ปัญญาประดิษฐ์ในเกมเอแมท

ARTIFICIAL INTELLIGENCE IN A-MATH GAME

นายธานี	บัวเทศ	รหัส 48362292
นางสาวกิตติยา	ขุมทอง	รหัส 48364647
นายธิติวุฒิ	สำเภาเงิน	รหัส 48364746

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 5 เม.ย. 2553
เลขทะเบียน 1 499 3287
เลขเรียกหนังสือ ฟร.
มหาวิทยาลัยนเรศวร ส514 ป
2551

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2551

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	**ปัญญาประดิษฐ์ในเกมเอแมท**		
ผู้ดำเนินโครงการ	**นายธานี**	**บัวเทศ**	**รหัส 48362292**
	นางสาวกิตติยา	ขุมทอง	รหัส 48364647
	นายธิติวุฒิ	สำเภาเงิน	รหัส 48364746
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ	ริยะมงคล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2551		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ

(ดร.ไพศาล มุณีสว่าง)

.....กรรมการ

(อาจารย์เศรษฐา ตั้งค้าวานิช)

หัวข้อโครงงาน	ปัญญาประดิษฐ์ในเกมเอแมท		
ผู้ดำเนินโครงการ	นายธานี	บัวเทศ	รหัส 48362292
	นางสาวกิตติยา	ขุมทอง	รหัส 48364647
	นายธิติวุฒิ	สำเภาเงิน	รหัส 48364746
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ	ริยะมงคล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2551		

บทคัดย่อ

โครงการนี้ได้พัฒนาโปรแกรมประเภทปัญญาประดิษฐ์ (Artificial Intelligence) เพื่อเล่นเกมเอแมทกับผู้เล่น โดยโปรแกรมที่พัฒนานี้จะมีลักษณะผลัดกันเล่นระหว่างผู้เล่นกับคอมพิวเตอร์ ซึ่งในส่วนการทำงานของคอมพิวเตอร์ ได้นำหลักการของอัลกอริทึมการค้นหาแบบลำดับ (Linear Search Algorithm) และอัลกอริทึมแบบย้อนรอย (Backtracking Algorithm) มาใช้ในการคำนวณสมการและลงสมการในกระดาน จากการทดลองพบว่าคอมพิวเตอร์สามารถทำงานตามเงื่อนไขการลงสมการที่ได้ออกแบบไว้ได้ทุกกรณี

Project title	ARTIFICIAL INTELLIGENCE IN A-MATH GAME			
Name	Mr.Tanee	Buateat	ID.	48362292
	Miss.Kittiya	Khumtong	ID.	48364647
	Mr.Thitiwut	Sampaongearn	ID.	48364746
Project Advisor	Panomkhawn Riyamongkol, Ph.D.			
Major	Computer Engineering			
Department	Electrical and Computer Engineering			
Academic Year	2008			

Abstract

This project develops a program for playing A-math game between a player and a computer. This program is turn base game. The solution for finding equation and playing with player applies Linear Search Algorithm and Backtracking Algorithm. Our experiments show that the developed program can play and each number of equation.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์สำเร็จได้ดี เนื่องด้วยความอนุเคราะห์จากอาจารย์ท่านที่ปรึกษา ดร.พนมขวัญ ริยะมงคล ที่กรุณาให้คำปรึกษา แนะนำวิธีการในการทำงานพร้อมทั้งเสนอแนะแนวทางการแก้ไขตลอดระยะเวลาการทำโครงงาน สุดท้ายต้องขอขอบพระคุณอาจารย์ทุกท่านและเพื่อนๆ พี่ๆ ทุกคนที่ยังไม่ได้เอ่ยนามที่คอยให้คำแนะนำในด้านต่างๆและสนับสนุนในการทำโครงงานครั้งนี้

นายธานี บัวเทศ
นางสาวกิตติยา ขุมทอง
นายธิติวุฒิ สำเภาเงิน

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์โครงงาน.....	1
1.3 ขอบข่ายการทำงาน.....	2
1.4 ขั้นตอนดำเนินงาน.....	2
1.5 แผนการดำเนินงาน.....	2
1.6 ผลที่คาดว่าจะได้รับ.....	3
1.7 งบประมาณ.....	3
บทที่ 2 หลักการและทฤษฎีเบื้องต้น.....	4
2.1 กติกาการเล่นเกมเอแมท.....	4
2.2 ทฤษฎีทางหลักปัญญาประดิษฐ์.....	6
บทที่ 3 วิธีการดำเนินการ.....	8
3.1 Start.....	9
3.2 Search piece in board.....	9
3.3 Generate Equation.....	11
3.4 Choose Equation.....	12
3.5 Can Play.....	12
3.6 Play chose equation.....	12
3.7 Class Diagram.....	13

สารบัญ (ต่อ)

	หน้า
3.8 Use case Diagram.....	16
บทที่ 4 ผลการทดลอง.....	18
4.1 ผลการทดสอบโปรแกรม.....	18
4.2 ผลการทดลองภาคปฏิบัติ.....	29
บทที่ 5 สรุปผลและข้อเสนอแนะ.....	38
5.1 สรุปผลการทดลอง.....	38
5.2 ปัญหาและอุปสรรค์.....	39
5.3 ข้อเสนอแนะ.....	39
เอกสารอ้างอิง.....	40
ประวัติผู้เขียนโครงงาน.....	41

สารบัญตาราง

ตารางที่	หน้า
1.1 แสดงขั้นตอนการดำเนินงาน.....	3
4.1 แสดงเปอร์เซ็นต์การลงสมการของคอมพิวเตอร์ต่อจำนวนเบี้ยต่อการเล่นเกมในครั้งหนึ่งๆ ซึ่งมีจำนวนเบี้ย คือ 2 ตัว, 3 ตัว, 4 ตัว, 5 ตัว, 6 ตัว, 7 ตัว และ 8 ตัว และจำนวนในการชนะ.....	36

สารบัญรูป

รูปที่	หน้า
2.1 แสดงตัวอย่างกระดานของเกมเอแมท	4
2.2 แสดงตัวอย่างอัลกอลิทึมการย้อนรอยที่ใช้หาคำตอบในเกมปริศนาพระราชินี ขนาด 4 ตัว	7
3.1 แสดงแผนผังแสดงขั้นตอนการทำงานของโปรแกรม	8
3.2 แสดงการค้นหาเพื่อลงสมการโดยการสร้างสมการใหม่	9
3.3 แสดงการต่อสมการใหม่จากการค้นหาเพื่อลงสมการโดยการสร้างสมการใหม่	10
3.4 แสดงการค้นหาเพื่อลงสมการโดยการต่อเติมจากสมการเดิม	10
3.5 แสดงการต่อเติมจากสมการเดิมจากการค้นหาเพื่อลงสมการโดยการต่อเติมจากสมการเดิม	11
3.6 แสดง class diagram ของโปรแกรมทั้งหมด	14
3.7 แสดงรายละเอียดของคลาสต่างๆ	15
3.8 แสดง Use Case Diagram ของระบบ	16
4.1 แสดงหน้าตาของโปรแกรมเมื่อเปิดโปรแกรม	18
4.2 แสดงหน้าตาของโปรแกรมเมื่อต้องการเริ่มเล่นเกมใหม่หรือออกจากเกม	19
4.3 แสดงการเรียกใช้ฟังก์ชั่นของตัวโปรแกรม	19
4.4 แสดงการกดใช้เบี้ยแบล็ง (blank) ในการสร้างสมการ	20
4.5 แสดงวิธีการลงสมการ	21
4.6 แสดงการเตือนเมื่อลงเบี้ยครั้งแรก โดยไม่มีเบี้ยตัวใดเลยทับช่องสัญลักษณ์รูปดาว	21
4.7 แสดงการคืนค่าเบี้ยที่อยู่บนกระดานกลับมาอยู่บนแป้น	22
4.8 แสดงการผ่านรอบของผู้เล่น	23
4.9 แสดงการเปลี่ยนเบี้ยบนแป้นของผู้เล่น	23
4.10 แสดงเบี้ยใหม่ของผู้เล่นเมื่อทำการเปลี่ยนเบี้ย	24
4.11 แสดงการเรียงสมการที่ไม่สามารถคำนวณได้	25
4.12 แสดงการวางเลขศูนย์ไว้ด้านหน้าตัวเลข ซึ่งเป็นสมการที่ไม่ถูกต้อง	25
4.13 แสดงการวางเลขหลักหน่วยต่อเลขหลักสิบ ซึ่งเป็นสมการที่ไม่ถูกต้อง	26
4.14 แสดงการวางเครื่องหมายบวกไว้ด้านหน้าตัวเลขใดๆได้ ซึ่งเป็นสมการที่ไม่ถูกต้อง	26
4.15 แสดงการวางเครื่องหมายลบหน้าเลขศูนย์ ซึ่งเป็นสมการที่ไม่ถูกต้อง	27
4.16 แสดงการวางเบี้ยหลักหน่วยเรียงกัน 4 ตัว ซึ่งเป็นการผิดกฎการเล่นเกมเอแมท	27

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.17	แสดงการเรียงสมการให้ศูนย์หารตัวเลขใดๆ ซึ่งเป็นสมการที่ไม่ถูกต้อง.....	28
4.18	แสดงการเตือนเมื่อผู้เล่นลงสมการที่ไม่ถูกต้อง.....	28
4.19	แสดงหน้าต่างโปรแกรมเมื่อเริ่มเล่นเกมครั้งแรก.....	29
4.20	แสดงการลงสมการของผู้เล่นในการเล่นเกมครั้งแรก.....	29
4.21	แสดงเบี้ยของคอมพิวเตอร์เมื่อจบรอบของผู้เล่น.....	30
4.22	แสดงสมการที่คอมพิวเตอร์คำนวณได้.....	31
4.23	แสดงการลงเบี้ยบนกระดานของคอมพิวเตอร์.....	32
4.24	แสดงการลงสมการผู้เล่นต่อจากคอมพิวเตอร์.....	32
4.25	แสดงเบี้ยของคอมพิวเตอร์หลังจากที่ได้รับเบี้ยมาใหม่.....	33
4.26	แสดงสมการที่คอมพิวเตอร์คำนวณได้ใหม่.....	34
4.27	แสดงการลงสมการของคอมพิวเตอร์.....	34
4.28	แสดงการลงเบี้ยของผู้เล่นจนหมดแป้น (rack) ในรอบ (turn) สุดท้าย.....	35
4.29	แสดงผลแพ้ชนะในการเล่นเกมเอแมท.....	35
4.30	แสดงกราฟของเปอร์เซ็นต์การลงสมการของคอมพิวเตอร์ต่อจำนวนเบี้ย.....	36

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

โลกในปัจจุบันนี้ได้มีวิวัฒนาการในด้านต่างๆอย่างก้าวไกลไม่ว่าจะเป็นในด้านการอุปโภคบริโภค การคมนาคมขนส่งการติดต่อสื่อสาร และการรักษาพยาบาล เป็นต้น โดยเฉพาะอย่างยิ่งในด้านเทคโนโลยี และคอมพิวเตอร์ก็ยิ่งมีวิวัฒนาการก้าวกระโดดกว่าด้านอื่นๆ เนื่องจากว่าในปัจจุบันนี้ มนุษย์ได้มีการนำเอาเทคโนโลยีมาเกี่ยวพันกับการดำเนินชีวิตในปัจจุบันมากขึ้น สิ่งอำนวยความสะดวกต่างๆในชีวิตประจำวันจึงมักจะมีเทคโนโลยีคอมพิวเตอร์เป็นองค์ประกอบหลัก และคาดว่าโลกในอนาคตเทคโนโลยีคอมพิวเตอร์จะมีบทบาทเพิ่มเข้ามาในชีวิตประจำวันของมนุษย์มากยิ่งขึ้น และศาสตร์ที่ช่วยในการพัฒนาเทคโนโลยีคอมพิวเตอร์ให้สามารถคิดประมวลผลได้อย่างชาญฉลาดนั้นก็คือ AI (Artificial Intelligent) หรือปัญญาประดิษฐ์ ซึ่งเป็นศาสตร์แขนงหนึ่งของวิทยาศาสตร์คอมพิวเตอร์ ที่เกี่ยวข้องกับวิธีการที่จะทำให้คอมพิวเตอร์มีความสามารถคล้ายมนุษย์หรือเลียนแบบพฤติกรรมมนุษย์ได้ โดยเฉพาะความสามารถในการคิดเองได้เช่นเดียวกับมนุษย์ ปัญญาประดิษฐ์เป็นศาสตร์ที่มีการศึกษาการแก้ปัญหาที่ซับซ้อนของมนุษย์ โดยอาศัยรากฐานมาจากการคิดและการกระทำของมนุษย์ ซึ่งตัวอย่างที่จะสื่อให้เห็นถึงความสามารถของหลักปัญญาประดิษฐ์ที่เห็นได้อย่างชัดเจนก็คือ เกม โดยเกมส่วนใหญ่มักจะใช้ศาสตร์ของหลักปัญญาประดิษฐ์ในการทำให้ตัวเกมนั้นสามารถทำงานตอบโต้กับผู้เล่นได้ เช่น เกมหมากรุกไทย ที่คอมพิวเตอร์นั้นสามารถเล่นตอบโต้กับผู้เล่นได้อย่างชาญฉลาด เหมือนกับผู้เล่นกำลังเล่นกับผู้เล่นจริงๆ ไม่ว่าจะเป็นการรุกหรือรับ การวางแผนการเดินหมากเพื่อให้ได้รับชัยชนะ ซึ่งในบางครั้งคอมพิวเตอร์ก็สามารถเอาชนะผู้เล่นที่เป็นมนุษย์ได้เช่นกัน

จากที่ได้กล่าวมาข้างต้นจะเห็นได้ว่าศาสตร์ของหลักปัญญาประดิษฐ์นั้น เป็นศาสตร์ที่มีความน่าสนใจเป็นอย่างมาก ซึ่งในอนาคตก็จะเป็นศาสตร์ที่จะมามีบทบาทในชีวิตประจำวันของมนุษย์มากยิ่งขึ้น และเพื่อเป็นการพัฒนาทักษะพื้นฐานในหลักปัญญาประดิษฐ์โครงงานนี้ต้องการพัฒนาเกมเอแมท (A-Math) ซึ่งจะเป็นเกมต่อสมการทางคณิตศาสตร์ โดยลักษณะของเกมจะแบ่งผู้เล่นเป็น 2 ฝ่าย มีกระดานและเบี้ยอีกจำนวน 100 ตัว เกมเอแมท (A-Math) ที่จะพัฒนาขึ้นเป็นพื้นฐานในการสร้างความเข้าใจเกี่ยวกับหลักปัญญาประดิษฐ์และนำไปสู่การพัฒนาที่ดีขึ้น และนอกจากนี้ยังได้รับความสนุก-สนานและเป็นการเสริมสร้างทักษะในด้านคณิตศาสตร์แก่ผู้เล่นอีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อเป็นการศึกษาขั้นตอนและวิธีการทำงานของหลักปัญญาประดิษฐ์

1.2.2 เพื่อให้ได้เกมที่สามารถฝึกฝนทักษะการเล่นเกมเอแมทของผู้เล่นได้

1.3 ขอบข่ายของโครงงาน

1.3.1 ใช้กฎกติกาการเล่นเกมเอแมทตามมาตรฐานสากล

1.3.2 สามารถเล่นได้เพียงคนเดียว โดยผู้เล่นอีกฝ่ายจะเป็นคอมพิวเตอร์

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาและรวบรวมข้อมูลที่เกี่ยวข้องกับหลักการปัญญาประดิษฐ์และเกมเอแมท

1.4.2 ออกแบบขั้นตอนการทำงานของตัวเกม

1.4.3 ทำการสร้างตัวเกมที่สามารถเล่นตอบโต้กับผู้เล่นที่เป็นมนุษย์ได้

1.4.4 ตรวจสอบตัวเกมว่าสามารถเล่นได้ สามารถเล่นได้ถูกต้องตามกฎของการเล่น และสมการถูกต้อง

1.4.5 จัดทำคู่มือโครงงาน

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 แสดงขั้นตอนการดำเนินงาน

กิจกรรม	ปี 2551						ปี 2552	
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1. ศึกษาและรวบรวมข้อมูลที่เกี่ยวข้องกับหัวข้อโครงงาน								
2. ออกแบบขั้นตอนการทำงานของตัวเกม								
3. ทำการสร้างตัวเกมที่สามารถเล่นตอบโต้กับผู้เล่นที่เป็นมนุษย์ได้								
4. ตรวจสอบตัวเกมว่าสามารถเล่นได้ สามารถเล่นได้ถูกต้องตามกฎของการเล่น และสมการถูกต้อง								
5. จัดทำคู่มือโครงงาน								

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ความรู้ความเข้าใจในเขียนโปรแกรมโดยใช้ภาษา C#

1.6.2 เกมเอแมทที่ถูกพัฒนาขึ้นสามารถเล่นตอบโต้กับผู้เล่นได้จริง

1.7 งบประมาณของโครงงาน

1.7.1 ค่าเอกสารและคู่มือ	2,000 บาท
1.7.2 ค่าถ่ายเอกสาร	500 บาท
1.7.3 ค่าวัสดุ	500 บาท
รวมทั้งสิ้น (สามพันบาทถ้วน)	3,000 บาท

หมายเหตุ ขออนุมัติโดยถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีเบื้องต้น

