

สร้างเกมด้วย Unity

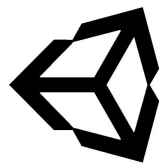
สำหรับผู้เริ่มต้น 2021



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Unity คืออะไร

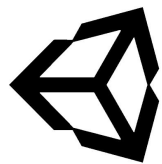
Unity คือ Game Engine หรือเครื่องมือสำหรับการพัฒนา
เกมได้ทั้งรูปแบบ 2 มิติและ 3 มิติที่มีการทำงานข้าม Platform
ได้ ทั้งแบบ Desktop (Windows, Mac, Linux), Mobile
(iOS, Android), Web (HTML5 | WebGL)



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Unity คืออะไร

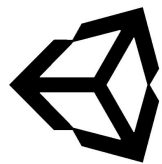
ผู้ใช้สามารถสร้างเกมให้รองรับใน Platform ดังกล่าวได้โดยการพัฒนาเกมเพียงครั้งเดียว ตัว Engine ประกอบด้วย API ในการช่วยเหลือนักพัฒนาด้านการเขียน Script โปรแกรม โดยภาษาโปรแกรมที่ใช้เขียนคือ C# (ก่อนหน้านี้คือใช้ JavaScript , C# ได้)



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Unity คืออะไร

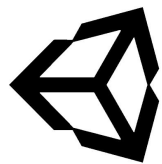
ในปัจจุบัน Unity ได้ถูกนำไปใช้ในงานด้านต่างๆที่ไม่ได้
เกี่ยวข้องกับการทำเกมอย่างเดียว เช่น งานด้านภาพยนตร์ ,
สถาปัตยกรรม , วิศวกรรม รวมไปถึงการโต้ตอบกับผู้ใช้โดย
การพัฒนาในรูปแบบของงานด้าน AR และ VR เป็นต้น



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Unity คืออะไร

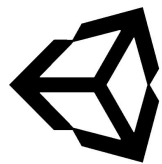
การใช้งานโปรแกรม Unity สามารถใช้งานได้ฟรี ไม่เสียค่าใช้จ่าย แต่อาจจะมีบาง Feature ที่ต้องทำการชำระเงินถึงจะใช้งานได้ เช่น Package เสริมบางตัว โดยต้องทำการสมัครสมาชิกเพื่อรับใบอนุญาตฟรีสำหรับงานส่วนบุคคลหรือบริษัทขนาดเล็ก



<https://www.youtube.com/c/kongruksiamofficial>



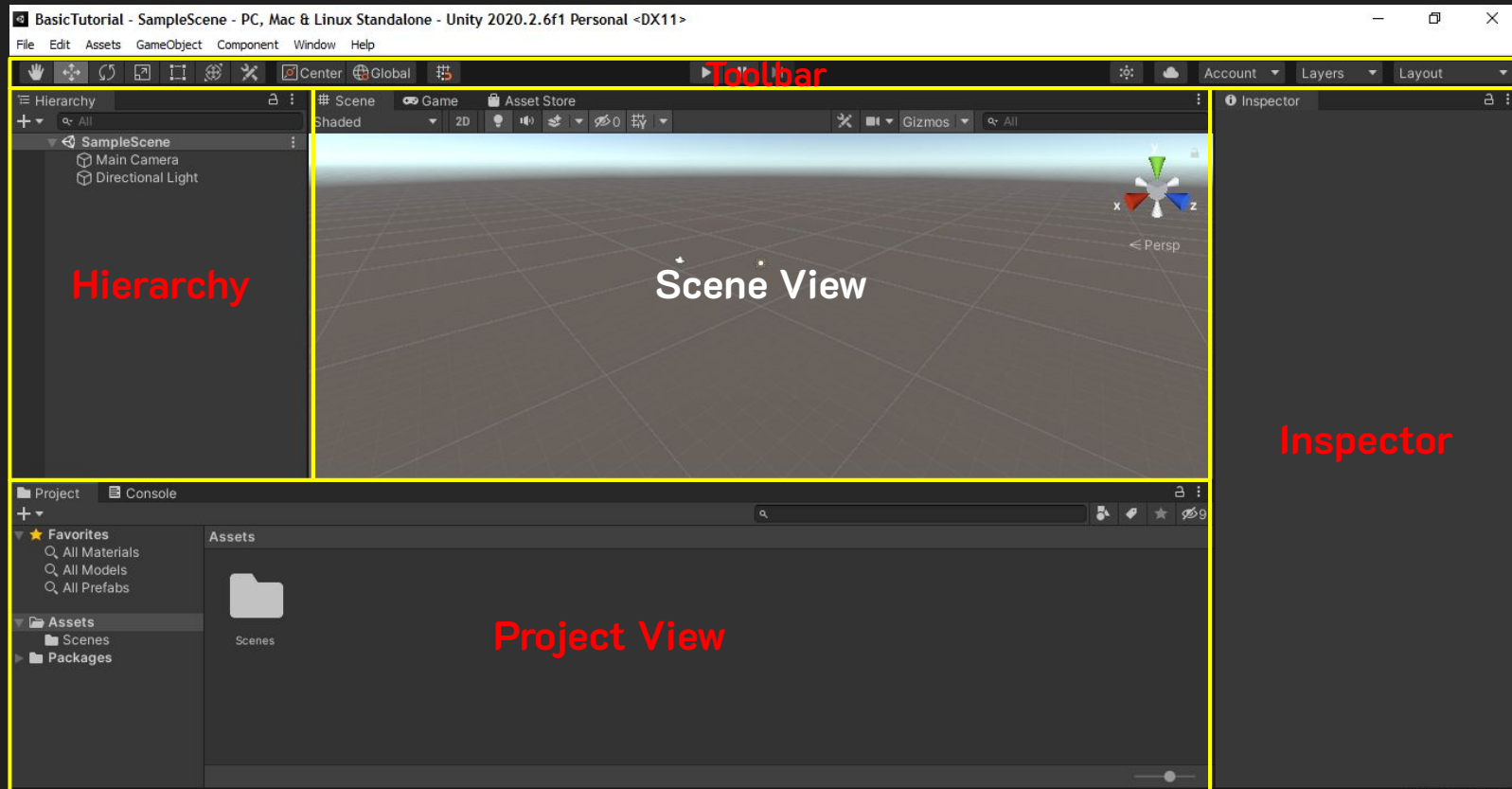
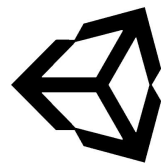
<https://www.facebook.com/KongRuksiamTutorial/>

