



پنتاگول!

• بازی پنتاگو:

شاید تا به حال اسم بازی پنتاگو^۱ را شنیده باشید. این بازی، یک بازی رومیزی^۲ دونفره است که بسیار شبیه به بازی دوز^۳ است. در این بازی، هر نفر یک دسته از مهره‌های بازی را در اختیار دارد و صفحه‌ی بازی، ۶ در ۶ است. هر بازیکنی که بتواند ۵ مهره را در یک خط قرار دهد، برنده‌ی بازی خواهد بود. اما تفاوت این بازی با دوز چیست؟ صفحه‌ی بازی به ۴ مربع ۳ در ۳ تقسیم شده است و هر بازیکن در نوبت خود، باید یک مهره را در صفحه قرار داده و همچنین باید یکی از این ۴ مربع را به انتخاب خود (مستقل از اینکه مهره‌ی قرار داده شده در کدام یک از این ۴ مربع است) در جهت عقربه‌های ساعت یا خلاف آن، بچرخاند؛ و سپس نوبت به بازیکن دوم خواهد رسید. همان‌طور که ذکر شد؛ بازی تا جایی ادامه می‌یابد که یک بازیکن بتواند ۵ مهره را در یک خط (افقی، عمودی و یا اریب) قرار دهد؛ مگر آن‌که صفحه‌ی بازی پر شده و هیچ بازیکنی برنده نشود. جهت آشنایی بیشتر با نحوه بازی، مشاهده‌ی صفحه بازی و بررسی یک دست از بازی به صورت نمونه، می‌توانید به این [لینک](#) مراجعه کنید.

• بازی پنتاگول:

از آن‌جا که در دانشکده‌ی برق و کامپیوتر، تعداد کثیری از دانشجویها در بازی پنتاگو ماهر هستند و بسیار به این بازی علاقه‌مندند؛ تصمیم گرفته‌ایم تا یک مسابقه‌ی پیشرفته برای پیدا کردن شاخ‌ترین پنتاگوباز برگزار کنیم؛ اما چون استراتژی برد این بازی خیلی معروف است؛ تصمیم گرفتیم پنتاگو را عوض کرده و پنتاگول را جایگزین آن کنیم.

تفاوت پنتاگول با پنتاگو تنها در آن است که باید به جای قرار دادن ۵ مهره در یک خط، آن‌ها را به فرمی شبیه به حرکت اسب (۴ مهره در یک خط و یکی در راست یا چپ آخرین مهره) در شطرنج در آورد (در این‌جا نیز می‌توانند به صورت افقی، عمودی و یا اریب قرار گیرند). توجه کنید که تفاوت حرکت اسب با حرف L در آن است که اسب می‌تواند در هر دو جهت راست یا چپ (به این شکل L یا L) حرکت کند. باقی قوانین و شرط خاتمه‌ی بازی، مشابه قبل است.

کارهای برگزاری مسابقه در حال انجام بود؛ اما متأسفانه بعد از شیوع ویروس کرونا^۴ و تعطیلی دانشگاه، دیگر امکان برگزاری این مسابقه به شکل حضوری غیرممکن شد. به همین دلیل، مشابه ارائه دروس به صورت آموزش مجازی، تصمیم گرفتیم تا مسابقه را هم به صورت اینترنتی برگزار کنیم؛ و در این راستا دانشکده از شما می‌خواهد تا برنامه‌ای بنویسید که این بازی را شبیه‌سازی کند.

¹ Pentago

² Board Game

³ Tic-Tak-Toe

⁴ Corona Virus

• توضیح پروژه:

این پروژه از دو بخش تشکیل شده است که توضیحات مربوط به هر کدام از آن‌ها، در ادامه به تفصیل آمده است.

