**CS-1002L**

**OBJECT-ORIENTED PROGRAMMING**

**BRICK\_BREAKER GAME**

**Project Report By  
cs201003-Abdul Malik**

**Group Members:**

**<<AFTAB AHMED>> <<cs201010>>**

**<<Aamir Saleem>> <<cs201002>>**

**<<Abdul Waleed Kalwar>> <<cs201006>>**

Table of Contents

[1.Introduction](#_Toc511892420)

[2.Scope](#_Toc511892421)

[3UML Class **.**](#_Toc511892422)

[4. Java code](#_Toc511892423)

[5. Hardware/Software Requirements](#_Toc511892424)

[5.1 Tools](#_Toc511892425)

[5.2 Libraries , IDE \](#_Toc511892426)

[6. Screen Shots](#_Toc511892427)

[7. Task Sheet](#_Toc511892428)

# Introduction:

The Project is Consist of Game which is Known as Brick Breaker Game

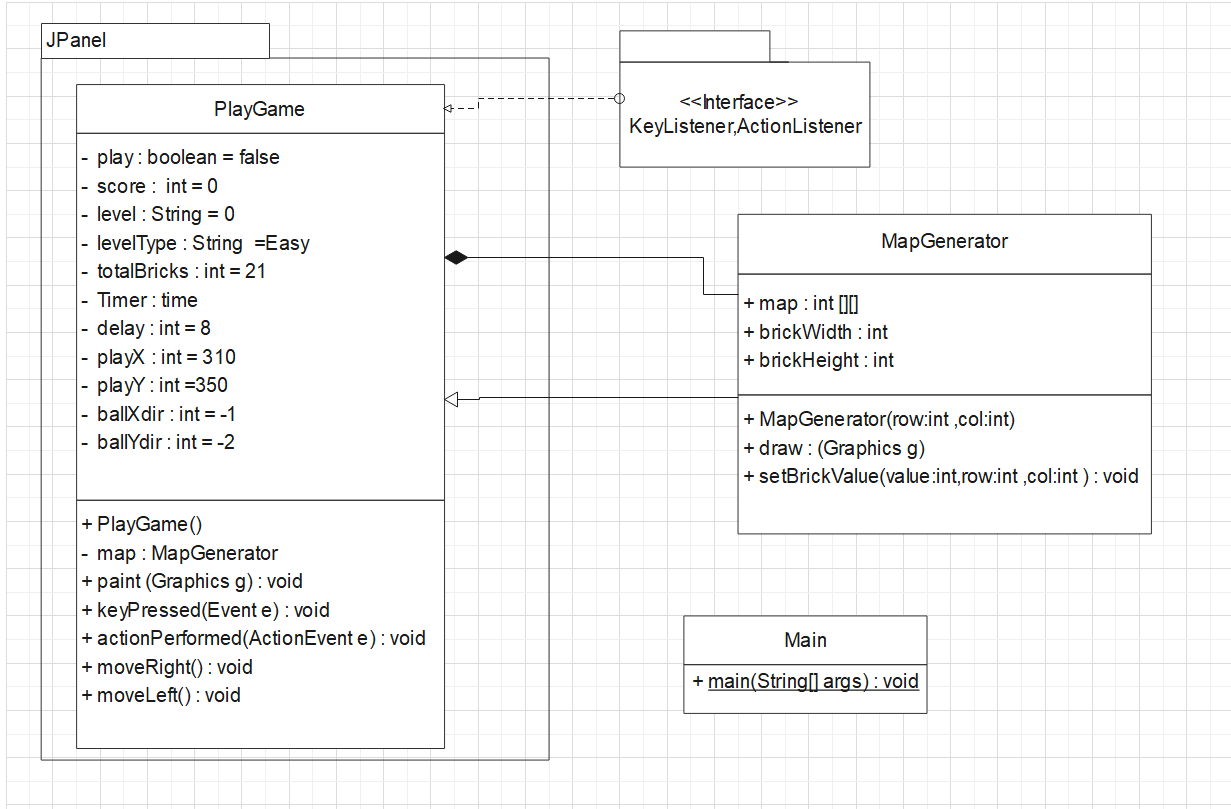
Objectives that we need to create this Game

* + 1. Window(JFRAME)
    2. Bricks
    3. Ball
    4. Paddle

# 2. Scope

The objective of the game is to break all of the bricks that are on the top of the screen by deflecting the ball with the bottom bar. As the game progresses it becomes more challenging as asteroid objects come onto the screen and can also deflect the ball.

# UML \_DIAGRAM



# JAVA Code

MapGenerator\_Class

package com.company.java\_project;

/\*\*

\*

\* @author Abdul Malik

\*/

import java.awt.BasicStroke;

import java.awt.Color;

import java.awt.Graphics2D;

public class MapGenerator {

public int map[][];

public int brickWidth;

public int brickHeight;

public MapGenerator(int row,int col){

map = new int[row][col];

for(int i = 0; i<map.length;i++){

for(int j=0;j<map[0].length;j++){

map[i][j] = 1;

}

}

brickWidth = 540/col;

brickHeight = 150/row;

}

public void draw(Graphics2D g){

for(int i = 0; i<map.length;i++){

for(int j=0;j<map[0].length;j++){

if(map[i][j] > 0){

g.setColor(new Color(211, 84, 0));

g.fillRect(j\*brickWidth+80,i\*brickHeight+50, brickWidth, brickHeight);

g.setStroke(new BasicStroke(3));

g.setColor(Color.white);

g.drawRect(j\*brickWidth+80,i\*brickHeight+50, brickWidth, brickHeight);

}

}

}

}

public void setBrickValue(int value, int row, int col){

map[row][col] = value;

}

}

**PlayGameClass**

package com.company.java\_project;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Rectangle;

import java.awt.event.ActionEvent;

import javax.swing.JPanel;

import java.awt.event.ActionListener;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import javax.swing.Timer;

public class PlayGame extends JPanel implements KeyListener,ActionListener{

private boolean play = false;

private int score = 0;

private String level = "01";

private String levelType = "Easy";

private int totalBricks = 10;

private Timer time;

private int delay = 8;

private int playX = 310;

private int ballposX = 120;

private int ballposY = 350;

private int ballXdir = -1;

private int ballYdir = -2;

private MapGenerator map;

public PlayGame(){

map = new MapGenerator(2,5);//4x7 Bricks will be draw

addKeyListener(this);

setFocusable(true);

setFocusTraversalKeysEnabled(false);

time = new Timer(delay,this);

time.start();

}

@Override

public void paint(Graphics g){

//Backgrounds

g.setColor(new Color(33, 47, 60));

g.fillRect(1, 1, 692, 592);

//map

map.draw((Graphics2D)g);

//Score+Level

g.setColor(new Color(3, 255, 70));

g.setFont(new Font("Arial",Font.BOLD,25));

g.drawString("Level: "+level,30,30);

g.setFont(new Font("Arial",Font.BOLD,25));

g.drawString(" "+levelType,320,30);

g.setFont(new Font("Arial",Font.BOLD,25));

g.drawString("Score: "+score,560,30);

//Paddle Colors

g.setColor(new Color(173,255,47));

g.fillRect(playX, 550, 100, 8);

//the Ball

g.setColor(new Color(252,251,13));

g.fillOval(ballposX , ballposY, 20, 20);

if(score==50 && totalBricks <= 0){

play = false;

ballXdir = 0;

ballYdir = 0;

g.setColor(new Color(88, 214, 141));

g.setFont(new Font("serif",Font.BOLD,30));

g.drawString("\*\*\* CONGRATULATION YOU WON!! \*\*\*",75,300);

g.setFont(new Font("serif",Font.BOLD,20));

g.drawString("Press N Button to Move On Next Level", 160, 330);

}

if(score==105 && totalBricks <= 0){

play = false;

ballXdir = 0;

ballYdir = 0;

g.setColor(new Color(88, 214, 141));

g.setFont(new Font("Arial",Font.BOLD,25));

g.drawString("\*\*\* CONGRATULATION YOU WON 2ND LEVEL!! \*\*\*",55,300);

g.setFont(new Font("Arial",Font.BOLD,20));

g.drawString("Press TAB Button to Move On Next Level", 160, 330);

}

if(score==140 && totalBricks <= 0){

play = false;

ballXdir = 0;

ballYdir = 0;

g.setColor(new Color(88, 214, 141));

g.setFont(new Font("Arial",Font.BOLD,25));

g.drawString("Congratulations you Complete all Level",75,300);

g.setFont(new Font("Arial",Font.BOLD,20));

g.drawString("Now Press Enter Button to Start Again From Level\_01!!", 160, 330);

}

if(ballposY > 570){

play = false;

ballXdir = 0;

ballYdir = 0;

g.setColor(new Color(88, 214, 141));

g.setFont(new Font("Arial",Font.BOLD,30));

g.drawString("Game Over, Score: "+score,190,300);

g.setFont(new Font("Arial",Font.BOLD,20));

g.drawString("Press Enter to Restart", 230, 350);

}

g.dispose();

}

@Override

public void keyPressed(KeyEvent e){

if(e.getKeyCode() == KeyEvent.VK\_RIGHT){

if(playX >= 600){

playX = 600;

}else{

moveRight();

}

}

if(e.getKeyCode() == KeyEvent.VK\_LEFT){

if(playX < 10){

playX = 10;

}else{

moveLeft();

}

}

if(e.getKeyCode() == KeyEvent.VK\_ENTER){

if(!play){

play = true;

ballposX = 120;

ballposY = 350;

ballXdir = -1;

ballYdir = -2;

playX = 310;

score = 0;

level = "01";

levelType="Easy";

totalBricks = 10;

map = new MapGenerator(2,5);

repaint();