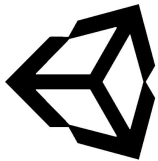


การติดตั้ง Unity Hub

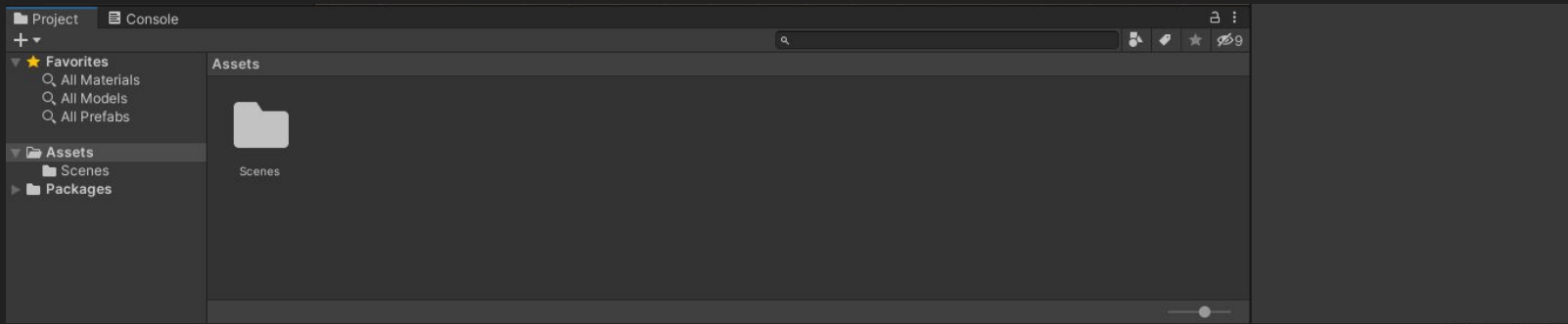
1. <https://store.unity.com/>
2. สร้างบัญชีของ Unity (Unity ID)
3. โหลด Unity Hub และ Unity Version ที่ต้องการใช้งานรวมถึง Component ต่างๆที่ต้องการใช้งาน เช่น Build Android / iOS App , WebGL เป็นต้น
4. Activate License

หน้าจอของโปรแกรม

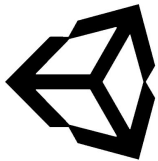




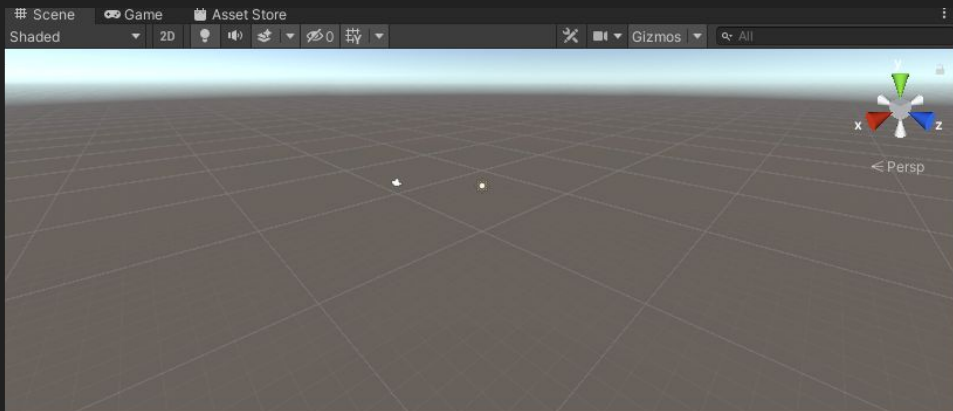
Project View



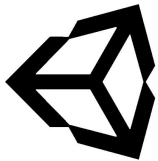
เป็นหน้าต่างที่แสดงโฟลเดอร์และไฟล์ต่างๆที่เก็บและใช้งานในโปรเจค เช่น โมเดล , ภาพ , เสียง , Scene รวมไปถึง Script ไฟล์ จะเรียกส่วนนี้ว่า Asset หากต้องการนำไฟล์จากด้านนอกมาทำงานต้องลากมาไว้ที่ Project



