

Particle System

Danai Jedsadathitikul

Jarut Busarathid

DcG.IT@PBRU



**Particle System เป็นเครื่องมือสำคัญใน
Unity ที่ช่วยสร้างเอฟเฟกต์ภาพ เช่น ควัน,
ไฟ, ฝุ่น, หิมะ, ฝน, และประกายไฟ**



โดยการจัดการอนุภาค (Particle) หลาย ๆ
อนุภาคพร้อมกัน ซึ่งสามารถปรับแต่งการ
แสดงผล, การเคลื่อนไหว, และพฤติกรรมได้
อย่างละเอียด



เกมแอ็กชัน: เอฟเฟกต์การระเบิดเมื่อศัตรูถูก โจมตี



เกมแฟนตาซี: แสงประกายขณะใช้เวทมนตร์

www.gamedevpbru.com



เกมแข่งรถ: ฟุนจากล้อรถที่เคลื่อนไหว

www.gamedevpbru.com



ส่วนประกอบสำคัญของ Particle System



- ควบคุมพฤติกรรมหลัก เช่น อายุอนุภาค, ความเร็วเริ่มต้น, ขนาดเริ่มต้น และสีเริ่มต้น



- กำหนดอัตราการปล่อยอนุภาค เช่น จำนวนอนุภาคต่อวินาที หรือปล่อยแบบ Burst เป็นต้น



- กำหนดรูปทรงพื้นที่ที่อนุภาคถูกปล่อย เช่น จุด (Point), วงกลม (Circle), ทรงกลม (Sphere), หรือกรวย (Cone)



4

Velocity over Lifetime Module

- กำหนดความเร็วของอนุภาคที่เปลี่ยนแปลงตลอดอายุการใช้งาน



5

Color over Lifetime Module

- กำหนดการเปลี่ยนสีของอนุภาคตามอายุการใช้งาน



6

Size over Lifetime Module

- กำหนดการเปลี่ยนแปลงขนาดของอนุภาคตลอดอายุการใช้งาน



- ควบคุมการแสดงผลของอนุภาค เช่น การใช้ Material หรือ Shader



ขั้นตอนการสร้าง Particle System



- คลื่นขวาทึ่
Hierarchy > Effects > Particle System
- Unity จะเพิ่ม GameObject พร้อม Component Particle System ลงในฉาก



- **Duration:** ระยะเวลาของการเล่น Particle System (วินาที)
- **Looping:** เปิดใช้งานหากต้องการให้อนุภาคเล่นซ้ำอย่างต่อเนื่อง
- **Start Lifetime:** อายุของอนุภาคก่อนที่จะหายไป
- **Start Speed:** ความเร็วเริ่มต้นของอนุภาค
- **Start Size:** ขนาดเริ่มต้นของอนุภาค



3 เพิ่ม Material ให้กับ Particle System

- สร้าง Material ใหม่ โดยคลิกขวาที่ Assets > Create > Material
- เลือก Shader เป็น Particles/Standard Unlit และเพิ่ม Texture ที่ต้องการ



- Emission Module
เพิ่มอัตราการปล่อยอนุภาค เช่น 10 อนุภาคต่อวินาที
- Shape Module
เปลี่ยนรูปทรงการปล่อยอนุภาค เช่น Cone สำหรับ
เอฟเฟกต์เปลวไฟ



```
using UnityEngine;
```

```
public class SmokeEffect : MonoBehaviour  
{
```

```
    public ParticleSystem particleSystem;
```

```
    void Start() {
```

```
        var main = particleSystem.main;
```

```
        main.startLifetime = 5f; // อายุของอนุภาค 5 วินาที
```

```
        main.startSpeed = 1f; // ความเร็วเริ่มต้น 1 หน่วย
```

```
        main.startSize = 0.5f; // ขนาดเริ่มต้น 0.5
```

```
        main.startColor = new Color(0.5f, 0.5f, 0.5f, 0.5f); // สีเทาโปร่งใส
```

```
        var emission = particleSystem.emission;
```

```
        emission.rateOverTime = 20f; // ปล่องอนุภาค 20 ต่อวินาที
```

```
        var shape = particleSystem.shape;
```

```
        shape.shapeType = ParticleSystemShapeType.Cone; // ใช้ทรงกรวยสำหรับการปล่องอนุภาค
```

```
    }
```

```
}
```



