

**Computer Science Department
presents**



**Pumpkin Carving
Contest 2008**

**Thursday, October 16, 2008
6:00 pm — 222 Clapp**

All majors, minors, and prospective Computer Science majors/minors are welcome and encouraged to come to this fun event! Carve your own pumpkin, then vote on whose pumpkin is best!

Prizes for first, second, and third places!
Please RSVP to wqueiros@mtholyoke.edu

1

Graphical Objects

October 9, 2008

2

Thursday, October 9, 2008

Creating Graphical Objects

FilledRect, FramedRect, FilledOval, FramedOval

- Specify left, top, width, height, canvas
- `new FilledRect (50, 200, 300, 100, canvas);`

FilledRect



FilledOval

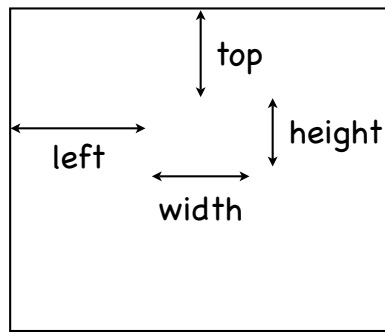


3

Creating Graphical Objects

FilledRect, FramedRect, FilledOval, FramedOval

- Specify left, top, width, height, canvas
- `new FilledRect (50, 200, 150, 100, canvas);`

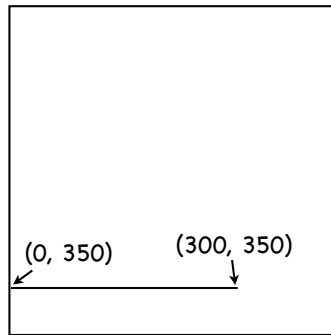


4

Creating Graphical Objects

Line

- Specify 2 endpoints as x,y values, canvas
- `new Line (0, 350, 300, 350, canvas);`

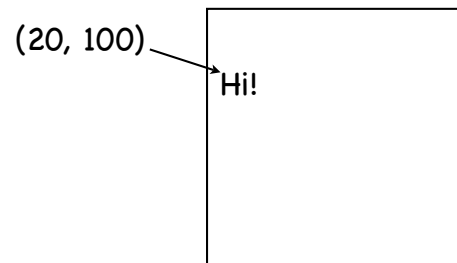


5

Creating Graphical Objects

Text

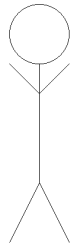
- Specify String, left, top, canvas
- `new Text ("Hi!", 20, 100, canvas);`



6

Constructing a Stick Figure

```
public StickFigure (DrawingCanvas figureCanvas) {  
    new FramedOval (100, 100, 100, 100, figureCanvas);  
    new Line (150, 200, 150, 400, figureCanvas);  
    new Line (150, 250, 100, 200, figureCanvas);  
    new Line (150, 250, 200, 200, figureCanvas);  
    new Line (150, 400, 100, 500, figureCanvas);  
    new Line (150, 400, 200, 500, figureCanvas);  
}
```



7

Too Many Numbers!

```
new FramedOval (100, 100, 100, 100, figureCanvas);  
new Line (150, 200, 150, 400, figureCanvas);  
new Line (150, 250, 100, 200, figureCanvas);  
new Line (150, 250, 200, 200, figureCanvas);  
new Line (150, 400, 100, 500, figureCanvas);  
new Line (150, 400, 200, 500, figureCanvas);
```

- ⦿ What do all the numbers mean?
- ⦿ Is it important that some numbers are the same or is it coincidental?

8

Too Many Numbers!

```
new FramedOval (100, 100, 100, 100, figureCanvas);
```

vs.

```
new FramedOval (HEAD_LEFT, HEAD_TOP,  
                HEAD_SIZE, HEAD_SIZE,  
                figureCanvas);
```

- ◉ What do all the numbers mean?
- ◉ Is it important that some numbers are the same or coincidental?

9

Named Constants

```
private static final int HEAD_SIZE = 100;
```

- ◉ "static final" means constant
- ◉ Naming convention: all capital letters, words separated by _
- ◉ Useful to define constants in terms of each other

```
private static final int BODY_SIZE = HEAD_SIZE * 2;
```

10

Drawing a Stick Figure

```
private static final int HEAD_SIZE = 100;
private static final int BODY_SIZE = HEAD_SIZE * 2;

private static final int HEAD_LEFT = 100;
private static final int HEAD_TOP = 100;
private static final int BODY_LEFT = HEAD_LEFT + (HEAD_SIZE / 2);
private static final int BODY_TOP = HEAD_TOP + HEAD_SIZE;

public StickFigure (DrawingCanvas figureCanvas) {
    new FramedOval (HEAD_LEFT, HEAD_TOP,
        HEAD_SIZE, HEAD_SIZE, figureCanvas);
    new Line (BODY_LEFT, BODY_TOP,
        BODY_LEFT, BODY_TOP + BODY_SIZE, figureCanvas);
    ...
}
```

