การแก้ปัญหาการเดินของตัวละครในเกมส์คอมพิวเตอร์ประเภทสวมบทบาท กรณีศึกษา : เกมส์ Captain Mandela

นางสาวลภัส ดิษฐ์ภัทรกุล

นางสาวหนึ่งฤทัย ครึกครื้นจิตร

นายสยาม วรรณโกษิตย์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาสถิติประยุกต์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

Solving in movement of computer role-playing game

case study : Catain Mandela Game

Lapas Dispatarakul

Neungluthai Krogkrunjit

Siam Wannakosit

A Special Project Submitted in Partial Fulfillment of the

Requirement for the Degree of Bachelor of science

Department of Applied Statistics

Faculty of Science

King Monkut's Institute of Technology Ladkrabang

Academic Year 2007

หัวข้อปัญหาพิเศษ	การแก้ปัญหาการเดินของตัวละครในเกมส์คอมพิวเตอร์ประเภทสวมบทบาท กรณีศึกษา : เกมส์ Captain Mandela
นักศึกษา	นางสาวลภัส ดิษฐ์ภัทรกุล นางสาวหนึ่งฤทัย ครึกครื้นจิตร นายสยาม วรรณโกษิตย์
ภาควิชา	สถิติประยุกต์ คณะวิทยาศาสตร์
สาขาวิชา	สถิติประยุกต์
อาจารย์ที่ปรึกษา	อาจารย์ บุญญสิทธิ์ วรจันทร์

ภาควิชาสถิติประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

คณะกรรมการตรวจสอบ	ลายมือชื่อ
ประธานกรรมการ อาจารย์ บุญญสิทธิ์ วรจันทร์ กรรมการ ดร. ชานินทร์ ศรีสุวรรณนภา กรรมการ ผศ. สายชล สินสมบูรณ์ทอง	

....

(ผศ.ดร. มนัส ไพฑูรย์เจริญลาภ)

หัวหน้าภาควิชาสถิติประยุกต์

หัวข้อปัญหาพิเศษ	การแก้ปัญหาการเดินของตัวละครในเกมส์คอมพิวเตอร์ ประเภทสวมบทบาท กรณีศึกษา : เกมส์ Captain Mandela	
นักศึกษา	นางสาวลภัส	ดิษฐ์ภัทรกุล
	นางสาวหนึ่งฤทัย	ครึกครื้นจิตร
	นายสยาม	วรรณโกษิตย์
ภาควิชา	สถิติประยุกต์ คณะวิทยาศาสตร์	
สาขาวิชา	สถิติประยุกต์	
ปีการศึกษา	2550	
อาจารย์ที่ปรึกษา	อาจารย์ บุญญสิทธิ์	วรจันทร์

บทคัดย่อ

การวิจัยครั้งนี้มีจุดประสงค์เพื่อแก้ปัญหาการเดินของตัวละครในเกมส์ Catain Mandela โดยประยุกต์ใช้วิธี A* algorithm ซึ่งเป็นหนึ่งในวิธีการหาเส้นทางสั้นที่สุดที่มีประสิทธิภาพในการแก้ปัญหาการเดินของตัวละคร ซึ่งการแก้ปัญหาการเดินทำโดยใช้ภาษา C++ เขียนชุดคำสั่ง (Code) A* algorithm ลงในชุดโปรแกรม Game Engine เพื่อควบคุมการทำงานของ Client ที่จะแสดงผลการเดินและ Server ที่คำนวณการเดิน หลังจากนั้นทำการสำรวจข้อมูลจำนวน 20 คน เพื่อประเมินความพึงพอใจหลังการใช้งานโปรแกรม โดยใช้สถิติ ค่าเฉลี่ย ค่าส่วนเบนเบี่ยงมาตรฐาน สถิติ Shapiro – Wilk , The Wilcoxon Matched-Pairs Signed - Rank Test , The Kruskal-Wallis Oneway ANOVA by Rank Test ในการทดสอบผลการวิจัยพบว่า

จากการพัฒนาโปรแกรมการเดินทำให้การเดินของตัวละครมีประสิทธิภาพมากขึ้น และจากการประเมินความพึงพอใจพบว่า กลุ่มตัวอย่างมีความพึงพอใจโดยเฉลี่ยหลังการพัฒนาเกมส์มากกว่าก่อนการพัฒนาอย่างมีนัยสำคัญ

Special Project Title	Solving in movement of computer role-playing game case study : Catain Mandela Game	
Name	Lapas	Dispatarakul
	Neungluthai	Krogkrunjit
	Siam	Wannakosit
Department	Applied Statistics	
Program	Applied Statistics	
Academic Year	2007	
Special Project Advisor	Bunyasit	Vorrajun

ABSTRACT

The object of the research is to improve walking in Catain Mandela Game . A* algorithm , some of more effective Shortest – path algorithms , was apptied to solve walking . C++ was also used to write A* algorithm code in Game Engine software to control both Client in displayins walking on server and Server in calculating walking way . 20 sample were selected to assess the improvement resectl $\bar{x}$, SD , Shapiro – Wilk statistics , The wilcoxon Matched-Pairs Signed - Rank Test , The Kruskal-Wallis Oneway ANOVA by Rank Test were used to analysedata . The results were as follows .

Developing walking in Game made walking more efficient . From analysing data on attitude , It was found that sample had more average satisfied attitude . After Game development them blfore development significantty

กิตติกรรมประกาศ

ปัญหาพิเศษฉบับนี้สำเร็จลุล่วงได้ด้วยดี และมีความถูกต้องในเนื้อหา ก็เนื่องด้วยความอนุเคราะห์จากคณะอาจารย์ และบุคคลที่เกี่ยวข้อง

ขอขอบพระคุณอาจารย์ บุญญสิทธิ์ วรจันทร์ อาจารย์ที่ปรึกษาในการทำปัญหาพิเศษฉบับนี้ ซึ่งกรุณาให้ความรู้ คำแนะนำ คำปรึกษา และตรวจทานความถูกต้องในการทำปัญหาพิเศษจนกระทั่งเสร็จสมบูรณ์

ขอกราบขอบพระคุณ ดร. ชานินทร์ ศรีสุวรรณนภา และ ผศ. สายชล สินสมบูรณ์ทอง ที่ได้ให้ความรู้และคำแนะนำในการทำปัญหาพิเศษฉบับนี้

ขอขอบพระคุณคณาจารย์ภาควิชาสถิติประยุกต์ทุกท่าน ที่ได้ประสิทธิ์ประสาทความรู้

ขอขอบพระคุณ เจ้าหน้าที่ของภาควิชาสถิติประยุกต์ ที่ประสานงานและอำนวยความสะดวกให้แก่ผู้จัดทำปัญหาพิเศษตลอดการดำเนินงาน

ขอขอบคุณ เพื่อน ๆ น้อง ๆ ที่ให้คำปรึกษาและให้ความร่วมมือในการตอบแบบประเมินด้วยความตั้งใจ

สุดท้ายนี้ ขอขอบพระคุณผู้ที่เกี่ยวข้องกับความสำเร็จของปัญหาพิเศษฉบับนี้ ซึ่งไม่ได้กล่าวนามไว้ทุกท่าน

นางสาวลภัส ดิษฐ์ภัทรกุล
นางสาวหนึ่งฤทัย ครึกครื้นจิตร
นายสยาม วรรณโกษิตย์

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ในช่วง 2-3 ปีที่ผ่านมา คงเป็นที่ทราบกันดีว่าอุตสาหกรรมเกมส์เติบโตขึ้นมาก และกำลังแพร่หลายเป็นที่นิยมในหมู่เยาวชนไทยในขณะนี้ ซึ่งส่วนใหญ่มักจะเป็นกลุ่มเด็กและวัยรุ่นเพศชาย รวมถึงผู้ใหญ่ที่เล่นเกมส์มาก่อนในวัยเด็ก ดังนั้นคงไม่แปลกถ้าจะพบว่าบรรดาร้านเกมส์ต่าง ๆ ผู้ที่มาใช้บริการจะเป็นบรรดาเด็ก ตั้งแต่ประถมศึกษาเรื่อยมาจนถึงมหาวิทยาลัย หรือบางทีอาจเป็นกลุ่มในวัยทำงานบางกลุ่มได้เลยทีเดียว ธุรกิจเกมส์จึงกลายเป็นธุรกิจที่กำลังได้รับความนิยมเป็นอย่างมากในปัจจุบัน และแน่นอนที่สุดว่าเกมส์ส่วนใหญ่ที่กำลังได้รับความนิยมนั้นเป็นเกมส์ที่ไม่ใช่เกมส์เชิงเนื้อหาสาระความรู้หรือเกมส์เชิงวิชาการ

มีคำถามที่น่าสนใจว่า ทำไมเกมส์ทางด้านวิชาการจึงไม่ได้รับความนิยมเท่าที่ควรหรือขาดอะไรที่ทำให้น่าสนใจไป อาจจะทั้งในด้านรูปภาพประกอบ การเขียนโปรแกรม หรืออาจจะเป็นสีสัน ความคมชัดรวมถึงภาพที่สวยงาม สิ่งจูงใจเหล่านี้เกมส์วิชาการก็ล้วนมีครบถ้วนสมบูรณ์ บางเกมส์อาจจะมีมากกว่าด้วยซ้ำไป หรือจะเป็นในด้านของอุปกรณ์เสริมในการเล่นเกมส์ ก็มีครบถ้วนสมบูรณ์ คงมีอยู่สิ่งเดียวที่เกมส์ส่งเสริมทางด้านวิชาการไม่มี แต่เกมส์ที่ไม่ใช่เชิงวิชาการมี และเป็นสิ่งที่ดึงดูดใจผู้เล่นเป็นอย่างมากก็คือ ระบบการเล่น หรือเนื้อเรื่องที่น่าสนใจ น่าติดตาม เพราะที่ผ่าน ๆ มาเกมส์วิชาการเกมส์ที่มีเนื้อเรื่องที่น่าสนใจ น่าติดตาม มีอยู่น้อยมาก

สำหรับผู้ที่ต้องการสร้างเกมส์วิชาการที่จะคิดค้นเกมส์ใหม่ ๆ ที่มีทั้งระบบการเล่นที่น่าสนใจ หรือเนื้อเรื่องที่ดึงดูดใจผู้เล่นเกมส์เป็นเรื่องที่ยากเพราะปัญหาอยู่ที่การเพิ่มเติมสาระความรู้เข้าไปในตัวเกมส์ แต่ก็คงไม่ใช่ว่าจะหมดหนทางกันเสียทีเดียว อย่างน้อยในขณะนี้ได้มีการพัฒนาระบบเกมส์วิชาการที่มีทั้งรูปภาพประกอบที่สวยงาม รวมถึงเนื้อหาที่น่าสนใจ และเล่นได้สนุกในระดับหนึ่งเตรียมออกมาสู่ตลาด โดยเนื้อหาของเกมส์แต่ละเกมส์ได้ผ่านการตรวจสอบอย่างละเอียดจากผู้เชี่ยวชาญด้านเกมส์ เพื่อไม่ก่อให้เกิดผลเสียหายทั้งในด้านศีลธรรมและจริยธรรมต่อเยาวชนไทย

อุตสาหกรรมเกมส์ในบ้านเรายังไม่ได้รับการสนับสนุนจากทางภาครัฐบาลเท่าที่ควรจะเป็นเมื่อเทียบกับภาคอุตสาหกรรมอื่น ๆ และดูเหมือนจะเพิ่งเริ่มตื่นตัวเมื่อไม่กี่ปีที่ผ่านมา อีกทั้งนโยบายบางอย่างก็ยังขัดแย้งกันเองอยู่ในตัว ทำให้ความก้าวหน้าของอุตสาหกรรมเกมส์จำกัดอยู่ในวงแคบ ๆ คือผู้เล่นเกมส์นั่นเอง เพราะสำหรับสังคมในปัจจุบันนั้น เกมส์ก็คือเรื่องไร้สาระที่เป็นตัวบ่อนทำลายเยาวชนในสายตาของผู้ใหญ่บ้านเราไม่เปลี่ยนแปลง

ด้วยเหตุนี้ทำให้ในส่วนของภาคเอกชน การลงทุนให้กับอุตสาหกรรมเกมส์มีแนวโน้มที่เสี่ยงเกินไป แม้ว่าค่าใช้จ่ายในการสร้างเกมส์จะน้อยกว่า ถ้าเทียบกับประเทศอื่น ๆ ก็ตาม และเป็นที่น่ายินดีว่าในปัจจุบันมีสมาคมอุตสาหกรรมซอฟแวร์เกมส์ไทย หรือ TGA (Thailand Game Software Association) ซึ่งก่อตั้งจากความร่วมมือของบริษัทผู้ผลิตและจัดจำหน่ายซอฟแวร์เกมส์ไทย ภายใต้การสนับสนุนของสำนักงานส่งเสริมอุตสาหกรรมซอฟแวร์แห่งชาติ (องค์การมหาชน) หรือ SIPA ซึ่งเป็นหน่วยงานสังกัดกระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร ได้กล่าวไว้ว่า "ในอนาคตทางสมาคมจะมีเกมส์ใหม่ ๆ เข้ามาเพิ่มเติมตลอด และจะมีกิจกรรมดี ๆ เพื่อส่งเสริมอุตสาหกรรมเกมส์ไทยและสังคมไทย ซึ่งตรงกับวัตถุประสงค์ของสมาคมที่ต้องส่งเสริมอุตสาหกรรมเกมส์ในประเทศไทยทั้งทางด้านผู้พัฒนา ผู้เผยแพร่ ผู้จัดจำหน่าย และสื่อมวลชน" วัตถุประสงค์หลักของสมาคมคือการสนับสนุนอุตสาหกรรมเกมส์ของประเทศไทย ทั้งในแง่ผู้ผลิต ผู้จัดจำหน่าย และผู้มีส่วนเกี่ยวข้องในอุตสาหกรรมนี้ เพื่อการขยายตัวของอุตสาหกรรมและภาพลักษณ์อันดีของอุตสาหกรรมซอฟแวร์เกมส์ไทย

ปัญหาพิเศษครั้งนี้ผู้วิจัยจึงมีความคิดที่จะพัฒนาเกมส์ในด้านวิชาการ ที่นอกจากจะให้ความรู้ และยังสร้างความเพลิดเพลินแก่ผู้เล่นไปด้วยในตัว ทั้งนี้ทางคณะผู้จัดทำ จึงนำเกมส์ Captain Mandela มาพัฒนาเพิ่มเติมให้ดีขึ้นกว่าเดิม โดยเนื้อหาภายในเกมส์นอกจากจะมีกราฟฟิคที่น่าสนใจ เพื่อความเพลินเพลินแล้วยังสอดแทรกเนื้อหาเกี่ยวกับโรคเอดส์เข้าไปด้วย เพราะเป็นที่ทราบกันดีว่าโรคเอดส์กำลังสร้างปัญหาให้กับโลกของเราเป็นอย่างมาก เกมส์นี้จะทำให้ผู้เล่นทราบถึงพิษภัยของโรคเอดส์ได้มากยิ่งขึ้น อาทิเช่น เอดส์คืออะไร เอดส์สามารถติดต่อกันได้อย่างไร เอดส์มีอาการอย่างไร ป้องกันตัวอย่างไรไม่ให้ติดเอดส์ เอดส์รักษาได้หรือไม่ ใครบ้างที่ควรตรวจหาเชื้อเอดส์ ฯลฯ โดยเนื้อหาความรู้เหล่านี้จะสอดแทรกเข้าไปในตัวเกมส์ ซึ่งนอกจากผู้เล่นจะได้รับความเพลิดเพลินจากการเล่นเกมส์แล้ว ยังได้เนื้อหาสาระความรู้เกี่ยวกับโรคนี้เพิ่มขึ้นอีกด้วย

อย่างไรก็ตามในเกมส์จะพบปัญหาการเดินของตัวละคร ซึ่งโดยธรรมชาติแล้วการเดินของตัวละครในเกมส์นั้น จะเดินเป็นเส้นตรงทำให้ตัวละครในเกมส์นั้นจะเดินไปชนและติดที่วัตถุต่างๆ ที่ขวางเส้นทางการเดินของตัวละคร จากปัญหาข้างต้นนั้นสามารถที่จะแก้ไขได้โดยการประยุกต์การหาเส้นทางสั้นที่สุดโดยใช้ A-Star Search Algorithm เข้าไปพัฒนา ในส่วนของ Game Engine ซึ่งจะใช้ภาษา C++ ในการพัฒนา โดยจุดหลักที่เรามุ่งเน้นคือการประยุกต์ใช้ Dynamic Programming และ Branch and Bound ของการวิจัยการดำเนินงาน เพื่อทำการพัฒนา Artificial Intelligence (AI) ของ Game Engine โดยทำให้วัตถุที่สามารถเคลื่อนที่ในเกมส์ได้นั้นเคลื่อนที่ด้วย Artificial Intelligence ที่จะสามารถหลบหลีกสิ่งกีดขวางต่าง ๆ ได้ง่ายขึ้น

1.2 วัตถุประสงค์

วัตถุประสงค์ในการศึกษาครั้งนี้คือเพื่อนำความรู้เรื่องการหาเส้นทางที่สั้นที่สุดมาพัฒนาการเดินของตัวละครในเกมส์ ภายใต้การคำนวณการเดิน และคำนวณหาเส้นทางการเดินที่เหมาะสมที่สุดด้วย A* Algorithm รวมถึงพัฒนาโปรแกรมประยุกต์ Game Engine ให้มีความสมบูรณ์ทางด้านการเดินของตัวละครมากยิ่งขึ้น ทำให้ผู้เล่นได้สนุกเพลิดเพลินกับการเล่นเกมส์ และได้ความรู้เกี่ยวกับโรคเอดส์ ทราบที่มา พิษภัย รวมถึงการป้องกันที่นำเสนอสอดแทรกในตัวเกมส์ ทำให้ง่ายต่อการจดจำ

1.3 สมมติฐานของการวิจัย

1. ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
2. รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
3. ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
4. ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
5. ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
6. ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้ (เกี่ยวกับโรคเอดส์) ได้มากขึ้น ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
7. ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน ,การโจมตีในเกมส์มากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
8. ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
9. ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
10. รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
11. ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

12. ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
13. ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
14. ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
15. ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตีในเกมส์มากน้อยเพียงใด ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
16. ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ
17. มัธยฐานความพึงพอใจในหัวข้อ ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน
18. มัธยฐานความพึงพอใจในหัวข้อ รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน
19. มัธยฐานความพึงพอใจในหัวข้อ ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน
20. มัธยฐานความพึงพอใจในหัวข้อ ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน
21. มัธยฐานความพึงพอใจในหัวข้อ ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน
22. มัธยฐานความพึงพอใจในหัวข้อ ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์) ได้มากขึ้น ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน
23. มัธยฐานความพึงพอใจในหัวข้อ ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตีในเกมส์มากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน
24. มัธยฐานความพึงพอใจในหัวข้อ ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน
25. มัธยฐานของระดับการศึกษา 4 กลุ่มทั้งก่อนและหลังการปรับปรุเกมส์แตกต่างกัน

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำความรู้การวิจัยการดำเนินงานไปประยุกต์และต่อยอดความรู้ทาง Dynamic Programming และ Branch and Bound ให้นำไปใช้งานได้จริงบน Game Engine เพื่อหาเส้นทางที่เหมาะสมที่สุดได้
2. เพื่อพัฒนาซอฟต์แวร์ไทย ให้เทียบเท่ามาตรฐานสากล และสามารถแข่งขันในตลาดโลก
3. ตัวเกมส์มีประสิทธิภาพมากขึ้นเพื่อใช้เป็นต้นแบบของเครื่องมือในการเรียนรู้ศึกษาประวัติความเป็นมาของโรคเอดส์ การติดเชื้อ การป้องกัน การใช้ชีวิตอยู่ร่วมกับผู้ป่วยโรคเอดส์
4. มีเกมส์เชิงวิชาการที่เหมาะสำหรับการเผยแพร่ความรู้เกี่ยวกับโรคเอดส์ และรู้จักวิธีการป้องกันอย่างถูกต้อง

1.5 ขอบเขตของการศึกษา

ปัญหาพิเศษนี้ ได้กำหนดขอบเขตของปัญหาไว้พอสังเขป ดังนี้

1. ศึกษาการหาเส้นทางในการเดินที่ใช้ระยะทางที่สั้นและเหมาะสมที่สุด (Optimization) ในเกมส์โดยใช้ A-Star Search Algorithm สำหรับตัวละครหลักเท่านั้น
2. เป็นการคำนวณหาเส้นทางที่เหมาะสมที่จะเกิดขึ้นบนพื้นที่ในเกมส์ที่ตัวละครสามารถเคลื่อนที่ไปได้จากจุดเริ่มต้นจนถึงจุดหมายปลายทาง
3. เพื่อศึกษาการหาเส้นทางการเดินที่สั้นที่สุด (Shortest Path) ของตัวละครภายในเกมส์
4. ใช้ทฤษฎี Shortest Path , Dynamic Programming และ Branch and Bound ในการพัฒนาระบบการเดินของตัวละครในเกมส์
5. พัฒนาระบบการเดินของตัวละครบน Game Engine ด้วยภาษา C++

1.6 คำจำกัดความที่ใช้ในการศึกษา

1. A-Star คือ ชื่อของขั้นตอนวิธีของการค้นหาในทฤษฎีกราฟ ในการศึกษาครั้งนี้หมายถึงการพัฒนาโปรแกรมประยุกต์ Game Engine โดย A-Star Search Algorithm จะทำงานอยู่ในส่วนของ Game Engine โดยมีหน้าที่ควบคุมและวิเคราะห์การเดินของทุก ๆ วัตถุ หรือตัวละครในเกมส์ที่ทำงานบน Game Engine ให้เดินภายใต้การคำนวณแบบ A-Star Search Algorithm

2. Shortest Path คือ เส้นทางที่สั้นที่สุด ในการศึกษาครั้งนี้หมายถึง เส้นทางที่ถูกคำนวณด้วย A-Star Search Algorithm โดยจะพิจารณาที่จุดเริ่มต้น (จุดที่ตัวละครยืนอยู่) และจุดปลายทางที่จะเดินไปจุดนั้น ๆ (จุดที่คลิกด้วย mouse) โดยผลที่จะได้จะเป็นเส้นทางที่สั้นที่สุด

3. Optimization คือ การทำให้เหมาะสมที่สุด ในการศึกษาครั้งนี้หมายถึง การทำให้ระบบการเดินของตัวละครภายในเกมส์เหมาะสมที่สุด ซึ่งทุก ๆ การคำนวณภายใต้ A-Star Search Algorithm จะเป็นเส้นทางที่เหมาะสมที่สุด

4. Pathfinding คือ ระเบียบวิธีการที่ใช้ในขั้นตอนวิธีของการค้นหาแบบทฤษฎีกราฟ ในการศึกษาครั้งนี้หมายถึง วิธีที่ใช้ในการวิเคราะห์เพื่อพัฒนา Game Engine ให้ทำงานได้อย่างถูกต้องและมีประสิทธิภาพสูง

5. Pseudocode (มาจากคำว่า Pseudo และ Code) คือ โค้ดที่อธิบายถึงหลักการและขั้นตอนวิธีของการเขียนโปรแกรม นิยมใช้ในการเขียน Algorithm เพื่อง่ายต่อการศึกษา ในการศึกษาครั้งนี้หมายถึง จะใช้ในการเขียน A-Star Search Algorithm เพื่อนำไปพัฒนาระบบการเดินของตัวละครใน Game Engine ด้วยภาษา C++

6. Role-Playing Game คือ เกมส์ประเภทสวมบทบาท ในการศึกษาครั้งนี้หมายถึง เกมส์ที่ผู้เล่นจะรับบทบาทเป็นผู้ช่วยของกัปตัน Mandela โดยผู้เล่นจะมีภารกิจรณรงค์ต้านโรคเอดส์

7. Game Engine คือ โปรแกรมคอมพิวเตอร์ที่เป็นเครื่องมือในการพัฒนาเกมส์ โดยการทำงานทุกอย่างภายในเกมส์จะถูกควบคุมด้วย Game Engine ทั้งหมด ในการศึกษาครั้งนี้หมายถึง การปรับแต่ง พัฒนาหรือแก้ไข Game Engine จะใช้ภาษาระดับสูงทางคอมพิวเตอร์ ในที่นี้จะใช้ภาษา C++ เข้าไปแก้ไข ตัวอย่างเช่น การติดต่อระหว่าง Hardware และ Software เข้าด้วยกัน การแสดงวัตถุต่าง ๆ ภายในเกมส์ การแสดงภาพ เสียง การเคลื่อนไหวของตัวละครและการเชื่อมต่อผ่านระบบอินเตอร์เน็ต ฯลฯ

8. Artificial Intelligence คือ ปัญญาประดิษฐ์ที่เกิดจากการพัฒนาโปรแกรมคอมพิวเตอร์ให้มีความสามารถในการแก้ปัญหาอย่างชาญฉลาด ในการศึกษาครั้งนี้หมายถึง A-Star Search Algorithm จะใช้พัฒนา Game Engine ให้ตัวละครในเกมส์และวัตถุที่สามารถเคลื่อนที่ได้ทุกชนิดมีระบบการเดินอย่างชาญฉลาดด้วย A-Star Search Algorithm

9. Heuristic คือ ฟังก์ชันทางคณิตศาสตร์ที่ใช้แก้ปัญหาแบบง่าย ๆ โดยอาศัย Algorithm ที่มนุษย์คิดขึ้นมา เช่นวิธีการเดินทางจากบ้านไปที่ทำงาน ตลาด และกลับมาบ้านโดยที่ระยะทางสั้นที่สุด ในการทำปัญหาพิเศษครั้งนี้หมายถึง วิธีการแก้ปัญหาการเดินของตัวละคร เมื่อต้องการหาเส้นทางที่สั้นที่สุด

10. Tie-Breaker คือ ฟังก์ชันทางคณิตศาสตร์ที่ใช้สำหรับแก้ปัญหาร่วมกับฟังก์ชัน Heuristic เพื่อให้มีประสิทธิภาพสูงขึ้นในกรณีที่ต้องการได้คำตอบที่ดีที่สุด ในการทำปัญหาพิเศษครั้งนี้หมายถึง การนำฟังก์ชัน Tie-Breaker มาใช้ร่วมกับฟังก์ชัน Heuristic ในการคำณวนหาเส้นทางที่สั้นที่สุดบนพื้นที่ 2 มิติ ในแนว แกน X และ Y

บทที่ 2
รายงานการวิจัยและทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

การศึกษาปัญหาพิเศษนี้มีวัตถุประสงค์เพื่อจะนำทฤษฎี Dynamic Programming , Branch and Bound , Shorted Path และ Search Algorithm ไปประยุกต์ในการพัฒนา Game Engine ทางด้านระบบการเดินของวัตถุด้วย A* Algorithm ให้มีประสิทธิภาพในการหลบหลีกสิ่งกีดขวางอย่างชาญฉลาดและรวดเร็ว โดยมีขั้นตอนในการศึกษาปัญหาพิเศษดังนี้

1. ศึกษาทฤษฎีที่เกี่ยวข้อง (Dynamic Programming, Branch and Bound, Shorted Path, A* algorithm และ Graph Search algorithm)
2. พัฒนา Game Engine ด้วยภาษา C++
3. เปรียบเทียบความพึงพอใจของผู้เล่นก่อนและหลังการพัฒนาเกมส์

เกมส์ Captain Mandela ได้รับการสนับสนุนจากสถาบันการเงินจากประเทศสหรัฐอเมริกา ซึ่งมุ่งเน้นที่จะพัฒนาและทำประโยชน์เพื่อสังคมส่วนรวม ในด้านการต่อต้านโรคเอดส์

คณะผู้จัดทำได้เล็งเห็นถึงปัญหาที่เยาวชนไทยส่วนใหญ่มีปัจจัยเสี่ยงต่อการป่วยด้วยโรคเอดส์จากการมีเพศสัมพันธ์สูงถึงร้อยละ 84 (83.87) เป็นชายที่อยู่ในวัยเจริญพันธุ์และวัยแรงงาน ร้อยละ 57.75 เป็นชายรักต่างเพศ และร้อยละ 26.12 เป็นหญิงรักต่างเพศ รองลงมาเป็นผู้ใช้ยาเสพติดชนิดฉีด ร้อยละ 4.67 กลุ่มที่ติดเชื้อจากมารดา พบร้อยละ 4.00 กลุ่มรับเลือด ร้อยละ 0.03 กลุ่มที่ไม่ทราบปัจจัยเสี่ยง และอื่น ๆ ร้อยละ 7.44 ตามลำดับ ผู้ป่วยโรคเอดส์ส่วนใหญ่มีฐานะยากจน ประมาณร้อยละ 70 มีการศึกษาน้อย มีรายได้ต่ำ ส่วนหนึ่งประกอบอาชีพการใช้แรงงาน/รับจ้างทั่วไป ลูกจ้างโรงงาน ขับรถรับจ้าง กรรมกร ร้อยละ 46.43 รองลงมาเป็นผู้ประกอบอาชีพเกษตรกรรม ร้อยละ 20.81 และอื่นๆ ประชากรส่วนหนึ่งมีการเคลื่อนย้ายแรงงานภายในประเทศ ซึ่งยังไม่รวมกลุ่มผู้อพยพหรือผู้ใช้แรงงานต่างด้าวที่เข้ามาในประเทศแบบถูกกฎหมายและผิดกฎหมาย ในการประกอบอาชีพผู้ใช้แรงงาน ขายบริการทางเพศ ประมง ฯลฯ

ในบริเวณตามแนวจังหวัดชายแดนหรือจังหวัดที่มีการขยายตัวทางเศรษฐกิจ อุตสาหกรรม การท่องเที่ยวสูง ซึ่งพบว่ามีอัตราการติดเชื้อเฉลี่ย ร้อยละ 0.75 ของประเทศที่ส่งผลกระทบต่อการป้องกันและแก้ไขปัญหาโรคเอดส์ ในด้านการแพร่ระบาดของ HIV และโรคเอดส์ในบางพื้นที่ของประเทศไทย มีเพียงบางจังหวัดที่มีการรายงาน (ข้อมูล http://203.157.15.12/centeraids/indexb.php)

เช่น ประจวบคีรีขันธ์ กาญจนบุรี ปัตตานี สระแก้ว ระนอง และแม่ฮ่องสอนซึ่งยังไม่รวมจังหวัดสำคัญ ๆ อีกหลายจังหวัด โดยเฉพาะไม่ได้มีการดำเนินการเฝ้าระวังและสำรวจเก็บเป็นฐานข้อมูลเพื่อใช้ประโยชน์ในการจัดเตรียมแผนการรองรับบริการและการดูแลรักษาในระบบบริการทางการแพทย์ แผนงานยุทธศาสตร์หรืองบประมาณ รวมทั้งปัญหาในด้านการสื่อสารเพื่อให้เกิดความเข้าใจในด้านการป้องกันและแก้ไขปัญหาโรคเอดส์ ด้วยเหตุจูงใจในด้านเศรษฐกิจมีผลต่อการเคลื่อนย้ายถิ่นฐานของประชากรในวัยแรงงานและวัยเจริญพันธ์อยู่มากที่มีผลต่อพฤติกรรมเสี่ยงต่อการติดเชื้อโรคทางเพศสัมพันธ์หรือโรคเอดส์เพิ่มมากขึ้น

ในช่วงที่ (ปี พ.ศ.2527-2548) อัตราป่วยเอดส์ส่วนใหญ่อยู่ในพื้นที่ภาคเหนือสูงกว่าภาคอื่น ๆ รองลงมาภาคกลาง ภาคใต้ และภาคตะวันออกเฉียงเหนือ เท่ากับ 40.96, 35.61, 32.21 และ18.01 ต่อประชากรในพื้นที่แสนคนตามลำดับ ปัจจุบันพบผู้ป่วยโรคเอดส์รายใหม่ในปี 2550 จำนวน 143 ราย และในปี 2551 พบผู้ติดเชื้อและผู้ป่วยโรคเอดส์แล้ว จำนวน 12 ราย ซึ่งพบมากในเพศชาย ร้อยละ 58.78 เพศหญิง ร้อยละ 41.22 และกลุ่มอายุพบอัตราความชุก การติดเชื้อในกลุ่มอายุ 30-34 ปี สูงที่สุดร้อยละ 28.06 รองลงมาเป็นกลุ่มอายุ 25-29 ปี ร้อยละ 22.09 และ 35-39 ปี ร้อยละ 20.04 ตามลำดับ พบว่าผู้ป่วยส่วนใหญ่ประกอบอาชีพเกษตรกรรม อัตราความชุกการติดเชื้อ ร้อยละ 45.14 รับจ้าง ร้อยละ 25.10 ผู้ว่างงาน ร้อยละ 4.14 และเด็กต่ำกว่าวัยเรียน ร้อยละ 2.10 ตามลำดับ

การรายงานสถานการณ์เอดส์ทั่วโลกล่าสุด www.unaids.org/unaids/06.20e (English Orignial, May 2006) โดย UNAIDS/WHO พบว่าอัตราการติดเชื้อในบางประเทศจะลดลงก็ตาม แต่คาดว่าอัตราการติดเชื้อ HIV และโรคเอดส์ในภาพรวมยังคงเพิ่มขึ้นในทุก ๆ พื้นที่ทั่วโลก ในปี พ.ศ. 2549 มีจำนวนผู้ติดเชื้อและผู้ป่วยโรคเอดส์ทั่วโลกประมาณ 39.5 ล้านคน (34.1-47.1 ล้าน) เป็นผู้ใหญ่ประมาณ 37.2 ล้านคน (32.1-44.5 ล้าน) มีผู้หญิงที่ติดเชื้อ ประมาณ 17.7 ล้านคน (15.1-20.9 ล้าน) เป็นเด็กที่อายุต่ำกว่า 15 ปี ประมาณ 2.3 ล้านคน (1.7-3.5 ล้าน) และเป็นผู้ที่เสียชีวิตจากโรคเอดส์ประมาณ 2.9 ล้านคน (2.5-3.5 ล้าน) ทั้งนี้ คาดว่าจะมีผู้ติดเชื้อรายใหม่เพิ่มขึ้น ประมาณ 4.3 ล้านคน (3.8-6.6 ล้าน) ทั่วโลก โดยคาดว่าจะมีเด็กวัยรุ่นที่อายุ 15-24 ปี มีการติดเชื้อเพิ่มขึ้นกว่าร้อยละ 40 การติดเชื้อส่วนใหญ่พบมากใน กึ่งสะฮารา แอฟริกา (ราชบัณฑิตยสถาน) ประมาณ 24.7 ล้านคน รองลงมาอยู่ในแถบเอเชียใต้และเอเชียตะวันออกเฉียงใต้ ประมาณ 7.8 ล้านคน ส่วนใหญ่ประเทศที่มีการติดเชื้อสูง ร้อยละ 95 จะอยู่ในประเทศที่มีรายได้ต่ำหรือฐานะยากจน และอยู่ในกลุ่มประเทศที่มีรายได้ปานกลาง ส่วนใหญ่การแพร่ระบาดของเชื้อไวรัส HIV ทั่วโลก มีอัตราการติดเชื้อเฉลี่ยรายใหม่ต่อวันมากกว่าวันละ 11,000 คนต่อวัน

