1/2563

ชื่อ นส.ปาริชาติ คีรีแก้ว

รหัสนักศึกษา 612110054

ตอนที่

สาขาวิชาแอนนิเมชันและเกม วิทยาลัยศิลปะ สื่อ และเทคโนโลยี มหาวิทยาลัยเชียงใหม่

เนื้อหา (Contents)

บทที่ 1

พื้นฐานแนวคิดและระบบต่าง ๆ ของเกมเอนจิน Unity (Basic Concepts of Unity Game Engine Systems)

วัตถุประสงค์ (Objectives)

1. เพื่อศึกษาระบบคอมโพเนนท์ และการใช้งานสคริปต์ใน Unity
2. เขียนคอมโพเนนท์สคริปต์เพื่อกำหนดคุณสมบัติและพฤติกรรม ให้กับวัตถุเกม
3. ศึกษาลำดับการทำงานของเมธอดที่สำคัญของการเขียนสคริปต์ใน Unity

1.1 บทนำ

บทนี้ศึกษาแนวคิด กระบวนการทำงานของวงจรเกมลูป (Gameloop Cycle) และเมธอดต่าง ๆ ที่จะถูกเรียกเมื่อเกิดเหตุการณ์ต่าง ๆ ขึ้นระหว่างที่โปรแกรมเกมกำลังทำงานอยู่

นอกจากนี้ยังรวมถึงระบบของการพัฒนาที่เป็นหัวใจหลักสำคัญของเกมเอนจิน Unity นั่นคือระบบคอมโพเนนท์ที่เปรียบเสมือนการติดตั้งและถอดความสามารถให้กับวัตถุต่าง ๆ ภายในเกม

1.2 พื้นฐานของเกมเอนจิน Unity และระบบคอมโพเนนท์

วัตถุต่าง ๆ ที่ถูกวางเข้าไปอยู่ภายในฉาก (Scene) ของเกมเรียกว่า GameObject หรือวัตถุเกม ซึ่งมีคอมโพเนนท์พื้นฐานที่สำคัญคือคอมโพเนนท์การจัดวาง หรือ Transform เป็นคอมโพเนนท์พื้นฐานที่สุดที่ทุก ๆ วัตถุเกมภายในฉากจำเป็นต้องมี และคอมโพเนนท์นี้ไม่สามารถถูกลบ ปิดการใช้งาน หรือถอดออกได้

คอมโพเนนท์เปรียบเสมือนความสามารถ คุณสมบัติ หรือพฤติกรรม ที่ถูกติดตั้งให้กับวัตถุเกมเพื่อให้วัตถุเกมนั้น ๆ มีความสามารถที่แตกต่างกัน มีคุณสมบัติที่เฉพาะเจาะจงสำหรับงาน หรือมีพฤติกรรมที่ผู้พัฒนาต้องการให้เป็น

โดยทั่วไปคอมโพเนนท์สามารถถูกติดตั้ง ปิดการใช้งาน หรือถอดออกจากวัตถุเกมได้ ยกเว้นคอมโพเนนท์พิเศษบางประเภทที่กล่าวไปข้างต้นที่เป็นคอมโพเนนท์พื้นฐานไม่สามารถถอดออกได้

ลักษณะของระบบคอมโพเนนท์อาจเปรียบเทียบกับชีวิตประจำวันของมนุษย์ได้เช่น วันนี้ไปเรียนหนังสือสะพายกระเป๋าเป้ และพกโทรศัพท์มือถือไปด้วย ทำให้วันนี้นักศึกษามีคอมโพเนนท์กระเป๋าที่สามารถเอาไว้ใส่สัมภาระได้ มีโทรศัพท์เคลื่อนที่ทำให้มีคอมโพเนนท์ที่สามารถติดต่อสื่อการกับผู้อื่นได้ เป็นต้น หากวันใดไม่ได้นำประเป๋าไปด้วจะทำให้วันนั้น ไม่มีคอมโพเนนท์สำหรับใส่สัมภาระ จะเห็นได้ว่าคอมโพเนนท์ดังกล่าวสามารถถูกติดตั้งหรือถอดออกได้ ซึ่งมีผลทำให้วัตถุที่ถูกติดตั้งคอมโพเนนท์นั้น ๆ มีความสามารถเพิ่มเติมตามลักษณะของคอมโพเนนท์ที่ติดตั้งเข้าไป

