THE LAZY DEVELOPER’S TOOLKIT

Avery, Brogan M

Index

March 11, 2021

Project Proposal

Project Idea Title: The Lazy Developer's Tool

A core motto of mine has always been to work smart, not hard. So when I recently came across a few articles describing why "lazy people" make the best programmers, I really resonated with many of the concepts discussed.

Basic Concept: A program that provides a number of tools for an efficient (or a lazy) developer to optimize their time when writing and working with C++ code through the use of resource/ information consolidation, minimizing redundant coding, etc. with the ultimate goal of minimizing the time spent working on a project as much as possible (specifically to minimize my time sitting in front of a computer screen writing hard code).

Possible parts/ sections of program:

- Job Search Automation
  - Use APIs to make looking to a job easier

- Two "Custom Libraries"
  1. A help library that contains functions with print methods describing information on an aspect of C++ such as a syntax rule or other explanation of how something in the language works. Examples could be:
     - A method that prints out a list of all of the operators used in C++
     - A method that gives me the definition and description for a topic/ feature such as abstraction or how to use operator overloading
     - A list of the keywords in C++ and what they mean
     - The difference between different containers

  2. A list of common functions or (short cuts) I can call to limit redundant typing and make the main function and other files in future projects look cleaner and be completed sooner.

- A way to automate testing better

There really are (almost) a million different features that could be added on to this project. The above examples are some of the ones I would like to focus on, however there is a lot of room to adjust this project as I go and to continue to add to it outside of the scope of this class assignment/ time frame. Not only is this an actually useful program specific to me, this is going to make me spend a lot of time really getting to understand some of the concepts of C++ that I feel my skills are most shaky in.

March 18, 2021
Lazy Developer Progress Report 1

Detailed Project Description

Due to some issues I encountered when looking into APIs for a job search, I have decided to redefine a few parts of my project. To replace this part of the program, I am still considering what specific direction I want to take, but I will still be using python to either call an API or web scrape some information (the likely option is a daily code problem that I can pull from a web page). Then to access that information in the C++ part of the program, I will either save the results to a text/csv file where the C++ program can be ran after the python program runs to write to/ read from the same file. Or, I will use some other method to communicate data from the python code to the C++ code (requires further investigating).

Additionally, While I am enjoying how this project is making me do a lot of research and really get a better understanding of how some of the important parts of C++ work, and although some of my code is going to be a bit challenging to think out, I felt like I could make this a little more complex, and also because I just really have a strong desire to include n Arduino in a final project, I am also going to add a another part to the project. Depending on how simple or challenging this task takes for me to complete, I would minimally like to instruct an Arduino to allow me to turn on a red or a green LED to let other people in my work area to know whether or not to disturb me.

Because I am working with some stuff that I have little to no experience with, I am trying to build the project in a way that allows for wiggle room to grow or shrink the project once I dive deeper into some of it.

The updated outline for this project is as follows:

Main parts:

1. Custom Library and Shortcuts
2. Code Problem of the day
3. Do not disturb button and lights Arduino project
4. Tying it all together

Break down / description of each part:

Custom Library

Tentative methods / parts

- Dictionary that stores definitions for C++ related terms and way to access it and store it externally
- Quick print
- Generate a template for new classes
- Create new file
- More TBD…

Code Problem of the day

Specifics TBD

Arduino portion

1-2 buttons that let me control 2 LEDs powered by an Arduino

Tying it all together

All though this is not a separate part on its own, this will be by far the biggest challenge of the project to attempt. While I have not full determined if I will be able to make my 3 separate parts of the project accessible under one singular menu selection type C++ program, that is my goal. This will be challenging because I will need to make a way for the C++ program to control the python code and the Arduino.

