YEAH session #3

4 February 2013, 7:30p-8:30p
Miles Seiver
# Review session schedule

<table>
<thead>
<tr>
<th>Topic</th>
<th>Date</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>assignment 3</td>
<td>today!</td>
<td>now!</td>
<td>here!</td>
</tr>
<tr>
<td>midterm 1</td>
<td>Sun 9 Feb</td>
<td>1p - 4p</td>
<td>Hewlett 200</td>
</tr>
<tr>
<td>assignment 4</td>
<td>Thu 13 Feb</td>
<td>5:30p - 6:30p</td>
<td>Hewlett 200</td>
</tr>
<tr>
<td>assignment 5</td>
<td>Sun 23 Feb</td>
<td>7p - 8p</td>
<td>Hewlett 200</td>
</tr>
<tr>
<td>midterm 2</td>
<td>Sun 2 Mar</td>
<td>1p - 3p</td>
<td>Hewlett 200</td>
</tr>
<tr>
<td>assignment 6</td>
<td>Thu 6 Mar</td>
<td>5:30p - 6:30p</td>
<td>Hewlett 200</td>
</tr>
<tr>
<td>assignment 7</td>
<td>Sun 16 Mar</td>
<td>7p - 8p</td>
<td>Hewlett 200</td>
</tr>
</tbody>
</table>
Assignment 3

- Breakout
- Due Monday, February 10 at 3:15pm
Variable scoping

```java
for (int i = 0; i < 5; i++) {
    int y = i * 4;
}
i = 3; // Error!
y = 2; // Error!
```
Variable scoping cont.

```java
public void run() {
    int x = 5;
    someOtherMethod();
}

private void someOtherMethod() {
    x = 4; // Error!
}
```
Many returns

```java
private int thisIsLegal(int x) {
    if (x == 5) {
        return 0;
    }
    return 1;
}
```

The only way we can get here is if `x` is not equal to 5.
Breakout graphics
/* Width and height of application window in pixels */
public static final int APPLICATION_WIDTH = 400;
public static final int APPLICATION_HEIGHT = 600;

/* Dimensions of game board (usually the same) */
private static final int WIDTH = APPLICATION_WIDTH;
private static final int HEIGHT = APPLICATION_HEIGHT;

/* Dimensions of the paddle */
private static final int PADDLE_WIDTH = 60;
private static final int PADDLE_HEIGHT = 10;

/* Offset of the paddle up from the bottom */
private static final int PADDLE_Y_OFFSET = 30;

/* Number of bricks per row */
private static final int NBRICKS_PER_ROW = 10;

/* Number of rows of bricks */
private static final int NBRICK_ROWS = 10;

/* Separation between bricks */
private static final int BRICK_SEP = 4;

/* Width of a brick */
private static final int BRICK_WIDTH =
    (WIDTH - (NBRICKS_PER_ROW - 1) * BRICK_SEP) / NBRICKS_PER_ROW;

/* Height of a brick */
private static final int BRICK_HEIGHT = 8;

/* Radius of the ball in pixels */
private static final int BALL_RADIUS = 10;

/* Offset of the top brick row from the top */
private static final int BRICK_Y_OFFSET = 70;

/* Number of turns */
private static final int NTURNS = 3;
Paddle
Mouse movement

addMouseListeners()

public void mouseMoved(MouseEvent e) {
    double mouseX = e.getX();
    double mouseY = e.getY();
    // ...
}

The ball

Which dimensions do the G0val constructor take?
while (not-done-condition) {
    update graphics  obj.move(dx, dy);
    pause(pause-time);
}
Ball movement

double vx;
double vy;

while (not-done-condition) {
    ball.move(vx, vy);
    pause(pause-time);
}

private RandomGenerator rgen = RandomGenerator.getInstance();
vx = rgen.nextDouble(1.0, 3.0);
if (rgen.nextBoolean(0.5)) vx = -vx;

waitForClick();

nextDouble(-3.0, +3.0)
Collisions

public GObject getElementAt(double x, double y)

- Why not the middle of each side?
- Two types of collisions
  - A hit: getElementAt != null
  - A “hit”: off the screen
To consider...

- What gets inverted in a normal bounce?
- What if the ball “hits” multiple corners?
Playing the game
Ending the game

- Remove the ball once it goes off the screen, doesn’t disappear automatically
  - `remove();`
- Detect winning and losing
  - how?
  - track bricks remaining
Testing
Try changing the boxed constants.
They each change an aspect of the game.
The game must still work.
common bug: ball stuck in paddle
the auto-play trick!

a.k.a the one-line A.I.
Click to serve!
Click to serve!

(+ one line of code)
extreme mode
SUPER
extreme
mode

Click to serve!
SUPER

extreme

mode

MEGA

Click to serve!
Click to serve!
Style
Should I use an instance variable?

General rules for when an instance variable is appropriate:

1. If you need to access the variable in `mouseListener` methods, or
2. You access and change the variable ALL over the place, or
3. There’s just no other way.

Avoid using instance variables unless you need them. It is poor style to make something an instance variable when it could have been a local variable.
Instance variables in Breakout

You’ll justify each of your instance variables in the writeup.

- Ball? Yes, probably
- Bricks? No
- Paddle? Yes, definitely
Breakout has a ton of possible extensions!

```java
AudioClip bounceClip = MediaTools.loadAudioClip("bounce.au");

bounceClip.play();
```
• Follow the specifications carefully
• Comment
• Go to the LaIR if you get stuck
• Incorporate IG feedback!

• Have fun!