Chris Piech
CS 106A

Handout #18
Feb 1, 2017

UFO Game Example

Based on a handout by Patrick Young.

```
/*
 * File: UfoGame.java
 * ------------------
 * This program plays a game where the user tries to
 * shoot a UFO before the UFO "lands".
 */

import acm.program.*;
import acm.graphics.*;
import java.awt.*;
import java.awt.event.*;

public class UfoGame extends GraphicsProgram {

   /** Size and speed of UFO */
   private static final int UFO_WIDTH = 40;
   private static final int UFO_HEIGHT = UFO_WIDTH / 2;
   private static final int UFO_SPEED = 5;

   /** Size and speed of bullets */
   private static final int BULLET_SPEED = 10;
   private static final int BULLET_DIAM = 5;

   /** Animation cycle delay */
   private static final int DELAY = 10;


   public void run() {
      setup();
      while (!gameOver()) {
         moveUFO();
         moveBullet();
         checkForCollisions();
         pause(DELAY);
      }
   }

   /** setup UFO and add mouse listeners */
   private void setup() {
      ufo = new GRect(UFO_WIDTH, UFO_HEIGHT);
      ufo.setFilled(true);
      add(ufo, getWidth(), 0); // UFO starts at top right
      ufoToLeft = true;
      addMouseListeners();
   }

   /** determines if game is over -- true if either
    * the UFO is destroyed or if the UFO lands */
   private boolean gameOver() {
      return (ufo == null) ||
             (ufo.getY() >= getHeight() - UFO_HEIGHT);
   }


   /** when mouse is clicked create bullet, unless a bullet
    * already exists.
    */
   public void mouseClicked(MouseEvent e) {
      if (bullet == null) {
         bullet = new GOval(BULLET_DIAM, BULLET_DIAM);
         bullet.setFilled(true);
         bullet.setColor(Color.RED);
         add(bullet, (getWidth() - BULLET_DIAM) / 2,
                     getHeight() - BULLET_DIAM);
      }
   }


   /** moves UFO, if UFO has moved to edge of screen, moves
    * UFO down and changes UFO direction.
    */
   private void moveUFO() {
      if (ufoToLeft) {
         ufo.move(-UFO_SPEED, 0);
         if (ufo.getX() <= 0) {
            ufoToLeft = false;
            ufo.move(0, UFO_HEIGHT);
         }
      } else {
         ufo.move(UFO_SPEED, 0);
         if (ufo.getX() >= getWidth() - UFO_WIDTH) {
            ufoToLeft = true;
            ufo.move(0, UFO_HEIGHT);
         }
      }
   }


   /** moves bullet */
   private void moveBullet() {
      if (bullet != null) {
         bullet.move(0, -BULLET_SPEED);
      }
   }


   /** checks for bullet interaction with the world
    * (either colliding with the UFO or moving offscreen
    */
   private void checkForCollisions() {
      collideWithUFO();
      moveOffScreen();
   }
   /** checks to see if UFO and bullet collide, if so
    * bullet and UFO are removed and both variables are
    * set to null.
    */
   private void collideWithUFO() {
      if (bullet != null) {
         GObject collObj = getElementAt(bullet.getX(), bullet.getY());
         if (collObj == ufo) {
            remove(ufo);
            remove(bullet);
            ufo = null;
            bullet = null;
         }
      }
   }


   /** determines if bullet has moved of screen,
    * if it has, removes bullet, sets variable to null
    */
   private void moveOffScreen() {
      if (bullet != null) {
         if (bullet.getY() <= -BULLET_DIAM) {
            remove(bullet);
            bullet = null;
         }
      }
   }


   /* private instance variables */
   private GRect ufo;
   private GOval bullet;
   private boolean ufoToLeft; // when true, UFO is moving to left

}
```