
LPCG

Release 1.4.3

Soren I. Bjornstad

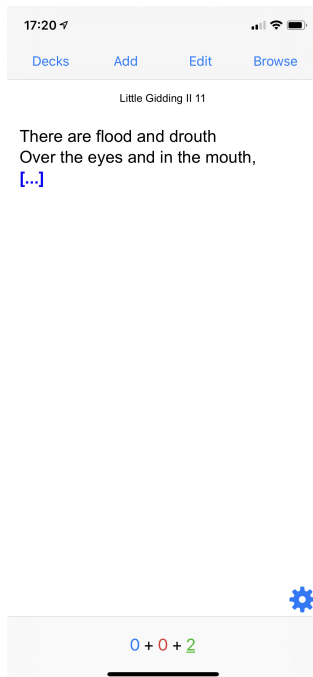
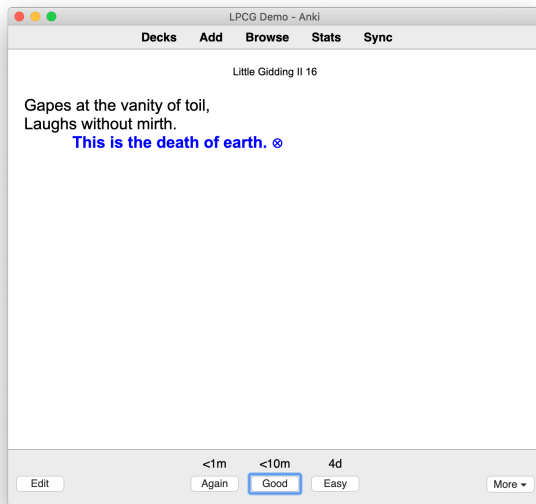
Jan 16, 2024

CONTENTS

1	Getting Help	3
1.1	Theory	3
1.1.1	Spaced-repetition strategies for poetry	3
1.1.2	Initial memorization	3
1.2	Installation	4
1.2.1	From AnkiWeb	4
1.2.2	From source	4
1.3	Importing Poems	5
1.3.1	Basic options	5
1.3.2	The poem editor	6
1.3.3	Generation settings	7
1.3.4	Editing LPCG notes	8
1.4	Technical Details and Customization	8
1.4.1	The LPCG note type	8
1.4.2	Why separate notes?	9
1.4.3	Customizing styling	9
1.5	Changelog	10
1.5.1	LPCG 1.4.3	10
1.5.2	LPCG 1.4.2	10
1.5.3	LPCG 1.4.1	10
1.5.4	LPCG 1.4.0	10
1.5.5	LPCG 1.3.0	10
1.5.6	LPCG 1.2.1	11
1.5.7	LPCG 1.2.0	11
1.5.8	LPCG 1.1.0	11
1.5.9	LPCG 1.0.0	11
1.5.10	LPCG 0.9.4	11
1.5.11	LPCG 0.9.3	11
1.5.12	LPCG 0.9.2	12
1.5.13	LPCG 0.9.1	12
1.5.14	LPCG 0.9.0	12

Anki **Lyrics/Poetry Cloze Generator** (LPCG) is an add-on for [Anki](#) for studying long passages of verbatim text, like poetry or song lyrics.

For clarity, throughout this documentation, we'll discuss “poetry,” but everything applies equally to songs, speeches, or any other text you might want to memorize.



GETTING HELP

If you have questions or technical issues which are not answered by this manual, you can email me at anki@sorenbjornstad.com.

If you have found a bug, would like to suggest a feature, or have improvements to contribute, please post on the issue tracker or submit a pull request at [GitHub](#).

1.1 Theory

1.1.1 Spaced-repetition strategies for poetry

Memorizing poetry can be a tough problem for Anki users: it isn't obvious how to divide it into discrete cards. People often try one of the following approaches:

- **Putting the whole poem onto one card.** This is a bad idea because if you have trouble with any part of the poem, you'll have to fail the entire card, thus you will review far more often than you need to.
- **Creating one card per line and defining an order for the cards,** so Anki asks for the lines in order. Aside from this being impossible in the current version of Anki, spaced-repetition scheduling doesn't work properly unless the cards are in random order.

