Yaffaif Manual and Walkthrough

For version v0.47

Table of Contents

1 Introduction

This guide is intended as a manual and walk through for the Yaffaif game. It should go without saying that it contains spoilers for the game. I leave it up to the player to determine how best to put it to use – whether you read it from start to finish, or dip in when you need help with a specific problem. You can use the links in the table of contents (previous page) to jump to a specific section.

> Scattered through out this document you will find developer notes like this one. This manual mainly concerns itself with how things are, but these notes help to explain *why*.

1.1 Download and Install

If you are reading this guide you'll be able to download the game from my Patreon – look for the newest posts tagged release:

http://www.patreon.com/dingotush

Click on the link in the post to download and save the file – it will go wherever your browser saves downloaded files. This should leave you with a file called `yaffaif-word-v029p.zip` where *word* is the release name and *029* is the version number, both will change from release to release. Put this file where you want to unpack the game.

1.1.1 Install Java

You will need **Java 8** in order to run the game. If you don't already have it go to java.com, download it and then install it. There are later versions of Java, but the game is targeted to version 8 (as it is the current public available version), and there are problems running it on later versions.

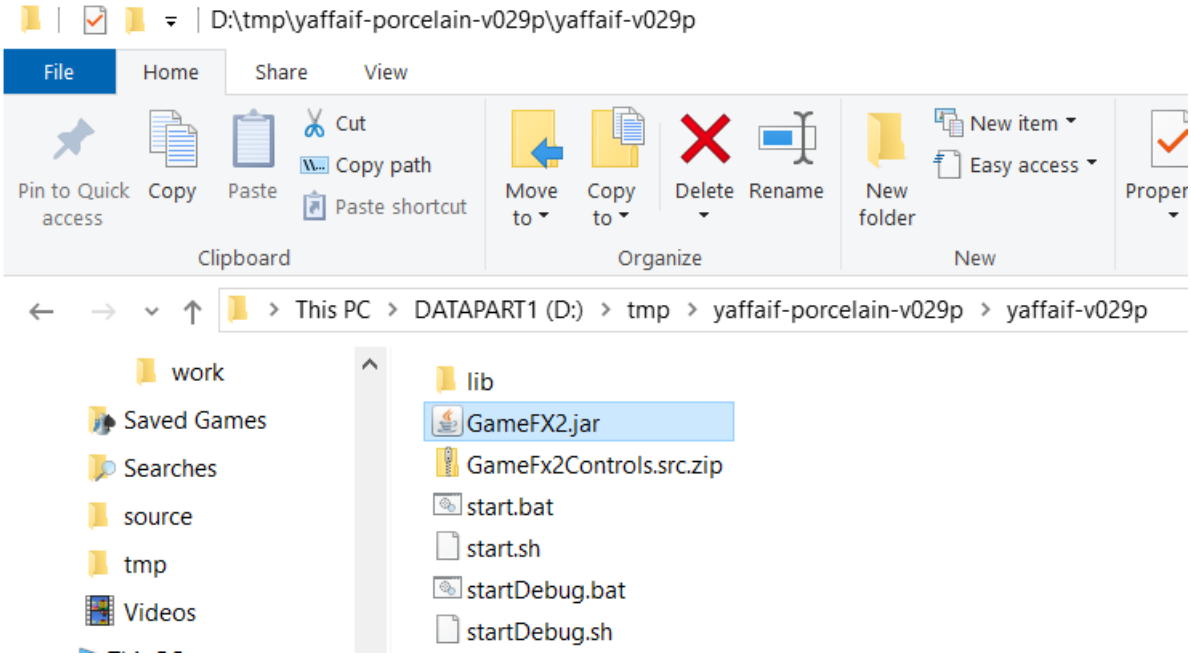
It is possible to install Java 8 alongside another version of Java, but you will need to edit the start scripts, see 1.1.4 Java version message.

1.1.2 Unpack

The next step is to unpack the archive. This is dependent on what operating system you are working on, but they all have a way of unpacking zip files which you are likely to be familiar with, or you have a favourite archiver tool. On recent versions of Windows you can right-click on the file and choose `Extract All`.

NOTE: On Windows don't double click on the zip file to navigate into it as though it were a folder/directory – this doesn't unpack the archive properly. Also, there have been issues with an archiver called BreeZip which unpacked the .jar files as well. I personally use 7zip on Windows – it's free and open source.

Depending on your archive tool and settings you may get a directory/folder called `yaffaif-v029p` or one called `yaffaif-word-v029p` with `yaffaif-v029p` in it. Descend into the `yaffaif-v029p` folder/directory and you should see this:



1.1.3 Launch the game

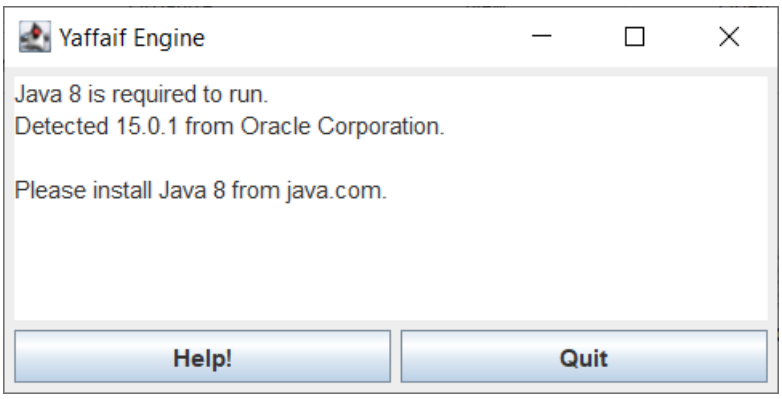
The file `GameFX2.jar` is the main file of the game. If it has the Java coffee cup icon like it does here you should be able to just double click it and the game will start.

The other files here are:

- `lib` Java libraries used by the game
- `GameFx2Controls.src.zip` Source code for the auto-complete text box
- `start.bat` Windows start script
- `start.sh` macOS/Linux/Unix start script
- `startDebug.bat` Windows debug start script
- `startDebug.sh` macOS/Linux/Unix debug start script

1.1.4 Java version message

The game checks on start-up that it is being run with Java 8. If another version is being used to run the game you will a window similar to this:



Here the detected version is a later version of Java. Another version of Java installed as well as Java 8. By default Java will try to use the newer version, but the game will not work due to the way Java was changed in these versions.

The `Help!` button tries to show the install troubleshooting web page in your browser (may or may not work). The `Quit` button closes the window.

Here are some solutions:

1.1.4.1 Remove the newer version

If you installed the newer version just to run Yaffaif and it didn't work, the uninstall it and install Java 8.

However, if you are using the newer version for something else, install Java 8 as well and edit the Yaffaif start scripts to use Java 8 (see below).

1.1.4.2 Windows

A work around is to edit the `start.bat` file (and `startDebug.bat` for debug).

First identify where Java 8 was installed, typically something like `C:\Program Files\Java\jre1.8.0_221`. The final three digits will change based on the patch level.

Then edit the `start.bat` file remove the four characters "`REM `" before the two set lines and change the first set statement for `JAVA_HOME` to match where your Java 8 runtime (or JDK) is. When complete the lines should look like this:

```
set JAVA_HOME=C:\Program Files\Java\jre1.8.0_221
set PATH=%JAVA_HOME%\bin;%PATH%
```

Launch the game by double-clicking the edited `start.bat`, or running it from a command prompt.

Note: If you update Java 8 then and remove the old version you will need to edit the script again to update the value of `JAVA_HOME`.

1.1.4.3 Unix/Linux and possibly Mac

A work around is to edit the `start.sh` file (and `startDebug.sh` for debug).

First identify where Java 8 was installed, maybe something like `/usr/bin/java/jre1.8.0_221`. The final three digits will change based on the patch level. You *might* have some success typing "`where java`" at a command prompt, but this will only search your PATH.

Then edit the `start.sh` file and remove the comment characters "`#` " before the two export lines and change the first export statement for `JAVA_HOME` to match where your Java 8 runtime (or JDK) is. When complete the lines should look like this:

```
export JAVA_HOME=/usr/bin/java/jre1.8.0_221
export PATH=$JAVA_HOME/bin:$PATH
```

Launch the game by double-clicking the edited `start.sh`, (if your file explorer supports running shell scripts by clicking) or running it from a terminal prompt.

Note: If you update Java 8 then and remove the old version you will need to edit the script again to update the value of `JAVA_HOME`.

1.1.5 Not starting at all?

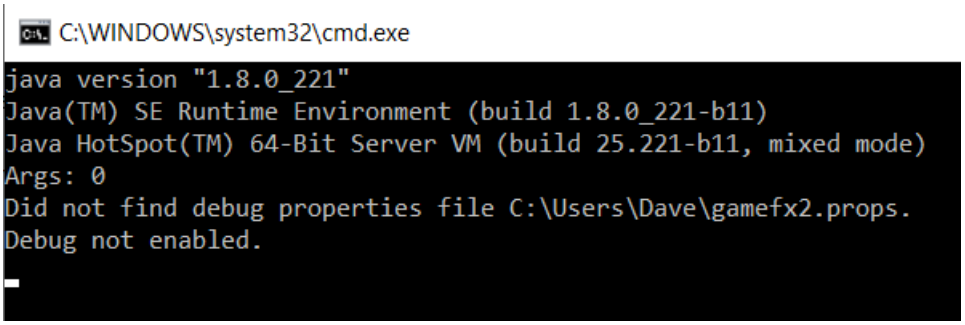
If the folder/directory contents don't look like those in the unpack section then your archiver may not have unpack the zip file correctly. Try another archiver.

If the folder contents look fine try:

On Windows, try double clicking on `start.bat`.

On macOS/Linux/Unix, start a terminal/shell and run `start.sh`.

This should start a command window and display some debug messages as the game starts that should provide some clues as to what has gone wrong. This is what it should look like:



If the output suggest the command "`java`" cannot be found, then please try re-installing Java. If the reported Java version is less than `1.8.0_25` then you have a very old version and it will need to be updated. If the reported version is `1.9` or greater see the section on Java Version Message above.

If the messages suggest a "class" cannot be found, check that the lib folder/directory contains just 3 other jar files. If has been unpacked into many folders of class files then your archiver tool has been over eager and unpacked these too. Try a different archiver.

More information can be found here:

http://wolfnose.org/fur/plump/gamefx-inst.html

If you are still having trouble, please message me.

1.2 Run in browser

Yaffaif was originally developed to run in the browser as well as be a downloadable game. This required the Java plug-in technology that has been dropped from modern browsers due to security concerns. However, it is still possible to run Yaffaif in browser if you have Internet Explorer and the Java 8 runtime installed. IE is the last of the browsers that still supports the old plug-in technology.

I don't actually recommend this, but for completeness I've included these instructions on how to run the game in IE. The version of the game will be the latest public version.

Obviously, the reason the old plug-in technology is being dropped is that it could be exploited, and the steps below explicitly override some of the security controls just for this game, but it still runs in the Java sandbox. Ultimately it comes down to the level of trust you have in me as a developer. The game requires access to the file system to save and load games, but you do have control over which files it accesses. There's no network activity or telemetry once the game is downloaded and running.

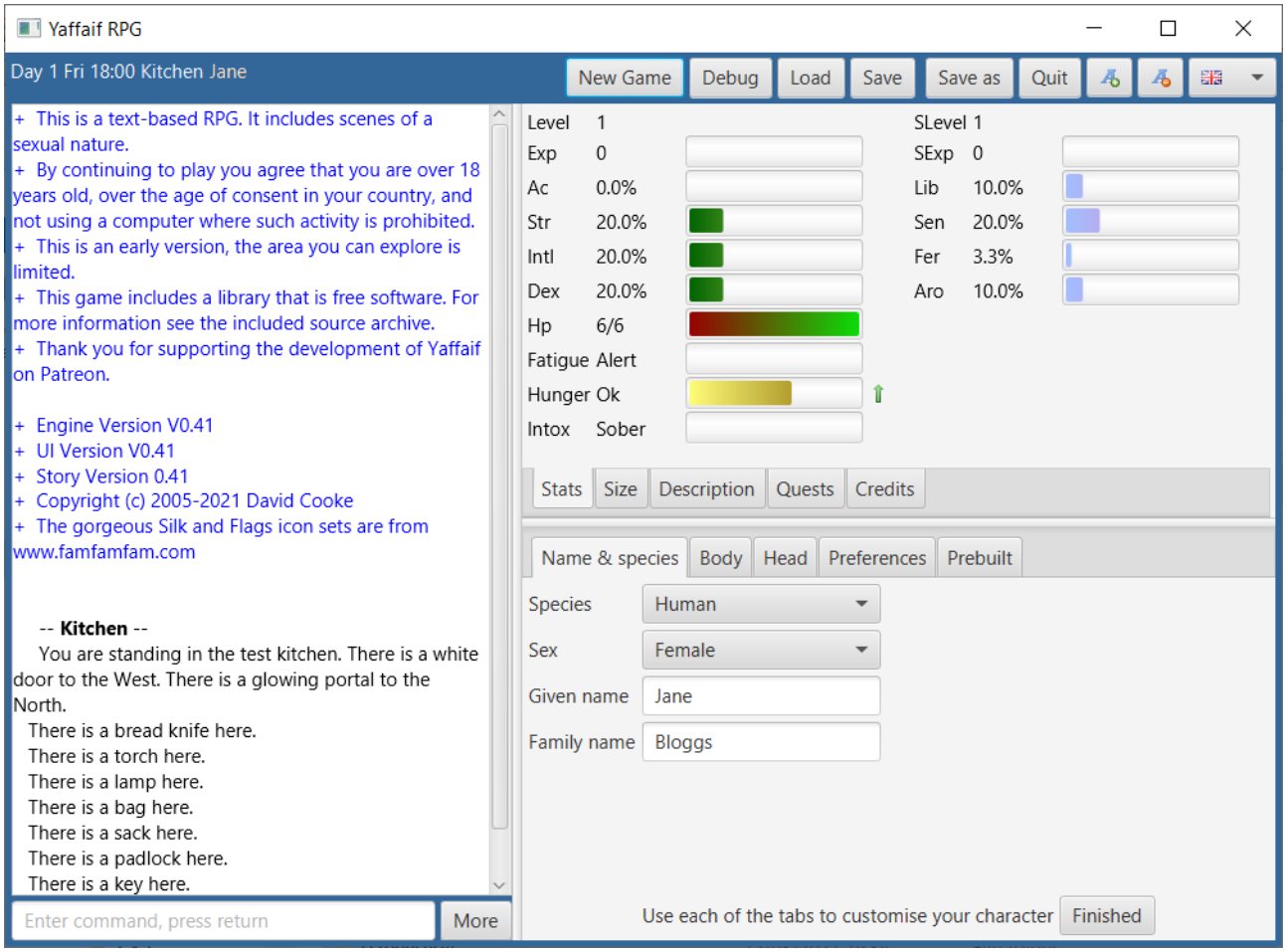
It's a bit of a hack, but here goes (Windows 10 - but similar steps work from XP onwards):

- Install Java 8 runtime if you don't have it already from java.com
- Click on the `windows logo`, and type `Java`, click on `Configure Java`. This starts the Java Control Panel.
- Click on the `Security tab`, go down to the Exception Site list, and click `Edit Site List`.
- In the pop-up click on `Add` then paste into the new box `http://wolfnose.org/fur/plump/gamefxapplet.html`
- A warning pops up, click `Continue`, then `OK` on the first pop-up to close it.
- Click on `Apply` in the original Java Control Panel.
- Start Internet Explorer (it won't work with Edge). `Windows Logo`, type `internet`, click on `Internet Explorer`.
- Paste `http://wolfnose.org/fur/plump/gamefxapplet.html` in to the address bar. You may need to agree to enabling the Java plugin.
- Scroll down, and after a short delay you should see the game running!

I'm not sure how much longer IE will exist, and this continue to work. When I started development this worked in all browsers (except Opera I think).

2 Getting Started

Once the game is running you should see something like this:



If you don't please refer to the troubleshooting page:

http://wolfnose.org/fur/plump/gamefx-inst.html

This interface is divided into four areas. At the top is the `Menu`, to the left is the `Story` pane, on the right at the top is the `Status` pane, and bottom right is the `Control` pane.

Yaffaif is a text adventure game (or interactive fiction if you prefer). You decide what your character will do, and the game displays what happens as text in the Story area.

Traditionally, these games were played by typing in text such as `go north`, `get stick`, and `hit troll`, and pressing the enter key. Yaffaif allows you to do this in the `Command` text box at the bottom of the `Story` pane.

This kind of keyboard input has a number of problems that make the game harder to play than it should be. It is easy to mistype commands: `gte sick`. Sometimes it isn't obvious what verb to use; should I hit the troll, whack it, fight it? Yaffaif attempts to get around this by providing buttons in the `Control` pane so you don't have to type commands, and by auto-completing typed commands.

The `Status` area allows you to see how your character is doing without have to type `look me` every few turns. As transformation is a central part of this game, this would soon become irritating.

2.1 Resizing

The whole user interface can be resized like a normal window. Just grab the sides or corners and drag to the size you want, or use the maximise button on your system's window decoration to have it fill the screen. The game will adjust its layout to fit the size you give it.

2.1.1 Font Size

The initial font size comes from your machine's settings. On current Windows versions this is from the System Display setting Scale and Layout.

The font plus and font minus buttons change the size of the text, adjusting the window to suit. If you run the game maximised you may want to increase the font size. There is a lower limit on the font size to stop things becoming unreadable.

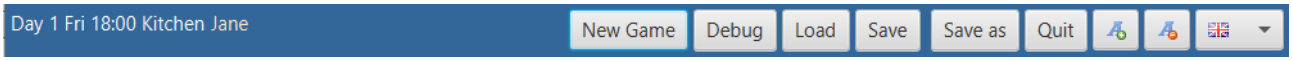
If the font size is too big for the window size then scroll bars will start to appear and text get replaced with "…" where it doesn't fit. The game will still work, but it won't be as easy to use. Either make the window bigger or the font size smaller.

2.1.2 Grab Bars

There are "grab bars" between the sections that allow you to change the size of each area if you wish. Hover your pointer over the gap between sections and it should change to a double-headed arrow – hold down the left button and drag it about. The interface can also be resized or made full screen, however you want.

Here's a little more detail about each part:

2.2 Menu



Top left is a little status area that displays the in-game day and time and your character's location and name. Knowing the time can be useful as the various non-player characters (NPCs) go about their lives and can be in different places at different times or days of the week.

`New Game` starts a new game as you might expect. This will wipe your current game, so if you want to keep that save it first.

`Debug` will only appear if the game is running in debug mode (see section 7). If pressed it launches another window that allows you to inspect and adjust the game model. It's for testing (or cheating), see the Debug Mode chapter for more information.

`Load` loads a game from a save game file. It pops up a file selector where you can choose the file it reads.

> Dev note: At the moment save games are *not* compatible between versions, something I hope to address in the future.

`Save` saves your current game to the current save game file. The name of that file appears between the `Save` and `Save as` buttons. If there's no file selected `Save` will behave like `Save as`. There are no checkpoints in Yaffaif, you can save whenever you want, even in the middle of combat or a dream. However, you may want to know that the state of the important random number generator is stored in that save file, so save-scumming won't (shouldn't) work (making different choices will).

`Save as` saves the game to a new file. It pops up a file selector where you can choose the file it writes to (or overwrites). If you want to keep your old save, then this is is the button to choose to create a new one. Don't ask it to save to any file you want to keep; it will overwrite it.

`Quit` closes the game. Unsaved progress will be lost. The same happens if you close the game with your window manager controls.

The `font plus` button makes the text in the game larger and easier to read. You can make the game window bigger by stretching the corners like a regular window or by maximising it.

The `font minus` button makes the text in the game smaller, fitting more on the screen.

The `flags` drop-down at the top right selects the measurement system used for descriptions of your character and others. Try selecting the Size tab in the status area and then choose a different flag. It doesn't (and never will) change the language used. It attempts to select the right measurements if your operating system reports the country on start up, otherwise it defaults to United Kingdom (Imperial units) because that's where I'm from. The other non-metric setting is the USA (freedom units – which are different in places from Imperial). All the others settings are metric, and differ principally in the way bra sizes are encoded.

> Dev Note: I often run with it set to European Union (metric) because the internally everything is metric.

2.3 Story

The textual output describing what you've done and observed. There's a scroll bar to go back some way if you wish in the history. Each time you take an action the existing text is dimmed slightly so that the new text is more obvious.

Messages from the game engine that are informational, warnings, or errors also show up here. Those are the ones the begin with a `+`.

