

به نام خدا



درس برنامه سازی پیشرفته

فاز اول پروژه

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۰۱-۰۰

استاد:

دکتر محمد امین فضلی

مهلت ارسال:

چک پوینت ۱: ۲ اردیبهشت

چک پوینت ۲: ۱۱ اردیبهشت

فاز اول: ۱۶ اردیبهشت

ساعت ۲۳:۵۹:۵۹

مسئول پروژه:

امیرمهدی کوششی

مسئول فاز اول:

آرمان بابایی

طراحان فاز اول:

علی پاشا منتصری، رضا وحیدی مجد، حسین گلی، امیرمحمد افلاطونیان، امیرمحمد فحیمی، زهرا اذر، مهدی صابر، درین ستوده، پوریا ممتاز، امیررضا ابوطالی، علی بناقتی زاده، ابوالفضل قلندری

مسئولین تنظیم مستند:

امیرمهدی کوششی، هیربد بهنام، ناصر کاظمی، علی ثالثی

فهرست

۲	نکات قابل توجه
۳	مقدمه
۳	اهداف قابل توجه
۳	کلیات پروژه
۶	معرفی بازی
۶	توضیح بخش‌های مختلف پروژه
۶	کاربر، منوها و موارد عمومی
۷	دستورات مرتبط با منو
۱۲	نقشه بازی
۱۳	کد تقلب
۱۴	پیاده سازی فاز ۱
۱۴	چک پوینت
۱۵	چک پوینت ۱
۱۵	چک پوینت ۲
۱۵	فاز ۱



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 کافیست، 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.
- شما باید حداقل برای یکی از فرایندهای موجود در پروژه، Unit Test بنویسید. همچنین برای code coverage بالاتر از ۷۰ درصد، به صورت خطی با درصد coverage، نمره امتیازی تعلق خواهد گرفت.



مقدمه

اهداف پروژه

- هدف این پروژه، طراحی یک بازی مشابه عنوان civilization 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. مثلاً در فاز آخر پروژه باید امکان چت (Global Chat) میان کاربران فراهم شود؛ اما برای این فاز با توجه به نبود شبکه، نیازی به پیاده‌سازی آن نیست.

همچنین توجه کنید که در فاز دوم قرار است برای منطق بازی که در این فاز طراحی شده است، رابط کاربری گرافیکی طراحی کنید. در نتیجه جداسازی درست منطق بازی از رابط کاربری آن باعث می‌شود که در فاز ۲ نیاز به بازنویسی کدهای منطق به شدت کاهش یابد. در ادامه مستند، موجودیت‌ها، نمای کلی رابط کاربری سیستم، نقش‌ها و دستورات لازم شرح داده شده است.

نکته ۱: دقت کنید که دو داک در اختیار شما قرار می‌گیرد، یک داک توضیحات کاملی از بازی است، و همه چیز راجع به بازی در آن توضیح داده شده است، یک داک دیگر مربوط به فاز اول پروژه شما می‌باشد که همین داک است، و در این داک، قسمت‌هایی از بازی که شما باید در فاز اول پیاده‌سازی کنید آمده است، دقت کنید چیزهایی



که در این داک نیامده است، نیاز نیست در این فاز طراحی کنید و تحویل دهید، فقط به آن‌ها برای طراحی منطق کد خود، توجه داشته باشید.

نکته ۲: تمامی اطلاعات، اعم از اطلاعات کاربران 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** یا **YaGson** استفاده کنید.

نکته ۳: 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 ۴: 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 ۵: در فاز ۱ شما باید از طریق کنسول با برنامه خود ارتباط برقرار کنید، به همین دلیل نیاز به یک سری دستورات مشخص می‌باشد که شما از طریق آن دستورات می‌توانید بازی را جلو ببرید. به همین منظور یک شیت از دستورات پیشنهادی که می‌توانید از آن‌ها استفاده کنید در این **لینک** آمده است. (دقت کنید این دستورات پیشنهادی هستند و شما می‌توانید دستورات خود را به سلیقه خود پیاده‌سازی کنید، اما نباید دستورات شما از یک چهارچوب معین خارج شود، در ادامه نیز بیشتر با این موضوع آشنا خواهید شد.)

