برنامه نویسی پیشرفته

زمستان ۱۴۰۴ - بهار ۱۴۰۵ - فاز اول پروژه

مقدمه

به فاز اول پروژه درس برنامه‌نویسی پیشرفته خوش آمدید. امسال، شما مسئولیت طراحی و پیاده‌سازی یک بازی استراتژیک تک‌نفره را بر عهده دارید. این پروژه فرصتی است تا با اصول برنامه‌نویسی شی‌گرا به‌طور عمیق آشنا شوید. برای درک بهتر از محیط بازی می‌توانید به بازی Civilization 6 مراجعه کنید.

در این فاز، شما مخلوطی از گرافیک، ساختار داده‌ایِ دنیای بازی و بخش قابل توجهی از منطقِ مدیریت منابع و تعاملات را پیاده‌سازی خواهید کرد. اهداف آموزشی این فاز شامل گرافیک پیشرفته، مدل‌سازی منطق بازی، بهره‌گیری از مفاهیم جریان‌ها (Streams)، جنریک‌ها و پیاده‌سازی هوشمندانه ساختارهای داده‌ای برای پشتیبانی از مقیاس بالاتر بازی در فازهای بعدی است.

اکیداً توصیه می‌شود قبل از دست به کد شدن، داک را چند بار بخوانید و یک ساختار منظم از بخش‌های مختلف کد در ذهنتان داشته باشید. بخش‌های داک به ترتیبی چیده شده‌اند که بهترین تصویر ذهنی از بازی را به شما بدهند؛ پیاده‌سازی با ترتیب تیترها می‌تواند کمک‌کننده باشد، اگرچه تصمیم نهایی با خودتان است.

منوی بازی

منوی بازی ساده و شامل ۳ گزینه است

- **Start**: با زدن این دکمه بازی شروع می‌شود.
- **Settings**: در این بخش می‌توان صدای آهنگ بازی را با استفاده از یک Scroll Bar تنظیم کرد. (بازی باید یک آهنگ پس‌زمینه و ساندترک دائمی داشته باشد)
- **Exit**: یک پنجره تاییدیه (Pop-up) نمایش داده می‌شود. در صورت تأیید، بازی بسته می‌شود.

نقشه بازی و انواع مناطق

نقشه بازی از شبکه‌ای از خانه‌های شش‌ضلعی (Hex) تشکیل شده است. هر خانه یک منطقه با نوع و منابع خاص خود است. در ابتدا، به‌جز هکسی که Town Hall شما در آن قرار دارد، بقیه مناطق ناشناخته‌اند. با حرکت Explorer(ها) در نقشه، مناطق جدید کشف می‌شوند. نمی‌توان از منابع مناطق کشف‌نشده استفاده کرد، و هر منطقه ظرفیت منبعی محدود دارد و پس از اتمام منبع باید به گونه‌ای نشان دهید که منبع آن هکس تمام شده است.

انواع مناطق

- **جنگل:** منبع چوب. با استفاده از چوب‌بری قابل بهره‌برداری است. تمامی هکس های جنگل چوب دارند.
- **دشت:** برخی هکس‌های دشت دارای حیوانات اهلی (گاو و گوسفند) هستند.
- **کوهستان:** منابع معدنی (سنگ و آهن). همه ی هکس های کوهستان دارای سنگ اند ولی همه ی انها آهن ندارند.
- **سبزه‌زار:** زمین‌های گندم و برنج. همه ی هکس سبزه زار دارای منبع غذایی نیستند. – منبع اصلی غذا از طریق مزرعه.

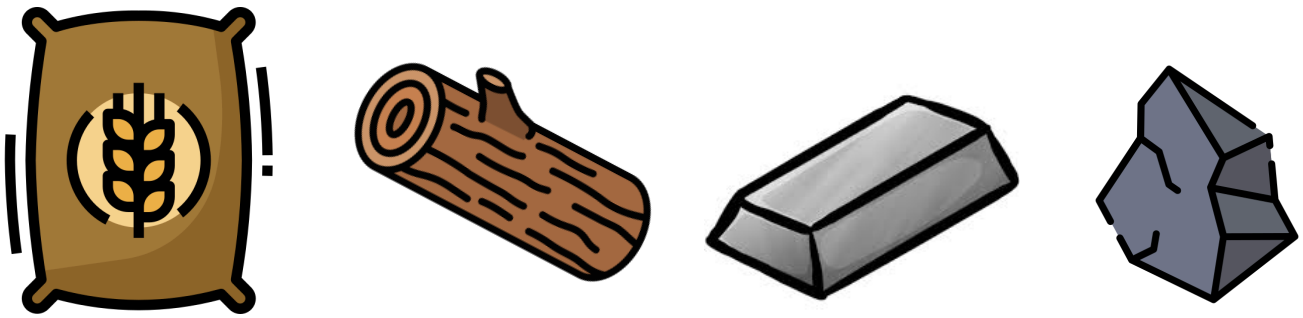
توجه کنید ظاهر زمین های دارای منابع و زمین های بدون منابع باید متفاوت باشد. همچنین باید نشان داده شود که هر زمین دارای چه منبعی است.