ดังนั้นการรณรงค์ต่อต้านโรคเอดส์ จึงเป็นสิ่งที่สำคัญมาก เพื่อที่จะชะลอการติดเชื้อให้น้อยลง สิ่งที่น่าจะมีผลต่อการลดอัตราป่วยของโรคได้คือ การให้ความรู้กับกลุ่มเป้าหมาย การให้ความรู้อย่างหนึ่งก็คือการสร้างเกมส์วิชาการให้ผู้เล่นความสนุกสนานพร้อม ๆ กันด้วยระหว่างการ

เล่นได้สอดแทรกความรู้เกี่ยวกับโรคเอดส์ โดยนำเสนอพฤติกรรมเสี่ยงต่อการติดเชื้อและวิธีการป้องกันที่ถูกต้อง

จากการทดลองเล่นเกมส์ Captain Mandela พบปัญหาว่า การเดินของตัวละครในเกมส์ Captain Mandela นั้นไม่ไหลลื่นและติดขัดอยู่เสมอ คณะผู้จัดทำจึงได้ทำการค้นคว้าและหาวิธีการแก้ปัญหาการเดินของตัวละคร โดย A* (A Star) เป็นอัลกอริทึมที่ใช้ในการหาเส้นทางระหว่างตำแหน่งสองตำแหน่งบนแผนที่ ซึ่งก็มีอัลกอริทึมอื่น ๆ ที่ใช้ในการหาเส้นทางได้เหมือนกัน แต่ A* จะหาเส้นทางที่สั้นที่สุดให้ (ในกรณีที่มีเส้นทางที่สั้นที่สุด) ในเวลาที่รวดเร็ว A* เป็นอัลกอริทึมที่อาศัยฟังก์ชันทางคณิตศาสตร์ที่เรียกว่า ฮิวริสติก นั่นหมายความว่าจะไม่มีการค้นหาแบบสุ่ม แต่จะมีการกำหนดค่าเพื่อใช้ในการตัดสินใจว่าเส้นทางไหนเป็นเส้นทางที่ดีและสั้นที่สุดแทน และจะมีการคำนวณเส้นทางย้อนกลับเพื่อตรวจสอบว่าเส้นทางที่ได้นั้นเป็นเส้นทางที่ดีและสั้นที่สุดหรือไม่ ซึ่งทำให้การค้นหาเส้นทางโดยใช้ A* มีความยืดหยุ่นสูง

ในการศึกษาเกี่ยวกับ A* นั้นมีสิ่งจำเป็นที่จะต้องทำความเข้าใจดังนี้

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

Nodes คือ โครงสร้างที่แทนตำแหน่งใน Map ซึ่งโหนดนี้จะเก็บข้อมูลต่าง ๆ ที่สำคัญสำหรับ A* เช่น ข้อมูลตำแหน่ง ดังนั้น Node จะทำหน้าที่เหมือนเป็นที่ที่เก็บความก้าวหน้าของการค้นหา

Distance (หรือ Heuristic) เป็นส่วนที่จะตัดสินใจว่าจะทำการค้นหาไปที่ Node ใด

Cost เป็นค่าปัจจัยอื่น ๆ ที่จะต้องนำมาพิจารณา เช่น เวลา ค่าใช้จ่าย หรือค่าความสิ้นเปลืองอื่น ๆ

นอกจากนี้ในการค้นหาด้วย A * ยังจะต้องมีค่าคุณลักษณะ 3 อย่าง คือ ค่า g(n), h(n) และ f(n) ซึ่งมีรายละเอียดดังนี้

g(n) เป็นค่าที่คิดจาก Node เริ่มต้น (Start state) ถึง Node ปัจจุบัน โดยอาจจะมีหลาย เส้นทางที่เริ่มจาก Node เริ่มต้นจนถึงตำแหน่งใน Map ปัจจุบัน แต่ก็จะมี Node นี้เท่านั้นที่แทน เส้นทางที่ได้

h(n) คือค่าประมาณจาก Node ปัจจุบันถึง Node เป้าหมาย (Goal state)

f(n) เป็นค่าผลรวมของ g และ h ซึ่งจะเป็นตัวบอกว่าจะไปที่ Node ใดต่อไป โดยจะเลือกเอา Node ที่มีค่า f(n) น้อยที่สุด

นอกจากนี้ A* ยังเก็บข้อมูลสถานะของ Node ไว้ 2 ประเภท คือ Open list และ Close list โดย Open list ประกอบด้วย Node ที่ยังไม่ได้ทำการค้นหา ส่วน Close list ประกอบด้วย Node ที่เคยค้นหาไปแล้ว

2.1.1 ทฤษฎีกราฟ

ในทางคณิตศาสตร์และทางวิทยาการคอมพิวเตอร์ ทฤษฎีกราฟเป็นการศึกษาเกี่ยวข้องกับกราฟ ซึ่งในโครงสร้างทางคณิตศาสตร์จะใช้ตัวแบบความสัมพันธ์เป็นคู่ ๆ (Pairwise) ระหว่างวัตถุ ซึ่งกราฟในที่นี้หมายถึง โหนด (Node) และเส้นที่เชื่อมต่อระหว่างโหนดเข้าด้วยกัน โดยทั่วไปแล้ว การศึกษาทฤษฎีกราฟจะเน้นหนักที่ปัญหาเส้นทาง ซึ่งสามารถนำไปประยุกต์ใช้ได้และปัญหาเส้นทางก็สามารถจำแนกได้ดังนี้

1. Hamiltonian Path and Cycle Problem
2. Minimum Spanning Tree
3. Route Inspection Problem
4. Seven Bridges of Königsberg
5. Shortest Path Problem
6. Steiner Tree
7. Three Cottage Problem
8. Traveling Salesman Problem

2.1.2 ทฤษฎีเส้นทางที่สั้นที่สุด (Shortest Path Theorem)

ปัญหาเส้นทางที่สั้นที่สุด คือปัญหาสำหรับการหาเส้นทางระหว่างจุดสองจุดโดยจะเลือกผลรวมของค่าน้ำหนักที่มีค่าน้อยที่สุด ซึ่งในบางครั้งจะเรียกการแก้ปัญหานี้ว่า ปัญหาเส้นทางเดียว

วิธีการแก้ปัญหามีหลายวิธี ดังนี้

1. Dijkstra's Algorithm
2. Best-Firat-Search Algorithom
3. A* Search Algorithm
4. Bellman-Ford Algorithm
5. Floyd-Warshall Algorithm

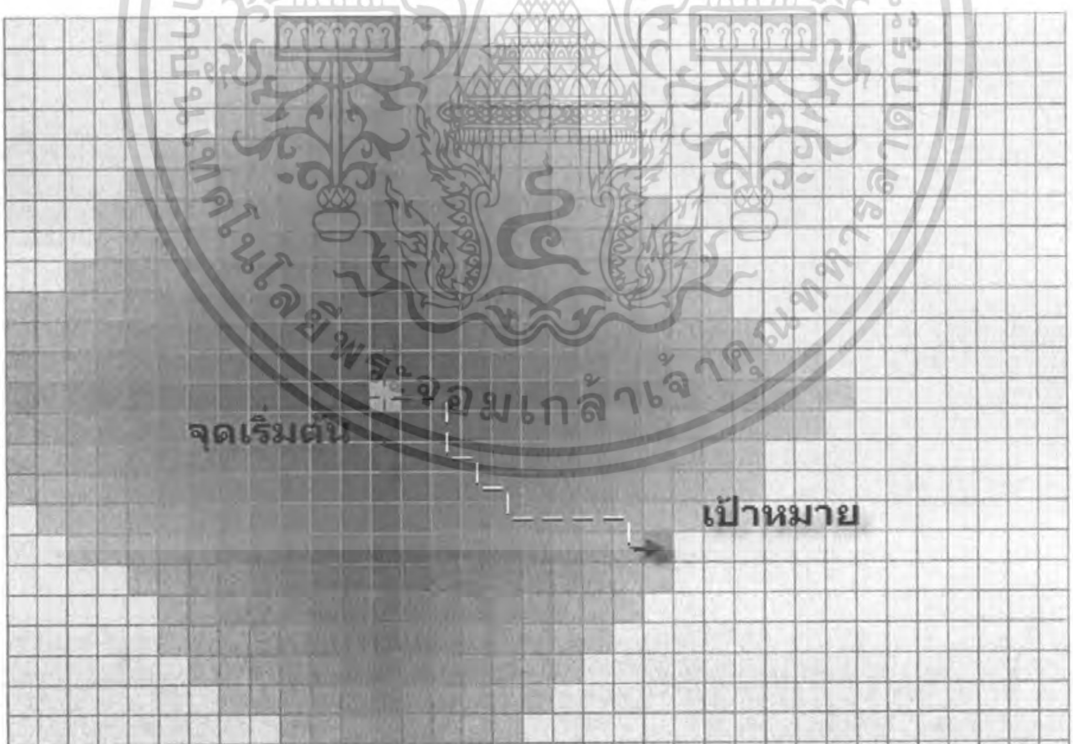
6. Johnson's Algorithm

7. Perturbation Theory

2.1.2.1 Dijkstra's Algorithm

จะทำงานโดยการเข้าไปด้านในของกราฟ โดยให้วัตถุเริ่มที่จุดเริ่มต้น หลังจากนั้นทำการตรวจสอบบริเวณรอบ ๆ ซ้ำไปซ้ำมา และทำการบันทึกจุดที่ได้ทำการเข้าไปถึงด้านในของกราฟแล้วลงไปในรายชื่อกลุ่มที่เข้าถึงกราฟแล้ว ซึ่งจะขยายอาณาเขตออกไปสู่ภายนอกจากจุดเริ่มต้นจนกระทั่งถึงเป้าหมาย โดย Dijkstra's Algorithm เป็นที่ยืนยันแน่ชัดว่าสามารถหาเส้นทางสั้นที่สุดได้จากจุดเริ่มต้นไปสู่เป้าหมาย

จากรูปที่ 2.1 ตารางสี่เหลี่ยมที่มีลูกศรชี้ออกคือจุดเริ่มต้น ส่วนตารางสี่เหลี่ยมที่มีลูกศรชี้เข้าคือเป้าหมาย และพื้นที่ที่แสดงนั้นคือพื้นที่ที่ Dijkstra's Algorithm ได้ค้นหา จะเห็นได้ว่าพื้นที่นั้นขยายออกไปไกลจากจุดเริ่มต้นทั้ง 4 ทิศทาง นั่นคือพื้นที่ที่สำรวจ



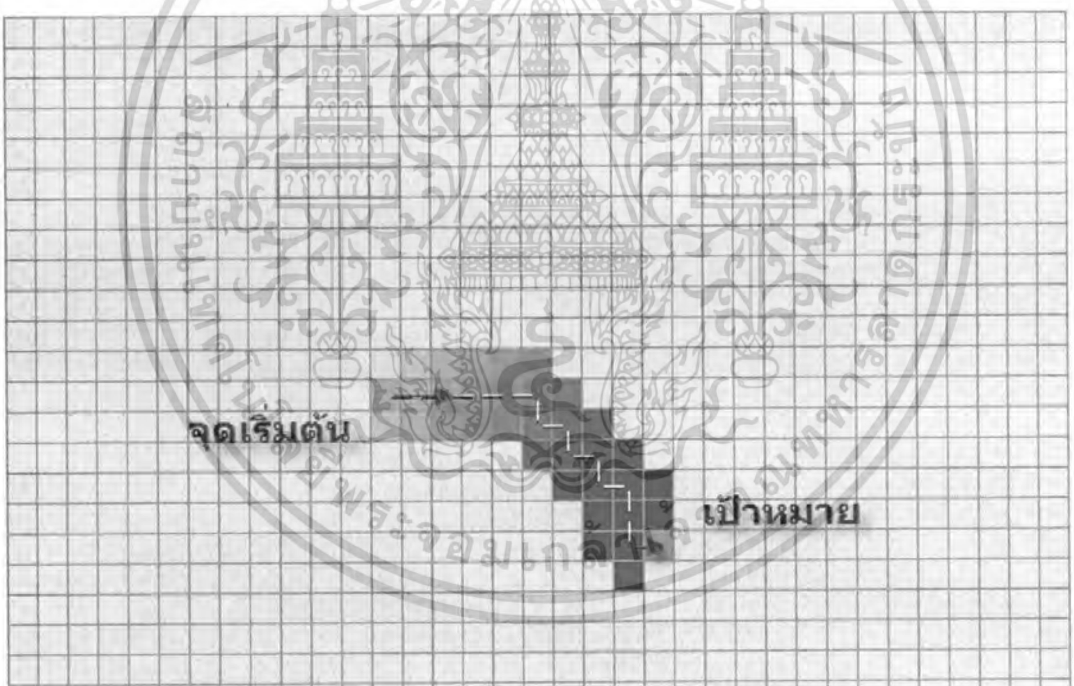
รูปที่ 2.1 การทำงานของ Dijkstra's Algorithm ในกรณีที่ไม่มีสิ่งกีดขวาง

2.1.2.2 Best-First-Search (BFS) Algorithm

จะทำงานในทิศทางที่คล้ายคลึงกันยกเว้น Best-First-Search จะมีการประมาณค่าที่เรียกว่า ฮิวริสติก เป็นระยะทางไกลจากจุดเป้าหมาย แทนที่จะเลือกจุดที่ใกล้ที่สุดไปสู่จุดเริ่มต้น โดย Best-First-Search จะเลือกจุดที่ใกล้ที่สุดไปสู่เป้าหมาย

Best-First-Search ไม่สามารถหาเส้นทางที่สั้นที่สุดได้ แต่อย่างไรก็ตามวิธีนี้คำนวณได้รวดเร็วกว่า Dijkstra's Algorithm เพราะว่าใช้ฟังก์ชัน ฮิวริสติก เพื่อที่จะเป็นแนวทางในการเคลื่อนที่ไปหาเป้าหมายได้อย่างรวดเร็ว

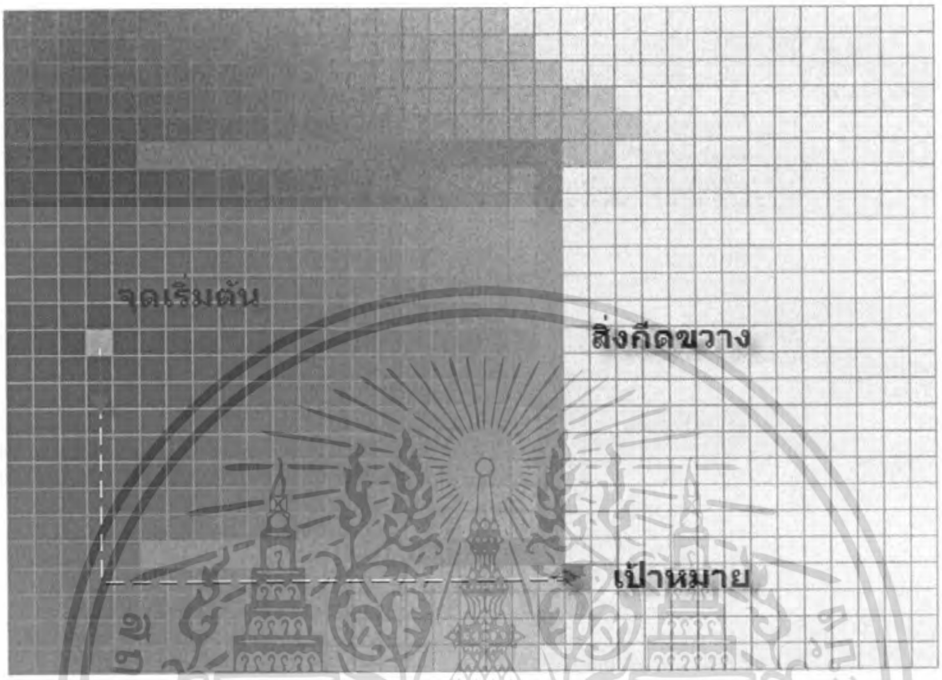
จากรูปที่ 2.2 บริเวณรอบ ๆ ของลูกศรชี้ออกจะเป็นบริเวณที่แสดงถึงโหนดที่มีค่าฮิวริสติกสูง และส่วนบริเวณรอบ ๆ ของลูกศรชี้เข้าจะแสดงถึงโหนดที่มีค่าฮิวริสติกต่ำ แสดงว่า Best-First-Search สามารถหาเส้นทางได้อย่างรวดเร็วเมื่อเปรียบเทียบกับ Dijkstra's Algoritm



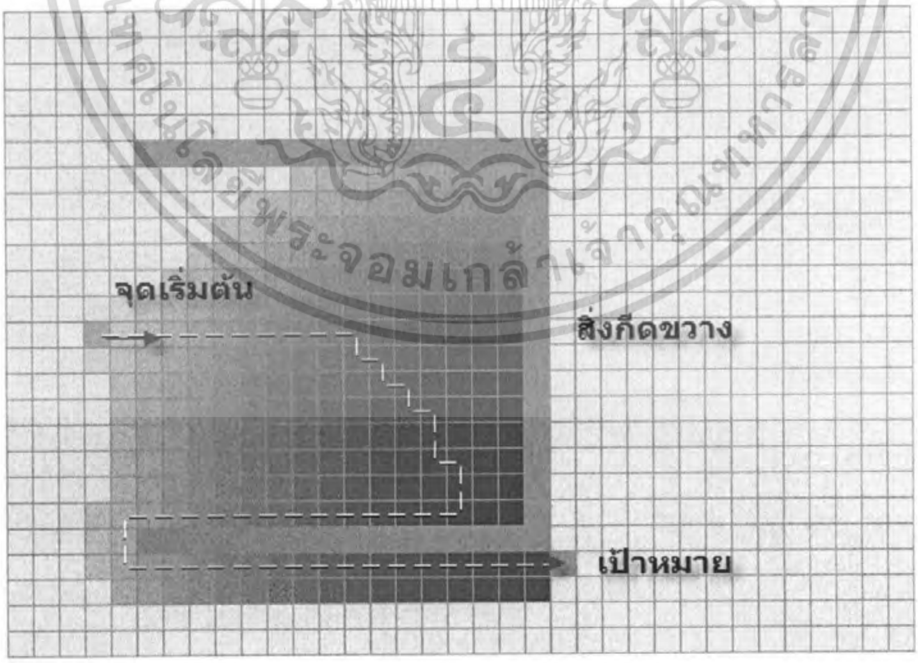
รูปที่ 2.2 การทำงานของ Best-First-Search ในกรณีที่ไม่มีสิ่งกีดขวาง

อย่างไรก็ตาม ตัวอย่างทั้ง 2 นี้ แสดงตัวอย่างแบบง่ายเมื่อแผนที่ไม่มีสิ่งกีดขวางและเส้นทางที่สั้นที่สุดเป็นเส้นตรง

การพิจารณาในกรณีที่มีสิ่งกีดขวาง ซึ่ง Dijkstra's Algorithm จะทำงานมากกว่า แต่สามารถยืนยันได้แน่นอนว่าเป็นเส้นทางที่สั้นที่สุด ส่วน Best-First-Search จะทำงานน้อยกว่า แต่เส้นทางที่ได้ไม่ใช่เส้นทางที่สั้นที่สุด



รูปที่ 2.3 การทำงานของ Dijkstra's Algorithm ในกรณีที่มีสิ่งกีดขวาง



รูปที่ 2.4 การทำงานของ Best-First-Search ในกรณีที่มีสิ่งกีดขวาง

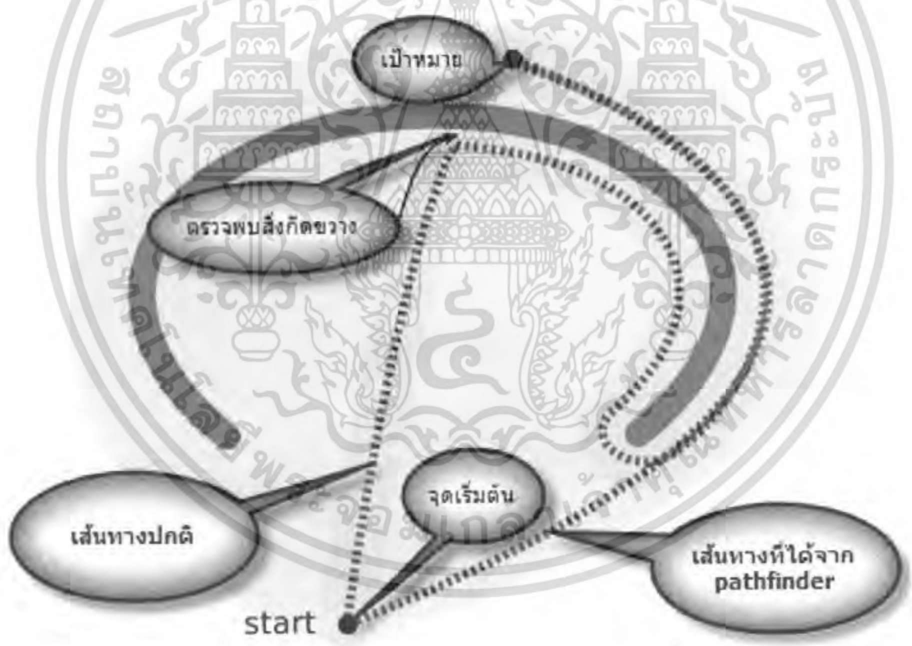
ปัญหาคือ Best-First-Search จะเน้นที่ต้องการจะไปหาเป้าหมายข้างหน้าอย่างเดียว ซึ่งทำให้ผลที่ได้ไม่ใช่เส้นทางที่สั้นที่สุด แต่เป็นเส้นทางที่ค่อนข้างที่จะยาว

วิธีการที่ดี ถ้าจะรวมทั้ง 2 วิธีนี้เข้าด้วยกัน ซึ่ง A* ได้ถูกพัฒนาขึ้นในปี 1968 เพื่อที่จะรวมฮิวริสติก จาก Best-First-Search เข้าสู่ Dijsktra's Algorithm ซึ่งค่อนข้างที่จะดูแตกต่างที่ Best-First-Search ก็ยังคงหาเส้นทางประมาณที่จะแก้ปัญหา โดยไม่สามารถยืนยันได้ว่าจะเป็นเส้นทางที่ดีที่สุด

อย่างไรก็ตาม A* ได้ถูกสร้างบนส่วนที่ดีที่สุดของ Heuristic และ A* สามารถยืนยันได้ว่าเป็นเส้นทางที่สั้นที่สุด

2.1.2.3 A* Search Algorithm

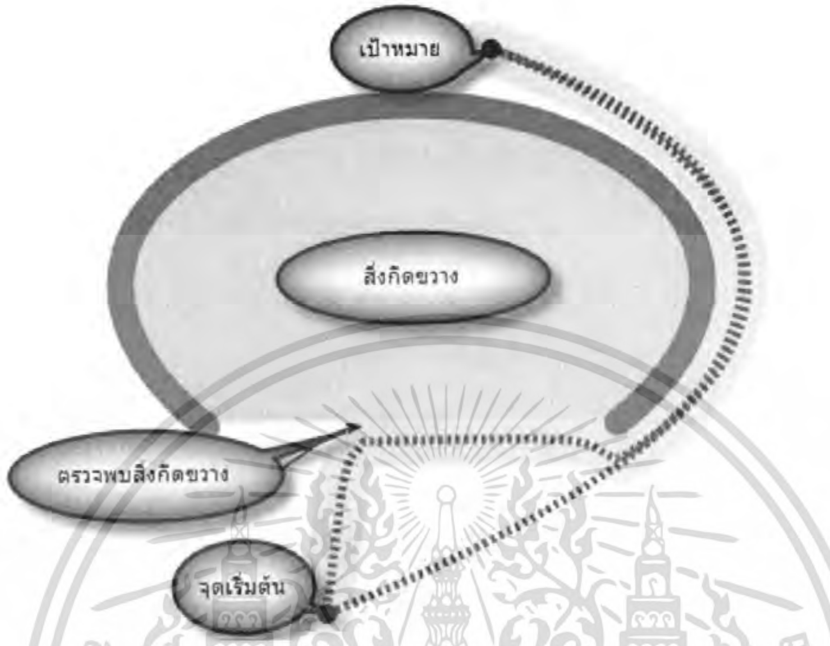
Pathfindr คือ กระบวนการในการหาเส้นทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง ซึ่งเป็นขั้นตอนวิธีที่มีประสิทธิภาพในการหาเส้นทางที่สั้นที่สุดในเส้นกราฟ



รูปที่ 2.5 เส้นทางระหว่างเส้นทางปกติและเส้นทางที่ได้จาก Pathfinder

จากจุดเริ่มต้นจะเดินทางไปยังจุดด้านบนหรือจุดเป้าหมาย (แสดงในพื้นที่ใต้เส้นประ) เพื่อที่จะเดินทางไปยังจุดเป้าหมาย จะเห็นว่าพบสิ่งกีดขวาง (แสดงในเส้นทึบหนา) และทำการเปลี่ยนทิศทาง ซึ่งได้พบทางรอบ ๆ สิ่งกีดขวางเป็นรูปตัว U (แสดงในเส้นประที่เดินอ้อมเส้นทึบ) ในทางตรงกันข้าม Pathfinder จะต้องค้นหาเป็นบริเวณกว้าง (แสดงในพื้นที่ใต้เส้นทั้งหมด) และค้นพบเส้นทางที่สั้นที่สุดคือ เส้นประที่ลากจากจุดเป้าหมายไปถึงจุดเริ่มต้น และไม่ทำการเดินเข้าไป

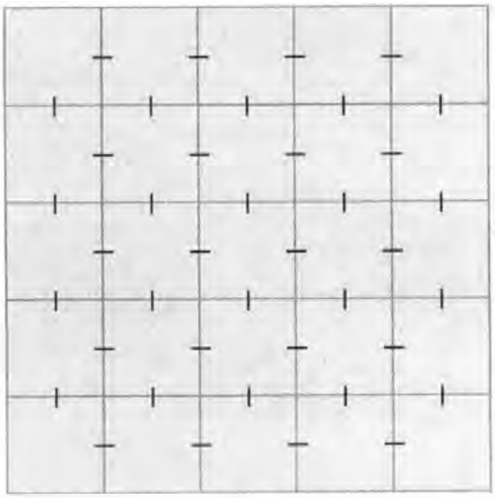
ด้านในของสิ่งกีดขวางรูปตัว U ในกรณีที่จุดเป้าหมายอยู่ในสิ่งกีดขวางรูปตัว U เท่านั้น ถึงจะทำการเดินเข้าไปด้านใน



รูปที่ 2.6 การตรวจสอบสิ่งกีดขวาง

Pathfinder ทำให้มองไปข้างหน้าและสร้างแผนเส้นทางการเดินมากกว่าการที่จะรอจนกระทั่งค้นพบปัญหาของการเดินทาง ในการเดินทางโดยไม่ใช้ Pathfinder สามารถทำงานได้ในหลาย ๆ กรณี และยังสามารถที่จะนำไปประยุกต์ใช้อีกแต่ Pathfinder นั้นเป็นที่ยอมรับว่าเป็นมากกว่าเครื่องมือทั่วไปที่สามารถแก้ไขปัญหาได้อย่างกว้างขวาง

ขั้นตอนวิธีของ Pathfinder Algorithm นั้นมาจากหนังสือทางวิทยาศาสตร์คอมพิวเตอร์ โดยทำงานในรูปแบบกราฟ ซึ่งพื้นที่ในเกมส์สามารถอธิบายได้ด้วยทฤษฎีกราฟ ตัวอย่างเช่น พื้นที่เริ่มต้น พื้นที่รอบ ๆ เป็นต้น

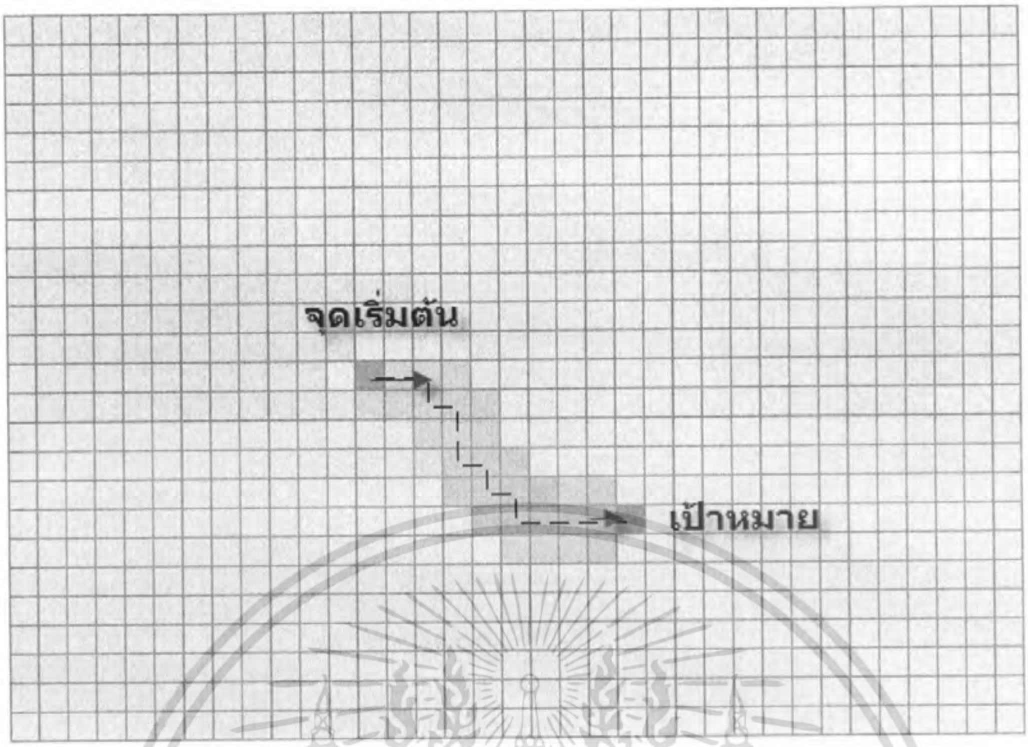


รูปที่ 2.7 ตารางสี่เหลี่ยมจตุรัส 2 มิติ แสดงถึงพื้นที่ที่ใช้คำนวณหาเส้นทางสั้นที่สุด

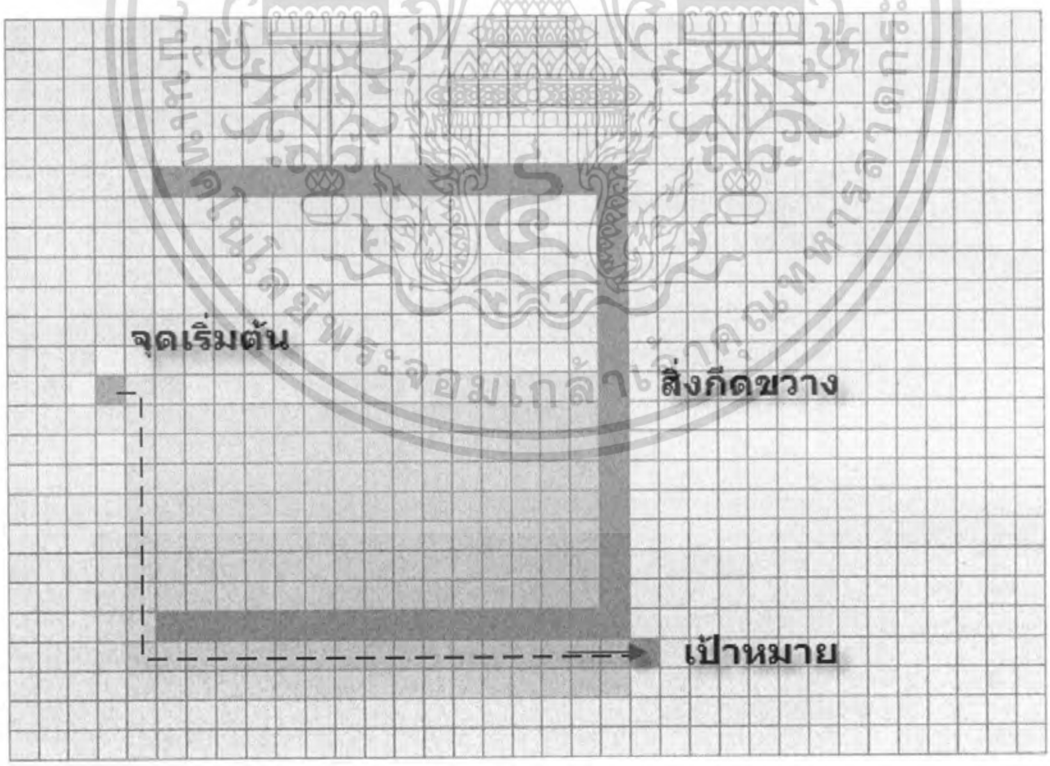
ส่วนใหญ่แล้ว Pathfinder Algorithm จากปัญญาประดิษฐ์ (AI) หรือ การค้นหา Algorithm ได้ถูกออกแบบสำหรับกราฟทั่วไปมากกว่าตารางสี่เหลี่ยมจตุรัสบนพื้นฐานของเกมส์ ซึ่งพยายามที่จะหาข้อได้เปรียบของตารางสี่เหลี่ยมจัตุรัส แต่นั่นก็มีบางสิ่งที่พบและพิจารณาโดยใช้สามัญสำนึก แต่ไม่สามารถคำนวณด้วยวิธีการทางคอมพิวเตอร์ได้

ขั้นตอนวิธีของ A*

A* เป็นทางเลือกที่นิยมใช้สำหรับ Pathfinder เพราะว่ามีความคล่องตัวและสามารถใช้ในพื้นที่กว้างได้ A* ก็เหมือนกับทฤษฎีกราฟทั่วไป สามารถค้นหาได้ในพื้นที่ขนาดใหญ่ และผลลัพธ์ที่ได้คือเส้นทางที่สั้นที่สุด เหมือนกับ Dijkstra's Algorithm และยังสามารถใช้ฟังก์ชัน ฮิวริสติก เพื่อจะนำทางตัวเองไปยังเป้าหมาย ซึ่งรวดเร็วเท่ากับ Best-First-Search นั่นเอง



รูปที่ 2.8 การทำงานของ A*



รูปที่ 2.9 พื้นที่การค้นหาของ A*

จากรูปที่ 2.9 จะพบว่า A* หาเส้นทางได้ดีเหมือนกับ Dijkstra's Algorithm

การนำไปสู่ความสำเร็จคือการรวมข้อมูลต่าง ๆ ที่ Dijkstra's Algorithm และ Best-First-Search ใช้

จากรูปที่ 2.9 บริเวณรอบ ๆ จุดเริ่มต้น แสดงถึงโหนดที่ห่างจากเป้าหมาย และส่วนบริเวณรอบ ๆ จุดเป้าหมาย แสดงถึงโหนดที่ห่างจากจุดเริ่มต้น ซึ่ง A* ปรับสมดุลระหว่าง 2 ทางนี้จากจุดเริ่มต้นไปสู่เป้าหมายซึ่งสามารถอธิบายได้ดังนี้

$$f(n) = g(n) + h(n)$$

Heuristic

Heuristic คือ ฟังก์ชัน h(n) บอกระยะทางกับ A* โดยที่ระยะทางนั้นเป็นระยะทางประมาณของค่าที่น้อยที่สุดจากโหนด n ไปสู่เป้าหมาย ซึ่งเป็นสิ่งสำคัญที่จะเลือกฟังก์ชัน Heuristic ที่ดี

