

Tom & Jerry - A Java Game

A PROJECT REPORT

Submitted in partial fulfillment for the award of the degree of

B.TECH

in

Information Technology

by

Priya Gupta	14BIT0162
--------------------	------------------

Tushar Bhatia	14BIT0163
----------------------	------------------

Rohan Kumar Sachdeva	14BIT0172
-----------------------------	------------------

Under the Guidance of

Prof. THANDEESWARAN R.

School of Information Technology & Engineering (SITE)

May 2017

DECLARATION BY THE CANDIDATE

I here by declare that the project report entitled "**Tom & Jerry - A Java Game**" submitted by me to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the degree of **B.Tech.(Information Technology)** is a record of bonafide project work carried out by us under the guidance of **Prof.Thandeeswaran R.** I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: **Vellore**

Date: **3rd May 2017**

Priya Gupta

Tushar Bhatia

Rohan Kumar Sachdeva

School of Information Technology & Engineering [SITE]

CERTIFICATE

This is to certify that the project report entitled **"Tom & Jerry - A Java Game"** submitted by **Priya Gupta (14BIT0162), Tushar Bhatia (14BIT0163) and Rohan Kumar Sachdeva (14BIT0172)** to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the degree of **B.Tech.(Information Technology)** is a record of bonafide work carried out by them under my guidance. The project fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Prof. Thandeeswaran R.

SUPERVISOR

Assistant Professor (Selection Grade), SITE

The Project Report is satisfactory / unsatisfactory

Name & Signature of the Examiners

ACKNOWLEDGEMENT

We would like to express our gratitude towards **Dr. G. Viswanathan** (Chancellor), **Dr. Anand Samuel** (Vice Chancellor), **Dr. S. Narayanan** (Pro-Vice Chancellor), **Dr. Aswani Kumar Ch.** (Dean - SITE), **Dr. M. Dhinakaran** (HoD - B.Tech IT) and **Prof. P.G. Shynu** (Year Coordinator) for giving us this wonderful opportunity to work on a project that has helped us not only in doing a lot of research but has also helped us apply our skills and has expanded our skill set.

We are highly indebted to **Prof. Thandeeswaran R** (Assistant Professor, Selection Grade - SITE) for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project. Without his encouragement and guidance this project would not have materialized.

TABLE OF CONTENTS

Chapter No	Contents	Page No
	SYNOPSIS	I
	LIST OF FIGURES	II
1	Introduction	1
1.1	Background	1
1.2	Problem Statement	1
1.3	Importance	1
1.4	Organization of Report	2
2	Overview and Planning	3
2.1	Proposed System Overview	3
2.2	Challenges	3
2.3	Assumptions	3
2.4	Architecture Specifications	4
2.5	Hardware Requirements	7
2.6	Software Requirements	7
2.7	Project Schedule (Gantt Chart)	7
2.8	Work Breakdown Structure	8
3	Literature Survey and Review	9

3.1	Literature Survey	9
3.2	Literature Review	10
4	System Design	11
4.1	High-Level Design	11
4.2	Low-Level Design	12
5	System Implementation	13
5.1	Code and/or Architecture Development	13
5.2	Unit Testing	14
5.3	Integration Testing	14
5.4	Test Results	14
6	Results and Discussion	18
6.1	Output/Results	18
6.2	Results Analysis	22
6.3	Discussion	23
7	Conclusion and Future Work	24
7.1	Conclusion	26
7.2	Scope of Future Work	28
8	References	30

SYNOPSIS

The following project "TOM & JERRY - A JAVA GAME" is based on java. The game is based on a simple algorithm where an enemy, Tom chases the user, Jerry and Jerry in turn has to safeguard itself from Tom. In doing so, Jerry loses its health each time it collides with Tom and finally when the health reaches 0, the game gets over. There are also various difficulty levels in the game represented by spawning of various enemies throughout the course of the game. The game also has a pause option to stop the game for a little while and also maintains the score of the player. The player can also select the difficulty level from the menu.

This game also used various Object Oriented Concepts such as inheritance, encapsulation, polymorphism etc. The game has a very minimal number of software and hardware requirements. It requires only a Java enabled system to run properly.

Innovativeness and Usefulness:

- The game is user-friendly and does not require much technical skills for user to play.
- It involves simple java concepts and hence can be understood easily and also it has been broken into various modules that make it easy to understand different aspects of the game.

Current Status:

- The game is complete as per our initial planning.
- There is still some scope for improvement or rather addition of new parameters in the game.