particleSystem.main

ใช้ปรับค่าหลัก เช่น อายุ, ความเร็ว และขนาดของอนุภาค



particleSystem.emission

ใช้ปรับอัตราการปล่อยอนุภาค



particleSystem.shape

ใช้กำหนดรูปทรงการปล่อยอนุภาค



ตัวอย่างการใช้งาน Particle System



- ใช้ Shape Module แบบ Cone (กรวย) เพื่อให้อนุภาคพุ่งขึ้น
- เพิ่ม Color over Lifetime ให้อนุภาคเปลี่ยนสีจากสีแดงเป็นสีเหลือง
- ใช้ Size over Lifetime เพื่อให้อนุภาคค่อย ๆ หดตัว



- เพิ่ม Particle System ในฉาก
- Shape Module: เลือก Sphere เพื่อให้ประกายไฟกระจายทุกทิศทาง
- Emission Module: ตั้ง Burst ให้ปล่อยอนุภาค 30-50 เมื่อเริ่มต้น
- Renderer Module: ใช้ Material แบบ Glow หรือ Emissive เพื่อเพิ่มความสว่าง



- **Main Module**

- **Start Lifetime: 1**
- **Start Speed: 5**
- **Start Size: 0.1**
- **Start Color: สีแดง (RGBA: 1, 0.5, 0, 1)**



ขั้นตอนการสร้างเอฟเฟกต์ไฟ

- สร้าง Particle System
 - คลิกขวาที่ Hierarchy > Effects > Particle System
 - ตั้งชื่อ GameObject เช่น FireEffect



- ตั้งค่าหลักใน Particle System โดยปรับค่าใน Inspector ของ Particle System
 - Duration: 3
 - Looping: เปิดใช้งาน (เพื่อให้เปลวไฟเล่นซ้ำ)
 - Start Lifetime: 1 ถึง 3 (สุ่มอายุของอนุภาค)
 - Start Speed: 1 ถึง 2
 - Start Size: 0.5 ถึง 1
 - Start Color: เลือก Gradient สีที่ไล่จากส้มไปแดง หรือเหลืองไปส้ม
 - Gravity Modifier: -0.1 (อนุภาคลอยขึ้น)



- **ปรับแต่ง Modules Emission**
 - Rate Over Time: 30 (จำนวนอนุภาคต่อวินาที)
- **ปรับแต่ง Modules Shape**
 - เปลี่ยน Shape เป็น Cone
 - ปรับ Angle เป็น 25 (เพื่อให้อนุภาคกระจาย)
 - ปรับ Radius เป็น 0.2
- **ปรับแต่ง Modules Color over Lifetime**
 - เปิดใช้งาน และตั้งค่า Gradient เป็นสีไล่จากเหลือง (เริ่มต้น) ไปส้ม และจางหาย (ใส)



- ปรับแต่ง Modules Size over Lifetime
 - เปิดใช้งาน และตั้งค่าให้ขนาดของอนุภาคลดลงเรื่อย ๆ
- ปรับแต่ง Modules Renderer เพิ่ม Material แบบโปร่งใส
 - สร้าง Material ใหม่ (Assets > Create > Material)
 - เลือก Shader เป็น Particles/Standard Unlit
 - เพิ่ม Texture เช่น Texture รูป "Smoke" หรือ "Fire"



การตั้งค่า

- สร้าง Fire Effect
 - สร้าง Particle System ตามที่ได้อธิบายไว้ด้านบน
 - ปรับแต่งให้เหมาะสมกับลักษณะของไฟ



