

به نام خدا



## درس برنامه سازی پیشرفته

فاز سه بخش اول

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۰۳-۰۲

---

استاد:

دکتر محمد امین فضلی

# فهرست مطالب

۲	قوانین و نکات قابل توجه
۳	نکات قابل توجه در پیاده‌سازی پروژه
۴	معماری شبکه
۴	معماری Server-Client
۴	بازی همزمان
۴	احراز هویت
۵	اجزای قابل پیاده‌سازی
۵	دوستی
۵	تشکیل بازی
۵	جدول امتیازات
۵	واکنش در بازی
۵	پخش تلویزیون
۶	گفتگو در پخش زنده
۶	جام حذفی (امتیازی)
۶	ادامه بازی ناتمام (امتیازی)
۷	پایگاه داده (امتیازی)



## قوانین و نکات قابل توجه

- در روز تحویل حضوری مشارکت تمام اعضای تیم در پروژه بررسی خواهد شد و در صورت عدم مشارکت بعضی از اعضا، نمره ایشان برای آن فاز ۰ لحاظ میگردد. مشارکت با توجه به commit های افراد تیم در مخزن گیت‌هاب پروژه بررسی میشود.
- در صورت کشف تقلب، 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.



## نکات قابل توجه در پیاده‌سازی پروژه

- توصیه می‌شود برای پروژه از پکیج‌منیجر Maven استفاده کنید. می‌توانید در اینجا نحوه استفاده از این ابزار را ببینید.
- توصیه می‌شود از `hardcode` کردن کارت‌ها پرهیز کنید. یعنی به ازای `N` کارتی که باید پیاده‌سازی کنید، `N` کلاس جداگانه ایجاد نکنید. سعی کنید ساختار کلاس‌هایتان بصورت سلسله‌مراتبی از طریق `inheritence` یا با استفاده از `composition` باشد تا در آینده بتوانید به راحتی آن را `extend` کنید.
- در انتخاب ورژن جاوای مورد استفاده مختار هستید اما حتما ورژن جاوای نصب‌شده و مورد استفاده توسط همه افراد هر تیم یکسان باشد. در صورت تفاوت ورژن جاوا بین اعضای تیم، احتمال بروز مشکلات فنی در حین توسعه پروژه بالا است.
- هنگام استفاده از گیت، توجه کنید که فایل `gitignore` به درستی تشکیل دهید تا موارد نادرست را روی مخزن خود اضافه نکنید. این ابزار می‌تواند به شما در ساخت بهتر این فایل کمک بکند.
- در پیاده‌سازی فاز شبکه، باید از معماری `server-client` استفاده کنید و نحوه پیاده‌سازی شما باید به گونه‌ای باشد که در سمت `client` منطقی از برنامه اجرا نشود و درخواست‌ها در سمت `server` پردازش شوند.



## معماری شبکه

### معماری Server-Client

معماری کارخواه- کارگزار یا Server-Client یکی از معروف‌ترین معماری‌های شبکه‌های کامپیوتری است. در معماری Server-Client کاربران معمولی کلاینت نامیده شده و هر کدام از آن‌ها درخواست‌هایی را برای سرور ارسال می‌کنند. سرور به منابع و اطلاعات اصلی برنامه دسترسی دارد و پردازش‌های اصلی داده‌ها در آن انجام شده و در نهایت نتیجه به شکل مناسبی به کلاینت اطلاع داده می‌شود. برای این فاز از پروژه باید از این معماری استفاده کنید. به این شکل که سرور، اطلاعات اصلی نظیر بازی‌های در حال انجام و ... را در اختیار داشته و بسته به درخواست‌هایی که برای آن ارسال می‌شود، پاسخ مناسب را برای هر کلاینت ارسال می‌کند. به بیان دیگر، بخش عمده منطق برنامه باید در سمت سرور رسیدگی شود. برای درک عمیق‌تر شبکه می‌توانید داک شبکه را مطالعه کنید.

### بازی همزمان

بازی شما تا کنون به این صورت بود که شما از طریق یک کامپیوتر به جای افراد مختلف بازی می‌کردید و هر کدام از این افراد، تنها می‌توانستند در نوبت خودشان حرکت کنند. به این نوع بازی‌ها، بازی‌های turn-based می‌گویند. در این فاز، شما باید معماری شبکه خود را به گونه‌ای طراحی کنید که در یک لحظه، چند بازی متفاوت بین بازیکن‌های مختلف در جریان باشند.