LPCG takes a different approach: it generates cards that can stand on their own and make perfect sense in random order. Each card shows several lines from the poem and then asks you to recite some following lines. By default, it shows two lines of context and asks you to recite one line, and tests each line exactly once, but the details can be changed if needed.

In my experience, LPCG is not a good way to *memorize* poetry: not seeing the full context makes initial acquisition unnecessarily difficult. However, it is a great way to *review* poetry. This is fine, because as most Anki users know, the big problem isn't learning things in the first place, it's remembering them for long periods of time.

If you don't know a good way of initially learning poems, look at the *Initial memorization* section, below.

1.1.2 Initial memorization

The LPCG method works best if you memorize your poems before you start reviewing them in Anki. As alluded to in the *Spaced-repetition strategies for poetry* section, there are a lot of good ways to memorize poetry initially; it's retaining it that's hard.

If you're already good at memorizing poetry, don't listen to me, just use your own method; there's nothing magical about this one. But if you don't have a good method, here's one that works well for me and many others.

1. If you haven't already done so, read through the poem carefully so you have a general idea of how it goes.

2. Read through the poem again, line by line. This time, immediately after you read each line, look away from the page and repeat the line back. If you stumble or get it wrong, try again until you get it right: read the line again, then look away and repeat it again. It is helpful but not strictly necessary to speak out loud.
3. Repeat step 2, but take two lines at a time – then repeat with three, four, five, and six. You will find it gets a little bit harder each time, but not much, since you’re getting more familiar with the poem as well; before you know it you’ll be able to remember six lines at a time. It is unnecessary to continue this process beyond six lines.
4. Set the poem aside until the next day – it is truly remarkable how much easier it is to continue after a good night’s sleep. Do not skip this step! The rest will be easy if you do it and quite frustrating if you don’t.
5. Try reciting the poem straight through. If the text is easy, you may have it already; otherwise, go back and work on any spots you don’t have down yet. No particular method is necessary at this point. Don’t work for more than a few minutes; just touching all the difficult spots is enough.
6. The next day, repeat step 5. Most likely the poem will now come very naturally to you. Particularly difficult texts may require one more day.

Once you have the poem learned, put it into Anki if you want to remember it beyond the next couple of weeks – this is where LPCG comes into play.

It is totally fine to take breaks during step 3, even doing it over several days; you’ll spend 90% of your concentrated time in this step, and it’s not helpful to try to work when you’re tired.

1.2 Installation

1.2.1 From AnkiWeb

Most users will want to install LPCG through AnkiWeb. This is the easiest method, and it lets you get updates automatically.

You can find LPCG on [its AnkiWeb page](#), which will show instructions for completing the installation. LPCG, like Anki, is licensed under the [GNU AGPL3](#) license.

1.2.2 From source

If you want to do development on LPCG, you can install it from source. The source is available at [GitHub](#).

1. Clone the Git repository.
2. Create a virtual environment with `python -m venv venv`, and activate it (usually `. venv/bin/activate`).
3. Install Python dependencies with `pip install -r requirements.txt`. LPCG is tested on Python 3.10 but will work on 3.9 too.
4. Ensure you have PyQt5/6 and the `pyuic5 // pyuic6` command available (depending on which version(s) of Qt you want to build against; both 5 and 6 are supported at the time of this writing).
5. Run `make` to generate code for the dialog from Qt Designer (among other things).

To run the add-on within Anki, symlink or move the `src` directory into your Anki add-ons directory. Running `pytest` from the root directory will run the unit tests.

1.3 Importing Poems

To add a poem to your collection, choose *Tools* → *Import Lyrics/Poetry*. You will see the following window:

Deck: Poems

Title: Little Gidding II

Author (optional): T.S. Eliot

Tags (optional): gidding

Lines of Context: 2

Lines to Recite: 1

Lines in Groups of: 1

Poem:

The sacrifice that we denied.
 Water and fire shall rot
 The marred foundations we forgot,
 Of sanctuary and choir.
 This is the death of water and fire.

