

Chapter 1

Introduction

1.1. Background:

Gaming is for entertainment. People play game for pass time, beat boredom, to set new milestone and basically for fun. Online gaming is sociable, you may offer to play together or against each other with your friends, you can chat with your friends or make new friends from across the world in the online gaming platform. Games mainly include simplicity, graphics, user interface, reaction time etc. In past history there is negligible amount of games have been developed. Because of it visionless folks are unable to find fun in digital world.

Now days gaming have become big entertainment platform as well as social communication lots of games of mobile phone target amuse, relax, enjoy, and Refresh you. There are 37 million people in the world who are blind, and over 15 million are from India. There are very few computer games in market for blind people. Visionless people can't entertain them self in gaming field. As the technology is developing we are leaving behind special people.

There are very few games and the game based on popular game just like cricket. Such game works on voice command. Player get the information of the game via voice output from the game. Player needs to wear headphone for best experience of the game. Player response is based on change of voice or change in frequency of voice. Player reflex action is swiping left or right based on voice coming from the side of game. This type of method used in the game that is built for blind people. This needs to be upgrade.

1.2 Objectives:

It will give a voice-command instruction

Information of levels

Level 1(easy)

Level 2(hard)

Level 3(very hard)

Credits

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

The main purpose of the game is entertainment of blind persons. It will improve the reflex action of player it can be used for stress relaxations. Voice command will improve the understandable ability of blind person.

1.3.2 Scope

Our game can be played by blind persons and normal persons as well. it does not need any external accessories for playing such a game which makes it cost efficient. Saved person can see the score set by him.

1.3.3 Applicability

Our game is for the entertainment purpose.

It is for the passing the time.

It will help in improvement of reflex action.

An interactive application for blind people.

It will be the entertainment platform for people who have disability of eyes.

It will have some basic key function which will make it easy to play.

It will be interesting game because of uncertainty of game.

Chapter 2

Survey of technology

2.1 PYTHON

Python is an interpreted high-level programming language and basically used for general purpose programming language. It is created by Guido van Rossum and python was first released in 1991, Python has a design type which focus on code readability, notably using significant whitespace. It provides constructs that is helpful for big project as well as small projects

Front-end Web Development refers to building the interactive page from which user can do whatever he/she wants to do. When you search anything on your browser and the output you get is made up of HTML, CSS and JavaScript, the main components to any website.

Back-end Development refers to the parts of the website that user can not interact with the page directly. The back end handles algorithms, database interaction, application logic, and the processing of user requests.

Python has dynamic type system and it also support automatic memory management. It includes object oriented, functional, imperative, and procedural and it has a large library which provides a lot of useful predefine functions. Python is available for many operating system. CPython is a reference implementation of python. Even python has development model which is based on community which helps to learn it easily. Python software foundation manages the python.

Server is very useful when it come to household application we have to develop a such a application which is easy to use and has a good user interface so that other can know about it and they can operate it very easily., and containerization another way servers provision their operating systems can be out for building a system for locals.

2.2 Hardware requirement

1. Processor:	Intel Pentium 4 (Or) Higher
2. RAM:	512 Mb & Above hard
3. Disk Drive:	500 Mb Free Space Or Above
4. Speakers	
5. Functional keyboard	

When it comes to hardware need python is one of the best applications to use for coding. It does not need any extra requirements than the system. It has very simple User interface so but provided with large number of library which provides great number of pre defined function. When it comes to space management python provides option to use less number of libraries and if user want to use any extra library than user can install it from command line by simply entering installation command or by downloading it online.

Chapter 3

Requirement and Analysis

3.1 Problem definition:

3.1.1 Lack of references:

There is no such a project existing according to our research. So it is difficult to build project by own.

3.1.2 New language usage:

We are using python language for coding this very new language which is not much popular. And less known language. The building a game can take a time and efforts which can be a complex work to do.

3.1.3 Processing time:

After every new updating the code we have to save all data and then have to run which will take lots of time (Iteration)

3.1.4 Voice command:

We have to give command to player so we have to generate hole game instruction in voice command file which will take a lot of time and efforts.

3.1.5 Error management:

We have to take back-up each and every time of project before running it. Because one error can crash the whole project.

3.2 Software Requirement and Specifications

3.2.1 Functional Requirements:

The project needs strong internet connection

Good audio recorder

Photoshop application to create character and environment

Voice edit application

Platform for building the application

3.2.2 Non-functional requirements:

Nonfunctional requirements define system attributes such as security, reliability, accuracy, maintainability, portability, supportability, scalability and usability.

Reliability:

The system must be reliable.

Portability:

System needs to be portable and platform independent. It must be available for the users who use different Operating Systems.

Maintainability:

The codes written in the project must be easy to maintain and must also be easy to modify. The code needs to be flexible, hence making it maintainable.

Scalability:

The capability of software is basically meant by scalability. Since the project is making use of latest technologies, the capability of software is scalable. For e.g.: If an application run in 2 seconds for 100 users would it be run also in 2 seconds with 200 users.

Usability:

The easier the software performs a specific task, the better is the usability. Our software is simple and easy to use hence maintaining its Usability.

Performance:

How fast and efficiently the software works, the better is the performance. Our software meets the following requirements hence its performance is not compromised. Supportability (serviceability). The project will have the ability to install, configure and monitor the software. And also is used to identify the faults and debug them.

3.3 Planning and Scheduling

One of the important part of project management is planning and scheduling. As we know that this is a game and it is expected to face a lot of glitches and errors while coding so we have plan and schedule according to that. Let us see the points that we consider while planning and scheduling.

3.3.1 Planning:

Planning can be thought as determining all the small tasks that must be carried out in order to accomplish the goal. Planning also takes into account, rules, and known as constraints, which, control when certain tasks can or cannot happen.

3.3.2 Scheduling:

Scheduling can be thought as determining whether adequate resources are available to carry out the plan. Proper Gantt chart and Program Evaluation Review Technique of the project will be shown.

3.3.3 Gantt chart and network diagram

One of the most important consideration in development of project is time evolution. When it comes to time schedule management it is very important since more development time effect machine time, cost and cause delay in the development of some other system.

Our project need time management. Hence gantt chart is very important factor for us.

This are some screenshots of gantt chart.

