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 برد5**

**مختصات فضای خالی برد5**

**طراحی پنجره5**

**تابع کشیدن شکل6**

**ساختن Label6**

**تنظیم سایز پنجره6**

**تنظیم کشیدن خطوط درون پنجره6**

**تنظیم دکمه های ورودی و عملکرد آنها6**

**تنظیم رنگ پس زمینه پنجره6**

**برگرداندن مقدار تابع6**

**تنظیم تابع WINAPI6**

منطق پروژه5

**منطق طراحی برد پروژه6**

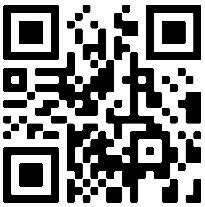
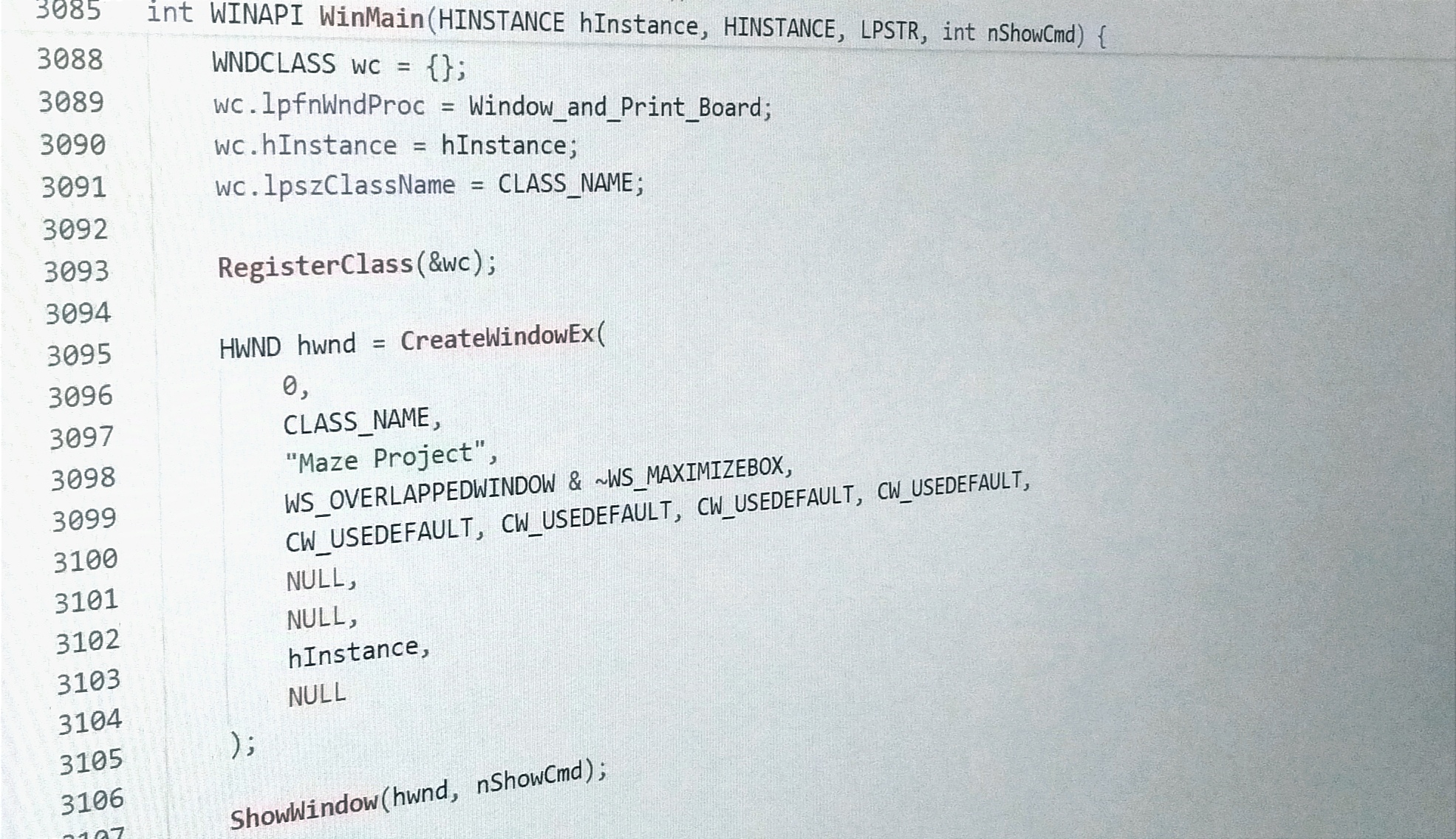
**منطق تعیین مختصات موانع برد6**