In the uncertain hour before the morning
 Near the ending of interminable night
 At the recurrent end of the unending
 After the dark dove with the flickering tongue
 Had passed below the horizon of his homing
 While the dead leaves still rattled on like tin
 Over the asphalt where no other sound was
 Between three districts whence the smoke arose
 I met one walking, loitering and hurried

Buttons: Help, Open file, Add notes, Cancel

Set the options and enter a poem as described below, then click *Add notes* to add notes for the new poem to your collection.

1.3.1 Basic options

The **Title** of your poem will appear at the top of all your cards so you know what you're reviewing. It's also used to warn you if you already have a poem by that title in your collection.

The **Author**, if provided, will also appear at the top of your cards, underneath the title.

The **Tags** and **Deck** will be attached to all notes generated from this poem and work the same way as they do elsewhere in Anki.

1.3.2 The poem editor

The large text box, called the *poem editor*, contains the text from which your cards will be generated. You can either type or paste a poem directly into the editor or open a plain text file somewhere on your computer using the **Open file** button.

The poem editor recognizes standard typographical conventions for poetry:

Stanza breaks

Leaving a blank line (that is, a line containing no characters, or only whitespace) starts a new stanza. Blank lines at the start or end of the poem are ignored, and consecutive blank lines are treated as a single blank line.

Stanza breaks don't appear as vertical space during reviews. Instead, the character ¶ appears at the end of a line that ends a stanza, and the character ¶ appears at the end of the poem's final line.

Indentation

Placing any number of spaces or tabs at the start of a line creates an indented line. Only one level of indentation is recognized, as poems that use more than one level of indentation are extremely rare, and this frees you from minutiae about how many spaces or tabs to use.

LPCG also applies a hanging indent at review time to any lines that do not fit on your device's screen in their entirety. This hanging indent is twice the size of a "hard" indent that's part of the poem.

Comments

Lines beginning with the comment character # are ignored, as is anything after the first # within a text line; everything else in the editor will be treated as the text of your poem.

If you keep your poems in text files, you might want to use this to include annotations or information about the title or author.

Here is an example of a correctly formatted poem:

```
# from "Little Gidding," movement II
# by T.S. Eliot

Ash on an old man's sleeve
Is all the ash the burnt roses leave.
Dust in the air suspended
Marks the place where a story ended.
Dust inbreathed was a house -
The wall, the wainscot and the mouse.
The death of hope and despair,
    This is the death of air.

There are flood and drouth
Over the eyes and in the mouth,
Dead water and dead sand
Contending for the upper hand.
The parched eviscerate soil
Gapes at the vanity of toil,
Laughs without mirth.
    This is the death of earth.
```

1.3.3 Generation settings

The three numeric spin-boxes allow you to customize how much text appears on each card. Tweaking these settings is entirely optional; the default values are sufficient for most poems. You can change the default values in the add-on config (*Tools* → *Add-ons* → *LPCG* → *Config*).

Lines of Context

The number of lines that will be shown on the question side of each card as a prompt. With the default of 2, you'll see 2 lines of the poem and then be asked to recite the following line(s).

At the very beginning of the poem, fewer lines of context will be available (e.g., there is no context at all for the first line save the title of the poem). If fewer lines of context are available than the value of this option, the text [Beginning] will appear on the question side of the card, followed by whatever context is available (if any).

Lines to Recite

The number of lines that will be revealed on the answer side of each card. With the default of 1, a single line will be shown after the context lines.

If *Lines to Recite* is more than 1, the cloze prompt [...] on the generated cards will show the number of lines that you need to recite, e.g., [...] 3] if there are three lines to recite.

If the number of lines in your poem is not evenly divisible by *Lines to Recite* (e.g., you set it to 3 in a 16-line poem), one or more cards at the end will have fewer lines to recite; the cloze prompt will be adjusted accordingly.

New in LPCG 1.3.

Lines in Groups of

Warning: This option can be confusing. If the two preceding options are enough to meet your needs, don't even bother reading this section!

If this option is greater than 1, the physical lines in the poem editor will be grouped into "virtual lines" which will then be treated in accordance with the *Lines of Context* and *Lines to Recite* options. This can be useful if your poem has a large number of extremely short lines.

