

1/2563

ชื่อ.....

รหัสนักศึกษา.....

ตอนที่.....

สาขาวิชาเอนนิเมชันและเกม วิทยาลัยศิลปะ สื่อ และเทคโนโลยี มหาวิทยาลัยเชียงใหม่



เนื้อหา (Contents)

9 การจัดแสงในฉาก แสงทางอ้อม และเอฟเฟคต์หลังกระบวนการเรนเดอร์ (Scene Lighting, Global Illumination, and Post Processing Effects)	3
9.1 บทนำ	3
9.2 ชนิดของแหล่งกำเนิดแสงพื้นฐานใน Unity (Basic Types of Light Source in Unity)	3
9.2.1 แหล่งกำเนิดแสงแบบกำหนดทิศทาง (Directional Light)	3
9.2.2 แหล่งกำเนิดแสงแบบจุด (Point Light)	5
9.2.3 แหล่งกำเนิดแสงแบบ Spot (Spot Light)	7
9.2.4 โหมดต่าง ๆ ของแหล่งกำเนิดแสง (Light Modes)	8
9.3 คุณสมบัติ Lighting บนคอมโพเนนท์ Mesh Renderer	8
9.4 การอบแสง (Baked Lighting)	10
9.4.1 การกำหนดแสงและองค์ประกอบจากให้มีการอบแสง (Setting Up Baked Lighting)	10
9.4.2 การกำหนดอุปกรณ์ตัวคำนวณการอบแสง (Setting Up Compute Device for Baking)	11
9.4.3 ตัวคูณแสงทางอ้อม (Indirect Multiplier)	12

9.4.4	พื้นผิวเปล่งแสง (Emission Material)	13
9.5	การทดลอง สร้างจากทดลองเรื่องแสงและการอบแสง	14
9.5.1	Non Static Room (Realtime Light)	14
9.5.2	Non Static Room (Baked Light)	16
9.5.3	Static Room (Baked Light)	17
9.5.4	ทดลองเพิ่มเติม	18
9.6	Light Probes	18
9.7	Reflection Probes	20
9.8	ท้องฟ้าจำลอง (Skybox)	21
9.9	พระอาทิตย์จำลอง (Artificial Sun)	22
9.10	การตั้งค่าวัตถุเพื่อให้สามารถรับ Lightmap จากการอบแสง	24
9.10.1	ไฟล์โมเดล FBX	24
9.10.2	วัตถุที่สร้างจาก ProBuilder	25
9.11	Emissive Texture	25
9.12	Post-Processing	25
9.12.1	การตั้งค่าเพื่อใช้งาน Post-Processing	33
9.13	การแก้ปัญหาต่าง ๆ ที่อาจจะเกิดขึ้น (Troubleshooting)	38
9.14	คำถามและปัญหาชวนคิด (Questions and Problems)	39

บทที่ 9

การจัดแสงในฉาก แสงทางอ้อม และอฟเฟคต์หลังกระบวนการเรนเดอร์ (Scene Lighting, Global Illumination, and Post Processing Effects)

วัตถุประสงค์ (Objectives)

- ศึกษาชนิดของแหล่งกำเนิดแสงแบบต่าง ๆ
- ศึกษา Realtime
- ศึกษา Baked Lighting
- ศึกษา Global Illumination

9.1 บทนำ

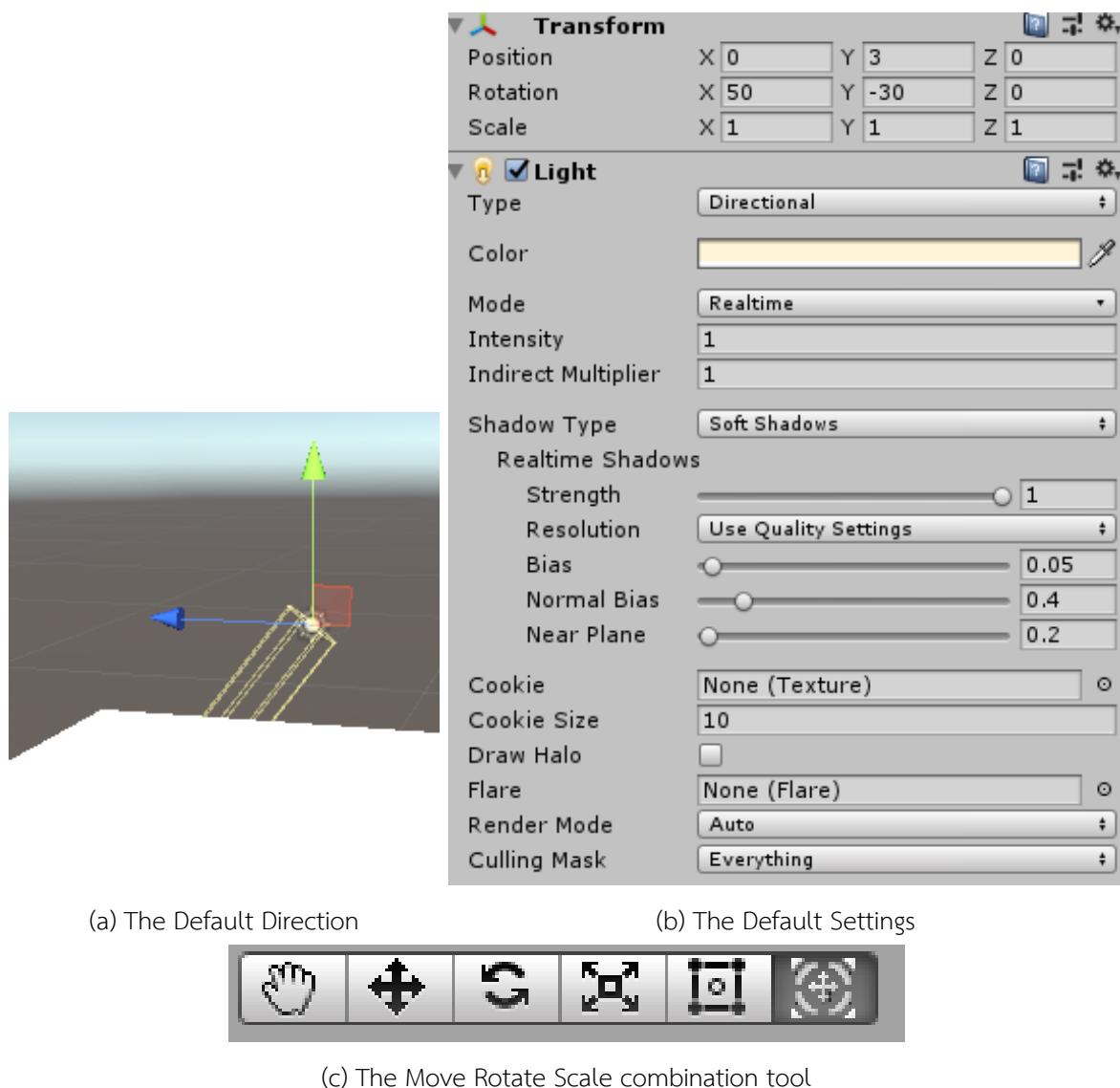
บทนี้ศึกษาการใช้งานแหล่งกำเนิดแสงชนิดต่าง ๆ และเทคนิคในการจัดแสงให้กับฉากในเกม เพื่อให้ฉากของเกมมีความสวยงามและสามารถทำงานได้อย่างมีประสิทธิภาพ

9.2 ชนิดของแหล่งกำเนิดแสงพื้นฐานใน Unity (Basic Types of Light Source in Unity)

9.2.1 แหล่งกำเนิดแสงแบบกำหนดทิศทาง (Directional Light)

แหล่งกำเนิดแสงพื้นฐานที่สุดใน Unity เมื่อผู้พัฒนาสร้างจากขึ้นมาหนึ่งจากคือแสงแบบ Directional Light ที่ถูกสร้างขึ้นมาโดยอัตโนมัติด้วยรูปที่ 9.1 โดยที่ตัวแหล่งกำเนิดแสงมีการจัดวาง (Transform) ที่ตำแหน่ง (0,3,0) และทำมุ่ง (50,-30,0) กับแกน (X,Y,Z) ตามลำดับ เป็นค่าเริ่มต้น

ตำแหน่งของแหล่งกำเนิดแสงแบบ Directional Light ไม่มีผลต่อการให้แสงในฉากของเกม มีเพียงมุ่งของการจัดวางของแหล่งกำเนิดแสงเท่านั้นที่มีผลต่อแสง ค่าปกติของมุ่งการจัดวางคือ $(50, -30, 0)$ ซึ่งทำให้แสงส่องลงทำมุมดังกล่าว โดยสังเกตได้จาก รูปที่ 9.1a นอกจากนี้แสงประเภท Directional Light สามารถถูกปรับมุ่งของแสงด้วยการปรับที่คุณสมบัติ Rotation ของคอมโพเนนท์ Transform เพื่อให้ลักษณะของแสงที่ส่องลงบนฉากเปลี่ยนแปลงไป เครื่องมือดัง รูปที่ 9.1c สามารถใช้งานเพื่อปรับทิศทางของแสงได้ด้วยการลากเม้าส์บนหน้าต่าง Scene



รูปที่ 9.1: The Default Directional Light

ให้ทดลองปรับแสง Directional Light ให้มีมุ่ง $(-15, -60, 80)$ จะทำให้ฉากมีดลิง

9.2.2 แหล่งกำเนิดแสงแบบจุด (Point Light)

แหล่งกำเนิดแสงแบบจุดหรือ Point Light มีลักษณะเป็นแหล่งกำเนิดที่กระจายแสงออกจากแหล่งกำเนิดอย่างรอบทิศทาง ในลักษณะทรงกลมแสดงดัง รูปที่ 9.2

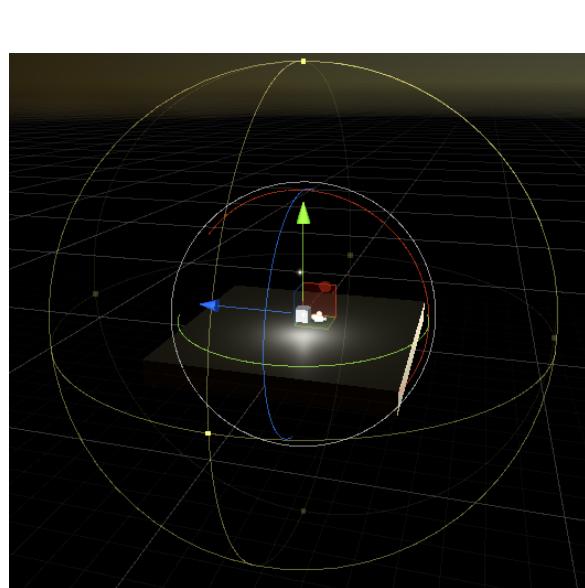
มีคุณสมบัติต่าง ๆ ที่สำคัญคือ

- Type ชนิดของแสง ในที่นี้คือ Point
- Range ขอบเขตซึ่มีความกว้างของแสงมีหน่วยเป็นเมตร สั้นเกตค่าปกติคือ 10 คือรัศมี 10 เมตร แสดงด้วยขอบเขตทรงกลมสีเหลือง
- Color สีของแหล่งกำเนิดแสง
- Mode โหมดของแสงให้เป็นแบบ Realtime Mixed หรือ Baked
- Intensity ความเข้มของแสง
- Indirect Multiplier ตัวคูณความเข้มของแสงเมื่อสะท้อนห้องที่สอง ยังไม่มีผลหากใช้โหมดแบบ Realtime
- Shadow Type ให้มีการทดสอบจากแหล่งกำเนิดแสงนี้หรือไม่ และอย่างไร

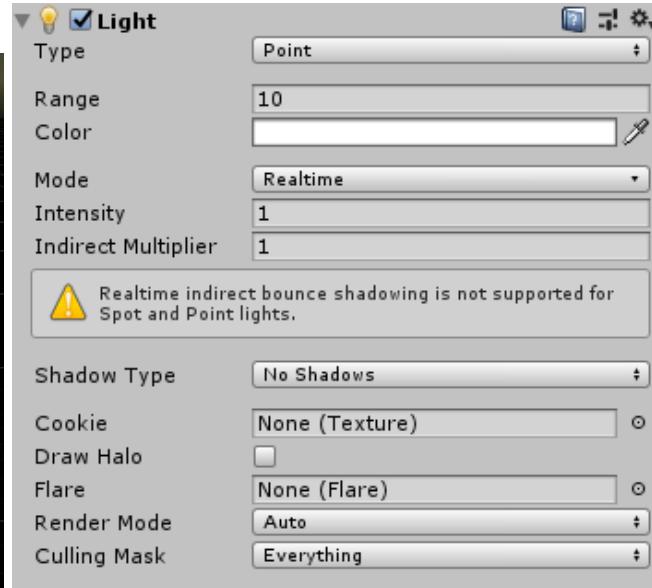
No Shadows ไม่ทดสอบที่เกิดขึ้นจากแหล่งกำเนิดแสงแหล่งนี้

Hard Shadows ทดสอบที่เกิดขึ้นจากแหล่งกำเนิดแสงแหล่งนี้แบบของเงาแข็ง

Soft Shadows ทดสอบที่เกิดขึ้นจากแหล่งกำเนิดแสงแหล่งนี้แบบของเงานุ่มนวล



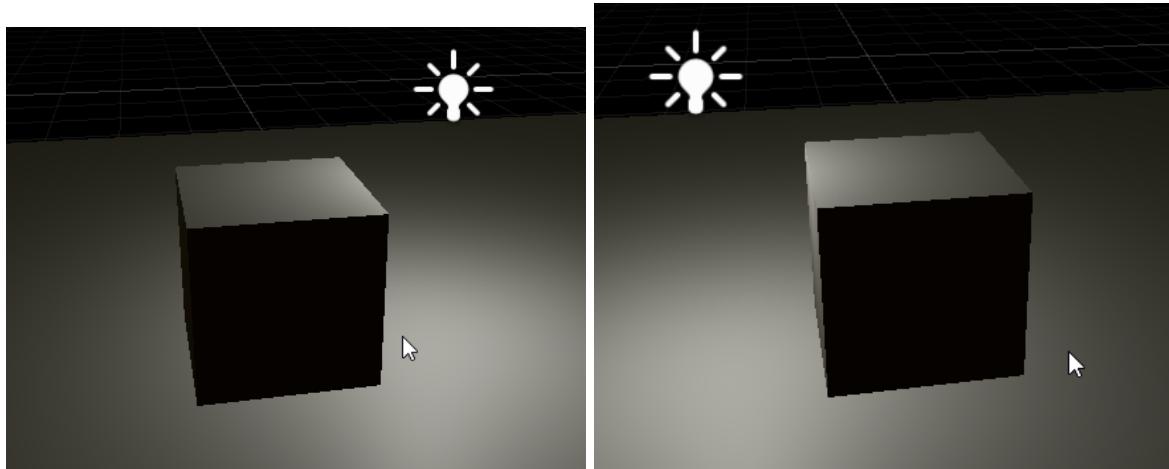
(a) Point Light



(b) Point Light Settings

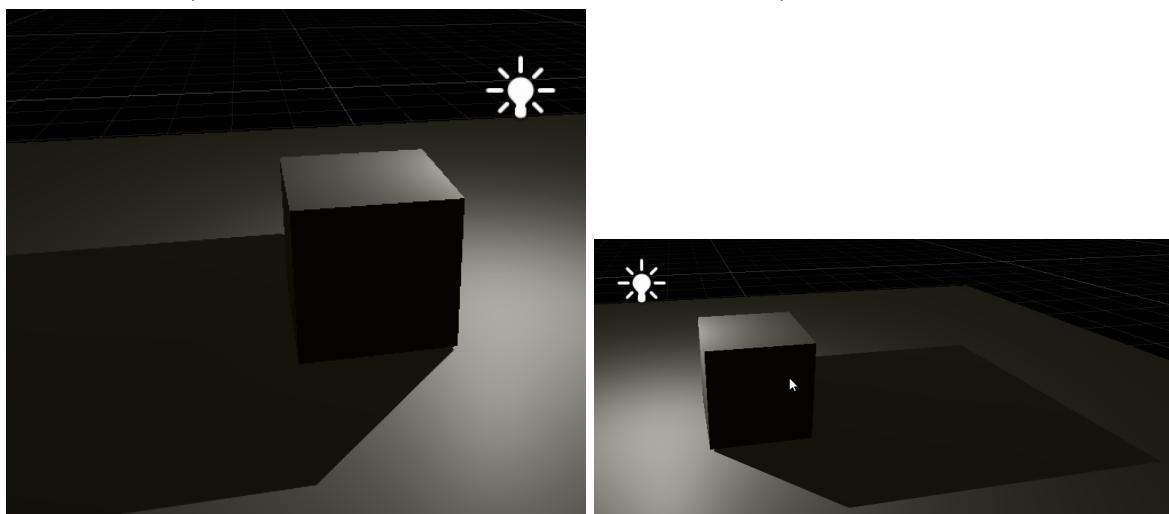
รูปที่ 9.2: The Point Light

ตำแหน่งที่ตั้งของแหล่งกำเนิดแสงแบบ Point Light ส่งผลต่อการให้แสงในรายการทิศทางตั้งตัวอย่าง รูปที่ 9.3 แต่หมุนจะไม่มีผลเนื่องจากแสงมีลักษณะแผ่นกว้างแบบทรงกลม (Spherical)



