

ライトやフォグ

フリーカメラ機能を追加しましたので、3Dワールドを自由に移動できるようになりました。

このタイミングで、3D技術をまたいくつか紹介させていただきます。

まずは、3Dの初期設定で少し触れました、ライトです。

DxLibでは3種類のライトが用意されています。

```
SetUseLighting(true);
```

まずは、上記でライトを有効にすることができます。

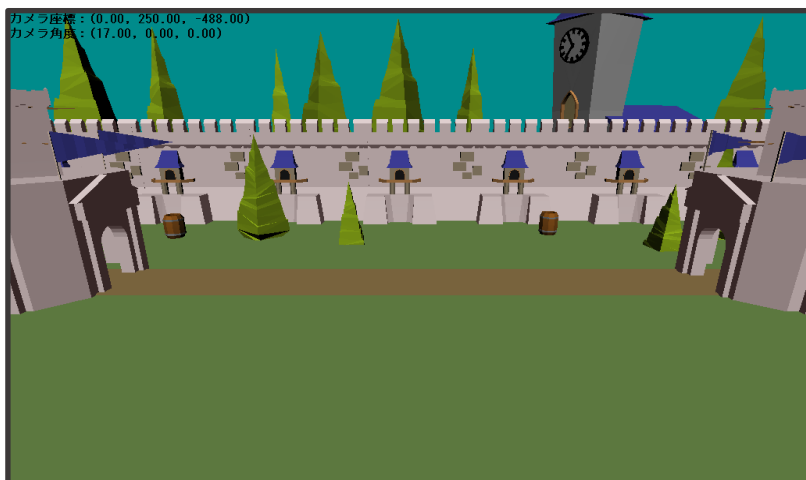
以下、リファレンスからの引用ですが、標準ライトを変更する機能です。

① ディレクショナルライト

ディレクショナルライトとは位置を持たず方向だけを設定するライト。
太陽の光などの光源が遠すぎて、
何処に居ても同じ方向から照らされるライト効果などに使用します。

```
int ChangeLightTypeDir( VECTOR Direction );
```

いま、プロジェクトで使っているライトです。
使いやすく処理負荷も他のライトに比べると高くないので、
光表現に特にこだわりが無ければ、これ一本でゲームが制作できると思います。



② ポイントライト

ディレクショナルライトの次に使用頻度の高い、指定した位置から全方向に光を放つライトです。松明や街灯など、局所的な自然光を表現する時に使用します。

```
int ChangeLightTypePoint(  
    VECTOR Position, float Range,  
    float Atten0, float Atten1, float Atten2 );
```

