

پروژه مبانی

بازی Maze

1403/10/30

محمد یاسین خانی

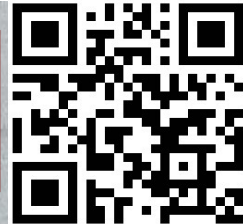
مرتضی حیدری

کلاس مبانی برنامه نویسی

فهرست مطالب

مقدمه	الف
زبان برنامه نویسی C++	ب
کتابخانه گرافیکی	ب
توضیح مختصر پروژه	ب
توضیحات پروژه	۱
طراحی برد بازی	۲
مختصات موانع برد	3
مختصات فضای خالی برد	4
طراحی پنجره	5
تابع کشیدن شکل	6
Label ساختن	6
تنظیم سایز پنجره	6
تنظیم کشیدن خطوط درون پنجره	7
تنظیم دکمه های ورودی و عملکرد آنها	8
تنظیم رنگ پس زمینه پنجره	۱۳
تنظیم تابع WINAPI	۱۳
منطق پروژه	۱۴
منطق طراحی برد پروژه	۱۵
منطق تعیین مختصات موانع و فضای خالی برد	۱۵
منطق تعیین عملکرد دکمه ها	۱۶
منطق اعلام برنده و بازنده	۱۶
نتیجه گیری	۱۶

مقدمه



```
wc.lpfnWndProc = Window_and_Print_Board;
```

```
wc.hInstance = hInstance;
```

```
wc.lpszClassName = CLASS_NAME;
```

```
RegisterClass(&wc);
```

```
HWND hwnd = CreateWindowEx(
```

```
0,
```

```
CLASS_NAME,
```

```
"Maze Project",
```

```
WS_OVERLAPPEDWINDOW & ~WS_MAXIMIZEBOX,
```

```
CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
```

```
NULL,
```

```
NULL,
```

```
hInstance,
```

```
NULL
```

```
);
```

```
ShowWindow(hwnd, nShowCmd);
```

تنها کسی از دیگران پیشی خواهد گرفت. که از شروع کردن نهراسد.

استراس تروپ، ریاضیدان و دانشمند دانمارکی در آزمایشگاه بل به منظور توسعه زبان C، زبان C++ را ساخت؛ که در ابتدا این زبان را C با کلاس (C With Classes) نامید، نقطه عطف زبان C++ نسبت به زبان C مواردی مانند کلاسها، توابع مجازی، سربارگزاری عملگرها، وراثت چندگانه، قالب توابع و پردازش استثناء بوده، که باعث شد C++ نسب به C بهینه تر باشد؛ همچنین به این موضوع هم باید توجه داشته باشیم که، زبان C++ یک زبان سطح میانی هست، که ویژگی های یک زبان سطح بالا را درون خود دارد و همچنین مانند یک زبان سطح پایین به راحتی کامپایل می شود؛ همچنین ویژگی های دیگری که آن را همانند یک زبان سطح پایین هم می کند.

همچنین خوب است بدانید که در سال ۱۹۸۳ نا این زبان به C++ تغییر کرد؛ و این زبان دارای قابلیت های انواع داده ایستا، نوشتار آزاد و چندمدلی می باشد و معمولاً زبان ترجمه شده با پشتیبانی از برنامه نویسی ساخت یافته، برنامه نویسی شیء گرا و برنامه نویسی جنریک است. از آنجا که در C++ اشیاء را می توان ابتدا به ساکن از کلاس هایی ایجاد کرد که به هیچگونه سلسله مراتب رده ها و وراثت مقید نیستند، لذا C++ از برنامه سازی شیء بنیاد نیز پشتیبانی می کند. این زبان برنامه نویسی در سال ۱۹۹۸ تحت نام ISO/IEC ۱۴۸۸۲:۱۹۹۸ استاندارد شد. نسخه فعلی استاندارد این زبان ISO/IEC ۱۴۸۸۲:۲۰۱۴ است.

کتابخانه گرافیکی

این کتابخانه گرافیکی که به برنامه نویسان اجازه می دهد تا از توابع و امکانات مختلف سیستم عامل ویندوز استفاده کنند. این کتابخانه به طور خاص برای توسعه برنامه های کاربردی ویندوز طراحی شده است و شامل توابعی برای مدیریت پنجره ها، پردازش پیام ها، مدیریت فایل ها، کار با رشته ها و بسیاری از ویژگی های دیگر است. مایکروسافت، شرکتی که سیستم عامل ویندوز را ایجاد کرده است، مسئول توسعه windows.h و API مربوط به آن است. این کتابخانه به طور مداوم به روزرسانی شده تا با نسخه های جدید ویندوز و نیازهای برنامه نویسان همگام باشد.

این زبان به طور گسترده ای در توسعه برنامه های گرافیکی و بازی ها استفاده می شود. برای مثال، در کتابخانه هایی مانند DirectX، توابع این کتابخانه برای مدیریت پنجره ها و پردازش ورودی کاربر ضروری هستند، این کتابخانه امکاناتی برای مدیریت فایل ها، حافظه، و سایر منابع سیستم فراهم می کند که برای هر نوع نرم افزاری که بر روی ویندوز اجرا می شود، ضروری است.

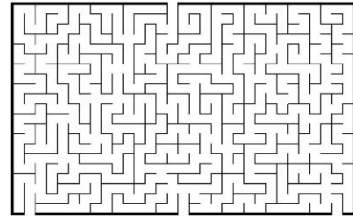
کتابخانه windows.h یکی از اجزای کلیدی برای برنامه نویسی در محیط ویندوز است که توسط مایکروسافت توسعه یافته و به برنامه نویسان این امکان را می دهد تا از قابلیت های گسترده سیستم عامل بهره برداری کنند. این کتابخانه با توجه به نیازهای روزافزون توسعه دهندگان، به طور مداوم در حال بهبود و گسترش است.

توضیح مختصر پروژه

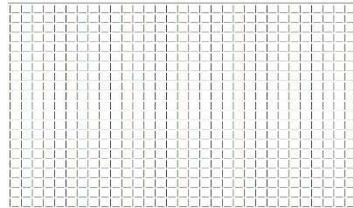
در این پروژه از کتابخانه گرافیکی windows.h استفاده شده، با استفاده از این کتابخانه گرافیکی، توابعی تعریف شده و همچنین دستوراتی نوشته شده؛ در جهت باز شدن یک پنجره و نمایش برد بازی و در آخر منطق بازی بر روی پروژه پیاده شده، به این نکته توجه داشته باشید که به منظور تقلب نکردن کاربر، برنامه برخی از حرکات که کاربر انجام می دهد را به صورت مستقیم به منظور تقلب کاربر گرفته و کاربر را بازنده اعلام می کند، اما اگر کاربر به صورت نرمال بازی را انجام دهد حتی اگر در مانعی گیر بکند، فقط به تعداد دفعات یکبار می تواند از مانع بپرد، اما اگر بار بعد در مانع گیر کرد و راه گذرنده نداشته باشد؛ کاربر بازنده اعلام می شود.



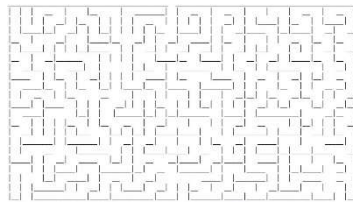
طراحی برد بازی



شکل 1-1 برد بازی MAZE از روی این شکل الهام گرفته شده است.



شکل 2-2 این برد خام بازی MAZE است.



شکل 3-3 این برد اصلی بازی است ، که با استفاده از برد خام که طراحی شده بود ، آن را طراحی کردم ، فقط این برد در فایل با فرمت [txt] ذخیره شده که در ادامه توضیحات متوجه خواهید شد که وجود این برد برای کشیدن برد به صورت گرافیکی چقدر حیاتی است.

بازی Maze می تواند برد های مختلفی داشته باشد ، و زمانیکه ما بازی Maze را طراحی می کنیم باید ابتدا به

دنبال یک برد بازی خوب برای آن باشیم ، اما برد پیشنهادی این پروژه در شکل 1-1 قابل مشاهده خواهد بود ، این برد پیشنهادی شامل موانع افقی و عمودی بسیار زیادی است ؛ و مسئله ای که طراح در وهله اول با آن مواجه می شود آن است که چگونه می توان برد بازی را طراحی کرد ، چرا که برد بازی Maze دارای تعداد بسیار زیادی خطوط افقی و عمودی است و به همین علت نمی توان آن را به راحتی و به صورت دستی طراحی کرد به همین علت می توان از روش های دیگری بهره برد.

در ادامه قصد بر این است که منطق پیاده سازی برد بازی را به صورت کامل توضیح دهیم، پس در ادامه به کد زیر توجه کنید.

