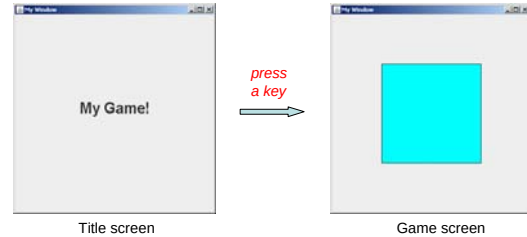


Lecture 28



Assignment
Discussion

Showing a title screen



```
public class MyJPanel extends JPanel implements KeyListener {
    private boolean showTitleScreen;
    public MyJPanel() {
        showTitleScreen = true;
    }
    public void keyPressed(KeyEvent e) {
        showTitleScreen = false;
        repaint();
    }
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        if (showTitleScreen) {
            ...
        } else {
            ...
        }
    }
}
```

Annotations:

- declare instance variables
- initialise instance variables
- update instance variables
- draw based on instance variables

Using Fonts



Using Fonts

```
public void paintComponent(Graphics g) {
    super.paintComponent(g);

    g.setFont( ... );
    g.drawString("My Game!", 100, 100);
}
```

need to pass a **Font** object to the `setFont()` method as a parameter

Creating a Font object

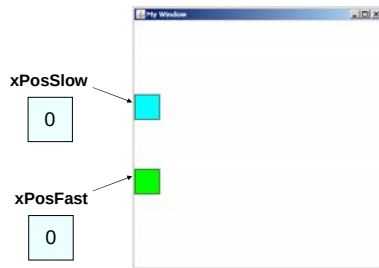
```
new Font( "SansSerif" , Font.BOLD , 25 )
```

logical font name font style font size

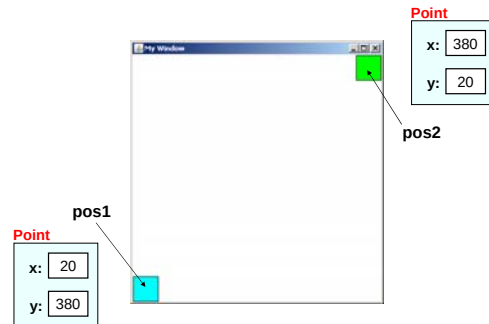
SansSerif
Serif
Monospaced

Font.BOLD
Font.PLAIN
Font.ITALIC
Font.BOLD + Font.ITALIC

Different speeds



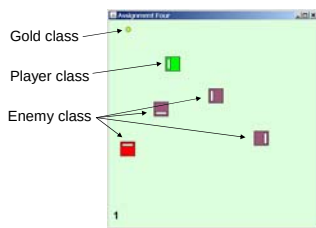
Collision testing



Classes for game objects

- You should think of defining a class for each distinct type of game object
- Using suitable classes will **simplify** the JPanel class

```
public class A4JPanel ... {
    private Gold gold;
    private Player player;
    private Enemy[] enemies;
    ...
}
```



```
public class MyJPanel extends JPanel implements KeyListener {
```

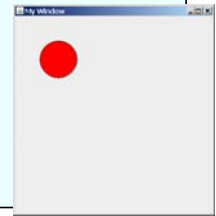
```
    private Point pos;
    private int size;
```

```
    public MyJPanel() {
        pos = new Point(50, 50);
        size = 75;
        addKeyListener(this);
    }
```

```
    public void keyPressed(KeyEvent e) {
        pos.x += size;
        pos.y += size;
        repaint();
    }
```

```
    public void paintComponent(Graphics g) {
        super.paintComponent(g);

        g.setColor(Color.red);
        g.fillOval(pos.x, pos.y, size, size);
        g.setColor(Color.black);
        g.drawOval(pos.x, pos.y, size, size);
    }
}
```



```
public class MyJPanel extends JPanel implements KeyListener {
```

```
    private Point pos;
    private int size;
```

```
    public MyJPanel() {
        pos = new Point(50, 50);
        size = 75;
        addKeyListener(this);
    }
```

```
    public void keyPressed(KeyEvent e) {
        pos.x += size;
        pos.y += size;
        repaint();
    }
```

```
    public void paintComponent(Graphics g) {
        super.paintComponent(g);

        g.setColor(Color.red);
        g.fillOval(pos.x, pos.y, size, size);
        g.setColor(Color.black);
        g.drawOval(pos.x, pos.y, size, size);
    }
}
```

```
public class GameObject {
    private Point pos;
    private int size;
```

```
public class MyJPanel extends JPanel implements KeyListener {
```

```
    private GameObject go;
```