1.3 ระบบคอมโพเนนท์ (Component-Based System)

เกมเอนจิน Unity ใช้แนวคิดการพัฒนาระบบในแบบคอมโพเนนท์เป็นหลักในการทำงานของระบบต่าง ๆ ในการพัฒนาเกม อย่างไรก็ตาม ผู้พัฒนายังคงสามารถใช้ความรู้พื้นฐานด้านการเขียนโปรแกรมเชิงวัตถุมาประยุกต์ใช้ร่วมการแนวคิดแบบคอมโพเนนท์ได้

แนวคิดการพัฒนาระบบแบบคอมโพเนนท์และระบบแบบการเขียนโปรแกรมเชิงวัตถุมีทั้งข้อดีและข้อเสียที่แตกต่างกัน วิธีที่ดีที่สุดคือการประยุกต์ใช้และผสมผสานเอาข้อดีของทั้งสองระบบมาใช้ร่วมกัน ซึ่งสามารถกระทำได้ในระบบสคริปต์ของเกมเอนจิน Unity

1.3.1 Anatomy of the Primitive “Cube”

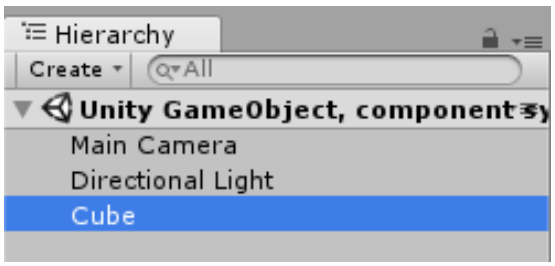
สร้างวัตถุทรงสี่เหลี่ยม Cube และวางลงในฉาก

คลิกเลือกวัตถุ Cube บนหน้าต่าง Hierarchy ทางด้านซ้าย (รูปที่ 1.1a) และสำรวจคุณสมบัติของ Cube บนหน้าต่าง Inspector ทางด้านขวามือ (รูปที่ 1.1c) มีจำนวนกี่คอมโพเนนท์และมีคอมโพเนนท์ใดบ้าง เขียนลงในช่องว่าง

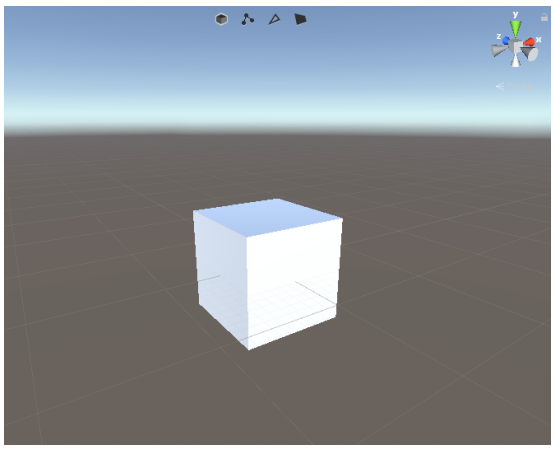
. . . . . .
. . . . . .
. . . . . .

ให้ทดลองลบคอมโพเนนท์ทั้งหมดของวัตถุ Cube โดยการคลิกที่สัญลักษณ์รูปฟันเฟือง และเลือกเมนู Remove Component

