แบบฝึกหัดเรื่องเวกเตอร์ สำหรับนักศึกษาปี 1 CS

ภาพรวมกิจกรรม

ส่วน	เครื่องมือ	เวลาโดยประมาณ	เป้าหมาย
Part 1	GeoGebra	15–20 นาที	เข้าใจเวกเตอร์เชิงภาพ: ขนาด ทิศทาง การบวก และการคูณด้วยสเกลาร์
Part 2	Python / NumPy	20–25 นาที	ใช้เวกเตอร์แทนข้อมูลและคำนวณด้วยโค้ด
Part 3	Unity / C#	20–25 นาที	ใช้เวกเตอร์แทนตำแหน่ง ทิศทาง และการเคลื่อนที่ในเกม

Part 1: GeoGebra Concept Exercise

เวลา: 15–20 นาที

จุดประสงค์

หลังทำกิจกรรมนี้ นักศึกษาควรอธิบายได้ว่า

- เวกเตอร์มีขนาดและทิศทาง
- เวกเตอร์ $[2, 4]$ หมายถึงเคลื่อนที่ตามแกน x 2 หน่วย และแกน y 4 หน่วย
- การบวกเวกเตอร์คือการรวมการเคลื่อนที่
- การคูณเวกเตอร์ด้วยสเกลาร์ทำให้เวกเตอร์ยาวขึ้น สั้นลง หรือกลับทิศทาง

กิจกรรม 1.1: สร้างเวกเตอร์จากจุดสองจุด

คำสั่ง

1. เปิด GeoGebra Graphing Calculator
2. สร้างจุด

$$A = (0, 0)$$

และ

$$B = (2, 4)$$

1. สร้างเวกเตอร์จาก A ไป B

$$\vec{AB}$$

1. สังเกตทิศทางและความยาวของเวกเตอร์

คำถาม

1. เวกเตอร์ $\vec{AB}$ เคลื่อนที่ไปทางแกน x กี่หน่วย?
2. เวกเตอร์ $\vec{AB}$ เคลื่อนที่ไปทางแกน y กี่หน่วย?
3. เวกเตอร์นี้เขียนในรูป list ได้อย่างไร?

คำตอบที่คาดหวัง

$$\vec{AB} = [2, 4]$$

หมายถึงไปทางแกน x 2 หน่วย และไปทางแกน y 4 หน่วย

กิจกรรม 1.2: เปลี่ยนตำแหน่งปลายเวกเตอร์

คำสั่ง

เปลี่ยนจุด B เป็น

$$B = (3, 5)$$

แล้วสังเกตเวกเตอร์ใหม่

คำถาม

1. เวกเตอร์ใหม่คืออะไร?
2. เวกเตอร์ใหม่ยาวขึ้นหรือสั้นลงเมื่อเทียบกับ $[2, 4]$?
3. ทิศทางเปลี่ยนหรือไม่?

คำตอบที่คาดหวัง

$$\vec{AB} = [3, 5]$$

เวกเตอร์เปลี่ยนทั้งขนาดและทิศทางเล็กน้อย เพราะค่าตามแกน x และ y เปลี่ยน

กิจกรรม 1.3: การบวกเวกเตอร์

คำสั่ง

สร้างเวกเตอร์ 2 ตัว

$$\vec{u} = [2, 4]$$

$$\vec{v} = [3, 1]$$

จากนั้นสร้างผลบวก

$$\vec{u} + \vec{v}$$

คำถาม

1. ผลบวกของเวกเตอร์คืออะไร?
2. ถ้าคิดแบบการเคลื่อนที่ หมายความว่าอย่างไร?
3. ผลลัพธ์สุดท้ายอยู่ที่จุดใด ถ้าเริ่มจาก $(0, 0)$?

คำตอบที่คาดหวัง

$$\vec{u} + \vec{v} = [2 + 3,\ 4 + 1]$$

$$\vec{u} + \vec{v} = [5, 5]$$

ถ้าเริ่มจาก $(0, 0)$ จะไปสิ้นสุดที่จุด $(5, 5)$

กิจกรรม 1.4: การคูณเวกเตอร์ด้วยสเกลาร์

คำสั่ง

ใช้เวกเตอร์

$$\vec{u} = [2, 4]$$

แล้วสร้างเวกเตอร์ต่อไปนี้

$$2\vec{u}$$

$$0.5\vec{u}$$

$$-\vec{u}$$

คำถาม

1. $2\vec{u}$ มีขนาดเปลี่ยนอย่างไร?
2. $0.5\vec{u}$ มีขนาดเปลี่ยนอย่างไร?

