

StringCheese

Sasha Fomina, Daniel Ju, Kevin Li

Period 4

Project: Bubble Shooter

Rules of Play:

The screen is set up with several row of bubbles of 5 distinct colors. The player must shoot a bubble of a randomly generated color at their choice of angle. with the objective of popping as many bubbles as possible. If a bubble attaches to the rows of bubbles, so that a group of 3 or more neighboring bubbles of the same color are created, then that group of bubbles are popped. If a player clears the screen, they win. The bubble can attach to the rows of bubbles without popping anything. Once a certain number of rows contain at least one ball, then the player loses.

Design.

There will be a BubbleShooter class which will contain a 2d- array of Bubbles. This 2d array will be filled to form a hex array like so: (1 represents balls and 0 are meaningless values)

[0 1 0 1 0 1 0 1 0 1 0 1 0]

[1 0 1 0 1 0 1 0 1 0 1 0 1]

[0 1 0 1 0 1 0 1 0 1 0 1 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0] → empty row to be filled, after all empty rows have at least one bubble the player loses

The Bubble class will contain attributes including color, xcor, ycor, and an array of 6 Bubble neighbors, effectively creating a tree. When a Bubble is shot its neighbor array is created and the tree is traversed - if 2 of its direct neighbors are the same color as it then look in the neighbor arrays of each of these 2 matching Bubbles and continue one until none of the neighbors are matching colors.

To shoot, a Bubble of a random color is generated.

Ways that we implement data structures learned this year:

1. The color of the balls coming up next will be stored in a queue to take advantage of queues' first in first out property.
2. Every time the player successfully shoots down a group of bubbles, the bubbles that just fell down would be stored, in order, in a stack. That allows for us to create a *undo* button, where the last move made can be reversed.
3. Every time the player successfully shoots down a group of bubbles, he/she gains a certain amount of points. Every time a player scores a certain number of points, the point gained as well as the combination of bubbles that the player shot down will be stored in an

arraylist (or any other data structure). At the end of the game, whatever data structure we choose to use will be sorted using a sexy second semester sort (maybe heap or merge sort), this way the player can find out which one of his/her moves gained them the most points

4. The popping mechanism, wherein a bubble launched hits a group of 3 or more connected bubbles of the same color, uses recursive backtracking, as we search through the arrayList of neighbors of the ball launch and then search through the neighbor's neighbors, and so on, finally backtracking back to the original group of neighbors and choosing a different neighbor.