

# Final Project

## Realtime

เรื่อง

Rainfall simulation

เสนอ

อาจารย์ ณัฐพงศ์ ชินรเนศ

จัดทำโดย

นาย กิติพงษ์ ศิริเรืองสกุล 5731008221

นาย ปранิน โลกโบว์ 5731071721

นาย อภิรุจ ชุ่มวัฒนา 5731113421

รายงานนี้เป็นส่วนหนึ่งของวิชา 2110594 Realtime Computer Graphic And Physics Simulation ภาคเรียนที่ 1 ปีการศึกษา 2559 คณะวิศวกรรมศาสตร์ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย

# คำนำ

รายงานวิชานี้คือการนำเสนอรายละเอียดเกี่ยวกับโปรเจค ชื่อ Rainfall simulation โดยโปรเจคชื่นี้เป็น Final project ของวิชา Realtime Computer Graphic And Physics Simulation โดยเป็นการนำเสนอที่เรียนมาในเรื่องต่างๆ มาประยุกต์ใช้แล้วนำมาสร้างสรรค์ผลงานของตนเอง โดยในส่วนของงานที่กลุ่มของข้าพเจ้าทำนั้น ได้ศึกษาความรู้ในเรื่องต่างๆ เพิ่มเติม และต่อยอดจากสิ่งที่เรียน พร้อมทั้ง ศึกษาข้อมูลของผลงานบางชิ้นที่มีคนเคยสร้างมาแล้วเป็นความรู้ แล้วบางส่วนนำมาต่อยอด ผนวกกับสิ่งที่เรียนมาในเรื่องต่างๆ แล้วนำมาสร้างสรรค์เป็นผลงาน โดยรายงานชื่นี้จะแสดงให้เห็นถึง รายละเอียดของ ผลงานที่ทำ และ ขั้นตอนการทำ การศึกษาหาข้อมูล และ ซึ่งจะแสดงให้เห็นถึงการนำเสนอความรู้ในการทดลอง ในเชิงพิสิกส์มา ผนวกกับ ความรู้ ที่ได้เรียนมา และส่วนที่ศึกษาจากผลงานที่มีอยู่แล้ว ว่าจุดไหนเป็นจุดไหนบ้าง พร้อมทั้งบอก ปัญหาที่พบ และ แนวทางการต่อยอดผลงาน กลุ่มของข้าพเจ้าจึงหวังเป็นอย่างยิ่งว่ารายงานชื่นี้ จะมีประโยชน์ต่อผู้ที่สนใจไม่มากก็น้อย

คณะผู้จัดทำ

## สารบัญ

ที่มาและความสำคัญ	3
เกี่ยวกับ	4
ขอบเขต	4
ผลงานที่ได้	5
รายละเอียด	6
ปัญหาที่พบ	15
แนวทางการพัฒนาต่อ	15
หน้าที่ของสมาชิก	16
วีดีโอ	16
บรรณานุกรม	17

## ที่มาและความสำคัญ

ทุกวันนี้เรารสามารถจำลองสิ่งที่เกิดขึ้นจริงตามธรรมชาติอุปกรณ์ในรูปแบบต่างๆ ผ่านการใช้คอมพิวเตอร์ได้ เพราะเทคโนโลยีทุกวันนี้ก้าวหน้าไปอย่างมาก many จึงมีการจำลองสิ่งต่างๆ ของมาให้เห็นกันอยู่มาก แต่ธรรมชาติของโลกเรานั้นไม่มีขีดจำกัด เพราะฉะนั้นยังเปิดกว้างมากสำหรับศาสตร์นี้ในการรอให้นักพัฒนามาจำลองสิ่งต่างๆ ซึ่งหลายอย่างในธรรมชาตินั้นเกิดขึ้นและเปลี่ยนแปลงตลอดเวลา การจำลองจึงต้องทำแบบ Real Time คือตลอดเวลานั้นเอง ซึ่งในปัจจุบันนี้ก็เช่นกัน เราจะจำลองสิ่งที่เกิดขึ้นแบบตลอดเวลา คือ สถานการณ์ฝนตก ซึ่งคนส่วนมากจะคุ้นชินกับสถานการณ์ดังกล่าวเป็นอย่างมาก เพราะเกิดเป็นประจำในช่วงฤดูฝน ปกติเราสามารถมองเห็นฝนโดยตรงหรือผ่านตัวกลางต่างๆ ซึ่งพฤติกรรมของน้ำฝนก็จะแตกต่างกันไปตามตัวกลาง โดยเราจะจำลองการฝนตกพร้อมทั้งปรากฏการณ์ที่ตามมากับฝน เช่น ฟ้าผ่า ฟ้าร้อง ซึ่งจะช่วยให้เราสามารถแสดงสถานการณ์เหล่านี้ให้คนที่ไม่เคยเห็น เห็นได้ รวมถึงนำไปใช้เคราะห์พยากรณ์เพื่อศึกษาข้อมูลบางสิ่งบางอย่างได้อย่างละเอียดโดยมิต้องเห็นจริงๆ อีกทั้งยังเอาไปใช้ในงานที่ใช้เพื่อความสวยงามได้ เช่น ภาพนิทรรศหรืออนิเมชั่นต่างๆ และยังทำเป็นสื่อความรู้ให้กับคนศึกษาได้ด้วย เช่น เด็กในต่างแดนที่ไม่มีโอกาสได้เห็น ให้ได้ศึกษาและเพลิดเพลินไปกับมัน