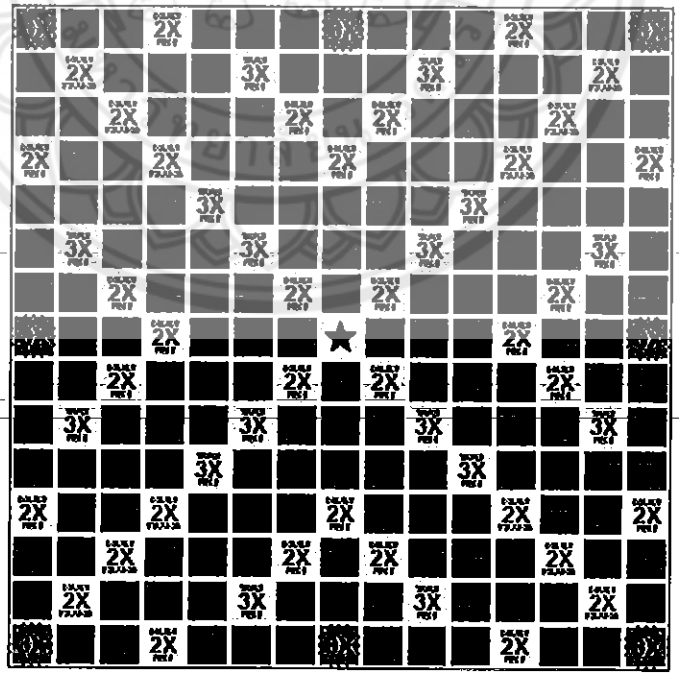
ในบทนี้จะกล่าวถึงกฎกติกาการเล่นของเกมเอแมท (A-Math) เบื้องต้น รวมไปถึงทฤษฎีที่นำมาใช้ในการหาคำตอบ ซึ่งประกอบไปด้วยอัลกอริทึมการค้นหาแบบลำดับ (Linear Search Algorithm) ซึ่งเป็นอัลกอริทึมที่ใช้ในการหาตำแหน่งของเบี้ยที่อยู่บนกระดานที่สามารถลงได้ และอัลกอริทึมแบบย้อนรอย (Backtracking Algorithm) ที่ใช้ในการสลับเบี้ยเพื่อสร้างสมการเพื่อนำมาลงต่อกับเบี้ยที่อยู่บนกระดานในตำแหน่งที่สามารถลงสมการได้

2.1 กติกาการเล่นเกมเอแมท

เอแมท (A-Math) [3] เป็นเกมต่อสมการคณิตศาสตร์ หลักในการเล่นคือต่อตัวเลขตามหลักการทางคณิตศาสตร์ลงบนช่องตารางให้ได้ผลดีมากที่สุด เมื่อจบเกมผู้ที่ได้คะแนนมากกว่าจะเป็นผู้ชนะ

2.1.1 อุปกรณ์การเล่น

- กระดาน (Board) มีขนาดกว้าง 15 ช่อง และสูง 15 ช่อง รวมทั้งสิ้น 225 ช่อง



รูปที่ 2.1 แสดงตัวอย่างกระดานของเกมเอแมท

รูปภาพจาก http://fwt.txdnl.com/6-30/n/o/nooyok/amathboard.JPG

- เบี้ย (Piece) มีทั้งสิ้น 100 ตัวในถุง
- แป้น (Rack) ใช้ในการวางเบี้ย โดยแต่ละฝ่ายจะมีเบี้ยในแป้นฝ่ายละ 8 ตัว

2.1.2 กติกาการเล่นทั่วไป

- ในการลงแต่ละครั้ง (Turn) ให้ผู้เล่นลงเบี้ยกี่ตัวก็ได้ในแนวเดียวกันและต่อกับเบี้ยเดิมในกระดาน เพื่อให้เกิดเป็นสมการคณิตศาสตร์ใหม่ที่ถูกต้อง
- การลงครั้งแรก ต้องมีเบี้ยหนึ่งตัวลงทับกับช่องดาวบนกระดาน
- การคิดคะแนน นำคะแนนของเบี้ยแต่ละตัวในสมการมาบวกกัน หากลงเบี้ยใหม่ในช่องคะแนนพิเศษจะได้คะแนนเพิ่ม ดังนี้
  - ช่องสีแดง นำคะแนนของทั้งสมการคูณสาม
  - ช่องสีเหลือง นำคะแนนของทั้งสมการคูณสอง
  - ช่องสีฟ้า นำคะแนนของตัวเบี้ยที่ทับช่องคูณสาม
  - ช่องสีส้ม นำคะแนนของตัวเบี้ยที่ทับช่องคูณสอง

2.1.3 การสิ้นสุดเกม มีสองกรณี คือ

- เมื่อเบี้ยในถุงหมด และผู้เล่นฝ่ายใดฝ่ายหนึ่งใช้เบี้ยที่มีอยู่จนหมด หรือ
- เมื่อผู้เล่นทั้ง 2 ฝ่ายมีการเล่นไปแล้ว และหลังจากนั้นไม่มีการลงเบี้ยในกระดานติดต่อกันรวม 6 ครั้ง

2.1.4 กติกาการเล่นพิเศษ

- การขอเปลี่ยนตัว ผู้เล่นสามารถเปลี่ยนตัวได้ตั้งแต่ 1-8 ตัว และจะเสียตาในการวางเบี้ย 1 ครั้ง โดยให้นำเบี้ยที่ต้องการเปลี่ยนมาวางบนกระดานและขอสิทธิ์การเปลี่ยนตัว แต่ถ้าเบี้ยในถุงเหลือน้อยกว่า 5 ตัวจะไม่สามารถเปลี่ยนได้
- การทำบิงโก (bingo) หากผู้เล่นสามารถลงเบี้ยทั้ง 8 ตัวได้ในตาเล่นครั้งเดียว จะได้คะแนนพิเศษเพิ่มอีก 40 คะแนน

2.1.5 สมการที่ถูกต้อง

- ค่าของสมการฝั่งซ้ายและขวาต้องเท่ากัน โดยคำนวณเครื่องหมายคูณหรือหาร ก่อนเครื่องหมายบวกหรือลบ และทำตามลำดับจากซ้ายไปขวา หรือบนลงล่าง
- เครื่องหมายบวก ลบ คูณ หาร ห้ามวางติดกัน

- เครื่องหมายลบสามารถนำมาต่อหน้าตัวเลขเพื่อทำเป็นเลขติดลบได้ แต่ห้ามวางเป็นเลขติดลบศูนย์

- เลขศูนย์ห้ามเป็นตัวหาร

- เลขหลักเดียวสามารถนำมาต่อเป็นเลข 2 และ 3 หลักได้ แต่ห้ามนำเลขศูนย์มาต่อหน้าเลขตัวอื่น

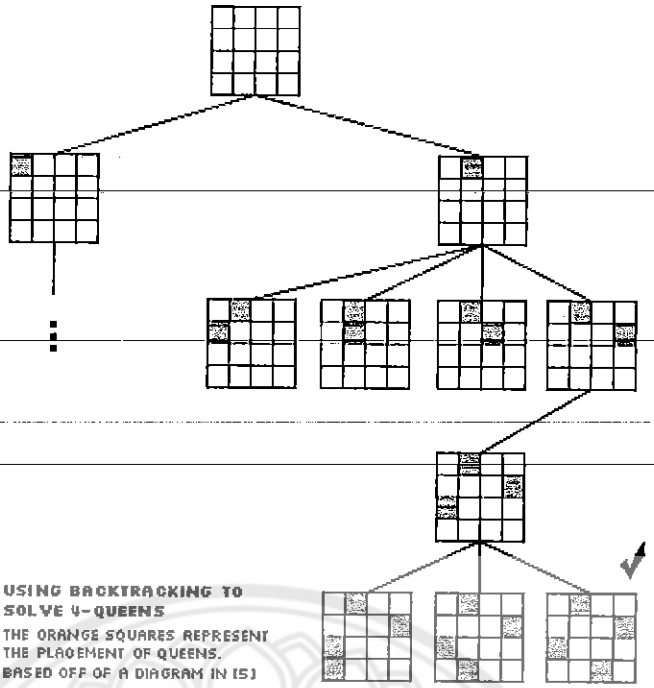
- เลขหลักเดียวไม่สามารถนำมาต่อกับเลขสองหลักเพื่อทำเป็นเลข3หลักได้

2.2 ทฤษฎีทางหลักปัญญาประดิษฐ์

2.2.1 Backtracking Algorithm

อัลกอริทึมการย้อนรอย (Backtracking Algorithm) [2] เป็นอีกอัลกอริทึมหนึ่งที่เป็นที่นิยมในการใช้หาคำตอบต่างๆเนื่องจากเป็นอัลกอริทึมที่เข้าใจง่ายและมีประสิทธิภาพ โดยหลักการทำงานของอัลกอริทึมการย้อนรอย คือจะสุ่มเลือกคำตอบจากเซตของคำตอบทั้งหมดโดยจะสุ่มไปเรื่อยๆจนกว่าจะพบคำตอบที่ต้องการ แต่ถ้าในขณะที่กำลังสุ่มอยู่นั้นเกิดสุ่มเจอคำตอบที่ผิดก็จะกลับมาทำในระดับขั้นก่อนหน้านั้นและถ้ายังหาคำตอบไม่ได้ก็จะถอยกลับมาทำยังระดับขั้นก่อนหน้านั้นถัดไปอีก ซึ่งวิธีการของอัลกอริทึมการค้นหาแบบย้อนรอย (Backtracking Algorithm) นั้นคล้ายกับการลองผิดลองถูก หรืออาจกล่าวคือ ถ้าคำตอบที่เลือกมานั้นถูกต้องก็จะสุ่มหาคำตอบตัวถัดไปต่อไป แต่ถ้าคำตอบที่สุ่มมานั้นผิดก็จะกลับไปเลือกคำตอบก่อนหน้านี้ใหม่

อัลกอริทึมการย้อนรอย (Backtracking Algorithm) ที่โครงงานนี้นำมาใช้นั้นจะเป็นแบบการค้นหาในแนวลึก (Depth – first search) ซึ่งหลักการในการหาคำตอบของวิธีนี้คือการสุ่มเลือกคำตอบที่สามารถเป็นไปได้ทั้งหมดไปเรื่อยๆ และเมื่อใดที่ไม่สามารถหาคำตอบต่อไปได้ก็จะย้อนกลับมายังในระดับขั้นก่อนหน้าและหาคำตอบต่อในตัวเลือกอื่นๆ ซึ่งจะไม่เลือกคำตอบที่ผิดตัวนั้นอีก ท้ายที่สุดแล้วถ้าไม่มีคำตอบใดที่สามารถเลือกได้แล้วก็จะถือว่าหาคำตอบไม่ได้



รูปที่ 2.2 แสดงตัวอย่างอัลกอริทึมการย้อนรอยที่ใช้หาคำตอบในเกมปริศนาพระราชินีขนาด 4 ตัว

รูปภาพจาก http://4c.ucc.ie/web/outreach/backtracking.jpg

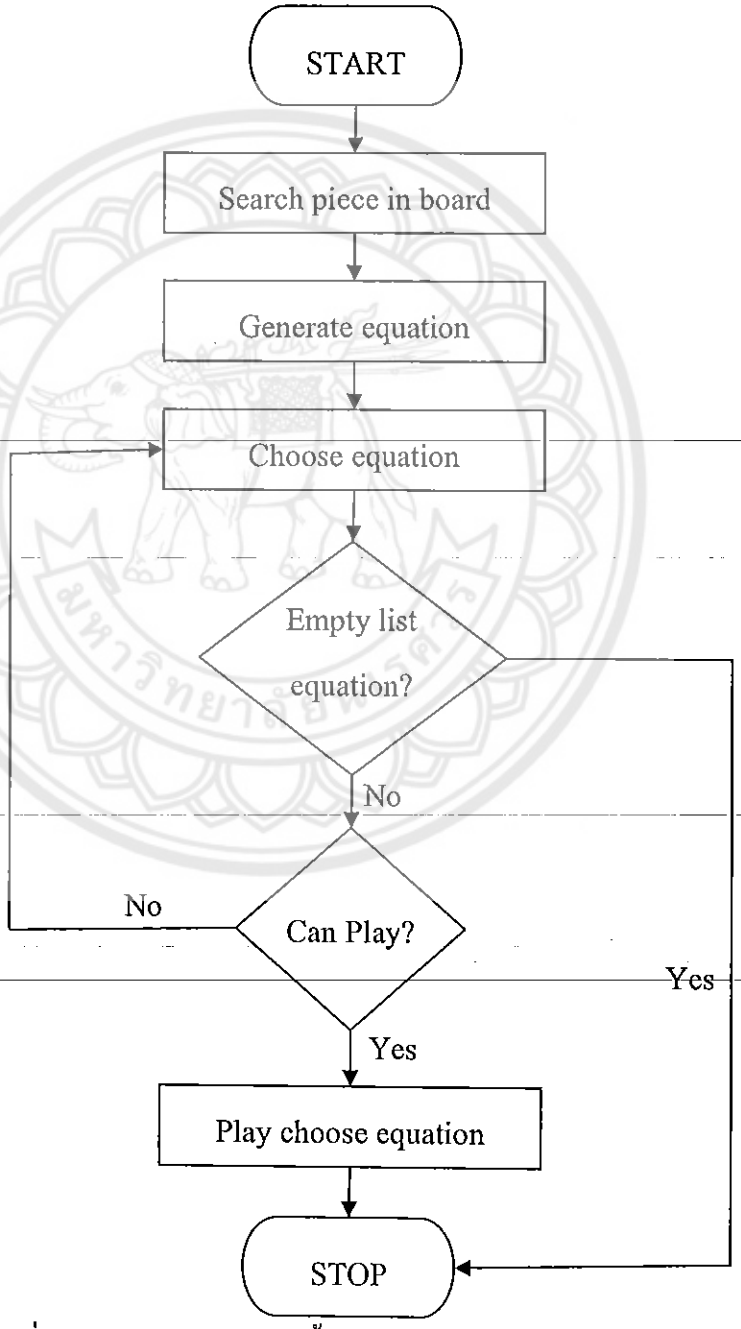
2.2.2 Linear Search

อัลกอริทึมการค้นหาแบบลำดับ (Linear Search Algorithm) [1] เป็นอัลกอริทึมการค้นหาซึ่งเหมาะสำหรับการค้นหากลุ่มของข้อมูล (list) ที่มีค่าคงที่ โดยหลักการทำงานของวิธีการค้นหาแบบลำดับ (Linear Search) ก็คือจะทำการตรวจสอบทุกๆสมาชิกที่อยู่ในกลุ่มของข้อมูล (list) จนกว่าจะเจอตัวที่ตรงกับความต้องการ ซึ่งในโครงงานนี้จะนำวิธีการค้นหาแบบลำดับ (Linear Search) ไปใช้ในการค้นหาเบี้ยที่อยู่บนกระดาน โดยที่จะการค้นหาไปทุกๆช่องในกระดานจนกว่าจะเจอเบี้ยที่ต้องการ

บทที่ 3

วิธีการดำเนินการ

หลังจากที่ได้ศึกษากติกาการเล่นเกมเอแมทและอัลกอริทึมที่สามารถนำมาใช้ในการพัฒนาทักษะของหลักปัญญาประดิษฐ์ ในบทนี้จะกล่าวถึงขั้นตอนการทำงานของหลักปัญญาประดิษฐ์ในการเล่นเกมเอแมทโต้ตอบกับผู้เล่น ซึ่งโปรแกรมที่พัฒนานั้นมีแผนผังการทำงานดังนี้



รูปที่ 3.1 แสดงแผนผังแสดงขั้นตอนการทำงานของโปรแกรม

3.1 Start

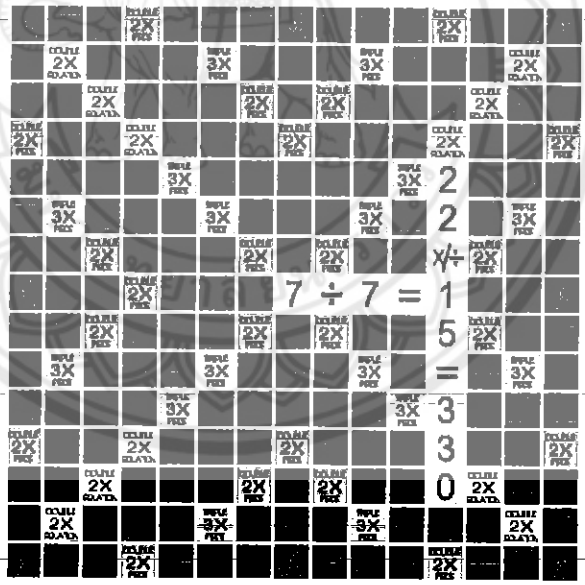
คอมพิวเตอร์จะเริ่มทำการคำนวณหลังจากที่ฝ่ายผู้เล่นทำการคำนวณ และลงสมการโดยการกดปุ่มตกลง (submit) เพื่อเป็นการยืนยันการลงสมการหรือกดปุ่มผ่าน (pass) หรือกดปุ่มเปลี่ยนตัวเบี้ย (fishing)

3.2 Search piece in board

ในการลงสมการของคอมพิวเตอร์ สามารถแบ่งออกได้เป็น 2 ลักษณะ ซึ่งการค้นหานี้จะทำได้ด้วยวิธีการที่แตกต่างกัน คือ

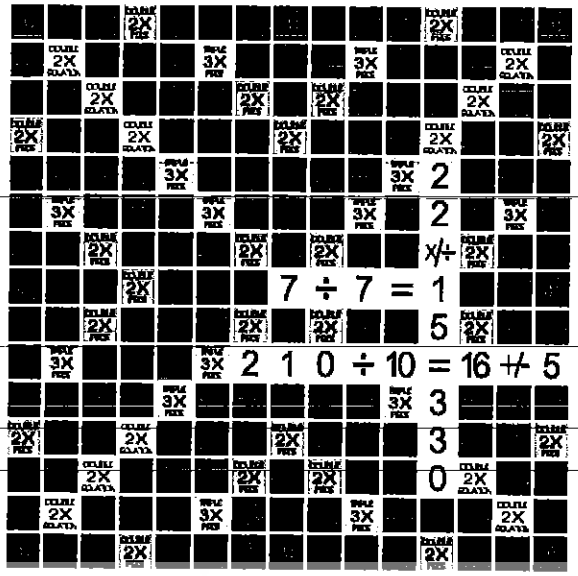
3.2.1 การค้นหาเพื่อลงสมการโดยการสร้างสมการใหม่

