

最終課題

- 📌 [課題] OpenGLを使ってオセロのCGプログラムを制作する
 - 📌 オセロのCGを描画するプログラムを作成する
 - 📌 オセロのAI部分は作成しない
 - 📌 オセロの対局の1試合分のCGアニメーションを作成する

ファイル形式

- 5 この授業ではオセロの表示が目的であり、オセロのAIは作らない
- 5 オセロのAIは別のソフトで実装し、その対戦結果をファイルとして出力するものと仮定する
- 5 WebClassに対局ファイルがあるので、それを読み込んで表示すること
- 5 対局ファイルは独自の形式で、以下で説明する

バイナリ形式 エンディアンネスはIntel系CPUと同じエンディアン

ヘッダ	未定義	4バイト符号つき整数	1個	計4バイト	36バイト
	未定義	1バイト符号つき整数	16個	計16バイト	
	未定義	1バイト符号つき整数	16個	計16バイト	
データ	⋮				260×Nバイト
	盤面情報	4バイト符号つき整数	64個	計256バイト	
	終了判定	4バイト符号つき整数	1個	計4バイト	
	⋮				
フッタ	未定義	4バイト符号つき整数	1個	計4バイト	8バイト
	未定義	4バイト符号つき整数	1個	計4バイト	

Nを手順とする(初期状態を含めて通常は61)(61以外もありうる)

「終了判定」の値は終了時に-1となりそれ以外は10または11とする

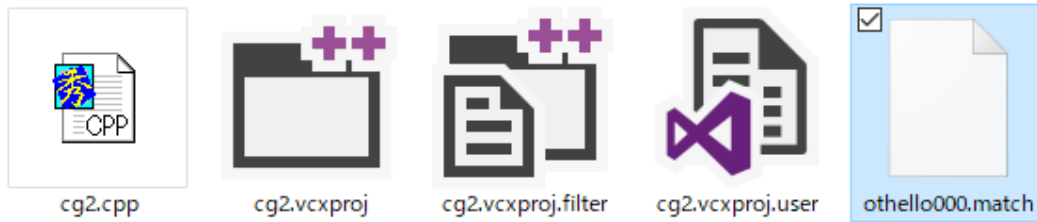
盤面情報は左上のマスから右下のマスの順に並んでいる
左上から右に1マスずつ進み、右端に到達したのち、1行下の左端に行くものとする

- 1 黒の石が置かれている
- 1 白の石が置かれている
- 0 石が置かれていない

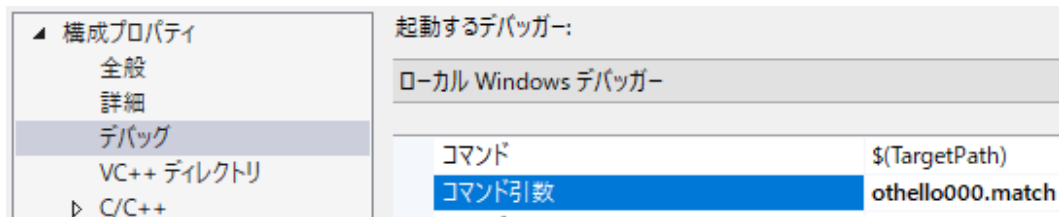
[0][0]	[0][1]	[0][2]	[0][3]	[0][4]	[0][5]	[0][6]	[0][7]
[1][0]	[1][1]	[1][2]	[1][3]	[1][4]	[1][5]	[1][6]	[1][7]
[2][0]	[2][1]	[2][2]	[2][3]	[2][4]	[2][5]	[2][6]	[2][7]
[3][0]	[3][1]	[3][2]	[3][3]	[3][4]	[3][5]	[3][6]	[3][7]
[4][0]	[4][1]	[4][2]	[4][3]	[4][4]	[4][5]	[4][6]	[4][7]
[5][0]	[5][1]	[5][2]	[5][3]	[5][4]	[5][5]	[5][6]	[5][7]
[6][0]	[6][1]	[6][2]	[6][3]	[6][4]	[6][5]	[6][6]	[6][7]
[7][0]	[7][1]	[7][2]	[7][3]	[7][4]	[7][5]	[7][6]	[7][7]

