**BCS Digital Industries Apprenticeship**

**Project B – Maze Game**

**Software Development Technician**

**October 15, 2018**  
Student Name : Jason Fernandes

Tutor : Vy Tran

Contents

[Introduction 2](#_Toc527674728)

[Project Brief 2](#_Toc527674729)

[Purpose of the document 3](#_Toc527674730)

[Scope 3](#_Toc527674731)

[Who with? 3](#_Toc527674732)

[Project Overview 4](#_Toc527674733)

[- Solution Overview 4](#_Toc527674734)

[Analysis 5](#_Toc527674735)

[- Non-Functional Requirements 7](#_Toc527674736)

[- Choosing between platforms 9](#_Toc527674737)

[World Wide Web 10](#_Toc527674738)

[Desktop 13](#_Toc527674739)

[Mobile 15](#_Toc527674740)

[Recommendation: Desktop 17](#_Toc527674741)

[Implementation 18](#_Toc527674742)

[- Best practices for code development 18](#_Toc527674743)

[Testing 19](#_Toc527674744)

[Test plan (After carrying out testing) 22](#_Toc527674745)

[Documentation 23](#_Toc527674746)

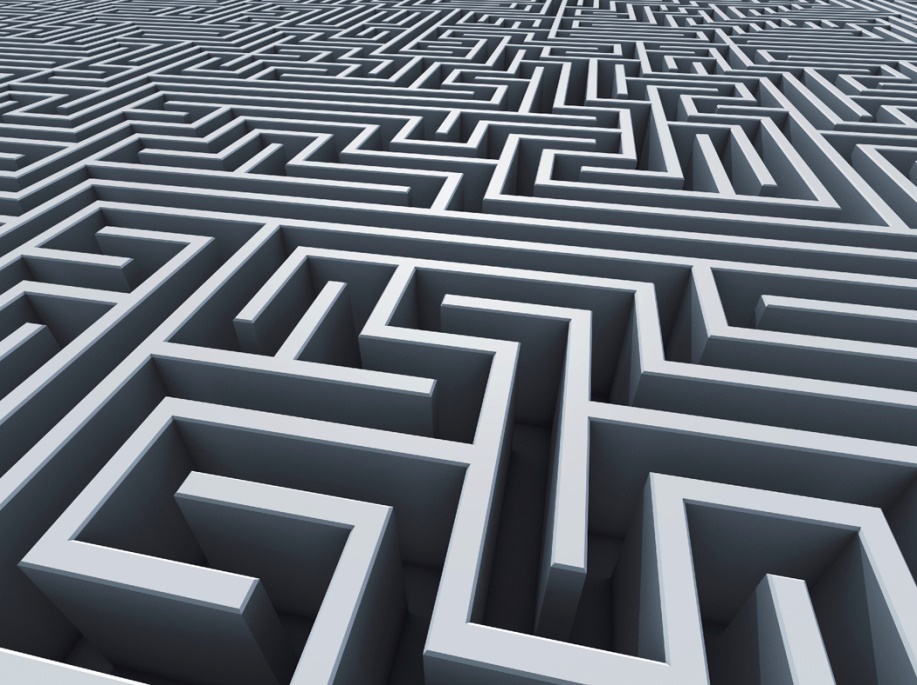
[- Minimum Hardware and Software requirements Python – Minimum Requirements for Python 23](#_Toc527674747)

[User Guide 24](#_Toc527674748)

[How to play the Maze game? 25](#_Toc527674749)

# Introduction

Project Brief

In order to complete the apprenticeship, my line manager: Katie Hirschovits assigned me “Project B – Maze Game.” This project would entail planning, designing, implementing and documenting a maze game in order to increase the number of visitors that click on Olde Worlde Phunne website. As said in the initial brief, the steps that should be taken are outlined as follows:  
  
1. Review all the key information and create a design for the maze game  
2. Construct the maze game in accordance with the design  
3. Test the maze game meets its requirements  
4. Document what you built  
  
The reason why Olde Worlde Phunne, a gaming company has suggested to create a maze game is because they want this program to be fun and interactive, in which will entice users to their website.

Purpose of the document

I am creating this document because it will show the evolution of the maze game from start to finish. The reason why Olde Worlde Phunne wanted a Maze game to be created is because it is a fun, interactive way in which will entice users to visit their website and consequently look at the other games in which they provide. I have chosen to develop this game on Python 3.6 because I am most proficient on this software, as well as the client, Olde Worlde Phunne have said that they would like me to use Python, since their current programmers use Python and therefore if editing needs to be done in the future, they will have the skills/be equipped to carry out the necessary tasks.  
  
The reasoning behind the Maze game is because it is a puzzle that requires logical thinking in order to win. Furthermore, it also requires being able to predict the next step as in the Maze game, the user will be required to choose the direction in which they want to travel to, in order to try escaping the maze.

Scope  
  
As Olde Worlde Phunne “want to increase the number of visitors to their website by offering a free maze game”, this consequently indicates that this game will be released to all users whom choose to visit Olde Worlde Phunne website. To get full access to the game, potential customers will need to sign up and create a new login on the website.

Who with?

I will be conducting all of the stages in the software development life cycle alone for this project; however, I will be regularly documenting all of the required information within this document. The reason for this is because, if improvements/edits need to be made, then a developer will be able to follow this documentation in order to edit the program successfully.

# Project Overview

## - Solution Overview

**How the game should work**

I will be creating a maze game on python that will aim to attract potential gamers to the website of Olde Worlde Phunne and consequently entice them to sign up/create a login to play the full version of the Maze game and play other games that Olde Worlde Phunne provides.  
  
How the maze game should work is that when the program is run, a maze should be automatically configured from a text file and outputted to the user if it is within the correct format. Once the maze is created, it will allow the user to start playing, the default player being set to no wealth and dropped in a random room within the maze.   
  
To make the game more interesting, Olde Worlde Phunne has asked to integrate a level of complexity in which items are found in each room. Both threats and treasures are scattered within the various rooms of the Maze and will enable the user to get either rewards or penalties towards their wealth. Every move that the user makes, they will be able to identify the current threats/treasures in that room, so that they will be able to make a judgement about which passage they should choose for their next move. It is mandatory that each threat is eliminated before the user chooses a pathway that leads to another room. The end goal is that the user identifies the correct pathway that leads them to the exit of the maze in which they are able to see their total wealth.

Analysis  
- Functional Requirements

Functional requirements - concerned with system services such as the scope of the system, the necessary business functions and the required data structure.

|  |  |  |  |
| --- | --- | --- | --- |
| Req Number | Functional Requirement | Reasoning | Output |
| 1 | Specifies the text file that contains the maze configuration data and outputs it on the screen | As stated in the brief, the python program should retrieve the particular data within the text file which will determine the maze shape |  |
| 2 | The game starts only when the maze has been configured properly | If the maze game is not retrieved from the text file then the game should not run, since it would not work properly. Therefore an error message should appear when this occurs, so the user can try running the program again |  |
| 3 | When the game starts, the player is automatically launched in a random room within the maze | It is important that the player starts in a random room because if the user chooses to play the game multiple times, then each time is able to be different |  |
| 4 | When the game starts, the player should begin with no wealth | Player should start with no wealth, as when they make their own way through the maze they will receive points for the treasures found |  |
| 5 | Each room within the maze are connected to each other by pathways | In order to travel from one room to another within the maze, it is essential that pathways are created. |  |
| 6 | The game will report to the user which threats and treasures are to be seen in the current room. | As stated in the brief, the user should be able to see the current treasure or threat within the room they are in until they collect/deactivate the item |  |
| 7 | When Treasure is found/threat is defeated, it should be removed from that specific room for the remainder of the game | The game should update specifically when either the treasure is found/threat is detected, so the user is able to determine what their next move should be |  |
| 8 | When treasure/threat is found/deactivated it gets added/subtracted to the current wealth of the user | A calculation needs to be performed in order for the user to be awarded/deducted for finding the treasure/threat |  |
| 9 | All treasures and threats should be collected/eliminated before leaving the room | As said in the brief, all items should be retrieved before the maze game finishes, since otherwise there would be no point of a score. |  |
| 10 | A threat is eliminated by carrying out the Action that defeats it. | Since there is going to be both treasures and threats within the Maze game, it is essential that all threats are found and deactivated as said in the brief |  |
| 11 | A game may be started at any time during play and will reset the maze template from the text file randomly | The user should have the ability to start the game again if they choose to. By adding this, it will make the game more flexible and less rigid. |  |
| 12 | Each room may have 1 to 4 passages leaving in the directions of North, East, South or West | The number of passages available to take from each of the rooms will differ, thus making it harder to find the exit. |  |
| 13 | If the user chooses a direction in which there is no passage, the program should announce “You may not go that way” | As said in the brief, the program should indicate to the user that there is no passage the way they are choosing to go. |  |
| 14 | When exit pathway has been found, the game finishes | The maze game should end automatically once the user has found the correct pathway leading to the exit. |  |
| 15 | When the game is finished, the user’s final wealth should be calculated and outputted. | When the user has found the exit pathway, the game should automatically finish and output the final wealth of the user |  |

- Non-Functional Requirements

Non-functional requirements - Deals with system constraints on the proposed software such as operation, how easy it is to use, performance and security. Potential requirements are methodically analyzed and written on the requirement specification that serves the basis for all future development on the software.

|  |  |  |  |
| --- | --- | --- | --- |
| Req Number | Non-Functional Requirement | Reasoning | Output |
| 1 | The game must be easy to use and play | As the game is aimed at new users, it must be easy to play otherwise the user will lose interest. |  |
| 2 | The game must be available 24 hours per day, 364 days per year | New users should be able to use the game whenever they load the website. Websites are able to be accessed as long as there is a valid internet connection. Therefore, the game should be readily available all the time. |  |
| 3 | The program should look into the text file and copy template of maze within 10 seconds | The game should not take more than 10 seconds to retrieve the data from the text file due to performance targets and new users may leave the game because it is taking too long to load |  |
| 4 | The game should not disclose any personal information about the user to the operators of the system when new users create a login due to data confidentiality |  |  |
| 5 | There should be maintenance and administration support that take care of adding and removing users and their privileges, monitoring the application and software distribution/installation. |  |  |
| 5 | Two different types of users,  - Free trial user who has access to a limited amount of the game.  - Full Member – Access to the all parts of the Maze game |  |  |
| 6 | The game must be reliable and if unable to process the request then appropriate error message should show |  |  |
| 7 | The game should be fully protected against competitors, so they are not able to copy right the game |  |  |
| 8 | Yellow circles should be shown on the maze, when treasure/gold needs to be collected |  |  |
| 9 | Red circles should be shown on the maze, when threats are present |  |  |
| 10 | A green square should be shown on the maze, which indicates the finish point that the user has to go to. |  |  |
| 11 | There should be multiple threats within the maze, if touched then it should be game over. |  |  |
| 12 | A blue square represents the user playing the game. This should be able to move around the maze accordingly. |  |  |

- Choosing between platforms  
  
In order to choose the best platform to deploy the maze game for Olde Worlde Phunne, I will conduct my own research into the platforms highlighting the features, advantages and disadvantages of each and ultimately implementing the Maze Game on that platform. The platforms in which I will be looking into further are Desktop / Mobile / World Wide Web / Cloud.

### **World Wide Web**

The World Wide Web [WWW], commonly known as “The Web” invented by Tim Berners-Lee thought it was necessary to create an easier way to access information scattered across the internet. Today the web is defined as a collection of internet resources ranging from hyperlinked text, audio and video files that can be accessed and searched by browsers, [eg; Google Chrome] based on standards such as HTTP, which defines how messages are formatted/transmitted as well as what actions web servers and browsers should take in response to various commands and TCP/IP governs the connection between multiple computer systems to the internet.   
  
**Features of the World Wide Web**  
 - Hypertext 🡪 Text that is outputted to the user on a computer screen or an electronic device which is hyperlinked, meaning that once clicked on highlighted words or image it should automatically reference to other text that the user is able to immediately access/view more information.  
   
 - Cross-Platform 🡪 Makes the World Wide Web accessible to all, as a user is able to use it on different computers, as well as different software packages.   
  
 - Standards  
TCP/IP 🡪 Governs the connection between multiple computer systems to the internet.  
HTTP 🡪 Defines how messages are formatted/transmitted as well as what actions web servers and browsers should take in response to various commands  
  
 - HyperText Markup Language [HTML] 🡪 Is the most common language used to create documents on the World Wide Web. HTML uses hundreds of different tags to define a layout for web pages.

**Considerations to take when developing on the World Wide Web:**

The website of the newly developed software should be accessible from a wide range of devices available so that as many users possible can be able to use the system without any need to change their main device

The newly developed software should follow web standards, including HTML for presenting content on the World Wide Web.

The compression of the content from the software should also be considered. You do not want your users to have to wait for a long time for a page to load - which causes inefficiency and loss of productivity

The servers where the software will be hosted on will need to be ensured that it is secured. There is always a risk of vulnerability for anything on the internet.

#### Advantages and Disadvantages of the World Wide Web:

|  |  |  |
| --- | --- | --- |
| **Platform** | **Advantages** | **Disadvantages** |
| **Web** | **Worldwide access**  Employees are able to work from different offices, sites or from home.  **Always up-to-date**  Accessing through a web platform would ensure that users are accessing the same URL [Unique Resource Locator], meaning that all employees will be able to see the most up-to-date version of the software.   **Easy to set up/use**  A new user needs to be provided with the URL, username and password in order to use a web platform  **No need to install** Web-based applications do not require any installation process. Moreover, they don’t occupy space on the hard drive  **No upgrades needed** Applications on the web do not require upgrading, as users always get the latest version whenever the web application is launched.  **Access from Mobile Device** A web application is adaptable through a mobile as web apps can be accessed from a mobile device without any issues. This makes employees able to sign in and monitor the software from their phones, since more and more people are using smartphones for their daily routines. | **Internet Reliance** Requires internet connectivity at all times to access the website.   **Security**  Data is less secure when on the web, as users from all over the world are accessing the same server hosted by a third party. However there are ways to strengthen security such as encryption, etc.  **Reduced Speed** Likely to be a bit slower than an application hosted on a company’s server, since the web platform is worldwide access.   **Different Browsers** As different people use various browsers; the web platform must be supported across all browsers and all screen sizes.   **Performance** Transmitting data via the internet rather than on a desktop changes the performance on how quick the tasks take. As the internet depends on the connection, users can be annoyed and easily irritated by the slower performance. |

#### **Desktop**

A desktop platform consists of an application that runs stand-alone on a computer. The primary desktop platforms are x86-based computers running Windows and Mac.   
 **Features of a Desktop Platform:**

- Type of Processor [CPU] 🡪 A processor responds to and processes the basic instructions that drive a computer; the four main tasks it performs are fetch, decode, execute and writeback.   
  
 - Operating System 🡪 It manages all of the software and hardware of the computer, as well as the computer’s memory and processes. To add to this, it allows a regular user to communicate with the computer, without knowing the computer language.  
  
 - Random Access Memory [RAM] 🡪 Hardware devise that allows information to be stored and retrieved on a computer. As information is accessed randomly rather than sequentially, like on a Hard drive or CD, the desktop can access the data much faster.

**Considerations to take when developing on desktops:**

The look and feel of the user interface should be aimed to be consistent across all operating systems

The operating systems that the software will be developed for should also be considered, including major platforms such as Windows, Linux, Mac OS

The software developed should be as efficient as possible including - low memory footprint, optimised for specific architecture and processor efficient.

Ease for troubleshooting: As well as the software being easy for the user to navigate around, if any problems occur, it should be easy for the user to be able to report problems to the developer when problems arise.

#### Advantages and Disadvantages of Desktop:

|  |  |  |
| --- | --- | --- |
| **Platform** | **Advantages** | **Disadvantages** |
| **Desktop** | **Security Enhanced**  Since the user is keeping their data on their own computer systems, [as the application is running off the user’s desktop], this makes it harder for hackers to gain access to the data, meaning that data is more secure when compared to the web platform.  **No Internet Reliance Required** Doesn’t require internet connectivity to access the application.  **Performance** Typically one would find that a well-written desktop software running on a top-notch computer runs faster than a web-based application  **Integration with other products** It is much easier to integrate products within a desktop application rather than a browser. | **Lack of Portability** Desktop applications are not portable and restricted to one location, which is the computer. The desktop application will need to be installed   **Installation is necessary**  Desktop applications requires to be installed on the desktop, as well as some place on the hard drive. The installation process is quick, however can easily irritate a user who is needing to access the application immediately |

### **Mobile**

A mobile platform in the context of software development has 3 types of mobile applications that can be developed.

1) Native – These are apps which have been developed for a specific operating system

2) WebApp – Where an application is accessible online and is optimized for mobile use

3 Multi-platform – where applications work across different mobile devices and desktops (discussed in another section)

**Considerations to take when developing on mobile devices:**

Considerations are quite similar to the ones when developing for a desktop operating system.

The mobile operating system to develop for should be considered, especially major operating systems such as iOS, android

The user interface of the application developed should aim to be consistent across all operating systems that the application is developed for

The memory foot print of the application should be low and should no cause the processor of the device to be strained

The user interface of the device should be user friendly and easy to troubleshoot problems when they occur.

The use of colours should also be considered to notify users of any action that they take eg red for error, green for good. It should also consider that some users will be unable to distinguish certain colours.

#### Advantages and Disadvantages of Mobile:

|  |  |  |
| --- | --- | --- |
| Platform | Advantages | Disadvantages |
| Mobile  (native) | Fully optimised for the operating system that it is developed on  Full access to the device for software and hardware resources  Integration with peripheral devices if needed e.g smartwatches, camera | Each operating system has its own specificsed programming language and skill set required to program applications  There are high costs and lengthy development periods  If done natively it will become hard for developers to keep versions of an application the same on different platforms |
| Mobile  (WebApp) | Easier for developers to maintain, just one set of programming code needed  No need for individual installation on user devices, saves storage space  Each user has the same version of the application, accessible online  Lower cost of development | Internet connection is required  There is no access to device hardware, e.g fingerprint reader – it may be a security flaw if dealing with sensitive data  Not all access to devices hardware and software resources – may limit the performance of the application |

### Recommendation: Desktop

# Implementation

## - Best practices for code development

Throughout the development of my code, I have followed and aimed to maintain the following practices for the project:

**Commenting**

Within my code, I have added comments in a consistent manner and I have always defined what each line does and if there are any libraries involved in making the code work

**Indentation of code**

In the code that I have developed, the indentation of the code is always kept consistent and followed to prevent any errors before running the code

**Understandable variable names**

I always ensure that any variables that I create can be understood by someone who has not read my code before

**Version control**

Whenever I make a significant change to parts of my code, I ensure to save them under a new version so that the code can either be compared to the previous version or rolled back in case anything goes wrong

**Simple code**

I always aim to keep the code to be as simple and minimal as possible, avoiding unnecessary lines if it can be done

**Test driven development**

After designing each version, I test what functionality is there and what is not there, and as a result it leads to the next version of the code if something is missing. It also allows me to see how a version of the code would really run for the end user

**File and folder organisation**

I always keep my files and folders organised to ensure that I can achieve version control and keep track of where everything is

Testing   
Test plan (Before carrying out testing)

Before testing the code that I made, I created the following test plan to test my game program.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | What to do | Given Input | Expected Output | Actual Output |
| 1 | Run the program and wait for the welcome message | N/A | “Welcome to the Maze Game” |  |
| 2 | Run the program and wait for it to ask the user to input their name | N/A | “What is your name” |  |
| 3 | Run the program, input a name and wait for a second greeting message and instructions to appear | Jason | “Nice to meet you Jason” …. |  |
| 4 | Run the program and let it randomly choose a number between 1-10 to open a text file | B/A | “OPENED file x” |  |
| 5 | Run the program and make sure a game starts only when the maze has been configured properly | N/A | Completed maze |  |
| 6 | When starting to play the Maze game, the player should be randomly allocated to an empty space | N/A | Player allocated to an empty space |  |
| 7 | When the game starts, the player should begin with 0 wealth | N/A | Instructions at the beginning |  |
| 8 | When the Maze is shown, all rooms should be able to be seen and passages identified | N/A | Completed maze |  |
| 9 | Run program and collect the treasure | User should touch the treasure [yellow circle] | The treasure should disappear when the player goes over it and 10 wealth points get added |  |
| 10 | Run program and destroy a threat | User should go near the threat [red circle] and press SPACEBAR on the keyboard to destroy it. | The threat should disappear when the player destroys it and 5 wealth points get added |  |
| 11 | All threats and treasures should be removed once destroyed/collected before finishing | Going to the end point [green square] | Congratulations message should be displayed or if not finished then attempted finish message displayed |  |
| 12 | When collecting treasure, user should gain 10 wealth points | Going over a treasure | User gets 10 wealth added to their current wealth points |  |
| 13 | When destroying a threat, user should gain 5 wealth points | Pressing space bar when near a threat | User gets 5 wealth points added to their current wealth points |  |
| 14 | When user touches a threat, game over | User going over a threat which is a red circle | Game over message should be displayed to user |  |
| 15 | When finishing the game too early, user should get deducted 10 wealth points from total and launched back in their starting position | User going to endpoint while there are still threats / treasures that have not been collected | Game not finished yet message should be displayed to the user |  |
| 16 | When the game finishes, the total wealth points and time taken to complete the Maze is calculated and outputted on GUI | User going over the end point once all treasures and threats have been collected / destroyed | Congratulations message along with their score and time taken to complete the game |  |

Non-Functional Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | What to do | Given Input | Expected Output | Actual Output |
| 1 | The game must be easy to use and play | Running the program | Welcome message with clear instructions and easy controls to move the player | ? |
| 2 | The program should look into the chosen text file and copy template of maze within 10 seconds | Randomly selected file number | Beginning of maze printed |  |
| 3 | The game must be reliable and if unable to process the request then appropriate error message should show | ? | ? | ? |
| 4 | White squares within the maze should show where the walls are and therefore the player shouldn’t be allowed in that area | Player attempting to go into a wall (white square) | Player can’t go into wall (white square) |  |
| 6 | Yellow circles should be shown on the maze, when treasure/gold are located | User starting the program and the level containing the configuration of the treasure locations | Yellow circles on the level the is loaded and placed in their configured locations |  |
| 7 | Red circles should be shown on the maze, when threats are present | User starting the program and the level containing the configuration of the threat locations | Red circles on the level the is loaded and placed in their configured locations |  |
| 8 | A green square should be shown on the maze, which indicates the finish point that the user has to go to. | User starting the program and the level containing the configuration of the end point location | Green circles on the level the is loaded and placed in their configured locations |  |
| 9 | A blue square represents the user playing the game. This should be able to move around the maze accordingly. | User starting the program and the program randomly allocating the player to an empty space | User being placed in a randomly assigned location that is empty |  |