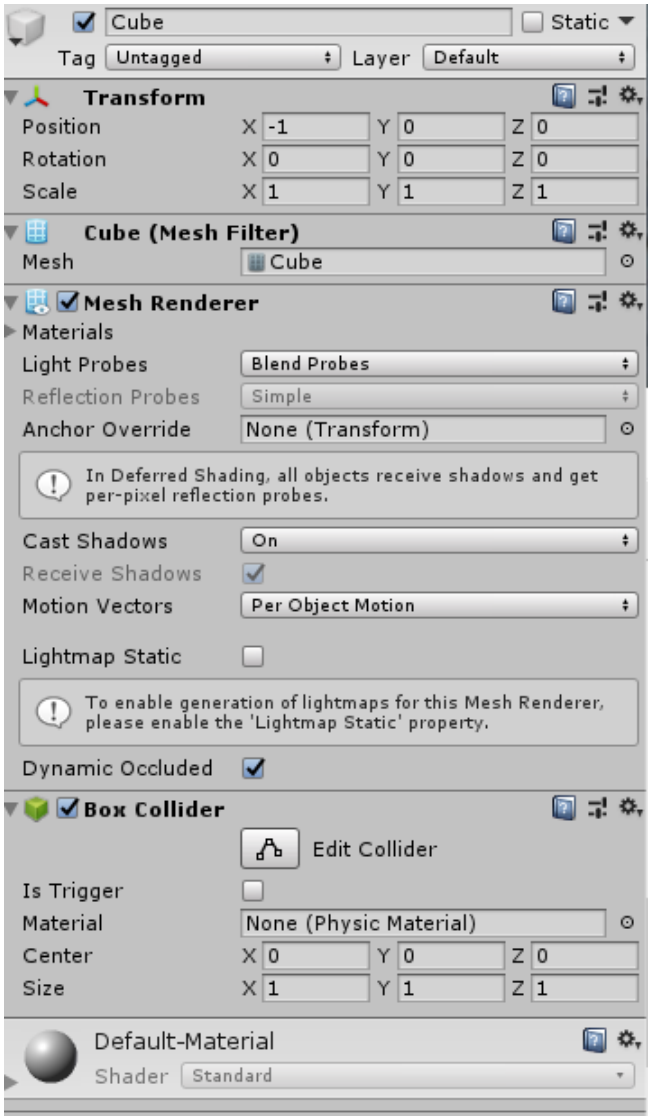
คอมโพเนนท์ใดไม่สามารถลบออกได้ และจงให้เหตุผล



(a) วัตถุเกม Cube ในหน้าต่าง Hierarchy



(b) Cube ใน Scene



(c) หน้าต่าง Inspector

รูปที่ 1.1: Cube ที่สร้างขึ้นและวางลงในฉาก

. . . . . .

อธิบายลักษณะที่ปรากฏของวัตถุ Cube มีลักษณะอย่างไร หลังจากลบคอมโพเนนท์ทั้งหมดออกแล้ว

. . . . . .

1.3.2 วัตถุเกมเปล่า (Empty GameObject)

วัตถุเกมเปล่า (Empty GameObject) สามารถสร้างได้จากเมนู GameObject->Create Empty

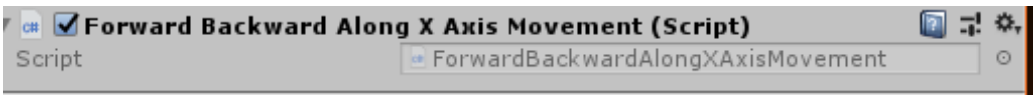
จะสังเกตเห็นว่าเมื่อวัตถุ Cube ถูกลบคอมโพเนนท์ที่มีอยู่เดิมออกทั้งหมด จะทำให้วัตถุ Cube มีลักษณะเช่นเดียวกับวัตถุเกมเปล่าที่ประกอบไปด้วยคอมโพเนนท์พื้นฐานที่สุด คือคอมโพเนนท์การจัดวางของวัตถุในฉาก หรือ Transform นั่นเอง

การจัดวางของวัตถุ (Object Orientation)

คอมโพเนนท์ Transform ประกอบไปด้วยข้อมูลการจัดวางของวัตถุภายในโลกเสมือนจริงที่มีพิกัดคาร์ทีเซียนสามมิติเป็นจุดอ้างอิง ตำแหน่ง X Y และ Z รวมถึงองศาการหมุนของวัตถุรอบพิกัด X Y และ Z ซึ่งเป็นการกำหนดการจัดวางองศาในรูปแบบ Euler Angles แต่ทว่าหากศึกษา Unity ให้ลึกลงไปอีก การแทนการจัดวางการหมุน (Rotation) ของวัตถุใน Unity จะใช้คณิตศาสตร์ Quaternion ซึ่งเป็นการกำจัดข้อเสียของการจัดวางการหมุนด้วย Euler Angles นั่นคือ Gimbal Lock การทำงานภายในเอนจินจะมีการแปลงไปและแปลงกลับระหว่าง Quaternion และ Euler Angles เพื่อให้สะดวกแก่ผู้ใช้งานเนื่องจาก Euler Angles สามารถทำความเข้าใจและเห็นภาพได้มากกว่า Quaternion

