

# 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 across the globe who are blind, over 15 million are from India. There are very few computer games in market for blind people. Visionless people can't entertain themselves 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.

### 1.2Objectives:

- 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 dose not needed 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

### PYTHON

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language community after 30 years.

Front-end Web Development refers to building web interfaces, specifically the parts of the website that the user will interact with. When you're browsing the web, everything you see, from images and headings to sliders and buttons is made using HTML, CSS and JavaScript, the main components to any website.

Back-end Development refers to the parts of the website that a user doesn't see or directly interact with. The back end handles application logic, algorithms, database interaction and the processing of user requests.

In this section, you will learn how to make Python Web applications more user friendly by leveraging the power of both the Front-end and Back-end. These articles cover how to integrate Back-end frameworks like Flask and Django with popular Front-end libraries and frameworks.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms.

including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open sourcesoftware and has a community-based development

model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

Back-end development can be much more varied than front-end development, which is largely driven by JavaScript, HTML, CSS, and various front-end frameworks using these languages. To simplify things, we'll break the server side down into four main components of a “software stack”: the server, the database, the operating system, and the software.

Here's some more information about each of these components of the backend.

Back-end stack's four components, whether it's on-site or in the cloud, the server acts as the lifeblood of the network. These high-powered computers provide shared resources that networks need to run, including file storage, security and encryption, databases, email, and web services. Learn all about types of servers, what they do, and how they play into the back-end stack with our guide to survey of technology.

Once you've gotten the basics down about on-site servers, expand your knowledge with a look at virtualization, how servers get provisioned to house multiple apps, and containerization another way servers provision their operating systems out to house compartmentalized applications.

## 2.3 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	

# **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 whole 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 Nonfunctional 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**

Planning and scheduling is a complicated part of software development.

**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.

**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.

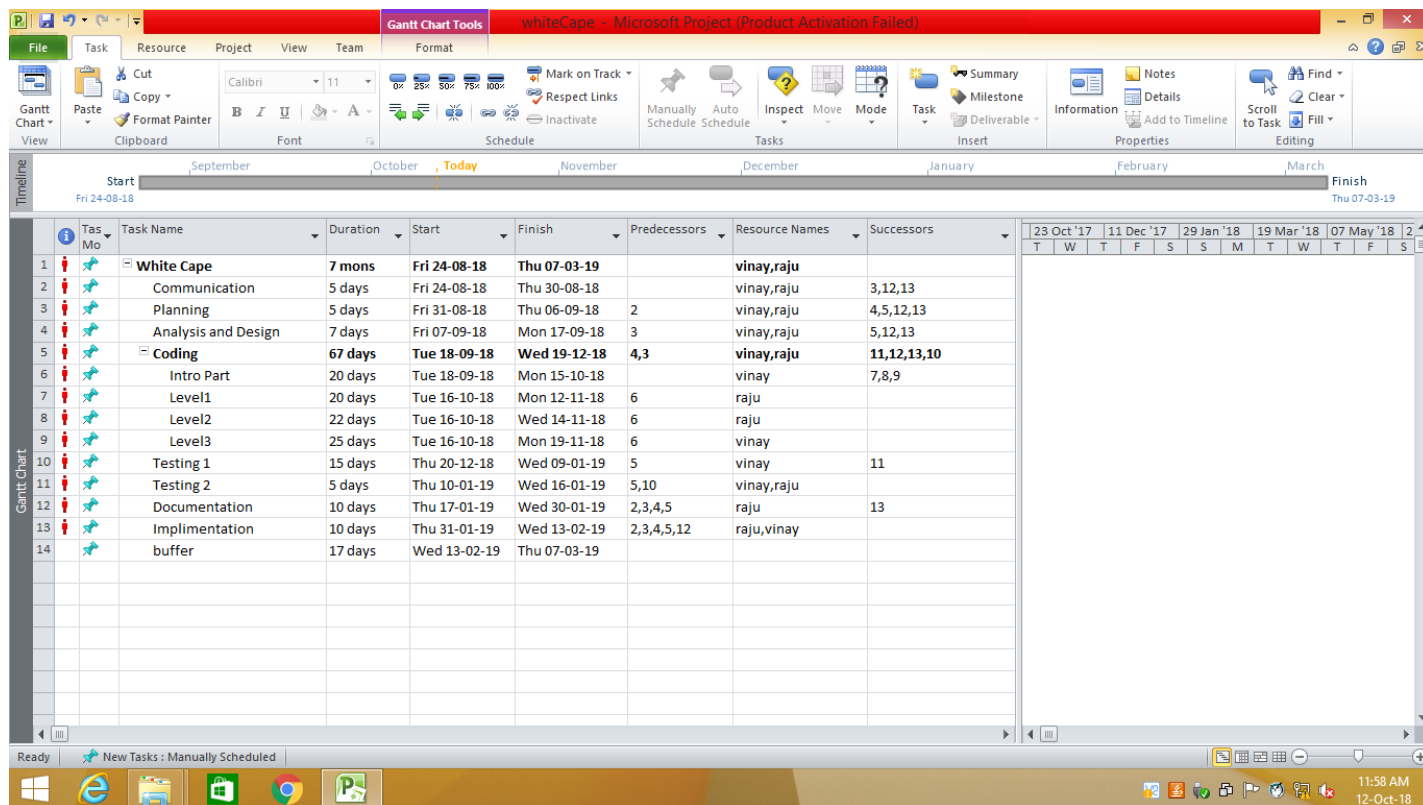
# Planning and Scheduling

## (gantt chart and network diagram)

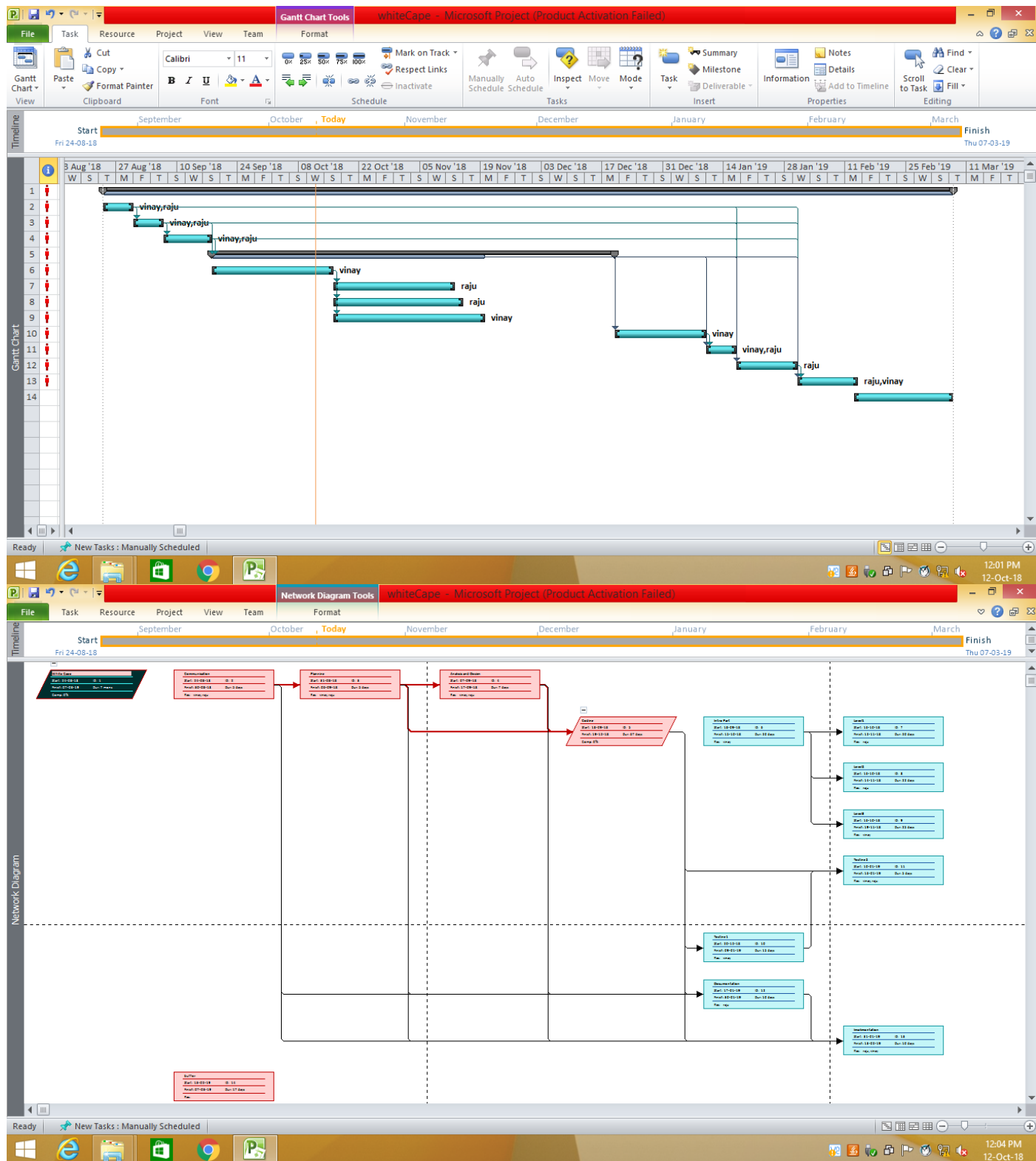
Time evaluation is the most important consideration in the development of project. The time schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development of other systems.

