

Effekseer (エフェクシアー) を使おう

カッコいいエフェクトをゲーム内に表示させたい！ ですが、エフェクト自体は、基本、デザイナーが作成します。

でも、ゲーム制作で最低限のエフェクトは使用していきたいですし、プログラマーは、デザイナーに作って貰ったエフェクトをプログラムで制御していくこともあります。

ということで、あまり労力をかけずにエフェクトを作成して、制御に慣れていきましょう。

【Effekseerの導入】

<https://effekseer.github.io/jp/>

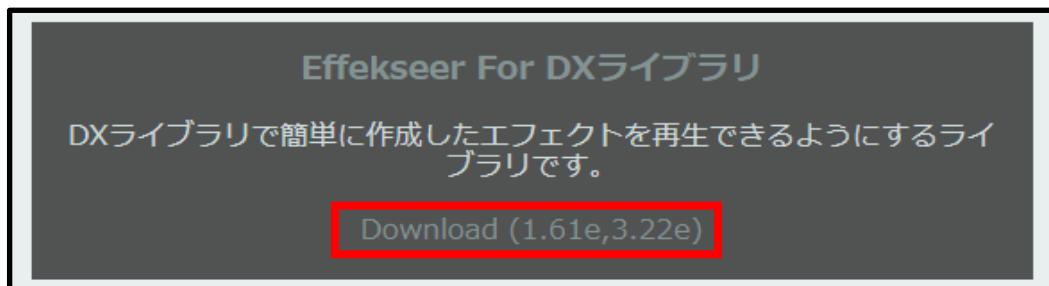
エフェクトを作成するソフトと、それをDxLibで使用するためのライブラリをダウンロードしましょう！