- **เพิ่ม Script**

- สร้าง C# Script ชื่อ FireEffectController.cs และเพิ่มลงใน GameObject หลัก

- **เชื่อมโยง Particle System**

- ลาก Particle System ไปยังตัวแปร fireEffect ใน Inspector

- **ทดสอบ**

- กดปุ่ม Play และกดแป้น E เพื่อเปิด/ปิดเอฟเฟกต์ไฟ



```
using UnityEngine;
```

```
public class FireEffectController : MonoBehaviour
{
    public ParticleSystem fireEffect; // อ้างอิง Particle System
    void Update()
    {
        // ตรวจสอบว่ากดแป้น E เพื่อเปิด/ปิดไฟ
        if (Input.GetKeyDown(KeyCode.E)) {
            if (fireEffect.isPlaying) {
                fireEffect.Stop();
                Debug.Log("Fire Effect หยุด");
            } else {
                fireEffect.Play();
                Debug.Log("Fire Effect เล่น");
            }
        }
    }
}
```



เพิ่มควัน

สร้าง Particle System อีกชุดสำหรับควัน

1. ใช้ Shape แบบ Cone
2. ลด Start Speed และ Start Size
3. ตั้ง Color over Lifetime ให้เป็นสีเทา
4. เปิดใช้งาน Noise Module เพื่อให้ควันเคลื่อนไหวแบบสุ่ม



เพิ่มเสียงไฟ

```
public AudioSource fireSound;
void Update() {
    if (Input.GetKeyDown(KeyCode.E)) {
        if (fireEffect.isPlaying) {
            fireEffect.Stop();
            fireSound.Stop(); // หยุดเสียง
        } else {
            fireEffect.Play();
            fireSound.Play(); // เล่นเสียง
        }
    }
}
```



เพิ่มแสง (Light)

เพิ่ม Light เพื่อทำให้ไฟดูสมจริง

1. สร้าง GameObject Light และวางไว้ที่ตำแหน่งเดียวกับเอฟเฟกต์ไฟ
2. ตั้งค่า Light เป็น Point Light
3. ใช้ Script เพื่อเปลี่ยนความเข้มของแสงให้กระพริบแบบสุ่ม:



```
public Light fireLight;
```

```
void Update()
```

```
{  
    // ความสว่างเปลี่ยนแปลงแบบสุ่ม  
    fireLight.intensity = Random.Range(1f, 2f);  
}
```



- ใช้ Shape Module แบบ Box เพื่อให้อนุภาคตกลงมาจากด้านบน
- ลด Start Speed เพื่อให้อนุภาคเคลื่อนที่ช้า
- เพิ่ม Start Lifetime เพื่อให้อนุภาคอยู่บนจอได้นานขึ้น



การสร้างเอฟเฟกต์หิมะใน Unity สามารถทำได้โดยใช้ Particle System ปรับแต่งให้อนุภาคเคลื่อนที่จากด้านบนลงล่างอย่างช้า ๆ พร้อมเพิ่มคุณสมบัติเสริมเพื่อให้สมจริง เช่น ความสุมในทิศทางการตก และความหนาแน่นของหิมะ



ขั้นตอนการสร้างเอฟเฟกต์พิเศษ

- สร้าง Particle System
 - คลิกขวาที่ Hierarchy > Effects > Particle System
 - ตั้งชื่อ GameObject เช่น SnowEffect



- ตั้งค่าหลักใน Particle System โดยปรับค่าใน Inspector ของ Particle System
 - Duration: 5 (ระยะเวลาที่อนุภาคถูกสร้าง)
 - Looping: เปิดใช้งาน (เพื่อให้หิมะเล่นซ้ำเรื่อย ๆ)
 - Start Lifetime: 5 ถึง 10 (สุ่มอายุของอนุภาค)
 - Start Speed: 0.5 ถึง 1 (ความเร็วในการตก)
 - Start Size: 0.1 ถึง 0.3 (ขนาดอนุภาคหิมะ)
 - Start Color: สีขาวหรือสีเทาอ่อน (RGB: 255, 255, 255 หรือ 200, 200, 200)



- **ปรับแต่ง Modules Emission**
 - Rate Over Time: 100 (จำนวนอนุภาคที่ปล่อยต่อวินาที)
- **ปรับแต่ง Modules Shape**
 - เลือก Shape เป็น Box เพื่อจำลองหิมะตกจากพื้นที่ขนาดใหญ่
 - ปรับขนาดของ Box
 - X: 10 (ความกว้าง)
 - Y: 0 (ความสูงของต้นทางการปล่อย)
 - Z: 10 (ความลึก)
 - ย้ายตำแหน่ง Box ให้อยู่ด้านบนของพื้นที่เล่น (เช่นที่แกน $Y = 10$)