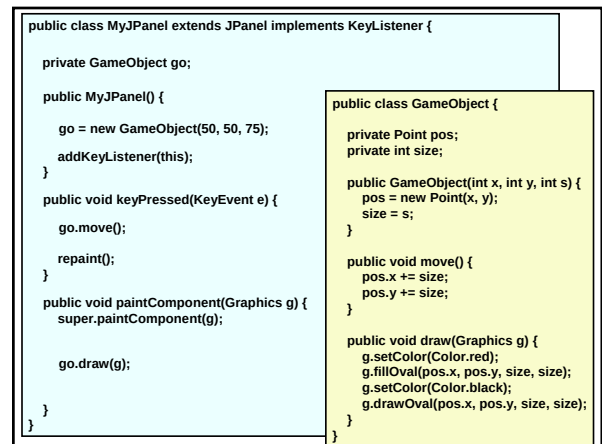
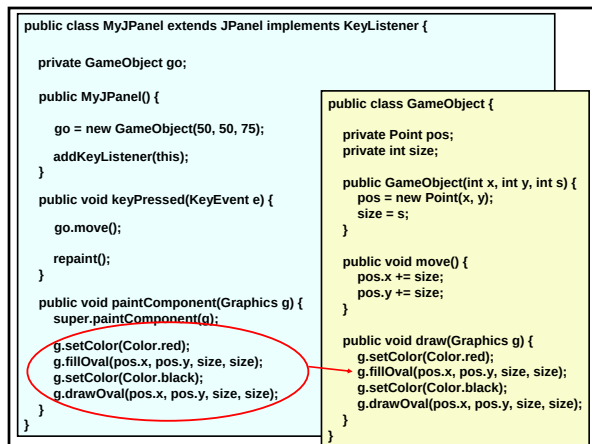
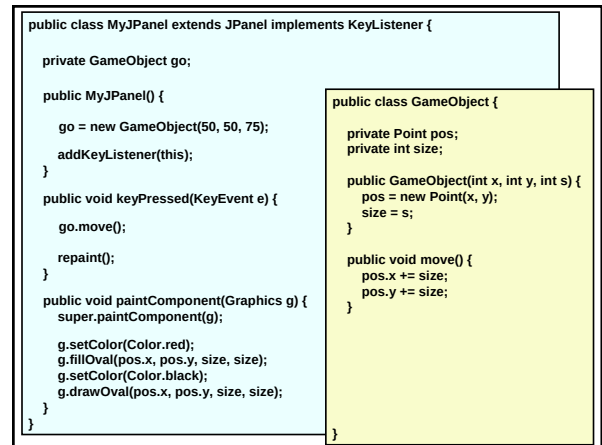
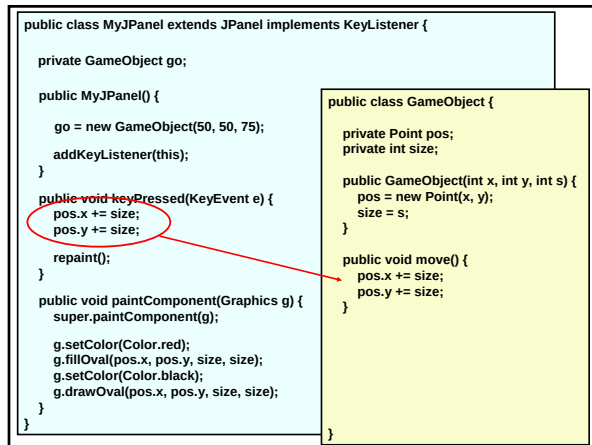
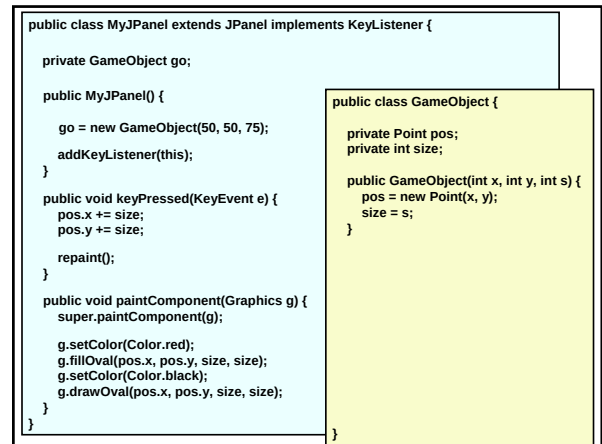
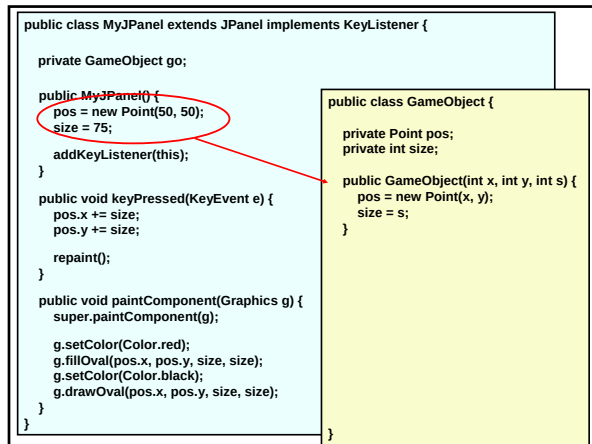
```
    public MyJPanel() {
        pos = new Point(50, 50);
        size = 75;
        addKeyListener(this);
    }
```

```
    public void keyPressed(KeyEvent e) {
        pos.x += size;
        pos.y += size;
        repaint();
    }
```

```
    public void paintComponent(Graphics g) {
        super.paintComponent(g);

        g.setColor(Color.red);
        g.fillOval(pos.x, pos.y, size, size);
        g.setColor(Color.black);
        g.drawOval(pos.x, pos.y, size, size);
    }
}
```

```
public class GameObject {
    private Point pos;
    private int size;
```



Who is this?