LIST OF FIGURES

Figure No	Title	Page No.
Fig. 2.4.1	Health Bar	4
Fig. 2.4.2	Collision Detection	4
Fig. 2.4.3	Spawning	5
Fig. 2.4.4	Level-Up	5
Fig. 2.4.5	Basic Enemy	5
Fig. 2.4.6	Fast Enemy	5
Fig. 2.4.7	Smart Enemy	6
Fig. 2.4.8	Boss Character	6
Fig. 2.4.9	Menu System	6
Fig. 2.4.10	Game Over	6
Fig. 2.4.11	Score	7
Fig. 2.4.12	Pause	7
Fig. 2.4.13	Game Modes	7
Fig 6.1.1	Health	15
Fig. 6.1.2	Collision	15
Fig. 6.1.3	Player	16
Fig. 6.1.4	Basic Enemy	16
Fig. 6.1.5	Smart Enemy	16
Fig. 6.1.6	Boss Enemy	17
Fig. 6.1.7	Menu	17
Fig. 6.1.8	Difficulty	17
Fig. 6.1.9	Game over	18
Fig. 6.1.10	Paused game	18

Fig. 6.1.11	Shop	18
Fig. 6.1.12	Original Health	18
Fig. 6.1.13	Increased health	18
Fig. 6.2.1	Result Analysis	19

CHAPTER NUMBER 1

INTRODUCTION

1.1 Background:

This project aims to bring the fun and simplicity of Tom and Jerry game with some new features. It will include opponents as Tom and player as Jerry. The game offers two level of difficulties enumerating, normal and difficult. The aim of the game is to challenge the survival of Jerry, who has to save its health while scoring the maximum. This project explores a new dimension in the traditional Tom and Jerry chase game to make it more intriguing and challenging. Jerry has to survive for the maximum time. More time it survives, more scores are earned by the jerry. Jerry can purchase the health bar from shop, by giving its points. The game offers various levels of Tom including random moving, fast moving, intelligent player following and finally a boss. The simplicity of this game makes it an ideal choice for a mini project, as we can focus on core java paradigms.

1.2 Problem Statement:

A leisure time activity must always be fun and easy to deal with. The user must interact with a free mind with much vulnerabilities in the application. Moreover, it must be user friendly and user must be able to connect well with it. The Tom and Jerry game is designed to fulfill all such requirements. It uses simple java for implementation and the UI is user friendly that gives all instructions properly to the user for game playing. Moreover the functionalities involved require minimal skill set.

1.3 Importance:

Tom and Jerry game is a good leisure time activity that isolates the player from boredom. The main importance of this game lies in its architecture. The game has been designed using module-by-module approach that makes it easy to understand the concepts and to recreate it adding different parameters. It helps anyone who wants to understand java. The application uses the very basics of java as an object oriented programming language and can also be referenced to as java tour.

1.4 Organization of the Report:

The project report is designed in such a way that it is possible to trace any element as and when requirement. Initially, the report contains the declaration and the certificate by the project guide followed by the list of contents enumerated in the index table. Also, this list of all the tables and figures in various chapters have been provided. The contents are divided into various chapters and each chapter includes subdivisions where every aspect of the project has been elaborated deeply.

CHAPTER NUMBER 2

OVERVIEW AND PLANNING

2.1 Proposed System Overview:

Tom and Jerry Game is a simple Java Application in which the enemy chases the user until user's health is over. The game involves various levels that increments regularly and also run indifferent modes. The game is platform and user friendly since it is built in java. The WSAD keys are used to move the user up, down, left and right respectively. The game also includes a store where extra health can be bought in exchange for scores. Other functionalities include pause, help etc for better interaction. The application requires java enabled system to run and not much technical skills on part of the user.

2.2 Challenges:

The game is built using core java. Hence, one challenge could be lack of understanding if the developer is not used to java. Moreover, the challenges faced during the developement of the game were manipulating screen sizes, positioning the characters in the screen to make the application responsive, generating various levels and deciding scores based on various levels. Also, to prevent image border collisions the images had to be modified.

2.3 Assumptions:

It is well assumed that player/user is familiar with Tom and Jerry concept. Tom chases jerry all the time. Other assumptions involved are the use of basic gaming keys WSAD, pause option and knowlegde of game modes and levels.

2.4 Architecture Specifications:

The project has been divided into several modules to ease the process of development. The modules are as follows:

2.4.1 Health Bar:

The health bar indicates Jerry's health status at a given point of time. Health keeps Jerry alive in the game. Each time Jerry is hit by Tom, its health decreases by a definite amount. Moreover, when Jerry purchases extra health or wins boosters, its health increases accordingly and same is reflected in the health bar.