در وهله اول کتابخانه های مورد نیاز را فراخوانی می کنیم :

(کتابخانه fstream جهت اجرای دستورات مرتبط با فایل هاست.)

```
#include <iostream>
```

```
#include <fstream>
```

سپس دستور namespace را پیاده سازی می کنیم تا در کد هر کجا که نیاز به std بود برنامه به صورت پیش فرض آن را صدا بزند:

```
using namespace std
```

سپس تابع main را تعریف می کنیم تا دستورات مورد نیاز برنامه را در آن بنویسیم.

```
int main(){
```

سپس دستور ofstream را می نویسیم و بعد از آن یک اسم به انتخاب خودمان می نویسیم و سپس اسم فایل مورد نظر ما که می خواهیم اطلاعات در آن ذخیره شود را می نویسیم.

```
ofstream file("Board_0.txt");
```

حلقه for بعدی به این دلیل است که ما به برنامه یک بازه می دهیم تا برنامه در آن بازه برای ما مقادیر را خروجی دهد و در آخر کار اطلاعات را در فایل ذخیره کنید ؛ سپس فایل را ببندد. (خروجی برنامه را در شکل 1-2 می بینید.)

```
for(int i = 0; i < 22 ; i++){
    for(int j = 0; j < 80 ; j++){
        if (i == 0){
            if (j < 79) file << "_";
            else file << endl;
        }
    }
    for(int j = 0; j <= 63 ; j++){
        if ((i > 0) and (i < 22)){
            if ((j % 2) == 0) file << "|";
            else if (j == 63) file << endl;
            else file << "_";
        }
    }
}
file.close();
}
```

```
1 #include <iostream>
2 #include <fstream>
3
4 using namespace std;
5
6 int main(){
7     ofstream file("Board_0.txt");
8     for(int i = 0; i < 22 ; i++){
9         for(int j = 0; j < 80 ; j++){
10             if (i == 0){
11                 if (j < 79) file << "_";
12                 else file << endl;
13             }
14         }
15         for(int j = 0; j <= 63 ; j++){
16             if ((i > 0) and (i < 22)){
17                 if ((j % 2) == 0) file << "|";
18                 else if (j == 63) file << endl;
19                 else file << "_";
20             }
21         }
22     }
23     file.close();
24 }
```

شکل 1-4 این شکل یک ، نمای کلی از کد ایجاد برد خام بازی است.

مختصات موانع برد

آیا می دانید که

مصریان از نخستین اقوامی بودند که مازها را ساخته‌اند البته که برای سرگرمی و بازی آن را ساخته‌اند؛ بلکه آنها با ساخت مسیرهای تو در تو و پرتاب و خم، ثروت و گنجینه‌هایشان را پنهان می‌کردند.

برای بدست آوردن مختصات موانع، به علت آنکه تعداد موانع زیاد است، نمی‌توان مختصات موانع را به صورت دستی تنظیم کرد پس باید به دنبال یک روش مفید و بهینه برای یافتن این مختصات باشیم، اما روش بهینه چیست؟ چطور می‌توان المان‌های زیادی را دریافت کرد و سپس مختصات آنها را خروجی داد؟

ابتدا فایل مورد نظر که قرار است مختصات موانع در آن نگهداری کنیم را میسازیم:

```
ofstream file("board_pos_1.text");
```

با استفاده از حلقه زیر مختصات موانع را مشخص می‌کنیم، به هر مانع 4 مقدار تعلق می‌گیرد که برای کشیدن برد کافی است.

```
for (int i = 0; i < L.size(); i++){
```

```
if (L[i] == ','){
```

```
    //cout << endl;
```

```
    x = 66, y += 25;
```

```
}
```

```
else {
```

```
    //cout << L[i];
```

```
if (L[i] == '|'){
```

```
    x += 10;
```

```
    XY[n][0] = x, XY[n][1] = y;
```

```
    XY[n][2] = x, XY[n][3] = y - 25;
```

```
    n++;
```

```
}
```

```
else if (L[i] == '_'){
```

```
    XY[n][0] = x, XY[n][1] = y;
```

```
    x += 10;
```

```
    XY[n][2] = x, XY[n][3] = y;
```

```
    n++;
```

```
}
```

```
else if (L[i] == ' ') x += 10;
```

```
}
```

```
1  #include <iostream>
2  #include <string>
3  #include <fstream>
4
5  using namespace std;
6
7  int main(){
8      ofstream file("board_pos_1.text");
9      // (75 - 625) , (66 - 1006)
10     long XY[2068][4];
11     long x = 66, y = 75, n = 0;
12     string L = "
13
14     for (int i = 0 ; i < L.size() ; i++){
15         if (L[i] == ','){
16             //cout << endl;
17             x = 66 , y += 25;
18         }
19         else {
20             //cout << L[i];
21             if (L[i] == '|'){
22                 x += 10;
23                 XY[n][0] = x, XY[n][1] = y;
24                 XY[n][2] = x, XY[n][3] = y - 25;
25                 n++;
26             }
27             else if (L[i] == '_'){
28                 XY[n][0] = x, XY[n][1] = y;
29                 x += 10;
30                 XY[n][2] = x, XY[n][3] = y;
31                 n++;
32             }
33             else if (L[i] == ' ') x += 10;
34         }
35     }
36     file << "{";
37     for (int i = 0 ; i < n ; i++){
38         file << "{" << " " << XY[i][0] << " , " << XY[i][1] << " , " << XY[i][2] << " , " << XY[i][3] << " }" << " , ";
39     }
40     file << "}";
41     file.close();
42 }
```

شکل ۱-۵ این شکل یک، نمای کلی از کد مختصات موانع برد است.

برای آنکه از آن مختصات استفاده کنیم باید آن را به صورت یک آرایه دوبعدی ذخیره کنیم:

```
file << "{";
```

```
for (int i = 0; i < n; i++){
```

```
file << "{" << " " << XY[i][0] << " , " << XY[i][1] << " , " << XY[i][2] << " , " << XY[i][3] << " }" << " , ";
```

```
}
```

```
file << "}";
```

آیا می دانید که

برخی در پارک ها یا مزرعه های
خویش مازهایی ساخته اند و مردم
برای رفتن و گشتن در آن مازها هم
پول می دهند. البته مازهای هستند
که واقعا سخت هستند آن قدر سخت
که برخی از درماندگان در آن مازها
دست به دامان پلیس و آتش نشانی
شده اند؛ تا آنها را از ماز نجات دهند!

مختصات فضای خالی برد

برای بدست آوردن مختصات فضای خالی ، به علت آنکه تعداد فضای خالی هم زیاد است ، نمی توان مختصات آن را
به صورت دستی تنظیم کرد پس باید به دنبال یک روش مفید و بهینه برای یافتن این مختصات باشیم ، اما روش بهینه چیست؟
چطور می توان المان های زیادی را دریافت کرد و سپس مختصات آنها را خروجی داد؟
ابتدا فایل مورد نظر که قرار است مختصات را در آن نگهداری کنیم را میسازیم :

```
ofstream file("S_pos_.text");
```

با استفاده از حلقه زیر مختصات فضای خالی را مشخص می کنیم ، به هر مانع ۳ مقدار تعلق می گیرد که برای کشیدن برد کافی
است.

```
for (int i = 0; i < L.size(); i++){  
  
    if (L[i] == ','){  
  
        //cout << endl;  
  
        x = 66 , y += 25;  
    }  
  
    else {  
  
        //cout << L[i];  
  
        if (L[i] == '|') x += 10;  
  
        else if (L[i] == '_'){  
  
            XY[n][0] = x, XY[n][2] = y;  
  
            x += 10;  
  
            XY[n][1] = x;  
  
            n++;  
        }  
  
        else if (L[i] == ' '){  
  
            XY[n][0] = x, XY[n][2] = y;  
  
            x += 10;  
  
            XY[n][1] = x;  
  
            n++;  
        }  
  
    }  
  
}
```

```
1  #include <iostream>  
2  #include <string>  
3  #include <fstream>  
4  
5  using namespace std;  
6  
7  int main(){  
8      ofstream file("S_pos_.text");  
9      // (75 - 625) , (66 - 1006)  
10     long XY[2068][3];  
11     long x = 66, y = 75, n = 0;  
12     string L = "  
13  
14     for (int i = 0 ; i < L.size() ; i++){  
15         if (L[i] == ','){  
16             //cout << endl;  
17             x = 66 , y += 25;  
18         }  
19         else {  
20             //cout << L[i];  
21             if (L[i] == '|') x += 10;  
22             else if (L[i] == '_'){  
23                 XY[n][0] = x, XY[n][2] = y;  
24                 x += 10;  
25                 XY[n][1] = x;  
26                 n++;  
27             }  
28             else if (L[i] == ' '){  
29                 XY[n][0] = x, XY[n][2] = y;  
30                 x += 10;  
31                 XY[n][1] = x;  
32                 n++;  
33             }  
34         }  
35     }  
36     file << "{";  
37     for (int i = 0 ; i < n ; i++){  
38         file << "{" << " " << XY[i][0] << " , " << XY[i][1] << " , " << XY[i][2] << " }" << " , ";  
39     }  
40     file << "}";  
41     file.close();  
42 }
```

شکل ۱-۶ این شکل یک ، نمای کلی از کد مختصات فضای خالی برد است.

برای آنکه از آن مختصات استفاده کنیم باید آن را به صورت یک آرایه دوبعدی ذخیر کنیم :

```
file << "{";  
  
for (int i = 0; i < n; i++){  
  
file << "{" << " " << XY[i][0] << " , " << XY[i][1] << " , " << XY[i][2] << " }" << " , ";  
  
file << "}";
```


طراحی پنجره

برخی از فواید ماز :

۱. تقویت و پرورش مهارت

۲. تقویت دقت و توجه

۳. افزایش تمرکز کودکان

۴. تقویت مهارتهای حرکتی

۵. یک سرگرمی خوب

۶. تقویت مهارت مسیر یابی

۷. تقویت جهت یابی

۸. ایجاد اعتماد به نفس

۹. یک نوع بازی فکری

۱۰. بهبود حواس پرتی

۱۱. افزایش توانمندی‌های ذهنی

۱۲. تقویت مهارت حل مسأله

۱۳. تقویت حافظه فعال

۱۴. افزایش سرعت عمل کودک

۱۵. تقویت سرعت پردازش،

برنامه‌ریزی، تصمیم‌گیری و اقدام

۱۶ و در نهایت با پرورش و تقویت

مهارتهای فوق کودک دریادگیری

مواد درسی و مهارت املانویسی،

خواندن و ریاضی رشد خواهد

یافت.

