“DEVELOPING A 2-D GAME IN JAVA”

Project Report submitted in partial fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

By

VANSHIKA CHAUHAN

Enrollment No: **161383**

UNDER THE GUIDANCE OF

Dr. Ruchi Verma

Department of Computer Science & Engineering and Information Technology
JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,
WAKNAGHAT, SOLAN-173234, HIMACHAL PRADESH

DECLARATION

I hereby declare that the work presented in this report entitled “Developing a 2D Game In Java” in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, is an authentic record of my own work carried out over a period from 24 February 2020 to 20 March 2020 under the supervision of Dr. Ruchi Verma(Assistant Professor, Senior Grade, Computer Science & Technology and Information Technology). The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Vanshika Chauhan

161383

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)
Dr. Ruchi Verma
Assistant Professor (Senior Grade)
Computer Science & Engineering and Information Technology
Dated:

PROJECT REPORT UNDERTAKING

I Mr. /Ms.________VANSHIKA CHAUHAN________-Roll No.__ _____161383__ _____Branch______COMPUTER SCIENCE AND ENGINEERING_______ is doing my internship with ______INFOSYS PVT. LTD._____from__________24 FEBRUARY 2020___________to________20 MARCH 2020__________

As per procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report. But due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So I hereby declare that the project report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TnP Cell whenever COVID-19 situation gets normal.

Signature_________ Vanshika _____________

Name__________Vanshika Chauhan____________

Date________13/06/2020_________________

ACKNOWLEDGEMENT

I have made efforts towards this project. However, it would not have been possible to complete my work without the inclusion and guidance of many individuals.

I would like to express my special thanks of gratitude to my mentor **Dr. Ruchi Verma** who gave me the golden opportunity to do this project on the topic ‘**Developing A 2D Game In Java’**, which also helped me in doing a lot of research and I came to know about so many new things. She constantly guided and helped me throughout the project work and provided necessary information.

I am really thankful to her.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AABB	AXIS ALIGNED BOUNDING BOX
API	APPLICATION PROGRAM INTERFACE
AWT	ABSTRACT WINDOW TOOLKIT
GUI	GRAPHIC USER INTERFACE
JDK	JAVA DEVELOPMENT KIT
JFC	JAVA FRAMEWORK CLASSES
OOP	OBJECT ORIENTED PROGRAMMING
PC	PERSONAL COMPUTER
2D	2 DIMENSIONAL
3D	3 DIMENSIONAL

ABSTRACT

Technological developments have been increasing rapidly and supplementing daily needs of our lives. Despite of the fact that computer games have entertainment as their prime focus, they are still the lead of computer sciences. Game technology is rapidly increasing and is one of the most sought- after research areas because it has the immense potential of scientific knowledge and commercial success. The objective of developing a 2D game using java was to fully understand and implement the various concepts of java like abstraction, encapsulation, inheritance and polymorphism. Other concepts like multithreading are also implemented and understood, through this project. I also gained an insight about the many classes and methods that are already defined in Java, which are helpful in implementing graphics and the classes like draw and paint, that help the user to create and work with graphics and objects of his choice.

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION

Technological developments have been increasing rapidly and supplementing daily needs of our lives. Despite of the fact that computer games have entertainment as their prime focus, they are still the lead of computer sciences. Game technology is rapidly increasing and is one of the most sought- after research areas because it has the immense potential of scientific knowledge and commercial success. A game is a method of interaction between humans and computers, accumulated in such a way so as to determine a winner in some fashion.

Game dexterity can be explained as an apparatus that is used to advance computer and mobile platform- based video games through assimilating graphic technology, digital techniques, replication methodologies, communicating methods and multi-user functioning factors.

Development and advancement in science and technology gave birth to the revolutionary technology of mobile phones, that were extremely convenient and portable to interact with. This led to the rise in development of mobile phone games. Android- based mobile games are very popular. A game having good graphics and many resources increase its playability and demand in the market. 2-D games are an example of such technology.

Android- games are written in Java programming language. Then this code is compiled to a byte code which is converted to a .dex file for execution. This is further compiled to an .apk file which can be inducted on Android devices for playing the game.

The studies of creating Android-based games have been extensively developed. The games are being developed so that not only they stand entertaining, but also prove to be informative. There are various kinds of such informative games. Some provide cultural insights, some mathematical and scientific knowledge, some increase the vocabulary and some provide awareness about plenty of issues.

This approach towards game development has significantly cut down on the criticism being received by the former on not being knowledge enhancing and meant only for entertainment purposes. This world of gaming has had a significant success in the commercial marketplace.

A game should be attractive and innovative, with good robustness and high- quality graphics too. It should be able to live up to the performance statistics associated with it and should allow smooth functioning of the game. Furthermore, it should also mee the appropriate standards of the Android platform and must include the correct procedures of the instalment, usage and quitting of the game.

Java is preferred for game development as it provides object-oriented platforms for game development which is more efficient. Also, it can be directly accessed for its performance and has a decent user interface.

A game can have two type of elements in it. They are called active and passive elements of a game. The passive game components are the proxy, the elements that keep a track of the progression of the game. Examples of these components are the board and checkers in aboard game, the cards used in playing Rummy etc. These elements are manipulated according to the rules defined in the game, by the players of the game.

Then there are the active elements of the game. These components make their own decisions about ways in which they contribute to the game. For example, the dice rolls accordingly and shows up a certain number by itself, without the intervention of the players.

Rules of a game define the ways in which the various elements interact with each other. The initialising of the configurations, the appropriate moves permitted in the game, the evaluation of the score based on some type of parameter or configuration, the knowledge that the game has finally ended and the declaration of the winner of the game after it ends. All these are critical components of defining the rules of any game.

A certain set of rules can be applied to the various elements of the game without altering the game. For example, a game of chess can be played on a proper chess board with life-like components or on a simple piece of paper too. The components can differ in their respective size and scale, but no in their very kind or nature. Thus, a similar set or rules will lead to a similar type of game in development.

The latter does not stand true for games haring similar elements. For example, all games of card use a deck of 52 cards that have four suites and thirteen components in each suite. The connection between the elements, their communication with each other is based on the rules of the game, is very unique and diverse. The factors like starting conformation, each player’s respective moves and intentions and end goals differ from game to game. Then it becomes a totally new and different game to play.

One can also make use of various game design patterns, such as object- oriented programming, and other modules that can act as templates for code and can be reused again and again for code development and game structure reusability. Many mathematical concepts are also useful in the idea of development of games like linear algebra, matrices, vectors etc. Many popular games like MineCraft, SpaceKnights etc. have been developed using the concepts of Java and OOP's. Any game is comprised of various elements like the game plot, the characters in the game, the motive or goal for the player, the various restrictions on the way the players play the game, the graphical appeal of the game, the various options available to the player during the play of the game and the final result announcement of the winner and the loser of the game. All these components put together can make a useful contribution to game development.

Figure 1. 1 Introduction to 2D games

1.2 PROBLEM STATEMENT

Developing a 2-D game in Java

Java is a great platform for development of 2-D games as it provides the usage of object-oriented concepts for game development. Game technology is rapidly increasing and is one of the most sought- after research areas because it has the immense potential of scientific knowledge and commercial success.

Development and improvement in science and technology gave rise to the ground-breaking technology of mobile phones, that were extremely suitable and moveable to interact with. This led to the increase in expansion of mobile phone games. Android- based mobile games are very prevalent and successful. A game having effective graphics and many properties upsurge its playability and request in the market. 2-D games are an example of such technology.

The games are written in Java programming language. Then this code is assembled to a byte code which is converted to a .dex file for implementation. This is further compiled to an .apk file which can be invested on Android devices for playing the game. The development of an object -oriented system is easy, and helps in maintaining the system as modular and flexible.

The creation of a 2-D game in java includes the usage of many concepts of Java that have also been used in this project. To draw and paint our graphics and images, we need a surface or a window to paint on. This surface object is given to us by JPanel. Since a surface cannot exist on its own and needs a frame to be embodied inside, the JPanel is framed inside a window that we call the object of the JFrame class.

This window can be accordingly resized and reset, or minimized and maximised. The creation of a JFrame object initiates the role of an engine that has the responsibility of managing the user-interface. This then corresponds with the operating system to edit the screen as well as to collect material from the hardware devices like mouse and keyboard. This is the Swing Engine, and it uses the former libraries as well as the concept of many threads.

Any program is executed in a line-by-line method by the processor. But by using the concept of threads, the processor can initiate several implementations consecutively. It behaves as if there exist more than one processor in the system, running simultaneously, their respective instruction arrangements. Each thread behaves as if it is an autonomous program running at the same time like the other threads. Problems using threads can arise if they try to use the same variables at the same time.

The AWT Engine provides the facility of initiating multiple threads.

Now, to be capable of painting on our surface, we need to have the idea that WHERE we should actually paint. This concept of where is provided to us by the JPanel, which is a part of the window. The JPanel class is extended so that it can override the paint method which is that method which has been initiated by the AWT Engine to practice to paint whatsoever appears on the screen.

The method called paint will receive the parameters as an object of the Graphics 2D which is further an extend of the Graphics class. The Graphics class, an older class of the AWT Engine, has now been completely taken over by the Graphics 2D class which has improved functions and a wider variety of functions. The parameters being passed to the method are still the Graphics type because of the compatibility. Using this method, we can select the colours we want to paint our objects with and then draw the figures we need for the game like squares, circles etc.

Now, to actually draw in the surface we have created, we need to specify the positions and coordinates that we will be using for the painting. All the points of our draw surface are given certain x-coordinates and y-coordinates, beginning from the top-leftmost corner of the screen. So, in the Graphics method, we pass the first two parameters as the respective coordinates we want our object to be positioned at, ad the next two parameters passed to the method are the particular height and width that we want of our object. The fill keyword is used to paint an object full of colour whereas the draw keyword is used to simply draw the object without the fill.

The paint method is called every time by the AWT Engine whenever the operation system sends a report that the surface needs to be painted. It is also called the first time the window is shaped. Even when we use the methods of maximizing or minimization or when we alter the size of the window, this paint method will be called. The position of the object created needs to be changed again and again due the movement of the object on the screen. For tis, we can make another method, where the respective x and y coordinates of the object can be incremented by a certain number to the left and right side accordingly.

To loop the game, we can initiate the above made method in an infinite loop situation, to constantly change the position of the object, and then we can call the repaint method which compels the AWT Engine to call the paint method yet again and the surface gets painted again. This cycle of game loop carries out two main tasks:

Update: the updates is given by the method that increments the values in the x and y coordinates respectively.

Render: it paints the present state as well as well as the changes made previously. It is done by the repaint method and then the corresponding call made by the AWT Engine to the paint method. The borders of the objects we create can also be made smoother or rougher, using certain methods.

The Thread.sleep method informs the processor that the thread currently running and is in execution must be put to sleep for the number of milliseconds specified to the method. This permits the processor for the execution of the other threads, especially the thread that calls the paint method. This process can also be achieved by other methods.

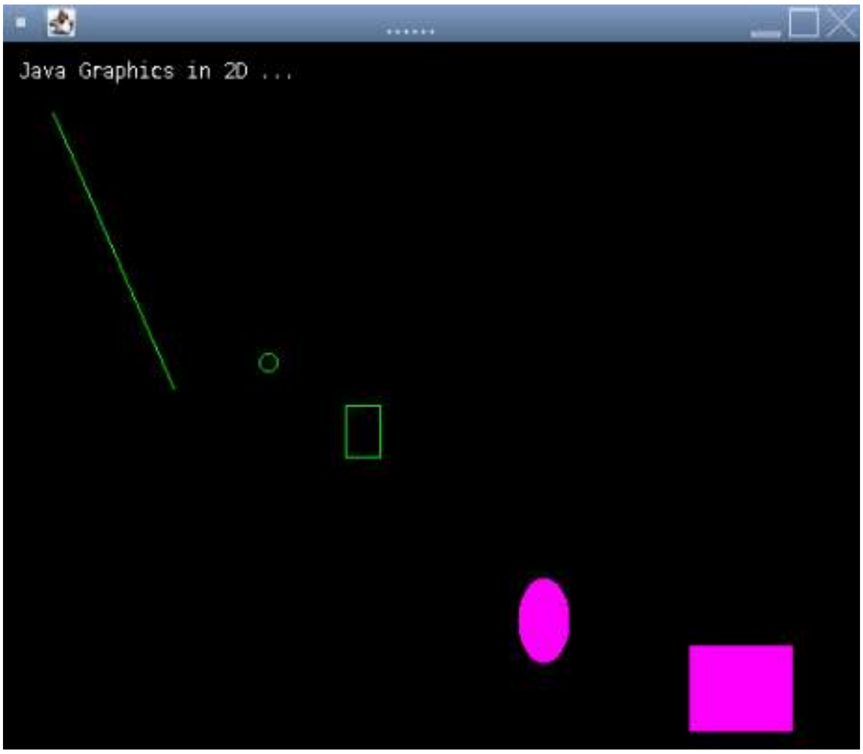


Figure 1. 2 Java Graphics in games

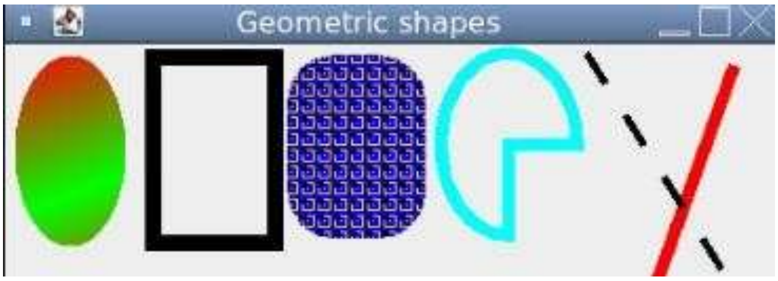


Figure 1. 3 Various geometric shapes in java

1.3 OBJECTIVES

Java is popularly and widely used for game development and making mobile games, because of its versatility and a good collection of open source knowledge packages. Many top level and popular games like Minecraft, Asphalt and Mission Impossible have been developed with the use of java. Java is convenient in usage compared to other languages, and has the property of code reusage, making it platform independent and more efficient. It is comparatively easier to write, compile and debug drawn in comparison to C++.

Because Java is an application based on layers type of programming language, it is not very difficult to comprehend. Because of its well-made structure and logics, it is convenient for working with while working with the process of game development. For beginning with game development, we need to be well acquainted with the framework of Java concepts. The basic Java concepts of learning the usage of objects, classes and methods along with datatypes of all kinds and strings, arrays and loops should be known well.

The most important and fundamental concepts of Java are the OOP concepts. The usage of abstraction, polymorphism, inheritance and encapsulation should be very well known for game development process. The Java Collection Framework describes many objects and classes to characterize them as a single unit. The concepts of data structures can be learnt and implemented using this framework. The exception handling in Java is helpful in catching the bugs in the code.

The concept of threads and multithreading is very important to learn in java. The Thread.sleep method informs the processor that the thread currently running and is in execution must be put to sleep for the number of milliseconds specified to the method. This permits the processor for the execution of the other threads, especially the thread that calls the paint method. This process can also be achieved by other methods.

Using Java, we can create programs that run on web browsers and services too. Influential and well-organized applications can be written in Java for mobile phones due to its versatility and robustness. Through the implementation of this project, I tried to learn the aspects of Java related to game development, and understand the usage of the various concepts of Java in the creation of games. I learnt the practical implementation of the concepts like abstraction, threads, polymorphism etc. through this project work.

I also gained knowledge about the various pre-defined classes and methods in Java that help the user with the implementation of graphics and drawings according to their will of creation. I also learnt that creation of games using java can be very fun, entertaining and informative. It is definitely a very interesting way of learning the implementation of the fundamental concepts of Java, alongside creating a game module for the performing purpose and honing my coding skills. Through the implementation of this project, I enjoyed practicing myjava concepts while doing game development.

The objective of developing a 2D game using javawas to fully understand and implement the various concepts of java like abstraction, encapsulation, inheritance and polymorphism. Other concepts like multithreading are also implemented and understood, through this project.

I also gained an insight about the many classes and methods that are already defined in Java, which are helpful in implementing graphics and the classes like draw and paint, that help the user to create and work with graphics and objects of his choice.

Game development is a forefront of computer science and has extreme potential for research in scientific areas and commercial success value. Games are also the most popular virtual entertainment source and can also be made knowledgeable and informative by adding storylines and concepts related to various topics that increment knowledge of the player while fulfilling the purpose of entertainment.

1.4 METHODOLOGY

In the methodology of game development, the first task is the selection of the platform in which the game needs to be created. The selection for this 2D game project was Java, mainly because it is object oriented and platform independent, and comparatively easier and more convenient and efficient in implementation of the object-oriented concepts than languages like C++. So, Java was chosen for this game creation.

Then, the necessary input is fed through the code written in Java, making use of the various concepts of Java, the pre-defined classes and methods, and functions that are used for creation of the canvas, creating the objects to be used in the game, adding the necessary graphics, images and sounds to the panel created, placing and locating the objects and images on the panel by working with their respective x-coordinates and y-coordinates and positioning and re-positioning them accordingly.

Following this, the user will give some input through the keyboard or mouse and this input will be handled accordingly by the event handler. The event handler holds the responsibility of performing all events related to the execution of the game as per the commands given by the user. It is connected with the logical rules that are being adopted by the game and the graphics that are being used in the game. The usage of the AWT package is needed in the creation of buttons, text-fields, labels and other features used for the graphical user interface.

For executing the above and successfully implementing the process defined, a thorough knowledge of Java and its core concepts is required.

The first process in the development of this project was the thorough understanding and study of the various pre- defined classes in Java that were used in the graphic designing and implementation of methods like paint()

and repaint(). This was further combined with the study of the concept of threads and multithreading and their usage in game development.

To draw and paint our graphics and images, we need a surface or a window to paint on. This surface object is given to us by JPanel. Since a surface cannot exist on its own and needs a frame to be embodied inside, the JPanel is framed inside a window that we call the object of the JFrame class.

The creation of a JFrame object initiates the role of an engine that has the responsibility of managing the user-interface. This then corresponds with the operating system to edit the screen as well as to collect material from the hardware devices like mouse and keyboard.

Now, to be capable of painting on our surface, we need to have the idea that WHERE we should actually paint. This concept of where is provided to us by the JPanel, which is a part of the window. The JPanel class is extended so that it can override the paint method which is that method which has been initiated by the AWT Engine to practice to paint whatsoever appears on the screen.

The method called paint will receive the parameters as an object of the Graphics 2D which is further an extend of the Graphics class. Now, to actually draw in the surface we have created, we need to specify the positions and coordinates that we will be using for the painting. All the points of our draw surface are given certain x-coordinates and y-coordinates, beginning from the top-leftmost corner of the screen. So, in the Graphics method, we pass the first two parameters as the respective coordinates we want our object to be positioned at, and the next two parameters passed to the method are the particular height and width that we want of our object.

The position of the object created needs to be changed again and again due the movement of the object on the screen. For this, we can make another method, where the respective x and y coordinates of the object can be incremented by a certain number to the left and right side accordingly.

To loop the game, we can initiate the above made method in an infinite loop situation, to constantly change the position of the object, and then we can call the repaint method which compels the AWT Engine to call the paint method yet again and the surface gets painted again.

After understanding these concepts, the visuals and sounds needed for the game being developed in the project were downloaded and made part of the same folder in which the java game was being created. After completing the above tasks, the coding of the game was initiated.

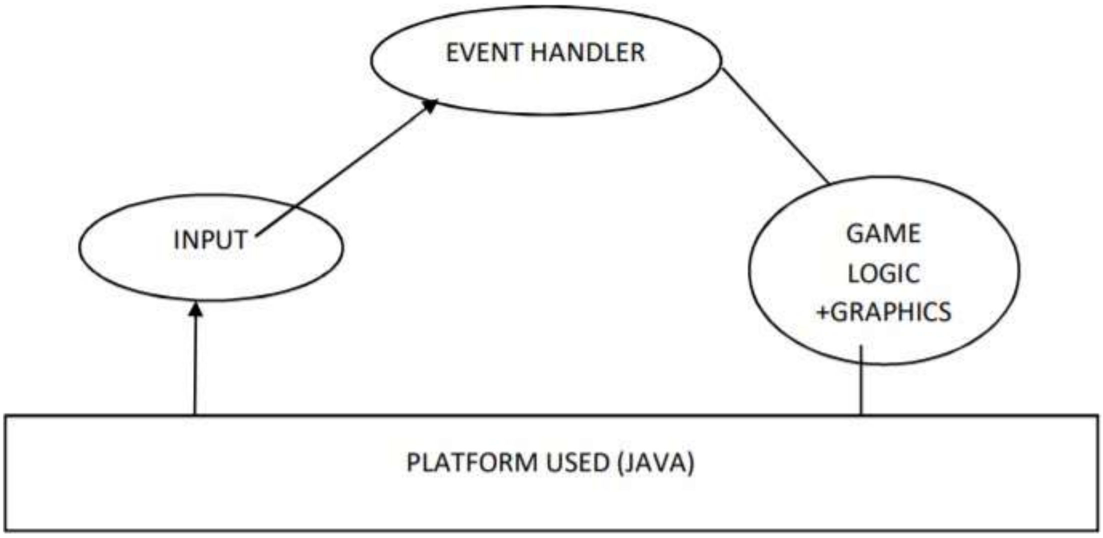


Figure 1. 4 Game making procedure

1.5 ORGANISATION

In chapter 1, we have discussed the growing commercial value and research success of the gaming industry. We talked about the idea of game development, and how it is an emerging field in computer sciences to learn the game creation process and also enhance the knowledge about the programming language used in the process of game making, which, in this case, is Java. We discussed about the various important concepts of Java and why it is an essential and popular language for game coding. We also talked about the methods used in creating a java -based project that produces a 2D game for the user.

In chapter 2, we discussed about the various research papers we studied to get a deep knowledge about the process of game development using the concept of Java. A thorough study about the concepts of java, pre-defined graphics classes and functions to be used, and the motive behind making such games was studied, to emphasize the significance of game creation in today's scenario.