Figure 2.4.1

2.4.2 Collision Detection:

Collision detector detects Jerry's encounter with Tom, boosters and the side walls. Tom and Jerry can move within a definite boundary. When Tom hits Jerry, it is detected by collision detector and Jerry's health status gets reduced by a certain amount.



Fig: 2.4.2

2.4.3 Spawning:

As difficulty level increases with the score, more and more Toms spawn at random places and Jerry is supposed to save itself from all of them to stay in the game for a longer time.

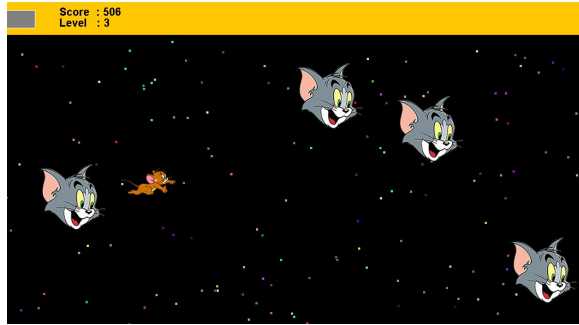


Fig: 2.4.3

2.4.4 Level-Up:

Every time the score increments by 250, the difficulty level rises. As the level increases, the number of Toms in the field increases. Finally, the boss character comes that generates more Toms at random.



Fig: 2.4.4

2.4.5 Basic Enemy:

This enemy appears as soon as the game starts. It keeps doubling in number each time the score increments by 250 till the score hits 500.



Fig: 2.4.5

2.4.6 Fast Enemy:

Fast Enemy spawns when the score hits 750 and again at a score of 1000. The speed of fast enemy increases the difficulty level.



Fig: 2.4.6

2.4.7 Smart Enemy That Follows:

It gets harder for Jerry each time the level is up. At a particular score level up spawns a smart enemy that follows Jerry's moves.



Fig: 2.4.7

2.4.8 Boss Character:

Boss character denotes the last level in the game. The boss character generates fire balls randomly to raise the difficulty level to the fullest.



Fig: 2.4.8

2.4.9 Menu System:

The menu system allows the player to change understand how to play, start a new game, exit the game etc.

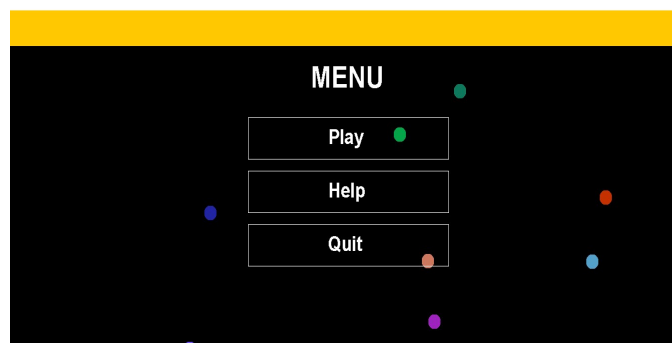


Fig: 2.4.9

2.4.10 Game Over:

This records and displays the game over score and a provides try again button.

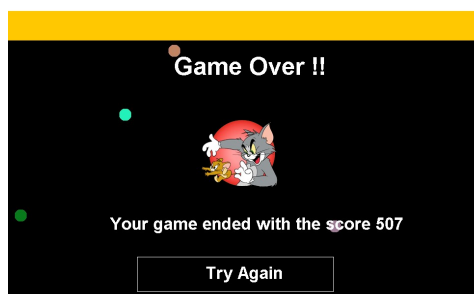


Fig: 2.4.10

2.4.11 Score:

Score keeps increasing till Jerry survives the game. Based on the score, Jerry can buy health in the middle of the game. The final score is the one where health bar hits zero.



Fig: 2.4.11

2.4.12 Pause:

unlike other basic java games Tom and Jerry game can be paused without interrupting game performance.



Fig: 2.4.12

2.4.13 Game Modes:

The game has two modes- Normal and Difficult. Each mode provides lifesaving options in their own manner and it's up to the user to use them wisely.

