

Input Controller ใน XNA

Suphot Sawattiwong
tohpus@hotmail.com

อุปกรณ์ควบคุมใน XNA

- ใน XNA ได้แบ่งอุปกรณ์ควบคุมเป็นทั้งหมด 3 ประเภท ได้แก่ Keyboard, Mouse, XBOX360 Controller



- โดยในที่นี้ขอเน้นการใช้งาน Keyboard กับ Mouse

การสนับสนุนอุปกรณ์ควบคุมใน XNA

| Device | Windows | XBOX360 |
|----------|---------|---------|
| Keyboard | X | X |
| Mouse | X | |
| GamePad | X | X |

- จากตารางเห็นได้ว่าในเครื่อง **XBOX360** ไม่มีการสนับสนุน **Mouse** และ **GamePad** ที่ใช้ใน XNA ต้องเป็น **GamePad** ที่สนับสนุน **XBOX360 Controller** เท่านั้น ไม่เช่นนั้นจะใช้ไม่ได้

Keyboard

- ใน XNA แบ่ง Keyboard ที่ใช้แบ่งเป็น 2 ประเภท คือ Keyboard ทั่วไป กับ ChatPad



ChatPad



Keyboard

- โดยที่การทำงานหลักของ **Keyboard** ใน **XNA** เน้นการใช้งาน 2 **Class** หลัก ได้แก่ **Keyboard** และ **KeyboardState**
- โดยมี **enums** ไว้เรียกอ้างอิงค่าปุ่มต่างๆ ของ **Keyboard** คือ **Keys**
- วิธีการเรียกใช้ **Keyboard** สามารถทำได้โดยใน **Method Update** เช่น

ตัวอย่างการเขียนควบคุม Keyboard ใน XNA

```
protected override void Update(GameTime gameTime)
{
    // ทดสอบการใช้ Keyboard โดยหากกดปุ่ม Escape ให้ออกจากเกม
    if (Keyboard.GetState().IsKeyDown(Keys.Escape))
    {
        this.Exit();
    }
    base.Update(gameTime);
}
```

Keyboard

- ส่วน **KeyboardState** เป็นตัวที่แสดงสถานะว่าขณะนี้ **keyboard** อยู่ในสถานะ **KeyDown** หรือว่า **KeyUp**
 - **KeyDown** คือจังหวะที่เรากดปุ่ม
 - **KeyUp** คือจังหวะที่เราปล่อยปุ่ม

การเอาสถานะของ Keyboard

- ชุดคำสั่งการนำสถานะของ Keyboard ไปใช้งานสามารถทำได้ผ่านทางชุดคำสั่ง `Keyboard.GetState()`
- หากใช้คำสั่งดังกล่าว ทำให้ได้ค่าของ `KeyboardState` มาเพื่อพร้อมใช้งานต่อไป โดยชุดคำสั่ง `Keyboard.GetState()` มีการเรียกใช้ได้ 2 แบบดังนี้

| ชื่อ | คำอธิบาย |
|-------------------------------------------------|-------------------------------------------------------------------------------------------|
| Keyboard.GetState () | เป็นคำสั่งส่งค่า keyboard state ปัจจุบันออกมา |
| Keyboard.GetState (PlayerIndex) | เป็นคำสั่งส่งค่า Chatpad state ของแต่ละผู้เล่นออกมา ใช้คู่กับ XBOX360 Controller เท่านั้น |

KeyboardState

- Method ของ KeyboardState ที่จำเป็นต้องรู้

| Name | Description |
|-------------------------|----------------------------------------------------------------------------------------|
| <u>GetPressedKeys()</u> | จะส่งค่าที่มีการ Key ที่มีการกดขณะนั้นเป็น Array มาให้ |
| <u>IsKeyDown(Keys)</u> | จะส่งค่า true หรือ false ออกมา โดยจะส่งค่า true ออกมาก็ต่อเมื่อ key นั้นได้รับการกด |
| <u>IsKeyUp(Keys)</u> | จะส่งค่า true หรือ false ออกมา โดยจะส่งค่า true ออกมาก็ต่อเมื่อ key นั้นไม่ได้รับการกด |

ตัวอย่างของการรับค่า `KeyboardState`

- ให้เปิดดู Project ที่ชื่อว่า `InputKeyboardState`

Mouse

- ใน XNA นั้น Mouse เป็น input ที่สามารถใช้ได้ใน platform Windows เท่านั้น (ไม่สามารถใช้ได้ใน XBOX360) โดยมี Class Mouse และ MouseState
- โดย Mouse จะเป็นตัวเชื่อมต่อกับ Hardware
- โดยปกติ Mouse Cursor ใน XNA ถูกซ่อนไว้หากต้องการให้เห็น Mouse Cursor ต้องใช้คำสั่งดังต่อไปนี้ใน Class Game1

```
this.IsMouseVisible = true;
```

ดูตัวอย่างได้ใน Project ชื่อว่า InputMouse

Mouse

- โดย **MouseState** เป็นตัวแสดงสถานะต่างๆ ของ **mouse** โดยสามารถกำหนดได้ดังนี้

```
MouseState mouseState = Mouse.GetState();
```

ตัวแปรใน **MouseState** มีดังต่อไปนี้

| Name | Description |
|---------------------------------|---------------------------------|
| <u>XButton1</u> | ส่งค่า ButtonState ของ XButton1 |
| <u>XButton2</u> | ส่งค่า ButtonState ของ XButton2 |

