

เกมบนโปรแกรมเชิงจินตภาพ และ แนวคิดเชิงคำนวณอย่างเป็นระบบ Visual Programming and Computational Thinking Game

บุญญพนธ์ พูลสวัสดิ์¹, พนมพร ดอกประโคน²
^{1, 2} ภาควิชาการออกแบบเชิงโต้ตอบ และการพัฒนาเกม

วิทยาลัยครีเอทีฟ ดีไซน์ แอนด์ เอ็นเตอร์เทนเมนต์ เทคโนโลยี, มหาวิทยาลัยธุรกิจบัณฑิตย์

ABSTRACT – Computational thinking becomes a fundamental skill for learners in 21st century and Visual programming is the idea of using graphical icons to create and improve programming skill. This paper describes the implementation and evaluation of a visual programming and computational thinking module in game programming project and pre-post questionnaires. The result is to shorten the developing time and improve their programming skills better.

KEY WORDS -- Computational Thinking; Game Programming; Visual programming

บทคัดย่อ -- แนวคิดเชิงคำนวณอย่างเป็นระบบ กลายเป็นความสามารถพื้นฐานสำหรับผู้เรียนในศตวรรษที่ 21 และมีการใช้โปรแกรมเชิงจินตภาพเข้ามาช่วยสร้าง และปรับปรุงทักษะการเขียนโปรแกรม งานวิจัยนี้นำเสนอแนวทางการออกแบบการเรียนรู้เพื่อวัดผลจากการใช้แนวคิดเชิงคำนวณอย่างเป็นระบบและโปรแกรมเชิงจินตภาพ สรุปผลการทดลองผ่านโครงการพัฒนาเกม และแบบทดสอบก่อนหลังการเรียนรู้ ผลลัพธ์ที่ได้คือระยะเวลาในการพัฒนาเกมที่สั้นลง และผู้เรียนมีทักษะที่เพิ่มขึ้นเกี่ยวกับการเขียนโปรแกรม

คำสำคัญ --แนวคิดเชิงคำนวณอย่างเป็นระบบ; การเขียน โปรแกรม; โปรแกรมเชิงจินตภาพ

1. บทนำ

1.1 ความเป็นมา และความสำคัญของปัญหา

ทักษะการเขียนโปรแกรมในปัจจุบันกลายเป็นทักษะสำคัญที่ต้องมีการเพิ่มระดับการเรียนรู้ที่สูงขึ้น เพื่อเป็นการพัฒนาทรัพยากรบุคคลให้สอดคล้องตามนโยบาย ประเทศไทย 4.0 ที่ว่าด้วยการขับเคลื่อนเศรษฐกิจประเทศไทยด้วยนวัตกรรม โดย นวัตกรรมหนึ่งที่ปรากฏอยู่นั้นคือส่วนของ ดิจิทัล คอนเทนต์ หรือสื่อดิจิทัล ที่มีทั้งสื่อการเรียนรู้พื้นฐานอย่างแอปพลิเคชันการเรียนการสอน หนังสืออิเล็กทรอนิกส์ และความบันเทิง เช่น ภาพยนตร์ แอนิเมชัน สารคดี เกม หรือ สื่อผสมผสานทั้งสองอย่างรวมกันคือ เกมเพื่อการเรียนรู้ ในส่วนของเกมสำหรับการเรียนรู้ ได้กลายเป็นรูปแบบการเรียนรู้รูปแบบหนึ่งที่เป็นได้ทั้งเครื่องมือเสริมสร้างความรู้ และเป็นเครื่องมือในการเพิ่มทักษะการแก้ปัญหาด้านตรรกศาสตร์ด้านการออกแบบ และการเขียนโปรแกรม ซึ่งแนวคิดหลักสำคัญสำหรับการพัฒนาโปรแกรมประเภทเกมเพื่อการเรียนรู้ นั้น คือ

แนวคิดเชิงคำนวณอย่างเป็นระบบ Computational Thinking (CT) เป็นทักษะประเภทหนึ่งที่ทำเป็นต่อผู้เรียนในศตวรรษที่ 21 ในเรื่องของการวิเคราะห์ปัญหาหลัก โดยมีการสังเกตรูปแบบของปัญหา และสถานการณ์ เพื่อนำไปออกแบบแนวทางการแก้ปัญหา และใช้การออกแบบดังกล่าวที่วิเคราะห์ออกมาได้ปรับใช้ในลักษณะที่เป็นรูปแบบ (Pattern) อีกทั้งในปัจจุบันเทคโนโลยีสำหรับพัฒนาเกม หรือเกมส์เอ็นจินนั้นก็มีให้เลือกใช้อยู่หลายรูปแบบ และภาษาโปรแกรมสำหรับการพัฒนาเองนั้นก็เป็ภาษาโปรแกรมพื้นฐานที่ผู้เรียนสามารถเรียนรู้ด้วยตนเอง และมีการทำวิจัยเกี่ยวกับการออกแบบเกมส์เอ็นจินให้อยู่ในรูปแบบสื่อการเรียนรู้อัตโนมัติที่สามารถเพิ่มทักษะการเขียนโปรแกรมที่สูงขึ้นโดยไม่อยู่ในรูปแบบการเรียนการสอนแบบเดิมในตำราที่ไม่เห็นผลลัพธ์ทันทีจากการเขียนโปรแกรม[1][2] อีกทั้งมีรูปแบบการเรียนภาษาโปรแกรมด้วยรูปแบบโปรแกรมเชิงจินตภาพ (Visual Programming) ที่สามารถทำให้ผู้เรียนใช้ประกอบการเรียนรู้ผ่านการลากวางวัตถุ

