OOPM MINI PROJECT

GAME: Brick Breaker

Group members:

Prabhat Jaiswar 15

Abhijeet Gaonkar 10

Saurav Bhosale 03

Sanskruti Jadhav 14

INTRODUCTION

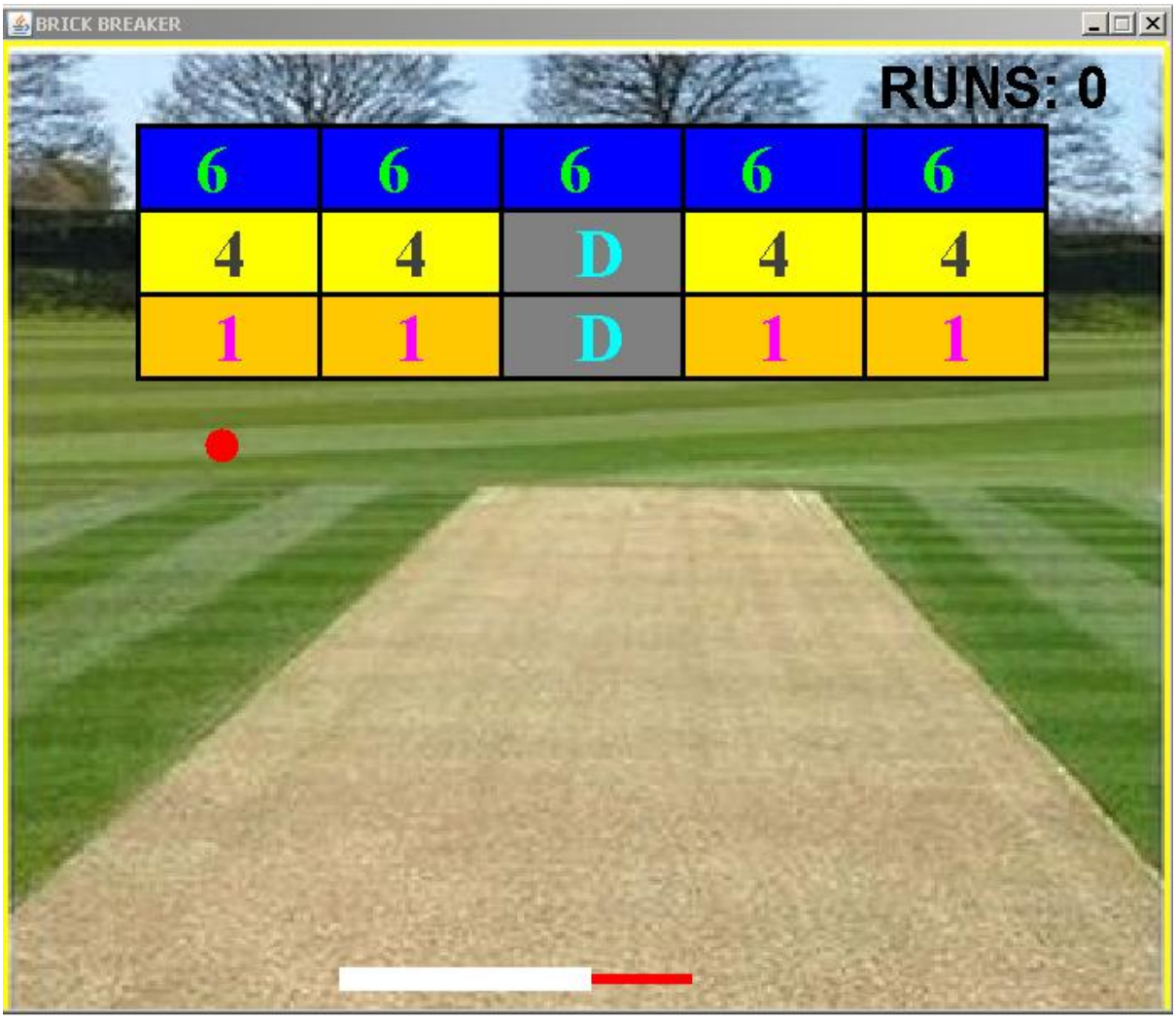
We made an interactive game based upon the classic game brick breaker. The objective of the ball is to break all bricks that are distributed around top of the game screen.

The bricks are numbered as 1, 4 , 6, 2 and score will increase accordingly. At the bottom is the bat that moves horizontally and is according to user input. The user has to make sure the ball bounces off the bat without going off the bottom screen.

We have introduced 2 levels in the game.

Level 1

Level 1 has bricks distributed in the 3:5 rows and columns .First row consist of bricks with number 6 hence if you break these bricks you score 6 points for each brick . Second row consist of bricks with number 4 and D which is dead ball hence if you hit D your score doesn't change and if you hit bricks with 4 your score increments with 4 . Third row consist of bricks with number 1 and D which is dead ball hence if you hit D your score doesn't change and if you hit bricks with 1 your score increments with 1 .If you break all the bricks in level 1 you score a half century.

