## INDEX

|  |  |  |
| --- | --- | --- |
| S.No | Contents | Page No. |
| 1. | Bonafide Certificate | 2 |
| 2. | Acknowledgement | 3 |
| 3. | Synopsis | 4 |
| 4. | Login Form | 6 |
| 5. | Sign Up Form | 9 |
| 6. | Home Page | 11 |
| 7. | Air Hockey | 13 |
| 8. | Bird Hunt | 33 |
| 9. | Flappy Bird | 44 |
| 10. | Rating Form | 55 |
| 11. | Closing Form | 57 |
| 12. | Back End - Database | 58 |
| 13. | Form Sequence | 63 |

**Bonafide**

Certified to the Bonafide Report of work done by Master Vignesh K of Class XII in Sri Sankara Senior Secondary School, Adyar, Chennai- 600020

During the year 2016-2017

Dated . Subject Teacher

Submitted for All-India Higher Secondary Practical Examination held in Informatics Practices at Sri Sankara Senior Secondary School, Adyar, Chennai-20

External Examiner

Dated . Seal

**Acknowledgement**

We would like to express our gratitude and deep regards to our teachers Mrs. Revathy and Mrs. Vidyalakshmi for their guidance and encouragement given throughout the course of this project entitled Vibrating Cantilever.

We would also like to thank our laboratory assistant Mr. Jose for his help in completion of our project.

We would also like to thank to our Principal Mrs. Mita Venkatesh and the staff members of Sri Sankara Senior Secondary School, for the apparatus and valuable information provided by them.

We are obliged to our parents and our classmates in helping us to finalize this project report.

**Synopsis**

This project in Informatics practices involves game developing. The development of games for commercial purpose is a branch under e-business.

The front end application is programmed using Netbeans IDE 8.0.2 and the back end database used is MySQL. This project includes three games each including a GUI. The three games are:

1. Air Hockey
2. Bird Hunt
3. Flappy Bird

All the three games use GUI components available in netbeans like Buttons, Labels and Radio buttons. The air hockey game uses 2D graphics of ‘Graphics’ class in java. The other two games use GUI components and their methods. Apart from these the project features other JFrame classes too.

All the three games require almost similar kind of skill from the user. Different modes of game play like difficulty level are available for the user.

To access these games one has to create an account and login using the account. The sign up page asks for a username and password and a captcha. After creating an account one can enter their details and login. Once logged on one can play any of the games. When playing for the first time the user will be asked to rate the application before quitting.

The back end database contains tables for the user’s login details, scores in the games bird hunt and flappy bird. Using Java database connectivity the details of the signed up users are stored. It is ensured that the details entered by the user are valid by retrieving data from the tables.

The high score for every user in each game is stored and the user can view the leader board for every game. The ratings given by the users are accepted and the average rating of the application is displayed to the user.

Referential integrity is maintained using foreign key constraint in tables containing data about high scores. The application has been tested and is found to be working properly. Some more possible improvements can be done in each game.

**Login Form**

public static String user;

public static int rating;

public static String avgrating;

**private void formWindowOpened(java.awt.event.WindowEvent evt)** {

jLabel4.setVisible(false);

jLabel3.setVisible(false);

jTextField1.setVisible(false);

jPasswordField1.setVisible(false);

jButton3.setVisible(false);

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

jLabel4.setVisible(true);

jLabel3.setVisible(true);

jTextField1.setVisible(true);

jPasswordField1.setVisible(true);

jButton3.setVisible(true);

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)** {

int exist = 1, match = 1;

user = jTextField1.getText();

String password=new String(jPasswordField1.getPassword());

if((user.equals(""))&&(password.equals("")))

JOptionPane.showMessageDialog(this,"Please enter username and password");

else if(user.equals(""))

JOptionPane.showMessageDialog(this,"Username has not been entered");

else if(password.equals(""))

JOptionPane.showMessageDialog(this,"Password field is empty");

else{

try{

Class.forName("java.sql.Driver");

Connection con = DriverManager.getConnection(“jdbc:mysql://localhost:/java

project”,”root”,”mysql”);

Statement st = con.createStatement();

String check = "select \* from userdata;";

String compute= "select round(avg(rating),1) from userdata where rating!=-1;”;

ResultSet rs = st.executeQuery(check);

while(rs.next()){

if(user.equals(rs.getString(1))){

exist=0;

if(password.equals(rs.getString(2))){

match=0;

rating = rs.getInt(3);

}

}

}

rs = st.executeQuery(compute);

rs.next();

avgrating = rs.getString(1);

if((exist==0)&&(match==0)){

this.setVisible(false);

new IntroForm().setVisible(true);

}

else if(exist==1)

JOptionPane.showMessageDialog(this,"Enter valid username");

else

JOptionPane.showMessageDialog(this,"Password does not match");

}

catch(Exception e){

System.out.print(e.getMessage());

}

}

}

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new Signupform().setVisible(true);

}

**Sign Up Form**



**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

int exist = 1;

String name = jTextField1.getText();

String password = new String(jPasswordField1.getPassword());

String conpassword = new String(jPasswordField2.getPassword());

String captcha =jTextField4.getText();

if(password.compareTo(conpassword)!=0)

JOptionPane.showMessageDialog(this,"Please enter the same password in both the password fields");

else if(name.equals(""))

JOptionPane.showMessageDialog(this,"Name has not been entered");

else if(password.equals(""))

JOptionPane.showMessageDialog(this,"Password field is empty");

else if(captcha.equals(""))

JOptionPane.showMessageDialog(this,"Captcha field is empty");

else if(captcha.compareTo("duBX8B")!=0)

JOptionPane.showMessageDialog(this,"The captcha has been entered wrongly");

else{

try{

Class.forName("java.sql.Driver");

Connection con = DriverManager.getConnection(“jdbc:mysql://localhost:/java

project”,”root”,”mysql”);

Statement st = con.createStatement();

String check = "select \* from userdata;";

String include = "insert into userdata values('"+name+"','"+password+"',-1);";

ResultSet rs = st.executeQuery(check);

while(rs.next()){

if(name.equals(rs.getString(1))){

exist=0;

JOptionPane.showMessageDialog(this,"This username already exists");

break;

}

}

if(exist==1){

st.executeUpdate(include);

st.executeUpdate("insert into birdhunt values('"+name+"',0,0,0);");

st.executeUpdate("insert into flappybird values('"+name+"',0);");

this.setVisible(false);

new loginform().setVisible(true);

}

}

catch(Exception e){

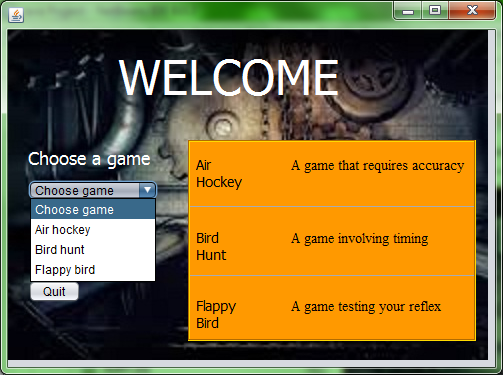
System.out.print(e.getMessage());

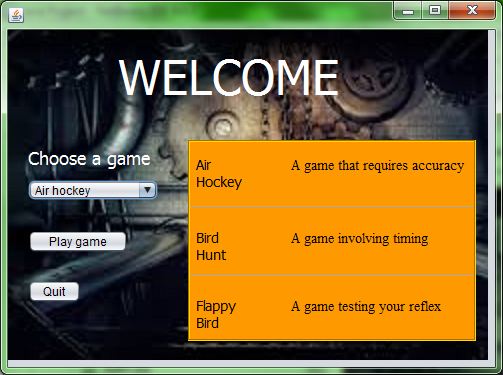
}

}

}

**Home Page**





int a;

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

int a =cb.getSelectedIndex();

if(a==0)

JOptionPane.showMessageDialog(this,"Please choose a game");

if(a==1){

this.setVisible(false);

new Airhockey().setVisible(true);

}

if(a==2){

this.setVisible(false);

new Shootthekuruvi().setVisible(true);

}

if(a==3){

this.setVisible(false);

new flappybird().setVisible(true);

}

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)** {

if(loginform.rating==-1){

a = JOptionPane.showConfirmDialog(this,"Please rate this app");

if(a==0){

this.setVisible(false);

new rating().setVisible(true);

}

}

else{

this.setVisible(false);

new Endframe().setVisible(true);

}

**Air Hockey**

Introduction Form:



static int i;//Color reference

void chooseColor(){

this.setVisible(false);

new Colorchooser().setVisible(true);