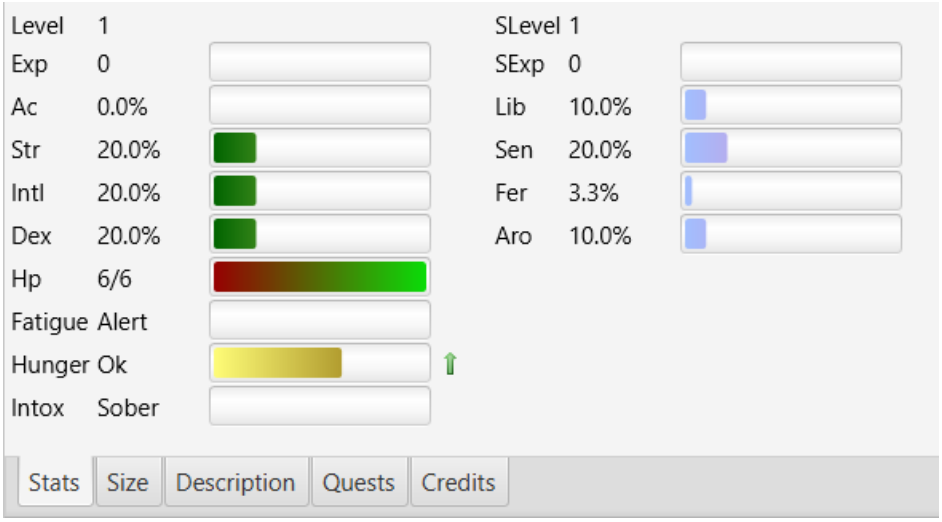
At the bottom of the story pane there's a button labelled `More`. This does absolutely nothing.

> Dev Note: It isn't possible to cut and paste from the Story pane at the moment. If you need to send me an error message please take a screenshot and send me that. If you run the game from `start.bat` or from a command line you can cut and paste the story text from that terminal as it also appears there.

2.4 Status

The status area is at the top right has four tabs with details about yourself: Stats, Size, Description, and Quests. The fifth Credits tab lists the writer(s) of the game and sponsors of the development.

2.4.1 Stats



These are the stats of your character. The ones on the left begin with standard RPG stats. The ones on the right are related to sex.

Label	Meaning	Explanation
Level	Your current level	Levelling up by filling the experience bar increases your character's abilities
Exp	Experience points and progress to the next level	Gain experience by fighting, solving puzzles and doing quests
Ac	Armour class	Degree of protection you armour offers against attacks. 0% is un-armoured, 100% is nearly impregnable.
Str	How strong you are	Stronger characters can carry more weight and get advantages in combat
Intl	How intelligent you are	Intelligent characters are less likely to act compulsively, and may be better at magic

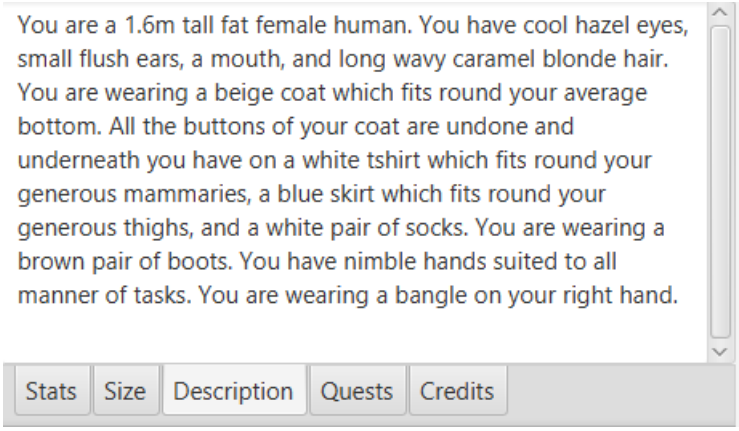
Label	Meaning	Explanation
Dex	How dextrous you are	Dextrous characters are better with ranged weapons and any delicate hand work.
Hp	Hit points and health bar	If this reaches zero (empty) you may die.
Fatigue	Your fatigue level	Fatigued characters suffer penalties in combat. Sleep to recover.
Hunger	Your hunger level	Eating is one way to heal damage, but you can only eat so much.
Intox	Your level of intoxication	A few drinks with friends is a good thing. Getting paralytic before a fight, not so much
SLevel	Your character's experience level in the ways of love	Sexual level measures how good you are in bed, and elsewhere
SExp	Sexual experience and progress to next level	Gain experience by being a good lover
Lib	Libido	Character's with a higher libido have a greater need for sexual release and their arousal rises faster
Sen	Sensitivity	The more sensitive a character is the more their arousal is impacted by what they see and do
Fer	Fertility	Fertile characters find it easier to get pregnant, or get someone else pregnant
Aro	Arousal	Highly aroused characters may better seduce others but may also succumb to their advances

> Dev Note: The sexual aspects of the game are not fully fleshed out at the moment.

2.4.2 Size

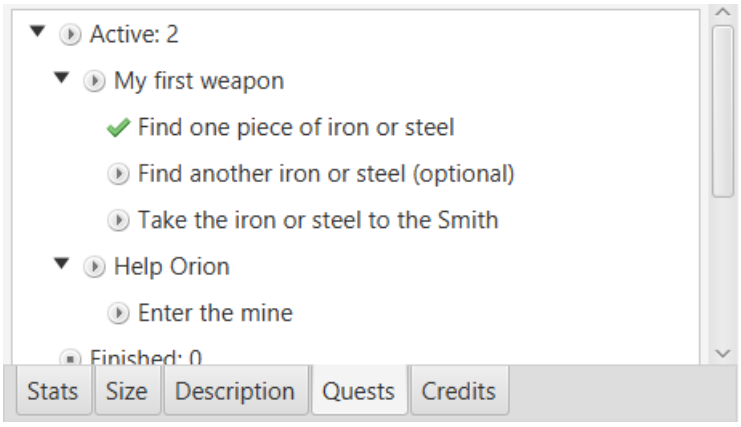
If you want to know how big your character is numerically you can see this here. If something has changed since your last turn it will be marked with an up-arrow (bigger) or down-arrow (smaller). You can change the units with the flags drop down in the menu.

2.4.3 Description



A textual description of your character and their clothing. This is the same output as you would get in the story pane from looking at yourself. As transformation can be a large part of this game this is provided as an alternative to doing that.

2.4.4 Quests



The quests tab is a handy reference to which quests you are currently on and what you need to do next. It is divided in to two lists of quests: `Active` lists (with the play button icon) the ones you have started, and `Finished` lists the ones you have completed (marked with a tick) or failed (marked with a cross).

Each quest has various steps listed which are also marked with a tick if completed or a cross if failed. In the above screenshot the player could choose to look for more iron or steel, but that step is optional, so they could also return to the smith.

A quest fails if any non-optional step is failed. A quest is completed in all non-optional parts are completed.

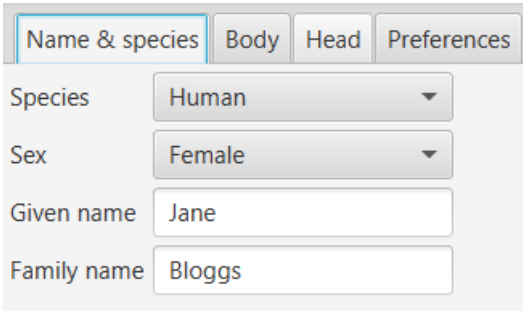
The little black triangles allow you to open or collapse a list if you wish.

2.5 Character Generation

The first step on starting the game is to generate your character using the tabs at the bottom right. You can do these in any order, but it is recommended to work left to right. You can observe the changes to your character in the `Size` and `Description` tabs above.

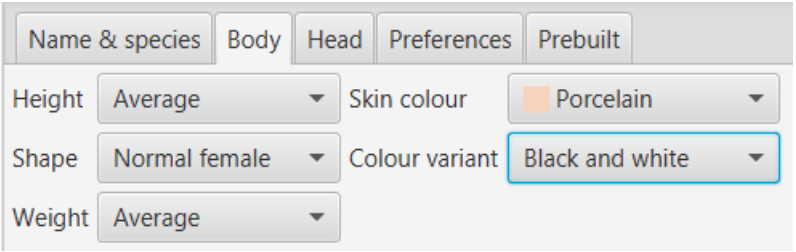
2.5.1 Name & Species

Use this tab to select your species and sex, and name your character:



2.5.2 Body

Use this tab to define your height, body shape, starting weight, skin colour, and, where appropriate, fur colour and pattern. The available body shapes will depend on the sex you have selected.



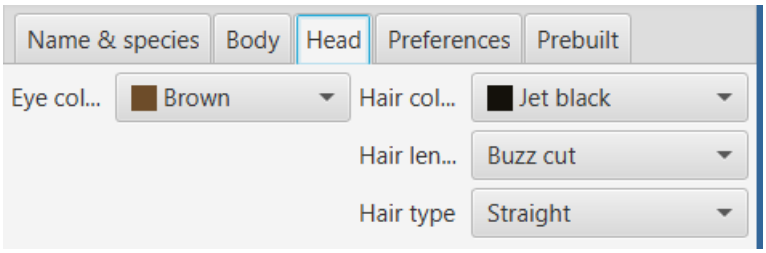
Note: If your species has fur your skin colour won't normally be visible unless there's an unfortunate incident.

Note: If your sex is hermaphroditic then you will have a choice of body shapes from both genders.

> Dev Note: More body shapes are planned. Yaffaif tries to model the body fairly accurately, but this makes adding new shapes troublesome. This is to some extent a work in progress.

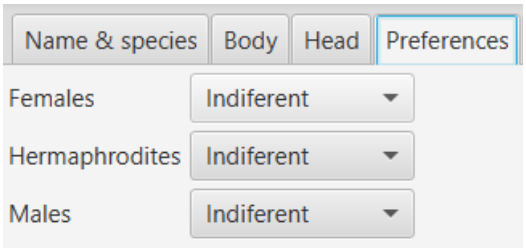
2.5.3 Head

Use this tab to change the attributes of your head and face. This is purely cosmetic at the moment and does not impact game play.



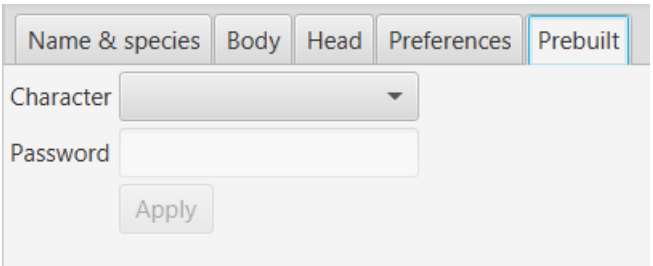
2.5.4 Preferences

Use this tab to set your character's preferences for others. Currently, this does not impact game play.



2.5.5 Pre Built Characters (Patreon build feature)

If you are using a Patreon release there will be an additional tab at the far right labelled `Prebuilt`. This allows the selection of characters created for/by people who have selected the "Custom Character" tier. If this tab is present but unavailable there are no pre-built characters commissioned yet.



To use a custom character select a character from the `Character` drop down. Some characters may be available to all, others may be password protected with a password provided by the commissioner. If a password is needed, enter it in the password field and then press the `return` key. If the password is correct its status will change to unlocked. Use the `Apply` button to select and load this character.

You can go back and adjust this character with the other tabs if you wish before using the `Finish` button..

2.5.6 Finishing

Once your character is to your satisfaction click on the `Finish` button at the bottom right to start the game.

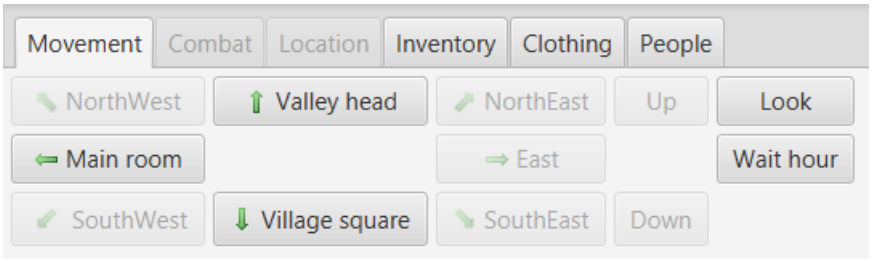
2.6 Control

Once character generation is complete this bottom right pane now becomes the main way of having your character do things in the game.

It has tabs for movement, combat, location, inventory, clothing, people, and shops. A tab that is grey and cannot be selected has nothing on it.

Most tabs feature a list of items with a drop down menu for each item giving you options of what to do with it. Left click the drop down to see the list, then click on the action you want to take. If you choose not to take one of the actions, just click outside the list to close it. Many of these menus and buttons have tool tips that you will see if you let the pointer rest over them.

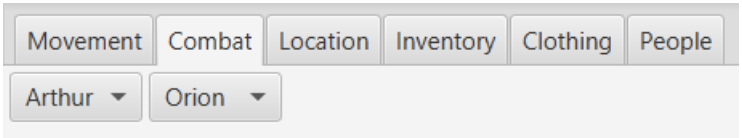
2.6.1 Movement



The movement tab allows the player character to move around the map. There are the normal compass points to the left, and up and down to the right. This tab also has buttons to look at the current location and wait one hour in-game.

The movement buttons can either show the direction, or if the character knows, or can see, where that direction leads the name of the location it goes to. The tool tip for these buttons describe the exit.

2.6.2 Combat

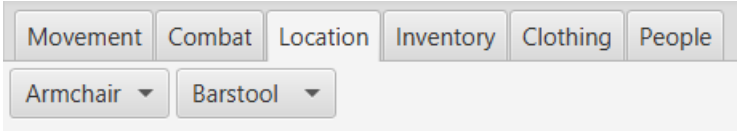


This tab allows you to engage in combat with any people or animals in the location. Click the drop down and choose to fight. The game doesn't stop you from fighting friendly

characters, like Arthur here; just realise he won't be friendly for long. Of course, before fighting you probably want to find a weapon in your `Inventory` and wield it!

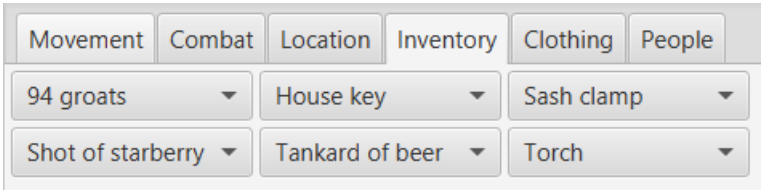
If you've equipped a weapon you can throw at an enemy, and throw it the game will automatically select the next weapon of the same type for your next attack.

2.6.3 Location



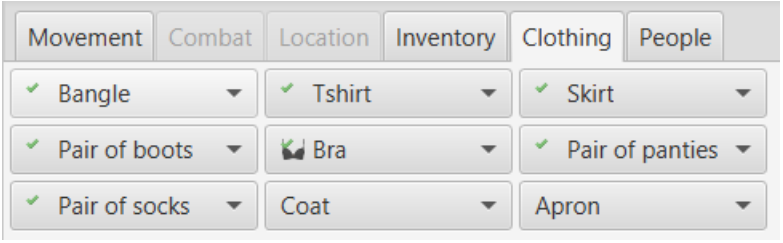
Here you can interact with things that are in the same location as you are. If there's nothing where you are this tab will be unavailable.

2.6.4 Inventory



This tab allows you to perform actions on the things you are carrying. Your clothes are in the next tab, `clothing`.

2.6.5 Clothing



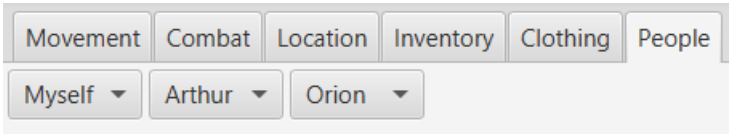
The clothing tab is an inventory for clothes. Each drop down control allows you to select what to do with that item. Worn items appear before others, and broken items last. Within this they are ordered outer garments first, then head to toe, then well fitting to worse fitting.

Little icons to the left of the name of the item indicate its current state:

- A green tick means the item is being worn.
- A red cross means the item is broken and cannot be worn.
- One downwards triangle means the item is going to be tight. A second means really tight, and a third means it won't fit.
- One up triangle means the item will be too loose. A second that it will be really loose, and a third may mean it is too big to wear.

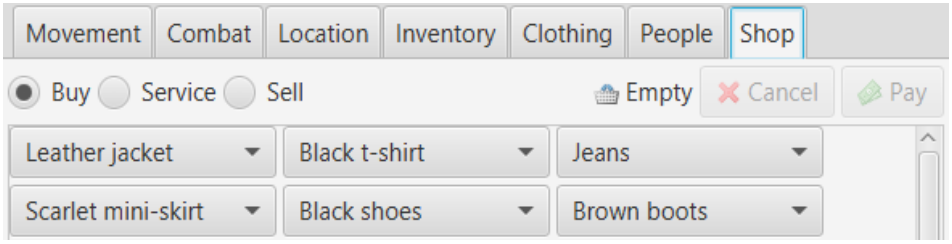
> Dev Note: I'm experimenting with adding an icon for the garment too, like the bra here, but I'm thinking it's too fussy.

2.6.6 People



This tab allows you to interact with yourself and other people in your location. Look at them, talk to them, buy things, and so on.

2.6.7 Shop



This tab allow you to buy items, services, and sell things in shops, depending on what the shop offers. To add things to your basket use the item drop-down menus. You can also remove things this way. Tool tips on these menus describe the item. To purchase the items in the basket click on the pay button. The cancel button empties the basket. All purchases are final.

2.7 Command

You can enter your commands in this text box underneath the story pane. The text box has an auto-complete feature so that you need not spend as much time typing (and spelling) commands. Click on one of the offered completions to select it, or you can use the down arrow key to highlight one of the entries and press `return`. The selection will replace the contents of the text box. You can then continue typing or press `return` to submit the command.

The command text box also remembers the last few commands. Use the up arrow key to access earlier entries in your command history. Use the down arrow key to access later ones, or return to the current command.

The game's ability to accept textual input sometimes lags behind the control interface. It can recognise most of the typical verb-noun commands. Nouns may also be replaced with pronouns such as "it". Actions performed with in the control area do affect the text parser, so if you click on `Beer` in your location tab and choose `Get`, then type `drink it` you will drink the beer you just picked up.

If you want to talk to another character you will find forms like `ask Grimtooth about trap`, `ask Doctor for jelly baby`, or `tell Egon about ghost`.

You can copy (highlight and `ctrl-C`, `command-C`) and paste (`ctrl-V`, `command-V`) in the command box. Right-clicking on the box also brings up these options in a list. Note: the `Undo` option reverses the last text edit, not your last action in the game.

The command parser understands some commonly used text adventure abbreviations:

n	go north
e	go east
s	go south
w	go west
u	go up
d	go down
i	inventory – list what you are carrying
l	look – on it's own refers to the current location
x	examine – search or look closely at something
z	wait

The text commands `load`, `save`, and `quit` do the same actions as the buttons on the `menu` at the top.

> Dev Note: I chose *not* to implement the abbreviation q for quit, as it's just too easily caught instead of w for west – at least on a Qwerty keyboard. I've lost too much progress this way playing other games!

The command field is also where console commands can be entered in 7.5 Console commands.

2.8 Errors and Warnings

The game is still in development and won't be perfect. When something unexpected happens in the engine, it attempts to report it to the Story Pane. There are three types of errors:

Warnings: are reported when something unexpected happened but is not likely to impact playing the game.

Errors: are where the problems are more serious, they may impact your ability to play the game as they will leave the game in an unexpected state.

Fatal: are the worst kind of fault: a Java Exception has occurred in the code and wasn't handled by the engine. Fatal errors may well leave the game in an unplayable state, or the game may behave very strangely afterwards.

If you encounter an error or warning please save the game to a new file – this save game will include details of the exception and the events that lead up to it. Please upload the save game to a file sharing site, and send me a link to download it so I can investigate the problem.

2.9 Initial Steps

At this point you are free to play the game as you wish. Wander about, look at things, pick stuff up, talk to people. If you need to heal, eat something, find a potion, or sleep. However, you may want to begin by going west into your kitchen, opening the cupboard, taking the bread knife, and wielding it.

The remainder of this guide contains a description of the internal mechanics of the engine, and walk though for each of the quests. In general there are different ways to approach completing each quest, feel free to experiment!

3 Mechanics

This section describes the internal mechanics of the engine, as you might find in the player's handbook of a tabletop role playing game.

3.1 Turns, Time and Actions

Yaffaif is a turn-based Role Playing Game. Until you select something to do, an action, nothing is happening. There is no real-time aspect, so you can walk away, come, back, and the game will be in exactly the same state as you left it.

3.1.1 Actions

An action is something you chose for your character to do, either through the `control` pane, the `command` text box, or the `menu`. Most actions take some time to complete, but some are considered instantaneous, like looking at things, checking your inventory, and saving the game.

> Dev Note: In debug mode Console commands and changes through the Debug screen don't count as actions and are instantaneous.

Actions in Yaffaif are deliberately "coarse grained". That is they may cause several steps to occur in sequence. For example if you choose to go north through a locked door then this implies that you unlock the door, open it, pass through, and optionally close and lock it behind you (depending on the door). This saves having to be needlessly precise about simple tasks.

The same actions you can do can also be taken by NPCs as part of their turn.

Choosing an action starts a turn.

3.1.2 Turns

Broadly speaking the following happens when you start a turn by choosing an action to do:

1. Your action is performed and the internal state of the game is updated.
2. NPCs that can see what you have done may react and take actions themselves.
3. Other NPCs can react to what those NPCs did as well, and so on.
4. You act on any compulsions you may have acquired. If so, repeat from step 2.
5. The in game time advances by the time it took to you perform the action.
6. NPCs may use the time that has passed to do various jobs, taking actions themselves.
7. The engine builds a list of all the actions you can do next.
8. The story is updated with all the things you saw happen.
9. The action list is used to control which buttons and drop downs are available and allow auto completion of commands is updated.

