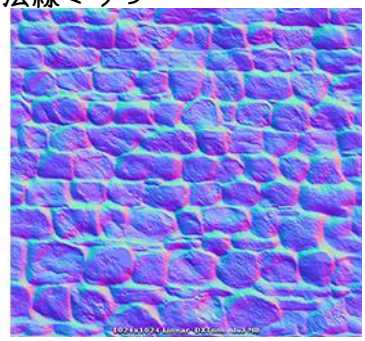
法線マップで立体的な表現

法線マップとは？(Unityマニュアルより)

バンプマップ の一種です。
モデル表面に凹凸や溝、傷などのディティールを追加することができます。

通常の画像

法線マップ



法線マップなし

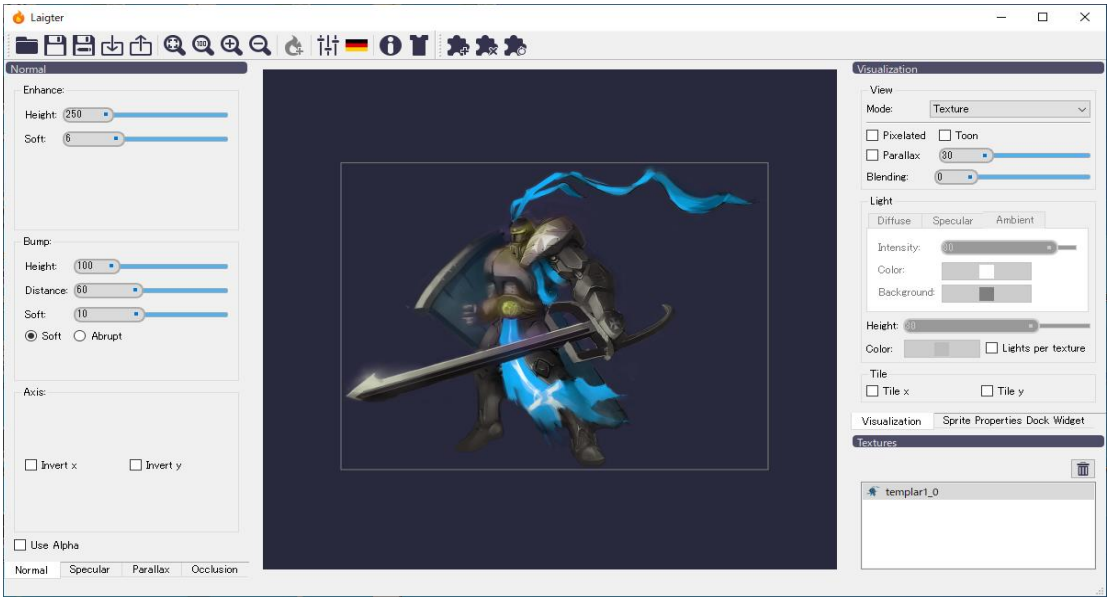
法線マップ使用

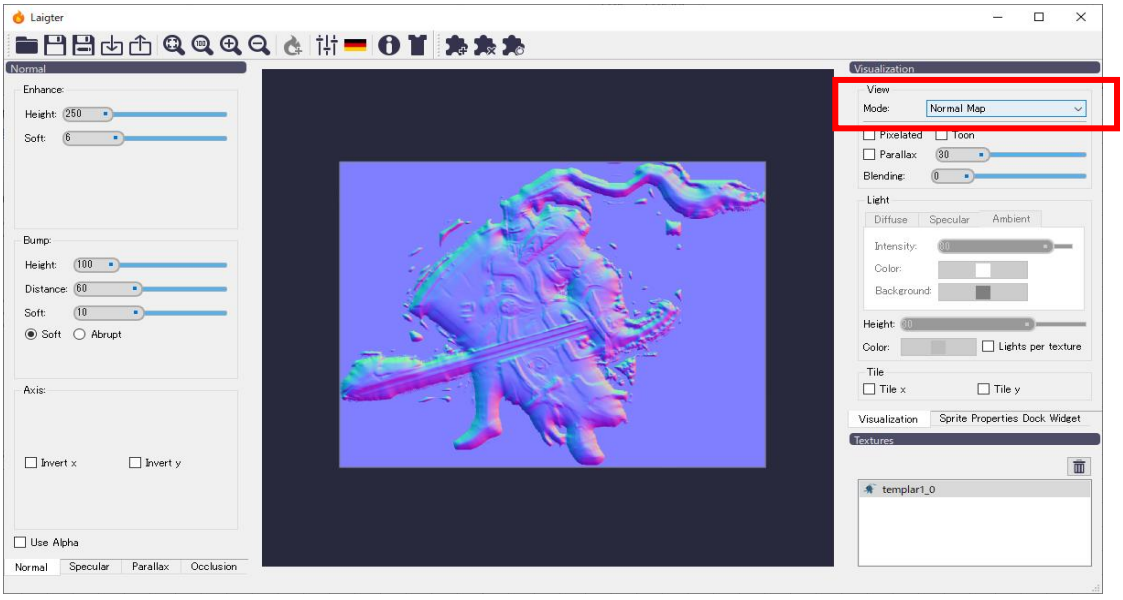
３Ｄモデルのポリゴン法線のイメージが強いかもしれませんが、
このように２Ｄ画像にもしっかりと使われます。

法線マップの入手法

私たちはプログラマーですので、デザインコストは最小にするため、
フリーのツールを使用しましょう。

「Laigter(ライター)」
https://azagaya.itch.io/laigter





法線マップの自動生成以外にも、便利な機能がありますので、
色々お試しください。

法線マップなし

法線マップ使用

青紫色になる理由

法線マップ(画像)にあるRGBのカラー値が、
X,Y,Zのベクトルの方向を意味します。
一般的には、Z か Y が“上方向”になります。
また、RGBのカラー値をベクトル方向に変換するためには、
2倍して1を引く必要があります。
RGB値が (0.5, 0.5, 1)の場合、ベクトルにした結果は(0,0,1)になります。

画像の凹凸(法線マップ)に対して、ライトがあたっているように、
陰影をつけるシェーダをこれから作成していきます。
まずは、対象の画像と、法線マップをシェーダに渡します。

```
// シェーダーにテクスチャを転送
SetUseTextureToShader(0, texDragon);
SetUseTextureToShader(1, texDragonNor);
```

シェーダ内で、法線マップのRGBをベクトルに変換

```
// ベクトルへ変換
float3 normalVec = 2.0f * norCol - 1.0f;

// 正規化
normalVec = normalize(normalVec);
```

ライト方向と法線ベクトルで、内積を使用し、
光の当たり具合を数値化する。

```
// ライト方向と法線方向の内積 = 光の当たり具合
// -1.0:反対方向 1.0:方向一致 0.5:直交
float3 bright = dot(g_light, normalVec);
```

光の当たり具合によって、明るさを調整すると、
陰影がかった表現ができる。

シェーダを使って、表現豊かなゲームを作成してみましょう！