

به نام خدا



## درس برنامه‌سازی پیشرفته

فاز ۱

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۰۴ - ۰۳

---

استاد:

دکتر محمد امین فضلی

# فهرست مطالب

۲	قوانين و نکات قابل توجه
۳	معرفی پروژه
۳	اهداف پروژه
۳	شرح خلاصه‌ی بازی
۳	پیاده‌سازی
۵	کارهایی که در فاز ۰ باید انجام شوند
۵	ساخت مخزن گیت‌هاب
۵	ایجاد UML
۶	کارهایی که در فاز ۱ باید انجام شوند
۷	شرح پروژه
۷	مقدمه‌ای بر ابزارهای ساخت
۱۱	شروع بازی و منوها
۱۶	منوی بازی
۱۹	سیستم زمان و تاریخ
۲۱	سیستم آب و هوا
۲۳	نقشه‌ی بازی و حرکت
۲۶	انرژی و توانایی
۲۸	ابزارالات
۳۳	نحوه‌ی بازی و پیشرفت
۳۷	ساخت و ساز
۳۹	آشپزی
۴۱	دامداری
۴۴	فراوری
۴۵	معامله
۴۷	تعاملات و روابط با دیگر بازیکن‌ها
۵۱	NPC ها



## قوانین و نکات قابل توجه

- در روز تحویل حضوری مشارکت تمام اغصای 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 گیت‌هاپ پروژه بررسی می‌شود.
- در صورت کشف تقلب، برای بار اول منفی نمره آن فاز برای آن فرد ثبت می‌شود و برای بار دوم نمره منفی کل پروژه برای فرد لحاظ خواهد شد.



## معرفی پروژه

استاردو ولی (Stardew Valley) یک بازی نقش‌آفرینی و شبیه‌سازی مزرعه‌داری است که توسط ConcernedApe توسعه داده شده است. در این بازی، شما نقش یک کشاورز را بر عهده دارید که مزرعه‌ای قدیمی و فرسوده را احیا می‌کند.

### اهداف پروژه

- توصیه می‌شود قبل از شروع پروژه، با مکانیک‌های بازی آشنا شوید. می‌توانید از وبسایت رسمی بازی یا دیگر منابع برای یادگیری استفاده کنید.
- در این فاز از پروژه، طراحی شی‌گرا و تفکیک مناسب بخش‌های مختلف بازی اهمیت دارد.
- یکی از اهداف پروژه، آشنایی با design patterns و معماری سیستم‌های نرم‌افزاری مثل MVC (Model View Control) است.
- آشنایی با سیستم‌های مدیریت نسخه مانند Git و کار در بستر یک مخزن GitHub از نکات مهم این پروژه است. تغییرات باید در بازه‌های کوتاه‌مدت commit شوند.

### شرح خلاصه‌ی بازی

همان‌طور که گفته شد، این یک بازی مزرعه‌داری است، به این معنی که شما تکه‌ای زمین اختصاص داده می‌شود و شما می‌توانید در آن زمین محصولاتی را پرورش دهید، از گندم و جو گرفته تا میوه و دام. علاوه بر زمینی که مزرعه‌ی شما در آن قرار دارد، شما می‌توانید به شهر کوچک کنار مزرعه‌ی خود نیز سر برزید. در این شهر شما می‌توانید داد و ستد کنید، محصولات خود را بفروشید، لوازم پیشرفته‌تری برای مزرعه بخرید، با دیگر بازیکنان تعامل کنید و دوستی برقرار کنید و ...

### پیاده‌سازی

بخش‌هایی که قرار است در این پروژه پیاده‌سازی کنید به طور مختصر عنوان شده‌اند، جلوتر می‌توانید جزئیات هر کدام را مشاهده کنید.

### ورود و شروع بازی

قبل از وارد خود بازی شدن، نیاز است تا برای خود حساب کاربری بسازید، سپس برای خود کاراکتر شخصیت و نقشه‌ی اولیه‌ی شروع را انتخاب می‌کنید و وارد بازی می‌شوید.

### نقشه‌ی بازی

شما باید یک نقشه با انواع و اقسام بخش‌ها از جمله خانه، دریاچه، سنگ، درخت، زمین ساده و تکه‌های دیگر را پیاده‌سازی کنید و امکان جابه‌جا شدن در آن را فراهم کنید.

### کاشت، داشت و برداشت

همچنین نیاز است که شما مکانیزم‌های کاشت محصول، مراقبت از آن و برداشت آن را نیز پیاده کنید. همچنین امکان دامداری نیز در بازی محیا است.



## فروش و ارتقا

شما می‌توانید با فروش محصولات خود، پول کسب کنید و از آن پول برای خرید محصولات، دستگاه‌ها و یا ارتقا دادن دارایی‌های فعلی‌تان استفاده کنید.

## انرژی و توانایی

هرکاری که قصد انجام آن را داشته باشد، مقداری انرژی از شما کسر خواهد کرد و انرژی روزانه‌ی شما محدود است. همچنین با انجام دادن هرچه بیشتر یک کار، در آن کار حرفه‌ای‌تر می‌شوید که باعث بهره‌وری بیشتر از آن کار برای شما می‌شود.

## ساخت و ساز و آشپزی

شما با مواد اولیه‌ای که در دست خود دارید می‌توانید ابزار یا دستگاه‌های پیشرفته‌تری تولید کنید و یا محصولات خود را به غذا تبدیل کنید که ارزش فروش بیشتری دارد.

## گذر زمان

هر روز در بازی به بازه‌های کوتاهی تبدیل شده‌است که در هر یک، مقدار محدودی کار می‌توانید انجام دهید و پس از پایان کارهایتان می‌توانید نوبت را جلو ببرید، اگر تمام بازیکن‌ها نوبت را جلو ببرند، زمان بازی به اندازه‌ی یک واحد رو به جلو حرکت می‌کند.

## فراوری و ماهیگیری

شما می‌توانید از برخی دستگاه‌ها استفاده کنید تا محصولی که دارید را به محصولی دیگر با ارزش بیشتر تبدیل کنید. همچنین در بازی برای شما امکان ماهیگیری وجود دارد که فرایند کمی متفاوت نسبت به کشاورزی دارد.

## تعاملات

و در نهایت، شما می‌توانید با دیگر بازیکنان معامله، دوستی و ازدواج داشته باشید. همچنین در بازی تعدادی NPC موجود است که می‌توانید با آنها نیز دوستی برقرار کنید و برایشان مأموریت انجام دهید و پاداشی بگیرید.



# کارهایی که در فاز ۰ باید انجام شوند

## ساخت مخزن گیت‌هاب

در این بخش باید repository گیت‌هابی که تیمان قرار است در آن کد بزند را بسازید. ابتدا باید در کلاس مجازی مربوط به پروژه اضافه شوید. برای این کار به [این لینک](#) بروید. سپس در لیست داده شده با دقت شماره دانشجویی خود را انتخاب کنید. سپس  **فقط یکی از اعضای تیم** باید یک تیم سازد. باقی اعضا باید به تیم ساخته شده اضافه (join) شوند. این به این معنیست که آنها نیاز به عملیات `create team` ندارند و نباید آن را انجام دهند. دقت کنید که فرمت اسم گروه‌تان باید به شکل زیر باشد:

**Group-##**

بطوری که ## شماره گروه تعیین شده‌تان باشد. مثلاً اگر گروه ۱ هستید باید نام گروه به صورت Group-01 باشد. پس از ساخته شدن مخزن، هریک از اعضای تیم باید نام و شماره دانشجویی خود را به فایل README.md مخزن اضافه کند.

## ایجاد UML

در این بخش باید با توجه به قسمت شرح پروژه که در همین سند آمده است یک طراحی UML استاندارد کنید و فایل حاصل را به صورت تصویر یا pdf روی repository تیم خود قرار دهید. توجه کنید که حق ایجاد UML به صورت دست‌نویس را ندارید و حتماً باید با ابزارهای دیجیتال کشیده شود. باید فایل خام UML را هم روی مخزن گیت‌هاب خود قرار دهید. UML حاصل باید شامل تمام کلاس‌های لازم برای پیاده‌سازی منطق پروژه باشد و طبق استانداردها و نوتاسیون UML کشیده شده باشد.



## کارهایی که در فاز ۱ باید انجام شوند

در این فاز، صرفاً منطق پروژه، بدون پیاده‌سازی گرافیکی یا معماری شبکه‌ای آن، باید پیاده‌سازی شود. نحوه ارتباط با کاربر نیز از طریق واسط کاربری کنسول است.

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 می‌مستند، موجودیت‌ها، نمای کلی رابط کاربری سیستم، نقش‌ها و دستورات لازم شرح داده شده است.

**نکته ۱:** در هر جایی از پروژه می‌توانید هرگونه خلاصه‌ی را به کار 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 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، به صورت خودکار اطلاعات قبلی خوانده شده 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، های مربوطه را به فایل pom.xml اضافه کنید.

**نکته ۴:** اضافه کردن چیت‌کدها برای بازی به تست راحت‌تر و همچنین تحويل ساده و سریع‌تر کمک می‌کند، لذا خواهشمندیم اگر قصد زدن بخشی از پروژه را دارید، حتماً چیت‌کدهای آن بخش را نیز بزنید. چیت‌کدها با بلاک‌های زرد رنگ مشخص شده‌اند.

دستور عادی

test

چیت کد

cheat code



## شرح پروژه

### مقدمه‌ای بر ابزارهای ساخت

در فرایند توسعه نرم‌افزار، تبدیل کد منع به یک خروجی قابل اجرا (اعم از فایل باینری، کتابخانه یا بسته قابل استقرار) شامل مجموعه‌ای از مراحل تکراری و فنی نظیر کامپایل کردن<sup>۱</sup>، مدیریت وایستگی‌ها<sup>۲</sup>، اجرای تست‌ها<sup>۳</sup>، تولید مستندات<sup>۴</sup> و بسته‌بندی نهایی<sup>۵</sup> است. انجام این مراحل به صورت دستی نه تنها زمان‌بر و مستعد خطاست، بلکه در پروژه‌های بزرگ و تیمی منجر به ناهمانگی و پیچیدگی در نگهداری پروژه می‌شود.

ابزارهای ساخت با هدف خودکارسازی این فرایند طراحی شده‌اند و تلاش می‌کنند تا با کمترین میزان دخالت انسانی، مراحل مورد نیاز برای ساخت یک پروژه را به شکل استاندارد و قابل تکرار انجام دهند. این ابزارها با ایجاد یک لایه‌ی انتزاعی، فرایند ساخت را از جزئیات اجرایی آن جدا می‌کنند؛ به این معنا که توسعه‌دهنده صرفاً مشخص می‌کند چه خروجی‌ای نیاز است و ابزار ساخت وظیفه دارد چگونگی تولید آن را مدیریت کند.

استفاده از ابزارهای ساخت، بهویژه در پروژه‌هایی با ساختار پیچیده یا تیم‌های چندنفره، موجب افزایش انسجام، قابلیت تکرار<sup>۶</sup> و قابلیت نگهداری<sup>۷</sup> می‌شود. همچنین بسیاری از این ابزارها از اجرای خودکار تست‌ها پشتیبانی کرده و به راحتی در فرایندهای CI/CD<sup>۸</sup> قابل استفاده هستند؛ موضوعی که در توسعه حرفه‌ای نرم‌افزار نقش اساسی دارد.

از جمله ابزارهای رایج در این حوزه می‌توان به Maven و Gradle، ابزارهای Java و CMake در زبان‌های C و C++، npm در پروژه‌های JavaScript و Bazel برای پروژه‌های بزرگ چندزبانه اشاره کرد. انتخاب صحیح ابزار ساخت متناسب با زبان، معماری و مقیاس پروژه، یکی از ارکان کلیدی موقیت در مدیریت و اجرای پروژه‌های نرم‌افزاری پیشرفته به شمار می‌رود.

در پروژه‌ی حاضر، با توجه به استفاده از زبان برنامه‌نویسی Java و نیاز به مدیریت دقق وایستگی‌ها و ساختار مازولا، ابزار Apache Maven به عنوان سیستم ساخت پروژه انتخاب شده است. Maven با فراهم‌سازی امکاناتی نظیر مدیریت خودکار وایستگی‌ها از طریق مخازن مرکزی<sup>۹</sup> و پشتیبانی از ساختار استاندارد پوشه‌بندی، امکان توسعه‌ای منظم، قابل نگهداری و قابل گسترش را فراهم می‌سازد. همچنین، انتخاب این ابزار به دلیل تطابق آن با محیط‌های توسعه یکپارچه (نظیر IntelliJ IDEA و Eclipse) و قابلیت ادغام آسان با سامانه‌های کنترل نسخه و CI، منجر به تسهیل فرایند توسعه و استقرار خواهد شد.

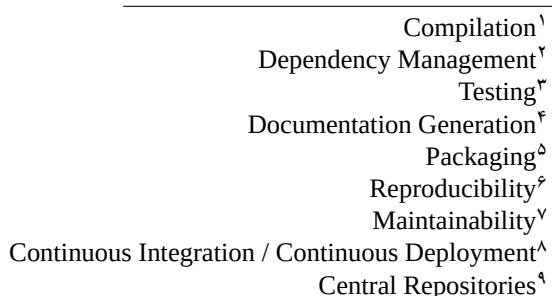
### مراحل استفاده از Maven Apache

در صورتی که از محیط برنامه‌نویسی IntelliJ IDEA استفاده می‌کنید، توصیه می‌شود از امکانات خود این محیط برای کار با Maven استفاده کنید. در غیر این صورت، مراحل اصلی استفاده از Maven به شرح زیر است:

۱. **نصب Maven**: ابتدا باید ابزار Apache Maven را از وبسایت رسمی آن دریافت و نصب کنید. اطمینان حاصل شود که مسیر bin آن به متغیر PATH سیستم اضافه شده باشد.

۲. **ساختار دهی پروژه**: پروژه باید مطابق با ساختار استاندارد Maven تنظیم شود. مسیرهای متداول به شکل زیر هستند:

- - کدهای اصلی src/main/java
- - کدهای تست src/test/java





- فایل pom.xml در ریشه پروژه – فایل پیکربندی Maven

۳. ایجاد فایل pom.xml: این فایل حاوی اطلاعات پروژه، نسخه، وابستگی‌ها، افزونه‌ها و تنظیمات ساخت است. با تعریف درست این فایل، Maven می‌تواند فرایند ساخت را به صورت خودکار مدیریت کند.

۴. تعریف وابستگی‌ها: کتابخانه‌های مورد نیاز پروژه از طریق فایل pom.xml و بخش <dependencies> تعریف می‌شوند. آن‌ها را به صورت خودکار از مخازن مرکزی دریافت کرده و به پروژه اضافه می‌کند.

در نهایت لازم به ذکر است که Maven امکانات زیر را نیز فراهم می‌کند:

• اجرای تست‌ها: Maven به صورت پیش‌فرض از فریم‌ورک JUnit پشتیبانی می‌کند. تست‌ها با اجرای دستور mvn test اجرا شده و نتایج آن‌ها گزارش داده می‌شود.

• ادغام با محیط توسعه: ابزار Maven به صورت کامل با محیط‌های توسعه‌ی پرکاربرد نظری Eclipse و IntelliJ IDEA یکپارچه می‌شود و امکان اجرای دستورات ساخت و مدیریت وابستگی‌ها را از طریق رابط گرافیکی فراهم می‌سازد.



## بخش امتیازی - تست نویسی

به منظور افزایش اطمینان از صحت عملکرد بخش‌های مختلف سیستم و رعایت اصول توسعه‌ی نرم‌افزار، در این پروژه نوشتن تست‌های واحد برای یکی از بخش‌های پروژه در زیر لیست شده است

- منوها

- زمان، تاریخ و آب و هوای

- حرکت و نحوه بازی

- ابزارآلات، ساخت و ساز و آشپزی

- دامداری و ماهیگیری و فراوری

- معامله و تعاملات بین بازیکنان

- انرژی، NPC و inventory

به انتخاب شما به عنوان یک بخش امتیازی در نظر گرفته شده است.

### نمونه‌ای از منطق قابل تست

برای مثال، دستور زیر را در نظر بگیرید:

```
register -u username -p password -e email
```

در پیاده‌سازی این دستور، منطق‌های کنترلی زیر لحاظ شده‌اند:

- username نباید بیش از ۸ کاراکتر باشد.

