پروژه مبانی

بازی Maze

1403/10/30

مرتضی حیدری محمد یاسین خانی

کلاس مبانی برنامه نویسی

فهرست مطالب

مقدمهالف

زبان برنامه نویسیc++ ب

کتابخانه گرافیکی ب

توضیح مختصرپروژه ب

توضیحات پروژه1

**طراحی برد بازی2**

**مختصات موانع برد3**

**مختصات فضای خالی برد4**

**طراحی پنجره5**

**تابع کشیدن شکل6**

**ساختن Label6**

**تنظیم سایز پنجره6**

**تنظیم کشیدن خطوط درون پنجره7**

**تنظیم دکمه های ورودی و عملکرد آنها8**

**تنظیم رنگ پس زمینه پنجره13**

**تنظیم تابع WINAPI13**

منطق پروژه14

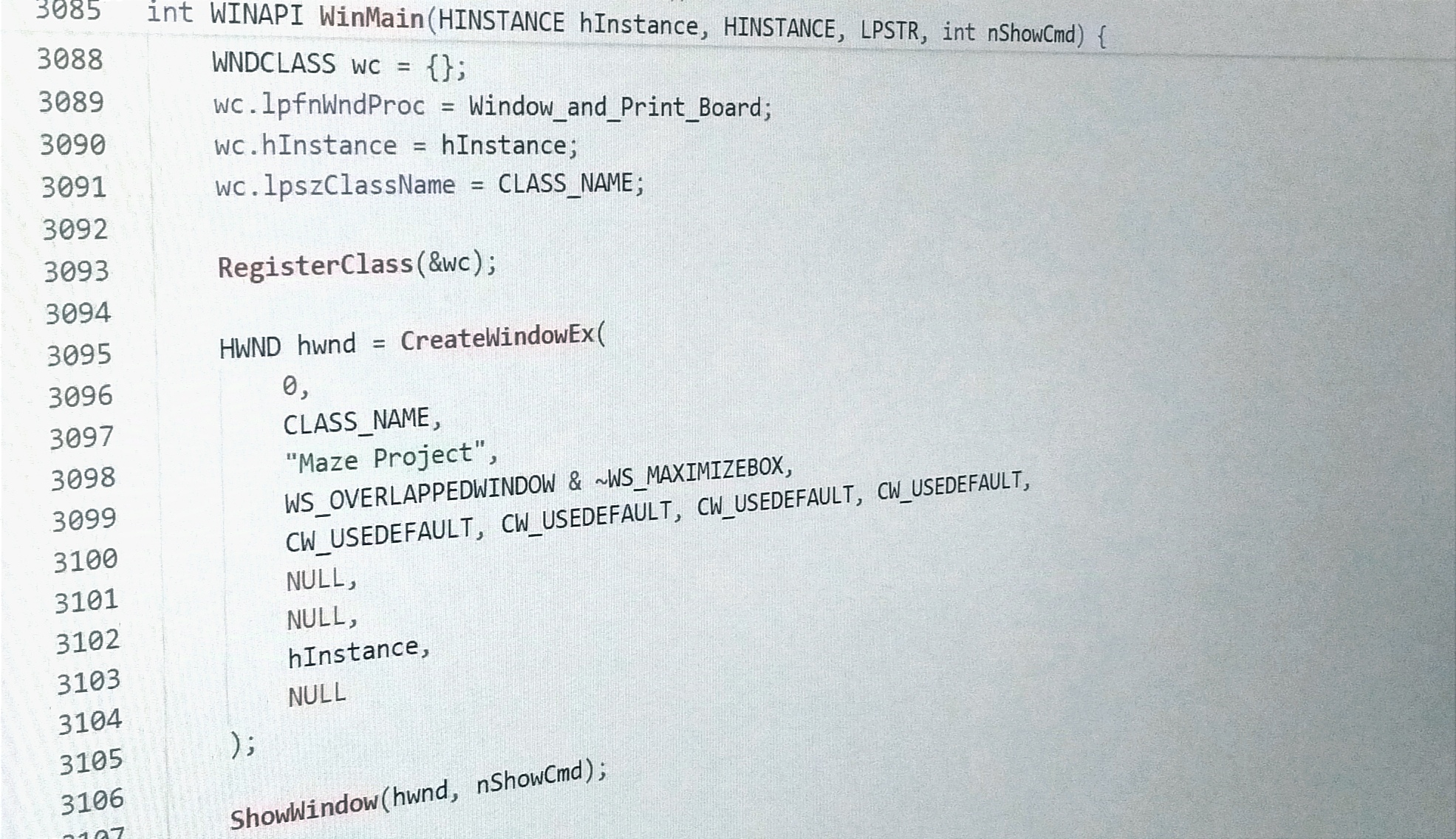
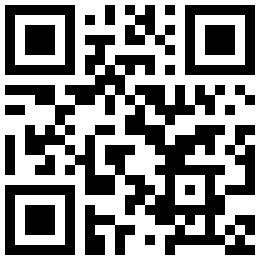
**منطق طراحی برد پروژه15**