วัตถุเกมเปล่าสามารถประยุกต์ใช้ได้ในหลายรูปแบบ อาทิ

1. ใช้เป็นตัววางสคริปต์การทำงานของระบบเกม อาทิ SceneManager SoundManager และ GameStateManager เป็นต้น
2. ใช้เป็น root และนำวัตถุอื่น ๆ มาจัดวางให้เป็น Child ในหน้าต่าง Hierarchy เพื่อจัดระเบียบกลุ่มของวัตถุเกมในฉาก
3. ใช้เปลี่ยนจุด Pivot
4. ใช้เป็นข้อความคั่นเพื่อจัดหมวดหมู่ในหน้าต่าง Hierarchy



รูปที่ 1.2: ผลลัพธ์ของการสร้างและติดตั้งสคริปต์ให้กับ Cube

1.3.3 การใช้งานสคริปต์คอมโพเนนท์เบื้องต้น (Basic Usage of a Script Component)

ผู้พัฒนาสามารถกำหนดการทำงานของวัตถุเกมให้เป็นไปตามที่ผู้พัฒนาต้องการได้ด้วยการเพิ่มสคริปต์คอมโพเนนท์ให้กับวัตถุ และจึงเขียนโปรแกรมเพื่อควบคุมพฤติกรรมของวัตถุนั้นด้วยภาษา C#

ให้ทดลองสร้างวัตถุทรงสี่เหลี่ยม Cube และจัดวางลงในฉาก

คลิกเลือก Cube บนหน้าต่าง Hierarchy หรือบนฉาก จากนั้นคลิกปุ่ม Add Component ที่ด้านล่างสุดของหน้าต่าง Inspector

เลือก New Script และตั้งชื่อสคริปต์ว่า ForwardBackwardAlongXAxisMovement

เลือก Create and Add

ผลลัพธ์คือการเพิ่มคอมโพเนนท์สคริปต์และติดตั้งให้กับทรงสี่เหลี่ยม Cube ดังรูปที่ 1.2

ดับเบิลคลิกที่ชื่อสคริปต์ในคอมโพเนนท์ เพื่อเปิดหน้าต่างแก้ไขซอสโคดสำหรับสคริปต์ ForwardBackwardAlongXAxisMovement.cs หรือสามารถเปิดไฟล์สคริปต์ได้อีกทางจากหน้าต่าง Project->Assets เพื่อแก้ไขซอสโคด จากนั้นจึงทดลองเขียนซอสโคดตาม Sourcecode 1.1

Source code 1.1: ForwardBackwardAlongXAxisMovement.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class ForwardBackwardAlongXAxisMovement : MonoBehaviour {
6      public float MAX_MOVEMENT_DISTANCE = 2.0f;
7
8      float _displacementCounter = 0;
9
10     [SerializeField]
11     private float _xComponentSpeed = 0.02f;
12
13     Vector3 _movementSpeed = Vector3.zero;
14
```