3.1.3 Time

The game keeps track of the days passed, time of day and the day of the week, and this is displayed in the `menu`. The NPCs in the game may do different things at different times of the day, or on different days of the week. When they've nothing particular to do they idle, often wandering around within their bounds, eating and drinking and so on. They can also pick things up that are lying around, so do not leave anything you want to keep where an NPC might find it.

> Dev Note: Idle NPCs are my garbage collectors, they can remove spare items from the game world entirely!

3.2 Experience and Levels

Experience points (xp) are gained through beating opponents in combat and completing quests.

3.2.1 Quest experience

The experience reward for a quest is fixed, and awarded if and when the quest is completed successfully. Only the player character can gain experience this way.

3.2.2 Combat Experience

The experience through combat if you strike the killing blow is one third of the opponent's hit points, rounded, plus 1. So if you beat an opponent with 5hp, you'll gain 3xp. If you've fought the opponent, but don't strike the killing blow, this value is halved, but rounded up, so in the above case you get 2xp.

Both the player character and NPCs can gain experience through combat.

3.2.3 Levelling Up

Once enough experience has been gained the character levels up. This has not been tuned yet, but the additional experience required goes up with each level as the table shows:

Level	Experience
1	0
2	100
3	250
4	450
5	700

The benefits of levelling up are:

- Increase maximum hit points between 3 and 7hp, so you are harder to kill
- Increased chance to hit an opponents

- Opportunity to learn new skills (TBD)
- Better saving throw chances

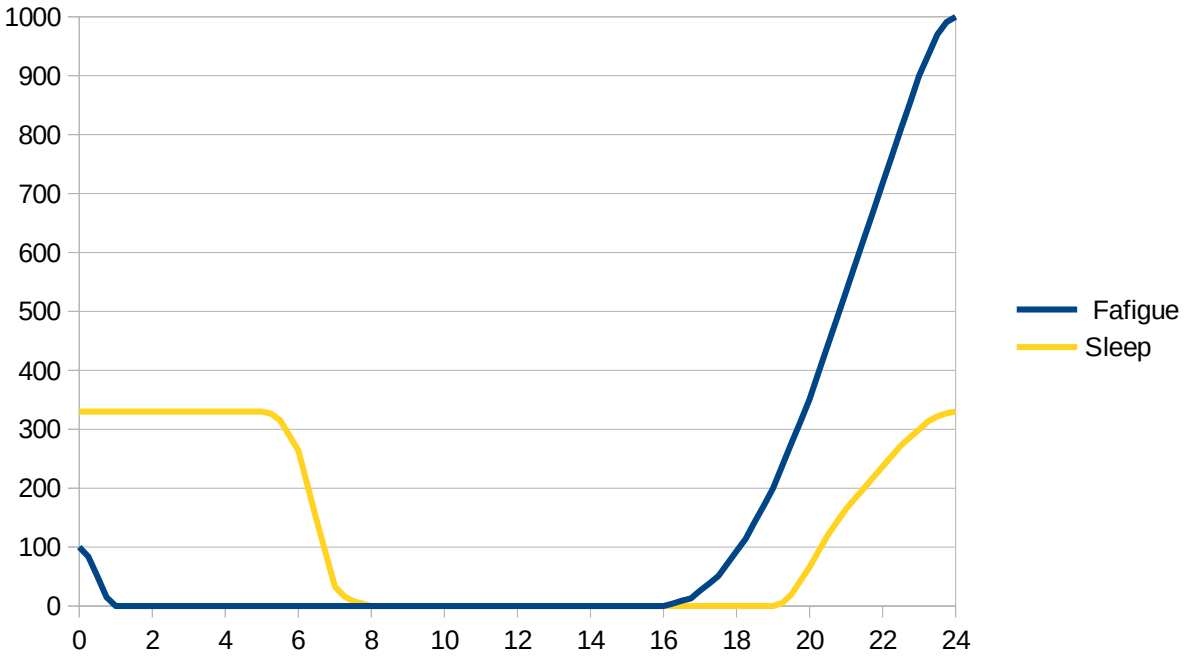
NPCs can also level up if they gain enough experience.

3.3 Sleep and Fatigue

Characters need to sleep regularly to refresh themselves and recover from injuries.

Total fatigue is a measure of how tired you are. It is calculated from two factors. The first, and most important, is how long you have been awake compared to the norm for your species. The second, lesser, factor is the normal sleep pattern for your species, which could be diurnal, nocturnal, crepuscular (dawn and dusk), or some other pattern.

The chart below is for a typical diurnal species. The fatigue line is the first factor and is shown in hours since you woke. Sleeping will reset this back to 0 hours. The sleep line is the second factor, and depends only on the hour of the day. Summing these gives your fatigue score.



You can also gain a bonus from sleeping in a good bed, or a penalty from sleeping somewhere uncomfortable (like on the floor) or having a bad dream.

Low fatigue can give a bonus to combat, and high fatigue a significant penalty.

3.3.1 Dreams

When you sleep you may dream, and find yourself in unusual, unexpected, or all too familiar situations which may be complete nonsense, give insights into the past, or presage the future. You, the player, remain in control of the dreaming player character during the dream. Dreams will come to an end when:

- A certain amount of time has been spent in the dream world, or

- Some optional hidden condition has been met, or
- The dreamer dies in the dream.

It is always possible to just wait for the dream to end.

In general what happens in the dream has no effect on the player character when they wake, however:

- Particularly unpleasant dreams may cause a fatigue penalty during the following day.
- The player may learn of things that are relevant to the player character.

3.4 Combat

Combat largely follows the traditional RPG rules, but instead of rolling, say, a twenty sided dice (d20) the engine works with probabilities to 4 places of decimals – or a thousand sided dice if you'd prefer.

The same rules apply to the player, animals, monsters, and NPCs. In each case you make a roll to determine if you hit, and then a roll to see how much damage you do. When an opponent's hit points are reduced to zero they are marked as dead or unconscious and are beaten.

Opponents (including the player) can also be beaten by maxing out their arousal during combat.

> Dev Note: There are no specific arousal based attacks in the game at the moment, but this mechanic does work, and arousal builds slowly over time, so take care to go into combat with a clear head.

3.4.1 Hit Points

Hit Points (hp) are a measure of how much damage something can take before it becomes unconscious or dies. Your character and everything you might fight have hit points. Hit points can be recovered by sleeping, eating, or by drinking potions.

3.4.2 Combat Turns

During combat each turn takes 1 in-game minute. Some actions may take several combat turns to complete, but simple attacks and other actions take just one.

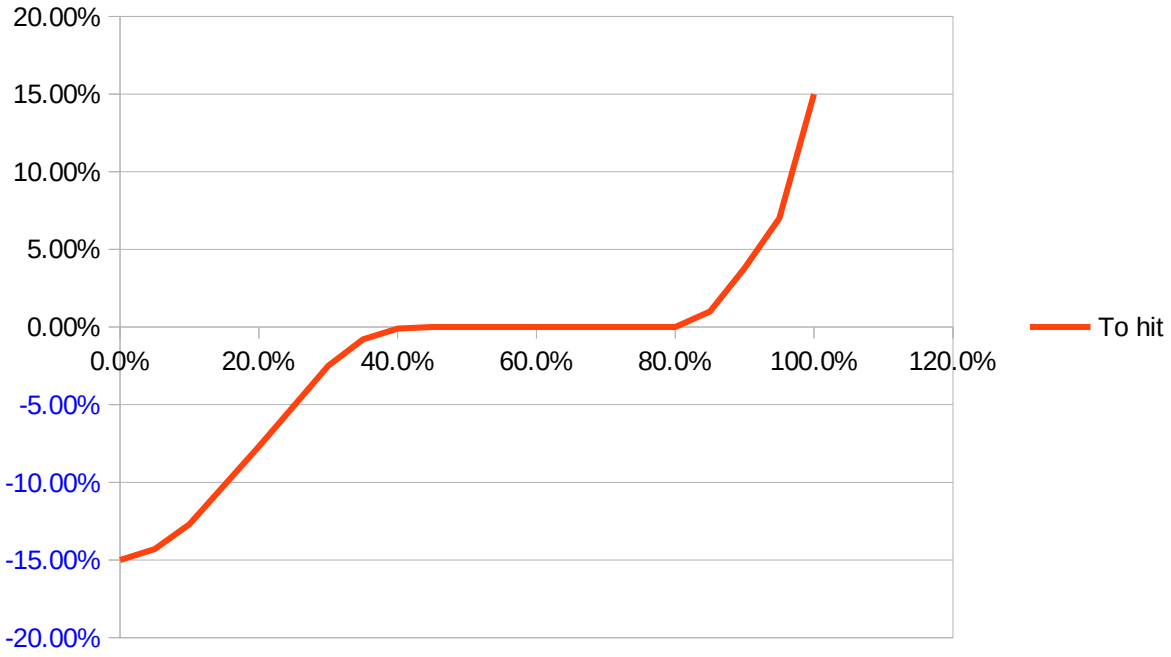
During combat everything is focused on the combat location. NPCs who are not involved do not perform their normal time-based jobs described in Turns above. Once combat is complete the time it took passes in other locations and those NPCs can act.

3.4.3 To Hit Chance

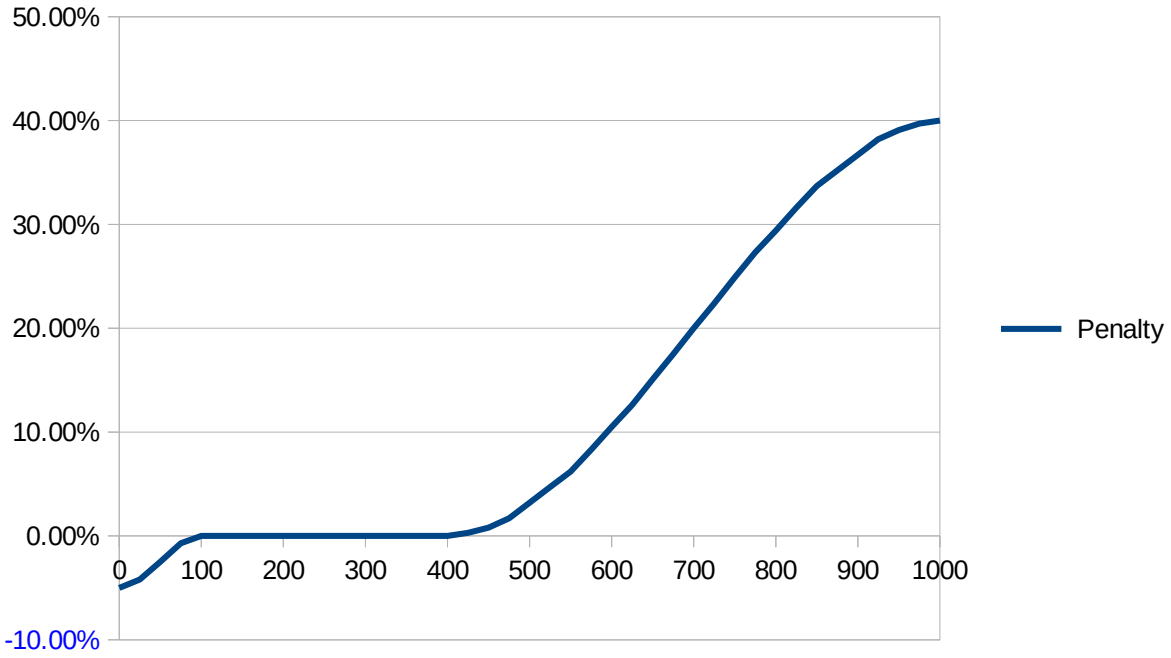
The probability of hitting an opponent is determined by several factors.

It begins with a base probability of 50%. For each level of your experience you gain a 5% bonus, plus any attack bonuses that may apply from magical items. The opponent's armour class percentage is then subtracted.

Depending on your strength you will receive a bonus for particularly high strength (above 80%) of up to 15% or a penalty for low strength (below 40%) of up to 15%. This is determined by a smooth spline curve (rather than the traditional tables):



Depending on your fatigue you will receive a bonus of up to 5% (energised) or a penalty of up to 40% (collapsing). This is determined by a smooth spline curve:



If your opponent is aroused you gain a 20% bonus, if they are more or less incapacitated by lust that increases to 30%.

Your own arousal decreases your probability of hitting by one twentieth of your arousal stat. So, if your arousal is 40% you receive a 2% penalty.

For each hypertrophic body part you have you also lose 2%, if it is incapacitating you lose 5%.

Your intoxication also affects your probability of hitting. If you are tipsy you lose 10%, drunk 15%, emotional 30%, and paralytic 40%.

What ever the determined hit chance works out to there's *always* a 5% chance of missing (rolling a 1 on a d20), and a 5% chance of a "perfect" critical hit (a natural 20).

3.4.4 Damage

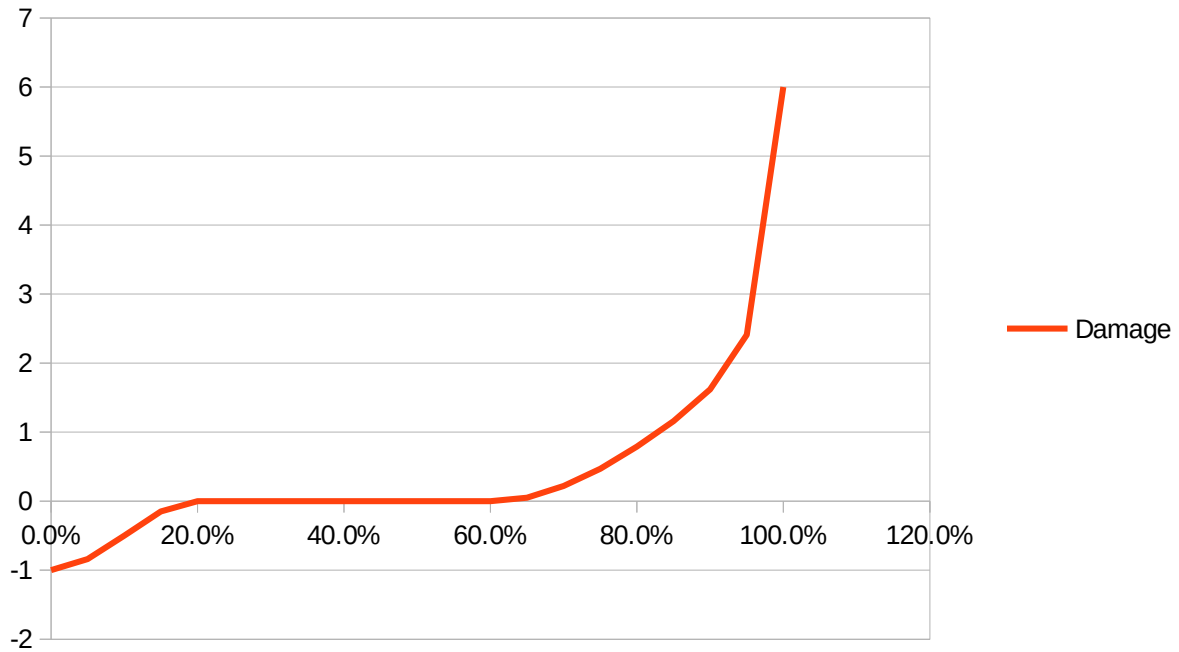
The number of hit points of damage done by a hit is determined by the weapon you are using:

Weapon	Damage
knife	1d3 + 1
dagger	1d4 + 1
sword	1d6 + 1
broad sword	2d4 + 1
long sword	1d8 + 1

> Dev Note: These are slightly different to the standard rules. As part of the testing I run Monte Carlo simulations of various encounters and had to tune the damage values to favour players.

Some weapons are magical and can have enchantments that add to these values. They can also have secondary damage, for example burning from a flaming sword.

Depending on your strength you will receive a bonus for high strength (above 60%) of up to +6 damage or a penalty for low strength (below 20%) of up to -1. This is determined by a smooth spline curve:



If you score a critical hit these damage values are doubled.

Opponents can have natural or magical immunity (decreasing damage) or susceptibility (increased damage) to certain types of damage.

For elemental damage the opponent can halve the damage if they make a saving throw. For projectile weapons they can halve the damage by making a save vs dexterity. And for transformational damage they can avoid it entirely with a save vs petrification or polymorph.

3.4.5 Saving Throws

Saving throws are a chance to avoid or mitigate the consequences of something that happens to you. There are two types. One is a check against one of your stats, the others are level based.

3.5 Potions

Potions can have temporary or permanent effects. Most will only last a few hours and can give temporary advantages (or disadvantages) that will wear off, though some may leave some minor changes behind.

Transformational items make permanent changes to your body (or to whoever you use the potion on). These can, however, be reverted using potions.

3.6 Body & Weight Mechanics

The engine has a model for the changing body weight of the player and NPCs and this is one of the transformation themes. Eating and drinking provides energy, actions use it up. Eat more than your body needs and a character will gain weight. Don't eat enough and you'll lose weight. Just as in the real world these processes occur slowly.

3.6.1 Body Shapes

The body shape you selected during character creation controls where on your body you gain weight. Various potions are available to change where you tend to gain.

> Dev Note: More body shapes are planned, but they take some tuning to work well and are a little time consuming to do.

3.6.2 Starving

It just isn't possible to starve to death in the game. So if you aren't interested in weight mechanics, just don't eat and you will remain thin – unless you need to eat to heal and recover hit points.

3.6.3 Clothes

As your body changes shape, either through weight gain/loss or other transformations, you may find your clothes won't fit as well any more. All clothing has some give in it, depending on the material it is made from. You can tell when the fit is changing through comments in the story, and icons on the clothing tab. Unless you want to deliberately destroy something, it is probably best to remove it before it gets too tight.

When you buy clothes or find clothes that haven't been worn by someone else they'll fit your current body shape. If you need to have clothes altered to fit better, find a shop that does alterations.

> Dev Note: There's no real penalty at the moment for running around naked either – clothing is largely cosmetic other than those items that work as armour.

3.6.4 Conservation of Mass

If you encounter a transformation that changes your height then it won't also change your body mass. If you get shorter you'll get fatter, taller and you will get thinner.

3.6.5 Hypertrophy

If part of your body becomes unreasonably large you may find it restricts your ability to get around, or that you can't fit through openings such as doorways. You should have plenty of warning that this is going to become a problem in the story pane.

3.7 Gods, Fetishes, Shrines, and Karma

In the world of Yaffaif the gods may occasionally act in your favour, or at least *in their own interests* through you – which might benefit you, possibly not immediately. In general they will only act if you've a sufficient karma balance with the god in question.

These divine actions may be subtle, such as a feeling that you should take a specific course of action, or more direct intervention - including transformations. In general, again, these divine actions will reduce your karmic balance with the god in question.

3.7.1 Fetishes

A fetish (the religious kind) is an object that can be carried or worn and is a focus for a particular god. Carrying a fetish will produce a regular overnight increase in karma for that god. If it is carried by a person then it affects them alone. If it is placed in a room it will distribute its karma amongst those present.

3.7.2 Shrines

A shrine is another kind of focus for a god where you can make monetary donations, increasing your karma. Some shrines may also accept other kinds of donations.

3.8 Transformations

Some actions in the game can cause you to transform. These can include potions, weapons, and sexual encounters (TBA).

Transformations are gradual and the sequence can depend on the form you are transforming into. In general they proceed as follows:

- Gaining the distinguishing characteristics of the new form.
- Adjusting your stature to the new form.

- Losing the distinguishing characteristics of your old form.

Transformations do not wear off, like potion effects do; they are permanent, unless you begin to transform into something else, or find some way of restoring your original form.

NPCs in the game may react differently to you as your form changes, including being friendly or hostile to you. NPCs who don’t know you will tend to address you based on your apparent gender based on your perceived body and clothing (or lack thereof).

3.9 Skills

Both the PC and NPCs can gain skills in various techniques and trades. Having a skill may allow additional options in various circumstances.

In general skills can be gained and advanced through study, training, or practice. Advancing skills can make existing functions easier, and more tricky ones available.

Currently the only skill in the game is lockpicking.

Spoilers Ahead!