ที่ปรากฏในเครื่องมือตามเงื่อนไข เพื่อแสดงผลลัพธ์ของโปรแกรมได้เทียบเท่าการเขียนโปรแกรม และยังสามารถแปลงรูปแบบเชิงจินตภาพให้ออกมาเป็นโครงสร้างลำดับขั้นตอนของภาษาโปรแกรมได้อีกด้วย

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้มุ่งเป้าไปที่การนำรูปแบบของโปรแกรมเชิงจินตภาพมาใช้งานร่วมกับปัญหาที่กำหนดขึ้นสำหรับทดสอบแนวคิดเชิงคำนวณอย่างเป็นระบบให้เกิดทักษะการเขียนโปรแกรมที่ดีขึ้นในกลุ่มของผู้เรียนเพื่อนำไปประยุกต์ใช้กับโครงงานการพัฒนาเกมในระยะเวลาที่จำกัด 2 สัปดาห์ เพื่อเป็นการศึกษาว่าการผสมผสานรูปแบบการเรียนรู้ผ่านสื่อการเรียนรู้ที่พัฒนาขึ้นสามารถสร้างความสนใจของผู้เรียนให้รู้จักการแก้ปัญหา และคิดรูปแบบการรับมือได้อย่างมีประสิทธิภาพมากขึ้นผ่านผลลัพธ์ของโครงงานการพัฒนาเกม

1.3 สมมติฐานของการวิจัย

การพัฒนาสื่อการเรียนรู้ออนไลน์ในรูปแบบโปรแกรมเชิงจินตภาพร่วมกับการกำหนดปัญหาเชิงวิเคราะห์สำหรับแนวคิดเชิงคำนวณอย่างเป็นระบบ เป็นกิจกรรมย่อยในแต่ละครั้งที่มีการสอนจะสามารถสร้างความเข้าใจการเขียนโปรแกรมและปัญหาที่เกิดขึ้น แนวคิดและทักษะที่ได้รับการฝึกฝนประจำจะมีผลต่อการพัฒนาโครงงานเกมในระยะเวลาที่กำหนดได้

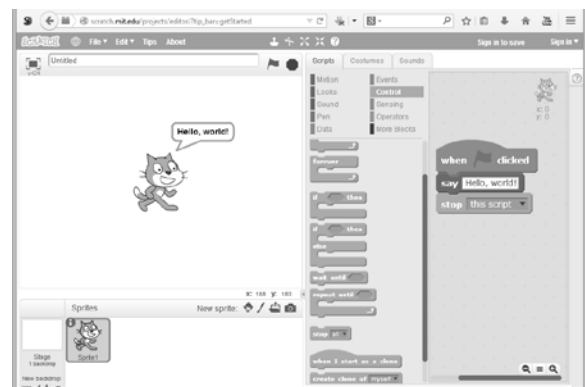
2. ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 การโปรแกรมเชิงจินตภาพ (Visual Programming Language, VPL)

การเขียนโปรแกรมเชิงจินตภาพ คือ รูปแบบการเขียนโปรแกรมที่ลดความจำเป็นในส่วนของการสร้างภาษาโปรแกรม เพราะตัวแปลภาษาโปรแกรม (Compiler) มีการเตรียมเครื่องมือ และ สภาพแวดล้อมในการทำงาน (Development Environment) ที่พร้อมใช้งานได้ทันทีที่เรียกว่าคอมโพเนนต์ (Component) ที่อาจจะอยู่ในรูปแบบของสัญลักษณ์ กราฟิกรูปภาพ ปุ่ม ในรูปแบบวัตถุให้ลากวางบนหน้าจอการทำงาน และสามารถประมวลผลได้เทียบเท่าการเขียนโปรแกรมด้วยภาษาโปรแกรมปรกติ ข้อดีของการเขียนโปรแกรมเชิงจินตภาพนั้นคือการลดขั้นตอนที่ซับซ้อนในการ

พัฒนาโปรแกรม หรือโครงงาน มีการประยุกต์ใช้งานได้กับผู้ใช้ทุกระดับผ่านการแปลผลด้านตรรกศาสตร์เชิงสัญลักษณ์ และลดการจดจำคำสั่งที่แตกต่างกันในภาษาโปรแกรม (Programming Syntax) แต่กลับให้ความเข้าใจในเรื่องโครงสร้างของภาษาโปรแกรม (Programming Structure) ได้ดีกว่า การเขียนโปรแกรมในรูปแบบตัวอักษร (Text-Based Programming) [3] รูปแบบโปรแกรมเชิงจินตภาพที่ได้รับความนิยมในปัจจุบันที่เกี่ยวข้องกับการแทรกลำดับขั้นตอนการทำงานเชิงตรรกศาสตร์ คือ Scratch (<http://scratch.mit.edu>) และ Google Blockly

