1.

Name:
Snake_mode

members:
int centerx
int centery,
int radius,
int pcenterx,
int pcentery;

method :
snake_mode(){

2.

Name:
Snakethegame2

Members:
JMenuBar menu = null;
JMenuBar info = null;
JMenu help = null;
JMenu controls = null;
JMenuItem play = null;
JMenuItem restart = null;
JMenuItem instructions = null;
JMenuItem email = null;
JMenuItem Phonenumber = null;

method:

3.

Name:
Snakethegame2()

methods:
menu = new JMenuBar();
info = new JMenuBar();
play = new JMenuItem("Play Game");
controls = new JMenu("Controls for Game");
help = new JMenu("Help"); //help on the user bar
restart = new JMenuItem("Restart Game");
instructions = new JMenuItem("Instructions");
email = new JMenuItem("Email: Mason_schaefer@yahoo.com");
Phonenumber = new JMenuItem("For Phone Number, Email Me, Thanks!");

```
        restart.setMnemonic(KeyEvent.VK_R); //shift r allows users to reset game
      restart.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_R,
          InputEvent.SHIFT_MASK));
        restart.setMnemonic(KeyEvent.VK_I); //shift I allows user to see the instructions
        instructions.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_I,                InputEvent.SHIFT_MASK));


Methods:
  play.setMnemonic(KeyEvent.VK_P);  //alows user to play the game with shift
  play.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_P, InputEvent.SHIFT_MASK));

  this.add(panel, BorderLayout.CENTER);
  this.setLocationRelativeTo(null);
  this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  play.addActionListener(new ActionListener(){public void actionPerformed(ActionEvent eee)
{play();}})
  restart.addActionListener(new ActionListener(){public void actionPerformed(ActionEvent ee)
{restart();}});
  instructions.addActionListener(new ActionListener(){public void actionPerformed(ActionEvent
ee){instruct();}});
  this.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent e)
{panel.keyPress(e);}});
  this.setVisible(true);
  this.pack();


4.
Name: public void play(){

members:
          if(firstplay){
        panel.t.start();
         firstplay = false;
  methods:
        play.setEnabled(false);

5.
Name:
public void restart(){

members
 int pos = 0;
  panel.vector.clear();
  Snake_mode[] mode = new Snake_mode[5];  for(int i =0; i < 5; i++){
   mode[i] = new Snake_mode();
   mode[i].centerx = 200;
```

```
   mode[i].centery = 150-pos;
   mode[i].pcentery = 150-pos;
   mode[i].pcenterx = 200;
   panel.vector.add(mode[i]); //adds body part
   pos = pos+10;

 methods:
  panel.gameover = false;
  panel.dir = 'D';
  panel.repaint();
```

6.
Name:
public void instruct(){

method:
```
 JOptionPane.showMessageDialog(this, "1. Shift+R - Restart \n2. J/Z - Increase Snake speed
\n3. K/X - Decrease Snake speed \n4. LEFT/A - Turn left \n5. RIGHT/D - Turn right \n6. UP/W -
go up \n7. DOWN/S - go down \n8. Shift+I - to view Instructions", "Instructions",1);
```

7.
Name:
public void paintComponent(Graphics g)

members:
```
  super.paintComponent(g);
  Graphics2D g2 = (Graphics2D)g;
  score = vector.size()-5;
  g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
  int red = 30+(int)(Math.random()*220);
int green = 30+(int)(Math.random()*220);
 int black = 30+(int)(Math.random()*220);
```

