บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 บทนำ

ทฤษฎีที่เกี่ยวข้องในการนำมาใช้ออกแบบและพัฒนาเกม Simulation 3 มิติ แบ่งได้เป็น 5 ประเภท ได้แก่

2.1.1 ทฤษฎีการออกแบบ Model Game

เพื่อให้เห็นภาพรวมและการทำงานของเกม Simulation ที่ได้ออกแบบไว้ จนสามารถนำไปใช้เป็นแนวทางในการ Coding ตัวเกม

2.1.2 ทฤษฎี AI ที่นำมาใช้ภายในเกม

เพื่อนำความรู้ความเข้าใจที่ได้รับ ไปประยุกต์ใช้ในการออกแบบและสร้าง NPCs ที่มีการตัดสินใจในลักษณะที่คล้ายคลึงกับมนุษย์

2.1.3 ทฤษฎีการสร้างอารมณ์ของตัวละครในเกม

เพื่อให้สามารถสร้างตัวละครมีการความรู้สึกตอบสนองต่อเหตุการณ์ต่างๆ ตามที่ควรจะเป็น และทำให้ผู้เล่นมีความรู้สึกร่วมไปกับตัวละครในเกมมายิ่งขึ้น

2.1.4 ทฤษฎีการสร้าง Model 3 มิติ

เพื่อให้มีความเข้าใจเกี่ยวกับพื้นฐานการสร้างภาพสามมิติ รวมไปถึงวิธีใช้เครื่องมือชนิดต่างๆ จนสามารถสร้างฉาก, ตัวละครและองค์ประกอบต่างๆ ที่ปรากฏภายในเกม ในมุมมองที่เหมือนจริง

2.1.5 Game Engine

เพื่อให้การพัฒนาเกมทำได้ง่าย โดยสามารถนำเสนอองค์ประกอบต่างๆ ภายในเกมออกมาได้อย่างสมบูรณ์ตามที่ได้ออกแบบไว้ ในระยะเวลาอันรวดเร็ว

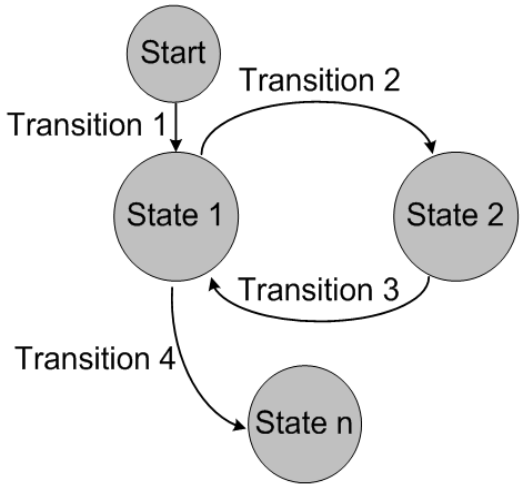
2.2 ทฤษฎีการออกแบบ Model Game

การออกแบบ Model Game โดยอาศัย State - Driven Design เป็นเครื่องมือในการออกแบบเกม โดยอาศัย FSMs (Finite state machines) ซึ่งเกิดจาก เซต 2 เซต ได้แก่

1. เซตของ States (สถานะ) ที่แสดงถึงเหตุการณ์ที่เป็นไปได้ที่จะเกิดขึ้นทั้งหมด

2. เซตของ Transitions (การเปลี่ยนแปลง) ซึ่งเป็นเงื่อนไขที่เชื่อมต่อระหว่าง States 2 States ที่ติดกัน

โดยสามารถพิจารณาการเปลี่ยนแปลงระหว่าง States ด้วย If-Then Conditionals เพื่อเช็คเงื่อนไขหรือกฎเกณฑ์ที่วางไว้



ภาพที่ 1 ตัวอย่าง FSMs (Finite State Machines)

ข้อดีของการใช้ State - Driven Design ในการออกแบบ Model Game มีดังต่อไปนี้

1. Coding ได้ง่ายและรวดเร็ว
2. สะดวกในการทำ Debugging
3. ไม่เปลืองทรัพยากรในการประมวลผล
4. อยู่บนพื้นฐานของความรู้สึกมากกว่าความเป็นจริง
5. มีความยืดหยุ่นสูง

2.3 ทฤษฎี AI ที่นำมาใช้ภายในเกม

2.3.1 Artificial Life

Artificial life (A - life) สร้างขึ้นโดย Chris Langton เป็นเทคนิคขั้นพื้นฐานในการสร้าง Robotic Applications และผู้พัฒนาเกมก็สามารถปรับใช้เทคนิคเหล่านี้ เพื่อให้เกมประสบความสำเร็จ โดยพื้นฐานแล้ว A - life System คือ ระบบที่ถูกสร้างขึ้นเพื่อที่จะแสดง

พฤติกรรมตามธรรมชาติ เช่น ระบบนิเวศของแมลงหรือพฤติกรรมของสัตว์ พฤติกรรมเหล่านี้จะถูกปลดปล่อยออกมาและพัฒนาเป็นผลลัพธ์

เกมที่พัฒนาขึ้นได้นำ A - life มาใช้ในการสร้างพฤติกรรมการตอบสนองของตัวละคร ซึ่งแตกต่างกันไปตามอาชีพ และบุคคลิกภาพ โดยอ้างอิงจากข้อมูลที่ได้จากการทำแบบสอบถาม เพื่อสำรวจการตอบสนองต่อเหตุการณ์ที่เกิดขึ้นในชีวิตประจำวันของบุคคลากรในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

2.3.2 Pathfinding

ประสิทธิภาพและประสิทธิผลของการค้นหาเส้นทางหรือ Pathfinding คือ คุณสมบัติพื้นฐานของ NPCs ที่ต้องมีในเกมทุกๆ ประเภท ตัวอย่างเช่น เกม War Simulation ที่ NPCs ต้องสามารถค้นหาข้าศึกผ่านภูมิประเทศต่างๆ รวมไปถึงการหลบหลีกอุปสรรค เช่น แม่น้ำและภูเขา เพื่อไปให้ถึงผู้เล่น หรือข้าศึก เป็นต้น ซึ่งสิ่งเหล่านี้ทำให้เนื้อเรื่องของเกมดำเนินต่อไปโดยไม่มีวันจบ จึงไม่ต้องสงสัยเลยว่าทำไมผู้พัฒนา AI ในเกมจึงให้ความสำคัญกับ Pathfinding อย่างมาก

เกมที่พัฒนาขึ้นได้นำ Pathfinding มาใช้ เพื่อให้ตัวละครที่กำลังดำเนินชีวิตอิสระและ NPCs ต่างๆ สามารถเดินไปยังสถานที่ที่ถูกต้องตามที่ได้กำหนดไว้ และไม่ชนกับสิ่งกีดขวาง เช่น ต้นไม้, อาคารต่างๆ รวมไปถึงตัวละครตัวอื่น โดยอาศัย Plug – in ของ DarkBADIC Professional ที่มีชื่อว่า Dark AI

2.3.3 Rule Systems

แม้ว่า FSMs จะเป็นเครื่องมือชั้นยอดในการออกแบบ AI ในเกม เพราะสามารถอธิบายพฤติกรรมต่างๆ ได้ง่าย แต่ก็มีบางเหตุการณ์ที่อธิบายด้วย State และ Transition ได้ยาก ดังตัวอย่าง Virtual Dog ด้านล่างนี้

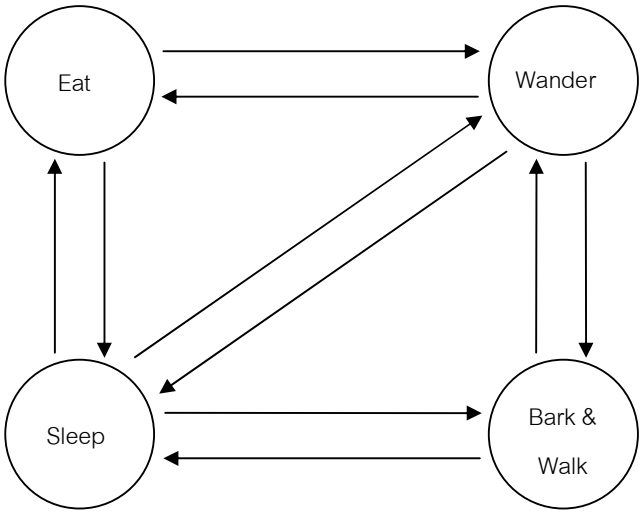
ถ้าฉันหิว และ มีกระดูกอยู่ใกล้ๆ , ฉันจะกินมัน

ถ้าฉันหิว แต่ ไม่มีกระดูกอยู่ใกล้ๆ , ฉันจะมองหามัน

ถ้าฉันไม่หิว แต่ ฉันง่วง , ฉันจะนอนหลับ

ถ้าฉันไม่หิว และ ฉันไม่ง่วง , ฉันจะเห่าและเดินไปมา

จากข้อความด้านบน สามารถออกแบบ FSMs ได้ดังนี้



ภาพที่ 2 FSM ที่ได้จากตัวอย่าง Virtual dog

จะเห็นได้ว่าแต่ละข้อความแทนด้วย State ซึ่งแต่ละ State สามารถ Transition ไปที่ State ใดก็ได้ Model ที่ได้จึงไม่ถูกต้องนัก ดังนั้นจึงสามารถอธิบายได้ว่า FSMs มีลักษณะสำคัญคือ พฤติกรรมที่จะนำมาออกแบบควรมีผลลัพธ์ของพฤติกรรมไม่มากนัก และควรเป็นการกระทำที่เป็นลำดับ โดยต้องเสร็จงานหนึ่งก่อนถึงจะสามารถทำอีกงานหนึ่งได้