- password نباید کمتر از ۸ کاراکتر باشد.

- رشته‌ی ایمیل باید شامل کاراکتر @ باشد.

نوشتن تست برای این دستور، مستلزم پوشش تمامی حالات ممکن است؛ از جمله ورودهای معتبر، نامعتبر، خالی و مرزی. برای نمونه، باید تستی وجود داشته باشد که بررسی کند در صورت ورود یک نام کاربری ۹ کاراکتری، پیام خطای مناسب بازگردانده می‌شود. همچنین باید در همین تست یک ورودی با نام کاربری درست وجود داشته باشد که نشان دهد این شرط به درستی بررسی می‌شود. تستی مشابه همین تست برای بخش password نیز باید وجود داشته باشد. در نهایت یک تست نیز باید وجود داشته باشد که دادن ایمیل درست و یا نادرست به عنوان ورودی را در خود داشته باشد و خروجی‌های ایجاد شده را بررسی کند.

### اهمیت پوشش کامل منطق‌ها

در تست نویسی، بررسی تمامی منطق‌ها و ساریوهای مختلف، امری ضروری است. هدف از تست نویسی، کشف رفتارهای نادرست پیش از اجرا در محیط واقعی است. تست‌ها باید تمامی ریزمنطق‌های داخلی را پوشش دهند تا سیستم در مواجهه با ورودی‌های متنوع، رفتاری قابل پیش‌بینی و صحیح داشته باشد.

### تفاوت تست واحد و تست یکپارچه

تست واحد (Unit Test) به بررسی عملکرد مستقل یک تابع یا ماثول کوچک از سیستم اختصاص دارد، در حالی که تست یکپارچه (Integration Test) تعامل بین چند بخش مختلف سیستم را مورد ارزیابی قرار می‌دهد. با توجه به ماهیت این پروژه و تمرکز آن بر منطق دستورات مستقل، اجرای تست‌های واحد کفایت می‌کند و نیازی به طراحی تست‌های یکپارچه احساس نمی‌شود.



## محدوده‌ی پیاده‌سازی تست‌ها در پروژه

در این پروژه، از دانشجویان انتظار می‌رود با استفاده از فریم‌ورک JUnit، بر روی یکی از بخش‌های اصلی سیستم بازی، مجموعه‌ای از تست‌های واحد برای تمامی دستورات آن بخش بنویسن. به عنوان مثال، اگر بخش Trading انتخاب شود، باید برای دستوراتی مانند cancel، accept، ... تست‌های کامل و متنوعی نوشته شود.

### استانداردهای مورد انتظار در نگارش تست‌ها

تست‌های نوشته شده باید دارای ویژگی‌های زیر باشند:

- خوانایی بالا و ساختار منظم
- پوشش مناسب ورودی‌های مختلف (معتبر، نامعتبر، مرزی)
- استفاده از کامنت‌گذاری واضح برای بیان هدف هر تست

پیاده‌سازی تست‌های دقیق و مستند، نه تنها موجب افزایش امتیاز پروژه خواهد شد، بلکه نمایانگر درک عمیق دانشجو از منطق‌های داخلی برنامه و توانایی تحلیل و طراحی دقیق در توسعه نرم‌افزار است.

**توجه:** هنگام تحويل، با توجه به بخش انتخاب شده توسط شما، نمره امتیازی‌ای که کسب می‌کنید از حاصل ضرب نسبتی از منطق که توسط شما تست شده در بارم بخش تست نویسی بدست می‌آید. برای مثال اگر فقط تابع بالا در بخش مربوط وجود داشته باشد و شما صرفاً منطق مربوط به ایمیل را بررسی کنید، یک سوم نمره امتیازی را کسب می‌کنید.



## شروع بازی و منوها

### منوها

در این بازی به طور کلی ۳ منوی اصلی داریم :

اولین منو مربوط به login/register است. پس از آن، کاربر وارد منوی اصلی بازی می‌شود. از منوی main می‌توان به منوهای game و profile رفت که دستورات مربوط به آنها در ادامه می‌بینیم.  
**ورود به یک منو**

menu enter <menu\_name>

با وارد کردن این دستور کاربر به منوی مدنظر وارد می‌شود. در صورتی که کاربر در منوی دیگر باشد باید خطای مربوط به آن نشان داده شود. به طور مثال از داخل منوی profile نمی‌توان مستقیم به منوی game وارد شد.  
**خروج از منو**

menu exit

اگر در منوی login/register باشیم و این دستور را اجرا کنیم، از برنامه خارج می‌شویم.  
**منوی فعلی**

show current menu

این دستور نام منوی فعلی را نمایش می‌دهد.  
**نکات**

- دستورات مربوط به هر منو باید داخل همان منو قابل استفاده باشند و در سایر منوها خطای مناسب چاپ شود.

### منوی ثبت‌نام

**ساخت کاربر جدید:** در این بخش کاربر می‌تواند یک حساب کاربری جدید برای خود بسازد.

register -u <username> -p <password> <password\_confirm> -n <nickname> -e <email> -g <gender>

### خطاهای

- **نام کاربری تکراری:** نام کاربری باید از قبل در سامانه موجود باشد. در این حالت شما باید به اضافه کردن اعداد تصادفی یا - به نام کاربری وارد شده، یک نام کاربری جدید بسازید و در صورت تایید کاربر از آن استفاده کنید.
- **نام کاربری نامعتبر:** نام کاربری تنها می‌تواند شامل حروف کوچک و بزرگ، اعداد و نماد - باشد.
- **رایانامه نامعتبر:** رایانامه یا آدرس ایمیل وارد شده باید معتبر باشد.

## قوانین اعتبارسنجی ایمیل

- ایمیل باید شامل یک و فقط یک نماد @ باشد.



- ایمیل باید دارای یک نام کاربری معتبر در بخش قبل از @ باشد:
  - \* فقط شامل حروف انگلیسی (A-Z a-z)، اعداد (0-9)، نقطه (.)، خط تیره (-) و آندرلاین (\_) باشد.
  - \* با حرف یا عدد شروع و پایان باید.
  - \* نقطه (. ) باید دو بار پشت سر هم باشد.
- بخش دامنه (بعد از @) باید:
  - \* شامل حداقل یک نقطه (.) باشد.
  - \* پسوند دامنه ( مثل .com ، .ir ) باید حداقل دو حرف باشد.
  - \* فقط شامل حروف انگلیسی (A-Z a-z)، اعداد (0-9)، خط تیره (-) باشد.
  - \* با حرف یا عدد شروع و پایان باید.
- ایمیل باید شامل نمادهای غیرمجاز مانند:

? > < , " ' ; : \ / | ] [ } { + = ) ( \* & ^ % \$ # !

باشد.

## مثال‌های نامعتبر

- john..doe@example.com (دو نقطه پشت سر هم)
- user@domain (پسوند دامنه ندارد)
- user@domain.c (پسوند یک حرفی نامعتبر است)
- user@domain..com (نمط‌های دوتایی در دامنه)
- user@.com (دامنه با نقطه شروع شده)

- رمز عبور نامعتبر: در رمز عبور تنها مجاز به استفاده از نمادهای خاص، اعداد و حروف کوچک و بزرگ هستیم.
- رمز عبور ضعیف: در این بخش در صورت قوی نبودن رمز عبور، علاوه بر نمایش خطای باید دلیل آن نیز مشخص شود. رمز عبور قوی رمز عبوری تعريف می‌شود که طول آن حداقل 8 حرف باشد، از حروف کوچک و بزرگ، اعداد و نمادهای خاص نیز در آن استفاده شده باشد.
- عدم تطابق رمز عبور و تکرار آن: در این صورت از کاربر خواسته می‌شود که دوباره رمز عبور را وارد کند یا به منوی ثبت‌نام بازگردد. کاربر در صورت اشتباه می‌تواند چند بار رمز عبور را وارد کند.

نکته: نمادهای خاص شامل موارد زیر می‌باشد :

? > < , " ' ; : \ / | ] [ } { + = ) ( \* & ^ % \$ # !

**رمز عبور تصادفی:** کاربر می‌تواند به جای وارد کردن رمز عبور و تکرار آن از کلمه تصادفی استفاده کند، در این صورت شما باید یک رمز عبور تصادفی که تمام معیارهای یک رمز عبور قوی را دارد، تولید کنید و در صورت تایید کاربر آن را به عنوان رمز عبور استفاده کنید. در غیر این صورت به انتخاب کاربر یا رمز عبور جدید تولید کنید یا به منوی ثبت‌نام بازگردید. پس از بررسی تمام خطاهای بالا در صورت وجود هر گونه خطای هیچ عملیاتی نباید انجام شود و منتظر دستور بعد کاربر می‌مانیم. در صورتی که هیچ‌گونه خطای رخداد لیست سوالات امنیتی به کاربر نشان داده شده و کاربر می‌تواند یک پرسش امنیتی را به دلخواه انتخاب کند. توجه شود که لیست سوالات امنیتی توسط خود شما تعیین می‌شود و در بازی قرار داده می‌شود.

pick question -q <question\_number> -a <answer> -c <answer\_confirm>

in profile menu



## بخش امتیازی: SHA-256

الگوریتم SHA-256 (مخفف Secure Hash Algorithm 256-bit) یک تابع هش رمزنگاری شده‌ی یک‌طرفه است که هر ورودی دلخواه (مانند رمز عبور) را به خروجی‌ای با طول ثابت ۲۵۶ بیت تبدیل می‌کند. این الگوریتم به‌گونه‌ای طراحی شده که بازیابی مقدار اولیه از خروجی تولیدشده عملاً غیرممکن باشد.

### عملت استفاده از رمزنگاری یک‌طرفه

رمزنگاری یک‌طرفه برای حفاظت از اطلاعات حساس (مانند رمز عبور) طراحی شده است تا حتی در صورت دسترسی غیرمجاز به فایل‌های ذخیره‌سازی، امکان بازیابی رمز اصلی وجود نداشته باشد. در این روش، رمز عبور کاربران هنگام ذخیره‌سازی، هش می‌شود و هنگام ورود مجدد، تنها خروجی هش‌شده رمز واردشده با مقدار ذخیره‌شده مقایسه می‌گردد.

پیاده‌سازی رمزنگاری رمزهای عبور با استفاده از الگوریتم‌هایی مانند SHA-256، امنیت سیستم را به‌طور قابل توجهی افزایش داده و از افشاری مستقیم اطلاعات کاربران جلوگیری می‌کند. انجام این پیاده‌سازی برای رمز عبور کاربران، به‌عنوان یک بخش امتیازی در ارزیابی در نظر گرفته شده است.

ذخیره‌سازی اطلاعات کاربران باید به‌صورت فایل خارجی انجام شود تا در هنگام اجرای مجدد برنامه، اطلاعات قبلی بازیابی شده و در دسترس باشد. این فرآیند شامل موارد زیر است:

- ذخیره‌سازی کلیه کاربران ثبت‌نام شده به‌همراه اطلاعات آن‌ها
- بازیابی خودکار اطلاعات کاربران در شروع اجرای برنامه
- عدم نیاز به ثبت‌نام مجدد کاربران قبلی پس از بسته‌شدن برنامه

### پیشنهاد فنی

برای پیاده‌سازی این قابلیت، می‌توان از قالب‌هایی مانند JSON یا XML برای ذخیره‌سازی داده‌ها به‌صورت ساختاریافته استفاده کرد. همچنین فایل خروجی می‌تواند با پسوند .txt یا .json ذخیره شود. در پروژه‌های پیشرفته‌تر، امکان استفاده از پایگاه داده‌های سبک مانند SQLite نیز وجود دارد.

## منوی ورود

### بخش ورود

در این منو کاربران می‌توانند با وارد کردن نام کاربری و رمز عبور خود وارد منوی اصلی شوند. در صورت نبودن نام کاربری وارد شده در سیستم و یا صحیح نبودن رمز عبور وارد شده، نیاز است تا پیغام خطای مناسب نمایش داده شود. دستور پیشنهادی برای این قسمت مانند زیر می‌باشد اما آزادید بدون تغییر در ماهیت دستور و ورودی‌های آن را برای خودتان شخصی‌سازی کنید. توجه شود که فلگ in stay logged که کاربر اگر بازی را نیز بیند و مجدد باز کند همچنان در حالت logged in که از پیش قرار داشت، بماند.

```
login -u <username> -p <password> –stay-logged-in
```

### فراموشی رمز عبور

کاربر می‌تواند در این بخش با وارد کردن نام کاربری خود و پاسخ دادن به پرسش‌های امنیتی که در حساب کاربری ذخیره شده، یک کلمه عبور جدید برای خود انتخاب کند که می‌تواند تصادفی یا انتخابی باشد. در ادامه اگر جواب درست بود به صورت خودکار پیغام داده شود که رمز عبور جدید را وارد کنید و اگر پاسخ درست نبود هم به اول منو بازگردد. توجه کنید که در این بخش هم مانند منوی ثبت‌نام در صورت نیاز لازم است تا پیغام خطای مناسب نمایش داده شود.



```
forget password -u <username>
```

### دو تابع جدا

```
answer -a <answer>
```

### منوی اصلی

از منوی اصلی می‌توان به منوهای `profile`, `avatar` و `game` منتقل شد. همچنین کاربران برای خروج از اکانت خود باید در این صفحه باشند.

#### خروج کاربر

```
user logout
```

بعد از اینستور کاربر به منوی `login/register` منتقل می‌شود.

### منوی پروفایل

در این منو اطلاعات مربوط به کاربر نمایش داده می‌شود و همچنین قابلیت تغییر آنها باید پیاده‌سازی شود.

```
change username -u <username>
```

```
change nickname -u <nickname>
```

```
change email -e <email>
```

```
change password -p <new_password> -o <old_password>
```

- تمامی خطاهای مربوط به فیلدها در قسمت عضویت باید در نظر گرفته شوند.

- در صورتی که نام کاربری، نام مستعار و ایمیل با مقدار نام کاربری، نام مستعار و ایمیل کنونی کاربر برابر باشد، خطایی متمایز و مناسب نشان دهید.

- در صورتی که مقدار گذرواژه جدید با گذرواژه کنونی کاربر یکسان بود، پیام مناسبی نمایش دهید.

- در صورتی که گذرواژه قدیمی اشتباه باشد، اخطاری مناسب با آن چاپ کنید.

- دقت کنید که جنسیت را نمی‌توان عوض کرد. **ایا یک منو کنترلر پدر داشته باشیم و بقیه ازوون ارث بری کنن؟**

**برای چک یوزرنیم و ایمیل که متدها دوباره کاری شدن**

#### نمایش اطلاعات کاربران

با وارد کردن دستور زیر، اطلاعاتی را که در ادامه گفته شده است را نمایش دهید.

```
user info
```



لیست اطلاعاتی که باید نمایش داده شوند:

- نام کاربری
- نام مستعار
- بالاترین پول کسب شده در یک بازی
- تعداد بازی‌های انجام شده



## منوی بازی

برای ساخت بازی، لازم است ۱ تا ۳ بازیکن دیگر به بازی اضافه شوند. برای این کار بازیکنی که در حال حاضر با آن وارد شده است در دستور ساخت بازی جدید نام کاربری بازیکنان دیگر را وارد می‌کند. پس از این مرحله، بدون آنکه بازی شروع و بارگذاری شود، بازیکنان به نوبت نقشه خود را انتخاب می‌کنند. سپس یک بازی شروع می‌شود و تا همه بازیکنان رای به حذف آن بازی ندهنده، ادامه خواهد داشت و بازیکنی نمی‌تواند آن را ترک کند. در صورت رای به حذف بازی، آن بازی حذف شده و دیگر قابل بارگذاری نیست. بازیکن برای شروع یک بازی جدید باید همین مراحل را دوباره طی کند و یک بازی را از اول بسازد.

### شروع بازی جدید

```
game new -u <username_1> <username_2> <username_3>
```

در این دستور، کاربر اقدام به ایجاد یک بازی جدید می‌کند و می‌تواند بعد فلگ ۱ تا ۳ نام کاربری را وارد کند تا به بازی خود اضافه کنند. توجه کنید که وجود داشتن کاربر با چنین نام کاربری کافی است و در این فاز نیازی به تایید کاربرهای دیگر نیست. این دستور بازیکنان را به یک بخش دیگر منوی بازی می‌برد و پس از آن بازی شروع می‌شود.

