

```
1  #include <iostream>
2  #include <fstream>
3  #include <windows.h>
4  #include <dos.h>
5  #include <conio.h>
6  #define x_max 1000
7  #define y_max 1000
8  using namespace std;
9
10 typedef struct{
11     int type;
12 }note;
13
14 typedef struct{
15     int x, y;
16 }point;
17
18 point nextpoint(point now, note a[][y_max]);
19 void clearline(int n);
20 void pause(int n);
21 void gotoxy(int x, int y);
22 void ConsoleFullScreen();
23 void setColor(int color);
24 void char_to_notearray(note a[][y_max], char in[], int n, int *x, int *y);
25 void printSwitch(int a);
26 void print_map(note a[][y_max], int x, int y);
27 void foodpoint(point p, note a[][y_max]);
28 void mouse_stack(note a[][y_max], int x, int y, int gox, int goy);
29 void print_mouse_point(int x);
30 void runtime();
31 bool CheakAll(point p, note a);
32 bool notlor3(point p, note a[][y_max]);
33 int InDate(char in[]);
34
35 //全域變數
36 int mouse_point = 500;
37 int op = 1;
38 int op4or8 = 1;
39 int time = 10;
40 int timeop = 10;
41
42 int main ( void )
43 {
44     int t;
45     int x = 0, y = 0, n = 0;
46     int gox, goy;
47     char in[x_max * y_max];
48     note a[x_max][y_max];
49
50     ConsoleFullScreen();
51
52     //讀入迷宮地圖
53     n = InDate(in);
```

```
54 char_to_notearray(a, in, n, &x, &y);
55 //列印地圖
56 print_map(a, x, y);
57
58 //輸入老鼠投放位置
59 bool run = true;
60 cout << "\n請輸入老鼠要投置位置(x, y):";
61 do
62 {
63     cin >> gox >> goy;
64     if(a[gox][goy].type == '1') clearline(x + 3), gotoxy(x + 3, 0), cout << "牆壁⤴
        無法投放，再輸入一次。";
65     else if(gox >= x || goy >= y || (gox >= x && goy >= y)) clearline(x + 3), ⤴
        gotoxy(x + 3, 0), cout << "超出範圍，再輸入一次。";
66     else run = false;
67 }
68 while(run);
69
70 cout << "請輸入老鼠要行走的速度(數字越小越快):";
71 do
72 {
73     cin >> time;
74     if(time <= 0) clearline(x + 4), gotoxy(x + 4, 0), cout << "時間輸入錯誤";
75 }
76 while(time <= 0);
77 timeop = time;
78
79 cout << "請輸入老鼠可行走的方位(4 or 8):";
80 do
81 {
82     cin >> t;
83     if (t == 4 || t == 8) op4or8 = t;
84     else clearline(x + 5), gotoxy(x + 5, 0), cout << "方位輸入錯誤";
85 }
86 while(t != 8 && t != 4);
87
88 //老鼠投放至起始位置
89 gotoxy(2 + gox, goy * 2);
90 printSwitch('@');
91 print_mouse_point(x);
92 pause(x + 16);
93
94 //老鼠開跑囉!!!!
95 mouse_stack(a, x, y, gox, goy);
96
97 //結束!!
