

پروژه ی برنامه نویسی پیشرفته

تاریخ تحویل ۲۸ خرداد ۱۳۹۳

در این پروژه قصد داریم یک بازی ساده و خوب را که با تاس انجام می شود پیاده سازی کنیم. این بازی شامل یک صفحه N در N است که از تعدادی خانه ی تله و یا قدرت تشکیل شده است و به کمک تاس بازیکن ها در آن جلو می روند. هدف رسیدن به خانه ی آخر قبل از رقبای دیگر است.

قسمت های اصلی

در ابتدا همه ی بازیکن ها بیرون از زمین بازی قرار دارند و به ترتیب تاس می اندازند. هر نفر که عدد ۶ بیاورد می تواند وارد زمین بازی شود. در حالت کلی نیز بعد از هر عدد ۶ بازیکن می تواند یک تاس دیگر بیندازد. سپس بازی ادامه پیدا می کند تا اینکه اولین نفر به خانه ی آخر برسد. دقت کنید که هیچ وقت نباید در خانه ی آخر تله یا قدرتی قرار داشته باشد.

زمین بازی

همانطور که گفته شد زمین بازی مربعی N در N است (حداکثر مقدار N ، ۲۰ است). خانه های این زمین یکی از سه حالت زیر است:

- خانه معمولی: این خانه ها هیچ خطری ندارند و بازیکنان می توانند در آنها تا نوبت بعدی استراحت کنند. تنها خطری که در این خانه ها وجود دارد وارد شدن یک بازیکن دیگر به آن است که باعث بیرون رفتن بازیکن اول از زمین می شود.
- تله: خانه های تله اصولاً خانه های خطرناکی هستند، و بعد از هر بار اثر کردن، آن تله از بین می رود و آن خانه به یک خانه معمولی تبدیل می شود. تله ها به شرح زیر هستند:
 - ۱- تله ی نابودگر: اگر بازیکنی وارد این تله شود، آن بازیکن از زمین خارج می شود و باید از نو بازی را شروع کند.
 - ۲- تله کند: اگر بازیکنی وارد این تله شود برای دو حرکت بعدی اش ۵۰ درصد کند خواهد شد. مثلاً اگر تاس ۴ یا ۵ بیاورد فقط ۲ خانه به جلو خواهد رفت.
 - ۳- تله غیرفعال گر: اگر بازیکنی وارد این تله شود در نوبت بعدی قادر به بازی نخواهد بود. توجه کنید که مثلاً اگر بازیکن با تاس ۶ وارد این خانه شود دیگر نمی تواند حرکت کند، ولی از حرکت بعدی می تواند به حرکت خود ادامه دهد (یعنی فقط جایزه اش از بین می رود) ولی اگر با ۶ وارد نشده بود در نوبت بعدی قادر به بازی کردن نیست و یک نوبت از او گرفته می شود.
 - ۴- تله فرستادن به جای تصادفی: اگر کسی وارد این تله شود به یکی از خانه های قبلی به صورت تصادفی فرستاده می شود، دقت کنید که اگر خانه مقصد تله یا قدرت یا بازیکنی دیگر باشد باید برای هر کدام کاری مناسب انجام شود.
 - ۵- تله معکوس کننده: اگر کسی وارد آن تله شود، برای دو دور مسیر حرکتش برعکس می شود، یعنی اگر تاسی بیندازد که او را ۵ خانه به جلو ببرد، باید ۵ خانه عقب برده شود.
- قدرت: قدرت نیز مانند تله فقط یک بار مصرف می شود و از بین می رود. لیست قدرت ها در زیر آمده است:
 - ۱- قدرت فرستادن به عقب: بعد از به دست آوردن این قدرت بازیکن حق دارد یک بازیکن دلخواه را به عقب هل دهد.
 - ۲- قدرت فرستادن به نزدیکترین تله: بعد از به دست آوردن این قدرت بازیکن می تواند یک بازیکن دیگر را انتخاب کند و بازیکن انتخاب شده به نزدیک ترین تله فرستاده می شود.
 - ۳- قدرت سرعت: با گرفتن این قدرت بازیکن در دو دور بعد دو برابر پرس می کند، یعنی به جای ۲ خانه ۴ خانه جلو می رود.

لازم به ذکر است یک بازیکن هیچ وقت خودش را هل نخواهد داد یا به تله ای نخواهد فرستاد.

