สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เกม 3 มิติเพื่อการศึกษา 1

3D Game for Edutainment I

นาย ปรัชญา พุฒพนาทรัพย์

เลขหมู่.....
เลขทะเบียน 61475
วัน,เดือน,ปี 18 ก.ค. 2549

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกม 3 มิติเพื่อการศึกษา 1

3D Game for Edutainment I

จัดทำโดย

ทีมพัฒนาโครงการ

นาย ปรัชญา พุฒพนาทรัพย์ รหัส44010290 ชั้นปีที่4 ห้องD

เสนอต่อ

อาจารย์ที่ปรึกษาโครงการ

ดร.วรวัฒน์ ลิ้มโภคา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

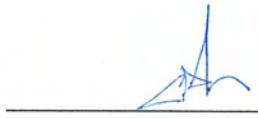
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เกม 3 มิติเพื่อการศึกษา 1

3D Game for Edutainment I

ผู้จัดทำ

1 นาย ปรัชญา พุฒพนาทรัพย์ รหัส44010290 ชั้นปีที่4 ห้องD



อาจารย์ที่ปรึกษา

(ดร.วรวัฒน์ ลิ้มโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกม 3 มิติเพื่อการศึกษา 1

นาย ปรัชญา พุฒพนาทรัพย์
รหัสประจำตัว 44010290

ดร.วรวัฒน์ ลิ้มโภคา
ปีการศึกษา พ.ศ. 2547

บทคัดย่อโครงการ

วิทยานิพนธ์ฉบับนี้ต้องการนำเสนอวิธีการนำความรู้ทางซอฟท์แวร์เอนจิเนียริ่ง(software engineering)ไปประยุกต์ใช้เพื่อการพัฒนาเอนจินของเกม เอดดูเทรนเม้นท์(Edutainment)เพื่อการเรียนรู้วิธีการโปรแกรมมิ่ง(programming)

โดยในระบบเกมของเรานั้นจะประกอบไปด้วย2ส่วนคือ

1 ส่วนของการเรียนรู้โปรแกรม(program)ที่เราจะทำการพัฒนาเป็นลักษณะของเอไอ ซิมมูเลชั่น เอนจิน (AI Simulation Engine) ในเกมเพื่อให้ผู้ใช้(user)เขียนโปรแกรมแล้วซิมมูเลท ให้ดู ว่าทำงานตามที่ผู้ใช้เขียนไว้หรือไม่ โดยมีความตั้งใจที่จะสอนการประยุกต์ใช้โครงสร้างของเอเจนท์(Structure of Agent)เป็นหลัก

2 ส่วนของความบันเทิงโปรแกรมที่เราจะทำการพัฒนาเป็นลักษณะของเกม3Dในเชิง แอ็คชั่น อาพีจี (Action RPG) ซึ่งสามารถเล่นได้เหมือนเกมจริงๆ โดยผู้ใช้สามารถควบคุมตัวละครของตนได้ด้วยเมาส์(mouse) และคีย์บอรด์(keyboard)แต่ถ้าจะควบคุมเอนพีซี(NPC)ต้องใช้วิธีการเขียนโปรแกรม ซึ่งจะได้ลักษณะของเกมสเตรท์ทีจี(strategy)ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3D Game for Edutainment I

Mr.Prachaya Puttapanasub

ID 44010290

Dr. Voravat Limpoca

Year 2004

Abstract

This thesis deals with the study of applying the software engineering for building an Edutainment Game's Engine for Programming Learning

There are 2 Engines in Our Project.

1 AI Simulation Engine for Education part .This Engine use for simulating user's code which is true or not in our project it use for training " Structure of Agent" knowledge.

2 3D Simulation Engine for Entertainment part. This engine use for make feeling like playing normal 3D Action RPG . It can play by control Player Character by mouse and keyboard , And NPC by Programming from AI Simulation Engine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ดร.วรวัฒน์ ลิ้มโภคา ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประการณ์ในการทำงานจริงแก่ข้าพเจ้า

ขอขอบพระคุณอาจารย์ ดร. วิศิษฏ์ หิรัญกิตติ ที่ให้ความช่วยเหลือในการแนะนำภาษาไพทอน (Python) และให้ยืมหนังสือเกี่ยวกับUML 3เล่ม 1 The Unified Modeling Language Reference Manual 2 The Unified Software Development Process 3 The Unified Modeling Language User Guide

ขอขอบพระคุณอาจารย์รศ. ดร. เอื้อน ปิ่นเงินที่ให้ความช่วยเหลือในการแนะนำแนวการพัฒนาเอนจินต่อสู่การค้นคว้าด้านเอไอ(AI)ระดับสูง(Soft Computing)และเทคนิคต่างๆทางด้านเอไอ

ขอขอบพระคุณอาจารย์ ผศ. อภิเนตร อูนากูล สำหรับความรู้เรื่องเบส แพคทิส ออฟ ซอฟท์แวร์เอนจิเนียริ่ง(Best Practices of Software Engineering)และการประยุกต์ใช้งานซอฟท์แวร์เอนจิเนียริ่ง

ขอขอบพระคุณอาจารย์ เกียรติกูล เจียรนัยธนะกิจสำหรับความรู้เรื่องโครงสร้างของเอเจนท์(Structure of Agent)

ขอขอบพระคุณครูอาจารย์ทุกท่านที่ได้ประสิทธิประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

ขอขอบคุณนาย ประกิจ อิงคกิตติ นักศึกษาปริญญาโทสำหรับการสอนเรื่องเกมเอนจิน(Game Engine)

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

นาย ปรัชญา พุฒพนาทรัพย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
สารบัญตาราง	VIII
บทที่1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	2
1.3 สมมุติฐานของการศึกษา	3
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย	3
1.5 ขอบเขตของงานวิจัย	3
บทที่2 วรรณกรรมหรืองานวิจัยที่เกี่ยวข้อง	4
2.1 บทความช่วงอายุที่สัมพันธ์กับการเรียนรู้อ้างอิงโดย	4
2.2 บทความด้านการจัดแบ่งประเภทของเกมเอดคูเทรนเม้นท์	4
2.3 ทฤษฎีโครงสร้างเอเจนท์(Structure of Agent)	5
2.4 ทฤษฎีเบสเพ็คทิสออฟซอฟท์แวร์เอนจิเนียริ่ง	10
2.5 โครงสร้างองค์กรตามหลักยูนิไฟซอฟท์แวร์โพรเสค	11
2.5.1 เฟสการจับประเด็นความต้องการ(Requirement Captures)	11
2.5.2 เฟสการวิเคราะห์(Analysis)	12
2.5.3 เฟสการออกแบบ(Design)	13
2.5.4 เฟสปฏิบัติการ (Implementation)	14
2.5.5 เฟสการทดสอบ(Test)	15
2.6 ทฤษฎีเกมเอนจิน	16
2.6.1 Direct-X	16
2.6.1.1 Direct3D	16
2.6.1.2 DirectInput	16
2.6.2 กราฟิก 3 มิติทั่วไป	17
2.6.2.1 โครงสร้างพื้นฐานของโมเดล3D	17
2.6.2.2ระบบพิกัด 3มิติ(3D Coordinate System)	17
2.6.2.3การทรานฟอร์ม (Transformation)	18
2.6.3เทคนิคของออบเจ็กท์โอเร็นเต็ดเพื่อการสร้างเกมเอนจิน	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ทฤษฎีประยุกต์ซีเควนท์ไดอะแกรมแบบบนลงล่าง(Top Down Design Syntax) 19
บทที่3 วิธีดำเนินการวิจัย 22
3.1 การรวบรวม ความต้องการ(Requirement)สำหรับออกแบบเอดดูเทรนเม้นท์ 22
3.1.1ความต้องการ(Requirement) 22
3.1.2 วิเคราะห์(Analysis) 23
3.1.3 ออกแบบจากโครงสร้างของเอเจนท์(Structure of Agent) 25
3.2เครื่องมือที่ใช้ในการพัฒนา 28
3.3 ขั้นตอนในการสร้างโครงงาน 29
3.3.1การสำรวจความต้องการ(Requirement Capture) 29
3.3.1.1 ทำการลิสต์ลักษณะของระบบ(Feature List) 29
3.3.1.2 โมเดลธุรกิจ(Business Model) 30
3.3.1.3 ความต้องการที่เห็นได้ชัดของระบบ 31
3.3.1.4 ความต้องการที่เห็นได้ไม่ชัดของระบบ 34
3.3.1.5 เครื่องมือต้นแบบ (Prototype) 35
3.3.2การวิเคราะห์(Analysis) 37
3.3.2.1 วิเคราะห์ยูสเคส(Analysis use case view) 37
3.3.2.2 วิเคราะห์สถาปัตย์กรรม(Architect Overview) 38
3.3.3 การออกแบบ(Design) 39
3.3.3.1 การออกแบบคลาส(class) 39
3.3.4 ปฏิบัติการ (Implementation)&ทดสอบ(test) 42
บทที่4 ผลการทดลอง 47
4.1 ด้านซอฟแวรเอนจิเนียริ่ง 47
4.2 ด้านผู้เล่นเอดูเทรนเมนท์ 53
บทที่5 สรุปผลการทดลอง 55
บทที่6 บรรณานุกรม (Bibliography) 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้าที่
รูปที่2.1 เอเจนท์แบบง่ายตอบสนองทันที(Simple Reflex Agent)	5
รูปที่2.2 เอเจนท์แบบจดจำตอบสนองทันที(Model-Based reflex Agent)	6
รูปที่2.3 เอเจนท์แบบค้นหาเป้าหมาย(Goal-based agents)	7
รูปที่2.4 เอเจนท์แบบค้นหาเป้าหมายที่พึงพอใจที่สุด(Utility-based agents)	8
รูปที่2.5 เอเจนท์แบบเรียนรู้(Learning Agent)	9
รูปที่2.6 รูปรอบของการพัฒนาตามหลักของ(Unified Process Cycle)	10
รูปที่2.7รูปเปรียบเทียบความก้าวหน้าต่อเวลาระหว่างแบบรวดเดียว กับหลักเพิ่มเติมด้วยการทำซ้ำ	10
รูปที่2.8 คนงานในเฟสการจับประเด็นความต้องการ(Requirement Captures)	11
รูปที่2.9 วิธีการแบ่งเวลาในเฟสการจับประเด็นความต้องการ(Requirement Captures)	11
รูปที่2.10 คนงานในเฟสการวิเคราะห์(Analysis)	12
รูปที่2.11 วิธีการแบ่งเวลาในเฟสการวิเคราะห์(Analysis)	12
รูปที่2.12 คนงานในการออกแบบ(Design)	13
รูปที่2.13 วิธีการแบ่งเวลาในเฟสการออกแบบ(Design)	13
รูปที่2.14 คนงานในเฟสปฏิบัติการ (Implementation)	14
รูปที่2.15 วิธีการแบ่งเวลาในเฟสปฏิบัติการ (Implementation)	14
รูปที่2.16 คนงานในเฟสการทดสอบ(Test)	15
รูปที่2.17 วิธีการแบ่งเวลาในเฟสการทดสอบ(Test)	15
รูปที่2.18 โพลีกอน2Dและโพลีกอน3D	17
รูปที่2.19 ระบบพิกัด 3มิติ(3D Coordinate System)	17
รูปที่2.20 รูปโครงสร้างเมทริกท์(Matrix)	18
รูปที่2.21 รูปองค์ประกอบพื้นฐานของซีเควนท์ไดอะแกรมแบบบนลงล่าง	20
รูปที่2.22 ภาพตัวหลักที่เราพิจารณา(Main)ทำการเรียกฟังก์ชันตัวเอง	21
รูปที่2.23 แสดงการใช้ไดอะแกรมสถานะ(state machine diagram)ร่วมกับรูปที่2.21	21
รูปที่3.1แผนที่ของด่านเอเจนท์แบบง่ายตอบสนองทันที	25
รูปที่3.2แผนที่ของด่านเอเจนท์แบบจดจำตอบสนองทันที	26
รูปที่3.3แผนที่ของด่านเอเจนท์แบบค้นหาเป้าหมาย	26
รูปที่3.4เอเจนท์แบบค้นหาเป้าหมายที่พึงพอใจที่สุด	27
รูปที่3.5ระบบโหลดระบบเซฟ	30
รูปที่3.6ระบบเล่น	30
รูปที่3.7 Main Use Case Diagram	31
รูปที่3.8 Camera Control	31
รูปที่3.9 Control with Character Use case diagram	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่3.10Game Rules Use Case Diagram 33
รูปที่3.11 ShowText&play Menu 34
รูปที่3.12ต้นแบบเครื่องมือติดต่อกับผู้ใช้เขียนโปรแกรม(GUI Prototype) 35
รูปที่3.13ต้นแบบGUIก่อนเข้าเกม 35
รูปที่3.14ต้นแบบGUIขณะเล่นเกม 36
รูปที่3.15เกมต้นแบบ 36
รูปที่3.16Use case choose Menu 37
รูปที่3.17Architect descriptions & Glossary 38
รูปที่3.18Main Diagram 39
รูปที่3.19CGame System 40
รูปที่3.20 CGameEngine 41
รูปที่3.21 CGameEngine3 (GamePlay)1(CharLength) 42
รูปที่3.22 CGameEngine3 (GamePlay)2(AddCharacter) 43
รูปที่3.23 CGameEngine3 (GamePlay)4(SetInformation) 44
รูปที่3.24 CGameEngine3 (GamePlay)5(DeleteCharacter) 45
รูปที่3.25 CGameEngine3 (GamePlay)6(GetOrder) 46
รูปที่4.1 Gameengine.__init__.py 48
รูปที่4.2 FileManager.py 49
รูปที่4.3 ProcessInformation.py 50
รูปที่4.4 GameRules 51
รูปที่4.5 AIInfomation.py 51
รูปที่4.6 ScenceInformation.py 52
รูปที่4.7 ส่วนของGameEngineที่เป็น C++ ที่ทำหน้าที่ติดต่อกับGameSystemแบบโลจิคอล 52
รูปที่4.8 คลาสAIที่จะเขียนลงไฟล์ AI1.py , AI2.py 53
รูปที่4.9 คลาสไลบรารี่ที่ผู้ใช้ต้องติดต่อด้วย 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่2.1 ขั้นตอนการเติบโต(Stages of Growth) 4

ตารางที่3.1 ลิสต์ลักษณะของระบบ(Feature List) 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1

บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากปัจจุบันนี้ผลิตภัณฑ์เกมในหลายสายงานได้เติบโตไปมาก แต่ตลาดเกมทางด้านเอดดูเทรนเม้นท์เพื่องานโปรแกรมมิ่ง ยังเติบโตไม่ได้เท่าที่ควรเนื่องจากการกระบวนการทางซอฟแวร์เอนจิเนียริ่งยังไม่เหมาะสมและการใช้ภาษาสคริป(script)เพื่อเอดดูเทรนเม้นท์ทางด้านนี้ยังมีข้อจำกัดสูงมาก เช่น 1 ต้องใช้ซินแทค(syntax)ที่ตายตัวมากจนต้องเจนเนอเรท(generate)มาจากจียูไอ(G.U.I.)โดยเฉพาะเพื่อป้องกันความผิดพลาดจากผู้ใช้ 2 ทำการรีเคอซีฟ(recursive)บนภาษาสคริป ไม่ได้ 3 ไม่สามารถเรียกใช้งานฟังก์ชัน(function)ได้ ซึ่งการทำแบบนี้มีทั้งข้อดีและเสียคือ
(หมายเหตุ *จากคำว่า"ภาษาสคริปที่มีข้อจำกัด" ภาษาสคริปดังที่กล่าวไว้ไม่ได้หมายถึงภาษาที่สร้างเป็นโครงสร้างพิเศษ เช่น ภาษาจำพวกรูเบส(rules base)ที่เป็นภาษาพิเศษที่ใช้กับงานเอไอ แต่หมายถึงไฮเลเวลแลงเกวจ(High Level Language)ธรรมดาที่มีโครงสร้างคล้ายC++หรือPASCAL)

1ข้อดีข้อเสียของเอนจินที่ใช้ภาษาสคริปแบบมีข้อจำกัดสูง

ข้อดีต่อผู้ใช้

- ทำให้ความผิดพลาดเกิดขึ้นน้อย
-ตรวจสอบบั๊ก(bug)ได้ง่าย

ข้อเสียต่อuser

- ขาดความคิดสร้างสรรค์ในการประดิษฐ์หรือพลิกแพลง เพราะความจำกัดของภาษาสคริป
-ทำให้ผู้ใช้ไม่เข้าใจถึงการนำไปใช้งานจริง หรือโครงสร้างจริงของตัวเอนจินภายใน

ข้อเสียต่อระบบ

-ทำให้ระบบพัฒนาต่อได้ยากเพราะความจำกัดของภาษาสคริป

ในโครงงานนี้ได้ทำการวิจัยโดยนำภาษาสคริประดับเวรี ไฮเลเวล แลงเกวจ(Very High Level Language(VHLL))ซึ่งก็คือPythonมาพัฒนาเพื่อตัดข้อเสียของภาษาสคริปแบบ ไฮเลเวล แลงเกวจ(HLL)ที่มีข้อจำกัดสูงและเนื่องด้วยคุณสมบัติที่น่าสนใจในการซ่อน(embedding)ภาษาต่างเข้าไปได้ ทำให้มีแนวโน้มจะembeddedกับภาษาที่มีโครงสร้างพิเศษอย่างรูเบส(rules base)เพื่อให้สามารถขยายต่อไปใช้งานเฉพาะด้านได้อีกด้วย ซึ่งการใช้VHLLมาช่วยทำให้เกิดการพัฒนาที่ดีขึ้นกว่าการใช้HLLที่ใช้ในวงการสร้างเกม เอนจินธรรมดาโดยมีข้อดีข้อเสียดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2ข้อดีข้อเสียของการใช้ภาษาสคริปไพทอน(Python)

ข้อดีต่อผู้ใช้

- ทำให้ความผิดพลาดเกิดขึ้นน้อย ถ้าเอนจินสร้างส่วนที่จะโหลดโค้ด ของผู้ใช้มาใช้งานได้ดีทำให้ตรวจสอบบั๊กได้ง่าย เพราะมีความสามารถในการป้องกันความผิดพลาด(handle exception)

- ส่งเสริมความคิดสร้างสรรค์ในการประดิษฐ์หรือพลิกแพลง เพราะความจำกัดของภาษาสคริปมีน้อย และยังมีความสามารถรีเคอรซีฟสูงมากทำให้พลิกแหลงได้มากกว่าHLLหลายเท่า

-ทำให้ผู้ใช้เข้าใจถึงการนำไปใช้งานจริง หรือโครงสร้างจริงของตัวเอนจินภายในหรือมองเห็นภาพรวมได้มากขึ้นซึ่งอันนี้ขึ้นกับเลือกระดับของการซ่อนโค้ด(embedding code)ไว้ในเอนจินด้วย (VHLLเลือกระดับที่จะซ่อนโค้ดได้ง่ายกว่า HLLที่ส่วนใหญ่มักคอมไพ(compile)กลายเป็น.exe ไฟล์เดียว ยกเว้นพวกที่มีคุณสมบัติเป็นอินเตอร์พีเตอร์(interpreter)อย่างLisp Prolog)

ข้อดีต่อระบบ

-ทำให้เกม เอนจินสามารถขยายต่อไปได้เรื่อย เพราะคุณสมบัติในการขยาย(extend)รวมถึงซินแทค์ที่ง่ายมากทำให้เพิ่มแอดทิบิวส์(attribute)โดยไม่ต้องแก้โค้ด(code)เก่าก็ได้

ข้อเสียต่อระบบ

-ต้องทำเอกสารอย่างเต็มรูปแบบ เพราะไพทอนมีซินแทค์ที่ยืดหยุ่นมากจนอาจทำให้ผู้พัฒนาคนอื่นอาจไม่เข้าใจรูปแบบของการส่งเวริค์โฟว์(work flow)ได้ ซึ่งผู้สร้างได้คิดซินแทค์ของซีเควนท์ไดอะแกรม(sequence diagram)ขึ้นมาใหม่จากการนำUMLของ Grady Booch ,Ivar Jacobson ,James Rumbaugh มาปรับปรุงแล้วเรียกมันใหม่ว่าซีเควนท์ไดอะแกรมแบบบนลงล่าง “Top Down Design Syntax”ซึ่งมีคุณสมบัติในการทำเอกสารได้ตั้งแต่ปานกลางไปจนถึงละเอียดทุกบรรทัดซึ่งแล้วแต่ความต้องการของผู้พัฒนาจะนำไปดัดแปลง

1.2ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1ศึกษาการนำทฤษฎีออปเจ็กท์โอเรนเตด(Object Oriented) มาใช้งานประยุกต์กับเกมเอนจิน

1.2.2ศึกษาการสร้างทฤษฎีเกมเอนจิน

1.2.3ศึกษาการนำทฤษฎีเอไอมาใช้งานประยุกต์กับเกมเอนจิน

1.2.4เพื่อการวิจัยและสร้างเอดดูเทรนเม้นท์ เกมแนวใหม่สู่ตลาดเกม

1.2.5เพื่อพัฒนาและปรับปรุงโมเดลทางซอฟท์แวร์เอนจิเนียริ่งให้มีความสามารถในการจัดเก็บเอกสารมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3สมมุติฐานของการศึกษา

เราสามารถนำแนวความคิดของซอฟท์แวร์เอนจิเนียริ่งเพื่อการพัฒนาโครงงานนี้ให้สำเร็จตามขั้นตอนดังนี้

1.3.1 การรวบรวมความต้องการ(Requirement)สำหรับออกแบบเอดดูเทรนเม้นท์

1.3.2 จัดโครงสร้างองค์กรตามหลักยูนิไฟซอฟท์แวร์โพรเสด(The Unified Software Development Process)

1.3.3 ทำการปฏิบัติตามหลักเบสแพ็คทิส(Best Practices of Software Engineering)

1.4ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

จากสมมุติฐานสามารถนำทฤษฎีมาประยุกต์ใช้ได้ดังนี้

1.4.1 การรวบรวมความต้องการสำหรับออกแบบเอดดูเทรนเม้นท์

-บทความช่วงอายุที่สัมพันธ์กับการเรียนรู้อ้างอิงโดย Renu Ahuja, Sugata Mitra, Rashmi Kumar and Monica Singh "Education Through Digital Entertainment" เพื่อออกแบบระดับความยากง่ายให้เหมาะสมกับผู้เล่น

-บทความด้านเอดดูเทรนเม้นท์โดย J.Ajdari ,I.Aulaskari ,J.brown "edutainment in educational environments" เพื่อการจัดแบ่งประเภทของเกมเอดดูเทรนเม้นท์

-ทฤษฎี โครงสร้างเอเจนท์(Structure of Agent)(สิ่งที่จะสอน)อ้างอิงโดย Artificial Intelligence A Modern Approachของ Stuart Russell & Peter Norvig

1.4.2 จัดโครงสร้างองค์กรตามหลักยูนิไฟซอฟท์แวร์โพรเสด

-ทฤษฎีเบสแพ็คทิสออฟซอฟท์แวร์เอนจิเนียริ่ง

1.4.3 ทำการปฏิบัติตามหลักเบสแพ็คทิส

-ทฤษฎีเบสแพ็คทิสออฟซอฟท์แวร์เอนจิเนียริ่ง

-ทฤษฎีเกมเอนจิน

-ทฤษฎีประยุกต์ซีเควนท์ไดอะแกรมแบบบนลงล่าง(Top Down Design Syntax)

1.5ขอบเขตของงานวิจัย

โครงงานนี้ถูกสร้างขึ้นในลักษณะของโปรแกรมต้นแบบ(Prototype)ของเกม เอนจินที่จะนำไปใช้งานในเชิงพาณิชย์ได้จริงๆ ซึ่งโปรแกรมต้นแบบดังกล่าว ถูกสร้างในลักษณะของสถาปัตยกรรม(architecture) แบบลำดับชั้น(layer) โดยใช้ภาษาpython,C++,Direct-x9.0เป็นภาษาต้นแบบ อย่างไรก็ดีเนื่องจากงบประมาณทางซอฟท์แวร์เอนจิเนียริ่งค่อนข้างจำกัดมาก จึงอาจทำการดัดระบบดีบั๊ก (debug)และรายละเอียดของเกมพาณิชย์จริงๆ หลายอย่างออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2

วรรณกรรมหรืองานวิจัยที่เกี่ยวข้อง

2.1<u>บทความช่วงอายุที่สัมพันธ์กับการเรียนรู้อ้างอิงโดย</u> Renu Ahuja, Sugata Mitra, Rashmi Kumar and Monica Singh "Education Through Digital Entertainment" เพื่อออกแบบระดับความยากง่ายให้เหมาะสมกับผู้เล่น

Age	Grade	Piaget	Bruner	Kohlberg
		(cognitive)	**(cognitive)**	**(moral)**
1		การเคลื่อนไหว		
2.		การแสดงอารมณ์	การจดจำ	
3,4,5	ระดับอนุบาล	การติดต่อพูดคุยและการแสดงแนวคิดเบื้องต้นและเริ่มมีภาวะความคิดในใจเกิดขึ้น	เรียนในเชิงรูปภาพ	Level 1 - ศีลธรรมเบื้องต้น Type 1 การลงโทษเบื้องต้น & การเชื่อฟัง
6,7,8,9,10	ระดับประถม	การติดต่อพูดคุยรู้เรื่องมากขึ้น	---do-----	----do-----
11,12,13	ระดับมัธยมส์ต้น	การติดต่อพูดคุยอย่างเป็นทางการมีความคิดเชิงนามธรรม	เรียนรู้สัญลักษณ์และเครื่องหมาย	Level II มีระเบียบแบบแผน Type III มีการเข้าสังคม