نکته ۶: در مستند بعضی از دستورهایی که مشاهده می‌کنید فرمتی به شکل زیر دارند:

دستور

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

درواقع ما زمانی از دو دش استفاده می‌کنیم، که بخواهیم یک فیلد را از ورودی بگیریم، در این دستور دو فیلد username و password به این منظور استفاده می‌شود، در نمونه‌ای دیگر نیز اگر شما بخواهید به عنوان مثال مشخصات یک خانه را دریافت کنید، می‌توانید از `<coordinates> --coordinates` استفاده کنید، شما باید در دستورات ساخت خودتان نیز برای گرفتن یک فیلد از دو دش استفاده کنید. (این یک نمونه از همان چهارچوبی است که شما باید در ساخت دستورات خود به آن توجه کنید.) همچنین 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>
```

همچنین توجه کنید که اگر دستور نامعتبری وارد شد که در قسمت مربوطه، برای آن خطای به خصوصی در نظر گرفته نشده بود، پیام زیر را چاپ کنید:

پیغام به کاربر

```
invalid command
```



معرفی بازی

در مستند ضمیمه، به معرفی بازی و قواعد آن مستقل از خواسته‌های پروژه پرداخته شده است. توصیه می‌شود قبل از مطالعه این مستند که در مورد نکات پیاده‌سازی خود پروژه است، به مطالعه داک مربوط به بازی بپردازید.

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.

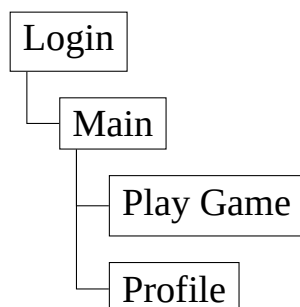
پس از ورود به بازی، وارد منوی ورود و ثبت نام (Login Menu) می‌شویم و پس از ساخت اکانت و وارد شدن کاربر به منوی اصلی (Main Menu) می‌رویم که راه ارتباطی میان تمام اجزای مختلف بازی است. توجه کنید اگر کاربر بدون لاگین کردن قصد رفتن به منوی دیگری را داشته باشد، باید با خطای زیر رو به رو شود:

پیغام به کاربر

please login first



در کل ساختار منوهای ما به شکل زیر است:



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

دستورات مرتبط با منو

ورود به یک منو:

دستور

menu enter <menu name>

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 ثبت نام خواهیم شد. در صورتی که در منوی ورود و ثبت نام 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 نام کاربری تصادفی را امتحان کند، در صورتی که نام کاربری و پسورد مطابقت نداشت، نباید از پیام ما متوجه بشود که نام کاربری 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!

بازی:

همانطور که گفته شد، این بازی یک بازی چند نفره است، لذا شما باید یوزرنیم افرادی که میخواهید با آن‌ها بازی کنید را داشته باشید تا به صورت زیر وارد بازی شوید.

دستور

play game --player1 <username> --player2 <username>



دقت کنید اگر تعداد افرادی که قرار بود با آن‌ها بازی کنید، بیشتر بود، نیاز است که به همین ترتیب در دستور شروع بازی، ابتدا player-- و سپس شماره آن بازیکن و سپس بعد از آن یوزرنیم آن را وارد کنید. در صورتی که یک یا چند یوزرنیم وجود نداشت پیغام خطای مناسب به کاربر نشان داده شود. و همینطور در صورت ورود به بازی نیز پیغام مناسب چاپ شود.

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:

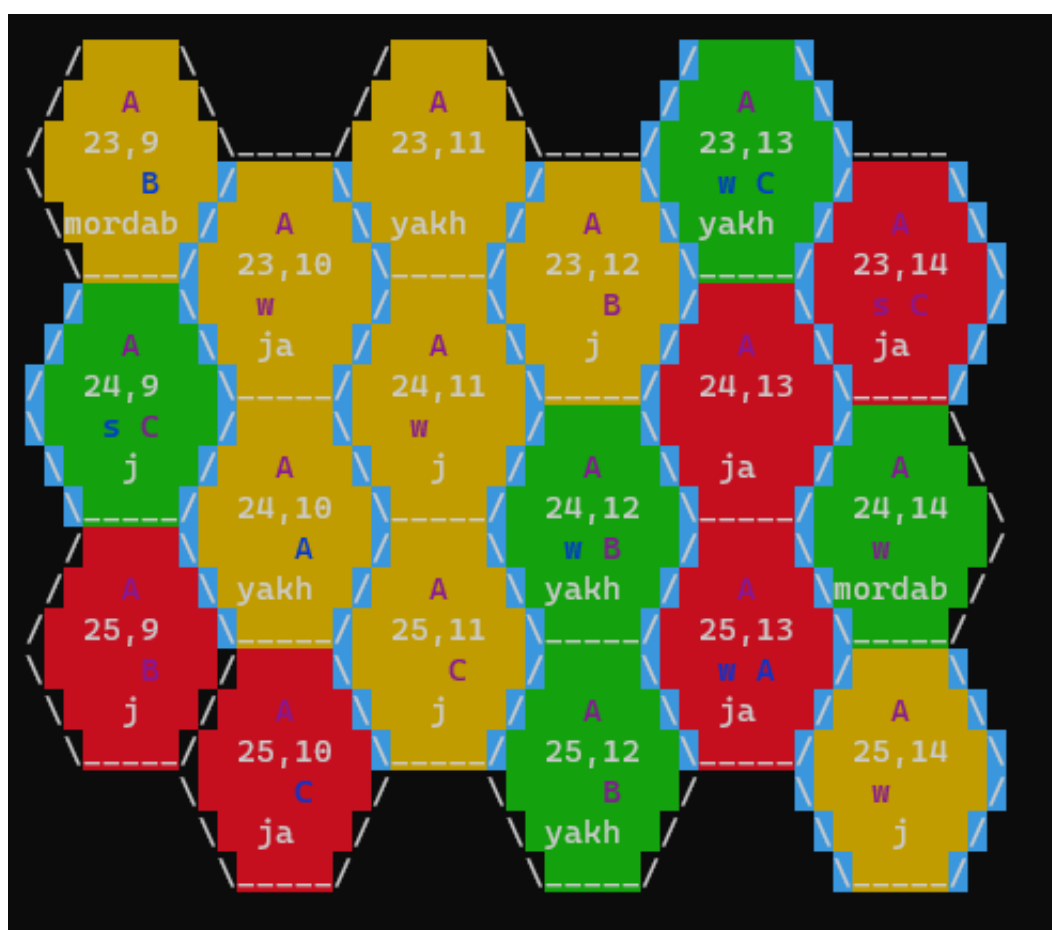
پیغام به کاربر

please enter a new password

نقشه بازی

پیشنهاد ما این است که نقشه‌ی کوچکی در ترمینال خود نمایش دهید، و با هر دستور جدید این نقشه را بروز کنید و بعد از ابتدا چاپ کنید.

نقشه‌ی شما می‌تواند عنصرهای مختلف درون نقشه را به صورت رنگ، حروف اختصاری، و یا کلمات کامل نشان دهد. همچنین می‌توانید برخی از اطلاعات را به زمان انتخاب شدن یک کاشی واگذار کنید. پیشنهاد می‌شود که مختصات هر خانه را چاپ کنید تا انتخاب یک خانه‌ی خاص در دستورات ممکن باشد. مثلاً یک نقشه‌ی ممکن می‌تواند مانند شکل زیر باشد:



به طور مثال کاشی بالا سمت چپ نقشه، در مختصات ۲۳،۹ قرار دارد. این کاشی در محدوده‌ی مرزهای تمدن A (با رنگ بنفش) است، در زمین صحرایی (با توجه به رنگ زرد پس‌زمینه‌ی آن) واقع شده و ویژگی مرداب را دارا است. در این خانه یک یگان نظامی از نوع Ballista قرار دارد که مربوط به تمدنی‌ست که رنگ آبی را دارد (حرف B)، اما یگان غیرنظامی‌ای در آن وجود ندارد. در ضلع پایین سمت راست این کاشی، یک رودخانه واقع شده است که با رنگ پس‌زمینه‌ی آبی مشخص شده است. از طرف دیگر، کاشی پایینی



آن در مختصات ۲۴،۹ در زمین علفزار (با توجه به رنگ سبز پس‌زمینه‌ی آن) قرار دارد. در این کاشی که از ویژگی جنگل (با توجه به حرف z) برخوردار است، یک یگان غیرنظامی مهاجر متعلق به تمدن آبی (با توجه به حرف s) و با توجه به حرف C یک واحد نظامی Catapult متعلق به تمدن بنفش (همان A) قرار دارد. شما باید نقشه‌ای مثل شکل بالا ایجاد کنید. البته شما در انتخاب این که چطور می‌خواهید اطلاعات درون نقشه را نشان دهید، مختارید. مثلاً می‌توانید رودخانه‌ها را به جای رنگ پس‌زمینه، با مرزهای «-» برای کاشی نشان دهید. یا به جای استفاده از رنگ پس‌زمینه برای نشان دادن نوع زمین، این اطلاعات را به طور مستقیم روی نقشه بنویسید. آنچه که در نهایت اهمیت دارد این است که وضعیت رودخانه‌ها، مالکیت زمین‌ها، یگان‌های موجود در کاشی و مالکیت آن‌ها، مختصات کاشی، نوع زمین کاشی، ویژگی زمین کاشی، منبع موجود در کاشی و پیشرفت‌های موجود در کاشی در نقشه مشخص باشد.

کد تقلب

احتمالاً در بازی‌های زیادی دیدید (شنیدید) که یک سری کدهای تقلب وجود دارد که بازی را از حالت عادی خارج می‌کند، نمونه‌ای از این کدها مثلاً در بازی GTA وجود دارد، می‌تواند باعث شود تمامی اسلحه‌ها برای بازکن در دسترس باشد، یا در مثال دیگر می‌تواند باعث شوند بازیکن هیچگاه جان خودش را از دست ندهد. در پروژه شما نیز این قابلیت باید پیاده‌سازی شود. در این حالت پروژه شما باید بتواند کارهایی که در حالت عادی نمی‌توانید انجام دهد را پیاده‌سازی کند، نمونه‌ای از این کارها را می‌توان افزایش پول به مقدار دلخواه، گذاشتن چند نوبت پشت سر هم، پیروزی در یک نبرد، به دست آوردن غذا و تکنولوژی و ... باشد. در اینجا نمونه‌ای از دستورات پیشنهادی را برای شما آورده‌ایم. شما نیز می‌توانید خودتان دستورات دیگری برای این حالت توسعه دهید و در زمینه‌های دیگر استفاده کنید. (دقت کنید برای تمام فعالیت‌های بازی اعم از برد یک دست بازی، حرکت یگان‌ها، ساختمان‌ها، یونیت‌ها، شهرها و ... کد تقلب داشته باشید.)

