

```

1  #include "DxLib.h"
2  #include "Block.h"
3  #include "Ball.h"
4  #include "Game.h"
5
6
7  static int mImageBlock[BCOLOR];    //画像ハンドル格納用変数
8
9  // ブロックデータの宣言
10 struct ObjectBlock    iBlock[HEIGHT][WIDTH];
11
12 //ステージ番号
13 static int StageNumber; //ステージ番号;
14 static int StgClear;    //ステージクリア
15
16 int iBlockData[STAGE][HEIGHT][WIDTH] = {    // ブロック配列
17 //デバッグ用{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
18
19 {
20     // 0ステージ
21     { 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3 },
22     { 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3 },
23     { 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3 },
24     { 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 },
25     { 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 },
26     { 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 },
27     { 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 },
28     { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 },
29     { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 },
30     { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 },
31 },
32 {
33     // 1ステージ
34     { 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0 },
35     { 0, 0, 0, 0, 0, 2, 3, 3, 3, 3, 2, 0, 0, 0, 0, 0 },
36     { 0, 0, 0, 2, 2, 3, 4, 4, 4, 4, 3, 2, 2, 0, 0, 0 },
37     { 0, 2, 2, 3, 3, 4, 5, 5, 5, 5, 4, 3, 3, 2, 2, 0 },
38     { 2, 3, 3, 4, 4, 5, 0, 0, 0, 0, 5, 4, 4, 3, 3, 2 },
39     { 2, 3, 3, 4, 4, 5, 0, 0, 0, 0, 5, 4, 4, 3, 3, 2 },
40     { 0, 2, 2, 3, 3, 4, 5, 5, 5, 5, 4, 3, 3, 2, 2, 0 },
41     { 0, 0, 0, 2, 2, 3, 4, 4, 4, 4, 3, 2, 2, 0, 0, 0 },
42     { 0, 0, 0, 0, 0, 2, 3, 3, 3, 3, 2, 0, 0, 0, 0, 0 },
43     { 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0 },
44 },
45 {
46     // 2ステージ
47     { 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 },
48     { 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 },
49     { 0, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 0 },
50     { 0, 0, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 0, 0 },
51     { 0, 0, 0, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 0, 0, 0 },
52     { 0, 0, 0, 0, 5, 5, 5, 5, 5, 5, 5, 5, 0, 0, 0, 0 },
53     { 0, 0, 0, 0, 0, 4, 4, 4, 4, 4, 4, 0, 0, 0, 0, 0 },
54     { 0, 0, 0, 0, 0, 0, 4, 4, 4, 4, 0, 0, 0, 0, 0, 0 },
55     { 0, 0, 0, 0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 0, 0, 0 },
56     { 0, 0, 0, 0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 0, 0, 0 },
57 },
58 {
59     // 3ステージ
60     { 7, 1, 0, 0, 1, 7, 3, 3, 3, 0, 7, 0, 4, 4, 0, 7 },
61     { 7, 1, 0, 1, 1, 7, 3, 0, 0, 3, 7, 4, 4, 4, 4, 7 },
62     { 7, 1, 1, 1, 0, 7, 3, 0, 0, 3, 7, 4, 0, 0, 4, 7 },
63     { 7, 1, 1, 0, 0, 7, 3, 0, 0, 3, 7, 4, 0, 0, 0, 7 },

```

```

61     { 7, 1, 0, 0, 0, 7, 3, 3, 3, 0, 7, 4, 0, 0, 0, 7 },
62     { 7, 1, 1, 0, 0, 7, 3, 0, 0, 3, 7, 4, 0, 0, 0, 7 },
63     { 7, 1, 1, 1, 0, 7, 3, 0, 0, 3, 7, 4, 0, 0, 4, 7 },
64     { 7, 1, 0, 1, 1, 7, 3, 0, 0, 3, 7, 4, 4, 4, 4, 7 },
65     { 7, 1, 0, 0, 1, 7, 3, 3, 3, 0, 7, 0, 4, 4, 0, 7 },
66     { 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 }
67 }
68 };
69
70 //初期化
71 void Block_Initialize() {
72     StageNumber = 0;           //0ステージからスタート
73     StgClear = 0;             //ステージクリアフラグ
74
75     //ブロック画像データの読み込み
76     LoadDivGraph("images/Block.png", 8, 8, 1, 40, BCOLOR, mImageBlock);
77
78     Block_StageSet();
79 }
80
81 // ブロック配列の初期化
82 void Block_StageSet() {
83     for (int i = 0; i < HEIGHT; i++) {
84         for (int j = 0; j < WIDTH; j++) {
85             if (iBlockData[StageNumber][i][j] != 0) {
86                 iBlock[i][j].flg = 1;
87                 iBlock[i][j].x = j * 40;
88                 iBlock[i][j].y = i * 16 + 80;
89                 iBlock[i][j].w = 40;
90                 iBlock[i][j].h = 8;
91                 iBlock[i][j].image = iBlockData[StageNumber][i][j];
92                 iBlock[i][j].score = iBlock[i][j].image * 10;
93             }
94         }
95     }
96 }
97
98 //終了処理
99 void Block_Finalize() {
100     for (int i = 0; i < BCOLOR; i++) {
101         DeleteGraph(mImageBlock[i]); //画像の解放
102     }
103 }
104
105 //更新
106 void Block_Update() {
107     if (StgClear) {
108         StageNumber++;           //次のステージ
109         Block_StageSet();       //ブロック配置
110         Ball_SetBallStatus();   //ボールを初期化
111         StgClear = 0;           //ステージクリアフラグを0へ
112     }
113 }
114
115 //描画
116 void Block_Draw() {
117     int checkflg = 0;           //ステージクリアチェックフラグ
118     for (int i = 0; i < HEIGHT; i++) {
119         for (int j = 0; j < WIDTH; j++) {
120             if (iBlock[i][j].flg == 1) {

```

```
121         DrawGraph(iBlock[i][j].x, iBlock[i][j].y, mImageBlock[iBlock[i]
           [j].image], TRUE);
122         checkflg = 1;
123     }
124 }
125 }
126
127 //ブロックが1つも書かれなかったら、ステージクリア
128 if (!checkflg) {
129     StgClear = 1;
130 }
131 }
132
133 //ボールとブロックの当たり判定
134 void HitBlocks() {
135     // ボールとブロックの当たり判定
136     int x = mBall.fPosX / 40;
137     int y = (mBall.fPosY - 80) / 16;
138
139     if (x >= 0 && x < WIDTH && y >= 0 && y < HEIGHT && iBlock[y][x].flg != 0) {
140
141         iBlock[y][x].flg = 0; //非表示ブロック
142         Game_ScoreAdd(iBlock[y][x].score);
143
144         mBall.fMoveY = -mBall.fMoveY;
145     }
146 }
147
```