

**FIRE ALARM SYSTEM USING 8051
MICROCONTROLLER**

A

Semester Project

Research

BACHELOR OF SCIENCE

IN

ELECTRICAL AND COMPUTER ENGINEERING

COMPUTER STREAM

Submitted

By

YDENKAL TAMRAT (5279)

MESFIN MATHEWOS (0400)

ASHENAFI ABATE (0075)

YUSUF SEMAN (0752)

UNDER THE ESTEEMED GUIDANCE OF

Dr.R.Gopi Reddy



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

WACHEMO UNIVERSITY

3/August/2021

Abstract

Ping pong game is one of the most famous arcade games, simulating table tennis. Each player controls a paddle in the game by dragging it vertically across the screen's left or right side. Players use their paddles to strike back and forth on the ball. Turtle is an inbuilt graphic module in Python. It uses a panel and pen to depict illustrations. We provide a more interesting and convenient way for people to play ping pong game on pc. In this paper, we propose an Augmented Reality game system (ARPP) for two players on the same pc.

The ping pong game is normally created by using python programming language and a pycharm to write all the codes. The experiment results show that the proposed game system can work effectively and provide winner results on the computer devices.

Declaration

We hereby declare that the work presented in this project is the outcome of the investigation performed by us under the supervisor of **DR.R.GOPI REDDY**, Assistant Professor for Department of Electrical & computer Engineering, in wachemo university. We clearly declare that no part of this project has been submitted else-where for the award of any degree or diploma and is solely edited to us.

Countersigned**ADVISOR**

(Dr.R.Gopi Reddy)

Signature

(Ydenkal Tamrat)

(Mesfin Mathewos)

(Yusuf Seman)

(Ashenafi Abate)

Acknowledgment

First of all we would like to thanks our god for everything. Today we are successful in completing our Project work with such ease because He gives us the ability, chance and co-operating supervisor. We would like to thanks our advisor **Dr.R.GOPI REDDY** for his help us to made this project successfully and for his valuable advice. Finally we are grateful to our family who are always with us in every step of our life.

TABLE OF CONTENT

Abstract	I
Declaration	II
Acknowledgment	III
List of figures	VI
keywords	VII
CHAPTER ONE	1
1. INTRODUCTION	1
1.1 Background	1
1.2 Significance of the project.....	2
1.3 Statement of the problem	2
1.4 Objective	2
1.4.1 Main Objective	2
1.4.2 Specific Objective	2
1.5 Relevance	3
1.6 Goal of the project	3
1.7 Scope of the project	3
CHAPTER TWO	4
2. LITERATURE REVIEW	4
2.1 Review on ping pong game	4
2.2 Review on python	4
2.3 Review on pycharm	5
CHAPTER THREE	6
3. METHODOLOGY	6
3.1 Basic overview	6

3.2 Software requirements	7
3.2.1 python programming language	7
3.2.2 Pycharm community	9
CHAPTER FOUR.....	11
4. SYSTEM DESIGN	11
4.1 Introduction to the system.....	11
4.2 System design	11
4.3 System architecture	13
CHAPTER FIVE.....	14
5. Result and discussion.....	14
5.1 Result and discussion	14
CHAPTER SIX.....	18
6. Conclusion And Recommendation	18
6.1 Conclusion	18
6.2 Recommendation	23
References	19
Appendix	21

List Of Figures

<u>Figure 3.1</u> Steps for completion of the project	6
<u>Figure 4.1</u> Architectural design of the system	13
<u>Figure 5.1</u> Output result on python	25
<u>Figure 5.2</u> Output of the system	26
<u>Figure 5.3</u> Image for upated score	27
<u>Figure 5.3</u> Screen display when the game ends	27

KEYWORDS

Ping pong: very attractive game that is played like a tennis.

VGA: video graphics array connector.

VHDL: its programming language.

CHAPTER - 1

INTRODUCTION

1.1 Background

Ping pang game is multiplayers game build on turtle module. Where two persons can play this game. You can simply run the file to play the game by downloading all the files that are necessary to the game. You need to download music effects also to get good experience with the game. Ping-Pong is a very popular game. People need to play it in some fixed physical locations and by connecting it with Wi-Fi. We provide a more interesting and convenient way for people to play Ping-Pong game on pc. In this paper, we propose an Augmented Reality game system (ARPP) for two players on the same pc where two persons can play this game. you can simply run the file to play the game by downloading all the files that are necessary to the game. You need to download music effects also to get good experience with the game.

The goal of the project was to create an active ping pong game which is controlled using the VGA board. To clearly understand how to do such a task one must understand what is required to undergo it. The method to interact with the machine is through a readable language which is VHDL. The VHDL language will act as an instruction set for the controls, display and movement for the ping pong program. The project overall purpose is to show how the applicability of the interface interaction is possible using the software. Each specified switch interacts with a role for a paddle. In this lab the paddles were used to move both up and down using the switches orientation. Since there are multiple switches, the ones used on either end were selected as the controls for the paddles. This for a practical purpose so that if 2 players were to play the switches would not interfere with the players movements. The display of the ping pong table was a set field which uses the pixel settings with settings for the H-sync, V-sync for the placement of the pixels. The placement of the pixels horizontally and vertically changes because of the motion ball so what occurs is that the pixels that were meant for the ball refresh and the balls values replace the current pixel value and sets the previous ball position back to the background value. To distinguish the parts of the field aparta color association is used which requires the use of the Red, Green and Blue control signals to represent the RGB light display. Any combination of these colors will cause images to be colored in as whatever is wished. The

movement of ping pong as well as any changes that occur when an action occurs is programmed in the VHDL code to occur as changes in the pixels for the moving object and any animation effect after. For this project at the end of every goal there was a signal that should

1.2 Significance of the project

An intense game of ping pang game stimulates mental alertness and concentration and develops mental acuity. Improving reflexes, Due to the fast-paced, short-distance nature of the sport, both gross and fine muscle movements are improved. It's easy on the joints.

1.3 Statement of the Problem

Ping Pong is one of the most famous arcade games, simulating table tennis. Each player controls a paddle in the game by dragging it vertically across the screen's left or right side. Players use their paddles to strike back and forth on the ball. Turtle is an inbuilt graphic module in Python. It uses a panel and pen to depict illustrations.

