

فاز اول پروژه برنامه‌سازی پیشرفته

بازی ابتکار عمل و حافظه

بازی ابتکار عمل و حافظه

ارزجنگ

«جستجویی در سرزمین داستانهای افسانه‌ای.»

«جستجویی در سرزمین داستانهای افسانه‌ای.»

نکات کلی

- موعده تحویل ساعت تا ساعت ۱۶:۰۰ روز سه شنبه ۲۰ تیر ۱۳۹۶ است.
- تأخیر به ازای هر یک دقیقه باعث کسر ۰/۱٪ از نمره‌ی شما خواهد شد.
- سوالات خود را در پیاذا و با تگ مناسب مطرح کنید.



مقدمه

پروژه‌ی نهایی این درس، پیاده‌سازی یک بازی بسیار جالب به اسم «راز جنگل» است. این بازی روی یک زمین با تعدادی خانه در یک نمودار گراف شکل انجام می‌شود. تعداد بازیکنان بازی دو تا چهار نفر است که هر کدام یک مهره در زمین دارند که مکان آنها را در زمین نشان می‌دهد. بازی توسط دو عدد تاس انجام می‌شود. در فاز اول، شما این بازی را برای اجرا در کنسول پیاده‌سازی می‌کنید و همچنین به صورت امتیازی می‌توانید آن را برای محیط شبکه نیز پیاده کنید. در فاز بعدی محیط گرافیکی را به آن اضافه خواهید کرد. فاز اول به صورت فردی و فاز دوم به صورت گروه‌های دو نفری خواهد بود.

قوانین بازی

زمین بازی

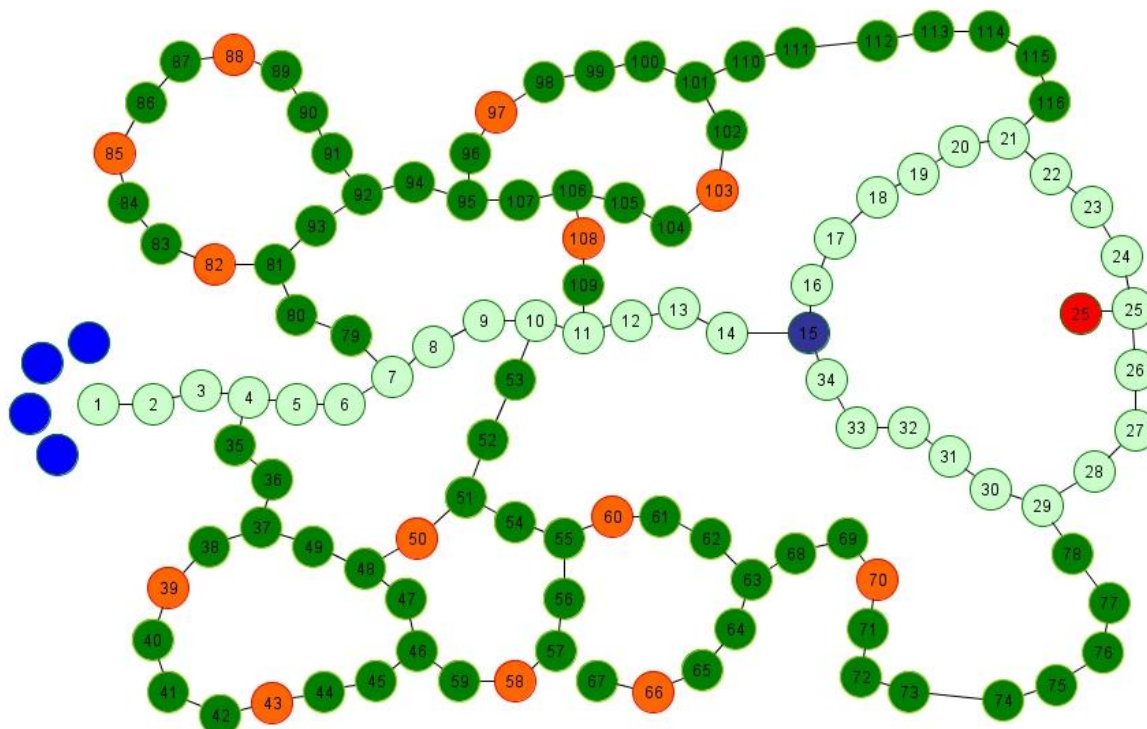
زمین بازی یک نمودار گراف شکل است که در شکل ۱ نشان داده شده است. خانه‌های سبز رنگ خانه‌های عادی هستند. در زیر هر یک از سیزده خانه‌ی نارنجی رنگ یک گنج پنهان شده است (در این فاز نام این گنج‌ها را حرف‌های A تا M در نظر بگیرید). خانه‌های قرمز و بنفش نیز توضیح داده می‌شوند.