A* ใช้ค่า Heuristic เพื่อการหาเส้นทางสั้นที่สุด

Heuristic สามารถใช้ควบคุม A* ได้โดยใช้ฟังก์ชัน Heuristic มาตัดสินใจในการเลือกเส้นทางที่สั้นที่สุด

- ถ้า h(n) มีค่าเป็น 0 แล้ว A* จะกลับไปทำงานแบบ Dijkstra's Algorithm คือสามารถหาเส้นทางที่สั้นที่สุด
- ถ้า h(n) มีค่าน้อยกว่าค่าของการเดินจากจุด n ไปยังเป้าหมาย แล้ว A* สามารถยืนยันที่จะหาเส้นทางที่สั้นที่สุดได้
- ถ้า h(n) มีค่าที่แม่นยำเท่ากับค่าการเดินทางจาก n ไปสู่เป้าหมายแล้ว A* จะเดินตามเส้นทางที่ดีที่สุด
- ถ้า h(n) มีค่าที่มากกว่าค่าการเดินทางจาก n ไปสู่เป้าหมายแล้ว A* จะไม่สามารถยืนยันที่จะหาเส้นทางที่สั้นที่สุดได้ แต่จะทำงานได้เร็วกว่า
- ถ้า h(n) มีค่าที่มีความสัมพันธ์กับ g(n) มาก แล้ว A* จะกลับไปทำงานแบบ Best-First-Search คือทำงานได้รวดเร็ว

Scale

A* คำนวณโดยใช้สูตร $f(n) = g(n) + h(n)$

การรวม 2 ค่าเข้าด้วยกัน โดยทั้งสองค่านี้จะต้องเป็น Scale เดียวกัน

ถ้าเปรียบเทียบ g(n) เป็นเวลา (ชั่วโมง) และ h(n) เป็นความยาว (เมตร) แล้ว A* กำลังพิจารณา g หรือ h ซึ่งสามารถตัดสินใจได้ว่า จะเลือกทำตามวิธีใดวิธีหนึ่ง ที่จะให้คำตอบที่ดีที่สุด

การพิจารณา Linear exact heuristic

ในภาวะพิเศษ เราสามารถสร้าง Heuristic ได้ โดยปราศจากการคำนวณก่อน โดยไม่ต้องคำนวณล่วงหน้า ถ้ามีแผนที่โครงแบบที่ไม่มีข้อขัดข้องแล้ว เส้นทางที่สั้นที่สุดจากจุดเริ่มต้นไปยังจุดเป้าหมายสมควรเป็นเส้นตรง

ถ้าใช้การคำนวณก่อนแบบง่าย (ไม่ทราบเกี่ยวกับอุปสรรคในโครงแบบ) แล้ว ก็สมควรเทียบเท่า/เหมือนกับการคำนวณก่อนหน้า/ล่วงหน้า

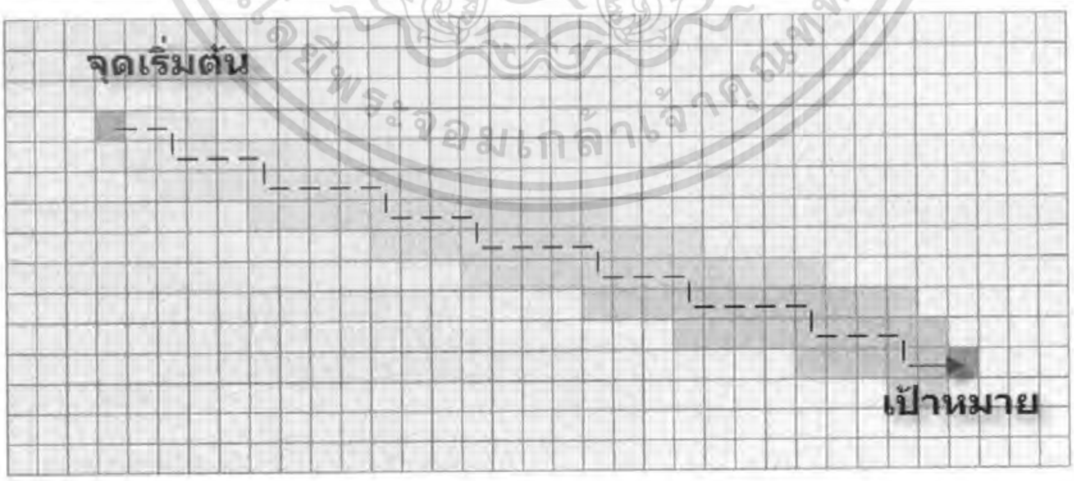
ถ้าไม่เท่ากันแล้ว อาจมีปัญหากับ Scale หรือ ชนิดของการทดลองที่เลือก

การพิจารณาระยะทางระหว่างจุดสองจุด

ค่ามาตรฐานของ Heuristic คือ Manhattan Distance ตรวจดูค่าของฟังก์ชันและหาค่า D ที่ต่ำสุด เพื่อการเคลื่อนย้ายจากที่ว่างอันหนึ่งไปสู่ที่ว่างที่อยู่ใกล้อีกอันหนึ่ง ดังนั้น Heuristic ในเกมส์ควรเป็นช่วงระยะเวลา/อัตราความเร็ว D ของ Manhattan Distance

h(n) = D * (abs(n.x-goal.x) + abs(n.y-goal.y))

โดยที่ abs() เป็นฟังก์ชันทางคอมพิวเตอร์ที่ใช้ปรับให้ค่าในฟังก์ชันมีค่าเป็นบวกเสมอ ควรใช้ Scale ที่เหมาะสมกับค่าฟังก์ชัน



รูปที่ 2.10 การเดินแบบระยะทางระหว่างจุดสองจุด

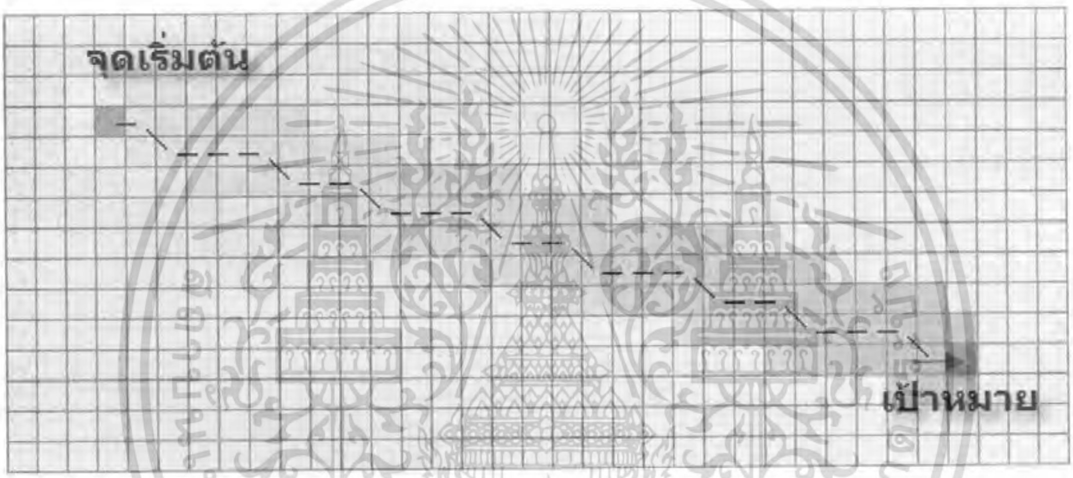
จากรูป 2.10 มีการเพิ่ม Tie-Breaker เข้าไปใน Heuristic

ระยะทางในแนวทะแยงมุม

ถ้าในแผนที่ยอมให้มีการเคลื่อนที่แบบเส้นแยงมุมได้นั้น ต้องใช้ Heuristic อย่างอื่นที่แตกต่างกันออกไป จากระยะทางระหว่างจุดสองจุด สำหรับการเลือก (4 ตะวันออก , 4 เหนือ) นั้นจะเป็น 8*D โดย D หมายถึง ค่าระยะทางที่น้อยที่สุดที่ใช้สำหรับการเคลื่อนที่ไปพื้นที่ว่างข้าง ๆ

อย่างไรก็ตาม สามารถย้ายไปทาง 4 ตะวันออกเฉียงเหนือ ได้อย่างง่าย ๆ ตรงไปตรงมาได้ ดังนั้น Heuristic สมควรเป็น 4*D โดยระบบควบคุมในแนวทแยงมุม ซึ่งเป็นการสมมุติว่าทั้งสองการเคลื่อนที่ของเส้นตรงและเส้นทแยงมุมของค่า D

h(n) = D * max(abs(n.x-goal.x), abs(n.y-goal.y))



รูปที่ 2.11 การเดินในแนวทะแยงมุม

จากรูปที่ 2.11 ถ้าค่าของการเคลื่อนที่ในแนวทะแยงมุมไม่ใช่ D แล้วจะมีค่าเท่ากับ

D2 = $\sqrt{2}$ * D และค่า Heuristic ไม่ใช่ Heuristic ที่ถูกต้องซึ่งเราต้องการบางอย่างที่จะดูซับซ้อนมากขึ้น

h_diagonal(n) = min(abs(n.x-goal.x), abs(n.y-goal.y))

h_straight(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h(n) = D2 * h_diagonal(n) + D * (h_straight(n) - 2*h_diagonal(n))

ซึ่งเป็นวิธีการคำนวณค่า

h_diagonal(n) = จำนวนตัวเลขที่จะนำเราไปตามแนวทแยงมุม

h_straight(n) = ค่า Manhattan distance

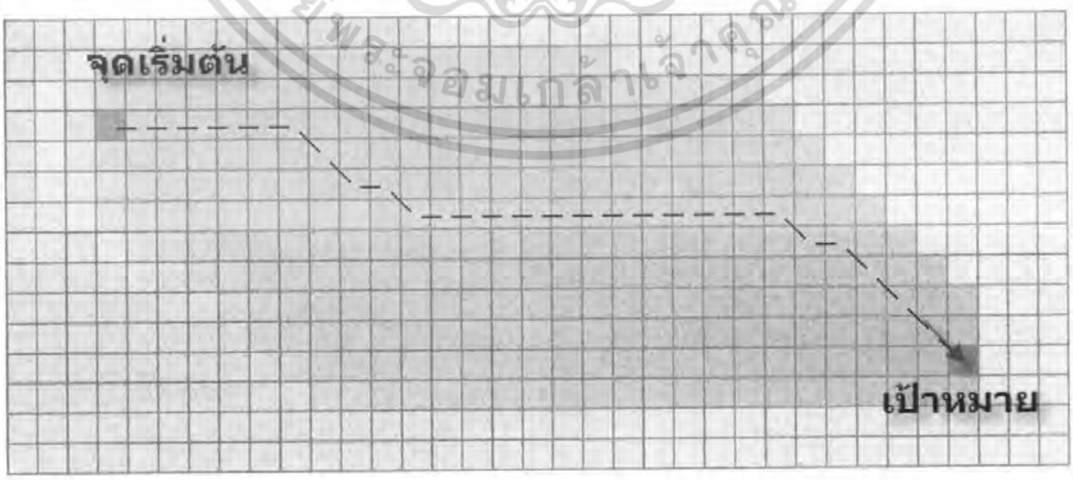
หลังจากนั้นทำการรวมทั้ง 2 ค่าเข้าด้วยกันจะได้ค่า D2 ดังนั้นเส้นตรงที่เหลือทั้งหมดก็คือค่า D

ระยะทางยูคลิเดียน (Euclidean Distance)

ถ้าตัวละครในเกมส์สามารถเคลื่อนที่ไปตามมุมต่าง ๆ ได้ มีความจำเป็นอย่างยิ่งที่จะต้องใช้ระยะทางเส้นตรง

$$h(n) = D * \sqrt{(n.x\text{-}goal.x)^2+(n.y\text{-}goal.y)^2}$$

อย่างไรก็ตามในกรณีนี้เราจะมีปัญหาโดยตรงกับการใช้ A* เพราะว่าค่าฟังก์ชัน g จะไม่ตรงกับค่าฟังก์ชัน heuristic h เนื่องด้วยระยะทางยูคลิเดียนมีค่าที่สั้นกว่าระยะทางระหว่างจุดสองจุด หรือค่าจากแนวทแยงมุม ดังนั้นเส้นทางที่ได้ก็ยังคงเป็นเส้นทางที่สั้นที่สุด แต่ A* จะใช้เวลาในการทำงานที่นานกว่า

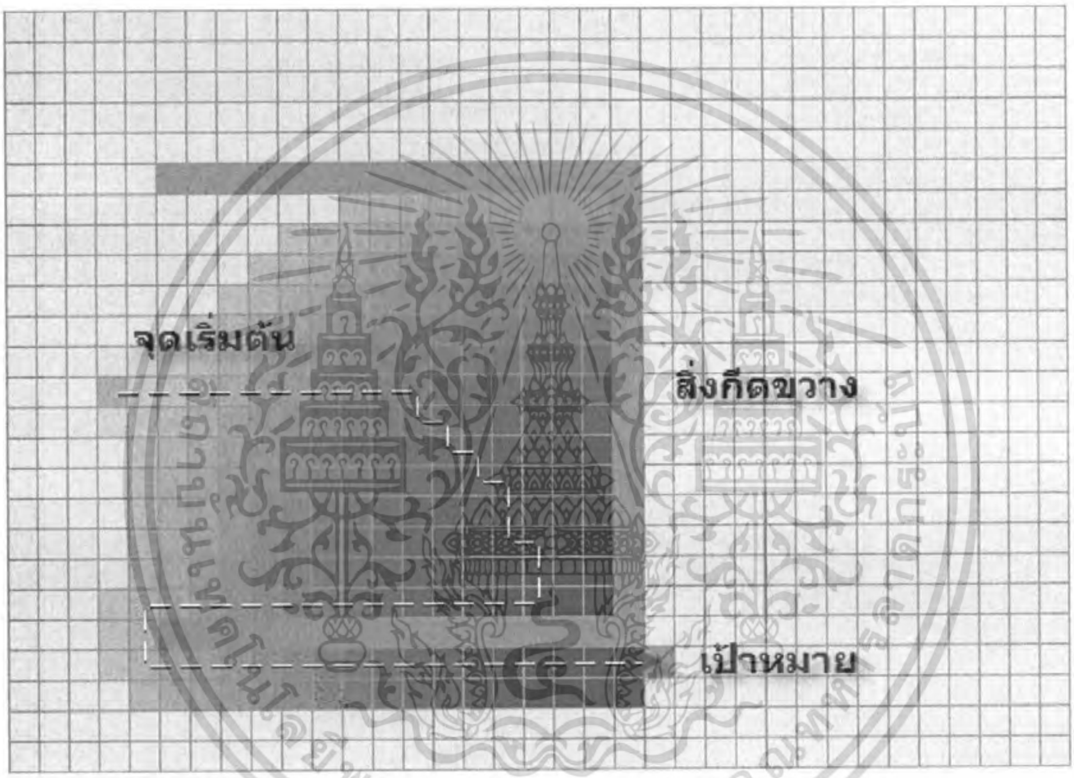


รูปที่ 2.12 การทำงานเมื่อใช้ ทฤษฎีระยะทางยูคลิเดียน

ระยะทางยูคลิเดียนกำลังสอง (Square Euclidean Distance)

โดยทั่วไปแล้วเราจะหลีกเลี่ยงการคำนวณรากที่ 2 ใน Euclidean Distance ซึ่งสามารถใช้ระยะทางกำลังสองมาแทนได้

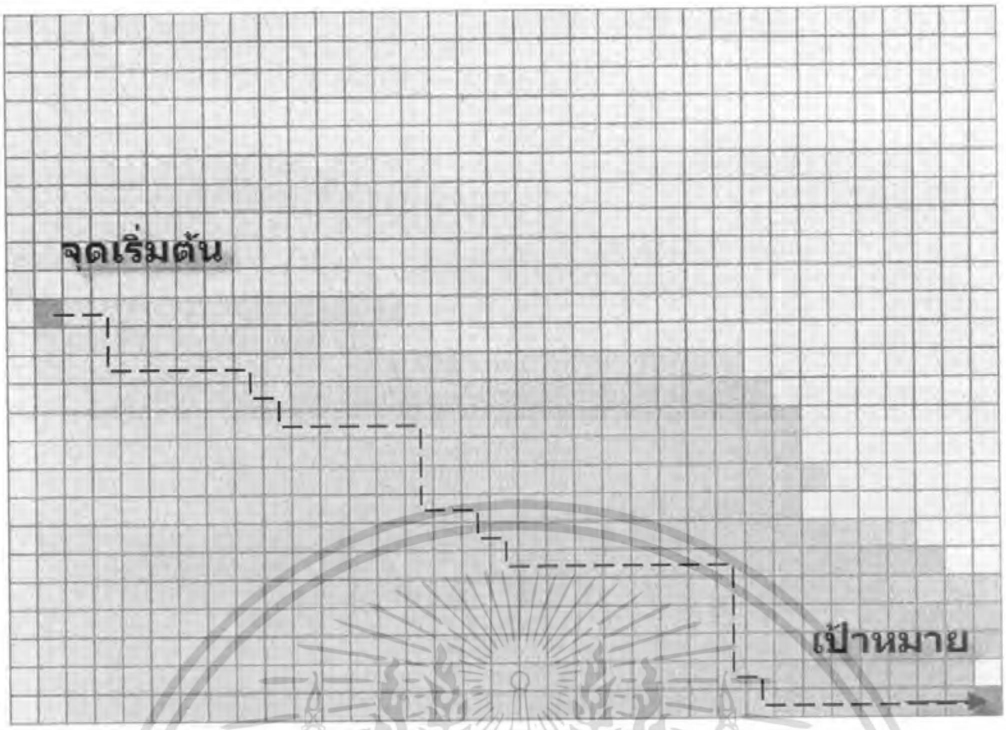
$$h(n) = D * ((n.x - goal.x)^2 + (n.y - goal.y))^2$$



รูปที่ 2.13 การทำงานเมื่อใช้ระยะทางกำลังสอง

Ties Breakier

มีอยู่หนึ่งสิ่งที่สามารถทำให้ประสิทธิภาพด้อยลงได้คือ Tie ต่าง ๆ ใน Heuristic เมื่อใดที่หลาย ๆ เส้นทางมีค่า f เหมือน ๆ กัน ส่งผลให้ทุกเส้นทางนั้นต้องถูกสำรวจ แต่ต้องการเพียงแค่เส้นทางเดียวเท่านั้น



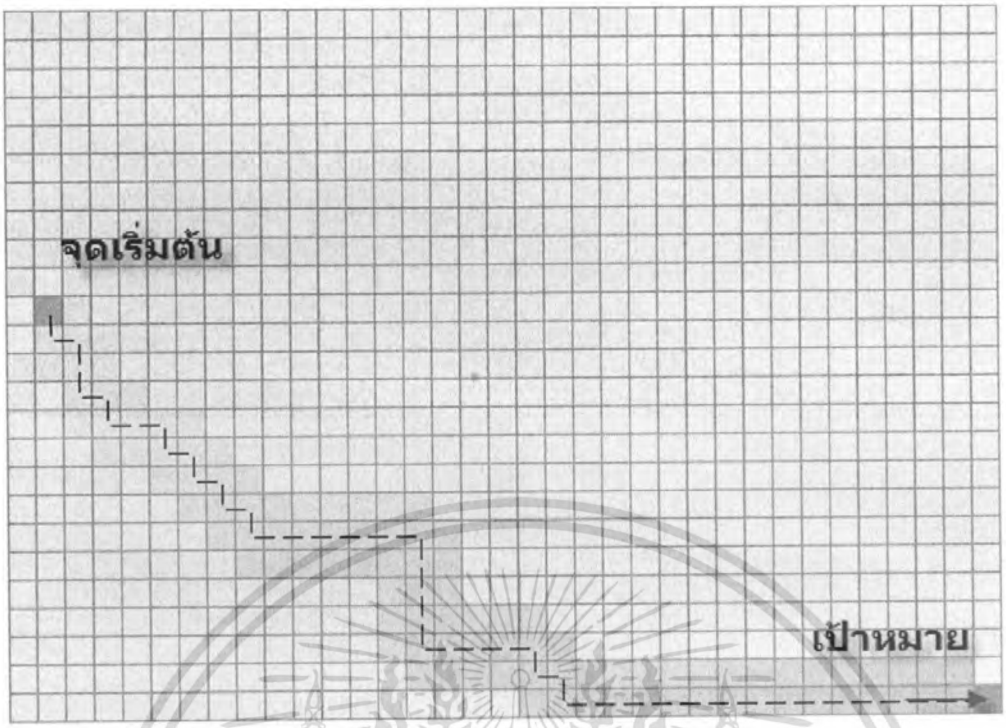
รูปที่ 2.14 Ties ในค่า f

ในการที่จะแก้ปัญหานี้ สามารถเพิ่ม Tie Breaker ขนาดเล็กไปที่ Heuristic ซึ่ง Tie Breaker ต้องการโหนดที่มีความแน่นอน (ไม่ใช่ค่าที่มาจากการสุ่มตัวเลข) และต้องทำให้ค่า f มีค่าแตกต่างกัน หลังจากนั้น A* จะทำการเรียงค่า f และหมายความว่า หนึ่งเดียวเท่านั้นจากค่าที่เท่ากันจะถูกนำไปสำรวจ

อีกหนทางที่จะทำให้ขนาด h เพิ่มขึ้นเล็กน้อย ถ้าค่า f มีค่ามากขึ้นเรื่อย ๆ หมายความว่ากำลังมุ่งหน้าสู่เป้าหมาย แต่หมายถึง A* จะคำนวนใกล้กับโหนดจุดเริ่มต้น แทนที่จะเป็นโหนดของเป้าหมาย สามารถที่จะเพิ่มขนาด h ขึ้นเล็กน้อย (ทีละ 0.1%) ซึ่ง A* จะคำนวนเพิ่มใกล้กับโหนดของเป้าหมาย

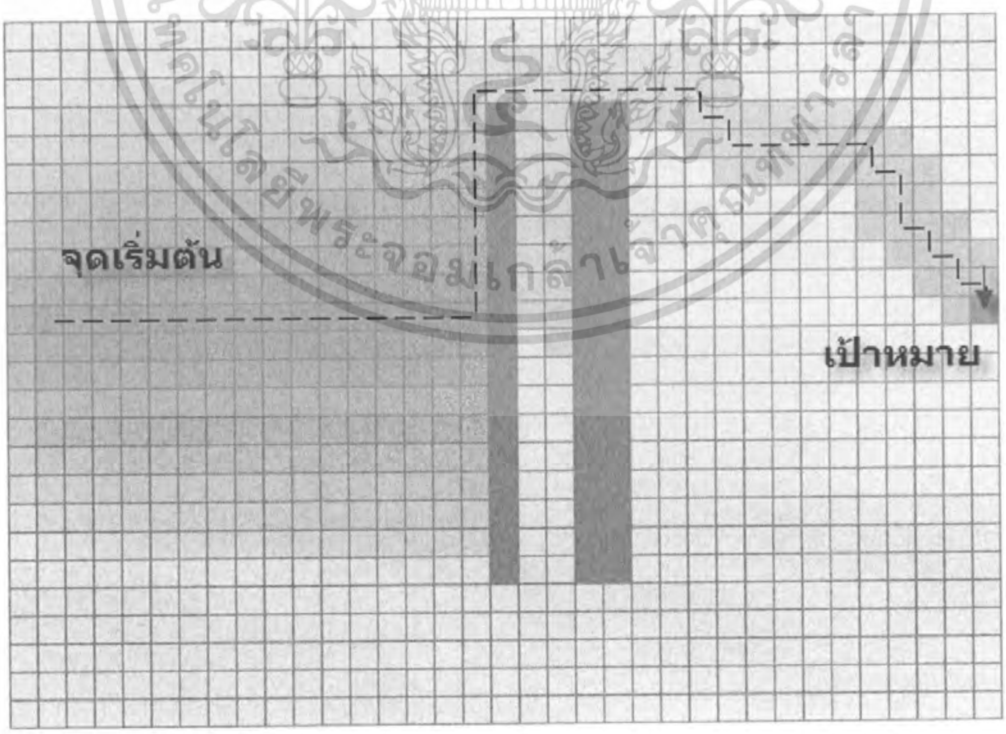
$$\text{Heuristic} *= 1.0 + p$$

ค่าของ p ควรจะถูกเลือกโดยที่ $p <$ (ค่าระยะทางที่น้อยที่สุดในการคำนวณขั้นแรก) / (ค่าคาดหมายของเส้นทางที่มากที่สุด) สมมุติได้ว่าเราไม่ต้องการเส้นทางที่มีขั้นตอนมากกว่า 1000 ขั้นตอน และสามารถเลือกค่า $p = 1/1000$ และในผลลัพธ์ของ Tie-Breaking Nudge จะทำให้ A* สำรวจพื้นที่น้อยกว่าก่อนหน้านี้



รูปที่ 2.15 การเพิ่ม Tie-Breaking Scaling เข้าสู่ Heuristic

เมื่อมีสิ่งกีดขวางเป็นที่แน่นอนว่ายังคงต้องสำรวจเพื่อหาเส้นทางรอบ ๆ สิ่งกีดขวาง แต่จะเกิดขึ้นหลังจากที่ A* ได้คำนวณไว้แล้ว



รูปที่ 2.16 การเพิ่ม Tie-Breaking Scaling เข้าสู่ Heuristic เมื่อมีสิ่งกีดขวาง

จากรูปที่ 2.16 การเพิ่มขนาด Tie-Breaking เข้าสู่ Heuristic จะส่งผลให้ทำงานได้ดีเมื่อมีสิ่งกีดขวาง

dx1 = current.x - goal.x

dy1 = current.y - goal.y

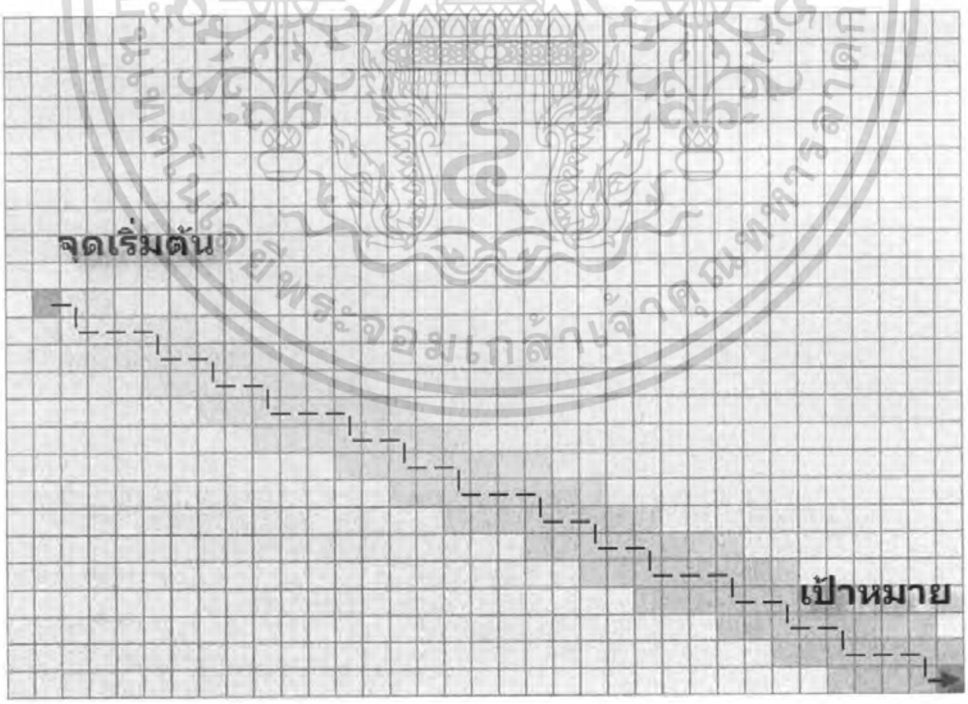
dx2 = start.x - goal.x

dy2 = start.y - goal.y

cross = abs(dx1*dy2 - dx2*dy1)

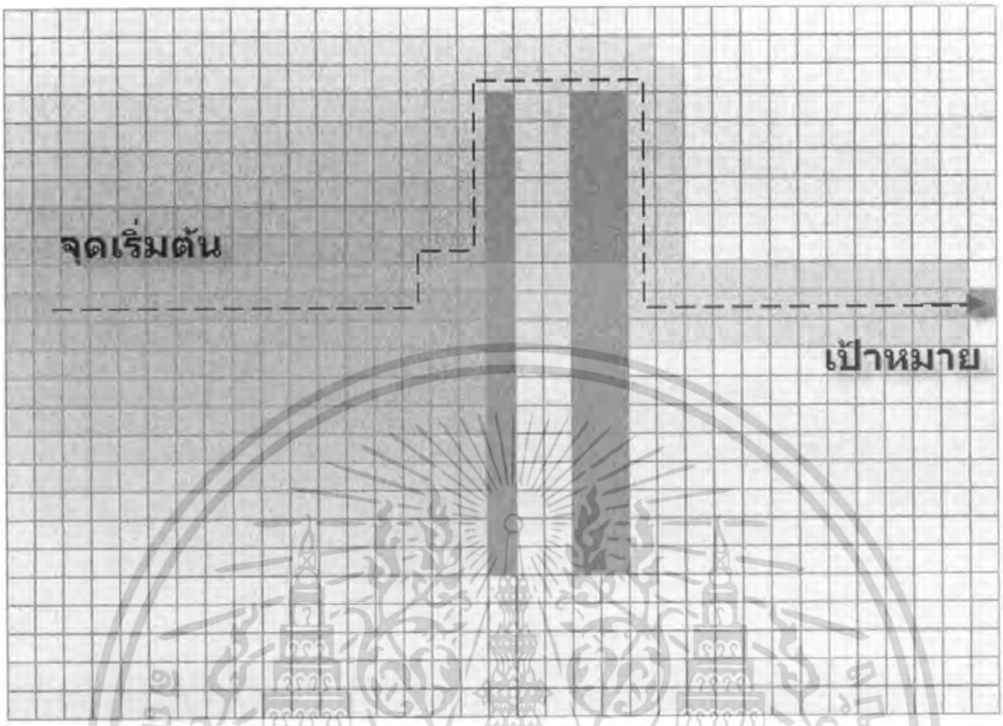
heuristic = heuristic + cross*0.001 หรือ heuristic += cross*0.001

จากโค้ดข้างบน จะทำงานไขว้กันระหว่างจุดเริ่มต้นและจุดเป้าหมาย เมื่อ A* ไม่พบสิ่งกีดขวาง จะทำให้เส้นทางสำรวจลดน้อยลงอย่างมากและเส้นทางการเดินจะเหมาะสมอย่างมาก



รูปที่ 2.17 การเพิ่ม Tie-Breaking Cross-Product ไปสู่ Heuristic

อย่างไรก็ตาม Tie-Breaker ยังคงมีแนวโน้มเป็นเส้นตรงจากจุดเริ่มต้นถึงเป้าหมายและจะเกิดอะไรขึ้นถ้าเดินทางรอบ ๆ สิ่งกีดขวาง



รูปที่ 2.18 การเพิ่ม Tie-Breaking Cross-Product เข้าสู่ Heuristic เมื่อมีสิ่งกีดขวาง

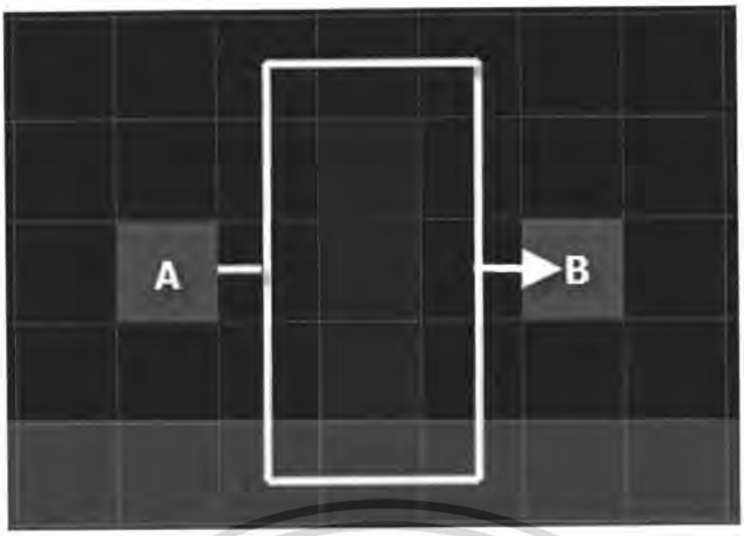
จากรูปที่ 2.18 การเพิ่ม Tie-Breaking Cross-Product เข้าสู่ Heuristic เมื่อมีสิ่งกีดขวาง การเดินค่อนข้างจะเหมาะสม

การประยุกต์ทฤษฎี

A* Algorithm ได้ถูกพัฒนาจากทฤษฎี Dynamic Programming , Branch and Bound และ Graph Search Algorithm โดยใช้พื้นฐานของ Graph Search Algorithm ในการค้นหาบนเส้นกราฟ และใช้ทฤษฎี Dynamic Programming จะช่วยให้สามารถหาเส้นทางที่เป็นไปได้บนเส้นกราฟ ประกอบกับทฤษฎี Branch and Bound ในการตัดเส้นทางที่มีค่ามากกว่า Lower Bound ซึ่งทำให้สามารถค้นหาเส้นทางที่สั้นที่สุด (Shortest Path) ที่มีความรวดเร็วและชาญฉลาด

พื้นที่การค้นหา

สมมติว่าต้องการเดินทางจากจุด A ไปยังจุด B โดยมีกำแพงแบ่งแยกทั้ง 2 จุดออกจากกัน ตามรูป 2.19 จุดเริ่มต้นคือ A จุดปลายทางคือ B และมีกำแพงที่อยู่ระหว่างจุดทั้งสองจุด



รูปที่ 2.19 จุดเริ่มต้นและเป้าหมาย

ในขั้นแรกนั้น ต้องเข้าใจก่อนว่าพื้นที่การค้นหาได้ถูกแยกออกเป็นตารางสี่เหลี่ยมอย่างง่าย เพื่อใช้ในการค้นหาแบบอะเรย์ 2 มิติ โดยแต่ละชิ้นสี่เหลี่ยมแสดงถึง 1 ส่วนของตารางสี่เหลี่ยมทั้งหมด และหมายถึงสถานะที่วัตถุสามารถเคลื่อนที่ได้หรือไม่ได้ ซึ่งเส้นทางได้ถูกค้นหาจะเดินจากจุดกึ่งกลางทางด้านซ้ายไปยังจุดกึ่งกลางถัดไปจนกระทั่งถึงที่หมายปลายทาง โดยจะเรียกจุดกึ่งกลางนี้ว่าโหนด