}

}

if(e.getKeyCode() == KeyEvent.VK\_N){

if(!play){

play = true;

ballposX = 120;

ballposY = 350;

ballXdir = -1;

ballYdir = -2;

playX = 310;

score = 0;

level = "02";

levelType="Medium";

totalBricks = 21;

map = new MapGenerator(3,7);

if(e.getKeyCode() == KeyEvent.VK\_ENTER){

if(!play){

play = true;

ballposX = 120;

ballposY = 350;

ballXdir = -1;

ballYdir = -2;

playX = 310;

score = 0;

level = "01";

levelType="Easy";

totalBricks = 10;

map = new MapGenerator(2,5);

repaint();

}

}

}

}

if(score==140){

if(e.getKeyCode() == KeyEvent.VK\_ENTER){

if(!play){

play = true;

ballposX = 120;

ballposY = 350;

ballXdir = -1;

ballYdir = -2;

playX = 310;

score = 0;

level = "01";

levelType="Easy";

totalBricks = 10;

map = new MapGenerator(2,5);

repaint();

}

}

}

}

}

@Override

public void actionPerformed(ActionEvent e) {

time.start();

if(play){

if(new Rectangle(ballposX,ballposY,20,20).intersects(new Rectangle(playX,550,100,8))){

ballYdir = -ballYdir;

}

//Remove Bricks

A:for(int i=0;i<map.map.length;i++){

for(int j=0;j<map.map[0].length;j++){

if(map.map[i][j]>0){

int brickX = j\*map.brickWidth+80;

int brickY = i\*map.brickHeight+50;

int brickWidth = map.brickWidth;

int brickHeight = map.brickHeight;

Rectangle rect = new Rectangle(brickX,brickY,brickWidth,brickHeight);

Rectangle ballrec = new Rectangle(ballposX,ballposY,20,20);

Rectangle brickRect = rect;

if(ballrec.intersects(brickRect)){

map.setBrickValue(0, i, j);

totalBricks--;

score += 5;

if(ballposX + 19 <= brickRect.x || ballposX + 1 >= brickRect.x + brickRect.width){

ballXdir = -ballXdir;

}

else{

ballYdir = -ballYdir;

}

break A;

}

}

}

}

ballposX += ballXdir;

ballposY += ballYdir;

if(ballposX < 0){

ballXdir = -ballXdir;

}

if(ballposY < 0){

ballYdir = -ballYdir;

}

if(ballposX > 670){

ballXdir = -ballXdir;

}

}

repaint();

}

private void setFocusTraversalKeyEnabled(boolean b) {

//To change body of generated methods, choose Tools | Templates.

}

public void moveRight(){

play = true;

playX += 20;

}

public void moveLeft(){

play = true;

playX -= 20;

}

}

**Main\_Class**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.company.java\_project;

/\*\*

\*

\* @author Abdul Malik

\*/

import javax.imageio.ImageIO;

import java.awt.Color;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

public class Main {

public static void main(String[] args) {

JFrame jfrm = new JFrame();

PlayGame gameplay = new PlayGame();

jfrm.setBounds(10, 10, 700, 600);

jfrm.setTitle("Brick Breaker Game\_Java");

jfrm.setResizable(false);

jfrm.setVisible(true);

jfrm.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//Inserting gameplay Method in JFRAME

jfrm.add(gameplay);

//Game Logo

ImageIcon image = new ImageIcon("brick2.png");

jfrm.setIconImage(image.getImage());

}

}

# 5. Hardware/Software Requirement

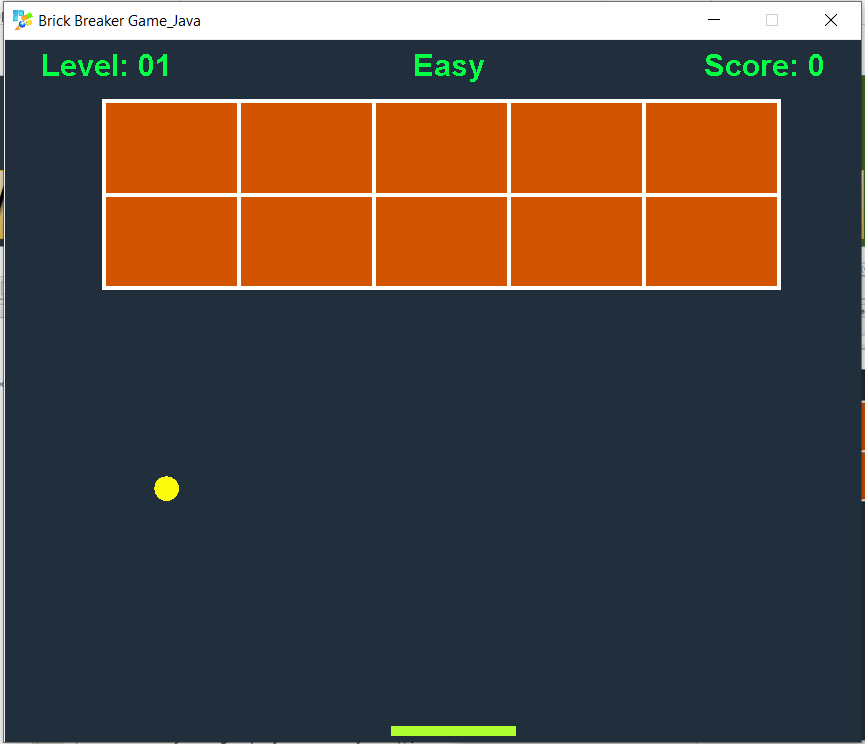
## 5.1 Topics

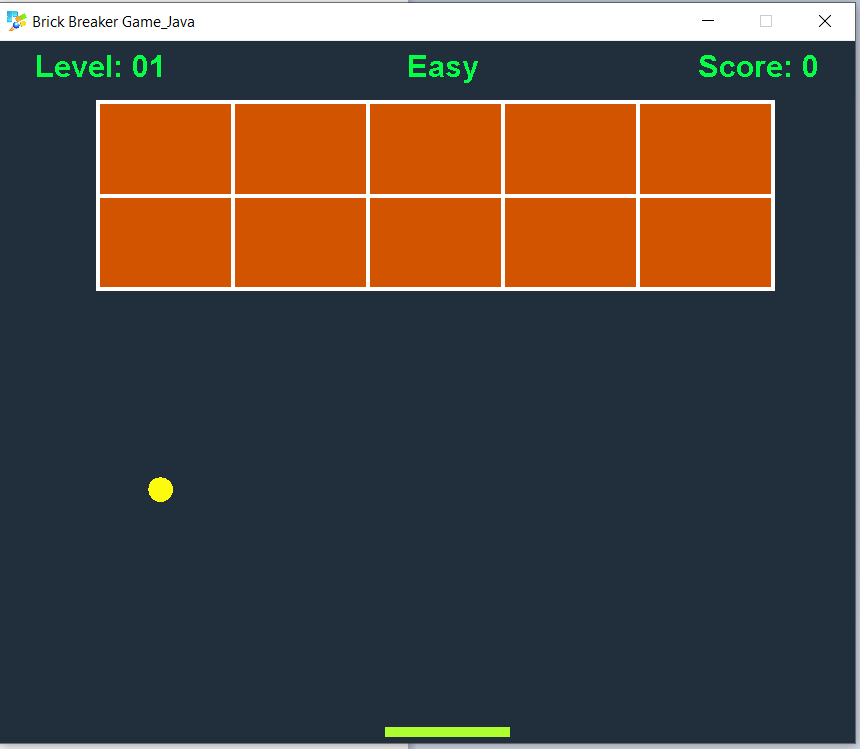
* + - **GUI**
    - **BUILT\_IN FUNCTIONS**
    - **ASSOCIATION**
    - **INHERITANCE**
    - **ENCAPSULATION**
    - **POLYMORPHISM**
    - **Method Overriding**
    - **Private, Instance and Public VARIABLES**