(a) Example Position 1 No Shadows

(b) Example Position 2 No Shadows



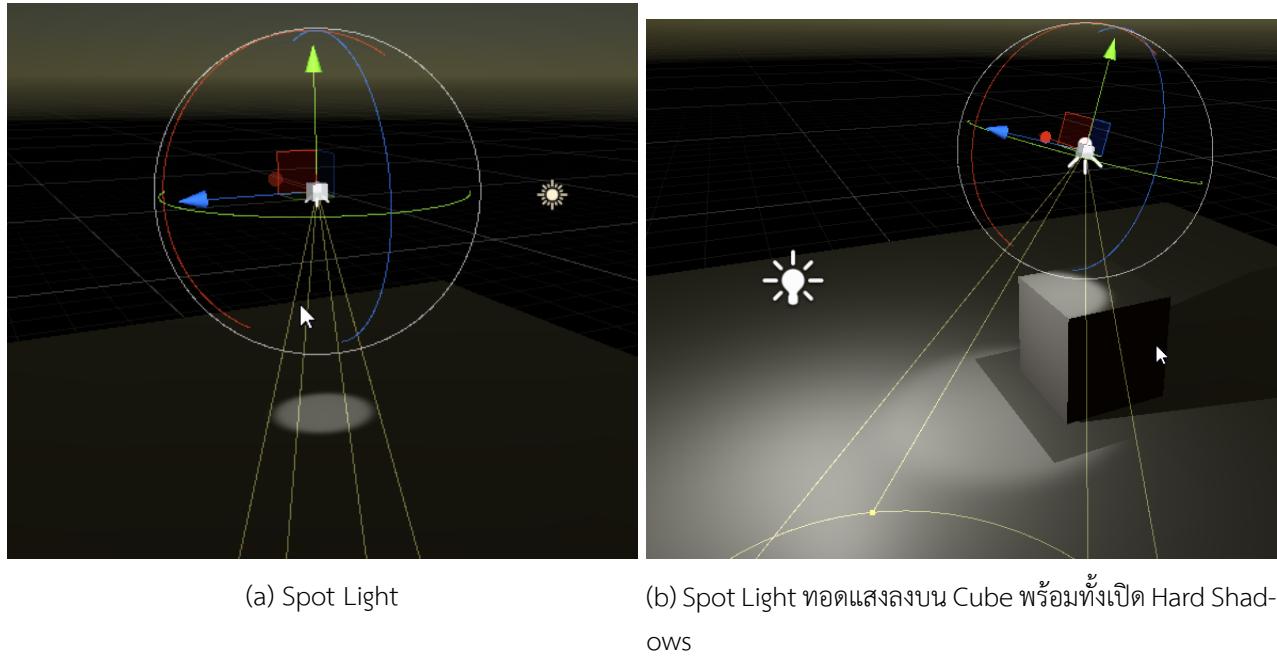
(c) Example Position 1 Hard Shadows

(d) Example Position 2 Hard Shadows

รูปที่ 9.3: The Point Light Positioning and Shadow

9.2.3 แหล่งกำเนิดแสงแบบ Spot (Spot Light)

แหล่งกำเนิดแสงแบบสปอตไลท์ ซึ่งมีพิเศษทางของแสงฉายออกໄไปในลักษณะโคนแสดงตั้ง รูปที่ 9.4 คุณสมบัติต่าง ๆ ของ Spot Light มีลักษณะคล้ายคลึงกับ หัวข้อ 9.2.2 แหล่งกำเนิดแสงแบบจุด (Point Light) แตกต่างตรงที่ Range จะเป็นค่าของระยะที่พุ่งออกไปจากแหล่งกำเนิดแสงแทนรัศมี มีหน่วยเป็นเมตร และมีคุณสมบัติ Spot Angle เพิ่มเติมขึ้นมา คือค่ามุมองศาการแผ่กระจายของแสง



รูปที่ 9.4: The Spot Light

9.2.4 โหมดต่าง ๆ ของแหล่งกำเนิดแสง (Light Modes)

Realtime

การคำนวณแสงจะเกิดขึ้นทุก ๆ เฟรมของเกมลูป ณ Runtime

Unity จะไม่มีการคำนวณแสงล่วงหน้า (Precompute) สำหรับโหมดแสงแบบ Realtime

Baked

มีการคำนวณแสงล่วงหน้า (Precompute) เป็น Lightmaps และนำไปรวมผลกับภาพพื้นผิวเพื่อให้เกิดเป็นพื้นผิวเสมือนว่าถูกแสงส่องสว่างถึงจริง แต่ไม่มีการเปลี่ยนแปลง ณ Runtime ทำให้มีศูนย์เสียการคำนวณแสง ณ runtime

Mixed

ในโหมดนี้จะมีการคำนวณแสงบางอย่างล่วงหน้า (Precompute) และบางอย่าง ณ runtime ซึ่งเป็นการผสมผสานแบบ Realtime และ Baked lightings

9.3 คุณสมบัติ Lighting บนคอมโพเนนท์ Mesh Renderer

รูปที่ 9.5 แสดงคอมโพเนนท์ Mesh Renderer ซึ่งมีคุณสมบัติที่เกี่ยวข้องกับการกำหนดการคำนวณเรื่องแสงอยู่ในหัวข้อต่อไปนี้

- Lighting สำหรับกำหนดพฤติกรรมของวัตถุ Mesh ที่มีต่อแหล่งกำเนิดแสง

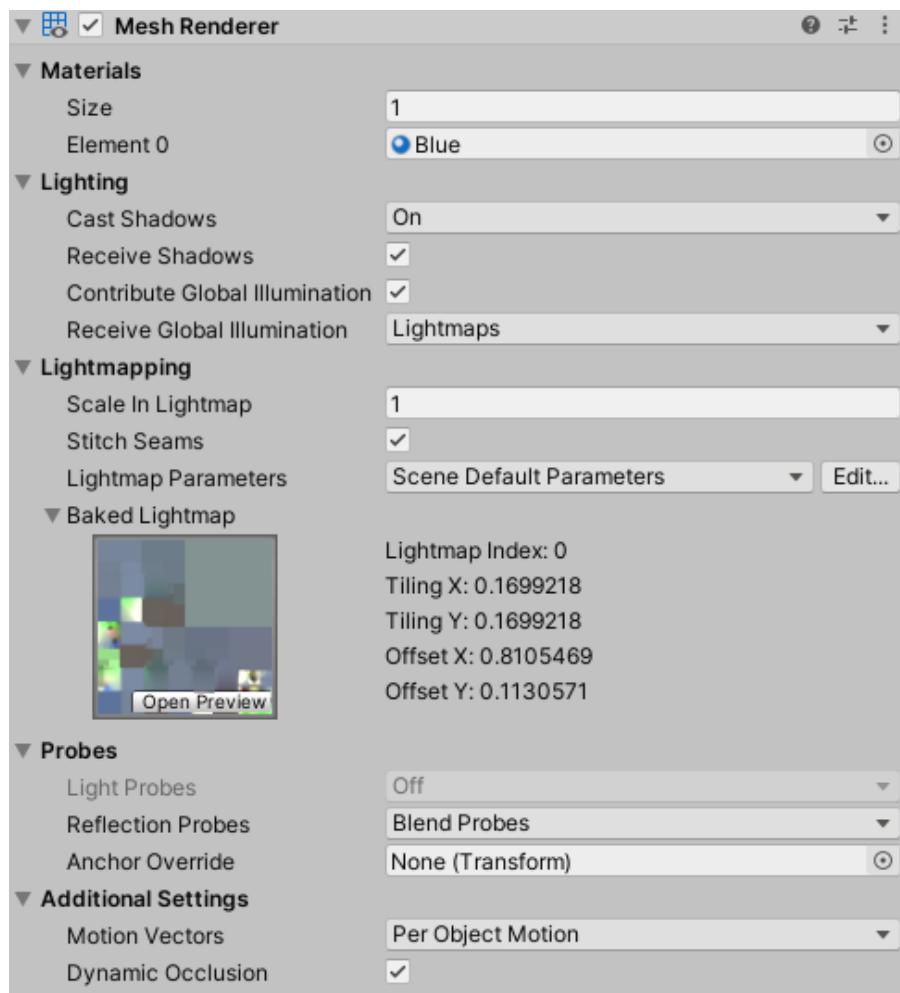
Cast Shadows คือการกำหนดให้ Mesh ชิ้นนี้ถ่ายทอดเงาหรือไม่ เมื่อมีแสงตกกระทบ

Receive Shadows คือการกำหนดให้ mesh รับทอดเงาหรือไม่ เมื่อมีการถ่ายทอดเงางบน Mesh

Contribute Global Illumination คือการกำหนดว่า Mesh ชิ้นนี้จะให้ผลส่งผลกระทบสะท้อนแสงทางอ้อมไปยังวัตถุอื่น ๆ หรือไม่

Receive Global Illumination คือการกำหนดลักษณะการรับแสง GI แบบ Lightmaps หรือ Light Probes

- Lightmapping กำหนดรายละเอียดเพื่อควบคุมคุณภาพเกี่ยวกับ Light Map
- Probes กำหนดพฤติกรรมของวัตถุ mesh ที่มีต่อ Light Probes และ Reflection Probes



(a) MeshRenderer Component

รูปที่ 9.5: คอมโพnenท์ MeshRenderer

9.4 การอบแสง (Baked Lighting)

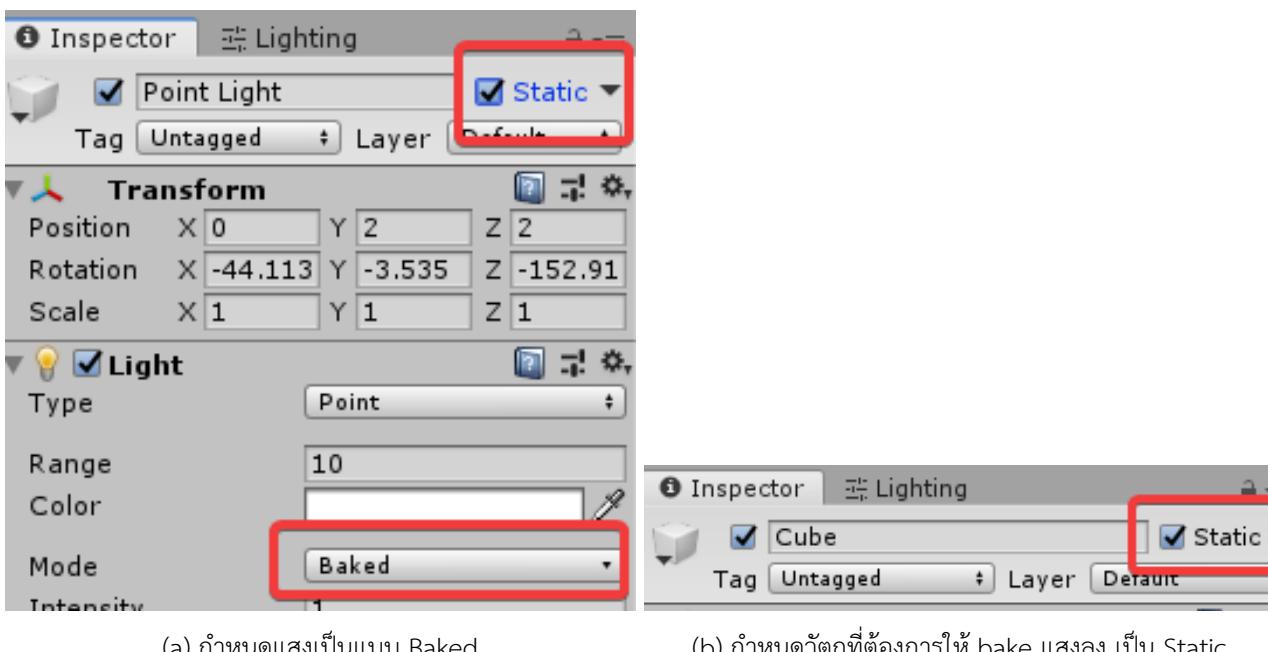
การอบแสง คือ การคำนวณค่าความเข้มของแสงที่ทอตลงบนวัตถุไว้ล่วงหน้า ณ ตอนออกแบบฉาก (Design time rather than runtime) และเก็บข้อมูลไว้เป็นภาพพื้นผิวแสง (Lightmapping) เพื่อเป็นการประหยัดทรัพยากรหุ่นว่ายประมวลผลในการคำนวณแสงแบบ Realtime

ในการจัดและตกแต่งฉากของเกมจาก ๆ หนึ่งจะประกอบไปด้วยทั้งวัตถุที่เคลื่อนที่ได้ (Dynamic) และวัตถุที่อยู่กับที่ (Static) เกมเอนจินได้ใช้จุดนี้เป็นเทคนิคในการคำนวณแสงล่วงหน้า (precompute) ทำให้เกมมีประสิทธิภาพที่ดีขึ้นเนื่องจากไม่จำเป็นต้องคำนวณแสงแบบ Realtime สำหรับทุก ๆ แหล่งกำเนิดแสงในฉาก ในขณะที่จากมีความพยายามจากการคำนวณแสงอย่างละเอียดไว้บน Lightmapping texture

9.4.1 การกำหนดแสงและองค์ประกอบฉากให้มีการอบแสง (Setting Up Baked Lighting)

การทำการอบแสงจะต้องมีองค์ประกอบสองส่วนด้วยกันคือ 1.เลือกแหล่งกำเนิดแสงเป็นโหมดแบบ Baked และ 2.วัตถุในฉากที่จะถูกคำนวณการตกลงระทบของแสงเป็นวัตถุที่กำหนดให้อยู่กับที่ (Static)

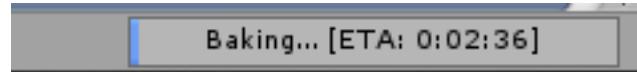
- ให้เลือกแหล่งกำเนิดแสง และกำหนด Mode ของแหล่งกำเนิดแสงจากปกติ Realtime เป็น Baked แสดงดังรูปที่ 9.6a
- กำหนดวัตถุประกอบฉากให้เป็น Static ดังรูปที่ 9.6b



รูปที่ 9.6: Setting Point Light for Baking

เมื่อกำหนดองค์ประกอบของการคำนวณการอบแสงได้ถูกต้องแล้ว Unity จะคำนวณแสงตกลงระทบ โดยมีการแสดงสถานะ

การคำนวณที่ด้านล่างของหน้าต่าง Unity ดังรูปที่ 9.7 แสดงเวลาประมาณในการคำนวณแสง ในที่นี่คือใช้เวลาประมาณ 2.36 นาที