98 pause(x + 16);
99 }
100
101 void clearline(int n)
102 {
103     gotoxy(n, 0);
104     cout << "
```

```
        ";
105 }
106
107 void ConsoleFullScreen()
108 {
109     keybd_event(VK_MENU,0x38,0,0);
110     keybd_event(VK_RETURN,0x1c,0,0);
111     keybd_event(VK_MENU,0xb8,KEYEVENTF_KEYUP,0);
112     keybd_event(VK_RETURN,0x9c,KEYEVENTF_KEYUP,0);
113 }
114
115 void pause(int n)
116 {
117     gotoxy(n, 0);
118     system("pause");
119     gotoxy(n, 0);
120     cout << "        ";
121 }
122
123 void gotoxy(int x, int y)
124 {
125     static HANDLE o = GetStdHandle (STD_OUTPUT_HANDLE);
126     COORD c = {y, x};
127     SetConsoleCursorPosition (o, c);
128 }
129
130 void setColor(int color)
131 {
132     HANDLE hConsole;
133     hConsole = GetStdHandle (STD_OUTPUT_HANDLE);
134     SetConsoleTextAttribute(hConsole, color);
135 }
136
137 int InDate(char in[])
138 {
139     fstream InF;
140     int n = 0;
141
142     char FName[20], ch;
143     cout << "輸入方程式檔名:";
144     cin >> FName;
145     InF.open(FName, ios::in);
146     if(!InF)
147     {
148         cout << "檔案無法開啟\n";
149         exit(1);
150     }
151     else
152     {
153         while(InF.get(ch))
154         {
155             in[n] = ch;
156             n++;
157         }
158     }
159 }
```

```
156     }
157     InF.close();
158 }
159 return n;
160 }
161
162 void char_to_notearray(note a[][y_max], char in[], int n, int *x, int *y)
163 {
164     int xx = 0, yy = 0, maxY = 0;
165     for (int i = 0; i < n; i++)
166     {
167         if (in[i] == '\n')
168         {
169             xx += 1;
170             yy = 0;
171         }
172         else
173         {
174             a[xx][yy++].type = in[i];
175             if (yy > maxY) maxY = yy;
176         }
177     }
178     if (in[n - 1] != '\n') xx++;
179     *x = xx;
180     *y = maxY;
181 }
182
183 void printSwitch(int a)
184 {
185     /*
186     ## 0 = 未走過的路          -- 1 = 牆壁
187     ## 2 = 走過正確的路        -- 3 = 走過錯誤的路
188     ## + = 體力+50的食物       -- * = 體力+100的食物
189     ## $ = 體力+200的食物      -- # = 出口
190     ## + == 43, * == 42, $ == 36
191     */
192     switch(a)
193     {
194         case '0': //未走過的路
195             case 8: //沒能量
196                 setColor(15);
197                 cout << " ";
198                 break;
199             case '1': //牆壁
200                 setColor(155);
201                 cout << "■";
202                 setColor(15);
203                 break;
204             case 9: //有能量
205                 setColor(200);
206                 cout << " ";
207                 setColor(15);
208                 break;
```

```
209     case '2': //走過錯誤的路
210         setColor(127);
211         cout << ".";
212         setColor(15);
213         break;
214     case '3': //走過正確的路
215         setColor(14);
216         cout << ".";
217         setColor(15);
218         break;
219     case '+': //體力+50的食物
220         setColor(78);
221         cout << "+";
222         setColor(15);
223         break;
224     case '*': //體力+80的食物
225         setColor(78);
226         cout << "*";
227         setColor(15);
228         break;
229     case '$': //體力+100的食物
230         setColor(78);
231         cout << "$";
232         setColor(15);
233         break;
234     case '#': //出口
235         setColor(117);
236         cout << "#";
237         setColor(15);
238         break;
239     case '\n': //換行切換
240         setColor(15);
241         cout << " ";
242         cout << "\n";
243         break;
244     case '@': //老鼠
245         setColor(160);
246         cout << "@";
247         setColor(15);
248         break;
249     }
250 }
251
252 void print_map(note a[][y_max], int x, int y)
253 {
254     cout << "迷宮大小 x = " << x << ", y = " << y << endl;
255     for (int i = 0; i < x; i++)
256     {
257         for (int l = 0; l < y; l++)
258         {
259             printSwitch(a[i][l].type);
260         }
261         setColor(15);
262     }
```

```
262     printSwitch('\n');
263 }
264 }
265
266 void print_mouse_point(int x)
267 {
268     gotoxy(7 + x, 0);
