

به نام خدا.

- این فایل را برای توضیح بیشتر توابع و کارکرد آن ها ایجاد کردم تا در صورت نیاز از آن استفاده شود.
- تمامی توابع در JavaDocs توضیح داده شده و کاربرد آن ها نیز مشخص است اما درمورد کارکرد بعضی توابع توضیح مختصری در این فایل می دهم.
- اول این نکته را بیان کنم که به دلیل کمبود وقت و تاخیر در تدریس نکات وراثت در این پروژه از قواعد وراثت زیاد استفاده نشده ( کلاس Board ) اما سعی شده برنامه تا جای ممکن به توابع کوچک تر شکسته شود و کلاس های جدا برای راحتی کار ایجاد شوند. اما نکات وراثت در پروژه های 2 و 3 به خوبی رعایت شده اند.

توضیح کارکرد توابع و کلاس ها :

همان طور که ملاحظه می کنید برنامه ما تشکیل شده از هفت کلاس می باشد که مهم ترین آن ها کلاس Board ( که کل توابع بازی را شامل می شود ) و کلاس Othello ( که رابط بین بازی و کاربر است ) هستند.

بازی به این گونه طراحی شده که یک آرایه  $6 * 6$  دو بعدی برای صفحه در نظر گرفته شده و دو آرای لیست دیسک های توی بازی (inBoardDisk) و دیسک های راهنما (hintDisk) را شامل می شود. دیسک های راهنما با استفاده از توابع findHintHouses و makeHintHouses ابتدا شناسایی و سپس ساخته می شوند ( به این صورت که هر خانه دارای مهره خود را در نظر گرفته و 8 جهت ممکن را بررسی می کنیم تا به مهره هایی از جنس مخالف مهره های خود برسیم ). اما دیسک های توی بازی نقش کمکی را برای بازی کامپیوتر و همچنین ساختن دیسک های راهنما ایجاد می کنند.

روش کار برای یافتن hintDisk به این صورت است که یک خانه را از خانه های board انتخاب می کنیم و بررسی می کنیم که آیا در 8 جهت خانه ی انتخابی به مهره ای با جنس مخالف می رسیم یا خیر. سپس از بین آن ها یکی را انتخاب و مهره را قرار می دهیم و دوباره 8 مسیر در 8 جهت را طی کرده و تغییرات را پیاده سازی می کنیم.

مهم ترین تابع این کلاس تابع placeADisk است که توسط تابع changeInput صدا زده می شود و اینگونه ورودی کاربر تبدیل به مختصات صفحه شده و پس از اینکه صحت خانه ی وارد شده بررسی شد مهره داخل صفحه قرار می گیرد و تبدیل های لازم در صفحه را نیز ایجاد می کند ( توابع changeDisks و searchLine و checkLine ).

تابع makeInput تابع هوش مصنوعی این بازی است که در دو حالت کار می کند ، یا یک خانه ی تصادفی را انتخاب می کند و یا بهترین خانه از لحاظ گرفتن بیشترین مهره با جنس مخالف را انتخاب می کند. ( تابع findBestHouse ). و تابع showInput نشان می دهد که کامپیوتر چه خانه ای را انتخاب کرده است.

کلاس Othello نیز یک حلقه دارد تا فرآیند بازی را پیاده سازی کند و نوبت را تغییر دهد و صفحه را نمایش دهد و ورودی ها را دریافت کند همچنین پایان بازی را بررسی کند.