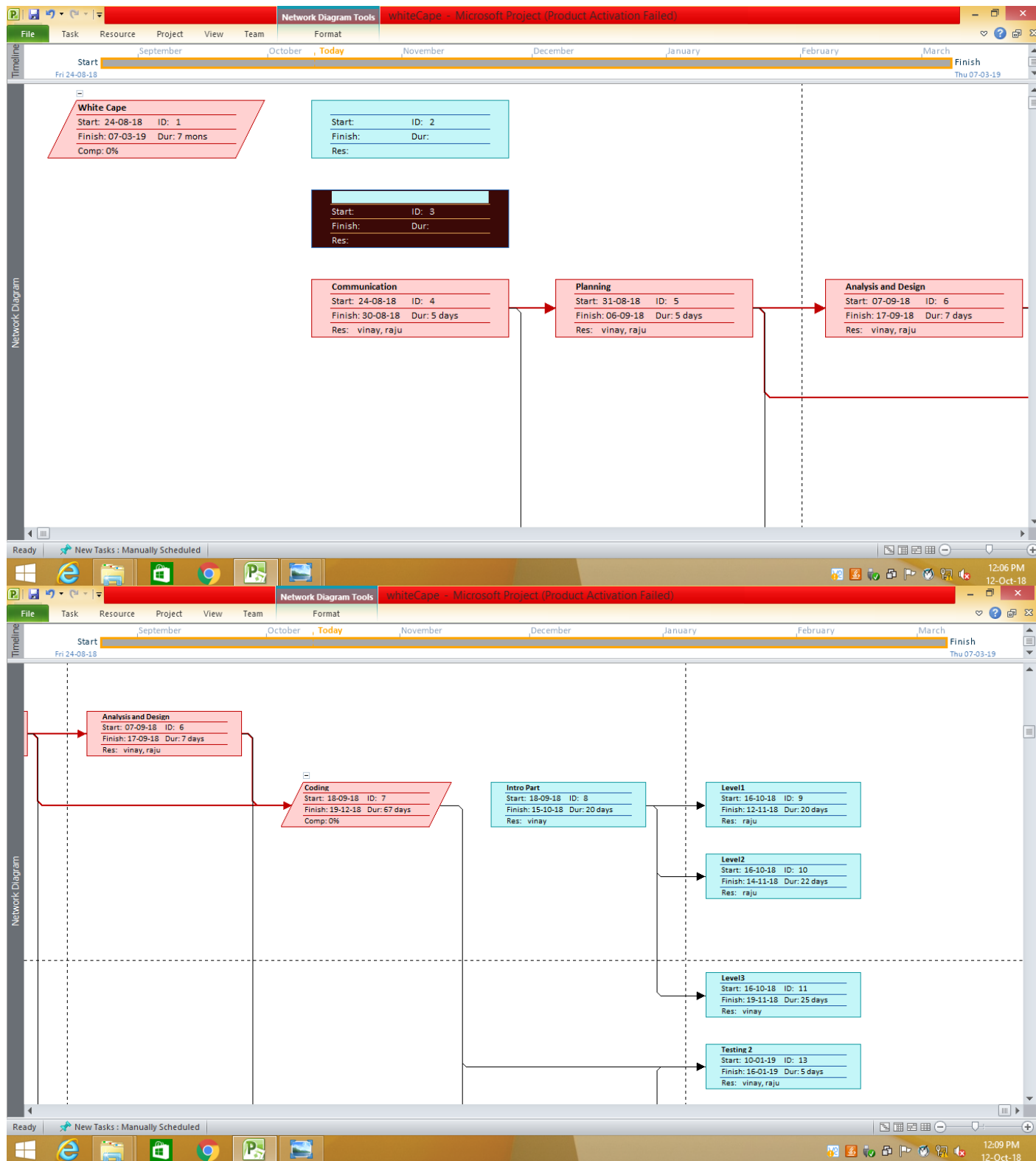
Our project need time management. Hence gantt chart is very important factor for us.

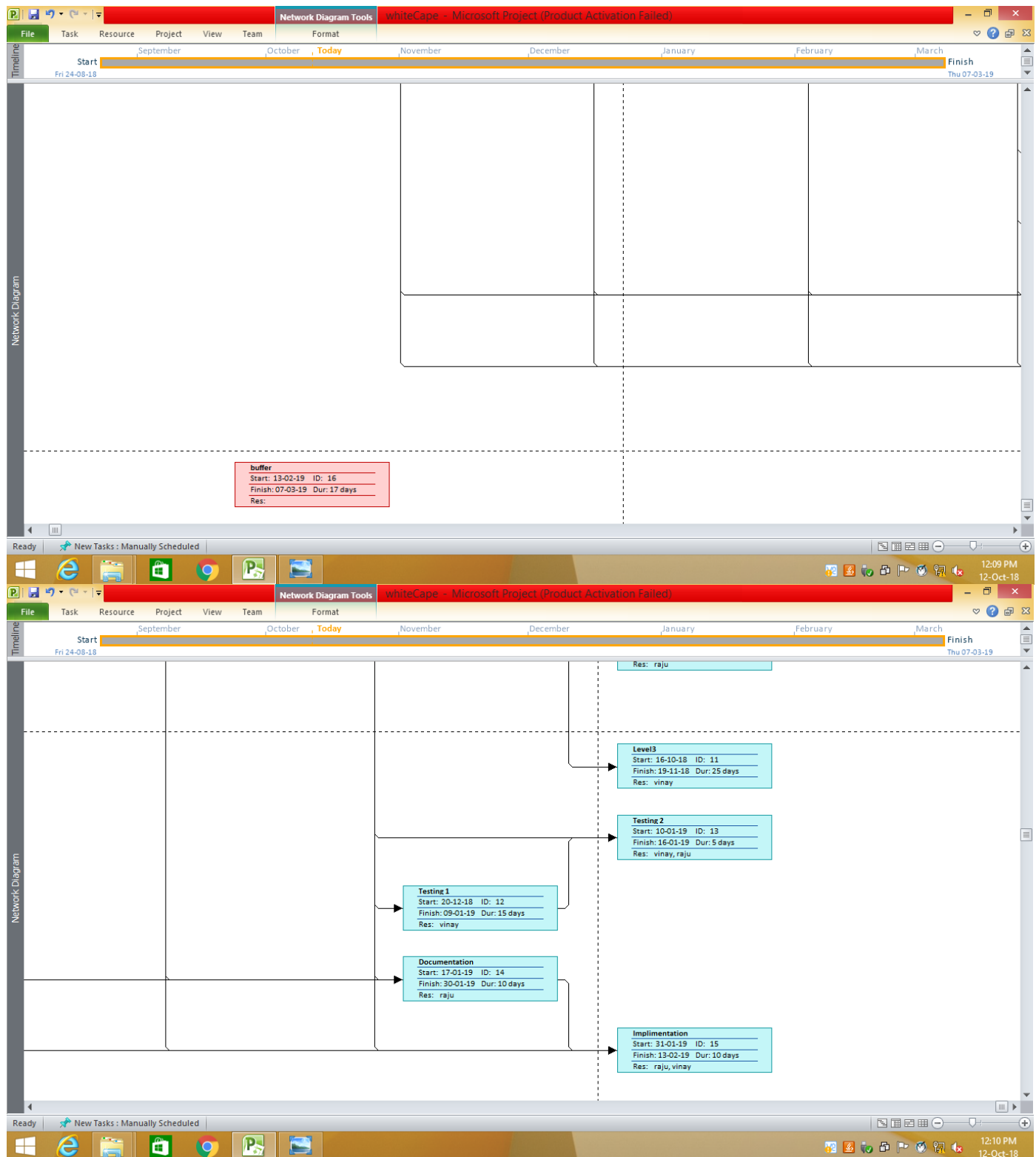
This are some screenshots of gantt chart.











### 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 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAM
- Intel® Xeon® processor E5-2698 v3 at 2.30 GHz (2 sockets, 16 cores each, 1 thread per core), 64 GB of DRAM
- Intel® Xeon Phi™ processor 7210 at 1.30 GHz (1 socket, 64 cores, 4 threads per core), 32 GB of DRAM, 16 GB of MCDRAM (flat mode enabled) Disk space: 2 to 3 GB
- Operating systems: Windows® 10, macOS\*, and Linux

## **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.

### **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.

### **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.

### **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.

## **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) Introduction
- 2) Level 1(easy)
- 3) Level 2 (normal)
- 4) Level 3 (hard)
- 5) Credits
- 6) Exit game