منابع بازی (Resources)

در بازی چهار نوع منبع اصلی وجود دارد که بازیکن باید آن‌ها را تولید، ذخیره و مدیریت کند. هر منبع نقش مشخصی در اقتصاد امپراتوری دارد و کمبود هر کدام می‌تواند روند بازی را مختل کند.

- **غذا (Food):** حیاتی‌ترین منبع بازی. هر یونیت زنده در هر Turn مقدار مشخصی غذا مصرف می‌کند. در صورت کمبود غذا، بازی وارد فاز بحران (Starvation) می‌شود. غذا از طریق مزرعه (Farm) روی هکس‌های دارای گندم یا برنج و طویله (Stable) روی هکس‌های دارای حیوانات اهلی تولید می‌شود.
- **چوب (Wood):** منبع پایه ساخت‌وساز. برای احداث اکثر ساختمان‌ها و برخی آپگریدها نیاز است. از طریق چوب‌بری (Lumber Mill) روی هکس‌های جنگلی تولید می‌شود.
- **سنگ (Stone):** منبع ساخت‌وساز پیشرفته‌تر. برای احداث سازه‌های سنگین‌تر مثل شهر/روستا و برخی آپگریدها نیاز است. از طریق معدن سنگ (Stone Mine) روی هکس‌های کوهستانی تولید می‌شود.
- **آهن (Iron):** نادرترین و ارزشمندترین منبع. برای آپگریدهای پیشرفته و ساخت سازه‌های سطح بالا نیاز است. از طریق معدن آهن (Iron Mine) روی هکس‌های کوهستانی دارای آهن تولید می‌شود.

همه منابع در انبار Town Hall ذخیره می‌شوند و ظرفیت انبار محدود است. با ارتقای انبار از طریق آپگریدهای شهر می‌توان این ظرفیت را افزایش داد.



مکانیزم تولید منابع (Resource Production)

برای اینکه یک کارگر شروع به تولید منبع کند، باید در تاسیسات که قبلاً توسط بیلدرها ساخته شدند (مانند معدن یا چوب‌بری) **مستقر (Station)** شود و تا زمانی که استقرار صورت نگیرد، تولیدی آغاز نخواهد شد. تولید منابع در بازی بر اساس ترکیب دو عامل محاسبه می‌شود: **نوع ساختمان** و **تعداد کارگرهای مستقر در آن**. هر ساختمان به تنهایی و بدون کارگر هیچ منبعی تولید نمی‌کند.

نرخ پایه تولید (Base Production Rate)

هر کارگر مستقر در یک ساختمان استخراجی، در هر Turn مقدار مشخصی از منبع آن ساختمان تولید می‌کند. این مقدار برای هر نوع ساختمان ثابت است و شما باید آن را به‌صورت منطقی تعیین کنید. به‌عنوان مثال:

- چوب‌بری: هر کارگر X واحد چوب در هر Turn
- معدن سنگ: هر کارگر X واحد سنگ در هر Turn
- مزرعه: هر کارگر X واحد غذا در هر Turn

تولید کل هر ساختمان برابر است با: **تعداد کارگران مستقر × نرخ پایه تولید** (بدیهی است که تعداد کارگرهای مستقر در یک سازه محدود است)

ظرفیت کارگر هر ساختمان

هر ساختمان حداکثر تعداد مشخصی کارگر می‌پذیرد. این سقف اولیه است و مستقر کردن کارگر بیش از ظرفیت مجاز ممکن نیست.

منابع اولیه و شروع بازی

هر بازیکن بازی را با یک Town Hall و ۵ یونیت آغاز می‌کند:

- ۲ یونیت Builder
- ۲ یونیت کارگر (Worker)
- ۱ یونیت Explorer

Town Hall به‌صورت پیش‌فرض دارای تولید پایه مشخصی از منابع (غذا و Production) است. ذخایر اولیه شامل مقادیر کمی از غذا، چوب و سنگ نیز در انبار موجود است. این مقادیر را خودتان باید طوری تعیین کنید که روند بازی منطقی جلو برود.