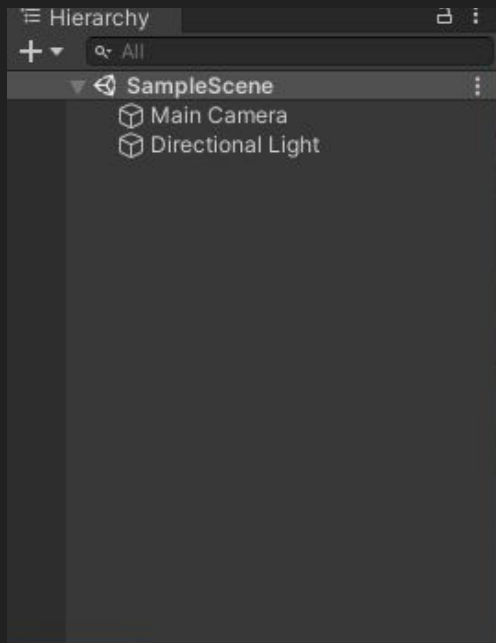
Scene View



ส่วนที่ใช้จัดมุมมองในการออกแบบเกมขึ้นอยู่กับว่าจะแสดงมุมมอง 2 มิติ หรือ 3 มิติ รวมไปถึงการปรับแต่งวัตถุใน Scene เช่น การหมุน ย่อ- ขยายวัตถุ นักพัฒนาสามารถที่จะลาก Asset เข้าไปใน Scene View ได้ซึ่งส่วนที่ออกแบบใน Scene จะถูกแสดงในส่วนหน้าจอเกมด้วย

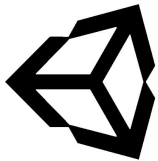


Hierarchy

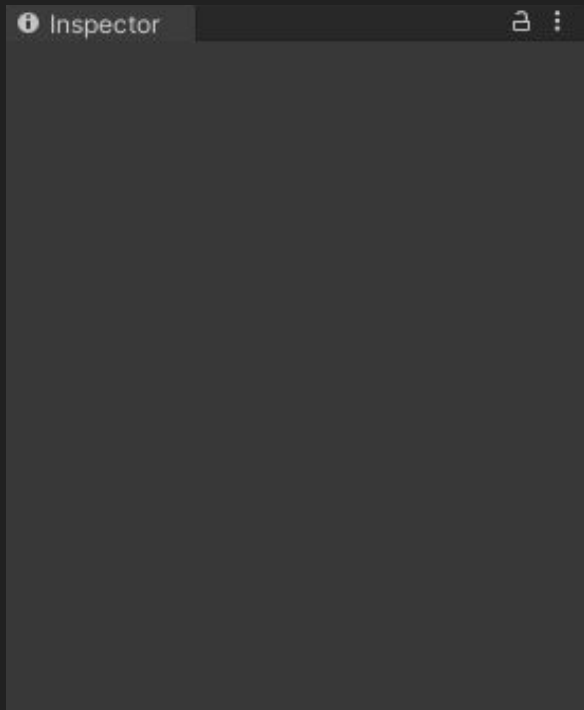


ส่วนที่แสดงลำดับชั้นของวัตถุที่ปรากฏ
ใน Scene View โดยจะบอกชื่อวัตถุที่ทำงาน
ภายใน Scene View ว่ามีวัตถุอะไรบ้าง เช่น
Main Camera , Direction Light เป็นต้น