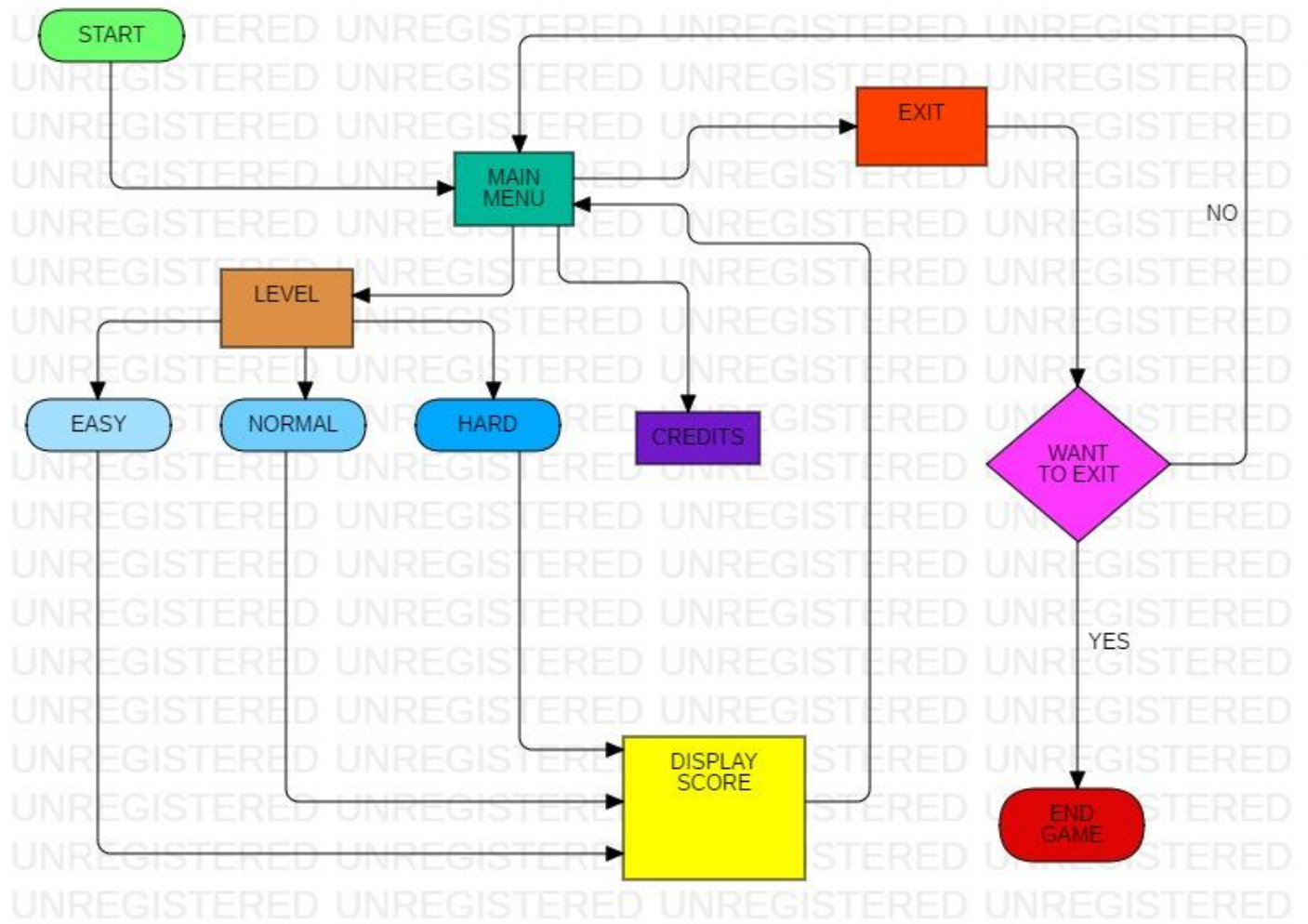
### **4.2 Procedural Design**

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**

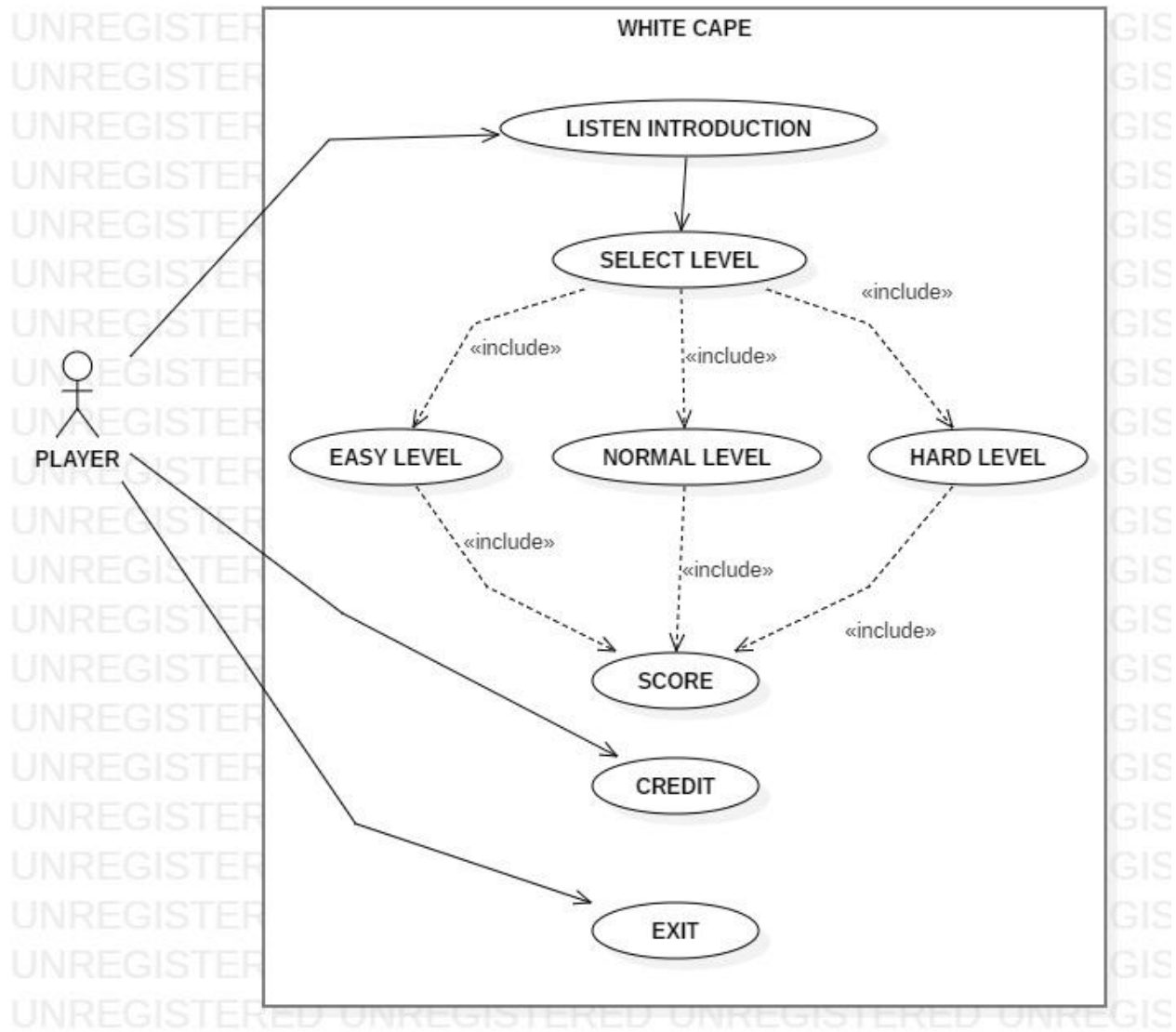
We will be creating some UML (Unified Modeling Language) Diagrams to define the systematic flow of procedure that improves its comprehension and will help us during implementation. For example, Use Case Diagram, State Diagram.