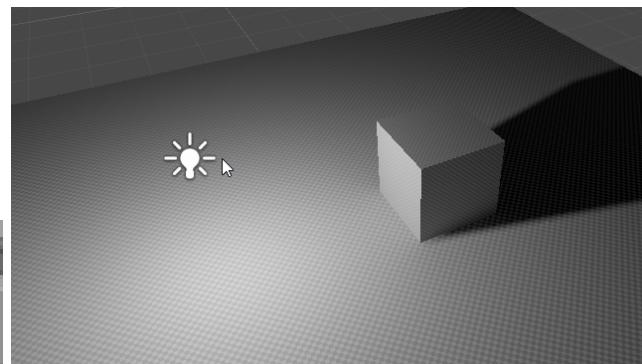
(a) Baking Progress

รูปที่ 9.7: Baking Status

ผู้พัฒนาสามารถดูผลของการอบแสง (Lightmap) ได้ด้วยการเลือกเมนูจากหน้าต่าง Scene จาก Shaded เป็น Baked Lightmap ดังรูปที่ 9.8a



(a) แสดงผล Texture ของการอบแสงบนหน้าต่าง Scene



(b)



(c) ผลของการ Baked Lightmap บน texture ของวัตถุที่หน้าต่าง inspector ของวัตถุ

รูปที่ 9.8: Baked Light Map

9.4.2 การกำหนดอุปกรณ์ตัวคำนวณการอบแสง (Setting Up Compute Device for Baking)

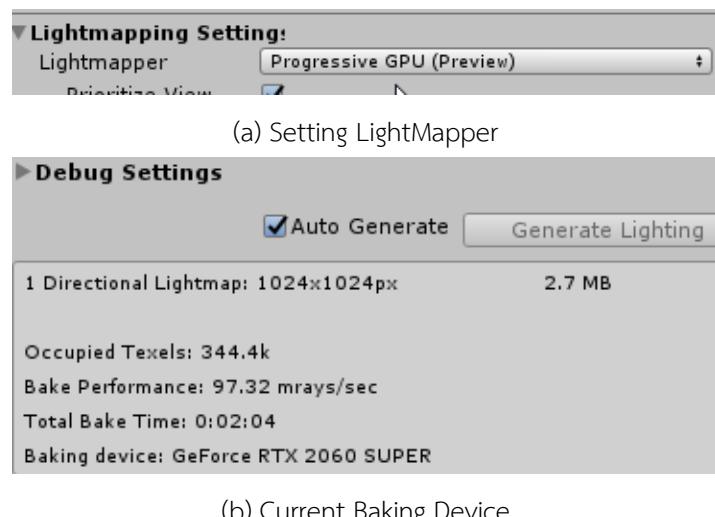
การคำนวณการอบแสงต้องใช้เวลาและพลังในการคำนวณ ค่าปกติ Unity จะใช้หน่วยประมวลผลกลาง (CPU) ในการคำนวณ ซึ่งมีจำนวนแกนคิด (Compute Core) ในการคำนวณค่อนข้างจำกัด การใช้หน่วยประมวลผลกราฟิก (GPU) จะช่วยให้การคำนวณแสงได้เร็วกว่า

เปิดหน้าต่าง Window->Rendering->Lighting Settings

ตั้งค่าส่วนของ Lightmapper Settings->Lightmapper จาก Progressive CPU ให้เป็น Progressive GPU (preview) ดังรูปที่ 9.9

หากตั้งค่าได้ถูกต้องส่วนของ Lighting Settings ในส่วนล่างสุดจะแสดงอุปกรณ์คำนวณดัง รูปที่ 9.9b เป็นกราฟิก GPU แทนที่ CPU หรือการ์ดจอ On-Board

Option Auto Generate เป็นการบอกให้ Unity คำนวณการอบแสงใหม่ทุกครั้งที่มีการขับเคลื่อนโดยเนิดแสงหรือวัตถุ Static ซึ่งต้อง Bake ใหม่ทุก ๆ ครั้งที่มีการขับปรับเปลี่ยน หากไม่ต้องการให้กระทำโดยอัตโนมัติให้ปิด Option นี้ และทำการสั่งคำนวณแสงใหม่เมื่อการกดปุ่ม Generate Lighting ที่หน้าต่างนี้

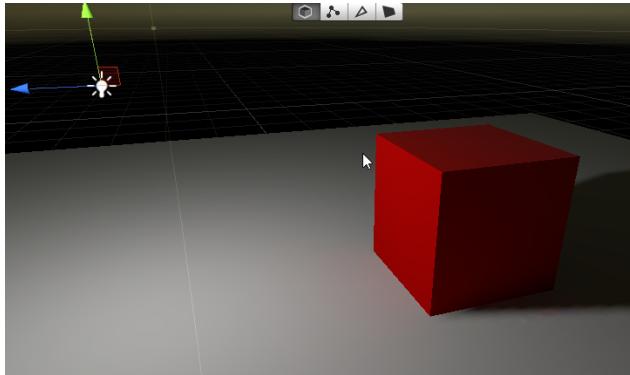


(b) Current Baking Device

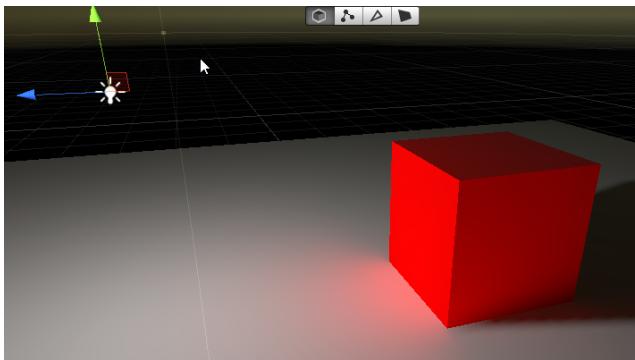
รูปที่ 9.9: Light Settings

9.4.3 ตัวคูณแสงทางอ้อม (Indirect Multiplier)

ตัวคูณแสงทางอ้อมจะมีผลกับ Baked Lighting ทำให้แสงที่ตกกระทบกับวัตถุหนึ่งสะท้อนถ่ายทอดความเข้มและสีไปยังอีกวัตถุหนึ่งแสดงดังรูปที่ 9.10



(a) Point Light Indirect Multiplier = 1



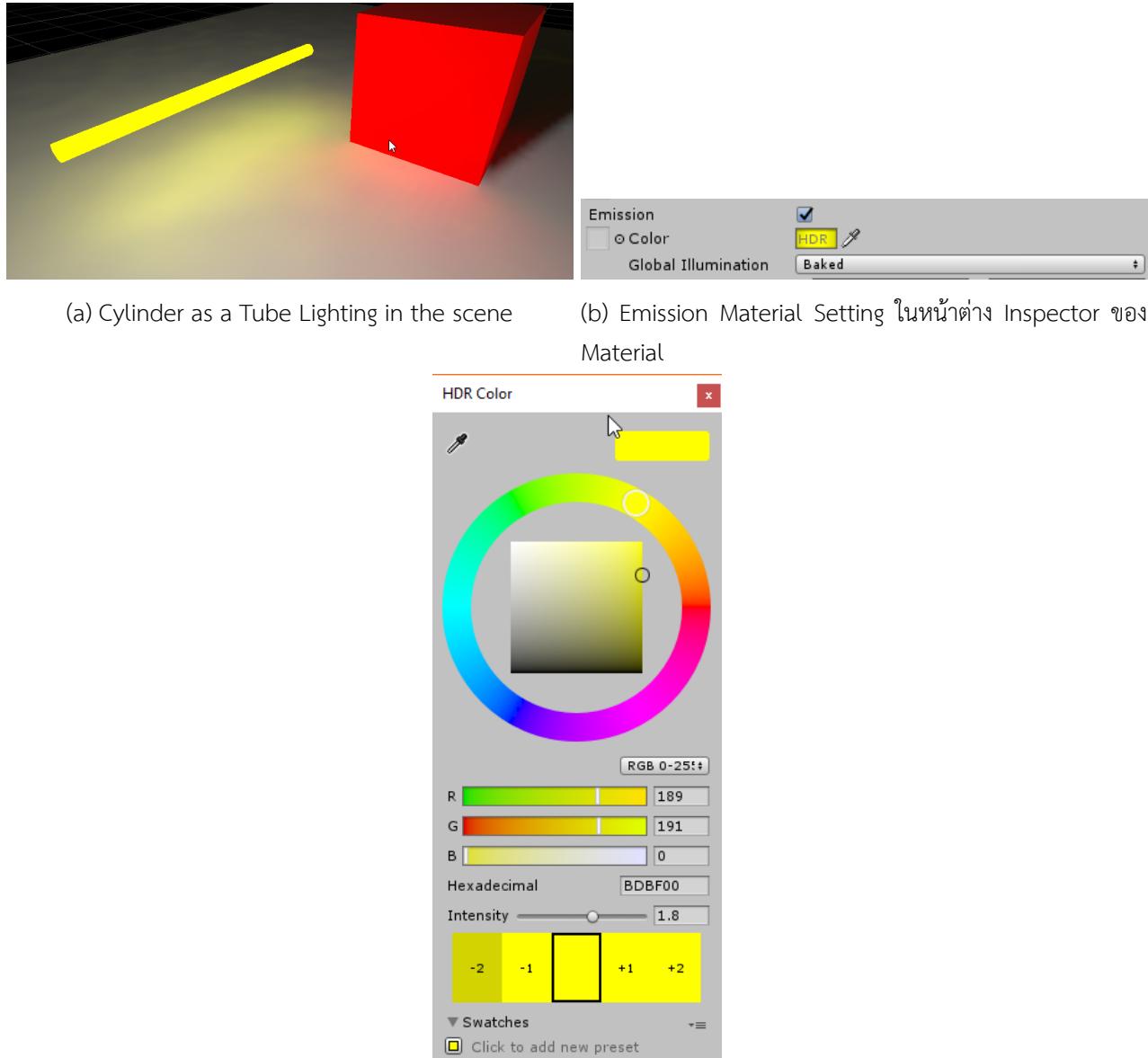
(b) Point Light Indirect Multiplier = 3

รูปที่ 9.10: IndirectLight Multiplier Setting

9.4.4 พื้นผิวเปล่งแสง (Emission Material)

วัตถุรูปทรงต่าง ๆ ในฉากรสามารถถูกกำหนดให้เปล่งแสงได้ด้วยการกำหนดพื้นผิว (Material) เป็นแบบ Emission และมีผลต่อการให้แสงในฉากด้วยการกำหนดให้เป็นวัตถุ Static ผลลัพธ์แสดงตั้งรูปที่ [9.11](#)

Material ที่สร้างขึ้น ให้กำหนดใช้งาน Emission รูปที่ [9.11b](#) และตั้งค่า Intensity ด้วยการกดเข้าไปที่หน้าต่าง HDR Color ดังรูปที่ [9.11c](#)



รูปที่ 9.11: Emission Material

9.5 การทดลอง สร้างฉากทดลองเรื่องแสงและการอบแสง

สร้างโค้ดและการแยกขึ้นมาใหม่ ปิดการใช้งานแสง Directional Light ที่ถูกสร้างไว้ต้นพร้อมกับฉาก

9.5.1 Non Static Room (Realtime Light)

จัดฉากดังรูปที่ 9.12a โดยมีวัตถุเกมต่าง ๆ ดังต่อไปนี้

- พื้นชื่อ Floor

ตั้งค่าคอมโพเนนท์ Transform ดังนี้ Position(0,0,0) Rotation(0,0,0) Scale(5,1,5)

- กำแพงด้านขวา ชื่อ Wall

ตั้งค่าคอมโพเนนท์ Transform ดังนี้ Position(3,2,0) Rotation(0,0,90) Scale(5,1,5)

กำหนด Material สีแดง

- กำแพงด้านซ้าย ชื่อ Wall(1)

ตั้งค่าคอมโพเนนท์ Transform ดังนี้ Position(-3,2,0) Rotation(0,0,90) Scale(5,1,5)

กำหนด Material สีขาว

- กำแพงด้านหลัง ชื่อ Wall(2)

ตั้งค่าคอมโพเนนท์ Transform ดังนี้ Position(0,2,3) Rotation(0,90,90) Scale(5,1,5)

กำหนด Material สีเขียว

- ลูกบอลกลม ชื่อ Sphere

ตั้งค่าคอมโพเนนท์ Transform ดังนี้ Position(0.3,1.5,1) Rotation(0,0,0) Scale(2,2,2)

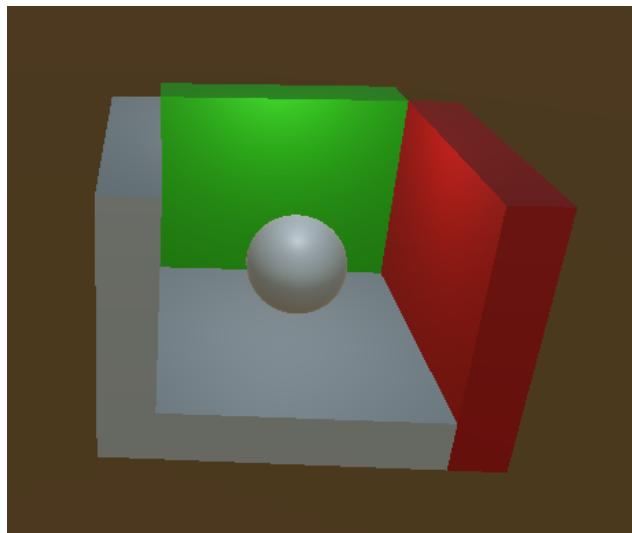
กำหนด Material สีขาว

- Point Light

ตั้งค่าคอมโพเนนท์ Transform ดังนี้ Position(0,5,0)

Light->Indirect Multiplier จาก 1 เป็น 3

สร้างวัตถุเปล่า (Empty Game Object) กำหนด Position(0,0,0) เปลี่ยนชื่อเป็น “Non Static Room (Realtime Light)” และนำวัตถุข้างต้นลากเข้าเป็น Child ของวัตถุเปล่าดังรูปที่ 9.12b



(a) Non Static Room



(b) Non Static Room

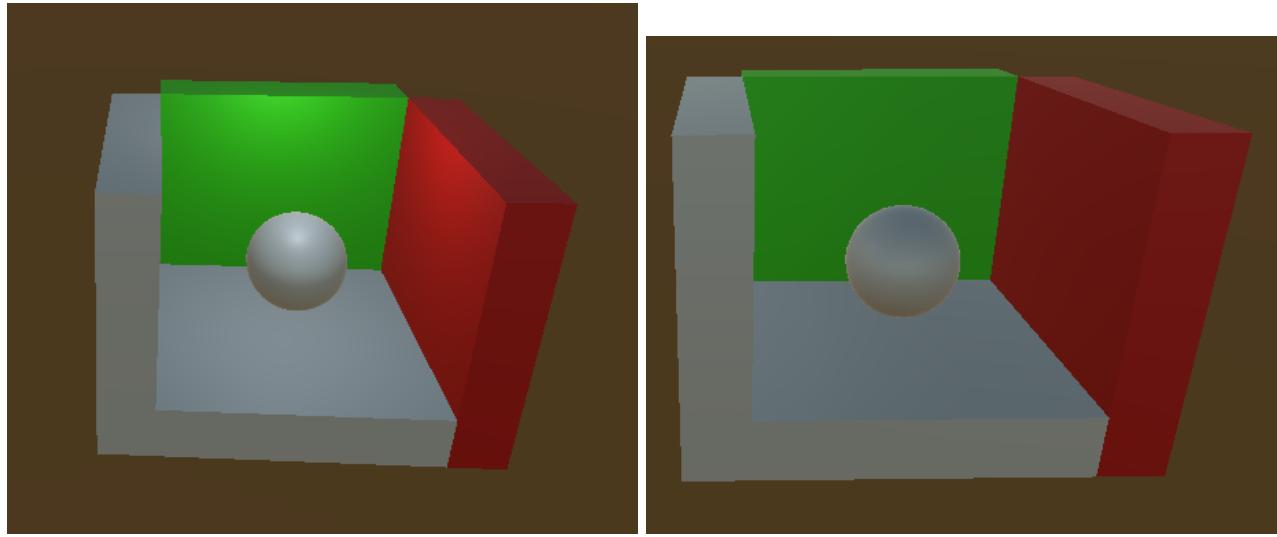
รูปที่ 9.12: Non static objects room

9.5.2 Non Static Room (Baked Light)

เลือกวัตถุเปล่า “Non Static Room (Realtime Light)” ทำการสำเนา (Duplicate) ด้วยการกด Ctrl+D กำหนด Transform->Position(8,0,0)

