

## Lab 7 (no lab 6)

### Goals:

- Learn about the usage of vectors
- Learn about Java's mouse-events
- Explore and experiment with the PinBall game application code

Using the Konqueror browser, go to the course web-page: <http://www.cs.uni.edu/~fienup/cs062s06> and click on the Lab7.zip link and "Save to disk" in the cs2 subdirectory created in lab 1. In a shell window, change to the cs2 subdirectory. Use the "ls" command to list the files in the directory and look for the file "Lab7.zip". Decompress the "Lab7.zip" using the command "unzip Lab7.zip", then change to the Lab7 subdirectory.

Compile and run the PinBallGameDriver.java. Play around with the program for a while.

I want you to fix the following problems in the PinBall game code:

1. The items on the pallet are still stored in the targets vector, so that they will be checked for a hit, even though they can never be hit by a ball. A better solution would have been to create a new vector pallet that will hold these items. You'll need to redraw both the pallet and the targets items on a repaint, but only if a target in the targets vector is hit by a ball.
2. It's very annoying to move a target from the pallet and drop it in the wrong place. So, extend the solution to allow for the targets in the playing field to be moved. One way to do this is to modify the mousePressed method of the MouseKeeper class so it checks to see if you pressed on an item in the targets Vector. If a target was pressed on, then use instance variables "element" in MouseKeeper to remember which target was pressed on and a flag called "TargetPressedOn" to indicate that a target was pressed on. That way whenever the mouse is released, the "TargetPressedOn" can be checked. If it is set, then you can move the remembered target to the spot where the mouse was released.

If you have time and want a challenge, you can get a little **extra credit** for implementing one or all of the following:

- a) Use the mouseDragged method of MouseMotionListener(/MouseMotionAdapter) to improve the "dragging look-and-feel"
- b) Currently balls do not test to see if they intersect with other balls. We could support this modification by making PinBall implement the PinBallTarget interface, and adding balls to the list of targets as well as the list of balls. Modify the code to support balls bouncing off each other.
- c) Develop a "paddle" target object. When a user clicks the mouse over the paddle, the paddle should move back and forth (perhaps only once). If a paddle encounters a ball, the ball is reflected off the paddle. Additionally, you should include code that adds a paddle to the pallet.

### Turn In:

After you complete the lab, turn in a print out of PinBallGame.java and any other modified files. (If the printer stops working again, raise your hand and you can demo your modified program for me or the TA.)