คอมพิวเตอร์จะทำการค้นหาเบี้ยทุกๆตัวที่อยู่บนกระดานและเก็บเฉพาะเบี้ยที่มีช่องว่างอยู่รอบๆตัว โดยการเลือกเก็บเบี้ยที่มีช่องอยู่นั้น คอมพิวเตอร์จะเลือกจากเบี้ยที่มีช่องว่างจากตำแหน่งของเบี้ยที่จะเก็บมากกว่า 1 ช่องในลักษณะด้านบนหรือด้านซ้าย และ ด้านล่างหรือด้านขวาสำหรับลงสมการโดยการสร้างสมการใหม่



รูปที่ 3.2 แสดงการค้นหาเพื่อลงสมการโดยการสร้างสมการใหม่

จากรูปที่ 3.2 คอมพิวเตอร์จะทำการเก็บเบี้ย {7, ÷, 7} ในแนวนอนและ {2, 2, =, 3, 3, 0} ในแนวตั้งเพื่อนำมาคำนวณในการสร้างสมการใหม่

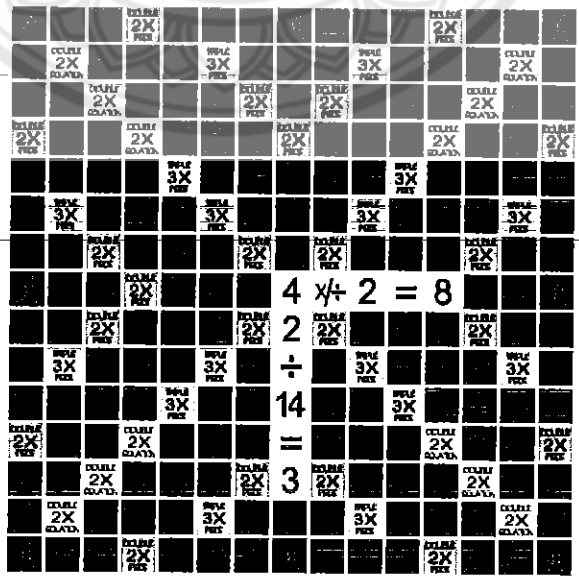


รูปที่ 3.3 แสดงการต่อสมการใหม่จากการค้นหาเพื่อลงสมการโดยการสร้างสมการใหม่

จากรูปที่ 3.3 คอมพิวเตอร์จะทำการลงสมการใหม่ซึ่งก็คือ {210÷10 = 16+5} ซึ่งเป็นผลมาจากการค้นหาเบี้ยที่อยู่บนกระดานแล้วนำมาเก็บเพื่อนำไปคำนวณกับเบี้ยที่อยู่ในแป้น (rack) ของฝ่ายคอมพิวเตอร์ทำให้ได้สมการใหม่ โดยคอมพิวเตอร์นำเครื่องหมาย {=} บนกระดานมาคำนวณกับเบี้ยในแป้น (rack) ทั้ง 8 ตัวได้แก่ {2, 1, 0, ÷, 10, 16, +/-, 5}

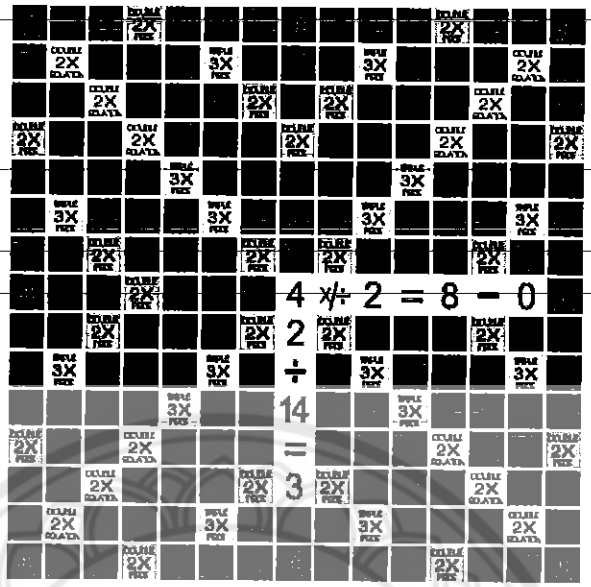
3.2.2 การค้นหาเพื่อลงสมการโดยการต่อเติมจากสมการเดิม

คอมพิวเตอร์จะทำการค้นหาเบี้ยทุกๆตัวที่อยู่บนกระดานและเก็บเบี้ยไว้ในลักษณะของสมการ เพื่อนำไปต่อเติม และทำคะแนนต่อไป



รูปที่ 3.4 แสดงการค้นหาเพื่อลงสมการโดยการต่อเติมจากสมการเดิม

จากรูปที่ 3.4 คอมพิวเตอร์จะทำการเก็บสมการ {4×2 = 8} ในแนวนอนและสมการ {42÷14 = 3} ในแนวตั้งเพื่อนำไปต่อเติมสมการ



รูปที่ 3.5 แสดงการต่อเติมจากสมการเดิมจากการค้นหาเพื่อลงสมการโดยการต่อ**เติมจากสมการเดิม**

จากรูปที่ 3.5 คอมพิวเตอร์จะทำการต่อเติมสมการเดิมที่ทำการค้นหาแล้ว**นำมาเก็บไว้ก็คือ** สมการ {4×2 = 8} โดยทำการต่อเติมสมการเป็น {4×2 = 8-0} โดยคอมพิวเตอร์**จะนำเบี้ยที่มีอยู่ใน** แป้น (rack) มาคำนวณแต่เนื่องจากว่าไม่สามารถลงเป็นสมการได้ คอมพิวเตอร์**จึงนำ** {-, 0} จากใน แป้น (rack) ซึ่งมีเบี้ยได้แก่ {10, 18, 9, -, 1, 16, 7, 0} มาต่อเติมสมการเดิมบนกระดาน

3.3 Generate Equation

หลังจากทำการค้นหาเบี้ยบนกระดานแล้วนำมาเก็บไว้ในขั้นตอนที่แล้ว **ต่อมาคอมพิวเตอร์** จะทำการประมวลผลโดยแบ่งออกเป็น 2 ลักษณะเช่นเดียวกับขั้นตอนการ search piece in board คือ

3.3.1 Generate Equation สำหรับลงเบี้ยเพื่อสร้างสมการใหม่

คอมพิวเตอร์จะนำเบี้ยที่ได้มาทำการเก็บไว้เพื่อนำมาสร้างสมการใหม่มาคิดร่วมกับเบี้ยที่ อยู่บนแป้นของคอมพิวเตอร์แล้วใช้วิธีการย้อนรอย (backtracking) เพื่อสร้างสมการใหม่ โดยการ ย้อนรอย (backtracking) จะทำการสร้างสมการขนาด 3 ตัว, 4 ตัว, 5 ตัว, 6 ตัว, 7 ตัว, 8 ตัวและ 9 ตัว สำหรับลงในสถานการณ์ที่แตกต่างออกไปมาเก็บไว้ในชุดของสมการทั้งหมด (list)

3.3.2 Generate Equation สำหรับต่อเติมสมการที่มีอยู่แล้ว

คอมพิวเตอร์จะทำการ Brute Force สลับเบี้ยบนแป้นที่มีอยู่แล้วเพื่อให้ได้การเรียงเบี้ยทุกกรณีที่สามารถเป็นไปได้ จากนั้นคอมพิวเตอร์จะทำการดึงชุดของเบี้ยที่ต่อกันแล้วสามารถนำไปต่อสมการเดิมแล้วได้ค่าที่ไม่เปลี่ยนแปลง เช่น +0 (บวกด้วยศูนย์), x1 (คูณด้วยหนึ่ง) เก็บไว้ในชุดของสมการทั้งหมด (list)

3.4 Choose Equation

จากผลการ Generate Equation ในขั้นตอนที่ผ่านมาคอมพิวเตอร์จะทำการเลือกสมการที่มีความยาวมากที่สุดในชุดของสมการทั้งหมดมาลง แต่ถ้าไม่สามารถลงสมการได้คอมพิวเตอร์จะทำการเลือกสมการที่มีความยาวน้อยกว่าลงมาจากสมการก่อนหน้าแทน ถ้าหากเลือกจนไม่มีสมการใดให้เลือกได้แล้วคอมพิวเตอร์ก็จะสิ้นสุดการเล่นในรอบของตนเองโดยการผ่าน (pass) หรือเปลี่ยนเบี้ย (fishing)

3.5 Can Play?

ในขั้นตอนนี้คอมพิวเตอร์จะทำการตรวจสอบว่าสมการที่เลือกมานั้นสามารถลงบนกระดานได้หรือไม่ โดยดูจากช่องว่างบนกระดานว่ามีเพียงพอต่อการลงเบี้ย เพื่อต่อสมการได้หรือไม่ ถ้าสามารถลงได้ก็จะทำการลงเบี้ยในกระดาน แต่ถ้าไม่สามารถลงได้ก็จะทำการเลือกสมการเพื่อที่จะลงใหม่อีกครั้ง

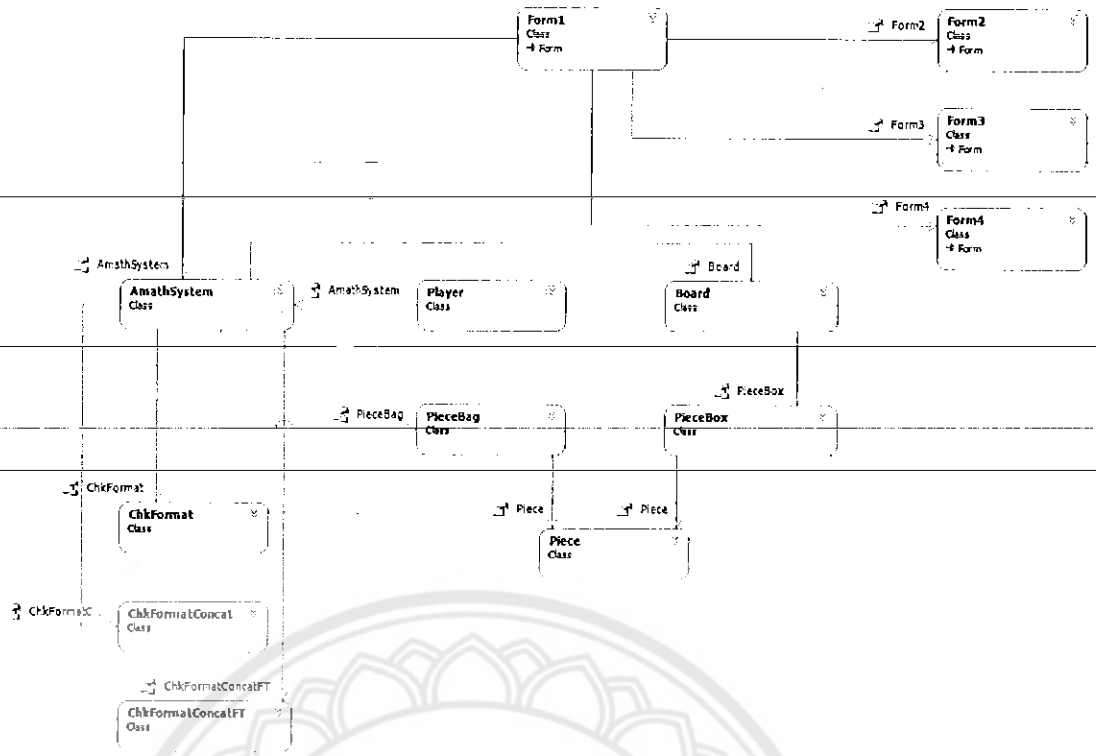
3.6 Play choose equation

ถ้าบนกระดานมีพื้นที่ว่างเพียงพอสำหรับลงสมการคอมพิวเตอร์จะทำการลงสมการ โดยเลือกเบี้ยบนแป้น (rack) มาลงตามสมการที่ได้เลือกไว้ หลังจากนั้นระบบจะทำการคำนวณคะแนนและสิ้นสุดการเล่นในเทอมของคอมพิวเตอร์

3.7 Class Diagram

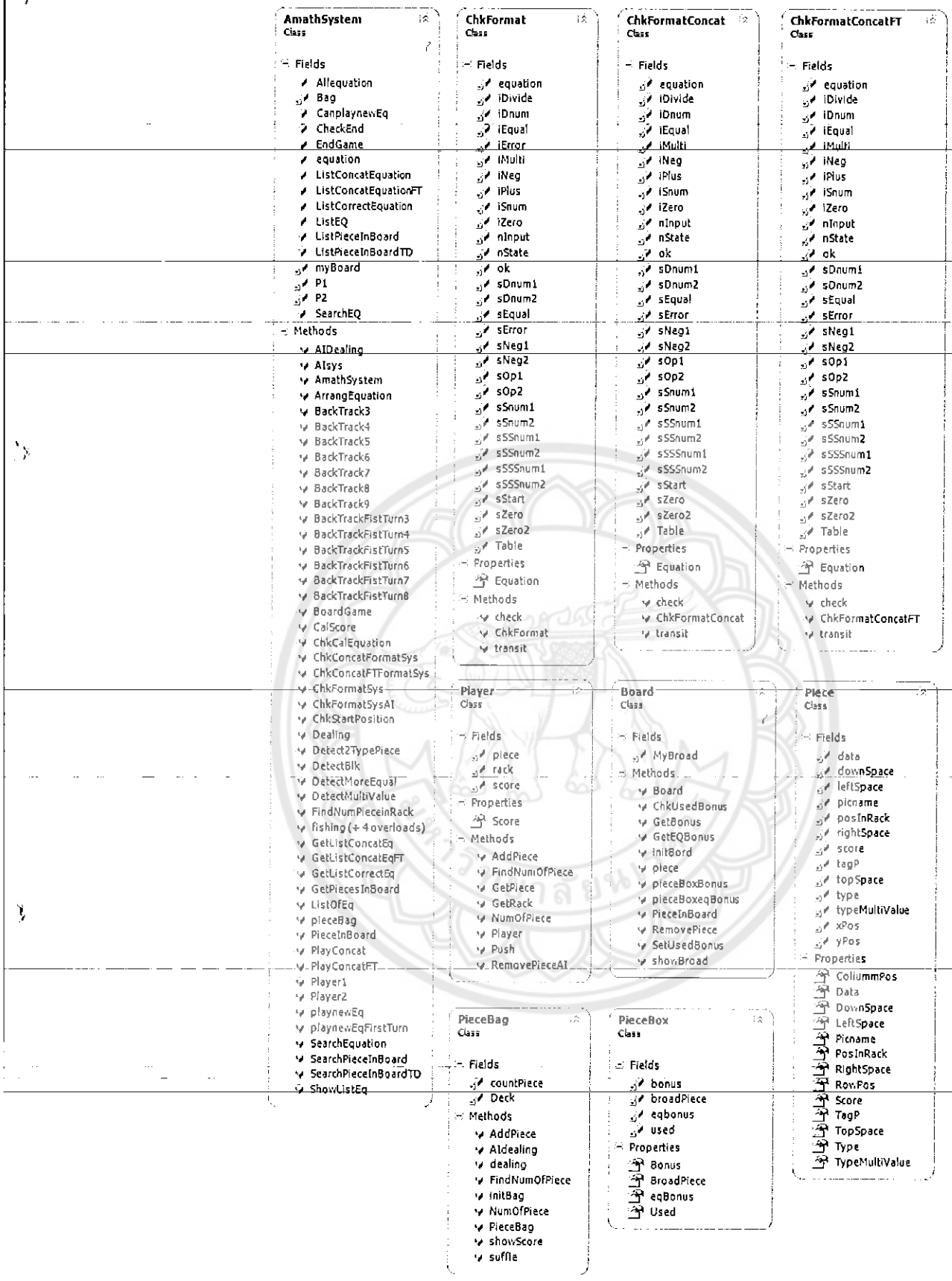
โครงสร้างหลักของโปรแกรมประกอบด้วยการทำงานระหว่างส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface, GUI) หลักทำการเชื่อมต่อกับตัวระบบ โดย ในที่นี้คลาสที่ทำหน้าที่เป็น กราฟฟิกยูสเซอร์อินเตอร์เฟสหลักคือ Class Form1 ซึ่งจะมีหน้าที่ในการแสดงผลเป็นหน้าต่างหลักให้กับโปรแกรมและมี Form2, Form3 และ Form4 เป็น Form สนับสนุนในระหว่างการเลือกค่าข้อมูลให้กับตัวเบี้ยพิเศษต่างๆ และคลาสที่ทำหน้าที่เป็นตัวระบบหลักคือ AmathSystem ซึ่งเป็นคลาสหลักที่ใช้ในการดำเนินเกม โดยคลาส AmathSystem จะทำการสร้างกระดาน โดยเรียกใช้คลาส Board ซึ่งคลาส Board ใช้ ArrayList สร้างเป็นกระดานของเกม โดยที่แต่ละช่องของกระดานประกอบด้วยอ็อบเจกต์ของคลาส PieceBox ซึ่งจะมีหน้าที่ในการเชื่อมโยงตำแหน่งต่างๆของเบี้ยแต่ละตัวที่อยู่บนกระดานแล้วคลาส PieceBox จะเก็บอ็อบเจกต์ของคลาส Piece ไว้ภายใน โดยที่คลาส Piece คือคลาสที่ทำหน้าที่เป็นเบี้ยของเกม และหลังจากการสร้างกระดานแล้วคลาส AmathSystem จะทำการสร้างถุงเบี้ยของระบบขึ้นมา โดยการเรียกใช้คลาส PieceBag ซึ่งเป็นคลาสที่ทำหน้าที่ในการสร้างเบี้ยสำหรับเกมขึ้นมา คลาส Piecebag จึงประกอบด้วยอ็อบเจกต์ของคลาส Piece อยู่ภายในตามจำนวนเบี้ยที่มีอยู่ในเกม

