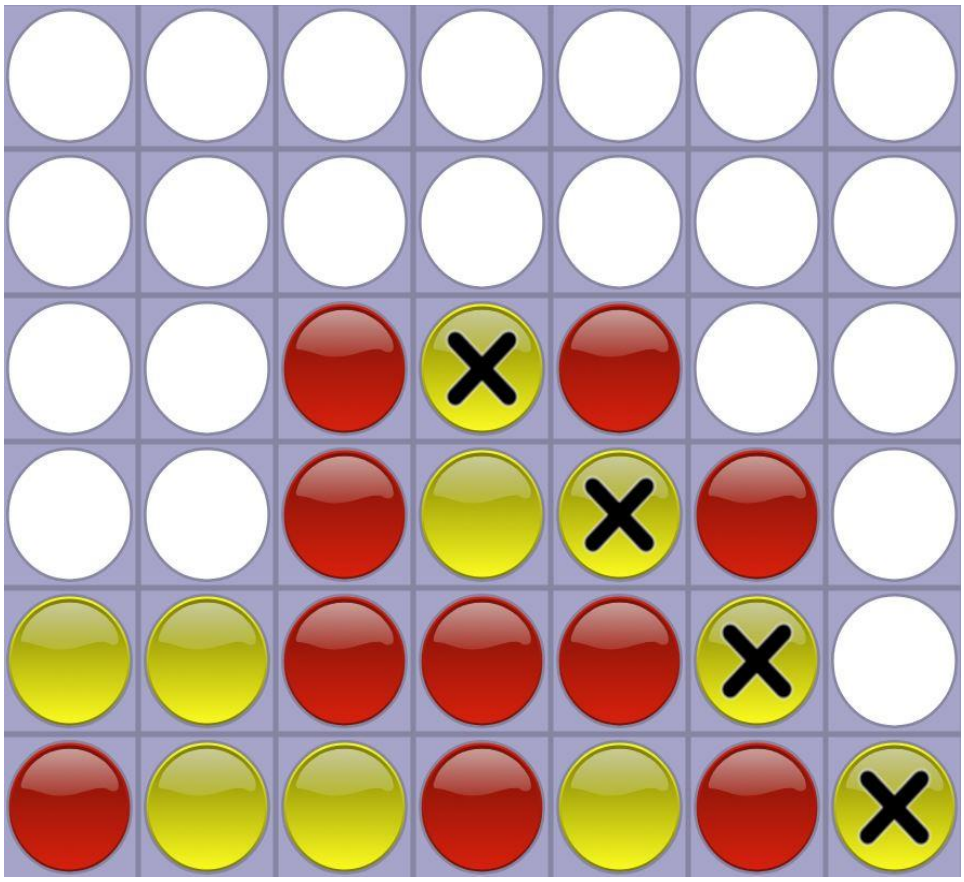
به نام حق

پروژه درس مبانی کامپیوتر و برنامه‌سازی

استاد درس: دکتر سعید ابریشمی

پیاده‌سازی بازی connect four



در این پروژه شما باید نسخه‌ای از بازی معروف connect four را در محیط کنسول پیاده سازی کنید.

قوانین بازی

این بازی دونفره بر روی یک صفحه‌ی ایستا و با همراه داشتن ۴۲ مهره انجام می‌شود. هر بازیکن در هر نوبت یکی از مهره‌هایش را از بالا در یکی از ستون‌های جدول می‌اندازد. مهره‌ی وارد شده تا جای ممکن سقوط می‌کند که می‌تواند کف صفحه بازی و یا بر روی مهره‌ای دیگر باشد. هدف از این بازی ایجاد یک ردیف چهارتایی از مهره‌های همرنگ است. اولین بازیکنی که موفق شود چهار مهره از مهره‌های خود را در یک خط عمودی ، افقی و یا مورب پشت سرهم در صفحه بازی خط کند، برنده بازی می‌شود. بنابراین هر بازیکن علاوه بر تلاش برای رسیدن به این هدف، مانع رسیدن حریف به چهارتایی هم می‌شود.