ตารางที่2.1 ขั้นตอนการเติบโต(Stages of Growth)

2.2<u>บทความด้านการจัดแบ่งประเภทของเกมเอดดูเทรนเม้นท์</u>โดย J.Ajdari ,I.Aulaskari ,J.brown "Edutainment in educational environments"

1 สื่อบันเทิงเพื่อการศึกษา**(Educational Entertainment)**หรือที่เรียกว่าเล่นและเรียนรู้(Play & Learn) ซึ่งต้องพิจรณาใน 2 ประเด็น

1.1 ความบันเทิงเพื่อบุคคล(Entertainment for the individual) ต้องดูช่วงอายุหรือลักษณะของบุคคลที่เป็นเป้าหมายเพื่อที่สร้างโปรแกรมขึ้นมาเพื่อให้ความรู้

1.2 ความเข้มข้นของความบันเทิง(Varying degrees of entertainment) การที่จะสร้างโปรแกรมเพื่อการเรียนรู้ประเภทนี้ได้ควรจะมี เนื้อเรื่อง(Story Line)ที่เหมาะสมกับการสอดแทรกความรู้เพื่อสร้างความรู้สึกสมจริงที่จะนำความรู้นั้นไปใช้ในเกมและให้ความบันเทิงแก่ผู้เรียนรู้

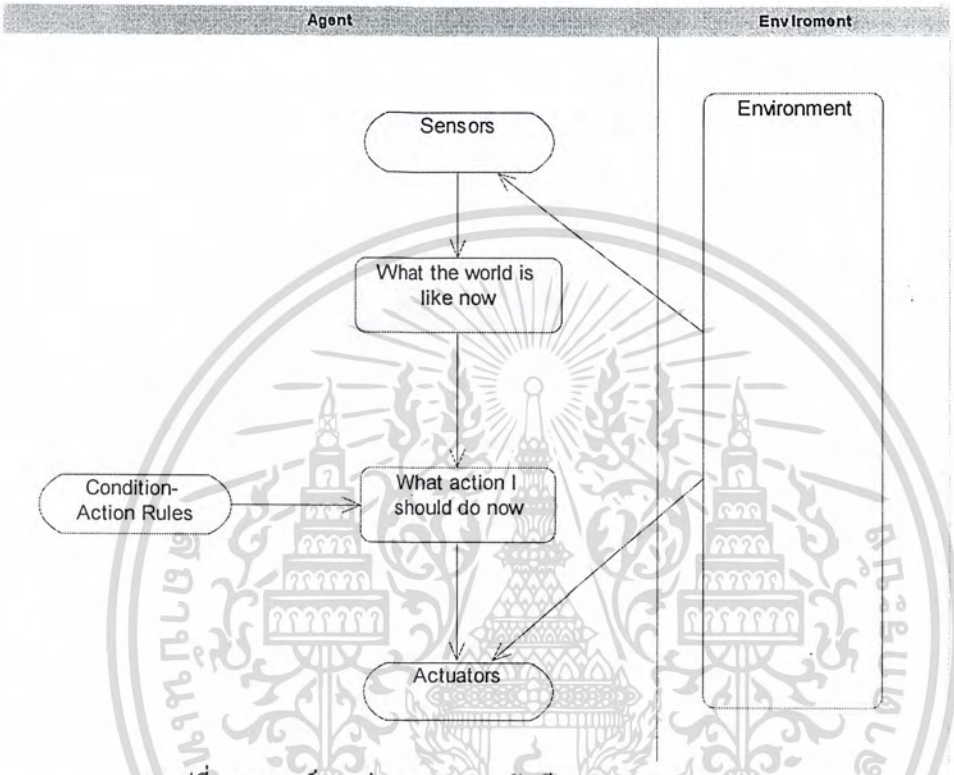
2 การศึกษาบันเทิงEntertaining Education การศึกษาโดยใช้สื่อบันเทิง (ใช้สอนเพื่อปฏิบัติจริงในเชิง ซี เอ ไอ CAI(Computer-Assisted Instruction))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3<u>ทฤษฎีโครงสร้างเอเจนท์(Structure of Agent)</u>(สิ่งที่จะสอน)อ้างอิงโดย

Artificial Intelligence A Modern Approachของ Stuart Russell & Peter Norvig

1 เอเจนท์แบบง่ายตอบสนองทันที(Simple Reflex Agent) :เป็นเอเจนท์ที่มีเฉพาะความสามารถในการรับรู้ ไม่มีความสามารถในการจดจำได้และไม่มีความสามารถนำในการคิดหาข้อมูลย้อนหลังย้อนหน้าได้ (backward forward chaining)



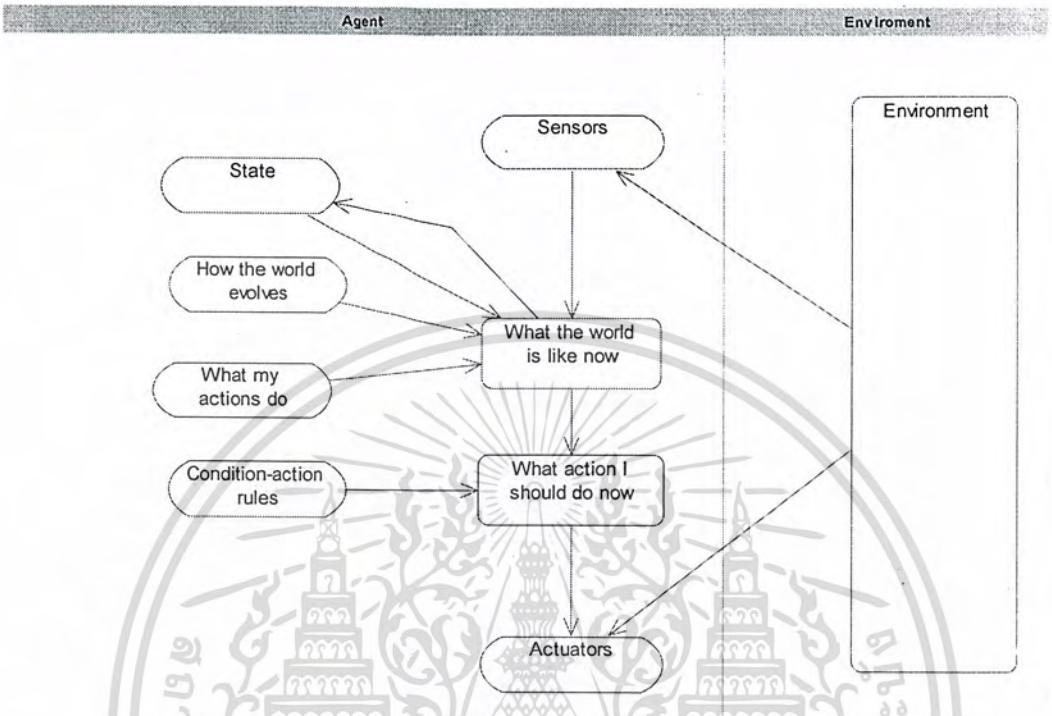
รูปที่2.1เอเจนท์แบบง่ายตอบสนองทันที(Simple Reflex Agent)

โดย 1 เอเจนท์ประเภทนี้จะมีเซนเซอร์(Sensors)ของตนเองใช้เพื่อการรับรู้(percept) โดยจับการเปลี่ยนแปลงจากภายนอกเพื่อคาดเดาสภาพของสิ่งแวดล้อม(Environment)

2 จากนั้นนำสภาพของสิ่งแวดล้อม(Environment)ปัจจุบันที่ได้จากการรับรู้(percept) มาเทียบกับกฎเพื่อดูว่าต้องทำ(Action)อะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 เอเจนท์แบบจดจำตอบสนองทันที(Model-Based reflex Agent):เป็นเอเจนท์ที่มีเฉพาะความสามารถในการรับรู้และจดจำได้แต่ว่าไม่มีความสามารถนำในการคิดหาข้อมูลย้อนหลังย้อนหน้าได้(backward forward chaining) เช่นกัน



รูปที่2.2 เอเจนท์แบบจดจำตอบสนองทันที(Model-Based reflex Agent)

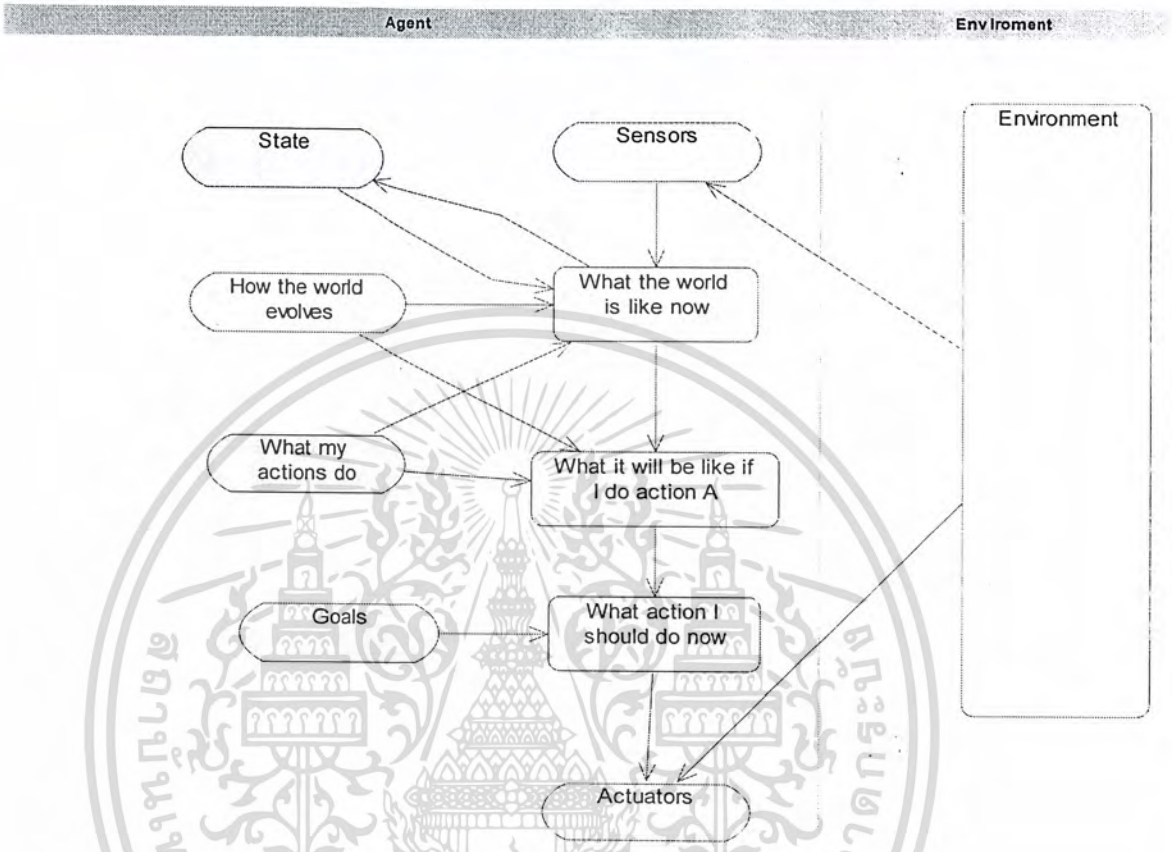
โดย 1 เอเจนท์ประเภทนี้จะมีเซนเซอร์(Sensors)ของตนเองใช้เพื่อการรับรู้(percept)คู่กับ การกระทำ(action)ที่ตนได้กระทำไว้เพื่อจับการเปลี่ยนแปลงจากภายนอกเพื่อคาดเดาสภาพของสิ่งแวดล้อม(Environment)

2 จากนั้นนำสภาพของสิ่งแวดล้อม(Environment)ที่ได้รับการจำลองขึ้นมาก่อนหน้านั้น มาเทียบกับกฎเพื่อดูว่าต้องทำ(Action)อะไร

3 แล้วจึงนำมาเทียบกับกฎเพื่อดูว่าต้องทำ(Action)อะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 เอเจนท์แบบค้นหาเป้าหมาย(Goal-based agents):เป็นเอเจนท์ที่มีความสามารถในการรับรู้ จดจำได้ และความสามารถในการคิดหาข้อมูลย้อนหลังย้อนหน้า(มีbackward forward chaining)แต่ใช้กับปัญหาที่มีเพียงคำตอบเดียว



รูปที่2.3 เอเจนท์แบบค้นหาเป้าหมาย(Goal-based agents)

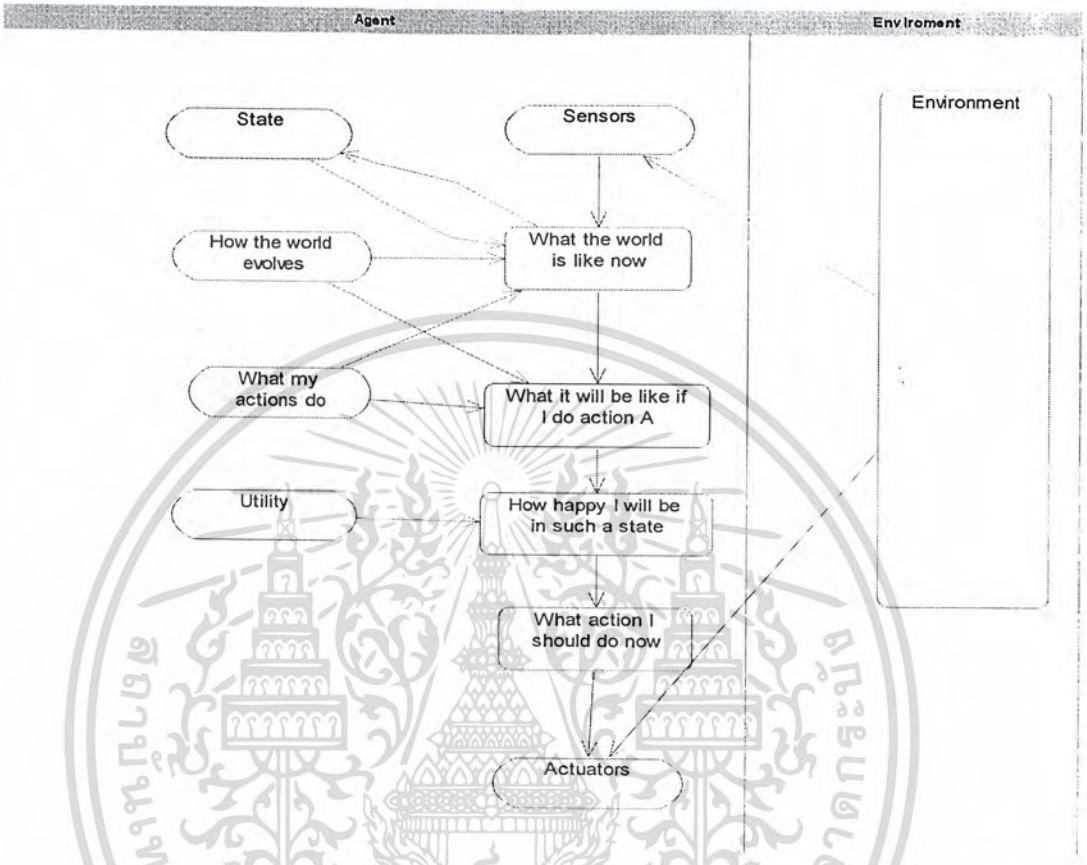
โดย 1 เอเจนท์ประเภทนี้จะมีเซนเซอร์(Sensors)ของตนเองใช้เพื่อการรับรู้(percept)คู่กับ การกระทำ(action)ที่ตนได้กระทำไว้เพื่อจับการเปลี่ยนแปลงจากภายนอกเพื่อคาดเดาสภาพของสิ่งแวดล้อม (Environment)

2 จากนั้นนำสภาพของสิ่งแวดล้อม(Environment)ที่ได้รับการจำลองขึ้นมาก่อนหน้านั้น มาเทียบกับกฎเพื่อทำการรวบรวมข้อมูลวางแผน(Planning)ที่จะทำให้ได้เป้าหมายมา

3 เมื่อได้ข้อมูลเพียงพอแล้วจึงนำมาเทียบกับกฎเพื่อดูว่าต้องทำ(Action)อะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 เอเจนท์แบบค้นหาเป้าหมายที่พึงพอใจที่สุด(Utility-based agents):เป็นเอเจนท์ที่มีความสามารถในการรับรู้ จดจำได้ และความสามารถในการคิดคำนวณได้(มีbackward forward chaining)สามารถเลือกคำตอบจากปัญหาที่มีหลายคำตอบได้ตามค่าฮิลรูริสติก(heuristic)



รูปที่2.4 เอเจนท์แบบค้นหาเป้าหมายที่พึงพอใจที่สุด(Utility-based agents)

โดย 1 เอเจนท์ประเภทนี้จะมีเซนเซอร์(Sensors)ของตนเองใช้เพื่อการรับรู้(percept)คู่กับ การกระทำ(action)ที่ตนได้กระทำไว้เพื่อจับการเปลี่ยนแปลงจากภายนอกเพื่อคาดเดาสภาพของสิ่งแวดล้อม(Environment)

2 จากนั้นนำสภาพของสิ่งแวดล้อม(Environment)ที่ได้รับการจำลองขึ้นมาก่อนหน้านั้น มาเทียบกับกฎเพื่อทำการรวบรวมข้อมูลวางแผน(Planning)ที่จะทำให้ได้เป้าหมาย(Goal)มา

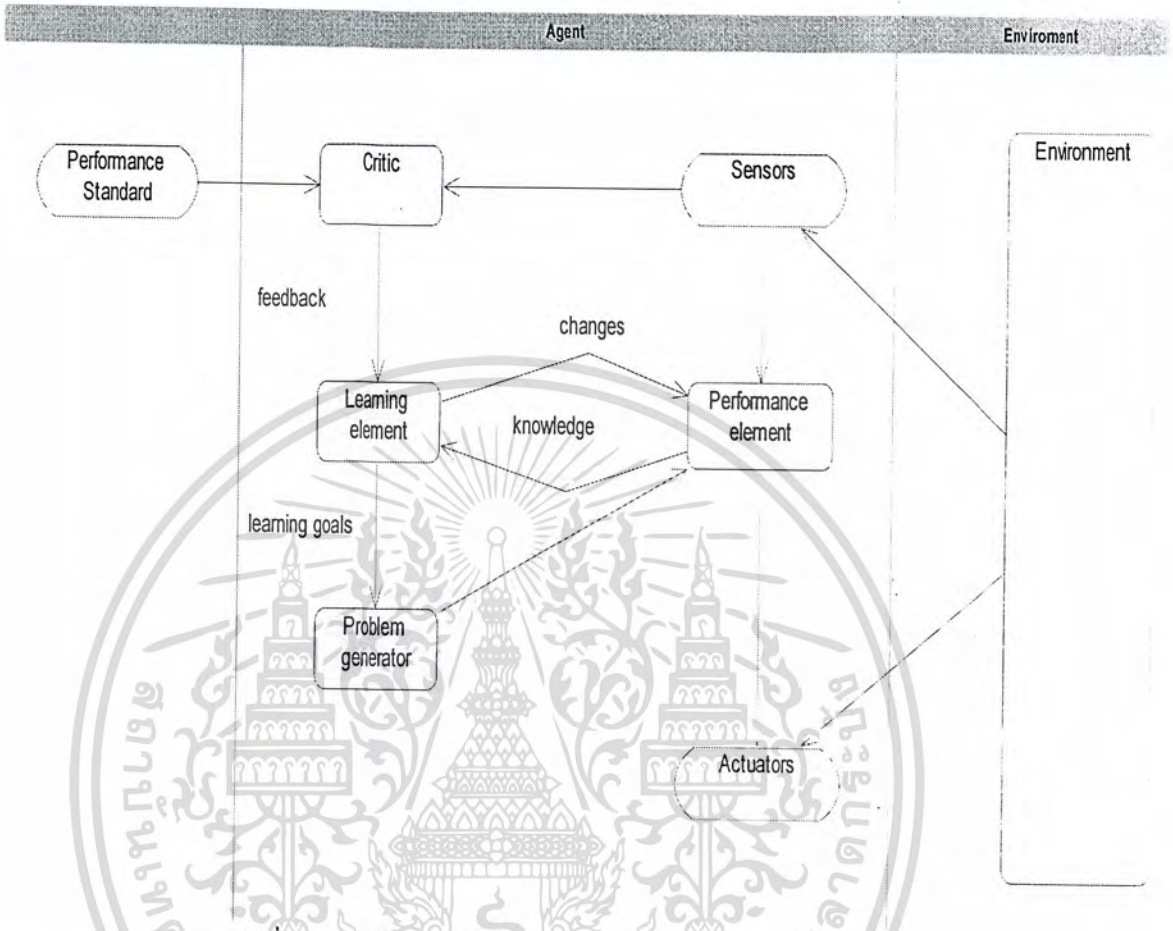
3 เมื่อได้ข้อมูลเพียงพอแล้วจึงนำมาเทียบกับกฎเพื่อดูว่าต้องทำ(Action)อะไร

4 ดูว่าเป้าหมายที่เท่าไรทำให้เกิดพึงพอใจที่สุด(Utility)ตามหลักheuristic

5 เมื่อได้ข้อมูลเพียงพอแล้วจึงนำมาเทียบกับกฎเพื่อดูว่าต้องทำ(Action)อะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 เอเจนท์แบบเรียนรู้(Learning Agent):เป็นเอเจนท์ที่มีความสามารถในการทดลองหาข้อมูลที่คิดว่าดีที่สุดได้



รูปที่2.5 เอเจนท์แบบเรียนรู้(Learning Agent)

โดยเอเจนท์ประเภทนี้จะมีทั้งหมด 4 องค์ประกอบซึ่งนำมาใช้งานดังนี้

1 ส่วนของการเรียนรู้(learning element)ซึ่งจะต้องคอยตอบสนองต่อการปรับปรุงจากส่วนของการปรับปรุง(performance element)

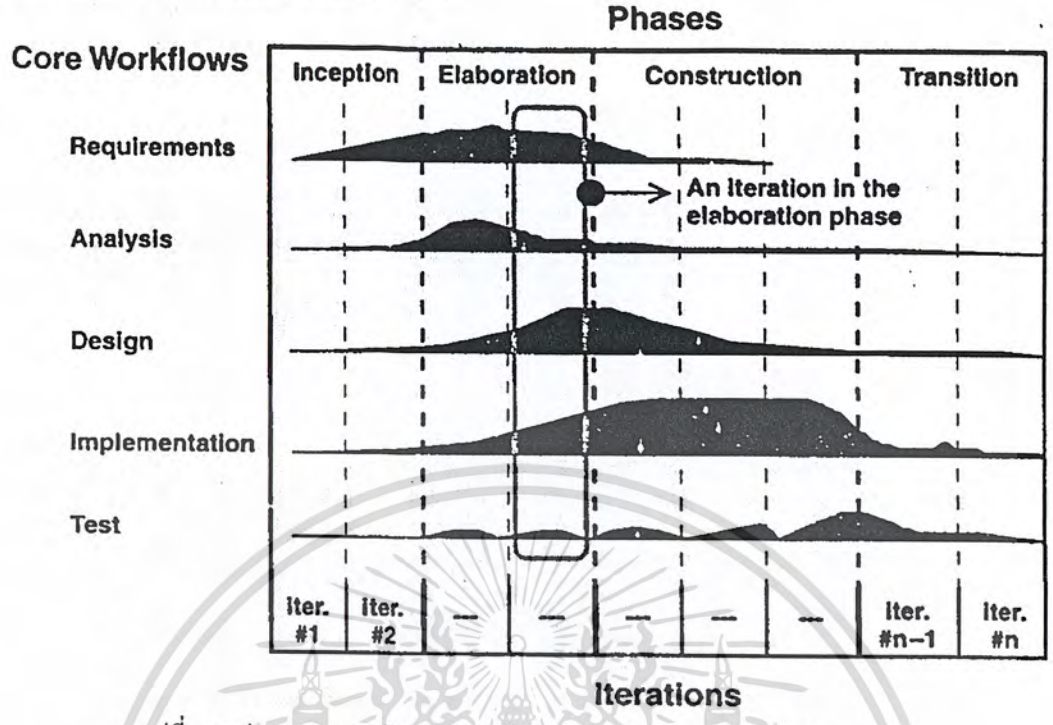
2 ส่วนของการปรับปรุง(performance element)จะเป็นตัวเลือกการกระทำ(action)ที่คิดว่าดีที่สุดในขณะนั้น

3 ส่วนของการวิจัย(Critic) จะเป็นตัวคอยส่งข้อมูลย้อนกลับ(feedback)กลับมาที่ส่วนของการเรียนรู้(learning element) ว่าเอเจนท์ควรทำหรือตัดสินใจอย่างไรให้มีการปรับปรุงส่วนของการปรับปรุง(performance element)ให้ดีขึ้นเพื่อในอนาคต

4 ส่วนของการวิเคราะห์ปัญหา(Problem generator) จะเป็นตัวตอบสนองต่อการแนะนำการกระทำ(action)ที่จะนำไปสู่การเกิดประสบการณ์ใหม่