**منطق تعیین مختصات موانع و فضای خالی برد15**

**منطق تعیین عملکرد دکمه ها16**

**منطق اعلام برنده و بازنده16**

نتیجه گیری16



مقدمه

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

زبان برنامه نویسیc++

استراس تروپ ، ریاضیدان و دانشمند دانمارکی در آزمایشگاه بل به منظور توسعه زبان C ، زبان C++ را ساخت ؛ که در ابتدا این زبان راC با کلاسC With Classes) ) نامید ، نقطه عطف زبان C++ نسبت به زبان C مواردی مانند [کلاس‌ها](https://fa.wikipedia.org/wiki/%DA%A9%D9%84%D8%A7%D8%B3) ، [توابع مجازی](https://fa.wikipedia.org/wiki/%D8%AA%D9%88%D8%A7%D8%A8%D8%B9_%D9%85%D8%AC%D8%A7%D8%B2%DB%8C)، [سربارگزاری عملگرها](https://fa.wikipedia.org/wiki/%D8%B3%D8%B1%D8%A8%D8%A7%D8%B1%DA%AF%D8%B2%D8%A7%D8%B1%DB%8C_%D8%B9%D9%85%D9%84%DA%AF%D8%B1%D9%87%D8%A7)، [وراثت چندگانه](https://fa.wikipedia.org/wiki/%D9%88%D8%B1%D8%A7%D8%AB%D8%AA_%DA%86%D9%86%D8%AF%DA%AF%D8%A7%D9%86%D9%87)، قالب [توابع](https://fa.wikipedia.org/wiki/%D8%AA%D9%88%D8%A7%D8%A8%D8%B9) و [پردازش استثناء](https://fa.wikipedia.org/w/index.php?title=%D9%BE%D8%B1%D8%AF%D8%A7%D8%B2%D8%B4_%D8%A7%D8%B3%D8%AA%D8%AB%D9%86%D8%A7%D8%A1&action=edit&redlink=1) 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 1983 نا این زبان به C++ تغییر کرد ؛ و این زبان دارای قابلیت‌ های [انواع داده ایستا](https://fa.wikipedia.org/wiki/%D8%B3%DB%8C%D8%B3%D8%AA%D9%85_%D9%86%D9%88%D8%B9)، [نوشتار آزاد](https://fa.wikipedia.org/wiki/%D8%B2%D8%A8%D8%A7%D9%86_%D9%86%D9%88%D8%B4%D8%AA%D8%A7%D8%B1_%D8%A2%D8%B2%D8%A7%D8%AF) و [چندمدلی](https://fa.wikipedia.org/wiki/%D8%B2%D8%A8%D8%A7%D9%86_%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3%DB%8C_%DA%86%D9%86%D8%AF%D9%85%D8%AF%D9%84%DB%8C) می‌باشد و معمولاً [زبان ترجمه‌شده](https://fa.wikipedia.org/wiki/%D8%B2%D8%A8%D8%A7%D9%86_%D8%AA%D8%B1%D8%AC%D9%85%D9%87%E2%80%8C%D8%B4%D8%AF%D9%87) با پشتیبانی از [برنامه‌نویسی ساخت‌یافته](https://fa.wikipedia.org/wiki/%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3%DB%8C_%D8%B3%D8%A7%D8%AE%D8%AA%E2%80%8C%DB%8C%D8%A7%D9%81%D8%AA%D9%87)، [برنامه‌نویسی شیءگرا](https://fa.wikipedia.org/wiki/%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3%DB%8C_%D8%B4%DB%8C%D8%A1%DA%AF%D8%B1%D8%A7) و [برنامه‌نویسی جنریک](https://fa.wikipedia.org/wiki/%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3%DB%8C_%D8%AC%D9%86%D8%B1%DB%8C%DA%A9) است. از آنجا که در ++C اشیاء را می‌توان ابتدا به ساکن از کلاس‌هایی ایجاد کرد که به هیچگونه [سلسله مراتب](https://fa.wikipedia.org/wiki/%D8%B3%D9%84%D8%B3%D9%84%D9%87%E2%80%8C%D9%85%D8%B1%D8%A7%D8%AA%D8%A8) رده‌ها و وراثت مقید نیستند، لذا ++C از [برنامه‌سازی شیء بنیاد](https://fa.wikipedia.org/w/index.php?title=%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D8%B3%D8%A7%D8%B2%DB%8C_%D8%B4%DB%8C%D8%A1_%D8%A8%D9%86%DB%8C%D8%A7%D8%AF&action=edit&redlink=1) نیز پشتیبانی می‌کند.این زبان برنامه‌نویسی در سال [۱۹۹۸](https://fa.wikipedia.org/wiki/%DB%B1%DB%B9%DB%B9%DB%B8_(%D9%85%DB%8C%D9%84%D8%A7%D8%AF%DB%8C)) تحت نام ISO/[IEC](https://fa.wikipedia.org/wiki/%DA%A9%D9%85%DB%8C%D8%B3%DB%8C%D9%88%D9%86_%D8%A8%DB%8C%D9%86%E2%80%8C%D8%A7%D9%84%D9%85%D9%84%D9%84%DB%8C_%D8%A7%D9%84%DA%A9%D8%AA%D8%B1%D9%88%D8%AA%DA%A9%D9%86%DB%8C%DA%A9) ۱۴۸۸۲:۱۹۹۸ استاندارد شد. نسخه فعلی استاندارد این زبان ISO/IEC ۱۴۸۸۲:۲۰۱۴ است.

کتابخانه گرافیکی

این کتابخانه گرافیکی که به برنامه‌نویسان اجازه می‌دهد تا از توابع و امکانات مختلف سیستم‌عامل ویندوز استفاده کنند. این کتابخانه به طور خاص برای توسعه برنامه‌های کاربردی ویندوز طراحی شده است و شامل توابعی برای مدیریت پنجره‌ها، پردازش پیام‌ها، مدیریت فایل‌ها، کار با رشته‌ها و بسیاری از ویژگی‌های دیگر است. مایکروسافت، شرکتی که سیستم‌عامل ویندوز را ایجاد کرده است، مسئول توسعه windows.h و API مربوط به آن است. این کتابخانه به طور مداوم به‌روزرسانی شده تا با نسخه‌های جدید ویندوز و نیازهای برنامه‌نویسان همگام باشد.