2.1.1 โปรแกรม Scratch



รูปที่ 1. โปรแกรม Scratch ของ สถาบันเทคโนโลยีแมสซาชูเซตส์ (MIT)

พัฒนาจากการวิจัยของสถาบันเทคโนโลยีแมสซาชูเซตส์ (MIT) ไปจนถึงรูปแบบการพัฒนาโปรแกรมรูปแบบของแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์เช่น App Inventor (<http://appinventor.mit.edu>) ก็เป็นรูปแบบการเขียนโปรแกรมเชิงจินตภาพอีกรูปแบบหนึ่งที่ผู้ใช้งานสามารถนำมาสร้างแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ได้อย่างสะดวก

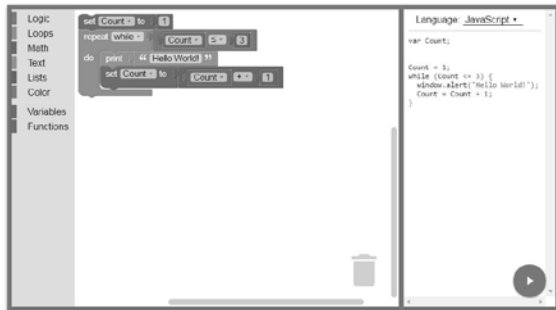


รูปที่ 2. โปรแกรม App Inventor ของ สถาบันเทคโนโลยีแมสซาชูเซตส์ (MIT)

2.1.2 Google Blockly

Blockly เป็นชุดพัฒนาสำหรับการสร้างโปรแกรมเชิงจินตภาพที่พัฒนาโดย Google [4] โดยรูปแบบของคอมโพเนนต์ที่

สามารถใช้งานได้ออยู่ในรูปแบบของ กราฟิกรูปภาพที่เรียกว่า บล็อก (Block) ที่ประกอบไปด้วยเงื่อนไขที่แตกต่างกัน นักพัฒนาที่ต้องการพัฒนาโปรแกรมเชิงจินตภาพร่วมกับโครงการที่ต้องการใช้งานสามารถเข้าไปดาวน์โหลดและติดตั้งชุดพัฒนาเพื่อนำมาใช้ร่วมกับการเขียนโปรแกรมประยุกต์ได้ทั้งรูปแบบของเว็บไซต์ และแอปพลิเคชันบนระบบปฏิบัติการ Android



รูปที่ 3. หน้าจอหลักของชุดพัฒนาโปรแกรมด้วย Blockly



รูปที่ 4. รูปแบบวิธีการ Define กำหนดโครงสร้างด้วย

ตัวเองบน Blockly

สำหรับการนำมาใช้ในหัวข้อนี้ได้เลือก Blockly มาใช้เป็นเครื่องมือในการพัฒนาสื่อการเรียนรู้ในรูปแบบโปรแกรมเชิงจินตภาพ เนื่องจาก Scratch ของ MIT เหมาะที่จะเป็นเครื่องมือในการฝึกแนวคิดเชิงคำนวณในระดับของเด็กประถม หรือมัธยมให้รู้จักรูปแบบของการเขียนโปรแกรม แต่ Blockly กลับเป็นเครื่องมือที่เน้นในส่วนของการเรียนภาษาโปรแกรมคอมพิวเตอร์ผ่านการออกแบบของชิ้นตัวต่อแบบบล็อก พร้อมกับส่วนขยายที่ต่อยอดไปสู่การประมวลผล (Interpreter) ให้กลายเป็นรูปแบบภาษาโปรแกรมได้

2.2 แนวคิดเชิงคำนวณอย่างเป็นระบบ (Computational Thinking)

แนวคิดเชิงคำนวณอย่างเป็นระบบเป็นกระบวนการคิดที่ต้องใช้ทักษะและเทคนิคเพื่อแก้ไขปัญหาอย่างเช่นที่ นักพัฒนาซอฟต์แวร์ (Software Developer) หรือ วิศวกรซอฟต์แวร์