① エフェクトを作るためのソフト



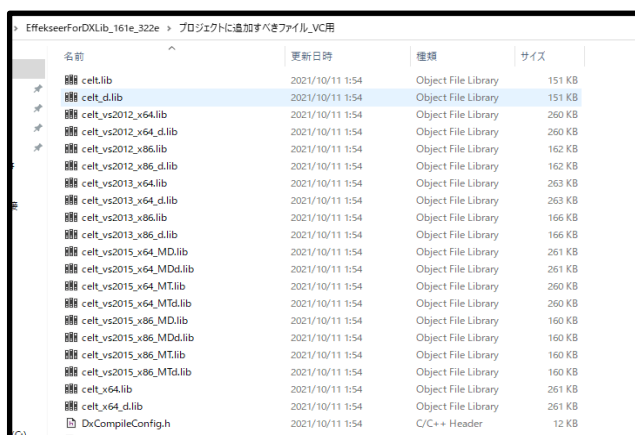
エフェクトを作りたい方は、
こちらをダウンロード

② EffekseerをDXライブラリで使用するためのライブラリ

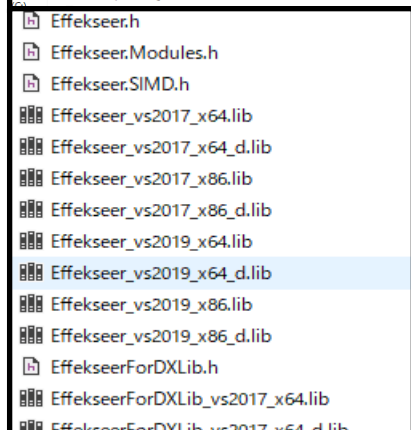


ダウンロードすると、Dxライブラリも含めた、Effekseerのライブラリがフォルダの中に入っていますので、Dxライブラリを置いているパスにコピーします。

もし、Dxライブラリのバージョンを変更したくなければ、以前のバージョンから、一致するバージョンを探しましょう。



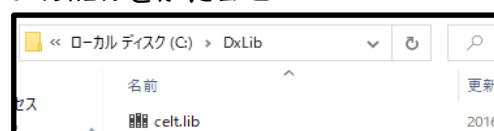
Dxライブラリ



Effekseerライブラリ



C:¥DxLibとかにコピー

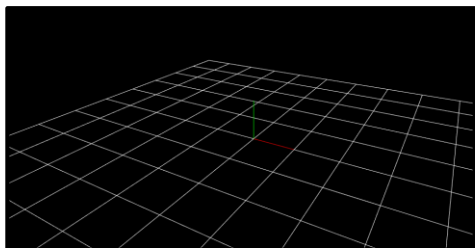
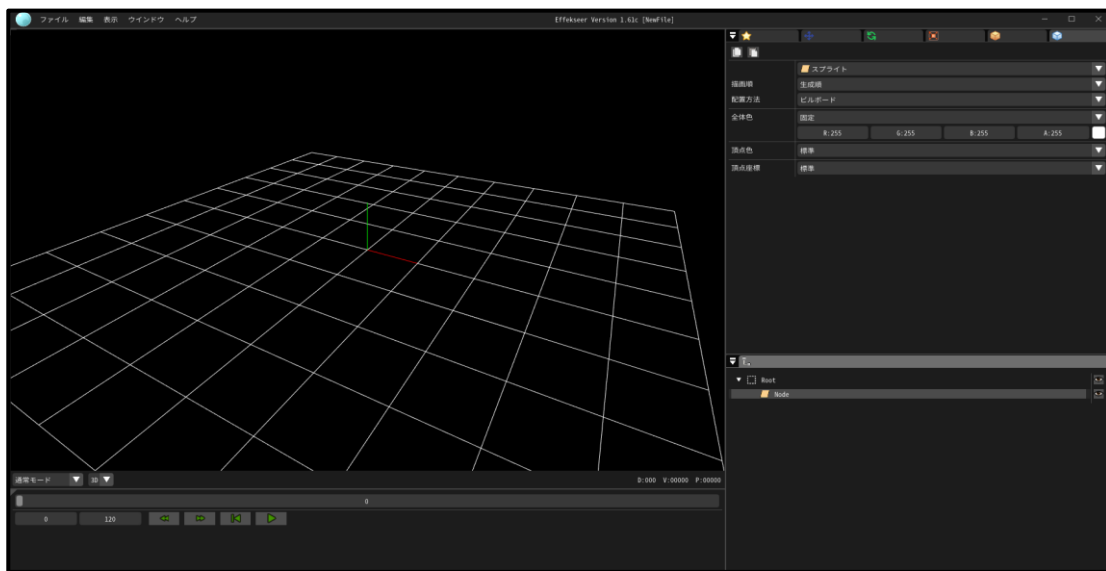


