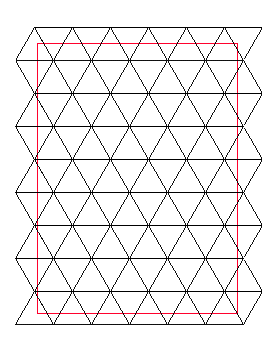
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 仕様書 | | 前作スペルカード背景 | | |
| 種類 | プログラミング資料 | | 作成者 | たかなし |
| 日付 | 2014/8/25 | |

**1　内容**

　前作スペルカード背景で使用した、ポリゴンを使ったスペルカード背景について述べる。

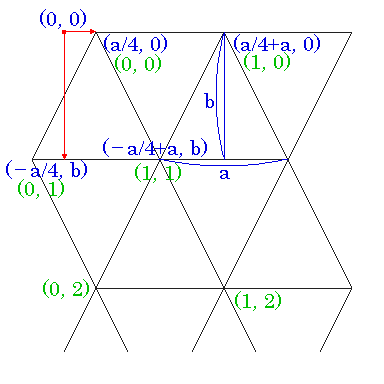


**2　原理**

2. 1　三角形の集合と位置・UV座標

　スペルカード背景を描画する際は、2Dポリゴンの三角形の集合を使ってそのテクスチャに画像を割り当て、UV座標を変化させることで変形する背景を描画している。

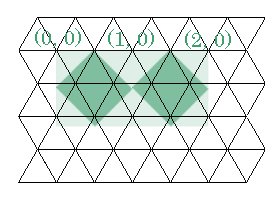
　まず、三角形の集合とは右図のようなもの。赤枠はゲーム画面。三角形の集合は画面を覆うように配置している。

　基本となる三角形の大きさをあらかじめ決めておけば、三角形の各点の座標は自動で決まる（下図）。ここで、三角形の大きさは正三角形でも良いが、√の計算が入ってきてややこしいので、底辺と高さの値をそれぞれ決めた二等辺三角形とした。また、座標の計算式が割とシンプルになるように、下図のような位置を原点とした。

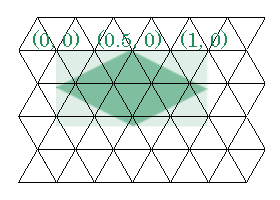
　前回は、aをゲーム画面の幅を20で割った値、bは高さを28で割った値にした。多分おおむね正三角形になるように紙面上で計算した結果だと思うが、正三角形に近い必要はあまりない。細かい模様の時に歪む可能性が出てくるくらい。

　この三角形の集合の各点は、最初に座標を決めて座標は固定し、その後適宜UV座標を割り当てる。このUV座標はエフェクトの種類や時間によって変化する。

　最も基本になるのが、三角形の座標とUV座標をそのまま対応させるものである。すなわち、テクスチャ画像のサイズを256として、画面にそのままテクスチャを敷き詰めたいときは、ある点の座標が(x, y)であれば、UV座標を(x/256, y/256)とすれば良い。

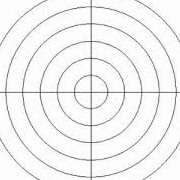
2. 2　簡単な変形

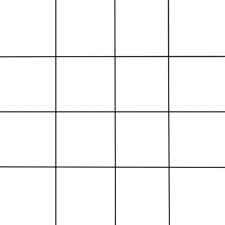
　次に、画像を拡大縮小することを考えてみると、例えば画像を横方向に2倍に引き延ばしたい時、UV座標の値をどうすれば良いかというと、1/2にすれば良い。そうすると、本来UV座標が2になる位置で初めてUV座標が1になるので、結果的に引き延ばされたようになる（右図）。

　このように、UV座標を経由して画像を変形させる場合、「逆の変形を加える（上の例なら、2倍にしたいけど、UV座標は1/2にする）」と正しく変形される感じになる。

　同様に、例えば画像を時間により平行移動させたい場合、例えば画像を1フレームに2ピクセル、x軸正方向に動かしたければ、UV座標を1フレームに2ピクセル分、負の方向にずらしていけばよい。

　画像を波のように変形させたい場合は、例えば横方向にずれるような波を作る場合、UV座標のU座標を、三角形の座標のy値に沿っていくらかずらしてやれば良い。パラメータを調整することで、時間変化する波にしたり、周期が長い・短い波を作ったり出来る。

2. 3　極座標変換

　よく使うけどもちょっと難しい変換に、極座標変換がある。これは、画像をある位置を中心にした円状に変形させる方法である。（右図）

　これは、中心点からの距離と角度を元に、UV座標を計算して割り当てる方法である。

2. 1　画像のフォルダ分けについて

今回のプロジェクトは、画像を以下のように複数の構造体に分け、そのいずれかにハンドルを持たせている。

IMAGE\_WINDOW構造体　：　ウィンドウに使う画像のハンドルを保持する構造体。

IMAGE\_CHARS構造体　：　文字画像・情報関係の画像のハンドルを保持する構造体。

IMAGE\_ILLUST構造体　：　イラスト関連の画像のハンドルを保持する構造体。

IMAGE\_ENEMY構造体　：　敵の画像を保持する構造体。

IMAGE\_ICON\_構造体　：　状態異常アイコンなど、小さめの画像を保持する構造体。

　　　　　　　　　　　　　アンダーバーがついているのは、IMAGE\_ICON\_が

　　　　　　　　　　　　　既にWindowsで定義されているため。

IMAGE\_BG構造体　：　背景画像の構造体。

IMAGE\_EFFECT構造体　：　エフェクトに関する画像の構造体。

以下のフォルダにあてはまらないものがある場合、相談の上追加しても良い。