In chapter 3, we have discussed about the system analysis and design of the project. We have discussed about the method and algorithm that have been used in the procedure of game development, and how, step-by-step, we have put together the various elements of the game to a level where all of them function together in certain synchronisation to produce the game output in a desired way.

In chapter 4, we do an analysis of the performance of our 2D game by executing the given set of code and checking whether it matches with the desired output. We also check the various elements and sub-parts of the game individually, making certain that each function does the required work while maintaining a proper synchronisation with the other parts. It is an experimental way of checking the performance measure of the game, by making sure that the game runs smoothly and as desired by the user. The results at different stages are compared with each other and a full picture is given at the end.

In the last chapter, we have discussed about the future scope of the project. We have talked about how this 2D project can be improved further to be converted to a 3D project, alongside the concept of real time gameplay and multi-user play.

CHAPTER-2

LITERATURE REVIEW

2.1 ABSTRACT-1

The process of development of games using java has been introduced here. Games are quick in adapting to the current trends in the gaming world and flexible in every environment they are put into. The codes generated for these games are machine independent and can be transferred from platform to platform without much problem. These codes are transformable and can be maintained for a good duration of time.

The demarcation between games can be made by the various versions of the game that are developed and made available in the commercial gaming market. The researchers demarcate the games based on some conflicting designs in the history of that particular game that had been made.

One of the most significant changes that have been observed by the software industry are the modifications related to the gaming world. The game world has turned out to be one of the most economically beneficial and sought-after field in comparison to any other field.

Java is preferred over other languages for the process of game development because it is object – oriented in nature, giving a platform to the developer that is very efficient and easy to work on. Models of games provide interesting research ideas and fields for people.

In todays time, the usage of computers and technology is immense and ground-breaking when it comes to the development of amazing graphics for games. Thus, by using the technological advancements made by science of computers, the graphics being introduced in games can also be improvised with a huge building scope inthe coming future too.

The game architecture for a fairly simple game in java is firstly deciding on the platform to be used for development. The choice has been made as java. Then the making of the process of providing input to the game by the layer has to be created, followed by the tasks being performed by an event handler, which undertakes the responsibility of handling all events related in response to the gameplay.

Lastly, all logical developments and graphical advancements related to the game are linked to all the above events, so that the game can function as a whole without any discrepancies. The objective of the paper was to introduce game development for beginners using the concepts of java and understanding the need and benefits of game development in todays world of technology and booming market for the gaming world.

2.2 ABSTRACT-2

This study was determined was to create a 2D adventure game using a collision of bounding box methods. The game is meant for entertainment purpose and for fun in free time. The collision detection method is used here to find out the act of collision from various objects in the game.

This game is assembled using Eclipse IDE and Java programming language. The collision detection by a bounding box method takes place when the players object comes into contact with the bounding box of the target object. When the players object reaches into the targets box range, the detection for collision occurs. As a consequence, the target object gets destroyed.

Technology expansion has become quickly incrementing and being secondary to daily human needs. Computer games are more than entertainment and leisure activity, rather they are a major research area. Development related to technological advancement is spreading very speedily and has become a highly researched area by people because it contains the immense potential of both commercial value and scientific knowledge simultaneously.

In today's time, the usage of computers and technology is immense and ground-breaking when it comes to the development of amazing graphics for games. Thus, by using the technological advancements made by science of computers, the graphics being introduced in games can also be improvised with a huge building scope in the coming future too

Collision - detection algorithms in 2D games depend on the shapes that are being collided. The shapes can be rectangular versus rectangular, circle versus circle and circle versus rectangle. Most of the times, we have convenient shapes hiding the shape known as a hitbox, so even when the collisions are not perfect by pixel-matching, it works just fine and gives a good performance measure across numerous units.

There should be detection of objects colliding in some virtual environment, in collision detection. It is mainly used to refrain objects from passing through each other. Collision Detection can be found in every aspect of computer games: amongst characters, between characters and objects, etc. Collision detection implies the recognition of the joint of two or more objects.

A bounding box or hitbox is a box that engulfs an entire object inside it. It can be characterized as a rectangular figure which has an x coordinate and a y coordinate, and some height and width too. A common manifestation of this exists in 2D games.

The objective of this study was to understand the logic behind the collision between objects in any game using the collision detection method of bounding boxes.

2.3 ABSTRACT-3

The process of development of games using java has been introduced here. Games are quick in adapting to the current trends in the gaming world and flexible in every environment they are put into. The codes generated for these games are machine independent and can be transferred from platform to platform without much problem. These codes are transformable and can be maintained for a good duration of time.

One of the most significant changes that have been observed by the software industry are the modifications related to the gaming world. The game world has turned out to be one of the most economically beneficial and sought-after field in comparison to any other field.

In today's time, the usage of computers and technology is immense and ground-breaking when it comes to the development of amazing graphics for games. Thus, by using the technological advancements made by science of computers, the graphics being introduced in games can also be improvised with a huge building scope inthe coming future too

This game is assembled using Eclipse IDE and Java programming language. The collision detection by a bounding box method takes place when the players object comes into contact with the bounding box of the target object. When the players object reaches into the targets box range, the detection for collision occurs. As a consequence, the target object gets destroyed.

There should be detection of objects colliding in some virtual environment, in collision detection. It is mainly used to refrain objects from passing through each other. Collision Detection can be found in every aspect of computer games: amongst characters, between characters and objects, etc. Collision detection implies the recognition of the joint of two or more objects.

AXIS-ALIGNED BOUNDING BOX (AABB): the more simple and easy forms of collision is when two rectangles collide when their axis are aligned toc each other, i.e. there is no rotation in the objects. This algorithm works only by making sure that there is no gap between any of the four sides of the rectangle. Gaps indicate that collision will not happen.

A bounding box or hitbox is a box that engulfs an entire object inside it. It can be characterized as a rectangular figure which has an x coordinate and a y coordinate, and some height and width too. A common manifestation of this exists in 2D games.

There should be detection of objects colliding in some virtual environment, in collision detection. It is mainly used to refrain objects from passing through each other. Collision Detection can be found in every aspect of computer games: amongst characters, between characters and objects, etc. Collision detection implies the recognition of the joint of two or more objects.

2.4 ABSTRACT-4

Any software game has inter-disciplinary uses, which are way more than entertainment, like in the fields of education, health, business etc. A game development is a complex process including the combination of various things like sounds, graphics, human factor, artificial intelligence, thus, making the whole procedure very unique and different from regular software development.

However, the basic principles of software engineering are helpful in achieving the desired goals of flexibility, maintainability, robustness, cost effectiveness and better outcomes. The pre-production phase related to game development is more-full of activity and usage of these software engineering principles rather than the post development period of the game. Therefore, the research can be well extended into the post creation phase of game development.

The game architecture for a fairly simple game in java is firstly deciding on the platform to be used for development. The choice has been made as java. Then the making of the process of providing input to the game by the layer has to be created, followed by the tasks being performed by an event handler, which undertakes the responsibility of handling all events related in response to the gameplay.

Various tech giants have invested in the business of game development using numerous software and this field has thus attracted youth in a large number. Not only is this market of online gaming providing an opportunity to the people in the world, it is also serving as a means for the students to enrol into professional courses which include learning various computer languages and developing new software. The young minds with basic needs and guidance are making best use of their talent to make use in expanding this gaming platform.

Development and advancement in science and technology gave birth to the revolutionary technology of mobile phones, that were extremely convenient and portable to interact with. This led to the rise in development of mobile phone games. Android- based mobile games are very popular. A game having good graphics and many resources increase its playability and demand in the market. 2-D games are an example of such technology

Lastly, all logical developments and graphical advancements related to the game are linked to all the above events, so that the game can function as a whole without any discrepancies. The objective of the paper was to introduce game development for beginners using the concepts of java and understanding the need and benefits of game development in today's world of technology and booming market for the gaming world.

2.5 ABSTRACT-5

About half century before, the creation of the first software game had been done. Since then, the world of game development and programming has changed drastically. Now the situation is so that the gaming world is at par with the other entertainment businesses like cinema and the music industry.

As a consequence of this growth, the game software development has spurred greatly, and has made a lot of profit, and has also gained maturity and strength with the passage of time. The creation of games has impacted the industry as well, which builds on unique methods and mature ways for growth, therefore causing a highly advanced process of game creation.

Developers try and create games that are unlike any other already available in the market. This uniqueness can be made possible by having creative viewpoints, new game-stories, variation in genres used, improved graphics and special characters. So games have to be novel for their success in the market. Also, their should be good quality of the games being made.

Only games possessing a proper quality are successful in the long run, and this has turned into a critical success factor for any software development company. To say differently, a poor quality game will lose its players to a better quality one in no time. So it is essential for companies to change themselves according to the needs and demands of their customers.

The major concerns in the procedure for making games is the habit and necessity to work with the best practicality and process from the usage of software engineering basics. The process of development of a game can be categorised to the following main ideas:

Firstly, the introduction of the concept for the game. This can be followed by the pre-production idea, and then the process of actual production, and finally the idea of post-production. It further contains a number of processes like research about the story background, creating a good script, the idea of visual concepts and artistic skill development, designing and requisite animation, the proper programming, editing, combination, checking and issuing.

The characterisation of software games is done based on whichever genre they are working on. There are various genres like adventure, horror, comedy, music, educational etc. each category has some requisites to be fulfilled for the game to be put into that category, and this is done during the pre-production period. For this factor, the development of a game is considered a combination of many disciplines that work together effectively to get some result.

The process of game development also needs a wide variety of services and talents that comprise of the designing of the game, the whole process of managing the project, expansion, and advantage formation. There are people involved from multiple fields and disciplines e.g., designers for the game, programming people, and developers.

Information and insight of top- level practices for creation of new games is essential and effective. By the end, this data will support the developers of games in taking proper and apt decisions related to the making of their game. A good and quality research of prime factors from a game-creators standpoint will pay to the comprehension of existing creation procedure suggestions.

The progress of the game development business is gigantic and is earning value by each passing day. This progress enforces a lot of heaviness and a plenty of questions and tests on the process and people into the making of games. Game development is a procedure full of complexity, and an effective game creation technique is to take into consideration the viewpoint of the developer to put forth apt gaming software by refining the process of game creation.

Its prime focus is basically on an experimental research of the consequence of significant factors related to the developer on the process of development of gaming software process and consequently on the excellence of the game being resulting in formation.

The results of this study give enough proof that game creation businesses and industries should have an arrangement with numerous significant features to stay ahead in competition and to bear immense burden of the ever -developing software program gaming industry. The prime influence of this study is to explore the impact of prime inventor features on the procedure of development of games.

2.6 ABSTRACT-6

The creation of a 2-D game in java includes the usage of many concepts of Java that have also been used in this project. To draw and paint our graphics and images, we need a surface or a window to paint on. This surface object is given to us by JPanel. Since a surface cannot exist on its own and needs a frame to be embodied inside, the JPanel is framed inside a window that we call the object of the JFrame class.

This window can be accordingly resized and reset, or minimized and maximised. The creation of a JFrame object initiates the role of an engine that has the responsibility of managing the user-interface. This then corresponds with the operating system to edit the screen as well as to collect material from the hardware devices like mouse and keyboard. This is the Swing Engine, and it uses the former libraries as well as the concept of many threads.

Then, the necessary input is fed through the code written in Java, making use of the various concepts of Java, the pre-defined classes and methods, and functions that are used for creation of the canvas, creating the objects to be used in the game, adding the necessary graphics, images and sounds to the panel created, placing and locating the objects and images on the panel by working with their respective x-coordinates and y-coordinates and positioning and re-positioning them accordingly.

Following this, the user will give some input through the keyboard or mouse and this input will be handled accordingly by the event handler. The event handler holds the responsibility of performing all events related to the execution of the game as per the commands given by the user. It is connected with the logical rules that are being adopted by the game and the graphics that are being used in the game. The usage of the AWT package is needed in the creation of buttons, text-fields, labels and other features used for the graphical user interface.

For executing the above and successfully implementing the process defined, a thorough knowledge of Java and its core concepts is required.

The first process in the development of this project was the thorough understanding and study of the various pre- defined classes in Java that were used in the graphic designing and implementation of methods like paint()

and repaint(). This was further combined with the study of the concept of threads and multithreading and their usage in game development.

To draw and paint our graphics and images, we need a surface or a window to paint on. This surface object is given to us by JPanel. Since a surface cannot exist on its own and needs a frame to be embodied inside, the JPanel is framed inside a window that we call the object of the JFrame class.

The creation of a JFrame object initiates the role of an engine that has the responsibility of managing the user-interface. This then corresponds with the operating system to edit the screen as well as to collect material from the hardware devices like mouse and keyboard.

The method called paint will receive the parameters as an object of the Graphics 2D which is further an extend of the Graphics class. Now, to actually draw in the surface we have created, we need to specify the positions and coordinates that we will be using for the painting. All the points of our draw surface are given certain x-coordinates and y-coordinates, beginning from the top-leftmost corner of the screen. So, in the Graphics method, we pass the first two parameters as the respective coordinates we want our object to be positioned at, and the next two parameters passed to the method are the particular height and width that we want of our object.

The position of the object created needs to be changed again and again due the movement of the object on the screen. For this, we can make another method, where the respective x and y coordinates of the object can be incremented by a certain number to the left and right side accordingly. [1]

To loop the game, we can initiate the above made method in an infinite loop situation, to constantly change the position of the object, and then we can call the repaint method which compels the AWT Engine to call the paint method yet again and the surface gets painted again.

2.7 ABSTRACT-7

Technological developments have been increasing rapidly and supplementing daily needs of our lives. Despite of the fact that computer games have entertainment as their prime focus, they are still the lead of computer sciences. Game technology is rapidly increasing and is one of the most sought- after research areas because it has the immense potential of scientific knowledge and commercial success. A game is a method of interaction between humans and computers, accumulated in such a way so as to determine a winner in some fashion.

Game dexterity can be explained as an apparatus that is used to advance computer and mobile platform- based video games through assimilating graphic technology, digital techniques, replication methodologies, communicating methods and multi-user functioning factors.

Development and advancement in science and technology gave birth to the revolutionary technology of mobile phones, that were extremely convenient and portable to interact with. This led to the rise in development of mobile phone games. Android- based mobile games are very popular. A game having good graphics and many resources increase its playability and demand in the market. 2-D games are an example of such technology.

Android- games are written in Java programming language. Then this code is compiled to a byte code which is converted to a .dex file for execution. This is further compiled to an .apk file which can be inducted on Android devices for playing the game.

The major concerns in the procedure for making games is the habit and necessity to work with the best practicality and process from the usage of software engineering basics. The process of development of a game can be categorised to the following main ideas:

Firstly, the introduction of the concept for the game. This can be followed by the pre-production idea, and then the process of actual production, and finally the idea of post-production. It further contains a number of processes like research about the story background, creating a good script, the idea of visual concepts and artistic skill development, designing and requisite animation, the proper programming, editing, combination, checking and issuing.

The characterisation of software games is done based on whichever genre they are working on. There are various genres like adventure, horror, comedy, music, educational etc. each category has some requisites to be fulfilled for the game to be put into that category, and this is done during the pre-production period. For this factor, the development of a game is considered a combination of many disciplines that work together effectively to get some result.

The process of game development also needs a wide variety of services and talents that comprise of the designing of the game, the whole process of managing the project, expansion, and advantage formation. There are people involved from multiple fields and disciplines e.g., designers for the game, programming people, and developers.

Information and insight of top- level practices for creation of new games is essential and effective. By the end, this data will support the developers of games in taking proper and apt decisions related to the making of their game. A good and quality research of prime factors from a game-creators standpoint will pay to the comprehension of existing creation procedure suggestions.

The progress of the game development business is gigantic and is earning value by each passing day. This progress enforces a lot of heaviness and a plenty of questions and tests on the process and people into the making of games. Game development is a procedure full of complexity, and an effective game creation technique is to take into consideration the viewpoint of the developer to put forth apt gaming software by refining the process of game creation.

Development and advancement in science and technology gave birth to the revolutionary technology of mobile phones, that were extremely convenient and portable to interact with. This led to the rise in development of mobile phone games. Android- based mobile games are very popular. A game having good graphics and many resources increase its playability and demand in the market. 2-D games are an example of such technology.

Android- games are written in Java programming language. Then this code is compiled to a byte code which is converted to a .dex file for execution. This is further compiled to an .apk file which can be inducted on Android devices for playing the game.

The studies of creating Android-based games have been extensively developed. The games are being developed so that not only they stand entertaining, but also prove to be informative. There are various kinds of such informative games. Some provide cultural insights, some mathematical and scientific knowledge, some increase the vocabulary and some provide awareness about plenty of issues.

A game should be attractive and innovative, with good robustness and high- quality graphics too. It should be able to live up to the performance statistics associated with it and should allow smooth functioning of the game. Furthermore, it should also mee the appropriate standards of the Android platform and must include the correct procedures of the instalment, usage and quitting of the game.

Java is preferred for game development as it provides object-oriented platforms for game development which is more efficient. Also, it can be directly accessed for its performance and has a decent user interface.

A game can have two type of elements in it. They are called active and passive elements of a game. The passive game components are the proxy, the elements that keep a track of the progression of the game. Examples of these components are the board and checkers in aboard game, the cards used in playing Rummy etc. These elements are manipulated according to the rules defined in the game, by the players of the game.

Then there are the active elements of the game. These components make their own decisions about ways in which they contribute to the game. For example, the dice rolls accordingly and shows up a certain number by itself, without the intervention of the players.

CHAPTER-3

SYSTEM DEVELOPMENT

3.1 LEARNING ABOUT THE JAVA AWT:

The Java Abstract Window Toolkit (AWT) is an API using which we can develop GUI or window- based applications in java. Platform dependency is a prime aspect of the java AWT components, i.e. that the components are demonstrated in accordance to the view of the system. The components belonging to AWT use the resources of the system, hence making them heavyweight. [2] [3]

The java.awt is the package that offers classes for the AWT API like text-fields, text-areas, labels, checkboxes, lists, shapes, graphics, colours etc. these various classes in the AWT package follow a hierarchy.

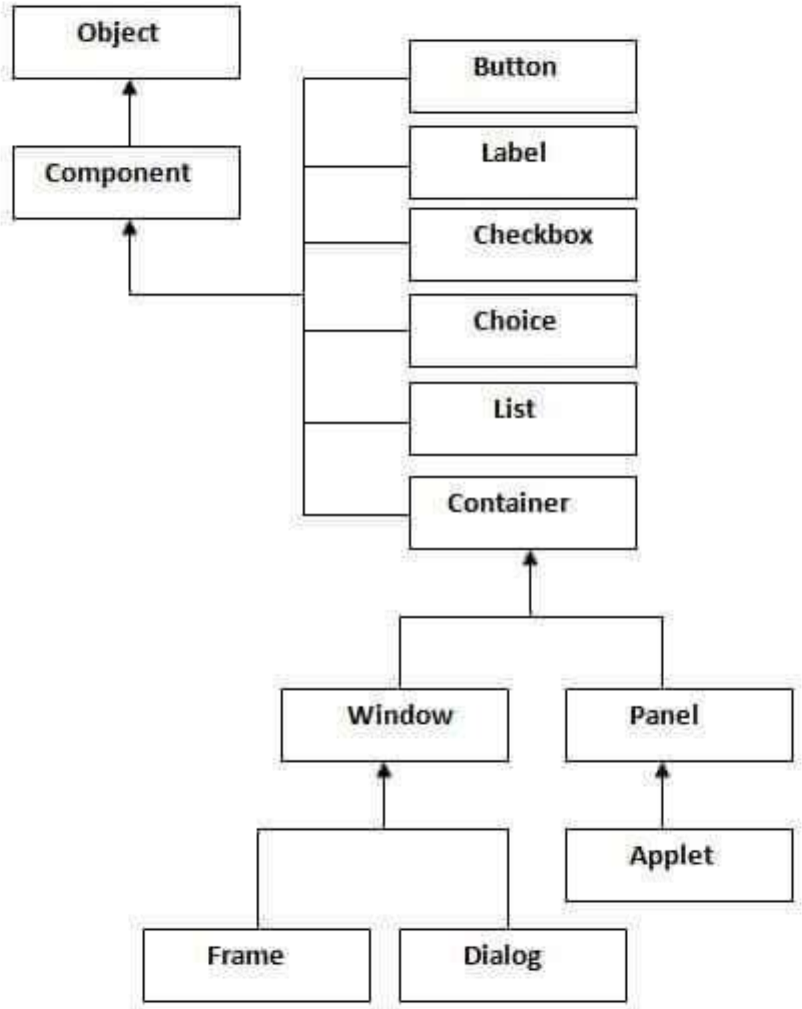


Figure 3. 1 Java AWT hierarchy

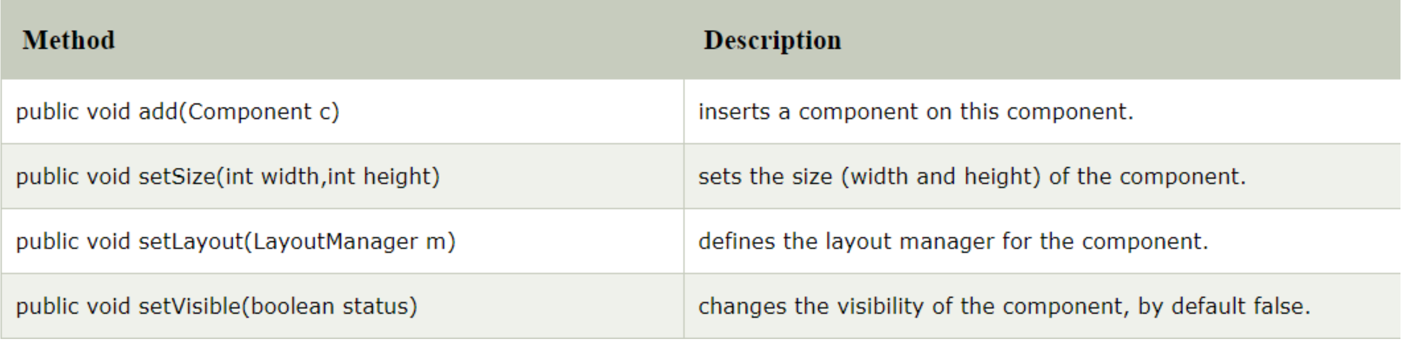
CONTAINER: the container component contains in it various other components like labels, buttons, text-fields etc. the classes extending this class are recognized as container like Frame, Panel and Dialog.

