

شطرنج

در این پروژه قصد داریم تا بازی شطرنج را پیاده‌سازی کنیم.

همه با قواعد بازی شطرنج آشنا هستیم و این قواعد را تکرار نمیکنم. در [اینجا](#) قواعد بازی و نحوه حرکت مهره‌ها توضیح داده شده.

تو این پروژه قصد نداریم که تمامی قواعد بازی شطرنج رو پیاده‌سازی کنیم و تنها یک حالت ساده از بازی رو پیاده‌سازی خواهیم کرد. مهم‌ترین چیز که باید پیاده‌سازی بشه اینه که حرکت مهره‌ها به درستی انجام بشه. به عنوان مثال وقتی کاربر می‌خواد مهره اسب رو حرکت بده تنها حرکت‌های به صورت L براش قابل انجام بشه.

بعضی از چیزهایی که ارزش تو این بازی **صرفه نظر می‌کنیم** به صورت زیر هست:

- تبدیل شدن مهره سرباز در پایان مسیر.
- محدودیت کیش شدن.
- جابجایی رخ و شاه.

به طور کلی با استثنای بازی (مثلا آن‌پاسان) کاری نداریم و فقط می‌خوایم یه بازی ساده شطرنج پیاده‌سازی کنیم.

برنده شدن رو به این صورت تعریف می‌کنیم که شاه طرف مقابل از بین بره. توجه کنید چون که قاعده کیش شدن رو پیاده‌سازی نمی‌کنیم مشکلی پیش نخواهد آمد.

بازی ما به صورت **گرافیکی** انجام میشه برای این کار از کتابخانه **swing** استفاده کنید.

سعی کنید برای اینکه پیاده‌سازی تمیزی از بازی داشته باشید ابتدا تمام موجودیت‌های بازی شطرنج رو شناسایی کنید مثلا مهره‌ها، صفحه شطرنج، بازیکن‌ها و وقتی که این موجودیت‌ها رو شناسایی کردید به ازای هر کدوم یک کلاس درست کنید. بعد از این باید روابط رو هم شناسایی کنید مثلا می‌دونیم که چند نوع مهره داریم این مهره‌ها ویژگی خاص خودشون دارن ولی با این حال تمام ویژگی‌های مهره بودن رو هم دارا می‌باشن. ما می‌تونیم این واقعیت رو اینطور پیاده‌سازی کنیم که کلاسی مانند اسب بسازیم و اون رو فرزند کلاس **مهره** قرار بدیم.

اول از همه می‌ریم سراغ انواع بازیکن:

بازیکن‌ها

تو این بازی شطرنج ما دو نوع بازیکن داریم:

- بازیکن انسانی <- HumanPlayer
- بازیکن ماشینی با هوش مصنوعی <- AIPlayer

هر دو HumanPlayer و AIPlayer، رابط Player را پیاده‌سازی می‌کنند. کلاس AIPlayer و رابط Player در اختیار شما قرار می‌گیرد اما HumanPlayer باید توسط شما پیاده‌سازی شود. این نوع بازیکن از طریق رابط کاربری و کلیک کردن بازی را کنترل می‌کند.

```
1 public interface Player {
2
3     int[][] nextMove(int[][] positions, boolean white);
4 }
```

کلاس AIPlayer و کلاس‌های مرتبط با اون رو از [اینجا](#) دانلود کنید.

در ادامه توضیح می‌دیم که ورودی و خروجی این تابع چی هستند. پارامتر white مشخص می‌کنه که بازیکن مهره‌های سیاه رو تکون می‌ده یا سفید یعنی اگر این ورودی true باشه، بازیکن مهره سفید رو تکون میده. ورودی positions و خروجی این تابع هر دو یک آرایه دو بعدی هستند که وضعیت صفحه شطرنج رو نشون می‌دن. توجه کنید که هر دو این آرایه‌ها 8×8 هستند. ورودی positions وضعیت بازی در زمان حال رو نشون میده و خروجی وضعیت صفحه رو بعد از حرکت بازیکن نشون میده.