ด้วยข้อจำกัดดังกล่าวจึงมีวิธีการออกแบบ AI อีกวิธีหนึ่งที่เรียกว่า Rule System (RS) ที่สามารถออกแบบพฤติกรรมซึ่งยากต่อการใช้ FSMs มีรูปแบบดังนี้

Condition → Action

ดังนั้นจึงสามารถออกแบบตัวอย่าง virtual dog ข้างต้นได้ดังนี้

(Hungry) & (Bone nearby) → Eat it

(Hungry) & (No bone nearby) → Wander

(Not hungry) & (Sleepy) → Sleep

(Not hungry) & (Not sleepy) → Bark and walk

การนำ Rule Systems มาแปลงเป็นโปรแกรมสำหรับกำหนดพฤติกรรมของตัวละครในเกม ทำได้โดยใช้หลักการของ Decision Tree เป็นตัวช่วยในการเรียงลำดับความสำคัญของกฎต่างๆ ที่ผู้พัฒนาเกมได้กำหนดขึ้น ด้วย “If” Tree ซึ่งสามารถสร้างได้ง่าย และมองเห็นเป็นระบบที่ชัดเจน อีกทั้งยังให้ Cost Ratio ต่ำ จึงสามารถ Run คำสั่งในโปรแกรมได้อย่างรวดเร็ว ซึ่ง If ตัวแรกถือเป็นกฎเกณฑ์แรกที่ต้องพิจารณา

เกมที่พัฒนาขึ้นได้นำ Rule Systems มาใช้ในการกำหนดลำดับความสำคัญของกิจกรรมที่ตัวละครแต่ละตัวควรจะทำในเกม เช่น กำหนดให้ NPCs ทุกตัวจะเลือกทำตามความต้องการทางด้านต่างๆ ต่อไปนี้ตามลำดับ

ขับถ่าย > ความหิว > สังคม > ความสนุก

ดังนั้นขณะที่ NPC แม่บ้านทำความสะอาดพื้น แล้วรู้สึกอยากเข้าห้องน้ำ พร้อมทั้งอยากพักเพื่อยืดเส้นยืดสาย NPC แม่บ้านจะเลือกที่จะไปเข้าห้องน้ำก่อน เป็นต้น

2.4 ทฤษฎีการสร้างอารมณ์ของตัวละครในเกม

การสร้างอารมณ์ของตัวละครได้นำทฤษฎี Affect Control Theory (Heise, 1977;1979;1986; Smith-Lovin and Heise, 1988) มาประยุกต์ โดยจำลองสถานการณ์ที่เกิดขึ้นในเกมแล้วใช้โปรแกรม Interact (คิดค้นโดย David Heise มหาวิทยาลัย Indiana) เพื่อทำนายอารมณ์ที่อาจเกิดขึ้นได้ของตัวละครตามพื้นฐานของบุคลิกและมุมมองต่อคู่สนทนา

Affect Control Theory กล่าวไว้ว่า อารมณ์เป็นสัญญาณที่บ่งบอกว่าเหตุการณ์ต่างๆ ที่เข้ามามีผลต่อสถานการณ์ปัจจุบันของมนุษย์อย่างไร หากการปฏิสัมพันธ์ยังคงรักษาสถานการณ์ปกติของตัวละครไว้ได้ ตัวละครจะไม่เกิดอารมณ์ใดๆ แต่หากการปฏิสัมพันธ์ขัดแย้งต่อลักษณะนิสัยตัวละครจะเกิดอารมณ์ ซึ่งส่งผลกระทบต่อตัวละครแต่ละตัวต่างกัน พฤติกรรมที่แสดงออกมาก็แตกต่างกันตามแต่นิสัยและมุมมองพื้นฐานเฉพาะบุคคล อาจกล่าวได้ว่าผลกระทบที่เกิดต่อมนุษย์มีลักษณะเฉกเช่นเดียวกับการที่สัญชาตญาณมีผลต่อสัตว์ ซึ่งเป็นการนำทางชีวิตไปในทิศทางที่คุ้นเคยและตรงตามค่านิยมของกลุ่มสังคม

องค์ประกอบพื้นฐานใน Affect Control Theory

1. สถานการณ์ ซึ่งเป็นการให้นิยามของสิ่งต่างๆ ดังต่อไปนี้
    1.1 ตัวละครที่กำลังพิจารณาคือใคร
    1.2 คนที่ปฏิสัมพันธ์ด้วยคือใคร
    1.3 การปฏิสัมพันธ์เกิดขึ้นที่ใด
2. ความรู้สึก ซึ่งประกอบขึ้นจาก 3 มุมมอง ดังนี้
    2.1 Evaluation (E) คือ การประเมินว่าสิ่งที่กำลังพิจารณานั้น ดี หรือไม่ดี
    2.2 Potency (P) คือ ความมีอิทธิพลของสิ่งที่กำลังพิจารณาว่ามีอิทธิพลต่อตัวละครมาก หรือน้อย
    2.3 Activity (A) คือ สิ่งที่กำลังพิจารณานั้นมีชีวิตชีวา หรือ จืดชืด

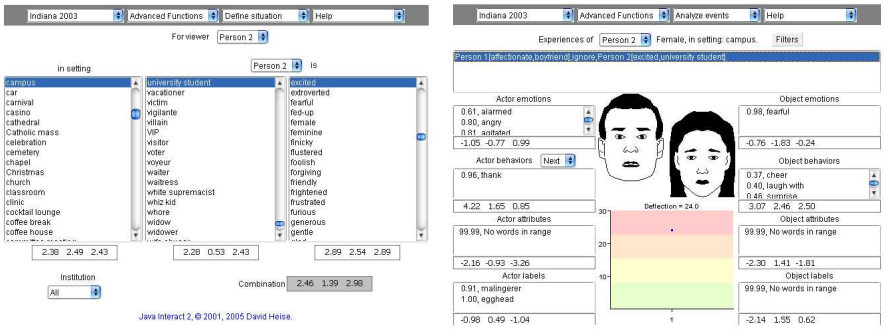
ซึ่งแต่ละมุมมองก็จะมีระดับต่างๆ กัน เช่น บางสิ่งอาจค่อนข้างดี, เกือบดี หรือดีมากที่สุด

3. สมการทางคณิตศาสตร์ซึ่งใช้ทำนายว่า เหตุการณ์ที่เข้ามาจะเปลี่ยนแปลงอามณ์ความรู้สึก เดิมของตัวละครอย่างไร สมการดังกล่าวได้พัฒนาโดยตรงจากการทำวิจัยเกี่ยวกับการตอบสนองของมนุษย์ ส่วนสมการอื่นๆ ได้มาจากการ Derive ทางคณิตศาสตร์

โปรแกรม Interact (David Heise 2002) มีความสามารถดังต่อไปนี้

1. ทำนายว่าเหตุการณ์ใดอาจเกิดขึ้นหากตัวละครมีบุคลิกตามที่กำหนดไว้
2. อารมณ์ใดอาจเกิดขึ้นในระหว่างการปฏิสัมพันธ์
3. ตัวละครจะแปลความหมายของการกระทำของผู้ที่ปฏิสัมพันธ์ด้วยว่าควรจะเป็นอย่างไรหลังจากเหตุการณ์นั้นเกิดขึ้น

นอกจากนี้ Interact ยังแยกมุมมองตามเพศอีกด้วยเพื่อนำเสนอทัศนคติที่ใกล้เคียงความจริงมากยิ่งขึ้น



ภาพที่ 3 โปรแกรม Interact

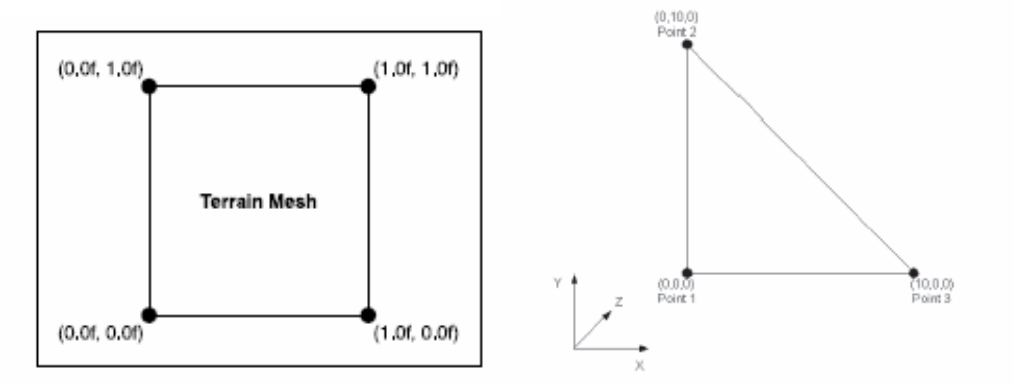
2.5 ทฤษฎีการสร้าง Model 3 มิติ

2.5.1 ลักษณะของ Graphic 3 มิติ โดยทั่วไปประกอบด้วย

1. โครงสร้างพื้นฐานของ Model 3D

Polygon คือ รูปหลายเลี่ยมโดยมีรูปทรงสามเหลี่ยมเป็นพื้นฐานที่ต่ำที่สุดประกอบด้วย 3 จุด โดยจุดหรือ Vertex คือ ส่วนย่อยที่เล็กที่สุดในระบบ 3 มิติ ซึ่งก็คือ จุดที่ตั้งอยู่ภายใน Space 2 มิติ หรือ 3 มิติ เราสามารถสร้างเส้นได้โดยการรวม Vertex 2 Vertex เข้าด้วยกัน เส้นสามเส้นจะรวมกันเป็น Polygon ความสัมพันธ์ของ

