

FavLibrary Usage (v1.01)

1 Fav Library 環境構築

1.1 必要環境

- Xercesc (XML Parser) : <http://xerces.apache.org/>

1.2 Windows 環境での注意点 (FavLibrary.Win.sln を使用する場合)

- NuGet 経由でも「xercesc」を取得できますが、Windows や VisualStudio のバージョンが異なると、要求される xercesc バージョンも異なり、不整合が起きる場合があります。
そのため、上記 URL からお使いの環境に適する xercesc バージョンを取得のうえ、ビルド・配置・プロジェクト設定を行ってください。

2 Fav Library 使用方法

2.1 FavLibrary の include

手順：

お使いの環境の構成に合わせて、Fav.h を include します。

FavLibrary は namespace FavLibrary で定義されているため、using 宣言をしてください。

サンプルコード：

```
#include "YourEnvironment/FavLibrary/Fav.h"  
using namespace FavLibrary;
```

2.2 FAV ファイルの Load, および Save

手順：

- (1) 正しい FAV ファイルのパスを指定します。
- (2) 書き出し先の FAV ファイルパスを指定します。

サンプルコード：

```
Fav fav;  
fav.read(SourceFilepath); _____(1)  
fav.write(DestinationFilepath, FavLibrary::CompressionMode::base64); _____(2)
```

解説：

- (1)では、Load する FAV ファイルパスを指定します。Load 時に、FAV ファイルフォーマットのスキーマ (fav.xsd)によって Validation が行われます。(FavReader::validation)
- (2)では、Save する FAV ファイルパスと、圧縮方法 (FavLibrary::CompressionMode) を指定します (zlib 圧縮は未実装)。FAV ファイル内の voxel_map.layer, color_map.layer のみ指定の方法で圧縮されます (link_map は未実装)。それ以外は Readable な XML で記述されます。

2.3 FAV ファイルの新規作成

手順：

【FAV ファイルの準備】

- (1) FAV クラスのインスタンス (fav1) を生成します。
- (2) fav1 に **Metadata** を保存します。
- (3) FAV データで使用する **Geometry** (geometry1) を定義し、fav1 の **palette** に **geometry1** を登録します。
- (4) FAV データで使用する **Material** (material1) を定義し、fav1 の **palette** に **material1** を登録します。

【三次元形状を記述するための準備】

- (5) 三次元形状の構成要素となる **Voxel** (voxel1) を定義し、fav1 に **voxel1** を登録します。
- (6) 三次元形状を構成する **Voxel** を配置するための **Grid** 空間 (grid1) を定義します。
- (7) **Voxel** で構成される三次元形状の実態となる **Structure** クラスのインスタンス (structure1) を生成し、以下の初期化を行います。
 - **Voxel** 記述方法 (**BitPerVoxel**) を指定します。
 - 本インスタンス (structure1) で管理する **Grid** インスタンス (grid1) のポインタを登録します。
 - **VoxelMap** を初期化します。
- (8) ボクセルに色を指定したい場合、色の記述方法 (**ColorMode**) を指定して **ColorMap** を初期化します。

(色数が少ない場合は、**ColorMap** の代わりに、色を定義した **Voxel** (**Voxel.display**) も使用できます。)

【実際の三次元形状の記述(ボクセルの配置)】

- (9) 現在の **Grid** 位置にボクセルを配置する際には、fav1 に登録済みの **Voxel** の **ID** を指定します。
- (10) 配置したボクセルに色を指定したい場合、ボクセルを配置したのと同じ **Grid** 位置に、(8) で指定した色の記述方法 (**ColorMode**) に従って色を指定します。
- (11) 現在の **Grid** 位置にボクセルが存在しない場合には、**0** を指定します。

【記述した三次元形状を FAV ファイル化】

- (12) 実際の FAV データを表す **Objecct** のインスタンス (object1) を生成し、**Voxel** を配置する **Grid** 空間 (grid1) と、実際に **Voxel** を配置した **Structure** (structure1) を登録します。
- (13) fav1 に **object1** を登録します。
- (14) fav1 を書き出します。

サンプルコード：

```
Fav fav = Fav(); (1)

fav.setMetadataId("id"); (2)
fav.setMetadataAuthor("author");
fav.setMetadataTitle("title");
fav.setMetadataLicense("license");
fav.setMetadataNote("note");

Geometry geometry1 = Geometry(1, "geometry1"); (3)
geometry1.setScale(1, 1, 1);
geometry1.setShape(FavLibrary::GeometryShape::cube);
fav.palette.addGeometry(geometry1);

Material material1 = Material(1, "material1"); (4)
material1.addMaterialName("soft_mat1");
fav.palette.addMaterial(material1);

Voxel voxel1 = Voxel(1, "voxel1"); (5)
voxel1.addMaterialInfo(material1.getId(), 1.0);
voxel1.setGeometryInfo(GeometryInfo(geometry1.getId()));
fav.addVoxel(voxel1);

Grid grid1 = Grid(); (6)
grid1.setUnit(1, 1, 1);
grid1.setOrigin(25.3, -3.45, 65.2);
grid1.setDimension(width, depth, height);

Structure structure1 = Structure(); (7)
structure1.setBitPerVoxel(FavLibrary::BitPerVoxel::Bit8);
structure1.setGridPointer(&grid1);
structure1.initVoxelMap();

structure1.initColorMap(FavLibrary::ColorMode::RGB); (8)

for (int z = 0; z < grid1.getDimensionZ(); z++)
    for (int y = 0; y < grid1.getDimensionY(); y++)
        for (int x = 0; x < grid1.getDimensionX(); x++)
        {
            if (YourVoxelSource[z][y][x] != 0)
            {
                structure1.setVoxel(x, y, z, voxel1.getId()); (9)

                ColorRGBA rgba = ColorRGBA(YourVoxelColor); (10)
                structure1.setColor(x, y, z, rgba);

            }
            else
                structure1.setVoxel(x, y, z, 0); (11)
        }

Object object1 = Object(1, "object1"); (12)
object1.grid = grid1;
object1.structure = structure1;

fav.addObject(object1); (13)

fav.write(DestinationFilepath, FavLibrary::CompressionMode::base64); (14)
```

解説：

- ・ (5)で Voxel を定義する際は、fav1 の palette に登録してある Geometry, Material の ID を指定してください。
- ・ (7)では、1Voxel を何 bit で記述するかを表す FavLibrary::BitPerVoxel を指定します。ボクセルに色を指定する場合、色のフォーマットを表す FavLibrary::ColorMode を指定します。
- ・ (9)で Grid に配置する Voxel は、fav1 に登録してある Voxel の ID を指定してください。
- ・ (10)で指定する色情報は、(7)で指定した色のフォーマット (FavLibrary::ColorMode) 通りの色情報を指定してください。色は、Voxel を配置したときだけ指定してください。
- ・ (14)では、Save する FAV ファイルパスと、圧縮方法 (FavLibrary::CompressionMode) を指定します (zlib 圧縮は未実装)。FAV ファイル内の voxel_map.layer, color_map.layer のみ指定の方法で圧縮されます (link_map は未実装)。それ以外は Readable な XML で記述されます。

3 変更履歴

2017/05/29 (v1) : 初版発行

2017/08/14 (v1.01) :

- ・ 変更履歴を巻頭から巻末へ移動
- ・ 1.2 Windows 環境での `xercesc` バージョンについて `NuGet` を使用しない旨の説明に変更
- ・ 3.3 サンプルコードと説明を修正 (`setGridPointer`, `initVoxelMap`, `initColorMap`)