

پروژه پایانترم درس برنامه نویسی پیشرفته

مقدمه:

این احتمالاً یکی از **جذابترین** پروژه های دوره کارشناسی شماست...

امیدوارم ازش لذت ببرین ;)

نبرد قهرمانان:

شاید از اسم پروژه فهمیده باشید که داریم از یک بازی احتمالاً استراتژیک صحبت میکنیم. احتمالاً این نوع بازی هارا امتحان کرده اید. ولی اینبار وظیفه شما کمی متفاوت تر از بازی کردن است. **اینبار شما باید زمین بازی را پیاده سازی کنید.**

درواقع شما باید قوانین بازی را پیاده سازی کنید؛ هرکدام از قهرمانان بازی را بسازید و سپس بازی را شروع کنید. بگذارید کمی بیشتر توضیح بدم:

در یک بازی که شامل دو تیم است، می توان حداقل سه بخش را متصور شد:

(۱) استراتژی ها و تعداد و نوع قهرمانان(Hero) های تیم اول

(۲) زمین بازی و قوانین بازی

(۳) استراتژی ها و تعداد و نوع قهرمانان(Hero) های تیم دوم

قبل از شروع بازی هایی که شما قبلاً بازی کرده اید، برای نبرد با تیم روبهرو یک سری استراتژی ها را در ذهنتان قرار می دهید و طبق آن عمل می کنید. حال اینبار شما باید قواعد زمین بازی پیاده سازی کنید.

نکته مهم: برای ساده سازی در نظر میگیریم حرکات هردو تیم صرفاً به سمت قلعه های حریف است و در مسیر هیچ مانعی وجود ندارد(اگر متوجه ساختار بازی نشدید نگران نباشد، در ادامه اجزا بازی را بررسی می کنیم)

اجزا بازی:

(۱) در این بازی یک زمین بازی $m \times n$ داریم که شامل درواقع m, n همان طول و عرض های زمین بازی شما هستند.

(۲) هر تیم یک قلعه دارد که باید در ابتدای زمین خودش باشد(در سطر اول زمین اش باشد). در واقع اگر زمین بازی 10 در 10 باشد باید قلعه ما در سطر اول یا همان `[arr[0][i]` باشد و **i** میتواند از ستون اول تا ستون آخر باشد.

۳) برای کلاس قلعه یک شاخص Health تعریف شود که نشان دهنده میزان خرابی قلعه می باشد و در صورتی که Health قلعه هر تیم تمام گردد آن تیم شکست می خورد.

۴) در این بازی چهار نوع هیرو داریم که همه آنها ویژگی های ثابتی دارند. در پایین توضیحات کامل آن ها داده خواهد شد.
برای هر هیرو سه نوع توانایی (ability) تعریف می گردد:

1) قدرت

2) سلامتی

3) سرعت (میتواند یک یا دو باشد و اگر دو باشد در صورتی که بتواند هر دو خانه را در یک پرش طی می کند مثلا از خانه ۱ و ۳ به خانه ۳ و ۳ میبرد)

۵) برای هر چهار نوع هیرو باید این سه ویژگی را بطوری که تعادل برقرار گردد تقسیم نمایید (مسلما تقسیم منطقی مقدار هر توانایی نمره دارد)
بطور مثال داریم:

```
class first_hero:  
    power = 10  
    speed = 1  
    health = 30  
    x = 1  
    y = 3
```

بطور مثال هیرو اول قدرت ده دارد و بدین معنی است که به ازای هر ضربه ای که میزند حریفش جان حریفش منهای ده می گردد یا اگر به قلعه هم بزند به همان میزان از health قلعه کم می شود.
دو متغیر x , y هم مختصات هر ابجکت بر صفحه مختصات است که براساس حرکت هیرو قطعا اپدیت می گردد.

۶) حرکت هیرو ها همواره به سمت قلعه می باشد و اینکار برای ساده سازی پیاده سازی شما انجام می گردد. در واقع با توجه به مختصات قلعه مثال زیر را میزنیم:

اگر مختصات قلعه x0, y0 باشد و مختصات هیرو x1, y1 باشد باید همواره چک کند که آیا فاصله اش در راستای محور y با قلعه طولانی تر است یا در راستای محور x و در راستای هر کدام که بزرگ تر است حرکت کند.

۷) نبرد دو هیرو: زمانی دو قهرمان به مبارزه می پردازند که در یک خانه قرار بگیرند و بدین صورت مقدار Health هر هیرو بصورت زیر محاسبه می گردد:

```
hero_first_team.health -= hero_second_team.power  
hero_second_team.health -= hero_first_team.power
```

هر کدام از هیروهایی که زودتر کشته شود از بازی حذف می گردد و هیرو دیگر با health باقی مانده با حرکت ادامه می دهد.