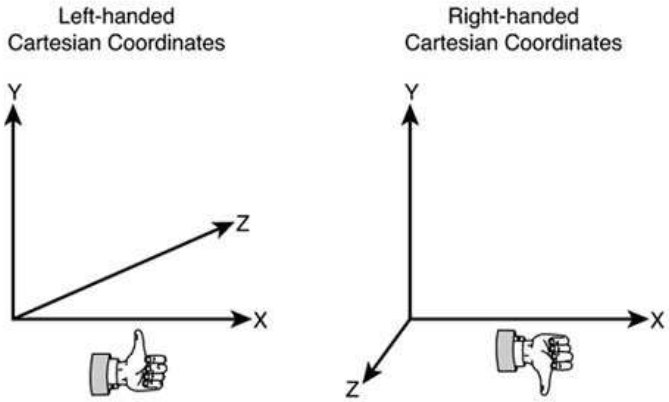
Vertex ที่สามารถวาดเส้นเชื่อมต่อกันเป็นรูปร่าง คือ พื้นผิวโพลีกอน (Polygon Face)



ภาพที่ 4 โพลีกอน2Dและโพลีกอน3D

2. ระบบพิกัด 3 มิติ (3D Coordinate System)

ใน DirectX จะใช้กฎหัวแม่มือซ้าย ซึ่งจะแตกต่างจาก OpenGL ซึ่งใช้กฎหัวแม่มือขวา



ภาพที่ 5 ระบบพิกัด 3 มิติ (3D Coordinate System)

3. Transformation คือ การที่ต้องการเปลี่ยนแปลง Vertex ของวัตถุ โดย Translation คือ ย้ายจุดกึ่งกลาง, Scale คือ ขยาย (หรือกระจาย Vertex จากจุดกึ่งกลาง), Rotate คือ หมุนเวอร์เท็กซ์รอบจุดกึ่งกลาง หากต้องการทำทุกอย่างให้คูณกันตามลำดับ Scale Rotate และ Translation

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

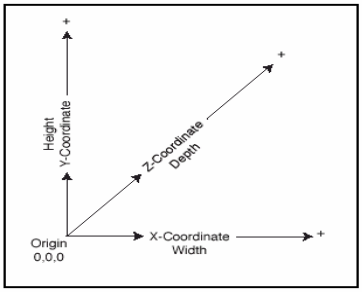
cos r	sin r	0	0
-sin r	cos r	0	0
0	0	1	0
0	0	0	1

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



ภาพที่ 6 รูปโครงสร้างเมทริกท์ (Matrix)

2.5.2 Model 3 มิติ

1. ส่วนประกอบของ Model 3 มิติ มีดังต่อไปนี้

1.1 Vertex คือ จุด (point) ซึ่งวางเรียงกันเป็นรูปทรงต่างๆ

1.2 Vertices คือ เซตของ Vertex ใช้แทนตำแหน่งของวัตถุ เช่น ระนาบ (Face) หรือ Mesh ใน 3D Space โดยที่แต่ละ Vertex อาจจะแทนด้วยค่า 3 ค่า (x, y, z) ซึ่งมีลักษณะเฉพาะของจุดหรือ coordinate ในรูปทรง 3 มิติ

1.3 Edge คือ เส้นที่เชื่อมต่อระหว่าง Vertex หนึ่งไปยังอีก Vertex หนึ่ง

1.4 Polygon คือ ระนาบที่เกิดจากการนำ Edge มาวางเรียงต่อกัน โดยที่ 1 Polygon จะต้องมีอย่างน้อย 3 Edge การทำ Model ที่มีลักษณะโค้งจะใช้หลายๆ Polygon มาเรียงต่อกันให้ดูโค้งแทนการทำเป็นเส้นโค้งจริง เพื่อประหยัดเวลาในการคำนวณ Model 3 มิติที่มีลักษณะโค้ง ซึ่งมีจำนวนของ

Polygon มาก แม้ว่าเส้นโค้งจริงจะทำให้ภาพมีความโค้งมนดูสมจริงมากขึ้นก็ตาม

1.5 Face คือ ส่วนประกอบที่อยู่ใน Mesh หรือ Poly ที่ถูกแบ่งครึ่งหรือ Vertices ที่มีตั้งแต่ 3 จุดขึ้นไปมาเชื่อมต่อกันเป็นรูปทรงต่างๆในแนวระนาบ Vertex เป็นตัวกำหนดมุมของ Face ทำให้ทุกๆ Vertex ใน Face จะต้องถูกกำหนดให้อยู่ในแนวระนาบ

1.6 Mesh เกิดจากการรวมกันของ Face ที่เชื่อมต่อกัน ซึ่ง 1 Mesh สามารถมี Face ได้ตั้งแต่ 1 Face ขึ้นไป การรวมของ Face นี้ทำให้ง่ายต่อการจัดการวัตถุในการทำ Animation, Material และ Texture ชนิดการรวมกันของ Face มีดังนี้คือ

1.6.1 Fan คือ กลุ่มของรูปทรงสามเหลี่ยม ซึ่งทุกรูปมีการใช้ Vertex ร่วมกัน 1 จุด โดยการกำหนดให้ Vertex นั้นๆ อยู่ระหว่างกลางของสามเหลี่ยมเหล่านั้นชนิดของการรวม Face แบบ Fan นี้คล้ายกับแบบ Strip คือ มีการกำหนดค่า Vertices 3 ค่าแรกสำหรับสามเหลี่ยมแรก จากนั้นในสามเหลี่ยมต่อไปก็เพียงแค่เพิ่มขึ้นมารูปละ 1 Vertex เท่านั้น

1.6.2 Strip คือ กลุ่มของสามเหลี่ยม ซึ่งแต่ละรูปจะมีการใช้เส้นร่วมกันกับสามเหลี่ยมส่วนหน้าซึ่งหมายความว่า หลังจากที่มีการกำหนดค่า Vertices 3 ค่าแรกสำหรับสามเหลี่ยมแรก จากนั้นในสามเหลี่ยมต่อไปก็เพียงเพิ่มขึ้นมารูปละ 1 Vertex

1.6.3 List คือ กลุ่มของสามเหลี่ยม ซึ่งทุกรูปไม่มีการใช้เส้นหรือ Vertex ร่วมกันเลย กล่าวคือ ค่า 3 ค่าของสามเหลี่ยมทุกรูปต้องกำหนดเองทั้งหมด

2. Texture แบ่งเป็น

2.1 Shading คือ การทำรายละเอียดของตัวพื้นผิว เช่น ความมันวาว การสะท้อนของพื้นผิว หรือความโปร่งแสง ทึบแสงของวัตถุ เป็นการให้สีเป็นลำดับชั้น

2.1.1 Flat Shading Lighting เป็นการลงรายละเอียดพื้นผิวที่มีสีเสมอกันทั่วทั้ง polygon

2.1.2 Vertex Shading หรือ Gouraud Shading เป็นการให้สีแก่ vertex แต่ละจุดตามสีที่ได้กำหนดเอาไว้แล้ว

2.1.3 Phong Shading มีลักษณะคล้ายกับ Gouraud Shading แต่จะให้แสงที่นวลกว่า แต่ก็ใช้เวลาในการ render นานกว่าด้วย

2.2 Texture คือ ลวดลายของพื้นผิว โดยที่จะเป็น Bitmap ที่เป็น Pattern หรือ Image มักจะเก็บในรูปแบบไฟล์ BMP, PCX หรือ GIF เพื่อเป็นการใส่รายละเอียดให้แก่พื้นผิวของวัตถุ ทำให้วัตถุมีความสมจริง โดยการโมเดลตัวละครในเกมสมัยนี้ นิยมใช้ Low Polygon Models ซึ่งมีพื้นฐานมาจากการใช้ Texture ในการช่วยทำให้ผู้เล่นมีความรู้สึกเหมือนว่าภาพที่เห็นมาความตื้นลึกหนาบางจริง

2.2.1 Texture Coordinate ใช้กำหนดการเชื่อมต่อกันระหว่าง Vertices ของ Face กับ Pixel ของ Bitmap โดย Texture Coordinate นี้ใช้แทน 2 มิติของ Coordinate System

2.2.2 Texture Mapping คือ การวาดรูปลงบนพื้นผิวของ Face หรือ Polygon และในการทำ Texture Mapping นี้ต้องคำนึงถึงการคำนวณค่าต่างๆด้วย จึงต้องมีการกำหนดค่าของ Vertices ด้วย จากการที่ Texture Coordinate กำหนด Pixel ของ Texture ที่จะวาดลงในส่วนของ Face แล้วก็จะมีการ Wrapping เพื่อ Generate Texture Coordinate สำหรับ Object นั้นซึ่งการ Wrapping นั้นมี 4 ชนิด คือ

