UNIVERSIDAD NACIONAL DE SAN AGUSTIN

ESTRUCTURAS DE DATOS AVANZADAS

Práctica 02

Realizado por:

Pucho Zevallos, Kelvin Paul

2020

1. Elaboracion del Algoritmo QuadTree

1.1. Creacion del archivo main.html

```
<html>
<head>
<title> QuadTree </title>
    <script src ="p5.min.js" ></script>
    <script src ="quadtree.js"></script>
    <script src ="sketch.js"></script>
</head>
<body>
</body>
</html>
```

1.2. Creacion del archivo quadtree.js

En este archivo, tiene las clases con sus respectivos miembros y metodos por ejemplo:

- **La clase Point**.- Es el objeto con el cual nos ayudara para crear puntos en el plano
- **La clase Rectangle**.- El objetivo es generar un espacio rectangular, en el cual se encontraran los puntos.
    - El Metodo contains(point).- Se encarga de registrar si el punto se encuentra dentro de las dimensiones del rectangulo
    - El Metodo intersects(range).-Se encarga de verificar si el rectangulo actual se intersecta con otro rectangulo

```
class Point {
    constructor (x, y,userData){
        this.x = x;
        this.y = y;
        this.userData = userData;
    }
}

class Rectangle{
    constructor (x, y, w, h){
```

```
11      this.x = x;
12      this.y = y;
13      this.w = w;
14      this.h = h;
15      }
16
17      contains(point){
18          if(point.x >= this.x - this.w && point.x <= this.x + this
               .w && point.y >= this.y - this.h && point.y <= this.y
               + this.h){
19              return true;
20          }
21          return false;
22      }
23
24      intersects(range){
25          if(range.x - range.w > this.x + this.w || range.x + range
               .w < this.x - this.w ||
26              range.y - range.h > this.y + this.h || range.y +
                   range.h < this.y - this.h){
27              return false;
28          }
29          return true;
30      }
31
32  }
```

- **La clase Quadtree**.- Es la mas importante porque con ella se genera la parte logica de la creacion del Quadtree para generar una estructura de datos jerarquica descomponiendo de forma recursiva el espacio.Las propiedades de esta clase son los miembros y metodos las cuales son:

  Miembros Datos de la clase:

  - **boundary:** Contiene un objeto de tipo rectangle con sus dimensiones.
  - **capacity:** Contiene un entero, que indica el limite de hijos.
  - **points:** Almacena los puntos.
  - **divided:** Indica que si el Quadtree actual esta o no dividido.

  Metodos de la clase:

- **El Metodo subdivide()**.- Su funcion es dividir el espacio del Quadtree en 4 celdas de igual tamaño, creando rectangulos y heredando los metodos del padre de forma recursiva.A estas celdas se les llama: southeast, southeast, northeast y northwest.
  Tambien se confirma que el espacio actuales fue dividido.
- **El Metodo insert()**.- Su funcion es insertar los puntos en el arreglo de la clase Quadtree hasta llegar a su limite, entonces cuando sobrepasa dicho limite del padre se divide en 4 usando la funcion "subdivide".Finalmente los puntos que no pertenecen al padre son colocados a sus hijos, comprobando a que hijo pertenece.

- **El Metodo subdivide()**.- Encargado de renderizar los datos en forma grafica.

```
class QuadTree {
    constructor(boundary,n){
    this.boundary = boundary ;
    this.capacity = n;
    this.points = [];
    this.divided = false;
    }
    subdivide(){
        let x = this.boundary.x;
        let y = this.boundary.y;
        let w = this.boundary.w;
        let h = this.boundary.h;

        let se = new Rectangle(x+w/2,y+h/2,w/2,h/2);
        this.southeast = new QuadTree(se,this.capacity);

        let sw = new Rectangle(x-w/2,y+h/2,w/2,h/2);
        this.southwest = new QuadTree(sw,this.capacity);

        let ne = new Rectangle(x+w/2,y-h/2,w/2,h/2);
        this.northeast = new QuadTree(ne,this.capacity);

        let nw = new Rectangle(x-w/2,y-h/2,w/2,h/2);
        this.northwest = new QuadTree(nw,this.capacity);

        this.divided = true;

    }

