

توضیحات پروژه

طریقه نحوه اجرا و بازی :

بازی همانطور که در فایل دستور کار آمده است در کنسول ران می شود . ابتدا منو باز می شود و کاربر یکی از حالت های بازی را انتخاب می کند :

۱. بازی ۲ نفره

۲. بازی با کامپیوتر

در اینجا کافیسست عدد ۱ یا ۲ تایپ شود .

در هر قسمت از بازی برای حرکت اول :

x y

X و y مورد نظر را تایپ کنید و برای چرخش هم مثل همین ، ابتدا شماره بلوک و سپس شماره جهت را تایپ کنید :

مثال :

1 2

یعنی بلوک ۱ در جهت anti clockwise چرخانده شود .

توضیح شماره جهات :

جهت شماره ۲: anti clockwise

جهت شماره ۱: clockwise

لازم به ذکر است همه اینها به طور خلاصه در طی مراحل بازی گفته شده است .

روند حل سوال :

- توضیحات کلاس ها : من پروژه را به ۳ قسمت مهم تبدیل کردم . همانطور که در بازی مفاهیم بازیکن ، مهره ها و روند بازی و تغییرات نقشه مهم ترین اجزا هستند ، کلاس ها را به همین ترتیب تقسیم بندی کردم .

کلاس token همان مهره ی ماست که در آن یک رشته برای رنگش وجود دارد، همینطور مکانش که با x و y مشخص می شود ، یک flag دارد که نشان می دهد مکانش در روند rotation تغییر کرده است یا نه (بعداً توضیحات بیشتر داده شده است) و دیگر متود های لازمه .

کلاس player برای بازیکن هاست ، هر بازیکن لیستی از مهره هایش دارد و رنگ مخصوص به خود را دارد که یک رشته است و دیگر متود های لازمه .

کلاس PlayingSystem مهم ترین کلاس است . در آن اصلی ترین کار های بازی از نظیر چاپ نقشه ، چک کردن اینکه آیا بازی هنوز ادامه دارد یا نه ، گذاشتن یک مهره و آپدیت کردن نقشه و چرخاندن هر بلوک و بازی با کامپیوتر انجام می شود .

چالش های مهم و حل آن ها :

● چاپ نقشه

چاپ کردن نقشه توسط یک متود به نام printBoard در کلاس playingSystem و با استفاده از ۲ تا for تو در تو انجام می شود . که در هر بار چاپ در بین تمام مهره ها چک می کند اگر مهره ای با آن x و y وجود دارد به جای مهره سفید آن را چاپ کند . هر جای دیگر که مهره ای نباشد مهره ی سفید چاپ می شود .

● چک کردن اتمام بازی

توسط متود checkIfWin و checkTie در کلاس playingSystem انجام می شود .

متود checkIfWin یک رشته می گیرد که مشخص کننده رنگی است که می خواهد مهره های آن را چک کند .

سپس چک می کند که آیا یک توالی ۵ تایی از آن موجود است یا نه . این کار با چک کردن جلو و پشت آن مهره در ۴ جهت انجام می شود . من هر جهت را با i و j مشخص کردم:

۱. خط افقی : $i=1, j=0$

۲. خط عمودی : $i=0, j=1$

COMPANY NAME

۳. خط مورب حالت ۱: $j=1, i=1$



۴. خط مورب حالت ۲: $i=-1, j=1$

حالا برای هر کدام از این حالت ها ، این الگوریتم اجرا می شود :

برای هر مهره در لیست مهره تا جایی که مهره جلویییش وجود دارد و هم رنگش است ، counter را اضافه می کنیم و همین کار را برای پشتش می کنیم :

```
while (checkTokenExistence(xForward, yUpper) &&  
unicode.equals(getTokenUnicode(xForward, yUpper))) {  
    ctr++;  
    xForward += i;  
    yUpper -= j;  
}
```

//checks if there is a token behind it and the color is the same :

```
while (checkTokenExistence(xBackward, yLower) &&  
unicode.equals(getTokenUnicode(xBackward, yLower))) {  
    ctr++;  
    xBackward -= i;  
    yLower += j;  
}
```

در آخر ctr که همان counter ماست نشان می دهد چند تا مهره در یک توالی وجود دارند که اگر بزرگتر مساوی ۵ باشد ، true بر می گرداند :

```
if (ctr >= 5) {  
    return true;  
}
```

متود چک کردن تساوی (checkTie) دو چیز را چک می کند: یکی این که حرکتی دیگر برای هیچ کدام از بازیکن ها نمانده باشد و به اصطلاح صفحه پر شده باشد و هیچ کس هم برنده نشده باشد و دوم اینکه پس از انجام یک چرخش، دو توالی از دو بازیکن دیده شود .

● چرخاندن بلوک ها

چرخاندن بلوک ها را برای هر کدام از بلوک ها با توجه به محدوده x و y آن ها نوشتیم و به الگویی (گاها با استفاده از یک counter) رسیدیم و همان را برای هر کدام پیاده سازی کردیم .

برای هر دو جهات clockwise و anti clockwise همینطور است . نکته قابل توجه این است که flag مربوط به هر مهره را در ابتدای هر بار اجرا 0 میکنم ، اگر جایش را تغییر دادم flag را ۱ می کنم ، و استفاده از flag به این دلیل است که می خواهم هر token فقط یک بار تغییر کند .

تتغییر دادن مکان مهره هم توسط متود changePlace در token انجام می شود .

● بازی با کامپیوتر

توسط متود playComputer در کلاس playingSystem انجام می شود که هر بار به طور رندوم یک مکان برای مهره و یک بلوک به صورت تصادفی انتخاب می کند و جهتش هم به صورت انتخاب می شود و چرخانده می شود . (البته در انتخاب عدد رندوم برای جای مهره چک می کند که آیا از قبل در آنجا مهره ای وجود دارد یا نه و اگر وجود دارد ، یک جای رندوم دیگر پیدا میکند .)

درست است که روند بازی با همچین کامپیوتری آسان است ولی از آنجایی که همه چی به صورت رندوم انجام می شود ، کامپیوتر مثل سیستمی است که در حال بهم زدن سیستم منظم بازی است و مثلا ممکن است ناگهان بلوکی که می خواستید بچرخانید تا یک ۵ تایی بسازید را کامپیوتر در جهت مخالف بچرخاند و شما را عقب بیاندازد .

توضیحات جزئی دیگر در جاوداک و کامنت های بین کد داده شده است.