

به نام خدا

کلیات فاز اول پروژه برنامه‌سازی پیشرفته

قبل از خواندن صورت پروژه، بازی Farm Frenzy 3 را ببینید، لینک بازی روی کانال گذاشته شده است. این بازی مبنای پروژه شما خواهد بود. بدون توضیح اضافه و با فرض این که شما این بازی را دیده‌اید، می‌رویم به سراغ آیتم‌هایی که باید پیاده‌سازی کنید:

۱. حیوانات

۱.۱. حیوانات تولیدکننده

- علف روی زمین را می‌خورد و در ازای آن کالایی تولید می‌کند، در صورت نبودن علف می‌میرد
- ۱،۱،۱. مرغ = تخم مرغ تولید می‌کند
- ۱،۱،۲. گوسفند = پشم تولید می‌کند
- ۱،۱،۳. گاو = شیر تولید می‌کند

۱.۲. حیوانات وحشی

به صورت تصادفی هر یک دقیقه، دو تا از آن‌ها روی زمین بازی می‌افتد و در صورت برخورد با حیوانات یا محصولات روی زمین مانده آن‌ها را از بین می‌برد، کاربر می‌تواند آن را زندانی و به انبار منتقل کند

۱،۲،۱. شیر

۱،۲،۲. خرس

۱،۳. سگ

به محض ورود یک حیوان وحشی با سرعت مشخصی به سمت آن می‌رود و وقتی به آن رسید هر دو می‌میرند

۱،۴. گربه

کالا‌های روی زمین را جمع می‌کند

۲. حمل و نقل

۲،۱. وانت: برای فروش کالا‌های موجود در انبار

۲،۲. بالگرد: برای خرید کالا

۳. آبیاری

سیستم آبیاری مانند بازی مبتنی بر چاهی است که پر کردن آن هزینه دارد و پس از پر شدن به اندازه چند بار علف سبز می‌کند

۴. کارگاه‌ها

هر زمین بازی ظرفیت ۶ کارگاه را دارد، کارگاه تعدادی ورودی و یک خروجی دارد، مثلاً ساده‌ترین کارگاه که در مراحل اولیه بازی می‌بینید ورودی اش تخم مرغ و خروجی اش نوعی آرد است. البته نیازی به پیاده‌سازی همه‌ی کارگاه‌های بازی نیست، فقط تعداد محدودی از آن‌ها که در API خواهد آمد را پیاده‌سازی کنید، همچنین در این بخش کاربر می‌تواند قبل از شروع بازی کارگاه Custom تعریف کند به این معنا که در فایلی مشخصات کارگاه دلخواه شامل نام، ورودی(ها) و خروجی آن را می‌نویسد و آن کارگاه هم به بازی اضافه می‌شود.

۵. انبار

۶. مرحله

هر مرحله شامل تعدادی خواسته است، مثلاً رسیدن به ۶ تخم مرغ و ۱۰۰۰ سکه. در صورتی که کاربر به همه‌ی خواسته‌ها را تامین کند در مرحله برنده می‌شود. می‌توان مرحله را به صورت Custom در فایل‌های Json تعریف کرد.

همچنین کاربر باید امکان Save و Load بازی با استفاده از استاندارد Json را داشته باشد. در مورد نحوه انجام آن با مسئول حل تمرین صحبت کنید.

رابط کاربری این فاز Command Line است به این صورت که مانند تمرین دوم دستوراتی داده می شود و بازی بر مبنای آن انجام می شود، در فاز بعدی گرافیک اضافه خواهد شد.

بر طراحی های مربوط به زمان پیشنهاد می شود یک واحد زمانی مرجع در نظر بگیرید و همه ی کارهایی که با زمان مرتبط می شوند (مثل تولید محصولات حیوانات) را بر اساس آن انجام دهید. مثلاً می توانید در این فاز واحد مرجع را ۱ ثانیه در نظر بگیرید و هر کاری دارای Interval مخصوص خود بر اساس واحد زمانی مرجع باشد. زمان بازی هم با یک دستور خاص مثلاً 1 turn جلو می رود، یعنی هرگاه کاربر دستور مربوط را وارد کرد همه یک واحد زمانی جلو می روند.

