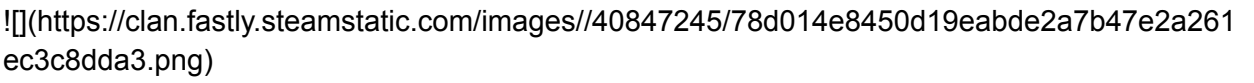
You can set your language under game options. This will change all text, including the user interface, tooltips, dialogue, everything.



Please note that these translations are in **alpha** and should be considered an early build. We ask that you please report any typos, grammar issues, or odd phrasing you encounter. Some minor UI issues, such as text going out of bounds, still exist and will be fixed over time.

Thank you for your patience and support.

(To those waiting on a new content update, this update is focused only on the new translations. Another update is coming soon)

Save files from v0.5.17 are compatible, but it's always a good idea to back up your saves just in case.

Translation Process

For those who are interested, this section outlines our translation process. We want to be as transparent as possible about our use of Large Language Models (LLMs).

This is a detailed and somewhat technical explanation, so feel free to skip if you prefer.

When it came to translating The Doors of Trithius, we faced several significant challenges:

- The amount of text: around ~103,000 words to translate.

- Unique vocabulary: hundreds of unique terms related to the game's lore or specific gameplay mechanics.

- Maintenance over time: since the game is under active development, we needed process to ensure translations are kept up to date every update as new content is added.

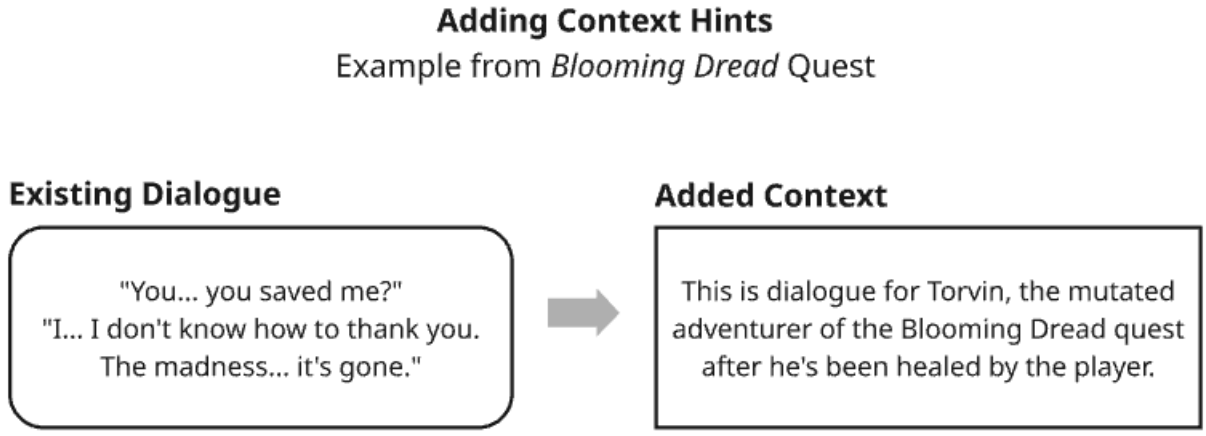
For our approach, we created a comprehensive glossary of unique terms and lore, used automation (Gemini Pro 2.5), and developed highly specific, context-rich prompts.

For example, when translating a quest, we included not only the dialogue but also quest details, character descriptions, and other related text to ensure tonal consistency. We did not want to fall into the trap of over-reliance on Gemini, which for all its impressiveness, like all LLMs, can produce inconsistent results unless given specific surrounding context about the text it is translating.

Context Clues

Our first step was to add context clues directly into the games data files. We added these wherever we thought they would be useful. This was especially important for character dialogue, which is often unclear when taken out of context.

These clues are not visible to players. Instead, they are collected by our later extraction script to be included in the prompts we send to the LLM.



Initial Glossary

To ensure consistent usage of the game's large and unique vocabulary, our next step was to build a comprehensive glossary.