- Flat Warp จะทำการวาดลงบน Face โดยตรง
- Cubical Warp จะทำการ Warp Texture ใน Cube รอบๆ Object เหมือนกับการแปะลวดลายบนวัตถุที่มีลักษณะเป็นกล่อง
- Cylindrical Warp จะทำการ Warp Texture ในทรงกระบอกรอบๆ Object คือจะนำลวดลายในการ Map รอบโมเดลในลักษณะการห่อเหมือนทรงกระบอก ซึ่งเหมาะกับวัตถุที่มีลักษะเป็นทรงกระบอกหรือเป็นแท่งๆ
- Spherical Warp จะทำการ Warp Texture ใน Sphere รอบๆ Object คือจะนำภาพมาแปะในลักษณะห่อรอบทรงกลม เหมาะกับการแปะลวดลายบนวัตถุทรงกลม
- Shrink Wrap คล้ายกับการ Map แบบ Spherical แต่โปรแกรมจะรวมจุดปลายของลวดลายที่ นำมา Map เข้าหากันเป็นจุดเดียว

3. Animation

เป็นที่ทราบกันทั่วไปว่า หลักการทำ Animation ขั้นพื้นฐานนั้นก็คือ การนำเอาภาพนิ่งที่ต่อเนื่องกันจำนวนมากๆ มาเปิดทีละภาพต่อกันด้วยความเร็วสูง ภาพชุดนั้นๆ ก็จะดูเป็นภาพที่มีเคลื่อนไหว หรือที่เรียกว่า Key Frame Animation และภาพแต่ละภาพที่ถูกเปิดขึ้นมานั้นเรียกว่า Frame โดยหน่วยที่ใช้วัดคุณภาพของ Animation ก็จะใช้การนับจำนวนของภาพ หรือ Frame ที่จะถูกเปิดขึ้นมาในช่วงเวลา 1 วินาที เช่น Animation แบบ 8 Frame ต่อวินาทีนั้นหมายถึง ในหนึ่งช่วงเวลา 1 วินาทีจะต้องใช้ภาพนิ่งจำนวน 8 ภาพ ซึ่งการทำ Animation ในลักษณะนี้จะเห็นว่ายุ่งยากมาก เพราะถ้าเป็น Animation เรื่องยาวๆ ก็เป็นอันต้องเขียนภาพจำนวนมากๆ อย่างหลีกหนีไปไม่ได้ แต่วิธีการทำAnimationแบบนี้ก็ถือว่าเป็นวิธีพื้นฐานที่สุดแม้ทุกวันนี้การทำ Animation หลายๆ เรื่องก็ยังคงใช้การทำงานแบบนี้อยู่

สำหรับการทำงาน Animation ในงาน 3D นั้นก็ยังอาศัยหลักการเดียวกันกับการทำ Animation ที่ได้กล่าวมาแล้ว แต่ก็ได้มีการลดขั้นตอนการทำงานให้ย่นย่อสะดวกสบายยิ่งขึ้น เรียกว่าการทำ Animation แบบ In - Between Frame ซึ่งการทำ Animation ในลักษณะนี้เราจะกำหนดให้มี Frame บาง Frame ทำหน้าที่เป็นตัวบันทึกจังหวะในการเคลื่อนที่ในแต่ละช่วงเวลาของวัตถุใน Scene ซึ่งเราจะเรียก Frame ที่ทำหน้าที่บันทึกการเคลื่อนที่ของวัตถุนั้นว่า Key Frame สำหรับการทำ Animation ในแบบของการกำหนด Key Frame นั้นจะทำโดยการนำเอาตำแหน่งของในแต่ละ Key Frame มาคำนวณหาความเป็นไปได้ของตำแหน่งใน Frame ที่อยู่ระหว่าง Key Frame แบบอัตโนมัติ ซึ่งการทำงานในลักษณะนี้จะประหยัดเวลากว่ามากสำหรับการทำ Animation แบบเดิมๆ โดยเฉพาะกับ Animation ที่มีความยาวมากๆ แต่กลับใช้คนน้อยๆ ในการทำงานซึ่งโปรแกรมคอมพิวเตอร์ที่ใช้ทำงานด้าน Animation เกือบจะทั้งหมดต่างก็ใช้หลักการทำ Animation แบบ Key Frame เป็นหลักทั้งสิ้น

2.5.3 การแสดงผล Model 3 มิติ ประกอบด้วย

1. Camera คือ มุมกล้องที่ทำให้เราสามารถมองเห็น Model จากทิศทางต่างๆ เปรียบเสมือนการนำกล้องไปตั้งไว้บริเวณพื้นที่ที่มี Model ดังกล่าว ดังนั้นการเปลี่ยนมุมกล้องก็จะทำให้ภาพที่ปรากฏบนจอมีลักษณะแตกต่างกันไป การมี Model เพียงชิ้นเดียวอาจจะไม่มีปัญหาอะไรในการมอง Model นั้น แต่ถ้ามี Model หลายๆ ชิ้นหลายๆ รูปทรงในบริเวณเดียวกัน การลำดับตำแหน่งของ Model จึงเป็นสิ่งจำเป็นที่จะต้องฝึกให้คุ้นเคย จนสามารถมองภาพให้เป็นมิติที่สมจริงได้ ไม่ว่าภาพ

นั้นจะแสดงผลในรูปแบบใด ดังนั้นลำดับเป็นเรื่องที่ต้องแยกแยะให้ออก หากว่าไม่สามารถที่จะมองภาพที่เราสร้างขึ้นให้เป็นมิติได้ ก็ยากที่จะรู้ได้ว่า Model ชิ้นไหนอยู่ข้างหน้า หรืออยู่ข้างหลัง และอาจทำให้เกิดความสับสนในเรื่องของการลำดับ Model และทำให้การสร้าง Model มีความยุ่งยากซับซ้อนมากยิ่งขึ้น จนไม่สามารถสร้างเกมที่มีความสมบูรณ์และมีประสิทธิภาพได้

2. Viewport

อยู่ในส่วนของ Camera โดย Viewport นั้นจะรวมไปถึงตำแหน่งและทิศทางของ Scene จากการมองเห็นซึ่งมักใช้ Viewport ร่วมกับ Field of View Front & Back Clipping และ Perspective Transformation

2.1 ส่วนประกอบของ Viewport ได้แก่

2.1.1 Eyepoint คือ จุดที่ตั้งกล้อง

2.1.2 Lookat คือ จุดที่กล้องมอง

2.1.3 Upvector คือ Vector ด้านบนของกล้อง

2.2 การ View ดู Object มี 2 ชนิด

2.2.1 Perspective เป็นวิธีที่ทำให้รูปที่วาดลงบน Screen มีลักษณะที่ใกล้เคียงกับ Object จริงๆมากที่สุดเพราะจะสนใจทั้งจุด x, y, z ทำให้เกิดความลึกของภาพ

2.2.2 Orthographic เป็นวิธีที่ทำให้รูปที่วาดลงบน Screen แบบง่ายๆจึงละเลยในส่วนของจุด z ไปทำให้ภาพที่ได้คล้ายกับ 2 มิติ

3. Light

แสงทำให้เกิดการเปลี่ยนแปลงของสีของ Vertices โดย Module ทำให้เกิด Vertex Normal เพราะสิ่งนี้ขึ้นอยู่กับมุมของแหล่งกำเนิดแสงตามปกติจะมีแสงสีขาวเพราะเป็น การรวมกันอย่างหนาแน่นของสีทุกสี และโดยมากมักใช้รูปแบบของ RGB ในการกำหนดสีของแหล่งกำเนิดแสง

3.1 การตั้งค่า RGB ของแม่สีต่างๆ ใน 3D เป็นดังนี้

3.1.1 แสงสีขาว เป็น 1, 1, 1

3.1.2 แสงสีแดง เป็น 1, 0, 0

3.1.3 แสงสีน้ำเงิน เป็น 0, 0, 1

และยังสามารถใช้แสง 3 สีนี้ มาผสมเป็นแสงสีใหม่ได้

3.2 ลักษณะของแหล่งกำเนิดแสงมี 4 ชนิด ได้แก่

3.2.1 Ambient Light คือแหล่งกำเนิดแสงที่ง่ายที่สุด เพราะไม่ต้องมีการกำหนดตำแหน่งของแหล่งกำเนิดแสง และยังให้ความสว่างทั่วทุก Object

3.2.2 Point Light เป็นแหล่งกำเนิดแสงที่ทำการกระจายแสงไปทุกทิศทาง แต่ต้องระบุตำแหน่งของแหล่งกำเนิดแสง โดยไม่ต้องกำหนดทิศทางของแสง

3.2.3 Directional Light เป็นแหล่งกำเนิดแสงที่มีประสิทธิภาพมากที่สุดเพราะ เป็นแหล่งกำเนิดแสงที่มีทิศทาง โดยต้องระบุตำแหน่งของแหล่งกำเนิดแสง

3.2.4 Spot Light เป็นแหล่งกำเนิดแสงที่ต้องมีการระบุทั้งทิศทางและตำแหน่งของแหล่งกำเนิดแสง โดยการผลิตแสงจะเป็นรูปร่าง

4. Rendering

การ Render คือการสั่งให้โปรแกรมประมวลผลทุกสิ่งทุกอย่างไม่ว่าจะเป็น Material หรือแสงเงาของภาพ 3 มิติให้ออกมาเป็นภาพ 2 มิติสำหรับการนำไปใช้งานหรือเป็นภาพอ้างอิงก่อนที่จะนำ Models ไปลงในโปรแกรมที่เขียนขึ้น

2.5.4 ตัวอย่างเครื่องมือหรือโปรแกรมที่ใช้สร้าง Modle 3 มิติ

เครื่องมือหรือโปรแกรมที่ใช้สร้างภาพ 3 มิติที่พบได้ทั่วไปมีหลายชนิด ซึ่งแต่ละชนิดก็มีจุดเด่นและการนำมาใช้ประโชยน์ที่แตกต่างกันไป สรุปได้ดังนี้