(Software Engineer) ใช้ในการเขียนโปรแกรม ซึ่งแก่นแท้คือการแก้ปัญหา แบบมีลำดับขั้นตอนให้กลายเป็นเรื่องที่สายอาชีพอื่น สามารถนำแนวคิดลำดับขั้นตอนไปแก้ปัญหาในเชิงนามธรรม โดยทักษะที่มีอยู่ใน แนวคิดเชิงคำนวณอย่างเป็นระบบ ประกอบไปด้วยข้อมูลสำคัญต่อไปนี้ ได้แก่ Decomposition, การวิเคราะห์ปัญหาเชิงลึกในส่วนประกอบย่อยเพื่อศึกษาความซับซ้อนของผลลัพธ์หรือปัญหา หรือแจกแจงปัญหาไปสู่ส่วนประกอบย่อยเพื่อปรับปรุงให้ดีขึ้น Pattern Recognition การมองหารูปแบบของปัญหา(Pattern) ที่เกิดขึ้นซ้ำ เพื่อนำมาประเมินสถานการณ์ Pattern Generalization and Abstraction การมองภาพรวมเพื่อยืนยันสิ่งที่ป็นรายละเอียดปลีกย่อยแบบมุกกว้าง และ Algorithm Design การออกแบบลำดับการทำงานที่สามารถ กำหนดปัญหา เพื่อออกแบบแนวทางการปรับปรุงให้เกิดผลลัพธ์ตามที่กำหนด

- **Decomposition** การลงลึกเพื่อวิเคราะห์ส่วนประกอบย่อยเพื่อศึกษาความซับซ้อนของผลลัพธ์หรือปัญหา หรือแจกแจงปัญหาไปสู่ส่วนประกอบย่อยเพื่อปรับปรุงให้ดีขึ้น
- **Pattern Recognition** การมองหารูปแบบของปัญหา (Pattern) ที่เกิดขึ้นซ้ำ เพื่อนำมาประเมินสถานการณ์
- **Pattern Generalization and Abstraction** การมองภาพรวมเพื่อยืนยันสิ่งที่ป็นรายละเอียดปลีกย่อยแบบมุกกว้าง
- **Algorithm Design** การออกแบบลำดับการทำงานที่สามารถ แก้ปัญหา เพื่อออกแบบแนวทางการปรับปรุงให้เกิดผลลัพธ์ตามที่กำหนด

3. วิธีดำเนินการวิจัย

งานวิจัยนี้เป็นเชิงทดลองโดยการทำการทดสอบ บันทึกลงและประเมินผลการใช้แนวคิดเชิงคำนวณอย่างเป็นระบบมาเป็นส่วนหนึ่งในการออกแบบการเรียนการสอนโดยการนำบทเรียนออนไลน์ของ Research at Google: Computational Thinking[5] มาประยุกต์ใช้ร่วมในการสอน และนำโปรแกรมเชิงจินตภาพ Blockly เข้ามาเป็นส่วนประกอบในการพัฒนาส่วนของแบบฝึกหัดในห้องเรียน

3.1 กรอบแนวคิดการวิจัยด้านการออกแบบการเรียนรู้

ศึกษาคำอธิบาย และจุดประสงค์ของเนื้อหาวิชา การเขียนโปรแกรมแบบสองมิติ รหัสวิชา GT368 โดยการสอนตามบทเรียนปกติเกี่ยวกับการเขียนโปรแกรมพัฒนาเกมด้วยเกมส์เอ็นจิน และแทรกกิจกรรม โครงการย่อยสำหรับฝึกแก้ปัญหาด้วยการวางแผนคิดเชิงคำนวณอย่างเป็นระบบลงไป พร้อมออกแบบตัวแบบทดสอบในเชิงการช่วยเหลือสำหรับการเขียนโปรแกรมด้วยโปรแกรมเชิงจินตภาพ เพื่อให้ผู้เรียนสามารถตรวจสอบว่ารูปแบบการเขียนโปรแกรมของผู้เรียนนั้นทำงานได้อย่างถูกต้องหรือไม่ ผ่านการวาง Block ของโปรแกรมเชิงจินตภาพผ่านแอปพลิเคชัน พร้อมกำหนดโครงงานเกมที่ผู้เรียนต้องศึกษาด้วยตนเองภายใน 2 สัปดาห์เป็นการประเมินเพื่อวัดทักษะการแก้ปัญหาของผู้เรียนดีขึ้น และ ระยะเวลาในการพัฒนาโครงงานที่ลดลง

3.2 ประชากร และกลุ่มตัวอย่าง

นักศึกษารุ่นปีที่ 3 และปีที่ 4 จำนวน 30 คนที่ลงทะเบียนเรียนวิชา GT368 การเขียนโปรแกรมเกม 2 มิติ

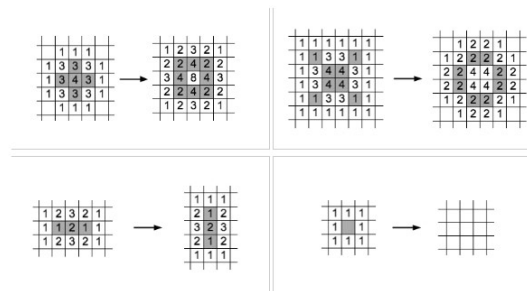
3.3 สมมติฐานของงานวิจัย

การแทรกแนวคิด และออกแบบกระบวนการจะเพิ่มประสิทธิภาพการแก้ปัญหา และทักษะการเขียนโปรแกรม อีกทั้งใช้เวลาแก้ปัญหาที่เร็วขึ้น

