

การพัฒนาเกมแนวต่อสู้โดยใช้ชุดคำสั่งซีดีเอกซ์เอนจิน และอัลกอริธึมแบบเอ-สตาร์

สัญญา เครือหงษ์* และ สุรางค์รัตน์ เชาวโคกสูง*

บทคัดย่อ

งานวิจัยนี้ได้นำเสนอการพัฒนาเกมแนวต่อสู้โดยใช้ชุดคำสั่งสำหรับการสร้างเกม (CDX Engine) มาเป็นส่วนประกอบของโปรแกรมในด้านการแสดงผลมัลติมีเดีย ซึ่งเป็นเครื่องมือในการอำนวยความสะดวกต่อการพัฒนาโปรแกรมเกม อาทิ การแสดงผลทางด้านภาพ เสียง และการติดต่อกับอุปกรณ์อินพุต รวมทั้งการนำหลักการของปัญญาประดิษฐ์ (Artificial Intelligence) แบบเอ-สตาร์ (A-Star Algorithm) เข้ามาร่วมด้วยเพื่อใช้สำหรับพัฒนาระบบโจมตีของศัตรูโดยอัตโนมัติ จากนั้นจึงทำการออกแบบตามแนวความคิดโครงสร้างและส่วนประกอบต่างๆ ของเกม ทั้งตัวละคร ฉาก และเสียงประกอบ เมื่อพัฒนาโปรแกรมเกมเสร็จ ผลการประเมินโดยใช้แบบสอบถามจากผู้ทดลองใช้จำนวน 50 คน ซึ่งมีผลคะแนนเฉลี่ยคือ 4.22 และส่วนเบี่ยงเบนมาตรฐาน 0.57 ถือว่าอยู่ในระดับดี

คำสำคัญ: เกมแนวต่อสู้ ชุดคำสั่งสำหรับการสร้างเกม ซีดีเอกซ์เอนจิน อัลกอริธึมแบบเอ-สตาร์

1. บทนำ

การพัฒนาเกมและแอนิเมชันในปัจจุบันขยายตัวเป็นอุตสาหกรรมที่พัฒนาขึ้น และสามารถพัฒนามาจากโปรแกรมที่หลากหลายมากขึ้น สมควรที่จะมีการฝึกฝนทักษะการพัฒนาในด้านดังกล่าว โดยเลือกภาษาที่นิยมใช้ คือ Microsoft Visual C++ ควบคู่กับ CDX engine

CDX Engine [1] คือชุดคำสั่งของการสร้างเกมใช้กับภาษา C/C++ ที่ถูกพัฒนาการเขียนมาเพื่อช่วยลดความซับซ้อนของการสร้างเกมด้วยไดเรกต์เอ็กซ์ (DirectX) ซึ่ง CDX Engine จะถูกเรียกใช้เพื่อให้ง่ายขึ้นโดยใช้ไดเรกต์เอ็กซ์ได้ง่ายขึ้น จากนั้น

ไดเรกต์เอ็กซ์จะเป็นตัวจัดการเรื่องการแสดงผลของเกม (DirectDraw) จัดการเกี่ยวกับเรื่องเสียง (DirectSound) และจัดการเรื่องการรับข้อมูลจากอุปกรณ์นำเข้า (DirectInput) ซึ่งสามารถจัดการทั้ง 3 ระบบ ดังกล่าว ได้ในชุดคำสั่ง DirectDraw ตัวเดียวกัน ซึ่งจากเดิมต้องมีระบบจัดการที่แยกจากกันคือ ชุดคำสั่ง GDI (Graphics Device Interface) และชุดคำสั่ง MCI ในการเล่นไฟล์เสียง หรือการแสดงรูปภาพ

เกมต่อสู้ [2] (Fighting Game) เป็นเกมที่สนุกสนานและเป็นที่นิยมมาก คู่ต่อสู้หรือศัตรูในเกมจะสามารถต่อสู้กับเราโดยอัตโนมัติ ดังนั้น ควรจะพัฒนาศัตรูในเกมให้สามารถต่อสู้กับผู้เล่นได้อย่าง หลากหลายมากขึ้น เช่น อาจแบ่งอาวุธหรือทักษะ (Skill) ของศัตรูให้เป็นแต่ละระดับตามระดับพลังมากน้อยของศัตรูในเกม ทั้งนี้การพัฒนาศัตรูในเกมดังกล่าว ควรจะใช้หลักการของปัญญาประดิษฐ์เข้ามาช่วยเพื่อเสริมสร้างการพัฒนาเกมของเราให้ดียิ่งขึ้น และทำให้เกมเป็นที่น่าสนใจมากขึ้นด้วย

2. แนวคิดและหลักการ

การพัฒนาเกมหรือการเขียนโปรแกรม [3] ที่มีการออกแบบและวิเคราะห์โปรแกรมไว้เป็นอย่างดี เช่น มีการออกแบบตัวแปรและฟังก์ชันต่างๆ ที่แก้ไขได้ง่าย หรือออกแบบที่เน้นการเขียนโปรแกรมเชิงวัตถุ (OOP) ทำให้โปรแกรมนั้นออกมาได้ดี สามารถแก้ไขและพัฒนาโปรแกรมต่อได้ง่าย รวมทั้งเป็นพื้นฐานของการเขียนโปรแกรมต่อไปได้ดี

การพัฒนาศัตรูในเกมให้หลากหลาย [4] ควรพัฒนาให้มีการเคลื่อนไหว มีอาวุธและความสามารถในการต่อสู้ที่เป็นอัตโนมัติ ควรเสริมให้ศัตรูมีการต่อสู้ที่คิดได้เอง แบบปัญญาประดิษฐ์ (Artificial Intelligence) ซึ่งภายในเกมผู้เล่นจะ

* คณะวิทยาศาสตร์ มหาวิทยาลัยนครสวรรค์



สามารถเลือกตัวผู้เล่นได้ตาม ประเภทที่ผู้เล่นต้องการ เช่น นักเวทมนตร์ นักต่อสู้ด้วยดาบ โดยผู้เล่นแต่ละตัวจะมีความสามารถในการปล่อยพลัง ระดับพลัง (HP) และระดับเวทย์มนตร์ (SP) ที่แตกต่างกัน จากนั้นผู้เล่นสามารถเลือกด่านต่อสู้ตามภูมิประเทศที่ต้องการได้ เมื่อเข้าสู่การต่อสู้ศัตรูในเกมจะสามารถปล่อยพลังต่อสู้และหาตำแหน่งของผู้เล่นเพื่อเดินเข้าโจมตีผู้เล่นได้อัตโนมัติ ซึ่งการปล่อยพลังนั้นจะมีการเปลี่ยนรูปแบบตามระดับพลัง (EnemyHP) ที่เหลืออยู่ของศัตรูไปตามระดับในแต่ละชั้น จนศัตรูพ่ายแพ้ไปในที่สุด

