



การสร้างเกมสามมิติด้วยโปรแกรมยูนิตี้

หัวข้อ “ส่งเสริมการท่องเที่ยวลดโลกร้อน”

กรณีศึกษา บริษัท ซีจีโพลิกอน จำกัด

3D GAME CREATOR WITH UNITY PROGRAM

TITLE “LOW CARBON TOURISM”

A CASE STUDY OF CG POLYGON CO.,LTD.

นายภาณุวิชญ์ ศาสตร์เปล่ง

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาเทคโนโลยีมีเดีย

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

พ.ศ. 2556

การสร้างเกมสามมิติด้วยโปรแกรมยูนิตี้
หัวข้อ “ส่งเสริมการท่องเที่ยวลดโลกร้อน”

กรณีศึกษา บริษัท ซีจีโพลิกอน จำกัด

3D GAME CREATOR WITH UNITY PROGRAM

TITLE “LOW CARBON TOURISM”

A CASE STUDY : CG POLYGON CO.,LTD.

นายภาณุวิชญ์ ศาสตร์เปล่ง

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาเทคโนโลยีมีเดีย

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

พ.ศ. 2556

คณะกรรมการสอบ

..... ประธานกรรมการสอบ

(อาจารย์สลิลา ชีวกิตการ)

..... กรรมการสอบ

(อาจารย์อภิขญา นิ่มคุ้มภัย)

..... อาจารย์ที่ปรึกษา

(อาจารย์ภูวดล ศิริกองธรรม)

..... ประธานสหกิจศึกษาสาขาวิชา

(อาจารย์ภูวดล ศิริกองธรรม)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทย-ญี่ปุ่น

ชื่อโครงการ	การสร้างเกมสามมิติด้วยโปรแกรมยูนิตี้ หัวข้อ “ส่งเสริมการท่องเที่ยวลดโลกร้อน” กรณีศึกษา บริษัท ซีจีโพลิกอน จำกัด พ.ศ. 2556 A 3D Game creator with Unity Progame Title “Low Carbon Tourism” A Case Study : CG Polygon Co.,Ltd.
ผู้เขียน	นายภาณุวิชญ์ ศาสตร์เปล่ง
คณะวิชา	เทคโนโลยีสารสนเทศ สาขาเทคโนโลยีมัลติมีเดีย
อาจารย์ที่ปรึกษา	อาจารย์ภูวดล ศิริกองธรรม
พนักงานที่ปรึกษา	นายวัฒนสิน คชฤทธิ์
ชื่อบริษัท	บริษัท ซีจีโพลิกอน จำกัด
ประเภทธุรกิจ	ผลิตสื่อมัลติมีเดีย AR Code, 3D Model และ Mapping

บทคัดย่อ

จากการทำเกมด้วยโปรแกรมยูนิตี้ ที่ทางบริษัทได้มอบหมาย โดยมีหัวข้อว่า “ส่งเสริมการท่องเที่ยวลดโลกร้อน” ซึ่งต้องเน้นสถานที่ท่องเที่ยวเป็นหลัก ทำให้ต้องเลือกสถานที่สักแห่งโดยเลือกเป็นวัดเก่าของอยุธยา ซึ่งดูเหมาะแก่การอนุรักษ์ ซึ่งกำหนดแนวเกมไว้เป็นการปกป้องสถานที่และยังเป็นรูปแบบของบุคคลที่ 3 โดยมีเหล่ามัลพิชจะคอยมาโจมตีในแต่ละรอบ ซึ่งเราต้องกำจัดให้หมดโดยที่สถานที่ไม่ถูกทำลาย

จากการดำเนินการสร้างเกม และได้ลองทดสอบ พบว่าตัวเกมที่สร้างนั้นได้ทำให้ผู้เล่นรู้สึกต้องการปกป้องและอนุรักษ์สถานที่ ที่ถูกโจมตีจากเหล่ามัลพิช เป็นการจำลองสถานที่ของวัดเก่า อีกทั้งยังสามารถเล่นได้ทุกเพศทุกวัย ให้ความบันเทิงในรูปแบบแฟนตาซี

Summary

Project's name	A 3D Game creator with Unity Progame Title "Low Carbon Tourism" A Case Study : CG Polygon Co.,Ltd.
Writer	Mr. Panuwit Satpleng
Faculty	Information Technology, Multimedia Technology
Faculty Advisor	Mr. Puwadol Sirikongtham
Job Supervisor	Mr. Watsin Kotehsrit
Company's name	CG Polygon Co.,Ltd.
Business Type	Multimedia, AR Code, 3D Model and Mapping

Summary

according to Unity game project that my company assign. With the topic "traveling with encourage to decreasing Global Warming" It focus on Attractions. So I have to decide only one Attractions. It is Ayuthaya temple that look good enough to Conserve The game style is base-defence and Third-Person-Views. The old Temple Which will be attacked by The pollution. Our job is to destroy The pollution and the old temple still remaining.

According to our game making process and Testing. Not only I have found that this game make players want to protect and Conserve the Attractions. that will be attacked by The pollution. But also easy to play by anyone and entertain in Fantasy style.

TNI

- คลท่านอื่นๆ ที่ไม่ได้กล่าวนามทุกท่านที่ได้ให้คำแนะนำช่วยเหลือในการจัดทำรายงาน
ข้าพเจ้าใคร่ขอขอบคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่าน ที่ให้ข้อมูล เป็นที่ปรึกษาในการทำรายงานฉบับนี้
จนเสร็จสมบูรณ์ตลอดจนให้การดูแลและให้ความสนใจเกี่ยวกับชีวิตการทำงานจริง ข้าพเจ้า
□□ขอบคุณไว้ ณ ที่นี้

สารบัญ

หน้า

บทสรุป

ก

Summary

ข

กิตติกรรมประกาศ

ค

สารบัญ

ง

สารบัญตาราง

จ

สารบัญภาพประกอบ

ฉ

บทที่

1. บทนำ

1

1.1 ชื่อและสถานที่ตั้งของสถานประกอบการ

1

1.2 ลักษณะของสถานประกอบการ

2

1.3 รูปแบบของสถานประกอบการ

2

1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

3

1.5 พนักงานที่ปรึกษา

3

1.6 ระยะเวลาที่ปฏิบัติงาน

3

1.7 ที่มาและความสำคัญของปัญหา

3

1.8 วัตถุประสงค์ของการปฏิบัติงาน

4

1.9 ผลที่คาดว่าจะได้รับจากการปฏิบัติงาน

4

หรือโครงการที่ได้รับมอบหมาย

1.10 นิยามศัพท์

5

สารบัญ(ต่อ)

บทที่	หน้า
2. ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน	8
2.1 เครื่องมือที่ใช้ในการทำงาน	8
2.2 □□□□□	9
2.2.1 ทฤษฎีองค์ประกอบงานโมเดล	9
2.2.2 ท□□□□□□□□□	10
2.2.3 ทฤษฎีสี□□□□□□□	11
2.2.4 ท□□□อนาโตมี	12
2.2.5 ท□□□□□□□□□□□□□□	12
2.2.6 ท□□□□□□□□□□□□□□□□	14
2.3 โปรแกรมที่ใช้	28
2.3.1 Adobe Photoshop	28
2.3.2 Adobe Illustrator	29
2.3.3 Adobe After Effect	29
2.3.4 Zbrush	30
2.3.5 Unity	31
2.3.6 Autodesk Maya 2013	31
2.3.7 Unfold 3D	32
2.3.8 UV Layout	32

สารบัญ(ต่อ)

บทที่	หน้า
3. แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน	33
3.1 วัตถุประสงค์	33
3.2 รายละเอียดงานที่นักศึกษาปฏิบัติใน	34
3.2.1 โมเดล และ อนิเมชัน สมุดบัญชีที่สามารถเปิด-ปิด	35
3.2.2 โมเดล	35
3.2.3 โมเดล Miyabi	36
3.2.4 โมเดล	36
3.2.5 โมเดล	37
3.2.6 ทำโมเดล Miyabi	37
3.2.7 ทำโมเดล Miyabi	38
3.2.8 โมเดล Miyabi	38
3.2.9 โมเดลเครื่องดนตรี vigo	39
3.2.10 โมเดลปล	39
3.2.11 โมเดลคอ	40
3.2.12 จัดแสงในห้องล๊อบบี้	40
3.2.13 แผนที่คอนโดมิเนียมชั้น	41
3.2.14 หนังสือ	41
3.2.15 จัดหนังสือการ์ตูนของค่ายบรรลือสาร	42
3.2.16 ทำหน้าเมนูด้วย NGUI	42
3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือ	43
3.3.1 การวางเนื้อเรื่อง	44
3.3.2 การวาง	44

สารบัญ(ต่อ)

บทที่	หน้า
3.3.3 การออกแบบระบบการดำเนินงาน	44
3.3.4 การดำเนินงานที่ปฏิบัติ	44
3.4 การประเมินผล	45
4. ผลการดำเนินงานการวิเคราะห์และสรุปผลต่างๆ	46
4.1 ขั้นตอนและผลการดำเนินงาน	46
4.1.1 รวบรวมข้อมูลและดูตัวอย่างเบื้องต้น	46
4.1.2 วางโครงเรื่องเบื้องต้น	47
4.1.3 สร้างโมเดลตัวละคร	47
4.1.4 สร้างตัวละคร AI	49
4.1.5 สร้างฉาก	51
4.1.6 สร้างสิ่งของ	53
4.1.7 การกางผิวโมเดล	54
4.1.8 การทำสีและตา	54
4.1.9 การทำอนิเมชัน	56
4.1.10 การ Paint Skin Weight	58
4.1.11 การคำนวณพื้นผิวโมเดล	59
4.1.12 การย้ายไฟล์โมเดลลงยูนิตี้	59
4.1.13 การจัดสถานที่และสภาพแวดล้อม	60
4.1.14 การเขียนโค้ดควบคุม	61

สารบัญ(ต่อ)

บทที่	หน้า
4.1.15 การเจ็ Network	62
4.1.16 การเขียน โค้ด Random Item	63
4.1.17 การเจ็	64
4.1.18 การทำเอฟเฟค	66
4.1.19 การสร้างหน้าเมนูหลัก	67
4.2	68
4.3 วิเค ผลที่ได้รับ	69
5. บทสรุปและข้อเสนอแนะ	71
5.1 ส	71
5.2 แ	71
5.3 ขั	72
เอกสารอ้างอิง	73
ภาคผนวก	74
ก การใช้โปรแกรมAutodesk Maya เบื้องต้น	75
ข บทเกมการเล่น การจัดลำดับของการเล่นเกม	97
ประวัติผู้จัดทำโครงการ	100

สารบัญตาราง

ตารางที่

หน้า

3.1 □□□□□□□□□□□□

33

4.2 ผลการวิเคราะห์ข้อมูล

68



สารบัญภาพ

ภาพที่	หน้า
1.1 ภาพประกอบ ภาพประกอบ ภาพประกอบ	1
1.2 แผนที่ตั้งบริษัท CGpolygon co.,ltd	1
1.3 แผนที่ประกอบประกอบประกอบประกอบประกอบ	2
2.1 Notebook คอมพิวเตอร์ PC	8
2.2 ตัวอย่างพื้นผิว	9
2.3 ภาพประกอบ UV Mapping	9
2.4 ภาพประกอบประกอบประกอบประกอบประกอบ	10
2.5 ภาพประกอบประกอบประกอบประกอบประกอบ	11
2.6 ทฤษฎีอนุกรม	12
2.7 Logo Adobe Photoshop	28
2.8 Logo Adobe Illustrator	29
2.9 Logo Adobe After effect	29
2.10 Logo Zbrush	30
2.11 Logo Unity	31
2.12 Logo Autodesk Maya	31
2.13 Logo Unfold 3D	32
2.14 Logo UV Layout	32
3.1 ภาพประกอบประกอบประกอบประกอบประกอบ	35
3.2 ภาพประกอบประกอบประกอบประกอบประกอบ	35
3.3 ภาพประกอบประกอบประกอบประกอบประกอบ	36
3.4 ภาพประกอบประกอบประกอบประกอบประกอบ	36
3.5 ภาพประกอบประกอบประกอบประกอบประกอบ	37

สารบัญภาพ(ต่อ)

ภาพที่	หน้า
3.6 ภาพประกอบ	37
3.7 ภาพประกอบ	38
3.8 ภาพประกอบ	38
3.9 ภาพประกอบเครื่องยนต์	39
3.10 ภาพประกอบ	39
3.11 ภาพประกอบ	40
3.12 จัดแสงโมเดลล้อบับ	40
3.13 แผนที่คอนโดมิเนียมชั้น	41
3.14 ภาพประกอบ	41
3.15 ภาพประกอบ	42
3.16 ภาพ NGUI	42
3.17 ภาพประกอบ	43
4.1 ภาพประกอบ VDO ที่ใส่	46
4.2 ภาพประกอบที่เลือก	47
4.3 โมเดลตัวละครที่สร้างขึ้น	48
4.4 ภาพประกอบ	48
4.5 ภาพประกอบ	49
4.6 ภาพประกอบ AI ชนิดที่ 1	49
4.7 ภาพประกอบ AI ชนิดที่ 2	50
4.8 ภาพประกอบ AI ชนิดที่ 3	50
4.9 ภาพประกอบ AI ชนิดที่ 4	51
4.10 ตัวอย่างสถานที่ ที่นำมาใช้สร้างฉาก	51

สารบัญภาพ(ต่อ)

ภาพที่	หน้า
4.11 ภาพที่แสดงการตั้งค่า UV Mapping	52
4.12 ภาพที่แสดงการตั้งค่า UV Mapping	52
4.13 สิ่งของภายในฉาก	53
4.14 ภาพที่แสดงการตั้งค่า UV Mapping	53
4.15 ภาพที่แสดงการตั้งค่า UV Mapping	54
4.16 ภาพที่แสดง Texture Mapping	55
4.17 ผลที่ออกมาจากกล้องหรือ Texture Mapping UV	55
4.18 ภาพที่แสดง Bump	56
4.19 ภาพที่แสดง Bump ที่ได้ออกมา	56
4.20 การทำอนิเมชันตัวละคร	57
4.21 ตัวอย่างอนิเมชัน	57
4.22 IK ที่ใช้เคลื่อนไหวของตัวละคร	58
4.23 ภาพ Paint Skin Weight ภาพที่ออกมาจากกล้องหรือ Texture Mapping UV	58
4.24 พื้นผิวโมเดล	59
4.25 ภาพ Export Model ภาพ Maya ภาพ FBX	60
4.26 ภาพ FBX ที่ได้จาก maya พร้อมอนิเมชัน	60
4.27 ภาพที่แสดงการตั้งค่า UV Mapping	61
4.28 ภาพที่แสดงการตั้งค่า UV Mapping	61
4.29 ภาพที่แสดงการตั้งค่า UV Mapping AI	62
4.30 ภาพที่แสดงการตั้งค่า UV Mapping Network Games	62
4.31 ผลที่ได้ออกมาจาก Network Games	63
4.32 ภาพที่แสดงการตั้งค่า UV Mapping	63
4.33 ไอเทมที่ถูกสุ่มออกมาตามจุด	64

สารบัญภาพต่อ)

ภาพที่	หน้า
4.34	64
4.35	65
4.36 ตัวอย่างการแสดงผลเลียดเหลือครึ่ง	65
4.37 ตัวอย่างการแสดงผลเลียดลดเกินครึ่ง	65
4.38	66
4.39	66
4.40 เปลี่ยน Texture GUI	67
4.41	67
4.42 หน้าเมนูหลักของเกม โดยเมื่อมีผู้เข้าเล่น 1 คน จะมีส่วนของการเข้าร่วมหลายคนขึ้นมา	70
4.43 ภายในเกมตอนเริ่มต้น	70

TNI

บทที่ 1

บทนำ

1.1 ชื่อและที่ตั้งของสถานประกอบการ



ภาพที่ 1.1 โลโก้บริษัท ซีจีโพลีกอน จำกัด

ชื่อสถานประกอบการ : CG polygon Co.,LTD (บริษัท ซีจีโพลีกอน จำกัด)
 ที่ตั้งของสถานประกอบการ : 1/862 การ์เด็นโฮมพลาซ่า โซน8 ถนนพหลโยธิน 60
 ตำบลคูคต อำเภอลำลูกกา ปทุมธานี 12130
 โทรศัพท์ : 080-264-3336
 E-mail : cgpolygon@gmail.com
 Website : www.mesise.com , www.cgpolygon.com



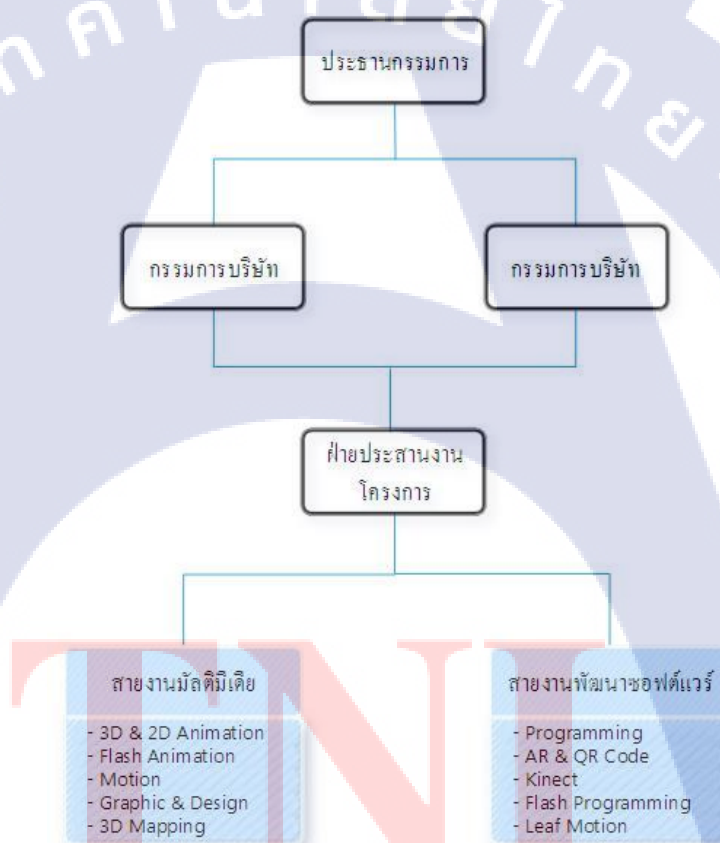
ภาพที่ 1.2 แผนที่ตั้ง CGpolygon co.,ltd

1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร

CGpolygon เป็นบริษัทรับผลิตงาน ARCode, Model 3D, 3D Mapping, Kinect และงานด้านอนิเมชั่น โดยเปิดรับงานด้านการผลิตโมเดลหลากหลายรูปลงใน AR Code ตั้งแต่การสร้าง 3D Model, 3D Background, Storyboard, Layout, Texture, Lighting, Animation, Video Mapping ไปจนถึง Programming

1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร

แผนผังโครงสร้างภายในบริษัท



ภาพที่ 1.3 แผนผังโครงสร้างภายในบริษัท

1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

ตำแหน่ง : พนักงานฝ่าย 3D Model

หน้าที่ : 3D Modeling, 3D Mapping, UV Mapping ,Texturing, Lightning, 3D Animation

1.5 พนักงานที่ปรึกษา และ ตำแหน่งของพนักงานที่ปรึกษา

พนักงานที่ปรึกษา : คุณวัฒน์สิน คชฤทธิ์

ตำแหน่ง : 3D

1.6 ระยะเวลาที่ปฏิบัติงาน

วันที่ 3 มีนาคม 2566 ถึง 4 มีนาคม 2566 รวมระยะเวลาทั้งหมด 4 วัน

1.7 ที่มาและความสำคัญของปัญหา

ที่มา : เนื่องจากในช่วงเวลานั้นได้มีโปรเจกต์ทำหนังสือเรื่อง “Low Carbon” การท่องเที่ยวเชิงอนุรักษ์ ทำให้ต้องเลือกที่จะทำเป็น อนิเมชั่น หรือ เกม ซึ่งได้เลือกที่จะทำเป็นเกม เนื่องจากเป็นสิ่งที่น่าลอง และมีเวลาให้ฝึกและเรียนรู้จึงได้เริ่มทำเกม

เหตุผล : เนื่องจากโปรแกรมที่ใช้ทำ มีความโดดเด่นในยุคเวลานี้ ซึ่งสามารถเขียนเกม หรือ โปรแกรมได้ในตัวเองและมีความต้องการจะศึกษา และบริษัทนี้ก็รองรับการทำงานกับโปรแกรมที่สนใจพอดีจึงได้เริ่มทำ

1.9 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย

- 1.8.1 เพื่อให้ผู้เล่นรู้สึกสนุกโดยมีและมีข้อคิดเชิงปกป้องสถานที่
- 1.8.2 ดึงดูดความสนใจของเด็กๆ ให้รู้จักอนุรักษ์สถานที่
- 1.8.3 ฝึกความอดทน
- 1.8.4 เพื่อใช้เกมให้เป็นประโยชน์ต่อสังคมมากกว่าความบันเทิง
- 1.8.5 เพื่อให้สังคมยอมรับว่าเกมมีบทบาทต่อสังคมในด้านอื่นๆ
- 1.8.6 เพื่อศึกษาการสร้างเกม

1.9 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย

- 1.9.1 ได้รับ..... ตรงที่ในห้องเรียน และอาจารย์ก็ไป.....
- 1.9.2 พัฒนาไปในทางด้านที่ชอบและสนใจ จ.....
- 1.9.3 เรียนความรู้จากสถานที่จริง ทำงานจริง
- 1.9.4 ความสำคัญของทีม พวกพ้อง ในการทำงานที่มีรายละเอียดสูง การส่งต่องาน
- 1.9.5 สามารถใช้โปรแกรมได้ไวและมากขึ้น
- 1.9.6 ตัวผู้เล่นจะรู้จักอนุรักษ์ รักษาสถานที่ท่องเที่ยว
- 1.9.7 เปรียบเทียบสถานการณ์ความเสื่อมโทรมในปัจจุบัน
- 1.9.8 ศึกษา..... Unity Code
- 1.9.9
- 1.9.10 เรียนรู้สิ่งใหม่

1.10 เทคนิคการเรนเดอร์

- Animation** : การสร้างภาพเคลื่อนไหว โดยการนำภาพนิ่งหลายๆภาพที่มีต่อเนื่อง มาฉายด้วยความเร็วที่เหมาะสม ทำให้เกิดภาพลวงตาของการเคลื่อนไหว
- AR Code** : Augmented Reality คือการนำเทคโนโลยีคอมพิวเตอร์กราฟิกส์มาใช้ ซึ่งต้องมีรูปเป็นสื่อกลางเหมือน QR Code หรือ AR Code เพื่อใช้ในการแสดงข้อมูล 3 มิติ
- Mapping** : ซึ่งก็คือการฉายภาพจากโปรเจกเตอร์ ลงบนวัตถุ ซึ่งอาจจะเป็นรูปทรงเรขาคณิต หรือรูปทรงอิสระก็ได้ ซึ่งการเคลื่อนไหว เป็นการเพิ่มสีสันให้กับตัวสินค้า
- UV Mapping** : คือการนำภาพ 2 มิติมาฉายลงบนวัตถุ 3 มิติ Map เพื่อโยงแกน 2 มิติ (U V) ไปกับแกน 3 มิติ (X Y Z) ซึ่งการนำภาพ 2 มิติมาฉายลงบนวัตถุ 3 มิติหนึ่งๆอาจใช้ UV ได้หลายรูปแบบขึ้นกับความต้องการ โดยทั่วไปมักใช้ใน 2 มิติลงบนพื้นผิว 3 มิติ texture mapping UV ก็มีหน้าที่ใช้ถ่ายโอนข้อมูลจากแบบจำลองสามมิติลงบนพื้นผิว 2 มิติ ซึ่งการนำภาพ 2 มิติมาฉายลงบนวัตถุ 3 มิติ
- Blend Shape** : คือการนำภาพ 2 มิติมาฉายลงบนวัตถุ 3 มิติ ซึ่งการนำภาพ 2 มิติมาฉายลงบนวัตถุ 3 มิติ
- Texture Mapping** : คือการนำ UV map หรือการหาความสัมพันธ์อื่น ๆ เพื่อคำนวณว่า face (3 มิติ) จะนำ texture 2 มิติ ซึ่งนี่คือค่าสีบน Texture map เป็นค่าสีจริงที่จะนำไปเรนเดอร์

Texture : รูปภาพหรือสี ที่ใส่ให้กับวัตถุเพื่อให้รู้ว่าวัตถุนั้นคืออะไร

IK : การใส่กระดูกให้กับโมเดลให้สามารถเคลื่อนไหวได้

Normal mapping : เป็นเทคนิคที่ใช้ Texture Map เพื่อสร้างภาพลวงตาของพื้นผิว โดยจะทำการแปลงค่าสีใน Texture Map ให้เป็นค่า Normal Vector ซึ่งจากหลักการนี้ทำให้เราสามารถสร้างภาพลวดลายต่างๆ บนพื้นผิวได้โดยไม่ต้องใช้ Geometry มากเกินไป เพื่อช่วยให้โมเดลสามมิติที่มีความละเอียดต่ำดูเหมือนมีรายละเอียดมากขึ้น ทั้งที่มีจำนวนโพลีกอนเท่าเดิม

Paint weight : เป็นการกำหนดพื้นที่ให้กับ Joint IK ที่เป็นกระดูกของโมเดล โดยการขยับ โดยที่จะช่วยไม่ให้ส่วนที่ขยับเสียรูปทรงหรือบิดเบี้ยวไปจากเดิม

Key Frame : Frame ที่มีวัตถุหรือมีการเปลี่ยนแปลง เราสามารถสังเกตได้จากช่องซึ่งจะมีช่วงเฟรมให้เลือกจะมีจุดที่จุดเริ่มต้นของ Animation Keyframe และจุดสิ้นสุดของ Animation Keyframe ที่ได้จึงจะมีชื่อที่ติดข้องการ

Extrude : การดึงเพื่อเพิ่มจำนวน เส้น และพื้นผิวของวัตถุ

Duplicate : คัดลอกหรือก๊อปปี้โมเดลชิ้นที่มีอยู่ออกมาเป็นอีกชิ้น

Edge : เส้นเชื่อมระหว่างวัตถุ

Face : พื้นผิวของวัตถุ

Combine : การรวมกลุ่มของวัตถุที่เลือกกันให้อยู่ในกลุ่มเดียวกัน

Separate : การแยกวัตถุที่ถูก combine กันออกจากกัน

Merge	:	การเชื่อมจุดของวัตถุซึ่งต้องเป็นวัตถุชิ้นเดียวกันเท่านั้น
Polygon	:	รูปร่างแบบปิดในพื้นที่ 3 มิติซึ่งมีหลายเหลี่ยม
Image Plane	:	การนำภาพที่มีเข้ามาเป็นแบบ
Smooth	:	การเพิ่มเส้นเพื่อไม่ให้วัตถุเป็นเหลี่ยม
Bind skin	:	IK ใช้ในการเคลื่อนไหวสำหรับทำอนิเมชัน
Parent	:	เป็นคำสั่งที่สั่งให้วัตถุย่อยตามวัตถุหลักในการเคลื่อนไหว
Vertex	:	จุดที่อยู่ตามมุมขอบของวัตถุ
Terrain	:	พื้นสำหรับวางวัตถุเพื่อใช้ในการทำงาน
Prefab	:	ค่าของวัตถุที่พร้อมถูกเรียกใช้งาน สามารถเรียกใช้งานได้ซ้ำๆ
GUI	:	Graphical User Interface

บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.1 เครื่องมือที่ใช้ในการทำงาน



ภาพที่ 2.1 Notebook และ PC

คอมพิวเตอร์ หรือ โน้ตบุ๊ก ส่วนตัวเพื่อการทำงาน และต้องมีการรองรับโปรแกรมที่ใช้งานภายในบริษัทในรูปแบบต่างๆ ซึ่งต้องแรงพอสมควร เช่น สเปคเครื่องที่ใช้

CPU : Core I5 2500K

Ram : Gskill 4 GB

Window : 7 64bit

Mainboard : Maximus 4 GEN E

NVIDIA GeForce GTX560 Ti

2.2 ทฤษฎี

2.2.1 ทฤษฎีองค์ประกอบงานโมเดล (Element of Model)

2.2.1.1 พื้นผิว (Texture)

งานโมเดล พื้นผิวเป็นหนึ่งในองค์ประกอบที่สำคัญที่สุดที่สื่อความหมายของงานได้ชัดเจน และบ่งบอกถึงวัตถุชิ้นนั้นๆ เช่น วัตถุที่มีพื้นผิวมันเงา อย่างเหล็ก



ภาพที่ 2.2 ตัวอย่างพื้นผิว

2.2.1.2 การทำ Mapping โมเดล

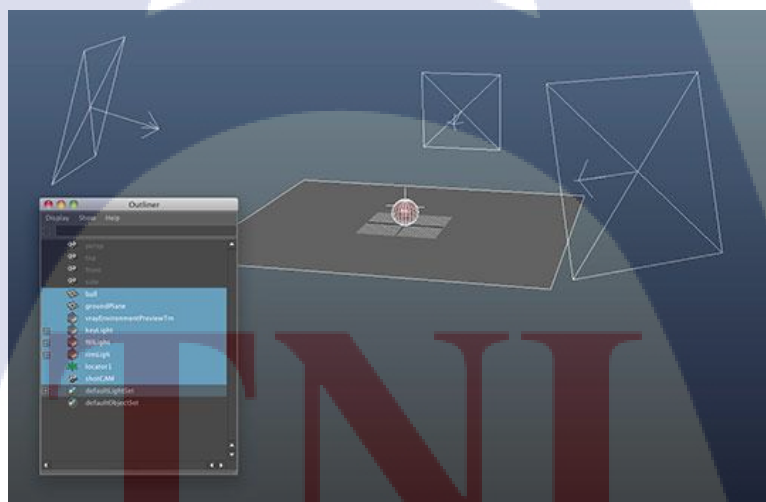
เป็นอีกส่วนหนึ่งที่สำคัญที่สุดของการใส่พื้นผิวให้กับวัตถุชิ้นนั้น โดยทางวัตถุออกเป็นแผ่นแบน เพื่อที่จะกำหนดพื้นผิว



ภาพที่ 2.3 ตัวอย่าง UV Mapping

2.2.2 แสงสว่าง(Light)

- 1) บริเวณแสงสว่างจัด (Hi-light) เป็นบริเวณที่อยู่ใกล้แหล่งกำเนิดแสงมากที่สุด จะมีความสว่างมากที่สุด ในวัตถุที่มีผิวมันวาวจะสะท้อนแหล่งกำเนิดแสงออกมาให้เห็นได้ชัด
- 2) บริเวณแสงสว่าง (Light) เป็นบริเวณที่ได้รับแสงสว่าง ร่องลงมาจากบริเวณแสงสว่างจัด เนื่องจากอยู่ห่างจากแหล่งกำเนิดแสงออกมา และเริ่มมีค่าน้ำหนักอ่อน ๆ
- 3) บริเวณเงา (Shade) เป็นบริเวณที่ไม่ได้รับแสงสว่าง หรือเป็นบริเวณที่ถูกบดบังจากแสงสว่าง ซึ่งจะมีค่าน้ำหนักเข้มมากขึ้นกว่าบริเวณแสงสว่าง
- 4) บริเวณเงา เข้มจัด (Hi-Shade) เป็นบริเวณที่อยู่ห่างจากแหล่งกำเนิดแสงมากที่สุด หรือเป็นบริเวณที่ถูกบดบังมาก ๆ หลาย ๆ ชั้น จะมีค่าน้ำหนักที่เข้มมากไปจนถึงเข้มที่สุด
- 5) บริเวณเงาตกทอด เป็นบริเวณของพื้นหลังที่เงาของวัตถุทาบลงไป เป็นบริเวณเงาที่อยู่ภายนอกวัตถุ และจะมีความเข้มของค่าน้ำหนักขึ้นอยู่กับ ความเข้มของเงา น้ำหนักของพื้น หลัง ทิศทางและระยะของเงา



ภาพที่ 2.4 ตัวอย่างการจัดแสง

2.2.3 สีตรงข้าม

สีตรงข้ามก็คือสีที่อยู่ตรงข้ามกันในวงจรสีนั่นเองการที่เราจะทราบว่าสีคู่ใดเป็นสีตรงข้ามกันอย่างแท้จริงหรือไม่ให้นำเอาสีคู่ นั้นมาผสมกันดูถ้าผลการผสมกันออกมาเป็นสีกลางนั้นหมายถึงว่าสีคู่ นั้นเป็นคู่สีตัดกันอย่างแท้จริง ปริมาณของสีที่ตัดกันกับวรรณะของสีทั้งหมดในภาพต้อง อย่างเกิน 10% ของเนื้อที่ในภาพเขียน ในลักษณะการนำไปใช้ ในทางประยุกต์ศิลป์หรือเชิงพาณิชย์ควรใช้ตาม หลักเกณฑ์ดังนี้

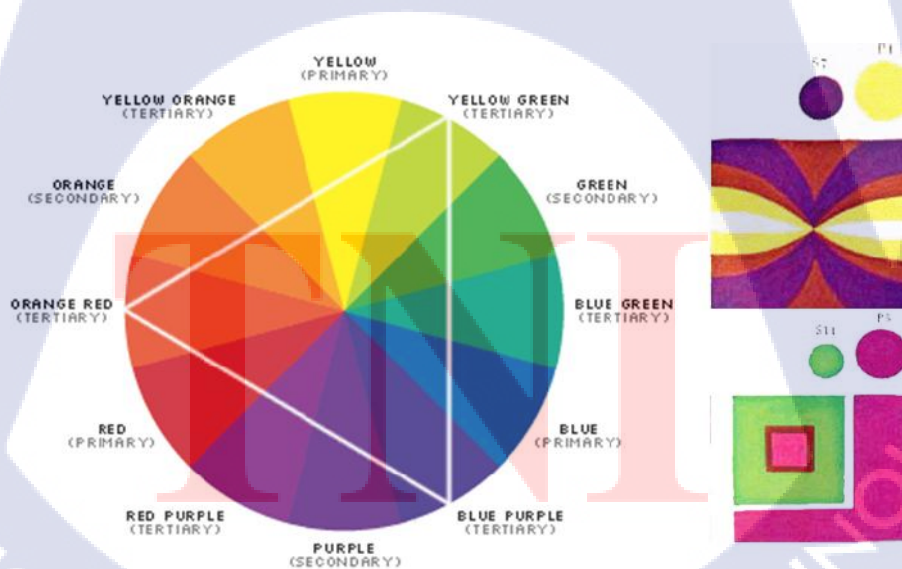
- การใช้สีตรงข้ามหรือสีตัดกัน ต้องใช้สีใดสีหนึ่งจำนวน 80% อีกฝ่ายหนึ่ง ต้องเป็น 20%

จึงจะ

มีคุณค่าทางศิลปะ

- หากจำเป็นต้องใช้สีคู่ใดคู่หนึ่งปริมาณเท่าๆกัน ควรต้องลดค่าของคู่สีลง
- หากภาพเป็นลายเล็กๆ เช่น ภาพที่เต็มไปด้วยต้นไม้ใบไม้เล็ก การใช้สีตัดกันอย่างสดๆ สดๆ ผลก็จะผสมผสานกันเอง
- หากจำเป็นต้องใช้สีตัดกันในภาพใหญ่หรือพื้นที่ภาพมากและสีคู่ นั้นติดกัน ควรใช้เส้น

คำ



ภาพที่ 2.5 ทฤษฎีสีคู่ตรงข้าม

2.2.4 อณัตมี (Anatomy)

รูปร่างและโครงสร้างของสิ่งที่มีชีวิต รวมทั้งพืชและสัตว์ คำว่า Anatomy แยกออกได้เป็น Ana = Apart แปลว่า เป็นชิ้นหรือเป็นส่วนๆ ส่วน Tomy หรือ Tome = Cutting แปลว่า ตัด ดังนั้น Anatomy เมื่อรวมกันแล้ว จึงหมายถึง การตัดหรือชำแหละออกเป็นส่วนๆ ซึ่งในการเรียน Gross Anatomy ใช้การชำแหละ ด้วยตาเปล่าเป็นหลัก

ในการทำอนิเมชันจำเป็นมากที่ต้องศึกษาเรื่องของสรีระ ซึ่งใช้ในการเคลื่อนไหวส่วนต่างๆ ของร่างกาย และขอบเขตหรือข้อจำกัดของการขยับในแต่ละส่วน จำพวกกระดูก ข้อต่อ และ กล้ามเนื้อ



ภาพที่ 2.6 สรีระการเคลื่อนไหว

2.2.5 เกมและการเรียนรู้

เกมในปัจจุบันเป็นเกมในลักษณะเล่นอย่างเดียว “mere play” คือ วัตถุประสงค์เพื่อความเพลิดเพลิน สนุกสนานในการเล่นของผู้เล่น กระบวนการสร้างเกมประเภทนี้ไม่ซับซ้อนและยุ่งยากหากต้องการให้บรรลุวัตถุประสงค์ข้างต้น แต่เกมเพื่อการศึกษาเป็นเกมที่มีลักษณะการเล่นเพื่อการเรียนรู้ “Play to Learning” วัตถุประสงค์หลักเพื่อให้ผู้เรียนเกิดการเรียนรู้ในขณะหรือหลังจากการเล่น เกม เรียนไปด้วยและก็สนุกไปด้วยพร้อมกัน ทำให้ผู้เรียนมีการเรียนรู้ด้วยความหมาย กระบวนการสร้างเกมเพื่อการศึกษาจำเป็นต้องผ่านการออกแบบแบบลักษณะของเกมโดยยึด

ตามหลักทฤษฎีต่างๆ ที่เกี่ยวข้อง ดังนั้นเกมการศึกษาจึงเป็นเกมที่มีกระบวนการสร้างที่ซับซ้อน และใช้เวลามากในการสร้างและพัฒนาการใช้และการสร้างเกมเพื่อการศึกษาในปัจจุบันมีลักษณะคล้ายๆกันคือ การนำเนื้อหาที่ต้องการให้ผู้เรียน เรียนนำไปแทรกในเกม แล้วให้ผู้เรียนได้เล่นเกมโดยเชื่อว่าความรู้หรือเนื้อหานั้นจะส่งผ่านไปยัง ผู้เรียนจนผู้เรียนเกิดการเรียนรู้ได้ในที่สุด โดยรูปแบบเกมที่สร้างขึ้นส่วนใหญ่จะคำนึงถึงเฉพาะความสะดวกและความสะดวกในการ สร้างและพัฒนาเกมเท่านั้น ทำให้เกมทางการศึกษาจึงมีการสร้างเพียงไม่กี่รูปแบบ และยังใช้เกมรูปแบบเดียวใช้สอนเนื้อหาที่ต่างกันเพื่อความสะดวกและสะดวกใน การสร้าง ทำให้เกิดข้อสงสัยว่ารูปแบบเกมแบบเดียว สามารถใช้กับเนื้อหาที่แตกต่างกันได้จริงหรือ สิ่งที่ควรคำนึงถึงในการเลือกรูปแบบเกมเพื่อนำมาใช้เพื่อการศึกษาจะต้อง คำนึงถึงจุดประสงค์การเรียนรู้เป็นหลัก โดยสามารถจำแนกลักษณะของเกมแบ่งจุดประสงค์ของการเรียนรู้กับรูปแบบเกมที่ เหมาะสม ดังนี้ (เกมคอมพิวเตอร์)

1. ความจำ ความคงทนในการจำ ลักษณะเกมเป็นชุดของเนื้อหาและแบบประเมินหลังจากการอ่านชุดเนื้อหาต่างๆ แล้ว รูปแบบเกม เช่น เกมแบบฝึกหัด, Quiz, เกม Crossword และเกม puzzles ต่างๆ เป็นต้น
2. ทักษะ การกระทำ เป็นเกมในลักษณะจำลองสถานการณ์เรื่องราว การกระทำ การเลียนแบบ โดยมีการให้ผลป้อนกลับและมีตัวแปรอื่นๆ เข้ามาเกี่ยวข้อง เช่น เวลา รูปแบบเกม เช่น เกม Simulation ต่างๆ เช่น เกมยิง, เกมขับรถ เป็นต้น
3. ประยุกต์ความคิดรวบยอดและกฎข้อบังคับต่างๆ เป็นเกมในลักษณะกฎและขั้นตอนวิธีการในการปฏิบัติ มีเงื่อนไขในการกระทำ เช่น เกมกีฬาต่างๆ
4. ตัดสินใจ การแก้ปัญหา ลักษณะเป็นเกมแบบเป็นเรื่องราว สถานการณ์ สามารถแสดงผลการกระทำได้ในทันที Real Time รูปแบบเกม เช่น เกมวางแผน, เกมผจญภัย เป็นต้น
5. การอยู่ร่วมกับสังคม ลักษณะเป็นเกมเกมเกี่ยวกับการสื่อสาร การเล่าเรื่องแล้วมีทางเลือกรูปแบบเกม เช่น เกมวางแผน, เกมผจญภัย, เกมเล่าเรื่องราวแล้วให้เลือก(เกมภาษา) เป็นต้น

จะเห็นได้ว่าจุดประสงค์การเรียนรู้ที่ต่างกัน เกมที่ใช้ก็แตกต่างกันไปด้วย ดังนั้นในการเลือกใช้และพัฒนาเกมในอนาคตเราจึงควรคำนึงถึงจุดประสงค์การเรียนรู้ควบคู่ไปกับรูปแบบเกมที่เหมาะสม เพื่อให้ผู้เรียนรู้สึกสนุกไปพร้อมกับการเกิดการเรียนรู้ได้จริง

2.2.6 อัลกอริทึม

อัลกอริทึม (Algorithm) คือ กระบวนการ การทำงานที่ใช้การตัดสินใจ โดยนำหลักเหตุผล และคณิตศาสตร์มาช่วยเลือกวิธีการหรือขั้นตอนการดำเนินงานต่อไป จนกระทั่งถึงขั้นตอนสุดท้าย เป็นวิธีการที่ใช้แยกย่อยและเรียงลำดับขั้นตอนของกระบวนการในการทำงานต่างๆ เพื่อเพิ่ม ประสิทธิภาพในการค้นหาและแก้ไขปัญหา

ที่มาของคำว่าอัลกอริทึม (Algorithm) คือ คำที่ตั้งให้เป็นเกียรติแก่ อแลน เดอะ กอริทิม ทิวริง (Alan The Gorithm Turing) ผู้ค้นพบว่าการพิสูจน์ทางคณิตศาสตร์หรือปัญหาทางตรรกะ สามารถหาคำตอบได้ด้วยชุดของขั้นตอนวิธีที่ถูกต้อง

ประโยชน์ของอัลกอริทึม (Algorithm) คือ ทำให้ไม่สับสนกับวิธีดำเนินงาน เพราะทุกอย่าง จะถูกจัดเรียงเป็นขั้นตอนมีวิธีการและทางเลือกไว้ให้ เมื่อนำมาใช้จะทำให้การทำงานสำเร็จอย่าง รวดเร็ว ทำให้ปัญหาลดลงหรือสามารถค้นหาต้นเหตุของปัญหาได้อย่างรวดเร็ว เนื่องจาก กระบวนการถูกแยกแยะกิจกรรม ขั้นตอน และความสัมพันธ์ ออกมาให้เห็นอย่างชัดเจน

คำว่า Algorithm มีที่มาจากชื่อของนักคณิตศาสตร์ชาวเปอร์เซียในยุคศตวรรษที่ 9 อะบู อับดิลลาห์ บิน มุซา อัลคาวาริซมียฺ (Abu Abdillah Muhammad bin Musa al-Khawarizmi) คำว่า al-Khawarizmi ได้เขียนเป็น Algoritmi เมื่องานเขียนของเขาได้รับการแปลเป็นภาษาละติน แล้ว กลายเป็น Algorithm อัลกอริทึม ซึ่งใช้หมายถึงกฎที่ใช้ในการคิดคำนวณเลขคณิต และได้กลายมา เป็นคำ ขั้นตอนวิธี ในช่วงศตวรรษที่ 18.

ในปัจจุบัน คำนี้ได้มีความหมายที่กว้างขึ้น หมายรวมถึง ขั้นตอนวิธีการในการแก้ปัญหา ต่างๆ ขั้นตอนวิธีแรกสำหรับคอมพิวเตอร์นั้น เขียนขึ้นในปี ค.ศ. 1842 โดย เอดา ไบรอน ใน notes on the analytical engine ทำให้ถือกันว่า เอดาเป็นนักพัฒนาโปรแกรมหรือโปรแกรมเมอร์คนแรก ของโลก แต่เนื่องจาก ชาร์ลส แบบเบจ ไม่ได้สร้าง analytical engine จนเสร็จ ขั้นตอนวิธีของเอดา นั้นจึงไม่ได้มีการใช้จริง ถึงแม้ว่าขั้นตอนวิธีนั้นเป็น ขั้นตอนวิธี การแก้ปัญหา ที่ถูกระบุไว้อย่าง ชัดเจน แต่ก็ขาดรูปแบบการวิเคราะห์ในรูปแบบจำลองทางคณิตศาสตร์ที่ชัดเจน ปัญหาในทาง ขั้นตอนวิธีนี้โดยส่วนมากจึงมักจะถูกวิเคราะห์โดยใช้ เครื่องจักรทัวริง ซึ่งเป็นแบบจำลองนามธรรม ของคอมพิวเตอร์ คิดค้นขึ้น

โดย แอลัน ทัวริง ซึ่งเป็นเครื่องจักรที่ใช้ในการจำลองการทำงานของขั้นตอนวิธีใดๆ

□□□□□□ (Algorithm) หรือขั้นตอนวิธีทางคอมพิวเตอร์ต้องมีการวัดประสิทธิภาพโดยนำความรู้ทางคณิตศาสตร์มาใช้ ดังนี้

1. การพิสูจน์โดยการเหนี่ยวนำเชิงคณิตศาสตร์ มีหลักการคือ กล่าวถึงว่าทฤษฎีนั้นถูกต้องสำหรับค่าตัวแรกเริ่มต้นหรือค่าต่ำสุด โดยแทนค่าดังกล่าวลงในทฤษฎีเพื่อแสดงให้เห็นว่า ทฤษฎีนั้นถูกต้อง จากนั้นสมมติให้ทฤษฎีนั้นยังถูกต้องเมื่อเพิ่มค่าตัวแปรจนถึงค่าที่ $n-1$ แล้วจึงนำสมมติฐานนั้นมาพิสูจน์ทฤษฎี โดยการทำให้ค่าของตัวแปรเพิ่มขึ้นอีกหนึ่งจำนวน

2. การหาผลบวกและผลคูณ

3. ทฤษฎีตัวเลข

4. การประมาณค่า

5. การหาสัมประสิทธิ์แบบ Binomial

6. การสรุปโดยการประมาณขอบเขตบนสุด

การประยุกต์ใช้อัลกอริทึม สามารถ นำอัลกอริทึมมาปรับใช้กับการทำงานของเรา โดยที่เรา กำหนดอัลกอริทึมของงานขึ้นมา ซึ่งอาจทำให้ลดเวลาการทำงาน และเพิ่มประสิทธิภาพในการทำงานได้ หรือกระทั่งช่วยการวางแผนชีวิต เช่น ขั้นตอนการลงทุนจนถึงผลของการลงทุน เป็นต้น

Algorithm คือ กระบวนการแก้ปัญหาที่สามารถอธิบายออกมาเป็นขั้นตอนที่ชัดเจน เมื่อนำเข้าอะไร แล้วจะต้องได้ผลลัพธ์เช่นไร กระบวนการนี้ประกอบด้วยจะประกอบด้วย วิธีการเป็นขั้นๆ และมีส่วนที่ต้องทำแบบวนซ้ำอีก จนกระทั่งเสร็จสิ้นการทำงาน Algorithm ไม่ใช่คำตอบแต่เป็นชุดคำสั่งที่ทำให้ได้คำตอบ วิธีการในการอธิบาย Algorithm ได้แก่

1. Natural Language อธิบายแบบใช้ภาษาที่เราสื่อสารกันทั่วไป

2. Pseudocode อธิบายด้วยรหัสจำลองหรือรหัสเทียม

3. Flowchart อธิบายด้วยแผนผัง การนำขั้นตอนวิธีไปใช้แก้ปัญหา ไม่จำกัดเฉพาะการเขียนโปรแกรมคอมพิวเตอร์ แต่สามารถใช้กับปัญหาอื่น ๆ ได้เช่นเดียวกัน ตัวอย่างเช่น

ในการวางแผนการใช้ทรัพยากรทางธุรกิจขององค์กร หรือ Enterprise Resource Planning (ERP) เพื่อให้เกิดการใช้ทรัพยากรอย่างมีประสิทธิภาพสูงสุด ซึ่งจำเป็นต้องวางแผนอย่างเป็นระบบเป็นขั้นตอน จึงจำเป็นต้องอาศัย Algorithm ด้วย เพื่อให้ทราบถึงขั้นตอนต่างๆ และสามารถตัดทอนขั้นตอนที่เกินความจำเป็น อีกทั้งยังสามารถปรับปรุง และเพิ่มเติมขั้นตอนใหม่ เข้าไปได้ ช่วยลดความสับสนขณะทำงานด้วย

ภาษา C# เป็น ภาษาโครงสร้างแบบเดียวกับภาษา C ที่ทำความเข้าใจได้ง่าย มีกฎระเบียบข้อบังคับที่รัดกุม อีกทั้งคำสั่งส่วนใหญ่เป็นภาษาอังกฤษจึงมีความหมายตรงตัวในบางคำสั่ง

(บางครั้งปัญหาเรื่องภาษาอังกฤษก็ทำให้เกิดปัญหาในการเขียนโปรแกรมได้ครับ ดังนั้นจึงควรเรียนรู้ควบคู่กันไป) ดังนั้นจึงไม่คิดที่จะเริ่มฝึกเขียนโปรแกรมจากภาษา C#

สิ่งที่ต้องมีเพื่อที่จะให้บรรลุการเป็นโปรแกรมเมอร์

1. ทำให้รักการ โปรแกรม ผมพูดได้เลยว่าไม่เพียงเฉพาะการเขียนโปรแกรมเท่านั้น ถ้าคุณทำให้รักสิ่งใดสิ่งหนึ่งซักอย่างได้ คุณก็จะทำมันได้ดีแน่นอน
2. อดทนอดกลั้น การ โปรแกรมไม่ใช่เรื่องง่าย แต่ก็อย่าท้อไป โดยให้นึกอย่างผมว่า คนสมัยก่อนยังสร้างภาษามาให้เขียน โปรแกรมแบบนี้ได้ แล้วทำไมเราจะไม่พยายามทำให้ได้อย่างเค้าบ้างล่ะ
3. เพียรพยายาม ขยัน แน่ใจทุกสิ่งทุกอย่างต้องหมั่นฝึกฝน ใช้นั้นให้บ่อยที่สุดแล้วคุณจะสามารถไปเอง
4. แสวงหาความรู้จากแหล่งความรู้อื่น เพื่อให้เข้าใจการ โปรแกรมอย่างลึกซึ้งควรแสวงหาเพิ่มเติม เพราะไม่มีใครทุกคนหรอกที่จะเข้าใจคำพูดของคนคนเดียวไปเสียหมด ต้องออกไปแสวงหาทำความเข้าใจเอาเองเสียบ้าง แต่ก็เข้าใจ

สำหรับมือใหม่บางคนที่ไม่มีพื้นฐาน ก็มีเทคนิคให้

1. ในตอนเริ่มต้นยังไม่ต้องทำความเข้าใจเกี่ยวกับที่มาของโค้ดมากนัก ให้พยายามจำรูปแบบของโค้ดเอาไว้พอ
2. มือใหม่ห้ามยุ่งกับส่วนอื่น ให้สนใจเฉพาะ Method Main เป็นหลัก เพราะ โปรแกรมจะเริ่มทำงานในส่วนนี้
3. ขยายความจากข้อ 1 จำคำสั่งเป็นรูปแบบ เช่น
 - Console.WriteLine() เป็นคำสั่ง
 - output Console.ReadLine() เป็นคำสั่งรับ input โดยยังไม่ต้องสนใจที่มา
4. นำความรู้ความจำได้ในบางส่วนมาลองเขียนโปรแกรมเล่นๆดู โดยลองนึกโจทย์เอง
5. ฝึกทำความเข้าใจจนคุ้นเคยแล้วจึงมาศึกษาที่มาที่จะช่วยให้เข้าใจได้เร็วขึ้นครับ
6. หลังเครื่องหมาย ** จะเป็นสิ่งที่สำคัญและควรจำเสมอ

ภาษา C# (C# Programming Language) จะ เขียนว่า C Sharp (ซี-ชาร์พ) ก็ได้ไม่ผิด เราอาจจะเคยได้ยินภาษาอื่นๆที่คล้าย C# เช่น ภาษา C, Java, C++ ซึ่งแน่นอนมันเป็นที่มาของ C# นั่นเอง (ตัว C เป็นตัวบ่งบอกให้รู้ว่า C# มีต้นกำเนิดมาจาก C นั่นเอง) เครื่องหมาย # ดังที่เคยพูดคือ เป็นสิ่งที่แสดงถึงความก้าวหน้ากว่า C++ ไปอีกระดับหนึ่ง (มอง # ให้เป็นเครื่องหมาย + ซ้อนกันสี่มุมสิครับ อาจมองแบบ C++++ ก็ได้ไม่ผิด)

ถ้าจะให้พูดกันเข้าใจง่ายอีกก็คือ C# ได้รวบรวมข้อดีของภาษาต่างๆเช่น Java, Delphi, C++ เข้าไว้ด้วยกัน อีกทั้งยังมีความเรียบง่ายกว่า (ใครเคยใช้ Java จะรู้ว่ามีความคล้ายกับ C# มากที่สุด) อีกทั้งยังมีเครื่องมือดีๆ อย่าง Visual C# 2008 ของทางไมโครซอฟท์อีกด้วย จึงลดความยุ่งยากในการโปรแกรมได้มาก สามารถพัฒนาโปรแกรมระดับสูงได้ด้วย

