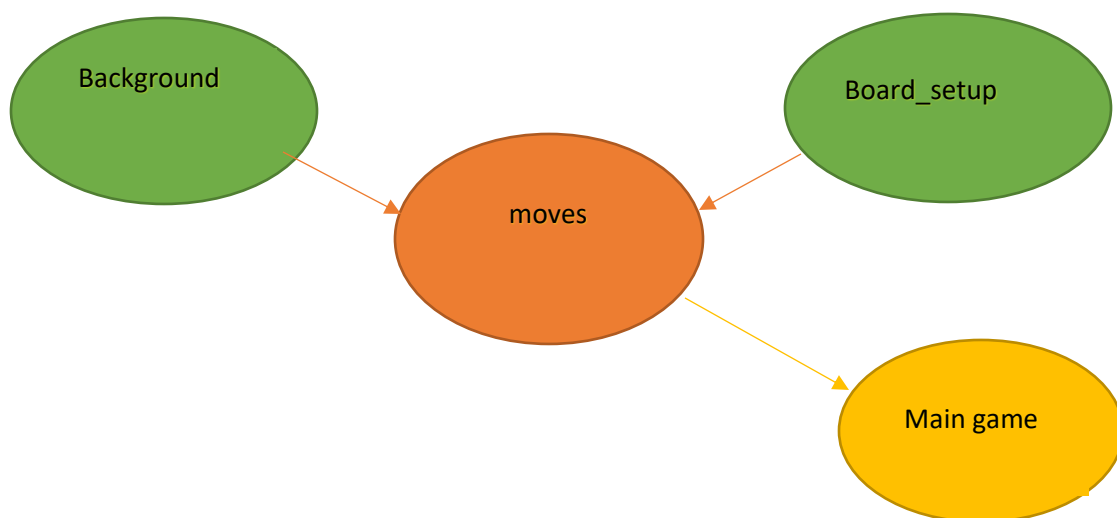


چکرز

تاریخچه: قدیمی‌ترین شیء یافت شده که به بازی چکرز شباهت دارد تخته‌ای خشتی است مربوط به ۵۰۰۰ سال پیش. این بازی در روزگار قدیم با سنگ‌های مخروطی‌شکل انجام می‌شد. در اروپا نام این بازی از ملکه فرانسه در سده‌های میانه یعنی فیرژ که بعداً «دامه» (بانو، ملکه) نامیده شد گرفته شده است.

توضیحات و هدف بازی: بازی «چکرز» در کشورهای مختلفی و البته به روش‌های متفاوتی بازی می‌شود که هرکدام از نسخه‌های آن، طرفداران خاص خود را دارد و اکثر این روش‌ها، شباهت‌های زیادی به هم دارند. حال به سراغ قوانین بازی می‌رویم. هر یک از نفرات در نوبت خود می‌تواند یکی از مهره‌هایش را یک خانه جابه‌جا کند ولی مهره‌ها همیشه باید در خانه‌های سفید باقی بمانند. همچنین هرکس باید مهره‌هایش را به سمت جلو (نسبت به خودش) حرکت دهد. اگر دو مهره غیرهم‌رنگ به طور قطری یا اریب در کنار هم باشند، می‌توان از روی مهره حریف پرید و آن را تصرف کرد (از بازی حذف کرد). البته این اتفاق در حالتی امکان‌پذیر است که پشت مهره حریف خالی باشد. اگر کسی در نوبت خود بتواند مهره‌ای را بزند، مجبور است آن را بزند و اگر بتواند چند مهره را بزند، باید یکی از آن‌ها را انتخاب کند و آن را بزند. یعنی اگر مهره‌ای برای زدن داشته باشد نمی‌تواند چیزی را نزند. همچنین اگر تعدادی از مهره‌های حریف چنان قرار داشته باشند که بازیکن بتواند با یکی از مهره‌هایش و به طور متوالی آن‌ها را بزند باید همه آن‌ها را در یک نوبت بزند. همان‌طور که در شطرنج اگر پیاده به سطر آخر برسد، تبدیل به وزیر (یا اسب یا فیل یا رخ) می‌شود، در این بازی هم اگر مهره‌ای به سطر آخر برسد تبدیل به شاه می‌شود. برتری شاه نسبت به سایر مهره‌ها این است که به عقب هم می‌تواند برگردد ولی مهره‌های عادی فقط به سمت جلو حرکت می‌کنند. کسی که مهره‌ای در زمین برایش باقی نماند، بازنده بازی است.

کد بازی و ایده



Back ground

در این قسمت ما با تعریف کردن یک آرایه هشت در هشت صفحه مهره ها و خانه های چکرز را شبیه سازی می کنیم.

در این کد 1 و -1 به ترتیب مهره سرباز X و O و 11 و -11 مهره شاه آنها هستند 0 هم خانه خالی است.

0	-1	0	-1	0	-1	0	-1
-1	0	-1	0	-1	0	-1	0
0	-1	0	-1	0	-1	0	-1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0

```
int back[8][8]={{0,-1,0,-1,0,-1},{-1,0,-1,0,-1,0},{0,-1,0,-1,0,-1},{0,0,0,0,0,0},{0,0,0,0,0,0},{1,0,1,0,1,0},{0,1,0,1,0,1},{1,0,1,0,1,0}};
```

در قسمت بعدی با گرفتن حرکت از بازیکن در صورت درست بودن آن حرکت تغییرات در ماتریس روی می دهد.

Board_setup

در این بخش با گذر از چند حلقه صفحه بازی شکل می گیرد

```

1  #include<iostream>
2  #include <Windows.h>
3  void gotoxy(int x, int y)
4  {
5      COORD coord;
6      coord.X = x;
7      coord.Y = y;
8      SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
9  }
10 }
11 using namespace std;
12 void board(){
13     cout<<"    a    b    c    d    e    f    g    h    \n"<<" ";
14     for(int i=1;i<=33;i++){
15         cout<<"- ";
16         if(i==33){
17             cout<<"\n";
18         }
19     }
20     for(int i=8;i>=1;i--){
21         cout<<" "<<i<<" ";
22         cout<<"| | | | | | | | "<<"\n ";
23         for(int i=1;i<=33;i++){
24             cout<<"- ";
25             if(i==33){
26                 cout<<"\n";
27             }
28         }
29     }
30 }
31 for(int i=9;i<=33;i=i+8){
32     gotoxy(i,2);
33     cout<<"O";
34     gotoxy(i,6);
35     cout<<"O";
36     if(i==33){
37         for(int j=5;j<=29;j=j+8){
38             gotoxy(j,4);
39             cout<<"O";
40         }
41     }
42 }
43 for(int i=5;i<=29;i=i+8){
44     gotoxy(i,12);
45     cout<<"X";
46     gotoxy(i,16);
47     cout<<"X";
48     if(i==29){
49         for(int j=9;j<=33;j=j+8){
50             gotoxy(j,14);
51             cout<<"X";
52         }
53     }
54 }
55 gotoxy(0,18);
56 }
57
58

```

	a	b	c	d	e	f	g	h
8		o		o		o		o
7	o		o		o		o	
6		o		o		o		o
5								
4								
3	x		x		x		x	
2		x		x		x		x
1	x		x		x		x	

در آخر این دو هدر فایل پیامی به فایل Moves

می فرستد سپس این فایل کار گرافیکی را روی کنسول انجام می دهد و ما را به فایل اصلی هدایت می کند و نوبت عوض می شود .

پایان