Project B: Interactive Data Visualization

Carlos Coca
Professor Lomelino
Foundation of Software Design

Concept Summary

The UNSD yearly provides global environmental statistics. As technology is constantly changing, and as us humans are still consuming natural resources, the world is encouraged to develop technologies to produce energy in a “cleaner” way. The UNSD has been keeping track of the way of production of energy at a global scale.

This case study shows the comparison through time of the total percentage of energy being produce by non renewable sources vs renewable sources.

Concept Definition

Part 1: Concept Definition

Who

The results of this project is targeted to be used by the UNSD (United Nations Statistics Division). The UNSD servers under the United Nations Department of Economics and Social Affairs. This specific department was established in 1947, with the purpose of supplying the statistical needs and coordinate activities for the global statistical system. This program records global statistics from all aspects: environment, energy, population, consumerism, economics, etc. From the data that they gather, the UN, G-16, and the G-8, decide the paths that the countries and the should should take in the future.

Part 1: Concept Definition

What

The latest Total Energy Supply report were compiled in 2015. The UNSD has already gathered the data and presented all the statistics, but they're trying to find a way to better explain what the data that was recollected actually means; they want to make the data "easier" to be read by the public. Because of it, they have hired me to create a Data Visualization interactive and non interactive diagram. I have gathered data statistics from multiple sources to compare if the data from one source and the other are equal, slightly different, completely different, with the purpose of validating the data that I'm going to use.

Part 1: Concept Definition

When

Since updated report has already been released to the public, the UNSD would like to have the Data Visualization ready by the end of November, so the information can be checked, compiled, and released by December; before the new period starts.

Where

The info-graphics that I'm going to provide are going to be used in the Energy Statistics Yearbook provided by the Environmental Statistics division of the UNSD.

Why

As the UNSD has been releasing their studies to the public, it has come to their attention that people are not quite grasping the results of their collective data. If you open the Energy Supply statistics per region, the Excel sheet is extensive; filled with numbers and percentage. To provide an easier insight of what's happening in the world the UNSD has decided to provide Data Visualization.

Sketches and Ideation

Part II

Main Page: 2 sections

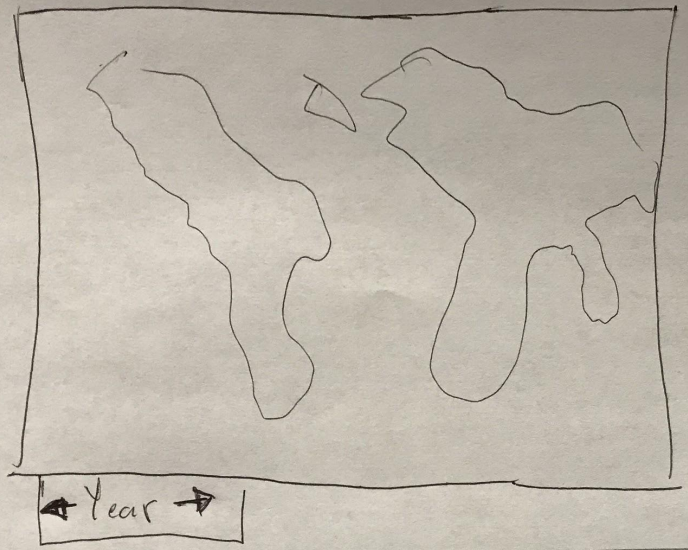
Energy Supply Introduction

- Visual 0
- Visual 2

Comparison of Renewable Energy

- Visual 1
- Visual 3

Visual 1



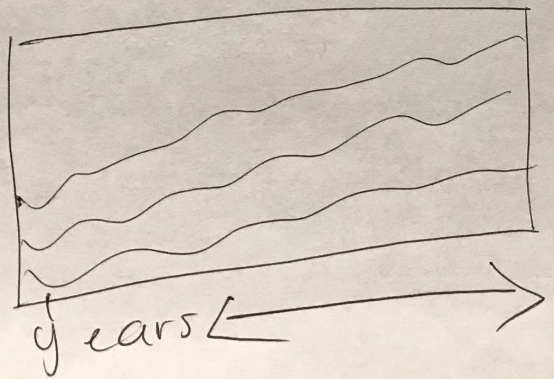
Based on Renewable Electric Production Excel