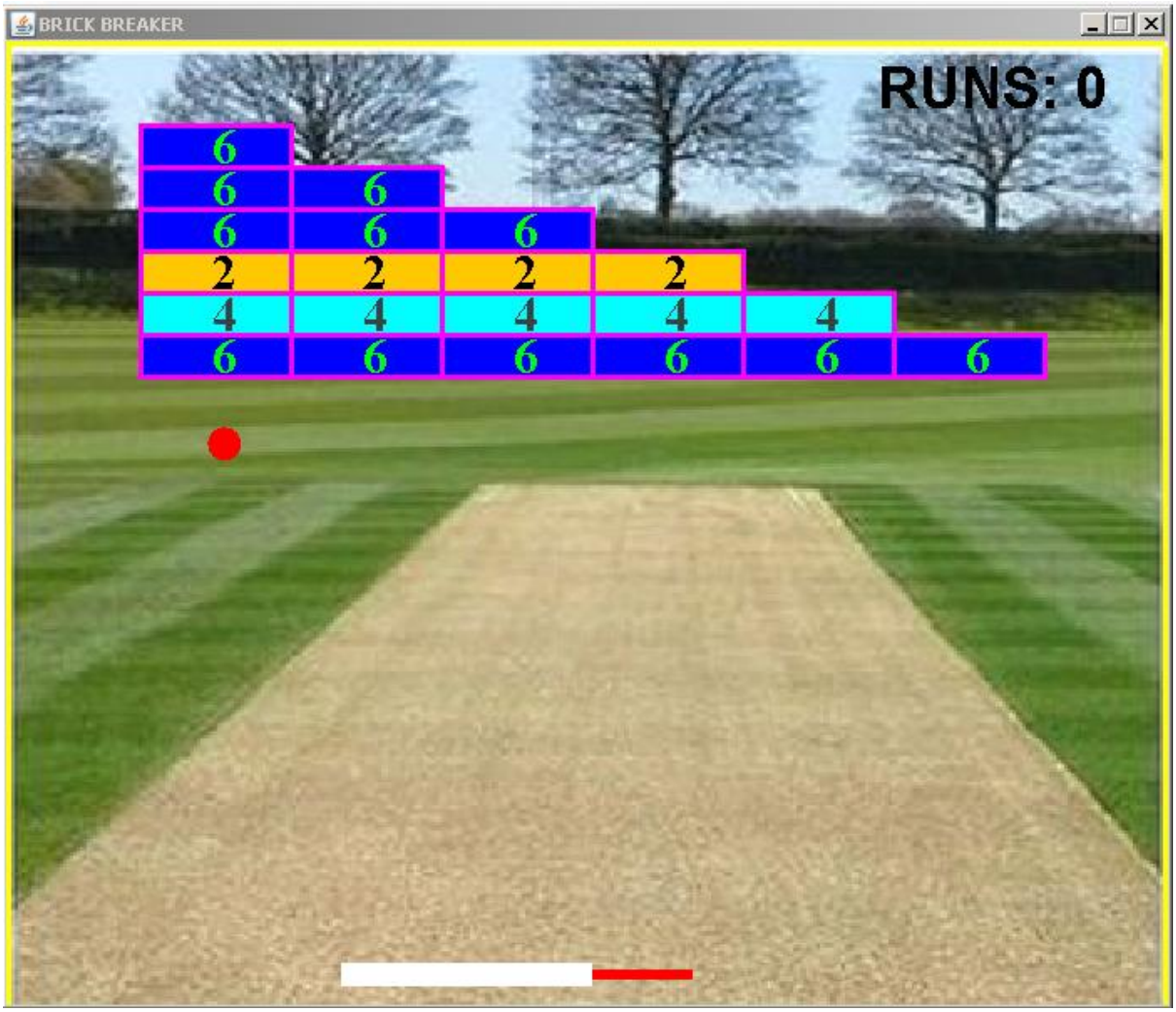


LEVEL 1

Level 2

Level 2 has bricks arranged in the half pyramid pattern as shown below.

And if you break all the bricks in level 2 you score a century.



LEVEL 2

<u>The game consists of 5 files:</u>

MainClass.java

GamePlay.java

MapGenerator.java

Level2.java

MapLevel2.java

Package demogame has been introduced

This package consists of class called MainClass

JFrame object has been created and title has been set as Brick Breaker, along with that other parameter are created like size, location, resizeable ,etc

After that a new class has been created " GamePlay"

The class GamePlay is added to Main Class

Buttons are created for level 1 ,level2,help and quit using Jbutton.

Menu

In menu, we have first created a JFrame and then we have added a background image to it.

We created four buttons(b1,b2,b3,b4) on it using JButton.