نکته مهم: نبردهای هیرو ها باید بصورت موازی باشد و حرکت هر چهار نوع هیرو باید به صورت موازی انجام شود.
(برای کسانی که هنوز متوجه نشدند در واقع باید با استفاده از ترد ها و برنامه نویسی موازی این کار را انجام دهید.)

هیرو ها و شی گرایی :

همان طور که گفته شد هر هیرو دارای سه ویژگی قدرت ، سلامتی و سرعت میباشد.

هیرو های بازی چهار دسته خواهند شد. (یخ ، آتش ، باد ، خاک) هر دسته ویژگی های خاص خود را دارد برای مثال گروه یخ دارای سلامتی زیاد و قدرت و سرعت کم ، گروه آتش قدرت زیاد ، گروه باد سرعت زیاد و گروه خاک دارای هر سه ویژگی بصورت متعادل را دارد.

با توجه به ویژگی های گفته شده کلاس ها و نحوه اثربری را باید تشخیص دهید و سپس برای هر کدام از گروه ها به دلخواه و با توجه به منطق و خلاقیت خود ، هیروهای مورد نظر خود را تشکیل دهید. (هیچ توضیح اضافه دیگری نیاز نیست و تمامی نکات متعادل سازی و چگونگی مقدار متغیر ها و نام و شکل هیرو ها بر عهده شماست.)
(یکی از ایده هایی که برای دانشجویان فعال تر میتوان داشت ، داشتن توانایی ویژه برای هر هیروی بصورت خاص است که بعد از چند ضربه میتواند از آن استفاده کند.)

جدول امتیازات و ذخیره سازی:

امتیازات بازیکنان و هیرو ها و تاریخچه هر بازی را در فایل ذخیره کرده و پس از شروع مجدد برنامه آن ها را لود کنید.

خب تا اینجا پروژه تقریبا همیشه گفت خیلی آسون بوده و همش رو میشه توی یک روز پیاده کرد اما صبر کنید بخش جذاب ماجرا مونده...

گرافیک پروژه:

باید هر هیرو با یک المان نشان داده شود و موقعیت هر کدام از هیروها هم در آن صفحه نمایش داده شود.
در واقع باید ابتدا با استفاده از فریم های گرافیکی زمین بازی را بصورت گرافیکی طراحی کرده و برج های دو طرف بازی را نیز بصورت اشیا ساده (در حد یک مربع رنگی) نمایش دهید. سپس هیرو ها را نیز بصورت اشیا ساده (دایره های رنگی طبق رنگ بندی منطقی) در دو طرف زمین جایگزاری کرده و حرکت های آن ها را بصورت لحظه ای نشان دهید.

دکمه های شروع نبرد ، خروج از بازی و ذخیره بازی و امتیازات نیز در بالای برنامه باید مشهود باشد.

برای اطلاعات بیشتر درمورد حرکت اشیا روی فریم های گرافیکی به آموزش قدم به قدم زیر مراجعه کنید بطور خاص بخش 4.2 حرکت یک شی با استفاده از دکمه ها را برای شما توضیح میدهد اما دقت کنید که هیرو های شما نیازی به دکمه حرکتی ندارند و پس از شروع بازی خودشان حرکت میکنند.

احتمالا جلسه توضیحات مفصل تر این روند گرافیکی پس چند روز پس از شروع پروژه توسط تی ای ها برگزار خواهد شد.

اگر در سایر بخش های پروژه ابهام یا توضیحات کمی داده شده طبق فرضیات خود و به ساده ترین روشی که به ذهنتان میرسد جلو بروید و پروژه را برای خود سخت نکنید!

بخش های اضافه :

- طراحی زیبای فضای بازی ، برج ها و هیرو ها ، (حتی استفاده از تمپلیت های آماده گرافیکی برای هر هیرو که در گوگل موجود است)
- گرفتن لحظه ای اطلاعات برج ها و هیرو های حریف از فایل مخصوص و جایگزاری آن برای هر بازی جدید (در واقع بجای اینکه زمین حریف را خودتان بسازید ، رویه ای طراحی کنید که پروژه بتواند بصورت داینامیک از فایل اطلاعات را بخواند و در واقع هر بازی با بازی قبلی تفاوت داشته باشد).
- پیاده سازی بازی بصورت دونفره واقعی با استفاده از سوکت پروگرامینگ
- پیاده سازی پروژه روی بستر گیت هاب
- سایر ویژگی ها و قابلیت های اضافه که توسط ذهن خلاق شما به پروژه اضافه شود.