WINDOW: the window container contains no borders and bars like the menu bars etc. we have to make use of components like frame, panel etc or create a completely separate window for making this window.

PANEL: it is the kind of a container having no menu and title bars but can have many other components like text-areas, buttons etc.

FRAME: the frame is that container which includes the bars like title bar and menu bar. It can also contain many other components like buttons, text-fields etc. [4]

The following methods are commonly used:



Method	Description
public void add(Component c)	inserts a component on this component.
public void setSize(int width,int height)	sets the size (width and height) of the component.
public void setLayout(LayoutManager m)	defines the layout manager for the component.
public void setVisible(boolean status)	changes the visibility of the component, by default false.

Figure 3. 2 Various methods used in Java AWT

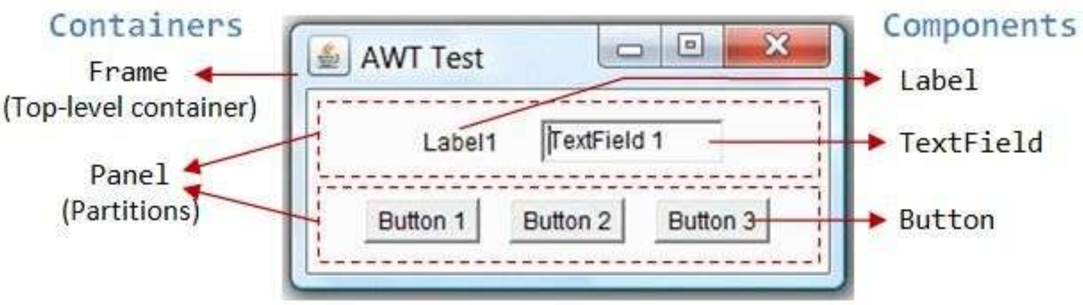


Figure 3. 3 Description of an AWT class

To understand the usage of AWT, the frame needs to be created. The frame can be made in the following ways:

By using the concept of inheritance and extending the Frame class.

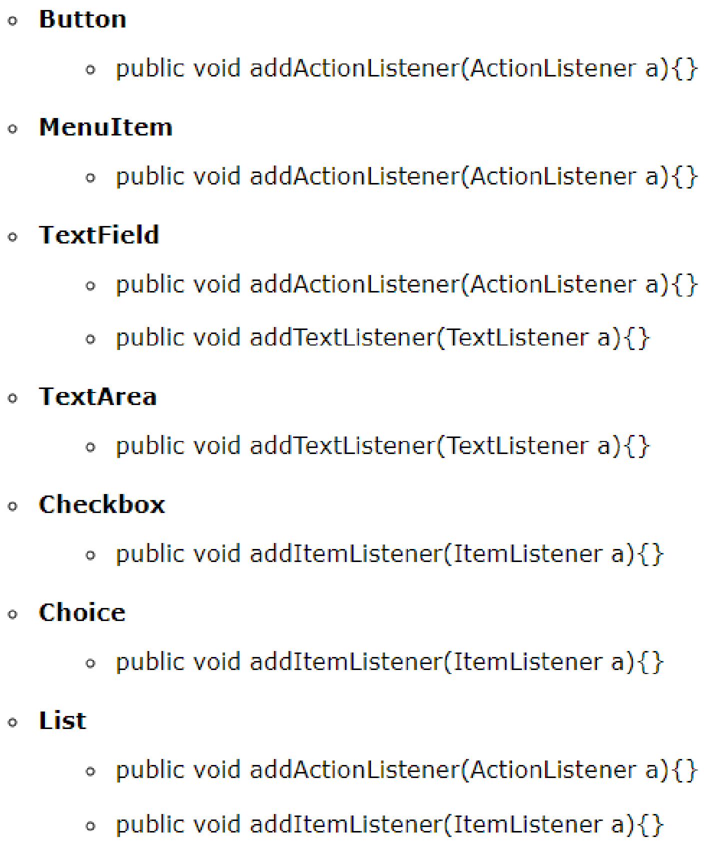
By using association and making an object of the Frame class.

3.2 EVENT HANDLING IN JAVA: the changes in the state and configuration of any object in java is called as event handling in java. Examples of event handling include button clicks, mouse selection or drag etc. The required event classes and listener objects are provided in java using the java.awt.event package.

The following steps are used while event handling:

The component is firstly registered with the Listener. The registration methods are provided by many classes as mentioned below. These methods are basically the button, text-field, text-area, menu-item, checkbox, choice and list classes. The classes help us with the implementation of the process of event handling in java embedded within the java.awt package.

The classes are illustrated as follows:

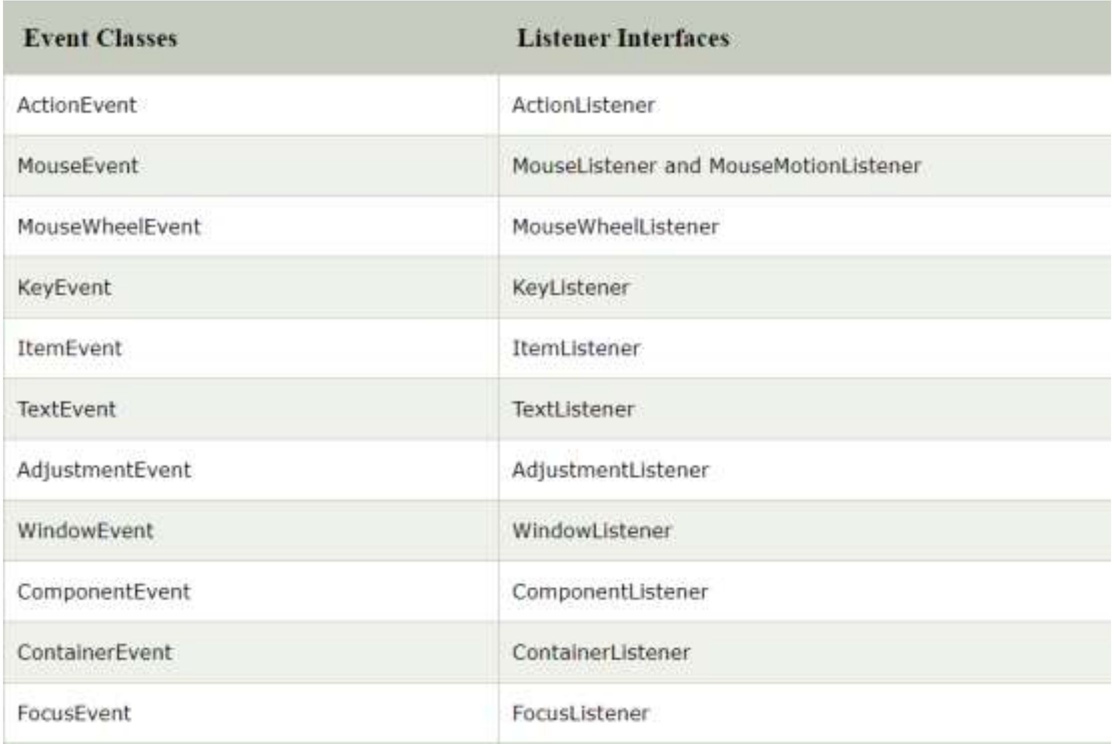


- **Button**
  - public void addActionListener(ActionListener a){}
- **MenuItem**
  - public void addActionListener(ActionListener a){}
- **TextField**
  - public void addActionListener(ActionListener a){}
  - public void addTextListener(TextListener a){}
- **TextArea**
  - public void addTextListener(TextListener a){}
- **Checkbox**
  - public void addItemListener(ItemListener a){}
- **Choice**
  - public void addItemListener(ItemListener a){}
- **List**
  - public void addActionListener(ActionListener a){}
  - public void addItemListener(ItemListener a){}

Figure 3. 4 Classes in event handling in java

This process of event handling can be done in three ways, it can be done within a class, outside a class and by making an anonymous class. In our project, it has been implemented using the ActionListener class.

The various event and listener classes and interfaces used in java are as follows:



Event Classes	Listener Interfaces
ActionEvent	ActionListener
MouseEvent	MouseListener and MouseMotionListener
MouseWheelEvent	MouseWheelListener
KeyEvent	KeyListener
ItemEvent	ItemListener
TextEvent	TextListener
AdjustmentEvent	AdjustmentListener
WindowEvent	WindowListener
ComponentEvent	ComponentListener
ContainerEvent	ContainerListener
FocusEvent	FocusListener

Figure 3. 5 Various event and listener classes in java

THE JAVA ACTIONLISTENER INTERFACE: the actionlistener gets a notification whenever a click is made on any button or the item called menu. It is a part of the java.awt.apckage and has only one method in it called the ActionPerformed() method.

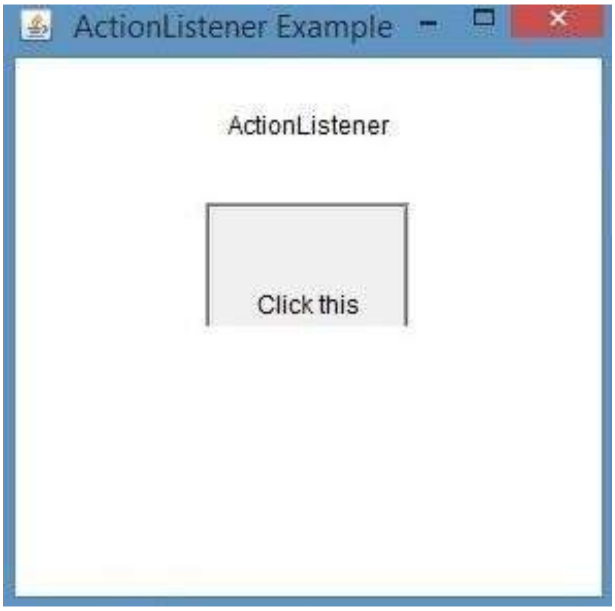


Figure 3. 6 Action listener in java

THE JAVA KEYLISTENER INTERFACE: the keylistener will get a notification whenever the state of the key is changed. It is a part of the java.awt.apckage. It contains the following three methods:

public abstract void keyPressed (KeyEvent e);
public abstract void keyReleased (KeyEvent e);
public abstract void keyTyped (KeyEvent e); [5]

3.3 JAVA SWING: using java swing, which is a part of the Java Foundation Classes (JFC), window-based applications can be created. It is made on top of the AWT package and is written totally in java. Unlike AWT, java swing is independent of the platform, and is also light-weight in nature. The javax.swing package offers classes for java swing API like JButton, JTextField, JTextArea, JCheckbox, JMenu etc.

The java swing API follows a hierarchical order given as follows:

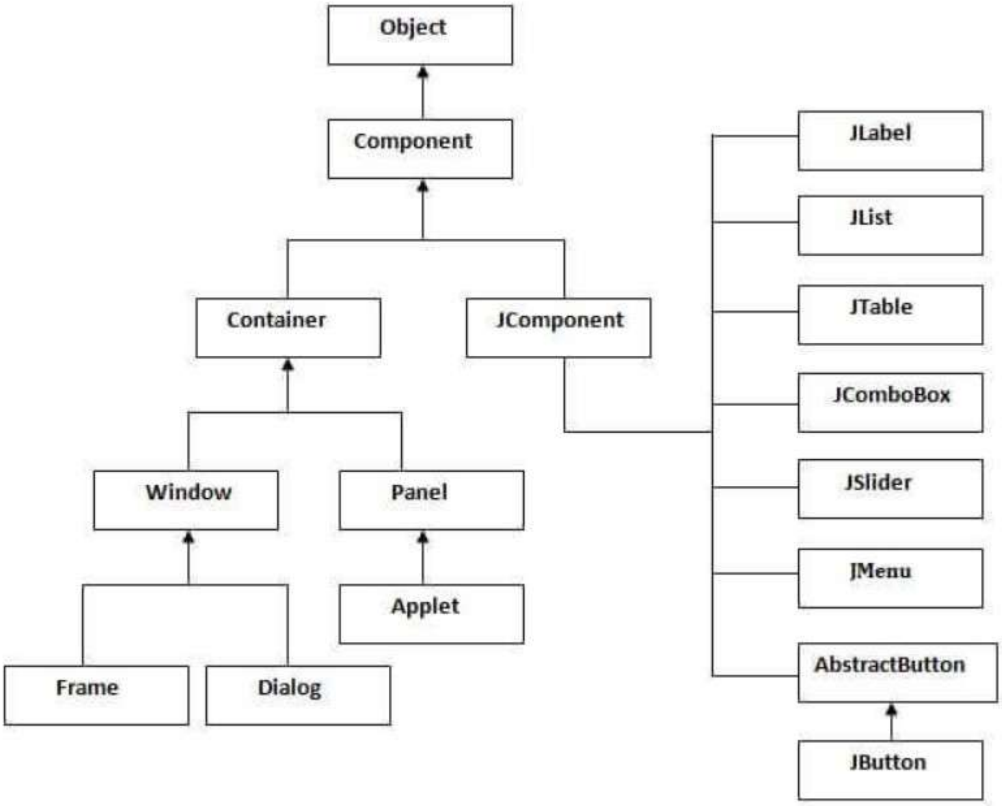
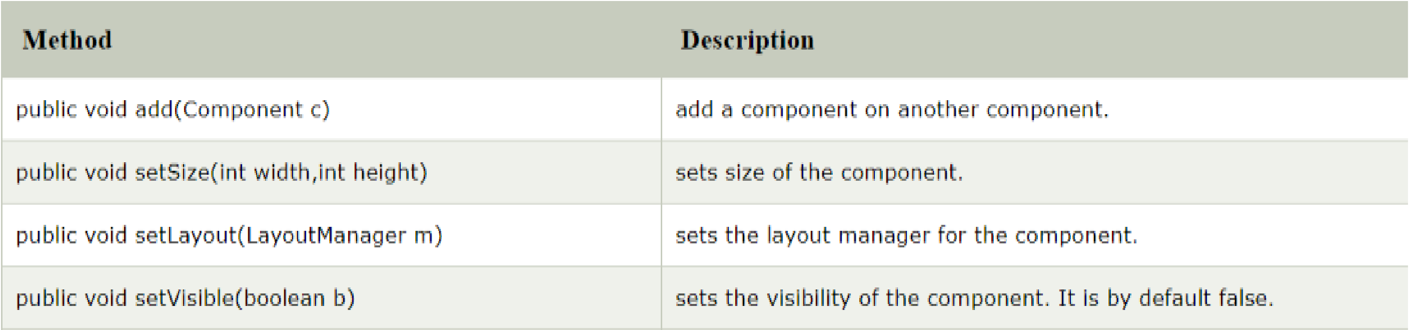


Figure 3. 7 Hierarchy in Java Swing

Various components can be used in the swing package. The most frequently used components in swing are as follows:

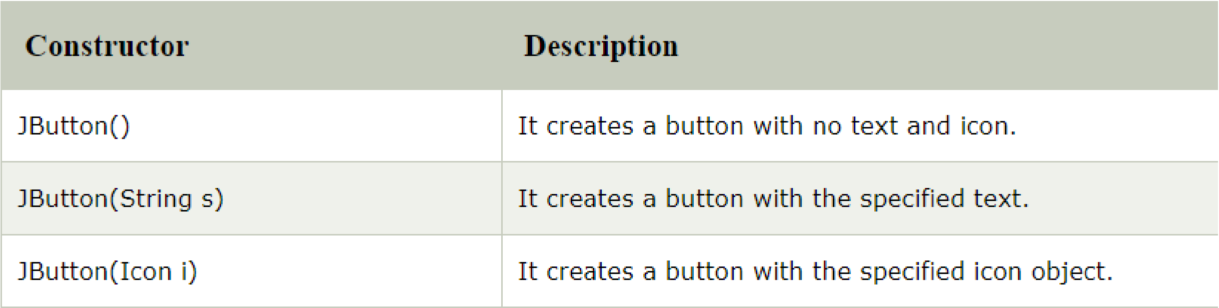


Method	Description
public void add(Component c)	add a component on another component.
public void setSize(int width,int height)	sets size of the component.
public void setLayout(LayoutManager m)	sets the layout manager for the component.
public void setVisible(boolean b)	sets the visibility of the component. It is by default false.

Figure 3. 8 Commonly used components in swing

JBUTTON IN JAVA SWING: the button created using this class is labelled in nature and is platform independent. The outcome is some sort of an action when the button is clicked onto or pressed. Its inheritance class is the abstract button class.

Frequently used constructors with this class are as follows:



Constructor	Description
JButton()	It creates a button with no text and icon.
JButton(String s)	It creates a button with the specified text.
JButton(Icon i)	It creates a button with the specified icon object.

Figure 3. 9 Constructors of JButton in java

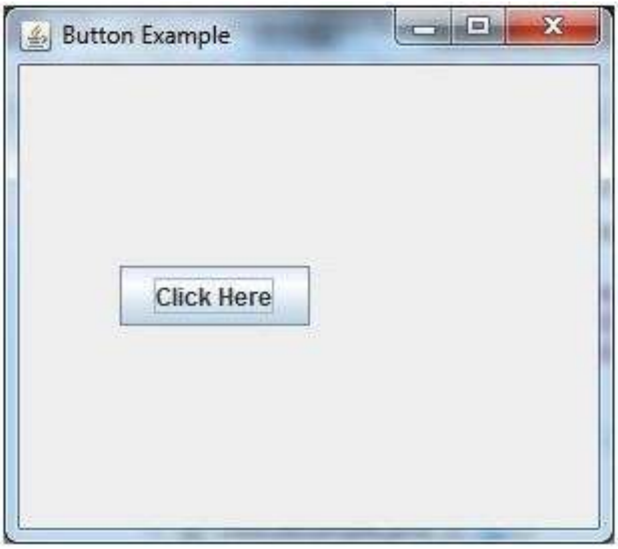
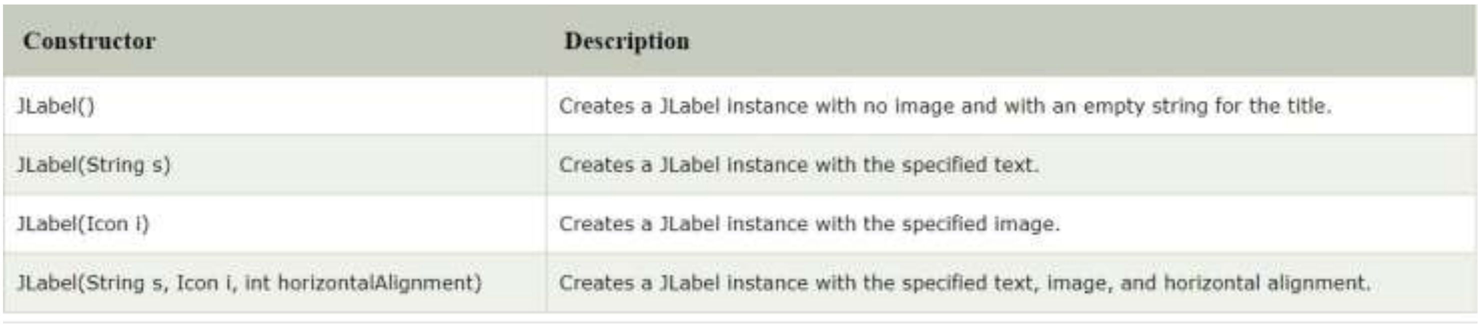


Figure 3. 10 JButton in java

JLABEL IN JAVA SWING: the usage of jlabel is for the display of a single text line which is only in read only format. The object in the jlabel class is an element for inserting text in a container. The text in the box cannot be altered directly by the user and needs a proper application for any changes in the text.

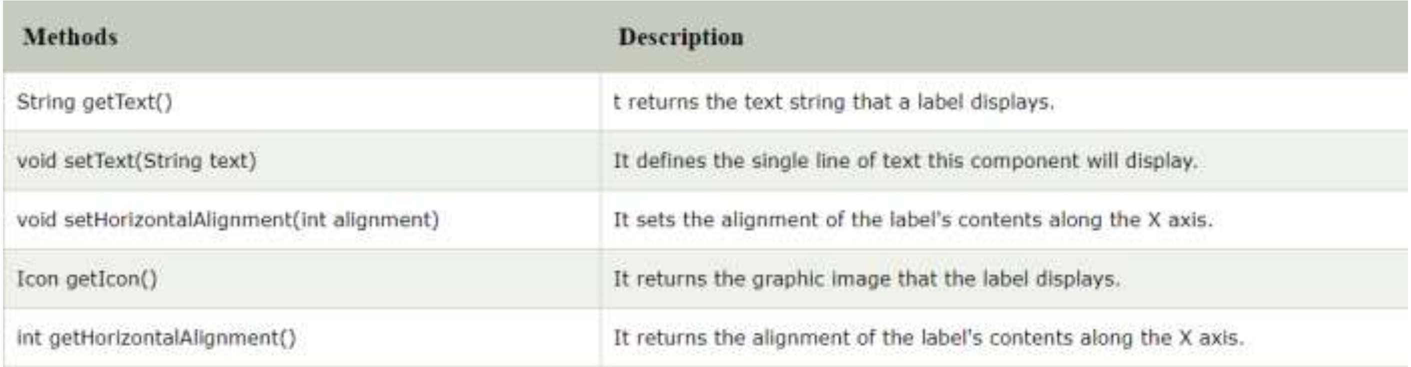
Frequently used constructors with this class are as follows:



Constructor	Description
JLabel()	Creates a JLabel instance with no image and with an empty string for the title.
JLabel(String s)	Creates a JLabel instance with the specified text.
JLabel(Icon i)	Creates a JLabel instance with the specified image.
JLabel(String s, Icon i, int horizontalAlignment)	Creates a JLabel instance with the specified text, image, and horizontal alignment.

