****

**Project Report**

**ON**

**Two Players Snake Game**

SUBMITTED TO

**CHANDIGARH GROUP OF COLLEGES**

In Partial Fulfilment of the Requirements for the Award of Degree of

**BACHELOR OF COMPUTER APPLICATION**

**SUBMITTED BY SUPERVISED BY**

BHUNESHWAR PRASAD Mr.Krishna Tuli

(Prof)

**CHANDIGARH GROUP OF COLLEGES**

**LANDRAN(MOHALI)**

**PUNJAB**

**Table of Content:**

Certificate Page........................................................................

Acknowledgement Page...........................................................

Abstract....................................................................................

**System Planning.....................................................................**

* Project Aim and Objective
* Project Description
* Background of the project
* Defining Problem

**Requirement Analysis.............................................................**

* Software Requirement Specification

**System Design...........................................................................**

* DFD

**Programming............................................................................**

* Coding

**System Testing...........................................................................**

* Movement Testing
* Collision Testing
* Level and Score Testing
* Multiplayer Mode Testing

**Implementation............................................................................**

* **Screen Shots**

**Evolution......................................................................................**

* Future Scopes

**Conclusion....................................................................................**

**Reference......................................................................................**

**CERTIFICATE**

This is to certify that Bhuneshwar Prasad has completed his project “Multiplayer Snake Game”

Under my guidance and supervision. To the best of my knowledge, the present work is the result of his original study, hard work, and commitment. No part of the report has ever been submitted for any other degree or diploma. The report is fit for the submission and the partial fulfilment of the conditions for the award of Bachelor of Computer Application.

**[MEMBER] [SUPERVISOR NAME]**

Name : Bhuneshwar Prasad Mr. Krishna Tuli

Roll : 1615765

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Date : \_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

First of all I would like to thank our Mentor/Supervisor Ms. Davinder Kaur for the effort, support and the guidance. She tirelessly helped and supported me. So I deeply extend my sincere appreciattion, gratitude and prayers to her.

I also want to thank our all faculty members for their support and cooperation Especially Ms. Inderpreet Kaur

Finally, I would like to thank my friends for giving me ideas and their support in making this project. Although my game was not a fantasy type but still they put their effort in seeing it. This encouraged me to make it even better and better..

Thank You!!

**ABSTRACT**

This project aims to bring the fun and simplicity of snake game with some new features. It will include unique objects, player’s head, skin changing, and enemies to beat the player.

It will also have the multiplayer feature that will allow another player to play using the same keyboard.

This project explores a new dimension in the traditional snake game to make it more interesting and challenging. The Simplicity of this game makes it an ideal candidate for a minor project as we can focus on advanced topics like multiplayer functionality.

**PROJECT AIM AND OBJECTIVE**

The objective of this snake game is to make the player crash.

The project is aimed at developing a multiplayer snake game better than the traditional snake game.

Traditional Snake Game is very simple game. In this, the snake is eating some food and get its larger as it eats..

However in my game project, i have added lots of unique feature to this traditional game.

The game description is given in later pages..

**PROJECT DESCRIPTION**

The “Snake Game” is a very old game. This version is the extension of an old snake game with the additional feature. This snake game is made in java language which makes it platform independent. This game is very easy to play.

The snake game is one of the simplest game concept ever. Your goal is to move the snake and eat as many food as possible. There are two food at nay given time. Apple gives 1 score and banana gives 2.

The game starts by pressing any arrow key.

Game Cheat has also been included in this snake game. Player can skip any level by just pressing ENTER key.

**Features :**

* A very useful friendly game
* Simple and easy game to understand
* Presentation of level, score and length
* There are two types of food : apple(+1 score) and banana(+2 score)
* There are about 12 levels. 5 score is required for each level
* Skin of the snake changes at every level.
* There are transparent objects which make sure that solid objects are going to appear in the next level.. so be prepared for that.
* At some stages an enemy also appears which moves opposite to your key pressed for movement. This snake keeps increasing in length as time passes... so you will have to be quick in leveling up.
* At some stages two enemy appears. One of them move opposite of your movement and another one moves randomly. These snakes increases in length rapidly.
* There are about 11 skins for the snake. 4 out of them are misc type.. it means that the snake shows my friend’s letter in it after each direction change.
  + GURI
  + HIPO
  + PUBG
  + ANKU
  + MOTA (Not available yet)
* There are three types of objects with 5 types of colors.
  + NIL : This one is completely solid object.
  + UP : This object contains a pipe which allows snake to pass through (in the 2nd top)
  + DOWN : This object contains a pipe which allows snake to pass through (in the 2nd last)
* On the last stage i.e level 12 : Another player appears which makes this game multiplayer at the last stage.
  + Single Player Keys : up, down, left, right arrow keys.
  + Multiplayer Keys : w, a, s, d

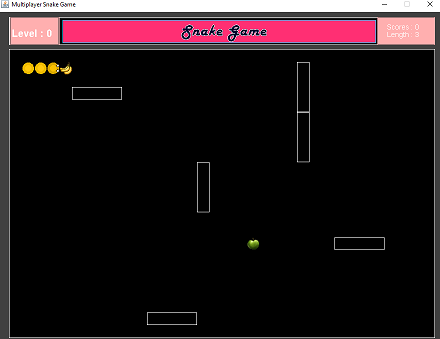
**Background of Multiplayer Snake Game**

This is the extended version of the most popular computer game called “Snake”. The main objective is to feed an increasing length of a snake with food particles which are found at random positions, picking up bonus bananas that also occur along with the regular food i.e apple.

The game starts with a paused screen where the snake appears at the default location of the whole frame. Some transparent objects also occur in the frame. These objects alert the user that the object is going to be appeared in the next stage.

Level : 0

+1 every 5 score

****

Score : 0

Apple : +1

Banana : +2

Length : 3

+1 every food

Snake :

Default position of the snake

Food :

Each food when consumed is added in score.

Object :

Transparent object which will become solid in next stage

**DEFINING PROBLEM**

To create a snake game that allows users to control the movement of a snake on a screen, to get points for eating food and avoiding running into the walls or the growing tail of the snake itself.

In this problem, we want to write a game where a graphical representation of a snake moves across the screen. When it encounters a piece of food, the snake grows longer and we gain a point. If it hits the runs into itself or hit the object, the snake dies. The snake is controlled by the keyboard keys.

To write this program, we are going to need :

* A way representing the snake
* A way of representing the food
* A way of representing the object
* A way to display the score, level and length
* A way which allow snake not to hit the border
* A way for our instructions to reach the snake
* And A way to know when we have run into something and died

I am going to use an object oriented approach and provide some detail here.

We have to think about the classes that we want to build with the associated variables and functions that will make sense for the development.

Lets start by looking at the snake itself, the hero of the game. The snake has a location on the screen, and contains multiple visual elements, as it can grow, and the snsake’s head is connected to the rest of the snake and snake’s body follows it around the screen. If the snake “Eats” food , It grows. The snake also keeps track of which way its going, using the array snakexlength[0] and snakeylength[0] for the head and the rest are for the tail.

Here are the requirements (functional requirements) for how the snake moves.