4 Village Area

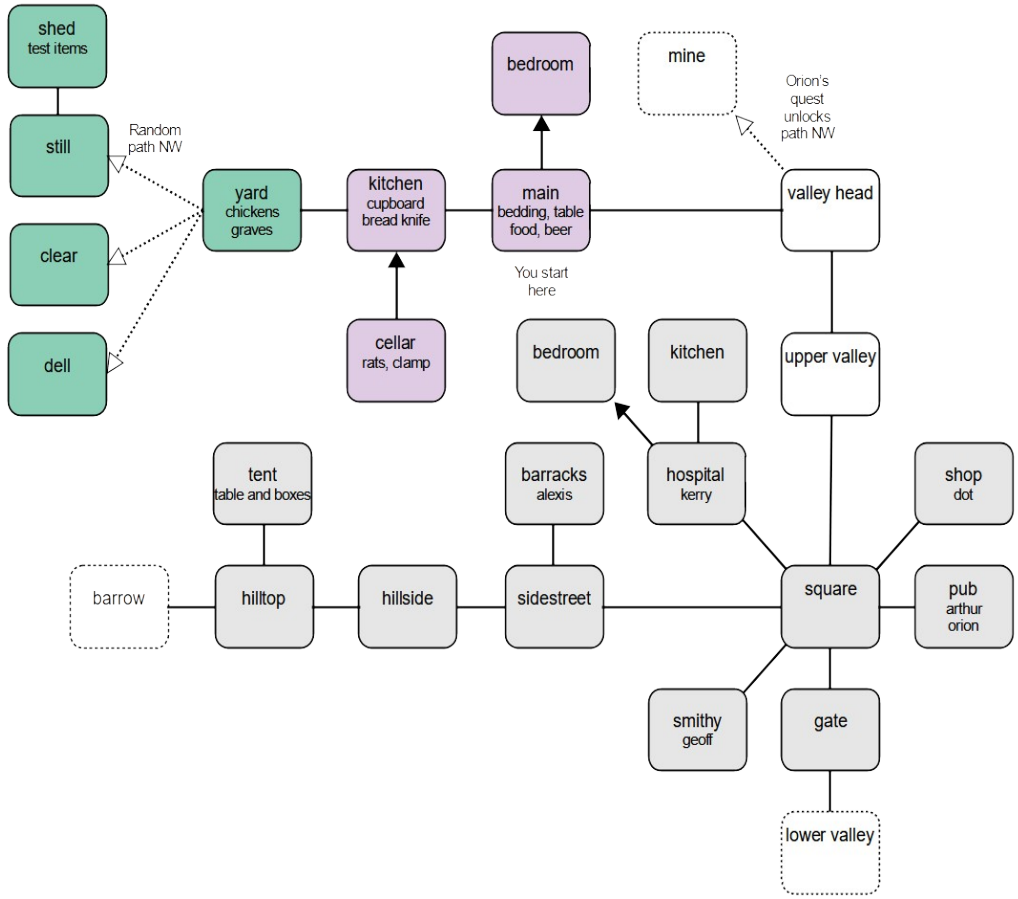
The introductory village area is partly a tutorial level. It isn't possible for your character to die in this area.

4.1 Village

The village you grew up in.

4.1.1 Maps

To orient you here is a map of your house, the forest, upper valley, and village:



Key:

Blue:	your home
Green:	forest area
Grey:	village
White:	valley (dotted areas are covered in other maps)

4.1.2 Characters

You	the protagonist

Alexis Ward	a guard 09:00 – 17:30 guarding the gate 23:30 – 08:00 sleeping
Arthur Butterworth	the barman in The Roses
Dot Drostock	an elderly seamstress and shop owner 09:00 – 17:30 shop open 21:00 – 07:00 sleeping
Geoff Smith	the village smith 07:00 – 12:00 available for training
Kerry Sidbothom	a healer and nurse 09:00 – 18:00 working in the hospital 23:30 – 08:00 sleeping
Orion Griffiths	the stranger in The Roses

4.1.3 Notes

- The paths in the woods are difficult to follow and bring you to a “random” destination each time.
- The shed in the woods is contains a number of test items for messing around. These re-spawn if you leave the shed for an hour. The shed is part of the story, and it’s contents will be changed when it is properly integrated.
- Sleeping in your own home gives you a minor fatigue advantage, compared with sleeping elsewhere.
- You can work in the bar to earn money, or food.
- You can work in the smithy to build your strength up to a point where your to-hit penalty is negligible. You will need to buy an apron from Geoff and wear it before you can work.

> Dev-Note: The intention is to have a skill system for different professions, and this would be one of the events that advances that.

- The smith’s apron also provides some (minimal) armour.
- Clothing can be bought from Dot in the village shop. The boots she sells come with a minor armour benefit, and every little helps.
- Dot also offers laundry (to clean your clothing), alterations (to make clothing fit you), and repairs (to fix broken or damaged clothes). These services take some time, and you will have to hand the items over for the duration, so you may want to buy some other items first to wear in the meantime.
- Dot will also buy clothing you no longer want. These second-hand items are put on sale in her shop for a while.
- Kerry sells the healing potions she makes during the day.

4.1.4 Quests

4.1.4.1 My First Weapon (required)

The goal here is to obtain a real weapon to replace the bread knife you can find in your kitchen cupboard and 50xp. It is suggested that this quest be started before progressing Orion's Quest as you'll need to enter the mine to complete it.

You'll need to complete this quest before you can travel south of the village.

1. Start the quest by visiting Geoff Smith in the smithy and talk to him about weapons and how to have him make one.
2. You'll need to collect 1 to 3 pieces of iron or steel.
3. Start Orion's Quest and enter the mine.
4. Grab a torch from the first wall sconce you come to.
5. (Optional) Retrieve the old ladle (`iron`) from the cauldron in the `Armoury`. This will cause a partial kobold transformation.
6. (Optional) Backtrack to the dark tunnel NW of the mine entrance. Now you have light you can see a route to the NE. Go this way, fall into a pit with some eels (kill them if you wish). Take the `iron` grating and leave to the East. Go up to enter the bar, then West, then SW twice to get back to the dark tunnel – you may fall through the trapdoor again, just repeat these steps. Depending on your status with the kobolds and the time of day you may be attacked by them.
7. Return to your home with the torch, go into the cellar, and collect the sash-clamp (`steel`).
8. Return to the smithy and ask Geoff to forge you a weapon. Depending on how many of these items you've collected he will make you a dagger, a sword, or a broad sword. The more items you collected the better your weapon and the more damage it will do. This completes the quest.
9. Equip your new weapon!

4.1.4.2 Orion's Quest (optional completion)

You will need to start this quest in order to gain access to the mine area for the required `My First Weapon` quest. However, completing it is entirely optional, but you'll get 50xp and some cash.

Part 1:

1. Start the quest by entering The Roses, and introduce yourself to the stranger in the pub. Keep talking to him until the quest starts.
2. Leave the pub and head north to the valley head. Now your attention has been drawn to it, you can find the mine entrance. Go inside.
3. Head NW until you meet Rochaine. Introduce yourself and take her little tour if you want and investigate as you go. Take the fertility statue if you wish (see notes above). Get the confectionery at the end of the tour if you wish. Before you enter the throne room there's a choice of approach you need to take. Rochaine is a mischievous air-head and she's leading you into a trap.

> Dev-Note: I wanted to establish that not all NPCs in the game can be entirely trusted!

The two sections below outline the basic approach, but there are more ways to do this.

Death and Glory Path:

1. If you are going this route, I'd recommend completing the `My First Weapon` quest at this point as having a real weapon will put you in a better position for what comes next. It is worth buying the apron from Geoff, and the boots from Dot. Be sure to put these on.
2. Wield your weapon, if you are carrying it, and go N into the throne room. Fight and kill everyone. Of course, you may be knocked out, re-awaken in the hospital, have to heal up, then go back to the mine and try again. Oh, and it turns out the queen isn't even in the throne room.
3. Work your way around to the royal bed chamber. The first time you enter the queen offers to parley. If you want to accept the offer, talk to her and her companion, Arawan. Or you can kill her in combat, or kill them both.
4. If Arawan is still alive, talk to him and get him to return with you.
5. Return to Orion.

Diplomatic but Transformational Path:

1. Don't enter the throne room through the N door!
2. Check your description and make sure you have some visible Kobold transformations. It will be easier to get past the other kobold NPCs if you do. If not go back to the Treasure Room, or to the Armoury and eat the custard.
3. Grab one of the torches from the wall sconces, and back track to the dark tunnel.

4. Take the path NE that you couldn't see before. You'll walk out onto a bridge and fall through a trapdoor into a water filled pit. If you return to this area then you'll fall though again unless you are a similar weight to a kobold, or make a save vs dexterity.
5. Optionally fight the eels, but do pick up the grating for the `My First Weapon` quest.
6. Head E then UP and make your way to the `Royal Bedchamber`. Talk to the Queen and her companion.
7. Optionally take the offer to join the tribe. If you do you will become part of their faction and they won't attack on sight. You can also continue your transformation if you want with more potions from the Queen, and get free beer when the bar is manned.
8. Return to Orion; you can go through the `Throne Room` to avoid the trapdoor on the bridge.

Finishing Up:

- If Arawan is alive, and with you, he and Orion will leave together.
- If Arawan is dead, you have the option to tell the truth to Orion, or lie.
- If you've met Arawan but he's still in the mine, then answer Orion's questions, and then he'll leave.

4.1.4.3 Find Your Brother (main, unfinished)

You need to find out what has happened to your brother. At the start of the game you do not know if he is on the run, captured, or even already dead.

This is part of the main quest series and spans multiple areas.

The quest is started after your first dream about the night your parents were killed by two civilian guards come to arrest your brother.

1. Gather information by talking to people around the village. Alexis has some additional dialogue options once this quest triggers.
2. Travel to Twizelford, you'll need Alexis to open the gate first, which requires that you complete the My First Weapon quest. It will take around 10 hours to reach the town so rest up first and bring food for your journey.
3. At the start of your journey beyond the village there is a shrine. Make a donation for luck if you wish.
4. When you reach the North side of the bridge, you've three choices (see the Twizelford Road section):
    1. **Wet path:** Head East and fall down the rocks, look for a way across the stream. You can keep your clothes dry by stripping off before crossing. See the quest Get Warm and Dry (optional) 5.1.4.1.
    2. **Bandit path:** Cross the bridge and deal with a bandit, then follow your nose.
    3. **Orc path:** Head North West into the forest and follow your nose. This path takes you through Nug's cave, so you may want to do it when he's asleep. Entering the cave after midnight should work.
5. Gather information about what has happened (incomplete).

NOTE: Twizelford itself is still being written.

4.1.4.4 Village Bandits (optional)

Dealing with the bandits will get you 50xp, increased friendship with Alexis, and a peck on the cheek.

Alexis guards the village gate during the day and won't let you leave until you've got a decent weapon.

1. Ask Alexis about the gate when she is guarding it. She will mention she is keeping it closed because of bandits on the road.
2. Ask her what is being done about the bandits.
3. To proceed further you will need to complete the `My First Weapon quest`.
4. Talk to Alexis and offer to help. This starts the quest. Optionally ask where the bandits are. Ask her to open the gate.
5. You can now step outside the village, and head down the road. You will find the first bandit South of the bridge. Kill him.
6. Go West to the camp. Kill the two bandits there.
7. Return to Alexis and tell her the bandits are dead. This completes the quest.

> Dev-Note: The bandit fight is going to be difficult. It may need further tuning.

Note: There's a separate monetary reward for killing this group on the notice board in the Civilian Guard building in Twizelford.

4.1.4.5 Revenge your Parents (optional, unfinished)

Your parents were killed by Civilian Guards. It's time to deal some justice to those involved.

The quest is started after your first dream about the night your parents were killed by two civilian guards come to arrest your brother.

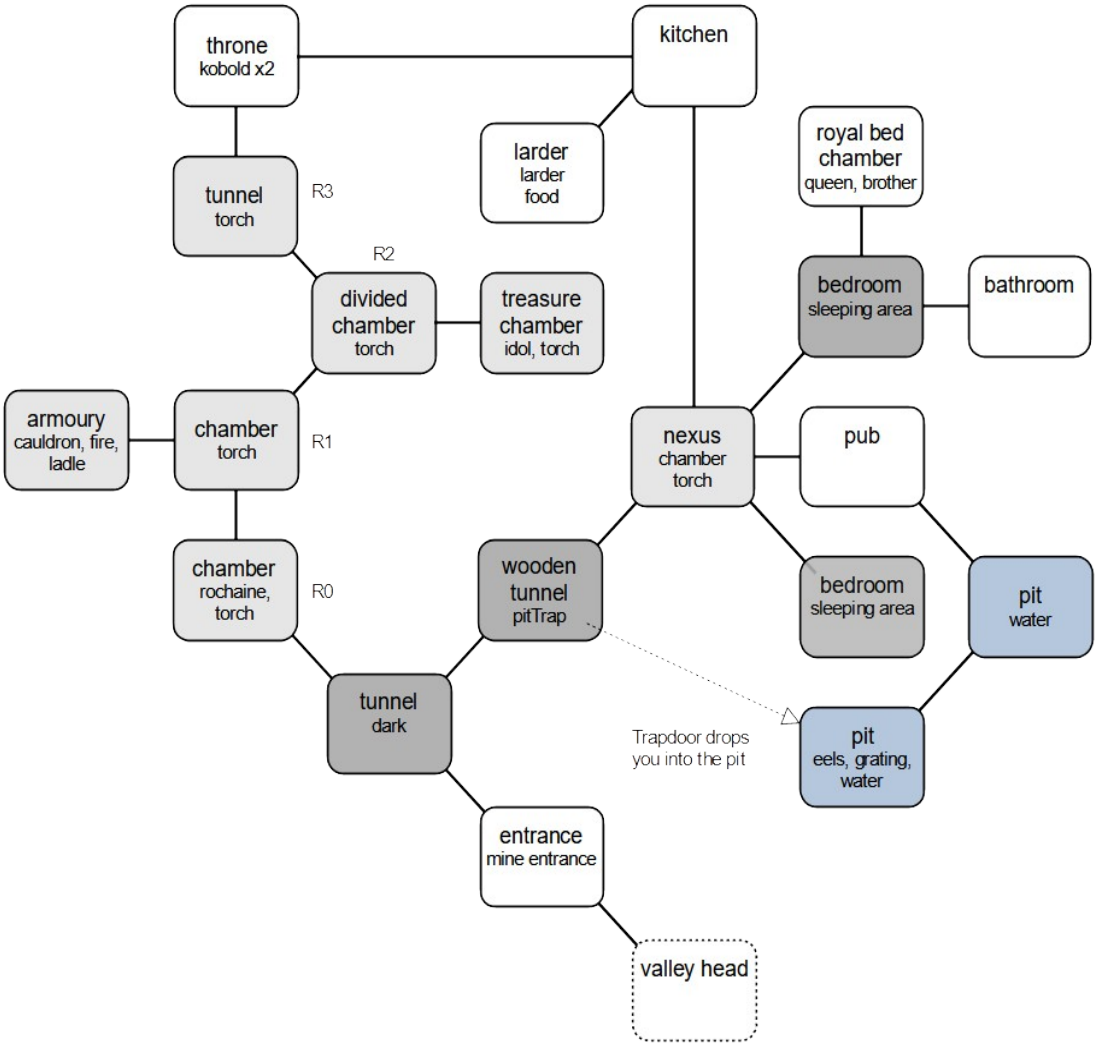
- Talking to Alexis the gate guard will reveal that two Civilian Guards left that night, one with a broken nose.
- If you talk to Kerry while she's working in the hospital she will reluctantly reveal that she treated someone called Darthen.
- Travel to Twizelford (requires you to complete My First Weapon 4.1.4.1).
- Enter the guard's building and ask Ruffus about Darthen. He inadvertently reveals Darthen's family name, and that he is a constable.
- You can also discover who Darthen is and more about him by spending an evening in Beers, and waiting until he comes in to drink after finishing his shift. Talk to Lola after he's drunk enough to become emotional.
- Once he's had a skinful Darthen will leave Beers and head to the dark alleyway to relieve himself against the wall. Follow him.
- Here, you can kill him or talk to him.

- If you chose to kill him then this completes the "Kill Darthen" quest part. Search his body for useful things, including the key to the staff areas of the Civilian Guard building.
- If you chose to talk to him then you can demand he surrender his key, various garments, and even his coin purse. Dismissing him completes the "Pardon Darthen" quest part and he will leave the game.
- In either case being dressed as a guard will give you an easier time if/when you enter the Civilian Guard building staff areas. If his clothes don’t fit you may need to get them altered. NOTE: there is currently no way of getting some of the items altered to fit you.
- At this point remainder of this quest is unfinished.

4.2 Mine

4.2.1 Maps

This is the Kobold mine area:



Key:

Blue: waterlogged areas

4.2.2 Characters

Alfred Garnett	the night guard 08:00 – 16:00 sleeping 18:00 – 06:00 on patrol
Arawan Crowther	the business partner
Dezinah Hronfelt	the queen

Frank Coke	the put-upon cook and barkeep 02:00 – 10:00 sleeping 11:00 – 13:00 cooking 17:00 – 01:00 bar keep
Gart Rogan	a guard 20:00 – 04:00 sleeping 06:00 – 18:00 guarding the throne room *
Karl Hronfelt	a guard and a prince 20:00 – 04:00 sleeping 06:00 – 18:00 guarding the throne room *
Rochaine Kalach	a “tour guide” and airhead 23:00 – 08:00 sleeping 09:00 – 18:00 “tour guide” *

* These characters will always be doing these jobs until you enter the royal bed chamber, or are attacked in the throne room.

4.2.3 Notes

- It’s impossible to die and get a game over in either of these two areas as it is intended as an introduction/tutorial. Your unconscious form will be rescued and taken to the hospital where Kerry will work her healing skills.
- In general the guards in the mine area will attack you if you don’t appear at least slightly Kobold.
- The custard is a transformative item that will gradually turn you into a Kobold and temporarily increase your lust.
- The guards can throw custard in the form of pies.
- You can make your own pies from pie cases and custard. And, yes, you can wield and throw them too.
- While he remains alive Frank Coke will gradually re-stock the larder, produce pie cases, and replenish the stew.
- The statue in the treasure room is a fetish (see 3.7.1) for the fertility god Yaridan. While it is carried or in the same room the karma with this god increases, and the following gradual changes may happen overnight (using up karma):
    - An increase in fertility
    - Breast development, and in males this may lead to female breasts forming
    - Butt size increases
    - Thigh size increases
    - For those with female bits: increases in capacity and wetness
    - For those with male bits: increases in length and girth

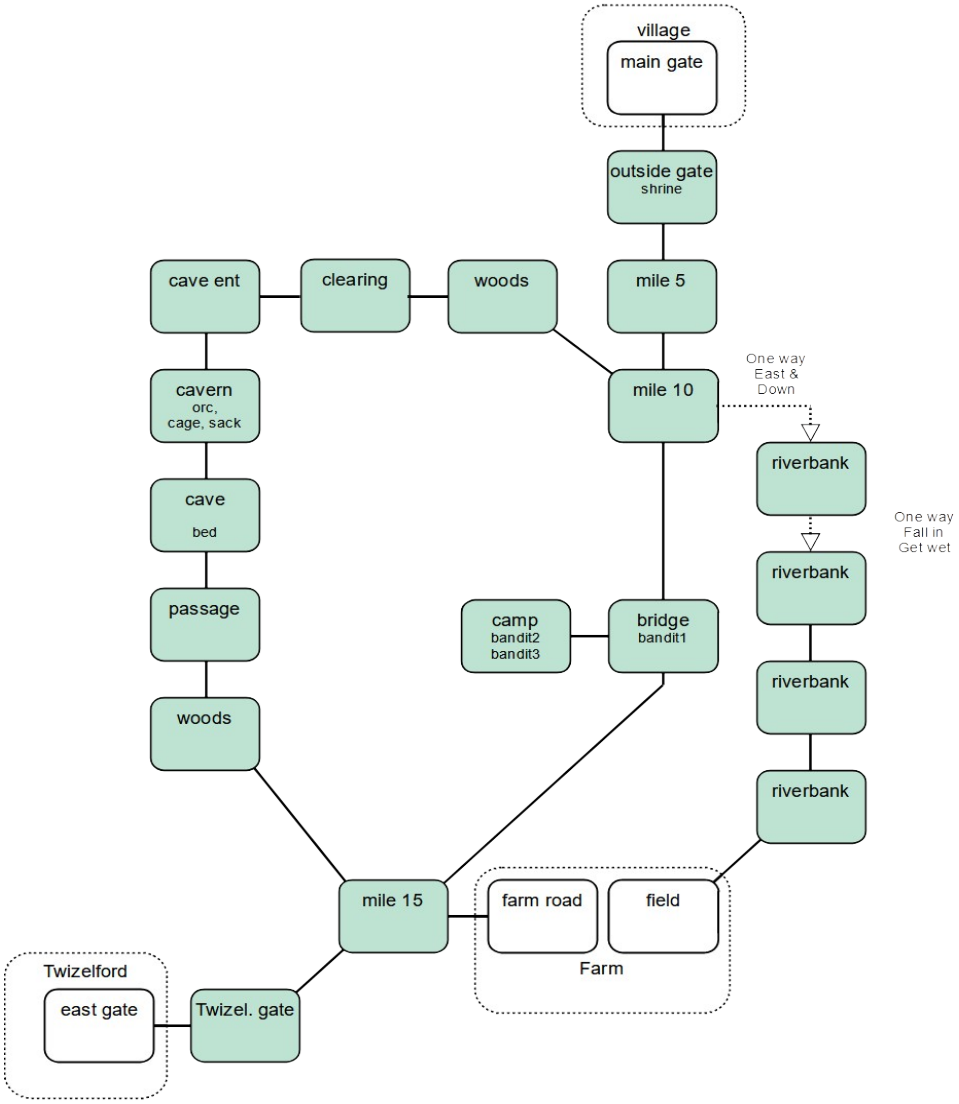
- For those with testes: increases in sperm production and volume

5 Twizelford Area

5.1 Twizelford Road

The road between the starting village and the town of Twizelford.