خطاهای:

- نامعتبر بودن نام کاربری
- خالی بودن فلگ
- وجود بیش از ۳ نام کاربری (معتبر یا نامعتبر) جلوی فلگ
- عضو بازی دیگری بودن کاربر

### انتخاب نقشه

بعد از این که یک کاربر تعداد دیگری کاربر را به یک بازی اضافه کرد، قبل از آنکه بازی شروع و بارگذاری شود، همه کاربران به صورت نوبتی و turn based باید دستور زیر را وارد کنند تا نقشه خود را انتخاب کنند. اینکه چگونه این نقشه‌ها در کنار هم قرار می‌گیرند در بخش map توضیح داده شده است. نهایتاً بعد از این که همه بازیکنان نقشه خود را کردن، بازی شروع شده و به داخل بازی می‌رویم.

```
game map <map_number>
```

خطاهای:

- نامعتبر بودن شماره نقشه
- وارد کردن هر دستور شناخته شده یا نشده دیگری در این بخش از منوی بازی (در این بخش تنها دستور انتخاب نقشه معتبر است)

### بارگذاری بازی

اگر دستور شروع بازی جدید اجرا شود تمام اعضای آن بازی عضو بازی حساب می‌شوند و نمی‌توانند عضو بازی جدیدی شوند، پس هر بازیکن حداقل عضو یک بازی (چه فعال چه ذخیره شده) است. بازیکن اگر بازی ذخیره شده‌ای داشته باشد می‌تواند با دستور



زیر بازی را بارگذاری کند و اعضای آن بازی به ادامه بازی بپردازند. دقت کنید که هر کدام از اعضا باید بتوانند بازی را بارگذاری کنند، برای مثال فرض کنید علی و حسن در حال بازی هستند (که علی آن را ساخته است) حال تنها علی می‌تواند دستور خروج از بازی را بزنند اما پس از آن، هردو نفر می‌توانند دستور بارگذاری بازی را بزنند و بازی را ادامه دهند. اگر حسن دستور بارگذاری بازی را بزند، او به بازیکن اصلی بازی تبدیل شده و می‌تواند دستور خروج از بازی را استفاده بکند.

load game

اگر بازی ذخیره شده‌ای وجود نداشت خطأ دهد.

### ذخیره‌سازی وضعیت بازی

در این ویژگی امتیازی، انتظار می‌رود دانشجو امکانی فراهم کند که وضعیت فعلی بازی (از جمله اطلاعات کاربران، موجودی‌ها، موقعیت‌ها و ...) به گونه‌ای ذخیره شود که پس از بسته‌شدن برنامه و اجرای مجدد آن، بازی از همان نقطه قبلی ادامه یابد. برای پیاده‌سازی این قابلیت، می‌توان از روش‌های مختلف ذخیره‌سازی داده استفاده کرد. به عنوان مثال:

- استفاده از فایل‌های XML یا JSON برای ذخیره‌سازی داده‌ها به صورت ساختاریافته و ذخیره کردن این داده‌ها در یک موقعیت مشخص، و بارگذاری داده هنگام شروع مجدد.
- استفاده از پایگاه‌داده‌های سبک مانند SQLite یا H2 Database.

انتخاب روش مناسب به حجم داده، میزان پیچیدگی و سطح کنترل مورد نیاز بستگی دارد.

### روال بازی

#### خروج از بازی

تنها بازیکن اصلی (که بازی را ساخته یا در آخرین بار بارگذاری کرده است) می‌تواند از دستور زیر استفاده کند. هنگام زدن این دستور بازی از بین نمی‌رود و صرفاً باید در جایی خارج از برنامه ذخیره شود تا در آینده بتوان دوباره آن را بارگذاری کرد. در آینده می‌توان با دستور بارگذاری بازی ذخیره شده را دوباره به حالت اجرا درآورد.

exit game

دقت کنید این دستور فقط زمانی که نوبت دست بازیکن سازنده بازی باشد قابل اجرا است و اگر زمان نوبت بازیکن دیگری زده شود باید خطأ دهد. همچنین این دستور در تمام لحظه‌های بازی قابل اجرا است. و در ادامه بازیکن به منوی بازی بازمی‌گردد.

### حذف بازی فعلی

برای حذف و خروج از یک بازی که در جریان است، باید در منوی داخل بازی دستوری قرار داده شود که رای‌گیری برای terminate بازی را شروع کند. به این صورت که بازیکنی که این رای‌گیری را شروع کرده باید رای مثبت برای او ثبت کنید، سپس بازیکنان دیگر هر یک در turn خود ابتدا باید رای مثبت یا منفی را ثبت کنند و وقتی به نفر آخر این دور رسید، تنها در صورتی که همه بازیکنان رای مثبت داده باشند، بازیکنان از بازی حذف شده و بازیکن اصلی به منوی بازی برگرد و خود بازی نیز به طور کامل حذف می‌شود و دیگر قابل بارگذاری کردن نیست.

### رفتن به نوبت بعد

از آنجایی که این بازی چند نفره است، برای اینکه همه بازیکنان بتوانند آن را بازی کنند، آن را نوبتی می‌کنیم. با اجرای دستور زیر نوبت نفر بعد برای بازی می‌شود. هریار که همه بازیکنان یک نوبت بازی کنند، ۱ ساعت از زمان بازی می‌گذرد و دوباره نوبت نفر اول می‌شود (در مورد زمان بازی در بخش سیستم زمان و تاریخ بیشتر توضیح داده شده است). بازیکنان در هر نوبت خود تعداد



محدودی حرکت می‌توانند انجام دهند. این حرکات باید به گونه‌ای باشد که هر شخص در هر نوبت حداً کثر ۵۰ واحد انرژی مصرف کند (توضیحات انرژی در ادامه می‌آید). در صورتی که تعداد حرکات نوبت بازیکن تمام شده باشد و بخواهد راه برود یا از ابزاری استفاده کند، خطای مناسی را نمایش دهید. با استفاده از دستور زیر می‌توان این محدودیت را برداشت. برای رفتن به نوبت بعد، باید از دستور زیر استفاده کنید:

next turn

این دستور را در هر صورت (به جز وقته که در منوی اصلی بازی نیستیم و در یک منوی دیگر مانند منوی فروشگاه هستیم) می‌توان اجرا کرد، حتی اگر در آن نوبت بازیکن کاری نکرده باشد. با اجرای این دستور مطابق توضیحات بالا، نوبت بازیکن بعدی می‌شود. حواس‌تان باشد که با زدن این دستور تنها در همان دستگاه به نوبت بعد می‌رویم و به جای کاربرهای دیگر بازی می‌کنیم. نیازی به خروج و ورود با حساب کاربری فرد دیگر نیست.



## سیستم زمان و تاریخ

در این بازی از یک سیستم زمان و تاریخ فرضی استفاده می‌شود که در ابتدای بازی ساعت و تاریخ اولیه‌ای تنظیم می‌شود. با پیشرفت بازی و گذر نوبت‌ها، زمان به تدریج به جلو حرکت می‌کند و هر دور کامل باعث افزایش یک ساعت در زمان بازی می‌شود. اگر یک روز به اتمام برسد، یک روز به تاریخ بازی اضافه می‌شود. این سیستم کاملاً شبیه‌سازی شده و از زمان واقعی سیستم کاربر جدا است. بازیکنان همچنین می‌توانند از دستورات ویژه‌ای برای مشاهده یا تغییر زمان و تاریخ بازی استفاده کنند که شامل نمایش ساعت و تاریخ فعلی و چیز کدھایی برای جلو بردن زمان یا تاریخ می‌شود.

### پیکربندی اولیه

مقادیر اولیه زمان را هنگامی که بازی شروع می‌شود، این‌گونه تنظیم کنید:

- **زمان شروع:** ۹:۰۰ صبح
- **تاریخ شروع:** روز اول بهار
- پس از هر دور کامل بازی (چرخش کامل نوبت بازیکنان)، یک ساعت به زمان اضافه می‌شود.

### دستورات مدیریت زمان

#### نمایش وضعیت

۱. نمایش ساعت فعلی:

time

۲. نمایش تاریخ فعلی:

date

۳. نمایش ترکیبی ساعت و تاریخ:

datetime

۴. نمایش اینکه امروز چند شنبه است:

day of the week

### تغییر زمان (چیت کد)

این دستورات به جهت راحت‌تر شدن تست بازی هستند:

۱. جلو بردن زمان:

cheat advance time <X>h

این دستور زمان را به اندازه X ساعت جلو می‌برد.



۲. جلو بردن تاریخ:

### cheat advance date <X>d

این دستور زمان را به اندازه X روز جلو می‌برد. در هر دو مورد در صورتی که X عددی منفی بود خطای مناسب چاپ کنید.  
دقت کنید که با زدن این دو چیز کد باید تاثیرات منطقی بازی (مانند `stage` و `foraging` گیاهان) اعمال شوند.

### فصل‌ها

فصل‌ها یکی از جنبه‌های اصلی بازی *Stardew Valley* هستند. این بازی شامل ۴ فصل می‌باشد که هر فصل ۲۸ روز طول می‌کشد.  
فصل‌ها شامل بهار (**Spring**)، تابستان (**Summer**)، پاییز (**Fall**) و زمستان (**Winter**) هستند. هر فصل آیتم‌های خاصی برای جمع‌آوری در سراسر نقشه دارد.  
هر فصل ویژگی‌های خاص خود را دارد. مثلاً در بهار برخی محصولات خاص بیشتر رشد می‌کنند یا هر یک تاثیری در ماهیگیری دارند. توضیحات بیشتر اینکه هر محصول در هر فصل چه تغییراتی دارد، در بخش‌های مربوط به آن محصول آمده است.

### پیاده‌سازی

- فصل انتخاب شده تا ۲۸ روز بازی ثابت می‌ماند.
- پس از ۲۸ روز خود به خود به فصل بعدی می‌رویم.

نمایش فصل فعلی:

### season

این دستور نام فصل فعلی بازی را نمایش می‌دهد.



## سیستم آب و هوای

### انواع وضعیت‌های جوی

فصل‌ها	اثر خاص	وضعیت جوی
تمام فصل‌ها	بدون اثر خاص	آفتابی (Sunny)
بهار، تابستان، پاییز	آبیاری خودکار محصولات کاشته شده یک و نیم برابر شدن مصرف انرژی هنگام استفاده از ابزارها	بارانی (Rain)
بهار، تابستان، پاییز	امکان شکستن درختان و نابودی محصولات توسط رعدوبرق شامل تمام اثرات باران	طوفانی (Storm)
زمستان	دو برابر شدن مصرف انرژی هنگام استفاده از ابزارها	برفی (Snow)

جدول ۱: وضعیت‌های جوی و فصل‌های مربوطه

## عملکرد رعدوبرق

در روزهایی که آب و هوای طوفانی است، امکان رخ دادن رعدوبرق وجود دارد. عملکرد آن به این صورت است که قبل از شروع روز، ۳ خانه‌ی هر مزرعه به صورت تصادفی انتخاب شده و مورد اصابت رعدوبرق قرار می‌گیرند.

cheat Thor -l <x , y>

با استفاده از این چت کد می‌توان به یک لوکیشن دلخواه رعدوبرق بزنید!

## دستورات آب و هوای

### نمایش وضعیت

۱. نمایش آب و هوای فعلی:

weather

۲. پیش‌بینی آب و هوای برای فردا:

weather forecast

توجه کنید که ابتدای هر روز، مقدار آب و هوای روز بعد بر حسب فصل آن روز، به صورت تصادفی انتخاب می‌شود.



## تغییر آب و هوای (چیت کد)

cheat weather set &lt;Type&gt;

دقت کنید که این چیت کد آب و هوای فردا را عوض می‌کند نه امروز!

## (Greenhouse) گلخانه

گلخانه، ساختمانی است که از ابتدا در مزرعه وجود دارد ولی نیاز به تعمیر دارد و نمی‌توان وارد آن شد. برای ساخت آن نیاز به ۵۰۰ چوب و ۱۰۰۰ سکه است. برای تعمیر آن، هنگامی که منابع مورد نظر را داشتید با وارد کردن دستور زیر می‌توانید گلخانه را بازسازی کنید.

greenhouse build

## ساختار و قابلیت‌ها

گلخانه شامل یک زمین کشت مستطیلی به ابعاد ۵ ردیف در ۶ ستون (بدون احتساب دیوار) است که ویژگی‌های خاصی ارائه می‌دهد:

- کشت فرا فصلی: محصولات بدون محدودیت فصل رشد می‌کنند.

- عدم نیاز به مترسک: کلاً غذا به محصولات حمله نمی‌کنند.

- آبیاری اجباری: حتی در روزهای بارانی نیاز به آبیاری دستی وجود دارد.

- مخزن آب داخلی: واقع در دیوار شمالی برای پر کردن سطل آب.

- صاعقه: هیچ تاثیری بر محصولات یا درختان نمی‌گذارد.

- محصولات غولپیکر: امکان رشد وجود ندارد.

حاشیه زمین کشت با یک قاب چوبی به ضخامت ۱ کاشی احاطه شده است (بجز دیوار شمالی که مخزن آب در آن قرار دارد). این ناحیه به گونه‌ای طراحی شده است که اجازه می‌دهد تا آبپاش‌ها از بیرون زمین اصلی، کاشی‌های داخلی را آبیاری کنند. دقت کنید که منظور از آبپاش در این بخش، آبپاش‌های خودکار (sprinkler) است که در بخش‌های بعدی مفصل‌تر در موردشان توضیح داده شده است.

## ورود به روز بعد

در ساعت ۲۲ هر روز، تمامی افراد حاضر در بازی پس از رد کردن نوبت خود به خواب می‌روند و فردا را آغاز می‌کنند. ساعت از ۹ صبح فردا منتقل می‌شود، همچنین بازیکنان به صورت خودکار به خانه‌ی خود می‌روند (مطابق با توضیحات دستور walk) و فردا را از آنجا آغاز می‌کنند. اگر انرژی کافی برای رفتن به خانه را نداشته باشند، همان جا غش می‌کنند، برای جزئیات بیشتر غش کردن، به بخش انرژی مراجعه کنید. در این بین، رشد محصولات به اندازه‌ی یک روز جلو می‌رود، آب و هوای تنظیم می‌شود و مقداری foraging و کائی معدنی در مزرعه‌ی هر کس تولید می‌شود. همچنین shipping bin ها خالی شده و پول محصولات فروشی دیروز به حساب شخص واریز می‌شود.



## نقشه‌ی بازی و حرکت

### ساختن مزرعه

هر مزرعه شامل یک سری عناصر ثابت است که جایگاهشان باید مشخص شده باشد این عناصر ثابت شامل موارد زیر هستند:

- دریاچه
- گلخانه
- کلبه
- معدن سنگ یا quarry

مکان این موارد باید به صورت ثابت مشخص شده باشد. دقت کنید که کلبه ۴ در ۴ و گلخانه ۵ در ۶ (بدون احتساب دیوار) است. دریاچه و معدن سنگ باید به صورت مستطیلی با ابعاد دلخواه در نظر گرفته شوند. دقت کنید که مکان این موارد باید از پیش تعیین شده باشد و تصادفی نیست. همچنین جهت تنوع باید حداقل دو نوع مزرعه از پیش تعیین شده داشته باشید برای مثال می‌توانید در یک نقشه دو دریاچه بگذارید تا مزرعه قابلیت ماهیگیری بیشتری داشته باشد و در مزرعه دیگر بخش معدن سنگ را بزرگ‌تر در نظر بگیرید. در ادامه برای ساخت مزرعه یک سری آیتم به صورت تصادفی باید در ابتدای بازی پخش شوند که شامل موارد زیر اند:

- انواع درخت‌ها
- انواع سنگ‌ها
- انواع foraging

ها که در مورد آنها توضیح داده خواهد شد.