این پروژه تنها با استفاده از یک کتابخانه نوشته شده است ، که آن کتابخانه windows.h است ، کتابخانه

windows.h در واقع یک کتابخانه گرافیکی است ، که ما با استفاده از این کتابخانه ، پروژه را به صورت گرافیکی پیاده سازی

کرده ایم ، در ادامه در رابطه با نوع پیاده سازی پروژه به صورت گرافیکی مطالبی را ارائه خواهیم داد ، که کد اصلی کل

پروژه همین کد است و کد های قبلی فقط برای بدست آوردن داده مورد نیاز پروژه زده شده بود.

موارد بالا از جمله داده ها ، پارامترها و ماکرو بود که خوب است قبل از آنکه به بررسی کد بپردازیم آن را مطالعه کرده تا به یا

نحوه عملکرد آنها آشنا شوید ، و بدانید که هر کدام چه هستند و چه کاربردی در کد ما دارند ، پس قبل از بررسی کد به موارد

زیر می پردازیم :

LRESULT : نوعی داده است ، برای برگرداندن مقدار در windows API

CALLBACK : یک ماکرو که این تابع برای پردازش رویدادها استفاده می شود.

HWND : نوعی داده که یک هندل به یک پنجره در windows API است.

UINT : نوعی داده بدون علامت صحیح که معمولاً برای شناسایی نوع پیام ارسال شده به پنجره استفاده می شود.

WPARAM : نوعی داده برای نگهداری اطلاعات اضافی پیام است.

LPARAM : نوعی داده که نسبت به WPARAM اندازه بزرگتری از داده را نگهداری می کند ؛ ولی همان کاربرد را دارد.

WINAPI : یک نوع فراخوانی که مقادیر ورودی به تابع WinMain را بررسی می کند ، همچنین پارامترهای ارسالی و ...

HINSTANCE : پارامتر اول ، به شما اجازه دسترسی به منابع را می دهد.

HINSTANCE : پارامتر دوم نشانگر نمونه والد است که در اکثر برنامه ها نادیده گرفته می شود.

LPSTR : پارامتر سوم یک رشته است که معمولاً برای آرگومان های خط فرمان استفاده می شود.

nShowCmd : پارامتر چهارم تعیین می کند که پنجره چگونه نمایش داده شود.

حال که با موارد بالا آشنا شده اید می توانیم در ادامه به بررسی کد پروژه بپردازیم.

تابع کشیدن شکل

در حین اجرای بازی ما نیاز داریم که یک شی در پنجره تعریف شده باشد و آن شی در حین اجرای بازی وقتی کاربر روی دکمه مورد نظرش می زند آن شی مکانش تغییر کند؛ به همین دلیل تابع زیر که مربوط به کشیدن آن شی است را تعریف کرده ایم.

تابع زیر برای کشیدن یک مستطیل طراحی شده است، پارامتر HDC که برای این تابع مشخص شده است در واقع در درون خود اطلاعات مربوط به دستگاه گرافیکی را نگه داری می کند.

```
void Draw(HDC hdc){  
    Rectangle(hdc, pos_0_x, pos_0_y, pos_0_x + 10, pos_0_y + 10);  
}
```

ساختن Label

برای آنکه به پروژه یک هویت کلاسی بدهیم (به این معنا که این پروژه برای یک درس مشخص طراحی شده است این لیبل را ایجاد کرده ایم.

```
case WM_CREATE: {  
    HRGN hRgn = CreateRoundRectRgn(20, 20, 400, 50, 50, 50);  
    HWND Label = CreateWindow("STATIC", "Maze_Project_for_BP_Class|Teacher : Taybeh Rafiei|۱۴۰۲-۱۴۰۴",  
    WS_VISIBLE | WS_CHILD, ۱۵, ۵, ۱۰۴۰, ۲۰,  
    hwnd, nullptr, nullptr, nullptr);  
    SetWindowRgn(Label, hRgn, TRUE);  
    break;  
}
```

تنظیم سایز پنجره

برای آنکه کاربر نتواند اندازه پنجره را تغییر دهد ما یک اندازه مشخص برای پنجره تعیین کرده ایم. و زمانی که