1.4 objective

1.4.1 Main objective

- The project overall objective is to show how the applicability of the interface interaction is possible using the software. The ping pong paddles are each controlled by the users using the. Each specified switch interacts with a role for a paddle. In this project the paddles were used to move both up and down using the switches orientation. Since there are multiple switches, the ones used on either end were selected as the controls for the paddles.
- Design of ping pong game with turtle module where two players are played together.

1.4.2 Specific objective

Players use the paddles to hit a ball back and forth. The goal is for each player to reach eleven points before the opponent; points are earned when one fails to return the ball to the other.

- To simulate the design results.

1.5 Goal of the project

The goal of the project was to create an active ping pong game which is controlled using the the keyboard. To clearly understand how to do such a task one must understand what is required to undergo it. The method to interact with the machine is through a readable language which is

python. The python language will act as an instruction set for the controls, display and movement for the ping pong program.

1.6 Relevance

Playing ping pong game has many relevance

- It boosts problem solving skills
- It increase creative output
- It improves moral and team building
- It reduces stress
- They improve our modes and etc...

1.7 Scope and limitation of the project

Scope

- The purpose of this project is to provide complete solution for effectively monitoring ping pong game that makes to play multiplayer on the same computer or pc.
- And when one player win or score 20 the screen will display that the game is end and that player is won.

Limitation

- The game is only played on one pc it's not connected via wifi with another pc
- It is not worked without running on python.

CHAPTER - 2

LITERATURE REVIEW

2.1 Literature review on ping pong game

The name "ping-pong" was in wide use before British manufacturer J. Jaques & Son Ltd trademarked it in 1901. The name "ping-pong" then came to describe the game played using the rather expensive Jaques's equipment, with other manufacturers calling it table tennis. A similar situation arose in the United States, where Jaques sold the rights to the "ping-pong" name to Parker Brothers. Parker Brothers then enforced its trademark for the term in the 1920s, making the various associations change their names to "table tennis" instead of the more common, but trademarked, term.[9]

The next major innovation was by James W. Gibb, a British enthusiast of table tennis, who discovered novelty celluloid balls on a trip to the US in 1901 and found them to be ideal for the game. This was followed by E.C. Goode who, in 1901, invented the modern version of the racket by fixing a sheet of pimples, or stippled, rubber to the wooden blade. Table tennis was growing in popularity by 1901 to the extent that tournaments were being organized, books being written on the subject,[7] and an unofficial world championship was held in 1902. In those early days, the scoring system was the same as in lawn tennis.[10]

Although both a "Table Tennis Association" and a "Ping Pong Association" existed by 1910,[10] a new Table Tennis Association was founded in 1921, and renamed the English Table Tennis Association in 1926.[11] The International Table Tennis Federation (ITTF) followed in 1926.[1][12] London hosted the first official World Championships in 1926. In 1933, the United States Table Tennis Association, now called USA Table Tennis, was formed.[1][13]

2.2 Literature review on python

Python was conceived in the late 1980s[04] by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to ABC programming language, which was inspired by SETL,[16] capable of exception handling and interfacing with the Amoeba operating system.[11] Its implementation began in December 1989.[15] Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his

"permanent vacation" from his responsibilities as Python's Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a 5-member "Steering Council" to lead the project.[4] As of 2021, the current members of this council are Barry Warsaw, Brett Cannon, Carol Willing, Thomas Wouters, and Pablo Galindo Salgado.[20]

Python 2.0 was released on 16 October 2000, with many major new features, including a cycle-detecting garbage collector and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. [17] Many of its major features were back ported to Python 2.6.x[18] and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.[49]

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.[16] No more security patches or other improvements will be released for it.[19] With Python 2's end-of-life, only Python 3.6.x[14] and later are supported.

Python 3.9.2 and 3.8.8 were expedited [15] as all versions of Python (including 2.7) had security issues, leading to possible remote code execution and web cache poisoning.

2.3 Literature review on pycharm

PyCharm is an IDE developed by JetBrains since 2010. Yes, the same company that developed IntelliJ, the de-facto default for Java. It costs \$89 per year for private people which is pretty expensive if you just get your feet wet with Python. It's pretty cheap if you consider that the editor/IDE is one of the most important tools of a professional software developer[9].

It is used by many Python developers as one can see in the number of StackOverflow questions (12,455), in the 2019 JetBrains Survey, and the 2019 StackOverflow survey.

CHAPTER - 3

METHODOLOGY

3.1 Basic Overview

We use python programming language for designing our game and pycharm in order to write all the python codes.

For successful completion of this project some steps have been followed to carry out different tasks. Different literatures were revised relating to this project and data has been collected from nearby system. Based on the data collected system model have been designed. Finally using appropriate software, the programming has been developed.

- Generally, the steps are described by the following block diagram.