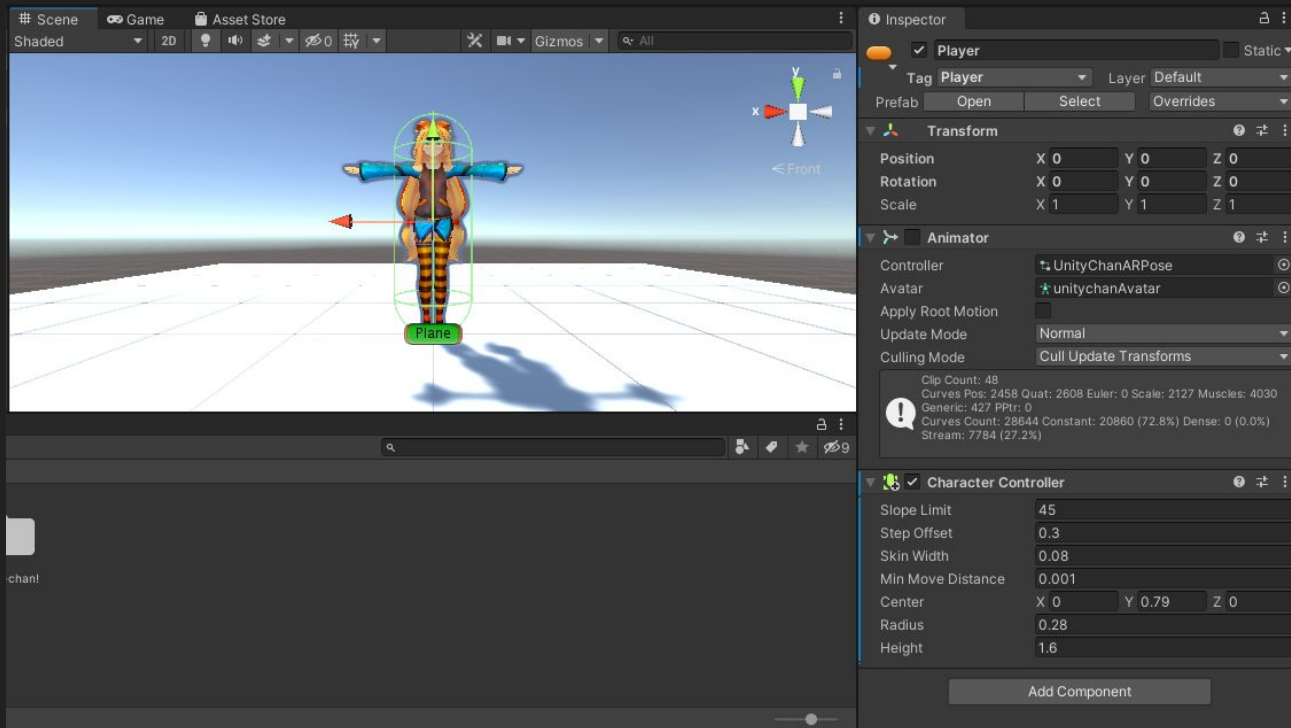
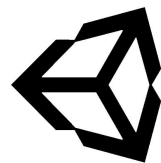


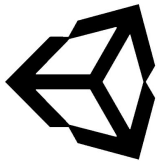
Inspector



หน้าต่างอำนวยความสะดวกในการจัดการของคุณสมบัติทั้งหมดของวัตถุที่เรา กำลังทำงานด้วย เช่น การกำหนดตำแหน่ง หมุนย่อ - ขยาย วัตถุ การตั้งชื่อวัตถุ การใส่ Tag หรือการเพิ่ม Component ลงไปใน วัตถุ เช่น ไฟล์ Script เป็นต้น

Inspector





Toolbar



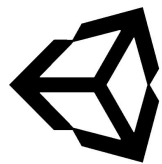
แถบเครื่องมือพื้นฐานสำหรับกำหนดคุณสมบัติต่างๆที่อยู่ภายใน Scene เกม เช่น การปรับตำแหน่ง , ย่อ-ขยาย , หมุน วัตถุการควบคุมการเล่นหรือหยุดเกมและการจัดการบัญชีผู้ใช้



<https://www.youtube.com/c/kongruksiamofficial>

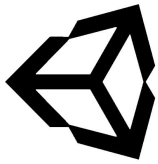


<https://www.facebook.com/KongRuksiamTutorial/>



คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

1.Asset หมายถึงสิ่งที่นำเข้ามาทำงานใน Project สำหรับพัฒนาเกม ยกตัวอย่าง เช่น Model ตัวละคร, Animation , Script, Texture , เสียงและอื่นๆ ซึ่ง Asset จะถูกเก็บไว้ใน Project มีชื่อไฟล์เดอร์ว่า Assets ถ้าหากไม่ได้อยู่ในโฟลเดอร์นี้ก็ไม่สามารถพัฒนาเกมได้



คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

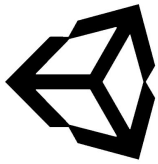
2.Scene(ฉาก) หมายถึง หน้าจอแสดงผลหรือฉากการทำงาน
ของเกมโดยภายใน 1 เกมสามารถแบ่งฉากออกเป็นหลายๆฉากได้
เช่น ฉากเมนู , ฉากเปิดเกม , ฉากเนื้อเรื่อง , ฉากจบเกม เป็นต้น



<https://www.youtube.com/c/kongruksiamofficial>

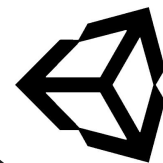


<https://www.facebook.com/KongRuksiamTutorial/>



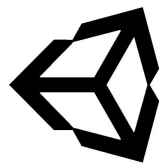
คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

3.Camera หมายถึง กล้องที่ใช้จับภาพต่างๆภายในเกม ภาพที่แสดงผลมีมุมมองอย่างไรขึ้นอยู่กับนักพัฒนาเกม โดยภายในเกมสามารถมีกล้องได้มากกว่า 1 ตัว ถ้าไม่มีกล้องก็จะไม่มีภาพปรากฏในเกมนั่นเอง



คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

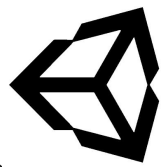
4.Light หมายถึง วัตถุประเภทแสง ใช้ปรับความมืด
ความสว่างภายในเกมรวมไปถึงเงาของวัตถุ



คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

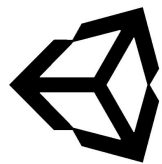
5.GameObject หมายถึง วัตถุหรือองค์ประกอบต่างๆที่อยู่ภายในเกม เช่น คน สัตว์ สิ่งของ เป็นต้น

5.1 Empty Object วัตถุว่างเปล่าที่ไม่ปรากฏใน Scene
เกมนิยมนำมาใช้สำหรับรัน Script ไฟล์ , จัดกลุ่มวัตถุ ,
เรียกใช้งาน Component ต่างๆเช่น การเล่น Sound
Background , การหยุดเกม (Play/Pause) เป็นต้น



คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

6.Component หมายถึง คุณสมบัติต่างๆที่อยู่ใน GameObject เช่น ระบบฟิสิกส์ ระบบควบคุม Animation , ระบบเสียง หรือ Script ต่างๆที่อยากให้ GameObject นั้นสามารถทำงานตามที่ต้องการ



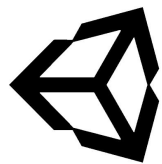
คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

7.Transform หมายถึง Component ที่ต้องมีอยู่ในวัตถุทุกตัวเป็นสิ่งที่ขาดไม่ได้ โดยจะเป็นตัวที่เก็บค่า 3 ค่าได้แก่

Position - ตำแหน่งของวัตถุ (พิกัด x , y , z ถ้าอยู่ในรูป 3 มิติ)

Rotation - การหมุนวัตถุ (พิกัด x , y , z ถ้าอยู่ในรูป 3 มิติ)

Scale - ขนาดของวัตถุ (พิกัด x , y , z ถ้าอยู่ในรูป 3 มิติ)

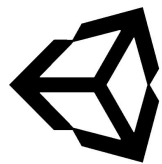


คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

8.Rigidbody / Rigidbody2D หมายถึง Component ที่จัดการด้านระบบฟิสิกส์ในวัตถุ เช่น มวล(mass) , แรงโน้มถ่วง (Gravity) , การเคลื่อนไหวของวัตถุ (Kinematic) , การล็อควัตถุ (Freeze)

9.Sprite คือ ภาพที่ทำมาใช้งานในระบบเกม (2 มิติ)



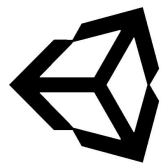


คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

10.Particle System หมายถึง ระบบ Effect ภายในเกม เช่น Effect ระเบิด , Effect ไฟ , ฝุ่น ใช้เพิ่มสีสันภายในเกม

11.Texture หมายถึง รูปภาพที่นำมาประกอบในโมเดลให้มีความสมจริงมากยิ่งขึ้น

12.Material หมายถึง เม็ดสีที่ปรากฏในตัววัตถุ

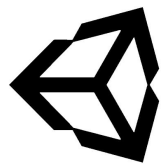


คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

13.SkyBox หมายถึง Asset ที่ทำงานเกี่ยวกับท้องฟ้าในเกม

14.Terrain หมายถึง ส่วนที่จัดการเกี่ยวกับสภาพแวดล้อมภายในเกม เช่น ทำภูเขา ต้นไม้ หิน เป็นต้น

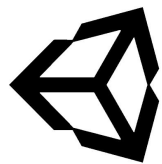
15.Wind Zone หมายถึง Game Object ที่จัดการเกี่ยวกับลม ช่วยเพิ่มความสมจริงให้กับเกม เช่น พายุ , ใบไม้ขยับตามลม เป็นต้น



คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

16.Vector2 คือ ตัวแปรที่เก็บตำแหน่งแกน X , Y ในระบบเกมแบบ 2 มิติ

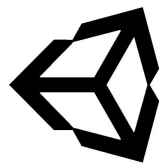
17.Vector3 คือ ตัวแปรที่เก็บตำแหน่งแกน X , Y , Z ในระบบเกมแบบ 3 มิติ



คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

18.Collider / Collider2D หมายถึง Component ที่ใช้ตรวจสอบการชนของวัตถุในเกม เช่น ถ้าเก็บไอเทมก็จะได้รับ HP เป็นต้น

19.Ray / RayCast หมายถึง ส่วนที่ใช้ในการจัดการเกี่ยวกับการคำนวณระยะห่างของวัตถุ โดยอาศัย Ray เป็นตัวเช็คการตกกระทบกับวัตถุ แล้วจะทำให้เกิด Action อย่างไร สามารถนำมาใช้ประโยชน์ได้หลายๆอย่างรวมถึงเช็คการชนของวัตถุด้วย เป็นต้น