3. $-\vec{u}$ มีทิศทางอย่างไร?

คำตอบที่คาดหวัง

$$2\vec{u} = [4, 8]$$

ยาวขึ้น 2 เท่า

$$0.5\vec{u} = [1, 2]$$

สั้นลงครึ่งหนึ่ง

$$-\vec{u} = [-2, -4]$$

มีทิศทางตรงข้ามกับเวกเตอร์เดิม

Part 2: Python / Data Science Exercise

เวลา: 20–25 นาที

จุดประสงค์

หลังทำกิจกรรมนี้ นักศึกษาควรอธิบายได้ว่า

- เวกเตอร์ใน Python สามารถแทนข้อมูลหนึ่งรายการได้
- NumPy array ใช้แทนเวกเตอร์ได้
- สามารถบวกเวกเตอร์ คำนวณ dot product และหาขนาดของเวกเตอร์ได้
- เวกเตอร์ใน Data Science มักใช้เป็น feature vector

กิจกรรม 2.1: สร้างเวกเตอร์ข้อมูลนักศึกษา

คำสั่ง

ให้นักศึกษารันโค้ดนี้

```
import numpy as np

student = np.array([70, 165, 3, 85])

print(student)
```

โดยกำหนดให้ข้อมูลหมายถึง

[น้ำหนัก, ส่วนสูง, จำนวนวิชา, คะแนนสอบ]

คำถาม

1. เวกเตอร์นี้มีทั้งหมดกี่ค่า?
2. ค่า 165 หมายถึงอะไร?
3. ค่า 85 หมายถึงอะไร?
4. เวกเตอร์นี้แทนการเคลื่อนที่หรือแทนข้อมูล?

คำตอบที่คาดหวัง

เวกเตอร์นี้มี 4 ค่า
ค่า 165 หมายถึงส่วนสูง
ค่า 85 หมายถึงคะแนนสอบ
เวกเตอร์นี้แทนข้อมูล ไม่ใช่การเคลื่อนที่

กิจกรรม 2.2: เปรียบเทียบข้อมูลนักศึกษา 2 คน

คำสั่ง

ให้นักศึกษารันโค้ดนี้

```
import numpy as np

student_a = np.array([70, 165, 3, 85])
student_b = np.array([60, 160, 4, 90])

print("Student A:", student_a)
print("Student B:", student_b)
```

คำถาม

1. นักศึกษาคนที่ A มีคะแนนสอบเท่าไร?
2. นักศึกษาคนที่ B ลงทะเบียนกี่วิชา?
3. เวกเตอร์ทั้งสองมีจำนวนมิติเท่ากันหรือไม่?
4. ทำไมจำนวนมิติของเวกเตอร์ต้องเท่ากันเมื่อต้องการนำมาบวกกัน?

คำตอบที่คาดหวัง

Student A มีคะแนนสอบ 85
Student B ลงทะเบียน 4 วิชา
เวกเตอร์ทั้งสองมี 4 มิติเท่ากัน
ถ้าจะบวกเวกเตอร์ ต้องบวกค่าที่อยู่ตำแหน่งเดียวกัน ดังนั้นจำนวนมิติต้องเท่ากัน

กิจกรรม 2.3: การบวกเวกเตอร์ใน NumPy

คำสั่ง

```
import numpy as np

a = np.array([2, 4])
b = np.array([3, 5])

result = a + b

print(result)
```

คำถาม

1. ผลลัพธ์คืออะไร?
2. ผลลัพธ์เหมือนกับที่เห็นใน GeoGebra หรือไม่?
3. การบวกนี้เป็นการบวกแบบ component-wise หรือไม่?

คำตอบที่คาดหวัง

$$[2, 4] + [3, 5] = [5, 9]$$

ใช่ เป็นการบวกทีละตำแหน่ง

กิจกรรม 2.4: Dot Product และ Magnitude

คำสั่ง

```
import numpy as np

x = np.array([3, 4])
y = np.array([1, 2])

dot_product = np.dot(x, y)
magnitude_x = np.linalg.norm(x)

print("Dot product:", dot_product)
print("Magnitude of x:", magnitude_x)
```

คำถาม

1. ค่า dot product เท่ากับเท่าไร?
2. ขนาดของเวกเตอร์ $x = [3, 4]$ เท่ากับเท่าไร?

