

```
#include <string>
#include <vector>
#include <fstream>
#include "DxLib.h"
#include "Utility.h"
#include "GameScene.h"
#include "Stage.h"

Stage::Stage(GameScene* scene)
{
    mGameScene = scene;
}

void Stage::Init(int stageNo)
{
    LoadDivGraph(
        "Image/StageChip.png",
        4,
        2, 2,
        BLOCK_SIZE, BLOCK_SIZE,
        &mImages[0], true);

    // オブジェクト無し(芝生)
    mImageOutFloor = mImages[0];

    // 床
    mImageInFloor = mImages[1];

    // 内壁
    mImageInWall = mImages[2];

    // 外壁
    mImageOutWall = mImages[3];

    // コードでステージを設定する
    //SetStage(stageNo);

    // 外部データを使用して、ステージを設定する
    LoadData(stageNo);
}

void Stage::Update(void)
{
}

void Stage::Draw(void)
```

```
{

    // ゲームエリアの描画
    int mapNo;
    int tmpImage = -1;

    for (int y = 0; y < MAP_SIZE_Y; y++)
    {
        for (int x = 0; x < MAP_SIZE_X; x++)
        {

            mapNo = mMap[y][x];
            switch (mapNo)
            {
            case -1:
                // オブジェクト無し(芝生)
                tmpImage = mImageOutFloor;
                break;
            case 1:
                // 床
                tmpImage = mImageInFloor;
                break;
            case 2:
                // 内壁
                tmpImage = mImageInWall;
                break;
            default:
                break;
            }

            DrawGraph(
                GAME_AREA_X + (x * BLOCK_SIZE),
                GAME_AREA_Y + (y * BLOCK_SIZE),
                tmpImage, true);

        }
    }

    // ゲームエリア外側の描画
    int scSizeX = SCREEN_SIZE_X / BLOCK_SIZE;
    int scSizeY = SCREEN_SIZE_Y / BLOCK_SIZE;

    // 横
    for (int x = 0; x < scSizeX; x++)
    {
        DrawGraph(
            x * BLOCK_SIZE, 0 * BLOCK_SIZE,
            mImageOutWall, true);
    }
}
```



```
std::vector<std::string> strvec =
    Utility::Split(line, ',');

// Xで分割されたstrvec
int size = strvec.size();
for (int x = 0; x < size; x++)
{
    mMap[y][x] = stoi(strvec[x]);
}

y++;
}
}
```