سقف تعداد یونیت‌ها (Unit Cap)

Unit Cap: حداکثر تعداد یونیت‌های فعال قابل داشتن است. این مکانیزم برای جلوگیری از Spamming Unit و ترغیب به مدیریت اقتصادی طراحی شده است.

- سقف اولیه: مقداری که خودتان به‌صورت منطقی تعیین می‌کنید.
- افزایش سقف: با ساخت ساختمان‌های جمعیتی (شهرک یا روستا).

سیستم اکشن‌پوینت (Action Point)

هسته اصلی مدیریت نوبت‌ها (Turn-based) بر پایه سیستم AP (Action Points) استوار است. مدیریت بهینه AP برای پیشرفت در بازی الزامی است.

مکانیزم AP برای هر یونیت

سیستم AP **مختص به هر یونیت (Unit-Specific)** است، نه سراسری.

- تخصیص: در ابتدای هر Turn، به هر یونیت زنده مقدار مشخصی AP اختصاص می‌یابد.
- مقدار ثابت: ظرفیت AP برای هر نوع یونیت ثابت است تا ارتقا نیابد و برای هر نوع یونیت متفاوت است (به عهده خود شما)
- چرخه مصرف: پس از اتمام AP یک یونیت، آن یونیت در آن Turn دیگر نمی‌تواند کاری انجام دهد. بازیکن می‌تواند نوبت خود را هر زمان تمام کند یا Skip کند.
- **نکته:** وقتی کارگر از ساختمان خارج می‌شه، AP اش همون مقدار کاهش‌یافته باقی بمونه تا آخر Turn. یعنی خروج از ساختمان AP رو بازنمی‌گردونه.

سیستم اقتصاد، مصرف و نگهداری

مصرف روزمره یونیت‌ها

هر یونیت زنده در هر Turn مقدار مشخصی غذا مصرف می‌کند. این مکانیزم بازیکن را مجبور به ساخت زیرساخت‌های تأمین غذا (مزارع) می‌کند و از تولید بی‌نهایت یونیت جلوگیری می‌کند.

بحران قحطی (Starvation)

اگر برآیند تولید و مصرف غذا منفی شود و ذخیره غذا کمتر از ۰ باشد، بازی وارد فاز بحران می‌شود:

- رشد جمعیت شهر کاملاً متوقف می‌شود.
- یونیت‌ها به‌دلیل ضعف مقداری از Action Point خود را از دست می‌دهند.

هزینه نگهداری ساختمان‌ها (Building Upkeep)

علاوه بر هزینه اولیه ساخت، هر ساختمان فعال در هر Turn مقدار کمی از منبعی که با آن ساخته شده مصرف می‌کند (به عهده خود شما) تا از اسپم شدن ساختمان‌ها جلوگیری شود. اگر 3 ترن پشت سر هم به اندازه کافی منابع برای نگهداری از ساختمان ها نداشته باشید، ساختمان خراب شده و دیگر قابل استفاده نیست و باید از اول بیلدر به آن هکس بفرستید تا از اول آن سازه را بسازد.

ساختمان‌ها و زیرساخت‌ها

قوانین کلی ساخت‌وساز

- محدودیت مرزی: هر سازه باید روی یک هکس داخل مرزهای اختصاصی بازیکن احداث شود.
- Upkeep: هر ساختمان فعال در هر Turn مقدار کمی منبع مصرف می‌کند.
- تناسب با هکس: ساختمان استخراجی باید روی هکسی ساخته شود که دارای آن منبع خاص است.

Town Hall (ساختمان اصلی)

- محل اصلی تولید یونیت‌ها و انجام آپگریدها.
- Safeguard: با نرخ بسیار کم، منابع پایه (چوب +۱، غذا +۱) در هر ترن تولید می‌کند تا بازیکن در صورت اشتباه محاسباتی کاملاً بی‌منبع نشود.

ساختمان‌ها:

- چوب‌بری (Lumber Mill): فقط روی هکس‌های جنگلی.
- معدن سنگ (Stone Mine): هکس‌های دارای سنگ. پیش‌نیاز: چوب.
- معدن آهن (Iron Mine): هکس‌های دارای آهن. پیش‌نیاز: چوب.
- مزرعه (Farm): هکس‌های دارای گندم یا برنج – منبع اصلی غذا.
- شهرک: برای افزایش Unit Cap. نیازمند سنگ و آهن و چوب.
- طویله (Stable): برای استفاده از حیوانات روی هکس های دارای گاو و گوسفند - منبع دیگر غذا. پیش‌نیاز: چوب.