【超簡単なエフェクトを作ってみよう】

① Effekseerを起動

解凍フォルダの中に、

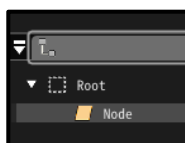
“Effekseer¥Tool¥Effekseer.exe” があるので起動。



プレビュー画面



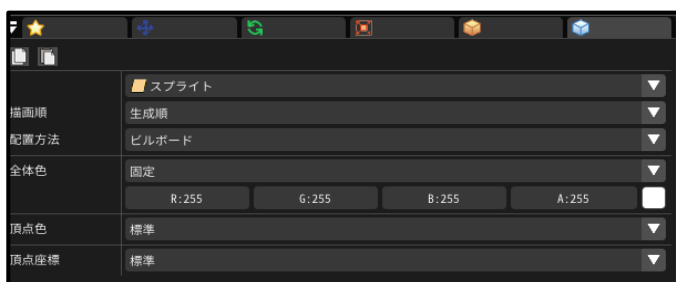
エフェクトの再生ボタン



エフェクトノード。

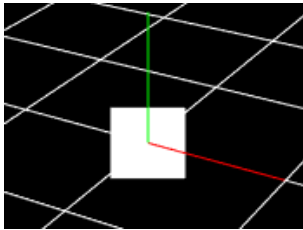
これをまずは作って、下図の設定を行う。

回転や生成条件で親子関係を作れるのが便利。

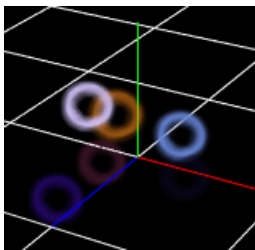


エフェクトの設定

② まず、再生してみよう



再生ボタンを表示すると、
白い四角形が表示されました。



この状態から、カラフルなリングが
ランダムに表示されるエフェクトを
作っていきたいと思います。

③ 位置をランダムにする

画面右下のNodeを選択します。

エフェクト設定画面の青色の十字矢印を選択します。

コンボボックスの中身を、「位置・速度・加速度」に変更します。



この設定の中の位置の振幅の値を、
XYZ、それぞれ「1」に設定します。

位置の振幅が「1」ということは、
生成する際の座標が、 $X(-1 \sim 1)$ 、 $Y(-1 \sim 1)$ 、 $Z(-1 \sim 1)$ のランダムになる、
ということになりますので、ランダムに表示されるようになったと思います。

④ 四角形をたくさん表示する

エフェクトを1つの粒子と見立てて、大量に表示させ、
様々な表現を行うことを、パーティクルエフェクトと呼びます。



今度は、エフェクト設定画面の★のマークを選択して、
生成数を20に設定します。

20の粒子が生成されるようになりました。

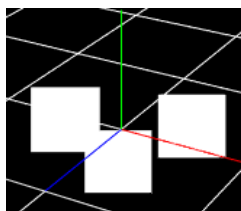
この粒子は、一度生成されたら、ずっと描画され続けていますので、
生存時間を短くして、粒子が消えるようにしていきます。

生存時間を「1.5」に設定します。

少しずつ消えるようになりました。

次に『徐々に』粒子が生成されるようにしていきます。

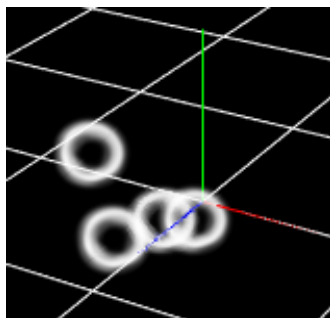
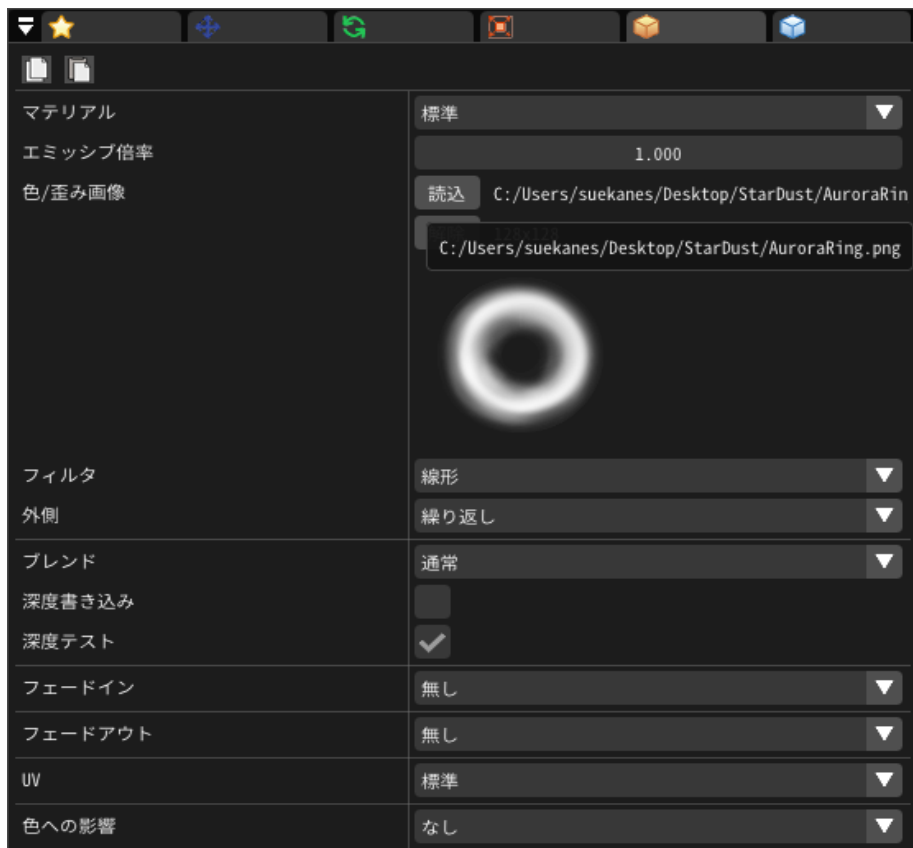
徐々に、粒子の数が増えるように、
生成時間の中心を「5」に設定して、5フレームごとに生成される
ようにしていきます。