```
15      // Use this for initialization
16    void Start () {
17          _movementSpeed.x = _xComponentSpeed;
18      }
19
20    // Update is called once per frame
21    void Update () {
22          this.transform.position += _movementSpeed;
23
24          _displacementCounter += _movementSpeed.x;
25
26          if (Mathf.Abs(_displacementCounter) > MAX_MOVEMENT_DISTANCE) {
27              _displacementCounter = 0;
28              _movementSpeed *= -1;
29          }
30      }
31  }
```

ทดสอบรันโปรแกรม (ForwardBackwardAlongXAxisMovement) และอธิบายการทำงานของโปรแกรมมาพอสังเขป

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

คอมโพเนนท์ Spin Movement

สร้างสคริปต์ชื่อ SpinMovmement.cs เขียนซอสโคดดัง Sourcecode 1.2 และติดตั้งให้กับ Cube ด้วยการลากสคริปต์ไปวางบนหน้าต่าง Inspector ของ Cube

Source code 1.2: SpinMovement.cs

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class SpinMovement : MonoBehaviour {
6       [SerializeField]
7       private float _angularSpeed = 5.0f;
```

```
8       [SerializeField]
9       private Vector3 _axisOfRotation = new Vector3(1.0f,0,0);
10
11      Transform _objTransform;
12
13    // Use this for initialization
14    void Start () {
15          _objTransform = this.gameObject.GetComponent<Transform>();
16      }
17
18    // Update is called once per frame
19    void Update () {
20          _objTransform.Rotate(_axisOfRotation, _angularSpeed);
21    }
22  }
```

ทดสอบรันโปรแกรม

1.4 การสร้างคอมโพเนนท์ (Creating a Component)

การสร้างคอมโพเนนท์และติดตั้งให้กับวัตถุเกม (GameObject) มีวัตถุประสงค์เพื่อ

- ทำงานตามที่ผู้พัฒนาต้องการ
- มีคุณสมบัติเพิ่มเติมตามที่ผู้พัฒนาต้องการ
- แสดงผลตามที่ผู้พัฒนาต้องการ

ดังตัวอย่างในหัวข้อที่ผ่านมา คอมโพเนนท์ ForwardBackwardAlongXAxisMovement และ SpinMovement เป็นคอมโพเนนท์ที่สร้างขึ้นเพื่อทำให้วัตถุเกมมีพฤติกรรมเคลื่อนที่กลับไป กลับมาในแนวแกน X และหมุนรอบแกน ตามลำดับ

คอมโพเนนท์ ForwardBackwardAlongXAxisMovement

ทำให้วัตถุเคลื่อนที่บนแนวแกน X ด้วยคุณสมบัติที่กำหนดไว้

มีคุณสมบัติสำหรับกำหนดพฤติกรรมการเคลื่อนที่ดังต่อไปนี้

- ขีดจำกัดการเคลื่อนที่ก่อนจะกลับด้านความเร็ว(Max movement distance)

- ความเร็วในแกน X (Speed on X axis)

คอมโพเนนท์ SpinMovement

ทำให้วัตถุหมุนรอบแกนและความเร็วเชิงมุมที่กำหนด

มีคุณสมบัติสำหรับกำหนดพฤติกรรมการหมุนดังต่อไปนี้

- ความเร็วเชิงมุม (Angular Speed)
- แกนการหมุน (Axis of Rotation)

อธิบายถึงระบบคอมโพเนนท์ใน Unity ตามที่นักศึกษาเข้าใจ

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

1.4.1 ส่วนกำหนดสิทธิการเข้าถึงของสมาชิกในคอมโพเนนท์ (Component Member Modifiers)

การกำหนด modifier ของสมาชิกในคอมโพเนนท์สามารถใช้พื้นฐานความรู้เรื่องการเขียนโปรแกรมเชิงวัตถุมากำหนดให้กับสมาชิก เช่น private protected และ public เป็นต้น

หลักการปกป้องข้อมูล (Encapsulation) ในการเขียนโปรแกรมเชิงวัตถุ สามารถนำมาประยุกต์ใช้ได้ในการเขียน C# สคริปต์สำหรับการกำหนดคุณสมบัติ (Variables) และพฤติกรรม (Methods) ให้กับคอมโพเนนท์

โดยใน Unity มีลักษณะพิเศษเพิ่มเติมสำหรับกำหนดให้สมาชิกที่ถูกกำหนด modifier เป็น private สามารถมองเห็นได้ในหน้าต่าง Inspector ของเกมเอนจิน ด้วยการใช้คำสั่ง [SerializeField] ในบรรทัดก่อนหน้าของการประกาศสมาชิกนั้น ทำให้สมาชิกที่เป็น private สามารถถูกกำหนดตั้งค่าผ่านหน้าต่าง Inspector ได้ ซึ่งมีความจำเป็นเนื่องจากเป็นการเปิดโอกาสให้ผู้อื่น ๆ เช่น ออกแบบกราฟฟิก ผู้ออกแบบเกม สามารถใช้ช่องทางนี้สำหรับปรับแต่งค่าให้กับสมาชิกของ