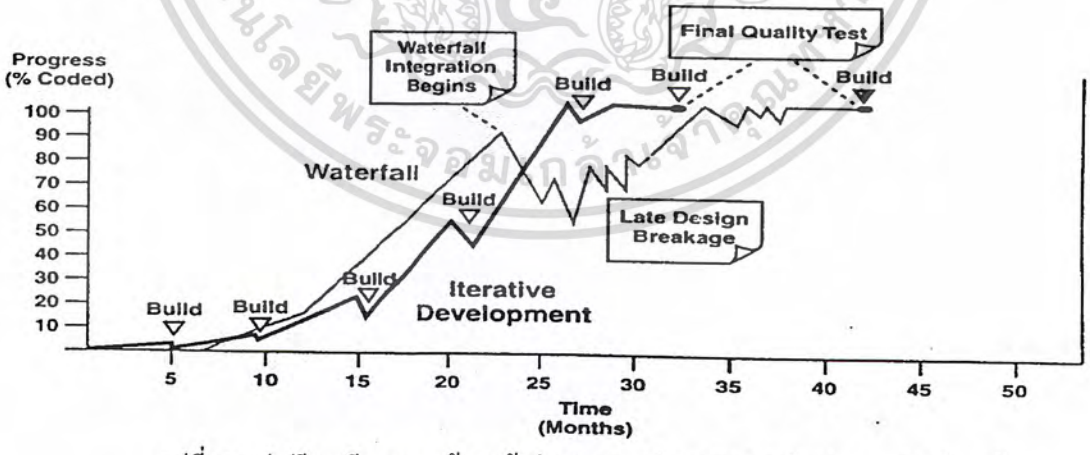
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 <u>ทฤษฎีเบสแพ็คทิสออฟซอฟท์แวร์เอนจิเนียริ่ง</u>



รูปที่2.6 รูปรอบของการพัฒนาตามหลักของ(Unified Process Cycle)

เป็นทฤษฎีการสร้างโครงงานขนาดใหญ่จะทำโดยการแบ่งขั้นตอนของซอฟท์แวร์เอนจิเนียริ่ง ออกเป็นช่วงๆแล้วใช้วิธีการค่อยๆเพิ่มขยาย(iterative and Incremental)ทุกขั้นตอนทางซอฟท์แวร์เอนจิเนียริ่งปกติ(อยู่ในส่วนของเวริค์โฟร์หลัก(core workflow))จนกระทั่งสร้างซอฟท์แวร์ได้เสร็จสมบรูณ์ ซึ่งจะช่วยให้ได้ผลลัพธ์ที่ดีกว่าการทำซอฟท์แวร์เอนจิเนียริ่ง แบบธรรมดาดังกราฟต่อไปนี้



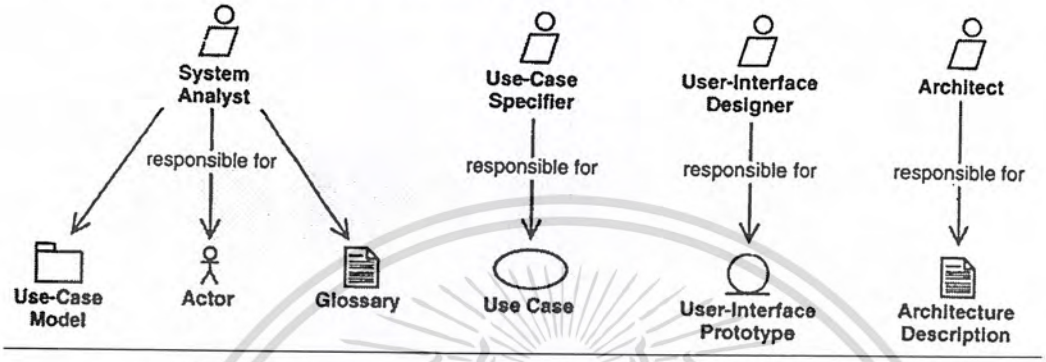
รูปที่2.7 รูปเปรียบเทียบความก้าวหน้าต่อเวลาระหว่างแบบรวดเดียว(water fall)กับหลักเพิ่มเติมด้วยการทำซ้ำ(iterative and incremental)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งนี้เนื่องจากมีกระบวนการตรวจสอบผลลัพธ์ที่ได้ตลอดเวลาทำให้ความผิดพลาดเกิดได้น้อย

2.5 โครงสร้างองค์กรตามหลักยูนิไฟซอฟท์แวร์โพรเสด จากการประยุกต์รวมกับเบสแพ็คทิสออฟซอฟท์แวร์เอนจิเนียริ่ง ได้นำมาใช้แบ่งหน้าที่คนได้ดังนี้

2.5.1 เฟสการจับประเด็นความต้องการ(Requirement Captures)



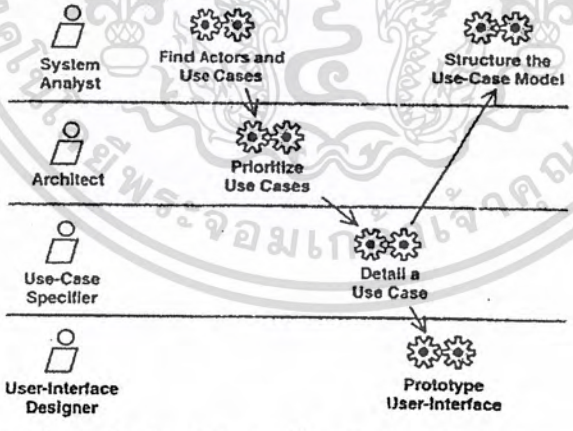
รูปที่2.8คนงานในเฟสการจับประเด็นความต้องการ(Requirement Captures)

นักวิเคราะห์ระบบ (System Analyst) ทำการวิเคราะห์ความเสี่ยงและทำการลิสต์ลักษณะของระบบ (Feature List) , สร้างโมเดลธุรกิจ(Business Model)

นักสถาปัตย์กรรม(Architecture)จัดลำดับuse case

ผู้ช่วยเจาะจงยูสเคส(Use-case Specifier)ทำการระบุรายละเอียดยูสเคส(Use-case)

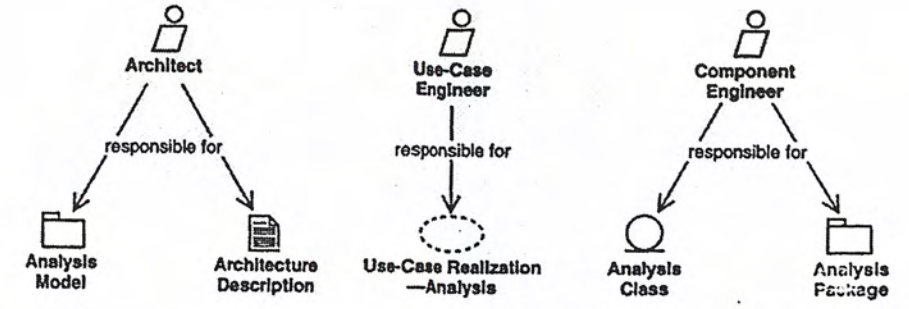
คนออกแบบระบบ(user-Interface Designer)ติดต่อกับผู้ใช้



รูปที่2.9 วิธีการแบ่งเวลาในเฟสการจับประเด็นความต้องการ(Requirement Captures)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 เฟสการวิเคราะห์(Analysis)

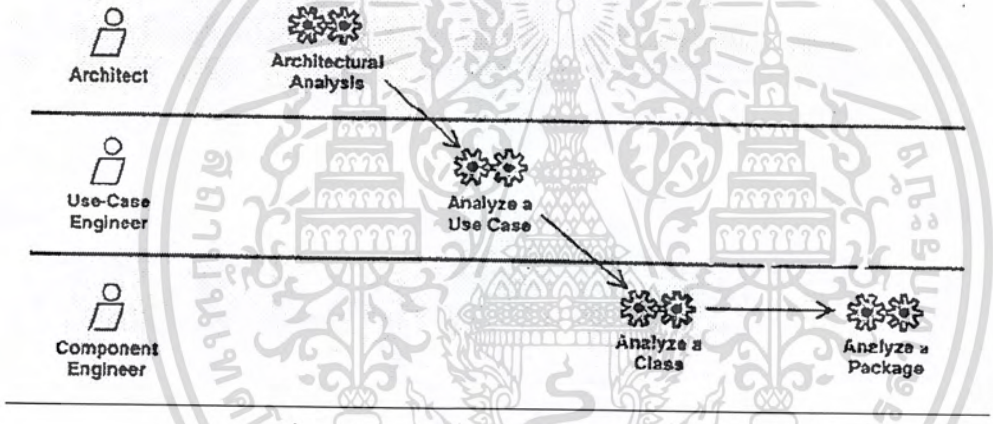


รูปที่2.10 คนงานในเฟสการวิเคราะห์

นักสถาปัตย์กรรม(Architecture) วิเคราะห์สถาปัตย์กรรม

ยูสเคสเอนจิเนีย(Use-case Engineer)วิเคราะห์ยูสเคส

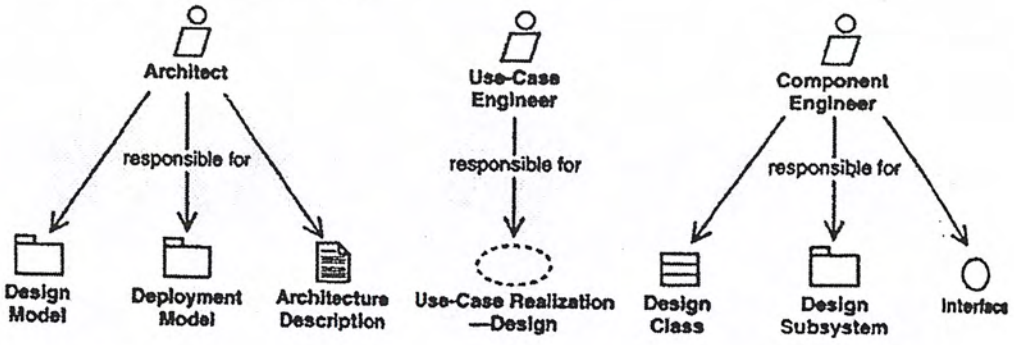
คอมโพเนนท์เอนจิเนีย(Component Engineer) วิเคราะคลาส



รูปที่2.11วิธีการแบ่งเวลาในเฟสการวิเคราะห์(Analysis)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 เฟสการออกแบบ(Design)

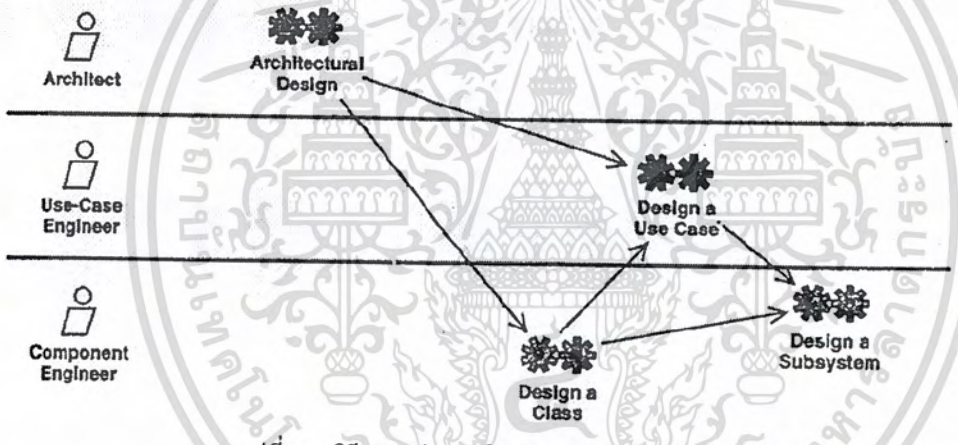


รูปที่2.12 คนงานในการออกแบบ(Design)

นักสถาปัตย์กรรม(Architecture) ออกแบบสถาปัตย์กรรมและจัดหาฮาร์ดแวร์(hardware)

ยูสเคสเอนจิเนีย(Use-case Engineer)วิเคราะห์ยูสเคส

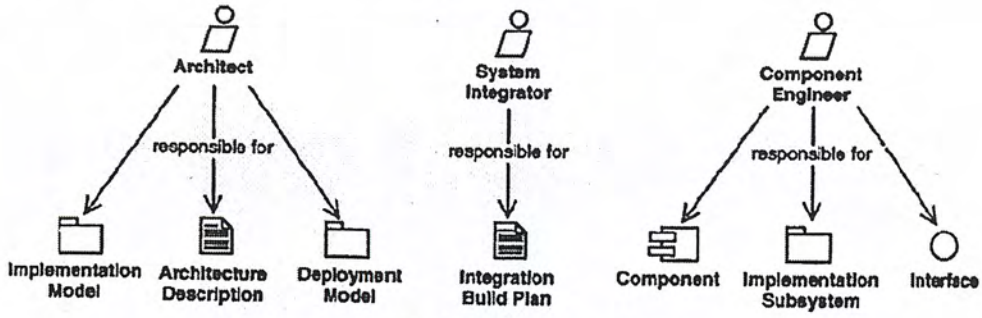
คอมโพเนนท์เอนจิเนีย(Component Engineer) ออกแบบคลาส



รูปที่2.13 วิธีการแบ่งเวลาในเฟสการออกแบบ(Design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 เฟสปฏิบัติการ (Implementation)

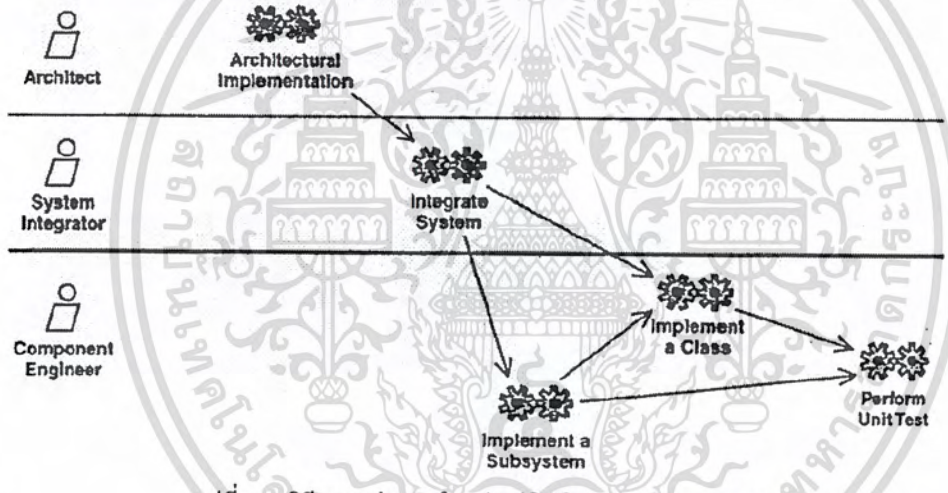


รูปที่2.14 คนงานในเฟสปฏิบัติการ (Implementation)

นักสถาปัตย์กรรม(Architecture) ออกแบบสถาปัตย์กรรมและจัดหาฮาร์ดแวร์(hardware)

ผู้รวบรวมระบบ(System Integrator)ทำการรวบรวมแผนการ

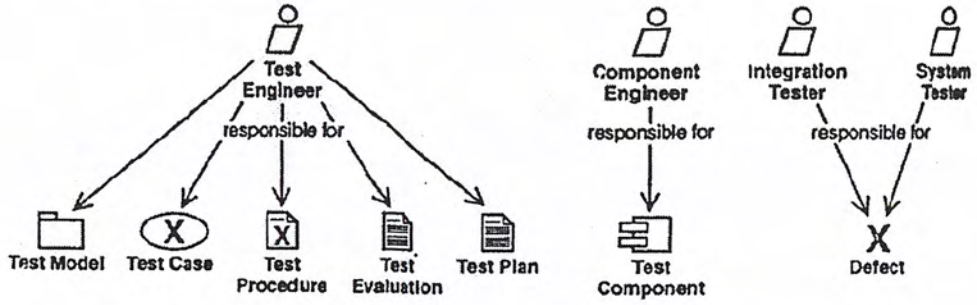
คอมโพเนนท์เอนจิเนีย(Component Engineer) ออกแบบคอมโพเนนท์



รูปที่2.15 วิธีการแบ่งเวลาในเฟสปฏิบัติการ (Implementation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.5 เฟสการทดสอบ(Test)



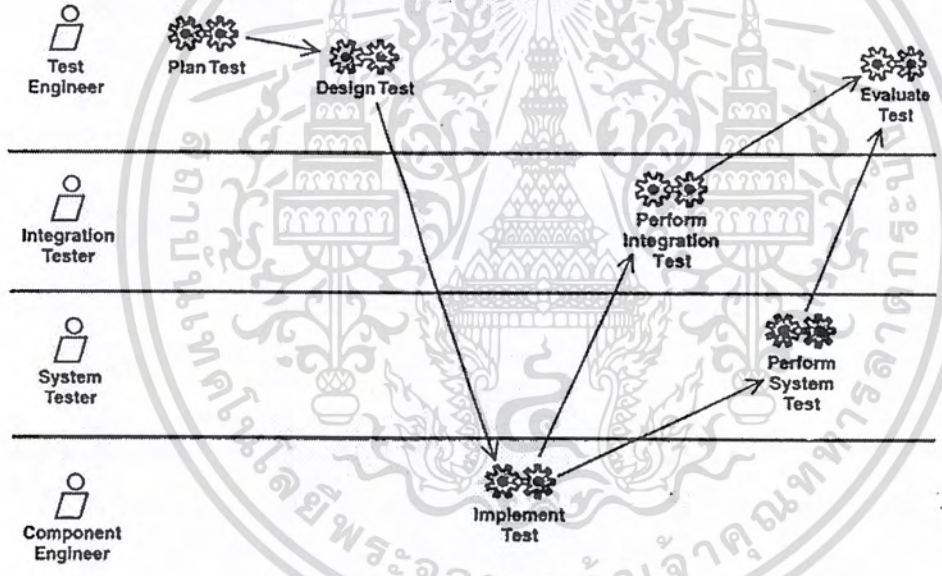
รูปที่2.16 คนงานในเฟสการทดสอบ(Test)

เอนจิเนียทดสอบ(Test Engineer) ออกแบบการทดสอบ

คอมโพเนนท์เอนจิเนีย(Component Engineer) ทดสอบคอมโพเนนท์

ผู้ทดสอบระบบ(System Tester)ตอบสนองข้อบกพร่อง

ผู้รวบรวมการทดสอบ(Integratortion Tester)ตอบสนองข้อบกพร่อง



รูปที่2.17 วิธีการแบ่งเวลาในเฟสการทดสอบ(Test)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6ทฤษฎีเกมเอนจิน

สำหรับทฤษฎีเกมเอนจิน ควรต้องมีความรู้เกี่ยวกับDirect-xหรือOpen-GLเป็นอย่างน้อย(หากใช้งานบนwindow)และหลักการของกราฟิก 3 มิติรวมไปถึงเทคนิคของออบเจกท์โอเร็นเต็ด(object-oriented)เพื่อการสร้างเกม เอนจินขึ้น โดยจะกล่าวไว้พอสังเขป ดังนี้

2.6.1 **Direct-X**

เป็น API ของไมโครซอฟต์ที่พัฒนาขึ้นเพื่อใช้เป็นอินเตอร์เฟซสำหรับควบคุมฮาร์ดแวร์มัลติมีเดียบนระบบ Microsoft Windows โดยAPI ของDirectX ดังกล่าวสร้างขึ้นมาบนรากฐานของ HAL (Hardware Abstraction Layer) สามารถซ่อนลักษณะเฉพาะของดีไวซ์ (Device) ที่เกี่ยวพันอยู่กับฮาร์ดแวร์ และเนื่องจาก DirectX ได้รับการออกแบบให้มีความสามารถในการเสริมขยายได้ในอนาคต ดังนั้นมันจึงรองรับความสามารถของฮาร์ดแวร์เร่งความเร็วต่างๆมากมายที่ปัจจุบันยังไม่มี ด้วยความสามารถจำลองการทำงานผ่าน HEL (Hardware Emulation Layer) ซึ่งจากหลักการดังกล่าวจะเห็นได้ชัดว่าเราควรทำเกมเอนจินในลักษณะของarchitectureแบบLayer
โดยชุด API ที่มีอยู่ใน DirectX ประกอบด้วย DirectDraw, Direct3D, DirectMusic, DirectSound, DirectPlay, DirectInput และ DirectSetup

แต่ในที่นี้ข้าพเจ้าจะขอกล่าวเฉพาะบางส่วนที่นำมาใช้ในโครงงานเท่านั้น

2.6.1.1 **Direct3D** ในยุคแรกมี API อยู่สองโหมดคือ โหมด Immediate (IM) และโหมด Retained (RM) ในโหมด IM เป็นโหมดที่ใช้งานยากแต่มีความยืดหยุ่นสูง เป็น API ในระดับล่างสำหรับใช้เขียนเกมที่ทำงานได้เร็วและมีประสิทธิภาพสูงกว่าโหมด RM ในโหมด RM เป็นโหมดที่สร้างขึ้นมาเป็นเลเยอร์ที่อยู่บนสุดของโหมด IM โดยโหมดนี้จะจัดเตรียมบริการต่างๆเช่น การจัดการ Texture, การโหลด Object File, การจัดลำดับเฟรม และการจัดทำออบเจ็กต์เคลื่อนไหว การศึกษาและใช้งานโหมด RM นั้นง่ายกว่าเมื่อเทียบกับโหมด IM แต่การทำงานในโหมด RM ได้หยุดลงใน DirectX เวอร์ชัน 6 และมุ่งพัฒนาการทำงานในโหมด IM ให้มีความสามารถและใช้งานง่ายในเวอร์ชันต่อมา

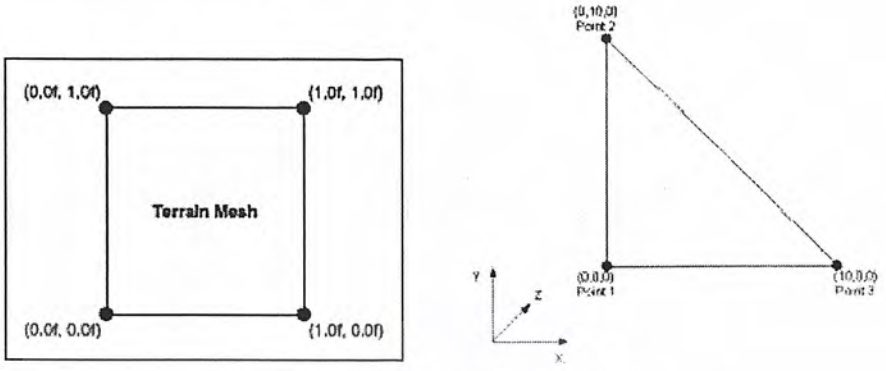
2.6.1.2 **DirectInput** เป็น API ของ DirectX ที่สนับสนุนอินพุตความเร็วต่ำ (Low-latency Input)ซึ่งเป็นอุปกรณ์อินพุตส่วนใหญ่ที่มีในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.6.2 กราฟิก 3 มิติทั่วไป

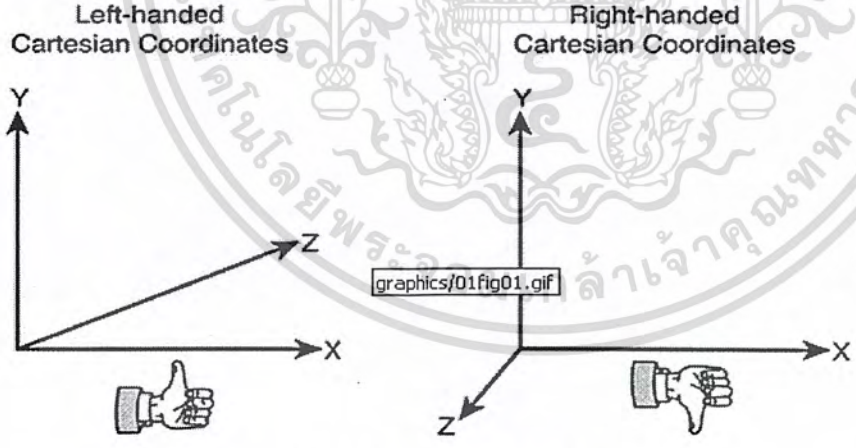
2.6.2.1 โครงสร้างพื้นฐานของโมเดล3D



รูปที่2.18 โพลีกอน2Dและโพลีกอน3D

โพลีกอน(Polygon) คือ รูปหลายเลี่ยมโดยมีรูปทรงสามเหลี่ยมเป็นพื้นฐานที่ต่ำที่สุดประกอบด้วย 3 จุด และจุดหรือ เวอร์เท็กซ์(Vertex) คือ ส่วนย่อยที่เล็กที่สุดในระบบ 3 มิติ ซึ่งก็คือจุดที่ตั้งอยู่ภายในสเปซ 2มิติหรือ 3มิติ เราสามารถสร้างเส้นได้โดยการรวมสองเวอร์เท็กซ์เข้าด้วยกัน เส้นสามเส้นจะรวมกันเป็นโพลีกอน ความสัมพันธ์ของจุดเวอร์เท็กซ์ที่สามารถวาดเส้นเชื่อมต่อกันเป็นรูปร่าง คือ พื้นผิวโพลีกอน (Polygon face)

2.6.2.2ระบบพิกัด 3มิติ(3D Coordinate System)