コマンドライン引数

- ✖ コマンド引数にファイル名を指定して実行しないといけない
- ✖ まず、配布したデータファイル(以降othello000.matchと呼ぶことにする)をcppファイルのあるディレクトリにコピーする



- ✖ Visual Studioでslnファイルを開いて、プロジェクトのプロパティを開く
- ✖ [構成プロパティ]-[デバッグ]-[コマンド引数]でothello000.matchと書く



授業では扱っていない内容

- OpenGLは巨大なライブラリであり、授業で扱っていない内容は多いです
 - 最低限必要な内容のみを厳選して授業で扱っています
- この授業では、`glutTimerFunc`を使ってアニメーションを実現しましたが、`glutIdleFunc`を使ったものもあります
- `glutTimerFunc`によるタイマーは複数作ることができます
- `glutTimerFunc`で時間を操作するため、スリープは不要なため、授業では使いませんでした
 - スリープの使い所がないですが、使っても構いません

```
#include <windows.h>
Sleep(1000);
```
- 円盤の形状は $(\cos\theta, \sin\theta)$ を利用して計算することも出来ますし、`gluDisk`関数というものもあります
- 円柱の側面は $(\cos\theta, \sin\theta)$ を利用して計算することも出来ますし、`gluCylinder`関数というものもあります
- `glutSolidSphere`以外に`glutSolidCube`や`glutSolidCone`などもあります
- 授業では透視投影`gluPerspective`をやりましたが、正投影は`glOrtho`を使います
- 授業では`glFlush`を使わず、`glutSwapBuffers`だけを使いました
- 授業では`glVertex3d`などを使い、`glVertex3dv`などは使いませんでした
- 授業では点光源のみを扱いましたが、スポットライト光源などの光源もあります
- `glScaled`と`glutSpecialFunc`や`glutMotionFunc`については別途資料を参照

Tips

- テクスチャマッピングをおこなう画像は24bit形式**bmp**ファイルでなければいけない
 - `jpg`や`png`は[ペイント]などのソフトで24bit形式**bmp**ファイルに変換すること
- テクスチャマッピングをおこなう画像のサイズは2の累乗でなければいけない
 - 画像のサイズは256×256や512×256などの適切なサイズになるように[ペイント]などのソフトで切り取ったり拡大縮小したりすること
- オセロ対戦ファイルとテクスチャ画像ファイルのように、複数のファイルを読み込む場合
 - 例えば、[構成プロパティ]-[デバッグ]-[コマンド引数]で`othello000.match dice.bmp`と入力した場合、`argv[1]`が`othello000.match`になり、`argv[2]`が`dice.bmp`になる
- 2Dテキストの表示は別途資料を参照

画面に何も表示されない

- Q. 画面に何も表示されません
A. まずは、画面に何か表示される状態にしましょう
- 画面に何か表示されていれば何が問題か目で見て確認できます
 - 「形がおかしい」「色がおかしい」「位置がおかしい」「動きがおかしい」...

FAQ

- 動作が遅い
 - `glutSolidSphere(1.0, 100, 100)`だとポリゴン数が多すぎるので`glutSolidSphere(1.0, 10, 10)`ぐらいに減らす(ただの球に10000ポリゴンは無駄すぎる)(球ごとき、100ポリゴンで十分だろ)