- ปรับแต่ง Modules Velocity over Lifetime
 - เปิดใช้งาน และปรับแกน Y เป็นค่าลบ (เช่น -1) เพื่อให้อนุภาคตกลงด้านล่าง
 - เพิ่ม Noise หรือ Randomness เพื่อทำให้การตกไม่เป็นเส้นตรง
- ปรับแต่ง Modules Renderer
 - เพิ่ม Material แบบโปร่งใส
 - สร้าง Material ใหม่ (Assets > Create > Material)
 - เลือก Shader เป็น Particles/Standard Unlit
 - เพิ่ม Texture รูป "Snowflake" หรือจุดวงกลมโปร่งใส



```
using UnityEngine;
```

```
public class SnowEffectController : MonoBehaviour
```

```
{  
    public ParticleSystem snowEffect; // อ้างอิง Particle System  
    void Update() {  
        // เปิด/ปิดหิมะเมื่อกดแป้น E  
        if (Input.GetKeyDown(KeyCode.E)) {  
            if (snowEffect.isPlaying) {  
                snowEffect.Stop();  
                Debug.Log("หยุดเอฟเฟกต์หิมะ");  
            } else {  
                snowEffect.Play();  
                Debug.Log("เริ่มเอฟเฟกต์หิมะ");  
            }  
        }  
    }  
}
```



เพิ่มความสมจริงด้วย Noise

1. เปิด Noise Module ใน Particle System

2. ตั้งค่า

1. Strength: 0.2 ถึง 0.5 (ความเบี่ยงเบนของทิศทางการตก)

2. Frequency: 0.5 (ความถี่ของการเปลี่ยนทิศทาง)

3. Scroll Speed: 0.2 (ความเร็วของการสุ่ม)



เพิ่มการละลายของหิมะเมื่อชนพื้น

```
using UnityEngine;
```

```
public class SnowCollision : MonoBehaviour  
{  
    void OnParticleCollision(GameObject other)  
    {  
        // ทำสิ่งที่ต้องการ เช่น การลบอนุภาค  
        Debug.Log("หิมะชนกับพื้น");  
    }  
}
```



เพิ่มลมเพื่อเปลี่ยนทิศทางของหิมะ

1. เปิด Force over Lifetime Module
 1. ตั้งค่าแกน X หรือ Z เป็นค่าบวกหรือลบ (เช่น 0.5) เพื่อให้หิมะเบี่ยงไปในทิศทางของลม
2. เพิ่ม Noise เพื่อเพิ่มความแรงของลม



เพิ่มเสียงประกอบ

```
public AudioSource snowSound;
```

```
void Start()
```

```
{
```

```
    snowSound.Play(); // เล่นเสียงเมื่อเริ่มต้น
```

```
}
```



เพิ่มเสียงประกอบ

```
public AudioSource snowSound;
```

```
void Start()
```

```
{
```

```
    snowSound.Play(); // เล่นเสียงเมื่อเริ่มต้น
```

```
}
```



- ใช้ Shape Module แบบ Cone เพื่อให้ควันพุ่งขึ้น
- เพิ่ม Noise Module เพื่อให้ควันเคลื่อนไหวแบบสุ่ม



- เพิ่ม Particle System ในฉาก
- Shape Module: เลือก Cone เพื่อให้ควันพุ่งขึ้น
- Emission Module: ตั้ง Rate Over Time เป็น 10-20
- Color Over Lifetime Module: เปลี่ยนสีจากเทาอ่อนเป็นโปร่งใส
- Renderer Module: ใช้ Material แบบโปร่งใสสำหรับควัน



- **Main Module**

- **Start Lifetime: 5**
- **Start Speed: 1**
- **Start Size: 0.5**
- **Start Color: เทาอ่อน (RGBA: 0.8, 0.8, 0.8, 1)**