* The snake must appear to move around the screen
* The snake must turn in response to user input
* The snake will increase in length if it eats food
* The snake will die if it runs over itself
* The snake will die if it hits the object
* The snake will die if it hits the enemy
* The snake will never stop moving
* The enemy must appear at certain levels.
* The player2 must turn in response to user input
* The enemy must increase in length if it eats food
* The enemy will never stop moving

To keep the game interesting

* Several levels must be made with different objects at different levels
* Enemies appear at certain level
* The code must be object oriented(Curriculum requirement)
* The game should be playable with a really, really long snake
* The snake should pass from the objects at certain points.
* The skin of the snake should be changed at each level
* The head of the snake should change at certain level to make it more interesting.
* Multiplayer at the last stage should also be accomplished to make the game very interesting.

The snake moves in a very precise way. Based on what the user presses, the snake will move in a given direction. Every time the snake moves, the head will go in the new direction and every piece of the snake will move up, by occupying the space that was formerly occupied by the piece in front of it.

To grow in size, the snake has to eat food. How can we show the snake eating? The simplest answer is that if the head of the snake and the food are in the same place, we consider that the snake eats the food. This means that we have to know where the food is. When it’s eaten, it disappears the snake grows, and the food shsows up somewhere else.

To make the game interesting, we probably want the food to be at the random location, which means thaat we will have to make sure that our program can generate random numbers. That is the random function.

The following things we have to do :

* Draw the playing area with bounding frame
* Draw the snake in a starting position
* Draw the food in a starting location
* On user input, Change the snake direction
* Increament the move variable.
* If the snake goes over the food, increase the score, length and move the food somewhere else
* If the snake gets over itself or object or enemy, the snake die.
* Go back to 4th point
* Until the snake dies

The game will be coded in java using the ECLIPSE Software.

**SYSTEM REQUIREMENT ANALYSIS**

* **Hardware Interface:**
* 1)20 GB hard disk
* 2)256 MB RAM
* 3)Peripheral devices

**Software Interface:**

The software is developed with all the basic controls and class provided in JAVA. Windows XP or above installed on the system. Application Package must be installed. It work easily in any OS.

**Communication Interface:**

JAVA is platform independent language.

**Tools information:**

To make a system interactive with the user the use of front-end tools comes into the picture. The front-end tools make the user interface with the system easier and also provide a user-friendly environment to the system.

This rich language enables you to develop many different types of applications. You can show and can create programs that intersect with the hardware.

**SOFTWARE REQUIREMENT**

**ECLIPSE PHOTON:**

**Eclipse** is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming), and is the most widely used Java IDE. It contains a base [workspace](https://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](https://en.wikipedia.org/wiki/Plug-in_(computing)) system for customizing the environment. Eclipse is written mostly in [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) and its primary use is for developing Java applications, but it may also be used to develop applications in other [programming languages](https://en.wikipedia.org/wiki/Programming_language) via plug-ins, including [Ada](https://en.wikipedia.org/wiki/Ada_(programming_language)), [ABAP](https://en.wikipedia.org/wiki/ABAP), [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [Clojure](https://en.wikipedia.org/wiki/Clojure), [COBOL](https://en.wikipedia.org/wiki/COBOL), [D](https://en.wikipedia.org/wiki/D_(programming_language)), [Erlang](https://en.wikipedia.org/wiki/Erlang_(programming_language)), [Fortran](https://en.wikipedia.org/wiki/Fortran), [Groovy](https://en.wikipedia.org/wiki/Groovy_(programming_language)), [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), [Julia](https://en.wikipedia.org/wiki/Julia_(programming_language)), [Lasso](https://en.wikipedia.org/wiki/Lasso_(programming_language)), [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)), [NATURAL](https://en.wikipedia.org/wiki/Software_AG), [Perl](https://en.wikipedia.org/wiki/Perl), [PHP](https://en.wikipedia.org/wiki/PHP), [Prolog](https://en.wikipedia.org/wiki/Prolog), [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [R](https://en.wikipedia.org/wiki/R_(programming_language)), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)) (including [Ruby on Rails](https://en.wikipedia.org/wiki/Ruby_on_Rails) framework), [Rust](https://en.wikipedia.org/wiki/Rust_(programming_language)), [Scala](https://en.wikipedia.org/wiki/Scala_(programming_language)), and [Scheme](https://en.wikipedia.org/wiki/Scheme_(programming_language)). It can also be used to develop documents with [LaTeX](https://en.wikipedia.org/wiki/LaTeX) (via a TeXlipse plug-in) and packages for the software [Mathematica](https://en.wikipedia.org/wiki/Mathematica). Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial [codebase](https://en.wikipedia.org/wiki/Codebase) originated from [IBM VisualAge](https://en.wikipedia.org/wiki/IBM_VisualAge). The Eclipse [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since the introduction of the [OSGi](https://en.wikipedia.org/wiki/OSGi) implementation ([Equinox](https://en.wikipedia.org/wiki/Equinox_(OSGi))) in version 3 of Eclipse, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundles

Eclipse [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) (SDK) is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software), released under the terms of the [Eclipse Public License](https://en.wikipedia.org/wiki/Eclipse_Public_License), although it is incompatible with the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License). It was one of the first IDEs to run under [GNU Classpath](https://en.wikipedia.org/wiki/GNU_Classpath) and it runs without problems under [IcedTea](https://en.wikipedia.org/wiki/IcedTea).

**JAVA PLATFORM:**

**1 – JAVA SE RUNTIME ENVIRONMENT (JRE) :** The JRE provides the libraries, Java Virtual Machine and other components necessary for you to run applications written in the java programming language.

**2 – JAVA SE DEVELOPMENT KIT :** The JDK includes the JRE plus command line development tools such as compilers and debuggers that are necessary or useful for developing applets and applications.

JDK consists of JAVA API, Java Compiler and JVM

**The Java Application Programming Interface (API) :** The API is a large collection of ready made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces. These libraries are known as packages.

**The JVM is a Virtual Machine**, that is a software machine that simulates what a real machine does.

Like real machines, it has an instruction set(the bytecodes), a virtual computer architecture and an execution model.

It is capable of running code written with this virtual instruction set, pretty muvh like a real machine can run machine code.