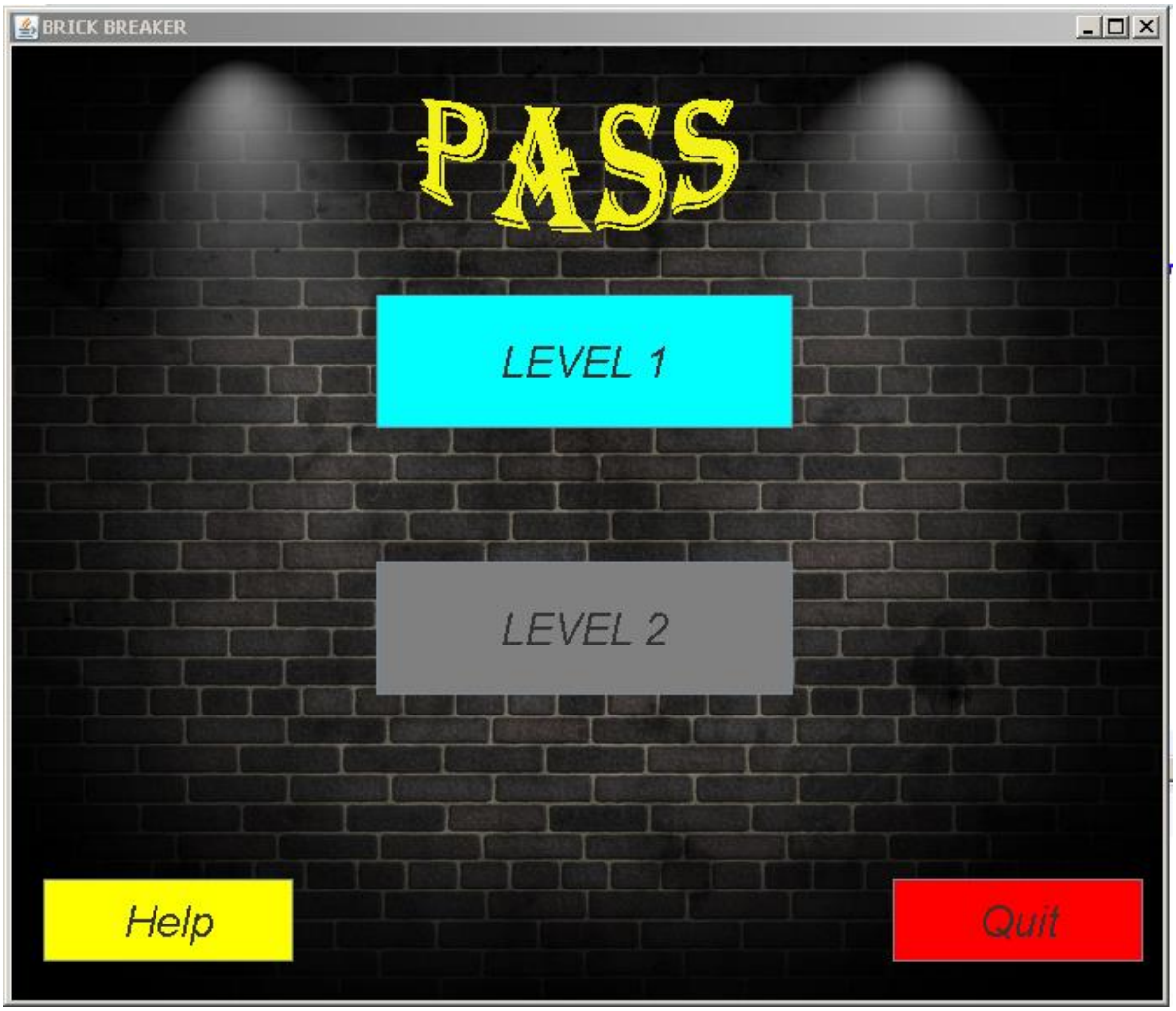
b1 = Level 1

b2 = Level 2

b3 = Help

b4 = Quit

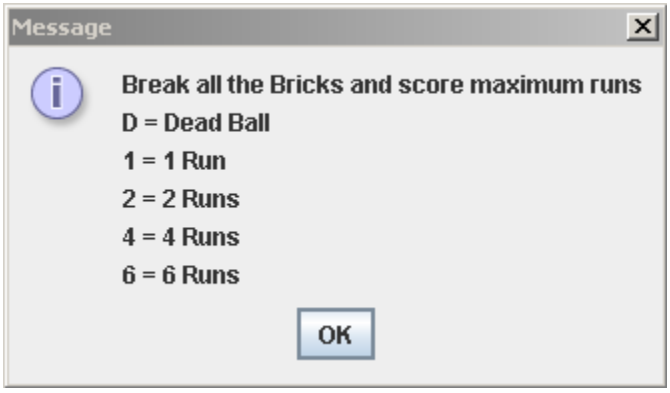
And thus we created our menu.



Menu

Help button

When you click on Help button a message is displayed.



HELP DIALOGUE BOX

Class GamePlay

First of all, we have imported all the required packages for our program.

Then all the necessary variables like score, totalBricks, timer, delay, ballposX,ballposY,ballXdir,ballYdir,playerX have been declared and set to a particular value.

GamePlay() Constructor has been created.

Inbuilt method paint created and argument “graphics g” is passed.

Background image is added with dimensions (692,592)

Bat created using two rectangles, one with dimensions(150,14) and other with dimensions(60,6) and there x and y positions are set so that it looks like a bat.

Ball created with dimensions(ballposX,ballposY,20,20)

Actionlistener and keylistener are implemented.

ActionListener

ActionListener in Java is a class that is responsible in handling all action events such as when the user clicks on a component. Mostly, action listeners are used for JButton. An ActionListener can be used by the implements keyword to the class definition.

Keylistener

The listener interface for receiving keyboard events (keystrokes). The class that is interested in processing a keyboard event either implements this interface (and all the methods it contains) or extends the abstract KeyAdapter class (overriding only the methods of interest).

The listener object created from that class is then registered with a component using the component's addKeyListener method. A keyboard event is generated when a key is pressed, released, or typed. The relevant method in the listener object is then invoked, and the KeyEvent is passed to it.

In our case we use two methods keypressed and actionperformed.

Inside keypressed

Left key , right key along with it “A” and “D” are used for left and right movement respectively.

