Break Out

# Overview

Your goal in this assignment is to create a breakout game. We will start by making the bricks.

There are NUM\_ROWS rows of bricks, with NUM\_BRICKS\_PER\_ROW bricks per row. You should write everything in terms of these constants, and use loops and functions to avoid redundant code.

## Specifications

The first two rows are red, then the next two are orange, then the next two are green, then the next two are blue. Then the same pattern repeats for rows 9 and up. (hint: %)

There should be BRICK\_SPACING pixels between all bricks and the sides in all directions.

The bricks should start BRICK\_TOP\_OFFSET from the top of the window.

Hints

Our solution is about 50 lines of code, and uses four different helper functions to break down the problem. We strongly recommend that you write this using multiple functions.

How can you break down this problem of drawing all the bricks into a smaller problem?

Starter Code

/\* Constants for bricks \*/

var NUM\_ROWS = 8;

var BRICK\_TOP\_OFFSET = 10;

var BRICK\_SPACING = 2;

var NUM\_BRICKS\_PER\_ROW = 10;

var BRICK\_HEIGHT = 10;

var SPACE\_FOR\_BRICKS = getWidth() - (NUM\_BRICKS\_PER\_ROW + 1) \* BRICK\_SPACING;

var BRICK\_WIDTH = SPACE\_FOR\_BRICKS / NUM\_BRICKS\_PER\_ROW;

/\* Constants for ball and paddle \*/

var PADDLE\_WIDTH = 80;

var PADDLE\_HEIGHT = 15;

var PADDLE\_OFFSET = 10;

var BALL\_RADIUS = 15;

function start(){

}

# Ball and Paddle

Add the ball and paddle. The ball should bounce around the screen. The paddle should move with the mouse.

Add Ball and Paddle

The first step is to add the ball to the center of the screen, and the paddle to the bottom of the screen.

The next step is to get the paddle to move when you move the mouse. The paddle should be centered under the mouse, and should not go offscreen.

Move Paddle, Bounce Ball

The next step is to get the ball to bounce around the screen. We may have done this exact problem before…

# Ball and Paddle

The last part of the game is to get collisions working. You should remove bricks when the balls collide with them. This is VERY similar to bouncing the ball.

You just need to use the getElementAt(x, y) function to see if there is an element where the ball is. If the element is not null, then you can remove it with the remove function.

You also need the ball to bounce up if it collides with the paddle

You want to check the four corners around the ball to see if there is an element there.

Last, there are just some miscellaneous things you can do:

* Pause the ball when it gets reset until you click. Kind of like the Pause Ball exercise.
* Stop the game after the ball falls to the bottom 3 times.
* Stop the game when all the bricks are removed.
* Display messages on the screen when you win or lose.

Then, you can just go crazy with extensions like these...

* Display a score.
* Add powerups.
* Add cheat codes.
* Add multiple difficulty levels.
* Anything else that you can think of

Plan the project out with pseudocode, flowcharts or storyboards. Do not tackle the project without thinking it out frist.

# Reflection

These should be answer in complete sentences. You should attempt to answer these questions to the best of your ability. If you have a hard time writing answer out; you may record or use a speech to text tool to anwser the questions.

1. What was the most challenging part of this Project?
2. Why is the project breakdown into 3 parts? How was it helpful?
3. How did you use functions in this program? How do those functions relate to the different parts of the project (i.e A, B, C)
4. What events did you use in the game? Why?

Rubric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skill** | **1** | **2** | **3** | **4** |
| **10. Discuss and apply fundamental concepts of programming language. (i.e.: Data types, memory models, data structures, etc.)** | Students can recognize the fundamental concepts of programming languages | Student can identify and categorize the fundamental concepts of programming languages | Student can compare fundamental concepts and explain the differences with evidence to support them. | Student can identify the fundamental concepts of programming and justify how and why someone would use them with specific examples |
| **11. Demonstrate and use a variety of software development tools for program implementation** | Student can recognize and identify different software development tools | Student can explain the difference between two or more software development tools and use different software development tools to solve routine problems | Student analyze and select an appropriate software development tool and can use different software development tools to solve non routine problems | Students can evaluate a problem and justify which software development tools they would use to solve the problem with evidence to support their argument |
| **12. Write computer programs utilizing the structured programming paradigm.** | Student can identify the structured programming paradigm and how it functions | Student can use the structured programming paradigm to solve a routine problem | Student can use the structured programming paradigm to solve a non routine problem | Student can analyze a series of complex problems and modify them with justification of the structure in the new designs |
| **14. Write programs using modularization techniques to reduce program complexity and improve program maintainability** | Student can identify modularization techniques | Student can explain why modularization is used and how it can be useful and use it to solve routine problems | Student can formulate a modularized program for a non routine problem | Student can critique the use of modularization in a program with an explanation and evidence to support their argument |
| **19. Write programs that use events to cause program execution to react to the event by writing the appropriate event handler code** | Student can identify what events and event handlers | Student can explain the difference between an event and an event handler and solve routine problems with events | Student can explain the difference between an event and an event handler with evidence and solve non routine problems with events | Student can create a program that uses events and critique the use or justify use of those events |
| Target Grade: 12 | | TOTAL | |  |