**JVM**

**JAVA API**

**JRE**

**JAVA COMPILER**

**JDK**

**HARDWARE BASED PLATFORM**

**JAVA CODE COMPILATION AND EXECUTION**

Runtime

JIT CODE

Generator

Interpreter

Byte Code Verifier

Class Loader

**JVM**

Machine Code or Byte Code

(HelloWorld.class)

Java HelloWorld

Compiler (Javac) :

Javac HelloWorld.java

Source File : (HelloWorld.java)

**OS and Hardware**

**SYSTEM DESIGN**

**DFD**

UP KEY

DOWN KEY

LEFT KEY

RIGHT KEY

Start the game

Press any key to start the game

START

Background, Snake, Object & Food

Check Snake Direction

GOES UP

GOES DOWN

GOES LEFT

GOES RIGHT

Collision with Food

Next Food Appear

Collision with Enemy

Collision with Object

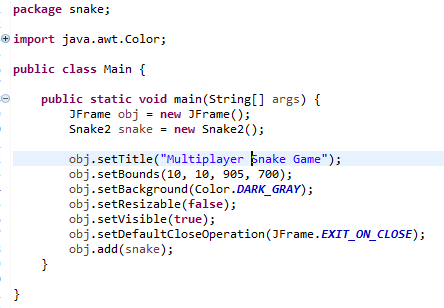
Collision with Itself

Game Over

**CODING EXPLANATION**

The game is divided into five classes

1. **Main Class**

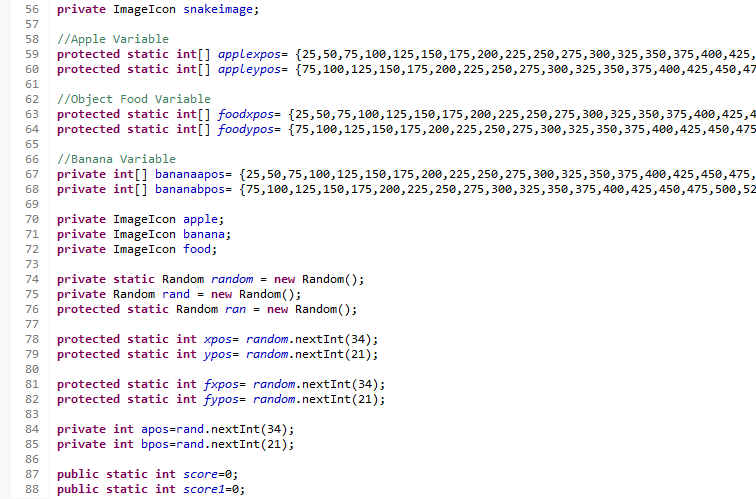


1. **Snake2 Class**

This is the major class. The followings are the important functions which it contains :

* All variables required for the game
* Snake Variables, Objects Variables, Enemy Variables, Food Variables
* Paint Component (Graphical Part)
* All the stages
* All Objects location at different positions in different classes
* All object skins
* All snake skins
* Key Pressed Key Event and Action Listener

**Sample of this class :**

Variable Sample :

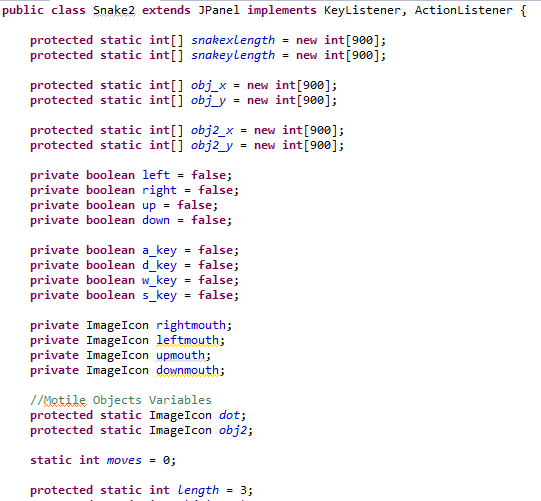
Snake Variables

Object Variables

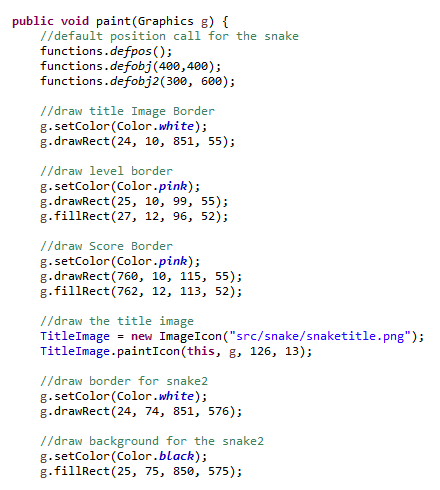
Enemy Variables

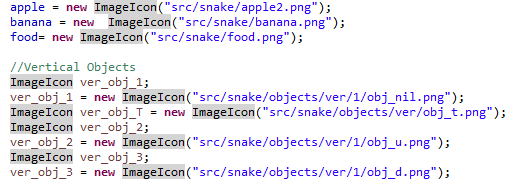
Movement Variables

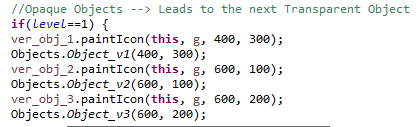
Graphics Variables etc...



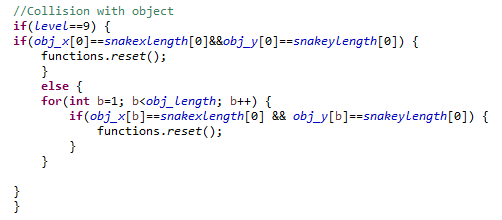
Paint Component Sample :



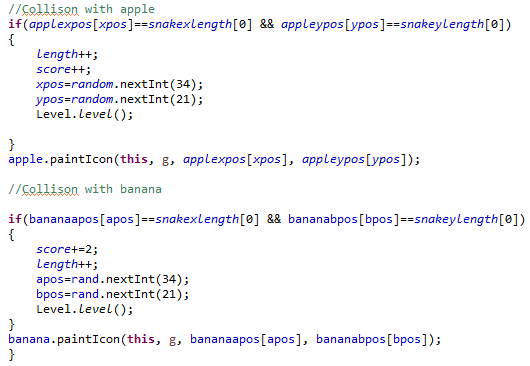
****

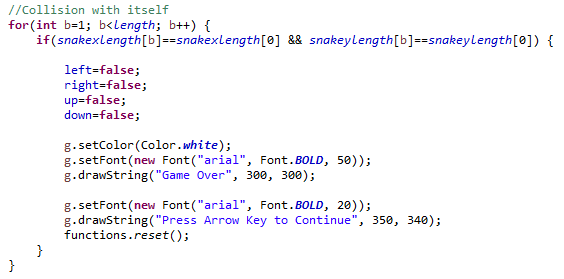
****

Collision with object

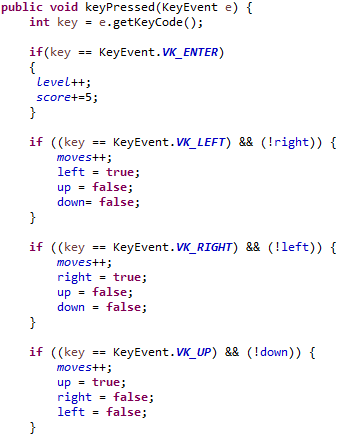


Collision with the food



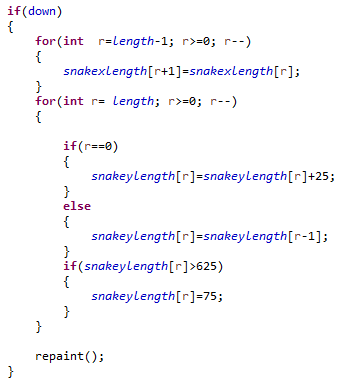
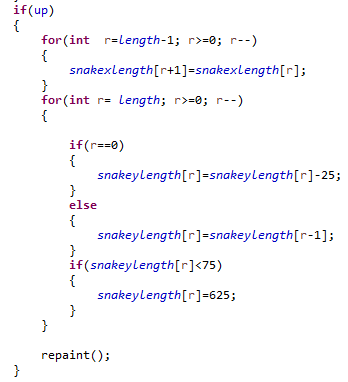
Collision with itself

