



การสร้างเกมเศรษฐี เกมบันไดงู และเกมลูได้ พร้อมกัน ด้วยวิธีสร้างสินทรัพย์หลัก
 CREATE MONOPOLY, SNAKES AND LADDERS AND LUDO
 SIMULTANEOUSLY BY CONSTRUCTING CORE ASSET

นายกิตติภูมิ ขุมเพชร

รหัส 52362465

นายกิตติศักดิ์ ยวงหิรัญ

รหัส 52362472

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาตรีวิศวกรรมศาสตรบัณฑิต
 สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
 คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555

ห้องสมุดคณะวิศวกรรมศาสตร์
 วันที่รับ..... 22/11/2555
 เลขทะเบียน..... 16826866
 เลขเรียกหนังสือ..... สค.
 มหาวิทยาลัยนเรศวร 167 4 11 2555



ใบรับรองปริญญาโท

ชื่อหัวข้อโครงการ การสร้างเกมเศรษฐี เกมบันไดงู และเกมลูโต้ พร้อมกัน
 ด้วยวิธีสร้างสินทรัพย์หลัก

ผู้ดำเนินโครงการ นายกิตติภูมิ ชุมเพชร รหัสสนិត 52362465
 นายกิตติศักดิ์ ขวงหิรัญ รหัสสนិត 52362472


ที่ปรึกษาโครงการ ดร. สุรเดช จิตประไพกุลศาล


สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์


ปีการศึกษา 2555

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธนเรศวร อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่งของ
การศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์


.....ที่ปรึกษาโครงการ
(ดร. สุรเดช จิตประไพกุลศาล)


.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ธิยะมงคล)


.....กรรมการ
(ดร.พงศ์พันธ์ กิจสนาโยธิน)


.....กรรมการ
(ดร.วรตัญญ์ ทงเค่นฟ้า)

ชื่อหัวข้อโครงการ การสร้างเกมเศรษฐี เกมบันไดงู และเกมลูโต้ พร้อมกัน
ด้วยวิธีสร้างสินทรัพย์หลัก

ผู้ดำเนินโครงการ นายกิตติภูมิ ขุมเพชร รหัสสนិត 52362465
นายกิตติศักดิ์ ขวงหิรัญ รหัสสนិត 52362472

ที่ปรึกษาโครงการ คร. สุรเดช จิตประไพกุลศาล
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2555

บทคัดย่อ

โครงการนี้เป็นการศึกษาเทคนิคสายผลิตภัณฑ์ซอฟต์แวร์ (Software Product Lines) ด้วยวิธีการสร้าง Core Assets จากเกม 3 เกม คือ เกมเศรษฐี (Monopoly) เกมบันไดงู (Snakes and Ladders) และเกมลูโต้ (Ludo) เพื่อลดแรงงานและต้นทุนในการสร้างเกม โดย Core Asset จะได้มาจากการวิเคราะห์จุดร่วมกันของทั้ง 3 เกม และออกแบบ Core Asset เพื่อให้สามารถใช้อย่างเต็มประสิทธิภาพซึ่งจะมีทั้งการออกแบบ User Interface การออกแบบตัวละครเกมให้สามารถใช้ได้ทุกเกมในสายการผลิต และสามารถนำแกนของเกมไปทำเป็น Library เพื่อให้ง่ายในการเรียกใช้ และลดปริมาณโค้ดที่ต้องเขียนเพิ่ม

Project title Create Monopoly, Snakes and Ladders and Ludo Simultaneously by Constructing Core Assets

Name Mr. Kittipoom Kumpet ID. 52362465

Mr. Kittisak Younghirun ID. 52362472

Project advisor Dr. Suradet Jitprapaikulsaarn

Major Computer Engineering

Department Electrical and Computer Engineering

Academic year 2012

Abstracts

This project is to study techniques, software product line method by constructing Core Assets of 3 games is Monopoly game, Snakes and Ladders game and Ludo game to reduce the labor and cost. Core Asset in the game are derived from the common analysis of 3 games and Core Asset is to be fully effective. It is well-designed User Interface design game characters to be available for all games on the production line. And can be made into the core of the game is so easy to run the Library. And reduce the amount of code that must be added.

กิตติกรรมประกาศ

โครงการวิศวกรรมคอมพิวเตอร์ฉบับนี้สำเร็จลุล่วงมาได้ นั้น เนื่องจากความอนุเคราะห์จากท่านอาจารย์ที่ปรึกษาโครงการ ดร.สุรเดช จิตประไพกุลศาล ที่กรุณาสละเวลาให้คำแนะนำในการทำงาน ตลอดจนการตรวจสอบการทำงาน พร้อมทั้งเสนอแนวทางการแก้ไขปัญหาตลอดระยะเวลาการทำงาน พร้อมทั้งให้คำแนะนำที่เป็นประโยชน์ทำให้การทำงานเป็นไปอย่างราบรื่น

ทั้งนี้ต้องขอขอบพระคุณกรรมการทั้งสามท่านอันได้แก่ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ธิยะมงคล ดร.วรลักษณ์ กองเค่งฟ้า และดร.พงศ์พันธ์ กิจสนาโยธิน อาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้า และคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ที่เสียสละเวลาอันมีค่าให้ปรึกษาและแนะแนวทางในการแก้ปัญหาต่างๆ

สุดท้ายนี้ผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา และอาจารย์ทุกท่าน ที่คอยสั่งสอนให้ความรู้จนผู้จัดทำสำเร็จการศึกษา และขอขอบคุณเพื่อน ๆ ที่คอยให้กำลังใจ ช่วยให้คำปรึกษาทั้งในเรื่องเรียน เรื่องส่วนตัวจนสำเร็จลุล่วงมาได้ด้วยดี

ขออำนาจคุณพระศรีรัตนตรัย และสิ่งศักดิ์สิทธิ์ทั้งหลายบันดาลให้บิดามารดาและอาจารย์ทุกท่านสุขภาพแข็งแรงและเป็นแรงผลักดันให้คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร มีความก้าวหน้าต่อไป

นายกิตติภูมิ บุมเพ็ชร
นายกิตติศักดิ์ ยวงหิรัญ

สารบัญ

	หน้า
ใบรับรองปริญญาโท ก	ก
บทคัดย่อภาษาไทย ข	ข
บทคัดย่อภาษาอังกฤษ ค	ค
กิตติกรรมประกาศ ง	ง
สารบัญ จ	จ
สารบัญตาราง ช	ช
สารบัญรูป ซ	ซ
บทที่ 1 บทนำ 1	1
1.1 ที่มาและความสำคัญของโครงการ 1	1
1.2 วัตถุประสงค์ 1	1
1.3 ขอบเขตของโครงการ 1	1
1.4 ขั้นตอนการดำเนินงาน 2	2
1.5 ระยะเวลาการดำเนินงาน 2	2
1.6 ผลที่คาดว่าจะได้รับ 3	3
1.7 งบประมาณ 3	3
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 4	4
2.1 สายผลิตภัณฑ์ซอฟต์แวร์ 4	4
2.2 Bill of Material..... 6	6
2.3 Dynamic Link Libraries 7	7
2.4 ภาษาซีชาร์ป 7	7
2.5 XNA Framework 8	8
2.6 เกมเศรษฐกิจ 9	9
2.7 เกมคู่ได้ 10	10
2.8 เกมบันไดงู 11	11

สารบัญ (ต่อ)

	หน้า
บทที่ 3 วิธีการดำเนินการ	13
3.1 โครงสร้างและขอบเขตของระบบ	13
3.2 เครื่องมือที่ใช้ในการพัฒนาระบบ	14
3.3 ความต้องการของระบบ	14
3.4 ขั้นตอนการทำงานของระบบ	15
3.4.1 การรวบรวมความต้องการของผู้ใช้	15
3.4.2 การหาจุดร่วมของทั้ง 3 เกม	17
3.4.3 การสร้าง Bill of Material	19
3.4.4 การออกแบบลำดับการทำงานของเกม	21
3.4.5 การออกแบบ User Interface และ Character Image	24
3.4.6 การเขียนโปรแกรม	25
บทที่ 4 ผลการทดลอง	27
4.1 วิธีการทดลอง	27
4.2 ผลการทดลอง	27
4.2.1 ผลการทดลองระดับได้ค	27
4.2.2 ผลการทดลองระดับการออกแบบ	30
บทที่ 5 สรุปผลการดำเนินงานและแนวทางการพัฒนา	37
5.1 สรุปผลการทดลอง	38
5.2 ปัญหาและแนวทางการแก้ไข	38
5.3 ข้อเสนอแนะ	39
เอกสารอ้างอิง	40
ภาคผนวก	41
ประวัติผู้ทำโครงการ	49

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน	2
3.1 แสดงความต้องการของผู้ใช้จากการสอบถาม	16
3.2 แสดงความต้องการของผู้ใช้จากการวิเคราะห์	17
3.3 แสดงจุดร่วมของทั้ง 3 เกม โดยแบ่งตามหมวดหมู่	18
4.1 แสดงตารางคลาสทั้งหมดใน Library เทียบกับจำนวน Line of Code	28
4.2 แสดงตารางปริมาณแรงงานทั้งหมดที่ลดลงเทียบกับแต่ละเกม	29
4.3 แสดงตารางเปรียบเทียบจุดร่วมในระดับต่างๆของแต่ละเกม	30
5.1 แสดงตารางสรุปผลการทดลอง	37
5.2 ปัญหาและแนวทางการแก้ไขในการดำเนินโครงการ	38

สารบัญรูป

รูปที่	หน้า
2.1 แสดงโครงสร้างการแบ่งส่วนประกอบของการผลิต	6
2.2 แสดงโครงสร้าง XNA Framework	9
2.3 แสดงกระดานดัชนีฉบับของเกมเศรษฐี	10
2.4 แสดงกระดานดัชนีฉบับของเกมลูโต้	11
2.5 แสดงตัวอย่างกระดานของเกมบันไดงู	12
3.1 แสดงขั้นตอนการทำงานของระบบ	15
3.2 แสดง Bill of Material ของ Core Assets	19
3.3 แสดงความสัมพันธ์ของโมดูลภายในแต่ละเกม	19
3.4 แสดง Bill of Material ทั้งหมด	20
3.5 แสดงเงื่อนไขการจบเกมของเกมเศรษฐี	21
3.6 แสดงเงื่อนไขการจบเกมของเกมบันไดงู	22
3.7 แสดงเงื่อนไขการจบเกมของเกมลูโต้	23
3.8 ตัวอย่างการออกแบบ User Interface หน้า Index	24
3.9 ตัวอย่างการออกแบบ User Interface หน้า Game Play	25
3.10 ตัวอย่างการออกแบบ Character Image	25
4.1 แสดงตัวอย่างการออกแบบ Use Interface ของ Index Page	31
4.2 แสดงตัวอย่างการออกแบบ Use Interface ของ Game Play Page	31
4.3 แสดงตัวอย่างการออกแบบ Use Interface ของ Player System Page	32
4.4 แสดงตัวอย่างการออกแบบ Character Image ในรูปแบบ Logo	33
4.5 แสดงตัวอย่างการออกแบบ Character Image ในรูปแบบการเลือกตัวละคร	33
4.6 แสดงตัวอย่างการออกแบบ Character Image ในรูปแบบที่สามารถเคลื่อนไหวได้	33
4.7 แสดงระบบ Sound Effect / Music Volume	34
4.8 แสดงระบบ Save / Load	34
4.9 แสดงระบบ Manual Guide	35
4.10 แสดงระบบ Player System	35
4.11 แสดงภาพกระดานของทั้ง 3 เกม	35

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ปัจจุบันเกมคอมพิวเตอร์เข้ามามีบทบาทต่อวงการเกมไทยอย่างมาก ทำให้เกิดนักพัฒนาเกมรุ่นใหม่ขึ้นมาเรื่อยๆ นักพัฒนาเกมแต่ละคนมีแนวทางการพัฒนาเกมเป็นของตัวเอง ทำให้เกมในปัจจุบันมีแนวทางการพัฒนาที่หลากหลาย แต่ไม่สามารถนำ Core Asset เก่ากลับมาใช้ใหม่ได้ แนวทางหนึ่งที่จะช่วยแก้ปัญหาที่คือ การใช้หลักการใช้ซ้ำ เข้ามาวิเคราะห์และออกแบบระบบแกนของเกม เพื่อสร้างเกมที่สามารถนำแกนของเกมนั้นๆ มาใช้กับเกมอื่นๆ ได้ เป็นการลดต้นทุนทั้งในด้านเวลา ด้านจำนวนเงินสำหรับการผลิตเกม ทั้งนี้เกมที่ได้จาก Core Asset นั้นสามารถที่จะออกแบบ เพื่อให้เป็นไปตามความนิยมของผู้เล่นและตลาดได้อีกด้วย

โดยเกมที่นำมาเป็นกรณีศึกษาคือ เกมเศรษฐี (Monopoly), เกมบันไดงู (Snakes and Ladders) และเกมลูโต้ (Ludo) ซึ่งทั้งสามเกมเป็นเกมที่มีรูปแบบ และกฎการเล่นที่เข้าใจได้ง่าย อีกทั้งยังมีจุดร่วมที่คล้ายคลึงกัน สามารถนำมาสร้าง Core Asset ร่วมกันได้

1.2 วัตถุประสงค์

1. ทดสอบสร้างเกมเศรษฐี เกมบันไดงู และเกมลูโต้ ด้วยการใช้วิธี Constructing Core Asset
2. เพื่อออกแบบ โมเดลจาก Core Asset นำไปใช้ในการพัฒนาเกมที่สามารถประยุกต์ต่อ กับเกมอื่นๆ ได้ โดยโมเดลที่ได้จะช่วยลดระยะเวลาในการสร้างเกมใหม่

1.3 ขอบเขตของโครงการ

1. Core Asset ของโครงการนี้มาจาก 3 เกม คือ เกมเศรษฐี (Monopoly), เกมบันไดงู (Snakes and Ladders) และเกมลูโต้ (Ludo)
2. แสดงผลได้เฉพาะบนวินโดวส์เท่านั้น

1.4 ขั้นตอนการดำเนินงาน

1. วางแผนการทำงาน รวบรวมความต้องการของผู้ใช้
2. ศึกษาค้นคว้าข้อมูล หา Core Asset ของเกม
3. ออกแบบการทำงานของเกม
4. พัฒนาเกม
5. ทดสอบการทำงานของเกม
6. แก้ไขข้อผิดพลาด และเก็บรายละเอียดส่วนต่างๆของโครงการ
7. สรุปผลการดำเนินโครงการและจัดทำรูปเล่มวิทยานิพนธ์

1.5 ระยะเวลาการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	ปี 2555							ปี 2556		
	มี.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. วางแผนการทำงาน รวบรวมความต้องการของผู้ใช้										
2. ศึกษาค้นคว้าข้อมูล หา Core Asset ของเกม										
3. ออกแบบการทำงานของเกม										
4. พัฒนาเกม										
5. ทดสอบการทำงานของเกม										
6. แก้ไขข้อผิดพลาดของโครงการ										

ตารางที่ 1.1 (ต่อ) แผนการดำเนินงาน

กิจกรรม	ปี 2555							ปี 2556		
	มี.ย.	ก.ก.	ส.ก.	ก.ย.	ต.ก.	พ.ย.	ธ.ก.	ม.ค.	ก.พ.	มี.ค.
7. สรุปผลการดำเนินโครงการ และจัดทำรูปเล่มวิทยานิพนธ์										

