

# CS-259

## Data Structures with Java

Project 2: Breakout v3:  
Trivial Breakout



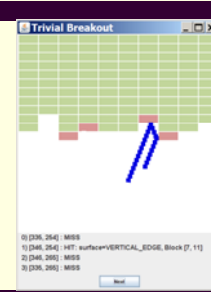
Instructor: **Joel Castellanos**

e-mail: [joel@unm.edu](mailto:joel@unm.edu)

web: <http://cs.unm.edu/~joel/>

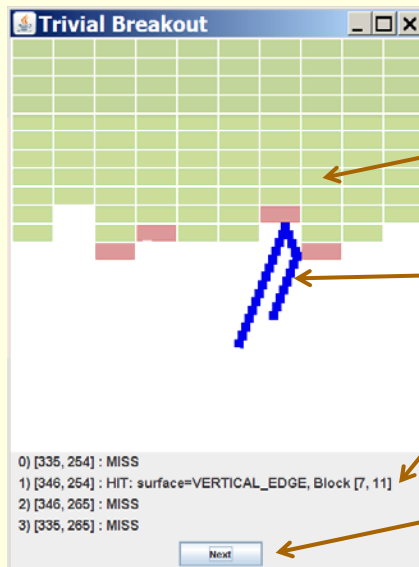
Course website:

<http://cs.unm.edu/~joel/cs259/>



10/14/2009

## Breakout: Project 2, Milestone 3



Due Sunday, Oct 25

All blocks start green

On 1<sup>st</sup> hit, blocks turn red.

On 2<sup>nd</sup> hit, blocks are destroyed.

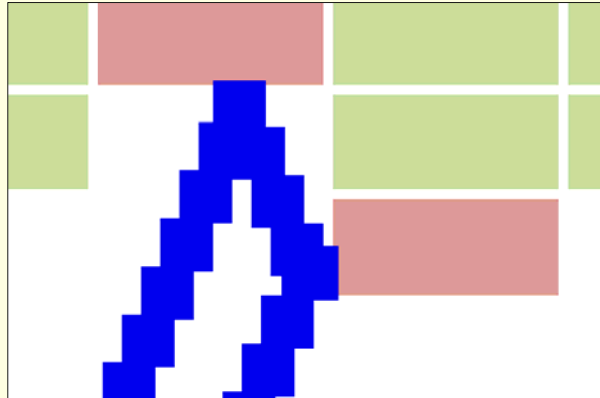
Ball shows 25 turn history.

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.

2

## Four Corner's 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

3

## Data class in BreakoutDraw.java

```
import   
  
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  
{   
}
```

4

## class Collision: Fields

```
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.

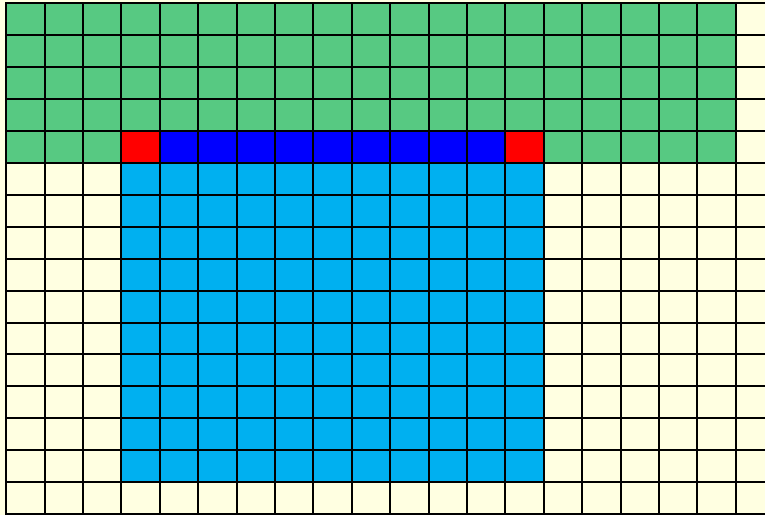
    public String toString() ☒
}
5
```

## class Collision: toString()

```
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";

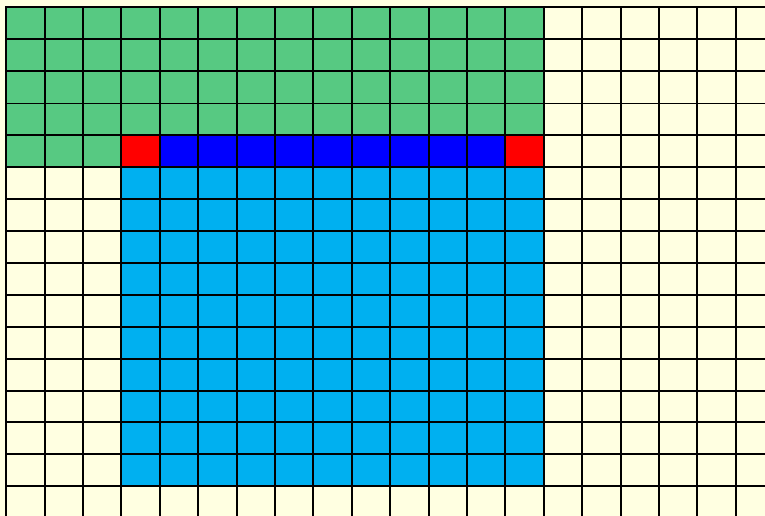
        if (hitBlock)
        {
            str += ", Block [" + col + ", " + row + "];"
        }
    }
    else str = "MISS";
    return str;
}
6
```