คอมโพเนนท์ได้โดยที่ไม่ต้องเข้าไปแก้ในซอสโคด

1.4.2 โครงสร้างของสคริปต์คอมโพเนนท์ (Structure of a Component)

ทุกคอมโพเนนท์ใน Unity สืบทอดมาจากคลาส MonoBehavior มีกลไกการทำงานพื้นฐานที่สำคัญในการพัฒนาเกม คือเมธอด Start และ Update ซึ่งเป็นเมธอดที่จะถูกเรียกเมื่อฉากถูกเรียกขึ้นมา และฉากของเกมกำลังเล่นอยู่ ตามลำดับ

ข้อมูลเพิ่มเติม https://docs.unity3d.com/ScriptReference/GameObject.html

การเข้าถึงและการค้นหา Component ของวัตถุเกมใด ๆ

การเข้าถึงคอมโพเนนท์ของวัตถุเกมใด ๆ สามารถเข้าถึงได้ด้วยการใช้คำสั่งเมธอด GetComponent<Type>() โดยระบุชนิด (Type) ของคอมโพเนนท์ที่จะเข้าถึงโดยหากวัตถุเกมนั้น ๆ มีคอมโพเนนท์ที่ต้องการถูกติดตั้งไว้อยู่ เมธอดจะคืนค่ากลับมาให้ หากไม่มีคอมโพเนนท์นั้นอยู่ เมธอดจะคืนค่า null กลับมา ดังตัวอย่าง Sourcecode 1.3

Source code 1.3: GetComponent

```
1       GameObject go;
2
3       Type comp1 = go.GetComponent<Type>();
```

1.4.3 ตัวอย่างการประยุกต์ คอมโพเนนท์พลังชีวิต (Example: Health Point Component)

ในส่วนนี้จะยกตัวอย่างการประยุกต์ใช้คอมโพเนนท์สคริปต์เพื่อใช้สำหรับวัตถุในเกมที่ต้องการให้มีคุณสมบัติพลังชีวิต (Health Point) แสดงดัง Sourcecode 1.5

ในกรณีตัวอย่างนี้ได้กำหนดให้ค่าพลังชีวิตมีค่าได้สูงสุดเป็นเลขทศนิยมเท่ากับค่า 100 หน่วย และมีการห่อหุ้มข้อมูล (encapsulation) ตัวแปรที่ใช้เก็บค่าพลังชีวิตโดยกำหนดสิทธิเข้าถึงเป็น private ให้กับตัวแปร _healthPoint (float) และจึงกำหนดช่องทางการเข้าถึงตัวแปรในลักษณะของคุณสมบัติ Property ในทำนองเดียวกันกับเมธอด Get และ Set ในภาษาจาวา โดยรูปแบบการกำหนด Property เป็นดัง Sourcecode 1.4

Source code 1.4: การกำหนด Property ในภาษา C#

```
1   <Access Modifier> <Type> <Property Name>{
2     get{
3       return <Variable>;
4     }
5
```

```
6       set{
7         <Variable> = value;
8       }
9     }
```

