

بسمه تعالی



دانشگاه صنعتی شریف

گزارش پروژه پایانی درس مبانی برنامه سازی (۲۵۷۶۸)

استاد درس: دکتر آراسته

دستیارهای آموزشی بترتیب الفبا:

امیرحسین رستمی

محمدامین علم الهدی

امیرمهدی سلیمانی فر (دانشجوی ترم اول مهندسی برق)

شماره دانشجویی: ۹۸۱۰۱۷۴۷

نیمسال اول ۹۹-۱۳۹۸

## فهرست مطالب

۳	.....مقدمه
۴	.....بدنه اصلی پروژه و فایل تعریف تابع های ابتدایی
۵	.....انتقال داینامیک بین صفحات
۶	.....توابع و کارکرد pages.h
۷	.....صفحه بازی connect four

## مقدمه

پروژه پایانی درس مبانی برنامه نویسی طراحی بازی connect four که یک بازی فکری دو نفره مشابه دوز است که در آن هدف هریک از بازیکنان قرار دادن چهار مهره متوالی خود در یک سطر، ستون یا قطر است. قسمت های مختلف این پروژه که در بازه زمانی حدودا یک هفته ای طراحی شده اند در بخش های مرتبط توضیح داده خواهند شد و همچنین توجه به نکات مهندسی نرم افزار و نوشتن کد قابل استفاده مجدد در نوشتن این پروژه مد نظر دانشجو بوده اند که در هر بخش که لازم باشد به آن اشاره خواهد شد.

سعی اصلی در طراحی پروژه یک محیط کاربر پسند با رابط کاربری قابل استفاده و زیبا و همچنین راحتی استفاده در ویژگی ها و بخش های مختلف بازی بوده که باعث شوند این پروژه هرچه بیشتر به یک پروژه در دنیای واقعی شباهت پیدا کند و لذا سعی شده است تا حد امکان از بکارگیری کنسول برنامه برای گرفتن ورودی و نمایش خروجی استفاده نشود که البته در یک مورد (دریافت نام کاربر) ناگزیر به استفاده از این مورد شدم.

## بدنه اصلی پروژه و فایل تعریف تابع های ابتدایی

سورس کد پروژه متشکل از یک فایل با پسوند `.cpp` است که در حقیقت همان برنامه اصلی است که وظیفه آن نگه داشتن کاربر در برنامه و انتقال داینامیک بین صفحات برنامه، بازی و دیگر بخش ها تا زمان خروج است.

همچنین کد مربوط به پخش آهنگ پس زمینه در هنگام اجرای برنامه و دریافت نام دو نام کاربری از دیگر وظایف این بخش است.

سه فایل دیگر شامل طراحی صفحات، خود بازی و ساز و کار آن و دیگر عناصر گرافیکی است.

فایل `graphical.h` بخش تعریف کردن تابع های مربوط به المان های اصلی هر یک از صفحات است که شامل بخش های زیر می شود:

- تابع `background(int clr[])` در اصل مسئول ایجاد پس زمینه برای هر یک از صفحات برنامه است که رنگ های آن از یک آرایه ۹۰ عنصره متشکل از مقادیر `RGB` برای رنگ های هارمونیک انتخاب میشوند. در حقیقت این رنگ ها بدلیل کنتراست و پویایی خود ظاهری زیبا به برنامه میدهند. این تابع همچنین آرایه `RGB` رنگ های انتخاب شده را در `clr[]` میریزد تا در ادامه بتواند توسط دیگر عناصر صفحه مورد نظر استفاده شود.
- تابع `icon(...)` مختصات آیکن و رنگ آن و همچنین متن نوشته ای که باید روی آن چاپ شود و رنگ این نوشته را دریافت کرده و در رابط گرافیکی یک آیکن مستطیل شکل با مشخصات مورد نظر را ایجاد میکند، بیشترین کاربرد آن در تهیه منوهای مختلف برنامه و جابجایی بین صفحات است.
- تابع `text(...)` نیز فونت، رنگ، و نوشته مورد نظر و همچنین رنگ پس زمینه این نوشته و مختصات مورد نظر را دریافت و متنی را در صفحه در جایگاه مورد نظر و با ویژگی های خواسته شده قرار میدهد.
- تابع `colorp(...)` مختصات مرکز، شعاع و رنگ یک دایره را دریافت کرده و آنرا در نقطه مورد نظر میکشد که در طراحی مهره های بازیکنان در صفحه بازی، انتخاب رنگ در منوی انتخاب ویژگی های هر بازیکن و دیگر موارد کاربرد دارد.

دقت شود که بدلیل اینکه خروجی تمام موارد زیر در صفحه گرافیکی `winbgi` نمایش داده میشود لذا مقدار برگشتی ندارند و همگی از نوع `void` هستند و تنها تغییر در متغیرها در تابع طراحی صفحه پس زمینه است که مقادیر آرایه آرگومان عوض میشود تا امکان اطلاع از رنگ های بکار گرفته شده در زمینه صفحه ها را برای ایجاد هارمونی بیشتر با دیگر عناصر به ساختار های دیگر بکار رفته در صفحه بدهد.