Figure 3. 11 Constructors of JLabel in java

Frequently used methods with this class are as follows:



Methods	Description
String getText()	t returns the text string that a label displays.
void setText(String text)	It defines the single line of text this component will display.
void setHorizontalAlignment(int alignment)	It sets the alignment of the label's contents along the X axis.
Icon getIcon()	It returns the graphic image that the label displays.
int getHorizontalAlignment()	It returns the alignment of the label's contents along the X axis.

Figure 3. 12 Methods of JLabel in java

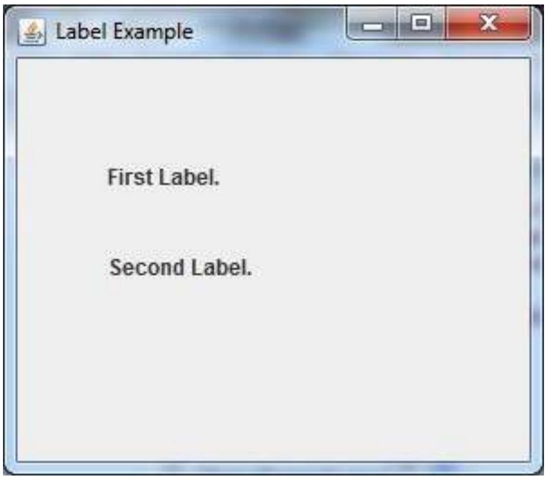
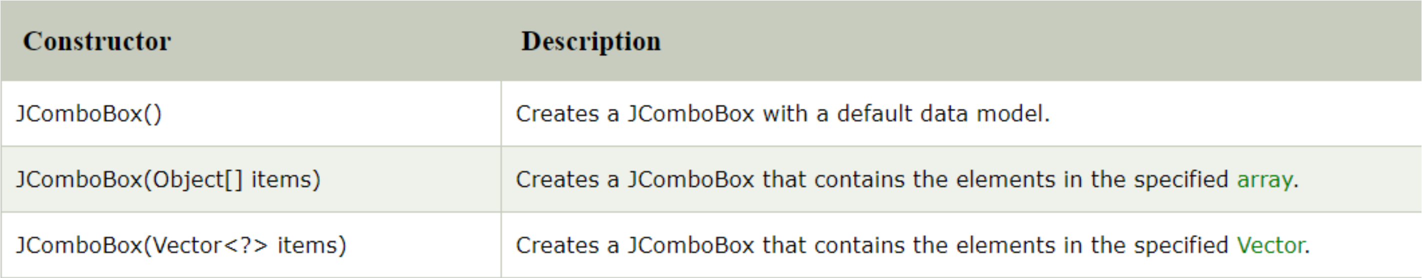


Figure 3. 13 JLabel in java

JCOMBOBOX IN JAVA: the pop-up menu of choices is shown as an object of the Choice class. The users choice is displayed at the top of the menu.

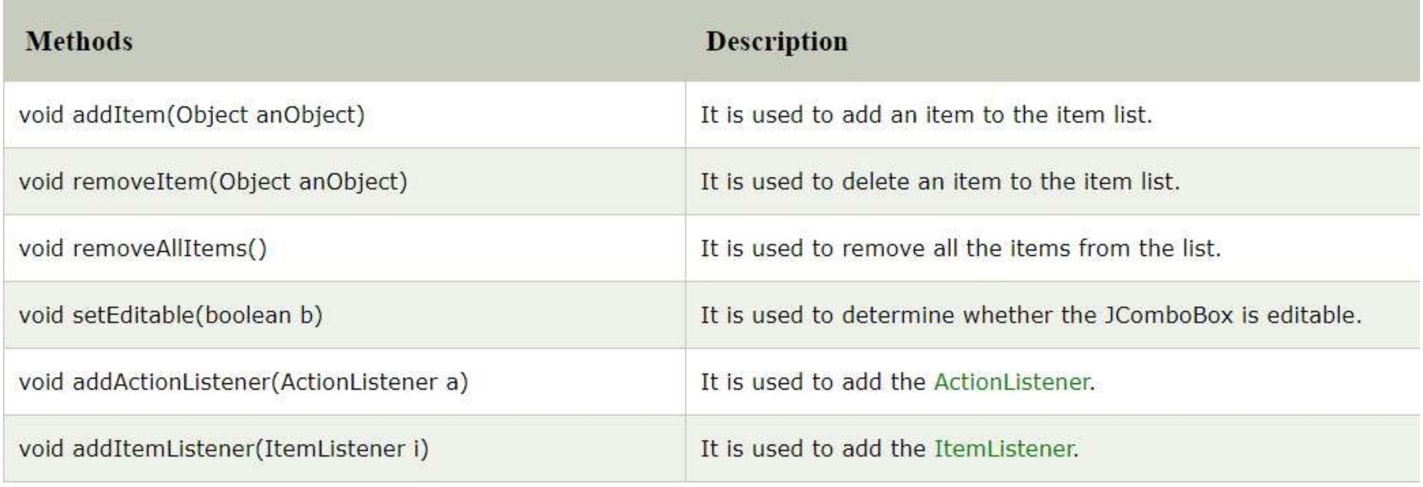
Frequently used constructors are as follows:



Constructor	Description
JComboBox()	Creates a JComboBox with a default data model.
JComboBox(Object[] items)	Creates a JComboBox that contains the elements in the specified array.
JComboBox(Vector<?> items)	Creates a JComboBox that contains the elements in the specified Vector.

Figure 3. 14 Constructors used with JCombobox in java

Frequently used methods are as follows:



Methods	Description
void addItem(Object anObject)	It is used to add an item to the item list.
void removeItem(Object anObject)	It is used to delete an item to the item list.
void removeAllItems()	It is used to remove all the items from the list.
void setEditable(boolean b)	It is used to determine whether the JComboBox is editable.
void addActionListener(ActionListener a)	It is used to add the ActionListener.
void addItemListener(ItemListener i)	It is used to add the ItemListener.

Figure 3. 15 Methods used with JCombobox

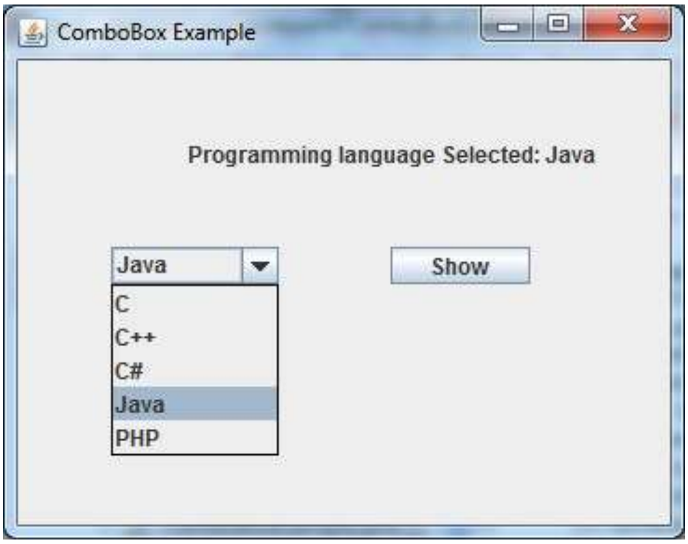
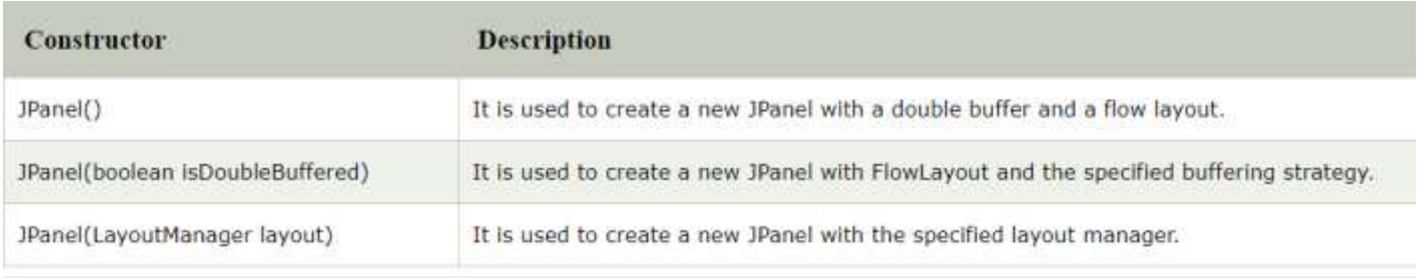


Figure 3. 16 JCombobox in java

JPANEL IN JAVA SWING: The most simple container class is the jpanel class. It offers a space where any application can ascribe to any other element. Also, it does not contain the title bar.

Frequently used constructors are as follows:



Constructor	Description
JPanel()	It is used to create a new JPanel with a double buffer and a flow layout.
JPanel(boolean isDoubleBuffered)	It is used to create a new JPanel with FlowLayout and the specified buffering strategy.
JPanel(LayoutManager layout)	It is used to create a new JPanel with the specified layout manager.

Figure 3. 17 Constructors used with JPanel

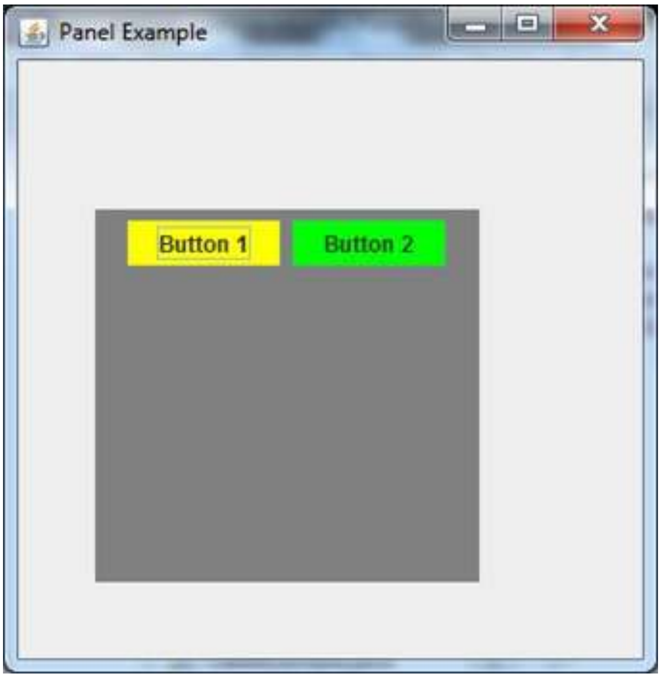
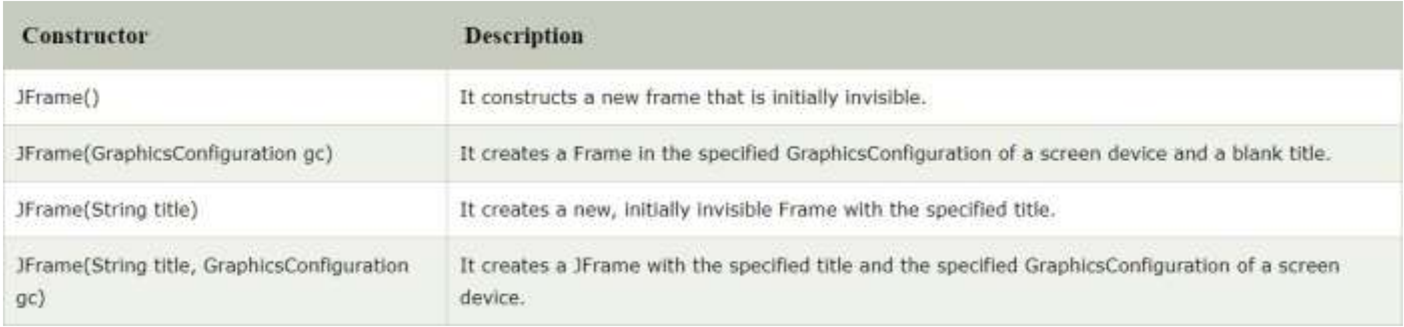


Figure 3. 18 JPanel in java

JFRAME IN JAVA SWING: the jframe is such a class which inherits the java.awt.frame class. It is the main window where all other components and elements like the buttons, labels etc. are situated to create a GUI. It also contains the feature of hiding and closing the window, unlike the frame class in AWT.

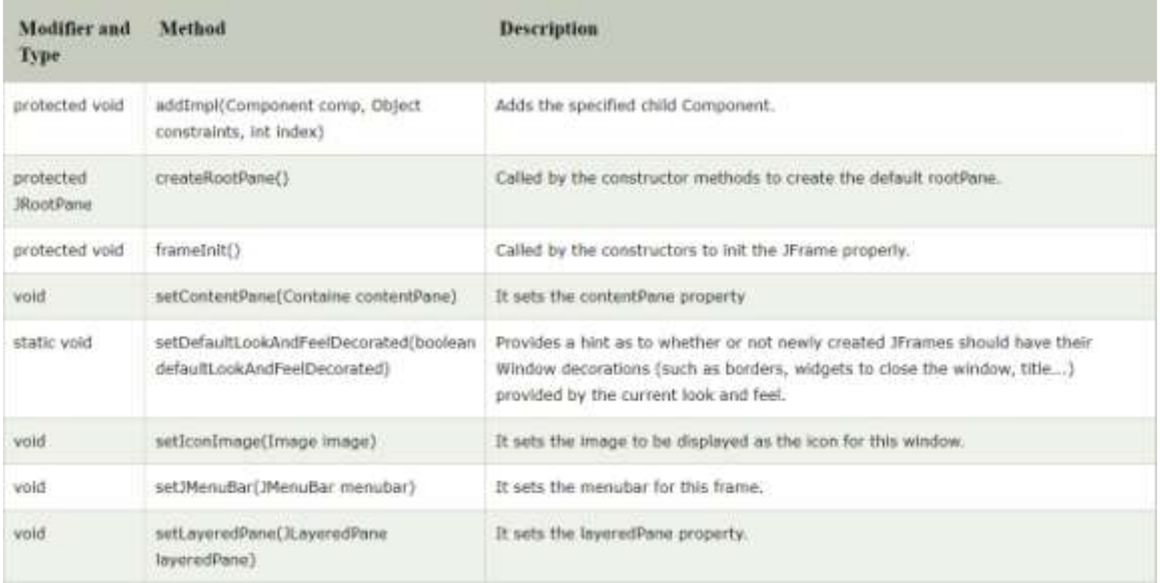
Frequently used constructors are as follows:



Constructor	Description
JFrame()	It constructs a new frame that is initially invisible.
JFrame(GraphicsConfiguration gc)	It creates a Frame in the specified GraphicsConfiguration of a screen device and a blank title.
JFrame(String title)	It creates a new, initially invisible Frame with the specified title.
JFrame(String title, GraphicsConfiguration gc)	It creates a JFrame with the specified title and the specified GraphicsConfiguration of a screen device.

Figure 3. 19 Constructors used with JFrame

Frequently used methods are as follows:



Modifier and Type	Method	Description
protected void	addImpl(Component comp, Object constraints, int index)	Adds the specified child Component.
protected JRootPane	createRootPane()	Called by the constructor methods to create the default rootPane.
protected void	frameInit()	Called by the constructors to init the JFrame properly.
void	setContentPane(Containe contentPane)	It sets the contentPane property
static void	setDefaultLookAndFeelDecorated(boolean defaultLookAndFeelDecorated)	Provides a hint as to whether or not newly created JFrames should have their Window decorations (such as borders, widgets to close the window, title...) provided by the current look and feel.
void	setIconImage(Image image)	It sets the image to be displayed as the icon for this window.
void	setJMenuBar(JMenuBar menubar)	It sets the menubar for this frame.
void	setLayeredPane(JLayeredPane layeredPane)	It sets the layeredPane property.

Figure 3. 20 Methods used with JFrame

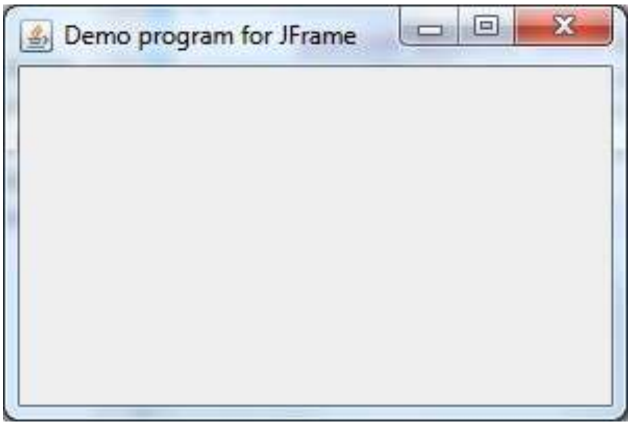
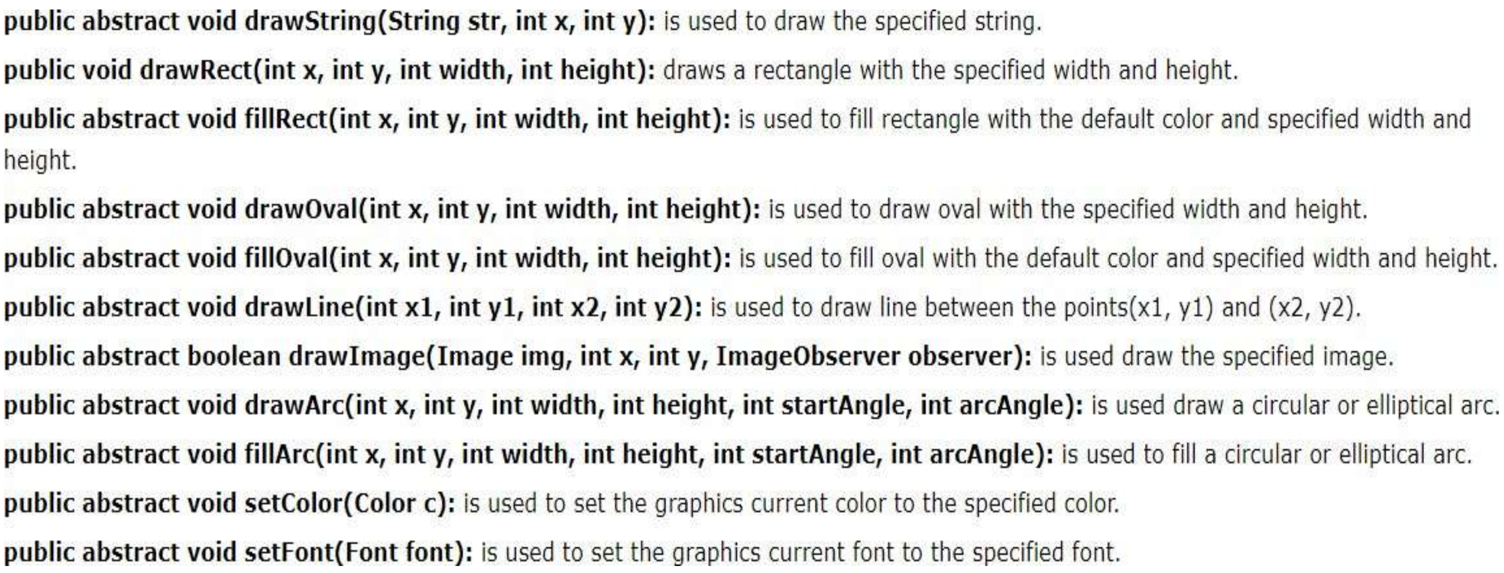


Figure 3. 21 JFrame in java

USAGE OF GRAPHICS IN SWING: implementation of graphics in java is done using the java.awt.Graphics class. The following methods are frequently used in implementation of graphics in java using swing class:



public abstract void drawString(String str, int x, int y): is used to draw the specified string.

public void drawRect(int x, int y, int width, int height): draws a rectangle with the specified width and height.

public abstract void fillRect(int x, int y, int width, int height): is used to fill rectangle with the default color and specified width and height.

public abstract void drawOval(int x, int y, int width, int height): is used to draw oval with the specified width and height.

public abstract void fillOval(int x, int y, int width, int height): is used to fill oval with the default color and specified width and height.

public abstract void drawLine(int x1, int y1, int x2, int y2): is used to draw line between the points(x1, y1) and (x2, y2).

public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer): is used draw the specified image.

public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle): is used draw a circular or elliptical arc.

public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle): is used to fill a circular or elliptical arc.

public abstract void setColor(Color c): is used to set the graphics current color to the specified color.

public abstract void setFont(Font font): is used to set the graphics current font to the specified font.

Figure 3. 22 Graphics implementation in java

The output of using the above methods for graphics implementation is as follows:

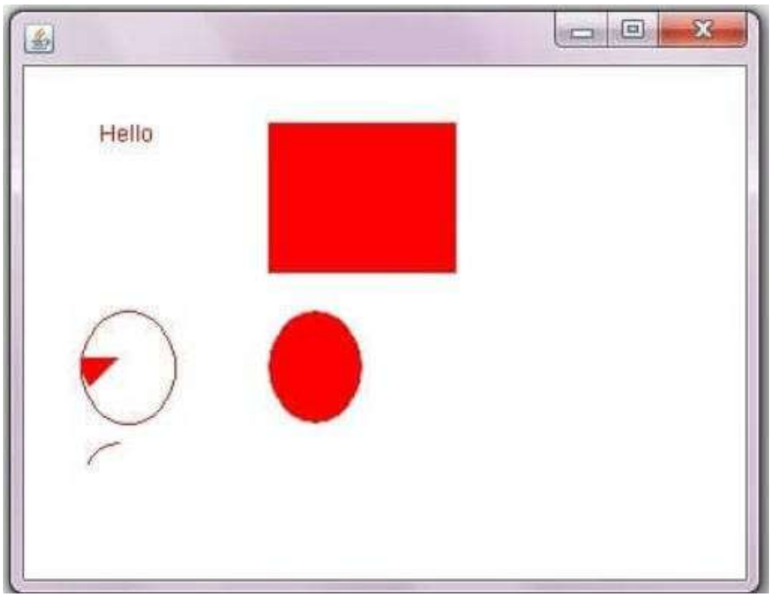
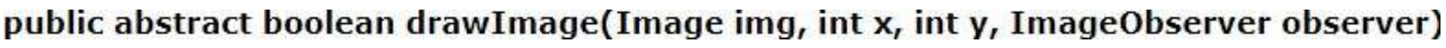


Figure 3. 23 Output of the above implementations

IMAGE DISPLAY IN SWING: an image can be displayed in java using the drawimage() method which is a part of the graphics class.

