

بازی توسط شی Game ایجاد شده و تغییرات از طریق توابع کلاس Othello انجام میشوند. ویژگی های بازی، یک صفحه ی بازی در حال حاضر، یک صفحه حاوی حرکت قبلی برای undo کردن، بازیکن، برنده، مهره های باقیمانده برای هر بازیکن (که برای بررسی و انتخاب برنده استفاده میشود)، مهره های هر بازیکن در زمین و backup این ویژگی هاست.

در ابتدا چهار مهره در برد قرار میگیرد و بازیکن اول، بازیکن سیاه است. این عملیات توسط تابع newgame انجام میشود. مشابه این تابع، تابع restart وجود دارد که در صورت تمایل کاربر برد را به حالت اولیه بر میگرداند.

تابع continueGame از فایل حاوی آخرین بازی، اطلاعات آخرین بازی را میخواند و اجازه میدهد که بجای ساخت بازی جدید، در برد بازی قبل ادامه دهید.

تابع saveFile برای ذخیره ی بازی هنگام خروج از بازی است. هنگام خروجی از بازی از شما پرسیده میشود که آیا میخواهید بازی را ذخیره کنید؟ در صورت تمایل، بازی شما بعنوان آخرین بازی در فایل game.txt ذخیره میشود.

تابع displayBoard صفحه ی بازی را نمایش میدهد که در واقع در یک آرایه ی دوبعدی 8×8 ذخیره شده. جهت ایجاد ویژگی امتیازی، هر خانه ای که بازیکن بتواند در آن مهره ی خود را قرار دهد با * نمایش داده میشود. شماره خانه های صفحه ی بازی از ۰ تا ۷ برای سطر و ستون است.

تابع getPlayer و تابع getOppPlayer برای مشخص کردن بازیکن استفاده میشود و در واقع چون شماره ی بازیکن یک ویژگی پراپیت است، بااستفاده از این تابع میتوان به این ویژگی دسترسی داشت

تابع isGameOver در ابتدا به کمک شمارش مهره های روی صفحه بررسی میکند که آیا برنده ای داریم یا نه، در صورتی که از این طریق برنده قابل تشخیص نبود، به کمک مهره های باقیمانده بررسی میکند که بازی به پایان رسیده یا نه-مهره های باقیمانده صفر باشند- اگر بازی به پایان نرسیده باشد، false برمیگرداند، در غیر اینصورت به کمک تعداد مهره های روی صفحه برنده را تعیین میکند و true بر میگرداند. برنامه هربار این تابع را فراخوانی میکند تا چک کند بازی برنده ای دارد، و به پایان رسیده یا نه. در صورتی که بازی به پایان رسیده باشد ولی برنده ای نداشته باشد، مقدار winner برابر با ۰، اگر بازیکن سفید ببرد مقدار winner برابر با یک و اگر بازیکن سیاه ببرد این مقدار برابر با منفی یک لحاظ میشود.

در همین تابع، اطلاعات بازی نیز در فایل ذخیره میشود تا آخرین ده بازی در فایل نگهداری شوند. برنده این بازی در فایل gamesit.txt ذخیره میشود.

درهربار، بازیکن باید شماره سطر و ستون را وارد کند، در صورتی که از محدوده ۰ تا ۷ خارج باشند، تا زمانی که کاربر عدد صحیح وارد کند برنامه از کاربر عدد دریافت میکند. در صورتی که کاربر بخواهد برنامه را restart کند کافیست عدد ۱۰ را وارد کند، و در صورت اینکه بخواهد برنامه را ببندد کافیست عدد ۱۱ را وارد کند.

سپس به کمک تابع `isValidMove` بررسی میشود که آیا حرکتی که کاربر مدنظر دارد، قابل انجام است یا خیر و هشت جهت آن بررسی میشود. بررسی این عمل به کمک `checkFlip` انجام میشود.

در صورتی که حرکت قابل انجام بود، به کمک تابع `makeMove` حرکت انجام میشود. در این تابع هم از `checkFlip` استفاده شده که قبلاً توضیح داده شد، و همچنین از `changeCoins` استفاده شده که در واقع خانه هایی که باید به نفع بازیکن تغییر کنند را با توجه به اینکه قابل تغییر هستند یا نه، تغییر میدهد.

سپس به کاربر، صفحه ی بازی جدید نمایش داده میشود و از او پرسیده میشود که میخواهد حرکتش را تایید کند، یا حرکتش را برگرداند. در صورتی که بخواهد برگرداند، از آنجا که در تابع `makeMove` مقدمات بکاپ گیری انجام شده، به کمک تابع `undo` این عمل انجام میشود، در صورت تایید، بازیکن دیگر میتواند بازی را ادامه دهد.

هنگامی که تابع `isGameOver`، `true` برگرداند بازی تمام میشود و برنده به کمک `getWinner` که تابعی برای دسترسی به برنده ی بازیست، معلوم میشود.

تابع `thelast10games` برای نمایش نتیجه ی آخرین ده بازی بکار میرود.

داده های استفاده شده در این بازی:

Class

Array 2D

Int

Char

Bool