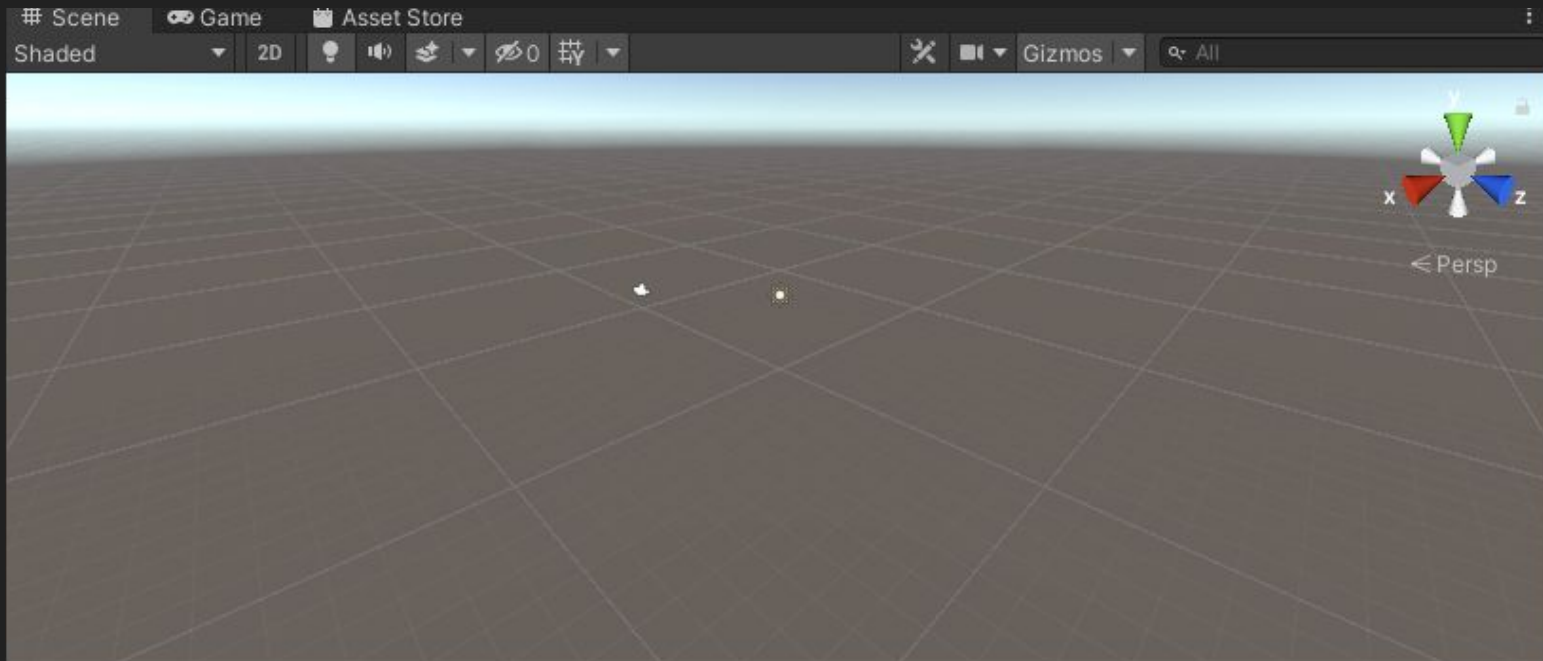
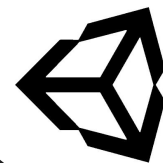
คำศัพท์ที่ควรรู้ก่อนใช้โปรแกรม Unity

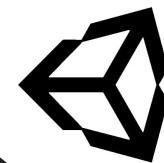
20.Prefab หมายถึง การสร้าง GameObject ตัวต้นแบบ โดยทำการโคลนนิ่งวัตถุที่มีคุณสมบัติต่างๆที่ต้องการ เพื่อนำวัตถุนั้นมาใช้งานซ้ำโดยที่ไม่ต้องสร้างหรือกำหนดคุณสมบัติหลายครั้ง

21.Tag หมายถึง ป้ายกำกับที่ใช้แบ่งประเภทของวัตถุ เช่น Player , Camera , Enemy , NPC, Ground เป็นต้น (ไม่จำเป็นต้องกำหนดก็ได้)

22.Layer หมายถึง ลำดับชั้นการทำงานของวัตถุ

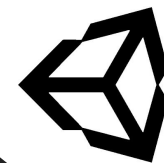
Scene View เบื้องต้น





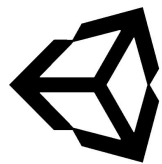
Scene View เบื้องต้น

คำสั่ง	คำอธิบาย
เมาส์ซ้าย	ใช้เลือกวัตถุที่อยู่ใน Scene View
เมาส์ขวา	คลิกข้างไว้เพื่อหมุนดูวัตถุโดยรอบ
เมาส์กลาง (Scroll)	เลื่อนดูมุมมองแบบอิสระ (Hand Tool)
Scroll เมาส์ขึ้น	ซูมเข้า
Scroll เมาส์ลง	ซูมออก
Double Click	เลือกวัตถุแล้วซูมเข้า



Scene View เบื้องต้น

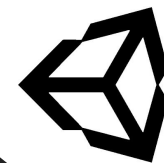
คำสั่ง	คำอธิบาย
Arrow (ลูกศร)	ควบคุมมุมมองแสดงผลตามทิศทางที่ต้องการ
Number 2	ปรับการแสดงผลแบบ 2 มิติ หรือ 3 มิติ
W (Move Tool)	ควบคุมการเคลื่อนที่วัตถุในแกนที่ต้องการ
E (Rotate Tool)	หมุนวัตถุรอบแกนที่ต้องการ
R (Scale)	ย่อ - ขยายวัตถุตามแกนที่ต้องการ