Syntax of this method is as follows:



```
public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer)
```

Figure 3. 24 Syntax of drawImage method

The result of using this method is shown as follows:

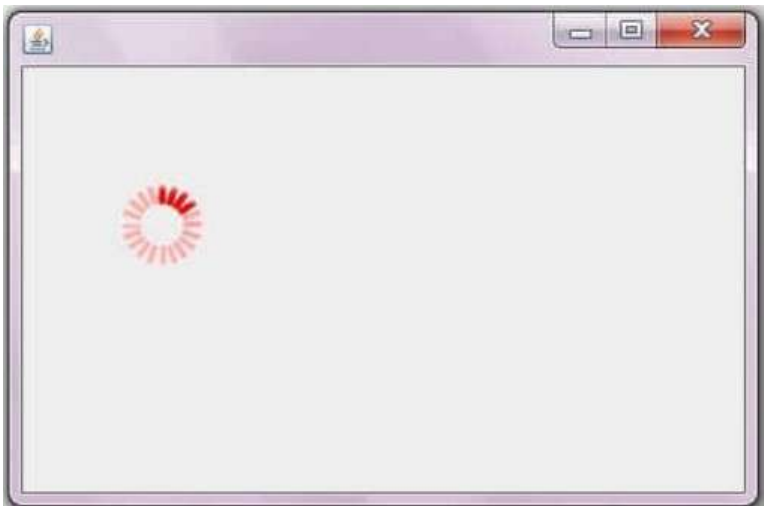
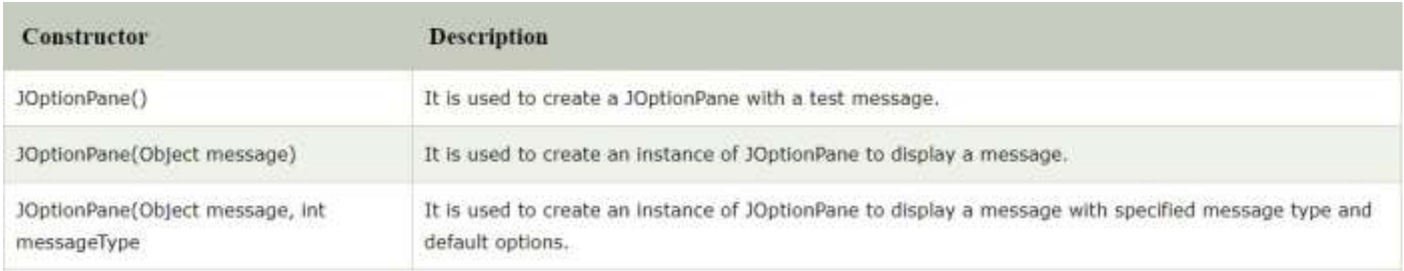


Figure 3. 25 Output of the above method

USAGE OF JOPTIONPANE IN SWING: using the Joptionpane in java, we can create and use the basic dialog boxes like confirm box, message box etc. They display some messages on the screen.

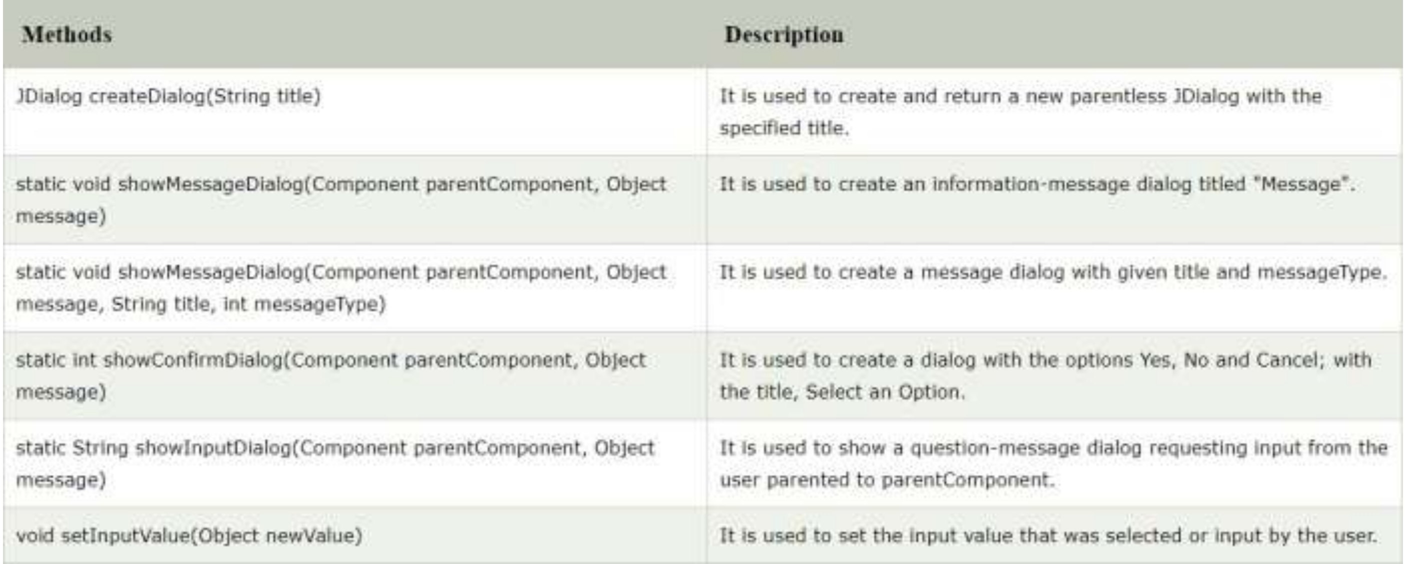
Frequently used constructors:



Constructor	Description
JOptionPane()	It is used to create a JOptionPane with a test message.
JOptionPane(Object message)	It is used to create an instance of JOptionPane to display a message.
JOptionPane(Object message, int messageType	It is used to create an instance of JOptionPane to display a message with specified message type and default options.

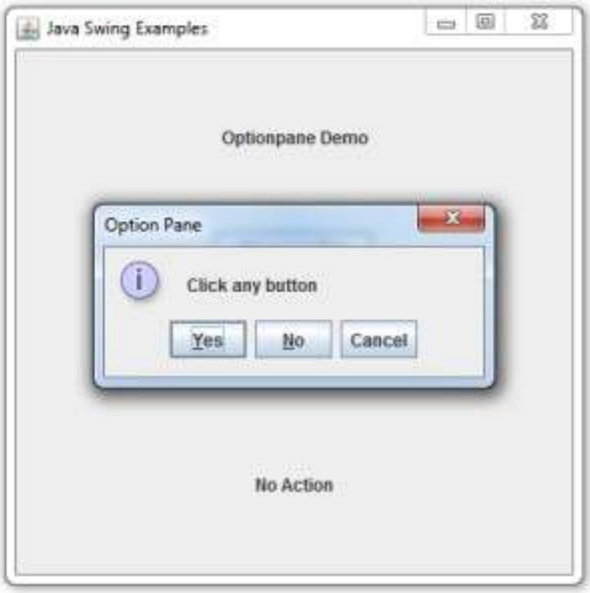
Figure 3. 26 Constructors used with JOptionPane

Frequently used methods:



Methods	Description
JDialog createDialog(String title)	It is used to create and return a new parentless JDialog with the specified title.
static void showMessageDialog(Component parentComponent, Object message)	It is used to create an information-message dialog titled "Message".
static void showMessageDialog(Component parentComponent, Object message, String title, int messageType)	It is used to create a message dialog with given title and messageType.
static int showConfirmDialog(Component parentComponent, Object message)	It is used to create a dialog with the options Yes, No and Cancel; with the title, Select an Option.
static String showInputDialog(Component parentComponent, Object message)	It is used to show a question-message dialog requesting input from the user parented to parentComponent.
void setInputValue(Object newValue)	It is used to set the input value that was selected or input by the user.

Figure 3. 27 Methods used with JOptionPane



Optionpane Demo

Click any button

No Action

Figure 3. 28 JOptionPane in java

3.4 COLLISION-DETECTION ALGORITHM IN 2D GAMES: collision - detection algorithms in 2D games depend on the shapes that are being collided. The shapes can be rectangular versus rectangular, circle versus circle and circle versus rectangle. Most of the times, we have convenient shapes hiding the shape known as a hitbox, so even when the collisions are not perfect by pixel-matching, it works just fine and gives a good performance measure across numerous units. [6]

There should be detection of objects colliding in some virtual environment, in collision detection. It is mainly used to refrain objects from passing through each other. Collision Detection can be found in every aspect of computer games: amongst characters, between characters and objects, etc. Collision detection implies the recognition of the joint of two or more objects.

There are two types of collision detection:

Continuous type: it is very costly in nature. It tries to simulate compact objects in actual life.

Discrete type: objects result in probing each other.

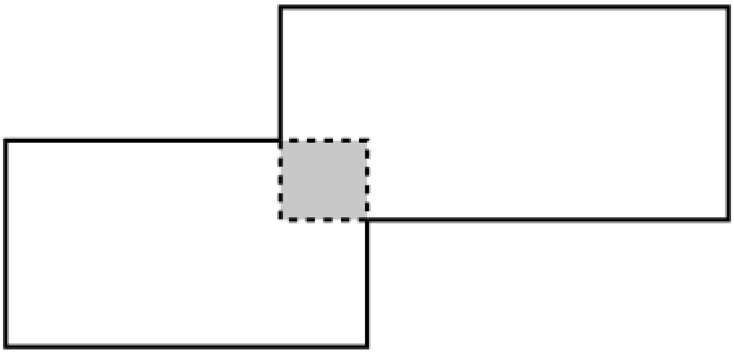


Figure 3. 29 Collision detection in games

AXIS-ALIGNED BOUNDING BOX (AABB): the more simple and easy forms of collision is when two rectangles collide when their axis are aligned toc each other, i.e. there is no rotation in the objects. This algorithm works only by making sure that there is no gap between any of the four sides of the rectangle. Gaps indicate that collision will not happen.

A bounding box or hitbox is a box that engulfs an entire object inside it. It can be characterized as a rectangular figure which has an x coordinate and a y coordinate, and some height and width too. A common manifestation of this exists in 2D games.

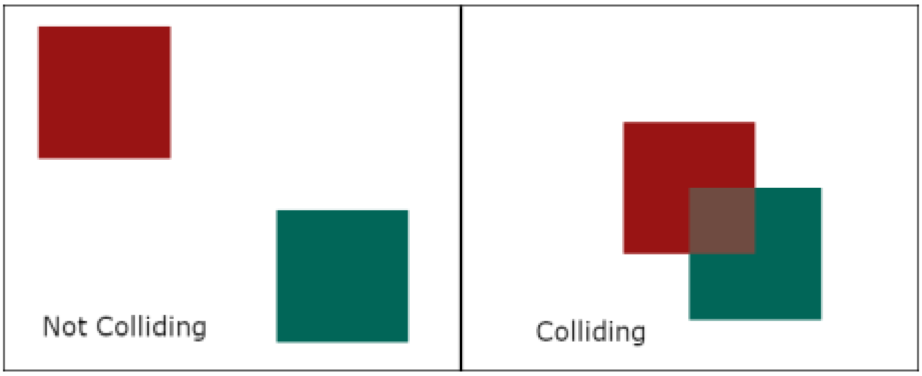


Figure 3. 30 Deciding the point of collision of objects

The following figure is an example of the bounding box used in games:

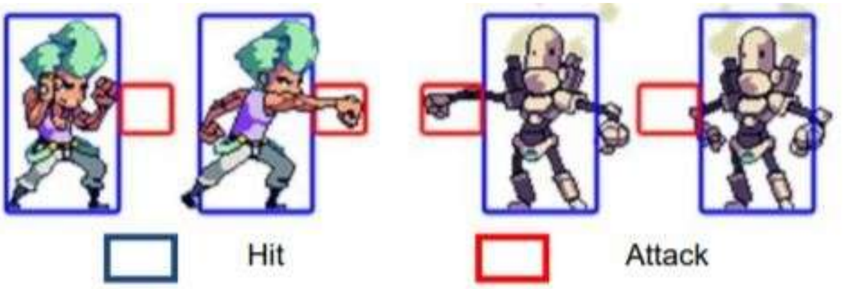


Figure 3. 31 Example figure of bounding box

Every object fashioned in the game has its self -bounding box. This technique is pretty easy as it only needs to equate whether an intersection occurs on any coordinate between the objects being collided. It is especially apt for rectangle objects. The area of the bounding box for the object can be specified by:

$$R\ area = \{ (x,y) \mid minx \le x \le maxx,\ miny \le y \le maxy \}$$

R area	=	area of bounding box collision
x, y	=	coordinate x, y
minx, miny	=	the minimum coordinate x, y
maxx, maxy	=	the maximum coordinate x, y

Figure 3. 32 Formula for Detecting area of collision in bounding box

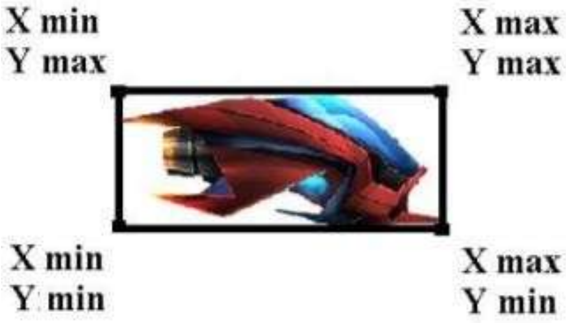
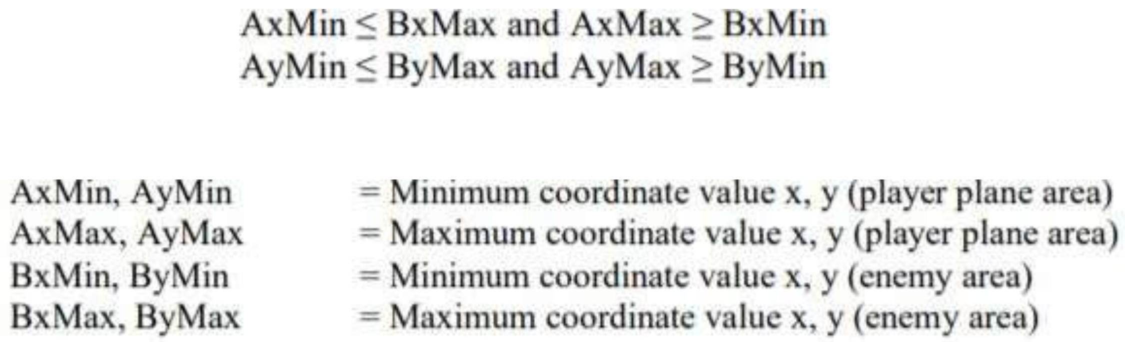


Figure 3. 33 Respective axis of an object

The areas that collide with each other will be made to disappear or will be muted. Testing of the collision is carried out by the comparison between the maximum and the minimum values of the x and y coordinates. Two colliding box object areas will be subdued if:



$$\text{AxMin} \leq \text{BxMax and AxMax} \geq \text{BxMin}$$
$$\text{AyMin} \leq \text{ByMax and AyMax} \geq \text{ByMin}$$

AxMin, AyMin	= Minimum coordinate value x, y (player plane area)
AxMax, AyMax	= Maximum coordinate value x, y (player plane area)
BxMin, ByMin	= Minimum coordinate value x, y (enemy area)
BxMax, ByMax	= Maximum coordinate value x, y (enemy area)

Figure 3. 34 Conditions for object collision using bounding box

A graphical representation can be shown as:

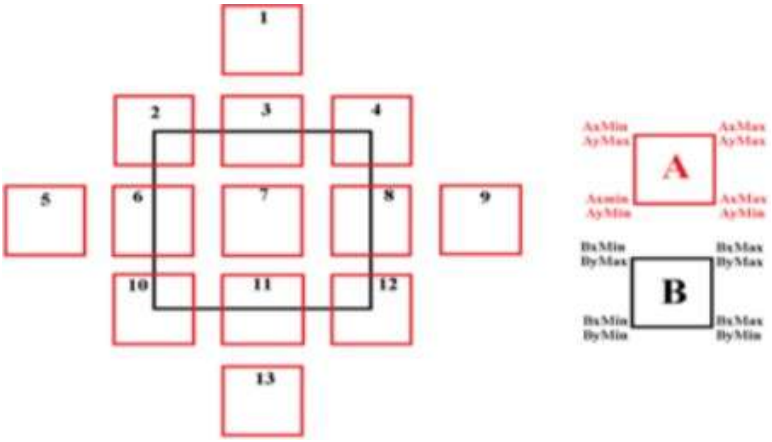


Figure 3. 35 Graphical representation of bounding box collision detection

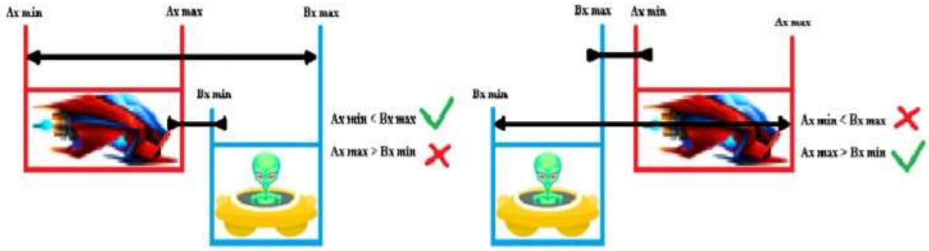


Figure 3. 36 Output of the above conditions inside a game

CHAPTER-4

PERFORMANCE ANALYSIS

The code is converted into a runnable .jar file. Then the game is run. The first window to pop up on the screen is the main window, called as the ‘welcome’ window, along with the background music. The opening window looks like the following:

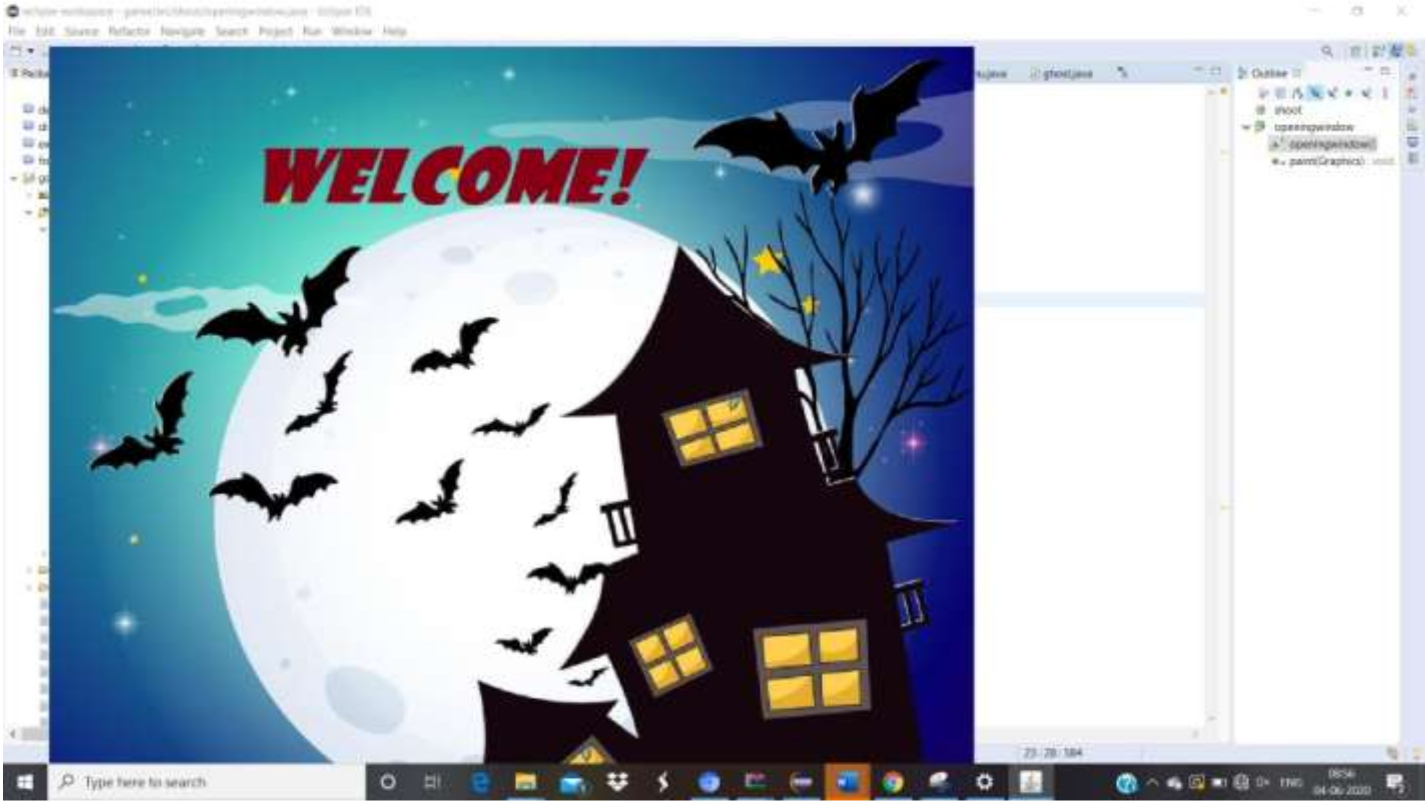


Figure 4. 1 Opening page of the game

This opening window is displayed only for 3 seconds, and after that, the next window pops up. It contains the PLAY button on the top, clicking on which will take the player to the next window. It also has the EXIT button at the bottom, in case the user wishes to back out and quit the game. Thebackground music still keeps playing. Various kind of images and graphics are also displayed in this window, according to theme of the game.