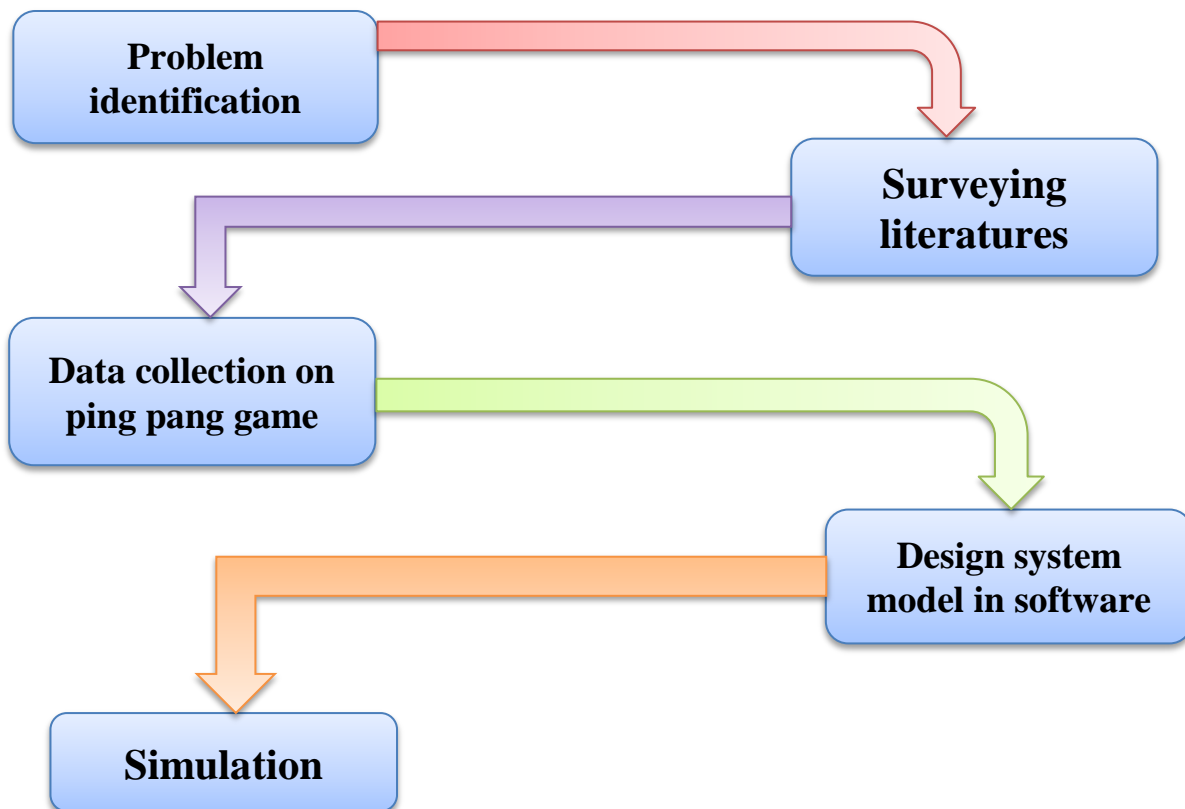


Fig 3.1 steps for completion of the project

3.2 SOFTWARE REQUIREMENTS

- We have used python programming language for simulation of our project and pycharm community for writing the code.

3.2.1 Python Programming Language

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Applications of Python

Python supports cross-platform operating systems which makes building applications with it all the more convenient. Some of the globally known applications such as [YouTube](#), BitTorrent, DropBox, etc. use Python to achieve their functionality.

- **Web development**

Python can be used to make web-applications at a rapid rate. There is common-backend logic that goes into making these frameworks and a number of libraries that can help integrate protocols such as HTTPS, FTP, SSL etc. and even help in the processing of JSON, XML, E-Mail and so much more.

- **Game Development**

Python is also used in the development of interactive games. There are libraries such as PySoy which is a 3D game engine supporting Python 3, PyGame which provides functionality and a library for game development.

- **Machine Learning and Artificial Intelligence:** are the talks of the town as they yield the most promising careers for the future. We make the computer learn based on past

experiences through the data stored or better yet, create algorithms which makes the computer learn by itself. The programming language that mostly everyone chooses? It's Python.

- **Desktop GUI:** We use Python to program desktop applications. It provides the Tkinter library that can be used to develop user interfaces. There are some other useful toolkits such as the wxWidgets, Kivy, PYQT that can be used to create applications on several platforms.
- **Web Scraping Applications:** Python is a savior when it comes to pull a large amount of data from websites which can then be helpful in various real-world processes such as price comparison, job listings, research and development and much more.

Features in Python

- Easy to code
- Free and Open Source
- Object-Oriented Language
- GUI Programming Support
- High-Level Language
- Extensible feature
- Python is Portable language
- Python is Integrated language
- Interpreted Language
- Large Standard Library
- Dynamically Typed Language

Modules of python

There are several modules in python we have discussed with some of them that we used in the development of our game.

- a) **Pygame:** is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.

Pygame uses the Simple DirectMedia Layer (SDL) library, with the intention of allowing real-time computer game development without the low-level mechanics of the C

programming language and its derivatives. This is based on the assumption that the most expensive functions inside games can be abstracted from the game logic, making it possible to use a high-level programming language, such as Python, to structure the game.

Other features that SDL doesn't have include vector math, collision detection, 2D sprite scene graph management, MIDI support, camera, pixel-array manipulation, transformations, filtering, advanced freetype font support, and drawing.

Applications using Pygame can run on Android phones and tablets with the use of Pygame Subset for Android (pgs4a). Sound, vibration, keyboard, and accelerometer are supported on Android.

Turtle: this module is built in module in python, meaning you do not have to install it. It is used to create basic 2D shapes and drawings and is extremely beginner friendly. without the low-level mechanics of the C programming language and its derivatives. This is based on the assumption that the most expensive functions inside games can be abstracted from the game logic, making it possible to use a high-level programming language, such as Python, to structure the game. Its used for building childs game most of the time.

- The main advantages of turtle are that it is extremely **simple** and makes it **very easy** to draw things to the screen.
- We have used turtle module to draw the objects on the screen

3.2.2 Pycharm Community

PyCharm is an integrated development environment used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

Features of PyCharm:

- **Intelligent Code Editor:** It helps us write high-quality codes. It consists of color schemes for keywords, classes, and functions. This helps increase the readability and understanding of the code. It helps identify errors easily.
- It provides the autocomplete feature and instructions for the completion of the code.

- **Code Navigation:** It helps developers in editing and enhancing the code with less effort and time.
- **Refactoring:** It has the advantage of making efficient and quick changes to both local and global variables.
- **Assistance for Many Other Web Technologies:** It helps developers create web applications in Python. supports popular web technologies such as HTML, CSS, and JavaScript
- **Support for Popular Python Web Frameworks:** PyCharm supports web frameworks such as Django. It provides the autocomplete feature and suggestions for the parameters of Django. It also assist web2py and Pyramid, the other popular web frameworks.
- **Assistance for Python Scientific Libraries:** supports Python's scientific libraries such as Matplotlib, NumPy, and Anaconda.

Advantage and disadvantage of Using PyCharm

PyCharm is a wonderful IDE for Python programmers and application developers. However, there are some pros and cons of using PyCharm. Let's see.

Advantage

- Installing PyCharm is very easy.
- Is an IDE that is easy to use.
- There are a lot of useful plugins and productive shortcuts in PyCharm.
- It integrates library and IDE features such as auto-completing and coloring.
- It allows viewing of the source code in a click.
- Software development is much faster using PyCharm.
- The feature of error spotlighting in the code further enhances the development process.
- The community of Python Developers is extremely large so that we can resolve our queries/doubts easily.