مراحل بازی

بازی در سیزده مرحله انجام می‌شود. در هر مرحله یکی از گنج‌های پیدا نشده به صورت تصادفی مشخص می‌شود و بازیکنان باید سعی کنند به آن دست پیدا کنند. هر مرحله با دستیابی یکی از بازیکنان به گنج آن مرحله پایان می‌یابد.

آغاز بازی

در ابتدای بازی گنج‌ها به صورت تصادفی در خانه‌های نارنجی پخش می‌شوند. بازیکنان نیز در ابتدا در خانه‌های آبی رنگ قرار دارند.



شکل ۱

فرایند بازی

نوبت بازی بر اساس ترتیب بازیکنان بین آنها می‌چرخد و در هر نوبت، دو تاس هم‌زمان انداخته می‌شود و با توجه به اعداد آمده بازیکن مهره‌اش را حرکت می‌دهد. حرکت مربوط به هر تاس مستقل انجام می‌شود و در واقع در هر نوبت هر بازیکن دو حرکت انجام می‌دهد. ترتیب انتخاب اعداد دو تاس در یک نوبت به دلخواه بازیکن است. واضح است که جهت هر حرکت باید مشخص باشد و در یک حرکت نمی‌توان روی یک خانه چند بار رفت ولی جهت دو حرکت یک نوبت می‌توانید دو چیز مختلف باشد (برای مثال ممکن است حرکت اول رو به جلو و حرکت بعدی رو به عقب باشد).

در صورتی که مهره یک بازیکن در انتهای یک حرکت، در یکی از خانه‌های نارنجی رنگ قرار بگیرد، او می‌تواند نام گنج آن خانه را ببیند (در بازی واقعی نام گنج فقط به همان بازیکن نشان داده می‌شود ولی مشخصاً در کنسول نمی‌توان این قانون را به درستی اعمال کرد).

در صورتی که مهره یک بازیکن در انتهای یک حرکت، در خانه قرمز قرار بگیرد، او می‌تواند چنانچه بخواهد نظر خود را در مورد محل گنج آن مرحله را اعلام کند.

در صورت درست بودن محل گنج اعلام شده، او گنج را به دست آورده و مرحله فعلی پایان می‌پذیرد و نوبت اول مرحله‌ی بعدی نیز به عنوان جایزه به همان بازیکن تعلق می‌گیرد. در این نوبت بازیکن می‌تواند محل گنج مرحله جدید را اعلام کند و یا با انداختن تاس به بازی ادامه بدهد.

در صورت اشتباه بودن نیز، مهره بازیکن به خانه‌های آبی رنگ بر می‌گردد.

در صورتی که مهره‌ی بازیکن در حرکت تاس اول به خانه قرمز رسیده بود و او بخواهد محل گنج را بگوید حرکت تاس دوم او از دستش می‌رود.

در صورتی که مهره یک بازیکن در انتهای یک حرکت، بر روی مهره‌ی بازیکن دیگری قرار بگیرد، جای آن را گرفته و مهره‌ی بازیکن دوم به خانه‌های آبی منتقل می‌شود.

در صورتی که در یک نوبت، اعداد دو تاس انداخته شده مشابه باشند، بازیکن می‌تواند در صورتی که بخواهد به جای حرکت دادن مهره‌ها، یکی از کارهای زیر را انجام دهد:

- مهره‌ی خود را به یکی از خانه‌های نارنجی خالی منتقل کند و نام گنج آن را ببیند. (به شرط اینکه در آن خانه‌ی نارنجی مهره‌ای نباشد)
- مهره خود را به خانه بنفش منتقل کند. (به شرط اینکه در آنجا مهره‌ای نباشد)
- گنج هدف مرحله فعلی را تغییر دهد تا گنج جدیدی به صورت تصادفی مشخص شود.

با پایان یافتن سیزده مرحله، بازیکنی که تعداد گنج‌های بیشتری را به دست آورده است برنده می‌شود.

پایان بازی

ورودی و خروجی

نکته: از آنجایی که بخشی از نمره شما با تصحیح به شکل خودکار مشخص می‌شود، لازم است در پیاده‌سازی ورودی و خروجی دقت زیادی کنید.

در ابتدا برنامه با پیام زیر تعداد بازیکنان را درخواست می‌کند:

```
How many players are going to play?
```

تعداد بازیکنان به عنوان ورودی به برنامه داده خواهد شد.

