

به نام خدا



## درس برنامه سازی پیشرفته

فاز اول پروژه

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۹۹-۰۰

---

استاد:

دکتر محمدامین فضلی

مهلت ارسال:

۲۳ اردیبهشت - ساعت ۲۳:۵۹:۵۹

مسئول پروژه:

امیرمهدی نامجو

مسئول فاز اول:

عرشیا اخوان

طراحان فاز اول:

سروش جهانزاد، متین شجاع، امیرصدرا عبدالحی، امیرمهدی کوشی، یاسمین گلزار، امیرحسین هادیان

مسئولین تنظیم مستند:

سروش جهانزاد و پارسا محمدیان

# فهرست

## نکات قابل توجه

۲

## مقدمه

۳

اهداف قابل توجه

۲

کلیات پروژه

۲

## معرفی بازی

۵

## توضیح بخش‌های مختلف پروژه

۵

کاربر، منوها و موارد عمومی

۵

دستورات مرتبط با کاربر و منو

۶

معرفی ساختار کارت‌ها

۱۱

دک و دسته کارت‌ها

۱۱

دک اصلی (Main Deck):

۱۱

فروشگاه

۱۲

گیم پلی بازی

۱۲

هوش مصنوعی

۱۲

حالت توسعه دهنده

۱۲

لاگ نوشتن

۱۲

لیست کارت‌ها

۱۲

کارت‌های هیولا

۱۲

کارت‌های جادو

۱۲

کارت‌های تله

۱۲



## نکات قابل توجه

- پس از اتمام این فاز، در گیت خود یک تگ با ورژن "v1.0.0" بزنید. در روز تحویل حضوری این tag بررسی خواهد شد و کدهای پس از آن نمره‌ای نخواهد گرفت. برای اطلاعات بیشتر در مورد شیوه ورژن‌گذاری، می‌توانید به [این لینک](#) مراجعه کنید. البته برای این پروژه صرفاً رعایت کردن همان ورژن گفته شده کافیهست، اما خوب است که با منطق ورژن‌بندی هم آشنا بشوید.
- در روز تحویل حضوری مشارکت تمام اعضای تیم در پروژه بررسی خواهد شد و در صورت عدم مشارکت بعضی از اعضا، نمره‌ی ایشان برای آن فاز پروژه "صفر" لحاظ می‌گردد. مشارکت، با توجه به commit های افراد تیم در مخزن گیت‌هاب پروژه بررسی می‌شود.
- در هر فاز می‌توانید سه روز تاخیر به ازای کسر نمره داشته باشید که به ازای هر روز آن، ۱۰ درصد از نمره آن فاز را از دست خواهید داد. در مجموع سه‌فاز پروژه، سه روز تاخیر نیز بخشیده خواهد شد.
- به ازای هر ساعتی که پروژه را زودتر تحویل دهید، ۱۵ دقیقه به مهلت تاخیر بدون کسر نمره شما اضافه خواهد شد. این مقدار حداکثر یک روز خواهد بود که در صورت ارسال ۴ روز زودتر از ددلاین به شما تعلق خواهد گرفت. **بنابراین ددلاین‌های پروژه تحت هیچ شرایطی تمدید نخواهد شد.** توصیه می‌شود با برنامه‌ریزی مناسب به ددلاین‌های درس پایبند باشید.
- در صورت کشف تقلب از هریک از تیم‌ها، برای بار اول منفی نمره آن فاز برای آن تیم ثبت می‌شود و برای بار دوم، نمره منفی کل پروژه برای تیم لحاظ خواهد شد که معادل مردود شدن در درس است.



## مقدمه

### اهداف پروژه

- هدف این پروژه، طراحی یک بازی کارتی مشابه عنوان Yu-Gi-Oh! است. فاز اول به شکل عمده به منطق بازی اختصاص دارد.
- در این فاز از پروژه، طراحی شیء‌گرای بازی و جداسازی صحیح منطق بخش‌های مختلف از یکدیگر مورد نظر است.
- یکی از اهداف پروژه، آشنایی با برخی ابزارها و Pattern های استاندارد برنامه‌نویسی، مثل ابزار مدیریت و تعریف پروژه [Apache Maven](#)، است.
- آشنایی با سیستم مدیریت نسخه Git و کار تیمی بر روی پروژه بر بستر یک مخزن Github، یکی از اهداف مهم پروژه است. در این مورد توصیه می‌شود تغییرات خود را در دوره‌های کوتاه مدت commit کنید.