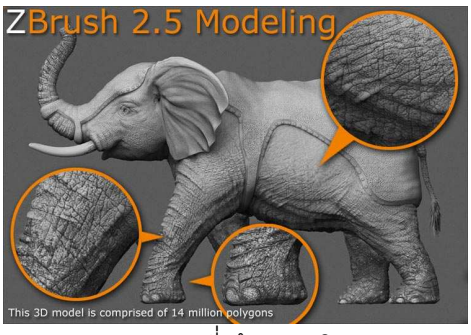
1. Poser

โปรแกรมนี้เป็นโปรแกรมที่มีจุดเด่นในการสร้าง Model ของมนุษย์ที่ง่ายและรวดเร็ว แต่เนื่องจากความสามารถในการสร้างสภาพทางกายภาพของมนุษย์ที่สมจริง ส่งผลให้ Model ที่ได้มีรายละเอียดสูง ดังนั้น File ที่ได้จากโปรแกรมนี้จึงมีขนาดใหญ่ และไม่เหมาะอย่างยิ่งที่จะนำมาสร้างตัวละครภายในเกม แต่นิยมในมาใช้ในการสร้าง Model มนุษย์ที่เป็น 3 มิติ

ภาพที่ 7 ตัวอย่าง Model ที่สร้างจากโปรแกรม Poser4

2. Pixologic ZBrush

โปรแกรมนี้เป็นโปรแกรมที่มีจุดเด่นในการสร้าง Model ที่ต้องการความละเอียดสูง ซึ่งสามารถให้ความละเอียดสูงสุดได้ถึงหลายล้าน Polygon ทำให้ File ที่ได้มีขนาดใหญ่ และไม่เหมาะที่จะนำมาสร้างตัวละครภายในเกม แต่เหมาะสำหรับการสร้าง Model 3 มิติ ที่ต้องการความละเอียดสูง และมี Face เป็นจำนวนมาก



ภาพที่ 8 ตัวอย่าง Model ที่สร้างจากโปรแกรม ZBrush 2.5

3. LightWave 3D

LightWave 3D เป็น Software ที่ Studio ส่วนใหญ่ นิยมนำมาใช้ในการสร้าง Computer Graphic เนื่องจาก LightWave 3D เป็นโปรแกรมที่นักสร้างภาพสามมิติ (3D Animator) ทั่วโลกต่างยอมรับว่าเป็นโปรแกรมที่สามารถสร้าง Model ได้ง่ายและรวดเร็ว อีกทั้งยังสามารถ Render ภาพได้สมจริงที่สุด ซึ่งไม่แปลกที่ LightWave 3D ได้รับความนิยมสูงในงานโทรทัศน์ และภาพยนตร์หลายๆ เรื่อง เช่น The Passion of the Christ, The Last Samurai, Immortel, The Perfect Score, The Italian Job, Holes, Master and Commander, Gothika และ The League เป็นต้น ซึ่งในประเทศไทยเองก็มีผลงานมากมายที่นำเอา LightWave มาใช้ ตัวอย่างเช่น Logo ของ China Doll รวมทั้งการ์ตูน 3 มิติ อย่าง แก้วจอมแก่น เป็นต้น

ภาพที่ 9 ตัวอย่าง Model ที่สร้างจากโปรแกรม LightWave 3D ซึ่งนำมาใช้ในวงการโฆษณา

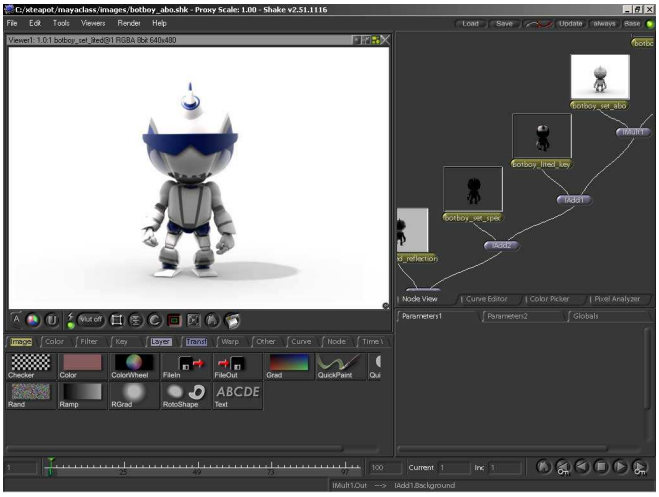
4. 3D Studio MAX

โปรแกรม 3D Studio MAX นี้ เป็นโปรแกรมสร้าง Computer Garphic ที่ได้รับความนิยมเป็นอย่างมาก โดยทั่วไปมักจะนำโปรแกรมนี้มาใช้สร้างในการงานเพื่อการนำเสนอทางด้าน Architecture, Product Design, Furniture Design, Interior Design, Exhibition Design, Graphic Design ฯลฯ ตั้งแต่ขั้น Conceptual Design ไปจนถึงการทำ Final Presentation ทั้งในรูปแบบ 2 มิติ และ 3 มิติ, การทำภาพเคลื่อนไหวต่างๆ ไม่ว่าจะเป็นการทำภาพแบบ Panorama หรือ Animation ที่เน้นการตอบสนองงานทางด้านออกแบบโดยเฉพาะ

5. Aliaswavefront MAYA

โปรแกรม Maya เป็นโปรแกรมที่ได้รับความนิยมมากที่สุดในธุรกิจภาพยนตร์ทั้งในประเทศและต่างประเทศ มีจุดเด่นในเรื่องของความยืดหยุ่นในการทำงาน โดยใช้การทำงานแบบ Node - base กล่าวคือ Function การทำงานแต่ละแบบจะถูกเก็บเป็น Node แต่ละ Node จะมี Input และ Output ซึ่งสามารถเชื่อม Node เหล่านี้เข้าด้วยกัน ทำให้เกิดผลลัพธ์ที่แตกต่างกันเป็นจำนวนมาก รวมถึงการใช้งาน Script ที่เรียกว่า MEL ซึ่งใช้งานได้ง่ายและมีประสิทธิภาพ

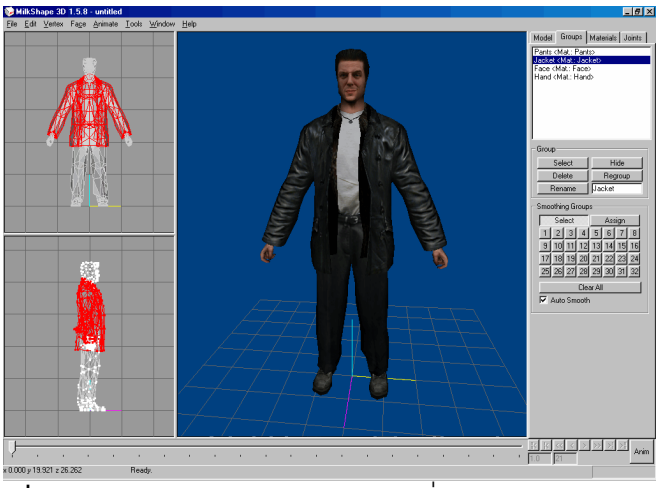
นอกจาก Maya จะสามารถสร้างงานเคลื่อนไหวได้สมจริงแล้ว Maya ยังสามารถนำไปใช้ในงานด้านออกแบบตกแต่งภายใน และงานออกแบบอื่นๆได้อีกด้วย



ภาพที่ 10 การใช้ Maya ในงานแบบมืออาชีพ

6. MilkShape 3D

MilkShape 3D เป็นโปรแกรมจัดการเกี่ยวกับ Model 3 มิติที่ผู้ที่สร้าง Model ตัวละครต่างๆ ภายในเกมรู้จักดี ซึ่งโปรแกรมนี้จะทำงานในรูปแบบของ Low Polygons อีกทั้งตัวโปรแกรมยังรองรับ 3D Format ต่างๆ โดยเฉพาะอย่างยิ่งการสร้าง DirectX Format (.X) ที่สมบูรณ์แบบ ทั้ง Animation และ Texture ของตัวละคร นอกจากนี้ MilkShape 3D ยังมี Build in Tool อยู่เป็นจำนวนมากที่ช่วยในการสร้างเกม เช่น Tool ที่ช่วยลดจำนวน Polygons, Tool ที่ช่วยสร้าง Terrains นั่นหมายความว่า MilkShape3D สามารถใช้สร้างฉากในเกมได้อีกด้วย



ภาพที่ 11 Model จากเกม Max Payne 2 ที่สร้างจาก MilkShape3D

เกมที่พัฒนาขึ้นได้นำ MilkShape3D มาใช้ในการสร้าง Model 3 มิติที่นำมาใช้ในเกม เนื่องจากโปรแกรมมีการพัฒนาอย่างต่อเนื่อง อีกทั้งมีความสามารถในการสร้างและนำ Model ออกไปใช้กับงานได้หลายประเภท โดยอาศัยสั่งพื้นฐานเช่นเดียวกับโปรแกรมจัดการ Model 3 มิติทั่วไป ไม่ว่าจะเป็น Select, Move, Rotate, Scale, Extrude, Turn Edge หรือ Subdivide อีกทั้งยังอนุญาตให้แก้ไข Vertex และ Face ของ Model ได้ นอกจากนี้ก็ยังมีคำสั่งในการสร้าง Primitive Model เช่น Spheres, Boxes และ Cylinders สำหรับนำไปประยุกต์ใช้สร้างส่วน Model ต่างๆ พร้อมทั้งรองรับการทำ Skeletal Animation หรือการสร้างการเคลื่อนไหวให้กับ Model โดยการใส่กระดูกให้กับ Model นั้นอีกด้วย

