

2D Basic Programming for XNA

Suphot Sawattiwong
tohpus@hotmail.com



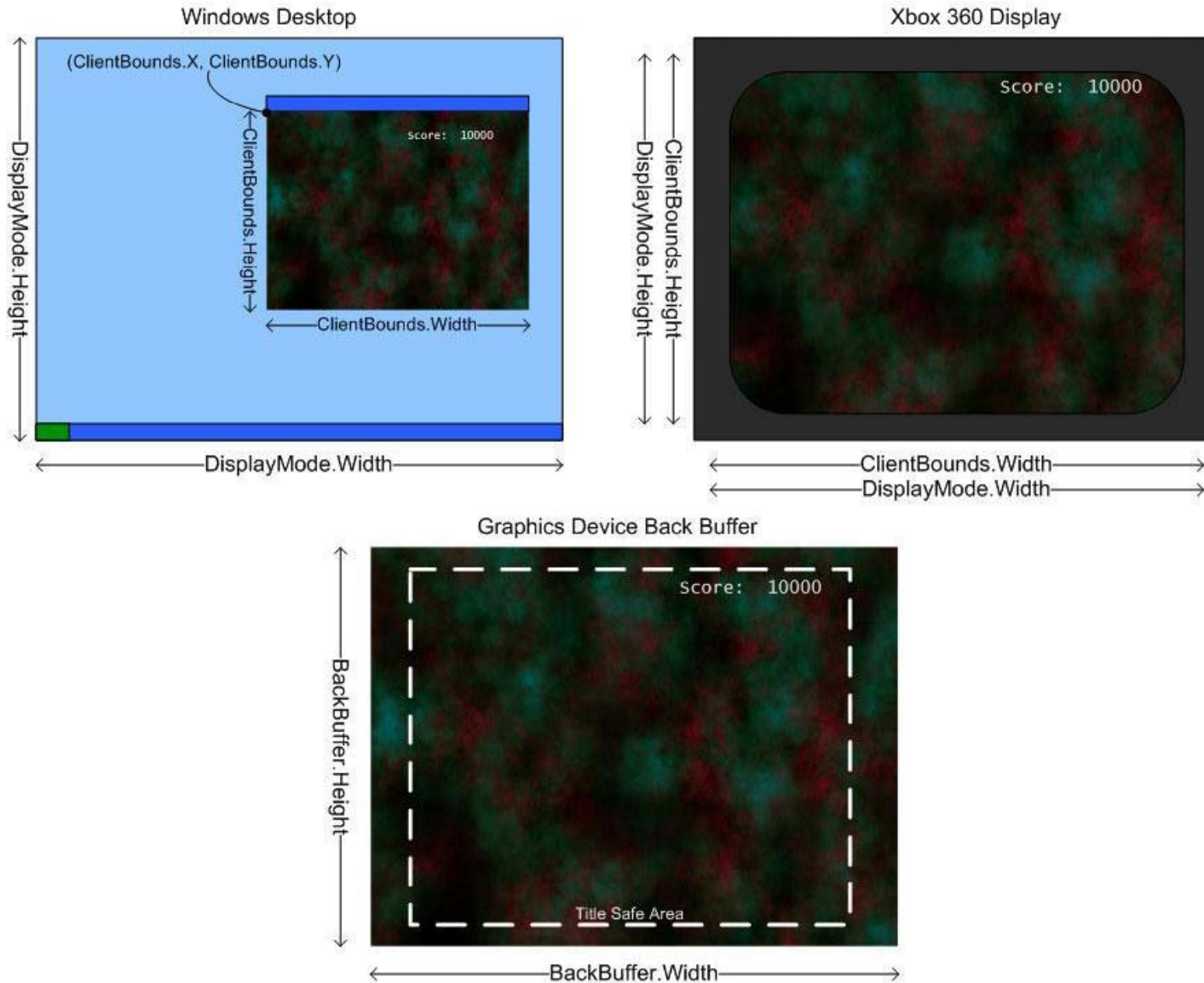
Display Resolution

- **Display Resolution** เป็นเรื่องที่ต้องทำความเข้าใจกันก่อน โดยปกติ **Resolution** ของ **PC** จะขึ้นอยู่กับ **Computer** ที่ใช้ โดยมาตรฐานปัจจุบัน อยู่ที่ **1024x768** แต่ถ้าหากเป็น **Wide Screen** ก็จะมีจุดที่แตกต่างกัน ออกไปตาม **ratio** ที่เปลี่ยนไป
- สำหรับ **XBOX360** เนื่องจากเครื่องเกมได้ถูกใช้มาตรฐาน **HD(High Definition)** ซึ่งจะมีตั้งแต่ **480i, 480p, 720i, 720p** จนถึง **1080i**
- การที่เราทำเกมเราอาจจะต้องระวังเรื่อง **Resolution** ไปด้วยเช่นกัน

ความกว้างXยาวของจอภาพใน XNA

- ความกว้างยาวของจอภาพ แบ่งเป็น 3 ลักษณะดังนี้
 - **Display Mode** เป็นขอบเขตของหน้าจอของระบบ ซึ่งแตกต่างกับหน้าจอ หน้าต่างของเกม
 - **Back Buffer** เป็นขอบเขตเฉพาะในการแสดงผล ซึ่งใช้ในการคำนวณ **next render** โดยใน **Windows Back Buffer** จะมีค่าเดียวกับหน้าจอของเกม แต่สำหรับใน **XBOX360** ถูกสร้างตาม **User** กำหนด และเมื่อ **Fullscreen Mode** บน **Windows Back Buffer** จะเป็นไปตาม **Display Mode** ทันที
 - **ViewPort** เป็นหน้าจอที่ใช้แสดงผล เป็นลักษณะของกล้อง ซึ่งความกว้าง และความยาว เป็นความกว้างความยาวของกล้องด้วยเช่นกัน โดย **viewport** ไม่มีความเกี่ยวข้องกับ ขนาดของ **Game Windows** แต่ **viewport** สัมพันธ์กับค่า **Display Mode**

ความกว้างXยาวของจอภาพใน XNA



การทำงานกับ GraphicsDevice ใน XNA