}

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

i = 1;

chooseColor();

}

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)** {

i = 2;

chooseColor();

}

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)** {

i = 3;

chooseColor();

}

**private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)** {

i = 4;

chooseColor();

}

static int diffselection = 1;

static int modeselection = 1;

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

if(easy.isSelected())

diffselection = 1;

if(hard.isSelected())

diffselection = 2;

if(classic.isSelected())

modeselection = 1;

if(timed.isSelected())

modeselection = 2;

if(Colorchooser.bgColor==Color.black)

JOptionPane.showMessageDialog(this,"You can't choose black color for the background");

else if(Colorchooser.bgColor==Colorchooser.myMalletColor)

JOptionPane.showMessageDialog(this,"You have chosen the same colour for your mallet and background");

else if(Colorchooser.bgColor==Colorchooser.oppMalletColor)

JOptionPane.showMessageDialog(this,"You have chosen the same colour for opponent's mallet and background");

else if(Colorchooser.bgColor==Colorchooser.puckColor)

JOptionPane.showMessageDialog(this,"You have chosen the same colour for the puck and background");

else{

this.setVisible(false);

new sop().setVisible(true);

}

}

**private void formWindowOpened(java.awt.event.WindowEvent evt)** {

easy.setSelected(true);

classic.setSelected(true);

}

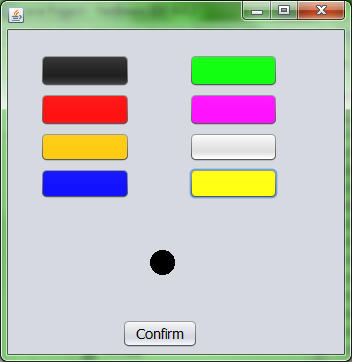
**private void jButton6ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new IntroForm().setVisible(true);

}

Color Choosing Form



//Set default colors

static Color color = Color.lightGray;

static Color myMalletColor = Color.red;

static Color oppMalletColor = Color.red;

static Color puckColor = Color.blue;

static Color bgColor = Color.lightGray;

**public void paint(Graphics g)** {

g.setColor(color);

g.drawOval(150, 250, 24, 24);

g.fillOval(150, 250, 24, 24);

if(Airhockey.i==4)

g.fillRect(0, 0, 400, 400);

}

**private void formWindowOpened(java.awt.event.WindowEvent evt)** {

jButton1.setBackground(Color.black);

jButton2.setBackground(Color.red);

jButton3.setBackground(Color.orange);

jButton4.setBackground(Color.blue);

jButton6.setBackground(Color.green);

jButton7.setBackground(Color.magenta);

jButton8.setBackground(Color.lightGray);

jButton9.setBackground(Color.yellow);

setFocus();

}

**void assignColor()** {

if(Airhockey.i==1)

myMalletColor = color;

if(Airhockey.i==2)

oppMalletColor = color;

if(Airhockey.i==3)

puckColor = color;

if(Airhockey.i==4)

bgColor = color;

}

**void setFocus()** {

jButton1.requestFocus();

jButton2.requestFocus();

jButton3.requestFocus();

jButton4.requestFocus();

jButton5.requestFocus();

jButton6.requestFocus();

jButton7.requestFocus();

jButton8.requestFocus();

jButton9.requestFocus();

}

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

color = Color.black;

assignColor();

repaint();

setFocus();

}

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)** {

color = Color.red;

assignColor();

repaint();

setFocus();

}

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)** {

color = Color.orange;

assignColor();

repaint();

setFocus();

}

**private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)** {

color = Color.blue;

assignColor();

repaint();

setFocus();

}

**private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new Airhockey().setVisible(true);

}

**private void jButton6ActionPerformed(java.awt.event.ActionEvent evt)** {

color = Color.green;

assignColor();

repaint();

setFocus();

}

**private void jButton7ActionPerformed(java.awt.event.ActionEvent evt)** {

color = Color.magenta;

assignColor();

repaint();

setFocus();

}

**private void jButton8ActionPerformed(java.awt.event.ActionEvent evt)** {

color = Color.lightGray;

assignColor();

repaint();

setFocus();

}

**private void jButton9ActionPerformed(java.awt.event.ActionEvent evt)** {

color = Color.yellow;

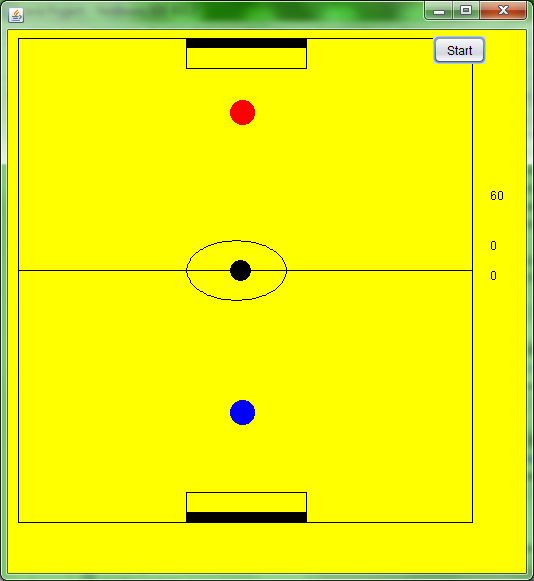
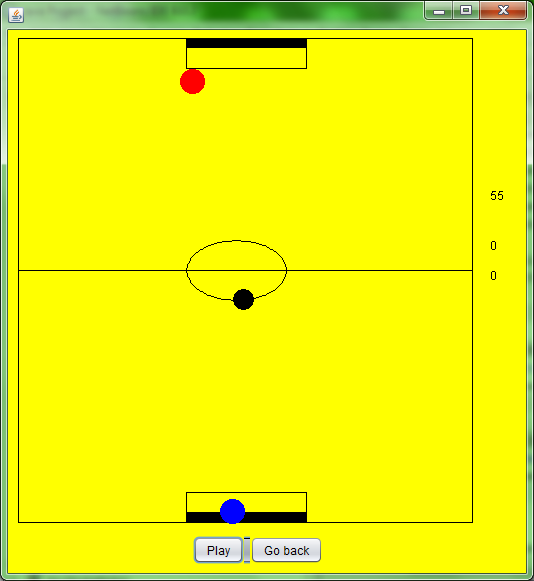
assignColor();

repaint();

setFocus();

}

Game Form

int x = 230, y = 260;

int p = 230, q = 400;

int prevx = 0, prevy = 0 ;

int prevp = 0, prevq = 0;

int prevop = 0, prevoq = 0;

long vx = 0, vy = 0;

double speedx = 0, speedy = 0;

**public void paint(Graphics g)** {

//Background

if(start == 0){

g.setColor(Colorchooser.bgColor);

g.fillRect(0,0,600,600);

}

//Border

g.setColor(Color.black);

g.drawLine(18,38,18,522);

g.drawLine(18,38,472,38);

g.drawLine(472,38,472,522);

g.drawLine(18,522,472,522);

g.drawLine(18,270,472,270);

g.drawRect(186,38,120,30);

g.drawRect(186,492,120,30);

g.fillRect(186,38,120,10);

g.fillRect(186,512,120,10);

g.drawOval(186,240,100,60);

//Erase previously drawn objects

g.setColor(Colorchooser.bgColor);

g.drawOval(prevx,prevy,20,20);

g.fillOval(prevx,prevy,20,20);

g.drawOval(prevp,prevq,24,24);

g.fillOval(prevp,prevq,24,24);

g.drawOval(prevop,prevoq,24,24);

g.fillOval(prevop,prevoq,24,24);

//Draw objects

g.setColor(Colorchooser.myMalletColor);

g.drawOval(p,q,24,24);

g.fillOval(p,q,24,24);

g.setColor(Colorchooser.oppMalletColor);

g.drawOval(pc.p,pc.q,24,24);

g.fillOval(pc.p,pc.q,24,24);

g.setColor(Colorchooser.puckColor);

g.drawOval(x,y,20,20);

g.fillOval(x,y,20,20);

//Display score

g.setColor(Colorchooser.bgColor);

g.fillRect(480,150,30,150);

g.setColor(Color.black);

g.drawString(Integer.toString(pcscore),490,250);

g.drawString(Integer.toString(myscore),490,280);

//Display timer

if(Airhockey.modeselection==2)

if(time%100==0)

g.drawString(Integer.toString(time/100),490,200);

else

g.drawString(Integer.toString(prevtime),490,200);

if((x>186)&&(x<296)&&((y<40)||(y>500))){

g.drawRect(186,38,120,30);

g.drawRect(186,492,120,30);

g.fillRect(186,38,120,10);

g.fillRect(186,512,120,10);

g.setColor(Colorchooser.bgColor);

g.fillRect(100,0,300,38);

g.fillRect(100,523,300,40);