ภาพจาก <http://www.publicdomainpictures.net/view-image.php?image=150629&picture=raindrops>

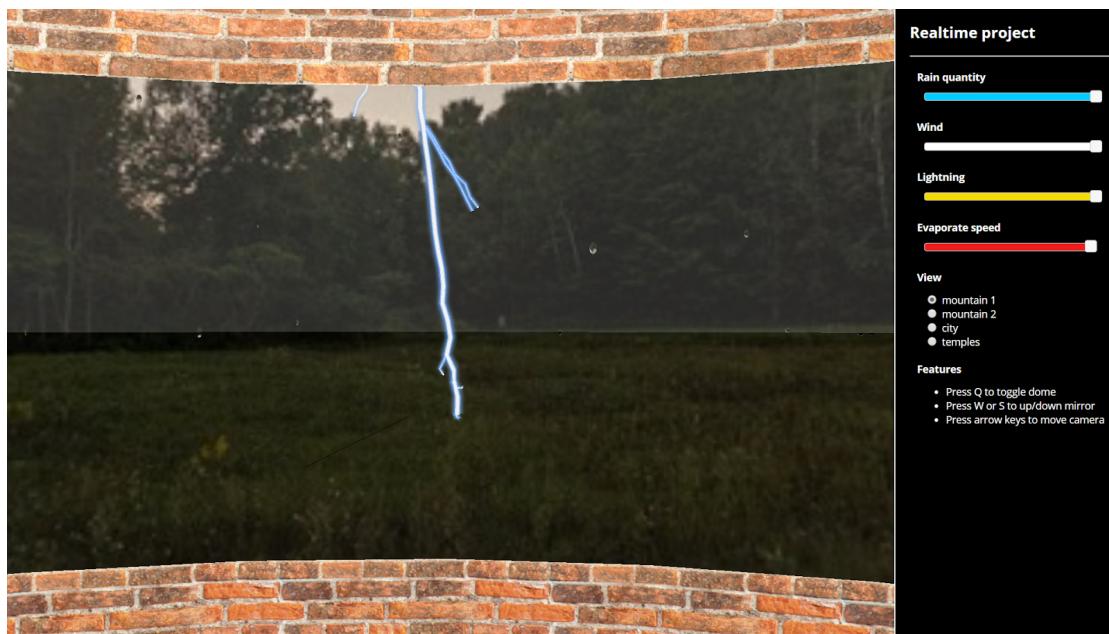
## Rainfall simulation เกี่ยวกับอะไร

โปรเจคนี้จะจำลองสถานการณ์ ฝนตก และอีกสถานการณ์หนึ่งก็จะเป็นสถานการณ์ที่พ่วง เรายุคเห็นกันตลอด และน่าจะได้มองมันบ่อยครั้ง คือหยดน้ำที่ไปติดกับกระจกรถหรือกำแพงแล้ว ค่อยๆหล่นลงมา เพราะทุกวันที่ฝนตกหลายครั้งเราจะอยู่ในที่ร่มแล้วมองรอฝนให้หยุดตก บาง คนก็คงจะเห็นพฤติกรรมน้ำที่ตกลงมาแล้วรู้สึกมันดูสวย ดูเพลิน แต่ถ้าเรามองสถานการณ์นี้จากที่ เราเห็นกันจนชินจนริงๆแล้วนั้น แท้จริงแล้วพฤติกรรมเป็นยังอย่างไรกันแน่ ทำไมบางจุดมันไปเกาะอยู่ด้วยกัน และทำไมน้ำบางหยดบางเม็ดถึงหล่นลงมา บางครั้งก็หยุด และเราจะมาดูผ่านสถานการณ์จำลองกัน พิรุ่งทั้งสิ่งที่เกิดขึ้นหลังจากเรามองลอดกระจกไปบางครั้งเป็นผ่านฟ้าที่ แปรปรวนและบางที่จะมีฟ้าผ่าลงมาด้วยเราก็จะได้เห็นกัน

### ขอบเขต

- เป็นการจำลองเหตุการณ์ฝนตก ที่จะประกอบไปด้วย เม็ดฝนบนกระจกใส สายฝน ฟ้าผ่า ฟ้าร้อง ลม เท่านั้น ให้ใกล้เคียงกับความเคียงด้วยทฤษฎีพื้นฐานทางฟิสิกส์

## ผลลัพธ์ของงาน



### 1. จำลองฝนและลม

- กำหนดความหนักเบาของฝนได้
- กำหนดทิศทางและความแรงของลมได้
- สามารถหมุนดูสถานที่ได้ทั้ง 360 องศา
- สามารถเปลี่ยนสถานที่ได้
- มีเสียงลมและฝน ตามความแรงของลมและฝนที่กำหนด
- เม็ดฝนมีการ refraction ตามสถานที่

### 2. จำลองฟ้าผ่า

- กำหนดความถี่ของฟ้าผ่าได้
- มีเสียงของฟ้าผ่าเมื่อฟ้าผ่า
- มีการแตกกิงเมื่อมีอนจริง