プロトタイプ・その1

```
#include <stdlib.h>
#include <stdio.h>
#include <GL/glut.h>

int winw, winh;
int boardnow[8][8];

void myGround()
{
    float ground[2][4] = {
        {0.6f, 0.6f, 0.6f, 1.0f},
        {0.3f, 0.3f, 0.3f, 1.0f}
    };
    int i, j;
    int x, y;

    glBegin(GL_QUADS);
    glNormal3d(0.0, 0.0, 1.0);
    for (i = 0; i < 8; i++) {
        for (j = 0; j < 8; j++) {
            y = i - 4;
            x = j - 4;
            glMaterialfv(GL_FRONT, GL_DIFFUSE, ground[abs(i + j) % 2]);
            glVertex3d((double)x, (double)y, 0.0);
            glVertex3d((double)(x + 1), (double)y, 0.0);
            glVertex3d((double)(x + 1), (double)(y + 1), 0.0);
            glVertex3d((double)x, (double)(y + 1), 0.0);
        }
    }
    glEnd();
}
```

```
void myDisplay()
{
    float white[4] = { 0.8f, 0.8f, 0.8f, 1.0f };
    float black[4] = { 0.2f, 0.2f, 0.2f, 1.0f };
    int i, j;
    double posy, posx;
    char text[256];
    char* p;

    glClearColor(0.0f, 0.5f, 0.0f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60.0, (double)winw / (double)winh, 0.1, 100.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0.0, -5.0, 10.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

    myGround();

    for (i = 0; i < 8; i++) {
        for (j = 0; j < 8; j++) {
            if (boardnow[i][j] != 0) {
                posy = (double)i - 4.0 + 0.5;
                posx = (double)j - 4.0 + 0.5;
                glPushMatrix();
                glTranslated(posx, posy, 0.5);
                if (boardnow[i][j] > 0) glMaterialfv(GL_FRONT, GL_DIFFUSE, black);
                else glMaterialfv(GL_FRONT, GL_DIFFUSE, white);
                glutSolidSphere(0.5, 10, 10);
                glPopMatrix();
            }
        }
    }

    glDisable(GL_DEPTH_TEST);
    glDisable(GL_LIGHTING);
    glDisable(GL_LIGHT0);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    glColor3d(0.0, 0.0, 0.0);
    sprintf_s(text, "Final report");
    glRasterPos2d(-0.9, 0.9);
    for (p = text; *p; p++) {
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, *p);
    }

    glutSwapBuffers();
}

void myReshape(int width, int height)
{
    winw = width;
    winh = height;
    glViewport(0, 0, winw, winh);
}

void myKeyboard(unsigned char key, int x, int y)
{
    if (key == 0x1B) exit(0);
}
```

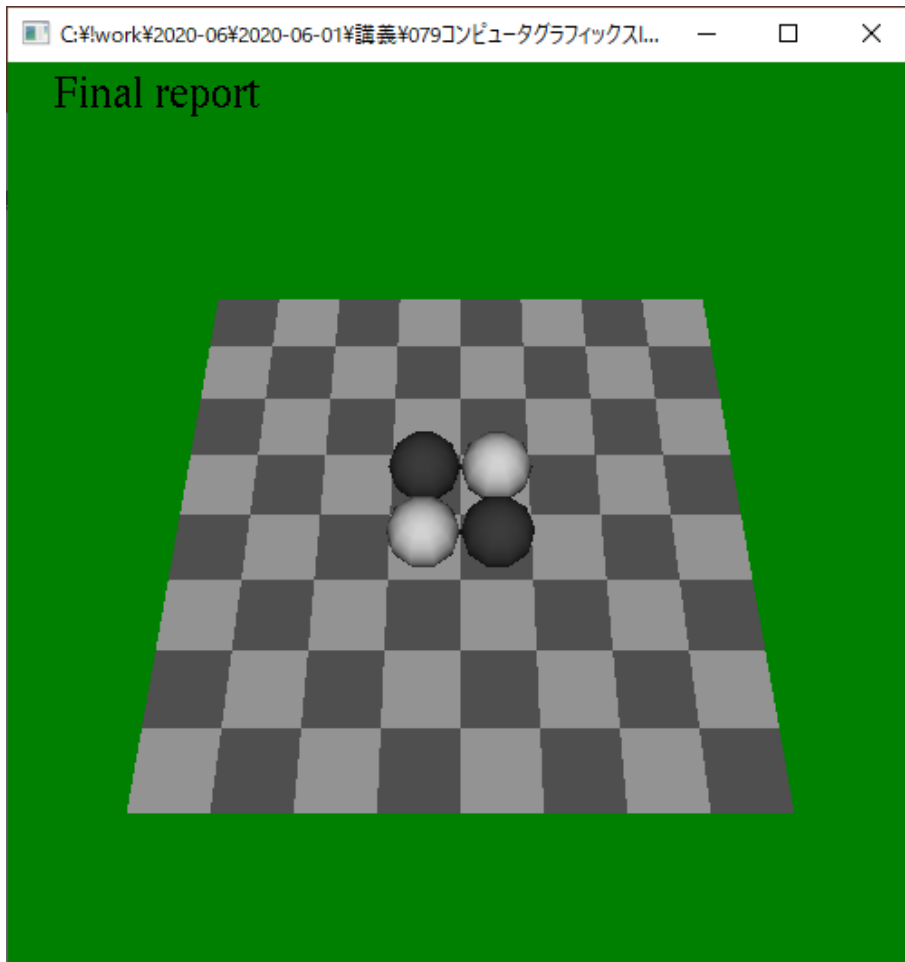
プロトタイプ・その1

```
void myInit(char* progame)
{
    glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEPTH);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(0, 0);
    glutCreateWindow(progame);
}

int main(int argc, char* argv[])
{
    int i, j;

    for (i = 0; i < 8; i++) {
        for (j = 0; j < 8; j++) {
            boardnow[i][j] = 0;
        }
    }
    boardnow[3][4] = boardnow[4][3] = 1;
    boardnow[3][3] = boardnow[4][4] = -1;

    glutInit(&argc, argv);
    myInit(argv[0]);
    glutKeyboardFunc(myKeyboard);
    glutReshapeFunc(myReshape);
    glutDisplayFunc(myDisplay);
    glutMainLoop();
    return 0;
}
```



