COMP 401
ANIMATION: MVC

Instructor: Prasun Dewan
Prerequisite

- Animation: Loops
SHUTTLE ANIMATION
Basic Architecture

APIlotted Shuttle

setShuttleX(Y)()

animateFromOrigin()

AShuttleAnimator

Main Class

main
public void animateFromOrigin(PlottedShuttle shuttle,  
    int animationStep, int animationPauseTime) {
    int originalX = shuttle.getShuttleX();
    int originalY = shuttle.getShuttleY();
    int curX = 0;
    int curY = 0;
    shuttle.setShuttleX(curX);
    shuttle.setShuttleY(curY);
    animateYFromOrigin(shuttle, animationStep,  
        animationPauseTime, curY, originalY);
    animateXFromOrigin(shuttle, animationStep,  
        animationPauseTime, curX, originalX);
}
protected void animateYFromOrigin(PlottedShuttle shuttle,
        int animationStep, int animationPauseTime,
        int startY, int endY) {
    // make sure we don’t go past final Y position
    while (startY < endY) {
        ThreadSupport.sleep(animationPauseTime);
        startY += animationStep;
        shuttle.setShuttleY(startY);
    }
    // move to destination Y position
    shuttle.setShuttleY(endY);
}
public static void main(String[] args) {
    PlottedShuttle shuttle = new APlottedShuttle(50, 100);
    OEFrame oeFrame = ObjectEditor.edit(shuttle);
    oeFrame.hideMainPanel();
    oeFrame.setSize(450, 450);
    ShuttleAnimator shuttleAnimator = new AShuttleAnimator();
    shuttleAnimator.animateFromOrigin(aShuttle, 5, 100, anOEFrame);
}
protected void animateYFromOrigin(PlottedShuttle shuttle, int animationStep, int animationPauseTime, int startY, int endY, OEFrame frame) {
    // make sure we don't go past final Y position
    while (startY < endY) {
        ThreadSupport.sleep(animationPauseTime);
        startY += animationStep;
        shuttle.setShuttleY(startY);
        frame.refresh();
    }
    // move to destination Y position
    shuttle.setShuttleY(endY);
}
Both animateFromOrigin methods take the same amount of time.

The refreshing one changes the display, the other one does not.
Refreshing Architecture

- APlottedShuttle
  - setShuttleX(Y)()
- Component
  - repaint()
  - paint()
- ObjectEditor
  - refresh()
- AShuttleAnimator
- animateFromOrigin()
- animateFromOrigin
- Main Class
  - main
PROBLEM WITH FULL REFRESH

- The entire UI is refreshed, not just the part that changes
- Multiple GUIs (views) may be displaying the same object
- Should create animated objects as observables
public static void demoShuttleAnimation(
    ShuttleAnimator aShuttleAnimator,
    PlottedShuttle aShuttle,
    OEFrame anOEFrame) {
    aShuttleAnimator.animateFromOrigin(aShuttle, 5, 100);
    aShuttleAnimator.animateFromOrigin(aShuttle, 5, 100, anOEFrame);
}

public class ObservableShuttleAnimationDriver extends ShuttleAnimationDriver {
    public class ObservablePlottedShuttle extends PlottedShuttle {
        public static void main(String[] args) {
            ObservablePlottedShuttle shuttle =
                new AnObservablePlottedShuttle(50, 100);
            OEFrame oeFrame = ObjectEditor.edit(shuttle);
            oeFrame.hideMainPanel();
            oeFrame.setSize(400, 400);
            oeFrame.setLocation(400, 0);
            PropertyChangeListener view = new APlottedShuttleView(shuttle);
            shuttle.addPropertyChangeListener(view);
            JFrame frame = new JFrame("Plotted Shuttle");
            frame.add((Component) view);
            frame.setSize(400, 400);
            frame.setVisible(true);
            ShuttleAnimator shuttleAnimator = new AShuttleAnimator();
            demoShuttleAnimation(shuttleAnimator, shuttle, oeFrame);
        }
    }
}

using mvc
ANIMATION AND MVC
Non Observable Architecture

- APlottedShuttle
  - setShuttleX(Y)()
- AShuttleAnimator
  - animateFromOrigin()
  - refresh()
- Component
  - repaint()
  - paint()
- ObjectEditor
  - refresh()
- MainClass
  - main
Observable Architecture

- APlottedShuttle
  - setShuttleX(Y)()
  - repaint()

- Component
  - paint()

- ObjectEditor
  - refresh()

- animateFromOrigin()

- AShuttleAnimator

- animateFromOrigin

- MainClass
  - main
Incremental Updates

- After each animation step, all displays of the animation must be updated.
- The refresh operation can be used to ensure these updates are made.
- Better, the observable-observer concept can be used to ensure these updates are made:
  - for each graphical property changed by the animation, the class of the property should allow observers to be registered and the setter of the property informs the observers about the update.
- ObjectEditor requires the JavaBeans observer-observer approach based around the PropertyChangeListener interface.
Extra
public static void main(String[] args) {
   PlottedShuttle shuttle = new APlottedShuttle(50, 100);
   OEFrame oeFrame = ObjectEditor.edit(shuttle);
   oeFrame.hideMainPanel();
   oeFrame.setSize(450, 450);
   ShuttleAnimator shuttleAnimator = new AShuttleAnimator();
   shuttleAnimator.animateFromOrigin(aShuttle, 5, 100);
}

public static void main(String[] args) {
   PlottedShuttle shuttle = new APlottedShuttle(50, 100);
   OEFrame oeFrame = ObjectEditor.edit(shuttle);
   oeFrame.hideMainPanel();
   oeFrame.setSize(450, 450);
   FancyShuttleAnimator shuttleAnimator = new AFancyShuttleAnimator();
   ObjectEditor.edit(shuttleAnimator);
}
public class AFancyShuttleAnimator extends AShuttleAnimator implements FancyShuttleAnimator {

    int animationStep = 5;
    int animationPauseTime = 100;
    PlottedShuttle shuttle;

    public AFancyShuttleAnimator(PlottedShuttle theShuttle) {
        shuttle = theShuttle;
    }

    public int getAnimationStep() {
        return animationStep;
    }

    public void setAnimationStep(int animationStep) {
        thisanimationStep = animationStep;
    }

    public int getAnimationPauseTime() {
        return animationPauseTime;
    }

    public void setAnimationPauseTime(int animationPauseTime) {
        thisanimationPauseTime = animationPauseTime;
    }

    public void animateShuttle() {
        animateFromOrigin(shuttle, animationStep, animationPauseTime);
    }
}
GUI
VIDEO
Problem is that the user interface both executes loop and does the redraws the screen

Redrawing of screen done after loop is over

In the main case, the man thread or activity, executes loop

The UI thread or activity redraws the screen
ASKING OBJECTEDITOR TO INTERACTIVE CALL IN SEPARATE THREAD

```java
@SeparateThread(true)
public void animateShuttle() {
    animateFromOrigin(shuttle, animationStep, animationPauseTime);
}
```