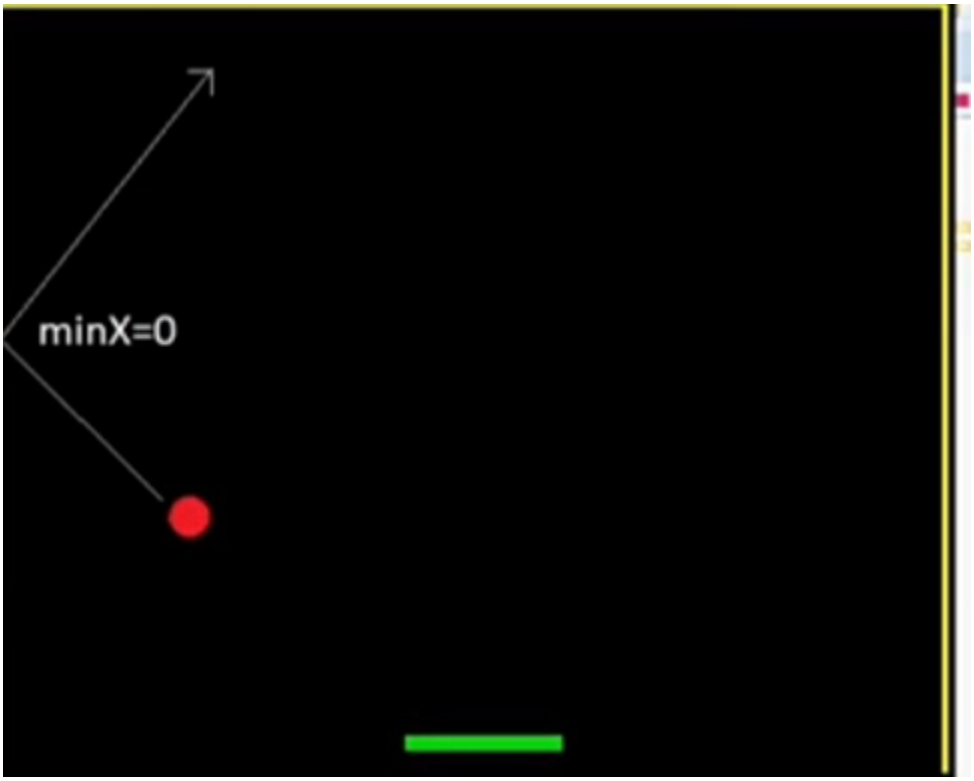
Inside actionperformed

Check play is true then inside the loop the condition for ball is written

condition applied for ball

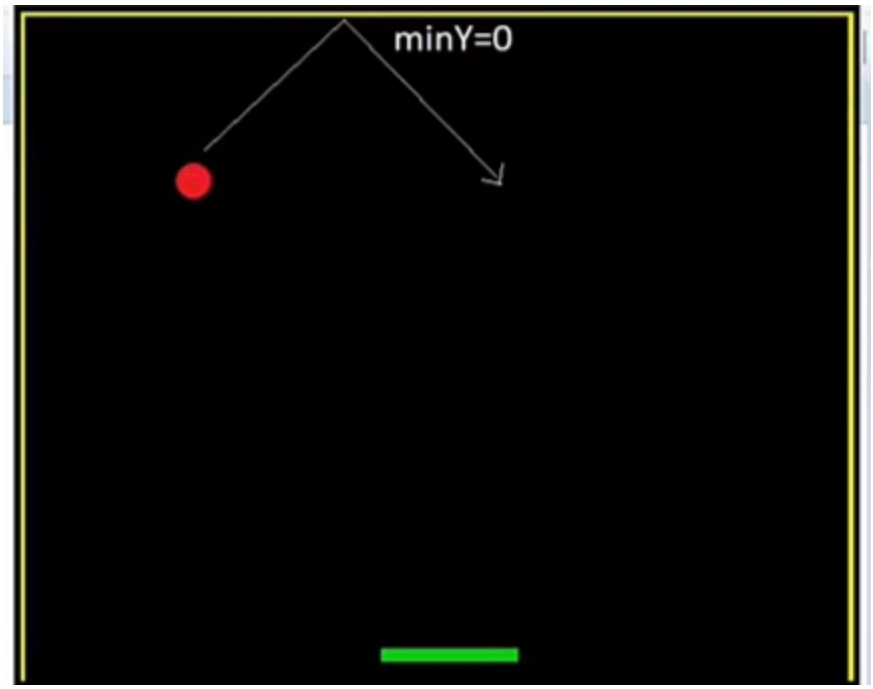
ballposX<=0

ballposX+=ballXdir



ballposY<=0

ballposY+=ballYdir



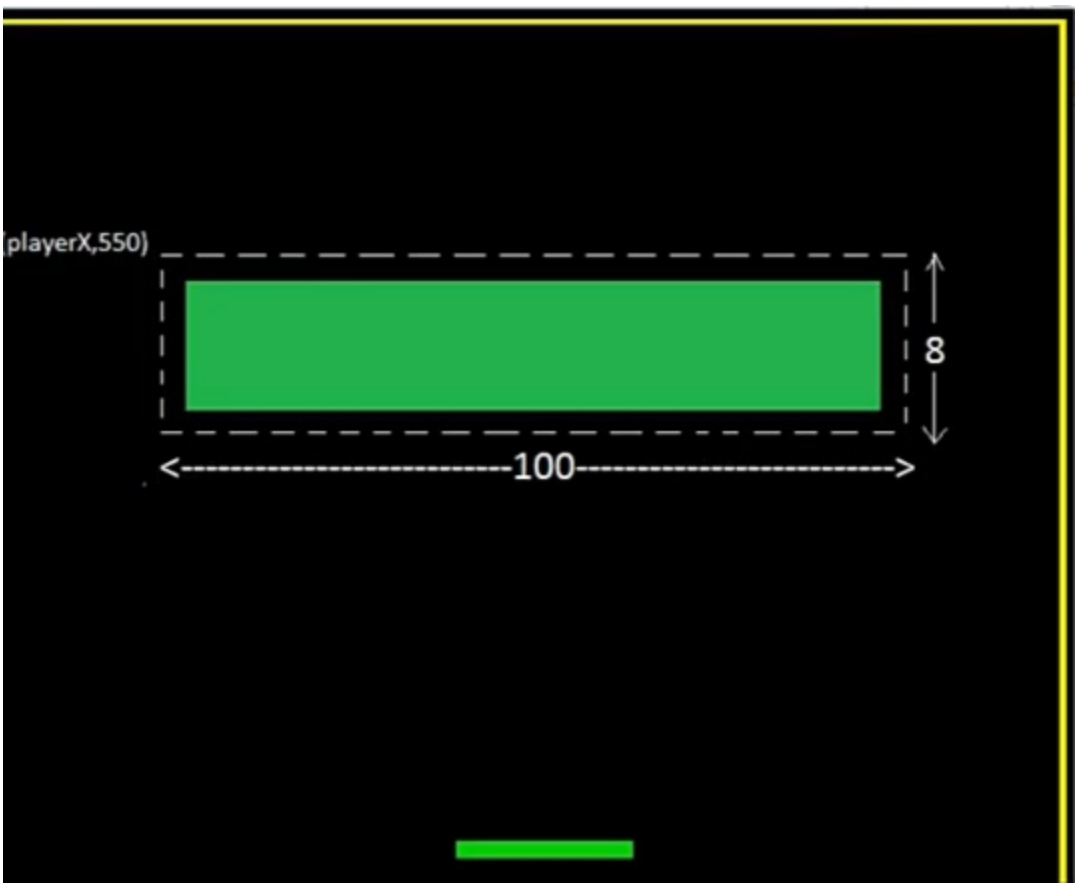
For position of paddle

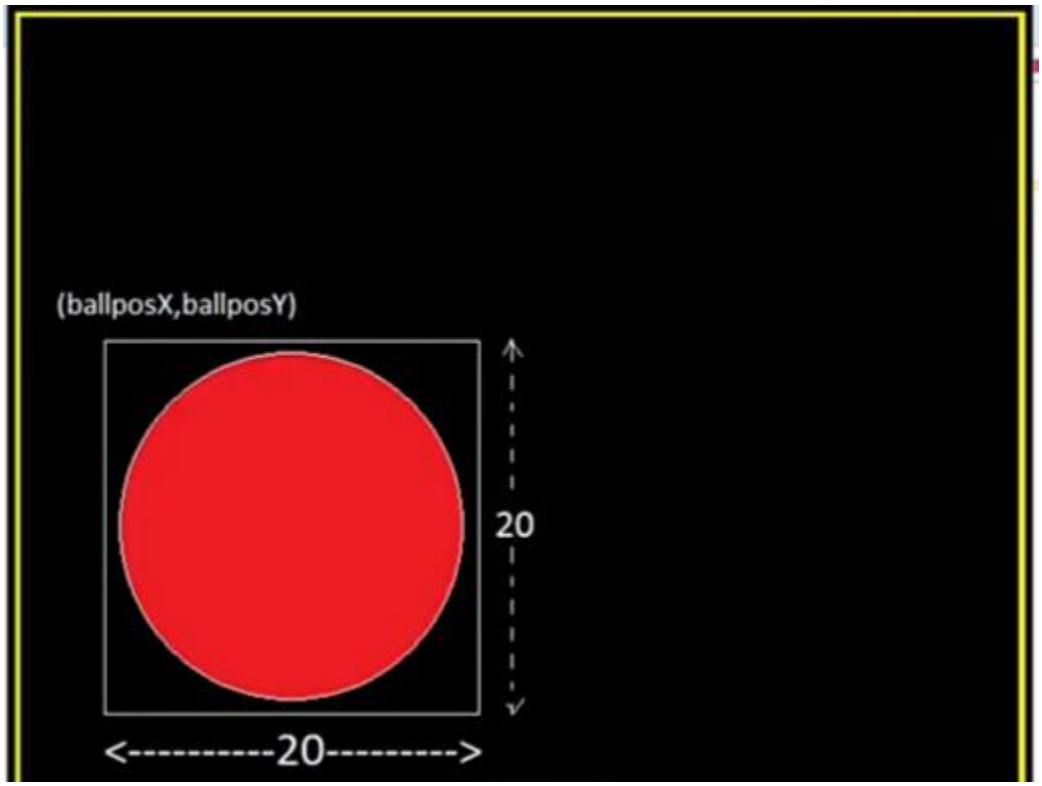
Ballrect is created and balls X position and Y position is written