نکات

- تابعی برای چاپ نقشه اولیه بازی به همراه کتابخانه‌ها و توابع مفید برای پیاده سازی در **پیوست** آمده است که از طریق لینک github.com/hosein-m/FundamentalOfProgramming_CE2017 قابل دسترس است.
- ابعاد صفحه بازی برخلاف نمونه‌های موجود که ۶ × ۷ است باید به صورت پارامتری و قابل تغییر در کد منبع باشد.
- برای تحویل هر یک از فازهای پروژه، تنها کافی‌است فایل‌های سورس برنامه‌تان با پسوند cpp را در سایت کویرا[1] آپلود نمایید.
- زمان تحویل هر یک از فازهای پروژه، متعاقبا توسط گروه حل‌تمرین اعلام می‌شود.
- پیاده‌سازی پروژه به صورت **انفرادی** است و تحویل نهایی آن به صورت حضوری می‌باشد.
- کپی‌برداری و یا عدم تسلط در ارائه پروژه، نمره 0 در پی خواهد داشت.
- پیاده‌سازی باید به صورت ساخت‌یافته باشد.
- تا حد امکان از تکرار کدها خودداری نمایید و از توابع استفاده کنید.
- تا حد امکان از تعریف متغیرهای سراسری خودداری نمایید.
- رعایت خوانایی کد از جمله نامگذاری معنادار متغیرها و توابع، دندانه‌گذاری و کامنت‌گذاری مهم است.
- هرگونه ایده خلاقانه در بازی و همچنین زیباسازی و رنگی کردن ظاهر بازی به منظور ایجاد تجربه کاربری بهتر می‌تواند نمره اضافه داشته باشد.
- برای ارتباط با گروه حل‌تمرین می‌توانید از طریق یکی از ایمیل‌های Hosein.mohebbi75@gmail.com و یا m.dara000@gmail.com اقدام نمایید. ضمن اینکه می‌توانید سوالات خود را در سایت کویرا و در بخش پرسش‌وپاسخ مطرح نمایید.

[1] quera.ir

فاز اول:

چاپ صفحه بازی و پیمایش بر روی آن

برای چاپ صفحه بازی می‌توانید از تابعی که در **پیوست** آمده‌است استفاده کرده و در صورت نیاز آن را به دلخواه تغییر دهید.
همچنین در این فاز از پروژه، کاربر باید با استفاده از کلیدهای جهت‌نما امکان پیمایش بر روی ستون‌های صفحه بازی را داشته باشد. برای خواندن کلید وارد شده از صفحه‌کلید، می‌توانید از تابع `getch()` در کتابخانه conio.h استفاده کنید.

فاز دوم:

قرار دادن مهره در صفحه بازی

در این مرحله از پروژه باید امکان قرار دادن مهره در صفحه بازی را برای هر دو بازیکن فراهم کنید. کاربر پس از پیمایش بر روی ستون مورد نظر، با فشردن کلید enter مهره را در ستون مورد نظر وارد می‌کند. سپس نوبت بازیکن دوم است. برای هر دو بازیکن یک مهره یکسان ولی با رنگ متفاوت در نظر بگیرید. برای نمایش مهره‌ها می‌توانید از کد اسکی 254 استفاده کنید.

فاز سوم:

پیاده‌سازی قوانین بازی

برای این منظور برنامه شما می‌بایست چهارتایی شدن مهره‌ها را در جهت‌های افقی، عمودی و یا مورب را تشخیص داده و پس از اعلام بازیکن برنده، بازی را تمام کنید.

فاز چهارم:

بازی یک‌نفره با کامپیوتر

این شرایط را در نظر داشته باشید که بیشترین حالت ممکن برای درج مهره جدید معادل است با تعداد ستون‌های جدول.
در شرایطی که یکی از ستون‌ها پر باشد، تعداد انتخاب‌ها کم‌تر هم می‌شود.

حال می‌توان نتیجه بازی را برای یک گام بعدتر، به ازای گذاشتن مهره خودی (کامپیوتر) بررسی کرد.

همچنین لازم است همین نتیجه را به ازای گذاشتن مهره حریف هم بررسی کرد.

پیاده‌سازی این فاز از پروژه با بررسی یک گام بعدتر کافی‌است هرچند برای اطمینان از پیروز شدن کامپیوتر کافی نیست.

فاز پنجم:

ذخیره و بازیابی

کاربر باید بتواند با زدن کلیدی مشخص، بازی را در یک فایل ذخیره کرده و امکان load بازی را داشته باشد.

برای این منظور برای شروع بازی یک منو بنویسید تا کاربر بین بازی جدید و یا ادامه بازی، انتخاب نماید.