盤面の情報を三次元で静止画表示するサンプルソースコード

プロトタイプ・その2

```

#pragma warning(disable:4996)
#include <stdlib.h>
#include <stdio.h>
#include <GL/glut.h>

const int delay = 500;
const int CYCLE = 5;
const int STATE_GAME = 1;
const int STATE_END = 2;
int state;
int boardnow[8][8];
int boardold[8][8];
int timestep;
FILE* fp;

void countStone(int* numb, int* numw)
{
    int i, j;
    *numb = 0;
    *numw = 0;
    for (i = 0; i < 8; i++) {
        for (j = 0; j < 8; j++) {
            if (boardnow[i][j] > 0) (*numb)++;
            if (boardnow[i][j] < 0) (*numw)++;
        }
    }
}

void myDisplay()
{
    int i, j;
    int numw, numb;
    char text[256];
    char* p;

    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3d(0.0, 0.0, 0.0);
    sprintf_s(text, "Please see the console window");
    glRasterPos2d(-0.9, 0.9);
    for (p = text; *p; p++) {
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, *p);
    }
    glutSwapBuffers();

    for (i = 0; i < 8; i++) {
        for (j = 0; j < 8; j++) {
            if (boardnow[i][j] > 0) {
                if (boardnow[i][j] == boardold[i][j]) printf("o");
                else printf("O");
            }
            else if (boardnow[i][j] < 0) {
                if (boardnow[i][j] == boardold[i][j]) printf("x");
                else printf("X");
            }
            else printf(".");
        }
        printf("\n");
    }
    printf("\n\n");

    countStone(&numb, &numw);
    printf("black %d, white %d\n", numb, numw);

    printf("timestep %d/%d, state %d\n\n", timestep, CYCLE, state);
}

```



```
void myTimer(int value)
{
    int i, j;
    int currentstate;

    if (value == 1) {
        timestep++;
        if (state == STATE_GAME) {
            if (timestep == CYCLE) {
                timestep = 0;
                for (i = 0; i < 8; i++) {
                    for (j = 0; j < 8; j++) {
                        boardold[i][j] = boardnow[i][j];
                        fread(&boardnow[i][j], sizeof(int), 1, fp);
                    }
                }
                fread(&currentstate, sizeof(int), 1, fp);
                if (currentstate == -1) {
                    state = STATE_END;
                    fclose(fp);
                }
            }
        }
        glutTimerFunc(delay, myTimer, 1);
        glutPostRedisplay();
    }
}

void myKeyboard(unsigned char key, int x, int y)
{
    if (key == 0x1B) exit(0);
}

void myInit(char* progname)
{
    glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(0, 0);
    glutCreateWindow(progname);
}
```