```

```
30      insert(point){
31          if(!this.boundary.contains(point)){
32              return;
33          }
34          if(this.points.length < this.capacity){
35              this.points.push(point)
36          }else{
37              if(!this.divided){
38                  this.subdivide();
39              }
40              this.northeast.insert(point);
41              this.northwest.insert(point);
42              this.southeast.insert(point);
43              this.southwest.insert(point);
44          }
45      }
46
47
48      show(){
49          stroke(225);
50          strokeWeight(1);
51          noFill();
52          rectMode(CENTER);
53          rect(this.boundary.x,this.boundary.y,this.boundary.w*2,
                this.boundary.h*2);
54          if(this.divided){
55              this.northeast.show();
56              this.northwest.show();
57              this.southeast.show();
58              this.southwest.show();
59          }
60          for(let p of this.points){
61              strokeWeight(4);
62              point(p.x,p.y);
63          }
64      }
65  }
```

1.3. Creacion del archivo sketch.js

```
let qt;
let count = 0;

function setup(){
    createCanvas(400,400);
    let boundary = new Rectangle(200,200,200,200);
    qt = new QuadTree(boundary,4);
    console.log(qt);
    for (let i = 0; i < 3; i++){
        let p = new Point(Math.random()*400,Math.random()*400);
        qt.insert(p);
    }
    background(0);
    qt.show();
}
```

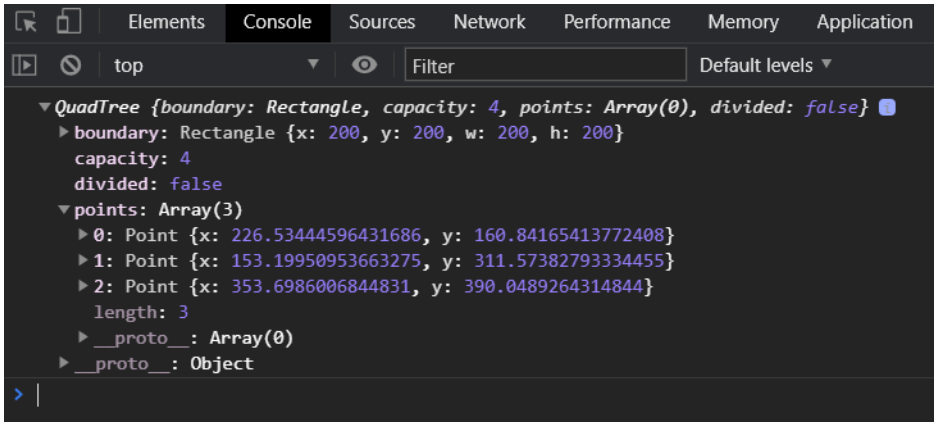
Este archivo es el encargo de colocar las dimensiones del primer Quadtree con sus respectivos hijos.

1.3.1. Ejemplos 1:

- **Resultados en el navegador:** Se muestra los resultados en el navegador web, donde se observa que los 3 puntos estan dispersos aleatoriamente y que el Quadtree aun no esta dividido porque los puntos son menores que la capacidad.

1.Visualizacion del Quadtree con 3 puntos.

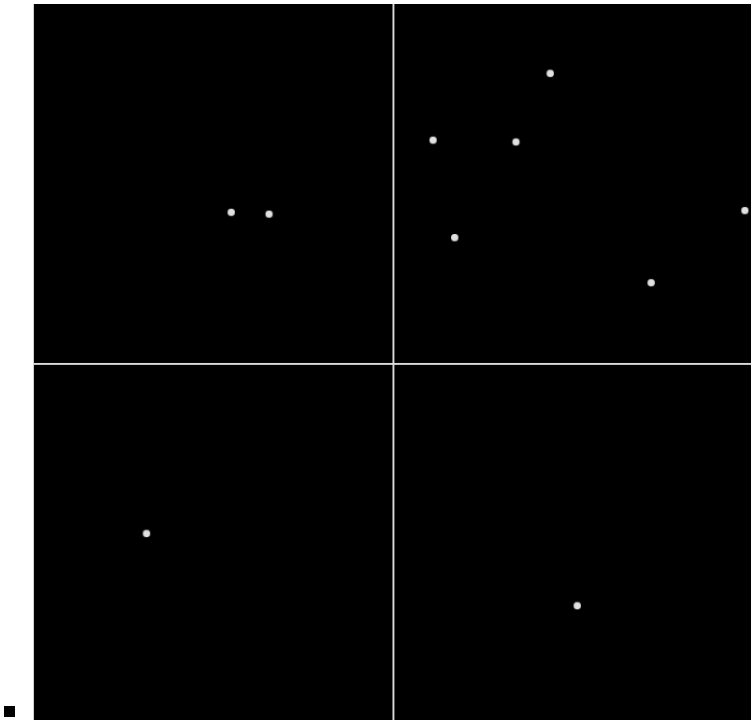
- **Resultados en la *console* del navegador:** Se puede observar los resultados de las dimensiones del Quadtree,su capacidad de puntos que este caso tiene 3 puntos, tambien indica que el Quadtree no esta dividido