نکته مهم: صرف ساخت معادن و چوب‌بری‌ها کافی نیست – حتماً باید یونیت‌های کارگر در آن‌ها مستقر (Station) شوند تا تولید آغاز شود.

هزینه اعمال مختلف

هزینه حرکت (Movement Cost): بر اساس نوع زمین (Terrain Cost) محاسبه می‌شود:

- دشت: ۱ AP
- جنگل: ۲ AP
- کوهستان: ۴ AP

(می‌تونید به انتخاب خودتون مقادیر رو عوض کنید و اجباری نیست)

این مقادیر صرفاً مثال هستند؛ می‌توانید آن‌ها را تغییر دهید اما روند بازی باید منطقی باشد.

هزینه ساخت‌وساز: بسته به نوع سازه متغیر است. مثال: چوب‌بری ۱ AP، سازه‌های پیچیده‌تر ۲ AP. هزینه ساخت شهرک و روستا باید بیشتر از بقیه سازه ها باشد.

سیستم شارژ بیلدرها و Instant Build

ساخت درجا (Instant Build): به محض صدور دستور ساخت و کسر هزینه‌ها (منابع + AP)، ساختمان فوراً روی هکس ظاهر می‌شود.

متغیر Charge: هر Builder تعداد مشخصی شارژ دارد (مثلاً ۳). با هر ساخت، ۱ شارژ کم می‌شود. وقتی شارژ به صفر رسید، Builder از نقشه حذف (Consume) می‌شود.

سیستم جابجایی و نقشه

حرکت در هکس‌ها

هر یونیت برای رفتن از یک هکس به هکس مجاور، نیازمند صرف AP است. جابجایی در نقشه شش‌ضلعی از قوانین بازی‌های نوبتی پیروی می‌کند.

جابجایی یونیت‌ها بین هکس‌ها باید به صورت انیمیشن نمایش داده شود. یونیت نباید به صورت آنی از یک هکس به هکس دیگر teleport کند، بلکه باید حرکت آن به صورت روان و قابل مشاهده روی نقشه اتفاق بیفتد. (دایره ساده برای نشون دادن یونیت هم کفایت می‌کند.)

هزینه حرکت و نوع زمین

هر هکس بسته به Terrain خود، جریمه حرکتی متفاوتی اعمال می‌کند تا مسیرهای مختلف ارزش استراتژیک متفاوتی داشته باشند.

بزرگ‌نمایی و جابجایی نقشه

بازیکن باید بتواند نقشه را بزرگ‌نمایی و کوچک‌نمایی کند. زوم باید گسسته (Discrete) باشد – یعنی در سطوح مشخصی جابجا شود. همچنین باید بتواند دوربین را روی نقشه جابجا کند (Pan). نحوه پیاده‌سازی این دو مورد (کلید، موس، یا هر روش دیگری) به سلیقه خودتان است.

سیستم مه‌جنگ (Fog of War)

در ابتدای بازی، بیشتر نقشه تاریک و ناشناخته است.

- بازیکن تنها هکس‌هایی را می‌بیند که یونیت‌هایش یا Town Hall و یا ساختمان های دیگرش در آن قرار دارند.
- هر یونیت و یا ساختمان یک شعاع دید (Vision Radius) دارد. با حرکت در نقشه، هکس‌های تاریک روشن شده و نوع زمین و منابع آشکار می‌شوند.

معرفی یونیت‌ها

Explorer (اکسپلورر)

- نقش: کاوش نقشه و کنار زدن Fog of War.
- عملکرد: پیدا کردن منابع استراتژیک (معادن، زمین‌های کشاورزی).

- بیشترین AP و شعاع دید را در بین تمام یونیت‌ها دارد. هر هکسی که از آن عبور کند یا در شعاع دیدش باشد برای همیشه از Fog of War خارج می‌شود.

Builder (بیلدر)

- نقش: احداث ساختمان‌ها و زیرساخت‌ها.
- عملکرد: ساخت چوب‌بری، مزرعه، معادن، دامداری و ... روی هکس‌های مناسب، با مصرف منابع و AP.
- مکانیزم Consume/Charges: هر Builder تعداد مشخصی شارژ دارد. با رسیدن شارژ به صفر، یونیت از بین می‌رود.