- การตั้งค่าหน้าจอ Windows
`graphics.PreferredBackBufferWidth = <ความกว้าง>;`
`graphics.PreferredBackBufferHeight = <ความยาว>;`
- การทำให้ Windows เป็น โหมด Full Screen
`graphics.IsFullScreen = <true|false>;`
- การสลับโหมด Windows จาก Windows เป็น Full Screen
หรือ Full Screen เป็น Windows Mode
`graphics.ToggleFullScreen();`
- ดูตัวอย่างใน Project ScreenWidthHeight

Resource ที่ใช้ในเกม

- **File** รูปภาพ ใน **XNA** สามารถอนุญาตให้ใช้รูปภาพเป็น **jpg, gif,** และ **png** ก็ได้ โดยปกติ เพื่อความสะดวกเราจะใช้ **format png** ในการทำภาพในเกม เนื่องจากว่าสะดวกในการเจาะ **Transparent**
- **Sound** ที่ใช้ **Support** มี **wav, aif** และ **aiff** แต่หากจะนำมาใช้ ต้องผ่าน **tools** ที่ชื่อว่า **Microsoft Cross-Platform Audio Creation Tool (XACT)**

Content Pipeline

- โดยปกติ **Game Resource** จะถูก **Load** เก็บมาไว้ใน **memory** ก่อนที่จะทำการใช้งาน โดยที่ **Content Pipeline** จะเป็นตัวช่วยในการ **load** เข้ามา ซึ่งเป็นการนำ **File** ที่มีคุณสมบัติต่างกัน ไม่ว่าจะเป็นรูปภาพ หรือว่าเสียงจะถูกเก็บใน **format** ที่ง่ายต่อการโหลดเพื่อมาใช้ในเกมเมื่อเกม **start**
- ชั้นแรกรู้จักตัวแปร ที่ชื่อ **“Content”** ใน
Content.RootDirectory = "Content";
- เป็นตัวแปรที่ **XNA** กำหนดมาใน **Class game** ของ **XNA** เพื่อให้สะดวกต่อการใช้งาน
- **Content.RootDirectory** เป็นการบอกว่า **Folder** ที่ **Content** อยู่ นั้น อยู่ใน **Folder “Content”** นั้นเอง

