モーションデータは　\*.cskla から作成されます。

.cskla を変換して LittleEndian/.mot にします。

.mot をまとめたものが .gmp になります。

Powerアーキテクチャの場合はBigEndianになるので

Gflib/prog/include/qt/gfl\_QtFile.h の MODE\_BIG\_ENDIAN

のコメントアウトをはずしてコンバーターをコンパイルし直してください。

**GMPフォーマット**

.gmp のフォーマットは以下のようになっています。

バイナリファイルをリンクさせた標準的なものです。

総数やオフセットのサイズは４バイトです。

|  |  |  |  |
| --- | --- | --- | --- |
| .mot の数 | １番目のオフセット | ２番目のオフセット | ３番目のオフセット |
| … |  |  |  |
| １番目の .mot | ２番目の .mot | ３番目の .mot | … |

**Basepose .mot フォーマット**

１番めの .mot は特別なデータで以下のようになっています。

（ゲームで使用するモーション番号が２から始まるのはそのためです。）

１バイトのデータが並びます。

（なのでジョイント数の上限は0xff になります。）

１番めのジョイントはルートなのでジョイントデータは２番目から始まります。

これがジョイントの数だけ続きます。

|  |  |  |  |
| --- | --- | --- | --- |
| ジョイントの数 | エフェクターの数 | ２番目のジョイントの親ID | ２番目のジョイントのIK型 |
| ２番目のジョイント  の子供ジョイント数 | ３番目のジョイントの親ID | ３番目のジョイント  のIK型 | ３番目のジョイント  の子供ジョイント数 |
| … |  |  |  |

次に来るのはジョイント名（ASCII）です。 Null terminate されています。

（モデルとのマッチングがあるので、ハッシュ化していません。）

これも２番目のジョイント名から始まり、ジョイントの数だけ続きます。

|  |  |  |  |
| --- | --- | --- | --- |
| ２番目のジョイント名 | ３番目のジョイント名 | … |  |

次はジョイントローカル座標とジョイントグローバルクォータニオンです。

浮動小数点型データなので、このデータの前に１－３バイトのパディングデータが入る可能性があります。

データサイズは４バイトです。

このデータは１番目のジョイントから始まります。

|  |  |  |  |
| --- | --- | --- | --- |
| １番目のジョイントのX座標 | １番目のジョイント  のY座標 | １番目のジョイント  のZ座標 | １番目のジョイント  のXクォータニオン |
| １番目のジョイント  のYクォータニオン | １番目のジョイント  のZクォータニオン | １番目のジョイント  のWクォータニオン | ２番目のジョイントのX座標 |
| ２番目のジョイント  のY座標 | ２番目のジョイント  のZ座標 | ２番目のジョイント  のXクォータニオン | ２番目のジョイント  のYクォータニオン |
| ２番目のジョイント  のZクォータニオン | ２番目のジョイント  のWクォータニオン | … |  |

この後ろにモーションコンフィグレーションファイル .mcnf を変換したものが入ります。

最初の１バイトは設定項目が存在するかのフラグがビット格納されています。

Xy\_project/src/system/motion/MotionModel.h に定義されています。

enum {

// それぞれデータが存在するのか？のフラグ

M\_INTEREST\_LIMIT = (1 << 0),

M\_EXPRESSION = (1 << 1),

M\_SPRING = (1 << 2),

M\_MOUTH = (1 << 3),

M\_COLLISION\_BALL = (1 << 4),

M\_EFFECT\_NODE = (1 << 5),

M\_EYE = (1 << 6),

};

|  |  |  |  |
| --- | --- | --- | --- |
| 注視点処理 | エクスプレッション | 物理バネ処理 | 置換ジョイント口 |
| コリジョンボール | エフェクトノード | 置換ジョイント目 | 未定義 |

各フラグが立っていれば、データが存在しますが

フラグが立っていなければ、データブロック自体が存在しません。

注視点処理フラグが立っている場合は、注視点データブロックが続きます。

１バイトデータです。

角度は符号ありの１バイトデータとして参照されます。

|  |  |  |  |
| --- | --- | --- | --- |
| 注視点ジョイント数 | 注視点タイプ | 注視点角速度 | 注視点下限角度 |
| 注視点上限角度 | 注視点タイプ | … |  |

注視点タイプは

Xy\_project/prog/src/system/motion/LookDef.h

に定義されています。

注視点タイプは注視点ジョイントの軸の向きによって決定されます。

次のフラグが立っていればエクスプレッションデータです。

１バイトデータです。

|  |  |  |  |
| --- | --- | --- | --- |
| エクスプレッション数 | エクスプレッション  タイプ | エクスプレッション  結果収納ジョイント | エクスプレッション  係数 |
| エクスプレッション  参照ジョイント１ | エクスプレッション  参照ジョイント２ | エクスプレッション  参照ジョイント３ | … |

エクスプレッションタイプは、ExpressonDef.h で定義されています。

計算式自体は Expression.cpp に記述されています。

次のフラグは物理バネジョイントです。

１バイトデータです。

|  |  |  |  |
| --- | --- | --- | --- |
| バネ数 | ジョイントID | バネ係数 | 剛性 |
| 質量 | 重力加速度 | … |  |

次のフラグは置換ジョイントの口データです。

１バイトデータです。

|  |  |  |  |
| --- | --- | --- | --- |
| 口ジョイント数 | ジョイントID | … |  |

口を別モーションで制御したい場合に登録します。

次のフラグはコリジョンボールデータです。

１バイトデータです。

|  |  |  |  |
| --- | --- | --- | --- |
| コリジョンボール数 | ジョイントID | 半径 | X座標オフセット |
| … |  |  |  |

次のフラグはエフェクトノードデータです。

このデータは .mcnf ではなく EffectNode.xml に記述されています。

１バイトデータの後に４バイトデータが入ります。

（パディングが入る場合があります）

|  |  |  |  |
| --- | --- | --- | --- |
| エフェクトノード数 | 親ジョイントID | 親ジョイントID | …（ノード数分） |
| ハッシュ | X座標 | Y座標 | Z座標 |
| … |  |  |  |

次のフラグは置換ジョイントの目データです。

１バイトデータです。

|  |  |  |  |
| --- | --- | --- | --- |
| 目ジョイント数 | ジョイントID | … |  |

目を別モーションで制御したい場合に登録します。

以上がBasepose.mot フォーマットになります。

**.mot フォーマット**

モーションデータは基本的に各ジョイントの座標、角度、スケールのファンクションカーブで構成されています。

InverseKinematics が設定されているモーションの場合はエフェクター座標が保存されています。

最初の２バイトはファンクションカーブの数です。

ファンクションカーブの種類は３ビットで格納されています。

Gflib/prog/include/motion/gfl\_Fcurve.h で定義されています。

スタティックなデータの場合はファンクションカーブデータを持ちません。

次からのデータはモーションの長さが256 フレームより短ければ１バイト、

長ければ２バイトになります。

|  |  |  |  |
| --- | --- | --- | --- |
| ファンクションカーブキー数 | キーフレーム | キーフレーム | … |

次は実際のファンクションカーブ値が入ります。

浮動小数点型４バイトデータです。