ภาษา C, C++, C# จัดเป็นภาษาระดับกลาง กล่าวคือ ได้รวมเอาข้อดีจาก ภาษาระดับต่ำ และสูงเข้าไว้ด้วยกัน

บิดาของภาษา C# (C# เวอร์ชันต่างๆทั้งหลายที่ Microsoft พัฒนา ก็มีท่านคนนี้แหละครับ เป็นต้นแบบที่มั่นคงของภาษาที่ได้รับความนิยมนี้) คือ Anders Hejlsberg (แอนเดรส ฮาเยสเบิร์ก)

ใครๆก็สามารถสร้างภาษาใหม่ๆขึ้นได้ แต่ความเรียบง่าย และสวยงามของ C# ที่ Hejlsberg ได้บรรจงสร้างมาให้เหล่าโปรแกรมเมอร์ยุคใหม่ใช้กันอย่างสะดวกสบาย เป็นอะไรที่หาที่เปรียบไม่ได้แล้ว

แม้ต้นกำเนิดจะมาจาก Anders Hejlsberg นั้น แต่ไมโครซอฟท์ต้องการให้ภาษานี้ "อยู่ไปอีกล้านปี" จึงได้พัฒนา C# เรื่อยมาจนปัจจุบันเป็นเวอร์ชัน C# 2008 แล้ว (2010 ก็ใกล้จะออกมาแล้วเหมือนกันนะ)

อีกประการที่ควรเลือก C# ของไมโครซอฟท์เพราะ เราสามารถมั่นใจได้ว่า ถ้าเราศึกษาตรงนี้ไปจะไม่เสียเปล่า เพราะจะมีการพัฒนาต่อไปเรื่อยๆ (ไม่ตกยุคนั่นเอง ไม่ต้องกลัวว่าไมโครซอฟท์จะทิ้งเรา) ฟรีครับฟรี ไมโครซอฟท์เค้าใจดีปล่อย Visual C# 2008 Express Edition ซึ่งเป็นเวอร์ชันสำหรับการศึกษา แม้จะขาด component และคำสั่งบางส่วนไป แต่ในระดับเริ่มต้นก็ยังไม่ต้องใช้ครับ (นอกจาก C# แล้วภาษาอื่นก็มีเช่น Basic, C++)

Variable (ตัวแปร) ถ้าพูดถึงวิชาคณิตศาสตร์ ตัวแปรนั้นจะหมายถึงสิ่งที่เราไม่ทราบค่าและต้องการหาค่าเหล่านั้น แต่ในทางโปรแกรมมิ่ง ตัวแปรคือ สิ่งที่ใช้เก็บค่าของข้อมูลนั้นๆ ในเงื่อนไขหนึ่งๆ

ตัวแปร คือ การเก็บค่าของข้อมูลโดยมีสิ่งที่อ้างอิงได้จากการตั้งชื่อให้กับหน่วยความจำ เพื่อใช้ในการเก็บค่าต่างๆ ตัวแปรแต่ละตัวจะสามารถเก็บค่าได้เพียง 1 ค่าเท่านั้น

ในการเขียนโปรแกรมจะมีการจองพื้นที่หน่วยความจำ เพื่อใช้เก็บข้อมูล และต้องกำหนดชื่อเรียกเพื่อนอ้างอิงพื้นที่หน่วยความจำที่จอง ข้อมูลในตัวแปรสามารถเปลี่ยนแปลงได้ตลอดเวลา

กฎการตั้งชื่อตัวแปร

1. ห้ามนำคำสงวน (reserved word) มาตั้งเป็นชื่อตัวแปร
2. ต้องขึ้นต้นด้วย ตัวอักษรภาษาอังกฤษ อักขระพิเศษ @ และ _ (underscore) เท่านั้น
3. ต้องไม่ขึ้นต้นด้วยตัวเลข เช่น 9number, 105Ages ถือว่าผิด
4. ต้องไม่ประกอบไปด้วยอักขระพิเศษทุกชนิด (เครื่องหมาย /, +, -, *, %, !)
5. ตัวอักษรใหญ่และเล็กจะถือว่าไม่เหมือนกัน เช่น mVar, MVAR จะถือว่าเป็นคนละตัวแปร
6. ไม่ควรใช้ชื่อยาวเกินไป และควรตั้งให้เหมาะสมกับการนำไปใช้ เช่น ตัวแปรเก็บอายุก็ควรตั้งว่า Age เป็นต้น

ตัวอย่างการตั้งชื่อตัวแปรที่ถูก

- _Number (ถูกต้องเพราะขึ้นต้นด้วย _ และประกอบไปด้วยตัวอักษร)
- @humor123 - InputNumber (ถูกต้องเพราะขึ้นต้นด้วยตัวอักษร) ตัวอย่างการตั้งชื่อตัวแปรที่

ผิด

- 12Var (ผิดเพราะ ขึ้นต้นด้วยตัวเลข)
- inter-net (ผิดเพราะมีอักขระพิเศษ -)
- case (ผิดเพราะ case เป็นคำสงวน)

การตั้งชื่อตัวแปรในช่วงแรกๆนั้น จะนิยมตั้ง นำหน้าด้วยตัวอักษรที่สื่อถึงชนิดของข้อมูล เช่น

InCost (ตัวแปรเก็บค่าราคา ชนิด integer)

Dpi (ตัวแปรเก็บค่าพาย ชนิด Double)

bChecker (ตัวแปรเก็บค่าความจริงชนิด Boolean) เรียกการตั้งชื่อตัวแปรแบบนี้ว่า "อันการเย็น โนเดชั่น"

การตั้งชื่อตัวแปรในรูปแบบนี้ ได้รับความนิยมมากเนื่องจาก ป้องกันการลืมนิยามข้อมูลของตัวแปร เมื่อเราเขียน โปรแกรมที่มีขนาดใหญ่แต่ปัจจุบันนี้ Compiler Visual Studio สามารถบอกชนิดของข้อมูลของตัวแปรได้เพียงแค่นำเมาส์ไปชี้ที่ชื่อตัวแปร การตั้งชื่อรูปแบบนี้จึงไม่นิยมอีกต่อไป

การประกาศตัวแปร ก่อนที่เราจะได้ใช้งานตัวแปร เราจะต้องทำการประกาศตัวแปรเสียก่อน เพื่อที่จะให้โปรแกรมได้จองพื้นที่หน่วยความจำไว้สำหรับรองรับค่าที่จะรับเข้ามาเก็บไว้ในตัวแปรต่อไป

ขนาดของหน่วยความจำที่ใช้จะขึ้นอยู่กับชนิดของตัวแปร (ชนิดของตัวแปร) (ชื่อของตัวแปร); // เป็นการประกาศค่าตัวแปรโดยที่ตัวแปรจะไม่มีค่าใดๆในหน่วยความจำ เช่น `int number ;`
// เป็นการประกาศตัวแปร `number` ชนิด `int` (จำนวนเต็ม)

เราสามารถประกาศตัวแปรได้หลายตัวพร้อมกัน (ชนิดของตัวแปร) (ชื่อของตัวแปร 1), (ชื่อของตัวแปร 2), (ชื่อของตัวแปร 3), ..., (ชื่อของตัวแปร n); เช่น

`int a,b,c,d,e ;` ตัวแปรที่ได้จะเป็นชนิด `int` (จำนวนเต็ม) ทั้งหมด เราสามารถประกาศตัวแปรพร้อมกับใส่ค่าเริ่มต้นให้กับตัวแปรได้ (ชนิดของตัวแปร) (ชื่อของตัวแปร) = ค่าที่ของตัวแปร ; เช่น
`double pi = 3.14 ;` // เป็นการประกาศค่าตัวแปร `pi` ชนิด `double` (จำนวนจริง) พร้อมกับใส่ค่าให้กับตัวแปร `pi`

สำหรับการประกาศค่าของ `string` และ `char` จะมีการกำหนดค่าเริ่มต้นที่ต่างจากตัวเลขดังนี้
`string text = "Oak";` // จะมีเครื่องหมาย `"...."` มาครอบข้อความเอาไว้
`char word = 'O';` // จะมีเครื่องหมาย `'...'` มาครอบข้อความเอาไว้

การประกาศตัวแปรค่าคงที่ (Constant) ค่าคงที่ในโค้ดโปรแกรมมีหน้าที่เฉพาะเจาะจงในแต่ละค่า เช่น ค่า พาย ค่า e หรือค่าอะไรก็ได้แล้วแต่ ซึ่งเมื่อเราประกาศตัวแปรค่าคงที่มาแล้วนั้น จะไม่สามารถเปลี่ยนแปลงค่าได้อีก ไม่ว่าในส่วนของโปรแกรมใดๆ

รูปแบบการประกาศตัวแปรค่าคงที่ const (ชนิดของตัวแปรค่าคงที่) = (ค่าที่ต้องการ) ;
การประกาศตัวแปร const จะต้องประกาศพร้อมกับใส่ค่าให้กับตัวแปร จะประกาศเพียงชื่อไม่ได้
const int oak ; (ถือว่าผิด จะไม่สามารถรันโปรแกรมได้) x x x x x const double PI = 3.14159 ;
ส่วนใหญ่มักจะตั้งชื่อค่าคงที่ด้วยตัวอักษรใหญ่

การเขียนโปรแกรมแบบมีเงื่อนไข

โปรแกรมในปัจจุบันมีการทำงานด้วยรูปแบบเงื่อนไขต่างๆมากมายเพื่อควบคุมทิศทางการทำงานของโค้ดโปรแกรม เปรียบดังในชีวิตประจำวันที่เรามีการตัดสินใจในการทำสิ่งต่างๆ ตลอดเวลา เลือกสถานที่ที่จะไป เลือกร้านอาหารที่จะรับประทานเราจะใช้กระบวนการนี้ทำให้โปรแกรมมีประสิทธิภาพมากขึ้น ซึ่งก็หมายความว่าโปรแกรมสามารถที่จะตัดสินใจทำงานได้ด้วยตนเอง โดยที่ผู้ใช้โปรแกรมไม่ต้องระบุหน้าที่อย่างชัดเจนจนเกินไป

□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□

1. การตัดสินใจ (Decision)
2. การทำงานซ้ำ (Iteration)

1. □□□□□□□□ (Decision) การทำงานตามเงื่อนไข โดยมีทางเลือกที่แน่นอน

คำสั่ง if else

ขอบอกไว้ ณ ที่นี้เลยว่า รู้ภาษาอังกฤษก็มีชัยไปกว่าครึ่งแล้วในการเขียน โปรแกรม if else ก็เช่นกัน ความหมายตรงตัว if(ถ้า) ซึ่งจะตามมาด้วยโค้ดเงื่อนไข else(นอกจาก) ซึ่งต้องตามหลังคำสั่ง if เท่านั้น ซึ่งก็หมายถึงนอกเหนือจากเงื่อนไขของ if นั่นเอง เพื่อให้เข้าใจยิ่งขึ้นมาดูรูปแบบคำสั่งของ if กันสักหน่อย if (เงื่อนไข-1) { คำสั่ง ; }

การพิมพ์โค้ดคำสั่ง if เทคนิคการพิมพ์ให้รวดเร็วคือ พิมพ์ if ตามปกติ จากนั้นกด TAB โปรแกรมจะเขียน โค้ดขึ้นมาให้บางส่วนสร้าง loop และ () ของเงื่อนไข ซึ่งถือว่าช่วยให้เราสะดวก

มากขึ้นในการเขียนโค้ด

ในส่วนของ เงื่อนไข-1 ส่วนใหญ่เราจะใช้การเปรียบเทียบเชิงตรรกะ โดยใช้ตัวดำเนินการทางตรรกะ (logical Operator) หมายความว่า ถ้าเงื่อนไข เป็นจริง ให้เข้ามาทำ คำสั่ง ; ภายใน loop ของ if รูปแบบคำสั่ง else if (เงื่อนไข-1) { คำสั่ง-1 ; } else { คำสั่ง-2 ; // สังเกตให้ดูว่า else ไม่ต้องมีเงื่อนไขก็ได้ ซึ่งจำทำ คำสั่ง-2 ก็ต่อเมื่อ เงื่อนไข-1 เป็นเท็จ } เมื่อคำสั่งใน loop if มีเพียงคำสั่งเดียว เรานิยมที่จะไม่ใส่ "{", "}"

ตัวอย่างของโปรแกรมที่มีการใช้ if (โค้ดเหล่านี้ให้พิมพ์ใน เมธอด static void Main(string[]args))

```
int a = int.Parse( Console.ReadLine() );
int b = int.Parse( Console.ReadLine() );
if ( a<b )
{
    Console.WriteLine(" {0} less than {1} ", a, b);
}
else if ( a>b )
{
    Console.WriteLine(" {0} more than {1} ", a, b);
}
```

คำสั่ง switch

คำสั่ง switch เป็นคำสั่งที่ใช้ตรวจสอบ ค่าของ object กับเงื่อนไข ซึ่งสามารถตรวจสอบเงื่อนไขได้หลายเงื่อนไขโดยจะทำการตรวจสอบทีละเงื่อนไข ซึ่งถ้าตรงกับค่าที่กำหนด (หลัง case) โปรแกรมจะทำงานในคำสั่ง หลังเครื่องหมายโครอน ":" ไปเรื่อยๆ จนกว่าจะเจอคำสั่ง break เมื่อเจอคำสั่ง break โปรแกรมจะกระโดดการทำงานไปยังส่วนสุดท้ายของคำสั่งคือ เครื่องหมายปีกกาปิด " } "

ในการตรวจสอบเงื่อนไขนั้นโปรแกรมจะทำงานแบบบนลงล่างไปเรื่อยๆ จนถึงเครื่องหมายปิดกาปิด แต่ถ้าไม่ตรงกับเงื่อนไขใดๆเลย โปรแกรมจะทำงานในคำสั่งที่อยู่หลังคำสั่ง default ไปเรื่อยๆจนเจอคำสั่ง break เรามาดูรูปแบบคำสั่ง และ Flowchart ของ switch

switch (object ที่ต้องการตรวจสอบเงื่อนไข)

```
{
    case เงื่อนไข-1 : คำสั่ง-1 ; break ;
    case เงื่อนไข-2 : คำสั่ง-2 ; break ;
    case เงื่อนไข-3 : คำสั่ง-3 ; break ;
    . . case เงื่อนไข-n : คำสั่ง-n ; break ;
    default : คำสั่ง-n+1 ; break ;
}
```

จากรูป (expression = objectที่เราใช้เปรียบเทียบ, condition = เงื่อนไข, statement = คำสั่ง)

โปรแกรมจะนำ object หรือ expression เข้ามาตรวจสอบเงื่อนไขว่าตรงกับ คำที่ case ใด โดยจะค้นหาจนกว่าจะเจอ โดยถ้าเจอ (หรือได้ค่าความจริงเป็นจริง) จะกระโดดไปที่ส่วนสุดท้ายของโปรแกรม แต่ถ้าไม่เจอ (ได้ค่าความจริงเป็นเท็จ) จะทำการตรวจสอบไปเรื่อยๆ และเมื่อตรวจสอบไม่พบ จนมาถึงส่วนของ default โปรแกรมจะทำงานในคำสั่ง statementX ;

ตัวอย่างโปรแกรมที่มีการใช้ switch (โค้ดเหล่านี้ให้พิมพ์ใน เมธอด static void Main(string[]args))

```
Console.Write("a = ");
int a = int.Parse( Console.ReadLine() );
Console.Write("b = ");
int b = int.Parse( Console.ReadLine() );
Console.WriteLine( "Easy Calculator Press 1 to a+b Press 2 to a-b Press 3 to a*b" );
string c = Console.ReadLine();
switch (c)
```

```

{
    case "1" : Console.WriteLine(a+b) ; break ;
    case "2" : Console.WriteLine(a-b) ; break ;
    case "3" : Console.WriteLine(a*b) ; break ;
    default : Console.WriteLine( "Error" ); break ;
}

```

การเขียนโปรแกรมแบบวนซ้ำ (Iteration)

การทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ หรือจนกว่าจะหมดรอบที่กำหนด นักดูเล่นๆถ้าเราจะเขียนโปรแกรมที่ต้องการให้พิมพ์ค่าๆหนึ่งซ้ำ 1,000 คำ เราคงจะพิมพ์ Console.WriteLine() ไม่ไหวแน่ นี่เป็นเหตุให้เกิดคำสั่งวนซ้ำ ซึ่งมันช่วยลดความวุ่นวายได้มากในการทำอะไรซ้ำไปซ้ำมา ไม่เพียงแต่การพิมพ์ข้อความหลายๆบรรทัด แต่มันมีประโยชน์มากกว่านั้น

คำสั่ง for

คำสั่ง for เป็นคำสั่งวนรอบรูปแบบหนึ่งซึ่งพิเศษตรงที่เรา สามารถกำหนดค่า เริ่มต้น, เงื่อนไข และการเพิ่มค่าตัวแปร ที่เราใช้ในการวนรอบได้โดยอัตโนมัติซึ่งการวนรอบโดยมีลำดับการทำงาน (statements) ที่อยู่ในเครื่องหมายปีกกาเปิด และปีกกาปิดเมื่อเงื่อนไข (condition) เป็นจริงโดยจะทำงานซ้ำจนกว่าเงื่อนไข (condition) เป็นเท็จเพื่อให้ง่ายต่อความเข้าใจ จะอธิบายโครงสร้างของคำสั่งกับฟลัวร์ชาร์ต

จุดเด่นของ for คือ รู้จุดเริ่มต้น และ จุดสิ้นสุดที่แน่นอน

for (ตัวแปร = ค่าเริ่มต้น , [ตัวแปร] [ตัวดำเนินการเปรียบเทียบ] [ค่าคงที่] , อัตราการเพิ่มของตัวแปร)

```

{
    คำสั่ง ;
}

```

หรือ รูปแบบของ for อีกแบบที่มีการประกาศตัวแปรไปเพิ่ม ๆ กับบอกค่าเริ่มต้นของการวนรอบ

```
for ( [ชนิดของตัวแปร] [ชื่อตัวแปร] = ค่าเริ่มต้น , [ตัวแปร] [ตัวดำเนินการเปรียบเทียบ]
[ค่าคงที่] , อัตราการเพิ่มของตัวแปร )
{
    คำสั่ง ;
}
```

เมื่อมีคำสั่งเดียวใน loop `for` เราสามารถลดรูปของโค้ดได้ดังนี้

```
for ( ตัวแปร = ค่าเริ่มต้น , [ตัวแปร] [ตัวดำเนินการเปรียบเทียบ] [ค่าคงที่] , อัตราการเพิ่ม
ของตัวแปร ) คำสั่ง ;
```

ในส่วน of condition = ขอบเขตของการทำซ้ำ จะเป็นในรูปแบบของการเปรียบเทียบ เช่น
 $x < 5$ $a \leq 6$ เป็นต้น

ในส่วน of อัตราการเพิ่มของตัวแปร จะเป็นรูปแบบของ postfix และ prefix - postfix

รูปแบบ [ตัวแปร]++ ; หรือ [ตัวแปร]-- ;

```
int a = 2 ; a++ ;
```

```
Console.WriteLine(a) ; ผลลัพธ์ 3 - prefix รูปแบบ ++[ตัวแปร] ;
```

หรือ --

```
[ตัวแปร] ; int b = 3 ; --b ; Console.WriteLine(b) ; ผลลัพธ์ 2
```

ความต่างของ postfix กับ prefix จะเห็นได้ชัดในส่วนของการ assign ค่า

```
int c = 5 ;
```

```
int x = ++c ;
```

```
int y = c++ ;
```

ผลลัพธ์ของค่า x คือ 6 ผลลัพธ์ของค่า y คือ 5 อาจกล่าวได้ว่า โปรแกรมจะอ่านจากซ้ายไปขวา ซึ่งก็คือ prefix นั้นได้ค่านับเครื่องหมาย ++c จึงทำการบวกค่าอีก 1 แล้วจึงเก็บไว้ที่ตัวแปร x แต่ postfix เครื่องหมาย ++ อยู่หลัง c จึงทำให้โปรแกรมเก็บค่า c เดิมก่อน แล้วจึงค่อยบวก เพื่อไปใช้ในโค้ดถัดๆไป

คำสั่ง while

เป็นคำสั่งวนรอบอีกรูปแบบหนึ่งที่มีจุดเด่นในการ ตรวจสอบเงื่อนไขโดยไม่สนใจจุดเริ่มต้นของตัวแปร (ไม่มีการกำหนดจุดเริ่มต้นเหมือนคำสั่ง for) คำสั่ง while จะเป็นคำสั่งให้ลำดับการทำงาน (Statement) ที่อยู่ในเครื่องหมายปีกกาเปิดและปีกกาปิด เมื่อเงื่อนไข (condition) เป็นจริงซึ่งจะทำงานซ้ำจนกว่าเงื่อนไข (condition) เป็นเท็จเพื่อให้เข้าใจได้ง่ายยิ่งขึ้น จะอธิบายด้วยฟลอว์ชาร์ตการทำงานของคำสั่ง while

condition = เงื่อนไข , Statements1 = คำสั่ง เมื่อโปรแกรมอ่านเจอคำสั่ง while ก็จะเริ่มตรวจสอบ เงื่อนไข(condition) ว่าเป็นจริงหรือไม่ ถ้าจริง(TRUE) ก็จะเข้ามาใน loop เพื่อทำคำสั่ง (Statements1) แล้วจะกลับไปตรวจสอบเงื่อนไขอีกครั้ง ว่าเป็นจริงอยู่หรือไม่ ซึ่งถ้าเป็นจริงก็จะทำคำสั่งใน loop ซ้ำไปเรื่อยๆ จนกว่า เงื่อนไข(condition)จะเป็นเท็จ เมื่อเป็นเท็จ โปรแกรมจะกระโดดออกมายังส่วนสุดท้ายของคำสั่ง while โดยไม่ทำคำสั่ง Statements1

loop while ที่มีเงื่อนไขเป็นจริงตลอดเวลาเราเรียกว่า "ลูปตลอดเวลา" หรือ "Forever loops"

while (เงื่อนไข) คำสั่ง ; while (เงื่อนไข)

```
{
    คำสั่ง 1 ;
    คำสั่ง 2 ;
    คำสั่ง 3 ;
}
```

คำสั่ง do while

เป็นคำสั่งที่คล้ายคลึงกับ while เพียงแต่ว่าจุดเด่นของมันอยู่ที่ การทำคำสั่งใน loop ก่อน 1 ครั้งแล้วจึงตรวจสอบเงื่อนไข ตรงข้ามกับ while ที่จะตรวจสอบก่อนแล้วค่อยเข้ามาทำคำสั่งที่อยู่ใน loop

คำสั่ง do while จะมีหลักการทำงานคล้ายกับคำสั่ง while คือ จะเป็นคำสั่งให้ลำดับการทำงาน (statements) ที่อยู่ในเครื่องหมายปีกกาเปิด และปีกกาปิด เมื่อเงื่อนไข (condition) เป็นจริง โดยจะทำซ้ำจนกว่าเงื่อนไข (condition) จะเป็นเท็จ แต่จะต่างกันตรงที่ การตรวจสอบเงื่อนไขของคำสั่ง do while จะตรวจสอบในลำดับท้ายสุด เพื่อให้สอดคล้องความเข้าใจ จะอธิบายโครงสร้างของคำสั่งกับฟลัวร์ชาร์ต

```
do
{
    คำสั่ง 1 ;
    คำสั่ง 2 ; // คำสั่ง 1 , คำสั่ง 2 และ คำสั่ง 3 จะถูกโปรแกรมทำก่อน 1 รอบ คำสั่ง 3 ;
}
while (เงื่อนไข) ;    // ห้ามลืม ; หลัง while () เด็ดขาด
```

ตัวอย่างของโค้ดการทำงานซ้ำ ตัวอย่างต่อไปนี้จะเป็นโปรแกรมสุตรคูณ โดยจะแสดงผลเป็นสูตรคูณแม่ 3 (โค้ดต่อไปนี้ให้พิมพ์ในส่วนของคุณ Main)

```
รูปแบบ for
for( int i = 1 ; i <= 12 ; i++ )
{
    Console.WriteLine( "3 x {0} = {1} " , i, 3*i );
}
```

หรือ

```

int i ;
for( i = 1 ; i <= 12 ; i++ )
{
    Console.WriteLine( "3 x {0} = {1} ", i, 3*i );
}

```

จะเห็นได้ว่ารูปแบบของ for จะไม่ต้องประกาศตัวแปรเพื่อมารับจำนวนรอบ เพราะจะสามารถประกาศไปพร้อมกับการใช้คำสั่งได้เลย และยังมีความชัดเจนในด้านการระบุจุดเริ่มต้น

while

```

int i = 1 ;
while( i <= 12 )
{
    Console.WriteLine( "3 x {0} = {1} ", i, 3*i ); i++ ;
}

```

จะเห็นได้ว่าโค้ดคำสั่ง while จะไม่ระบุจุดเริ่มต้นที่แน่ชัด และมีการเพิ่มค่า i อยู่ใน loop ของ while ด้วย

do...while

```

int i = 1 ; do
{
    Console.WriteLine( "3 x {0} = {1} ", i, 3*i ); i++ ;
}
while( i <= 12 );

```

จริงๆ ไม่ต่างกับคำสั่ง while เท่าไหร่ เพียงแต่จะทำในรอบที่ 1 (i=1) โดยที่ไม่ตรวจสอบเงื่อนไขหนึ่งครั้ง แล้วจึงมาเริ่มตรวจสอบที่ i = 2

2.3 โปรแกรมที่ใช้

2.3.1 Adobe Photoshop

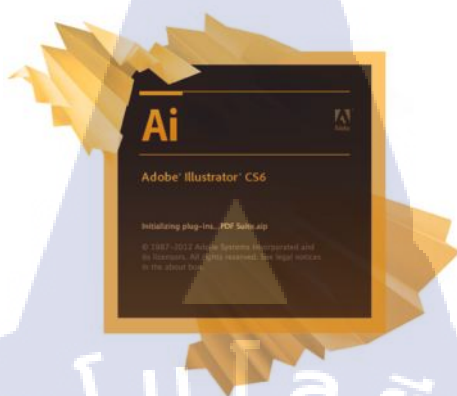


ภาพที่ 2.7 Logo Adobe Photoshop

Adobe Photoshop เป็นโปรแกรมประยุกต์ที่มีความสามารถในการจัดการแก้ไข และตกแต่งรูปภาพ (photo editing and retouching) แบบ bitmap ผลิตโดยบริษัท Adobe Systems ปัจจุบัน โปรแกรม Photoshop พัฒนามาถึงรุ่น CS6 (Creative Suite 6)

TNI

2.3.2 Adobe Illustrator



ภาพที่ 2.8 Logo Adobe Illustrator

Adobe Illustrator) เป็นโปรแกรมวาดภาพกราฟิกแบบเวกเตอร์ ซึ่งพัฒนาโดยบริษัท Adobe Systems รุ่นแรก จัดทำขึ้นในปี ค.ศ. 1986 เพื่อใช้งานกับเครื่อง Macintosh และได้พัฒนารุ่นที่ 2 ออกมาให้ใช้งานได้กับ Window ซึ่งได้รับความพึงพอใจ และการตอบรับที่ดี