การนำภาพเข้า Project

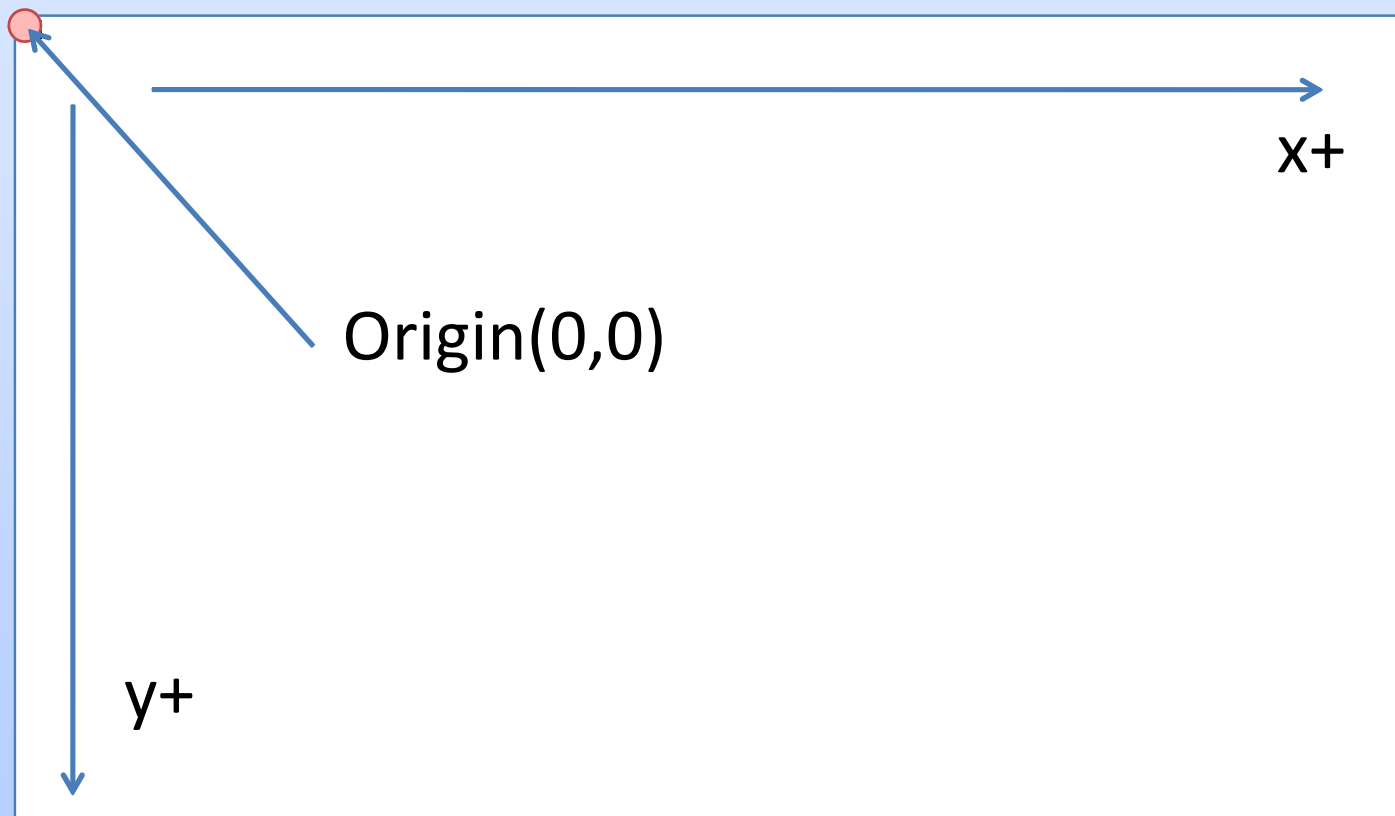
- Click ที่ Folder Content ใน Solution Explorer
- เลือก Add>Add Exist Item แล้วไปยัง Folder ที่เรา Save ไฟล์ ภาพไว้ (ในแผ่น DVD Folder “Resource Example”) File ภาพจะถูก copy เข้ามาไว้ใน Content ทันที