}

if(flag==1){

g.setColor(Colorchooser.bgColor);

g.fillRect(100,523,300,40);

flag = 0;

}

prevx = x; prevy = y;

prevp = p; prevq = q;

prevop = pc.p; prevoq = pc.q;

if(time%100==0)

prevtime = time/100;

}

//Detect collisions

**public boolean hit()** {

return (p-x)\*(p-x) + (q-y)\*(q-y) <= 484;

}

//Calculate horizontal speed of mallet

double mousespeedx(int x){

int dx = MouseInfo.getPointerInfo().getLocation().x - x;

int speed = Math.abs(dx);

return speed;

}

//Calculate vertical speed of mallet

**double mousespeedy(int y)** {

int dy = MouseInfo.getPointerInfo().getLocation().y - y;

int speed = Math.abs(dy);

return speed;

}

//Assign directions

**int xdirection()** {

if((x-p)<0)

return -1;

else

return 1;

}

**int ydirection()** {

if((y-q)<0)

return -1;

else

return 1;

}

**void reset()** {

start = 0;

jButton1.setText("Play Again");

jButton1.setVisible(true);

jButton3.setVisible(true);

jButton3.requestFocus();

jButton1.requestFocus();

jButton4.setVisible(false);

vx = 0; vy = 0;

x = 230; y = 260;

p = 230; q = 400;

pc.p = 230; pc.q = 100;

pcscore = 0; myscore = 0;

t1.stop();

t2.stop();

repaint();

}

int i=0; //Time reference in speed calculation

int j = 500; //Time reference in collisions

int refx, refy;

long refvx, refvy;

double instspeedx, instspeedy;

double avgspeedx, avgspeedy;

int sqrdist = 1000;

int a = 1; //Pc speed coefficient

int myscore = 0, pcscore = 0;

int start = 0;

int time = 6000;

int prevtime;

static int maxspeed;

int xfactor, yfactor;

int whohit;

int pause = 0;

int flag = 0;

pc pc = new pc();

Timer t1, t2;

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

jButton1.setVisible(false);

jButton2.setVisible(false);

jButton3.setVisible(false);

jButton4.setVisible(true);

jButton2.doClick();

start = 1;

if(Airhockey.diffselection==1)

maxspeed = 3;

if(Airhockey.diffselection==2)

maxspeed = 5;

ActionListener action;

action = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

//Calculate speed of mallet

if(i==0){

refx = p;

refy = q;

}

i++; j++; pc.j++;

if(!((x>186)&&(x<296)&&(y<40))&&!((x>186)&&(x<296)&&(y>500)))

time--;

if(i==20){

avgspeedx = mousespeedx(refx);

avgspeedy = mousespeedy(refy);

i=0;

}

instspeedx = mousespeedx(p);

instspeedy = mousespeedx(q);

//Opponent movement

sqrdist = (pc.p-x)\*(pc.p-x) + (pc.q-y)\*(pc.q-y);

pc.sx = vx + (x-pc.p)/2;

if(Math.abs(pc.sx)>maxspeed)

pc.sx/= Math.ceil((float)(Math.abs(pc.sx)/maxspeed));

if((x<400)&&(x>40)&&(pc.q<y))

pc.p+=pc.sx;

if((y<200)&&(x>40)&&(x<400)&&(vy<0)&&(pc.j>50)&&(!pc.hit(sqrdist))&&(y-

pc.q>50)&&(Math.abs(x-pc.p)<30)&&(pc.q<250)&&(pc.q<y))

pc.attack(a,vy);

else

pc.defend(vy);

if((vx==0)&&(vy==0)&&(y==150))

pc.q+=3;

//Change velocities during collision

if(hit()&&((j>20)||(vx==-refvx)||(vy==-refvy))){

whohit=0;

j=0;

vx = vx - vx\*Math.abs(x-p)/25;

vy = vy - vy\*Math.abs(y-q)/25;

if((p-x)\*(p-x) + (q-y)\*(q-y)<350){

xfactor = Math.abs(x-p)+20;

yfactor = Math.abs(y-q)+20;

}

else{

xfactor = Math.abs(x-p);

yfactor = Math.abs(y-q);

}

if((avgspeedx<instspeedx)&&(instspeedx>50))

speedx = instspeedx;

else

speedx = avgspeedx;

if((avgspeedy<instspeedy)&&(instspeedy>50))

speedy = instspeedy;

else

speedy = avgspeedy;

vx =(Math.abs(vx)\*3/5 + Math.round(speedy\*xfactor)/150)\*xdirection();

vy =(Math.abs(vy)\*3/5 + Math.round(speedy\*yfactor)/150)\*ydirection();

if((vx==0)&&(speedx==0))

vx = xdirection();

if((vy==0)&&(speedy==0))

vy = ydirection();

if(vx>8) vx = 8;

if(vx<-8) vx = -8;

if(vy>8) vy = 8;

if(vy<-8) vy = -8;

refvx = vx; refvy = vy;

}

if(pc.hit(sqrdist)){

whohit=1;

pc.j=0;

vx = vx - vx\*Math.abs(x-pc.p)/3 + 3;

vy = vy - vy\*Math.abs(y-pc.q)/3 + 3;

vx = (Math.abs(vx)\*4/5 + Math.round(pc.sx\*Math.abs(x-

pc.p))/150)\*pc.xdirection(x);

vy = (Math.abs(vy)\*4/5 + Math.round(pc.sy\*Math.abs(y-

pc.q))/150)\*pc.ydirection(y);

if(vx==0)

vx = xdirection();

if((vy==0)&&(pc.sy==0))

vy = ydirection();

if(vx>8) vx = 8;

if(vx<-8) vx = -8;

if(vy>8) vy = 8;

if(vy<-8) vy = -8;

Random r = new Random();

a = r.nextInt(2)+1;

}

}

};

t1 = new Timer(100, action);

t1.setDelay(1);

t1.setRepeats(true);

t1.start();

this.getContentPane().setBackground(Colorchooser.bgColor);

}

**private void formMouseMoved(java.awt.event.MouseEvent evt)** {

//Mallet movement

if(!hit()&&(start==1)){

if(MouseInfo.getPointerInfo().getLocation().x < 450 &&

MouseInfo.getPointerInfo().getLocation().x > 20)

p = MouseInfo.getPointerInfo().getLocation().x;

if(MouseInfo.getPointerInfo().getLocation().y < 500 &&

MouseInfo.getPointerInfo().getLocation().y >272)

q = MouseInfo.getPointerInfo().getLocation().y;

} }

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)** {

//Puck movement

ActionListener action;

action = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

repaint();

x+=vx; y+=vy;

if(((x>450)&&(vx>0))||((x<20)&&(vx<0)))

vx = -vx;

if(((y>500)&&(vy>0)&&((x<186)||(x>296)))||((y<40)&&(vy<0)&&((x<186)||(x>296))))

vy = -vy;

if((x>186)&&(x<296)&&(y<40)){

repaint();

if(whohit==0)

JOptionPane.showMessageDialog(null,"Your goal");

else

JOptionPane.showMessageDialog(null,"Own goal !");

x = 230; y = 150;

pc.p = 230; pc.q = 100;

vx = 0; vy = 0;

myscore++;

if((Airhockey.modeselection==1)&&(myscore==5)){

repaint();

JOptionPane.showMessageDialog(null,"You win");

reset();

}

}

if((x>186)&&(x<296)&&(y>500)){

repaint();

if(whohit==1)

JOptionPane.showMessageDialog(null,"Opponent's goal");

else

JOptionPane.showMessageDialog(null,"Own goal !");

x = 230; y = 400;

vx = 0; vy = 0;

pcscore++;

if((Airhockey.modeselection==1)&&(pcscore==5)){

repaint();

JOptionPane.showMessageDialog(null,"You lose");

reset();

}

}

if((Airhockey.modeselection==2)&&(time==0)){

if(myscore>pcscore)

JOptionPane.showMessageDialog(null,"You win");

if(myscore<pcscore)

JOptionPane.showMessageDialog(null,"You lose");

if(myscore==pcscore)

JOptionPane.showMessageDialog(null,"It is a draw");

reset();

time = 6000;

}

}

};

t2 = new Timer(100, action);

t2.setDelay(10);

t2.setRepeats(true);

t2.start();

}

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new Airhockey().setVisible(true);