اندازه پنجره تغییر کند این پیام (WM_SIZE) به برنامه ارسال می شود.

```
case WM_SIZE:  
    SetWindowPos(hwnd, NULL, ۰, ۰, ۱۰۸۰, ۷۲۰, SWP_NOZORDER);  
    return ۰;
```

```
2875 void Draw(HDC hdc){  
2876     Rectangle(hdc, pos_0_x, pos_0_y, pos_0_x + 10, pos_0_y + 10);  
2877 }  
2878  
2879 LRESULT CALLBACK Window_and_Print_Board(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam) {  
2880     HDC hdc;  
2881     switch (uMsg) {  
2882     case WM_CREATE: {  
2883  
2884         HRGN hRgn = CreateRoundRectRgn(20, 20, 400, 50, 50, 50);  
2885         HWND Label = CreateWindow("STATIC", " Maze_Project_for_BP_Class  
2886             WS_VISIBLE | WS_CHILD, 15, 5, 1040, 20,  
2887             hwnd, nullptr, nullptr, nullptr);  
2888         SetWindowRgn(Label, hRgn, TRUE);  
2889         break;  
2890     }  
2891  
2892  
2893     case WM_DESTROY:  
2894         PostQuitMessage(0);  
2895         return 0;  
2896  
2897     case WM_SIZE:  
2898         SetWindowPos(hwnd, NULL, 0, 0, 1080, 720, SWP_NOZORDER);  
2899         return 0;  
2900  
2901  
2902  
2903  
2904  
2905  
2906  
2907  
2908  
2909  
2910  
2911  
2912  
2913  
2914  
2915  
2916  
2917  
2918  
2919  
2920  
2921  
2922  
2923  
2924  
2925  
2926  
2927  
2928  
2929  
2930  
2931  
2932  
2933  
2934  
2935  
2936  
2937  
2938  
2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953  
2954  
2955  
2956  
2957  
2958  
2959  
2960  
2961  
2962  
2963  
2964  
2965  
2966  
2967  
2968  
2969  
2970  
2971  
2972  
2973  
2974  
2975  
2976  
2977  
2978  
2979  
2980  
2981  
2982  
2983  
2984  
2985  
2986  
2987  
2988  
2989  
2990  
2991  
2992  
2993  
2994  
2995  
2996  
2997  
2998  
2999  
3000  
3001  
3002  
3003  
3004  
3005  
3006  
3007  
3008  
3009  
3010  
3011  
3012  
3013  
3014  
3015  
3016  
3017  
3018  
3019  
3020  
3021  
3022  
3023  
3024  
3025  
3026  
3027  
3028  
3029  
3030  
3031  
3032  
3033  
3034  
3035  
3036  
3037  
3038  
3039  
3040  
3041  
3042  
3043  
3044  
3045  
3046  
3047  
3048  
3049  
3050  
3051  
3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082  
3083  
3084  
3085  
3086  
3087  
3088  
3089  
3090  
3091  
3092  
3093  
3094  
3095  
3096  
3097  
3098  
3099  
3100  
3101  
3102  
3103  
3104  
3105  
3106  
3107  
3108  
3109  
3110  
3111  
3112  
3113  
3114  
3115  
3116  
3117  
3118  
3119  
3120  
3121  
3122  
3123  
3124  
3125  
3126  
3127  
3128  
3129  
3130  
3131  
3132  
3133  
3134  
3135  
3136  
3137  
3138  
3139  
3140  
3141  
3142  
3143  
3144  
3145  
3146  
3147  
3148  
3149  
3150  
3151  
3152  
3153  
3154  
3155  
3156  
3157  
3158  
3159  
3160  
3161  
3162  
3163  
3164  
3165  
3166  
3167  
3168  
3169  
3170  
3171  
3172  
3173  
3174  
3175  
3176  
3177  
3178  
3179  
3180  
3181  
3182  
3183  
3184  
3185  
3186  
3187  
3188  
3189  
3190  
3191  
3192  
3193  
3194  
3195  
3196  
3197  
3198  
3199  
3200  
3201  
3202  
3203  
3204  
3205  
3206  
3207  
3208  
3209  
3210  
3211  
3212  
3213  
3214  
3215  
3216  
3217  
3218  
3219  
3220  
3221  
3222  
3223  
3224  
3225  
3226  
3227  
3228  
3229  
3230  
3231  
3232  
3233  
3234  
3235  
3236  
3237  
3238  
3239  
3240  
3241  
3242  
3243  
3244  
3245  
3246  
3247  
3248  
3249  
3250  
3251  
3252  
3253  
3254  
3255  
3256  
3257  
3258  
3259  
3260  
3261  
3262  
3263  
3264  
3265  
3266  
3267  
3268  
3269  
3270  
3271  
3272  
3273  
3274  
3275  
3276  
3277  
3278  
3279  
3280  
3281  
3282  
3283  
3284  
3285  
3286  
3287  
3288  
3289  
3290  
3291  
3292  
3293  
3294  
3295  
3296  
3297  
3298  
3299  
3300  
3301  
3302  
3303  
3304  
3305  
3306  
3307  
3308  
3309  
3310  
3311  
3312  
3313  
3314  
3315  
3316  
3317  
3318  
3319  
3320  
3321  
3322  
3323  
3324  
3325  
3326  
3327  
3328  
3329  
3330  
3331  
3332  
3333  
3334  
3335  
3336  
3337  
3338  
3339  
3340  
3341  
3342  
3343  
3344  
3345  
3346  
3347  
3348  
3349  
3350  
3351  
3352  
3353  
3354  
3355  
3356  
3357  
3358  
3359  
3360  
3361  
3362  
3363  
3364  
3365  
3366  
3367  
3368  
3369  
3370  
3371  
3372  
3373  
3374  
3375  
3376  
3377  
3378  
3379  
3380  
3381  
3382  
3383  
3384  
3385  
3386  
3387  
3388  
3389  
3390  
3391  
3392  
3393  
3394  
3395  
3396  
3397  
3398  
3399  
3400  
3401  
3402  
3403  
3404  
3405  
3406  
3407  
3408  
3409  
3410  
3411  
3412  
3413  
3414  
3415  
3416  
3417  
3418  
3419  
3420  
3421  
3422  
3423  
3424  
3425  
3426  
3427  
3428  
3429  
3430  
3431  
3432  
3433  
3434  
3435  
3436  
3437  
3438  
3439  
3440  
3441  
3442  
3443  
3444  
3445  
3446  
3447  
3448  
3449  
3450  
3451  
3452  
3453  
3454  
3455  
3456  
3457  
3458  
3459  
3460  
3461  
3462  
3463  
3464  
3465  
3466  
3467  
3468  
3469  
3470  
3471  
3472  
3473  
3474  
3475  
3476  
3477  
3478  
3479  
3480  
3481  
3482  
3483  
3484  
3485  
3486  
3487  
3488  
3489  
3490  
3491  
3492  
3493  
3494  
3495  
3496  
3497  
3498  
3499  
3500  
3501  
3502  
3503  
3504  
3505  
3506  
3507  
3508  
3509  
3510  
3511  
3512  
3513  
3514  
3515  
3516  
3517  
3518  
3519  
3520  
3521  
3522  
3523  
3524  
3525  
3526  
3527  
3528  
3529  
3530  
3531  
3532  
3533  
3534  
3535  
3536  
3537  
3538  
3539  
3540  
3541  
3542  
3543  
3544  
3545  
3546  
3547  
3548  
3549  
3550  
3551  
3552  
3553  
3554  
3555  
3556  
3557  
3558  
3559  
3560  
3561  
3562  
3563  
3564  
3565  
3566  
3567  
3568  
3569  
3570  
3571  
3572  
3573  
3574  
3575  
3576  
3577  
3578  
3579  
3580  
3581  
3582  
3583  
3584  
3585  
3586  
3587  
3588  
3589  
3590  
3591  
3592  
3593  
3594  
3595  
3596  
3597  
3598  
3599  
3600  
3601  
3602  
3603  
3604  
3605  
3606  
3607  
3608  
3609  
3610  
3611  
3612  
3613  
3614  
3615  
3616  
3617  
3618  
3619  
3620  
3621  
3622  
3623  
3624  
3625  
3626  
3627  
3628  
3629  
3630  
3631  
3632  
3633  
3634  
3635  
3636  
3637  
3638  
3639  
3640  
3641  
3642  
3643  
3644  
3645  
3646  
3647  
3648  
3649  
3650  
3651  
3652  
3653  
3654  
3655  
3656  
3657  
3658  
3659  
3660  
3661  
3662  
3663  
3664  
3665  
3666  
3667  
3668  
3669  
3670  
3671  
3672  
3673  
3674  
3675  
3676  
3677  
3678  
3679  
3680  
3681  
3682  
3683  
3684  
3685  
3686  
3687  
3688  
3689  
3690  
3691  
3692  
3693  
3694  
3695  
3696  
3697  
3698  
3699  
3700  
3701  
3702  
3703  
3704  
3705  
3706  
3707  
3708  
3709  
3710  
3711  
3712  
3713  
3714  
3715  
3716  
3717  
3718  
3719  
3720  
3721  
3722  
3723  
3724  
3725  
3726  
3727  
3728  
3729  
3730  
3731  
3732  
3733  
3734  
3735  
3736  
3737  
3738  
3739  
3740  
3741  
3742  
3743  
3744  
3745  
3746  
3747  
3748  
3749  
3750  
3751  
3752  
3753  
3754  
3755  
3756  
3757  
3758  
3759  
3760  
3761  
3762  
3763  
3764  
3765  
3766  
3767  
3768  
3769  
3770  
3771  
3772  
3773  
3774  
3775  
3776  
3777  
3778  
3779  
3780  
3781  
3782  
3783  
3784  
3785  
3786  
3787  
3788  
3789  
3790  
3791  
3792  
3793  
3794  
3795  
3796  
3797  
3798  
3799  
3800  
3801  
3802  
3803  
3804  
3805  
3806  
3807  
3808  
3809  
3810  
3811  
3812  
3813  
3814  
3815  
3816  
3817  
3818  
3819  
3820  
3821  
3822  
3823  
3824  
3825  
3826  
3827  
3828  
3829  
3830  
3831  
3832  
3833  
3834  
3835  
3836  
3837  
3838  
3839  
3840  
3841  
3842  
3843  
3844  
3845  
3846  
3847  
3848  
3849  
3850  
3851  
3852  
3853  
3854  
3855  
3856  
3857  
3858  
3859  
3860  
3861  
3862  
3863  
3864  
3865  
3866  
3867  
3868  
3869  
3870  
3871  
3872  
3873  
3874  
3875  
3876  
3877  
3878  
3879  
3880  
3881  
3882  
3883  
3884  
3885  
3886  
3887  
3888  
3889  
3890  
3891  
3892  
3893  
3894  
3895  
3896  
3897  
3898  
3899  
3900  
3901  
3902  
3903  
3904  
3905  
3906  
3907  
3908  
3909  
3910  
3911  
3912  
3913  
3914  
3915  
3916  
3917  
3918  
3919  
3920  
3921  
3922  
3923  
3924  
3925  
3926  
3927  
3928  
3929  
3930  
3931  
3932  
3933  
3934  
3935  
3936  
3937  
3938  
3939  
3940  
3941  
3942  
3943  
3944  
3945  
3946  
3947  
3948  
3949  
3950  
3951  
3952  
3953  
3954  
3955  
3956  
3957  
3958  
3959  
3960  
3961  
3962  
3963  
3964  
3965  
3966  
3967  
3968  
3969  
3970  
3971  
3972  
3973  
3974  
3975  
3976  
3977  
3978  
3979  
3980  
3981  
3982  
3983  
3984  
3985  
3986  
3987  
3988  
3989  
3990  
3991  
3992  
3993  
3994  
3995  
3996  
3997  
3998  
3999  
4000  
4001  
4002  
4003  
4004  
4005  
4006  
4007  
4008  
4009  
4010  
4011  
4012  
4013  
4014  
4015  
4016  
4017  
4018  
4019  
4020  
4021  
4022  
4023  
4024  
4025  
4026  
4027  
4028  
4029  
4030  
4031  
4032  
4033  
4034  
4035  
4036  
4037  
4038  
4039  
4040  
4041  
4042  
4043  
4044  
4045  
4046  
4047  
4048  
4049  
4050  
4051  
4052  
4053  
4054  
4055  
4056  
4057  
4058  
4059  
4060  
4061  
4062  
4063  
4064  
4065  
4066  
4067  
4068  
4069  
4070  
4071  
4072  
4073  
4074  
4075  
4076  
4077  
4078  
4079  
4080  
4081  
4082  
4083  
4084  
4085  
4086  
4087  
4088  
4089  
4090  
4091  
4092  
4093  
4094  
4095  
4096  
4097  
4098  
4099  
4100  
4101  
4102  
4103  
4104  
4105  
4106  
4107  
4108  
4109  
4110  
4111  
4112  
4113  
4114  
4115  
4116  
4117  
4118  
4119  
4120  
4121  
4122  
4123  
4124  
4125  
4126  
4127  
4128  
4129  
4130  
4131  
4132  
4133  
4134  
4135  
4136  
4137  
4138  
4139  
4140  
4141  
4142  
4143  
4144  
4145  
4146  
4147  
4148  
4149  
4150  
4151  
4152  
4153  
4154  
4155  
4156  
4157  
4158  
4159  
4160  
4161  
4162  
4163  
4164  
4165  
4166  
4167  
4168  
4169  
4170  
4171  
4172  
4173  
4174  
4175  
4176  
4177  
4178  
4179  
4180  
4181  
4182  
4183  
4184  
4185  
4186  
4187  
4188  
4189  
4190  
4191  
4192  
4193  
4194  
4195  
4196  
4197  
4198  
4199  
4200  
4201  
4202  
4203  
4204  
4205  
4206  
4207  
4208  
4209  
4210  
4211  
4212  
4213  
4214  
4215  
4216  
4217  
4218  
4219  
4220  
4221  
4222  
4223  
4224  
4225  
4226  
4227  
4228  
4229  
4230  
4231  
4232  
4233  
4234  
4235  
4236  
4237  
4238  
4239  
4240  
4241  
4242  
4243  
4244  
4245  
4246  
4247  
4248  
4249  
4250  
4251  
4252  
4253  
4254  
4255  
4256  
4257  
4258  
4259  
4260  
4261  
4262  
4263  
4264  
4265  
4266  
4267  
4268  
4269  
4270  
4271  
4272  
4273  
4274  
4275  
4276  
4277  
4278  
4279  
4280  
4281  
4282  
4283  
4284  
4285  
4286  
4287  
4288  
4289  
4290  
4291  
4292  
4293  
4294  
4295  
4296  
4297  
4298  
4299  
4300  
4301  
4302  
4303  
4304  
4305  
4306  
4307  
4308  
4309  
4310  
4311  
4312  
4313  
4314  
4315  
4316  
4317  
4318  
4319  
4320  
4321  
4322  
4323  
4324  
4325  
4326  
4327  
4328  
4329  
4330  
4331  
4332  
4333  
4334  
4335  
4336  
4337  
4338  
4339  
4340  
4341  
4342  
4343  
4344  
4345  
4346  
4347  
4348  
4349  
4350  
4351  
4352  
4353  
4354  
4355  
4356  
4357  
4358  
4359  
4360  
4361  
4362  
4363  
4364  
4365  
4366  
4367  
4368  
4369  
4370  
4371  
4372  
4373  
4374  
4375  
4376  
4377  
4378  
4379  
4380  
4381  
4382  
4383  
4384  
4385  
4386  
4387  
4388  
4389  
4390  
4391  
4392  
4393  
4394  
4395  
4396  
4397  
4398  
4399  
4400  
4401  
4402  
4403  
4404  
4405  
4406  
4407  
4408  
4409  
4410  
4411  
4412  
4413  
4414  
4415  
4416  
4417  
4418  
4419  
4420  
4421  
4422  
4423  
4424  
4425  
4426  
4427  
4428  
4429  
4430  
4431  
4432  
4433  
4434  
4435  
4436  
4437  
4438  
4439  
4440  
4441  
4442  
4443  
4444  
4445  
4446  
4447  
4448  
4449  
4450  
4451  
4452  
4453  
4454  
4455  
4456  
4457  
4458  
4459  
4460  
4461  
4462  
4463  
4464  
4465  
4466  
4467  
4468  
4469  
4470  
4471  
4472  
4473  
4474  
4475  
4476  
4477  
4478  
4479  
4480  
4481  
4482  
4483  
4484  
4485  
4486  
4487  
4488  
4489  
4490  
4491  
4492  
4493  
4494  
4495  
4496  
4497  
4498  
4499  
4500  
4501  
4502  
4503  
4504  
4505  
4506  
4507  
4508  
4509  
4510  
4511  
4512  
4513  
4514  
4515  
4516  
4517  
4518  
4519  
4520  
4521  
4522  
4523  
4524  
4525  
4526  
4527  
4528  
4529  
4530  
4531  
4532  
4533  
4534  
4535  
4536  
4537  
4538  
4539  
4540  
4541  
4542  
4543  
4544  
4545  
4546  
4547  
4548  
4549  
4550  
4551  
4552  
4553  
4554  
4555  
4556  
4557  
4558  
4559  
4560  
4561  
4562  
4563  
4564  
4565  
4566  
4567  
4568  
4569  
4570  
4571  
4572  
4573  
4574  
4575  
4576  
4577  
4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585  
4586  
4587  
4588  
4589  
4590  
4591  
4592  
4593  
4594  
4595  
4596  
4597  
4598  
4599  
4600  
4601  
4602  
4603  
4604  
4605  
4606  
4607  
4608  
4609  
4610  
4611  
4612  
4613  
4614  
4615  
4616  
4617  
4618  
4619  
4620  
4621  
4622  
4623  
4624  
4625  
4626  
4627  
4628  
4629  
4630  
4631  
4632  
4633  
4634  
4635  
4636  
4637  
4638  
4639  
4640  
4641  
4642  
4643  
4644  
4645  
4646  
4647  
4648  
4649  
4650  
4651  
4652  
4653  
4654  
4655  
4656  
4657  
4658  
4659  
4660  
4661  
4662  
4663  
4664  
4665  
4666  
4667  
4668  
4669  
4670  
4671  
4672  
4673  
4674  
4675  
4676  
4677  
4678  
4679  
4680  
4681  
4682  
4683  
4684  
4685  
4686  
4687  
4688  
4689  
4690  
4691  
4692  
4693  
4694  
4695  
4696  
4697  
4698  
4699  
4700  
4701  
4702  
4703  
4704  
4705  
4706  
4707  
4708  
4709  
4710  
4711  
4712  
4713  
4714  
4715  
4716  
4717  
4718  
4719  
4720  
4721  
4722  
4723  
4724  
4725  
4726  
4727  
4728  
4729  
4730  
4731  
4732  
4733  
4734  
4735  
4736  
4737  
4738  
4739  
4740  
4741  
4742  
4743  
4744  
4
```