2.5.5 การใช้ MilkShape 3D ในการสร้าง Model 3 มิติ

MilkShape3D เป็นโปรแกรมจัดการ 3 มิติ ที่สร้างโดย Chumbalum Soft ซึ่งในอดีต MilkShape3D เป็นโปรแกรมที่ใช้สำหรับทำ Low Polygons Model สำหรับเกม Half-Life เท่านั้น ต่อมาโปรแกรมได้การพัฒนาโดยเพิ่มรูปแบบ File Format เพื่อที่จะสามารถรองรับการใช้งานในเกมอื่นๆ ได้มากขึ้น

ตารางที่ 1 แสดง File Formats ที่ MilkShape3D รองรับ

Format	Import	Export	3rd party
Half-Life SMD	Yes	Yes	
Half-Life MDL	Decompile Only	Yes (Compile)	
Quake MDL		Yes (Compile)	
Quake II MD2	Yes	Yes	
Quake III: Arena MD3	Yes	Yes	
Unreal/UT Engine 3D	Yes	Yes	
Unreal/UT Engine Skeletal Mesh PSK	Yes	Yes	
Vampire: the Masquerade NOD	Yes	Yes	
Genesis3D 1.0 BDY	Yes	Yes	
Genesis3D 1.0 MOT	Yes	Yes	
Genesis3D 1.0 ACT	Decompile Only		
Serious Sam MDL	Yes		
Serious Sam LWO/SCR		Yes	
Max Payne KF2	Yes	Yes	
Max Payne KFS	Yes	Yes	
Max Payne SKD	Yes	Yes	
The Sims SKN	Yes	Yes	
Wavefront OBJ	Yes	Yes	
3D Studio ASC	Yes	Yes	
LightWave LOW	Yes	Yes	

ตารางที่ 1 แสดง File Formats ที่ MilkShape3D รองรับ (ต่อ)

Format	Import	Export	3rd party
LightWave 6.5x LOW		Yes	
AutoCAD DXF	Yes	Yes	
POV-Ray INC		Yes	
VRML1 WRL		Yes	
VRML97 WRL		Yes	Yes
Autodesk 3DS	Yes	Yes	
BioVision Motion Capture BVH	Yes		Yes
Nebula Script		Yes	Yes
Jedi Knight 3DO		Yes	Yes
GenEdit 3DT		Yes	Yes
DirectX 8.0		Yes	Yes
DirectX 8.0 (JT)		Yes	Yes

วิธีการใช้งานของ MilkShape3D แบ่งออกเป็น 3 ลักษณะ ได้แก่

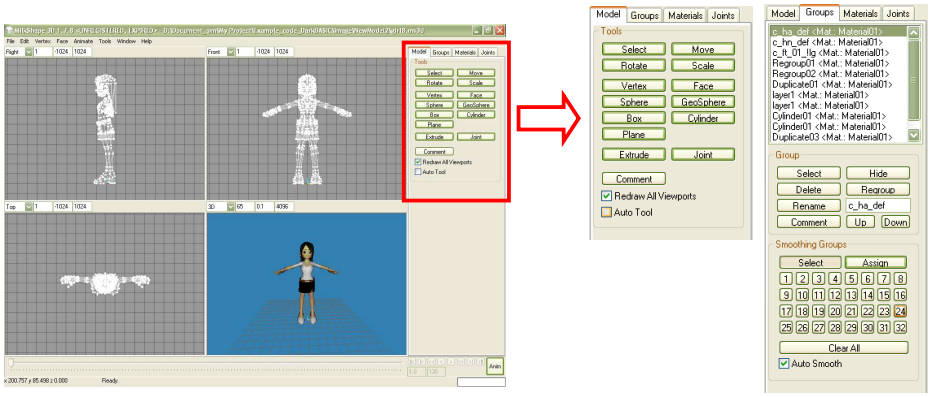
1. การขึ้นรูป Model จะอาศัย Tool ต่างๆ ทางขวามือดังนี้

   1.1 Model เป็นกลุ่มเครื่องมือที่ใช้ในการแก้ไข Vertex และ Face ของ Model และสร้าง Primitive Model

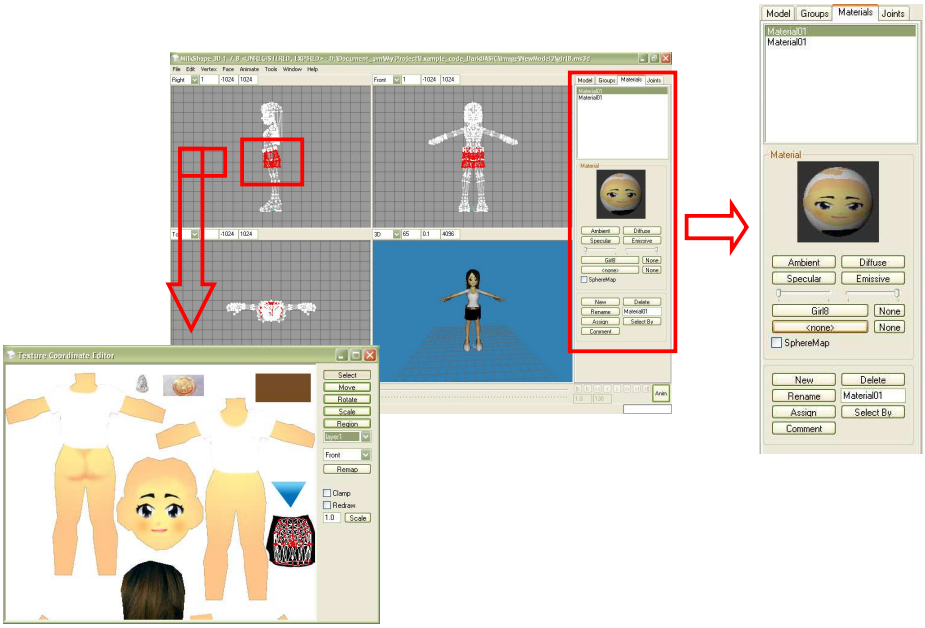
   1.2 Group เป็นกลุ่มเคื่องมือที่ทำการรวม Vertex และ Face เข้าด้วยกันเป็น โดยสามารถตั้งชื่อให้กับกลุ่มของ Vertex และ Face นั้น รวมไปถึงการแสดง หรือซ่อนกลุ่มของ Vertex และ Face เพื่อให้ง่ายในการแก้ไขส่วนอื่นๆ ของ Model

2. การใส่ Texture .ให้กับ Model จะอาศัย Tool Material ทางด้านขวามือ โดยจะทำการเพิ่ม Texture ที่เป็น File ชนิดรูปภาพ .jpg, .png ฯลฯ ในกับ Model ซึ่งสามารถกำหนด Texture แยกตามกลุ่มของ Vertex และ Face ได้

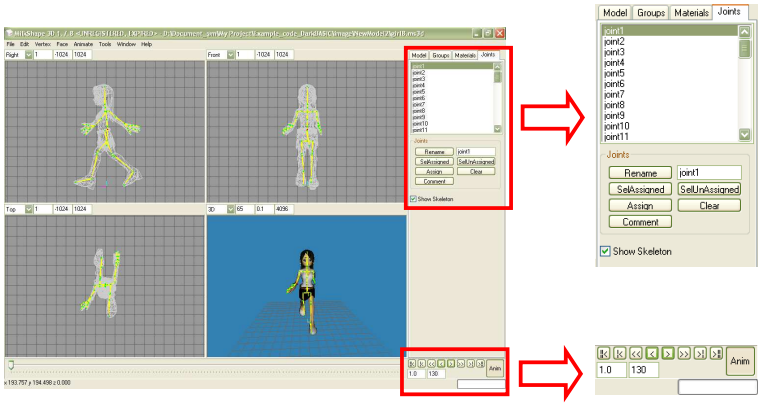
3. การใส่ Animation ให้กับ Model จะอาศัย Tool Joint ทางขวามือ โดยใช้ Joint ใช้แทนข้อต่อต่างๆ โครงกระดูกของ Model แล้วจึงกำหนดกลุ่มของ Vertex และ Face ให้กับ Joint ต่างๆ เพื่อให้เคลื่อนไหวไปตามท่าทางที่กำหนดให้ ต่อมาจึงทำการใส่ทางท่าทางให้กับ Model โดยใช้ Tool Anime ทางด้านขวาล่าง แล้วกำหนดท่าทางทีละ Frame ด้วยคำสั่ง Setkeyframe



ภาพที่ 12 แสดง Tool ในการขึ้นรูป Model ของ MilkShape3D



ภาพที่ 13 แสดง Tool ในการใส่ Texture .ให้กับ Model



ภาพที่ 14 แสดง Tool ในการใส่ Animation ให้กับ Model

2.6 Game Engine

ปัจจุบัน Game Engine เป็นเครื่องมือที่นักพัฒนาเกมนิยมเลือกใช้ในการสร้างสรรค์เกมใหม่ๆ ออกสู่ตลาด เนื่องจากการสร้างเกมโดยอาศัยคุณสมบัติของ DirectX9 บน C++ หรือ C# โดยตรงนั้น มีวิธีการที่ยุ่งยากและซับซ้อน ผู้ใช้จึงจำเป็นต้องใช้เวลาในการศึกษาระยะหนึ่ง เพื่อให้มีความรู้ความเข้าใจในการใช้งาน DirectX ได้อย่างเต็มประสิทธิภาพ