เมื่อดำเนินเกมคลาส AmathSystem จะทำการเรียกใช้คลาส ChkFormat สำหรับตรวจสอบความถูกต้องของสมการปรกติ ซึ่งหากผลการตรวจสอบจากคลาส ChkFormat ออกมาเป็นสมการที่ถูกต้อง คลาส AmathSystem จะทำการคำควณความถูกต้องของสมการ โดยใช้ method ChkCalEquation ที่อยู่ภายในคลาส AmathSystem ซึ่งหากได้ผลออกมาถูกต้องก็จะนำไปคำนวณเป็นคะแนน โดยใช้ method CalScore และทำการเปลี่ยนค่าคะแนนให้กับ อ็อบเจกต์ของคลาส Player โดยที่คลาส Player จะถูกสร้างเป็นอ็อบเจกต์ภายในคลาส AmathSystem เพื่อเรียกใช้ methods ต่างๆในคลาส AmathSystem โดยที่ AmathSystem จะจัดการเรื่องคะแนน, การดึงเบี้ยใหม่ขึ้นมาจากถุงในกรณีที่เบี้ยไม่ครบ ,การเปลี่ยนรอบระหว่างผู้เล่น และทำหน้าที่ควบคุมระบบของผู้เล่นคอมพิวเตอร์ด้วย โดยการทำงานหลักของผู้เล่นฝ่ายคอมพิวเตอร์จะถูกเก็บไว้ภายใน method AIsys ซึ่งจะทำหน้าที่คำนวณเบี้ยเพื่อสร้างสมการ โดยใช้ method BatckTrack9 และอื่นๆ แล้วจะนำไปตรวจสอบโดยคลาส ChkFormat,ChkFormatConcat หรือ ChkFormatConcatFT และ ChkCalEquation เพื่อให้ได้สมการที่ถูกต้องนำไปใช้ในการลงบนกระดานต่อไป ซึ่งจะแสดงใน Class diagram ดังรูปที่ 3.6



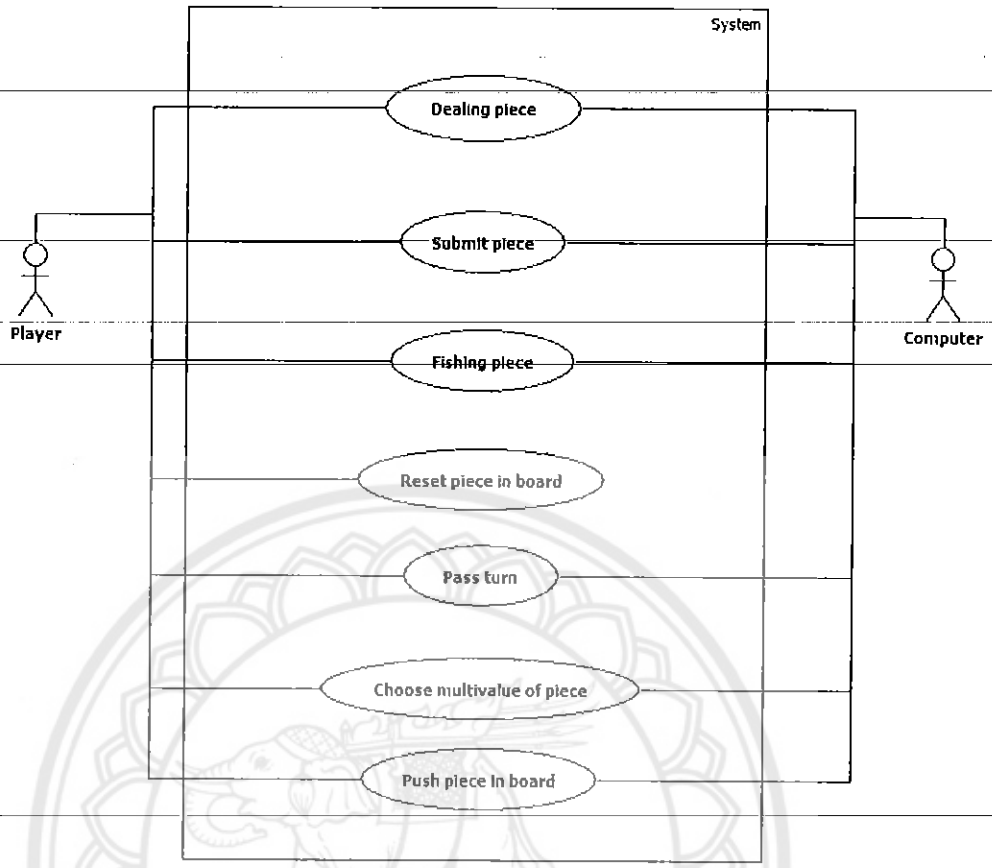
รูปที่ 3.6 แสดง class diagram ของโปรแกรมทั้งหมด

จากรูปที่ 3.6 ซึ่งแสดง class diagram ของโปรแกรมทั้งหมดจะสามารถแยกรายละเอียดภายในแต่ละคราสได้ดังรูปที่ 3.7



รูปที่ 3.7 แสดงรายละเอียดของคลาสต่างๆ

3.8 Use Case Diagram



รูปที่ 3.8 แสดง Use Case Diagram ของระบบ

จากรูปที่ 3.8 ซึ่งแสดง Use case diagram ของระบบ โดยแบ่งออกเป็น 2 ฝ่ายคือ Player และ Computer ดังนี้

1. Actor player สามารถเรียกใช้งานฟังก์ชันได้ดังนี้

- Dealing Piece จะถูกเรียกใช้อัตโนมัติ เมื่อผู้เล่นมีเบี้ยไม่ครบตามจำนวนที่กำหนดไว้

- Submit Piece ผู้เล่นสามารถเรียกใช้ได้เมื่อต้องการทำการลงเบี้ยเพื่อทำคะแนน

- Fishing Piece ผู้เล่นสามารถเรียกใช้ได้เมื่อต้องการเปลี่ยนเบี้ยในถุง โดยมีข้อกำหนดคือต้องมีเบี้ยไม่น้อยกว่า 5 ตัวในถุง

- Reset piece in board ผู้เล่นสามารถเรียกใช้ได้เมื่อต้องการยกเลิกเบี้ยที่ผู้เล่นวางไว้บนกระดานก่อนที่เบี้ยนั้นจะถูกตรวจสอบว่าถูกต้องโดยปุ่มตกลง (Submit)

- Pass turn ผู้เล่นสามารถเรียกใช้ได้เมื่อต้องการคงสภาพเบี้ยที่มีอยู่บนแป้น (rack) ของตนเอง แล้วเปลี่ยนรอบให้ผู้เล่นอีกฝ่าย

- Choose multivalue of piece ผู้เล่นสามารถเรียกใช้ได้เมื่อต้องการเลือกค่าเบี้ยที่สมารถเปลี่ยนแปลงค่าได้

- Push piece in board ผู้เล่นสามารถเรียกใช้ได้เมื่อต้องการวางเบี้ยลงบนกระดาน

2. Actor Computer สามารถเรียกใช้งานฟังก์ชันได้ดังนี้

- Dealing Piece จะถูกเรียกใช้อัตโนมัติ เมื่อผู้เล่นมีเบี้ยไม่ครบตามจำนวนที่กำหนดไว้

- Submit Piece คอมพิวเตอร์สามารถเรียกใช้ได้เมื่อต้องการทำการลงเบี้ยเพื่อทำคะแนน

- Fishing Piece คอมพิวเตอร์สามารถเรียกใช้ได้เมื่อต้องการเปลี่ยนเบี้ยในถุง โดยมีข้อกำหนดคือต้องมีเบี้ยไม่น้อยกว่า 5 ตัวในถุง

- Reset piece in board คอมพิวเตอร์ไม่สามารถเรียกใช้ได้เนื่องจากก่อนทำการลงสมการคอมพิวเตอร์จะตรวจสอบความถูกต้องของเบี้ยที่จะทำการลงก่อนแล้ว

- Pass turn คอมพิวเตอร์สามารถเรียกใช้ได้เมื่อต้องการคงสภาพเบี้ยที่มีอยู่บนแป้น (rack) ของตนเอง แล้วเปลี่ยนรอบให้ผู้เล่นอีกฝ่าย

- Choose multivalue of piece คอมพิวเตอร์สามารถเรียกใช้ได้เมื่อต้องการเลือกค่าเบี้ยที่สามารถเปลี่ยนแปลงค่าได้

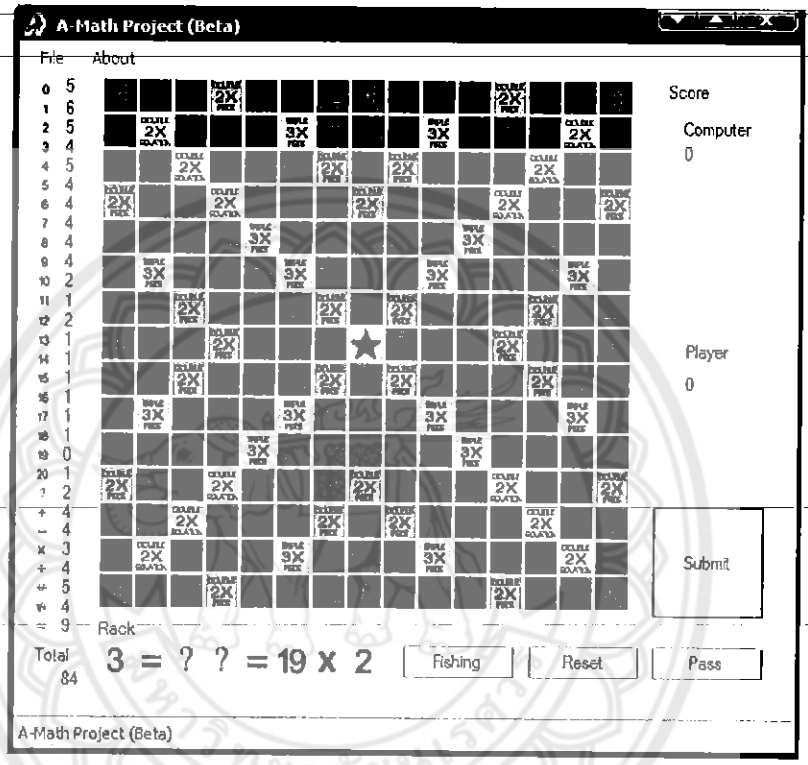
- Push piece in board คอมพิวเตอร์สามารถเรียกใช้ได้เมื่อต้องการวางเบี้ยลงบนกระดาน

บทที่ 4

ผลการทดลอง

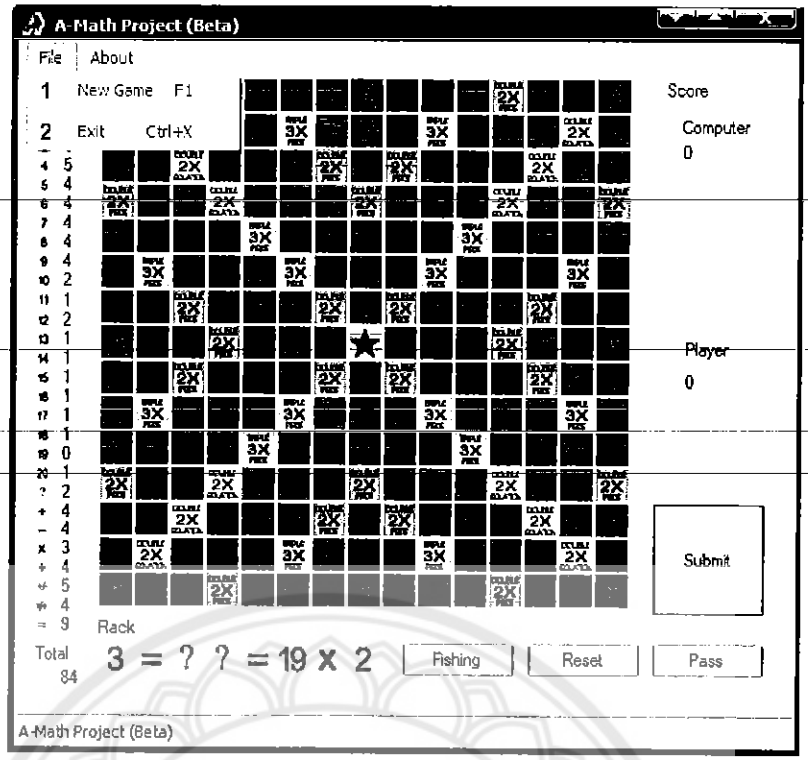
4.1 ผลการทดสอบโปรแกรม

เมื่อเปิดโปรแกรมขึ้นมาจะปรากฏหน้าหน้าของโปรแกรมดังรูปที่ 4.1



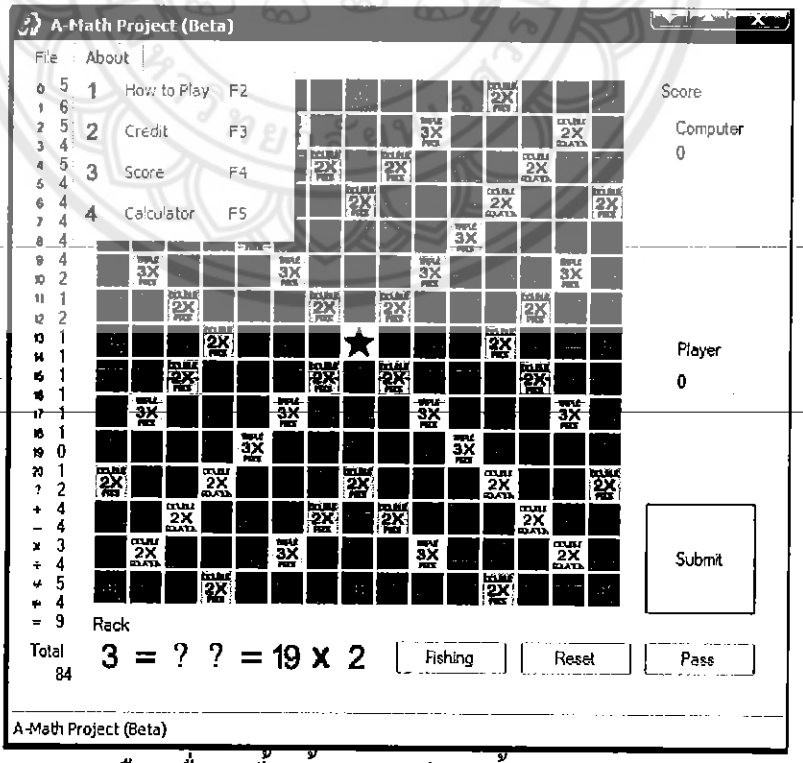
รูปที่ 4.1 แสดงหน้าตาของโปรแกรมเมื่อเปิดโปรแกรม

โปรแกรมสามารถเลือกเริ่มเล่นเกมใหม่ หรือ ออกจากโปรแกรมได้ ดังรูปที่ 4.2



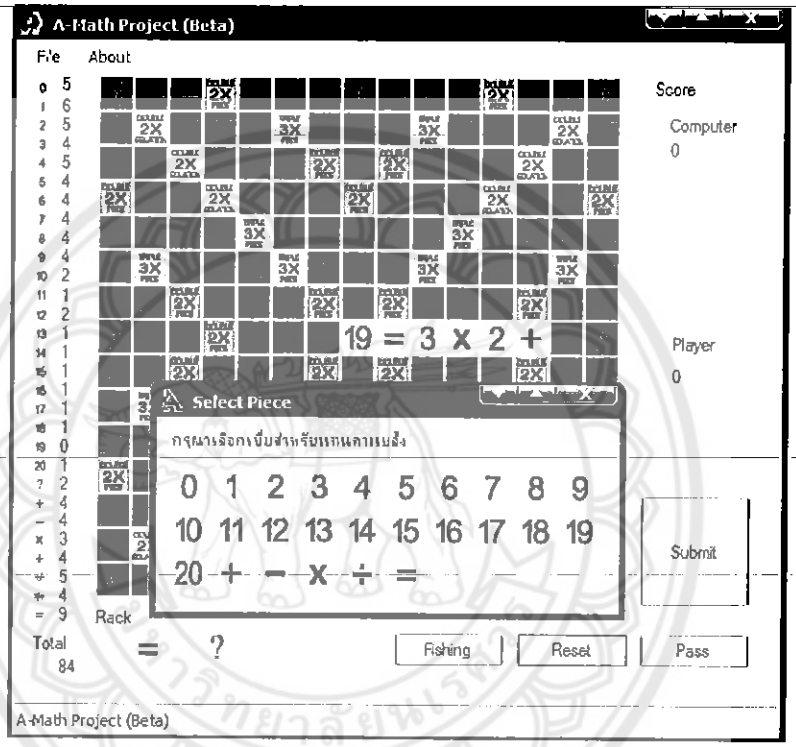
รูปที่ 4.2 แสดงหน้าตาของโปรแกรมเมื่อต้องการเริ่มเล่นเกมใหม่หรือออกจากเกม