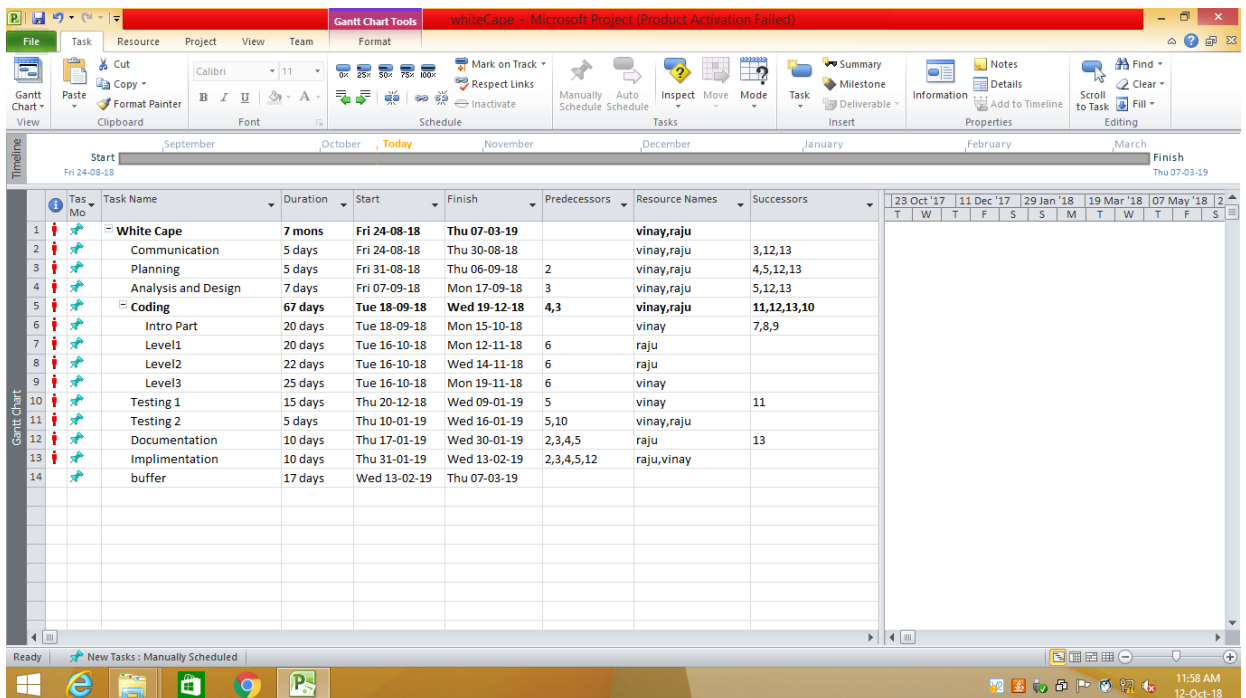


Figure 3.3.1 gantt chart

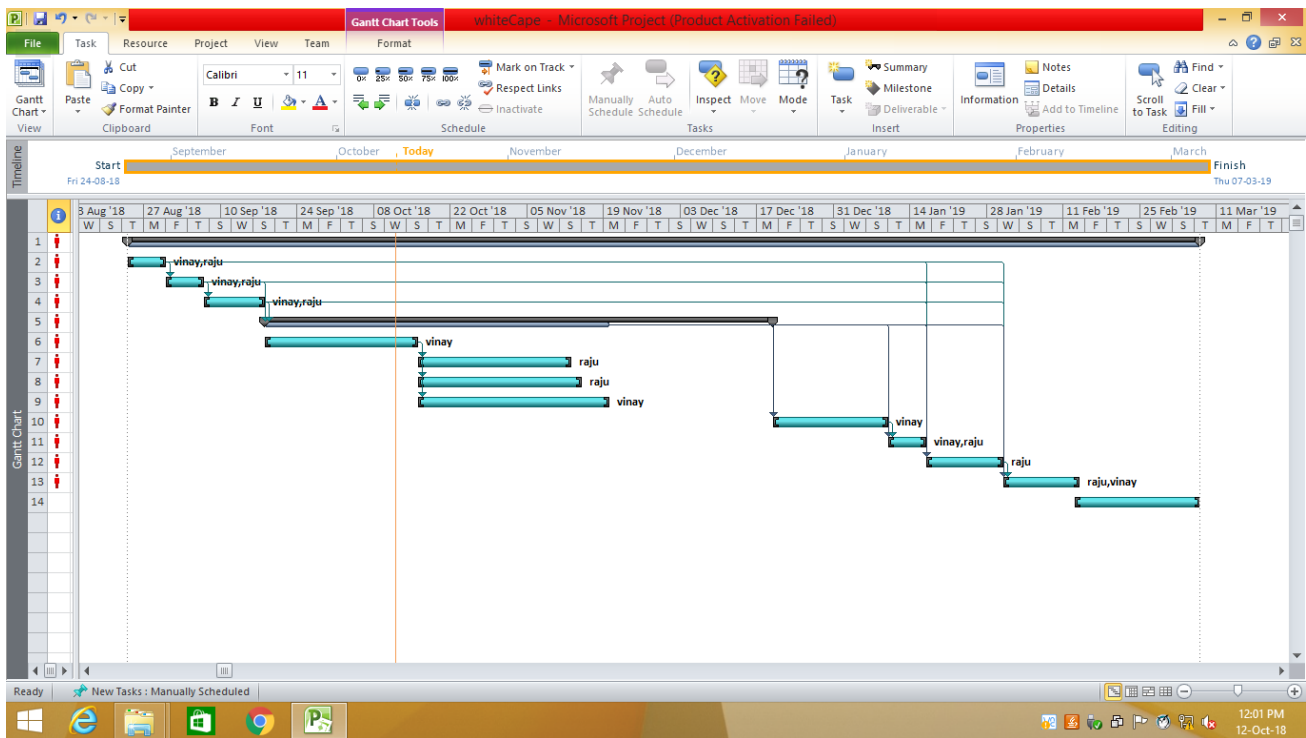


Figure 3.3.2 gantt chart-2

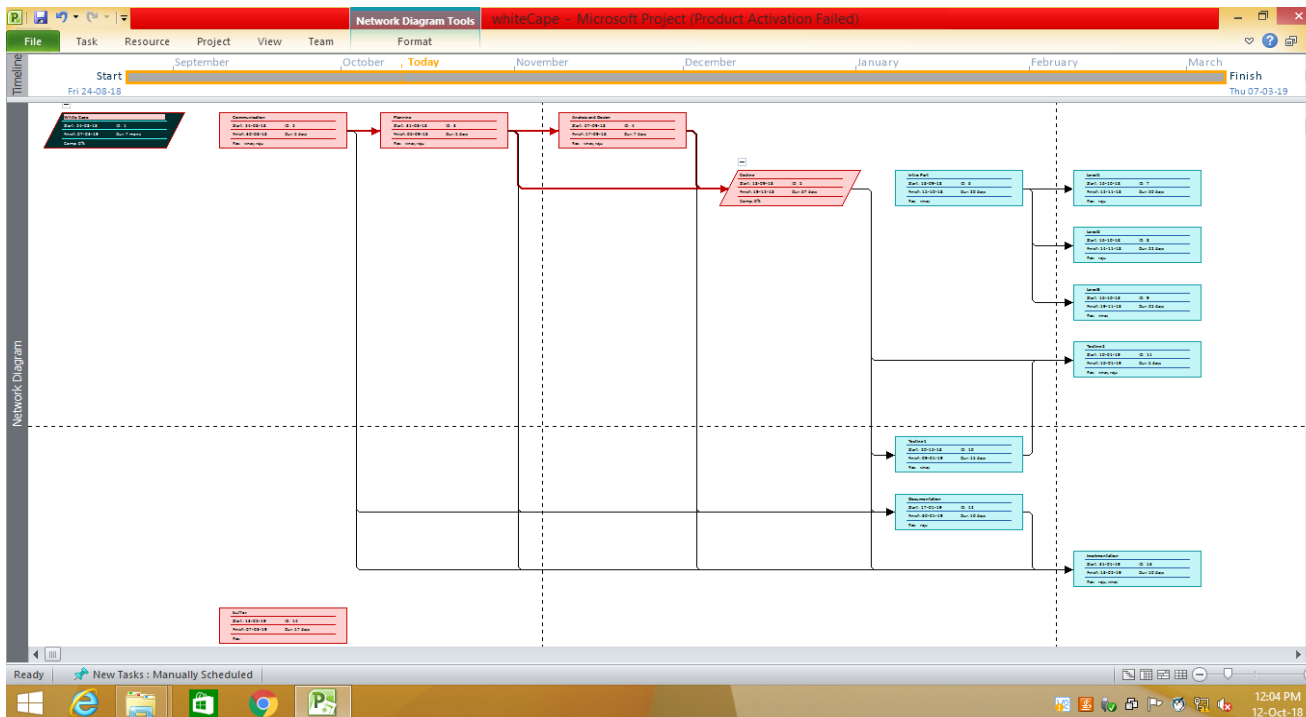


Figure 3.3.3 gantt chart-3

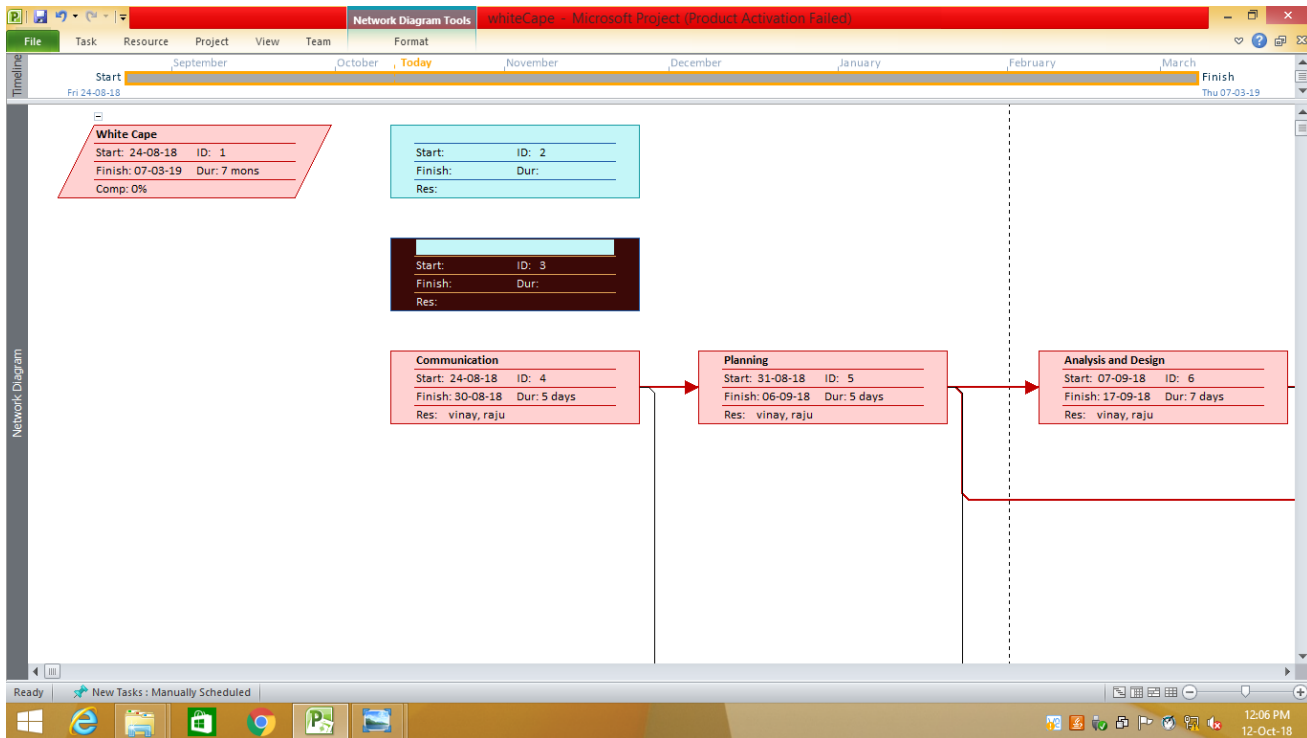


Figure 3.3.4 gantt chart-4

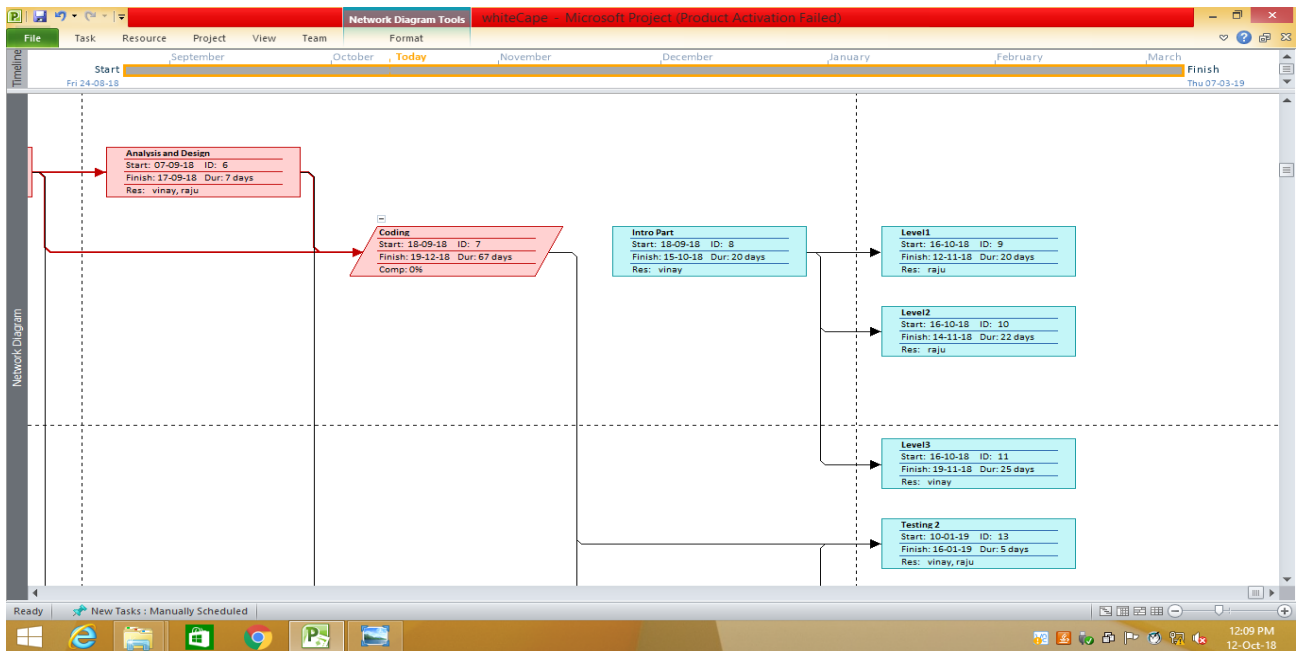


Figure 3.3.5 gantt chart-5

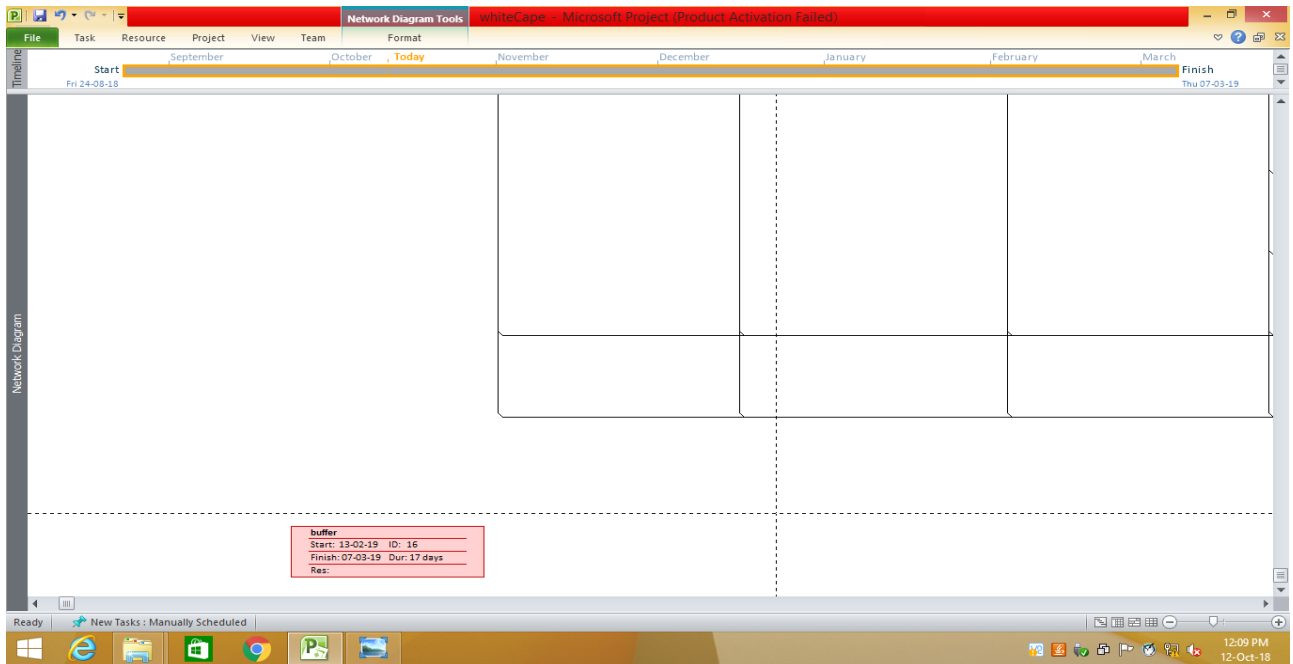


Figure 3.3.6 gantt chart-6

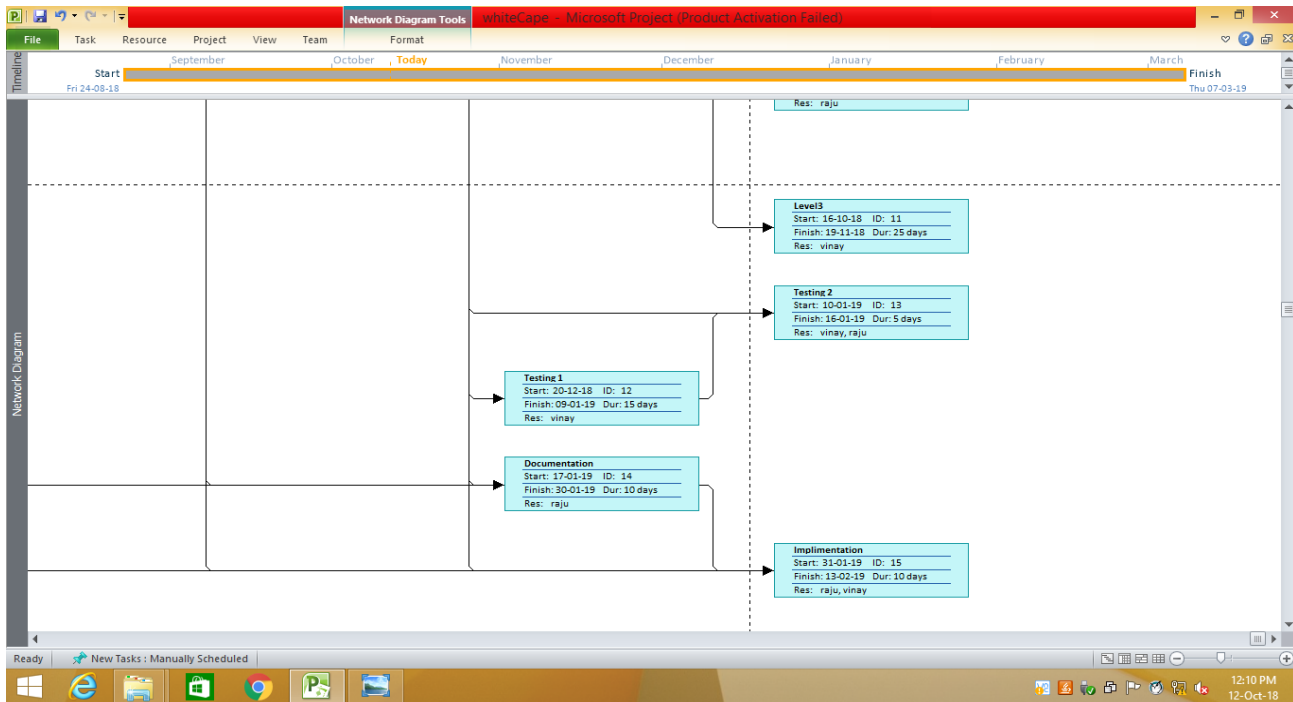


Figure 3.3.7 gantt chart-7

3.4 Software Requirements:

1. Python 3.6.2
2. Operating system Windows or macOS or Linux
3. Safari
4. Internet Explorer 9
5. Google Chrome
6. Microsoft paint
7. Photoshop
8. Photo editing tools
9. Audio editing tools
10. Intel® Distribution for Python* 2018
11. Etc.

3.5 hardware requirements:

1. Processor:	Intel Pentium 4 (Or) Higher
2. RAM:	512 Mb & Above hard
3. Disk Drive:	2 to 3GB Free Space Or Above
4. Speakers	
5. Functional keyboard	

Recommended System Requirements:

Processors:

- Intel Core i5 processor 8 GB of DRAM
- Intel Xeon processor E5-2698 v3 4 GB of DRAM
- Intel Xeon Phi processor, 16 GB of MCDRAM Disk space: 2 to 3 GB

Minimum System Requirement

- 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

3.6 Risk identification

The development and many risks may have to be dealt post development. Risk identification will be an important step of the process, where we need to identify the risks and prioritize of them.

According to business needs Risks identified as of now are as follows.

3.6.1 Backing up data

Our project is a game we have to make a backup of every file because a single error in code can lead to crashing of application. We have to make system backup as well to make sure no loss of any data.

3.6.2 Technical Performance issue

As there are lot of technical challenges involved in development of the project and due to inexperience of team members, there is a chance that overall system performance may be compromised in order for certain functionality to work.