methods:
```
  g2.setColor(Color.white);
  g2.drawRect(600, 0, 199, 550);
  g2.setColor(Color.red);ange a certain color
  g2.setFont(new Font("Arial", 1, 50));
  g2.drawString("Snake", 630, 80);
  g2.setFont(new Font("Arial", 1, 20));
  g2.drawString("By Mason Schaefer", 610, 160);
  g2.setFont(new Font("Arial", 1, 20));
  g2.drawString("Score: "+score,650, 230);
  g2.setFont(new Font("Arial", 1, 20));
  g2.drawString("High Score:  ", 650, 200);
  g2.fillRect(0, 0, dim.width-200, dim.height);
```

```
  g2.setColor(Color.white);
  g2.fillRect(10, 10, dim.width-220, dim.height-65);
  g2.setColor(new Color(red, green, blue, black));
  g2.fillOval(foodx, foody, 10, 10);
  g2.setColor(Color.red);
  if(gameover){
    g2.setColor(Color.green); //prints texts in green
   g2.setFont(new Font("Monotype Corsiva", 1, 120));
    g2.setColor(Color.green); //prints in green "you suck"
   g2.drawString("YOU SUCK!", 60, 260);
    g2.setColor(Color.green); //type of text
   g2.setFont(new Font("typewriter",1,14));
    g2.setColor(Color.green); //prints the phrase get a job
     g2.drawString("Get A Job!", 260,300 );
    g2.setColor(Color.green); //tells who its made by
   g2.drawString("   Proudly Made By", 220, 330);
    g2.setColor(Color.green); //tells the users name
   g2.drawString("Mason Schaefer", 255, 370);

 else{
   g2.fillOval(vector.get(0).centerx, vector.get(0).centery, vector.get(0).radius, vector.get(0).ra-
dius);
   for(int i=1; i < vector.size(); i++){
    g2.fillOval(vector.get(i).centerx, vector.get(i).centery, vector.get(i).radius, vector.get(i).radius);
    vector.get(i).pcenterx = vector.get(i).centerx; //piece of snake body
    vector.get(i).pcentery = vector.get(i).centery; //piece of snake body
    vector.get(i).centerx = vector.get(i-1).pcenterx; //piece of snake body
    vector.get(i).centery = vector.get(i-1).pcentery; //piece of snake body


8.
Name
wrongWay(int x, int y){


members: if(x < 10 || x > dim.width-220 || y < 10 || y > dim.height-61)
 else if(x==foodx && y==foody){
   Snake_mode modes = new Snake_mode(); //establishing a new snake mode
   modes.centerx = vector.get(vector.size()-1).pcenterx;
   modes.centery = vector.get(vector.size()-1).pcentery;
   vector.add(modes);//addes modes
   foodx = 20+(int)(Math.random()*56)*10;
   foody = 20+(int)(Math.random()*52)*10;
   for(int i=1; i < vector.size(); i++){
   if(x == vector.get(i).centerx && y == vector.get(i).centery){   gameover = true;
```

Methods:
```
try{Thread.sleep(1000);}catch(Exception e){}
   repaint();
   gameover = true;
    try{Thread.sleep(1000);}catch(Exception e){}
    repaint();
```

9.
Name:
public void run

members:
```
  while (true){
   switch(dir){
    case 'L':/    vector.get(0).centerx = (vector.get(0).centerx-inc);
     vector.get(0).pcenterx = vector.get(0).centerx;
     wrongWay(vector.get(0).centerx, vector.get(0).centery);
     repaint();
     break;
    case 'R':        vector.get(0).centerx = (vector.get(0).centerx+inc);
     vector.get(0).pcenterx = vector.get(0).centerx;
     wrongWay(vector.get(0).centerx, vector.get(0).centery);
     repaint();
     break;
    case 'U':      vector.get(0).pcentery = vector.get(0).centery;
     wrongWay(vector.get(0).centerx, vector.get(0).centery);
     repaint();
     break;
    case 'D':     vector.get(0).centery = (vector.get(0).centery+inc);
     vector.get(0).pcentery = vector.get(0).centery;
     wrongWay(vector.get(0).centerx, vector.get(0).centery);
```

methods:
```
repaint();
     break;
   try{
    Thread.sleep(time)
   catch(Exception e){}
```

10.
Name:
keyPress(KeyEvent e){

members:

```
 if(e.getKeyCode() == KeyEvent.VK_DOWN ||e.getKeyCode() == KeyEvent.VK_S){
   if(dir != 'U')
    dir = 'D';

  else if(e.getKeyCode() == KeyEvent.VK_UP || e.getKeyCode() == KeyEvent.VK_W){
   if(dir != 'D')    dir = 'U';
  }
  else if(e.getKeyCode() == KeyEvent.VK_LEFT || e.getKeyCode() == KeyEvent.VK_A){
   if(dir != 'R')    dir = 'L';
  }
  else if(e.getKeyCode() == KeyEvent.VK_RIGHT || e.getKeyCode() == KeyEvent.VK_D){
   if(dir != 'L')    dir = 'R';
  else if(e.getKeyCode() == KeyEvent.VK_J || e.getKeyCode() == KeyEvent.VK_Z){
   time--;  }
  else if(e.getKeyCode() == KeyEvent.VK_K || e.getKeyCode() == KeyEvent.VK_X){
   time++;
  else if(e.getKeyCode() == KeyEvent.VK_ESCAPE)
  else if(e.getKeyCode() == KeyEvent.VK_SPACE)
if(gamepause){
```

methods:

```
 t.suspend();
gamepause = false
   else{
    t.resume();
   gamepause = true
System.exit(0);
```