

Asteraiser のしくみ

**OBLIQUEGLASS**  
S O R G E R Y

### 3 スプライトのしくみ

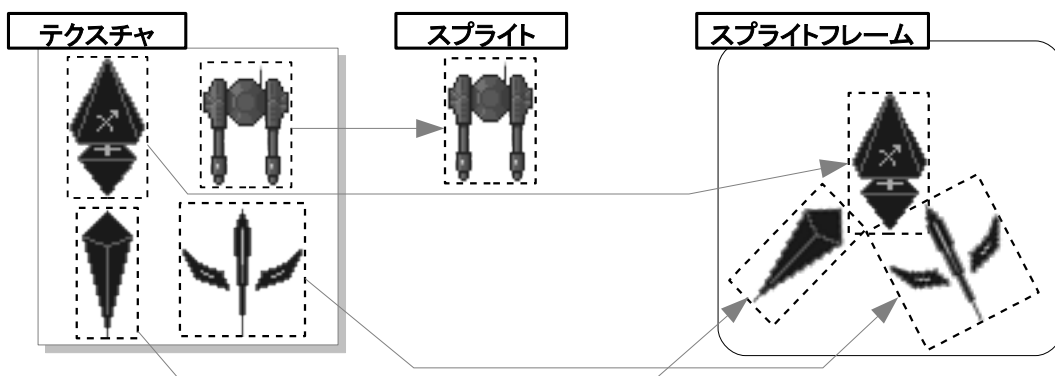
データの記述方法の説明が終わったので、次は実際に使っているデータの構造に進みます。  
 まずは画面に表示するもので最も基本となる、スプライトの解説から始めていきます。  
 「Asteraiser」では、画面に表示されるものはほぼ全て「スプライトフレーム」という仕組みで構成されています。

#### ●スプライトフレームとは

まず、スプライトというのは絵付きの四角形のことで、  
 テクスチャ(画像ファイル)から、任意の場所を切り抜いてきて表示します。

スプライトフレームとは、これらスプライトを「ボーン」として組み合わせ、  
 ひとつかたまりの物体として扱う仕組みのことを言います。

▼ スプライトフレームの例 : スプライトは単独の四角形なのに対し、スプライトフレームはその複合体。

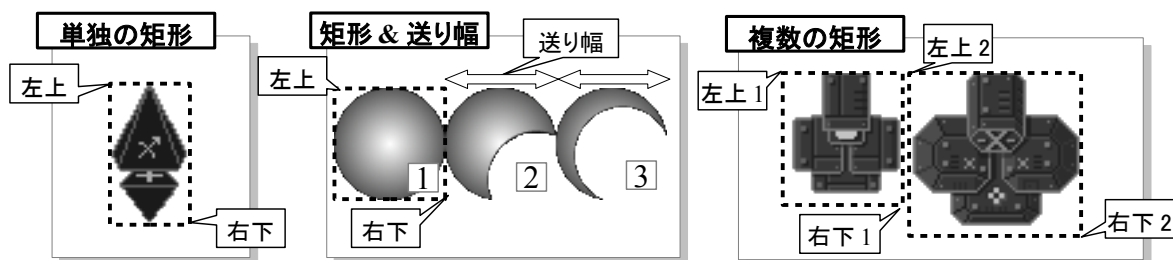


こうすることで、個々の制御側からは絵の個数を気にしなくても、  
 一つのデータを指定するだけでまとめて表示したり動かしたりができるというわけです。

#### ●ピース

テクスチャのどこを切り抜いてくるか、の情報をピースと呼んでいます。  
 ピースには、コマアニメ用にページ送りの指定方法から3つのタイプがあります。

<b>単独の矩形</b>	テクスチャ内任意の矩形座標(左上、右下)をひとつだけ指定します。
<b>矩形&amp;送り幅</b>	矩形座標と、送り幅を指定。大きさの変わらないコマアニメ用に。
<b>複数の矩形</b>	矩形座標を複数指定。大きさの変わるコマアニメをしたいときに。



また、テクスチャを指定しないでサイズだけを持ったダミーのピースも作ることができます。  
 これはメニューの背景など、無地のスプライトを扱いたいときに使います。

## ●ボーン

スプライトフレームはボーンの組み合わせで構成されています。  
ボーンは使用するピースの情報と、位置・回転角度・拡大縮小・補正色の情報を持っています。

ボーンは親子関係を持っていて、親を動かすと子も追従します。  
これは多関節のキャラクターをアニメーションさせるのに役立ちます。

▼ 親子関係： 本体→腕→剣と関係が結ばれ、腕の動きを剣が追従する。



ボーンは表示させる形によって4つのタイプがあります。

矩形ボーン	四角形のスプライトを表示する基本的なボーン。 ピースで切り抜かれた画像がそのまま表示されます。
輪形ボーン	円形に頂点を組んで表示するボーン。半径・幅・頂点数等を指定します。 円グラフの表示やエフェクトなどで使います。
文字ボーン	絵を表示する代わりに、フォントを指定して文字を表示させるボーン。 表示する文字列やアラインメントを登録し、メニューやフロントエンドなどで使います。
ダミーボーン	何も表示されない特殊なボーン。位置や回転情報のみを持ちます。 他のボーンに親子関係を組み合わせて、弾やエフェクトの発生位置指定に使います。