เปลี่ยนชื่อเป็น Non Static Room (Baked Light)

เปลี่ยนคุณสมบัติ “Non Static Room (Baked Light)”->Point Light->Mode จาก Realtime เป็น Baked สังเกต ความเปลี่ยนแปลงดังรูปที่ 9.13



(a) Realtime

(b) Baked

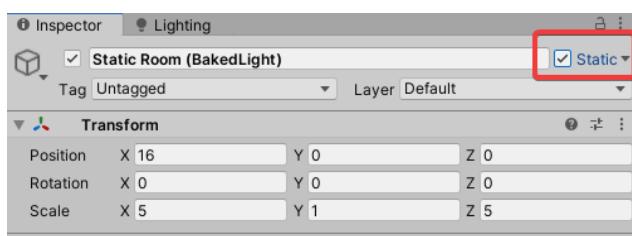
รูปที่ 9.13: Non Static Room: Realtime vs Baked Lightings

9.5.3 Static Room (Baked Light)

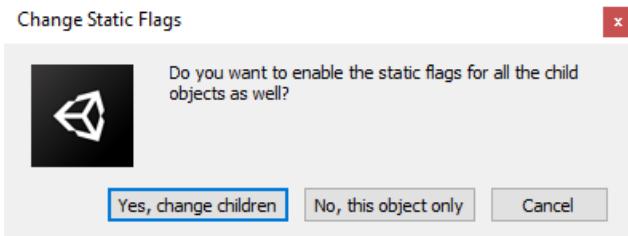
เลือกวัตถุเปล่า “Non Static Room (Baked Light)” ทำการสำเนา (Duplicate) ด้วยการกด Ctrl+D กำหนด Transform->Position(16,0,0)

เปลี่ยนชื่อเป็น Static Room (Baked Light)

เลือกวัตถุเปล่า Static Room (Baked Light) เปลี่ยนเป็นวัตถุ Static โดยการกดเลือกดังรูปที่ 9.14a และเลือก Change Children ดังรูปที่ 9.14b



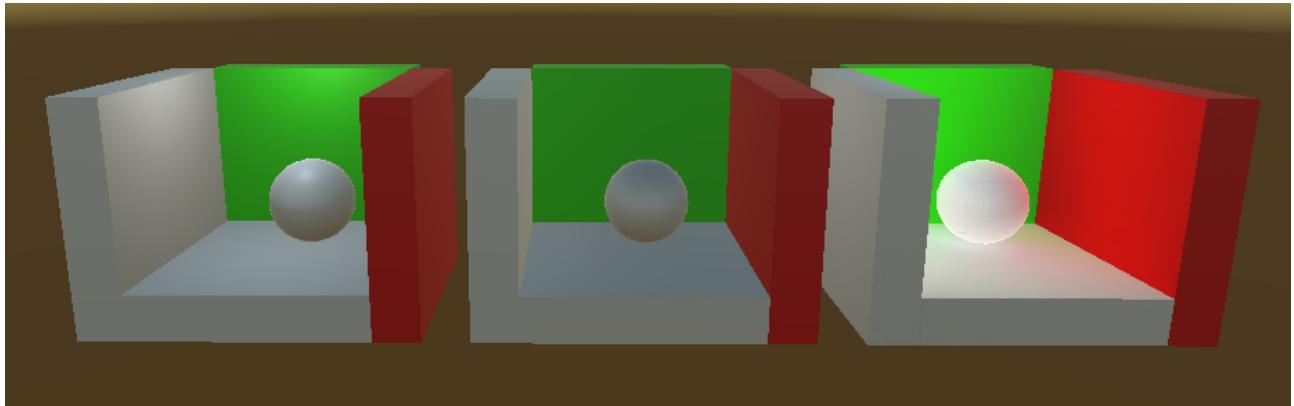
(a) Change the object to be a static object



(b) Baked

รูปที่ 9.14: Static Room Baked Lighting

ทดสอบผลลัพธ์ดังตัวอย่างรูปที่ 9.15



(a) Shaded view



(b) Baked Lightmap view

รูปที่ 9.15: Three rooms comparison

9.5.4 ทดลองเพิ่มเติม

ทดลองปิดการใช้งานไฟ Point Light ทั้งสามดวงในห้องทั้งสาม เปิดใช้งาน Directional Light พร้อมทั้งตั้งค่า Indirect Multiplier เป็น 3 และเปลี่ยนโหมดเป็น Mixed หรือ Baked

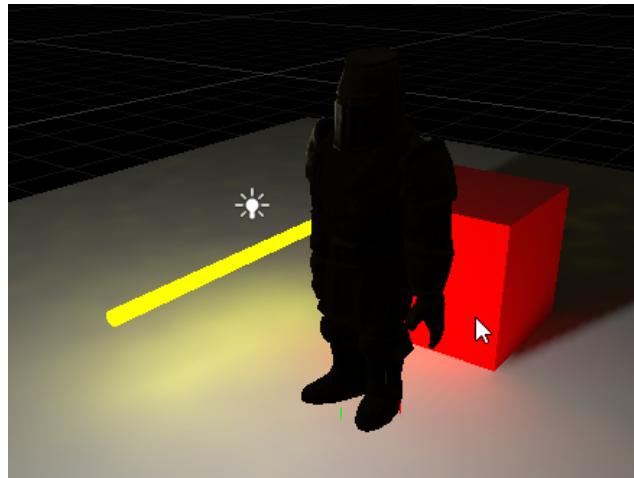
สังเกตผลลัพธ์ที่ได้

9.6 Light Probes

เนื่องจากระบบ Baked Lighting คือระบบที่คำนวณค่าของแสงที่ตกกระทบไว้ล่วงหน้า ซึ่งมีผลต่อวัตถุประเภท Static (วัตถุที่อยู่ที่เดิมตลอดเวลา) ซึ่งทำให้เกิดข้อด้อยคือวัตถุที่สามารถเคลื่อนที่ได้ (Dynamic GameObject) จะไม่ได้รับแสง Baked Lighting จากการสะท้อนของแสงจากสภาพแวดล้อม

รูปที่ 9.16 แสดงให้เห็นว่าวัตถุที่เคลื่อนไหวได้ (Dynamic Object) จะไม่ได้รับแสงที่มาจากการ Baked Lighting

แสง Rim Light เล็ก ๆ ที่เห็นเป็นผลมาจากการ Directional Light



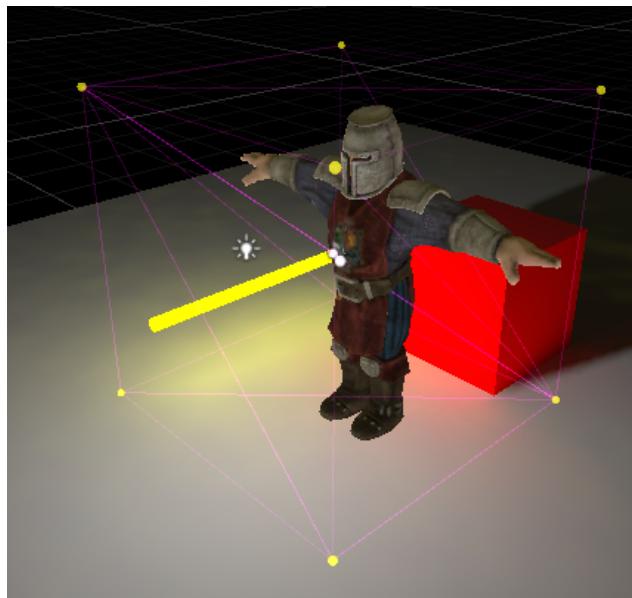
(a) Castle Guard Character

รูปที่ 9.16: Dynamic Character Object Does Not Get Baked Lighting

Light Probes คือตัวช่วยในการกำหนดให้บริเวณในฉากสามมิติ มีการคำนวณค่าความเข้มของแสงไว้ล่วงหน้า และนำมาเก็บไว้ยังจุดต่าง ๆ ในปริมาตรที่ครอบคลุมของ Light Probes

จากนั้นตอน Runtime Light Probes เหล่านี้ จะมีหน้าที่ในการให้แสงกับวัตถุที่เคลื่อนที่ผ่าน เนื่องจากได้มีการคำนวณค่าความเข้มของแสงเก็บไว้ที่จุดเหล่านี้ล่วงหน้าไว้แล้ว

การติดตั้ง Light Probes ทำได้การสร้างจาก Game Object->Light->Light Probes ทั้งนี้การติดตั้ง Light Probes ต้องติดตั้งให้ครอบคลุมพื้นที่ทั้งพื้นที่แสงสว่างและพื้นที่มืด

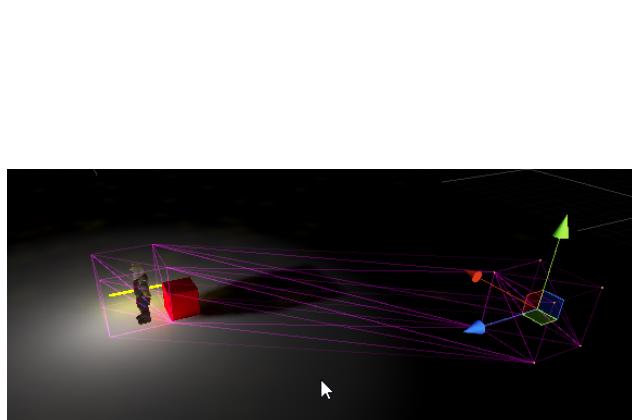


(a) ลักษณะของ Light Probes



(b) No Light Probes in the dark zone of the scene. The character is still lit by the probes.

รูปที่ 9.17: Setting Up Scene Light Probes



(a) Light Probes are also settled up in the dark zone



(b) ตัวละครจะไม่ได้รับแสง ทำให้มองไม่เห็นตัวละครในตำแหน่งที่ไม่มีแสงส่องถึง

รูปที่ 9.18: Setting Up Scene Light Probes in the dark zone

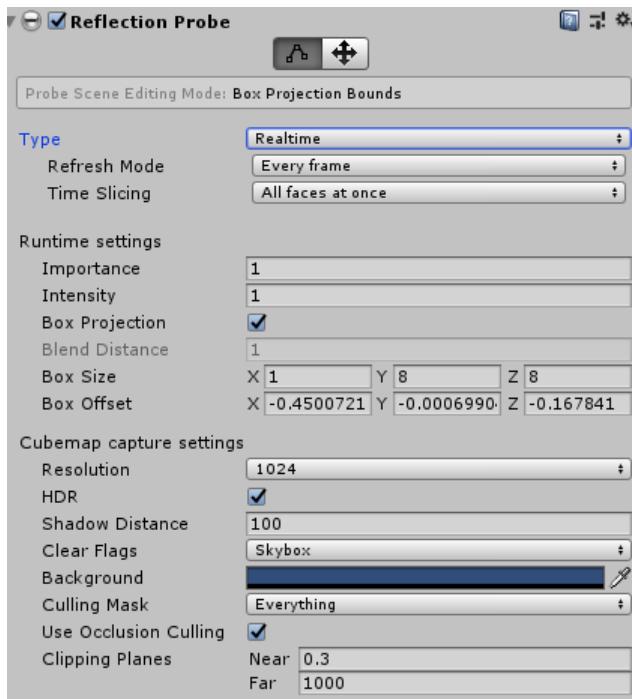
9.7 Reflection Probes

Game Object->Light->Reflection Probes

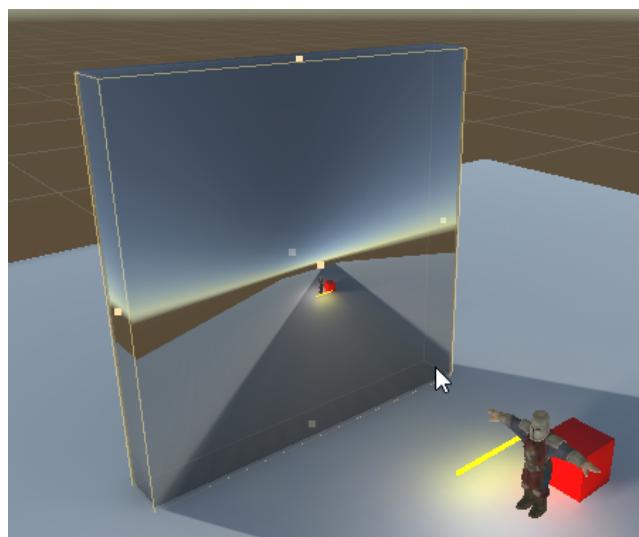
ในการทำงานเดียวกับ Light Probes สำหรับวัตถุที่มีพื้นผิวมันวาว ที่สามารถเห็นภาพสะท้อนบนพื้นผิวได้ สามารถกำหนด

ให้มีกลไกการคำนวณเงาสะท้อนได้ 3 แบบ

- Baked คำนวณเงาสะท้อนของสภาพแวดล้อมลงบนพื้นผิวไว้ล่วงหน้า
- Custom กำหนดวัตถุบางชิ้นที่เป็นวัตถุที่เคลื่อนที่ (Dynamic)
- Realtime กำหนดให้คำนวณเงาสะท้อนตอน runtime



(a) คุณสมบัติของ Reflection Probes

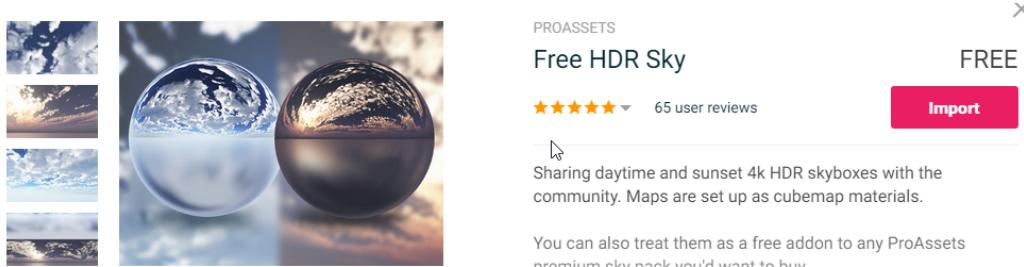


(b) การปรับกรอบของ Reflection Probes

รูปที่ 9.19: Setting Up Scene Light Probes in the dark zone

9.8 ห้องฟ้าจำลอง (Skybox)

เข้า Asset Store และค้นหา “Free HDR Sky” และทำการ Download เข้ามาใช้งานในโครงการ

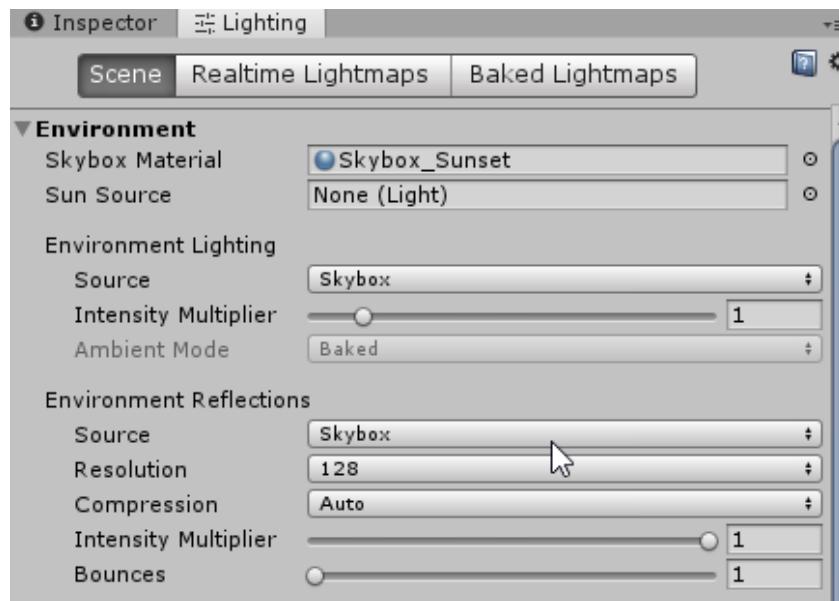


(a)