Test plan (After carrying out testing)

# 

# Documentation

# - Minimum Hardware and Software requirements Python – Minimum Requirements for Python

As the Maze game runs on python, anyone whom wishes to edit it is encouraged to make sure that they are able to run python. As stated on the python website, below are the minimum requirements needed to run python.

Processors: Intel Atom® processor or Intel® Core™ i3 processor

Disk space: 1 GB

Operating systems: Windows\* 7 or later, macOS, and Linux

Python\* versions: 2.7.X, 3.6.X

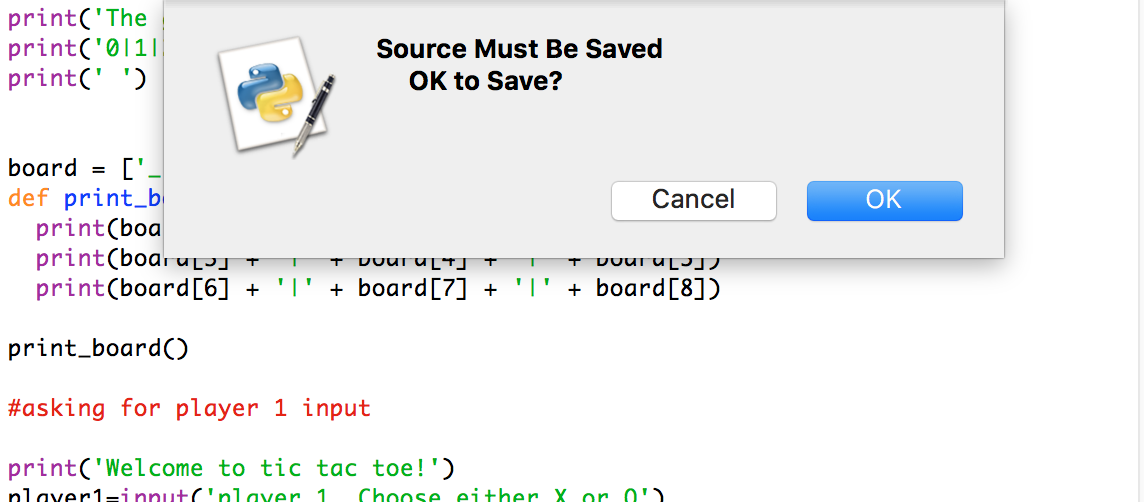
Included development tools: conda\*, conda-env, Jupyter Notebook\* (IPython)

Compatible tools: Microsoft Visual Studio\*, PyCharm\*

Included Python packages: NumPy, SciPy, scikit-learn\*, pandas, Matplotlib, Numba\*, Intel® Threading Building Blocks, pyDAAL, Jupyter, mpi4py, PIP\*, and other

# **User Guide**

  
How to prepare to play the Maze game?  
  
Make sure that you have downloaded the Maze.py to your desktop computer, as well as all of the txt files so that the program is able to retrieve the maze layout.   
  
In order to prepare this game or open this game in Python, follow the steps:  
• From desktop or start menu, select python 3.6 (or IDLE Python 3.6 32- bit)   
• Click the File menu and Open option and select Maze.py file from your chosen directory/folder. Make sure that you have all the txt documents within the same folder.   
• Now you can see source code of the game program.  
  
How to run the Maze game?  
  
When your game program’s source code is available on the screen, press F5 from keyboard or select Run Module/F5 from Run menu to run the game program.



Python may ask you that the source must be saved. Click OK and the program will be opened

How to play the Maze game?