

باسمه تعالی

پروژه پایان ترم درس برنامه نویسی پیشرفته

ویرایش ۱

**مقدمه:** پروژه پایانی درس، طراحی و پیاده‌سازی بازی منچ می‌باشد. همانطور که می‌دانید این بازی چهار نفره است و هر بازیکن ۴ مهره دارد. در پیاده‌سازی این بازی کاربر یک نفر از بازیکن‌هاست و سه بازیگر دیگر توسط برنامه شبیه‌سازی می‌شوند.

اگر قوانین بازی را نمی‌دانید به لینک زیر مراجعه کنید.

<http://fa.wikipedia.org/wiki/%D9%85%D9%86%DA%86>

**طراحی:** برای طراحی یک برنامه به زبان شی گرا باید کلاس‌های مورد نیاز آن را مشخص کرد. یک قانون ساده برای این کار این است که اسم‌هایی که در صورت مساله ذکر شده اند را به عنوان کاندیدایی برای کلاس‌های برنامه در نظر گرفت. در مرحله‌ی بعد فیلدهای هر کلاس و بعد ارتباط بین کلاس‌ها را باید یافت.

مثلاً برای پروژه منچ صورت مساله به صورت زیر است:

**بازی چهار بازیکن** دارد. هر بازیکن **چهار مهره** دارد. مهره‌ها روی **صفحه** قرار می‌گیرند...

کلمات مشخص شده اسم‌هایی هستند که در صورت مساله بیان شده اند. البته یافتن بعضی از کلاس‌ها به همین راحتی نیست ولی با بالا رفتن تجربه کشف آنها آسانتر می‌شود. (پروژه‌های کتاب تمرین خوبی برای این کار هستند. یکبار دیگر پروژه کتاب را از منظر یافتن کلاس‌ها مرور کنید.)

بعد از شناسایی هر کلاس باید فیلدهای آن را یافت:

مهره **رنگ** دارد، **مکان** دارد،...

مرحله بعد یافتن ارتباط بین آنهاست:

بازیکن مهره دارد...

البته رابطه‌ی بالا در مرحله‌ی شناسایی فیلدها نیز قابل کشف است.

در زمان اعمال روند فوق و یا بعد از آن معیارها و راهکارهای بیان شده در فصل ۶ (refactoring، RDD، cohesion، coupling) و ... را نیز باید در نظر گرفت.

**نحوه‌ی اجرای برنامه:** کاربر با استفاده از منو یا با کلیک بر روی یک دکمه بازی را شروع می‌کند. ابتدا اسم و رنگ مهره‌های کاربر از وی دریافت می‌شود. (برای این منظور از دیالوگ استفاده شود.) صفحه‌ی بازی در حالت اولیه نمایش داده می‌شود. کاربر بازی را با کلیک بر روی دکمه‌ای که متناظر با انداختن تاس است شروع می‌کند. هر بار که بازیکنی تاس را می‌اندازد عدد تاس نمایش داده می‌شود. در ابتدا

اگر ۶ آمد یکی از مهره‌های کاربر روی خانه شروع متناظر با رنگ انتخاب شده نمایش داده می‌شود و کاربر دوباره باید تاس را بیاندازد. مهره به اندازه عدد آمده جابجا می‌شود. بعد از کاربر به ترتیب نوبت به سه بازیکن دیگر است که هر یک اسم و رنگ مشخصی دارند. در زمانی که کاربر بیش از یک مهره در بازی دارد، باید به طریقی مشخص کند کدام مهره باید حرکت کند. این عمل می‌تواند توسط ماوس و با کلیک بر روی مهره مورد نظر انجام شود و یا به روش مناسب دیگری از کاربر دریافت گردد. هر بار که نوبت به یک بازیکن رسید و کار آن انجام شد، مشخصات و فعالیت آن بازیکن در یک سطر فایلی متنی ذخیره شود. این فایل که به آن فایل log (ثبت وقایع) گفته می‌شود، برای اشکالزدایی (debug) بسیار مفید است. اطلاعات ذخیره شده شامل موارد زیر می‌شود:

- مشخصات بازیکن
- عدد تاس
- مهره جدید که وارد بازی شده یا مهره جابجا شده (مکان اولیه و مکان نهایی)
- مهره حذف شده‌ی حریف (در صورت وجود)

بعد از اتمام بازی صفحه‌ای نمایش داده شود که در آن لیست بازیکنان اول (بازیکن اول در این بازی همراه با بازیکن‌های اول در بازی‌های قبلی) به ترتیب نزولی بر اساس تعداد دفعات اول شدن نمایش داده شود. اطلاعات بازیکن‌های اول در بازی‌های قبلی از فایل خوانده شود. البته اطلاعات این فایل باید قبل از نمایش با نتیجه‌ی بازی فعلی به روز گردد.

در هر لحظه از بازی باید بتوان بازی را از ابتدا شروع کرد.

## ویژگی‌های برنامه:

- داشتن رابط گرافیکی
  - نمایش صفحه بازی. صفحه‌ی بازی را می‌توان با دو روش رسم کرد:
    - نمایش عکسی که از یک صفحه‌ی واقعی منچ گرفته شده (یا اسکن شده) است و قرار دادن مهره‌ها (دایره‌های رنگی) بر روی آن
    - ایجاد صفحه‌ی تصویر با رسم دایره‌های سفید و رنگی
  - نمایش عدد تاس در جای مناسبی از صفحه‌ی بازی
  - داشتن منو (شروع بازی، ذخیره، بارگیری، درباره و ...)
- نمایش اسم و تعداد دفعات اول شدن بازیکن‌ها (در این بازی و بازی‌های قبل) به ترتیب نزولی
- کار با فایل
  - ذخیره/بارگیری بازی: بازیکن می‌تواند در هر لحظه که خواست بازی را ذخیره (save) کرده و از بازی خارج شود. بعداً هر زمان که خواست وارد برنامه شده و با بارگیری (load) بازی، آن را ادامه دهد. (راهنمایی: از فایل سریالی استفاده کنید.)
  - ذخیره‌ی فعالیت بازیکنان در هر مرحله در یک سطر از یک فایل متنی.
  - ذخیره نام و تعداد دفعات پیروزی نفرات اول: اسم بازیکنانی که در بازی‌های مختلف نفر اول می‌شوند به همراه دفعاتی که اول شده‌اند در فایل جداگانه‌ای ذخیره شود. هر بار بعد از بازی محتویات فایل بررسی شده اگر نام بازیکن در فایل

بود به تعداد دفعات اول شدنش یک واحد اضافه شود. اگر اسم بازیکن اول در فایل نبود اسمش و عدد یک در جلوی آن ذخیره شود. (از فایل متنی استفاده شود).

- استفاده‌ی مناسب از استثناء
- ظاهر مناسب و سادگی کار با برنامه (user friendly)
  - کاربر تا حد امکان انتخاب کند و نه تایپ.
- رعایت معیارهای cohesion و coupling

### ویژگی‌های اختیاری: (نمره اضافه)

- تعیین سطح سختی بازی توسط کاربر:
  - بازی سه سطح ساده، متوسط و سخت داشته باشد و کاربر بتواند قبل از شروع بازی سطح مورد نظر را انتخاب کند.
    - در سطح ساده سه بازیگر شبیه سازی شده مهره‌های کاربر را حتی المقدور حذف نمی‌کنند.
    - در سطح متوسط جابجایی مهره‌های سه بازیگر شبیه سازی شده به طور تصادفی انجام می‌شود.
    - در سطح سخت سه بازیگر شبیه سازی شده مهره‌های کاربر را حتی المقدور حذف می‌کنند.
- پخش افکت‌های صوتی در زمان جابجایی/حذف مهره، پیروزی یک بازیکن و پایان بازی
  - برای این کار می‌توانید به سادگی از کلاس MusicPlayer پروژه فصل چهارم استفاده کنید. البته فایل‌های صوتی را باید خودتان تولید کرده و یا از جایی پیدا کنید.

موفق باشید.