รูปที่ 9.20: HDR Sky Asset

เปลี่ยน Environment->Skybox Material จาก Default-Skybox เป็น Skybox_Sunset หรือ Skybox_Daytime หรือ Skybox อื่น ๆ ตามที่ผู้พัฒนาต้องการ

โดยที่ดำเนินการอยู่ที่ Assets->Skybox->Materials



(a)

รูปที่ 9.21: Lighting Environment

9.9 พระอาทิตย์จำลอง (Artificial Sun)

แหล่งกำเนิดแสงประเภท Directional Light สามารถสร้างเป็นแหล่งกำเนิดแสงพระอาทิตย์จำลองได้

ให้เปลี่ยนชื่อ Directional Light เป็นชื่อ Sun

เขียนสคริปต์ Sourcecode 9.1 และติดตั้งให้กับ Sun

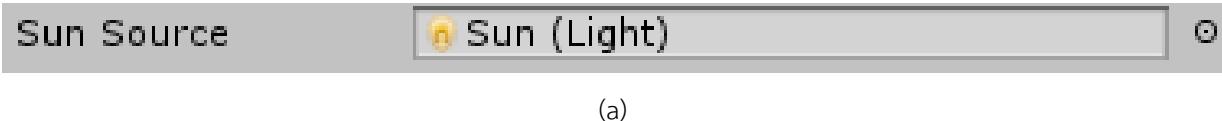
กำหนดค่า Sun Step ให้เหมาะสมตามที่ต้องการ เพื่อควบคุมการเคลื่อนที่ของดวงพระอาทิตย์จำลอง

Source code 9.1: SunControl.cs

```

1  using UnityEngine;
2
3  public class SunControl : MonoBehaviour
4  {
5      [SerializeField] private float sunRotationDegreePerSecond = 30.0f;
6
7      private Light sun;
8
9      // Start is called before the first frame update
10     void Start()
11     {
12         sun = this.GetComponent<Light>();
13     }
14
15     // Update is called once per frame
16     void Update()
17     {
18         sun.transform.Rotate(new Vector3(0,1,0),this.sunRotationDegreePerSecond*Time.
19             deltaTime);
20     }
}

```



(a)

รูปที่ 9.22: Lighting Setting: Sun Source

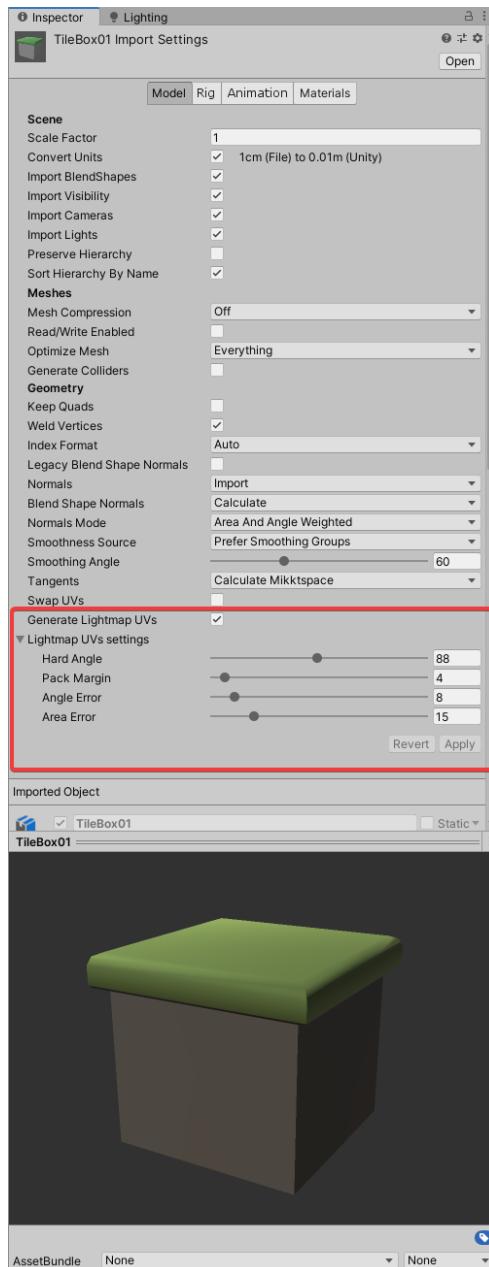
เปิดແນບการตั้งค่าแสงโดยการເຂົ້າມັງ Window->Lighting->Settings

ลາກວັດຖຸ Sun ເຂົ້າສູ່ Sun Source ດັ່ງຮູບທີ 9.22a

9.10 การตั้งค่าวัตถุเพื่อให้สามารถรับ Lightmap จากการอบแสง

9.10.1 ไฟล์โมเดล FBX

วัตถุที่จะสามารถรับแสงจาก Baked Lighting ได้วัตถุเคนน์จะต้องมีการกาง UV และกำหนดให้สร้าง Lightmap UVs ตอนนำเข้าวัตถุ (Import) แสดงดังรูปที่ 9.23

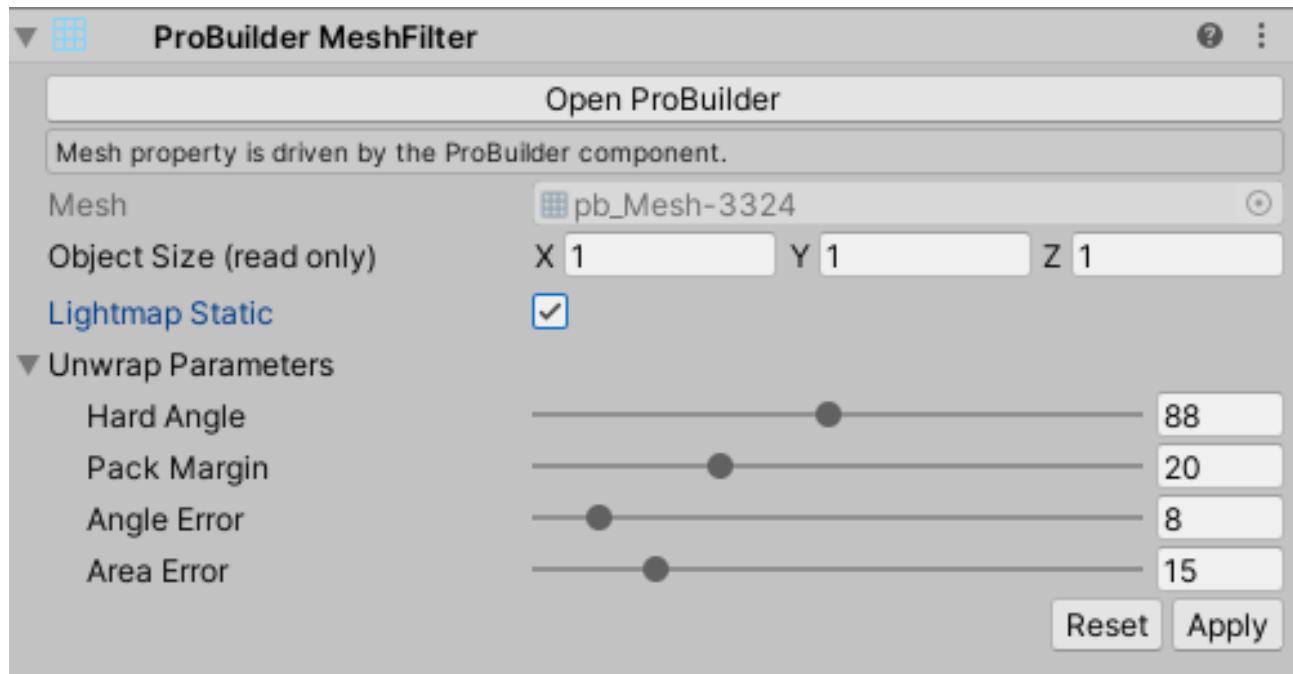


(a) การเปิดใช้งานการสร้าง Lightmap UVs ของโมเดลที่ Import FBX เข้ามาใช้งานใน Unity

รูปที่ 9.23

9.10.2 วัตถุที่สร้างจาก ProBuilder

สำหรับวัตถุที่สร้างจาก ProBuilder ให้เปิดใช้งาน Lightmap Static บนคอมโพเนนท์ ProBuilder MeshFilter ดังรูปที่ 9.24



รูปที่ 9.24: To Enable Probuilder Lightmap Static

9.11 Emissive Texture

การสร้าง Emission Texture เพื่อให้ วัตถุเปล่งแสงตามลวดลาย Texture

https://www.camt.cmu.ac.th/th/logo/camt_horizontal.psd

See lecture demonstration...

9.12 Post-Processing

Post processing คือกระบวนการที่กระทำในขั้นตอนท้ายสุดใน Rendering Pipeline เพื่อประมวลผลปรับแต่งภาพ เอฟเฟคต์ต่าง ๆ ที่จะօอกมาให้สวยงามในขั้นตอนสุดท้ายก่อนที่ภาพจะถูกถ่ายทอดให้เห็นเป็นภาพสองมิติบนหน้าจออนิเตอร์

เอฟเฟคต์ต่าง ๆ ที่มีใน Unity Standard Shader ใน Post Processing

- Bloom

เอฟเฟคต์ที่ทำให้เกิดความพลุ่มว้าวเมื่อมองลงแสงที่กระจายออกโดยรอบแหล่งกำเนิดแสง ตัวอย่างดังรูปที่ 9.25

- Ambient Occlusion

เอฟเฟคต์ที่ทำให้เกิดเงาบริเวณมุมอับ ณ ตำแหน่งต่าง ๆ ที่แสง Ambient ส่องไม่ถึง ตัวอย่างดังรูปที่ [9.26](#)

- Auto Exposure

เอฟเฟคต์ที่จะทำการปรับความสว่างของภาพในขั้นตอนสุดท้ายโดยอัตโนมัติ คล้ายกับการใช้งานกล้องถ่ายรูป มือถือที่จะมีการปรับความสว่างของภาพให้ออกมาพอดีโดยอัตโนมัติ

- Colour Grading

เป็นการปรับโทนสีของภาพให้ได้โทนตามที่ต้องการในขั้นตอนสุดท้าย ตัวอย่างดังรูปที่ [9.30](#)

- Depth of Field

กำหนดความชัดลึกของภาพ ตัวอย่างดังรูปที่ [9.28](#)

- Screen Space Reflection

กำหนดให้มีการเอฟเฟคต์การสะท้อนแสงบนพื้นผิวที่มีความมันวาว โดยคำนวณจากมุมมองของกล้องเท่านั้น ตัวอย่างดังรูปที่ [9.31](#)

- Grain

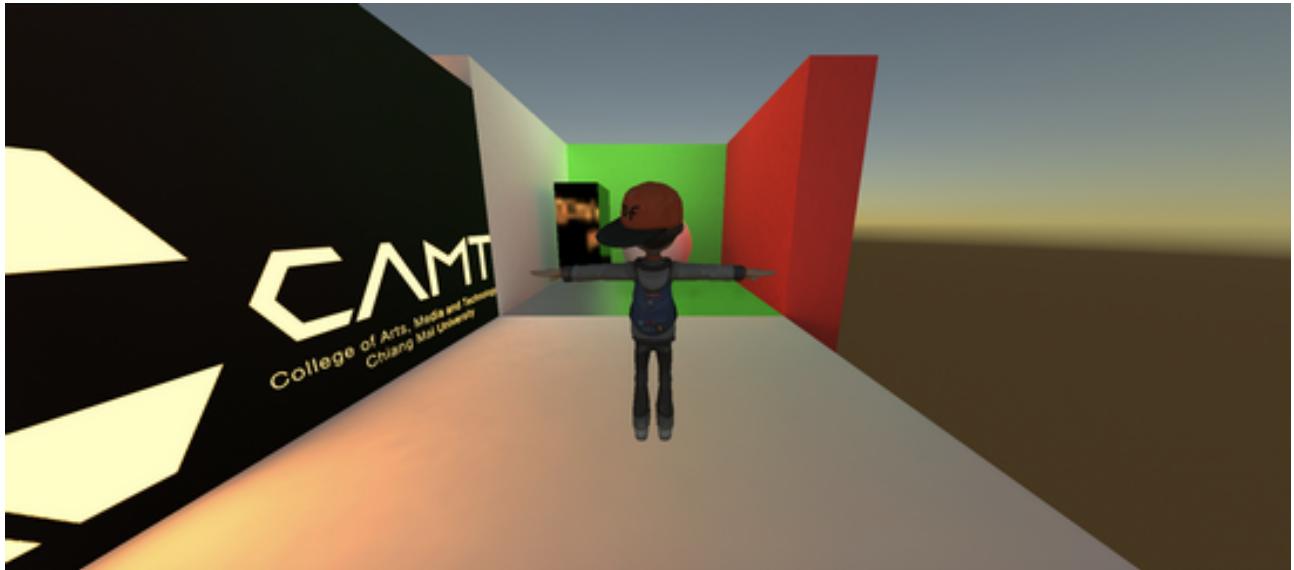
เอฟเฟคต์ที่จะทำให้เกิดการจำลองภาพมีสัญญาณรบกวนมีลักษณะเป็นอนุภาคเม็ดเล็ก ๆ กระจายทั่วภาพ ตัวอย่างดังรูปที่ [9.29](#)

- Chromatic Aberration

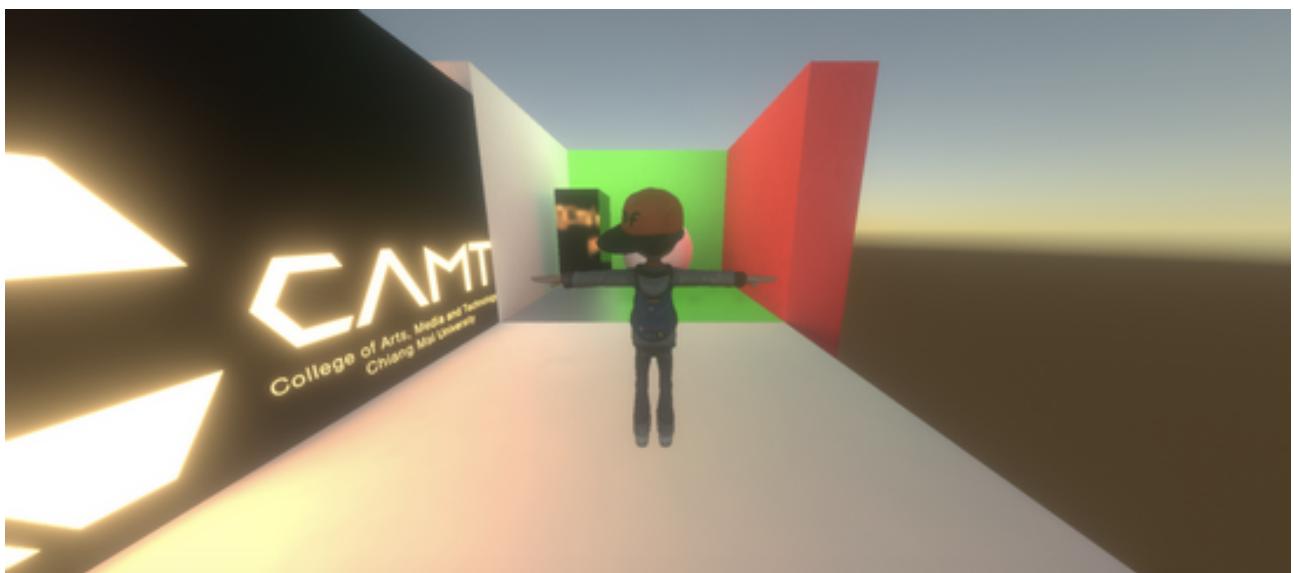
เอฟเฟคต์ที่ทำให้เกิดการจำลองการเกิดขอบม่วงที่เกิดจากการตกรอบของแม่สีของแสงไม่ตรงตำแหน่งเดียวกัน โดยเอฟเฟคต์นี้จะเกิดบริเวณที่มี Contrast สูง ตัวอย่างดังรูปที่ [9.27](#)