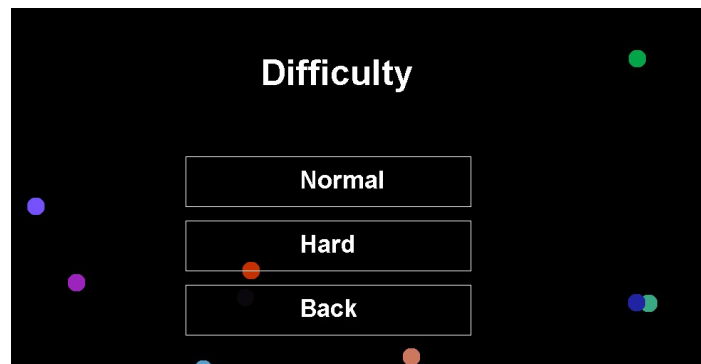


Fig: 2.4.13

2.4.14 Shop Upgrades:

Jerry has the option to increase the speed by collecting boosters. It is also possible for this character to increase its health during the course of the game. Extra health can be bought based on the score.

2.5 Hardware Requirements:

A system with jdk installed and proper configurations.

2.6 Software Requirements:

Java 8, Eclipse Neon and Photoshop

2.7 Project Schedule:

Topic Selection and approval	Design Confirmation and basic GUI implementation	Creating enemies and user along with home page and end page.	Creating shop upgrades and new levels.
5/1/17	9/1/17	14/2/17	22/3/17
			2/5/17

2.8 Work Breakdown Structure:

- The entire project is divided into various modules as discussed in section 2.4 and is dealt with module-by-module.
- Modules were divided among team members and unit testing was performed on each one of them.
- The modules were later compiled and integration testing was performed at each stage.
- Initially, the GUI design was confirmed and then the characters were chosen accordingly.
- The different characters were made to move inside the window according to the dimensions.
- The text in various screens was set according to the dimensions.

CHAPTER NUMBER 3:

LITERATURE SURVEY AND REVIEW

3.1 Literature Survey:

This proposal is organized into several subsections. The objectives of proposing a JAVA based game application is to consolidate our understanding on the core paradigm. The objective is successfully fulfilled during its development.

Information about some of the existing Tom and Jerry games have been collected from the internet. Lots of variants of traditional chase game exists but none of these game provide the experience of shopping of upgrades while playing. Also the gradual increase in difficulty in the form of various opponents.

The scope of this project has been discussed taking into account it's fun and educational quotient. The fun value if this project will help out in one's leisure time. Whereas the educational value builds a deep insight of core JAVA concepts while building an intriguing game.

What's already there?

1. Tom and Jerry: Mouse Maze - A maze running game developed in java. The basic functionality of the game is that Jerry wants cheese! But Jerry must be careful as Tom is hunting him. Help Jerry pick up all cheese avoiding Tom in this funny cat and mouse game. The game uses bonuses and things from everyday life, react the attacks fast, collect all food on the level into the box and uses prizes, which will add power to the mouse or make his moves faster.
2. Java Snake Game - Snake is an older classic video game. It was first created in late 70s. Later it was brought to PCs. In this game the player controls a snake. The objective is to eat as many apples as possible. Each time the snake eats an apple, its body grows. The snake must avoid the walls and its own body. This game is sometimes called Nibbles. The size of each of the joints of a snake is 10px. The snake

is controlled with the cursor keys. Initially, the snake has three joints. If the game is finished, the "Game Over" message is displayed in the middle of the board.

3. Puzzle Game in Java - The user is given a 3*3 size board of numbers from 1 to 8 arranged randomly with a blank space in box. The user can move the blank space left, right, top and bottom and arrange the numbers in the order.

3.2 Literature Review:

Initially, the game starts with only one attacking random Tom in the game. As the score increments by 250, another random Tom spawns. The game now contains two Toms till the score reaches 500. As soon as the score hits 500, two more basic enemies appear on the screen. Further increments by 250 includes fast moving Toms at scores 750 and 1000 respectively. If the player scores 1300 points, another enemy enters. This is an intelligent enemy that follows Jerry all the time. When score hits 1700, all the existing enemies vanish and Boss Enemy enters that throws fire balls randomly and Jerry is supposed to safeguard itself from the fire balls.

Spawning System: A spawning manager is developed to offer different level of complexities as the level advances. It makes the game more challenging and intuitive to play.

Smart Enemy: A special enemy bot which adjusts it's path according to player's position.

Game Pause: unlike other basic java games Tom and Jerry game can be paused without interrupting game performance.

Shop Upgrade System: This is the bonus inventory created for the player to increase player's chance of survival.

What's for future?

A two player game can be extended from this with more than two difficulty levels.

