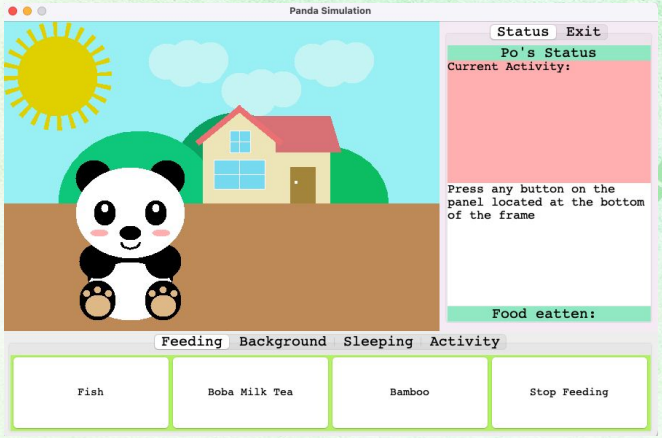
Overview

Login (immediately after running the program)

Simulation (after pressing the Start button

Day Time

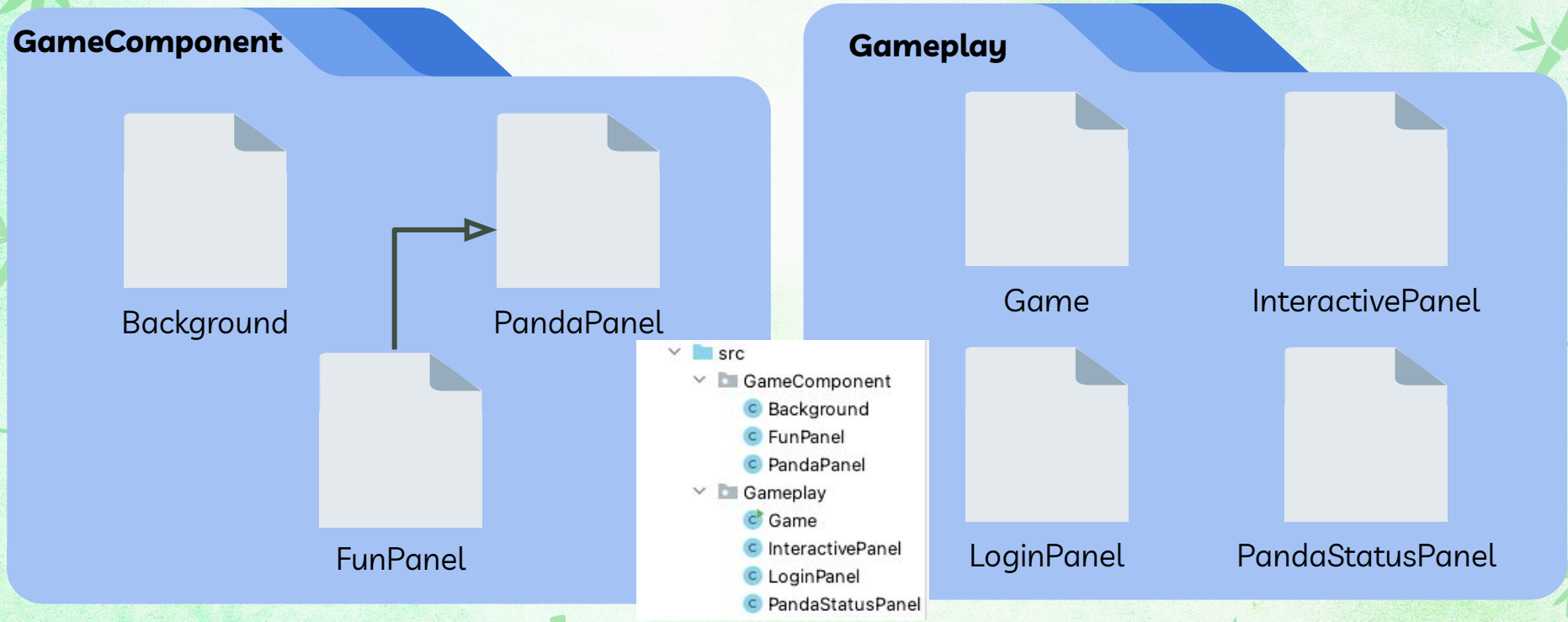


Night Time


```
static Timer timeofdayTimer = new Timer( delay: 1000, new DayandNightHandler());
```

Checklist

1. 3 Classes, at least 1 derived from other **3.** At least 2 packages



Responsible for drawing Graphics of the simulation

Responsible for user interaction and Displaying the simulation

Checklist

2. Implicit and Explicit Casting

```
static JPanel graphicsDrawnPanel;  graphicsDrawnPanel = new FunPanel();
```

```
Graphics2D g2 = (Graphics2D) g;
```
// g is Graphics. Graphics2D is subclass of Graphics (Explicit)

4. Keywords public, protected, private, this, and super

```
private void drawFoodBox(Graphics g){...}

private void drawFish(Graphics g){...}
```

