پروژه مبانی کامپیوتر

نام و نام خانوادگی : مهدی خوش‌نویس

کد دانشجویی : ۴۰۲۱۰۰۴۳۱

خلاقیت‌ها : نحوه نام‌گذاری خانه‌ها، flip و قرینه کردن صفحه بازی با عوض شدن نوبت سیاه و سفید، اضافه کردن سیستم امتیاز به بازی و تغییر رنگ صفحه و مهره‌های بازی

در پیاده سازی این پروژه از ۱۰ class و یک interface استفاده شده که در ادامه هرکدام توضیح داده شده است:

پکیج Pieces:

این پکیج شامل یک کلاس Piece است که این اطلاعات را ذخیره می‌کند.

- Row: ردیف این خانه در صفحه بازی است.
- Col: ستون این خانه در صفحه بازی است.
- Name: اسم این خانه (Unicode آن در کلاس PieceName) است. در صورت null بودن این متغیر یعنی این خانه خالی است.
- Color: رنگ این خانه (سیاه یا سفید) را نشان می‌دهد.
- Backcolor: این متغیر رنگ background خانه را ذخیره می‌کند.
- firstMove: این متغیر از نوع Boolean است و دو حالت آن به این معنا است: در صورت false بودن یعنی این مهره تا الان حرکتی انجام نداده و true بودن آن هم نشان دهنده این است که حداقل یه حرکت انجام داده است. (در حرکت Pawn ها کمک می‌کند.
- gameManger: این متغیر از جنس GameManager است که در ادامه آن را توضیح آن آورده شده است و باعث می‌شود به مولفه‌های دیگر بازی هم دسترسی داشته باشیم.

این کلاس دو نوع constructor دارد که یکی از آنها برای مقداردهی row و col خلنه‌های خالی است و دیگری برای مقداردهی متغیرهای خانه شامل مهره است.

این کلاس دو متود isValid و isValid2 است که به این صورت عمل می‌کنند:

- isValid: چک می‌کند که خانه مورد نظر داخل صفحه بازی باشد و شامل مهره هم رنگ نباشد.
- isValid2: خالی بودن و داخل صفحه بازی بودن یک خانه را بررسی می‌کند.
- getValue: ارزش هرمهره را برمی‌گرداند؛ به این صورت که ارزش وزیر ۹، رخ ۵، فیل و اسب ۳ و سرباز یک است. از این متود برای سیستم امتیازدهی بازی استفاده می‌شود.

شش کلاس King، Queen، Rook، Knight، Bishop، Pawn هم کلاس این کلاس را extends کرده و همگی شـــامل تابع move هســـتند که حرکات مهره‌ها را با توجه به نوع آنها انجام می دهد و آنها را در isValid و isValid2 چـک می‌کنـد. (البتـه کـه این دو متود کـافی نیســـتنـد). این کلاس‌هـا اینترفیس Directions را implements می‌کنند که با توجه به نام گذاری متفاوت خانه‌های صـــفحه بازی به پیاده‌سازی راحت move کمک می‌کند.

در اینترفیس Directions:

در این اینترفیس ۱۲ جهت (همه جهت‌ها به جز حرکات اسب) نگه داشته می‌شود.

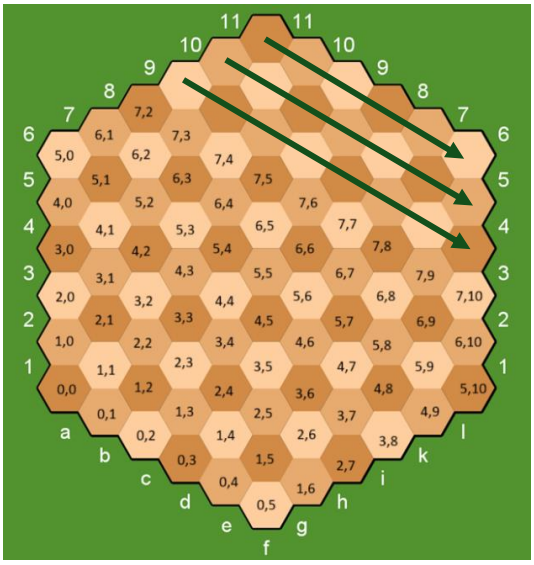
کلاس Pair:

با توجه به نداشتن Pair در جاوا این کلاس شاخته شد که پیاده‌سازی پروژه ساده‌تر شود.