2.3.3 Adobe After Effect



ภาพที่ 2.9 Logo Adobe After Effect

Adobe After Effects เป็นโปรแกรมประยุกต์เกี่ยวกับการสร้างกราฟิกเคลื่อนไหว , ภาพเอฟเฟ็ก และคอมโพสิตในระบบดิจิทัล ผลิตโดยบริษัท Adobe Systems ที่ใช้ในกระบวนการผลิตเบื้องหลังของการสร้างภาพยนตร์ และ รายการโทรทัศน์ การใช้งานหลัก

คือการก่อกำเนิดแอนิเมชันแบบ 2 มิติ และ 2.5 มิติ ,องค์ประกอบภาพเอฟเฟ็ก และการ ตกแต่ง

2.3.4 Zbrush



ภาพที่ 2.10 Logo Zbrush

เป็นโปรแกรมที่ใช้ในการปั้นและแต่งเติมรายละเอียดให้กับโมเดลที่มีความละเอียดสูง

(High-resolution models) (จำนวน Polygon สูงกว่า 1 ล้าน Polygon ขึ้นไป) เพื่อนำ Model ไปใช้ใน ภาพยนตร์, เกมส์, และแอนิเมชัน เป็นต้น บริษัทต่างๆ ของโลกที่นำโปรแกรมนี้ไปใช้แล้ว ยกตัวอย่าง เช่น ILM, Electronic Arts เป็นต้น โปรแกรม ZBrush นั้น ใช้เทคนิค Dynamic Levels of Resolution ในการ ทำงานเพื่อช่วยให้ ผู้ใช้สามารถเลือกแก้ไขโมเดลในระดับ Global หรือ Local ได้ ทำให้มีทางเลือกในการแก้ไขโมเดลได้มาก

TNI

2.3.5 Unity



ภาพที่ 2.11 Logo Unity

Unity คือเกมเอนจินสำหรับการสร้างเกมสามมิติ ที่ถูกพัฒนาขึ้นมาอย่างต่อเนื่อง ตั้งแต่ในช่วงแรกๆ ที่รองรับการพอร์ตเกมลงบน Windows, OS X และเว็บเท่านั้น แต่ปัจจุบันได้เพิ่มความสามารถในการพอร์ตลงบน iOS, Android และล่าสุดบน Flash Player อีกด้วย

2.3.6 Autodesk Maya

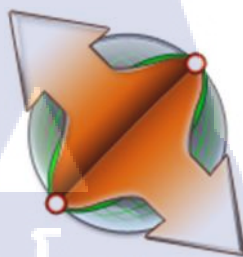


ภาพที่ 2.12 Logo Autodesk Maya

เป็นโปรแกรม มากประสิทธิภาพสำหรับงาน 3D และ visual effect ที่ประกอบไปด้วย ความสามารถในการสร้าง model เครื่องมือในการสร้างงาน animation และ เครื่องมือในการจัดการ rendering ที่มีโครงสร้างสถาปัตยกรรม แบบ open architecture โดยทุกสิ่งทุกอย่างที่จำเป็นในกระบวนการ การผลิตผลงาน สามารถเขียน script หรือเขียนคำสั่ง โดยใช้ MEL หรือ Python ซึ่งเป็นภาษาที่สื่อสาร กันได้โดยตรงกับAutodeskMayaหรือการเขียนคำสั่งการทำงานร่วมกันไปกับ API (application programming interface) AutodeskMaya เป็น สุดยอดโปรแกรม 3D ที่ได้รับการยอมรับ จากผู้สร้าง วิดีโอและภาพยนตร์ ผู้พัฒนา

เกมส์ และ ศิลปินต่างๆ ที่ต้องการถ่ายทอดความคิดและจินตนาการของตน ออกไปอย่างไร
จึงจำกัด

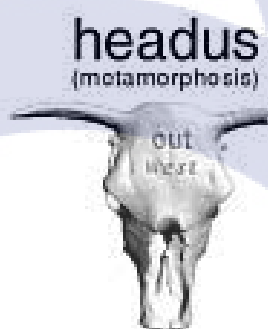
2.3.7 Unfold 3D



ภาพที่ 2.13 Logo Unfold 3D

โปรแกรมที่ช่วยในการตัดและกาง UV เพื่อใช้ทำ Texture ให้กับโมเดลได้ง่ายขึ้น

2.3.8 UV Layout



ภาพที่ 2.14 Logo UV Layout

โปรแกรมที่ช่วยในการตัดและกาง UV เพื่อใช้ทำ Texture ให้กับโมเดลได้ง่ายขึ้น

บทที่ 3

แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1 แผนงานปฏิบัติ

ตารางที่ 3.1 แผนงานปฏิบัติ

งานที่ปฏิบัติ	เดือน 1	เดือน 2	เดือน 3	เดือน 4
โมเดลสนามกีฬา				
โมเดลซูชิ และผัก				
โมเดลที่นั่งรถไฟเหาะ				
โมเดลโอวัลติน				
โมเดลสมุดบัญชี				
โมเดลโทรศัพท์มือถือ				
โมเดลเครื่องยนตร์ D4D				
โมเดลชุดขวด				
โมเดลชุดอาหารหวาน และของว่าง				
โมเดลเครื่องดื่ม				
โมเดลเนื้อ				
โมเดลปลา				
โมเดลคอนโด				
จัดแสงโมเดลลอบบี้				
แผนที่คอนโด				
โมเดลหนังสือการ์ตูน				
จัดโมเดลหนังสือ				

ตารางที่ 3.1 แผนงานปฏิบัติ (ต่อ)

งานที่ปฏิบัติ	□□□□ 1				□□□□ 2				□□□□ 3				□□□□ 4			
รายงานสหกิจศึกษา																
โปรเจกสหกิจศึกษา																

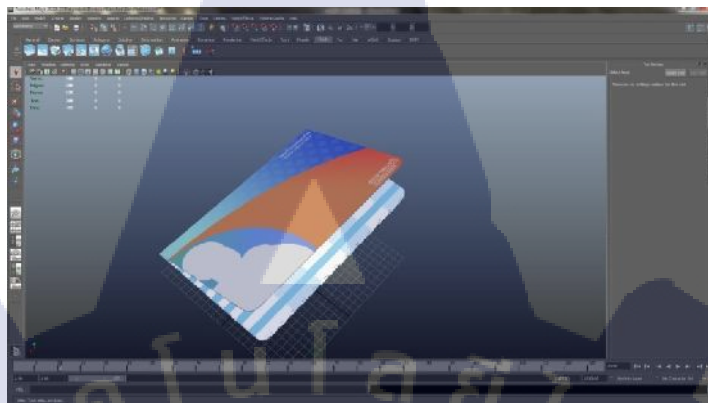
3.2 รายละเอียดงานที่นักศึกษาปฏิบัติในงานสหกิจศึกษา หรือรายละเอียดโครงการที่ □□□□□□□□□□

จากระยะเวลาในการทำงานที่ผ่านมาเป็นระยะเวลา 4 เดือนในบริษัทซีจีโพลีคอน จำกัด หรือ CGpolygo Co.,Ltd ได้เข้าฝึกสหกิจในตำแหน่ง 3D Model ซึ่งนักศึกษาได้รับการส่งเสริมประสบการณ์ในการทำงานต่างๆ มากมาย ได้รับความรู้ในหลายๆ แบบ

งานชิ้นแรกที่ได้รับมอบหมายคือการศึกษาเรื่องการปั้นโมเดลรวมทั้งเทคนิคต่างๆ ที่สำคัญ ซึ่งได้ลองกับโมเดลคน รวมทั้งการทำอนิเมชั่น ให้กับโมเดล หลังจากนั้นก็เริ่มมีงานมาเรื่อยๆ โดยที่ส่วนใหญ่จะเป็นทำโมเดล มากกว่าทำอนิเมชั่นเสียส่วนใหญ่ และเริ่มมีการกำหนดหัวข้อทำโครงการสหกิจซึ่งได้หัวข้อ “Low Carbon” หรือ การท่องเที่ยวเชิงอนุรักษ์ และพร้อมให้ฟรีเชนในทันที

3.2.1 โมเดล และ อนิเมชัน สมุดบัญชีที่สามารถเปิด-ปิด AR

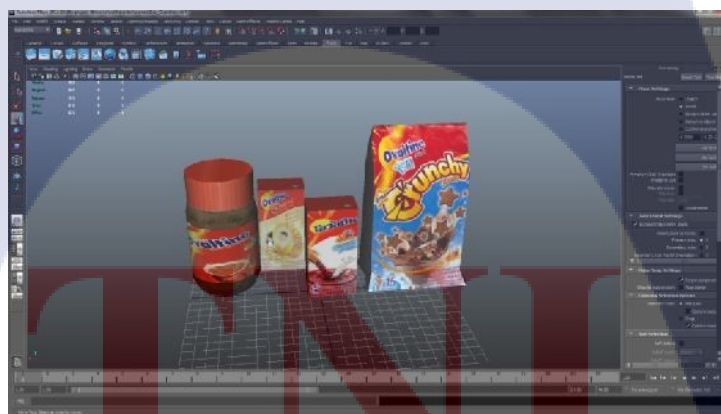
โปรแกรมที่ใช้ Adobe Photoshop, Adobe Illustrator, Autodesk Maya



รูปภาพที่ 3.1 โมเดลสมุดบัญชี

3.2.2 โมเดล ไอวอลล์ดิส AR

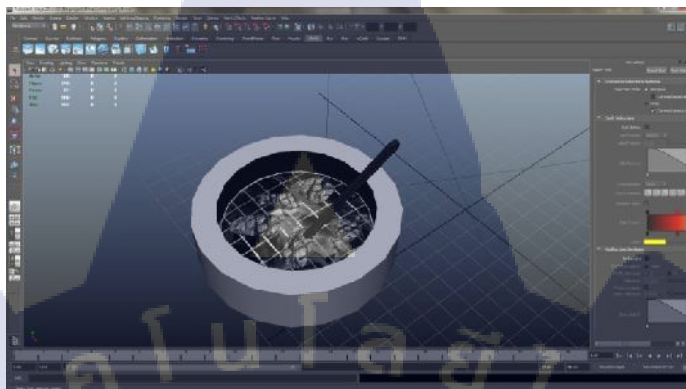
โปรแกรมที่ใช้ Adobe Photoshop, Autodesk Maya



รูปภาพที่ 3.2 โมเดลไอวอลล์ดิส

3.2.3 การออกแบบฉาก Miyabi และการทำ AR

โปรแกรมที่ใช้ Adobe Photoshop, Autodesk Maya



รูปภาพที่ 3.3 เตาช่าง

3.2.4 การออกแบบฉาก และการทำ AR

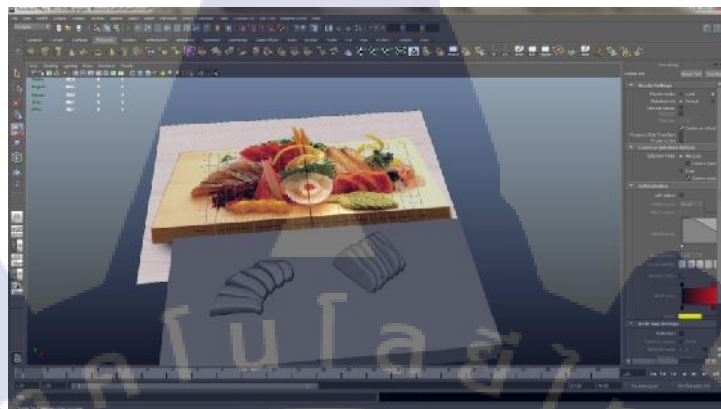
โปรแกรมที่ใช้ Adobe Photoshop, Autodesk Maya



รูปภาพที่ 3.4 โมเดลมือถือ

3.2.5 โมเดลพลาสติกสำหรับทำ AR

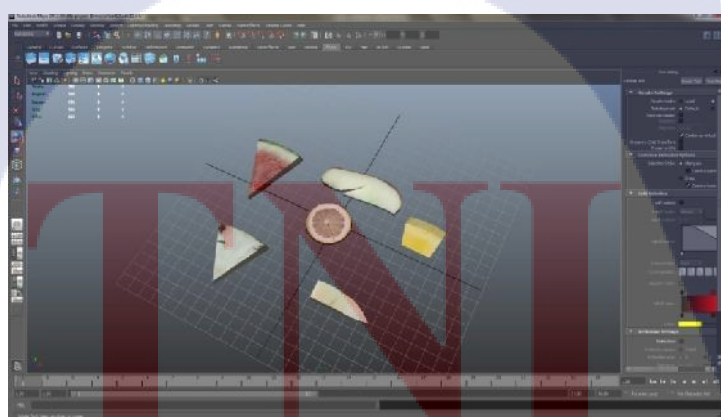
โปรแกรมที่ใช้ Autodesk Maya



รูปภาพที่ 3.5 โมเดลซูชิ

3.2.6 โมเดลผลไม้ Miyabi สำหรับทำ AR

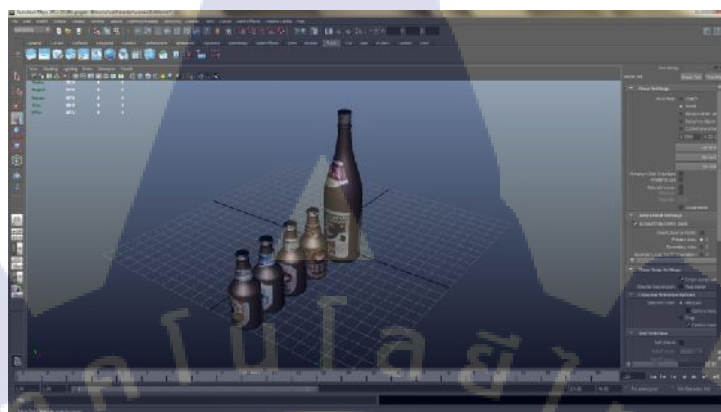
โปรแกรมที่ใช้ Adobe Photoshop, Autodesk Maya



รูปภาพที่ 3.6 โมเดลผลไม้

3.2.7 โมเดลเบียร์ Miyabi เบียร์ AR

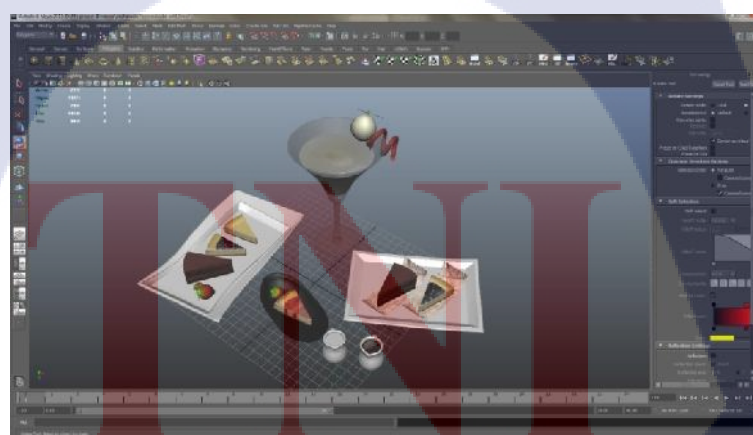
โปรแกรมที่ใช้ Adobe Photoshop, Autodesk Maya



รูปภาพที่ 3.7 โมเดลขวด

3.2.8 โมเดลของหวาน Miyabi เบียร์ AR

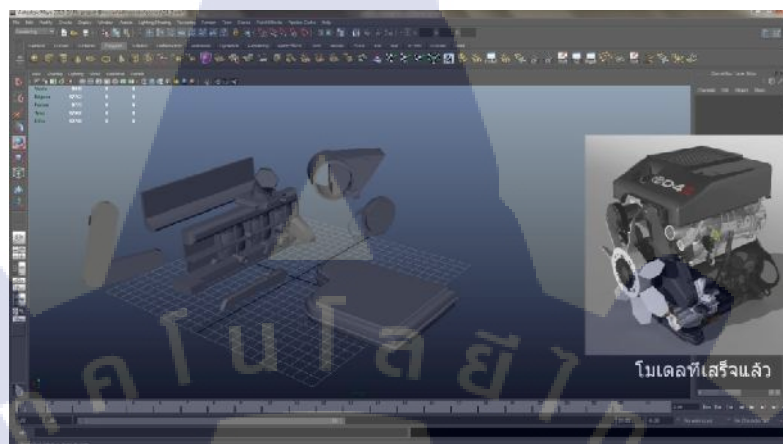
โปรแกรมที่ใช้ Adobe Photoshop, Autodesk Maya



รูปภาพที่ 3.8 โมเดลของหวาน

3.2.9 โมเดลเครื่องยนต์รถ ViGO

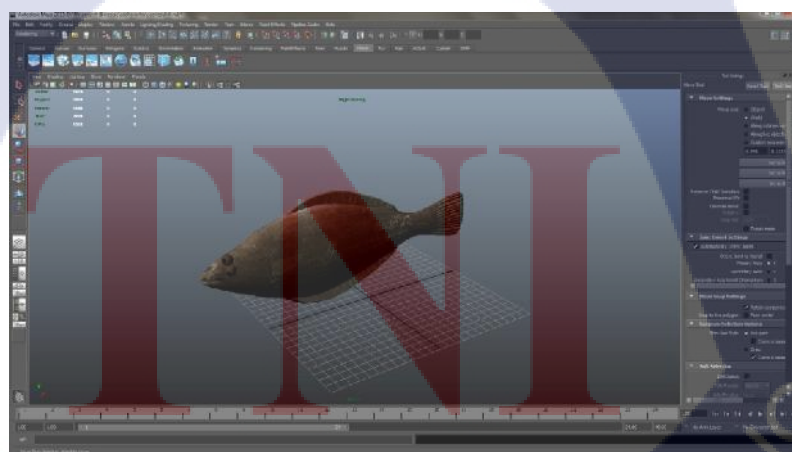
โปรแกรมที่ใช้ Autodesk Maya



รูปภาพที่ 3.9 โมเดลเครื่องยนต์

3.2.10 โมเดลปลาเตี๋ย

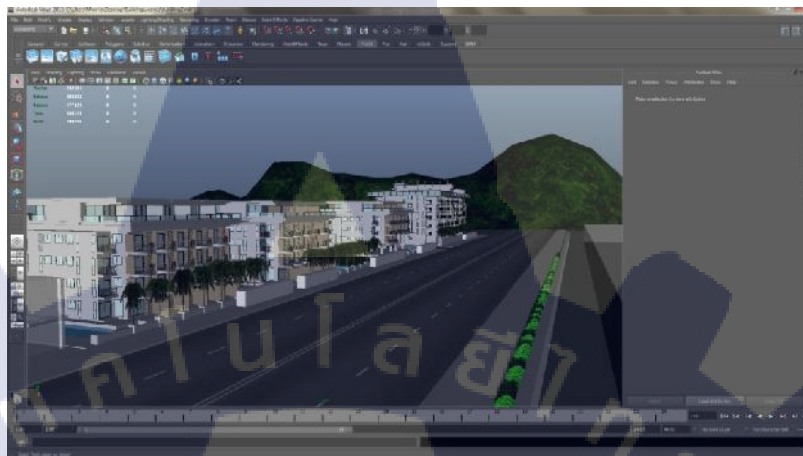
โปรแกรมที่ใช้ Adobe Photoshop, Autodesk Maya, Zbrush



รูปภาพที่ 3.10 โมเดลปลาเตี๋ย

3.2.11 โมเดลคอนโด

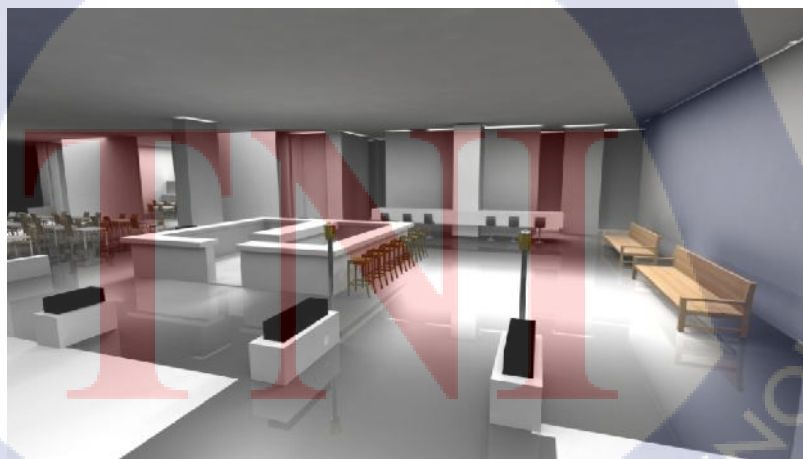
โปรแกรมที่ใช้ Adobe Photoshop, Autodesk Maya



รูปภาพที่ 3.11 โมเดลคอนโด

3.2.12 จัดแสงในห้องล็อบบี้

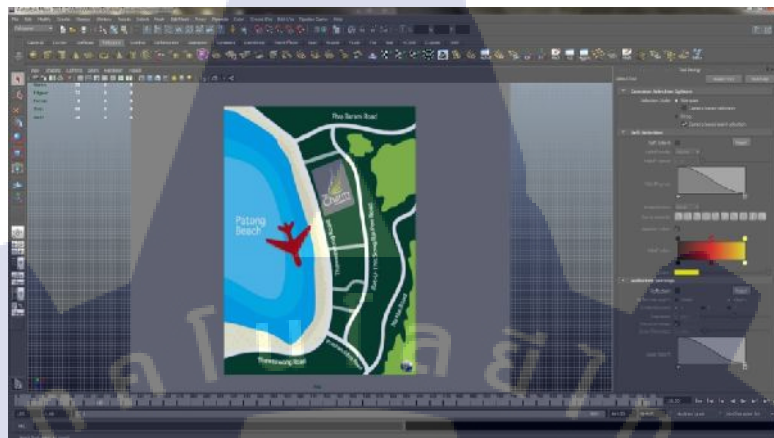
โปรแกรมที่ใช้ Autodesk Maya



รูปภาพที่ 3.12 จัดแสงโมเดลล็อบบี้

3.2.13 แผนที่คอนโดมิเนียมชั้น

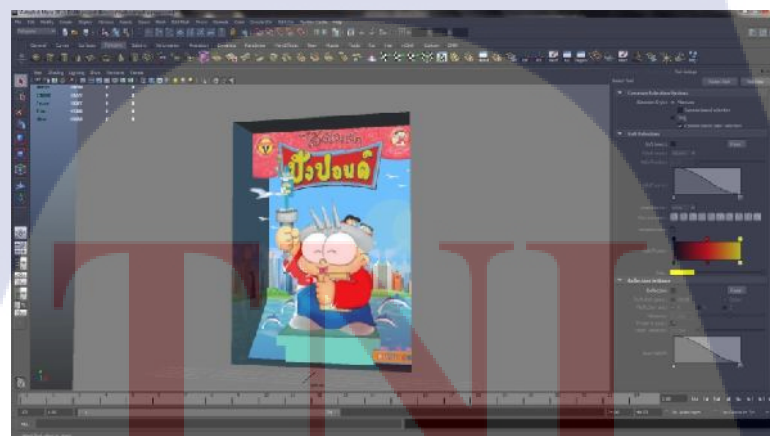
โปรแกรมที่ใช้ Autodesk Maya



รูปภาพที่ 3.13 แผนที่คอนโดมิเนียมชั้น

3.2.14 หนังสือโปสเตอร์

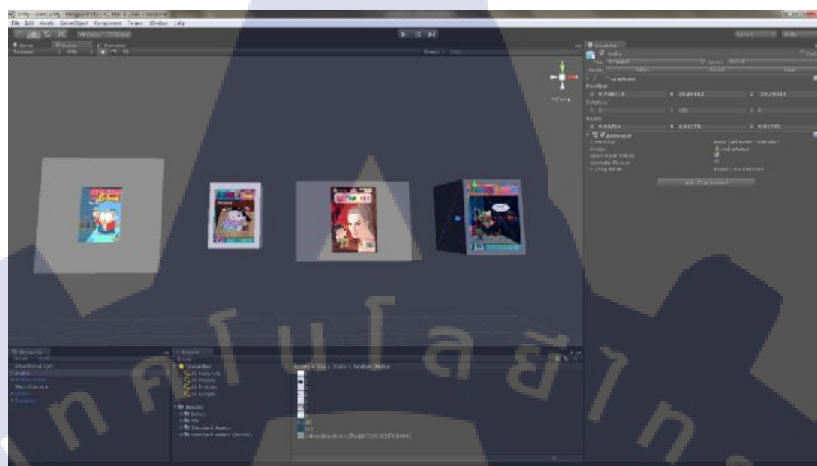
โปรแกรมที่ใช้ Autodesk Maya, Adobe Photoshop



รูปภาพที่ 3.14 โมเดลหนังสือการ์ตูน

3.2.15 จัดหนังสือการ์ตูน

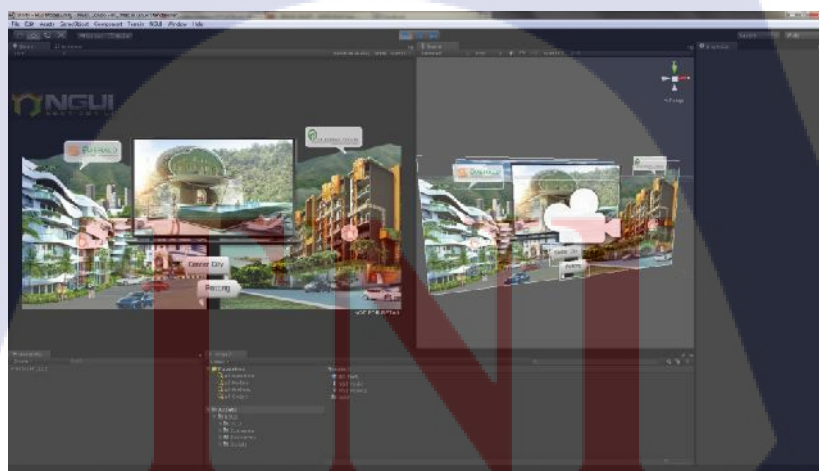
โปรแกรมที่ใช้ Unity3D



รูปภาพที่ 3.15 จัดโมเดลหนังสือการ์ตูนใน Unity

3.2.16 ทำหน้า NGUI

โปรแกรมที่ใช้ Unity3D

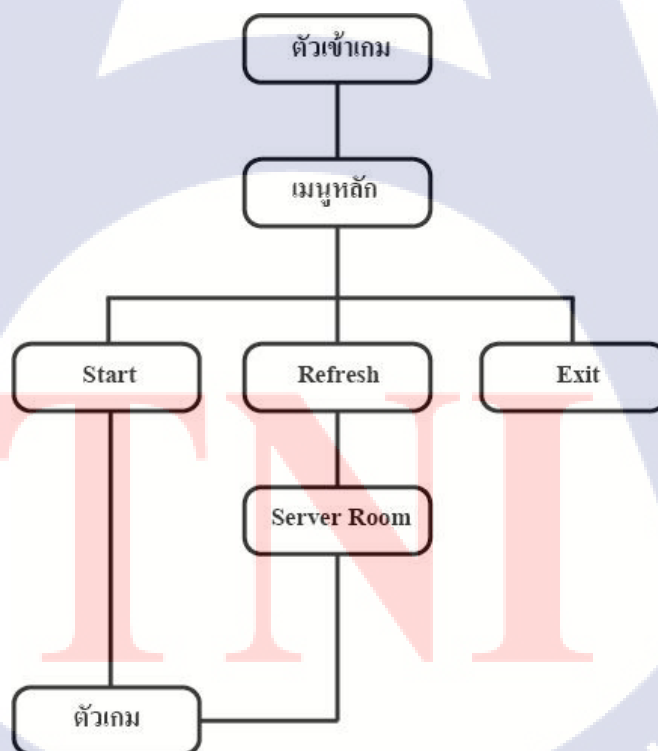


รูปภาพที่ 3.16 เมนู NGUI

3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือโค

เริ่มจากหัวข้อที่ทางบริษัทได้มอบหมายให้คือ “Low Carbon” การท่องเที่ยวเชิงอนุรักษ์ ซึ่งข้าพเจ้าคิดว่า จะทำเป็นเกมแนวป้องกันสถานที่ท่องเที่ยวจากมลพิษ