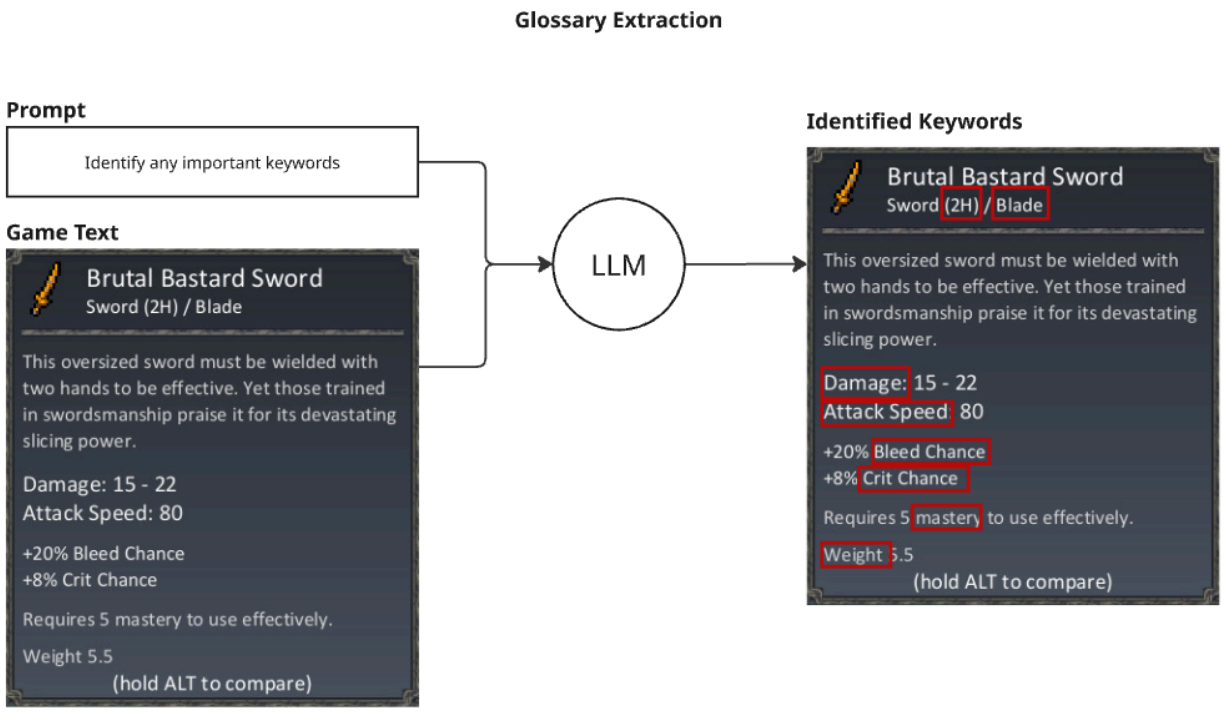
We started by manually defining terms based on our lore documents and knowledge of the game:

```
<gloss>
  <rootTerm>Lamassu</rootTerm>
  <rootDescription>
    Proper name of a behemoth creature, a monster, created by Machinists.
    Inspired by Assyrian mythology.
  </rootDescription>
</gloss>
<gloss>
  <rootTerm>Leafborn</rootTerm>
  <rootDescription>
    Proper name of a human faction.

    The Leafborn are a faction of forest druids who worship and live in harmony with the realm of
    Enalia, which they refer to as the 'Great Mother'. They act as guardians of the woodlands, seeking
    to preserve the balance of the forest. Their connection to Enalia allows them to grant others access
    to Nature Magic and to use it themselves for healing, binding, and beckoning the creatures of the
    wild.
  </rootDescription>
</gloss>
```

Glossary - Extraction

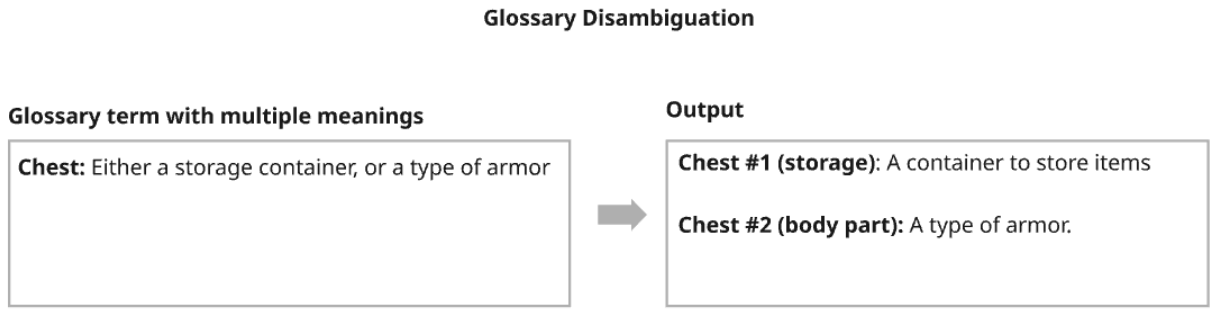
Next, we used Gemini Pro 2.5 to scan all the 103k words of the game's text for additional unique terms that we may have missed in our manual pass. This was necessary for identifying abbreviations or jargon, for example "2H" (meaning two-handed), that will require consistent translation everywhere they appear.



