```
/* SandLab - makes a window that has an interactive game with particles that have
different attributes.
 * @author Claire Waidelich
 * @date 2017-03-?? Went in to Jacob Hertzler with this one to work on it.
 * @date 2017-03-?? Worked on the lab put in logics  with particle movements
 * @date  added Helium as the new particle
 */
 import java.awt.*;

import java.util.*;

public class SandLab
{
    abstract class Particle { // parent
        int row = -1, col = -1;
        abstract public Color getColor();
        abstract public Particle clone();
        public void step() {
        }

        public String getName() {
            return this.getClass().getSimpleName();
        }
        public void moveTo(int row, int col) {
            this.row = row;
            this.col = col;
            grid[row][col] = this;
        }
        public void swapWith(Particle p) {
            int r = row;
            int c = col;
            this.moveTo(p.row, p.col);
            p.moveTo(r, c);
        }

        public Particle above() {
            return grid[row-1][col];
        }
        public Particle below() {
            return grid[row+1][col];
        }
        public Particle onLeft() {
            return grid[row][col-1];
        }
        public Particle onRight() {
            return grid[row][col+1];
        }
    }

    class Empty extends Particle {
        public Color getColor() {
            return Color.BLACK;
        }
        public Particle clone() {
            return new Empty();
    }
}

class Metal extends Particle {
    public Color getColor() {
        return Color.DARK_GRAY;
    }
    public Particle clone() { // spawns a new particle every time you click
        return new Metal();
    }
}

class Sand extends Particle {
    public Color getColor() {
        return Color.YELLOW;
    }
    public Particle clone() {
        return new Sand();
    }
    public void step() { //
        Particle other = below();
        if (other instanceof Sand)
            if (new Random().nextInt(2) == 0)
                other = below().onLeft();
            else
                other = below().onRight();
        if (other instanceof Empty || other instanceof Water) {
            other.swapWith(this);
        }
    }
}

class Water extends Particle {
    public Color getColor() {
        return Color.CYAN;
    }
    public Particle clone() {
        return new Water();
    }
    public void step() {
        Particle other;
        int n = new Random().nextInt(3);
        if (n == 0)
            other = onLeft();
        else if (n == 1)
            other = onRight();
        else
            other = below();
        if (other instanceof Empty) {
            other.swapWith(this);
        }
    }
}

class Helium extends Particle { // one addition! :D
```

```
        public Color getColor() {
            return Color.white;
        }
        public Particle clone() {
            return new Helium();
        }
        public void step() {
            Particle other;
            int n = new Random().nextInt(8);
            if (n == 0)
                other = onLeft();
            else if (n == 1)
                other = onRight();
            else if (n == 2)
                other = null;
            else
                other = above();
            if (other instanceof Empty || other instanceof Water || other
instanceof Sand) {
                other.swapWith(this);
            }
        }
    }

    public static void main(String[] args)
    {
        SandLab lab = new SandLab(120, 80);
        lab.run();
    }

    //do not add any more fields
    private Particle[][] grid;
    private ArrayList<Particle> particles;
    private SandDisplay display;

    public SandLab(int numRows, int numCols) // constructor
    {
        particles = new ArrayList<Particle>();
        particles.add(new Empty());
        particles.add(new Metal());
        particles.add(new Sand());
        particles.add(new Water());
        particles.add(new Helium());


        String[] names = new String[particles.size()];
        for (int i = 0; i < particles.size(); i++)
            names[i] = particles.get(i).getName();

        display = new SandDisplay("Falling Sand", numRows, numCols, names);
        grid = new Particle[numRows][numCols]; //
        for (int r = 0; r < grid.length; r++) // looks through every pixel of
the grid
            for (int c = 0; c < grid[r].length; c++)
```

```
                                new Empty().moveTo(r, c); // puts a new particle on the 
screen
                
        }

        //called when the user clicks on a location using the given tool
        private void locationClicked(int row, int col, int tool)
        {
                particles.get(tool).clone().moveTo(row, col);
        }

        //copies each element of grid into the display
        public void updateDisplay()
        {
                for (int r = 0; r < grid.length; r++)
                        for (int c = 0; c < grid[r].length; c++) {
                                display.setColor(r, c, grid[r][c].getColor());
                        }
        }

        //called repeatedly.
        //causes one random particle to maybe do something.
        public void step()
        {
                try {
                        Random rnd = new Random();
                        int r = rnd.nextInt(grid.length);
                        int c = rnd.nextInt(grid[r].length);
                        grid[r][c].step();
                } catch( java.lang.ArrayIndexOutOfBoundsException e) {
                }
        }

        //do not modify
        public void run()
        {
                while (true)
                {
                        for (int i = 0; i < display.getSpeed(); i++)
                                step();
                        updateDisplay();
                        display.repaint();
                        display.pause(1);  //wait for redrawing and for mouse
                        int[] mouseLoc = display.getMouseLocation();
                        if (mouseLoc != null)  //test if mouse clicked
                                locationClicked(mouseLoc[0], mouseLoc[1], 
display.getTool());
                }
        }
}
```