### احراز هویت

در فرآیندهای ثبت‌نام و ورود، باید یک روش مناسب را که در داک احراز هویت ارائه شده است، استفاده کنید و فرآیندهای ثبت‌نام و ورود را بر اساس آن پیاده‌سازی کنید.

**امتیازی:** فرآیند ثبت‌نام همراه با ایمیل احراز هویت و تایید شدن آن توسط کاربر با کلیک بر روی لینک ایمیل شده باشد.

**امتیازی:** فرآیند ورود به صورت احراز هویت دو مرحله‌ای باشد. به این صورت که پس از وارد کردن نام کاربری و رمز عبور، به مرحله بعدی وارد شویم. سپس یک کد ۶ رقمی به فرد ایمیل شود و با وارد کردن آن کد، فرآیند احراز هویت تکمیل شود. (در رابطه با 2FA جست‌وجو کنید)



## اجزای قابل پیاده‌سازی

### دوستی

شما باید این قابلیت را پیاده‌سازی کنید که هر کاربر بتواند به کاربران دیگر درخواست دوستی بدهد و در صورت موافقت طرف مقابل، طرفین به لیست دوستان یکدیگر اضافه می‌شوند. این فرآیند باید در منوی پروفایل انجام شود. به این صورت که با سرچ کردن نام کاربری یک بازیکن و مشاهده در پروفایل، درخواست دوستی ارسال شود. سپس در منوی پروفایل فردی که به او درخواست ارسال شده است، تاریخچه درخواست‌های دوستی او با وضعیت رد، تایید و یا در انتظار پاسخ قرار داشته باشد. سپس با پاسخ مثبت فرد مقابل، دوستی بین دو فرد شکل می‌گیرد. دوستان می‌توانند به همدیگر درخواست مستقیم بازی بدهند.

### تشکیل بازی

پیش از تشکیل بازی، فرد باید به داخل منوی پیش‌بازی برود و پس از انتخاب قلمرو و کارت‌های داخل دست به دو صورت می‌تواند یک بازی را شروع کند:

- فرد در منوی پیش‌بازی نام یکی از دوستانش را وارد می‌کند و در صورتی که آن فرد آنلاین بود و داخل بازی دیگری نبود، یک فرم به صورت pop-up ظاهر می‌شود و درخواست بازی را به اطلاع فرد می‌رساند. در این حالت در صورت تایید طرف مقابل، خودکار به منوی پیش‌بازی منتقل می‌شود و سپس پس از انتخاب قلمرو و دست کارت وارد بازی می‌شوند.
- با انتخاب حالت بازی رندوم، فرد منتظر کاربری که دنبال بازی رندوم است می‌ماند. در صورت پیدا شدن فرد دیگر در حال انتظار، بین طرفین بازی شروع می‌شود.

هنگام تشکیل هر بازی، public یا private بودن آن نیز باید مشخص شود.

### جدول امتیازات

جدول امتیازات فاز پیش باید قابلیت‌های زیر را نیز داشته باشد.

- نمایش آنلاین یا آفلاین بود افراد
- آپدیت شدن خودکار اطلاعات جدول در صورتی تغییر
- امتیازی امکان مشاهده ویدئو آخرین بازی ۱۰ نفر برتر

### واکنش در بازی

دو نفر در جریان بازی باید بتوانند نسبت به یکدیگر واکنش نشان دهند. این واکنش می‌تواند یک پیام از پیش آماده، یک پیام دلخواه با حداکثر ده کاراکتر و یا یک ایموجی باشد. فرستنده این واکنش باید مشخص باشد، و پس از ۷ ثانیه از روی صفحه بازی پاک شود.

**امتیازی:** بتوان به هر قراردادن کارت حریف روی زمین بازی به صورتی که واکنش نشان‌دادن، قابل تشخیص باشد.

### پخش تلویزیون

تلویزیون بخش جدیدی است که در این فاز اضافه شده است. این بخش دارای دو قسمت پخش زنده بازی و بازپخش بازی‌های قبلی کاربر است.



