

## PMMD - Plus Minus Multiple Divide

### วัตถุประสงค์

Application นี้คือ Application สำหรับเล่นเกมบวก ลบ คูณ และหารเลข โดยมีวัตถุประสงค์คือเพื่อฝึกฝน ทดสอบ และให้ความบันเทิงแก่ผู้ใช้

### วิธีการเล่น

1. ในตอนเริ่มโปรแกรมจะถามผู้เล่นว่าต้องการเล่นหรือไม่ โดยหากผู้เล่นต้องตอบ Y เพื่อเข้าสู่เกม หากต้องการออกให้ตอบ N แต่ถ้าผู้เล่นส่งคำตอบอื่นเข้ามา โปรแกรมจะให้ผู้เล่นตอบใหม่จนกว่าจะตอบ Y หรือ N
2. เมื่อเข้าสู่เกมแล้วผู้เล่นต้องเลือกระดับความยาก (Level) ก่อนเล่น โดยโปรแกรมจะมีความยากตั้งแต่ระดับที่ 2 ถึงระดับที่ 9 ซึ่งระดับความยากจะมีค่าเท่ากับจำนวนตัวเลขในคำถาม
3. ตัวเลขที่จะนำมาสร้างเป็นคำถามนั้นจะมีค่าตั้งแต่ 1 ถึง 9999 และ Operator ที่จะนำมาตั้งคำถามจะมีเพียง บวก ลบ คูณ และหาร
4. หลังจากเลือกระดับความยาก โปรแกรมก็จะแสดงคำถามเพื่อให้ผู้เล่นตอบ โดยจะสามารถตอบได้ครั้งเดียวเท่านั้น
  - 4.1. หากคำตอบเป็นทศนิยมต้องปัดให้เป็น 2 ตำแหน่ง
  - 4.2. ถ้าผู้เล่นตอบถูก จะได้คะแนนเพิ่ม 1 คะแนน
  - 4.3. ถ้าผู้เล่นตอบผิด จะเสียคะแนน 1 คะแนน
  - 4.4. ถ้าผู้เล่นขอข้ามไปยังข้อถัดไป จะเสียคะแนน 2 คะแนน
  - 4.5. ถ้าผู้เล่นตอบมาใน Format ที่ไม่ถูกต้อง เช่น ตอบเป็นตัวอักษรอื่นนอกจากคำว่า "Over" และ "Pass" ผู้ใช้จะต้องตอบใหม่จนกว่าจะตอบในรูปแบบของจำนวนเต็มหรือตัวเลขทศนิยม
5. หากผู้ใช้ต้องการขอข้ามไปยังข้อถัดไปให้ส่งคำว่า Pass แทนคำตอบ
6. หากผู้ใช้ต้องการออกจากโปรแกรมให้ส่งคำว่า Over แทนคำตอบ

## คุณลักษณะ

### Reliability (ความน่าเชื่อถือ)

การสื่อสารระหว่าง Server และ Client มีการตรวจสอบข้อมูลที่ส่งและรับเสมอ ทำให้มั่นใจได้ว่าข้อมูลจะไม่ถูกตัดแปลง หรือสูญหาย นอกจากนี้โปรแกรมยังออกแบบมาสำหรับให้บริการผู้ใช้เพียงคนเดียวจึงทำให้การรักษาข้อมูลการเล่นและคะแนนของผู้ใช้เป็นไปอย่างมีประสิทธิภาพ และน่าเชื่อถือ

### Delay (ความล่าช้า)

โปรแกรมตอบสนองต่อ Input ของผู้เล่นในเวลาเกือบเรียลไทม์ โดยการสื่อสารระหว่าง Server และ Client มีความล่าช้าต่ำ เพราะใช้การเชื่อมต่อ TCP ที่มีความเสถียรและรวดเร็ว

### Throughput (ปริมาณข้อมูลที่ส่งผ่าน)

ข้อมูลที่ส่งระหว่าง Server และ Client มีขนาดเล็ก และเนื่องจาก Server รองรับผู้ใช้เพียงคนเดียวจึงไม่จำเป็นต้องคำนึงถึงการประมวลผลหลาย Session พร้อมกัน

### Security (ความปลอดภัย)

การสื่อสารระหว่าง Server และ Client ใช้การเข้ารหัสเสมอทำให้สามารถป้องกันการดักฟังและการโจมตีจากผู้ไม่ประสงค์ดีได้ และโปรแกรมยังการตรวจสอบความถูกต้องของข้อมูลที่ส่งและรับ เพื่อป้องกันการตัดแปลงข้อมูล

## Transport Layer Service Model

ในโปรแกรมนี้อาศัย TCP เพราะต้องการให้ Client และ Server มีการ Contact กันเสมอเพราะต้องมีการโต้ตอบกันตลอดในระหว่างเล่น และไม่ต้องการให้ข้อมูลสูญหายในระหว่างส่งข้อมูล นอกจากนี้ TCP ยังมีมาตรการการเข้ารหัสและการตรวจสอบความถูกต้องทำให้มั่นใจได้ว่า Message ที่ส่งและรับจะมีความถูกต้อง