3.6.3 Inadequate initial data

Main targeted users of our platform are beginners who lack coding knowledge. But in the initial stages of our application, there won't be much data for them to access, and this may motivate the users to quit the platform.

3.6.4 Challenging User Interface

As we know that we are building application for blind persons so it is very big challenge to use to build the application which is very user-friendly easy to understand and functional on commands.

Poor UI design will lead to navigation and discoverability issues, where users may not understand how to reach a particular section or a page, or may not know if a particular page exists leading to discovery of functionality

Chapter 4

System design

4.1 Basic Modules

Here the overall system will be divided into small modules and will be made ready to be executed one by one. These modules will be further implemented together to make a whole project.

The main objective to divide the overall components into small modules is to manage each of the parts and develop each part or module separately. After successful development of each module we will/can integrate all the modules into one system

Some important modules that are divided from the system to develop it separately are:-

- 1) Game Introduction
- 2) Key introduction
- 3) Menu
- 4) Pause
- 5) Credits
- 6) Exit game

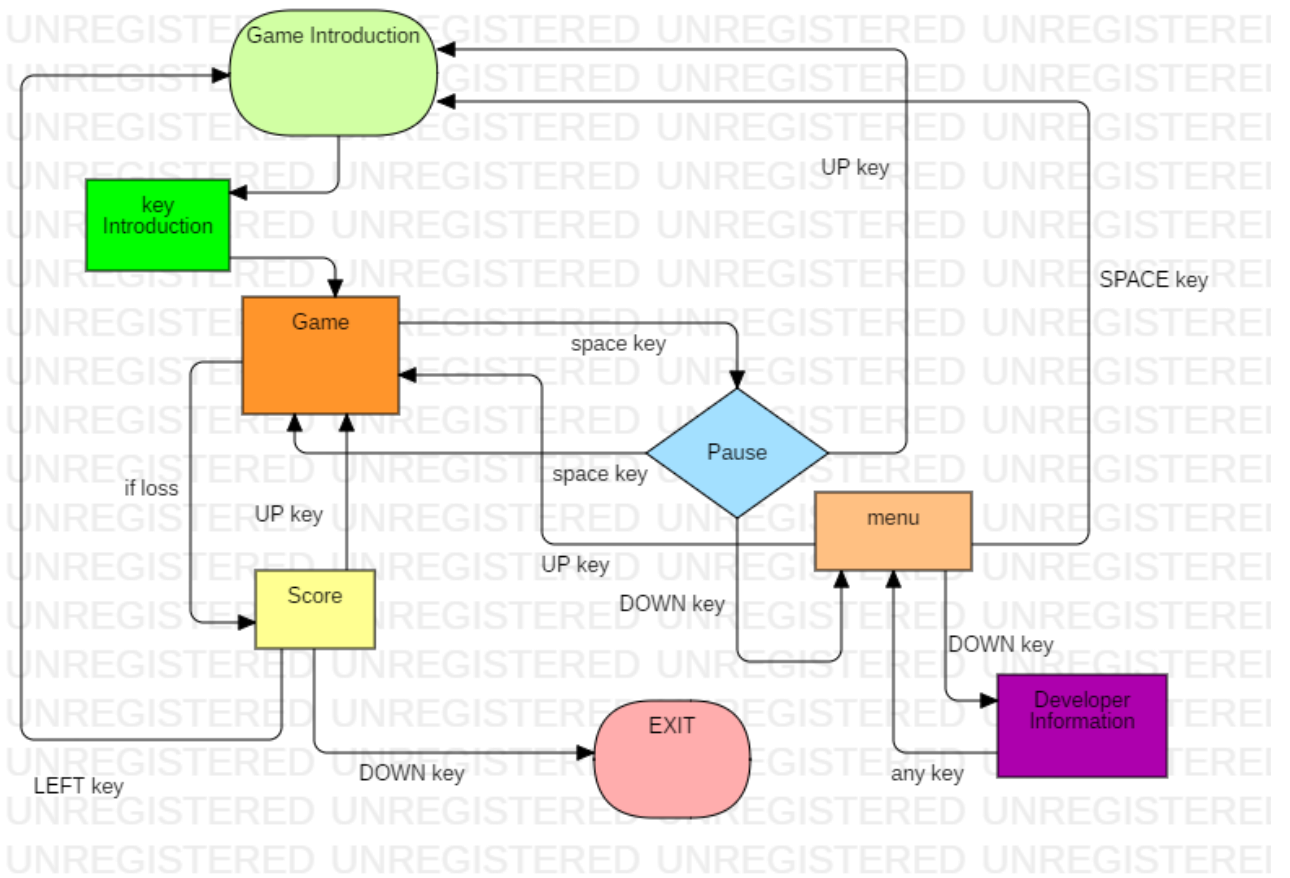
4.2 Procedural Design

Procedural design is a systematic way for developing algorithms or procedurals. We can use procedural design to get an initial idea of the flow of a particular function.

4.3 Logic Diagrams

We will be creating some UML (Unified Modeling Language) Diagrams to define the flow of procedure that improves its comprehension and will help us during implementation. For example, Use Case Diagram, State Diagram.