120フレーム中、常に粒子が生成されるよう
になりました。

⑤ 表示する画像を変える

エフェクト設定画面の黄色い四角を選択して、表示する画像を変更します。



表示される画像が変わりました。

段々と透明化していきながら消えるように、
フェードアウトを「有り」にして、フレーム数を「10」に設定します。

⑥ 色をランダムに変える

エフェクト設定画面の青い四角を選択します。



全体色をランダムに変更して、
色の最小設定のRGBをゼロ(黒色)にします。
(α は255のまま)

⑦ ファイルを保存する

これで、オリジナルのエフェクトを作成することができました。
凝ったエフェクトを作成したければ、

Effekseer¥Sample

の中に、汎用的に使用できる画像や、
Effekseerの設定がたくさんありますので、
そちらを参考に制作してみましょう。

但し、私達はプログラマーを目指していきますので、
エフェクト制作は、程々にして、
プログラミング技術向上に時間をかけていきましょう。

【エフェクトをDxLibで使用する】

Effekseerで作成したエフェクトの再生

Effekseerは、編集ファイルと実行ファイルが同じですので、専用フォーマットにエクスポートする作業は必要ありません。拡張子は、「efkefc」です。

このファイルを再生するために、いくつか手順が必要です。

① 初期化を行う

```
void SceneManager::InitEffekseer(void)
{

    // Effekseerの初期化
    SetUseDirect3DVersion(DX_DIRECT3D_11);

    if (Effekseer_Init(8000) == -1)
    {
        DxLib_End();
    }

    SetChangeScreenModeGraphicsSystemResetFlag(FALSE);

    Effekseer_SetGraphicsDeviceLostCallbackFunctions();

}
```

② カメラとの同期を行う

```
void Camera::SetBeforeDraw(void)
{

    // クリップ距離を設定する(SetDrawScreenでリセットされる)
    SetCameraNearFar(10.0f, 30000.0f);

}
```



```

switch (mMode)
{
case Camera::MODE::FIXED_POINT:
    SetBeforeDrawFixedPoint();
    break;
case Camera::MODE::FOLLOW:
    SetBeforeDrawFollow();
    break;
}

// カメラの設定
SetCameraPositionAndTargetAndUpVec(
    mPos,
    mTargetPos,
    mCameraUp
);

// DXライブラリのカメラとEffekseerのカメラを同期する。
Effekseer_Sync3DSetting();

}

```

③ Effekseerの更新、描画処理を行う

```

// 描画先グラフィック領域の指定
// (3D描画で使用するカメラの設定などがリセットされる)
SetDrawScreen(DX_SCREEN_BACK);

// 画面を初期化
ClearDrawScreen();

// カメラ設定
mCamera->SetBeforeDraw();

// Effekseerにより再生中のエフェクトを更新する。
UpdateEffekseer3D();

// 描画
mScene->Draw();
mCamera->Draw();

```

```
// Effekseerにより再生中のエフェクトを描画する。  
DrawEffekseer3D();
```

```
// 最後  
mFader->Draw();
```