11

Stick Figure Class

```
public class StickFigure {
    private static final int HEAD_SIZE = 100;
    private static final int BODY_SIZE = HEAD_SIZE * 2;
    ...

    private static final int HEAD_LEFT = 100;
    private static final int HEAD_TOP = 100;
    private static final int BODY_LEFT = HEAD_LEFT + (HEAD_SIZE / 2);
    private static final int BODY_TOP = HEAD_TOP + HEAD_SIZE;
    ...

    public StickFigure (DrawingCanvas figureCanvas) {
        new FramedOval (HEAD_LEFT, HEAD_TOP,
            HEAD_SIZE, HEAD_SIZE, figureCanvas);
        new Line (BODY_LEFT, BODY_TOP,
            BODY_LEFT, BODY_TOP + BODY_SIZE, figureCanvas);
        ...
    }
}
```

12

Constructing a Stick Figure

- To construct a stick figure, you must call the constructor in the StickFigure class, defined as:

```
public StickFigure(DrawingCanvas figureCanvas)
{
  ...
}
```

- To call the constructor, say:

```
new StickFigure(???)
```

↑
means "call a
constructor"

13

begin Method

- Like Alice's "when the world starts" event
- Called when program first starts running
- Used to create the initial scene
- Let's call the StickFigure constructor from the begin method of our program:

```
public class DrawAStickFigure extends WindowController
{
  public void begin() {
    new StickFigure (canvas);
  }
}
```

14

Putting the Pieces Together

```
public class DrawAStickFigure extends WindowController
{
    public void begin() {
        new StickFigure (canvas);
    }
}

public class StickFigure {
    private static final int HEAD_SIZE = 100;
    ...

    public StickFigure (DrawingCanvas figureCanvas) {
        new FramedOval (HEAD_LEFT, HEAD_TOP,
            HEAD_SIZE, HEAD_SIZE, figureCanvas);
    }
    ...
}
}
```

15

Putting the Pieces Together

```
public class DrawAStickFigure extends WindowController
{
    public void begin() {
        new StickFigure (canvas);
    }
}

public class StickFigure {
    private static final int HEAD_SIZE = 100;
    ...

    public StickFigure (DrawingCanvas figureCanvas) {
        new FramedOval (HEAD_LEFT, HEAD_TOP,
            HEAD_SIZE, HEAD_SIZE, figureCanvas);
    }
    ...
}
}
```

1. Program starts

16

Putting the Pieces Together

```
public class DrawAStickFigure extends WindowController
{
    public void begin() {
        new StickFigure (canvas);
    }
}
public class StickFigure {
    private static final int HEAD_SIZE = 100;
    ...

    public StickFigure (DrawingCanvas figureCanvas) {
        new FramedOval (HEAD_LEFT, HEAD_TOP,
            HEAD_SIZE, HEAD_SIZE, figureCanvas);
    }
}
}
```

2. Execute 1st statement of begin method.
Note: canvas is a predefined variable for all classes that extend WindowController

17

Putting the Pieces Together

```
public class DrawAStickFigure extends WindowController
{
    public void begin() {
        new StickFigure (canvas);
    }
}
public class StickFigure {
    private static final int HEAD_SIZE = 100;
    ...

    public StickFigure (DrawingCanvas figureCanvas) {
        new FramedOval (HEAD_LEFT, HEAD_TOP,
            HEAD_SIZE, HEAD_SIZE, figureCanvas);
    }
}
}
```

3. Call the StickFigure constructor. The figureCanvas parameter gets the value of canvas.

18

Putting the Pieces Together

```
public class DrawAStickFigure extends WindowController
{
    public void begin() {
        new StickFigure (canvas); 4. Draw the framed oval for the
    }                                head.
}

public class StickFigure {
    private static final int HEAD_SIZE = 100;
    ...

    public StickFigure (DrawingCanvas figureCanvas) {
        new FramedOval (HEAD_LEFT, HEAD_TOP,
                        HEAD_SIZE, HEAD_SIZE, figureCanvas);
    }
    ...
}
}
```

19

Putting the Pieces Together

```
public class DrawAStickFigure extends WindowController
{
    public void begin() {
        new StickFigure (canvas); 5. Draw the other parts -
    }                                statements not shown here.
}

public class StickFigure {
    private static final int HEAD_SIZE = 100;
    ...

    public StickFigure (DrawingCanvas figureCanvas) {
        new FramedOval (HEAD_LEFT, HEAD_TOP,
                        HEAD_SIZE, HEAD_SIZE, figureCanvas);
    }
    ...
}
}
```

20

Putting the Pieces Together

```
public class DrawAStickFigure extends WindowController
{
    public void begin() {
        new StickFigure (canvas);
    }
}

public class StickFigure {
    private static final int HEAD_SIZE = 100;
    ...

    public StickFigure (DrawingCanvas figureCanvas) {
        new FramedOval (HEAD_LEFT, HEAD_TOP,
            HEAD_SIZE, HEAD_SIZE, figureCanvas);
    }
    ...
}
}
```

6. Return to the begin method. There is nothing more to do in the begin method. At this point the stick figure appears on the screen.

21

Reacting to the mouse

- Let's have the stick figure say hi when the user presses the mouse button down
- Mouse handling events go in the class that extends WindowController

```
public class DrawAStickFigure extends WindowController {
    public void begin () {
        ...
    }

    public void onMousePress (Location point) {
        new Text ("Hi!", 250, 150, canvas);
    }
}
```

22

Mouse Event Handling

Methods

public void onClick (Location point)

- ◀ Called when mouse pressed and released without moving it
- ◀ Following methods have similar declarations

public void onClick (Location point)

onClick	onClick
onMousePress	onMouseRelease
onMouseEnter	onMouseExit
onMouseMove	onMouseDrag