ورودی برنامه

ورودی برنامه در چندین خط است که به شکل کلی زیر است:

در خط اول N خوانده می شود که اندازه صفحه بازی است. سپس در خط بعد M وارد می شود که تعداد بازیکن ها است و در خط بعدی M رشته خوانده می شود که نام بازیکن ها است. بازیکن ها به همان ترتیبی که در ورودی نام برده شده اند بازی خواهند کرد. سپس عدد P خوانده می شود که مجموع تعداد تله ها و قدرت ها است و در هر کدام از P خط بعدی نیز سه عدد خوانده می شود که به ترتیب شماره سطر و ستون و نوع آن خانه است. انواع خانه شامل TR (تله نابودگر)، TS (تله کند)، TD (تله غیرفعال گر)، TG (تله فرستادن به جای تصادفی)، TM (تله معکوس کننده)، PB (قدرت فرستادن به عقب)، PN (قدرت فرستادن به نزدیک ترین تله) و PS (قدرت سرعت) است.

یک نمونه از ورودی به صورت زیر است:

```
4
2
Me You
5
2 3 TS
3 3 PB
4 3 TR
1 2 TD
1 3 PN
```

نکته هایی که باید در نظر بگیرید به شرح زیر است:

- اعداد سطرها و ستون ها از 1 تا N هستند.
- هیچ تله یا قدرتی در خانه آخر قرار ندارد.
- اگر به کمک قدرت، بازیکن A بازیکن B را به خانه ای فرستاد که بازیکن C در آن قرار دارد، بازیکن C از بازی خارج می شود.
- اگر نزدیک ترین تله از دو سمت بازیکن به یک فاصله باشد اولویت با تله عقب است.
- گر هیچ نزدیک ترین تله ای وجود نداشته باشد، هیچ اتفاقی برای بازیکن نمی افتد و چیزی هم چاپ نمی شود.
- نوبت اضافی بعد از هر تاس 6 فراموش نشود.

مکانیزم تاس انداختن

به علت مشکلاتی که در تحویل حضوری به وجود می آید، بازی شما باید دو مکانیزم برای تاس انداختن داشته باشد. یکی مکانیزم تصادفی که فرد بتواند با آن تاس بیندازد، دومی امکان اینکه کاربر عدد تاس را خودش وارد کند و انگار آن عدد واقعا عدد تاس است.

مدیریت خطاها

شما باید خطاهایی که ممکن است در برنامه به وجود آیند را به طور مناسب مدیریت کنید. بعضی از این خطاها عبارتند از:

- بازیکنی خارج زمین باشد: این خطا مربوط به وقتی است که کسی بازیکنی را انتخاب کند که بیرون از زمین است، مثلاً برای استفاده از قدرت!
- بازیکنی نتواند عقب برود: وقتی که کسی را آنقدر به عقب بفرستیم که از صفحه خارج شود، در این صورت فرد به عقب نمی رود و خطایی ایجاد می شود.
- بازیکنی از جلوی زمین بازی بیرون بیفتد: مثلاً در خانه ی یکی مانده به آخر باشد و ۵ بیاورد، در این هنگام نیز باید با خطا به طور مناسب رفتار شود.
- عدم وجود نزدیک ترین تله: وقتی نزدیک ترین تله ای وجود نداشته باشد.

این ها خطاهایی هستند که برخورد با آنها اجباری است، یافتن بقیه ی انواع خطاهای بازی اختیاری است.

نحوه ی پیاده سازی

کلاس هایی که پیشنهاد می شود در نظر گرفته شوند به شرح زیر است (البته دقت کنید هنگام نوشتن پروژه ممکن است کلاس های دیگری هم به ذهنتان برسد) :

کلاس BoardCell:

این کلاس به ازای هر کدام از خانه های صفحه ی بازی است؛ پس به تعداد $N*N$ تا از این کلاس باید نمونه گرفته شود (یعنی باید $N*N$ تا شی از این کلاس ساخته شود). باید مشخصات مربوط به هر خانه ای در این کلاس قابل دستیابی و یا تغییر باشد.

کلاس Board:

این کلاس به ازای یک صفحه ی بازی است، پس باید در آن $N*N$ تا از BoardCell ها باشد. همین طور کارهای مربوط به صفحه ی بازی باید در این کلاس انجام شود. مثلاً باید یک تابع داشته باشد که با گرفتن شماره ی بازیکن و تاسی که آورده است حرکت مربوطه را روی صفحه انجام دهد.

کلاس Player:

این کلاس باید مشخصات و مکان حضور و همینطور قدرت ها یا تله هایی که به این بازیکن برخورد کرده است را جوری پیاده سازی کند تا مشخص باشد که این بازیکن باید چگونه حرکت کند. این کلاس باید کارهای مربوط به بازیکن را نیز انجام دهد مثلاً به روز رسانی ویژگی های بازیکن.