### 3. จำลองโดมกระจก

- จำลองฝนให้ตามกระจากใส
- จำลองฝนติดขอบกระจาก

## รายละเอียดของงานที่ทำ

### ขั้นตอนการดำเนินงาน

1. ศึกษาพฤติกรรมของฝน และน้ำที่หลอยู่บนวัสดุต่างๆ
2. ศึกษาปรากฏการณ์ฟ้าฝ่า พื้นท้อง ที่เกิดเมื่อเวลาฝนตก
3. ศึกษาการใช้ Html CSS เพื่อให้ผู้ใช้สามารถปรับแต่งค่าต่างๆได้
4. ศึกษาการใช้ library three.js ในการสร้างวัตถุ และองค์ประกอบต่าง ๆ เช่น ฝน ฉาก ฟ้าฝ่า
5. ศึกษาและใช้งาน library จาก <https://github.com/jeromeetienne/threex.geometricglow> ในการทำ สี รอบๆเส้นสายฟ้า
6. สร้างฉาก 360 องศา โดยใช้ภาพ equirectangular
7. ทำการสร้าง model แบบจำลองน้ำฝนที่มีคุณสมบัติ การหักเห
8. ทำการสร้าง simulate รูปร่างและสีของสายฟ้า

### องค์ประกอบหลัก

1. สถานที่



ภาพ equirectangular จาก <http://fly.mpi-cbg.de/~saalfeld/Projects/panorama.html>

เพื่อให้สามารถเคลื่อนไหวและเห็นมุมต่างๆของสถานที่ได้ชัดเจน จึงได้เลือกใช้ภาพ equirectangular ซึ่งจะทำให้สามารถมองภาพได้ครบทั้ง 360 องศา ซึ่งได้เลือกมาทั้งหมด 4 ภาพ เพื่อใช้เป็นภาพประกอบฝนที่ตกลงมา

```

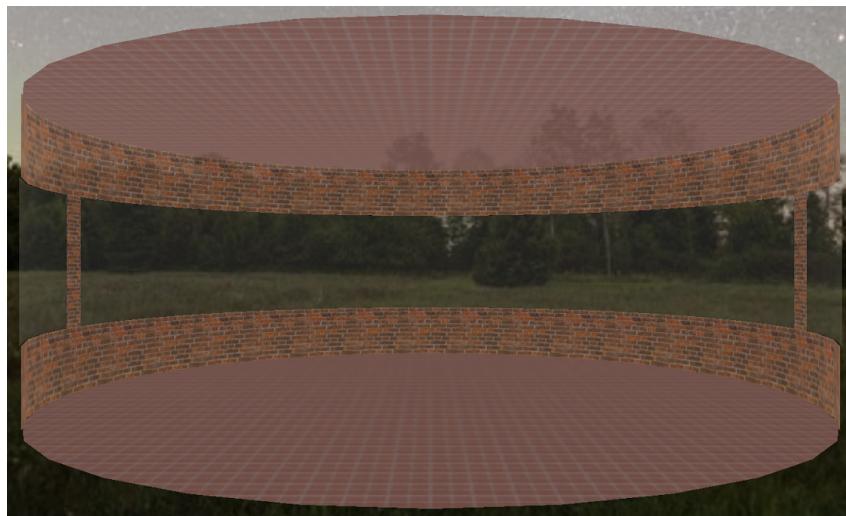
///////////
// view
var geometry = new THREE.SphereGeometry( 60, 60, 40 );
geometry.scale( - 1, 1, 1 );
var material = new THREE.MeshBasicMaterial( {
    map: new THREE.TextureLoader().load( './resources/img/1/eq.jpg' ),
} );
this.global = new THREE.Mesh( geometry, material );
this.scene.add( this.global );
/////////

```

code ในการใส่ texture ให้กับทรงกลม

โดยการใส่ภาพนั้นเป็น texture ให้กับพื้นผิวของทรงกลม โดยใช้ parameter map ของ threejs แต่โดยปกติการทำแบบนี้จะเป็นการใส่ผิวรอบนอกของทรงกลม จึงทำให้ต้องสั่ง geometry.scale( -1,1,1); เพื่อเป็นการสลับให้ texture ของเรานั้นไปอยู่ด้านในของทรงกลมแทน

## 2. โดม



ภาพโดมในงานที่สร้างขึ้น

โดมซึ่งเป็นที่ผู้ใช้สามารถหมุนหรือเคลื่อนย้ายกล้องไปมาภายในโดม ถูกสร้างจาก

- Cylinder 360 องศา 2 อัน เพื่อเป็นกำแพงกั้นทางด้านบนและด้านล่าง
- Cylinder 5 องศา 2 อัน สำหรับเป็นเสาค้ำด้านบนกับด้านล่าง
- Circle 2 วง ปิดด้านบนและด้านล่าง

```

///////////////////////////////////////////////////////////////////
// create cylinder_up for dome
var texturewall = new THREE.TextureLoader().load('./resources/img/wall.jpg');
texturewall.wrapS = texturewall.wrapT = THREE.RepeatWrapping;
texturewall.repeat.set( 30, 1 );

var cylinderup_geo = new THREE.CylinderGeometry(Manager.getWidth() / 2 - 0.04, Manager.getWidth() / 2 - 0.04, parse
var cylinderup_mat = new THREE.MeshLambertMaterial({map: texturewall, transparent: true, opacity: 0.9});
var cylinderup = new THREE.Mesh(cylinderup_geo, cylinderup_mat);
cylinderup.position.set(0, 1.5, 0);
cylinderup.scale(-1, 1, 1);

this.objectdome.add(cylinderup);
///////////////////////////////////////////////////////////////////

