

CLUE

گزارش پروژه برنامه نویسی پیشرفته

آناهیتا حیدری

1399

کلیت برنامه

برنامه دارای سه کلاس Main, Clue و Player است. کاربر با بازیکنانی رقابت می کند که هر کدام پس از ارائه ی حدس خود، در صورت اطمینان، حدس نهایی می زند.

کلاس Player شامل متغیر های public برای هر بازیکن و کلاس Clue شامل توابع و متغیرهای مربوط به بازی است. تمامی توابع استفاده شده در Main متعلق به کلاس Clue هستند. در ابتدای هر بازی، لیست weapons, characters, rooms و همچنین تعداد بازیکنان که رندوم است، نمایش داده می شود.

کاربر، همیشه آخرین شماره را دارد اما شروع کننده به صورت رندوم انتخاب می شود و بازیکنان به صورت افزایشی(افزایش عدد منتسب به هر بازیکن) بازی می کنند. این به این معناست که کاربر لزوما آخرین بازیکن نیست.

در ادامه، توابع و متغیر ها، با توجه به تقدم در Main بررسی می شوند.

```
public void start(int numberOfPlayers);
```

هدف این تابع، تعیین یک کارت از هر دسته و جداسازی آن ها، همچنین پر کردن آرایه ۱۸ تایی total از باقی مانده کارت ها می باشد. آرایه های Oweapon, Ocharacter, Oroom شامل سه دسته کارت با ترتیب اصلی و ثابت در هر بازی هستند. آرایه های weapon, character و Sroom از کلاس Clue کمکی هستند و در نهایت با اجرا شدن این تابع، weapon[0], character[0] و Sroom[0] سه کارت انتخاب شده برای این بازی خواهند شد.

```
public void YatesShuffle();
```

۱۸ کارت موجود در total را بر می زند.

سپس در Main به هر کدام از بازیکنان به صورت رندوم، تعدادی کارت داده می شود. ترتیب رندوم به این دلیل است که چون کاربر آخرین شماره را دارد، احتمال داشتن کارت های بیشتر (مثلا وقتی تعداد بازیکنان ۵ باشد، برخی بازیکنان ۳ کارت و برخی ۴ کارت می گیرند) را هم داشته باشد. هر بازیکن در کلاس Player دارای متغیر numberOfCards و آرایه card است. در ابتدای بازی، کارت های کاربر نمایش داده می شوند.

public int search(Player plr,String card,int n,boolean b);

هر بازیکن در کلاس Player آرایه های Ocharacter, Oroom و Oweapon دارد که در طی بازی عدد هایی با مفاهیم زیر به آن ها منتسب می شود:

1	0	-1	-2
کارت جزء سه کارت انتخاب شده است	مقدار اولیه برای هر کارت	کارت متعلق به بازیکن های دیگر است	کارت متعلق به خود بازیکن است

این تابع، اندیس کارتی نامشخص را در دسته کارتی که با n مشخص شده، return می کند. ضمن این کار، آرگومان آخر بیانگر این است که کارت برای خود بازیکن است یا خیر و به همین ترتیب، مقدار کارت را 2- یا 1- می کند.

هر کارت که مقدار آن ۱ شود، با توجه به نوع آن، در یکی از متغیر های sfroom, sfweapon یا sfcharacter ذخیره می شود. همچنین متغیر nf نشان دهنده ی تعداد کارت هایی است که ۱ شده اند و بدیها هرگاه برای هر بازیکن، مقدار این متغیر ۳ شود، بازی تمام می شود.

public void guess(Player plr);

while(flag) حلقه اصلی برنامه (Main) است که در آن هر دور بازی تا زمانی انجام می شود که یکی از بازیکنان حدس نهایی بزند

تابع guess برای هر بازیکن غیر کاربر حدس می زند. هر کدام از سه کارت حدسی، در متغیر های room, character و weapon ذخیره می شوند. آرایه بولین broom از کلاس Clue، مشخص کننده ی پر بودن یا خالی بودن هر اتاق است. آرایه valid شامل شماره اتاق های مجاز (با توجه به قوانین بازی برای انتخاب اتاق) برای هر بازیکن با توجه به عدد تاس و متغیر j تعداد اتاق های مجاز می باشد که تا زمانی که j مخالف صفر شود عدد تاس ساخته می شود (اگر تعداد بازیکنان ۶ باشد و اتاق کنونی بازیکنی غیر ۱ باشد و دو اتاق مجاورش خالی باشند، تنها یک اتاق مجاز خالی برای بازیکن موجود است که اگر زوجیت تاس با زوجیت عدد اتاق یکی نباشد، j صفر باقی می ماند؛ به همین دلیل تعیین اتاق های مجاز در حلقه قرار گرفته است). استراتژی کلی برای انتخاب هر کارت بدین صورت است که اگر متغیر nororoom, norowapon یا norocharacter (با توجه به نوع کارت) که بیانگر تعداد کارت هایی از دسته مربوطه که مقدار آن ها 0 است،

مثبت باشد، کارتی که مقدارش 0 است را انتخاب می کند. (این متغیر ها تا زمانی که کارت مربوط به قتل پیدا نشود، مثبت هستند)

اگر مقدار این متغیر صفر باشد (یعنی کارت مربوط به قتل در دسته ی مربوطه پیدا شده است) در درجه اول، کارتی را انتخاب می کند که مقدارش 2- باشد و اگر پیدا نشد، رندوم انتخاب می کند.

