アイテム複数選択で認識されるようになり、コマンドを実行する為の条件になります。

- 18 : これを踏まえて、左足のゴールアイテムと左足の目安アイテムを選択します。選択順に注意してください。
(目安アイテムはビュー上で選択出来ないようにしている為、スケマティックビューから選択してください。)



- 19 : 15 フレームへ移動して Parenter_ds を起動します。
「Teach a name.」にキャラ名を入力して、LockToItem タブの「Lock」のチェックボックスをオンにします。
- 20 : フレームスライダを動かすと再度固定されたのが確認できます。
そのまま 30 フレームへ移動して UnLock しましょう。

- 21 : Lockした時の手順と同様にアイテム選択がされているのを確認したら **Parenter_ds** を起動します。**LockToItem** タブの「**UnLock**」と「**LockBaker**」の2つのチェックボックスをオンにします。
(**LockBaker** は単体で利用出来ません。**UnLock** と同時に行ってください。)
- 22 : すると、始まり(S)と終わり(E)のフレームが指定出来ます。
15～30フレームの固定範囲をモーションベイクしても良いですが、無駄にキーを作成したくないので上手く固定出来ていない15～22フレームの範囲を指定します。



- 23 : OK ボタンを押すと、モーションベイクとロック解除が実行されて、今度はしっかり固定された動きになりました。**LockBaker** は **LockInPlace** からでも行うことが出来るので、あらかじめ固定させるフレーム範囲が分かっている場合は **LockToItem** の修正作業を行う必要はありません。

これで、**Parenter_ds** の一通りの説明は終わりです。
次は **LockToItem** の本来の使用方法をご説明します。

◆ 足で玉転がし -----

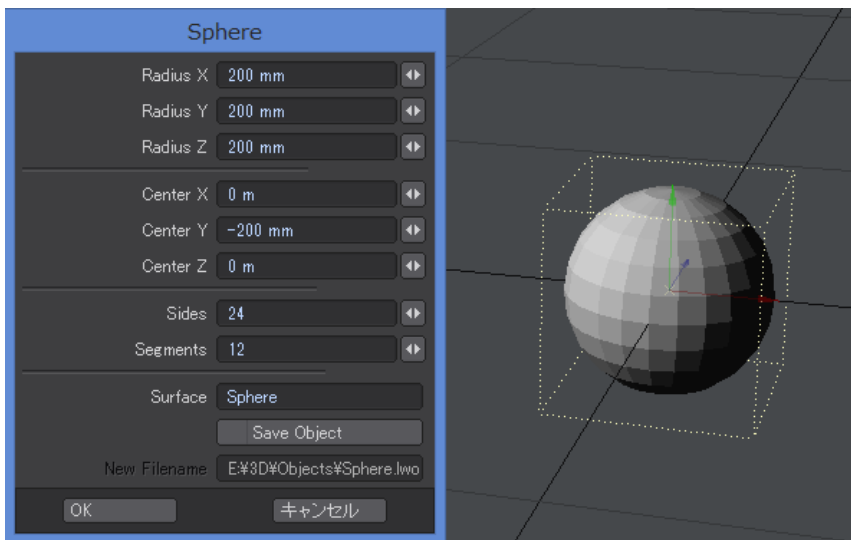
LockToItem を利用して玉転がしをします。球のリグを作成する為に「BallRig_ds」の L スクリプトを利用していきます。

[準備]

- 1 : アイテムタブ→開く→別シーンから「0001_Sample.lws」を開いて下半身のリグを追加します。Parenter_ds で BaseItem を追加します。名前は先ほどと同じ「_sam」にしましょう。

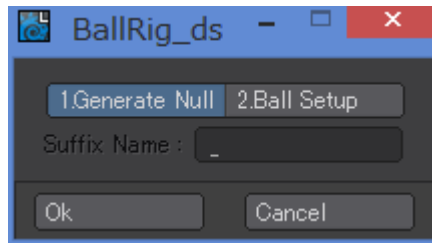
(LockToItem を利用するだけなら BaseItem の子アイテム「R_Lc_sam」、「C_Lc_sam」、「L_Lc_sam」を削除して構いません。)

