# Python

## Python Libraries

After some research into different Python libraries, I have selected a few which might be beneficial to my project, this includes mainly GUI libraries, as most code will be handled by me. These libraries will only act to make my software more presentable to my audience. Python has a wide range of libraries to chose from when it comes to GUI programming, some libraries come preinstalled with my python install and some I will have install myself.

The first python library which I will be researching it Tkinter; Tkinter is a GUI python library which is pre-installed with python, this will not require me to try and install this library from an external source, making it far easier to use. Tkinter is considered one of the most popular python GUI libraries due to this point and its simplicity, this makes Tkinter a great start for me to get into GUI programming with Python. After using Tkinter to get a grasp on how I would like to use it in my project, I found the library very easy to use; although there was no designer to help me manage widgets in my window, I found the techniques used to manage the window space very easy to learn and I would consider this to be a great start for me to get my head around GUI programming in Python.

Tkinter also has very detailed documentation which is easy to follow, this allows me to get a grasp on anything I need to know about the library and how to use it (Python 2018). Although Tkinter may be the easiest GUI library to access, it is not the best GUI tool to as said by Chun (2006). This may consider Tkinter as the “go-to” python GUI library for beginners by default, allowing this to be a perfect introduction for me into GUI python Programming. As stated by Lutz (2014) Tkinter is a very simple and popular language with a lot of benefits such as documentation and its portability.

WxPython is another Python GUI library and is a common alternative to Tkinter which gives the ability to have a python GUI native on common operating systems such as Windows (Dunn, 2018). wxPython is an adaption of the wxWidgets cross platform C++ library. Similar to Tkinter, wxPython is well documented by providing all of the relevant information on how to use the various widgets. You can use the wxWidgets documentation to help create a GUI, however most of this documentation is written in C++, therefore requiring a broad knowledge of python to convert the syntax. As suggested by (Dunn, 2017) wxPython uses many different classes which will mirror most wxWidget classes, this allows the use of the wxWidgets documentation to be adapted to python as needed. As well as detailed documentation, wxPython provides a wide variety of classes to choose from, giving heavy customisability in your GUI application.

Another Python GUI library to consider would be Kivy; Kivy is an open source Python GUI library with a wider range of customisability in comparison to other libraries discussed already. Kivy can be used to create apps from many different platforms including mobile platforms such as IOS and android, allowing you to write one piece of code with compatibility with many different device operating systems (Kivy, 2018). Kivy also provide a well-documented guide on how to install and use Kivy, there is also a Python Programming guide included for users which have a vague understanding on the language. The library provides many different features such as using Widgets, drawing, managing layouts and graphics; this makes this library an excellent choice to consider when picking a Python GUI library for my project. This library is not installed with python, so I will have to go through the various steps on their installation guide on how to install the library. Kivy provides the option to design your program with forms not limited to the windows-based form giving strong customisability when it comes to design.

One other Python GUI library I would consider using would be PyQt. PyQt is a popular python library which has been adapted from a C++ framework. PyQt also covers a wide range of classes which can be used to create GUI’s, handle networking, databases and more (Python, 2018). PyQt has a wide range of documentation for both versions PyQt4 and PyQt5, as well as books to help create GUI’s (Summerfield, 2008). The Qt library provides a Designer which can be used to help design and manage my GUI application, this is a easier method than using any of the other libraries mentioned thus far as I will not need to code the positioning of each of the widgets and graphics on the window. However, this will still require me to manage the receiving and sending of data by each widget (Qt Company Ltd, 2018)

## Why Python?

As part of my requirements for my final year project, python is a very easy language to use which some people may consider as simple as pseudo code. As well as being one of simplest programming languages to learn, python can be considered one of the most useful with heavy use in computer networking. Python is one of the top 10 programming languages used and has many different uses; as stated by the TIOBE (2018) index, such as development in GUI applications, web development and data analysis according to Mindfire Solutions (2017). Python also has a wide range of libraries and a popular website such as PyPi provides easy access to a variety of libraries. Python can also perform high level programming tasks in significantly less lines of code compared to languages such as C++. Although python may be an easy language to use, it may not be your best choice depending on your task; however, in my case this language is efficient for my task of building a programmable Turing machine.