```

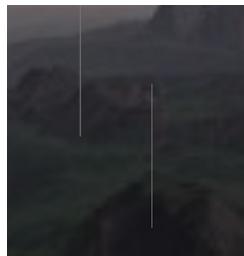
code สำหรับการใส่ cylinder ด้านบน

Code ข้างต้นเป็นตัวอย่างในสี Cylinder ลงเป็นส่วนประกอบของโดม โดยการโหลด texture ที่สามารถ repeat แล้วลวดลายต่อกันได้ และทำการกำหนดจำนวนครั้งในการ repeat แต่ละแกนด้วยคำสั่ง repeat.set จากนั้น ทำการใส่ texture ให้กับ Cylinder

### 3. น้ำฝน

ตัวของน้ำฝนนั้นจะมีเสียงประกอบฝนตกด้วย โดยถ้ามีตากหนักจะทำให้มีเสียงฝนตกที่ดังขึ้น และถ้าตกเบาเสียงก็จะเบาลงตามที่ผู้ใช้ได้ตั้งค่าไว้ โดยตัวของน้ำฝนนั้นแบ่งออกเป็น 3 แบบ ดังนี้

#### 1. สายฝน



สายฝนจะเป็นเส้นตรงที่จะเป็น background ให้ตลอดเมื่อมีการตั้งให้มีฝนตกโดยจะมีจำนวนมากขึ้นตามระดับความแรงของฝนที่ผู้ใช้เลือก โดยแต่ละเส้นจะปรากฏเพียงครั้งเดียวและถูกลบออกไป เพื่อไม่ให้เกิดการสะสมจำนวนของสายฝนจนทำให้ใช้เวลานานในการ update ภาพในแต่ละ frame และ เพื่อเป็นการลดจำนวนของ object จะทำการ render ฝน ลงไปเฉพาะรอบ ๆ โดมฯหรือจุดศูนย์กลางเท่านั้น

```

function addrainbg(nums) {
    while(nums--){

        var start = math.complex(Manager.randomWidth(), Manager.randomHeight());
        var length = Manager.random(0.7,1.5);
        var seta = manager.wind * Math.PI / 360.0;
        var end = math.add(start,math.multiply(length,math.complex(math.sin(seta),-1*math.cos(seta))));

        var globalR = Manager.getWidth()/2;
        var startz = Math.sqrt(globalR*globalR - start.re*start.re);
        var mul = Manager.random(-1,1)>=0 ? 1 : -1;
        startz *= mul;
        var endz = Math.sqrt(globalR*globalR - end.re*end.re);
        endz *= mul;

        var geometry = new THREE.Geometry();
        geometry.vertices.push(new THREE.Vector3(start.re, start.im, startz));
        geometry.vertices.push(new THREE.Vector3(end.re, end.im, endz));

        particle = new THREE.Line( geometry, manager.sphereMaterial );
        rains.push(new Rain_back(particle));
        parent.add(particle);
    }
}

```

การสร้างเส้นแต่ละเส้นจะใช้วิธีการซึ่งจุดเริ่มของเส้น และทำการสุ่มความยาวของเส้น และทำการหมุน  $(0, -1)$  โดยการคูณด้วยเมตริกซ์การหมุน 2D กำหนดให้เป็น  $x$  และทำการหาจุดปลายของเส้นโดยเอา จุดเริ่ม  $+ x * \text{ความยาวของเส้น}$  และทำการหา แกน  $z$  ของเส้นจากสมการวงกลม  $x^2 + z^2 = r^2$

## 2. เม็ดฝนขนาดใหญ่



ฝนใน layer นี้จะมีลักษณะเป็นเม็ด ซึ่งจะมีขนาดแตกต่าง และรูปทรงแตกต่างกันไป ซึ่งจะทำการ simulate เนพาริวกระจากของโถมเท่านั้น และเมื่อฝนใหญ่หลุดขอบกระจาดไปจะทำการเอาออก เพื่อไม่ให้จำนวนฝนเยอะเกินไป และตัวเม็ดสายฝนนี้จะมีการตั้งค่า refraction ไว้เป็น 0.75 เนื่องจากด้านนี้หักเหของแสงในอากาศ คือ 1.00 และ ด้านนี้หักเหของแสงในน้ำคือ 1.33 และใช้ cubemap เป็นภาพที่ปราภูบเน็ตด้านน้ำ โดยแปลงภาพสถานที่จาก equirectangular เป็นภาพ 6 view จาก <http://gonchar.me/panorama/> โดยรูปทรงของน้ำจะมีการตั้งไว้ทั้งหมด 3 แบบ คือ