การวาดภาพใน XNA

- ก่อนอื่นมาทำความเข้าใจของคำว่า **Sprite** คือรูปภาพนั่นเอง **XNA** ทำการอ้างอิงถึงรูปภาพผ่านทาง **Texture2D Object** ซึ่งเราจะทำการวาดภาพเหล่านี้ลงบนหน้าจอโดยตรง
- การวาดภาพใน **XNA** ต้องทำความรู้จักกับระบบ **2D Coordinate** เสียก่อน เพื่อเข้าใจในตำแหน่งการวางภาพแบบ **2D**
- โดยในโลกของ **2D** จะมีแค่ แกน **x,y** เท่านั้น ซึ่งต่างจาก **3D** ที่ใช้ **x, y, และ z**
- **origin** ของ **XNA** อยู่บนซ้ายสุดของหน้าจอ **(0,0)** คือ **x= 0** และ **y= 0**

2D Coordinate System

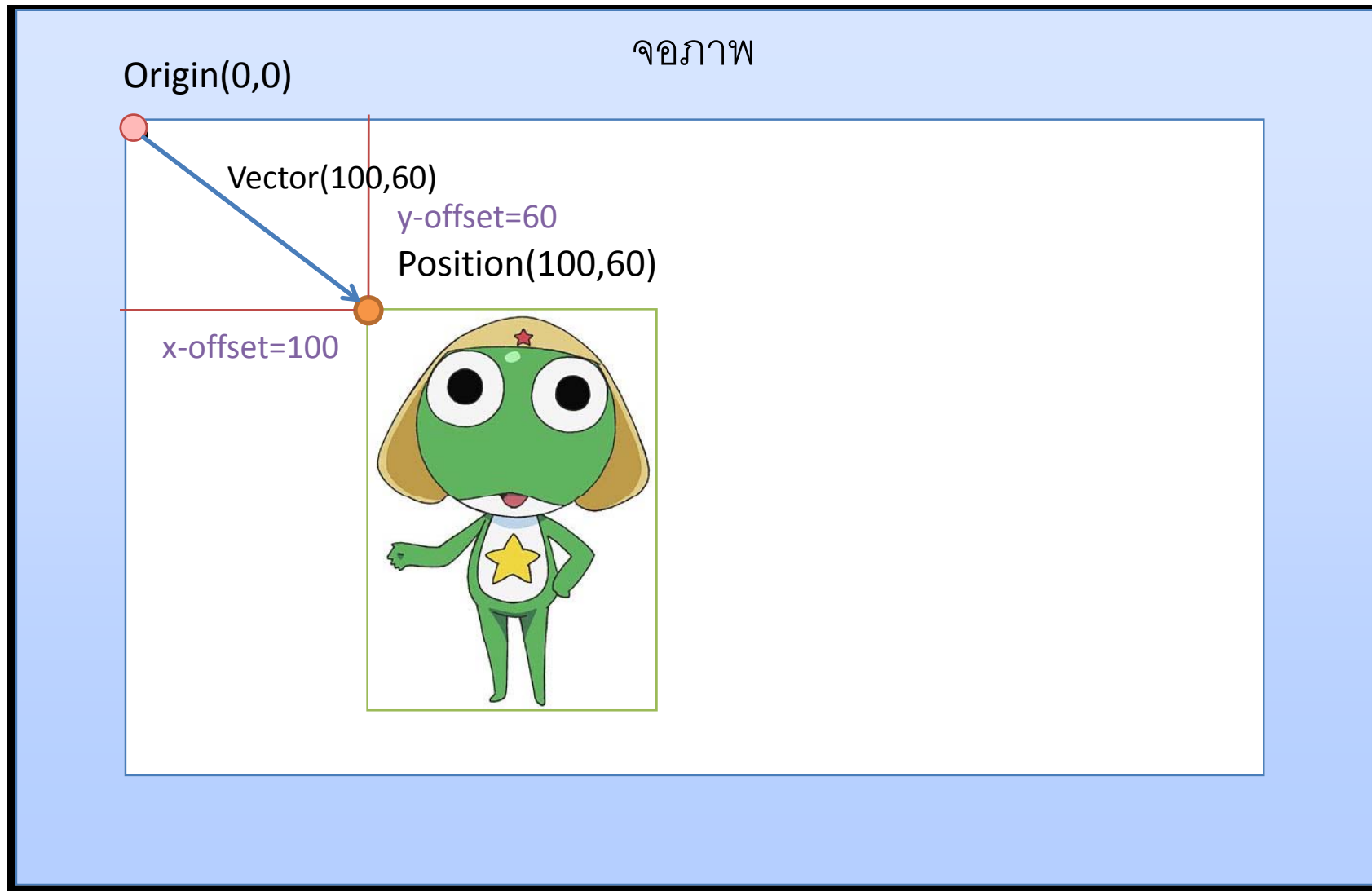
จอภาพ



2D Coordinate System

- จะเห็นได้ว่า หากหน้าจอมี Resolution 1024x768 จุดจะเริ่มจาก 0-1023 ในแนวแกน x และ 0-767 ในแนวแกน y และมุมขวาล่างสุดจะมีจุด (1023, 767)
- ซึ่งใน XNA จะเก็บค่าตำแหน่งเหล่านี้ในตัวแปร Vector2 และตัวแปรแกน x และ y ในรูปแบบ int

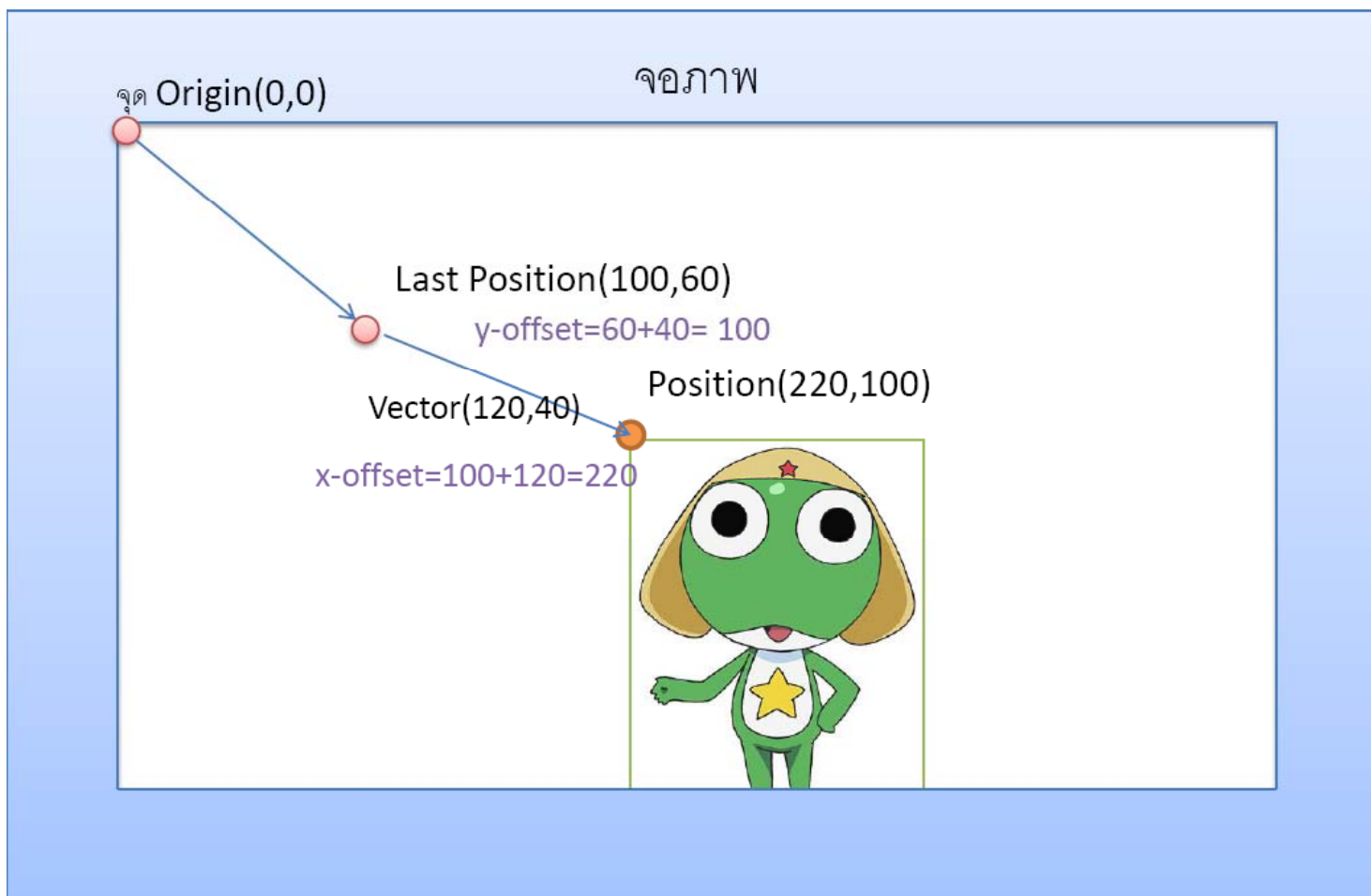
ตัวอย่างการวาดภาพในตำแหน่งที่ (100,60)