Number changes as you swap the years

World wide

Visual 2

Linegraph, by 5 1990-2015

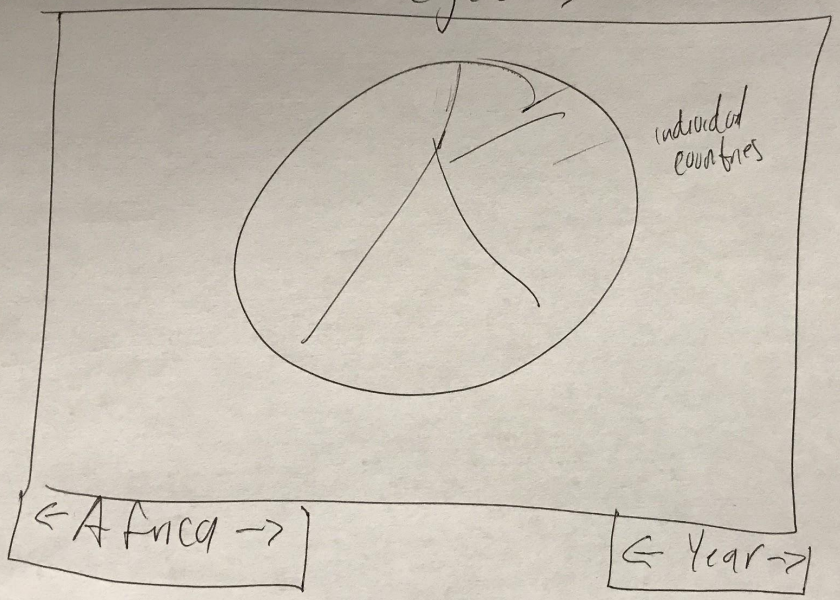


~~tot~~ Energy Supply by region Avg --

Visual 0

Tot Energy Supply

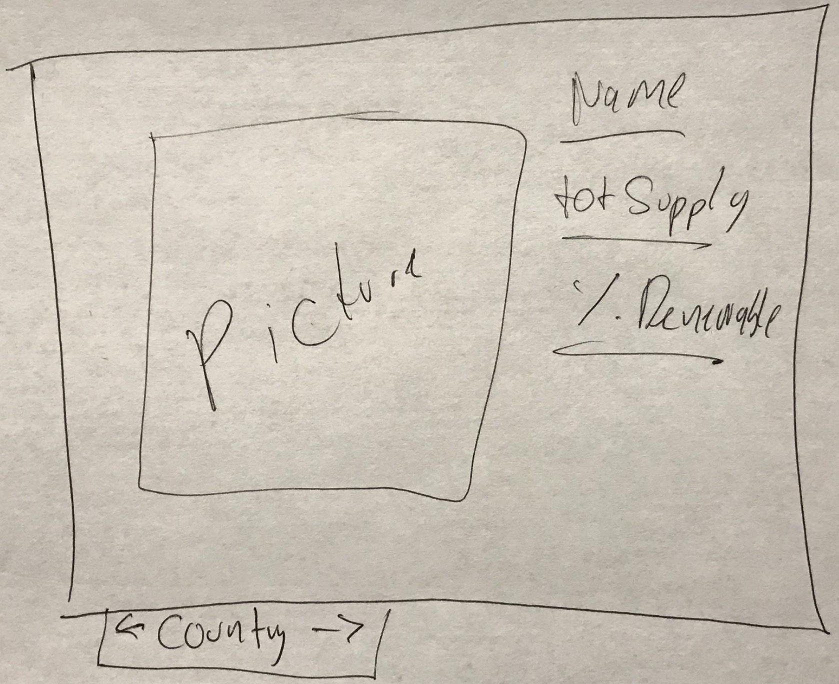
Regions



One knob changes geographic region

Time period of the region

Visuals 2013 Case study