## Flowchart diagram

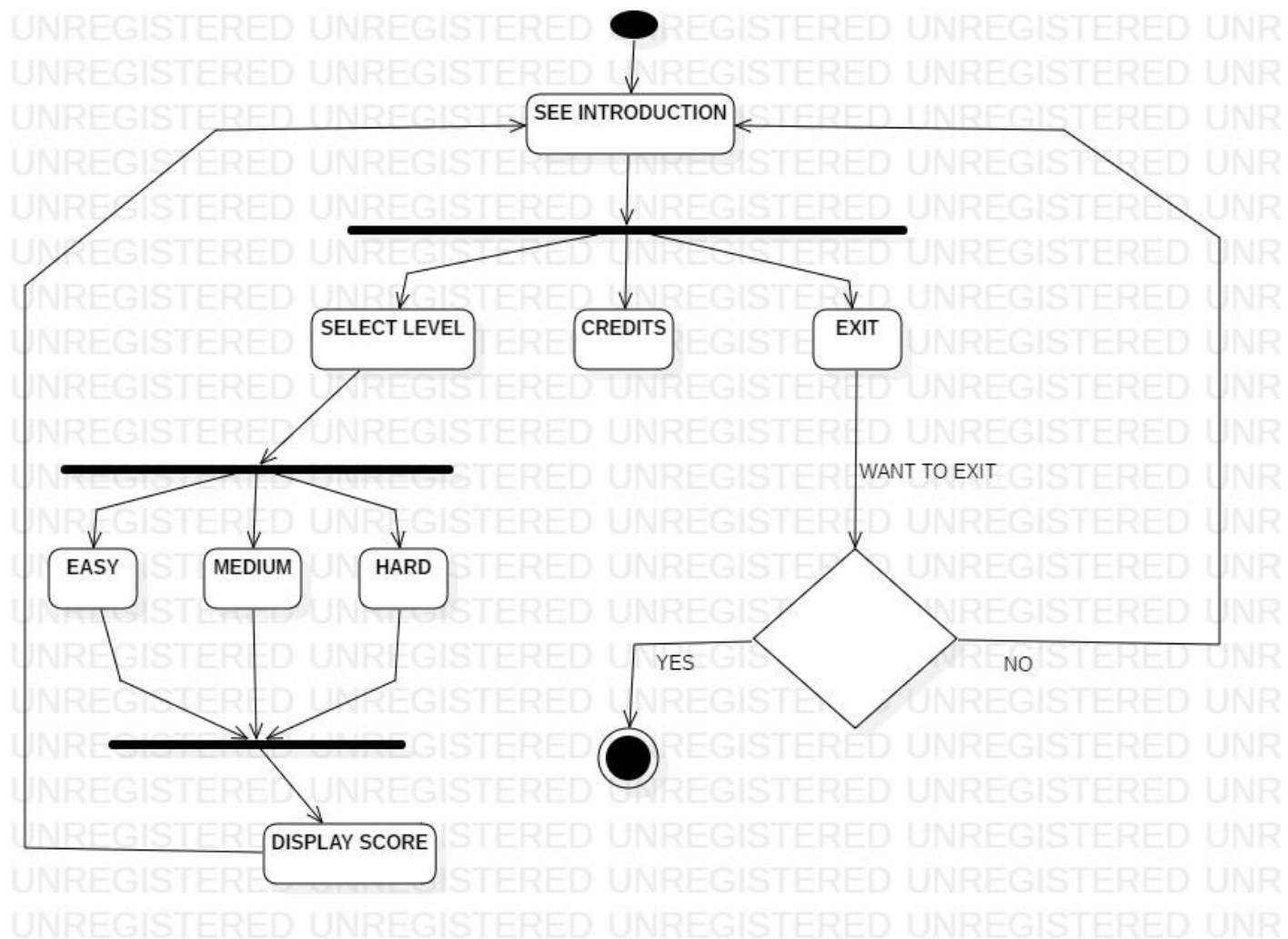




## USECASE diagram



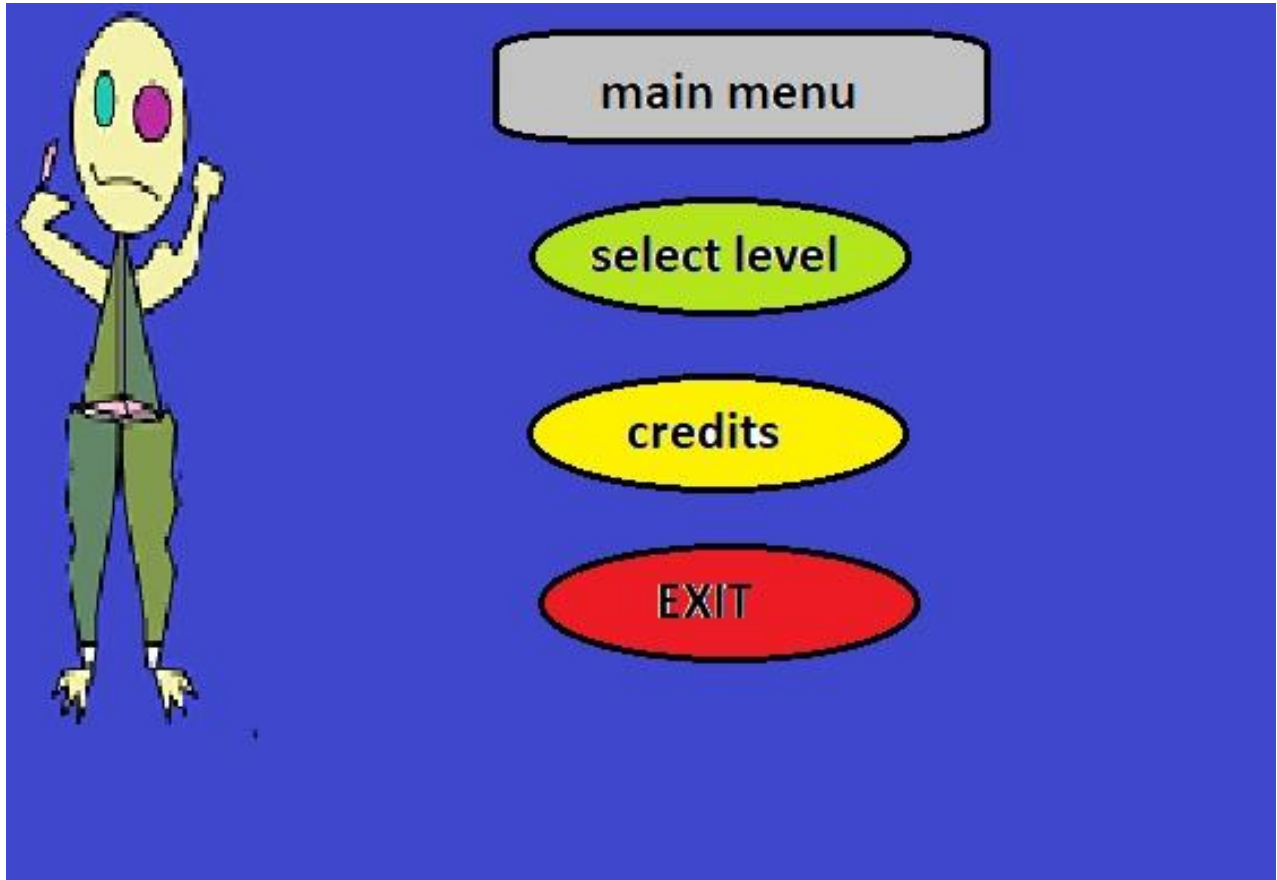
## Activity diagram



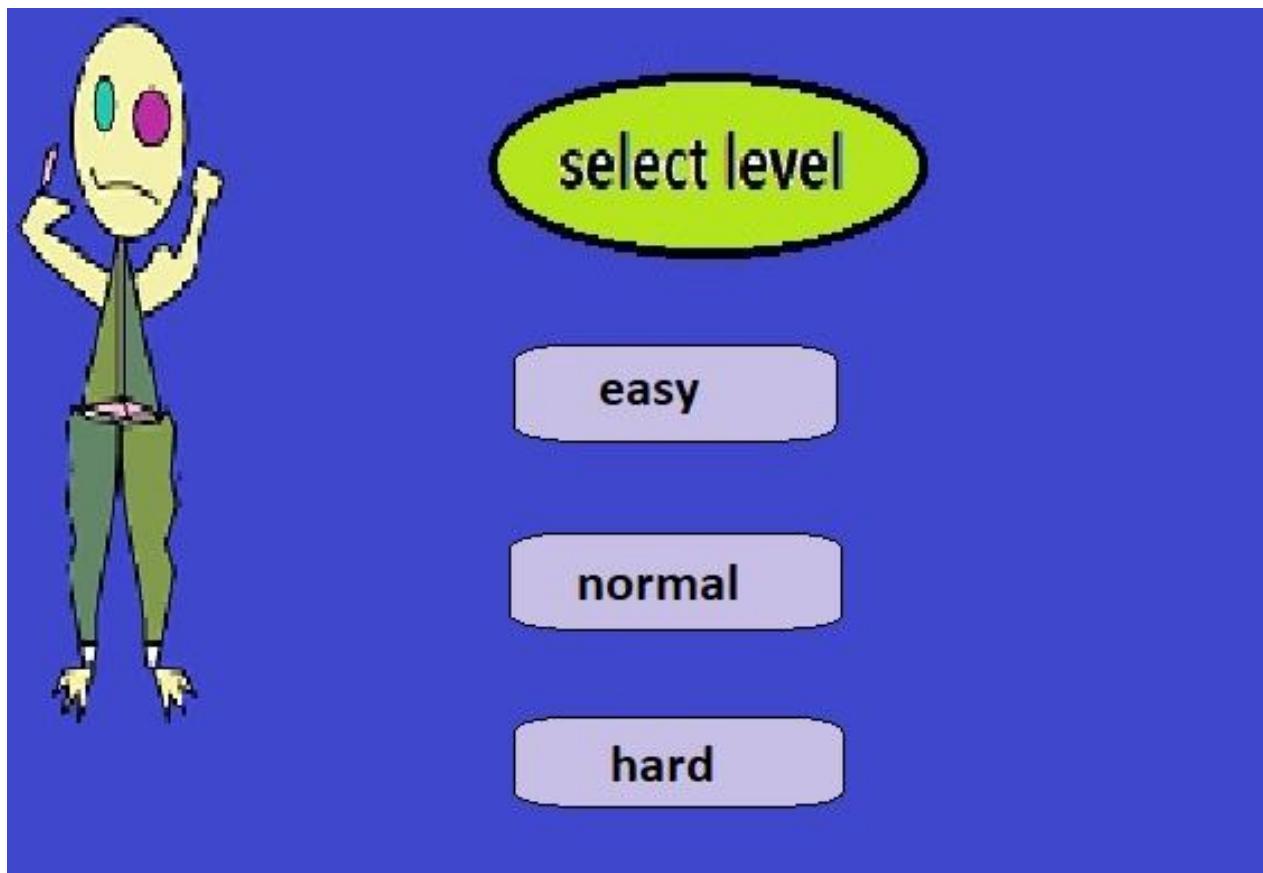
## 4.4 Application interface:

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

- **Main menu view**



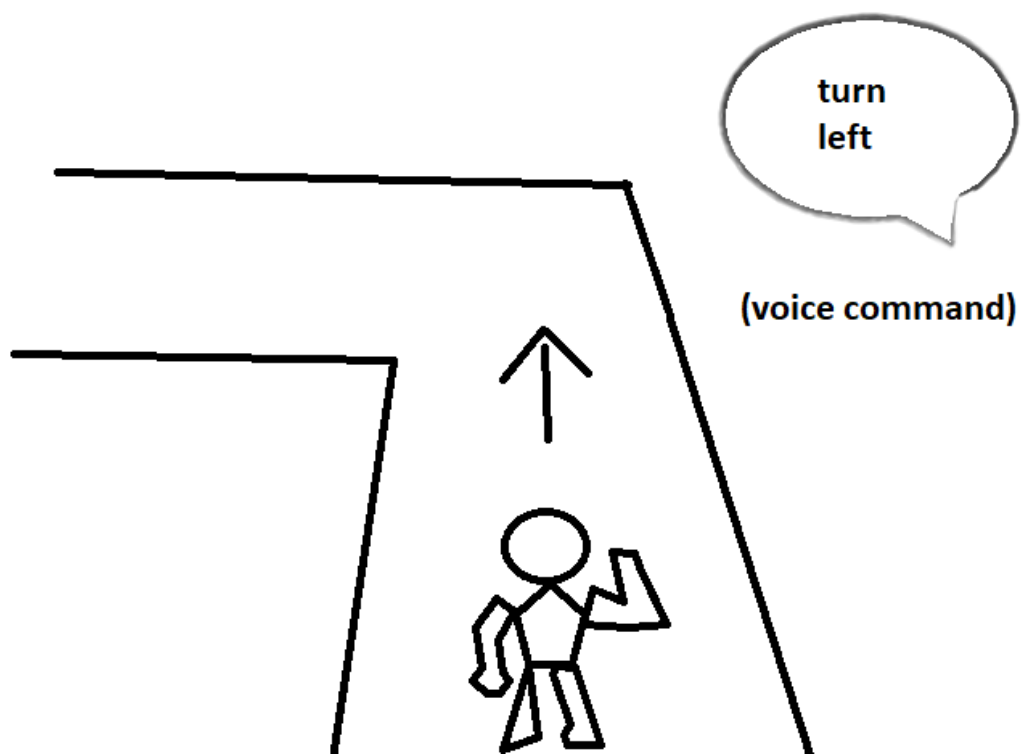
- **Select level view**



## 4.5 Graphical representation of gaming interphase:

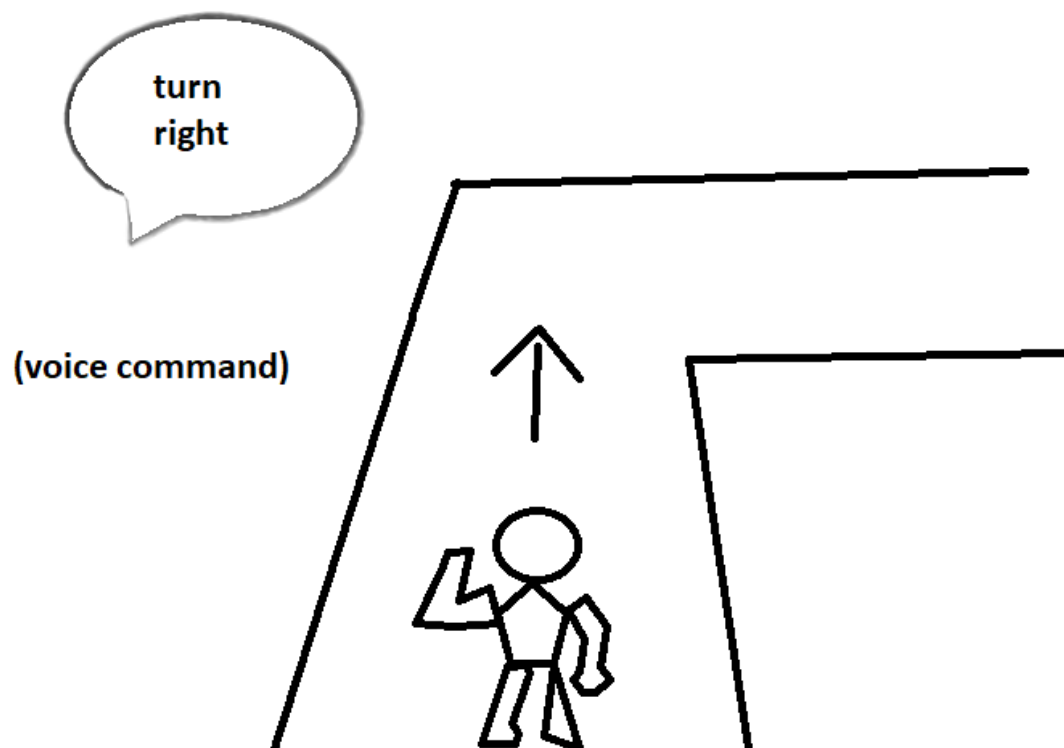
### First Event

Situation where player have to take a turn to left hand side.



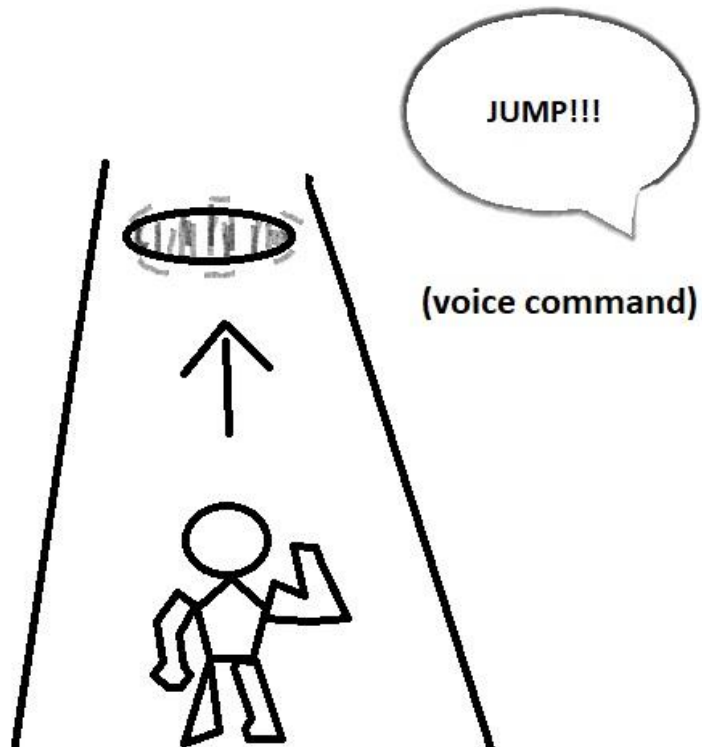
## Second Event

Situation where player have to take a turn to right hand side.



### Third Event

Situation where player have to jump.



# **Chapter 5**

## **Future enhancement**

In future enhancements we can discuss the possible up gradation of project. This can be implemented as per the need of the project.

### **5.1 Integration and System Testing**

In this report I will be writing about integration and system testing which is a type of testing which is conducted in the software development process. I will start off by writing about integration testing.

### **5.2 Introduction**

Integration and system testing is a type of software testing, this makes sure that tests such as the system and integration are done before releasing the product. Software testing has very strict set of rules and guidelines that it follows to make sure each individual part of the software is thoroughly checked before it is given the OK, this makes sure that there are no errors and that the software runs how it's supposed to. Integration and system testing is mainly done by a team who focuses only on the software testing phase in the system development life cycle.

In software testing each testing level build on from the previous level so it is important that the testing is done in the correct order, access to the information is passed on to the next level.

### **5.3 Integration**

Integration testing in the software testing model comes before system testing and after the unit testing has been done.

The way that integration testing works is by, getting the individual modules that have been through the unit testing phase and integrating each module into a group. The integration testing phase will make sure when the modules are being integrated together that any problems, for example errors or bugs, caused due to the integration of the modules are eliminated.



Integration testing does not deal with the integration of the whole system but deals with the integration of a process in the system.

In the integration testing stage there are three things that are created, to ensure that the integration of the modules is successful and that it runs successfully as well, a test plan, test cases and test data is produced to effectively test that the integration is successful. Test data is normally used by test cases but I have mentioned each type below;

## **5.4 Integration Test Plan**

When producing a test plan it must include the following information to be effective,

- 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.

Above are just some of the most important information that is in the test plan but as a test plan is simply planning there is more information and points that can be included in the test plan by a company.

Most test plans are approved and worked on with the client so they may order some changes later on, so a test plan may have to include more information and it is best to get this approved so no problems are encountered later on.

## **5.5 Critical Path Method (CPM)**

Takes the following as input.

1. A schedule priority chart.
2. Estimates of how long each task in the schedule priority chart will take.
3. An assignment of each task to a programmer.
4. An ordering on the tasks assigned to each programmer.

**Computes the following.**

1. Earliest start time of each task.
2. Earliest completion time for all tasks.
3. Slack time, the amount of time starting on a task can be delayed without affecting the earliest completion date.

## **5.6 Multilevel game**

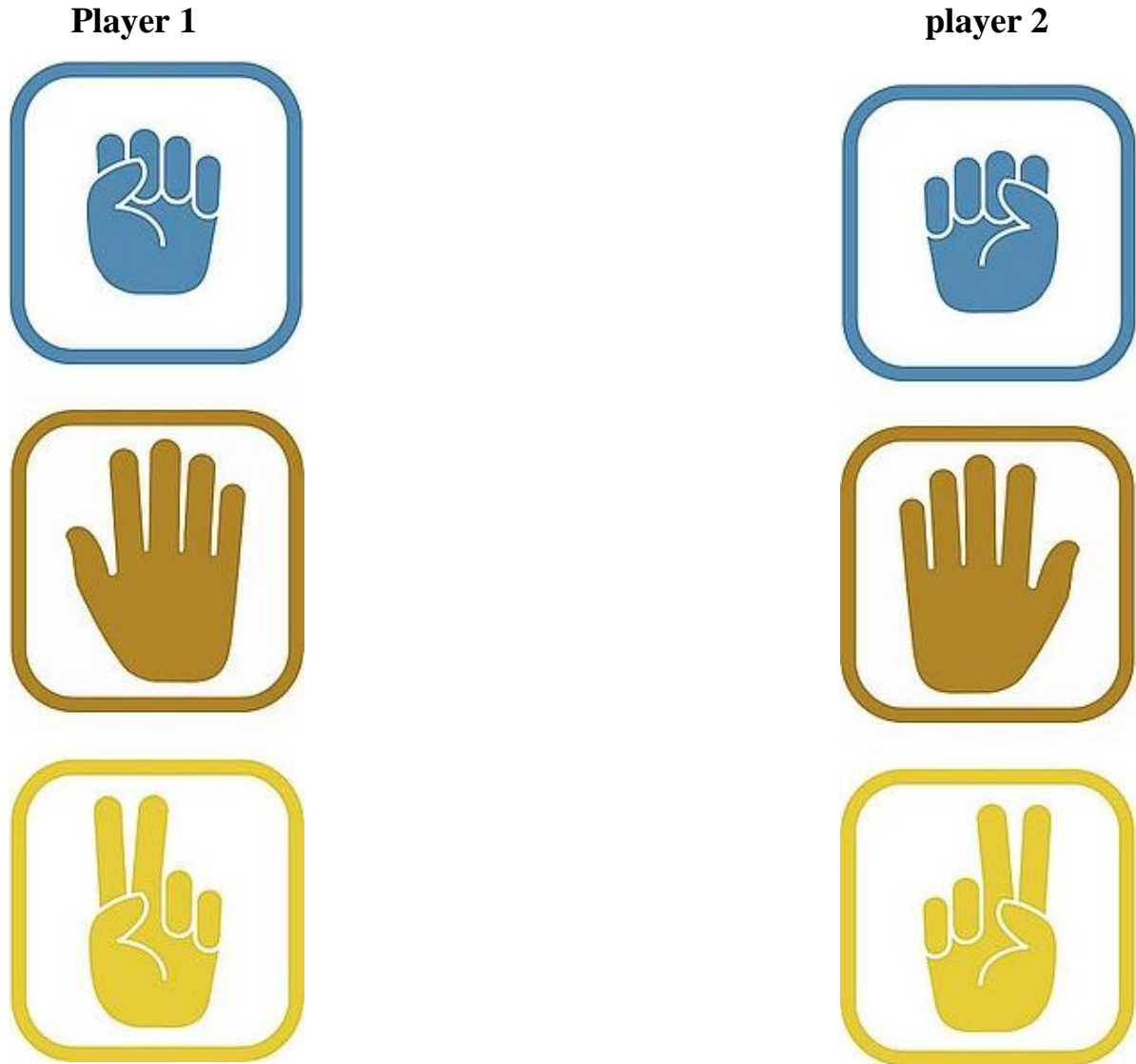
As we know the core project is built for the single player but it can get more interesting if multiplayer can play the game.