DarkBASIC Professional เป็น Game Engine ชนิดหนึ่งที่ได้รับความนิยมอย่างแพร่หลายในกลุ่มของนักพัฒนาเกม เนื่องจากสามารถทำให้การสร้างองค์ประกอบต่างๆ ภายในเกมที่มีความสมจริง ซึ่งเดิมเป็นเพียงความเพ้อฝันของนักพัฒนาเกมให้กลับกลายเป็นความจริงได้ด้วยการใช้คำสั่งเพียงคำสั่งเดียว

2.6.1 ความเป็นมาของ DarkBASIC Professional

DarkBASIC ถูกออกแบบมาครั้งแรกในปีพ.ศ. 2542 โดยมีวัตถุประสงค์เพื่อทำให้การพัฒนาเกมคอมพิวเตอร์ 3 มิติ บนเครื่อง PC ทำได้ง่ายขึ้น โดยอาศัยพื้นนฐานของภาษา BASIC

DarkBASIC Professional เป็น version ที่ได้รับการพัฒนาต่อมา สำหรับใช้ในการพัฒนาเกมขั้นสูง ร่วมกับเทคโนโลยี Microsofts DirectX 9 ซึ่งใช้ได้กับการพัฒนาเกมบนเครื่อง PC ที่ใช้ Microsoft Windows เท่านั้น

ตารางที่ 2 แสดงการเปรียบเทียบความแตกต่างระหว่าง DarkBASIC และ DarkBASIC Professional

Feature	DarkBASIC	DarkBASIC Professional
Compiler Type	Interpretated	Compiled to Assembly
BSP World Support	No	Yes
Native Windows Editor	No **	Yes
Sphere Mapping	No	Yes
Cubic Environment Mapping	No	Yes
Light Mapping	No	Yes
Bump Mapping	No	Yes
Pixel / Vertex Shaders	No	Yes
FTP Commands	Yes*	Yes

ตารางที่ 2 แสดงการเปรียบเทียบความแตกต่างระหว่าง DarkBASIC และ DarkBASIC Professional (ต่อ)

Feature	DarkBASIC	DarkBASIC Professional
Multiplayer Support	No *	Yes
3D Maths Commands	No	Yes
DLL Support	Yes*	Yes
Particles	No	Yes
DVD Video Playback	No	Yes
CD Audio Playback	Yes	Yes
Memblock Access	Yes*	Yes
Sprite Rotation	No	Yes
Transparent Sprites	No	Yes
Animated Sprites	No	Yes
Stacks / Queues	No	Yes
Clipboard Access	Yes*	Yes
Registry Access	No *	Yes
Force Feedback Support	Yes	Yes
Gradient Filled Boxes	No	Yes

* DarkBASIC supports this but only with the Enhancement Pack installed.

** Alternative Windows editors available.

2.6.2 DarkBASIC Professional Software Requirement

เพื่อให้การทำงานเป็นไปได้อย่างมีประสิทธิภาพ เครื่องคอมพิวเตอร์ที่นำมาใช้ควรมีคุณสมบัติต่างๆ ดังต่อไปนี้

1. 733 MHz Pentium III Processor หรือสูงกว่า
2. Windows 98 / 2000 / ME / XP (Home/Pro)
3. Hard disk space ขนาด 400 MB
4. Ram ขนาด 128 MB
5. DirectX Version 9.0b ขึ้นไป
6. Fully DirectX compatible Graphics Card ซึ่งมี Memory ขนาด 64 MB หรือมากกว่า และ Hardware 3D Acceleration

7. Direct X compatible Sound Card

8. 16x Speed CD-Rom Drive

9. หมายเหตุ องค์ประกอบต่างๆ ภายใน DarkBASIC Professional (เช่น vertex/pixel shaders) ต้องการ graphics card ที่มีประสิทธิภาพสูง

2.6.3 ตัวอย่างของ Features พื้นฐานที่ใช้ใน DarkBASIC Professional

1. Compiler

Complier ของ DarkBASIC Professional ถูกเขียนขึ้นใหม่ทั้งหมด โดยไม่ได้เลียนแบบ Complier ของ DarkBASIC เดิม ทำให้ Complier ของ DarkBASIC Professional มีความสามารถที่เหนือกว่า ดังนี้

1.1 การ Compile โปรแกรมที่ถูกเขียนขึ้นทั้งหมดให้อยู่ในลัษณะของ Optimized Machine Code

1.2 มี Debugger ที่อนุญาติให้สร้าง Breakpoint, Step through และการดูค่าตัวแปรต่างๆ ขณะที่โปรแกรมกำลังทำงาน

1.3 เอาเฉพาะกลุ่มคำสั่งของโปรแกรมที่ถูกเขียนไปสร้างเป็น EXE file ทำให้ได้ File ที่มีขนาดเล็ก

1.4 สามารถสร้าง Stand alone program ซึ่งใช้งานร่วมกับทั้ง Internal และ External file ได้

1.5 สามารถแทน Icon และ Mouse ของโปรแกรมด้วยรูปที่สร้างขึ้นเอง

1.6 Encryption ทุก Internal media ทำให้ผลลัพธ์สุดท้ายมีขนาดเล็กและปลอดภัย

1.7 สามารถสร้างโปรแกรมที่เก็บ DLLs แบบ Internal เพื่อให้ง่ายต่อการทำ EXE Distribution

2. BSP Support and BSP Complier

BSP (Binary Space Partitioning) เป็น 3D Format ชนิดหนึ่งที่นิยมนำมาใช้ในเกมที่เป็นที่รู้จักกันดี เช่น Quake2/3, Half Life ฯลฯ โดยจะเก็บ Scene ต่อไปที่จะวาดไว้ล่วงหน้า ทำให้การวาด Scene ไปยัง Screen เป็นไปได้ด้วยความรวดเร็ว

DarkBASIC Professional BSP Renderer จะแสดง Scene ด้วยความเร็วสูงระดับใด ขึ้นกับการกำหนดค่าคงที่ของ Frame rate

3. Advanced Rendering Techniques

DarkBASIC Professional ประกอบด้วย Built-in commands ที่หลากหลายซึ่งช่วยในการสร้าง Effects ต่างๆ ที่ต้องการให้มีภายในเกมได้มากขึ้น เช่น

3.1 Sphere Mapping จะห้อหุ้ม Texture ชั้นที่ 2 ให้กับ Object แล้วทำให้ Texture นั้นกลมกลืนไปกับ Texture เดิมที่มีอยู่ โดยผลลัพธ์ที่ได้จะเหมือนมีความแวววาวของโลหะบนผิวของ Object นั้น ซึ่งสามารถเปลี่ยนแปลงไปตามการเคลื่อนไหวของ Object หรือมุมกล้องได้

3.2 Cubic Mapping เป็นเทคนิคที่ใช้ Six-sided texture โดยนำเอา Texture ที่นำมาประกอบเป็นฉาก 6 Texture มาย่อขนาด แล้วนำไปห่อหุ้ม Object ทำให้ Object ดูเหมือนสามารถสะท้อนภาพของสิ่งแวดล้อมรอบๆ ตัวออกมา ซึ่งเปลี่ยนแปลงได้ตามการเคลื่อนไหวของ Object

3.3 Bump Mapping เป็นเทคนิคที่ใช้เพิ่มรายละเอียดให้กับ Object โดยนำเอา Texture ของ Object ไปคำนวณหาทิศทางของแสงที่จะตกกระทบ Object ทีละ Pixel ทำให้ Object ที่มีลักษณะเป็น Low-polygon model คงรูปและมีรูปร่างที่เปลี่ยนไปในที่ถูกขณะชน

3.4 Multi-Texturing ทำการรวม Texture ได้สูงสุด 8 Texture บน Object หนึ่งๆ เช่นเดียวกับ Microsoft Direct3D

3.5 Light Mapping

Light map เป็น Texture 1 Texture หรือ กลุ่มของ Texture ที่เก็บข้อมูลเกี่ยวกับแสงใน 3D Scene หนึ่งๆ ถึงแม้ว่า DarkBASIC Professional จะสามารถสร้างแสงแบบ Point, Omni และ Spot light ได้ แต่บางครั้งแสงเหล่านั้นอาจจะไม่เหมาะกับภาพภายในเกม DarkBASIC Professional จึงอนุญาติให้สร้าง Light mapped texture ได้ตามที่ต้องการ แล้วจะทำการใส่ Light mapped texture นั้นให้กับ Object ทำให้ได้ผลลัพธ์ที่ดี และไม่ทำให้เกิน Processing overhead

4. Particles เป็นส่วนรายละเอียดเล็กๆ น้อยๆ ที่ช่วยทำให้เกมดูสมจริง DarkBASIC Professional ทำให้การสร้างและควบคุมการทำงานของ Particles ได้ง่ายขึ้น

5. Reflections เป็นการสร้างภาพการสะท้อนขององค์ประกอบต่างๆ ภายใน Scene บนรูปร่างของ Object โดยจะ Render ภาพสะท้อนให้กับ Object จากการพิจารณาการทำมุมของ Object กับองค์ประกอบเหล่านั้น

6. Box, Sphere and Polygon Collision ใช้ในการตรวจสอบการชนกันของ Objects โดย Box และ Sphere Collision จะเหมาะกับการตรวจสอบการชนเพื่อหลบหลีกสิ่งกีดขวาด เนื่องจากการต้องการความเร็วในการตรวจสอบ โดยเฉพาะ