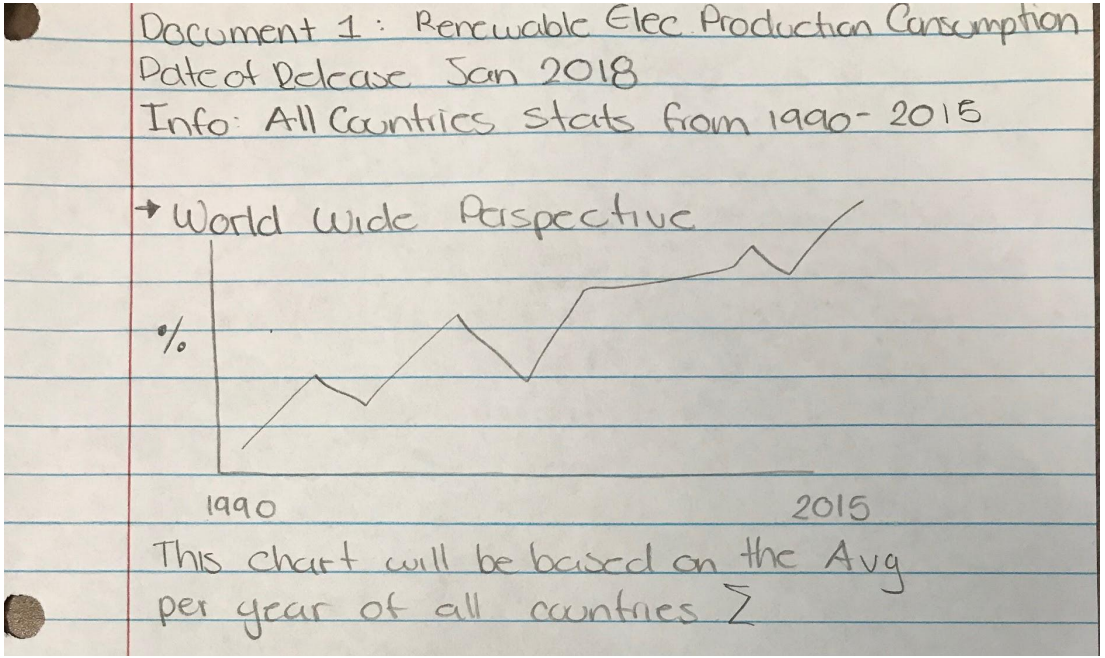


Sketches and Ideation

Part 3: Final Sketches

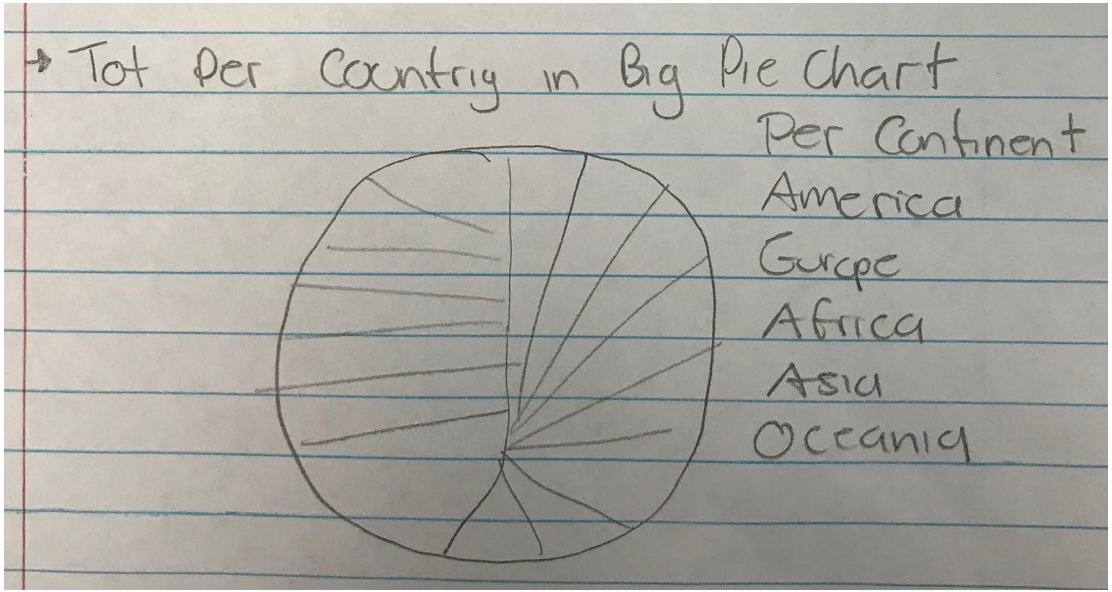
World Wide Perspective

This graph will represent the Renewable Electricity Production Consumption in terms of percentage at a worldwide perspective. I would add the values of all the countries and get the average per year.