این زبان به طور گسترده‌ای در توسعه برنامه‌های گرافیکی و بازی‌ها استفاده می‌شود. برای مثال، در کتابخانه‌هایی مانند DirectX ، توابع این کتابخانه برای مدیریت پنجره‌ها و پردازش ورودی کاربر ضروری هستند، این کتابخانه امکاناتی برای مدیریت فایل‌ها، حافظه، و سایر منابع سیستم فراهم می‌کند که برای هر نوع نرم‌افزاری که بر روی ویندوز اجرا می‌شود، ضروری است.

کتابخانه windows.h یکی از اجزای کلیدی برای برنامه‌نویسی در محیط ویندوز است که توسط مایکروسافت توسعه یافته و به برنامه‌نویسان این امکان را می‌دهد تا از قابلیت‌های گسترده سیستم‌عامل بهره‌برداری کنند. این کتابخانه با توجه به نیازهای روزافزون توسعه‌دهندگان، به طور مداوم در حال بهبود و گسترش است.

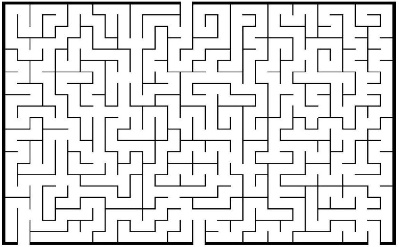
توضیح مختصرپروژه

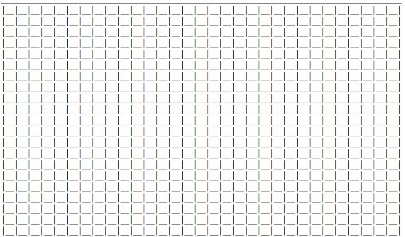
در این پروژه از کتابخانه گرافیکی windows.h استفاده شده ، با استفاده از این کتابخانه گرافیکی ، توابعی تعریف شده و همچنین دستوراتی نوشته شده ؛ در جهت باز شدن یک پنجره و نمایش برد بازی و در آخر منطق بازی بر روی پروژه پیاده شده ، به این نکته توجه داشته باشید که به منظور تقلب نکردن کاربر، برنامه برخی از حرکات که گاربر 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.



توضیحات پروژه

ما از تجربه درس نمی‌گیریم، ما از تامل بر روی تجربه درس می‌گیریم

طراحی برد بازی

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.

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

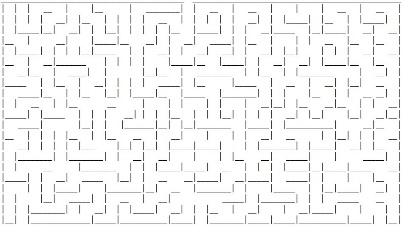
در ادامه قصد بر این است که منطق پیاده سازی برد بازی را به صورت کامل توضیح دهیم، پس در ادامه به کد زیر توجه کنید.

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

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

شکل 2-2 این برد خام بازی Maze است.

#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

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 متوحه خواهید شد که وجود این برد برای کشیدن برد به صورت گرافیکی چقدر حیاتی سات.

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();

}

شکل 4-1 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 نساخته‌اند ؛ بلکه آنها با ساخت مسیرهای تو در تو و پرتاب و خم، ثروت و گنجینه‌هایشان را پنهان می‌کردند.

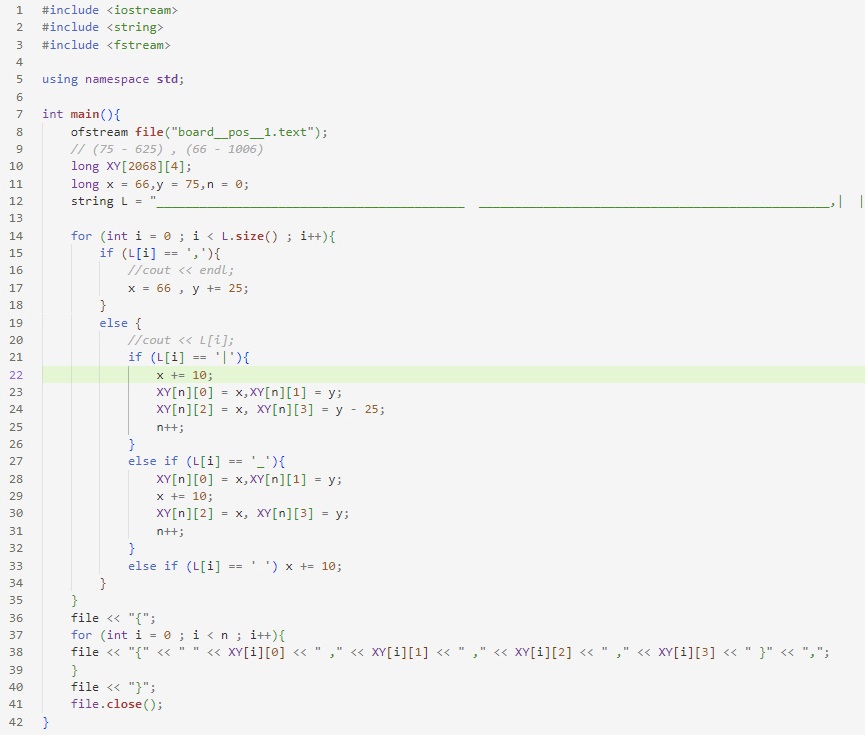
برای بدست آوردن مختصات موانع ، به علت آنکه تعداد موانع زیاد است ، نمی توان مختصات موانع را به صورت دستی تنظیم کرد پس باید به دنبال یک روش مفید وبهینه برای یافتن این مختصات باشیم ، اما روش بهینه چیست؟ چطور می توان المان های زیادی را دریافت کرد و سپس مختصات آنها را خروجی داد؟

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

ofstream file("board\_\_pos\_\_1.text");

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

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

**        if (L[i] == ','){

شکل 5-1 این شکل یک ، نمای کلی از کد مختصات موانع برد است.