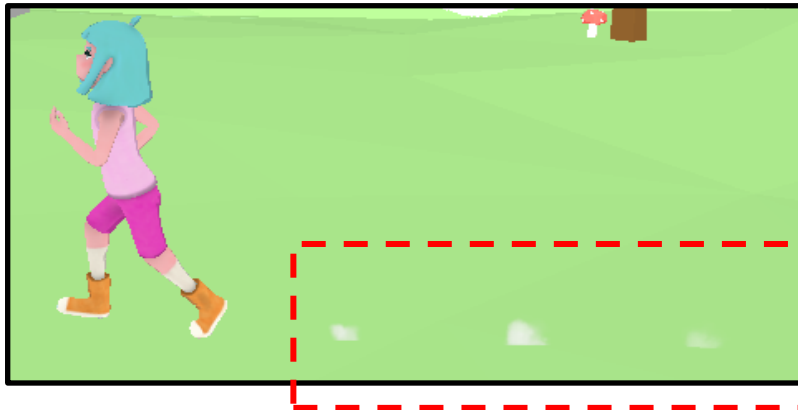
④ Effekseerの後始末をする

```
bool Application::Release(void)  
{  
    mSceneManager.Release();  
  
    // Effekseerを終了する。  
    Effkseer_End();  
  
    DxLib_End();  
  
    return true;  
}
```

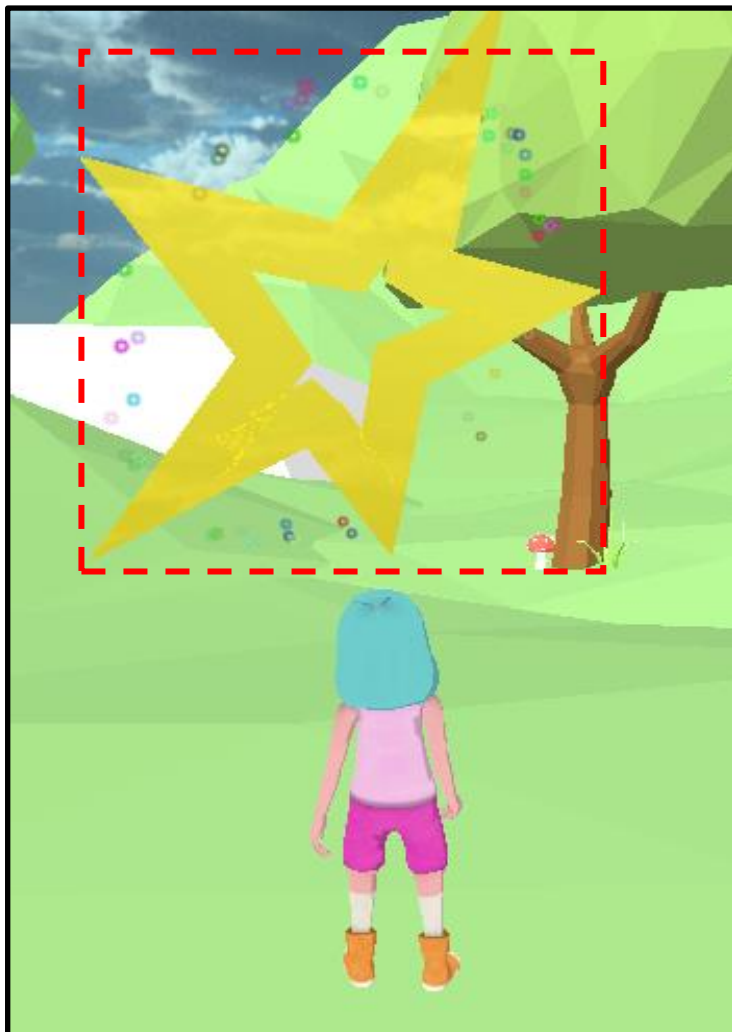
⑤ いよいよ再生する

```
// LoadGraphと同じ感じでロード  
mHandleId = LoadEffekseerEffect("Smoke.efkefc");  
  
~~中略~~  
  
// エフェクトを再生する。  
int playHandle = PlayEffekseer3DEffect(mHandleId);  
  
// エフェクトの大きさを変える  
SetScalePlayingEffekseer3DEffect(playHandle, 5.0f, 5.0f, 5.0f);  
  
// エフェクトの位置を変える  
SetPosPlayingEffekseer3DEffect(playHandle, pos.x, pos.y, pos.z);
```

こんな感じになります。



砂煙を定期的に
表示



キラキラを定期的に
表示