QUESTION1:

```
PACKAGE CREATIONGAME
Import  javax.swing.Jframe
```

In this part, I import the javax. swing and I create the package where we can see all the attributes or objects that my came to need.

So for the first, we create the Main class

Main.java:

```
Public class Main{
//where we create the interface Game .
public static void main { jFrame obj = new J=jFrame

obj.setBounds(this functions for the game dimension)
obj.setTitle(this function is for the title)
obj.setVisible(this function is to poster the interface game)
obj.setResizable( prevents the game from being played on this farm )
obj.setDeafultcloseoperation(this one is to maintain the game on )
 }
}
```

This class is just necessary for the graphic face

For the next steps I create the the class Gameplay this class i necessary because it contain all the objects and the functions actions that the players need .

Gameplay.java:

```
PACKAGE CREATIONGAME
Import  java.awt.Graphics;
Import  java.awt.Graphics2D;
Import  java.awt.Color;
Import  java.awt.Font;
Import  java.awt.event.*;
Import  javax.awt.Rectangle;
Import  javax.swing.jPanel;
```

//we creat the inheritance because this project we just have one inheritance .This class suppose to have all the action that the players need . for example move the rectangle hit the ball and win points .We will use more the Bolean in this part .

Public class Gameplay extends Jpanel implements keyListerne ,Action Listener{

Public gameplay() {
addKeylistnenr
setFocusable
setFocusTraversalKeysEnabled
 time = new Timer(delay,this);}
Public void paint (graphics g){
//Background
g.setColor
g.fillRect

//Borders
g.setColor()
g.fillRect()
g.fillRect()
g.fillRect()

//scores
g.setColor()
g.font()
g.drawString()

//drawing
g.setColor()
g.fillRec()

//ball
g.setColor()
g.fillRec()

//padlle
g.setColor()
g.fillRe()

If (**totalbricks <-0**){
}}

```
        play=false;
        ballXdir=0;
        ballYdir=0;
```

```
        g.setColor(Color.red);
        g.setFont(new Font("serif",Font.BOLD,30));
        g.drawString("you won ", 260, 300);

        g.setFont(new Font("serif",Font.BOLD,20));
```
g.drawString("Press Enter to Restart", 230, 350);}

If (totalbricks<=0){

```
        play=false;
        ballXdir=0;
        ballYdir=0;
        g.setColor(Color.red);
        g.setFont(new Font("serif",Font.BOLD,30));
        g.drawString("you won ", 260, 300);

        g.setFont(new Font("serif",Font.BOLD,20));
        g.drawString("Press Enter to Restart", 230, 350);
```
}