4. ระเบียบวิธีวิจัย

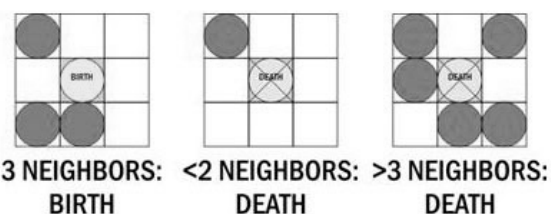
4.1 การแทรกแนวคิดเชิงคำนวณอย่างมีระบบ

ในการออกแบบการเรียนการสอนในแต่ละครั้งที่มีการอธิบายถึงคำสั่ง หรือฟังก์ชันของการเขียนโปรแกรมจะมีการแทรกเนื้อหาการเรียนผ่านอินเทอร์เน็ตด้วยแบบฝึกหัดของ Research at Google: Computational Thinking ในการสร้างกิจกรรมภายในห้องเรียนและแทรกบททดสอบของเนื้อหา และเก็บคะแนนย่อยของผู้เรียนรายบุคคลในแบบทดสอบท้ายการเรียน ตัวอย่างในงานวิจัยนี้เป็นบทเรียนสัปดาห์ที่ 5 ของการเรียนอย่างของทฤษฎี Game of Life รูปแบบลำดับกระบวนการที่ใช้แนวคิด Cellular Automata[6] โดยให้ผู้เรียนได้ลงมือทดสอบแบบฝึกหัดเบื้องต้นผ่านเว็บเบราว์เซอร์ด้วยตัวเอง (Robins et al., 2003) ดังรูปที่ 5 และ 6

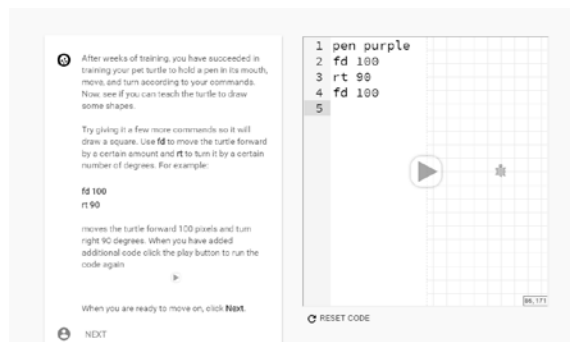


รูปที่ 5. แนวคิด Cellular Automata ในบทเรียนของ

Research at Google: Computational Thinking



รูปที่ 6. กฎของ Cellular Automata [7] เงื่อนไขการเกิด (Birth) การตาย (Death)

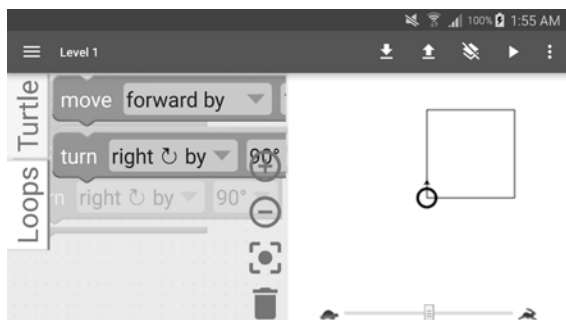


รูปที่ 7. การฝึกบทเรียน Geometry ผ่านบทเรียนออนไลน์

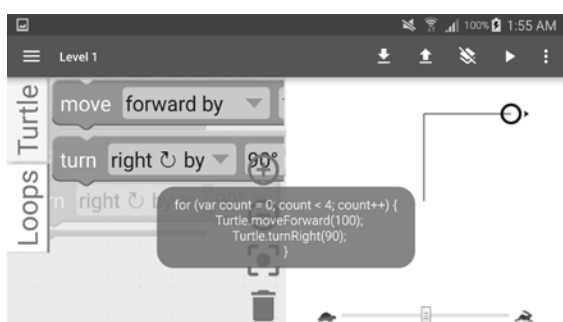
4.2 พัฒนาแอปพลิเคชันเกมด้วยโปรแกรมเชิงจินตภาพ

ส่วนของการตรวจสอบว่าผู้เรียนมีความเข้าใจในเนื้อหาการเรียนโปรแกรมผ่านการเรียนรู้บนบทเรียนบนเว็บไซต์ของ Research at Google: Computational Thinking ด้วยแนวคิดเชิงคำนวณอย่างเป็นระบบ และเนื้อหาด้านทฤษฎีของบทเรียนในห้องเรียน ก่อนที่จะทำการประเมินความรู้ความเข้าใจผ่านผลสัมฤทธิ์ทางการเรียนรู้ของผู้เรียน ผู้วิจัยได้พัฒนาเกมสำหรับบททวนการเรียนรู้ในรูปแบบโปรแกรมเชิงจินตภาพด้วย Blockly ของ Google ในรูปแบบของ แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ให้กับผู้เรียนได้ดาวน์โหลดไว้ใช้ประกอบกิจกรรมระหว่างการเรียนรู้และการเรียน และหลังการเรียนรู้