269     if (mouse_point > 0)
270     {
271         printf("老鼠目前能量 = %4d ", mouse_point);
272         int hundreds, tens, n_total;
273         hundreds = mouse_point / 100;
274         tens = mouse_point % 100;
275         if (tens > 50) n_total = hundreds * 2 + 1;
276         else n_total = hundreds * 2;
277         for (int i = 0; i < n_total; i++) printSwitch(9);
278         for (int i = 0; i < 11 - n_total; i++) printSwitch(8);
279     }
280     else
281     {
282         cout << " ";
283         cout << "老鼠能量用盡死亡!!!";
284     }
285 }
286
287 bool notlor3(point p, note a[][y_max])
288 {
289     int k = a[p.x][p.y].type;
290     if(k == '0' || k == '+' || k == '*' || k == '$' || k == '#') return true;
291     else return false;
292 }
293
294 void foodpoint(point p, note a[][y_max])
295 {
296     switch(a[p.x][p.y].type)
297     {
298         case '+':
299             mouse_point += 50;
300             a[p.x][p.y].type = '3';
301             break;
302         case '*':
303             mouse_point += 80;
304             a[p.x][p.y].type = '3';
305             break;
306         case '$':
307             mouse_point += 100;
308             a[p.x][p.y].type = '3';
309             break;
310         case '#':
311             op = 0;
312             break;
313         default:
314             a[p.x][p.y].type = '3';
```

```
315     }
316 }
317
318 point nextpoint(point now, note a[][y_max])
319 {
320     /*
321     // 老鼠收尋順序 = 東 > 南 > 西 > 北
322     // 老鼠先判斷是否有下一步路 bool can_go_road
323     // 如果下一步是食物導入 int food_road return food_point
324     // 如果往下個方向不能前進，給定 out(-1, -1)
325     */
326     point out, p;
327
328     if(notlor3(p = {now.x, now.y + 1}, a) == true)
329     {
330         out.x = now.x;
331         out.y = now.y + 1;
332         foodpoint(out, a);
333     }
334     else if (notlor3(p = {now.x + 1, now.y + 1}, a) == true && op4or8 == 8)
335     {
336         out.x = now.x + 1;
337         out.y = now.y + 1;
338         foodpoint(out, a);
339     }
340     else if (notlor3(p = {now.x + 1, now.y}, a) == true)
341     {
342         out.x = now.x + 1;
343         out.y = now.y;
344         foodpoint(out, a);
345     }
346     else if (notlor3(p = {now.x + 1, now.y - 1}, a) == true && op4or8 == 8)
347     {
348         out.x = now.x + 1;
349         out.y = now.y - 1;
350         foodpoint(out, a);
351     }
352     else if (notlor3(p = {now.x, now.y - 1}, a) == true)
353     {
354         out.x = now.x;
355         out.y = now.y - 1;
356         foodpoint(out, a);
357     }
358     else if (notlor3(p = {now.x - 1, now.y - 1}, a) == true && op4or8 == 8)
359     {
360         out.x = now.x - 1;
361         out.y = now.y - 1;
362         foodpoint(out, a);
363     }
364     else if (notlor3(p = {now.x - 1, now.y}, a) == true)
365     {
366         out.x = now.x - 1;
367         out.y = now.y;
```

```
368     foodpoint(out, a);
369 }
370 else if (notlor3(p = {now.x - 1, now.y + 1}, a) == true && op4or8 == 8)
371 {
372     out.x = now.x - 1;
373     out.y = now.y + 1;
374     foodpoint(out, a);
375 }
376 else
377 {
378     out.x = -1;
379     out.y = -1;
380 }
381 return out;
382 }
383
384 void runtime()
385 {
386     switch(mouse_point)
387     {
388         case 1 ... 200:
389             time = timeop * 0.5;
390             break;
391         case 201 ... 300:
392             time = timeop * 0.75;
393             break;
394         case 301 ... 400:
395             time = timeop * 0.8;
396             break;
397         case 401 ... 500:
398             time = timeop;
399             break;
400         case 501 ... 1000:
401             time = timeop * 1.5;
402             break;
403         default:
404             time = timeop * 0.3;
405     }
406 }
407
408 bool CheakAll(point p, note a[][y_max])
409 {
410     int t;
411     t = a[p.x][p.y].type;
412     if (t == '0' || t == '+' || t == '*' || t == '$') return true;
413     else return false;
414 }
415
416 void mouse_stack(note a[][y_max], int x, int y, int gox, int goy)
417 {
418     point next, stack_p[1000];