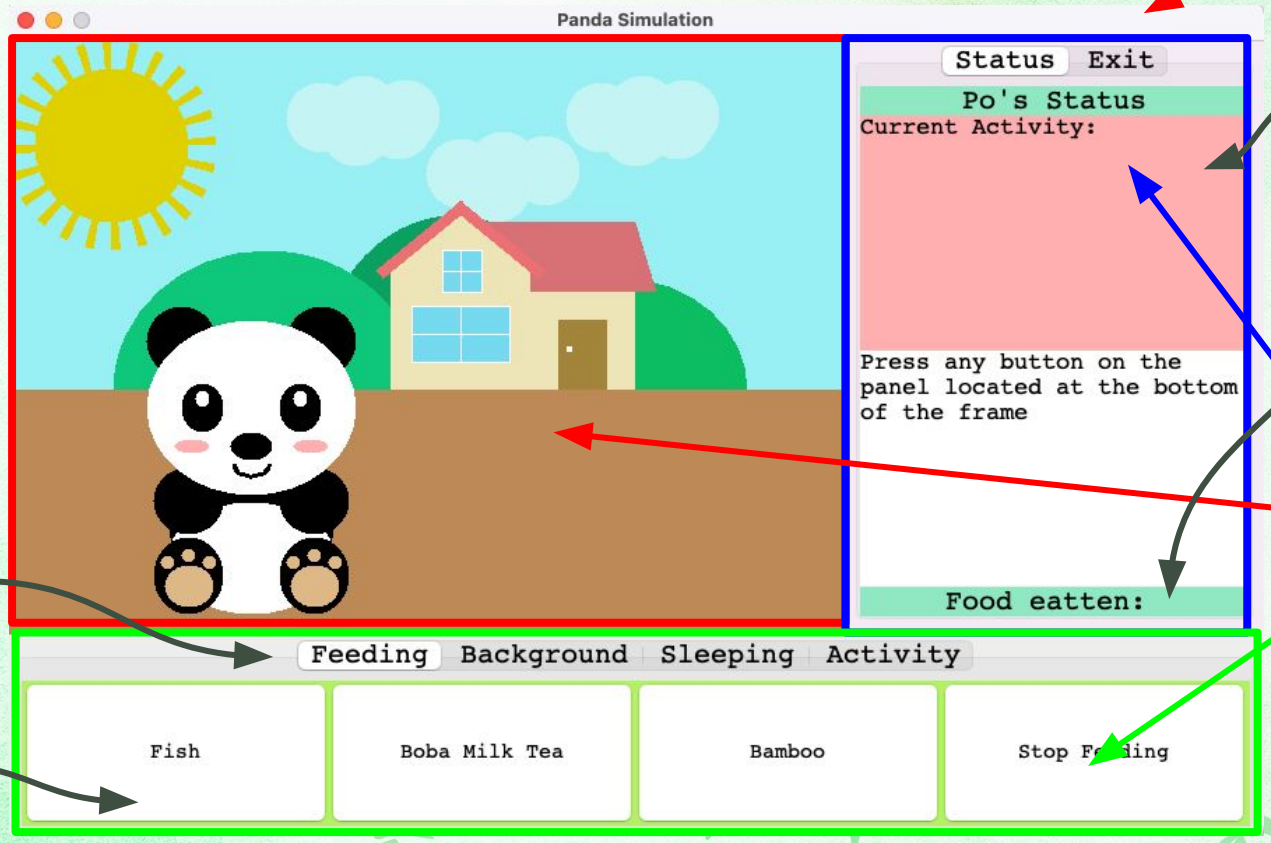
```
public static int xPos = 100, yPos = 160;

protected int yFace = 200;
protected int yDiff = 65;
protected int yBody = yFace + yDiff;
```

```
public PandaPanel(){
    this.setLayout(null);
    this.setBounds( x: 0, y: 0, width: 600, height: 425);
    this.setPreferredSize(new Dimension( width: 600, height: 425));
}
```

```
public void paintComponent(Graphics g){
    Background.drawBackground(g);
    // Draw Panda
    super.paintComponent(g);

    switch(foodtype){...}
```

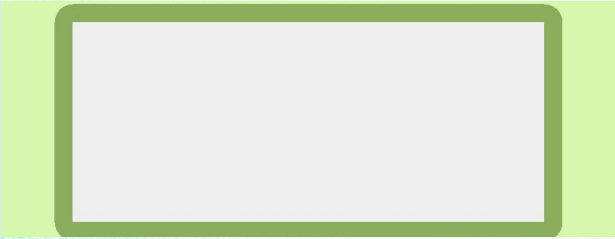
Checklist

5. 3 Containers **6.** 4 distinct components



Checklist

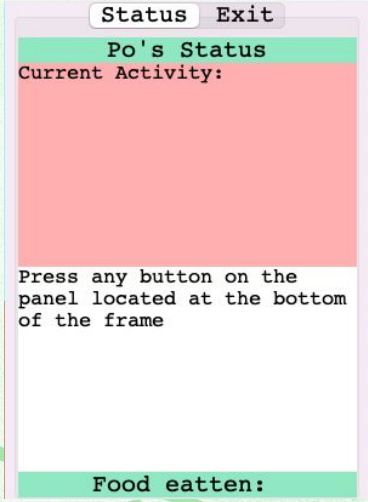
7. Two distinct layout managers



```
CardLayout cl = new CardLayout();

JPanel card = new JPanel(cl);
JPanel namePanel = new JPanel();
```


```
sleepingPanel.setLayout(new GridLayout( rows: 2, cols: 2));
```

```
statusPanel.setLayout(new BorderLayout());
```