3. ประเภทของซอฟต์แวร์เกม

ซอฟต์แวร์เกมนั้น [2], [5], [6] แม้ว่าในปัจจุบันซอฟต์แวร์จะถูกพัฒนาขึ้นตามกาลเวลา หลายเกมที่ใช้แนวคิดแปลกๆ ในการเล่นหรือนำเกมหลายประเภทมารวมกันจนเกิดเป็นแนวใหม่ขึ้นมา แต่เรายังอาจแบ่งประเภทของเกมออกเป็นประเภทใหญ่ๆ ได้ดังนี้

3.1 เกมแอ็คชั่น (Action game)

เกมประเภทนี้มักได้รับความนิยมสูงในหมู่คนทั่วไป เนื่องจากเป็นประเภทของเกมที่ได้ง่ายที่สุด ใช้การบังคับทิศทางและการกระทำของตัวละครในเกมเพื่อผ่านด่านต่างๆ ไปให้ได้มีตั้งแต่เกมที่มีรูปแบบง่ายๆ เหมาะกับคนทุกเพศทุกวัยอย่างเช่น เกมมาริโอ้ ร็อคแมนไปจนถึงเกมแอ็คชั่นที่มีเนื้อหารุนแรงไม่เหมาะสำหรับเด็ก บางเกมมีการใส่ลูกเล่นต่างๆ เข้ามาเพิ่มความสนุกของเกม จนกลายเป็นเกมแนวใหม่ไปเลย เช่น Sneak Tactical Action คือ เกมแอ็คชั่นที่ไม่เน้นการบุกตะลุย แต่ใช้การหลอกล่อฝ่ายศัตรูเพื่อผ่านอุปสรรคไปให้ได้

3.2 เกมสวมบทบาท (RPG: Role-Playing Game) หรือ อาร์พีจี

ที่นิยมเรียกกันว่าเกมภาษา เนื่องจากในช่วงแรกเกมอาร์พีจีที่ออกมาจะเป็นภาษาอังกฤษหรือญี่ปุ่น ซึ่งต้องใช้ความรู้ด้านภาษานั้นๆ ในการเล่น เกมประเภทนี้จะกำหนดตัวผู้เล่นอยู่ในโลกที่สมมติขึ้น และให้ผู้เล่นสวมบทบาทเป็นตัวละครหนึ่งในโลกนั้นๆ ผจญภัยไปตามเนื้อเรื่องที่กำหนด โดยมีจุดเด่นทางด้านการพัฒนาระดับของตัวละคร (Experience-ประสบการณ์) เมื่อผจญภัยไปมากขึ้น และเอาชนะศัตรูตัวร้ายที่สุดในเกมให้ได้ ตัวเกมไม่เน้นการบังคับทิศทางหรือหา แต่จะให้ผู้ผู้เล่นสัมผัสกับเรื่องราวแทน

3.3 เกมการยิง (Shooting game)

เกมประเภทนี้จะเน้นการยิงศัตรูเป็นหลัก ตัวผู้เล่นในเกมมักจะมีอุปกรณ์ประเภทปืนหรือเครื่องบินที่ยิงกระสุนได้ไม่มีวันหมด ทำการฝ่าฟันอุปสรรคเพื่อไปสู่ฉากต่อไปด้วยการยิงทำลายเป้าหมายในเกมให้ได้ ในสมัยก่อนเกมประเภทนี้มักจะพบในรูปแบบของยานอวกาศ (หรือเครื่องบิน) ความสนุกของมันอยู่ที่การทำลายศัตรูด้วยอาวุธที่มี ขณะเดียวกันก็ต้องคอยหลบกระสุนของฝ่ายศัตรูให้ได้

3.4 เกมผจญภัย (Adventure game)

เกมประเภทนี้จะเน้นไปที่เรื่องราวของเกม แต่จะต่างจากเกม RPG ตรงที่อาจไม่มีการเพิ่มระดับของตัวละครก็ได้ และในการเล่นก็ไม่จำเป็นต้องใช้การบังคับที่ยากนัก แต่สิ่งสำคัญคือการเปิดเผยเรื่องราวที่ชวนติดตามไปทีละน้อยๆ จนกระทั่งถึงตอนจบ เนื้อเรื่องมักจะอิงเรื่องที่เกี่ยวข้องกับความลึกลับหรือการผจญภัยตื่นเต้นทั้งหลาย

3.5 เกมต่อสู้ (Fighting game)

เกมประเภทนี้ต่างจากเกมแอ็คชั่นตรงที่ จะไม่มีการผ่านด่านระดับสูงขึ้น แต่จะเน้นการต่อสู้ตัวต่อตัว (บางครั้งก็เป็นการแตกทีมสู้) มีการเตะ ต่อย และใช้ศิลปะการต่อสู้ประเภทต่างๆ เข้าสู้กัน รวมถึงมีการทำท่าซัดหรือที่เรียกว่า คอมโบ (Combo) ด้วย เป็นที่นิยมมากในหมู่เพื่อนฝูง เพราะการเล่นขึ้นอยู่กับฝีมือของผู้ควบคุมเป็นหลัก ไม่ใช่ตัวละครในเกม ซึ่งเกมแนวนี้ ก็มักจะทำให้ตัวละครในเกมมีความสามารถแตกต่างกัน ผู้เล่นมักจะเลือกใช้ตัวละครที่ตนถนัดหรือใช้ประจำในการต่อสู้กับผู้เล่นอื่น เกมต่อสู้ที่ได้รับความนิยมเช่น เกม Street Fighter เกม Tekken

3.6 เกมปริศนา (Puzzle game)

เกมประเภทพัซเซล เป็นเกมแนวที่เล่นได้ทุกวัย ตัวเกมมักจะเน้นการแก้ปริศนา ปัญหาต่างๆ มีตั้งแต่ระดับง่ายไปจนถึงซับซ้อน ในอดีตตัวเกมมักนำมาจากเกมปริศนาตามนิตยสาร เช่น เกมตัวเลข เกมอักษรไขว้ ต่อมาจึงมีเกมปริศนาที่เล่นบนคอมพิวเตอร์อย่างเกมเททริสออกมา ปัจจุบันมีเกมแนวพัซเซลแบบใหม่ๆ ออกมามากมาย เกมแนวนี้เป็นเกมที่เล่นได้ทุกยุคทุกสมัย จึงเป็นเรื่องปกติที่จะเห็นผู้เล่นบางคนยังติดใจกับเกมเททริส เกมอาร์คานอยด์ ไปจนถึงเกมพัซเซลใหม่ๆ อย่าง Polarium หรือ Puzzle Bubble

3.7 เกมการจำลอง (Simulation game)

เป็นเกมประเภทที่จำลองสถานการณ์ต่างๆ มาให้ผู้เล่นได้สวมบทบาท เป็นผู้อยู่ในสถานการณ์นั้นๆ และตัดสินใจในการกระทำเพื่อลองดูว่าจะเป็นอย่างไร เหตุการณ์ต่างๆ อาจจะ