กรณีที่มีสิ่งกีดขวางอยู่เป็นจำนวนมาก ส่วน Polygon Collision จะเหมาะกับการตวรจสอบการชนที่ต้องการรายละเอียดเกี่ยวกับการชนนั้น เนื่องจากสนใจความถูกต้องแม่นยำมากกว่าการตรวจสอบอย่างรวดเร็ว

7. Shadows ใช้ในการสร้างเงาให้กับ Object ใดๆ ที่ต้องการ โดยเงาสามารถพาดผ่านวัตถุได้ทุกชนิด ไม่ว่าจะเป็นตัว Object เอง, Object อื่นๆ หรือ ฉาก

8. Bone Animation เป็น Mesh รูปแบบหนึ่งที่ใช้สำหรับเปลี่ยนลักษณะการเคลื่อนไหวของ Animation โดยกลุ่มของ Bone จะถูกนำมาประกอบกันเป็นโครงสร้างของ Model ซึ่งถูกผังอยู่ภายใน Model นั้นๆ เพื่อนำไปใช้ในการสร้าง Animation ในลักษณะต่างๆ ให้กับ Model

9. Debugger เป็นสิ่งที่ใช้ในการตรวจสอบการทำงานของโปรแกรม โดยสามารถดูค่าของตัวแปรต่างๆ , ลำดับการทำงานของโปรแกรมที่ละคำสั่ง รวมไปถึงเพิ่มคำสั่งบางคำสั่ง ขณะโปรแกรมกำลังทำงานอยู่ได้

10. Editor

DarkBASIC Professional ได้พัฒนา Editor ที่มี Interface ซึ่งช่วยให้ผู้พัฒนาเกมสามารถพัฒนาเกมของตนได้อย่างรวดเร็ว เช่น Project Management Tool, Editor Option, Syntax formatting, Search และ Replace, Media Brower, Command line parameters, Project summaries รวมไปถึง การควบคุม Breakpoint สำหรับ Debugging

11. Fast 2D Sprites

ถ้าโปรแกรมสามารถวาด Sprites ได้เร็วมากเท่าไหร่ จำนวน Sprites ที่ปรากฏในเกมก็มีได้มากขึ้นเท่านั้น DarkBASIC Professional ช่วยให้การวาด Sprites จำนวนมากทำได้อย่างรวดเร็ว จนทำให้จำนวน Sprites ที่มีได้ภายในเกมขึ้นอยู่กับความเร็วของ 3D Card ที่มี

12. DLL Expandability

DarkBASIC Professional อนุญาติให้ผู้ใช้สามารถสร้างคำสั่งใหม่ ซึ่งไม่เคยปรากฏอยู่ในภาษา DarkBASIC Professional มาก่อน ด้วยการกำหนดชื่อคำสั่งและเขียนการทำงานของคำสั่งนั้นใน DLLs (Dynamically Linked Libraries) file แล้วนำไปเก็บใน Folder ของ DarkBASIC Professional ซึ่งวิธีนี้เป็นการเพิ่มคำสั่งให้กับภาษา DarkBASIC Professional ได้ทางหนึ่ง

2.6.4 Feature อื่นๆ ที่เพิ่มความสามารถในการทำงานของ DarkBASIC Professional

1. Dark Physics เป็น Physics system ที่สมบูรณ์ โดยอาศัยพื้นฐานของ Ageia PhysX system ทำให้สามารถสร้างความสมจริงภายในเกม ไม่ว่าจะเป็นเสื้อผ้าที่ฉีกขาดจากการต่อสู้ หรือ ร่องรอยของกระสุนเมื่อถูกยิง, หมอกและควัน บริเวณที่เกิดการชนกันของ Object ภายใน Scene (Particle), ยานพาหนะที่กำลังเคลื่อนที่, ความหนืดของของเหลว ฯลฯ

2. ConvSEO (Convert SketchUp-Exported Objects) เป็น Plug-in ทีช่วยให้การใช้งาน Model ซึ่งสร้างด้วย Google SketchUp กับ DarkBASIC Professional โดยตรงทำได้ง่ายยิ่งขึ้น

3. Dark A.I. เป็นส่วนที่ช่วยสร้าง A.I. ให้กับตัวละครต่างๆ ภายในเกม เช่น Path Finding, Zones, Teams, Entity Commands, Manual Control, Avoidance, Containers เป็นต้น

4. Enhancement Expansion Pack เป็นผลิตภัณฑ์ใหม่ของผู้ผลิต DarkBASIC Professional ที่ช่วยเพิ่มประสิทธิภาพให้กับ DarkBASIC Professional ด้วย Feature ต่างๆ ดังนี้

4.1 Ogg Vorbis

4.2 File Blocks

4.3 File Mapping

4.4 Speech

4.5 CPU Information

4.6 Memory

4.7 Environmental Audio (EAX) Support

4.8 Drive Information

4.9 Display

4.10 Common Dialogs

4.11 Internet

4.12 Data Files

5. Extends Expansion Pack เป็นชุดคำสั่งที่เพิ่มความสามารถหลักของ DarkBASIC Professional ซึ่งประกอบด้วย Feature ต่างๆ ดังนี้

5.1 Dynamic Media Handling

5.2 XGui System

5.3 XFont System

5.4 2D FX

5.5 Perfect Pixel Sprite Collision

5.6 Billboard system

5.7 Virtual 3D Lights system

5.8 Custom Particle System

5.9 Real-Time Sky System

5.10 Bonus: Asteroids Game

6. BlueGUI Windows Commands Pack เป็นชุดคำสั่งที่ช่วยสร้าง Windows control หรือ gadgets เช่น Buttons, Tabs, Treeviews ฯลฯ ในโปรแกรมภาษา DarkBASIC Professional

7. Unity Scripting Pack (LUA5) เป็นชุดคำสั่งที่ช่วยให้ DarkBASIC Professional สามารถใช้งานร่วมกับ Lua scripting language ซึ่งเป็นภาษาที่ผู้ผลิตเกมนิยมใช้ในปัจจุบัน

8. EZrotate Enhanced เป็นชุดคำสั่งที่เปลี่ยนการคำนวณทางคณิตศาสตร์ที่ยุ่งยากให้กลายเป็นเรื่องง่ายสำหรับการเขียนโปรแกรม ทำให้สามารถใช้คำสั่งทั่วไปแทน Math routines ที่ซับซ้อน ในเรื่องต่างๆ ดังต่อไปนี้

8.1 Conversion

8.2 True Global

8.3 Local, Orbit และ Vector Rotation

8.4 Rotate และ Turn Pitch Towards

8.5 Find Axis Angles

8.6 Find Point From Offset

8.7 Find Offset From Point

8.8 Find Total Angle Between Points

9. Nuclear Glory Collision DLL เป็นชุดคำสั่งที่ช่วยจัดการเกี่ยวกับการตรวจสอบการชนที่ซับซ้อนให้เป็นเรื่องที่สามารถทำได้ในเกม โดยจะคำนวณการชนให้กับตัวละครตลอดเวลา

10. TextureMax เป็นเครื่องมือที่ใช้ในการแยก Texture ของ Object ออกเป็นส่วนๆ แล้วกำหนดสีที่แตกต่างกันให้กับแต่ละส่วนของ Texture นั้น ทำให้ Object แบบเดียวกับ สามารถสร้างเป็น Objects ที่มีสีแตกต่างกันได้เป็นจำนวนมาก

11. Cloth and Particles Expansion Pack เป็นชุดคำสั่งที่ใช้สำหรับสร้าง Emitters (สิ่งที่ใช้ในการสร้าง Particles), Effectors (สิ่งที่ใช้สร้าง Effect ต่างๆ เช่น ลมพัด เป็นต้น), colliders (สิ่งที่ใช้ควบคุมการชนกันของ Object) และ Cloth (สิ่งที่ใช้สร้างการเคลื่อนไหวของธง หรือสิ่งที่ทำด้วยผ้าอื่นๆ)

2.6.5 File Formats ที่สามารถนำไปใช้ใน DarkBASIC Professional

ตารางที่ 3 แสดง File Formats ที่ DarkBASIC Professional รองรับ

File Format	Name	Load	Save
2D Image - BMP	BMP	Yes	Yes
2D Image - Device Independant Bitmap	DIB	Yes	Yes
2D Image - Portable Network Graphics	PNG	Yes	
2D Image - JPEG	JPG	Yes	Yes
2D Image - DirectDraw Surface	DDS	Yes	Yes
2D Image - Targa	TGA	Yes	
3D Object - 3D Studio	3DS	Yes	
3D Object - DirectX	X	Yes	Yes
3D Object - MDL	MDL	Yes	
3D Object - MD2	MD2	Yes	
3D Object - MD3	MD3	Yes	
BSP World Support	Quake 2	Yes	
BSP World Support	Quake 3	Yes	
BSP World Support	Half Life	Yes	
Sound - Wave File	WAV	Yes	Yes
Sound - Windows Media Audio	WMA	Yes	
Sound - Windows Media Audio	AIFF	Yes	
Sound - Windows Media Audio	AU	Yes	
Sound - Windows Media Audio	SND	Yes	
Sound - Mpeg Layer 3	MP3	Yes	
Music - MIDI playback	MIDI	Yes	
Music - CD Audio Tracks	CD-Audio	Yes	

ตารางที่ 3 แสดง File Formats ที่ DarkBASIC Professional รองรับ (ต่อ)

File Format	Name	Load	Save
Animation	AVI	Yes	
Animation	MPEG	Yes	
Animation	Indeo	Yes	
Animation	Cinepak	Yes	

* DVD Playback ต้อง Install Windows codec ก่อน