- Octahedron
- Icosahedron
- Sphere

```

function addrain(nums,sizerain,offset) {
    if(manager.dome) {
        while(nums--){
            var radius = Manager.random(0,sizerain)+offset;

            var geometry;
            var type = parseInt(Manager.random(0,2.99));
            var detail = parseInt(Manager.random(1,2.99));
            if(type==0)
                geometry = new THREE.OctahedronGeometry( radius , detail);
            else if(type==1)
                geometry = new THREE.IcosahedronGeometry( radius , detail);
            else
                geometry = new THREE.SphereGeometry( radius , detail+1);

            particle = new THREE.Mesh( geometry , manager.sphereMaterial );

            particle.position.x = Manager.randomWidth();
            particle.position.y = Manager.random(manager.mirror.position.y-1,Manager.getHeight()/2);

            var globalR = Manager.getWidth()/2;
            particle.position.z = Math.sqrt(globalR*globalR - particle.position.x*particle.position.x);
            particle.position.z *= Manager.random(-1,1)>=0 ? 1 : -1;
            particle.scale.y *= 1.4;

            if(radius>=0.015)
                rains.push(new Rain_drop(particle,radius,Manager.random(0.005,0.015)));
            else
                rains.push(new Rain_drop(particle,radius,0));
            parent.add(particle);
        }
    }
}

```

การสร้าง ทำการสุ่มขนาดและรูปทรงและตำแหน่งของเม็ดฝน โดย ตำแหน่งจะทำการสุ่มเฉพาะแกน x,y เท่านั้น และทำการคำนวณหาแกน z จาก สมการวงการแบบเดียวกับสายฝน ซึ่งถ้ามีขนาดเล็กจะถูกจะอยู่ในฝนแบบที่ 3 (เม็ดฝนขนาดเล็ก)

```

this.object.position.y -= Manager.random(max(this.speed*(manager.rain+10)/30-0.01

for(var i=rains.length-1;i>=0;i--){
    if(rains[i].name == this.name || rains[i] instanceof Rain_back) continue;
    if(this.object.position.y < rains[i].object.position.y) continue;
    if(intersect(this,rains[i])){
        rains[i].active = false;
        this.speed += rains[i].radius;
        var new_r = Rain.getRadiusFromVolume(this.volume()+rains[i].volume());
        this.object.scale.z = new_r/this.radius - (1 - this.object.scale.x);
        this.object.scale.y = new_r/this.radius + (this.object.scale.y-1);
        this.radius = new_r;
        if(rains[i].break !=0){
            this.breaker = 1;
            this.speed = 0.2;
        }
    }
}

if(this.object.scale.y < 1.8){
    var sca = Manager.random(0.999,1.0);
    this.object.scale.x *= sca;
    this.object.scale.y *= 1.0/sca;
}

this.object.scale.x -= Manager.random(0,manager.sun/10000);
this.object.scale.y -= Manager.random(0,manager.sun/10000);
if(this.object.scale.x <=0 || this.object.scale.y <= 0)
    return true;

if(this.object.position.y < -(Manager.getHeight()/2))
    return true;

return false;

```

การอัพเดท ทำการเลื่อนเม็ดฟันตามแนวแกน y หรือตามแรงโน้มถ่วงตามค่าความเร็วของหยดน้ำที่ได้ทำการสูญเสียตั้งแต่ตอนสร้าง และทำการตรวจสอบว่า มีการชน กับหยดน้ำอื่นๆหรือไม่ ถ้าชน ให้ทำการรวมสองเม็ดเข้าด้วยกัน และเพิ่มความเร็วของเม็ดฟัน และทำการขยาย scale ของเม็ดฟันในแนวแกน y เพิ่มทุกๆ การ update เนื่องจากเวลาหน้าไฟลจะมีส่วนที่ไม่ได้สัมผัสพื้นผิวซึ่งจะดึงให้ฟันกล้ายเป็นรูปเรียวกามากกว่าทรงกลม และถ้าเกินขอบกระเจาให้ทำการลบออก

การระเหย จะทำการลดขนาด scale ของแกน x,y ของเม็ดฟันลง ถ้าผู้ใช้มีการตั้งค่าความเร็วในการระเหยมาก ก็จะทำให้เม็ดฟันนั้นระเหยเร็วขึ้น

### 3. เม็ดฟันขนาดเล็ก

จะคล้ายกับเม็ดฟันขนาดใหญ่ แตกต่างกันตรงที่ความเร็วในการไฟลของเม็ดฟันนั้นจะเท่ากับ 0 หรือคือจะอยู่นิ่งกับที่ติดกับกระเจา

### 4. สายฟ้า



สายฟ้านี้เกิดจากเหตุการณ์ที่ลมพัดด้วยความเร็วสูง จะทำให้เกิดการ ขัดสีกับผิวพื้นโลกและสิ่งก่อสร้างต่าง ๆ จึงทำให้ โมเลกุลของลม ได้รับประจุลบ (อิเล็กตรอน) เมื่อลมได้รับอิเลคตรอน และไปถ่ายเทให้กับด้านล่างของก้อนเมฆ เมื่ออิเลคตรอน รวมตัวกันที่ด้านล่างของก้อนเมฆมากขึ้นเรื่อยๆ จนถึงขนาดหนึ่ง แรงผลักกระหัวง อิเลคตรอนบนก้อนเมฆ จะผลักให้อิเลคตรอนที่ผิวโลกแยกตัวออกจากประจุบวก จนทำให้ผิวโลกมีประจุเป็นบวกเพิ่มมากขึ้น ประจุลบนก้อนเมฆ จะผลักกันเองและขณะเดียวกันจะถูกดูดโดยประจุ