تنظیم کشیدن خطوط درون پنجره

مواردی که درون پنجره کشیده می شود، به چهار دسته تقسیم می شود، دسته اول آن خطی است، که زیر Label برای زیبایی آن قرار داده شده است، دسته دوم همان شکلی است، که درون پنجره از پایین به بالا با دستور کاربر حرکت می کند، تا کاربر بتواند بازی را انجام دهد، دسته سوم همان همان برد بازی است، که اگر به آن توجه کنید ابتدا مختصات موانع برد را دریافت می کند، سپس برد را بر اساس آن مختصات می کشد، و دسته چهارم، همان موردی است، که در زمان انجام بازی ما حرکات کاربر را در یک آرایه دو بعدی ذخیره می کنیم و سپس در پایان بازی، اگر کاربر برنده شد، مسیر حرکت کاربر را نشان می دهد.

case WM_PAINT:

```
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hwnd, &ps);

    HPEN hPenRed = CreatePen(PS_SOLID, 5, RGB(255, 0, 0));
    SelectObject(hdc, hPenRed);

    MoveToEx(hdc, 17, 25, NULL);

    LineTo(hdc, 1052, 25);

    Draw(hdc);

    HPEN Maze_Board_G = CreatePen(PS_SOLID, 10, RGB(40, 152, 7));

    for (int i = 0; i < 1161; i++){
        if ((XY[i][1] > 65) and (XY[i][0] > 66)){
            SelectObject(hdc, Maze_Board_G);

            MoveToEx(hdc, XY[i][0], XY[i][1], NULL);

            LineTo(hdc, XY[i][2], XY[i][3]);
        }
    }

    if (((pos_O_x > 496) and (pos_O_x < 516)) and (pos_O_y == 75)){
        HPEN Maze_Board_B = CreatePen(PS_SOLID, 7, RGB(0, 0, 255));

        for (int i = 0; i < 1161; i++){
            SelectObject(hdc, Maze_Board_B);

            MoveToEx(hdc, W_P[i][0], W_P[i][1], NULL);

            LineTo(hdc, W_P[i][2], W_P[i][3]);
        }
    }

    EndPaint(hwnd, &ps);

    Sleep(500);

    break;
}
```

آیا می دانید که:

پرورش مهارت حل مسئله به کودکان کمک میکند تا در رویارویی با چالش و مشکلات پیش رو بهترین عملکرد را داشته باشند بازی ماز برای کودکان این امکان را ایجاد میکند تا کودکان بهترین استراتژی و برنامه ریزی را و ایجاد راه حل های مختلف در زمانی که به بن بست می خورند را انتخاب نمایند به عبارتی بازی ماز عملکرد های اجرایی کودک را بهبود می بخشد. برای تقویت مهارت های حل مسئله کودک پیشنهاد می شود حتما این مقاله را مطالعه کنید.

```
2901 case WM_PAINT: {
2902     PAINTSTRUCT ps;
2903     HDC hdc = BeginPaint(hwnd, &ps);
2904
2905     HPEN hPenRed = CreatePen(PS_SOLID, 5, RGB(255, 0, 0));
2906     SelectObject(hdc, hPenRed);
2907     MoveToEx(hdc, 17, 25, NULL);
2908     LineTo(hdc, 1052, 25);
2909
2910     Draw(hdc);
2911
2912     HPEN Maze_Board_G = CreatePen(PS_SOLID, 10, RGB(40, 152, 7));
2913     for (int i = 0; i < 1161; i++){
2914         if ((XY[i][1] > 65) and (XY[i][0] > 66)){
2915             SelectObject(hdc, Maze_Board_G);
2916             MoveToEx(hdc, XY[i][0], XY[i][1], NULL);
2917             LineTo(hdc, XY[i][2], XY[i][3]);
2918         }
2919     }
2920     if (((pos_O_x > 496) and (pos_O_x < 516)) and (pos_O_y == 75)){
2921         HPEN Maze_Board_B = CreatePen(PS_SOLID, 7, RGB(0, 0, 255));
2922         for (int i = 0; i < 1161; i++){
2923             SelectObject(hdc, Maze_Board_B);
2924             MoveToEx(hdc, W_P[i][0], W_P[i][1], NULL);
2925             LineTo(hdc, W_P[i][2], W_P[i][3]);
2926         }
2927     }
2928
2929     EndPaint(hwnd, &ps);
2930     Sleep(500);
2931     break;
2932 }
```

شکل ۸-۱ این شکل یک، نمای کلی از کد کشیدن خطوط درون پنجره است.

تنظیم دکمه های ورودی و عملکرد آن

برای این پروژه ۵ دکمه تعریف شده است؛ که ۴ دکمه برای بالا، پایین و چپ و راست رفتن کاربر تنظیم شده است و همچنین یک دکمه برای پرش کاربر تنظیم شده است؛ اما وقتی که کاربر بر روی یک دکمه کلیک کند، ما قبل از آنکه بررسی کنیم که کاربر می خواهد به چه جهتی برود، ابتدا بررسی می کنیم، که کاربر آیا به پایان مسیر رسیده است یا خیر در واقع این بررسی کار ما را در طراحی بازی راحت می کند و باعث می شود ما به راحتی بفهمیم؛ که آیا کاربر برنده شده است، یا خیر؟

این قسمت برای بررسی برد کاربر تنظیم شده است :

```
if (((pos_O_x > ۴۹۶) and (pos_O_x < ۵۱۶)) and (pos_O_y == ۷۵)){
    MessageBox(hwnd, "Congratulations, champion!", "Winner", MB_OK);
    Sleep(۲۰۰۰);
    ExitProcess(۰);
}
```

```
2934 if (((pos_O_x > 496) and (pos_O_x < 516)) and (pos_O_y == 75)){
2935     MessageBox(hwnd, "Congratulations, champion!", "Winner", MB_OK);
2936     Sleep(2000);
2937     ExitProcess(0);
2938 }
```

شکل ۹-۱ این شکل یک، نمای کلی از کد بررسی برنده شدن کاربر در بازی است.