2D Transformation

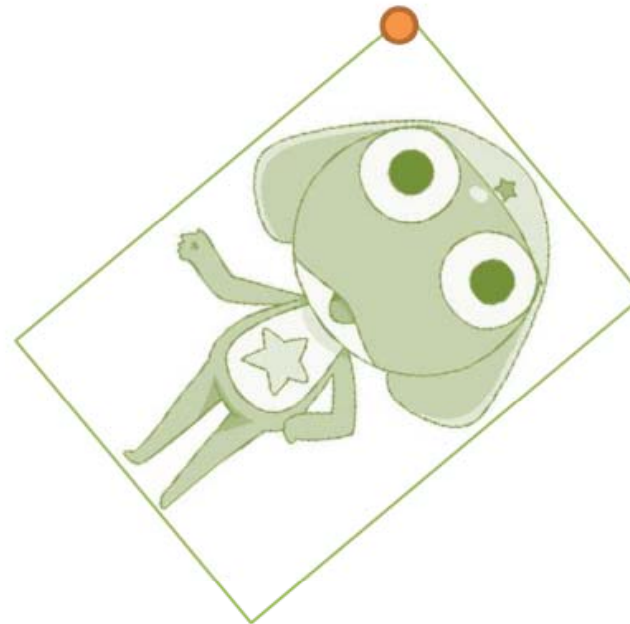
- ในโลก **2D** นั้นนอกจากตำแหน่งของภาพแล้ว ยังสามารถทำการย้ายตำแหน่งของภาพได้ รวมถึง ย่อ/ขยาย และ หมุนภาพได้ สิ่งเหล่านี้เรียกรวมๆ ว่า **“2D Transformation”**

หากต้องการเลื่อนจากจุดเดิมไปอีก **120, 40** จุด



การหมุน (Rotate)

- ใน XNA มีการสร้างจุด **Origin** ของ **Sprite** เพื่ออำนวยความสะดวกใน **Rotate** หากเราใช้จุดเดิมในการหมุนจะออกมาเป็นแบบนี้



การใช้จุด Origin ของ Sprite เพื่อใช้ในการหมุน

- ใน XNA สามารถกำหนดจุด Origin ของ Sprite ได้ว่าต้องการให้อยู่ในจุดใดของภาพเพื่อสะดวกในการหมุน



การย่อ/ขยาย(scale)

- การทำให้ **Sprite** มีขนาดใหญ่ขึ้นในการวาด ซึ่งใช้จุด **origin** ในการขยายเช่นกัน



Vector2

- Vector2 เป็นตัวแปรที่เก็บค่า 2 ตัว เรามักใช้กับการเก็บตำแหน่ง มี parameter ดังต่อไปนี้

```
Vector2 <ชื่อตัวแปร> = new Vector2(<จุด Position X>, <จุด Position Y>);
```