- 2 : 足で転がす為の球を作ります。
レイアウトの「モデラーツール」タブ→「作成」カテゴリ→「球」を選択します。
- 3 : Radius の XYZ の値を 200 mm、Center Y の値を -200 mm にして OK ボタンを押します。

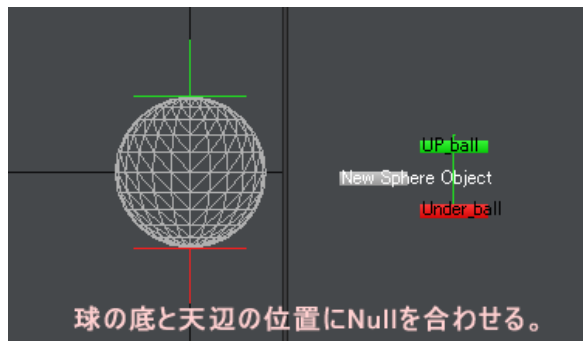


※ 球が見易いように下半身のリグは非表示で説明していきます。

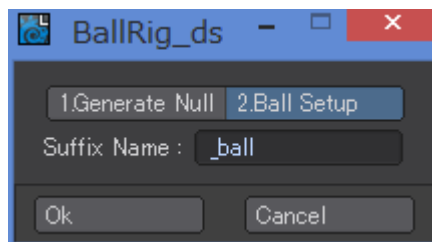
- 4 : 球のオブジェクトが作成されましたので、次はリグを作成します。



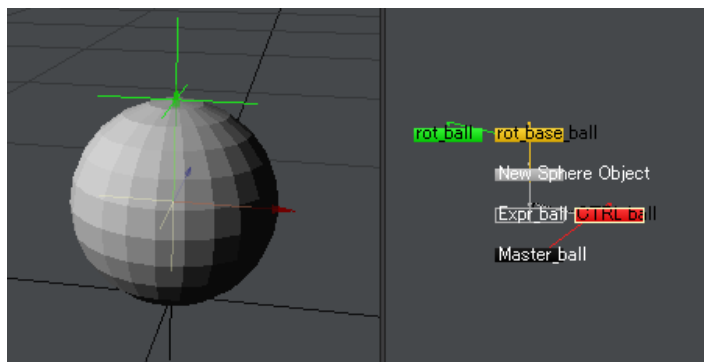
- 5 : 球が選択されていることを確認したら BallRig_ds のコマンドボタンを押します。「1.Generate Null」がアクティブになっているのを確認して OK ボタンを押しましょう。



- 6 : リグを作成する為の 2 つの Null が追加されました。これは球の大きさを測る為のアイテムです。
赤色の Null「Under_ball」をボールの底 -200mm に移動して、
緑色の Null「UP_ball」をボールの天辺 400mm に移動します。



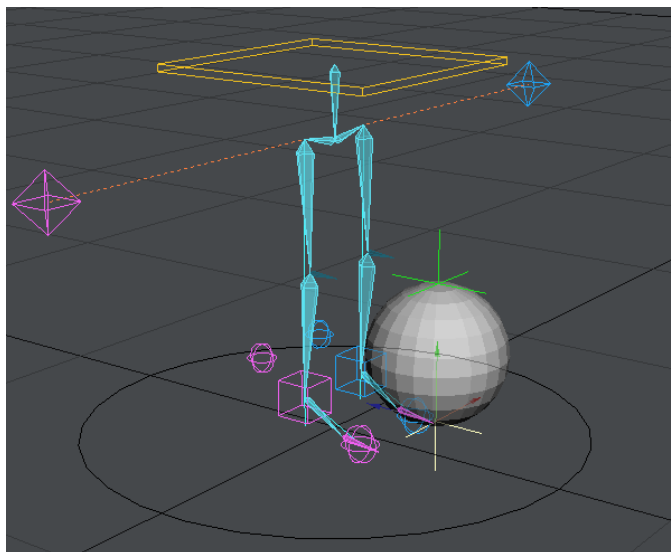
- 7 : 球が選択されていることを確認して、もう一度 BallRig_ds のコマンドボタンを押して「2.Ball Setup」を選択します。すると、Suffix_Name がアクティブになるので「_ball」と適当に名前を付けて OK ボタンを押します。