For example, if you set *Lines in Groups of* to 2, *Lines of Context* to 2, and *Lines to Recite* to 1, you'll get cards that show 4 physical lines of context and ask you to recite 2 lines.

At first glance, this may appear to be exactly the same thing as doubling the values of *Lines of Context* and *Lines to Recite*. However, increasing those values merely increases the number of lines that appear on each card, keeping the number of cards and the amount of overlap the same, whereas grouping lines results in generating fewer cards that have less overlap. The best way to understand this is by example. Say we have the following uninspired poem:

A
B
C
D
E
F
G
H

With the lines in groups of two using the settings described above, we would get the following cards:

[Beginning] ==> A B
[Beginning] A B ==> C D

(continues on next page)

(continued from previous page)

```
A B C D ==> E F
...et cetera
```

If we instead were to set *Lines in Groups of* to 1, *Lines of Context* to 4, and *Lines to Recite* to 2, we would get:

```
[Beginning] ==> A B
[Beginning] A => B C
[Beginning] A B => C D
[Beginning] A B C => D E
A B C D => E F
B C D E => F G
...et cetera
```

New in LPCG 1.3.

1.3.4 Editing LPCG notes

Sooner or later you will probably find that you made a typo in one of your poems. To correct the typo completely, you must *search for the typo in the browser*, rather than just pressing edit, since the typo will be included on several generated notes.

1.4 Technical Details and Customization

Here's some technical and architectural mumbo-jumbo that you probably don't need to use LPCG but might be useful if you want to customize LPCG or are just curious.

1.4.1 The LPCG note type

The note type has five fields:

Line

Contains the line or lines this card asks you to recite. Since LPCG 1.3, each line is enclosed in a `<p>` tag. For backwards compatibility, the template is also required to correctly display this field if it contains a single line without any `<p>` tags.

Both the *Line* and *Context* fields look kind of oddly spaced in the editor due to the use of `<p>` tags, because the editor doesn't use the same styling as the review window and this can't easily be customized.

Context

Contains the several lines prior to *Line*, to give you an idea of what to recite.

Title

The title of the poem that this line comes from.

Sequence

The line number (counting only text lines, not comments or blank lines) of the first line of the *Line* field in the original poem.

If the *Lines in Groups of* [setting](#) was set to a value greater than 1, the actual line number in the original poem is the sequence number multiplied by *Lines in Groups of*, minus *Lines in Groups of*, plus 1.

Prompt

If this field is populated, it will appear instead of [...] on the question side.

New in LPCG 1.3.

It is safe to add additional fields to the note type if you wish to include more metadata on your poems. The only likely issue would be if LPCG added a new field in the future and its name happened to conflict with yours; even then, this wouldn't cause any data loss, you'd just have to rename your own field if you wanted to continue using it.

1.4.2 Why separate notes?

LPCG takes the approach of creating a series of unrelated notes, rather than a single note. Creating one note per poem would have a number of serious disadvantages:

- Sibling burying would get in the way – it can of course be turned off, but this would be confusing to some users.
- The amount of template boilerplate required would be absurd, and extremely large templates can cause syncing and other issues.
- LPCG would not be able to support poems of arbitrary length, since note types cannot contain an arbitrary number of fields or card types; at some point you would have to create multiple notes for the same poem, which would break the neat correspondence anyway.

The Sequence field helps to mitigate the disadvantages of having separate notes – by searching for a given poem (e.g., "note:LPCG 1.0" "Title:My Poem") and then sorting by the Sequence field, you can still see and select the whole poem at once.

One possible area for future improvement would be caching the poem or its parsed representation so that a poem's notes could be edited from a poem-editor-like text box after initial creation. At the moment, this feels like gold-plating since poems rarely need editing and I have not seen any demand for such a feature.

1.4.3 Customizing styling

If you'd like to customize the width of indents, you can make some changes in the styling section of the card types dialog:

- **Hanging indent** (when a line is too long for the screen): Change `margin-left` and `text-indent` in the `.lines` class. They should be inverses of each other (e.g., if `text-indent` is 30 pixels, `margin-left` should be -30 pixels). You'll also need to change `margin-left` in the `.cloze` section to match `text-indent`.
- **Manual indent** (when an indent is given in the original poem text): Change `margin-left` in the `.indent` class.