การพัฒนาเกมด้วยเกมเอนจิน Unity (Game Development using Unity)

บทที่ 9. การจัดแสงในฉาก แสงทางอ้อม และเอฟเฟคต์หลังกระบวนการเรนเดอร์ (SCENE LIGHTING, GLOBAL ILLUMINATION, AND POST PROCESSING EFFECTS)

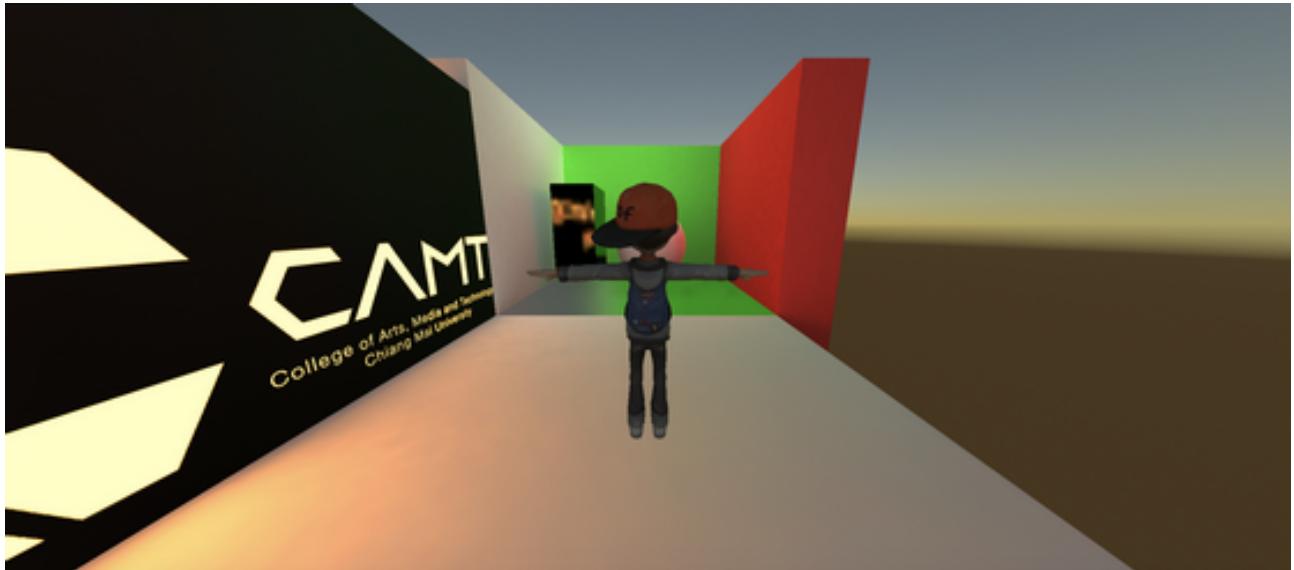


(a) ไม่ใช้ Bloom

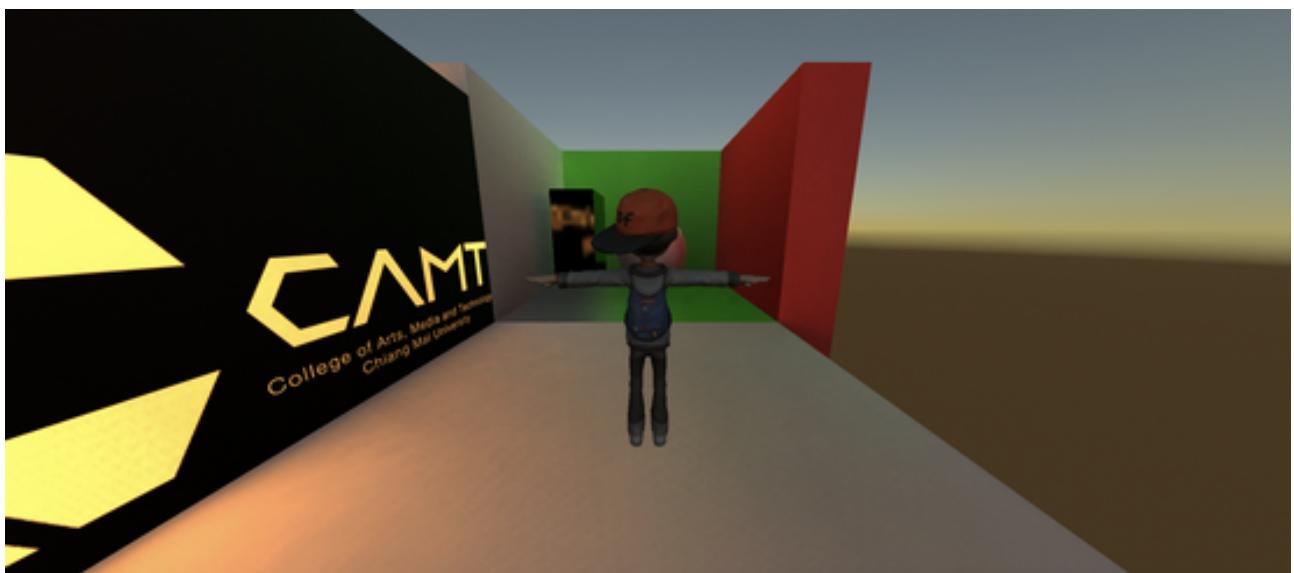


(b) ใช้ Bloom intensity = 2

รูปที่ 9.25: Post-Processing: Bloom Effect

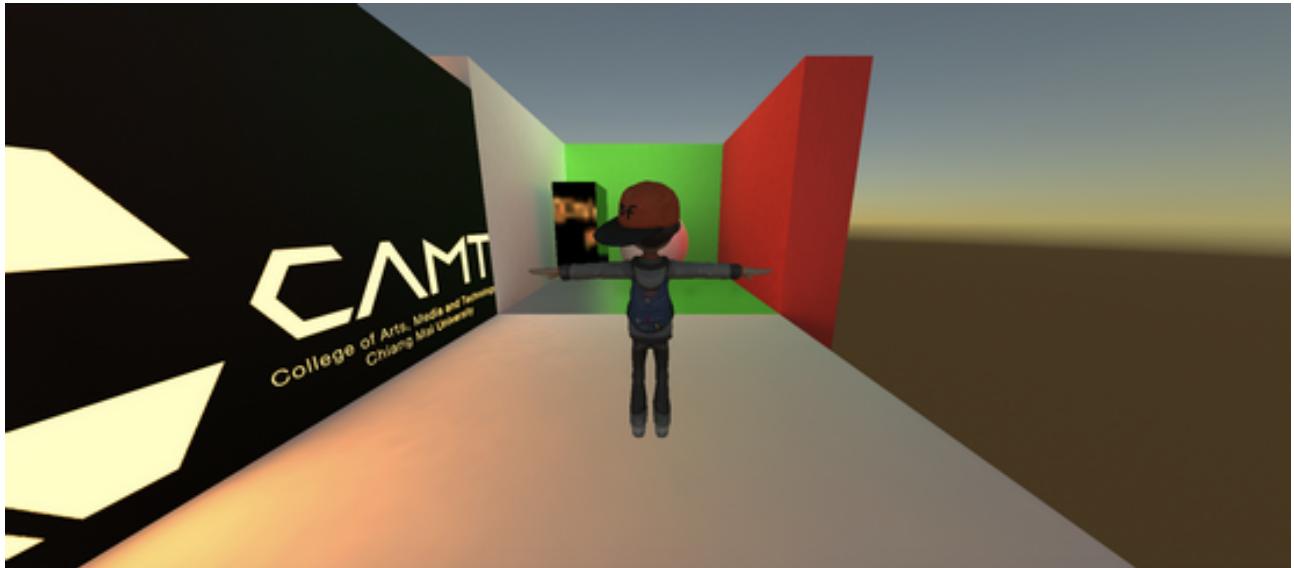


(a) ไม่ใช้ Ambient Occlusion

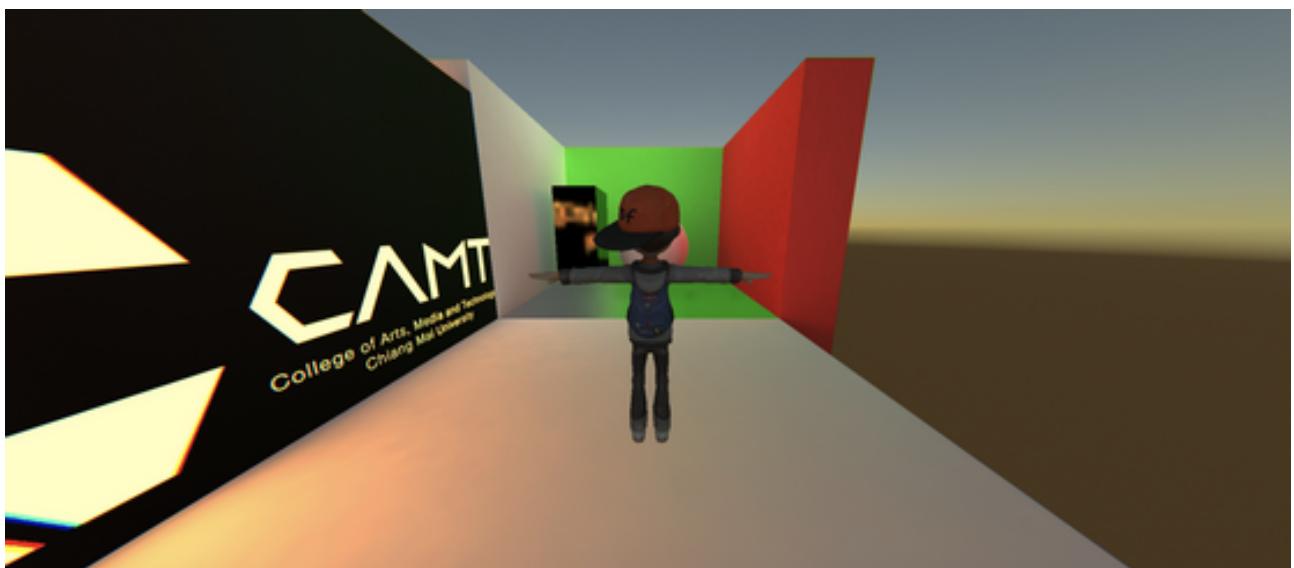


(b) ใช้ Ambient Occlusion intensity = 1

รูปที่ 9.26: Post-Processing: Ambient Occlusion Effect

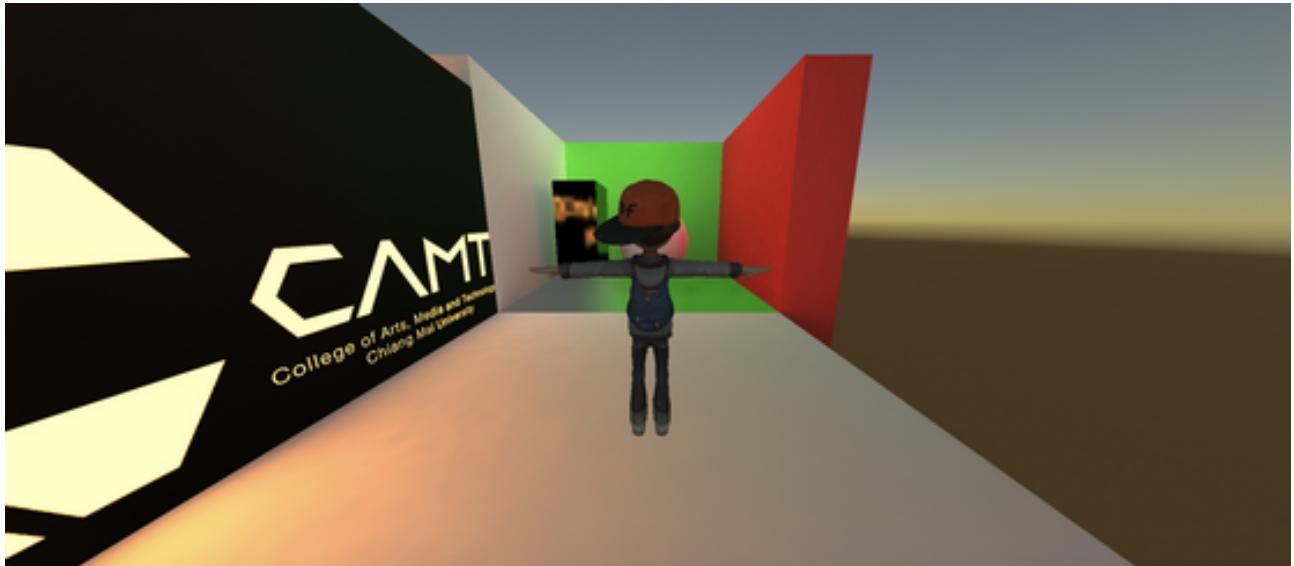


(a) ไม่ใช้ Chromatic Aberration

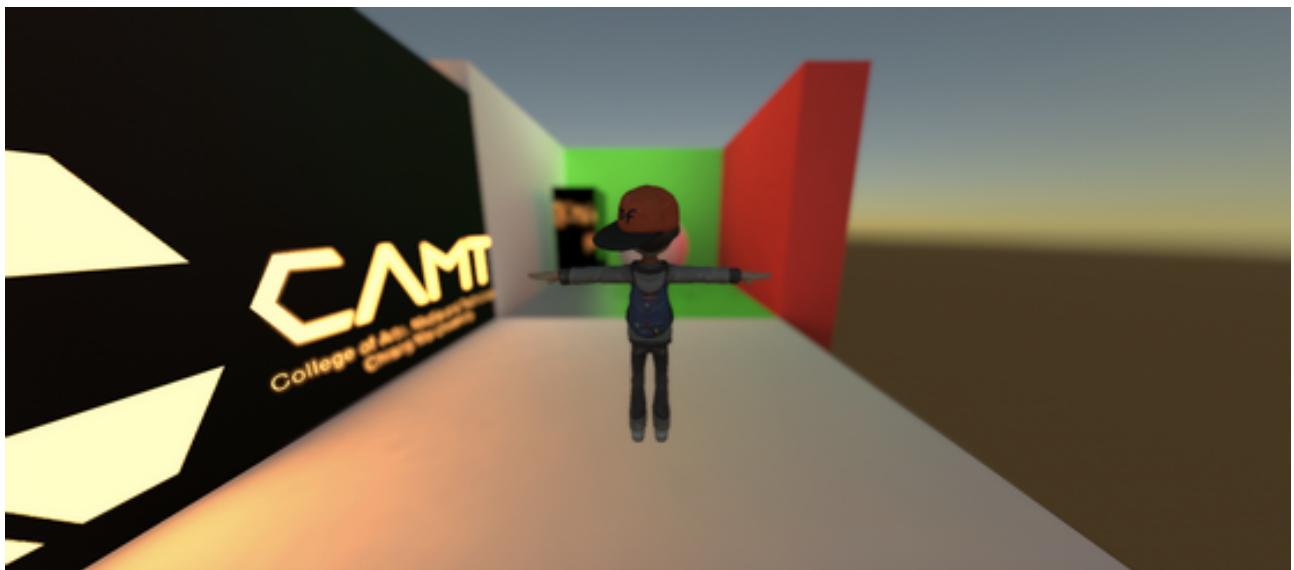


(b) ใช้ Chromatic Aberration

รูปที่ 9.27: Post-Processing: Chromatic Aberration Effect

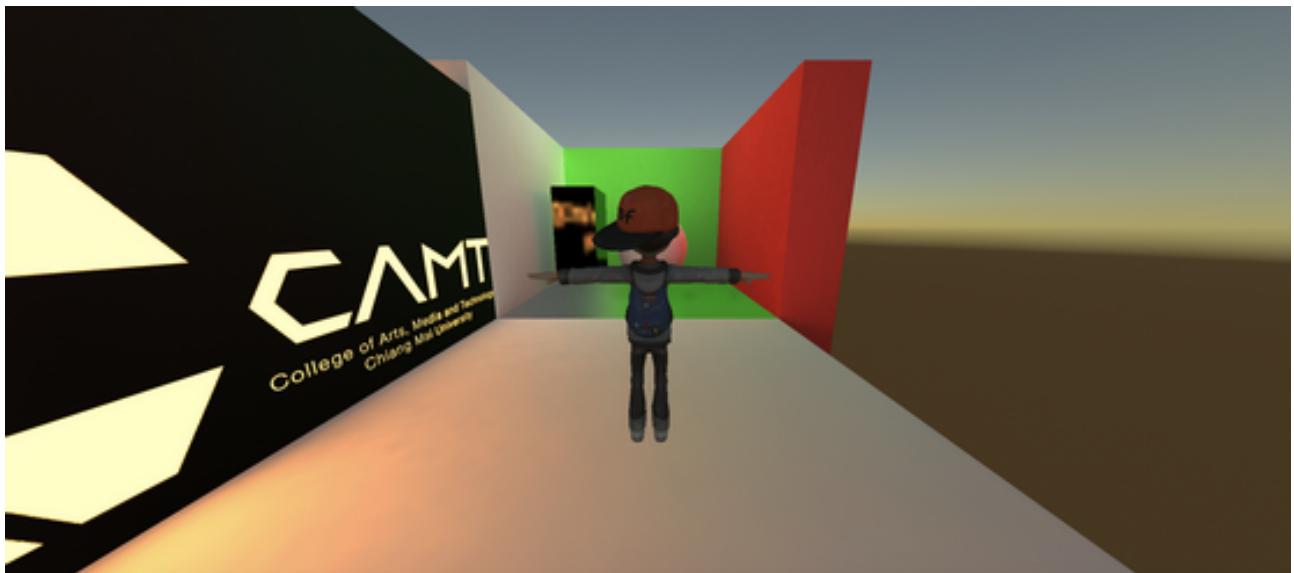


(a) ไม่ใช้ Depth of Field

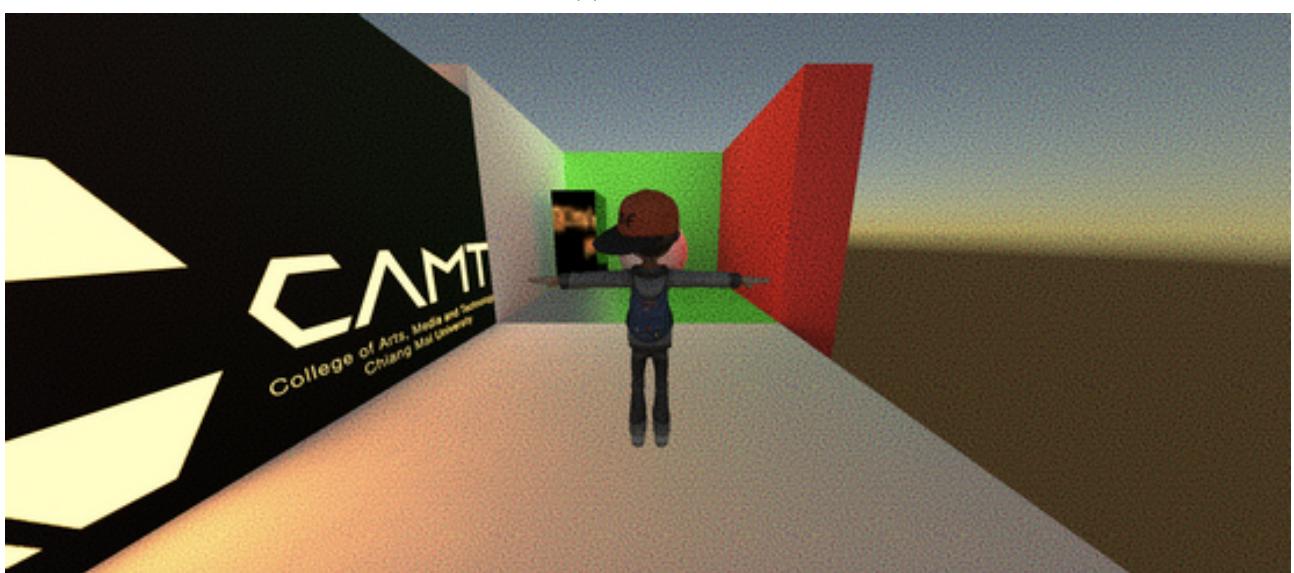


(b) ใช้ Depth of Field

รูปที่ 9.28: Post-Processing: Depth of Field Effect