4.3.1 Flowchart diagram:



Fig, 4.3.1 flow chart

4.3.2 Use Case diagram:

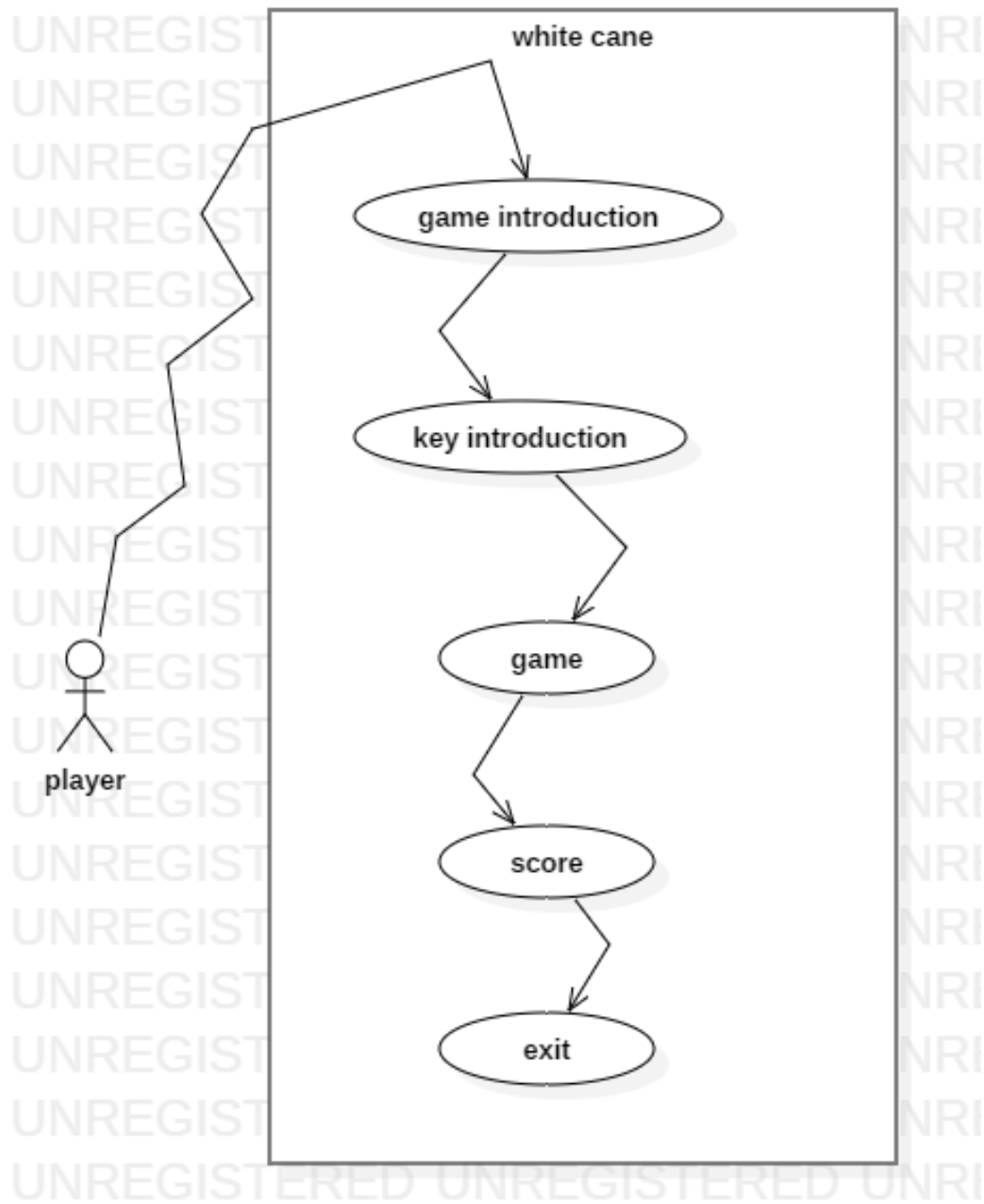
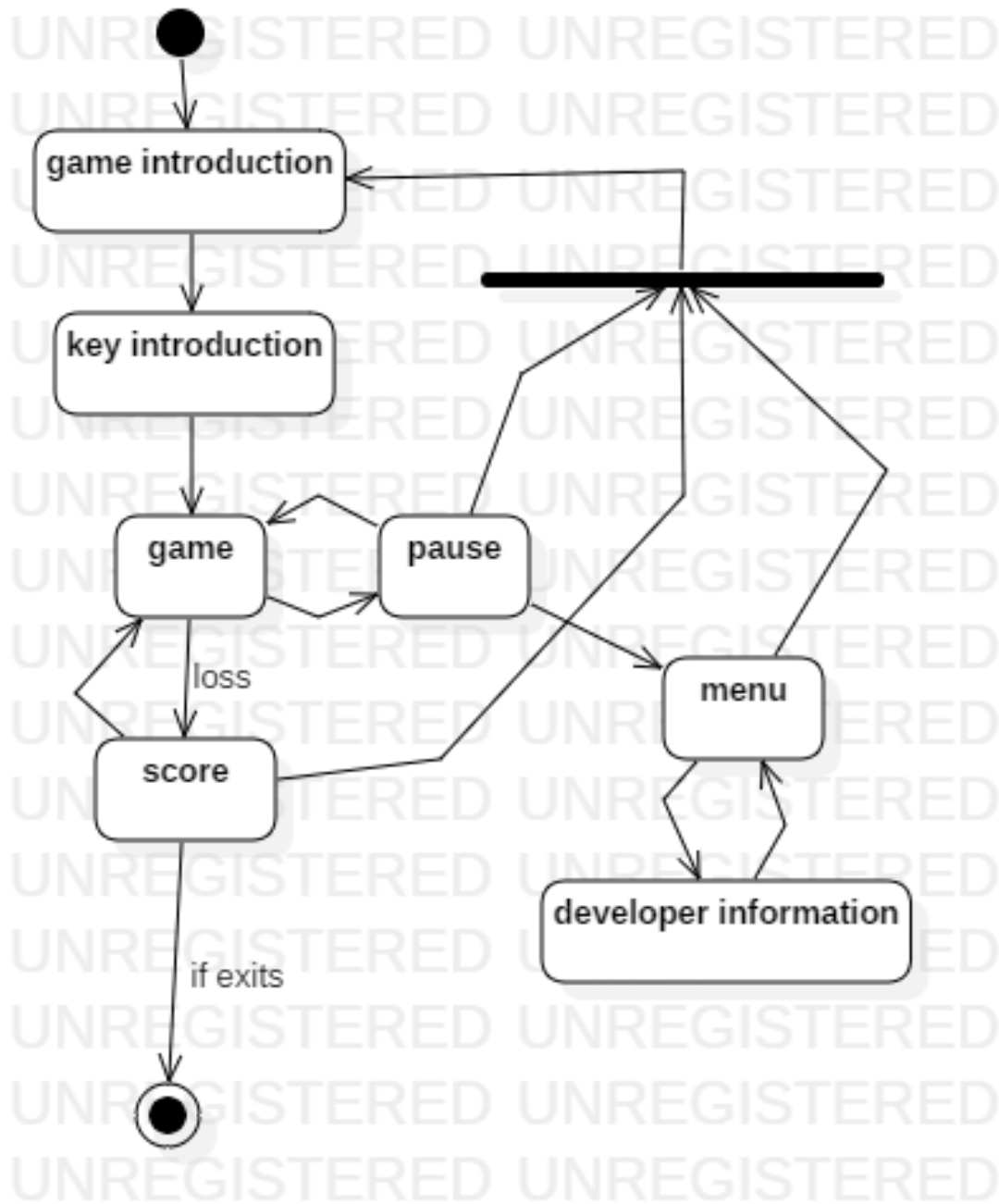


Fig 4.3.2 Use case diagram

4.3.3 Activity diagram:



Fig, 4.3.3 Activity Diagram

4.4 Application Interface:

This is the graphical representation of a game in easy words.

- **Main menu view**

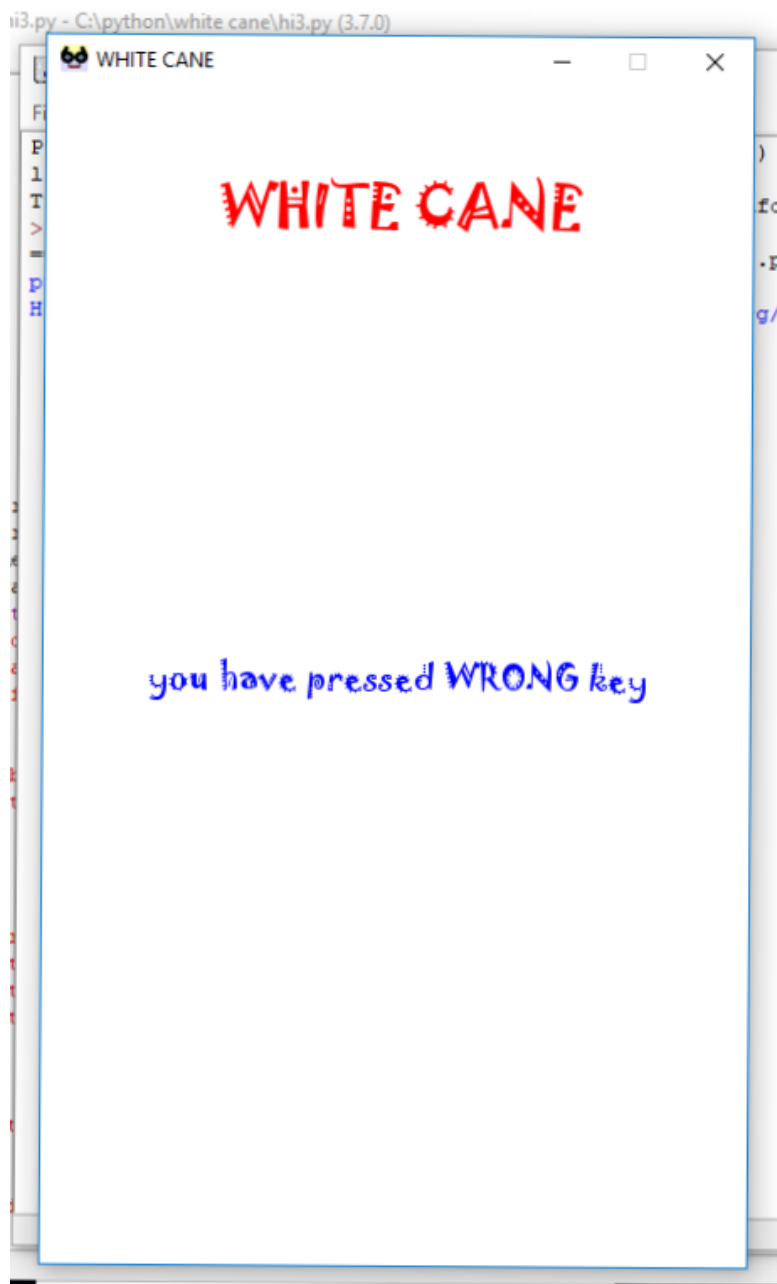


Fig 4.4 Main Menu

In this user get to know about the game and after word user get instruction about key and player have press key so that he can use it while playing. If he press correct key get feedback according to that.

For e.g. Player press right key.

O/p will be YOU HAVE PRESSED RIGHT Key in voice

If player press any other key than the usable key then player get to know that pressed key is wrong try again.

E.g. YOU HAVE PRESSED WRONG KEY in voice

4.5 Graphical representation of gaming interface:

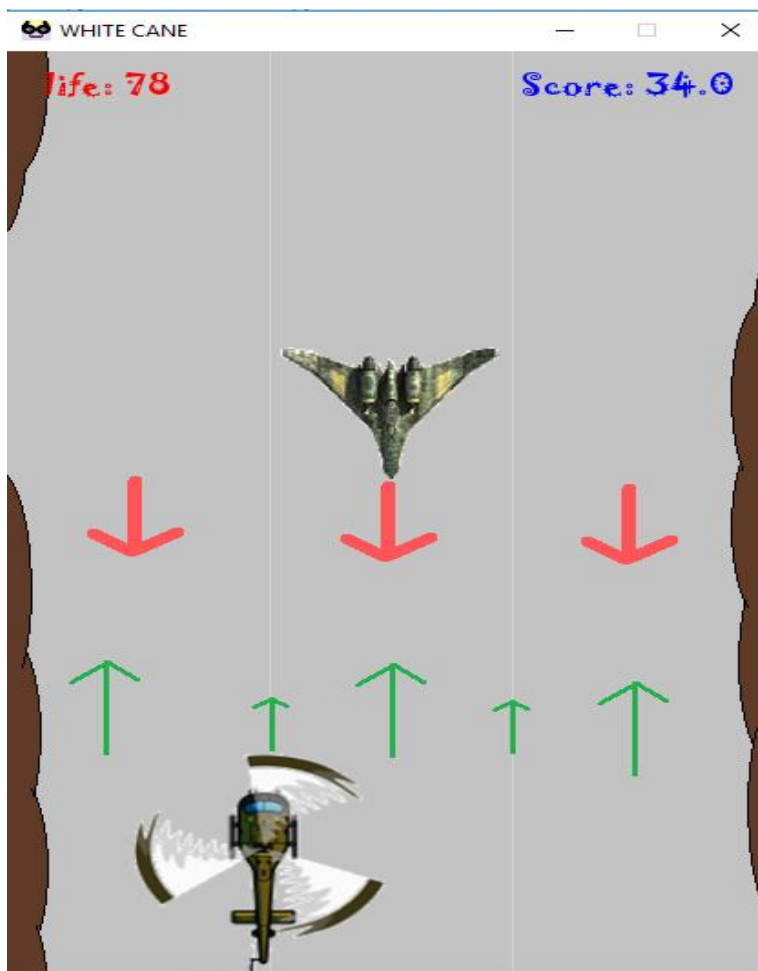


Fig 4.5 Graphical representation of gaming interface

There are total 5 position available for helicopter and 3 possible position for the plane. Crash can occur if they touch each other. (Green arrow)