این قسمت برای بررسی وضعیت پرش کاربر تنظیم شده است :

```
if (wParam == VK_RETURN){
    if (num == ۰) {
        num_enter = ۱;
        num++;
    }
    else {
        W_P[pos][۲] = pos_O_y;
        pos++;
        num_enter = ۰;
        MessageBox(hwnd, "Game Over!!!", "Noob", MB_OK);
        Sleep(۲۰۰۰);
        ExitProcess(۰);
    }
}
```

```
if (wParam == VK_RETURN){
    if (num == 0) {
        num_enter = 1;
        num++;
    }
    else {
        W_P[pos][3] = pos_O_y;
        pos++;
        num_enter = 0;
        MessageBox(hwnd, "Game Over!!!", "Noob", MB_OK);
        Sleep(2000);
        ExitProcess(0);
    }
}
```

شکل ۱۰-۱ این شکل یک، نمای کلی از کد بررسی پرش کاربر در بازی است.

این قسمت برای بررسی وضعیت حرکت کاربر به سمت راست تنظیم شده است :

```
else if (wParam == VK_RIGHT){
    if (pos_O_x < X_۱){
        if (num_enter == ۱){
            W_P[pos][۰] = pos_O_x, W_P[pos][۱] = pos_O_y, W_P[pos][۲] = pos_O_y;
            pos_O_x += ۱۵;
            W_P[pos][۲] = pos_O_x;
            num_enter = ۰;
            pos++;
        }
    }
}
```

```
2953 else if (wParam == VK_RIGHT){
2954     if (pos_O_x < X_1){
2955         if (num_enter == 1){
2956             W_P[pos][0] = pos_O_x, W_P[pos][1] = pos_O_y, W_P[pos][3] = pos_O_y;
2957             pos_O_x += 15;
2958             W_P[pos][2] = pos_O_x;
2959             num_enter = 0;
2960             pos++;
2961         }
2962     }
2963 }
```

شکل ۱۱-۱ این شکل یک، نمای کلی از کد بررسی حرکت کاربر در بازی است.

آیا می دانید که:

در مهارت های دیداری چشم

عضو اصلی آن است حال

کودکانی که در این مهارت ضعیف

عمل میکند در دوران مدرسه در

برخی از دروس دچار مشکل می

شوند البته حافظه دیداری با

تمرین میتوان بهبود بخشید حال

با توجه به این نکته این بازی باید

دقت بالایی داشته باشید و کودک

کاوش زیبای برای پیدا کردن

راحل انتخابی میکنند این امر

سبب میشود تا کودکان بتوانند به

تعادل و کنترل مهارت های

حرکتی و چشمی دست یابند.

آیا می دانید که:

مهارت های مهم مغز مانند فکر کردن، خواندن، یادگیری استدلال و توجه تمامی از مهارت های شناختی محسوب میشود پیچ و خم های ماز به بهبود مهارت های شناختی کمک می کند کودکان قبل از شروع بازی راه حلهای خود را بر نامه ریزی و استدلال می کنند چرا که ماز های همانند حل معمایی هستند که نیاز به فکر و استدلال دارد در واقع مجموعه چالش های رو به رو به تقویت مهارت شناختی کودک کمک میکند.

```

2962     else {
2963         int pos_x = pos_O_x;
2964         for (int i = 0 ; i < 1691 ; i++){
2965             W_P[pos][0] = pos_O_x;W_P[pos][2] = pos_O_y;W_P[pos][3] = pos_O_y;
2966             if ((pos_O_y >= S[i][2] - 25) and (pos_O_y <= S[i][2])){
2967                 if ((pos_O_x >= S[i][0]) and (pos_O_x <= S[i][1])){
2968                     pos_O_x += 5;
2969                     W_P[pos][2] = pos_O_x;
2970                     pos++;
2971                     break;
2972                 }
2973             }
2974         }
2975         if (pos_x == pos_O_x){
2976             if(num == 0){
2977                 MessageBox(hwnd, "Please press Enter_key then press (Up_Down_Left_Right)_key", "Warning!!!", MB_OK);
2978             }
2979             else {
2980                 W_P[pos][3] = pos_O_y;
2981                 pos++;
2982                 MessageBox(hwnd, "Game Over!!!", "Noob", MB_OK);
2983                 Sleep(2000);
2984                 ExitProcess(0);
2985             }
2986         }
2987     }
2988 }

```

این قسمت برای بررسی وضعیت حرکت کاربر به سمت چپ تنظیم شده است:

شکل ۱۳-۱ این شکل یک، نمای کلمه از کد بر رسم حرکت کاربر در بازی است.

آیا می دانید که:

پیچ و خم های ماز کمی پیچیدگی دارند و گاهی نیاز دارند که بچه ها برای حل آنها وقت بگذارند این امر سبب میشود تا کودکان ارزش پشت کار را درک کنند و در حین بازی می آموزند که تسلیم شدند و راحل های احتمالی پیش رو امتحان کنند و در نهایت در برای چالش ها صبوری کنند.

```
int pos_x1 = pos_O_x;
for (int i = ۰; i < ۱۶۹۱; i++){
    W_P[pos][۰] = pos_O_x.W_P[pos][۱] = pos_O_y.W_P[pos][۲] = pos_O_y;
    if ((pos_O_y > S[i][۲] - ۲۵) and (pos_O_y < S[i][۲])){
        if ((pos_O_x >= S[i][۰] + ۵) and (pos_O_x <= S[i][۱])){
            pos_O_x -= ۵;
            W_P[pos][۲] = pos_O_x;
            pos++;
            break;
        }
    }
}
if (pos_x1 == pos_O_x){
    if(num == ۰){
        MessageBox(hwnd, "Please press Enter_key then press (Up_Down_Left_Right)_key", "Warning!!!", MB_OK);
    }
    else {
        W_P[pos][۲] = pos_O_y;
        pos++;
        MessageBox(hwnd, "Game Over!!!", "Noob", MB_OK);
        Sleep(۲۰۰۰);
        ExitProcess(۰);
    }
}
}
```

```
2999     else {
3000         int pos_x1 = pos_O_x;
3001         for (int i = 0 ; i < 1691 ; i++){
3002             W_P[pos][0] = pos_O_x.W_P[pos][1] = pos_O_y.W_P[pos][3] = pos_O_y;
3003             if ((pos_O_y > S[i][2] - 25) and (pos_O_y < S[i][2])){
3004                 if ((pos_O_x >= S[i][0] + 5) and (pos_O_x <= S[i][1])){
3005                     pos_O_x -= 5;
3006                     W_P[pos][2] = pos_O_x;
3007                     pos++;
3008                     break;
3009                 }
3010             }
3011         }
3012         if (pos_x1 == pos_O_x){
3013             if(num == 0){
3014                 MessageBox(hwnd, "Please press Enter_key then press (Up_Down_Left_Right)_key", "Warning!!!", MB_OK);
3015             }
3016             else {
3017                 W_P[pos][3] = pos_O_y;
3018                 pos++;
3019                 MessageBox(hwnd, "Game Over!!!", "Noob", MB_OK);
3020                 Sleep(2000);
3021                 ExitProcess(0);
3022             }
3023         }
3024     }
3025 }
3026 }
```

شکل ۱-۱ این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

این قسمت برای بررسی وضعیت حرکت کاربر به سمت بالا تنظیم شده است :

```
else if (wParam == VK_UP){
    int n = ۰;
    if (pos_O_y > Y_۰) {
        if (num_enter == ۱){
            W_P[pos][۰] = pos_O_x.W_P[pos][۲] = pos_O_x.W_P[pos][۱] = pos_O_y;
            pos_O_y -= ۲۵;
            W_P[pos][۲] = pos_O_y;
            pos++;
            num_enter = ۰;
        }
        else {
            3027     else if (wParam == VK_UP){
            3028         int n = 0;
            3029         if (pos_O_y > Y_0) {
            3030             if (num_enter == 1){
            3031                 W_P[pos][0] = pos_O_x.W_P[pos][2] = pos_O_x.W_P[pos][1] = pos_O_y;
            3032                 pos_O_y -= 25;
            3033                 W_P[pos][3] = pos_O_y;
            3034                 pos++;
            3035                 num_enter = 0;
            3036             }
            3037             else {
```

شکل ۱-۵ این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

آیا می دانید که:

بازی ماز برای کودکان فعالیت خوبی است که اعتماد به نفسشان را بالا ببرد به این گونه ابتدا شما از مراحل آسان و مبتدی این بازی را برای فرزند خود تهیه کنید حال با بالا رفتن مهارت او در این بازی مراحل پیچیده تر را در اختیار وی قرار دهید کودکان با حل بازی ماز در هر سطحی با این حقیقت که انها توانسته اند این بازی را حل کند احساس غرور و موفقیت میکنند.

