**ACKNOWLEDGEMENT**

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking , behavior and acts during the course of study.

We express our sincere gratitude to ***Prof Name*** worthy Principal for providing me an opportunity to undergo Major Project as the part of the curriculum.

We also extend our sincere appreciation to ***Prof. name*** who provided his valuable suggestions and precious time in accomplishing our Industrial Training Report .

Lastly, We would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to day experience and received lots of suggestions that improves our quality of work.

**<name>**

**CANDIDATE’S DECLARATION**

We, **<name>,** Roll No. <Rollno>B.Tech (Semester- VII) of the Vyas Inst./college of engineering & technology, jodhpur hereby declare that the Training Report entitled **“Flappy Bird Game”** is an original work and data provided in the study is authentic to the best of our knowledge.This report has not been submitted to any other Institute for the award of any other degree.

**Sign. of Student**

**Date:**

Place:

**ABSTRACT**

Flappy Bird Game is well known game among children even among matured people.

The rules and regulation of the game are as well-known as the game. The case study meant for implementing this game without losing its interesting and attraction.

The user can interact with the game using keyboard.

The points increases according to increases the level, if we the bird touch any pipe or any things then the Game is Over.

The purpose of the game is to fly and reach to final level without any attachment with any things. It has obstacles in between the journey, which the gamer will have to overcome or being overthrown by the opponent.

The game explore the classic game with a new twist

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TOPIC** |  |
| 01 | An Introduction of Python |  |
| 02 | What is Flappy Bird Gmae |  |
| 03 | FLOW diagram |  |
| 04 | Screenshots |  |
| 05 | Testing |  |
| 06 | Conclusion |  |
| 07 | Reference |  |

1. **An Introduction to Python**
2. A Brief History Of Python
3. Python Versions
4. Installing Python
5. Executing Python from the command Line
6. **History of Python**

[Python](https://www.geeksforgeeks.org/python-programming-language/) is a widely used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

**Let’s dig deeper–**

In the late 1980s, history was about to be written. It was that time when working on Python started. Soon after that, Guido Van Rossum began doing its application based work in December of 1989 by at Centrum Wiskunde & Informatica (CWI) which is situated in Netherland. It was started firstly as a hobby project because he was looking for an interesting project to keep him occupied during Christmas. The programming language which Python is said to have succeeded is ABC Programming Language, which had the interfacing with the Amoeba Operating System and had the feature of exception handling. He had already helped to create ABC earlier in his career and he had seen some issues with ABC but liked most of the features. After that what he did as really very clever. He had taken the syntax of ABC, and some of its good features. It came with a lot of complaints too, so he fixed those issues completely and had created a good scripting language which had removed all the flaws. The inspiration for the name came from BBC’s TV Show – ‘Monty Python’s Flying Circus’, as he was a big fan of the TV show and also he wanted a short, unique and slightly mysterious name for his invention and hence he named it Python! He was the “Benevolent dictator for life” (BDFL) until he stepped down from the position as the leader on 12th July 2018. For quite some time he used to work for Google, but currently, he is working at Dropbox.  
The language was finally released in 1991. When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & C. Its design philosophy was quite good too. Its main objective is to provide code readability and advanced developer productivity. When it was released it had more than enough capability to provide classes with inheritance, several core data types exception handling and functions.

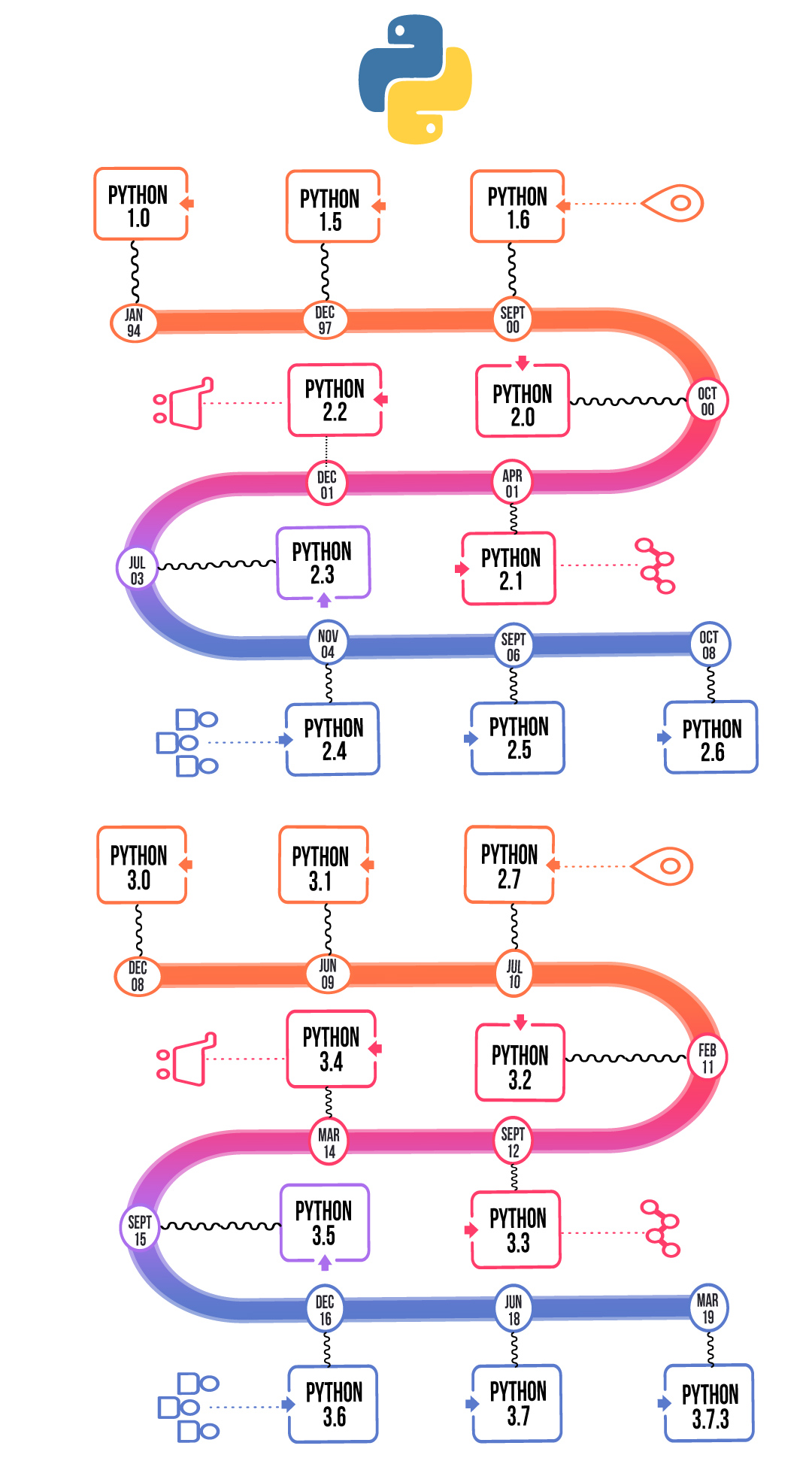
1. **Python 3.9 is the latest version**

The two of the most used versions has to Python 2.x & 3.x. There is a lot of competition between the two and both of them seem to have quite a number of different fanbase.

For various purposes such as developing, scripting, generation and software testing, this language is utilized. Due to its elegance and simplicity, top technology organizations like Dropbox, Google, Quora, Mozilla, Hewlett-Packard, Qualcomm, IBM, and Cisco have implemented Python.

Python has come a long way to become the most popular coding language in the world. Python has just turned 30, but it still has that unknown charm & X factor which can be clearly seen from the fact that Google users have consistently searched for Python much more than they have searched for Kim Kardashian, Donald Trump, Tom Cruise etc.

Python has been an inspiration for many other coding languages such as Ruby, Cobra, Boo, Coffee Script ECMAScript, Groovy, Swift Go, OCaml, Julia etc.



1. **Installing Python**

Installing Python is a simple matter, regardless of which platform you’re using.

The first step is to obtain a recent distribution for your machine; the most recent

One can always be found at www.python.org.

Having more than one version of Python

You may already have an earlier version of Python installed on your machine. Many Linux distributions and Mac OS X come with Python 2.x as part of the operating system.

Because Python 3 isn’t completely compatible with Python 2, it’s reasonable to wonder if installing both versions on the same computer will cause a conflict.

There’s no need to worry; you can have multiple versions of Python on the same computer.

In the case of UNIX-based systems like OS X and Linux, Python 3 installs alongside the older version and doesn’t replace it. When your system looks for “python,” it still finds the one it expects; and when you want to access Python 3, you can run python3.6.8 or idle. In Windows, the different versions are installed in separate locations and have separate menu entries.

Some basic platform-specific descriptions for the Python installation are given next. The specifics can vary quite a bit depending on your platform, so be sure to read the instructions on the download pages and with the various versions. You’re probably familiar with installing software on your particular machine, so I’ll keep these descriptions

short:

* **Microsoft Windows**—Python can be installed in most versions of Windows by using the Python installer program, currently called python-3.1.msi. Just download it, execute it, and follow the installer’s prompts. You may need to be

logged in as administrator to run the install. If you’re on a network and don’t have the administrator password, ask your system administrator to do the installation

for you.

* **Macintosh**—You need to get a version of Python 3 that matches your OS X version

and your processor. After you determine the correct version, download the disk image file, double-click to mount it, and run the installer inside. The OSX

installer sets up everything automatically, and Python 3 will be in a subfolder inside the Applications folder, labeled with the version number. Mac OS X ships with various versions of Python as part of the system, but you don’t need to worry about that—Python 3 will be installed in addition to the system version.

You can find more information about using Python on OS X by following the

links on the Python home page.

* **Linux/UNIX**—Most Linux distributions come with Python installed. But the versions of Python vary, and the version of Python installed may not be version 3; for this book, you need to be sure you have the Python 3 packages installed. It’s also possible that IDLE isn’t installed by default, and you’ll need to install that package separately. Although it’s also

1. **Executing Python from the command Line**

**Running Python on your OS**

## Run the Python command-line interpreter, under your OS of choice,

Windows

* Open Command line:   Start menu -> Run  and type cmd
* Type:   C:\python27\python.exe
* Note: This is the default path for Python 2.7. If you are using a computer where Python is not installed in this path, change the path accordingly.

**IDLE and the basic interactive mode**

You have two built-in options for obtaining interactive access to the Python interpreter: the original basic (command-line) mode and IDLE. IDLE is available on many platforms, including Windows, Mac, and Linux, but it may not be available on others.

You may need to do more work and install additional software packages to get IDLE running, but it will be worth it because it’s a large step up from the basic interactive mode. On the other hand, even if you normally use IDLE, at times you’ll likely want to fire up the basic mode. You should be familiar enough to start and use either one.

The basic interactive mode is a rather primitive environment. But the interactive

examples in this book are generally small; and later in this book, you’ll learn how to easily bring code you’ve placed in a file into your session (using the module mechanism).

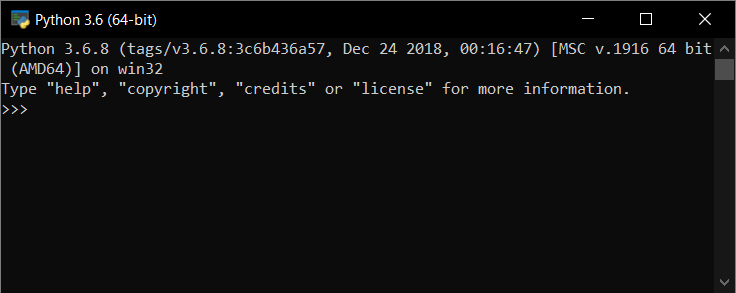
Let’s look at how to start a basic session on Windows, Mac OS X, and UNIX:

* Starting a basic session on Windows—For version 3.x of Python, you navigate to the Python (command-line) entry on the Python 3.x submenu of the Programs

folder on the Start menu, and click it. Alternatively, you can directly find the Python.exe executable (for example, in C:\Python36) and double-click it.

Doing so brings up the window

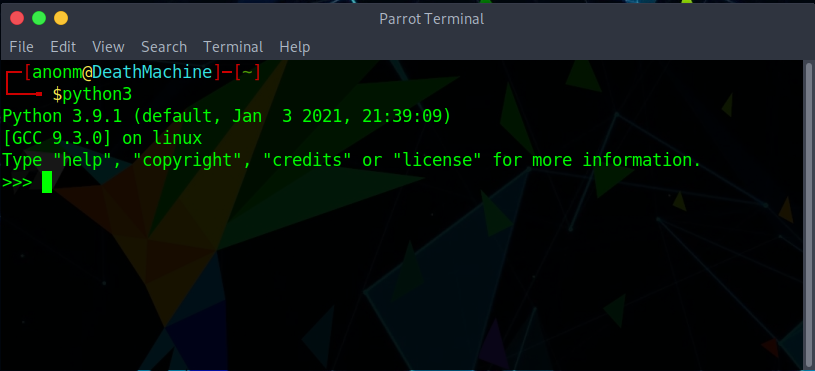
* Starting a basic session on Mac OS X—Open a terminal window, and type python3. If you get a “Command not found” error, run the Update Shell Profile. Command script found in the Python3 subfolder in the Applications folder.
* Starting a basic session on UNIX—Type python3.6.8 at a command prompt. A version message similar to the one followed by the Python prompt >>> appears in the current window.



Basic interactive mode on Windows 10

* Starting IDLE on Mac OS X—Navigate to the Python 3.x subfolder in the Applications folder, and run IDLE from there.
* Starting IDLE on Linux or UNIX—Type idle at a command prompt. This

brings up a window similar to the one If you installed IDLE through your distribution’s package manager, there should also be a menu entry for IDLE under the Programming submenu or something similar.



Choosing between basic interactive mode and IDLE

Which should you use, IDLE or the basic shell window? To begin, use either IDLE or the Python Shell window. Both have all you need to work through the code examples in this book until you reach chapter 10. From there, we’ll cover writing your own modules, and IDLE will be a convenient way to create and edit files. But if you have a strong preference for another editor, you may find that a basic shell window and your favorite editor serve you just as well. If you don’t have any strong editor preferences, I suggest using IDLE from the beginning.

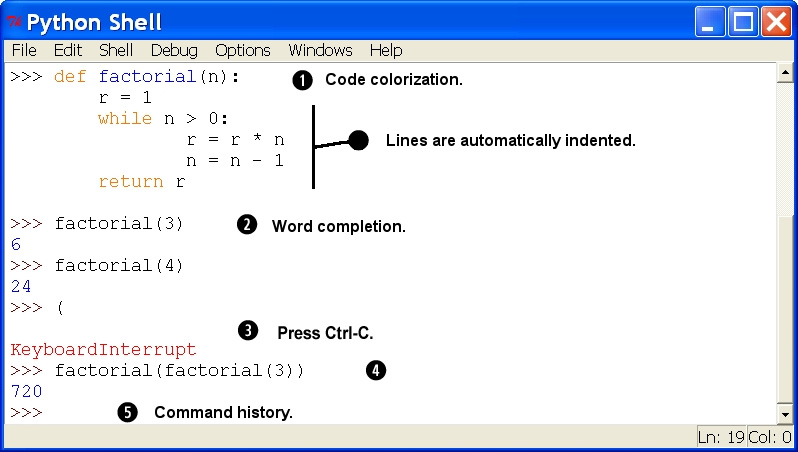
**Using IDLE’s Python Shell window**

The Python Shell window opens when you fire up IDLE. It provides automatic

Indentation and colors your code as you type it in, based on Python syntax types.

You can move around the buffer using the mouse, the arrow keys, the Page Up and Page Down keys, and/or a number of the standard Emacs key bindings. Check the Help menu for the details.

Everything in your session is buffered. You can scroll or search up, place the cursor on any line, and press Enter (creating a hard return), and that line will be copied to the bottom of the screen, where you can edit it and then send it to the interpreter by pressing the Enter key again. Or, leaving the cursor at the bottom, you can toggle up and down through the previously entered commands using Alt-P and Alt-N. This will successively bring copies of the lines to the bottom. When you have the one you want, you can again edit it and then send it to the interpreter by pressing the Enter key. You can complete Python keywords or user-defined values by pressing Alt-/.



1. **What is Flappy Bird Game**

* In this project, we design and implement a Flappy Bird like video game on the SoCKit development board. Flappy Bird is a very popular mobile game on Android platform, driving a lot of people crazy. In this game, the player can control the vertical movement of bird ( every pressing on the keyboard makes the bird leap upward for a little bit, and the bird will fall freely without control ). As soon as the game begins, tubes will keep appearing from the right side of the screen and moving leftwards. (so that it seems like the bird flying forward). The goal of this game is to control the bird, dodging and passing the incoming tubes, as many as possible. The game is endless until the bird eventually hit one of the tubes, ground, or ceiling. Figure 1 is the start screen of Flappy Bird. The title "Flappy Bird" is shown in the middle of the uppers side of the screen. The bird is also displayed on the background. **Features**

This project is the replica of Flappy Bird Game with some changes and some of the features of this project are:

* + GUI interface
  + Keyboard interaction
* **Objectives**

We made this game or project so that we could give our best in possible ways and show what we learned. The objectives of this project are:

* + To give a GUI interface to the classic game.
  + To make it user friendly.
  + To provide an easy interface.
  + To entertain people in their leisure time.

**Pygame :**

Pygame is a set of Python modules designed for writing video games. Pygame adds functionality on top of the excellent SDL library. This allows you to create fully featured games and multimedia programs in the python language.

Pygame is highly portable and runs on nearly every platform and operating system.

Pygame is free. Released under the LGPL licence, you can create open source, freeware, shareware, and commercial games with it. See the licence for full details.

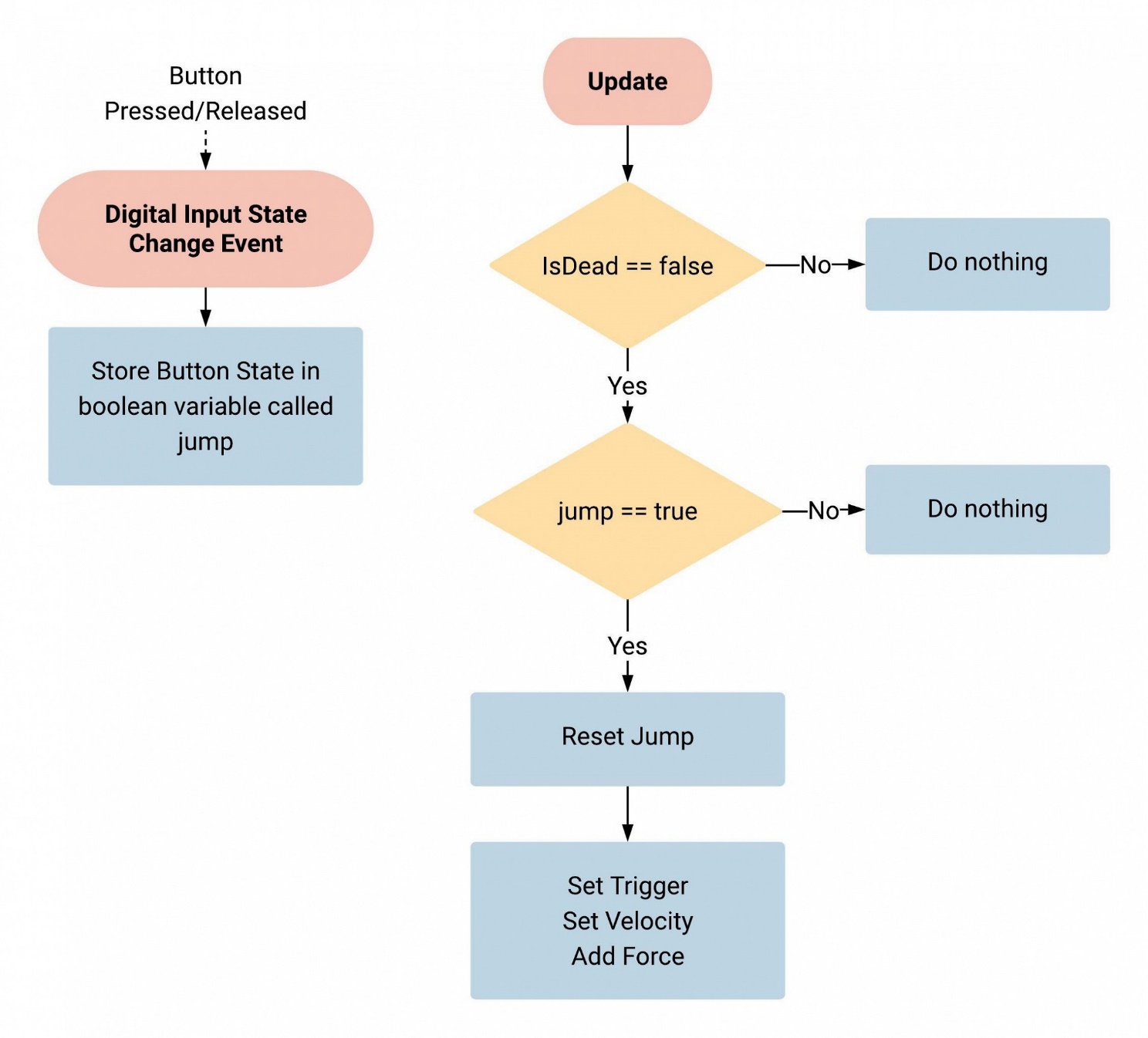
Truly portable. Supports Linux, MacOS, FreeBSD, NetBSD, OpenBSD, BSD/OS, Solaris, IRIX, and QNX. The code contains support for AmigaOS, Dreamcast, Atari, AIX, OSF/Tru64, RISC OS, SymbianOS and OS/2, but these are not officially supported. You can use it on hand held devices, game consoles and the One Laptop Per Child (OLPC) computer.

It's Simple and easy to use. Kids and adults make shooter games with pygame. Pygame is used in the OLPC project and has been taught in essay courses to young kids and college students. It's also used by people who first programmed in z80 assembler or c64 basic.

You control your main loop. You call pygame functions, they don't call your functions. This gives you greater control when using other libraries, and for different types of programs.

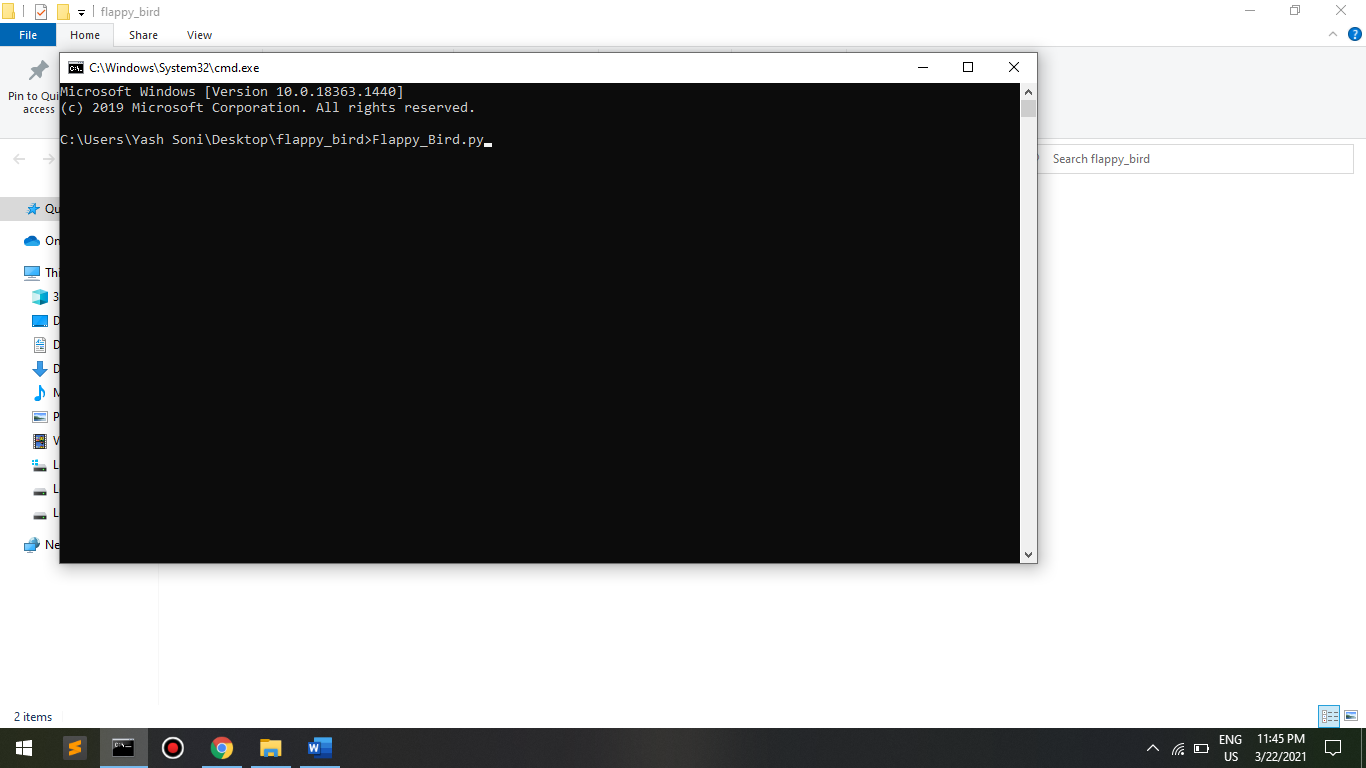
Does not require a GUI to use all functions. You can use pygame from a command line if you want to use it just to process images, get joystick input, or play sounds.

1. **FLOW Diagram**

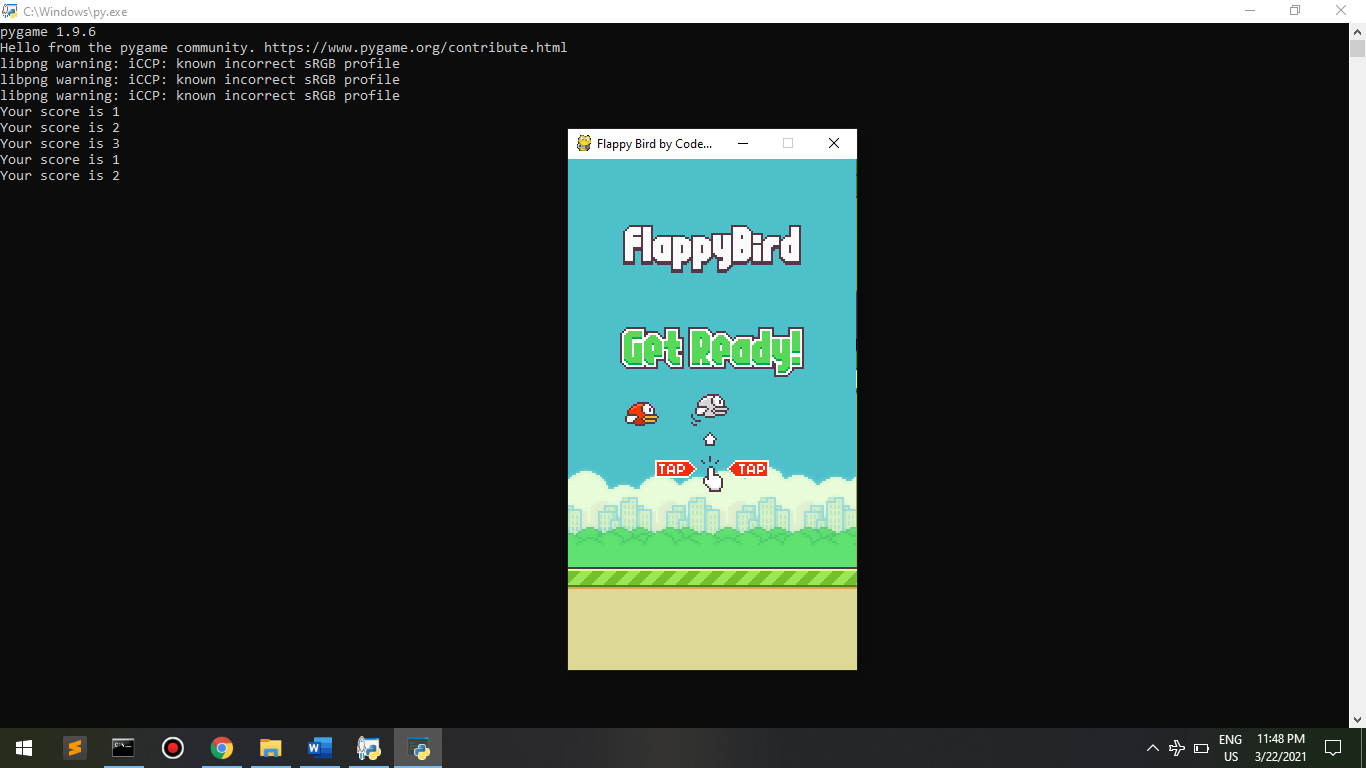


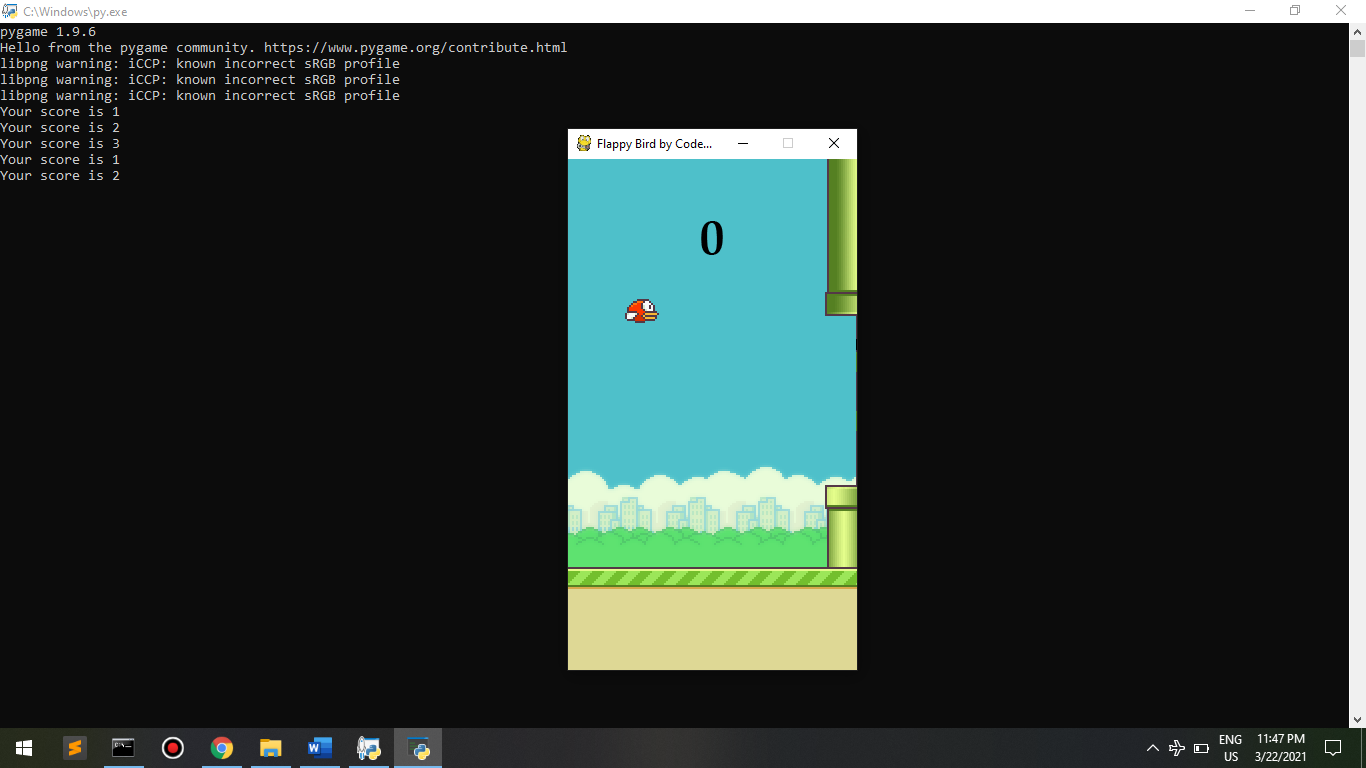
1. **ScreenShots:**

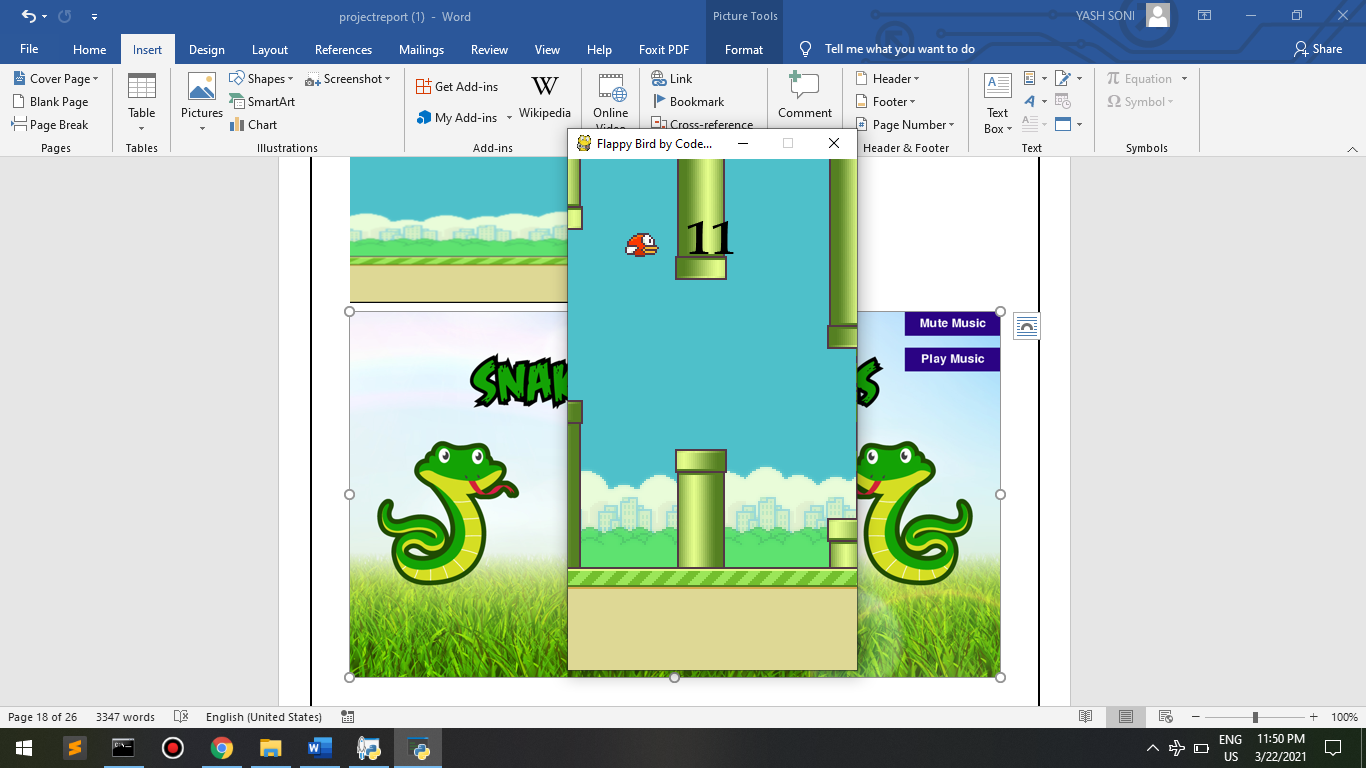
**Run:** To run the game open cmd and type “Flappy Bird Game.py” and hit enter.



Start: This is first screen you’ll see when game starts







1. **TESTING**

**Objective:**

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. We will also test the user friendliness of our app .The application will be used as an important tool, but we would like to ensure that it could be run on a variety of platforms with little impact on performance or usability.

* **Process Overview :**

The following represents the overall flow of the testing process:  Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.

* Identify which particular test(s) will be used to test each module.
* Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
* Identify the expected results for each test.
* Document the test case configuration, test data, and expected results.
* Perform the test(s).
* Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
* Successful unit testing is required before the unit is eligible for component integration/system testing.
* Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
* Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

**Testing Process:**

The Test Process approach that will be followed.

* Organize Project involves creating a System Test Plan, Schedule & Test Approach, and assigning responsibilities.
* Design/Build System Test involves identifying Test Cycles, Test Cases, Entrance & Exit Criteria, Expected Results, etc. In general, test conditions/expected results will be identified by the Test Team in conjunction with the Development Team. The Test Team will then identify Test Cases and the Data required. The Test conditions are derived from the Program Specifications Document.
* **Design/Build Test Procedures** includes setting up procedures such as Error Management systems and Status reporting.
* **Build Test Environment** includes requesting/building hardware, software and data setups.
* **Execute System Tests** identified in the Design/Build Test Procedures will be executed. All results will be documented and Bug Report Forms filled out and given to the Development Team as necessary.
* **Signoff** happens when all pre-defined exit criteria have been achieved.

1. **Conclusion**

A software project means a lot of experience. In this section we summarize the experience gained by project team during development of “Flappy Bird Game”.

**The Obstacles**

1. Working with game library completely a new experience for us. Normally we are working with different OO languages, DBMS, mark up languages etc.

2. We adopt these things by video tutorials, text tutorials, internet and learning materials given by the tools themselves. It's a matter of time, patience and hard work.

3. It is very sensible work and it demands much time because the game libraries try to connect game environment with the real world.

4. The Exists game libraries demands vast knowledge about its properties, sections and sub-sections.

**The Achievements**

1. Now we know much more about game libraries. How it works? The properties, objects and others.

2. We know how a model is constructed and how it is animated.

3. Growing creative thinking and imagination capability

**Future Plan**

1. Level Extension
2. Improve Graphical Representation
3. Introduce new game features
4. Introduce new environment and scenes

**Last Few Words**

I learned a lot through this project. This project has sharpened my concept of Game Libraries, animation and the software-hardware interface. We learned a lot about different documentation. The piece of software we developed is intended to serve the gamers of the world. The success of this project may give pleasure to billions of game lovers among the universe. This project not only tested my technical skills but also my temperament. There were times that i almost lost hope but I recovered through constant concentration and hard work.

1. **Reference**
   1. <https://www.pygame.org/>
   2. <https://www.geeksforgeeks.org>
   3. <https://devdocs.io/pygame/>