*//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;

        } }

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 << "}";

مختصات فضای خالی برد

آیا می دانید که

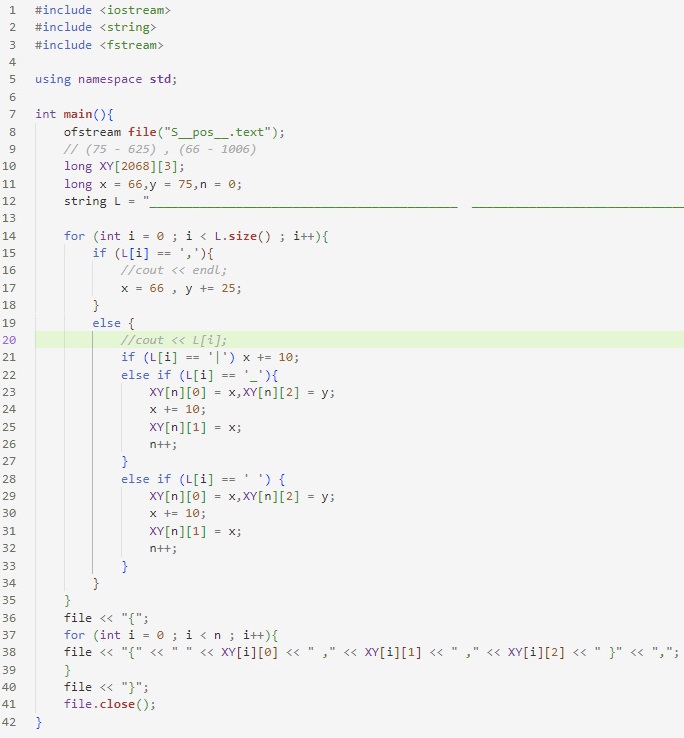
برخی‌ در پارک ها یا مزرعه ‌های خویش مازهایی ساخته‌اند و مردم برای رفتن و گشتن در آن مازها هم پول می‌دهند. البته مازهای هستند که واقعا سخت هستند آن قدر سخت که برخی از درماندگان در آن مازها دست به دامان پلیس و آتش‌نشانی شده‌اند؛ تا آنها را از ماز نجات دهند!

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 وبهینه 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 3 مقدار تعلق می گیرد که برای کشیدن برد کافی است.

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

شکل 6-1 این شکل یک ، نمای کلی از کد مختصات فضای خالی برد است.

        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++;

            }

        }

    }

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 << "}";

    file.close();

طراحی پنجره

برخی از فواید ماز :

1. تقویت و پرورش مهارت

2. تقویت دقت و توجه

3. افزایش تمرکز کودکان

4. تقویت مهارتهای حرکتی

5. یک سرگرمی خوب

6. تقویت مهارت مسیر یابی

7. تقویت جهت یابی

8. ایجاد اعتماد به نفس

9. یک نوع بازی فکری

10. بهبود حواس پرتی

11. افزایش توانمندی‌های ذهنی

12. تقویت مهارت حل مساله

13. تقویت حافظه فعال

14. افزایش سرعت عمل کودک

15. تقویت سرعت پردازش، برنامه‌ریزی، تصمیم‌گیری و اقدام

16 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 معمولا 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 پارامتر های 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 معمولا 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

کاربردهای علمی ماز:

روانشناسان با مطالعه‌ی رفتار حیوانات و بحث یادگیری آنها از آن بهره می‌برند. ( همان قرار دادن موش در ماز و یافتن غذا )همچنین در کتاب چه کسی پنیر مرا جا به جا کرد که نوشته اسپنسر جانسون و کن بلانچارد است همه‌اش موشها را در ماز می‌دوانند! موشها نمادی از شخصیت و درون آدمها هستند؛ یعنی آنها را در مازهای اندرون خویش می‌دوانند. و دانشمندان و ریاضیدانان جهت ترسیم نقشه و رسم مسیر عبور سیمهای تلفن و راههای شهری و مدار الکترونیکی و… از مازها بهره می‌گیرند؛ کدام مسیر بهترین، مطمئن‌ترین، نزدیک‌ترین یا کمترین هزینه را خواهد داشت؟

در حین اجرای بازی ما نیاز داریم که یک شئ در پنجره تعریف شده باشد و آن شئ 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 شئ 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\_O\_x , pos\_O\_y , pos\_O\_x + 10 , pos\_O\_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|1403\_1404 ",

        WS\_VISIBLE | WS\_CHILD, 15, 5, 1040, 20,

        hwnd, nullptr, nullptr, nullptr);

        SetWindowRgn(Label, hRgn, TRUE);

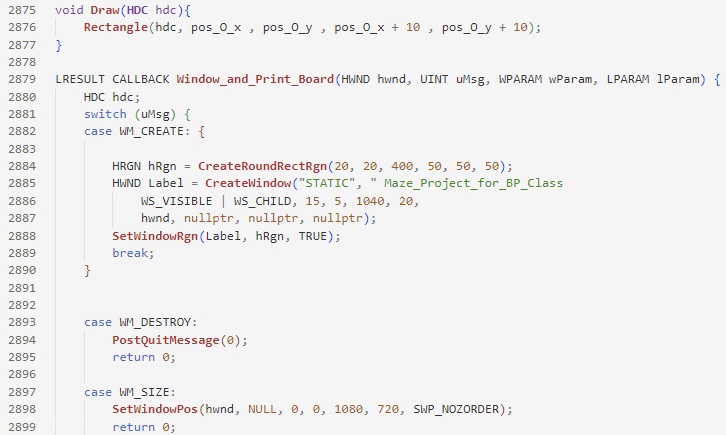
        break;

    }

تنظیم سایز پنجره

برای آنکه کاربر نتواند اندازه پنجره را تغییر دهد ما یک اندازه مشخص برای پنجره تعیین کرده ایم. و زمانیکه اندازه پنجره تغییر کند این پیام ( WM\_SIZE ) به برنامه ارسال می شود.

case WM\_SIZE:

        SetWindowPos(hwnd, NULL, 0, 0, 1080, 720, SWP\_NOZORDER);

        return 0;

شکل 7-1 این شکل یک ، نمای کلی از کد مواری است ، که در بالا توضیح داده شده اند.