419     int top = 0, Exit = 0;
420     stack_p[top] = {gox, goy};
```



```
421     a[stack_p[top].x][stack_p[top].y].type = '3';
422     while (Exit == 0 && mouse_point > 0)
423     {
424         do
425         {
426             print_mouse_point(x);
427             next = nextpoint(stack_p[top], a);
428             //先讓上一步路重新顯示
429             gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
430             printSwitch(a[stack_p[top].x][stack_p[top].y].type);
431
432             gotoxy(9 + x, 0);
433             printf("老鼠目前位置 x, y = (%2d, %2d)", stack_p[top].x, stack_p[top].y);
434
435             if (next.x != -1 && next.y != -1)
436             {
437                 //老鼠成功找到下一步，再前往下一步。
438
439                 if (a[stack_p[top].x][stack_p[top].y].type == '2' && op4or8 == 4)
440                 {
441                     a[stack_p[top].x][stack_p[top].y].type = '3';
442                     gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
443                     printSwitch(a[stack_p[top].x][stack_p[top].y].type);
444                 }
445
446                 if (op4or8 == 8)
447                 {
448                     a[stack_p[top].x][stack_p[top].y].type = '3';
449                     gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
450                     printSwitch(a[stack_p[top].x][stack_p[top].y].type);
451                 }
452
453                 top++;
454                 stack_p[top] = next;
455
456                 gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
457                 printSwitch('@');
458
459                 if (op == 0)
460                 {
461                     gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
462                     printSwitch('@');
463                     Exit = 1;
464                     break;
465                 }
466             }
467             else if (next.x == -1 && next.y == -1 && top >= 0)
468             {
469                 //老鼠找不到下一步路，退到上一步路，並讓目前這部重新顯示錯誤的路。
470                 a[stack_p[top].x][stack_p[top].y].type = '2';
471                 gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
472                 printSwitch(a[stack_p[top].x][stack_p[top].y].type);
473             }
```

```
474         top--;
475
476         a[stack_p[top].x][stack_p[top].y].type = '2';
477         gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
478         printSwitch('@');
479     }
480     else
481     {
482         Exit = 1;
483         break;
484     }
485     runtime();
486     gotoxy(8 + x, 0);
487     printf("老鼠目前速度 %4d(ms)", time);
488     _sleep(time);
489     mouse_point--;
490 } while(mouse_point > 0 && top > 0); //判斷老鼠能量
491
492 point q;
493 int k = 0;
494 if (top == 0)
495 {
496     if (CheakAll(q = {stack_p[top].x, stack_p[top].y + 1}, a)) k = 1;
497     if (CheakAll(q = {stack_p[top].x + 1, stack_p[top].y}, a)) k = 1;
498     if (CheakAll(q = {stack_p[top].x, stack_p[top].y - 1}, a)) k = 1;
499     if (CheakAll(q = {stack_p[top].x - 1, stack_p[top].y}, a)) k = 1;
500 }
501 if (top == 0 && op == 8)
502 {
503     if (CheakAll(q = {stack_p[top].x + 1, stack_p[top].y + 1}, a)) k = 1;
504     if (CheakAll(q = {stack_p[top].x + 1, stack_p[top].y - 1}, a)) k = 1;
505     if (CheakAll(q = {stack_p[top].x - 1, stack_p[top].y - 1}, a)) k = 1;
506     if (CheakAll(q = {stack_p[top].x - 1, stack_p[top].y + 1}, a)) k = 1;
507 }
508 if (top == 0 && k == 0) Exit = 1;
509 }
510
511 //最後結算顯示
512 if (op == 0 && mouse_point != 0)
513 {
514     gotoxy(2 + stack_p[top-1].x, stack_p[top-1].y * 2);
515     printSwitch(a[stack_p[top-1].x][stack_p[top-1].y].type);
516     gotoxy(8 + x, 0);
517     printf("老鼠目前位置 x, y = (%2d, %2d)\n", stack_p[top].x, stack_p[top].y);
518     gotoxy(10 + x, 0);
519     cout << "The mouse successfully escaped the maze!!!";
520 }
521 if (mouse_point == 0)
522 {
523     print_mouse_point(x);
524 }
525 if (top <= 0)
526 {
```

```
527         gotoxy(10 + x, 0);
528         cout << "No exit from the maze.";
529     }
530 }
531
```