Similarly written for paddle

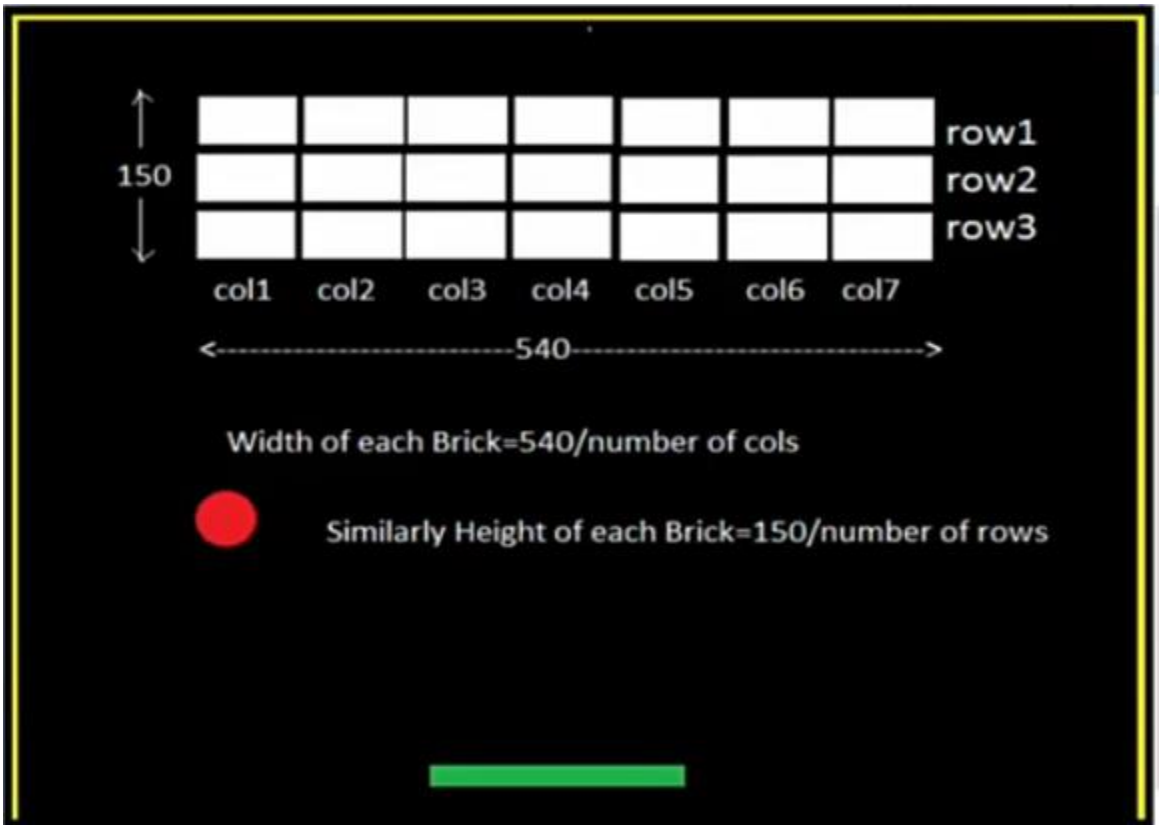
If ballrect intersects paadlerect

BallYdir is changed


(playerX,550)
8
100


(ballposX,ballposY)
20
20

Class MapGenerator



For making rows and columns a 2darray is made and width and height of bricks mentioned.

A constructor is created and parameters rows and col are used in it.

And loop is made for rows and columns which is explained above.

Bricks

Bricks are made using paint inside graphics and dimensions are mentioned.

And border for the bricks are made.

Intersection of ball and brick

If ballrect intersects brickrect then value of bricks becomes 0 and total number of bricks decreases.

After collision the ball should rebound in opposite position .

if(ballposX+19<=brickXpos ||

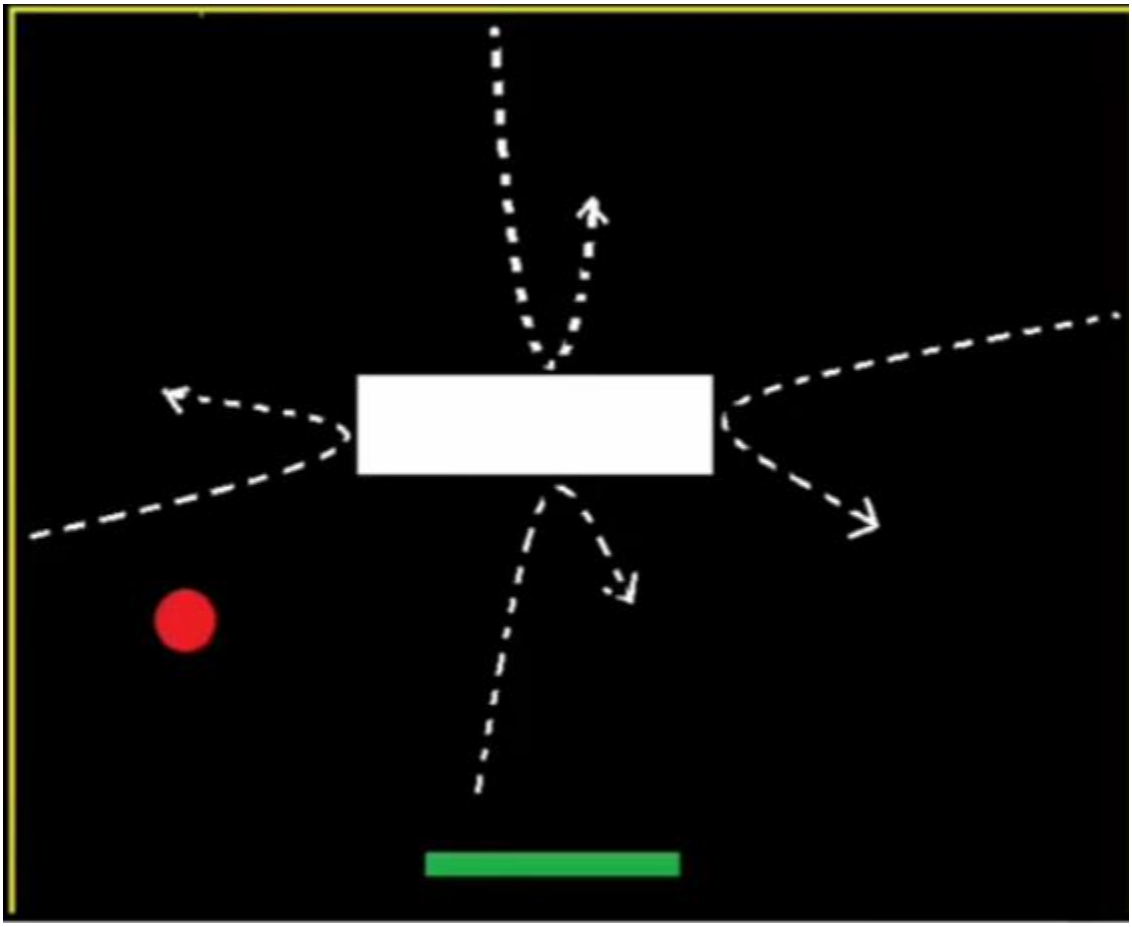
ballposX+1>=brickXpos+width) {//Change of Direction when the ball intersects with brick.

ballXdir=-ballXdir;

}

else {

ballYdir=-ballYdir;



For displaying score

A variable is declared whose intial value is 0

And in the paint method the score is drawn.