Let us see the 3 possible from which plane can come. (Red arrow)

First position-

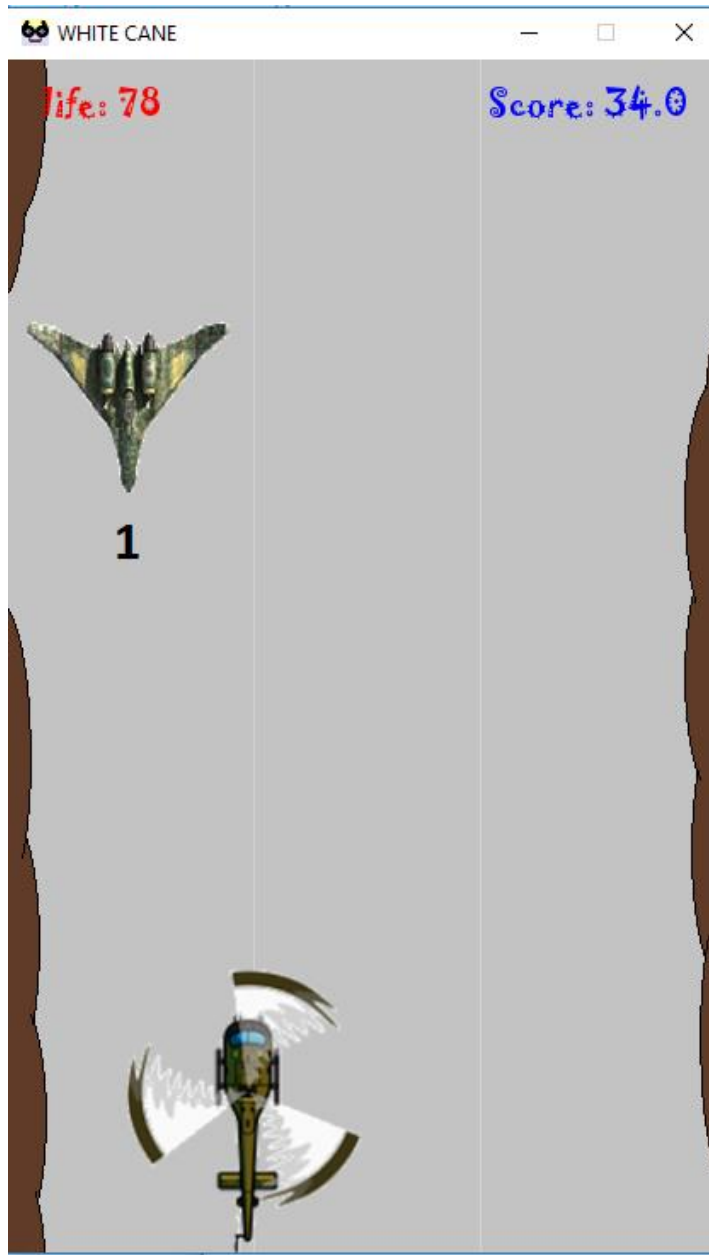


Fig 4.5.1 first position

Second position-

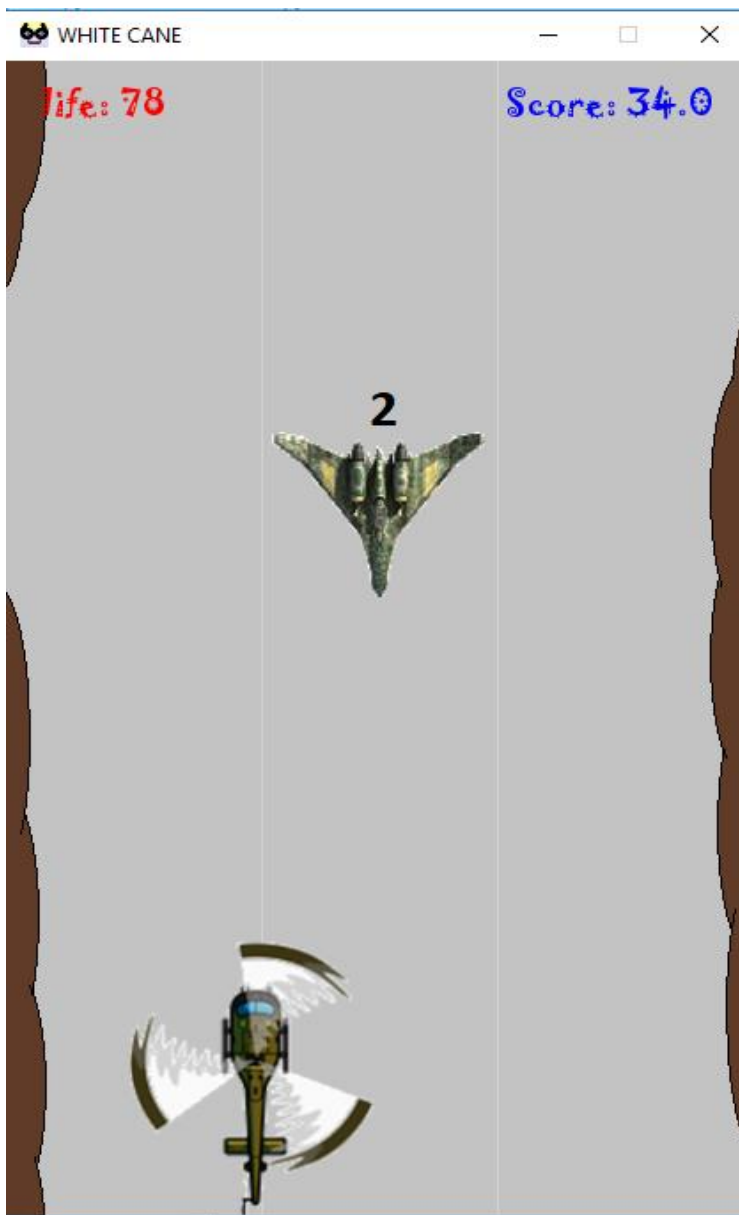


Fig 4.5.2 Second position

Third position-



Fig 4.5.3 Third position

Chapter 5

Implementation and Testing

5.1 Implementation Approaches

Key press logic code:

```
if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_SPACE:
        stop_all()
        press_space.play()
        message_display('you have pressed SPACE key',20,dw/2,dh/2,blue)
        e=e+1
    elif event.key == pygame.K_UP:
        stop_all()
        press_up.play()
        message_display('you have pressed UP key',20,dw/2,dh/2,blue)
        a=a+1
    elif event.key == pygame.K_DOWN:
        stop_all()
        press_down.play()
        message_display('you have pressed DOWN key',20,dw/2,dh/2,blue)
        b=b+1
    elif event.key == pygame.K_LEFT:
        stop_all()
```



```

        press_left.play()

        message_display('you have pressed LEFT key',20,dw/2,dh/2,blue)

        c=c+1

    elif event.key == pygame.K_RIGHT:

        stop_all()

        press_right.play()

        message_display('you have pressed RIGHT key',20,dw/2,dh/2,blue)

        d=d+1

    elif event.key != pygame.K_UP or pygame.K_LEFT or pygame.K_DOWN or
pygame.K_RIGHT or pygame.K_SPACE:

        stop_all()

        press_wrong.play()

        message_display('you have pressed WRONG key',20,dw/2,dh/2,blue)

    else:

        game_loop()

```

Menu Screen code:

```

def menu():

    stop_all()

    menu_options.play()

    gameDisplay.fill(white)

    message_display('WHITE CANE',33,dw/2,dh/10,red)

    message_display('** MENU **',20,dw/2,dh/10*2,green)

    message_display('for HOME press SPACE',20,dw/2,dh/10*3,blue)

    message_display('to play again press UP key',20,dw/2,dh/10*4,blue)

    message_display('to know about developers DOWN key',20,dw/2,dh/10*5,blue)

```

```

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            quit()
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                intro()
            if event.key == pygame.K_UP:
                game_loop()
            if event.key == pygame.K_DOWN:
                developer()
        else:
            menu()

```

Crash event Screen code:

```

def event_of_crash():                                     #the event where player can choose
    what to do if he loses the game
    gameDisplay.fill(white)
    message_display('good game player!!!!',33,dw/2,dh/20,red)
    message_display('for play again press UP key ',20,dw/2,dh/10*2,blue)
    message_display('for QUIT press key DOWN key',20,dw/2,dh/10*3,blue)
    message_display('for INTRO press LEFT key',20,dw/2,dh/10*4,blue)
    stop_all()
    crashed_and_options.play()
    something = False
    while not something:
        for event in pygame.event.get():

```

```

if event.type == pygame.QUIT:
    print("quit function is running")
    pygame.quit()
    quit()

if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_UP:
        game_loop()
    elif event.key == pygame.K_DOWN:
        stop_all()
        pygame.quit()
        quit()
    elif event.key == pygame.K_LEFT:
        intro1()
    else:
        event_of_crash()

```

5.2 Coding Details and Coding Efficiency

5.2.1 Code Efficiency

Our project contains not more than 1k line of code. In our project we have imported approximate 100 number of voice files. Here we are using the cocomo model which is based on regression model that is based on line of code.

$$PM = 3.0 * (SIZE) * 1.2$$

Thus the required equation is $3.0 * (1) * 1.2 = 3.6 \text{ kloc}$

5.2.2 Coding Details

The application uses simple logic and it has more than 70 files which include backup, code, image, audio, icon, font, etc.

We have used loops in coding we have defined the different functions so that we can avoid re writing the codes.

We have to run only single code to run the application so we do not required any kind of arrangement so we have sorted files according to their name and types. As you can see the arrangement of the files.

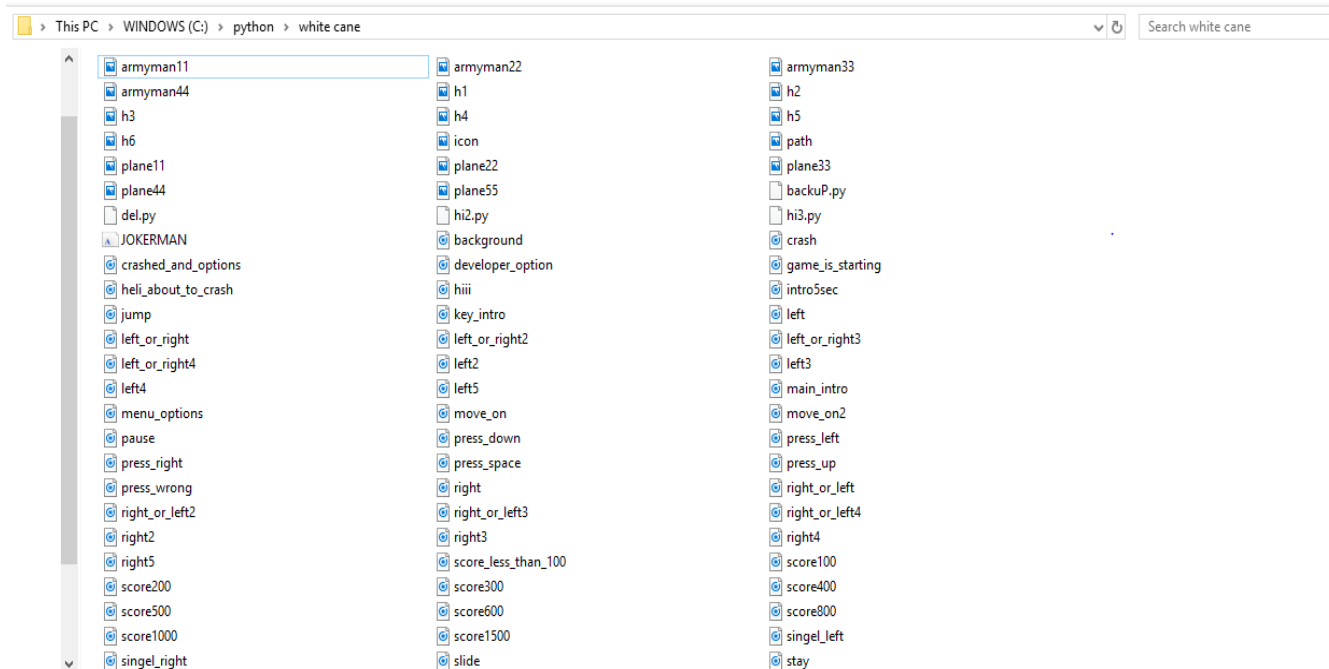


Fig 5.2.1 Code & Files Structure

5.3 Testing Approach

Test case 1 - To verify the structure of the application and how it supports our device. [PASS]