تنظیم کشیدن خطوط درون پنجره

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**](https://magazine.nabaat.ir/teaching-problem-solving-to-children/) کودک پیشنهاد می شود حتما این مقاله را مطالعه کنید.

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 هد.

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);

شکل 8-1 این شکل یک ، نمای کلی از کد کشیدن خطوط درون پنجره است.

                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;

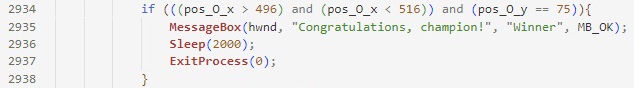
    }

تنظیم دکمه های ورودی و عملکرد آن

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 ازدروس 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 زیای برای پیدا کردن راحل انتخابی میکنند این امر سبب میشود تا کودکان بتوانند به تعادل و کنترل مهارت های حرکتی و چشمی دست یابند.

برای این پروژه 5 دکمه تعریف شده است ؛ که 4 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 > 496) and (pos\_O\_x < 516)) and (pos\_O\_y == 75)){

                MessageBox(hwnd, "Congratulations, champion!", "Winner", MB\_OK);

                Sleep(2000);

                ExitProcess(0);

شکل 9-1 این شکل یک ، نمای کلی از کد بررسی برنده شدن کاربر در بازی است.

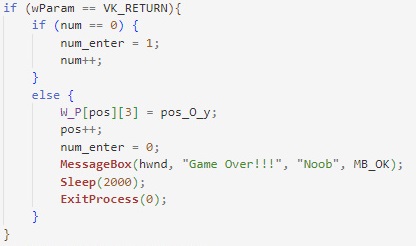
            }

این قسمت برای بررسی وضعیت پرش کاربر تنظیم شده است :

            if (wParam == VK\_RETURN){

                if (num == 0) {

                    num\_enter = 1;

                    num++;

شکل 10-1 این شکل یک ، نمای کلی از کد بررسی پرش کاربر در بازی است.

                }

                else {

                    W\_P[pos][3] = pos\_O\_y;

                    pos++;

                    num\_enter = 0;

                    MessageBox(hwnd, "Game Over!!!", "Noob", MB\_OK);

                    Sleep(2000);

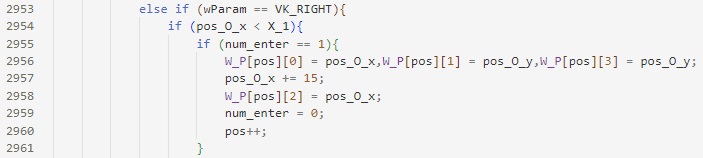
                    ExitProcess(0);

                }

            }

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

            else if (wParam == VK\_RIGHT){

                if (pos\_O\_x < X\_1){

شکل 11-1 این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

                    if (num\_enter == 1){

                        W\_P[pos][0] = pos\_O\_x,W\_P[pos][1] = pos\_O\_y,W\_P[pos][3] = pos\_O\_y;

                        pos\_O\_x += 15;

                        W\_P[pos][2] = pos\_O\_x;

                        num\_enter = 0;

                        pos++;

                    }

                    else {

آیا می دانید که:

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

                        int pos\_\_x = pos\_O\_x;

                        for (int i = 0 ; i < 1691 ; i++){

                            W\_P[pos][0] = pos\_O\_x,W\_P[pos][1] = pos\_O\_y,W\_P[pos][3] = pos\_O\_y;

                            if ((pos\_O\_y >= S[i][2] - 25) and (pos\_O\_y <= S[i][2])){

                                if ((pos\_O\_x >= S[i][0]) and (pos\_O\_x <= S[i][1])){

                                    pos\_O\_x += 5;

                                    W\_P[pos][2] = pos\_O\_x;

                                    pos++;

                                    break;

                                }

                            }

                        }

                        if (pos\_\_x == pos\_O\_x){

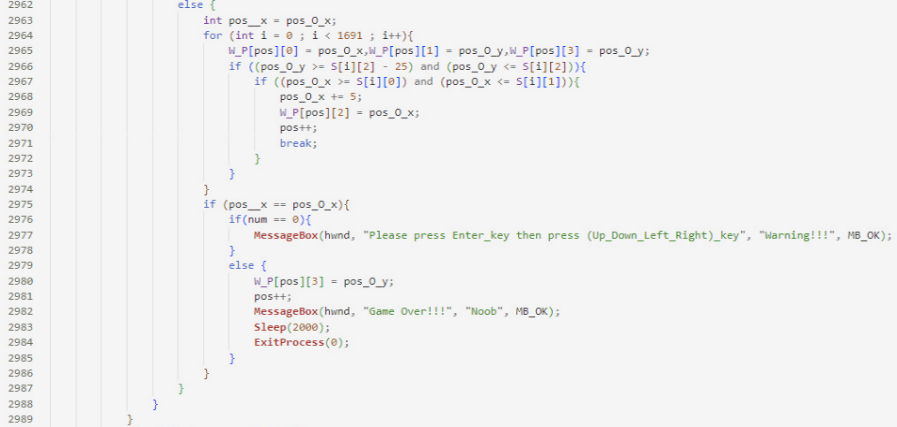
                            if(num == 0){