Disadvantage

- PyCharm is not free and its Professional version is pretty costly.
- It is not considered good for beginners due to its auto-complete feature.
- It may create trouble while fixing up tools such as venv.

CHAPTER - 4

SYSTEM DESIGN

4.1 Introduction to the System

- In this Chapter we are going to Explain about the system Design regarding to how we use the software to implement and got the final result of our project. In addition, the chapter elaborates the software stage by stage.

4.2 Software design

Steps for the software design of our project are as follows.

- **Download Python 3.9.6 button.**

The Python download requires about 25 Mb of disk space; keep it on your machine, in case you need to re-install Python. When installed, Python requires about an additional 90 Mb of disk space.

- **Installing python-3.9.6-amd64.exe.**

Ensure that both the Install launcher for all users (recommended) and the Add Python 3.9 to PATH checkboxes at the bottom are checked: typically only first is checked by default. and move the progress bar towards completion. Soon, a new Python 3.9.6 (64-bit) Setup pop-up window will appear with a Setup was successfully message.

- **Try to verify installation**

1. Navigate to the directory C:\Users\Pattis\AppData\Local\Programs\Python\Python39 (or to whatever directory Python was installed).
2. Double-click the icon/file python.exe. A pop-up window with the title C:\Users\Pattis\AppData\Local\Programs\Python\Python39\python.exe appears, and inside the window; on the first line is the text Python 3.9.6

Download pycharm installer .exe.

- To verify the integrity of the installer, use the SHA checksum linked from the Download page.
- Run the installer and follow the wizard steps.

- Mind the following options in the installation wizard
- 64-bit launcher: Adds a launching icon to the Desktop.
- Open Folder as Project: Adds an option to the folder context menu that will allow opening the selected directory as a PyCharm project.
- .py: Establishes an association with Python files to open them in PyCharm.
- Add launchers dir to the PATH: Allows running this PyCharm instance from the Console without specifying the path to it.
- To run PyCharm, find it in the Windows Start menu or use the desktop shortcut. You can also run the launcher batch script or executable in the installation directory under bin.
- There are very important module/package that are used to develop any game with python,

Finally After we have downloaded and install all the above softwares Below are the steps used to design the game:

Step 1) Create two paddles A and B on the left and right side of the screen.

Step 2) Create a ball.

Step 3) Create an event to move the paddle vertically on pressing a certain key.

Step 4) Create the function to update the score after each player misses a collision.

Step 5) Create the function that shows the end of the game.

4.3 System Architecture

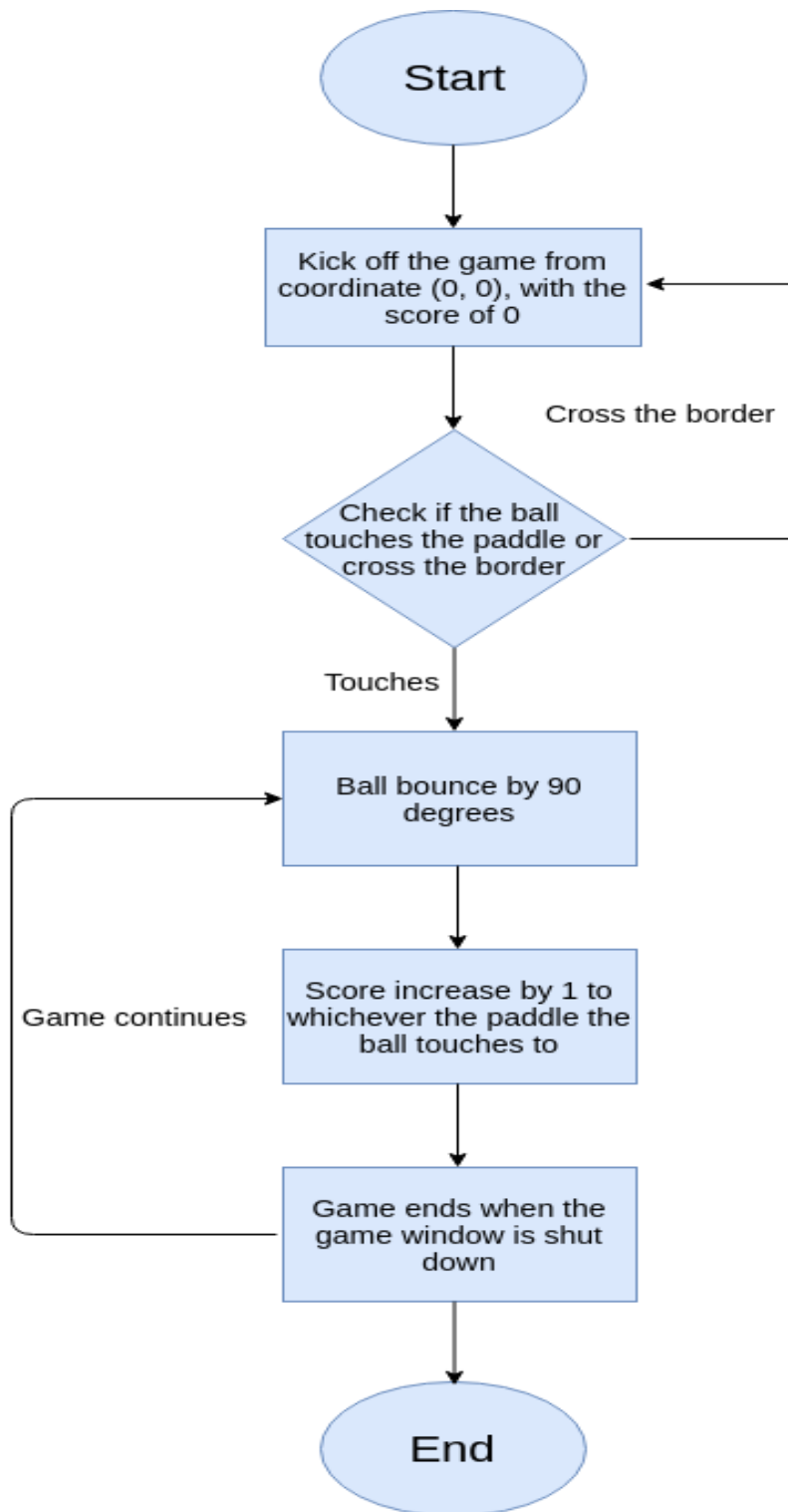


Fig 4.1 Architectural design of the system

CHAPTER - 5

RESULT AND DISCUSSION

5.1 Result and discussion

The aim of the project was to implement the design of the ping pong game with python turtle module.