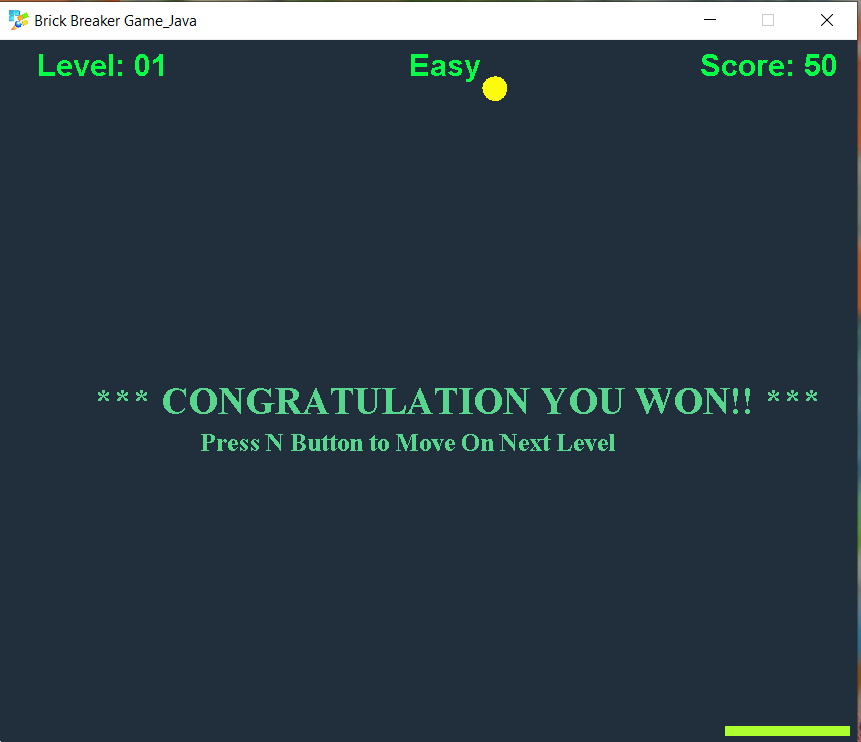
## 5.2 Imported Packages

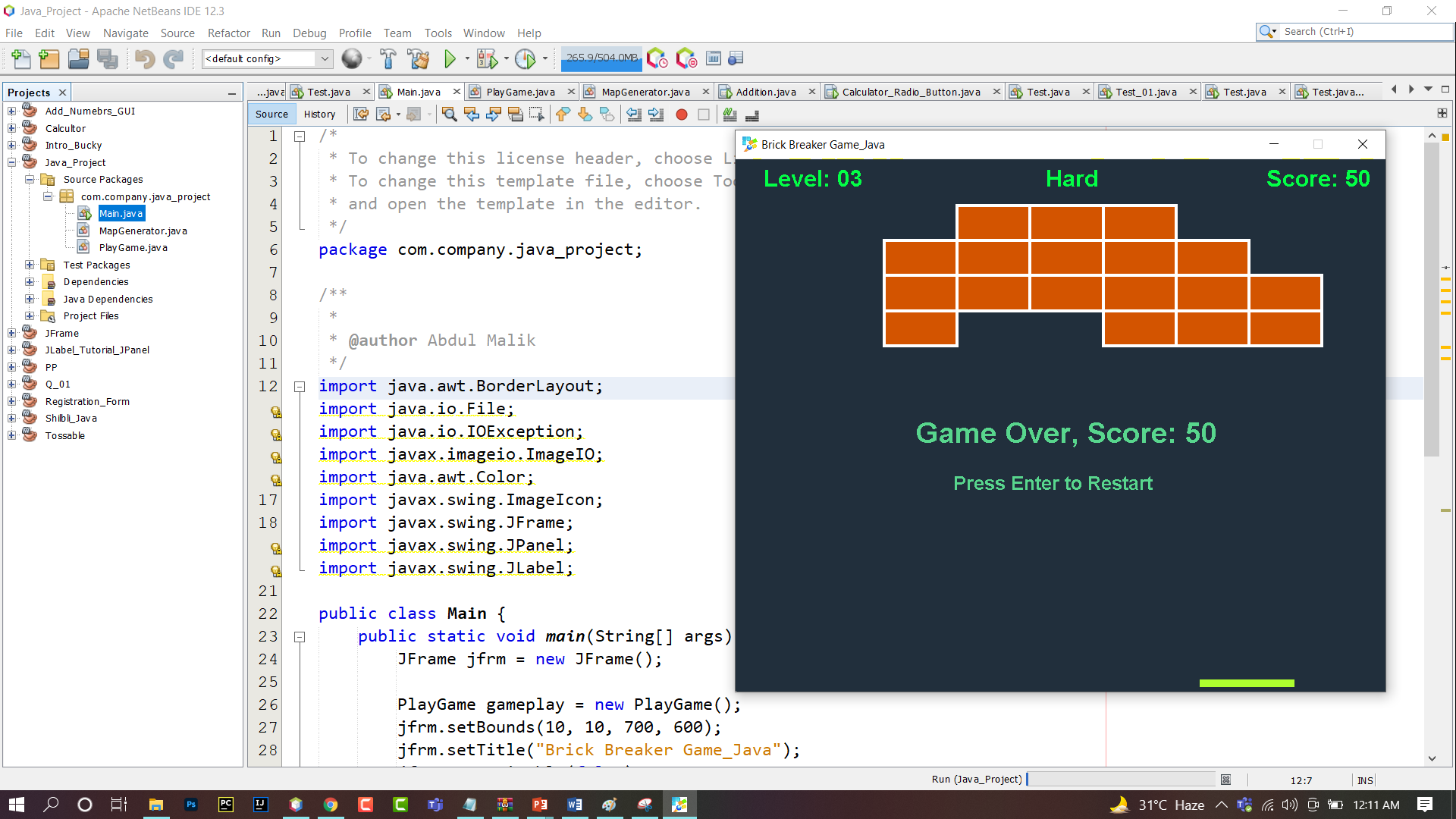
* + - **JFrame**
    - **JPanel**
    - **Color**
    - **Image Icon**
    - **Border Layout**
    - **Graphics and Graphics2D**
    - **Rectangle**
    - **Action Event**
    - **Action Listener**
    - **Key Event**
    - **Key Listener**
    - **Timer**
    - **Font**
    - **Basic Stroke**