## پخش زنده بازی

هدف این قسمت به اشتراک گذاری بر خط بازی‌ها است. برای مشاهده بازی‌های در حال انجام، باید یک منوی جدید ایجاد کنید که در آن یک بازی در جریان را انتخاب کنید و پس از انتخاب پخش زنده آن را مشاهده کنید. اگر بازی به صورت public باشد، برای همه افراد قابل مشاهده است. در غیر این صورت مشاهده پخش زنده بازی، تنها برای افرادی مقدور است که دوست هر دو طرف بازی باشند.

یک روش پیشنهادی برای پیاده‌سازی این بخش ثبت گزارش یا log است. در این روش ابتدا حالت اولیه بازی ذخیره می‌شود و سپس تمام حرکات بعدی ذخیره می‌شود. سپس چون حالت اولیه بازی را می‌دانیم، برای هر کاربری که قصد پخش زنده داشته باشد، یک نمونه از بازی را می‌سازیم و تمام حرکات را بر اساس این گزارش اجرا می‌کنیم. اگر به طور مداوم این گزارش‌ها را اجرا کنیم، پخش زنده ادامه می‌یابد.

## بازپخش بازی‌های قبلی

در این بخش کاربر می‌تواند بازی‌های قبلی خود را دوباره مشاهده و بررسی کند.

**امتیازی:** هر بازی باید با یک تصویر از آخرین وضعیت بازی مشخص شود و در طی زمان آن تصویر نیز آپدیت شود.

**امتیازی:** روش پیاده‌سازی به این صورت باشد که اطلاعات دریافت شده برای کاربر مشاهده‌کننده بازی داده‌های ویدیویی باشند و نتیجه مشاهده شده توسط کاربر استریم یک فایل ویدیویی قابل ذخیره‌سازی باشد.

## گفتگو در پخش زنده

شما باید قابلیت chat عمومی بین تمام افرادی که پخش زنده بازی را مشاهده می‌کنند و همچنین دو نفری که بازی را بازی می‌کنند، ایجاد کنید. به این صورت که در منوی اصلی بازی و پخش زنده، یک منوی جانبی اضافه کنید و در آنجا افراد بتوانند پیام ارسال کنند و برای سایر افراد پیام آن‌ها قابل مشاهده باشد.

در هر پیام فرستنده و ساعت فرستاده شدن پیام باید مشخص باشد.

**امتیازی:** قابلیت reply به پیام‌ها وجود داشته باشد.

**امتیازی:** قابلیت reaction به پیام‌ها وجود داشته باشد.

## جام حذفی (امتیازی)

شما باید یک بخش به منوی پیش‌بازی اضافه کنید تا با انتخاب آن، فرد منتظر شروع یک جام حذفی می‌شود. سپس با ورود ۸ نفر به این حالت، یک جام‌حذفی به صورت دو حذفی بین ۸ نفر تشکیل می‌شود. به این صورتی که هر فرد پس از اولین باخت وارد دسته بازنده‌ها می‌شود و با بازنده طرف خود مسابقه می‌دهد تا به فینال برسد. همچنین در صورت برد، فرد منتظر طرف می‌ماند تا نتیجه بازی مقابل مشخص شود و تا بازی مرحله بعدی شروع شود.

یک فرد در صورت دو بار باختن، از دوره مسابقات حذف می‌شود و باید بتواند پخش زنده تمام بازی‌های در جریان را مشاهده کند و بین آنها جابه‌جا شود. همچنین علاوه بر پخش زنده بازی‌های در جریان باید بتواند جدول نتایج مسابقات جام‌حذفی را به صورت یک گراف تا آن لحظه ببیند و در صورتی که نتیجه مسابقه جدیدی مشخص شد، آن جدول آپدیت شود.

## ادامه بازی ناتمام (امتیازی)

اگر در میان یک بازی ارتباط یک بازیکن با سرور قطع شود، به مدت محدودی (مثلاً حداکثر دو دقیقه) فرصت دارد تا دوباره به بازی متصل شود. به این صورت که اگر قبل از دو دقیقه دوباره به سرور متصل شود، پس از لاگین به ادامه بازی می‌پردازد و در غیر این صورت بازنده اعلام می‌شود. در این بازه باید برای طرف حاضر در بازی، پیامی در ارتباط با این مسئله نمایش داده شود.



## پایگاه داده (امتیازی)

شما باید برای ذخیره‌سازی اطلاعات به جای نگهداری آنها در فایل متنی، در قالب JSON یا XML آنها را در یک پایگاه داده ذخیره کنید. برای اطلاعات بیشتر به داک پایگاه داده مراجعه کنید.