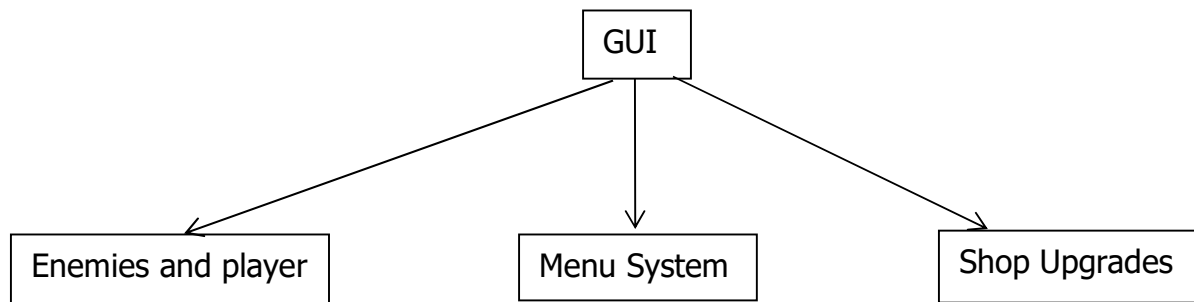
Also can be made to run on LAN.

CHAPTER NUMBER 4

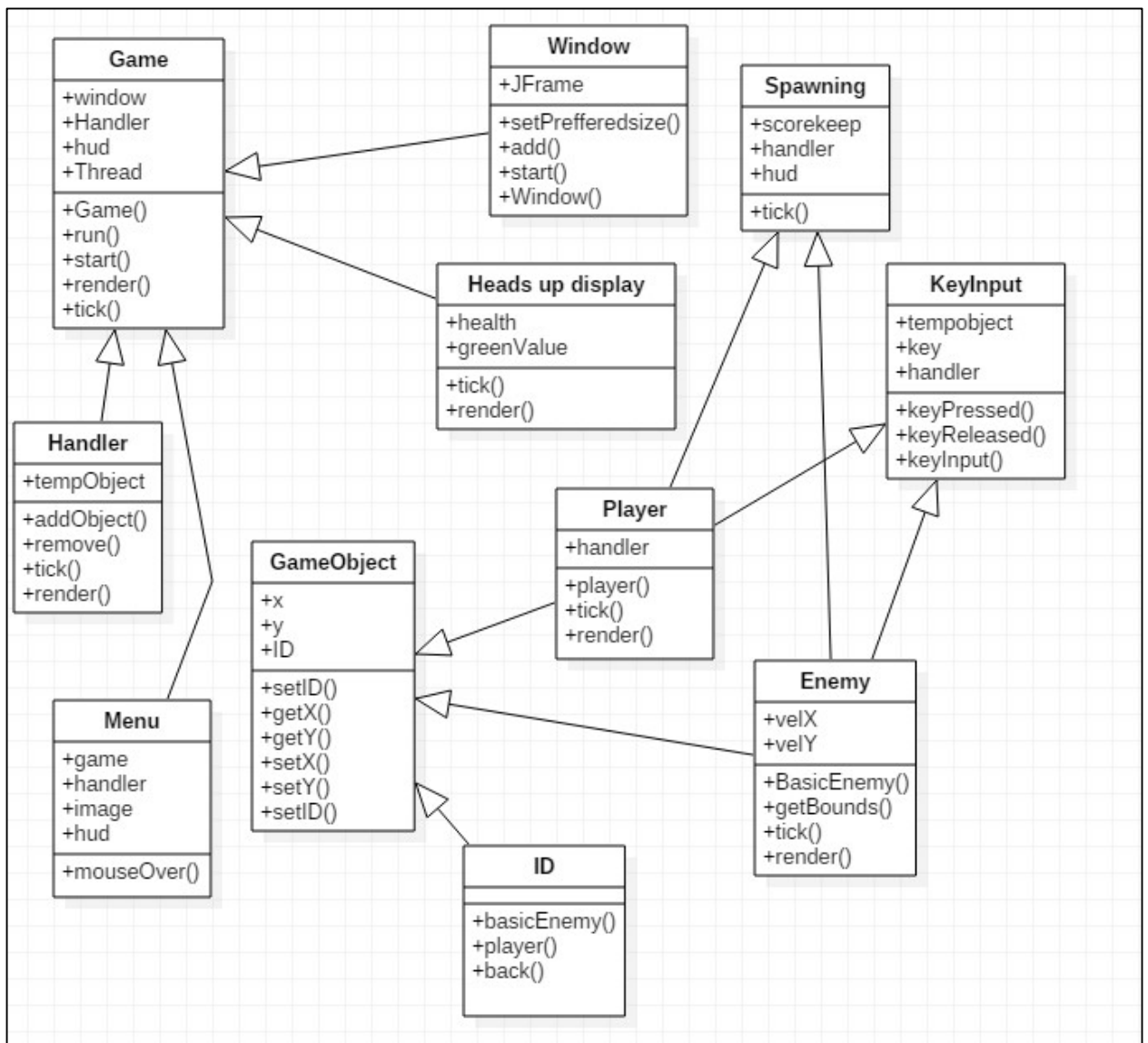
SYSTEM DESIGN

4.1 High Level Design:

High-level design (HLD) explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.



4.2 Low Level Design:



CHAPTER NUMBER 5

System Implementation

5.1 Code and/or Architecture Development:

This game has lots of classes and codes in java. For this module we would like to present our driver class. Driver class is the one which gives the fuel to the project to

```
public Game(){
    handler=new Handler();
    hud=new HUD();
    menu=new Menu(this,handler,hud);

    this.addMouseListener(menu);
    this.addKeyListener(new KeyInput(handler,this));
    new Window(WIDTH, HEIGHT, "TOM AND JERRY", this);// this will call the window constructor which starts the run method
    Random r=new Random();
    spawner=new Spawn(handler,hud,this);

    if(gameState== STATE.Game)
    {
        for(int i=0;i<20;i++)handler.addObject(new Background(r.nextInt(1350),66+r.nextInt(1350/12*6),ID.back,handler));
        handler.addObject(new Player(WIDTH/2-32,HEIGHT/2-32,ID.player,handler));// here we are adding game object player
        handler.addObject(new BasicEnemy(65,65,ID.BasicEnemy,handler));
    }else
    {
        for(int i=0;i<10;i++)handler.addObject(new MenuParticle(r.nextInt(1150)+50,66+r.nextInt(500)+50,ID.Particle,handl
    }
}
```

run. Our driver class is Game.java

```
import java.awt.Canvas;
public class Game extends Canvas implements Runnable{ // Extends in to inherit any other class
    // Here Canvas is inherited by Game

    private static final long serialVersionUID = 7141660673071146540L;

    public static final int WIDTH = 1350, HEIGHT = WIDTH/12*6;

    private Thread thread;
    private boolean running = false;
    private Handler handler;