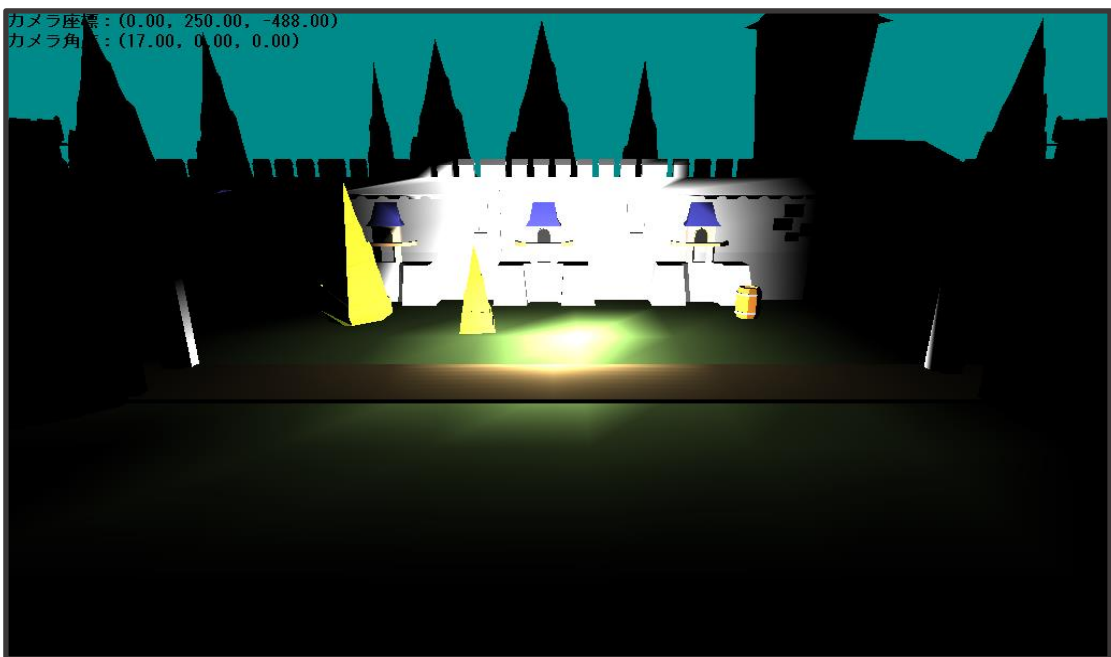
VECTOR Position : ライトの位置

float Range : ライトの有効距離

float Atten0 : 距離減衰パラメータ 0

float Atten1 : 距離減衰パラメータ 1

float Atten2 : 距離減衰パラメータ 2



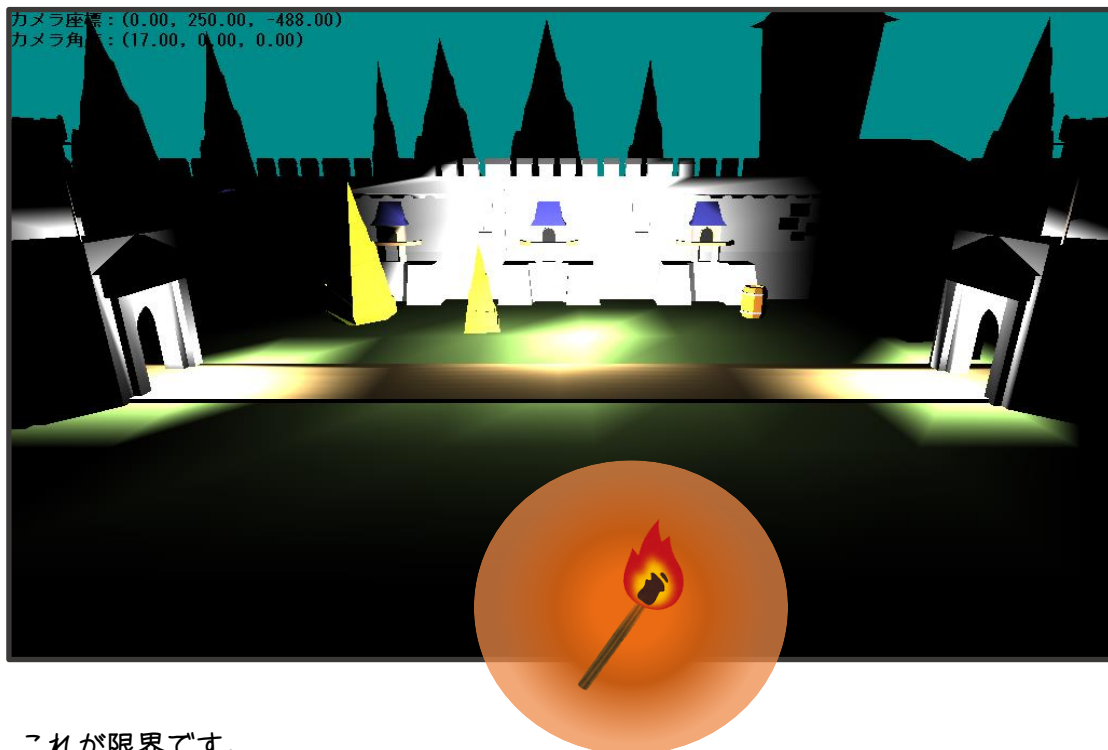
参考値

```
ChangeLightTypePoint({ 0.0f, 40.0f, 180.0f },  
    600.0f, 0.000f, 0.001f, 0.000f);
```

画面中央奥のちよい上にポイントライトを配置

このポイントライトの性質上、標準ライトというよりかは、ステージに配置されてある、松明ごとに配置していく使い方が多いです。但し、DxLibでは、標準ライトを含め、同時に3つのライトしか有効にできません。

https://dxlib.xsrv.jp/function/dxfunc_3d_light.html



これが限界です。
ゲームエンジンであれば、もっと配置できますが、
配置すればするほど、処理負荷も高くなるので気を付けましょう。

参考値

```
// 追加ポイントライト 1
pointLight1Pos_ = { -390.0f, 100.0f, 50.0f };
pointLight1_ = CreatePointLightHandle(
    pointLight1Pos_, 100.0f, 0.000f, 0.002f, 0.000f);
```

```
// 追加ポイントライト 2
pointLight2Pos_ = { 390.0f, 100.0f, 50.0f };
pointLight2_ = CreatePointLightHandle(
    pointLight2Pos_, 100.0f, 0.000f, 0.002f, 0.000f);
```