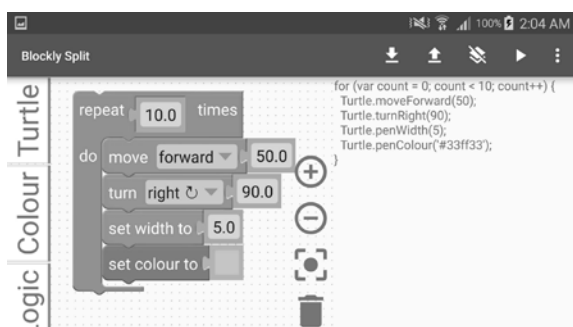
ในห้องเรียน ดังรูปที่ 8-11 เป็นการนำทดสอบการทำงานของแอปพลิเคชันที่พัฒนาขึ้น



รูปที่ 8. เกม Turtle สำหรับเรียนรู้การสร้าง Geometry บนแอปพลิเคชันมือถือ



รูปที่ 9. ผลลัพธ์ในการตรวจทานรูปแบบคำสั่งจากเกมที่พัฒนาให้แสดงผลในรูปแบบของภาษาโปรแกรม JavaScript



รูปที่ 10. หมวดหมู่ Console แสดงผลสำหรับการเรียงการทำงานของเกม

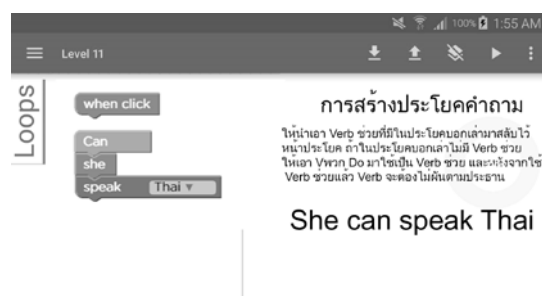
4.3 การออกแบบสภาพแวดล้อมและเพิ่มเงื่อนไขของเกม

หลังจากที่มีการพัฒนาแอปพลิเคชันด้วยโปรแกรมเชิงจินตภาพในรูปแบบเกมสำหรับทบทวน และใช้ประกอบกิจกรรมการเรียนการสอนในบทเรียนตามวิธีข้างต้นที่ดำเนินการมา จะมีการเพิ่มความซับซ้อนของตัวเกมออกมาทั้งหมด 3 ฉากผ่านเงื่อนไขที่มากขึ้น และระดับความซับซ้อนของเกมที่ต้องแก้ปัญหาต้องใช้ทักษะความรู้และความเข้าใจในเนื้อหา และการฝึกปฏิบัติทั้งหมดมาประยุกต์ใช้ ผ่านโปรแกรมเชิงจินตภาพ

ที่ถูกปรับแต่ง (Customization) ให้แตกต่างจากการเขียนโปรแกรม [8]



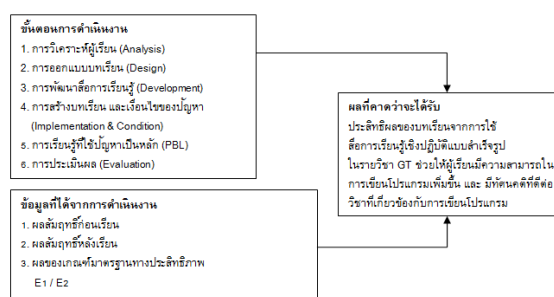
รูปที่ 11. สถานการณ์การประยุกต์การลำดับการทำงานให้เป็นเกมที่กำหนดเอง



รูปที่ 12. การออกแบบ Blockly ให้อยู่ในรูปแบบของบทเรียนวิชาการสร้างเกมภาษาอังกฤษ

4.4 กรอบแนวคิดด้านเนื้อหาสำหรับประเมินผลผู้เรียน

ในการวิจัยครั้งนี้ ผู้วิจัยได้กำหนดกรอบแนวคิดในการพัฒนาบทเรียนวิชา GT359 เพื่อวัดประสิทธิภาพของผู้เรียนในการเขียนโปรแกรมออกมาเป็น 5 ขั้นตอน



รูปที่ 13. แผนภาพอธิบายการดำเนินงานภายใต้กรอบแนวคิดการวิจัย

4.4.1. การวิเคราะห์ผู้เรียน (Analysis)

จากกลุ่มนักศึกษาชั้นปีที่ 4 วิทยาลัยนวัตกรรมการเทคโนโลยีและวิศวกรรมศาสตร์ สาขาการออกแบบเชิงโต้ตอบและพัฒนาเกมจำนวน 30 คน เป็นกลุ่มตัวอย่างในการดำเนินการใน

ขั้นตอนคือการวิเคราะห์ผู้เรียนผ่านการใช้แอปพลิเคชันเกมด้วยโปรแกรมเชิงจินตภาพ

4.4.2. การออกแบบบทเรียน (Design)