โดยตั้งต้นจากการเขียน Story board ออกแบบตัวละคร เลือกสถานที่ กำหนดฉาก และคิดเนื้อเรื่องที่จะเกิดขึ้นในเกม รวมทั้งการเขียน โค้ดในการควบคุมภายในทั้งหมด โปรแกรมที่ใช้มี Autodesk Maya 2013 , UV Layout , Adobe Photoshop CS6, Adobe Illustrator CS6 , Unity 4.0 กำหนดการทำโครงงานในช่วงต้นของเดือนแรกที่ลงมือทำคือการปั้นโมเดลทั้งหมดใน Autodesk Maya และนำมาทดสอบใน Unity 4.0 เพื่อเช็คว่ามีโมเดลที่จะผ่านหรือเปล่า พอถึงเดือนที่สอง เริ่มการลงมือเขียน Code ทั้งหมดในตัวเกมเพื่อกำหนดในส่วนต่างๆ ทั้งตัวละคร การเขียนชุดคำสั่ง AI (ชุดคำสั่งที่ทำให้วัตถุเคลื่อนที่อัตโนมัติ) และ Interface เกม ในช่วงตอนท้ายที่เริ่มลงมือทำได้เพิ่มระบบ Network ไว้สำหรับการเล่นหลายคน โดยมีการวางแผนผังดังนี้



รูปที่ 3.17 แผนผังของเกม

3.3.1 ธีมของเนื้อเรื่อง

เนื้อเรื่องโดยรวม เราจะได้รับบทเป็นเทพที่ถูกส่งมาให้ปกป้องสถานที่แห่งหนึ่ง ซึ่งถูกเหล่า มลพิษโจมตี และหวังจะทำลายสถานที่ท่องเที่ยวที่สำคัญบนโลกภายใต้การนำของมังกรดำ สถานที่ ที่เลือกใช้คือวัดเก่าในจังหวัดอยุธยา ซึ่งสอดคล้องกับหัวข้อสถานที่ท่องเที่ยว เมื่อเริ่มเกมตัวเราจะ ล่วงลงมาจากท้องฟ้าสู่พื้นที่ ที่เกิดเหตุของเรื่อง และจะมีวัดเพียงวัดเดียวที่ยังไม่ถูกโจมตี ซึ่งเราต้อง กำจัดเหล่ามลพิษที่พร้อมเข้ามาโจมตีให้หมดในแต่ละรอบ จะชนะก็ต่อเมื่อกำจัดได้หมด และห้าม ทำให้วัดถูกทำลาย

3.3.2 ธีมของกราฟิก

ต้องเริ่มตั้งต้นจากการคิดกิจกรรมหรือเหตุการณ์กรณีต่างๆ ที่จะเกิดขึ้นภายในเกม ซึ่งต้อง จัดแยกรูปแบบ และการส่งค่าในแต่ละส่วนซึ่งมีความซับซ้อนสูง และต้องมีการอัปเดตตลอดเวลา ในช่วงที่เกมเริ่มต้น โดยที่เริ่มจากการควบคุม การใช้คำสั่งบนคีย์บอร์ด และการเก็บค่าหรือข้อมูล ต่างๆ

3.3.3 การออกแบบเสียง

ภายในเนื้อเรื่องของเกมมีความจำเป็นมากที่ตรงให้ตรงทั้ง เทพ และ ตัว AI ที่จะเข้ามาโจมตี

3.3.4 การดำเนินเกมที่ปฏิบัติ

เริ่มต้นทุกอย่างต้องศึกษา ฝึก และทดลองโดยผ่านเว็บไซต์และวิดีโอสอนทุกอย่าง ไม่ว่าจะเป็น การสร้างโมเดล ทำ Texture การเขียนโค้ดหรือแม้กระทั่งทฤษฎีต่างๆ ที่จำเป็น ซึ่งในแต่ละงานต้อง ศึกษาและหา Texture หรือจัดแต่งเองทั้งหมด โดยมีพี่ๆ คอยแนะนำและสอนให้รู้ถึงเทคนิคในการ ปฏิบัติงานจริงๆ โดยจะมีงานมาให้เรื่อยๆ สลับกับการทำโครงการไปด้วยและศึกษาบางเรื่องที่คุณ สนใจ

3.4 ปัญหาและอุปสรรค

การทำโครงงานชิ้นนี้มีความยุ่งยากในการทำในส่วนของการเขียนโปรแกรมควบคุมทั้งหมดอีกทั้งยังมีระบบ Network Lan ซึ่งทำเพื่อรองรับการเล่นหลายคน และยังมีอนิเมชันในหลายจุดซึ่งจะทำให้ใช้เวลานานพอสมควร ซึ่งจำเป็นต้องมีการกำหนดและวางแผนที่ดีปัญหาที่จะเกิดจะน้อยลงมากที่สุดจึงจำเป็นต้องทำส่วนหลักที่สำคัญก่อน โดยรวมทำส่วนพื้นฐานของเกมเรียบร้อยหมดทุกอย่าง แต่ส่วน Network เป็นเพียงกรณีศึกษาเพิ่มเติม



บทที่ 4

ผลการดำเนินงาน การวิเคราะห์และสรุปผลต่างๆ

4.1 ขั้นตอนและผลการดำเนินงาน

เริ่มจากรับหัวข้อโครงการมาคิดในเบื้องต้น และข้าพเจ้ามีความสนใจในการสร้างเกมเป็นพิเศษพร้อมทั้งได้มีโอกาสลองหลายสิ่งหลายอย่าง และด้วยคำแนะนำของประธานบริษัทเรื่องด้วยการทำเกม ทำให้ข้าพเจ้ามีความสนใจและต้องการทำเกมอย่างยิ่ง จึงได้เสนอเกมที่ต้องการลงมือทำแก่บริษัท เริ่มแรกเสนอแนวทางการดำเนินงาน ต้นแบบของเกมซึ่งช่วงหลังเริ่มปรับเปลี่ยนไปเรื่อยๆ ในช่วงของการเริ่มโครงการ ได้มีการนำเสนอโดยเป็นรูปแบบ Storyboard ตัวละคร และ สถานที่ ภายในเกมเบื้องต้น และได้นำเสนอต่ออาจารย์ที่ปรึกษา และได้เริ่มโดยมีขั้นตอนดังนี้

4.1.1 รวบรวมข้อมูลและตัวอย่างเบื้องต้น

ค้นหาตัวอย่างเกมในแนวต่างๆ เพื่อดูเรื่อง Interface โดยรวม ระบบภายในเกม การจัดวางออกแบบจากโดยรวม เพื่อค้นหาแนวเกมที่ตนต้องการทำ ซึ่งแหล่งข้อมูลส่วนใหญ่มาจากเว็บไซต์ youtube.com



ภาพที่ 4.1 ตัวอย่าง VDO ที่ได้นำมาเป็นข้อมูลอ้างอิง

4.1.2 วางโครงเรื่องเบื้องต้น

วางโครงเรื่องเบื้องต้นว่าควรจะทำเนื้อเรื่องอะไร และการเล่นสามารถแสดงถึงอะไร และจะใช้สถานที่ไหนเป็นมาอยู่เกม รวมทั้งการวางตัวละคร ออกแบบทั้งหมด รวมไปถึงการคำนวณ ฟิสิกส์ โครงสร้างโมเดลอีกด้วย

4.1.3 สร้างโมเดลตัวละคร (Character)

เริ่มเรียนรู้จากเว็บไซต์ตัวอย่างการสร้างและค่อยๆ เรียนรู้ หลักการต่างๆ เทคนิค โดยเริ่มจากหาภาพตัวอย่างมาทำการตีกรอบขนาดของโมเดลที่จะสร้างขึ้น



ภาพที่ 4.2 ตัวอย่างแบบโมเดลที่เลือกมาทำ

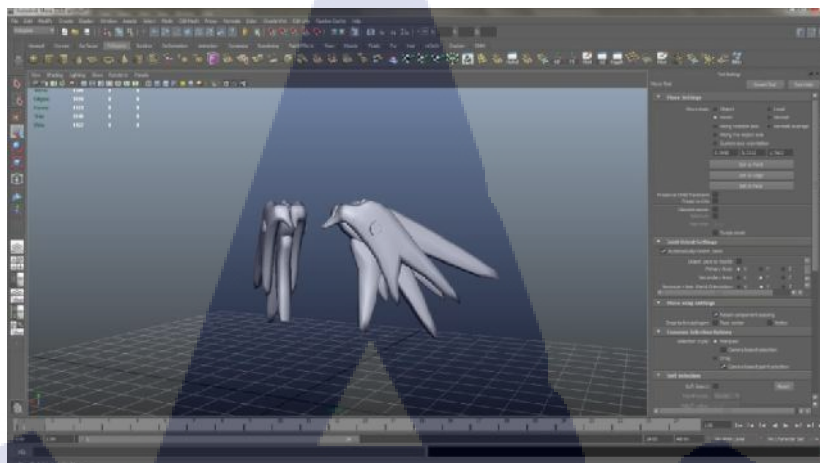
ตัวละครที่เป็นตัวหลัก ซึ่งต้องคงความเป็นเทพ ที่ถูกส่งลงมาเพื่อปกป้องสถานที่ตามเนื้อเรื่อง ข้าพเจ้าจึงได้ออกแบบมาให้เป็นตัวการ์ตูนผู้หญิง โดยมีอนิเมชันและสภาพแวดล้อมเป็นตัวอธิบายสาเหตุของการมายังสถานที่นั้น



ภาพที่ 4.3 โมเดลตัวละครที่สร้างขึ้น



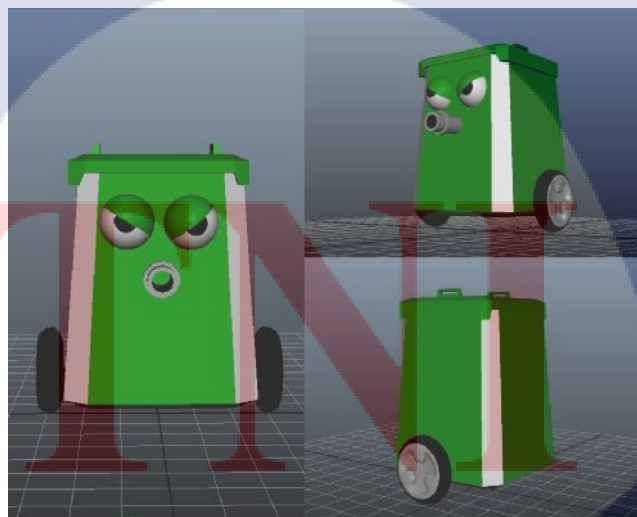
ภาพที่ 4.4 โมเดลตัวละครโดยสมบูรณ์



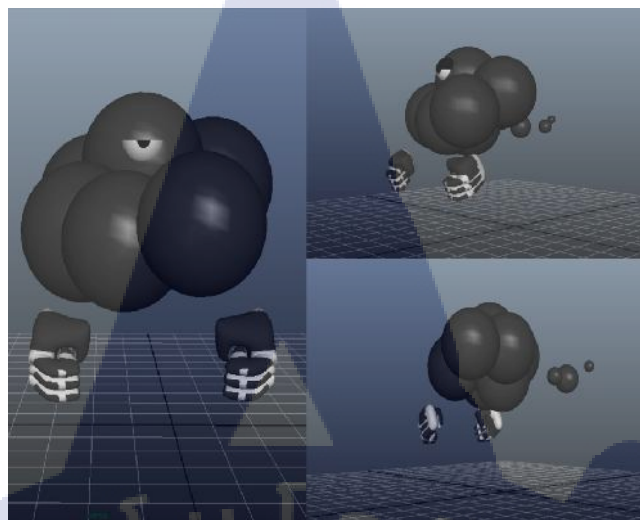
ภาพที่ 4.5 โมเดลปากของตัวละคร

4.1.4 สร้างตัวละคร AI (Artificial Intelligence)

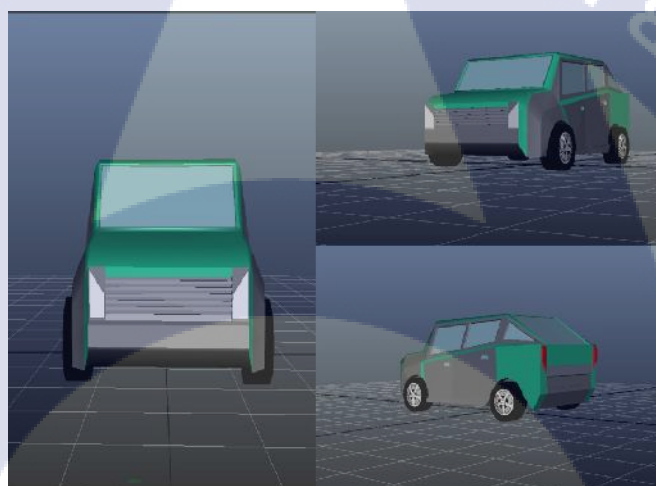
AI คือ Artificial Intelligence หรือที่เรียกสั้นๆ ว่า “บอท” ซึ่งเป็นตัวละครที่ถูกควบคุมอัตโนมัติโดยมีการกำหนดรูปแบบต่างๆ พร้อมเงื่อนไขไว้แล้วให้ระบบสั่งการด้วยตัวของมันเอง โดยที่เราไม่มีส่วนเข้าไปควบคุมได้ ซึ่งเกมทุกเกมและทุกประเภทจะต้องมี เสมอ โดยภายในเกมที่ข้าพเจ้าได้ทำขึ้นนั้นมีทั้งหมด 4 ชนิด ดังภาพต่อไปนี้



ภาพที่ 4.6 โมเดล AI ชนิดที่ 1



ภาพที่ 4.7 ภาพ AI ชนิดที่ 2



ภาพที่ 4.8 ภาพ AI ชนิดที่ 3

TNI



ภาพที่ 4.9 ภาพ AI ชนิดที่ 4

4.1.5 สร้างฉาก (Background)

ฉากที่ใช้จำเป็นมากที่ต้องเป็นสถานที่ท่องเที่ยวเพื่อให้ตรงตามหัวข้อที่ถูกกำหนดไว้ ซึ่งสถานที่นั้นมีอยู่มากมาย แต่ต้องเลือกมาเพียงที่เดียวจึงได้เลือกวัดสมยักรังรัตน โกสินทร์ จังหวัดอยุธยาซึ่งเป็นสถานที่เก่าแก่และยังเป็นสถานที่ท่องเที่ยวที่สำคัญที่สุด



ภาพที่ 4.10 ตัวอย่างสถานที่ ที่นำมาใช้สร้างฉาก



ภาพที่ 4.11 ภาพจำลอง 3D ของวัดสุทัศน์

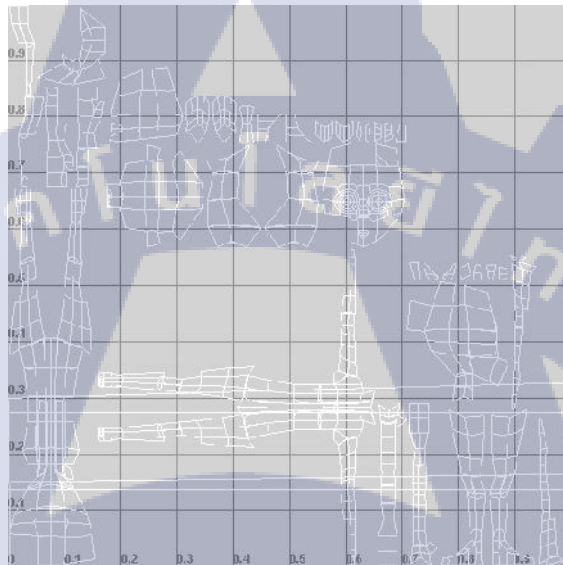


ภาพที่ 4.12 ภาพจำลอง 3D ของวัดสุทัศน์

TNI

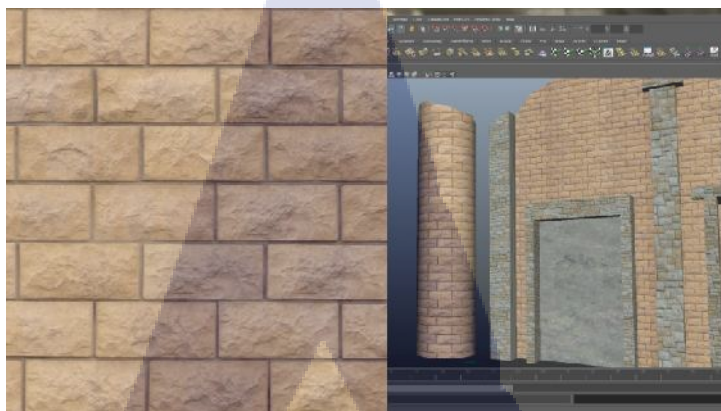
เป็นขั้นตอนที่สำคัญที่สุดก่อนการใส่พื้นผิวหรือทำลวดลายบนวัตถุหรือ โมเดล ซึ่งจะทำให้

□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□ UV □□□□□□□□



ภาพที่ 4.15 □□□□ที่ถูกทาง UV Mapping

การทำสีหรือที่เรียกกันว่า Texture ซึ่งทำให้ดูออกกว่าตัวถุนั้นคือสิ่งใด เป็นอีกสิ่งหนึ่งที่
สำคัญต่อฉากอย่างยิ่งว่า สีและสภาพผิว Texture และ UV ทุกครั้งที่ทำ

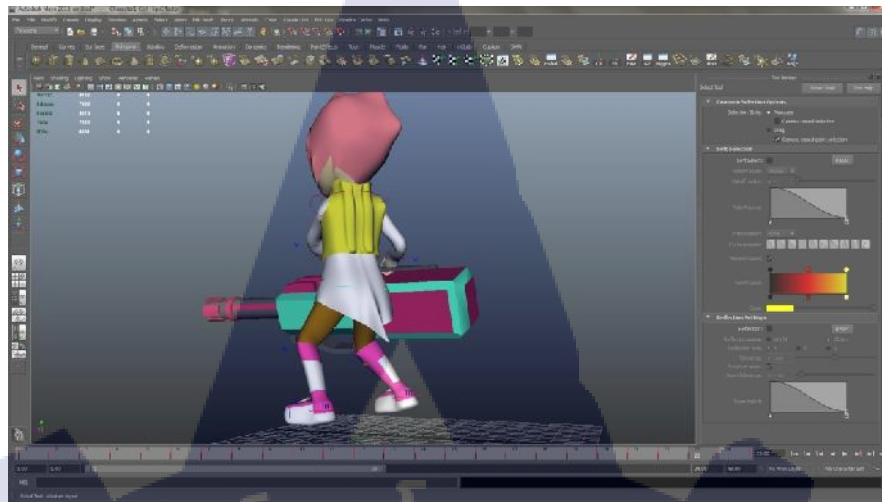


ภาพที่ 4.16 ตัวอย่าง Texture ที่ใช้

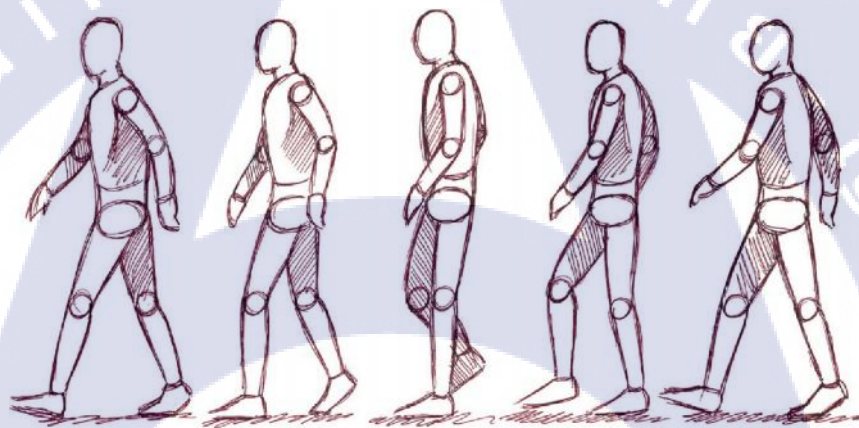


ภาพที่ 4.17 ผลที่ออกมาจากเงาหรือ Texture ที่ UV

Texture ยังมีส่วนของการทำลวดลายให้ดูพื้นผิวมีมิติอย่างการทำ Bump Map อีกชื่อว่า Normal Map ซึ่งจะช่วยให้วัตถุดูสมจริงมากขึ้น ทั้งๆ ที่ยังมีจำนวน Face เท่าเดิม ซึ่ง UV ก่อนทุกครั้งเพื่อให้ค่าต่างๆ เสียหาย หรือไม่ตรงตามลวดลาย และยังเป็นสิ่งที่จำเป็นก่อนที่จะทำสีหรือลง Texture ในโปรแกรม Zbrush ที่ช่วยในการตัด Texture ทำให้ค่าสีที่คำนวณออกมาไม่ผิดเพี้ยน



ภาพที่ 4.20 การทำอนิเมชันตัวละคร

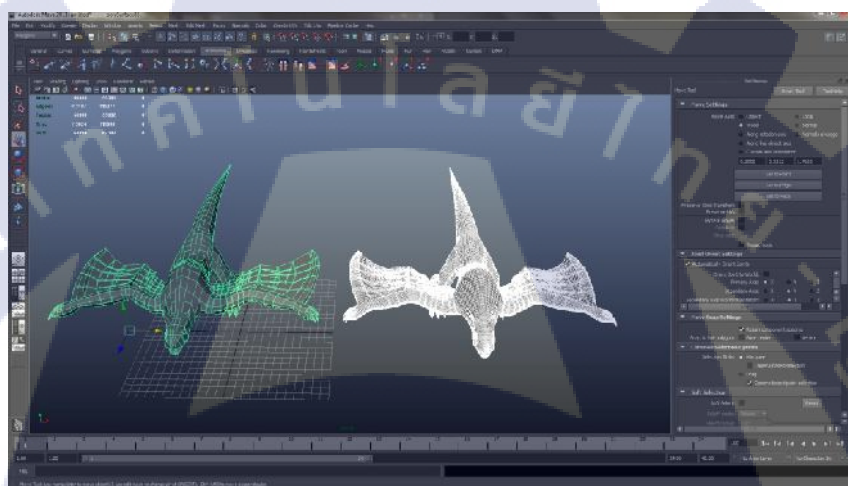


ภาพที่ 4.21 ตัวอย่างอนิเมชัน

IK หรือที่เรียกกันว่าการใส่กระดูกให้แต่ตัวละคร ซึ่งเป็นหัวใจหลักของการทำ
ท่าทางหรือเคลื่อนไหววัตถุอื่นๆ โดยต้องมีการกำหนดส่วนข้อต่อต่างๆ และการเคลื่อนไหวว่าตัว
ละครนั้นจะเคลื่อนไหวอย่างไร แล้วต้องขยับส่วนไหน อีกทั้งยังต้องศึกษาอนาโตมี
ของมนุษย์

4.1.11 การคำนวณพื้นผิวโมเดล (Face of model)

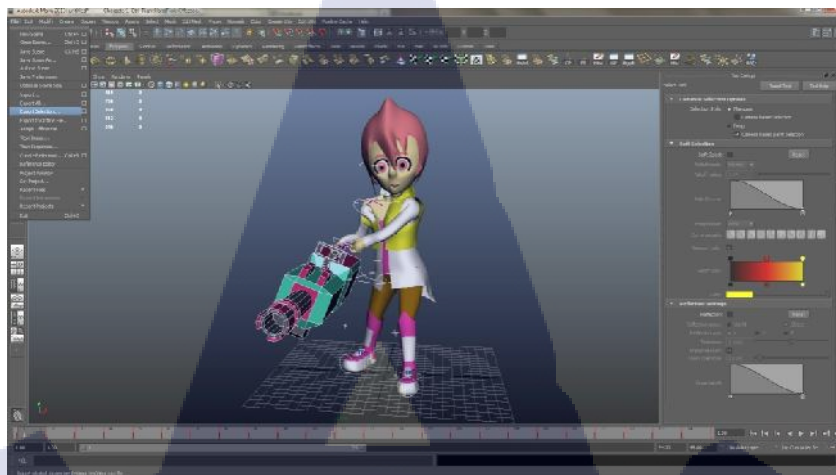
การคำนวณพื้นผิวคือหลักสำคัญอีกอย่างในการปั้นโมเดลทุกชิ้น โดยไม่ต่างจากน้ำหนักของชิ้นนั้น ยังมีจำนวน Face มาก วัตถุชิ้นนั้นก็จะหนักตามไปด้วย ทำให้มีปัญหาในการปั้นหรือย้ายไปยังโปรแกรมอื่นๆ ซึ่งจำเป็นต้องมีการคิดคำนวณค่า Face ของวัตถุชิ้นนั้น เพื่อไม่ให้เกิดปัญหา โดยเครื่องคอมพิวเตอร์จะเริ่มช้าและกระตุกขึ้นเรื่อยๆ โดยภาษาที่ใช้คือ Low Polygon กับ High Polygon ซึ่งดูได้จากรูป ด้านซ้ายคือ Low Polygon ด้านขวาคือ High Polygon