}

**private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)** {

if(pause==1){

jButton3.setVisible(false);

jButton4.setText("Pause");

jButton1.doClick();

pause = 0;

flag = 1;

}

else{

t1.stop();

t2.stop();

pause = 1;

jButton3.setVisible(true);

jButton3.requestFocus();

jButton4.setText("Play");

jButton4.requestFocus();

}

repaint();

AI Player Class

**public class pc** {

int p = 230, q = 100;

long sx, sy;

int j = 0;

//Detect collisions

**public boolean hit(int sqrdist)** {

return sqrdist <= 484;

}

//Assign directions

**int xdirection(int x)** {

if((x-p)<0)

return -1;

else

return 1;

}

**int ydirection(int y)** {

if((y-q)<0)

return -1;

else

return 1;

}

**void attack(int a,long vy)** {

sy=-vy\*a;

if(Math.abs(sy)>sop.maxspeed)

sy/= Math.abs(sy)/sop.maxspeed;

if(sy==0)

sy=3;

q+=sy;

}

**void defend(long vy)** {

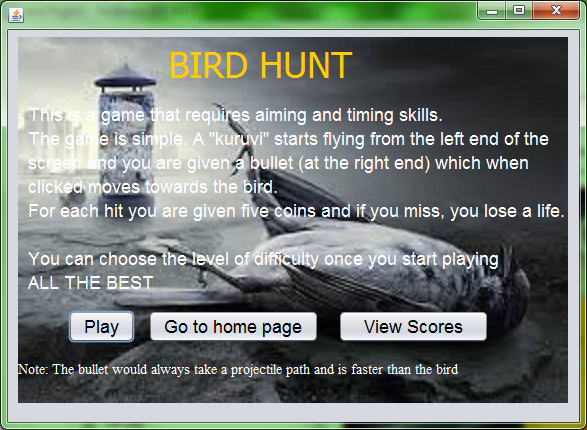
if(q>70)

q-=vy/2; }

}

**Bird Hunt**

Introduction Form



**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new movebtnn().setVisible(true);

}

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new IntroForm().setVisible(true);

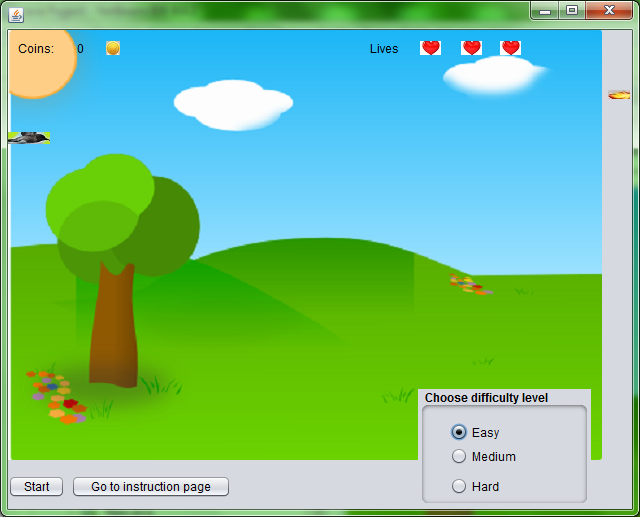
}

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new huntscore().setVisible(true); }

Game Form





Timer t2;

float slope;

double fact;

Timer t;

Random r=new Random();

int i=0, j=0, m=0, k=0, l=0, score=0, n=0, ref=0, a=2;

int x=0, y=0, y2=60, x2=600;

float y1= r.nextInt(100);

double y3;

String[] difficulty = {"Easy","Medium","Hard"};

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

if(m==1)

t2.stop();

ActionListener action;

action = new ActionListener(){ @Override

public void actionPerformed(ActionEvent e){} };

if(m==1){

action = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

y+=k-5;

jButton1.setLocation(x,y);

k++;

}

};

}

int m1 = r.nextInt(2);

int m2 = r.nextInt(2);

int md = r.nextInt(10);

slope = (float)(m1) + (float)md/10;

y1=r.nextInt(100);

jButton1.setIcon(new ImageIcon("F:\\Java Project\\src\\kuruvi.jpg"));

if((slope==0)||(a==2)){

y1 = r.nextInt(200)+100;

}

fact = (r.nextInt(50)+20);

if(m==0){

y = (int)y1;

action = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if(a==0){

x+=8;

y1 += 5\*slope;

y =Math.round(y1);

jButton1.setLocation(x,y);

}

if(a==1){

y+=(2\*l+1)/5;

x+=10;

jButton1.setLocation(x,y);

l++;

}

if(a==2){

x+=8;

y3 =(double)y1;

double ref = y3;

y3 = ref + Math.sin(x)\*fact;

y1 = (float)y3;

y = Math.round(y1);

jButton1.setLocation(x,y);

}

if(((y>=550)||(x>600))&&(m==0)){

j=0; m=0; k=0; x=0; l=0;

y2=60; x2=600;

jButton3.setLocation(600,60);

t2.stop();

i++;

if(ref<i)

JOptionPane.showMessageDialog(null,"You missed it\nYou lost a

life\n\nClick OK to continue");

jButton1.doClick();

}

}

};

}

t2 = new Timer(100, action);

t2.setDelay(100);

t2.setRepeats(true);

t2.start();

}

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)** {

int b=0;

if((jRadioButton1.isSelected()==false)&&(jRadioButton2.isSelected()==false)&&(jRadioButton3.isSelected()==false)){

JOptionPane.showMessageDialog(this,"Please choose a difficulty level");

b++;

}

if(b==0){

jButton7.setVisible(true);

jButton6.setVisible(false);

jPanel2.setVisible(false);

jButton2.setVisible(false);

jButton1.doClick();

}

}

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)** {

m++;n=0;

ActionListener action;

action = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if(n==0){y2+=(2\*j+1)/5;

x2-=10;

jButton3.setLocation(x2,y2);

j++;

}

if((y>=550)||(x>600)||(y2>600)){

j=0; m=0; k=0; x=0; l=0;

y2=60; x2=600;

jButton3.setLocation(600,60);

t.stop();

t2.stop();

i++;

if(ref<i)JOptionPane.showMessageDialog(null,"You missed it\nYou lost a

life\n\nClick OK to continue");

jButton1.doClick();

}

if(((x-x2)<2)&&((x-x2)>-45)&&((y-y2)<10)&&((y-y2)>-5)){

i--;score+=5;n++;ref=i+1;

jButton3.setLocation(600,60);

jLabel6.setText(score+"");

jButton1.doClick();

JOptionPane.showMessageDialog(null,"That was great\nYou earned 5

coins\n\nClick OK to continue");

}

if(i==1)

jLabel1.setIcon(new ImageIcon("F:\\Java Project\\src\\hrtc.png"));

if(i==2)

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\hrtc.png"));

if(i==3){

jLabel3.setIcon(new ImageIcon("F:\\Java Project\\src\\hrtc.png"));

jLabel7.setVisible(true);

try{

Class.forName("java.sql.Driver");

Connection con = DriverManager.getConnection(“jdbc:mysql://localhost:/

java project”,”root”,”mysql”);

Statement st = con.createStatement();

String check = "select "+ difficulty[a] +" from birdhunt where username

= '"+loginform.user+"';";

ResultSet rs = st.executeQuery(check);

rs.next();

if(score>rs.getInt(1)){

String add = "update birdhunt set "+ difficulty[a] +" = "+score+" where

username = '"+loginform.user+"';";

st.executeUpdate(add);

}

}

catch(Exception E){

System.out.print(E.getMessage());

}

t.stop();

t2.stop();

jButton5.setVisible(true);

jButton7.setVisible(false);

jButton6.setVisible(true);

}

}

};

t = new Timer(100, action);

t.setDelay(60);

t.setRepeats(true);

t.start();

}

**private void formWindowOpened(java.awt.event.WindowEvent evt)** {

jLabel1.setIcon(new ImageIcon("F:\\Java Project\\src\\hrt.jpg"));

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\hrt.jpg"));

jLabel3.setIcon(new ImageIcon("F:\\Java Project\\src\\hrt.jpg"));

jLabel8.setIcon(new ImageIcon("F:\\Java Project\\src\\coin.jpg"));

jLabel7.setVisible(false);

jButton5.setVisible(false);

jLabel6.setText(score+"");

jButton3.setIcon(new ImageIcon("F:\\Java Project\\src\\bullet.jpg"));

jButton1.setIcon(new ImageIcon("F:\\Java Project\\src\\kuruvi.jpg"));

jButton7.setVisible(false);