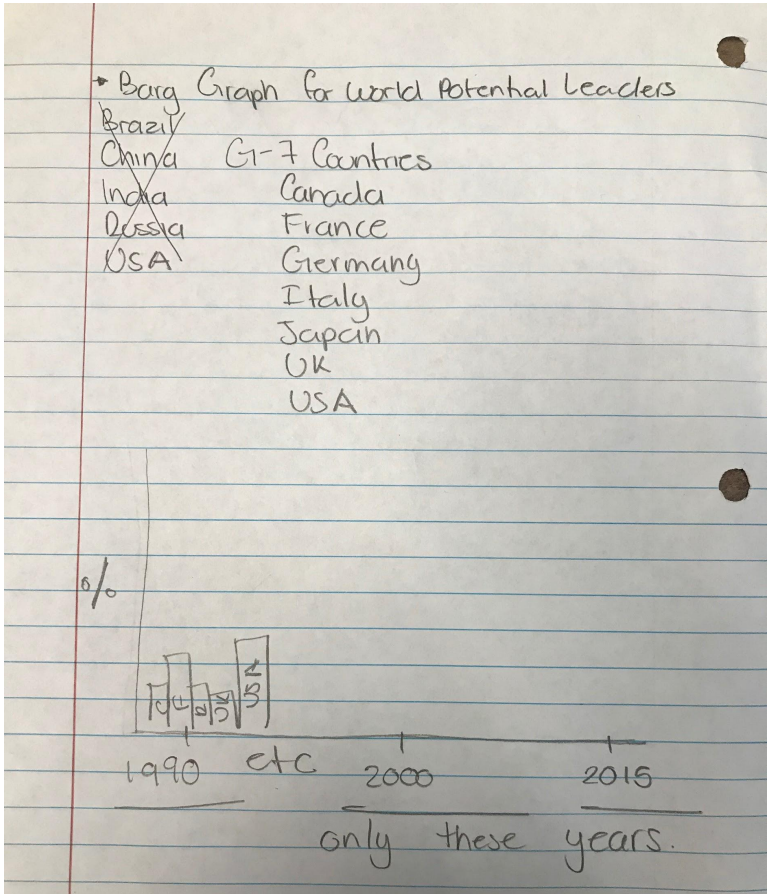
Total Production Per Country, Per Region Pie Chart

For this chart, I'm going to split the data and arrange the countries into their respective continent, then make a Pie Chart per continent based on the total production made of each country.



G-7 Comparison Bar Graph

In this chart i would be comparing the total production of the years 1990, 2000, and 2015 of the countries that form the G-7.



Pseduocode

Part IV

Part 4: Pseudocode

- Arrange the data in the files into the appropriate order.
- Create canvas in Processing.
- Write statements that will load the data into Processing.
- **Focus on Chart 1: World Wide Perspective Scatter Plot.**
- Grab in the appropriate values for the **single row - multiple column**: arrays.
- Plot the points.
- Compare the data visualization with the actual data to check for consistency and inconsistency.
- **Focus on Chart 2: Total Production Per Continent Pie Chart**
- *Repeat the steps below for each continent*
- Grab in the appropriate values for the **multiple row - single column**: arrays.
- Plot the points.
- Compare the data visualization with the actual data to check for consistency and inconsistency.

Part 4: Pseudocode

- **Focus on Chart 3: Comparison of Total Production of G-7 Countries Bar Graph**
- *Repeat the steps below for the sets of years: 1990, 2000, 2015:* each year elements are multiple row, single column
- Grab in the appropriate values for the **multiple row - single column**: arrays.
- Plot the points.
- Compare the data visualization with the actual data to check for consistency and inconsistency.

Step Back

Overview

In the transition of developing this project, I bumped through a lot of coding barriers; for example,

- Reading data
- Getting the data in the right format
- Making values dynamic
- Separating figures from one another
- Display of values

I went through seven diverse and different attempts to address the project and on the seventh time I got it to work.

In this processing book, I'm not going to include all seven folders, but I would be glad to talk about it if needed.