Auxiliary GitHub/ project files

- A read me that defines what the project is and how to use it
- The csv files, or database

April 4, 2021
Lazy Developer Progress Report 2

Updates

While there are still a few places of the program that I need to figure out how I would like to make them work, I have at this point, tested out enough possible options that I am much more comfortable giving a more accurate and detailed description of what can be accomplished in the scope of this project. I have determined that this product (the developer toolkit) will function like a dashboard with many (6 as of now) different developer tool widgets. Because this will most likely remain text based, the user will navigate from the dashboard into the widgets and back out again via a series of Menu screen. I have created an outline of the structure of the main files of the project, but will likely include several more helper classes / files. In addition to those already mentioned, the other deliverables I have for this week are :

- Updates to back log and time log
- What the user will see as output at each Menu Screen (this really helps to understand the structure and navigation of the program.
- Pseudocode

Dead Lines

Due by END of week:

Week 10:
-Pseudocode/ working model
- Progress Report 2
- push to GitHub?

Week 11:
- Progress Report 3
- produce phase 1 of code
- push to GitHub

Week 12:
- Progress Report 4
- produce phase 2 of code
- push to GitHub

Week 13:
- Progress Report 5
- produce phase 3 of code (essentially final version of code)
- push to GitHub

Week 14:
- writing and running tests/ final fixes and changes
- Progress Report 6
- push to GitHub

Week 15:
- create video and other requirements for final turn in
- Final Report
- push to GitHub

Time Log

March 14 – research and brain storming project components – 4hrs
March 16- API research -3hrs
Other research - 1hr
March 17 – API and web scrape research - 4hrs
March 18 – API research – 3hrs
March 19 - brain storm and research new ideas to replace a

- Came up with some ideas that I can use an Arduino or raspberry pi for such as a button I can push that turns on a red or a green light at my desk to signify if people are allowed to approach me or if I need to be left alone in silence to focus. And/ or a blue light that is connected to a timer so that it turns on every 30 min/ 1 hour to remind me to get up and stretch and drink water.

March 23 – wrote an updated outline - 1hr
March 25 – did some research on how the general code and set up for the Arduino will work – 1hr
March 26 – looked into different ways to make a C++ program control the Arduino and Python – 2hrs
Created back log to hold “extra ideas”
March 27 – worked on pseudocode and still trying to figure out how to connect the pieces – 4 hrs
March 28 – worked on dictionary functions and with the stl libraries 6 hours debugging libraries finally made some progress with the python script via terminal commands
Added a few more things to back log
March 29 – set up basic project structure in Xcode and decided to learn how to play sound files
March 31- add formatting to report, add more to pseudocode
April 1- reformatted project, worked with python more, added to the relaxation folder
April 3- worked on setting up python libs that will work with C++
April 4 – organized the most recent progress reports and updates
April 11- completed the first phase of code, pushed to GitHub
April 12 – added another dev tool class that will inherit from all the others so all of their functionalities can be accessed by one object without making the user go through layers of menu selection screens if they do not want to.
April 13- added hangman and Candyland games to the 5th dev tool
April 16- experimented with JavaScript and looked into other alternatives for opening a webpage, ended up deciding I have spent too much time trying to do it the most programmatically and have decided to settle for a minimally viable product in this area
April 18- worked with servers on dev tool 3
April 19- worked with servers on dev tool 3, attempting to get the CSS to work
April 21 – ran to code to add in more comments and make sure that the comment make sense
April 22 – began writing the product documentation and user manual and creating the power point
April 23 – began testing
April 26 – continued testing and writing the final reports, collecting all of the files, etc.

Back Log

Ideas:

- Play stress reliving music
- Way to store website articles/ links better
- Command line management (run command line commands from C++ file)
  - Text to voice
  - Run / execute other files and programs
- More Arduino things
- Some sort of GUI?
- Testing
- Debugging/ reading error messages and codes
- Expand a developer tools section to include metrics
- Version control connection to GitHub
- Continue to add to My Custom Library as I learn and progress as a developer and see the need for more shortcut functions (or ideas for categories that would include multiple functions) such as:
  - Error handling
  - Menu selection
  - For loops
  - While loops
  - If/ case/ switch/ or general logic handling
  - Build data structures (or custom variations of the ones included in the stl such as searching and sorting algorithms, etc……
  - Creating databases and editing tables
  - String formatting (spacing, punctuation marks, etc)
- Continue to add to the dictionary CSV these terms / topics:
  - Common and most usful libraries in the STL (and there most usful methods / what they do)
  - Define many different operations in C++ and their associated syntax by including a definition and an example of how to create or use it, for example:

Call method that looks up a string key in dict/map("vectors")
Returns string "

Explanation: *A vector is a container template that is similar to an array. However the main difference is that they are dynamic. Etc……*

List Methods and how to use it: push_back(), pop(), Etc…

Examples:

Int <verctor>myVector;

"

April 4, 2021

Program File Structure / Organizational Outline

1. Main/ Driver File
    a. Description: Contains program start/ end functionality, including product introductions/ instructions. Also contains Main Menu Screen type of control structure for the user to navigate to other parts of the program and back again.
2. DevTool0
    a. Description: *Toolkit Dashboard*. A base tool that has 'My Custom Library Functions' and a few virtual functions that all of the other Developer Tools will use and be based from.
3. DevTool1
    a. Description: *Skills Builder.* This Tool is meant to help user improve their coding skills
4. DevTool2
    a. Description: *Arduino Based Tools.* This is a collection of tools that in some way incorporate an Arduino to make a developers life easier.
5. DevTool3
    a. Description: *Project Management.* This tool has features that helps a developer manage and organize all of the parts of any new project from beginning to end.
6. DevTool4
    a. Description: *Project Testing.* This goal of this tool is to improve the product testing process and will help a developer become more productive by spending less time testing and debugging.
7. DevTool5
    a. Description: *Relaxation Management.* This tool is meant to help a developer who needs to take a break and relax or improve their mental health in order to think more clearly on a project.
8. DevTool6
    a. Description: *Other Resources.* This tool contains various documentation and access too other files, etc.
9. Project
    a. Description: objects from the project class will be used to make this developer tool program usable with other real projects. They will be used to "start a new project". It will contain attributes for things like the project title and description as well as attributes that correspond to/ or enable use of some of the above DevTool classes. This is to allow the developer to quickly look up things about a product that they are working on such as project due dates.

Menu Screens seen by user :

Main Menu Screen:
1. Skills Builder
2. Arduino Based Tools
3. Project Management
4. Automated Project Testing
5. Relaxation Management
6. View Other Resources
7. Exit Program

Skills Builder Menu Screen:
1. Practice Coding Problems
2. Play a Programing-Related Quiz Game
3. Return To Developer Tools Dashboard

Arduino Based Tools Menu Screen:
1. Do Not Disturb Feature
2. Return To Developer Tools Dashboard

Project Management Menu Screen:
1. Generate Project Management Guide For New Project
2. Generate Wireframe
3. Return To Developer Tools Dashboard

Project Testing Menu Screen:
1. TBD
2. TBD
3. Return To Developer Tools Dashboard

Relaxation Management Menu Screen:
1. Listen To Relaxing Music
2. Play a Fun Game
3. Hear a Joke
4. Look at Baby Animals to Make You Smile
5. Return To Developer Tools Dashboard

Other Resources Menu Screen:
1. View My Dictionary
2. View My Useful Web Links
3. View Product Documentation
4. Return To Developer Tools Dashboard