## การออกแบบ Protocol

ชื่อ: PMMD Service Protocol

### Status Code และ Status Phrase

1. 200: OK หมายถึง การร้องขอ (request) สำเร็จและ Server ตอบกลับด้วยข้อมูลที่ร้องขอ ซึ่ง Server จะส่งไปให้ Client ในกรณีที่การรับส่งข้อมูลเป็นไปอย่างถูกต้อง
2. 400: Bad Request หมายถึง การร้องขอไม่ถูกต้อง เช่น ส่งข้อมูลที่ไม่ถูกต้อง ซึ่ง Server จะส่งไปให้ Client ในกรณีที่ Client ส่ง Level ที่ไม่ถูกต้อง หรือ ส่งคำตอบที่ไม่ถูก Format มายัง Server

### Message Format

#### Server

<Header>/<Body> โดย Header คือ <Status code>,<Status phrase> โดย Body 3 แบบคือ

1. ข้อความในรูปแบบ "คำถาม,ผลลัพธ์การตรวจสอบคำตอบ,คะแนน,เลขของข้อที่แล้ว" ในกรณีที่ Server ส่งคำถามไปยัง Client ในสถานการณ์ปกติ
2. ข้อความแจ้งข้อผิดพลาด ในกรณีที่ Server ได้รับ Message ที่ไม่ถูกต้องจาก Client

#### Client

<Body> คือ ในส่วนของ Client สามารถส่งได้โดยไม่ต้องมี Header

### ลำดับการทำงานระหว่าง Server และ Client

1. Client ส่ง Level ให้ Server: โดยจะส่งในรูปแบบ "LEVEL:ระดับ(2-9)" ให้กับ server
2. Server รับ Level จาก Client มา: Server จะตรวจสอบว่า Message ที่รับมามี Format ตรงกับ "LEVEL:ระดับ(2-9)" หรือไม่ หากไม่ตรงจะส่งข้อความผิดพลาดกลับไป และวนซ้ำข้อ 1-2 จนกว่า Server จะได้รับ Message ที่ถูกต้อง
3. Server สร้างคำถามแรกที่มีจำนวนตัวเลขเท่ากับ Level และให้คะแนนเริ่มต้นอยู่ที่ 0 คะแนน: สร้างคำถามแรกโดยสุ่มเลขตั้งแต่ 0 ถึง 9999 จำนวนเท่ากับ Level และคำตอบ

4. Server ส่ง Message ให้ Client โดยมีลักษณะดังนี้ "คำถาม,ผลลัพธ์การตรวจสอบคำตอบ,คะแนน,เฉลยของข้อที่แล้ว" โดยคำถามแรก ผลลัพธ์การตรวจสอบคำตอบและเฉลยของข้อที่แล้วจะเป็น "-":
5. Client ได้รับ Message จาก Server และ รอ Input จาก User
6. Client ส่ง Message กลับไปยัง Server โดยถ้า Message คือ Over Client จะหยุดและปิดการเชื่อมต่อกับ Server หลังจากการส่ง Message
7. Server ได้รับ Message จาก Client แล้วนำไปตรวจสอบคำตอบ:
  - 7.1. ถ้า Server ได้รับคำว่า "Over" Server จะปิดการเชื่อมต่อกับ Client
  - 7.2. ถ้า Server ได้รับคำว่า "Pass" Server ลบคะแนน 2 คะแนน เนื่องจากเป็นการข้ามคำถาม และสร้าง Message คำถามใหม่
  - 7.3. ถ้า Server ได้ Message ซึ่งไม่จัดว่าเป็นจำนวนเต็มหรือเลขทศนิยม (ยกเว้น "Over" และ "Pass") Server จะสร้าง Message คำถามเดิม โดยผลลัพธ์การตรวจสอบจะระบุว่ารูปแบบไม่ถูกต้องและเฉลยเป็น "-" ถือว่า Message ที่ได้รับมาไม่ถูกต้อง
  - 7.4. ถ้า Server ได้รับคำตอบที่ถูก Server จะเพิ่มคะแนน 1 คะแนน และสร้าง Message คำถามใหม่
  - 7.5. ถ้า Server ได้รับคำตอบที่ผิด Server จะลบคะแนน 1 คะแนน และสร้าง Message คำถามใหม่
8. ถ้า Server ยังไม่ปิด Server จะส่ง Message ให้ Client โดยมีลักษณะดังนี้ "คำถาม,ผลลัพธ์การตรวจสอบคำตอบ,คะแนน,เฉลยของข้อที่แล้ว"
9. ทำซ้ำข้อ 5-8

### ผู้จัดทำ

นางสาวจิตรลดา โยธินทะ รหัส 6510405377