بخش اول:

شما در این بخش موظف هستید که بازی پنتاگول را در محیط خط فرمان^۵ و بدون نمایش گرافیکی پیاده‌سازی کنید. بازی با نوبت بازیکن اول که **b** نامیده می‌شود، آغاز می‌شود. این بازیکن، با وارد کردن یک شماره (که نشان‌دهنده‌ی خانه‌ی انتخاب شده از صفحه است)، محل قرارگیری مهره‌ی خود را مشخص کرده و پس از قرار دادن مهره، به این بازیکن، اجازه‌ی چرخاندن یکی از مربع‌های بزرگ داده می‌شود. حال باید تغییرات لازم در صفحه‌ی بازی ایجاد شود. سپس نوبت بازیکن اول به اتمام رسیده و نوبت به بازیکن دوم می‌رسد؛ و بازی به همین منوال ادامه پیدا می‌کند تا زمانی که یا تمام محل‌های قرارگیری مهره‌ها پر شوند؛ و یا یکی از بازیکن‌ها بتواند شرایط برنده‌شدن را برآورده کند. در پایان، برنده‌ی بازی با نمایش پیامی مناسب مشخص می‌شود.

در طول بازی، اقدامات زیر باید انجام شوند:

۱. نمایش صفحه‌ی بازی

در هر مرحله، صفحه‌ی بازی که از ۶ ردیف و ۶ ستون تشکیل شده است باید نمایش داده شود. برای نمایش صفحه، شما باید به‌جای خانه‌های خالی از **.**، به‌جای خانه‌های پر شده با مهره‌های سیاه از **b** و به‌جای خانه‌های پر شده با مهره‌های سفید از **w** استفاده کنید. همچنین از خطوط، برای نشان دادن محدوده‌ی ۴ مربع اصلی استفاده می‌شود. برای مثال، صفحه‌ی بازی در هنگام شروع مطابق شکل زیر خواهد بود:

	1	2	3		4	5	6
1
2
3
--	--	--	--		--	--	--
4
5
6

⁵ Command Line

۲. دریافت مختصات به صورت ورودی و چرخاندن یکی از مربع‌ها

در هر مرحله، با توجه به این که نوبت کدام بازیکن است، باید مختصات خانه‌ای که بازیکن قصد انتخاب آن را دارد، به عنوان ورودی دریافت کنید. این مختصات را با یک عدد دو رقمی نشان می‌دهیم. با توجه به صفحه‌ی بازی، به هریک از خانه‌های جدول، عددی بین ۱۱ تا ۶۶ (تنها با در نظر گرفتن ارقام ۱ تا ۶) نسبت داده می‌شود. در این قسمت از پروژه، شما باید با نمایش پیامی مناسب به کاربر و گرفتن یک عدد به عنوان ورودی، خانه‌ی مورد نظر را پیدا کنید. به عنوان مثال، نحوه‌ی دریافت ورودی توسط شما می‌تواند به شکل زیر باشد:

BLACK player's turn :

در صورتی که در قسمت ورودی برنامه‌ی شما مختصات صحیحی وارد نشده باشد (مثلاً عدد ۱۹ نامعتبر خواهد بود)، باید این موضوع با نمایش پیامی مناسب اطلاع داده شود و مجدداً از بازیکن درخواست شود تا مختصات را وارد نماید. همچنین اگر بازیکن مختصاتی که قبلاً در آن مهره‌ای قرار داده شده را وارد کرد، باید با نمایش پیامی مناسب اطلاع داده شود و برنامه، دوباره از او مختصات را به عنوان ورودی بگیرد. در صورتی که مشکلی وجود نداشت؛ باید مهره‌ی که متعلق به این بازیکن در مختصات وارد شده قرار بگیرد، تغییرات لازم در صفحه‌ی بازی اعمال شود و سپس، صفحه‌ی بازی به‌روزشده و نمایش داده‌شود.

حال، باید شماره‌ی یکی از ۴ مربع بزرگ و سپس، جهت چرخش آن از بازیکن گرفته شود. به منظور تعیین مربع بزرگ، یکی از چهار عدد ۱، ۲، ۳ و ۴ به عنوان ورودی گرفته می‌شود. عدد ۱ نشان دهنده مربع واقع در ناحیه اول مختصات، عدد ۲ نشان دهنده مربع واقع در ناحیه دوم مختصات و باقی اعداد نیز به همین صورت هستند. جهت چرخش نیز با استفاده از یک علامت + و یا - مشخص می‌شود. به‌طوریکه علامت + نشان‌دهنده چرخش به اندازه مثبت ۹۰ درجه (در جهت عقربه‌های ساعت) و علامت - نشان‌دهنده چرخش به اندازه منفی ۹۰ درجه (خلاف جهت عقربه‌های ساعت) می‌باشد. این قسمت از ورودی نیز باید چک شود و در صورت وجود هرگونه مشکل در ورودی کاربر، باید با نمایش پیامی مناسب دوباره این ورودی را دریافت کنید. لازم به ذکر است که پس از انجام چرخش نیز باید صفحه‌ی بازی به‌روزشده و نمایش داده‌شود.

