

◆ 導入編

導入に関しては難しい点はないが、DXライブラリの特定のバージョンを使いたい時などは注意が必要。

- ① 公式サイトより、Effekseer For DXライブラリをダウンロードする。
ゲームエンジンおよびゲームライブラリごとに異なるので注意。
- ② ダウンロードしたら、展開する。
- ③ 展開したら、中にある『プロジェクトに追加すべきファイル_VC用』の中を、DXライブラリがインストールされている場所に追加する。
DXライブラリと一緒にしたくない場合は、別のフォルダでもいいが、それについてはここでは解説しない。
上書き確認が出るが、特定のDXライブラリを使いたい場合を除き、上書きを行う。

◆ 初期化編

使い方としては、エフェクトを読み込み、その際に取得できるIDで、再生を実行し、座標をセットする。
あとは、生成されたパーティクルの位置情報の更新と、描画関数を呼び出す。
雰囲気的にはDXライブラリで画像を読み込んで表示するのと、さほど違いはない。

順に解説をしていく。

まず、DxLib_Init()の前に、次の行を追加する。

```

145 // DirectX11を使用するようにする。(DirectX9も可)
146 SetUseDirect3DVersion(DX_DIRECT3D_11);
147
148 if (DxLib_Init() == -1)
149 {
150     return false;
151 }
152

```

その次に、ウィンドウとフルスクリーンのモード切替対応部分をDxLib_Init()以降に追加する。

```

155 if (Effekseer_Init(8000) == -1)
156 {
157     DxLib_End();
158     return false;
159 }
160 // フルスクリーンウィンドウの切り替えでリソースが消えるのを防ぐ。
161 SetChangeScreenModeGraphicsSystemResetFlag(FALSE);
162
163 // DXライブラリのデバイスロストした時のコールバックを設定する。
164 // ウィンドウとフルスクリーンの切り替えが発生する場合は必ず実行する。
165 Effekseer_SetGraphicsDeviceLostCallbackFunctions();
166

```

最後に、effekseerのモードに合わせた初期化を行う。

今回は2Dモードで初期化を行う。

```

170 // Effekseerに2D描画の設定をする。
171 Effekseer_Set2DSetting(ScreenSize.x, ScreenSize.y);
172 SetUseZBuffer3D(TRUE);
173 SetWriteZBuffer3D(TRUE);
174

```

これで、初期化は完了だ。

◆ 読み込み編

読み込みたいエフェクトを指定し、読み込むだけでよい。

第2引数は拡大率になる。

戻り値はエフェクトを再生する際に必要になる。

```
65 | | effectResourceHandle = LoadEffekseerEffect("effect/blast.efk", 5.0f);
```

構造的に、画像マネージャーと同じようにエフェクトマネージャーを使って管理すると、楽だろう。

なお、読み込んだエフェクトは、不要になったら、次のようにして、破棄を行おう。

```
73 | | DeleteEffekseerEffect(effectResourceHandle);
```

◆ 再生→更新→表示編

再生については、まず再生したいエフェクトを指定し、その戻り値に座標をセットする形になる。

```
12 | | auto flag = PlayEffekseer2DEffect(effectID);  
13 | | SetPosPlayingEffekseer2DEffect(flag, pos.x, pos.y, 0);
```

更新処理は、Update()系のどこかに書こう。

```
106 | | UpdateEffekseer2D();
```

最後に描画だが、これも描画のどこかで書こう。

```
84 | | DrawEffekseer2D();
```

以上で、表示されるはずだ。