## انتقال داینامیک بین صفحات

صفحات برنامه در حقیقت از یک ساختار مشابه پیروی میکنند که بصورت زیر دنبال می شود:

۱. ایجاد پس زمینه جدید و از بین بردن محتوای پیشین صفحه
۲. ایجاد آیکن ها و نوشته های نشان دهنده محتوای صفحه و دیگر محتوا
۳. ایجاد یک حلقه بوسیله Kbhit() و یک حلقه دیگر بوسیله Ismouseclick() در داخل آن و اجرای دستورات مربوط به آن صفحه با ایجاد شرط هایی در خطوط تغییرات وضعیت فعلی صفحه کلید و ماوس
۴. برگرداندن مقدار مناسب به تابع while() موجود در تابع main() و ادامه اجرای برنامه بوسیله مقدار بازگشتی و فراخوانی صفحه مناسب

مقادیر برگردانده شده برای انتقال صفحه ها عبارتند از:

- ۱ منوی اصلی
- ۲ تعیین سطح بازی
- ۳ تنظیمات
- ۴ بیشترین امتیازات
- ۵ محیط بازی
- ۰ خروج از برنامه

## توابع و کارکرد pages.h

این فایل هدر مسئول پیاده سازی همه صفحه های برنامه به غیر از صفحه اصلی بازی یا حالت ۵ است.

تابع ۳۰ **color** برای ذخیره سازی RGB رنگ های قابل انتخاب توسط کاربر در برنامه استفاده شده است و بقیه متغیرها و ثابت ها نیز کارکردی متناسب با **identifier** مشخص شده برای آنها دارند.

در برنامه سعی شده است تا در حد امکان از متغیرها بجای ثوابت استفاده شود تا در صورت نیاز و یا شخصی سازی تغییر برنامه موجود به ساده ترین نحو قابل انجام باشد.

کارکرد دکمه ها در برنامه نیز به این صورت است که یک آیکن در ابتدا توسط تابع ابتدایی تعریف و سپس در حلقه **ismouseclick()** صفحه در صورتی که مختصات نقطه کلیک شده در محدوده مورد نظر بود دستورات مورد انتظار را اجرا کند.

صفحات **mainmenu()** و **difficulty()** از همین روند ساده پیروی می کنند و تنها نکته قابل توجه استفاده از آیکن دایره ای شکل برای تعیین تعداد سطرها و ستون های برنامه است که در واقع برای مشخص سازی محدوده از فاصله از مرکز دایره استفاده شده است و همچنین برای آنکه بازی در سطح مناسب و معقولی قابل استفاده باشد محدود سازی مقادیر موجود برای سطرها و ستون ها نیز با بکارگیری % انجام شده است.

صفحه **options()** نیز از یک متد بسیار ساده اما کاربر پسند برای دریافت رنگ کاربر و بازیکن شروع کننده بازی استفاده میکند که باعث راحتی انجام تغییرات میشود. در این بخش به این نکته که دو کاربر نمیتوانند رنگ یکسانی داشته باشند نیز توجه شده و با یک پیغام خطا به کاربر اعلام میشود. تغییرات اعمال شده نیز روی تغییر تعدادی متغیر **extern** که در ابتدایی ترین هدر تعریف شده اند صورت می گیرد و برای استفاده نیز از همین استفاده می شود.

صفحه **highscore()** در واقع یک فراخوانی فایل و پرینت نتایج بصورت گرافیکی است. این فایل قبل از آن توسط خود این برنامه ایجاد و مقادیر آن نیز پس از بازی تغییر داده می شود. این فایل فقط برای خواندن اطلاعات در این بخش فراخوانی می شود.

از نکات دیگر قابل اشاره بکار رفته استفاده از **exitkey** برای دریافت کلید فشرده شده از صفحه کلید و **xmouse** و **ymouse** برای دریافت مختصات نقطه ای که روی آن کلیک شده است می باشد.

## صفحه بازی connect four

متغیرهایی که برای نگه داشتن وضعیت بازی تعریف شده اند شامل یک آرایه دو بعدی **gameboard** و یک آرایه ۴۲ عنصره **moves** است که در واقع توپ انداخته شده به هر ستون را در حرکت های پشت سرهم نگه میدارد که برای برگشت دادن حرکت ها کاربرد دارد. دقت کنید که مقادیر هر یک از این آرایه ها در ابتدای هر بار ورود به یک بازی جدید به مقادیر اولیه خود باز می گردند تا باعث ایجاد مشکل و تداخل نشود و روند بصورت معمول انجام شود.