    private HUD hud;
    private Spawn spawner;
    private Menu menu;
    public static boolean paused=false;
    public int diff=0;

    public enum STATE
    {
        Menu,Game,Help,End,Select
    };
    public static STATE gameState=STATE.Menu;
}
```

5.2 Unit Testing:

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. We were doing the Unit Testing in every module of our project, this helps us in getting the great work easily and efficiently.

5.3 Integration Testing:

Integration Testing is a level of software testing where individual units are combined and tested as a group. During the process of manufacturing a ballpoint pen, the cap, the body, the tail and clip, the ink cartridge and the ballpoint are produced separately and unit tested separately. When two or more units are ready, they are assembled and Integration Testing is performed. For example, whether the cap fits into the body or not .We were doing the Intergration Testing in every module of our project, this helps us in getting the great work easily and efficiently.

5.4 Test Results:

Integration testing will help to verify the overall system after developing the different modules. When issues observed in the integration test, the unit testing results for specific functionality will help to fix the issue. As a result both Unit testing and Integration testing is equally important. From our tests we underwent many testing phase. Which overall made our project more reliable, efficient and productive.

CHAPTER NUMBER 6

Results and Discussion

6.1 Output/Results:

This game has lots of classes and codes in java. For this module we would like to present our driver class. Driver class is the one which gives the fuel to the project to run. Our driver class is Game.java. These are the following outputs of the Game in the form of images.