کلاس Listener:

همانطور که در داک پروژه آمده این کلاس شامل ۴ متود است که هرکدام کارهای زیر را می‌کنند:

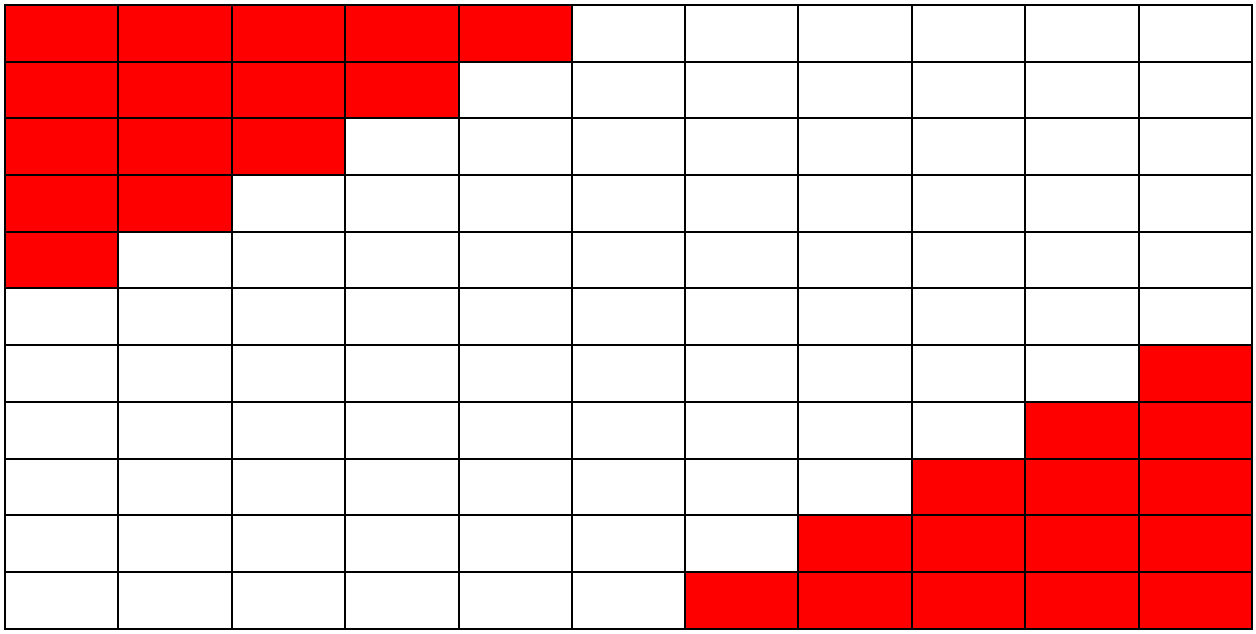
- Onload: زمانی که فایلی انتخاب می‌شود و باتوجه به آن فایل صفحه بازی جدیدی لود می‌شود.
- OnSave: فایل و صفحه بازی را در این موقعیت خاص save و دخیره می‌کند.
- OnNewGame: زمانی که می‌خواهیم یک بازی جدید را شروع کنیم. (بعد از بستن پیام popout در Mate و Stalemate این تابع صدا زده می‌شود.)
- Onclicked: در این تابع همراه گرفتن row و col خانه‌ای که روی آن کلیک شـده اسـت، آن را به row و col جدید (با توجه به نام‌گذاری جدید خانه‌ها) تبدیل می‌کند.



کلاس GameManger:

متغیرها:

- lastRow و lastCol: اشاره دارند به آخرین ردیف و آخرین ستونی که روی آنها کلیک شده.
- Counter: تعداد مهره‌های خارج از گیم را نشان می‌دهد.
- Board: این آرایه دو بعدی در واقع صفحه بازی است که همه خانه‌های آن به غیر از خانه‌های خارج از صفحه بازی از نوع Piece است (King، Queen، Rook، Knight، Bishop و Pawn هم در نظر گرفته شده اند).