Its color is set along with its X position and Y position.

Message diplaying game over and score is displayed

Exactly in the same way level 2 is created with diiferent patterns of brick.

Background

We imported images for background using bufferimage in level 1,level 2 and menu.

LEVEL 1 ,LEVEL 2

Menu

Methods used in the program:

- paint();

  Used to paint 2D Graphics on JFrame.
- moveLeft();

  Used to move bat towards left.
- moveReft();

  Used to move bat towards right.
- setBrick();

Used to refresh the brick pattern when ball hits a brick.

- repaint();

  Used to repaint the 2D graphics after every game loop.

Constructers used in program:

- GamePlay()

  Creates gameplay(i.e level 1) of game.

- MapGenerator()

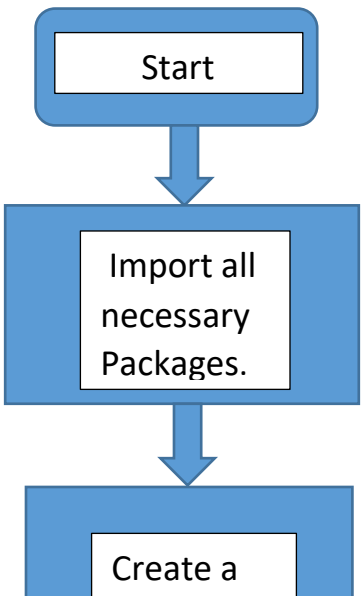
  Creates an array of bricks of given size.