Finessing the System

- Instead of loading one big data file, I loaded three different files
  - Three different for() loops to extract the data from each file
    - Each loop would draw a specific graph
- Global functions used were:
  - Void setup()
  - Void draw()
  - Void mousePressed()
    - Issue in stopping the infinite loop
      - Command: noLoop() , resolves that issue
- A lot of math involved for dynamic positioning

Upcoming Improvements

- Make each table pop out individually with a mousePressed() that would work a long my format of code.
- Improve table data for better Labeling

DataVisualization | Table

```
// DELCARING UNIVERSAL VARIABLES
color[] dessert = {#9F9694, #791F33, #BA3D49, #F1E6D4, #E2E1DC};
color[] palette = dessert;

Table t1;
int rowCount;

Table t2;
int rowCount2;

Table t3;
int rowCount3;

int YY = 900;

color fillVal = color(126);
/******************************************************************************
******************************************************************************/
void setup(){
  // canvas
  size (1900, 900);
  smooth();

  //load table and verify table reading is appropriate

  // TABLE 1
  t1 = new Table("TableOne.tsv");
  rowCount  = t1.getRowCount();
  println("rowCount = " + rowCount);

  // TABLE 2
  t2 = new Table("Table3.tsv");
  rowCount2 = t2.getRowCount();
```