5.1.1 Maps



5.1.2 Characters

Dave “gimmie” Cobble	a bandit
Elise “rabbit” Cobble	a bandit
Jim “big” Cobble	a bandit

Kelsi “click” Gale	a thief Kelsi gets captured by Nug a few days after you
Nug	an orc 08:00-19:30 looking after captives 22:00-06:00 sleeping [1]

5.1.3 Notes

- The journey to Twizelford is about 20 miles (32km) so you will best to set out well-rested in the morning, and take provisions. There are three possible routes there, and two back.
- If you “die” in this area you will be captured by Nug and placed in his cage. You won’t be able to escape until you’ve met the thief Kelsi. Kelsi can teach you some basic lock manipulation skills so you can escape, which you will have to do around midnight. It is possible to get a Game Over if you don’t escape within seven days.
- If you die in this area and for some reason you’ve killed Nug, then he won’t be available to “rescue” you, so that’s a Game Over.
- There’s a locked chest with a cash reward in the bandit camp if you’ve gained the basic lock pick skill and the skeleton key.
- It is possible to sneak past Nug while he’s asleep.
- The shrine outside the gate is to Vrithan, God of travellers.
- Donating at the shrine will get you a hint from Vrithan to avoid crossing the bridge.
- Going East and falling down the rock face will leave you in a place that seems to have no exits. Look at the stream to find the crossing.
- Crossing the stream will lead to you falling in and getting soaked. Starting the quest Get Warm and Dry (optional) 5.1.4.1 below.

5.1.4 Quests

5.1.4.1 Get Warm and Dry (optional)

This quest starts if you travel East from the North end of the Twizelford bridge and cross the river, falling in and getting soaked. You really want to get dry!

This quest is repeatable, so if you fall in again it starts from the beginning. There are no particular rewards for completing it other than clean and dry clothing.

1. Fall in the river.
2. Follow your nose until you reach Bluebelle working in the field who will take pity on you.
3. Follow Bluebelle into the farmhouse, and upstairs to the guest room.
4. Strip off your wet clothing, including shoes and bangle.

1 Sleeping times are approximate. It depends to some extent on how well rested the NPC is.

5. Put all your wet things into the laundry basket. Bluebelle will take it and launder them.
6. Wait, or otherwise amuse yourself, until she returns with your clothes. There are some free items of clothing in the blanket chest.
7. Take your clothes from the basket.

It is possible to fail this quest in steps 4 and 5 if you take too long to put your wet things in the basket. Bluebelle will go back to her work in the field.

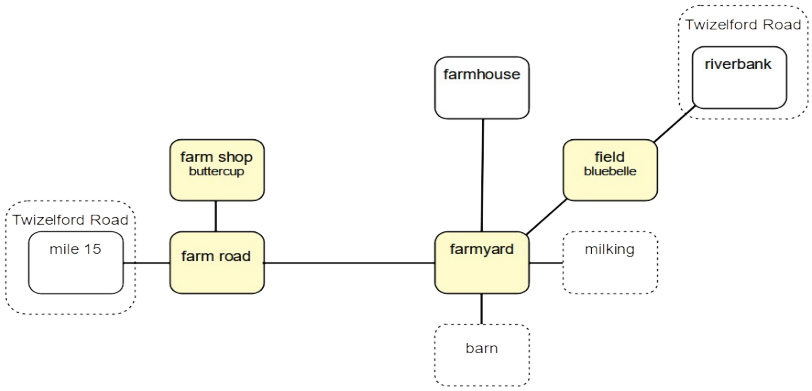
5.2 Farm

The Perfect Pastures farm lies off the Twizelford Road.

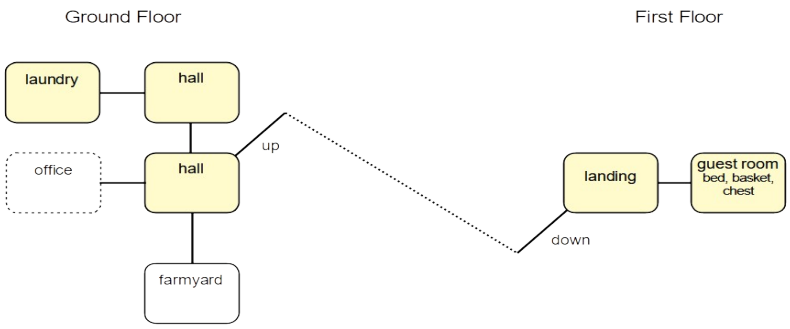
> Dev-Note: This area is subject to development in future releases.

5.2.1 Maps

Farm:



Farmhouse:



The locations here with dotted outlines are mentioned in the descriptions, but cannot be visited (yet).

5.2.2 Characters

Buttercup “Butter” Cream	a shopkeeper
Bluebelle Cream	a farm worker

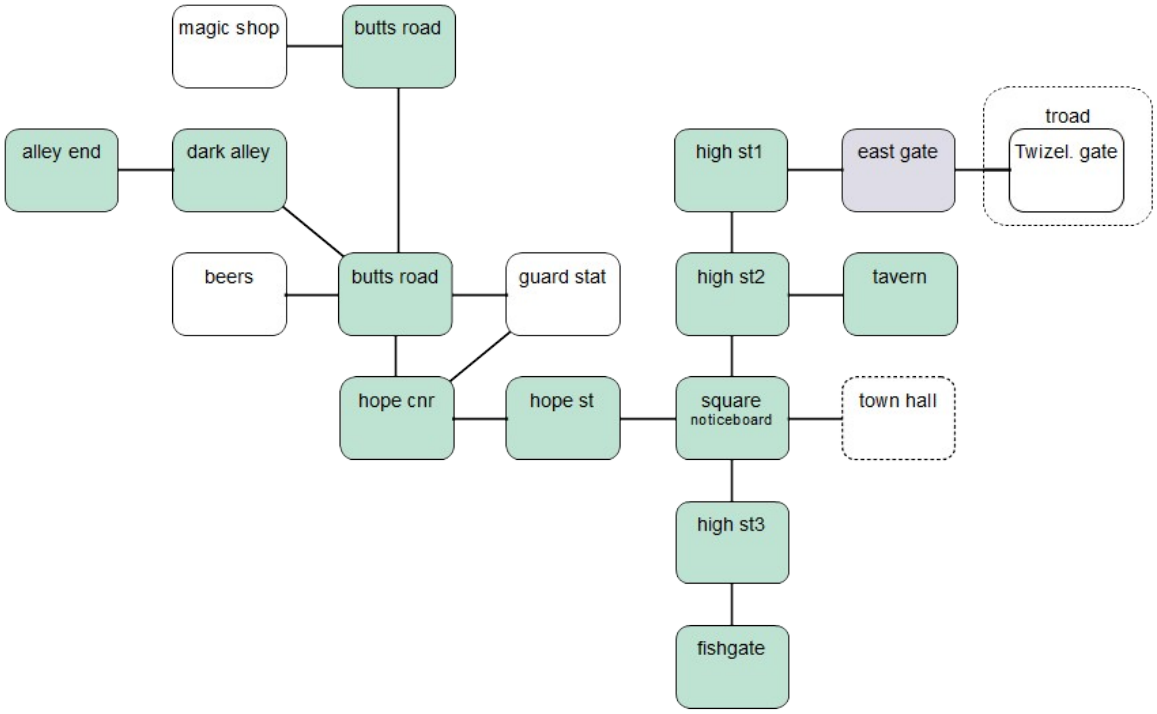
5.2.3 Notes

- If you’ve been to the farm fields then Buttercup will reveal where the dairy products come from.
- Buttercup will offer you milk straight from the source once you’ve bought (and paid for) enough of the regular bottled kind.
- Buttercup’s freshest milk is transformative.
- If you arrive at the farm from the river route, soaked, and Bluebelle is working in the field she will lead you to the guest room in the farmhouse, have you strip, and then launder your clothes. See the quest Get Warm and Dry (optional) 5.1.4.1.
- There are some emergency clothes in the guest room.

5.3 Twizelford Town

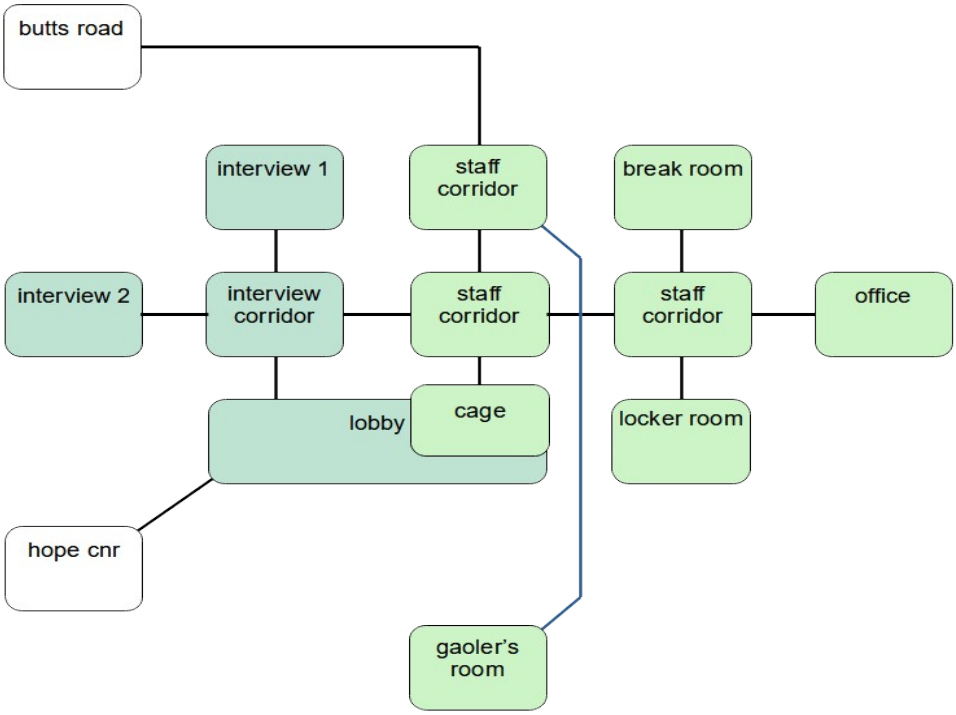
Twizelford is the town nearest to the village you grew up in.

5.3.1 Maps



The locations here with dotted outlines are mentioned in the descriptions, but cannot be visited (yet).

5.3.1.1 Civilian Guard Building



The Civilian Guard building is divided until two parts, the public areas, entered from the corner of Hope St, and the staff areas (in light green). In the current version the player cannot reach the staff areas unless the debug/cheat features are used.

The cage area is the staff-only part behind the counter in the lobby.

5.3.2 Characters

“Lady” Lola “Lilo” Lovell	owner of Beers 13:00 – 02:00 (except Mondays) serving 03:00 – 11:00 sleeping
Ruffus “Rusty” Steele	a sergeant in the Civilian Guard who often mans the front desk
Darthen Summat	a constable in the Civilian Guard 10:00 – 20:00 patrolling the streets of Twizelford
David Watts	the owner of the magic shop

5.3.3 Notes

- Ruffus will accidentally reveal Darthen’s last name and that he is a Constable.
- There’s a wanted board in the Civilian Guard building.
- If you read the wanted board then one of the postings is for the bandits on the Twizelford road. If you’ve killed the bandits then you can claim the reward from Ruffus at the counter.
- Darthen goes drinking in Beers after he’s finished his shift and eaten something.
- Various potions with permanent effects can be bought in the magic shop after discussing your needs with the owner. These include ways of reverting transformations, losing weight, changing body shape, and becoming female.

6 Dreams

This section details walk-through information for the various dreams in the game. Dreams are selected “randomly”, but may be subject to certain conditions which may be needed to unlock a dream. Dreams have a cool down period, so you won’t repeat the same dream for several days.

You can always end a dream by just waiting until you wake up back in the main game. You can also save in a dream if you wish.

In general you will begin the dream with whatever stats, skills, and equipment you had when you went to sleep. Similarly, the events in the dream won’t affect the main player character when you wake. though some nightmares may impact your fatigue as noted below.

6.1 Advent/Colossal Caves

This dream is a kind of homage to the original interactive fiction game Colossal Cave Adventure by by Will Crowther and Don Woods. This is the Yaffaif version which, more or less, follows the original 350 point version of the game. This dream is enabled by default.

> Dev-Note: This was the first adventure game I ever played, and it was on a Newbear 6800 system with only a few kilobytes of RAM, and some core mats this would be around 1978 or so. This dream functions as a proof-of-concept test for the Yaffaif engine – it should be able to handle all the original complexity while remaining playable with just the GUI and no typing. It also an experiment in how the original game plays without guess-the-verb puzzles (which I personally detest), and with the help of Yaffaif's movement mapping. Hopefully you will find it at least easier too while remaining enjoyable.

The current version is incomplete, as it is a background development process, so while it is possible to explore, not all the puzzles are yet solvable (without cheating), the NPCs are missing, and the end game does not activate.

The hints below refer to the "original" solution to the game. However, this being Yaffaif and you starting with your main game stats, skills, and equipment, you may be able to find other solutions not available in the original.

6.1.1 The scoring

You can determine your score by asking yourself what it is on the People tab.

You start the game with a score of 36. In the original you could lose points for dying and being resurrected, or by employing the hint system and various other things. These deductions don't make sense in the context of a Yaffaif dream (if you die, you wake up back in the main game after all), so they are not implemented.

The game contains a number of treasures which have descriptions with **bold** elements. Picking these up adds two to your score, and leaving them in the building scores additional points, ten or more, depending on the treasure and the difficulty in retrieving it. Getting all the treasures to the building nets you an additional 218 points.

Reaching the Hall of Mists (or other point further in) nets an additional 25 points (unimplemented).

> Dev-note: There's more to unpick here...

Finally the infamous "last point" is awarded for leaving the magazines in Witt's End.

6.1.2 The rod and the bridge

A casual examination of the rod hints that it is a magic wand of some kind, and can be waved. If you do this in the East Bank of Fissure room it brings into existence a crystal bridge to West Bank of Fissure. Waving it some more will alternately destroy and re-create the bridge. It has no other use, and once the bridge is in place you can drop it.

Solving this gives you access to the Diamonds treasure and the Pirate's Maze for the Treasure Chest.

6.1.3 The bird, the cage, and the snake

These are some of the first items you will encounter after entering the cave in the normal way. The snake in the Hall of the Mountain King prevents you from going North, South, and West and often prevents you from going SW too; cutting off a lot of the cave system.

To solve this puzzle you will need to get the wicker cage, and catch the bird in it. Noting that you can't catch the bird while you are carrying the black rod; it is sufficient, however, to drop the rod in the bird chamber, catch the bird, and pick the rod up again. In the Hall of the Mountain King, open the cage to release the bird, and it will drive the snake away permanently. You don't need the bird or the cage after this.

Solving this puzzle gets you access to the Silver and Jewellery treasures, and many other locations.

6.1.4 The clam and the pearl

Unfinished.

6.1.5 The dragon and the oriental rug

Unfinished

6.1.6 The dwarf and the axe

Unfinished

6.1.7 The giant room and the eggs

Unfinished

6.1.8 The golden nugget

The golden nugget is so large that you can't carry it up the rough stone steps from the Hall of Mists (or down these steps from the Pit Top – you'll trip, break your neck, and the dream will end).

To get the nugget back to the building pick up the nugget, go `Up`, then use the magic word `Y2`, go `Down`, and use the magic word `Plugh`.

6.1.9 The lamp and the time limit

The famous brass lantern has a limited life, so don't leave it on when you are in lit rooms or outside. Additionally, the dream-time is limited by the time the player character sleeps for. Once the lamp starts to dim you've only 30 minutes left.

Despite its appearance, the depiction of it in various artworks, and presence of the oil filled pit, the lamp is *not* an oil lamp but is battery powered! Fresh batteries can be bought from the vending machine, at the cost of the coins (one of the treasures), so you'll need to complete the adventure on the original batteries for the best score.

6.1.10 The magazines and Witt's End

The issues of Spelunker Today in the Anteroom have to be left in Witt's End to score the infamous last point; just go `east` to get there. Getting out is less straightforward as it's another twisty passage situation. However this is *not* a maze, just a single room where the random exits mostly take you back where you were, except west - that never takes you out. Just keep going in any direction *except west* to reach the anteroom, … eventually.

6.1.11 The magic words

There are several magic words in the game which can only be used in certain locations and teleport you about. While the intention is that you discover these as you go they are active from the start.

6.1.11.1 Fie, Fie, Foe etc

See the Giant Room and the Eggs for details.

6.1.11.2 Plover

Teleport between Y2 and the Plover room and back.

6.1.11.3 Plugh

Teleport between the Building and the Y2 room and back.

6.1.11.4 Xyzzy

Teleport between the Building and the Debris Room and back.

6.1.11.5 Y2

Teleport between the Hall of Mists and the Rock Jumble (East of Y2).

> Dev Note: It's arguable whether this is a fast-travel link, or a magic word in the original. I've treated it as the latter as it doesn't do what you'd expect – but it could be an off-by-one error in the original game.

6.1.12 The maze of twisty little passages, all alike

This is the Pirate's Maze. If you haven't encountered the pirate yet, then his treasure chest (and anything he's stolen) won't be here yet so there's no reason for you to be in here. If you entered from the Hall of Mists immediately go `up` to return to the Hall of Mists. If you entered from the stalactite, go `west`, then `up` to get back to the Hall of Mists.

It is possible to map this maze by the classic strategy of dropping different objects in each room to make them unique. Note that going back the way you came won't necessarily take you back where you came from (the passages *are* twisty), and some exits silently take you back to the *same* room.

See "The Pirate and the treasure chest" when you need to navigate this maze.

6.1.13 The Ming vase and the cushion

Unfinished

6.1.14 The pirate and the treasure chest

Unfinished

When it is time to retrieve the treasure, head to the Hall of Mists. Go `Down`, `East`, `South`, then `North`. Pick up the treasure chest and any other items. Then go `SouthEast`, and `North` to the Brink of Pit, and `Down` to the Bird Chamber.

You may find the Hall of Mists to Treasure Chest path documented elsewhere as `Down`, `East`, `South`, `South`, `South`, `North`, `East`, `East`, `NorthWest`. This also works, but is longer. It may be that the shorter route is only in some versions of the game.

6.1.15 The plant in the pit

The little plant in the west pit of the two pit room needs to be watered if it is to grow big and strong. Bring some water in the little bottle (which you can fill in any location with water) and water the plant. You'll need to do this twice before the plant grows enough to become a beanstalk and for you to be able to climb up enough to reach the hole in the ceiling.

Solving this puzzle gives you access to the giant room with the golden eggs, and the rusty door.

6.1.16 The plover's egg and the pyramid

The conventional entrance to the Plover Room from the Alcove is so narrow that you can't fit through with any of the dream's items, including the lamp. While the eerie green light allows you to see in the Plover Room itself you can't see what is in the dark room beyond where there's another treasure. You can squeeze back through to the Alcove with the plover's egg (a plover is a kind of bird) sized Emerald treasure.

Solving this requires you to realise Plover is another magic word, and using it will teleport you back and forth between Y2 and the Plover Room, bypassing the narrow passage. From the Building, with the lamp, say `Plugh`, then `Plover`. Get the emerald, go `NorthEast`, get the Pyramid, go `SouthWest`, say `Plover`, say `Plugh`, and drop off your treasures.

6.1.17 The rusty door

The rusty door in the immense N/S passage has seized hinges. You'll need to fill the little bottle with oil from the east pit in the two pit room and pour it on the door.

Solving this puzzle gives you access to the trident treasure which you'll need for the clam puzzle.

6.1.18 The troll and the bridge

Unfinished

6.1.19 The twisty maze of passageways, all different

This is the Vending Machine maze. Go `Down` immediately to return to the Very Long Hall. This maze not only has twisty passages from every exit, but where they go has a random element. Its only saving grace is that the description of each room is subtly different so if you really want to map it you don't need to drop objects. There are twelve rooms in total.

The only reason to be in this maze is to use the vending machine to exchange the coins treasure for a fresh set of lamp batteries. Since this will lead to a less than perfect score, it is best just to stay out!

However if you want to reach the vending machine go `NorthEast`, `South`, and `South`. To escape the maze from the machine, go `North`, `Up`, `SouthWest`, and `Down`.

6.1.20 Where am I? (Debug Mode)

In the original game each room (and a few other things) were given a number in a data file. In the Yaffaif version these numbers are retained (but not used, other than for testing). To find the current room number you can use the console command:

```
/?$0.location.advNum
+  $0.location.advNum=1
```

This may be useful if you've obtained a map elsewhere with the numbered rooms on it.

6.2 Falling

The falling dream is a little bit of nonsense. It is always available to be selected. There’s no alternative but to plummet to your death.

Any objects you drop will fall to the ground below, but your impact will not leave you with a chance to retrieve them. As they are dream-things this won’t impact your regular game.

6.3 Home Invasion

This dream is a flashback to the night when your parents were killed. It is available as soon as the game starts, and is likely to be the first dream you have.

> Dev-note: The game generates your parents based on your own original character creation, so they are always similar to yourself.