การเริ่มการค้นหา

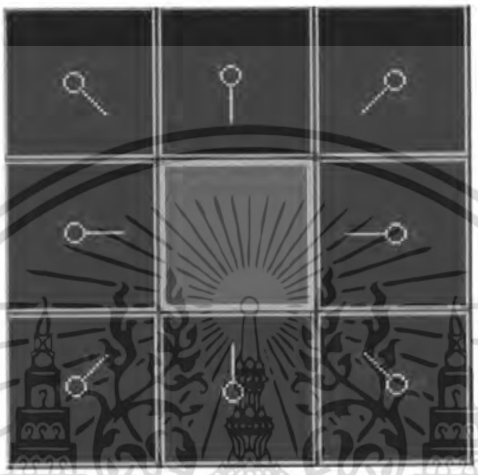
ในตอนนี้มีพื้นที่สำหรับการค้นหาอย่างง่ายที่จะสามารถควบคุมจำนวนของโหนด ขั้นต่อไปเป็นการปฏิบัติที่จะหาเส้นทางที่สั้นที่สุดโดยจะเริ่มค้นหาตามขั้นตอนดังนี้

1. เริ่มต้นที่จุด A และนำเข้าไปที่ "open list" ของตารางที่พิจารณา ซึ่ง "open list" เปรียบเสมือนกับรายการสินค้า และในตอนนี้มี 1 อันที่อยู่ในรายการ แต่ในภายหลังอาจจะมีมากกว่านี้ ซึ่งอาจจะมีเส้นทางที่จะไปหรืออาจจะไม่มีเลย โดยพื้นฐานแล้วตารางของรายการต้องถูกตรวจสอบ

2. มองไปยังที่ ๆ สามารถไปถึงหรือเดินไปได้บนตารางที่ติดกันโดยไม่สนใจตารางที่เป็นกำแพง น้ำ หรือ อื่น ๆ ที่ผิดไปจากสภาพภูมิประเทศ ต่อจากนั้นนำเข้าไปที่ "open list" สำหรับแต่ละตารางนี้ให้บันทึกเป็น "parent square" ซึ่งมีความสำคัญเมื่อต้องการที่จะเขียนแบบทางเดิน

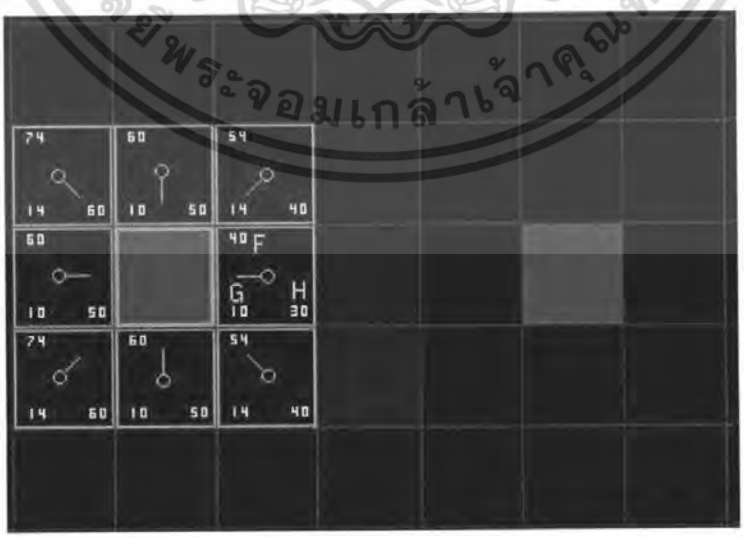
3. ละทิ้งตารางเริ่มต้น A จาก “open list” และนำเข้าไปสู่ “closed list” ของตารางที่ไม่ต้องการจะพิจารณาอีกครั้งในตอนนี้

ในที่นี้ในตารางที่อยู่ตรงกลางของภาพเป็นจุดเริ่มต้น และกรอบของตารางตรงกลางเป็นสิ่งบอกให้ทราบว่าได้ถูกเพิ่มเข้าสู่ “closed list” แล้ว และในส่วนของกรอบตารางอื่น ๆ คือ “open list” ที่จะต้องตรวจสอบโครงร่าง ส่วนตัวชี้ที่ออกมากจากจุดกำเนิดเป็นจุดเริ่มต้น



รูปที่ 2.20 ทิศทางการเดิน

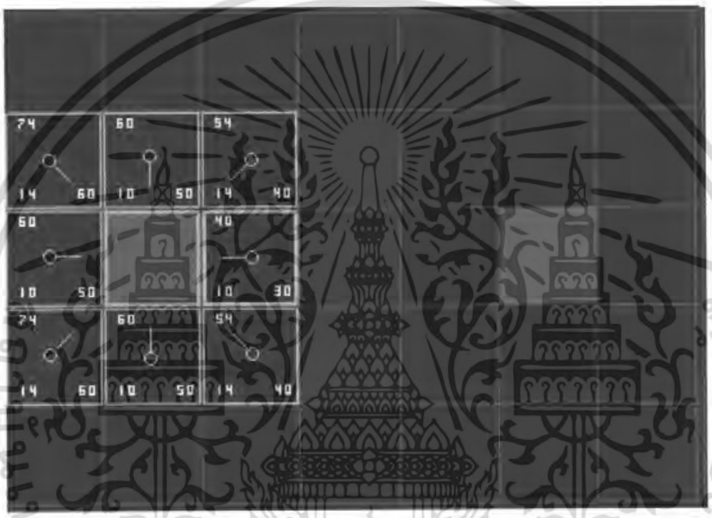
ขั้นต่อไปจะเลือกจุดที่ติดกัน 1 จุด หรือมากกว่านั้น บน “open list” ก่อนที่จะทำการประมวลผล ดังที่ได้อธิบายไว้ แต่ตารางไหนที่จะเลือกด้วยค่า F ที่ต่ำที่สุด



รูปที่ 2.21 ทิศทางการเดินจากจุดเริ่มต้น

เพื่อที่จะเริ่มค้นหาต่อ จะเลือกค่า F ที่น้อยที่สุดอย่างง่าย จาก “open list”

4. ทิ้งจาก “open list” และเพิ่มเข้าสู่ “closed list”
5. ตรวจสอบตารางที่ใกล้ๆกันทั้งหมด ยกเว้นตารางที่อยู่ใน “closed list” หรือไม่สามารถเดินไปได้ หลังจากนั้น ถ้าตารางไหนไม่มีอยู่ใน “open list” ก็ให้เลือกเป็นตาราง “parent”
6. ถ้าตารางรอบ ๆ นั้นอยู่ใน open list อยู่แล้ว ให้ทำการตรวจสอบเส้นทางว่าดีที่สุดหรือไม่ กล่าวคือ ถ้าค่า G ที่ได้นั้นมีค่าลดลง แสดงว่าได้เลือกเส้นทางที่จะไปยังเป้าหมายนั่นเอง



รูปที่ 2.22 การเคลื่อนที่จากจุดเริ่มต้น

การให้คะแนนเส้นทาง

การให้คะแนนเส้นทางแต่ละเส้นทางนั้นเพื่อจะได้นำคะแนนที่ได้ของแต่ละเส้นทางไปเปรียบเทียบว่าเส้นทางไหนที่ได้คะแนนมากที่สุด หมายความว่าเส้นทางที่ได้คะแนนมากที่สุด คือเส้นทางที่มีโอกาสจะเป็นเส้นทางที่สั้นมากที่สุด กุญแจที่จะใช้ตัดสินใจว่าจะเลือกตารางไหนที่จะใช้เป็นเส้นทางตามสมการ

$$F = G + H$$

โดยกำหนด

- G เป็นค่าการเคลื่อนที่ที่จะเคลื่อนที่จากจุดเริ่มต้น A ไปยังตารางที่มีตามเส้นทางที่ได้สร้างขึ้นเพื่อจะไปถึงปลายทาง
- H เป็นค่าประมาณการเคลื่อนที่ที่จะเคลื่อนที่จากจุดที่เรากำหนดไว้บนตารางไปสู่ปลายทางสุดท้าย (จุด B) ซึ่งเราจะนิยมอ้างอิงถึง Heuristic Function
- F เป็นค่าที่ได้จาก Heuristic บวกกับค่าระยะทาง โดยจะเป็นตัวบ่งบอกว่าเส้นทางดังกล่าวนั้นเหมาะสมที่จะผ่านหรือไม่ โดยวัดจากตัวเลข

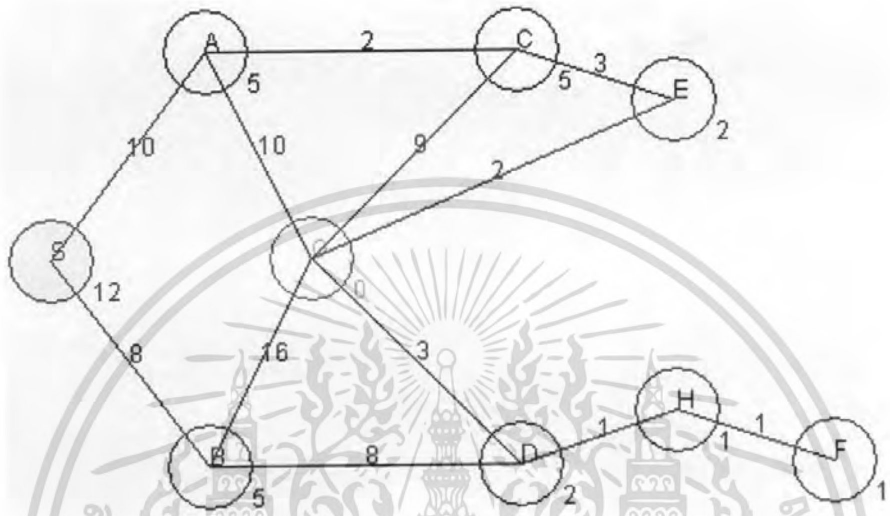
เส้นทางได้ถูกสร้างขึ้นโดยการเดินผ่าน “open list” ซ้ำ ๆ และเลือกตารางที่มีค่า F ต่ำที่สุดตามที่ได้อธิบาย

ตัวอย่างการทำงานของ A*

กำหนดให้

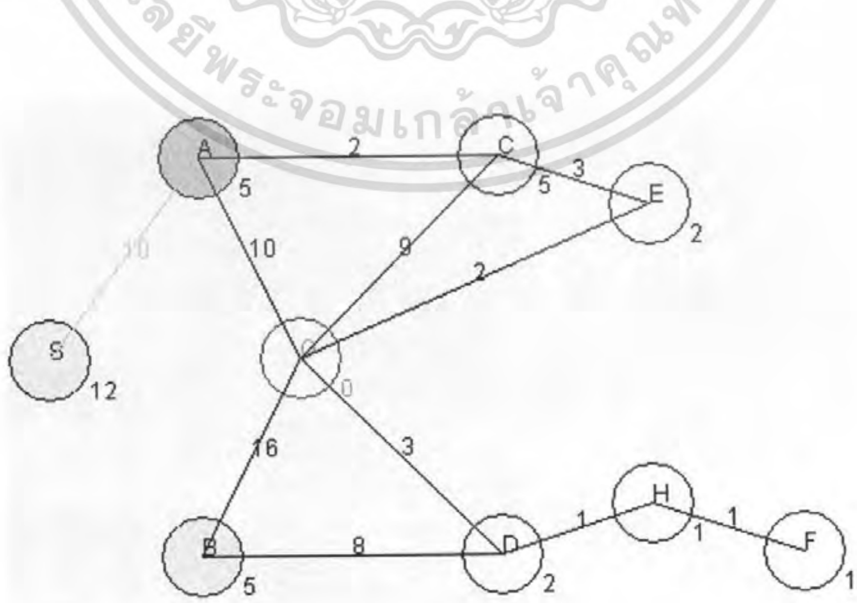
- **Heuristic = H**
- **Cost = C**
- **Priority = F**

1. S มีค่าคือ H:12,C:0,F:12 โดยมีจุดเชื่อมต่อสองจุดคือ A และ B จาก S คือจุดเริ่มต้น และมี G คือจุดปลายทาง



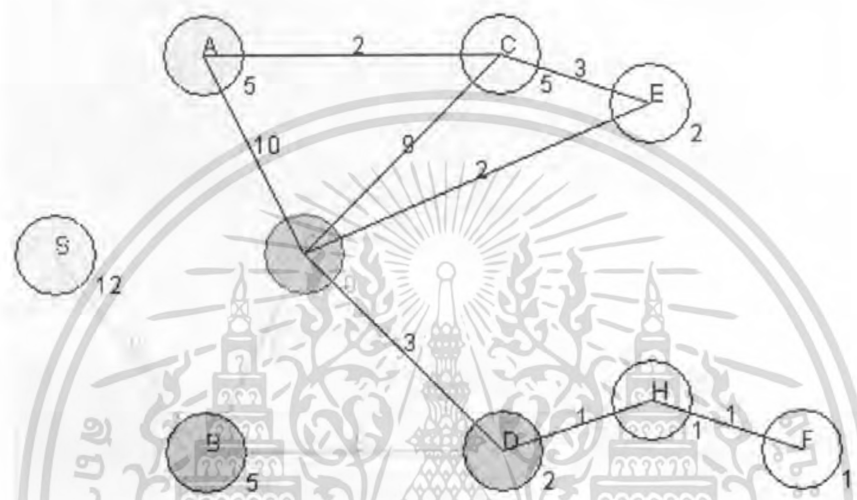
รูปที่ 2.23 การทำงานของ A* ในระดับ 1

2. ตัดสินใจว่าจะทำการทดสอบว่าเส้นทางไหนมี Priority น้อยที่สุดในกลุ่ม (ในที่นี้คือ 2 ตัวเลือกคือ S-B และ S-A) ซึ่งมีค่าดังต่อไปนี้ S-B มีค่า H:5,C:8,F:13 คือเส้นทางที่ดีที่สุด และส่วน S-A มีค่า H:5,C:10,F:15 คือเส้นทางที่น่าจะเป็นไปได้



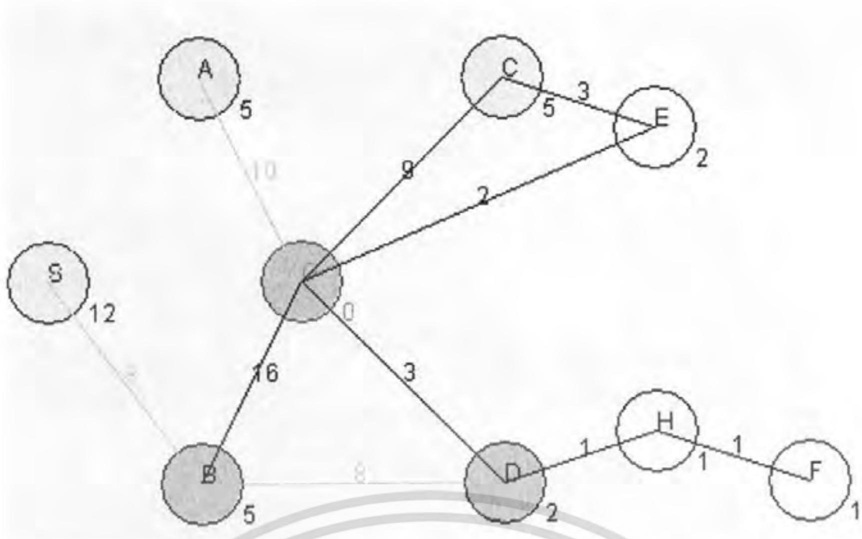
รูปที่ 2.24 การทำงานของ A* ในระดับ 2

3. ในตอนนี้เราตัดสินใจได้แล้วว่าเราจะไป S-B เพราะมีค่า F น้อยที่สุดในกลุ่ม เมื่อถึง S-B แล้วก็ทำการทดสอบเส้นทางอีกครั้งโดยมีจุดเชื่อมต่อ 2 จุดที่ต่อกับ B คือ D และ G จะได้ค่าที่ B-D และ B-G คือ S-B-D มีค่า H:2,C:16,F:18 และ S-B-G มีค่า H:0,C:24,F:24 แต่ยังมีค่าเก่าของ SA อยู่และยังเป็นเส้นทางที่ดีที่สุดและมีค่า H:5,C:10,F:15 ซึ่งต้องเอามาคิดด้วยก็จะได้ 3 ตัวเลือก โดยในตัวเลือกครั้งนี้นั้น S-A มีค่า F น้อยที่สุดในกลุ่ม และ G นั้นมีการเชื่อมต่อกับเส้นทางอื่น อยู่แสดงว่าน่าจะมีเส้นทางที่มีความเป็นไปได้ว่าจะดีกว่า S-B-G



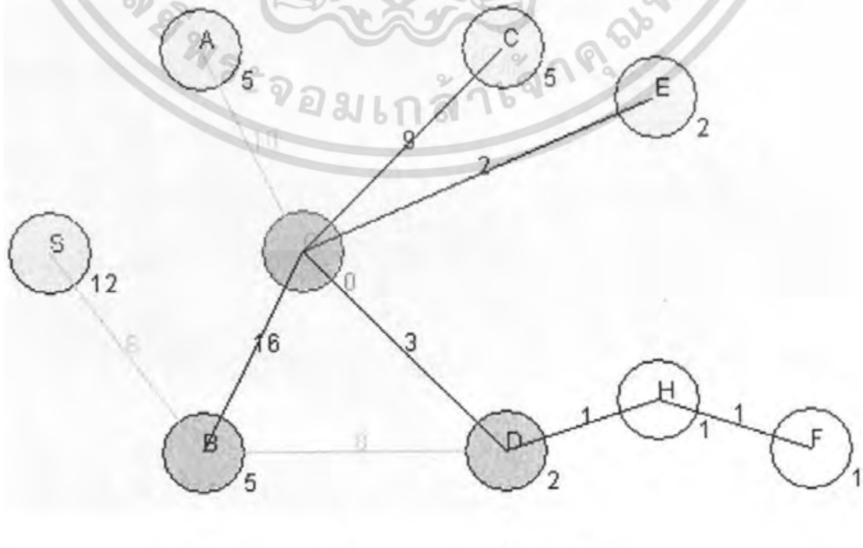
รูปที่ 2.25 การทำงานของ A* ในระดับ 3

4. เมื่อถึง S-A แล้วก็ทำการทดสอบเส้นทางที่เชื่อมต่อกับ A ซึ่งมีเส้นทางอยู่ 2 เส้นทางคือ C และ G ก็จะได้ S-A-C ที่มีค่า H:5,C:12,F:17 คือเส้นทางที่ดีที่สุด และ S-A-G ที่มีค่า H:0,C:20,F:20 คือเส้นทางที่น่าจะเป็นไปได้ แต่อย่าลืมว่ามีค่าเก่าอยู่อีก 2 ตัวคือ S-B-D มีค่า H:2,C:16,F:18 และ S-B-G มีค่า H:0,C:24,F:24 และ 2 เส้นนี้ก็เป็นเส้นทางที่น่าจะเป็นไปได้ แต่ S-B-G นั้นมีค่ามากที่สุด และยังมีจุด G ซึ่งซ้ำกับ S-A-G ที่มีค่าน้อยกว่า เลยตัดทิ้งไป เพราะจุดหมายปลายทางคือ G ซึ่งต้องการหาค่าเส้นทางที่ไปถึง G ที่น้อยที่สุดเท่านั้นจึงตัด S-B-G ทิ้งไป โดยในที่นี้ ค่า F ของ S-A-C นั้นน้อยที่สุดในกลุ่ม ก็เลือกให้เดินต่อไปที่ S-A-C โดย S-A-G นั้นยังมีเส้นทางอื่นที่เชื่อต่อแสดงว่าน่าจะมีเส้นทางที่ดีกว่า



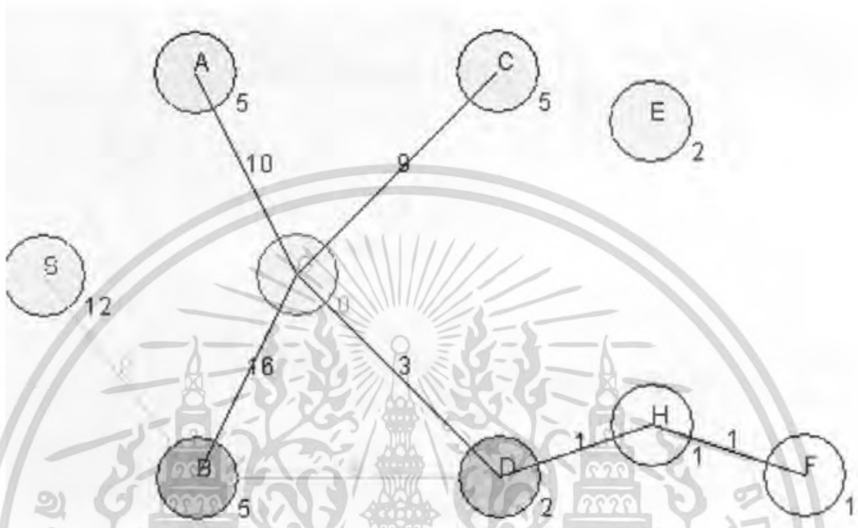
รูปที่ 2.26 การทำงานของ A* ในระดับ 4

5. เมื่อถึง S-A-C จะมีเส้นทางอยู่ 2 เส้นคือ E และ G โดยที่ S-A-C-E นั้นมีค่า H:2,C:15,F:17 และเป็นเส้นทางที่ดีที่สุด ส่วน S-A-C-G นั้นมีค่า H:5,C:21,F:26 ยังเป็นเส้นทางที่น่าจะเป็นอยู่และค่าที่เหลืออยู่คือ S-B-D มีค่า H:2,C:16,F:18 และ S-A-G มีค่า H:0,C:20,F:20 แต่ค่า F ของ S-A-C-E นั้นน้อยที่สุดในกลุ่มทั้ง 4 ตัว จึงเลือกเดินไปที่ E และ S-A-C-G ก็ต้องนำออกจากกลุ่มด้วยเพราะ S-A-G นั้นมีค่า F น้อยกว่า



รูปที่ 2.27 การทำงานของ A* ในระดับ 5

6. โดยเมื่อถึง E แล้ว เส้นทางมีเส้นเดียวคือเส้น S-A-C-E-G ซึ่งมีค่า H:0,C:17,F:17 คือเป็นเส้นทางที่ดีที่สุด ซึ่งมีค่าน้อยกว่า S-B-D ที่มีค่า H:2,C:16,F:18 และ S-A-G ที่มีค่า H:0,C:20,F:20 จะเลือกเส้นทาง S-A-C-E-G เป็นเส้นทางที่ดีที่สุดในการเดินทางจาก S ไปถึง G โดยใช้ Priority ที่ 17, Cost ที่ 17 และ Heuristic ที่ 0



รูปที่ 2.28 การทำงานของ A* ในระดับ 6

การทำให้มีประสิทธิภาพสูงสุด

ตามที่ได้พัฒนาโปรแกรม A* จะค้นพบว่า Pathfinder เป็นวิธีดั้งเดิมที่นิยมใช้กัน อัตราความเร็ว/สภาพการทำงานของ CPU โดยเฉพาะอย่างยิ่งถ้ามีหมายเลขที่เหมาะสมของPathfinding unit และมีแผนที่ที่สมบูรณ์กว้างขวางอย่างมีเหตุผล ถึงแม้สำหรับมืออาชีพหลายคนผู้ซึ่งออกแบบเกมส์ อาทิเช่น เกมส์ Starcraft หรือ เกมส์ Age of Empires ถ้าเห็นสิ่งของหรือการดำเนินเรื่องเริ่มช้าลง เนื่องจาก Pathfinder มีความคิดที่จะทำให้สิ่งเหล่านั้นเพิ่มความเร็วขึ้น

- พิจารณา/นึกถึง แผนที่ที่เล็กกว่าหรือมีจำนวนหน่วยน้อยกว่า

-ห้ามทำ Pathfinder สำหรับจำนวนหน่วยน้อยๆใน 1 ครั้ง เพราะจะทำให้การทำงานของระบบเกมส์ ทำงานหนักขึ้น ส่งผลกระทบให้การทำงานของตัวเกมส์ช้าลง

- พิจารณาการใช้ตารางพื้นที่สี่เหลี่ยมที่ใหญ่กว่า หรือรูปทรงที่ใช้สำหรับแผนที่ เป็นการลดจำนวนทั้งหมดของโหนด เพื่อหาแนวทางและสามารถออกแบบแนวคิดหรือแผนการออกเป็นสองหรือมากกว่าสองระบบ ซึ่งถูกใช้ในต่างสถานการณ์ ขึ้นอยู่กับระยะเวลาและระยะทางของเส้นทาง

- สำหรับเส้นทางที่ยาวมากขึ้น ให้พิจารณาการออกแบบและวางแผนก่อนคำนวณเส้นทาง

- ให้พิจารณาถึงการผ่านกระบวนการ การจัดการก่อนเพื่อคิดให้ได้ว่าพื้นที่ไหนยากต่อการเข้าถึงจากทั้งหมดในแผนที่ ในความเป็นจริงพื้นที่นั้น สามารถเป็นเกาะหรือพื้นที่อย่างอื่นก็ได้ซึ่งอย่างหนึ่งคือ ปิดกั้นและที่ที่เข้าไปไม่ได้

พิจารณาอีกด้านของ A* คือ ถ้าบอกให้ดูเส้นทางในหลาย ๆ พื้นที่ ก็จะค้นหาทั้งแผนที่ และจะหยุดเพียงแค่เมื่อทุก ๆ การเข้าถึงพื้นที่ได้ถูกผ่านขบวนการ แล้วผ่าน Open List และ Closed List ทำให้เสียเวลาในการคำนวณของ CPU มาก สามารถป้องกันโดยการตัดสินใจก่อน ซึ่งพื้นที่คือส่วนที่ยากต่อการเข้าถึง แล้วตรวจสอบก่อนการเริ่มตรวจสอบเส้นทาง

2.1.3 ทฤษฎีทางสถิติที่เกี่ยวข้อง

1. การทดสอบการแจกแจงแบบปกติ

1.1 การทดสอบของชาพิโร - วิลค์ (Shapiro – Wilk Test) ใช้กับกรณีที่ทราบหรือไม่ทราบค่าเฉลี่ยและความแปรปรวนของประชากรก็ได้ แต่ตัวอย่างต้องมีขนาดเล็ก $n \leq 50$ กำหนดสมมติฐาน ดังนี้

H_0: ข้อมูลจากประชากรมีการแจกแจงแบบปกติ
H_1: ข้อมูลจากประชากรไม่ได้มีการแจกแจงแบบปกติ

สถิติทดสอบ

$$W = \frac{c^2}{\sum (y_i - \bar{y})^2}$$

เมื่อ $c = \sum_{i=1}^{k} a_{n-1+i} \, (y_{n-i+1} - y_i)$

เขตปฏิเสธ

ทำการเปรียบเทียบค่าสถิติทดสอบ W กับค่าวิกฤต $W_{\alpha,n}$ (จากตาราง) โดยถ้า $W < W_{\alpha,n}$ จะปฏิเสธสมมติฐานหลักนั่นคือข้อมูลไม่ได้มีการแจกแจงปกติ

ในการทำปัญหาพิเศษนี้ไม่มีการกำหนดค่าพารามิเตอร์ของประชากรไว้และมีขนาดตัวอย่าง 20 หน่วย เพราะฉะนั้นจึงใช้วิธีการทดสอบของ Shapiro – Wilk Test

(ข้อมูล http://cyberclass.msu.ac.th)

1.2 การทดสอบโครโมโกรอฟ – สไมร์นอฟ (Kolmogorov-Smirnov Test)

หลักเกณฑ์ของ Kolmogorov-Smirnov Test ในการทดสอบการแจกแจงของประชากร คือ การเปรียบเทียบความน่าจะเป็นสะสมของตัวอย่าง (S(x)) กับความน่าจะเป็นสะสมภายใต้สมมติฐานว่าง H_0 (F(x))

H_0 : ประชากรมีการแจกแจงแบบปกติ

H_1 : ประชากรไม่มีการแจกแจงแบบปกติ

สมมติฐานว่าง H_0 จริง (S(x)) และ (F(x)) จะมีค่าใกล้เคียงกันทุกค่าของ x แต่ถ้า H_0 ไม่เป็นจริงคือประชากรไม่ได้มีการแจกแจงแบบปกติ ค่า (S(x)) และ (F(x)) จะแตกต่างกันมากสำหรับบางค่าของ x

โดยที่ $F(x) = P(X \leq x)$

$$F(x) = \sum_{0}^{x} P(X \leq x) \qquad \text{ถ้า x เป็นตัวแปรสุ่มแบบไม่ต่อเนื่อง}$$

หรือ

$$F(x) = \int_{-\infty}^{x} f(x)dx \qquad \text{ถ้า x เป็นตัวแปรสุ่มแบบต่อเนื่อง}$$

สถิติทดสอบ

$$D = \max |F(x)-S(x)|$$

เขตปฏิเสธ

ถ้า D มีค่ามากแสดงว่า $F(x)$ และ $S(x)$ แตกต่างกันมาก จึงปฏิเสธ H_0 นั่นคือ ถ้า D > ค่าวิกฤตจะปฏิเสธสมมติฐาน H_0 นั่นคือ ประชากรไม่ได้มีการแจกแจงแบบปกติ และถ้า D < ค่าวิกฤต จะยอมรับ H_0 นั่นคือ ประชากรมีการแจกแจงแบบปกติ

(กัลยา, 2545)

1.3 การทดสอบลิลลิฟอร์ส (Lilliefor's Test)

Lilliefors Test เป็นการทดสอบการแจกแจงของประชากรว่ามีการแจกแจงแบบปกติหรือไม่ โดยต่างจาก Kolmogorov-Smirnov Test คือ Kolmogorov-Smirnov Test จะต้องกำหนดค่าเฉลี่ย $\mu = \mu_0$ และค่าเบี่ยงเบนมาตรฐาน $\sigma = \sigma_0$ ไว้ในสมมติฐาน H_0 แต่ Lilliefors Test จะไม่กำหนดค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน จึงต้องประมาณ μ ด้วย $\overline{X}$ และประมาณ σ ด้วย S

H_0 ประชากรมีการแจกแจงแบบปกติ

H_1: ประชากรไม่ได้มีการแจกแจงแบบปกติ

สถิติทดสอบ

$$D = \max |F(x) - S(x)|$$

โดย $F(x) = P(X \le x) = P\left(Z < \frac{X - \overline{X}}{s}\right)$

เขตปฏิเสธ

จะปฏิเสธ H_0 ถ้า D > ค่าวิกฤตที่ได้จากตาราง Lilliefors Test นั่นคือ ถ้า D > ค่าวิกฤตจะปฏิเสธสมมติฐาน H_0 นั่นคือ ประชากรไม่ได้มีการแจกแจงแบบปกติ และถ้า D < ค่าวิกฤต จะยอมรับ H_0 นั่นคือ ประชากรมีการแจกแจงแบบปกติ

(กัลยา, 2545)

1.4 การทดสอบแบบไค – สแควร์ (Chi-Square Test)

Chi-Square Test จะใช้ในการทดสอบลักษณะต่าง ๆ ของประชากรว่าเป็นไปตามคาดไว้หรือไม่ จะใช้สำหรับตัวแปรเชิงกลุ่ม เช่น ความคิดเห็น ลักษณะของสินค้า ช่วงของรายได้ ซึ่งตัวแปรเชิงกลุ่มที่ใช้ Chi-square Test อาจจะมี 2 ค่า หรือมากกว่า 2 ค่าก็ได้ และสามารถใช้ทดสอบเกี่ยวกับการแจกแจงของประชากร หรือทดสอบความเป็นอิสระกันของลักษณะ 2 ลักษณะ

H_0 : สัดส่วนของกลุ่มต่าง ๆ เป็นไปตามที่คาดหวังตามทฤษฎี

H_1 : สัดส่วนของกลุ่มต่าง ๆ ไม่ได้เป็นไปตามที่คาดหวังตามทฤษฎี

สถิติทดสอบ

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

เมื่อ O_i คือ ความถี่ที่สังเกตได้

E_i คือ ความถี่ที่คาดหวัง และ $E_i = n_i \cdot p$; $i=1,2,\ldots,k$

เขตปฏิเสธ

จะปฏิเสธ H_0 ถ้า $\chi^2 >$ ค่าวิกฤตที่ได้จากตาราง **Chi-Square Test** นั่นคือ ถ้า $\chi^2 >$ ค่าวิกฤตจะปฏิเสธสมมติฐาน H_0 นั่นคือ สัดส่วนของกลุ่มต่าง ๆ ไม่ได้เป็นไปตามที่คาดหวังตามทฤษฎี และถ้า $\chi^2 <$ ค่าวิกฤตจะยอมรับ H_0 นั่นคือ สัดส่วนของกลุ่มต่าง ๆ เป็นไปตามที่คาดหวังตามทฤษฎี

ถ้า $k = 2$ และจำนวนความถี่ทั้งหมดน้อยกว่า 50 ควรใช้

$$\chi^2 = \sum_{i=1}^{2} \frac{\left(|O_i - E_i - 0.5|\right)^2}{E_i}$$

และในแต่ละกลุ่มควรจะมีความถี่ที่คาดหวังไม่ต่ำกว่า 1 ในกรณีที่ความถี่คาดหวังเซลใดมีค่าน้อยกว่า 1 ให้ทำการยุบเซล โดยรวมความถี่เซลนั้นกับเซลที่ใกล้เคียง ในกรณีมีความถี่คาดหวังเซลมีค่าระหว่าง 1 ถึง 5 คิดเป็นร้อยละ เกินร้อยละ 20 ของจำนวนเซลทั้งหมด ควรรวมเซลที่อยู่ติดกันเข้าเป็นพวกเดียวกัน

(ข้อมูล http://ph.mykku.net)

2. การแปลงข้อมูล (Transformation of Data)

เมื่อข้อมูลไม่เป็นไปตามข้อสมมติ หากยังต้องการใช้การทดสอบแบบพาราเมตริก (Parametric Test) ควรทำการแปลงข้อมูลเพื่อให้ข้อมูลใหม่เป็นไปตามข้อสมมติ การแปลงข้อมูลอาจกระทำได้ดังนี้

2.1 การแปลงโดยใช้รากที่สอง (Square Root Transformation)

ถ้าทราบความสัมพันธ์ระหว่างค่าเฉลี่ยและความแปรปรวนโดย $\sigma_i^2 = k\mu_i$ นั่นคือ ค่าเฉลี่ยและความแปรปรวนเป็นสัดส่วนกัน จะทำให้ความแปรปรวนไม่เป็นอิสระจากค่าเฉลี่ย การแปลงข้อมูลโดยใช้รากที่สองคือ

$$y_{ij}^* = \sqrt{y_{ij}}$$

เมื่อ y_{ij} คือค่าสังเกตเดิม

ถ้า y_{ij} มีค่าน้อยกว่า 10 ควรแปลงค่าสังเกตโดยใช้

$$y_{ij}^* = \sqrt{y_{ij}} + \sqrt{1+y_{ij}}$$

การแปลงข้อมูลโดยใช้รากที่สองเหมาะสำหรับข้อมูลที่มีการแจกแจงแบบพัวซอง (Poisson Distribution) นั่นคือค่าสังเกตเป็นจำนวนนับ

2.2 การแปลงโดยใช้อาร์คไซน์ (Arcsine Transformation)

ถ้าข้อมูลเป็นสัดส่วนกัน กล่าวคือข้อมูลมีการแจกแจงแบบทวินาม (Binomial Distribution) ควรแปลงข้อมูลโดยใช้

$$y_{ij}^* = \arcsin\sqrt{y_{ij}}$$

ถ้าค่าสังเกตเป็นเปอร์เซ็นต์ต้องทำให้เป็นสัดส่วนเสียก่อนจึงค่อยทำการแปลงข้อมูล การแปลงโดยวิธีนี้จะช่วยทำให้ข้อมูลมีการแจกแจงแบบปกติ และความแปรปรวนมีความเป็นเอกภาพมากขึ้น