این موارد از پیش تعیین شده نیستند و باید روی نقشه‌ای که بازیکن انتخاب کرده است به صورت تصادفی پخش شوند. برای مثال می‌توانید جهت شهود می‌توانید به دو تصویر زیر نگاه کنید:



شکل ۱: یک نقشه با دو دریاچه و یک معدن سنگ و بدون عناصر تصادفی

در ادامه دقت کنید که هر بازیکن هنگام ورود به بازی انتخاب می‌کند که کدام مزرعه خود را داشته باشد، سپس آیتم‌های تصادفی روی آن مزرعه شکل می‌گیرند. در نهایت این مزرعه‌ها روی یک نقشه بزرگ‌تر قرار می‌گیرند به گونه‌ای که هر مزرعه در یک گوشه از ۴ گوشی نقشه بزرگ‌تر است. در وسط نقشه بزرگ‌تر نیز دهکده NPC ها قرار دارد که توضیحات آن در ادامه



شکل ۲: همان نقشه پس از اضافه شدن عناصر تصادفی

خواهد آمد.

شما در هر خانه که باشید فقط به ۸ خانه‌ی کنارたن که با آنها مجاور هستید می‌توانید دسترسی داشته باشید و برای مثال برای آب کردن سطل آبتان باید به کنار برکه بروید به طوری که با یکی از خانه‌های برکه مجاور باشید. از دستور زیر برای جابه‌جایی و رفتن به نقطه‌ی مذکور استفاده می‌شود:

```
walk -l <x, y>
```

در این صورت در ابتدا باید بررسی کنید که می‌توانیم به مکان مذکور برسیم یا نه؛ یعنی آیا بین این خانه‌ای که در آن هستیم و خانه‌ای که می‌خواهیم به آن برویم مسیری وجود دارد یا نه و در صورت وجود نداشتن راهی خطاب مناسب دهید. در صورتی که مسیری وجود دارد از بین تمامی مسیرها کم‌انرژی‌ترین مسیر را انتخاب کرده و انرژی مورد نیاز برای رسیدن به آن خانه را خروجی دهید و منتظر تایید کاربر باشید که آیا می‌خواهد با آن انرژی به آن جا برود یا خیر. انرژی مسیری که نیاز به طی شدن دارد طبق فرمول زیر محاسبه می‌شود.

$$\text{energy needed} = \frac{\text{distance in tiles}}{20}$$

که در آن *ditance in tiles* اشاره دارد به تعداد خانه‌هایی که باید طی شود تا به مقصد برسیم. در صورت تایید کاربر، اگر انرژی کاربر به اندازه‌ی کافی بود کاربر جابه‌جا شده و به نقطه‌ی مذکور می‌رود و در غیر این صورت قبل حرکت غش می‌کند. برای توضیح غش کردن به بخش انرژی مراجعه کنید. دقت کنید که برای اینکه مسیری بین این نقطه و نقطه‌ی مذکور بپیدا کنیم باید خانه‌های مجاور را بررسی کنیم و در واقع از هر خانه فقط به خانه‌ی مجاور آن می‌توان رفت. توجه کنید که بازیکنان در حالت عادی نمی‌توانند به مزرعه دیگر بازیکنان وارد شوند، و اگر مقصد مزرعه بازیکن دیگری بود باید خطای مناسب نشان داده شود. با این دستور شما باید از نقطه‌ای که مشخص شده است، یک مربع به طول و عرض *size* از نقشه را چاپ کنید.

```
print map -l <x, y> -s <size>
```

برای مثال اگر در در خانه‌ای درخت بود می‌توانید *T* چاپ کنید و این موارد که برای هر عنصر چه حرفي به کار بگيريد به اختيار خودتان است ولی باید قابل جداسازی و فهميدن باشد.

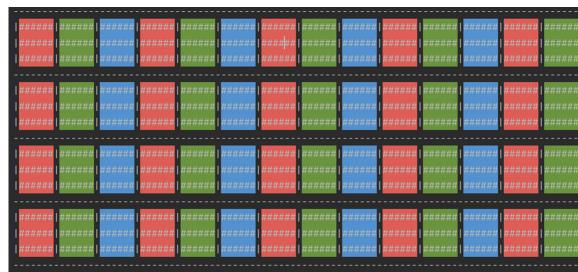


برای اینکار باید دستور زیر را داشته باشد که می‌گوید برای هر عنصر از چه حرffi استفاده کردید:

help reading map

### امتیازی

- نمایش رنگی نقشه - یک قرارداد مشخص برای عناصر موجود در هرخانه تعریف کرده و با توجه به آن، هرخانه از نقشه را با توجه به عنصر موجود در آن با رنگ مناسب نمایش دهد.



شکل ۳: قالب مورد قبول برای نقشه (نقشه شما لزوماً نباید به این شکل باشد ولی باید تمام موارد خواسته شده روی آن مشخص شده باشد)

- محاسبه‌ی مسافت با فرمول پیچیده‌قرو - به عنوان بخش امتیازی می‌توانید فرمول امتیاز را به صورت زیر محاسبه کنید و آن را در محاسبه‌ی مسیر کمینه کنید.

$$\text{energy needed} = \frac{\text{distance in tiles} + 10 \times \text{number of turns}}{20}$$

در این فرمول، number of turns نشان‌دهنده‌ی تعداد دفعاتی است که دارای پیچ در حرکت هستیم و مسیر مستقیم نیست.

- غش کردن هنگام میسر - به عنوان امتیازی می‌توانید هنگام کمبود انرژی برای کل مسیر و تایید کاربر، تا جایی که انرژی دارید حرکت کنید و در آن مکان غش کنید.



## انرژی و توانایی

در بازی به ازای هر شخص یک میزان انرژی در نظر گرفته شده است که می‌توان آن را به دستور زیر مشاهده کرد:

**energy show**

مقدار اولیه‌ی انرژی هر شخص ۲۰۰ واحد است و در روز به ازای کارهایی که انجام می‌دهد، کاهش می‌یابد (این میزان انرژی در بخش‌های مربوطه گفته شده است). توجه کنید که این ۲۰۰ واحد سقف انرژی (حداکثر انرژی) ممکن است و انرژی نمی‌تواند از آن بیشتر شود؛ البته راههایی برای افزایش یا کاهش این سقف انرژی وجود دارد که در بخش‌های بعد در مورد آن‌ها بیشتر توضیح خواهیم داد. همچنین می‌توان با خوردنی‌ها، انرژی خود را افزایش داد. در ابتدای هر روز، به غیر از شرایط خاص، انرژی شخص به ۲۰ واحد تنظیم می‌شود.

### غش کردن

در صورت به صفر رسیدن انرژی در طی روز، بازیکن غش خواهد کرد، نوبت او در باقی روز skip می‌شود و مستقیم به فردا می‌رود. فردا صبح نیز در همان جایی که غش کرده‌است، بیدار خواهد شد و ۷۵ درصد انرژی روزانه را خواهد داشت.

### تنظیم انرژی (چیت کد)

می‌توان با چیت کد زیر، میزان انرژی بازیکن را به عدد مشخص شده تنظیم کرد.

**energy set -v <value>**

با استفاده از دستور زیر نیز می‌توان مقدار انرژی را نامحدود کرد. دقت کنید که پس از زدن این دستور دیگر محدودیت مقدار انرژی ای که بازیکن می‌تواند در یک نوبت استفاده کند را نیز نداریم.

**energy unlimited**

## inventory

هر شخص می‌تواند تعدادی آیتم را در کوله‌پشتی خود حمل کنند که به آن inventory آن شخص گفته می‌شود. هر بازیکن می‌تواند آیتم‌های موجود در کوله‌پشتی خود را با دستور زیر مشاهده کند.

**inventory show**

با اجرای این دستور، تمام آیتم‌هایی که در کوله‌پشتی خود دارد نمایش داده می‌شود. دقت کنید که جای کوله‌پشتی به ازای تفاوت آیتم‌ها اشغال می‌شود نه تعداد هر کدام، به این معنی که اگر ۱۰۰ عدد هویج داشته باشیم، تنها یک جا از کوله را خواهند گرفت. اطلاعات مربوط به انواع کوله را در بخش ابزار مشاهده می‌کنید.

در ادامه با دستور زیر می‌توان یک آیتم را از inventory حذف نمود.

**inventory trash -i <item's name> -n <number>**

فلگ n - مشخص می‌کند که چه تعداد از آن آیتم حذف شود. دقت کنید که این فلگ اختیاری است و در صورتی که نیاید باید آن آیتم به صورت کلی حذف شود. برای اطلاع از جزئیات مکانیزم حذف آیتم از inventory به توضیحات سطل آشغال در



بخش ابزارآلات رجوع کنید.

## توانایی‌ها

در این بازی، هرچه بیشتر یک کاری انجام دهید، در آن ماهرتر خواهد شد. این مفهوم با نام توانایی در بازی وجود دارد. ۴ توانایی در بازی موجود است که هرکدام در زیر توضیح داده شده است. در ابتدا، تمامی این توانایی‌ها در سطح صفر شروع می‌کنند. فرمول پیش روی سطح توانایی طبق فرمول زیر محاسبه می‌شود.

$$1 + 50 \times i = \text{ واحد مورد نیاز برای رفتن به سطح } i$$

هر توانایی حداقل تا سطح ۴ قابل ارتقا است و با جلو رفتن در هر توانایی علاوه بر مزیت‌های زیر، دستورالعمل‌های مربوط به آن بخش نیز برای شما باز خواهد شد که در بخش ساخت و ساز توضیح داده شده‌اند. حال به توضیح توانایی‌های مختلف می‌پردازم. توجه کنید که بالا رفتن سطح هر مهارت روی بخش‌های زیادی از بازی تاثیر می‌گذاردند که درمورد آنها در بخش‌های بعدی بیشتر توضیح داده شده است.

## کشاورزی

هربار که محصولی را برداشت می‌کنید، چه دامی و چه کشاورزی، ۵ واحد به توانایی کشاورزی شما اضافه می‌شود. با افزایش سطح کشاورزی احتمال دریافت محصولاتی که کیفیت بالاتری دارند بیشتر می‌شود.

## استخراج

با هربار که سنگ را نابود کنید به ازای هر سنگ، ۱۰ واحد به توانایی شما اضافه می‌شود. علاوه بر این، در لول ۲، به ازای هر سنگ یا کانی، یک عنصر اضافه‌تر نیز گرفته می‌شود.

## طبیعت‌گردی

در بخش foraging می‌توانید توضیح مناسب برای فرایندهای آن را بیابید. به ازای هربار گرفتن عنصری از طبیعت، ۱۰ واحد به توانایی طبیعت‌گردی شما اضافه خواهد شد.

## ماهیگیری

با هربار گرفتن ماهی به شما ۵ واحد توانایی ماهیگیری اضافه خواهد شد. در بخش ماهی‌گیری درمورد تاثیرات سطح مهارت ماهی‌گیری بیشتر توضیح داده شده است.



## ابزارالات

در این بارهای زیادی ابزار داریم که در بخش‌های مختلف از جمله استخراج از معدن، کشاورزی، دامداری و ماهگیری استفاده می‌شوند. در ابتدا قوانین کلی مربوط به ابزارها را بیان می‌کنیم و سپس جزئیات مربوط به هر یک از این ابزارها را در بخش خودش توضیح خواهیم داد.

### دستورات مشترک ابزارها

#### تجهیز شدن به ابزار

```
tools equip <tool_name>
```

با وارد کردن این دستور، در صورتی که ابزار مورد نظر در کوله پشتی بازیکن وجود داشته باشد، بازیکن به آن ابزار تجهیز می‌شود و می‌تواند از آن استفاده کند. در غیر این صورت باید خطای متناسب نمایش داده شود.

#### ابزار فعلی

```
tools show current
```

با وارد کردن این دستور، در صورتی که ابزار فعلی که در دست بازیکن است در زیرشاخه tools قرار بگیرد، نام آن را نشان می‌دهد. در غیر این صورت باید خطای متناسب نشان داده شود.

#### ابزار موجود

```
tools show available
```

این دستور تمامی ابزارهایی که در حال حاضر در کوله پشتی بازیکن قرار دارند را نمایش می‌دهد.

#### ارتقا ابزار

```
tools upgrade <tool_name>
```

این دستور تنها زمانی کار می‌کند که بازیکن داخل آهنگری باشد، در غیر این صورت خطای متناسب نمایش داده شود. توضیحات لازم درمورد فروشگاه‌ها در ادامه وجود دارد. همچنین در صورتی که پول کافی یا منابع لازم برای ارتقای آن ابزار وجود نداشته باشد، باید خطای متناسب نمایش داده شود و در غیر این صورت با کم کردن منابع، ابزار ارتقا یافته به بازیکن داده می‌شود.

#### استفاده از ابزار

```
tools use -d <direction>
```

با استفاده از این دستور، ابزاری که هم‌اکنون در دست بازیکن قرار دارد باید بر روی کاشی مجاور با جهت داده شده (در یکی از ۸ جهت اصلی) قابلیت خود را اجرا کند. توجه کنید برخی از ابزارها مانند داس می‌توانند دو کار یا بیشتر انجام دهند، اینکه کدام کار بر روی آن کاشی انجام می‌شود در دستور نمی‌آید و برنامه شما باید بتواند تشخیص بدهد که چه آیتمی در آن کاشی وجود دارد و چه کاری روی آن می‌تواند انجام شود.

در ضمن برای ابزار چوب ماهی‌گیری سیستم دیگری برای استفاده در نظر گرفته شده و دستور آن در بخش خودش آورده شده است. همچنین این را در نظر بگیرید که تنها در صورتی از قابلیت ابزار استفاده می‌شود که بازیکن انرژی کافی برای آن کار را داشته باشد. در غیر این صورت خطای متناسب بدھید بدون اینکه از انرژی بازیکن کم کنید.

همچنین ممکن است دستور استفاده از قابلیت یک ابزار برای یک کاشی معتبر نباشد، مثلاً بازیکن قابلیت بیل را بر روی یک



کاشی که در رودخانه قرار دارد اعمال کند، در این صورت انرژی از بازیکن کسر می‌شود و خطای متناسب نیز نمایش داده خواهد شد.

**صرف انرژی:** برخی ابزارها برای کار کردن نیاز به مصرف انرژی دارند. در توضیحات مربوط به هر یک از این ابزارها این مقدار مصرف انرژی توضیح داده شده است. اما در صورتی که مهارت متناظر با برخی ابزارها به حداقل رسیده باشد در این صورت از مقدار گفته شده در آن بخش یک واحد انرژی کم می‌شود.

مثلاً اگر استفاده از بیل آهنی ۳ واحد انرژی مصرف کند، در صورتی که مهارت farming به سطح آخر رسیده باشد این مقدار ۲ واحد خواهد شد.

ابزارهای بیل و ظرف آبیاری با مهارت mining متاظراند، ابزار کلنگ با مهارت foraging متاظر است، ابزار تبر با مهارت fishing متاظر است و در نهایت چوب ماهیگیری با مهارت fishing متاظر است.

### (Hoe) بیل

#### موارد استفاده

- بیل زدن خاک برای کاشت محصول (شخم زدن)

**أنواع بیل:** در ابتدای بازی بیل اولیه باید به بازیکن داده شود و بیل‌های ارتقایافته (مسی، آهنی، طلایی و ایریدیومی) با استفاده از بیل قبل از خود (و مصرف شدن و از بین رفتن آن) ساخته می‌شوند.

**صرف انرژی:** استفاده از بیل (چه موفق و چه ناموفق) اولیه ۵ واحد، بیل مسی ۴ واحد، بیل آهنی ۳ واحد، بیل طلایی ۲ واحد و بیل ایریدیوم ۱ واحد انرژی کم می‌کند.

### (Pickaxe) کلنگ

#### موارد استفاده

- شکستن سنگ‌ها و کانی‌های معدن
- از بین بردن اثر بیل بر روی خاک
- از بین بردن آیتم‌هایی که بازیکن روی زمین گذاشته است

**أنواع کلنگ:** در ابتدای بازی یک کلنگ اولیه به بازیکن داده می‌شود. کلنگ اولیه را مانند بیل می‌توان به ترتیب به کلنگ مسی، آهنی، طلایی و ایریدیومی ارتقا داد. کلنگ‌ها می‌توانند سنگ‌ها و کانی‌های تا حداقل یک سطح بالاتر را نابود کنند، برای مثال با کلنگ مسی می‌توان سنگ‌های عادی، کانی مس و کانی آهن را نابود کرد، اما نمی‌توان کانی طلا یا ایریدیوم را نابود کرد. کانی جواهرات را هم می‌توان با کلنگ آهنی یا بهتر نابود کرد.