นำมาจากสถานการณ์จริงหรือสมมติขึ้นก็ได้

3.8 เกมวางแผนการรบ (Strategy game)

เกมประเภทนี้แยกย่อยมาจากเกมแนว Simulation เนื่องจากในระยหลังเกมประเภทนี้มีแนวทางของตัวเองที่ชัดเจนขึ้น คือเกมที่เน้นการควบคุมกองทัพซึ่งประกอบไปด้วยหน่วยทหารย่อยๆ เข้าเข้าทำการสู้รบกัน พบมากในเครื่องคอมพิวเตอร์เนื่องจากคีย์บอร์ดและเมาส์เอื้อต่อการควบคุมเกม และยังมีมักจะจะสามารถเล่นร่วมกันได้หลายคนอีกด้วย เนื้อเรื่องในเกมมีตั้งแต่จับความสัต์เวทย์มนตร์คาถา พ่อมด กองทหารยุคกลาง ไปจนถึงสงครามระหว่างดวงดาวเลยทีเดียว

3.9 เกมกีฬา (Sport game)

เป็นเกมที่จำลองการเล่นกีฬาชนิดต่างๆ ให้เล่นกันในแบบของวิดีโอเกม นิยมเล่นกันในหมู่เพื่อนฝูงเนื่องจากเข้าใจง่าย ใช้กติกาหลักๆ เหมือนกีฬาจริงๆ และใช้ฝีมือของผู้เล่นเองในการเอาชนะ ไม่มีตัวช่วยในเกมมากนัก ปัจจุบันมีเกมกีฬาออกมาหลายประเภทตั้งแต่เกมฟุตบอล เกมบาสเก็ตบอล เกมกอล์ฟ เกมชกมวย

3.10 เกมแข่งความเร็ว (Racing game)

ในอดีตจะเข้าใจกันว่าเป็นเกมแนวแข่งรถ เนื่องจากมักจะใช้รถแข่งหรือรถจักรยานยนต์มาแข่งขันกัน แต่ปัจจุบันมีหลากหลายการแข่งขันมากมายเช่นแข่งวิ่งเร็วเหนือมนุษย์ แข่งขี่ม้า แข่งรถโกคาร์ท เป็นต้น เป็นเกมอีกประเภทที่ได้รับความนิยมเนื่องจากเล่นง่าย สามารถตัดสินใจได้ในเวลาอันสั้น

3.11 เกมสารพันบันเทิง (Variety game)

เกมประเภทนี้มักจะเป็นเกมที่เน้นการเล่นแข่งกับผู้เล่นคนอื่นด้วยกันเพื่อความบันเทิงแบบง่ายๆ ตัวเกมจะบรรจุไว้ด้วยเกมย่อยๆ อีกหลายชนิด ผู้เล่นสามารถเลือกเล่นเกมประเภทต่าง ๆ ที่มีให้มาเล่นแข่งกันในเวลาสั้น ๆ ได้อย่างสนุกสนาน

3.12 เกมกระดาน (Board game)

เป็นเกมที่จำลองเกมกระดานของจริงประเภทเกมเศรษฐี เกมโมโนโพลี (Monopoly) หรือเกมกระดานอื่น ๆ มาใส่ลูกเล่นลงไปมากขึ้น เกมประเภทนี้นิยมเล่นแข่งกันในหมู่เพื่อนฝูงมากกว่าจะแข่งกับคอมพิวเตอร์ ด้วยความสามารถของตัวซอฟต์แวร์ทำให้สามารถเพิ่มรายละเอียดที่เกมกระดานของจริงทำไม่ได้ลงไปได้อย่างมากมาย

3.13 การ์ดเกม (Card game)

เกมประเภทนี้มีตั้งแต่เกมไพ่ธรรมดาไปจนถึงเกมการ์ดสะสม (Collectible card game) ที่นำมาลงบนเครื่องเกมเพื่อให้แสดงผลและสภาพแวดล้อมที่สวยงาม บางเกมเป็นเกมประเภทอื่นๆ แต่เพิ่มส่วนของการสะสมเข้าไปเพื่อดึงดูดให้ผู้เล่นเกมได้นานๆ ก็มี

3.14 เกมเข้าจังหวะ (Rhythm action game)

ในช่วงระยะไม่กี่ปีมานี้มีเกมอีกประเภทหนึ่งที่ถูกคิดค้นขึ้นและได้รับความนิยม จนกลายเป็นประเภทของเกมไปเป็นที่สุดนั่นก็คือเกมประเภทเข้าจังหวะ จุดเด่นของเกมประเภทนี้คือเสียงดนตรี และแนวการเล่นที่ให้ผู้เล่นเกมกดปุ่มหรือเคาะจังหวะบนเครื่องควบคุมเพื่อสร้างเสียงต่างๆ ในเกม โดยที่ตัวเกมจะกำหนดเครื่องดนตรีมาให้ เป็นเกมประเภท Simulation อย่างหนึ่งที่แยกออกมาเป็นเกมแนวเข้าจังหวะไปในที่สุด

4. ปัญญาประดิษฐ์ (Artificial Intelligence)

ปัญญาประดิษฐ์เป็นศาสตร์หนึ่งของสาขาวิชาวิทยาการคอมพิวเตอร์ อันก่อกำเนิดจากความพยายามเริ่มแรกของมนุษย์ในการนำเครื่องคอมพิวเตอร์มาช่วยในการค้นหาคำตอบหรือตัดสินใจปัญหาที่ต้องใช้ข้อมูลความรู้ที่มีจำนวนมาก รวมถึงการแก้ปัญหาที่ต้องใช้ความรู้หรือเวลาในการคำนวณค่อนข้างมาก ด้วยประสิทธิภาพของเครื่องคอมพิวเตอร์ในปัจจุบันที่มีความสามารถในการคำนวณสูงขึ้นอีกทั้งยังสามารถรองรับการเก็บข้อมูลขนาดใหญ่ที่มีจำนวนมากได้ ส่งผลให้งานทางด้านปัญญาประดิษฐ์ถูกพัฒนาขึ้นอย่างกว้างขวางและมีการใช้งานอย่างแพร่หลายมากขึ้น นอกจากนั้นงานเฉพาะด้านในหลากหลายสาขาล้วนมีความต้องการใช้ข้อมูลที่มีความละเอียดรวมไปถึงการใช้ข้อมูลหรือฐานความรู้ในการดำเนินงาน จึงอาศัยแนวทางการพัฒนางานด้านปัญญาประดิษฐ์มาทำการสร้างเครื่องมืออำนวยความสะดวกสำหรับงานเพื่อช่วยในการวิเคราะห์ หรือตัดสินใจปัญหาแทนการใช้มนุษย์ แม้ว่าการสร้างงานทางด้านปัญญาประดิษฐ์นั้นจะมีขั้นตอนที่ยุ้งยากซับซ้อนและมีรายละเอียดที่ต้องพิจารณาค่อนข้างมาก อย่างไรก็ตามแนวทางของงานแต่ละด้านจะมีรูปแบบหรือขั้นตอนการพิจารณาของปัญหาที่เฉพาะเจาะจงแตกต่างกัน ซึ่งจะขึ้นอยู่กับลักษณะของปัญหานั้น ๆ

