

FavLibrary Usage (v1)

1 変更履歴

2017/05/29 (v1) : 初版発行

2 FAV Library 環境構築

2.1 必要環境

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

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

- NuGet 経由で取得できる最新 (安定版) は「xercesc. 3. 1. 1」ですが, Windows10 環境では「xerces-c-3. 1. 4」が要求されるため, 上記 URL から取得ののち, 「./packages」(.sln 同階層¥packages) に配置してください.
- 「xerces-c-3. 1. 4」にパスを通すため, NuGet で「xercesc. 3. 1. 1」を構成に含めてしまうのが楽です.
- 現状の構成では「xerces-c-3. 1. 4」をリンクしています. 他のバージョンをリンクする場合, プロジェクトの構成で設定を変更してください.

3 FAV Library 使用方法

3.1 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;
```

3.2 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.

3.3 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、Voxel 記述方法を指定します。ボクセルに色を指定したい場合、色の記述方法を指定します。

【実際の三次元形状の記述】

- (8) 実際にボクセルを配置していきます。
- (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、(10) と同様に色情報を指定します。
- (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.setColorMode(FavLibrary::ColorMode::RGBA);

for (int z = 0; z < grid1.getDimensionZ(); z++) _____ (8)
    for (int y = 0; y < grid1.getDimensionY(); y++)
        for (int x = 0; x < grid1.getDimensionX(); x++)
            {
                if (YourVoxelSource[z][y][x] != 0) _____ (9), (10)
                {
                    structure1.setVoxel(x, y, z, voxel1.getId());

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