April 4, 2021

Pseudocode

Main() function in Driver file:

```
START PROGRAM
// declare variables
        DECLARE userInput = “ ”

        DECLARE devTool0 // This is an object of a class acts as a base for the others and contains many short
        cut functions that basically make common things faster to code. There is a lot of room to add more to
        this but here is a list of functions that will be included minimally in this base class:
        -    A virtual execute function for the derived classes
        -    A function to create a new IO file
        -    A function that generates a template for class headers so I can type less
        -    A function that capitalizes a word
        -    A function that allows user to append a line to the end of a file
        -    A function to get a line count from a file
        -    A function to read a file
        -    A function to add words to the Custom Dictionary feature of this product
        -    A function to look terms up in that dictionary
        -    A function to add web sites to a list of useful websites
        -    A way to look up and open those links in a browser
        -    A function that reads text to speech
        -    ( more possibilities can be found in the back log )

        //Declare an object of a sub-class (they will all be derived from a base class that has one or more virtual
        functions) that is associated with each developer tool ( essentially just responsible for executing the
        code that is associated with the functionality of that developer tool and navigating from the main
        program section to other sections of the program and back to main again)
        DECLARE devTool1
        DECLARE devTool2
        DECLARE devTool3
        DECLARE devTool4
        DECLARE devTool5
        DECLARE devTool6

PRINT basic program intro……..
PRINT basic program instructions………

PRINT “Select number associated with a developer tool
     1. Skills Builder
     2. Arduino Based Tools
     3. Project Management
     4. Automated Project Testing
     5. Relaxation Management
     6. View Other Resources
     7. Exit Program “
```

```
WHILE userInput is not “  7  “
    GET userInput from user

    IF userInput = “ 1 “
        devTool1.execute()  //Call method to execute the first Developer tool, Skills Builder
    IF userInput = “ 2 “
        devTool2.execute() //Call method to execute the second Developer tool, Arduino Based Tools
    IF userInput = “ 3 “
        devTool3.execute() //Call method to execute the third Developer tool, Automated Project
        Testing
    IF userInput = “ 4 “
        devTool4.execute() //Call method to execute the fourth Developer tool, Relaxation
        Management
    IF userInput = “ 5 “
        devTool5.execute() //Call method to execute the fifth Developer tool,
    IF userInput = “  6 “
        devTool6.execute() //Call method to execute the Sixth Developer tool, View Other Resources
    IF userInput = “  7 “
        // Exits loops and continues to the end of the program

PRINT “Thank you for using the Lazy Developer’s Toolkit, Goodbye”

END PROGRAM
```

Class DevToo1, Inside execute() function:

```
PRINT “ * Skills Builder * “

PRINT basic description about this Developer tool(s) and how to use it…….

PRINT “Select number associated with option
    1. Practice Coding Problems
    2. Play a Programing-Related Quiz Game
    3. Return To Developer Tools Dashboard “

WHILE userInput is not “  3  “
    GET userInput from user

    IF userInput = “ 1 “
        // calls method that executes a python script ( probably through some terminal command)  that
        web scrapes many titles of practice problems that the user can choose from. Then will allow
        user to select a problem and it will open a weblink in the user’s browsers for that specific
        problems explanations, help, and solutions. ( this will rely on the ‘command line system()
        function(s)’ of C++ or the use of an addition CSV if I can’t  make the other connection work
        properly)
    IF userInput = “ 2 “
        // similar to option one, only the Python script will call an API instead of web scraping
    IF userInput = “ 3 “
        // exits loops and returns to the main functions/ main program menu
```

Class DevToo2, Inside execute() function:

```
PRINT “ * Arduino Based Tools * “

PRINT basic description about this Developer tool(s) and how to use it…….

PRINT “Select number associated with option
    1. Do Not Disturb Feature
    2.  // maybe another Arduino based tool TBD
    3. Return To Developer Tools Dashboard “

WHILE userInput is not “  3  “
    GET userInput from user

    IF userInput = “ 1 “
        // Calls a method associated with that feature and inside it looks like this:
        START METHOD
            PRINT "Turn the green light on to indicate to co-workers (or other humans) if you are
            available to speak with or red if you are busy and do not want to be disturbed."

            PRINT  “ Select number
                1. Set light to GREEN
                2. Set light to RED
                3. Turn all lights off
                4. Return to Previous Menu Screen “

            WHILE userInput is not “  4  “
                GET userInput from user

                IF userInput = “ 1 “
                    // calls method that uses the Arduino CLI and associated functions to
                    trigger a file that instructs the green led on the Arduino to turn on
                IF userInput = “ 2 “
                    // calls method that uses the Arduino CLI and associated functions to
                    trigger a file that instructs the red led on the Arduino to turn on
                IF userInput = “ 3 “
                    // calls method that uses the Arduino CLI and associated functions to
                    trigger a file that is essentially blank so it instructs the Arduino to not
                    power either light any more if either are on.
                IF userInput = “ 4 “
                    // exits loops and returns to the Previous Menu Screen
        END METHOD

    IF userInput = “ 2 “
        // TBD if time allows
    IF userInput = “ 3 “
```

```
        // exits loops and returns to the main functions/ main program menu
```