# Alan Turing

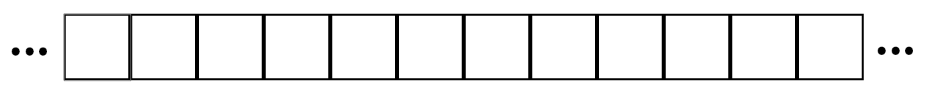
Alan Turing was an English mathematician who studied at Princeton University. One of Alan Turing’s major contributions to computer science was his paper named “*On Computable Numbers, with an application to the Entscheidungsproblem*” which is considered one of the fundamentals of modern computer science. This paper would cover the famous Turing machine and his Universal Machine which would in theory solve any computable problem by following a set of instructions (Watson, 2012).

## The Turing Machine

As Stated above the Turing machine is a machine which will be able to follow any set of instructions (Algorithm) to perform various computable tasks. The standard Turing machine is capable of writing down a series of 1s and 0s onto a tape with squares depending on the instructions based on each symbol the machine reads. The machine will also have a “head” to read through each square on the tape to gather its symbol, the machine would then use an instruction table to refer to before performing a task then advancing to the next square on the tape, (Mullins, 2012).

The Turing machine would work by being provided with a tape with a series of symbols (this tape could be infinite), the machine would also be provided with a series of instructions with different states and tasks (Instruction Table). The machine will then read the machine would then have to follow through the tape, read the symbols, then complete the task provided by the instruction table (depending on the symbol read on the tape). An example of a Turing Tape and instruction table will be provided below:

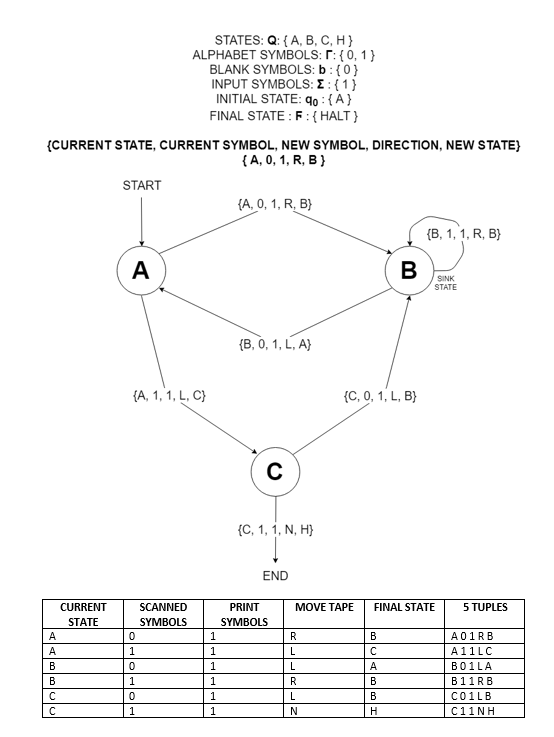
|  |  |  |  |
| --- | --- | --- | --- |
| **CONFIGURATION** | | **BEHAVIOUR** | |
| ***m-configuration*** | ***scanned symbol*** | ***operation*** | ***final m-configuration*** |
| A | NONE | P0, R | B |
| B | NONE | R | C |
| C | NONE | P1, R | D |
| D | NONE | R | A |



The Instruction table above provides an easy and simple to follow instruction set which will print an endless tape of 0s and 1s on the Turing tape. The machine would first check what m-configuration (state) the machine is in and then check if the scanned symbol matches the one presented on the current tape square, the machine will then perform the operation associated with the scanned symbol and the current state, then change to another m-configuration and perform the operation. The following symbols can be used in operation to tell the machine what to do next; the machine can erase the current square (**E**), Print a symbol (P followed by the symbol you wish to print (**P***n*)), and change direction (either Left or Right (**L, R**)). The machine can also take multiple instructions in its operation such as moving in any direction or print symbols more than once; this can be shown in the following simplified instruction table, Petzold (2008).