در ابتدای هر مرحله و یا با تغییر یافتن گنج هدف مرحله و پیش از شروع نوبت بازیکن، گنج هدف با پیامی به یکی از صورت‌های زیر در خروجی مشخص می‌شود:

```
Round number 5 started, this round's goal treasure is E
```

```
Round number 5, this round's goal treasure has changed to F
```

در ابتدای نوبت هر بازیکن، اعداد تاس‌های او به همراه انتخاب‌های او در خروجی مشخص می‌شوند. به همراه این اطلاعات، وضعیت بازی شامل موقعیت بازیکنان نیز مشخص می‌شود:

```
Current Positions: 1->22 2->57 3->91
```

```
Dice Numbers for Player 2: 3 5
```

```
Player 2's Choices:
```

1. Move to 54, and then move 5 cells
2. Move to 60 (Orange), and then move 5 cells
- ...

دقت کنید که انتخاب‌های کاملاً یکسان فقط یک بار باید نشان داده شوند.

بازیکن یکی از انتخاب‌ها را با دادن شماره آن مشخص می‌کند. حرکت دوم نیز به صورت زیر مشخص می‌شود:

```
Player 2's Choices (Second Dice):
```

1. Move to 10
2. Move to 66(Purple)
- ...

در صورت مشابه بودن اعداد تاس‌ها، انتخاب‌های زیر نیز به انتخاب‌های تاس اول اضافه می‌شوند:

5. Move to 85(Orange)
6. Move to 88(Orange)
- ...
19. Move to 15(Violet)
20. Change current round's goal treasure

بازیکن با وارد کردن شماره انتخاب در ورودی، انتخابش را مشخص می‌کند.
در صورتی که در پایان یک حرکت مهره بازیکن در خانه‌ی نارنجی قرار بگیرد، پیام زیر در خروجی گنج آن خانه را به او نشان می‌دهد (در کنسول البته همه این را می‌بینند):

```
Cell 39's treasure as seen by player 1 is E
```

در صورتی که در پایان یک حرکت مهره بازیکن در خانه قرمز قرار بگیرد، پیام زیر در خروجی از او می‌خواهد که محل گنج را در صورت تمایل مشخص کند:

```
Player 2 is in 25(Red), and can attempt to guess goal  
treasure's place:
```

1. I don't want to guess
2. It's 39
3. It's 43
- ...

در صورت درست بودن و به دست آوردن گنج، پیام زیر باید در خروجی این موضوع را مشخص کند:

```
Player 2's guess, 39, was correct and he/she has won this  
round's goal treasure, E
```

در غیر این صورت باید پیام زیر در خروجی درست نبودن را مشخص کند:

```
Player 2's guess, 39, was not correct
```

(پس از به دست آوردن گنج به بازیکن مجدداً اجازه داده می‌شود تا در صورت تمایل محل گنج جدید را مشخص کند. در صورتی که با انتخاب گزینه I don't want to guess عدم تمایل خود را اعلام کند، بازی با تاس انداختن در نوبت خودش ادامه پیدا می‌کند)

در صورتی که به هر دلیلی یک بازیکن به خانه‌های آبی منتقل شود لازم است که پیام زیر در خروجی چاپ شود:

```
Player 2 is moved to Blue cells.
```

به زودی یک نمونه از ورودی و خروجی در اختیارتان قرار داده خواهد شد.

نحوه پیاده سازی

در این بخش پیشنهاد کلی برای ساختار و معماری شیء گرای پیاده سازی ارائه می شود.

تعریف کلاس های زیر با توجه به بازی درست و منطقی به نظر می رسد:

BoardCell برای هر یک از خانه های زمین یک نمونه از این کلاس می تواند ساخته شود. (دقت کنید که خانه های زمین انواع مختلفی دارند که خوب است با وراثت مدل شوند).

Board زمین بازی که متشکل از مهره های بازیکنان و خانه های زمین و تاس ها است.

Player برای هر بازیکن یک نمونه از این کلاس وجود دارد که محل مهره و دیگر خواص آن بازیکن را مشخص می کند.

DicePool این کلاس فرایند انداختن تاس را انجام می دهد.

Game در این کلاس شروع و فرایند بازی پیاده سازی می شوند. در واقع این کلاس قوانین بازی مدنظر را پیاده سازی می کند. اطلاعات مربوط به وضعیت بازی (به طور مستقیم و یا غیر مستقیم) در اختیار این کلاس هستند.

ConsoleServer این کلاس حاوی پیاده سازی کنسول برنامه است که یک نمونه از **Game** ایجاد می کند و با کمک واسط آن، بازی را در کنسول اجرا می کند. حلقه اصلی برنامه در این کلاس قرار دارد.