ปัญญาประดิษฐ์ในเกม [7] จะเป็นการทำให้สภาพแวดล้อมในเกมรอบตัวผู้เล่นสามารถตอบสนองกับผู้เล่นได้ในขอบเขตของตัวเกม โดยจะมีวิธีการหลัก ๆ อยู่ 3 วิธี

และการสุ่ม ตรรกะ จะเป็นการให้คอมพิวเตอร์อ่านข้อมูลจากสถิติที่มีเก็บไว้ หรือสถานการณ์ปัจจุบัน เพื่อเลือกวิธีการที่จะตอบสนองต่อข้อมูลเหล่านั้น

การสุ่มเป็นการให้คอมพิวเตอร์เลือกวิธีการที่จะคิดหรือตอบสนองขึ้นมาว่า ๆ ในขอบเขตที่กำหนด

ตรรกะจะทำให้ปัญญาประดิษฐ์ในเกมฉลาด ที่จะตอบสนองกับคนอย่างมีเหตุผล ส่วนการสุ่มจะทำให้ AI ทำในสิ่งที่ผู้เล่นไม่สามารถคาดเดาได้ ซึ่งจะมีการใช้อย่างใดอย่างหนึ่งหรือใช้ร่วมกันตามแต่ผู้เขียนโปรแกรมจะวางแผนไว้ การใช้แต่ตรรกะจะทำให้ ปัญญาประดิษฐ์มีขีดจำกัด สามารถคาดเดาได้ และตายตัวแน่นอน ส่วนการใช้แต่การสุ่มจะทำให้ปัญญาประดิษฐ์ทำอะไร ๆ เพราะมันไม่ได้คิด

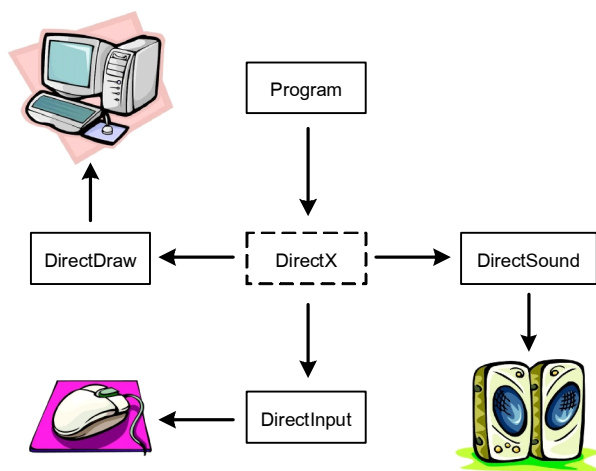
5. ลักษณะของ DirectX

DirectX [4], [8] เป็นซอฟต์แวร์หรือชุดคำสั่ง ที่พัฒนาโดยบริษัทไมโครซอฟท์ ออกแบบมาเพื่อช่วยในการรองรับงานด้านงานมัลติมีเดีย เป็นชุดคำสั่งที่ทำให้ผู้เขียนโปรแกรมเรียกใช้ความสามารถด้าน hardware ของระบบได้อย่างเต็มประสิทธิภาพซึ่งรวมถึงสามารถนำมาพัฒนาเกมคอมพิวเตอร์ได้เป็นอย่างดี

DirectX ประกอบด้วย 2 ส่วนที่สำคัญ คือ

- 1) DirectX Runtime เป็นส่วนของผู้เล่นเกม
- 2) DirectX SDK (DirectX Software Development Kit)

เป็นส่วนของผู้เขียนเกม



ภาพที่ 1 แสดงการจัดการอุปกรณ์ต่างๆของ DirectX

การนำไคเร็กเอ็กซ์มาพัฒนาเกม [8], [9] ก่อนอื่นต้องเข้าใจก่อนว่าไคเร็กเอ็กซ์ช่วยอะไรได้บ้าง ซึ่ง DirectX มีชุดคำสั่ง

ที่ช่วยในการจัดการหลายๆ เรื่อง ที่สำคัญได้แก่

- 1) จัดการเกี่ยวกับเรื่องของการแสดงผล โดยชุดคำสั่งที่เรียกว่า DirectDraw
- 2) จัดการเกี่ยวกับเรื่องเสียง โดยชุดคำสั่งที่เรียกว่า DirectSound
- 3) จัดการเกี่ยวกับเรื่องของการรับข้อมูลจาก keyboard, mouse หรือ joystick โดยชุดคำสั่งที่เรียกว่า DirectInput

6. ลักษณะของ CDX Engine [1], [9]

การสร้างเกมสำหรับ ระบบปฏิบัติการวินโดวส์ในปัจจุบันได้มีการพัฒนาชุดคำสั่งขึ้นมา เพื่อให้อ่านหน่วยความสะดวกในการเขียนเกม ซึ่งเป็นชุดคำสั่งที่ทำให้ผู้เขียนโปรแกรมเรียกใช้ความสามารถด้านฮาร์ดแวร์ของระบบได้อย่างเต็มประสิทธิภาพ ไม่ว่าจะเป็นภาพ เสียง หรือการควบคุมเกม ชุดคำสั่งนี้มีชื่อว่า CDX

CDX เป็นชุดคำสั่งสำเร็จรูปที่ถูกสร้างขึ้นมาเพื่อใช้ในการเขียนเกม โดยมีการเตรียมส่วนประกอบสำหรับการเขียนเกมไว้อย่างสมบูรณ์ เช่น การสร้างภาพ ภาพเคลื่อนไหว เสียง เพลงประกอบเกม วิดีโอ การสร้างเมนูเกม เป็นต้น CDX ถูกพัฒนาขึ้นมาโดย Dan Farley Bil Simser

ถ้าเขียนโปรแกรมเกมด้วย C/C++ กับ ไคเร็กเอ็กซ์ เพื่อสร้าง Surface โหลดภาพและทำการกลับภาพ (Flip) ชุดคำสั่งเกม CDX มันจะเป็นตัวกลางเพื่อให้เรียกใช้ DirectX ได้ง่ายขึ้นซึ่ง ทำให้ช่วยลดขั้นตอนในการสร้างเกมได้มากขึ้น