This window looks like the following:

The green button on the top is the PLAY GAME button, and the red cross button in the bottom is the QUIT GAME button. The rest of the window contains images and graphics.

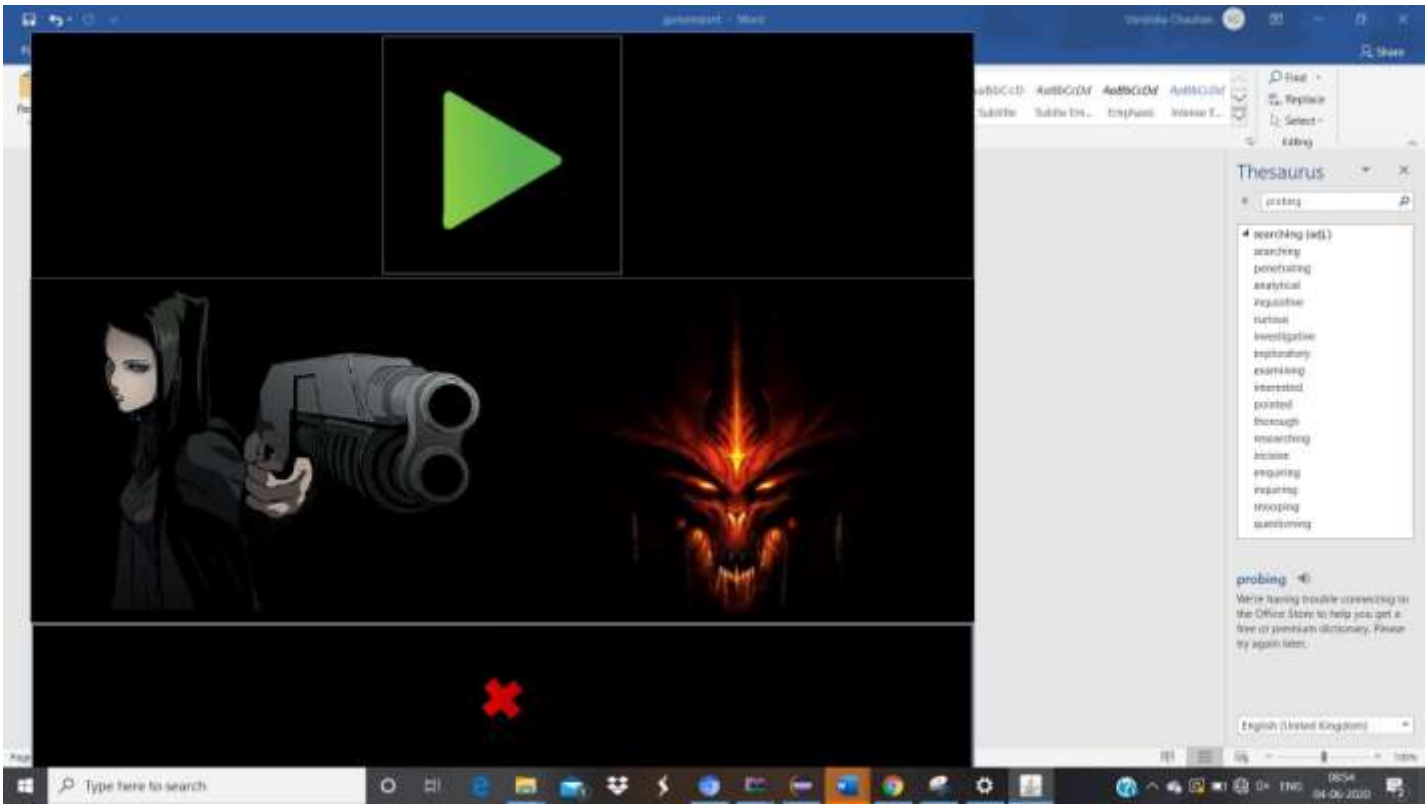


Figure 4. 2 The window for playing or quitting the game

Assuming that the player decides to play the game, and clicks on the PLAY button. Then the player is directed to the MENU page. The MENU page provides the different levels of the game. The game has three levels:

The GHOST level

The EVIL level

The DRAGON level

The player can choose any level to play, but the progression of the game works in the above fashion. The player can also click on the back button if they want to go back and quit the game. The player has to click on the screen to begin playing the selected level of the game. The game progresses from an easy level to a comparatively complex level as the player moves from level to level. The difficulty of the levels will rise accordingly, as and when the game proceeds.

The following is the MENU page:

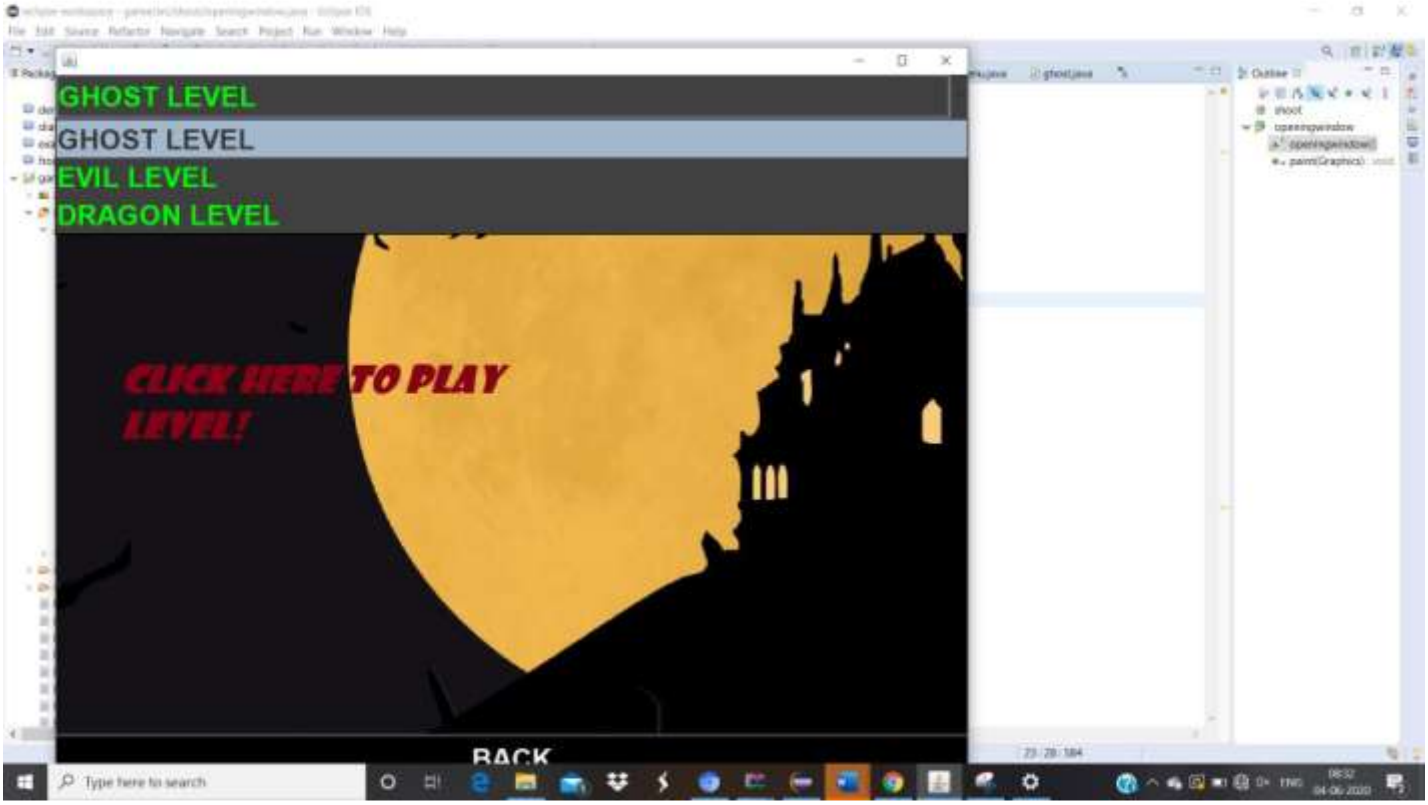


Figure 4. 3 The menu of choices for the player

If the player wishes to quit, he simply clicks on the BACK button, and the game returns to the following screen:

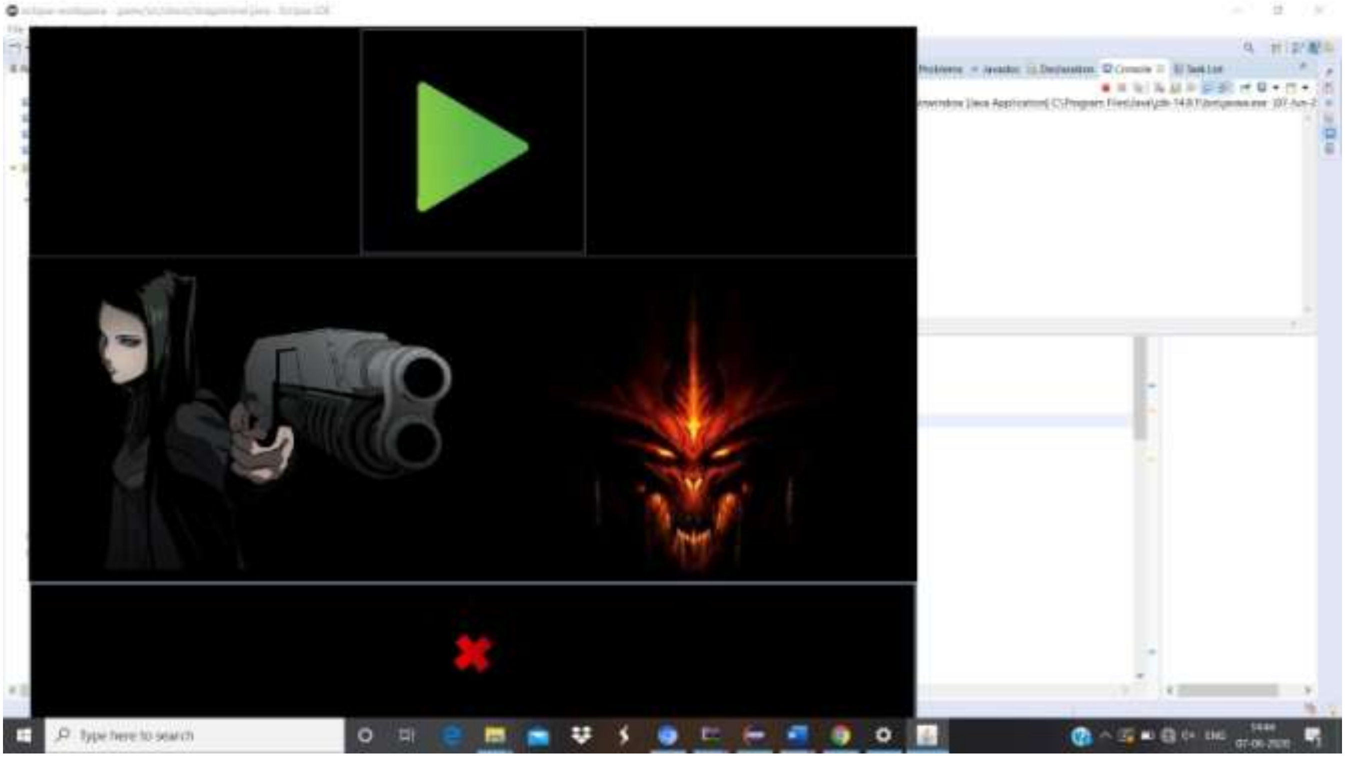


Figure 4. 4 The Back button takes back to this window

4.1 GHOST LEVEL: the ghost level is the first level of the game. The player, represented as a gun, has to fire and kill at the ghosts dropping from above, and has to kill a minimum of certain number of ghosts to win and cross over to the next level. The dropping speed of the ghosts and the fire speed of the bullets is also defined. The gun firing the bullets is movable in the x-coordinate. A limit of 1000 bullets is provided to the player to kill the ghosts.

After the ghosts drop to a certain level on the y coordinate, the game is over. If the player is able to get the required score by that time, the game moves to the next level with the display of the dialog boxes indicating the above.

Otherwise, the same level resumes again, and the score is also refreshed. The score is displayed at the right bottom corner of the screen. The sounds are playing in the background.

The GHOST level looks like the following:

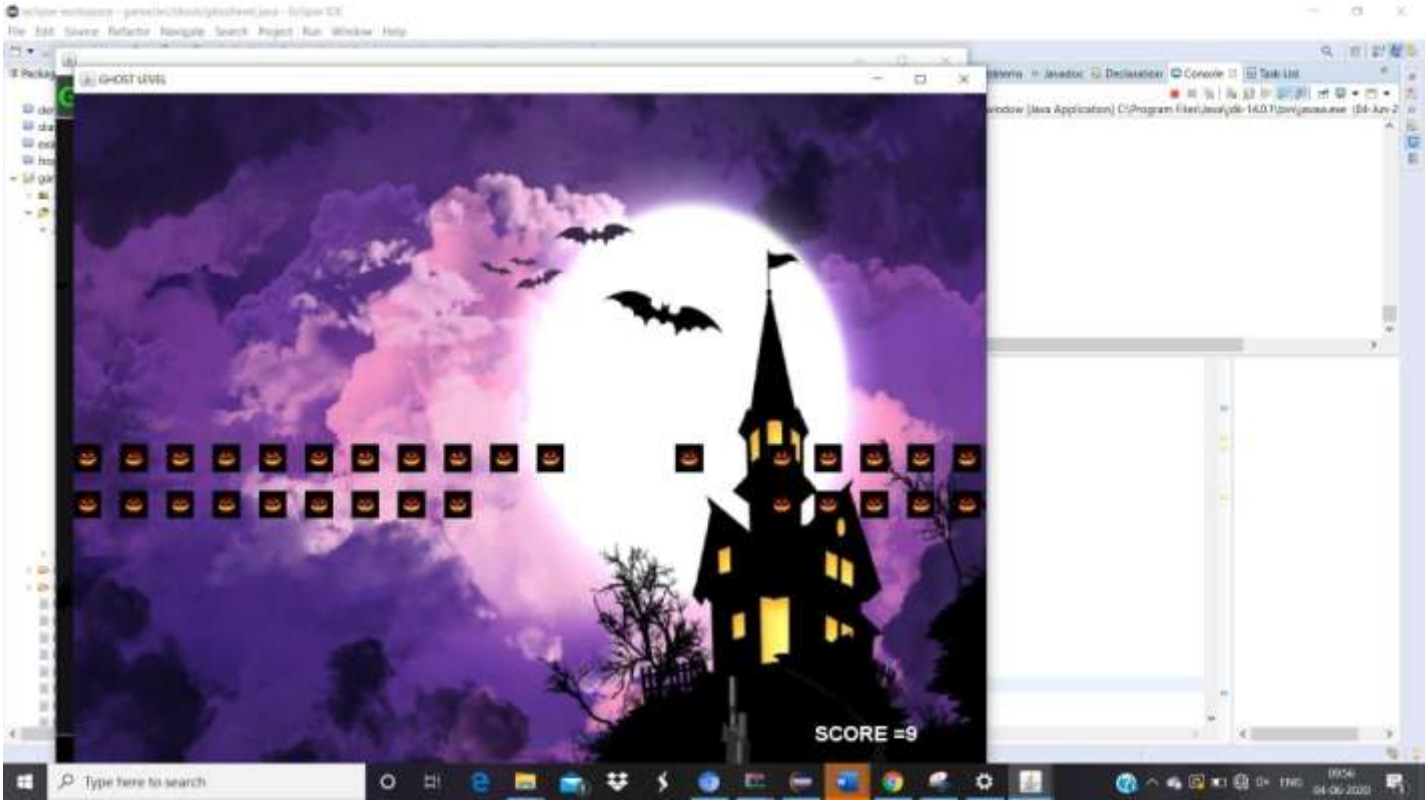
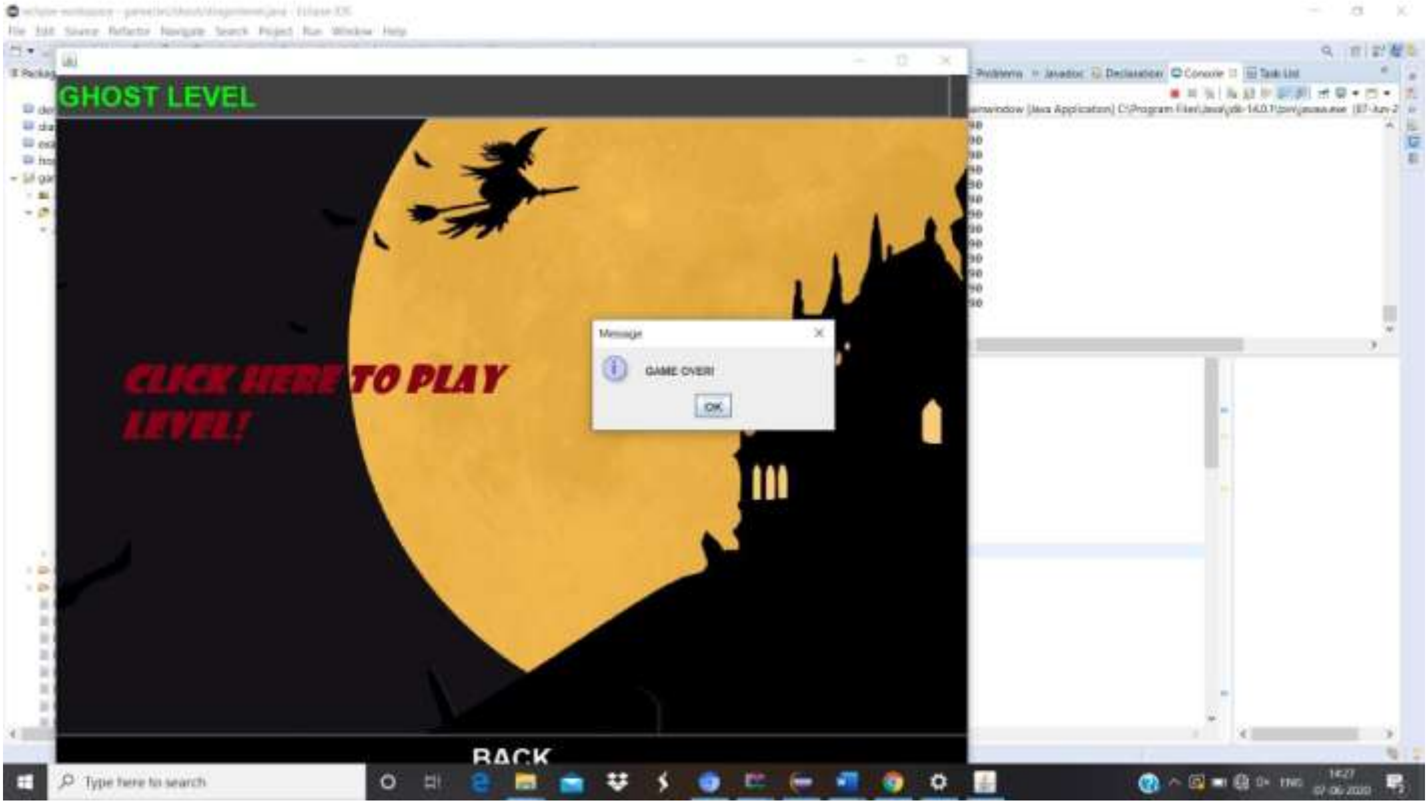


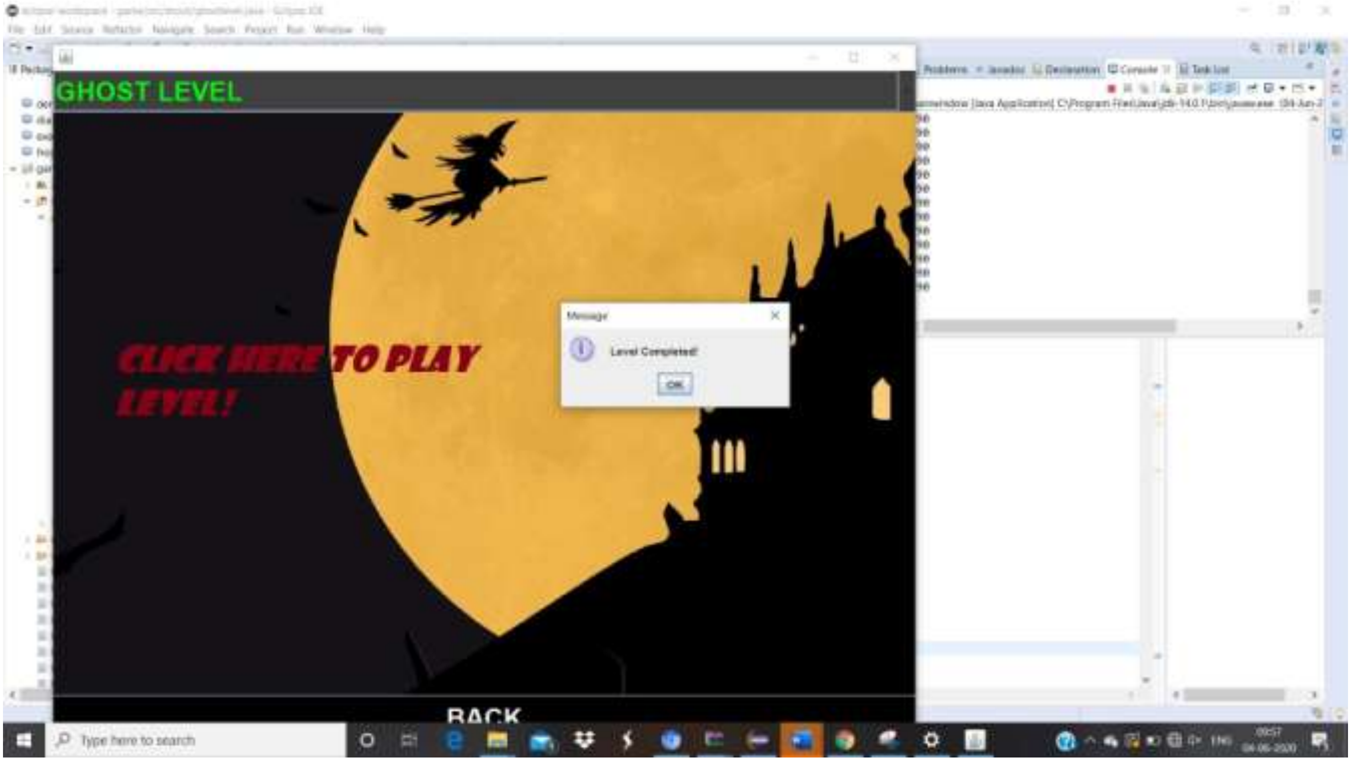
Figure 4. 5 The Ghost Level of the game

If the game is lost by not making the required score, the following screen is displayed:



GAME OVER!