- خانه‌های خالی (null) با قرمز مشخص شده‌اند.
- isBlackMove: همانطور که از اسم آن مشخص است، اگر نوبت سیاه باشد این متغیر true و در غیر این صورت اگر نوبت سفید باشد، false است.
- isSecondClick: اگر true باشد، به این معنا است که کلیک دوم است و باید جایی که مهره مورد نظر می‌خواهد برود را انتخاب کنید و اگر false باشد باید روی خانه‌ای که می‌خواهید مهره آن را جابجا کنید کلیک کنید. (در صورت کلیک روی مهره‌ای که نوبت رنگ آن نیست یا خانه خالی از مهره، مقدار این متغیر false باقی می‌ماند. در ضمن برای راحت‌تر تشخیص اگر true باشد پیام choose your move و در صورتی که false باشد choose your piece نمایش داده می‌شود.)
- isBlackCheck و isWhiteCheck: هر رنگی که کیش باشد متغیر مربوط به کیش بودن یا نبودن آن رنگ true خواهد شد.

- File -> Game.txt: در این فایل اطلاعات بازی بعد از هر نوبتی که طی می‌شود، ذخیره خواهد شد و در حالتی که وجود نداشته باشد این فایل ساخته شده و اطلاعات صفحه اولیه بازی از طریق متود defaultBoard در آن قرار داده می‌شود.
- stringColors: این آرایه که از جنس StringColor است برای نگهداری اطلاعات مهره‌های خارج از بازی استفاده می‌شود.
- validMoves: این آرایه و آرایه‌های مشابه آن حرکات معتبر (با توجه به توابع isValid و isValid2) مهره انتخاب شده را نگه می‌دارد.
- Min و Max : این دو آرایه کمترین و بیشترین شماره ردیف را برای هر ستون ذخیره می‌کنند.

در کانستراکتور GameManager، min و max هر ستون، شماره ردیف واقعی و کاراکتر مربوط به هر خانه (با توجه به توابع صفحه بازی ارائه شده در داک پروژه) تعیین می‌شود. همچنین اگر فایل "Game.txt" موجود باشد، صفحه بازی با توجه به آن فایل ساخته می‌شود و در غیر این صورت صفحه اولیه بازی نشان داده می‌شود.

متودها:

- defaultBoard: وظیفه این متود تشکیل صفحع اولیه و پیش فرض بازی است.
- readFile: این تابع در صورت کلیک روی دکمه Load یا وجود فایل "Game.txt" در شروع بازی صدا زده می‌شود. و صفحه بازی رو با توجه به فایلی که می‌گیرد تغییر می‌دهد. (اگر مهره‌ای مختصات "11,11" داشته باشد مه این معنا است که از بازی خارج شده است.)
- writeFile: در تابع اطلاعات بازی و مهره‌ها بعد از هر حرکت به این صورت ذخیره می‌شود:
  - اگر مهره داخل بازی باشد: ۱- ردیف ۲- ستون ۳- اسم آن (Unicode) ۴- رنگ ۵- آیا اولین حرکت خود را انجام داده است یا نه.

```
out.println(j + " " + i + " " + Board[j][i].name + " " + color + " " + firstMove);
```

  - اگر مهره داخل بازی نباشد: ۱- ردیف و ۲- ستون (هر دو ۱۱ نوشته می‌شوند.) ۳- اسم آن (Unicode) ۴- رنگ (در اسم و رنگ مهره از آرایه stringColors استفاده می‌شود.)

```
out.println("11 11 " + stringColors[i].getText() + " " + color);
```

- Clicked: این متود تقریبا به این صورت کار می‌کند: (در هر نوبت خانه‌ای که روی آن کلیک شده، آبی می‌شود.)