در ابتدا تعدادی تابع و متغیر کلی برای این فایل هدر تعریف شده اند که پیاده سازی بازی اصلی را ساده تر کنند. تابع **(settings)** برای بازکردن منوی حین بازی است که ساختاری همانند ساختار مورد استفاده برای جابجایی بین صفحات دارد و صفحه بعدی آن توسط مقدار بازگشت داده شده توسط آن تعیین می شود. گزینه های موجود در آن **resume** و **restart** و **cheat** و **exit** میباشند که هر کدام کارکردی مطابق تعریف شده دارند. تنها نکته قابل اشاره آن است که برای انجام تقلب لازم است عبارت **helpme** در مدت زمان ۳ ثانیه توسط کاربر نوشته شود و در غیر اینصورت تاثیری نخواهد داشت. در صورت انجام این کار توسط کاربر در حرکت بعدی این بازیکن برنده بازی شناخته میشود.

تابع **(movement)** مسئولیت انداختن توپ به داخل ستونی که بوسیله آرگومان برای آن ارسال میشود را دارد، بقیه اطلاعات مورد نیاز از متغیرهای **global** تعریف شده دریافت خواهند شد. افتادن بصورتی که از زیر ستون ها عبور کند و همچنین این موضوع که پس از برخورد به توپ پایینی به بالا بازگردد نیز توسط همین تابع پیاده سازی شده اند.

تابع **(endgame)** برای بررسی وضعیت بازی که آیا کسی برنده شده است یا خیر نوشته شده است یا نه پیاده سازی شده و برای اینکار هر یک از ستون ها، سطرها، و قطرها را مورد بررسی قرار میدهد تا از این امر مطمئن شده و برنده را بوسیله متغیرهایی که در همین فایل تعریف شده اند به سایر قسمت های برنامه اطلاع میدهد.

در تابع اصلی نیز یک تغییر فایل نیز انجام میشود که مربوط به جدول ذخیره برترین امتیازات است که معیار آن حداقل حرکت های انجام شده برای برد یکی از طرفین بوده است.

توابع easy و medium و hard نیز بصورت جداگانه برای هوش مصنوعی برنامه مورد نظر نوشته شده اند که این موضوع که کدامیک برای انجام حرکت انتخاب میشوند در منویی که قبل از ورود به بازی می آید تعیین میشود، این منو تنها در صورت انتخاب گزینه بازی با کامپیوتر نمایش داده میشود و در هنگام انتخاب بازی دو نفره تاثیری ندارد. اما کدهای بکار رفته در هر یک از این موارد یکسان بوده اند و من از نمره امتیازی این قسمت بدلیل عدم تناسب و میان نمره اضافی آن و نیاز به کار روی آن در تناسب با دیگر قسمت ها صرف نظر کرده ام اما تنها چیزی که نوشته نشده است الگوریتم انتخاب منطقی در سطح مناسب است، به غیر از آن تمام زیر ساخت ها از جمله قسمت ایجاد سه سطح و انتخاب سطح توسط کاربر تعبیه و کدهای آنها به بهترین نحو ممکن نوشته شده است.

مکانیزم دیگر مورد استفاده اخطار "ستون پر" است که در صورت انتخاب ستونی توسط کاربر که قبلا پر شده است به نمایش در می آید، اما این اخطار هیچگاه برای حرکت کامپیوتر نمایش داده نمیشود زیرا در این صورت handle آن توسط برنامه انجام میشود.

نمایش زمان نیز با کمک کتابخانه <time.h> و معرفی چند متغیر انجام میشود نکته مهم در این بخش عدم تغییر زمان هنگام استفاده از منوی تنظیمات، تغییر نوبت در صورت توقف بیشتر از ده ثانیه و همچنین نمایش زمان طول کشیدن بازی در آخر برنامه همراه با گزینه های تکمیلی مشابه بازی شناخته شده است.

از آنجا که نحوه پیاده سازی قابل مشاهده در کد بوده و همچنین ایده های اصلی از جلسات تدریس استاد برداشت شده تنها به اشاره به نحوه پیاده سازی توابع نسبتا جدید و دادن دید کلی به خواننده اکتفا کرده ام. به نظر شخصی خود متغیر ها و توابع دارای اسامی مشروح و کدهای به نسبت قابل درکی هستند که برای بسیاری از کاربردها صرفا در بخش های مختلف تکرار شده اند.

دوباره بر این نکته تاکید میکنم که هدف اصلی من ایجاد یک بازی کاربر پسند و قابل درک برای فردی که با این برنامه بعنوان یک بازی و نه پروژه ترم دانشجو روبرو میشود بوده و سعی کرده ام رضایت کلی کاربر پس از تجربه بازی را با ایجاد المان های مختلف و جذاب و نه استفاده از یک صفحه کنسول محقق سازم.

از میان بخش های اجباری تمامی قسمت ها نوشته شده اند و از میان قسمت های اختیاری نیز تمامی قسمت ها بجز بخش (ایجاد هوش مصنوعی قابل قبول) را در برنامه خود آورده ام که دلیل ننوشتن کد مربوط به قسمت ذکر شده را نیز در بالا توضیح داده ام. موسیقی پشت صحنه بدلیل حجم بالا در کد آپلود شده در کوئرا قرار داده نشده که امیدوارم اجازه استفاده از آن در ارائه نهایی داده شود.