รูปที่2.19 ระบบพิกัด 3มิติ(3D Coordinate System)

ในdirect3Dจะใช้กฎหัวแม่มือซ้าย ซึ่งจะแตกต่างจากOpenGLซึ่งใช้กฎหัวแม่มือขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61475

2.6.2.3การทรานฟอร์ม (Transformation)คือการที่ต้องการเปลี่ยนแปลงเวอร์เท็กซ์ของวัตถุ โดยทรานสเลชั่น(translation)คือย้ายจุดกึ่งกลาง สเกล(scale)คือขยาย(หรือกระจายเวอร์เท็กซ์จากจุดกึ่งกลาง) โรเตด(rotate)คือหมุนเวอร์เท็กซ์รอบจุดกึ่งกลาง

3D translation transformation matrix

1	0	0	0
0	1	0	0
0	0	1	0
Tx	Ty	Tz	1

3D scaling transformation matrix

Sx	0	0	0
0	Sy	0	0
0	0	Sz	0
0	0	0	1

z-axis, 3D rotation transformation matrix

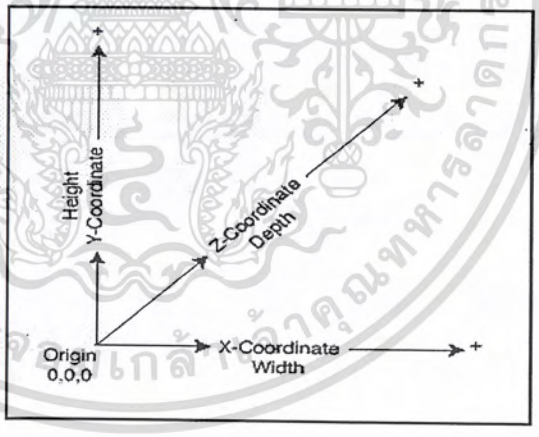
cos r	sin r	0	0
-sin r	cos r	0	0
0	0	1	0
0	0	0	1

x-axis, 3D rotation transformation matrix

1	0	0	0
0	cos r	sin r	0
0	-sin r	cos r	0
0	0	0	1

y-axis, 3D rotation transformation matrix

cos r	0	-sin r	0
0	1	0	0
sin r	0	cos r	0
0	0	0	1



รูปที่2.20 รูปโครงสร้างเมทริกท์(Matrix)

หากต้องการทำทุกอย่างให้คูณกันตามลำดับ สเกล โรเตดและทรานสเลชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3เทคนิคของออบเจ็กท์โอเร็นเต็ดเพื่อการสร้างเกมเอนจิน

อันได้แก่การทำFrustum ,เช็คการชน(Check collision) , โครงสร้างการเก็บข้อมูล(Data Structure) ,สร้างท้องฟ้า (Sky Box) ,ระบบโหลดและเซฟ นั้นไม่มีทฤษฎีตายตัวดังนั้นจึงขอยกไว้ไปกล่าวในขั้นตอนการดำเนินงาน

2.7 ทฤษฎีประยุกต์ซีเควนท์ไดอะแกรมแบบบนลงล่าง(Top Down Design Syntax)

ปัญหา 1 เนื่องจากเกมเอนจินค่อนข้างซับซ้อนมากจึงต้องสร้างสถาปัตย์กรรมแบบมีลักษณะเป็นลำดับชั้น(Layer architecture) การที่จะทำเอกสารจัดเก็บได้นั้นถ้ามีโครงสร้างที่ชัดเจนจะทำให้คนงานตามหลักทฤษฎีเบสแพ็คทีส สามารถทำงานได้ง่ายขึ้น

2 เนื่องจากไพทอนมีความสามารถที่ยืดหยุ่นมากจนอาจทำให้ผู้พัฒนาคนอื่นอาจไม่เข้าใจรูปแบบของการส่งข้อมูล(dataflow) ได้จึงต้องทำเอกสาร อย่างเต็มรูปแบบ

3 สำหรับการทำงานร่วมกับโปรแกรมเมอร์ โดยทั่วไปโปรแกรมเมอร์จะพิจารณาเฉพาะคลาสหรือออปเจกท์ที่เขาปฏิบัติงานเป็นหลักอยู่เท่านั้น แต่โดยมาก ในหลายๆบริษัท(อ้างอิงจาก GameGems3)ซอฟท์แวร์เอนจิเนียริ่งมักนำทุกชั้น(layer) มารวมกันทำให้โปรแกรมเมอร์ต้องพลิกดูclassประกอบทำให้เสียเวลามากในการทำความเข้าใจ

สมมุติฐาน

ไดอะแกรมแบบบนลงล่างนั้นเหมาะสมกับสถาปัตย์กรรมแบบที่มีลักษณะเป็นลำดับชั้น(Layer architecture)มากที่สุดและ เนื่องจากสถาปัตยกรรมของเกมเอนจินเป็นแบบลำดับชั้น ดังนั้นจึงได้ทำการทดลองทำเอกสารเพื่อให้แบ่งหน้าที่ในแต่ละชั้นได้อย่างชัดเจน

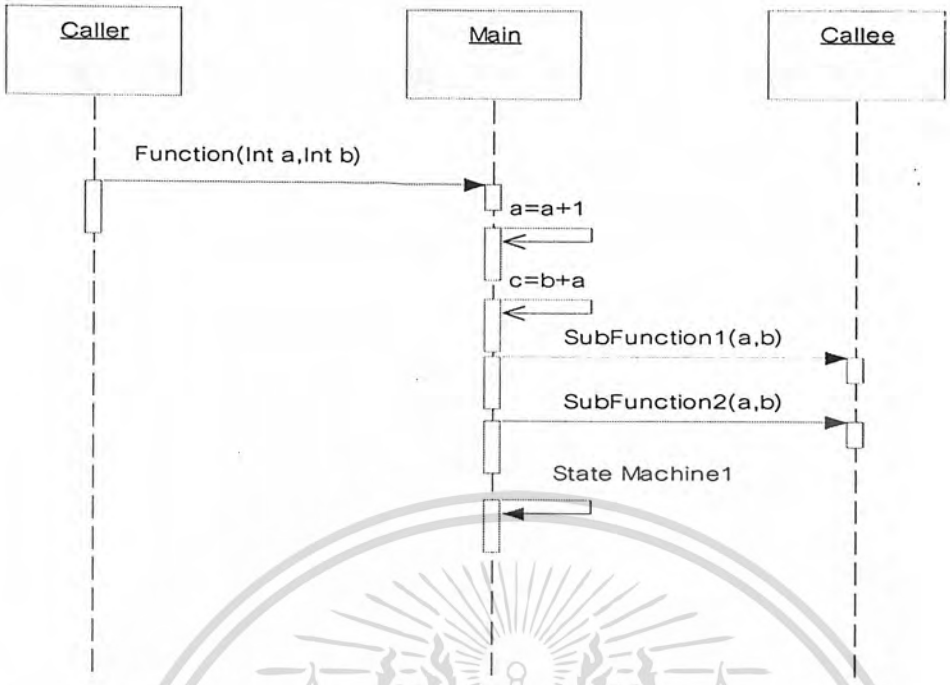
แนวทางแก้ปัญหา

ผู้สร้างได้คิดประยุกต์ซีเควนท์ไดอะแกรมขึ้นมาใหม่จากการนำUMLของ Grady Booch ,Ivar Jacobson ,James Rumbaugh มาปรับปรุงแล้วเรียกมันใหม่ว่าซีเควนท์ไดอะแกรมแบบบนลงล่าง(Top Down Design Syntax)ซึ่งมีคุณสมบัติในการทำเอกสาร ได้ตั้งแต่ปานกลางไปจนถึงละเอียดทุกบรรทัด** ซึ่งแล้วแต่ความต้องการของผู้พัฒนาจะนำไปดัดแปลง

แต่อย่างไรก็ดีซีเควนท์ไดอะแกรมแบบบนลงล่างยังไม่ได้มีการนำไปลองใช้กับเทคนิคหลายๆ อย่างบนไพทอนและภาษาอื่นๆทำยังไม่สามารถสรุปได้ ว่าสามารถบันทึกได้ทุกบรรทัดของทุกภาษาจริง

**หมายเหตุ คำว่า"ทุกบรรทัด"หมายถึง ความสามารถในการเก็บรายละเอียดของโค้ดดิ้งทุกบรรทัดโดยแปลงเป็นสิ่งที่เข้าใจง่ายอย่างไดอะแกรมสถานะ(state machine diagram)ซึ่งในไดอะแกรมแบบบนลงล่างสามารถเขียนให้ละเอียดได้แล้วค่อยนำมาทำโค้ดดิ้งอีกทีโดยไม่ต้องทำอะไรที่นอกเหนือกว่าในไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.21 รูปองค์ประกอบพื้นฐานของซีเควนท์ไดอะแกรมแบบบนลงล่างและการเรียกใช้งานไดอะแกรมสถานะ

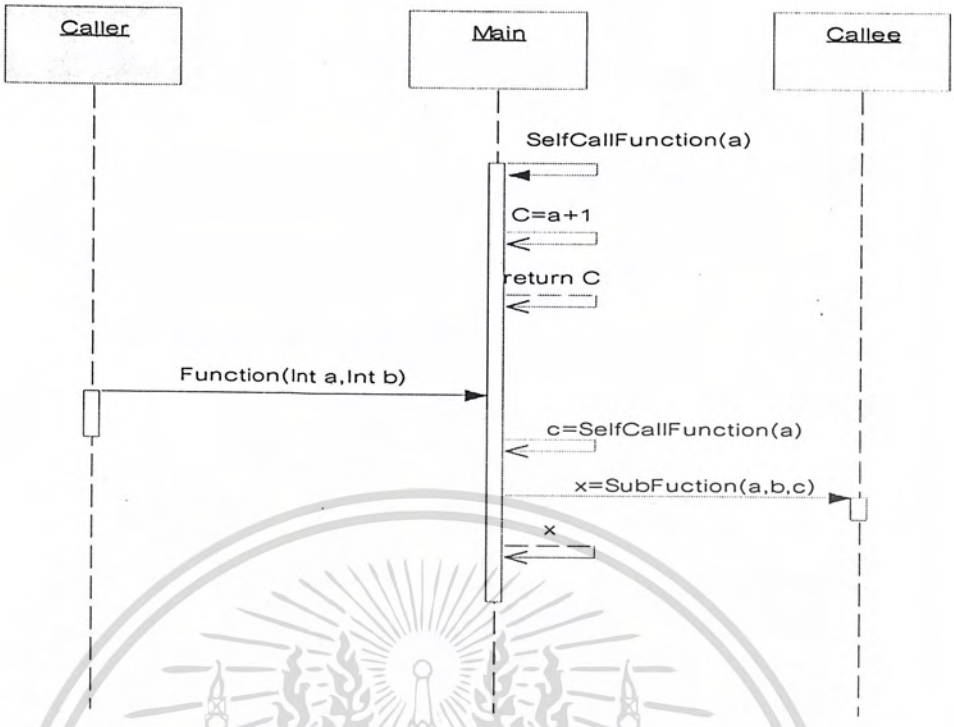
องค์ประกอบของไดอะแกรมแบบบนลงล่างที่ใช้ในสถาปัตย์กรรมแบบที่มีลักษณะเป็นลำดับชั้น (Layer architecture)นั้น ในแต่ละชั้น(Layer)จะประกอบด้วยออบเจกท์(object)3กลุ่มด้วยกันคือ**1 ผู้เรียก (Caller) 2 ตัวหลักที่เราพิจารณา(Main) 3 ผู้ถูกเรียก(Cal lee)** ซึ่งจะมุ่งเน้นพิจารณาเฉพาะใน **ตัวหลักที่เราพิจารณา(Main)**เป็นหลัก ส่วน **ผู้เรียก(Caller)** และ**ผู้ถูกเรียก(Cal lee)** จะเป็นแค่เพียงองค์ประกอบที่ให้เห็นภาพชัดเจนเท่านั้น (ใน**2 ตัวหลักที่เราพิจารณา(Main)** เท่านั้นที่แสดงการจัดการ(process))

ขั้นตอนการทำงาน

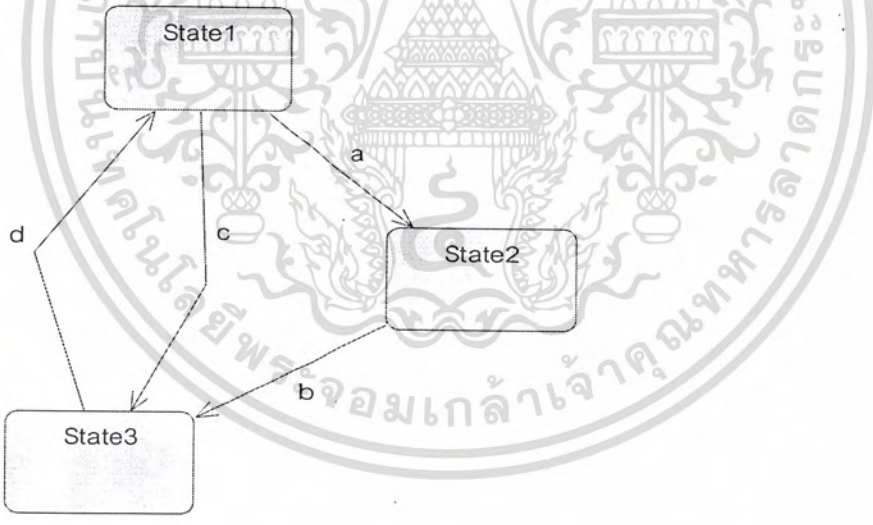
ผู้เรียก(Caller) จะไปเรียก **ตัวหลักที่เราพิจารณา(Main)**หลังจากนั้น จะแสดงการจัดการ โดยอาจจะไปเรียกผู้ถูกเรียกเพื่อใช้งานฟังก์ชัน(function)ของผู้ถูกเรียก หรือ ประมวลผลในตัวเอง โดยในผู้เรียก**(Caller)**คือ ผู้ที่อยู่ในชั้น(Layer)ที่ต่ำกว่า อาจเรียก**ตัวหลักที่เราพิจารณา(Main)**หลายครั้งแต่จะแสดงในชั้น(Layer)ของ**ตัวหลักที่เราพิจารณา(Main)**เป็นฟังก์ชัน(function)เพียงครั้งเดียว(ตามหลักการของการออกแบบ แบบบนลงล่างTop Down Design)

ในกลุ่มของ**ตัวหลักที่เราพิจารณา(Main)** อาจรวมถึงแอคทิบิวส์(attribute)ที่อยู่ในออบเจกท์ของ **ตัวหลักที่เราพิจารณา(Main)** ได้อีกด้วย แต่โดยทั่วไปควรจะอยู่แค่ระดับฟังก์ชัน(function)ของ**ตัวหลักที่เราพิจารณา(Main)**เพื่อประโยชน์การออกแบบ แบบบนลงล่าง(Top Down Design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.22 ภาพตัวหลักที่เราพิจารณา(Main)ทำการเรียกฟังก์ชันตัวเอง
ในบางครั้งในตัวหลักที่เราพิจารณา(Main) อาจทำการเรียกตัวเองได้



รูปที่2.23 State Machine1แสดงการใช้ไดอะแกรมสถานะ(state machine diagram)ร่วมกับงานไดอะแกรม
จากภาพที่1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3

วิธีดำเนินการวิจัย

โครงงานนี้ได้ใช้แนวความคิดของซอฟท์แวร์เอนจิเนียริ่งรวมกับทฤษฎีข้างต้นมาเพื่อการพัฒนาดังนี้

3.1 การรวบรวม ความต้องการ(Requirement)สำหรับออกแบบเอดดูเทรนเม้นท์

- ความต้องการ(Requirement)
  เกมของเราจะสอนใครและสอนอะไร (For Who & For What)
- วิเคราะห์(Analysis)
  เราต้องทำอะไรบ้างเพื่อสอนเขา(What we will go to do for the user)
- ออกแบบ(Design)
  เราจะสอนอย่างไร(How we will do for the User)

3.1.1ความต้องการ(Requirement)

วัตถุประสงค์

1 เพื่อสอนพื้นฐานของโครงสร้างของเอเจนท์(Structure of Agent) ที่เหมาะสมในการแก้ไขปัญหา(solving problem)

2 เพื่อสอนการนำพื้นฐานของโครงสร้างของเอเจนท์(Structure of Agent) ไปใช้ประยุกต์เข้ากับอัลกอริทึม(algorithm)แบบง่ายๆสำหรับบุคคลที่มีความรู้มาระดับหนึ่ง (อายุมากกว่า 16 ปี** และพอจะมีความรู้ด้านการเขียนโปรแกรมมาบ้าง โดยอ้างอิงเรื่องอายุมาจากบทความ“Education Through Digital Entertainment” ของRenu Ahuja, Sugata Mitra, Rashmi Kumar and Monica Singh ซึ่งกล่าวไว้ว่าในเด็กวัยตั้งแต่13ปีขึ้นไปสามารถเรียนรู้ในเชิงนามธรรม(abstract)ได้)

หมายเหตุ **แต่ไม่ควรเริ่มที่อายุ11ปีเพราะต้องมีการเรียนรู้การเขียนโปรแกรมพื้นฐานและโครงสร้างข้อมูล(data structure)ที่จำเป็นของภาษาpythonมาก่อน(หลักสูตรปัจจุบันกำหนดไว้ที่อายุ16ปีสำหรับโครงสร้างข้อมูล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 วิเคราะห์(Analysis)

1 <u>ต้องทำอะไรบ้างเพื่อให้เกมสนุก</u>

เพื่อความครอบคลุมในการจัดประเภทเกม เอดดูเทรนเม้นท์ เราได้แบ่ง Edutainment ของเราเป็น 2 ประเภทใหญ่ๆ (คัดลอกมาจาก J.Ajdari , I.Aulaskari , J.brown)

สื่อบันเทิงเพื่อการศึกษา(Educational Entertainment)หรือที่เรียกว่าเล่นและเรียนรู้(Play & Learn) ซึ่งต้องพิจรณาใน 2 ประเด็น

1 ความบันเทิงเพื่อบุคคล(Entertainment for the individual) ต้องดูช่วงอายุหรือลักษณะของบุคคลที่เป็นเป้าหมายเพื่อที่สร้างโปรแกรมขึ้นมาเพื่อให้ความรู้

2 ความเข้มข้นของความบันเทิง(Varying degrees of entertainment) การที่จะสร้างโปรแกรมเพื่อการเรียนรู้ประเภทนี้ได้ควรจะมี เนื้อเรื่อง(Story Line)ที่เหมาะสมกับการสอดแทรกความรู้เพื่อสร้างความรู้สึกสมจริงที่จะนำความรู้นั้นไปใช้ในเกมและให้ความบันเทิงแก่ผู้เรียนรู้

การศึกษาบันเทิงEntertaining Education การศึกษาโดยใช้สื่อบันเทิง (ใช้สอนเพื่อปฏิบัติจริง ในเชิง ซี เอ ไอ CAI(Computer-Assisted Instruction))

โดยโครงงานของเราพยายามจะจัดเป็น**สื่อบันเทิงเพื่อการศึกษา(Educational Entertainment)** เพราะต้องการชักจูงเยาวชนอายุ16ปีขึ้นไปโดยความสามารถของความบันเทิง

2 <u>ทำไมต้องเป็นแอ๊คชั่น อาพีจี(Action RPG)</u>

<u>ทำไมต้องเป็นอาพีจี(RPG)</u>

อ้างอิงจากหนังสือ Programming Role-playing Games With DirectXของ Jim Adam เคยกล่าวไว้ว่า "The RPG is Mother of all Game" เนื่องจากRPGมีคุณสมบัติทางด้านสิ่งแวดล้อม(environment) เพื่อจำลองเอไอ(AI)สูงมาก โดยเฉพาะกฎ(rules) ซึ่งสามารถครอบคลุมเกมทั้งโลกได้ แม้แต่เกมที่เป็นพิศวง(Puzzle)ก็ยังอาจระบุกฎ(rules)ของมันลงเงื่อนไขเกม(quest)ย่อยได้อีกด้วย การที่เรานำRPGเพราะมีกฎ(rules) หลายอย่างเหมาะที่จะเอามาใช้ทำสิ่งแวดล้อม(environment) ของเอไอ(AI)แม้แต่จะจำลองเกมพวกวัมปัสเวริด์(Wumpus world)กลับมาใช้ก็ยังทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง กฎ(rules) ที่เราจะนำมาใช้ในเกมของเรา

กฎRules1:ธาตุและการแพ้ทางในสายเวทย์มนต์ (element rules)

ธาตุไฟจะแพ้ น้ำแข็ง แต่จะชนะธาตุดิน

ธาตุน้ำแข็งจะแพ้ สายฟ้า แต่จะชนะไฟ

ธาตุสายฟ้าจะแพ้ ดิน แต่จะชนะน้ำแข็ง

ธาตุดินแพ้ไฟ แต่จะชนะสายฟ้า

ธาตุพิษที่จะลดทอนพลังชีวิตของศัตรูตามเวลา

ธาตุศักดิสิทธิ(holy)เพิ่มพลังชีวิตของพวกเรา

ซึ่งจากกฎข้างต้นเราสามารถนำมันไปเพิ่มการกระทำ(action)ให้กับเอเจนท์ได้ทำให้เอเจนท์มีความสามารถที่จะกลายเป็น เอเจนท์ผู้มีเหตุผล(Rationality agent)สูงขึ้น นอกนั้นให้ผลเป็นปกติ

กฎ(Rules2):ทักษะ(skill)และตัวละคร

ในเกมRPGตัวละครแต่ละตัวมีความสามารถในการใช้ทักษะ(skill)ไม่เท่ากันเช่น

-พ่อมดสามารถใช้เวทย์มนต์(magic)ระดับสูงได้แต่ใช้การโจมตีทางกายภาพ(physic attack)แทบไม่ได้เลย

-อัศวินใช้การโจมตีทางกายภาพได้แต่ใช้เวทย์มนต์ระดับสูงไม่ได้

-อัศวินเวทย์มนต์(paladin)มีความสามารถทั้งเวทย์มนต์(magic)ระดับสูงและการโจมตีทางกายภาพ(physic attack) แต่ไม่ได้มีความสามารถทางด้านใดด้านหนึ่งครบทั้งหมด

-สัตว์ประหลาด(Monster)สามารถใช้เวทย์มนต์(magic)ตามธาตุหลักของตนได้ดี

กฎ(Rules3):การมองเห็น(Range)

ในเกมอาพีจี(RPG)ตัวละครแต่ละตัวมีความสามารถในการใช้มองเห็นไม่เท่ากัน

-นักธนู ก็เหมือนคนที่ฝึกฝนมาเป็นพิเศษก็มองได้ไกล

-พ่อมดมีทักษะหยั่งรู้ก็อาจจะเปิดหมอกหรือม่านพรางได้ในตอนที่ใช้ทักษะ

-สัตว์ประหลาดบางประเภทมีความสามารถในการมองเห็นได้ไกล

-ตัวละครบางตัวมีทักษะที่ทำให้ศัตรูตาบอดทำให้การมองเห็น ของการมองเห็นสั้นลงช

ทำไมต้องเป็นแอ็คชั่น

แม้ว่านิยามอาพีจี(RPG)จะมีกฎที่เป็นสากลทุกอย่าง แต่ก็เกมของเราต้องการคุณสมบัติเป็นเวลาจริง(real time)คุณสมบัติแอ็คชั่น(action)จะทำให้สิ่งแวดล้อม(environment) ของเกมเป็นแบบเวลาจริง(real time)ทำให้ได้ทดลองเขียนAIในเวลาจริง(real time)ซึ่งต้องการความเร็วของอัลกอริทึ่ม(algorithm)และการรับรู้(percept)ที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรานำกฎ(rules)และการเขียนโปรแกรมแบบเวลาจริง(real time)ไปใช้ทำอะไร?
เพื่อสร้างความท้าทายในการโปรแกรม AI ให้เป็น เอเจนท์ผู้คิดอย่างมีเหตุผล

เอเจนท์ผู้คิดอย่างมีเหตุผล– เอเจนท์ผู้คิดอย่างมีเหตุผล ต้องขึ้นกับปัจจัย4อย่างดังนี้

-ประสิทธิภาพแห่งความสำเร็จ **ขึ้นกับสิ่งที่ผู้ใช้ เขียนโปรแกรมลงไป

-ความรอบรู้มาก่อนเกี่ยวกับสิ่งแวดล้อม **แบบเวลาจริงทำให้สิ่งแวดล้อมของระบบเปลี่ยนแปลงตลอดเวลา

-ความสามารถในการแสดงแอ็คชั่น(action)ของเอเจนท์ **กฎ 1,2

-ความสามารถในการรับรู้ **กฎ 3

3.1.3 ออกแบบจากโครงสร้างของเอเจนท์(Structure of Agent)

เอเจนท์แบบง่ายตอบสนองทันที(Simple Reflex Agent)