Final Code

Part 5

DataVisualization | Table

```
// DELCARING UNIVERSAL VARIABLES
color[] dessert = {#9F9694, #791F33, #BA3D49, #F1E6D4, #E2E1DC};
color[] palette = dessert;

Table t1;
int rowCount;

Table t2;
int rowCount2;

Table t3;
int rowCount3;

int YY = 900;

color fillVal = color(126);
/*****************************************************************************
*****************************************************************************/
void setup(){
  // canvas
  size (1900, 900);
  smooth();

  //load table and verify table reading is appropriate

  // TABLE 1
  t1 = new Table("TableOne.tsv");
  rowCount  = t1.getRowCount();
  println("rowCount = " + rowCount);

  // TABLE 2
  t2 = new Table("Table3.tsv");
  rowCount2 = t2.getRowCount();
```

```
  rowCount2 = t2.getRowCount();
  println("rowCount2 = " + rowCount2);

  // TABLE 3
  t3 = new Table("Last2.tsv");
  rowCount3 = t3.getRowCount();
  println("rowCount3 = " + rowCount3);

  //stroke(255);
  frameRate(2.6);
}
/*****************************************************************
*****************************************************************
int x = 300;
int y = 70;

void draw(){
  background(0);   // Clear the screen with a black background
  line(0, YY, width/2, YY);
  line(YY, 333, width/3, YY);

  for (int i = 0; i < width; i++) {
    float r = random(255);
    stroke(r);
    line(i, 0, i, height);
  }

  textSize(50);
  textAlign(CENTER);
  fill(palette[1]);
  text("To reveal the data, press your mouse",width/2,y);
  y = y + 100;

}
```

```
}
void mousePressed(){

  fill(fillVal);
  rect(25, 25, 50, 50);

  stroke (45);
  ellipse(200,200,50,50);
  rect(700,700,40,40);

  background (palette[0]);

  /********************************* TABLE 1 ****************
  //Aesthetics
  stroke (palette[3]);
  fill (palette[3]);
  textAlign(CENTER);
  textSize(20);
  text("World Wide R.E Production Percentage", width/4, 100);

  //DRAW CARTESIAN PLANE
  //x Axis
  stroke (3);
  textAlign (CENTER);
  line (100, 400, 890, 400);
  text ("Years", 450, 445);
  //Y Axis
  stroke (3);
  textAlign (RIGHT);
  textSize(15);
  line (100, 50, 100, 400);
  text("Percentage", 90, 250);

  //READ DATA FROM TABLE 1
```

```
  //READ DATA FROM TABLE 1
  //Non Dynamic to Dynamic data variables
  int Width = 120;
  int th = 420;

  //Get the Data
  for (int row = 0; row < rowCount; row ++){
    String dates = t1.getString(row, 0);
    float height = t1.getFloat(row, 1);
    String H = t1.getString(row, 1);
    println(dates + " " + height);
    height = height*10.9; // since my data values are so close from each other, I'm increasing their distance ratio
    //Draw the Data
    ellipseMode (CENTER);
    fill (palette[1]);
    stroke (12);
    ellipse (Width, (height - (height*2))+600, 9, 15); // math formula to be able to place the values in the right position

    textAlign(CENTER);
    textSize(10);
    text (dates, Width, th);
    textSize(10);
    text (H, Width,((height - (height*2))+600)-17 ); // same as above

    Width = Width + 30; //Re-declaring the variable Width, so it can space out my shapes from each other
  }
    /********************************* TABLE 2 **********************/
    //Aesthetics
    stroke (palette[3]);
    fill (palette[3]);
    textAlign (CENTER);
    textSize(20);
    text("G7 Comparison: Year 2015", width/4, 570);
```

```
132
133     //DRAW CARTESIAN PLANE
134     // X Axis
135     stroke (3);
136     textAlign (CENTER);
137     line (100, 800, 860, 800);
138     // Y Axis
139     stroke (3);
140     textAlign (RIGHT);
141     textSize (15);
142     line (100, 800, 100, 590);
143     text("Percentage", 90, 710);
144
145     //GET DATA FROM TABLE 2
146     //non dynamic to dynamic variables
147     int x = 150;
148
149     //Get Data
150     for (int row = 0; row < rowCount2; row++){
151       String dates = t2.getString(row, 0);
152       float height = t2.getFloat(row, 1);
153       String H = t2.getString(row, 1);
154       println (dates + " " + height);
155
156       //variables needed to draw
157       int y = 800;
158       int width = 80;
159
160       // Draw the data
161       rect(x, y, width, height-(height*2));
162       /*the subtraction from the line above what it does it to position the rectangle
163       in a regular postion. Computer coordinates are a 4th plane of a cartesian coordinate
164       so the calculation above translate the rectangle from being on a 4th plane to 1st plane*/
165       textSize(20);
```

```
    so the calculation above translate the rectangle from bei
    textSize(20);
    text(dates, x+50, 840);
    text(H, x+50, 720);
    x = x + 100; // increase the spacing of the graphs
  }

  /******************************** TABLE 2 ***************
  //Draw Cartesian Lines
  line (1000, 704, 1680, 704);
  line (1000, 704, 1000, 300);

  //GET DATA FOR TABLE THREE
  // non dynamic to dynamic variables
  int xx = 1050;
  int X = 1100;

  //Get Data
  for (int row = 0; row < rowCount3; row++){
    String continent = t3.getString(row, 0);
    float height = t3.getFloat(row, 1); //2015
    String H = t3.getString(row, 1); //2015
    String H2 = t3.getString(row, 2); //2000
    String H3 = t3.getString(row, 3); //1990
    float height2 = t3.getFloat(row, 2); //2000
    float height3 = t3.getFloat(row, 3); //1990

    //recalculate numbers to be able to display in this grid
    height = height /1000;
    height2 = height2 /1000;
    height3 = height3 /1000;

    println(continent + " " + H + " " + H2 + " " + H3);
```

```
    //DRAW DATA
    //static to dynamic values to draw the data
    int yy = 700;
    int width = 80;
    //2015
    fill(#ff8f00);
    rect(xx, yy, width, height-(height*2));
    //2000
    fill(#2b3a42);
    rect(xx, yy, width, height2-(height2*2)); // math formula to position rectangles in the right spot
    //1990
    fill(#f0f0df);
    rect(xx, yy, width, height3-(height3*2));
    xx = xx +100; //increase the spacing of the graphs
    //add labels to the graphs
    //continent
    textAlign(CENTER);
    textSize(12);
    text(continent, X, yy+20);
    //2015
    fill(#ff8f00);
    textSize(20);
    text(H,X,yy-390);
    //2000
    fill(#2b3A42);
    text(H2,X,yy-370);
    //1990
    fill(#f0f0df);
    text(H3,X,yy-350);
    X = X+100; // adds spacing to the text

    //add the years manually because theyre not provided in my data file
    fill(#f0f0df);
    text("1990",1200,780);
```

```
      fill(#2b3a42);
      text("2000",1300,780);
      fill(#ff8f00);
      text("2015", 1400, 780);
      // add label because it was not provided in data file
      fill(palette[3]);
      text("Electricity Supply Per Continent", 1300, 200);
    }
  noLoop(); // this will make the void Draw() function only run once

}

/*
void keyPressed(){
  if (key == CODED) {
    if (keyCode == UP) {
      fillVal = 255;
    } else if (keyCode == DOWN) {
      fillVal = 0;
    }
  } else {
    fillVal = 126;
  }

}
*/
```