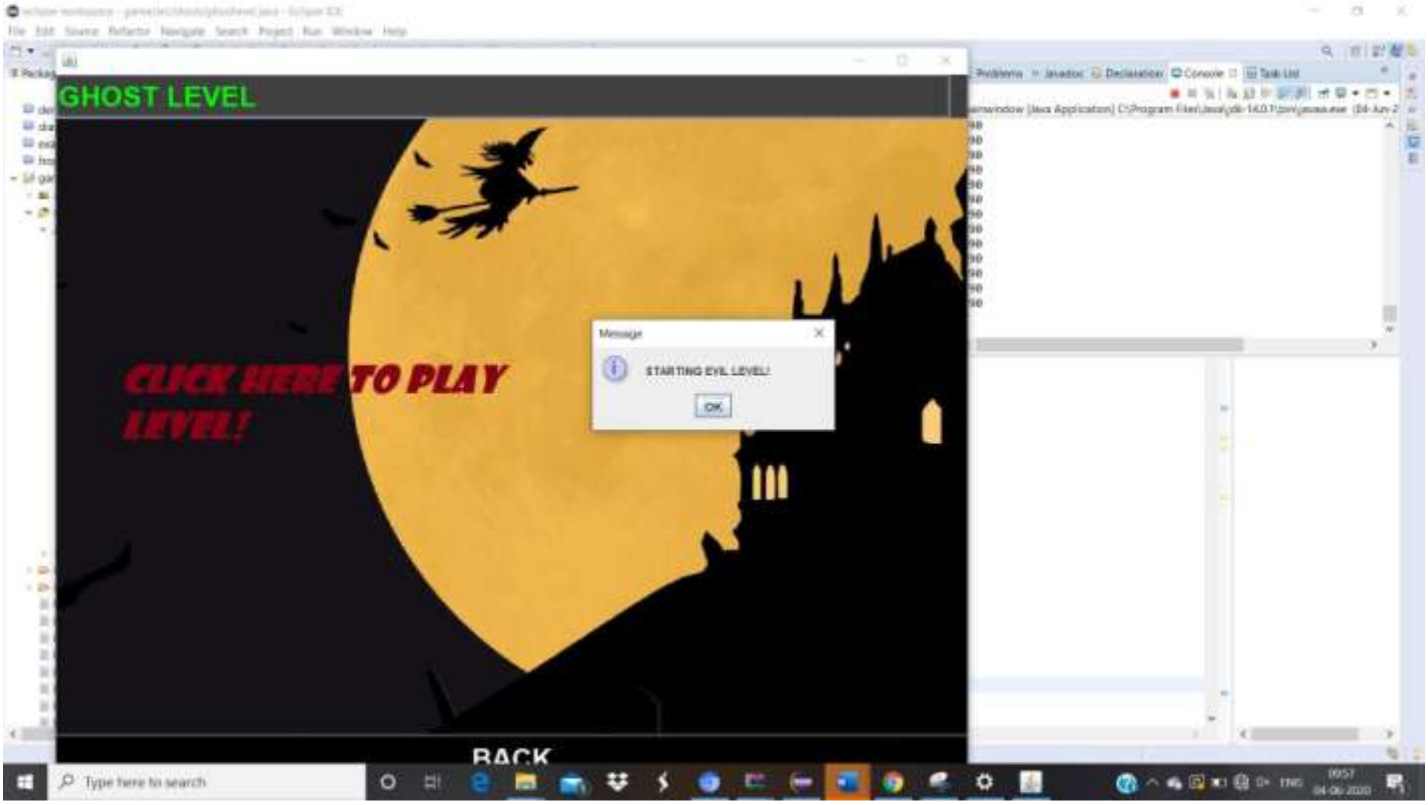
Figure 4. 6 This message pops up if the level is lost



Level Completed!

Figure 4. 7 Message displayed if the level is won

However, if the level is won by meeting the required score value, then the following dialog box is displayed and the next level resumes:



STARTING EVIL LEVEL!

Figure 4. 8 Message before next level begins

4.2 EVIL LEVEL: the evil level is the second level of the game. It is itself initiated if the player wins ghost level. A dialog box is displayed indicating the start of the evil level. The player, represented as a gun, has to fire and kill at the ghosts dropping from above, and has to kill a minimum of certain number of ghosts to win and cross over to the next level. The dropping speed of the ghosts and the fire speed of the bullets is also defined. The gun firing the bullets is movable in the x-coordinate. A limit of 100 bullets is provided to the player to kill the ghosts.

This level is tougher than ghost level as the minimum score needed to move to the next level is higher than ghost level, the rate at which the ghosts drop from above is faster and the number of bullets provided to kill the ghosts is reduced from 1000 to 100.

After the ghosts drop to a certain level on the y coordinate, the game is over. If the player is able to get the required score by that time, the game moves to the next level with the display of the dialog boxes indicating the above.

Otherwise, the same level resumes again, and the score is also refreshed. The score is displayed at the right bottom corner of the screen. The sounds are playing in the background.

The EVIL level looks like the following:

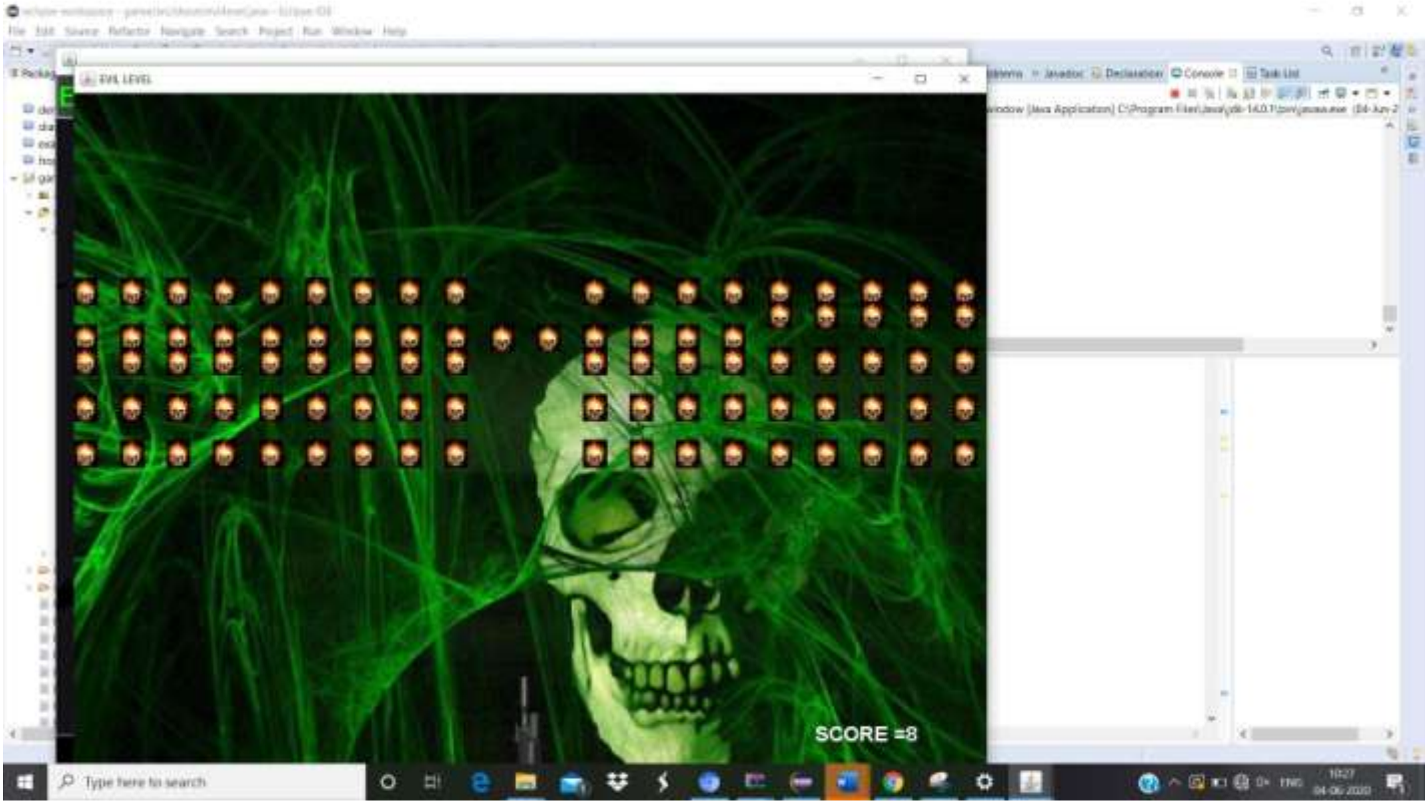
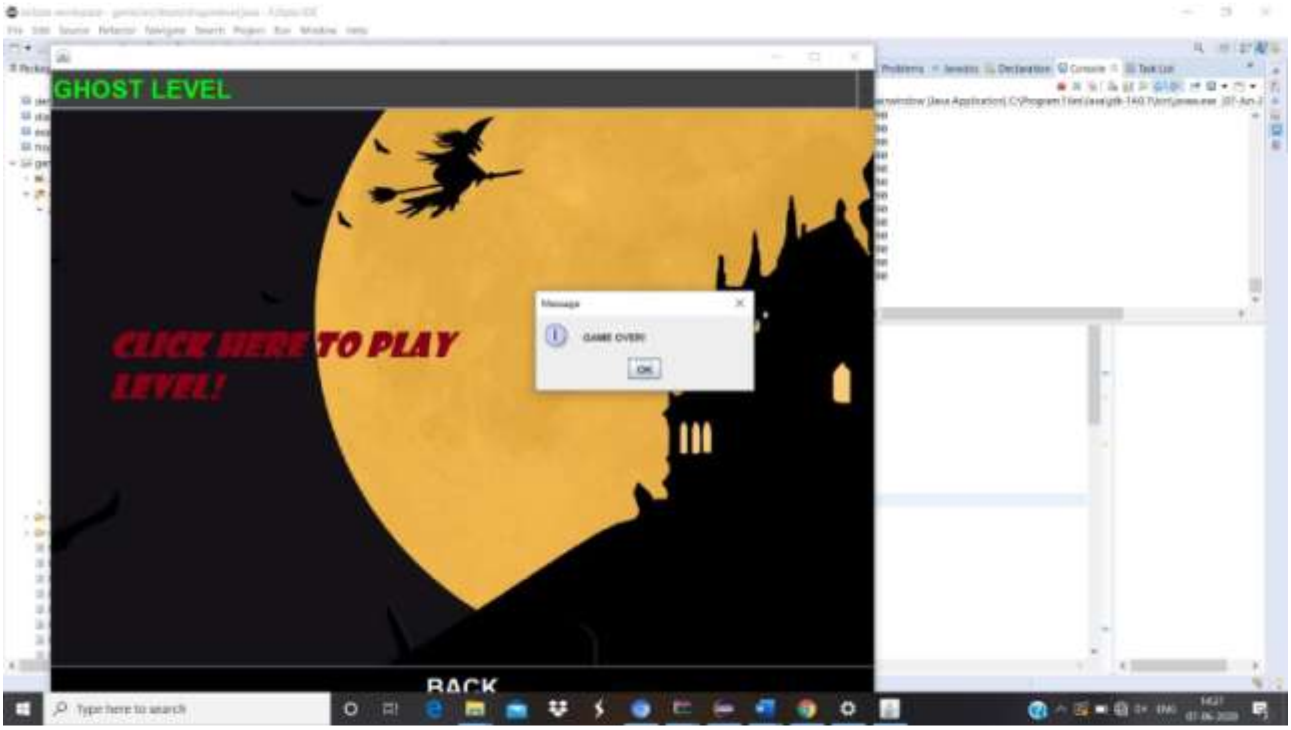


Figure 4. 9 The Evil Level

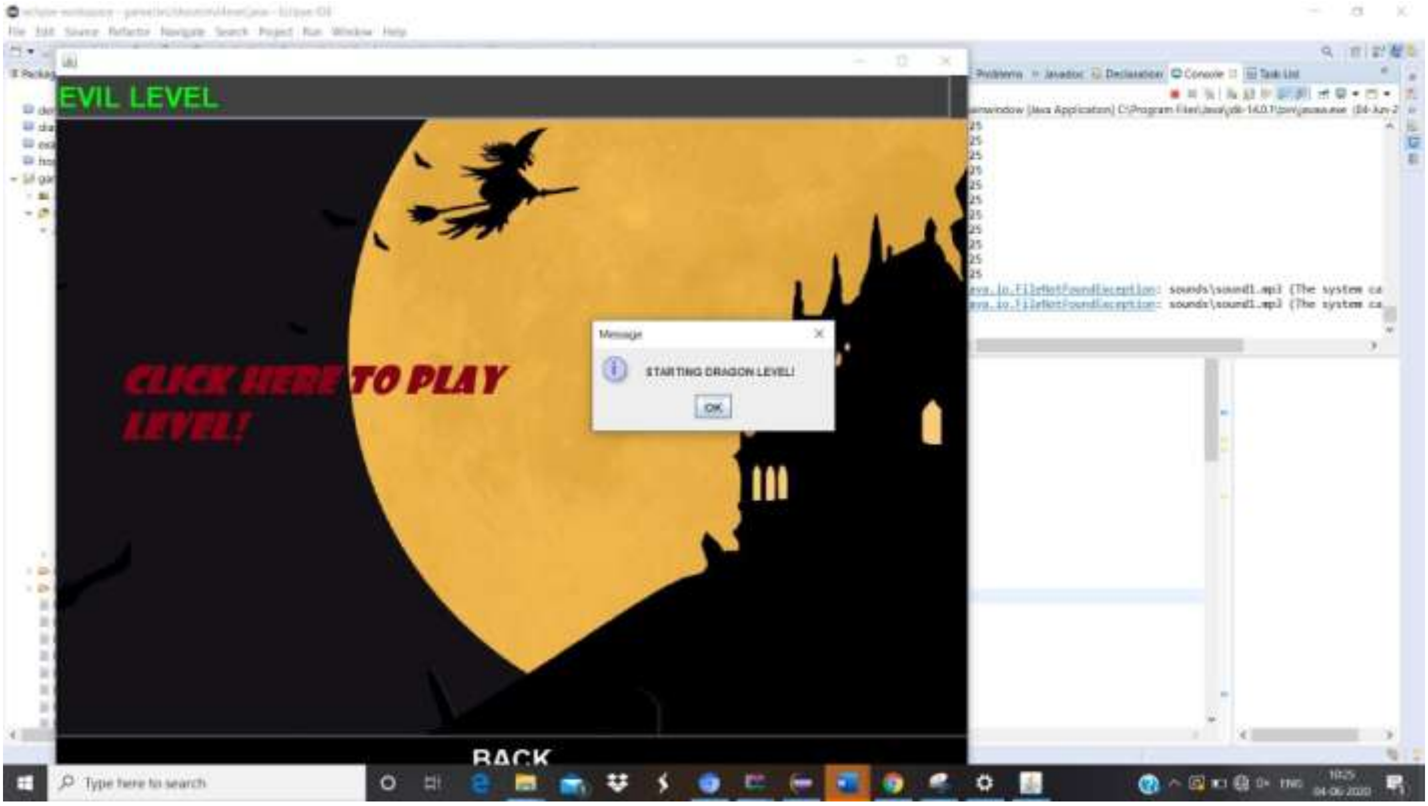
If the level is lost, the following dialog box is displayed:



GAME OVER!

Figure 4. 10 Message displayed if the game is lost

However, if the game is won, the following dialog box is displayed and the next level begins:



STARTING DRAGON LEVEL!

```
...window [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (04-Jun-2
25
25
25
25
25
25
25
25
25
25
25
java.io.FileNotFoundException: sounds\sound1.mp3 (The system ca
java.io.FileNotFoundException: sounds\sound1.mp3 (The system ca
```

Figure 4. 11 Message before the next level begins

4.3 DRAGON LEVEL: the last level of the game is the dragon level. It is itself initiated if the player wins evil level. A dialog box is displayed indicating the start of the dragon level. The player, represented as a gun, has to fire and kill at the ghosts dropping from above, and has to kill a minimum of certain number of ghosts to win and cross over to the next level. The dropping speed of the ghosts and the fire speed of the bullets is also defined. The gun firing the bullets is movable in the x-coordinate. A limit of 100 bullets is provided to the player to kill the ghosts.

This level is tougher than ghost level as the minimum score needed to move to the next level is higher than ghost level, the rate at which the ghosts drop from above is faster and the number of bullets provided to kill the ghosts is reduced from 100 to 80.

After the ghosts drop to a certain level on the y coordinate, the game is over. If the player is able to get the required score by that time, the game displays the dialog box with the message of game over and the game stops running.

Otherwise, the same level resumes again, and the score is also refreshed. The score is displayed at the right bottom corner of the screen. The sounds are playing in the background.

The DRAGON level looks like the following:

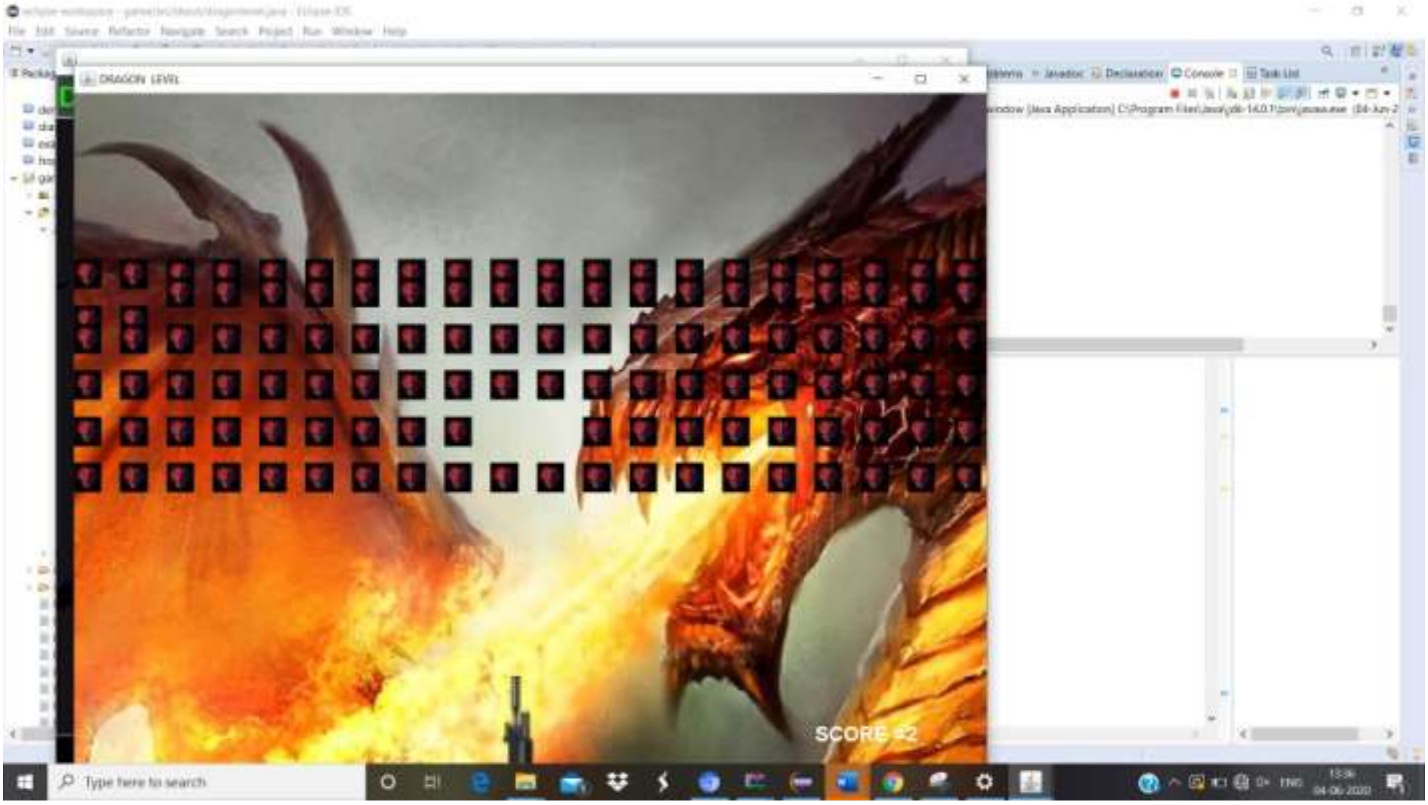
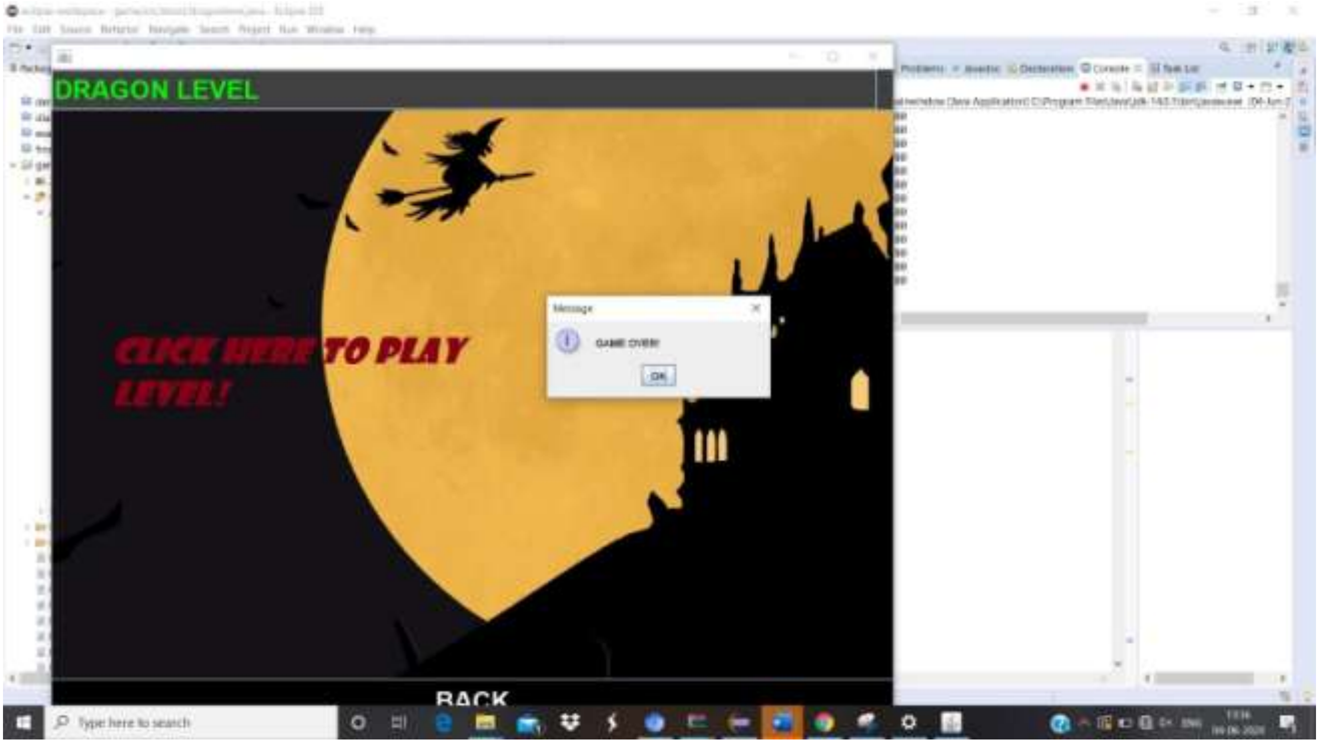


