

```
1 #include "DxLib.h"
2 #include "SceneMgr.h"
3 #include "Input.h"
4 #include "Enemy.h"
5 #include "Player.h"
6
7 extern int HitBoxPlayer (PLAYER*, ENEMY*); //当たり判定 (四角)
8
9 /*****
10 ** 変数 **
11 *****/
12 // エネミー
13 struct ENEMY mEnemy[ENEMY_MAX];
14 struct ENEMY enemyCar = { true, 0, 0, 0, -50, 63, 120, 0, 1, 0, 0 };
15 struct ENEMY enemyBike = { true, 3, 0, 0, -50, 32, 72, 0, 1, 1, 0 };
16
17 int mImageEnemy[ENEMY_IMAGE_MAX];
18 int EnemyCount[ENEMY_IMAGE_MAX];
19
20 //初期化
21 void Enemy_Initialize() {
22     LoadDivGraph("images/car.png", 3, 3, 1, 63, 120, mImageEnemy); //エネミー画像
23     mImageEnemy[3] = LoadGraph("images/bike.png"); //エネミー(バイク)
24
25     // エネミーの初期設定
26     for (int i = 0; i < ENEMY_MAX; i++) {
27         mEnemy[i].flg = false;
28     }
29 }
30
31 //終了処理
32 void Enemy_Finalize() {
33     DeleteGraph(mImageEnemy[0]); //画像の解放
34     DeleteGraph(mImageEnemy[1]); //画像の解放
35     DeleteGraph(mImageEnemy[2]); //画像の解放
36     DeleteGraph(mImageEnemy[3]); //画像の解放
37 }
38
39 //更新
40 void Enemy_Update() {
41     for (int i = 0; i < ENEMY_MAX; i++) {
42         if (mEnemy[i].flg == true) {
43             // 真っすぐ下に移動
44             mEnemy[i].y += mEnemy[i].speed + mPlayer.speed - PLAYER_SPEED;
45             //オートバイのジグザク走行
46             if (mEnemy[i].type == 3 && ++mEnemy[i].cnt < 30) {
47                 mEnemy[i].x += mEnemy[i].mx;
48             }
49             else {
50                 mEnemy[i].x += mEnemy[i].mx * -1;
51                 if (mEnemy[i].cnt > 60) mEnemy[i].cnt = 0;
52             }
53             // 画面をはみ出したら消去
54             if (mEnemy[i].y > SCREEN_HEIGHT + mEnemy[i].h) mEnemy[i].flg = FALSE;
55
56             // 敵機を追い越したらカウントする
```

```
61     if (mEnemy[i].y > mPlayer.y && mEnemy[i].point == 1) {
62         mEnemy[i].point = 0;
63         EnemyCount[mEnemy[i].type]++;
64     }
65
66     //敵車とのヒットチェック
67     if (HitBoxPlayer(&mPlayer, &mEnemy[i]) == TRUE) {
68         mPlayer.flg = false;
69         mPlayer.speed = PLAYER_SPEED;
70         mPlayer.count = 0;
71         mPlayer.hp -= 20;
72         mEnemy[i].flg = false;
73     }
74 }
75
76
77 // エネミーの生成
78 if (mPlayer.mileage % ENEMY_CREATE_TIME == 0) {
79     CreateEnemy();
80 }
81
82 }
83
84 //描画
85 void Enemy_Draw() {
86     for (int i = 0; i < ENEMY_MAX; i++) {
87         if (mEnemy[i].flg == true) {
88
89             DrawRotaGraph(mEnemy[i].x, mEnemy[i].y, 1.0f, 0, mEnemy[i].img, TRUE, FALSE);
90
91         }
92     }
93 }
94
95 /*****
96  * エネミーの生成
97  * 引 数：なし
98  * 戻り値：TRUE:成功 FALSE:失敗 <---- 戻り値はあえて使わない。
99  *****/
100 int CreateEnemy()
101 {
102     for (int i = 0; i < ENEMY_MAX; i++) {
103         if (mEnemy[i].flg == FALSE) {
104             mEnemy[i] = enemyCar;
105             mEnemy[i].type = GetRand(3);
106             if (mEnemy[i].type == 3) mEnemy[i] = enemyBike;
107             mEnemy[i].img = mImageEnemy[mEnemy[i].type];
108             mEnemy[i].x = GetRand(4) * 105 + 30 + GetRand(20);
109             mEnemy[i].speed = 3 + mEnemy[i].type * 1;
110             // 成功
111             return TRUE;
112         }
113     }
114
115     // 失敗
116     return FALSE;
117 }
118
119 /*****
120  * 当たり判定 (四角)
```

```
121 * 引 数 : PLAYERポインタ, ENEMYポインタ
122 * 戻り値 : TRUE:当たり, FALSE:なし
123 *****/
124 int HitBoxPlayer (PLAYER* p, ENEMY* e)
125 {
126     // x, y は中心座標とする
127     int sx1 = p->x - (int) (p->w * 0.45f);
128     int sy1 = p->y - (int) (p->h * 0.45f);
129     int sx2 = p->x + (int) (p->w * 0.45f);
130     int sy2 = p->y + (int) (p->h * 0.45f);
131
132     int dx1 = e->x - (int) (e->w * 0.4f);
133     int dy1 = e->y - (int) (e->h * 0.4f);
134     int dx2 = e->x + (int) (e->w * 0.4f);
135     int dy2 = e->y + (int) (e->h * 0.4f);
136
137     // 矩形が重なっていれば当たり
138     if (sx1 < dx2 && dx1 < sx2 && sy1 < dy2 && dy1 < sy2) {
139         return TRUE;
140     }
141
142     return FALSE;
143 }
```