Test case 2 - To check module 1 is working correctly or not . [PASS]

Test case 3 - To verify whether module 1 is not crashing or running without any error. [PASS]

Test case 4 - To check module 2 is working correctly or not. [PASS]

Test case 5 - To verify whether module 2 is not crashing or running without any error. [PASS]

Test case 6 - To check module 3 is working correctly or not. [PASS]

Test case 7- To verify whether module 3 is not crashing or running without any error. [PASS]

Test case 8 - To check module 4 is working correctly or not . [PASS]

Test case 9 - To verify whether module 4 is not crashing or running without any error. [PASS]

Test case 10 – To check whether event handling is working properly or not.
[PASS]

Test case 11 – To check whether used image is fitting in size or not. [PASS]

Test case 12- To check the click event for game module event

5.3.1 For Right key

Key press	o/p(event)	Pass/Fail/no result
Right key	Move to right (70px)	Pass
Other key press	Nothing happened	Fail
Nothing pressed	Waiting state	No result

5.3.2 For LEFT key

Key press	o/p(event)	Pass/Fail/no result
LEFT key	Move to left (70px)	Pass
Other key press	Nothing happened	Fail
Nothing pressed	Waiting state	No result

5.3.3 For UP key

Key press	o/p(event)	Pass/Fail/no result
UP key	Increase in speed(0.2%)	Pass
Other key press	Nothing happened	Fail
Nothing pressed	Waiting state	No result

5.3.4 For DOWN key

Key press	o/p(event)	Pass/Fail/no result
DOWN key	Decrease in speed(0.2%)	Pass
Other key press	Nothing happened	Fail
Nothing pressed	Waiting state	No result

5.3.5 For SPACE key

Key press	o/p(event)	Pass/Fail/no result
SPACE key	Pause the game	Pass
Other key press	Nothing happened	Fail

Nothing pressed	Waiting state	No result
--------------------	---------------	-----------

5.3.1 Integration and System Testing

In this chapter we are going to see the possible way we can develop the program through using better infrastructure and we are going to see that how can we upgrade our project to the next level.

5.3.2 Introduction

In software testing we have to build the test level on from the previous level so it is important that so that we can do testing in correct order without lacking any steps, after completion of level we Can go to the next level for testing.

5.3.3 Integration

Integration and system testing is a type of software testing, this type of testing is done just before releasing the product. Software testing follows very ideal set of test from which we can make sure that the project that is to be released is perfect and error less. Integration and system testing is basically done within the set of people/group who has one work to do that is to test only in the system development life cycle. As the name suggest it deals with small component not with whole software.

There are three main things which is to be checked for running the test successfully the first is a test plan second is a test cases and the third one is test data is produced to effectively test that the integration is successful. The test data is one of the part of integration testing.

5.3.4 Integration Test Plan

After the implementation of product testing this things to be consider.

- 1) A strategy to use when testing the integrated modules and how the tests will be conducted.
- 2) What will be tested for example software features.
- 3) What is the time scale and time management?
- 4) Responsibilities, e.g. personnel.
- 5) Testing pass and fail condition.
- 6) Risk involved
- 7) Approval from all important people involved.

5.3.5 Critical Path Method (CPM)

We can take following mention things as input.

1. A priority chart whit schedule.
2. Estimates of task time management.
3. A written work of each task to a programmer.
4. Set of order of module flow each programmer.

Result and Discussion

6.1 User Interface Output:

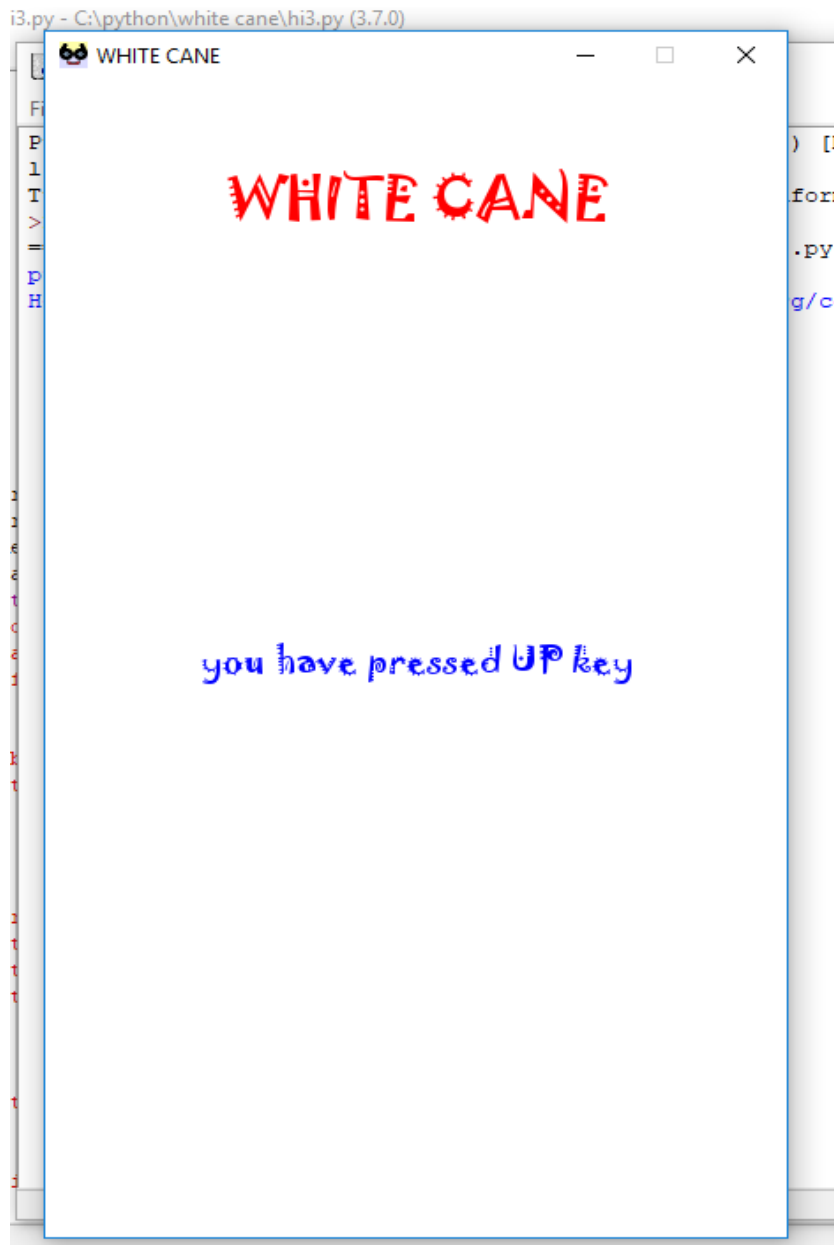


Fig 6.1 when game starts user can listen to the story of the game.

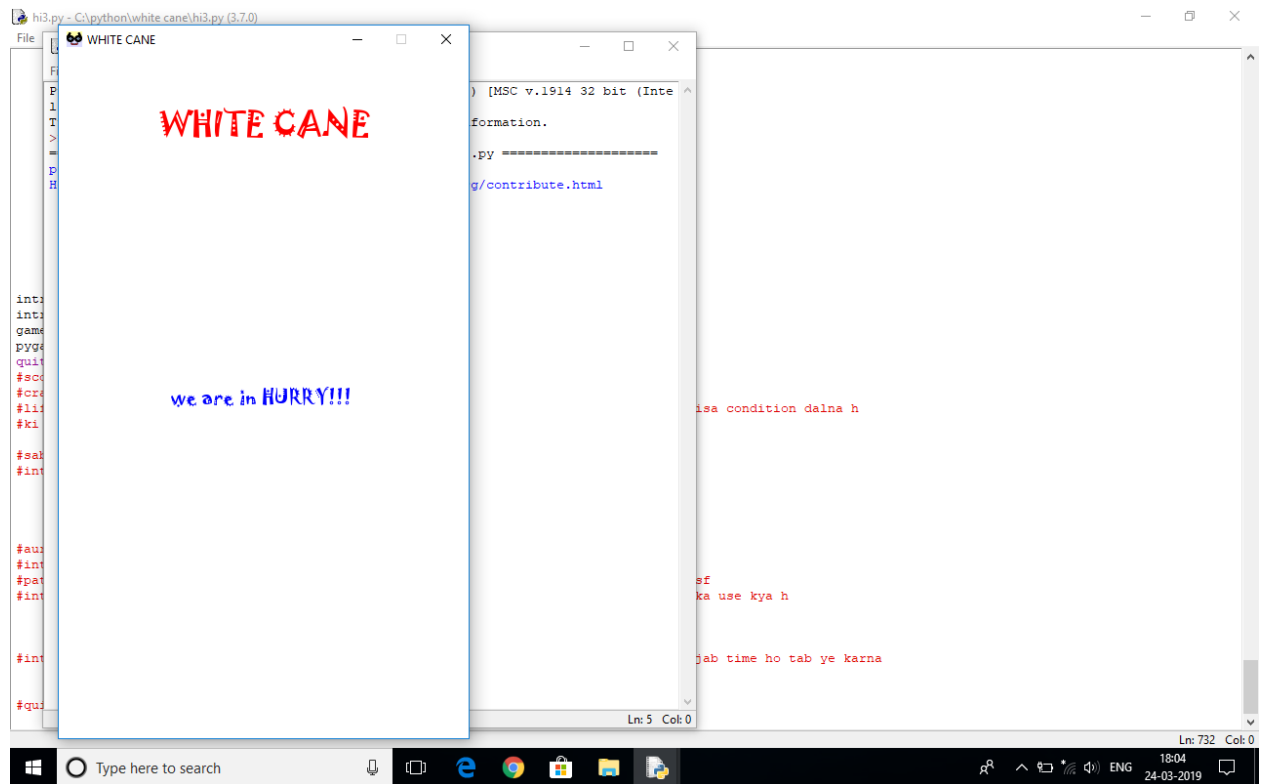


Fig 6.2 In this he listen the key which are going to be useful for the game (up , down, left, right arrow and space key)