เอฟเฟกต์ควันสามารถสร้างได้ด้วย Particle System
โดยปรับแต่งให้อนุภาคเคลื่อนที่ขึ้นไปแบบลุ่ม เพิ่มการ
เบี่ยงเบนด้วย Noise และใช้ Color Over Lifetime
เพื่อเปลี่ยนสีควันให้ค่อย ๆ จางหาย



ขั้นตอนการสร้างเอฟเฟกต์ควัน

- สร้าง Particle System
 - คลิกขวาที่ Hierarchy > Effects > Particle System
 - ตั้งชื่อ GameObject เช่น SmokeEffect



- ตั้งค่าหลักใน Particle System โดยปรับค่าใน Inspector ของ Particle System
 - Duration: 5 (ระยะเวลาการเล่นของ Particle System)
 - Looping: เปิดใช้งาน (เพื่อให้ควันเล่นซ้ำเรื่อย ๆ)
 - Start Lifetime: 3 ถึง 5 (สุ่มอายุของอนุภาค)
 - Start Speed: 1 ถึง 2 (ความเร็วเริ่มต้นของอนุภาค)
 - Start Size: 0.5 ถึง 1 (ขนาดเริ่มต้นของอนุภาค)
 - Start Color: สีเทาอ่อน (RGB: 200, 200, 200) หรือปรับเป็น Gradient สีที่ไล่จากสีเข้มไปจาง



- ปรับแต่ง Modules Emission
 - Rate Over Time: 20 ถึง 50 (จำนวนอนุภาคที่ปล่อยต่อวินาที)
- ปรับแต่ง Modules Shape
 - เลือก Shape เป็น Cone
 - ปรับมุมของกรวย (Angle) เป็น 10 เพื่อให้ควันพุ่งขึ้นเล็กน้อย
 - ปรับ Radius เป็น 0.2 ถึง 0.5 เพื่อจำกัดพื้นที่ปล่อยควัน



- **ปรับแต่ง Modules Size Over Lifetime**
 - เปิดใช้งาน และตั้งกราฟให้อนุภาคขยายใหญ่ขึ้นเล็กน้อยเมื่อเวลาผ่านไป (เช่นจาก 0.5 ไป 1)
- **ปรับแต่ง Modules Noise**
 - เปิด Noise Module เพื่อเพิ่มความเคลื่อนไหวแบบสุ่ม
 - Strength: 0.5 ถึง 1 (ความเบี่ยงเบนของอนุภาค)
 - Frequency: 0.2 ถึง 0.5 (ความถี่ของการเปลี่ยนทิศทาง)
 - Scroll Speed: 0.1 ถึง 0.3



- **ปรับแต่ง Modules Color Over Lifetime**
 - เปิดใช้งาน และตั้งค่า Gradient
 - เริ่มจากสีเทาเข้ม (RGBA: 100, 100, 100, 255)
 - ค่อย ๆ ไล่ไปสีเทาอ่อน (RGBA: 200, 200, 200, 100)
 - จางหายเป็นโปร่งใส (RGBA: 200, 200, 200, 0)
- **ปรับแต่ง Modules Renderer**
 - เพิ่ม Material แบบโปร่งใส
 - สร้าง Material ใหม่ (Assets > Create > Material)
 - เลือก Shader เป็น Particles/Standard Unlit
 - เพิ่ม Texture รูป "Smoke" หรือจุดเบลอ (blurred dot)



```
using UnityEngine;

public class SmokeEffectController : MonoBehaviour
{
    public ParticleSystem smokeEffect; // อ้างอิง Particle System

    void Update()
    {
        // เปิด/ปิดควันเมื่อกดแป้น E
        if (Input.GetKeyDown(KeyCode.E)) {
            if (smokeEffect.isPlaying) {
                smokeEffect.Stop();
                Debug.Log("หยุดเอฟเฟกต์ควัน");
            } else {
                smokeEffect.Play();
                Debug.Log("เริ่มเอฟเฟกต์ควัน");
            }
        }
    }
}
```



เพิ่มการลอยขึ้น

ใช้ Velocity over Lifetime Module โดยตั้งค่าแกน Y เป็น 1 ถึง 2 เพื่อให้ควันลอยขึ้น