۳. پایان بازی

در هر مرحله از بازی، شما موظف هستید که صفحه‌ی بازی را به درستی چک کنید. حالتی وجود دارد که در آن‌ها تمامی خانه‌های جدول، پر شده‌است و یا برنده بازی مشخص شده‌است. در این حالات بازی به اتمام رسیده و باید خاتمه یابد.

۴. نمایش نتیجه‌ی بازی

پس از اتمام بازی، برنامه‌ی شما باید برنده‌ی بازی را نمایش دهد. برای مثال، این پیغام می‌تواند به شکل زیر باشد:

The WHITE player has won!

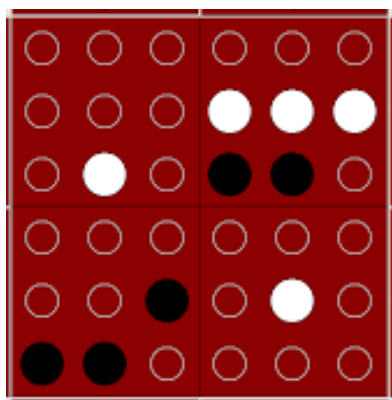
بخش دوم:

در این بخش، شما باید بتوانید با استفاده از منطق بازی که در قسمت قبلی پیاده‌سازی کرده‌اید، بازی را به صورت گرافیکی پیاده‌سازی کنید. برای این قسمت یک کتابخانه‌ی گرافیکی در اختیار شما قرار خواهد گرفت که باید از آن استفاده کنید.

مشابه بخش اول، در طول بازی باید اقدامات زیر انجام شوند:

۱. نمایش صفحه‌ی بازی

در هر مرحله، شما باید به جای نمایش صفحه‌ی بازی در محیط خط فرمان^۶، صفحه‌ی گرافیکی بازی را نشان دهید که مربعی ۶ در ۶ است. همچنین در خانه‌ی ۱۱ جدول، یک مربع کوچک و توخالی (مربع نشان‌گر) قرار گرفته‌است که نشان‌دهنده‌ی جای انتخابی مهره می‌باشد. لازم به ذکر است که در مراحل بعدی، کاربر باید بتواند برای انتخاب جای مهره‌ی موردنظر، این مربع را در صفحه حرکت دهد. ابعاد و رنگ‌های استفاده شده در این صفحه به صورت دلخواه بوده و مثالی از آن در شکل زیر آمده‌است:



۲. مشخص کردن جای مهره

در هر مرحله، با توجه به این که نوبت کدام بازیکن است، باید با گرفتن کاراکترهای w, a, s, d جای مهره را در بازی صفحه‌ی بازی مشخص کنید. همچنین، باید هم‌زمان با هر جهتی که وارد می‌شود، مربع نشان‌گر حرکت کرده و روی خانه‌ی جدید واقع شود. در انتها کاربر با فشردن کلید f مهره را در خانه‌ی موردنظرش قرار می‌دهد. توجه کنید که کاربر نمی‌تواند مهره‌ش را در خانه‌ای قرار دهد که قبلاً اشغال شده‌است. پس از قرار دادن مهره به صورت صحیح، صفحه‌ی گرافیکی باید به‌روز شده و نمایش داده‌شود. همچنین برای نشان دادن این که نوبت کدام بازیکن است، می‌توان کاری مشابه شکل زیر انجام داد:

BLACK player's turn :

حال شما باید کاملاً مشابه بخش اول، پس از تعیین محل مهره جدید به صورت صحیح، شماره‌ی یکی از ۴ مربع بزرگ و سپس، جهت چرخش آن را به عنوان ورودی دریافت کنید. سپس باید صفحه‌ی بازی در محیط گرافیکی به‌روز شده و نمایش داده‌شود؛ اما

⁶ Command Line

سایر مراحل (مانند ورودی گرفتن برای تعیین مربع بزرگ و جهت چرخش) نیازی به پیاده‌سازی گرافیکی ندارد. (در صورت پیاده‌سازی گرافیکی برای این قسمت، نمره‌ی امتیازی به آن تعلق خواهد گرفت.)