|  |  |  |  |
| --- | --- | --- | --- |
| **CONFIGURATION** | | **BEHAVIOUR** | |
| ***m-configuration*** | ***scanned symbol*** | ***operation*** | ***final m-configuration*** |
| A | NONE | P0 | A |
| 0 | R, R, P1 | A |
| 1 | R, R, P0 | A |

The head on the machine will only be able to scan one square at a time and will only be mindful of a one square at any given moment. The machine will then act appropriately depending on the current m-configuration of the machine and the scanned symbol. The instruction table can also be written as 5-tuples, this makes each instruction easily readable and gives a clear understanding on what the machine has to do. The table will now include 6 columns, with details on: the current state, symbol scanned, printed symbol, tape direction, final state and its 5-tuples representation as shown below with a state diagram for the machine (Wikipedia, 2018).



## Some Types of Turing Machine

In this section of my report I will be covering a few different types of Turing machine; all of which can be simulated on a single sided infinite Turing tape. One other variation of the Turing machine would be the multi-tape Turing machine, this machine is a standard Turing machine with several tapes. Each tape will have its own head which can operate individually to each other, initially the square on the first tape will have an input, whilst the other tapes are left blank.

MULTITAPE TURING MACHINE:

* Standard Turing machine with several tapes
* Each tape has its own head
* Initial input on tape 1
  + Other tapes are blank
* The transition function is changed slightly to comprehend multiple tapes from
  + - k is the number of tapes

## The Universal Turing Machine

Now that I have explained basic turing machines I can cover what the universal turing machine is…. A turing machine of turing machines, a machine which can do everything

## Fundamentals of a computer

Why turing machine makes up the basis of computers. READ THE ANNOTATED TURING BOOK

# My Plan

## how I will implement it

## pseudo code

## extra features (ideas)

* A binary counter
* Print hello world
* State diagram to follow the machine

### Ghantt Chart

# Appendix

## References

Python (2018) 24.1. Tkinter – Python interface to Tcl/Tk. Available at: <https://docs.python.org/2/library/tkinter.html> (Accessed 27 October 2018).

Chun, W, J. (2006) Core Python Programming: Second Edition.   
USA, Massachusetts: Pearson Education, Inc.

Lutz, Mark. (2014) Python Pocket Reference: Fifth Edition.   
United States of America: O’Reilly Media, Inc.

Dunn, R. (2018) wxPython 4.0.3. Available at:   
<https://pypi.org/project/wxPython/#description> (Accessed: 27 October 2018).

Dunn, R. (2017) wxPython Overview. Available at:  
<https://docs.wxwidgets.org/3.1/overview_python.html> (Accessed: 27 October 2018).

Kivy (2018) Available at:  
<https://kivy.org/doc/stable/gettingstarted/intro.html> (Accessed: 27 October 2018).

Python (2018) Available at:  
<https://wiki.python.org/moin/PyQt> (Accessed: 27 October 2018).

Summerfield, M. (2008) Rapid GUI Programming with Python and Qt.  
United States of America: Pearson Education, Inc.

Qt Company Ltd (2018) Qt Documentation. Available at:  
<http://doc.qt.io/qt-5/qtdesigner-manual.html> (Accessed: 27 October 2018).

TIOBE (2018). TIOBE Index for October 2018. Available at:  
<https://www.tiobe.com/tiobe-index/> (Accessed: 03 November 2018).

Mindfire Solutions (2017). Python: 7 Important Reasons Why You Should Use Python Available at:  
<https://medium.com/@mindfiresolutions.usa/python-7-important-reasons-why-you-should-use-python-5801a98a0d0b> (Accessed: 03 November 2018).

Watson, I. (2012) How Alan Turing Invented the Computer Age. Available at:  
<https://blogs.scientificamerican.com/guest-blog/how-alan-turing-invented-the-computer-age/> (Accessed: 04 November 2018)

Mullins, R. (2012) What is a Turing Machine. Available at:  
<https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/turing-machine/one.html> (Accessed: 04 November 2018)

Petzold, C. (2008) The Annotated Turing.  
Indianapolis, Indiana. Wiley Publishing, Inc.

Wikipedia (2018). Turing Machine. Available at:  
<https://en.wikipedia.org/wiki/Turing_machine> (Accessed: 04 November 2018)