Worker (کارگر)

- نقش: تولید و جمع‌آوری منابع.
- عملکرد: کارگرها توانایی ساخت ندارند. باید در تاسیسات که Builder ساخته مستقر (Station) شوند تا تولید آغاز شود.
- معادن، چوب بری، مزارع و اصطبل ها نیاز به کارگر دارند تا بتوان از منابع آنها استفاده کرد(همانطور که گفته شد ورود کارگر به سازه ها پس از ساخت ساختمان در آن هکس است). مستقر شدن هر یونیت کارگر در یک سازه از سقف AP آن یونیت در آن ترن کم میکند و پس از اتمام کار یا خارج شدن آن AP دوباره برمیگردد.
- پس از اتمام کار در سازه به هر دلیلی (تمام شدن منابع آن هکس یا دلایل دیگر ...) در آن هکس بیکار میمانند تا کار دیگری به آنها اختصاص داده شود.

Border Expander (مرزگشا)

- نقش: توسعه قلمرو.
- مکانیزم: یک هکس انتخابی به‌همراه تمام ۶ هکس همسایه‌اش را به مرزهای امپراتوری اضافه می‌کند. برای اضافه کردن هر هکس به نقشه باید از قبل توسط اکسپلورر قبلا اکسپلور شده باشد.
- مصرف: پس از استفاده از قابلیت تصرف، این یونیت کاملاً حذف (Consume) می‌شود.

جریان کلی بازی (Game Flow)

بازی به صورت نوبتی (Turn-based) و تک‌نفره پیش می‌رود. در هر نوبت، بازیکن با مدیریت یونیت‌ها و منابع خود تلاش می‌کند امپراتوری‌اش را گسترش دهد.

شروع بازی

پس از زدن دکمه Start، بازیکن با یک Town Hall در مرکز نقشه، ۵ یونیت پیش‌فرض و مقدار مشخصی از منابع اولیه وارد بازی می‌شود. بیشتر نقشه در ابتدا تاریک (Fog of War) است و تنها هکس‌های اطراف Town Hall قابل مشاهده‌اند.

ساختار هر نوبت (Turn Structure)

بازیکن پس از انجام اقدامات دلخواه، دکمه «پایان نوبت» (End Turn) را می‌زند. در این لحظه چرخه زیر به ترتیب اجرا می‌شود:

- **تجدید AP :** APهای همه یونیت‌های زنده برای نوبت جدید تجدید می‌شوند و نوبت رسما آغاز می‌شود.
- **تولید منابع:** ساختمان‌های فعال منابع خود را تولید کرده و به انبار اضافه می‌کنند. همزمان، صف تولید Town Hall (یونیت یا آپگرید در حال ساخت) یک مرحله به جلو پیش می‌رود. در صورت اتمام تولید، یونیت یا آپگرید مربوطه فعال می‌شود.
- **کسر Upkeep:** هزینه نگهداری یونیت‌ها و ساختمان‌های فعال از انبار کسر می‌شود. اگر منابع کافی نباشد، بازی وارد فاز بحران می‌شود.
- **اقدامات بازیکن:** بازیکن می‌تواند با استفاده از AP یونیت‌ها هر ترکیبی از اقدامات زیر را انجام دهد: حرکت در نقشه، ساخت ساختمان توسط Builder، استقرار کارگر در سازه، استفاده از Border Expander برای گسترش مرز، تولید یونیت جدید از Town Hall، یا شروع یک آپگرید. پس از اتمام اقدامات، بازیکن دوباره دکمه End Turn را می‌زند و چرخه از ابتدا تکرار می‌شود.

قوس کلی بازی (Progression Arc)

یک بازی معمول این‌گونه پیش می‌رود: بازیکن در نوبت‌های اولیه با Explorer نقشه را کشف می‌کند تا منابع و زمین‌های مناسب را پیدا کند. سپس با Builder زیرساخت‌های تولیدی (چوب‌بری، معدن، مزرعه) می‌سازد و کارگرها را در آن‌ها مستقر می‌کند. با تجمیع منابع، آپگریدهای Town Hall را فعال می‌کند، سقف یونیت را بالا می‌برد و با Border Expander قلمرو خود را گسترش می‌دهد. هدف کلی در این فاز، ساختن یک اقتصاد پایدار و گسترش کنترل‌شده امپراتوری است.

پایان بازی

در این فاز پروژه، بازی پایان مشخصی ندارد و به صورت آزاد (Sandbox) ادامه می‌یابد. بازیکن تا زمانی که بخواهد می‌تواند بازی کند. ارزیابی پروژه بر اساس صحت پیاده‌سازی مکانیزم‌ها خواهد بود، نه رسیدن به هدف خاصی در بازی.