There is no “right” way to complete this dream. In an ideal world your developing skills and equipment will allow you to defeat the civil guards, but at the start of the game you’d have to be very lucky to achieve this. Depending on how you proceed you will receive varying fatigue penalties:

- Avoiding the confrontation by never entering the main room of your house: 100
- Hiding in the cellar: 100
- Dying in the dream: 300
- Not saving one or both parents: 200
- Saving both parents: no penalty

Having this dream, whatever the outcome, will start the “Find Your Brother” main quest line the first time you have experience it. If one or more parents die then it also starts the “Revenge Parents” quest.

7 Debug Mode (aka Cheat Mode)

Early in the development I realised I was going to need a way to inspect and change the game model for testing and debugging. It's gradually become more feature rich and complex to the extent that it can just be used for messing around in the game.

It isn't practical to list *all* the things you can do in debug mode, as there are many thousands of things you can change.

7.1 What debug mode does

Debug mode makes four changes to the way the game works:

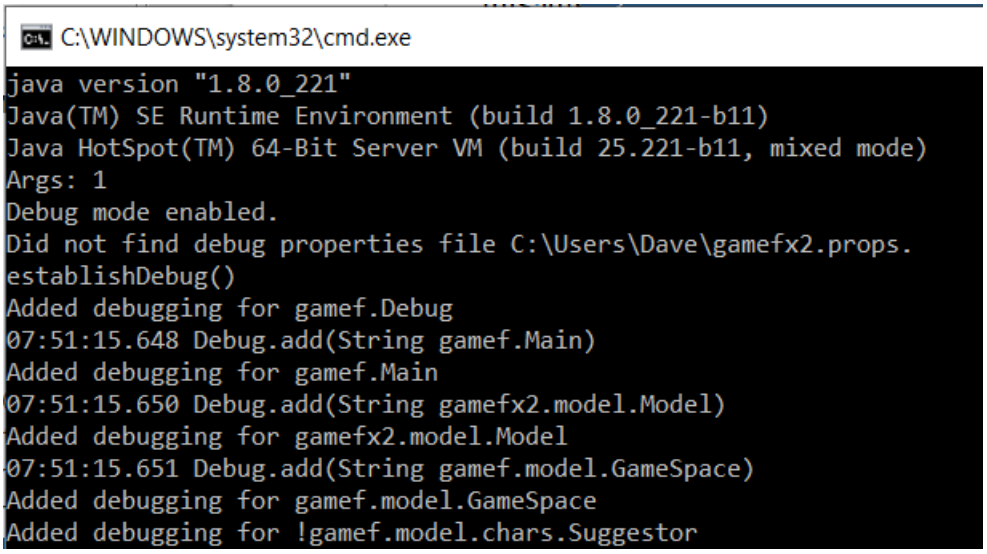
1. The game starts in a test room instead of your house. These test rooms have a varying collection of test items and characters in them, depending on what I am working on. There is a one-way portal, North, to the real game start. Any items you pick up in the test area you will take with you into the main game.
2. A debug button appears in the menu of the user interface. When pressed this launches a graphical explorer of the game model. You can also change settings via this interface. It also allows access to the dialogue tester.
3. The command text input recognises any text beginning with '/' or '~' as console commands. These have their own parser and syntax described below.
4. Debug log information is written to the terminal where Yaffaif was invoked. This gives detailed information about the operation of the code. If the game was launched from a file browser this is not visible.

7.2 How to configure debug mode

There are two ways to start the game in debug mode:

7.2.1 From the command line

The simplest way to launch in debug mode is to use the scripts `startDebug.bat` (Windows) or `startDebug.sh` from a shell prompt (macOS, Linux, Unix). Both of these pass the argument `--debug` to the game forcing it into debug mode. You should see something like this:



Here the game is starting in debug mode and setting up debugging for various classes from a built in list – basically what I've been working on recently. The line:

```
Did not find debug properties file C:\Users\Dave\gamefx2.props.
```

Is where it's looking for a debug property file, explained in the next section. Once the debug is set up you can see the game initialise:

```
07:51:16.606 Model.newGame()
07:51:16.706 GameSpace.loadXml()
07:51:17.070 GameSpace.setPlayer(IntelPerson:val1.home.main.player)
07:51:17.071 GameSpace.loadXml: player is IntelPerson:val1.home.main.player
07:51:17.072 GameSpace.verify(parent=root)
07:51:17.145 GameSpace.nextChapter(gamef.model.chap.Intro@3d88a0e6)
07:51:17.145 Intro.start()
07:51:17.149 Model.update()
07:51:17.149 Model.processMsgList()
07:51:17.207 Model.update: done
```

Each time you interact with the game more output is produced here. The text output to the story pane is also echoed here, should you need to cut and past anything.

7.2.2 From a property file

This method can be used to always start the game in debug mode, and is the way I typically use it as I am usually debugging!

When Yaffaif starts it looks for a property file in the user's home directory. If the file is present it reads it in. The location of the file is mentioned when running the game from a terminal (see above).

On **Linux** and **macOS** the file is called `.gamefx2.props` in your home directory and because of the initial '.' won't normally appear in directory listings. On **Windows** the file is called `gamefx2.props` without the initial '.'. Windows is less consistent about where a user's home directory is. On recent versions it is often where you'll also find "Downloads", "My Music", and "My Pictures". If you are not sure where the file should be put then launch the game from the `startDebug.bat` file as shown above and the game will indicate which file it is looking for.

Note: The file must have the extension **.props**, not **.props.txt**. Make sure you can see the file name extension in your file browser. On recent windows versions click on "View" and check the box for "File name extensions".

To enable debug create this file in the right location with a text editor and add the single line:

```
debug.enable=true
```

It will take effect next time the game is started. To turn debug off, delete the file or just change the `true` to `false`.

7.2.2.1 Controlling Debug Output

The property file can also be used to control which classes in the game produce output and how it is formatted. Here is a more complete example property file:

```
#
# Debug settings.
#
debug.enable=true
debug.package=false
debug.thread=false
debug.time=true
debug.classes=gamef.Debug,\
    gamef.model.GameSpace,\
    gamef.model.chap.Intro
```

Lines beginning with a '`#`' are comments.

If `debug.enable` is set `true` then debugging output is produced; otherwise it is hidden (except for exception and error reports).

If `debug.package` is set `true` then full class names are printed in the output instead of just the class name. For example `gamef.model.GameSpace` instead of just `GameSpace`.

If `debug.thread` is set `true` then the running thread is printed in the output before the class name. This is not terribly interesting in the context of the game as it is invariably `JavaFX Application Thread`.

If `debug.time` is set `true` then each line begins with the system time of day. For example `09:10:17.368`.

The property `debug.classes` is a comma-separated list of classes to output debug for. Each class must have its package and class name to be recognised. To spread the list over multiple lines, as shown above, each line except the last must end with '`,\`' with **no** characters after them.

If one of the class names is prefixed with '`!`' then debugging is not enabled for that class. This can be useful for temporarily turning off that classes debug without removing it from the list.

Debug does have a negative impact on the speed of the game; the more classes are enabled and more messages generated the slower it runs. For most actions this is not noticeable, but where a lot of time passes (working or sleeping) it is more apparent.

If one of the class names is "`all`" then debug is turned on for *all* classes. This is not recommended!

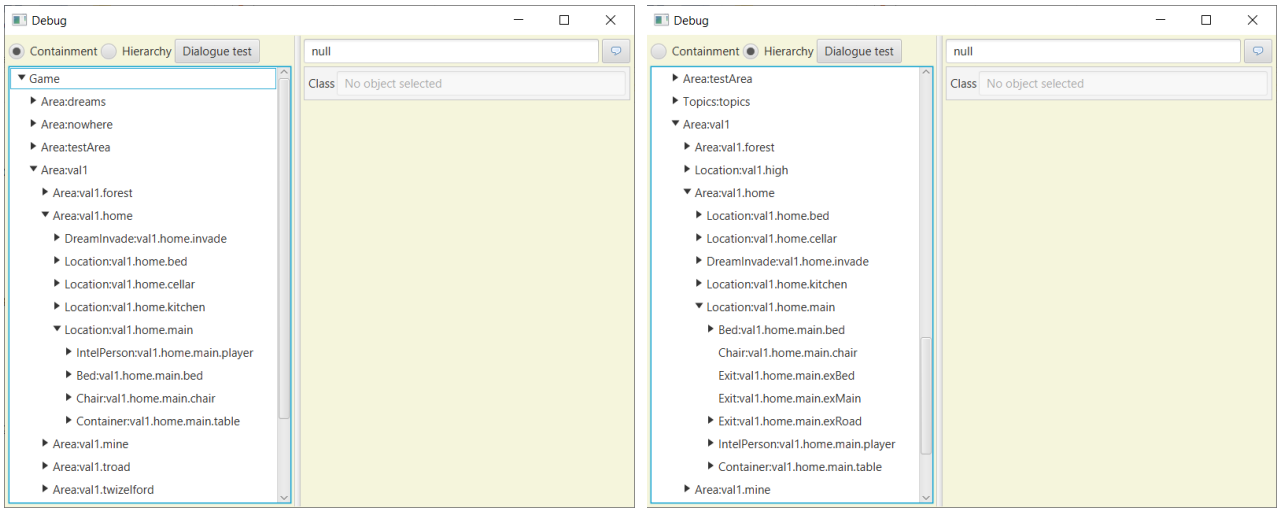
In the above example debug is turned on for:

Package and class	Function
`gamef.Debug`	Enabling debug on Debug will report any mistyped class names in the list. Keep this first in the list.
`gamef.model.GameSpace`	The main game model.
`gamef.model.chap.Intro`	The introductory chapter.

Not every class generates debug output. It is safe to enable it on ones that don't. Some generate a lot of output (like `gamef.text.util.TextReflFormatter`).

7.3 Debug Button

The debug button is in the menu area between the new game and load buttons. Pressing it pops up a new window over the top of the game window, just move it to one side. This is the Debug pane.

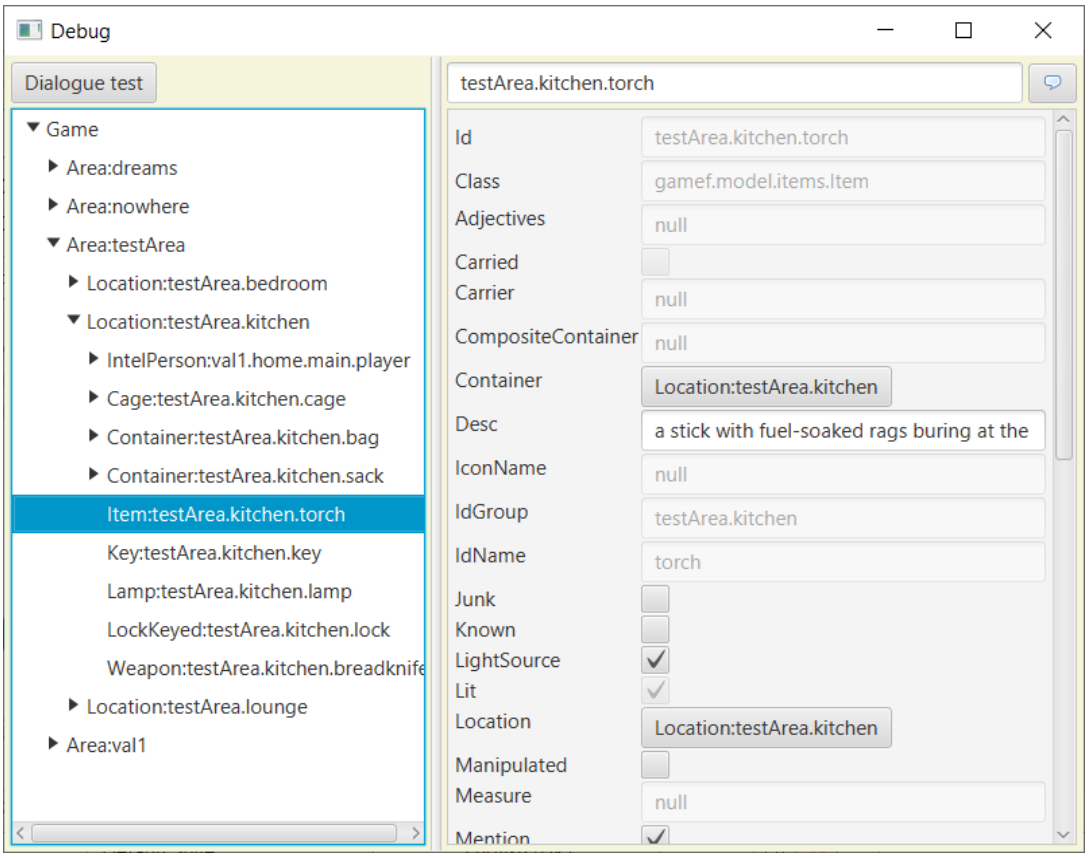


At the top left are two radio buttons that control the view of the game objects in the tree below. The `Containment` view (left) shows which objects are contained in other objects. As the game progresses the containment changes – if the PC picks something up from a location that object leaves the location and is transferred to the PC. It can show ID-less objects if they are contained by another object.

The `Hierarchy` view (right) is instead based on game object IDs. Once created an item's ID is constant. It does not show objects without and ID, but does show all objects with IDs. This includes those that are not part of the normal containment hierarchy such as the

flavours, gods, logger, and quests. It will also show prototypes until these get garbage collected.

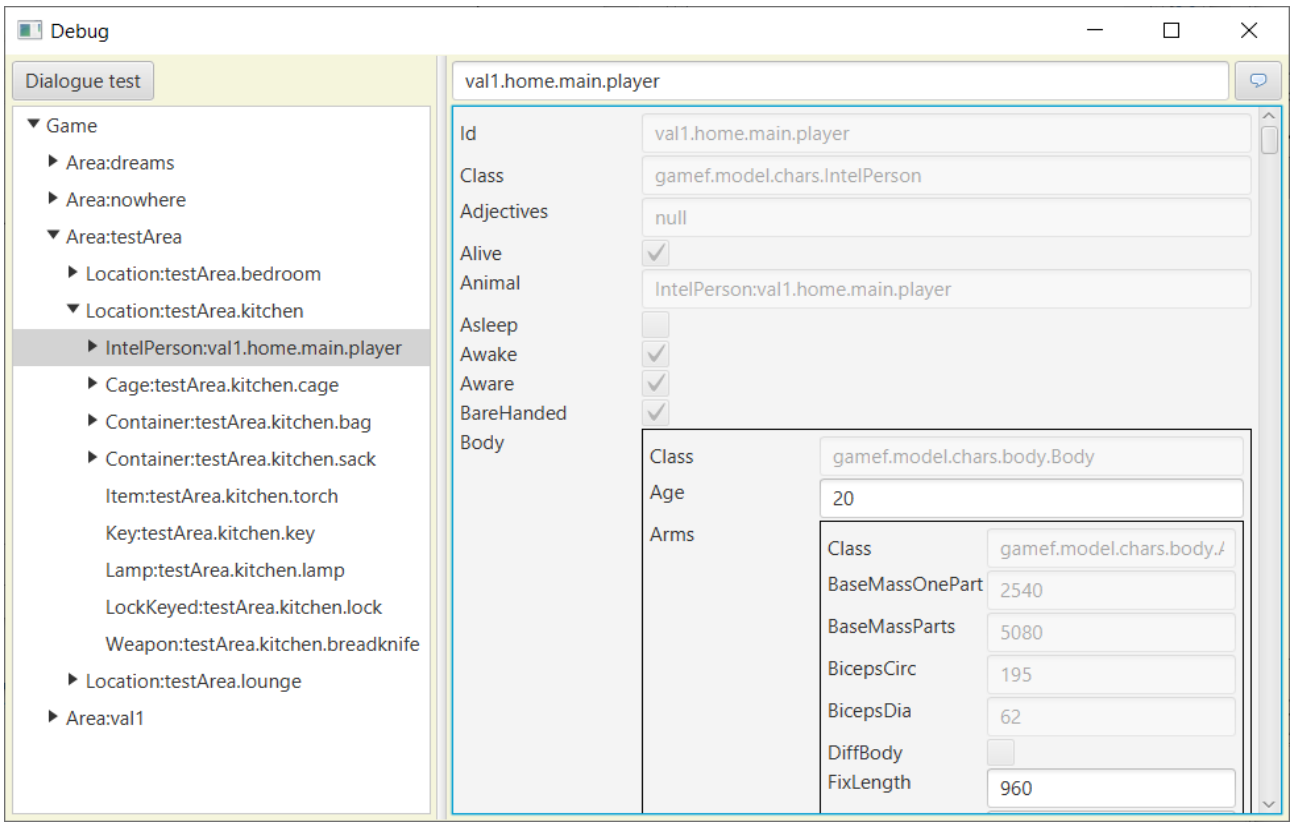
Choose the `Containment` view. On the top left hand side of the tree is a triangle next to the word `Game`. Clicking on the triangle shows the members of the Game node. Similarly, expand the `Area:testArea` and `Location:testArea.kitchen`. Expanding the location will show all the things in that location. Click on `Item:testArea.kitchen.torch`. The right hand side now show the various attributes of the torch:



Some of the attributes of the torch are read-only and greyed out like `Id` and `Class`. Others, like `Desc` can be altered.

As an experiment, enter `a cheap plastic model with a dead battery` in the `Desc` field and press `return`, then un-check the `LightSource` box. In the game window, try looking at the torch, its description should have changed. It also won't help much in the dark.

Try expanding `IntelPerson:val1.home.main.player`; the player character. This will show your inventory. Clicking on `IntelPerson:val1.home.main.player` will show your character's details in the right hand pane. At this point the window will need to be expanded, and like the main game window you can move the grab-bar between the two halves to suit. Characters have many nested objects shown by the rectangles.



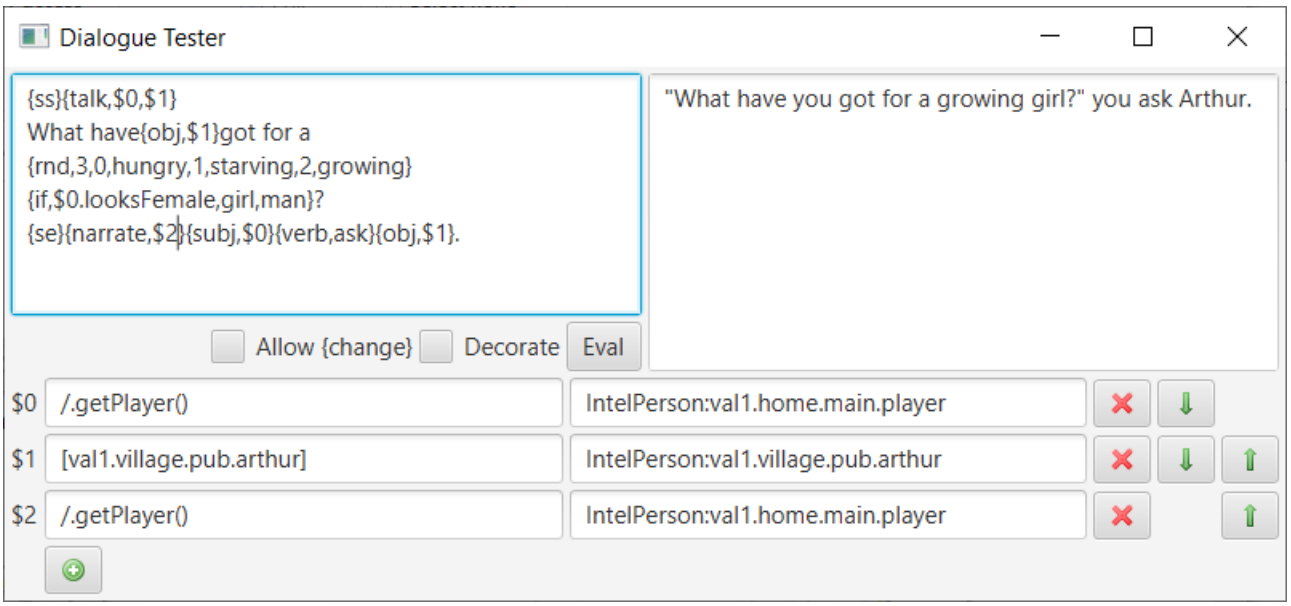
There are all sorts of things you could alter here. If you make changes they won't automatically change the tabs in the Stats area, you'll need to perform an action, such as looking at yourself first.

To understand more about what each attribute means it is helpful to have the game's Javadoc which is available with the Patreon "Experimenter" tier and above. Look up the `Class` in the Javadoc to find a description of each field.

Note: Running with the debug window open, particularly with a character selected, will slow down the game as all the user interface components are rebuilt.

7.4 Dialogue tester

The dialogue tester is intended for developing and experimenting with the text-substitution mechanism within the Yaffaif engine. It can be opened from the debug pane using the `Dialogue test` button, or by selecting an entry and using the button with the `speech bubble` icon, top right, which will add the entry to the dialogue arg-list.



This screen is divided into three main areas. On the top left is an area where the substitution text can be input, and some controls. The most important is the `Eval` button which causes the text to be parsed each time it is pressed.

On the top right is the output area. When the `Eval` button is pressed this will display the results of the substitution, or if there is an error, then the error message. In the above example there is a random element so the text will likely change each time the button is used.