1.6 ผลที่คาดว่าจะได้รับ

1. สามารถนำ โมเดลที่ได้จาก Core Assets ของสามเกมนี้ ไปใช้กับการพัฒนาเกมอื่นๆ ได้
2. เกมที่ได้นั้นจะง่ายในการนำไปพัฒนาต่อยอด เพื่อให้มีประสิทธิภาพดีขึ้น
3. สามารถลดแรงงานและปริมาณ Code ในการสร้างเกมใหม่ได้
4. สามารถนำความรู้และทักษะเกี่ยวกับการพัฒนาเกมนี้ ไปประยุกต์กับธุรกิจที่สามารถสร้างรายได้ในรูปแบบต่างๆ ได้

1.7 งบประมาณ

1. ค่าใช้จ่ายในการทำเอกสารและการเข้าเล่ม	1000	บาท
2. หนังสือคู่มือเกี่ยวกับการศึกษาค้นคว้า	500	บาท
3. ค่าใช้จ่ายอื่นๆ	500	บาท
รวมทั้งสิ้น	<u>2,000</u>	บาท

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะนำเสนอเกี่ยวกับหลักการและทฤษฎีที่เกี่ยวข้อง พร้อมกับเทคโนโลยีที่ใช้ในการพัฒนาโครงการเช่น หลักการใช้ Core Assets ในการผลิตซอฟต์แวร์ ว่ามีแนวคิดในการจัดการซอฟต์แวร์เป็นอย่างไร และในส่วนของเทคนิคต่างๆที่จะนำมาใช้ในการสร้างซอฟต์แวร์ เพื่อให้การใช้ Reuse เกิดประสิทธิภาพสูงสุด นอกจากนี้จะอธิบายเกี่ยวกับประวัติ รวมถึงกติกาการเล่นของเกมที่น่าสนใจ เป็นกรณีศึกษาทั้งสามเกมอีกด้วย

2.1 สายผลิตภัณฑ์ซอฟต์แวร์ (Software Product Lines)

สายผลิตภัณฑ์ซอฟต์แวร์ คือ การนำแนวคิดการผลิตเชิงอุตสาหกรรมมาใช้ โดยนำหลักการที่สำคัญ คือ " การจัดการสายการผลิต " ซึ่งเริ่มต้นตั้งแต่การวิจัยและสำรวจตลาดวิจัย ออกแบบผลิตภัณฑ์ วางแผนสายการผลิต จนถึงเริ่มต้นผลิตสินค้า ประโยชน์ที่ได้รับจากการจัดการสายการผลิตที่ดีคือ ช่วยลดต้นทุนและเวลาในการผลิต สนับสนุนให้เกิดกระบวนการการผลิตสินค้าที่มีความหลากหลายและปรับให้เข้ากับผู้บริโภคที่มีความหลากหลายได้เป็นอย่างดี และช่วยเพิ่มคุณภาพให้กับตัวสินค้า นอกจากนี้ยังมีประโยชน์อื่นอีกมากมาย โดยอยู่บนพื้นฐานของการนำกลับมาใช้ใหม่ แต่การนำกลับมาใช้ใหม่ต้องมีการวางแผนและจัดการอย่างดี มิใช่เป็นเพียง ประเด็นด้านวิชาการเท่านั้น แต่เป็นประเด็นในระดับบริหาร

การผลิตซอฟต์แวร์หรือโปรแกรมประยุกต์หลายชนิดที่มีความเกี่ยวข้องกัน หรือเรียกว่ากลุ่มผลิตภัณฑ์ (Product Family) เหมาะสำหรับการผลิตซอฟต์แวร์เชิงพาณิชย์หรือการพัฒนาโปรแกรมประยุกต์ขึ้นมาใช้เองภายในองค์กร ซึ่งทั้งซอฟต์แวร์และโปรแกรมประยุกต์ที่สร้างขึ้นมามีได้มีเพียงตัวเดียว แต่มีหลายตัวและมักไม่ได้ถูกสร้างขึ้นมาร่วมกัน และอาจมีการสร้างขึ้นมาเพื่อจำหน่ายหรือใช้งานอยู่เรื่อยๆ หลายองค์กรจึงมักมีปัญหา เช่น

1. ต้องแก้ไขรหัสต้นฉบับเดิมๆ เพื่อให้เข้ากับงานของลูกค้ารายใหม่ จนระบบหนึ่งๆ มีหลายรุ่นตามจำนวนลูกค้า
2. เสียเวลาและกำลังคนในการนำไปใช้กับระบบๆ ให้กับลูกค้าแต่ละราย เมื่อจะพัฒนาระบบๆ ใหม่ ก็ต้องเก็บความต้องการของผู้ใช้มาวิเคราะห์ ออกแบบ และสร้างใหม่หมดตั้งแต่ต้น โดยไม่

สามารถนำสิ่งที่มีอยู่แล้วกลับมาใช้ใหม่ได้

3. ไม่ทราบว่าจะอะไรสามารถนำกลับมาใช้ใหม่ได้ และนำกลับมาใช้ใหม่ไม่ได้ และจะทำให้อะไรให้นำกลับมาใช้ใหม่ได้บ้าง เลยมักนำกลับมาใช้ใหม่กันผิดๆ ถูกๆ และเสียเวลาไปกับการออกแบบเพื่อการนำกลับมาใช้ใหม่โดยเปล่าประโยชน์

4. ไม่เคยออกแบบสถาปัตยกรรมซอฟต์แวร์มาก่อน หรือไม่เคยออกแบบล่วงหน้าเลย หรือจะทำก็ต่อเมื่อมีการเริ่มพัฒนาระบบฯ ใหม่เท่านั้น

5. องค์กรคิดว่าระบบฯ ที่พัฒนาขึ้นแต่ละครั้งเป็นระบบฯ เล็กๆ จึงไม่เคยใส่ใจในการสร้างสถาปัตยกรรมซอฟต์แวร์ที่ระบบฯ ต่างๆ จะสามารถใช้ร่วมกันได้เพื่อประหยัดเวลาและต้นทุนในการพัฒนา

6. การมีช่องว่างในการทำงานที่สูงเกินไประหว่างฝ่ายบริหาร ฝ่ายการตลาด ฝ่ายผลิต (ฝ่ายพัฒนาซอฟต์แวร์) และฝ่ายอื่นๆ ทำให้ประสานงานและสร้างแผนงานร่วมกันได้ไม่ดีพอ

7. ไม่เคยคำนึงถึงความสำคัญของการจดสิทธิบัตรจากผลงานการออกแบบและสร้างสถาปัตยกรรมซอฟต์แวร์ที่สามารถถูกใช้ร่วมกันในระบบฯ ต่างๆ ได้

8. มองไม่ออกว่าส่วนประกอบใดมีประโยชน์ ควรค่าแก่การจัดเก็บและดูแลโดยระบบจัดการทรัพย์สิน

9. องค์กร ไม่มีเครื่องมือในการพัฒนาและภาษาในการเขียนโปรแกรมที่เป็นของตนเองหรือเป็นกลาง เพื่อใช้เป็นเครื่องมือกลาง ในการปรับแต่งระบบฯ ของตนเอง ทำให้ในแก้ไขก็ต้องลงไปแก้ถึงระบบรหัสต้นฉบับแทบทุกครั้ง

11. ขาดการคาดการณ์และวางแผนสำหรับอนาคตข้างหน้าอย่างชัดเจน และวางแผนจัดการร่วมกันทุกฝ่าย

12. การให้ความสำคัญและสนใจด้านการนำกลับมาใช้ใหม่โดยมีทัศนคติด้านวิชาการเป็นหลัก และมีมากเกินไป

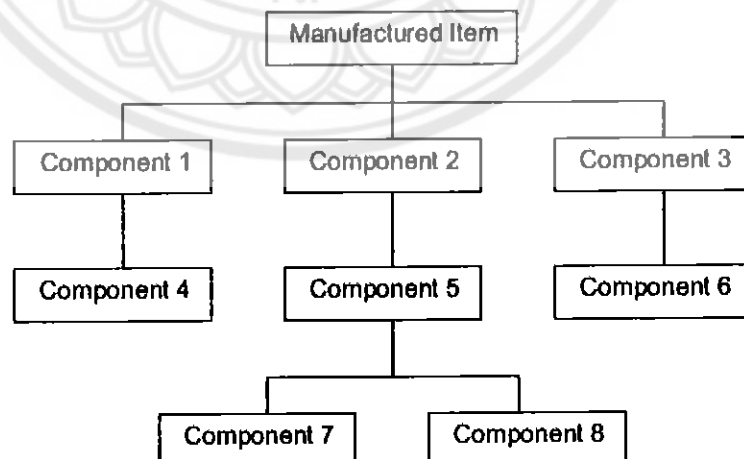
ดังนั้นการนำหลักการการจัดการสายการผลิตกลับมาประยุกต์กับการงานซอฟต์แวร์ จึงอยู่บนพื้นฐานของการตัดสินใจทั้งด้านวิชาการ และด้านธุรกิจ หรือ องค์กรประกอบด้วยกัน โดยมีการบริหารการตลาด และมีกรสร้างและจัดการสถาปัตยกรรมซอฟต์แวร์ที่ดีพอ จึงจะช่วยให้องค์กรประสบความสำเร็จ โดยเฉพาะการลดต้นทุน เวลา และเพิ่มคุณภาพให้ยิ่งๆ ขึ้นไป

2.2 Bill of Material (BOM)

Bill of Material คือ สูตรหรือโครงสร้างสินค้าที่จำเป็นต้องใช้ในการผลิต เป็นสิ่งที่สำคัญอย่างหนึ่งในกระบวนการผลิต โดย BOM จะแสดงข้อมูลดังต่อไปนี้ ส่วนประกอบ, จำนวนส่วนประกอบ, รายการสิ่งที่ผลิตขึ้นจากส่วนประกอบ, รายการวัตถุดิบ, รายการข้างต้นจะเป็นความต้องการต่อสินค้าหนึ่งหน่วย หนึ่งในข้อมูลที่สำคัญของการผลิตคือ สูตรการผลิต (Bill of Materials) การที่จะคิดตามส่วนประกอบของข้อมูลมากมาย เพื่อให้ทราบการเปลี่ยนแปลงของการส่งมอบความต้องการใช้ และกำลังการผลิต หากต้องการที่จะรู้ว่าชิ้นส่วนใดบ้างที่ต้องการเมื่อมีการเริ่มผลิต สิ่งเหล่านี้จะเกิดขึ้นได้ต้องมีข้อมูลของ BOM

ในการผลิตชิ้นส่วนอุปกรณ์ หรือเครื่องจักรต่างๆ จะต้องมีขั้นตอนในการคำนวณ ออกแบบ และการจัดทำแบบรายละเอียดของชิ้นงานที่ต้องการ เพื่อส่งต่อให้หน่วยงานผลิตทำการผลิตชิ้นงานตามแบบที่ต้องการ ถ้าหากผู้ที่ออกแบบและจัดทำแบบ ไม่ได้กำหนดรายละเอียดของชิ้นงานที่ดีพอ ก็อาจจะทำให้ผู้ผลิตชิ้นงานไม่สามารถผลิตชิ้นงานได้ถูกต้องตรงตามที่ต้องการ

ฉะนั้นในการออกแบบและจัดทำแบบ จึงควรทำตารางรายการวัสดุที่ระบุรายละเอียดต่างๆ ของชิ้นงาน เช่น ชื่อของชิ้นงานย่อยที่ประกอบกันเป็นชิ้นงานใหญ่ วัสดุที่ใช้ทำ จำนวนชิ้นงานที่ต้องการ นำหนักโดยประมาณของชิ้นงานใส่ลงในแบบด้วย เพื่อให้ผู้ผลิตสามารถจัดหาวัสดุได้ตรงตามชนิดและจำนวนที่ต้องใช้ ซึ่งจะช่วยให้การผลิตชิ้นงานนั้นเป็นไปด้วยความรวดเร็ว ถูกต้องมากขึ้นและลดปริมาณวัสดุที่จะต้องสูญเสียในกระบวนการผลิตลง



รูปที่ 2.1 แสดงโครงสร้างการแบ่งส่วนประกอบของการผลิต

2.3 Dynamic Link Libraries (DLL)

Dynamic Link Libraries คือ คุณลักษณะเด่นของระบบปฏิบัติการวินโดวส์ ภายในไฟล์ DLL จะประกอบด้วยฟังก์ชันที่โปรแกรมสามารถเรียกใช้งานระหว่าง Execution ได้ ในอีกความหมายหนึ่ง DLL คือ Library ของฟังก์ชันที่โปรแกรมสามารถเรียกใช้งานแบบพลวัต

ข้อดีของ DLL คือ

- ใช้ทรัพยากรน้อยลง คือ เมื่อโปรแกรมหลายๆโปรแกรมเรียกใช้ฟังก์ชันจาก Library เดียวกัน DLL สามารถลดความซ้ำซ้อนของ Code ที่จะถูกโหลดบนดิสก์และหน่วยความจำ ทำให้เป็นการเพิ่มประสิทธิภาพของโปรแกรมที่กำลังทำงานอยู่
- ช่วยส่งเสริมการพัฒนาโปรแกรมแบบย่อย ซึ่งจะช่วยให้การพัฒนาโปรแกรมขนาดใหญ่สามารถทำได้โดยง่าย ยกตัวอย่างเช่น โปรแกรมที่มีการใช้โปรแกรมย่อย (DLL) จะสามารถโหลดโปรแกรมย่อยนั้นขณะโปรแกรมทำงานอยู่ได้เลย
- ลดการสิ้นเปลืองทรัพยากรและการติดตั้ง เมื่อฟังก์ชันภายใน DLL ต้องมีการปรับปรุงหรือแก้ไข การปรับใช้และการติดตั้ง DLL ใหม่จะไม่เกิดปัญหาการเชื่อมโยงกับ DLL ส่วนอื่นๆ นอกจากนี้ถ้าหลายๆโปรแกรมเรียกใช้ DLL เดียวกัน ก็จะได้รับประโยชน์จากการแก้ไขเพียงครั้งเดียว

2.4 ภาษาซีชาร์ป

C# คือ ภาษาคอมพิวเตอร์ประเภท Object-Oriented Programming พัฒนาโดย Microsoft โดยมีจุดมุ่งหมายในการรวมความสามารถการคำนวณของ C++ ด้วยการโปรแกรมง่ายกว่าของ Visual Basic โดย C# มีพื้นฐานจาก C++ และส่วนการทำงานคล้ายกับ Java

C# ได้รับการออกแบบให้ทำงานกับ .NET platform ของ Microsoft จุดมุ่งหมายคือ อำนวยความสะดวกในการแลกเปลี่ยนสารสนเทศและบริการผ่านเว็บ และทำให้ผู้พัฒนาสร้างโปรแกรมประยุกต์ในขนาดกะทัดรัด C# ทำให้โปรแกรมง่ายขึ้นผ่านการใช้ Extensible Markup Language (XML) และ Simple Object Access Protocol (SOAP) ซึ่งยอมให้เข้าถึง Object ของโปรแกรมหรือเมธอด โดยปราศจากความต้อการให้ผู้เขียนโปรแกรมเขียนคำสั่งเพิ่มเติมในแต่ละขั้นตอน เนื่องจากผู้เขียนโปรแกรมสามารถสร้างบนคำสั่งที่มีอยู่แทนที่การคัดลอกซ้ำ C# ภาษา C# ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi ,

ภาษา C++ มาปรับปรุงเพื่อให้มีความเป็น OOP (โปรแกรมเชิงวัตถุ) มากขึ้น ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง (เรียบง่ายกว่าภาษา C++) และมีสิ่งที่เกินความจำเป็นน้อยลง (เมื่อเทียบกับ Java)

จุดเด่นของภาษา C#

- Component Oriented – เป็นภาษาที่เน้นชิ้นส่วน โดยถูกออกแบบมาเป็นอย่างดีทำให้สามารถนำมาใช้ต่อกันเป็นอะไรก็ได้