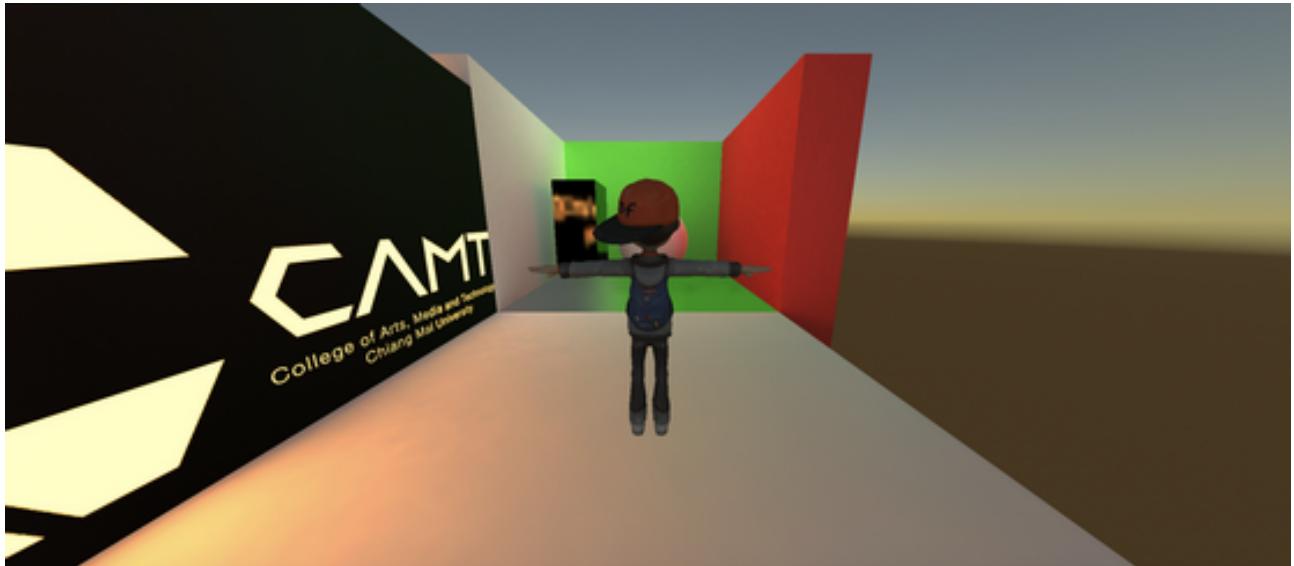


(a) ไม่มี Grains

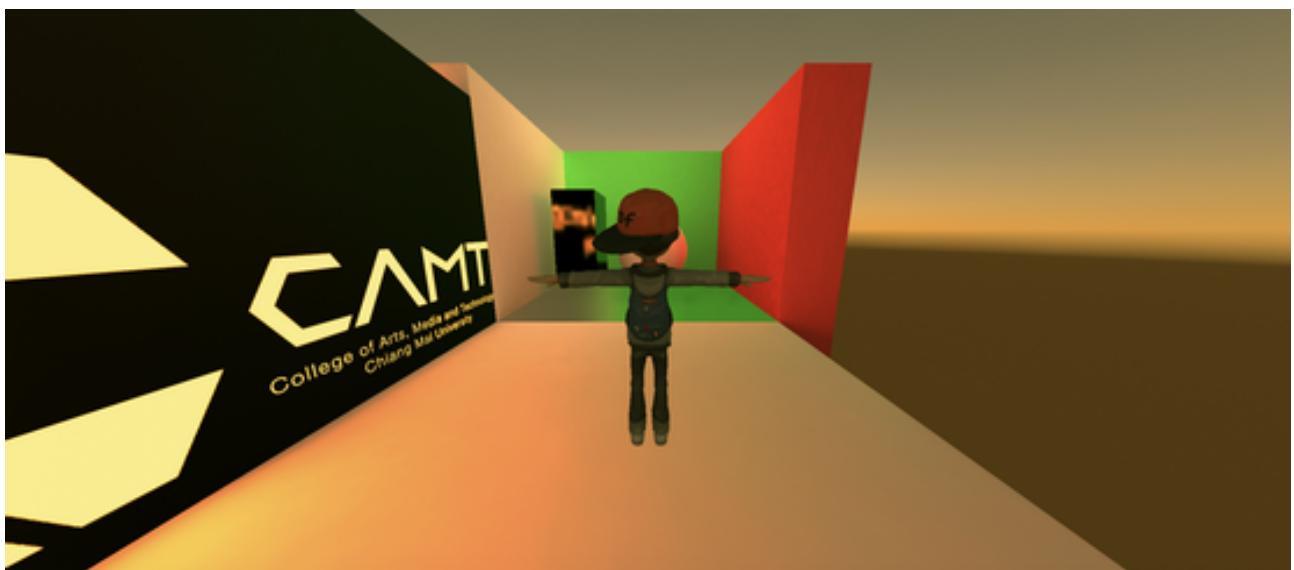


(b) มี Grains

รูปที่ 9.29: Post-Processing: Grain Effect

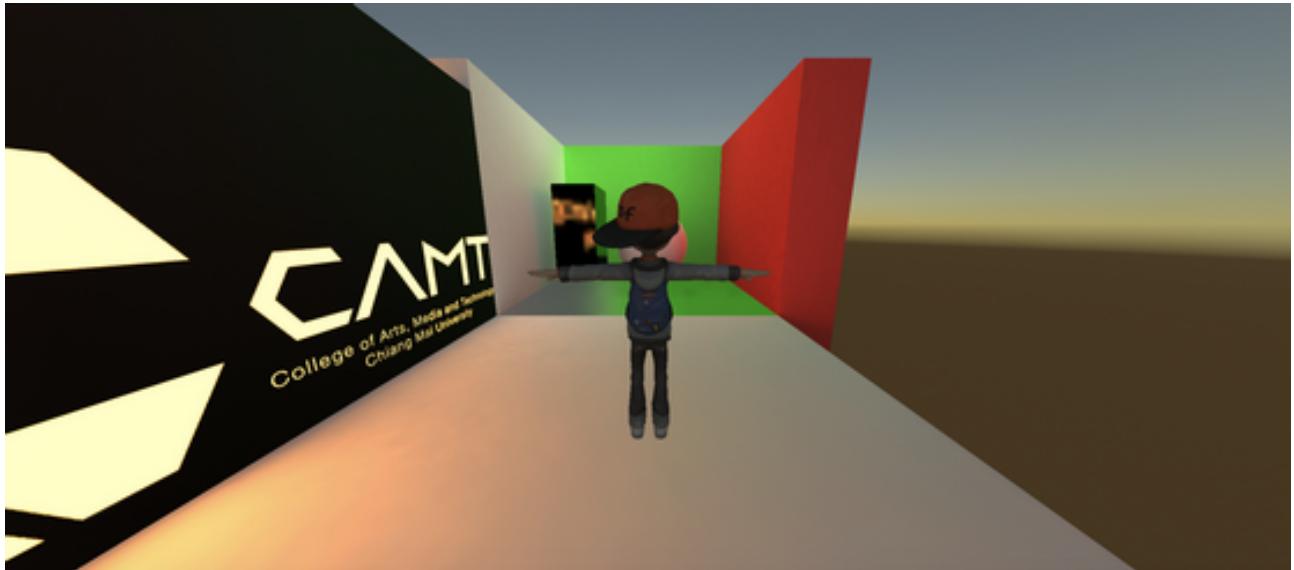


(a) ไม่ใช้ Colour Grading

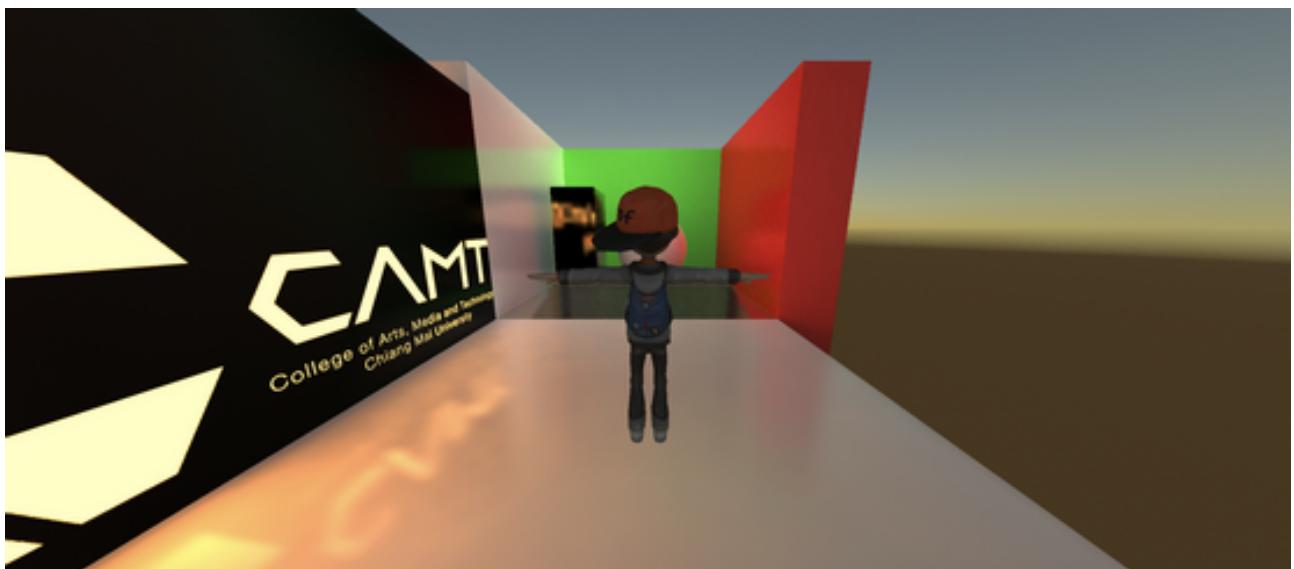


(b) ใช้ Colour Grading โดยปรับโทนภาพอุ่น

รูปที่ 9.30: Post-Processing: Colour Grading Effect



(a) ไม่ใช้ Screen Space Reflection



(b) ใช้ Screen Space Reflection

รูปที่ 9.31: Post-Processing: Screen Space Reflection Effect

9.12.1 การตั้งค่าเพื่อใช้งาน Post-Processing

1. ติดตั้งแพคเกจ Post Processing ดังรูปที่ [9.32](#)
2. สร้างวัตถุเกมที่มีคอมโพเนนท์ Post-process Volume จากเมนู GameObject->3D Object->Post-process Volume ดังรูปที่ [9.33](#)
3. คลิกเปิดใช้งาน Is Global บนคอมโพเนนท์ Post-process Volume และคลิก New เพื่อสร้าง Post Process

Profile ดังรูปที่ 9.34 เพื่อสร้างแอสเซ็ต Post Process Profile ดังรูปที่ 9.35

4. คลิกเลือก Main Camera และทำการเพิ่มคอมโพเนนท์ Post-process Layer ดังรูปที่ 9.36