Checklist

8. 2 distinct geometric shapes drawn by paintComponent method

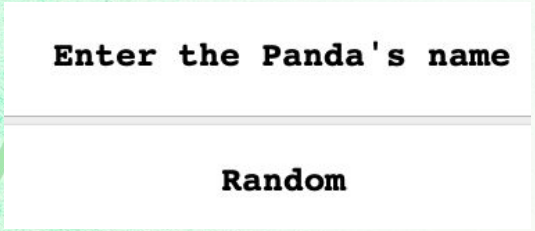
9. 2 different colors

```
private void drawBomb(Graphics g){
    // Bomb Body
    g.setColor(new Color( r: 138, g: 117, b: 117));
    int[] x = {425,350,350,425,500,500};
    int[] y = {320,345,420,375,420,345};
    g.fillPolygon(x,y, nPoints: 6);
    g.setColor(new Color( r: 107, g: 99, b: 99));
    g.fillRoundRect( x: 350, y: 135, width: 145, height
    g.setColor(new Color( r: 173, g: 173, b: 173));
    g.fillRoundRect( x: 350, y: 200, width: 145, height
    // Symbol
    g.setColor(Color.black);
    g.fillOval( x: 395, y: 208, width: 25, height: 25);
    g.fillOval( x: 428, y: 208, width: 25, height: 25);
    g.setColor(Color.white);
    g.fillOval( x: 400, y: 218, width: 50, height: 40);
```


```
public void paintComponent(Graphics g){
    Background.drawBackground(g);
    // Draw Panda
    super.paintComponent(g);

    switch(foodtype){...}
```

Checklist

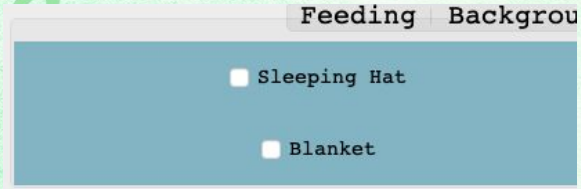
10. 3 sources fire events, at least 1 listener **11.** 3 types of events **12.** 1 static method in overridden method




```
pet_nameTF.addActionListener(new PetNameListener());
random_nameBT.addActionListener(new PetNameListener());
```

```
public void actionPerformed(ActionEvent e){
    // Random name for the Panda
    if(e.getSource() == random_nameBT){
        String[] vowel = {"a","e","i","o","u"};

        int first = (int) Math.floor(Math.random()*26+65);
        String firstChar = Character.toString((char) first);

        int second = (int) Math.floor(Math.random()*26+97);
        String endChar = Character.toString((char) second);

        panda_name = firstChar + vowel[(int)(Math.random()*5)] + endChar;
```

```
confirmBT.addActionListener(new ActionListener() {...});

confirmBT.addMouseListener(new mouseonExitBTListener());
```




```
hatCB.addItemListener(new SleepingAccessoriesListener());
eyepadCB.addItemListener(new SleepingAccessoriesListener());
blanketCB.addItemListener(new SleepingAccessoriesListener())
pillowCB.addItemListener(new SleepingAccessoriesListener());
```

Checklist

13. A timer is created and used

```
static Timer timeofdayTimer = new Timer( delay: 1000, new DayandNightHandler());
```

```
timeofdayTimer.start();
```

Thank you

Thank you

Thank you

Panda Simulation

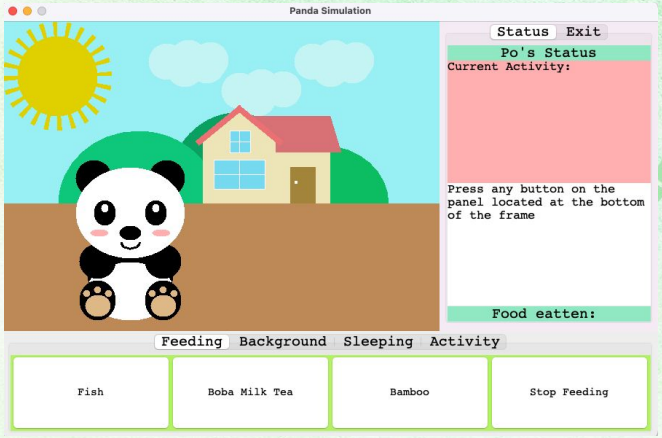
By: Chanon Charuchinda
ID: 6322770692

Overview

Login (immediately after running the program)