**صرف انرژی:** هر استفاده موفق از کلنگ اولیه ۵ انرژی کم می‌کند، کلنگ مسی ۴ انرژی، کلنگ آهنی ۳ انرژی، کلنگ طلایی ۲ انرژی و کلنگ ایریدیومی ۱ انرژی. استفاده ناموفق از هر کلنگ یک انرژی کمتر مصرف می‌کند. همچین اگر مهارت mining به سطح آخر رسیده باشد، هر استفاده از کلنگ یک انرژی کمتر مصرف می‌کند. (توجه کنید که اگر طبق این فرمول میزان انرژی مصرفی ابزاری عددی منفی شد، آن را صفر در نظر بگیرید).

### (Axe) تبر

#### موارد استفاده

- قطع کردن درختان برای چوب معمولی و عصاره برخی درختان
- از بین بردن شاخه‌هایی که روی زمین هستند.



**انواع تبر:** مانند بیل و کلنگ در ابتدای بازی تبر اولیه داده می‌شود که می‌توان آن را به ترتیب به تبر مسی، آهنی، طلایی و ایریدیومی ارتقا داد.

**صرف انرژی:** فرمول مصرف انرژی تبر مانند فرمول مصرف انرژی کلنگ است، با این تفاوت که به جای وابستگی به مهارت mining به مهارت foraging وابسته است.

### ظرف آبیاری (Watering can)

#### موارد استفاده

- آب دادن به محصولات (توجه کنید در صورتی که به یک محصول در یک نوبت آب ندهید، آن محصول هیچ رشدی نمی‌کند. به علاوه هوای بارانی آب مورد نیاز محصول در آن روز را می‌رساند و نیاز به آبیاری بازیکن نیست).
- پر کردن ظرف آبیاری از آب با انجام قابلیت (استفاده از دستور tool use) و انتخاب یک کاشی که در آن آب وجود دارد. این کاشی می‌تواند دریاچه یا منبع آب گلخانه یا هرجای دیگر که آب دارد باشد.

#### انواع ظرف آبیاری

- ظرف آبیاری اولیه: این ظرف آبیاری در ابتدای بازی باید به بازیکن داده شود و ظرف آبیاری‌های ارتقایافته با استفاده از این ظرف آبیاری (و مصرف شدن و از بین رفتن آن) ساخته می‌شوند. ظرفیت این آیتم ۴۰ است، همچنین توجه کنید برای ارتقا این آیتم باید به مغازه blacksmith رجوع شود و نرخ آن مانند ابزارهای دیگر (کلنگ و بیل و ...) است.
- ظرف آبیاری مسی: آبیاری ۵۵ کاشی از محصولات.

- ظرف آبیاری آهنی: آبیاری ۷۰ کاشی از محصولات.

- ظرف آبیاری طلایی: آبیاری ۸۵ کاشی از محصولات.

- ظرف آبیاری ایریدیوم: آبیاری ۱۰۰ کاشی از محصولات.

**صرف انرژی:** استفاده از ظرف آبیاری اولیه ۵ واحد، ظرف مسی ۴ واحد، ظرف آهنی ۳ واحد، ظرف طلایی ۲ واحد و ظرف ایریدیوم ۱ واحد انرژی کم می‌کند. همچنین اگر مهارت farming به سطح آخر رسیده باشد استفاده از ظرف آبیاری یک واحد کمتر انرژی مصرف می‌کند.

### چوب ماهی‌گیری (Fishing pole)

#### موارد استفاده

- برای گرفتن ماهی از رودخانه، دریاچه و دریا استفاده می‌شود.

#### انواع چوب ماهی‌گیری

- چوب ماهی‌گیری آموزشی: این چوب ماهی‌گیری در ابتدای بازی بدون نیاز به مهارت خاصی با ۲۵ سکه از مغازه ماهی‌گیری Fish Shop قابل خریدن است. قابلیت: فقط می‌تواند ارزان‌ترین ماهی هر فصل را بگیرد.
- چوب ماهی‌گیری بامبو: این چوب نیز بدون نیاز به مهارت خاصی با قیمت ۵۰۰ سکه در مغازه ماهی فروشی Willy موجود و قابل خریدن است. قابلیت: می‌تواند هر ماهی‌ای بگیرد.



• چوب ماهی‌گیری فایبرگلس: این چوب پس از آنکه بازیکن مهارت ماهی‌گیری سطح ۲ را باز کند، در مغازه Willy موجود می‌شود و با ۱۸۰۰ سکه قابل خرید است.

قابلیت: می‌تواند هر ماهی‌ای بگیرد و کمتر از چوب ماهی‌گیری بامبو انرژی مصرف می‌کند.

• چوب ماهی‌گیری ایریدیوم: این چوب پس از رسیدن به سطح ۴ مهارت ماهی‌گیری موجود می‌شود و با ۷۵۰۰ سکه قابل خریدن است.

قابلیت: کمتر از هر چوب ماهی‌گیری دیگری انرژی مصرف می‌کند.

**صرف انرژی:** هر پرتاپ چوب ماهی‌گیری (چه موفق و چه ناموفق) آموزشی ۸ واحد انرژی مصرف می‌کند، چوب بامبو نیز ۸ واحد انرژی مصرف می‌کند. چوب فایبرگلس ۶ واحد و چوب ایریدیوم ۴ واحد انرژی مصرف می‌کند. همچنین رسیدن به آخرین سطح مهارت ماهی‌گیری باعث می‌شود استفاده از این ابزار یک انرژی کمتر مصرف کند.

### داس (Sythe)

#### موارد استفاده

• قطع کردن هر نوع علف.

• برداشت محصولات.

**نحوه به دست آوردن:** از ابتدای بازی به بازیکن داده می‌شود.

**صرف انرژی:** به ازای هر استفاده چه موفق و چه ناموفق، ۲ واحد انرژی کم می‌کند.

### سطل شیر (Milk pail)

**موارد استفاده:** برای دوشیدن شیر از دام استفاده می‌شود.

**نحوه به دست آوردن:** با پرداخت ۱۰۰۰ سکه از مزرعه Marnie قابل خریدن است.

قابلیت: می‌تواند با استفاده روی دام‌های درست، از آنها شیر بدوشد.

**صرف انرژی:** به ازای هر استفاده چه موفق و چه ناموفق، ۴ واحد انرژی کم می‌کند.

### قیچی (Shear)

**موارد استفاده:** برای جمع‌آوری پشم از گوسفندان استفاده می‌شود.

**نحوه به دست آوردن:** با پرداخت ۱۰۰۰ سکه از مزرعه Marnie قابل خریدن است.

قابلیت: می‌تواند با استفاده بر روی گوسفندان، از روی آنها پشم جمع‌آوری کند.

**صرف انرژی:** به ازای هر استفاده چه موفق و چه ناموفق، ۴ واحد انرژی کم می‌کند.

### کوله (Backpack)

**موارد استفاده:** کوله‌پشتی برای نگهداری آیتم‌های همراه بازیکن است. دقت کنید که دستورات این بخش روی آن قابل استفاده نیست. برای اطلاع از نحوه کارکرد آن به بخش inventory مراجعه کنید.



## انواع کوله

- کوله اولیه: این کوله در ابتدای بازی باید به بازیکن داده شود، و در صورتی که بازیکن یک کوله بهتر بخرد، این کوله از بین می‌رود.  
قابلیت: ۱۲ جای موجود برای قرار گرفتن آیتم‌ها دارد.
- کوله بزرگ: این کوله از مغازه Pierre قابل خریدن است.  
قابلیت: ۲۴ جای موجود برای قرار گرفتن آیتم‌ها دارد.
- کوله دلوکس: این کوله پس از خریدن کوله بزرگ موجود می‌شود و قابل خریدن از مغازه Pierre است.  
قابلیت: ظرفیت این کوله بی‌نهایت است.

## سطل آشغال (Trash can)

**موارد استفاده:** سطل آشغال برای حذف آیتم‌ها از inventory مورد استفاده قرار می‌گیرد، وقت کنید که دستورات این بخش روی آن قابل استفاده نیست. برای اطلاع از نحوه کارکرد آن به بخش inventory مراجعه کنید.

### انواع سطل آشغال

- سطل آشغال اولیه: این سطل آشغال در ابتدای بازی باید به بازیکن داده شود و سطل آشغال‌های ارتقا یافته با استفاده از این سطل آشغال (و مصرف شدن و از بین رفتن آن) ساخته می‌شوند. این سطل آشغال در ازای حذف محصولات پولی باز نمی‌گرداند.
- سطل آشغال مسی: حذف کردن آیتم‌های داخل منوی inventory با برگرداندن ۱۵ درصد از ارزش (بر حسب سکه) فروش هر آیتم.
- سطل آشغال آهنی: حذف کردن آیتم‌های منوی inventory با برگرداندن ۳۰ درصد از ارزش فروش هر آیتم.
- سطل آشغال طلایی: حذف کردن آیتم‌های منوی inventory با برگرداندن ۴۵ درصد از ارزش فروش هر آیتم.
- سطل آشغال ایریدیوم: حذف کردن آیتم‌های منوی inventory با برگرداندن ۶۰ درصد از ارزش فروش هر آیتم.

**صرف انرژی:** استفاده از سطل آشغال انرژی مصرف نمی‌کند.



## نحوه‌ی بازی و پیشرفت

در بازی میوه‌ها و سبزیجات مختلفی وجود دارند که می‌توانیم آنها را بکاریم و برداشت کنیم و از آنها برای انرژی گرفتن و یا خرید و فروش استفاده کرد همچنین از آنها می‌توان موادی مانند آبمیوه یا ترشی و مربا تهیه کرد. می‌توانید در [این لینک](#) لیست کلیه گیاهان کشاورزی و foraging را مشاهده کنید. در ادامه این بخش با استفاده از این شیت بخش‌های مختلف را توضیح می‌دهیم.

در کل محصولات قابل برداشت شامل دو نوع سبزیجات و میوه‌ها هستند که شامل ویژگی‌های زیر هستند:

۱. **نام:** هر میوه یا سبزیجات نام دارند!
۲. **منبع:** برای هر محصول مشخص شده است که با کاشتن چه دانه‌ای می‌توان این محصول را برداشت کرد.
۳. **میزان رشد (stage):** هر محصول در بازی باید چندین مرحله را طی کند تا به رشد کامل برسد. در داک محصولات مدت زمانی که هر محصول لازم دارد تا یک مرحله را طی کند به صورت جدا شده با – نوشته شده است. محصولات در هنگام کاشته شدن در مرحله یک قرار دارند و پس از طی کردن تعداد روزهای مورد نیاز برای رد شدن از هر مرحله از صبح روز بعد در مرحله بعدی قرار می‌گیرند. مرحله نهایی نیز رشد کامل محصول و قابل برداشت شدن آن می‌باشد. در صورتی که محصول قابل به دست آمدن از farming نباشد این مقدار را ندارد.
۴. **مجموع مدت زمان لازم برای رشد کامل:** این عدد که در واقع مجموع مدت زمان تمامی مراحل است، نشان می‌دهد پس از کاشتن، چند روز باید تا رشد کامل صبر کنیم. در صورتی که محصول قابل به دست آمدن از farming نباشد این مقدار را ندارد.
۵. **قابلیت برداشت چندباره:** برخی محصولات در بازی پس از برداشت شدن کاملا از زمین مربوطه حذف می‌شوند اما برخی محصولات دیگر پس از رشد کامل قابلیت برداشت را در بازه‌های چند روزه دارند.
۶. **بازه‌های برداشت:** در صورتی که محصول دارای قابلیت برداشت چندباره باشد این عدد نشان می‌دهد تا از هر برداشت تا برداشت بعدی باید چند روز صبر کرد.
۷. **قیمت فروش پایه:** هر میوه یا سبزیجات یک میزان قیمت پایه دارند که در هنگام برداشت مطابق احتمالات موجود در داک ممکن است به یکی از ۴ کیفیت به دست آمده و با توجه به آن ضریب مربوطه در قیمت پایه ضرب شده و قیمت نهایی به دست می‌آید. می‌توانید ضرایب را نیز در داک مشاهده کنید.
۸. **قابل خوردن بودن:** بعضی از محصولات قابل خوردن هستند که پس از خوردن آن‌ها یک میزان به انرژی ما اضافه شده و محصول خورده شده از بین می‌رود.
۹. **انرژی:** پس از خوردن هر میوه این میزان انرژی به ما اضافه می‌شود. دقت کنید کیفیت میوه در انرژی آن تاثیری ندارد.
۱۰. **فصل:** هر محصول در یک فصل خاص رشد می‌کند. توضیحات کامل فصل‌ها را می‌توانید در بخش فصول مطالعه کنید. برخی محصولات که در هر فصلی قابل کاشت و برداشت هستند با عنوان special در داک قرار دارند.
۱۱. **قابلیت غولپیکر شدن:** این بخش نشان می‌دهد که آیا محصول قابلیت تبدیل شدن به یک محصول غولپیکر را دارد یا خیر. توضیحات این بخش در ادامه قرار دارد.

شما باید بتوانید با دستور زیر ویژگی‌های میوه یا سبزیجات وارد شده را ببینید:

```
craftinfo -n <craft_name>
```



در این دستور به هر طریقی که می‌خواستید باید تمام ویژگی‌های بالا را برای میوه یا سبزیجات داده شده چاپ کنید. یک نمونه از معرفی را می‌توانید در زیر ببینید:

Name: Carrot

Source: Carrot Seeds

Stages: 1-1-1

Total Harvest Time: 3

One Time: TRUE

Regrowth Time:

Base Sell Price: 35

Is Edible: TRUE

Base Energy: 75

Base Health: 33

Season: Spring

Can Become Giant: FALSE

در صورت وجود نداشتن همچین میوه یا سبزیجات خطأ مناسب دهد.

## محصولات غولپیکر

با کاشتن ۴ محصول از یک نوع در یک مریع دو در دو، اگر محصول قابلیت غولپیکر شدن را داشته باشد تمامی این محصولات حذف شده و به جای آنها یک محصول غولپیکر که کل ۴ خانه را در بر می‌گیرد قرار می‌گیرد. اگر حداقل یکی از محصولات آبیاری شده و یا کود داده شده بود محصول غولپیکر را نیز آبیاری شده و کود داده شده فرض کنید. همچنین میزان رشد این محصول را برابر با حداکثر میزان رشد چهار محصول جایگزین شده در نظر بگیرید. برای مثال اگر یکی از آن ۴ گیاه را پنج روز پیش کاشته باشید و سه محصول دیگر را روز پیش، هنگام غولپیکر شدن مدت زمان رشد این گیاه پنج روز گرفته می‌شود. هنگام برداشت این محصول به عنوان ده تا از محصول کوچک خودش محاسبه می‌شود.

## دانه‌های مخلوط

با کاشتن دانه مخلوط مربوط به هر فصل در همان فصل، از بین محصولات ذکر شده در داک مربوط به آن فصل یکی با احتمال برابر انتخاب شده و کاشته می‌شود.

## درخت‌ها

درخت‌ها نیز مشابه محصولات عادی هستند اما چند تفاوت مهم دارند. هر درخت پس از رسیدن به رشد کامل به صورت دوره‌ای میوه می‌دهد. فرایند برداشت و محاسبه میوه‌های درخت‌ها کاملاً مشابه محصولات چندبار برداشت هستند اما با برداشت از درخت به میوه آن دست می‌یابیم. در صورتی که درخت توسط رعدوبرق مورد اصابت قرار بگیرد به صورت سوخته در جای خود قرار گرفته و در صورت برداشت آن به ما ذغال سنگ (coal) اضافه می‌شود و سپس درخت سوخته از زمین برداشته خواهد شد. همچنین درختان هنگام قطع شدن، یک یا دو عدد از دانه‌ی خود را نیز به بازیکن می‌دهند.

