Design Document

1. Intro

Period 6
Brandon Hu
Group Name: Cobalt
Project Title: Chem Lab

2. Description

There are two main simulators to this. The first purpose is to display and explore different chemical reactions. There are precipitation, smoke, bubbling, and color change reactions. The second is the pH simulator which calculates the pH based on the amount and concentration of the specific acid added.

Functionalities:

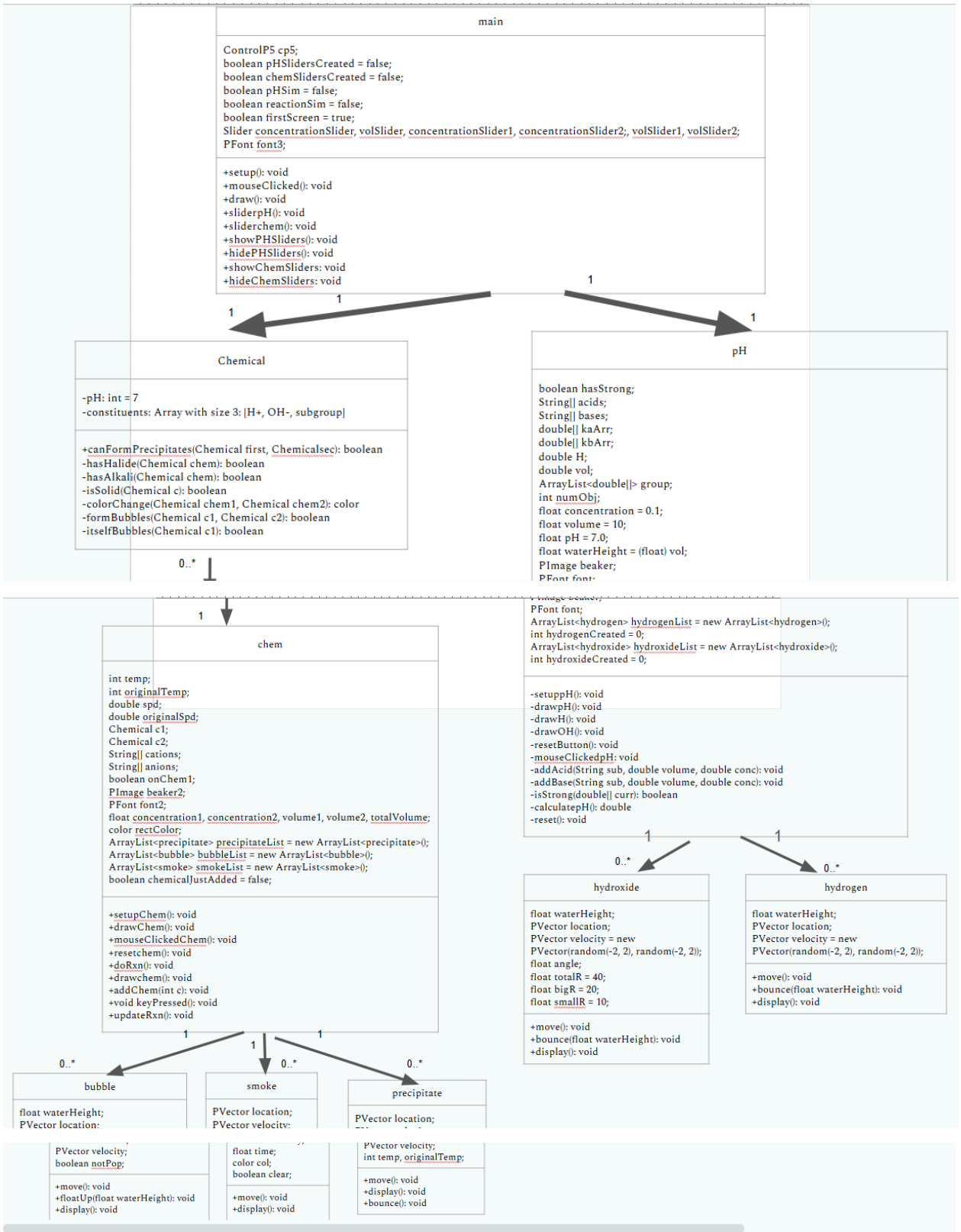
1. -setup(): sets the variables accordingly
2. -draw(): creates the sliders for the volume, concentration, and also the buttons for the bases and the acids
3. -mouseClicked: makes the buttons work
4. sliderpH(): creates the sliders for the pH class
5. sliderchem(): creates the sliders for the chem class
6. showPHSliders(): makes the sliders for the pH class visible
7. showChemSliders(): makes the sliders for the chem class visible
8. hidePHSliders(): hides pH class sliders
9. hideChemSliders: hides chem class sliders
10. hasHalide(Chemical chem), hasAlkali(Chemical chem): returns whether the input has a halide or alkali
11. isSolid(Chemical c): returns whether the chemical is solid
12. formBubbles(Chemical c1, Chemical c2): returns whether the chemical forms bubbles
13. itselfBubbles(Chemical c1): returns whether the input bubbles
14. setupChem(): sets variables for chem class
15. drawChem(): draws for chem class
16. mouseClickedChem(): makes buttons work for chem class
17. resetchem(): resets everything for chem class
18. doRxn(): executes bubbles, smoke, and precipitates functions