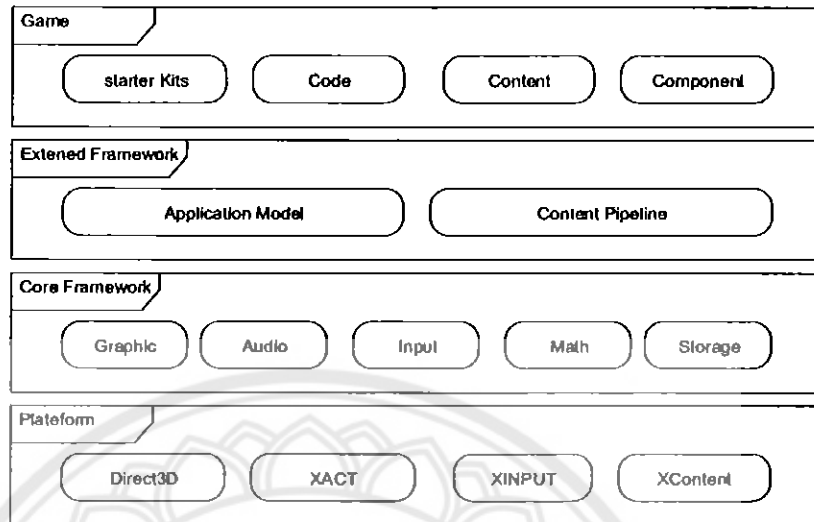
- สิ่งต่าง ๆ ใน C# เป็นออบเจกต์ทั้งหมด

- เป็นภาษา ที่ทนทาน (Robust) - ทนต่อความผิดพลาด ไม่ทำให้ระบบรวนหรือระบบทำงานช้า เพราะ C# มีข้อดีคือ Garbage Collection, Exception, Type-safety และ Versioning

- ภาษา C# จัดเตรียมกลไกไว้หลายอย่างที่จะช่วยให้ผู้เขียนโปรแกรมสามารถนำโค้ดที่เขียนไว้ใน Project หนึ่งไปใช้กับอีก Project หนึ่งได้ง่าย นอกจากนั้นภาษา C# ยังสามารถเรียกใช้คลาสหลายพันคลาสใน .NET Framework ได้โดยตรง ทำให้ลดเวลาการพัฒนาซอฟต์แวร์ได้มาก

2.5 XNA Framework

XNA เป็น Framework ที่อยู่ในกลุ่มของ .NET Library ใช้ในการเขียนเกมบน Windows OS และ XBOX โดยชื่อของเครื่องมือนี้คือ XNA Game Studio Express ทำงานร่วมกับ Microsoft Visual Studio C# เวอร์ชันล่าสุดคือ XNA Game Studio 4.0 เป็นที่นิยมสำหรับการเขียนเกมในปัจจุบัน Application Model ของ XNA เป็นส่วนที่ทำให้นักพัฒนาไม่ต้องกังวลถึง platform ที่กำลังเล่นอยู่ ส่วนนี้จะทำหน้าที่จัดการเรื่อง Window, Message Pump, Timer or Clock, รวมทั้งการจัดการ Graphics Component และ Graphics Device นักพัฒนาจึงไม่ต้องสนใจความแตกต่างของ PC และ XBOX ยกเว้นในเกมที่ต้องการคุณสมบัติเฉพาะของเครื่องเท่านั้น และสามารถประกอบเกมขึ้นจาก Game Components ได้ด้วย (เรียกส่วนนี้ว่า Component Model) ซึ่งทำให้การสร้างเกมง่ายขึ้น



รูปที่ 2.2 แสดงโครงสร้าง XNA Framework

2.6 เกมเศรษฐี (Monopoly)

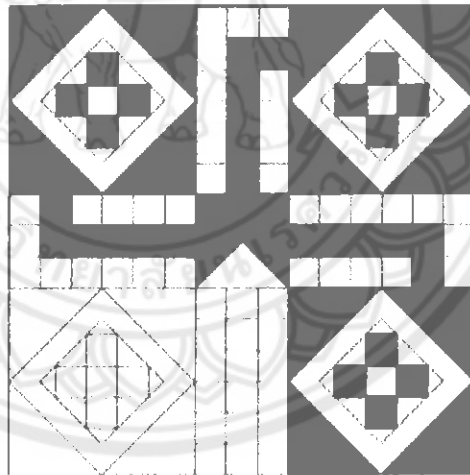
เกมเศรษฐี เป็นเกมกระดาน คิดค้นเป็นครั้งแรกโดย Elizabeth J. Magie จดลิขสิทธิ์ไว้เมื่อ ค.ศ. 1904 ต่อมาในปีค.ศ. 1935 ที่ห้องพาร์กเกอร์ (Parker Brothers) ได้ปรับปรุงให้รูปแบบการเล่นง่ายขึ้น ดังเช่นในปัจจุบัน และถือลิขสิทธิ์ โดย Hasbro บริษัทผลิตเกมจากสหรัฐอเมริกา

แผนที่ในเกมต้นฉบับ มีอยู่ 40 ช่อง โดยเป็นชื่อสถานที่จริงในเมือง แอตแลนติกซิตี ในรัฐ นิวเจอร์ซีย์ ตัวละครที่เป็นสัญลักษณ์ของเกม มีชื่อเฉพาะว่า Mr. Monopoly

กติกาการเล่นเกมเศรษฐี

1. เตรียมผู้เล่นอย่างน้อย 2 คน
2. ธนาคารจะแจกเงินเริ่มต้นจำนวนหนึ่งแก่ผู้เล่นทุกคน
3. เมื่อเริ่มเล่น ผู้เล่นสามารถซื้อ จำนวนที่ดินได้
4. เมื่อผู้เล่นเดินทางไปยังสถานที่ที่เป็นของผู้เล่นคนอื่น ผู้เล่นคนนั้นต้องจ่ายค่าเช่าให้กับผู้เล่นที่เป็นเจ้าของสถานที่นั้นๆ ตามที่ระบุไว้ใน โฉนด
5. ผู้เล่นคนใด ที่เดินทางไปยังช่อง โอกาสเศรษฐีหรือที่บสมบัตติ ผู้เล่นคนนั้นต้องเปิดบัตรแล้วทำตามคำสั่งนั้น
6. ผู้เล่นคนใดเดินทางผ่านจุดเริ่มต้น ผู้เล่นคนนั้นจะได้รับเงินจำนวนหนึ่ง

3. ผู้เล่นคนแรกจะเริ่มทอยลูกเต๋า จนกว่าจะได้หน้า 6 โดยมีโอกาส 3 ครั้ง ถ้าไม่ได้อีกก็จะเปลี่ยนผู้เล่นไปเรื่อยๆ
4. หากผู้เล่นทอยได้หน้า 6 จะสามารถดึงเบี้ยจากฐานเข้าสู่เกมได้ 1 ตัว โดยเบี้ยจะไปเกิดที่ตรงตำแหน่งใกล้ๆฐาน หลังจากนั้นผู้เล่นสามารถทอยลูกเต๋าดูได้อีกหนึ่งครั้ง
5. เมื่อผู้เล่นมีเบี้ยอยู่ในเกมแล้ว เมื่อทอยได้ก็แต้มสามารถนำเบี้ยในเกมเดินไปตามแต้มที่ได้
6. หากผู้เล่นมีเบี้ยอยู่ในเกม แต่ทอยลูกเต๋าดูได้หน้า 6 ผู้เล่นก็สามารถเลือกที่จะดึงเบี้ยออกจากฐานเพิ่มอีกหนึ่งตัว หรือเลือกที่จะใช้เบี้ยตัวเดิมเดิน 6 ช่องก็ได้
7. หากโดนเบี้ยของผู้เล่นอื่นเดินตกมายังตำแหน่งเดียวกันกับเบี้ยของเรา เบี้ยของเราจะถูกส่งกลับไปที่ฐาน
8. เมื่อเบี้ยเดินครบรอบจะเข้าไปยังเส้นชัยภายในสี่ของตน หากเบี้ยของผู้เล่นคนใดเข้าเส้นชัยครบ 4 ตัวก่อนก็จะเป็นผู้ชนะ



รูปที่ 2.4 แสดงกระดานต้นฉบับของเกมลูโด

ที่มา: [http:// http://en.wikipedia.org/wiki/Ludo_\(board_game\)](http://en.wikipedia.org/wiki/Ludo_(board_game))

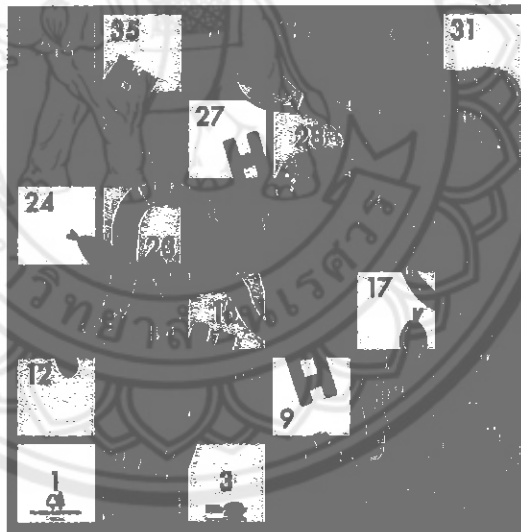
2.8 เกมบันไดงู (Snakes and Ladders)

เกมบันไดงูเป็นเกมกระดานที่เป็นที่นิยมมาอย่างยาวนาน ใช้ผู้เล่นตั้งแต่ 2 คนขึ้นไป บนกระดานสี่เหลี่ยมจัตุรัสของเกมบันไดงูจะมีตารางสี่เหลี่ยมเล็กๆ ซึ่งมีตัวเลขกำกับอยู่ซึ่งแต่ละตารางของเกมบันไดงูจะแตกต่างกันไป ปกติขนาดของช่องจะเป็น 8x8, 10x10 หรือ 12x12 ในบางช่องจะมี

"บันได" พาดเชื่อมกันระหว่างสองช่อง และมี "งู" เชื่อมระหว่างสองช่องเช่นกัน โดยบันไดและงูจำนวนหนึ่งจะพาดผ่านแทบทั่วทั้งกระดานอย่างไม่มีกฎเกณฑ์ตายตัว ซึ่งจะมีผลต่อการเดินไปตามช่องระหว่างเล่น

กติกาการเล่นเกมนันไดงู

1. เตรียมผู้เล่นอย่างน้อย 2 คน
2. ทอยลูกเต๋าแล้วเดินไปตามช่องตามตาราง
3. หากตกช่องบันไดก็ให้ขึ้นไปสู่คบันได
4. หากตกช่องที่มีปากงู ให้ลงจากช่องปากงู ไปยังช่องหางงู
5. หากตกช่องที่มีข้อความ ให้ทำตามข้อความนั้น
6. ผลัดกันเล่น ไปเรื่อยๆ จนกว่าจะถึงเส้นชัย หากคนไหนเข้าเส้นชัยได้ก่อนคนนั้นชนะ



รูปที่ 2.5 แสดงตัวอย่างกระดานของเกมนันไดงู

ที่มา: <http://sarahmillin.wordpress.com/2011/09/20/forestry-commission-brief-finished-the-final-snakes-and-ladders-game>

บทที่ 3

วิธีดำเนินการ

ในบทนี้จะนำเสนอเกี่ยวกับวิธีดำเนินการสำหรับการสร้างทั้ง 3 เกม ได้แก่ เกมเศรษฐี เกมบันไดงู และเกมลูโก้ ซึ่งในการพัฒนาจะอาศัยเทคนิคและเทคโนโลยีต่างๆที่ได้กล่าวไปในบทที่ 2 มาประยุกต์ใช้ โดยการดำเนินงานของโครงการจะมีหลายส่วนดังต่อไปนี้

- โครงสร้างและขอบเขตของระบบ
- เครื่องมือที่ใช้ในการพัฒนาระบบ
- ความต้องการของระบบ
- ขั้นตอนการทำงานของระบบ

3.1 โครงสร้างและขอบเขตของระบบ

ระบบการทำงานของโครงการนี้คือการออกแบบจตุรร่วมกันของทั้ง 3 เกม ได้แก่ เกมเศรษฐี เกมบันไดงู และเกมลูโก้ เพื่อลดประมาณ Code และแรงงานในการสร้างเกม เป็นการ Reuse ตามหลักการของ Software Product Line ทั้งนี้เกมจะต้องสามารถควบคุมได้ในแง่ของนักพัฒนา เพื่อให้สามารถนำสายการผลิตเก่ามาใช้ใหม่ได้ และเกิดประโยชน์สูงสุด ในแง่ของธุรกิจเพื่อสร้างรายได้ โดยเกมที่จะนำมาพัฒนาจะต้องมีรูปแบบคล้ายกัน เพื่อให้สามารถออกแบบจตุรร่วมกันได้ นอกจากนี้เครื่องมือที่นำมาพัฒนาเกมจะต้องมีความสามารถเพียงพอต่อการพัฒนาเกมในปัจจุบันที่เกมจะต้องมีความสลับไหล Graphic ของเกมจะต้องสามารถทำให้ผู้เล่นชื่นชอบได้ มีระบบต่างๆที่พร้อมรองรับผู้เล่นและนักพัฒนาอย่างเต็มที่ นอกจากนี้ต้องเข้ากันได้กับ Platform ที่คนส่วนใหญ่เข้าถึงเช่น ระบบปฏิบัติการ Windows ของบริษัท Microsoft

3.2 เครื่องมือที่ใช้ในการพัฒนาระบบ

3.2.1 ซอร์ฟแวร์และภาษาที่ใช้ในการพัฒนาระบบ

1. Microsoft Visual Studio 2010 Express
2. XNA Game Studio 4.0 Express
3. Photoshop CS
4. UMLet (Diagram Tool)
5. iPlotz (Wireframe Design Tool)

3.2.2 ฮาร์ดแวร์

เครื่องคอมพิวเตอร์ 1 เครื่องที่ใช้ในการพัฒนาระบบ

- CPU Intel Quad Core Q8200 2.33 GHz
- 4 GB DDR2 Ram
- 64-bit Windows 7
- DirectX 9.0
- NVIDIA GeForce GTS 250

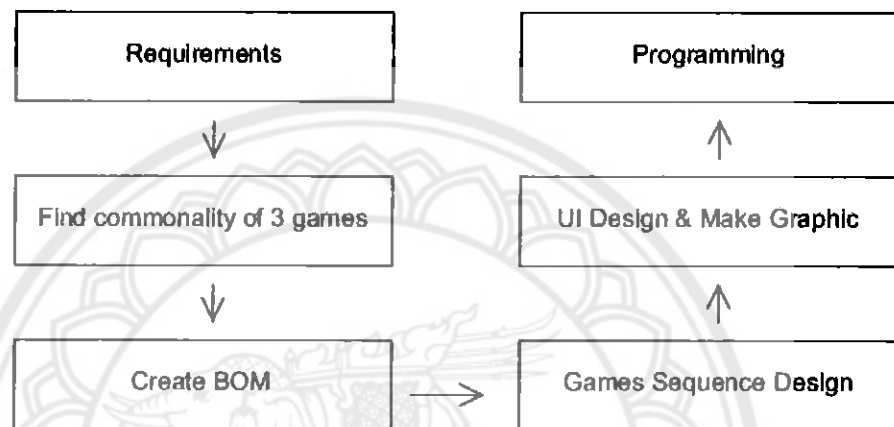
3.3 ความต้องการของระบบ

ความต้องการขั้นต่ำของระบบ

- Windows XP SP2 หรือ Windows Vista ขึ้นไป
- DirectX 9
- .NET Framework 4.0
- 512 MB Ram (Recommend 1-2 GB)
- 1 GHz CPU or Faster
- Shader Model 1.1 graphics card (GeForce 3 หรือ ATI 7000)