                                MessageBox(hwnd, "Please press Enter\_key then press (Up\_Down\_Left\_Right)\_key", "Warning!!!", MB\_OK);

                            }

                            else {

                                W\_P[pos][3] = pos\_O\_y;

                                pos++;

                                MessageBox(hwnd, "Game Over!!!", "Noob", MB\_OK);

                                Sleep(2000);

                                ExitProcess(0);

                            }

                        }

                    }

                }

            }

شکل 12- این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

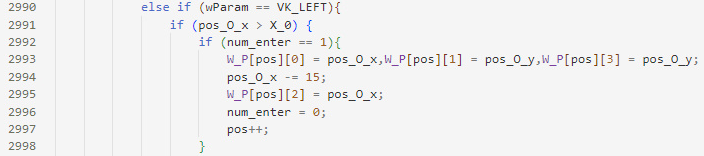
این قسمت برای بررسی وضعیت حرکت کاربر به سمت چپ تنظیم شده است :

            else if (wParam == VK\_LEFT){

                if (pos\_O\_x > X\_0) {

                    if (num\_enter == 1){

                        W\_P[pos][0] = pos\_O\_x,W\_P[pos][1] = pos\_O\_y,W\_P[pos][3] = pos\_O\_y;

                        pos\_O\_x -= 15;

شکل 13-1 این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

                        W\_P[pos][2] = pos\_O\_x;

                        num\_enter = 0;

                        pos++;

                    }

                    else {

                        int pos\_\_x1 = pos\_O\_x;

آیا می دانید که:

پیج 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.

                        for (int i = 0 ; i < 1691 ; i++){

                            W\_P[pos][0] = pos\_O\_x,W\_P[pos][1] = pos\_O\_y,W\_P[pos][3] = pos\_O\_y;

                            if ((pos\_O\_y > S[i][2] - 25) and (pos\_O\_y < S[i][2])){

                                if ((pos\_O\_x >= S[i][0] + 5) and (pos\_O\_x <= S[i][1])){

                                    pos\_O\_x -= 5;

                                    W\_P[pos][2] = pos\_O\_x;

                                    pos++;

                                    break;

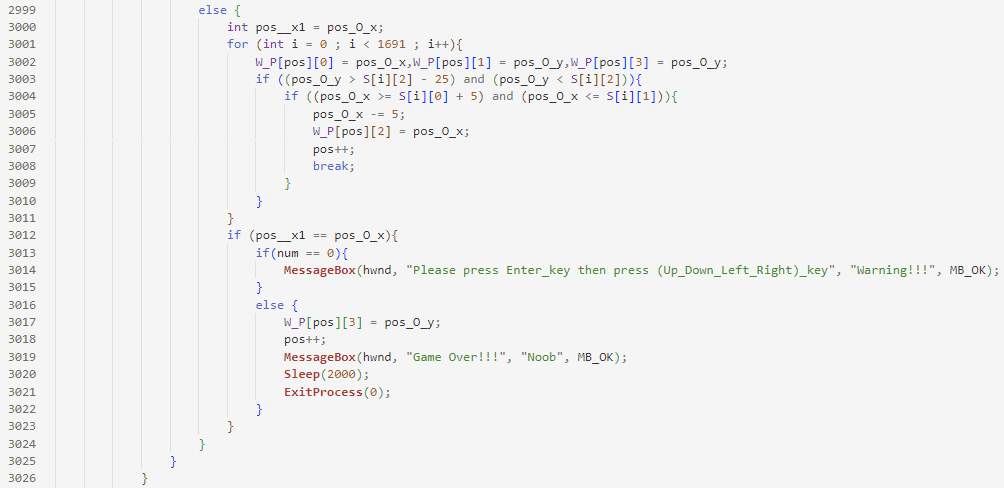
                                }

                            }

                        }

                        if (pos\_\_x1 == pos\_O\_x){

                            if(num == 0){

                                MessageBox(hwnd, "Please press Enter\_key then press (Up\_Down\_Left\_Right)\_key", "Warning!!!", MB\_OK);

                            }

شکل14-1 این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

                            else {

                                W\_P[pos][3] = pos\_O\_y;

                                pos++;

                                MessageBox(hwnd, "Game Over!!!", "Noob", MB\_OK);

                                Sleep(2000);

                                ExitProcess(0);

                            }

                        }

                    }

                }

            }

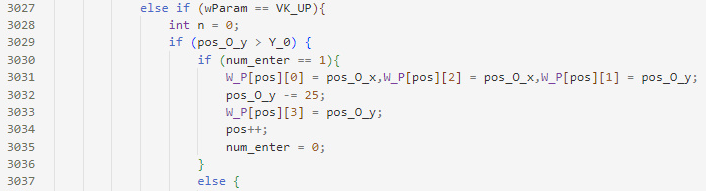
این قسمت برای بررسی وضعیت حرکت کاربر به سمت بالا تنظیم شده است :

            else if (wParam == VK\_UP){

                int n = 0;

                if (pos\_O\_y > Y\_0) {

                    if (num\_enter == 1){

                        W\_P[pos][0] = pos\_O\_x,W\_P[pos][2] = pos\_O\_x,W\_P[pos][1] = pos\_O\_y;

                        pos\_O\_y -= 25;

شکل15-1 این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

                        W\_P[pos][3] = pos\_O\_y;

                        pos++;

                        num\_enter = 0;

                    }

                    else {

                        W\_P[pos][0] = pos\_O\_x,W\_P[pos][2] = pos\_O\_x,W\_P[pos][1] = pos\_O\_y;

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.

