public class Grid	public class DotCluster	public class Driver (Woo)
- int height - int width - DotCluster[height][width] contents Contains the contents of the Grid	- int numDots + String color “red” or “green” if numDots > 0; “” if numDots = 0 ~~String rep~~ ~~The visual representation of the DotCluster to the user. Should use color and 1 or 4 characters~~ (see toString) - final int maxDots This will be set when the Grid is initialized, and should not be able to be changed (there are no circumstances in which this max will change)	- Scanner redPlayer Used to take in input from the red player - Scanner greenPlayer Used to take in input from the green player - boolean won true if a player has won the game and the game should end
Grid() Initializes a Grid with size 0x0 Grid(int[] size) Initializes a grid with specified size, using the constructor from above and initializing contents and size. For each DotCluster in contents, maxDots will be set based on the DotCluster’s position. toString() prints out grid with representation of DotClusters in each grid box explode(int x, int y) Helper method for method below. Takes	DotCluster() toString() returns a character (or 4) to represent the DotCluster, based on numDots and color. Necessary for toString in Grid to work. Preferable to having an instance variable for a String representation, because this more easily maintains the invariant that the String representation always matches what is specified in color and numDots.	runGame() First creates an instance of Grid. Then, prints out instructions to the player, as needed, then the Grid itself. Invokes checkWon(). If true, game ends, and winning player is congratulated. If false, invokes turn(). turn() Prints out a statement telling the (two) user(s for now) whose turn it is. Receives user input, alternating between using redPlayer and greenPlayer, on where they want to add a dot. Checks whether this location is permissible - if so, invokes addDot; if not, allows them to try again

dots from one box and distributes them to neighboring boxes. Uses DotCluster.numDots. Sets the color of neighboring clusters to that of of the cluster that just exploded. ~~explodeAll(int x, int y)~~ ~~Uses the above method to perform all of the explosions initiated when one box explodes~~ addDot(String color, int x, int y) This adds a Dot by increasing DotCluster.numDots of the relevant DotCluster by 1. It then checks whether this reaches maxDots for that cluster, and if so, it invokes explode(). (Note that using this function and explode() together obviates explodeAll.)		until they give a permissible location. boolean checkWon() Checks if a player has won by checking if DotCluster.color has the same value for every DotCluster in Grid.