برای این فاز جلسه ی رفع اشکال و طراحی در نظر گرفته خواهد شد، همچنین یک پیوست API دارد که رابط کاربری شما بر مبنای آن پیاده سازی می شود و جزئیات آپگریدها را نیز در بر دارد. در صورت وجود هر سوالی می توانید در پیام رسان یا کلاس طراحی بپرسید.

تحويل دادنی ها:

در فاز صفر:

- نمودار UML
- لیست تسک ها و شامل عنوان تسک، مسئول پیاده سازی تسک و تخمین زمانی هر یک
- رابطه پیشنیازی تسک ها
- زمان بندی پیشنهادی برای فاز یک

در فاز یک:

- کد پیاده سازی شده روی github
- نسخه برنامه جاوا که کار می کند

نکاتی در مورد کار تیمی:

- مسئول پیاده سازی تسک، وظیفه ی تست و دیباگ آن بخش را نیز به عهده بگیرد، یعنی تقسیم بندی شما بر اساس موضوعات مورد پیاده سازی باشد.
 - مثال درست: پیاده سازی، تست و دیباگ وانت با نفر اول
 - مثال غلط: کل پیاده سازی با نفر اول، کل دیباگ با نفر دوم
- به زمان بندی متعهد باشید و مسئول هر تسک بر اساس دو برابر میزان تخمین خود برای آن تسک زمان در نظر بگیرد، زمان قابل توجهی را هم برای وصل کردن بخش های آماده شده به هم در نظر بگیرید.
- زمان بندی شما به گونه ای باشد که چند روز زودتر از مهلت اعلام شده پروژه خود را تمام کنید تا مهلت برای تست کلی فاز اول داشته باشید، همچنین برای اتفاقات پیش بینی نشده نیز زمانی را در نظر بگیرید که مشکلات احتمالی باعث دیر شدن نشود.
- تلاش کنید همه ی اعضای تیم به اندازه معقولی درگیر طراحی و پیاده سازی شوند، تلاش کنید علاوه بر مهارت فنی، مهارت کار تیمی خود را نیز رشد دهید، بزرگترین اشتباه این است که یکی از اعضای پروژه بیشتر کارها را انجام

- دهد و بقیه در جریان قرار نگیرند، در این صورت همه‌ی اعضای تیم متضرر می‌شوند زیرا کار تیمی، تقسیم‌بندی درست و زمان‌بندی و تعامل مناسب را یاد نگرفته‌اند و همچنین از نظر مهارت فنی هم رشد چندانی نمی‌کنند.
- می‌توانید از پشتیبان خود برای حل چالش‌های فنی و همچنین چالش‌های مربوط به نحوه کار تیمی و تقسیم‌بندی وظایف کمک بگیرید.

پیوست API

buy [animal_name]

خرید یک حیوان غیر وحشی

pickup [x y]

برداشتن کالاهای تولید شده‌ی موجود در یک خانه

cage [x y]

زندانی کردن حیوانات وحشی موجود در یک خانه

نیازی به پیاده‌سازی فرار کردن حیوان پس از زندانی شدن نیست

plant [x y]

ایجاد یک دسته علف با مشخص کردن مختصات مرکز و به شعاع ۱ خانه همسایه یعنی در نهایت یک مربع کامل ۳*۳ پر از سبزه می‌شود

well

پر کردن چاه

start [workshop_name]

معادل کلیک کردن روی کارگاه

در صورت نبود مواد اولیه مورد نیاز یا در حال کار بودن کارگاه شروع نشود و پیغام مناسب بدهد

upgrade [workshop_name | cat | well | truck | helicopter | warehouse]

افزایش سطح

در صورت نبود پول مورد نیاز جلوی آن گرفته شود

منظور از افزایش سطح گربه باهوش شدن آن است، همان طور که در کلاس گفته شد گربه فقط دو سطح دارد و ارتقاء آن همه ی گربه های صفحه را ارتقاء می دهد

load custom [path_to_custom_directory]