## حمله کلاغ‌ها

به ازای هر ۱۶ محصول کاشته شده‌ای که دارید در طی شب یک کلاغ به احتمال ۲۵ درصد به مزرعه شما حمله کرده و یکی از محصولات شما را می‌خورد. در محاسبه تعداد محصولات درخت‌ها و محصولات غولپیکر را یک واحد در نظر بگیرید. در صورتی که کلاغ به یک محصول عادی یا غولپیکر حمله کند آن محصول از زمین شما حذف می‌شود و در صورتی که به درخت حمله کند، فردای آن روز درخت میوه‌ای نخواهد داد. در صورت حضور متسرک کلاغ‌ها به محصولات محافظت شده توسط آن حمله نخواهند کرد اما همچنان نیز آن محصولات در تعداد محصولات کل ما محاسبه می‌شوند. محصولات گلخانه در لیست محصولات ما نیستند.



## foraging

یا همان عناصر طبیعی تولید شده هر روز به صورت تصادفی در بخش‌هایی از نقشه ظاهر می‌شوند. در انتهای هر روز هر کاشی از مزرعه به احتمال ۱ درصد انتخاب شده و در صورت خالی بودن یکی از محصولات foraging seeds و یا foraging crops را به آن قرار می‌گیرند، دانه‌ها تنها در صورتی که خانه‌ی انتخاب شده شخمرده بود و دانه‌ی دیگری کاشته نشده بود ظاهر می‌شوند. دقت کنید در هر فصل باید عناصر مربوط به همان فصل را قرار دهید. در کنار اینها، هر روز تعدادی کانی معدنی از بین لیست minerals نیز در بخش معدن به صورت تصادفی تولید می‌شوند.

دقت کنید که قطع کردن درختان و استخراج چوب از تنه آن‌ها نیز جزو محصولات foraging حساب می‌شود. برای استخراج هر یک از انواع محصولات به ابزارات مخصوص نیاز دارید که توضیحات مربوطه در بخش ابزارها قابل مطالعه هستند: تنه درخت، انواع درخت‌ها، علف‌های هرز، انواع سنگ‌ها و کانی‌ها

در صورت انجام این قابلیت روی یکی از درخت‌های وحشی می‌توانید یکی از دانه‌های ذکر شده در داک بخش trees را به صورت رندوم دریافت کنید که این نیز به قابلیت شما وابسته است. باید در اطلاعات محصولاتی که از foraging به دست آمده‌اند مشخص شود که از این طریق به دست آمده‌اند و هنگام نمایش اطلاعات نیز این بخش به نمایش در بیاید.

## کشاورزی

حال به مراحل مختلف بخش کشاورزی می‌پردازیم:

**۱. شخم زدن:** برای کاشتن دانه باید حتما زمین شخم زده شده باشد. پس اولین مرحله در کاشت، شخم زدن زمین به وسیله‌ی بیل است.

**۲. کاشتن:** برای کاشتن یک دانه کافیست آن دانه را در inventory خود موجود داشته باشد و بر روی یک زمین شخم‌زده شده استفاده کنید.

```
plant -s <seed> -d <direction>
```

دقت کنید که direction یک عبارت است که مشخص می‌کند در کدام یک از ۸ کاشی مجاور کاشته شود. در همان کاشی که بازیکن حضور دارد نمی‌توان چیزی کاشت. در صورت شخم نزدیک بودن زمین، خطای مناسب نمایش داده شود. از دستور زیر می‌توان برای نشان دادن اطلاعات در مکان گفته شده کاشته شده استفاده کرد:

```
showplant -l <x, y>
```

در اینجا اگر در مکان گفته شده گیاهی کاشته نشده بود خطای مناسب دهید در غیر این صورت تمام اطلاعات گیاه اعم از نام، زمان باقی مانده تا به بار نشستن، مرحله فعلی گیاه، آیا امروز آبیاری شده است یا نه، کیفیت محصول و اینکه کود به گیاه داده شده است یا نه؛ نشان داده می‌شود.

**۳. کود دادن:** در بازی یک سری کود برای رشد میوه‌ها و سبزیجات وجود دارند که با نام fertilizer شناخته می‌شوند. برای استفاده کردن کود، دستور زیر را داریم:

```
fertilize -f <fertilizer> -d <direction>
```

با این دستور، در جهت گفته شده از کود ذکر شده استفاده می‌شود. در صورت نداشتن کود یا نبودن گیاهی در جهت ذکر شده خطای مناسب چاپ شود. جهت نیز مانند بخش کاشتن عمل می‌کند.

**۴. آب دادن:** هر روز باید به یک محصول آب داد و در صورت دوروز متواالی آب نخوردن گیاه از بین می‌رود و در واقع زمین شامل گیاه خود به خود خالی شده و می‌توان با شخم زدن مجدد در آن گیاه کاشت. برای نشان دادن اینکه هم اکنون در سلطمنان چقدر آب وجود دارد دستور زیر وجود دارد:

```
howmuch water
```



با زدن دستور فوق تعداد واحد آبی که در سطل باقی‌مانده است نشان داده می‌شود. وقت کنید که عملیات آب دادن با دستور `use tool` انجام می‌شود.

**۵. درو کردن:** می‌توان گیاهی که به مرحله‌ی نهایی رشد کرده است را می‌توان با داس برداشت کرد، حتی میوه‌های درخت را.



## ساخت و ساز

### صنعتگری (Crafting)

صنعتگری فعالیتی است که فقط در خانه بازیکن قابل انجام است. بازیکن باید منوی مختص به خانه را باز کرده و وارد بخش صنعتگری شود. تمام دستورات زیر با فرض حضور بازیکن در خانه بوده و در غیر این صورت خطای مناسب نمایش داده می‌شود. لیست تمام آیتم‌هایی که می‌توان در بخش صنعتگری ساخت را می‌توانید در [این لینک](#) ببینید.

#### نمایش دستورالعمل‌های ساخت

```
crafting show recipes
```

لیست تمام طرح‌هایی که بازیکن یاد گرفته است را نمایش می‌دهد.

#### یادگیری طرح جدید

بازیکن می‌تواند با دریافت طرح‌های جدید از فروشگاه‌ها، شخصیت‌ها یا ارتقای سطح مهارت، آن‌ها را به مجموعه طرح‌های خود اضافه کند.

#### ساخت آیتم

در صورت وجود مواد اولیه در inventory آیتم ساخته شده و به inventory اضافه می‌شود. عدم وجود مواد اولیه یا بلد نبودن طرح باعث نمایش خطای شود. همچنین به هنگام وارد شدن به این بخش در منوی خانه، باید لیستی از طرح‌ها نمایش داده شود و بر اساس قابل ساخت بودن یا نبودن ساخت هر طرح مشخص باشد.

```
crafting craft <item_name>
```

**صرف انرژی:** هر بار ساخت آیتم ۲ واحد انرژی مصرف می‌کند.

دقت کنید که در صورتی که در صورتی که recipe برای آیتم وجود نداشته باشد باید خطای دهد و در صورت وجود مطابق با آن recipe آیتم ساخته شود.

#### قوانين کلی

- در صورت پر بودن inventory، فرایند ساخت امکان‌پذیر نیست.

- آیتم‌های ساخته شده بلافاصله قابل استفاده هستند.

#### قراردادن آیتم‌ها روی زمین

با دستور زیر می‌توان یکی از آیتم‌هایی که قابل گذاشتن در زمین است را روی زمین گذاشت:

```
place item -n <item_name> -d <direction>
```

در صورتی که آیتم گفته شده در inventory موجود نبود، یا جهت داده شده یکی از ۸ جهت اصلی نبود، ارور مناسب دهید.



برای برداشتن آیتم‌ها از زمین و اضافه کردن آن به inventory می‌توان از کلنگ استفاده کرد (توضیحات آن در بخش ابزارآلات موجود است).

#### اضافه کردن آیتم‌ها به inventory (چیت کد)

با دستور زیر به هر تعداد از آیتم خواسته شده بدون نیاز به craft کردن آن به inventory بازیکن اضافه می‌شود.

```
cheat add item -n <item_name> -c <count>
```

این دستور آیتمی با نام <item\_name> را به تعداد <count> به inventory بازیکن اضافه می‌کند. در صورتی که در inventory امکان اضافه کردن آن آیتم وجود نداشته باشد یا آیتم گفته شده وجود نداشته باشد، ارور مناسب دهید. البته توجه کنید که با این دستور می‌توان کلیه آیتم‌های بازی را به inventory اضافه کرد، نه فقط آیتم‌های قابل craft شدن.

آشپزی

(Cooking) آشپزی

آشپزی فعالیتی است که فقط در خانه بازیکن قابل انجام است. به این صورت که بازیکن باید بتواند منوی مختص به خانه را باز کند و از آن وارد بخش آشپزی شود. همچنین تمام دستورات زیر با فرض این می‌باشد که بازیکن در خانه خود است و اگر نباشد باید خطای مناسب چاپ شود.

همچنین لیست تمام غذاهایی که با آشپزی می‌توان ساخت را در [این لینک](#) می‌توانید مشاهده کنید.

پختہ

بازیکن باید بتواند وقتی وارد خانه شود یک فضای ذخیره سازی اختصاصی برای مواد غذایی داشته باشد و بتواند با دستورات زیر مواد را از inventory به آن منتقل کند یا از آن بردارد.

cooking refrigerator [put/pick] <item>

همچنین در صورت نبودن بازیکن در خانه یا وجود نداشتن آیتم مورد نظر باید خطای مناسب نمایش داده شود. اگر آیتم مورد نظر قابل خوردن نبود نیز خطای نشان داده شود.

نمایش دستور العمل های غذایی

cooking show recipes

لیست تمام غذاهایی که بازیکن دستور پخت آنها را یاد گرفته است نمایش می‌دهد.

یادگیری دستور پخت جدید

بازیکن می‌تواند با دریافت طرح‌های جدید از فروشگاه‌ها، شخصیت‌ها یا ارتقای سطح مهارت، آن‌ها را به مجموعه طرح‌های خود اضافه کند.

تجهیه غذا

در صورت وجود اولیه در inventory یا یخچال، غذا پخته شده و به inventory اضافه می‌شود و در صورت وجود نداشتن مواد اولیه یا بلد نبودن دستور پخت باید خطای مناسب نمایش داده شود. همچنین پس از وارد شدن به این بخش در منوی خانه، باید لیستی از دستور پخت هایی که بازیکن از قبل بلد است نمایش داده شود و بر اساس قابلیت پخت در شرایط فعلی مشخص باشند.

cooking prepare <recipe\_name>

**صرف انرژی:** هر بار پخت غذا ۳ واحد انرژی مصرف می‌کند.  
دقت کنید که در صورتی که recipe برای آیتم وجود نداشته باشد یا بازیکن هنوز آن دستور پخت را دریافت نکرده باشد باید خطای دهد و در صورت وجود مطابق با آن recipe آیتم ساخته شود.



### قوانين کلی

- تمام مواد اولیه پس از ساخت/پخت از inventory یا یخچال کسر می‌شوند.
- در صورت پر بودن، operations عملیات با خطای مواجه می‌شود.
- غذاها و وسایل ساخته شده بلافاصله در inventory قرار می‌گیرند.
- دستور العمل‌های جدید با افزایش سطح مهارت‌ها یا تعامل با شخصیت‌ها باز می‌شوند.

### خوردن غذا

با دستور زیر می‌توان غذایی که در inventory قرار دارد را خورد. همه غذاها به بازیکنی که آن را خورده مقداری انرژی می‌دهند. بعضی از غذاها هم یک اثر دیگر (buff) دارند. اگر buff یک غذا مربوط به یک مهارت باشد، هنگام استفاده از ابزارهای آن مهارت (که در بخش ابزارها توضیح داده شدند) یک انرژی کمتر مصرف می‌شود. اگر هم buff مربوط به ماکسیمم انرژی باشد، خوردن آن غذا به آن اندازه به بازیکن انرژی داده و حداقل میزان انرژی را هم بالا می‌برد. توجه کنید که فقط آخرین غذایی که خورده شده به بازیکن buff می‌دهد و اگر بازیکن غذای دیگری بخورد، اثر غذای قبلی از بین می‌رود. همچنین هر buff به مدت چند ساعت فقط باقی می‌ماند و پس از گذشت آن مدت زمان، اثر buff از بین می‌رود.

eat <food\_name>

با دستور بالا در صورت وجود غذای مورد نظر در inventory بازیکن، یکی از آن کم شده و انرژی و آن غذا به بازیکن داده می‌شود. در صورتی که غذای مورد نظر در inventory بازیکن وجود نداشته باشد یا آن غذا کلاً وجود نداشته باشد، خطای مناسب را خروجی دهید.



## دامداری

### حیوانات

حیوانات بازی به دو دسته حیوانات قفسی و حیوانات طویله تقسیم بندی می‌شوند. در [اینجا](#) می‌توانید لیستی از تمام حیوانات بازی و توضیحات مربوط به آنها را مشاهده کنید.

برای ساخت طویله (barn) یا قفس (coop) باید ابتدا به مغازه Carpenter's Shop رفته و دستور زیر را وارد کنید (در ادامه لیست همه مغازه‌های بازی و آیتم‌های قابل خرید از آنها را می‌توانید مشاهده کنید).

```
build -a <building_name> -l <x , y>
```

در این دستور ابتدا بررسی شود که مختصات وارد شده داخل مزرعه است و همچنین ساختمان روی چیز دیگری (مانند گیاهان کاشته شده یا درختان) قرار نمی‌گیرد. همچنین چک شود که بازیکن پول و منابع مورد نیاز برای ساخت آن ساختمان را دارد یا نه. اگر به مشکلی نخورد، ساختمان در آن مکان ساخته شود.

برای خرید حیوان مورد نظر باید ابتدا به مغازه Marnie's Ranch رجوع کرد و از دستور زیر استفاده کرد. همچنین باید یک نام منحصر به فرد برای هر حیوان انتخاب کرد. توجه کنید که از آنجایی که حیوانات طویله و قفسی برای زندگی به خانه نیاز دارند، در صورتی که خانه‌ی مناسب آنها با ظرفیت خالی وجود نداشته باشد، پیغام مناسبی چاپ کنید.

```
buy animal -a <animal> -n <name>
```

بازیکن می‌توند با حیوانات دوستی برقار کند. بیشترین امتیاز دوستی با حیوانات ۱۰۰۰ می‌باشد. با دستور زیر می‌توان حیوانات را نوازش کرد. برای نوازش حیوانات لازم است که در یکی از ۸ کاشی اطراف آن حیوان باشیم.

```
pet -n <name>
```

کارهای تاثیرگذار در رابطه دوستی با حیوانات طویله و قفسی و اثر هر یک را در زیر مشاهده می‌کنید:

- نوازش: +۱۵
- دوشیدن شیر یا تراشیدن پشم تن گوسفند: +۵
- تغذیه از علف در محوطه بیرون: +۸
- تغذیه نشدن: -۲۰ - (در پایان روز محاسبه می‌شود.)
- شب را در بیرون سپری کردن: -۲۰ - (در پایان روز محاسبه می‌شود.)
- نوازش نشدن: ۱۰ - (۰ / سطح دوستی) (در پایان روز محاسبه می‌شود.)

با استفاده از چیت کد زیر می‌توان سطح دوستی حیوان مورد نظر را به صورت دلخواه تنظیم نمود:

```
cheat set friendship -n <animal name> -c <amount>
```

بازیکن می‌تواند با دستور زیر حیوانات خود، امتیاز دوستی با هر یک از آنها و اینکه آن روز نوازش و تغذیه شده اند یا خیر را مشاهده کند.

```
animals
```



حیوانات طویله و قفسی در صورتی که به محوطه بیرون بروند، می‌توانند از علف تازه تغذیه کنند. در غیر این صورت مجبور به تغذیه از یونجه می‌باشند که برای هر حیوان مقدار آن یک وعده است. توجه شود که در هوهای برفی، بارانی و طوفانی حیوانات نمی‌توانند از جای خود خارج شوند. بازیکن می‌تواند با دستور زیر حیوان با نام مشخص شده را به بیرون راه دهد یا آنها را داخل کند. اگر حیوانی با این نام وجود نداشت یا مختصات داده شده مناسب نبود، اورور مناسب را پرینت کنید.

**shepherd animals -n <animal name> -l <x , y>**

تغذیه حیوان مورد نظر با یونجه با دستور زیر انجام می‌شود.

**feed hay -n <animal name>**

محصول حیوانات از صبح قابلیت جمع آوری دارد. بازیکن می‌تواند با دستور زیر لیست حیواناتی که محصول آنها جمع آوری نشده است به همراه محصول تولیدی آنها و کیفیت محصول مشاهده کند.