เพิ่มแรงลม

ใช้ Force over Lifetime Module โดยตั้งค่าแกน X หรือ Z เป็น 0.2 ถึง 0.5 เพื่อให้ควันเบี่ยงไปตามลม



เพิ่มแสงเพื่อความสมจริง

เพิ่ม Light ให้กับ Particle System โดยใช้ Light แบบ Point Light หรือ Spot Light เพื่อให้ควันดูมีมิติ



เอฟเฟกต์ระเบิดสามารถสร้างได้โดยการใช้ Particle System เพื่อแสดงอนุภาคของไฟ, ประกาย, ควัน และ เศษวัตถุที่กระจัดกระจายออกไปพร้อมกัน นอกจากนี้ สามารถเพิ่มแสง (Light) และเสียง (Sound) เพื่อเพิ่มความสมจริง



ขั้นตอนการสร้างเอฟเฟกต์ควัน

- สร้าง Particle System
 - คลิกขวาที่ Hierarchy > Effects > Particle System
 - ตั้งชื่อ GameObject เช่น ExplosionEffect



- ตั้งค่าหลักใน Particle System โดยปรับค่าใน Inspector ของ Particle System

- Duration: 1 (ระยะเวลาของการระเบิด)
- Looping: ปิดใช้งาน (ระเบิดควรเกิดขึ้นครั้งเดียว)
- Start Lifetime: 0.5 ถึง 1 (สุ่มอายุของอนุภาค)
- Start Speed: 5 ถึง 10 (ความเร็วเริ่มต้นของอนุภาค)
- Start Size: 0.3 ถึง 1 (สุ่มขนาดของอนุภาค)
- Start Color: ใช้ Gradient ไล่สีจากส้ม-แดง-ดำ



- ปรับแต่ง Modules Emission
 - Rate Over Time: 0 (ตั้งค่าเป็น 0 เพราะจะใช้ Burst)
 - Bursts: เพิ่ม Burst ที่ Time = 0, Count = 50 ถึง 100
- ปรับแต่ง Modules Shape
 - เลือก Shape เป็น Sphere เพื่อให้อนุภาคกระจายทุกทิศทาง
 - ปรับ Radius เป็น 0.5 ถึง 1 เพื่อควบคุมพื้นที่ระเบิด
- ปรับแต่ง Modules Size Over Lifetime
 - เปิดใช้งาน และตั้งค่ากราฟให้อนุภาคเริ่มใหญ่แล้วค่อย ๆ หดตัว



- **ปรับแต่ง Modules Color Over Lifetime**
 - เปิดใช้งาน และตั้งค่า Gradient
 - เริ่มจากสีส้ม (RGBA: 255, 100, 0, 255)
 - ไล่สีเป็นแดง และจางหายเป็นโปร่งใส (RGBA: 50, 0, 0, 0)
- **ปรับแต่ง Modules Renderer**
 - เพิ่ม Material แบบโปร่งใส
 - สร้าง Material ใหม่ (Assets > Create > Material)
 - เลือก Shader เป็น Particles/Standard Unlit
 - เพิ่ม Texture รูป "Smoke" หรือจุดเบลอ (blurred dot)



- **เพิ่มแสง (Light)**

- คลิกขวาที่ Hierarchy > Light > Point Light

- ตั้งค่าหลัก

- Intensity: 5 ถึง 10

- Range: 5 ถึง 10

- เพิ่ม Script ควบคุมการลดความสว่างของแสงหลังระเบิด:



```
using UnityEngine;
```

```
public class LightFlicker : MonoBehaviour
```

```
{  
    public Light explosionLight;  
    private float decayRate = 10f;  
  
    void Update()  
    {  
        if (explosionLight.intensity > 0)  
        {  
            explosionLight.intensity -= decayRate * Time.deltaTime;  
        }  
    }  
}
```