บวกจากพื้นโลก จึงทำให้มีประจุลับเคลื่อนที่ลงสู่พิวโลก เนื่องจากแรงผลักจากด้านบนและแรงดูดจากด้านล่างซึ่งการที่ ประจุเคลื่อนที่จาก ก้อนเมฆไปสู่พิวโลก จะเรียกว่า **ฟ้าผ่า**

สายฟ้าที่ทำการจำลองขึ้นจะเป็นการสร้างเส้นหลัก 1 เส้นที่มีความเข้มสูง และทำการสุมแต่ก็จะระห่วงเส้น ซึ่งแต่ละก็จะสามารถแต่ก็ต่อไปอีกได้ แต่ก็จะมีความเข้มน้อยกว่าเส้นกว่า ในความจริงแล้วสายฟ้าจะมีการบิดมากกว่านี้ แต่เนื่องจากถ้าจะคำนวนอาจต้องใช้เวลามากขึ้นจนไม่สามารถทำได้ใน Realtime และในงานชิ้นนี้ได้มีการเล่นเสียงของฟ้าผ่า โดยจะเล่นเสียงตามความถี่ของสายฟ้าที่ผู้ใช้ได้ตั้ง

#### การสร้างลักษณะเส้น อ้างอิง code และสมการ จาก

<https://gamedevelopment.tutsplus.com/tutorials/how-to-generate-shocking-lightning-good-2d-lightning-effects--gamedev-2681> โดยการนำ code และสมการบางส่วนมาทำให้อยู่บน threejs และทำการเพิ่มในส่วนของการแต่ก็ซ้อนสองชั้น และทำให้ก็มีความเข้มและความยาวที่น้อยลงกว่าเดิม โดยมีหลักการดังนี้ คือ สุ่มจุดเริ่มต้น จุดจบจำนวนชั้น ทำการสุมจุดระหว่างจุดเริ่มต้นและจุดจบมาจำนวนหนึ่งและทำการสุมการขนาดในการเลื่อนจุดนั้นไปในแนวตั้งจากกับเส้น โดยพยายามคุณไม่ให้ต่างจาก การเลื่อนในครั้งก่อนๆมาก หลังจากนั้นถ่ายงไม่ถึงจำนวนชั้นที่สูงไว้ในตอนต้น ให้ทำการสุมจุดบนเส้นเพื่อใช้ในการแต่ก็ และทำการสร้างเส้นใหม่ ที่มีจุดเริ่มต้นคือ จุดนั้น และจุดจบคือจุดเดิมที่มีการหมุนออกด้านข้าง

#### การสร้างสีเส้น ใช้ library จาก

<https://github.com/jeromeetienne/threex.geometricglow> ซึ่งเป็น library ในการทำ สีรอบๆตุ่นด้วยการ shader ซึ่งได้นำมาใช้ห่อ กับเส้นที่สร้างขึ้นโดยใช้สีเป็นสีฟ้าอ่อน ดังรูปสายฟ้าที่เห็นข้างต้น

## 5. ล่ม



ใน project นี้ ผู้ใช้สามารถกำหนดทิศทางของลมและความแรงของลมได้ โดยกำหนดค่าที่ sidebar ซึ่งเมื่อลมไปพัดไปทางใด จะทำให้เส้นของฝนนั้นถูกหมุนไปตามทิศของลมด้วย และถ้าตั้งให้ลมแรง เสียงลมจะดังขึ้นตามความแรงของลมด้วย

## องค์ประกอบเสริม

### 1. การเช็คชน

```
function intersect(first, second) {  
    var firstx = first.object.position.x, firstly = first.object.position.y, firstz = first.object.position.z;  
    var secondx = second.object.position.x, secondy = second.object.position.y, secondz = second.object.position.z;  
  
    var dx = (firstx - secondx);  
    var dy = (firsty - secondy);  
    var dz = (firstz - secondz);  
  
    var dist = dx*dx + dy*dy + dz*dz;  
    var radius2 = (first.radius + second.radius)*(first.radius + second.radius);  
  
    if(dist<=radius2){  
        return true;  
    }  
  
    return false;  
}
```

ในการเช็คการชนกันของน้ำ ถึงแม้ว่าตัวเม็ดฝนนั้นจะมีรูปทรง 3 แบบ แต่ลักษณะโดยรวมค่อนข้างจะเป็นทรงกลม ดังนั้นการเช็คชนจึงเทียบระหว่างระยะห่าง กับ ผลรวมของรัศมี เพื่อนั้นเพื่อเป็นการลดภาระการคำนวณในการเช็คชนแต่ละครั้ง ซึ่งถ้า  $\sqrt{x^2 + y^2 + z^2} \leq R1 + R2$  เป็นจริง จะถือว่าเม็ดฝนนั้นชนกัน