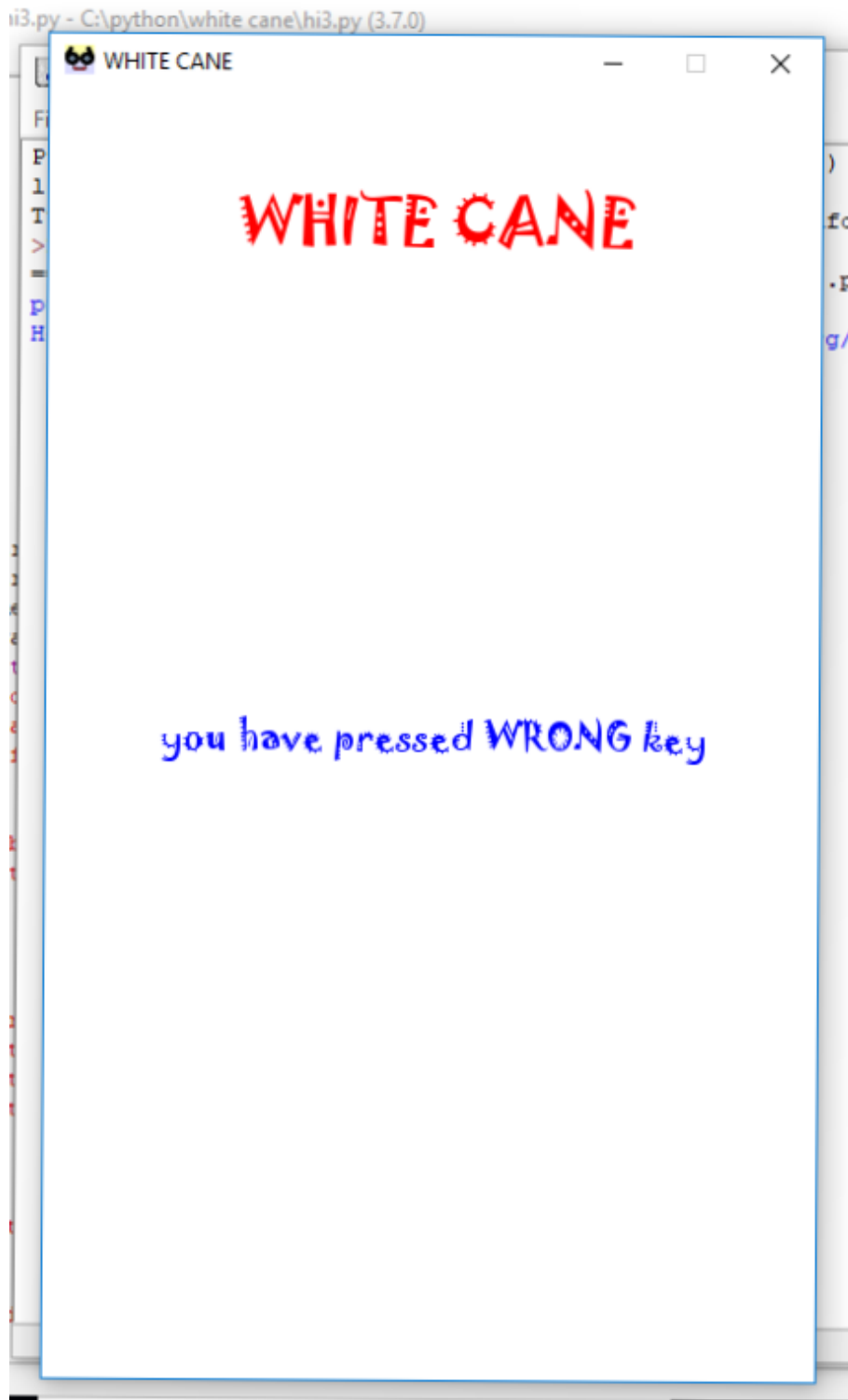


Fig 6.3 if user press any wrong key.

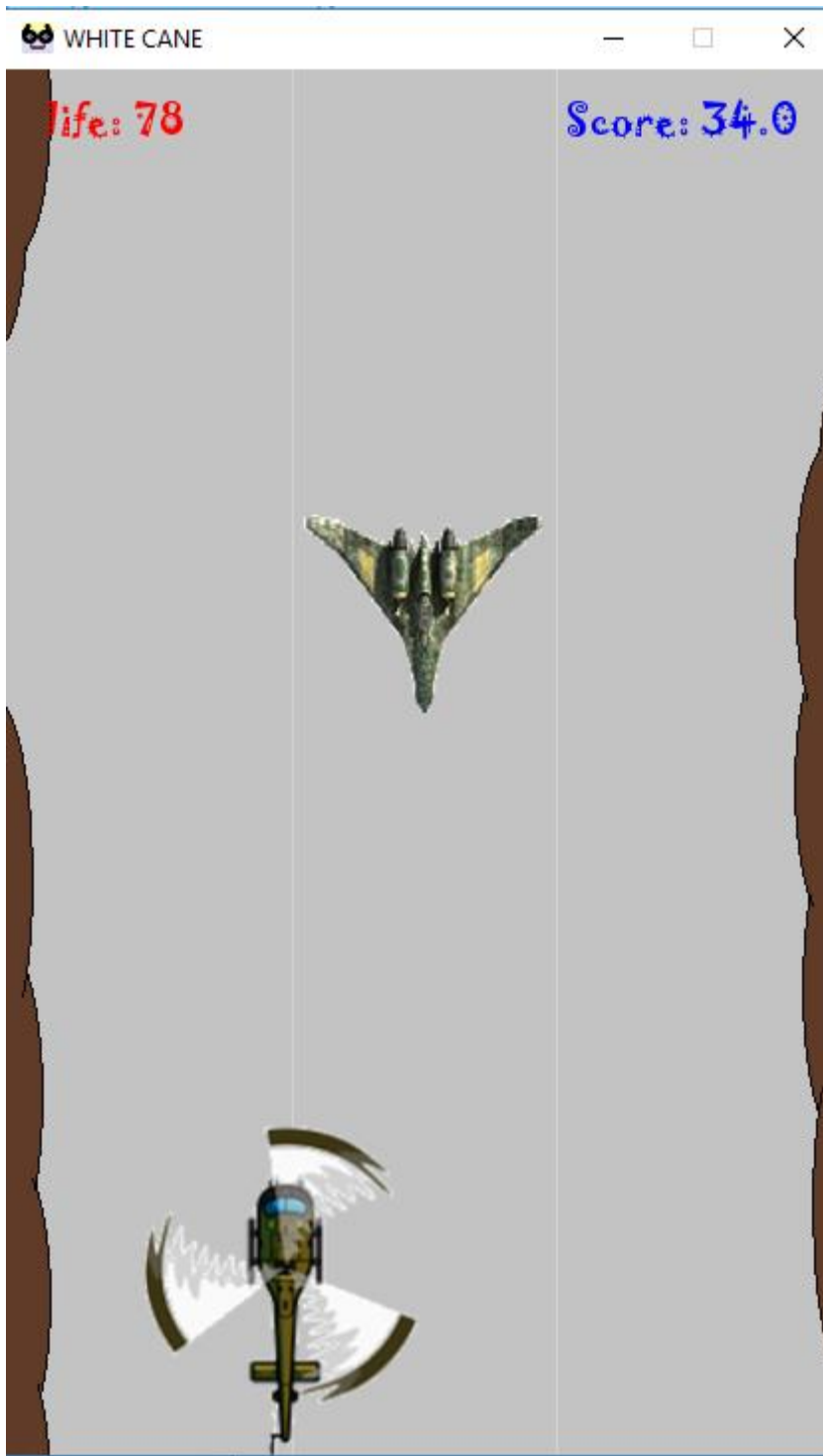


Fig 6.4 when game started.(see the life and the score)

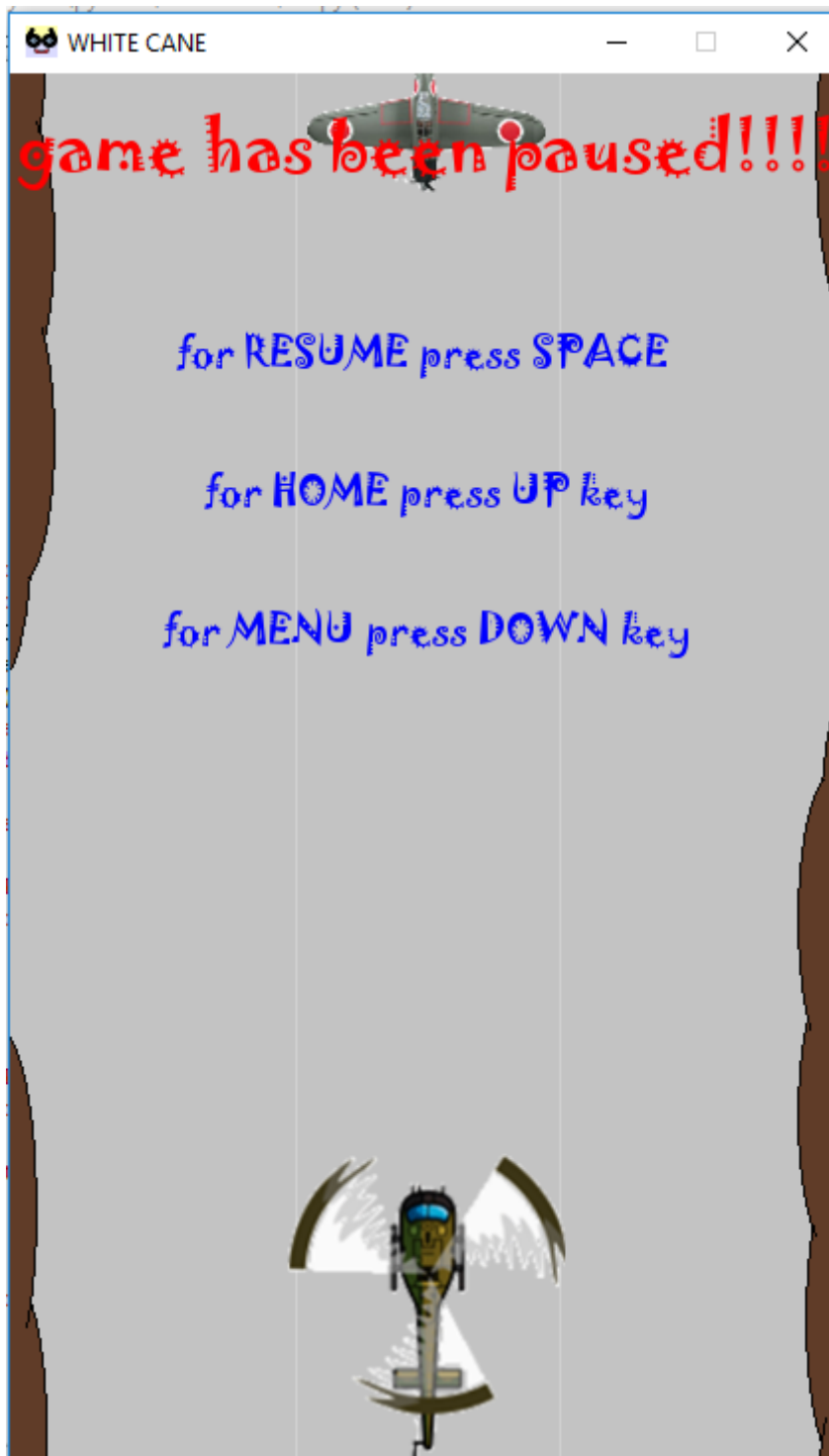


Fig 6.5 if pause event is called(by pressing space key) three option available.