2.3 การแปลงโดยใช้ลอกาลิทึม (Logarithmic Transformation)

ถ้าข้อมูลมีความแปรปรวนและค่าเฉลี่ยยกกำลังสองเป็นสัดส่วนกัน นั่นคือ $\sigma_i^2 = k\mu_i^{\ 2}$ กรณีอาจเกิดขึ้นเมื่อข้อมูลมีลักษณะเบ้มาก ๆ ควรแปลงข้อมูลโดยใช้

$$y_{ij}^* = \log(y_{ij})$$

ถ้าค่าสังเกตบางค่ามีค่าเป็นศูนย์หรือมีค่าเข้าใกล้ศูนย์ ควรแลงข้อมูลโดยใช้

$$y_{ij}^* = \log(y_{ij}+1)$$

2.4 การแปลงโดยใช้เศษส่วน (Reciprocal Transformation)

ถ้าส่วนเบี่ยงเบนมาตรฐานและค่าเฉลี่ยยกกำลังสองเป็นสัดส่วนกัน นั่นคือ $\sigma_i = k\mu_i^{\ 2}$ ควรแปลงข้อมูลโดยใช้

$$y_{ij}^* = \frac{1}{y_{ij}}$$

ถ้าค่าสังเกตบางค่าเป็นศูนย์หรือมีค่าเข้าใกล้ศูนย์ ควรแลงข้อมูลโดยใช้

$$y_{ij}^* = \frac{1}{y_{ij}+1}$$

(ข้อมูล http://cyberclass.msu.ac.th)

3. การทดสอบสมมติฐานความแตกต่างระหว่างค่าเฉลี่ย 2 ค่าที่ได้จากกลุ่มตัวอย่างสองกลุ่มที่ไม่อิสระกัน หรือกลุ่มตัวอย่างอย่างสัมพันธ์กัน

เป็นการทดสอบความแตกต่างของค่าเฉลี่ยระหว่างตัวอย่าง 2 กลุ่มเมื่อข้อมูลตัวอย่างที่ใช้ทดสอบมีความสัมพันธ์กัน ซึ่งความสัมพันธ์ดังกล่าวเกิดจากหน่วยทดลอง (Experiment Unit) เดียวกัน แต่มีการวัดค่าของข้อมูล 2 ครั้ง

ข้อกำหนดเบื้องต้น

1. ตั้งสมมติฐาน

การทดสอบแบบสองหาง

H_0 : ค่าเฉลี่ยของกลุ่มตัวอย่างไม่แตกต่างกัน หรือ $\mu_d = 0$

H_1 : ค่าเฉลี่ยของกลุ่มตัวอย่างแตกต่างกัน หรือ $\mu_d \neq 0$

การทดสอบแบบหางเดียวด้านขวา

$H_0 : \mu_d = 0$

$H_1 : \mu_d > 0$

การทดสอบแบบหางเดียวด้านซ้าย

$H_0 : \mu_d = 0$

$H_1 : \mu_d < 0$

2. กำหนดระดับนัยสำคัญ ($\alpha = 0.05$)
3. อาณาเขตวิกฤต

-กรณีการทดสอบแบบสองหาง คือ

$t < -t_{\frac{\alpha}{2},n-1}$ หรือ $t > t_{\frac{\alpha}{2},n-1}$

-กรณีการทดสอบแบบหางเดียวด้านขวา คือ

$t > t_{\alpha,n-1}$

-กรณีการทดสอบแบบหางเดียวด้านขวา คือ

$t < -t_{\alpha,n-1}$

4. ค่าสถิติที่ใช้ในการทดสอบ

$$t = \frac{\bar{d}}{S_{\bar{d}}} = \frac{\bar{d}}{S_d/\sqrt{n}}$$

เมื่อ

$$S_d = \sqrt{\frac{\sum_{i=1}^{n}(d_i-\bar{d})^2}{n-1}} = \sqrt{\frac{\sum_{i=1}^{n}d_i^2-n\bar{d}^2}{n-1}}$$

โดยที่ d_i เป็นผลต่างระหว่างคะแนนก่อนและหลังของข้อมูลที่ได้จากตัวอย่าง

$\bar{d}$ เป็นค่าเฉลี่ยของผลต่างระหว่าง 2 กลุ่มทดลองของข้อมูลที่ได้จากตัวอย่าง

S_d เป็นค่าส่วนเบี่ยงเบนมาตรฐานของผลต่างของข้อมูลที่ได้จากตัวอย่าง

n เป็นขนาดตัวอย่าง

5. นำค่าที่ได้ไปเรียบเทียบกับค่า t จากตาราง ถ้าค่าที่คำนวณได้มากกว่าค่าจากตาราง แสดงว่าค่าเฉลี่ยของสองกลุ่มประชากรแตกต่างกันอย่างมีนัยสำคัญทางสถิติ หรือพิจารณาจากค่า

P-value ถ้าค่า P-value มีค่าน้อยกว่า α จะปฏิเสธสมมติฐานหลัก แสดงว่าค่าเฉลี่ยของสองกลุ่มประชากรแตกต่างกัน ถ้าเป็นการทดสอบแบบหางเดียวด้านขวา เมื่อ $t > t_{\alpha,n-1}$ จะปฏิเสธสมมติฐานหลัก และเมื่อ $t < t_{\alpha,n-1}$ จะยอมรับสมมติฐานหลัก และถ้าเป็นการทดสอบแบบหางเดียวด้านซ้าย จะปฏิเสธสมมติฐานหลัก เมื่อ $t < -t_{\alpha,n-1}$ และจะยอมรับสมมติฐานหลัก เมื่อ $t > -t_{\alpha,n-1}$

(จันทรัตน์, 2550)

4. การทดสอบกลุ่มตัวอย่าง 2 กลุ่มที่มีความสัมพันธ์กันหรือไม่เป็นอิสระกัน

4.1 การทดสอบของวิลคอกซันชนิดอันดับที่มีเครื่องหมาย (The Wilcoxon Matched-Pairs Signed - Rank Test)

นำมาประยุกต์ใช้กับข้อมูลคู่ โดยดูความแตกต่างทั้ง "ขนาด" ; $D_i = X_i - Y_i$ และ "เครื่องหมาย" ในแต่ละคู่ และพิจารณาโดยยึดหลักว่า ถ้าค่ามัธยฐานของประชากรคู่ทั้งสองไม่แตกต่างกันแล้ว ผลรวมของอันดับของความแตกต่างที่มีเครื่องหมาย + และ – น่าจะใกล้เคียงกัน ซึ่งการทดสอบนี้มักจะมีอำนาจการทดสอบมากกว่าการทดสอบเครื่องหมาย เนื่องจากมีการใช้ข้อมูลได้สมบูรณ์กว่า คือ ทราบถึงขนาดของความแตกต่าง ในขณะที่การทดสอบเครื่องหมายทราบเพียงว่าในแต่ละคู่ ค่า X_i มากกว่า Y_i หรือ X_i น้อยกว่า Y_i เท่านั้น

ข้อกำหนดเบื้องต้น

1. ค่า (X_i, Y_i) สำหรับทุกค่า i เป็นตัวแปรสุ่มแบบคู่ (Bivariate Random Variable)
2. ค่า D_i เป็นตัวแปรสุ่มต่อเนื่อง และเป็นอิสระต่อกัน
3. ค่า D_i มีการแจกแจงที่สมมาตร
4. ค่า D_i มีมาตราวัดอย่างน้อยแบบอันตรภาค (Interval Scale)

ข้อมูล

จากข้อมูล (X_i, Y_i) คู่มือ $(X_1,Y_1),(X_2,Y_2),\ldots,(X_n,Y_n)$ ในตัวอย่างแต่ละคู่จะเปรียบเทียบค่า X_i และ Y_i โดยการหาค่าผลต่างสมบูรณ์ คือ $|D_i| = |X_i - Y_i|$ เมื่อ $i = 1,2,\ldots,n'$ ในกรณีที่ D = 0 จะเรียกว่า Tie ให้ n = จำนวนคู่ที่ไม่รวม tie แล้ว จัดลำดับให้ค่า $|D_i|$ โดยให้ค่าน้อยที่สุดเป็นอันดับที่ 1,..., ค่ามากที่สุดเป็นอันดับที่ n

สมมติฐาน อาจจะใช้ทดสอบหางเดียวหรือสองหาง ได้ดังนี้

การทดสอบแบบสองหาง

H_0 : มัธยฐานของประชากร 2 กลุ่มไม่แตกต่างกัน หรือ $M_1 = M_2$

H_1 : มัธยฐานของประชากร 2 กลุ่มแตกต่างกัน หรือ $M_1 \neq M_2$

การทดสอบแบบหางเดียวด้านขวา

$H_0 : M_1 \leq M_2$ $\quad$ $H_1 : M_1 > M_2$

การทดสอบแบบหางเดียวด้านซ้าย

$H_0 : M_1 \geq M_2$ $\quad$ $H_1 : M_1 < M_2$

สถิติที่ใช้ทดสอบ

หลังจากหาค่าอันดับของ $|D_i|$ แล้ว ให้ใส่เครื่องหมายหน้าอันดับแต่ละอันดับ ด้วยเครื่องหมายเดิมของ D_i แล้วหาผลรวมของอันดับที่มีเครื่องหมาย + ผลรวมของอันดับที่มีเครื่องหมาย + = T^+

ตัวอย่างขนาดเล็ก ($n < 30$)

จะใช้ตารางแสดงค่าควอไทล์ ของค่าสถิติทดสอบของวิลคอกซัน ; W_α

การทดสอบสองหาง อาณาเขตวิกฤต คือ $T^+ > W_{1-\frac{\alpha}{2}}$ หรือ $T^+ < W_{\frac{\alpha}{2}}$

การทดสอบหางเดียวด้านขวา อาณาเขตวิกฤต คือ $T^+ > W_{1-\alpha}$

การทดสอบหางเดียวด้านซ้าย อาณาเขตวิกฤต คือ $T^+ < W_\alpha$

เขตปฏิเสธ

นำค่า T^+ ที่คำนวณได้ไปเปรียบเทียบกับค่า W_α ที่ได้จากตาราง **The Wilcoxon Matched - Pairs Signed - Rank Test** โดยในตารางจะกำหนดค่า n ที่แตกต่างกัน เริ่มจาก $W_{.005}...W_{.50}$ ส่วนค่า W_α ที่ $\alpha > 0.5$ ให้ใช้คุณสมบัติ ดังนี้

$$W_\alpha = \frac{n(n+1)}{2} - W_{1-\alpha}$$

เมื่อค่า $\frac{n(n+1)}{2}$ ได้แสดงไว้ทางขวามือของตาราง

นั่นคือ ถ้าเป็นการทดสอบแบบสองหาง เมื่อ $T^+ > W_{1-\frac{\alpha}{2}}$ หรือ $T^+ < W_{\frac{\alpha}{2}}$ จะปฏิเสธสมมติฐาน H_0 นั่นคือ มัธยฐานของประชากร 2 กลุ่มแตกต่างกัน และถ้า

$T^+ < W_{1-\frac{\alpha}{2}}$ หรือ $T^+ > W_{\frac{\alpha}{2}}$ จะยอมรับ H_0 นั่นคือ มัธยฐานของประชากร 2 กลุ่มไม่แตกต่างกัน ถ้าเป็นการทดสอบแบบหางเดียวด้านขวา นั่นคือ เมื่อ $T^+ > W_{1-\alpha}$ จะปฏิเสธสมมติฐาน H_0 และเมื่อ $T^+ < W_{1-\alpha}$ จะยอมรับ H_0 และถ้าเป็นการทดสอบแบบหางเดียวด้านซ้าย นั่นคือ เมื่อ $T^+ < W_\alpha$ จะปฏิเสธสมมติฐาน H_0 และเมื่อ $T^+ > W_\alpha$ จะยอมรับ H_0

(อุมาพร, 2542)

4.2 การทดสอบของแมคนีมาร์ (McNemar Test For Significance of Changes)

เป็นการทดสอบเพื่อดูการเปลี่ยนแปลงในรูป “ก่อนและหลัง” (Before and After) โดยใช้กับกลุ่มตัวอย่างชุดเดียว แต่ทำการทดลอง 2 ครั้ง ก่อนและหลังจากการใช้ทรีทเมนต์หนึ่ง เพื่อดูว่ามีการเปลี่ยนแปลงระหว่างก่อนและหลังการทดลองหรือไม่ เช่น

การทดสอบความคิดเห็นของคนกลุ่มหนึ่งก่อนและหลังฟังการอภิปราย

การทดสอบอาการเปลี่ยนแปลงของคนกลุ่มหนึ่งก่อนและหลังการใช้ยาชนิดหนึ่ง

การทดสอบความคิดเห็นของคนกลุ่มหนึ่งก่อนและหลังการอ่านบทความในหนังสือพิมพ์

ดังนั้นสามารถวัดประสิทธิภาพของทรีทเมนต์หนึ่ง ๆ (การอภิปราย , ยาชนิดหนึ่ง , บทความในหนังสือพิมพ์) ว่ามีผลทำให้เกิดการเปลี่ยนแปลงหรือไม่

ข้อกำหนดเบื้องต้น

1. ข้อมูลประกอบด้วยข้อมูลคู่ n คู่ และสามารถจัดลงตาราง 2×2ได้
2. มาตราวัดของข้อมูลเป็นแบบนามบัญญัติหรือแบบเรียงลำดับ ที่มีเพียง 2 กลุ่มย่อยเท่านั้น เช่น ใช่ กับ ไม่ใช่ หรือ + กับ –
3. ข้อมูลคู่แต่ละคู่เป็นอิสระกัน แต่ภายในคู่สัมพันธ์กัน

ข้อมูล

เช่น ต้องการทดสอบทรีทเมนต์ คือ การเต้นรำประกอบเพลงว่าจะมีผลต่อการลดน้ำหนักหรือไม่ สุ่มตัวอย่างคนที่ต้องการลดน้ำหนักด้วยวิธีนี้มา n คน ในขั้นแรกถามความคิดเห็นเกี่ยวกับผลที่คาดว่าจะได้รับก่อนจากการเต้นรำ ว่าจะได้ผลหรือไม่ได้ผล (+,-) และหลังจากนั้นทดลองให้เต้นรำประกอบเพลงไประยะเวลาหนึ่ง แล้วถามความคิดเห็นเช่นเดิมอีกครั้ง

สมมติฐาน อาจจะใช้ทดสอบหางเดียวหรือสองหาง ได้ดังนี้

การทดสอบแบบสองหาง

H_0 : $P(A) = P(D) = .5$ หรือมีการเปลี่ยนแปลงก่อนและหลังการให้ทรีทเมนต์ในจำนวนเท่า ๆ กัน

H_1 : $P(A) \neq P(D)$ หรือมีการเปลี่ยนแปลงก่อนและหลังการให้ทรีทเมนต์ในจำนวนที่แตกต่างกัน

การทดสอบแบบหางเดียวด้านขวา

$H_0 : P(A) \leq P(D)$ $H_1 : P(A) > P(D)$

นั่นคือ สงสัยว่ามีการเปลี่ยนแปลงจาก + เป็น – มีมากกว่าการเปลี่ยนแปลงจาก – เป็น +

การทดสอบแบบหางเดียวด้านซ้าย

$H_0 : P(A) \leq P(D)$ $H_1 : P(A) > P(D)$

นั่นคือ สงสัยว่ามีการเปลี่ยนแปลงจาก + เป็น – มีน้อยกว่าการเปลี่ยนแปลงจาก – เป็น +

วิธีการ

การหาผลสรุปของสมมุติฐานข้างต้นนี้ อาจดูได้ในลักษณะเปรียบเทียบความถี่ที่ได้จากการทดลองกับความถี่คาดหวัง (ภายใต้ H_0) ว่าแตกต่างกันหรือไม่

โดยความถี่คาดหวังตาม H_0 จะเป็นดังนี้ จำนวนความถี่ที่คาดหวังในช่อง A และ D ควรมีค่าเท่ากัน = $\frac{A+D}{2}$ และใช้การทดสอบไคสแควร์ได้ดังสูตร

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

และในกรณีนี้สนใจเฉพาะเซล A และ D

ฉะนั้น
$$\chi^2 = \frac{\left[A - \frac{(A+D)}{2}\right]^2}{\frac{A+D}{2}} + \frac{\left[D - \frac{(A+D)}{2}\right]^2}{\frac{A+D}{2}}$$

จะได้
$$\chi^2 = \frac{(A - D)^2}{A + D}$$

แต่เนื่องจาก χ^2 เป็นการแจกแจงแบบต่อเนื่อง จึงจำเป็นต้องปรับค่าต่อเนื่อง จะได้

$$\chi^2 = \frac{(|A - D| - 1)^2}{A + D} \quad \text{ด้วย d.f.} = 1$$

การหาอาณาเขตวิกฤต หาได้จากตาราง χ^2 ที่ d.f. = 1

เขตปฏิเสธ

นำค่า χ^2 ที่คำนวณได้มาเปรียบเทียบกับค่า χ^2 ที่ได้จากตาราง Chi-Square ที่ d.f. = 1 ตัวสถิติทดสอบนี้จะใช้ได้เฉพาะการทดสอบแบบสองหางเท่านั้น นั่นคือ ถ้า $\chi^2 > \chi^2_{\text{ตาราง}}$ จะปฏิเสธสมมติฐาน H_0 และถ้า $\chi^2 < \chi^2_{\text{ตาราง}}$ จะยอมรับ H_0 ส่วนการทดสอบแบบหางเดียวจะใช้การทดสอบเครื่องหมาย (Sign Test)

(อุมาพร, 2542)

5. การทดสอบกลุ่มตัวอย่าง k กลุ่มที่เป็นอิสระกัน

5.1 การทดสอบแบบครัสคาลและวอลลิส (The Kruskal-Wallis Oneway ANOVA by Rank Test)

ใช้ทดสอบประชากร k กลุ่ม ว่ามีค่ามัธยฐานเท่ากันหรือไม่ โดยมีวิธีการที่สำคัญคือ ค่าคาดหมายของลำดับที่ของข้อมูลตัวอย่างแต่ละกลุ่มควรมีค่าพอ ๆ กัน ข้อมูลที่นำมาทดสอบประกอบด้วยข้อมูลจากตัวอย่างสุ่ม k ชุด แต่ละชุดอาจมีขนาดตัวอย่างแตกต่างกัน และมีการแจกแจงต่อเนื่อง

การทดสอบนี้นิยมใช้แทนการทดสอบแบบเอฟ (F-Test) ในสถิติที่ใช้พารามิเตอร์ ในกรณีที่ข้อกำหนดเบื้องต้นของการทดสอบเอฟ ไม่เป็นจริง

สมมติฐาน

H_0 : มัธยฐานของประชากร k กลุ่มไม่แตกต่างกัน

H_1 : มัธยฐานของประชากร k กลุ่มแตกต่างกัน

ขั้นตอนในการทดสอบสมมติฐาน

1.) จัดลำดับของข้อมูลทั้งหมดรวมกันจากน้อยไปหามากโดยให้คะแนนต่ำสุดเป็นลำดับที่ 1 และคะแนนสูงสุดเป็นลำดับที่ n เมื่อ n เป็นจำนวนข้อมูลทั้งหมด

2.) หาผลรวมของลำดับที่ในข้อมูลแต่ละชุด คือ R_i , $i = 1,2,\ldots,k$

3.) คำนวณค่าสถิติ $$H = \frac{12}{n(n+1)}\sum_{i=1}^{k}\frac{R_i^2}{n_i} - 3(n+1)$$

เมื่อ H เป็น Kruskal-Wallis statistics ที่ต้องการทดสอบ

k เป็นจำนวนกลุ่มของข้อมูล

R_i เป็นผลรวมของลำดับที่ในตัวอย่างที่ i , i = 1,….,k

n_i เป็นจำนวนข้อมูลของกลุ่มตัวอย่างชุดที่ i (หรือขนาดตัวอย่างที่ i) , i = 1,….,k

n เป็นจำนวนข้อมูลทั้งหมด

ตามทฤษฎีจะพบว่า ถ้า H_0 เป็นจริง H จะมีการแจกแจงประมาณได้ด้วย χ^2 ที่ d.f. = k-1 ถ้า n_i มีค่าใหญ่พอสมควร

4.) การหาอาณาเขตวิกฤตและการสรุปผล สามารถแยกได้ตามขนาดตัวอย่าง คือ

4.1 เมื่อ $n_i > 5$ การแจกแจงของค่าสถิติ H ประมาณได้ด้วย χ^2 ที่ d.f. = k-1 เมื่อกำหนดระดับนัยสำคัญ = α หาอาณาเขตวิกฤตจากตาราง χ^2 ที่ d.f. = k-1 จะปฏิเสธ H_0 เมื่อค่าของ H ที่คำนวณได้มากกว่าหรือเท่ากับ χ^2 จากตาราง

4.2 เมื่อ k = 3, $n_i \leq 5$ ในแต่ละ k ใช้ตารางที่ Kruskal สร้างไว้โดยแสดงค่าวิกฤต H พร้อมทั้งความน่าจะเป็นที่จะเกิดค่า H นั้น ๆ ตารางที่ Kruskal สามารถจะทำการเปรียบเทียบค่า H หรือ P-value ก็ได้คือจะปฏิเสธ H_0 เมื่อค่า H ที่คำนวณได้มากกว่าหรือเท่ากับค่า H จากตารางที่ระดับนัยสำคัญ α

(อุมาพร, 2542)

5.2 การทดสอบแบบครัสคาลและวอลลิสแบบมีซ้ำมาก (The Kruskal-Wallis Test With Ties)

ในกรณีที่ค่าข้อมูลใน k กลุ่มอิสระกันนั้น มีค่าซ้ำกันมาก อาจจะซ้ำกันภายในกลุ่มหรือระหว่างกลุ่มก็ได้ เช่นมีข้อมูลดังนี้

จากวิทยาลัย 3 แห่ง สุ่มตัวอย่างคณาจารย์ เพื่อประเมินการบริหารงานของผู้บริหาร ใน 5 ระดับ คือ แย่ที่สุด ดี ดีมาก ยอดเยี่ยม โดยให้มีคะแนน คือ 1 – 5 ตามลำดับ ถ้าได้ข้อมูลด้วยขนาดตัวอย่างจาก 3 วิทยาลัย คือ 15, 10 และ 12 ดังนี้

วิทยาลัย A	1 1 1 1 1 1 2 2 2 2 3 3 3 4 4	n_A = 15
วิทยาลัย B	1 1 2 2 3 3 3 4 4 5	n_B = 10
วิทยาลัย C	1 2 2 3 3 3 4 4 4 4 5 5	n_C = 12

จากข้อมูลข้างต้นจะเห็นว่ามีค่าซ้ำกันมากในกลุ่มค่า 1 – 5 ดังนั้นอาจจัดข้อมูลลงตาราง 2 ทาง จะได้ดังนี้

วิทยาลัย	ผลการประเมิน					
	1 แย่ที่สุด	2 ปานกลาง	3 ดี	4 ดีมาก	5 ยอดเยี่ยม	
A	6	4	3	2	0	n_A = 15
B	2	2	3	2	1	n_B = 10
C	1	2	3	4	2	n_C = 12

จะพบว่าตัวแปรทางแถวตั้งมีลักษณะเรียงลำดับ จึงเรียกว่า ตารางแบบ Singly Ordered Table

ถ้าทำการวิเคราะห์ความสัมพันธ์ด้วยสถิติทดสอบ χ^2 จะได้ค่า $\chi^2_{cal} = 7.02$ ด้วย P – value = 0.53 ยอมรับสมมติฐานหลัก แต่ถ้าสังเกตจะพบว่าค่าความถี่ในเซลต่าง ๆ มีค่าน้อยมาก และมีค่า 0 รวมอยู่ด้วย ดังนั้นการสถิติ χ^2 ไม่น่าเหมาะสมสำหรับข้อมูลชุดนี้ (เนื่องจากจะเกิดความถี่คาดหวังที่น้อยกว่า 5 มีมากเกิน 20% ของจำนวนเซลทั้งหมด)

สามารถใช้สถิติของ **The Kruskal-Wallis Test With Ties** วิเคราะห์ข้อมูลชุดนี้แทน ซึ่งจะได้ H = 6.43 ด้วย P – value = 0.04 ซึ่งพบว่าปฏิเสธสมมติฐานหลัก (ในขณะที่ χ^2 ไม่พบความแตกต่าง)

(อุมาพร, 2542)

5.3 การทดสอบของไคสแควร์สำหรับกลุ่มตัวอย่าง k กลุ่มที่เป็นอิสระต่อกัน หรือ การทดสอบของ χ^2 เกี่ยวกับความเป็นเอกภาพของสัดส่วน (The χ^2 Test For k Independent Samples หรือ The χ^2 Test For Homogeneity of Proportions)

การทดสอบนี้ขยายมาจากการทดสอบของ χ^2 สำหรับ 2 กลุ่มตัวอย่างที่เป็นอิสระกัน ดังนั้นเหมาะที่จะใช้เมื่อตัวอย่างมีขนาดใหญ่

ข้อมูล

ใช้เมื่อข้อมูลคือความถี่ (Frequency) ที่ได้จากมาตรวัดแบบนามบัญญัติหรือเรียงลำดับ การทดสอบยังคงคล้ายกับการทดสอบของ χ^2 สำหรับกลุ่มตัวอย่าง 2 กลุ่มที่เป็นอิสระต่อกัน คือ ทำการเปรียบเทียบความถี่ที่สังเกตได้ (Observed Frequency) กับความถี่คาดหวัง (Expected Frequency) โดยข้อมูลจะจัดลงในตารางแจกแจง 2 ทางชนิด $r \times k$ เมื่อ k แถวตั้งและ r แถวนอน โดยที่การทดสอบนี้จะสุ่มตัวอย่างจาก k ประชากร ด้วยขนาดตัวอย่าง $n_1, n_2, \ldots, n_k$ ตามลำดับ

สมมติฐาน คือ สัดส่วนในแต่ละลักษณะย่อยในแต่ละประชากรจะเท่ากันหรือเป็นเอกภาพ นั่นคือ

H_0: ตัวอย่าง k กลุ่มมาจากประชากรเดียวกัน หรือสัดส่วนของลักษณะย่อยในประชากร k กลุ่ม ไม่แตกต่างกัน

H_1 : ตัวอย่าง k กลุ่มนี้มาจากประชากรที่ต่างกัน

สถิติที่ใช้ทดสอบ

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{k}\frac{(O_{ij}-E_{ij})^2}{E_{ij}}$$

เมื่อ O_{ij} เป็นความถี่จากการสังเกตที่ได้จากตารางแจกแจง 2 ทางที่แถวนอน i และแถวตั้ง j

E_{ij} เป็นความถี่คาดหวังที่ได้จากตารางแจกแจง 2 ทางที่แถวนอน i และแถวตั้ง j

ซึ่งการหาค่า E_{ij} ภายใต้ H_0 ยังคงคล้ายกับการทดสอบ χ^2 ในตัวอย่าง 2 กลุ่ม

ค่า E_{ij} ไม่ควรมีค่า < 5 เป็นจำนวนมากกว่า 20% ของจำนวนเซลทั้งหมด ถ้าเกิดเหตุการณ์เช่นนี้ อาจแก้ไขโดยการรวมกลุ่มที่ใกล้กันเข้าด้วยกัน ซึ่งจะทำให้ d.f. ลดลง ค่าสถิตินี้สามารถประมาณการแจกแจงได้ด้วย χ^2 ที่ d.f. = (k-1)(r-1) ดังนั้นการหาอาณาเขตวิกฤต ก็ทำการเปรียบเทียบค่า χ^2 ที่ได้จากสูตรนี้ กับค่า χ^2 จากตารางที่ระดับนัยสำคัญหนึ่ง

เขตปฏิเสธ

นำค่า χ^2 ที่คำนวณได้มาเปรียบเทียบกับค่า χ^2 ที่ได้จากตาราง Chi-Square Test ที่ d.f. = (k-1)(r-1) นั่นคือ ถ้า χ^2 ที่คำนวณได้มากกว่า χ^2 ที่ได้จากตารางจะปฏิเสธสมมติฐาน H_0 และถ้า χ^2 ที่คำนวณได้น้อยกว่า χ^2 ที่ได้จากตารางถึงจะยอมรับ H_0

(อุมาพร, 2542)

5.4 การทดสอบมัธยฐานกรณีหลายกลุ่มตัวอย่าง (The Extension of The Median Test)

เป็นการทดสอบที่ใช้หลักการเช่นเดียวกับการทดสอบมัธยฐานสำหรับ 2 กลุ่มตัวอย่างที่เป็นอิสระต่อกันนั่นเอง โดยข้อมูลที่นำมาวิเคราะห์ต้องมีมาตราวัดอย่างน้อยแบบเรียงลำดับ

ข้อกำหนดเบื้องต้น

1. ตัวอย่าง k ชุดเป็นตัวอย่างสุ่มด้วยขนาด n_i ซึ่งสุ่มมาจาก k ประชากรที่เป็นอิสระกัน $i=1,2,\ldots,k$
2. ค่าสังเกตเป็นอิสระกัน ทั้งภายในกลุ่มตัวอย่าง และระหว่างกลุ่มตัวอย่าง
3. ข้อมูลมีมาตราวัดอย่างน้อยแบบเรียงลำดับ
4. ถ้าทุกประชากรมีค่ามัธยฐานเท่ากัน ดังนั้นทุกประชากรจะมีค่าความน่าจะเป็นที่ค่าสังเกตจะมากกว่าค่ามัธยฐานรวม เท่ากัน = p

สมมติฐานที่ใช้ทดสอบ

คือ H_0 : กลุ่มตัวอย่าง k กลุ่มมาจากประชากรเดียวกัน (พิจารณาในแง่มัธยฐานเท่ากัน)

H_1 : กลุ่มตัวอย่าง k กลุ่มมาจากประชากรที่แตกต่างกัน

สถิติที่ใช้ทดสอบ

จะใช้ χ^2 เพราะสามารถทำการเปรียบเทียบความถี่ที่ได้จากการสังเกตกับความถี่ที่ควรจะเป็นตาม H_0

ดังนั้น
$$\chi^2=\sum_{i=1}^{2}\sum_{j=1}^{k}\frac{\left(O_{ij}-E_{ij}\right)^2}{E_{ij}}$$

เมื่อ O_{ij} เป็นความถี่จากการสังเกตที่ได้จากตารางแจกแจง 2 ทางที่แถวนอน i และแถวตั้ง j

E_{ij} เป็นความถี่ที่คาดว่าควรจะเป็นตาม H_0 จากตารางแจกแจง 2 ทางที่แถวนอน i และแถวตั้ง j

พบว่าการแจกแจงของค่า χ^2 นี้ภายใต้ H_0 สามารถประมาณได้ด้วยการแจกแจงแบบ χ^2 ที่ d.f. = (k-1)(2-1) = (k-1)

ดังนั้นการหาอาณาเขตวิกฤต จะหาได้จากโค้งการแจกแจงของ χ^2

เขตปฏิเสธ

นำค่า χ^2 ที่คำนวณได้มาเปรียบเทียบกับค่า χ^2 ที่ได้จากตาราง **Chi-Square** ที่ d.f. = (k-1)(2-1) = (k-1)นั่นคือ ถ้า χ^2 ที่คำนวณได้มากกว่า χ^2 ที่ได้จากตารางจะปฏิเสธสมมติฐาน H_0 และถ้า χ^2 ที่คำนวณได้น้อยกว่า χ^2 ที่ได้จากตารางถึงจะยอมรับ H_0

(อุมาพร, 2542)

2.2 งานวิจัยที่เกี่ยวข้อง

ได้ค้นคว้าและศึกษาถึงงานวิจัยที่เกี่ยวข้องกับปัญหาพิเศษฉบับนี้ แล้วพบว่ามีงานวิจัยที่เกี่ยวข้องและสามารถสรุปได้ดังนี้

กรกช นฤปิยะกุลและคณะ (2545) ได้เสนอเกมส์ 3 มิติ แบบออนไลน์ ซึ่งเป็นแนวเกมส์แบบ Action/Adventure โดยมีชื่อว่า เกมส์นักสู้แห่งฝัน สามารถเล่นได้หลายผู้เล่นผ่านทางระบบเครือข่าย และตัวละครกับฉากทำจากโปรแกรม Maya และโปรแกรม Bryee ส่วนตัวโปรแกรมเกมส์เขียนด้วย Visual C++ และ DirectX8 Software Developer Kit และได้พัฒนาท่าทางการเดินของตัวละครในขณะที่มีสิ่งกีดขวางอยู่ โดยใช้อัลกอริทึม A* เข้ามาช่วยในการแก้ปัญหา

ผลการทดสอบโปรแกรม สามารถเล่นผ่านระบบเครือข่ายได้ มีเสียงประกอบภาพของตัวละครในท่าทางการเดิน การโจมตี และรวมถึงมุมมองต่าง ๆ ในท่าทางนั้น ๆ ของการเดินของตัวละครในขณะที่มีสิ่งกีดขวางอยู่ พบว่า A* สามารถช่วยให้การดำเนินเรื่องของการเล่นเกมส์เป็นได้ดีในระดับหนึ่ง

ชนินทร์ ชูประเสริฐและคณะ (2543) ได้เสนอการหาเส้นทางเดินรถเมล์ที่เหมาะสมที่สุดในกรุงเทพมหานคร การค้นหาเส้นทางจะจำกัดเฉพาะภายในกรุงเทพมหานครเท่านั้น โดยแปลงกราฟให้เป็นโครงสร้างข้อมูลพื้นฐานในการเก็บเส้นทางเดินรถที่ผ่านสถานที่ต่าง ๆ จากนั้นนำอัลกอริทึม A* มาค้นหาเส้นทางที่เหมาะสมที่สุด โดยใช้ระยะทางเป็นเกณฑ์ในการค้นหาเส้นทาง สร้างโปรแกรมที่ทำการค้นหาเส้นทางผ่านทางเว็บ โดยการใส่จุดเริ่มต้นและจุดหมายในการเดินทาง โดยสามารถแสดงแผนที่ในส่วนที่เป็นจุดเริ่มต้นและยังสามารถขยายแผนที่บริเวณที่ต้องการให้ละเอียดมากขึ้นได้ด้วย

ผลการทดสอบโปรแกรม ได้ทำการแก้ไขฟังก์ชั่นการคิดน้ำหนัก เพื่อให้ได้ผลลัพธ์ที่ถูกต้อง พบว่าถ้าการเก็บข้อมูลมีความเที่ยงตรงแน่นอนเหมือนจริงมากที่สุดแค่ไหนก็ยิ่งสามารถค้นหาเส้นทางได้ถูกต้องยิ่งขึ้น ดังนั้นการตรวจสอบข้อบกพร่องของฐานข้อมูลจึงเป็นสิ่งสำคัญไม่ยิ่งหย่อนไปกว่าอัลกอริทึมที่ใช้ในการค้นหาเส้นทาง