Simulation (after pressing the Start button

Day Time

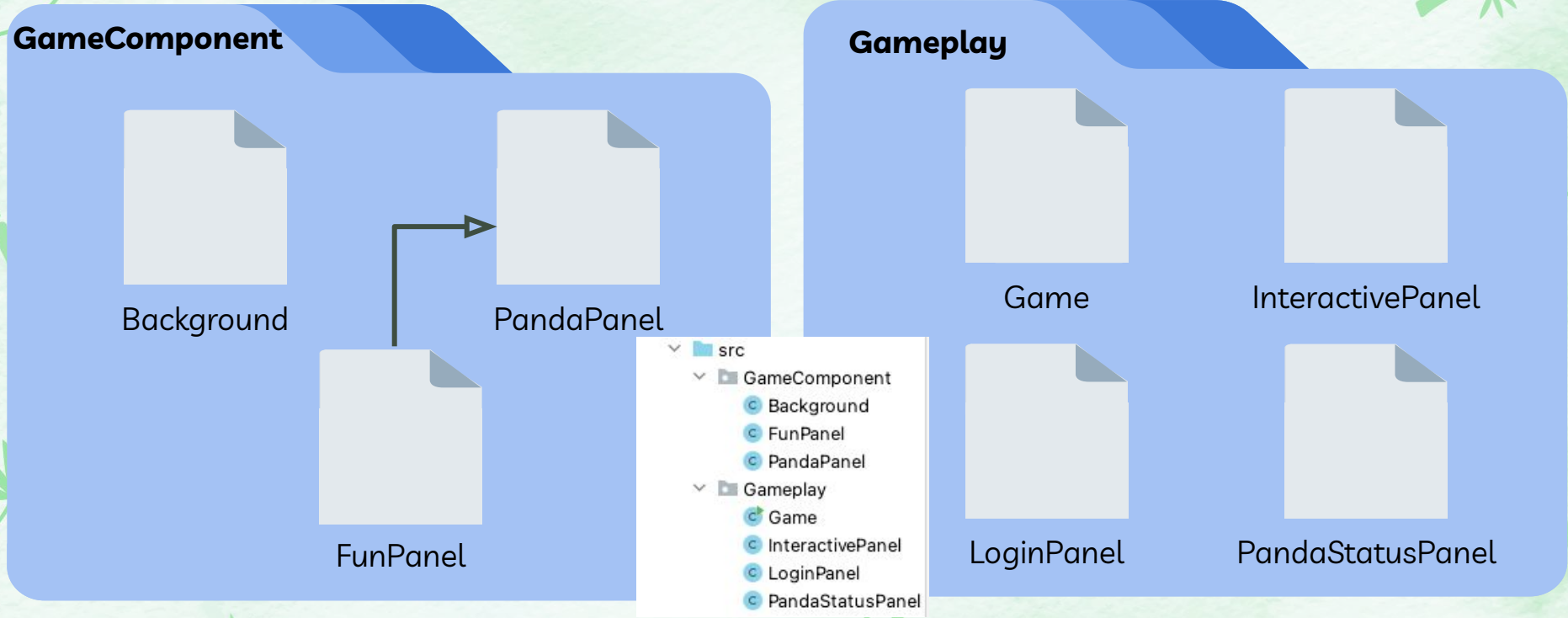


Night Time


```
static Timer timeofdayTimer = new Timer( delay: 1000, new DayandNightHandler());
```

Packages and Classes

1. 3 Classes, at least 1 derived from other **3.** At least 2 packages



Responsible for drawing Graphics of the simulation

Responsible for user interaction and Displaying the simulation

```
public LoginPanel(){
    // Setting Up login panel
    this.setLayout(null);
```

Login Page

Timer (Control the background color strips, making it keep changing color)

JFrame

JPanel (CarLayout, contains JPanel for Name, Information, and Exit)