Key Events



Movement of the Snake

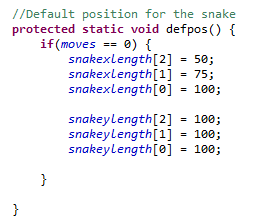


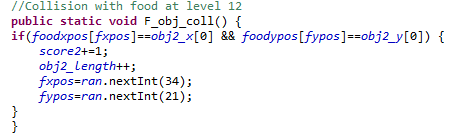


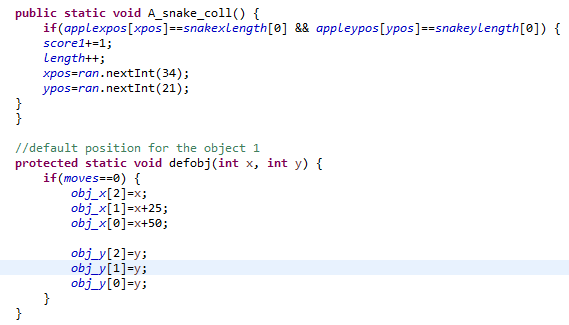
1. **Functions Class**

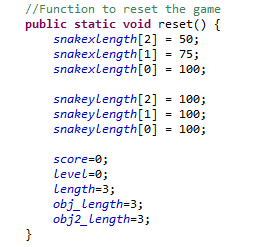
This class contains the following functions :

* Default Position of snake
* What happens after collision of the snake with the food
* What happens after collision of the snake with the enemy
* Default position of the object
* Game Reset Function



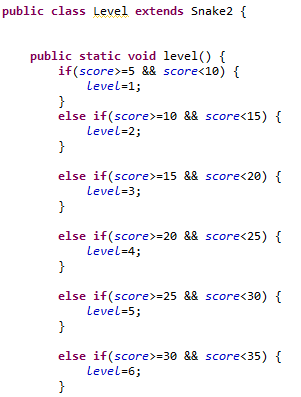






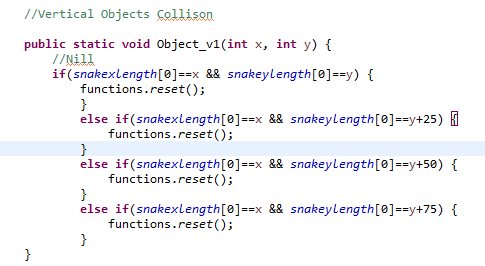
1. **Level Class**

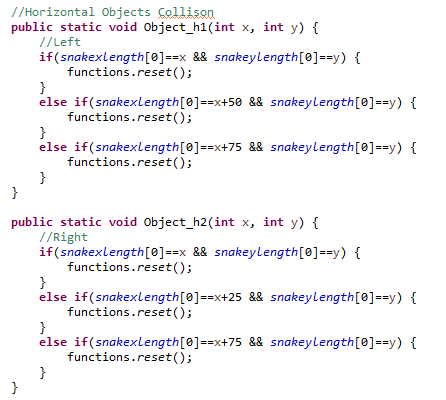
This class increases level based on the score



1. **Objects Class**

This class is for the collision of the snake with the different types of objects





**TESTING**

All testing was done manually in small sections as those sections of the code were finalized in order to ensure that the project worked as was and any bugs could be identified easily and solved quickly.

For quick testing everything, i have added a cheat in the game.. Press ENTER to level up

This allowed me to test each of the section very quickly.. i didnt need to score to level up.

1. **Movement Testing**

Movement was the first thing that was tested. This testing was broken up into two parts first the logic of the movement and second , movement with the keyboard.

To test the movement of the snake, keys were assigned for up, down, right and left movement of the snake. Using the action listener when key pressed, i was able to test the movement of the snake.

I made sure that the input and direction of the snake corresponded and that the user could not move illegally.

1. **Collision Testing**

This was tested colliding in every possible way. This also tested the fruit and made sure that it was possible to lose the game. This also tested the overall game logic to ensure that the second player or first player could continue to play the game in two player mode at level 12 even after collision.

Collision with the object, enemy, another player and the food was tested one by one.

Moreover collision was tested on each level.

1. **Level and Score Testing**

It was made sure that every time the food is consumed by the user, the score will be increased depending on the type of the food consumed.

Apple gives +1 score while the banna gives the score +2.

On every +5 score , a level is increased..

1. **Multiplayer Mode Testing**

At level 12, multiplayer mode was tested. Whether the food for each player is appearing or not.

And the score of each player was tested. Whether the consumed food is giving them scores or not.

**IMPLEMENTATION**

It was implemented using Java Swing on ECLIPSE Software.

Some of the sample codes have been shown already in the previous coding section.

We will see some screenshots in next section too..

**EVOLUTION**

This section is covered in the future scope.

As time passes, users demand the new things. So new things can be easily implemented in this snake game because of several modules.

Different types of new features can be added easily.

**SNAPSHOT**

Level 0 : Transparent Objects... Snake can pass through them.

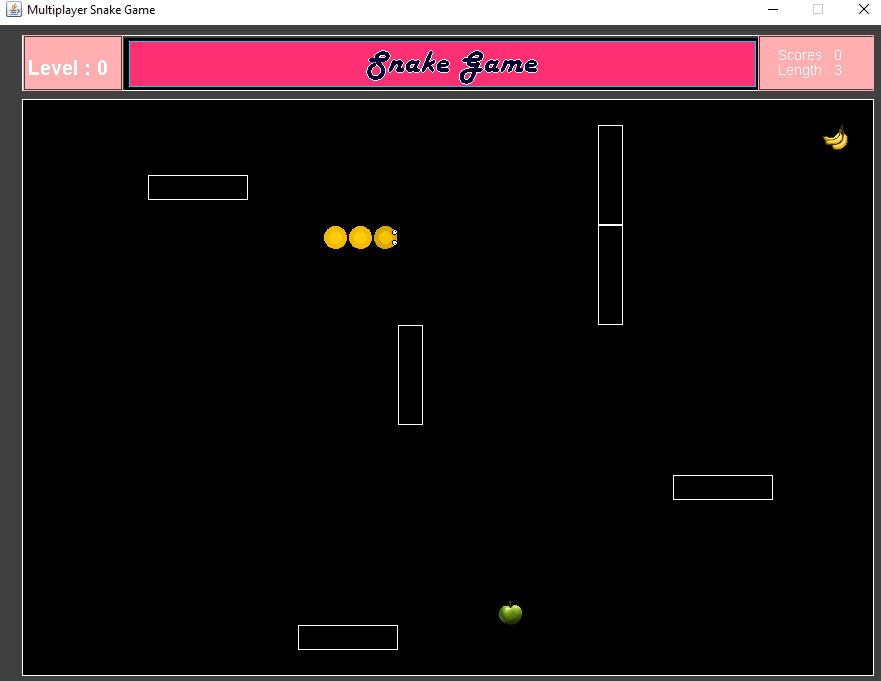
INIT SNAKE with the following things :