رابط کاربری بازی (HUD)

در حین بازی، اطلاعات ضروری امپراتوری باید به‌صورت همیشه‌نمایان (HUD) در اختیار بازیکن باشد. هدف از این رابط این است که بازیکن بدون نیاز به باز کردن هیچ منوی اضافه‌ای، در هر لحظه تصویر کاملی از وضعیت بازی داشته باشد.

نمایش منابع

برای هر چهار منبع (غذا، چوب، سنگ، آهن) باید مقدار فعلی ذخیره و ظرفیت انبار به‌صورت همزمان نمایش داده شود (مثال: چوب ۱۲۰/۲۰۰). علاوه بر این، نرخ خالص تولید هر منبع در هر Turn نیز باید مشخص باشد تا بازیکن بداند در هر نوبت چقدر منبع به دست می‌آورد یا از دست می‌دهد (مثال: آهن ۱۲/۲۰۰ | +۳ در هر نوبت). در صورتی که نرخ خالص منفی باشد، این عدد باید به‌شکل واضحی (مثلاً با رنگ قرمز) به بازیکن هشدار دهد.

نمایش یونیت‌ها

تعداد یونیت‌های فعال فعلی در برابر سقف مجاز (Unit Cap) باید نمایش داده شود (مثال: یونیت‌ها ۵/۱۰). این اطلاعات به بازیکن کمک می‌کند بداند آیا می‌تواند یونیت جدید تولید کند یا نیاز به ساخت شهرک دارد. همچنین باید قابلیت این وجود داشته باشد که تعداد یونیت‌ها به تفکیک نوع آن‌ها وجود داشته باشد.

شماره نوبت

نوبت جاری بازی باید به‌وضوح نمایش داده شود (مثال: نوبت ۱۴). این اطلاعات برای درک پیشرفت در صف تولید Town Hall و برنامه‌ریزی ضروری است.

صف تولید Town Hall

وضعیت فعلی صف تولید Town Hall باید قابل مشاهده باشد — چه چیزی در حال ساخت است، چند Turn تا اتمام مانده (مثال: در حال ساخت Explorer — ۳ نوبت مانده).

دکمه پایان نوبت (End Turn)

دکمه End Turn باید در جای ثابت و به‌راحتی قابل دسترس باشد. در صورتی که یونیت‌های بی‌کار (Idle) وجود داشته باشند که هنوز AP دارند، قبل از پایان نوبت باید یک هشدار یا اعلان به بازیکن نشان داده شود تا مطمئن شود نیرویی را هدر نمی‌دهد.

هشدار بحران

در صورت ورود به فاز قحطی (Starvation) یا هر بحران منبعی دیگر، یک اعلان واضح در HUD نمایش داده می‌شود تا بازیکن از وضعیت اضطراری آگاه شود.

جلوگیری از اقدامات نامعتبر

رابط کاربری باید طوری طراحی شود که بازیکن اصلا نتواند کار اشتباهی انجام دهد. به جای نمایش خطا بعد از اقدام، گزینه‌های غیرمجاز از همان ابتدا غیرفعال یا قفل نمایش داده شوند. به عنوان مثال، اگر AP کافی وجود نداشته باشد دکمه حرکت غیرفعال باشد یا اگر منابع کافی نباشد.

آپگریدهای شهر

هر شهر توانایی ارتقای تکنولوژی و برخی خاصیت‌های سازه‌ها را دارد. در فاز یک فقط امکان ارتقا town hall وجود دارد و تاثیر آن فقط افزایش انبارهای شهر است. تکنولوژی‌ها امکان انجام کاری را آنلاک کرده و یا قابلیتی را فعال خواهند کرد.

آیتم	توضیحات
ارتقا انبار ۱ و ۲	پس از اتمام، میزان فضای انبارهای شهر افزایش می‌یابد تا منابعی مثل چوب، سنگ و آهن بیشتر نگه‌داری شوند.
تکنولوژی معدن سنگ	قابلیت ساخت معدن سنگ داخل هکس‌های دارای سنگ توسط Builder باز می‌شود.
تکنولوژی معدن آهن	قابلیت ساخت معدن آهن داخل هکس‌های دارای آهن باز می‌شود. (پس از تکنولوژی معدن سنگ).
تکنولوژی ابزار آلات حرفه‌ای	میزان سنگ و آهن استخراج‌شده در هر Turn به ازای هر کارگر ۱.۵ برابر می‌شود. (پس از تکنولوژی معدن سنگ و آهن)
تکنولوژی ساخت شهرک	یکی از زمین‌های خالی داخل مرز به شهرک تبدیل می‌شود و Unit Cap افزایش می‌یابد. شهرک ها فقط روی زمین های فاقد منابع ساخته میشوند یعنی سبزه زار های فاقد برنج و گندم یا کوهستان بدون معدن یا دشت بدون حیوان و ...