2.Vista a la Console de las opciones de desarrollador del navegador Web.

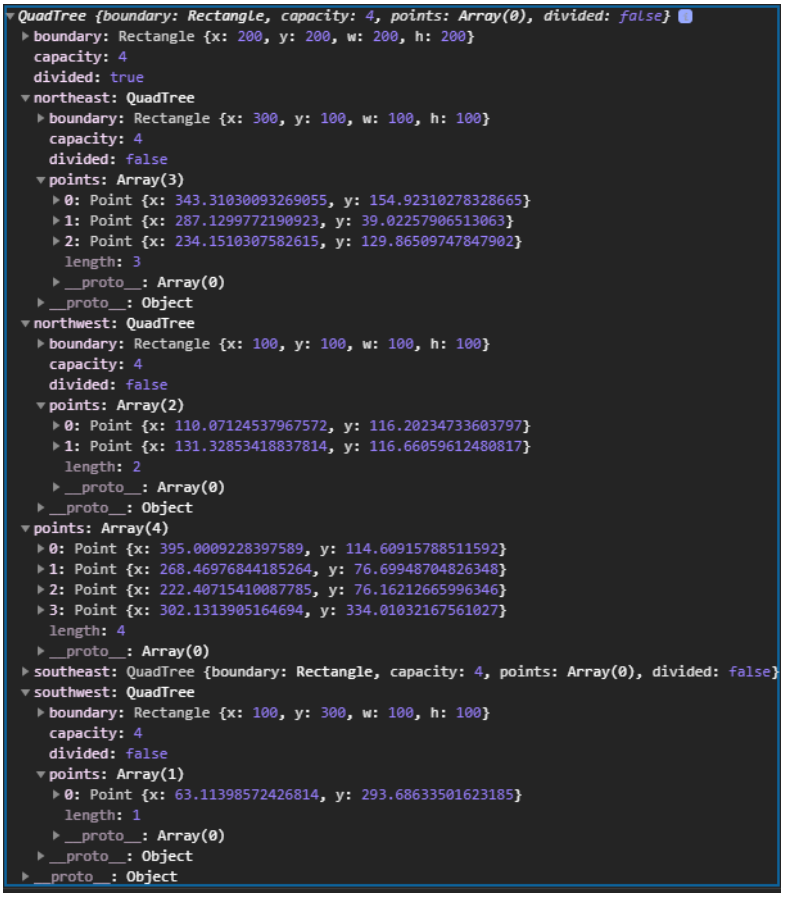
1.3.2. Ejemplos 2:

Esta vez se procedera colocar 10 puntos para tener una mejor visualizacion de la posicion de los puntos en el Quadtree.**Resultados en el navegador:** Primeramente se observa a ver los 10 puntos dispersos y la division que se hizo al Quadtree padre por superar el limite de puntos.



3.Visualizacion del Quadtree con 10 puntos.

- **Resultados en la *consola* del navegador:** Se puede observar que el padre Quadtree fue dividido en 4 celdas las cuales son sus hijos y contiene los puntos segun su capacidad, los demas puntos son añadidos a los hijos respectivos. Cada punto se verifica a que hijo le pertenece. Por ejemplo en la Figura 4 los siguientes hijos contiene puntos y estos son: northeast tiene 3 puntos, el hijo northwest tiene solo un punto, el hijo southwest tiene solo un punto y finalmente el hijo southeast no tine ningun punto. Notese que los 4 hijos no fueron divididos.



4.Visualizacion del Quadtree con 10 puntos.