มุมมอง Perspective & Orthographic

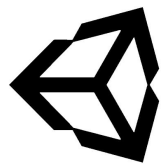
Perspective หมายถึง การแสดงผลมุมมองตามสายตาของมนุษย์ ถ้าวัตถุอยู่ใกล้จะมีขนาดใหญ่ อยู่ไกลจะมีขนาดเล็กตามระยะห่างของสายตา มีรูปแบบ View 2 แบบ ได้แก่ Local (มองเห็นวัตถุที่เลือก) , Global (มองเห็นวัตถุทั้งหมด)

Orthographic หมายถึง การแสดงผลมุมมองตามจริงของวัตถุโดยไม่ยืดกับระดับสายตาตามนุษย์ มุมมองนี้ไม่ว่าจะหมุนไปทางใด ขนาดของวัตถุก็จะคงที่เสมอไม่ขึ้นอยู่กักระยะใกล้ไกล เหมาะในการเปรียบเทียบขนาดวัตถุ



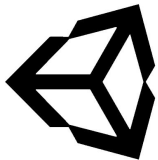
การสร้าง C# Script

1. Project > Create > C# Script > ตั้งชื่อ .cs
2. คลิกขวา > Create > C# Script > ตั้งชื่อ .cs
3. Assets > Create > C# Script > ตั้งชื่อ.cs



ข้อควรระวังในการทำงานกับไฟล์ Script

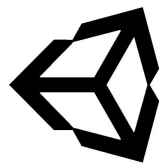
1. ควรตั้งชื่อไฟล์ Script ให้เสร็จก่อนเพื่อให้ระบบนำชื่อไปกำหนดเป็นชื่อ Class
2. ชื่อไฟล์ Script กับชื่อ Class ต้องเป็นชื่อเดียวกันเท่านั้น
3. ควรกำหนดอักขรตัวแรกของไฟล์ Script เป็นตัวพิมพ์ใหญ่
4. ถ้าไม่ตั้งชื่อ Script ในตอนแรกที่เราสร้าง แล้วมาแก้ไขชื่อ Class ในภายหลัง อาจจะส่งผลให้ Script นั้นไม่สามารถทำงานได้
5. ควรเลี่ยงการตั้งชื่อไฟล์ Script เป็นภาษาไทย



วิธีการนำไฟล์ Script ไปทำงาน

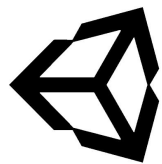
1. เลือกวัตถุที่ต้องการนำ Script ไปรัน > ลาก Script ไฟล์ (.cs) ไปใส่ในส่วนของ Inspector วัตถุ
2. เลือกวัตถุที่ต้องการนำ Script ไปรัน > ไปที่ Inspector > Add Component > Scripts > ชื่อไฟล์.cs





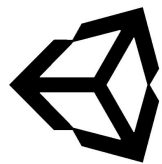
ฟังก์ชัน Start และ Update

Start() คือ ฟังก์ชันหรือเมธอดที่จะถูกเรียกใช้งานในตอนเริ่มต้น โดยคำสั่งที่เขียนภายใน Start จะถูกเรียกใช้งานแค่ครั้งเดียวและทำงานเมื่อสั่งให้ Script Enabled เท่านั้น



ฟังก์ชัน Start และ Update

Update() คือ ฟังก์ชันหรือเมธอดที่ใช้รันคำสั่งเป็น Frame ต่อ Frame คือ ทำงานตลอดเวลาโดยอ้างอิงตามความเร็วของเครื่องคอมพิวเตอร์ที่ใช้รันเกม



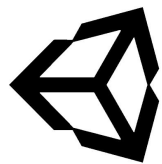
ฟังก์ชัน Start และ Update

Update() จะทำงานและอ้างอิงตามการคำนวณค่า Frame

Rate (FPS : Frame Per Second)

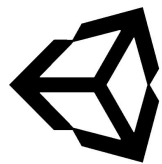
- ถ้า FPS ต่ำก็จะทำงานช้า
- ถ้า FPS สูงก็จะทำงานเร็ว

เช่น FPS = 60 หมายถึง ทำงาน 60 Frame ต่อวินาที นั่นเอง



ฟังก์ชัน Start และ Awake

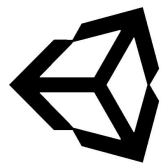
1. **Start()** คือ ฟังก์ชันหรือเมธอดที่จะถูกเรียกใช้งานในตอนเริ่มต้นโดยคำสั่งที่เขียนภายใน Start จะถูกเรียกใช้งานแค่ครั้งเดียวและทำงานเมื่อสั่งให้ Script Enabled เท่านั้น
2. **Awake()** มีการทำงานคล้ายกับ Start คือทำงานในตอนเริ่มต้น แต่ Awake จะทำงานเมื่อโหลด Component หรือ Script เข้ามาทำงาน คือสั่ง Enabled / Disabled ก็ทำงานเหมือนเดิม