Level – 0

Score – 0

Length – 3

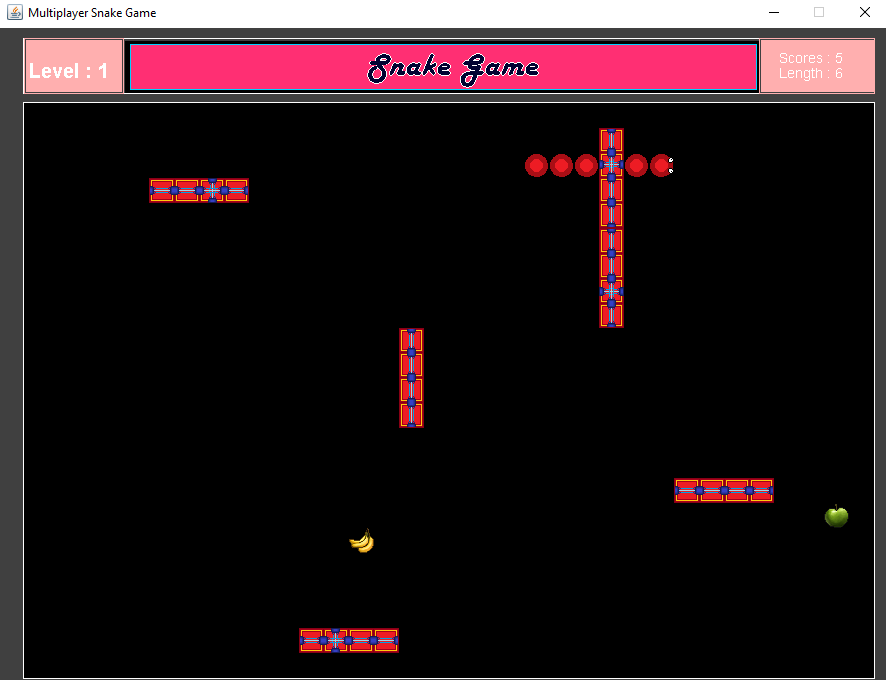
Skin - Yellow



Level 1 : On this level, The real solid objects appear.. If the snake head touches these objects, the snake will die and game will be over.

However, the snake can pass through pipes of the objects

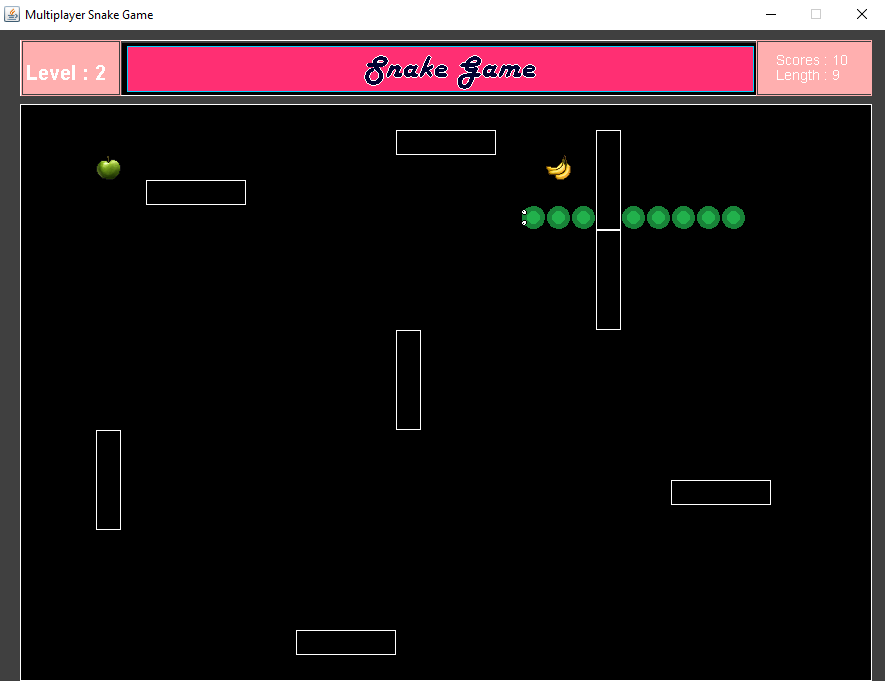
It is shown below.



Level 2 : Transparent Objects at different locations again appear to alert us for the next level.

The snake can pass through these transparent objects.

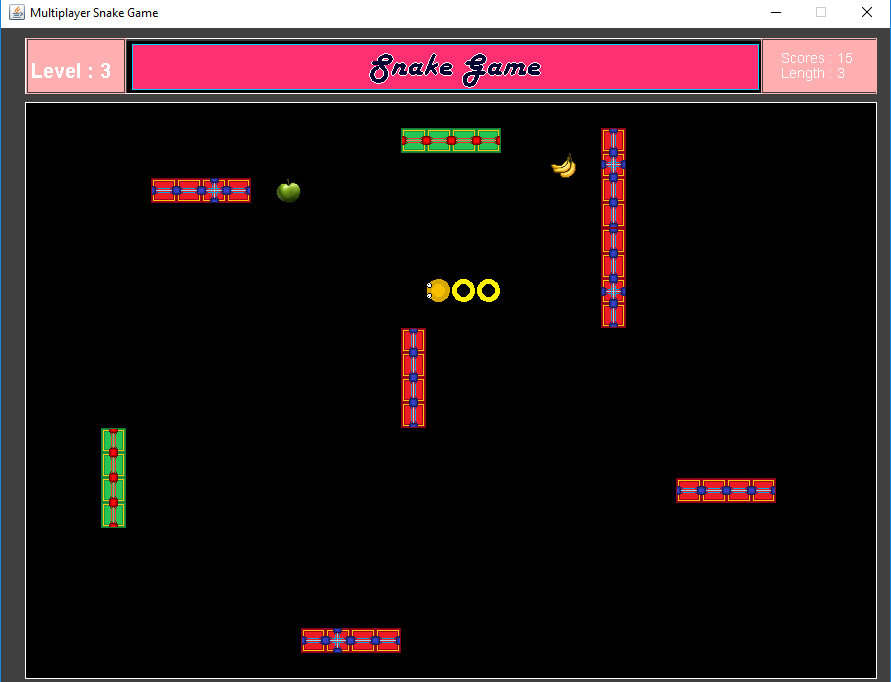
Notice the snake skin, it has changed.. it will keep changing skin on every level



Level 3 :