หากผู้ใช้โปรแกรมไม่เคยเล่นเกมเอแมทมาก่อนก็สามารถเรียกดูวิธีการเล่นเกม, คะแนนของตัวเบี้ยในเกม และสามารถเรียกใช้เครื่องคิดเลขได้ ดังรูปที่ 4.3



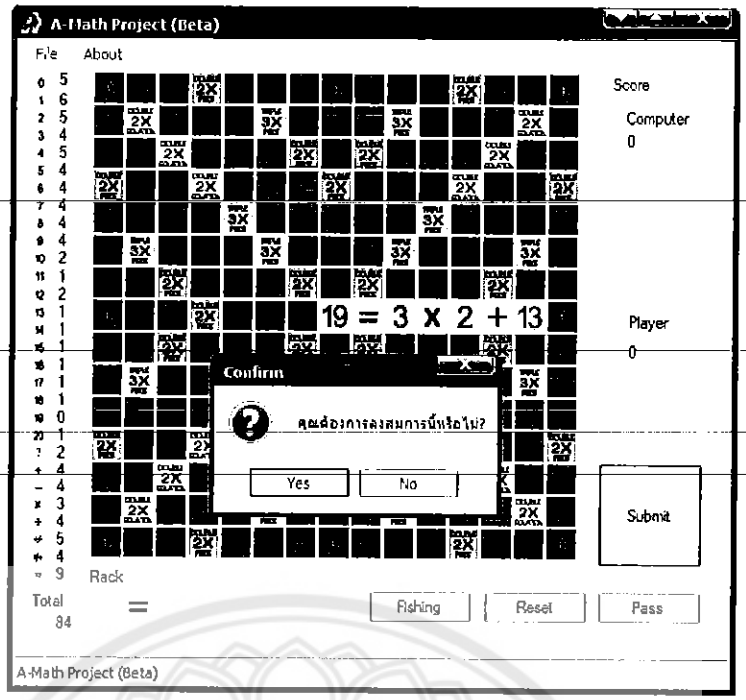
รูปที่ 4.6 แสดงการเตือนเมื่อลงเบี้ยครั้งแรก โดยไม่มีเบี้ยตัวใดเลยทับช่องสัญลักษณ์รูปดาว

ในการเริ่มเล่นเกมครั้งแรกฝ่ายผู้เล่นจะเป็นผู้เริ่มเกมก่อนเสมอ ซึ่งจากในแป้น (rack) ของผู้เล่นสามารถสลับเบี้ยให้เป็นสมการได้หลากหลายสมการ และเครื่องหมาย ? หรือเรียกอีกอย่างว่า แบล็ง (blank) จะเป็นเบี้ยที่สามารถกดเลือกให้เป็นตัวเลขใดก็ได้ตั้งแต่ 0 – 20 หรือเลือกเป็นเครื่องหมาย +, -, ×, ÷ และ = ก็ได้ เมื่อกดเลือกใช้แทนค่าแล้วเครื่องหมาย ? ก็จะเปลี่ยนเป็นค่าที่ได้เลือกและจะมีสีต่างจากเบี้ยตัวอื่นคือ มีสีน้ำเงิน ดังรูปที่ 4.4 ซึ่งฝ่ายผู้เล่นได้แบล็ง (blank) จำนวน 2 ตัวโดยเลือกให้ตัวแรกเป็นเครื่องหมาย + โดยจะปรากฏบนกระดานเป็นสีน้ำเงินหลังตัวเลข 2 และกำลังจะเลือกแทนค่าเบี้ยแบล็ง (blank) ตัวที่ 2 เป็นลำดับถัดมา เพื่อให้ได้สมการ {19 = 3×2+13}



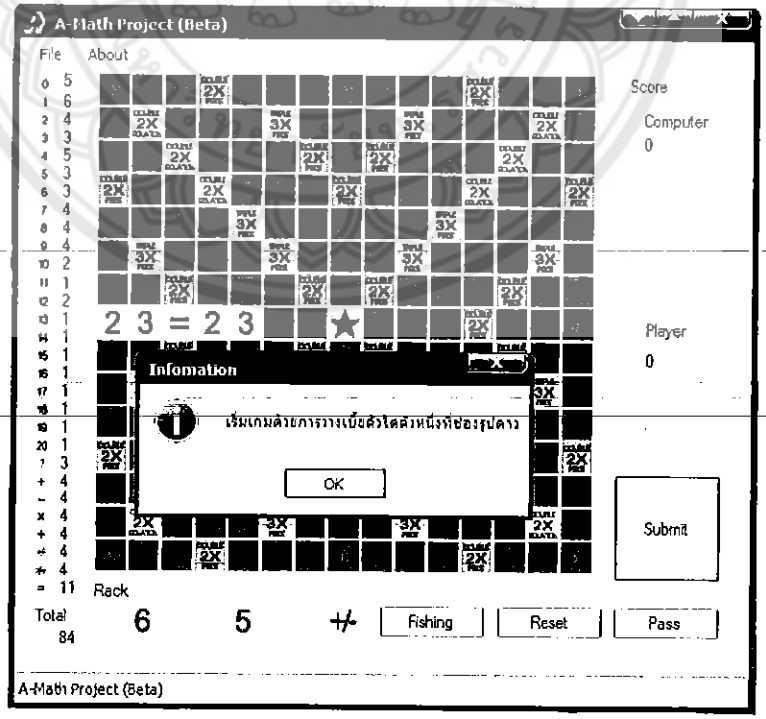
รูปที่ 4.4 แสดงการกดใช้เบี้ยแบล็ง (blank) ในการสร้างสมการ

ทำการลงสมการเรียบร้อยแล้วผู้เล่นจะกดปุ่มตกลง (submit) เพื่อเป็นการยืนยันการลงสมการในกระดาน เมื่อกดปุ่มแล้วจะมาหน้าต่างใหม่ปรากฏขึ้นให้เพื่อถามผู้เล่นว่าต้องการจะลงสมการนี้หรือไม่ หากผู้เล่นต้องการลงก็สามารถกดปุ่มตกลง (Yes) เพื่อยืนยันการลงสมการ หรือไม่ต้องการลงก็กดปุ่มยกเลิก (No) ดังรูปที่ 4.5



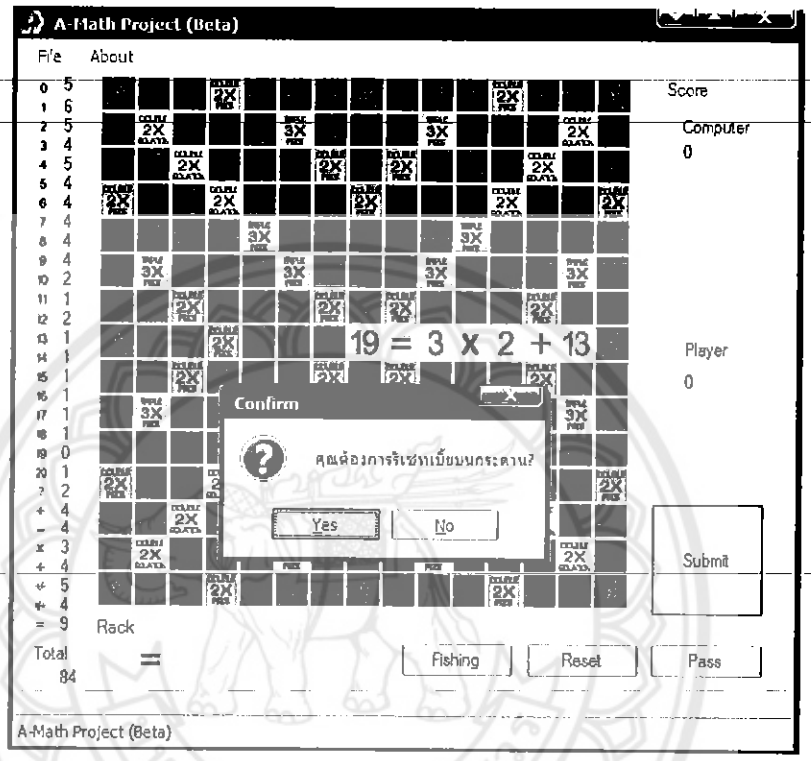
รูปที่ 4.5 แสดงวิธีการลงสมการ

หากเริ่มเล่นเกมครั้งแรกผู้เล่นทำการลงสมการลงบนกระดาน โดยที่ไม่มีเบี้ยตัวใดเลยทับช่องสัญลักษณ์รูปดาว เมื่อกดปุ่มตกลง (submit) จะปรากฏหน้าต่างเตือนผู้เล่นให้วางเบี้ยตัวใดตัวหนึ่งในสมการทับช่องสัญลักษณ์รูปดาว ดังรูปที่ 4.6



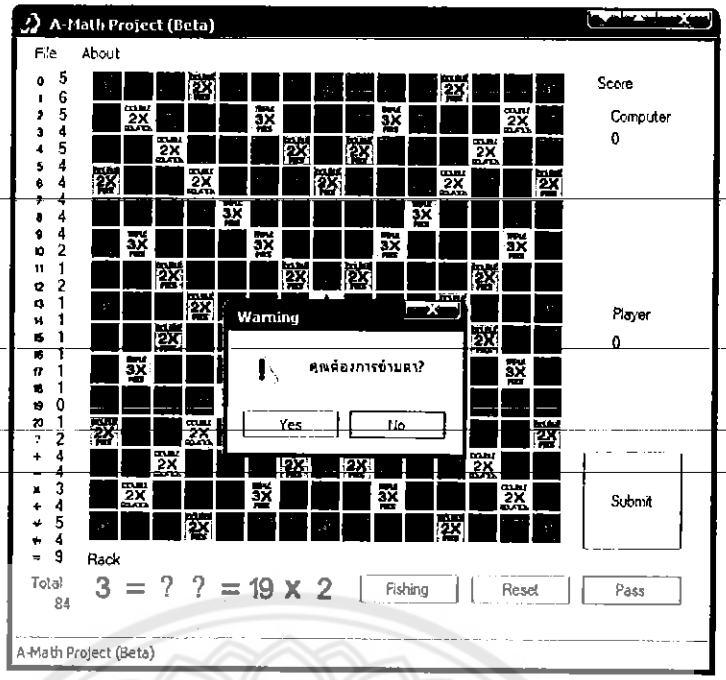
รูปที่ 4.6 แสดงการเตือนเมื่อลงเบี้ยครั้งแรก โดยไม่มีเบี้ยตัวใดเลยทับช่องสัญลักษณ์รูปดาว

หากผู้เล่นไม่ต้องการลงสมการที่ได้วางลงบนกระดานไปแล้ว โดยที่ยังไม่กดปุ่มตกลงเพื่อยืนยันการลงสมการ สามารถกดปุ่มคืนค่า (reset) เบี้ยที่วางอยู่บนกระดานก็จะคืนค่ากลับมาอยู่ในแป้น (rack) ของผู้เล่นใหม่อีกครั้ง ซึ่งค่าของเบี้ยพิเศษก็จะคืนค่ากลับมาเป็นเหมือนแรกเริ่มก่อนที่จะทำการเลือกแทนค่า เมื่อกดปุ่ม reset แล้วจะมีหน้าต่างใหม่ปรากฏขึ้นมาเพื่อให้ยืนยันการ reset ค่าของเบี้ยที่ลงไว้บนกระดาน ดังรูปที่ 4.7



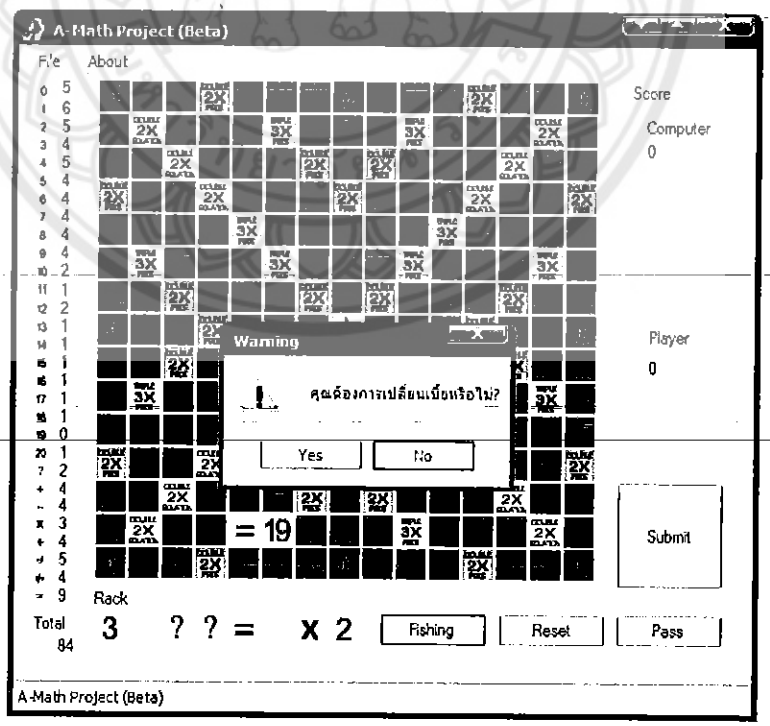
รูปที่ 4.7 แสดงการคืนค่าเบี้ยที่อยู่บนกระดานกลับมาอยู่บนแป้น

หากผู้เล่นไม่ต้องการลงสมการใดเลยก็สามารถกดปุ่มผ่าน (pass) เพื่อข้ามไปเป็นรอบ (turn) ของคอมพิวเตอร์ได้ ซึ่งเมื่อกดปุ่มแล้วจะปรากฏหน้าตาใหม่ขึ้นมาเพื่อให้ผู้เล่นกดยืนยันการข้ามรอบ (turn) ของตนเอง ดังรูปที่ 4.8



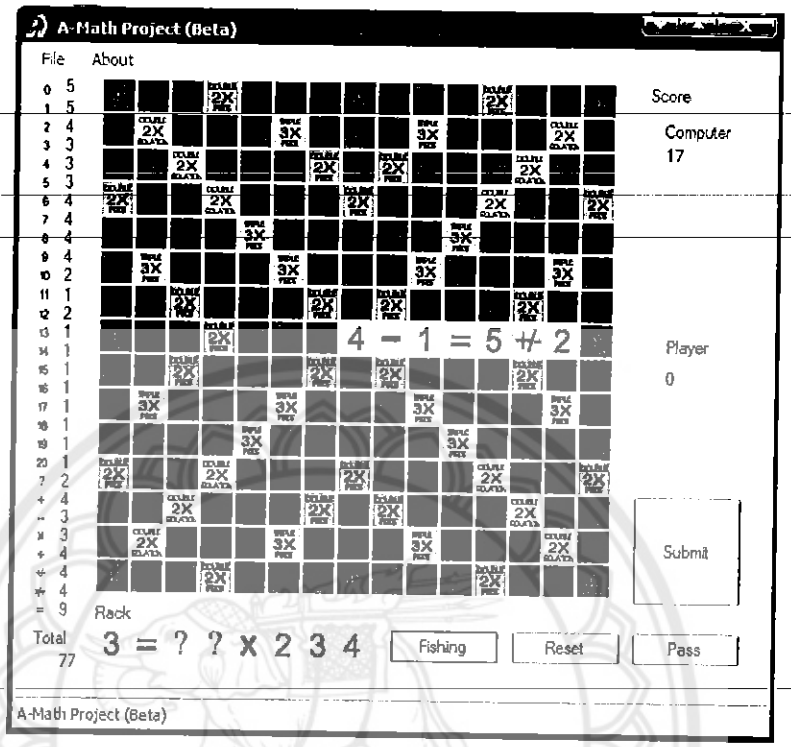
รูปที่ 4.8 แสดงการผ่านรอบของผู้เล่น

เมื่อผู้เล่นต้องการเปลี่ยนเบี้ยในแป้น (rack) ก็สามารถทำได้โดยการกดเลือกเบี้ยที่ต้องการเปลี่ยนมาวางไว้บนกระดานแล้วกดปุ่มเปลี่ยนเบี้ย (fishing) เมื่อกดปุ่มแล้วจะปรากฏหน้าต่างขึ้นมาให้ผู้เล่นกดปุ่มเพื่อยืนยันการเปลี่ยนเบี้ย ดังรูปที่ 4.9



รูปที่ 4.9 แสดงการเปลี่ยนเบี้ยบนแป้นของผู้เล่น

เมื่อทำการเปลี่ยนเบี้ยบนแป้น (rack) แล้วจะเป็นการผ่านรอบ (turn) ของผู้เล่นให้เป็นรอบของคอมพิวเตอร์ แล้วเบี้ยที่ผู้เล่นเปลี่ยนจะหายไปแล้วจะได้รับเบี้ยใหม่โดยปรากฏขึ้นมาบนแป้น (rack) ซึ่งผู้เล่นได้ทำการเปลี่ยนเบี้ย {=, 19} แล้วก็จะได้เบี้ยใหม่จากการสุ่มเป็น {3, 4} ดังรูปที่ 4.10