Source code 1.5: SimpleHealthPointComponent.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class SimpleHealthPointComponent : MonoBehaviour
6  {
7     [SerializeField]
8      public const float MAX_HP = 100;
9
10    [SerializeField]
11    private float _healthPoint;
12
13    //Property
14    public float HealthPoint{
15        get{
16            return _healthPoint;
17        }
18        set{
19            if(value > 0)
20            {
21                if(value <= MAX_HP)
22                 {
23                     _healthPoint = value;
24                 }else{
25                     _healthPoint = MAX_HP;
26                 }
27            }
28        }
29    }
30
31 }
```

Instructions

- ให้สร้างสคริปต์คอมโพเนนท์ตามซอสโคด Sourcecode 1.5
- ทดลองนำคอมโพเนนท์ดังกล่าวไปติดตั้งให้กับวัตถุใด ๆ ในฉาก

1.4.4 ตัวอย่างการประยุกต์ คอมโพเนนท์กำหนดประเภทของไอเท็มในเกม (Example: Item Type Component)

ในส่วนนี้จะยกตัวอย่างการประยุกต์ใช้คอมโพเนนท์สคริปต์เพื่อใช้ในการแยกแยะชนิดของไอเท็มในเกม โดยการผสมผสานการใช้งานการเขียนโปรแกรมเชิงวัตถุด้วย C# และสคริปต์คอมโพเนนท์ด้วยกันห

สมมติว่าไอเท็มของเกมมีอยู่ 4 ประเภทด้วยกันคือ Coin BigCoin PowerUp PowerDown จะสามารถกำหนดประเภทของตัวแปรแบบ enum ได้ดัง Sourcecode 1.6

จะสังเกตได้ว่าซอสโคด Sourcecode 1.6 ยังไม่ใช่คอมโพเนนท์ใน Unity แต่เป็นการประกาศประเภทข้อมูลแบบ enum ในภาษา C# ให้ชื่อว่า ItemType โดยมีสมาชิกอยู่ 4 ชนิดตามชนิดของไอเท็มนั่นเอง

Source code 1.6: ItemType.cs

```
1 public enum ItemType {
2     COIN,
3     BIGCOIN,
4     POWERUP,
5     POWERDOWN,
6 }
```

จากนั้นจึงออกแบบและเขียนโปรแกรมสคริปต์คอมโพเนนท์เพื่อใช้สำหรับติดตั้งให้กับวัตถุเกมใด ๆ ที่ต้องการให้มีการแยกแยะได้ว่าเป็นไอเท็มชนิดใด ดัง Sourcecode 1.7

Source code 1.7: ItemTypeComponent.cs

```
1 using UnityEngine;
2
3 public class ItemTypeComponent : MonoBehaviour
4 {
5     [SerializeField]
6     protected ItemType _itemType;
7     public ItemType Type
8     {
```

```
9           get
10          {
11              return _itemType;
12          }
13          set
14          {
15              _itemType = value;
16
17          }
18      }
19  }
```

Instructions

- สร้างโครงการ Unity ขึ้นมาใหม่ หรือใช้โครงการเดิม
- เขียนซอสโคด ItemType.cs ตาม Sourcecode 1.6
- เขียนซอสโคด ItemTypeComponent.cs ตาม Sourcecode 1.7
- เพิ่มวัตถุ Cube ลงในฉาก
- ติดตั้งคอมโพเนนท์ ItemTypeComponent ให้กับ Cube ที่สร้างขึ้น
- สังเกตคอมโพเนนท์ที่ถูกติดตั้งบน Cube ดังรูปที่ 1.3
- ทดลองเพิ่มวัตถุเกมประเภทอื่น และติดตั้งสคริปต์คอมโพเนนท์ ItemTypeComponent ให้กับวัตถุนั้น

รูปที่ 1.3: คอมโพเนนท์ ItemTypeComponent บนวัตถุ Cube

1.5 ศึกษาลำดับการทำงานของสคริปต์และเมธอดต่าง ๆ ใน Unity (Study the Order, Life Cycle, and Methods of a Unity Script)

สร้างสคริปต์ ScriptLifeCycleStudy.cs รูปที่ 1 และนำไปติดตั้งเพิ่มคอมโพเนนท์สคริปต์ให้กับวัตถุใด ๆ ก็ได้ในฉาก

Source code 1.8: ScriptLifecycleStudy.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class ScriptLifecycleStudy : MonoBehaviour {
6
7      void Awake() {
8          Debug.Log("Awake() has been called.");
9      }
10
11   // Use this for initialization
12   void Start () {
13         Debug.Log("Start() has been called.");
14     }
15
16   // Update is called once per frame
17   void Update () {
18
19   }
20
21     void OnDisable() {
22         Debug.Log("OnDisable() has been called.");
23     }
24
25     void OnDestroy()
26     {
27         Debug.Log("OnDestroy() has been called.");
28     }
29
30     void OnApplicationPause()
31     {
32         Debug.Log("OnApplicationPause() has been called.");
33     }
34
35     void OnApplicationQuit()
36     {
37         Debug.Log("OnApplicationQuit() has been called.");
38     }
```

```
39
40 }
```