※ 解放時に必ずライトハンドルをメモリから解放すること
DeleteLightHandle(pointLight1_);

③ スポットライト

一番計算負荷が高く、また頂点単位のライティングではポリゴンを細かく分割しておかないと思ったような結果が出ないなど扱いが難しいタイプのライトです。

ポイントライトは全方向だったかと思いますが、スポットライトには方向があります。

懐中電灯やサーチライトなどといった人工的な光源として使用されることが多いです。



キャラクターに懐中電灯をあててみました。

リファレンスにも記載があるとおり、

3Dモデルが細かくないと、上手く表現できないため、

今回使用しているステージモデルでは厳しく、別モデルを用意しました。

参考値

<code>ChangeLightTypeSpot(</code>	
<code>{ 20.0f, 260.0f, -210.0f },</code>	位置
<code>{ 0.0f, 0.0f, 1.0f },</code>	方向
<code>6.28f, 0.0f,</code>	コーンの外側、内側の角度
<code>200.0f, 0.000f, 0.001f, 0.000f);</code>	有効距離、減衰

前述で紹介させて頂いたライトですが、色も変えることができます。
ディフューズ、スペキュラ、アンビエント、それぞれ設定できます。

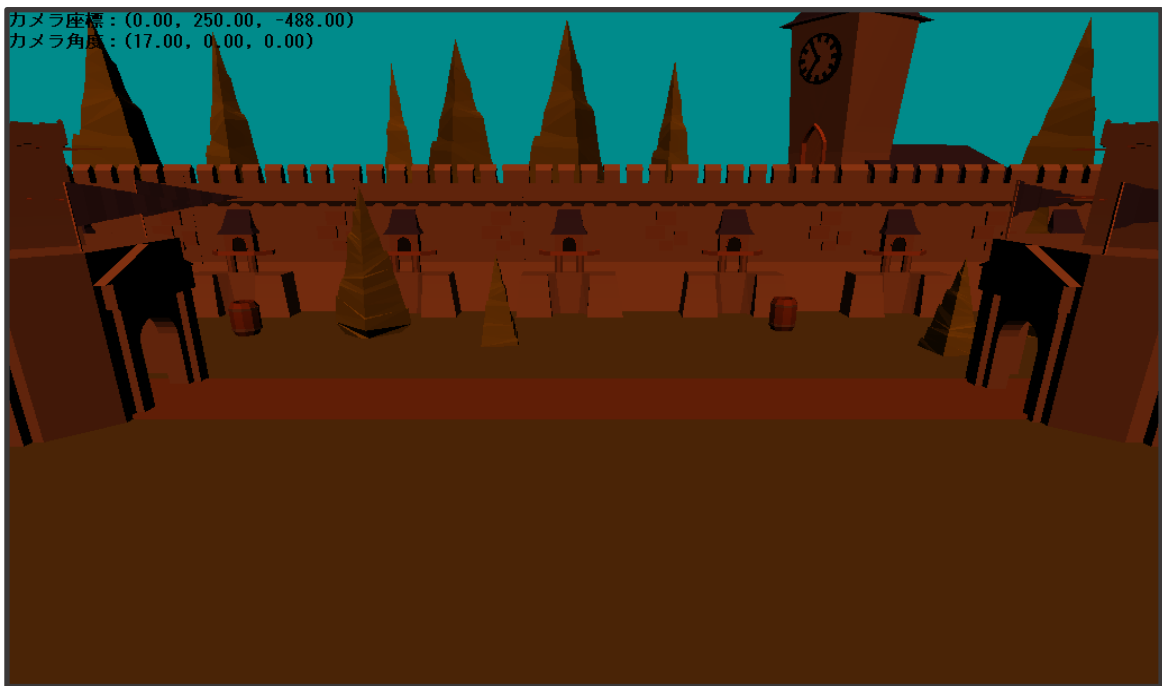
`SetLightDifColor`、`SetLightSpColor`、`SetLightAmbColor`