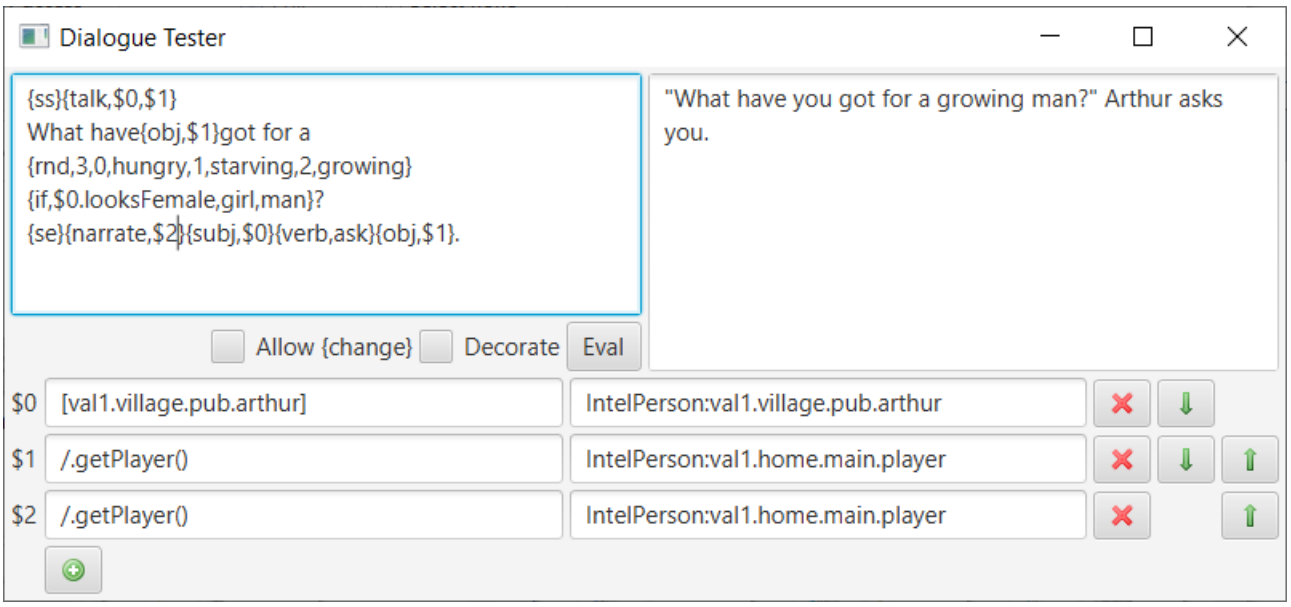
The bottom area defines the argument list that are used to replace `$0`, `$1`, `$2` and so on in the input text. On each line the left hand side text field is an expression; you can edit it, then press `return`. To the right is the result of evaluating that expression (or the error message if the expression is not valid). A new argument can be added with the `green plus` button, or from the debug pane with the speech bubble icon. An argument can be deleted with the `red cross` button.

When writing dialogue:

- `$0` is the person asking the question (player or NPC),
- `$1` is the person answering the question (player or NPC), and
- `$2` is always the player character for switching back to narration

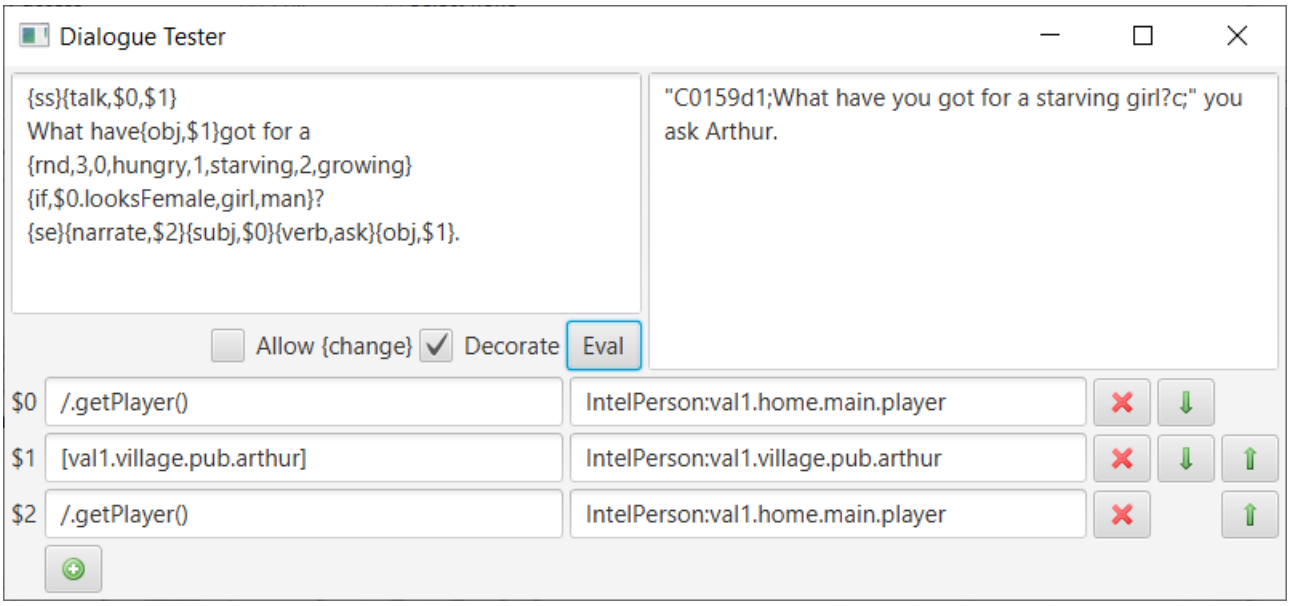
The details of the syntax of the text substitution are in the XML Format Document which is available with the Patreon “Experimenter” tier and above.

The `green arrows` can be used to re-order the arguments. In this example swapping the first two arguments makes the text work for Arthur talking to the player (without changing the text):



The `Allow {change}` check box is normally unchecked. In this state any `{change,…}` substitutions are syntax checked but not evaluated – evaluating them would change the game state. Checking this box allows `{change,…}` to be evaluated and change the model.

The `Decorate` check box controls whether control characters are added to show where bold and colour changes take place. Currently these control characters are shown as is, rather than being rendered in colour:



> Dev Note: A future version will use HTML markup for the colour changes rather than these terminal style escape sequences and then the output will then be able to show a rendering of the result.

7.4.1 Expression testing

The dialogue tester arg-list area can also be used for expression testing. Just enter the expression as an argument and press return. This can be simpler than working through the debug console.

7.5 Console commands

All console commands are entered in the command text box and start with '`/`' or '`~`'. The next character determines the operation:

`?`	query
`=`	assign
`>`	goto
`+`	creation
`a`	area diagnostics
`d`	dictionary
`g`	garbage/free memory
`h`	incident history
`i`	object ids
`t`	tokens
`v`	verbs
`w`	watch toggle

The query and assign operations need parameters that identify an object in the game, and then a getter/setter for that object:

- `$0` refers to the current player
- `/` refers to the `GameSpace`, which is the root of all items and areas
- A name in square brackets (such as `[val1.village.hospital.kerry]`) refers to an item/person by its `id`. Ids are typically formed from the initial area names "val1", "village", the location "hospital", and the object id "kerry" separated by '.'s.

Some detail of the structure of the game can be had by using the debug button. You can also open open the `GameF.jar` file in an archive utility (rather than running the game) and look at the `game.xml` file, and those it includes in the area directory.

A complete understanding of the various getters and setters really needs access to the Javadoc which is available with the Patreon “Experimenter” tier and above.

7.5.1 Get Values

Use the query `?` operator to display values. Getters can be chained by separating names with `.`'s.

How long is your hair?

```
/?$0.body.head.hair.lengthMm
+ $0.body.head.hair.lengthMm=30
```

How heavy are your boobs (total) in grams?

```
/?$0.body.bust.mass
+ $0.body.bust.mass=1515
```

How heavy are Kerry's boobs?

```
/?[val1.village.hospital.kerry].body.bust.mass
+ [val1.village.hospital.kerry].body.bust.mass=7579
```

7.5.2 Set Values

Use the assignment `=` operator to set values. The part after the last `.` is the variable to set. The new value is supplied as a parameter in brackets. The console performs auto type conversion as best it can to strings, integers, booleans, and enums. A detailed explanation of the syntax is provided in the Expression section below.

Note: Upper bounds to most numeric variables is `Integer.MAX` (2147483647)

Other objects can be referred to as parameters using the `/`, `$`, or `[]` syntax. Multiple parameters can be separated by `,`.

Internally these are converted to calls to getters and setters so `body.bust.mass(2000)` becomes `getBody().getBust().setMass(2000)`. If the setter does not exist then an attempt is made to find a method without the "set" prefix. So it is possible to call many other methods this way.

You can get into a lot of strange (potentially game-breaking) situations using the assignment operator!

Transform your eyes into feline ones:

```
/=$0.body.head.eyes.species(cat)
+ $0.body.head.eyes.species(cat) returned true
```

Start lactating:

```
/=$0.body.bust.lactating(true)
+ $0.body.bust.lactating(true) returned true
```

Glow-in-the-dark player who never needs torches:

```
/=$0.lightSource(true)
+ $0.lightSource(true) returned true
```

A pair of immobilising monster cocks (only use on a male/herm)

```
/=$0.body.genitals.male.erectLength(4000)
+ $0.body.genitals.male.erectLength(4000) returned true
/=$0.body.genitals.male.erectWidth(1100)
+ $0.body.genitals.male.erectWidth(1100) returned true
/=$0.body.genitals.male.number(2)
+ $0.body.genitals.male.number(2) returned true
```

Changing the player character to one of the NPCs: NOTE: Here be dragons; may break quests.

```
/=/player([val1.village.hospital.kerry])
+ /player([val1.village.hospital.kerry]) returned true
```

Or

```
/=/player([val1.village.pub.arthur])
/=/player([val1.mine.entCham.rochaine])
```

And back (best put the NPC where you found them first):

```
/=/player([val1.home.main.player])
```

7.5.3 Moving

Use the goto '>' operator to teleport. This bypasses all the checks normally performed with movement (combat, mobility, exits).

```
/>val1.mine.entrance

  You leave the main room.

  -- Mine entrance --
  You are standing in the entrance to the old mine workings. Ferns and
small plants grow on the damp rough hewn walls where daylight penetrates.
Within a short distance the vegetation gives way to dark stone. The
tunnel stretches on into the hillside. There is a doorway, SouthEast, to
the valley head. There is a tunnel to the NorthWest.
```

7.5.4 Creation

Use the add + operator to create items from factories The created item will be on the floor in the player's location.

> Dev-note: A “factory” is kind of class that has the job of creating new objects. Yaffaif uses factories for common objects in the game to avoid repeated declarations in the story files. However some objects are unique; they aren’t created from factories and because they aren’t included in factory classes cannot be created this way.

For example: creating a bacon butty:

```
/+food,baconButty
+ created Food:console1
```

The id of the new item is reported after the class name and the colon (here, `console1`), so you can modify it using the console commands for setting values.

Give it a new id (change "console1" as needed):

```
/=[console1].id(mybutty)
+ [console1].id(mybutty) returned true
```

And a silly number of calories:

```
/=[mybutty].kcals(1000000)
+ [mybutty].kcals(1000000) returned true
```

The following sections describe the (working) things that can be created with this console command. In each case you will need to enter `/+` and the left hand entry in the table. For example:

```
/+weapon,broadSword
```

7.5.4.1 Animals

The following animals can be created:

`animal,chicken`	a chicken
`animal,eel`	an eel, hostile by default
`animal,rat`	a rat

7.5.4.2 Clothing

The following items of clothing can be generated from the clothing factory. They are mostly description light, and these items can (and are) often customised by changing colours, materials, and descriptions. Clothing created this way will always fit you when you first wear it. Once you do it auto-fits to your current size.

`clothing,apron`	a generic apron
`clothing,blacksmithApron`	a blacksmith's apron (+1 AC) *
`clothing,bangle`	a generic bangle
`clothing,bra`	a generic bra
`clothing,boots`	a generic pair of brown leather boots
`clothing,camisole`	a generic silk camisole
`clothing,chainmail`	a set of chain mail (+2.5 AC)
`clothing,coat`	a generic coat
`clothing,croptop`	a generic crop top exposing the midriff
`clothing,flatCap`	a generic flat cap
`clothing,gambeson`	a generic gambeson (padded item worn under metal armour, or instead of armour +1 AC)
`clothing,gloves`	a pair of generic woollen gloves
`clothing,glovesLeather`	a pair of generic leather gloves
`clothing,hat`	a generic felt hat
`clothing,hatPointed`	a pointed felt hat
`clothing,helmetLeather`	a generic leather helmet (+0.5 AC)
`clothing,jacket`	a generic jacket
`clothing,nurseUniform`	a nurse's uniform (top half)

clothing,panties	a generic pair of panties
clothing,shirt	a generic white shirt
clothing,shoes	a generic pair of leather shoes
clothing,shorts	a generic pair of shorts
clothing,skirt	a generic shirt (mini)
clothing,socks	a generic pair of socks
clothing,stockings	a generic pair of stockings
clothing,stringVest	a generic string vest
clothing,tights	a generic pair of tights
clothing,trousers	a generic pair of trousers
clothing,trousersLeather	a generic pair of leather trousers
clothing,tshirt	a generic t-shirt
clothing,underpants	a generic pair of underpants
clothing,vest	a generic vest

NOTES: A blacksmith apron created this way won't work for the blacksmithing job – you need the specific one you can buy from Geoff.

7.5.4.3 Food & Drink

The following items can be created using the food factory:

food,baconButty	a bacon sandwich (300kcal)
food,beerBottle	a pint bottle of beer (182kcal, 4% abv)
food,beerGlass	a pint glass of beer (182kcal, 4% abv)
food,beerTankard	a pint leather tankard of beer (182kcal, 4% abv)
food,beerTankardKobold	a kobold-sized leather tankard of beer (60ml, 26kcal, 8% abv)
food,carrot	a raw carrot (100kcal) *
food,cheeseChunk	a chunk of cheddar (410kcal)
food,chickenRoast	a whole roast chicken (1,500kcal)
food,cucumber	a cucumber (100kcal) *
food,ham	a whole breaded ham (15,000kcal)
food,pieBeef	a beef pie (900kcal)
food,pieCase	a blind-baked pie case
food,pieChicken	a chicken pie (900kcal)
food,sandwichHam	a ham sandwich (200kcal)
food,starberry	a shot of starberry (50ml, 30kcal, 90% abv)
food,turkishDelight	a box of turkish delight (680kcal)

NOTES: The carrot and cucumber can also be used as impromptu sex toys.

7.5.4.4 Furniture

The following items of furniture can be created:

`furniture,armchair`	a generic armchair
`furniture,barstool`	a generic barstool
`furniture,bedDouble`	a generic double bed
`furniture,bedKing`	a generic king sized bed
`furniture,chair`	a generic chair
`furniture,chestofDrawers`	a generic chest of drawers
`furniture,cupboard`	a generic cupboard
`furniture,plinth`	a generic stone plinth
`furniture,singleBed`	a generic single bed
`furniture,stool`	a generic stool
`furniture,table`	a generic surface table
`furniture,torchSconce`	a generic sconce with a lit torch in it

7.5.4.5 Miscellaneous

Some miscellaneous items can be created:

`miscItem,bitsOfWood`	a generic pile of wood
`miscItem,menAtWork`	a sign indicating an unfinished part of the game
`miscItem,torch`	a lit torch

7.5.4.6 Potions

Potions primarily have temporary effects, but can have lasting ones too. The following potions can be created using the potion factory:

`potion,bellyGrow`	redistributes some body fat to the belly, causes more gain in the belly. When it wears off most (but not all) of the addition is redistributed
`potion,boobGrow`	redistributes some body fat to the breasts, causes more gain in the breasts. When it wears off most (but not all) of the addition is redistributed
`potion,bootyGrow`	redistributes some body fat to the butt, causes more gain in the butt. When it wears off most (but not all) of the addition is redistributed
`potion,bottomlessPit`	temporarily increases stomach capacity. Has a significant calorie content in it's own right (3,000kcal)

`potion,fatMelt`	Increases the speed at which the body consumes energy, resulting in gradual weight loss. The default potency increases energy use by a factor of 10 for a week.
`potion,healingPotion`	recover a fraction of the maximum hp, default is 30%. Single use, does not wear off.
`potion,hourGrow`	redistributes some body fat to the bust, butt, and thighs, causes more gain in those areas. When it wears off most (but not all) of the addition is redistributed
`potion,human`	testing item that sets species to fully human
`potion,koboldStrong`	a strong kobold transformation potion
`potion,lust`	a potion that temporarily increases lust. Can be wielded and thrown as a weapon
`potion,maleToFemale`	A potion that gradually converts a male to a female. Conversion is complete when the total potency of consumed potions is 1000.
`potion,sexChange`	testing item that reverses sexual gender
`potion,thighGrow`	redistributes some body fat to the thighs, causes more gain to the thighs. When it wears off most (but not all) of the addition is redistributed
`potion,weightGain`	a weight gain potion. Here the potency is the gain in kg.

Potions have a `potency` [1..1000], a `revert` [0..1000] which controls how much of the effect is removed when the potion wears of, and a `duration` in minutes (0 means the change is permanent). These can be changed after the item is created using the returned id:

Create a healing potion:

```
/+potion,healingPotion
+ created HealingPotion:console2
```

Make it maximum potency ("`console2`" is the id in this instance, change as needed in the step below):

```
/=[console2].potency(1000)
```

7.5.4.7 Weapons

The following weapons can be created:

`weapon,broadSword`	a generic broad sword
`weapon,custardPie`	a throwable custard pie that causes arousal, lust and Kobold transformation damage. Can also be eaten.
`weapon,dagger`	a generic dagger

`weapon,firebomb`	a throwable item much like a Molotov cocktail. Causes fire damage.
`weapon,firebrand`	a sword that causes fire damage in addition to normal sword physical damage
`weapon,knife`	a generic knife
`weapon,longSword`	a generic long sword
`weapon,sword`	a generic sword

7.5.5 Area diagnostics

The ‘a’ command lists all the areas in the game and what state they are in. This is used to debug where time is passing in the game as the engine only does simulation of NPC actions where needed. The states are:

State	Meaning
`Active`	The player is here and time is passing
`Child`	The area is a child of the Active area, and time is passing
`Parent`	The area is a parent of the Active area, and time is passing
`Standby`	Time is not passing
`Timeout`	The area is running down its timer, and time is still passing
`Wait`	The area still has active children, and time is still passing

When an area changes from Standby to one of the other states NPCs and other objects catch-up to the new time.

7.5.6 Dictionary

The ‘d’ command dumps the complete list of "words" the game parser recognises. Some words are actually short phrases, such as "ash blonde". The dictionary includes variants of words such as plurals of nouns, tenses of verbs, some of which are auto-generated. There may be occasional spelling errors in the list which the command is used to identify.

7.5.7 Garbage/Free Memory

The ‘g’ command runs the garbage collector and then reports how much Java memory the game is using and has free. This won't tally with the memory use reported by your operating system.

7.5.8 Incident History

The ‘h’ command dumps the saved incident history. This is a record of all the unexpected exceptions caught when executing player commands or NPCs performing their jobs. Each incident includes the player actions prior to the problem and the stack trace of the exception. These are recorded in the save game so they can be sent back to the developer and diagnosed.

7.5.9 Object IDs

The ‘`i`’ command lists all the object IDs present in the game and what kind of thing they are.

7.5.10 Tokens

The ‘`t`’ command lists the parser tokens for the words in the dictionary. Some words have multiple token matches, for example "orange" can be an adjective or a noun, and "pick" can be a verb or a noun.

7.5.11 Verbs

The ‘`v`’ command dumps the complete list of verbs the game parser recognises. The dictionary includes variants of verbs such as tenses, some of which are auto-generated. There may be occasional spelling errors in the list which the command is used to identify.

7.5.12 Watch toggle

The ‘`w`’ command takes a fully qualified class name as a parameter and toggles the debugging of instances of that class. Debug information is written to the console where the game was started as usual and will only be visible if run from a shell/command prompt. For example, to turn on debug for the lock up job:

```
/wgamef.model.chars.job.jobs.JobLockUp
```

If you want to debug on a specific object (rather than all objects of a given class) use the set value command to set the debug value to `true` or `false` on the object. For example:

```
/=$0.setDebug(true)
```

Will produce debug output for the player `IntelPerson` object.

7.6 Cheats

This section is a list of things I’ve used while testing, or people have asked for, which are basically cheats. Drop me a message if there’s something you’d like to see here.

7.6.1 Body

This section contains cheats for modifying the body. In general they can be applied to the player character by applying them to `$0`, or to an NPC by using the NPC’s Id such as `[val1.village.hospital.kerry]`.

7.6.1.1 Hair

Hair length can be set explicitly in in millimetres, so the following will set the player character's hair length to 1 metre (~3 feet) - approximately waist length depending on your character's height. Change the number as you wish:

```
/=$0.body.head.hair.setLengthMm(1000)
```

In the game hair grows at a slow, realistic rate, which can be changed. The command below will set the player character's hair to grow an inch a day (the value is in micro-

metres per day) though there's nothing to stop you using other values for even faster growth. So 240000 would be 1cm/hour or roughly an inch every 2.5 hours.

```
/=$0.body.head.hair.setGrowUmDay(25400)
```

Note: If your hair gets too long you will start tripping over it, and it will eventually immobilise you. There is currently no method in game to shorten your hair.