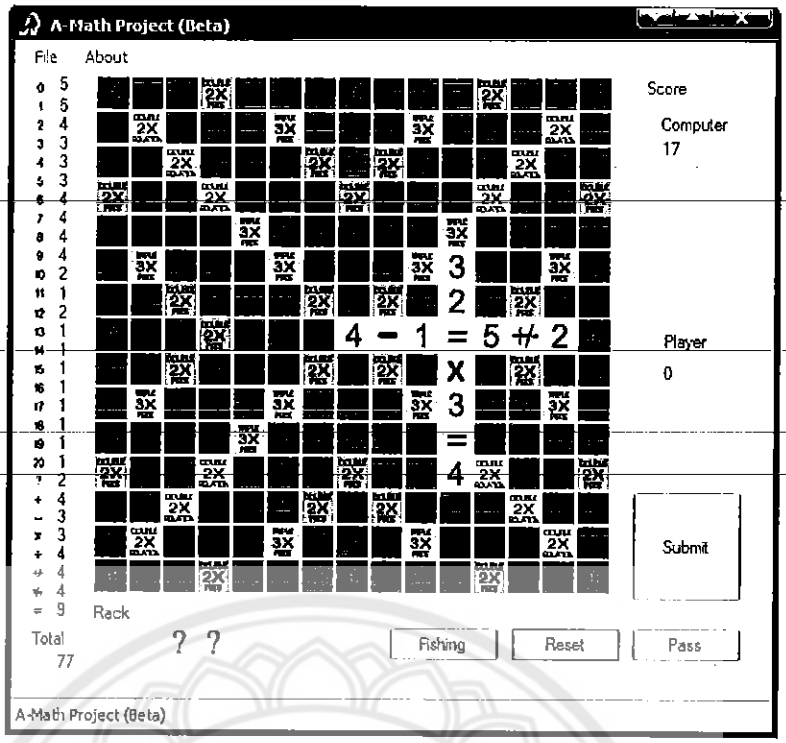
รูปที่ 4.10 แสดงเบี้ยใหม่ของผู้เล่นเมื่อทำการเปลี่ยนเบี้ย

เมื่อผู้เล่นวางเบี้ยลงบนกระดานโดยเป็นสมการที่ไม่ถูกต้อง เมื่อกดปุ่มตกลง (submit) และกดปุ่มยืนยันการลงสมการ จะปรากฏหน้าต่างขึ้นมาเตือนว่าผู้เล่นได้ลงสมการไม่ถูกต้อง แล้วเบี้ยที่อยู่บนกระดานจะถูกคืนค่ากลับมาอยู่บนแป้น (rack) ซึ่งสามารถแบ่งรูปแบบของสมการที่ไม่ถูกต้องได้ คือ

- การเรียงสมการที่ไม่สามารถคำนวณได้ เช่น {32 = ×3 = 4} ดังรูปที่ 4.11

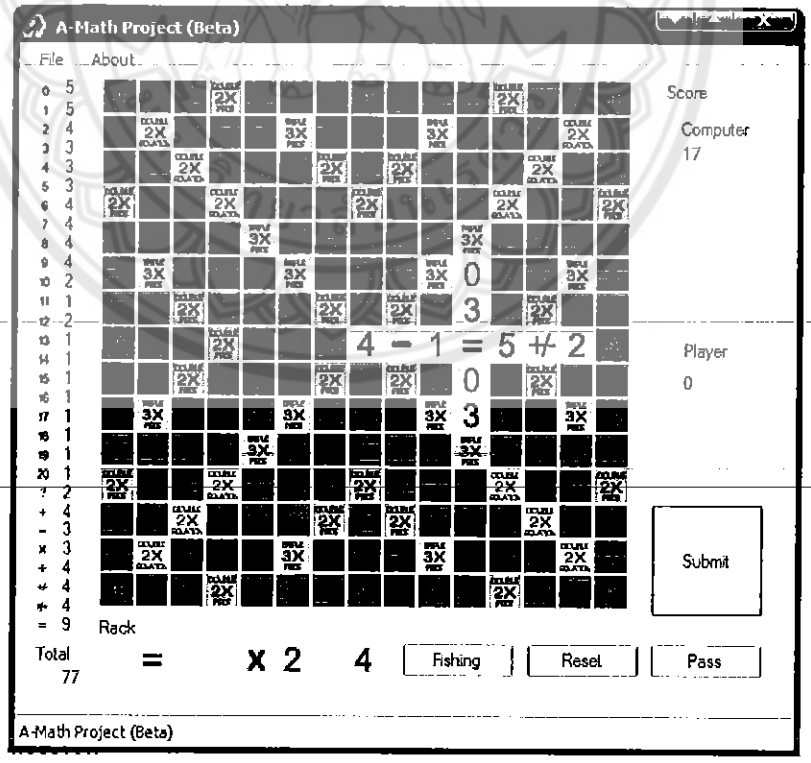
14993287

วร.
ว517 2551



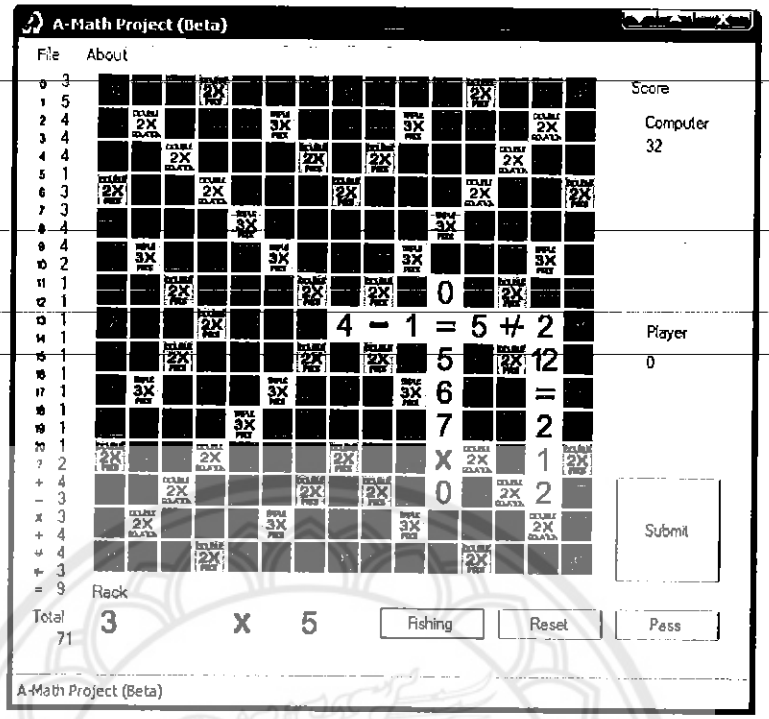
รูปที่ 4.11 แสดงการเรียงสมการที่ไม่สามารถคำนวณได้

- การวางเลขศูนย์ไว้ด้านหน้าตัวเลข เช่น {03 = 03} ดังรูปที่ 4.12



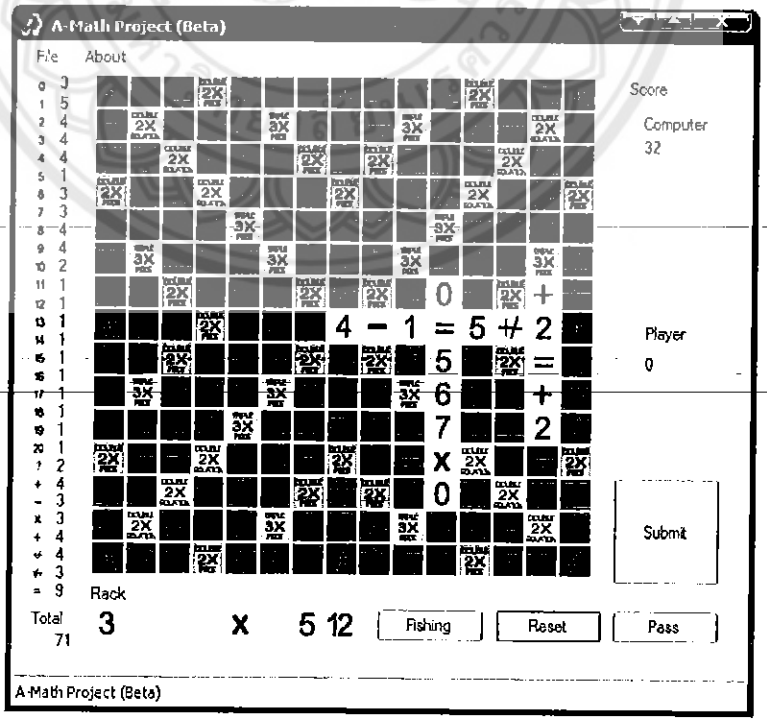
รูปที่ 4.12 แสดงการวางเลขศูนย์ไว้ด้านหน้าตัวเลข ซึ่งเป็นสมการที่ไม่ถูกต้อง

- การวางเลขหลักหน่วยต่อเลขหลักสิบ เช่น {2(12) = 212} ดังรูปที่ 4.13



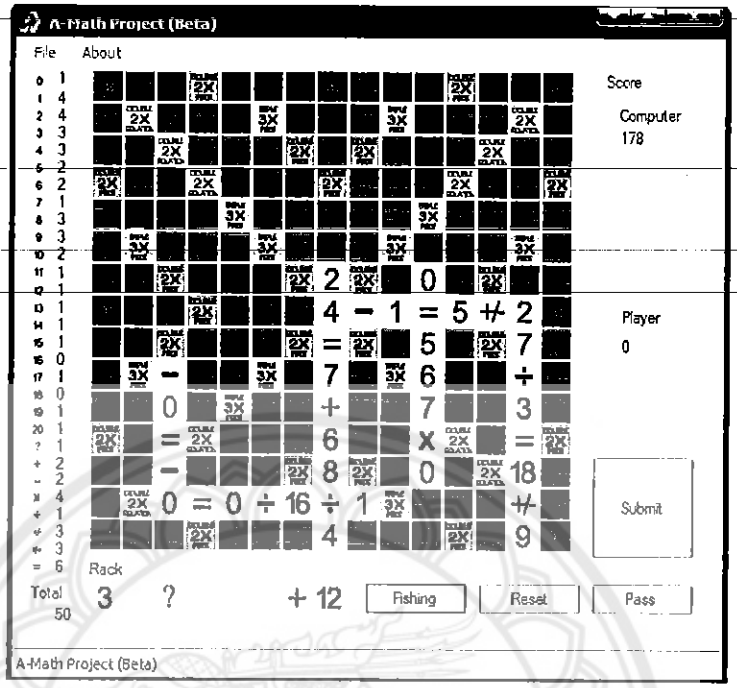
รูปที่ 4.13 แสดงการวางเลขหลักหน่วยต่อเลขหลักสิบ ซึ่งเป็นสมการที่ไม่ถูกต้อง

- ไม่สามารถวางเครื่องหมายบวกไว้ด้านหน้าตัวเลขใดๆได้ เช่น {+2 = +2} ดังรูปที่ 4.14



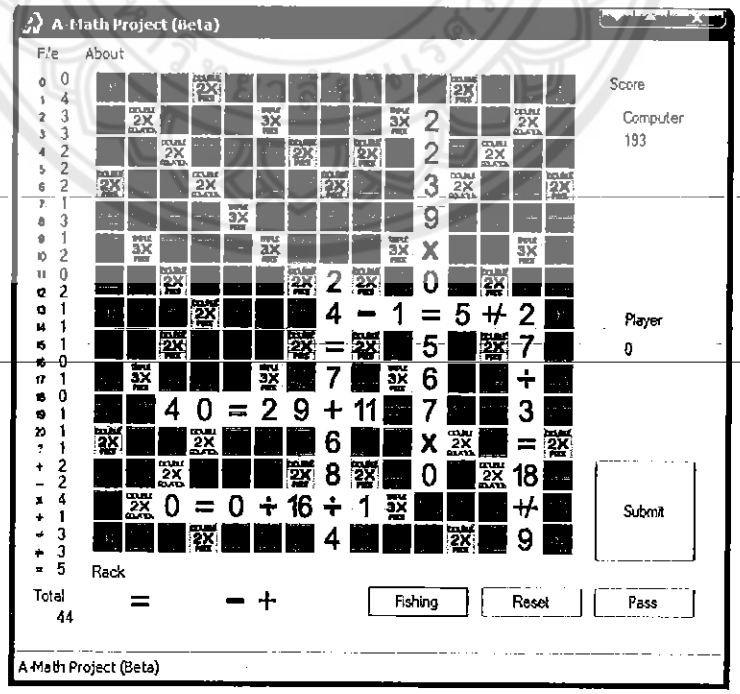
รูปที่ 4.14 แสดงการวางเครื่องหมายบวกไว้ด้านหน้าตัวเลขใดๆได้ ซึ่งเป็นสมการที่ไม่ถูกต้อง

- สามารถวางเครื่องหมายลบด้านหน้าเลขใดๆก็ได้ แต่ไม่สามารถวางเครื่องหมายลบหน้าเลขศูนย์ได้ ดังรูปที่ 4.15



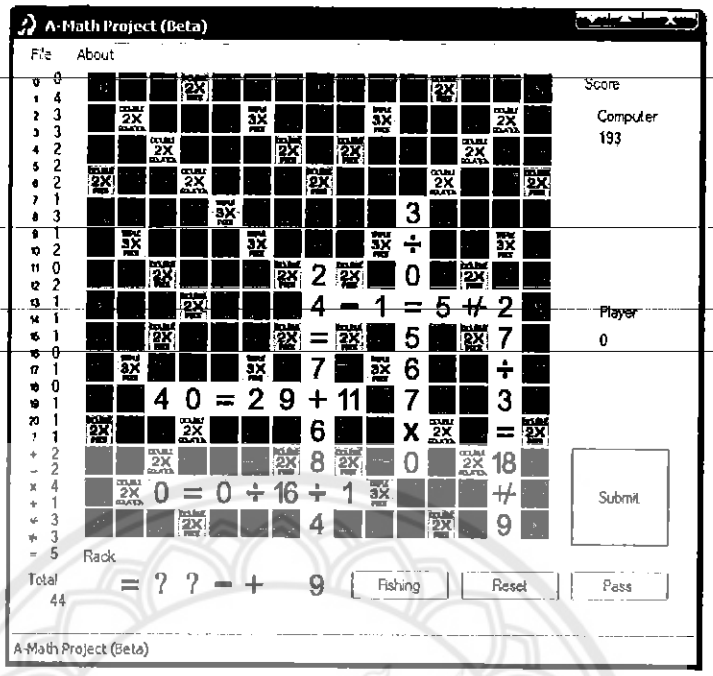
รูปที่ 4.15 แสดงการวางเครื่องหมายลบหน้าเลขศูนย์ ซึ่งเป็นสมการที่ไม่ถูกต้อง

- ไม่สามารถวางเบี้ยหลักหน่วยเรียงกัน 4 ตัวได้ เช่น {2239×0 = 567×0} ดังรูปที่ 4.16



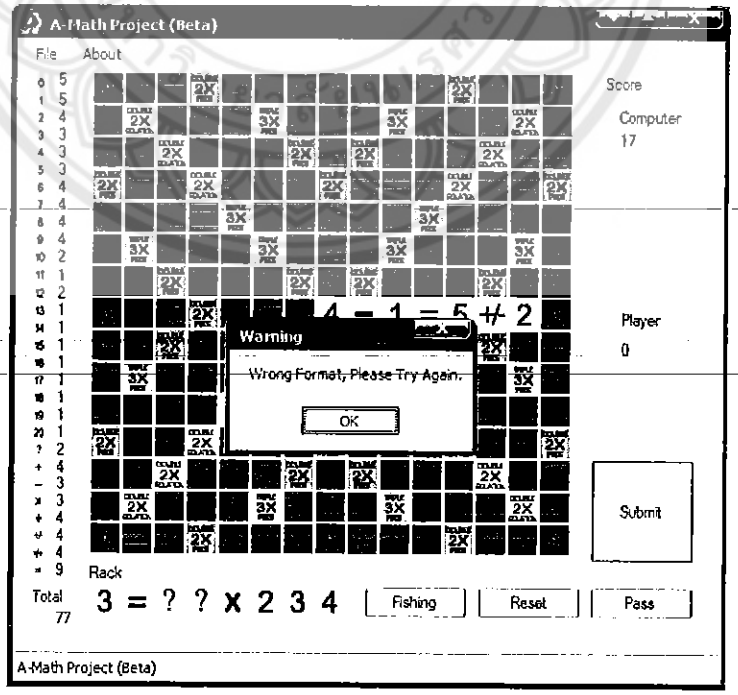
รูปที่ 4.16 แสดงการวางเบี้ยหลักหน่วยเรียงกัน 4 ตัว ซึ่งเป็นการผิดกฎการเล่นเกมเอแมท

- ไม่สามารถเรียงสมการให้ศูนย์หารตัวเลขใดๆได้ เช่น {3÷0 = 567×0} ดังรูปที่ 4.17



รูปที่ 4.17 แสดงการเรียงสมการให้ศูนย์หารตัวเลขใดๆ ซึ่งเป็นสมการที่ไม่ถูกต้อง

เมื่อผู้เล่นเรียงสมการที่ไม่ถูกต้องลงบนกระดาน เมื่อกดปุ่มตกลง (submit) จะปรากฏหน้าตาขึ้นเตือนผู้เล่นว่าสมการที่ต้องการลงนั้นไม่ถูกต้อง ดังรูปที่ 4.18