詳細は、CG概論で習ってください。

拡散光が、最も色が出やすいですので、
ディレクショナルライトで設定した上で、下記の設定を行うと

// ディフューズカラー

`SetLightDifColor(GetColorF(0.8f, 0.3f, 0.1f, 1.0f));`

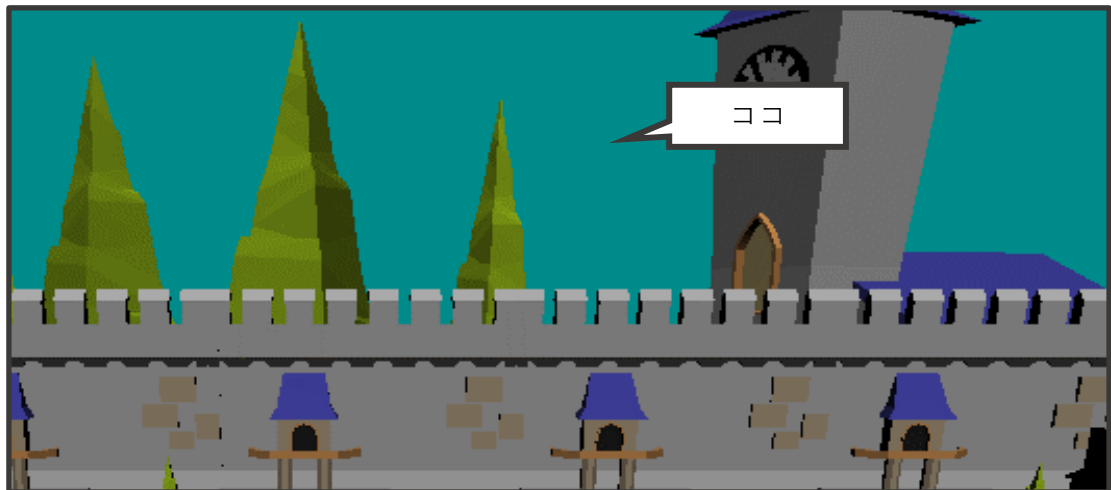


何となく夕焼け色になりますね。

最後にフォグ(霧)について、紹介します。

今回のステージは壁に覆われていますので、余り目立ちませんが、

3Dオブジェクトが無い部分は、背景色がそのまま描画されています。



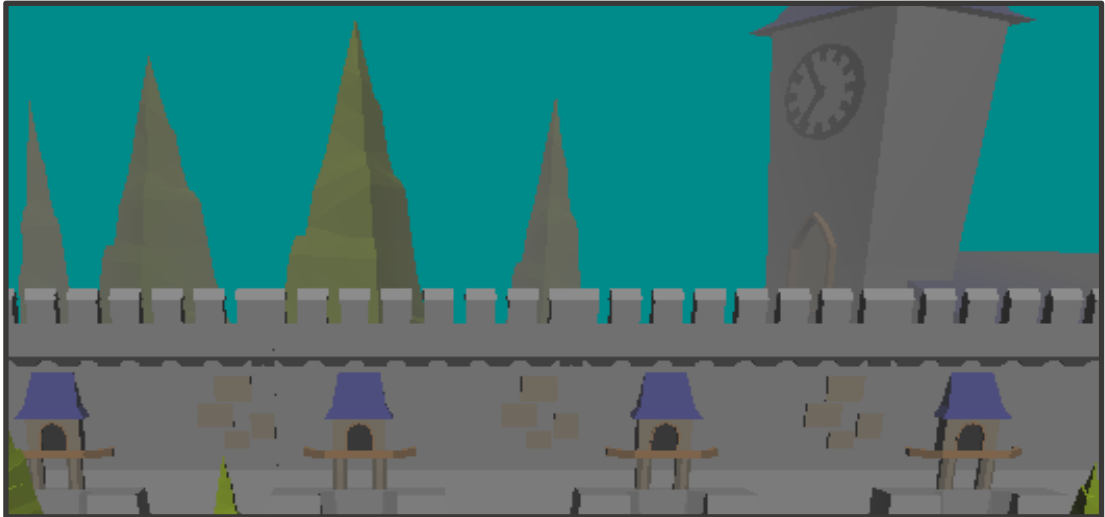
今回のゲームステージだと、背景画像を貼り付けるとそれっぽく見えますが、



広大な3Dワールドを歩き回るようなゲームの場合は、そうはいきません。

遠くの山や建物のステージを作っておかないと、見栄えが悪くなりますが、
フォグ(霧)で遠くが見えない、という風にすると、多少はマシになります。

```
// フォグ設定
SetFogEnable(true);
// フォグの色
SetFogColor(100, 100, 100);
// フォグを発生させる奥行きをの最小、最大距離
SetFogStartEnd(500.0f, 1500.0f);
```



マシになると言ったものの、そうでもないですね。

まだまだ、序の口で、3Dの技術はたくさんあります。
ほとんどDxLibのチカラを借りて、実装してきておりますが、
これをDirectXなどで、自力で作れるようになると、かなり技術が
上がります。

ゲームエンジンを利用して、色々な表現方法・技術を知って、
DxLib上なり、DirectX上で実装していくと、勉強もやり易いかと思います。