การใช้ชุดคำสั่ง CDX จะช่วยลดขั้นตอนได้มากในการสร้างเกม หากได้ศึกษาการเขียนโปรแกรมเกมด้วย C/C++ และ DirectX มาก่อน การมาศึกษาชุดคำสั่ง CDX จะช่วยให้เข้าใจถึงหลักการเขียนเกมมากขึ้น เพราะว่าถ้าเขียนโปรแกรมสร้าง Surface เองได้ เขียนโปรแกรมให้ฟิลิปเอง โหลดภาพเอง เมื่อมาใช้ชุดคำสั่ง CDX จะทำให้รู้เองว่าจะต้องเรียกใช้อย่างไร ทั้งนี้เพราะหลักการและวิธีการนำไปใช้นั้นคล้าย ๆ กัน แต่ถ้าไม่มีพื้นฐานในการเขียนเกมมาก่อน การศึกษาการใช้ชุดคำสั่ง CDX เลย ไม่ใช่เรื่องยาก

7. องค์ประกอบของ CDX Engine

ใน CDX จะประกอบไปด้วยคลาสหลาย ๆ คลาส ที่ทำหน้าที่ต่าง ๆ กัน ดังนี้

- CDXScreen ใช้สร้างหน้าจอเกม
- CDXSurface ควบคุม Surface

- CDXLayer ใช้ควบคุม Surface สำหรับทำภาพพื้นหลัง
 - CDXSprite ควบคุมตัวละครแบบสไปรท์ในเกม
 - CDXInput รับค่าจากผู้ใช้ผ่านทางคีย์บอร์ด เมาส์ และจอยสติ๊ก

- CDXSound เก็บไฟล์เสียง .wav
- CDXSoundBuffer เล่นไฟล์เสียง .wav
- CDXMidi เล่นไฟล์เสียง MIDI (.mid)

หากเปรียบเทียบกับส่วนต่าง ๆ ใน DirectX ก็จะได้ดังนี้

- DirectDraw คือ CDXScreen, CDXSprite, CDXSurface
- DirectInput คือชุดคำสั่ง CDXInput
- DirectSound คือ CDXSound และ CDXSoundBuffer
- DirectMusic คือ CDXMidi

ตัวอย่างการเรียกใช้งาน DirectX และ CDX การคืนหน่วยความจำ

DirectX

```
Case WM_DESTROY :
{
    primary->Release();
    primary = NULL;
    dd->Release();
    dd=NULL;
    PostQuitMessage(0);
}break;
```

CDX

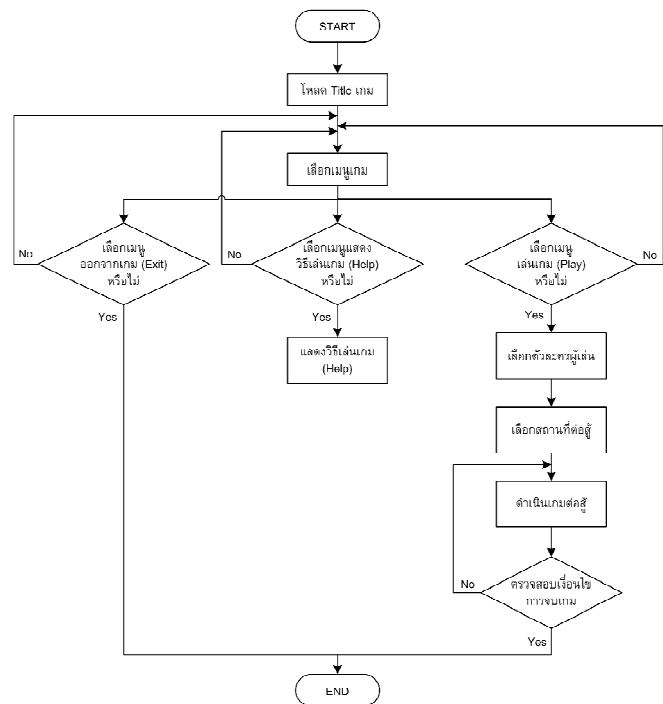
```
void cdx_DelInit (void)
{
    SAFEDELETE( Input );
    SAFEDELETE( Screen);
}
```

8. การพัฒนาโปรแกรม

เกมที่จะพัฒนาชื่อว่า ความมืดในเวทย์มนตร์ (Dark of Magic: NU) ขอบเขตของโปรแกรมที่พัฒนาจึงเป็นเกมในแนวต่อสู้ มีทั้งภาพและเสียงครบสมบูรณ์ ผู้เล่นสามารถใช้แป้นคีย์บอร์ดเพื่อเล่นเกมและเลือกออฟชั่นต่างๆ ของเกม ภายในเกมจะประกอบไปด้วย ภาพอนิเมชั่นและโลโก้หน้าเกม (Title) เมนูเลือกออฟชั่นต่าง ๆ หน้าจอช่วยเหลือที่อธิบายปุ่มต่าง ๆ ในการเล่นเกม (HELP) ขณะเล่นเกม ผู้เล่นสามารถกดแป้นคีย์บอร์ดเพื่อต่อสู้ ใช้เวทย์มนตร์ หรือยกเลิกการเล่นได้ เมื่อผู้เล่นทำลายศัตรูหรือปีศาจ ระดับพลังงานของปีศาจจะลดลงไปเรื่อย ๆ จนปีศาจสูญสลายไป และในขณะเดียวกัน หากผู้เล่นโดนปีศาจทำร้าย ระดับพลังงานของผู้เล่นก็จะลดลง เกมจะจบลงเมื่อระดับพลังงานของผู้เล่นหรือปีศาจลดลงจนไม่เหลือ และหากปีศาจสูญสลายผู้เล่นจะเป็นฝ่ายชนะ

ผู้วิจัยได้เริ่มศึกษาการเขียนโปรแกรมโดยใช้โปรแกรม Microsoft Visual C++ ร่วมกับ CDX Engine และฟังก์ชันแบบปัญญาประดิษฐ์ (Artificial Intelligence) จากนั้นได้

ออกแบบโครงสร้างและการทำงานของเกมโดยรวม ออกแบบจากการวิเคราะห์ความต้องการของระบบดังภาพที่ 2



ภาพที่ 2 แสดงผังงานการทำงานของเกม

9. การออกแบบรูปแบบตัวละครและฉาก

ตัวละครในเกมมี 2 ฝ่ายคือ ฝ่ายผู้เล่นกับศัตรู ตัวละครผู้เล่น (Hero) จะเป็นตัวแทนผู้เล่นของเกม ผู้เล่นจะต้องเลือกเพื่อไปต่อสู้กับศัตรู ในเกมจะมีให้เลือก 4 ตัวละครคือ อัศวิน นักบวชเวทย์ นินจาผู้ลอบสังหาร แม่มดเวทย์ แต่ขอยกตัวอย่างการออกแบบอัศวิน ดังนี้

อาวุธประจำกายของอัศวินคือ ดาบอัศวินลักษณะเด่นคือ มีระดับพลัง (HP) สูง มีเวทย์ดาบอาคม

