Readme:

Change Log: 0.07

- Added **strings** gem, used to align, truncate and wrap strings. Needed due to the length of some strings that come with a language learning application.
-

Change Log: 0.06

- Added rudimentary flash card functionality and necessary methods to pull vocab lists from lessons.
    - User interface needs refinement
    - Need to add session stats to keep track of user progress.

Change Log: 0.05

- Fixed flaws with lesson json files, inconsistent formatting.
- Have yet to update style guide for lesson content.
- On load, app will automatically find any json file and parse it (assuming it is a valid lesson.json). Error handling NOT yet added.
    - Cache metadata from each lesson, core content will be loaded upon request to save system memory.
- Added Study Menu and **Flashcard menu**. Flashcard menu is entirely placeholder, **Study** menu list all lessons found.
    - Selecting a lesson will display additional information about the course and its subject matter.
    - Does NOT yet lead to a way to study, that's next.

Dev Notes

- Added Curriculum singleton which handles lesson metadata.
- Added Lesson class which initializes with lesson metadata.
    - Have yet to add a method to load course material.
- Need to add a lot of error handling and figure out how to effectively test the code soon, but first goal is to get a working product out.

Change Log: 0.04

- Added more information to lesson.json format, in the form of module information. The rest is the same.

```
[
  {
    "Module": "It's / c'est" ,
    "Difficulty": "Beginner" ,
    "Author": "Sam O'Donnell" ,
    "Last Updated": "26/09/2021"
  },
  [  #Lesson block
      { Lesson Info:
```

~~~
        {
          Title: This is a title,
          Description: This is a description
        }
  ...
        ```




### Change Log: 0.04
* Puzzled out and implemented a content system that should work well and allow for
reasonably dynamic content that is also structured enough to not leave learner's
lost or out of their depth.  In early prototype stages but hopeful.  Content
structure described below.
* Removed outdated lessons to be replaced with the new structure.

**Todo**
* Add a system to cache vocab the user has covered.  Have yet to lock in how I
want this data to be structured and organised.
* Create a system to parse strings and replace relevent sections (eg [:noun :c])
with words in the user's study material.
* Create a system to parse json lessons and store relevant information in system
memory.
  * Probably two separate files
    * User profile info including all relevant user info and vocab (no further
information on the words)
    * dict.json, json with hash of each entry, key always matches the word
  * to find appropriate lists, would need to parse every word, map matching data
to system memory.
  * New vocab would be << to dict.json and appended to system memory cache
(instead of appending to dict then re-parsing and re-initializing them into system
memory)


**Content Structure**

To facilitate dynamic content words have been ascribed necessary information.
Type: :type :sub-type,

  Legend:

      :un           parse and replace with 'un/une' according to gender of subject

      :noun :c      common noun
      :noun :p      noun proper
      :noun :con    noun concrete
      :noun :a      noun abstract
      :noun :col    noun collective

      :adj :comp      adjective comparative
      :adj :super     adjective superlative
      :adj :pred      adjective predicate
      :adj :compound  adjective compound
~~~

```
      :adj :pos       adjective possessive
      :adj :dem       adjective demonstrative
      :adj :proper    adjective proper
      :adj :part      adjective participial
      :adj :lim       adjective limiting
      :adj :desc      adjective descriptive
      :adj :i         adjective interrogative
      :adj :adj       adjective attributive
      :adj :dist      adjective distributive

More will be added in the future, documentation will be updated accordingly.
```

[[#Lesson block { Lesson Info: { Title: This is a title, Description: This is a description } }, [# Vocab Block {"good": [{ "type": ":adj :desc" }, { "translation": ["bon", "bonne"] }, { "gender": "masculine" }] }], [# sentence structure to practice this lesson { "It is not [:adj], but it is [:adj]": "Ce n'est pas [:adj] mais c'est [:adj]" }, { "It is not [:adj], but it is a [:noun :c]": "Ce n'est pas [:adj] mais c'est [:un] [:noun :c]" }]], [# next lesson Block ...]]

```


**Dev Notes**

* Think I finally have a grasp on how to order and relate information in an
* Each lesson has associated vocab, this vocab will be added to the users vocab
list automatically.
* Definitions will be added to words on different variations. E.G. house can be a
verb and a noun etc.  The structure above allows us to make sure we only pull in
appropriate words that are symantically correct.
* Algorithm will heavily favour placing in new words that the user hasn't had much
exposure too.
* Algorithm will some what heavily favour words that are 'stale' to the user
(hasn't seen much recently), or that the user has a low 'rating' with, eg if they
have trouble remembering a particular word.
* Considering adding an option to allow more words of the same type to be fed in
if the users 'rating' for all matching vocab is very high.


* **Format** will initially just be flash cards. Other methods on the backlog
include:
  * Single word
    * Multiple-choice
    * Written
  * Sentences
    * Multiple-choice
    * Written (might be too much)
    * Place words in the correct order


### Change Log: 0.03
* Implemented an API lookup to a multilingual dictionary.  It is functional but
not plugged into the application proper yet.
* Added a simple splash screen, may try to make something better later.
* Menus are now all working properly, dead options are coloured black until functionality is implemented.
* Setup Dotenv for storing API keys, tempted to move session info into Dotenv, haven't decided if it is a good idea.
* Began setting up json files handling lesson content.  Still undecided on the layout I want to go with.

**Notes**
* New dependencies: tty-table, dotenv

### Change Log: 0.02
* Greatly simplified codebase for logging in
* Removed many now unnecessary files
  * removed default_user_opts.rb
  * removed flashcards_form.rb
  * removed help_form.rb
  * removed login_form.rb
  * removed menu_form.rb
  * removed print_options.rb
  * removed session_controller.rb
  * removed settings_form.rb
  * removed views.rb
* Added Main Menu
* Added Flash Card Menu
* Better accessibility, keywords highlighted

**Notes**
- user_controller needs to change now that users can no longer type their own inputs to navigate.
- Flash Card menu wish list
  * Would like to add a feature showing what % of words they know later
  * Would like to add a feature recommending a list to study
  * Would like to add a feature displaying if the list is |Fresh| |Stale| |Unlearnt| so they can better decide what to study.

###Change Log: 0.01
- Started an active devlog within README.md to provide better communication of changes.
- TTY-Prompt replacing old menus where user would have to type the response. Provides better user experience.  Allows for a simpler and easier to understand codebase.
- Colorize integrated, using it to increase accessibility and convey information more clearly.

**Dev Notes**

* Need to increase testing coverage.  Need to learn RSPec in more depth to do so.
* Views folder likely deprecated. Will remove after all functionality is handled elsewhere.


**Codebase Maintenance**

* Consolidate exceptions, too many unnecessary files.
```

```
  * Should be able to simplify code considerably, it's on the backlog.
    * Default user options probably redundant
    * Everything within views likely redundant
```