JButton Name

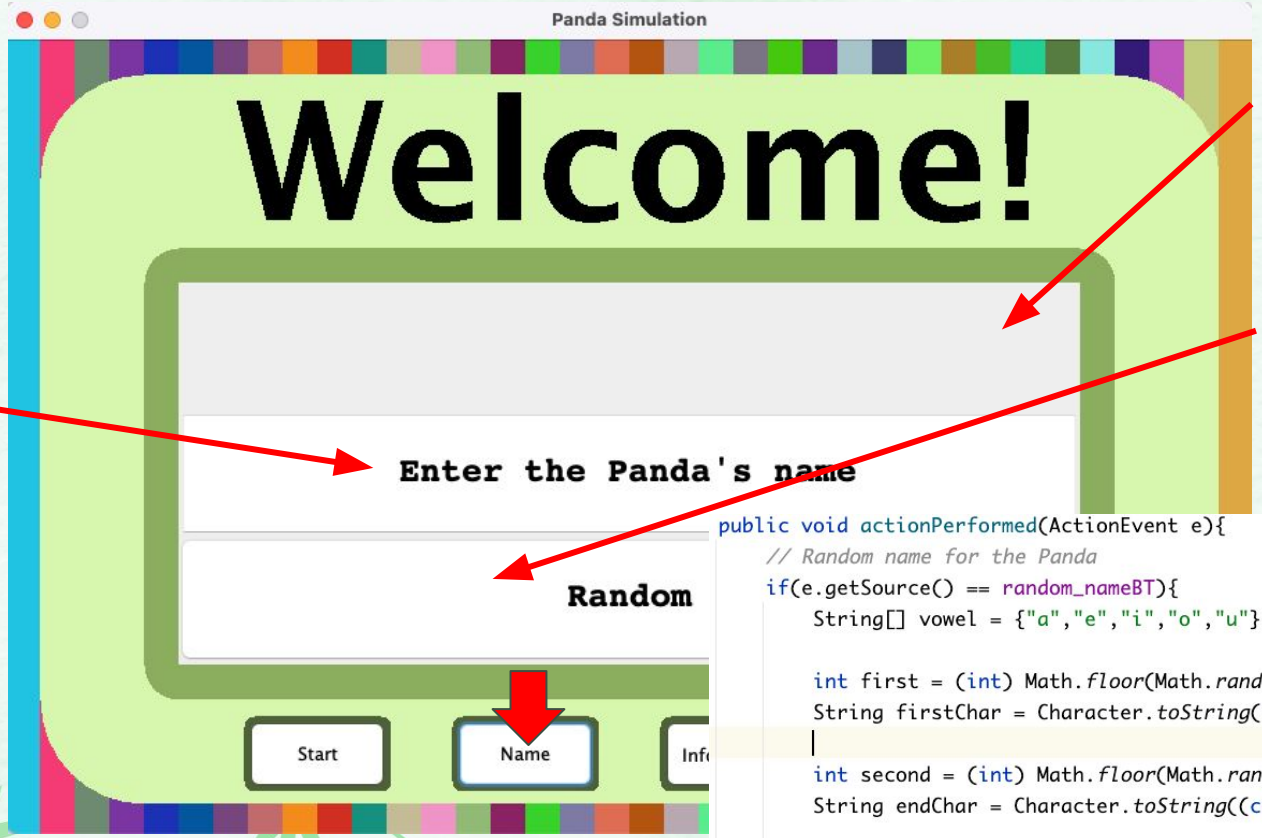
JButton Information

JButton Start

JButton Exit

After pressing the "Name" button

Name



JLabel (Display Panda's name)

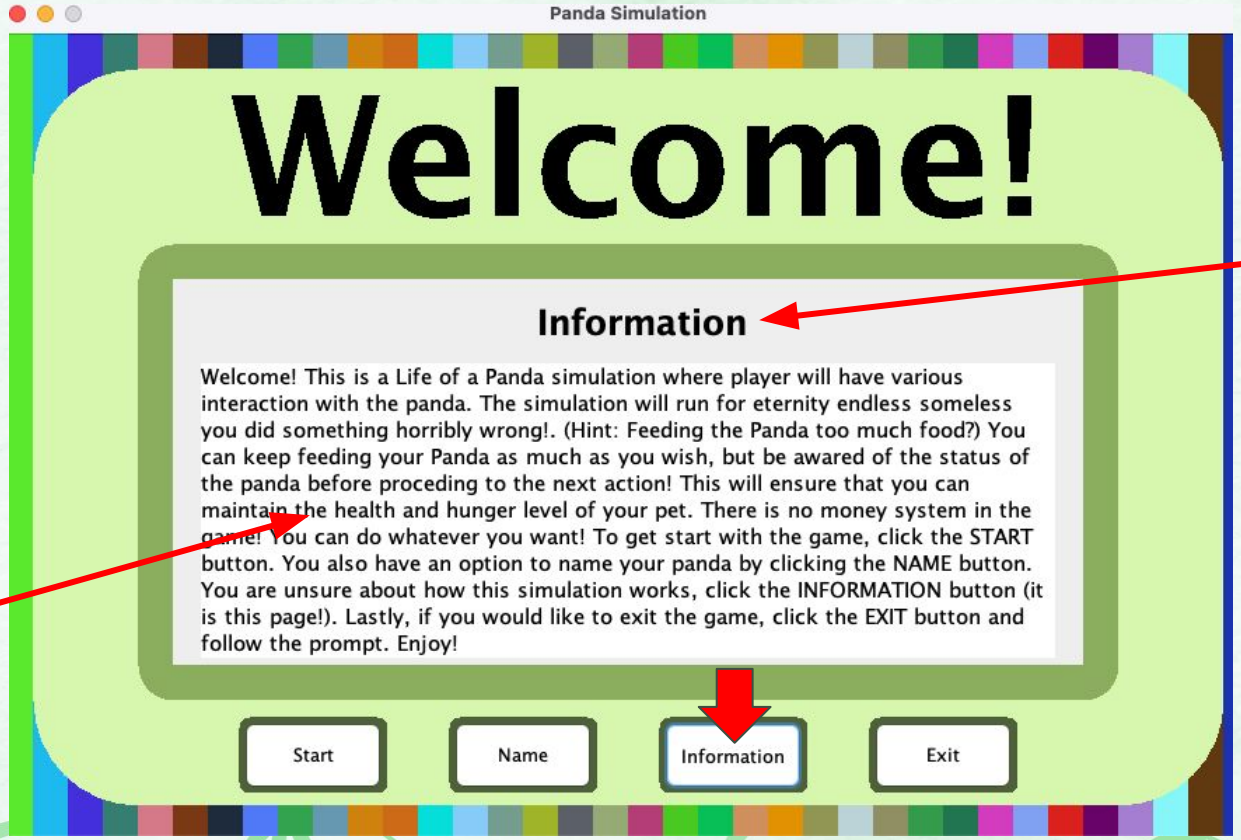
JTextField (For entering your Panda's name)