MouseState

| Name | Description |
|-------------------------|-----------------------------------------|
| <u>ScrollWheelValue</u> | เก็บค่าการเลื่อน Scroll ตั้งแต่เริ่มเกม |
| <u>X</u> | Mouse Position X |
| <u>Y</u> | Mouse Position Y |

MouseState

| Name | Description |
|-------------------------------------|--------------------------------------------|
| <u>LeftButton</u> | ส่งค่า ButtonState ของ Left Button ออกไป |
| <u>MiddleButton</u> | ส่งค่า ButtonState ของ Middle Button ออกไป |
| <u>RightButton</u> | ส่งค่า ButtonState ของ Right Button ออกไป |

ButtonState

- Enum ButtonState จะเก็บ Event อยู่ 2 แบบ คือ KeyPress และ Key Release โดยอุปกรณ์ควบคุมที่ใช้ ได้แก่ Mouse และ GamePad

| Member name | Description |
|-------------|-------------------|
| Pressed | สถานะตอนกดปุ่ม |
| Released | สถานะตอนปล่อยปุ่ม |

ตัวอย่างการ Click Mouse

- ให้เปิด **Project MouseClick** ขึ้นมา แล้วรัน เห็นได้ว่าไม่ว่า **click mouse** ซ้ายตรงไหน ก็เป็นปุ่มทั้งหมด
- ซึ่งหากทำเป็นปุ่มกดจริงๆ ต้องรู้จักเรื่องของการชนกัน หรือ **Collision Detection** ก่อน

Collision Detection

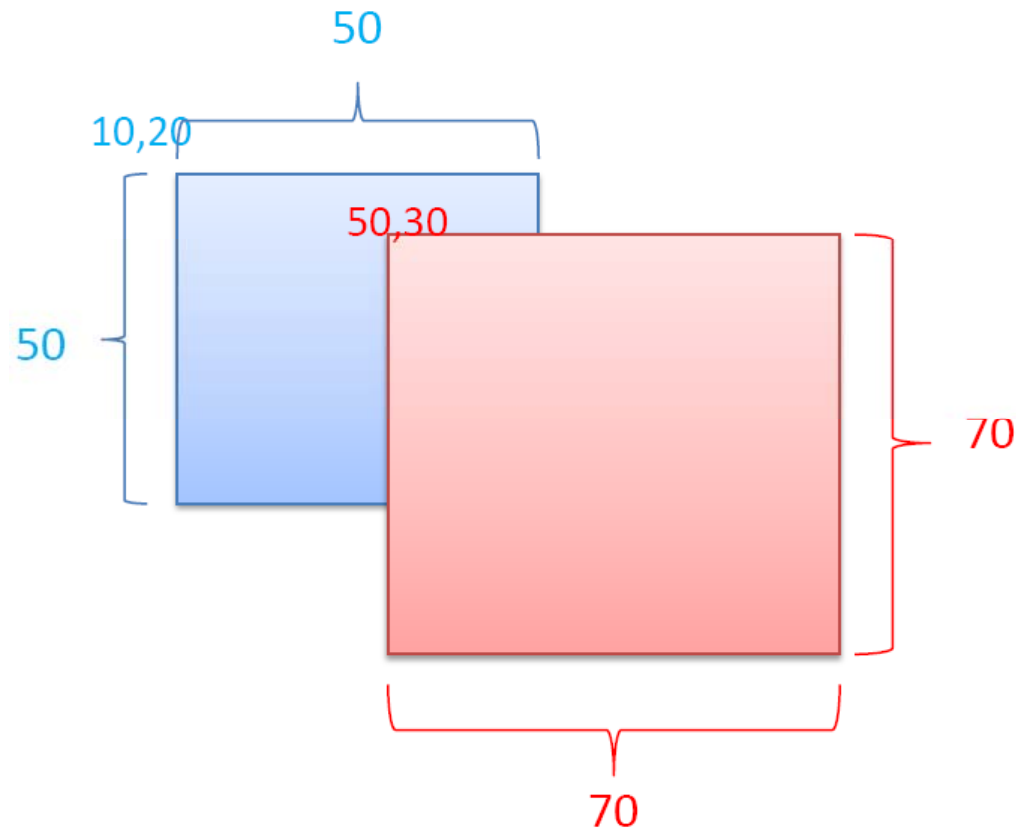
- การชนกันของวัตถุ ในเกมเราจะวัดค่าจากการชนได้โดยการดูว่า วัตถุ 2 อันได้มีการ ซ้อนทับกันหรือไม่



- โดย **Collision Detect** จะเป็นการเช็คค่าว่าวัตถุ 2 ชิ้นนี้ซ้อนทับกันหรือไม่

Collision Detection

- โดยจะเอาตำแหน่ง ของวัตถุ ความกว้างยาวของวัตถุมาวาดเป็นสี่เหลี่ยม



Collision Detection

- ชุดคำสั่งในการใช้ได้โดยอาศัย **Rectangle** ในการตรวจสอบเรื่อง การชนได้ดังนี้

สี่เหลี่ยมอันที่ 1.**Intersects**(สี่เหลี่ยมอันที่ 2)

เช่น **cursorRect.Intersects(playBtnRect)**

โดยคำสั่งนี้ ได้ส่งค่า **true** หากเกิดการชนกัน และ **false** หากไม่มีการชนกันเกิดขึ้น

- สามารถดูได้ใน **Project MouseClick2**