สัญญา ควรคิด (2546) ได้เสนอวิธีแก้ปัญหาความผิดพลาดในการตัดสินใจเลือกเส้นทางบนทางด่วน โดยใช้ฐานข้อมูลแผนที่ร่วมกับป้ายจราจรเพื่อใช้หาเส้นทางด่วนในประเทศไทย และนำอัลกอริทึม A* มาค้นหาเส้นทางที่สั้นที่สุด ทำให้การเลือกเส้นทางมีความถูกต้องและเป็นเส้นทางที่มีระยะทางสั้นที่สุด และแปลงโครงสร้างของแผนที่ถนนจากแบบกราฟให้เป็นโครงสร้างแบบต้นไม้ เพื่อลดปัญหาการวนกลับNodeเดิมและง่ายต่อการค้นหาเส้นทาง กรณีถ้าเส้นทางถนนมีการเปลี่ยนแปลง ระบบจะทำการค้นหาเส้นทางใหม่ได้ และจากการตัดสินใจเลือกเส้นทางบนทางด่วนให้ถูกต้องนั้น ต้องนำข้อมูลจากป้ายแนะนำจราจรมาพิจารณาร่วมกับข้อมูลที่ได้จากป้ายแนะนำจราจร เพื่อเลือกเส้นทางไปยังทางลงทางด่วนที่กำหนดไว้

ผลการทดสอบจากท่าเรือไปยังดาวคะนอง โดยทดสอบประสิทธิภาพของอัลกอริทึมของ A* เปรียบเทียบกับอัลกอริทึมไดจ์สตาร์ พบว่า A* สามารถหาเส้นทางที่มีระยะทางสั้นที่สุดจากทางลงทางด่วนไปยังจุดหมายได้อย่างถูกต้องและใช้เวลาน้อยกว่าด้วย และระบบนี้สามารถให้คำตอบได้ถูกต้องร้อยละ 69.42

สุภาภรณ์ เกียรติสิน (2542) ได้เสนอวิธีการหาเส้นทางการส่งข้อมูลที่เสียค่าใช้จ่ายน้อยที่สุด (Minimum Cost-path) ที่มีประสิทธิภาพสูง เพื่อใช้หาเส้นทางที่ค่าน้ำหนักสามารถเปลี่ยนแปลงได้ โดยนำอัลกอริทึม A* มาประยุกต์ใช้ในการเลือกเส้นทางเฉพาะNodeที่ถูกพิจารณา แปลงความสัมพันธ์แบบกราฟให้เป็นต้นไม้ เพื่อป้องกันไม่ให้เกิดการการวนกลับโดยจะเริ่มค้นหาจากเส้นทางที่มีน้ำหนักน้อยสุดเป็นอย่างแรกแล้วจึงหาในระดับต่อ ๆ ไป และต้องเป็นเส้นทางที่มีผลรวมน้ำหนักน้อยที่สุดตลอดเส้นทาง เมื่อเปรียบเทียบกับเส้นทางอื่นที่ผ่าน ๆ มา
โดยการทำงานจะทำซ้ำเรื่อย ๆ จนพบปลายทาง ในกรณีที่เครือข่ายล่มได้นำเสนอกระบวนการย้อนกลับในการหาเส้นทางที่น้อยที่สุดเพื่อที่จะได้ไม่ต้องเริ่มคำนวณใหม่

ผลการทดสอบอัลกอริทึม A* เปรียบเทียบกับอัลกอริทึม Dijkstra พบว่า A* สามารถหาเส้นทางที่มีผลรวมน้ำหนักน้อยที่สุด โดยที่ไม่เกิดการวนกลับได้เร็วกว่า และในกรณีที่เครือข่ายล่มขาดการเชื่อมโยง A* ก็ยังเป็นอัลกอริทึมที่สามารถหาได้เร็วกว่าเหมือนเดิม ดังนั้นค่าฟังก์ชันวัตถุประสงค์ของระบบที่มีการเชื่อมโยงสมบูรณ์และระบบที่เกิดการล่มเมื่อมีการเชื่อมโยงจึงมีค่าแตกต่างกันเพียงร้อยละ 5

บทที่ 3
วิธีดำเนินงานวิจัย

ในการดำเนินการพัฒนาโปรแกรมการเดินของตัวละครในเกมส์ Catain Mandela โดยประยุกต์ใช้วิธี A* algorithm นั้น สามารถแบ่งการดำเนินงานออกเป็น 2 ส่วนหลัก ๆ คือ ส่วนพัฒนาโปรแกรมการเดินของตัวละครในเกมส์ Catain Mandela และส่วนการดำเนินงานวิจัยของการเปรียบเทียบความพึงพอใจของผู้เล่นทั้งก่อนและหลังของการพัฒนาเกมส์ โดยมีรายละเอียดส่วนต่าง ๆ ดังนี้

3.1 ขั้นตอนการดำเนินงาน

1. พัฒนาโปรแกรมการเดินของตัวละครในเกมส์ Catain Mandela โดยประยุกต์ใช้วิธี A* algorithm
2. เปรียบเทียบความพึงพอใจก่อนและหลังการปรับปรุงเกมส์จากกลุ่มตัวอย่างเยาวชน
3. ศึกษาวิธีการวิเคราะห์ทางสถิติที่จะต้องนำมาใช้ในการวิเคราะห์เปรียบเทียบ
4. สร้างแบบสอบถาม ผู้เล่นเกมส์ทำการทดสอบแบบสอบถามและปรับปรุงแก้ไขให้มีความถูกต้องเหมาะสม
5. เลือกกลุ่มตัวอย่างของเยาวชน โดยทำการสุ่มตัวอย่างในรูปแบบที่มีความถูกต้องเหมาะสม
6. เก็บรวบรวมข้อมูลจากกลุ่มตัวอย่างของเยาวชนผู้เล่นเกมส์
7. วิเคราะห์ข้อมูล โดยใช้โปรแกรมสำเร็จรูปทางสถิติเข้ามาช่วยในการประมวลผลข้อมูล ซึ่งจะทำการประมวลผลข้อมูลตามสมมติฐานของการศึกษาที่ตั้งไว้
8. แปลความหมาย สรุปผลการวิเคราะห์ข้อมูลทางสถิติ ข้อเสนอแนะ รวมถึงแนวทางการแก้ไขปัญหาดังกล่าวโดยจะทำการสรุปผลตามสมมติฐานของการศึกษาที่ตั้งไว้
9. จัดทำรูปแบบในการนำเสนอเกี่ยวกับข้อแนะนำรวมถึงแนวทางในการแก้ไขปัญหาดังกล่าว

3.2 ขั้นตอนการพัฒนาโปรแกรม

1. ติดต่อบริษัท Debuz Co.,Ltd. เพื่อขออนุญาตใช้ Debuz Game Engine เพื่อการทำวิจัยทางด้านระบบการเดินของตัวละครในเกมส์
2. ศึกษา Debuz Game Engine ถึงระบบการทำงานเดิมที่มีอยู่ เครื่องมือ และคำสั่งต่างๆที่ใช้บน Game Engine ตัวนี้
3. ศึกษารายละเอียดระบบการเดินของ Debuz Game Engine ว่ามีการทำงานอย่างไร
4. ศึกษาถึงการรักษาความปลอดภัย บนการทำงานแบบเข้าจังหวะ
5. ได้วางแผนการพัฒนาเป็น 2 ส่วนหลัก ๆ ได้แก่ พัฒนา Client และ Server ซึ่งจะทำการพัฒนา 2 ส่วนนี้ คือ

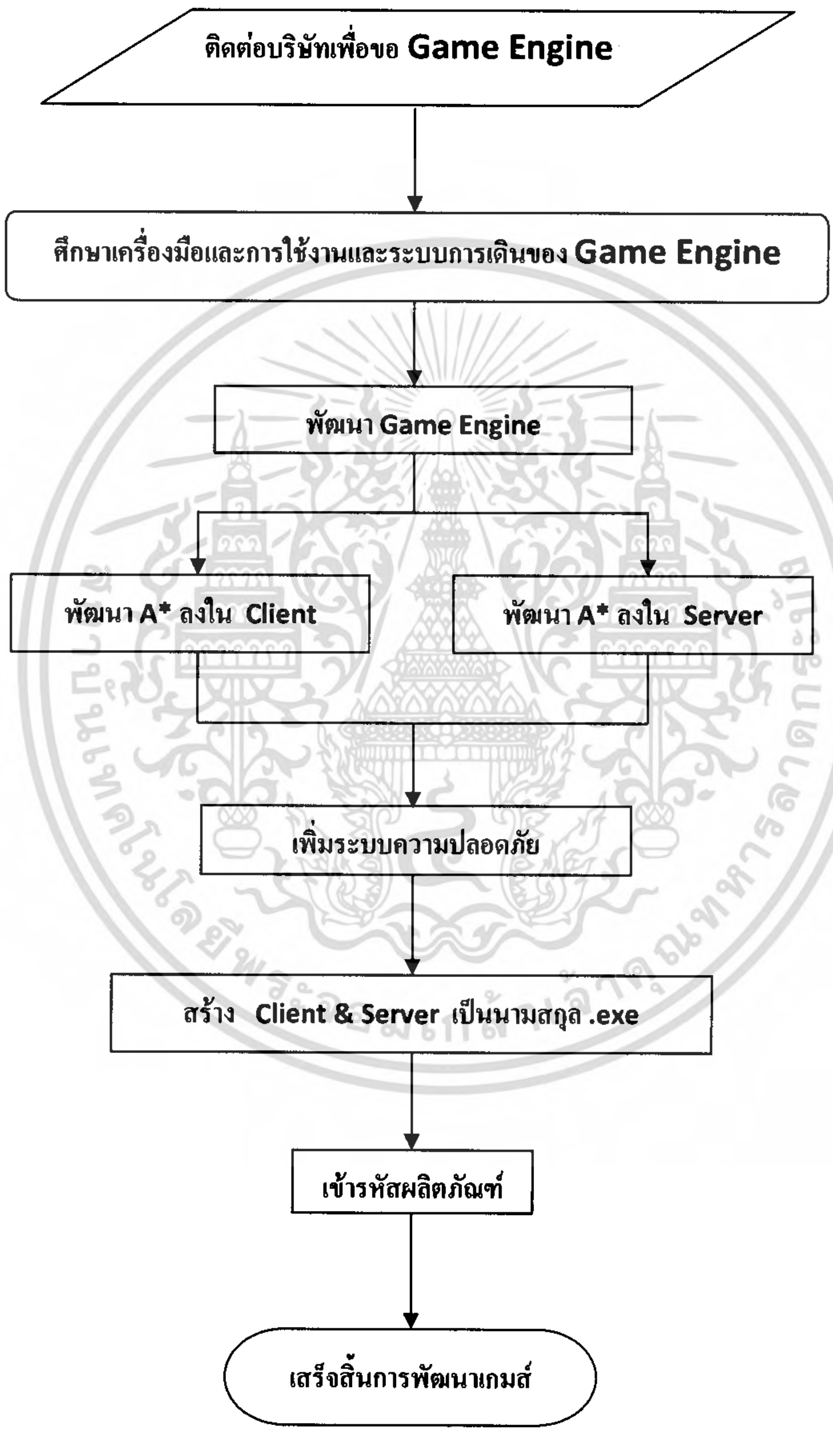
   5.1 การพัฒนา A* ลงในระบบ Server ซึ่งได้แสดง Source Code ไว้(ในภาคผนวก ก)

   5.2 การพัฒนา A* ลงในระบบ Client ซึ่งได้แสดง Source Code ไว้(ในภาคผนวก ก)

   โดยหลักการจะให้ระบบ Server คำนวณเส้นทางการเดินให้และส่งข้อมูลไปที่ Client ให้ทำการแสดงผล โดยทั้ง Client และ Server จะทำงานแบบเข้าจังหวะกัน
6. ทำการเข้ารหัสตัวเกมส์ และสร้างตัวเกมส์ที่สามารถทำงานได้อย่างสมบูรณ์

ขั้นตอนที่กล่าวมาสามารถสรุปเป็นรูปภาพดังต่อไปนี้

ผังงานการพัฒนา A* ลงใน Game Engine



3.3 ขนาดตัวอย่าง

ในการศึกษาปัญหาพิเศษครั้งนี้ได้เลือกใช้ตัวอย่างขนาดเล็ก เนื่องจากทางคณะผู้จัดมีข้อจำกัดในเรื่องของระยะเวลาในการทำปัญหาพิเศษ และยังมีเครื่องคอมพิวเตอร์จำกัดในการรองรับการเล่นเกมส์ของก่อนและหลังการปรับปรุงอีกด้วย เพราะฉะนั้นจำนวนที่ใช้เป็นขนาดตัวอย่าง คือ 20 คน โดยได้แบ่งตาม 4 ช่วงอายุ ดังนี้

ประถมศึกษา	(ช่วงอายุ 7 - 12 ปี)	จำนวน 5 คน
มัธยมศึกษาตอนต้น	(ช่วงอายุ 13 - 15 ปี)	จำนวน 5 คน
มัธยมศึกษาตอนปลาย	(ช่วงอายุ 16 - 18 ปี)	จำนวน 5 คน
ปริญญาตรี	(19 ปีขึ้นไป)	จำนวน 5 คน

3.4 เครื่องมือที่ใช้ในการเก็บรวบรวมข้อมูล

เครื่องมือที่ใช้ในการเก็บรวบรวมข้อมูลเป็นแบบสอบถามที่สร้างขึ้นดังแสดง (ในภาคผนวก ข) โดยมีโครงสร้างดังนี้

<u>ตอนที่ 1</u> ข้อมูลส่วนตัวของผู้ตอบแบบสอบถาม

<u>ตอนที่ 2</u> ความพึงพอใจของกลุ่มตัวอย่างที่มีต่อการเล่นเกมส์

<u>ตอนที่ 3</u> ความคิดเห็นและข้อเสนอแนะอื่น ๆ ที่มีต่อการเล่นเกมส์ มีลักษณะเป็นมาตรวัดเจตคติ สำหรับการทำปัญหาพิเศษครั้งนี้ได้ใช้มาตรวัดแบบลิเคอร์ท โดยกำหนดคำตอบเป็น 5 คำตอบ คือ มากที่สุด มาก ปานกลาง น้อย น้อยที่สุด

การให้คะแนนตัวแปรเกี่ยวกับความคิดเห็นแก่ข้อความนั้นจะยึดเนื้อความของข้อความเป็นหลัก โดยเกณฑ์การให้คะแนนเป็นดังนี้

น้อยที่สุด	ให้คะแนนเป็น	1
น้อย	ให้คะแนนเป็น	2
ปานกลาง	ให้คะแนนเป็น	3
มาก	ให้คะแนนเป็น	4
มากที่สุด	ให้คะแนนเป็น	5

<u>การให้คะแนนด้านทัศนคติที่มีต่อเกมส์ก่อนการปรับปรุงและหลังการปรับปรุง</u>

เนื่องจากแบบการวัดความพึงพอใจในแบบสอบถามมีข้อความทั้งหมด 16 ข้อของชุดก่อนและหลังการปรับปรุงเกมส์ ผู้ตอบมีคะแนนสูงสุดเท่ากับ 4.30 คะแนนต่ำสุดเท่ากับ 2.60 โดยกำหนดให้เกณฑ์การแบ่งคะแนนที่จะแสดงถึงระดับคะแนนความพึงพอใจโดยรวม (16 ข้อ) ดังนี้

ไม่เห็นด้วยอย่างยิ่ง	ระดับคะแนนระหว่าง	0 - 1
ไม่เห็นด้วย	ระดับคะแนนระหว่าง	1 - 2
ปานกลาง	ระดับคะแนนระหว่าง	2 - 3
เห็นด้วย	ระดับคะแนนระหว่าง	3 - 4
เห็นด้วยอย่างยิ่ง	ระดับคะแนนระหว่าง	4 - 5

(อุมาพร จันทศรและคณะ, 2545)

3.5 สถิติที่ใช้ในการวิเคราะห์ข้อมูล

สถิติที่ใช้ในการทดสอบสมมติฐาน ได้แก่

1. การทดสอบของ ชาพิโร - วิลค์ (Shapiro – Wilk Test) ใช้ในการวิเคราะห์การแจกแจงของข้อมูลของก่อนและหลังการปรับปรุงเกมส์ กับสมมติฐานข้อที่ 1 - 16

2. การทดสอบของวิลคอกซันชนิดอันดับที่มีเครื่องหมาย (The Wilcoxon Matched-Pairs Signed - Rank Test) ใช้ในการวิเคราะห์ความพึงพอใจของก่อนและหลังการปรับปรุงเกมส์ กับสมมติฐานข้อที่ 17 - 24

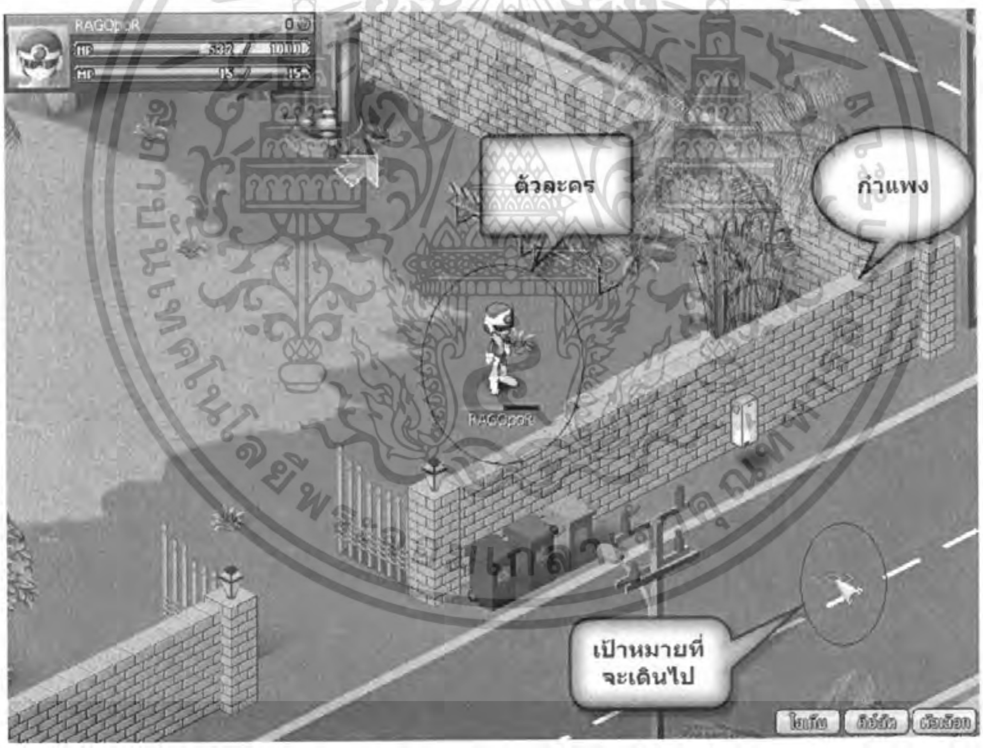
3. การทดสอบแบบครัสคาลและวอลลิส (The Kruskal-Wallis Oneway ANOVA by Rank Test) ใช้ในการวิเคราะห์ความแตกต่างของแต่ละระดับการศึกษาของก่อนและหลังการปรับปรุงเกมส์ กับสมมติฐานข้อที่ 25

บทที่ 4
ผลการวิเคราะห์ข้อมูล

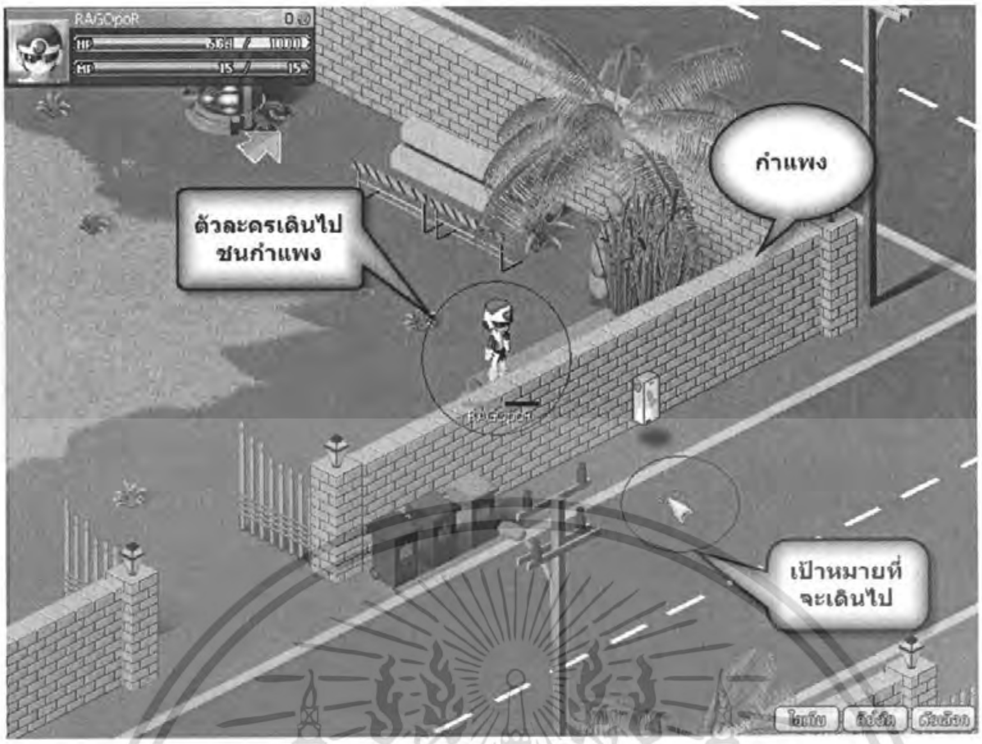
หลักจากที่ได้ทำการศึกษาทฤษฎี A* และทฤษฎีที่เกี่ยวข้องแล้ว จึงได้ดำเนินการพัฒนา A* ลงบน Game Engine ในส่วนของ Server และ Client และผลที่ได้จากการพัฒนาระบบการเดิน โดยจะเริ่มอธิบายในส่วนของการทำงานของระบบการเดินที่มีอยู่ก่อนการพัฒนาระบบ A* เปรียบเทียบกับระบบการเดินหลังการพัฒนาระบบ A* ดังนี้

4.1 ผลการพัฒนาโปรแกรม

4.1.1 โปรแกรมเกมส์ Captain Mandela ก่อนการปรับปรุง



รูปที่ 4.1 การเดินของตัวละครก่อนพัฒนาระบบ A*

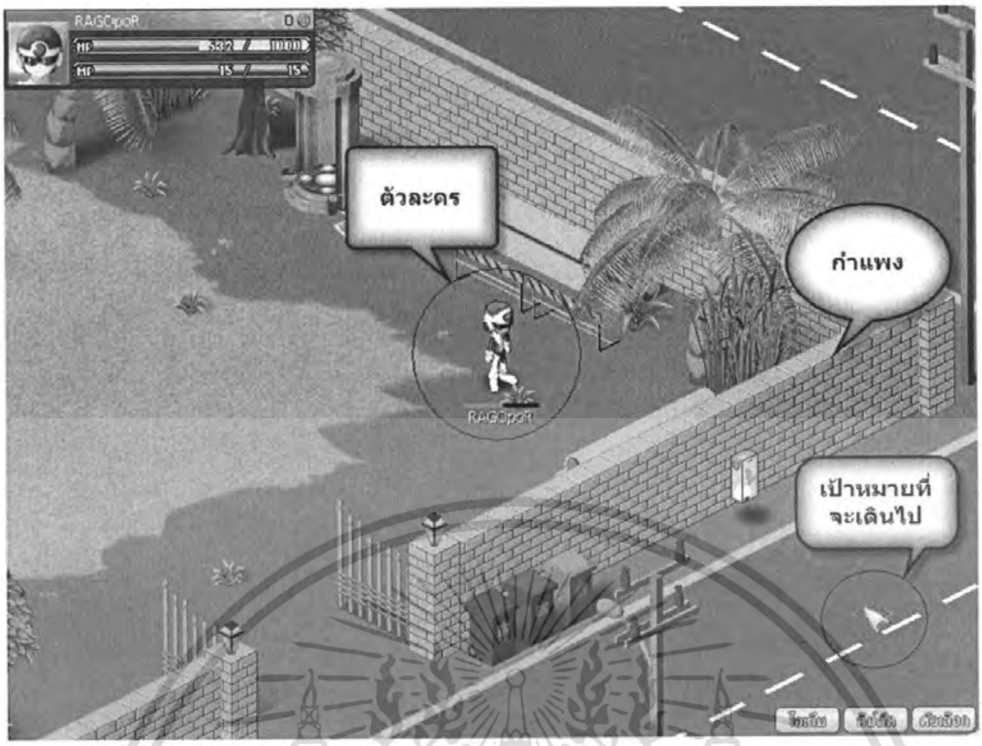


รูปที่ 4.2 การเดินของตัวละครเดิน ไปชนที่กำแพง

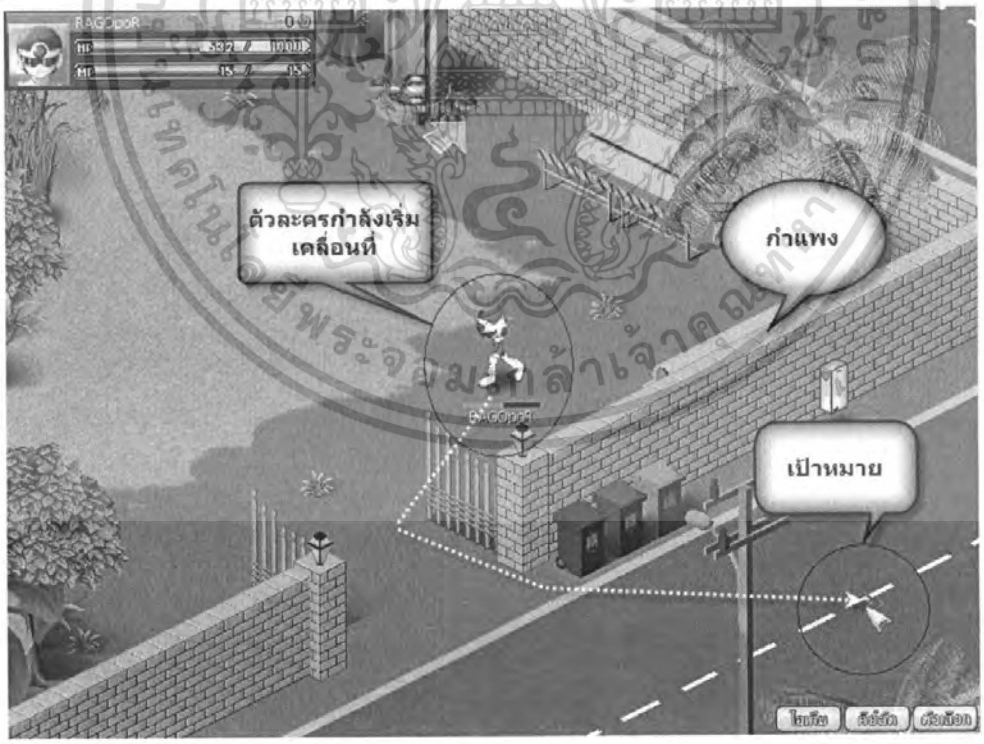
จากรูปที่ 4.1 จะเห็นได้ว่า เมื่อทำการคลิก 1 ครั้งเพื่อให้ตัวละครเดินไปยังเป้าหมาย แต่ตัวละครจะเดินไปติดที่กำแพง ไม่สามารถเดินไปยังเป้าหมายได้ เพราะระบบก่อนพัฒนา จะเป็นการเคลื่อนที่แบบเส้นตรง เมื่อตัวละครเดินไปชนวัตถุต่าง ๆ ก็จะติดไม่สามารถเดินต่อได้ ดังรูปที่ 4.2

4.1.2 โปรแกรมเกมส์ Captain Mandela หลังการปรับปรุง

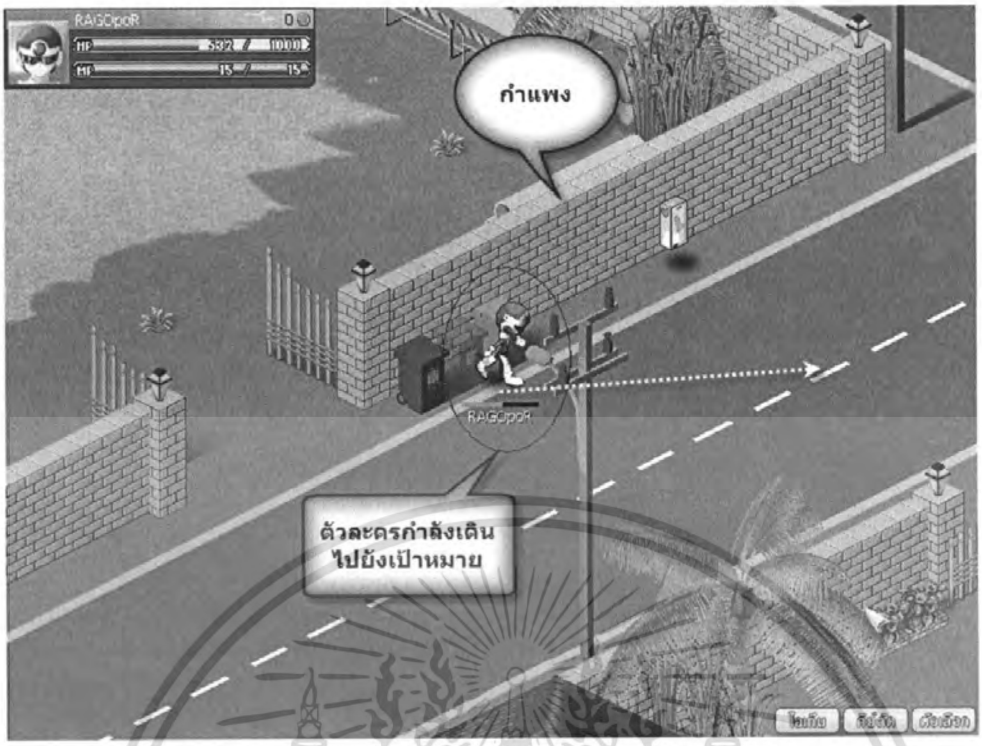
เมื่อทำการคลิก 1 ครั้งเพื่อให้ตัวละครเดินไปที่เป้าหมาย ตัวละครก็ทำการเดินอ้อมกำแพง และเดินไปถึงเป้าหมาย ได้อย่างรวดเร็วดังรูปที่ 4.2



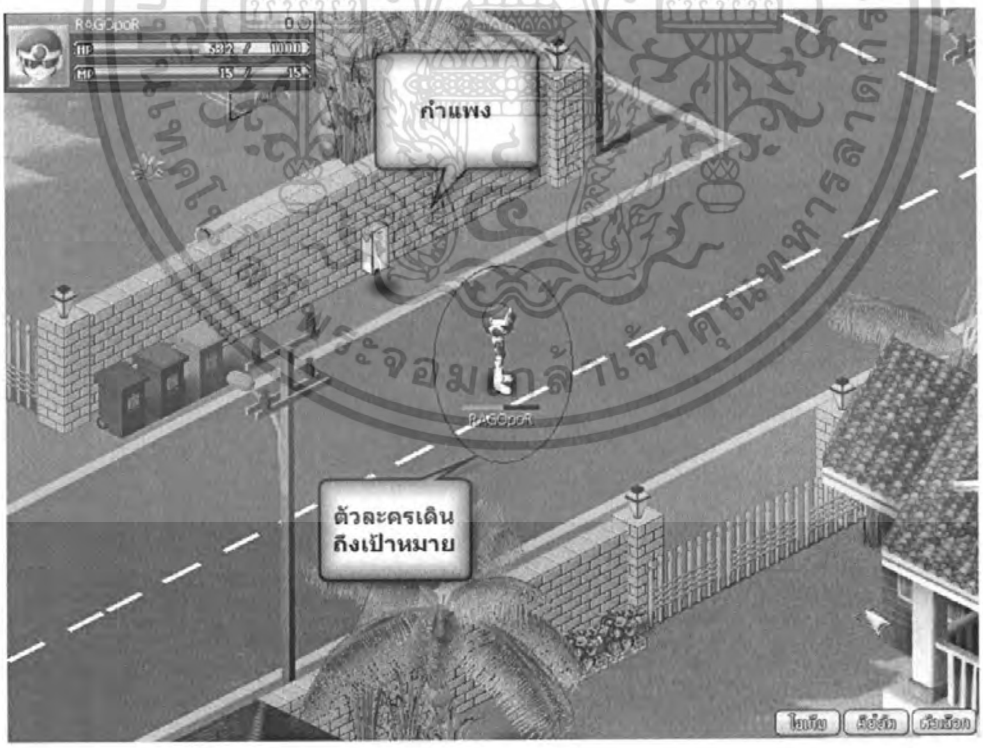
รูปที่ 4.3 การเดินของตัวละครหลังพัฒนาระบบ A*



รูปที่ 4.4 การทำงานของ A* ตัวละครกำลังเริ่มเคลื่อนที่ไปยังเป้าหมาย



รูปที่ 4.5 การทำงานของ A* ตัวละครกำลังเดินไปยังเป้าหมาย



รูปที่ 4.6 การทำงานของ A* เมื่อถึงเป้าหมาย

4.2 ส่วนแบบสอบถาม

จากข้อมูลการตอบแบบสอบถามเกี่ยวกับการวัดระดับความพึงพอใจของผู้เล่นที่มีต่อเกมส์ Captain Mandela ทั้งก่อนและหลังการปรับปรุงโดยมีผู้ตอบแบบสอบถามเป็นคน ๆ เดียวกัน ผู้วิจัยได้เก็บรวบรวมข้อมูลจากกลุ่มเยาวชนโดยแบ่งตามช่วงอายุ ได้สุ่มตัวอย่างเป็นจำนวน 20 คน แล้วทำการวิเคราะห์ข้อมูลโดยใช้โปรแกรมสำเร็จรูปทางสถิติ SPSS for Windows Version 14.0 ผลการวิเคราะห์เป็นดังนี้

4.3 ข้อมูลทั่วไป

ตารางที่ 4.1 จำนวนและร้อยละของตัวอย่างผู้ตอบแบบสอบถาม จำแนกตามเพศ

เพศ	จำนวนผู้ตอบแบบสอบถาม	ร้อยละ
ชาย	12	60
หญิง	8	40
รวม	20	100

จากตารางที่ 4.1 เมื่อจำแนกตัวอย่างผู้ตอบแบบสอบถามตามเพศ พบว่าตัวอย่างผู้ตอบแบบสอบถามรวมทั้งหมด 20 คน เป็นเพศชาย จำนวน 12 คน คิดเป็นร้อยละ 60 เป็นเพศหญิง จำนวน 8 คน คิดเป็นร้อยละ 40

ตารางที่ 4.2 จำนวนและร้อยละของตัวอย่างผู้ตอบแบบสอบถาม จำแนกตามอายุและเพศ

ข้อมูล	ชาย		หญิง		รวม	
	จำนวน	ร้อยละ	จำนวน	ร้อยละ	จำนวน	ร้อยละ
ช่วงอายุ 7-12 ปี	3	60	2	40	5	100
ช่วงอายุ 13-15 ปี	4	80	1	20	5	100
ช่วงอายุ 16-18 ปี	2	40	3	60	5	100
มากกว่า 19 ปี	3	60	2	40	5	100
รวม	12	60	8	40	20	100