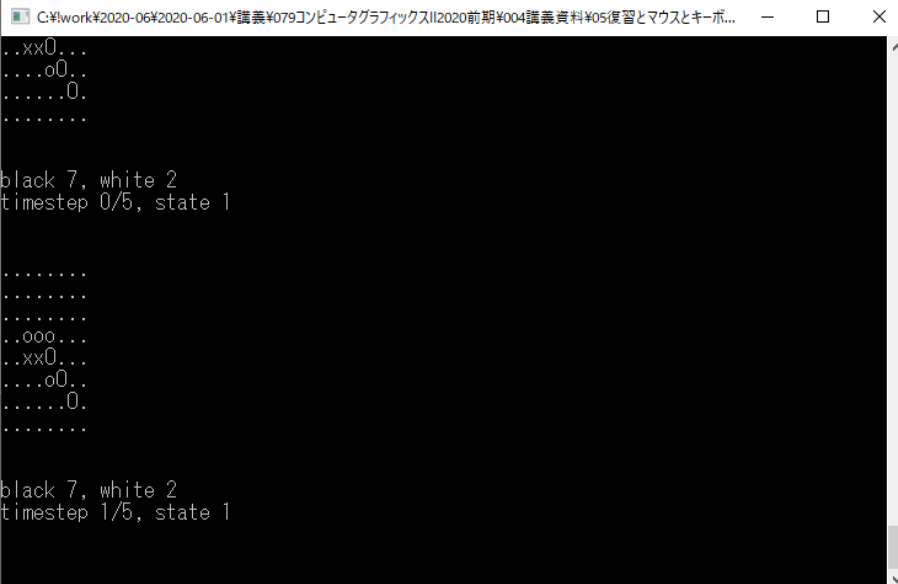
プロトタイプ・その2

```
int main(int argc, char* argv[])
{
    int i, j;
    int dummy1;
    char dummy2[16];

    if (argc <= 1) {
        printf("コマンド引数にファイル名が指定されていません\n");
        return 1;
    }
    fp = fopen(argv[1], "rb");
    if (fp == NULL) {
        printf("ファイル%sを開けません\n", argv[1]);
        return 1;
    }
    fread(&dummy1, sizeof(int), 1, fp);
    fread(dummy2, sizeof(char), 16, fp);
    fread(dummy2, sizeof(char), 16, fp);

    for (i = 0; i < 8; i++) {
        for (j = 0; j < 8; j++) {
            boardnow[i][j] = 0;
            boardold[i][j] = 0;
        }
    }
    boardnow[3][4] = boardnow[4][3] = 1;
    boardold[3][4] = boardold[4][3] = 1;
    boardnow[3][3] = boardnow[4][4] = -1;
    boardold[3][3] = boardold[4][4] = -1;
    state = STATE_GAME;
    timestep = 0;

    glutInit(&argc, argv);
    myInit(argv[0]);
    glutKeyboardFunc(myKeyboard);
    glutTimerFunc(delay, myTimer, 1);
    glutDisplayFunc(myDisplay);
    glutMainLoop();
    return 0;
}
```



```
C:\work\2020-06\2020-06-01\講義\079コンピュータグラフィックスII2020前期\004講義資料\05復習とマウスとキーボ...
..xx0..
...o0..
.....0.
.....
.....

black 7, white 2
timestep 0/5, state 1

.....
.....
.....
..ooo..
..xx0..
...o0..
.....0.
.....

black 7, white 2
timestep 1/5, state 1
```