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

## 5.7 Conceptual diagram

See the conceptual diagram as shown below.



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

First option in color blue is for stone.

Second option in color brown is for paper.

Third option in color yellow is for scissor.

## 5. IMPLEMENTATION AND TESTING

### 5.1 Implementation Approaches.

The application is developed using python language and pygame library.

The application is based on verbal interaction with player.

Graphical User Interface (GUI) – As we know this is been developed for blind people so we have implemented simple GUI using python only. But then too it is beautiful friendly interface.

Application does not save any user data so we haven't connected and data base.

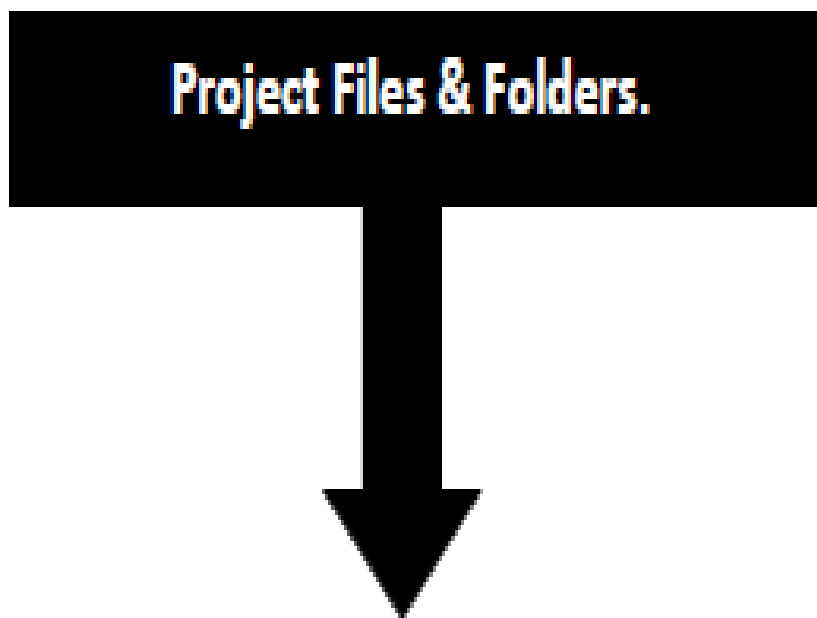
### 5.2 Coding Details & Code Efficiency.

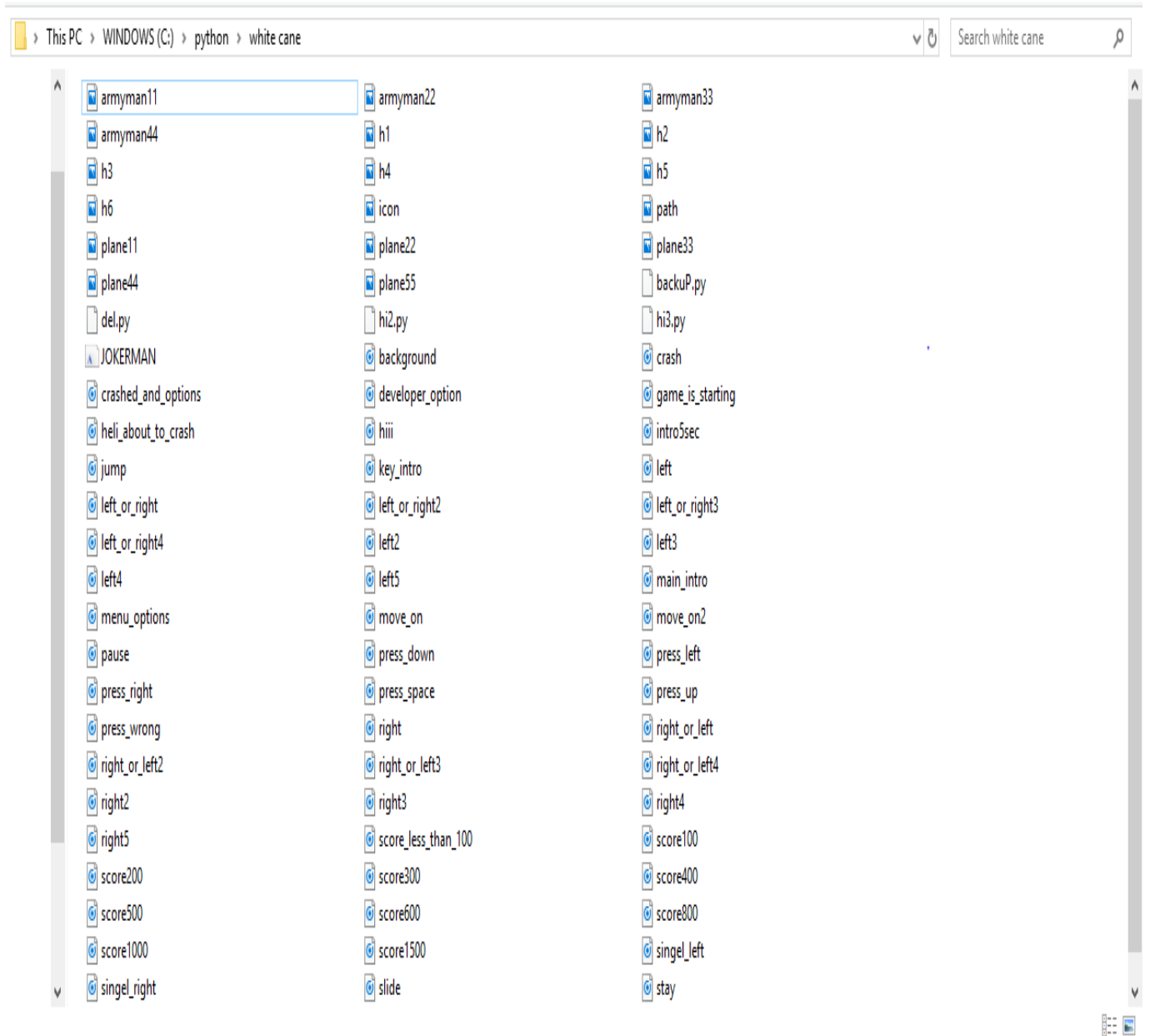
The application uses simple logic and it has more than 70 files which include backup, code, image, audio, icon, font, etc.

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 6.1 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]

## 6. RESULTS AND DISCUSSION

### User Interface Output:

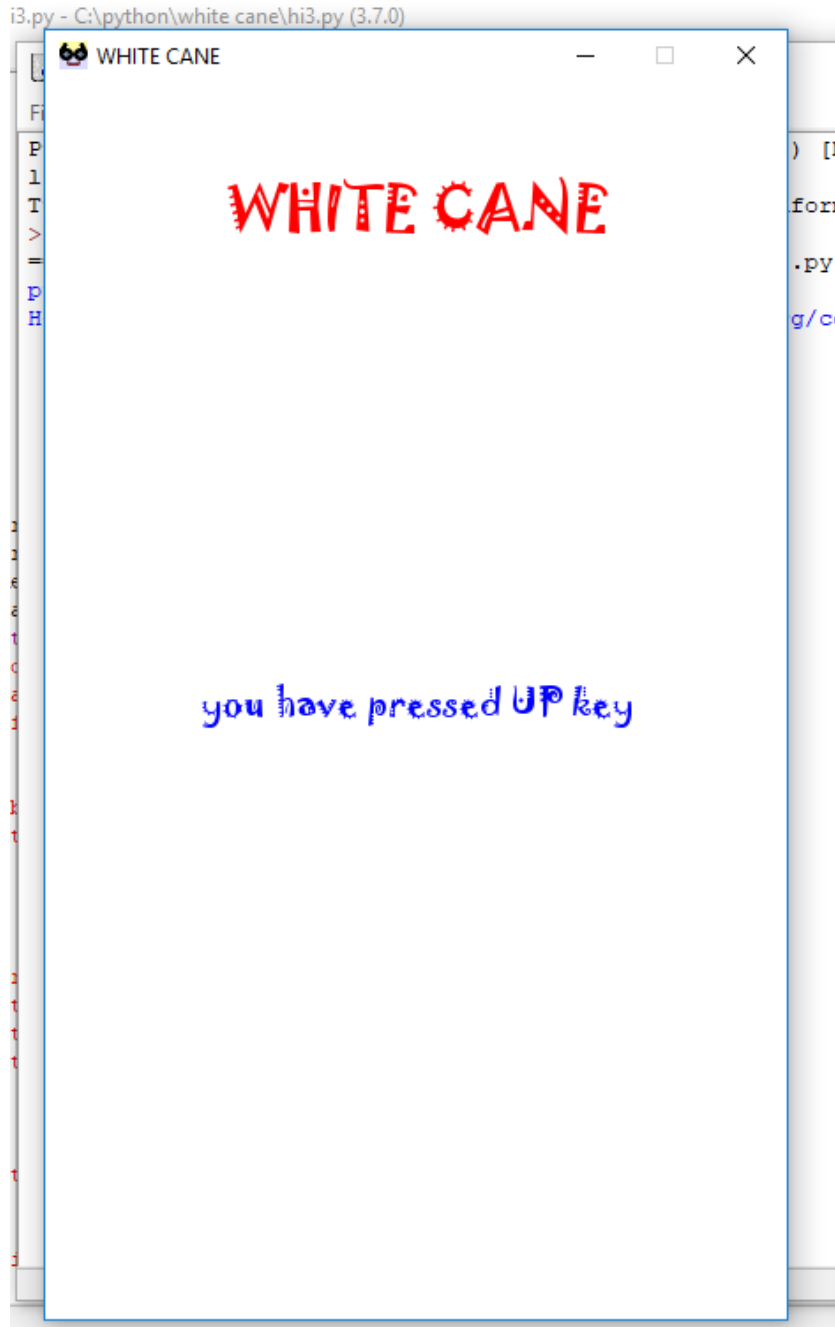


Fig 6.1 when game starts user can listen to the story of the game.