ออกแบบบทเรียนจากวิชาการพัฒนาเกม 2 มิติ รหัสวิชา GT368 ด้วยโปรแกรม Unity2D [9] โดยเนื้อหาของรายวิชาเป็นการเขียนโปรแกรมเกมด้วยภาษาโปรแกรม JavaScript และภาษาที่มีโครงสร้างใกล้เคียงกันอย่างภาษาโปรแกรม C# ควบคุมการทำงานของเกมที่ถูกออกแบบโดยใช้หลักแนวคิดการเขียนโปรแกรมเบื้องต้นมาใช้ในการแต่ละบทเรียน

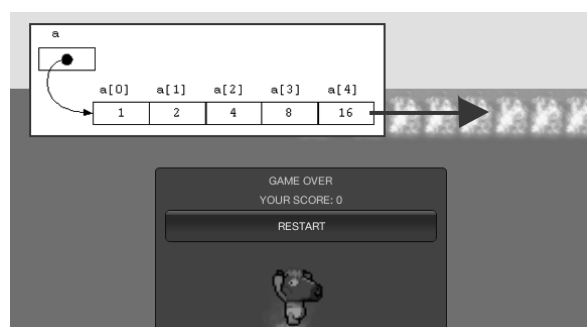
4.4.3. การพัฒนาสื่อการเรียนรู้ (Development)

พัฒนารูปแบบของสื่อการเรียนรู้ด้วยโปรแกรม Unity 2D ร่วมกับแบบจำลองสามมิติ โดยมีการเขียนคำสั่งภาษาโปรแกรมคอมพิวเตอร์ภาษา JavaScript และ ภาษา C# ใส่ไว้ในแบบจำลองแต่ละส่วนเพื่อให้ง่ายต่อการเรียกใช้งาน โดยสามารถทำงานแยกส่วนอย่างเป็นอิสระต่อกัน พร้อมกับปรับรูปแบบการนำเสนอในแบบจำลองให้สอดคล้องกับเนื้อหาวิชาการที่ต้องสอน โดยเน้นการสร้างเงื่อนไขเชิงปัญหาเป็นสิ่งที่ตั้งเข้าไปควบคุมการใช้งานกับผู้เรียน

4.4.4. การสร้างบทเรียน และเงื่อนไขของปัญหา (Implementation & Condition)

การใช้ความรู้จากบทเรียนร่วมกับสื่อการเรียนรู้เพื่อแก้ปัญหาโครงข่ายขั้นตอนข้อมูลเกี่ยวกับคำสั่งการเขียนโปรแกรม เช่นรูปที่

14



รูปที่ 14. ตัวอย่างการวนซ้ำ For...Loop และตัวแปรการเก็บข้อมูล Arrays

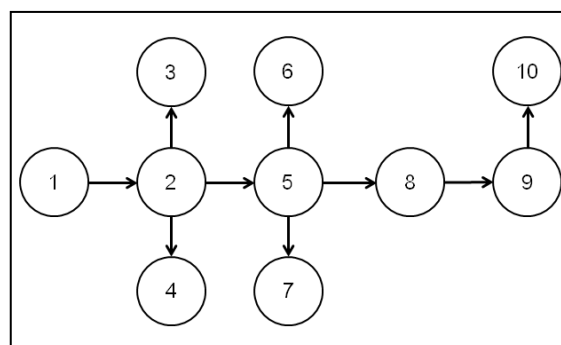
4.4.5. การประเมินผล (Evaluation)

ระยะเวลาในการพัฒนาโครงงานเกม 2 มิติด้วยตนเองโดยใช้ความรู้ทางด้านการเขียนโปรแกรมที่ได้ฝึกปฏิบัติผ่าน แนวคิดเชิงคำนวณอย่างเป็นระบบ และตัวช่วยเช่นแอปพลิเคชันสื่อการเรียนรู้ประเภทเกมด้วยโปรแกรมเชิงจินตภาพ โดยวัดผลทางกระบวนการทางสถิติเชิงพรรณนาทั่วไป และใช้เกณฑ์มาตรฐาน

ในส่วนของ ระยะเวลาที่สั้นลง และ ประสิทธิภาพของผลลัพธ์ตัวงานที่สมบูรณ์มาวัดผลว่ามากน้อยแค่ไหน

5. ผลการดำเนินงาน

หลังจากมีการใช้บทเรียนออนไลน์ และการพัฒนาเกมบนแอปพลิเคชันด้วยโปรแกรมเชิงจินตภาพที่ผู้วิจัยได้จัดทำให้แก่กลุ่มตัวอย่างเป็นที่เรียบร้อยแล้ว ขั้นตอนการวัดผลการเรียนรู้และการพัฒนาความก้าวหน้าของกลุ่มตัวอย่างคือการกำหนดปัญหาของโครงงานผ่านการสร้างเกมแนวเดินระนาบ (Platform-Side Scrolling Game) โดยมีเงื่อนไขของการเคลื่อนไหวของตัวละคร ระบบฟิสิกส์ของตัวละคร และ การใช้แนวคิดเชิงคำนวณอย่างเป็นระบบ เช่น การเกิดของจำนวนศัตรู การกำหนดทิศทางของการเดินอัตโนมัติ เงื่อนไขการสุ่มสิ่งของภายในเกม และการชนกันของวัตถุตามกฎของฟิสิกส์ โดยให้กลุ่มตัวอย่างทำการศึกษาด้วยตนเอง ขั้นตอนในการกำหนดการศึกษาระยะของการจัดทำโครงงานเกมของผู้เรียนหรือกลุ่มตัวอย่างได้ข้อมูลเป็นรูปแบบการแก้ปัญหาของกลุ่มตัวอย่างตลอดระยะเวลาที่ทำการทดลองผ่านการติดตามความก้าวหน้าของเนื้องานตาม สถานการณ์ของปัญหาดังภาพประกอบที่ 15