JButton Random (Randomize a name)

```
public void actionPerformed(ActionEvent e){
    // Random name for the Panda
    if(e.getSource() == random_nameBT){
        String[] vowel = {"a","e","i","o","u"};

        int first = (int) Math.floor(Math.random()*26+65);
        String firstChar = Character.toString((char) first);

        int second = (int) Math.floor(Math.random()*26+97);
        String endChar = Character.toString((char) second);

        panda_name = firstChar + vowel[(int)(Math.random()*5)] + endChar;
```

After pressing the “Information” button

Information



JLabel

JTextArea (Information about the game)

After pressing the “Exit” button

Exit

```
confirmBT.addActionListener(new ActionListener() {...});

confirmBT.addMouseListener(new mouseonExitBTListener());
```

JButton Confirm

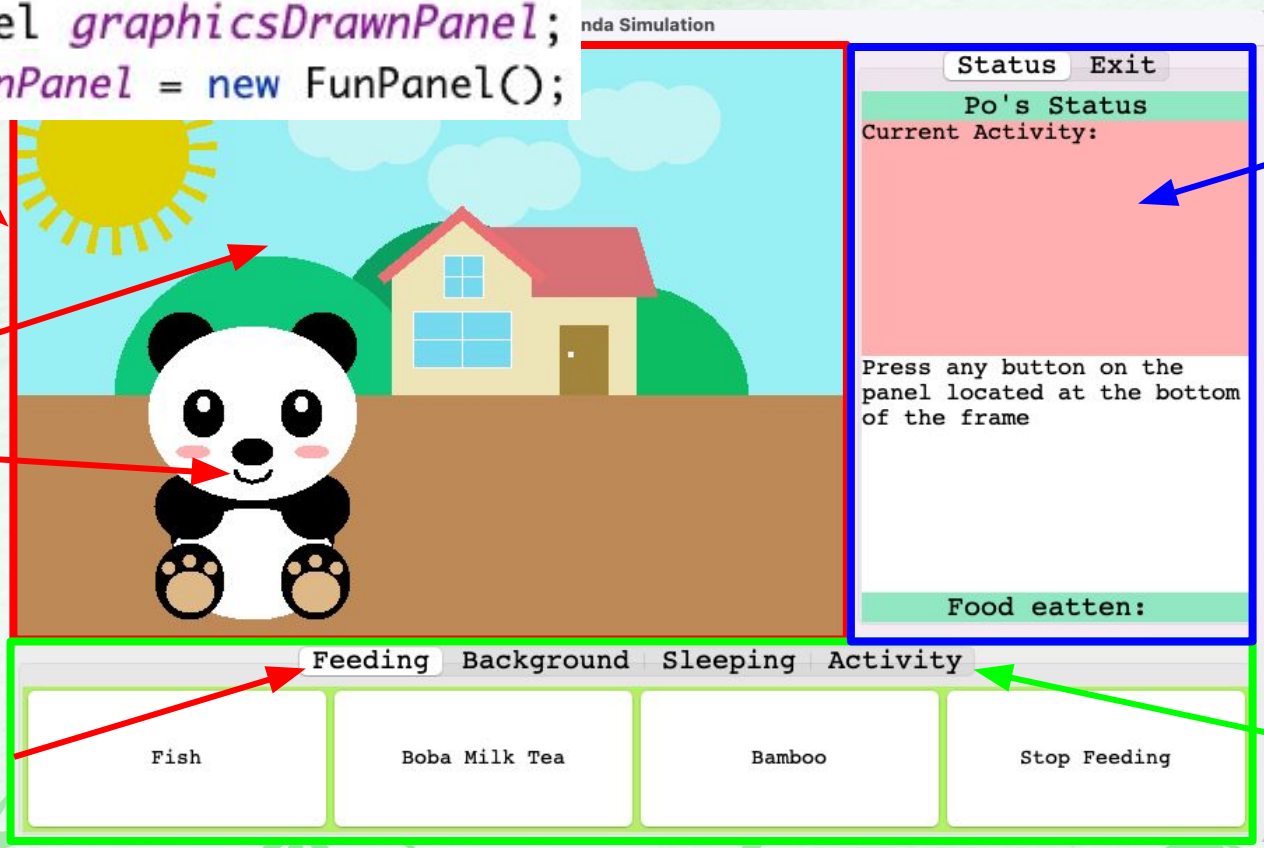
JLabel

JPanel
(BorderLayout)