W_P[pos][۰] = pos_O_x.W_P[pos][۲] = pos_O_x.W_P[pos][۱] = pos_O_y;

for (int i = ۰; i < ۱۱۶۲; i++){

if ((pos_O_x >= XY[i][۰]) and (pos_O_x <= XY[i][۲])){

if ((pos_O_y >= XY[i][۱]) and (pos_O_y <= XY[i][۳])){

n = n + ۵;

break;

}

}

}

if (n == ۰) {

pos_O_y -= ۵;

W_P[pos][۳] = pos_O_y;

pos++;

}

else {

if (num == ۰){

MessageBox(hwnd, "Please press Enter_key then press (Up_Down_Left_Right)_key", "Warning!!!", MB_OK);

}

else{

pos_O_y;

W_P[pos][۳] = pos_O_y;

pos++;

MessageBox(hwnd, "Game Over!!!", "Noob", MB_OK);

Sleep(۲۰۰۰);

ExitProcess(۰);

}

}

}

}

}

```
3038 W_P[pos][0] = pos_O_x,W_P[pos][2] = pos_O_x,W_P[pos][1] = pos_O_y;
3039 for (int i = 0; i < 1162; i++){
3040     if ((pos_O_x >= XY[i][0]) and (pos_O_x <= XY[i][2])){
3041         if ((pos_O_y >= XY[i][1]) and (pos_O_y <= XY[i][3])){
3042             n = n + 5;
3043             break;
3044         }
3045     }
3046 }
3047 if (n == 0) {
3048     pos_O_y -= 5;
3049     W_P[pos][3] = pos_O_y;
3050     pos++;
3051 }
3052 else {
3053     if (num == 0){
3054         MessageBox(hwnd, "Please press Enter_key then press (Up_Down_Left_Right)_key", "Warning!!!", MB_OK);
3055     }
3056     else{
3057         pos_O_y;
3058         W_P[pos][3] = pos_O_y;
3059         pos++;
3060         MessageBox(hwnd, "Game Over!!!", "Noob", MB_OK);
3061         Sleep(2000);
3062         ExitProcess(0);
3063     }
3064 }
3065 }
3066 }
```

شکل ۱۶-۱ این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

این قسمت برای بررسی وضعیت حرکت کاربر به سمت پایین تنظیم شده است :

else if (wParam == VK_DOWN){

int n = ۰;

if (pos_O_y < Y_1) {

if (num_enter == 1){

W_P[pos][۰] = pos_O_x.W_P[pos][۲] = pos_O_x.W_P[pos][۱] = pos_O_y;

pos_O_y += ۲۵;

W_P[pos][۳] = pos_O_y;

pos++;

```
3068 else if (wParam == VK_DOWN){
3069     int n = 0;
3070     if (pos_O_y < Y_1) {
3071         if (num_enter == 1){
3072             W_P[pos][0] = pos_O_x,W_P[pos][2] = pos_O_x,W_P[pos][1] = pos_O_y;
3073             pos_O_y += 25;
3074             W_P[pos][3] = pos_O_y;
3075             pos++;
3076         }
3077     }
3078 }
```

شکل ۱۷-۱ این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

```

num_enter = ۰;
}

else {

W_P[pos][۰] = pos_O_x,W_P[pos][۲] = pos_O_x,W_P[pos][۱] = pos_O_y;

for (int i = ۰ ; i < ۱۱۶۱ ; i++){

if ((pos_O_x >= XY[i][۰]) and (pos_O_x <= XY[i][۲])){

if ((pos_O_y >= XY[i][۱]) and (pos_O_y <= XY[i][۳])){

n = n + ۱;

break;

}

}

}

if (n == ۰) {

pos_O_y += ۵;

W_P[pos][۲] = pos_O_y;

pos++;

}

else {

if (num == ۰){

MessageBox(hwnd, "Please press Enter_key then press (Up_Down_Left_Right)_key", "Warning!!!", MB_OK);

}

else{

pos_O_y;

W_P[pos][۲] = pos_O_y;

pos++;

MessageBox(hwnd, "Game Over!!!", "Noob", MB_OK);

Sleep(۲۰۰۰);

ExitProcess(۰);

}

}

}

}

}

InvalidateRect(hwnd, NULL, TRUE);

break;

```

آیا می دانید که:

استفاده در آموزش: مازها به عنوان یک ابزار آموزشی بسیار موثر شناخته شده‌اند. این بازی‌ها می‌توانند به کودکان کمک کنند تا مهارت‌های حل مسئله، تصمیم‌گیری و تفکر منطقی خود را بهبود بخشند.

تأثیر روانی: حل معماهای ماز می‌تواند تأثیر قوی بر روی فعالیت مغزی داشته باشد. این نوع بازی‌ها ممکن است به بهبود حافظه، تمرکز و افزایش توانایی‌های شناختی کمک کنند.

```

3076 num_enter = 0;
3077 }
3078 else {
3079 W_P[pos][0] = pos_O_x,W_P[pos][2] = pos_O_x,W_P[pos][1] = pos_O_y;
3080 for (int i = 0 ; i < 1161 ; i++){
3081 if ((pos_O_x >= XY[i][0]) and (pos_O_x <= XY[i][2])){
3082 if ((pos_O_y >= XY[i][1]) and (pos_O_y <= XY[i][3])){
3083 n = n + 1;
3084 break;
3085 }
3086 }
3087 }
3088 if (n == 0) {
3089 pos_O_y += 5;
3090 W_P[pos][3] = pos_O_y;
3091 pos++;
3092 }
3093 else {
3094 if (num == 0){
3095 MessageBox(hwnd, "Please press Enter_key then press (Up_Down_Left_Right)_key", "Warning!!!", MB_OK);
3096 }
3097 else{
3098 pos_O_y;
3099 W_P[pos][3] = pos_O_y;
3100 pos++;
3101 MessageBox(hwnd, "Game Over!!!", "Noob", MB_OK);
3102 Sleep(2000);
3103 ExitProcess(0);
3104 }
3105 }
3106 }
3107 }
3108 }

```

شکل ۱-۱۸ این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

تنظیم رنگ پس زمینه

ما برای رنگ پس زمینه از دستور زیر استفاده کرده ایم و رنگ پس زمینه را از حالت پیش فرض به رنگ دلخواه خود تغییر داده ایم.

case WM_ERASEBKGD:

```
{
    HDC hdc = (HDC)wParam;
    RECT rect;
    GetClientRect(hwnd, &rect);
    HBRUSH brush = CreateSolidBrush(180, 170, 120);
    FillRect(hdc, &rect, brush);
    DeleteObject(brush);
    return 1;
}

return 0;
```

```
3111 case WM_ERASEBKGD:
3112 {
3113     HDC hdc = (HDC)wParam;
3114     RECT rect;
3115     GetClientRect(hwnd, &rect);
3116     HBRUSH brush = CreateSolidBrush(180, 170, 120);
3117     FillRect(hdc, &rect, brush);
3118     DeleteObject(brush);
3119     return 1;
3120 }
3121 return 0;
3122 }
3123 return DefWindowProc(hwnd, uMsg, wParam, lParam);
3124 }
```

شکل ۱۸-۱ این شکل یک ، نمای کلی از کد تنظیم رنگ بازی است .

تنظیم تابع WinAPI

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE, LPSTR, int nShowCmd) {

```
const char CLASS_NAME[] = "Sample Window Class";
WNDCLASS wc = {};
wc.lpfnWndProc = Window_and_Print_Board;
wc.hInstance = hInstance;
wc.lpszClassName = CLASS_NAME;
RegisterClass(&wc);
HWND hwnd = CreateWindowEx(
    0,
    CLASS_NAME,
    "Maze Project",
    WS_OVERLAPPEDWINDOW & ~WS_MAXIMIZEBOX,
    CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
    NULL,
    NULL,
    hInstance,
    NULL
);
ShowWindow(hwnd, nShowCmd);
MSG msg;
while (GetMessage(&msg, NULL, 0, 0)) {
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
```

```
3126 int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE, LPSTR, int nShowCmd) {
3127     const char CLASS_NAME[] = "Sample Window Class";
3128
3129     WNDCLASS wc = {};
3130     wc.lpfnWndProc = Window_and_Print_Board;
3131     wc.hInstance = hInstance;
3132     wc.lpszClassName = CLASS_NAME;
3133
3134     RegisterClass(&wc);
3135
3136     HWND hwnd = CreateWindowEx(
3137         0,
3138         CLASS_NAME,
3139         "Maze Project",
3140         WS_OVERLAPPEDWINDOW & ~WS_MAXIMIZEBOX,
3141         CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
3142         NULL,
3143         NULL,
3144         hInstance,
3145         NULL
3146     );
3147
3148     ShowWindow(hwnd, nShowCmd);
3149
3150     MSG msg;
3151     while (GetMessage(&msg, NULL, 0, 0)) {
3152         TranslateMessage(&msg);
3153         DispatchMessage(&msg);
3154     }
3155
3156     return 0;
3157 }
```

شکل ۱۹-۱ این شکل یک ، نمای کلی از کد تابع winapi در بازی است .

منطق پروژه



وقتی دلیل به اندازه کافی بزرگ باشد، شما مایل خواهید بود که تقریباً هر روشی را امتحان کنید

منطق طراحی برد پروژه

برای طراحی برد بازی ماز روش های مختلفی وجود دارد، اما یکی از بهینه ترین روش ها آن است، که ابتدا بر اساس یک فرم کلی و خام و با استفاده از دستورات برنامه نویسی برد را بسازیم و در ادامه آن را در فایل ذخیره کنیم، حال به صورت دستی برد را به آنچه که دلخواه ما است، تبدیل کنیم؛ البته همین الان که در حال نوشتن این فایل توضیحات هستیم، ایده جدیدی برای طراحی برد پروژه بازی ما به نظر رسید، که آن هم این است، که برای ساخت برد یک الگوریتم بسازیم و، حال اون الگوریتم برد را طراحی کند، ولی برد این پروژه به صورت دستی طراحی شده است.

