

CS102 Spring 2017

Assignment #8

Due on May 9th, by midnight

Breakout, Phase 2:

Attribution: This assignment appears in the book "The Art and Science of Java", by Eric Roberts, on page 406, Exercise 10.

In this assignment, you will complete the Breakout game you started in the last lab. The Breakout game was developed in 1976 by Steve Wozniak, who would later become one of the founders of Apple. In Breakout, your goal is to clear a collection of bricks by hitting each of them with a bouncing ball, using a paddle at the bottom of the screen to keep the ball in play.

The initial configuration of the Breakout game is shown in the figure below. The colored rectangles near the top of the screen are bricks, two rows each of red, orange, yellow, green, and cyan, with 10 bricks per row. The rectangle at the bottom of the screen is a paddle that can move back and forth in the horizontal dimension until it reaches the edge of its space.



A complete Breakout game consists of **three turns**. On each turn, a ball is launched from the midpoint toward the bottom of the canvas. The ball bounces off the paddle and the walls of the canvas. When the ball collides with one of the bricks at the top of the canvas, it should bounce as it does on any other collision, but the brick it collided with should disappear. Play continues like this until one of the following conditions holds:

1. The ball goes off the lower wall, indicating a miss with the paddle. In this case, the message "CLICK TO SERVE!" appears when a turn ends and the next ball is served when the user clicks the mouse on the scene if there are turns remaining. If no turns remain, the game ends in a loss, when the words "YOU LOSE" appear on the scene.
2. The last brick is eliminated. In this case, the game ends and you win, retiring in victory. An appropriate message such as "YOU WIN!" then appears on the scene.

After a hole has been made from the bottom to the top in a column of bricks, a path will be open to the top wall. When this happens, the ball will strike bricks on the top of the wall, removing them without you having to move the paddle. This condition is the reward for "breaking out," hence the name of the game. The code for checking whether the ball is in collision with objects was given to you in the starter code for the last lab.

Advice:

1. Start testing your code with only a couple of rows of bricks until you're sure it works as intended. Add more bricks only after you're sure it works correctly with a smaller number. You can implement the bricks as a two-dimensional array declared as an instance variable or using another data structure of your own choosing.
2. In the "checkForObjectBounciness" method, there is no need to check if each separate element of the brick array is the object returned. Just treat all objects that are not the paddle in the same way - remove them and change the direction of the ball.
3. Don't try to implement multiple rounds of the game until you have successfully created the wall of bricks. After getting your program to work with more than one brick you can decide how to implement multiple "rounds" of the game.
4. Keep your program as modular as possible, breaking the code into distinct methods when necessary.

Feel free to experiment and have some fun with this assignment. Change the velocity of the ball to make the game more challenging. Add images or include sounds when the ball hits a brick (I'll show you how to include sounds using the acm package in class). Make up your own rules...whatever. Be creative.

Having written the above paragraph, I need to add that I realize we all have different levels of expertise. If you have difficulty completing some part of the program, seek out help from the coaches (the schedule is posted on the course web page) or your professor (I work on weekends, from home.) Also, submitting a partially implemented but well-documented and elegant program is preferable to letting your frustration level get too high. As a rule of thumb, if you find yourself stumped by a particular problem for more than a couple of hours, put the program away and go do something else. Chances are, your subconscious mind will find an answer while you're sleeping. And if it doesn't, send your professor some email when you wake up.

Submitting your work

Submit your assignment in a compressed folder on the course Moodle page.