

# Project Self-Assessment

Initial Idea: Arithmetic study with guests

\* Other initial ideas included in repository ↓ \* Changed due to functionality issues

final Idea: French learning application

Progress 1: Loading screen with a

'translator' button

↳ English words could

be translated to the

50 most common French

words

\* Initial ideas code & files in repository

Progress 2: Home screen added with

a new 'start' button.

↳ 'Translator' button moved

to the home screen

Progress 3: Added a new 'dictionary.txt' file containing significant amounts of translation.

↳ Utilised MATLAB functions to read the text file  
\$ Subsequently greater translation was enabled.

Progress 4: Formulated a text animation for the loading screen \$ changed text, background & button colours to match French flag colours: 

Progress 5: 'Study Music' feature added to the home screen with a runtime label showing runtime of the .wav file.

↳ lofi music used due to my own love for it

**Progress 6:** 'Exit' button & additional labels such as, 'Developed by', 'Bienvenue!' & volume slider added to let users toggle volume for study music.

**Progress 7:** 'Challenge' mode added initially mimicking the translation buttons functionality.

**Progress 8:** 'Challenge' mode changed to allow for streaks & word guessing.

↳ Levels feature added with different text files.

→ Level 1: 50 most common words

→ Level 2: Next 100 most common words

↳ level 3: 100 far harder words

↳ 'Streaks' & responses added to user guesses.

↳ Levels increment when user gets 5 in a row also implemented.

Progress 9: 'Story' mode added with 3 graded readers:

- Sherlock Holmes
- Baskerville Hall
- Fixing the Nets.

Progress 10: 'Select your story' page added mimicking 'Select your level'.  
↳ Extra exit buttons added to streamline functionality.

# Review

- I initially had the idea to make an arithmetic quest game but due to drawbacks with functionality & a desire to use MATLAB's appdesigner I elected to develop something I would actually use.
- Incremental development was utilized as illustrated above
- 30+ functions & several more assets used
- Test cases ran with every progress in development

# Rubric

## Conceptual Coverage:

- Input/Output:
  - The application is essentially all input & output
  - Buttons, Sliders & text box's (translate/challenge) functions all take user input & produce output.
  - The output is greatly dependent on the function
- Loops:
  - A combination of for loops were included.
  - While loops weren't used primarily due to their issues with ful:
    - Blocking & state management.

- Vectors/Matrices:
  - The primary vectors/matrices
    - ↳ As well as: used within code for cell arrays, logical manipulating & sorting string vectors(boolean) & data. Often in the parsing made by the translation/challenge functions in the application.
  - Matrices are also often used for position & colour context. Ie: [100, 100, 600, 400] for the Challenge UI.
  - Cell arrays were also often used, ie for:
    - french words {}
    - english translations {}
      - ↓
      - Appending to cell arrays was also executed:
        - frenchWords{} and t1{} = parts {13}.

- Conditional

Execution:

↳ Switch  
if  
else  
else if

- Switch, if, else & else if  
Conditional statements were all used several times:

- Switch: Used for levels in challenge mode.
- If & else: Used several times for example:
  - line 710 - 715 for exiting out of challenge
- If, else & else if: Story type , line 317 - 323.

- Functions:

- 30+ functions made with their own files for each.
- Vital to the application for UI creation, interactivity & navigation .

## Value Add:

- **functionality:**
  - GUI & appdesigner used to add substantial value with great performance & execution
  - It can be used for any French learner such that they find value in all three main modes.
  - Text based animation & sounds adhering to the app's core purpose also create perfectly
- **Extensions:**
  - Multiple levels of challenges
  - Dynamic content loading
  - Streak & error tracking with conditional level placement
  - Error handling, making the app more user friendly

- Adaptive UI responses, with the immediate feedback users receive
- Navigation features with exit push buttons
- Additional features, such as study music, story mode, volume slider & levels.
- Clean UI that follows the theme of the app.

## • New Features

used:

- UI control & creation:
  - 'uifigure', 'uibutton', 'uilabel' & 'uieditfield'
- Dynamic Content Loading:
  - 'fileread'
- Object Orientated Programming:

- Class is identified & used.
- Nested functions:
  - 'checkfunc'
- Conditional checks
- txt file usage
- Cell arrays
- Timers
- Audio management.

## Incremental Development

- Various progress files were implemented as well as videos showing the apps development.

## Testing:

- All of the functions have driver files with various scenarios.
- Each progress is also a testing stage.

## Comments & style:

- Comments through the entire script
- Descriptive & useful comments added to aid understanding.