# 6. Screen Shots



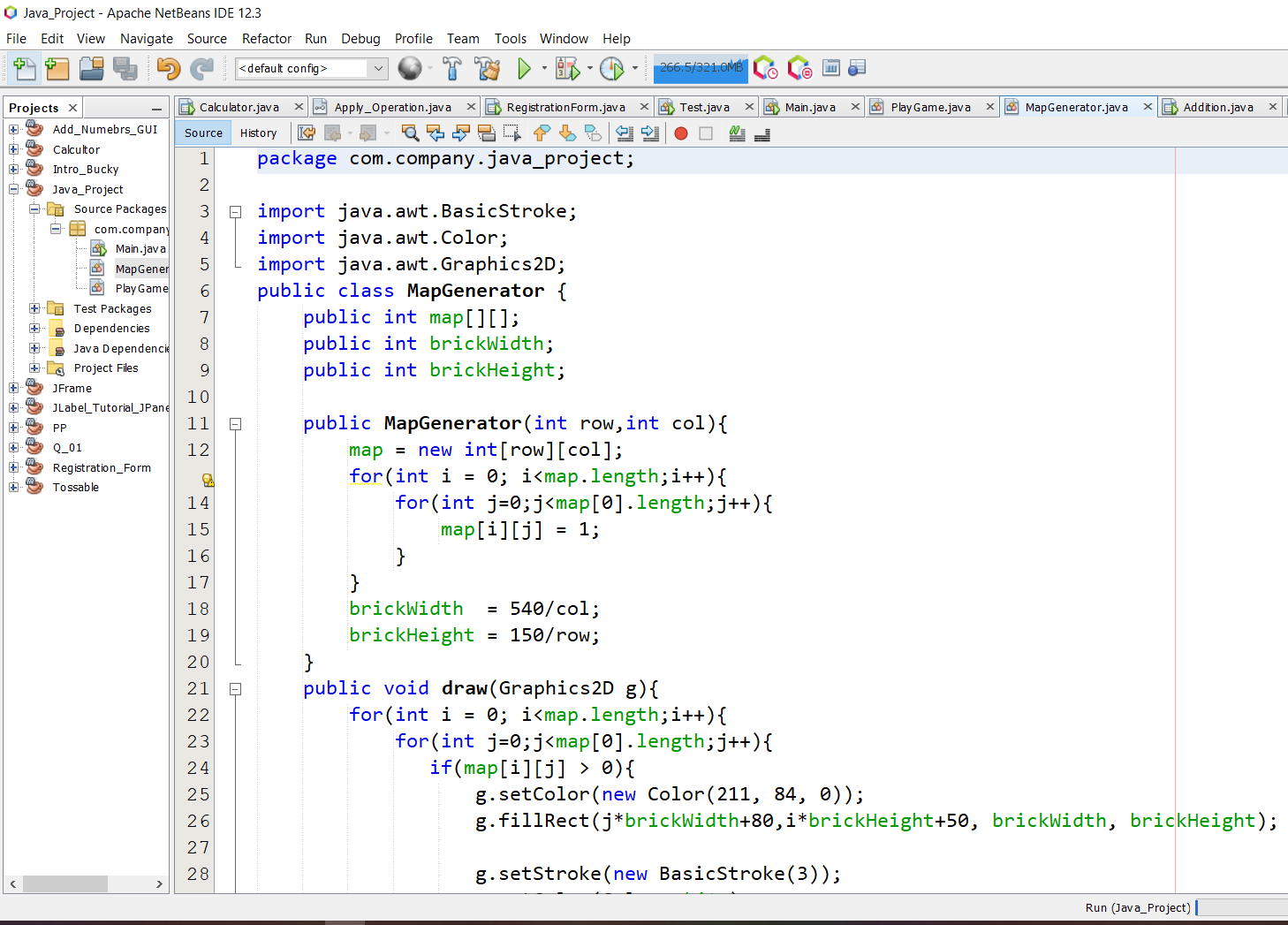


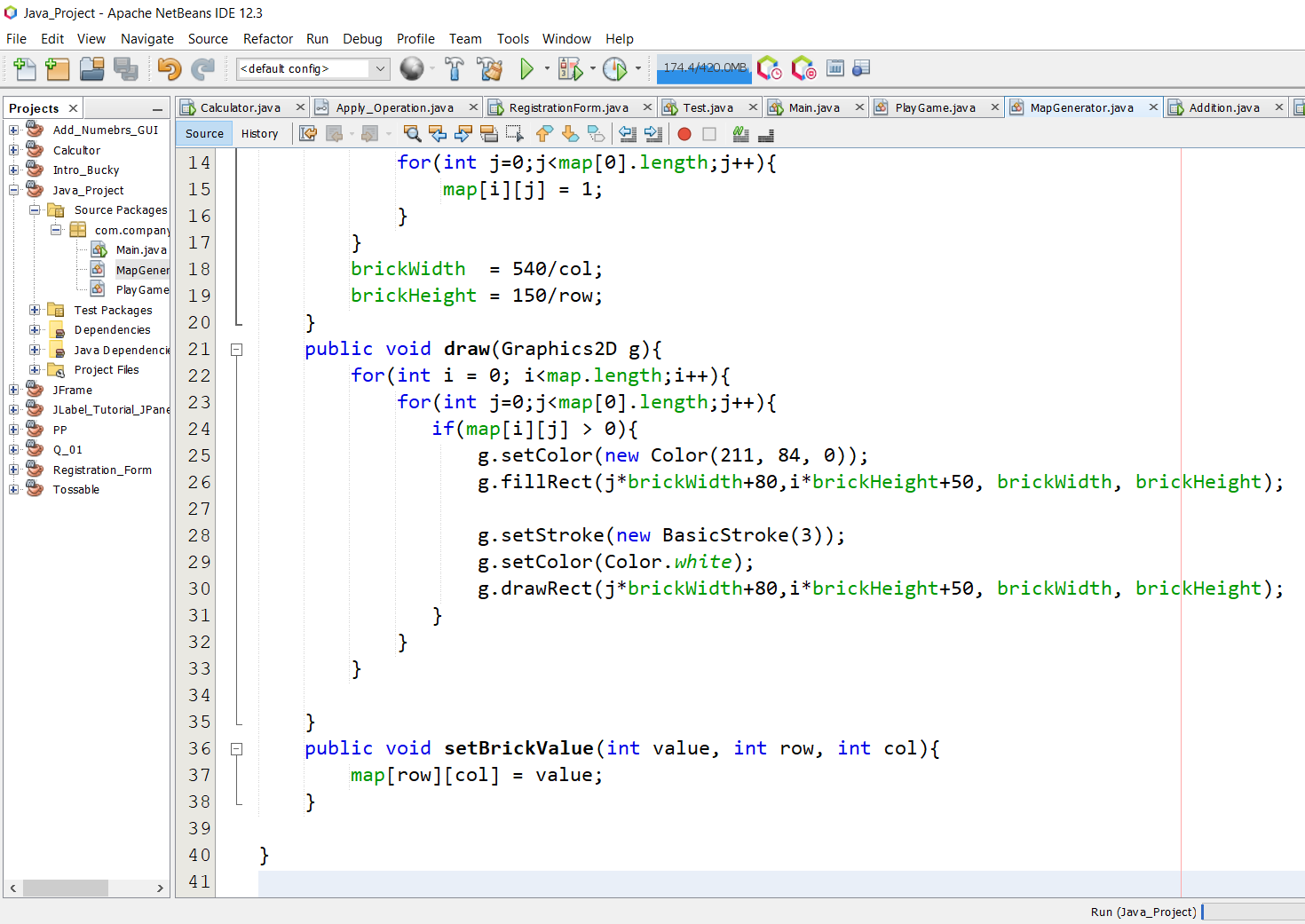




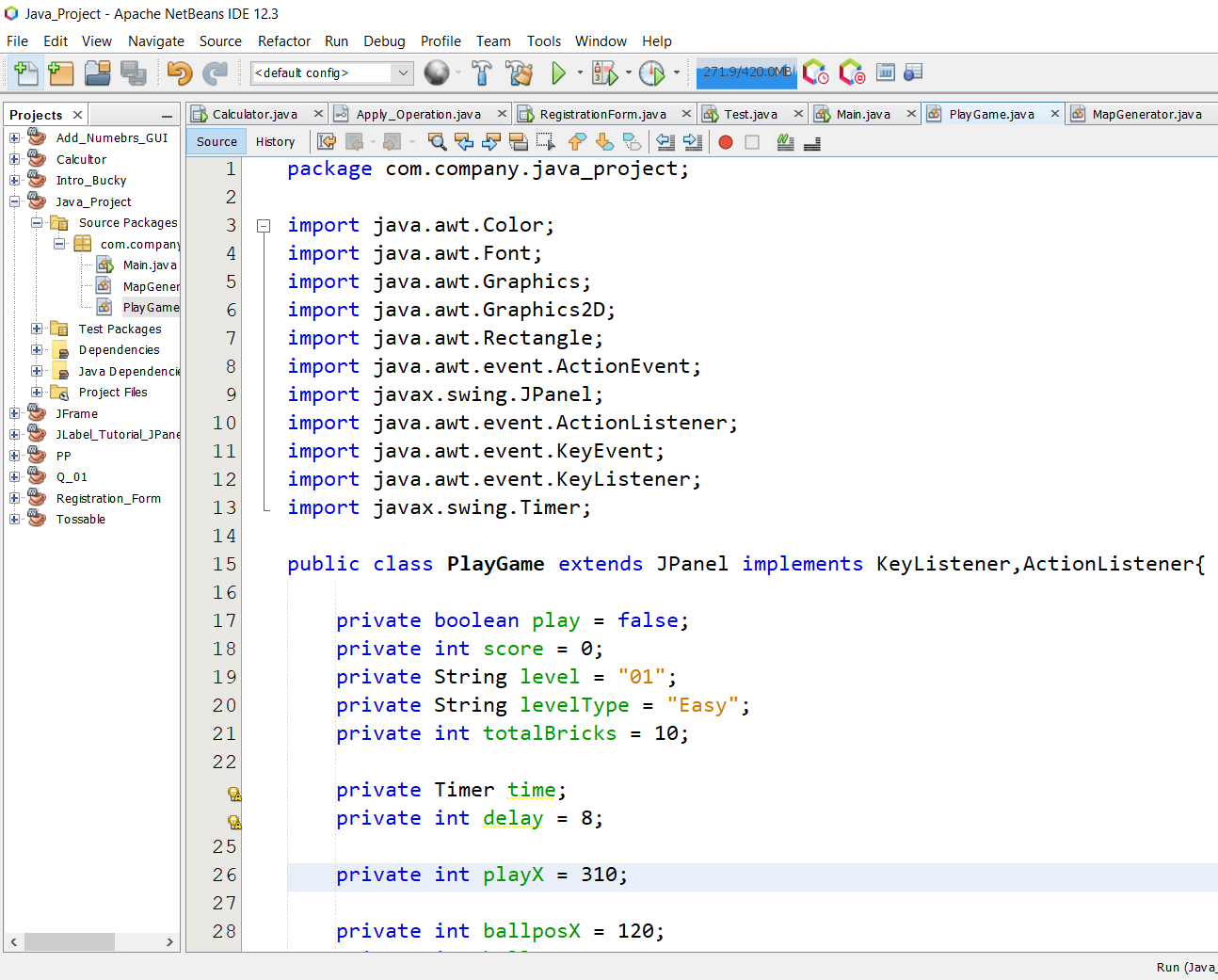
# Code Screen Shots:

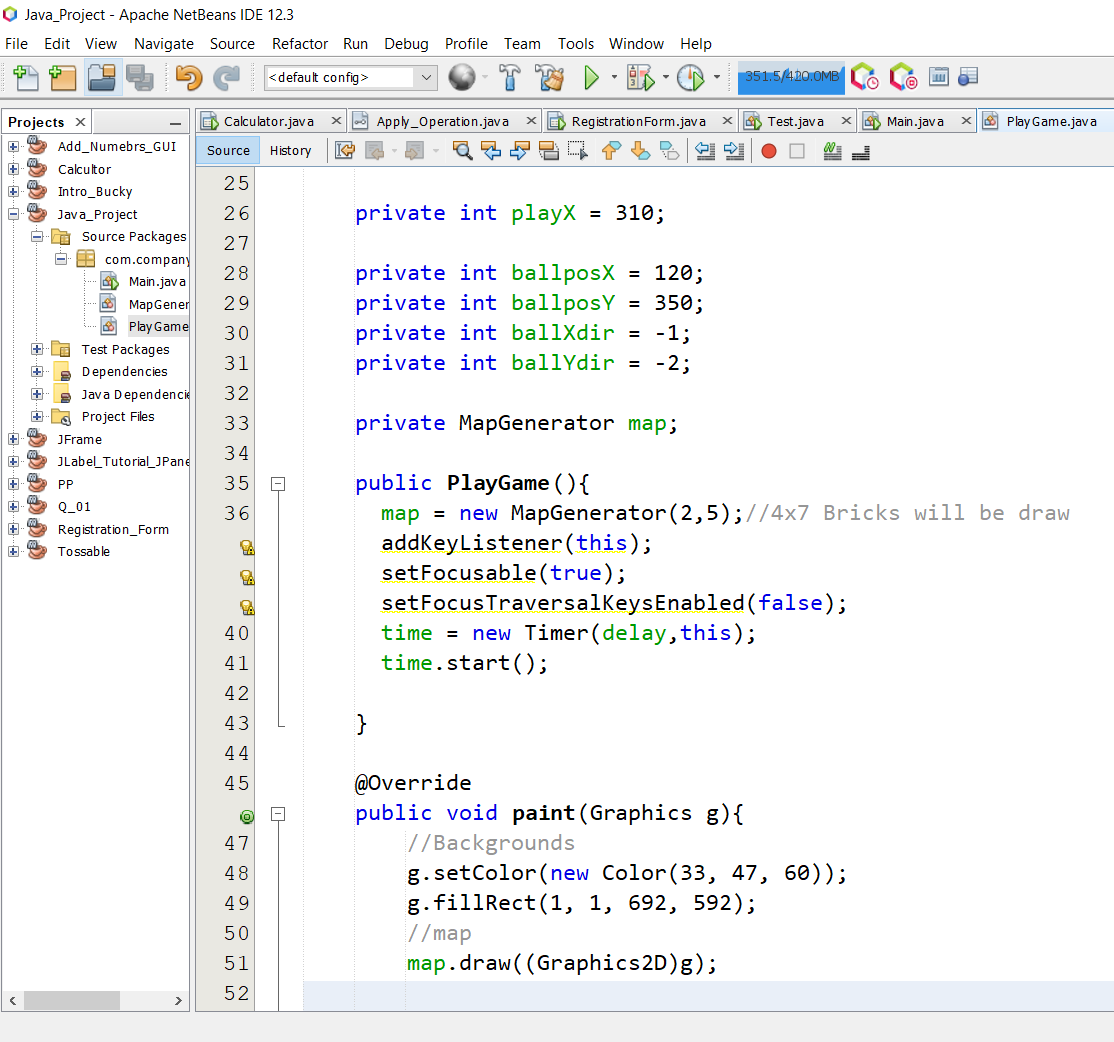
**Map Generator Class**



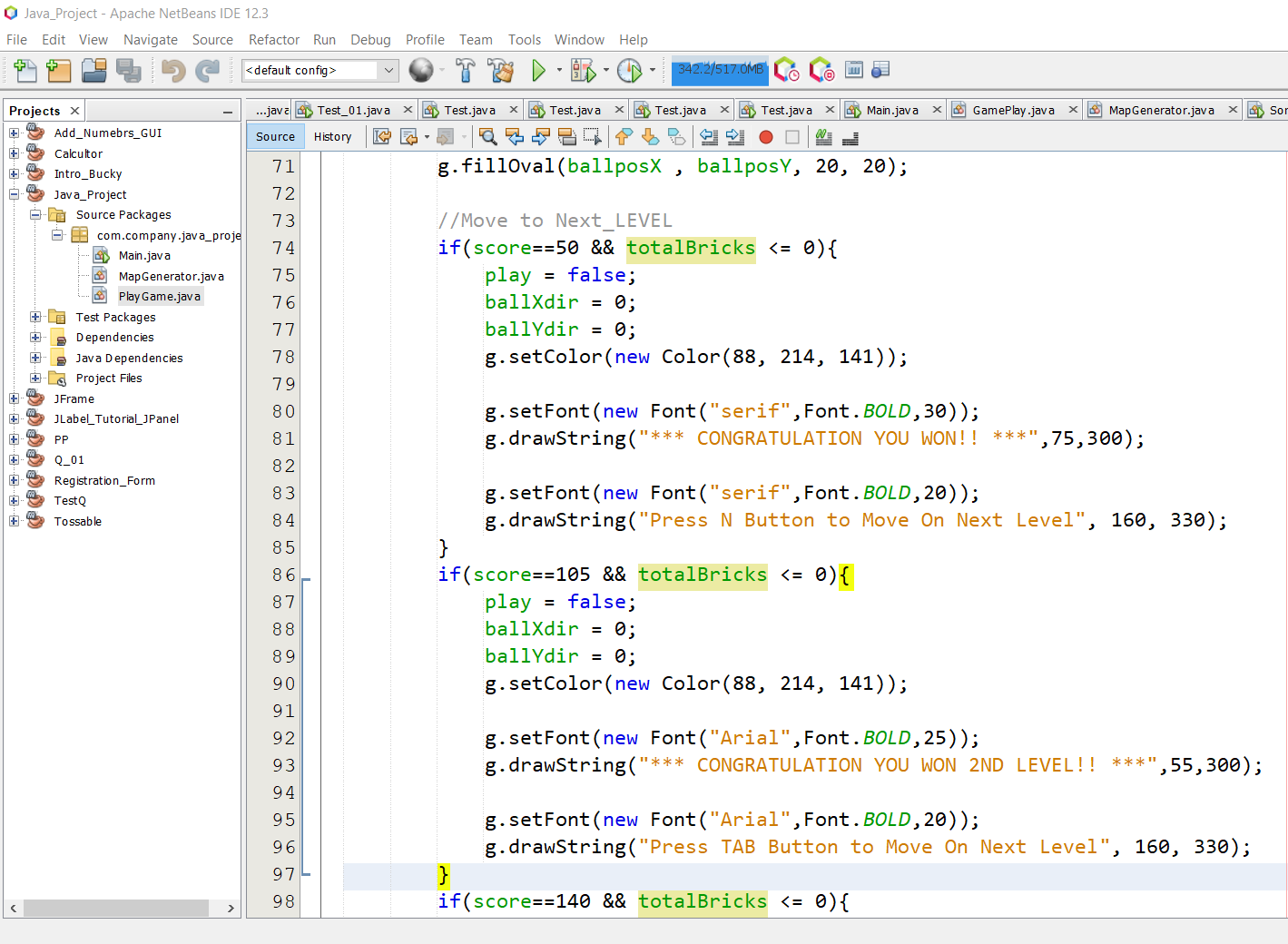


**PlayGame Class**

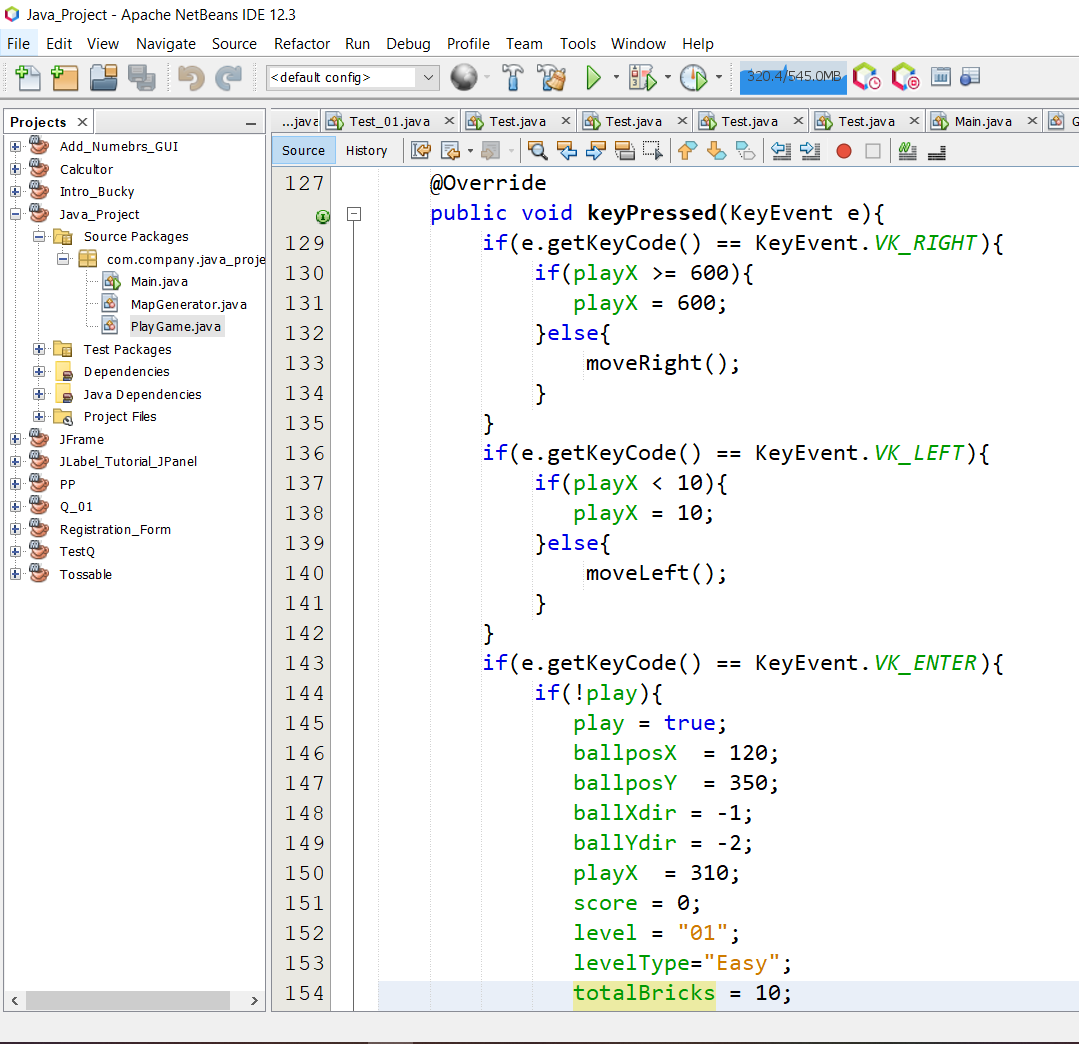


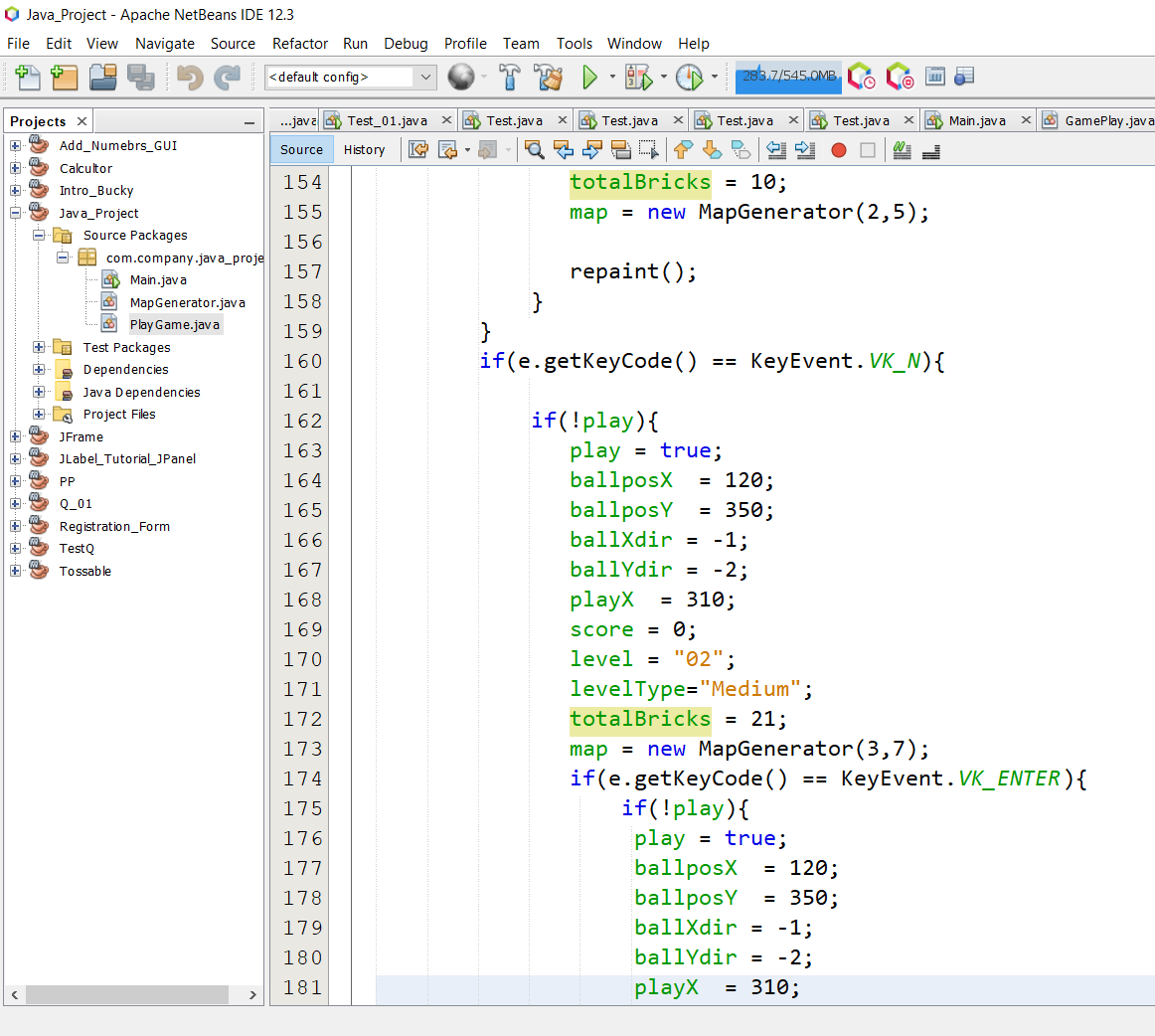


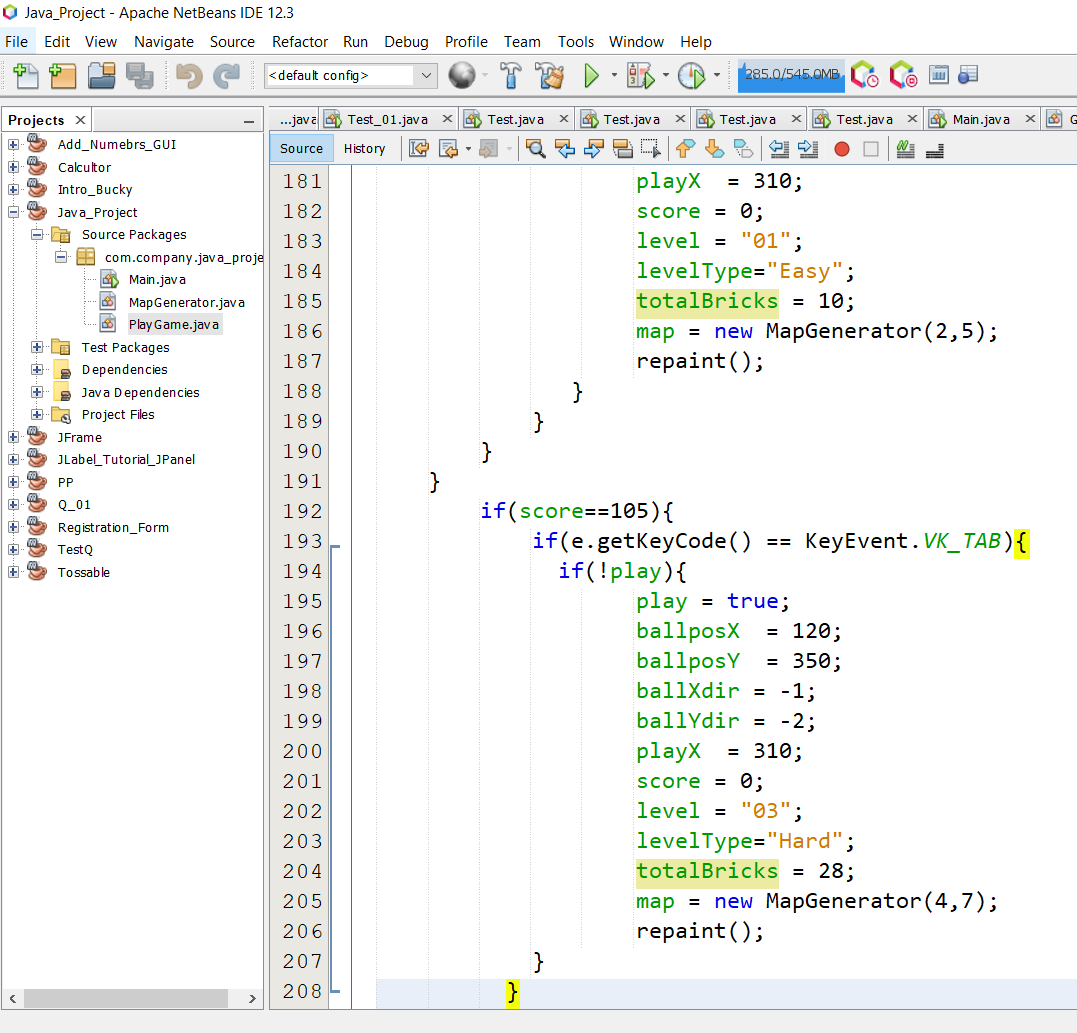