SpriteBatch

- ในเกมในปัจจุบันมีการใช้รูปภาพหลายรูปในหลายๆ ตำแหน่งของหน้าจอ มันเป็นเรื่องที่ยุ่งยากในการส่งผ่านข้อมูลของตำแหน่งและจัดการภาพเหล่านี้ให้มีประสิทธิภาพ
- โดย XNA มี Class พิเศษสำหรับจัดการนี้โดยตรงนั่นคือ **SpriteBatch**
- กำหนด **SpriteBatch** ไว้ใน **Class Game** ดังนี้

```
<private | public> SpriteBatch <ชื่อตัวแปร>;
```

เช่น `private SpriteBatch spriteBatch;`

- วางใน **LoadContent** ก็ได้ ให้ใส่การกำหนดค่าลงไปดังนี้

```
<ชื่อตัวแปร> = new SpriteBatch(<ตัวแปรGraphics>.GraphicsDevice);
```

เช่น

```
spriteBatch = new SpriteBatch(GraphicsDevice);
```

SpriteBatch

- การทำงานของ **Method SpriteBatch** ในส่วนการ **Draw** ใน **Draw Method** นั้นแบ่งเป็นหลาย **Method** ดังนี้
 - **Begin()** เป็นการบอกจุดเริ่มต้นของ **SpriteBatch** และยังมี การ **set** ค่าต่างๆ เช่น **SpriteBlendMode**, **SpriteSortMode**
 - **Draw()** ไว้สำหรับ **Draw** ภาพปกติโดยใช้ **Texture2D**
 - **DrawString()** ไว้สำหรับ **Draw** ข้อความที่ใช้ในเกม
 - **End()** เป็นการบอกจุดสิ้นสุดของ **SpriteBatch**

การ Draw ใน SpriteBatch

Method	คำอธิบาย
<code>Draw(Texture2D, Rectangle, Color)</code>	การวาดโดยระบุ Texture2D ที่ใช้, ตำแหน่งที่จะวางเป็นลักษณะของ Rectangle และสีที่ใช้
<code>Draw(Texture2D, Rectangle, Nullable<Rectangle>, Color)</code>	การวาดโดยระบุ Texture2D ที่ใช้, ตำแหน่งที่ภาพ ในรูปแบบ Rectangle, ตำแหน่งของภาพ ที่นำมาวาดเป็นตัวเลือกในรูปแบบ Rectangle และสีที่ใช้

การ Draw ใน SpriteBatch

Method	คำอธิบาย
<code>Draw(Texture2D, Rectangle, Nullable<Rectangle>, Color, float, Vector2, SpriteEffects, float)</code>	การวาดโดยระบุ Texture2D ที่ใช้, ตำแหน่งที่ภาพ ในรูปแบบ Rectangle, ตำแหน่งของภาพที่นำมาวาด เป็นตัวเลือกในรูปแบบ Rectangle, สีที่ใช้, ค่าสำหรับการหมุน, ตำแหน่ง origin ใหม่, Sprite Effects, และค่าความลึกของภาพในการจัดเรียง

การ Draw ใน SpriteBatch

Method	คำอธิบาย
Draw(Texture2D, Vector2, Color)	การวาดโดยระบุ Texture2D ที่ใช้, ตำแหน่งที่จะวางเป็นลักษณะของ Vector2 และสีที่ใช้
Draw(Texture2D, Vector2, Nullable<Rectangle>, Color)	การวาดโดยระบุ Texture2D ที่ใช้, ตำแหน่งที่จะวางเป็นลักษณะของ Vector2, ตำแหน่งของภาพที่นำมา วาด เป็นตัวเลือกและสีที่ใช้