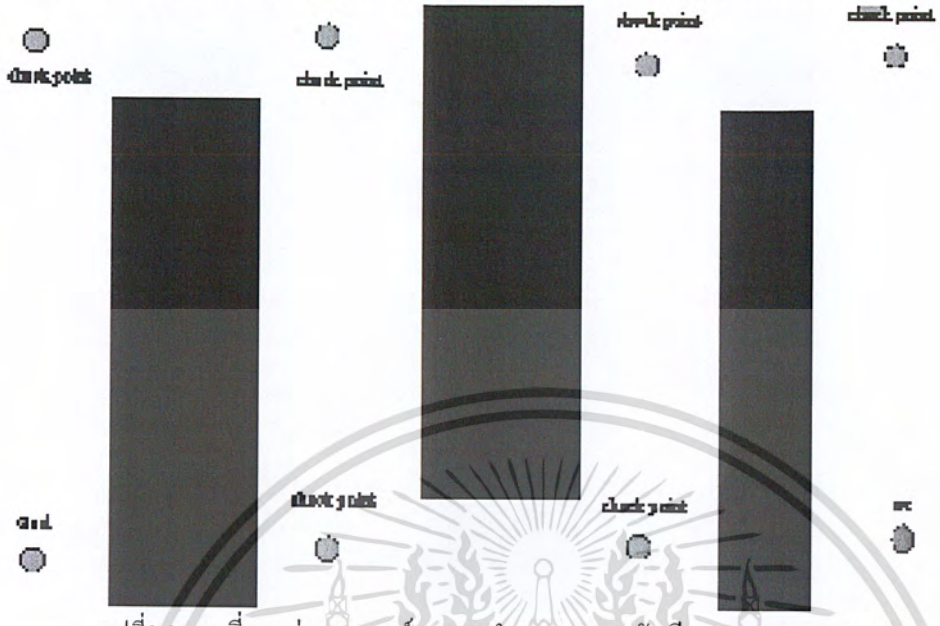
- นำไปประยุกต์เป็นเกมคือ ฉากที่1การสั่งยามเฝ้าประตู ซึ่งจะให้ผู้ใช้เขียนโปรแกรมโดยใช้พื้นฐานของเอเจนท์แบบง่ายตอบสนองทันที โดยจะให้เขียนโปรแกรมให้ยามรับรู้(percept)ว่าหากมีศัตรูเข้ามาในระยะดาบฟันถึงให้เลือกเวทย์ที่เป็นทักษะอัดพลังใส่ดาบให้เหมาะสมกับสัตว์ประหลาด(monster) เพื่อโจมตีให้ตายในดาบเดียวหากทำไม่ได้ยามจะเป็นฝ่ายตายและจบเกมไม่ผ่านด่านดังกล่าว

รูปที่3.1แผนที่ของด่านเอเจนท์แบบง่ายตอบสนองทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอเจนท์แบบจดจำตอบสนองทันที(Model-Based reflex Agent)

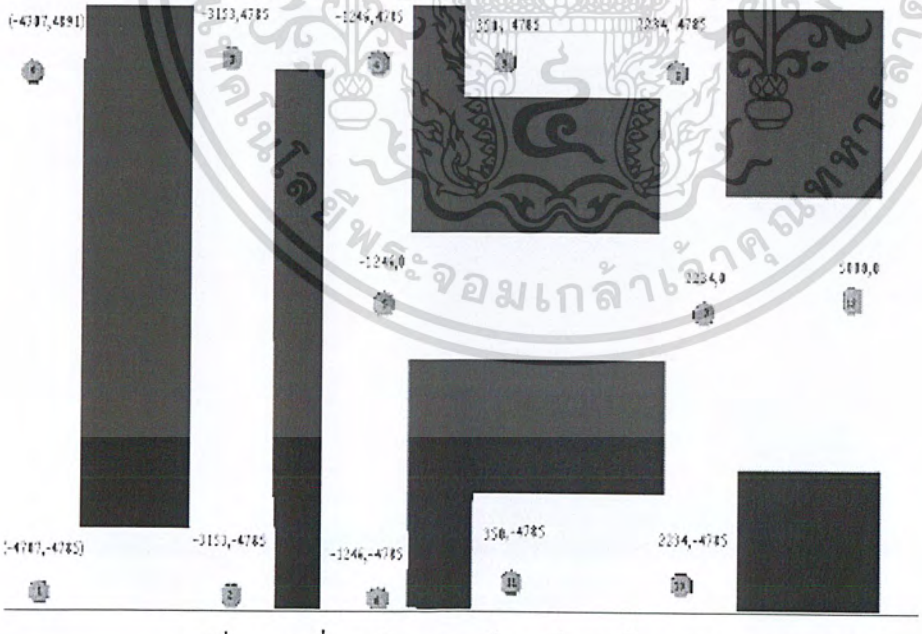
นำไปประยุกต์เป็นเกมคือ ฉากที่2: สั่งตัวละครเดินตามแผนที่ ที่มีลักษณะเป็นทางเดินทางเดียว



รูปที่3.2แผนที่ของด่านเอเจนท์แบบจดจำตอบสนองทันที(Model-Based reflex Agent)

เอเจนท์แบบค้นหาเป้าหมาย(Goal-based agents)

- นำไปประยุกต์เป็นเกมคือ ฉากที่ 3 สั่งตัวละครเดินตามmapที่มีลักษณะเป็นทางเดินทางหลายทาง แต่ถึงgoalได้แค่pathเดียว

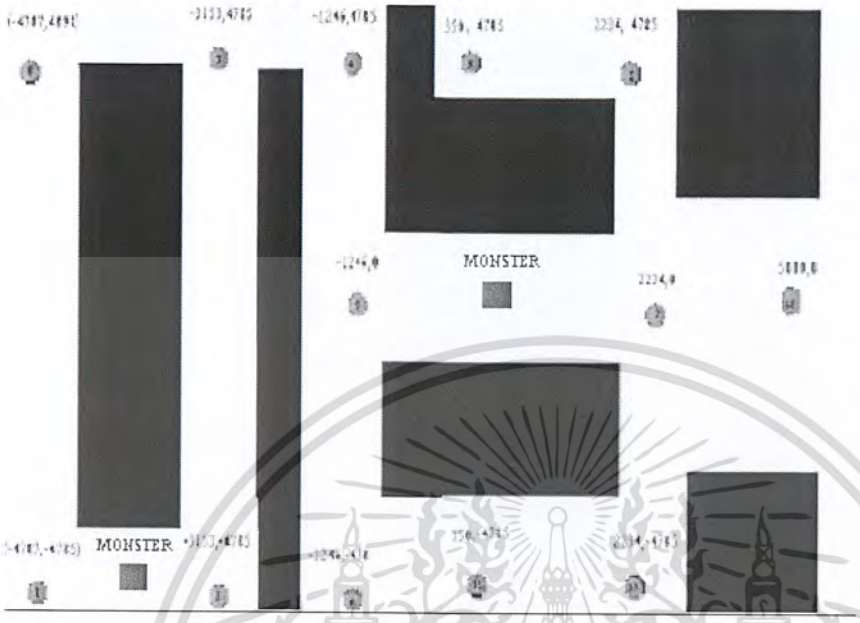


รูปที่3.3แผนที่ของด่านเอเจนท์แบบค้นหาเป้าหมาย(Goal-based agents)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอเจนท์แบบค้นหาเป้าหมายที่พึงพอใจที่สุด(Utility-based agents)

- นำไปประยุกต์เป็นเกมคือ ฉากที่ 4 สั่งตัวละครเดินตามแผนที่ที่มีลักษณะเป็นทางเดินทางหลายทางและถึงgoalได้โดยใช้หลายpath



รูปที่3.4เอเจนท์แบบค้นหาเป้าหมายที่พึงพอใจที่สุด(Utility-based agents)

มี4เส้นทางที่ทำให้ถึงเป้าหมาย(Goal) ให้เลือกเส้นทางสู่เป้าหมาย(goal)ที่มีสัตว์ประหลาด(monster)น้อยที่สุด ต้องวิเคราะห์2ขั้นคือหาgoalก่อนแล้วค่อยหา ค่าความพึงพอใจ(Utility)ที่มีประหลาด(monster)น้อยที่สุด

เอเจนท์แบบเรียนรู้(Learning Agent)

- การประยุกต์ใช้ในเกม ฉากที่ 5 จะทำเป็นการเรียนรู้การใช้ทักษะ(skill)ของตัวเอเจนท์เองต่อสภาพแวดล้อมใดๆโดยตัวเอเจนท์ไม่รู้อะไรเลย ผู้ใช้ต้องป้อนโปรแกรมลงในส่วนของการเรียนรู้(learning element) ให้ตัวเอเจนท์เลือกทักษะของตนเองมาทดลองใช้แบบลองผิดลองถูกแล้วเก็บไว้ในส่วนของการปรับปรุง(performance element)ซึ่ง ทักษะ(skill)แต่ละทักษะ(skill)จะมีคุณสมบัติต่างกันเช่นความเร็วในการพุ่งเข้าหาศัตรู ระยะการกระจาย ธาตุในการโจมตี และความรุนแรงให้เลือกมาใช้กับศัตรูที่เหมาะสมที่สุด

1 ส่วนของการวิเคราะห์ปัญหา(Problem generator) เนื่องจากเกมของเรามีความตั้งใจไว้ว่าจะทำเป็นRPGดังนั้น ส่วนของการวิเคราะห์ปัญหา(Problem generator)ของเราคือสั่งให้ ส่วนของการปรับปรุง(performance element) ทำการทดลองใช้ทักษะ(skill)พิเศษของตัวละครที่เพิ่มขึ้นมาตอนที่เลเวลอัพ(level up) จากค่าประสบการณ์(Experience) และเป้าหมายคือนำทักษะ(skill)นั้นไปใช้ให้เป็น

-สิ่งแวดล้อม(Environment)ในระดับเกมเอนจินจะเป็นป้อนให้ฟัซซี่เอกเพริดชิสเต็ม(Fuzzy Expert System)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 ส่วนของการเรียนรู้(learning element)คือจะเป็นตัวปรับปรุงการใช้ทักษะ(skill)โดยพิจารณาตามศัตรู(monster) ชนิดต่างๆ และการยืนเป็นกลุ่มห่างกันขนาดไหน และความเร็วของเวทย์มนต์เพื่อให้โดนโดยเป็นลักษณะของการคำนวณเพื่อตัดสินใจแล้วทำการปรับปรุง

-ใช้เจนเนติกส์(Genetic Algorithm)

3 ส่วนของการปรับปรุง(performance element)คือนำผลการทดลองที่ดีที่สุดและเหมาะที่สุดในแต่ละสถานการณ์ที่มาใช้ทดลองไป(return action)

-ใช้เจนเนติกส์(Genetic Algorithm)

4 ส่วนของการวิจัย(Critic)จะเป็นตัวคอยส่งผลการตอบสนอง(feed back)กลับมาที่ส่วนของการเรียนรู้(learning element)ว่าเอเจนท์ควรทำหรือตัดสินใจอย่างไรให้มีการปรับปรุงส่วนของการปรับปรุง(performance element)ให้ดีขึ้นไปอีก เช่นว่า ทักษะ(skill)นี้เคยใช้กับสัตว์ประหลาด(monster)ที่อยู่เดี่ยวๆแล้วน่าจะยังทดลองใช้กับสัตว์ประหลาด(monster)ที่อยู่เป็นกลุ่มด้วยนะ

-ใช้เจนเนติกส์(Genetic Algorithm)ร่วมกับฟัซซี่เอกเพริดซิสเต็ม(Fuzzy Expert System)

<u>สรุปผลการออกแบบเอดดูเทรนเม้นท์</u>

- ความต้องการ(Requirement)

  เกมของเราจะสอนใครและสอนอะไร (For Who & For What)

  =>วัตถุประสงค์เป็นตัวตอบคำถามโดยการDefine Goalเอาไว้

- วิเคราะห์(Analysis)

  เราต้องทำอะไรบ้างเพื่อสอนเขา(What we will go to do for the user)

  =>จากทำไมต้องเป็นAction RPG

- ออกแบบ(Design)

  เราจะสอนอย่างไร(How we will do for the User)

  =>เริ่มทำการออกแบบโดยยึดหลักการของโครงสร้างของเอเจนท์(structure of Agent)รวบกับเนื้อเรื่อง(story Line)ที่ได้จากวิเคราะห์ (analysis phase)

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย
ในพระบรมราชูปถัมภ์สมเด็จพระเจ้าอยู่หัว

3.2<u>เครื่องมือที่ใช้ในการพัฒนา</u>

1 Visual C++ ,Direct X 9.0 (สำหรับเกมและเอนจิน)

2 3DS Max 6,Panda Exporter , Qtip Exporter(สำหรับการสร้างModel)

3 Python2.3 (สำหรับภาษาสคริป)

4 Python Win IDE (สำหรับที่ใช้เขียนโปรแกรมและทดสอบ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ขั้นตอนในการสร้างโครงงาน

3.3.1การสำรวจความต้องการ(Requirement Capture)

3.3.1.1 ทำการลิสต์ลักษณะของระบบ(Feature List)

คุณสมบัติ	สถานะ	ค่าใช้จ่าย	ความสำคัญ	ระดับความเสี่ยงร้ายแรง (critical risk)	ระดับความเสี่ยงล่าช้า (significant risk)**
ระบบเกม (Game System)	พิจารณาแล้ว	1 ปี :1 คน	มากที่สุด	ปานกลาง	มาก
ระบบเอนจิน (Game Engine)	พิจารณาแล้ว	1 ปี:1 คน	มากที่สุด	มาก	มาก
เอเจนท์แบบง่ายตอบสนองทันที (Simple Reflex Agent)	พิจารณาแล้วในตอนแรกแต่ยกเลิก**	1 วัน:1 คน	มาก	มาก	ไม่ผ่านเพราะขึ้นตรงกับเกมเอนจิน
เอเจนท์แบบจดจำตอบสนองทันที (Model-Based reflex Agent)	พิจารณาแล้ว	2 วัน:1 คน	ปานกลาง	ปานกลาง	น้อย
เอเจนท์แบบค้นหาเป้าหมาย(Goal-based agents)	พิจารณาแล้ว	2 วัน:1 คน	ปานกลาง	ปานกลาง	น้อย
กราฟฟิก (Graphic)	ยังไม่แน่นอน	out sourcing	ปานกลาง	ปานกลาง	น้อย
เอเจนท์แบบค้นหาเป้าหมายที่พึงพอใจที่สุด (Utility-based)	ไม่ผ่าน เกิดความเสี่ยงร้ายแรงสูงมาก	----------	----------	----------	----------
เอเจนท์แบบเรียนรู้ (Learning Agent)	ความเสี่ยงร้ายแรงสูงมากที่สุด	----------	----------	----------	----------

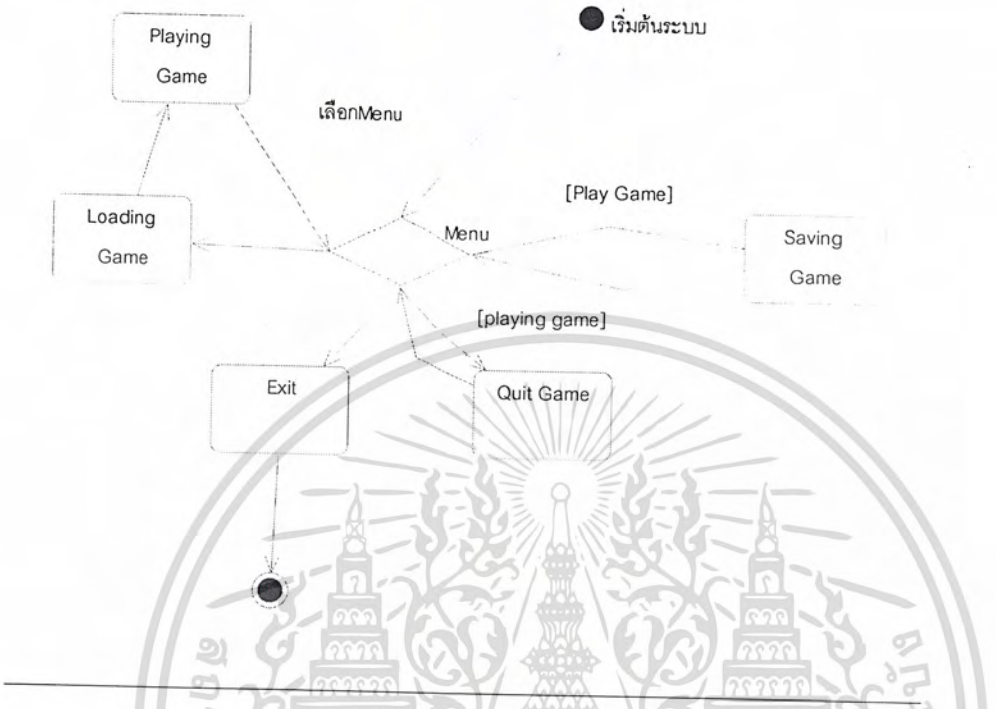
ตารางที่3.1 ลิสต์ลักษณะของระบบ(Feature List)

ระดับความเสี่ยงร้ายแรง(critical risks) คือระดับความเสี่ยงที่ทำให้ระบบล่ม

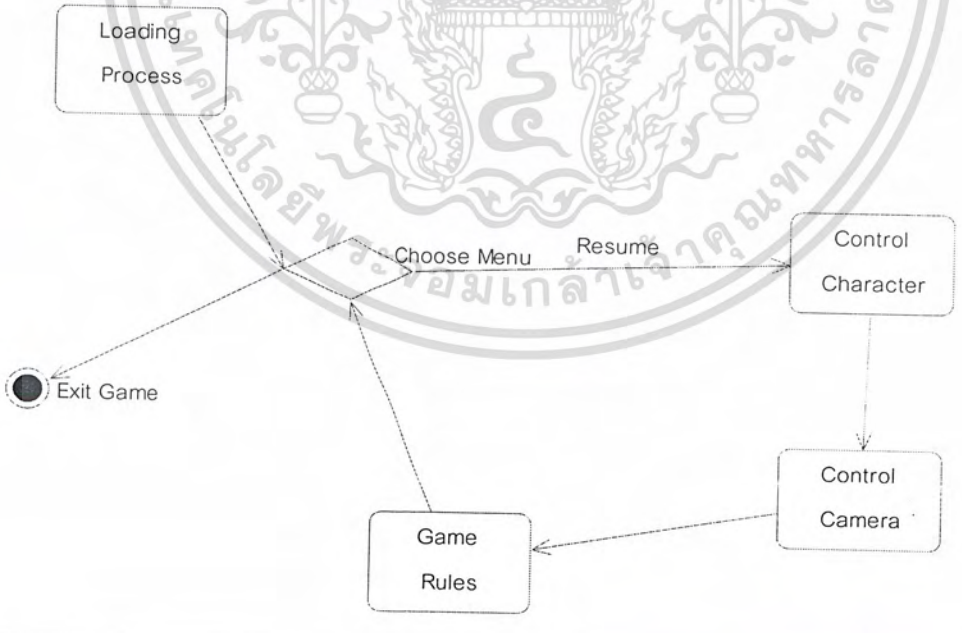
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับความเสี่ยงล่าช้า(significant risks)คือระดับความเสี่ยงที่ทำให้ระบบล่าช้า(มักพบปัญหานี้ได้ในช่วง Elaboration phase)

3.3.1.2 โมเดลธุรกิจ(Business Model)



รูปที่3.5ระบบโหลดระบบเซฟ



รูปที่3.6ระบบเล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

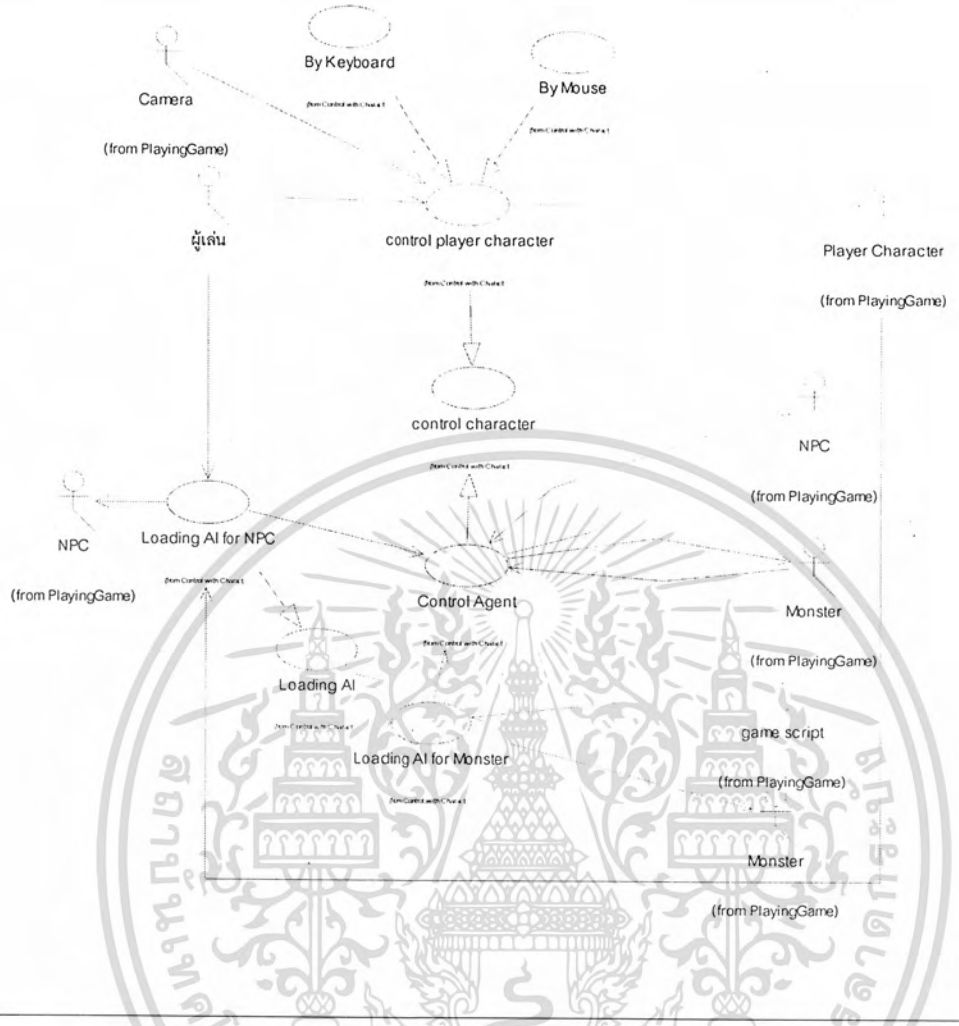
3.3.1.3 <u>ความต้องการที่เห็นได้ชัดของระบบ(**Functional Requirement**)</u> ซึ่งสามารถอธิบายด้วยuse case diagramทั้งหมด

รูปที่3.7 Main Use Case Diagram

<u>**ยูสเคสเพลยอิงเกม(Use case Playing Game)**</u>จะประกอบไปด้วยuse case diagramย่อย4 use case diagram คือ

รูปที่3.8 Camera Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.9 Control with Character Use case diagram

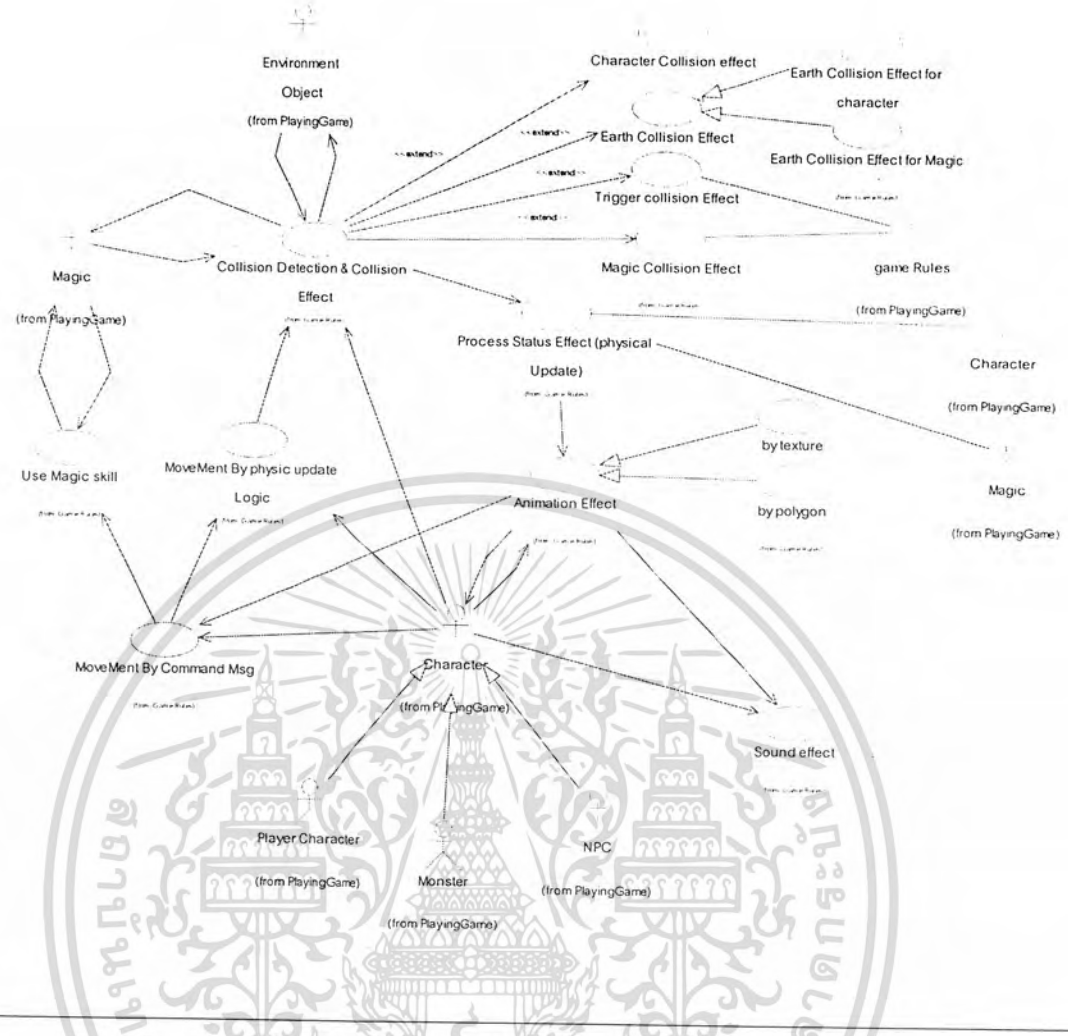
คำศัพท์(Glossary)