### 矩形ボーン



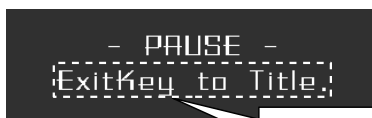
そのまま四角形が表示される

### 輪形ボーン



円形に頂点を組む一部を切り欠くことも

### 文字ボーン



文字列をボーン扱いで表示する

### ダミーボーン



実際には表示されず弾の発射位置を指定する

## ●アニメーション

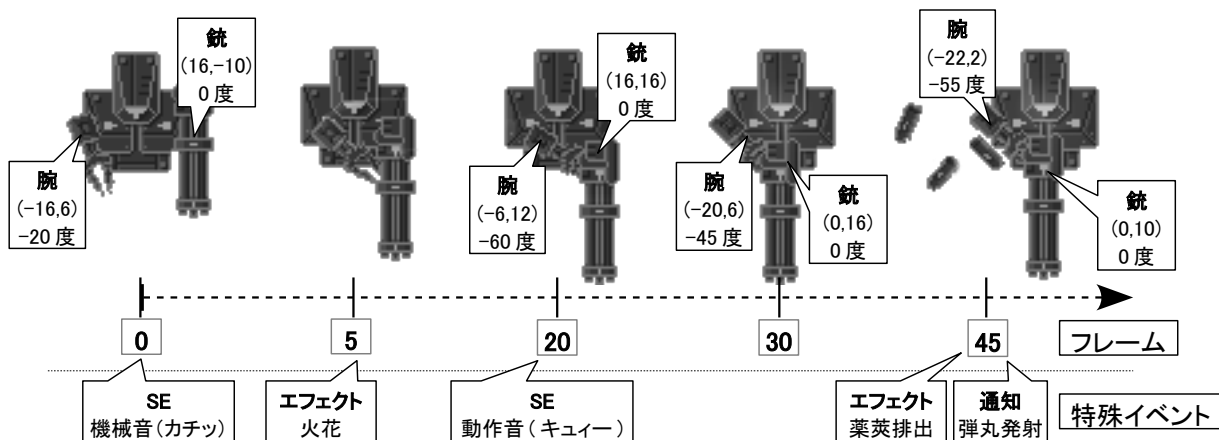
スプライトフレームはアニメーションのデータを持つことができます。

アニメは、ボーン毎にキーフレームを設定することで構成されています。  
キーフレームは、フレーム単位に位置・角度・拡大縮小・色等の変化する情報を登録する形で設定します。  
キーとキーの間のフレームは状態は、線形補間で自動的に繋がられます。

また、アニメでは特定のフレームで特殊な動作を起こす機能もあり、  
ピース・文字列の切り替えや、エフェクト・サウンドの発生などがあります。  
エフェクトは、単発で発生させる方法と、ボーンの位置に追従させて発生させる方法を選択できます。

加えて、プログラム側に任意の文字列を通知する機能もあり、  
アニメに合わせて技を発動させる／無敵時間を割り当てるような場合に使用しています。

▼ アニメーション： フレームごとにボーン位置(x,y)/角度と、発生するイベントを割り当てる。



### □ コラム:アニメのループ

アニメはループ再生するか、単発再生するかを設定できます。  
単発再生の場合、キーの最後までいくと最後のフレームの状態ですべて停止します。

ループ再生の場合はキーの最後までいくと0フレーム目に巻き戻りますが、  
オプションでループ時に戻るフレームを指定することもでき、  
構え動作の後に特定の動作を繰り返すような場合に役立ちます。

## ◆まとめ◆

スプライトフレームは、2D ゲームに必要な表示機能はほとんどカバーできる仕組みになっています。

そのおかげで、キャラ／背景／メニューといった種類を問わず、  
全てを同じ仕組みで表示することができるようになり、個々に表示物制御のコードを書く必要がなくなりました。

また、関節を使ったアニメーションは「Asteraiser」では必須ともいえる機能で、  
少しの画像で動きのバリエーションを付けることができ、素材の少なさを補う意味でも役立つと言えます。