เวทย์มนตร์คือใช้อาวุธประจำกาย ได้เร็วขึ้น ภายใน 10 วินาที (เสียค่า SP ครั้งละ 20) และพลังโจมตีเพิ่มขึ้น 2 เท่า (เสียค่า SP ครั้งละ 20)

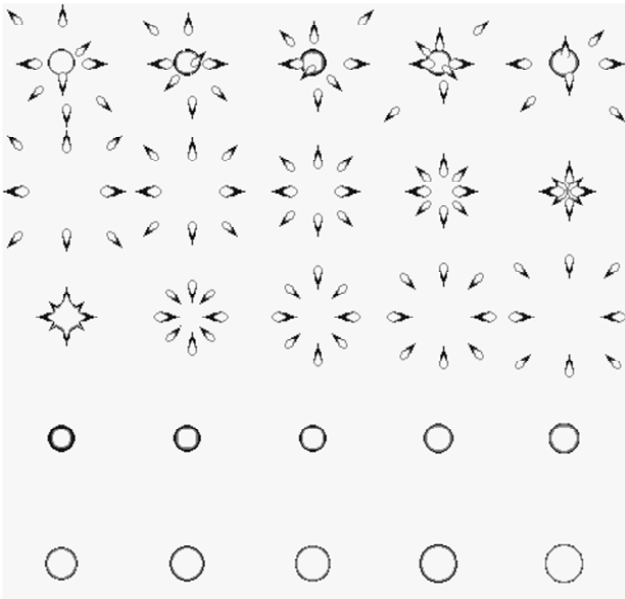


ภาพที่ 3 แสดงการออกแบบตัวละคร

จะเห็นว่า จะต้องออกแบบท่าทางผู้เล่นให้ละเอียด ตั้งแต่ การเดิน การใช้อาวุธ การใช้เวทย์มนตร์ การโดนโจมตี หรือแม้



กระทั่งท่าทางการตายของผู้เล่นเพื่อนำไปใช้แสดงผลภาพ Graphic ในการ code โปรแกรมต่อไป



ภาพที่ 4 แสดงการออกแบบเวทมนตร์ของอัศวิน

ศัตรู (Enemy) จะเป็นคู่ต่อสู้กับผู้เล่น สามารถใช้เวทมนตร์ได้ไม่จำกัด และร้ายแรงขึ้นตามระดับพลังชีวิต (HP) ที่ลดลง สามารถโจมตีผู้เล่นได้ทุกตำแหน่งและแม่นยำมาก สมุนของศัตรู คือ ค้างคาวดูดเลือด เวทมนตร์ของศัตรู คือ ปล่อยลูกสมุน (ค้างคาวดูดเลือด) พลังวงแสงม่วง วนติดตามตัว พลังมนต์ดำวิ่งโจมตี และพลังพายุน้ำแข็งวน

10. การควบคุมตัวละครแบบสไปรท์และการออกแบบตัวละคร

CDXSprite ควบคุมตัวละครแบบสไปรท์ในเกม โดยสัมพันธ์กับภาพที่ใช้ในโปรแกรม กล่าวคือ ภาพที่ใช้ต้องออกแบบขนาดของภาพย่อย ให้พอดีกับขนาดที่กำหนดใน Syntax นั้น โดยเทคนิคนี้ นิยมใช้ในการแสดงผลตัวละครต่างๆภายในเกมประกอบไปด้วย Syntax ดังนี้

new CDXSprite (Screen “ที่อยู่ของไฟล์ภาพ”, ขนาด width, ขนาด High, frame สุดท้าย, จุดเริ่มต้น)

จากโปรแกรมอธิบายได้ว่า มีการ Sprite ภาพที่กำหนด โดยแต่ละภาพมีขนาดความกว้าง 128 พิกเซลและขนาดความสูง 128 พิกเซล ซึ่งจะเก็บอยู่ในภาพเดียวกันถูกโหลดเก็บไว้ตอนเปิดโปรแกรมครั้งแรก แล้วจะถูกเรียกใช้แต่ละเฟรมตามคำสั่งที่กำหนด โดยเริ่มการ Sprite ภาพ ตั้งแต่

เฟรมที่ 1 จนถึง frame ที่ 88 (11 แถว x 8 คอลัมน์)

```
switch(Select){
case 0: Player = new CDXSprite(Screen,"resource/1/kni_female.bmp",128,128,88,1);
Player->SetColorKey();
sAttack->Load(WAV,"resource/1/attack_sword.wav",1);
sHit->Load(WAV,"resource/1/hit_sword.wav",1);
sSk11->Load(WAV,"resource/1/knight_twohandquicken.wav",1);
sSk12->Load(WAV,"resource/1/ef_bash.wav",1);
break;
```

11. อัลกอริทึมแบบเอ-สตาร์ (A-Star Algorithm)

ปัญหาประติมากรรมที่ใช้ในเกม [6] คือ อัลกอริทึมแบบ A* เพื่อใช้ในการหาเส้นทางจากจุดเริ่มต้นของศัตรู โดยพิจารณาจากค่าสถิติตำแหน่งการเคลื่อนไหวของผู้เล่น รวมทั้งได้นำการสุ่มมาใช้ร่วมกัน ไม่ว่าจะเป็นการเคลื่อนไหว การเลือกใช้เวทมนตร์เพื่อให้ผู้เล่นเกิดความสับสน ไม่สามารถคาดเดาได้ว่า ศัตรูจะเคลื่อนไหว หรือ โจมตี ในทิศทางใด เลือกใช้อาวุธเวทมนตร์แบบใด ทำให้ศัตรูสามารถโจมตีโต้ตอบกับผู้เล่นได้อัตโนมัติ และให้ผู้เล่นใช้ทักษะในการโจมตีที่ไม่ซ้ำเดิมมากขึ้น

A* (A Star) [6] เป็นอัลกอริทึมที่ใช้ในการหาเส้นทางระหว่างตำแหน่งสองตำแหน่งบนแผนที่ ซึ่งก็มีอัลกอริทึมอื่นๆที่ใช้ในการหาเส้นทางได้เหมือนกัน แต่ A* จะหาเส้นทางที่สั้นที่สุดให้ (ในกรณีที่มีเส้นทางที่สั้นที่สุด) ในเวลาที่รวดเร็ว A* เป็นอัลกอริทึมแบบ Directed นั้นหมายความว่า จะไม่มีการค้นหาแบบไม่รู้ทิศทาง แต่จะมีการกำหนดค่าในการค้นหาเส้นทางที่ดีที่สุดแทน และบางครั้งก็มีการย้อนกลับมาหาเส้นทางเก่าที่หามาแล้ว ซึ่งทำให้การค้นหาเส้นทางโดยใช้ A* มีความยืดหยุ่นสูง