برای اینکه بتونیم این آرایه رو بسازیم به هر نوع مهره یک عدد نسبت میدیم:

```
1 int KING = 1;
2 int QUEEN = 2;
3 int ROOK = 3;
4 int BISHOP = 4;
5 int KNIGHT = 5;
6 int PAWN = 6;
```

همونطور که مشاهده می‌کنید مثلا به اسب عدد 5 نسبت داده شده. مقادیری که در بالا اومدن برای مهره‌های سفید هستند. برای مهره‌های سیاه قرینه همین مقادیر را استفاده می‌کنیم.

حال با استفاده از مقادیر فوق وضعیت صفحه شطرنج در ابتدای بازی به صورت زیر خواهد بود.

که در واقع معادل با حالت زیر در شطرنج است:

در نتیجه برای استفاده از تابع nextMove باید وضعیت مناسب بازی به صورت همچین آرایه‌ای در بیاید و سپس به این تابع پاس داده شود. وضعیت در مرحله بعد خروجی خواهد بود.

ذخیره بازی

شما باید اطلاعات بازی خود را داخل یک فایل ذخیره کنید تا کاربر بتواند بعد از بستن برنامه دوباره بازی خود را از سر بگیرد. در صفحه اصلی خود دکمه‌ای برای ادامه دادن بازی قرار دهید و توجه کنید که اگر کاربر قبل از این بازی نکرده بود دکمه ادامه دادن مثل قبل کار نمی‌کند و تنها پیامی به کاربر نشان می‌دهد که بازی‌ای ذخیره نشده است.

در این بخش نیاز خواهیم داشت که تمام اطلاعات مورد نیاز برای ادامه بازی را ذخیره کنیم. مثلا این که چه مهره‌هایی بیرون هستند و این که کدام بازیکن انسانی است و کدام بازیکن ماشینی و رنگ هر بازیکن کدام است.

صفحه اصلی برنامه

در صفحه اصلی برنامه دکمه‌های مدیریت برنامه قرار گرفته است. به عنوان مثال دکمه‌های زیر باید وجود داشته باشد.

- شروع بازی جدید
- ادامه بازی قبل

جزئیات بازی

بازی دو حالت زیر را خواهد داشت:

- بازی بازیکن انسانی با بازیکن انسانی
- بازی بازیکن انسانی با بازیکن ماشینی

توجه کنید که بعد از شروع بازی جدید باید صفحه‌ای باز شود که نوع بازی را مشخص کند.

بعد از این مرحله باید رنگ مهره‌های هر بازیکن مشخص شود. اگر یکی از بازیکن‌ها ماشینی بود تنها رنگ بازیکن انسانی را مشخص کنید. اما اگر هر دو بازیکن انسانی بودند نیازی به مشخص کردن رنگ مهره‌ها نیست چرا که اولویتی وجود ندارد.

- آغاز بازی با مهره سفید است.
- در صفحه بازی باید مهره‌های حذف شده از هر بازیکن نمایش داده بشه.
- توابع اصلی خود را کامنت گذاری کنید تا ورودی، خروجی و نحوه استفاده از آنها مشخص باشد.
- بازی در حالت عادی بدون انیمیشن است و انجام آن به صورت امتیازی است.

بخش امتیازی

در بخش امتیازی حرکت اشیاء بازی به صورت انیمیشنی است. برای این کار باید بازی را به صورت **multi thread** `پیاده‌سازی کنید.

در **این** مخزن یک بازی مار پیاده‌سازی شده که به صورت **چندنخی** است می‌توانید از آن به عنوان راهنما استفاده کنید.

توضیحات

همه فایل‌های خود را در پوشه قرار دهید و به صورت فشرده بارگذاری کنید. توجه کنید که در بالای فایل‌های خود package تعریف نکرده باشید.

گزارش

حتما برای پروژه خود یک گزارش شامل معرفی همه پیاده‌سازی‌های انجام شده توسط شما ارائه کنید. در این گزارش باید توضیحات کامل راجع به روش حل مسائل، کلاس‌ها، ارشبری‌ها و سایر کدهای نوشته شده آورده شده باشد و ملاک اصلی نمره‌دهی این گزارش خواهد بود.

****** همچنین دقت داشته باشید که گزارش‌ها باید در فرمت پایان نامه‌های دانشگاه امیرکبیر باشد.******

لینک دریافت نمونه پایان‌نامه دانشگاه صنعتی امیرکبیر