- اگر اولین حرکت باشد، در صورتی که روی مهره رنگی که نوبت آن است کلیک کنید، حرکاتی که می‌تواند انجام دهد با سبز نشان داده می‌شوند و isSecondClick به true تغییر می‌کند.
- اگر اولین حرکت نباشد، در هر صورت بعد از کلیک isSecondClick به false تغییر می‌کند و اگر روی یکی از خانه های سبز کلیک کنید حرکت انجام می‌شود، isBlackTurn برعکس شده و متود isValid صدا زده می‌شود، در صورتی که این متود false به ما بدهد حرکت بازگردانده می‌شود (مهره‌ها به جای قبل بازگشته و isBlackTurn به حالت قبل باز می‌گردد) و اگر به ما true بدهد، مراحل زیر را طی می‌شود:
- آیا سرباز به پایان صفحه رسیده است؟
  - بله؛ پس پیام promote نشان داده شود و سرباز به مهره‌ای که بازیکن از بین Queen، Rook، Knight و Bishop انتخاب کرده تغییر کند.
- آیا رنگ مقابل چک شده است؟
  - بله. آیا حرکت دیگری دارد؟
    - بله؛ پس حریف چک شده است و رنگ پس زمینه مهره شاه حریف قرمز شود.
    - خیر؛ پس حریف مات شده است.
  - خیر. آیا حرکت دیگری دارد؟
    - خیر؛ پس بازی با stalemate به پایان می‌رسد.
- ردیف و ستونی که روی آن کلیک شده به عنوان lastRow و lastCol ذخیره می‌شود.
- این موقعیت و state بازی در فایل "Game.txt" نوشته و ذخیره شود.

- isThereAMove: نشان می‌دهد آیا رنگ مورد نظر حرکتی درستی دارد یا نه:
  - اگر رنگ مورد نظر کیش باشد:
    - اگر حرکتی نداشته باشد، مات می‌شود.
    - اگر حرکتی داشته باشد، کیش می‌شود.
  - اگر رنگ مورد نظر کیش نباشد:
    - اگر حرکتی نداشته باشد، بازی با stalemate تمام می‌شود.
    - اگر حرکتی داشته باشد، بازی ادامه پیدا می‌کند.

- isCheck: با مشاهده‌کردن تمام جاهایی که مهره‌های حریف می‌توانند آن را ببینند (جزو جاهایی باشد که می‌تواند برود.) چک می‌کند که آیا رنگی انتخاب شده کیش شده است یا نه. (در صورتی کیش شده که مهره‌های حریف بتوانند شاه رنگ انتخاب شده را ببینند.)
- Point: در این متود امتیاز هرمرحله بازی (بعد از تغییر امتیاز که در صورتی اتفاق می‌افتد که سرباز promote شود یا یک مهره از بازی خارج شود.) محاسبه می‌شود و به صورت یک پیام نشان داده می‌شود.
- isValid: کار آن به صورت آن این است که چک نهایی را برای معتبر بودن یک حرکت انجام دهد. به این صورت که اگر بعد از این حرکت مهره‌های حریف بتوانند شاه این رنگ را ببینند، این حرکت درست نیست و در غیر این صورت معتبر و درست است و نوبت تمام می‌شود.
- Compare: این متود که boolean تحویل می‌دهد، مهره‌های بیرون بازی را مقایسه می‌کند و در مرتب کردن و sort آنها کمک می‌کند. نحوه مرتب کردن این مهره‌ها به این صورت است:

سپس مهره‌های سیاه رنگ **ابتدا مهره‌های سفید رنگ**

- Draw: در این متود که در آخر هر نوبت یا بعد از خواندن فایل ها صدا می‌شود، صفحه بازی با دستوراتی که در application موجود است، رسم می‌شود که شامل جزئیات زیر است.
  - پیام که ۴ حالت دارد:
    - isSecondClick:false, isBlackMove:true <- Black’s Turn: Choose your piece
    - isSecondClick:true, isBlackMove:true <- Black’s Turn: Choose your piece
    - isSecondClick:false, isBlackMove:flase <- White’s Turn: Choose your piece
    - isSecondClick:true, isBlackMove:flase <- White’s Turn: Choose your move
  - مهره‌های خارج از بازی به صورت مرتب و sort شده.

- مهره‌های داخل بازی که دو حالت دارند: ۱- نوبت سفید است و به صورت عادی رسم می‌شود. ۲- نوبت سیاه هست و نسبت به خط وسط صفحه بازی قرینه می‌شود. (به این صورت در همه حالات و نوبت ها سربازها به سمت بالا حرکت می‌کنند.)

پایان :))