☆スパーリング関連リソース作成

○３Dリソース

・モデルデータ作成

　１：MAYAからデータを吐き出し

　　resource\sparringフォルダにcmdlと同名のフォルダを作成し、

　　モデルデータをエクスポートする

　　（普通にMAYAからエクスポートすればcmdlと同名のフォルダを作成してエクスポートするはずです）

　　例：hogeモデル

　　　　待機アニメ（csklaのみ）

歩きアニメ、ジャンプアニメ（cskla、cmata両方）があった場合、

　　　　hogeフォルダの構成は以下のようになります

resource

　└ sparring

　 　　└ hoge

　 　 ├ hoge.cmdl

　 　　 　 ├ hoge\_idle.cskla

　 　 ├ hoge\_jump.cmata

　 　 ├ hoge\_jump.cskla

　 　 ├ hoge\_walk.cmata

　 　 ├ hoge\_walk.cskla

　 　 └ Textures

　 　 　└ hoge.ctex

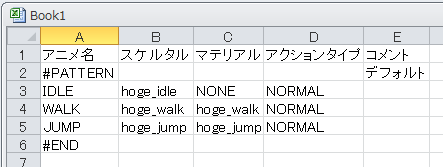
２：アニメーション登録

　　使用するアニメーションを呼び出せるようにエクセルファイルに登録します

　　エクセルファイルのファイル名は、cmdlと同じにしてください

　　エクセルファイルはcmdlと同じ階層においてください

　　エクセルファイルへの記述内容は、次のようになります



　※各項目の説明

　　アニメ名：プログラムから呼び出す時のラベルになります

　　スケルタル：csklaファイルを指定してください（拡張子は不要）

存在しない場合はNONEとしてください

　　マテリアル：cmataファイルを指定してください（拡張子は不要）

存在しない場合はNONEとしてください

　　アクションタイプ：NORMAL（通常）とETERNAL（無限ループ）を指定できます

　　　　　　　　　　　ただし、これはハーモナイトの仕様なので、

　　　　　　　　　　　Kujiraには移植していませんので、

NORMAL固定でお願いします

　　　コメント：コメント欄です

コンバートには影響がありませんの自由に記述してください

　　アニメーションの塊を#PATTERNと#ENDではさんで下さい

　　これもハーモナイトの仕様で同一モデルに対して、複数のアニメーションセットを

　　登録できるようにしていた名残りですが、これもkujiraには移植していないため、

　　アニメーションは1セット固定でお願いします

　３：OMakefileにコンバート対象として登録

　　OMakefile内のRESOURCE\_3D\_ANM\_DIRにフォルダ名（例でいうとhoge）を

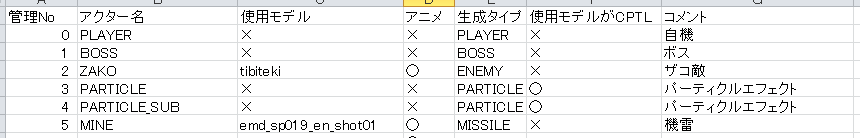
追加してください

これでmakeをかけることで、コンバートされます

・アクター登録

　　作成した3Dモデルを表示するためにアクター登録をします

　　resource\sparring\act\_dat\act\_dat.xlsを開くと以下のようになっています



※各項目の説明

管理No：アクターを管理するための通しナンバーです

アクター名：管理Noに紐づけられるプログラム側のラベル名になります

使用モデル：アクターが使用するモデルを指定します

　　　　　　ここに前項で作成した3Dモデルのフォルダ名を指定します

アニメ：使用モデルがアニメーションを持っている場合は○、ない場合は×を

　　　　指定してください

生成タイプ：アクターのタイプを指定します

　　　　　　Kujiraでは以下のものがあります

　　　　　　DEFAULT：ただ表示をするだけのもの

PLAYER：自機としての動作をするもの

　　　　　　BOSS：ボスとしての動作をするもの

　　　　　　ENEMY：ザコとしての動作をするもの

MISSILE：ミサイルとしての動作をするもの

EFFECT：エフェクトとしての動作をするもの

PARTICLE：使用モデルがパーティクルとしての動作をするもの

使用用途にあわせて、適宜、指定してください

また、アッパーにおいて新規タイプを追加する場合には、

別途、追加方法を記述します

使用モデルがCPTL：使用するモデルがパーティクルの場合は○、それ以外は×を

指定してください

コメント：コメント欄です。自由に記述してください

※新規に生成タイプを追加する方法

１：生成タイプの追加

　エクセルシートの下の方のタブにfuncというのがあるので、

　シートをfuncに切り替えてください

　内容は以下のようになります



関数リストに、生成タイプ名を指定し、関数名に生成関数名を登録してください

２：動作関数の作成

src\app\sparring\actorフォルダに動作関数のcppを作成してください

sparring\_生成タイプ\_actor.cppというファイルがいくつかあるので、

それらを参考にするとよいでしょう

　　３：生成関数の作成

src\app\sparring\actor\actor\_factory.cppに登録して関数名で

生成関数を作成してください

CreateDefaultの動作関数部分を新規に作成した動作関数に置き換えるだけで

基本的には大丈夫です

○既存の３Dリソース

別ファイルのsparring\_resource.txtにまとめてあります

○２Dリソース

　・スパーリングゲーム

　　上画面レイアウト

resource\sparring\upper\upper.clyt

下画面レイアウト

resource\sparring\lower\lower.clyt

・スパーリングメニュー

　　上画面レイアウト

　ステージセレクト

　　　resource\sparring\_menu\lyt\_sparring\_menu\stageselect\_upper.clyt

　　　リザルト画面

resource\sparring\_menu\lyt\_sparring\_menu\stageselect\_upper.clyt

下画面レイアウト

　ステージセレクト

　　　resource\sparring\_menu\lyt\_sparring\_menu\stageselect\_lower.clyt

　　　リザルト画面

resource\sparring\_menu\lyt\_sparring\_menu\stageselect\_lower.clyt

・フィールド下画面

下画面ベース

resource\training\training\_fld\_lower\training\_lower.clyt

メッセージウインドウ

resource\training\training\_fld\_lower\training\_messege\_lower.clyt

　・スパーリング関連共通素材

　　サンドバッグアイコン

resource\training\_common\sand\_bag\sand\_bag.clyt

　　ステージアイコン

　　　resource\training\_common\stage\_poke\_icon\stage\_poke\_icon.clyt

　　共通素材

　　　resource\training\_common\training\_common\_res\training\_common\_res.clyt

　・ローカライズ対象グラフィックフォント

　　resource\common\graphic\_font\sparring\_grpfont内の

各言語フォルダ内のdummy.clyt

各clytを適宜修正してください

以上になります