7.6.1.2 Lactation

There's some rudimentary lactation content in the game, but no current way to start lactating. If you wish you can start lactating with:

```
/=$0.body.bust.lactating(true)
```

While the game will give you some indication of how full the player is are a precise value with can be obtained with:

```
/?$0.body.bust.milkLevel
+  $0.body.bust.milkLevel=26.3%
```

There are a couple of settings that determine speed and volume. To set the player to fill up every two hours (could get annoying):

```
/=$0.body.bust.minsToFull(120)
```

There's also a setting that determines the volume of the breast that can be filled with milk. Currently this cannot be adjusted with console commands, but larger breasts produce proportionally more milk.

7.6.1.3 Proportions

When a character gains or loses weight it affects different parts of their body in different ways. The initial setup is controlled by the choices you make in character generation, specifically the body shape control.

The degree to which each part is affected can be adjusted (as the test items do). The following shows how to "allocate" more fat to the player's butt, but can be applied to other parts.

The first step is to find out what all the proportions are at the moment:

```
/?$0.body.partDeposit
```

which should result in something like:

```
+ $0.body.partDeposit = Partition[Torso=5142, Legs=2322, Arms=5,
Bust=858, Belly=90, Butt=238, Neck=212]
```

Most of their weight is going on their torso, and the next most significant target is their legs (this was from the std female shape). It's worth noting that the belly and butt values represent extra padding in those areas above and beyond the legs or torso, rather than the whole mass of the belly or butt.

Those total 8867, so when this character gains weight 238/8867 of the gain, or 2.7% goes to their butt. To make this a bigger proportion you'd need to change the butt value (if you want, you can also make it smaller by using a negative number):

```
/=$0.body.partDeposit.add(butt,500)
```

Check:

```
/?$0.body.partDeposit
+ $0.body.partDeposit = Partition[Torso=5142, Legs=2322, Arms=5,
Bust=858, Belly=90, Butt=738, Neck=212]
```

Now their butt will get 738/9367, or 7.9% of the new weight. You just need to eat to develop that booty!

Or, for a more cheaty way you can gain in grams like this for 10kg:

```
/=$0.body.addMass(10000)
```

7.6.2 Dreams

Dreams can be started without waiting to be tired enough to fall asleep, or for the random number generator to decide if you are going to dream, and which one:

The flashback to when your parents were alive:

```
/=/.dreams.start([val1.home.invade])
```

The falling dream:

```
/=/.dreams.start([dreams.falling])
```

The oddly familiar one (advent/colossal caves):

```
/=/.dreams.start([dreams.adv])
```

NOTE: Don't try to enter a dream using the "goto" `>` console command; the dream won't have been correctly initialised, and exceptions are likely to occur. Similarly, don't take the dreamer back into the main game as anything they pick up will be permanently lost when the dream ends. However, once a dream is running, it is safe to move around the *dream* with 'goto'.

7.6.3 Experience

To add experience points:

```
/=$0.stats.addExp(100)
+  $0.stats.addExp(100) returned false
```

This call returns `true` if you have advanced a level. To gain the rewarded hit points a second call is needed:

```
/=$0.stats.levelUpHp(/.glob)
+  $0.stats.levelUpHp(/.glob) returned null
```

To set you current experience level (does not change hit points):

```
/=$0.stats.level(2)
+  $0.stats.level(2) returned null
```

7.6.4 Hit Points

To change your current maximum hit points to an absolute value:

```
/=$0.stats.hpMax(30)
+  $0.stats.hpMax(30) returned null
```

7.6.5 Karma

The gods are represented by specialised topics in the game's global topic area. To interact with karma you need to know the ids of these topics:

God of:	Topic Id	Name
fertility	`[topics.godFertility]`	Yaridan
food	`[topics.godFood]`	Vumanom
travel	`[topics.godTravel]`	Vrithan

To determine your current karma with a specific god (or create a zero balance):

```
/=$0.karmaList.getCreate([topics.godFertility]).getKarma()
+ $0.karmaList.getCreate([topics.godFertility]).getKarma() returned 0
```

To add karma for a specific god (replace the topic with the relevant god) and discover your current balance:

```
/=$0.karmaList.add([topics.godFertility],100).getKarma()
+ $0.karmaList.add([topics.godFertility],100) returned 100
```

7.6.6 Location and mapping

Find the unique id of your current location (for example if you need to map a maze):

```
/?$0.location.id
+ $0.location=dreams.adv.alike1
```

And find where an exit goes (not for random exits though):

```
/=$0.location.getExit(east).to.id
+ $0.location.getExit(east).to.id returned dreams.adv.alike2
```

You can tell if an exit has a fixed destination with:

```
/=$0.location.getExit(east).isLogical
+ $0.location.getExit(east).isLogical returned true
```

7.6.7 Money

Need to be rolling in cash?

```
/=$0.purse.earn(10000)
+ $0.purse.earn(10000) returned null
```

7.6.8 NPC weight gain

Weight gain (and loss) is one of the basic transformations in the game. There's no direct way to cause NPCs to gain at the moment, but here are some approaches. You'll need to know the internal id of the target character, which you can find from the debug window. This bypasses the normal post-gain mechanisms, so you won't see any effects on clothing immediately.

7.6.8.1 Just adding mass

Add 1000g (1kg, 2.2lbs) of mass, in this case to Kerry:

```
/=[val1.village.hospital.kerry].body.addMass(1000)
```

Note: using `$0` instead of `[val1.village.hospital.kerry]` will affect the player character instead.

7.6.8.2 Add mass in a specific place

It is possible to specify where the gain takes place using the second form of `addMass`:, in this case adding 1000g (1kg, 2.2lbs) to her chest. This bypasses the normal post-gain mechanisms, so you won't see any effects on clothing immediately.

```
/=[val1.village.hospital.kerry].body.addMass(1000,bust)
```

Other body parts are: `torso`, `legs`, `arms`, `bust`, `belly`, `butt`, `head`, and `neck`.

Note: using `$0` instead of `[val1.village.hospital.kerry]` will affect the player character instead.

7.6.8.3 Taking control

Take control of the NPC character (then have them eat whatever you want):

```
/=/player([val1.village.hospital.kerry])
```

Changing back:

```
/=/player([val1.home.main.player])
```

7.6.8.4 Idling characters

Idling characters are ones that have jobs and sleep patterns defined, which you can see in the character roster for each area. When these characters aren't doing defined things they idle, which includes searching for food and eating. Several settings control what they do in this autonomous mode. This section shows changing them for Alexis in the village:

Changing their goal weight is the first step, and there are two ways: one relative to their current weight, and one absolute. As with all these examples you can adjust the numbers as you wish.

Increase goal weight by 20% (ie. 120% of what it is now):

```
/=[val1.village.gate.alexis].mind.setGoalWeightThou(1200)
```

OR:

Set an absolute target of 200kg (440lbs):

```
/=[val1.village.gate.alexis].mind.setMassTarget(200000)
```

This will take a long time to have an effect, as the default is a realistic gain, so you may want to also make the digestion more efficient by a factor of 10 (so an excess of 900kcals puts on 1kg):

```
/=[val1.village.gate.alexis].body.digestion.setEfficiency(10000)
```

It also helps to increase the stomach capacity by a factor of 10:

```
/=[val1.village.gate.alexis].body.digestion.setCapacityBoost(10000)
```

And as Alexis pays Arthur for her food she'll need more cash on hand:

```
/=[val1.village.gate.alexis].purse.earn(10000)
```

If you make these changes then go hang out in The Roses where Alexis goes for food then you should just be able to watch her stuff herself - arguably more fun than just adding to her mass though the earlier commands. Since Alexis will now be spending more cash with Arthur, he will have more cash to pay you for working too.

7.6.9 Quests

It is possible to start and complete quest parts if you know the id of the quest and the part. However, other than the experience reward, you won't get any of the items and so on for completing the steps and the quest. Quests just track what you've done in game and are intended mainly as a way to remind the player what they need to do. However some things depend on having completed certain quests.

Starting a quest:

```
/=[quest.findBrother].start()
+  [quest.findBrother].start() returned null
```

Succeeding quest part:

```
/=[quest.findBrother.start].success()
+  [quest.findBrother.start].success() returned null
```

7.6.10 Skills

Skills are identified by name. To add a skill to the PC both the name of the skill and the initial level [1..1000] are needed. Use:

```
/=$0.skills.addSkill(name, level)
```

To determine the level of a skill use:

```
/=$0.skills.getSkill(name).level
```

To improve a skill:

```
/=$0.skills.improveSkill(name,21)
```

7.6.10.1 Lockpicking

The lock manipulation skill is named `lockpicking`. To gain this skill at a low level use:

```
/=$0.skills.addSkill(lockpick,10)
```

As well as the level of the skill, this tracks the techniques, types of lock mechanisms, and lock models you know.

To gain knowledge of skeleton keys:

```
/=$0.skills.getSkill(lockpick).setSkeleton(true)
```

To know about warded locks:

```
/=$0.skills.getSkill(lockpick).setTypeKnown(warded)
```

To gain knowledge of the “Victor” lock model:

```
/=$0.skills.getSkill(lockpick).addLockModels("victor")
```

Alternatively, to know the type and model of a specific lock in the game:

```
/=$0.skills.getSkill(lockpick).setKnown([val1.troad.ocave1.cage.lock])
```

7.6.11 Transformations

Transformations proceed according to the contamination of a species to your body. To determine the current proportion of a species:

```
/=$0.trans.getProportion("bovine")
+  $0.trans.getProportion("bovine") returned 20.0%
```

Adding species transformations to a character is a two-step process. The first step can be performed multiple times to increase the percentage of a species (here on the player character, but you can replace $0 with an id):

```
/=$0.trans.add(cat)
+  $0.trans.add(cat) returned 2.0%
```

The second step is to queue up the transformation:

```
/=/.queueTf($0)
+  /.queueTf($0) returned null
```

The next time the player takes a turn the transformation will trigger.

> Dev-Note: not all transformations are in place yet, so you won’t get a complete transformation until you reach 100% of the new species.

7.7 Expressions

The following is an attempt to explain how to write expressions which can be used at the console, or in the definition files of the game. The parser (`gamef.util.ReflectUtilEval`), documented here works on the fly, parsing and evaluating as it goes.

7.7.1 Syntax

The table below describes the syntax:

Name	BNFish syntax	Examples
block	**{** *statements* **}**	`{ var player = $0; }`
statements	*statement* *statement statements*	

statement	*block*
	keyword-statement
	assignment
	expression
	;
		`{ }`
		`var x;`
		`x = 4;`
		`x.setName("fred");`
		`;`

assignment	*id* **=** *expression* **;**
	id **+=** *expression* **;**
	id **-=** *expression* **;**
	id ***=** *expression* **;**
	id **/=** *expression* **;**
	id **%=** *expression* **;**
	id **|=** *expression* **;**
	id **&=** *expression* **;**
		`x = 1 ;`
		`x += 2 ;`
		`x -= 3 ;`
		`x *= 4 ;`
		`x /= 5 ;`
		`x %= 6 ;`
		`x |= true ;`
		`x &= false ;`

keyword-statement	*break-statement* *continue-statement* *do-statement* *for-statement* *foreach-statement* *if-statement* *import-statement* *var-statement* *while-statement*	
break-statement	**break ;**	`break;`
continue-statement	**continue ;**	`continue;`
do-statement	**do** *statement* **while (***expression***);**	`do x=x+1; while (x<3);`

for-statement	**for (***statement expression* **;**
	assign-expr **)** *statement*
		`for (var x=0; x < 3;`
		`x=x+1)`
		`  y+=x;`

foreach-statment	**foreach (***id* **:** *term***)** *statement* **foreach (***id* **:** *term***,*** *term***)** *statement*	`foreach (x : 1,2,3,5)` `  y*=x;`

Name	BNFish syntax	Examples
if-statement	**`if (`** *expression* **`)`** *statement* **`if (`** *expression* **`)`** *statement* **`else`** *statement*	`if (x>3) x = 2;` `if (x<2) x=3; else x=5;`
import-statement	**`import`** *id* **`;`**	`import gamef.model.items.Item;`
var-statement	**`var`** *id* **`;`** **`var`** *id* **`=`** *expression* **`;`**	`var x ;` `var x = 4 ;`
while-statement	**`while (`** *expression* **`)`** *statement*	`while (x < 3 ) x = x + 1;`
assign-expr	*id* **`=`** *expression*	`x = 4`
expression	*or-expr*	`x`
or-expr	*and-expr* *and-expr* **`|`** *or-expr*	`x` `x | y | z`
and-expr	*eq-expr* *eq-expr* **`&`** *and expr*	`x` `x & y & z`

eq-expr	*rel-expr*
	rel-expr **`==`** *rel-expr*
	rel-expr **`!=`** *rel-expr*
		`x`
		`x == y`
		`x != y`

rel-expr	*add-expr*
	add-expr **`>`** *add-expr*
	add-expr **`>=`** *add-expr*
	add-expr **`<=`** *add-expr*
	add-expr **`<`** *add-expr*
		`x`
		`x > y`
		`x >= y`
		`x <= y`
		`x < y`

add-expr	*mul-expr*
	mul-expr **`+`** *add-expr*
	mul-expr **`-`** *add-expr*
		`x`
		`x + y + z`
		`x – y - z`

mul-expr	*unary-expr*
	unary-expr **`*`** *mul-expr*
	unary-expr **`/`** *mul-expr*
	unary-expr **`%`** *mul-expr*
		`x`
		`x * y * z`
		`x / y / z`
		`x % y % z`

unary-expr	*term*
	`!`*term*
	`-`*term*
		`x`
		`!x`
		`-x`

Name	BNFish syntax	Examples

term	*int-literal*
	str-literal
	enum-literal
	bool-literal
	null-literal
	class-literal
	this
	variable-identifier
	(*expression* **)**
	block
	function-call
	reflect-expr
	new-expr
	static-field
	static-method
	*term***.***method-invokes*
		`42`
		`'hello'` or `"hello"`
		`AWAKE` or `State.AWAKE`
		`true` or `false`
		`null`
		`Item.class`
		`this`
		`result`
		`( x )`
		`{ var player = $0; }`
		`isDead(x, y)`
		`$0`
		`new id(args)`
		`Integer.MAX_VALUE`
		`Math.max(1, 2)`
		`'hello'.length()`

new-expr	**new** *id arg-list*	`new gamef.model.items.Item( /)`
class-literal	*id* **.** **class**	`Item.class`
static-field	*id***.***identifier*	`Integer.MAX_VALUE`
static-method	*id***.***identifier arg-list*	`Math.max(1, 2)`
method-invokes	*method-invoke* *method-invoke***.***method-invokes*	`add($0)` `getList.add($0)`
method-invoke	*identifier arg-list*	`add($0)`
function-call	*identifier arg-list*	`isDead($0)`

arg-list	**()**
	(*arg***)**
	(*arg***,** *arg***,** **...** **)**
		`()`
		`(x)`
		`(x, y)`

arg	*expression*	`x`
reflect-expr	*reflect-root* *reflect-root* **.** *reflect-path*	`$0` `$0.body.isMale`

reflect-root	**$***int-literal*
	/
	[*id***]**
		`$0`
		`/`
		`[val1.home.chicken]`

reflect-path	*reflect-method* *reflect-method***.***reflect-path*	`body` `body.head.hair.lengthMm`
reflect-method	*identifier* *identifier arg-list*	`body` `rollIdx(10)`
id	*identifier* *identifier* **.** *id*	`val1` `val1.mine.entrance`
identifier	*[a-zA-Z][0-9a-zA-Z]**	`legLength2`
str-literal	**'** *chars* **'** **"** *chars* **"**	`'hello'` `"goodbye"`

Name	BNFish syntax	Examples
int-literal	*[0-9][0-9]**	`42`
enum-literal	*identifier*	`ASLEEP`
bool-literal	**`true`** **`false`**	`true` `false`
null-literal	**`null`**	`null`

Notes:

String literals can include and escape character '\' that prevents the next character matching the delimiter. To include a '\' write '\\'.

The syntax mostly follows Java except:

- The inclusion of reflection expressions that operate on the `GameSpace`
- There are no bitwise operators
- There is no character type.
- A single '&' or '|' represent `logical-and` and `logical-or`
- Enum literals don't need to specify their class.

Operators associate left to right and obey normal mathematical precedence, so 2 + 3 * 4 is 14, not 20.

Evaluation does not occur where it can be determined the rest of the expression has no impact on the result. Eg. `true | neverCalledFn()`.

The '+' operator can operate on mathematical expressions, or concatenate strings.

When using the characters &, <, > in XML they have to be replaced by their entities `&`, `<`, and `>`.

7.7.2 Type Conversion

The expression evaluation cannot type check until runtime, which is unfortunate. It also plays fairly fast and loose with conversions.

7.7.2.1 Conversion to Boolean

- A boolean is unchanged
- An integer converts to true if it is non-zero, false if it is zero
- An enum converts to true if it is non-null, false if it is null
- A string converts to true if it is "true" or "yes"; false otherwise
- Any other object converts to true if it non-null, false otherwise
  An expression like $0 & $0.isMale 'safely' tests the $0 is not null and is male.

7.7.2.2 Conversion to Enum

- A boolean cannot be converted
- An integer is converted to the enum with the ordinal of the integer
- An enum is unchanged
- A string is parsed for the enum name
- An object cannot be converted

7.7.2.3 Conversion to Integer

- A boolean true becomes 1, false becomes 0. Note: an expression of the form b1 + b2 + b3 > 2 tests if two or more of the booleans are true.
- An integer is unchanged
- An enum is converted to its ordinal value
- A string is parsed for an integer
- A Var, or VarConst is converted to it’s thou value
- An object becomes 1 if it is not null, 0 if it is null

7.7.2.4 Conversion to String

- A boolean is converted to “true” or “false”
- An integer is converted to a String in base 10
- An enum is converted to its name
- A string is unchanged
- A GameBase object is converted to its id
- An object has its toString method called
  This may not be desirable; most toString methods produce debug information

7.7.2.5 Conversion to Object

- A boolean is converted to a Boolean
- An integer is converted to an Integer object
- An enum is unchanged
- A string is unchanged
- An object is unchanged

7.7.2.6 Conversion to SpeciesRef

- SpeciesRefs are unchanged
- Other types are converted to String, then used to lookup a species ref

7.7.2.7 Conversion to Var, VarConst

- Var and VarConst values are unchanged.
- Other types are converted to integer, then this value is used as a thou value

7.7.3 Reflection

Reflection expressions refer to objects in the `GameSpace`.

The `GameSpace` itself is the root element, and is referred to by '/'. Any object in the game space can be referred to by its id in the form [id]. An array of parameters are provided to any evaluation, with the meaning determined by the caller. The objects in this array are referred to by '$' and the index, so $0, $1, and so on.

The reflection path is made up of a series of reflection methods. Each method is called in turn, and the result of one is the starting point for the next.

A method with no parameters is assumed to be a getter, so ".body" invokes the Java method "getBody()". If there is no getter with that name then the Java method body() will be invoked if it exists.

A method with parameters is invoked with the name given, so .setLength(3*60) will invoke the setLength(int) method with a parameter of 180. Type conversion is attempted to convert the parameter types to the available methods of that name. Where the method name matches various methods with different parameter types the selected method will be the one that has the same number of parameters and each one can be converted.

When used from the **console** or in a **change element** with the **`paramsExpr`** attribute there is an additional feature for backwards compatibility where a reflection method with parameters is also tried with the prefix "set", so "locked(false)" will invoke the Java method "setLocked(false)".

Where a method doesn't exist or the parameters can't be appropriately converted a fault occurs.

7.7.4 Functions

The expression parser has a number of built in functions:

7.7.4.1 has(container, object)

Tests to see if the container has the given object in it.

7.7.4.2 in(container, object)

Determines if the first parameter contains, recursively, the second. The first parameter must be an Area, Location, or a Container. The second must be an Item.

7.7.4.3 isDead(actor)

Determines if the first parameter is dead. The first parameter must be an Actor or something derived from Actor such as Animal, Person, IntelPerson. If the parameter is null it is assumed the actor is both dead and garbage collected and `true` is returned.

7.7.4.4 species(speciesEn, person)

Determines if a person is a specific species. The first parameter must be the name of a species. The second must be a Person.

8 And Finally ...

I hope you've enjoyed playing the game so far, that it has given you a flavour of where it is headed, and would like to continue supporting the development. At the same time, please understand you are under no obligation to do so at all. I hope you will join in the discussion around the game on Patreon, or which ever other forum you found this from.

If you've found something to not work as expected, or worse an error, please get in touch so I can fix them. Screenshots are good, console output even better. Despite the unit testing things do still slip through the net. Even if it's just a spelling mistake or a grammatical error I'd like to know.

Maybe this has inspired you to consider making your own game? I hope so as I enjoy playing games like this as much as I do creating them. In the future I'd like to make the engine that underpins Yaffaif available to others, but I don't think it is quite there yet. I need to finish more of it and tease the story apart from the engine, though I have made some progress in this direction. However there are plenty of others you can dip your toes in.

If you are really interested in modding the game in it's current state there's a Discord server:

https://discord.gg/uZRDWmpbPF

Thank you,

David Cooke / Dingotush