}

**private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)** {

jLabel1.setIcon(new ImageIcon("F:\\Java Project\\src\\hrt.jpg"));

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\hrt.jpg"));

jLabel3.setIcon(new ImageIcon("F:\\Java Project\\src\\hrt.jpg"));

jLabel7.setVisible(false);

i=0; j=0; m=0; k=0; score=0; n=0; ref=0;

x=0; y=0; y2=60; x2=600;

y1= r.nextInt(100);

jButton5.setVisible(false);

jButton2.setVisible(true);

jLabel6.setText(score+"");

jPanel2.setVisible(true);

}

**private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

a=0;

}

**private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt)** {

a=1;

}

**private void jRadioButton3ActionPerformed(java.awt.event.ActionEvent evt)** {

a=2;

}

**private void jButton6ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new Shootthekuruvi().setVisible(true);

}

**private void jButton7ActionPerformed(java.awt.event.ActionEvent evt)** {

jButton3.setLocation(600,60);

t.stop();

t2.stop();

jLabel1.setIcon(new ImageIcon("F:\\Java Project\\src\\hrt.jpg"));

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\hrt.jpg"));

jLabel3.setIcon(new ImageIcon("F:\\Java Project\\src\\hrt.jpg"));

jLabel7.setVisible(false);

i=0; j=0; m=0; k=0; score=0; n=0; ref=0; l=0;

x=0; y=0; y2=60; x2=600;

y1= r.nextInt(100);

jButton5.setVisible(false);

jButton2.setVisible(true);

jLabel6.setText(score+"");

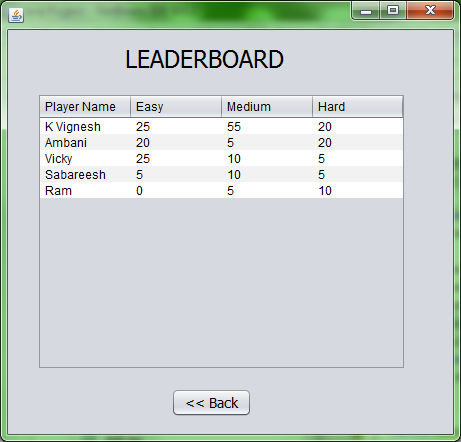
jPanel2.setVisible(true);

jButton6.setVisible(true);

jButton7.setVisible(false);

}

Scores Form



**private void formWindowOpened(java.awt.event.WindowEvent evt)** {

DefaultTableModel dtm = (DefaultTableModel)(jTable1.getModel());

try{

Class.forName("java.sql.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost/

javaproject","root","mysql");

Statement st = con.createStatement();

String select = "select \* from birdhunt order by easy+medium+hard desc;";

ResultSet rs = st.executeQuery(select);

while(rs.next()){

dtm.addRow(newObject[]

{rs.getString(1),rs.getInt(2),rs.getInt(3),rs.getInt(4)});

}

}

catch(Exception e){

System.out.print(e.getMessage());

}

}

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new Shootthekuruvi().setVisible(true);

}

**Flappy Bird**

Introduction Form



**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new flaps().setVisible(true);

}

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new IntroForm().setVisible(true);

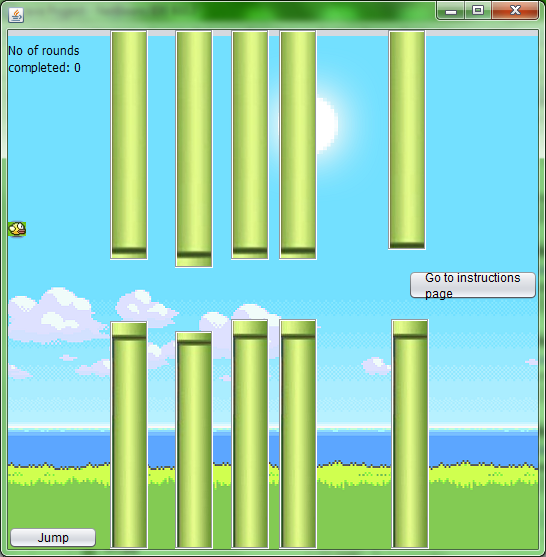
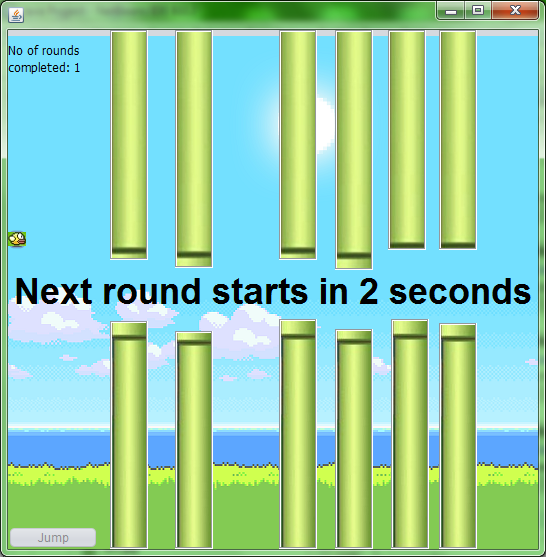
}

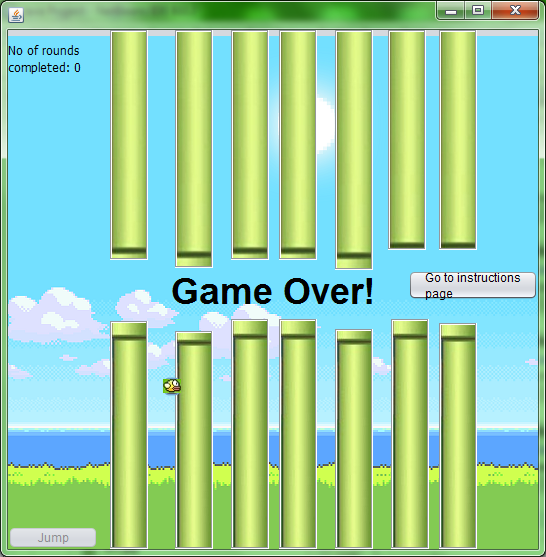
**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new flappyscore().setVisible(true); }

Game Form



**void pillars(int r1,int r3,int r5,int r7,int r9,int r11,int r13)** {

if(r1==0){

jLabel1.setVisible(false);

jLabel2.setVisible(false);

}

if(r3==0){

jLabel3.setVisible(false);

jLabel4.setVisible(false);

}

if(r5==0){

jLabel5.setVisible(false);

jLabel6.setVisible(false);

}

if(r7==0){

jLabel7.setVisible(false);

jLabel8.setVisible(false);

}

if(r9==0){

jLabel9.setVisible(false);

jLabel10.setVisible(false);

}

if(r11==0){

jLabel11.setVisible(false);

jLabel12.setVisible(false);

}

if(r13==0){

jLabel13.setVisible(false);

jLabel14.setVisible(false);

}

}

Timer t;

int x=0, y=200;

int i=-5, j=0, k=0;

Random r=new Random();

intr1=r.nextInt(3),r3=r.nextInt(3),r5=r.nextInt(3),r7=r.nextInt(3),r9=r.nextInt(3),r11=r.nextInt(3),r13=r.nextInt(3);

int l1, l2, l3, l4, l5, l6, l7, l8, l9, l10, l11, l12, l13, l14;

int m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, m14;

Timer t1, t2;

int m=0, p=4, round=1, score;

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)** {

jLabel15.setVisible(false);

jButton5.setVisible(false);

l1=jLabel1.getHeight();

l2=jLabel2.getLocation().y;

l3=jLabel3.getHeight();

l4=jLabel4.getLocation().y;

l5=jLabel5.getHeight();

l6=jLabel6.getLocation().y;

l7=jLabel7.getHeight();

l8=jLabel8.getLocation().y;

l9=jLabel9.getHeight();

l10=jLabel10.getLocation().y;

l11=jLabel11.getHeight();

l12=jLabel12.getLocation().y;

l13=jLabel13.getHeight();

l14=jLabel14.getLocation().y;

m1=jLabel1.getLocation().x;

m3=jLabel3.getLocation().x;

m5=jLabel5.getLocation().x;

m7=jLabel7.getLocation().x;

m9=jLabel9.getLocation().x;

m11=jLabel11.getLocation().x;

m13=jLabel13.getLocation().x;

if(j==1)t.stop();

i=-5; k=0;

j=1;

ActionListener action;

action = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

x+=5;

y+=2\*i;

jButton1.setLocation(x,y);

i++;

if(r1!=0){

if(((x>=m1-18)&&(x<=m1+45))&&((y<=l1)||(y>=l2-15))){

t.stop();jButton2.setEnabled(false);

jButton1.doClick(); jButton5.setVisible(true);