1 ตัวละครของผู้เล่น(Player Character)คือตัวละครของผู้เล่นที่ถูกควบคุมโดยตัวผู้เล่น

2 NPCคือ ตัวละครไร้ผู้เล่น(Non Player Character) เป็นผู้ไม่คุกคามตัวละครของผู้เล่น

3 สัตว์ประหลาด(Monster)คือตัวละครผู้คุกคามตัวละครของผู้เล่นและอาจจะคุกคามตัวละครไร้ผู้เล่นrด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.10Game Rules Use Case Diagram

คำศัพท์(Glossary)

1ตัวละคร(Character)คือตัวละครในเกมที่ประกอบไปด้วย NPC Monster และ Player Character

2 เวทย์(Magic) คือ (Magic object)ที่มีตัวตนเป็นรูปร่างในเกม

3วัตถุใดๆ(Environment Object)คือวัตถุที่ปรากฏในฉากนอกเหนือวัตถุใดๆในข้างต้นอันได้แก่ ตัวกระตุ้นประจำแผนที่(Map trigger) ของแผนที่(Map)

4 ใช้เวทย์(Use Magic Skill) คือ ยูสเคส(use case)ที่ก่อให้เกิดวัตถุประเภทเวทย์(Magic)

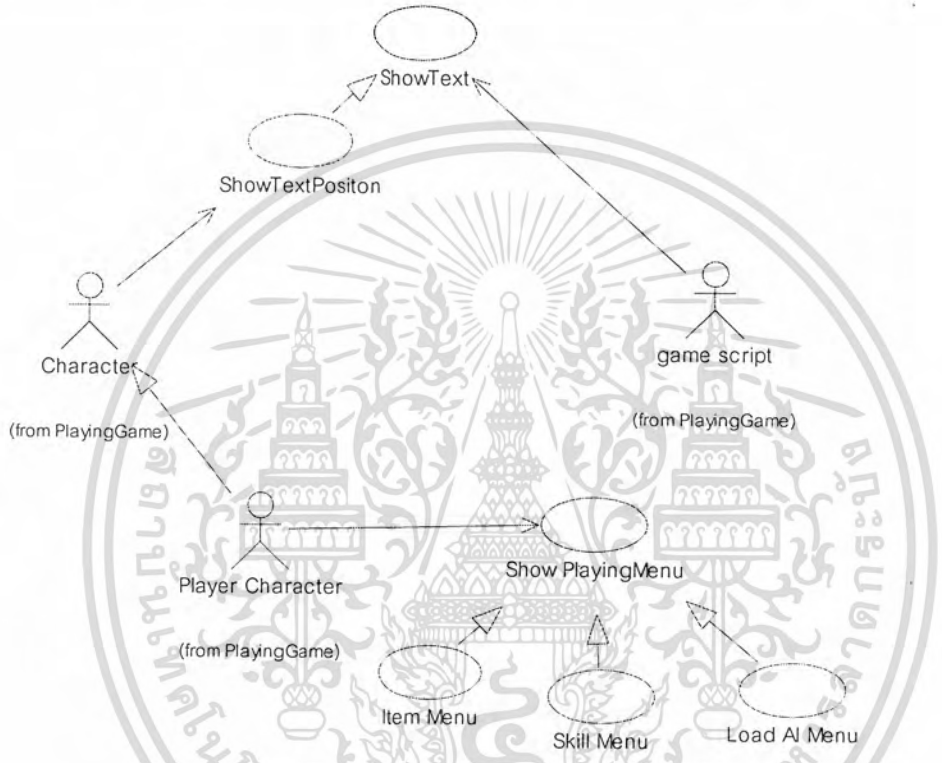
5 เช็กชน(Collision Detection) คือ ยูสเคส(use case)ที่คอยเช็ก(Detect)การเกิดชนและทำการประมวลผลคาดการในเชิงโลจิค(Logical effect)ก่อนที่จะทำการเปลี่ยนแปลง(Update)ในเชิงตอนทำจริง (physical effect)

6 ปรากฏการณ์(effect)คือปรากฏการณ์ที่นำการประมวลผลจากยูสเคส เช็กชน(use case Collision Detection)ไปทำการUpdateในเชิงphysical effect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 เช็กชนตัวกระตุ้น(Trigger Collision effect)คือ ปรากฏการณ์ที่ทำให้เกิดเหตุการณ์พิเศษที่ต้องไปเชื่อมต่อกับไฟล์สคริปต์ เช่นการโหลดแผนที่ใหม่ การเกิดสัตว์ประหลาด(Monster)ขึ้นมาใหม่ล้วนแต่ต้องผ่านไฟล์สคริปต์ทั้งสิ้น

8 เช็กชนเวทย์(Magic Collision effect)คือ ปรากฏการณ์ที่ทำให้เกิดการทำร้าย(Damage)บนตัวละคร(ในที่นี้รวมเรื่องระยะโจมตีกายภาพด้วย)ซึ่งจะส่งผลกระทบกับวัตถุเวทย์(Magic Object)และตัวละคร(Character)ด้วย เช่นทำให้วัตถุเวทย์(Magic Object)ถูกทำลายไป หรือแตกตัวไปแล้วเกิดการเรียกใช้ใช้เวทย์(Use Magic Skill)เพื่อสร้าง วัตถุเวทย์(Magic Object)ในรูปแบบอื่นๆอีกครั้ง



รูปที่3.11 ShowText&play Menu

3.3.1.4 <u>ความต้องการที่เห็นได้ไม่ชัดของระบบ(Non Funtional Requirement)</u>

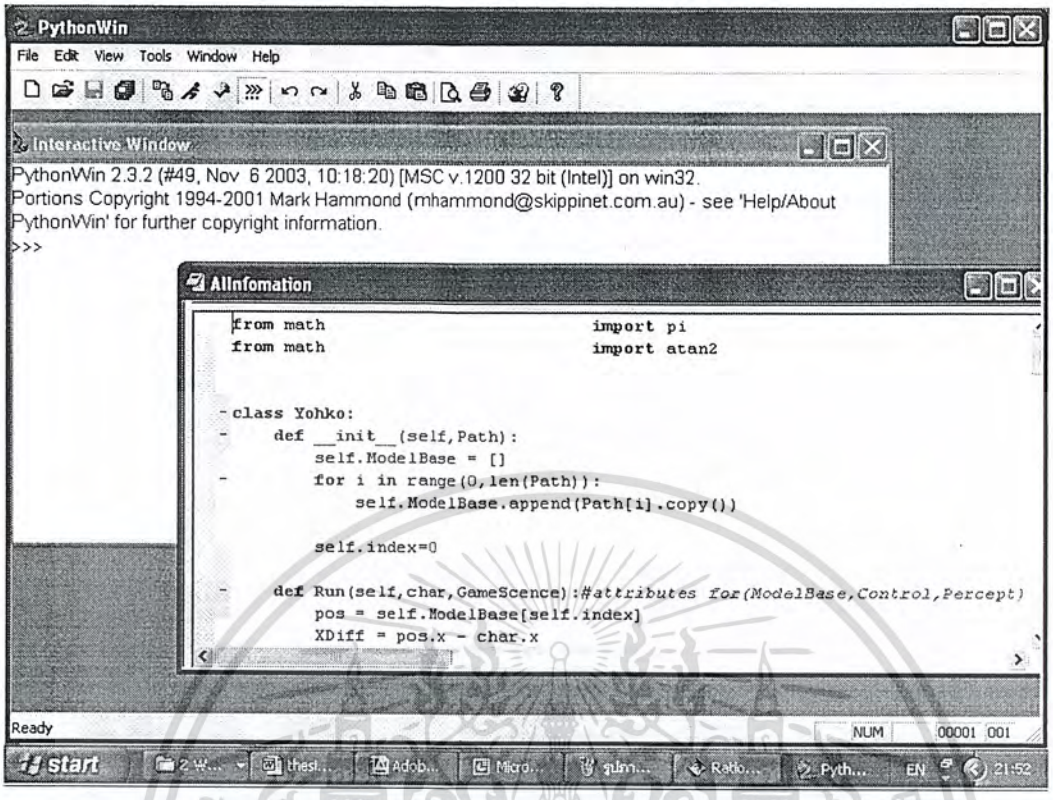
1 ระบบเกม(Game System)ต้องรองรับเกมเอนจิน(Game Engine)ที่สามารถรันสคริป AI

2ความสามารถของภาษาสคริปต้องอยู่ในระดับสูงพอที่จะรองรับเอเจนท์แบบค้นหาเป้าหมายเป็นอย่างน้อยและต้องทำรีเคอซีฟ(recursive)ได้

3 ต้องมีประสิทธิภาพ(performance)ที่เร็วพอที่จะไม่รบกวนทรัพยากร(resource)ตัวอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

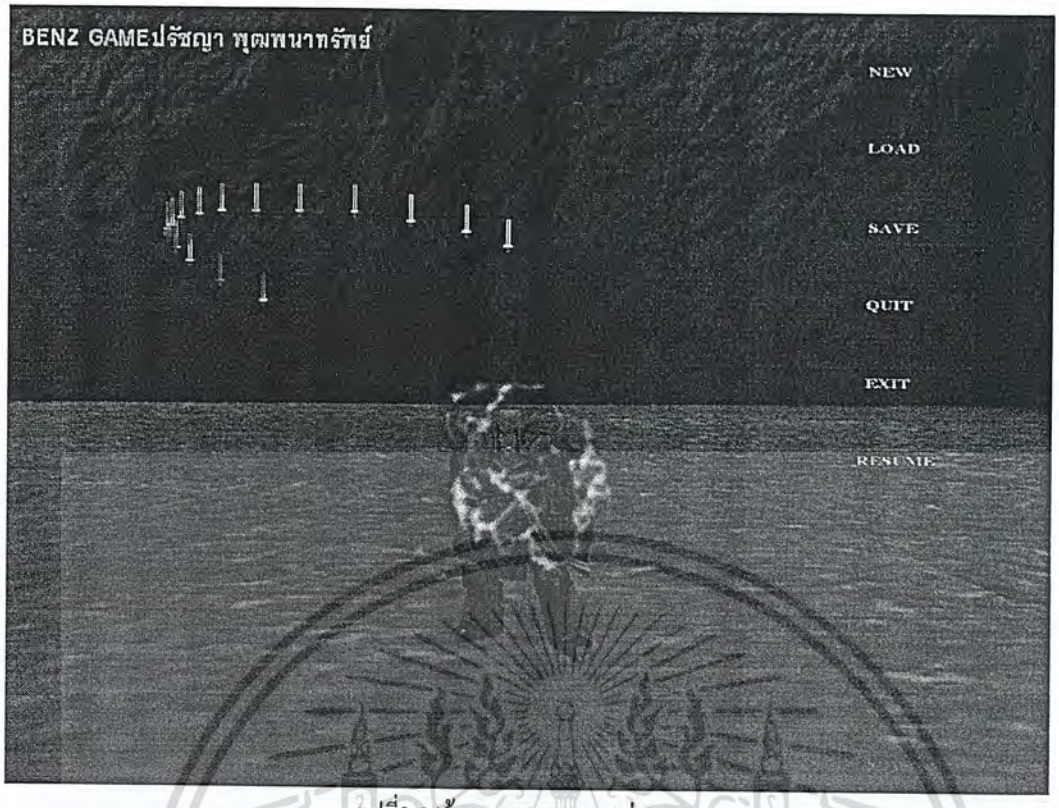
3.3.1.5 เครื่องมือต้นแบบ (Prototype)



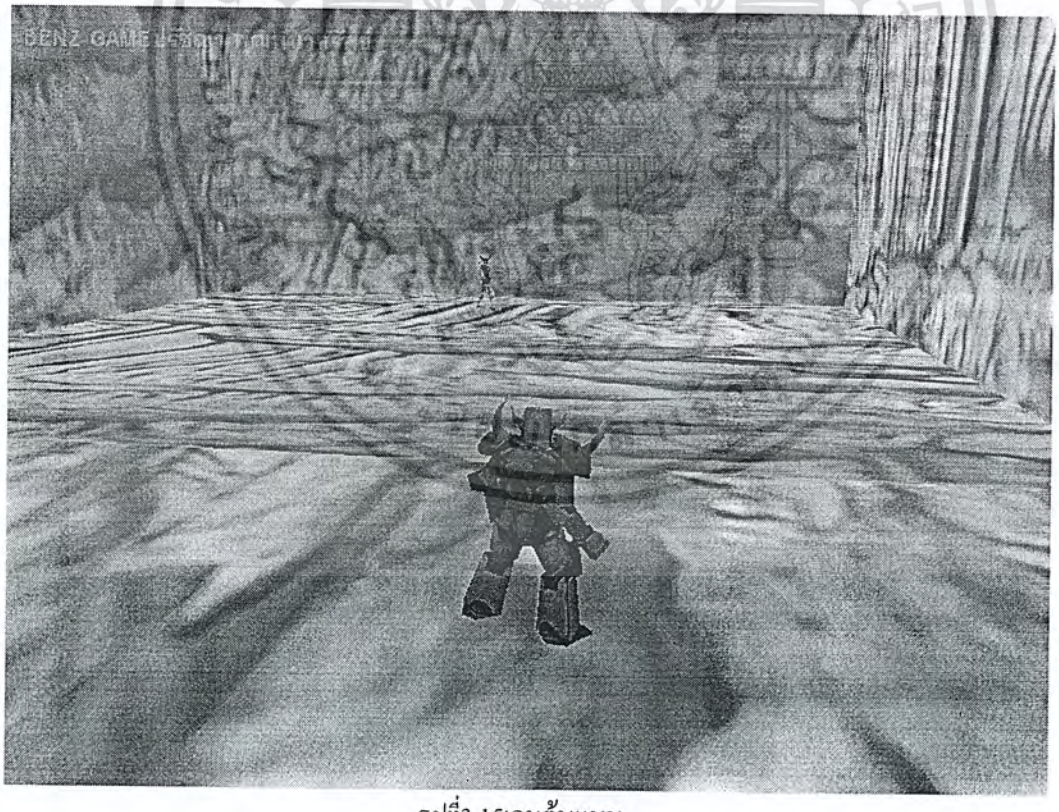
รูปที่3.12ต้นแบบเครื่องมือติดต่อกับผู้ใช้เขียนโปรแกรม(GUI Prototype)

รูปที่3.13ต้นแบบGUIก่อนเข้าเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.14ต้นแบบGUIขณะเล่นเกม



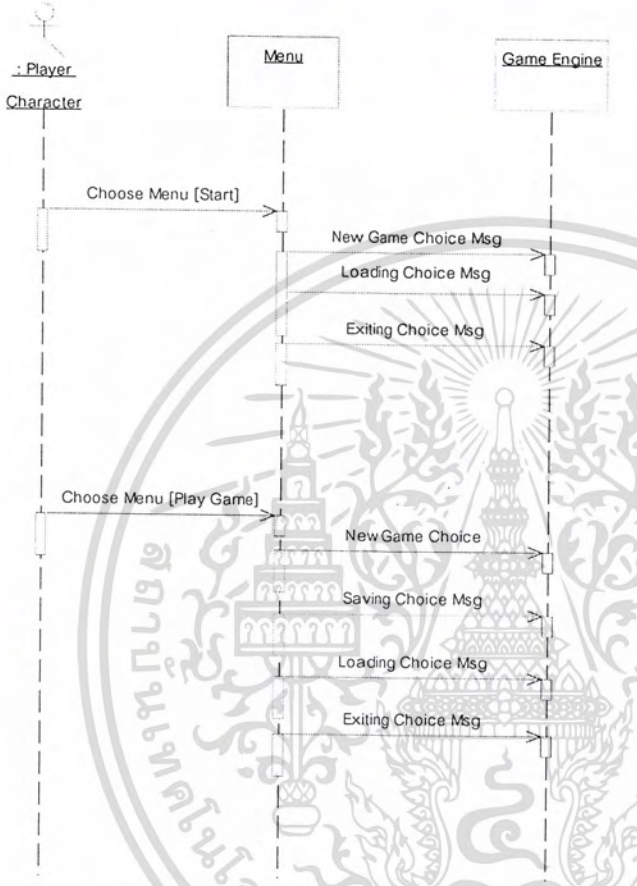
รูปที่3.15เกมต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2การวิเคราะห์(Analysis)

3.3.2.1 วิเคราะห์ยูสเคส(Analysis use case view)เราจะใช้ซีเควนท์ไดอะแกรมอธิบายในแต่ละยูเคสของยูสเคสไดอะแกรม

ในUse case Diagram Main Menu

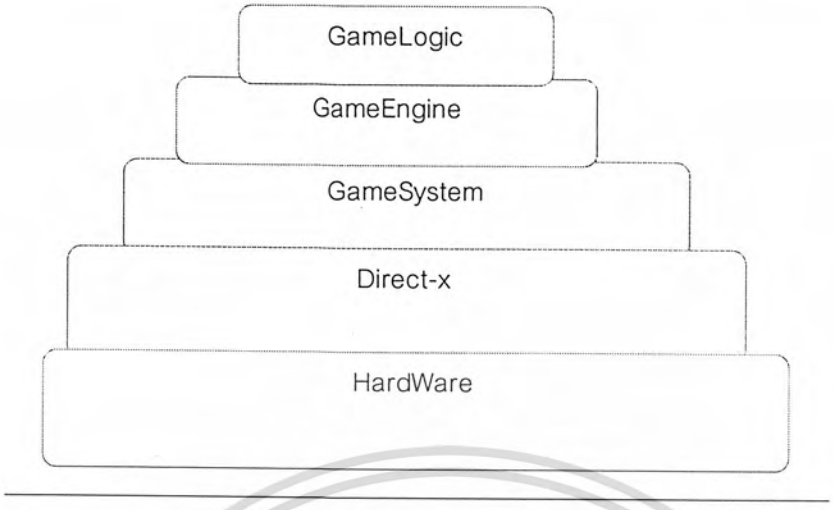


รูปที่3.16Use case choose Menu

เนื่องจากซีเควนท์ไดอะแกรมที่ใช้ในยูสเคสวิวจะกลายมาเป็นซีเควนท์ไดอะแกรมแบบบนลงล่างในภายหลังและไม่มีความจำเป็นต่อระบบอีกต่อไปจึงขออธิบายโดยสังเขป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.2 วิเคราะห์สถาปัตย์กรรม(Architect Overview)



รูปที่3.17Architect descriptions & Glossary

คำศัพท์(Glossary)

1 ระบบของเกม(Game System)คือระบบgameที่ประกอบไปด้วยเอนจินย่อยได้แก่ เอนจินเสียง (sound Engine) , เอนจินภาพ(Graphic Engine) , เอนจินฟิสิกส์(Physic Engine) ที่ติดต่อกับ(hardware)

2 เกมเอนจิน(Game Engine) คือระบบที่จัดการกับไฟล์(file)ซึ่งประกอบไปด้วยสคริป(script)เอนจินที่ทำหน้าที่นำไฟล์จากภายนอกเข้าไปประมวลผลภายในระบบของเกม

3 เกมโลจิค(Game Logic)คือไฟล์ที่ระบุข้อมูลของเกมและตัวละครสู่Game Engine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

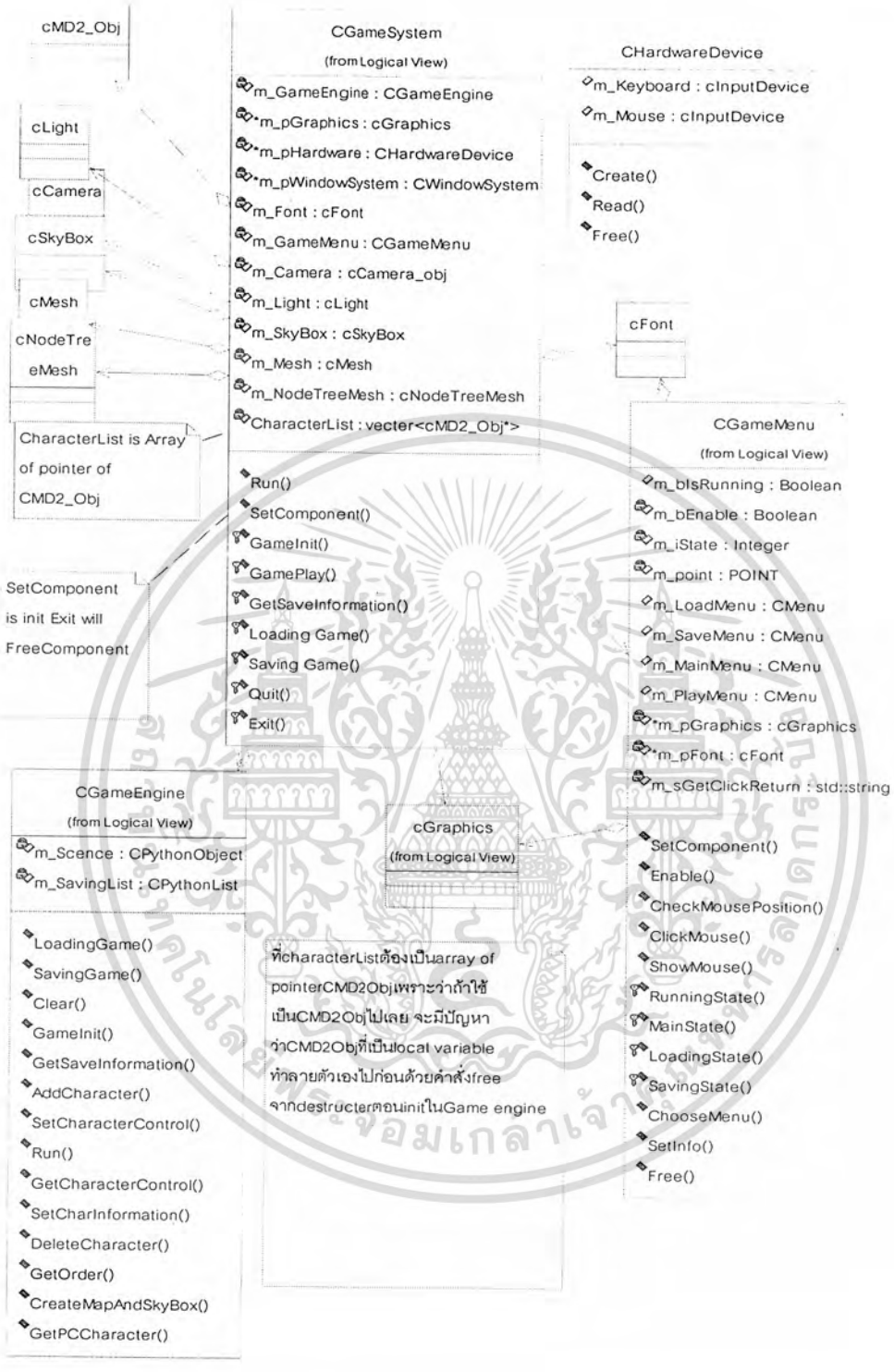
3.3.3 <u>การออกแบบ(Design)</u>

จะทำโดยใช้ซีเควนท์ไดอะแกรมแบบบนลงล่างอธิบายคลาสในโลจิคคอลวิว โดยจะยกตัวอย่างแค่ในเกมเอนจินเท่านั้นเพราะไดอะแกรมไม่สามารถทำให้ชัดเจนใน Microsoft wordได้

3.3.3.1 การออกแบบคลาส(class)

รูปที่3.18Main Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.19CGame System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่3.20CGameEngine