3.4 ขั้นตอนการทำงานของระบบ

แบ่งขั้นตอนการทำงานของระบบออกเป็น 6 ขั้นตอน โดยในแต่ละขั้นตอนล้วนมีความสำคัญในการพัฒนาสายการผลิต เพื่อให้เกมมีประสิทธิภาพและสามารถนำส่วนต่างๆ กลับมาใช้ใหม่ได้สูงสุด



รูปที่ 3.1 แสดงขั้นตอนการทำงานของระบบ

3.4.1 การรวบรวมความต้องการของผู้ใช้

สำหรับขั้นตอนแรกของการทำงานของระบบคือ การเก็บความต้องการของผู้ใช้ ซึ่งเป็นส่วนสำคัญต่อโครงการ เพราะขั้นตอนนี้ทำให้ทราบถึงสิ่งที่ผู้ใช้ต้องการ และสำหรับในแง่ธุรกิจ สิ่งนี้ทำให้ผู้ผลิตสามารถตอบสนองผู้บริโภคได้อย่างตรงเป้าหมาย

ในการเก็บความต้องการของผู้ใช้ในโครงการนี้ จะใช้การสอบถามจากผู้ใช้ที่มักเล่นเกมเป็นประจำ และการสืบค้นจากแนวโน้มของเกมที่เป็นที่นิยม ว่าเกมทั้งสาม ได้แก่ เกมเศรษฐี เกมบันไดงู และเกมลูกได้ ควรจะมีระบบใดบ้างที่จะทำให้เกมดูน่าสนใจ

ตารางที่ 3.1 แสดงความต้องการของผู้ใช้จากการสอบถาม

ความต้องการของผู้ใช้จากการสอบถาม	การคัดเลือก	หมายเหตุ
ระบบลงชื่อเข้าใช้	ไม่เลือก	เนื่องจากเกมทั้งสามเป็นเกมที่เล่นจบภายในเวลาอันสั้น ทำให้ไม่จำเป็นต้องมีระบบนี้
ระบบเพิ่มความเสี่ยงในเกม	เลือก	
ระบบอธิบายวิธีการเล่น	เลือก	
ระบบเซฟและโหลดเกม	เลือก	
ระบบเก็บคะแนน	ไม่เลือก	เนื่องจากไม่ได้ทำระบบลงชื่อเข้าใช้ การเก็บคะแนนจึงไม่จำเป็น
ระบบซื้อ ไอเทมพิเศษ	ไม่เลือก	เกมทั้งสามไม่ได้มีการเพิ่มค่าพิเศษ ทำให้ไม่จำเป็นต้องมีไอเทมพิเศษเข้ามาในการช่วยเล่น
ระบบเก็บประวัติสถานะของเกม	เลือก	
ระบบปรับระดับความยากของ AI	ไม่เลือก	การปรับระดับของ AI ในเกมประเภทนี้ มักไม่ค่อยเห็นความแตกต่าง
ระบบปฏิสัมพันธ์กับผู้ใช้ผ่าน Social Media	เลือก	
ระบบพูดคุยกับผู้เล่นอื่น	ไม่เลือก	เนื่องจากไม่ได้ทำระบบเล่นหลายบุคคล จึงไม่เลือกทำระบบนี้

ตารางที่ 3.2 แสดงความต้องการของผู้ใช้จากการวิเคราะห์

ความต้องการของผู้ใช้จากการวิเคราะห์	หมายเหตุ
ระบบแสดงเครื่องหมายการค้า	ระบบนี้เป็นส่วนสำคัญด้านธุรกิจ เพราะทำให้ Brand ของผู้ผลิตเป็นที่รู้จักมากขึ้น
ลักษณะพิเศษของตัวละคร	การสร้างจุดเด่นในกับตัวละครในเกม เป็นส่วนช่วยให้ผู้ใช้จดจำ Brand ของผู้ผลิต ได้อีกหนึ่งทาง
ระบบ Player System (Lobby)	เนื่องจากเกมมีการมากกว่าหนึ่งคน ทำให้ต้องมีระบบจัดการ Player ทั้งการตั้งชื่อให้กับ Player การตั้งให้ Player เป็นผู้เล่นหรือเป็น AI
ระบบแก้ไขกฎการเล่น	เนื่องจากเกมเศรษฐกิจเป็นเกมที่มีรายละเอียดค่อนข้างมาก ดังนั้นจึงควรสามารถปรับแก้ไขกฎการเล่นได้บางส่วน ให้เหมาะสมกับสิ่งที่ผู้เล่นต้องการ

3.4.2 การหาจุดร่วมของทั้ง 3 เกม

ขั้นตอนนี้คือการหาจุดร่วมของส่วนต่างๆ ในเกม เพื่อนำสิ่งที่เหมือนกันของทั้งสามเกมไปทำเป็น Library ซึ่งถือเป็นหัวใจสำคัญของโครงการ คือ หากสามารถหาจุดร่วมกันของเกมได้มาก ก็สามรถลดปริมาณ Code ที่ต้องเขียนซ้ำได้มากขึ้น ในการทำ Library ด้วย DLL สามารถแบ่ง Namespace ย่อยๆ ออกมาได้ เพื่อให้มองเห็นส่วนต่างๆ เป็นหมวดหมู่ และง่ายในการจัดการ หรือการเพิ่มจุดร่วมเข้าไป

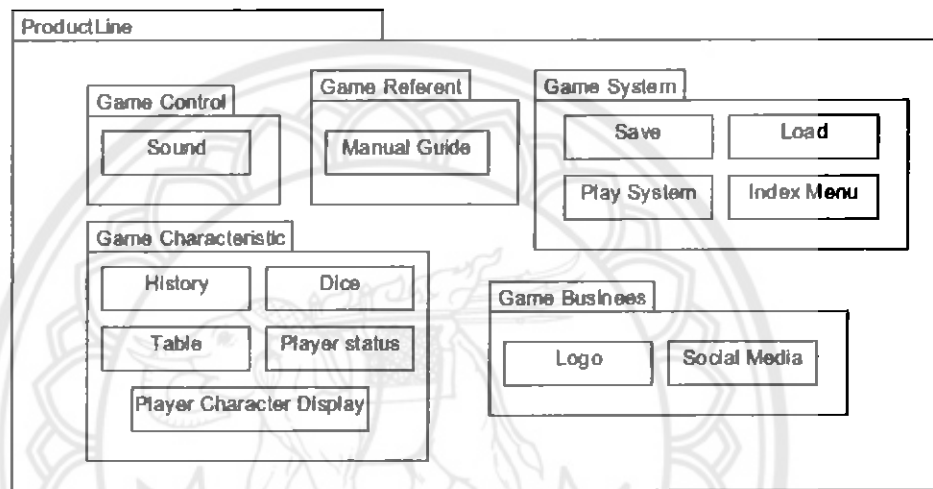
ตารางที่ 3.3 แสดงจุดร่วมของทั้ง 3 เกม โดยแบ่งตามหมวดหมู่

แบ่งตามหมวดหมู่	ระบบ
Game Control	<ul style="list-style-type: none"> ● Sound
Game Referent	<ul style="list-style-type: none"> ● Manual Guide
Game Characteristic	<ul style="list-style-type: none"> ● Dice ● Table ● Player Character Display ● History ● Player Status
Game System	<ul style="list-style-type: none"> ● Save ● Load ● Player System ● Index Menu
Game Business	<ul style="list-style-type: none"> ● Logo ● Social Media ● Character Image* ● Identity Design**

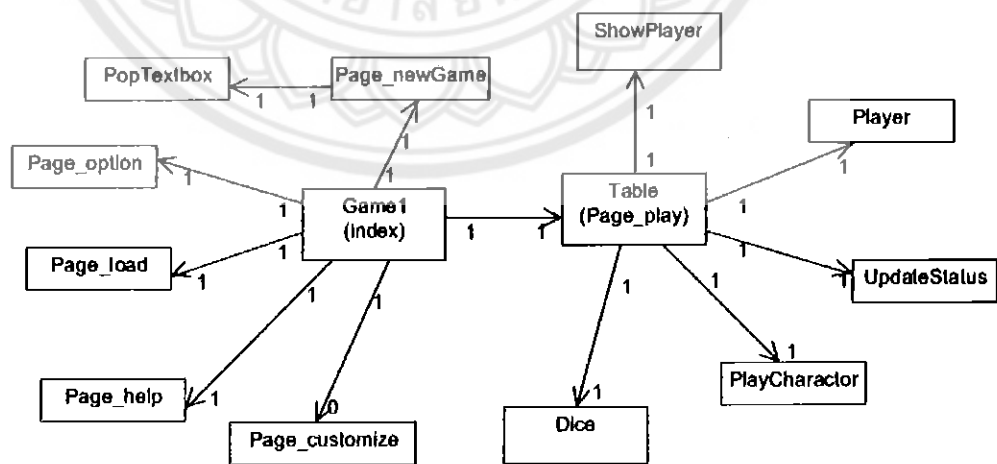
จากตารางที่ 3.3 แสดงให้เห็นว่าสามารถแบ่งจุดร่วมออกได้เป็น 5 หมวดหมู่ และ 15 ระบบ ซึ่งสามารถนำระบบเหล่านั้นไปทำเป็น Library ได้ ยกเว้นระบบ Character Image* และ Identity Design** ในหมวดหมู่ของ Game Business ที่ไม่สามารถนำไปทำ Library เนื่องจากทั้งสองระบบที่กล่าวมาเป็นการ Reuse ในระดับการออกแบบ

3.4.3 การสร้าง Bill of Material

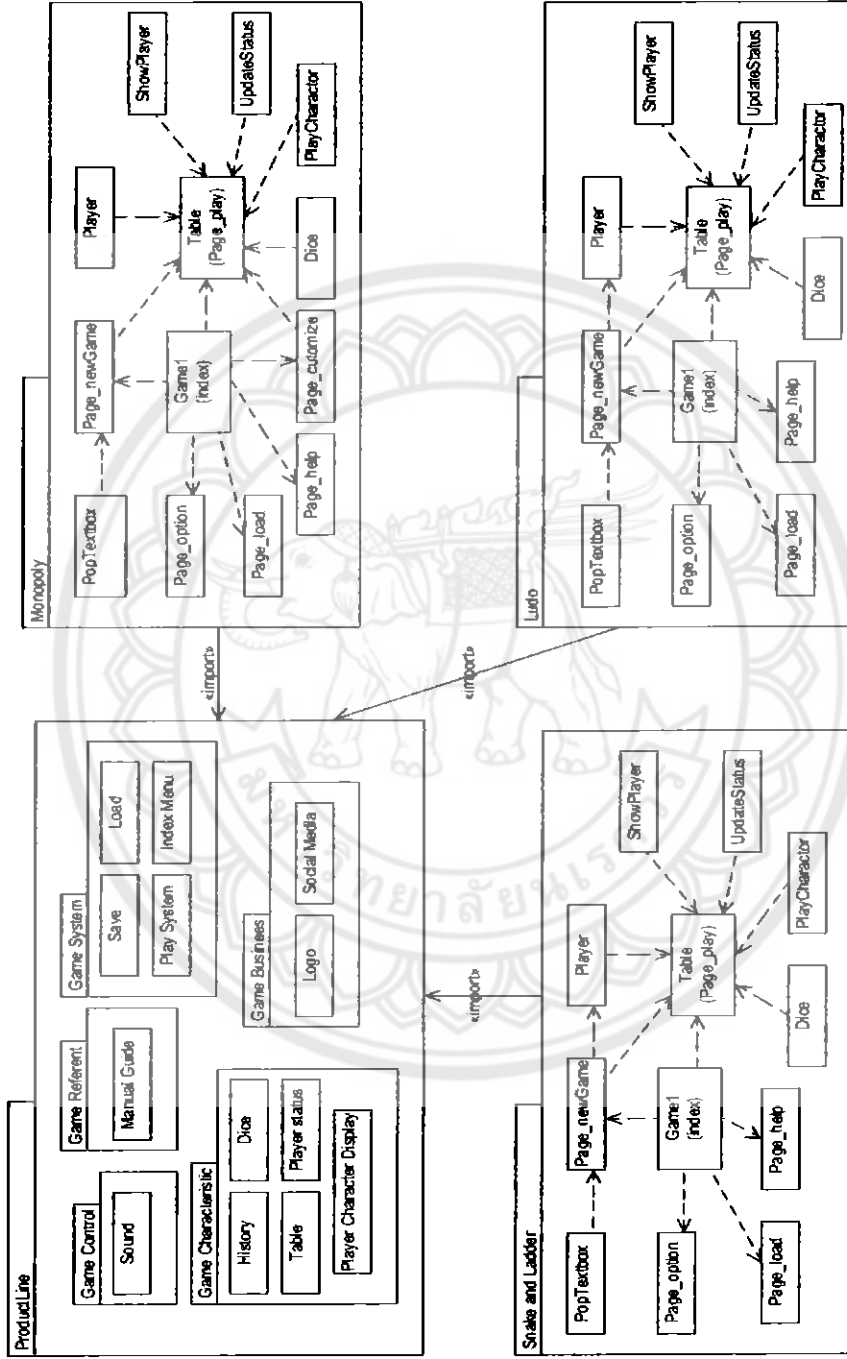
ก่อนการสร้างเกมส์ที่สำคัญต่อการผลิตอีกอย่างหนึ่งก็คือ การเขียนโครงสร้างทั้งหมดที่จำเป็นต้องใช้ว่าในเกมส์จะมีส่วนประกอบอะไรบ้าง เพื่อให้ทราบถึงสิ่งที่จะเขียนโปรแกรม นั้นจะทำให้สามารถเก็บรายละเอียดของทุกฟังก์ชันได้ครบ ถือว่าเป็นการเตรียมความพร้อมก่อนการเริ่มเขียนโปรแกรม เรียกขั้นตอนนี้ว่าการสร้าง Bill of Material



รูปที่ 3.2 แสดง Bill of Material ของ Core Assets



รูปที่ 3.3 แสดงความสัมพันธ์ของโมดูลภายในแต่ละเกมส์



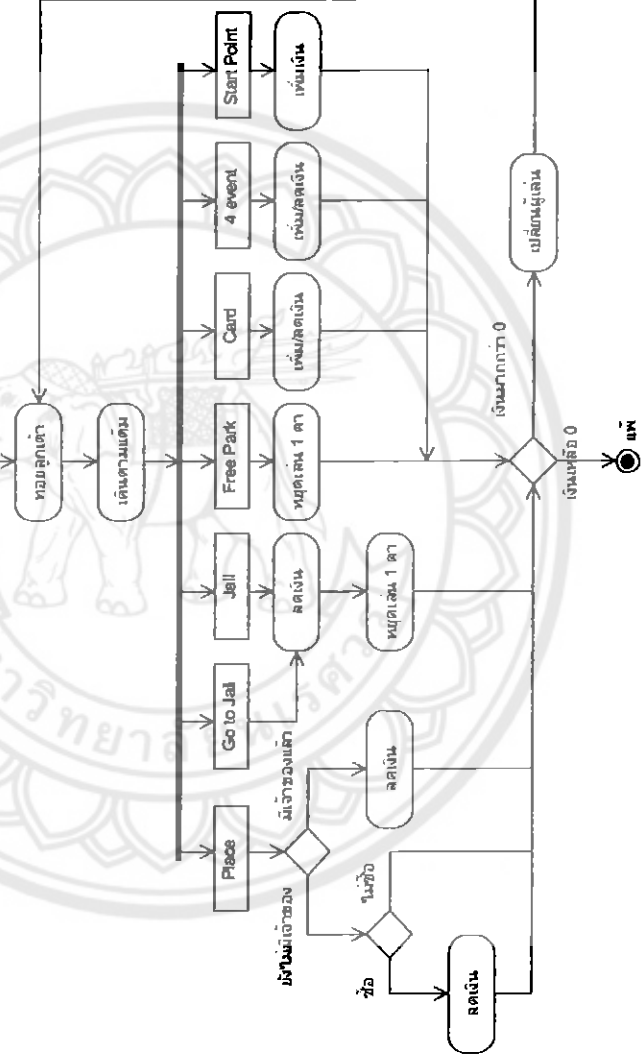
รูปที่ 3.4 แสดง Bill of Material ทั้งหมด