# 

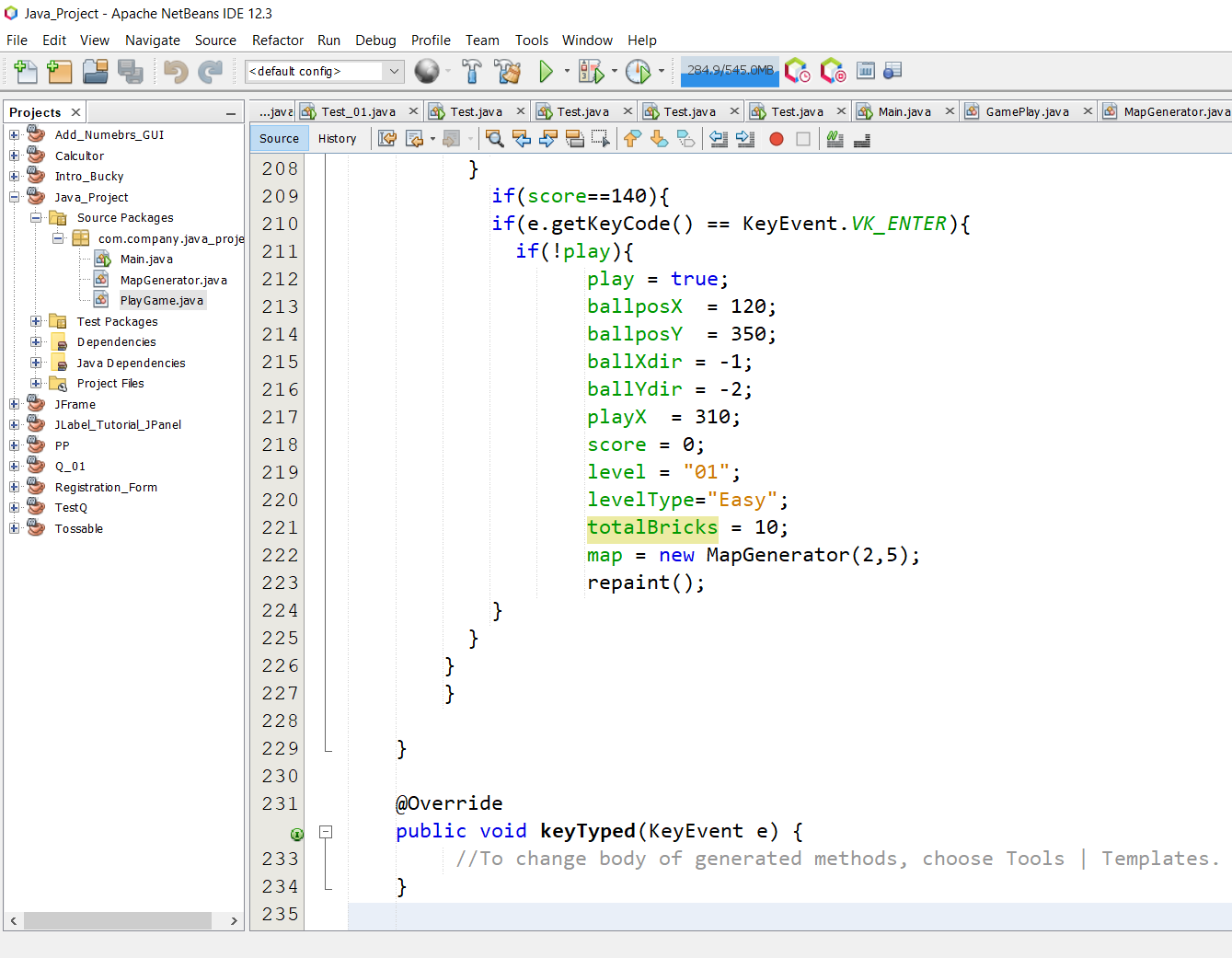


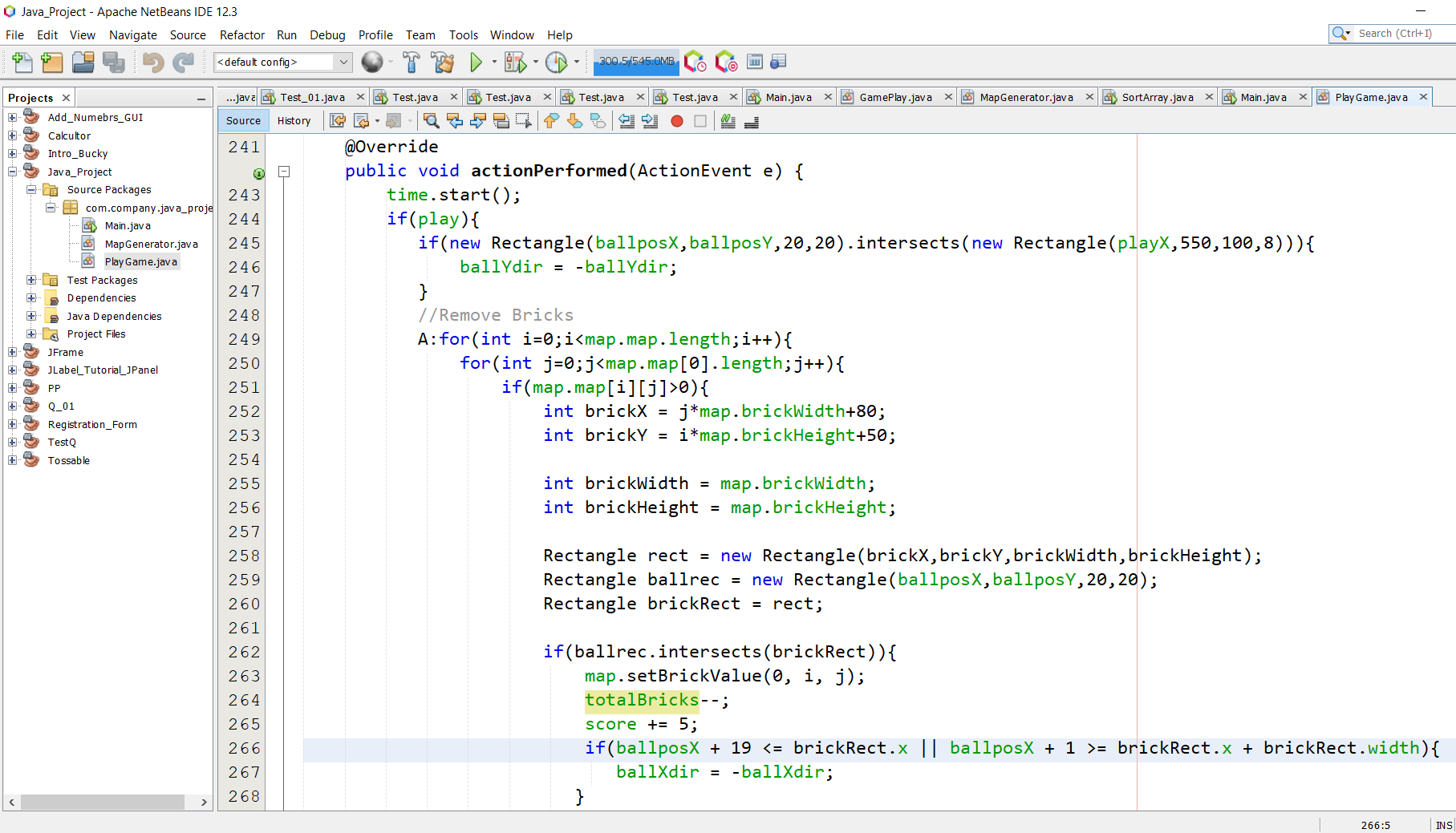
# 

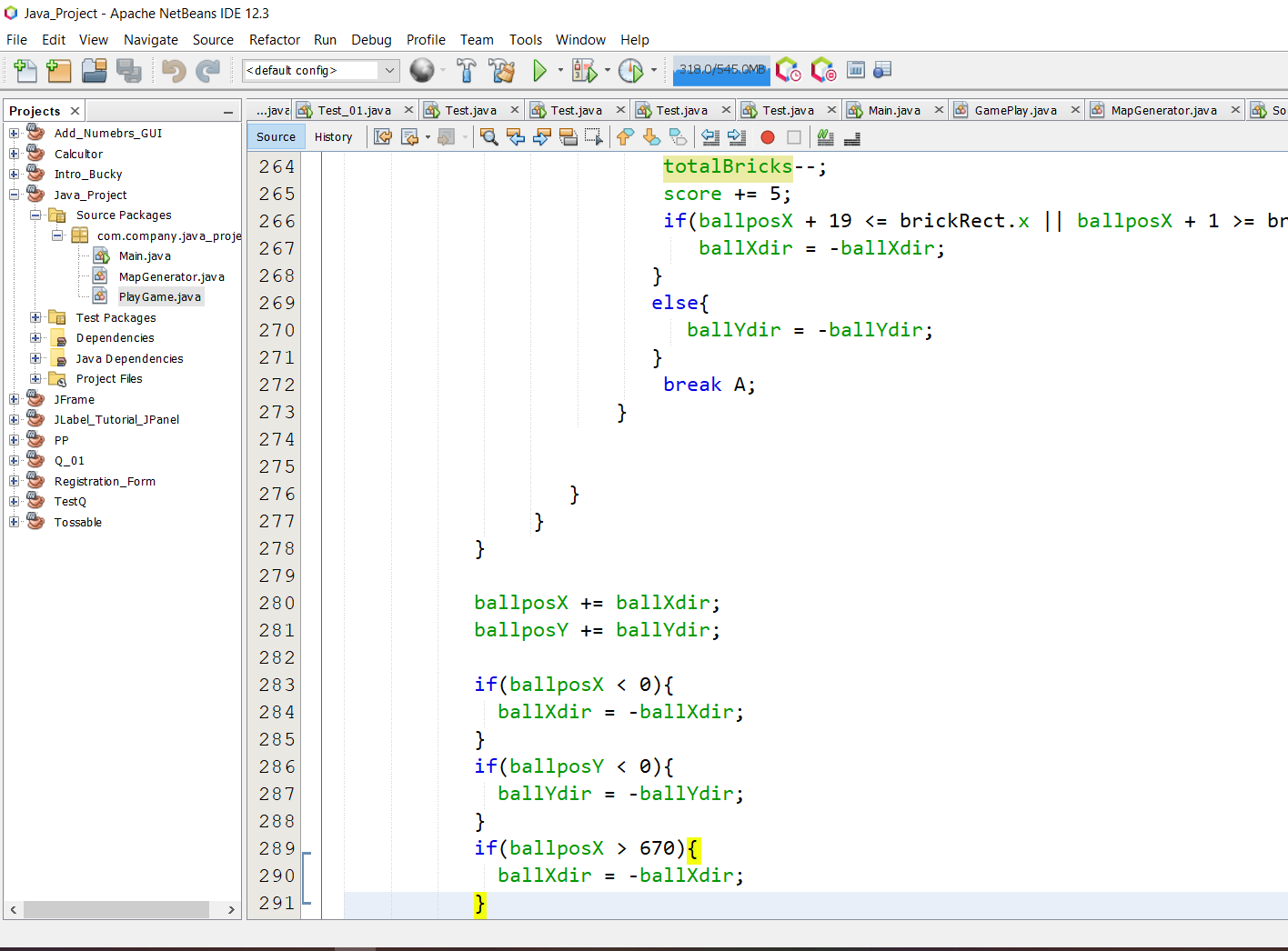


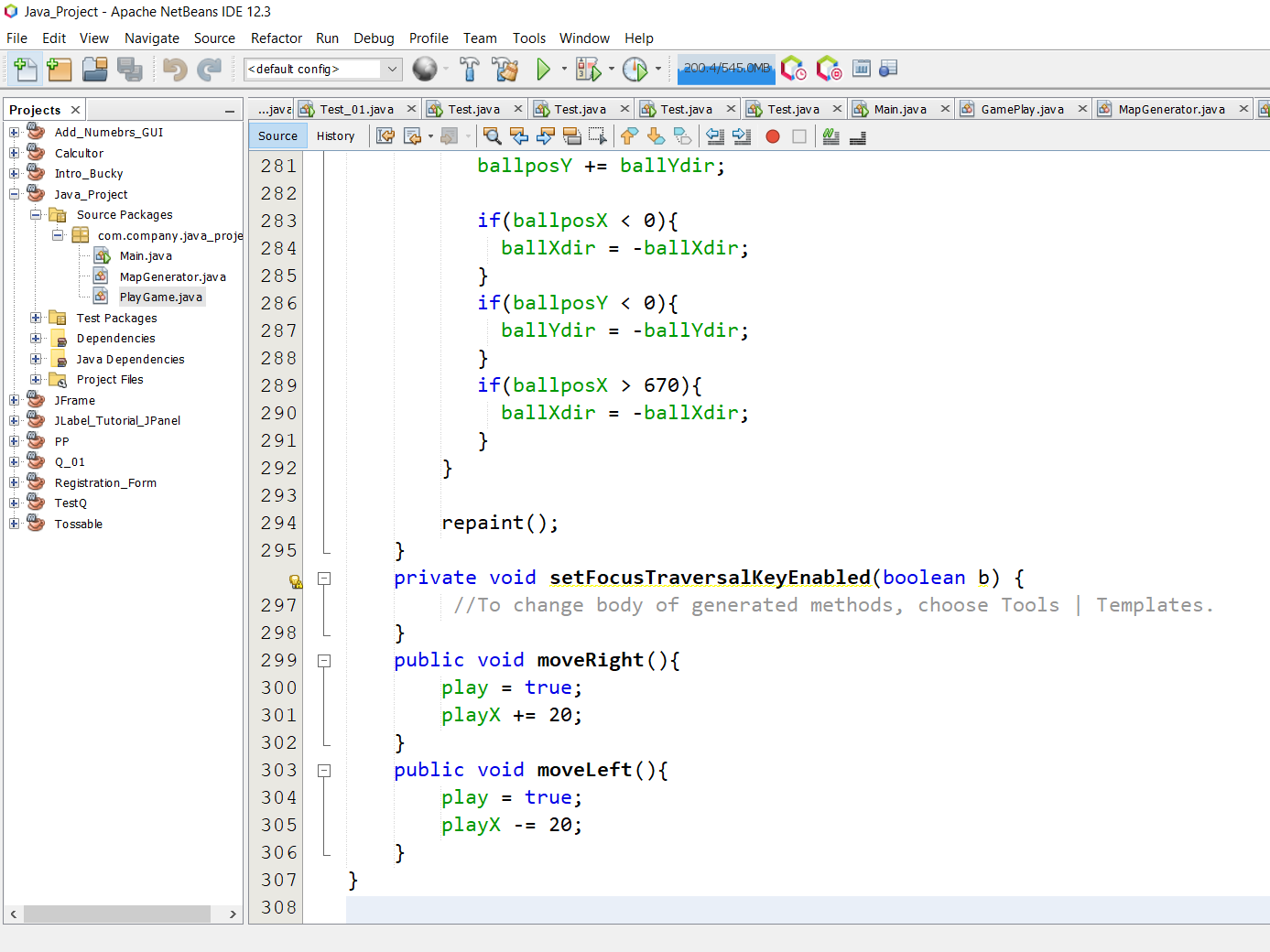




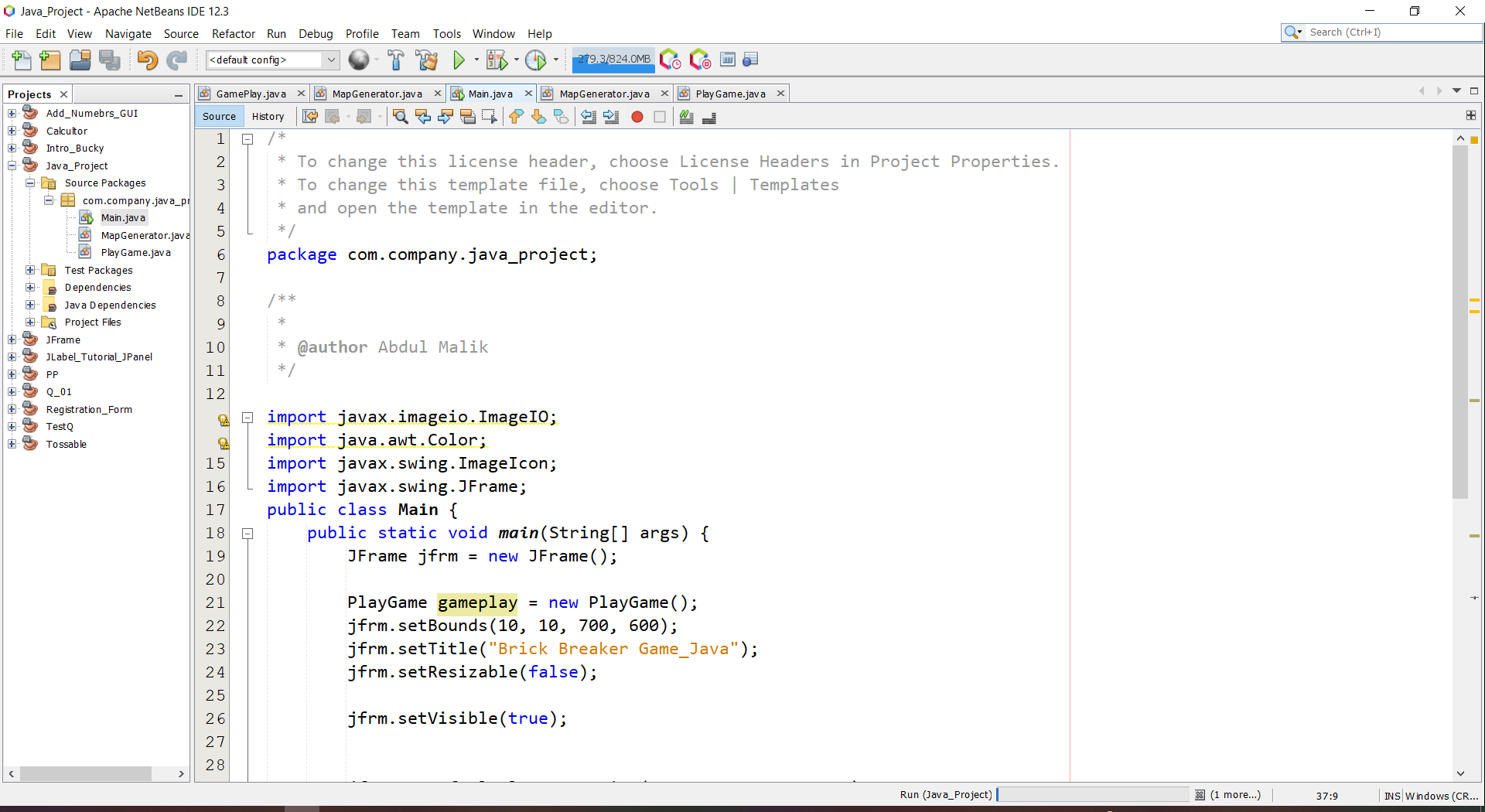