**منطق تعیین مختصات فضای خالی برد6**

**منطق تعیین عملکرد دکمه ها6**

**منطق اعلام برنده و بازنده6**

نتیجه گیری4



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

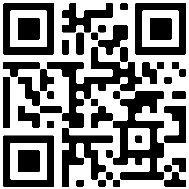
این کتابخانه گرافیکی که به برنامه‌نویسان اجازه می‌دهد تا از توابع و امکانات مختلف سیستم‌عامل ویندوز استفاده کنند. این کتابخانه به طور خاص برای توسعه برنامه‌های کاربردی ویندوز طراحی شده است و شامل توابعی برای مدیریت پنجره‌ها، پردازش پیام‌ها، مدیریت فایل‌ها، کار با رشته‌ها و بسیاری از ویژگی‌های دیگر است. 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.

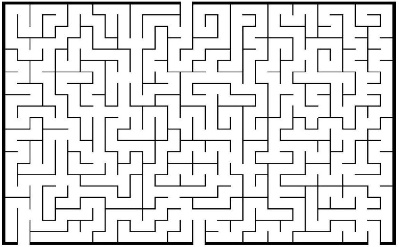
توضیح مختصرپروژه

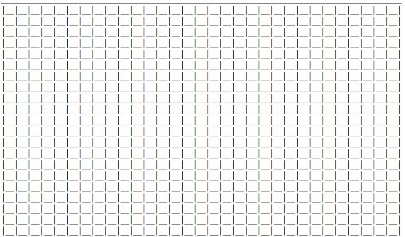
در این پروژه از کتابخانه گرافیکی windows.h استفاده شده ، با استفاده از این کتابخانه گرافیکی ، توابعی تعریف شده و همچنین دستوراتی نوشته شده ؛ در جهت باز شدن یک پنجره و نمایش برد بازی و در آخر منطق بازی بر روی پروژه پیاده شده ، به این نکته توجه داشته باشید که به منظور تقلب نکردن کاربر، برنامه برخی از حرکات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

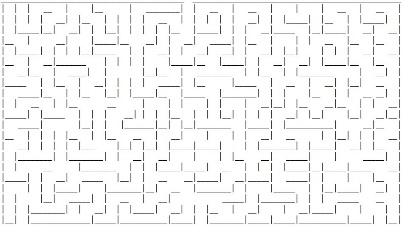
در ادامه قصد بر این است که منطق پیاده سازی برد بازی را به صورت کامل توضیح دهیم، پس در ادامه به کد زیر توجه کنید.

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.

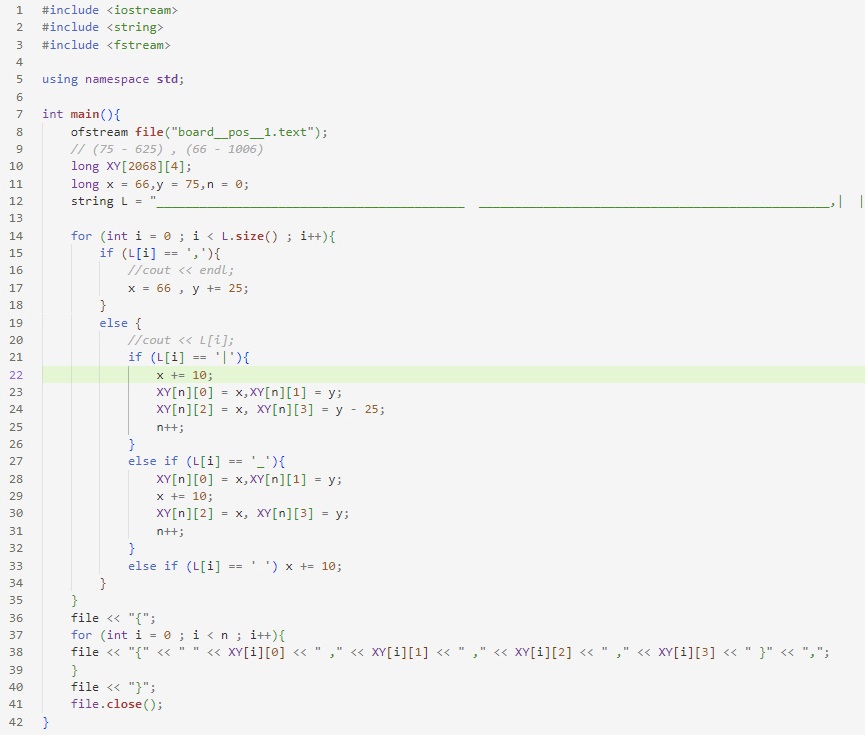
برای بدست آوردن مختصات موانع ، به علت آنکه تعداد موانع زیاد است ، نمی توان مختصات موانع را به صورت دستی تنظیم کرد پس باید به دنبال یک روش مفید وبهینه برای یافتن این مختصات باشیم ، اما روش بهینه چیست؟ چطور می توان المان های زیادی را دریافت کرد و سپس مختصات آنها را خروجی داد؟