}

}

if(r3!=0){

if(((x>=m3-18)&&(x<=m3+45))&&((y<=l3)||(y>=l4-15))){

t.stop();jButton2.setEnabled(false);

jButton1.doClick(); jButton5.setVisible(true);

}

}

if(r5!=0){

if(((x>=m5-18)&&(x<=m5+45))&&((y<=l5)||(y>=l6-15))){

t.stop();jButton2.setEnabled(false);

jButton1.doClick(); jButton5.setVisible(true);

}

}

if(r7!=0){

if(((x>=m7-18)&&(x<=m7+45))&&((y<=l7)||(y>=l8-15))){

t.stop();jButton2.setEnabled(false);

jButton1.doClick(); jButton5.setVisible(true);

}

}

if(r9!=0){

if(((x>=m9-18)&&(x<=m9+45))&&((y<=l9)||(y>=l10-15))){

t.stop();jButton2.setEnabled(false);

jButton1.doClick(); jButton5.setVisible(true);

}

}

if(r11!=0){

if(((x>=m11-18)&&(x<=m11+45))&&((y<=l11)||(y>=l12-15))){

t.stop();jButton2.setEnabled(false);

jButton1.doClick(); jButton5.setVisible(true);

}

}

if(r13!=0){

if(((x>=m13-18)&&(x<=m13+45))&&((y<=l13)||(y>=l14-15))){

t.stop();jButton2.setEnabled(false);

jButton1.doClick(); jButton5.setVisible(true);

}

}

if(x>=530){k=1;

t.stop();jButton2.setEnabled(false);

jButton1.doClick();

}

}

};

t = new Timer(100, action);

t.setDelay(60);

t.setRepeats(true);

t.start();

}

**private void formWindowOpened(java.awt.event.WindowEvent evt)** {

pillars(r1,r3,r5,r7,r9,r11,r13);

jLabel15.setVisible(false);

jButton3.setVisible(false);

jButton4.setVisible(false);

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\pillar.jpg"));

jLabel1.setIcon(new ImageIcon("F:\\Java Project\\src\\dpill.jpg"));

jLabel3.setIcon(new ImageIcon("F:\\Java Project\\src\\dpill.jpg"));

jLabel4.setIcon(new ImageIcon("F:\\Java Project\\src\\pillar.jpg"));

jLabel5.setIcon(new ImageIcon("F:\\Java Project\\src\\dpill.jpg"));

jLabel6.setIcon(new ImageIcon("F:\\Java Project\\src\\pillar.jpg"));

jLabel7.setIcon(new ImageIcon("F:\\Java Project\\src\\dpill.jpg"));

jLabel8.setIcon(new ImageIcon("F:\\Java Project\\src\\pillar.jpg"));

jLabel9.setIcon(new ImageIcon("F:\\Java Project\\src\\dpill.jpg"));

jLabel10.setIcon(new ImageIcon("F:\\Java Project\\src\\pillar.jpg"));

jLabel11.setIcon(new ImageIcon("F:\\Java Project\\src\\dpill.jpg"));

jLabel12.setIcon(new ImageIcon("F:\\Java Project\\src\\pillar.jpg"));

jLabel13.setIcon(new ImageIcon("F:\\Java Project\\src\\dpill.jpg"));

jLabel14.setIcon(new ImageIcon("F:\\Java Project\\src\\pillar.jpg"));

jButton1.setIcon(new ImageIcon("F:\\Java Project\\src\\bird.jpg"));

}

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

m=0;

ActionListener action;

action = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

jLabel15.setVisible(true);

if(k==1){

jLabel15.setText("Round "+round+" completed");

jLabel19.setText("completed: "+round);

}

else{score= (round-1)\*5;

jLabel15.setText("Game Over!");

try{

Class.forName("java.sql.Driver");

Connectioncon=DriverManager.getConnection("jdbc:mysql:

//localhost/javaproject","root","mysql");

Statement st = con.createStatement();

String check = "select highscore from flappybird where username =

'"+loginform.user+"';";

ResultSet rs = st.executeQuery(check);

rs.next();

if(score>rs.getInt(1)){

String add = "update flappybird set highscore = "+score+" where

username = '"+loginform.user+"';";

st.executeUpdate(add);

}

}

catch(Exception E){

System.out.print(E.getMessage());

}

}

y+=5;

jButton1.setLocation(x,y);

if(y>=520){round++;

t1.stop();

x=0; y=200;

jButton1.setLocation(x,y);

r1=r.nextInt(3);r3=r.nextInt(3);r5=r.nextInt(3);r7=r.nextInt(3);

r9=r.nextInt(3);r11=r.nextInt(3);r13=r.nextInt(3);

j=0;

if(k==0)jLabel15.setText("Your score is "+score);

if(k==1){jButton3.doClick();

}

else jButton4.setVisible(true);

}

}

};

t1 = new Timer(100, action);

t1.setDelay(60);

t1.setRepeats(true);

t1.start();

jLabel1.setVisible(true);

jLabel2.setVisible(true);

jLabel3.setVisible(true);

jLabel4.setVisible(true);

jLabel5.setVisible(true);

jLabel6.setVisible(true);

jLabel7.setVisible(true);

jLabel8.setVisible(true);

jLabel9.setVisible(true);

jLabel10.setVisible(true);

jLabel11.setVisible(true);

jLabel12.setVisible(true);

jLabel13.setVisible(true);

jLabel14.setVisible(true);

}

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)** {

pillars(r1,r3,r5,r7,r9,r11,r13);

p=4;

ActionListener action;

action = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

p--;

jLabel15.setText("Next round starts in "+p+" seconds");

if(p==0){

t2.stop();

jButton2.setEnabled(true);

jButton2.doClick();

}

}

};

t2 = new Timer(100, action);

t2.setDelay(1000);

t2.setRepeats(true);

t2.start();

}

**private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)** {

r1=r.nextInt(3);r3=r.nextInt(3);r5=r.nextInt(3);r7=r.nextInt(3);r9=r.nextInt(3);r11=r.nextInt(3);r13=r.nextInt(3);

pillars(r1,r3,r5,r7,r9,r11,r13);

jButton2.setEnabled(true);

jLabel15.setVisible(false);

jButton4.setVisible(false);

round =1;

jLabel19.setText("completed: 0");

jButton5.setVisible(false);

}

**private void jLabel15KeyPressed(java.awt.event.KeyEvent evt)** {

}

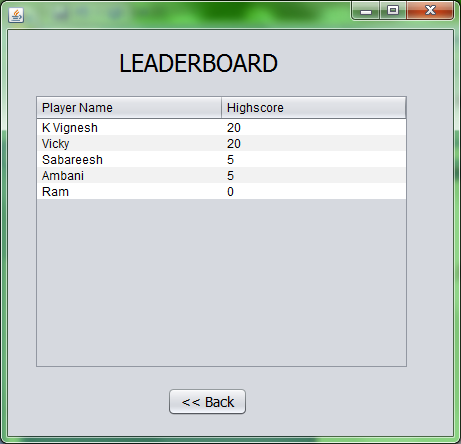
**private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new flappybird().setVisible(true);

}

Score Form



**private void formWindowOpened(java.awt.event.WindowEvent evt)** {

DefaultTableModel dtm = (DefaultTableModel)(jTable1.getModel());

try{

Class.forName("java.sql.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost/

javaproject","root","mysql");

Statement st = con.createStatement();

String select = "select \* from flappybird;";

ResultSet rs = st.executeQuery(select);

while(rs.next()){

dtm.addRow(new Object[]{rs.getString(1),rs.getInt(2)});

} }

catch(Exception e){

System.out.print(e.getMessage());

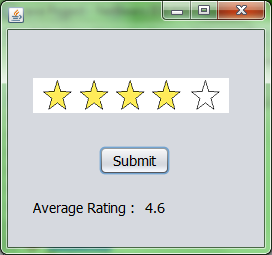
} }

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

this.setVisible(false);

new flappybird().setVisible(true); }

**Rating**

**** 

int rate=0;

**private void formWindowOpened(java.awt.event.WindowEvent evt)** {

l1.setText(loginform.avgrating);

}

**private void formMouseMoved(java.awt.event.MouseEvent evt)** {

if(MouseInfo.getPointerInfo().getLocation().y>65&& MouseInfo.getPointerInfo().getLocation().y<120&& MouseInfo.getPointerInfo().getLocation().x<220){

if(MouseInfo.getPointerInfo().getLocation().x>(jLabel2.getLocation().x+162)){

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\star5.png"));

rate=5;