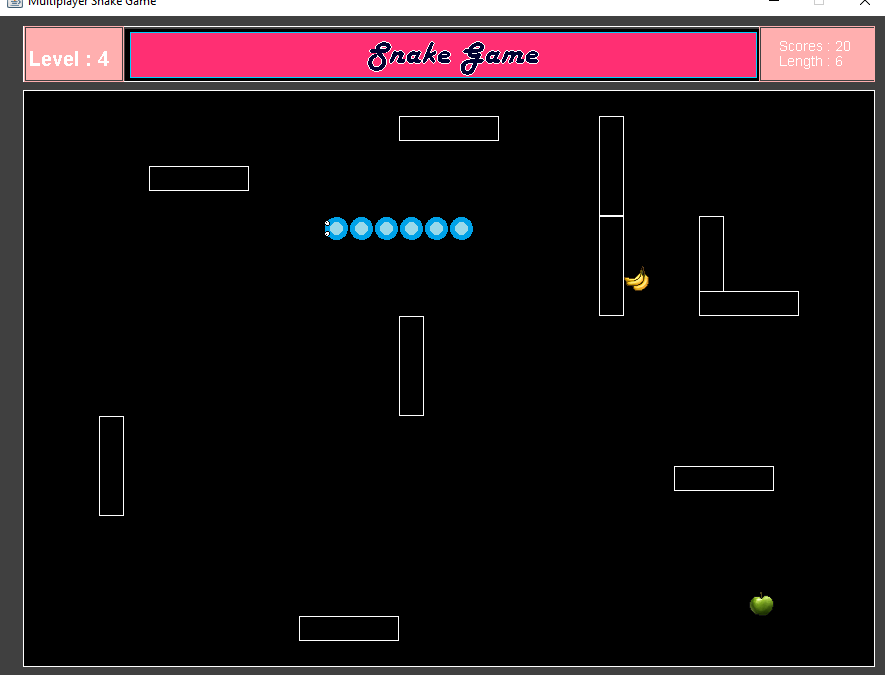
Game Cheat : Player can raise their level by pressing ENTER however, snake length wont increase... and so the score..

This is what happens after pressing ENTER key..



Level 4 :

Snake has 11 types of Skins and 3 types of objects with 5 different colors

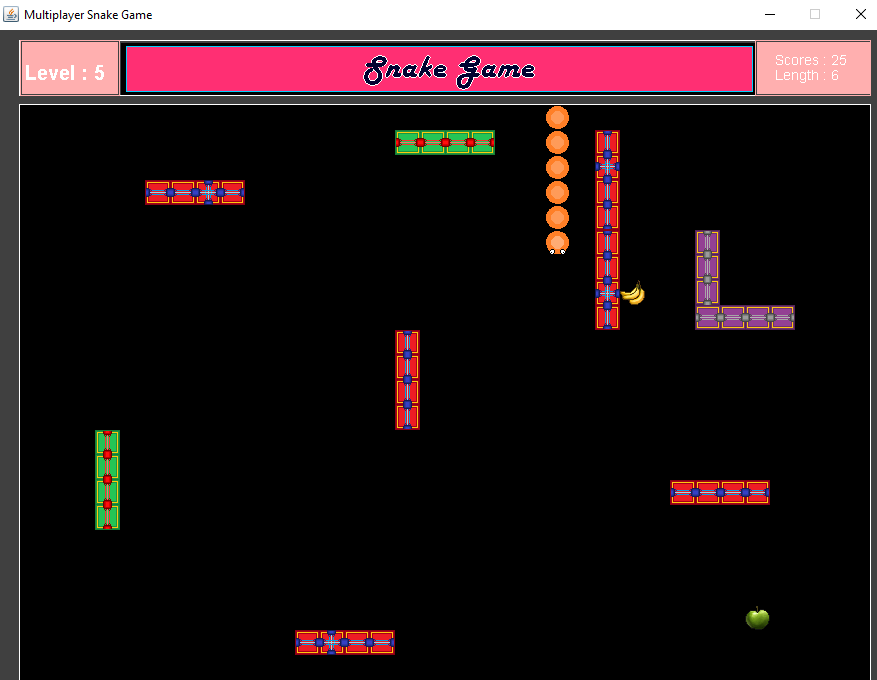


Level 5 :

Here again the skin has been changed.

We can see here three types of objects

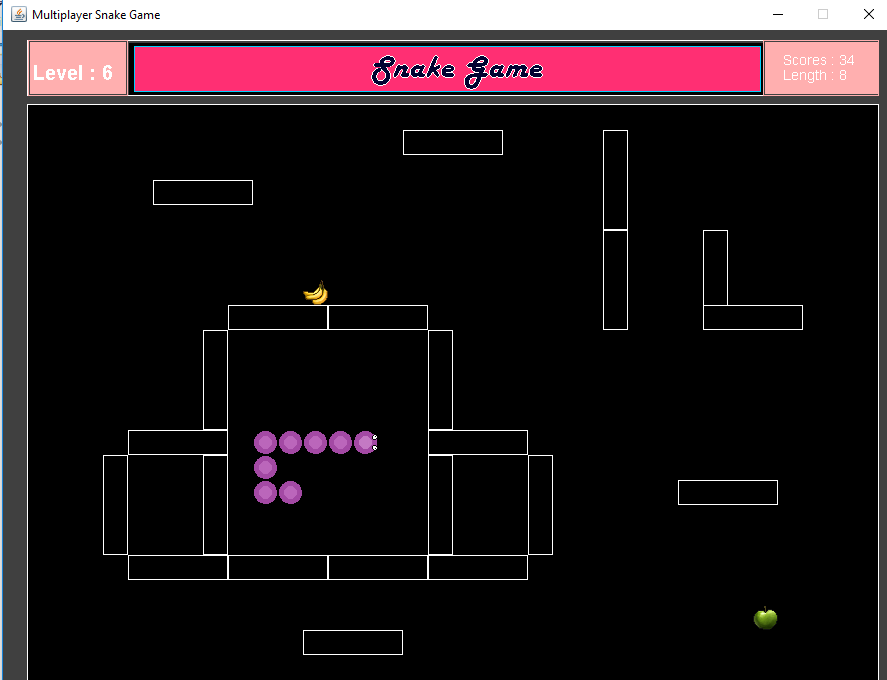
1. Nill type, it means it doesnt have any pipes
2. Right type, the pipe is in 2nd last from right
3. Left Type , the pipe is in 2nd last from left



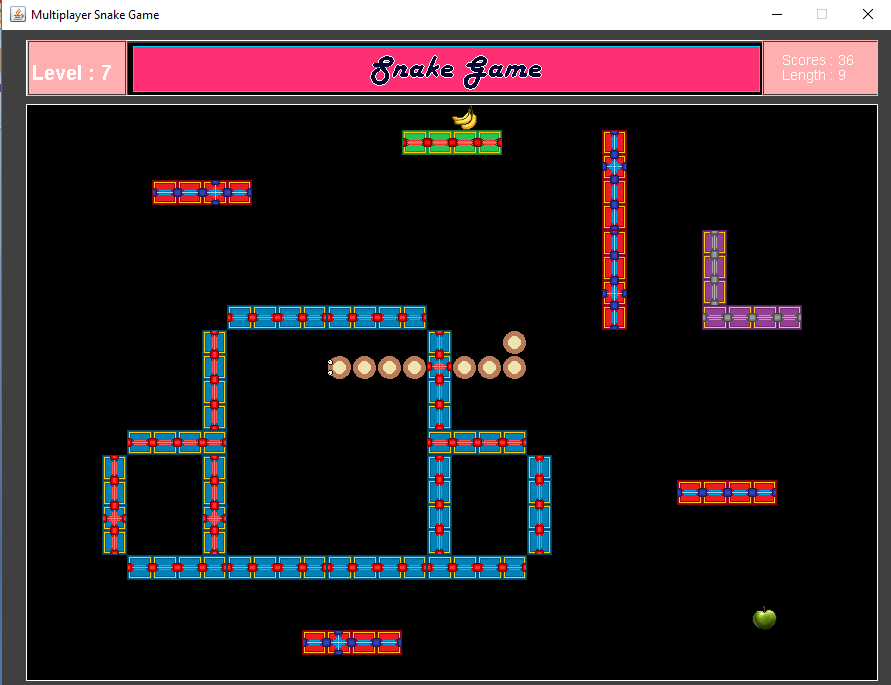
Level 6 : This is the special stage.