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 ذخیر کنیم :

    file << "{";

    for (int i = 0 ; i < n ; i++){

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

    }

    file << "}";

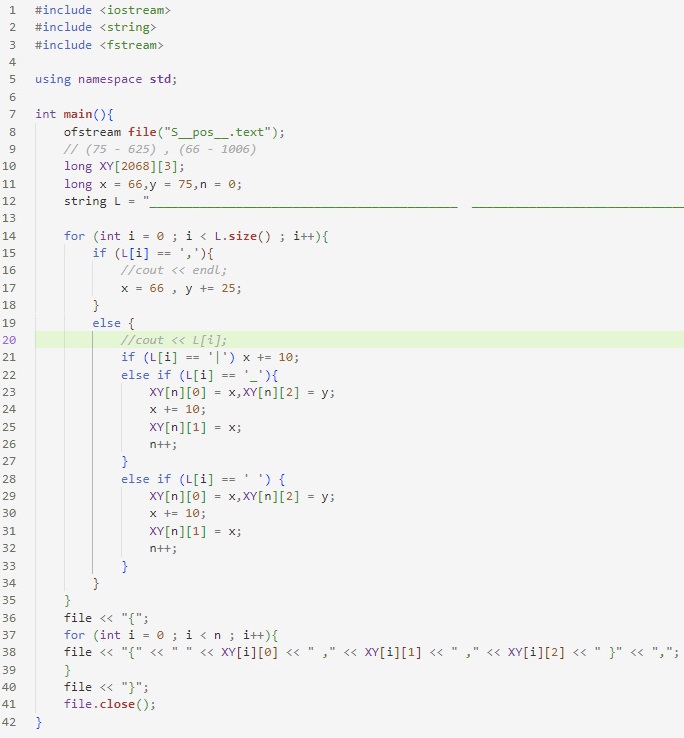
مختصات فضای خالی برد

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

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

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;

تنظیم کشیدن خطوط درون پنجره

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;

    }

تنظیم دکمه های ورودی و عملکرد آن

case WM\_KEYDOWN:

            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);

            }

            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);

                }

            }

            else if (wParam == VK\_RIGHT){

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

                    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 {

                        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(n == 0){

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

                                pos++;

                                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);

                            }

                        }

                    }

                }

            }

            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;

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

                        num\_enter = 0;

                        pos++;

                    }

                    else {

                        int pos\_\_x1 = 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] + 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(n == 0){

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

                                pos++;

                                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);

                            }

                        }

                    }

                }

            }

            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;

                        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;

                        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 (n == 0){

                                pos\_O\_y;

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

                                pos++;

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

                                Sleep(2000);

                                ExitProcess(0);

                            }

                            else{

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

                            }

                        }

                    }

                }

            }

            else if (wParam == VK\_DOWN){

                int n = 0;

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

                    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;

                    }

                    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 (n == 0){

                                pos\_O\_y;

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

                                pos++;

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

                                Sleep(2000);

                                ExitProcess(0);

                            }

                            else{

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

                                pos++;

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

                            }

                        }

                    }

                }

            }

            InvalidateRect(hwnd, NULL, TRUE);

            break;

تنظیم رنگ پس زمینه

case WM\_ERASEBKGND:

            {

            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;

        }

برگرداندن مقدار تابع

    return DefWindowProc(hwnd, uMsg, wParam, lParam);

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);

    }

    return 0;

}