Figure 4. 12 The Dragon Level

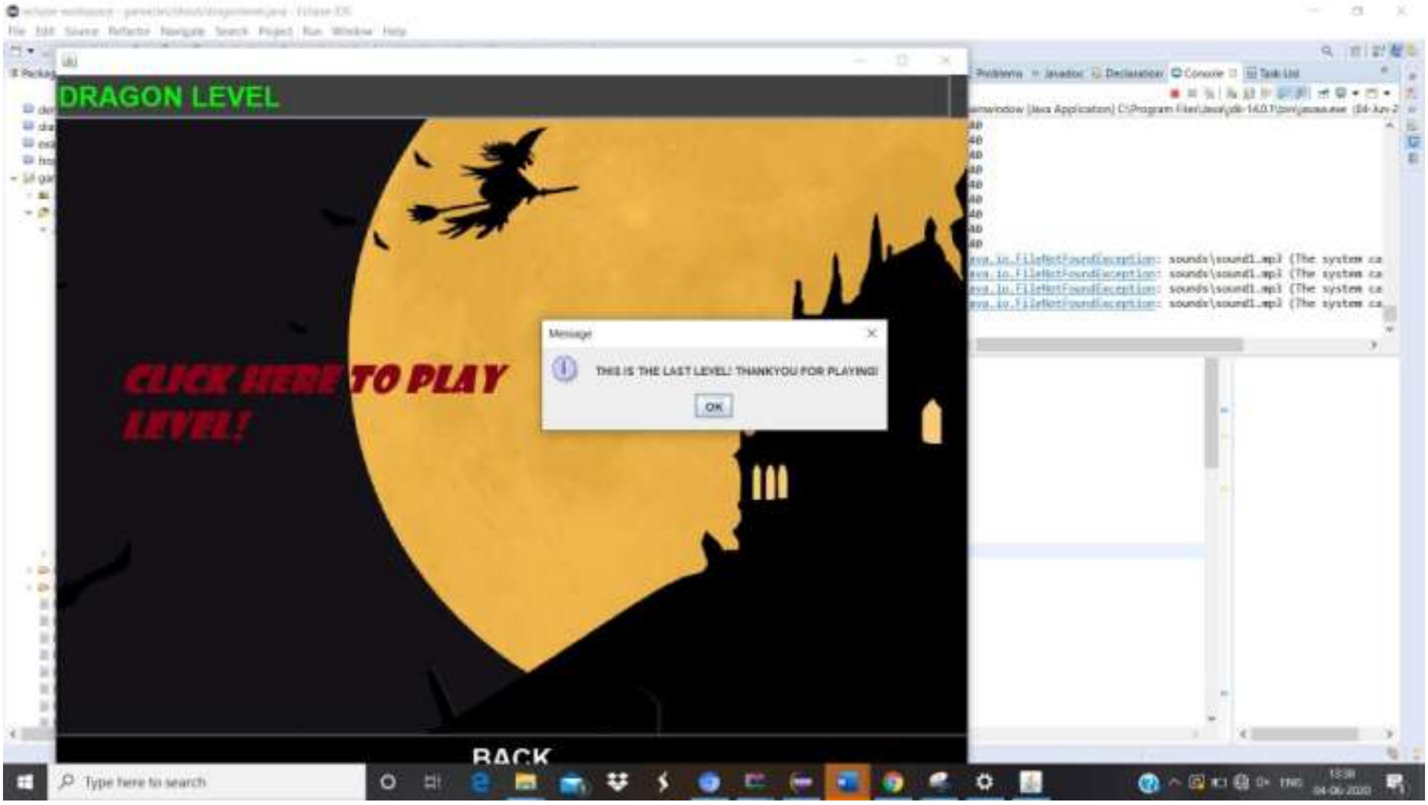
If the player loses the game the following is displayed:



GAME OVER!

Figure 4. 13 Message displayed if the level is lost

If the player wins the last level, the following message is displayed:



Message

THIS IS THE LAST LEVEL! THANKYOU FOR PLAYING!

```
40
40
40
40
40
40
40
40
40
ava.io.FileNotFoundException: sounds\sound1.mp3 (The system ca
ava.io.FileNotFoundException: sounds\sound1.mp3 (The system ca
ava.io.FileNotFoundException: sounds\sound1.mp3 (The system ca
ava.io.FileNotFoundException: sounds\sound1.mp3 (The system ca
```

Figure 4. 14 Message displayed after the level is won

The test results of the system are now summarized in a tabular format alongside the procedures for testing, the given inputs, the outputs we expected, the actual output we obtained, and the final conclusions.

The table containing the above is given as follows:

MODULE TESTED	PROCEDURE	INPUT	EXPECTED OUTPUT	RESULT	CONCLUSION
Opening window	To open the game	Pressing the game .jar file	-Showing opening window -Playing background music	-Shows opening window -plays background music	Valid

Playing/quitting	Open the app Press on icon	Press play/quit button	Performs the action associated with the button	Either moves to the next window or quits the game	Valid
Menu	Choosing the level to play	Selecting the level to play	Starts the chosen game level	Will begin the selected level to play	Valid
Ghost level	Fire bullets to kill ghosts	Pressing the space key. Move the gun using the navigation keys.	The ghosts hit by the bullets will disappear and score will increment. If the level is not complete, re-start again. If the level completes, the next level starts.	Score gets updated whenever bullet hits ghost. Failing the level re-starts it. Winning moves to next level.	Valid
Evil level	Fire bullets to kill ghosts	Press the space key to fire bullets. Move the gun using the navigation keys.	The ghosts hit by the bullets will disappear and score will increment. If the level is not complete, re-start again. If the level completes, the next level starts.	Score gets updated whenever bullet hits ghost. Failing the level re-starts it. Winning moves to next level.	Valid

Dragon level	Fire bullets to kill ghosts	Press the space key to fire bullets. Move the gun using the navigation keys.	The ghosts hit by the bullets will disappear and score will increment. If the level is not complete, re-start again. If the level completes, a dialog box should be displayed indicating end of game.	Score gets updated whenever bullet hits ghost. Failing the level re-starts it. Winning displays required dialog box.	Valid
Background music	Plays alongside the entire game	Automatic sound when game starts	Music should play	Music plays	Valid
Exit/quit/back	Open game	Choose exit/quit/back	Button should exit/quit/go back	Button performs the action of exit/quit/go back	Valid

Table 4. 1 Performance analysis of the game developed

The following objectives have been covered in the game developed:

It is a single-player game.

It is a PC based game, a .jar file is run to play the game.

The game has been developed on a 2- dimensional platform.

The game has a total of three levels in it.

It is an adventure based game where the player has to shoot the ghosts to proceed to the next level.

It makes usage of the 2D platform for graphical user interface and different menu systems.

It has been written in java with the usage of various java packages like AWT engine, java Swing, etc.

Back -end used is JDK and front-end used is 2D java GUI.

CHAPTER-5

CONCLUSION

5.1 CONCLUSION

Technological developments have been increasing rapidly and supplementing daily needs of our lives. Despite of the fact that computer games have entertainment as their prime focus, they are still the lead of computer sciences.

Game technology is rapidly increasing and is one of the most sought- after research areas

because it has the immense potential of scientific knowledge and commercial success.

Experimentally it can be said that the game created works efficiently on the collision- detection method. The objects in the game are given a bounding box or hit-box and pre-determined effects can be seen on the collision between the bullets and the objects.

The game was made using java programming in Eclipse IDE purely.

The world has become a close knitted place due to the globalisation and the world wide web with the internet bringing the world almost on a single platform. With this advancement in technology, people have become more or less dependent on virtual aspects relating to almost everything ranging from their daily lives to things which require scientific implementation.

The era of meeting up with people in real life is also undergoing this shift. As more people try to build friendships based on a single click of the mouse, the world of virtual gaming provides an immense opportunity to all the game geeks.

Earlier, the world of gaming was restricted to mere enjoyment purposes and a means of entertainment. The situation has had a role reversal in the past recent years with the platform for online gaming being turned into a professional work for earning name, fame and money.

Various tech giants have invested in the business of game development using numerous software and this field has thus attracted youth in a large number. Not only is this market of online gaming providing an opportunity to the people in the world, it is also serving as a means for the students to enrol into professional courses which include learning various computer languages and developing new software. The young minds with basic needs and guidance are making best use of their talent to make use in expanding this gaming platform.

Along with these developments, the unlimited number of online games being provided for the billion population of the world to play with while suiting their own taste and leisurely needs, has become one of the basic source of entertainment and avoid boring hours. Online gaming has also served as a blessing in disguise to the socially introverts.

The purpose of this game creation was to create a source of entertainment, and to create an easy game that can be played by people of all age groups. By developing this game, I also learnt the practical application of the concepts of java, and how the language is of importance in the real world of computer science.

The game developed, henceforth, is a good insight for people wanting to get into game developing and who are just beginning to learn the implementation of java in the gaming concepts. Also, for a booming success in the gaming industry, for the purpose of entertainment and commercial value, the development and creation of games like this is significant and important and of real-world value and demand.

5.2 FUTURE SCOPE:

The future of game development is immense. Firstly, a 2D game can be made into a 3D game, by working on the graphics and game dimensions. The game is only a single-player right now, but it can be made real-time multi-player by including the concept of networking.

In the game, new concepts can be introduced at every level, new challenges and bonus scores and gifts can be made to achieve by the player. The concept of movement in the ghosts and surprise attacks can make the game more exciting.

Basically, the game under development can improve on the following current features:

The game is single player, and can be made online multi-player for the future.

The current game is PC based, but it can be improvised to an application that is playable on android mobile phones.

It is a 2-dimensional game and can be made into a 3-dimensional game.

It has only 3 levels as of now, which can be incremented to more levels with increasing concepts and graphics.

It is an adventure-based game and has a huge scope of improvement.

It uses a 2D platform for graphical user interface and other systems.

The above given improvements can be made to the game developed.

REFERENCES

[1] Bill Buchanan (1999), Handbook of Data Communications and Network- Springer Nature

[2] JavatPoint website, https://www.javatpoint.java-awt

[3] JavatPoint website, https://www.javatpoint.java-awt

[4] JavatPoint website, https://www.javatpoint.java-jframe

[5] JavatPoint website, https://www.javatpoint.java-keylistener

[6] Hadikurniawati, Wiwien & Winarno, Edy & Cahyono, Taufiq & Abdullah, Dahlan. (2018). A Bounding Box Collision Detection Method Based On The Android System For Adventure Game. Journal of Physics: Conference Series. 1114. 012099. 10.1088/1742-6596/1114/1/012099.

van

by Van Van

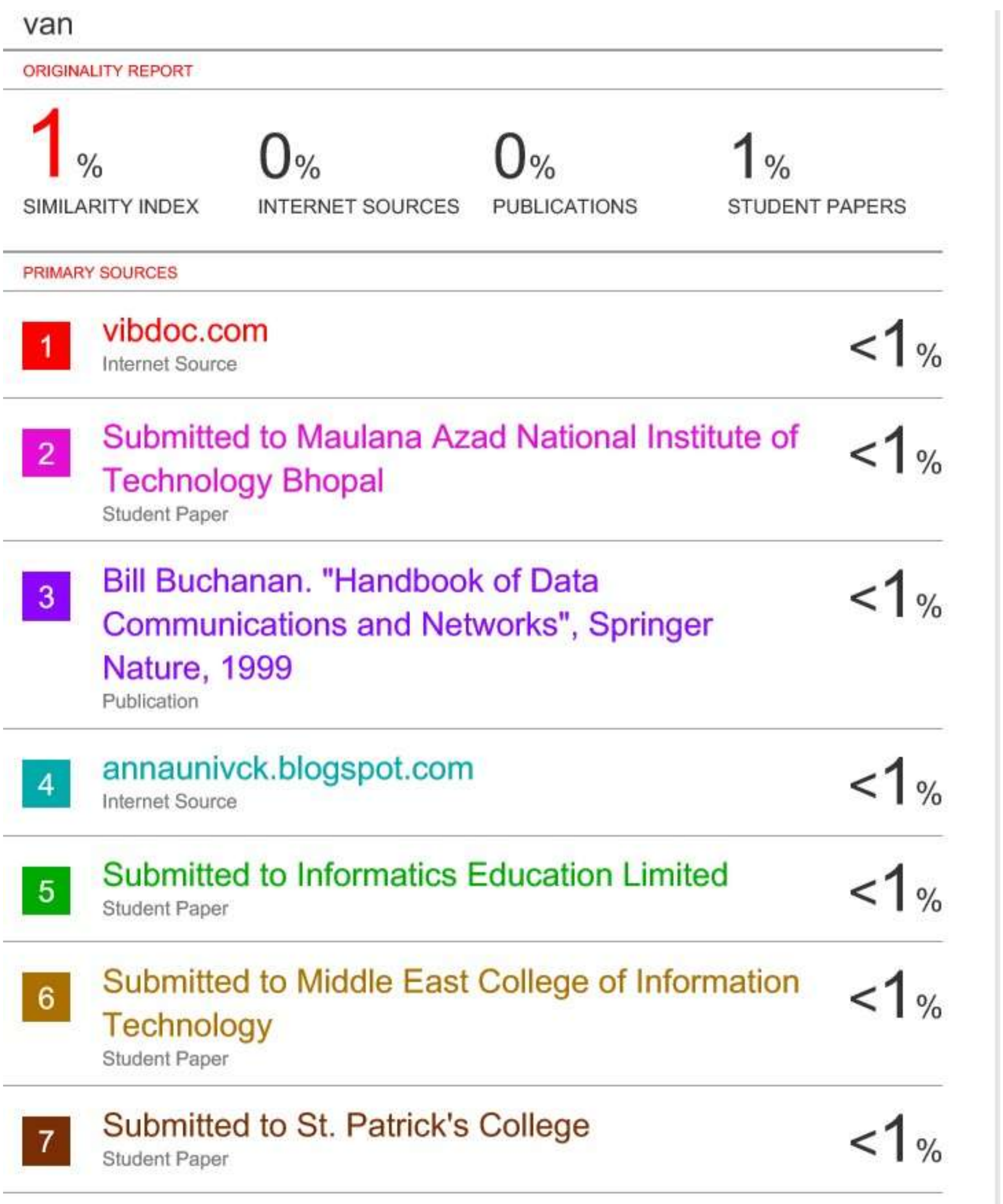
Submission date: 13-Jun-2020 07:19AM (UTC+0530)

Submission ID: 1342882211

File name: vanshika_report2020.docx (10.16M)

Word count: 11621

Character count: 57524



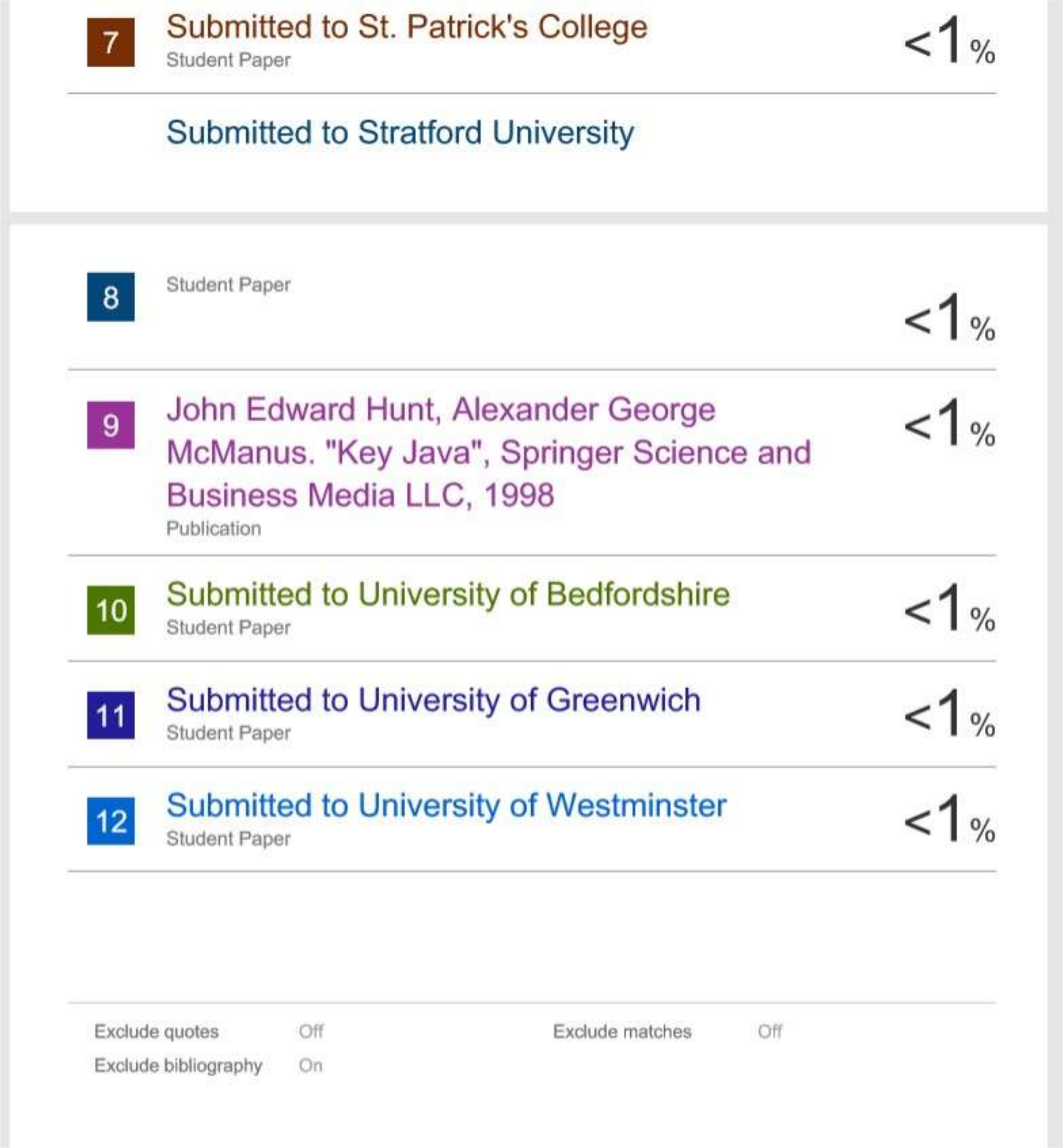
van

ORIGINALITY REPORT

1%	0%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	vibdoc.com Internet Source	<1%
2	Submitted to Maulana Azad National Institute of Technology Bhopal Student Paper	<1%
3	Bill Buchanan. "Handbook of Data Communications and Networks", Springer Nature, 1999 Publication	<1%
4	annaunivck.blogspot.com Internet Source	<1%
5	Submitted to Informatics Education Limited Student Paper	<1%
6	Submitted to Middle East College of Information Technology Student Paper	<1%
7	Submitted to St. Patrick's College Student Paper	<1%



7	Submitted to St. Patrick's College Student Paper	<1 %
	Submitted to Stratford University	
8	Student Paper	<1 %
9	John Edward Hunt, Alexander George McManus. "Key Java", Springer Science and Business Media LLC, 1998 Publication	<1 %
10	Submitted to University of Bedfordshire Student Paper	<1 %
11	Submitted to University of Greenwich Student Paper	<1 %
12	Submitted to University of Westminster Student Paper	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: …………………………

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **[✓ B.Tech Project Report]** **Paper**

Name: Vanshika Chauhan **Department:** CSE **Enrolment No** 161383

Contact No. 8629067910 **E-mail.** vanshikaa.chauhan.97@gmail.com

Name of the Supervisor: Dr. Ruchi Verma

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): Developing A 2-D Game in JAVA

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 67
- Total No. of Preliminary pages = 10
- Total No. of pages accommodate bibliography/references = 1

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 1(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	□ All Preliminary Pages □ Bibliography/Ima ges/Quotes □ 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com