If (ball pos)

```
    }

{
        play=false;
        ballXdir=0;
        ballYdir=0;
        g.setColor(Color.red);
        g.setFont(new Font("serif",Font.BOLD,30));
        g.drawString("Game over,score:", 190, 300);
        g.setFont(new Font("serif",Font.BOLD,20));
        g.drawString("Press Enter to Restart", 230, 350);
    }

    g.dispose();
```

```
    }

//This part has the object of my game .score, ball, the bricks ...
//so I add all the condition is necessary for my program
//



 // this public action is for the different movement that we can have .for
example we make the position bricks on the game or we can just do the
different movement of our ball and so that the ball does not disappear
during the game

    public void actionPerformed(ActionEvent e) {
        time.start();
        if(play){
            if (new Rectangle(ballposX,ballposY,20,20).intersects(new
Rectangle(playerX,550,100,8))){
                ballYdir =- ballposY;
            }
            A:for(int i = 0 ; i<map.map.length;i++){
                for (int j = 0 ; j<map.map.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 ballRect = new
Rectangle(ballposX,ballposY,20,20);
                        Rectangle brickRect = rect ;

                        if (ballRect.intersects(brickRect)){
                            map.setBrickvalue(0, i, j);
                            totalBricks--;
                            score +=5;
```

```
                            if(ballposX + 19 <= brickRect.x || ballposX +
1 >= brickRect.x + brickRect.width){
                                ballXdir = - ballYdir;
                            }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();
    }

  //theese(keyped and keyreleased) function was generate automatically
    public void keyTyped(KeyEvent e) {}

    public void keyReleased(KeyEvent e) {}
// The function Keypressed is basically to connect the keyboard with the
game thas is the really action of the this functions
    public void keyPressed(KeyEvent e) {if(e.getKeyCode()==
KeyEvent.VK_RIGHT){
```

```
        if (playerX>=600){
            playerX=600;
        }
        else{
            moveRight();
        }
// we call all the module which necessary to connect keyboard and game
like for example  we create the argument that we call e that we are
supposed to use and them we connect with the keyboard function that we
call Keycode
    }
    if(e.getKeyCode()== KeyEvent.VK_LEFT){
        if (playerX>=10){
            playerX=10;
        }
        else{
            moveLeft();
        }

// If the play is true (boolean constate ) like if I use my keyboard and
then I move right or left, the ball starts to move, and then we can start
to punch the ball
    }
    if (e.getKeyCode() == KeyEvent.VK_ENTER){
        if (play){
            play=true;
            ballposX            ballposY = 350 ;
            ballXdir             ballYdir = - 2;
            playerX            score    = 0;
            totalBricks
            map = new Generetor (3, 7);

            repaint();
        }
    }
}


public void moveRight()
{
```

```
play=true;
playerX+=20;
}

public void moveLeft()
{
play=true;
playerX-=20;
}}
```

```
/// on the first time I create the package (the folder)that we can put all
objects that we need and then we do like the first time we import the
color the, 2D graphics and the basic stroke(necessary for the reality
object or background)


package GAMECREATION;

import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.BasicStroke;

public class Generetor {

    public int map[][];(this function is a map like a reference )
    public int brick width;(this one is the width of the brick on the game
that we make the destruction)
    public int brickHeight;(this one is for the height of the brick in the
game )
//this method is to generate all the methods that I put.
    public Generetor (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 =1200/col; //this one is the width
        brickHeight =200 /row;//this one the height
    }

//this Method is for the graphics in 2D dimension so for this, we create
the method that we call a draw and we call the Grapcis 2D, and then we use
the loop for the different dimensions of our 2D.
    public void draw(Graphics2D g){
        for(int i=0; i <map.length;i ++){
            for(int j = 0 ; j <map.length;j++){
                if(map[i][j]>0){
                    g.setColor(Color.white);
                    g.fillRect(j*brickWidth+80,
i*brickHeight+50,brickWidth,brickHeight);

                    g.setStroke(new BasicStroke(8));
                    g.setColor(Color.black);(this one is for the
background of our game)
                    g.drawRect(j*brickWidth+80,
i*brickHeight+50,brickWidth,brickHeight);
                }
            }
        }
    }
    public void setBrickvalue(int value ,int row , int col ){
        map [row][col] = value ;
        //this one is for the different brick values .like when you break.
    }

}
```

QUESTION 2:

1) My java project has just on inheritance because :
    - I don’t need more than 2
    - My inheritance suppose to have many methods that I need of my project in the simple word (is an inheritance of my class Main so)

2)API JAVA:

An API makes data or functionality from an existing application available for other applications to use. This should make the notion of an application programming interface clearer.
Using an API, therefore, allows you to use an existing program rather than re-developing it. So it's a big time saver in the end.
The first thing an API does is expose, that is, it makes functionality or data available. To use them, most APIs require an API key, or sometimes two. This key allows the API to identify you as a user with the necessary rights to use the API.
The full form of API is the application programming interface. This is a document that gives you a list of all the packages, classes, and interfaces, along with their fields and methods.
By using these APIs, the programmer can know how to use the methods, fields, classes, interfaces provided by the Java libraries.