- 8 : 球のリグが作成されました。CTRL_ballを選択した状態で移動するとコロコロと転がります。

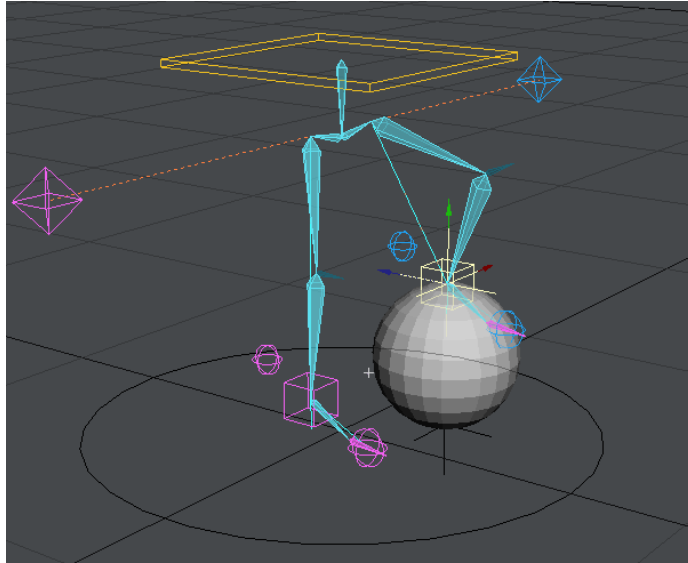
これで LockToItem を利用する為の準備が出来ました。

[LockItem を追加]



- 9 : 球を左足の手前に置きましょう。
一番親アイテム「Master_ball」を選択して X 方向に 200mm、Z 方向に -200mm 移動します。

- 10 : 左足のゴールアイテムをボールに固定していきます。左足のゴールアイテム「L_LegGoal」を選択して、Shiftキーを押しながらボールの緑の固定アイテム「rot_ball」を選択します。
- 11 : Parenter_dsを起動します。
「Teach a name.」にキャラ名「_sam」を入力して、LockToItemタブの「Lock」のチェックボックスをオンにします。



- 12 : 足が球に固定されました。玉の制御アイテム「CTRL_ball」を移動するとボールをグリグリと転がす動きが付けられます。
- 13 : NullとBoneだけでは分かり辛いですが、足が球にめり込んでいる場合は「rot_ball」に追加された子アイテム「rot_ball_none_lk」を移動して位置調整してください。

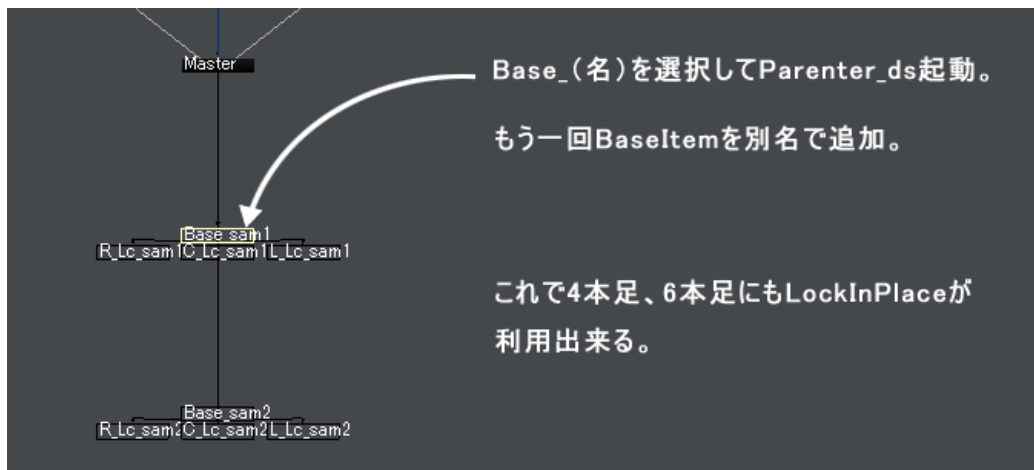
あと、球の黄色アイテム「rot_base_ball」を回転するとボールの形状に沿って足のゴールアイテムを移動させることが出来るので、「rot_ball_none_lk」と上手く利用してください。

- 14 : UnLockについては既に歩行で説明しましたので省きます。
これで使用例の説明はお終いです。

◆ 犬や昆虫など二足以上の場合-----

LockInPlace は二足歩行限定の利用を想定していましたが、使い
方次第で犬や昆虫といった 4 本足や 6 本足にも利用出来ます。これは
実例ではなく、読んで理解する程度に留めてください。