After pressing the "Start" button

Simulation

```
static JPanel graphicsDrawnPanel;
graphicsDrawnPanel = new FunPanel();
```



Background

FunPanel

JPanel (for displaying Status of the game)

JTabbedPane (registered to ChangeListener)

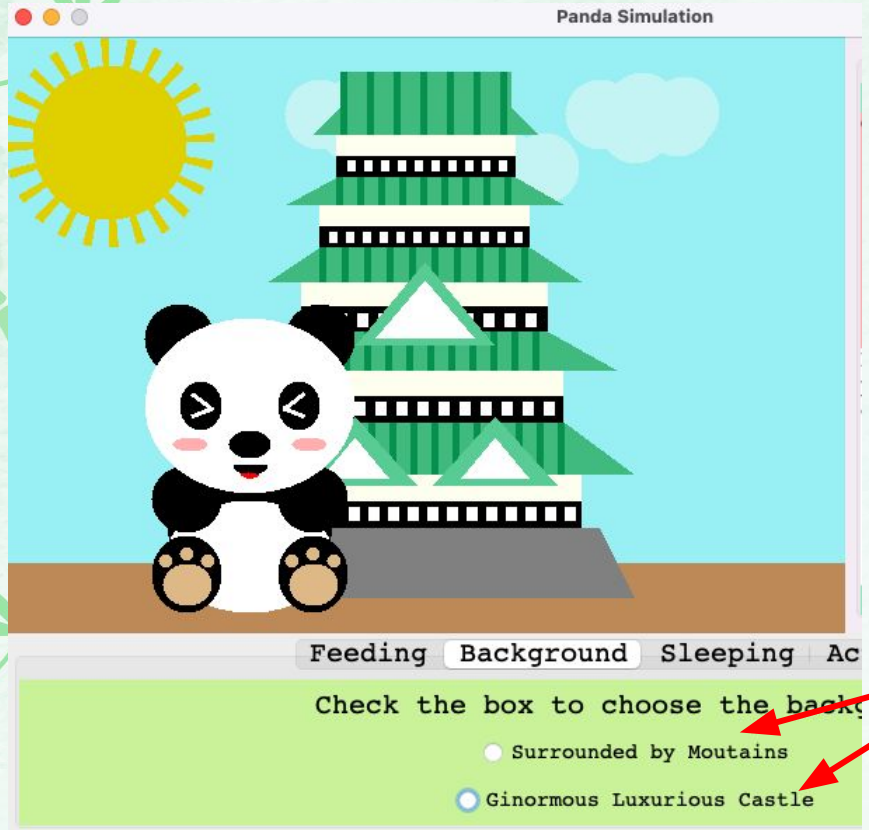
JPanel (for user interaction with the Panda)

Simulation

JButton
(registered to ActionListener)

Simulation

When selecting Background Tab



Default Background

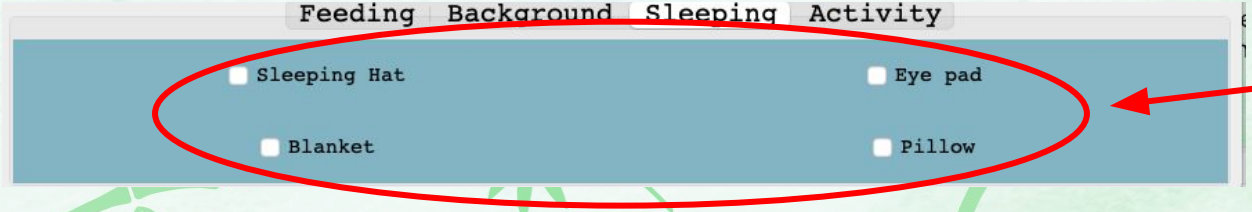
JRadioButton
(registered to ItemListener)