3. ทำไมขนาดของ $[3, 4]$ จึงเท่ากับ 5?
4. Dot product ให้ผลลัพธ์เป็น scalar หรือ vector?

คำตอบที่คาดหวัง

$$x \cdot y = 3(1) + 4(2) = 11$$

$$|x| = \sqrt{3^2 + 4^2} = 5$$

Dot product ให้ผลลัพธ์เป็น scalar

กิจกรรม 2.5: Mini Challenge

คำสั่ง

ให้นักศึกษาสร้างเวกเตอร์ของตนเอง 2 ตัว เช่น

```
a = np.array([4, 2])
b = np.array([1, 6])
```

แล้วคำนวณ

1. $a + b$
2. $a \cdot b$
3. $|a|$
4. $|b|$

คำตอบตัวอย่าง

ถ้า

$$a = [4, 2]$$

และ

$$b = [1, 6]$$

จะได้

$$a + b = [5, 8]$$

$$a \cdot b = 4(1) + 2(6) = 16$$

$$|a| = \sqrt{4^2 + 2^2} = \sqrt{20}$$

$$|b| = \sqrt{1^2 + 6^2} = \sqrt{37}$$

Part 3: Unity / Game Development Exercise

เวลา: 20–25 นาที

จุดประสงค์

หลังทำกิจกรรมนี้ นักศึกษาควรอธิบายได้ว่า

- `Vector3` ใช้แทนตำแหน่งหรือทิศทางใน Unity
- การบวกเวกเตอร์สามารถใช้เปลี่ยนตำแหน่งวัตถุได้
- `Vector3(1, 0, 0)` หมายถึงการเคลื่อนที่ตามแกน x
- แนวคิดเวกเตอร์ใน Unity เชื่อมโยงกับ GeoGebra และ Python ได้

กิจกรรม 3.1: เข้าใจ Vector3

คำอธิบาย

ใน Unity เวกเตอร์ 3 มิติใช้รูปแบบ

```
Vector3(x, y, z)
```

โดย

- x: ซ้าย-ขวา
- y: ขึ้น-ลง
- z: หน้า-หลัง

คำถาม

1. `Vector3(1, 0, 0)` หมายถึงทิศทางใด?
2. `Vector3(0, 1, 0)` หมายถึงทิศทางใด?
3. `Vector3(0, 0, 1)` หมายถึงทิศทางใด?

คำตอบที่คาดหวัง

`Vector3(1, 0, 0)` คือทิศทางตามแกน x
`Vector3(0, 1, 0)` คือทิศทางตามแกน y
`Vector3(0, 0, 1)` คือทิศทางตามแกน z

กิจกรรม 3.2: สร้างสคริปต์ให้วัตถุเคลื่อนที่

คำสั่ง

1. สร้าง Cube ใน Unity
2. สร้าง C# Script ชื่อ `VectorMoveDemo`
3. ใส่โค้ดนี้

```
using UnityEngine;

public class VectorMoveDemo : MonoBehaviour
{
    void Start()
    {
        Vector3 moveDirection = new Vector3(1, 0, 0);

        transform.position = transform.position + moveDirection;
    }
}
```

1. นำ script ไปใส่ใน Cube
2. กด Play

คำถาม

1. Cube เคลื่อนที่ไปทางแกนใด?
2. ตำแหน่งใหม่ของ Cube เปลี่ยนจากเดิมอย่างไร?
3. บรรทัดนี้หมายความว่าอะไร?

```
transform.position = transform.position + moveDirection;
```

คำตอบที่คาดหวัง

Cube เคลื่อนที่ไปทางแกน x จำนวน 1 หน่วย

บรรทัดนี้หมายถึง

$$\text{ตำแหน่งใหม่} = \text{ตำแหน่งเดิม} + \text{เวกเตอร์การเคลื่อนที่}$$

กิจกรรม 3.3: ทดลองเปลี่ยนทิศทาง

คำสั่ง

ให้นักศึกษาเปลี่ยนค่า `moveDirection`

รอบที่ 1:

```
Vector3 moveDirection = new Vector3(0, 1, 0);
```

รอบที่ 2:

```
Vector3 moveDirection = new Vector3(0, 0, 1);
```

รอบที่ 3:

```
Vector3 moveDirection = new Vector3(1, 1, 0);
```

คำถาม

1. รอบที่ 1 Cube เคลื่อนที่ไปทางใด?
2. รอบที่ 2 Cube เคลื่อนที่ไปทางใด?
3. รอบที่ 3 Cube เคลื่อนที่แบบใด?
4. ค่า `Vector3(1, 1, 0)` คล้ายกับการบวกเวกเตอร์ใน GeoGebra อย่างไร?