จากตารางที่ 4.2 เมื่อจำแนกตัวอย่างผู้ตอบแบบสอบถามตามอายุ พบว่าตัวอย่างผู้ตอบแบบสอบถามรวมทั้งหมด 20 คน เป็นผู้ตอบแบบสอบถามช่วงอายุ 7-12 ปี , 13-15 ปี , 16-18 ปี และมากกว่า 19 ปี จำนวนอย่างละ 5 คน เท่ากัน

เมื่อพิจารณาเฉพาะผู้ตอบแบบสอบถามเพศชาย พบว่า เป็นผู้ตอบแบบสอบถามช่วงอายุ 7-12 ปี จำนวน 3 คน คิดเป็นร้อยละ 60 เป็นผู้ตอบแบบสอบถามช่วงอายุ 13-15 ปี จำนวน 4 คน คิดเป็นร้อยละ 80 เป็นผู้ตอบแบบสอบถามช่วงอายุ 16-18 ปี จำนวน 2 คน คิดเป็นร้อยละ 40 เป็นผู้ตอบแบบสอบถามช่วงอายุมากกว่า 19 ปี จำนวน 3 คน คิดเป็นร้อยละ 60

เมื่อพิจารณาเฉพาะผู้ตอบแบบสอบถามเพศหญิง พบว่า เป็นผู้ตอบแบบสอบถามช่วงอายุ 7-12 ปี จำนวน 2 คน คิดเป็นร้อยละ 40 เป็นผู้ตอบแบบสอบถามช่วงอายุ 13-15 ปี จำนวน 1 คน คิดเป็นร้อยละ 20 เป็นผู้ตอบแบบสอบถามช่วงอายุ 16-18 ปี จำนวน 3 คน คิดเป็นร้อยละ 60 เป็นผู้คอบแบบสอบถามช่วงอายุมากกว่า 19 ปี จำนวน 2 คน คิดเป็นร้อยละ 40

ตารางที่ 4.3 ค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของผู้ตอบแบบสอบถาม จำแนกตามระดับการศึกษาและอายุ

ระดับการศึกษา	ค่าเฉลี่ย (อายุ)	ส่วยเบี่ยงเบนมาตรฐาน (อายุ)
ประถมศึกษา	10.6	0.3
มัธยมตอนต้น	14.8	0.2
มัธยมตอนปลาย	18	0.5
อุดมศึกษา	21.8	0.7
รวม	65.2	1.7

จากตารางที่ 4.3 การวิเคราะห์ผลผู้ตอบแบบสอบถามทั้งหมด 20 คน เป็นประถมศึกษาที่อยู่ในช่วงอายุ 7-12 ปี จำนวน 5 คน โดยมีอายุเฉลี่ย 10.6 ปี และส่วนเบี่ยงเบนมาตรฐาน 0.3 ปี เป็นมัธยมตอนต้นที่อยู่ในช่วงอายุ 13-15 ปี จำนวน 5 คน โดยมีอายุเฉลี่ย 14.8 ปี และส่วนเบี่ยงเบนมาตรฐาน 0.2 ปี เป็นมัธยมตอนปลายที่อยู่ในช่วงอายุ 16-18 ปี จำนวน 5 คน โดยมีอายุเฉลี่ย 18 ปี และส่วนเบี่ยงเบนมาตรฐาน 0.5 ปี เป็นอุดมศึกษาที่อยู่ในช่วงอายุ 19 ปีขึ้นไป จำนวน 5 คน โดยมีอายุเฉลี่ย 21.8 ปี และส่วนเบี่ยงเบนมาตรฐาน 0.7 ปี

ตารางที่ 4.4 จำนวน (n) ร้อยละ ค่าเฉลี่ย ($\bar{x}$) ส่วนเบี่ยงเบนมาตรฐาน (S.D.) และภาพรวมความพึงพอใจของผู้ตอบแบบสอบถาม ของการเล่นเกมส์ก่อนการปรับปรุง

คำถาม	ระดับความคิดเห็น					n (ร้อยละ)	$\bar{X}$ (s_d)	ภาพรวมความพึงพอใจ
	มากที่สุด	มาก	ปานกลาง	น้อย	น้อยที่สุด			
1.ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน	-	4 (20)	**11 (55)**	5 (25)	-	20 (100)	2.95 (0.686)	ปานกลาง
2.รูปแบบการเดินของตัวละครผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด	-	2 (10)	**9 (45)**	8 (40)	1 (5)	20 (100)	2.60 (0.754)	ปานกลาง
3.ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน	2 (10)	**10 (50)**	7 (35)	1 (5)	-	20 (100)	3.65 (0.745)	เห็นด้วย
4.ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด	-	3 (15)	**11 (55)**	3 (15)	3 (15)	20 (100)	2.70 (0.923)	ปานกลาง
5.ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด	-	2 (10)	**15 (75)**	3 (15)	-	20 (100)	2.95 (0.510)	ปานกลาง
6.ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้ (เกี่ยวกับโรคเอดส์)ได้มากขึ้น	4 (20)	**14 (70)**	2 (10)	-	-	20 (100)	4.10 (0.553)	เห็นด้วยอย่างยิ่ง
7.ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน การโจมตีในเกมส์มากน้อยเพียงใด	1 (5)	5 (25)	**12 (60)**	2 (10)	-	20 (100)	3.25 (0.716)	เห็นด้วย
8.ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด	1 (5)	3 (15)	**7 (35)**	8 (40)	1 (5)	20 (100)	2.75 (0.967)	ปานกลาง

จากตารางที่ 4.4 เป็นการสำรวจความคิดเห็นด้านระดับความพึงพอใจของการเล่นเกมส์ก่อนการปรับปรุง ผลการสำรวจในแต่ละประเด็นเป็นดังนี้

ในหัวข้อ ตามความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ที่ ระดับปานกลาง คิดเป็นร้อยละ 55 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับปานกลาง (ค่าเฉลี่ย เท่ากับ 2.95)

ในหัวข้อ รูปแบบการเดินของตัวละครผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับปานกลาง คิดเป็นร้อยละ 45 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับปานกลาง (ค่าเฉลี่ย เท่ากับ 2.60)

ในหัวข้อ ในความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับมาก คิดเป็นร้อยละ 50 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับเห็นด้วย (ค่าเฉลี่ย เท่ากับ 3.65)

ในหัวข้อ ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับปานกลาง คิดเป็นร้อยละ 55 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับปานกลาง (ค่าเฉลี่ย เท่ากับ 2.70)

ในหัวข้อ ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับปานกลาง คิดเป็นร้อยละ 75 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อด้านนี้อยู่ในระดับปานกลาง (ค่าเฉลี่ย เท่ากับ 2.95)

ในหัวข้อ ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับมาก คิดเป็นร้อยละ 70 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับเห็นด้วยอย่างยิ่ง (ค่าเฉลี่ย เท่ากับ 4.10)

ในหัวข้อ ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตีในเกมส์มากน้อยเพียงใด พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับปานกลาง คิดเป็นร้อยละ 60 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับเห็นด้วย (ค่าเฉลี่ย เท่ากับ 3.25)

ในหัวข้อ ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับปานกลาง คิดเป็นร้อยละ 35 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับปานกลาง (ค่าเฉลี่ย เท่ากับ 3.75)

ตารางที่ 4.5 จำนวน (n) ร้อยละ ค่าเฉลี่ย($\bar{x}$) ส่วนเบี่ยงเบนมาตรฐาน (S.D.) และภาพรวมความพึงพอใจของผู้ตอบแบบสอบถาม ของการเล่นเกมส์หลังการปรับปรุง

คำถาม	ระดับความคิดเห็น					n	$\bar{X}$	ภาพรวม
	มากที่สุด	มาก	ปานกลาง	น้อย	น้อยที่สุด	(ร้อยละ)	(s_d)	ความพึงพอใจ
1.ตามความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน	-	7 (35)	**12 (60)**	1 (5)	-	20 (100)	3.30 (0.571)	เห็นด้วย
2.รูปแบบการเดินของตัวละคร (หลังการปรับปรุง) ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด	4 (20)	**12 (60)**	4 (20)	-	-	20 (100)	4.00 (0.649)	เห็นด้วย
3.ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน	4 (20)	**8 (40)**	7 (35)	1 (5)	-	20 (100)	3.75 (0.851)	เห็นด้วย
4.ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด	-	4 (20)	**12 (60)**	4 (20)	-	20 (100)	3.00 (0.649)	ปานกลาง
5.ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด	1 (5)	**11 (55)**	8 (40)	-	-	20 (100)	3.65 (0.587)	เห็นด้วย
6.ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้ (เกี่ยวกับโรคเอดส์)ได้มากขึ้น	7 (35)	**12 (60)**	-	1 (5)	-	20 (100)	4.25 (0.716)	เห็นด้วยอย่างยิ่ง
7.ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตีในเกมส์มากน้อยเพียงใด	2 (10)	**10 (50)**	6 (30)	2 (10)	-	20 (100)	3.60 (0.821)	เห็นด้วย
8.ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด (หลังการปรับปรุง)	**9 (45)**	**9 (45)**	1 (5)	1 (5)	-	20 (100)	4.30 (0.801)	เห็นด้วยอย่างยิ่ง

จากตารางที่ 4.5 เป็นการสำรวจความคิดเห็นด้านระดับความพึงพอใจของการเล่นเกมส์หลังการปรับปรุง ผลการสำรวจในแต่ละประเด็นเป็นดังนี้

ในหัวข้อ ตามความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ที่ระดับปานกลาง คิดเป็นร้อยละ 60 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับเห็นด้วย (ค่าเฉลี่ย เท่ากับ 3.30)

ในหัวข้อ รูปแบบการเดินของตัวละคร(หลังการปรับปรุง) ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับมาก คิดเป็นร้อยละ 60 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับเห็นด้วย (ค่าเฉลี่ย เท่ากับ 4.00)

ในหัวข้อ ในความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับ มาก คิดเป็นร้อยละ 40 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับเห็นด้วย (ค่าเฉลี่ย เท่ากับ 3.75)

ในหัวข้อ ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับปานกลาง คิดเป็นร้อยละ 60 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับปานกลาง (ค่าเฉลี่ย เท่ากับ 3.00)

ในหัวข้อ ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับมาก คิดเป็นร้อยละ 55 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับเห็นด้วย (ค่าเฉลี่ย เท่ากับ 3.65)

ในหัวข้อ ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับมาก คิดเป็นร้อยละ 60 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับเห็นด้วยอย่างยิ่ง (ค่าเฉลี่ย เท่ากับ 4.25)

ในหัวข้อ ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน, การโจมตีในเกมส์มากน้อยเพียงใด พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับมาก คิดเป็นร้อยละ 50 และโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับเห็นด้วย (ค่าเฉลี่ย เท่ากับ 3.60)

ในหัวข้อ ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด(หลังการปรับปรุง) พบว่าผู้ตอบแบบสอบถามโดยรวมส่วนใหญ่มีระดับความคิดเห็นต่อหัวข้อนี้อยู่ในระดับมากที่สุดและมากเท่ากัน จึงคิดเป็นร้อยละ 45 เหมือนกันและโดยเฉลี่ยแล้วผู้ตอบมีความพึงพอใจต่อหัวข้อนี้อยู่ในระดับเห็นด้วยอย่างยิ่ง (ค่าเฉลี่ย เท่ากับ 4.30)

4.4 ผลการทดสอบสมมติฐาน

จะทำการทดสอบสมมติฐานของการวิจัยในหัวข้อ 1.3 ดังนี้ โดยจะกำหนดระดับนัยสำคัญ (α) เท่ากับ 0.05

สมมติฐานที่ 1

H_0: ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน ของก่อนการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 2

H_0: รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 3

H_0: ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชนของก่อนการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชนของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 4

H_0: ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใดของก่อนการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใดของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 5

H_0: ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใดของก่อนการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใดของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 6

H_0: ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น ของก่อนการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 7

H_0: ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตีในเกมส์มากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตีในเกมส์มากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 8

H_0: ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด ของก่อนการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 9

H_0: ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน ของหลังการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 10

H_0: รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 11

H_0: ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน ของหลังการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 12

H_0: ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 13

H_0: ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 14

H_0: ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น ของหลังการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 15

H_0: ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตีในเกมส์มากน้อยเพียงใด ของหลังการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตีในเกมส์มากน้อยเพียงใด ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 16

H_0: ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์มีการแจกแจงแบบปกติ

H_1: ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด ของหลังการปรับปรุงเกมส์ไม่ได้มีการแจกแจงแบบปกติ

สมมติฐานที่ 17

H_0: มัธยฐานความพึงพอใจในหัวข้อ ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน ของก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

H_1: มัธยฐานความพึงพอใจในหัวข้อ ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

สมมติฐานที่ 18

H_0: มัธยฐานความพึงพอใจในหัวข้อ รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

H_1: มัธยฐานความพึงพอใจในหัวข้อ รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

สมมติฐานที่ 19

H_0: มัธยฐานความพึงพอใจในหัวข้อ ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน ของก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

H_1: มัธยฐานความพึงพอใจในหัวข้อ ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

สมมติฐานที่ 20

H_0: มัธยฐานความพึงพอใจในหัวข้อ ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

H_1: มัธยฐานความพึงพอใจในหัวข้อ ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

สมมติฐานที่ 21

H_0: มัธยฐานความพึงพอใจในหัวข้อ ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

H_1: มัธยฐานความพึงพอใจในหัวข้อ ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

สมมติฐานที่ 22

H_0: มัธยฐานความพึงพอใจในหัวข้อ ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์) ได้มากขึ้น ของก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

H_1: มัธยฐานความพึงพอใจในหัวข้อ ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

สมมติฐานที่ 23

H_0: มัธยฐานความพึงพอใจในหัวข้อ ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละคร ในท่าทางการเดิน , การโจมตีในเกมส์มากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

H_1: มัธยฐานความพึงพอใจในหัวข้อ ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละคร ในท่าทางการเดิน , การโจมตีในเกมส์มากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

สมมติฐานที่ 24

H_0: มัธยฐานความพึงพอใจในหัวข้อ ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

H_1: มัธยฐานความพึงพอใจในหัวข้อ ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด ของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

ตารางที่ 4.6 ค่าเฉลี่ย ($\overline{X}$) ส่วนเบี่ยงเบนมาตรฐาน (S.D.) สถิติทดสอบการแจกแจงแบบปกติของประชากร (Shapiro-Wilk) และการทดสอบสมมติฐาน Wilcoxon Signed Ranks Test (z) ต่อระดับความพึงพอใจของการเล่นเกมส์ทั้งก่อนและหลังการปรับปรุง

คำถาม	ก่อนการปรับปรุง $\overline{X}$ (s_d)	หลังการปรับปรุง $\overline{X}$ (s_d)	Shapiro-Wilk ของก่อนและหลังการปรับปรุง		สถิติทดสอบ (Sig.)
			Statistic (Sig.)	Statistic (Sig.)	
1.ตามความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมที่น่าสนใจมากน้อยแค่ไหน	2.95 (0.686)	3.30 (0.571)	0.807 (0.001)	0.739 (<0.001)	z = -2.646 (0.008)
2.รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อย เพียงใด	2.60 (0.754)	4.00 (0.649)	0.858 (0.007)	0.793 (0.001)	z = -3.502 (<0.001)
3.ในความคิดของผู้เล่นคิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน	3.65 (0.745)	3.75 (0.851)	0.855 (0.006)	0.874 (0.014)	z = -0.816 (0.414)
4.ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด	2.70 (0.923)	3.00 (0.649)	0.828 (0.002)	0.793 (0.001)	z = -2.121 (0.034)
5.ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด	2.95 (0.510)	3.65 (0.587)	0.688 (<0.001)	0.744 (<0.001)	z = -2.952 (0.003)
6.ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น	4.10 (0.553)	4.25 (0.716)	0.728 (<0.001)	0.698 (<0.001)	z = -1.342 (0.180)
7.ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตีในเกมมากน้อยเพียงใด	3.25 (0.716)	3.60 (0.821)	0.821 (0.002)	0.863 (0.009)	z = -2.111 (0.035)
8.ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด	2.75 (0.967)	4.30 (0.801)	0.893 (0.031)	0.760 (<0.001)	z = -3.457 (0.001)

หมายเหตุ จากโปรแกรมสำเร็จได้ใช้ตัวสถิติทดสอบ z แทนตัวสถิติทดสอบ W

จากตารางที่ 4.6 เป็นการเปรียบเทียบค่าเฉลี่ย ($\overline{X}$) ส่วนเบี่ยงเบนมาตรฐาน (S.D.) สถิติทดสอบการแจกแจงแบบปกติของประชากร (Shapiro-Wilk) และการทดสอบสมมติฐาน Wilcoxon Signed Ranks Test (z) ต่อระดับความพึงพอใจของการเล่นเกมส์ทั้งก่อนและหลังการปรับปรุง ผลการวิเคราะห์ในแต่ละหัวข้อเป็นดังนี้

ในส่วนความคิดของผู้เล่น คิดว่าเกมนี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน พบว่า ค่า Sig. ใน Shapiro-Wilk ของก่อนการปรับปรุงเท่ากับ 0.001 และหลังการปรับปรุงน้อยกว่า 0.001 ซึ่งทั้งก่อนและหลังมีค่าน้อยกว่า 0.05 แสดงว่าข้อมูลทั้งก่อนและหลังมีการแจกแจงไม่ปกติ จึงใช้ Wilcoxon Signed Ranks Test ในการทดสอบผลต่างของค่ามัธยฐานก่อนและหลังการปรับปรุง ผลการทดสอบพบว่า ค่า Sig.(2-tailed)/2 = 0.008 < 0.05 ดังนั้นจึงปฏิเสธสมมติฐานหลัก และสรุปได้ว่า ในหัวข้อนี้มีค่ามัธยฐานของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

ส่วนรูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด พบว่า ค่า Sig. ใน Shapiro-Wilk ของก่อนการปรับปรุงเท่ากับ 0.007 และหลังการปรับปรุงเท่ากับ 0.001 ซึ่งทั้งก่อนและหลังมีค่าน้อยกว่า 0.05 แสดงว่าข้อมูลทั้งก่อนและหลังมีการแจกแจงไม่ปกติ จึงใช้ Wilcoxon Signed Ranks Test ในการทดสอบผลต่างของค่ามัธยฐานก่อนและหลังการปรับปรุง ผลการทดสอบพบว่า ค่า Sig.(2-tailed)/2 < 0.001 < 0.05 ดังนั้นจึงปฏิเสธสมมติฐานหลัก และสรุปได้ว่า ในหัวข้อนี้มีค่ามัธยฐานของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

ส่วนในแง่ความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน พบว่า ค่า Sig. ใน Shapiro-Wilk ของก่อนการปรับปรุงเท่ากับ 0.006 และหลังการปรับปรุงเท่ากับ 0.014 ซึ่งทั้งก่อนและหลังมีค่าน้อยกว่า 0.05 แสดงว่าข้อมูลทั้งก่อนและหลังมีการแจกแจงไม่ปกติ จึงใช้ Wilcoxon Signed Ranks Test ในการทดสอบผลต่างของค่ามัธยฐานก่อนและหลังการปรับปรุง ผลการทดสอบพบว่าค่า Sig.(2-tailed)/2 = 0.414 > 0.05 ดังนั้นจึงยอมรับสมมติฐานหลัก และสรุปได้ว่า ในหัวข้อนี้มีค่ามัธยฐานของก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

นอกจากนี้ในความคิดของผู้เล่น ต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด พบว่า ค่า Sig. ใน Shapiro-Wilk ของก่อนการปรับปรุงเท่ากับ 0.002 และหลังการปรับปรุงเท่ากับ 0.001 ซึ่งทั้งก่อนและหลังมีค่าน้อยกว่า 0.05 แสดงว่าข้อมูลทั้งก่อนและหลังมีการแจกแจงไม่ปกติ จึงใช้ Wilcoxon Signed Ranks Test ในการทดสอบผลต่างของค่ามัธยฐานก่อนและหลังการปรับปรุง ผลการทดสอบพบว่าค่า Sig.(2-tailed)/2 = 0.034 < 0.05 ดังนั้นจึงปฏิเสธสมมติฐานหลัก และสรุปได้ว่า ในหัวข้อนี้มีค่ามัธยฐานของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

ส่วนความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด พบว่า ค่า Sig. ใน Shapiro-Wilk ของทั้งก่อนการปรับปรุงและหลังการปรับปรุงน้อยกว่า 0.001 ซึ่งทั้งก่อน

และหลังมีค่าน้อยกว่า 0.05 แสดงว่าข้อมูลทั้งก่อนและหลังมีการแจกแจงไม่ปกติ จึงใช้ Wilcoxon Signed Ranks Test ในการทดสอบผลต่างของค่ามัธยฐานก่อนและหลังการปรับปรุง ผลการทดสอบพบว่าค่า Sig.(2-tailed)/2 = 0.003 < 0.05 ดังนั้นจึงปฏิเสธสมมติฐานหลัก และสรุปได้ว่า ในหัวข้อนี้มีค่ามัธยฐานของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

ในแง่ความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้ (เกี่ยวกับโรคเอดส์) ได้มากขึ้น พบว่า ค่า Sig. ใน Shapiro-Wilk ของทั้งก่อนการปรับปรุงน้อยกว่าและหลังการปรับปรุงน้อยกว่า 0.001 ซึ่งทั้งก่อนและหลังมีค่าน้อยกว่า 0.05 แสดงว่าข้อมูลทั้งก่อนและหลังมีการแจกแจงไม่ปกติ จึงใช้ Wilcoxon Signed Ranks Test ในการทดสอบผลต่างของค่ามัธยฐานก่อนและหลังการปรับปรุง ผลการทดสอบพบว่าค่า Sig.(2-tailed)/2 = 0.180 > 0.05 ดังนั้นจึงยอมรับสมมติฐานหลัก และสรุปได้ว่า ในหัวข้อนี้มีค่ามัธยฐานของก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

นอกจากนี้ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตี ในเกมส์มากน้อยเพียงใด พบว่า ค่า Sig. ใน Shapiro-Wilk ของก่อนการปรับปรุงเท่ากับ 0.002 และหลังการปรับปรุงเท่ากับ 0.009 ซึ่งทั้งก่อนและหลังมีค่าน้อยกว่า 0.05 แสดงว่าข้อมูลทั้งก่อนและหลังมีการแจกแจงไม่ปกติ จึงใช้ Wilcoxon Signed Ranks Test ในการทดสอบผลต่างของค่ามัธยฐานก่อนและหลังการปรับปรุง ผลการทดสอบพบว่า ค่า Sig.(2-tailed)/2 = 0.035 < 0.05 ดังนั้นจึงปฏิเสธสมมติฐานหลัก และสรุปได้ว่า ในหัวข้อนี้มีค่ามัธยฐานของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

และสุดท้ายผู้เล่นคิดว่าขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด พบว่า ค่า Sig. ใน Shapiro-Wilk ของก่อนการปรับปรุงเท่ากับ 0.031 และหลังการปรับปรุงน้อยกว่า 0.001 ซึ่งทั้งก่อนและหลังมีค่าน้อยกว่า 0.05 แสดงว่าข้อมูลทั้งก่อนและหลังมีการแจกแจงไม่ปกติ จึงใช้ Wilcoxon Signed Ranks Test ในการทดสอบผลต่างของค่ามัธยฐานก่อนและหลังการปรับปรุง ผลการทดสอบพบว่าค่า Sig.(2-tailed)/2 = 0.001 < 0.05 ดังนั้นจึงปฏิเสธสมมติฐานหลัก และสรุปได้ว่า ในหัวข้อนี้มีค่ามัธยฐานของก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

สมมติฐานที่ 25

H_0: มัธยฐานของระดับการศึกษา 4 กลุ่มทั้งก่อนและหลังการปรับปรุงเกมส์ไม่แตกต่างกัน

H_1: มัธยฐานของระดับการศึกษา 4 กลุ่มทั้งก่อนและหลังการปรับปรุงเกมส์แตกต่างกัน

ตารางที่ 4.7 การทดสอบของครัสคาลและวอลลิสของผู้ตอบแบบสอบถามที่มีต่อเกมส์ของก่อนและหลังการปรับปรุง จำแนกตามระดับการศึกษา

ระดับการศึกษา	**จำนวน**	**ผลรวมของลำดับที่**
ประถมศึกษา	5	9.6
มัธยมตอนต้น	5	9.4
มัธยมตอนปลาย	5	13
อุดมศึกษา	5	10
รวม	**20**	

ตารางที่ 4.8 สถิติที่ใช้ทดสอบ

	ค่าแตกต่างระหว่างคะแนนก่อน - คะแนนหลังการปรับปรุงเกมส์
สถิติที่ทดสอบ (χ^2)	1.23
องศาอิสระ	3
Asymp. Sig.	0.746

หมายเหตุ จากโปรแกรมสำเร็จได้ใช้ตัวสถิติทดสอบ χ^2 แทนตัวสถิติทดสอบ H

ผลการทดสอบของครัสคาลและวอลลิสของผู้ตอบแบบสอบถามที่มีต่อเกมส์ของก่อนและหลังการปรับปรุงจำแนกตามระดับการศึกษาได้ค่า χ^2 = 1.23 ที่องศาอิสระ = 3 ค่า P-value = 0.388 จะเห็นได้ว่าค่า P-value > 0.05 ดังนั้นจึงยอมรับสมมติฐานหลัก นั่นคือ ค่ามัธยฐานของแต่ละระดับการศึกษาไม่แตกต่างกันที่ระดับนัยสำคัญ $\alpha = 0.05$

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

เมื่อพูดถึงเกมส์คอมพิวเตอร์ในปัจจุบัน จะพบว่ากลุ่มเยาวชนไทยนิยมเล่นกันอย่างแพร่หลาย โดยพบได้จากร้านเกมส์ต่าง ๆ ที่เพิ่มขึ้นมาอย่างรวดเร็ว โดยเกมส์คอมพิวเตอร์ที่ได้รับความนิยมส่วนใหญ่จะเป็นเกมส์ที่ไม่ใช่เชิงวิชาการ เพราะเกมส์คอมพิวเตอร์เหล่านี้จะมีระบบการเล่นที่ดึงดูดใจ หรือมีเนื้อเรื่องที่น่าสนใจ น่าติดตาม ซึ่งเกมส์เชิงวิชาการนั้นไม่มี จึงเป็นที่น่าเสียดายที่เกมส์เชิงวิชาการนั้นไม่ค่อยได้รับความนิยม เพราะฉะนั้นทางผู้วิจัยจึงเล็งเห็นความสำคัญที่จะพัฒนาเกมส์เชิงวิชาการให้มีระบบการเล่นที่ดึงดูดใจมากขึ้น เพราะนอกจากจะให้สาระความรู้แล้วยังได้สร้างความเพลิดเพลินให้แก่ผู้เล่นภายในตัว ทั้งนี้ทางคณะผู้จัดทำจึงนำเกมส์ Captain Mandela มาเป็นกรณีศึกษา โดยเนื้อหาภายในเกมส์นอกจากจะมีกราฟฟิคที่น่าสนใจ เพื่อความเพลินเพลินแล้วยังสอดแทรกเนื้อหาเกี่ยวกับ โรคเอดส์ เข้าไปด้วย เพราะเป็นที่ทราบกันดีว่า โรคเอดส์นั้นกำลังสร้างปัญหาให้กับโลกของเราเป็นอย่างมาก เกมส์นี้จะทำให้ผู้เล่นทราบถึงพิษภัยของโรคเอดส์ได้มากยิ่งขึ้น อาทิเช่น เอดส์คืออะไร เอดส์สามารถติดต่อกันได้อย่างไร เอดส์มีอาการอย่างไร เอดส์รักษาได้หรือไม่ ฯลฯ โดยเนื้อหาความรู้เหล่านี้จะสอดแทรกเข้าไปในตัวเกมส์

อย่างไรก็ตามในเกมส์จะพบปัญหาการเดินของตัวละคร ซึ่งโดยธรรมชาติแล้วการเดินของตัวละครในเกมส์นั้น จะเดินเป็นเส้นตรงทำให้ตัวละครในเกมส์นั้นจะเดินไปชนและติดที่วัตถุต่างๆ ที่ขวางเส้นทางการเดินของตัวละคร จากปัญหาข้างต้นนั้นสามารถที่จะแก้ไขได้โดยการประยุกต์การหาเส้นทางสั้นที่สุดโดยใช้ A-Star search algorithm เข้าไปพัฒนา ในส่วนของ Game Engine ซึ่งจะใช้ภาษา C++ ในการพัฒนา โดยจุดหลักที่มุ่งเน้นคือการ ประยุกต์ใช้ Dynamic Programming และ branch and bound ของการวิจัยการดำเนินงาน เพื่อทำการพัฒนา Artificial Intelligence(AI) ของ Game Engine โดยทำให้วัตถุสามารถเคลื่อนที่ในเกมส์ได้นั้นเคลื่อนที่ด้วย Artificial Intelligence ที่จะสามารถหลบหลีกสิ่งกีดขวางต่างๆได้ง่ายขึ้น

เมื่อทำการพัฒนาในเกมส์เรียบร้อยแล้วพบว่าการเดินของตัวละครดีขึ้นคือ สามารถเดินหลบสิ่งกีดขวางต่าง ๆ ได้อย่างรวดเร็วโดยเพียงแค่คลิก 1 ครั้งที่จุดเป้าหมาย จากนั้นทำการทดสอบจากหน่วยตัวอย่าง 20 คน ที่เป็นกลุ่มเยาวชนโดยแบ่งตามระดับการศึกษา ๆ ละ 5 คน คือ ระดับประถม ระดับมัธยมศึกษาตอนต้น ระดับมัธยมศึกษาตอนปลายและระดับอุดมศึกษา โดยผู้วิจัยให้หน่วยตัวอย่างทำแบบสอบถามก่อนและหลังของการปรับปรุงเกมส์ ซึ่งเป็นแบบทดสอบ

แบบปรนัยจำนวนทั้งหมด 16 ข้อ จากนั้นนำคะแนนที่ได้จากการทดสอบมาวิเคราะห์ เพื่อทดสอบสมมติฐานดังนี้

1. ตามความคิดของผู้เล่น คิดว่าเกมนี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน โดยใช้สถิติ Wilcoxon Signed Ranks Test สำหรับการทดสอบความแตกต่างของค่ามัธยฐานก่อนและหลังการปรับปรุงเกมส์ พบว่า ผู้เล่นมีความสนใจเกมส์มากขึ้น ภายหลังจากได้เล่นเกมส์หลังการปรับปรุงอย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

2. รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด โดยใช้สถิติ Wilcoxon Signed Ranks Test สำหรับการทดสอบความแตกต่างของค่ามัธยฐานก่อนและหลังการปรับปรุงเกมส์ พบว่า ผู้เล่นมีความพึงพอใจรูปแบบการเดินของตัวละครในเกมส์มากขึ้น ภายหลังจากได้เล่นเกมส์หลังการปรับปรุงอย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

3. ในความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน โดยใช้สถิติ Wilcoxon Signed Ranks Test สำหรับการทดสอบความแตกต่างของค่ามัธยฐานก่อนและหลังการปรับปรุงเกมส์ พบว่า ผู้เล่นมีความพึงพอใจเท่าเดิมเกี่ยวกับเกมส์ Captain Mandela ว่าเป็นเกมส์ที่สร้างสรรค์เหมาะสมกับเยาวชน ทั้งก่อนและหลังการปรับปรุงเกมส์อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

4. ในความคิดของผู้เล่นต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด โดยใช้สถิติ Wilcoxon Signed Ranks Test สำหรับการทดสอบความแตกต่างของค่ามัธยฐานก่อนและหลังการปรับปรุงเกมส์ พบว่า ผู้เล่นมีความต้องการให้ผลิตเกมส์ Captain Mandela ออกจำหน่ายในตลาดมากขึ้น ภายหลังจากได้เล่นเกมส์หลังการปรับปรุงอย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

5. ตามความคิดของผู้เล่นคิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด โดยใช้สถิติ Wilcoxon Signed Ranks Test สำหรับการทดสอบความแตกต่างของค่ามัธยฐานก่อนและหลังการปรับปรุงเกมส์ พบว่า ผู้เล่นมีความพึงพอใจเกี่ยวกับการดำเนินเรื่องของการเล่นเกมส์มากขึ้น ภายหลังจากได้เล่นเกมส์หลังการปรับปรุงอย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

6. ตามความคิดของผู้เล่นคิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้ (เกี่ยวกับโรคเอดส์)ได้มากขึ้น โดยใช้สถิติ Wilcoxon Signed Ranks Test สำหรับการทดสอบความแตกต่างของค่ามัธยฐานก่อนและหลังการปรับปรุงเกมส์ พบว่า ผู้เล่นได้เนื้อหาความรู้ (เกี่ยวกับโรคเอดส์) เท่าเดิม ทั้งก่อนและหลังการปรับปรุงเกมส์อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

7. ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน, การโจมตี ในเกมส์มากน้อยเพียงใด โดยใช้สถิติ Wilcoxon Signed Ranks Test สำหรับการทดสอบความแตกต่างของค่ามัธยฐานก่อนและหลังการปรับปรุงเกมส์ พบว่า ผู้เล่นมีความพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน , การโจมตี ในเกมส์มากขึ้น ภายหลังจากได้เล่นเกมส์หลังการปรับปรุงอย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

8. ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด โดยใช้สถิติ Wilcoxon Signed Ranks Test สำหรับการทดสอบความแตกต่างของค่ามัธยฐานก่อนและหลังการปรับปรุงเกมส์ พบว่า ผู้เล่นเห็นขนาดของตัวอักษรในเกมส์มีความชัดเจนมากขึ้น ภายหลังจากได้เล่นเกมส์หลังการปรับปรุงอย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

นอกจากนี้ยังมีการทดสอบความแตกต่างของแต่ละระดับการศึกษาของก่อนและหลังการปรับปรุงเกมส์ โดยใช้สถิติ The Kruskal – Wallis Oneway ANOVA by Rank Test พบว่า กลุ่มตัวอย่างของแต่ละระดับการศึกษาเกิดการเรียนรู้เกี่ยวกับการเล่นเกมส์ได้ดีเท่า ๆ กันของทุกระดับการศึกษา ภายหลังจากได้เล่นเกมส์หลังการปรับปรุงอย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

5.2 ข้อเสนอแนะ

การพัฒนาเกมส์ Captain Mandela ในครั้งนี้ ทางคณะผู้จัดทำได้พัฒนาเฉพาะการเดินของตัวละคร และทำให้การเล่นเกมส์ Captain Mandela มีความสนุกมากขึ้นในระดับหนึ่ง แต่อย่างไรก็ตามการพัฒนาโปรแกรมนี้ยังไม่ได้พัฒนาส่วนโปรแกรมอื่น ๆ อีก เนื่องจากเวลาที่ใช้ในการศึกษาปัญหาพิเศษนี้มีจำกัด คณะผู้จัดทำจึงขอเสนอแนะไว้ในกรณีที่มีผู้สนใจต้องการพัฒนาปรับปรุงเกมส์ ควรพัฒนาเพิ่มดังต่อไปนี้ เสียงพูดในตัวละคร เพราะจะทำให้ผู้ที่เล่นเกมส์ไม่ต้องใช้เวลาอ่านหรือจำกัดผู้เล่นได้เฉพาะคนที่อ่านหนังสือออกเท่านั้น , สีสันของตัวละคร ควรออกแบบให้มีหลายชุด , สีสันของฉากในเกมส์ ควรออกแบบให้ดูน่าติดตามมากกว่านี้ เช่น ฉากของกำแพงเลือดเป็นสีที่ดูมืด ควรปรับแต่งแสงสีให้ดูชัดเจนมากกว่าเดิม , การดำเนินเรื่องในเกมส์น่าจะเพิ่มรูปแบบการเล่นมากขึ้น เช่น ถ้า Captain Mandela ตอบปัญหาถูกก็จะได้อาวุธหรือเงินรางวัลเป็นค่าตอบแทน และสุดท้ายด้านเนื้อหาที่สอดแทรกในเกมส์นั้น ถือว่าเป็นสิ่งสำคัญควรปรับปรุงให้ตรงกับระดับความรู้ตามระดับการศึกษาต่าง ๆ ของผู้เล่น และในด้านของผู้ที่สนใจต้องการพัฒนาปรับปรุงเกมส์ ควรจะต้องมีเครื่องคอมพิวเตอร์เป็นของตัวเองและต้องเป็นผู้ที่มีความรู้ทางด้านคอมพิวเตอร์หลายด้าน เช่น การประยุกต์การหาเส้นทางสั้นที่สุดโดยใช้ A-Star Search Algorithm, ภาษา C++ , การ ประยุกต์ใช้ Dynamic Programming และ Branch and Bound ของการวิจัยการดำเนินงาน , Graphic Design และ Sound Effect อย่างเชี่ยวชาญ

นอกจากนี้ การที่เก็บกลุ่มตัวอย่างได้น้อยนั้น เนื่องจากทางคณะผู้จัดทำมีข้อจำกัดในเรื่องของระยะเวลา และยังมีเครื่องคอมพิวเตอร์ไม่เพียงพอกับการรองรับการทดลองของการเล่นเกมส์ทั้งก่อนและหลังการปรับปรุงอีกด้วย เพราะฉะนั้นในการศึกษาครั้งต่อไปควรเพิ่มระยะเวลาในการเก็บตัวอย่างให้นานขึ้น เพื่อจะได้กลุ่มตัวอย่างที่มีคุณภาพมากยิ่งขึ้น

บรรณานุกรม

กรกช นฤปิยะกุลและคณะ. **เกม 3 มิติ แบบออนไลน์ : นักสู้แห่งฝัน (3D Game Online : Dream)**. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ. 2545.