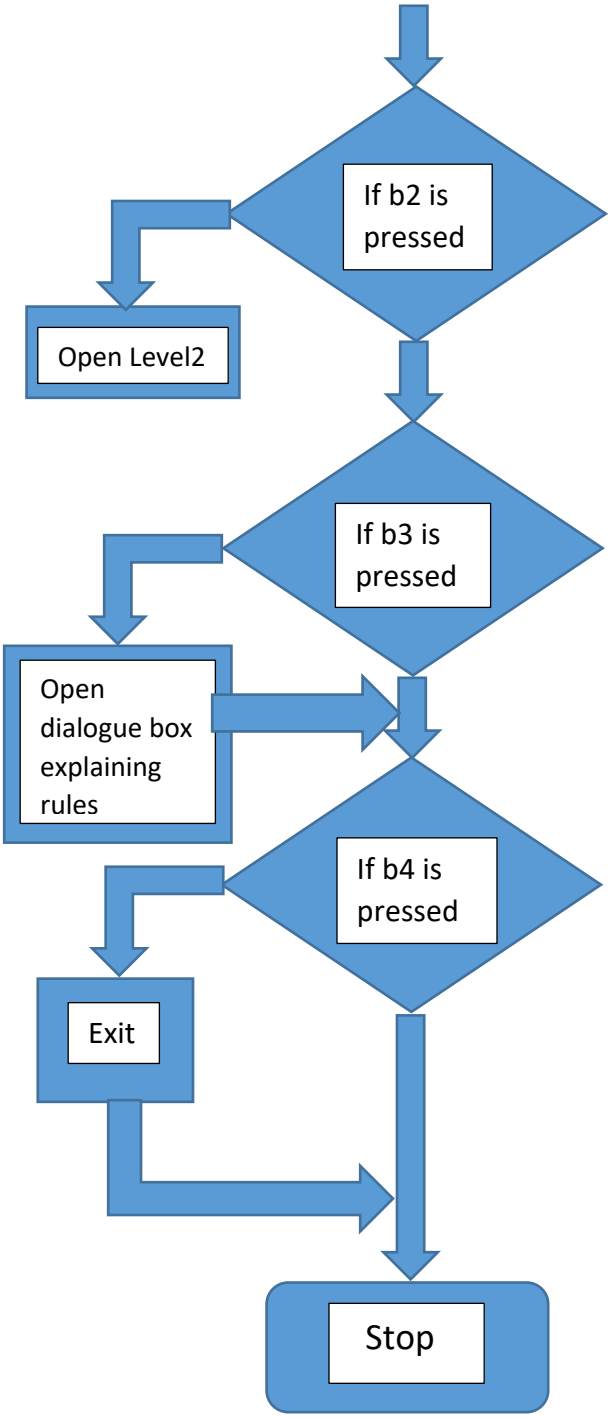
Thank You

----------Enjoy The Game----------

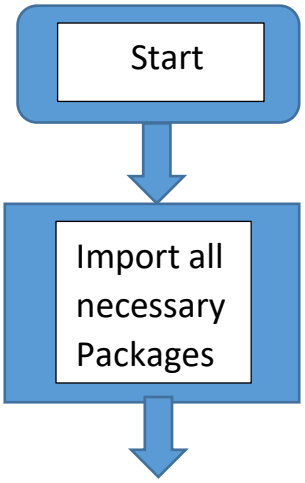
FLOWCHART

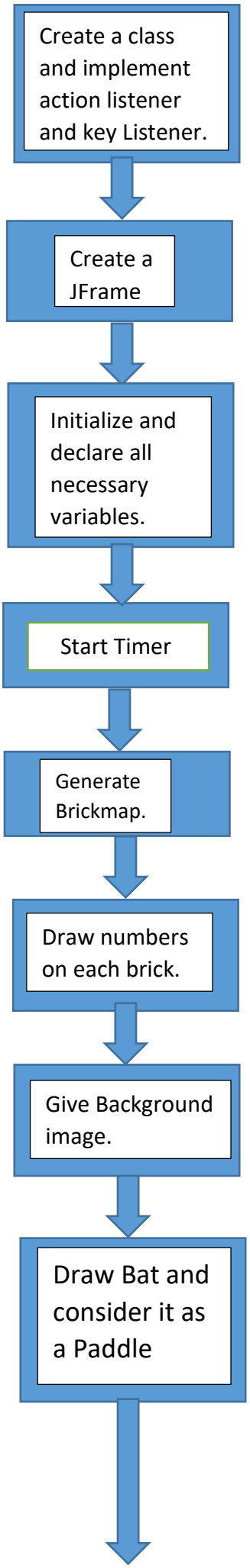
MENU FLOWCHART

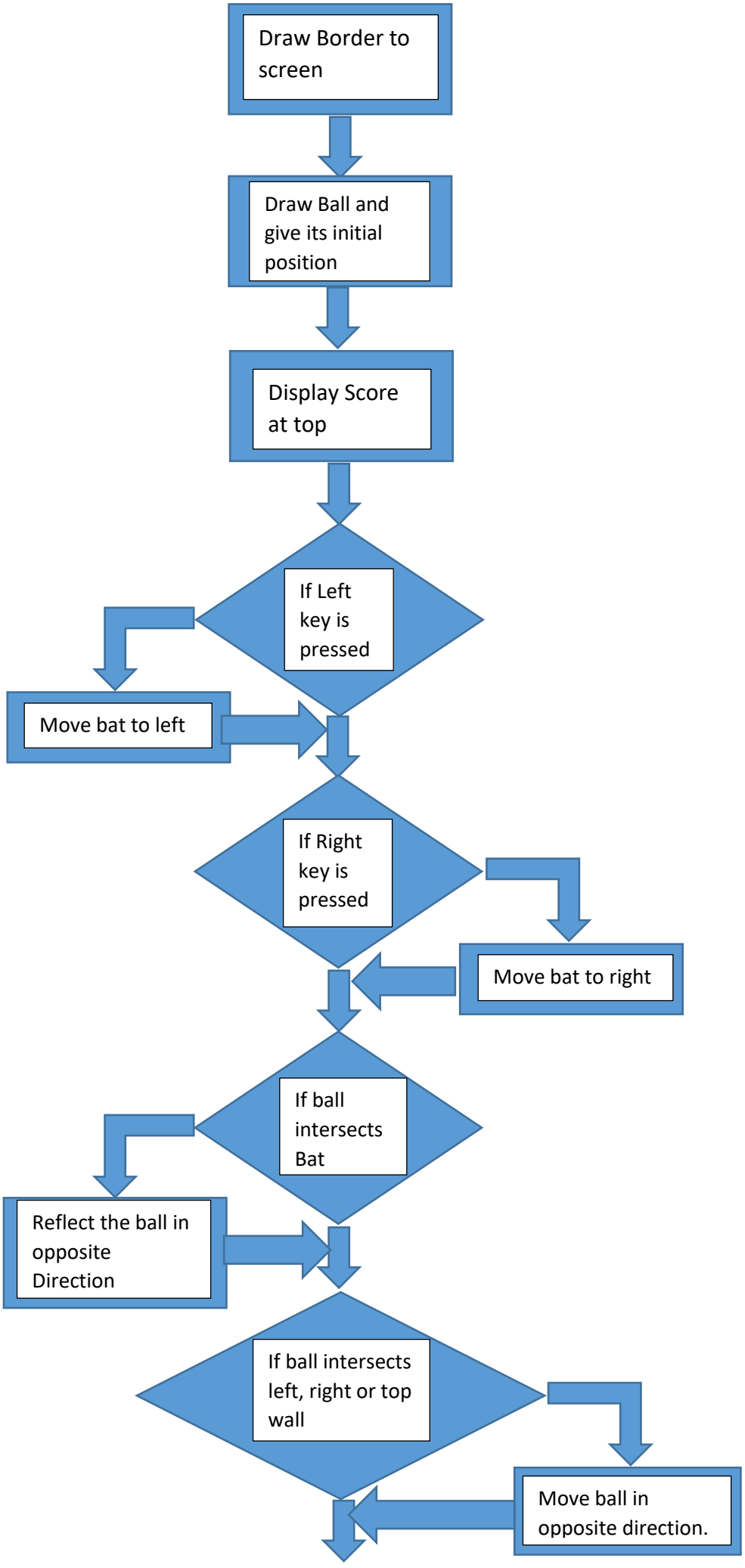


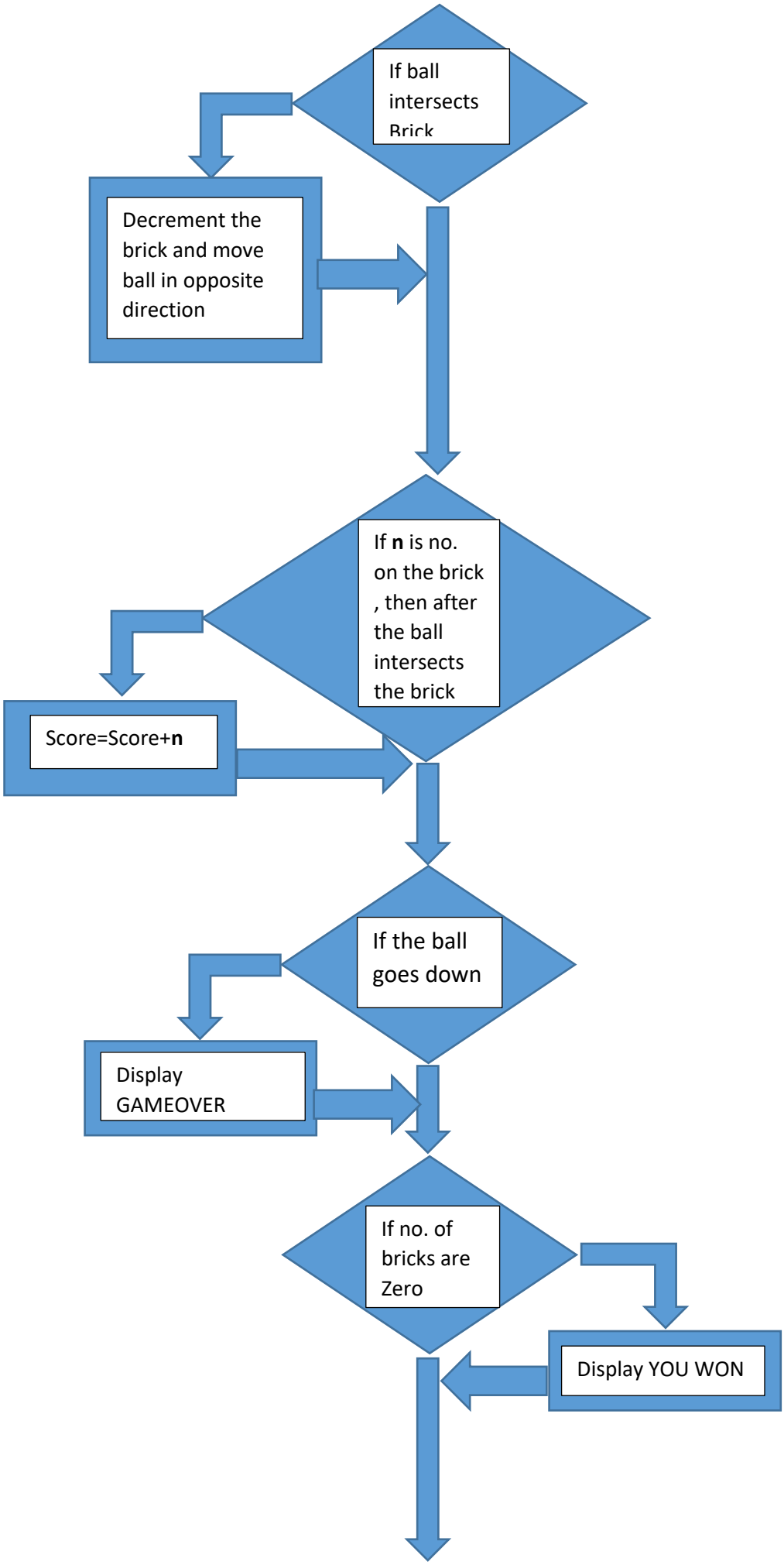


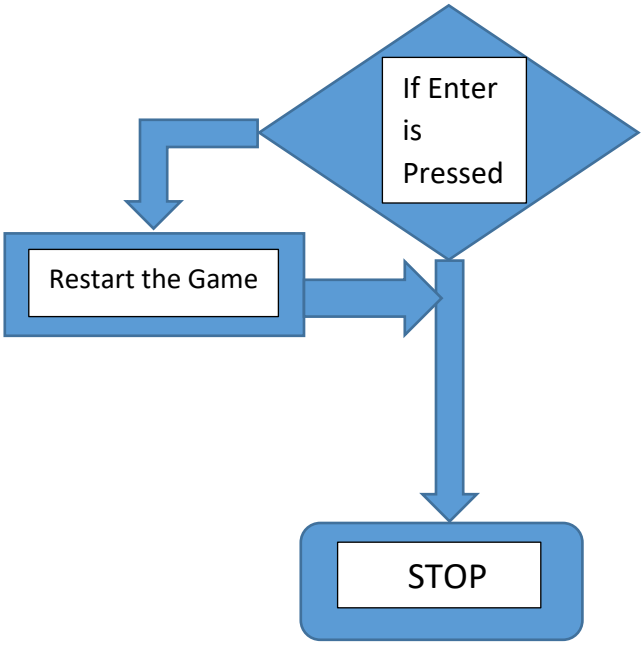
Gameplay Flowchart



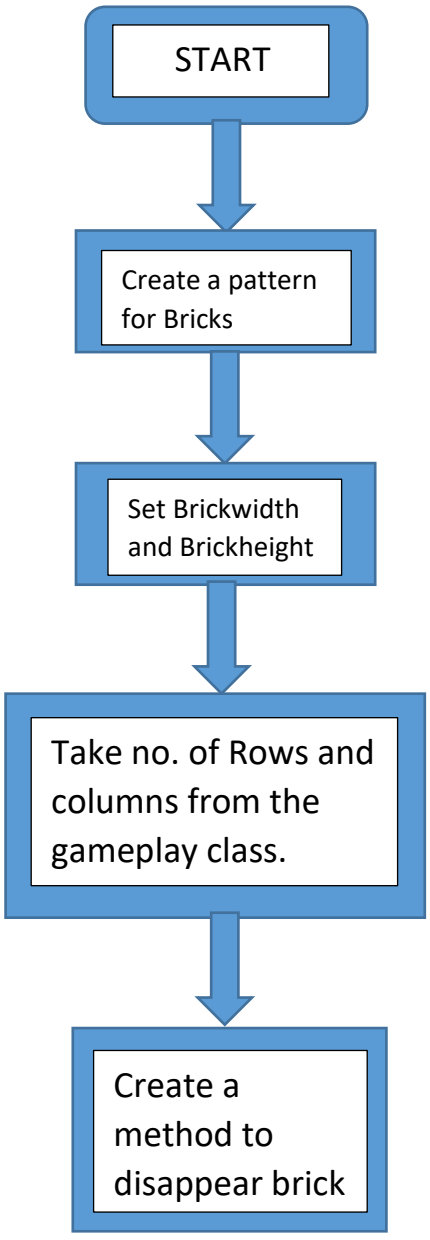

Create a class and implement action listener and key Listener.
Create a JFrame
Initialize and declare all necessary variables.
Start Timer
Generate Brickmap.
Draw numbers on each brick.
Give Background image.
Draw Bat and consider it as a Paddle

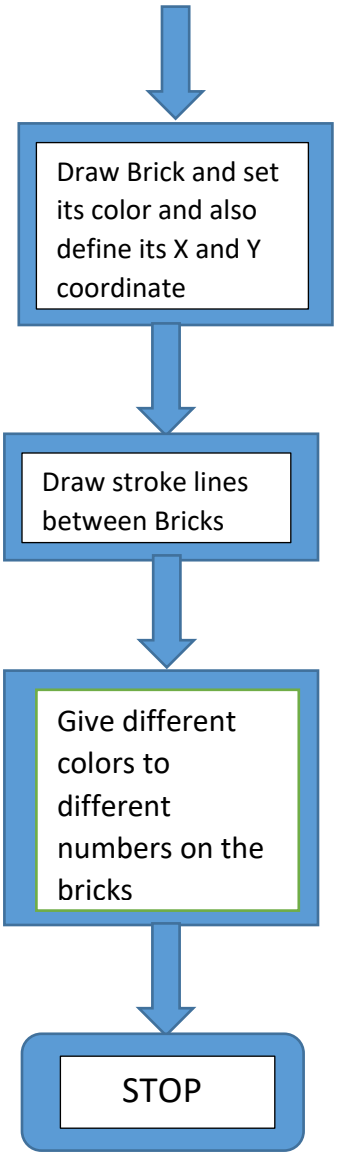

Draw Border to screen
Draw Ball and give its initial position
Display Score at top
If Left key is pressed
Move bat to left
If Right key is pressed
Move bat to right
If ball intersects Bat
Reflect the ball in opposite Direction
If ball intersects left, right or top wall
Move ball in opposite direction.


If ball intersects Brick
Decrement the brick and move ball in opposite direction
If **n** is no. on the brick , then after the ball intersects the brick
Score=Score+**n**
If the ball goes down
Display GAMEOVER
If no. of bricks are Zero
Display YOU WON



MAP GENERATOR FLOWCHART




Draw Brick and set its color and also define its X and Y coordinate
Draw stroke lines between Bricks
Give different colors to different numbers on the bricks
STOP