منطق تعیین مختصات موانع و فضای خالی برد

برای تعیین موانع برد، ابتدا، دو راستا را در نظر گرفتیم، یعنی راستای X و راستای Y و بعد از آن یک آرایه دو بعدی ساختیم و در آرایه دو بعدی، مختصات خطوط افقی و عمودی را گذاشتیم؛ نکته جالب توجه آن است که خطوط افقی و عمودی هر کدام دارای، خاصیتی متفاوت هستند، که آن خاصیت در خطوط افقی، آن است که مقدار Y برابر است؛ اما مقدار X ها نابرابر و ما برای X ها اینگونه در نظر گرفتیم که، مقدار X اولیه می شود همان X و بعد از آن مقدار X ثانویه می شود $X+10$ و به همین شیوه برای خطوط عمودی اینگونه در نظر گرفته ایم که خطوط عمودی، دارای Y های متفاوت هستند، به همین جهت، میزان X ها را یکسان گرفته و مقدار Y اولیه را Y گرفتیم و مقدار Y ثانویه را $Y - 25$ گرفتیم و سپس رو این روند یک حلقه زدیم تا بتوانیم خط به خط داده string را پیمایش کنیم و در آخر این را باید بدانیم که تعیین کرده ایم اگر در پیمایش جایی به کاراکتر '،' رسیدیم به این معنی است که رفتیم خط بعد پس شرطی برای آن تعیین کرده ایم.

که این شرط اینگونه عمل می کند که ، اگر درون یک خط از برد داشتیم پیمایش می گردیم کاراکتر ها را ، هر بار مقداری به اندازه ۱۰ به X ها اضافه می شود ، هر بار هم که رفقیم خط بعد ، ۲۵ واحد به Y ها کم می شود ، همچنین به این موضوع دقت داشته باشید ؛ که هر بار ما یک کاراکتر از رشته را بررسی می کنیم مختصات آن را به مقادیر مورد نظر که تبدیل کردیم ، سپس به برنامه دستور می دهیم که آن داده را در همان آرایه دو بعدی ذخیره کند ، سپس آن آرایه دو بعدی را در یک حلقه for پیمایش می کنیم و داده های آن را دقیقاً به صورت یک آرایه دو بعدی درون یک فایل ذخیره می کنیم ، که می توانید مقادیر آرایه های دو بعدی را در فایل های ضمیمه شده پروژه ببینید.

سپس مقادیر درون فایل را کپی می‌کنیم و درون فایل main پروژه خود پیاده‌سازی می‌کنیم، در آخر هم از آن در کد اصلی پروژه استفاده می‌کنیم، اما نکته قابل توجه در این موضوع آن است که، برای پیدا کردن، مختصات فضای خالی برد هم از همان روش قبل استفاده کرده ایم، فقط با مورد قبل یک تفاوت دارد آن هم اینکه، ما فضای خالی برد را در یک آرایه دو بعدی ذخیره کردیم، اما هر بار سه مقدار را می‌گرفتیم، یعنی دو مقدار X از برنامه دریافت می‌کردیم و در کنار آن یک مقدار Y دریافت می‌کردیم، اما در کد اصلی پروژه قطعا این موضوع را خواهید دید که وقتی که ما از داده مختصات فضای خالی استفاده می‌کنیم، داده Y مختصات فضای خالی را به صورت Y و $Y-25$ در نظر گرفتیم.

در واقعیت منطق اصلی پروژه برای اینکه مقادیر را اینگونه در نظر گرفته ایم این بوده است، که هر تکه کوچکی از برد ماز را یک مربع کوچک گرفتیم، تنها دلیل این موضوع آن بوده است، که بتوانیم، حرکت کاربر را در برد ماز کنترل کنیم، تا کاربر نتواند از موانع موجود رد شود، در ادامه این بحث، یعنی زمانیکه قرار است، تعیین کنیم که منطق استفاده از این مختصات برای جلوگیری از عبور کاربر از موانع چیست، خواهد دید که، این تعداد زیاد از مختصات که بدست آمده برای ما چه

کاربردی در طراحی پروژه داشته است.

شکل ۱-۲ این دو شکل داده های فضای خالی و موانع است.

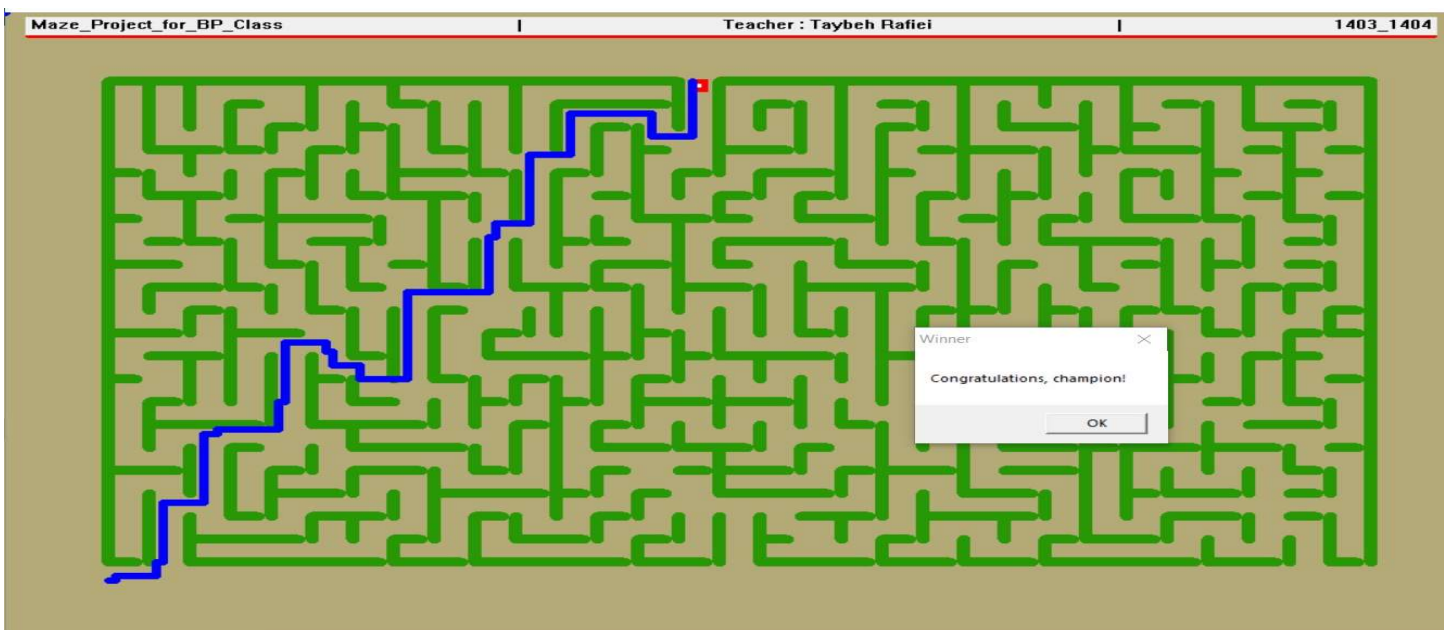
منطق تعیین عملکرد دکمه ها

اگر کد را دیده باشید می فهمید که زمانیکه از دکمه های بالا و پایین استفاده می کنید ، فقط موانع را در نظر می گیرند و ما در این قسمت اینگونه گفته این که در یک حلقه بررسی کند آیا آن شیء به مانع برخورد کرده است یا خیر ، دقت داشته باشید به علت آنکه پروژه گرافیکی انجام شده است ، دقت این عملکرد ها پایین است ، اما به میزان مطلوبی این عملکرد ها قابل اعتماد هستند ، و اما برای دکمه های چپ و راست اینگونه تنظیم کرده ایم که کاربر فقط در فضای خالی برد حرکت کند ، یعنی برای دکمه های چپ و راست اینگونه تنظیم کرده ایم که فقط کاربر در فضای خالی حرکت کند.

اما علاوه بر این چهار دکمه یک دکمه دیگر را هم اضافه کرده ایم که آن دکمه دیگر برای آن است ، که کاربر زمانیکه ، به مانعی برخورد کرد فقط برای یکبار می تواند ، از مانع ببرد ، اما برای بار های بعد این موضوع برایش فراهم نخواهد بود و بار های بعد بدون پرش از مانع باید تا آخر بازی به حرکت خود ادامه دهد تا برنده شود.

منطق اعلام برنده و بازنده

برای اعلام برنده و بازنده می توانیم روش های مختلفی را به کار ببریم ، اما یکی از روش هایی که به کار بردیم این بود که اگر بار اول به مانع برخورد کرد ، می تواند از روی آن بپرد یا از کنار آن رد شود ، اما بار دوم اگر به مانع برخورد کرد و در مانع گیر کرد دیگه نمی تواند از آن بپرد و گاهی حتی ممکن است با توجه به شرایطی که در نظر گرفته شده کاربر بازنده شود. برای اعلام برنده هم ، اینگونه عمل کردیم که اگر ، فرد مسیر را بدون هیچ گونه اشتباهی طی کرد و به مختصات پایانی رسید ، آنگاه فرد برنده اعلام می شود ، و همان موقع که برنده اعلام می شود ، مسیر طی شده توسط کاربر هم به نمایش در می آید.



نتیجه گیری

این پروژه یک پروژه بسیار کاربردی است ، که نه تنها برای تقویت مهارت کودکان مفید است ، بلکه در بسیاری از شاخه های علمی مانند کشیدن مدار های الکترونیکی و .. کاربرد دارد ، همچنین این پروژه به فردی که در حال یادگیری یک زبان برنامه نویسی است هم کمک زیادی می کند و باعث میشود فرد برنامه نویسی منطقی تر به مسائل فکر کند.