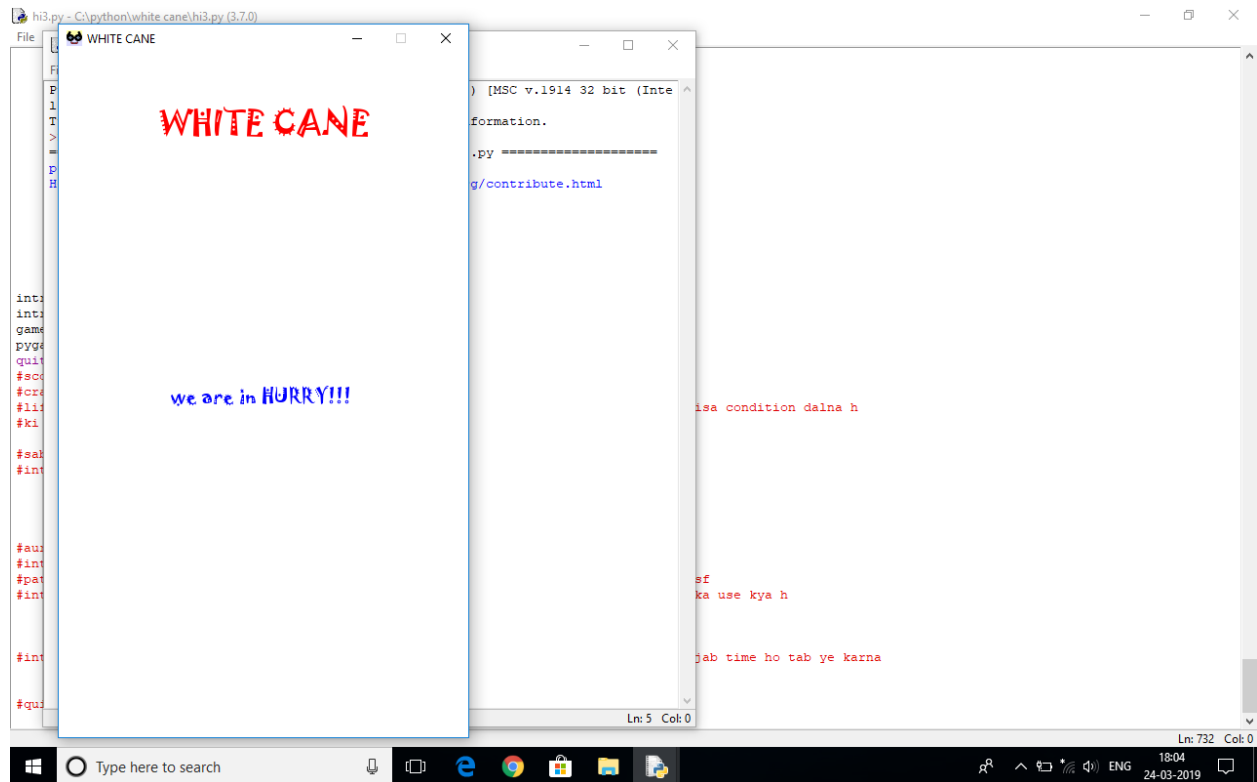


Fig 6.2 In this he listen the key which are going to be useful for the game (up , down, left, right arrow and space key)



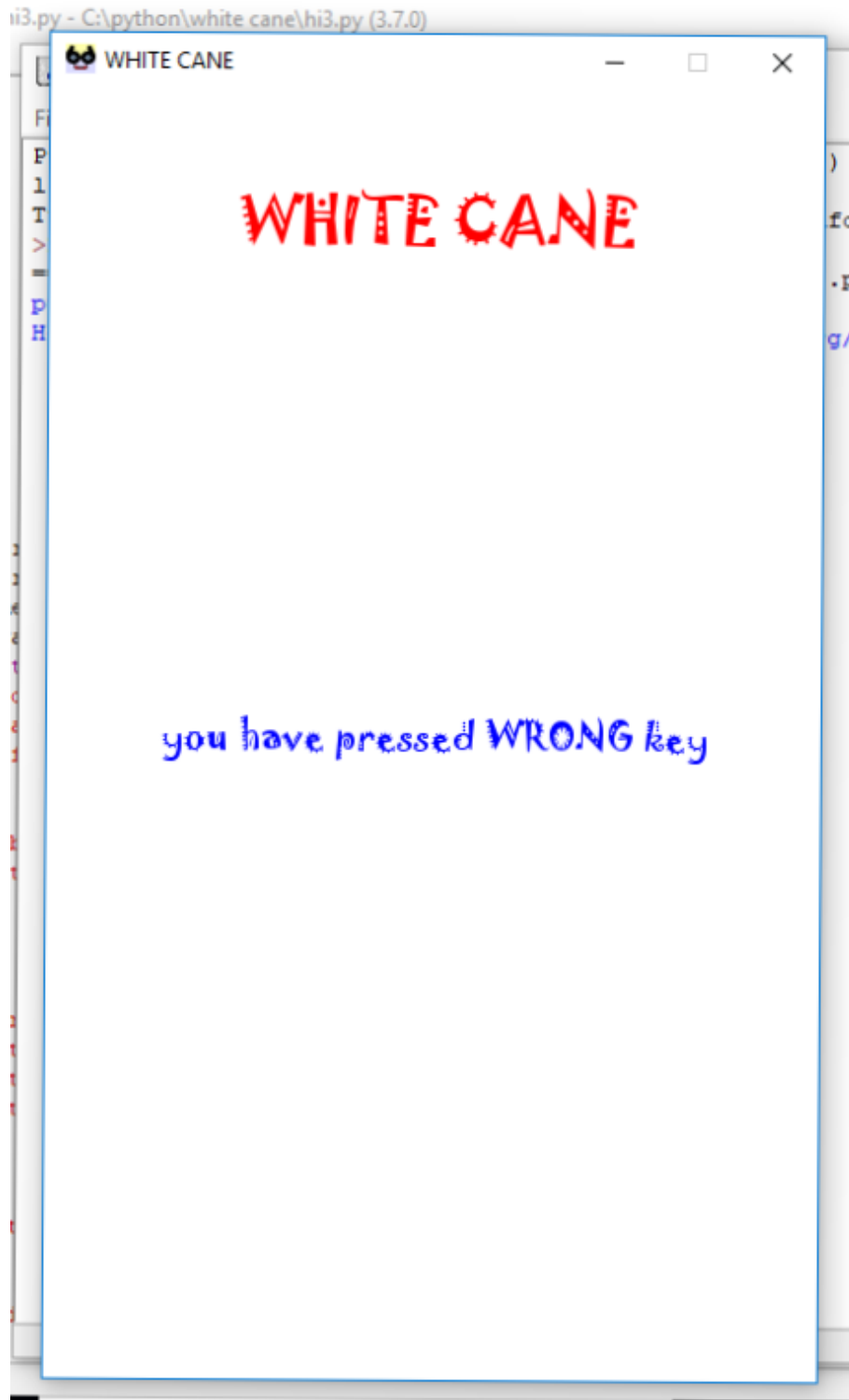


Fig 6.3 if user press any wrong key.



Fig 3.4 when game started.(see the life and the score)