## Reverse Y-speed



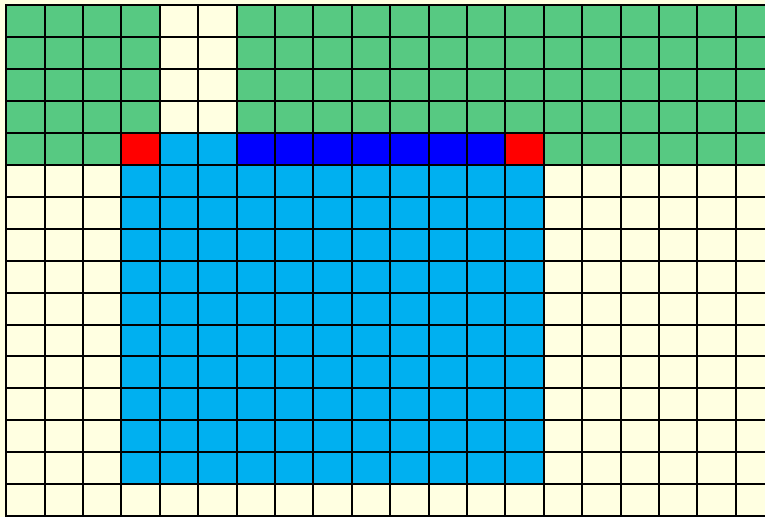
7

## Reverse Y-speed



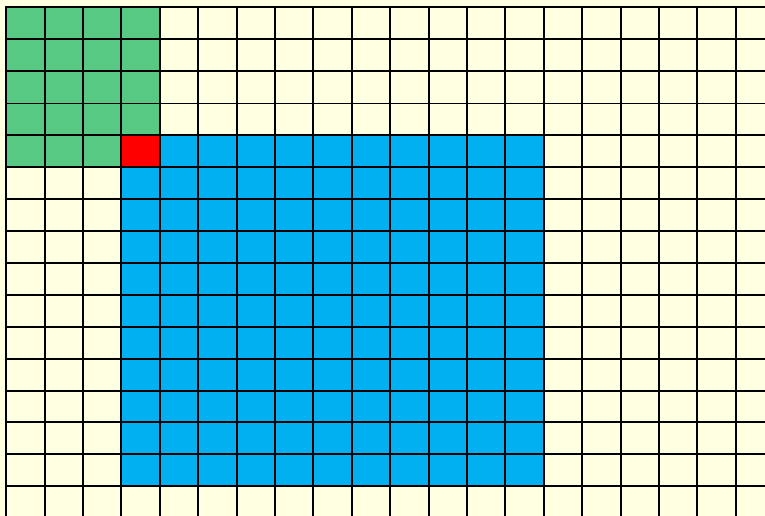
8

## Reverse Y-speed



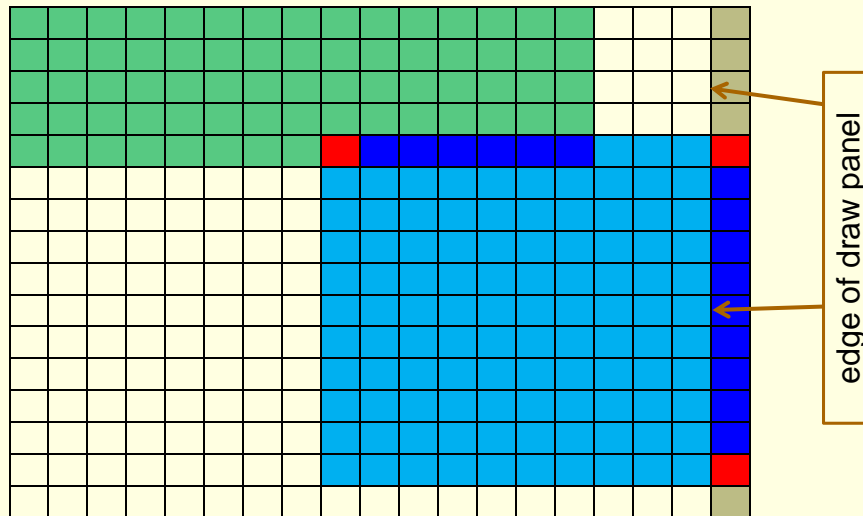
9

## Reverse Y-speed and X-speed



10

## Reverse Y-speed and X-speed



11

## 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.

12

## Grading Rubric

---

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

13