### 2. การควบคุมจากคีย์บอร์ด

```
document.addEventListener('keydown', function(event) {  
    if(event.keyCode == 37)  
        manager.camera.rotateY(0.02);  
    if(event.keyCode == 39)  
        manager.camera.rotateY(-0.02);  
    if(event.keyCode == 38) {  
        manager.camera.translateZ(-0.1);  
        if(Math.abs(manager.camera.position.x) >= Manager.getWidth()/3 || (Math.abs(manager.camera.position.z) >= Manager.getWidth()/3))  
            manager.camera.translateZ(0.1);  
    }  
    if(event.keyCode == 40){  
        manager.camera.translateZ(0.1);  
        if(Math.abs(manager.camera.position.x) >= Manager.getWidth()/3 || (Math.abs(manager.camera.position.z) >= Manager.getWidth()/3))  
            manager.camera.translateZ(-0.1);  
    }  
    if(event.keyCode == 87 && manager.mirror.position.y<=2){  
        manager.mirror.position.y+=0.03;  
    }  
    if(event.keyCode == 83 && manager.mirror.position.y>=-0.03){  
        manager.mirror.position.y-=0.03;  
    }  
    if(event.keyCode == 81){  
        manager.dome = !manager.dome;  
    }  
});  
});
```

เป็นฟังก์ชันที่คอยรับการตั้งค่าจาก keyboard

- ลูกศรซ้าย / ลูกศรขวา → ใช้ในการหมุนกล้องซ้ายขวา ซึ่งจะเป็นการหมุนแกนของกล้องโดยตรง

- ลูกศรขึ้น / ลูกศรลง → ใช้ในการเลื่อนไปข้างหน้าหรือถอยหลัง โดยจะไม่สามารถเดินออกจากพื้นที่ที่กำหนดไว้ได้ เนื่องจากเป็นแค่ภาพ 360 องศา ไม่ใช่ภาพ 3D จริงๆ
- W / S → ใช้ในการเลื่อนกระจากขึ้น หรือเลื่อนกระจากลง
- Q → ใช้ในการ toggle เดิมเข้า / ออก

### 3. การควบคุมจาก Sidebar

```

$( "#wind" ).slider({
  orientation: "horizontal",
  range: "min",
  min: -127,
  max: 127,
  value: 0,
  slide: updateparameter,
  change: updateparameter
});

$( "#rain" ).slider({
  orientation: "horizontal",
  range: "min",
  min: 0,
  max: 20,
  value: 0,
  slide: updateparameter,
  change: updateparameter
});

```

```

$( "#lightning" ).slider({
  orientation: "horizontal",
  range: "min",
  min: 0,
  max: 100,
  value: 0,
  slide: updateparameter,
  change: updateparameter
});

$( "#sun" ).slider({
  orientation: "horizontal",
  range: "min",
  min: 0,
  max: 100,
  value: 0,
  slide: updateparameter,
  change: updateparameter
});

```

เป็นฟังก์ชันในการกำหนดค่าต่ำสุด และสูงสุดของ slider ทั้ง ลม สายพาน และการระเหย โดยถ้ามีการเลื่อน slider จะถูกส่งให้เข้า function updateparameter ซึ่งจะทำการอัพเดตค่า config ต่างๆ ในตัว project

### 4. การเปลี่ยนขนาดหน้าจอ

```

function resize(){
  manager.renderer.setSize( window.innerWidth * 80.0 / 100, window.innerHeight );
  manager.renderer.setPixelRatio( window.devicePixelRatio );
  manager.camera.aspect = window.innerWidth * 80.0 / 100.0 / window.innerHeight;
  manager.camera.updateProjectionMatrix();
}

```

เนื่องจากต้องการให้สเกลของหน้าจอ กับ ตัวงาน เท่าเดิมเสมอ จึงทำการสร้าง function ที่จะคอยกำหนดค่า ความกว้างและความสูงของกล้อง และตัว renderer ให้เท่ากับขนาดหน้าจอเสมอ

## ปัญหาที่พบ

- Memory leak
  - ปัญหา object ที่สร้างขึ้นใน threejs เมื่อเอาออกจาก scene จะยังไม่ได้ถูกลบพื้นที่ที่จะคงไว้ออกจาก memory heap
  - วิธีแก้ ทำการใช้คำสั่ง dispose ให้กับ material และ geometry ของวัตถุหลังที่เอาออกจาก scene จะทำให้ได้ memory heap คืนทันที
- z-buffer
  - ปัญหา เมื่อมี transparent object สองอันซ้อนกัน แต่จะมองไม่เห็น transparent object ที่อยู่ด้านหลัง
  - วิธีแก้
    - กำหนดค่า depthwrite เป็น false ให้กับ material
    - ทำการ render ตามลำดับที่ต้องการ
- ผนไม่透
  - ปัญหา เนื่องจากช่วงนี้ผนไม่透 จึงไม่สามารถดูจากของจริงได้โดยตรง
  - วิธีแก้
    - ทำการจำลองกับน้ำธรรมชาติ แล้วดูว่าพฤติกรรมของน้ำเมื่อไฟลแทน
    - ศึกษาจาก video ที่ได้มีการถ่ายไว้