I tried to make a monkey face using the objects

The screenshot below after this one shows the solid objects



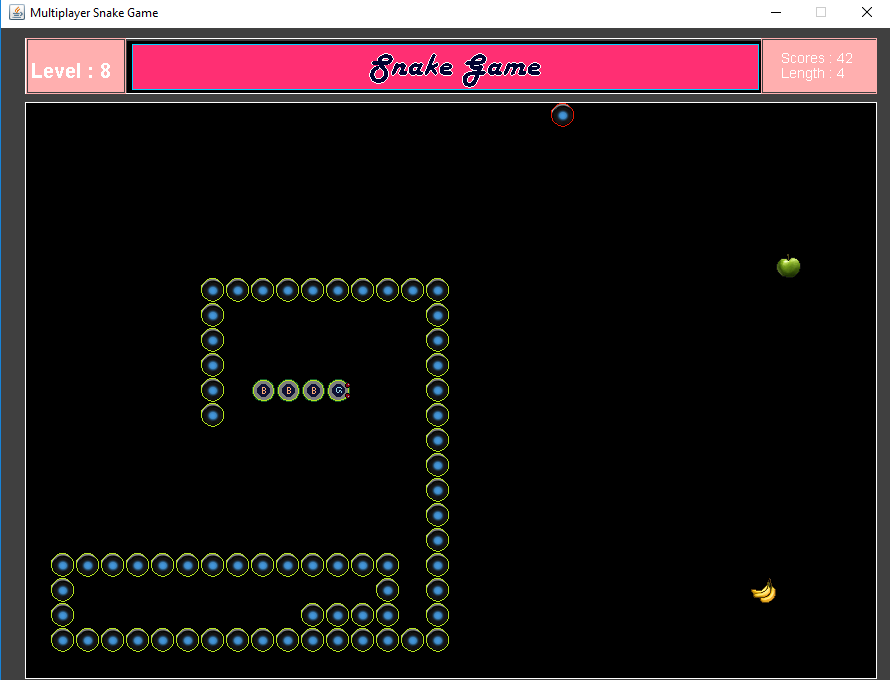
Level 7: This is the stage with the monkey face in it.



Level 8 : Enemy grows automatically with time and Enemy Food also appears when its eaten by the enemy.

As the player changes the snake direction, another letter appears on its head..

Its Based On My Friend’s Name : GURI (Player’s Head)



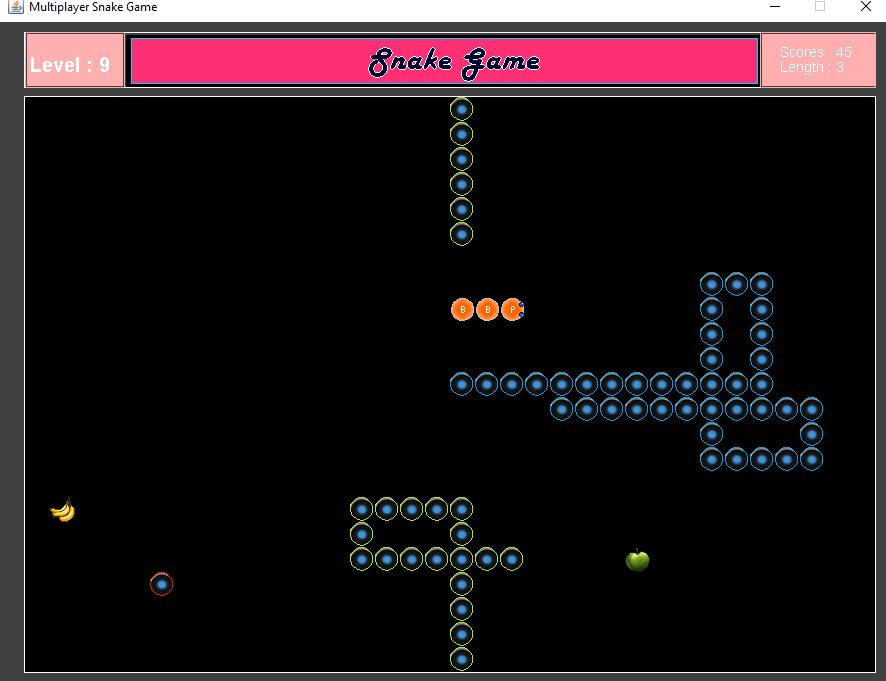
rightmouth.pngdownmouth.pngleftmouth.pngupmouth.png

Level 9 :

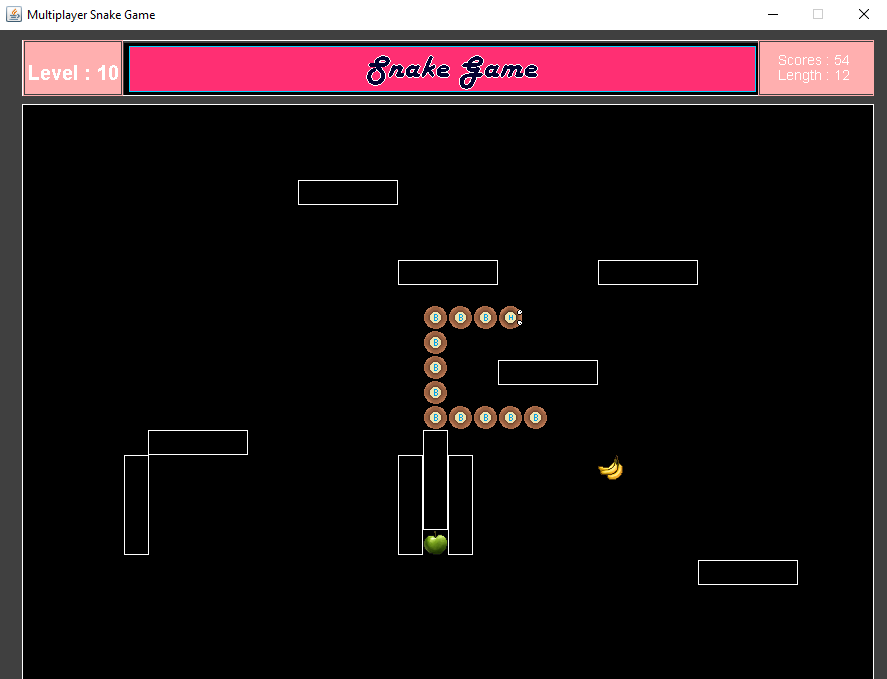
Two Enemies appear on this level. One of the enemy moves opposite of your direction and another one randomly.