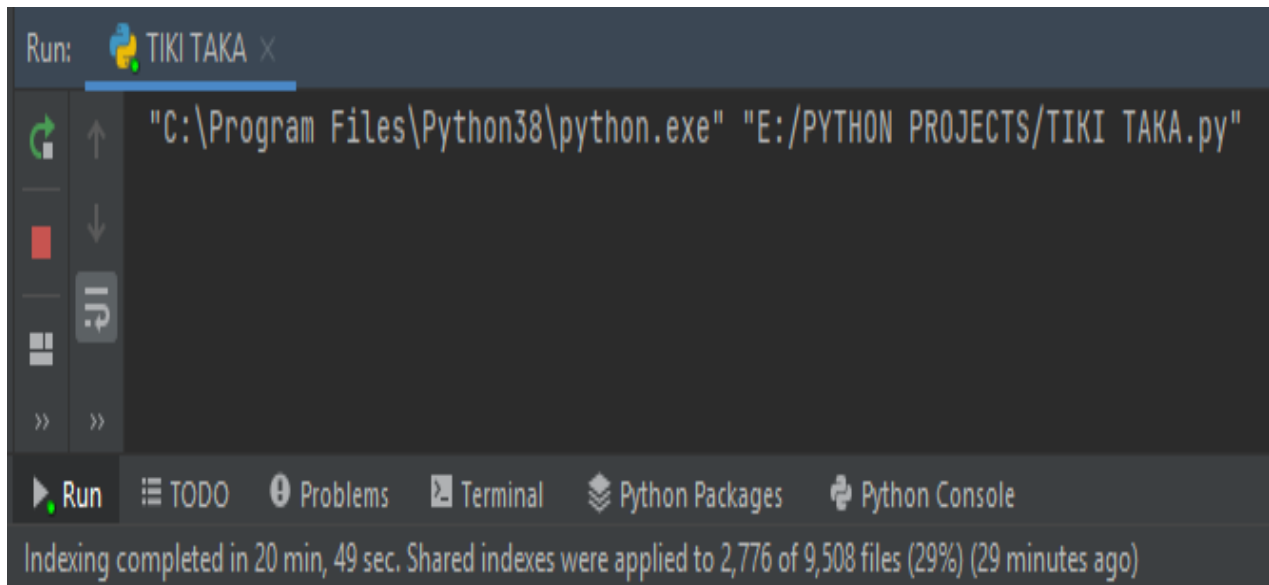


Fig 5.1 The output result of python code

Case 1 : After we have write all the code correctly then the last step is to run our program when we run our program it works correctly and print indexing complited as shown on the above figure(5.1)

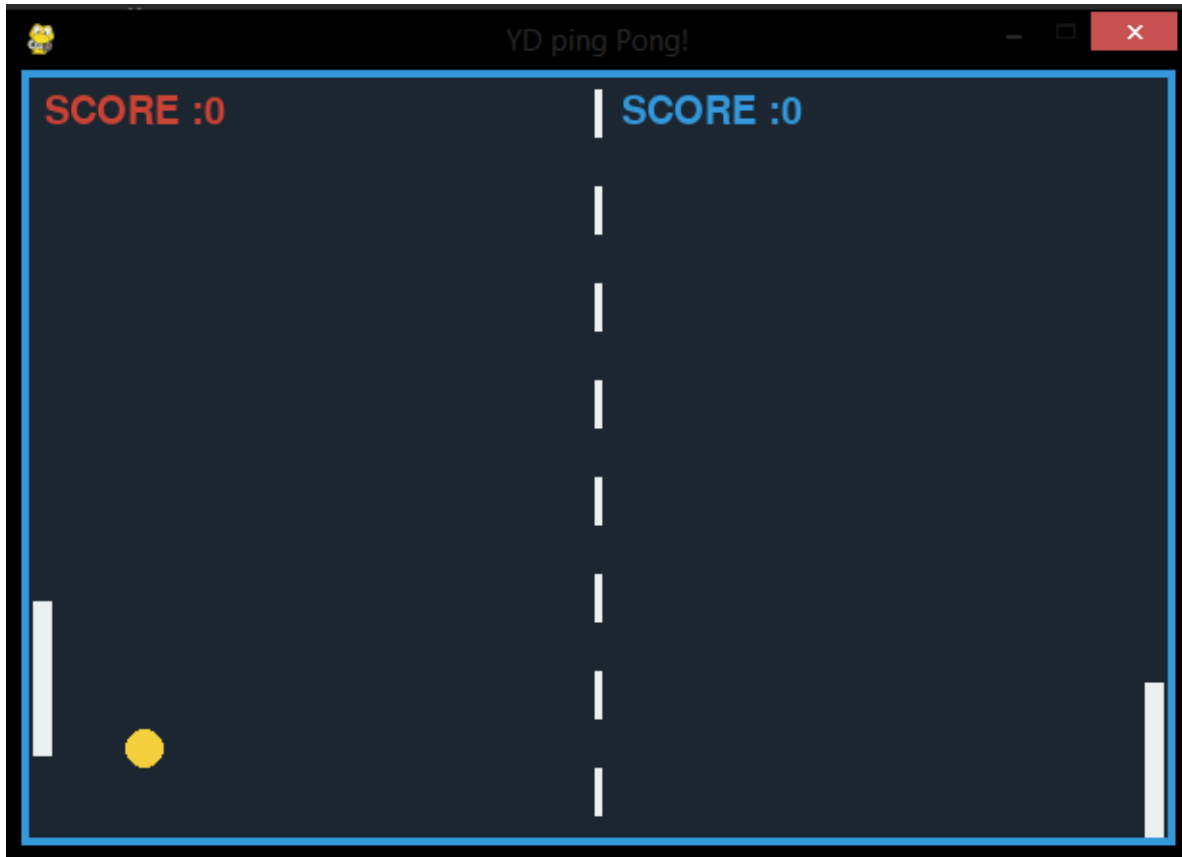


fig 5.2 output of the system

Case 2: when the program runs correctly the the output that is shown on the figure below will be displayed on the screen. In this case both the players have zero score.