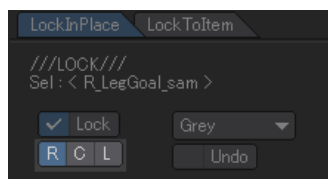
- 1 : **LockInPlace** や **LockToItem** を利用するのに **BaseItem** を追加するのはもう理解されていますね？ **BaseItem** を追加すると最大 3 つまでの **LockItem** を追加することが出来ます。
(**LockToItem** はいくつでも出来ます。)
- 2 : しかし、これでは 3 足までしか利用出来ません。もっと **LockInPlace** を利用する為の **Base** アイテムを増やしたい・・・
そうです、増やしてしまえば良いのです。追加した **BaseItem** にまた新しく **BaseItem** を作ってしまいましょう。
- 3 : 現在一番親に当たるアイテムは「**Base_**(キャラ名)」なので、このアイテムを選択してもう一度 **Parenter_ds** で **BaseItem** を追加します。「**Teach a name.**」には別の名前を入力するか、名前+番号を入力すると良いでしょう。



- 4 : **BaseItem** を二連続で追加した結果です。アイテムを **Lock** する時に、「**Teach a name.**」にそれぞれの **BaseItem** に付けた名前を入力すれば、各ベースアイテムの子として **LockItem** が追加されます。この並びでは目安アイテムが被ってしまうので、**BaseItem** のスキマティックの位置を横にずらして見易くすると良いです。

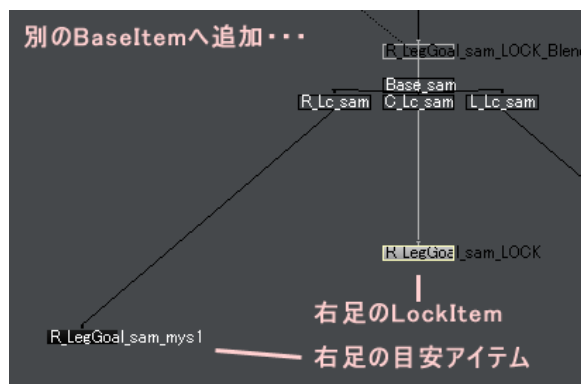
注意点

◆ Base アイテムと Lock アイテムの関係

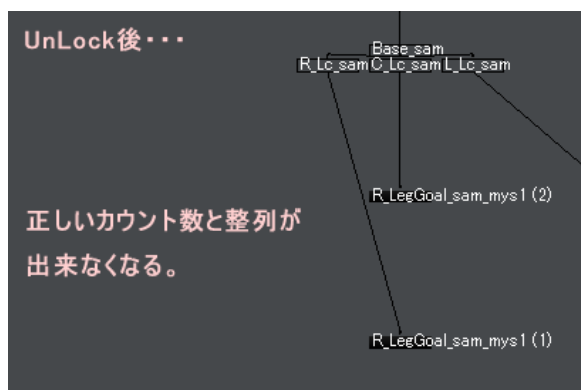


LockInPlace にはどの Base アイテムに LockItem (子) を作成するかを決めるボタンが存在しますね？このボタンを指定する事により UnLock した後に目安アイテムに番号が付加されスケマティックが整列されます。

しかし、途中で別の Base アイテムに LockItem (子) を作成するとどうなるでしょう？下の画像は「R」から「C」に変更して UnLock した結果です。



ビュー上の動作は何ら問題はありますが、スケマティックの整列や目安アイテムの番号付加が正しく実行されず、後々 LockToItem でモーションの修正が出来なくなります。



つまり、一度目安アイテムを作ってしまったら「他の Base アイテムに浮気しちゃ駄目よ。」という事です。もし、誤って別の Base アイテムを指定して Lock してしまった場合は「Undo」のチェックボックスでやり直すことができます。

しかし、そのまま続行して UnLock までやってしまった場合は画像のように Base アイテムとの関係が乱れてしまい、目安アイテムを全て削除してやり直す必要があります。逆に、ひとつの Base アイテムでいろいろなアイテムを固定する事は可能です。LockInPlace で LockItem を追加する際は注意してください。