My Friend’s Name : PUBG (Player’s Head)

The enemies keep growing as time passes... So you gotta hurry or you die here on this level

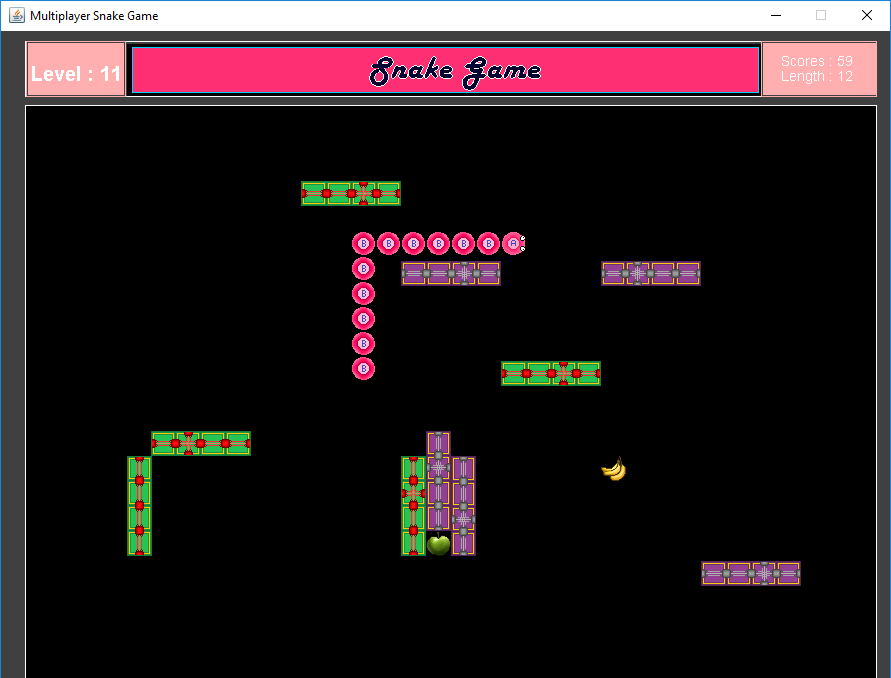


upmouth.pngleftmouth.pngdownmouth.pngrightmouth.png

Level 10 : My Friend’s Name : HIPPO (Player’s Head)

upmouth.pngleftmouth.pngdownmouth.pngrightmouth.png

Level 11: My Friend’s Name : ANKU (Player’s Head)



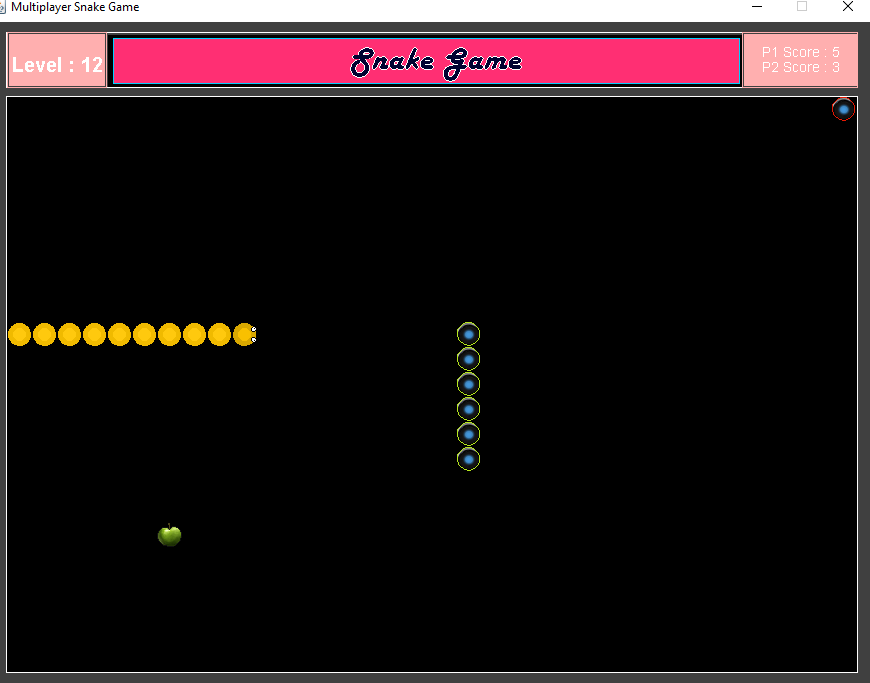
upmouth.pngleftmouth.png

downmouth.pngrightmouth.png

Level 12 : Multiplayer Mode at level 12

Player 2 has to use a,s,d,w to move here and there..

P1 score and P2 score also appears



**CONCLUSION**

Well, I really enjoyed making this project. This project pushed me up to learn lots of things. I had begun making this project without even knowing java, day by day after spending 14-16hrs per day, i was able to make something like this. Thats why it took me more than 20 days.

This project is just the beginning of my game life journey.

I would like to thank my teacher Ms. Palak Mam for so much support and having faith on me. And then Ms. Davinder Mam for helping me in making this project’s Synopsis and other written work.

External help was provided from Ms. Inderpreet Mam, Thanks to her too..

Finally, i would like to thank my friends who always believed in me and supported me mentally to overcome my problems. This project could not be completed without their support. They mean alot to me.

THANKS EVERYONE!!!!

**FUTURE SCOPE :**

This project can be further extended using some more functionalities.

The following are the future scope of this project..

* Online Multiplayer
* Motile Objects like IceBerg(Freezes snake for a while), FireBall (Decreases Snake Length), Speed Booster (Speeds up Snake), SpeedReducer (Slows Down Snake) etc.
* Several New types of Skin can be added with awesome graphics.
* There can be Easy, Moderate and Hard Mode

**REFERENCES**

<http://www.odec.ca/projects/2009/reci9l2/conclusion.htm>

<http://ahumbleopinion.com/the-classic-snake-game/>

<https://en.wikipedia.org/wiki/Snake_(video_game_genre)>

<http://www.academia.edu/12695666/Snake_game_documentation>

<https://www.slideshare.net/azharniaz3/report-on-snake-game>

<https://www.youtube.com>

<https://www.tutorialspoint.com/>

<https://www.javatpoint.com/>

<https://www.w3schools.com/java/default.asp>

<https://www.guru99.com/java-tutorial.html>

<https://www.udemy.com/java-tutorial/>

<https://www.softwaretestingmaterial.com/java-tutorial/>

<http://zetcode.com/tutorials/javagamestutorial/snake/>

<http://www.utdallas.edu/~veerasam/Alice_Java/GUI/Snake.java>

<https://codereview.stackexchange.com/questions/129719/java-snake-game>

<https://github.com/hexadeciman/Snake>

<https://dzone.com/articles/how-to-make-snake-game-in-java-swing>