                        for (int i = 0 ; i < 1162 ; i++){

                            if ((pos\_O\_x >= XY[i][0]) and (pos\_O\_x <= XY[i][2])){

                                if ((pos\_O\_y >= XY[i][1]) and (pos\_O\_y <= XY[i][3])){

                                    n = n + 5;

                                    break;

                                }

                            }

                        }

                        if (n == 0) {

                            pos\_O\_y -= 5;

                            W\_P[pos][3] = pos\_O\_y;

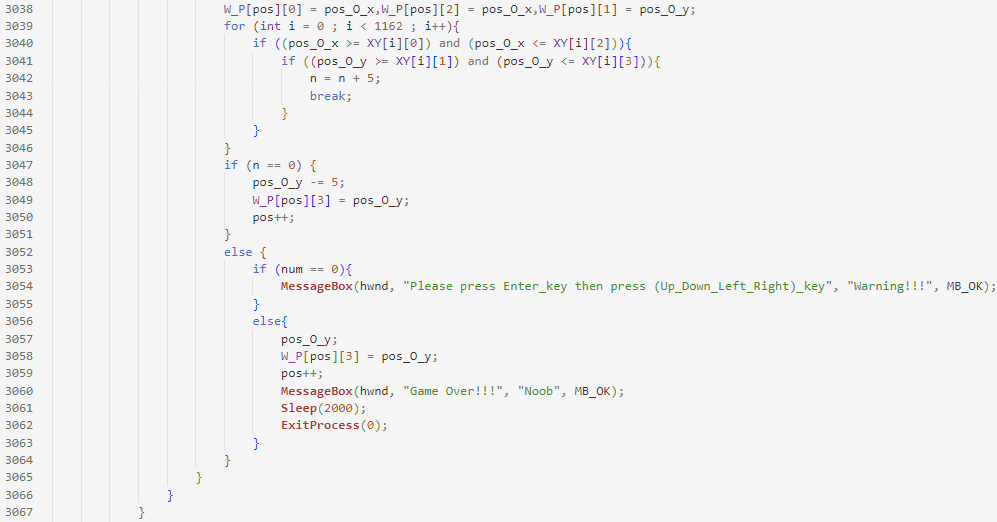
                            pos++;

                        }

                        else {

                            if (num == 0){

                                MessageBox(hwnd, "Please press Enter\_key then press (Up\_Down\_Left\_Right)\_key", "Warning!!!", MB\_OK);

                            }

                            else{

                                pos\_O\_y;

                                W\_P[pos][3] = pos\_O\_y;

                                pos++;

                                MessageBox(hwnd, "Game Over!!!", "Noob", MB\_OK);

                                Sleep(2000);

                                ExitProcess(0);

                            }

                        }

                    }

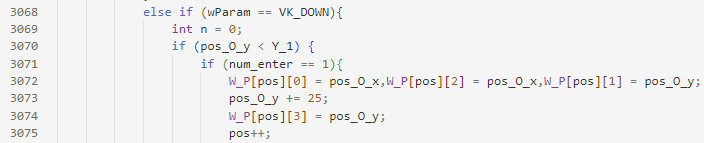
                }

            }

شکل16-1 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 = 0;

                if (pos\_O\_y < Y\_1) {

شکل17-1 این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

                    if (num\_enter == 1){

                        W\_P[pos][0] = pos\_O\_x,W\_P[pos][2] = pos\_O\_x,W\_P[pos][1] = pos\_O\_y;

                        pos\_O\_y += 25;

                        W\_P[pos][3] = pos\_O\_y;

                        pos++;

                        num\_enter = 0;

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.**

**تاثیر روانی: حل معماهای ماز می‌تواند تأثیر قوی بر روی فعالیت مغزی داشته باشد. این نوع بازی‌ها ممکن است به بهبود** [**حافظه**](https://bazi.wiki/index.php?title=%D8%AD%D8%A7%D9%81%D8%B8%D9%87&action=edit&redlink=1)**، تمرکز و افزایش توانایی‌های شناختی کمک کنند.**

                    }

                    else {

                        W\_P[pos][0] = pos\_O\_x,W\_P[pos][2] = pos\_O\_x,W\_P[pos][1] = pos\_O\_y;

                        for (int i = 0 ; i < 1161 ; i++){

                            if ((pos\_O\_x >= XY[i][0]) and (pos\_O\_x <= XY[i][2])){

                                if ((pos\_O\_y >= XY[i][1]) and (pos\_O\_y <= XY[i][3])){

                                    n = n + 1;

                                    break;

                                }

                            }

                        }

                        if (n == 0) {

                            pos\_O\_y += 5;

                            W\_P[pos][3] = pos\_O\_y;

                            pos++;

                        }

                        else {

                            if (num == 0){

                                MessageBox(hwnd, "Please press Enter\_key then press (Up\_Down\_Left\_Right)\_key", "Warning!!!", MB\_OK);

                            }

                            else{

شکل18-1 این شکل یک ، نمای کلی از کد بررسی حرکت کاربر در بازی است.

                                pos\_O\_y;

                                W\_P[pos][3] = pos\_O\_y;

                                pos++;

                                MessageBox(hwnd, "Game Over!!!", "Noob", MB\_OK);

                                Sleep(2000);

                                ExitProcess(0);

                            }

                        }

                    }

                }

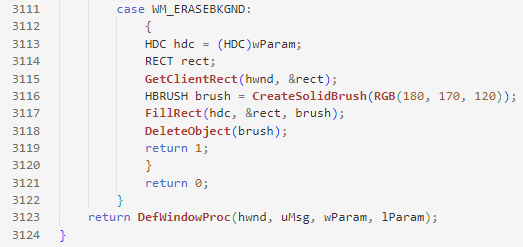
            }

            InvalidateRect(hwnd, NULL, TRUE);

            break;

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\_ERASEBKGND:

            {

شکل18-1 این شکل یک ، نمای کلی از کد تنظیم رنگ بازی است.