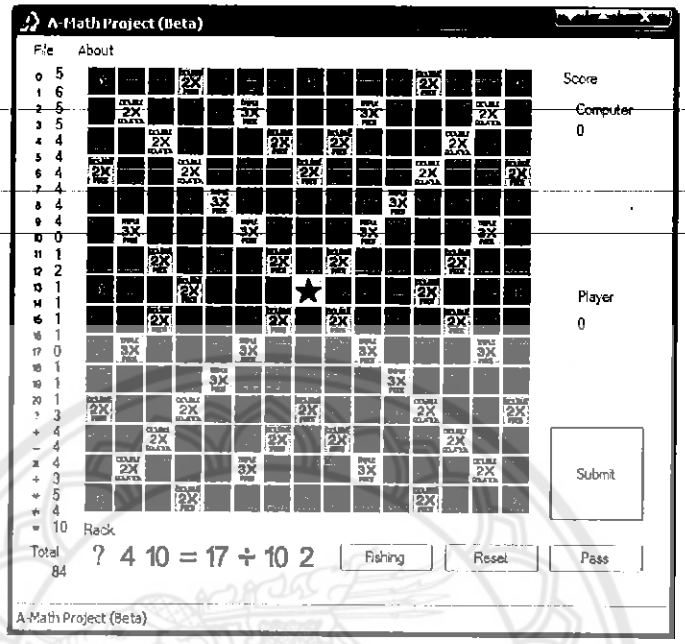


รูปที่ 4.18 แสดงการเตือนเมื่อผู้เล่นลงสมการที่ไม่ถูกต้อง

4.2 ผลการทดลองภาคปฏิบัติ

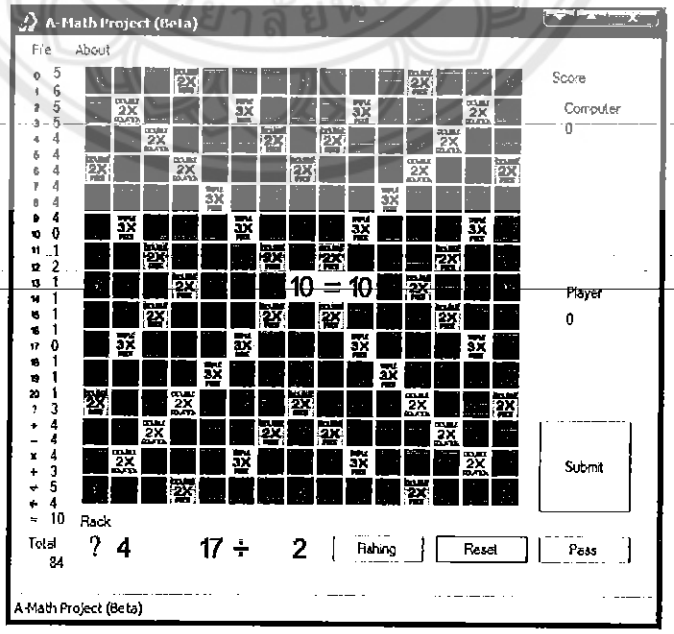
จากการทดลองโดยการลองเล่นเกมเอแมทกับคอมพิวเตอร์ แล้วจะได้ผลดังนี้

เมื่อเริ่มเกมครั้งแรกจะปรากฏหน้าต่างโปรแกรม ดังรูปที่ 4.19



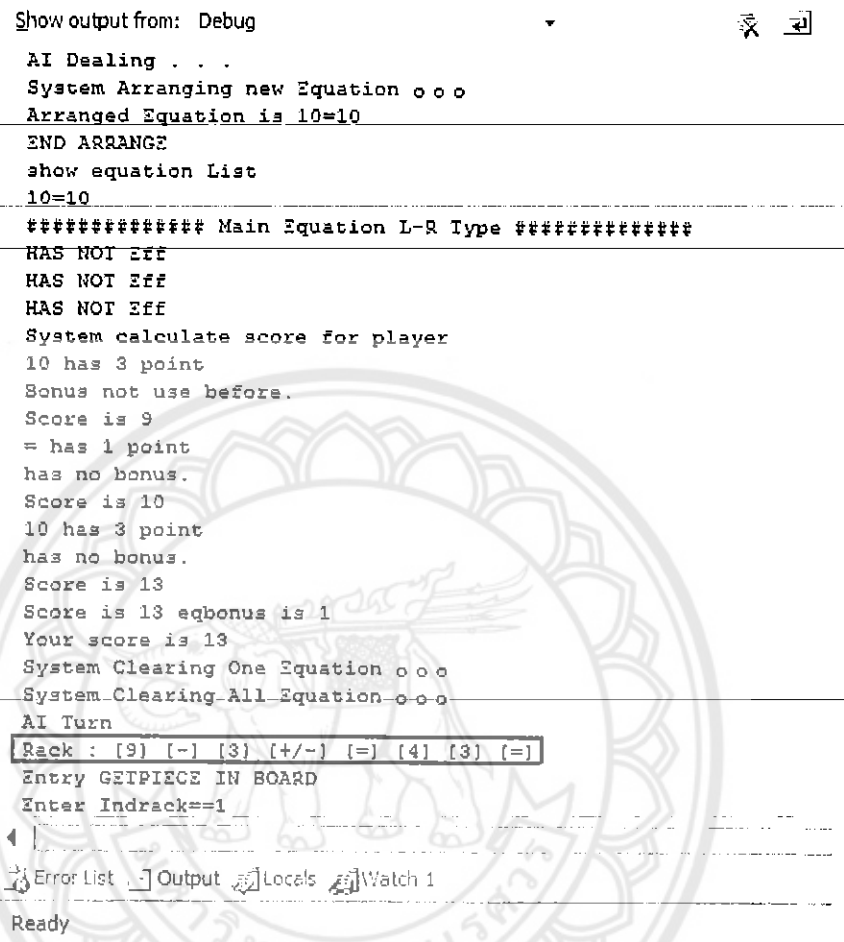
รูปที่ 4.19 แสดงหน้าต่างโปรแกรมเมื่อเริ่มเล่นเกมครั้งแรก

ผู้เล่นเรียงสมการแล้วจึงวางเบี้ยลงบนกระดาน โดยให้เบี้ยตัวใดตัวหนึ่งทับช่องสัญลักษณ์รูปดาว ซึ่งผู้เล่นได้ลงสมการ {10 = 10} ดังรูปที่ 4.20



รูปที่ 4.20 แสดงการลงสมการของผู้เล่นในการเล่นเกมครั้งแรก

เมื่อฝ่ายผู้เล่นลงสมการและกดปุ่มตกลง (submit) แล้วก็จะเป็นรอบของคอมพิวเตอร์โดยในแป้น (rack) ของคอมพิวเตอร์จะมีเบี้ยได้แก่ {9, -, 3, +/-, =, 4, 3, =} ดังรูปที่ 4.21

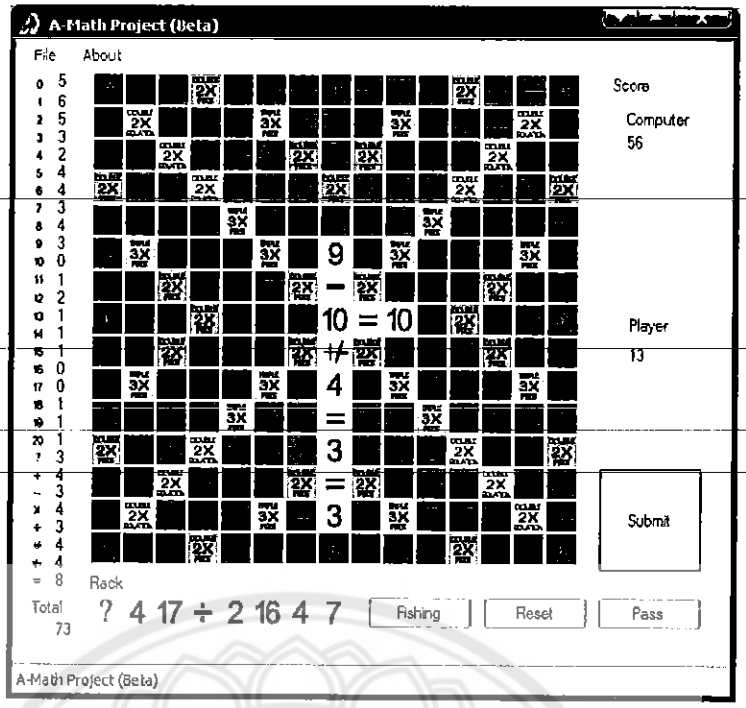


รูปที่ 4.21 แสดงเบี้ยของคอมพิวเตอร์เมื่อจบรอบของผู้เล่น

หลังจากนั้นคอมพิวเตอร์จะทำการคำนวณเบี้ยที่อยู่ในแป้น (rack) ของคอมพิวเตอร์เองและนำเบี้ยที่อยู่บนกระดานที่สามารถคำนวณได้มาคำนวณรวมด้วย จากผลการคำนวณสุดท้ายคอมพิวเตอร์จะได้สมการ {9-10+4 = 3 = 3} ดังรูปที่ 4.22

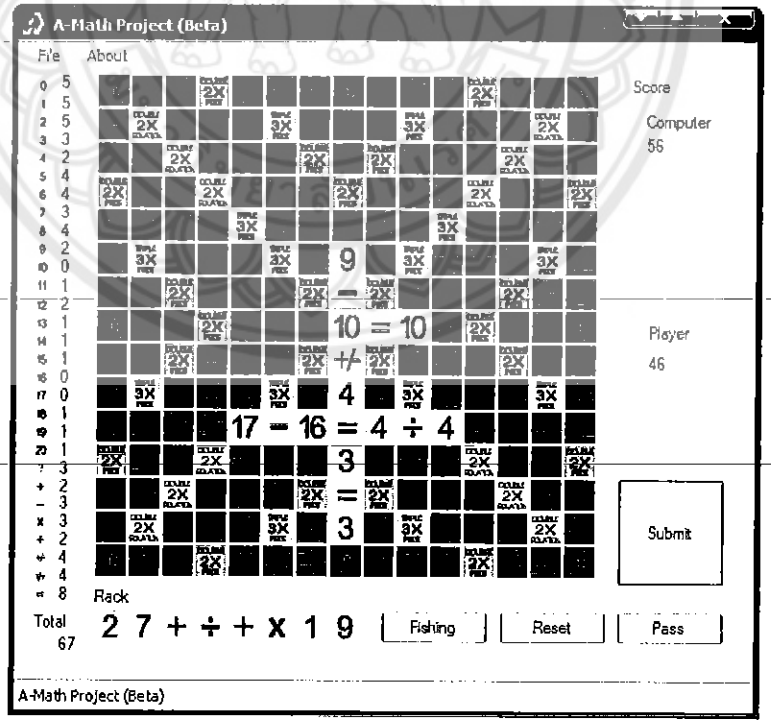
รูปที่ 4.22 แสดงสมการที่คอมพิวเตอร์คำนวณได้

เมื่อคอมพิวเตอร์ทำการคำนวณสมการเสร็จจะทำการลงสมการบนกระดาน ดังรูปที่ 4.23



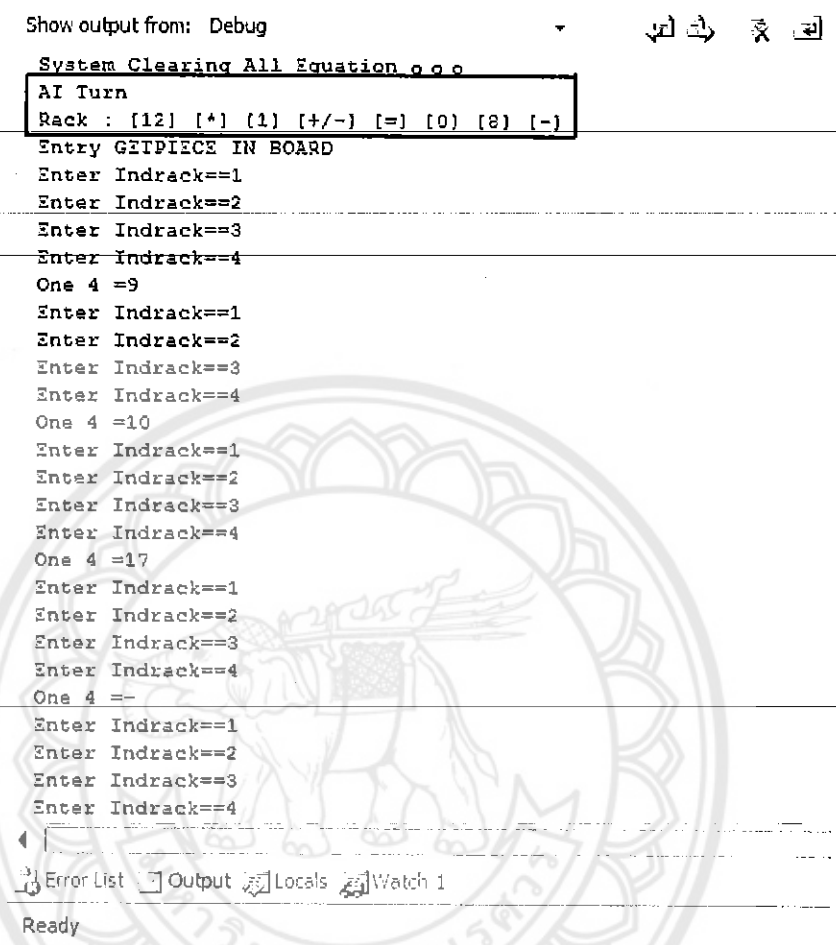
รูปที่ 4.23 แสดงการลงเบี้ยบนกระดานของคอมพิวเตอร์

หลังจากจบรอบ (turn) ของคอมพิวเตอร์จะเข้าสู่รอบ (turn) ของผู้เล่น โดยผู้เล่นได้ลงสมการ {17-16 = 4÷4} ดังรูปที่ 4.24



รูปที่ 4.24 แสดงการลงสมการผู้เล่นต่อจากคอมพิวเตอร์

เมื่อจบรอบ (turn) ของผู้เล่น คอมพิวเตอร์จะรับค่าบนกระดานมาทำการคำนวณใหม่ กับเบี้ยที่อยู่ในแป้น (rack) ได้แก่ {12, ×, 1, +/-, = , 0, 8, -} ดังรูปที่ 4.25

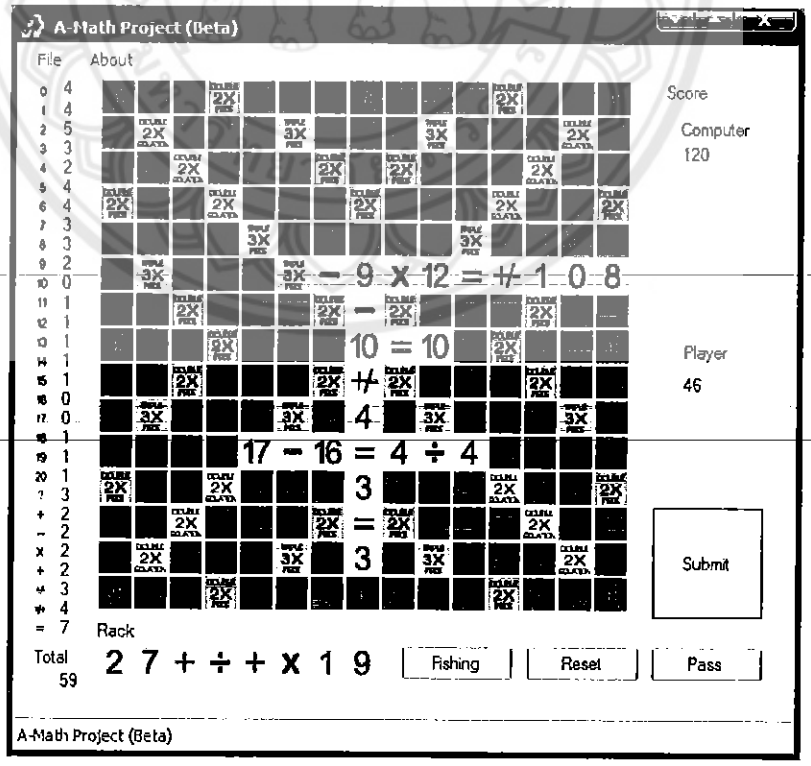


รูปที่ 4.25 แสดงเบี้ยของคอมพิวเตอร์หลังจากที่ได้รับเบี้ยมาใหม่

หลังจากนั้นคอมพิวเตอร์จะนำเบี้ยในแป้น (rack) และเบี้ยบนกระดานที่สามารถคำนวณได้มาคำนวณหาสมการใหม่ที่จะลง ซึ่งสมการใหม่ คือ {-9×12 = -108} ดังรูปที่ 4.26 และแสดงผลในการลงสมการดังรูปที่ 4.27

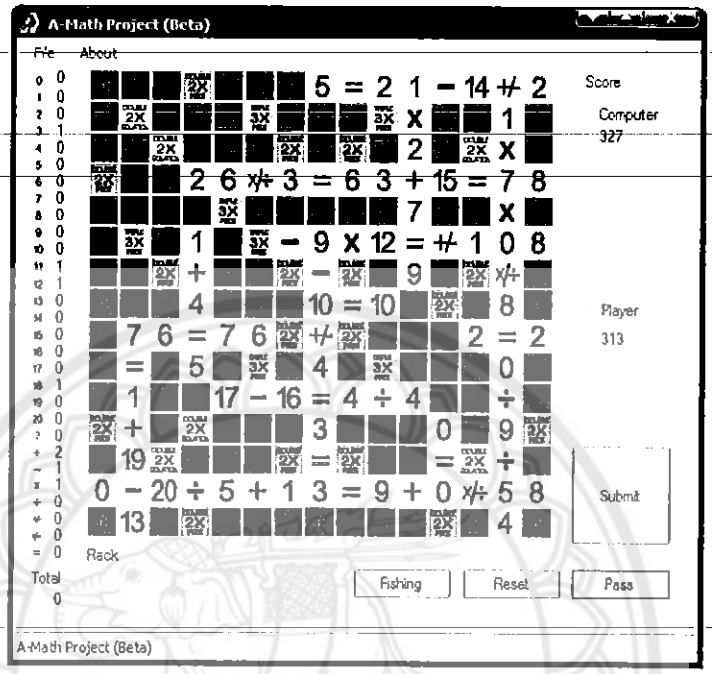
```
Show output from: Debug
show equation List
9-8=1-0*12
check one
one = 9  new = 9
count = 0
First Piece Board
Cant play.
show equation List
9-8=0*12+1
check one
one = 9  new = 9
count = 0
First Piece Board
Cant play.
show equation List
-9*12=-108
check one
one = 9  new = -
count = 1
Middle Piece Board
Try Left
newPiece.Data= -
newPiece.RowPos = 7
newPiece.CollummPos = 7
Try Right
newPiece.Data= *
newPiece.RowPos = 6
newPiece.CollummPos = 9
Error List  Output  Locals  Watch 1
Ready
```