شامل نقشه ها و کارگاه ها

آدرس پوشه ای را می گیرد که در آن تمام فایل های کارگاه های کاستوم و نقشه ها وجود دارد

run [map_name]

بازی را روی یکی از نقشه های لود شده شروع می کند

save game [path_to_json_file]

وسط بازی کل اطلاعات بازی در حال اجرا را ذخیره می کند

load game [path_to_json_file]

بر اساس فایل تولید شده در زمان ذخیره، بازی را لود می کند و کاربر می تواند از همان وضعیتی که ذخیره کرده بود ادامه دهد

print [info|map|levels|warehouse|well|workshops|truck|helicopter]

print info شامل پول، زمان گذشته، شرایط تمام شدن مرحله و میزان تامین شده ی هر کدام

print map چاپ نقشه و مشخص کردن خانه های پر و خالی (برای دیباگ تا قبل از اضافه شدن گرافیک)

در مورد فرمت خروجی دست شما باز است و می توانید به دلخواه خروجی مورد نظر را چاپ کنید اما لازم است تمام دستورات بالا را داشته باشید و اطلاعات مرتبط را چاپ کنید

turn [n]

زمان بازی را n واحد جلو می برد

کارگاه های پیش فرض بازی:



۶ مورد موجود در ردیف اول عکس را با نام خودشان پیاده سازی کنید، ورودی و خروجی این ۶ کارگاه را طبق نسخه خود بازی پیاده کنید.

نکات:

چیزی به نام Reward نداریم و آپگریدها بر اساس میزان پول و در حین مرحله انجام می‌شوند، همچنین برای سادگی آپگرید سگ و فقس نیز نداریم. آخرین سطح آپگرید چاه نیز از صورت پروژه حذف است چون مکانیزم عملکرد چاه اتوماتیک می‌شود که با مکانیزم سطوح قبلی متفاوت است.

تفاوت گربه معمولی (سطح ۱) با گربه زیرک (سطح ۲) در الگوریتم انتخاب آیتم بعدی می‌باشد که در کلاس توضیح داده شد، گربه معمولی آیتم بعدی را به صورت تصادفی انتخاب می‌کند اما گربه باهوش آیتم بعدی را بر اساس نزدیکی به خود انتخاب می‌کند.

اصلاحات:

```
[truck|helicopter] add [item_name] [count]
```

محصولات را در لیست فروش یا خرید قرار می‌دهد (مانند بازی لیست اجناس ماشین یا بالگرد را پر می‌کند)

```
[helicopter|truck] clear
```

لیست اجناس ماشین یا بالگرد را خالی می‌کند

[helicopter|truck] go

ماشین یا بالگرد حرکت می کند

دقت کنید علامت براکت در ورودی اصلی وجود ندارد مثلاً منظورمان از دستور [x y] cage به این صورت می باشد:

cage 5 3

همچنین وجود علامت | در براکت به معنای «یا» می باشد، مثلاً دستور print [info|map|...] در چند سطر قبل به معنای آن است که دستورات زیر معتبر هستند:

print info

print map

سعی کردیم داک پروژه را خیلی جزئی نکنیم، بنابراین در بسیاری از قسمت های پروژه دست شما را باز گذاشته ایم تا مطابق خلاقیت خود پیش بروید. البته در مواردی که دست شما را باز گذاشته ایم قبل از پیاده سازی با pm گروهتان هماهنگ باشید.

در صورت ورود دستورات اشتباه با فرمت مناسبی خطا چاپ کنید.

همچنین می توانید از پشتیبان ها بخواهید کد های شما را در گیتهاب ببینند و روی pull request نظر بدهند تا کیفیت کد شما بهتر شود، در صورتی که تا کنون بازی farm frenzy را انجام نداده اید پیشنهاد می کنیم از لینک روی کانال دانلود کنید و بازی کنید تا بتوانید با دید کامل تری پروژه را طراحی کنید.