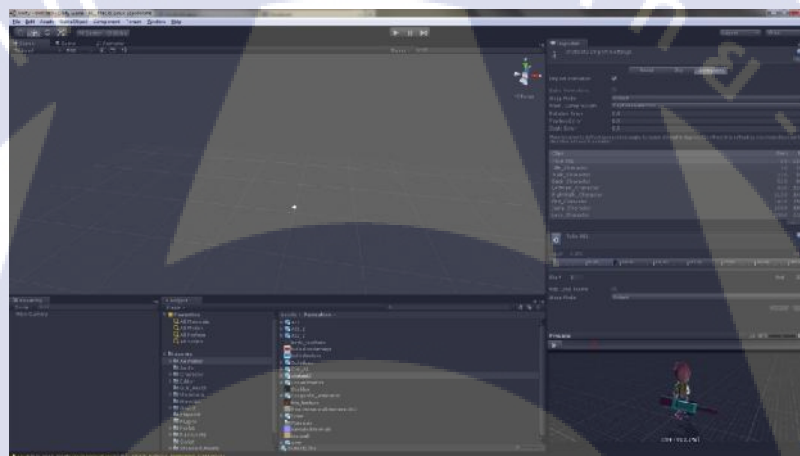
ภาพที่ 4.24 พื้นผิวโมเดล

4.1.12 การย้ายไฟล์โมเดลลงยูนิตี้ (Export file to unity)

หลังจากทำทุกอย่างที่เกี่ยวกับโมเดลเสร็จก็มาเริ่มเข้าสู่การย้ายไฟล์ลงโปรแกรมซึ่งต้องใช้ .FBX ซึ่งนามสกุลนี้ดีตรงที่จะจดจำค่าต่างๆ ของวัตถุแบบละเอียดมาก และพร้อมคำนวณเมื่อมีการใส่ Texture ในโปรแกรมอื่นเมื่อโมเดลนั้นถูกย้ายไป ไม่ว่าจะเป็น สี หรืออนิเมชันก็ตาม โดยเฉพาะอนิเมชันจำเป็นต้องมีการ Bake Simulation ทุกคีย์ที่มีการเคลื่อนไหวเสมอเพื่อไม่ให้เกิดการผิดพลาด



ภาพที่ 4.25 Export Model Maya เป็น FBX



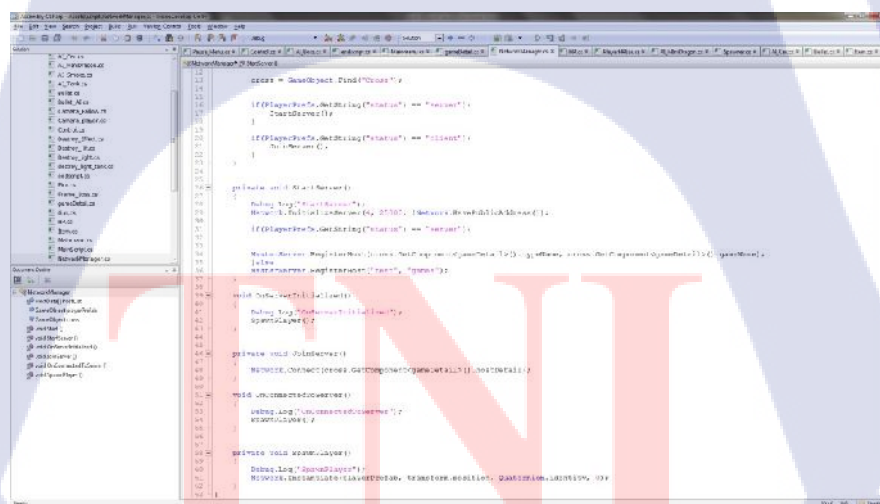
ภาพที่ 4.26 FBX ที่ได้จาก Maya พร้อมอนิเมชั่น

4.1.13 การจัดสถานที่และสภาพแวดล้อม

สภาพแวดล้อมที่ใช้เป็นรูปแบบที่เบาๆ มั่วๆ เหมือนช่วงหลังสงคราม ซึ่งท้องฟ้าจะครึ่ง มีฝนตกอีกทั้งยังมีไฟไหม้ตามจุดต่างๆ เพื่อให้เข้าใจว่า สถานที่นั้นถูกโจมตีและยังมีหมอกหนาปกคลุมตามพื้นที่เพื่อจำกัดการมองเห็นของผู้เล่น

4.1.15 การเขียนโค้ด Network

Server Network ซึ่งทำให้ผู้เล่นได้เล่นด้วยกัน มีความซับซ้อนสูงและยังมีการ
เนื่องจาก



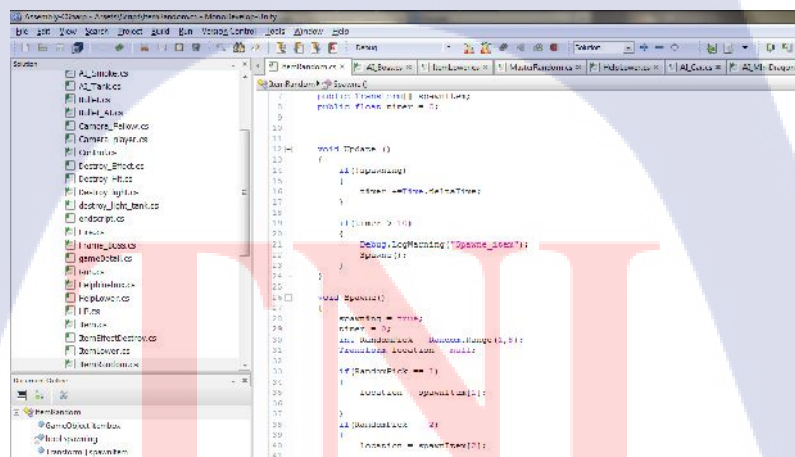
ภาพที่ 4.30 ██████████ Network Games



ภาพที่ 4.31 ผลที่ได้ออกมาจาก Network Games

4.1.16 การเขียนโค้ด Random Item

Network แต่ก็ยังสามารถทำระบบอื่นๆ ได้โดยที่ทำได้ไปคือ การสุ่มของให้เกิดขึ้นตามจุดต่างๆ ที่กำหนดไว้ โดยมีเวลาการเกิดของวัตถุ เพื่อให้ผู้เล่นสะดวกมากขึ้น โดยวัตถุที่สุ่มนั้นจะเป็นของที่ช่วยผู้เล่นได้มาก



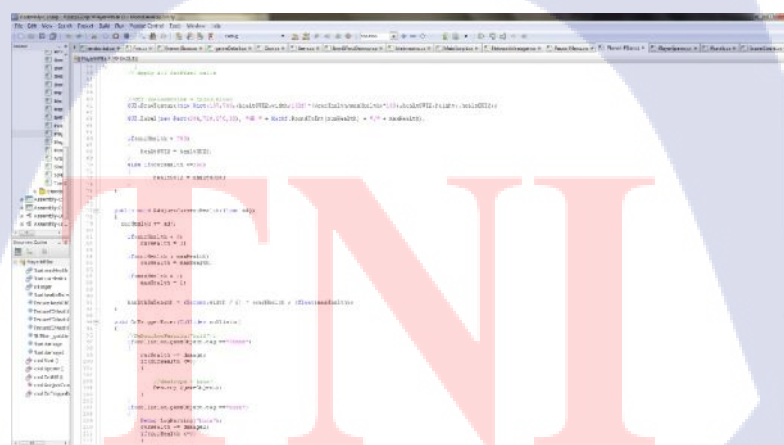
ภาพที่ 4.32



ภาพที่ 4.33 วัตถุที่ถูกส่งออกมาตามจุด

4.1.17 การเขียนโค้ดเลือด (HP BAR)

เป็นอีกโค้ดที่ต้องมีการเช็คค่าความถูกต้องตลอดเวลา เมื่อวัตถุหนึ่งส่งค่ามาให้แล้วก็ทำตามชุดคำสั่งที่เขียนไว้ โดยต้องแสดงออกมาทางหน้าจอ เป็นจุดเด่นที่ถึงขอบเขตการเล่นเมื่อผู้เล่นมีเลือดน้อยลง



ภาพที่ 4.34 วัตถุที่ถูกส่งออกมาตามจุด

4.2 ผลการวิเคราะห์ข้อมูล

ตารางที่ 4.2 ผลการวิเคราะห์

จุดประสงค์	ผลที่ได้รับ
วัตถุประสงค์ที่ 1	1. ได้รับประสบการณ์ในการทำงานจริงกับบุคคลที่ปฏิบัติงานจริง 2. ได้เรียนรู้สังคมการทำงาน ซึ่งสนุกกว่าใน 3. เรียนรู้ที่จะวางตัวให้เหมาะสมกับสถานที่ว
ความรู้ที่ได้รับ	1. ได้เรียนรู้การสร้างโมเดลที่ถูกต้อง 2. ได้เรียนรู้การสร้างโมเดลที่จำเป็นต่อการใช้ 3. ได้ศึกษาสิ่งใหม่ๆ จากเว็บไซต์ จากพี่ๆ ที่ 4. ได้เรียนรู้การสร้างโมเดลที่จำเป็นต่อการใช้ 5. ได้เรียนรู้แนวคิดจากพี่ๆ และหัวหน้าบริษัท 6. ได้เรียนรู้การสร้างโมเดลที่จำเป็นต่อการใช้ 7. แลกเปลี่ยนความรู้ความเข้าใจในสิ่งต่างๆ ที่

ตารางที่ 4.2 ผลการวิเคราะห์ (ต่อ)

	8. ได้ฝึกในด้านที่ตัวเองสนใจ และมีเวลาในการฝึกที่คุ้มค่า เพราะมีพี่ๆ ค่อยช่วยแนะนำ 9. ฝึกเขียน Code C# ฝึกเขียนโปรแกรม Java script แต่ก็มีการเขียนที่
--	---

4.3 วิเคราะห์และวิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมายในการปฏิบัติงานหรือการจัดทำโครงการ

งานสหกิจศึกษาในครั้งนี้ ได้รับมอบหมายให้สร้างเกมด้วยโปรแกรม Unity “Low Carbon” ซึ่งเป็นการศึกษาการเขียนโปรแกรมด้วยภาษา C# Code จากภาษาอื่นๆ มาอยู่ในรูปแบบของ C# โดยเริ่มจากคิดวางแผนว่าจะมีอะไรเกิดขึ้น และจำให้ทำอะไรบ้าง ซึ่งต้องศึกษา รูปแบบของแนวเกมที่ทำ มีทั้งหน้าเมนูหลัก และส่วนของ การเรียกค่าต่างๆ มาใช้ในแต่ละช่วงเวลาที่กำหนด และการคิดปัญหาที่มีความรู้ทางด้านภาษาที่ใช้เขียนน้อย จึงทำให้เสียเวลามากในการหาจุดบกพร่องการทำงาน Network และได้เพิ่มเติมส่วนบอกรอบที่อยู่ปัจจุบัน

TNI



ภาพที่ 4.42 หน้าเมนูหลักของเกม โดยเมื่อมีผู้เล่น 1 คน จะมีส่วนของการเข้าร่วมหลายคน
ขึ้นมา



ภาพที่ 4.43 ภายในเกมตอนเริ่มต้น

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลดำเนินงาน

การปฏิบัติงานในตลอด 4 เดือนที่ผ่านมา การปฏิบัติงานในส่วนที่ได้รับมอบหมายทำให้ได้รับสิ่งต่างๆ มากมาย และในช่วงที่ได้ดำเนิน โครงการงาน ซึ่งต้องทำในหลายๆ ด้าน เช่น สร้างโมเดล ออกแบบตัวละคร ฉาก หรือแม้กระทั่งการเขียนโค้ด ทำให้รู้ว่าต้องวางแผนให้มากกว่านี้และทำให้ ออกมาเรียบง่ายขึ้นแต่มีคุณภาพ เพราะการทำเกมเพียงเกมเดียวจำเป็นต้องมีคนในหลายๆ ด้าน เพียงคนเดียวจะทำให้งานล่าช้า ทำให้รู้จักการทำงานเป็นระบบ และได้ค้นพบเทคนิคต่างๆ ที่เกี่ยวข้องกับ การทำโมเดล และการสร้างเกมในชีวิตจริงมากขึ้น ทำให้ได้คิด ออกแบบ และพัฒนา Code ได้เอง ในมุมมองต่างๆ

จากการทดสอบผลการทำงานของโปรแกรม “Temple Defend” โดยผู้จัดทำได้นำ โปรแกรมทำการทดสอบในการเล่น สรุปได้ว่า ระบบพื้นฐานสามารถใช้งานได้เรียบร้อยและเป็น ระเบียบ ตามกำหนดที่คาดหวังไว้ แต่ยังคงสภาพปัญหาของการเล่นผ่าน Network ซึ่งมีความ ซับซ้อนทางด้านระบบอย่างมาก แต่ตัวเกมก็ยังสามารถดำเนินต่อไปได้ นอกจากนี้ยังมีการแสดง หน้าต่างตอนจบเกมเพื่อให้ผู้เล่นรับทราบได้

5.2 แนวทางการแก้ไขปัญหา

ตลอดระยะเวลา 4 เดือน ทำให้พบเจอกับปัญหาต่างๆ หลากรูปแบบแบบในการดำเนินงาน ถึงแม้จะเป็นการเขียนและออกแบบเกมโดยปัญหาที่พบในระหว่างการทำงาน ดังต่อไปนี้

5.2.1 ปัญหา สีของโมเดลไม่ตรง

แนวทางการแก้ไข ต้องทำการกาง UV ให้แก่โมเดลทุกชิ้นจะทำให้จำนวนค่าสีได้ง่าย และไม่เพี้ยนไปจากเดิม

5.2.2 ปัญหา อนิเมชันของตัวละครไม่สามารถเล่นได้ใน Unity

แนวทางการแก้ไข ทำการสร้าง IK(กระดูก)ใหม่ทั้งหมดตั้งแต่ต้น

5.2.3 ปัญหา ตัวโมเดลนามมีขนาดหนักและใหญ่เกินไป

แนวทางการแก้ไข ลดจำนวนการใช้เส้น แล้ว face(พื้นผิวในแต่ละช่องของโมเดล) ของ โมเดลลงเพื่อลดการใช้ทรัพยากร และการคำนวณของ โปรแกรม

5.2.4 ปัญหา ทางด้าน Code ตัวละครในการบังคับ

แนวทางการแก้ไข ศึกษาตามเว็บไซต์แล้วค่อยๆ แก้ไขและปรับมาจนสามารถเดิน และบังคับตัวละครได้

5.2.5 ปัญหา ด้านไม่เข้าใจ Code C#

แนวทางการแก้ไข ศึกษาตามเว็บไซต์ และให้พี่ที่บริษัททาง Code สอนหลักและความเข้าใจ

5.2.6 ปัญหา ด้านการใช้ GUI

แนวทางการแก้ไข พี่ที่บริษัททาง Code สอนหลักการใช้

5.2.7 ปัญหา ส่วนของการเล่นผ่าน Network

แนวทางการแก้ไข เปลี่ยนการส่งค่าระหว่าง Host และ Client ใหม่

5.3 ข้อเสนอแนะจากการดำเนินงาน

จากการดำเนินงานที่ผ่านมาทำให้ทราบว่า จะต้องศึกษาในเรื่องการคิดคำนวณในการสร้าง โมเดลและการออกแบบ Logic ในการเขียน Code โดยใช้ทั้ง Java Script และ C# รวมถึงการใช้ เครื่องมือต่างๆ ในโปรแกรม Unity และ Autodesk Maya จะได้ช่วยให้งานเสร็จไวยิ่งขึ้น หากรู้และ เข้าใจ

ในการพัฒนาโปรแกรมเกมนี้ต่อไป ควรปรับปรุงรูปแบบของส่วนการเล่นผ่าน Network ของผู้เล่นให้มีการแสดงผลที่ตรงมากขึ้น และทำการเชื่อมโยงทุกสิ่งในเกมลงบน Network

อ้างอิง

1. krunitat, 2011, krunitat Article [Online], Available : http://www.hrchonburi.or.th/external_newsblog.php?links=1884 [2013, September 5].
2. Jump' Kmutt, 2012, 3D Projection Mapping [Online], Available : <http://www.slideshare.net/JumpKmutt/3d-projection-mapping-12728272> [2013, September 5].
3. Banlu Kemiyatorn, 2011, Main Article/3D computer graphics/Rendering/UV Mapping [Online], Available : http://th.blender.wikia.com/wiki/Main_Article/3D_computer_graphics/Rendering/UV_Mapping [2013, September 6].
4. สอนพื้นฐานการทำ Blend Shape พื้นฐาน [Online], Available : <http://www.mesise.com/%E0%B8%AA%E0%B8%AD%E0%B8%99%E0%B8%9E%E0%B8%B7%E0%B9%89%E0%B8%99%E0%B8%90%E0%B8%B2%E0%B8%99%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%97%E0%B8%B3-blend-shape-%E0%B8%9E%E0%B8%B7%E0%B9%89%E0%B8%99%E0%B8%90/> [2013, September 6].
5. bankmono, 2013 , Animation [Online], Available : <http://bankmono.exteen.com/20101111/animation> [2013, September 6].
6. kasineepuipui, 2012, สอนการทำ Blend Shape พื้นฐาน [Online], Available : <http://kasineepuipui.wordpress.com/2012/09/07/%E0%B8%97%E0%B8%A4%E0%B8%A9%E0%B8%8E%E0%B8%B5%E0%B9%80%E0%B8%81%E0%B8%A1%E0%B8%81%E0%B8%B1%E0%B8%9A%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%A8%E0%B8%B6%E0%B8%81%E0%B8%A9%E0%B8%B2/> [2013, September 8].
7. Ratchanont SIAM.kp, 2013, สอนการทำ Blend Shape พื้นฐาน [Online], Available : <http://ratchanontc.blogspot.com/2013/02/c.html> [2013, September 9].

ภาคผนวก

TNI

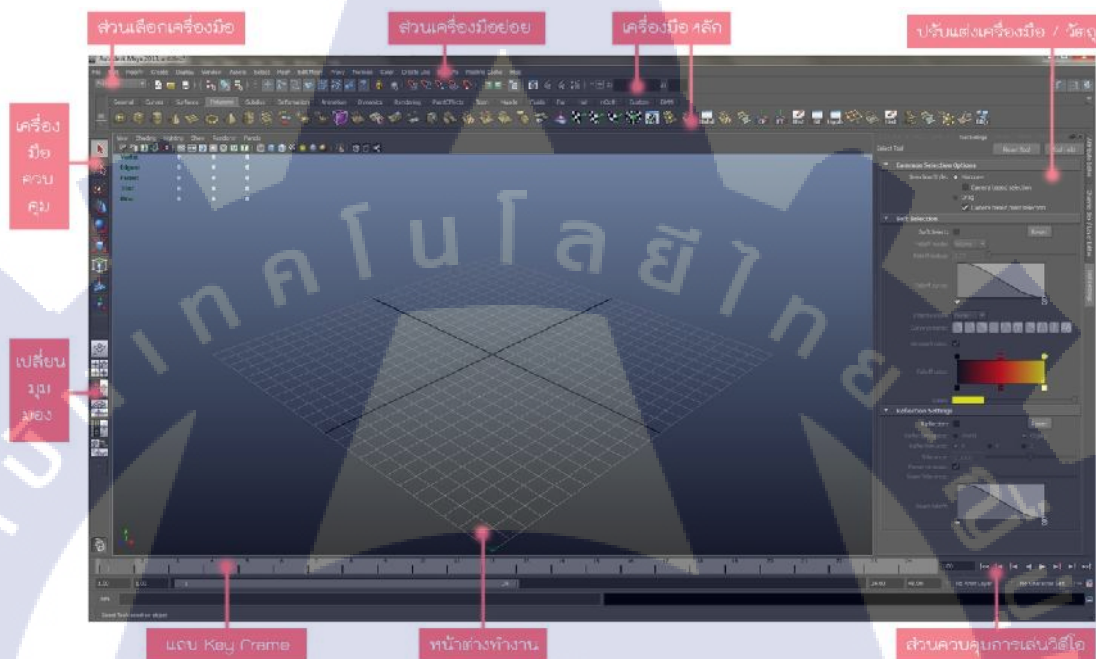
ภาคผนวก ก

การใช้งานโปรแกรม Autodesk Maya เบื้องต้น

TNI

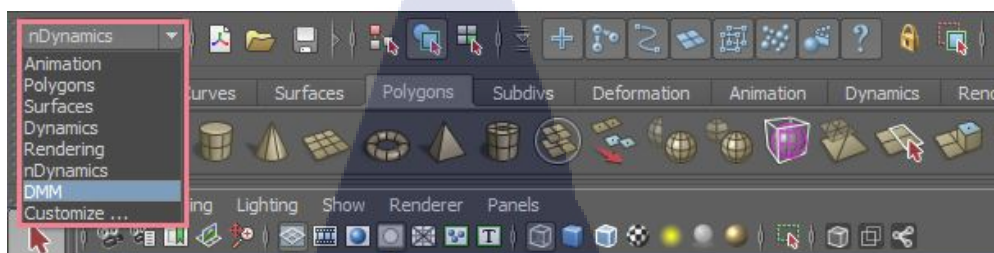
การใช้งานโปรแกรม Autodesk Maya เบื้องต้น

ส่วนประกอบของ Autodesk Maya



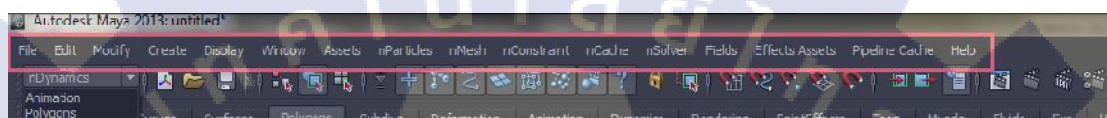
ภาพหน้าจอโปรแกรม Autodesk Maya 2013

ส่วนเลือกเครื่องมือ เป็นส่วนในการเลือกเครื่องมือที่จะใช้ โดยทุกครั้งที่เราเลือก ส่วนของเครื่องมือย่อยจะถูกเปลี่ยนตามไปด้วย โดยมีให้เลือกถึง 7 ส่วน คือ Animation, Polygon, Surface, Dynamics, Rendering, nDynamics และ Customize แต่หากเราลง Plug-in เพิ่ม ก็จะมีตัวเลือกเพิ่มขึ้นตามทีลง ในรูปจะมี DMM ที่เป็น Plug-in เสริมมา



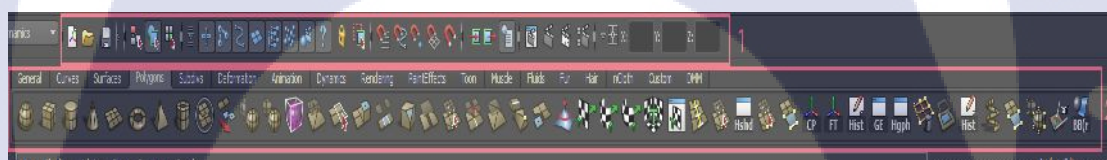
ภาพส่วนเลือกเครื่องมือโปรแกรม Autodesk Maya 2013

ส่วนเครื่องมือย่อย อยู่ในส่วนของเครื่องมือซึ่งถูกแยกออกมาให้ใช้งานตามที่ผู้ใช้เลือก หัวข้อเครื่องมือ ซึ่งมีการทำงานแตกต่างกัน



ภาพส่วนเครื่องมือย่อยโปรแกรม Autodesk Maya 2013

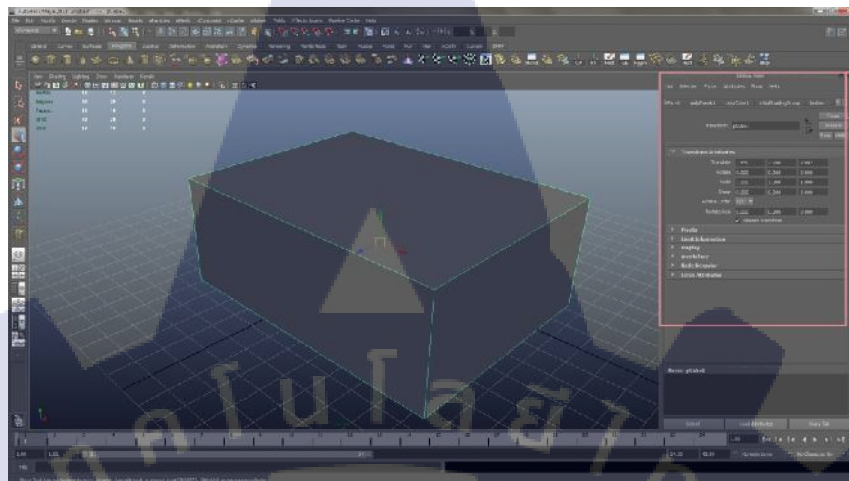
เครื่องมือหลัก เป็นเครื่องมือหลักในการใช้งานเช่น เปิดไฟล์งาน เซฟงาน หรือการเรนเดอร์งาน โดยมี 2 ส่วนหลัก ซึ่งส่วนแรกจะอยู่ช่องกลาง และส่วนที่สองจะอยู่ช่วงล่างแต่ละเปลี่ยนเครื่องมือย่อยการใช้งานได้



ภาพส่วนเครื่องมือหลักโปรแกรม Autodesk Maya 2013

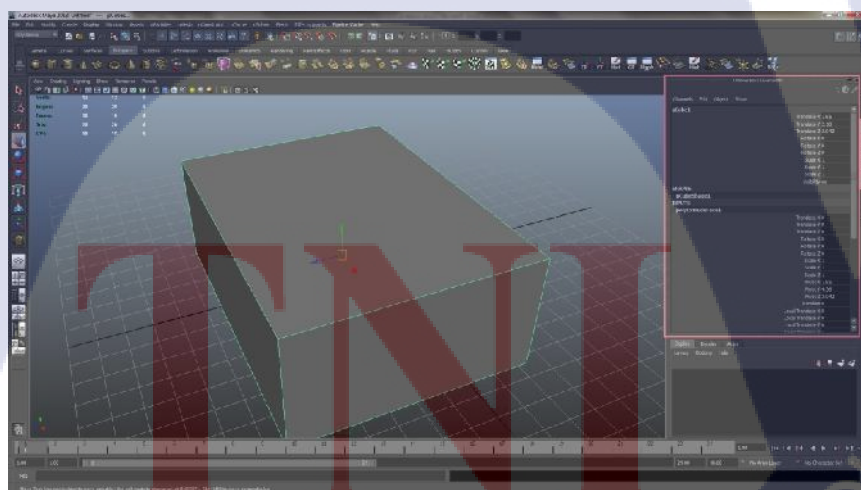
ปรับแต่งเครื่องมือ / วัตถุ ไว้ใช้ปรับแต่งค่าต่างๆ เพื่อการใช้งานโดยจะมี 3 ส่วนแยกออกจากกัน ได้แก่ Attribute Editor, Channel Box, Tool Setting สามารถดูได้จากตัวอย่างต่อไปนี้

1. Attribute Editor เป็นส่วนจัดการค่าของวัตถุในเรื่องของ Texture หรือการเปลี่ยนแปลง ตั้งแต่วัตถุ หรือการดึงพื้นผิวอย่าง Extrude ขึ้นต่อเมื่อมีวัตถุแล้ว