ทดลองรันโปรแกรมและจบการทำงานของโปรแกรม ตรวจสอบหน้าต่าง Console

อธิบายลำดับการทำงานของเมธอดทั้งหมดที่มีในสคริปต์ ScriptLifeCycleStudy.cs

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

1.6 บทสรุปท้ายบท (Summary)

บทนี้ได้ศึกษาและทำความเข้าใจกับระบบพื้นฐานของเกมเอนจิน Unity วัตถุเกมและคอมโพเนนท์ซึ่งเป็นตัวกำหนดคุณสมบัติและพฤติกรรมของวัตถุในเกม ทดลองการประยุกต์คอมโพเนนท์เพื่อสร้างระบบพื้นฐานของการพัฒนาเกม เช่น พลังชีวิต ประเภทของไอเท็ม เป็นต้น ศึกษาถึงลำดับการทำงานและวงจรชีวิตของสคริปต์ เมธอดพื้นฐานหลักของสคริปต์ Start() และ Update()

1/2563

ชื่อ นส.ปาริชาติ ศีรีแก้ว

รหัสนักศึกษา 612110054

ตอนที่.....

สาขาวิชาแอนนิเมชันและเกม วิทยาลัยศิลปะ สื่อ และเทคโนโลยี มหาวิทยาลัยเชียงใหม่

1.7 คำถามและปัญหาชวนคิด (Questions and Problems)

คลิกเลือกวัตถุ Cube บนหน้าต่าง Hierarchy ทางด้านซ้าย (รูปที่ 1.1a) และสำรวจคุณสมบัติของ Cube บนหน้าต่าง Inspector ทางด้านขวามือ (รูปที่ 1.1c) มีจำนวนกี่คอมโพเนนท์และมีคอมโพเนนท์ใดบ้าง เขียนลงในช่องว่าง

มี 5 คอมโพเนนท์ ประกอบด้วย 1.Transform 2.Cube(Mesh Filter) 3.Mesh Renderer 4.Box collider 5.Defaulf-Materail

ให้ทดลองลบคอมโพเนนท์ทั้งหมดของวัตถุ Cube โดยการคลิกที่สัญลักษณ์รูปฟันเฟือง และเลือกเมนู Remove Component

คอมโพเนนท์ใดไม่สามารถลบออกได้ และจงให้เหตุผล

Transform ไม่สามารถลบออกได้ เพราะถ้าไม่มี transform ก็จะไม่มี cube

อธิบายลักษณะที่ปรากฏของวัตถุ Cube มีลักษณะอย่างไร หลังจากลบคอมโพเนนท์ทั้งหมดออกแล้ว

ไม่มีอะไรเลย คล้ายกับ emptyGameObject

ทดสอบรันโปรแกรม (ForwardBackwardAlongXAxisMovement) และอธิบายการทำงานของโปรแกรมมาพอสังเขป

ตอนทีเขียนโค้ด จะมี [Serializefield]แล้วค่าทีเรากำหนดมันจำขึนมาอยู่ทีแถบคอมโพเนนท์ ซึงเราสามารถแก้ไขค่าทีเราต้งได้เลยโดยทีไม่จำเป็นต้องเข้าไปแก้ไขในตัวโค้ด

อธิบายถึงระบบคอมโพเนนท์ใน Unity ตามที่นักศึกษาเข้าใจ

คอมโพเนนท์ทีกำหนดค่าของตัว object ทีเราตั้งค่าที คอมโพเนนท์ทั้ง 5 อย่าง และเราสามารถเพิมคอมโพเนนท์ใหม่ทีเราต้องการเพิมขึนเองได้ทีแอดคอมโพเนนท์

อธิบายลำดับการทำงานของเมธอดทั้งหมดที่มีในสคริปต์ ScriptLifeCycleStudy.cs

1.Awake 2.Start 3.OnDisable 4.OnDestroy 5.OnApplicationPause 6.OnApplicationQuit