การ Draw ใน SpriteBatch

Method	คำอธิบาย
<code>Draw (Texture2D, Vector2, Nullable<Rectangle>, Color, float, Vector2, float, SpriteEffects, float)</code>	การวาดโดยระบุ Texture2D ที่ใช้, ตำแหน่งที่วางเป็น Vector2, ตำแหน่งและขนาดของภาพที่วาดเป็นตัวเลือก, สีที่ใช้, การหมุน, จุด origin, ค่าย่อขยายเป็น Single, sprite effect และค่าความลึกของภาพในการจัดเรียง
<code>Draw (Texture2D, Vector2, Nullable<Rectangle>, Color, float, Vector2, Vector2, SpriteEffects, float)</code>	การวาดโดยระบุ Texture2D ที่ใช้, ตำแหน่งที่จะวางเป็นลักษณะของ Vector2, ตำแหน่งของภาพที่นำมาวาดเป็นตัวเลือกในรูปแบบ Rectangle, สีที่ใช้, การหมุน, จุด origin, ค่าย่อขยายเป็น Vector2, sprite effects, และ ค่าความลึกของภาพในการจัดเรียง

SpriteEffects

- เป็น **Effects** ของ **Sprite** ที่ทำได้ในคำสั่ง **Draw** มีดังต่อไปนี้
 - **SpriteEffects.FlipHorizontally** ไว้พลิกด้านแนวนอน
 - **SpriteEffects.FlipVertically** ไว้พลิกด้านแนวตั้ง
 - **SpriteEffects.None** ไม่มี Effect

การใช้ SpriteFont ใน XNA

- ใน XNA มีวิธีการแนบ Font จาก Windows เข้ามาในเกมได้เลย โดยสามารถใช้ TrueType Font ในเกม
- การ Add SpriteFont เข้ามายัง project ก่อนมีขั้นตอนดังนี้
 - Click ขวาที่ Folder Content ใน Solution Explorer
 - เลือกที่ Add > Add New Item เลือก SpriteFont แล้วตั้งชื่อให้ตรงกับ font แต่ไม่ควรมี space แล้วตามด้วย Size เช่น Angsana14
 - จากนั้นให้เปิด sprite font แล้วดูที่บรรทัดตรงที่เขียนว่า <FontName>

<FontName>ชื่อFont</FontName>

เช่น <FontName>AngsanaUPC</FontName>

- จากนั้น ให้ดู บรรทัด <Size>ขนาด</Size> แก้ขนาดตามต้องการ

การใช้ SpriteFont ใน XNA

- สร้างตัวแปร SpriteFont ไว้ใน Class Game1 ดังนี้

```
private SpriteFont font;
```

- ให้ใส่ Code ลงใน Method LoadContent

```
font = Content.Load<SpriteFont>(@"AngsanaUPC14");
```

DrawString

Method

คำอธิบาย

DrawString
(SpriteFont, String,
Vector2, Color)

การวาดโดยระบุ Sprite Font, ข้อความที่
แสดง, Vector2 สำหรับตำแหน่ง และสีของ
Font

DrawString(SpriteFont,
String, Vector2, Color,
float, Vector2, float,
SpriteEffects, float)

การวาดโดยระบุ Sprite Font, ข้อความที่
แสดง, Vector2 สำหรับตำแหน่ง, สีของ
Font, การหมุน, จุด origin, ค่าย่อขยาย
เป็น float, Sprite Effect และ ค่าความ
ลึกของภาพในการจัดเรียง

DrawString

Method

คำอธิบาย

`DrawString(SpriteFont, String, Vector2, Color, float, Vector2, Vector2, SpriteEffects, float)`

การวาดโดยระบุ `Sprite Font`, ข้อความที่จะแสดง, `Vector2` สำหรับตำแหน่ง, สีของ `Font`, การหมุน, จุด `Origin`, ค่าย่อขยายเป็น `Vector2`, `Sprite Effect` และ ค่าความลึกของภาพในการจัดเรียง

การใช้ภาษาไทยใน XNA

- หากต้องการใช้ภาษาไทยให้เพิ่มข้อความในกรอบเหล่านี้ลงไปไฟล์ของ **SpriteFont** เพื่อเป็นการเพิ่มช่วงของภาษาไทยลงไป

```
<CharacterRegions>
```

```
<CharacterRegion>
```

```
<Start>#32;</Start>
```

```
<End>#126;</End>
```

```
</CharacterRegion>
```

```
<CharacterRegion>
```

```
<Start>#3584;</Start>
```

```
<End>#3711;</End>
```

```
</CharacterRegion>
```

```
</CharacterRegions>
```