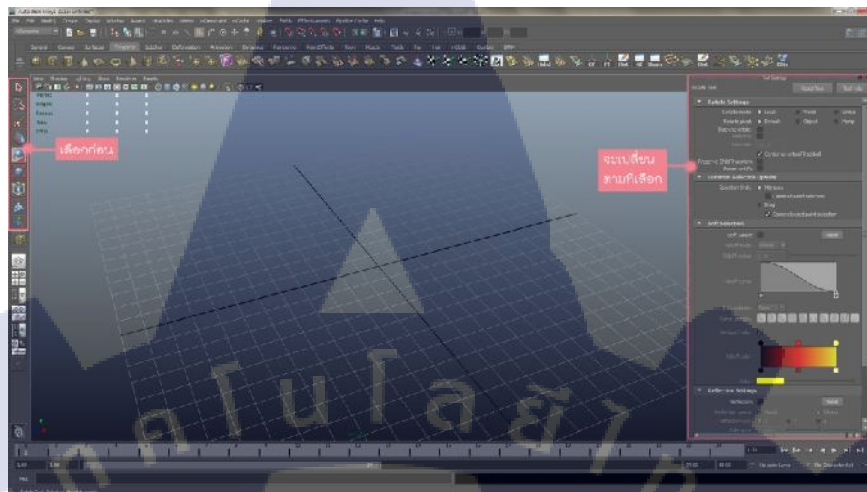
ภาพส่วนหน้าต่าง Attribute Editor

2. Channel Box ส่วนในการตั้งค่าเบื้องต้นของวัตถุ เช่น เพิ่มเส้น ปรับขนาด ขยายตำแหน่ง หรือ หมุนวัตถุ ซึ่งจะมีค่าขึ้นก็ต่อเมื่อเราไปคลิกที่วัตถุแล้ว



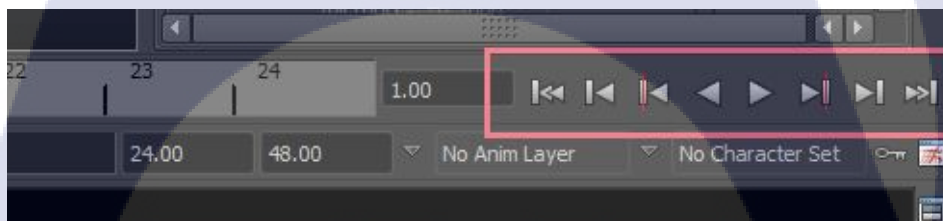
ภาพส่วนหน้าต่าง Channel Box

3. Tool Setting ส่วนปรับแต่งค่าของการควบคุม เปลี่ยนแกน หรือแม้แต่การดึงพื้นผิวของวัตถุ ซึ่งจะเปลี่ยนไปตามการควบคุมที่เลือก



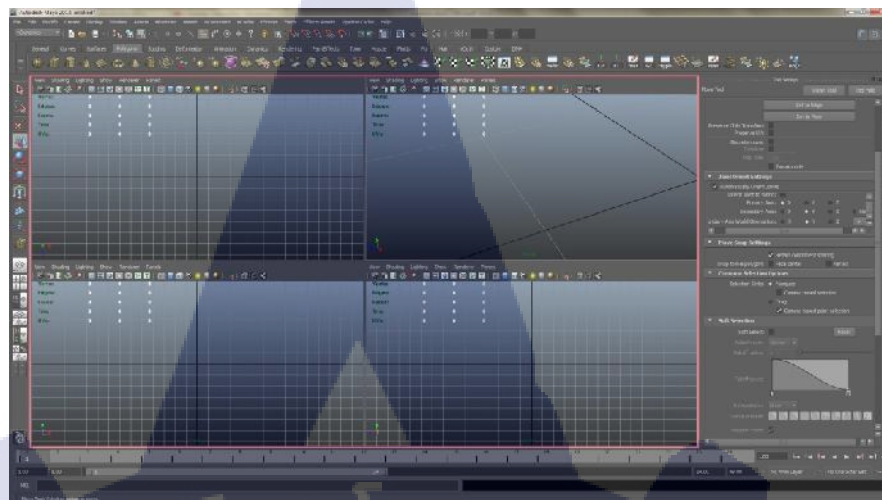
ภาพส่วนหน้าต่าง Tool Setting

ส่วนควบคุมการเล่นวิดีโอ ควบคุมการเล่น Animation ในแต่ละคีย์เฟรม สามารถเล่นเดินหน้า หรือ ย้อนหลังได้



ภาพส่วนควบคุมการเล่นวิดีโอ

หน้าต่างทำงาน เป็นหน้าต่างหลักซึ่งส่วนใหญ่จะอยู่ในมุมมอง Perspective ซึ่งสามารถหมุนได้รอบด้าน แต่สามารถเปลี่ยนมุมมองได้เพียงกด Space bar จะกลายเป็น 4 มุมมอง คือ Top, Side, Front และ Perspective เพื่อใช้ในการมองวัตถุ



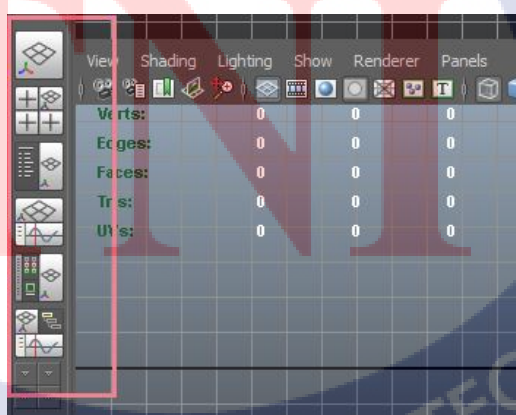
ภาพหน้าต่างทำงาน

แถบ Key Frame ส่วนในการกำหนดช่วงเวลาการทำ Animation เพียงแค่เลือกช่วงเฟรมที่ต้องการแล้วกด S เพื่อทำการเลือกเฟรมนั้นและจะเซฟค่าของวัตถุที่ถูกขยับในตอนนั้นทันที



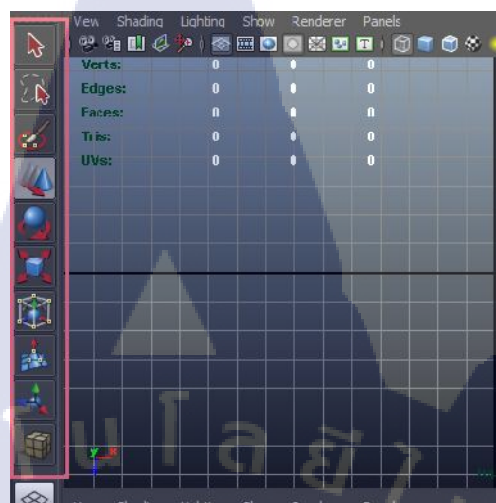
ภาพแถบ Key Frame

เปลี่ยนมุมมอง ใช้ในการจัดหน้าต่างทำงานที่ต้องการจะดูงานที่ทำ ในหลายๆแบบ หรือหลายๆ มุม สามารถเลือกกำหนดได้



ภาพเปลี่ยนมุมมอง

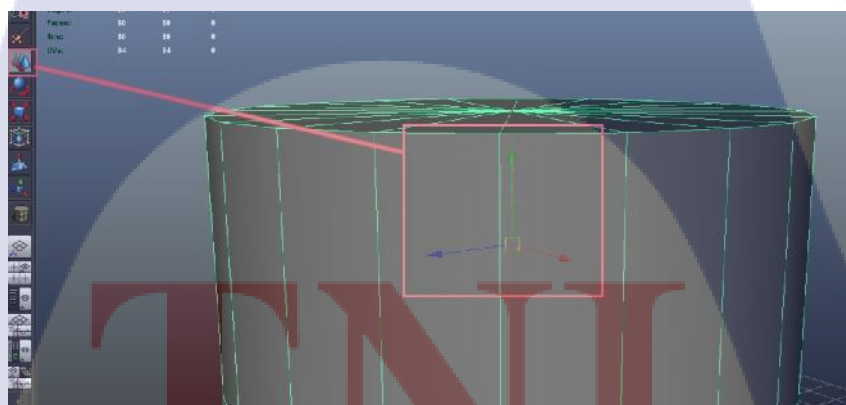
เครื่องมือควบคุม ใช้ในการควบคุมวัตถุโดยสามารถเลือกการควบคุมได้



ภาพเครื่องมือควบคุม

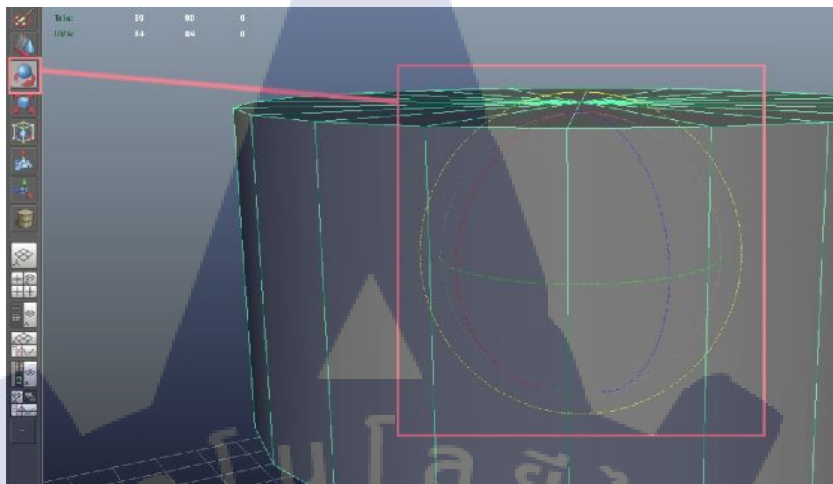
ในส่วนเครื่องมือควบคุมสามารถแบ่งคำสั่งหลักๆ ที่จำเป็นในการใช้งานได้

1. Move Tool เป็นการเคลื่อนย้ายวัตถุ โดยมีคีย์ลัดคือ W



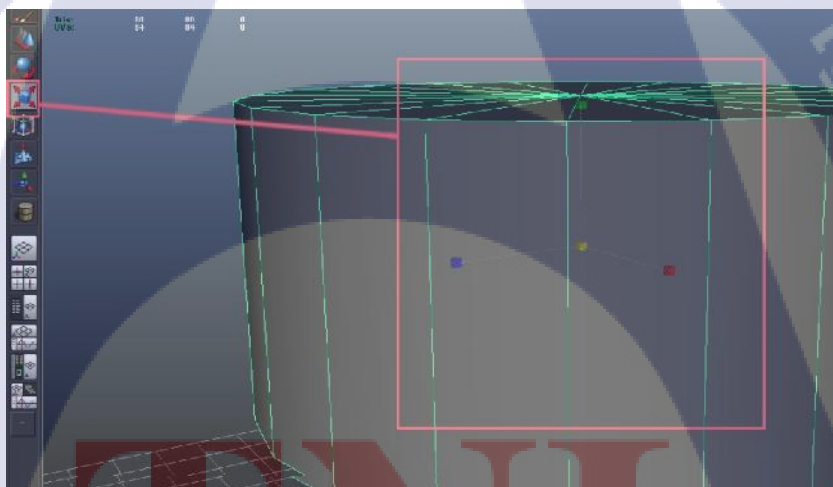
ภาพคำสั่ง Move

2. Rotate Tool เป็นการหมุนวัตถุ โดยมีคีย์ลัดคือ E



ภาพคำสั่ง Rotate

3. Scale Tool เป็นการย่อ หรือ ขยายวัตถุ โดยมีคีย์ลัดคือ R



ภาพคำสั่ง Scale

คีย์ลัดการควบคุมมุมมอง

1. Zoom ใช้ด้วยการกด Alt + Mouse Right Click [คลิกเมาส์ขวา] พร้อมกันแล้วเลื่อนเมาส์ไปด้วย หากเลื่อนขึ้นลงจะซูมเข้า/ไปช้าๆ แต่ถ้าซ้ายขวา จะซูมได้เร็วและลึกขึ้น
2. Move ใช้ด้วยการกด Alt + Scroll Mouse [ลูกกลิ้งเมาส์กลาง] พร้อมกันแล้วเลื่อนเมาส์ไปด้วย จะทำการย้ายมุมมองการทำงาน

3. Rotate ใช้ด้วยการกด Alt + Mouse Left Click [คลิกเมาส์ซ้าย] พร้อมกันแล้วเลื่อนเมาส์ไปด้วย จะทำการหมุนมุมมองการทำงาน

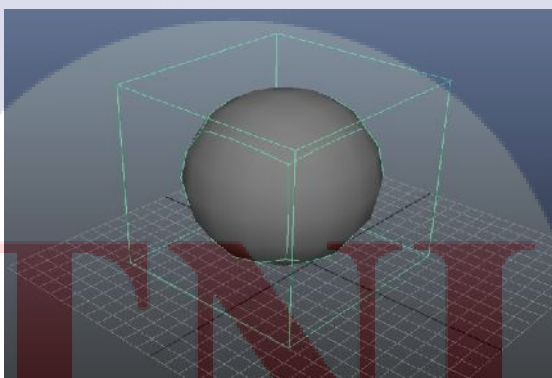
คีย์ลัดการแสดงผล

1. กดปุ่ม 1 แสดงผลปกติ



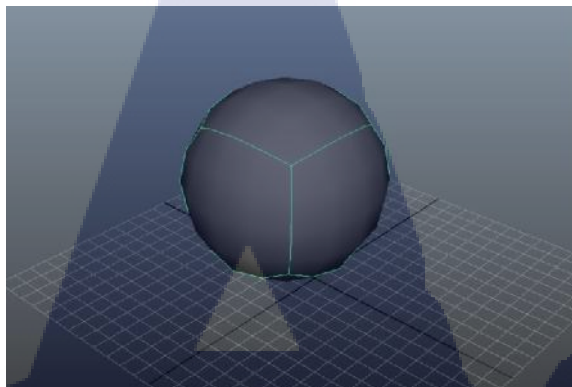
ภาพการแสดงผลปกติ

2. กดปุ่ม 2 แสดงผลวัตถุตอนที่โดนทำ Smooth แต่ยังมีเส้นบอกโครงเดิมก่อนใส่



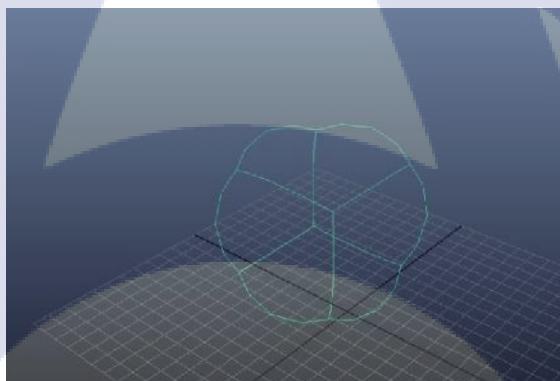
ภาพการแสดงผลแบบ Smooth ที่มีโครงเก่าอยู่

3. กดปุ่ม 3 แสดงผลวัตถุตอนที่โดนทำ Smooth



ภาพการแสดงผลแบบ Smooth

4. กดปุ่ม 4 แสดงเส้นวัตถุ



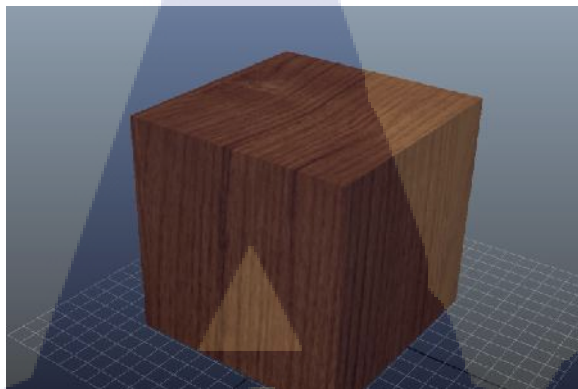
ภาพการแสดงผลเฉพาะเส้นวัตถุ

5. กดปุ่ม 5 แสดงพื้นผิววัตถุ



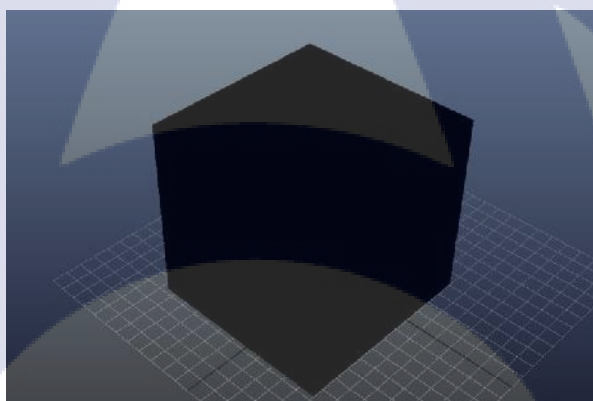
ภาพการแสดงผลพื้นผิววัตถุ

6. กดรูม 6 แสดง Texture ที่ใส่ให้กับวัตถุ



ภาพการแสดง Texture

7. กดรูม 7 แสดงเงาแต่ต้องมีการจัดแสงถึงจะเห็นวัตถุ

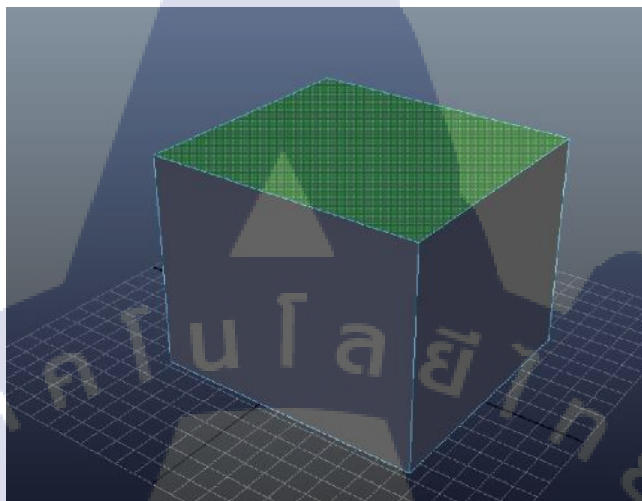


ภาพการแสดงเงา

TNI

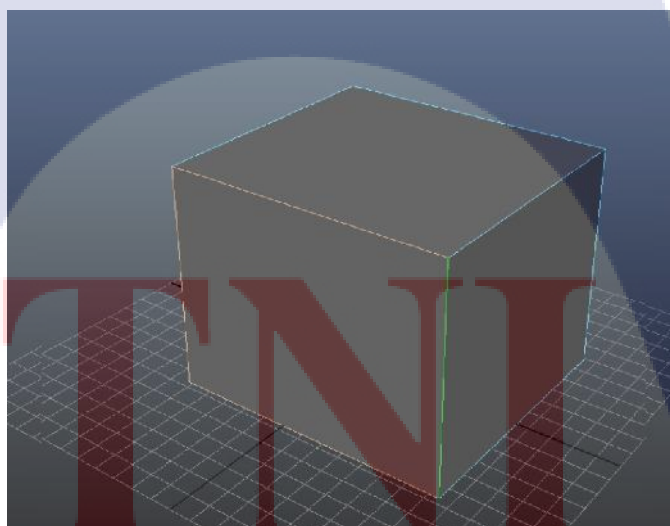
สิ่งที่ควรรู้

1. Face พื้นผิวของวัตถุ



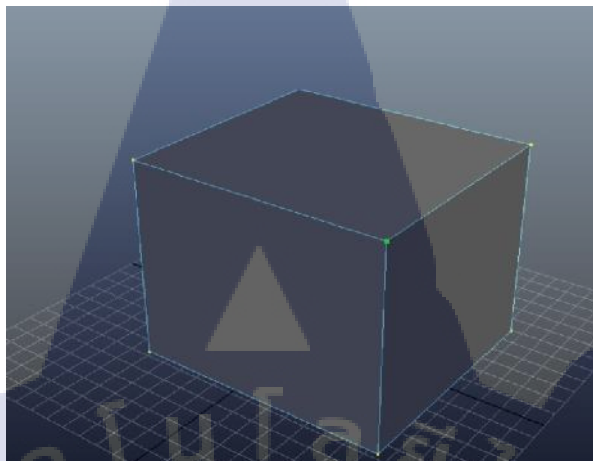
ภาพ Face ของวัตถุ

2. Edge เส้นของวัตถุ



ภาพ Edge ของวัตถุ

3. Vertex จุดหรือมุมของวัตถุ



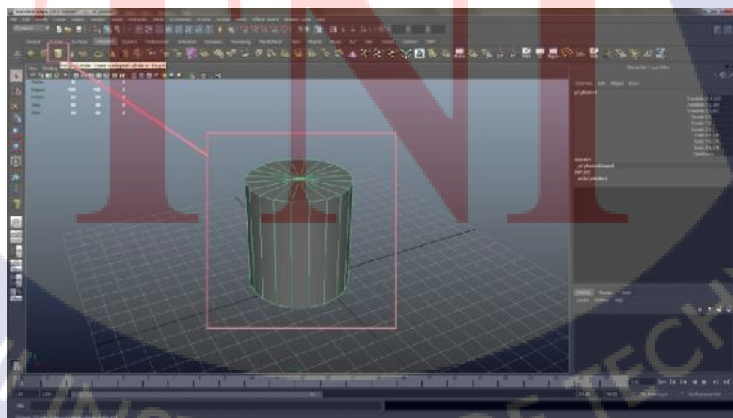
ภาพ Vertex ของวัตถุ

4. Surface การสร้างวัตถุซึ่งนิยามให้มีรูปทรงที่โค้ง

5. Polygon การสร้างวัตถุซึ่งนิยามให้มีรูปทรงที่เหลี่ยม

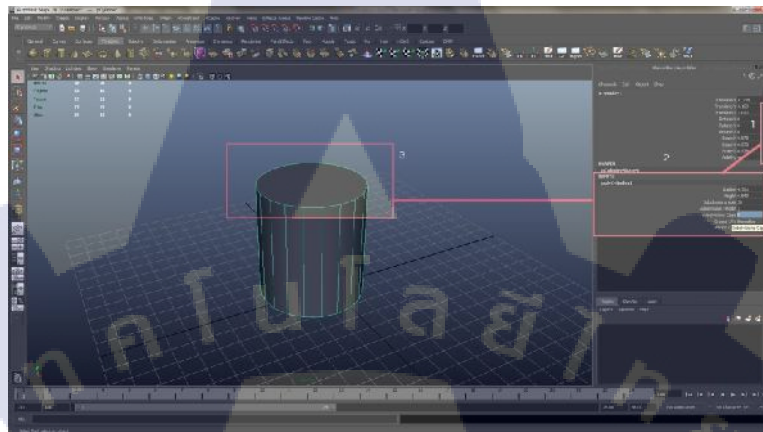
การขึ้นรูปทรงพื้นฐาน

การสร้างโมเดลสักชิ้นหนึ่งต้องคำนวณก่อนว่า สิ่งที่เราต้องการปั้นจะต้องเริ่มจากรูปทรงอะไรและต่อเติมอย่างไร เพราะรูปทรงพื้นฐานของโมเดลนั้นจะมีเพียง สี่เหลี่ยม สามเหลี่ยม ทรงกลม ทรงกระบอก ทรงกรวย และ วงแหวน อย่างเช่น ต้องการสร้างโมเดลกระป๋องน้ำอัดลมอย่าง PEPSI เราก็ต้องเริ่มจากทรงกระบอก



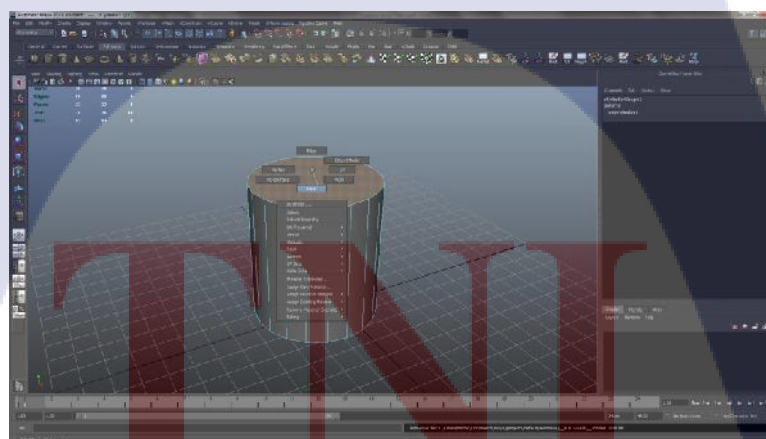
ภาพเลือกรูปทรงวัตถุ

เมื่อเราได้ทรงกระบอกแล้ว ให้กำหนดขนาดที่ต้องการจากนั้นทำการลบเส้นหัวท้ายออก โดยกดที่โมเดล แล้วไปที่ Channel Box จากนั้นจะมีส่วนของ Input ไปกดที่ Input ของวัตถุ แล้วไปใส่ค่า 0 ที่ Subdivision cap ดังภาพ



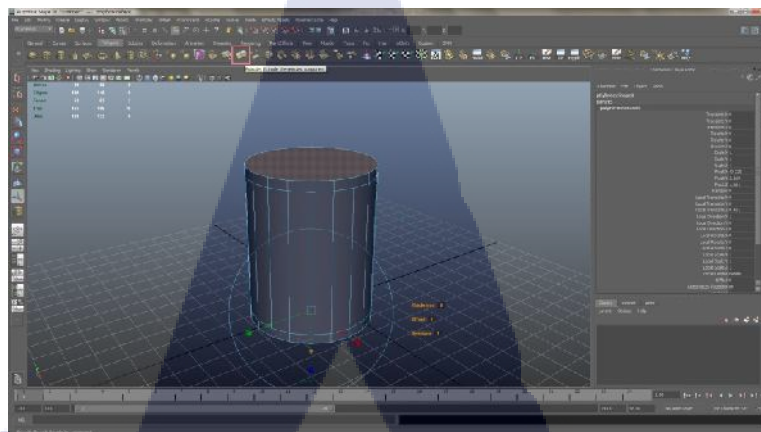
ภาพ Subdivision cap

จากนั้นดูรูปทรงของวัตถุที่เลือกคือ กระป๋อง PEPSI แล้วกลับมาที่โมเดล คลิกขวาที่โมเดล เลือกคำว่า Face (เลือกส่วนหัวกับท้าย)



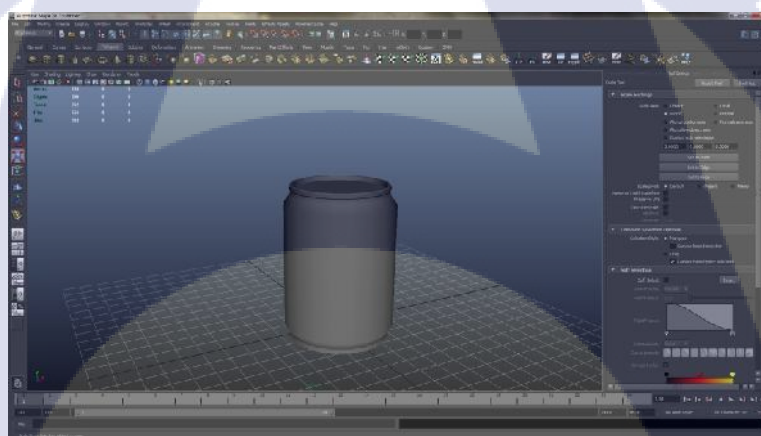
ภาพเลือก Face

จากนั้นไปที่คำสั่ง Extrude เพื่อทำการดึงพื้นผิวใหม่ขึ้นมา โดยเราจะได้หัวและท้ายที่เพิ่มขึ้นมาเพื่อใช้ทำ ปากกระป๋อง และฐานกระป๋อง