Class DevToo3, Inside execute() function:

```
PRINT “ * Project Management* “

PRINT basic description about this Developer tool(s) and how to use it…….

PRINT “Select number associated with option
    1.  Generate Project Management Guide For New Project
    2.  Generate Wireframe
    3. Return To Developer Tools Dashboard “

WHILE userInput is not “  3  “
    GET userInput from user

    IF userInput = “ 1 “
        // calls methods that generate an HTML based template similar to this report that will help
        keep aspects of a project in one place and well managed. The HTML file is not inted to be
        connected to a website or server, but it is a nice way to organize and display the information
    IF userInput = “ 2 “
        // similar to option 1, however, this tool will call an HTML file that help create something similar
        to a wireframe. Again, this is not actually intend to be used as a website or web page, but
        rather a tool that helps a developer conceptualize projects in a visual way for their own use or
        as a way to communicate ideas to co-workers, employers, etc.
    IF userInput = “ 3 “
        // exits loops and returns to the main functions/ main program menu
```

Class DevToo4, Inside execute() function:

```
PRINT “ * Project Testing* “

PRINT basic description about this Developer tool(s) and how to use it…….

PRINT “Select number associated with option
    1. TBD
    2.  TBD
    3. Return To Developer Tools Dashboard “

WHILE userInput is not “  3  “
    GET userInput from user

    IF userInput = “ 1 “
        TBD
    IF userInput = “ 2 “
        TBD
    IF userInput = “ 3 “
        // exits loops and returns to the main functions/ main program menu
```

Class DevToo5, Inside execute() function:

```
PRINT “ * Relaxation Management* “

PRINT basic description about this Developer tool(s) and how to use it…….

PRINT “Select number associated with option
        1. Listen To Relaxing Music
        2. Play a Fun Game
        3. Hear a Joke
        4. Look at Baby Animals to Make You Smile
        5. Return To Developer Tools Dashboard”

WHILE userInput is not “  5  “
        GET userInput from user

        IF userInput = “ 1 “
                // calls method that gives the user an option to pick from several relaxing songs. Then uses the
                ‘command line system() function(s)’ to make that song’s wav/mp3 file play. Still need to figure
                out a way to allowed user to make it stop however…..
        IF userInput = “ 2 “
                // STILL UNDECIDED: but will most likely be a very small quiz game, or could be based of games
                I have created in based projects or I may base something on a small game I find online
        IF userInput = “ 3 “
                // Calls methods that read in each line of a text file with jokes and responses. It then selects a
                random joke and reads it out loud using the text to speech functionality of the base DevTool
                class. It pauses before reading the answer to the joke.
        IF userInput = “ 4 “
                // calls method that gives the user an option to pick from several different websites that just
                contain images of cute baby animals. This is done using the ‘command line system()
                function(s)’, and will open a weblink in the users browser automatically
        IF userInput = “ 5 “
                // exits loops and returns to the main functions/ main program menu
```

Class DevToo6, Inside execute() function:

```
PRINT “ * Other Resources * “

PRINT basic description about this Developer tool(s) and how to use it…….

PRINT “Select number associated with option
    1.  View My Dictionary
    2.  View My Useful Web Links
    3.  View My Custom Library Documentation
    4.  Return To Developer Tools Dashboard”

WHILE userInput is not “  4  “
        GET userInput from user
```

```
IF userInput = " 1 "
    // calls method to open a CSV file of my custom dictionary or will read them into the program
IF userInput = " 2 "
    // calls method to open a CSV file of a collection of useful weblinks or will read them into the
    program
IF userInput = " 3 "
    // calls method to open a text document that will function as the documentation page for my
    custom library ( all of the actual function and code of which are in a header)
IF userInput = " 4 "
    // exits loops and returns to the main functions/ main program menu
```

As of April 22, 2021

User Manual / Product Documentation

Since this product is designed as a developer tool with the intended user being a developer or someone with a similar background and knowledge, it will be assumed that the user has an understanding of basic software development, and understating of general computer knowledge, and how to navigate and use their own OS and File Systems. It is also assumed that the user will be able to reconfigure various path files for their own systems.