### کلیات پروژه

در این فاز، صرفاً منطق پروژه، بدون پیاده‌سازی گرافیک یا معماری شبکه آن، باید پیاده‌سازی شود. نحوه ارتباط با کاربر نیز از طریق واسط کاربری کنسول است. توجه داشته باشید که در فاز سوم پروژه، باید سیستم را طبق یک معماری سرور-کلاینت طراحی کنید. در معماری اکثر بازی‌هایی که به این شکل انجام می‌شوند، منطق بازی در سرور و مستقل از واسط کاربری سمت کلاینت و کاربر است. هر چند برای فاز اول نیاز به این مسئله ندارید ولی خوب است که از الآن طراحی خوبی داشته باشید که قسمت‌های مختلف پروژه نظیر منطق اصلی انجام بازی، وابستگی اساسی به بخش‌هایی نظیر واسط کاربری نداشته باشد. در ادامه مستند، موجودیت‌ها، نمای کلی رابط کاربری سیستم، نقش‌ها و دستورات لازم شرح داده شده است.

نکته ۱: تمامی اطلاعات، اعم از اطلاعات کاربران، کارت‌ها و... باید در خارج از برنامه (مثلاً روی فایل) ذخیره شوند و پس از terminate شدن برنامه و اجرای مجدد آن، بصورت خودکار اطلاعات قبلی خوانده شود و قابل دسترسی باشد. برای این کار می‌توانید از ابزارهای کار با Json در جاوا، مثل [Gson](#) استفاده کنید.



نکته ۲: شما باید پروژه‌ی خود را بر بستر ابزار Apache Maven پیاده‌سازی کنید. همچنین برای اضافه کردن کتابخانه‌های مورد نیاز، dependency های مربوطه را به فایل pom.xml اضافه کنید.

نکته ۳: در هر جایی از پروژه می‌توانید هرگونه خلاقیتی را به‌کار ببرید. با این حال توجه کنید که خواسته‌های واضح پروژه بایستی انجام شوند و سیستم ورودی گرفتن و خروجی دادن شما باید مطابق جزئیات گفته شده در این مستند باشد.

نکته ۴: در مستند بعضی از دستورهای که مشاهده می‌کنید فرمتی به شکل زیر دارند:

#### دستور

```
user login -username <username> -password <password>
```

در چنین مواردی که شامل پارامترهایی هستند که با - یعنی دو کاراکتر دش، مشخص شده‌اند، باید بتوانیم از دستور با هر ترتیبی استفاده کنیم. یعنی دستور زیر

#### دستور

```
user login -password <password> -username <username>
```

هم دستور معتبری است.

همچنین به عنوان نمره امتیازی، می‌توانید حالت مخفف شده را هم برای آنان پیاده‌سازی کنید. نحوه مخفف سازی به صورت تک حرفی با و با یک نماد - خواهد بود و نحوه انتخاب حروف برعهده خودتان است. مثلاً برای نمونه بالا چنین عبارتی می‌تواند پیشنهاد خوبی باشد:

#### دستور

```
user login -p <password> -u <username>
```



## معرفی بازی

این دو عبارت زیر مربوط به داک سال قبلن. صرفا برای این که قالب نوشتن این مدل دستورات که label خاص کنارشون هست یا توی حاشیه خاص هستن دستتون باشه، گذاشتم. این خط رو پاک کنید بعدا. TODO

■ نام کاربری، نام، نام خانوادگی، ایمیل، شماره تلفن، رمز عبور

### دستور

```
create account [type] [username]
```

## توضیح بخش‌های مختلف پروژه

### کاربر، منوها و موارد عمومی

در این بازی مانند تمام بازی‌های دیگر یک عده کاربر وجود دارند که بازی می‌کنند. هر کاربر باید فیلدهای زیر را داشته باشد:

- username
- password
- nickname
- score

در ابتدای بازی هر فرد باید ثبت نام کند و سپس در دفعات بعدی فقط وارد شود و بازی کند.

همچنین یک تابلو امتیازات وجود دارد که کاربران را بر اساس امتیازی که دارند، رتبه‌بندی می‌کند و با نام مستعارشان نمایش می‌دهد.

