CS-259
Data Structures with Java
Project 2: Breakout v3:
Trivial Breakout

Instructor: Joel Castellanos
e-mail: joel@unm.edu
web: http://cs.unm.edu/~joel/
Course website:
http://cs.unm.edu/~joel/cs259/

10/14/2009

Breakout: Project 2, Milestone 3

Due Sunday, Oct 25
All blocks start green
On 1st hit, blocks turn red.
On 2nd hit, blocks are destroyed.

Ball shows 25 turn history.

Four rows of JLabels show results
of each corner when ball
collides.

Timer starts when "next" button
is clicked and stops when the ball
has a collision.
Four Corners on Collision

In which direction is the ball moving?

A collision occurs when the ball penetrates one pixel into a block or wall.

0) [335, 254]: MISS
1) [346, 254]: HIT: surface=VERTICAL_EDGE, Block [7,11]
2) [346, 265]: MISS
3) [335, 265]: MISS

Data class in BreakoutDraw.java

```java
import java;

public class BreakoutDraw extends JPanel {
    // The Collision class is at the same level as BreakoutDraw.
    // The Collision class cannot be public because the file name is not Collision.java.
    // The Collision class is used to hold the results of a collision.
    class Collision {
    }
}
```
**class Collision: Fields**

```java
class Collision {
    // The next four constants are possible values of surface.
    public static final int INTERIOR = 0; // The collision is somewhere in the middle of the block, thus it is unknown which side was entered.
    public static final int VERTICAL_EDGE = 1;
    public static final int HORIZONTAL_EDGE = 2;
    public static final int CORNER = 3;

    public int surface = 0; // One of the four surface types
    public boolean hit = false; // true when hit any surface
    public boolean hitBlock = false; // true when hit block.

    public int col = -1; // if hitBlock=true, column index of hit block.
    public int row = -1; // if hitBlock=true, row index of hit block.

    public String toString() {
        String str;
        if (hit) {
            if (surface == VERTICAL_EDGE) {
                str = "HIT: surface=VERTICAL_EDGE";
            } else if (surface == HORIZONTAL_EDGE) {
                str = "HIT: surface=HORIZONTAL_EDGE";
            } else if (surface == CORNER) str = "HIT: surface=CORNER";
            else str = "HIT: surface=UNKNOWN";
            if (hitBlock)
                str +=", Block [" + col + ", " + row + "]";
        }
        else str = "MISS";
        return str;
    }
}
```
Reverse Y-speed

Reverse Y-speed
Reverse Y-speed

Reverse Y-speed and X-speed
Reverse Y-speed and X-speed

Ball Start Location and Speed

- Start in a random location along the bottom of the draw area.
- Start with vertical velocity of -10.
- Start with horizontal velocity as random number between [-15, +15]
- Set timer to trigger every 20 milliseconds.
## Grading Rubric

<table>
<thead>
<tr>
<th>Points</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Follows CS-259 coding standard.</td>
</tr>
<tr>
<td>2</td>
<td>Uses and sets the Collision object.</td>
</tr>
<tr>
<td>4</td>
<td>Ball stops when it penetrates 1 pixel into a block or wall.</td>
</tr>
<tr>
<td>4</td>
<td>Correct collision information is displayed in JLabels for each corner of the ball.</td>
</tr>
<tr>
<td>4</td>
<td>When next button is clicked, ball moves to the next collision in the correct direction with the correct speed.</td>
</tr>
<tr>
<td>1</td>
<td>Displays correct ball history.</td>
</tr>
<tr>
<td>3</td>
<td>Screen resizes, repaints and draws correctly.</td>
</tr>
</tbody>
</table>