کلاس PlayerList:

این کلاس یک لیست از بازیکن ها را در خود نگهداری می کند و همینطور نگه می دارد که نوبت کدام یک از بازیکن ها است. از این کلاس می توان به عنوان این که هر بازیکن در کدام خانه قرار دارد و در چه وضعیتی است کمک گرفت، همینطور می توان تشخیص داد که آیا بازیکنی بازی را برده است یا خیر.

کلاس Game:

این کلاس، کلاس اصلی پروژه است که در آن باید کارهای کلی مربوط به بازی را انجام دهید، پس این کلاس دارای یک نمونه از کلاس Board و یک نمونه از کلاس PlayerList است و اعداد تصادفی تولید می کند. درواقع حلقه ی اصلی اجرای بازی در این کلاس است.

خروجی برنامه

مواردی که باید در خروجی چاپ شوند به شرح زیر است:

- وقتی کسی برنده می شود باید پیام زیر چاپ شود که اصولاً آخرین پیام بازی است. دقت کنید که در این مثال ورودی که ما استفاده کردیم بازی در همه پیام ها بین دو بازیکن Me و You است (چون فقط دو بازیکن داریم):

[Me] has won the game.

- وقتی کسی تاس می اندازد:

[You] dice=3.

- وقتی کسی دچار تله می شود:

[Me] is reset.

[Me] is slowed.

.

- وقتی کسی قدرتی را استفاده می کند:

[Me] send You back by 3.

[Me] send You to nearest trap at 50.

.

- وقتی کسی با ورود به خانه ای کس دیگری را می کشد:

[Me] killed You.

- وقتی نوبت کسی می شود:

[Me] 's turn to play.

- وقتی کسی می خواهد وارد خانه ای شود:

[Me] moved to 50.

- به محض وارد شدن بازیکنی به یک خانه:

[Me] @ X.

که X مشخص کننده خانه ی نرمال، تله ی کند، تله ی نابودگر، ...، یا قدرت سرعت، قدرت فرستادن به عقب، ... است (یعنی نوع خانه را مشخص می کند).

- بعد از پایان کار هر بازیکن نیز باید تخته ی بازی چاپ شود. منظور از پایان کار یعنی بعد از آنکه تعدادی ۶ یا غیره آورد و قبل از اینکه نفر بعدی تاس بیندازد. تخته به شکل زیر چاپ می شود:

تخته به صورت یک سری نماد دو حرفی مانند شکل زیر چاپ خواهد شد که __ نمایانگر خانه معمولی و نمادهای گفته شده برای تله ها و قدرت ها نماد هر یک از آنها و شماره ها هم نماد بازیکن ها خواهند بود. ضمناً حداکثر بازیکنان ۵ نفر است.

__ TD __ 01

__ TS __ 02

__ __ PB __

__ TR __ __

نکته ای که باید حتماً توجه کنید این است که تخته از نظر شماره خانه باید به صورت مارپیچ در نظر گرفته شود، یعنی به صورت زیر:

1 2 3 4

8	7	6	5
9	10	11	12
16	15	14	13

مثلا در شکل بالا بازیکن اول در خانه ۴ و بازیکن دوم در خانه ۵ قرار دارد و تله کند کننده نیز در خانه ۷ قرار گرفته است.

نکاتی که درمورد خروجی باید رعایت شود به شرح زیر است:

- بازی از خانه ۱ شروع می شود.
- بعد از هر ۶ دوباره عبارت [Me] 's turn to play چاپ شود.
- اگر خانه ی بعدی یک نفر خارج از صفحه است(قبل از درنظر گرفته تله و قدرت) عبارت [Me] moved to 50. نباید چاپ شود.

قسمت اختیاری(نمره اضافی):

انواع بازیکن ها

- بازیکن مهاجم: این بازیکن هر بار که تاس می اندازد 50 درصد احتمال دارد دو بار بپرد، یعنی انگار دو بار پشت هم تاس انداخته.
 - بازیکن مدافع: این بازیکن به احتمال 50 درصد وقتی وارد تله می شود، تله را از بین می برد و اتفاقی برایش نمی افتد، اگر هم بازیکنی وارد خانه ی او شود، باز به احتمال 50 درصد او بازیکن را خواهد کشت.
 - بازیکن مرده خور: این بازیکن بعد از هر حرکت 25 درصد شانس دارد تا جای خود را با یک بازیکن دیگر در زمین عوض کند، او اگر بخواهد می تواند از این فرصت استفاده نکند.
 - بازیکن معمولی که در طول پروژه توضیح داده شد.
- نوع بازیکن را هر نفر در اول بازی مشخص می کند.

موفق و پیروز باشید.