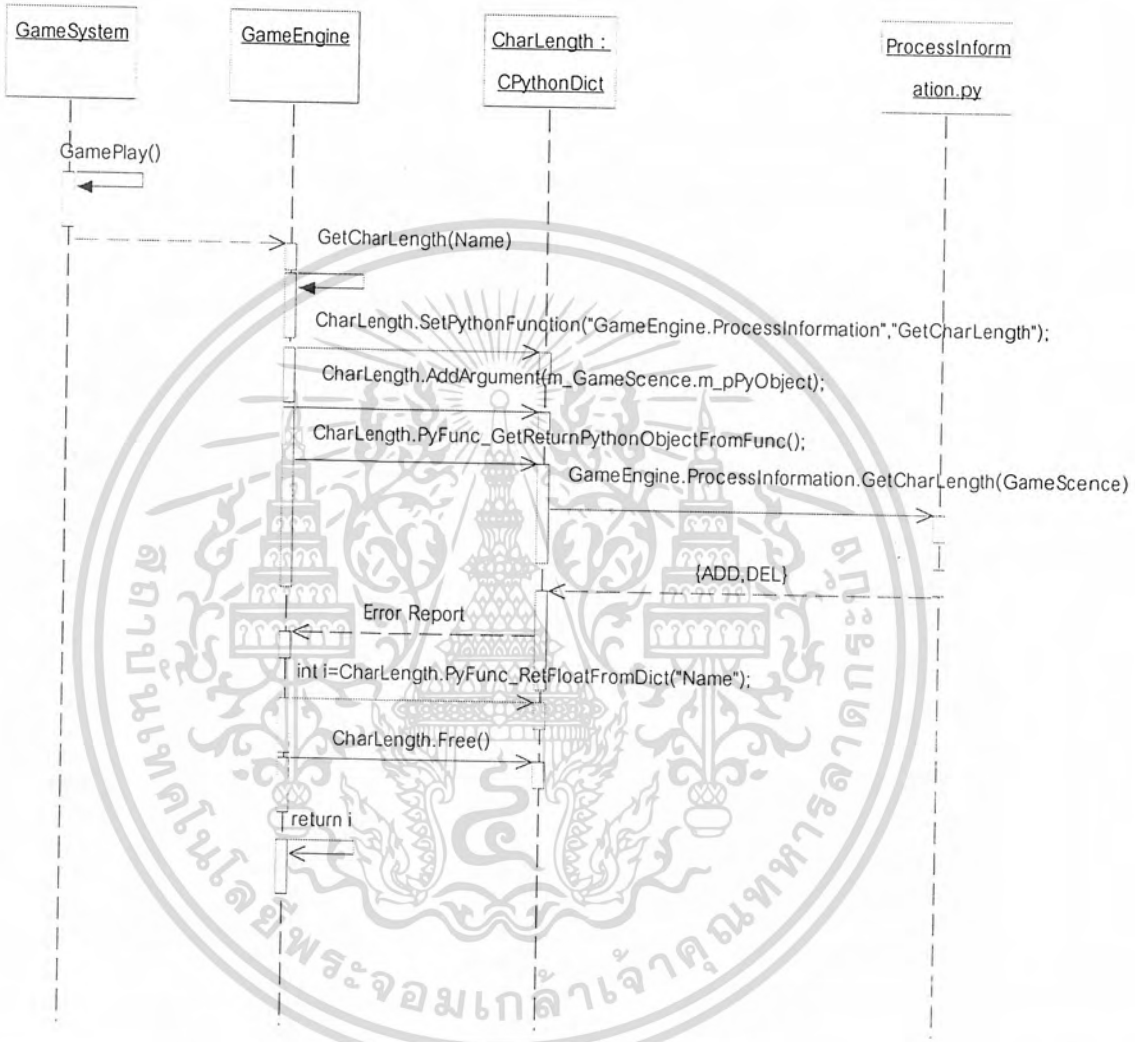
ไดอะแกรมนี้แสดงได้เฉพาะส่วนที่เป็นC++เท่านั้น แต่ในlayer CGameEngine นั้นยังประกอบไปด้วย ProcessInformationและFileManagerอีกซึ่งสามารถ เขียนลงในซีเควนท์ไดอะแกรมแบบบนลงล่าง แล้วถูกมองเป็น ตัวหลักที่เราพิจรณาได้ด้วย เพราะเป็นแอดทิบิว(attribute)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 ปฏิบัติการ (Implementation)&ทดสอบ(test)

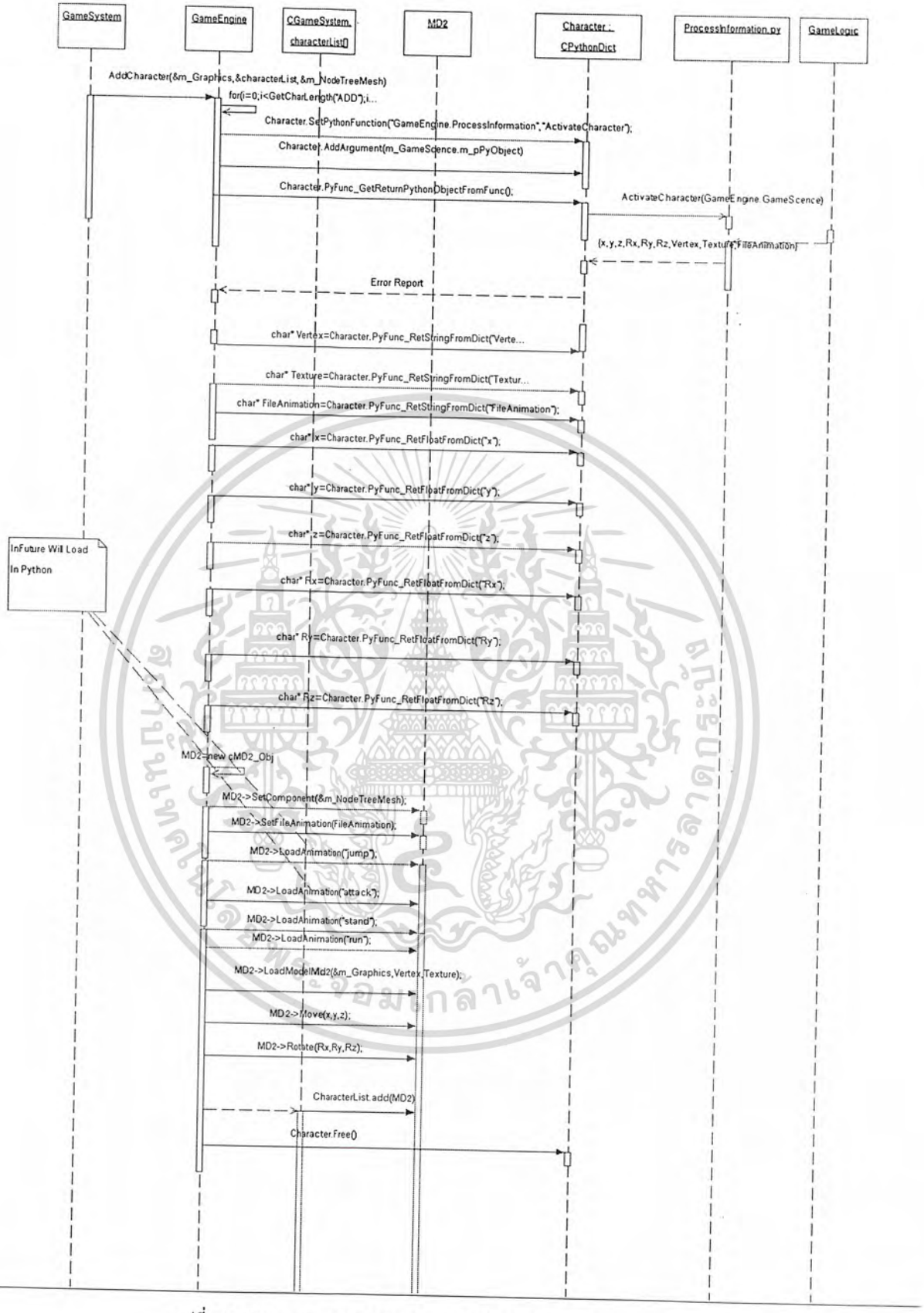
การปฏิบัติการจะใช้ซีเควนท์ไดอะแกรมแบบทอปดาวน์ดีไซน์ซินแทคก์ช่วยทำให้ก่อนจะเขียนโค้ดก็ร่างทุกอย่างไว้ในไดอะแกรม แล้วค่อยๆเปลี่ยนจนกระทั่งเสถียรตามหลักเพิ่มเติมด้วยการทำซ้ำ(iterative&incremental)

ภาพตัวอย่างการใช้งานซีเควนท์ไดอะแกรมแบบทอปดาวน์ดีไซน์ซินแทคก์



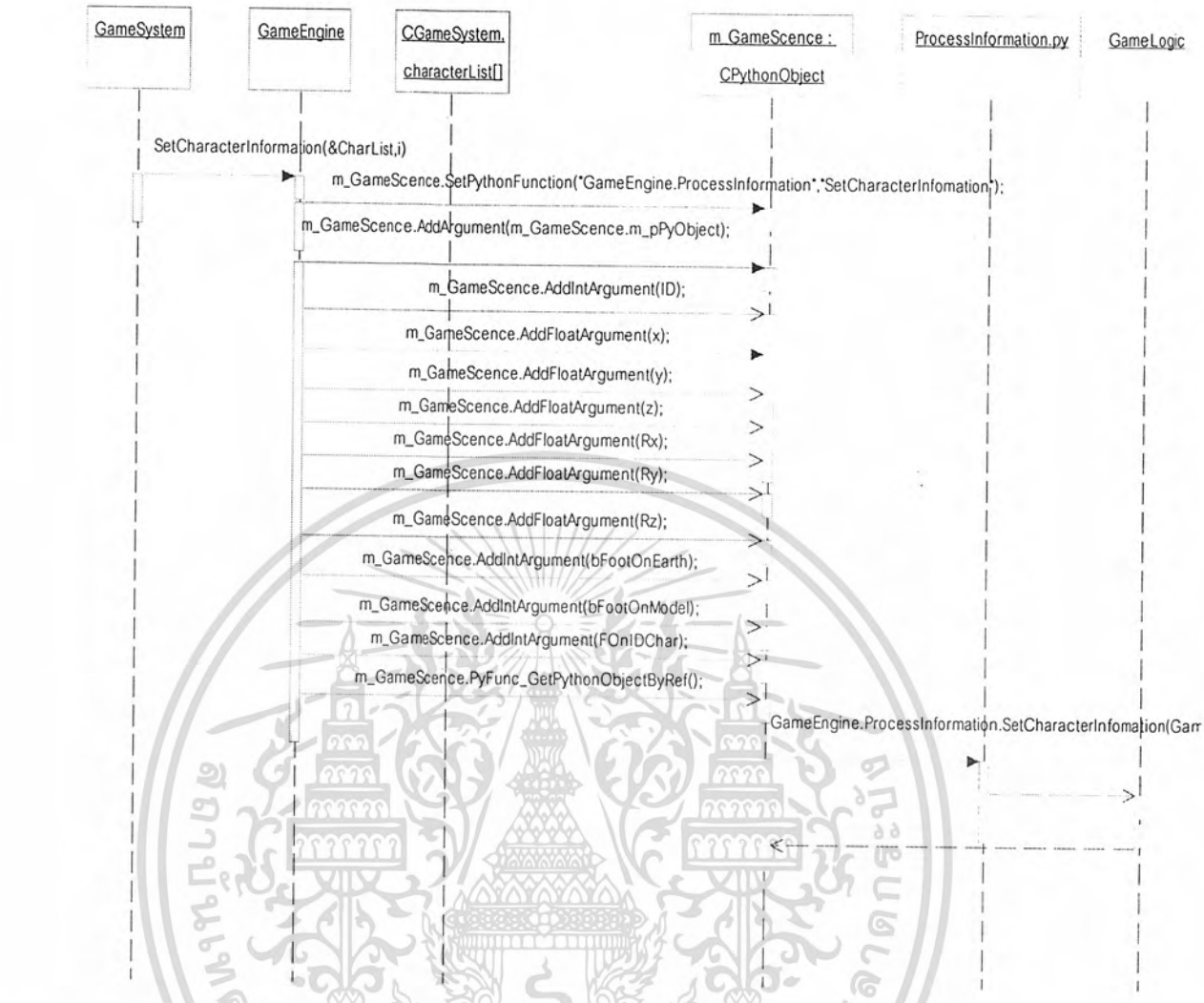
รูปที่3.21 CGameEngine3 (GamePlay)1(CharLength)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



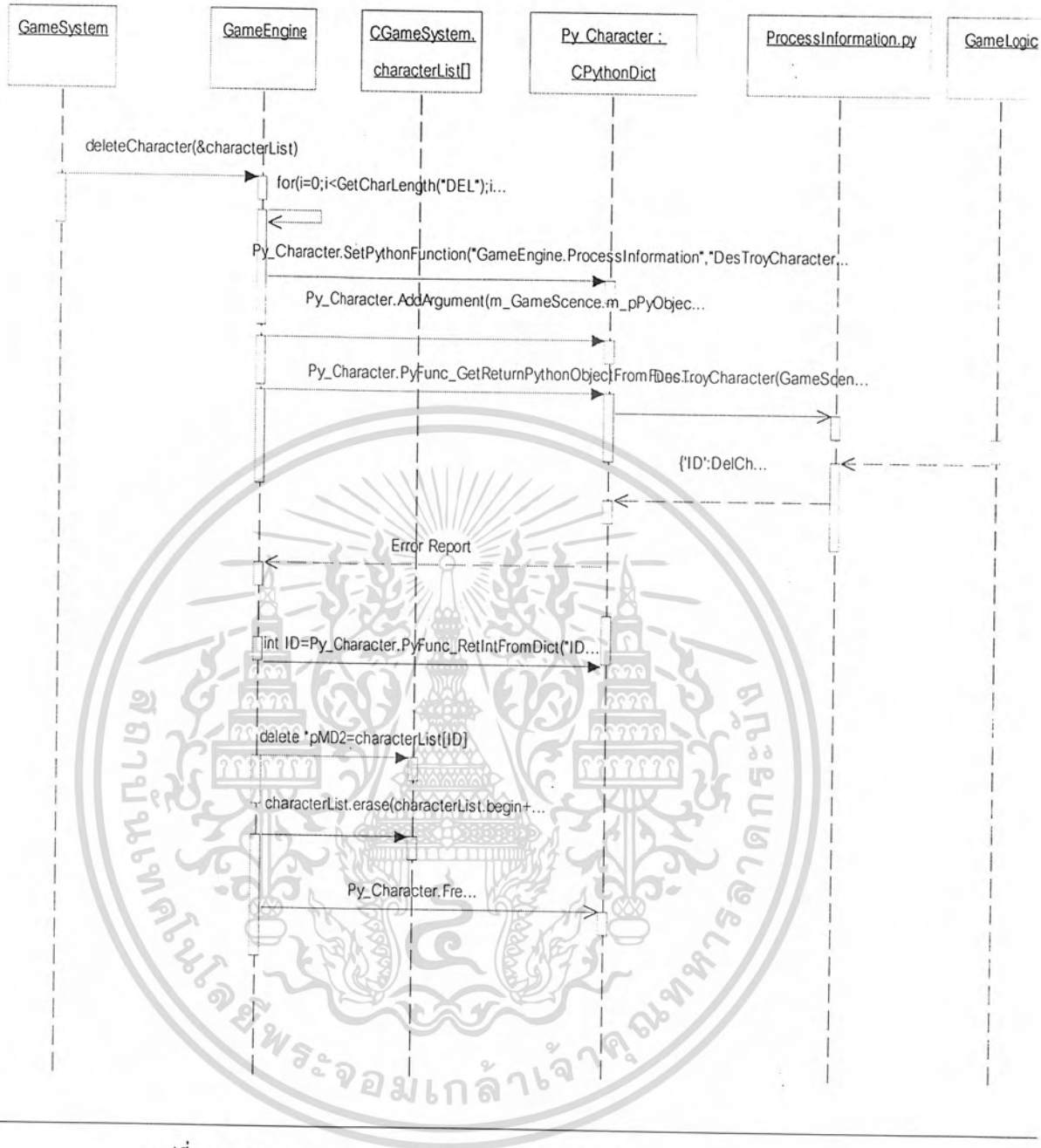
รูปที่3.22 CGameEngine3 (GamePlay)2(AddCharacter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



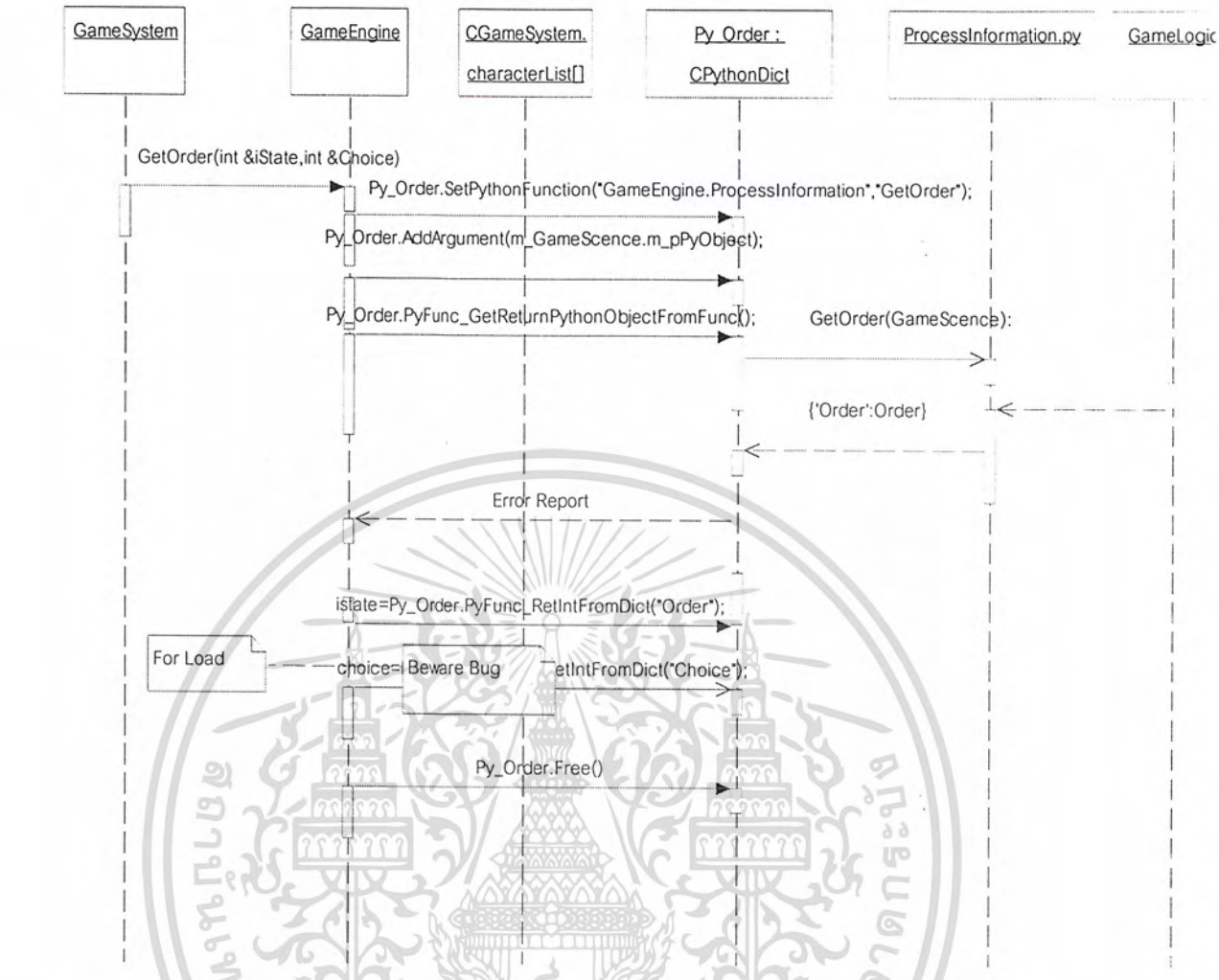
รูปที่3.23 CGameEngine3 (GamePlay)4(SetInformation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.24 CGameEngine3 (GamePlay)5(DeleteCharacter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.25 CGameEngine3 (GamePlay)6(GetOrder)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4

ผลการทดลอง

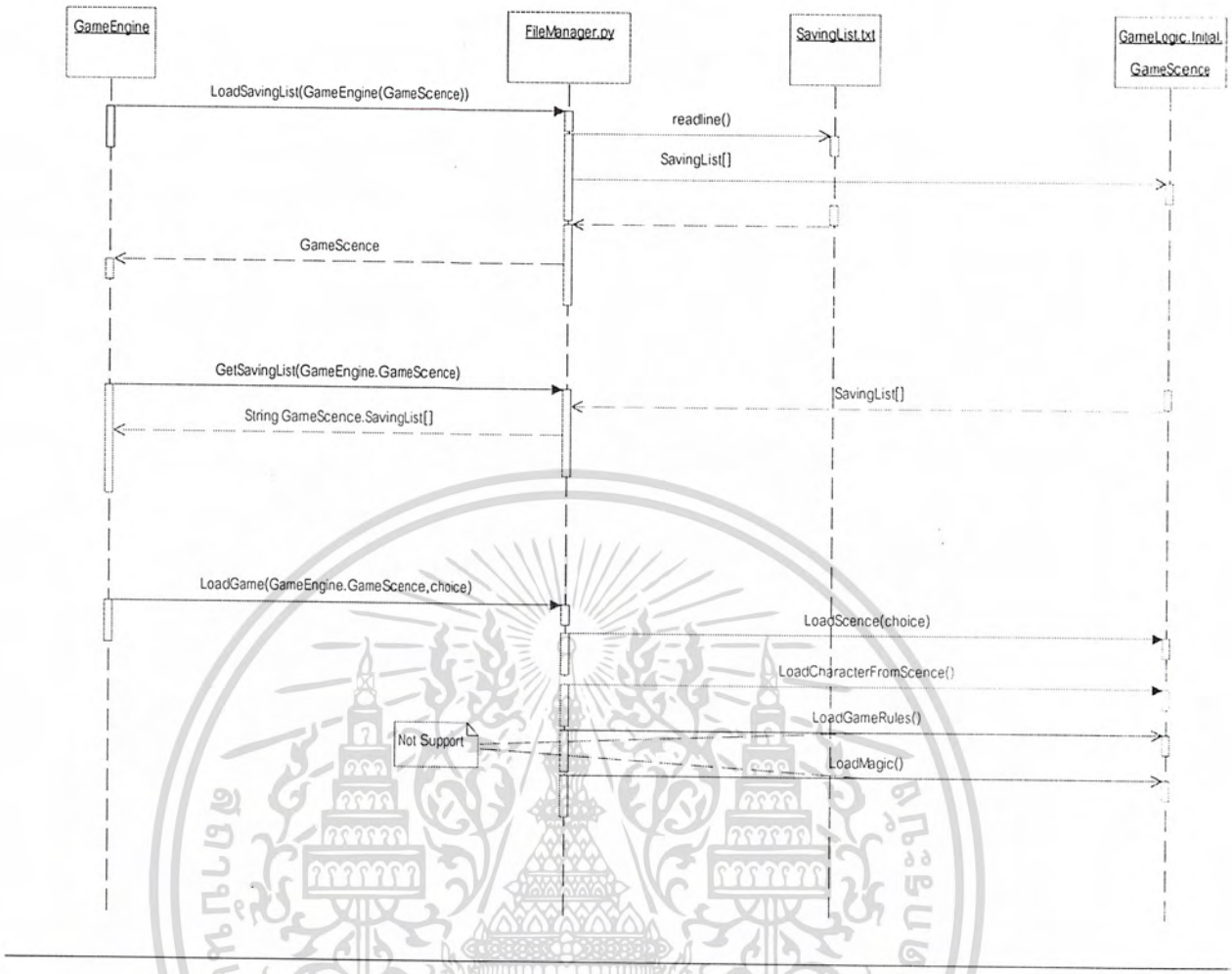
4.1ด้านซอฟแวรเอนจิเนียริ่ง

เนื่องจากในmicrosoft wordไม่สามารถทำให้เห็นภาพของไดอะแกรมได้ชัดเจนจึงต้องขอยกแค่บางไดอะแกรมมาพร้อมทำการตัดออกในส่วนที่ไม่จำเป็นต้องพิจารณา เพื่อนำมาแสดงให้ดูว่ามีการลงไปติดต่อข้ามระหว่างเลเยอร์หรือไม่ หากไม่มีการข้ามระหว่างเลเยอร์ถือว่า กระบวนการเพื่อพัฒนาเอนจินสำหรับเอดดูเทนเมนท์ประสบความสำเร็จ เพราะผู้พัฒนารายอื่นไม่จำเป็นต้องรู้วิธีการสร้างเกมอย่างลึกซึ้ง ในขณะเดียวกันที่ผู้เล่น เอดดูเทนเมนท์ก็ไม่จำเป็นต้องรู้หลักการเคลื่อนไหวของระบบเกม(Game System) ทำให้เขียนเพียงแค่AIผ่านAPIของเกมเอนจิน ก็ทำให้ตัวละครเคลื่อนที่ได้แล้ว ในที่นี้ได้ทดลองบังคำสั่งเอนจินด้วยคำสั่งMoveTo(point) ซึ่งจะทำให้วัตถุในเกมที่สามารถเขียนAI ได้เคลื่อนโดยไม่ต้องกำหนดท่าทางของตัวละคร(Animation)หรือองศาในการหมุน(rotate)ใดๆตามหลักการของdirect3D