- Health Bar



Fig: 6.1.1

- Collision



Fig: 6.1.2

- Player



Fig: 6.1.3

- Basic Enemy

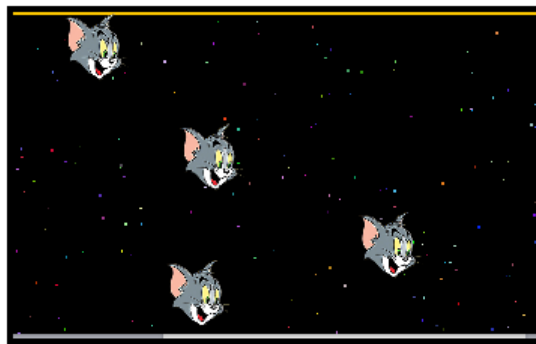


Fig: 6.1.4

- Smart Enemy



Fig: 6.1.5

- Boss Enemy

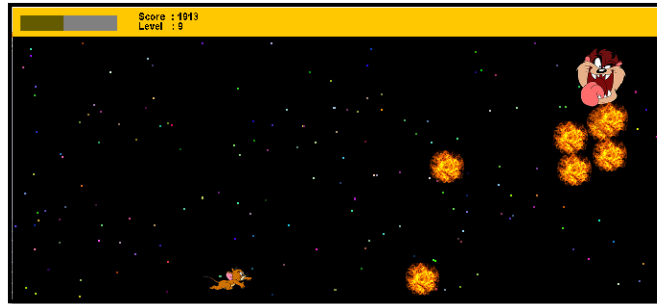


Fig: 6.1.6

- Menu

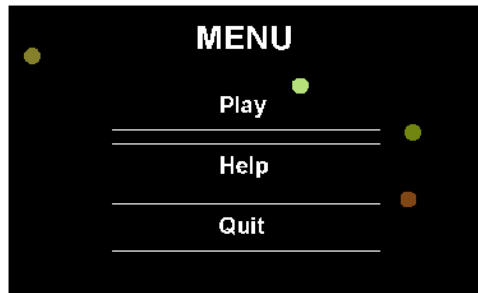


Fig: 6.1.7

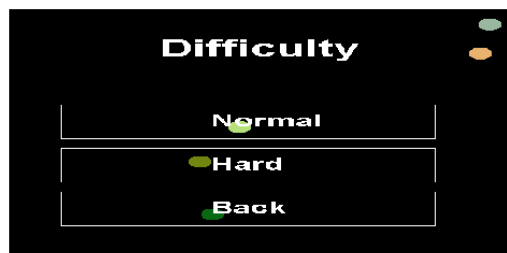


Fig: 6.1.8

- Game Over

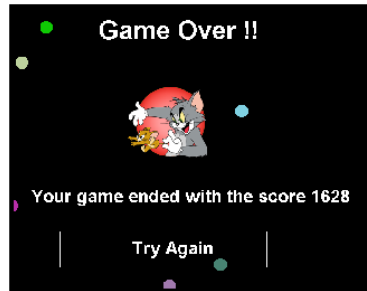


Fig: 6.1.9

Paused game



Fig: 6.1.10

The Shop

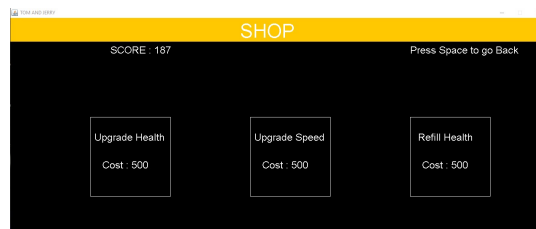


Fig: 6.1.11

Increased health



[ORIGINAL HEALTH BAR]

Fig: 6.1.12



[HEALTH BAR AFTER PURCHASE]

Fig: 6.1.13

6.2 Results Analysis:

Our game had many classes. Each class doing a specific and the important action. The game we came up is really challenging and interesting.