در فاز آخر پروژه نیز باید امکان چت (global chat) میان کاربران فراهم شود؛ اما برای این فاز با توجه به نبود شبکه، نیازی به پیاده‌سازی آن نیست.

پس از ورود به بازی، وارد منوی ورود و ثبت‌نام (Login Menu) می‌شویم و پس از ساخت



اکانت و وارد شدن کاربر به منوی اصلی (Main Menu) می‌رویم که راه ارتباطی میان تمام اجزای مختلف بازی است. لیست منوهایی که باید در منوی اصلی پشتیبانی شوند به شرح زیر است:

- New Game
- Deck
- Shop
- Profile
- Scoreboard
- Exit

دستورات هر منو فقط داخل آن‌ها معتبر است و اگر در منوی مربوطه صدا زده نشوند، باید خطای مناسب چاپ شود.

### دستورات مرتبط با کاربر و منو

ورود به یک منو:

#### دستور

```
menu enter <menu name>
```

در صورتی که کاربر در منوی اصلی باشد و وارد منوی Exit شود، بازی باید خاتمه یابد. در صورتی که کاربر داخل یک منوی دیگر باشد باید خطای زیر چاپ شود:

#### پیغام به کاربر

```
menu navigation is not possible
```

خروج از یک منو:

**دستور**`menu exit`

در صورتی داخل یک منو باشیم، این دستور ما را به منوی بالاتر می‌برد و اگر در منوی اصلی باشیم وارد منوی ورود و ثبت‌نام خواهیم شد. در صورتی که در منوی ورود و ثبت‌نام بودیم، بازی خاتمه خواهد یافت.

**منوی فعلی:****دستور**`menu show-current`

این دستور نام منوی فعلی را نشان می‌دهد و اگر در منوی اصلی باشیم Main Menu و اگر در منوی ورود و ثبت‌نام باشیم Login Menu چاپ می‌شود.

**ساخت کاربر جدید:****دستور**`user create -username <username> -nickname <nickname> -  
password <password>`

پیغام موفقیت:

**پیغام به کاربر**`user created successfully!`

خطاهای زیر در صورت وجود به همین ترتیب بررسی شوند:  
نام کاربری تکراری:

**پیغام به کاربر**`user with username <username> already exists`

نام مستعار تکراری:





### پیغام به کاربر

user with nickname <nickname> already exists

### ورود کاربر:

### دستور

user login -username <username> -password <password>

پیغام موفقیت:

### پیغام به کاربر

user logged in successfully!

خطاهای زیر در صورت وجود به همین ترتیب بررسی شوند:  
عدم وجود کاربر با این نام کاربری:

### پیغام به کاربر

Username and password didn't match!

رمز اشتباه:

### پیغام به کاربر

Username and password didn't match!

توجه کنید که از دیدگاه امنیت، یک تلاش ناموفق برای ورود هیچ‌گاه نباید دارای اطلاعاتی باشد که فرایند ورود را برای بار دوم راحت‌تر کند. مثلاً اگر فردی صرفاً قصد دزدیدن یکسری اکانت بدون توجه به صاحب آن‌ها را داشته باشد و در نتیجه یکسری نام‌کاربری تصادفی را امتحان کند، در صورتی که نام‌کاربری و پسورد مطابقت نداشت، نباید از پیام ما متوجه بشود که نام‌کاربری که وارد کرده، واقعا در سیستم وجود دارد. در نتیجه در هر دو حالت بالا پیغام خطای یکسانی چاپ می‌شود.

### خروج کاربر:

**دستور**`user logout`

این دستور تنها در منوی اصلی معتبر است و در صورت موفقیت، وارد منوی ورودی و ثبت‌نام می‌شود.  
پیغام موفقیت:

**پیغام به کاربر**`user logged out successfully!`**جدول امتیازات:****دستور**`scoreboard show`

پس از ورود به منوی scoreboard با اجرای دستور باید لیست کاربران که بر اساس امتیازشان به صورت نزولی مرتب شده است به همراه نام مستعارشان نمایش داده شود. اگر دو کاربر امتیاز برابر داشتند، نام مستعارشان را به ترتیب حروف الفبا اما با رتبه یکسان چاپ کنید. هر کاربر در یک خط و با فرمت زیر چاپ شود:

**پیغام به کاربر**`rank- <nickname>: <score>`

نمونه‌ای از خروجی به شکل زیر است:

**نمونه خروجی**

```
1- shahin: 5000
2- hovakhshatara: 4000
3- ebrahim_1379: 3000
3- the-ultimate-mahD: 3000
5- rostam dastan: 1000
```



## پرو فایل:

پس از ورود به منوی profile دستورات زیر معتبر هستند:  
تغییر نام مستعار:

### دستور

```
profile change -nickname <nickname>
```

پیغام موفقیت:

### پیغام به کاربر

```
nickname changed successfully!
```

در صورتی که کاربری با این نام مستعار وجود داشت خطای زیر چاپ شود:

### پیغام به کاربر

```
user with nickname <nickname> already exists
```

## تغییر رمز عبور:

### دستور

```
profile change -password -current <current password> -new <new password>
```

پیغام موفقیت:

### پیغام به کاربر

```
password changed successfully!
```

خطاهای زیر در صورت وجود به همین ترتیب بررسی شوند:  
نادرست بودن رمز فعلی:

### پیغام به کاربر

```
Current password is invalid
```



یکی بودن رمز قدیم و جدید:

پیغام به کاربر

please enter a new password

## معرفی ساختار کارت‌ها

### دک و دسته کارت‌ها

دسته کارت یا به اختصار د.ک. (Deck) در واقع مخزنی از کارت‌هاست که شما با آن به مصاف حریف خود می‌روید. هر دک به دو دسته تقسیم می‌شود:

- دک اصلی
- دک جانبی

#### دک اصلی (Main Deck):

دک اصلی شامل حداقل ۴۰ کارت و حداکثر ۶۰ کارت است. از هر کارت تنها ۳ عدد می‌تواند در یک دک باشد و بعد از اضافه شدن آن کارت به دک، آن کارت از موجودی کارت‌های بازیکن حذف می‌شود، به عبارتی اگر بخواهید از یک کارت در چند دک استفاده کنید، باید به تعدادی که از آن کارت در دک می‌خواهید قرار دهید، در موجودی خود کارت داشته باشید.

#### دک فرعی (Side Deck):

دک جانبی شامل حداقل صفر و حداکثر ۱۵ کارت است. کاربرد دک جانبی این است که شما می‌توانید در حین بازی، یک کارت از دک جانبی را با دک اصلی جابه‌جا کنید تا استراتژی خود را بهبود ببخشید. بنابراین تعداد کارت‌های دک جانبی در بازی همیشه ثابت است و تنها یک کارت میان دک جانبی و اصلی جابه‌جا می‌شود.



### نکات کلی:

- هر بازیکن برای بازی، باید حتماً یک دک معتبر داشته باشد و دکی معتبر است که دک اصلی آن حداقل ۴۰ کارت داشته باشد.
- هر بازیکن می‌تواند به تعداد دلخواه دک داشته باشد اما تنها یک دک به عنوان دک فعال (Active Deck) انتخاب شده و با آن بازی می‌شود.
- نام دک‌ها نباید تکراری باشد.
- از آنجایی که تعداد مجاز یک کارت در دک ۳ است، دک جانبی نیز نباید این قانون را نقض کند، به عبارتی مجموع یک کارت در دک اصلی و جانبی ۳ است و اگر از یک کارت ۳ عدد در دک اصلی وجود داشت، دیگری نمیتوان آن کارت را به دک جانبی اضافه کرد.

### دستورات مرتبط با دک:

پس از ورود به منوی Deck، دستورات زیر معتبر هستند: ساخت دک جدید:

#### دستور

```
deck create <deck name>
```

توجه: نام دک می‌تواند چند بخشی باشد. (به طور مثال: my first deck)  
پیغام موفقیت:

#### پیغام به کاربر

```
deck created successfully!
```

خطا در صورتی که دک با این نام وجود داشته باشد:

#### پیغام به کاربر

```
deck with name <deck name> already exists
```

### حذف دک:



### دستور

deck delete <deck name>

توجه: زمانی که یک دک حذف می‌شود، تمام کارت‌های موجود در آن به موجودی کارت‌های بازیکن باز می‌گردد.  
پیغام موفقیت:

### پیغام به کاربر

deck deleted successfully

خطا در صورتی که دک با این نام وجود نداشته باشد:

### پیغام به کاربر

deck with name <deck name> does not exist

## انتخاب دک به عنوان دک فعال:

### دستور

deck set-activate <deck name>

توجه: معتبر نبودن دک در هنگام شروع بازی بررسی خواهد شد.  
پیغام موفقیت:

### پیغام به کاربر

deck activated successfully

خطا در صورتی که دک با این نام وجود نداشته باشد:

### پیغام به کاربر

deck with name <deck name> does not exist

## اضافه کردن کارت به دک:

**دستور**

```
deck add-card -card <card name> -deck <deck name> -side(optional)
```

توجه کنید که تنها برای اضافه کردن کارت به side deck از پرچم side استفاده می‌کنیم. اگر از آن استفاده نکنیم، به معنی اضافه شدن به دک اصلی است. پیغام موفقیت:

**پیغام به کاربر**

```
card added to deck successfully
```

خطاهای زیر در صورت وجود باید به همین ترتیب بررسی شوند:  
عدم وجود کارت با این نام در موجودی کارت‌های بازیکن:

**پیغام به کاربر**

```
card with name <card name> does not exist
```

عدم وجود دک با این نام:

**پیغام به کاربر**

```
deck with name <deck name> does not exist
```

پر بودن دک اصلی یا جانبی:

**پیغام به کاربر**

```
<main/side> deck is full
```

اگر در حال حاضر از یک کارت ۳ تا در دک موجود باشد:

**پیغام به کاربر**

```
there are already three cards with name <card name> in deck <deck name>
```

**حذف کارت از دک:**

**دستور**

```
deck rm-card -card <card name> -deck <deck name> -side(optional)
```

توجه کنید که تنها برای حذف کردن کارت از deck side از پرچم side- استفاده می‌کنیم.  
توجه: زمانی که یک کارت از دک حذف می‌شود، به موجودی کارت‌های بازیکن برمی‌گردد.  
پیغام موفقیت:

**پیغام به کاربر**

```
card removed form deck successfully
```

خطاهای زیر در صورت وجود باید به همین ترتیب بررسی شوند:  
عدم وجود دک با این نام:

**پیغام به کاربر**

```
deck with name <deck name> does not exist
```

عدم وجود کارت با این نام در دک اصلی یا جانبی:

**پیغام به کاربر**

```
card with name <card name> does not exist in <main/side> deck
```

**نمایش دک‌های بازیکن:****دستور**

```
deck show -all
```

خروجی باید با فرمت زیر باشد:  
هر دک با فرمت زیر نمایش داده می‌شود که ابتدا نام دک، سپس تعداد کارت‌های موجود در دک اصلی و سپس دک جانبی و در انتها معتبر بودن یا نبودن دک می‌آید.



**پیغام به کاربر**

<deck name>: main deck <number of main deck cards>, side deck  
<number of side deck cards>, <valid/not valid>

به طور مثال:

**خروجی نمونه**

My best: main deck 50, side deck 5, valid

خروجی کلی به شکل زیر است:  
ابتدا در صورتی که یک دک به عنوان دک فعال انتخاب شده باشد، نمایش داده خواهد شد  
و در صورتی که دک فعالی وجود نداشته باشد چیزی زیر Active deck چاپ نمی‌شود،  
سپس بقیه دک‌ها زیر Other decks به ترتیب حروف الفبا چاپ می‌شوند:

**پیغام به کاربر**

Decks:  
Active deck:  
<active deck>  
Other decks:  
<other decks>

اگر کاربر هیچ دک‌ی نداشته باشد خروجی به شکل زیر خواهند بود:

**خروجی نمونه**

Decks:  
Active deck:  
Other decks:

نمایش یک دک:

**دستور**

deck show -deck-name <deck name> -side(Optional)



توجه: برای نمایش دک فرعی از پرچم side- استفاده می‌کنیم.  
خروجی:

### پیغام به کاربر

Deck: <deck name>

Side/Main deck:

Monsters:

<card name>: <card description>

Spell and Traps:

<card name>: <card description>

دقت شود که در هر قسمت ترتیب کارت‌ها بر اساس ترتیب الفبایی نامشان است.

**فروشگاه**

**گیم پلی بازی**

**هوش مصنوعی**

**حالت توسعه دهنده**

**لاگ نوشتن**

**لیست کارت‌ها**

**کارت‌های هیولا**

**کارت‌های جادو**

**کارت‌های تله**