**produces**

در صورتی که حیوان تغذیه نشده باشد، در صبح بعد محصولی نخواهد داشت. در غیر این صورت محصولی که تولید می‌کند و کیفیت آن تابع امتیاز دوستی با حیوان می‌شود.

همان طور که در لیست حیوانات مشاهده می‌کنید، برخی حیوانات محصول دومی با ارزش بیشتر دارند، برای اینکه حیوان بتواند این نوع محصولات را تولید کند باید ابتدا امتیاز دوستی آن حداقل ۱۰۰ باشد پس از آن احتمال تولید این نوع محصول از  $\text{فرمول } (1500 / 1500) \times (\text{Friendship} + \text{random})$  به دست می‌آید که در آن random یک عدد رندوم بین ۰.۵ و ۱.۵ است. در صورت شکست این این احتمال محصول حیوان از نوع اول می‌شود. پس از مشخص شدن نوع محصول تولیدی نوبت به تعیین کیفیت آن می‌رسد. ابتدا با استفاده از رابطه زیر مقدار عددی برای کیفیت تعیین می‌کنیم.  $\text{Quality} = \text{Friendship} / 1000 \times (0.5 + 0.5 \times R)$  آن از جدول زیر به دست می‌آید.

ضريب قيمت	كييفيت	مقدار به دست آمد
۱	معمولی	۰-۰.۵
۱.۲۵	نقره ای	۰.۵-۰.۷
۱.۵	طلایی	۰.۷-۰.۹
۲	ایریدیوم	بیشتر از ۰.۹

جدول ۲: کیفیت محصول تولید شده توسط حیوانات

برای برداشت محصولات گاو، بز و گوسفند به ابزار خاصی نیاز داریم که لازم است بازیکن آنها را در inventory خود داشته باشد. همچنین برای برداشت محصول خوک لازم است که به محوطه بیرون راه باید تا محصول خود را جمع آوری کند. در نهایت بازیکن با در نظر گرفتن شرایط ذکر شده می‌تواند با دستور زیر محصول حیوان مورد نظر را برداشت کند.

**collect produce -n <name>**

بازیکن می‌تواند با دستور زیر، حیوان مورد نظر را بفروشد که قیمت فروش آن از رابطه  $\text{Price} \times ((\text{Friendship} / 1000) + 0.3)$  به دست می‌آید.

**sell animal -n <name>**



### ماهی‌ها

می‌توانید لیست ماهی‌هایی که لازم است آنها را پیاده‌سازی کنید، در [اینجا](#) مشاهده کنید. بازیکن می‌تواند با دستور زیر، زمانی که نزدیک به دریاست، فرآیند ماهی‌گیری را شروع کند. توجه کنید که لازم است بازیکن چوب ماهی‌گیری مورد نظر را در inventory خود داشته باشد.

fishings -p <fishing pole>

پس از اجرای این دستور لازم است تعداد ماهی‌های گرفته شده، نوع آنها و کیفیت هر یک نمایش داده شود. نوع ماهی‌های گرفته شده به صورت زندوم و از میان ماهی‌هایی که در آن فصل یافت می‌شوند، انتخاب می‌شود. تعداد ماهی‌های گرفته شده را می‌توان با رابطه  $[R \times M \times (\text{skill} + 2)]$  به دست آورد که  $R$  در آن یک عدد زندوم بین ۰ و ۱ است. skill مهارت ماهی‌گیری بازیکن است و  $M$  به آب‌وهوا بستگی دارد. به این شکل که به ترتیب در آب‌وهواهای آفتابی، بارانی، طوفانی مقادیر ۰.۵، ۱.۲، ۱.۵ به خود می‌گیرد و در سایر آب‌وهواها مقدار آن ۱ است. توجه کنید که سقف تعداد ماهی‌های گرفته شده ۶ می‌باشد.

کیفیت ماهی‌های گرفته شده به مهارت ماهی‌گیری بازیکن و طناب ماهی‌گیری او و تعداد ماهی‌های گرفته شده بستگی دارد. نمره کیفیت هر یک از ماهی‌ها را می‌توان از رابطه  $\frac{R \times (\text{skill} + 2) \times \text{pole}}{7 - M}$  به دست آورد. در این رابطه  $R$  و skill به همان شکل قبل به کار می‌روند. مقدار pole برای طناب‌های ماهی‌گیری مختلف به صورت زیر می‌باشد:

• .۱: Training Rod •

• .۰.۵: Bamboo Pole •

• .۰.۹: Fiberglass Rod •

• .۱.۲: Iridium Rod •

برای تعیین کیفیت محصول و قیمت آن می‌توانید از جدول ۲ استفاده کنید. در لیست ماهی‌ها، ماهی‌های legendary را مشاهده می‌کنید که برای گرفتن آن‌ها لازم است مهارت ماهی‌گیری بازیکن در بیشترین حالت خود باشد و در آن زمان این ماهی‌ها نیز به لیست سایر ماهی‌ها با فصل مناسب اضافه می‌شوند.



## فراوری

در [این لینک](#) توضیحات کامل استفاده هر کدام از دستگاه‌ها آمده است. دقت کنید که برای استفاده از این دستگاه‌ها باید مطابق با داک بخش صنعت‌گری و دستورات آن ابتدا دستگاه مورد نظر را بسازید. برای استفاده از هر دستگاه کافیست بعد از ساخت دستگاه، از دستور زیر استفاده کنید:

```
artisan use <artisan_name> <item1_name>
```

برای مثال برای استفاده از Smoker Fish از دستور زیر استفاده می‌کنیم:

```
artisan use Fish_Smoker Salmon Coal
```

سپس بعد از تایم مشخص برای این آیتم، با دستور زیر می‌توانیم محصول را برداشت کنیم:

```
artisan get <artisan_name>
```

در صورت آماده نبودن یا موارد دیگر، پیغام مناسب نمایش داده شود.

**معامله****خرید**

در بازی فروشگاه‌های مختلفی وجود دارد که می‌توان آنها وسائل مورد نیاز را تهیه کرد که توضیح آنها در پایین آمده است.

۱. **Blacksmith**: این فروشگاه توسط شخصی به نام Clint اداره می‌شود. تایم کاری آن بین ساعت ۹ صبح تا ۴ عصر می‌باشد. در این فروشگاه بیشتر وسائل مربوط به استخراج و مواد معدنی به فروش می‌رسد. همچنین برای ارتقای ابزارهای خود باید به این فروشگاه بروید.

۲. **JojaMart**: این فروشگاه توسط شخصی به نام Morris اداره می‌شود. تایم کاری آن بین ساعت ۹ صبح تا ۱۱ شب می‌باشد. در این فروشگاه وسائل مربوط به کشاورزی و دانه‌های مختلف فروخته می‌شود.

۳. **Pierre's General Store**: این فروشگاه توسط شخصی به نام Pierre اداره می‌شود. تایم کاری آن بین ساعت ۹ صبح تا ۵ عصر می‌باشد. در این فروشگاه مواد مربوط به کشاورزی و تعدادی مواد متفرقه (مانند ارتقای کوله پشتی) می‌توان خرید.

۴. **Carpenter's Shop**: این فروشگاه توسط شخصی به نام Robin اداره می‌شود. تایم کاری آن بین ساعت ۹ صبح تا ۸ شب می‌باشد. ساخت طویله‌ها و خرید سنگ و چوب از این مغازه می‌تواند انجام شود.

۵. **Fish Shop**: این فروشگاه توسط شخصی به نام Willy اداره می‌شود. تایم کاری آن بین ساعت ۹ صبح تا ۵ عصر می‌باشد. در این فروشگاه وسائل مربوط به ماهیگیری فروخته می‌شود.

۶. **Marnie's Ranch**: این فروشگاه توسط شخصی به نام Marnie اداره می‌شود. تایم کاری آن بین ساعت ۹ صبح تا ۴ عصر می‌باشد. در این فروشگاه محصولات حیوانی و حیوانات مختلف فروخته می‌شود.

۷. **The Stardrop Saloon**: این فروشگاه توسط شخصی به نام Gus اداره می‌شود. تایم کاری آن بین ساعت ۱۲ ظهر تا ۱۲ شب می‌باشد. در این فروشگاه انواع غذا و نوشیدنی و دستورالعمل‌های پخت غذا فروخته می‌شود.

داک مربوط به قیمت‌ها را می‌توانید از [اینجا](#) ببینید.

**دستورات**

بعد از ورود به هر فروشگاه دستورات زیر باید قابل اجرا باشند.

**نشان دادن کلیه محصولات چه موجود و چه غیر موجود**

show all products

با این دستور باید تمام محصولات در یک فروشگاه به علاوه قیمت‌های آنها نشان داده شود.

**نشان دادن کلیه محصولات موجود**

show all available products

با این دستور باید تمام محصولات موجود در فروشگاه به علاوه قیمت حال حاضر هر کدام نشان داده شود.

**خرید**

purchase <product\_name> -n <count>



با این دستور باید بتوان محصولی با نام `product_name` خرید کرد. در صورتی که موجودی حساب کافی نباشد یا محصول مدنظر موجود نباشد یا کلا همچین محصولی وجود نداشته باشد خطای مناسب خروجی دهد. همچنین توجه کنید که بعضی از آیتم‌ها ظرفیت محدود روزانه دارند که این ظرفیت بین تمام بازیکنان مشترک است. برای مثال اگر ظرفیت روزانه یک آیتم ۳ باشد و بازیکن اول ۲ تا از آن آیتم خریده باشد، بازیکن دوم می‌تواند در آن روز حداکثر یکی دیگر از آن آیتم بخرد. در صورتی که بازیکنی می‌خواست بیشتر از ظرفیت روزانه یک آیتم آن را بخرد، خطای مناسب را چاپ کنید. تگ `n`- اختیاری است. در صورت آمدن این تگ می‌توان تعدادی که می‌خواهیم از محصول خاص را بخریم، تعیین می‌کنیم. مقدار پیش‌فروش در صورت نیامدن این تگ، یک است.

### اضافه کردن به موجودی (چیت کد)

با این دستور باید بتوانید به اندازه `count` به موجودی خود اضافه کنید.

cheat add <count> dollars

### فروش

در نقشه بازی سطلهایی وجود دارد که می‌توانید با استفاده از آنها محصولات خود را بفروشید. به این شکل که بعد از رسیدن به یکی از آنها با قرار دادن محصولی که می‌خواهید بفروشید در آن و گذشت یک روز پول به حساب شما اضافه می‌شود. بعضی از محصولات کیفیت‌های مختلفی دارند که در قیمت آنها تاثیرگذار است. کیفیت‌ها عبارت‌اند از :

۱. **regular** : قیمت آن همان قیمت پایه است و تغییری در قیمت ایجاد نمی‌کند.
۲. **silver** : قیمت جدید ۱.۲۵ برابر قیمت پایه می‌شود.
۳. **gold** : قیمت جدید ۱.۵ برابر قیمت پایه می‌شود.
۴. **Iridium** : قیمت جدید ۲ برابر قیمت پایه می‌شود.

در صورتی که قیمت اعشاری شود جز صحیح آن را در نظر بگیرید. دقت کنید اکثر محصولات در داک‌ها شامل قیمت فروش پایه هستند. برای محصولات قابل خرید که قیمت فروش آن‌ها یافت نمی‌شود (برای مثال چوب و سنگ) نصف قیمت خرید از ارزان‌ترین فروشگاه را در نظر بگیرید.

### فروش

با این دستور باید بتوان محصولی با نام `product_name` فروخت. تگ `n`- اختیاری است و می‌تواند نیاید. در صورت آمدن این تگ می‌توان تعدادی که می‌خواهیم از محصول خاص را بفروشیم تعیین کنیم. اگر این تگ نیاید تمام آیتم‌ها با این نام در `inventory` کاربر باید برای فروش قرار بگیرند. دقت کنید که برای فروش محصولات باید نزدیک یک `shipping bin` باشیم. پول حاصل از محصولات فردا صبح به حساب کاربر انتقال پیدا می‌کند.

sell <product\_name> -n <count>

خطاهای:

۱. چنین محصولی قابلیت فروش را نداشته باشد.
۲. چنین محصولی را نداشته باشیم.
۳. تعدادی که با استفاده از تگ `n`- قرار دادیم را نداشته باشیم.
۴. در فاصله‌ی یک کاشی از یک `shipping bin` نباشیم.

برای هر کدام خطای مناسب چاپ کنید.



## تعاملات و روابط با دیگر بازیکنان

### دستی

در بازی بازیکنان مختلف می‌توانند با یکدیگر دوست باشند که این دوستی می‌تواند سطوح مختلفی داشته باشد. در هر سطح، میزان دوستی به صورت  $xp$  می‌باشد و هر بازیکن با به دست آوردن میزان مشخصی از آن به سطح بعدی می‌رود. همچنین هر بازیکن با دستور زیر می‌تواند میزان و سطح دوستی خود را با بازیکنان دیگر مشاهده کند.  $xp$  لازم برای رفتن از هر سطح به سطح بعدی طبق فرمول  $100 \times (1 + \text{سطح فعلی})$  محاسبه می‌شود.

friendships

حال به توضیح سطوح مختلف دوستی می‌پردازیم:

- **سطح ۰:** در شروع بازی، روابط بازیکنان در این سطح قرار دارد. در این سطح بازیکنان می‌توانند با حرف زدن با یکدیگر و داد و ستد محبوبیت خود را میان یکدیگر افزایش دهند.
  - **سطح ۱:** در این سطح، بازیکنان علاوه بر کارهای سطح صفر، می‌توانند برای یکدیگر هدیه نیز ارسال کنند و محبوبیت خود را افزایش دهند.
  - **سطح ۲:** در این سطح، بازیکنان علاوه بر کارهای سطح قبلی، می‌توانند یکدیگر را بغل کنند. در این سطح پس از اینکه مقدار  $xp$  کامل شد، برای اینکه به سطح بعدی برویم باید یکی از بازیکنان برای دیگری گل (Bouquet) خریده و سپس سطح دوستی به سطح ۳ رسد.
  - **سطح ۳:** در این سطح علاوه بر کارهای سطح قبلی، می‌توانند با یکدیگر ازدواج کنند. پس از اینکه  $xp$  این سطح کامل شد، درخواست ازدواج داده می‌شود و تا قبل از آن نمی‌توان درخواست ازدواج کرد.
  - **سطح ۴:** زمانی که دو نفر با یکدیگر ازدواج کنند، به این سطح می‌رسند و این آخرین سطح دوستی است. در این سطح پول بازیکنان با یکدیگر جمع شده و هر دو از یک حساب مشترک استفاده می‌کنند. همچنین زمین‌های این دو بازیکن نیز متعلق به هردوی آن‌ها می‌شود.
- برای افزایش میزان دوستی میان دو نفر تعاملات زیر را می‌توان انجام داد. توجه کنید که هریک از این تعاملات زیر، در روز حداکثر یک بار دوستی بین دو بازیکن را افزایش می‌دهند و اگر در یک روز بیشتر از یک بار انجام شد، فقط دفعه اول سطح دوستی بین آنها را افزایش می‌دهد. همچنین اگر در روزی بین دو بازیکن هیچ تعاملی نباشد، میزان  $xp$  سطح دوستی آنها ۱۰ تا کاهش می‌یابد و اگر  $xp$  آنها از قبل ۰ باشد، سطح دوستی آنها یکی کاهش یافته و  $xp$  به اندازه ۱۰ تا کمتر از حداکثر  $xp$  در آن سطح جدید می‌شود.
۱. **صحبت کردن:** بازیکن‌ها با صحبت کردن با یکدیگر در هر سطحی، می‌توانند میزان دوستی خود را به اندازه ۲۰ واحد افزایش دهند. صحبت کردن میان دو نفر با دستور زیر انجام می‌شود.

talk -u <username> -m <message>

پس از این دستور، برای بازیکن مقابل یک اعلان رفته و بازیکن می‌تواند با دستور زیر صحبت‌هایی که از اول تا کنون میان او و بازیکن مورد نظر رد و بدل شده را به ترتیب مشاهده کند.

talk history -u <username>

دقت کنید که بازیکنان برای صحبت کردن با یکدیگر باید در کنار یکدیگر باشند. در غیر این صورت، پیغام مناسبی چاپ کنید.