As shown in the figure below the game field has

- Game field
- Ball
- Point updater
- Left paddle (for player 1) and
- The right paddle (for player 2) as shown in figure below. also In our first case (fig (5.2)) both the first player and the second player have the same score 0-0

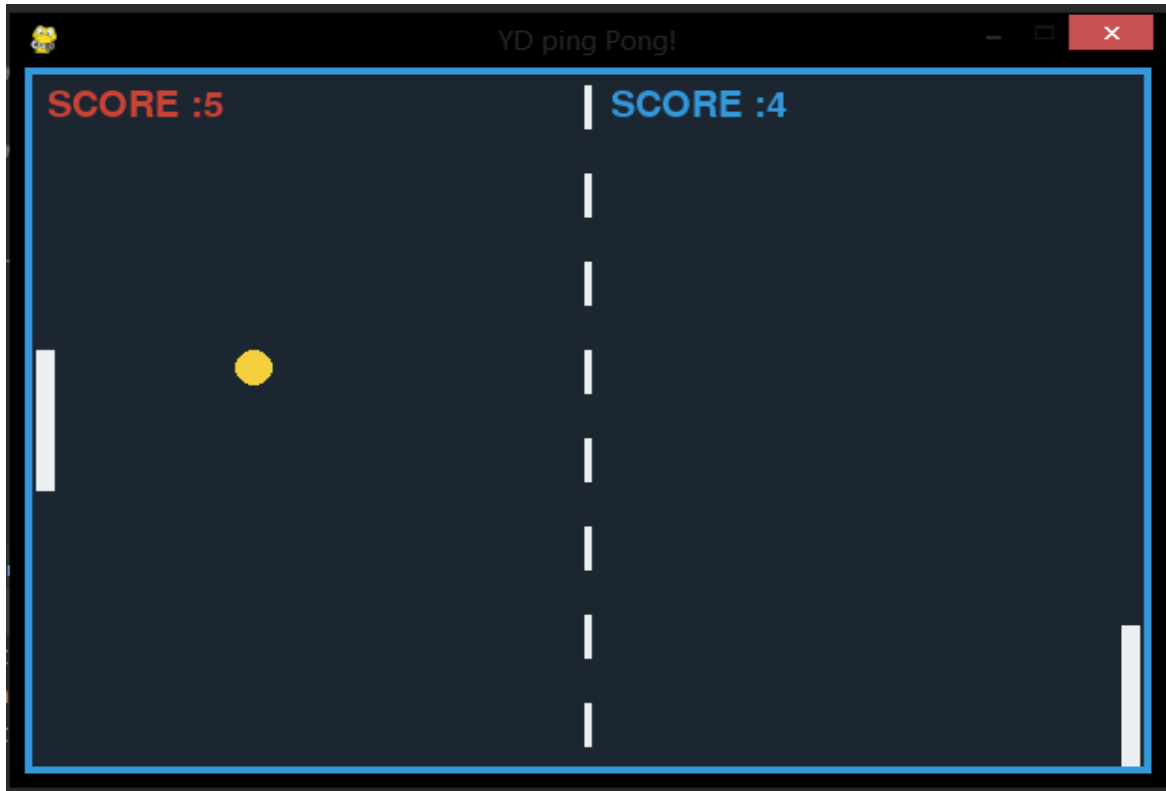


Fig 5.3 Image for updated score

Case 3

When the game is started both player will press the keyboard we have selected

- **R** – to reset the game
- **Space** – to pause the game
- **S** – to stop the game
- **A** - for player 1 to move the paddle to the upper boarder.
- **F**- for player 1 to move the paddle down the boarder.
- **Up key**- for player 2 to move the paddle to the upper boarder.
- **Down key**- for player 2 to move the paddle down the boarder.

As we see on the figure above figure (5.3) the scores are updated as the ball touch the border by passing the paddle of the players. The aim of the game is to kick the ball not to touch the border and score on the other player.



Fig 5.4 the screen when the game ends

Case 4. In this case when one of the two players scores 12 the game will end and the player that score the highest point will be the winner. And the screen will display that the player with the highest score win the game as shown in figure 5.4.

CHAPTER - 6

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

The design of ping pong game was successfully carried out and tested effectively. The system did not pose extra-ordinary constraint and no any other components are needed so everyone can use this game by running on his pc. Finally, the project was challenging because it needs deep knowledge of python coding. gave an exposure into the practical application of theoretical knowledge in solving problems associated with design and construction of the game.

More importantly, the knowledge of python programming is applied in this project, additionally, this project promotes interests in python programming learning concepts. Actually the class did not cover this knowledge, students recognized it as a useful tool in solving real-world problems and decided to explore the topic. This machine can still be improved in many different ways. If more time is allowed, firstly, a more complex machine learning model, or even neural network, will be used. The model will have a 3-dimensional velocity input and 3-dimensional location output for the ball. Secondly, an actual Ping Pong racket will be used instead of the basket, which is more realistic and similar to the real Ping Pong machine player. Also, the ball will be hit in a specific direction and velocity. Additionally, an actual Ping Pong table with a net will be built in the V-rep simulator.

6.2 Recommendation

Accomplishing the full task needs time and resource which gives smart and persuasive output. But the time for the project and sources of information is scarce due to lack of access. Even if we don't have anything to support and we don't have any lab for practice. Therefore such obstacles should be proved for the coming final project. And if it is possible in the future the department should supply students particularly for students who are doing there BSC. With relevant information regarding there project.