盤面の情報を取得するサンプルソースコード

プロトタイプ・その3

```
#pragma warning(disable:4996)
#include <stdlib.h>
#include <stdio.h>
#include <GL/glut.h>

const int delay = 500;
int boardnow[8][8];
FILE* fp;

void myDisplay()
{
    int i, j;
    double posy, posx;

    glClearColor(0.0, 0.5, 0.0, 1.0);
    glClear(GL_COLOR_BUFFER_BIT);

    for (i = 0; i < 8; i++) {
        for (j = 0; j < 8; j++) {
            if (boardnow[i][j] > 0) glColor3d(0.2, 0.2, 0.2);
            else glColor3d(0.8, 0.8, 0.8);
            if (boardnow[i][j] != 0) {
                posy = (double)i - 3.5;
                posx = (double)j - 3.5;
                glPushMatrix();
                glTranslated(posx * 0.2, posy * 0.2, 0.0);
                glutSolidSphere(0.08, 10, 10);
                glPopMatrix();
            }
        }
    }

    glutSwapBuffers();
}

void myTimer(int value)
{
    int i, j;
    int currentstate;
    if (value == 1) {
        for (i = 0; i < 8; i++) {
            for (j = 0; j < 8; j++) {
                fread(&boardnow[i][j], sizeof(int), 1, fp);
            }
        }
        fread(&currentstate, sizeof(int), 1, fp);
        if (currentstate == -1) {
            fclose(fp);
            exit(0);
        }
        glutTimerFunc(delay, myTimer, 1);
        glutPostRedisplay();
    }
}

void myKeyboard(unsigned char key, int x, int y)
{
    if (key == 0x1B) exit(0);
}

void myInit(char* progname)
{
    glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(0, 0);
    glutCreateWindow(progname);
}
```

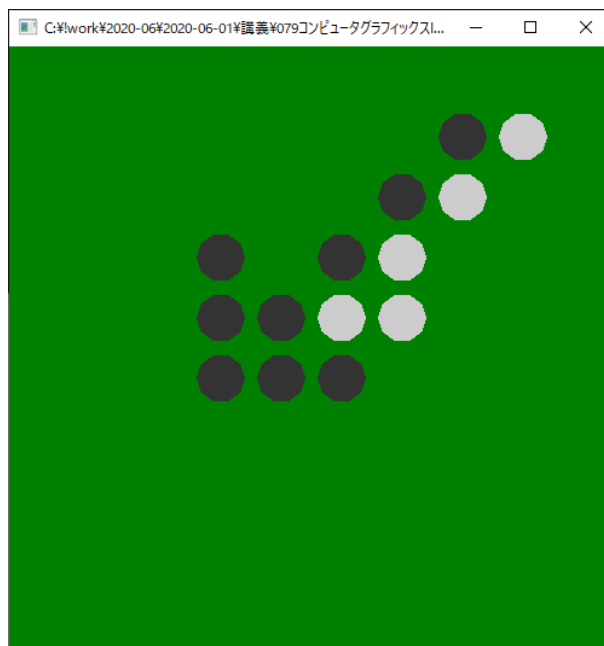
プロトタイプ・その3

```
int main(int argc, char* argv[])
{
    int i, j;
    int dummy1;
    char dummy2[16];

    if (argc <= 1) {
        printf("コマンド引数にファイル名が指定されていません\n");
        return 1;
    }
    fp = fopen(argv[1], "rb");
    if (fp == NULL) {
        printf("ファイル%sを開けません\n", argv[1]);
        return 1;
    }
    fread(&dummy1, sizeof(int), 1, fp);
    fread(dummy2, sizeof(char), 16, fp);
    fread(dummy2, sizeof(char), 16, fp);

    for (i = 0; i < 8; i++) {
        for (j = 0; j < 8; j++) {
            boardnow[i][j] = 0;
        }
    }

    glutInit(&argc, argv);
    myInit(argv[0]);
    glutKeyboardFunc(myKeyboard);
    glutTimerFunc(delay, myTimer, 1);
    glutDisplayFunc(myDisplay);
    glutMainLoop();
    return 0;
}
```



盤面の情報を取得して二次元アニメーションを表示するサンプルソースコード