1.4. Modificacion del archivo sketch.js

La modificacion del archivo fue para poder realizar una insercion de puntos en el Quadtree, manteniendo presionado el click del mouse e insertando puntos uno por uno.

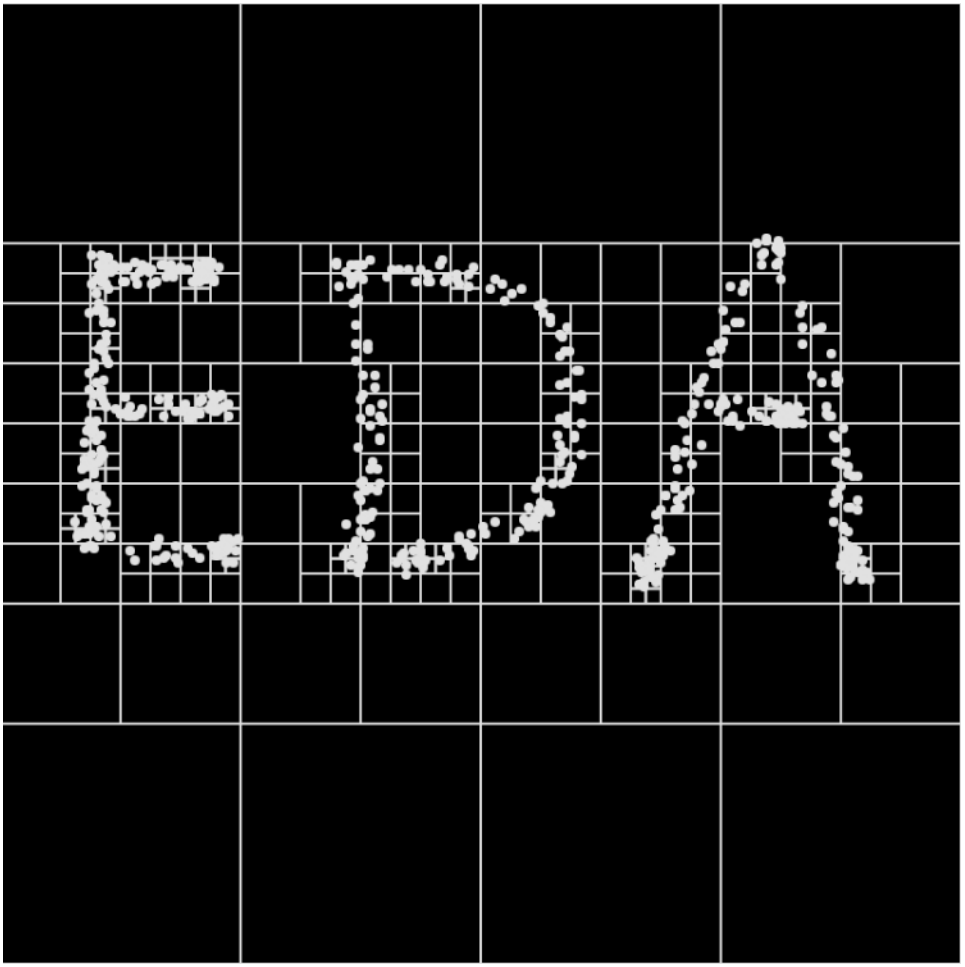
```
let qt;
let count = 0;

function setup(){
    createCanvas(400,400);
    let boundary = new Rectangle(200,200,200,200);
    qt = new QuadTree(boundary,4);
}
```

```
9
10 function draw(){
11     background(0);
12     if(mouseIsPressed){
13         for (let i = 0; i < 1; i++){
14             let m = new Point(mouseX+random(-5,5),mouseY+random
                   (-5,5));
15             qt.insert(m);
16         }
17     }
18     background(0);
19     qt.show();
20 }
```

1.4.1. Resultados

Se puede apreciar que gracias a la modificacion del codigo en el archivo podemos representar alguna figura como se muestra a continuacion. Cada ves que se mantiene presionado el click del mouse tiende superar la capacidad de puntos del quadtree, entonces se realiza la division del espacio en partes iguales que se generan recursivamente.



5.Visualizacion del Quadtree con n puntos.

2. Repositorio

`https://github.com/kpzaolod6000/Cursos_2020_II/tree/master/EDA/semana_2/quadtree`