            HDC hdc = (HDC)wParam;

            RECT rect;

            GetClientRect(hwnd, &rect);

            HBRUSH brush = CreateSolidBrush(RGB(180, 170, 120));

            FillRect(hdc, &rect, brush);

            DeleteObject(brush);

            return 1;

            }

            return 0;

        }

تنظیم تابع WINAPI

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

    const char CLASS\_NAME[] = "Sample Window Class";

شکل19-1 این شکل یک ، نمای کلی از کد تابع winapi در بازی است.

    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);



منطق پروژه

\

وقتی دلیل به اندازه کافی بزرگ باشد، شما مایل خواهید بود که تقریبا 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 هستم ، ایده جدیدی برای طراحی برد پروژه بازی ما به نظرم 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 کردیم کاراکتر ها را ، هر بار مقداری به اندازه 10 به X ها اضافه می شود ، هر بار هم که رفتیم خط بعد ، 25 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 دقیقا 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 ببنید.

سپس مقادیر درون فایل را کپی می کنیم و درون فایل main پروژه خود پیاده سازی می کنیم ، در آخر هم از آن در کد اصلی پروژه استفاده می کنیم ، اما نکته قابل توجه در این موضوع آن است گه ، برای پیدا کرد ، 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 می گرفتیم ، یعنی دو مقدار X از برنامه دریافت می کردیم و در کنار آن یک مقدار Y دریافت می کردیم ، اما در کد اصلی پروژه قطعا این موضوع را خواهید دید که وقتی که ما از داده مختصات فضای خالی استفاده می کنیم ، 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.



شکل1-2 این دو شکل داده های فضای خالی و موانع است.

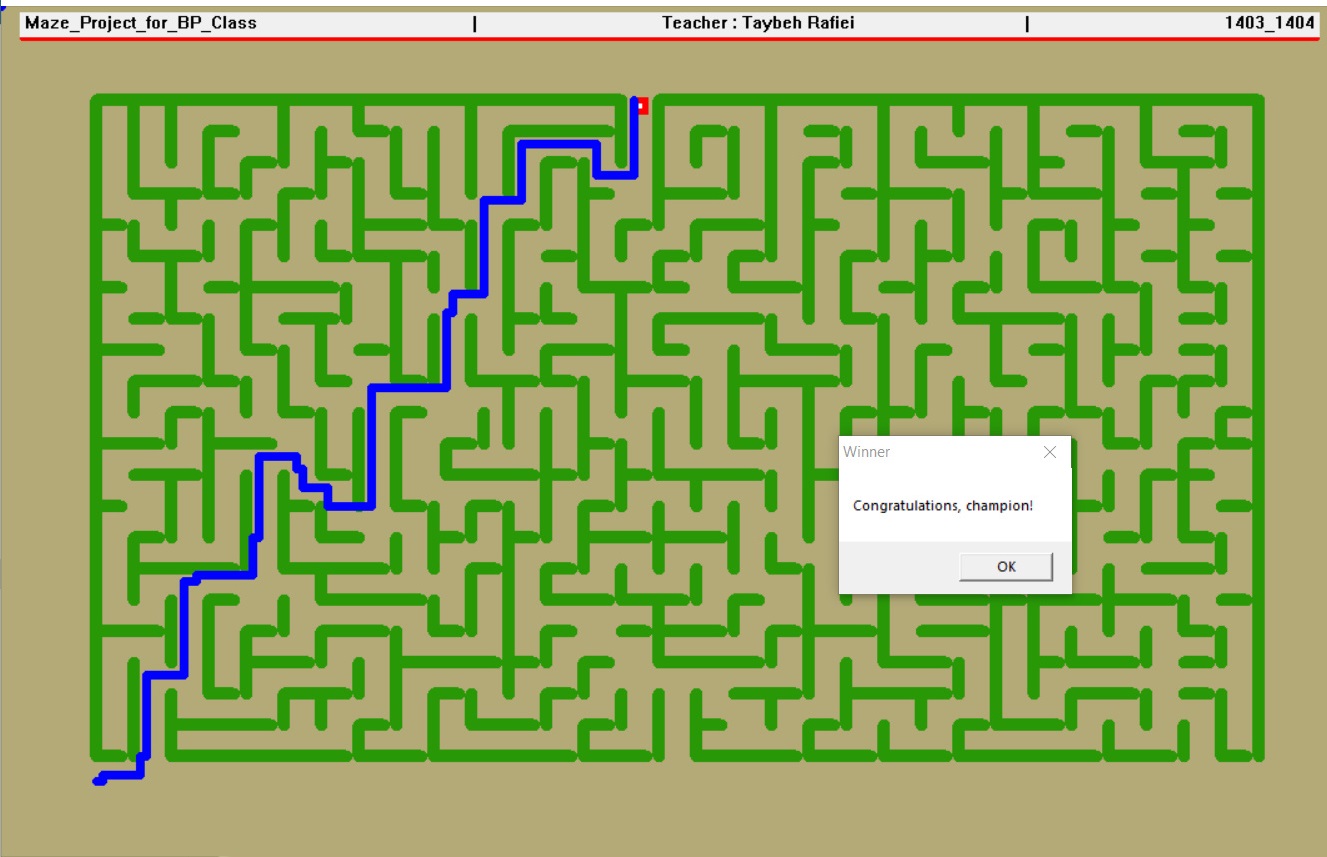
منطق تعیین عملکرد دکمه ها

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 شئ به مانع بر خورد 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 بپرد ، 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 رد 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 نویس منطقی تر به مسائل فکر کند.