3.4.4 การออกแบบลำดับการทำงานของเกม

การออกแบบลำดับการทำงานเป็นส่วนที่จำเป็นสำหรับผู้เล่นเกมในทิศทางใหม่ได้บ้าง เพื่อให้โปรแกรมเมอร์สามารถนำเอา Flow Chart ในกรณีรับลำดับเหตุการณ์ ซึ่งลำดับเหตุการณ์จะเล่น และเงื่อนไขในการจบเกม

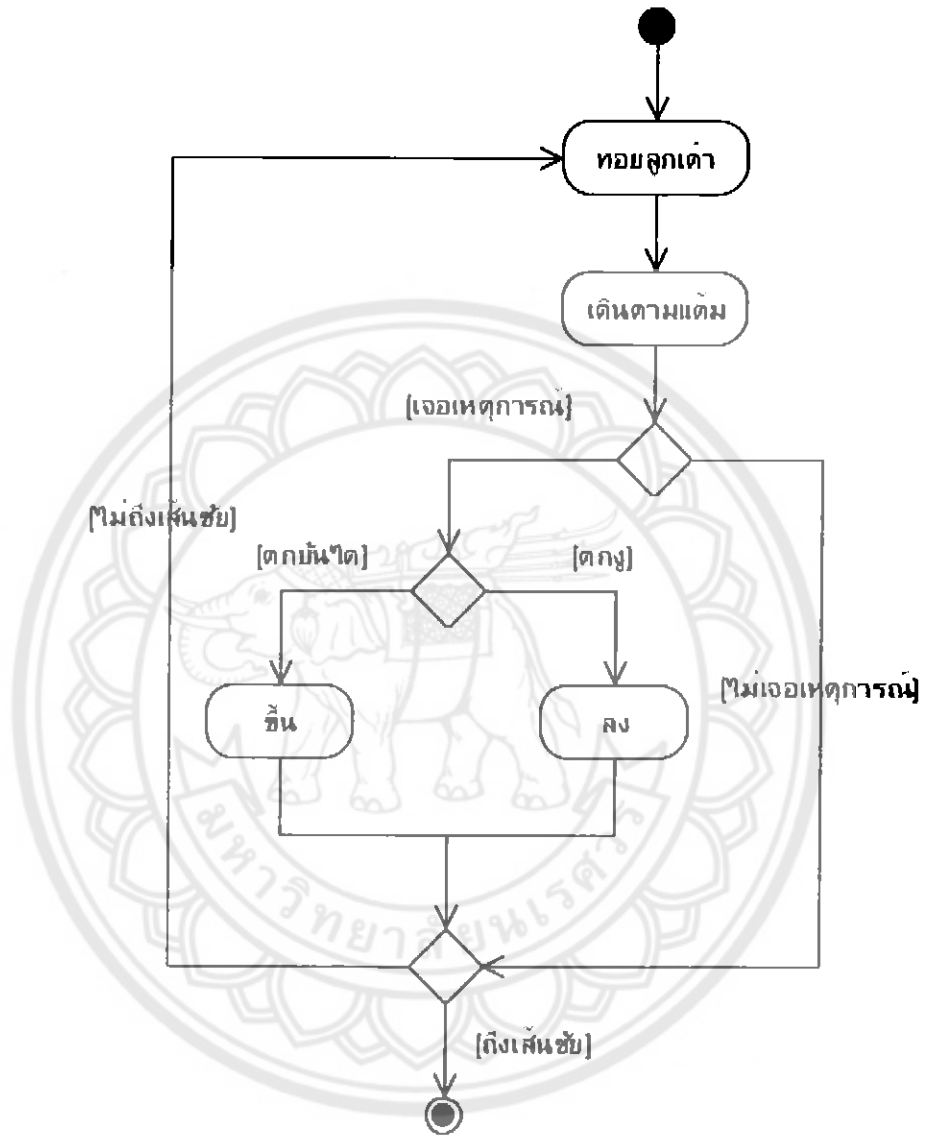
3.4.4.1 เกมเศรษฐี

เกม



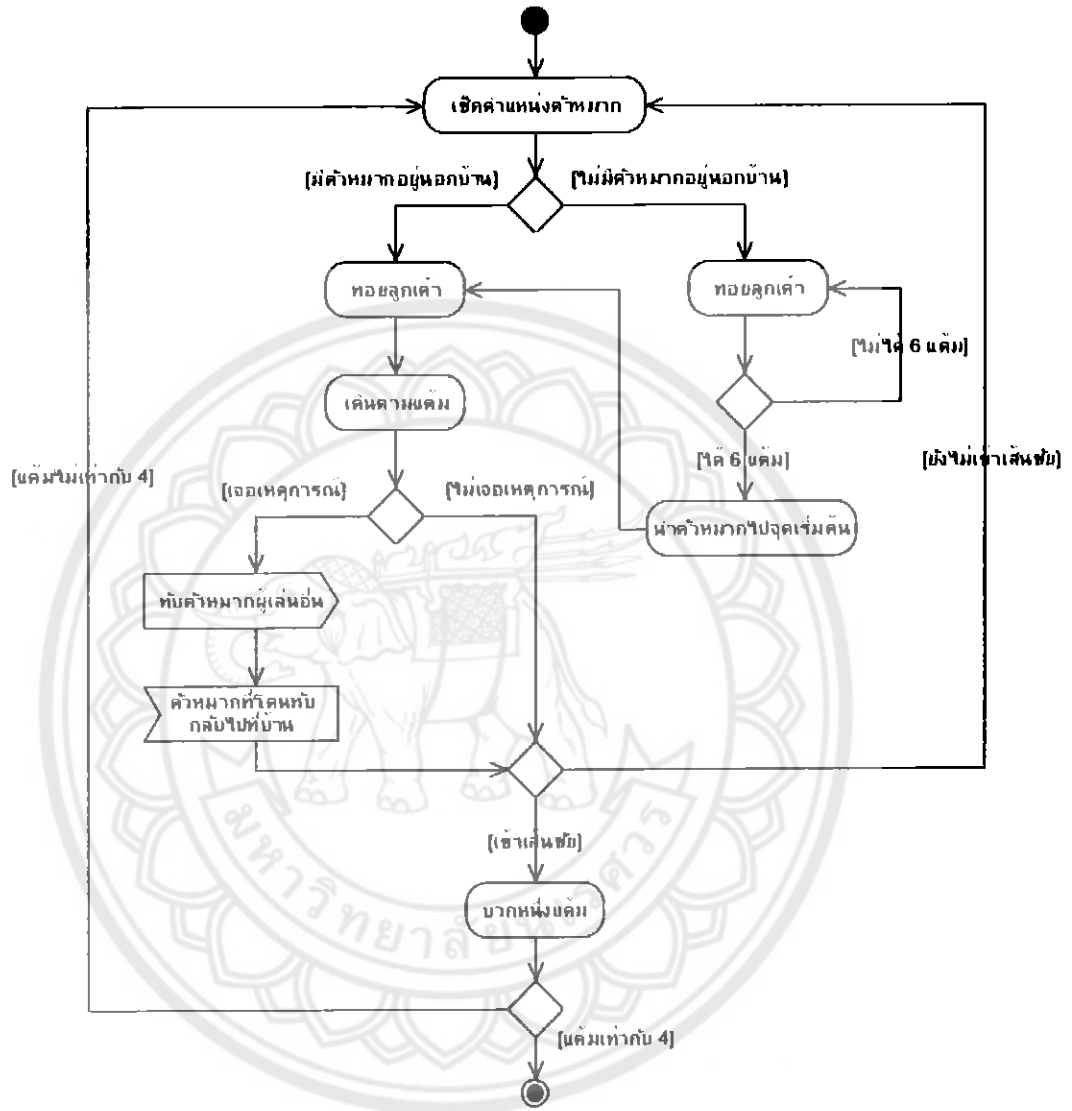
รูปที่ 3.5 แสดงเงื่อนไขการจบเกมของเศรษฐี

3.4.4.2 เกมบันไดงู



รูปที่ 3.6 แสดงเงื่อนไขการจบเกมของเกมบันไดงู

3.4.4.3 เกมลูโต้



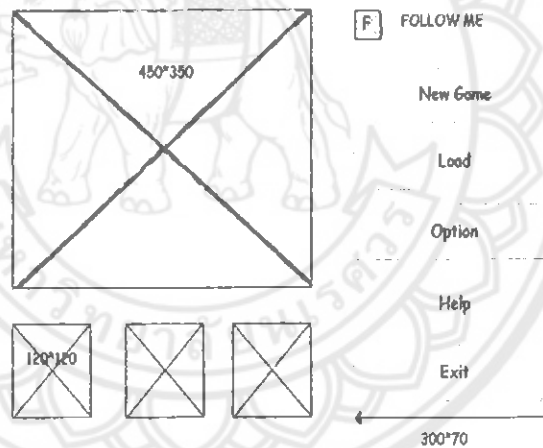
รูปที่ 3.7 แสดงเงื่อนไขการจบเกมของเกมลูโต้

3.4.5 การออกแบบ User Interface และ Character Image

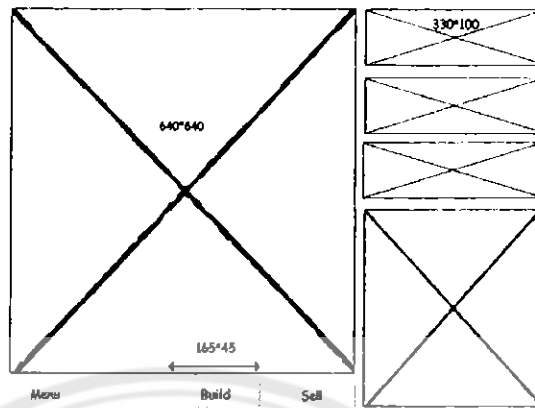
การออกแบบ User Interface และ Character Image เป็นอีกส่วนที่สำคัญต่อการ Reuse เพราะผู้ประกอบการ สามารถลดต้นทุนในส่วนนี้ได้ เริ่มจากการวิเคราะห์เกมที่ต้องการพัฒนา ว่าเป็น เกมแนวใด มีรูปแบบการเล่นอย่างไร เพื่อให้สามารถออกแบบ UI และตัวละคร ที่เหมาะสมเพียงครั้งเดียว ให้ใช้ได้ทั้งสามเกม เป็นการลดต้นทุน ไม่ต้องจ้างนักออกแบบหลายๆครั้ง

3.4.5.1 การออกแบบ User Interface

ในขั้นตอนการออกแบบ User Interface จะใช้โปรแกรม iPlotz เป็นเครื่องมือในการวาด Wireframe เพื่อออกแบบเค้าโครงของหน้าหลักต่างๆภายในเกม เช่น Index page, Lobby page, Game Play page and Option page โดยในการออกแบบจะกำหนดขนาดและตำแหน่งของส่วนแสดงผล เช่น ปุ่มกด รูปภาพ ตัวละคร และอื่นๆ



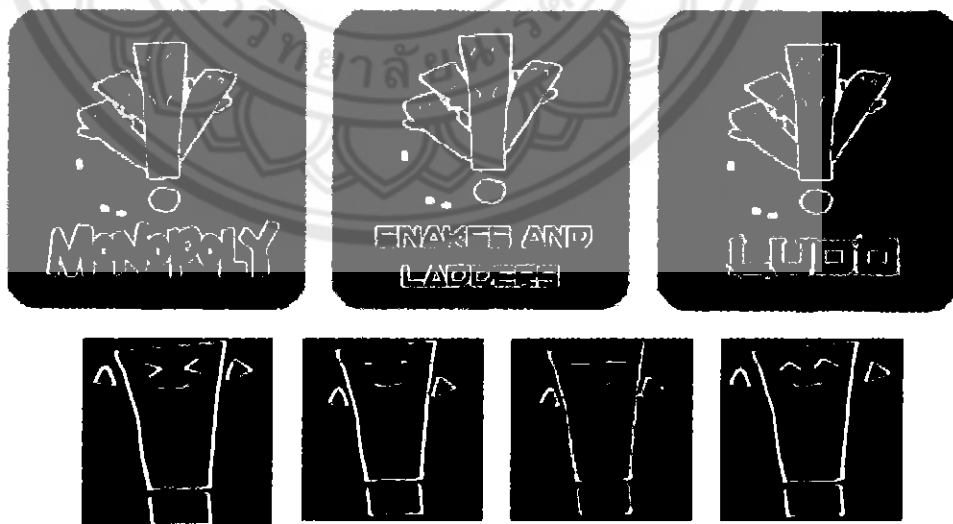
รูปที่ 3.8 ตัวอย่างการออกแบบ User Interface หน้า Index



รูปที่ 3.9 ตัวอย่างการออกแบบ User Interface หน้า Game Play

3.4.5.2 การออกแบบ Character Image

การออกแบบตัวละครของเกมให้เป็นที่จดจำ เป็นการเพิ่มคุณค่าให้เกมในแง่ของธุรกิจ ซึ่งการออกแบบที่ดีจะทำให้สามารถใช้ตัวละครกับเกมอื่นได้ เป็นการลดต้นทุน และทำให้ผู้เล่นคุ้นเคยกับตัวละครได้ง่าย ทั้งนี้ภาพลักษณ์ของตัวละครถูกออกแบบให้แสดงต่อผู้เล่นทั้ง 3 ส่วน คือ Logo ของเกม, ภาพตัวละครของผู้เล่น และส่วนตัวละครที่สามารถเคลื่อนไหว



รูปที่ 3.10 ตัวอย่างการออกแบบ Character Image

3.4.6 การเขียนโปรแกรม

ในขั้นตอนการเขียนโปรแกรม เกมจะถูกพัฒนาด้วยภาษา C# และใช้ XNA Framework เพื่อให้ง่ายสำหรับการเขียนโปรแกรมในลักษณะที่เป็นการเขียนเกม โดย Code ที่สามารถใช้ร่วมกันได้ ทั้ง 3 เกมจะถูกนำมาทำเป็น Library ด้วยการใช้ Dynamic Link Libraries (DLL) ซึ่ง Code เหล่านั้นสามารถเรียกใช้ได้ทันที โดยไม่มีการแก้ไขใดๆ ถือเป็น Reuse และประโยชน์สำหรับการนำ Code เก่ากลับมาใช้ใหม่คือ สามารถลดระยะเวลาเขียน Code ทั้งหมด และประโยชน์อื่นๆที่จะตามมาหลังจากสามารถลดระยะเวลาได้

สิ่งที่สำคัญอีกอย่างหนึ่งของการเขียนโปรแกรมคือ การเขียนทดสอบระบบ ในส่วนนี้จะใช้เวลาในการเขียนอีกพอสมควร ซึ่งหากสามารถสร้างจุดร่วมที่เป็นการใช้ Library ร่วมกันได้ในปริมาณมาก ก็จะส่งผลให้ลด Code ที่ต้องเขียนในการทดสอบระบบไปด้วย



บทที่ 4

ผลการทดลอง

4.1 วิธีการทดลอง

สำหรับวิธีการทดลองการใช้ Core Assets จะใช้การวัดจากจำนวน Line of Code, Name Space, Class และ Method ที่ใช้ในการเขียนเกมทั้งหมด สำหรับการวัดผลจะให้ความสำคัญกับส่วนที่มีการ Reuse หรือส่วนที่ไม่มีการแก้ไขใดๆเลยเท่านั้น ทั้งนี้จะเริ่มกระบวนการสร้าง เกมเสร็จสิ้น เกมบันไดงู และ เกมลูโต้ ตามลำดับ