- **เพิ่มเสียง (Sound)**

- **เพิ่ม Audio Source ลงใน GameObject ExplosionEffect**

- **ตั้งค่า Audio Source**

- **เพิ่มไฟล์เสียงระเบิด (.wav หรือ .mp3) ในช่อง Audio Clip**

- **ปิด Play on Awake และเรียกใช้ผ่าน Script**



```
using UnityEngine;
```

```
public class ExplosionController : MonoBehaviour
{
    public ParticleSystem explosionEffect; // อ้างอิง Particle System
    public AudioSource explosionSound; // อ้างอิงเสียงระเบิด
    public Light explosionLight; // อ้างอิงแสง

    void Start()
    { // ซ่อนเอฟเฟกต์และแสงเริ่มต้น
        explosionEffect.Stop();
        explosionLight.intensity = 0;
    }

    void Update()
    { // กดแป้น E เพื่อเรียกใช้งานเอฟเฟกต์ระเบิด
        if (Input.GetKeyDown(KeyCode.E))
        {
            TriggerExplosion();
        }
    }
}
```



```
void TriggerExplosion()
{
    // เล่นเอฟเฟกต์
    explosionEffect.Play();

    // เล่นเสียง
    explosionSound.Play();

    // ตั้งค่าแสง
    explosionLight.intensity = 5;
}
}
```



เพิ่มเศษวัตถุที่กระเด็น

1. สร้าง Prefab เศษวัตถุ
2. ใช้ Particle System แยกต่างหากสำหรับการกระจายเศษ
 1. ใช้ Shape แบบ Sphere
 2. เพิ่ม Gravity Modifier เพื่อให้เศษตกลงสู่พื้น



เพิ่มการสั่นหน้าจอ

```
using UnityEngine;
public class CameraShake : MonoBehaviour {
    public IEnumerator Shake(float duration, float magnitude) {
        Vector3 originalPos = transform.localPosition;
        float elapsed = 0.0f;
        while (elapsed < duration) {
            float x = Random.Range(-1f, 1f) * magnitude;
            float y = Random.Range(-1f, 1f) * magnitude;
            transform.localPosition = new Vector3(x, y, originalPos.z);
            elapsed += Time.deltaTime;
            yield return null;
        }
        transform.localPosition = originalPos;
    }
}
```



เพิ่มควันหลังการระเบิด

1. เพิ่ม Particle System สำหรับควัน
 1. ลด Start Speed และเพิ่ม Lifetime
 2. ใช้ Color Over Lifetime ให้ควันจางลง



การเพิ่ม Particle System ด้วยโค้ด



สามารถใช้โค้ดเพื่อเปิด/ปิด Particle System หรือปรับแต่งค่าขณะรันไทม์



```
using UnityEngine;
```

```
public class CreateParticleSystem : MonoBehaviour
{
    void Start()
    {
        // สร้าง GameObject ใหม่สำหรับ Particle System
        GameObject particleObject = new GameObject("Particle System");
        var particleSystem = particleObject.AddComponent<ParticleSystem>();
        // ตั้งค่าหลัก
        var main = particleSystem.main;
        main.startLifetime = 3f;
        main.startSpeed = 2f;
        main.startSize = 0.5f;
        main.startColor = Color.white;

        // ตั้งค่าการปล่อย
        var emission = particleSystem.emission;
        emission.rateOverTime = 10f;
    }
}
```



```
// ตั้งค่ารูปทรง  
var shape = particleSystem.shape;  
shape.shapeType = ParticleSystemShapeType.Cone;  
  
particleObject.transform.position = new Vector3(0, 0, 0); // ตำแหน่งเริ่มต้น  
}  
}
```



ข้อดีของ Particle System

- ความยืดหยุ่น
สามารถสร้างเอฟเฟกต์ที่ซับซ้อนได้ด้วยการปรับแต่ง Modules
- ประสิทธิภาพ
รองรับการแสดงผลอนุภาคจำนวนมากในฉากโดยไม่ส่งผลกระทบต่อประสิทธิภาพของเกมมากนัก
- การปรับแต่งผ่านโค้ด
สามารถเปลี่ยนค่าต่าง ๆ แบบ Runtime ได้



ข้อควรระวัง

- การใช้ทรัพยากร
ควรจำกัดจำนวนอนุภาค (Max Particles) เพื่อป้องกันการ
ใช้ทรัพยากรที่มากเกินไป
- การ Loop
หากต้องการให้ Particle System เล่นซ้ำ ควรเปิดใช้งาน
Looping ใน Main Module
- การแสดงผล
เลือก Material และ Shader ที่เหมาะสมเพื่อปรับปรุง
คุณภาพของเอฟเฟกต์



