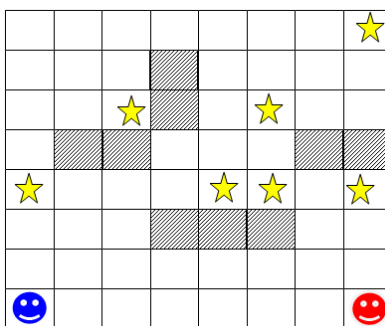


پروژه میان ترم درس برنامه نویسی پیشرفته (جاوا)

توضیح پروژه:

در این پروژه قرار است شما یک بازی دونفره را پیاده سازی کنید.

در این بازی دو بازیکن شرکت دارند: بازیکن قرمز و بازیکن آبی. هر بازیکن دارای یک مهره است. مهره بازیکن قرمز را مهره قرمز و مهره بازیکن آبی را مهره آبی می نامیم. این بازی یک صفحه شامل تعدادی سطر و تعدادی ستون دارد. بعنوان مثال، در تصویر زیر، یک نمونه از صفحه بازی نمایش داده شده است که دارای ۸ سطر و ۸ ستون است. مهره آبی در خانه $[1,1]$ قرار دارد (سطر ۱ و ستون ۱) و مهره قرمز در خانه $[1,8]$ قرار دارد (سطر ۱ و ستون ۸). علاوه بر مهره های قرمز و آبی، تعدادی دیوار، ستاره و سرعت گیر هم در صفحه وجود دارد. در شکل زیر، خانه های هاشور خورده، به معنای وجود دیوار هستند.

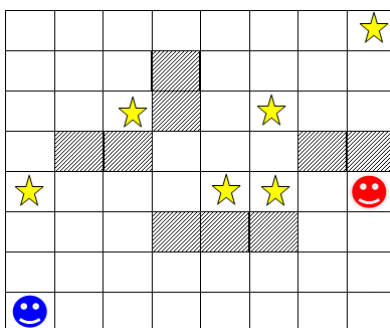


بازیکنان قرمز و آبی، به نوبت، یکی پس از دیگری، مهره خود را حرکت می دهند. هدف بازی، جمع کردن ستاره ها می باشد. وقتی تمام ستاره ها جمع شوند، بازی خاتمه می یابد و هر بازیکنی که ستاره های بیشتری جمع کرده باشد، برنده است.

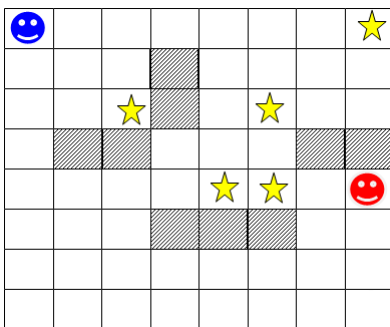
نحوه حرکت مهره ها و نحوه جمع کردن ستاره ها

هر بازیکنی که نوبتش است، باید مهره اش را به یک خانه جدید حرکت دهد. خانه مقصد باید در همان سطر یا در همان ستون که مهره قرار دارد قرار داشته باشد. بنابراین مهره ها فقط حرکت عمودی و افقی دارند و حرکت اریب مجاز نیست. نکته مهم آن است که بین خانه ای که مهره در آنجا قرار دارد، و خانه مقصد، نباید دیواری وجود داشته باشد. به بیان دیگر، مهره ها نمیتوانند از دیوار عبور کنند. ضمناً خانه مقصد نمیتواند خانه ای باشد که مهره حریف در آنجا قرار دارد. بنابراین دو مهره نمیتوانند در یک خانه قرار بگیرند. البته اگر مهره حریف بین مکان فعلی مهره و خانه مقصد قرار داشته باشد اشکالی ندارد. به بیان دیگر، یک مهره میتواند از روی مهره حریف بپرند اما نمیتواند از روی دیوار بپرد.