ในการศึกษาเกี่ยวกับ A* นั้นมีสิ่งจำเป็นที่จะต้องทำความเข้าใจดังนี้

Map (หรือ Graph) ซึ่งเป็นพื้นที่ที่ A* ใช้สำหรับการค้นหาเส้นทางระหว่างสองตำแหน่ง อาจจะประกอบด้วยรูปสี่เหลี่ยม หรือ หกเหลี่ยม หรืออาจจะเป็นพื้นที่ในระบบพิกัด 3 มิติ แต่อย่างไรก็ตามก็คือพื้นที่ที่ A* ใช้ในการทำงานนั่นเอง

Nodes คือโครงสร้างที่แทนตำแหน่งใน Map ซึ่งโหนดนี้จะเก็บข้อมูลต่าง ๆ ที่สำคัญสำหรับ A* เช่น ข้อมูลตำแหน่ง ดังนั้นโหนดจะทำหน้าที่เหมือนเป็นที่เก็บความก้าวหน้าของการค้นหา

Distance (หรือ Heuristic) เป็นส่วนที่จะตัดสินใจว่าจะทำการค้นหาไปที่โหนดใด

Cost เป็นค่าปัจจัยอื่น ๆ ที่จะต้องนำมาพิจารณา เช่น เวลา

ค่าใช้จ่าย หรือ ค่าความเปลี่ยนแปลงอื่น ๆ

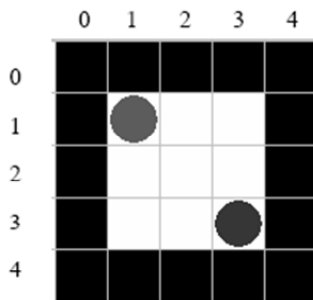
นอกจากนี้ในการค้นหาด้วย A* ยังจะต้องมีค่าคุณลักษณะ 3 อย่างคือ ค่า $g(n)$ $h(n)$ และ $f(n)$ ซึ่งมีรายละเอียดดังนี้

$g(n)$ เป็นค่าที่คิดจากโหนดเริ่มต้น (Start state) ถึงโหนดปัจจุบัน โดยอาจจะมีหลายเส้นทางที่เริ่มจากโหนดเริ่มต้นจนถึงตำแหน่งใน Map ปัจจุบัน

$h(n)$ คือค่าประมาณจากโหนดปัจจุบันถึงโหนดเป้าหมาย (Goal state)

$f(n)$ เป็นค่าผลรวมของ g และ h ซึ่งจะเป็นตัวบอกว่าจะไปโหนดใดต่อไป โดยจะเลือกเอาโหนดที่มีค่า $f(n)$ น้อยที่สุด

นอกจากนี้ A* ยังเก็บสถิติข้อมูลสถานะของโหนดไว้ 2 ประเภท คือ Open list และ Close list โดย Open list ประกอบด้วยโหนดที่ยังไม่ได้ทำการค้นหา ส่วน Close list ประกอบด้วยโหนดที่เคยค้นหาไปแล้ว ดังตัวอย่างต่อไปนี้



ภาพที่ 5 Map ที่ใช้ในการศึกษา A* อัลกอริธึม

กำหนดให้ โหนดเริ่มต้น พิกัด (1,1) โหนดเป้าหมาย พิกัด (3,3)

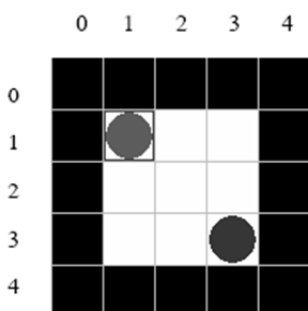
$g(n)$ หาได้จากระยะที่โหนดปัจจุบันห่างจากโหนดเริ่มต้น

$h(n)$ หาได้จาก $h(n) = |dx - sx| + |dy - sy|$ เมื่อ

(dx,dy) เป็นตำแหน่งเป้าหมาย และ

(sx,sy) เป็นตำแหน่งเริ่มต้น

$f(n)$ หาได้จาก $f(n) = g(n) + h(n)$



ภาพที่ 6 ตัวอย่างการหาเส้นทางโดยใช้ A* อัลกอริธึมขั้นที่ 1

เริ่มแรกเก็บโหนด (1,1) เข้า Open list แล้วทำการคำนวณหาค่า $f(1,1)$ จะได้

$$g(1,1) = 0$$

$$h(1,1) = |3 - 1| + |3 - 1| = 2 + 2 = 4$$

$$f(1,1) = 0 + 4 = 4$$

เลือกโหนด (1,1) เป็นโหนดที่ดีที่สุด (มีโหนดเดียว) แล้วดูโหนดต่อไปที่ติดกับโหนด (1,1) นั่นคือ โหนด (2,1) (2,2) และ (1,2) จากนั้นนำเข้า Open list

$$g(2,1) = 1$$

$$h(2,1) = |3 - 2| + |3 - 1| = 3$$

$$f(2,1) = 1 + 3 = 4$$

$$g(2,2) = 1$$

$$h(2,2) = |3 - 2| + |3 - 2| = 2$$

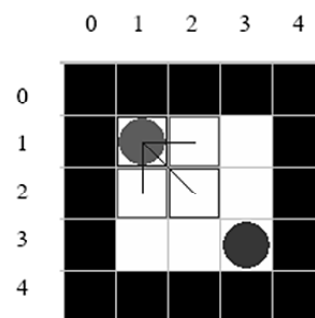
$$f(2,2) = 1 + 2 = 3$$

$$g(1,2) = 1$$

$$h(1,2) = |3 - 1| + |3 - 2| = 3$$

$$f(1,2) = 1 + 3 = 4$$

จากค่า $f(n)$ ใน Open list จะเลือกโหนดที่มีค่า $f(n)$ น้อยสุดมาพิจารณา ในตอนนี้จะต้องเลือกโหนด (2,2) เพราะมีค่า $f(n)$ น้อยที่สุด จากนั้นย้ายโหนด (1,1) จาก Open list เข้า Close list



ภาพที่ 7 ตัวอย่างการหาเส้นทางโดยใช้ A* อัลกอริธึมขั้นที่ 2

แล้วดูโหนดต่อไปที่ติดกับโหนด (2,2) นั่นคือโหนด (3,1) (3,2) (3,3) (2,3) (1,3) และทำการคำนวณหาค่า $f(n)$ แล้วจึงเก็บเข้า Open list

$$g(3,1) = 2$$

$$h(3,1) = |3 - 3| + |3 - 1| = 2$$

$$f(3,1) = 2 + 2 = 4$$

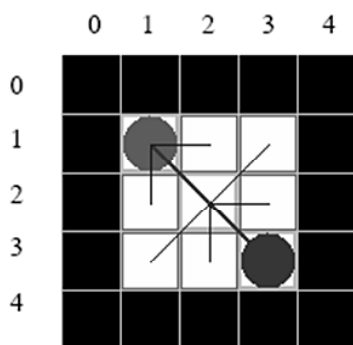
$$g(3,2) = 2$$

$$h(3,2) = |3 - 3| + |3 - 2| = 1$$

$$f(3,2) = 2 + 1 = 3$$

$$\begin{aligned}
 g(3,2) &= 2 \\
 h(3,2) &= |3-3| + |3-2| = 1 \\
 f(3,2) &= 2 + 1 = 3 \\
 g(3,3) &= 2 \\
 h(3,3) &= |3-3| + |3-3| = 0 \\
 f(3,3) &= 2 + 0 = 2 \\
 g(2,3) &= 2 \\
 h(2,3) &= |3-2| + |3-3| = 1 \\
 f(2,3) &= 2 + 1 = 3 \\
 g(1,3) &= 2 \\
 h(1,3) &= |3-1| + |3-3| = 2 \\
 f(1,3) &= 2 + 2 = 4
 \end{aligned}$$