ฟังก์ชัน Update และ Fixed Update

FixedUpdate() ทำงานคล้ายๆกับ Update แต่ว่าจะมีการคำนวณระยะเวลาคงที่ (ค่าตายตัว) และ Realtime ไม่ได้จำแนกตามความเร็วเครื่องใช้ร่วมกับระบบเวลาและระบบฟิสิกส์ (**FrameRate คงที่**)

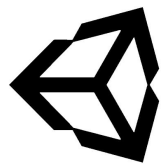
ยกตัวอย่าง เช่น ถ้าใช้ Update นานๆจะมีการ กระตุกเนื่องจากอ้างอิงตามความเร็วเครื่อง ส่วน FixedUpdate จะคงที่



ฟังก์ชัน LateUpdate

ฟังก์ชัน `LateUpdate()` จะทำงานหลังฟังก์ชัน `Update` อีกที
ใช้ทำงานคำสั่ง เมื่อ Object อัปเดตค่าเรียบร้อยแล้ว

`Time.deltaTime` คำนวณความต่างเวลาระหว่าง Frame ปัจจุบัน
กับ Frame ก่อนหน้า



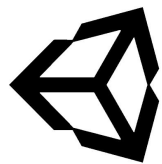
ควบคุมการเคลื่อนที่ด้วยคีย์บอร์ด

คำสั่งที่ใช้รับค่า `Input.GetAxis ()`

ชื่อ **Axis : Horizontal** เคลื่อนที่ในแนวแกน x

ควบคุมการทำงานด้วย

- ปุ่ม Left Arrow , Right Arrow
- ปุ่ม A , D



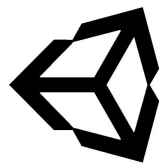
ควบคุมการเคลื่อนที่ด้วยคีย์บอร์ด

คำสั่งที่ใช้รับค่า `Input.GetAxis ()`

ชื่อ **Axis : Vertical** เคลื่อนที่ในแนวแกน z

ควบคุมการทำงานด้วย

- ปุ่ม Up Arrow , Down Arrow
- ปุ่ม W , S



ควบคุมการเคลื่อนที่ด้วยคีย์บอร์ด

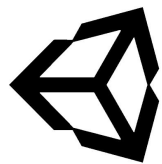
คำสั่งที่ใช้รับค่า `Input.GetAxis ()`

ชื่อ **Axis : Jump** เคลื่อนที่ในแนวแกน y

ควบคุมการทำงานด้วย

- ปุ่ม SpaceBar

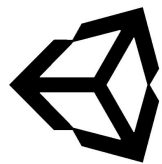




ควบคุมการเคลื่อนที่ด้วยคีย์บอร์ด

คำสั่งที่ทำให้วัตถุเคลื่อนที่ได้

- **Rigidbody.velocity** ทำให้วัตถุเคลื่อนที่ในแกนที่ต้องการด้วยความเร็วคงที่
- **Rigidbody.MovePosition** ทำให้เคลื่อนที่รอบๆแกนที่ต้องการโดยอ้างอิงตำแหน่งปัจจุบันกับตำแหน่งที่เคลื่อนที่ไป

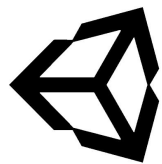


ควบคุมการเคลื่อนที่ด้วยคีย์บอร์ด

คำสั่งที่ทำให้วัตถุเคลื่อนที่ได้

- **Rigidbody.AddForce** คือ การทำให้วัตถุเคลื่อนที่โดยอาศัยการเพิ่มแรงเข้าไปโดยวัตถุจะเคลื่อนที่ความเร็วสูงสุดในตอนเริ่มต้น แล้วลดความเร็วลงเมื่อเวลาผ่านไป (คิดแรงเสียดทาน)





ควบคุมการเคลื่อนที่ด้วยคีย์บอร์ด

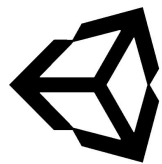
คำสั่งที่ทำให้วัตถุเคลื่อนที่ได้

- **AddForce** จะมีความเร็วสัมพันธ์กับขนาดมวลของวัตถุโดยอ้างอิงจากสมการแรงและการเคลื่อนที่ของนิวตัน

$$F=ma$$

มวลน้อย = ใช้แรงน้อย | มวลมาก = ใช้แรงมาก

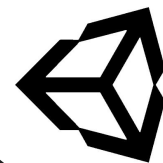




โคลนนิ่งวัตถุด้วย Prefab

Prefab หมายถึง การสร้าง GameObject ตัวต้นแบบ
โดยทำการโคลนนิ่งวัตถุที่มีคุณสมบัติต่างๆที่ต้องการเพื่อจะ
นำวัตถุเหล่านั้นมาใช้งานซ้ำโดยที่ไม่ต้องสร้างหรือกำหนด
คุณสมบัติหลายครั้ง





การสร้าง Terrain

Terrain หมายถึง ส่วนที่จัดการเกี่ยวกับสภาพแวดล้อม
ภายในเกม เช่น ทำภูเขา ต้นไม้ หญ้า เป็นต้น