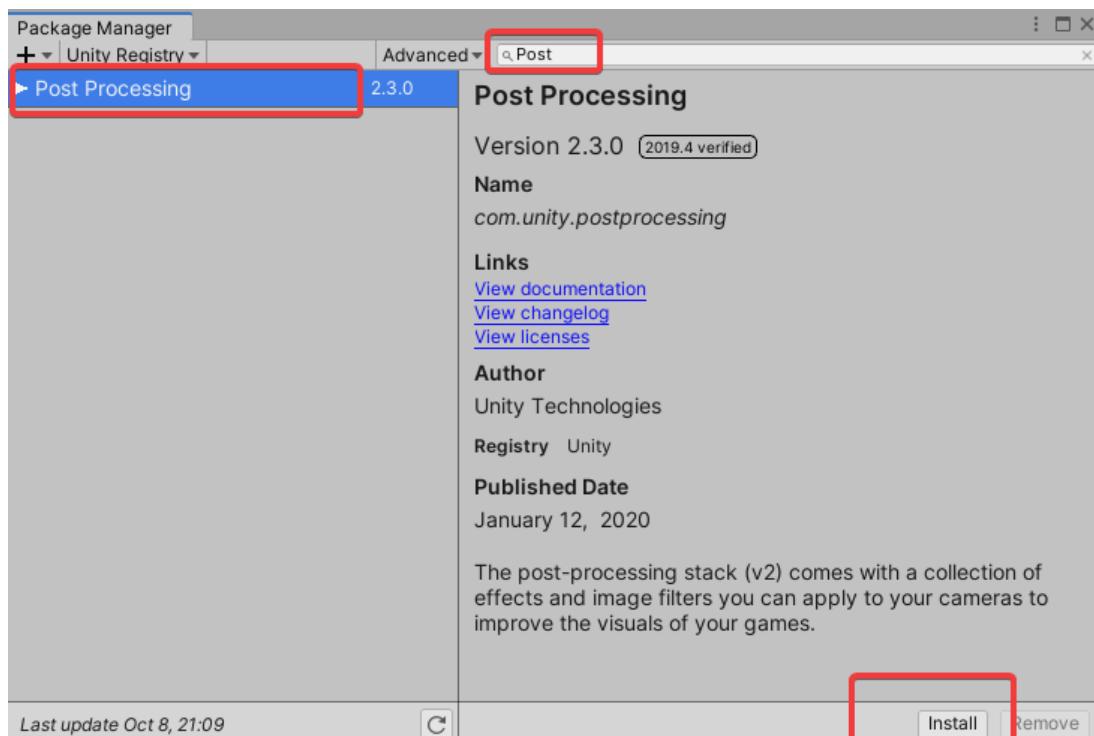
5. เพิ่ม “Post Processing” layer ดังรูปที่ 9.37

6. คลิกเลือก “Post Processing” Layer บนคอมโพเนนท์ Post-process Layer บน Main Camera ดังรูปที่ 9.38

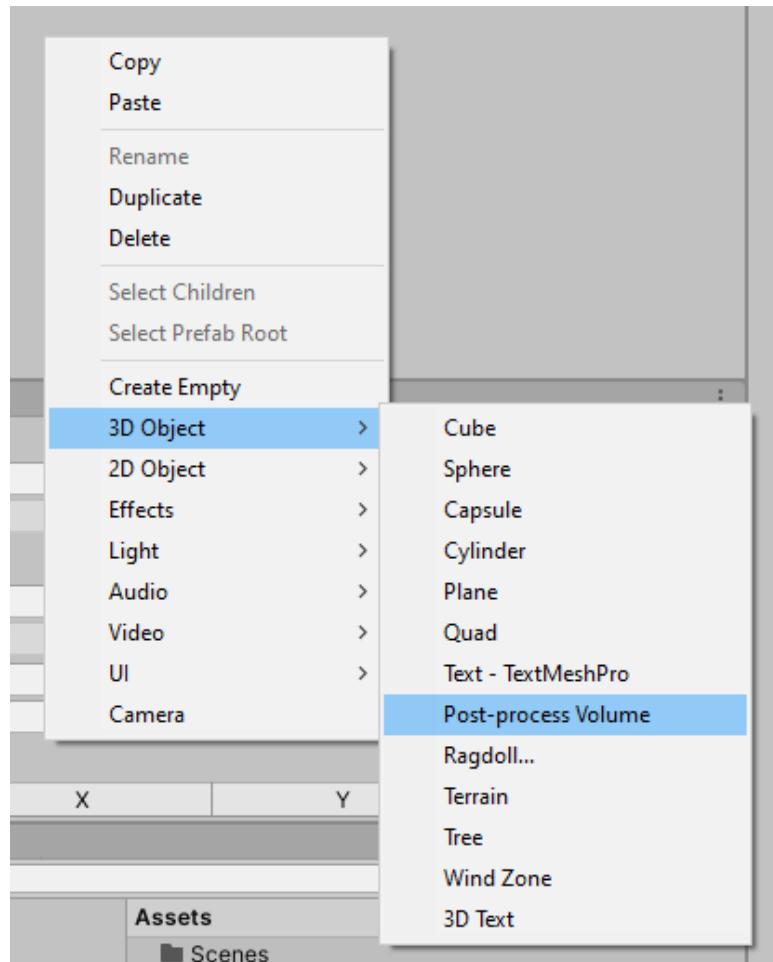
7. คลิกเลือกวัตถุเกม Post-process Volume และเปลี่ยน Layer เป็น Post Processing จากนั้นทดสอบการทำงานของเอฟเฟคต์ด้วยการ Add effect... Grain และเลือกเปิดใช้งาน เลือก All จากนั้นทดลองปรับค่าต่าง ๆ ของเอฟเฟคต์ ดังรูปที่ 9.39

8. จากนี้สามารถเพิ่ม Post Processing Effect ได้ตามต้องการตามเอฟเฟคต์ต่าง ๆ ที่อธิบายไปในหัวข้อ 9.12

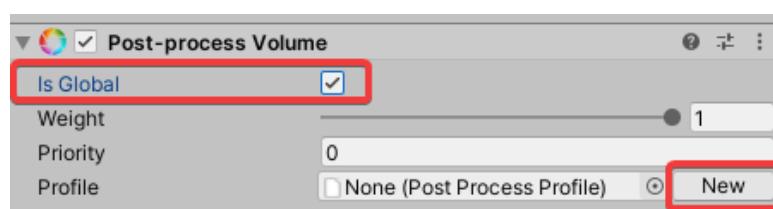
Post-Processing



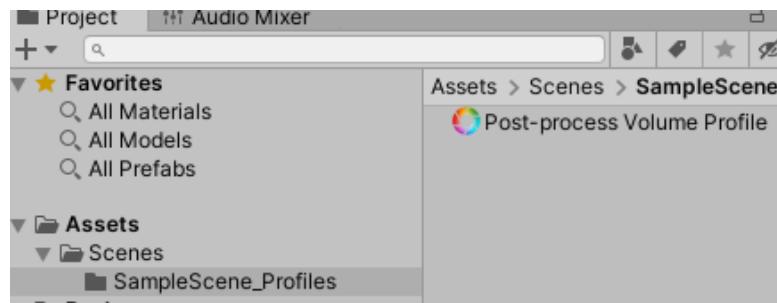
รูปที่ 9.32: Post Processing Package Installation



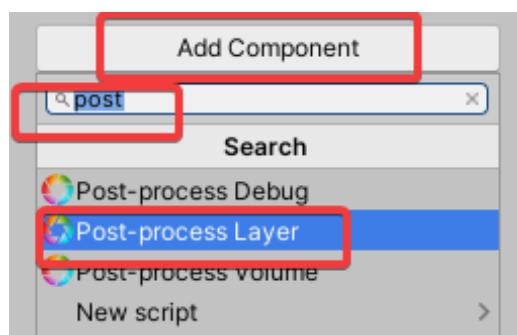
รูปที่ 9.33: Post Processing Volume Game Object



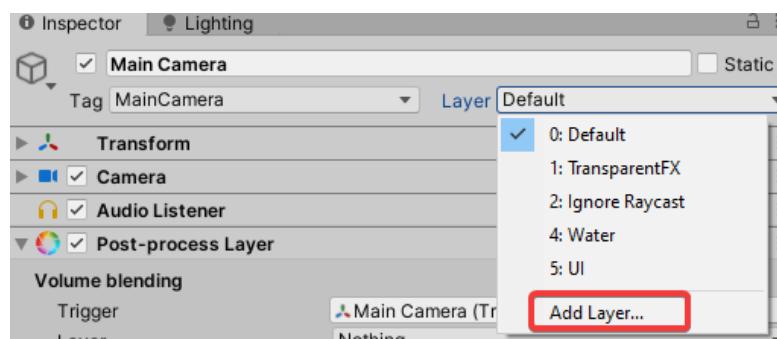
รูปที่ 9.34: Post Processing Volume Component Settings



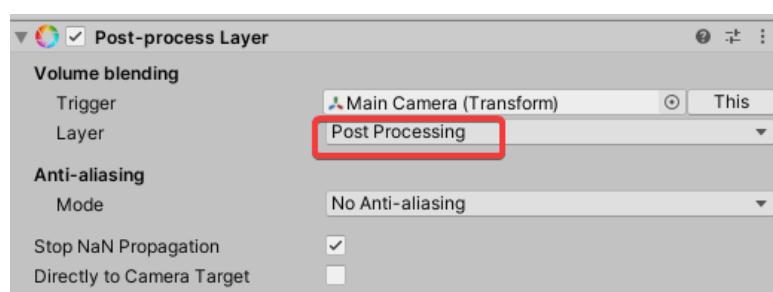
รูปที่ 9.35: Post Process Profile Stack



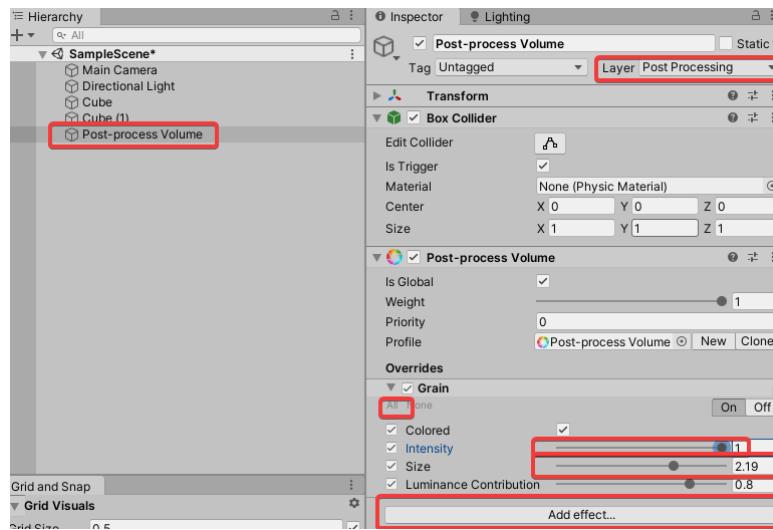
รูปที่ 9.36: Post Process Layer Component on Main Camera



รูปที่ 9.37: Add “Post Process” layer



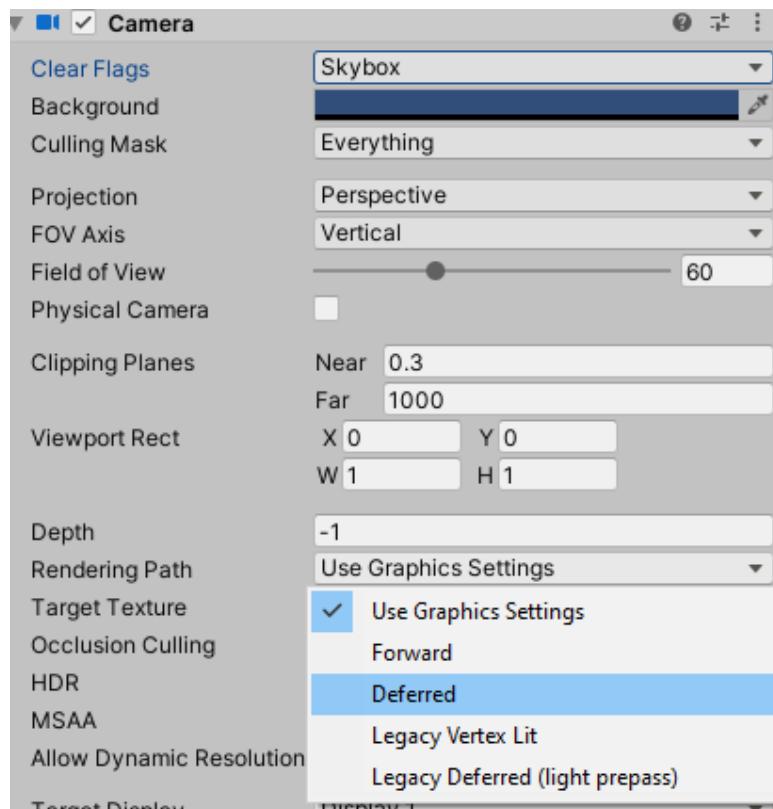
รูปที่ 9.38: Set “Post Process” layer on Post-process Layer component on the Main Camera



รูปที่ 9.39: Change Post-process Volume Game Object's layer and test

Screen Space Reflection - SSR

เอฟเฟคต์เงาสะท้อนแบบ Screen Space หรือ Screen Space Reflection จะทำได้ก็ต่อเมื่อทำการตั้งค่า Rendering Path บน Main Camera ดังรูปที่ 9.40



รูปที่ 9.40: Main Camera Rendering Path Setting to Deferred

9.13 การแก้ปัญหาต่าง ๆ ที่อาจจะเกิดขึ้น (Troubleshooting)

See lecture demonstration...

- GPU lightmapper problems...switch forth and back
- Light probe sometimes does not work as intended...restart unity
- GPU lightmapper is still in preview mode,...consider switching to Unity 2020.1 ..being developed

คำถ้ามสำหรับ chapter 9 การจัดแสงในฉาก แสงทางอ้อม และเอฟเฟคต์หลังกระบวนการเรนเดอร์ (Scene Lighting, Global Illumination, and Post Processing Effects)

ออกแบบสร้างจากที่มีการใช้งานการจัดแสงทุก ๆ แบบ โดยให้มีทั้งส่วน Outdoor และ Indoor

กำหนดให้นำตัวละครลงไปเดินลงบนฉากได้ด้วย โดยใช้ Module Camera ได้ ๆ ก็ได้อาที FreeLookCameraRig MultipurposeCameraRig หรือกล้องจาก Package Cinemachine

อัดวิดีโอประกอบผลงานพร้อมทั้งระบุชื่อไฟล์วิดีโอ

.....
.....
.....
.....
.....

1/2563

ชื่อ.....

รหัสนักศึกษา.....

ตอนที่.....

สาขาวิชาเอนนิเมชันและเกม วิทยาลัยศิลปะ สื่อ และเทคโนโลยี มหาวิทยาลัยเชียงใหม่



9.14 คำถามและปัญหาชนิด (Questions and Problems)

คำถามสำหรับ chapter 9 การจัดแสงในฉาก แสงทางอ้อม และเอฟเฟคต์หลังกระบวนการเรนเดอร์ (Scene Lighting, Global Illumination, and Post Processing Effects)

ออกแบบสร้างจากที่มีการใช้งานการจัดแสงทุก ๆ แบบ โดยให้มีทั้งส่วน Outdoor และ Indoor

กำหนดให้นำตัวละครลงไปเดินลงบนฉากได้ด้วย โดยใช้ Module Camera ได ๆ ก็ได้อาที FreeLookCameraRig MultipurposeCameraRig หรือกล้องจาก Package Cinemachine

อัดวิดีโอประกอบผลงานพร้อมทั้งระบุชื่อไฟล์วิดีโอ

.....
.....
.....
.....
.....
.....