จากค่า $f(n)$ ที่อยู่ใน Open list จะได้ว่าโหนด (3,3) จะถูกเลือกเพราะมีค่า $f(n)$ น้อยที่สุด และโหนด (3,3) ก็เป็นโหนดเป้าหมายด้วย จะได้เส้นทางจากตำแหน่งเริ่มต้นถึงตำแหน่งเป้าหมายตามโหนดดังนี้คือ (1,1) (2,2) และ (3,3)



ภาพที่ 8 ตัวอย่างการหาเส้นทางโดยใช้ A* อัลกอริทึมขั้นที่ 3

12. ผลการวิจัย

จากการพัฒนาโปรแกรมหรือเกมด้วยการใช้ Visual C++ และ CDX Engine (DirectX) ปรากฏว่าการทำงานของโปรแกรมสามารถนำชุดคำสั่ง CDX มาใช้งานร่วมกับโปรแกรม Visual C++ และชุดคำสั่ง DirectX ได้เป็นอย่างดี รวมถึงการออกแบบและพัฒนาโปรแกรมโดยใช้หลักการเขียนโปรแกรมเชิงวัตถุ (OOP) สามารถทำให้ผู้พัฒนาโปรแกรมหาจุดผิดพลาดของระบบและเข้าใจการทำงานของโปรแกรมได้เป็นอย่างดี ไม่ซับซ้อน และสุดท้ายการนำหลักการปัญญาประดิษฐ์ (Artificial Intelligence) มาประยุกต์ใช้กับระบบโจมตีของศัตรู ทำให้เกมมีความหลากหลายมากขึ้น ศัตรูสามารถโจมตีผู้เล่นได้โดยอัตโนมัติ

จากการพัฒนาระบบชุดติดตั้งโปรแกรม (Installation) ทำให้ได้โปรแกรมเกม The Dark OF Magic NU ที่อยู่ในรูปของไฟล์ติดตั้งโปรแกรม (Setup) คือ SETUP_The Dark OF Magic NU.exe มีขนาดไฟล์ประมาณ 58.3 MB ส่วนในตัวโปรแกรมหรือเกม จะอยู่ในโฟลเดอร์ The Dark OF Magic NU โฟลเดอร์จะมีขนาดประมาณ 111 – 200 MB สามารถเรียกใช้โปรแกรมหรือเกมโดยการดับเบิลคลิกไฟล์ The Dark OF Magic NU.EXE ซึ่งอาจเป็น Shortcut อยู่ที่ Desktop คอมพิวเตอร์ หรือหน้าจอคอมพิวเตอร์

การทดสอบเกมใช้วิธีการประเมินด้วยแบบสมการแบบ 5 ระดับ กระทำโดยนิสิตที่ได้จากการทดลองจำนวน 50 คน

13. สรุป

ผลการประเมินด้วยแบบสอบถามจากนิสิตผู้ทดลองใช้จำนวน 50 คน มีระดับการประเมินอยู่ 5 ระดับ โดยแยกเป็นหมวดต่าง ๆ คือ รูปแบบการทำงาน ปุ่มควบคุมการเล่นตัวละคร เสียงและกราฟิกต่าง ๆ ผลคะแนนเฉลี่ยคือ 4.22 ส่วนเบี่ยงเบนมาตรฐาน 0.57 ถือว่าอยู่ในระดับดี

การใช้ CDX Engine เป็นชุดคำสั่งที่เรียกใช้งานในการเขียนเกมทำให้การเขียนโปรแกรมง่ายขึ้น ตอบสนองการทำงานด้านมัลติมีเดียได้สะดวกขึ้น

การนำหลักการปัญญาประดิษฐ์ (Artificial Intelligence) มาประยุกต์ใช้กับการพัฒนาเกมต่อสู้ เช่น ในการโจมตีของศัตรูจะใช้หลักอัลกอริทึมแบบเอ-สตาร์ (A-Star) ในการหาตำแหน่งโจมตี ทำให้เกมมีความหลากหลายมากขึ้น ศัตรูสามารถโจมตีผู้เล่นได้โดยอัตโนมัติ

ข้อเสนอแนะเพิ่มเติมนั้นควรมีการพัฒนาในรูปแบบ เพื่อให้ใช้ในการเสริมความรู้ได้ โดยอาจจะแทรกความเป็นไทยพร้อมทั้งความรู้ต่าง ๆ เพื่อให้ผู้เล่นมีความสนใจและเพลิดเพลินในการเล่นมากขึ้น ควรเพิ่มการพัฒนาแบบให้สามารถเล่นเป็นเกมออนไลน์ได้ และเพิ่มการพัฒนาแบบเกมที่มีการต่อสู้กันหลายระดับ เพื่อให้ผู้เล่นสามารถเล่นเกมได้นานขึ้น

14. เอกสารอ้างอิง

- [1] นิรุช อำนวยศิลป์. เขียนเกมอย่างมืออาชีพด้วย Visual C++ และ DirectX. กรุงเทพฯ : อินโฟเพรส.2545.
- [2] Wikipedia, <http://www.wikipedia.org>
- [3] นิรุช อำนวยศิลป์. Visual C++ & MFC Programming. กรุงเทพฯ : อินโฟเพรส. 2549.

- [4] ฟิรภัทร์สว่างเพียร. เทคนิคการเขียนโปรแกรมและเกมส์ด้วย **Visual C++**. กรุงเทพฯ : ซีเอ็ดดูเคชั่น. 2545. devx.com
- [5] **Expert2You**: http://www.expert2you.com/view_question2.php?q_id=14465 [8] **DirectX in Game by Salitta Nokkhunthong**, http://angsila.compsci.buu.ac.th/~sc450618/css/DirectXInGame_present.pdf
- [6] **Game DevX New Generation**, <http://www.gamedevx.com> [9] อาษา ตั้งสมจิต. **Episode 1 : ทะลวงถึงแก่นเรื่องเขียนเกม**. กรุงเทพฯ : เอ อาร์ อินฟอร์เมชัน แอนด์ พับลิเคชัน. 2547.
- [7] Wikipedia OF Thaigamedev, <http://wiki.thaigame>