WHITE CANE

** MENU **

for HOME press SPACE

to play again press UP key

to know about developers DOWN key

Fig 6.6 Menu options

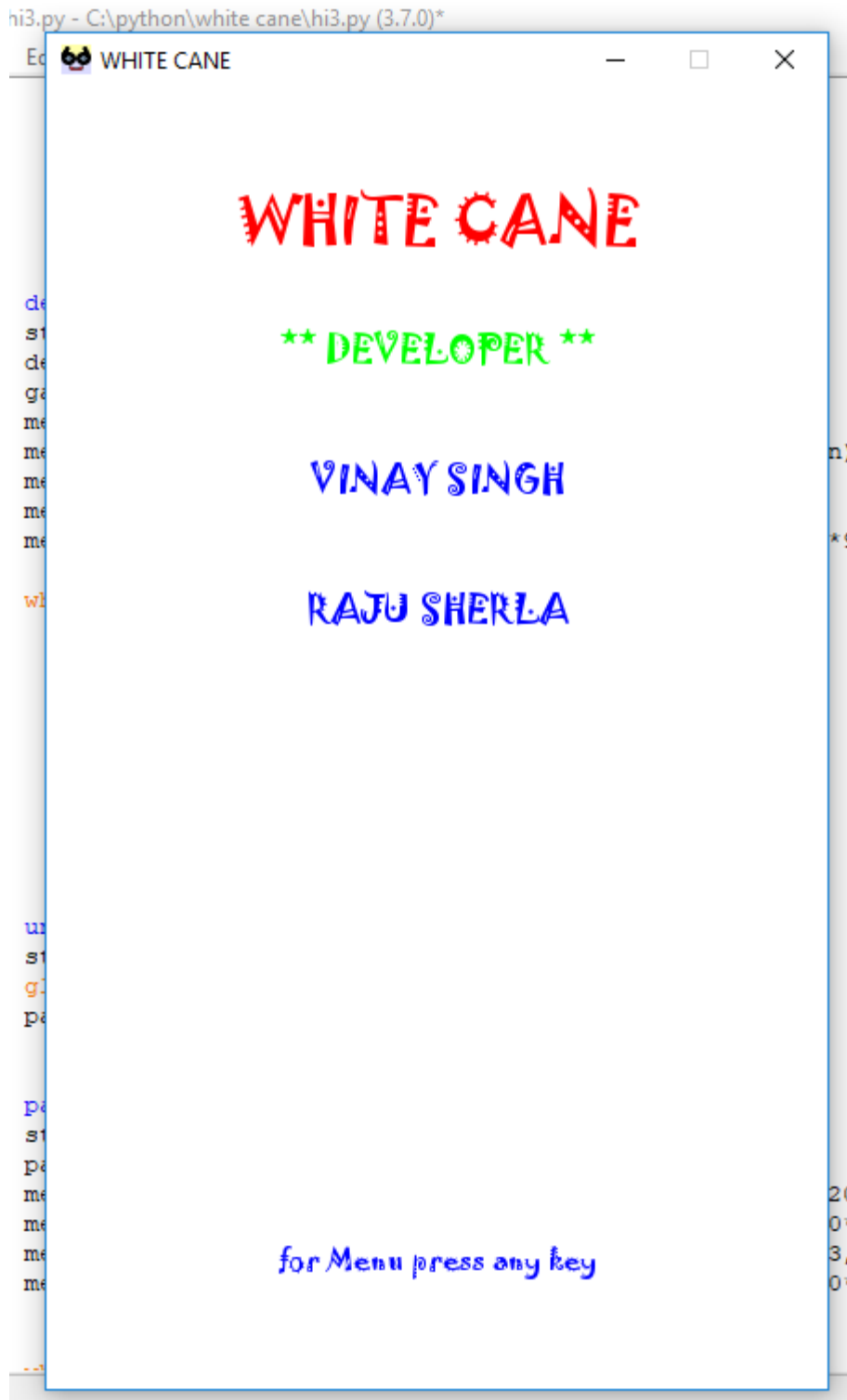


Fig 6.7 developer page player can listen the developer of the game and the college of the developer.

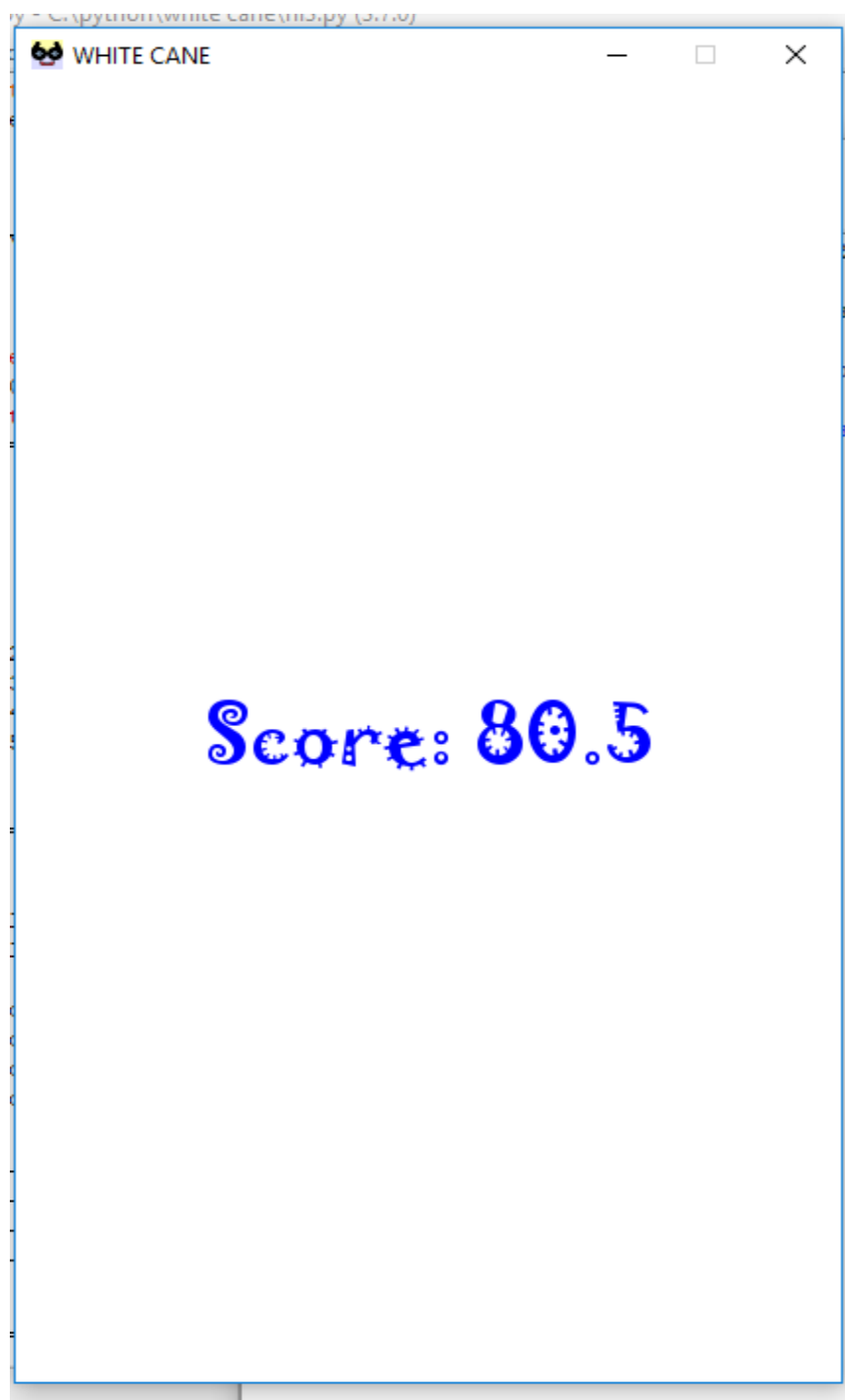


Fig 6.8 after the loss he can listen his score.



Fig 6.9 when player loss the game then he get to know his score and how well he played as per his score.

Chapter 7

Conclusion and Future Work

7.1 Conclusion & Significance of the System

The WHITE CANE is developed using python language. The project is developed under the guidance of Prof. Ms. Gauri Ansurkar. All the test cases are performed successfully with the desired positive results.

The white cane can be defined as the device used by many people who are blind or visually impaired. A white cane primarily allows its user to scan their surrounding for obstacles or orientation marks. Person can play the game without any external hardware connection. The game is portable and easy to play. To make it more interesting we have implemented a story in the game so player can relate the condition and play without any problem, even this is playable by any person.

Ultimately this application is build for blind person so he can connect to the technology and interact with machine and passing the time.

7.2 Limitations of the System

As of now the application does not save any user record just like name and high score, because we haven't connected any database to it. This game does not required any data connection which means the player can to showcase his/her character to global level.

This is build for single player only the game cannot be playable by multiplayer at a time.

7.3 Future Scope of the Project

1. Multiplayer player implementation.
2. Global connection
3. Increase User Interaction.
4. Better build in story of game.
5. Smooth game play.

7.3.1 Multilevel game

As we know the core project is built for the single player but it can get more interesting if multipliers can play the game.

We can implement the Stone Paper Scissor game. Where two player can play at a same time without any need of internet connection.

They have to react at a same time and the result will be declare via voice only.

7.3.2 Conceptual diagram

See the conceptual diagram as shown below.

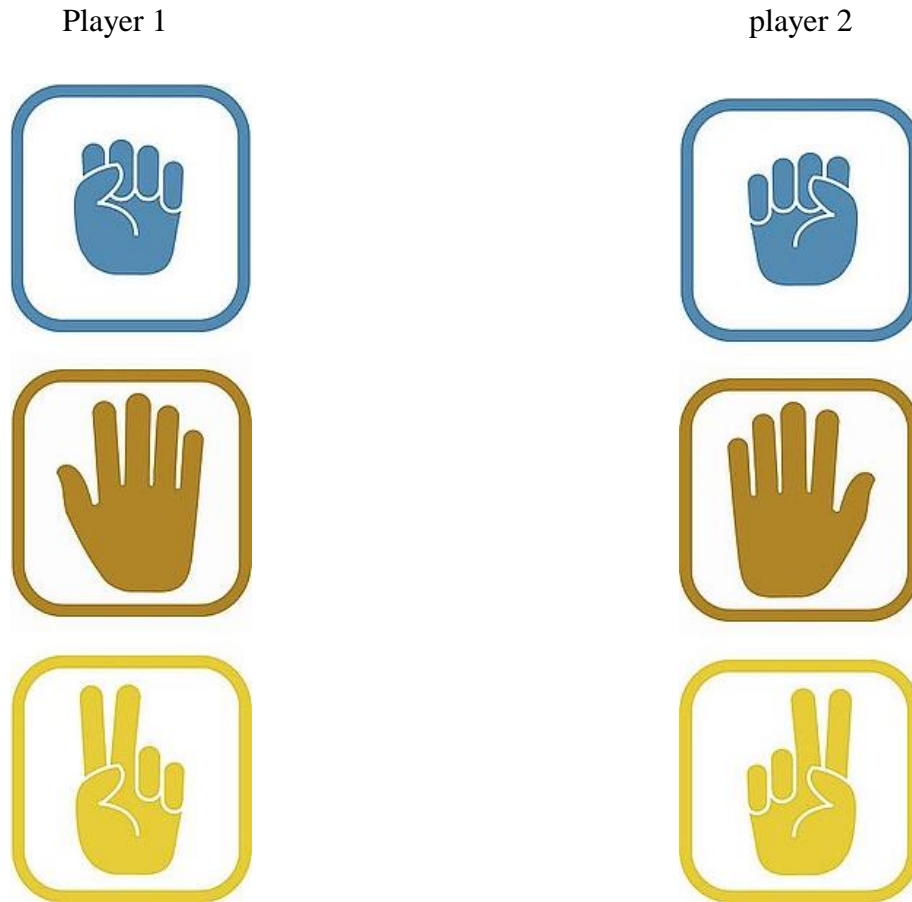


Fig 7.3 Conceptual Diagram

In this the two players will select one of the option from stone paper scissor and comparison will be done as per that the result will be declared via voice output

- 1) First option in color blue is for stone.
- 2) Second option in color brown is for paper.
- 3) Third option in color yellow is for scissor.

Chapter 8

References

8.1 Applications:

- Notepad ++
- Gimp
- Android application-
 - Pygame docs
 - Pygame tutorial and compiler
 - Sololearn
 - Voice changer
 - Voice editor
 - Youtube
 - T2S
 - Mp3 to wav converter
 - Quora etc
- Murach's python programming (book)
- IYOC games with python (pdf book)
- MS picture manager
- MS Power point
- Paint

8.2 Websites:

- Google.com
- Python.net
- Pygame.com
- Wikipedia.com
- Pythonprogramming.net
- Programiz.com , etc