Because it is a developer tool, the most common use of this is to simply only use a few parts of the entire program to meet the needs of whatever the developer does. It is unlikely that a developer will have the need to go through the entire program and use all of its pieces.

The program is designated as a multi-level menu screen and the user is free to use it this way, however, when the user becomes more familiar with the tools, they may find it more efficient and helpful to simply directly use the tools without having to navigate to them through all of the layers of menu.

Since this project has a lot of room to grow after the scope of this class, there are several areas in the menu navigations screens displayed to the user that describe some dev tools or features as "coming soon" or "still in development".

Dev Tool 0 – Base Tool Information

The base tool is a collection of functions that make shortcuts for common programming practices and needs. It includes the following methods/ features to perform the following operations:

- addDefinition(string term, string definition)
  - Accepts a term and a definition to add to the custom dictionary
- addWebLink(string title, string link)
  - Accepts a title/description of a website or webpage as well as a link to that site to be stored in a file
- appendLineToEOF(string fileName, string text)
  - Accepts a name of an existing file including the extension and the text that is to be appended to the end of the file
- capitalizeWord(string text)
  - Accepts a word and returns it with the first letter capitalized
- createFile(string fileName)
  - Accepts a name of a new file to create along with its extension and then creates the file
- createClassHeaderFile(string fileName)
  - Accepts the name of a new class header with the extension, for example: "MyNewClass.hpp". Then prompts user to enter the name of the class and the variables a long with their data types. Once the user is finished, it then generates a new HPP file where ever your machine creates new files (XCode of course being in an odd place and not right in the local project).
- getFileLineCount(string fileName)
  - Accepts a name of an existing file including the extension and returns the number of lines in the file

- lookUp(string term)
    - This method queries the Dev Tool Custom Dictionary for the term passed in as a parameter. Returns the definition of the term if it exists or returns a not found message.
- readFile(string fileName)
    - Accepts a name of an existing file including the extension and returns a vector with each item in the vector being a line of text from the file.
- textToSpeech(string text)
    - Accepts a string of text to write to a file. The file is then read allowed by the system commands.

Finding and Configuring Files and Project Parts

This project has a large number of external files. Due the variation in where files are stored on different machines, with different IDEs, or different compiler/processing behaviors, a more detailed explanation is required. Inside of the main project folder (where the main source file and other CPP and HPP files are located) there are 4 additional folders:

- Audio Files – this folder contains the mp4/wav files for relaxing music. All though they are directly sored in the same location as the other files of this project, the path currently written in the Dev 5 code is an absolute path based on my personal system and would need to be configured for your own system.
- Arduino Files – Similar to the audio files folder, this folder is stored directly in with this projects files but because it used the Arduino CLI which is on a different location on my machine, you will also need to configure to match your system.
- Python Files – This folder contains 3 files. One is a Python file that is used to web scrape information from a website, format the information and produce two files from the formatted text. This file is not meant to be run in the project, but is there to show how the files were generated 1 time to be used in other parts of the C++ project files.
- Other files – these are a copy of the files that would normally be generated directly in the same directory as other project files on Windows machines. But XCode IDE stores them in a different place outside of the project. These files are required for the functionality of much of this product so they will need to be moved to where ever the files are stored with in your machine/IDE. They directories will also need to be changed with in the source code to reflect your machine/IDE.

Other requirements/ important information:

- This product was designed on and for a mac OS. Using it with other operating systems could result in errors and unpredictable behavior
- Parts also rely on the addition of an Arduino microcontroller as well as Arduino software requirements and the Arduino CLI tool. The software components can be found at: https://arduino.github.io/arduino-cli/latest/commands/arduino-cli/ and https://www.arduino.cc/en/software as well as other pages on the Arduino official website

May 1, 2021

Final Report

Testing:

The testing of this project was done by running several different scenarios with good and bad input to test how the product would respond. This was done through the completion of the project as well as more heavily towards the final phases or versions of the product.

Summery:

Although I would like to add some more features to this project I am overall satisfied with the features I have provided. This tool will make planning and managing projects easier in the feature. It should also allow me to minimize my time spent writing code. The next thing I would like to work on this project outside of the scope of this class is to finished the dev 4 tool for testing and debugging.