## แนวทางพัฒนาต่อหากมีเวลาเพิ่ม

ต้องหาสมการที่แท้จริงว่า แท้จริงแล้วหยดน้ำตอนเคลื่อนตัว หากคำนึงถึง องค์ประกอบของแรงเสียดทาน น้ำหนัก และแรงยึดเหนี่ยว แรงไหนจะชนะแรงไหนต่อเมื่ออะไร ให้สามารถ generate ทุกเม็ดได้อย่างถูกต้อง และ สายฟ้าหากทำให้ตรวจสอบพื้นที่ ที่น่าจะเกิดฟ้าผ่าได้ ให้ฟ้าผ่าเฉพาะจุดนั้น แล้วการแตก branch ของฟ้าผ่าที่มี factor ของความต้านทานอากาศเข้ามาเกี่ยวข้อง ซึ่งในตอนนี้การเกิดสายฟ้านั้นยังไม่เหมือนกับการเกิดสายฟ้าจริงในธรรมชาตินัก

## หน้าที่ของสมาชิก

- นาย กิตติพงษ์ ศิริเรืองสกุล 5731008221  
ศึกษาและทำเกี่ยวกับ เม็ดฝน, refraction, cubemap และแสง
- นาย ปรานิน โลกโบว์ 5731071721  
ศึกษาและทำเกี่ยวกับ ลม, การหมุน, การเดิน และเสียงประกอบ
- นาย อภิรุจ ชูมวัฒนะ 5731113421  
ศึกษาและทำเกี่ยวกับ พื้นผ่า, ภาพ equirectangular และโอดม

## Video

- <https://youtu.be/yEa4seGmOIA>

## References

- "Stars.chromeexperiments.com". N.p., 2016. Web. 14 Dec. 2016.
- "An Interactive WebGL Sky + Sun Shader Runnable In Browsers.". *Plus.google.com*. N.p., 2016. Web. 14 Dec. 2016.
- "Cite A Website - Cite This For Me". *Threejs.org*. N.p., 2016. Web. 14 Dec. 2016.
- "Diffusion Limited Aggregation (2D DLA) 1". *YouTube*. N.p., 2016. Web. 14 Dec. 2016.
- "Experiments Behind "It Came Upon" | Kepo-Ing Zz85". *Lab4games.net*. N.p., 2016. Web. 14 Dec. 2016.
- "Graphics Noob On Twitter". *Twitter*. N.p., 2016. Web. 14 Dec. 2016.
- "Hoffman, Michael. "How To Generate Shockingly Good 2D Lightning Effects". *Game Development Envato Tuts+*. N.p., 2016. Web. 14 Dec. 2016.
- "Hoffman, Naty and Arcot J Preetham. *Rendering Outdoor Light Scattering In Real Time*". 1st ed. Markham, Ontario: ATI Technologies Inc., 2016. Web. 14 Dec. 2016.
- "Hoffman, Naty and Arcot J Preetham. *Rendering Outdoor Light Scattering In Real Time*". 1st ed. 2016. Web. 14 Dec. 2016.
- "How Far Away Is Lightning | Calculate Lightning Distance". *Live Science*. N.p., 2016. Web. 14 Dec. 2016.
- "Lightning Graphic 4". *Sb-photography-stock.deviantart.com*. N.p., 2016. Web. 14 Dec. 2016.

"Science, Live. "How Far Away Is Lightning | Calculate Lightning Distance". *Live Science*. N.p., 2016. Web. 14 Dec. 2016.

sun, Using. "Using THREE.Js To Make A Shining Sun". *Stackoverflow.com*. N.p., 2016. Web. 14 Dec. 2016.

"Three.Js | Kepo-Ing Zz85 | Page 2". *Lab4games.net*. N.p., 2016. Web. 14 Dec. 2016.

"Three.Js Webgl - Clouds - Godrays". *Blurspline.com*. N.p., 2016. Web. 14 Dec. 2016.

"Three.Js Webgl - Equirectangular Panorama". *Threejs.org*. N.p., 2016. Web. 14 Dec. 2016.

"how-to-generate-shockingly-good-2d-lightning-effects".

[gamedevelopment.tutsplus.com](http://gamedevelopment.tutsplus.com). Michael Hoffman, 2016. Web. 14 Dec. 2016.

"UDK - Hoffman & Preetham Sky Scattering". *YouTube*. N.p., 2016. Web. 14 Dec. 2016.

"threex.geometricglow". [github.com/jeromeetienne](https://github.com/jeromeetienne/threex.geometricglow), 2016. Web. 14 Dec. 2016.

"Rain & Water Effect Experiments". [tympanus.net](http://tympanus.net/Development/RainWaterEffect/). Lucas Bebber, 2016. Web. 14 Dec. 2016.

"UDK, Hoffman. "Hoffman & Preetham Sky Scattering In UDK".

[Dmg3d.blogspot.com.es](http://dmg3d.blogspot.com.es/2016/07/hoffman-preetham-sky-scattering-in.html). N.p., 2016. Web. 14 Dec. 2016.

"Use Three.Js Positionalaudio To Make A Cone Of Sound". *Stackoverflow.com*. N.p., 2016. Web. 14 Dec. 2016.