19. addChem(int i): adds the chem to the top
20. keyPressed(): changes temperature variable
21. updateRxn(): adds instances of bubble, smoke, and precipitate if appropriate
22. move(): moves the instance of the class
23. display(): displays the instance of the class
24. bounce() or floatUp(): uses barriers to prevent instance going
25. canFormPrecipitate(): checks whether the parameters can form a precipitate with each other
26. setuppH(): sets variables for pH class
27. drawpH(): draws the pH class
28. drawH(): draws the hydrogen instances
29. drawOH(): draws the hydroxide instances
30. mouseClickedpH(): makes the pH buttons work
31. -addAcid(String sub, double volume, double conc): creates the acid, updates both the volume and increases the number of objects created
32. -addBase(String sub, double volume, double conc): creates the base, updates both the volume and decreases the number of objects created
33. -calculatepH(): accurately determines the total amount of H added by checking whether the given acid/base is strong, then finding the calculations of pH based on that
34. reset(): resets variables for pH class

Uses the ControlP5 Library to create sliders that change the volume and concentration of the chemical or acid/base

Using the tools tab, have Constantia.vlw and Constantia-36.vlw downloaded in order to compile

3.


main
ControlP5 cp5;
boolean pHSlidersCreated = false;
boolean chemSlidersCreated = false;
boolean pHSim = false;
boolean reactionSim = false;
boolean firstScreen = true;
Slider concentrationSlider, volSlider, concentrationSlider1, concentrationSlider2;, volSlider1, volSlider2;
PFont font3;
+setup(): void
+mouseClicked(): void
+draw(): void
+sliderpH(): void
+sliderchem(): void
+showPHSliders(): void
+hidePHSliders(): void
+showChemSliders: void
+hideChemSliders: void
1
1
1
1
Chemical
-pH: int = 7
-constituents: Array with size 3: [H+, OH-, subgroup]
+canFormPrecipitates(Chemical first, Chemicalsec): boolean
-hasHalide(Chemical chem): boolean
-hasAlkali(Chemical chem): boolean
-isSolid(Chemical c): boolean
-colorChange(Chemical chem1, Chemical chem2): color
-formBubbles(Chemical c1, Chemical c2): boolean
-itselfBubbles(Chemical c1): boolean
pH
boolean hasStrong;
String[] acids;
String[] bases;
double[] kaArr;
double[] kbArr;
double H;
double vol;
ArrayList<double[]> group;
int numObj;
float concentration = 0.1;
float volume = 10;
float pH = 7.0;
float waterHeight = (float) vol;
PImage beaker;
PFont font;
0..*
1
chem
int temp;
int originalTemp;
double spd;
double originalSpd;
Chemical c1;
Chemical c2;
String[] cations;
String[] anions;
boolean onChem1;
PImage beaker2;
PFont font2;
float concentration1, concentration2, volume1, volume2, totalVolume;
color rectColor;
ArrayList<precipitate> precipitateList = new ArrayList<precipitate>();
ArrayList<bubble> bubbleList = new ArrayList<bubble>();
ArrayList<smoke> smokeList = new ArrayList<smoke>();
boolean chemicalJustAdded = false;
+setupChem(): void
+drawChem(): void
+mouseClickedChem(): void
+resetchem(): void
+doRxn(): void
+drawchem(): void
+addChem(int c): void
+void keyPressed(): void
+updateRxn(): void
PFont font;
ArrayList<hydrogen> hydrogenList = new ArrayList<hydrogen>();
int hydrogenCreated = 0;
ArrayList<hydroxide> hydroxideList = new ArrayList<hydroxide>();
int hydroxideCreated = 0;
-setuppH(): void
-drawpH(): void
-drawH(): void
-drawOH(): void
-resetButton(): void
-mouseClickedpH: void
-addAcid(String sub, double volume, double conc): void
-addBase(String sub, double volume, double conc): void
-isStrong(double[] curr): boolean
-calculatepH(): double
-reset(): void
1
1
0..*
0..*
hydroxide
float waterHeight;
PVector location;
PVector velocity = new
PVector(random(-2, 2), random(-2, 2));
float angle;
float totalR = 40;
float bigR = 20;
float smallR = 10;
+move(): void
+bounce(float waterHeight): void
+display(): void
hydrogen
float waterHeight;
PVector location;
PVector velocity = new
PVector(random(-2, 2), random(-2, 2));
+move(): void
+bounce(float waterHeight): void
+display(): void
1
1
1
0..*
0..*
0..*
bubble
float waterHeight;
PVector location;
PVector velocity;
boolean notPop;
+move(): void
+floatUp(float waterHeight): void
+display(): void
smoke
PVector location;
PVector velocity;
float time;
color col;
boolean clear;
+move(): void
+display(): void
precipitate
PVector location;
PVector velocity;
int temp, originalTemp;
+move(): void
+display(): void
+bounce(): void

4. How does it work?

The first screen just has you choose which simulator to start with. A slider, using the imported library ControlP5, will be displayed, which changes the volume and concentration of the selected acid/chemical. The acid and base will be selected by using the mouse to click it. For the other simulator, select one option from the left and right side using your mouse click. You can see your option at the top. In order to create chemical 2, press the button that says “Switch, Now 1”. You can change chemical 2 in the same way. Using the up and down arrows, you increase or decrease the temperature, which changes the speed at which the precipitates fall.

A reset button that resets it to the normal state exists for both simulators; uses the mouse click.