Glossary - Disambiguation

The next challenge was disambiguation. A single English word can have multiple meanings that requires different words in other languages. For example, "Chest" can refer to a storage container or a piece of body armor.

For any of these words we were able to identify, we separated them explicitly in the glossary.



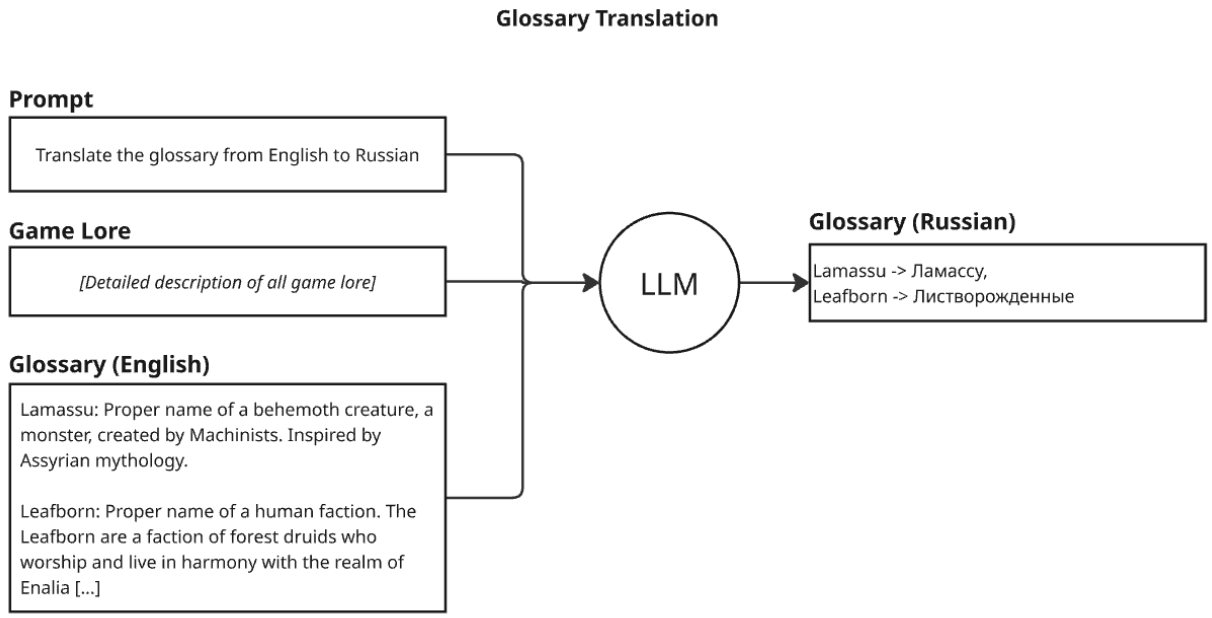
We then went through the game's text and replaced the original generic word (like "Chest") with the new, disambiguated term (like "Chest_::_storage") so that during the later translation process the LLM would know exactly which meaning is intended in each context.

```
<mapFeature>
    <key>gremlin_chest_bones</key>
    <name>Chest</name>
    <!-- More data here -->
</mapFeature>
```

```
<mapFeature>
    <key>gremlin_chest_bones</key>
    <name>Chest_::_storage</name>
    <!-- More data here -->
</mapFeature>
```

Glossary - Translation

With the glossary finalized, the final step was to translate every term into our target languages. To ensure maximum accuracy, we provided the LLM with deep meta-context, including detailed descriptions of the game's lore.



We started with Russian, as our programmer, Myrix, is a fluent speaker and could directly review the output. This allowed us to judge the translation quality and iteratively tweak our process for better results.

Text Extraction

The next step was to create a script to extract all the text from the game files (there are mountains of text!). The script pulled everything: combat log messages like "X creature dealt Y damage", UI strings, quest updates, lore books, item descriptions, stat names, and more.

The extraction script also gathered metadata alongside every piece of text. This provided context for the translation, such as knowing whether a word was part of a quest, an item description, or a UI element. The resulting file was 242,000 lines long with a file sizeof 11.4 MB.

```
</localizationValue>
<localizationValue>
  <key>Save Game</key>
  <description>Text on the button clicked by player to save the game from the main menu</description>
  <rootValue>Save Game</rootValue>
  <localValue>Save Game</localValue>
  <confirmed>true</confirmed>
  <usages>
    <usage>
      <scope>gui</scope>
      <javaPackageName>dot.gui.gamemenu</javaPackageName>
      <javaScopeName>GameMenuPane</javaScopeName>
      <javaMethodName>addButtons</javaMethodName>
      <codeStatement>var saveGameButton = new ComplexButton(xAt, yAt, L("Save Game"), saveGameService)</statement>
    </usage>
  </usages>
</localizationValue>
<localizationValue>
```

With this step complete, we had two key pieces ready: the finalized glossary and the complete list of text to be translated.