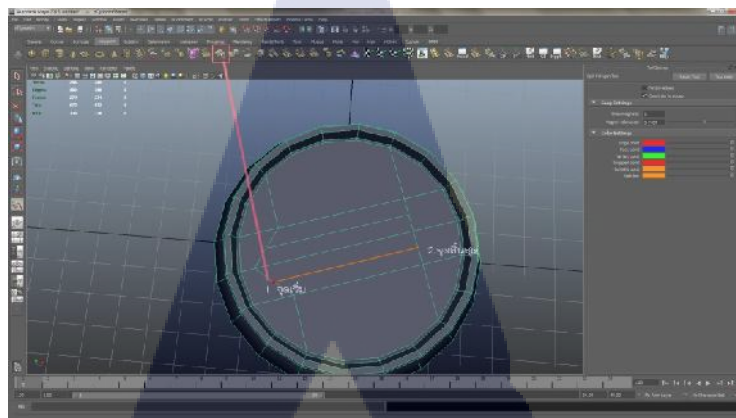
ภาพการ Extrude

จากนั้นทำการตัดรูปทรงโดยใช้ คีย์ลัด R ในการปรับรูปทรงพื้นผิวทั้งหมดและทำจนได้เป็นกระป๋อง และมีคีย์ลัด W เพื่อช่วยในการดึงพื้นผิว จะได้กระป๋องพอดี



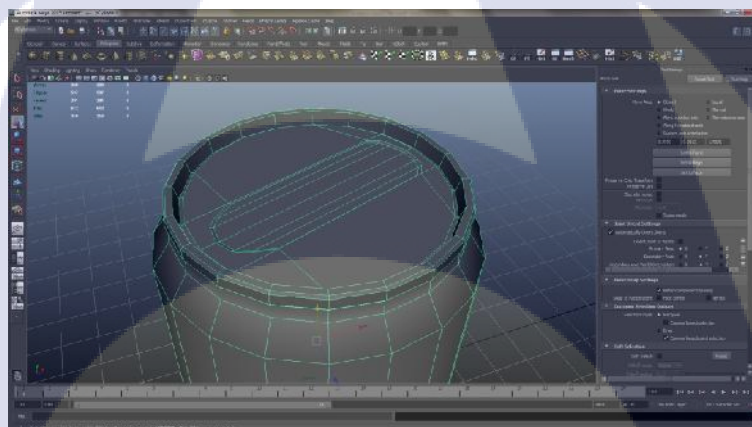
ภาพรูปทรงกระป๋อง

แล้วมาที่ส่วนบนของกระป๋อง ทำการสร้างเส้นด้วย Interactive spline ซึ่งวิธีการใช้คือ กำหนดจุดที่เส้นจะวิ่งผ่าน โดยมีจุดเริ่มต้น และจุดสิ้นสุด คลิกจุดที่ต้องการ จากนั้นลากเส้นไปยังจุดสิ้นสุด ซึ่งจะมีเส้นสีส้มๆ เป็นตัวบอกการตัดผ่านของเส้นที่จะสร้าง พอได้จุดสุดท้ายก็คลิกเพื่อวางเส้นจากนั้น กด Enter จะได้เส้นตามสร้าง



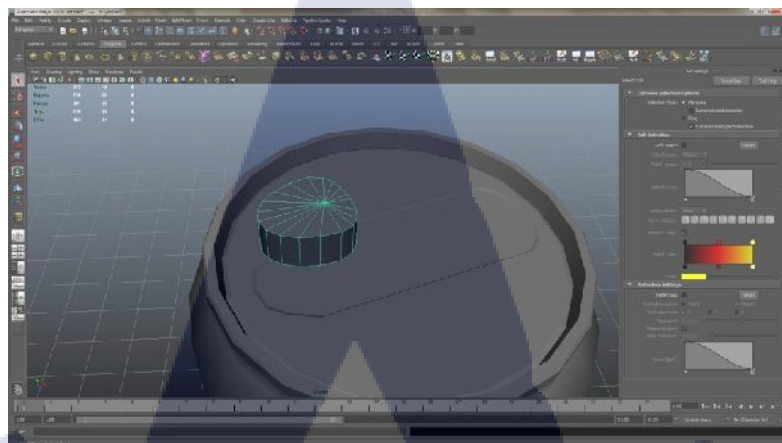
ภาพการสร้างเส้นด้วย Interactive split

ค่อยๆ จัดเส้น คัดทรงและทำการ Extrude เพื่อสร้างรอยบุบที่ส่วนบนกระป๋องจะได้รูปดังต่อไปนี้



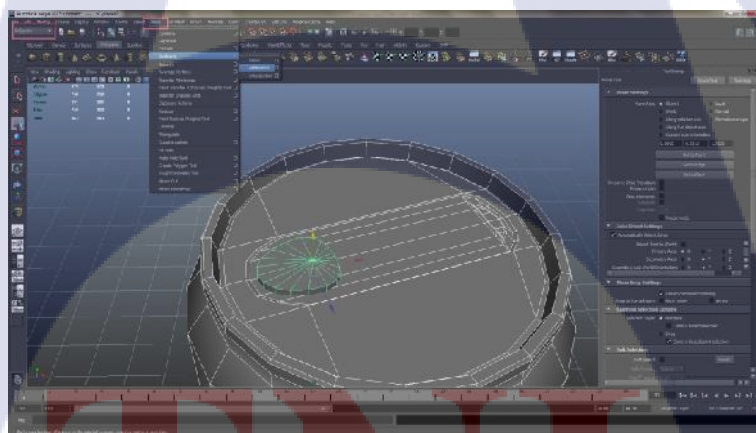
ภาพสร้างรอยบุบบนปากกระป๋อง

จากนั้น สร้างทรงกระบอกมา 1 อันแล้วทำการคัดทรงให้เป็นรูปของปากกระป๋อง เพื่อที่จะเจาะรูบนปากกระป๋อง แล้วนำมาวางไว้บนส่วนที่จะเจาะรู



ภาพสร้างโมเดลปากกระป๋อง

หลังจากนั้นให้กดที่โมเดลกระป๋องก่อนแล้วกด Shift ค้างไปกดที่ปากกระป๋องที่ทำไว้เพื่อเป็นการเลือก 2 ชิ้น ไปที่คำสั่ง Mesh > Booleans > Difference เพื่อทำการเจาะรู ถ้าไม่เจอ Mesh ในส่วนเครื่องมือย่อย ให้ไปเลือกเมนูหลักเป็น Polygon เสียก่อน

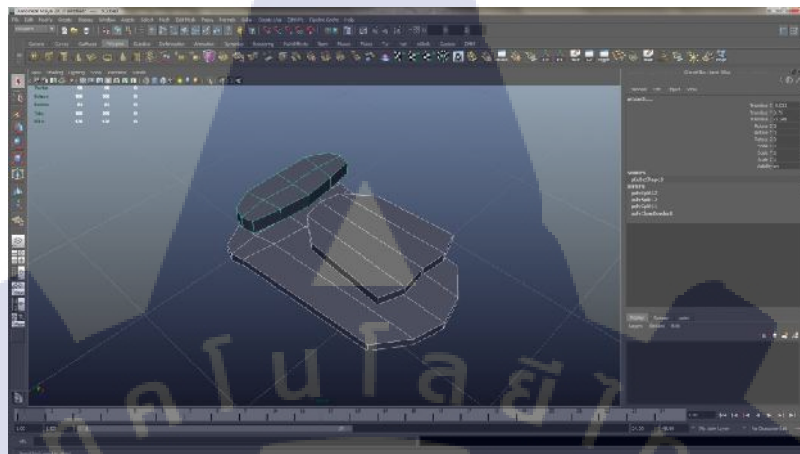


ภาพทำการเจาะรูกระป๋อง



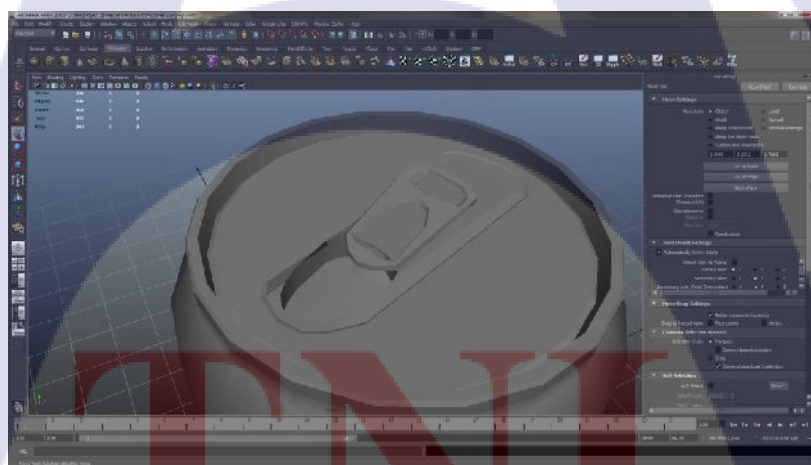
ภาพหลังจากเจาะรูกระป๋อง

ต่อไปให้สร้างฝากระป๋อง โดยการใช้โมเดลทรงกระบอกแล้วทำการตัด และเจาะรูเหมือนวิธีการที่ทำปากกระป๋อง



ภาพการทำฝากระป๋อง

เอาฝากระป๋องที่เจาะรูเสร็จแล้วมาวางบนปากกระป๋อง



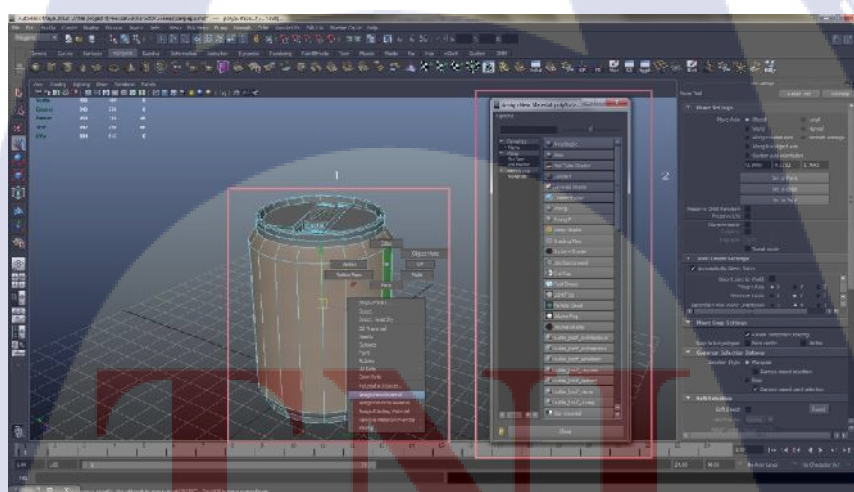
ภาพเอาฝากระป๋องมาจัดวาง

ต่อไปคือการทำสี หรือ ใส่ Texture ให้กับวัตถุ ซึ่งถ้าจะใช้ Texture ต้องหาภาพที่ต้องการมาก่อน



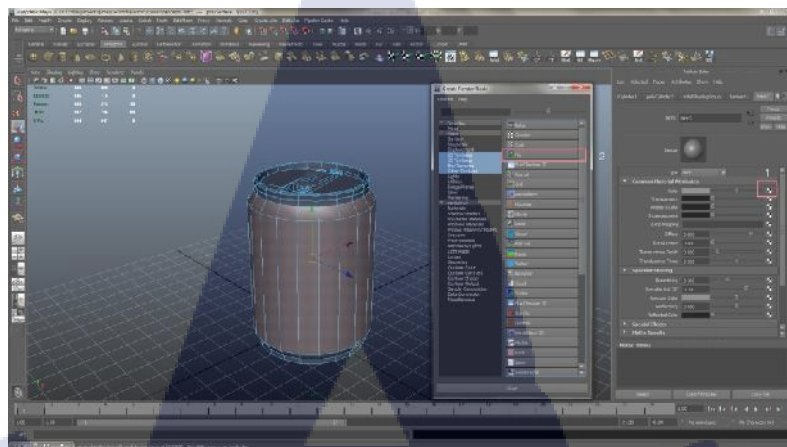
ภาพ Texture ที่ใช้

เลือกพื้นผิวที่วัตถุที่ต้องการใส่ Texture แล้วคลิกขวา เลือก Assign New Material เมื่อกดแล้วจะขึ้นช่องให้เลือกประเภทของพื้นผิว โดยหลักๆ แล้วจะใช้ Lambert กับ Blinn ในที่นี้ให้เลือก Blinn



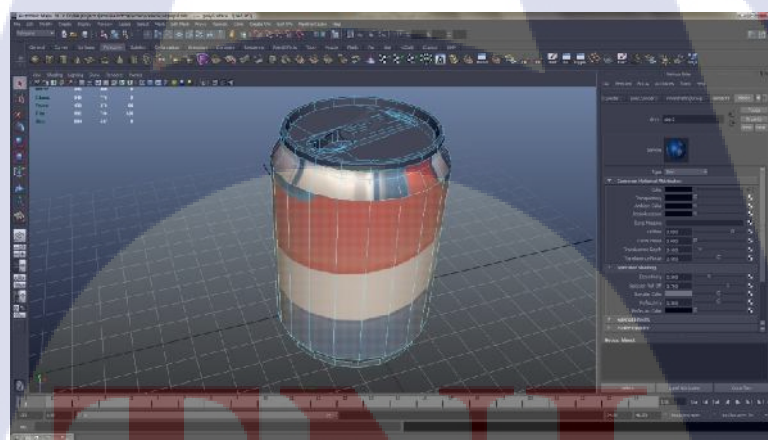
ภาพเลือกประเภท Texture ที่จะใช้

หลังจากที่เราเลือกประเภทของ Texture และ มีภาพที่จะใช้แล้ว ให้ไปยังส่วนของ Texture ที่เลือกเพื่อทำการนำเข้าภาพที่ต้องการใช้ โดยหลังคำว่า Color จะมีช่องสี่เหลี่ยมขาวดำสลับกัน ให้กดไปที่สี่เหลี่ยมนั้นแล้วจะขึ้นแถบแสดงด้านข้างให้เลือก File เมื่อกดไปแล้วจะขึ้นแสดงส่วนค้นหาภาพให้เราไปยังที่เก็บภาพเพื่อนำมาใช้ได้ทันที แต่ไฟล์เก็บนั้นห้ามเป็นภาษาไทยเด็ดขาด



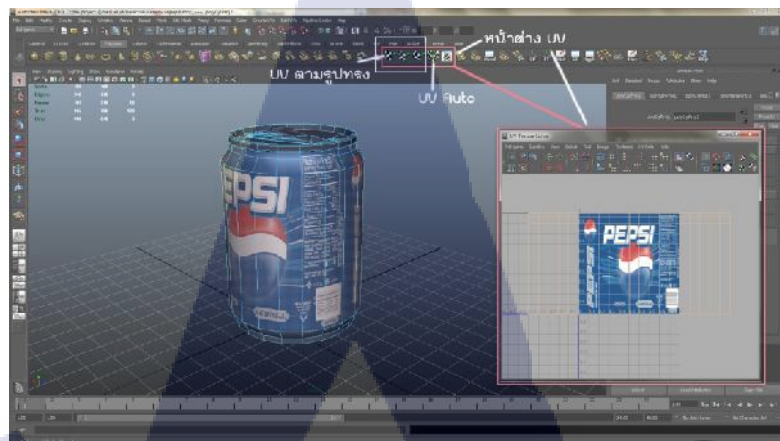
ภาพนำเข้า Texture

หลังจากเลือกไฟล์รูปมาแล้วตัวรูปที่อาจจะบิดเบี้ยวไปมาก แต่ก็สามารถแก้ไขได้ด้วยการกาง UV เพื่อบอกถึงขอบเขตการวาง Texture ซึ่งมีทั้ง แบบอัตโนมัติ แบบเรียบ แบบทรงกระบอก แบบทรงกลม หรือนำวัตถุไปตัด UV เองเพื่อความสวยงาม ก็ได้



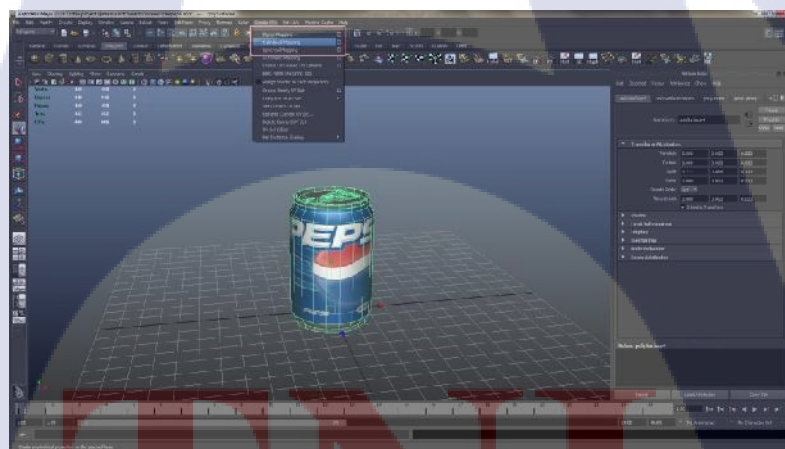
ภาพหลังจากการนำเข้า Texture

เมื่อทำการเลือก UV ที่ผมทำคือกางแบบทรงกระบอก โดยเราต้องเลือกพื้นผิวของวัตถุที่วาง Texture ไว้ก่อน



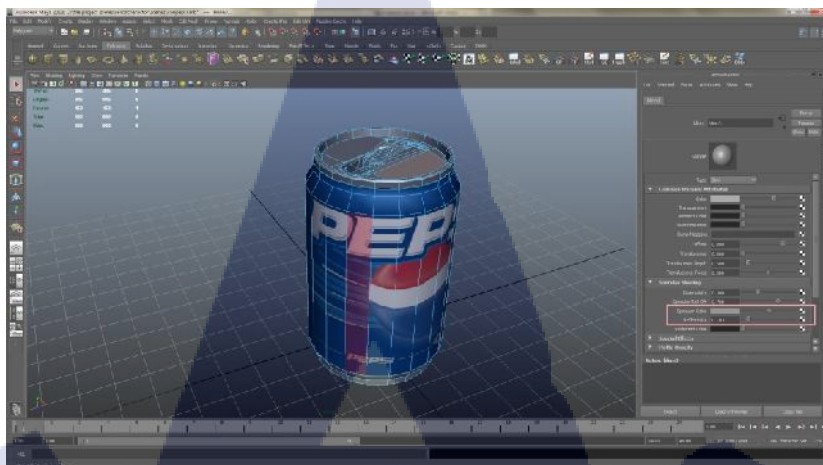
ภาพเลือก UV เพื่อวาง Texture

ถ้าเลือกการวาง UV แล้วไม่สวยสามารถปรับมุมได้ตาม Face ที่ต้องการ โดยการเลือกที่ Creates UVs ซึ่งจะเป็นส่วนจัดการ UV ถ้าเราเลือกการวางแบบทรงกระบอก เราก็เลือก Cylindrical Mapping เพื่อปรับการวาง Texture



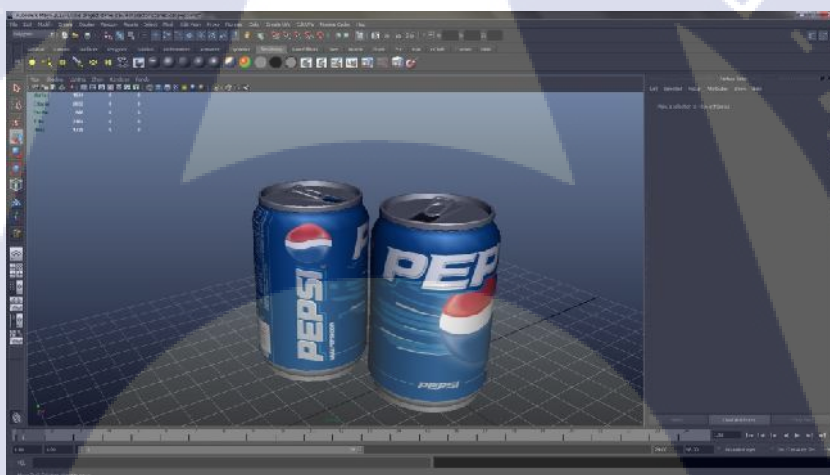
ภาพเลือก UV เพื่อปรับ Texture

จากนั้นทำสีให้กับส่วนบนและล่างของกระป๋อง โดยเราเลือกใช้ Blinn เนื่องจาก Blinn สามารถสะท้อนวัตถุอื่นๆ และยังสะท้อนแสงได้อีกด้วย ทำให้มีความเหมาะสมต่อการทำวัตถุพวก โลหะ โดยปรับในส่วนค่าของ Specular Color ซึ่งเป็นจุดรวมแสง กับ Reflectivity เป็นส่วนของการ สะท้อน



ภาพปรับค่าพื้นผิววัตถุ

เราก็จะได้โมเดล กระป๋อง PEPSI ตามที่ต้องการ



ภาพโมเดลกระป๋อง PEPSI

เมื่อเรามีพื้นฐานการปั้นโมเดลแล้วเราสามารถทำการปั้นโมเดลอื่นๆ เพื่อใช้ในงานโปรเจกต์ได้สบายขึ้น โดยโปรเจกต์ของข้าพเจ้าทั้งหมดก็มาจากพื้นฐานการปั้นโมเดลหลักๆ อย่างที่ยกตัวอย่างให้ได้ชม

ภาคผนวก ข

บทเกมการเล่น การจัดลำดับรอบของการเกม

TNI

บทเกมการเล่นและการจัดลำดับรอบของการเกม

เรื่องราวในเกม

เนื้อเรื่อง

การครั้งหนึ่งนานมาแล้ว โลกของเราเต็มไปด้วยมลพิษมากมายจากมวลมนุษยชาติ จนทำให้หลายสิ่งหลายอย่างเปลี่ยนไป ทั้งฤดูกาล สภาพอากาศ และทำให้เกิดภาวะโลกร้อน พอนานวันเข้ามลพิษก็ได้ก่อตัวกันเป็นกลุ่มก้อนแล้วทำให้มลพิษพวกนั้นเกิดมีมันสมองขึ้นมาและเริ่มคิดจะทำลายล้างโลกเพื่อให้มลพิษมีมากขึ้น และได้ทำลายสถานที่สำคัญของโลก ไม่ว่าจะเป็นโตเกียว ทาวน์เวอร์ หอเอนปิซ่า กรุงโซล จนกระทั่งวัดธรรมกาย แต่ยังมีสถานที่เหลือรอดอยู่หนึ่งคือ วัดโบราณสมัยกรุงรัตนโกสินทร์ จังหวัดอยุธยา ทำให้เหล่าเทพต่างพากันต้องช่วยมนุษยชาติ จึงส่งเทพลงมายังโลกเพื่อปกป้องสถานที่ ที่เหลือแห่งเดียว เพื่อให้คงอยู่ และต้องต่อสู้กับเหล่ามลพิษ

บทผู้เล่น

ผู้เล่นจะได้รับบทเป็นเป็นเทพที่ถูกส่งลงมาจากฟากฟ้า ซึ่งต้องกำจัดมลพิษที่เข้ามาโจมตีวัดที่เหลือเพียงหนึ่งเดียว โดยเริ่มแรกจะต้องเก็บกล่องเพื่อช่วยทำให้เราสามารถสู้กับเหล่ามลพิษได้ โดยห้ามเลือดหมด และ วัดห้ามถูกทำลาย

ไอเทมช่วยเหลือ

ช่วงเริ่มเกมจะมีกล่องมาให้เก็บ ซึ่งเป็นสิ่งที่ช่วยผู้เล่นอย่างมาก เพราะเกมนี้ถ้าไม่เก็บของเลยจะไม่มีทางชนะได้เลยแม้แต่น้อย โดยในช่วงแรกจะมีกล่องให้เก็บ ซึ่งกล่องจะมี 2 แบบ แต่ให้ผลที่ต่างกัน ซึ่งกล่อง 2 แบบนั้นเมื่อเก็บจะทำให้ยิ่งเร็ว ซึ่งความเร็วที่ได้จะต่างกัน และจะทับซ้อนค่าเพิ่มขึ้นไปเรื่อยๆ ซึ่งจะสุ่มสถานที่เกิดของไอเทม และยังมีเวลาการคงอยู่ของไอเทมที่เกิดออกมา ซึ่งทำให้ผู้เล่นต้องรีบเก็บโดยไว

AI ในเกม

AI ภายในเกมมีทั้งหมด 4 แบบ ซึ่งแต่ละแบบจะมีการ โจมตีที่แตกต่างกัน ซึ่งมีทั้งระยะประชิดกับระยะไกล ซึ่งแต่ละตัวนั้นจะมีความรุนแรงและพลังชีวิตแตกต่างกัน โดยที่มีจุดหมายชัดเจนคือทำลายวัตถุที่อยู่ตรงกลางฉาก และยังมี AI พิเศษที่เข้ามาโจมตีผู้เล่นเพื่อเป็นการขัดขวางภารกิจ เมื่อเรากำจัดได้ทั้งหมด ก็จะมีบอสออกมาให้กำจัดเพื่อเป็นการปิดฉากทั้งหมด

เมนูย่อย

ภายในเกมเมื่อกดปุ่ม P หรือ ESC จะแสดงหน้าต่างเมนูพร้อมหยุดเกมให้ซึ่ง จะมีตัวเลือกให้ผู้เล่นเช่น Resume (ย้อนเข้าสู่เกม) Restart (เริ่มเกมใหม่) Mainmenu (ไปยังหน้าเมนูหลัก) Volume (ตั้งค่าเสียง) Exit(ออกจากเกม)

TNI

ประวัติผู้จัดทำโครงการ

ชื่อ-สกุล นายภาณุวิชญ์ ศาสตร์เปล่ง

วันเดือนปีเกิด 27 มกราคม พ.ศ.2535

ประวัติการศึกษา

มัธยมศึกษาตอนต้น มัธยมศึกษาตอนต้น พ.ศ. 2549

โรงเรียนเมธีอมมาคุเลตคอนเวนต์

มัธยมศึกษาตอนปลาย มัธยมศึกษาตอนปลาย พ.ศ. 2552

โรงเรียนบางพลีราษฎร์บำรุง

อุดมศึกษา คณะเทคโนโลยีสารสนเทศ สาขาเทคโนโลยีมัลติมีเดีย

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ทุนการศึกษา -ไม่มี-

ประวัติการฝึกอบรม

1. Unity Workshop ณ สถาบันออกแบบพัฒนาซอฟต์แวร์โปรแกรมมิ่ง

2. Autodesk Factory Design Suite Synergysoft Workshop 2014 ณ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ผลงานที่ได้รับการตีพิมพ์ -ไม่มี-



TNI

TAI - NICHII INSTITUTE OF TECHNOLOGY