جلو رفتن نوبت‌های بازی:

دستور

increase –turn <amount>

زیاد شدن پول:

دستور

increase –gold <amount>



مواردی که باید در فاز ۱ پیاده سازی شوند

برای راحتی کار شما، در فاز اول، یک سری موارد کلی ای از بازی پیاده سازی می‌شوند که در واقع همان منطق اصلی بازی هستند، و سایر موارد به ترتیب در فازهای بعدی پروژه تکمیل می‌شوند.

علاوه بر موارد مخصوص به فاز ۱ که در همین داک دیدید (منوها، نقشه بازی، کدهای تقلب) شما باید مواردی از داک بازی نیز که مربوط به فاز ۱ هست را نیز پیاده سازی کنید. در واقع بازی پیاده سازی شده در فاز ۱ به شکلی به طور کلی اینگونه می‌باشد:

کاربران می‌توانند برای خودشان یک سری یوزر در بازی بسازند، اطلاعاتشان را عوض کنند، با سایر کاربران بازی کنند. در محیط بازی نیز در فاز ۱ شما باید زمین‌ها، ویژگی‌های آن‌ها، یگان‌ها، منابع، حرکت یگان‌ها بر روی زمین، منابع، تکنولوژی را داشته باشید. در فاز ۱ برد و باخت یک تمدن حمله کردن تمدن‌ها و یگان‌های نظامی به یکدیگر چک نمی‌شوند و شما باید آن‌ها را در فازهای بعدی پروژه تکمیل کنید.

به طور کلی‌تر، موارد زیر را نیاز نیست در فاز ۱ پیاده سازی کنید:

- برخی از اعمال تمدن‌ها (معامله کردن با یک تمدن دیگر، جنگ)
- برد و باخت در بازی
- خرابه‌ها (ruins)
- ساختمان‌ها

چک پوینت

یکی از مواردی که در این پروژه باید به آن دقت کنید، چک پوینت‌ها هستند. در واقع چک پوینت‌ها همان ددلاین‌های تقسیم شده در طول هر فاز می‌باشند. از شما خواسته می‌شود با توجه به زمان پایان هر چک پوینت، مواردی از فاز اول پروژه که در آن چک پوینت باید زده شوند را، با تگ "CP#.0.0" در مخزن گیت‌هاب خود قرار دهید.

در این فاز، شما ۲ چک پوینت و یک ددلاین پایانی فاز ۱ را دارید. همچنین هر چک پوینت نیز میزان نمره‌ای از کل نمره پروژه دارد.

مواردی که باید تا پایان زمان هر چک پوینت زده شوند در زیر آمده است.



چک پوینت ۱

- حجم موارد خواسته شده در این چک پوینت نسبت به کل مطالب این فاز: ۴۰٪

کامل کردن منوها (تمپلیت) و قابلیت ساخت اکانت وجود دیتابیس User ها و پیاده کردن Map بازی و معماری (لزومی به پیاده سازی کامل نیست صرفاً تقریباً مشخص باشد چه تابعایی و چه چیزهایی لازم است) و کلاس‌های لازم برای Object های اولیه مثل Unit - City - Block پیاده سازی Exploration and Movement (توانایی حرکت دادن نیروها در MAP و fog of war) هر نوع نیرو یک کلاس پدر Unit دارد و توانایی دستور دادن به این کلاس کلی مد نظر است (پیاده سازی نیروهای جنگی در ادامه)

چک پوینت ۲

- حجم موارد خواسته شده در این چک پوینت نسبت به کل مطالب این فاز: ۳۰٪

قابلیت احداث شهر و منطق مربوط به شهرها

فاز ۱

- حجم موارد خواسته شده در این چک پوینت نسبت به کل مطالب این فاز: ۳۰٪

موارد باقی مانده از پیاده سازی فاز اول پروژه