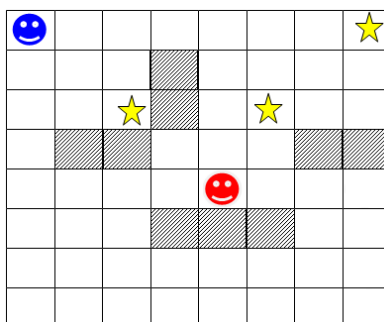
وقتی بازیکن، خانه مقصد را انتخاب کرد، مهره اش به آن خانه منتقل می شود و تمام ستاره هایی که در مسیر حرکت او (از خانه مبدأ تا خانه مقصد) قرار داشته اند، را جمع می کند و این ستاره ها از صفحه بازی حذف و به اندوخته آن مهره اضافه می شوند. همچنین اگر در این مسیر سرعت گیری وجود داشته باشد، آن سرعت گیر نیز جمع آوری می شود و از صفحه بازی حذف می شود. هر سرعت گیر یک ویژگی به نام حد (limit) دارد که مقدار آن عددی مثبت است. وقتی بازیکنی با حرکت مهره اش یک سرعت گیر که مقدار حدش برابر n است را جمع آوری می کند مهره حریف در نوبت بعد نمی تواند بیش از n خانه جابجا شود. به زبان دیگر، یک بازیکن با جمع آوری سرعت گیر، قابلیت تحرک مهره حریف را کاهش می دهد. اگر بازیکنی در یک حرکت چند سرعت گیر را جمع آوری کند آن سرعت گیر ها به همان ترتیبی که جمع آوری شده اند در نوبت های بعدی بر روی حرکت حریف اثر می گذارند. مثلاً اگر بازیکن آبی ۳ سرعت گیر با حدهای ۲ و ۳ و ۵ جمع آوری کند یعنی مهره قرمز در حرکت بعدش تنها ۲ خانه، در حرکت بعد از آن تنها ۳ خانه و در حرکت بعد از آن تنها ۵ خانه می تواند حرکت کند. فرض کنید وضعیت فعلی بازی همان چیزی است که در تصویر بالا نشان داده شده و نوبت بازی مهره قرمز است. مهره قرمز از خانه $[1,8]$ به خانه $[4,8]$ حرکت می کند (حرکت به بالا). و در نتیجه ستاره ای که در خانه $[4,8]$ قرار دارد را برمی دارد. اگر در خانه $[5,8]$ دیواری وجود نداشت، مهره قرمز، میتواندست بجای حرکت به خانه $[4,8]$ به خانه $[8,8]$ حرکت کند تا با یک حرکت هر دو ستاره موجود در خانه های $[4,8]$ و $[8,8]$ را تصاحب کند. اکنون وضعیت بازی بدین ترتیب است:



حال نوبت بازیکن آبی است. مهره آبی از خانه $[1,1]$ به خانه $[8,1]$ حرکت می کند و ستاره موجود در خانه $[4,1]$ را تصاحب می کند. اکنون وضعیت بازی بدین ترتیب است:



حال نوبت بازیکن قرمز است. مهره قرمز از خانه [4,8] به خانه [4,5] حرکت می کند و دو ستاره موجود در خانه های [4,6] و [4,5] را تصاحب می کند. اکنون وضعیت بازی بدین ترتیب است:



و بازی ادامه پیدا می کند....

خوب این از توصیف بازی. حال توضیح اینکه پروژه شما باید چه امکاناتی داشته باشد.

برنامه ای که شما می نویسید دارای ۲ قسمت می باشد.

قسمت اول، قسمت آماده سازی است که طی آن یک کاربر اقدام به طراحی صفحه بازی میکند. بدین منظور برنامه شما باید امکانات زیر را برای کاربر فراهم کند:

- تعیین ابعاد صفحه
- ایجاد و نمایش وضعیت اولیه صفحه
- قرار دادن دیوارها در موقعیت دلخواه
- قرار دادن ستاره ها در موقعیت دلخواه
- قرار دادن سرعت گیرها و تعیین مقدار حد آنها
- قرار دادن مهره قرمز و آبی در موقعیت دلخواه

قسمت دوم: قسمت اجرای بازی است. برنامه شما باید بازی را مدیریت کند. برنامه شما باید از واسط گرافیکی مناسبی برخوردار باشد و این امکان را بدهد که با استفاده از mouse، بازی را انجام داد. مثلاً برای جابجا کردن مهره، بازیکنی که نوبتش است باید روی خانه مقصد کلیک کند. همچنین برنامه شما باید حرکات بازیکنان را کنترل کند. یعنی اگر بازیکنی خواست حرکت غیرمجازی (مثلاً حرکت از روی دیوار) انجام دهد، جلوی این حرکت را بگیرد. برنامه شما باید در هر لحظه، وضعیت بازی را به شکل مناسبی نمایش دهد. یعنی وضعیت صفحه و مهره ها و ستاره ها و دیوارها، اینکه نوبت حرکت کدام مهره است، و همچنین امتیاز هر مهره را نمایش دهد. همچنین وضعیت سرعت گیرهایی که بر حرکت هر مهره باید اعمال شوند (توسط حریفش جمع آوری شده اند) باید نمایش داده شود.

نکات قابل توجه

برنامه شما باید از واسط گرافیکی مناسب برخوردار باشد و برای این منظور باید از چارچوب swing استفاده کنید.

پروژه را می توانید انفرادی و یا در گروه های حداکثر سه نفره انجام دهید. توصیه می شود پروژه را بصورت گروهی انجام دهید.

رعایت نکات برنامه نویسی شیء گرا در پیاده سازی پروژه ضروری می باشد.

رعایت قواعد رایج کدنویسی جاوا ضروری می باشد.

تلاش کنید تا آنجا که می توانید، از مزایایی که برای برنامه نویسی شیء گرا مطرح نموده ایم، استفاده نمایید.

راهنمایی

توصیه می شود انجام این پروژه را در قالب مراحل زیر انجام دهید.

- مرحله اول: تحلیل و طراحی برنامه از دیدگاه شیء گرا: ابتدا سعی کنید مفاهیم موجود در برنامه و ارتباط آنها با یکدیگر را مشخص کنید. چه اشیایی در برنامه وجود دارند. این اشیاء متعلق به چه کلاسهایی هستند. این اشیاء دارای چه خصیصه (فیلد) ها و چه رفتار و چه امکاناتی (متدها) می باشند؟ کدام اشیاء با کدام اشیاء دیگر در ارتباط و تماس هستند؟
- مرحله دوم: کلاسهای لازم را پیاده سازی نمایید. نیاز نیست این کلاسها را در یک گام بطور کامل پیاده سازی نمایید. می توانید آنها را به تدریج کامل کنید. در هر مرحله که قسمتی از کلاسها را پیاده سازی می کنید، یک برنامه نمونه بنویسید و امکاناتی را که تا آن موقع پیاده سازی کرده اید تست کنید. در این مرحله، بحث گرافیک برنامه را کنار بگذارید و بر روی عملکرد منطقی برنامه تمرکز نمایید. برای نمایش خروجی ها از چاپ مقادیر در خروجی استفاده نمایید. مثلاً می توانید برای حرکت دادن مهره ها، مکان مهره مقصد را از ورودی بصورت متنی بخوانید و وضعیت بازی را هم بصورت متنی در خروجی چاپ کنید.
- مرحله سوم: ایجاد برنامه اصلی که دارای واسط کاربری و امکانات گرافیکی مورد نیاز می باشد. کلاسهایی را که در مرحله قبل ایجاد و تست نموده اید، در قالب یک برنامه قرار داده و با افزودن امکانات واسط گرافیکی و مسائل مربوط به آن، پیاده سازی را کامل نمایید.

موفق باشید.

پایدار