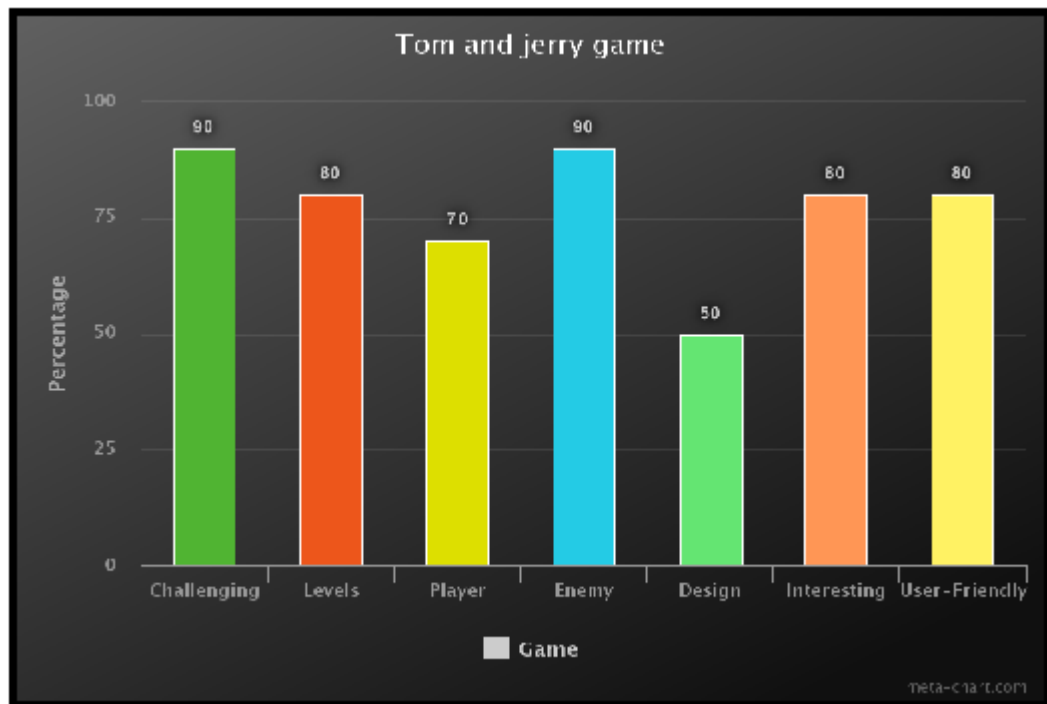


Fig: 6.2.1

6.3 Discussion:

A leisure time activity must always be fun and easy to deal with. The user must interact with a free mind with much vulnerabilities in the application. Moreover, it must be user friendly and user must be able to connect well with it. The Tom and Jerry game is designed to fulfill all such requirements. It uses simple java for implementation and the UI is user friendly that gives all instructions properly to the user for game playing. Moreover the functionalities involved require minimal skill set. Tom and Jerry game is a good leisure time activity that isolates the player from boredom. The main importance of this game lies in its architecture. The game has been designed using module-by-module approach that makes it easy to understand

CHAPTER NUMBER 7

Conclusion and Future Work

7.1 Conclusion:

It was a great experience to make this game. Being a coder it's our responsibility to make the non-existing user friendly game which are especially a great use to people in their leisure time. This helped us to get the better understanding of java libraries and its functions. It also helped us in getting better understanding of basic programming concepts of java language such as loops, control structure, classes, interface, abstract classes etc. After doing this project, we are in position to explain libraries, functions and little oops concepts and apply them to the modelling of real world systems by utilizing its offered facilities.

7.2 Scope of Future Work:

"It is never a mistake to say good-bye."

Games will continue to mostly fall into familiar categories, but since tools for developing games have become readily available and free for many, the massive number of game developers means we'll see plenty of experimentation. Mobile games have become the largest segment in terms of annual revenue, and that won't change. VR and AR games will be an interesting area to watch, but it's going to be years before either becomes a market big enough to challenge other game platforms.

Games will become a more accepted part of popular culture, and we will see the biggest games generating as much or more revenue than the most popular books, movies, or music franchises.

For the gamer, you'll see more games to choose from on every platform, and games will increasingly look to turn you into a permanent fan of the game, in much the same way that people become fans of popular bands or movie franchises and always look for the latest material from their favorites.

REFERENCES

S. M. Dorman, "Video and computer games: effect on children and implications for health education," *Journal of School Health*, vol. 67, no. 4, pp. 133–138, 1997. View at Google Scholar · View at Scopus

M. Prensky, "Digital game-based learning," *Computers in Entertainment*, vol. 1, no. 1, pp. 21–24, 2003. View at Google Scholar

J. Blow, "Game development: harder than you think," *Queue*, vol. 1, no. 10, pp. 28–37, 2004. View at Google Scholar

A. I. Wang and B. Wu, "An application of a game development framework in higher education," *International Journal of Computer Games Technology*, vol. 2009, Article ID 693267, 12 pages, 2009. View at Publisher · View at Google Scholar · View at Scopus

Journal Articles

Baloglu, S. and Brinberg, D. (1997) 'Affective images retrieval', *Journal of IMAGE Processing*, Vol. 35, No. 4, pp.11–15.

Books

Black Art of Java Game Programming, by Joel Fan, Eric Ries, and Calin Tenitchi (Waite Group Press)

Cutting-Edge Java Game Programming, by Neil Bartlett, Steve Simkin, and Chris Stranc (Coriolis Group Books)

Developing Java Entertainment Applets, by John Withers (Wiley)

Website link

- www.softwaretestingclass.com
- www.quora.com
- codeutopia.net
- softwaretestingfundamentals.com
- archive.sap.com