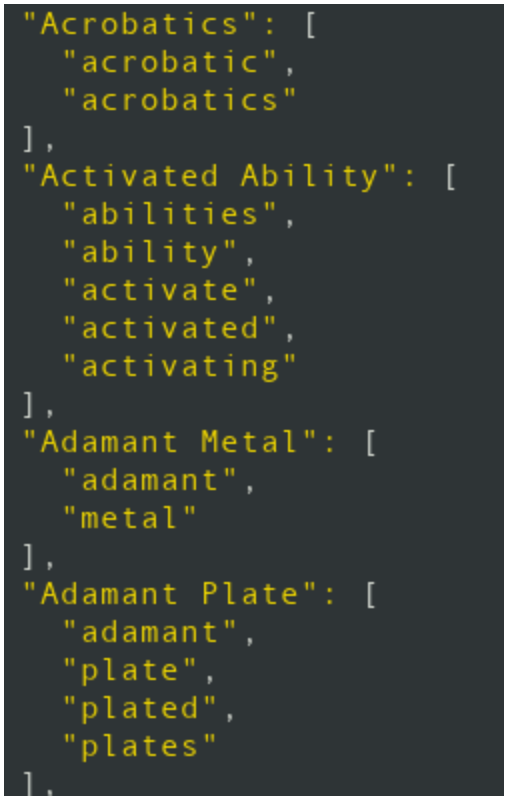
Lemmatization

This left one more technical hurdle: how do to reliably match the words in the game's text against the terms in the glossary.

A word in the text isn't always an exact match for its glossary entry. For example, a phrase might contain the word "activated" but the relevant term in our glossary could be the root word, "activate".

To solve this, we first had to lemmatize the text. This process analyzes each line and generates a list of all possible forms and root words. This will allow us to accurately match all relevant words in the text to their corresponding glossary entries.

See below for an example snippet from the output:



```
"Acrobatics": [
  "acrobatic",
  "acrobatics"
],
"Activated Ability": [
  "abilities",
  "ability",
  "activate",
  "activated",
  "activating"
],
"Adamant Metal": [
  "adamant",
  "metal"
],
"Adamant Plate": [
  "adamant",
  "plate",
  "plated",
  "plates"
],
```

Translation

Finally, we translated all text into our target languages. This was done in multiple prompts, breaking our text into small word chunks.

The biggest risk in this process is inconsistency. For example imagine a character's tone and personality changing from one line of dialogue to the next, or referencing the same lore term using different words. To prevent this, every prompt was packed with information. For example, a single dialogue tree was translated with the benefit of the entire quest script, character descriptions, relevant glossary terms, and world lore.

Our testing process for the final text mirrored the one we used for the glossary. We started with Russian so Myrix could review the translations in-game. After refining the full pipeline, we moved to Spanish with the help of community testers (shoutout to "Beep Beep Im A Jeep"), and then expanded to the other languages once we were confident in the results.

On Using LLMs

As a small team on a limited budget, hiring professional translators was not an option for us. Especially for a game that's constantly being updated with new quests and lore. Despite these

constraints we still wanted to ensure the best quality possible, which is why this work has been ongoing since the start of this year.

We know there is some contention in the gaming community about using LLMs. However, in my opinion (Sineso), this is an ideal use case. These translations wouldn't exist otherwise, and this work is not related to the core creative work or vision for the game.

Like any tool, Large Language Models (LLMs) can be used lazily to produce slop, or used thoughtfully as one part of a wider human-guided process. In our case, we've used the best possible tools, as part of carefully planned multi-step process, with human oversight throughout.

Next Steps

Once we've received enough community feedback we'll remove the "alpha" tag from the options menu and officially list these languages on our Steam page. From what we've seen so far, the quality is solid, but since we don't speak many of these languages, your feedback is essential.

Finally, a special thanks to Myrix for all his incredible coding work on this system, and a huge thank you to everyone in the community who has already helped us review the translations. We look forward to future feedback, and to continue working with the community to improve the quality of these translations.

Bug Fixes

- Fixed "Static Lance" exception when ability is used while completely surrounded by blocked tiles.

- Fixed "Reinforced Quarterstaff" not two-handed, as everything about it implies.

- Fixed Gremlin Totem still resurrecting Gremlins, even after being destroyed.

- Upon donating to the governor of a settlement, player reputation is now increased with the settlement's faction (in addition to the prosperity increase).