}

else if(MouseInfo.getPointerInfo().getLocation().x>(jLabel2.getLocation().x+129)) {

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\star4.png"));

rate=4;

}

else if(MouseInfo.getPointerInfo().getLocation().x>(jLabel2.getLocation().x+96)) {

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\star3.png"));

rate=3;

}

else if(MouseInfo.getPointerInfo().getLocation().x>(jLabel2.getLocation().x+63)) {

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\star2.png"));

rate=2;

}

else if(MouseInfo.getPointerInfo().getLocation().x>(jLabel2.getLocation().x+25)) {

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\star1.png"));

rate=1;

}

else

jLabel2.setIcon(new ImageIcon("F:\\Java Project\\src\\star0.png"));

}

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)** {

try{

Class.forName("java.sql.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost/

javaproject","root","mysql");

Statement st = con.createStatement();

String add = "update userdata set rating = "+rate+" where username = '"+loginform.user+"';";

st.executeUpdate(add);

}

catch(Exception e){

System.out.print(e.getMessage());

}

JOptionPane.showMessageDialog(this,"Thank you for rating this app");

this.setVisible(false);

new Endframe().setVisible(true);

}

**Closing**



int i=1;

**private void formWindowOpened(java.awt.event.WindowEvent evt)** {

ActionListener action;

action = new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

i++;

if(i==6)

System.exit(0);

}

};

Timer t = new Timer(100, action);

t.setDelay(1000);

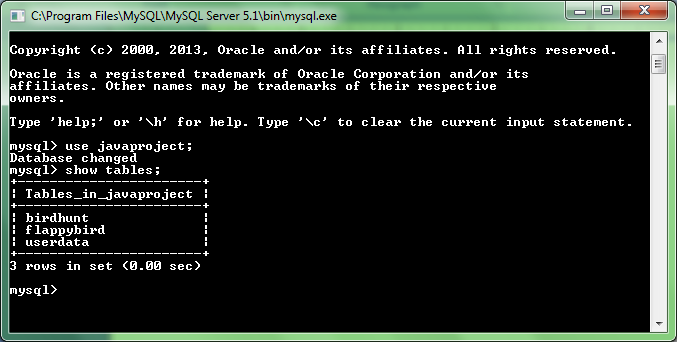
t.setRepeats(true);

t.start();

}

Database

Database name: javaproject

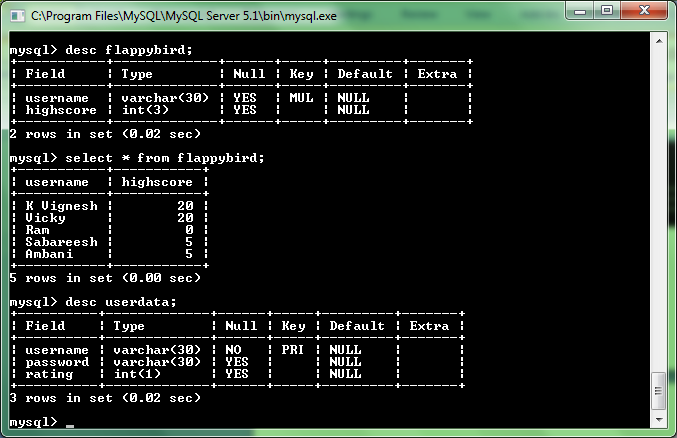


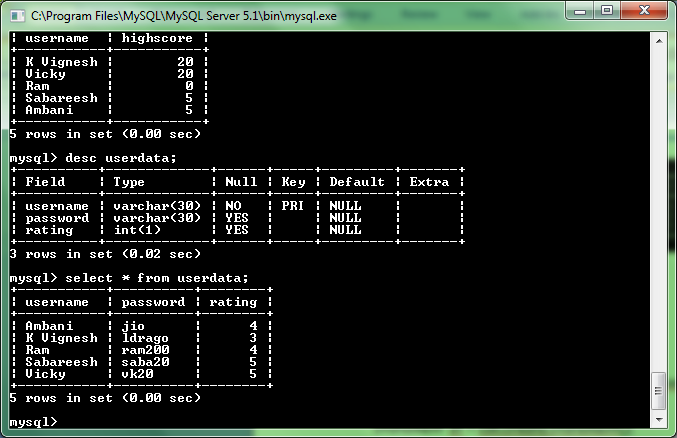
Tables:

1. Userdata

This table contains the login details of all users and the rating given by the user.







Connectivity code for userdata

Class.forName("java.sql.Driver");

Connection con = DriverManager.getConnection(“jdbc:mysql://localhost:/java

project”,”root”,”mysql”);

Statement st = con.createStatement();

String check = "select \* from userdata;";

String include = "insert into userdata values('"+name+"','"+password+"',-1);";

ResultSet rs = st.executeQuery(check);

while(rs.next()){

if(name.equals(rs.getString(1))){

exist=0;

JOptionPane.showMessageDialog(this,"This username already exists");

break;

}

}

if(exist==1){

st.executeUpdate(include);

st.executeUpdate("insert into birdhunt values('"+name+"',0,0,0);");

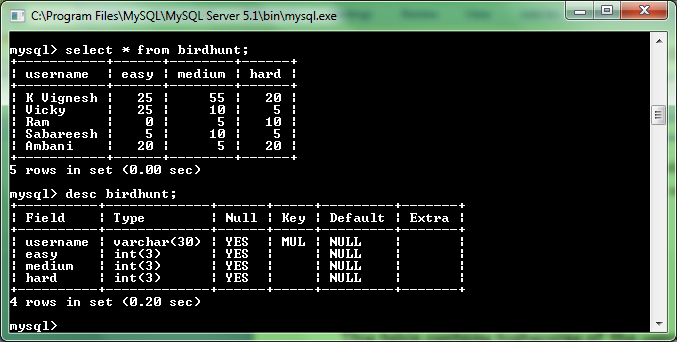
st.executeUpdate("insert into flappybird values('"+name+"',0);");

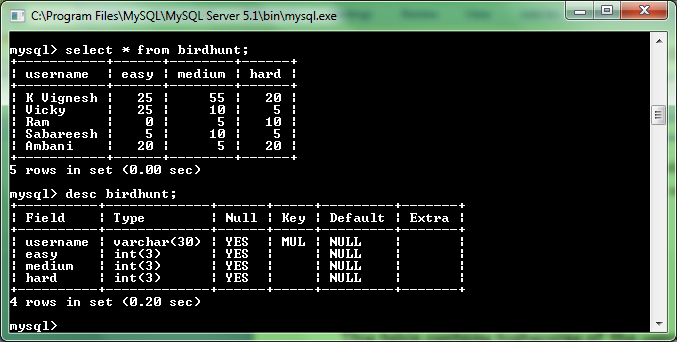
}

1. Birdhunt

This table contains highscores of the users in bird hunt game.







Connectivity code for birdhunt

Class.forName("java.sql.Driver");

Connectioncon = DriverManager.getConnection("jdbc:mysql://

localhost/javaproject","root","mysql");

Statement st = con.createStatement();

String check = "select "+ difficulty[a] +" from birdhunt where username

= '"+loginform.user+"';";

ResultSet rs = st.executeQuery(check);

rs.next();

if(score>rs.getInt(1)){

String add = "update birdhunt set "+ difficulty[a] +" = "+score+" where

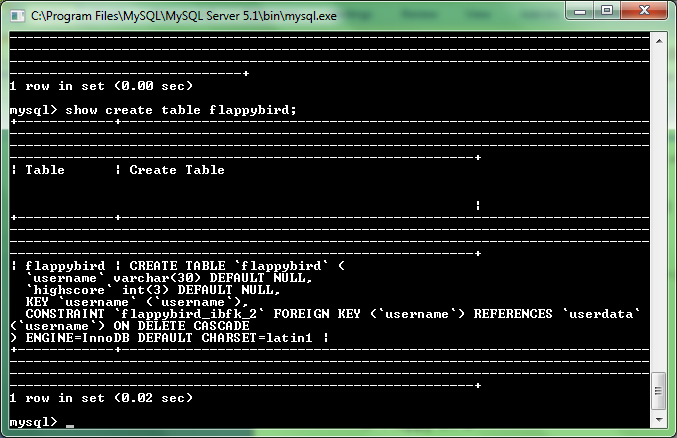
username = '"+loginform.user+"';";