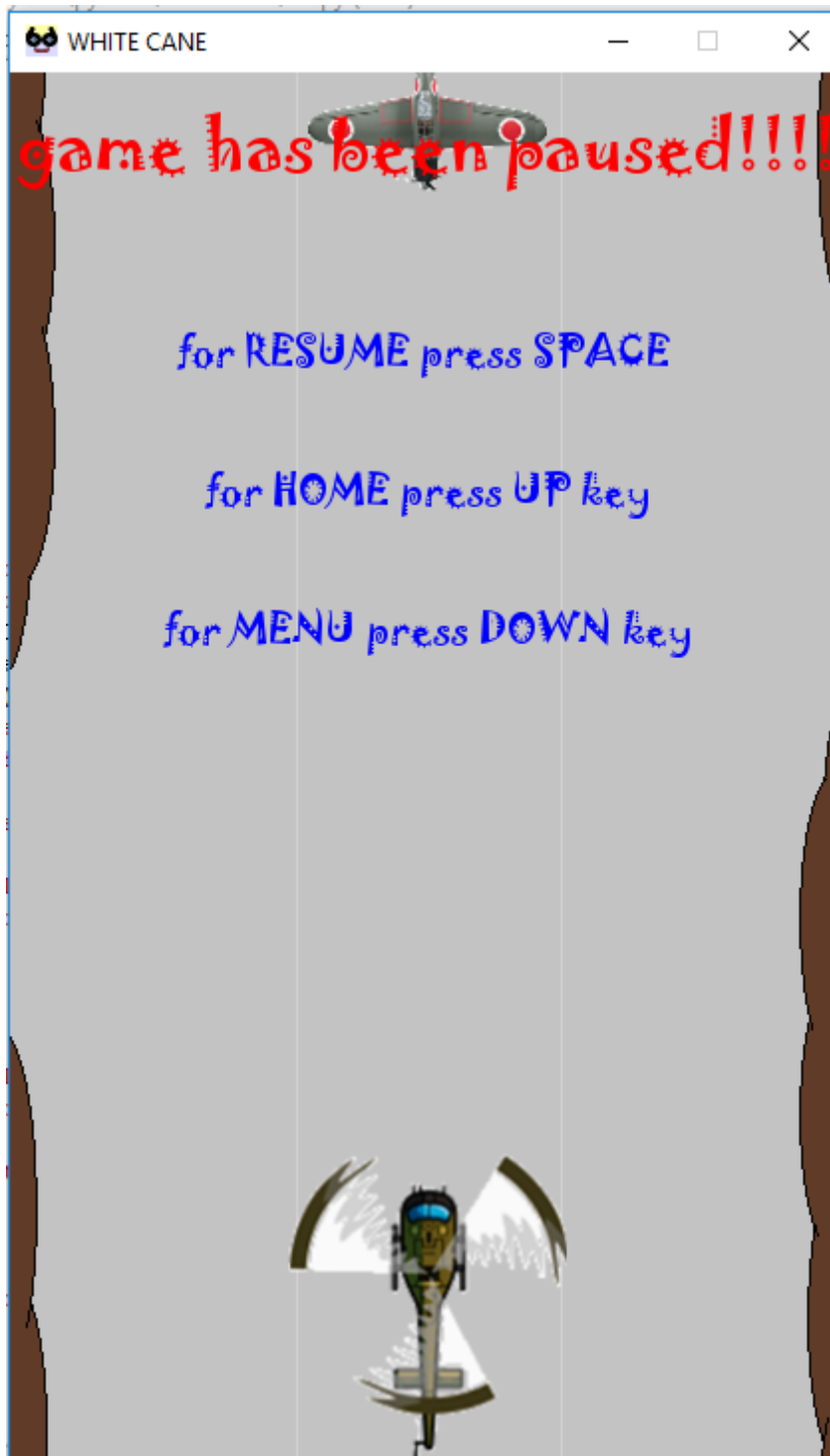


Fig 6.5 if pause event is called(by pressing space key) three option available.

WHITE CANE

\*\* MENU \*\*

for HOME press SPACE

to play again press UP key

to know about developers DOWN key

Fig 6.6 Menu options

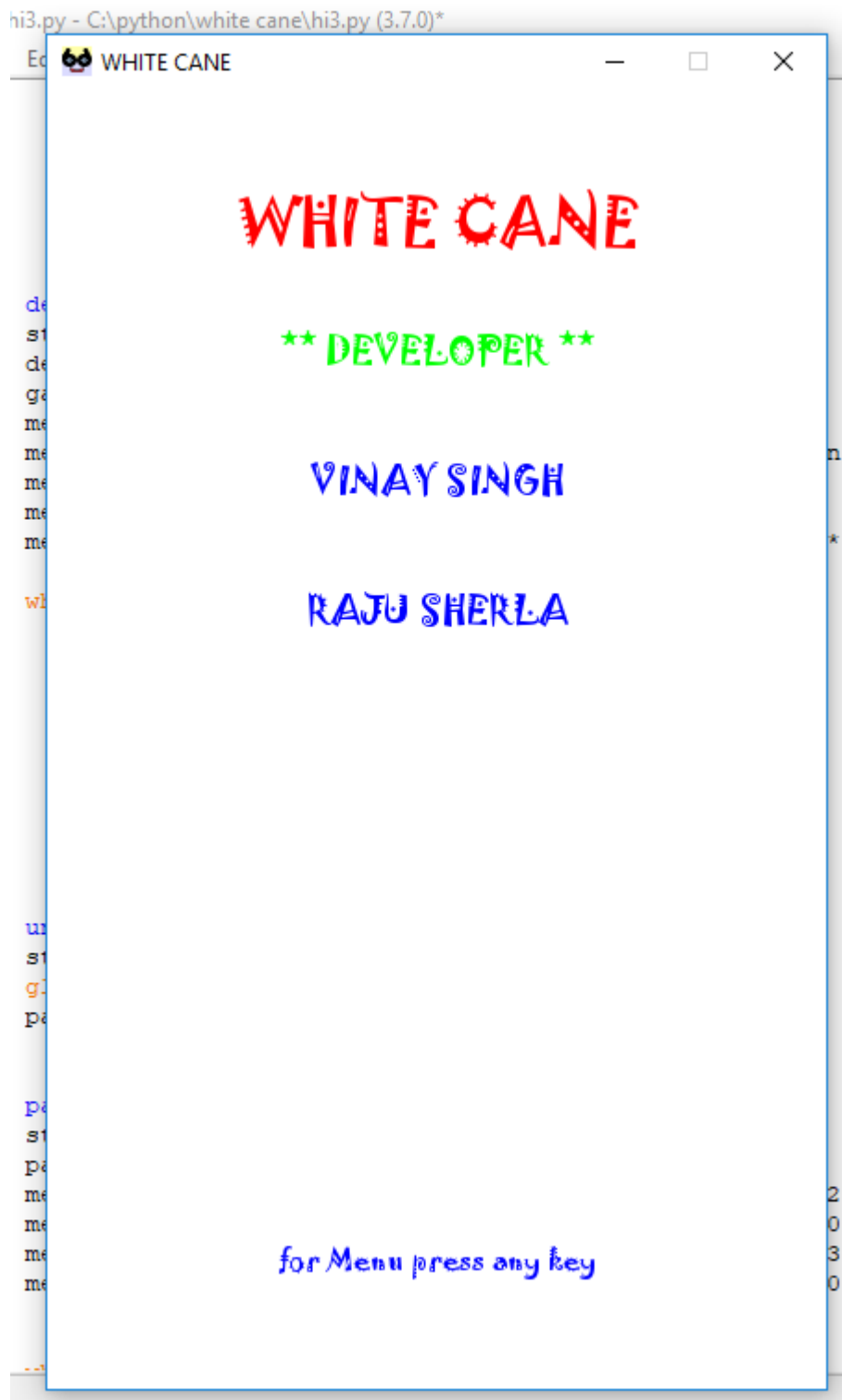


Fig 6.6 developer page player can listen the developer of the game and the college of the developer.

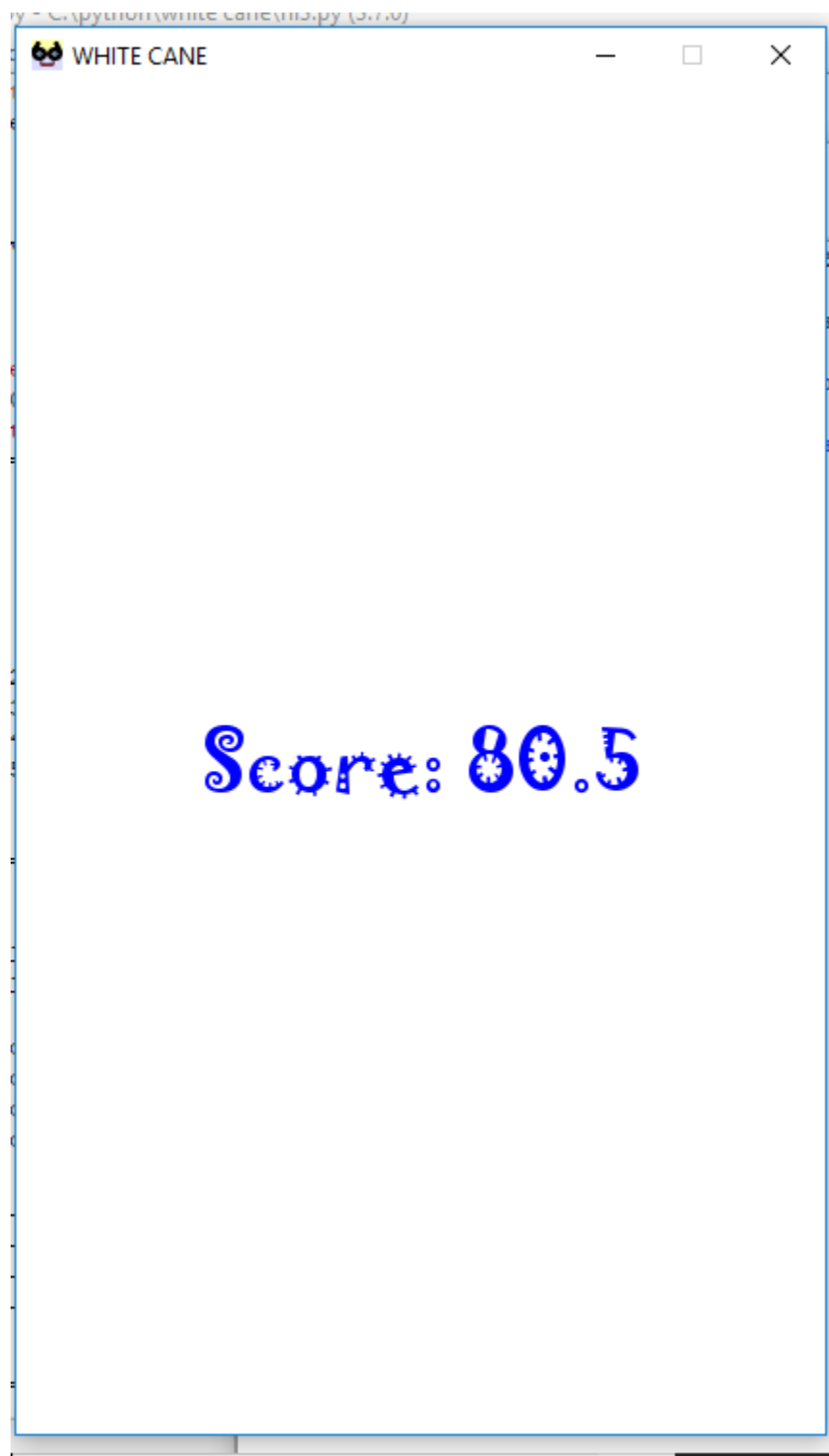


Fig 6.7 after the loss he can listen his score.



Fig 6.8 when player loss the game then he get to know his score and how well he played as per his score.

## **7. CONCLUSION**

### **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 can not 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.