# Self-Assessment

## Project Rubric

Criteria	Ratings	Pts
Conceptual coverage Demonstrates correct use of MATLAB programming concepts	<p><b>40 Pts</b> <b>Excellent</b> All concepts covered mentioned in the practical are demonstrated and correct.</p> <p><b>30 Pts</b> <b>Substantial</b> Most concepts covered.</p> <p><b>20 Pts</b> <b>Pass</b> Covers "required" concepts: I/O, Loops, Vectors or 2D Arrays, and Conditional Execution</p> <p><b>10 Pts</b> <b>Below Expectations</b> Some, but not all required concepts covered</p> <p><b>0 Pts</b> <b>Absent</b> No key concepts covered</p>	40 pts
Value-add The amount of value that you added in your assignment. How much coding and conceptual effort is demonstrated by your code. Is the code that other sources contributed clear from your comments? Is the functionality of that code (as the program runs) substantial?	<p><b>20 Pts</b> <b>Excellent</b> Excellent functionality. Interesting code behaviour with substantial personal contribution - Demonstration of creativity in design of interaction with user and/or functionality. expectation of at least 3 extensions beyond minimal functionality and evidence of self-initiated learning (use of new MATLAB functions or techniques). Other's contributions (if any) clearly delineated. Functional code length 400+ lines of modular code.</p> <p><b>17 Pts</b> <b>Substantial</b> Adds functionality beyond minimum requirements demonstrating thought about ways of improving usability or user experience. Other's contributions (if any) clearly delineated. Functional code length 250-400 lines of code showing modularity.</p> <p><b>13 Pts</b> <b>Pass</b> Implements the base functionality of the project (ie meets the minimum requirements to say the project 'works'). Others' contributions (if any) clearly delineated. Expected functional code length 150-250 lines.</p> <p><b>6 Pts</b> <b>Below Expectations</b> Incomplete project with partial functionality or Unclear contribution to code (unable to explain parts of code claimed as own). Functional code length &lt; 150 lines.</p> <p><b>0 Pts</b> <b>Absent</b> No or minimal functionality.</p>	20 pts
Incremental Development Are there intermediate MATLAB and test files that indicate a clear path to development of the project.?	<p><b>15 Pts</b> <b>Excellent</b> Substantial evidence of development through well-commented intermediate files embedding small increments of development. And, where-appropriate, files implementing individual components of the solution.</p> <p><b>11.25 Pts</b> <b>Substantial</b> Evidence of development through commented intermediate files allowing small increments to development to be tested.</p> <p><b>7.5 Pts</b> <b>Pass</b> Some files indicating a path to development of the program.</p> <p><b>3.75 Pts</b> <b>Below Expectations</b> Minimal evidence of a path to development.</p> <p><b>0 Pts</b> <b>Absent</b> No evidence of a path to development.</p>	15 pts

<p><b>Testing Strategy</b></p> <p>Evidence of testing through test files and intermediate versions of MATLAB code.</p>	<p><b>15 Pts</b></p> <p><b>Excellent</b></p> <p>Evidence of careful stages of each stage of program development and testing of individual program components.</p>	<p><b>11.25 Pts</b></p> <p><b>Substantial</b></p> <p>Evidence of careful testing of multiple stages of program development.</p>	<p><b>7.5 Pts</b></p> <p><b>Pass</b></p> <p>Some evidence of careful testing of at least one stage of program development.</p>	<p><b>3.75 Pts</b></p> <p><b>Below Expectations</b></p> <p>Some basic tests.</p>	<p><b>0 Pts</b></p> <p><b>Absent</b></p> <p>No tests files.</p>	<p><b>15 pts</b></p>
<p><b>Comments and style</b></p> <p>Consistent use of indenting. Consistent and sensible use of variable names.</p> <p>Commenting throughout all versions of program.</p>	<p><b>10 Pts</b></p> <p><b>Excellent</b></p> <p>Code consistently exhibits all elements of good code style consistently through all versions.</p>	<p><b>7.5 Pts</b></p> <p><b>Substantial</b></p> <p>Code exhibits elements of good style for nearly all versions with only minimal lapses.</p>	<p><b>5 Pts</b></p> <p><b>Pass</b></p> <p>Code shows some elements of good style across all criteria but inconsistent.</p>	<p><b>2.5 Pts</b></p> <p><b>Below Expectations</b></p> <p>Code addresses one or two style criteria but not others.</p>	<p><b>0 Pts</b></p> <p><b>Absent</b></p> <p>Poor or inconsistent code style or minimal coding.</p>	<p><b>10 pts</b></p>