ارتقا انبار دو بار انجام‌پذیر است و هر بار فضای انبار را افزایش می‌دهد.

قابلیت‌های امتیازی

- **Auto-Explore برای Explorer:** در حالت عادی، بازیکن باید به‌صورت دستی به Explorer دستور حرکت بدهد. در این قابلیت امتیازی، Explorer می‌تواند به حالت Auto-Explore سوئیچ کند و بدون نیاز به دستور دستی، به‌صورت هوشمند نقشه را کاوش کند. الگوریتم Auto-Explore باید به‌گونه‌ای طراحی شود که مناطق ناشناخته را اولویت‌بندی کند و از رفتن به مناطق قبلاً کشف‌شده تا حد امکان پرهیز کند. بازیکن باید بتواند در هر زمان Auto-Explore را غیرفعال کند و کنترل دستی را بازپس بگیرد. همچنین Explorer در حالت Auto-Explore نباید از مرزهای کشف‌نشده‌ای که AP کافی برای بازگشت ندارد عبور کند.
- **Minimap**: یک نقشه کوچک در گوشه‌ای از صفحه نمایش داده می‌شود که وضعیت کلی نقشه را به بازیکن نشان می‌دهد. Minimap باید موارد زیر را نمایش دهد: مناطق کشف‌شده و ناشناخته، موقعیت Town Hall و شهرک‌ها، موقعیت فعلی همه یونیت‌های فعال، و مرزهای فعلی امپراتوری. بازیکن باید بتواند با کلیک روی Minimap، دوربین اصلی بازی را به آن نقطه از نقشه منتقل کند.
- **گرافیک و UI بهتر**: هر بهبود بصری که تجربه بازی را ارتقا دهد امتیاز دارد. نمونه‌هایی که ارزش بالایی دارند: انیمیشن برای حرکت یونیت‌ها، عکس و نمادها برای نمایش ساختمان یونیت و غیره، افکت‌های بصری برای تولید منابع یا ساخت ساختمان، نمایش اطلاعات با tooltip هنگام hover روی یونیت‌ها یا ساختمان‌ها، و طراحی UI منسجم و حرفه‌ای برای تمام پنل‌ها و منوها.

- **Procedural Map Generator:** به جای اینکه مپ را به صورت دستی (Hardcode) بسازید یا از هکس‌های رندوم نامنظم استفاده کنید، مپ را با یک الگوریتم منطقی مشخص کنید. ایده: مپی که تولید می‌شود دارای رشته کوه‌های به هم پیوسته و یا جنگل‌های متراکم باشد. همچنین جهت دسترسی به ریسورس‌های طبیعی فاصله هکس جنگل از Town Hall نباید بیشتر از ۲ هکس باشد تا از عدم دسترسی به ریسورس‌ها و غیرمنطقی بودن گیم جلوگیری شود.
- **استفاده از Build Tool (میون و گریدل) :** پروژه شما باید شامل یک کانفیگ کامل از ابزارهای ،Gradle Maven باشد به طوری که فقط با استفاده از سورس پروژه بتوان آن را بیلد و اجرا کرد. به این معنا که کتابخانه‌های مورد استفاده در پروژه به طور خودکار دانلود و استفاده شوند. استفاده از ابزارهای مشابه مثل Apache Ant بلامانع است. همچنین در اینجا شما می‌توانید قسمت امتیازی این بخش را پیاده‌سازی کنید.

سخن اخر (نکات تکمیلی)

در طراحی این پروژه، تمایز مشخصی بین دو دسته تصمیم وجود دارد: تصمیم‌هایی که **ثابت و اجباری** هستند، و تصمیم‌هایی که **به خلاقیت و قضاوت شما** واگذار شده‌اند:

- **موارد ثابت** — مکانیزم‌های اصلی بازی که باید دقیقاً طبق داک پیاده‌سازی شوند: قوانین حرکت، سیستم AP، مکانیزم شارژ Builder، نحوه عملکرد Border Expander، ترتیب اجرای Turn، سیستم Starvation، ترتیب آنلاک شدن آپگریدها و منابع مورد نیاز آنها و ...
- **موارد آزاد** — مقادیر و جزئیاتی که شما باید با توجه به منطق بازی و خلاقیت خود تعیین کنید: هزینه منابع برای ساخت هر سازه، هزینه تولید هر یونیت، نرخ تولید منابع به ازای هر کارگر، هزینه AP هر اقدام، مقادیر مورد نیاز برای هر آپگرید، ظرفیت اولیه انبار، سقف اولیه یونیت، و رابط کاربری و نحوه تعامل بازیکن با یونیت‌ها و ساختمان‌ها (مثلاً چطور یونیت انتخاب می‌شود، چطور کارگر به معدن فرستاده می‌شود، منوها چه شکلی هستند). تنها معیار برای موارد آزاد این است که **روند بازی منطقی و قابل تجربه باشد**. مقادیر نامعقول — چه خیلی کم چه خیلی زیاد — مستقیماً در نمره‌دهی تأثیر منفی خواهند داشت.

استفاده از ابزار Version control مانند **Git** اجباری است و موارد زیر را رعایت کنید:

- یک ریپازیتوری **Private** داشته باشید (تاکید میشود که private باشد!)
- کامیت‌های منظم از ابتدا تا انتهای پروژه داشته باشین.
- پیام‌های commit باید معنادار و توصیفی باشند. پیام‌هایی مثل "fix" یا "gg" قابل قبول نیستند.
- فایل‌های غیرضروری (مثل فایل‌های build، کش IDE و...) نباید در ریپازیتوری قرار بگیرند. استفاده از **gitignore.** الزامی است.

نحوه انتخاب یونیت‌ها، فرستادن آن‌ها به معادن و چوب‌بری‌ها، باز کردن منوی Town Hall برای تولید یونیت یا فعال‌سازی آپگریدها، و **به‌طور کلی هر تعاملی که بازیکن با محیط بازی دارد**، کاملاً به سلیقه و خلاقیت شما واگذار شده است. مثلا اینکه انتخاب یونیت با کلیک چپ باشد یا راست، منوی ساخت به‌صورت پاپ‌آپ باز شود یا پنل کناری، یا اینکه آپگریدها از طریق کلیک روی Town Hall فعال شوند یا منوی جداگانه‌ای داشته باشند – همه اینها تصمیم شماست. تنها معیار ارزیابی این است که مکانیزم‌های اصلی بازی درست کار کنند و هیچ نمره‌ای کسر نمی‌شود. طراحی خلاقانه و روان در این بخش می‌تواند نمره امتیازی هم داشته باشد.

با توجه به ماهیت استراتژیک بازی، استفاده از شی‌گرایی به تنهایی کافی نیست. اکیداً توصیه می‌شود پیش از شروع کدنویسی، **معماری MVC** را مطالعه و به دقت در پروژه اعمال کنید – اگرچه در این فاز برای آن نمره‌ای لحاظ نشده. جداسازیِ «Model» از «View» در فازهای بعدی الزامی‌ست و از همین ابتدا کدتان را بسیار تمیزتر نگه می‌دارد. همچنین مطالعه اصول **SOLID** اکیداً توصیه می‌شود.

همانطور که احتمالاً می‌دانید، فرآیند تحویل تمام تمرینات و بالاخص پروژه با رویه‌ای به نام "ارزیابی" همراه است که در آن از شما انتظار می‌رود به تمام قسمت‌های کد خود تسلط کامل داشته باشید و بتوانید در حین تحویل، قسمت‌هایی از کد خود را به صورت جزئی یا کلی تغییر دهید. در غیر این صورت، تشخیص بر تقلب خواهد بود و مستقل از پیاده‌سازی شما و کامل بودن مابقی کد، **نمره بخش مربوطه صفر خواهد شد**. لذا خواهشمندیم از هرگونه تقلب پرهیز کنید. استفاده از ابزارهای هوش مصنوعی و کپی کردن کد از منابع اینترنتی بلامانع است اما همچنین از شما انتظار می‌رود به عملکرد هر قسمتی از کد خود کاملاً تسلط داشته باشید تا بتوانید فرآیند ارزیابی را با موفقیت پشت سر بگذارید.

تیم طراحی: روژان کریمقاسمی، حسام توکلی، پردیا رضوی، مهدی مشایخی، پرهام امیری و بهراد سلیمانی. (اسامی به اکانت بله لینک شده‌اند.)

مسئول پروژه: ماهان قره‌گوزلو

با آرزوی روزهای بهتر. تیم درس برنامه نویسی پیشرفته 🙂 دی: