

فاز سوم پروژه

درس برنامه نویسی شیء‌گرا

موعد ارسال: چهارشنبه ۲۱ تیر ۱۳۹۱



فاز سوم پروژه برنامه سازی شیء‌گرا

مهلت تحویل: چهارشنبه ۲۱ تیر ۱۳۹۱

نکات کلی

- برای ارسال به آدرس زیر مراجعه کنید:

<http://arsh.co/java902/>^۱

- موعد تحویل ساعت ۱۶:۵۹ روز چهارشنبه ۲۱ تیر ۱۳۹۱ بوده و به هیچ عنوان تمدید نخواهد شد.
- تأخیر تا ۱ ساعت، به ازای هر ۳ دقیقه موجب کسر شدن ۱٪ نمره، از ۱ ساعت تا ۲۴ ساعت موجب کسر شدن ۲۰٪ نمره، از ۲۴ ساعت تا ۲۵ ساعت، به ازای هر ۲ دقیقه موجب کسر شدن ۱٪ نمره‌ی دیگر و بین ۲۵ ساعت تا ۴۸ ساعت باعث کسر شدن ۵۰٪ نمره می‌شود. تأخیر بیشتر از ۴۸ ساعت قابل قبول نیست.
- قطع بودن اینترنت خوابگاه، کندی پایگاه اینترنتی دانشکده، تمام شدن کارت اینترنت و ... هیچ‌کدام دلیل موجهی برای تأخیر نیستند.
- سوالات خود را در سایت piazza و با تگ مناسب مطرح کنید.
- نحوه تحویل حضوری متعاقباً اعلام خواهد شد.



مقدمه

سلام،

هدف فاز سوم ایجاد امکان اجرای بازی در شبکه و بر روی رایانه‌های مجزا به همراه اضافه کردن دیگر قابلیت‌های مفید است.

مدل کلی بازی در شبکه

در مدل اصلی باید بتوانیم بازی را برای دو بازیکن و بر روی دو رایانه که به وسیله شبکه به هم متصل شده اند اجرا کنیم.

هر بازیکن با یکی از این دو رایانه تعامل دارد و بازی را از دید خودش می‌بیند.

تمامی پیغام‌ها و وقایع و روند بازی به صورت یکسان توسط دو بازیکن به صورت همزمان دیده می‌شود به جز موارد زیر:

- پیغام دیده شدن گنج یک خانه توسط بازیکنی که نوبت اوست دیده می‌شود (پیغامی که به بازیکن دیگر نشان داده می‌شود صرفاً نشان می‌دهد که این بازیکن گنج این خانه را دیده است).
- انتخاب‌ها توسط بازیکنی که نوبت اوست انجام می‌گیرد (دقت کنید که بازیکنی که نوبت او نیست نیز انتخاب‌ها را می‌بیند ولی نمیتواند انتخاب را انجام دهد).

نحوه ارتباط رایانه‌ها

برای ارتباط بین اجراها از پروتکل TCP/IP استفاده کنید. جزئیات آن دست خودتان است ولی خوب است از پروتکلی استفاده کنید که پیاده سازی آن توسط دیگران (و احیاناً در زبانهای برنامه نویسی دیگر) ساده باشد. پروتکل استفاده شده را به صورت خلاصه مستند کنید و در کنار منابع پروژه قرار دهید.

فرض بر این است که هر دو اجرای بازی روی نقشه اصلی (آنچه در فاز ۱ و ۲ استفاده کرده‌اید) است.

یکی از اجراها نقش سرور را برعهده دارد و اجرای دیگر به عنوان کلاینت شناخته می‌شود. فقط سرور است که گنج تمام خانه‌ها را می‌داند و این اطلاعات را فقط در صورت درخواست کلاینت و در مواقع لازم در اختیار او قرار می‌دهد.

می‌توانید فرض کنید که نه کلاینت و نه سرور در کلیت پروتکل ارتباطی سعی در خرابکاری ندارند ولی ممکن است کلاینت بخواهد اطلاعات غیر لازم از سرور درخواست کند و یا هر یک از دو اجرا بخواهند حرکات ناممکن انجام

دهند(مثلاً شماره تاس‌ها را ۷ و ۸ اعلام کنند و یا یک بازیکن بخواهد به طرز عجیبی به یک خانه بپرد یا گنج یک خانه توسط سرور دو چیز متفاوت اعلام شود).

تمامی ارتباطات و درخواست‌ها و حرکات باید به نحوی توسط طرفین بررسی شوند و در صورت تشخیص خرابکاری طرف دیگر با پیغامی متوقف شوند.

روند اجرا

دو نوع اجرای گرافیکی و تحت کنسول می‌بایست توسط برنامه شما پشتیبانی شوند.

اجرای گرافیکی

در ابتدا با انتخاب گزینه، کلاینت یا سرور بودن هر اجرا توسط کاربر مشخص می‌شود. سپس شماره پورت برقراری ارتباط به سرور و پس از آن IP سرور و شماره پورت ارتباط به کلاینت داده می‌شود. سرور در این مدت منتظر برقراری ارتباط توسط کلاینت بوده است.

پس از برقراری ارتباط هر بازیکن نام خودش را وارد می‌کند و بازی توسط بازیکن سرور شروع می‌شود.

روند بازی همانند آنچه در فازهای قبلی گفته شده و در محیط گرافیکی انجام می‌پذیرد. همانطور که گفته شد روند بازی باید توسط دو بازیکن به صورت یکسان دیده شود.

اجرای تحت کنسول

گزینه‌ی کلاینت یا سرور بودن، نام بازیکن و IP سرور و شماره پورت به صورت آرگومان به برنامه شما داده می‌شود. بازی توسط بازیکن سرور شروع می‌شود.

روند بازی همانند آنچه در فازهای اول گفته شده و تحت کنسول انجام می‌پذیرد. همانطور که گفته شد روند بازی باید توسط دو بازیکن به صورت یکسان دیده شود ولی هر انتخاب توسط بازیکنی که نوبت اوست انجام می‌پذیرد.

دقت کنید که ممکن است یکی از بازیکنان به صورت گرافیکی و دیگری تحت کنسول اجرا شود.

قابلیت صحبت بین بازیکنان

بازیکنان باید بتوانند در حین بازی با یکدیگر صحبت کنند. این صحبت کردن به صورت متنی و شبیه چت است. این قابلیت را در صفحه‌ی اصلی بازی و به صورت مینیمال قرار دهید.

امکان مشاهده تاریخچه بازی

در حین بازی بازیکن باید بتواند در تاریخچه حالت بازی حرکت کند بدین صورت که با عقب و جلو شدن بتواند وضعیت و مکان مهره ها و امتیاز بازیکنان و گنج هدف در گذشته بازی را بر روی صفحه‌ی اصلی دوباره مرور کند (با تغییر وضعیت صفحه‌ی اصلی به گذشته و تاکید بر اینکه دیگر حالت فعلی نمایش داده نشده است). در این حالت وضعیت فعلی بازی تغییر نمی‌کند و بازیکن می‌بایست بازی را از همان وضعیت فعلی ادامه دهد.

این قابلیت را با دو دکمه عقب و جلو (و بازگشت به حالت فعلی) در صفحه‌ی اصلی بازی پیاده سازی کنید.

نمره بندی و نمره اضافی

قسمت اصلی نمره به پیاده سازی مناسب و ساختار درست و کارا و همچنین اجرای درست تعلق دارد.

۱۰ درصد نمره به امکان انتقال نقشه از سرور به کلاینت اختصاص دارد (در این حالت برنامه شما باید قابلیت خواندن نقشه از فایل را داشته باشد).

۳۵ درصد نمره به پیاده سازی مستقل سرور و ایجاد امکان بازی بیش از دو نفر تعلق دارد. در این حالت هیچ یک از دو اجرای بازیکنان سرور نیست و اطلاعات مربوط به گنج‌ها در اختیار اجرای سومی به عنوان سرور است و هماهنگی روند کلی بازی نیز توسط این سرور انجام می‌گیرد. در این حالت باید امکان اتصال بیش از دو کلاینت به عنوان بازیکن نیز وجود داشته باشد. همچنین در این حالت باید سرور بتواند چند بازی مختلف را به صورت همزمان اجرا کند. سوالات مربوط به این بخش را در پیازا با تگ server بفرستید.

۱۰ درصد نمره به هماهنگ بودن پروتکل‌های ارتباطی دو یا چند نفر از دانشجویان تعلق می‌گیرد. در این حالت اجرای کلاینت یکی از دانشجویان با سرور دیگری امکان‌پذیر است (به نظر برخی از دستیاران استاد چقدر خوب می‌شود که پروتکل سرور مجزا مشخص شود و همه دانشجویان از آن پیروی کنند و کلاینت و سرور هماهنگ بنویسند). پیش از شروع این مورد را با دستیاران استاد هماهنگ کنید.

۱۰ درصد نمره به امکان ذخیره تاریخچه بازی در انتهای بازی در فایل و پخش آن اختصاص دارد.

۵۰ درصد نمره به پیاده سازی بازی و محیط گرافیکی آن بر روی پلتفرم‌های موبایل (اندروید، iOS و ...) تعلق می‌گیرد. پیش از شروع این مورد را با دستیاران استاد هماهنگ کنید.

نوشتن بازیکن هوشمند فوق‌العاده، کماکان نمره‌ی اضافی دارد.

در صورتی که ایده‌ای برای نمره اضافی دارید می‌توانید آن را با دستیاران استاد در میان بگذارید.

حداکثر نمره‌ی قابل کسب از نمرات اصلی و اضافی (پس از تحویل پروژه و مشخص شدن نمرات) ۱۷۰ درصد است.

نحوه فرستادن کد

کد خود را در قالب یک فایل zip بفرستید. ریشه این فایل zip باید حاوی یک فولدر با نام razeJangal باشد که معادل package ریشه برنامه با همین نام است. این فایل zip حاوی فایل‌های متنی java. (و نه class. های کامپایل شده) است. نام فایل zip را JAVA902P3.zip بگذارید.

مستند مربوط به پروتکل و احياناً دیگر مستندات را در درون فولدر doc بیرون از تمام پکیج‌ها قرار دهید.

متد Main در این فاز باید در کلاس razeJangal.multi.MultiRunner باشد.

به عنوان آرگومان اول به برنامه شما یکی از رشته‌های "console" یا "gui" به ترتیب نشان دهنده‌ی اجرای تحت کنسول و گرافیکی و سپس نام بازیکن داده می‌شود.

در صورت اجرای تحت کنسول پس از این دو آرگومان در ابتدا یکی از رشته‌های "server" و یا "client" به ترتیب نشان‌دهنده‌ی سرور یا کلاینت بودن داده می‌شود. در صورت کلاینت بودن در ادامه IP و شماره پورت سرور و در صورت سرور بودن تنها شماره پورت داده می‌شود.

در صورت پیاده سازی بخش‌های اضافی به نحوی مناسب امکان تنظیمات مربوط را فراهم کنید.

نمونه نحوه اجرا:

```
java razeJangal.multi.MultiRunner gui
```

```
java razeJangal.multi.MultiRunner console Packman server 3500
```

```
java razeJangal.multi.MultiRunner console Jackman client 192.168.1.10 3500
```

پیش از ارسال مطمئن شوید که کد شما به درستی کامپایل و اجرا می‌شود.

موفق باشید