<u>ที่เกมเอนจินเลเยอร์</u>

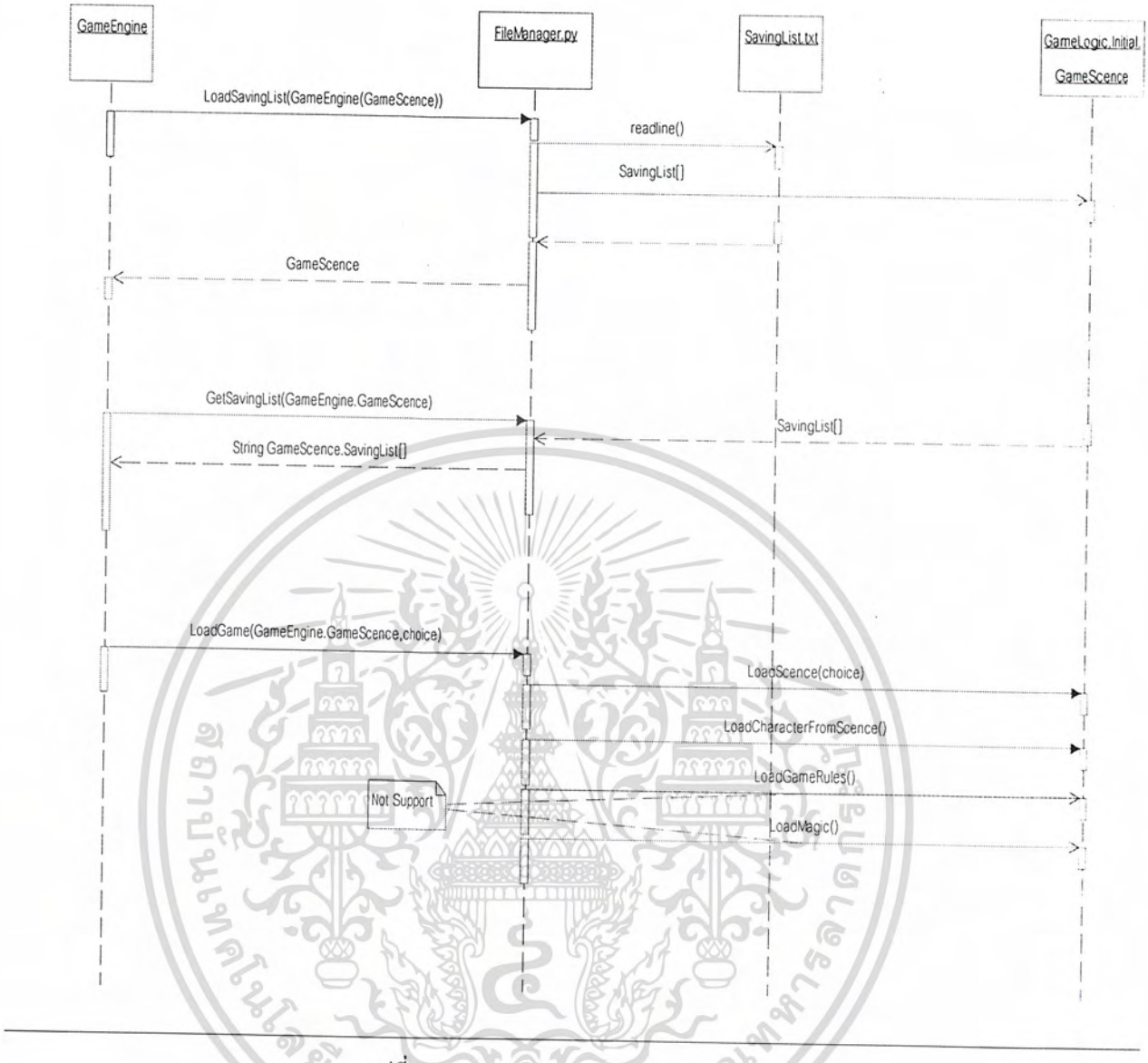
ซีเควนท์ไดอะแกรมแบบบนลงล่างสามารถลดความเข้มข้นของการทำเอกสารโดยทำให้อยู่ในรูปวิเคราะห์ได้ (Analysis) ซึ่งจะบันทึกรายละเอียดได้ไม่มากนัก แต่ก็พอทำให้เข้าใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



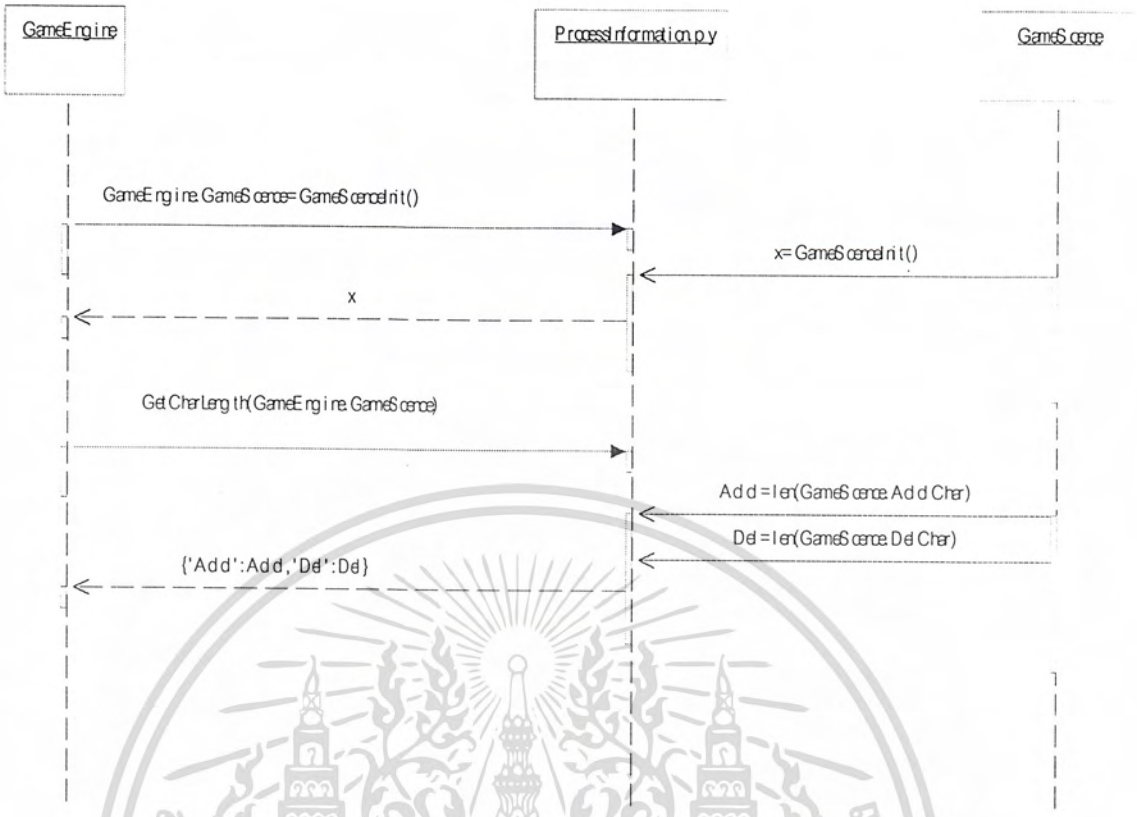
รูปที่4.1 Gameengine.__init__.py

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.2FileManager.py

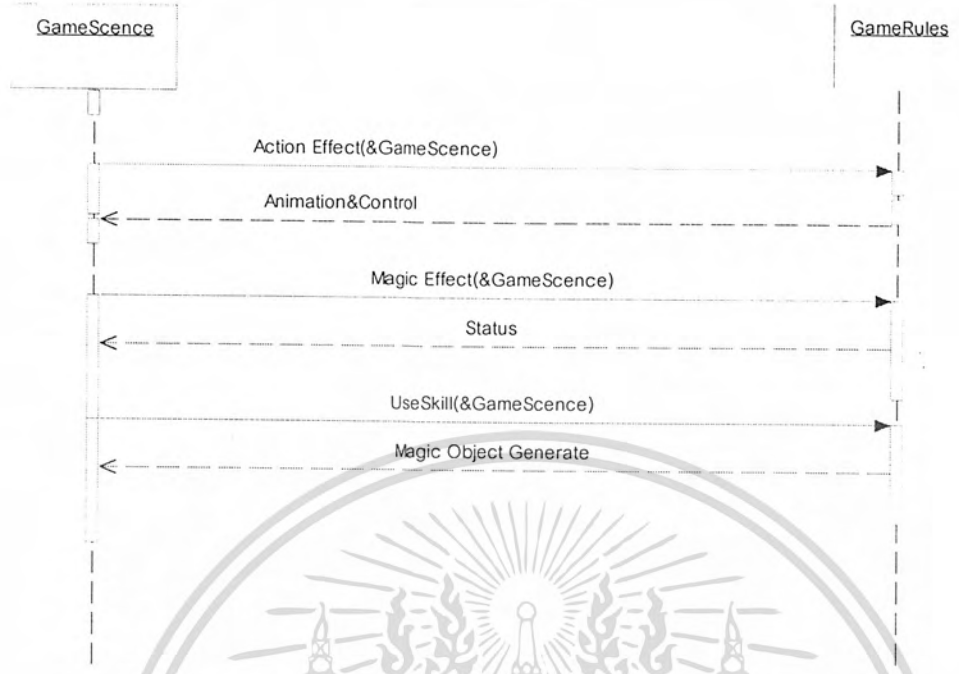
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



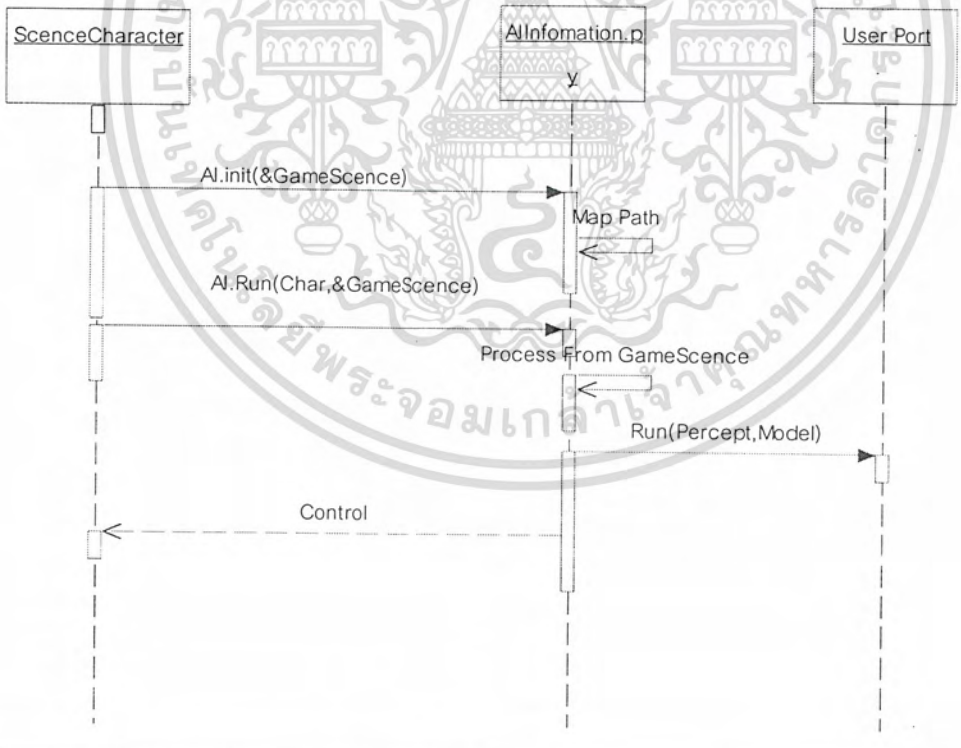
รูปที่4.3ProcessInformation.py

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เกมโลจิคเลเยอร์

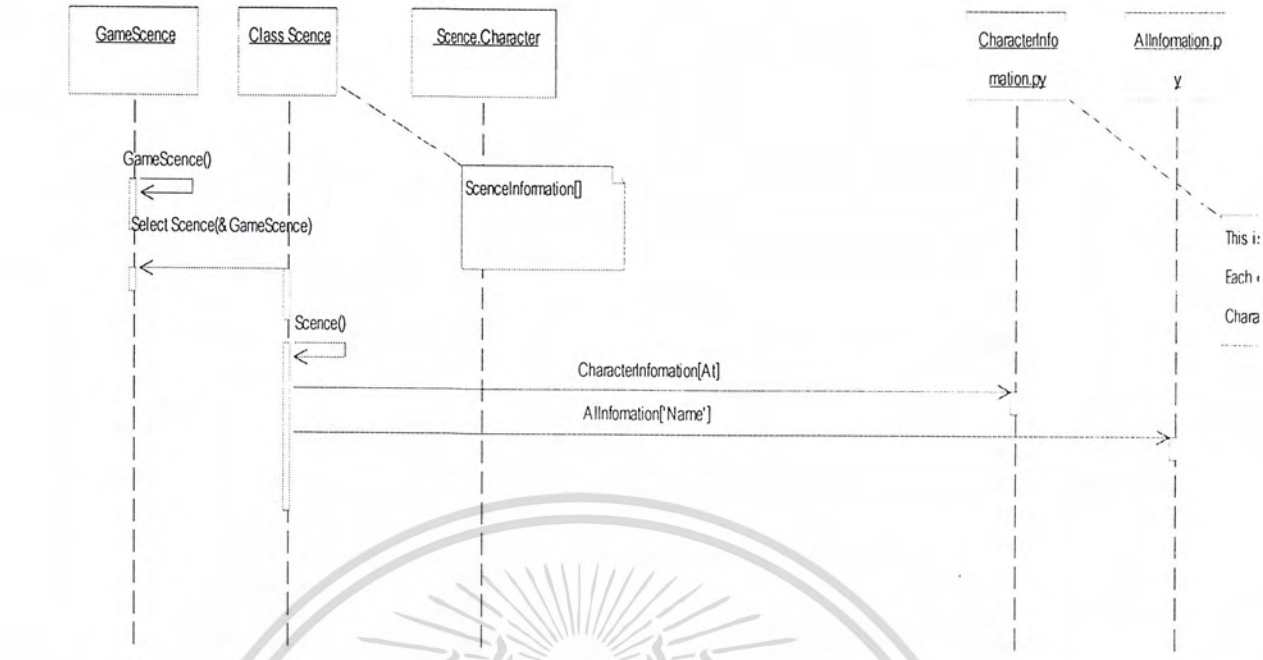


รูปที่4.4 GameRules

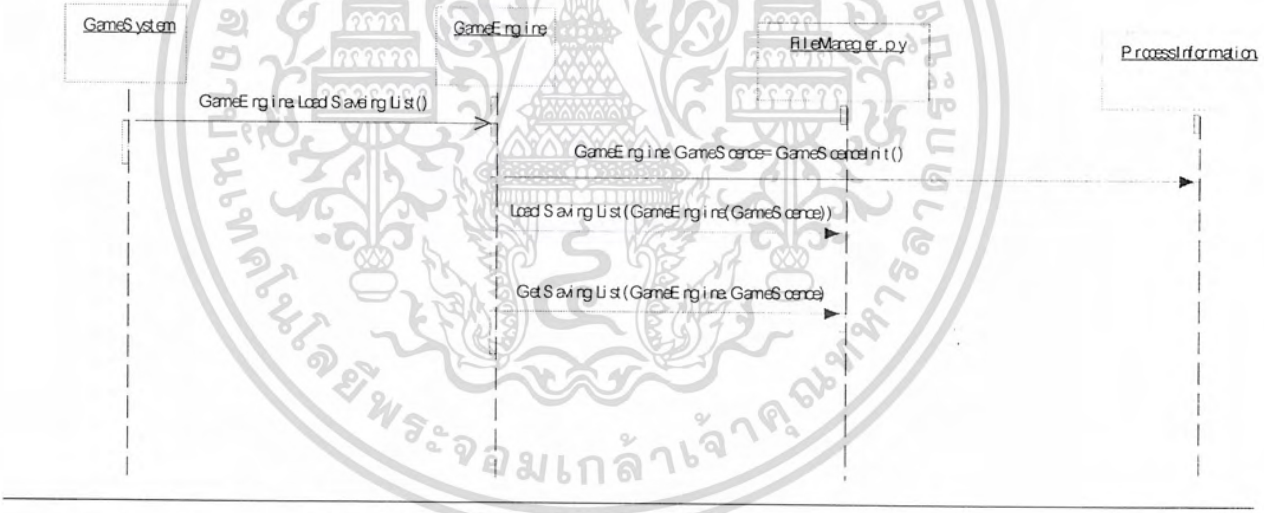


รูปที่4.5 AIInfomation.py

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.6 ScenceInformation.py



รูปที่4.7 ส่วนของGameEngineที่เป็น C++ ที่ทำหน้าที่ติดต่อกับGameSystemแบบโลจิคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ด้านผู้เล่นเอดูเทรนเมนท์

1 ผู้เล่นจะต้องเขียนโปรแกรมเป็นคลาสชื่อAIลงในไฟล์ที่ชื่อ AI1.py , AI2.py ของโฟลเดอร์(Folder) "USER"

2 ฟังกชัน__init__(self : AI) จะใช้เป็น ฟังกชันเรียกPathจากGetPath() จากlibrary เพื่อเก็บความจำของเส้นทางในแผนที่อันนั้น และอินนิทเทียล(initial)ตำแหน่งจุดเริ่มต้นที่เอเจนท์ยืนเริ่มต้นของแผนที่โดยใช้ฟังกชัน InitState()

3 ฟังกชันRun(self : AI, Percept : Position)ใช้ประมวลผลว่าจะเดินไปไหนโดยใช้คู่กับการรีเทรินaction จากMoveTo(point)ทุกครั้ง(ปัญหาคือการรีเทรินน์ ไม่ทำอะไรเลย(idle)ยังไม่ซับพอรต์เพราะต้องเอมเบดไว้ระดับสูงให้ผู้ใช้ไม่ต้องไปคำนวณการโรเตดและแอนนิเมชั่น)

4 ตัวแปรperceptจะเก็บค่าตำแหน่งปัจจุบันจากพื้นที่และอัพเดทให้ตลอดเวลาที่ตัวละครเดิน

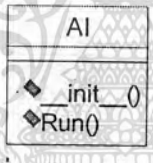
5 int InitState()จะเป็นค่าสถานะ(State)เริ่มต้นใช้กับOperator [ ]

6 Bool IsReachPoint(NextPoint,Percept)ใช้เช็กว่าถึงจุดถัดไปหรือยัง

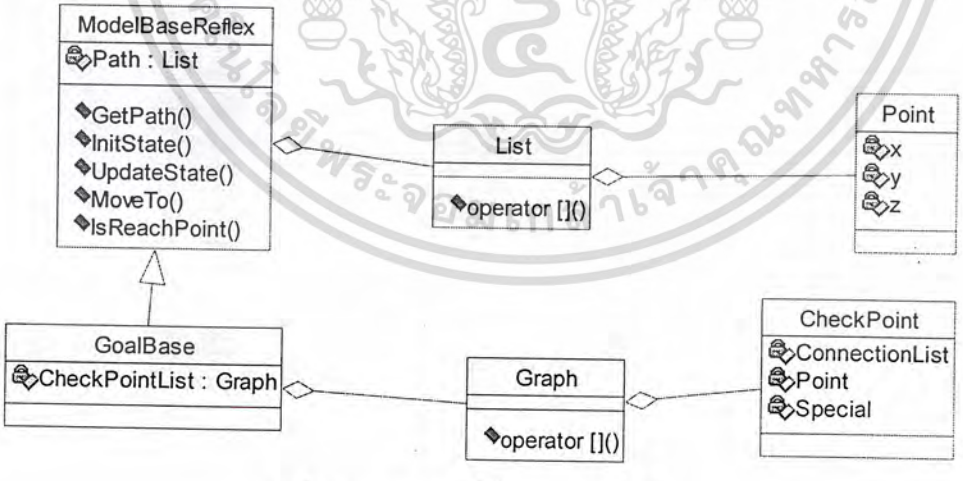
7 int UpdateState()จะเป็นตัวเปลี่ยนค่าเมื่อถึงผ่านจุดถัดไป

8 Path GetPath()จะreturn Pathออกมาซึ่งถ้าเป็นModelBase จะreturnออกมาเป็น Listของ point แต่ถ้าเป็นGoalBaseจะเป็นGraphของ CheckPoint

9 action MoveTo(point)ใช้return ให้เดินไปยังจุดหมาย



รูปที่4.8 คลาสAIที่จะเขียนลงไฟล์ AI1.py , AI2.py



รูปที่4.9 คลาสไลบรารี่ที่ผู้ใช้ต้องติดต่อด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในด่านที่ใช้ModelBase ใช้Path[CurrentState]return pointออกมาได้เลย

ในด่านที่ใช้GoalBaseควรมีฟังกชันที่ทำรีเดอรซีฟเพื่อหาGoal เพิ่มมาด้วย

CheckPointจะมีConnectionListที่เชื่อมไปCheckPointอื่นโดยนำค่าในCheckPoint.ConnectionList[i]ไปใส่ในPath[CheckPoint.ConnectionList[i]] เมื่อ Path=GoalBase.GetPath()และiคือindex=เริ่มจากinitial state Indexบอกตำแหน่งของมันบนpathนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5

สรุปผลการทดลอง

โครงงานการสร้างเกมเอนจินเอดดูเทรนเมนท์ครั้งนี้สามารถประสบความสำเร็จได้โดยใช้หลักการของเบสแพ็คทิสออฟซอฟท์แวร์เอนจิเนียริ่งซึ่งเป็น วิถีทางซอฟท์แวร์เอนจิเนียริ่งที่แนะนำให้สร้างสถาปัตย์กรรมขึ้นมา ในที่นี้เราใช้สถาปัตย์กรรมแบบชั้น และทำการออกแบบโดยใช้ซีเควนท์ไดอะแกรมแบบทอปดาวน์ดีไซน์เทียบกับระบบสถาปัตย์กรรมแบบชั้นได้ผลลัพธ์ที่ค่อนข้างสมบรูณ์ สามารถพิจารณาจากไดอะแกรม ข้อมูลของเกมทุกอย่างจะถูกเก็บในGameEngine.m_pPyObjectซึ่งก็คือออบเจกท์ GameEngine.GameScenceทำให้ในGame Logic layerถัดไปไม่มีการเรียกใช้งานฟังก์ชัน GameSystem layerเลยดังรูป แต่จะติดต่อผ่านเอนจินเฉพาะGameEngine.m_pPyObjectเท่านั้นแล้วให้เอนจินไปติดต่อกับระบบเกม(GameSystem)เองอีกที และการที่ผู้พัฒนาต่อไม่มีความจำเป็นต้องลงไปเกี่ยวข้องกับลำดับชั้น(layer) ที่ต่ำกว่า ทำให้ผู้ใช้งานในระดับเกมโลจิคไม่จำเป็นต้องมีความรู้direct-xเลยแม้แต่น้อย

สำหรับความสำเร็จของโครงงานในครั้งนี้ สามารถพิจารณาจากวัตถุประสงค์ดังนี้

<u>จากความมุ่งหมายและวัตถุประสงค์ของการศึกษา</u>

1 ศึกษาการนำทฤษฎีออปเจ็กท์โอเรนเตด(Object Oriented) มาใช้งานประยุกต์กับเกมเอนจิน

-ผ่านวัตถุประสงค์ โดยใช้เบสแพ็คทีสออฟซอฟท์แวร์เอนจิเนียริ่งและทอปดาวน์ดีไซน์ซินแทกค์

2 ศึกษาการสร้างทฤษฎีเกมเอนจิน

-ผ่านวัตถุประสงค์ สร้างได้โดยใช้สถาปัตย์กรรมแบบชั้น(layer Architecture)

3 ศึกษาการนำทฤษฎีเอไอมาใช้งานประยุกต์กับเกมเอนจิน

-ยังไม่ได้ผลเท่าที่ควร (รอวิจัยต่อระดับปริญญาโทต้องการพัฒนาlearning agent)

4 เพื่อการวิจัยและสร้างเอดดูเทรนเม้นท์ เกมแนวใหม่สู่ตลาดเกม

-ยังไม่ได้ผลเท่าที่ควร เพราะยังไม่มีระบบป้องกันbug(งบประมาณทางเวลาจำกัด)

-ไลบรารี่ยังคงใช้ยากเพราะงบประมาณทางเวลาจำกัด ทำให้คิดวิธีที่ใช้งานง่ายกว่านี้ไม่ทัน

5 เพื่อพัฒนาและปรับปรุงโมเดลทางซอฟท์แวร์เอนจิเนียริ่งให้มีความสามารถในการจัดเก็บเอกสารมากขึ้น

-ผ่านวัตถุประสงค์ ได้ผลค่อนข้างน่าพอใจ ได้ซีเควนท์แบบทอปดาวน์ดีไซน์ซินแทกค์แต่ยังไม่สมบรูณ์ เพราะยังไม่ได้วิจัยเต็มรูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่6

บรรณานุกรม (Bibliography)

1 เทคนิคด้านspecial effect อ้างอิงโดย Role Playing game programming ของJim Adams

2 บทความช่วงอายุที่สัมพันธ์กับการเรียนรู้อ้างอิงโดย Renu Ahuja, Sugata Mitra, Rashmi Kumar and Monica Singh "Education Through Digital Entertainment-A Structured Approach"*ปี-1994

*หมายเหตุ web เรื่องEdutainment มีการเปลี่ยนLinkเลยขอใช้วิธีsearchหาชื่อบทความเอานะครับ

3 บทความด้านedutainmentโดย J.Ajdari ,I.Aulaskari ,J.brown

"Edutainment in educational environments" **

Link => http://www.helsinki.fi/~jbrown/tao/rap7.html

4 The Unified Software Development Process

โดย Grady Booch, James Rumbaugh , Ivar Jacobson

6 ภาษาPython www.python.org

7 เรื่อง Structure of Agent อ้างอิงโดย

Artificial Intelligence A Modern Approachของ Stuart Russell & Peter Norvig

8 Learning Agent อ้างอิงโดยArtificial Intelligenceของ Michael Negnevitsky

9 Top-Down Design Syntax Sequence Diagram ของMr.Prachaya Puttapanasubกำลังอยู่ในระหว่างวิจัยอยู่ยังไม่มีการอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้