คำตอบที่คาดหวัง

รอบที่ 1 เคลื่อนที่ขึ้นตามแกน y
รอบที่ 2 เคลื่อนที่ไปตามแกน z
รอบที่ 3 เคลื่อนที่แนวทแยงระหว่างแกน x และ y

กิจกรรม 3.4: เคลื่อนที่ต่อเนื่องด้วย Time.deltaTime

คำสั่ง

เปลี่ยนโค้ดเป็นแบบนี้

```
using UnityEngine;

public class VectorMoveDemo : MonoBehaviour
{
    public float speed = 2.0f;

    void Update()
    {
        Vector3 moveDirection = new Vector3(1, 0, 0);
```

```
        transform.position = transform.position + moveDirection * speed *
Time.deltaTime;
    }
}
```

คำถาม

1. `Update()` ต่างจาก `Start()` อย่างไร?
2. `speed` มีหน้าที่อะไร?
3. `Time.deltaTime` ช่วยเรื่องอะไร?
4. ถ้าเปลี่ยน `speed` จาก 2 เป็น 5 จะเกิดอะไรขึ้น?

คำตอบที่คาดหวัง

`Start()` ทำงานครั้งเดียวตอนเริ่มเกม
`Update()` ทำงานซ้ำทุก frame
`speed` ควบคุมความเร็ว
`Time.deltaTime` ทำให้การเคลื่อนที่ราบรื่นและสัมพันธ์กับเวลา
ถ้าเพิ่ม speed วัตถุจะเคลื่อนที่เร็วขึ้น

กิจกรรม 3.5: Mini Challenge

คำสั่ง

ให้นักศึกษาแก้โค้ดเพื่อให้ Cube เคลื่อนที่แนวทแยงไปข้างหน้าและขวา

ใช้เวกเตอร์

```
Vector3 moveDirection = new Vector3(1, 0, 1);
```

คำถาม

1. Cube เคลื่อนที่ตามแกนใดบ้าง?
2. ถ้าต้องการให้ Cube เคลื่อนที่ขึ้นด้วย ต้องเปลี่ยนเป็นเวกเตอร์อะไร?
3. ถ้าต้องการให้ Cube เคลื่อนที่กลับทิศทาง ต้องทำอย่างไร?

คำตอบที่คาดหวัง

Cube เคลื่อนที่ตามแกน x และ z

ถ้าต้องการให้ขึ้นด้วย อาจใช้

```
Vector3 moveDirection = new Vector3(1, 1, 1);
```

ถ้าต้องการให้กลับทิศทาง ใช้ค่าติดลบ เช่น

```
Vector3 moveDirection = new Vector3(-1, 0, -1);
```

สรุปหลังทำกิจกรรม

การเชื่อมโยงทั้ง 3 ส่วน

เครื่องมือ	สิ่งที่เห็น	ความหมายของเวกเตอร์
GeoGebra	ลูกศรบนกราฟ	ขนาดและทิศทาง
Python	NumPy array	ข้อมูลหรือ feature vector
Unity	Vector3	ตำแหน่ง ทิศทาง และการเคลื่อนที่

คำถามสะท้อนคิด

1. เวกเตอร์ใน GeoGebra และ Python เหมือนหรือต่างกันอย่างไร?
2. เวกเตอร์ใน Python และ Unity ใช้แทนสิ่งเดียวกันเสมอหรือไม่?
3. ทำไม CS student ต้องเข้าใจเวกเตอร์ก่อนเรียน AI, Game, Graphics หรือ Data Science?

คำตอบแนวคิด

เวกเตอร์มีรูปแบบเป็นชุดของตัวเลขเหมือนกัน แต่ความหมายขึ้นอยู่กับบริบท

- ใน GeoGebra เวกเตอร์คือภาพของขนาดและทิศทาง
- ใน Python เวกเตอร์คือข้อมูล
- ใน Unity เวกเตอร์คือตำแหน่งหรือการเคลื่อนที่

แนวคิดกลางคือ

$$\text{Vector} = \text{ชุดของตัวเลขที่มีความหมายตามบริบท}$$