متغیر های room, rnc و rnw شماره کارت های انتخابی را نگه می دارند که مقدار room برای هر بازیکن در ابتدا، 1- در نظر گرفته شده است. (به این معنا که بازیکن در هیچ اتاقی نیست).

معرفی متغیر های o و q

در هر بار حدس زدن یک بازیکن، متغیر o برای بازیکن های دیگر، به معنای تعداد کارت های خود آن ها یا تعداد کارت های ۱ شده ی آن ها که در حدس بازیکن فعلی، موجود هستند.

متغیر q برای هر بازیکن غیر از بازیکن حدس زننده، صفر است اگر هیچ کس یا هیچ کس غیر از خودش کارتی به حدس زننده نشان ندهد و یک است اگر بازیکنی غیر از خودش، به بازیکن حدس زننده کارتی نشان دهد.

کاربرد آن ها در تابع OPQ توضیح داده می شود.

پس از اینکه یک بازیکن حدس زد، از بازیکن بعدی تا قبلی، کارت ها چک می شوند و همزمان متغیر های o و q در صورت درست بودن شرایط، مقدار دهی می شوند.

در نهایت اولین بازیکنی که کارتی از حدس را داشته باشد (حتی کاربر) کارت خود را فقط به بازیکن حدس زننده نشان می دهد و مقدار آن برای بازیکن حدس زننده 1- می شود.

اگر هیچ بازیکنی کارتی را نشان ندهد (یعنی هیچ کارتی از حدس را نداشته اند)، برای هر کارت حدسی چک می شود که اگر متعلق به کارت های خود حدس زننده نباشد، حتما جزء کارت های قتل است؛ پس مقدار آن برای حدس زننده 1 می شود.

public void OPQ(Player plr);

اگر برای بازیکنی، $q=1$ و $o=2$ شود (یعنی دو کارت از سه کارت حدسی، یا متعلق به این بازیکن است یا مقدارش برای این بازیکن ۱ است و بازیکنی کارتی را به حدس زنده نشان داده)، نتیجه آن است که کارتی که نه مقدارش ۱ است و نه متعلق به این بازیکن است، حتما متعلق به بازیکنی است که کارتی را نشان داده است؛ پس مقدار آن برای این بازیکن 1- می شود (کارت مورد نظر در qArray ذخیره شده است).

public boolean check(Player plr);

رای بازیکنی که حدس زده است، اگر مقدار متغیرهای `norochacter`، `nororoom` و `norowapon` برابر 1 باشد (یعنی از دسته ی مربوطه، تنها یک کارت وجود دارد که مقدارش صفر است و بقیه یا 1- هستند یا 2-) نتیجه می شود که آن کارت حتما مربوط به قتل است؛ پس مقدار آن را ۱ می کند.

خروجی تابع `true` است اگر `nf` برابر 3 باشد (یعنی هر سه کارت مربوط به قتل، توسط بازیکن شناسایی شده اند و بازی تمام است) و در غیر این صورت `false` است.

نوبت کاربر

public void die(Player plr);

این تابع به همان صورتی که در تابع `guess` توضیح داده شد، اتاق های مجاز را برای کاربر شناسایی کرده و چاپ می کند.

کاربر باید حدس خود را به صورت عددی سه رقمی به ترتیب لیست ها و با توجه به اتاق های مجاز، وارد نماید. مدیریت خطا برای این عدد ابتدا با `InputMismatchException` و سپس با تابع `exceptionHandlingS` برای جلوگیری از نقض قوانین بازی انجام می شود.

```
public boolean exceptionHandlingS(int n);
```

اگر عدد ورودی، سه رقمی نباشد، رقم اول و سوم غیر از ۱ تا ۶ باشند، رقم دوم صفر باشد و یا رقم دوم غیر از اتاق های مجاز باشد، خروجی تابع `false` و در غیر این صورت `true` خواهد بود.

در نهایت، اطلاعات این عدد با تابع `guessOfUser` بررسی می شوند. بازیکنان از حدس کاربر همانند حدس دیگر بازیکنان، استفاده خواهند کرد.

مشابه برای حدس نهایی کاربر، مدیریت استثنا (با تابع `exceptionHandlingF`) انجام می شود (اگر کاربر حدس نهایی نداشته باشد عدد صفر را وارد می کند).

اگر حدس نهایی کاربر نادرست باشد، با چاپ سه کارت مربوط به قتل، بازی تمام می شود.

پایان