ชนินทร์ ชูประเสริฐและคณะ. **การหาเส้นทางเดินรถที่เหมาะสมที่สุดในกรุงเทพมหานคร**. กรุงเทพฯ : มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี. 2543.

ผศ. อุมาพร จันทศร. สถิติที่ไม่ใช้พารามิเตอร์. กรุงเทพฯ : ฟิสิกส์เซ็นเตอร์, 2542.

สุภาภรณ์ เกียรติสิน. **การหาเส้นทางที่ค่าใช้จ่ายน้อยที่สุดสำหรับระบบเครือข่ายโดยใช้อัลกอริทึม A* แบบประสิทธิผลสูง.** กรุงเทพฯ : มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี. 2542.

สัญญา ควรคิด. **การเลือกเส้นทางบนทางด่วนของประเทศไทยแบบอัตโนมัติโดยใช้ป้ายแนะนำจราจร**. กรุงเทพฯ : มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี. 2546.

Amit Patel , **A-Star Algorithm** , 2007.

สืบค้นจาก : http://www-cs-students.stanford.edu/~amitp/gameprog.html

Andrew Moore and Chris Atkeson. **The Parti-game Algorithm for Variable Resolution Reinforcement Learning in Multidimensional State-spaces** . 1994.
สืบค้นจาก : http://www.autonlab.org/tutorials/astar.html

Heyes-Jones. **A*Algorithm Tutorial**. 2006.

สืบค้นจาก : http://www.geocities.com/jheyesjones/astar.html

John Paul Mueller. **Visual C++ .NET Developer's Guide**. McGraw-Hill, 2002.

Ray Lischner. **C++ in a Nutshell**. O'Reilly, 2003.

Steve Rabin. **AI Game Programming Wisdom 3**. Charles River Media Press, 2006.

Stuart J. Russell and Peter Norvig. **Artificial Intelligence : A Modern Approach,** Prentice Hall Press, 1995.

Stuart J. Russell and Peter Norvig. **Artificial Intelligence : A Modern Approach.** Prentice Hall Press, 2002.

Wayne L. Winston. **Operations Research : Applications and Algorithms.** Duxbury Press, 2003.

ภาคผนวก ก

รหัสต้นฉบับ(Source Code)ที่ใช้พัฒนาเกมส์ Game Engine

Server/gameobject.cpp

```
#define ASTAR

void GameObjectFightable::DoMove(int range2)
{
  DGE_TEST(_TargetX != -1 && _TargetY != -1);

#ifdef ASTAR

  if (_TargetX != _SaveTargetX || _TargetY != _SaveTargetY)
  { // reset path
    // TODO: remove _SaveSourceX, _SaveSourceY
    _SaveSourceX = _X;
    _SaveSourceY = _Y;
    _SaveTargetX = _TargetX;
    _SaveTargetY = _TargetY;

      int width  = _Map->GetWidth();
    int height = _Map->GetHeight();

      _PathCount = 0;
    PathFinder *pf = Global_ShambhalaServer->GetPathFinder();
    if (pf->FindPathEx(_Map->GetCollisionPtr(), 512, width, height,
_X, _Y, _TargetX, _TargetY))
    {
      //pf->PrepareResultEx();
      int len = pf->GetResultCount();

      // _MOVSpeed = number of cells that object can walk in 1000 ms
      if (len > _MOVSpeed * PATH_MAX - 1) len = _MOVSpeed * PATH_MAX
- 1;

      int pc = 0;
      int next_len = _MOVSpeed - 2;
      for (int i = 0; i < len; ++i)
      {
        int x  = pf->GetResultX(i);
        int y  = pf->GetResultY(i);
        int dx = x - _TargetX;
        int dy = y - _TargetY;
        if (dx*dx + dy*dy <= range2 || i == len - 1)
        {
          _Path[pc][0] = x;
          _Path[pc][1] = y;
          _Path[pc][2] = (((_MOVSpeed - next_len + i) * 1000 /
_MOVSpeed) + 99) / 100;
          ++pc; break;
        }
        if (i == next_len)
        {
          _Path[pc][0] = x;
          _Path[pc][1] = y;
          _Path[pc][2] = 10; // 1 sec.
          if (++pc == PATH_MAX) break;
          next_len += _MOVSpeed;
        }
      }
      _PathCount = pc;
    }
  }
```

```
    if (_PathCount <= 0)
  { // can't move
    if (GetObjectType() != TYPE_MONSTER || _State == STATE_NORMAL)
    { // pc or monster that not attack
      DoStop();
    }
    return;
  }

  // reset timer
  _DelayTimer = _Path[0][2];
  // move
  _X = _Path[0][0];
  _Y = _Path[0][1];
  --_PathCount;
  for (int i = 0; i < _PathCount; ++i)
  {
    _Path[i][0] = _Path[i + 1][0];
    _Path[i][1] = _Path[i + 1][1];
    _Path[i][2] = _Path[i + 1][2];
  }

  // save state change
  if (_PathCount == 0)
    SaveMoveObject(_X, _Y, _MOVSpeed);
  else
    SaveMoveObject(_Path[_PathCount - 1][0], _Path[_PathCount -
1][1], _MOVSpeed);

  // stop if reach target
  if (_X == _TargetX && _Y == _TargetY && _State == STATE_NORMAL)
  {
    SaveMoveObject(_X, _Y, 0);
    DoStop();
  }
#else
  if (_TargetX != _SaveTargetX || _TargetY != _SaveTargetY)
  { // reset path
    _SaveSourceX = _X;
    _SaveSourceY = _Y;
    _SaveTargetX = _TargetX;
    _SaveTargetY = _TargetY;
  }

  int block = 0;
  int width = _Map->GetWidth();
  int height = _Map->GetHeight();
  int dx = _TargetX - _SaveSourceX;
  int dy = _TargetY - _SaveSourceY;
  int ax = dx >= 0 ? dx : -dx;
  int ay = dy >= 0 ? dy : -dy;
  int m8 = _MOVSpeed * 8;
  int mx = _X;
  int my = _Y;
  int mblock = 0;

  // find the next block the object would be
  if (ax >= ay)
  { // x-axis is major axis
    int cx = dx >= 0 ? +1 : -1;
```

```
    int x  = _X;

    for (int i = 0; i < m8; ++i)
    {
      if (x == _TargetX) break;
      x += cx;
      int y = _SaveSourceY + (x - _SaveSourceX) * dy / dx;
      if (x < 0 || x >= width || y < 0 || y >= height)
        break;
      if (_Map->GetCollision(x, y) != 0)
      {
        if (GetObjectType() != TYPE_MONSTER || _State !=
STATE_ATTACK_1)
          break;
        x = mx; y = my;
        if (!find_path(_Map, x, y, _TargetX, _TargetY))
          break;
        m8 = 0;
        _SaveTargetX = -1;
        //_SaveTargetY = -1;
      }
      // update current position
      if (i < _MOVSpeed)
      {
        _X = x;
        _Y = y;
        ++block;
      }
      mx = x;
      my = y;
      ++mblock;
      // reach target (for attacking)
      if (range2 > 0)
      { // re-use ax, ay
        ax = _TargetX - x;
        ay = _TargetY - y;
        if (ax * ax + ay * ay <= range2)
          break;
      }
    }
  }
  else
  { // y-axis is major axis
    int cy = dy >= 0 ? +1 : -1;
    int y  = _Y;

    for (int i = 0; i < m8; ++i)
    {
      if (y == _TargetY) break;
      y += cy;
      int x = _SaveSourceX + (y - _SaveSourceY) * dx / dy;
      if (x < 0 || x >= width || y < 0 || y >= height)
        break;
      if (_Map->GetCollision(x, y) != 0)
      {
        if (GetObjectType() != TYPE_MONSTER || _State !=
STATE_ATTACK_1)
          break;
        x = mx; y = my;
        if (!find_path(_Map, x, y, _TargetX, _TargetY))
          break;
```

```
        m8 = 0;
        _SaveTargetX = -1;
        //_SaveTargetY = -1;
      }
      // update current position
      if (i < _MOVSpeed)
      {
        _X = x;
        _Y = y;
        ++block;
      }
      mx = x;
      my = y;
      ++mblock;
      // reach target (for attacking)
      if (range2 > 0)
      { // re-use ax, ay
        ax = _TargetX - x;
        ay = _TargetY - y;
        if (ax * ax + ay * ay <= range2)
          break;
      }
    }
  }

  if (block <= 0)
  { // can't move
    if (GetObjectType() != TYPE_MONSTER || _State == STATE_NORMAL)
    { // pc or monster that not attack
      DoStop();
      return;
    }
    block  = _MOVSpeed;
    mblock = _MOVSpeed;
  }

  // _MOVSpeed = number of cells that object can walk in 1000 ms
  // calculate delay
  int delay = block * 1000 / _MOVSpeed;

  // reset timer
  _DelayTimer = (delay + 99) / 100; // divide delay by 100 and round
up

  // save state change
  SaveMoveObject(mx, my, _MOVSpeed);

  // stop if reach target
  if (_X == _TargetX && _Y == _TargetY && _State == STATE_NORMAL)
  {
    DoStop();
  }
#endif
}
```

Server/ pathfinder.cpp

```
int PathFinder::FindPathEx(unsigned char *map, int map_pitch, int
map_width, int map_height, int x0, int y0, int x1, int y1, int fast)
{
  { // linear path search
```

```
int dx = x1 - x0;
int dy = y1 - y0;
if (dx == 0 && dy == 0) return 0;

int step = 0;
// find the next block the object would be
if ((dx >= 0 ? dx : -dx) >= (dy >= 0 ? dy : -dy))
{ // x-axis is major axis
  _Result.SetLength((dx >= 0 ? dx : -dx));
  int cx = dx >= 0 ? +1 : -1;
  int x = x0;
  while (x != x1)
  {
    x += cx;
    int y = y0 + (x - x0) * dy / dx;
    if (x < 0 || x >= map_width || y < 0 || y >= map_height ||
        *(map + y * map_pitch + x) != 0)
    { // path is blocked
      _Result.SetLength(step);
      step = -1;
      break;
    }
    // update current position
    ResultInfo &ri = _Result.Get(step++);
    ri.x = x; ri.y = y;
  }
}
else
{ // y-axis is major axis
  _Result.SetLength((dy >= 0 ? dy : -dy));
  int cy = dy >= 0 ? +1 : -1;
  int y = y0;
  while (y != y1)
  {
    y += cy;
    int x = x0 + (y - y0) * dx / dy;
    if (x < 0 || x >= map_width || y < 0 || y >= map_height ||
        *(map + y * map_pitch + x) != 0)
    { // path is blocked
      _Result.SetLength(step);
      step = -1;
      break;
    }
    // update current position
    ResultInfo &ri = _Result.Get(step++);
    ri.x = x; ri.y = y;
  }
}
if (step > 0) return 1; // path found
}

// none a* version
if (fast) return _Result.GetLength() > 0;

{ // a* path search
int x2 = x0 - MAP_WIDTH  / 2;
int y2 = y0 - MAP_HEIGHT / 2;

if (x2 < 0) x2 = 0; else if (x2 > map_width  - MAP_WIDTH)  x2 =
map_width  - MAP_WIDTH;
```

```
  if (y2 < 0) y2 = 0; else if (y2 > map_height - MAP_HEIGHT) y2 =
map_height - MAP_HEIGHT;

  _Map      = map + y2 * map_pitch + x2;
  _MapPitch = map_pitch;

  x0 -= x2; y0 -= y2;
  if (x0 < 0 || y0 < 0 || x0 >= MAP_WIDTH || y0 >= MAP_HEIGHT) return
0;
  x1 -= x2; y1 -= y2;
  if (x1 < 0 || y1 < 0 || x1 >= MAP_WIDTH || y1 >= MAP_HEIGHT) return
0;

  _RefX = x2; _RefY = y2;

  if (!FindPath(x0, y0, x1, y1)) return _Result.GetLength() > 0;
  PrepareResultEx();
  return 1;
  }
}

void PathFinder::PrepareResultEx()
{
  if (_State[_TargetY][_TargetX][0] != _CurrentState)
  { // not path found
    _Result.SetLength(0);
    return;
  }

  int rx = _RefX;
  int ry = _RefY;
  int x  = _TargetX;
  int y  = _TargetY;
  int step = _State[y][x][1] & 0xFFFF;

  _Result.SetLength(step);



  while (step > 0)
  {
    --step;
    ResultInfo &ri = _Result.Get(step);
    ri.x = x + rx;
    ri.y = y + ry;

    // eliminate some ugly paths by follow the same direction if
available (only horizontal and vertical path)
    if      (y < MAP_HEIGHT - 1 && (_State[y + 1][x][1] & 0xFFFF) ==
step && _State[y + 1][x][0] == _CurrentState) { ++y; continue; }
    else if (x < MAP_WIDTH  - 1 && (_State[y][x + 1][1] & 0xFFFF) ==
step && _State[y][x + 1][0] == _CurrentState) { ++x; continue; }
    else if (x > 0 && (_State[y][x - 1][1] & 0xFFFF) == step &&
_State[y][x - 1][0] == _CurrentState) { --x; continue; }
    else if (y > 0 && (_State[y - 1][x][1] & 0xFFFF) == step &&
_State[y - 1][x][0] == _CurrentState) { --y; continue; }

    int filter = _State[y][x][1] >> 24;
    if      (filter == 0) { ++x; ++y; }
    //else if (filter == 1) {      ++y; }
```

```
    else if (filter == 2) { --x; ++y; }
    //else if (filter == 3) { ++x;      }
    //else if (filter == 4) { --x;      }
    else if (filter == 5) { ++x; --y; }
    //else if (filter == 6) {      --y; }
    else if (filter == 7) { --x; --y; }
  }
}

int PathFinder::CopyPathEx(int path[][2], int *length)
{
  int r = CopyPath(path, length);

  if (r > 0)
  {
    int rx = _RefX;
    int ry = _RefY;
    for (int i = *length; --i >= 0; )
    {
      path[i][0] += rx;
      path[i][1] += ry;
    }
  }

  return r;
}

void PathFinder::MapToMap(unsigned char *map, int map_pitch)
{
  _Map      = map;
  _MapPitch = map_pitch;
}



int PathFinder::FindPath(int x0, int y0, int x1, int y1)
{
  //if (x0 < 0 || x0 >= MAP_WIDTH)  return 0;
  //if (y0 < 0 || y0 >= MAP_HEIGHT) return 0;
  //if (x1 < 0 || x1 >= MAP_WIDTH)  return 0;
  //if (y1 < 0 || y1 >= MAP_HEIGHT) return 0;

  // reset state
  if (_CurrentState == 0)
  {
    for (int i = MAP_HEIGHT; --i >= 0; )
    {
      for (int j = MAP_WIDTH; --j >= 0; )
      {
        _State[i][j][0] = 0;
        _State[i][j][1] = 0;
      }
    }
  }
  ++_CurrentState;
  _StepQueue.Clear();

  // save target
  _TargetX = x1;
  _TargetY = y1;
```

```
  // set original
  AppendToQueue(x0, y0, 0, 0xFF);

  // find path
  while (!_StepQueue.Empty())
  {
    StepInfo si = _StepQueue.GetHead();
    _StepQueue.Pop();

    if (si.x == x1 && si.y == y1)
    {
      // a path found
      return 1;
    }

    if (si.x != 0 && si.x != MAP_WIDTH - 1 && si.y != 0 && si.y !=
MAP_HEIGHT - 1)
    {
      int rval = (_State[si.y][si.x][1] & 0xFFFF) + 1;
      unsigned char *m = _Map + si.y * _MapPitch + si.x;
      if ((si.filter & 0x01) != 0 && *(m - _MapPitch - 1) == 0)
        AppendToQueue(si.x - 1, si.y - 1, rval, 0x002F);
      if ((si.filter & 0x02) != 0 && *(m - _MapPitch    ) == 0)
        AppendToQueue(si.x    , si.y - 1, rval, 0x0107);
      if ((si.filter & 0x04) != 0 && *(m - _MapPitch + 1) == 0)
        AppendToQueue(si.x + 1, si.y - 1, rval, 0x0297);
      if ((si.filter & 0x08) != 0 && *(m            - 1) == 0)
        AppendToQueue(si.x - 1, si.y    , rval, 0x0329);
      if ((si.filter & 0x10) != 0 && *(m            + 1) == 0)
        AppendToQueue(si.x + 1, si.y    , rval, 0x0494);
      if ((si.filter & 0x20) != 0 && *(m + _MapPitch - 1) == 0)
        AppendToQueue(si.x - 1, si.y + 1, rval, 0x05E9);
      if ((si.filter & 0x40) != 0 && *(m + _MapPitch    ) == 0)
        AppendToQueue(si.x    , si.y + 1, rval, 0x06E0);
      if ((si.filter & 0x80) != 0 && *(m + _MapPitch + 1) == 0)
        AppendToQueue(si.x + 1, si.y + 1, rval, 0x07F4);
    }
  }

  // path not found
  return 0;
}

int PathFinder::CopyPath(int path[][2], int *length)
{
  //if (_TargetX < 0 || _TargetX >= MAP_WIDTH)  return 0;
  //if (_TargetY < 0 || _TargetY >= MAP_HEIGHT) return 0;
  //if (length == NULL) return 0;

  if (_State[_TargetY][_TargetX][0] != _CurrentState)
  { // not path found
    *length = 0;
    return 1;
  }

  if (path == NULL)
  { // query path length
    *length = _State[_TargetY][_TargetX][1] & 0xFFFF;
    return 1;
  }
```

```
  if (*length < (_State[_TargetY][_TargetX][1] & 0xFFFF))
  { // not enough space
    *length = _State[_TargetY][_TargetX][1] & 0xFFFF;
    return 0;
  }

  int x = _TargetX;
  int y = _TargetY;
  int step = _State[y][x][1] & 0xFFFF;
  *length = step;

  while (step > 0)
  {
    --step;
    path[step][0] = x;
    path[step][1] = y;
//*
    // eliminate some ugly paths by follow the same direction if
available (only horizontal and vertical path)
    if      (y < MAP_HEIGHT - 1 && (_State[y + 1][x][1] & 0xFFFF) ==
step && _State[y + 1][x][0] == _CurrentState) { ++y; continue; }
    else if (x < MAP_WIDTH  - 1 && (_State[y][x + 1][1] & 0xFFFF) ==
step && _State[y][x + 1][0] == _CurrentState) { ++x; continue; }
    else if (x > 0 && (_State[y][x - 1][1] & 0xFFFF) == step &&
_State[y][x - 1][0] == _CurrentState) { --x; continue; }
    else if (y > 0 && (_State[y - 1][x][1] & 0xFFFF) == step &&
_State[y - 1][x][0] == _CurrentState) { --y; continue; }
/**/
/*
    switch (_State[y][x][1] >> 24)
    {
      case 0: ++x; ++y; break;
      //case 1:      ++y; break;
      case 2: --x; ++y; break;
      //case 3: ++x;      break;
      //case 4: --x;      break;
      case 5: ++x; --y; break;
      //case 6:      --y; break;
      case 7: --x; --y; break;
    }
/*/
    int filter = _State[y][x][1] >> 24;
    if      (filter == 0) { ++x; ++y; }
    //else if (filter == 1) {      ++y; }
    else if (filter == 2) { --x; ++y; }
    //else if (filter == 3) { ++x;      }
    //else if (filter == 4) { --x;      }
    else if (filter == 5) { ++x; --y; }
    //else if (filter == 6) {      --y; }
    else if (filter == 7) { --x; --y; }
/**/
  }
  return 1;
}

void PathFinder::PrepareResult()
{
  if (_State[_TargetY][_TargetX][0] != _CurrentState)
  { // not path found
    _Result.SetLength(0);
```

```
    return;
  }

  int x = _TargetX;
  int y = _TargetY;
  int step = _State[y][x][1] & 0xFFFF;

  _Result.SetLength(step);

  while (step > 0)
  {
    --step;
    _Result.Get(step).x = x;
    _Result.Get(step).y = y;



    // eliminate some ugly paths by follow the same direction if
available (only horizontal and vertical path)
    if      (y < MAP_HEIGHT - 1 && (_State[y + 1][x][1] & 0xFFFF) ==
step && _State[y + 1][x][0] == _CurrentState) { ++y; continue; }
    else if (x < MAP_WIDTH  - 1 && (_State[y][x + 1][1] & 0xFFFF) ==
step && _State[y][x + 1][0] == _CurrentState) { ++x; continue; }
    else if (x > 0 && (_State[y][x - 1][1] & 0xFFFF) == step &&
_State[y][x - 1][0] == _CurrentState) { --x; continue; }
    else if (y > 0 && (_State[y - 1][x][1] & 0xFFFF) == step &&
_State[y - 1][x][0] == _CurrentState) { --y; continue; }

    int filter = _State[y][x][1] >> 24;
    if      (filter == 0) { ++x; ++y; }
    //else if (filter == 1) {      ++y; }
    else if (filter == 2) { --x; ++y; }
    //else if (filter == 3) { ++x;      }
    //else if (filter == 4) { --x;      }
    else if (filter == 5) { ++x; --y; }
    //else if (filter == 6) {      --y; }
    else if (filter == 7) { --x; --y; }
  }
}

inline static int calc_h_dist(int x, int y)
{
  if (x < 0) x = -x;
  if (y < 0) y = -y;
  return x >= y ? x * 65536 + y : y * 65536 + x;
}

void PathFinder::AppendToQueue(int x, int y, int value, int filter)
{
  if (_State[y][x][0] == _CurrentState) return;

  int v = value * 65536 + calc_h_dist(_TargetX - x, _TargetY - y);

  _State[y][x][0] = _CurrentState;
  _State[y][x][1] = value | (filter << 16);

  _StepQueue.Push(StepInfo(x, y, v, filter));
```

ส่วนต่อไปนี้จะเป็น โค้ดที่ใช้พัฒนาให้ Client แสดงผลการเดินแบบ A* อย่างถูกต้อง

Client/gameobject.cpp

```
#define ASTAR

void Character::MoveTo(Point dst, int speed)
{
  if (_DstPos != dst)
  {
    if (speed > 0)
    { // move to command
      Engine *engine = Global::GetEngine();
      MapManager *mm = engine->GetMapManager();
      unsigned char *map = mm->GetMapCollisionPtr();
      PathFinder *pf = engine->GetPathFinder();
#ifdef ASTAR
      int r = pf->FindPathEx(map, MAX_COLLISION_X, mm->GetMapWidth()
* 16, mm->GetMapHeight() * 16, _Pos.X / 16, _Pos.Y / 16, dst.X / 16,
dst.Y / 16);
#else
      int r = 2;
#endif
      if (r == 1)
      { // non-linear walking
        _LinearMode = 0;
        int len = pf->GetResultCount();
        _MovePos.SetLength(len);
        for (int i = 0; i < len; ++i)
        { // copy by reversed order
          Point &p = _MovePos.Get(len - i - 1);
          p.X = pf->GetResultX(i) * 16;
          p.Y = pf->GetResultY(i) * 16;
        }
        _SrcPos = _Pos;
        _DstPos = _MovePos.Get(len - 1);
        _MoveSpeed = speed;
      }
      else if (r == 2)
      { // linear walking
        _LinearMode = 1;
        _SrcPos = _Pos;
        _DstPos = dst;
      }
      if (r != 0)
      {
        int dx = _Pos.X - _DstPos.X;
        int dy = _Pos.Y - _DstPos.Y;
        if (dx < 0) dx = -dx;
        if (dy < 0) dy = -dy;
        _MoveTime = 0;
        _TotalMoveTime = (dx > dy ? dx : dy) * 1000 / (speed * 16);
      }
    }
    else
    { // set position command
      int dx = _Pos.X - dst.X;
      int dy = _Pos.Y - dst.Y;
      if (dx*dx + dy*dy > 3*3*16*16) _Pos = dst;
      _DstPos = _Pos;
```

```
      _MoveTime = 1;
      _TotalMoveTime = 1;
    }
  }
  if (_Action <= ACTION_CAST && _MoveTime < _TotalMoveTime)
  {
    if (_CastEffect != NULL) SetCasting(false);
    SetAction(ACTION_WALK);
    if (_Pos != _DstPos)
    {
      SetDirection((_ActionData[_Action]->GetDirectionCount() == 4)
?
                   CalcDirection4( _Pos.X, _Pos.Y, _DstPos.X,
_DstPos.Y) :
                   CalcDirection8(_Pos.X, _Pos.Y, _DstPos.X,
_DstPos.Y));
    }
  }
}
```

ภาคผนวก ข

แบบสอบถาม

แบบสอบถามวัดระดับความพึงพอใจของผู้เล่นที่มี ต่อเกมส์ Captain Mandela

(ชุดที่ 1 : การเล่นเกมส์**ก่อน**การปรับปรุง)

ส่วนที่1 ข้อมูลเบื้องต้นผู้ตอบแบบสอบถาม

1.1) เพศ ☐ ชาย ☐ หญิง

1.2) อายุ.................................ปี

1.3) การศึกษา ☐ ประถมศึกษา ☐ มัธยมศึกษา

☐ อาชีวศึกษา ☐ ตั้งแต่ปริญญาขึ้นไป

☐ อื่นๆ(ระบุ) ....

<u>ส่วนที่2</u> ส่วนคำถาม

ระบุระดับคุณภาพความพึงพอใจที่มีต่อเกมส์ตามความรู้สึกของท่านว่ามากน้อยเพียงใด แล้วทำเครื่องหมาย ✓ ในช่องว่างที่เหมาะสมกับความเป็นจริงของตัวท่านมากที่สุดช่องใดช่องหนึ่งใน 5 ช่องที่เหล่านี้

5 = มากที่สุด, 4 = มาก, 3 = ปานกลาง, 2 = น้อย, 1 = น้อยที่สุด

ข้อ	คำถาม	ระดับความพึงพอใจต่อการเล่นเกมส์				
		มากที่สุด	มาก	ปานกลาง	น้อย	น้อยที่สุด
1	ตามความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน					
2	รูปแบบการเดินของตัวละคร ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด					
3	ในความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์ เหมาะสมกับเยาวชน					
4	ในความคิดของผู้เล่น ต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด					
5	ตามความคิดของผู้เล่น คิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด					
6	ตามความคิดของผู้เล่น คิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น					
7	ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน การโจมตีในเกมส์มากน้อยเพียงใด					
8	ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด					

<u>**ส่วนที่ 3**</u> **ข้อเสนอแนะอื่นๆ**

....
....
....
....
....

แบบสอบถามวัดระดับความพึงพอใจของผู้เล่นที่มี ต่อเกมส์ Captain Mandela
(ชุดที่ 2 : การเล่นเกมส์<u>หลัง</u>การปรับปรุง)

<u>ส่วนที่1</u> ข้อมูลเบื้องต้นผู้ตอบแบบสอบถาม

1.1) เพศ ☐ ชาย ☐ หญิง

1.2) อายุ.................................ปี

1.3) การศึกษา ☐ ประถมศึกษา ☐ มัธยมศึกษา
☐ อาชีวศึกษา ☐ ตั้งแต่ปริญญาขึ้นไป
☐ อื่นๆ(ระบุ) ....

ส่วนที่2 ส่วนคำถาม

ระบุระดับคุณภาพความพึงพอใจที่มีต่อเกมส์ตามความรู้สึกของท่านว่ามากน้อยเพียงใด แล้วทำเครื่องหมาย ✓ ในช่องว่างที่เหมาะสมกับความเป็นจริงของตัวท่านมากที่สุดช่องใดช่องหนึ่งใน 5 ช่องที่เหล่านี้

5 = มากที่สุด, 4 = มาก, 3 = ปานกลาง, 2 = น้อย, 1 = น้อยที่สุด

ข้อ	คำถาม	ระดับความพึงพอใจต่อการเล่นเกมส์				
		มากที่สุด	มาก	ปานกลาง	น้อย	น้อยที่สุด
1	ตามความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่น่าสนใจมากน้อยแค่ไหน					
2	รูปแบบการเดินของตัวละคร(หลังการปรับปรุง) ผู้เล่นรู้สึกพึงพอใจมากน้อยเพียงใด					
3	ในความคิดของผู้เล่น คิดว่าเกมส์นี้เป็นเกมส์ที่สร้างสรรค์ เหมาะสมกับเยาวชน					
4	ในความคิดของผู้เล่น ต้องการให้ผลิตเกมส์นี้ออกจำหน่ายในตลาดมากน้อยเพียงใด					
5	ตามความคิดของผู้เล่น คิดว่าการดำเนินเรื่องของการเล่นเป็นไปได้ดีมากน้อยเพียงใด					
6	ตามความคิดของผู้เล่น คิดว่าเป็นเกมส์ที่ให้เนื้อหาความรู้(เกี่ยวกับโรคเอดส์)ได้มากขึ้น					
7	ผู้เล่นรู้สึกพึงพอใจต่อสีสันภาพรวมของตัวละครในท่าทางการเดิน การโจมตีในเกมส์มากน้อยเพียงใด					
8	ขนาดของตัวอักษรในเกมส์มีความชัดเจนมากน้อยเพียงใด(หลังการปรับปรุง)					

<u>ส่วนที่ 3</u> **ข้อเสนอแนะอื่นๆ**

....

....

....

....

....

	IDEN	SEX	AGE	EDU	A1	A2	A3	A4	A5	A6	A7	A8	B1	B2	B3	B4	B5	B6	B7	B8
1	1	1	1	1	3	4	4	2	3	4	2	4	3	4	4	2	3	4	2	4
2	2	1	1	1	3	4	3	3	4	5	3	4	3	4	3	3	3	5	3	4
3	3	1	1	1	2	3	3	1	3	4	3	2	2	4	3	2	3	4	2	5
4	4	2	1	1	4	2	5	2	3	4	3	1	4	5	5	3	4	4	3	5
5	5	2	1	1	2	3	4	3	3	4	4	2	3	4	4	3	4	5	4	4
6	6	1	2	2	2	3	3	1	3	4	3	2	3	3	4	3	3	4	4	4
7	7	1	2	2	3	2	3	3	4	4	3	3	3	4	3	3	4	4	4	5
8	8	1	2	2	3	2	3	1	3	5	3	3	4	4	3	2	3	5	4	4
9	9	1	2	2	2	3	4	3	3	3	3	4	3	4	4	3	4	4	3	5
10	10	2	2	2	4	3	4	3	2	4	4	3	4	3	5	3	4	5	4	5
11	11	2	3	3	3	2	4	4	3	4	4	3	3	4	4	4	4	4	5	4
12	12	2	3	3	4	3	5	3	3	4	3	2	4	5	5	4	4	4	4	5
13	13	2	3	3	3	2	4	3	3	4	3	3	3	4	4	3	4	4	4	4
14	14	1	3	3	3	2	3	3	2	4	3	2	3	4	3	3	4	4	4	5
15	15	1	3	3	3	2	4	4	2	5	3	2	3	5	4	4	3	5	4	4
16	16	1	4	4	3	2	4	3	3	5	4	3	4	4	4	3	4	5	4	5
17	17	1	4	4	4	1	3	2	3	3	2	2	4	3	2	2	3	2	3	2
18	18	1	4	4	3	3	4	4	3	4	5	5	4	5	5	4	5	5	5	5
19	19	2	4	4	3	3	4	3	3	4	4	3	3	4	3	3	3	4	3	3
20	20	2	4	4	2	3	2	3	3	4	3	2	3	3	3	3	4	4	3	4

ประวัติคณะผู้จัดทำ

ชื่อ-นามสกุล	นางสาวลภัส ดิษฐ์ภัทรกุล
วัน เดือน ปีเกิด	31 มกราคม 2526
สถานที่เกิด	ประเทศมาเลเซีย
จบการศึกษามัธยมต้นจาก	โรงเรียนมัธยมวัดดาวคนอง
จบการศึกษามัธยมปลายจาก	โรงเรียนมัธยมวัดดาวคนอง

ชื่อ-นามสกุล	นางสาวหนึ่งฤทัย ครึ้กครื้นจิตร
วัน เดือน ปีเกิด	14 มิถุนายน 2527
สถานที่เกิด	นครราชสีมา
จบการศึกษามัธยมต้นจาก	โรงเรียนอรุณประดิษฐ์
จบการศึกษามัธยมปลายจาก	โรงเรียนพรหมานุสรณ์จังหวัดเพชรบุรี

ชื่อ-นามสกุล	นายสยาม วรรณโกษิตย์
วัน เดือน ปีเกิด	5 กรกฎาคม 2528
สถานที่เกิด	กรุงเทพมหานคร
จบการศึกษามัธยมต้นจาก	โรงเรียนปทุมคงคา
จบการศึกษามัธยมปลายจาก	โรงเรียนปทุมคงคา