پیوست برای نسخه ویندوز:

از آنجایی که پیاده سازی پروژه در محیط Visual C++ انجام می‌شود، پیاده سازی برخی توابع جهت سهولت استفاده از محیط کنسول در ذیل آمده است:

1- کتابخانه های مورد نیاز:

```
#include <stdio.h>
#include <windows.h>
#include <conio.h>
#include <time.h>
#include <ctime>
```

2- تابعی جهت جابه‌جا کردن مکان‌نما. جهت چاپ یک کاراکتر در مکانی مشخص. قبل از فراخوانی printf یا سایر توابع باید فراخوانی شود.

```
void gotoxy(int x, int y)
{
	HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
	COORD cursorCoord;
	cursorCoord.X = y;
	cursorCoord.Y = x;
	SetConsoleCursorPosition(consoleHandle, cursorCoord);
}
```

3- تابعی جهت پاک کردن صفحه نمایش

```
void clearScreen()
{
	system("cls");
}
```

4- تابعی جهت تغییر رنگ متن. با فراخوانی این تابع قبل از چاپ متن مورد نظر و دادن کد رنگ مورد نظر (از 0 تا 15) به عنوان پارامتر، رنگ نوشته به دلخواه تغییر خواهد کرد.

```
void setTextColor(int textColor, int backColor = 0)
{
	HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
	int colorAttribute = backColor << 4 | textColor;
	SetConsoleTextAttribute(consoleHandle, colorAttribute);
}
```

5- با فراخوانی این تابع و پاس دادن عددی در واحد میلی ثانیه، کار برنامه به اندازه دلخواه متوقف می‌شود.

```
void sleep(unsigned int mseconds)
{
	clock_t goal = mseconds + clock();
	while (goal > clock());
}
```

6- برای بررسی فشرده شدن کلیدی از روی صفحه کلید می‌توان از تابع `_kbhit()` استفاده کرد. این تابع در صورتی که کلیدی روی صفحه کلید فشرده نشده باشد عدد صفر و درصورت فشرده شدن کلیدی روی صفحه کلید عددی غیر صفر باز می‌گرداند. این تابع در کتابخانه conio.h پیاده سازی شده است.

7- برای محاسبه زمان می‌توان از تابع `clock()` در کتابخانه ctime استفاده کرد. این تابع زمان سپری شده از شروع برنامه را بر حسب میلی ثانیه به شما باز می‌گرداند.

8- برای به دست آوردن اعداد تصادفی می‌توان از تابع `rand()` استفاده کرد. این تابع با استفاد از الگوریتمی هر بار عددی بین 0 تا RAND_MAX برمی‌گرداند. این الگوریتم وابسته به زمان اجرای تابع نیست در نتیجه شما اگر دو بار برنامه را اجرا کنید اعداد تصادفی مشابه را دریافت می‌کنید. برای رفع این محدودیت در ابتدای برنامه تنها یک‌بار تابعِ `srand(time(NULL))` را فراخوانی کنید. از این پس در هر بار فراخوانی تابع `rand()` اعداد متفاوتی را دریافت می‌کنید.

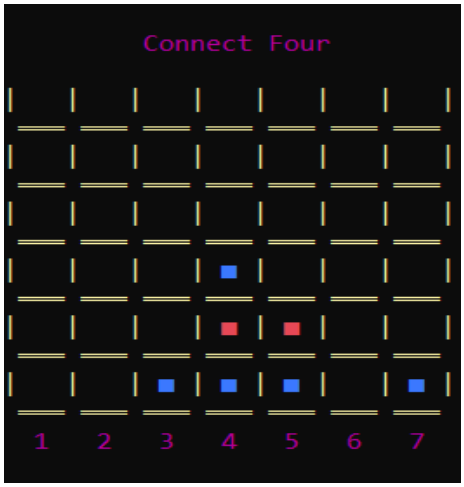
9- تابعی برای چاپ نقشه اولیه بازی:

```
#define BOARD_ROWS 6
#define BOARD_COLS 7

void printBoard() {

	printf("\n		Connect Four\n\n");
	for (int row = 0; row < BOARD_ROWS; row++) {
		for (int col = 0; col < BOARD_COLS; col++) {
			printf("|   ");
		}
		printf("|\n");
		for (int i = 0; i < 7; i++)
			printf(" %c%c%c", 205,205,205);
		printf("\n");
	}

	printf("  1   2   3   4   5   6   7 \n\n");
}
```



یک نمونه ساده از نقشه بازی

موفق باشد

تهیه شده توسط گروه حل تمرین