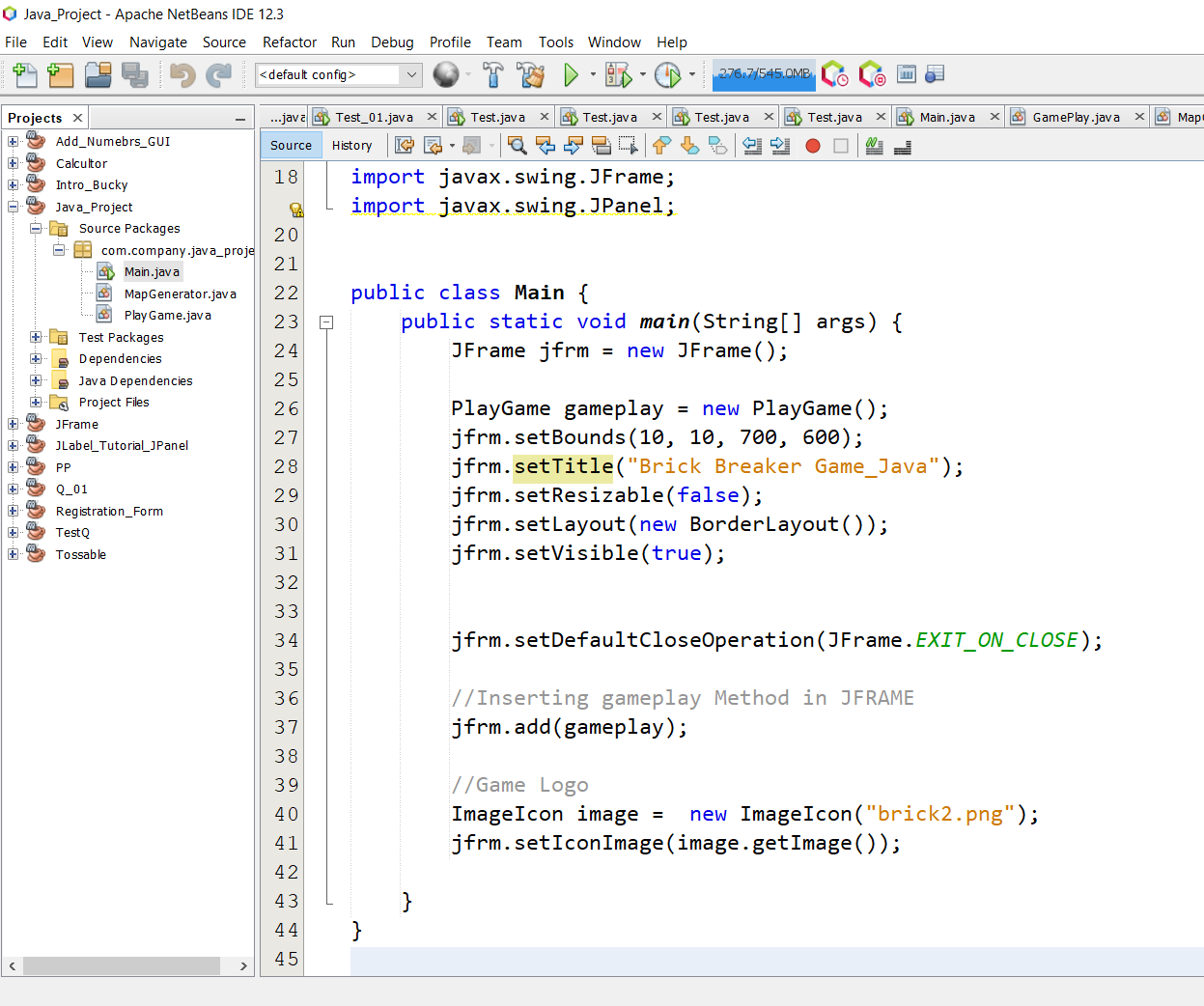






**Main Class**





# 8. Gantt Chart / Task Sheet

We have Done a work collectively on the Project

Abdul Waleed work was to study Rules and Laws of Game and Create a Main Frame window

Secondally Aftab work was to create Bricks using 2D Array and Insert Ball

After that Amir work was to insert Paddle and Movement of Paddle

Abdul Malik work was to set and Insert Levels(1 to 3) in a Game also to move paddle and to set colors of Game