

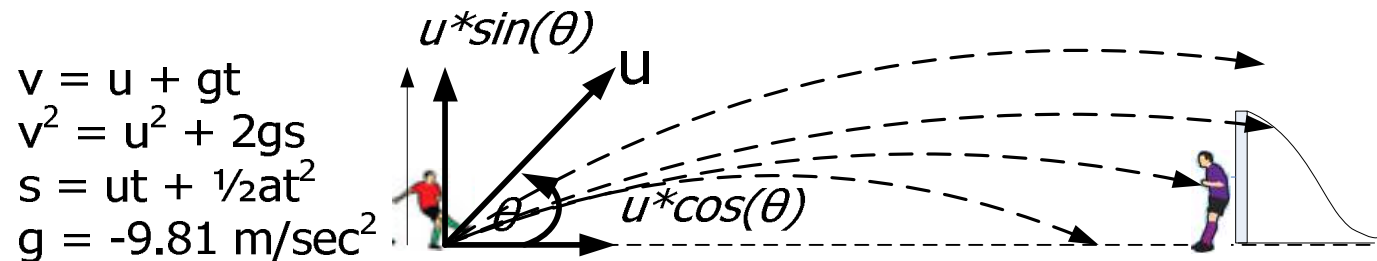
แผนการสอนปฏิบัติครั้งที่ 4

✚ **วัตถุประสงค์** เข้าใจการเขียนโปรแกรมแก้ปัญหาด้วยสูตร และผังงาน

✚ **โจทย์ปัญหา** ลูกฟุตบอลถูกเตะลอยเพื่อยิงประตูในแนวตรง ในทิศทำมุม θ (องศา) กับแนวระดับ ด้วยความเร็ว u เมตร/วินาที ระยะห่างจากประตู x เมตร โดยไม่คิดแรงต้านของลม และขนาดของลูกบอล **เสาประตูมีขนาดกว้าง x สูง 7.32×2.44 เมตร** ถ้าผู้รักษาประตูมีเวลามากกว่า **0.5 วินาที** จะสามารถป้องกันประตูไว้ได้ จงเขียนโปรแกรมเพื่อคำนวณข้อมูลของลูกบอล และผลการยิงประตูครั้งนี้

✚ **ผลลัพธ์การเรียนรู้**

- นักศึกษาสามารถเขียนผังงาน และโปรแกรม แก้ปัญหาตามสูตรที่วิเคราะห์ไว้เองได้



✚ **ขั้นตอนการวิเคราะห์**

- หาสูตรคำนวณความสูงของลูกบอลที่ตำแหน่งประตู และเวลาในการเคลื่อนที่ถึงประตู
- กำหนดเงื่อนไขการเข้า/ไม่เข้าประตู แบบต่างๆ สร้างผังงานของโปรแกรม
- แปลงผังงานเป็นโปรแกรม

✚ **ข้อมูลที่ต้องการแสดงผล**

1. ที่ตำแหน่งประตู

- กรณีที่ลูกบอลยังไม่ตกถึงพื้น ให้แสดงผลว่าลูกบอลลอยสูงจากพื้นเท่าไร ใช้เวลาเท่าไร
- กรณีที่ลูกบอลตกถึงพื้นก่อน ให้แสดงผลเป็นระยะก่อนถึงประตู และเวลาที่ใช้

2. สรุปว่าลูกบอลจะเข้าประตูหรือไม่ (เข้า : ยังไม่ตกถึงพื้น, ความสูง $< 2.44\text{m}$, เวลา $\leq 0.5\text{s}$)

```
if (Smax < Sgoal)
    printf("ToT Fail ball\n");
else if (Hgoal >= 2.44)
    printf("ToT Overshoot\n");
else if (Tgoal > 0.5)
    printf("ToT Save ball!\n");
else printf("Goal! \\^o^/ \n");
```

ตัวอย่างการทดสอบ (Test Case)

- + ข้อมูลหน้าจอที่ต้องส่ง หน้าจอแสดงผลคำตอบ 4 กรณี
 1. ทดสอบที่ความเร็ว 10 เมตร/วินาที มุม 10 องศา ระยะห่าง 10 เมตร (ลูกตกพื้นก่อน)
 - ที่ตำแหน่งเส้นประตก (ให้แสดงผลเป็นลูกบอลตกพื้นก่อนถึงประตก **6.51m, 0.35 sec**)
 - สรุป ไม่เข้า ลูกบอลตกพื้นก่อน
 2. ทดสอบที่ความเร็ว 25 เมตร/วินาที มุม 30 องศา ระยะห่าง 10 เมตร (ยิงโด่ง)
 - ที่ตำแหน่งเส้นประตก (**4.73 เมตร 0.46 sec**)
 - สรุป ไม่เข้า ลูกบอลข้ามประตก
 3. ทดสอบที่ความเร็ว 20 เมตร/วินาที มุม 20 องศา ระยะห่าง 20 เมตร (ยิงไม่แรงพอ)
 - ที่ตำแหน่งเส้นประตก (**1.72 m , 1.06 sec**)
 - สรุป ไม่เข้า ผู้รักษาประตูรับได้
 4. ทดสอบที่ความเร็ว 25 เมตร/วินาที มุม 10 องศา ระยะห่าง 10 เมตร (เข้าประตก)
 - ที่ตำแหน่งเส้นประตก (**0.95 m, 0.41 sec**)
 - สรุป ลูกบอลเข้าประตก
- + งานที่ทำเพิ่ม (หลังจากโปรแกรมทำงานได้ถูกต้องแล้ว)
 - เมื่อจบแต่ละครั้ง ให้ตั้งคำถามเพื่อเริ่มคำนวณใหม่อีกครั้ง จนกว่าผู้ใช้ต้องการเลิก
 - ปรับปรุง User Interface ให้โปรแกรมน่าใช้
 - ต้องมีการป้องกันความผิดพลาดจากการป้อนข้อมูลผิดด้วย
- + รายงานที่ต้องส่งตามหลัง (ต้องมีการประเมินตนเองด้วย)
 - ผังงานและคำอธิบายของโปรแกรมที่เขียนขึ้น พร้อมหน้าจอทดสอบ(4 กรณี)