เนื่องจากการ Reuse ที่เป็นหัวใจสำคัญของการสร้าง Core Assets สามารถนำไปใช้ได้หลายทาง ทั้งการ Reuse ในส่วนของ Code ที่ใช้ในการเขียนโปรแกรม การ Reuse แบบสอบถามความต้องการของผู้ใช้ การ Reuse User Interface ของทั้ง 3 เกม หรือแม้กระทั่งการ Reuse ตัวละครของเกม เป็นต้น ดังนั้นในโครงการนี้จะดูผลลัพธ์ที่ภาพรวมของการลดต้นทุนด้วยการใช้ Core Assets

4.2 ผลการทดลอง

ผลการทดลองที่ได้สามารถอธิบายได้สองรูปแบบคือ

1. ผลการทดลองในระดับโค้ด
2. ผลการทดลองในระดับการออกแบบ

4.2.1 ผลการทดลองในระดับโค้ด

ผลการทดลองในส่วนนี้จะเป็นการนำจำนวน Line of Code ของระบบที่สามารถใช้ร่วมกันได้ทั้ง 3 เกม (ส่วนนี้จะถูกนำไปทำ Library เพื่อให้เรียกใช้ได้ง่าย) เทียบกับปริมาณ Line of Code ของทั้ง 3 เกม เพื่อคำนวณหาปริมาณแรงงานทั้งหมดที่จะลดลง เมื่อเรียกใช้ Library ซึ่งใน Library จะจำแนกคลาส ตามหน้าที่ แบ่งเป็นออกเป็น 5 หมวดย่อย ตามตารางที่ 4.1

ตารางที่ 4.1 แสดงตารางคลาสทั้งหมดใน Library เทียบกับจำนวน Line of Code

แบ่งตามหมวดหมู่	คลาส (จำนวนบรรทัด)
Game Control	<ul style="list-style-type: none"> ● Sound (77)
Game Referent	<ul style="list-style-type: none"> ● Manual Guide (85)
Game Characteristic	<ul style="list-style-type: none"> ● Dice (167) ● Table (38) ● Player Character Display (73) ● History (95) ● Player Status (107)
Game System	<ul style="list-style-type: none"> ● Save (39) ● Load (119) ● Player System (471) ● Index Menu (66)
Game Business	<ul style="list-style-type: none"> ● Logo (65) ● Social Media (16)
รวม	1412 บรรทัด

สำหรับการวัดผลสามารถทำได้หลายรูปแบบ เช่น การคิดปริมาณแรงงาน (Line of Code) ทั้งหมดที่ลดลงเทียบกับแต่ละเกม ดังตารางที่ 4.2 หรือคิดปริมาณแรงงาน (Line of Code) ทั้งหมดที่ลดลงเทียบกับทั้ง 3 เกม ดังสมการที่ 4.1

ตารางที่ 4.2 แสดงตารางปริมาณแรงงานทั้งหมดที่ลดลงเทียบกับแต่ละเกม

	ชื่อเกม	Core Assets	Unique	Total	%Reuse
LOC	เกมเศรษฐี	1412	3935	5347	26.41%
	เกมบันไดงู	1412	2416	3828	36.89%
	เกมคู่ได้	1412	3157	4569	30.90%
Method	เกมเศรษฐี	67	68	135	49.63%
	เกมบันไดงู	67	69	136	49.26%
	เกมคู่ได้	67	59	126	53.17%
Class	เกมเศรษฐี	18	14	32	56.25%
	เกมบันไดงู	18	14	32	56.25%
	เกมคู่ได้	18	13	31	58.06%

การวัดผล โดยคิดปริมาณแรงงาน (Line of Code) ทั้งหมดที่ลดลงเทียบกับทั้ง 3 เกม จะมี
สมการในการคิดดังนี้

$$\frac{\text{ปริมาณ Line of Code ที่เขียนจริงทั้งหมด}}{\text{ปริมาณ Line of Code ทั้งหมด หากไม่มีการใช้ Library}}$$

สมการที่ 4.1 สมการการวัดผล โดยคิดปริมาณแรงงานทั้งหมดที่ลดลงเทียบกับเกมทั้งหมด

โดยที่ปริมาณ Line of Code ที่เขียนจริงทั้งหมด หมายถึง ปริมาณ Code ที่เขียนในแต่ละเกม เป็นจำนวน 3 เกม รวมกับปริมาณ Code ที่ทำเป็น Library ในเกมแรก ส่วนปริมาณ Line of Code ทั้งหมด หากไม่มีการใช้ Library หมายถึง จำนวน Code ทั้งหมดที่ควรจะเป็นหากไม่ได้ใช้ Library เพื่อลดปริมาณงาน โดยผลลัพธ์ที่ได้จะหมายถึง ปริมาณ Code ทั้งหมดที่เราเขียนจริง เทียบกับ Code ที่ไม่ได้ใช้ Library

$$\frac{(3935 + 3157 + 2416) + 1412}{(3935 + 1412) + (3157 + 1412) + (2416 + 1412)}$$

สมการที่ 4.2 สมการการวัดผลโดยคิดปริมาณแรงงานทั้งหมดที่ลดลงเทียบกับเกมทั้งหมด

จากสมการที่ 4.2 ทำให้ทราบว่าปริมาณ Code หลังการใช้ Library คือ 79.5 % นั้น หมายความว่าสามารถลดแรงงานจากการเขียน Code ทั้งหมดไป 20.5%

นอกจากนี้สามารถวัดผลจากการดูที่ Line of Code แล้วยังสามารถจำแนก วัดผลจากส่วน อื่นๆ ได้อีกด้วย เช่น Name Space, Class และ Method เป็นต้น ส่วนนี้จะทำให้ทราบถึงปริมาณจตุรร่วม เมื่อเทียบกับแต่ละเกม เพราะหากวัดที่จำนวน Line of Code เพียงอย่างเดียว อาจทำให้มองไม่เห็นถึง ประสิทธิภาพแท้จริงของ Software Product Line

ตารางที่ 4.3 แสดงตารางเปรียบเทียบจตุรร่วมในระดับต่างๆของแต่ละเกม

	จตุรร่วม	เกมเสริมฐึ	เกมบันไดงู	เกมลูได้
Name Space	5	1	1	1
Class	18	14	14	13
Method	67	68	69	59

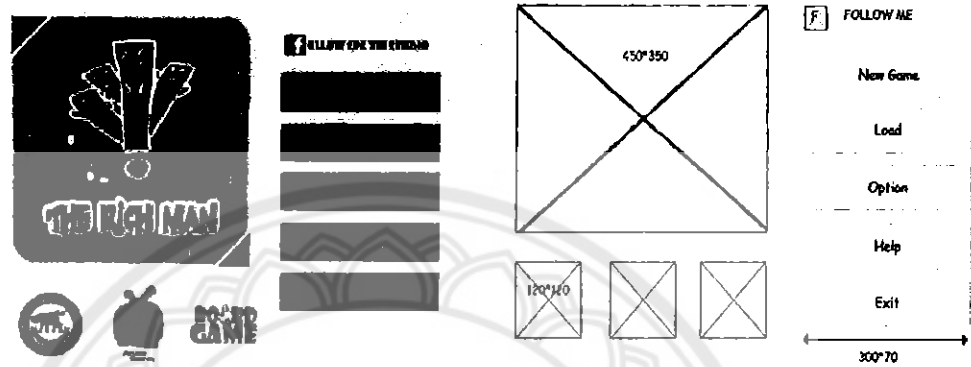
4.2.2 ผลการทดลองระดับการออกแบบ

สำหรับผลการทดลองในหัวข้อนี้ เป็นส่วนที่เกี่ยวข้องกับการ Reuse อีกประเภทหนึ่ง ซึ่งไม่สามารถวัดเป็นตัวเลขได้ นั่นคือการออกแบบ User Interface และ Character Image ทั้งสองอย่างที่กล่าว มาคือการนำกลับมาใช้ใหม่ ในรูปแบบที่ไม่ได้เป็น Code โปรแกรม ซึ่งในด้านธุรกิจอันเป็นส่วนหนึ่ง ของ Software Product Line จึงให้ความสำคัญกับส่วนนี้ด้วย

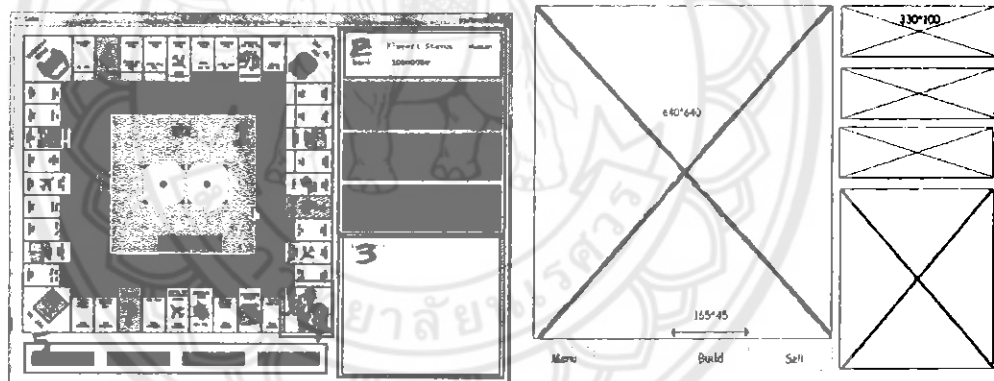
4.2.2.1 การออกแบบ User Interface

ถ้ารับการออกแบบ User Interface ที่ดีจะช่วยให้ เป็นการลดต้นทุนในการ ออกแบบเกมที่ 2 และเกมที่ 3 อาจกล่าวได้ว่าออกแบบเพียงครั้งเดียว สามารถใช้ได้หลายครั้งนอกจากนี้

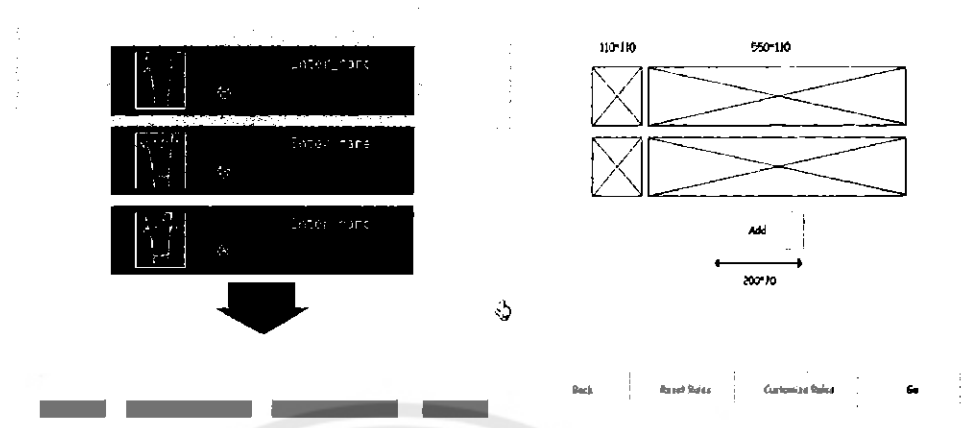
การใช้ User Interface เดียวกันในเกมประเภทเดียวกัน มาจากผู้ใช้ผลิตภัณฑ์เดียวกันยังสามารถทำให้ผู้ใช้คุ้นเคยกับเกมใหม่ๆ ได้ง่าย ถือเป็นเรื่องดีในการออกแบบเกม



รูปที่ 4.1 แสดงตัวอย่างการออกแบบ Use Interface ของ Index Page



รูปที่ 4.2 แสดงตัวอย่างการออกแบบ Use Interface ของ Game Play Page



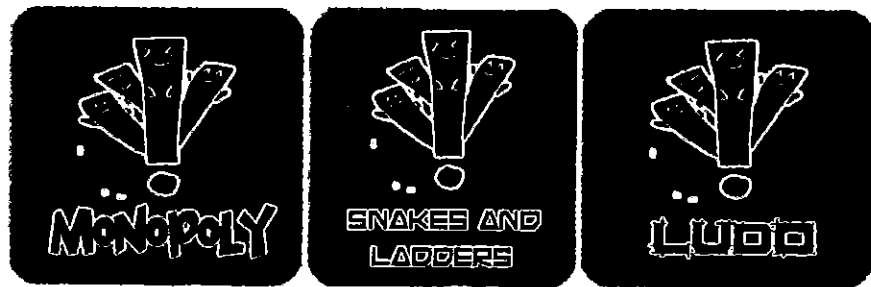
รูปที่ 4.4 แสดงตัวอย่างการออกแบบ Use Interface ของ Player System Page

ง การออกแบบ User Interface ทั้งหมด ทำให้สามารถพัฒนามเกมทั้งสามได้อย่างรวดเร็ว และเป็นระบบ ทางด้านธุรกิจทำให้ไม่จำเป็นต้องจ้างนักออกแบบหลายครั้ง ซึ่งถือเป็นหัวใจสำคัญในการ Reuse Core Asset

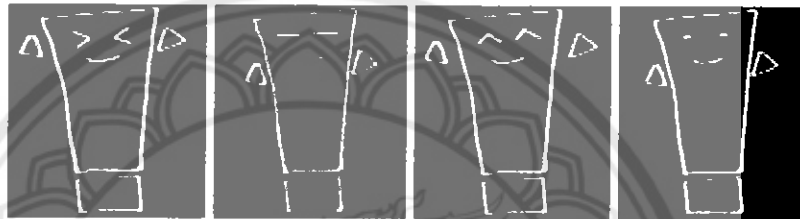
4.2.2.2 การออกแบบ Character Image

สำหรับการออกแบบตัวละครของเกมเป็นการ Reuse เช่นเดียวกับการออกแบบ User Interface เพราะประโยชน์ที่ได้จากการออกแบบตัวละคร คือการที่สามารถนำตัวละครกลับมาใช้กับเกมอื่นๆต่อไปได้ ซึ่งการที่จะออกแบบตัวละครให้สามารถใช้ได้หลายๆเกม จำเป็นต้องสร้างตัวละครที่มีเอกลักษณ์โดดเด่นไปประเภทของเกมนั้นๆ นอกจากนี้การทำให้ตัวละครของเกมโดดเด่นยังส่งผลถึงผลลัพธ์ด้านความพึงพอใจของผู้ใช้อีกด้วย

การออกแบบตัวละครของเกมให้เข้าถึงผู้เล่นจำเป็นต้องทำให้ผู้เล่นคุ้นเคยกับตัวละคร ซึ่งในที่นี้จะใช้ตัวละครมีโมเดลหลักของเกม โดยการแสดงภาพตัวละครที่ Logo ในหน้า Index ให้ผู้เล่นสามารถเลือกตัวละครได้ และจุดสำคัญคือตัวละครที่นำมาใช้ใน Game Play ต้องสามารถเคลื่อนไหว เพื่อตอบสนองความสนใจของผู้เล่นได้



รูปที่ 4.4 แสดงตัวอย่างการออกแบบ Character Image ในรูปแบบ Logo



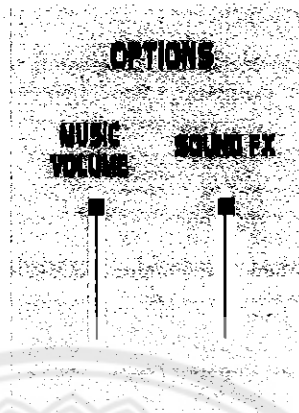
รูปที่ 4.5 แสดงตัวอย่างการออกแบบ Character Image ในรูปแบบการเลือกตัวละคร