مراحل "پایان بازی" و "نمایش نتیجه‌ی بازی" نیز مشابه بخش اول خواهند بود.

• چند نکته و راهنمایی:

- استفاده از هرگونه توابع مربوط به C++ ممنوع است و نمره‌ای به آن تعلق نمی‌گیرد.
- بهتر است هر کدام از مراحل مختلف بازی را در تابع‌های جداگانه بنویسید و از آن‌ها استفاده کنید. برای مثال می‌توانید یک تابع برای نمایش صفحه‌ی بازی بنویسید و در مرحله‌ی نمایش صفحه، تنها آن را صدا بزنید؛ و یا اینکه می‌توانید یک تابع برای چک کردن تمام شدن بازی بنویسید و در زمان لازم از آن استفاده کنید.
- بهتر است که از یک حلقه‌ی کنترلی کلی برای تکرار مراحل کلی بازی استفاده کنید.
- می‌توانید برای وضعیت هر کدام از خانه‌ها از یک آرایه‌ی دوبعدی کمک بگیرید.
- توجه داشته باشید که شما باید در این پروژه، توابع خود را در فایل‌های جداگانه با هدر فایل^۷های مخصوص خود بنویسید. شیوه‌ی تقسیم‌بندی توابع به خودتان مربوط است و هر روش قابل قبولی پذیرفتنی است.
- برای پیاده‌سازی قسمت گرافیکی این پروژه، کفایت از کد نوشته شده برای قسمت اول (خط فرمان^۸) استفاده کنید و در هر مرحله به جای نمایش صفحه‌ی بازی در خط فرمان، تغییرات لازم را در صفحه‌ی گرافیکی بازی اعمال کنید.
- برای آشنایی با توابع گرافیکی ویدیویی در صفحه‌ی درس قرار می‌گیرد که می‌توانید از آن استفاده کنید.
- همچنین می‌توانید از این [لینک](#) برای آشنایی با توابع کتابخانه گرافیکی استفاده کنید.

• شیوه‌ی نمره‌دهی:

عنوان	نمره
نام‌گذاری مناسب و اصولی متغیرها	۵
استفاده از تمام ورودی‌های تابع در آن و نام‌گذاری مناسب توابع	۵
عدم وجود قطعه کد تکراری	۵

^۷ Header File

^۸ Command Line

۵	نمایش درست صفحه ی گرافیکی بازی
۵	نمایش درست صفحه ی بازی در command line
۵	دریافت صحیح ورودی گرافیکی
۵	دریافت صحیح ورودی در command line
۵	استفاده از آرایه ی دوبعدی
۱۰	رسیدگی به خطاها
۵	نمایش پیام های مناسب در هر مرحله از بازی
۵	تغییر درست جای مربع مشخص کننده ی جای مهره
۵	تغییر درست نوبت ها با توجه به شرایط بازی و خطاها
۱۰	پیاده سازی منطق برد و باخت بازی به شکل صحیح
۵	کامنت گذاری مناسب در هرجایی که نیاز به مستندسازی دارد
۲۰	تست و اجرای برنامه

• نحوه ی تحویل پروژه:

۱. تنها فایل های با فرمت “c.” و “h.” را در یک فایل زیپ با فرمت “zip.” و با نام CA1_SID.zip قرار دهید که SID همان شماره ی دانشجویی شماست. برای مثال اگر شماره ی دانشجویی شما ۸۱۰۱۹۸۰۰۰ باشد باید نام فایل خود را CA1_810198000.zip قرار دهید.

۲. تمامی پروژه ها برای یادگیری برنامه نویسی و مباحث مطرح شده در کلاس طراحی می شوند و انجام آنها به صورت انفرادی خواهد بود. بنابراین، در صورت شباهت میان دو پروژه (که به وسیله ی نرم افزارهای مربوطه بررسی خواهند شد) برای هر دو نفر نمره ی صفر در نظر گرفته خواهد شد.

۳. در صورت وجود هرگونه سوال می توانید پرسش های خود را در فروم درس (در بخش مربوط به این پروژه) مطرح نمایید و یا به ut.icsp98.ca@gmail.com ایمیل بزنید.

موفق باشید

میرحامد جعفرزاده، یاسمین نیکنام و حمید سالمی