The end-of-stanza and end-of-poem markers can be changed in the add-on config (choose *Tools* → *Add-ons*, select LPCG in the list on the left, and click the *Config* button).

As of version 1.3, LPCG inverts the color of cloze deletions in night mode, as the default solid blue is quite difficult to read in night mode on many screens. If you don't like the color, you can change it in the styling rule `.nightMode .cloze`.

1.5 Changelog

1.5.1 LPCG 1.4.3

Released on December 6, 2023.

- Fix “wrapped C/C++ object of type QPushButton has been deleted” error occurring when generating cards with LPCG more than once per Anki session, or opening the add or edit screen after using LPCG.

1.5.2 LPCG 1.4.2

- Add support for Qt 6, in advance of Anki moving to that GUI toolkit version. Builds of Anki that use Qt 5 will continue to work.

1.5.3 LPCG 1.4.1

Released on July 24, 2021.

- Update all deprecated function and type names to current versions as of Anki 2.1.45.
- Drop support for earlier Anki versions.
- Accept Ctrl+Enter for *Add Notes* in the poem editor, just like in Anki’s standard add-notes dialog.

1.5.4 LPCG 1.4.0

Released on November 15, 2020.

- Add an *Author* field to the note type, which can optionally be set when adding poems (thanks to @cashweaver). The author displays just underneath the title and index when reviewing, if present.
- The Add Notes access key has been changed from a to d to resolve a conflict with *Author* (all the letters in *Author* were already taken!).
- Allow quotation marks to be used in the title of a poem, now that Anki is able to escape quotation marks in searches.
- Make the default values of the import spinboxes configurable.
- Add screenshots to the README.

1.5.5 LPCG 1.3.0

- Added options *Lines to Recite* and *Lines in Groups of*.
- Add a *Prompt* field to the note type. This was added to support showing the user how many lines they’re being asked to recite, but could be used for other things in the future. Backwards compatibility with existing notes is maintained by displaying [. . .] if the field is empty.
- The *Line* field now wraps each (or the only) line in <p> tags, to support multiple-line recitation with correct indentation. Existing LPCG notes without <p> tags will still display fine as well.
- Invert color of cloze deletions in night mode, as the default color is unnecessarily difficult to read on many screens.
- Added some automated regression testing.

- Add a *Help* button to the import dialog leading to the documentation.
- Under-the-hood refactorings and updates to newer features offered by Anki.
- Refresh the documentation and move to Read the Docs for a nice multi-page view instead of relying on GitHub's Markdown rendering.

1.5.6 LPCG 1.2.1

- Provide a useful error message when generating notes and the LPCG note type is missing a required field.

1.5.7 LPCG 1.2.0

- Support for Anki 2.1 (only).
- Fixed several minor bugs in splitting poems into stanzas that could result in end-of-stanza and end-of-poem markers appearing on lines by themselves.
- Fixed text flowing off the right side of mobile device screens (thanks to Jonta).

1.5.8 LPCG 1.1.0

- Added support for changing the number of lines of context you want to see for a given poem. The default remains 2.

This is the last version that supports Anki 2.0.

1.5.9 LPCG 1.0.0

LPCG was completely rewritten and became a proper Anki add-on instead of a stand-alone Python script. A new note type is also used with this version.

1.5.10 LPCG 0.9.4

- Improve the user experience when the temporary folder is not writable or the user specifies a nonexistent file.
- Add a message explaining that Anki should have opened to import notes, in case it doesn't and the user is left sitting there wondering what's supposed to happen next.

1.5.11 LPCG 0.9.3

- Fix typo in code that caused LPCG not to import successfully at all.

1.5.12 LPCG 0.9.2

- Fix off-by-one error sometimes resulting in incorrect card generation.
- Allow dragging and dropping files onto the terminal window to work correctly in more situations.

1.5.13 LPCG 0.9.1

- Tags should not be mandatory.

1.5.14 LPCG 0.9.0

- First public release.