รูปที่ 4.26 แสดงสมการที่คอมพิวเตอร์คำนวณได้ใหม่

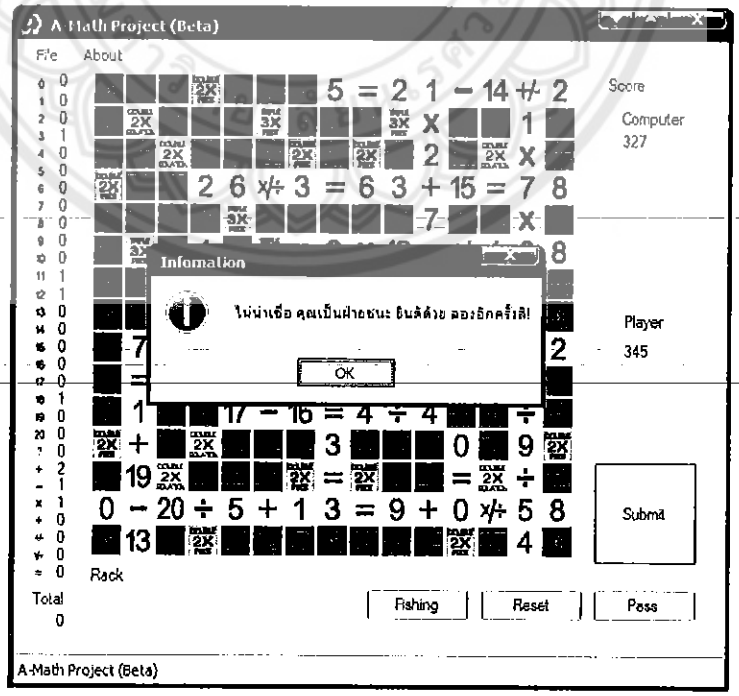


รูปที่ 4.27 แสดงการลงสมการของคอมพิวเตอร์

หลังจากที่เล่นจนถึงในรอบ (turn) สุดท้ายโดยผู้เล่นสามารถลงเบี้ยจนหมดแป้น (rack) ได้ จะทำให้ผู้เล่นได้คะแนนพิเศษโดยการนำเอาคะแนนจากเบี้ยที่เหลือในแป้น (rack) ของคอมพิวเตอร์มาคูณด้วยสองแล้วบวกกับคะแนนของผู้เล่น ดังรูปที่ 4.28 และ แสดงผลจากการบวกคะแนนเพิ่มให้กับฝ่ายผู้เล่น พร้อมกับผลการแพ้ชนะ ดังรูปที่ 4.29



รูปที่ 4.28 แสดงการลงเบี้ยของผู้เล่นจนหมดแป้น (rack) ในรอบ (turn) สุดท้าย

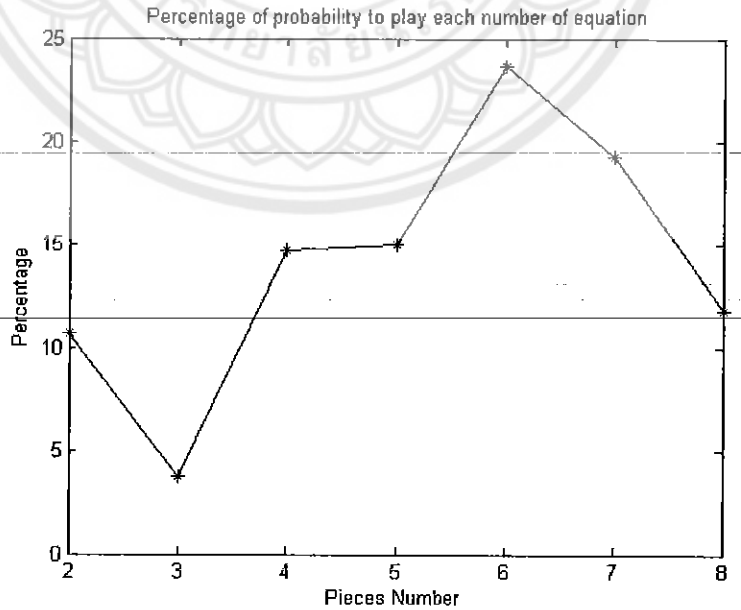


รูปที่ 4.29 แสดงผลแพ้ชนะในการเล่นเกมเอแมท

จากการทดลองเล่นเกมเอแมทกับคอมพิวเตอร์เป็นจำนวน 10 เกม ทำให้ได้ผลการทดลองดังตารางที่ 4.1 และเขียนเป็นกราฟได้ดังรูปที่ 4.30

ตารางที่ 4.1 แสดงเปอร์เซ็นต์การลงสมการของคอมพิวเตอร์ต่อจำนวนเบี้ยต่อการเล่นเกมในครั้งหนึ่งๆ ซึ่งมีจำนวนเบี้ย คือ 2 ตัว, 3 ตัว, 4 ตัว, 5 ตัว, 6 ตัว, 7 ตัว และ 8 ตัว และจำนวนในการชนะ

เกมที่	เปอร์เซ็นต์การลงสมการของคอมพิวเตอร์ต่อจำนวนเบี้ย							ผู้ชนะ	
	2 ตัว	3 ตัว	4 ตัว	5 ตัว	6 ตัว	7 ตัว	8 ตัว	ผู้เล่น	คอมพิวเตอร์
1	8.33	8.33	16.67	25	8.33	16.67	16.67		✓
2	22.22	-	22.22	-	11.11	22.22	22.22		✓
3	18.18	9.09	18.18	18.18	9.09	18.18	9.09	✓	
4	-	20	10	-	50	10	10		✓
5	12.5	-	-	-	37.5	12.5	37.5		✓
6	-	-	11.11	33.33	22.22	22.22	11.11	✓	
7	9.09	-	18.18	18.18	36.36	18.18	-	✓	
8	14.28	-	28.57	-	28.57	28.57	-	✓	
9	11.11	-	11.11	33.33	22.22	22.22	-	✓	
10	11.11	-	11.11	22.22	11.11	22.22	11.11	✓	
รวม	10.682	3.742	14.715	15.024	23.651	19.298	11.77	6	4



รูปที่ 4.30 แสดงกราฟของเปอร์เซ็นต์การลงสมการของคอมพิวเตอร์ต่อจำนวนเบี้ย

จากกราฟในรูปที่ 4.30 จะแสดงจำนวนเบี้ยของสมการที่คอมพิวเตอร์สามารถลงได้ใน 1 เกมโดยกำหนดให้ค่าของจำนวนเกมมีค่าดังนี้ 0-8 ซึ่งสรุปได้ว่า จำนวนเบี้ยของสมการที่คอมพิวเตอร์สามารถคำนวณได้ต่อการลงนั้นอยู่ในช่วง 4 – 7 ตัวเป็นส่วนมาก เนื่องจากในอัลกอริทึมได้มีการกำหนดให้คอมพิวเตอร์ลงสมการที่สามารถคำนวณแล้วได้จำนวนเบี้ยที่มีความยาวที่สุด ซึ่งทำให้ช่วงที่ลงสมการที่มีจำนวน 2 ตัว และ 3 ตัว มีเปอร์เซ็นต์การลงน้อยกว่าช่วง 4 – 7 ตัว และในการลง 8 ตัวที่มีการลงสมการได้น้อยนั้น อาจจะเป็นเพราะว่าตัวเบี้ยที่อยู่ในแป้น (rack) ไม่อำนวยต่อการเรียงเป็นสมการได้หรือพื้นที่ว่างบนกระดานไม่เพียงพอต่อการสร้างสมการ

บทที่ 5
สรุปผลและข้อเสนอแนะ

เอแมท (A-Math) เป็นเกมกระดานประเภทหนึ่ง ซึ่งมีลักษณะคล้ายกับเกมปริศนาอักษรไขว้ (crossword) โดยเกมเอแมทนี้จะเป็นการต่อสมการคณิตศาสตร์ หลักในการเล่นคือต่อตัวเลขตามหลักการทางคณิตศาสตร์ลงบนช่องตาราง สำหรับการเล่นจะแข่งกันทำคะแนนระหว่างผู้เล่น 2 ฝ่าย โดยที่โปรแกรมที่พัฒนานั้นจะมีหน้าที่เล่นตอบสนองกับผู้ใช้ ทำให้สามารถเล่นกับคอมพิวเตอร์เพื่อเป็นการฝึกฝนทักษะของตนได้ ซึ่งกฎและกติกาของเกมเอแมทที่นำมาใช้ในโครงงานนี้เป็นการเล่นเกมตามมาตรฐานสากลที่ใช้แข่งขันกันในปัจจุบัน

ทฤษฎีที่นำมาใช้ในการคิดสมการและลงสมการที่คิดได้ ลงไปในกระดานเพื่อทำคะแนนนั้น คืออัลกอริทึมการค้นหาแบบลำดับ (Linear Search Algorithm) ซึ่งเป็นอัลกอริทึมที่ใช้ในการหาตำแหน่งของเบี้ยบนกระดานที่สามารถลงสมการได้ และอัลกอริทึมแบบย้อนรอย (Backtracking Algorithm) ที่นำมาใช้ในการสลับเบี้ยเพื่อสร้างสมการแล้วนำมาลงต่อกับเบี้ยบนกระดานมนตำแหน่งที่ได้ทำการค้นหาไว้แล้ว โดยอัลกอริทึมแบบย้อนรอย (Backtracking Algorithm) ที่นำมาใช้ในโครงงานนี้จะเป็นการค้นหาแบบย้อนกลับ (Backtracking Search)

ขั้นตอนการทำงานของโปรแกรมจะเริ่มเมื่อถึงรอบของคอมพิวเตอร์ โดยจะเริ่มจากการค้นหาเบี้ยบนกระดาน เพื่อหาตำแหน่งเบี้ยบนกระดานที่มีโอกาสลงได้ และเก็บเบี้ยตัวนั้นมาไว้พิจารณาในการสลับสมการเพื่อที่จะทำการลงสมการ ต่อจากนั้นจึงใช้อัลกอริทึมแบบย้อนรอย (Backtracking Algorithm) สลับเบี้ยบนแป้น (rack) ของตัวเองรวมทั้งคิดร่วมกับเบี้ยในตำแหน่งที่จะทำการลง ซึ่งการทำแบบย้อนกลับ (backtracking) จะแบ่งการคิดออกเป็นการคิดสมการที่มีขนาดตั้งแต่ 3 – 9 ตัว โดยที่จะเลือกลงสมการที่มีความยาวมากๆก่อน

5.1 สรุปผลการทดลอง

จากการทดลองสรุปได้ว่าคอมพิวเตอร์สามารถทำการต่อสมการที่มีความยาวเบี้ยจำนวน 3 ตัว, 4 ตัว, 5 ตัว, 6 ตัว, 7 ตัว, 8 ตัวและ 9 ตัวได้ ทั้งนี้การต่อสมการในความยาวต่างๆมักจะขึ้นอยู่กับเบี้ยบนแป้น (rack) ของคอมพิวเตอร์และเบี้ยบนกระดาน ถ้าเบี้ยบนแป้น (rack) ของคอมพิวเตอร์สามารถสลับไปมาเพื่อที่จะนำมาต่อเป็นสมการกับเบี้ยที่อยู่บนกระดานได้หลายสมการ โอกาสที่คอมพิวเตอร์จะลงสมการที่มีความยาวหลายตัวก็จะมีสูงตาม และถ้าคอมพิวเตอร์ไม่สามารถต่อสมการใหม่ได้เลย หรือไม่มีเครื่องหมายเท่ากับคอมพิวเตอร์จะทำการต่อสมการต่อยอดจากสมการเดิมบนกระดาน และถ้าคอมพิวเตอร์ไม่สามารถลงสมการได้เลยคอมพิวเตอร์จะทำการเปลี่ยนเบี้ย (fishing) และเปลี่ยนกลับมาเป็นรอบ (turn) ของผู้เล่นตามที่ได้โปรแกรมแล้วอย่างสมบูรณ์

5.2 ปัญหาและอุปสรรค

1. การเลือกใช้อัลกอริทึมที่จะนำมาประยุกต์ใช้กับคอมพิวเตอร์ให้สามารถหาคำตอบได้อย่างรวดเร็วนั้น เนื่องจากจำนวนเบี้ยบนแป้น (rack) ที่มีขนาดสูงสุดถึง 8 ตัวและเบี้ยบนกระดานทำให้ตอนแรกในการใช้ Brute Force เพื่อหาความเป็นไปได้ทั้งหมดของการเรียงเบี้ยจะพบว่าใช้เวลานานในการคำนวณในแต่ละสมการ อีกทั้งถ้าเบี้ยเป็นลักษณ์พิเศษ เช่น เบี้ยที่เลือกได้ว่าจะเป็นเครื่องหมายบวกหรือลบ คูณหรือหาร และเป็นตัวใดๆก็ได้ ในจำนวนเบี้ยทั้งหมด จะทำให้ใช้เวลาในการคำนวณเพิ่มมากขึ้นไปอีก ต่อจากนั้นจึงได้ทดลองใช้อัลกอริทึมแบบย้อนรอย (Backtracking Algorithm) ซึ่งกำหนดเงื่อนไขในการต่อเบี้ยแต่ละตัวเข้าด้วยกัน จึงช่วยทำให้ตัดการคำนวณที่ไม่จำเป็นออกไปได้มากขึ้น ทำให้ได้ผลลัพธ์ของสมการที่ถูกต้องอย่างรวดเร็ว เมื่อเทียบกับวิธีการ Brute Force ในการเลือกใช้อัลกอริทึมในครั้งแรก

2. เนื่องจากการเล่นเกมเอแมทเป็นไปในลักษณะผลัดกันเล่นระหว่างผู้เล่นกับคอมพิวเตอร์ ซึ่งในตอนแรกระหว่างที่คอมพิวเตอร์ทำการคำนวณสมการ โปรแกรมจะไม่สามารถตอบสนองอะไรต่อผู้เล่นได้ จึงทำให้ต้องรอให้คอมพิวเตอร์คำนวณให้เสร็จก่อน ผู้เล่นจึงจะสามารถใช้งานโปรแกรมได้ ทำให้ผู้เล่นไม่สามารถสลับเบี้ยในแป้น (rack) ของตนเองเพื่อที่จะคิดคำนวณสมการที่จะลงในระหว่างที่คอมพิวเตอร์กำลังคำนวณสมการได้ ต่อมาจึงแก้ไขโดยการใช้ระบบทำงานเป็นเบื้องหลัง (working background) ทำให้โปรแกรมสามารถตอบสนองกับผู้เล่นระหว่างที่คอมพิวเตอร์คำนวณสมการไปด้วยได้โดยไม่ต้องรอให้คอมพิวเตอร์คำนวณเสร็จ

5.3 ข้อเสนอแนะ

หลักการในการลงสมการของคอมพิวเตอร์ในโปรแกรมนี้จะเป็นแบบที่เลือกสมการที่มีความยาวมากที่สุด ที่สามารถคำนวณได้มาลงทำคะแนน ซึ่งส่วนมากแล้วจะได้คะแนนที่สูง แต่ในบางครั้งสมการที่ไม่ยาวถ้าวางทับช่องโบนัสพิเศษที่จะคูณคะแนน อาจจะทำให้ได้คะแนนมากกว่า ซึ่งถ้าสามารถปรับปรุงหลักการเลือกสมการโดยให้หาที่ลงบนกระดานของคอมพิวเตอร์ให้ทับช่องโบนัสพิเศษที่คูณคะแนนแล้วอาจจะทำให้คอมพิวเตอร์มีประสิทธิภาพและสามารถทำคะแนนได้เพิ่มมากขึ้น

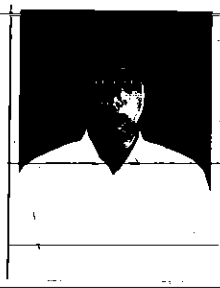
เอกสารอ้างอิง

[1] Donald Knuth. **The Art of Computer Programming, Volume 3: Sorting and Searching, Third Edition.** Addison-Wesley : 1997.

[2] Stuart J. Russell and Peter Norvig. **Artificial Intelligence A Modern Approach Second Edition.** United States of America : Pearson Education

[3] **"เกมเอแมท."** [Online]. Available : http://th.wikipedia.org/wiki/%E0%B9%80%E0%B8%81%E0%B8%A1%E0%B9%80%E0%B8%AD%E0%B9%81%E0%B8%A1%E0%B8%97

ประวัติผู้เขียนโครงงาน



ชื่อ นายธานี บัวเทศ

ภูมิลำเนา ตำบลในเมือง อำเภอเมือง จังหวัดพิษณุโลก 65000

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนเมืองพัทยา๘ จังหวัดชลบุรี
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : cpe_mon@hotmail.com

ชื่อ นางสาวกิตติยา ขุมทอง

ภูมิลำเนา 3/19 ถนนสามล้าน ตำบลพระสิงห์ อำเภอเมือง จังหวัดเชียงใหม่ 50200

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนวัฒโนทัยพายัพ จังหวัดเชียงใหม่
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : cpe_mimi@hotmail.com

ชื่อ นายธิติวุฒิ สำเภาเงิน

ภูมิลำเนา 4/34 หมู่2 ตำบลตะเคียนเตี้ย อำเภอบางละมุง
จังหวัดชลบุรี 20150

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนโพธิสัมพันธ์พิทยาคาร จังหวัดชลบุรี
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail : cpe_yok@hotmail.com