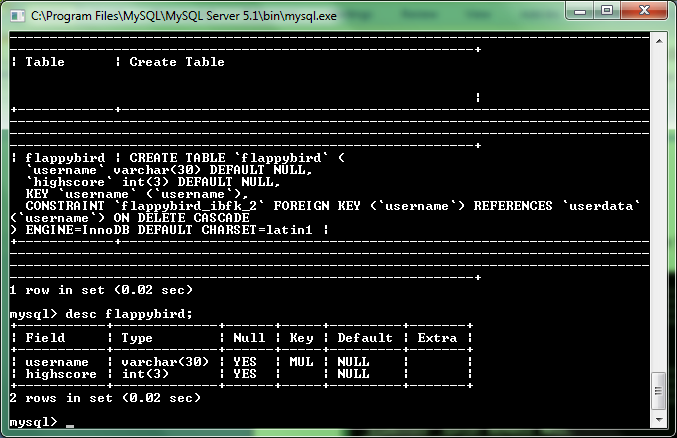
st.executeUpdate(add);

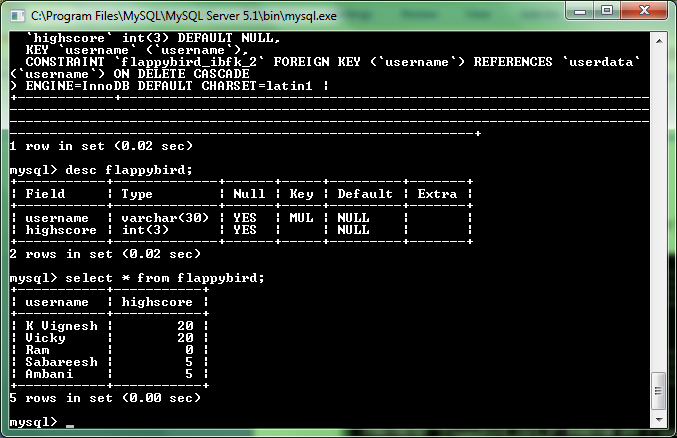
}

1. Flappy Bird

This table contains highscores of users in flappy bird game.







Connectivity code for flappybird

Class.forName("java.sql.Driver");

Connectioncon = DriverManager.getConnection("jdbc:mysql://

localhost/javaproject","root","mysql");

Statement st = con.createStatement();

String check = "select highscore from flappybird where username

= '"+loginform.user+"';";

ResultSet rs = st.executeQuery(check);

rs.next();

if(score>rs.getInt(1)){

String add = "update flappybird set highscore = "+score+" where

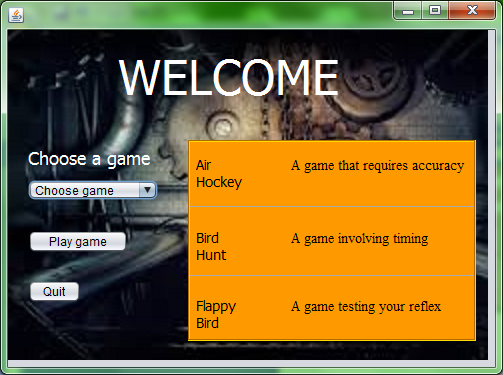
username = '"+loginform.user+"';";

st.executeUpdate(add);

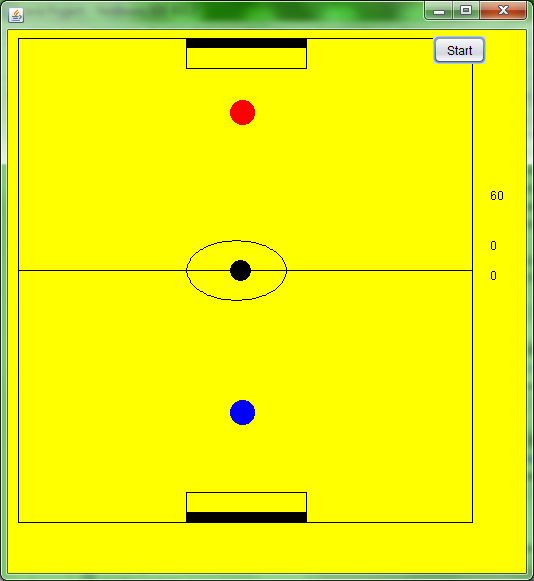
}

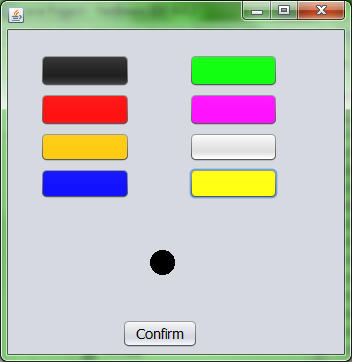
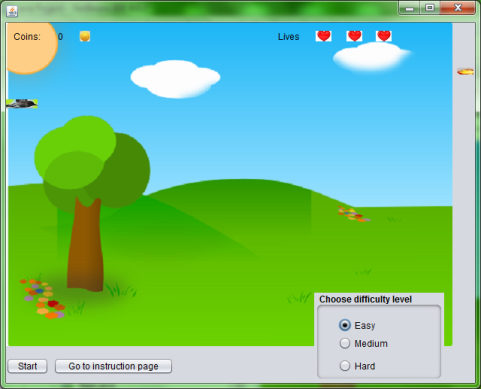
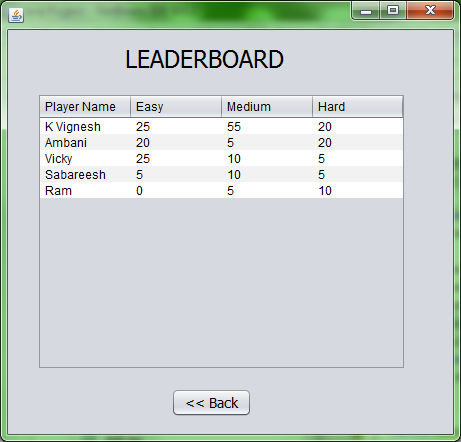
**Form Sequence**

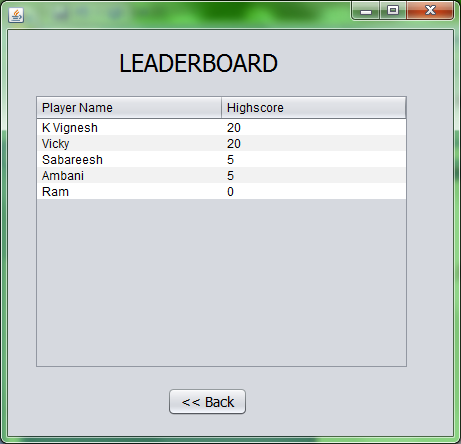
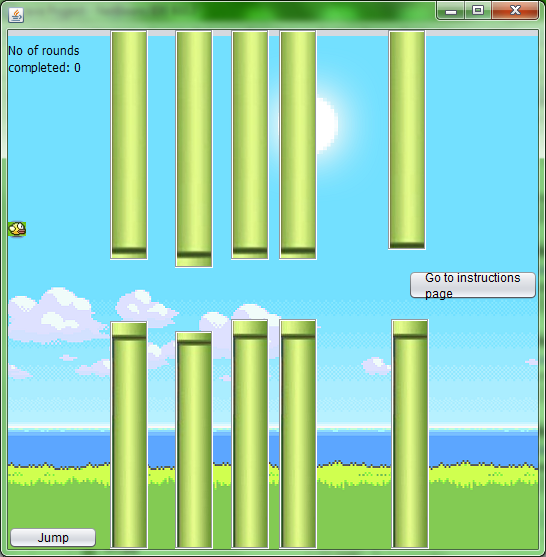
** **

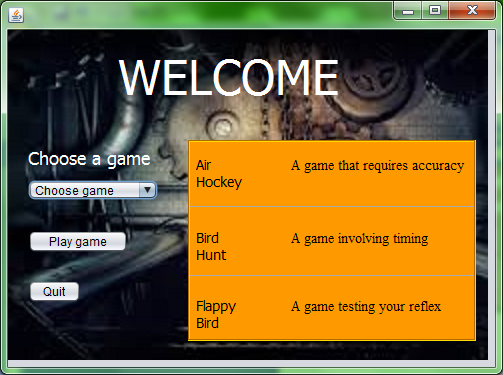
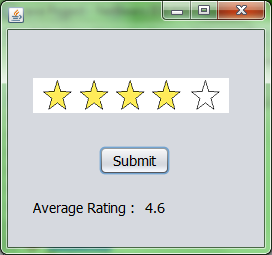
**** 

** **

**  **

**** 

**** 

**  **