รูปที่ 4.6 แสดงตัวอย่างการออกแบบ Character Image ในรูปแบบที่สามารถเคลื่อนไหวได้

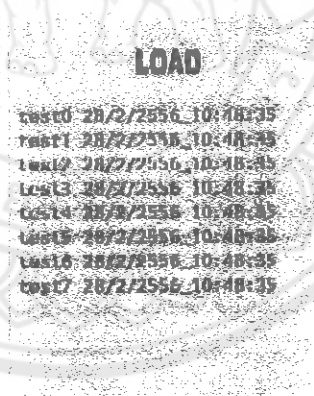
4.2.2.3 ตัวอย่าง User Interface ของระบบที่ใช้ร่วมกัน

หัวข้อนี้จะเป็นการยกตัวอย่างระบบที่ให้คอมพิวเตอร์ให้ใช้ร่วมกันทั้งสามเกม เพื่ออธิบายรายละเอียดของระบบที่ได้นำมาทำเป็น Library โดยโปรแกรมที่เรียกใช้ Library จะมีการรับค่าต่างๆ ที่จะแตกต่างกันไปในแต่ละเกม ทำให้แต่ละเกมมีส่วนที่แตกต่างกันเล็กน้อย ตามรายละเอียดปีกย่อยของเกม



รูปที่ 4.7 แสดงระบบ Sound Effect / Music Volume

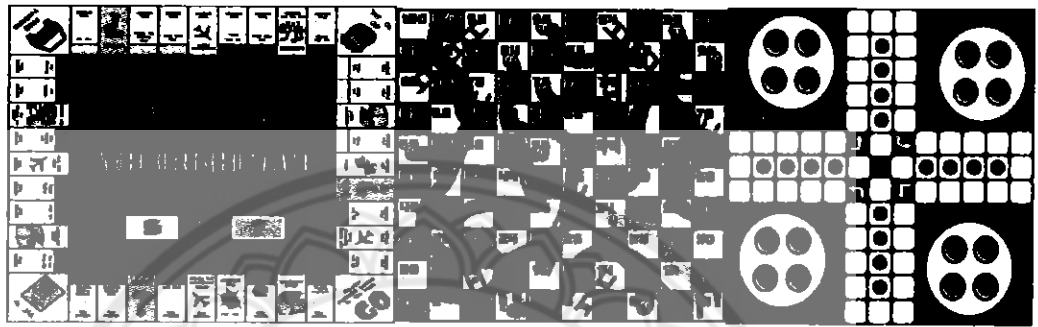
ในระบบ Save / Load ของทั้ง 3 เกม จะระบุ Save ID วันที่และเวลาในการเซฟ โดยสามารถเซฟได้ทั้งหมด 10 ครั้ง หากเกินจะทำการลบเซฟที่เก่าที่สุดไปเพื่อทดแทน Slot



รูปที่ 4.8 แสดงระบบ Save / Load

ในส่วนของการระบบ Manual Guide เป็นส่วนที่อธิบายวิธีการเล่นเกม โดยระบบจะเหมือนกันทั้ง 3 เกม และสามารถเข้าถึงได้จากหน้าหลัก และขณะเล่นเกม เนื้อหาของเกมจะแตกต่างกันไปตามเกมที่เล่น

กระดานของเกมทั้งสามมีความแตกต่างกันมาก ทำให้ในการเขียนโปรแกรมคลาสที่ใช้ในการสร้างกระดานเป็นคลาสที่ใช้ปริมาณ Code มากที่สุด ซึ่งหากใช้เกมที่มีรูปแบบตารางคล้ายกัน จะทำให้สามารถลดปริมาณ Code ที่ใช้เขียนคลาสสร้างกระดานไปได้อีก



รูปที่ 4.11 แสดงภาพกระดานของทั้ง 3 เกม



บทที่ 5

สรุปผลการดำเนินงานและแนวทางการพัฒนา

โครงการการสร้างเกมพร้อมกันด้วยวิธีสร้าง Core Assets เป็นโครงการที่จัดทำขึ้น โดยมีจุดประสงค์ที่จะสร้างเกมเศรษฐกิจ เกมบันไดงู และเกมลูกได้ โดยใช้หลักการ Reuse ซึ่งเป็นส่วนหนึ่งของ Software Product Line ในการพัฒนาสายผลิตภัณฑ์ โดยเน้นการลดต้นทุนของผู้ผลิตในด้านต่างๆ และเพิ่มคุณภาพให้กับผลิตภัณฑ์ โดยการพัฒนาเกมทั้ง 3 เกมนี้ทดสอบและรันบนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 7 ใช้ XNA Game Studio โดยมีพื้นฐานอยู่บนภาษา C#

ผลที่ได้รับคือ สามารถสร้างเกมทั้ง 3 เกมโดยใช้ Core Assets ในการสร้าง โดยจะทำการออกแบบว่าเกมทั้งหมดมีจุดร่วมอะไรบ้าง ซึ่งมองจากความต้องการจากผู้บริโภค เช่น ลักษณะของเกม แนวทางการเล่นเกม จุดประสงค์ของเกม ฯลฯ เป็นต้น เมื่อได้จุดร่วมของทั้ง 3 เกมมาแล้ว ขั้นตอนต่อไปคือการนำจุดร่วมนี้มาทำเป็น Library เพื่อให้สามารถเรียกใช้ได้ทันที ไม่ต้องเขียนใหม่ทุกครั้ง ซึ่งจะทำให้ลดเวลาในการสร้างเกม และได้เกมที่มีคุณภาพ โดยผลการทดลองสามารถสรุปได้ดังนี้คือ

ตารางที่ 5.1 แสดงตารางสรุปผลการทดลอง

การทดลอง	ผลการทดลอง
สร้าง 3 เกมด้วยการใช้ Core Assets เดียวกัน	✓
ออกแบบ โมเดลสำหรับจาก Core Assets	✓
ลดปริมาณ Code ในการสร้างเกม	✓

5.1 สรุปผลการทดลอง

จากตารางที่ 4.1 จะเห็นว่าสามารถสร้าง Library จากจุดร่วมของ 3 เกมได้ ทั้งหมด 1412 บรรทัด โดยที่สามารถเทียบกับจำนวนบรรทัดที่ต้องเขียนเพิ่มได้ดังสมการที่ 4.1 จะเห็นว่าการใช้จุดร่วมจากหลักการของ Software Product Line สามารถลดปริมาณ Code ที่ต้องเขียนไปได้ถึง 20.5% ของการเขียนเกมทั้งหมด 3 เกม

จากตารางที่ 4.2 สามารถบ่งชี้ถึงปริมาณหน่วยย่อยเช่น Class, Method เมื่อเทียบระหว่างจุดร่วมกับ Code ที่ต้องเขียนเพิ่มทั้ง 3 เกม ซึ่งจะเห็นได้ว่าการใช้ Library ที่สร้างมาจากการหาจุดร่วมก่อนนั้น สามารถลดปริมาณ Class และ Method ได้ในปริมาณครึ่งหนึ่งของ Class และ Method ทั้งหมดที่ต้องเขียน

นอกจากนี้ยังมีส่วนที่สามารถลดปริมาณงานที่ต้องทำได้อีกคือ การใช้ User Interface และ Character Image ร่วมกัน โดยที่ทั้งสองอย่างสามารถลด Cost ที่จะต้องเสียได้ในปริมาณมาก ทั้งในด้านระยะเวลาสำหรับการผลิต ต้นทุนในการจ้างนักออกแบบ และอื่นๆ

5.2 ปัญหาและแนวทางการแก้ไข

ตารางที่ 5.2 ปัญหาและแนวทางการแก้ไขในการดำเนินโครงการ

ปัญหาและอุปสรรค	แนวทางการแก้ไข
1. ในการเริ่มพัฒนาเกมแรก ใช้เวลาในการสร้างนานมากเพราะ ต้องศึกษา XNA Game Studio ให้เข้าใจในการเรียกใช้ Library ต่างๆให้ถูกต้อง	1. ศึกษาเฉพาะฟังก์ชันที่จำเป็นต้องใช้ในการเขียน XNA Game Studio ในส่วนของการพัฒนาเกม 2D
2. AI ที่ใช้ในการเล่นเกมเศรษฐกิจและเกมโรงพยาบาลได้ถูกตั้งทำให้เล่นแบบก้ำกึ่งค่าไว้ ไม่ใช่การเล่นแบบเลือกวิธีที่ดีที่สุด	2. ทำการเพิ่มเงื่อนไขในการเล่นของ AI โดยใช้ if else เพื่อให้ AI เลือกทางที่ดีที่สุด

ตารางที่ 5.2 (ต่อ) ปัญหาและแนวทางการแก้ไขในการดำเนินโครงการ

ปัญหาและอุปสรรค	แนวทางการแก้ไข
3. การใช้ BDD ไม่สามารถใช้ Test ของ Visual studio 2010 สร้างขึ้นมาได้ เนื่องจาก Class ของ XNA ไม่สามารถนำไปใช้ใน Test ได้	3. เขียนการทดสอบด้วยมือ หรือหาโปรแกรมทดสอบด้วย BDD ใหม่

5.3 ข้อเสนอแนะ

จากปัญหาและอุปสรรคที่พบในข้อ 5.2 คณะผู้จัดทำจึงมีข้อเสนอแนะสำหรับผู้ที่ต้องการนำไปศึกษาต่อ ควรจะต้องมีความรู้ในสิ่งต่อไปนี้เพื่อที่จะสามารถนำไปพัฒนาเพิ่มเติมได้คือ

1. การมีขงเกมโดยใช้ XNA Game Studio สามารถศึกษาเพิ่มเติมได้ที่ www.thaixna.com หรือ <http://rbwhitaker.wikidot.com/xna-tutorials>
2. สามารถเขียนโปรแกรมเพียงโดยการเปลี่ยน Platform เป็น Web application เพื่อสามารถทำให้เล่นกับผู้เล่นใน Internet ได้
3. ศึกษาการเขียนโปรแกรมด้วยภาษา C# และการใช้โปรแกรม Visual Studio C# เนื่องจาก XNA Game Studio จะใช้ภาษา C# ในการเขียน
4. ศึกษารูปแบบในการเขียน Code แบบการเขียนโปรแกรมเชิงวัตถุ (Object oriented) เพื่อให้เกมมีประสิทธิภาพในการทำงานมากขึ้น
5. ในการเลือกเกมที่จะนำมาสร้างสาขาการผลิตควรเลือกเกมที่มีคุณลักษณะคล้ายกัน ควรมองจากวิธีการเล่น หาดจุดที่เหมือนกันของเกมให้มากที่สุด

เอกสารอ้างอิง

- [1] John K. Bergey and Lawrence G. Jones, **Exploring Acquisition Strategies for Adopting a Software Product Line**, Software Engineering Institute, Carnegie Mellon University, 2010.
- [2] Paul Clements, Linda Northrop, **Software Product Lines: Practices and Patterns**, 3rd, 2001
- [3] กิตติพงษ์ กลยกลัสม, **การวิเคราะห์และออกแบบระบบเชิงใช้ซ้ำด้วย SPL**, เกร็ทที คอมพ์ แอนด์ คอนซัลท์, บจก.



ภาคผนวก ก

วิธีการติดตั้ง XNA Game Studio 4.0

XNA Game Studio คือเครื่องมือที่ประกอบด้วย Runtime ของ Microsoft โดยจะช่วยอำนวยความสะดวกแก่ผู้ออกแบบเกมคอมพิวเตอร์ การพัฒนาเกม และการจัดการต่างๆ ด้วยการใช้ XNA นี้ จะทำให้ผู้พัฒนาและออกแบบเกมไม่จำเป็นต้องเขียนโค้ดเป็นจำนวนมาก และทุกองค์ประกอบในการสร้างเกมจะมารวมอยู่ในระบบเดียวกัน

วิธีการติดตั้ง XNA Game Studio 4.0 มีความต้องการของระบบดังนี้

- การรันเกมโดยใช้ XNA Framework นั้นจำเป็นต้องมีการ์ดจอที่สนับสนุน Shader Model 1.1 ขึ้นไป
- ระบบปฏิบัติการ Windows XP Service Pack 2 ขึ้นไป
- ต้องมี DirectX 9.0c
- ต้องมี Visual Studio 2008 อย่างต่ำแนะนำปัจจุบันนี้การใช้ Visual Studio Express 2010
- ต้องการ .Net Framework 3.5 ขึ้นต่ำ

ขั้นตอนการติดตั้ง XNA Game Studio 4.0

1. ต้องทำการยกเลิกการติดตั้ง XNA Game Studio เวอร์ชันเก่าออกจากเครื่องก่อน
2. ทำการ Download XNA Game Studio 4.0 ได้จาก <http://www.microsoft.com/en-us/download/details.aspx?id=23714>

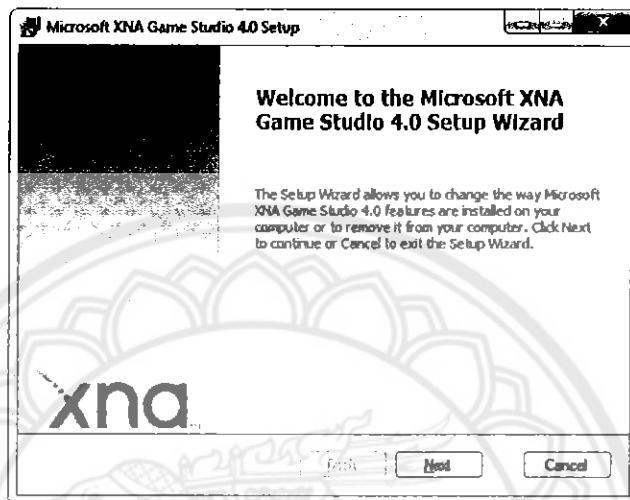
Microsoft XNA Game Studio 4.0 makes it easier than ever to create great video games for Windows-based PCs, Xbox 360 consoles, and Windows Phone

Quick details

Version:	4.0	Date published:	9/16/2010
Language:	English		
File name	Size		
XNAGS40_setup.exe	48.8 MB	DOWNLOAD	

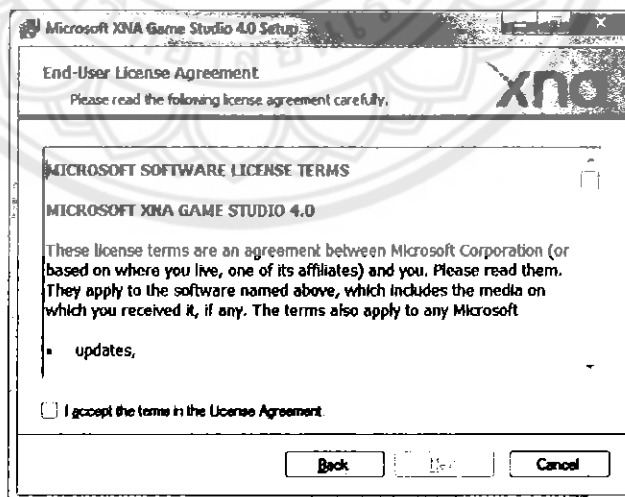
รูป ก.1 แสดงการ Download XNA Game Studio 4.0

3. หลังจากทำการ Download เสร็จสิ้นแล้วให้ดับเบิลคลิกที่ไฟล์ XNAGS40_setup.exe โปรแกรมจะทำการ Extract File เพื่อทำการลงโปรแกรม เมื่อทำ Extract File ก็จะเข้าสู่หน้าจอการลงโปรแกรม