During the night time

Simulation

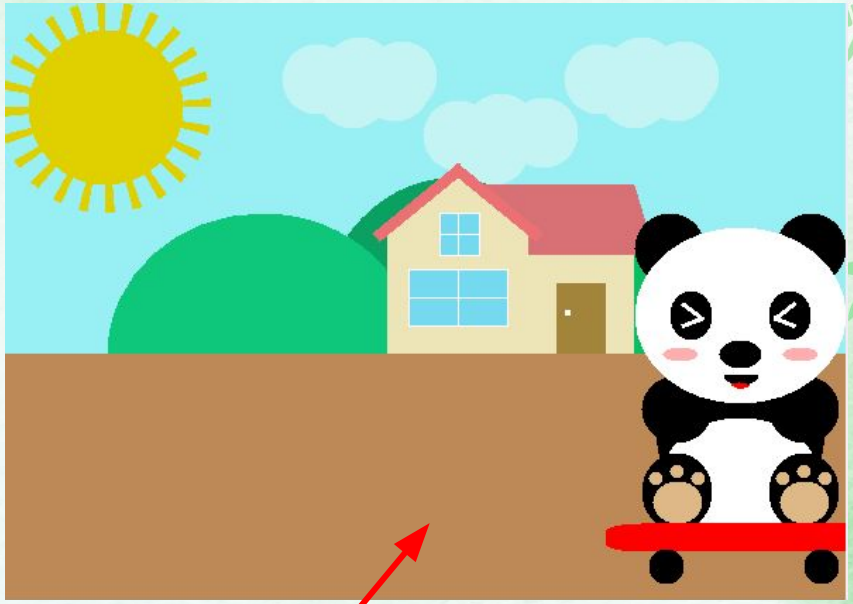
Default (not checking any JCheckBox)

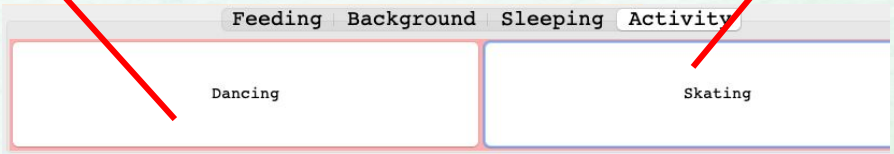
(Checking every JCheckBox)



JCheckBox
(registered to
ItemListener)

Simulation





JButton
(registered to
ActionListener)

Simulation

Food eatten: 17

Food eatten: -3

Simulation


Panda Simulation
Status
Exit
Exit?
Exit
The End
R.I.P
The simulation has ended. Either you feed the panda too much food OR you have taken away too much food!
Feeding
Background
Sleeping
Activity
Fish
Boba Milk Tea
Bamboo
Stop Feeding

Checklist

2. Implicit and Explicit Casting

```
static JPanel graphicsDrawnPanel;  graphicsDrawnPanel = new FunPanel();
```

```
Graphics2D g2 = (Graphics2D) g;
```
// g is Graphics. Graphics2D is subclass of Graphics (Explicit)

4. Keywords public, protected, private, this, and super

```
private void drawFoodBox(Graphics g){...}

private void drawFish(Graphics g){...}
```

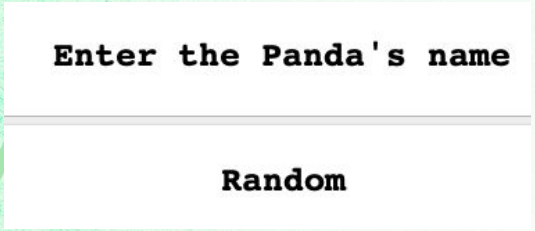
```
public static int xPos = 100, yPos = 160;

protected int yFace = 200;
protected int yDiff = 65;
protected int yBody = yFace + yDiff;
```

```
public PandaPanel(){
    this.setLayout(null);
    this.setBounds( x: 0, y: 0, width: 600, height: 425);
    this.setPreferredSize(new Dimension( width: 600, height: 425));
}
```

```
public void paintComponent(Graphics g){
    Background.drawBackground(g);
    // Draw Panda
    super.paintComponent(g);

    switch(foodtype){...}
```

Checklist

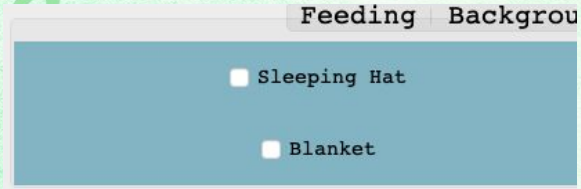
10. 3 sources fire events, at least 1 listener **11.** 3 types of events **12.** 1 static method in overridden method




```
pet_nameTF.addActionListener(new PetNameListener());
random_nameBT.addActionListener(new PetNameListener());
```

```
public void actionPerformed(ActionEvent e){
    // Random name for the Panda
    if(e.getSource() == random_nameBT){
        String[] vowel = {"a","e","i","o","u"};

        int first = (int) Math.floor(Math.random()*26+65);
        String firstChar = Character.toString((char) first);

        int second = (int) Math.floor(Math.random()*26+97);
        String endChar = Character.toString((char) second);

        panda_name = firstChar + vowel[(int)(Math.random()*5)] + endChar;
```

```
confirmBT.addActionListener(new ActionListener() {...});

confirmBT.addMouseListener(new mouseonExitBTListener());
```




```
hatCB.addItemListener(new SleepingAccessoriesListener());
eyepadCB.addItemListener(new SleepingAccessoriesListener());
blanketCB.addItemListener(new SleepingAccessoriesListener())
pillowCB.addItemListener(new SleepingAccessoriesListener());
```

Thank you

Thank you

Thanks

Do you have any questions?

youremail@freepik.com
+91 620 421 838
yourcompany.com

Icon pack