از آنجا که لازم است در فازهای بعدی محیط گرافیکی نیز به پروژه اضافه شود، خوب است از همین الان در پیاده سازی کلاس های **Game** و **ConsoleServer**، وظایف آنها و واسط ارتباطی بین آنها دقت کنید.

Game منطق بازی را در بر دارد و تا هنگامی که تابعی از آن صدا زده نشود کاری انجام نمی دهد. ارتباط با بازیکنان از طریق کنسول توسط **ConsoleServer** صورت می گیرد و در نتیجه کنترل روند بازی و اجرای توابع **Game** برای پیشبرد بازی در اختیار **ConsoleServer** است.

برای مثال می توانید توابعی در **Game** در نظر بگیرید که

- بتوان با صدا زدن آنها انتخاب های بازیکنان را به **Game** منتقل کرد و در نتیجه **Game** تغییرات لازم را در وضعیت بازی ایجاد کند
- با صدا زدن آنها بتوان وضعیت بازی (گزینه های بازیکنان، اینکه الان نوبت چه کسی است و ...) را پرسید و از اتفاقات بازی با خبر شد

به عنوان یک مثال دیگر اینکه پیغامی که باید در کنسول نمایش داده شود به جای اینکه در کلاس ConsoleServer قرار داشته باشد در کلاس Game قرار داده شود، نشانه‌ی این است که وظایف به درستی بین این دو کلاس تقسیم نشده‌اند.

ایده‌آل این است که اضافه کردن محیط گرافیکی نیاز به تغییر در کلاس Game را به همراه نداشته باشد.

نمره بندی فاز اول و نمره اضافی

۶۰ درصد نمره به درستی برنامه‌ی شما و ۴۰ درصد آن به بررسی‌های کد در تحویل حضوری اختصاص دارد.

در تحویل حضوری موارد زیر بررسی می‌شوند:

۱. تسلط بر کد
۲. استفاده از مفاهیم شیء‌گرا و درستی انتخاب اشیاء و کلاسها و تعریف واسطها
۳. نحوه‌ی کد زدن درست (دراز نبودن بیش از حد توابع، نام‌گذاری صحیح و ...)
۴. رعایت مواردی مانند Visibility در اعضای کلاسها
۵. استفاده درست از ساختار فایلها

به موارد زیر نمره اضافی تعلق می‌گیرد:

۱. نوشتن کامنت‌های مناسب (تا حداکثر ۱۰ درصد نمره)
۲. تمیز و خوانا بودن کد در سطح فوق‌العاده (تا حداکثر ۱۰ درصد نمره)
۳. نوشتن یک بازیکن نمونه (تا حداکثر ۱۰ درصد نمره)
۴. قابلیت خواندن گراف نقشه زمین از فایل (تا حداکثر ۱۰ درصد نمره)

نحوه فرستادن کد

کد خود را در قالب یک فایل zip بفرستید. ریشه این فایل zip باید حاوی یک فولدر با نام razeJangal باشد و فایل‌های متنی کد شما در این فولدر باشند. نام فایل zip را StudentID.zip بگذارید که StudentID را با شماره دانشجویی خود جایگزین کرده‌اید؛ مانند: 95105211.zip

کد شما بایستی توسط make و یا cmake در محیط msys2/mingw و یا linux قابل کامپایل باشد و فایل اجرایی raze_jangal را ایجاد کند. (دقت کنید که CLion خود از cmake استفاده می‌کند و کار اضافی لازم نیست انجام بدهید)

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

پیاده‌سازی تحت شبکه (امتیازی)

در پیاده‌سازی تحت شبکه می‌خواهیم این امکان را اضافه کنیم که بازیکنان بتوانند از طریق شبکه (برای مثال از دو کامپیوتر مستقل دیگر) به بازی متصل شوند و بازی را از دید خود ببینند.

در این حالت بازی یک سرور محسوب می‌شود و بازیکنان، هر یک، یک کلاینت محسوب می‌شوند. تمامی پیغام‌ها و وقایع و روند بازی به صورت یکسان توسط بازیکن‌ها 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).

همچنین در ابتدا تعداد بازیکنان از اولین بازیکنی که متصل می‌شود پرسیده می‌شود و سپس سرور صبر می‌کند تا همه متصل شوند و سپس بازی را شروع می‌کند.

برای ارتباط بین اجراها از پروتکل TCP/IP استفاده کنید. بر بستر این پروتکل کافی است پیغام‌های متنی بین بازیکنان و سرور بازی مانند آنچه در حالت کنسول گفتیم رد و بدل شود (با دو محدودیتی که بالاتر گفته شد)

موفق باشید