۲. **داد و ستد:** اگر داد و ستد میان دو بازیکن با موفقیت انجام شود، میزان دوستی آن دو به اندازه ۵۰ واحد افزایش می‌یابد. اما اگر داد و ستد میان آن دو شکست بخورد، میزان دوستی آنها ۳۰ واحد کاهش می‌یابد.



۳. هدیه خریدن: بازیکن‌ها از سطح یک می‌توانند برای یکدیگر هدیه ارسال کنند. برای ارسال هدیه از دستور زیر استفاده می‌کنیم.

```
gift -u <username> -i <item> -a <amount>
```

با این دستور، اگر بازیکن‌ها کنار یکدیگر باشند و بازیکن فرستنده کالای مورد نظر را به اندازه کافی داشته باشد، هدیه به طرف مقابله فرستاده می‌شود. در غیر این صورت پیغام مناسبی چاپ کنید. پس از دریافت هدیه، آن کالا به inventory آن بازیکن اضافه می‌شود و پیغامی را دریافت می‌کند.  
پس از ارسال هدیه، برای گیرنده یک پیغام می‌رود که هدیه به او داده شده است و از ۱ تا ۵ به آن امتیاز دهد، ۱ به معنی تنفر و ۵ به معنی عاشق آن هدیه بودن است. سپس طبق فرمول  $30 + 15 \times (3 - \text{امتیاز})$  به دوستی هر دو طرف اضافه یا کم می‌شود. یک بازیکن با دستور زیر، لیست هدیه‌های دریافت کرده را می‌تواند ببیند.

```
gift list
```

و با این دستور به آنها نظر دهد.

```
gift rate -i <gift-number> -r <rate>
```

در صورت نبود هدیه با آن شماره و یا امتیاز نامعتبر، خطای مناسب چاپ شود با دستور زیر می‌توانیم تاریخچه تمام هدیه‌های فرستاده شده و دریافت شده را به طور کامل مشاهده کنیم.

```
gift history -u <username>
```

۴. بغل کردن: بازیکن‌ها از سطح دو می‌توانند یکدیگر را بغل کنند. برای بغل کردن از دستور زیر استفاده می‌کنیم. بغل کردن دوستی بین دو نفر را ۶۰ واحد افزایش می‌دهد.

```
hug -u <username>
```

دقت کنید که بازیکن‌ها باید در کنار یکدیگر باشند تا بتوان این دستور را اجرا کرد.

۵. گل خریدن: بازیکن‌ها برای رسیدن از سطح ۲ به سطح ۳ باید به یکدیگر گل بدهنند. اگر گل از طرف یکی از بازیکنان نیز هدیه داده شود، سطح دوستی هر دو به سطح ۳ می‌رسد.

```
flower -u <username>
```

بعد از اجرای این دستور، گل از inventory بازیکن هدیه‌دهنده به inventory دیگر فرستاده می‌شود. برای اجرای این دستور بازیکن‌ها باید در کنار یکدیگر باشند و همچنین گل مورد نظر باید در inventory هدیه‌دهنده باشد.

## ازدواج

پس از اینکه سطح دوستی دو بازیکن به سطح ۳ رسید، این دو بازیکن می‌توانند با یکدیگر ازدواج کنند. برای این کار بازیکن پسر دستور زیر را اجرا می‌کند. دقต کنید که برای اجرای دستور زیر، بازیکنان باید در سطح دوستی ۳ باشند و همچنین در کنار یکدیگر باشند. همچنین بازیکن درخواست‌دهنده باید حلقه مورد نظر را در inventory خود داشته باشد. همچنین جنسیت طرف مقابل باید مخالف درخواست‌دهنده باشد.

```
ask marriage -u <username> -r <ring>
```



پس از اجرای این دستور، برای بازیکن مقابله یک پیام می‌رود و او می‌تواند با دستور زیر درخواست او را قبول و یا رد کند.

```
respond (-accept | -reject) -u <username>
```

در صورتی که درخواست ازدواج قبول شود، حلقه از inventory درخواست‌دهنده به بازیکن مقابله رفته و سطح دوستی دو بازیکن به سطح ۴ رسیده و ازدواج کرده‌اند. طبق توضیحات داده شده، در این لحظه پول دو بازیکن با یکدیگر جمع شده و مشترک می‌شود و پول از هر دو کسر و به هر دو همزمان اضافه می‌شود. همچنین زمین‌های آن دو متعلق به یکدیگر می‌باشد و می‌توانند در زمین‌های یکدیگر حرکت کنند. از این پس نیز یکبار در هر روز، هنگام تعامل با یکدیگر، ۵۰ انرژی می‌گیرند. اما اگر درخواست را رد کند، سطح دوستی بین آنها به سطح صفر تزلزل می‌یابد. همچنین انرژی پس در هفت روز پیش رو، نصف انرژی عادی خواهد بود.

### داد و ستد

باید در بازی این امکان وجود داشته باشد که بازیکن‌های مختلف بتوانند با یکدیگر داد و ستد انجام دهند. هر بازیکن می‌تواند درخواست منابع خود را ثبت و بازیکن دیگر می‌تواند درخواست آن را قبول یا رد کند.

در بازی باید منوی برای تجارت داشته باشید به این صورت که با زدن دستور `mrbot`، منوی جدیدی شامل لیست بازیکنان حاضر در بازی به شما نمایش داده می‌شود.

```
start trade
```

در این منو، هر بازیکن می‌توانند از بازیکن دیگر درخواست منابع کند یا به او منابعی را پیشنهاد دهد و در ازای آن، پول و یا کالا دریافت یا ارسال کند.

```
trade -u <username> -t <type> -i <item> -a <amount> [-p <price>] [-ti <targetItem> -ta <targetAmount>]
```

نوع داد و ستد با `request` و `offer` مشخص می‌شود. در ازای کالای ارسالی یا دریافته می‌توانیم پول و یا کالا از بازیکن مقابله درخواست کنیم ( فقط یک مورد را می‌توانیم انجام دهیم).

- در صورتی که بازیکن مورد نظر وجود نداشت، پیغام مناسبی چاپ کنید.
- در صورتی که item یا amount مقدار نامعتبری داشتند پیغام نامناسبی چاپ کنید.
- در صورتی که بازیکن به تعداد موردنظر کالا و یا پول نداشت، خطای مناسبی چاپ کنید.
- در صورتی که بازیکن هر دو روش پرداخت پول و کالا به کالا را انتخاب کرده بود، پیغام مناسبی چاپ کنید.

همچنین هر بازیکن می‌تواند تمام درخواست‌ها را ببیند تا بتواند در صورت تمایل، آنها را قبول کند. توجه کنید که به همراه تمام درخواست‌ها باید یک id جهت سهولت قبول کردن و یا رد کردن آنها اختصاص یابد.

```
trade list
```

قبول و یا رد کردن یک درخواست با دستور زیر انجام می‌شود که id در آن، همان id مشخص شده در دستور بالا است.

```
trade response (-accept | -reject) -i <id>
```

با استفاده از دستور زیر، بازیکن باید بتواند لیست تمام درخواست‌ها و پیشنهادهایی که ثبت و یا قبول کرده است را به شیوه مناسب و با داده کامل ببیند.



### trade history

هر بار که بازیکن منوی داد و ستد را باز می‌کند، باید پیامی حاوی تمام درخواست‌ها یا پیشنهادهای جدیدی که برای او ارسال شده است را به همراه پیام و بازیکن فرستنده آن ببیند. وقت کنید که اگر یک درخواست یا پیشنهاد یک بار چاپ شد، در دفعات بعدی نباید چاپ شود.

وقت کنید که اگر بازیکنی درخواست و یا پیشنهاد بازیکن دیگر را قبول کند، میزان دوستی آن‌ها افزایش پیدا کرده و اگر آن را رد کند، میزان دوستی آن‌ها کاهش می‌یابد.



## ها NPC

در بازی Valley Stardew، شخصیت‌های غیر قابل بازی (NPC) نقش اساسی در ایجاد دنیایی زنده و تعاملی ایفا می‌کنند. این شخصیت‌ها دارای جدول زمانی روزانه، روابط اجتماعی، علایق خاص و ظایف منحصر به فردی هستند که بازیکن می‌تواند با آنها تعامل داشته باشد. در این پژوهه، ما به پیاده‌سازی پنج NPC اصلی خواهیم پرداخت.

### دهکده‌های NPC

همانطور که در بخش مپ توضیح داده شد، در وسط هر مپ یک دهکده مختص به NPC‌ها وجود دارد. دقت کنید که این دهکده و NPC‌های درون آن بین تمام بازیکنان مشترک است. هر NPC خانه‌ی مخصوص به خود را دارد و پس از ورود به خانه‌ی آنها می‌توان به منوی آن خانه وارد شد و دستورات مرتبط را اجرا کرد. همچنین ملاحظه داشته باشید که NPC‌های فروشگاه نیز همراه با فروشگاه خود در این دهکده حضور دارند و پس از ورود به مغازه‌ی آن‌ها می‌توان به منوی آن فروشگاه دسترسی پیدا کرد.

### ساختار و ویژگی‌های اصلی NPC‌ها در پژوهه

در این پژوهه، هر NPC باید دارای ویژگی‌ها و قابلیت‌های زیر باشد:

#### مشخصات اساسی

- نام و شغل: هویت اصلی NPC

- شخصیت‌پردازی: ویژگی‌های رفتاری، علایق و سلایق

- محل سکونت: مکان خانه یا محل کار NPC، این محل توسط خودتان تعیین می‌شود با این شرط که باید قابل دسترسی باشد. موارد بالا همگی به طور کامل در اختیار شما قرار می‌گیرند.

#### سیستم تعامل و دیالوگ

- دیالوگ‌ها: مجموعه‌ای از جملات که NPC در شرایط مختلف بیان می‌کند.

- تغییر دیالوگ: بر اساس زمان روز، فصل، وضعیت آب و هوای و سطح دوستی

- شما می‌توانید از طریق دستور زیر با NPC‌ها ارتباط بگیرید:

```
meet NPC <npc_name>
```

در مقابل، دیالوگ مناسب باید برگردانده شود. توجه کنید که باید حتما در یکی از ۸ خانه اطراف مکان فعلی NPC باشد که بتوانید با آنها صحبت کنید. همچنین اولین باری که در هر روز با هر NPC صحبت می‌کنید با آن ۲۰ امتیاز دوستی می‌گیرید.

#### سیستم هدیه و علایق

- موارد مورد علاقه: هر NPC باید اقلام مورد علاقه خود را داشته باشد.

- تاثیر هدایا: افزایش سطح دوستی بر اساس هدایای دریافتی

- شما می‌توانید با دستور زیر برای NPC هدیه بفرستید:

```
gift NPC <npc_name> -i <item>
```

اولین باری که در یک روز به یک NPC هدیه می‌دهیم، به اندازه ۵۰ امتیاز دوستی با آن NPC می‌گیریم. اگر هدیه داده شده یکی از هدیه‌های مورد علاقه آن NPC باشد، به اندازه ۲۰۰ امتیاز دوستی با آن NPC می‌گیریم. توجه کنید که ابزارآلات را نمی‌توانیم به NPC‌ها هدیه دهیم.

#### سیستم دوستی

- هر NPC سه سطح دوستی دارد که با دریافت هدایای مورد علاقه‌اش رابطه دوستی او با شما پیشرفت می‌کند.

- سطح اول: یک ماموریت اضافه برای شما باز می‌شود.



- سطح دوم: در ازای انجام ماموریت‌ها، پاداش دو برابر دریافت می‌کنید.

- سطح سوم: NPC در این سطح، از آیتم‌هایی که خودش با آنها بیشتر سروکار دارد برای شما هدیه می‌فرستد. بر اساس ویژگی‌های NPC تعدادی آیتم دلخواه در نظر بگیرید که می‌تواند هدیه دهد و ابتدای هر روز به اندازه ۵۰ درصد احتمال دارد که هر NPC با سطح دوستی ۳ یکی از آن آیتم‌ها را به عنوان هدیه بفرستد.

در ابتدای بازی تمام بازیکنان با تمام NPC‌ها در سطح دوستی ۰ قرار دارند. هر بار که ۲۰۰ امتیاز دوستی بگیریم، سطح دوستی ما با آن NPC به اندازه یک سطح بالا می‌رود. به عبارت دیگر سطح دوستی به صورت تابع مقابله از امتیاز دوستی است:  $\text{friendship level} = \lfloor \frac{\text{friendship points}}{200} \rfloor$ . همچنین حداکثر میزان سطح دوستی با هر NPC مساوی ۷۹۹ است. همچنین با دستور زیر می‌توان امتیاز دوستی و سطح دوستی را با تمام NPC‌ها را مشاهده کرد:

friendship NPC list

#### سیستم ماموریت (Quest)

- ماموریت‌های اختصاصی: هر NPC باید حداقل سه ماموریت منحصر به فرد ارائه دهد

- شرایط فعالسازی: ماموریت ابتدایی هر NPC از همان ابتدای بازی فعال است. ماموریت دوم هر NPC پس از رسیدن به سطح دوستی ۱ با آن NPC فعال می‌شود. ماموریت سوم هم پس از گذشت مقداری زمان (برای مثال ۱ فصل) فعال می‌شود. مدت زمان مورد نیاز برای فعالسازی ماموریت سوم به انتخاب خودتان هست و فقط لازم است برای هر NPC متفاوت باشد. شما می‌توانید بعد از حرف زدن با NPC مورد نظر، با دستور زیر ماموریت‌های فعال برای خودتان را مشاهده کنید.

quests list

این دستور لیست تمام ماموریت‌های فعال آن بازیکن را نشان می‌دهد.

با دستور زیر می‌توانید (در صورت داشتن آیتم‌های کافی) ماموریت را به اتمام رسانده و آیتم‌های خواسته شده را به NPC مورد نظر می‌دهیم. توجه کنید که برای به اتمام رساندن ماموریت لازم است که در یکی از ۸ خانه اطراف NPC مورد نظر باشیم.

quests finish -i <index>

این دستور ماموریت شماره <index> مطابق خروجی دستور quests list را به اتمام می‌رساند. جایزه ماموریت بعد از انجام شدن ماموریت توسط بازیکن، با پیغام مناسب باید به بازیکن تعلق می‌گیرد. هر ماموریت فقط به بازیکن اولی که آن ماموریت را انجام دهد تعلق می‌گیرد.

#### NPC‌های اصلی

در این پروژه، ما به پیاده‌سازی (حداقل) این پنج NPC اصلی خواهیم پرداخت. در صورت تمایل می‌توانید به این لیست شخصیت‌های دیگر نیز اضافه کنید.



Reward	Request	Favorites	Name
۲ عدد الماس ۵،۰۰۰ سکه طلا ۵۰ عدد کوارتز	تحویل ۵۰ واحد آهن تحویل پای کدو تحویل ۱۵۰ واحد سنگ	پشم، پای کدو، پیترای سنگ، کانی آهن، قهوه	سباستین
۱ سطح دوستی ۵۰۰ سکه طلا یک عدد آپاش اتوماتیک ایریدیومی	تحویل یک شمش طلا تحویل یک کدو تبل تحویل ۵۰ عدد گندم	قهوه، کانی آهن، شراب	ابیگیل
۷۵۰ سکه طلا ۱ سطح دوستی ۵ تا سالاد	تحویل ۱۲ تا از یک گیاه دلخواه تحویل یک ماهی سالمون تحویل یک بطری شراب	قهوه، ترشی، شراب	هاروی
۵۰۰ سکه طلا ۱ دستورالعمل آشپزی (dinner salmon) ۳ عدد مترسک دلوکس scarecrow)	تحویل ۱۰ چوب سخت تحویل یک ماهی سالمون تحویل ۲۰۰ چوب	سالاد، انگور، شراب	لیا
۱،۰۰۰ سکه طلا ۳ عدد خانه زنبور عسل ۲۵،۰۰۰ سکه	تحویل ۸۰ چوب تحویل ۱۰ شمش آهن تحویل ۱۰۰۰ چوب	اسپاگتی، چوب، شمش آهن	رابین

جدول ۳: شخصیت‌های بازی و درخواست‌های آنها