REFERENCES

1. ^ "General Python FAQ — Python 3.9.2 documentation". docs.python.org. Retrieved 28 March 2021.
2. <https://python.plainenglish.io/pycharm-professional-python-editors-in-review-b978f0e5c4ba> April 2020
1. ^ Jump up to:^{a b c d e f} Hodges 1993, p. 2
2. ^ Jump up to:^{a b} Letts, Greg. "A Brief history oPing-Pong". About.com. The New York Times Company. Archived from the original on 10 May 2011. Retrieved 29 August 2013.
3. ^ "Member Associations". ITTF. Archived from the original on 7 August 2017. Retrieved 25 June 2017.
4. ^ International Table Tennis Federation 2011, index 2
5. ^ Hurt III, Harry (5 April 2008). "Ping-Pong as Mind Game (Although a Good Topspin Helps)". The New York Times. Archived from the original on 19 June 2011. Retrieved 28 August 2010.
6. ^ WashingtonPost.com. Accessed 2 August 2012.
7. ^ Jump up to:^{a b c d} "A Comprehensive History of Table Tennis". www.ittf.com. ITTF. Archived from the original on 13 March 2015. Retrieved 29 August 2010.
8. ^ Hamilton, Fiona (2 September 2008). "Inventors of ping-pong say Mayor Boris Johnson is wrong". The Times. Archived from the original on 8 January 2019. Retrieved 6 November 2010.
9. ^ Dean R. Karau (August 2008). "Are There Any Trademark Rights Left In The Term Ping-Pong?". www.fredlaw.com. Fredrikson & Byron, P.A. Archived from the original on 30 May 2011.
10. ^ Jump up to:^{a b} Chisholm, Hugh, ed. (1911). "Ping-pong". *Encyclopædia Britannica*. **21**(11th ed.). Cambridge University Press. p. 626.
11. ^ M. Itoh (26 September 2011). The Origin of Ping-Pong Diplomacy: The Forgotten Architect of Sino-U.S. Rapprochement. p. 1. ISBN 9780230339354.

12. ^ ["International Table Tennis Federation Archives"](#). www.ittf.com. ITTF. Archived from [the original](#) on 1 March 2011. Retrieved 29 August 2010.
13. ^ [Jump up to:](#)^{a b c} ["About USATT"](#). United States Olympic Committee. [Archived](#) from the original on 4 November 2010. Retrieved 6 November 2010.
14. ^ [Jump up to:](#)^{a b} Gutttag, John V. (12 August 2016). Introduction to Computation and Programming Using Python: With Application to Understanding Data. MIT Press. [ISBN 978-0-262-52962-4](#).
15. ^ ["Python 3.9.6, 3.8.11, 3.7.11, and 3.6.14 are now available"](#). 28 June 2021. Retrieved 29 June 2021.
16. ["Python 3.10.0b4 is available"](#). 10 July 2021. Retrieved 12 July 2021.
17. ["Why is Python a dynamic language and also a strongly typed language - Python Wiki"](#). wiki.python.org. Retrieved 27 January 2021.
18. ["PEP 483 -- The Theory of Type Hints"](#). Python.org.
19. ["Download Python"](#). Python.org. Retrieved 24 May 2021.
20. File extension .pyo was removed in Python 3.5. See [PEP 0488](#)

APPENDIX

```
# Ping Pong
# Language - Python
# Modules - pygame, sys, random, math
# Controls - Arrow Keys for Right Paddle and WASD Keys for Left Paddle
# importing the modules
import pygame
import sys
import random
from math import *
#initializing the pygame
pygame.init()
#creating the window of the game
width = 600
height = 400
display = pygame.display.set_mode((width, height))
pygame.display.set_caption("YD ping Pong!")
clock = pygame.time.Clock()
#creating the color of the window
background = (27, 38, 49)
white = (236, 240, 241)
red = (203, 67, 53)
blue = (52, 152, 219)
yellow = (244, 208, 63)
#The color of the boarder
top = blue
bottom = blue
left = blue
right = blue
margin = 4
#initial and final score
```

```
scoreLeft = 0
scoreRight = 0
maxScore = 12
font = pygame.font.SysFont("Small Fonts", 30)
largeFont = pygame.font.SysFont("Small Fonts", 60)
```

Draw the Boundary of Board

```
def boundary():
    global top, bottom, left, right
    pygame.draw.rect(display, left, (0, 0, margin, height))
    pygame.draw.rect(display, top, (0, 0, width, margin))
    pygame.draw.rect(display, right, (width - margin, 0, margin, height))
    pygame.draw.rect(display, bottom, (0, height - margin, width, margin))

    pygame.draw.rect(display, white, (width / 2 - margin / 2, 10, margin, 1))
    pygame.draw.rect(display, white, (width / 2 - margin / 2, 60, margin, 1))
    pygame.draw.rect(display, white, (width / 2 - margin / 2, 110, margin, 1))
    pygame.draw.rect(display, white, (width / 2 - margin / 2, 160, margin, 1))
    pygame.draw.rect(display, white, (width / 2 - margin / 2, 210, margin, 1))
    pygame.draw.rect(display, white, (width / 2 - margin / 2, 260, margin, 1))
    pygame.draw.rect(display, white, (width / 2 - margin / 2, 310, margin, 1))
    pygame.draw.rect(display, white, (width / 2 - margin / 2, 360, margin, 1))
```

the Class of the paddle

```
class Paddle:
    def __init__(self, position):
        self.w = 10
        self.h = self.w * 8
        self.paddleSpeed = 6
        if position == -1:
            self.x = 1.5 * margin
```

```
else:
```

```
    self.x = width - 1.5 * margin - self.w
```

```
    self.y = height / 2 - self.h / 2
```

```
# Show the Paddle
```

```
def show(self):
```

```
    pygame.draw.rect(display, white, (self.x, self.y, self.w, self.h))
```

```
# Move the Paddle
```

```
def move(self, ydir):
```

```
    self.y += self.paddleSpeed * ydir
```

```
    if self.y < 0:
```

```
        self.y -= self.paddleSpeed * ydir
```

```
    elif self.y + self.h > height:
```

```
        self.y -= self.paddleSpeed * ydir
```

```
leftPaddle = Paddle(-1)
```

```
rightPaddle = Paddle(1)
```

```
# Ball Class
```

```
class Ball:
```

```
    def __init__(self, color):
```

```
        self.r = 20
```

```
        self.x = width / 2 - self.r / 2
```

```
        self.y = height / 2 - self.r / 2
```

```
        self.color = color
```

```
        self.angle = random.randint(-75, 75)
```

```
        if random.randint(0, 1):
```

```
            self.angle += 180
```

```
        self.speed = 8
```