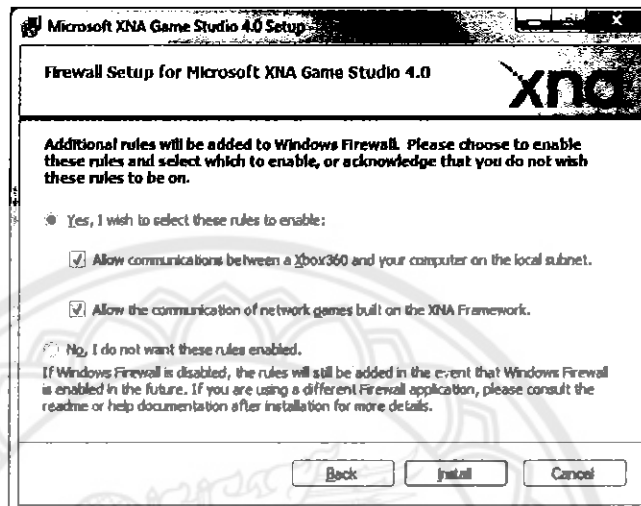
รูป ก.2 แสดงหน้าจอการลงโปรแกรม

4. หลังจากกด Next จะเข้าสู่หน้าจอแสดงรายละเอียดเงื่อนไขข้อตกลงเกี่ยวกับการนำโปรแกรมไปใช้งาน License Agreement



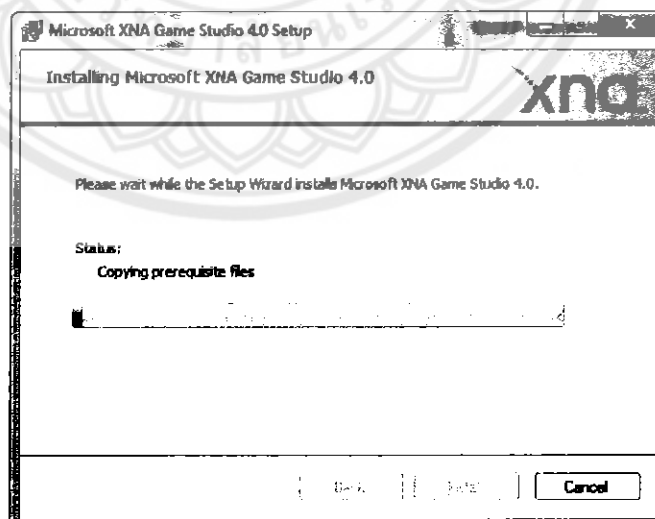
รูป ก.3 แสดงหน้าจอ License Agreement

5. เมื่อคลิกยอมรับข้อตกลงและกด Next หน้าต่อไปนี้จะแสดงเกี่ยวกับการจัดการ Firewall ของเครื่องและระบบ Network ของเกมที่จะสร้างขึ้น



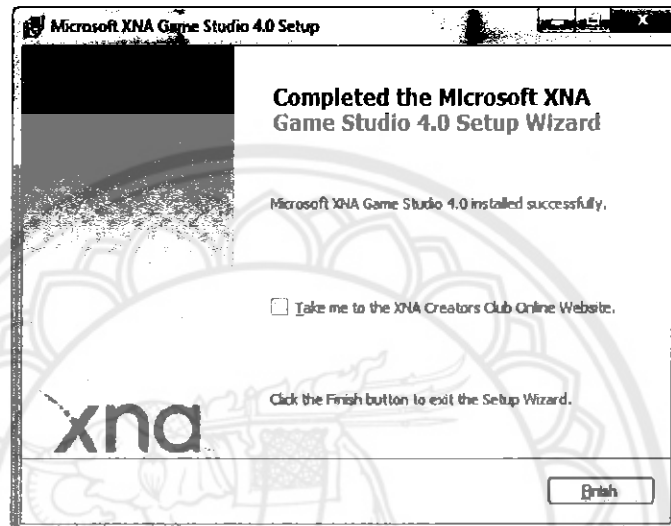
รูป ก.4 แสดงหน้าจอการจัดการเกี่ยวกับ Firewall สำหรับ XNA

6. หลังจากทำการยอมรับกด Yes เพื่อจัดการระบบ Firewall ให้สามารถทำงานกับ โปรแกรม
ได้ร้วกด Install เพื่อทำการลงโปรแกรม



รูป ก.5 แสดงหน้าจอการ Install โปรแกรม

7. เมื่อทำการติดตั้งโปรแกรมเสร็จแล้ว จะปรากฏหน้าจอว่าทำการติดตั้งสำเร็จแล้ว และกด Finish เพื่อทำการติคหน้าต่างการติดตั้งโปรแกรม



รูป ก.6 แสดงหน้าจอการติดตั้งสำเร็จ

ภาคผนวก ข

วิธีการติดตั้ง Microsoft .NET Framework 4.0

.NET Framework เป็นสภาพแวดล้อมการทำงานซึ่งรองรับการเขียนโปรแกรมในรูปแบบของ Microsoft .NET หรือภาษาตระกูล C#, VB, J#, F# และ C++ .NET ซึ่งผู้ใช้งานคอมพิวเตอร์ส่วนใหญ่ มักไม่เข้าใจว่ามันคืออะไร จริงๆแล้ว .Net Framework เป็นกลุ่มของ Runtime Library (.dll) ที่เป็น ตัวเชื่อมระหว่าง OS (Windows, Linux, Mac) กับซอฟต์แวร์ที่เขียนขึ้น โดยภาษาในเครื่องข่านั้น

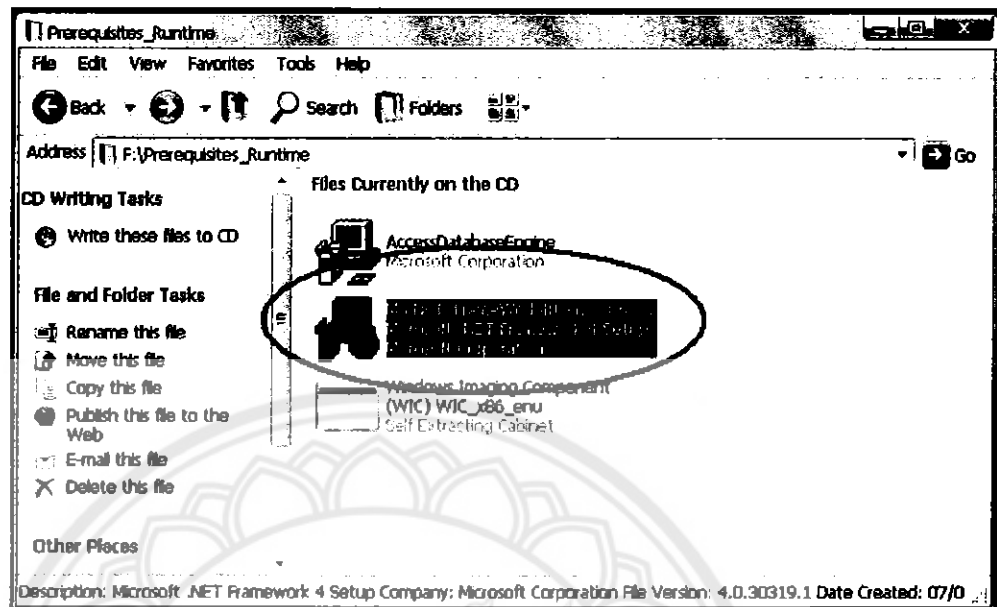
ขั้นตอนการติดตั้ง Microsoft .NET Framework 4.0

1. Download ไฟล์ติดตั้ง Microsoft .NET Framework 4.0 จาก <http://www.microsoft.com/en-us/download/details.aspx?id=17718> เพื่อนำไฟล์มาติดตั้ง



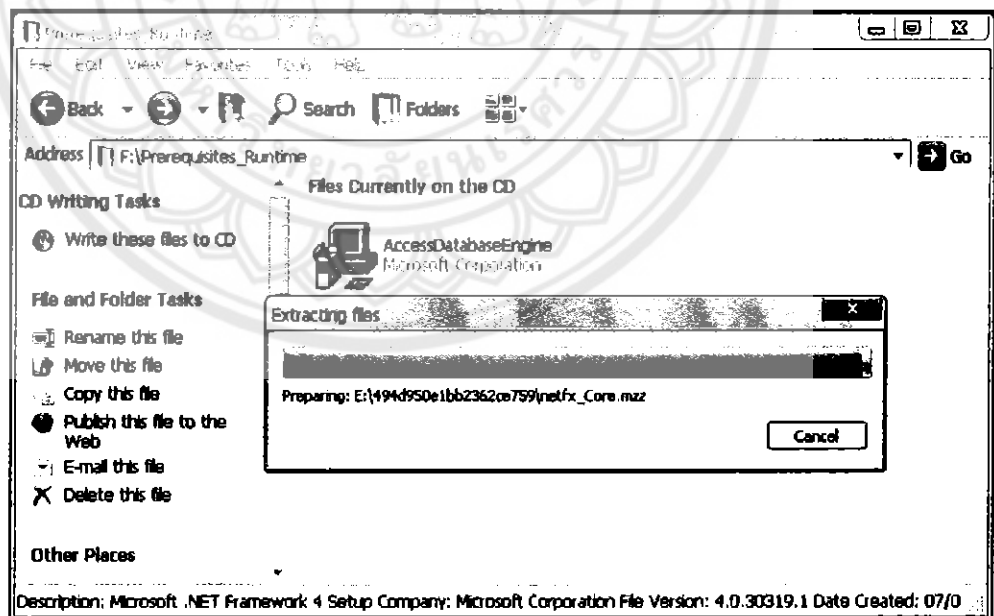
รูป ข.1 แสดงการ Microsoft .NET Framework 4.0

2. เข้าไปยังโฟลเดอร์ที่อยู่ของไฟล์ที่ Download แล้วดับเบิลคลิกที่ ไฟล์ MS.NET_FrameWork40_Full_x86_x64.exe เพื่อเริ่มติดตั้ง .NET Framework 4.0



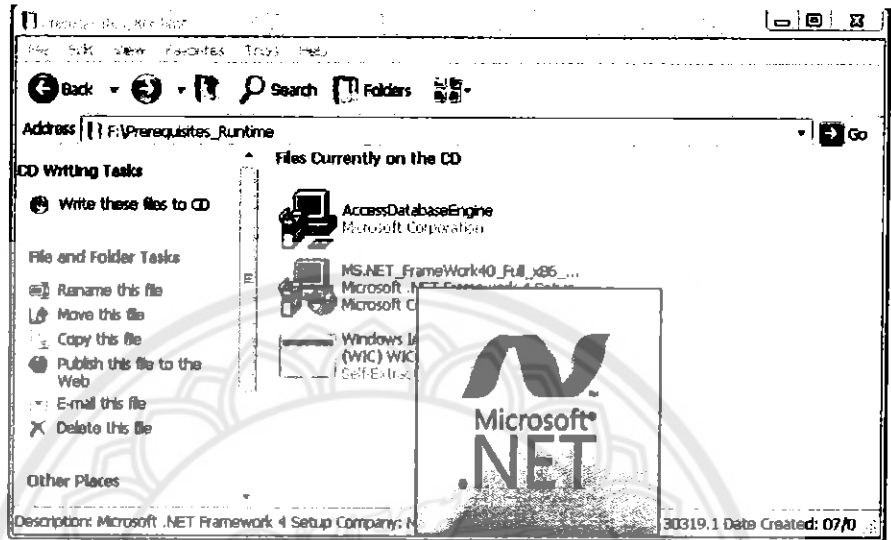
รูป ข.2 แสดงไฟล์ที่ใช้ในการติดตั้ง Microsoft .NET Framework 4.0

3. รอโปรแกรมกำลังทำการแตกไฟล์เพื่อทำการติดตั้ง จนกว่าจะเสร็จ



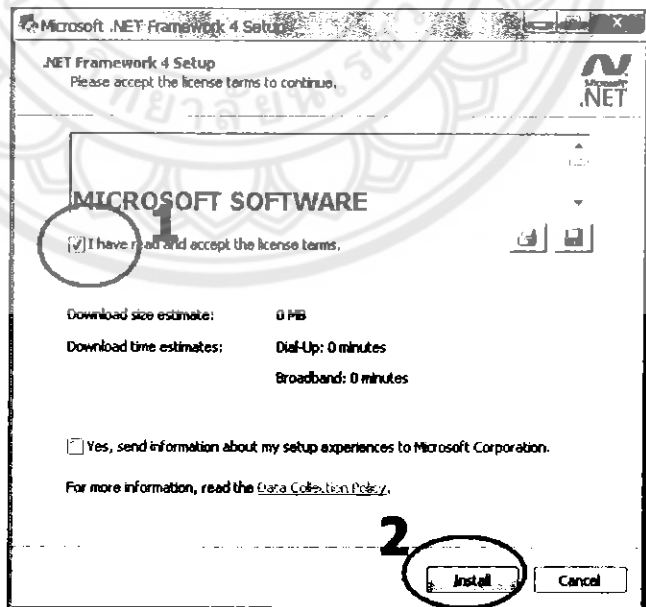
รูป ข.3 แสดงการแตกไฟล์เพื่อนำมาทำการติดตั้ง

4. ให้รอสักครู่เพื่อโหลดติดตั้ง .NET Framework



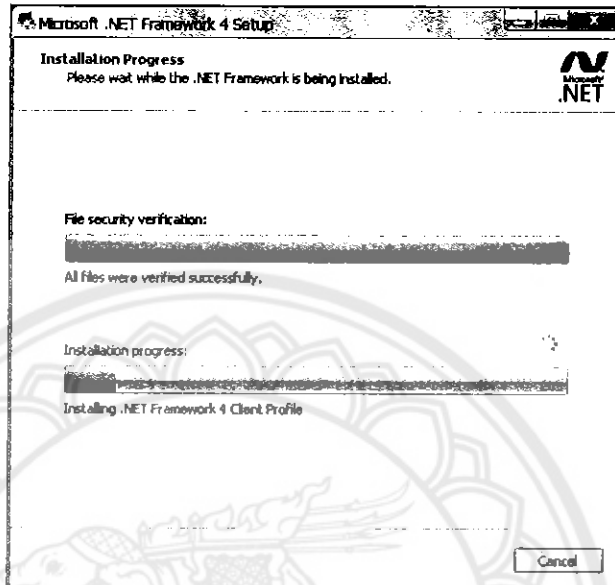
รูป ข.4 แสดง GUI ก่อนเริ่มการติดตั้ง

5. ให้ทำเครื่องหมายถูกตรงจุดที่ 1 และ คลิกปุ่ม Install ตรงจุดที่ 2 เพื่อเริ่มติดตั้ง .NET Framework



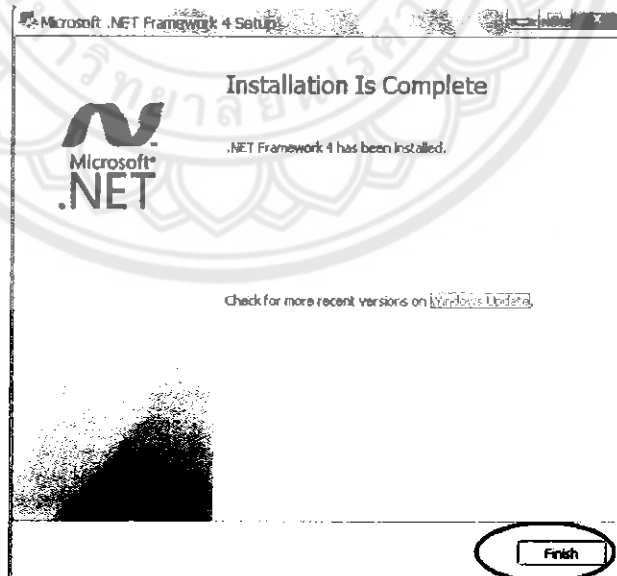
รูป ข.5 แสดงการยอมรับเงื่อนไข เพื่อบำทำการลงโปรแกรม

6. รอโปรแกรมกำลังดำเนินการติดตั้งลงเครื่อง



รูป ข.6 แสดงหน้าจอการติดตั้งโปรแกรม

7. หลังจากติดตั้งสำเร็จ กด Finish สิ้นสุดการติดตั้ง



รูป ข.7 แสดงหน้าจอการติดตั้งเสร็จสิ้น