ตัวอย่างการแสดง Particle เมื่อผู้เล่นกดแป้น E



ในการแสดงผล Particle System เมื่อผู้เล่น
กดแป้น E คุณสามารถควบคุม Particle
System ผ่านสคริปต์โดยใช้คำสั่ง Play()
และ Stop() เพื่อเริ่มหรือหยุดการแสดงผล



การตั้งค่า

- เพิ่ม Particle System
 - คลิกขวาที่ Hierarchy > Effects > Particle System
 - ตั้งค่าคุณสมบัติของ Particle System เช่น สี, ขนาด, หรือรูปทรงตามที่ต้องการ



- **เพิ่ม Script**
 - สร้าง C# Script ชื่อ ShowParticleOnKeyPress และเพิ่มลงใน GameObject หัก เช่น Player
- **เชื่อมโยง Particle System**
 - ลาก Particle System จาก Hierarchy ไปยังช่อง Particle System ใน Inspector ของ Script
- **ทดสอบการทำงาน**
 - กดปุ่ม Play และกดแป้น E เพื่อเริ่มแสดง Particle System



```
using UnityEngine;
```

```
public class ShowParticleOnKeyPress : MonoBehaviour
{
    public ParticleSystem particleSystem; // อ้างอิงถึง Particle System ที่จะเล่น

    void Update()
    {
        // ตรวจสอบว่าผู้เล่นกดแป้น E
        if (Input.GetKeyDown(KeyCode.E))
        {
            // เล่น Particle System
            if (particleSystem != null)
            {
                particleSystem.Play();
                Debug.Log("Particle System เล่นแล้ว");
            }
        }
    }
}
```



```
// ตรวจสอบว่าผู้เล่นปล่อยแป้น E
if (Input.GetKeyUp(KeyCode.E))
{
    // หยุด Particle System
    if (particleSystem != null)
    {
        particleSystem.Stop();
        Debug.Log("Particle System หยุดแล้ว");
    }
}
}
```



ตรวจจับการกดแป้น E

```
Input.GetKeyDown(KeyCode.E)
```

ตรวจจับเมื่อแป้น E ถูกกด และหากตรวจพบการกด จะเรียก

```
particleSystem.Play()
```

เพื่อเริ่มการแสดงผล Particle System



ตรวจจับการปล่อยแป้น E

```
Input.GetKeyUp(KeyCode.E)
```

ตรวจจับเมื่อแป้น E ว่าถูกปล่อยจะเรียก

```
particleSystem.Stop()
```

เพื่อหยุดการแสดงผล Particle System



การตรวจสอบ Particle System

ใช้การตรวจสอบ

```
if (particleSystem != null)
```

เพื่อป้องกันข้อผิดพลาดหาก Particle System ไม่ได้ถูกตั้งค่าใน
Inspector



เพิ่มเวลาในการเล่น Particle System

```
if (Input.GetKeyDown(KeyCode.E)) {  
    if (particleSystem != null) {  
        particleSystem.Play();  
        Invoke("StopParticle", 2f); // หยุดหลังจาก 2 วินาที  
    }  
}  
  
void StopParticle() {  
    if (particleSystem != null) {  
        particleSystem.Stop();  
    }  
}
```



เพิ่มเสียงเมื่อกดแป้น E

```
public AudioSource audioSource; // เพิ่ม Audio Source

if (Input.GetKeyDown(KeyCode.E))
{
    if (particleSystem != null)
    {
        particleSystem.Play();
        audioSource.Play(); // เล่นเสียง
    }
}
```



แสดง Particle System ที่ตำแหน่งเฉพาะ

```
public Transform targetPosition; // ตำแหน่งเป้าหมาย

if (Input.GetKeyDown(KeyCode.E))
{
    if (particleSystem != null)
    {
        // ตั้งตำแหน่งใหม่
        particleSystem.transform.position = targetPosition.position;
        particleSystem.Play();
    }
}
```



Q & A

www.gamedevpbru.com