Show the Ball

```
def show(self):  
    pygame.draw.ellipse(display, self.color, (self.x, self.y, self.r, self.r))
```

Move the Ball

```
def move(self):  
    global scoreLeft, scoreRight  
    self.x += self.speed * cos(radians(self.angle))  
    self.y += self.speed * sin(radians(self.angle))  
    if self.x + self.r > width - margin:  
        scoreLeft += 1  
        self.angle = 180 - self.angle  
    if self.x < margin:  
        scoreRight += 1  
        self.angle = 180 - self.angle  
    if self.y < margin:  
        self.angle = - self.angle  
    if self.y + self.r >= height - margin:  
        self.angle = - self.angle
```

Check and Reflect the Ball when it hits the paddle with several angles

```
def checkForPaddle(self):  
    if self.x < width / 2:  
        if leftPaddle.x < self.x < leftPaddle.x + leftPaddle.w:  
            if leftPaddle.y < self.y < leftPaddle.y + 10 or leftPaddle.y < self.y + self.r < leftPaddle.y + 10:  
                self.angle = -45  
            if leftPaddle.y + 10 < self.y < leftPaddle.y + 20 or leftPaddle.y + 10 < self.y + self.r <  
leftPaddle.y + 20:  
                self.angle = -30  
            if leftPaddle.y + 20 < self.y < leftPaddle.y + 30 or leftPaddle.y + 20 < self.y + self.r <  
leftPaddle.y + 30:  
                self.angle = -15
```



```
    if leftPaddle.y + 30 < self.y < leftPaddle.y + 40 or leftPaddle.y + 30 < self.y + self.r <
leftPaddle.y + 40:
        self.angle = -10
    if leftPaddle.y + 40 < self.y < leftPaddle.y + 50 or leftPaddle.y + 40 < self.y + self.r <
leftPaddle.y + 50:
        self.angle = 10
    if leftPaddle.y + 50 < self.y < leftPaddle.y + 60 or leftPaddle.y + 50 < self.y + self.r <
leftPaddle.y + 60:
        self.angle = 15
    if leftPaddle.y + 60 < self.y < leftPaddle.y + 70 or leftPaddle.y + 60 < self.y + self.r <
leftPaddle.y + 70:
        self.angle = 30
    if leftPaddle.y + 70 < self.y < leftPaddle.y + 80 or leftPaddle.y + 70 < self.y + self.r <
leftPaddle.y + 80:
        self.angle = 45
    else:
        if rightPaddle.x + rightPaddle.w > self.x + self.r > rightPaddle.x:
            if rightPaddle.y < self.y < leftPaddle.y + 10 or leftPaddle.y < self.y + self.r <
leftPaddle.y + 10:
                self.angle = -135
            if rightPaddle.y + 10 < self.y < rightPaddle.y + 20 or rightPaddle.y + 10 < self.y +
self.r < rightPaddle.y + 20:
                self.angle = -150
            if rightPaddle.y + 20 < self.y < rightPaddle.y + 30 or rightPaddle.y + 20 < self.y +
self.r < rightPaddle.y + 30:
                self.angle = -165
            if rightPaddle.y + 30 < self.y < rightPaddle.y + 40 or rightPaddle.y + 30 < self.y +
self.r < rightPaddle.y + 40:
                self.angle = 170
            if rightPaddle.y + 40 < self.y < rightPaddle.y + 50 or rightPaddle.y + 40 < self.y +
self.r < rightPaddle.y + 50:
```

```
        self.angle = 190
        if rightPaddle.y + 50 < self.y < rightPaddle.y + 60 or rightPaddle.y + 50 < self.y +
self.r < rightPaddle.y + 60:
            self.angle = 165
        if rightPaddle.y + 60 < self.y < rightPaddle.y + 70 or rightPaddle.y + 60 < self.y +
self.r < rightPaddle.y + 70:
            self.angle = 150
        if rightPaddle.y + 70 < self.y < rightPaddle.y + 80 or rightPaddle.y + 70 < self.y +
self.r < rightPaddle.y + 80:
            self.angle = 135
```

Show the Score

```
def showScore():
    leftScoreText = font.render("SCORE :" + str(scoreLeft), True, red)
    rightScoreText = font.render("SCORE :" + str(scoreRight), True, blue)
    display.blit(leftScoreText, (3 * margin, 3 * margin))
    display.blit(rightScoreText, (width / 2 + 3 * margin, 3 * margin))
```

when the Game is Over

```
def gameOver():
    if scoreLeft == maxScore or scoreRight == maxScore:
        while True:
            for event in pygame.event.get():
                if event.type == pygame.QUIT:
                    close()
                if event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_q:
                        close()
                    if event.key == pygame.K_r:
                        reset()
            if scoreLeft == maxScore:
                playerWins = largeFont.render("LEFT PLAYER WINS!", True, red)
            elif scoreRight == maxScore:
```

```
playerWins = largeFont.render("RIGHT PLAYER WINS!", True, blue)

display.blit(playerWins, (width / 2 - 100, height / 2))
pygame.display.update()

def reset():
    global scoreLeft, scoreRight
    scoreLeft = 0
    scoreRight = 0
    board()

def close():
    pygame.quit()
    sys.exit()

def board():
    loop = True
    leftChange = 0
    rightChange = 0
    ball = Ball(yellow)

#creating the loop and the keyboard of the game
while loop:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            close()

        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_q:
                close()

            if event.key == pygame.K_SPACE or event.key == pygame.K_p:
                Pause()

            if event.key == pygame.K_r:
                reset()

            if event.key == pygame.K_a:
                leftChange = -1
```

```
    if event.key == pygame.K_f:
        leftChange = 1
    if event.key == pygame.K_UP:
        rightChange = -1
    if event.key == pygame.K_DOWN:
        rightChange = 1
    if event.type == pygame.KEYUP:
        leftChange = 0
        rightChange = 0
#declaring the variable
leftPaddle.move(leftChange)
rightPaddle.move(rightChange)
ball.move()
ball.checkForPaddle()
display.fill(background)
showScore()
ball.show()
leftPaddle.show()
rightPaddle.show()
boundary()
gameOver()
pygame.display.update()
clock.tick(60)
board()
```