รูปที่ 15. รูปแบบการแก้ปัญหาแบ่งตามระดับความก้าวหน้าของกลุ่มตัวอย่าง

กำหนดตัวเลขแทนระดับความยาก ระดับความยากระดับที่ 1 แทนความรู้ความเข้าใจในการใช้งานเครื่องมือของโปรแกรม Unity ซึ่งเป็นปัญหาพื้นฐานที่กลุ่มตัวอย่างผู้เรียนสามารถเข้าถึงเมนูแถบเครื่องมือที่นอกเหนือจากบทเรียน โดยอาศัยการสังเกตและทดลองใช้งาน ระดับความยาก 2 ถึง 4 เป็นการคิดฟังก่อนการทำงานที่เริ่มซับซ้อนระดับปานกลาง เช่นกำหนดความท้าทายของโครงงานให้มีเงื่อนไขของการตรวจจบการชนกันของวัตถุผ่านการอธิบายความก้าวหน้าของโครงงานผ่านการให้ฟังก่อนคำสั่งการเขียนโปรแกรมที่ศึกษามาและตีความหมายของปัญหาด้วยทักษะที่พบทวนจากแนวคิดเชิงคำนวณอย่างเป็น

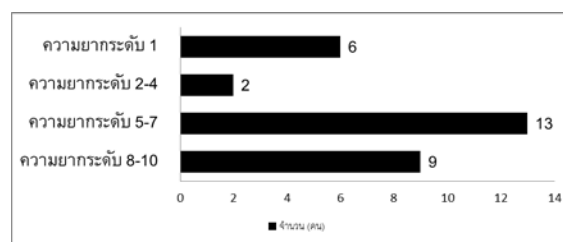
ระบบ และบททวนคำสั่งของการเขียน โปรแกรมภาษา JavaScript และ C# ด้วยแอปพลิเคชันเกมโปรแกรมเชิงจินตภาพที่ได้พัฒนาขึ้นมา ในส่วนของระดับ 5 ไปจนถึง 7 เป็นการคิดรูปแบบที่ออกแบบปัญหาด้วยตนเองเพิ่มเติมผ่านข้อกำหนดเดิมของตัวโครงงานที่กำหนดไว้เช่น กลุ่มตัวอย่างผู้เรียนสามารถใช้รูปแบบการสุ่มวัตถุที่ตกลงมาจากท้องฟ้าให้เป็นศัตรู หรือบางสิ่งๆที่สร้างความเสียหายได้โดยอาศัยรูปแบบการทำงานจากฟังก์ชันเดิมที่มีอยู่มาออกแบบเป็นฟังก์ชันใหม่และสร้างเงื่อนไขของตัวระบบเกมและพัฒนาขึ้นจากระดับความยากที่ 5 นี้มีผู้เรียนบางคนสามารถกำหนดเงื่อนไขใหม่ และเปลี่ยนรูปแบบของโครงงานให้แตกต่างโดยการกำหนดปัญหาที่นอกเหนือจากที่กำหนด หรือปรากฏในบทเรียนให้ออกมาเป็นรูปแบบใหม่พร้อมแนวทางการแก้ปัญหาสำหรับผู้เล่นในเกมด้วยตนเอง ดังผลลัพธ์ตัวอย่างตามรูปที่ 16



รูปที่ 16. ตัวอย่างเกม Side-Scrolling Platform ที่กำหนดให้พัฒนาขึ้นผ่านโครงงานพัฒนาเกมโดยมีการชนกันของวัตถุ การควบคุมตัวละคร และ กำหนดการ โจมตีศัตรู

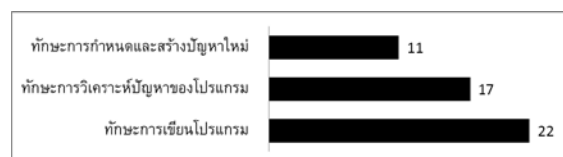
จากตัวอย่างข้างต้นคือการสร้างแนวคิดของเกมในโครงงานใหม่ คืออัตราการก้าวหน้าจากระดับ 5 ไปยังระดับที่ 8-10 ได้เมื่อมีการวัดระดับความก้าวหน้าจากรูปแบบของตัวโครงงาน ในระยะเวลา 2 สัปดาห์ พบว่าจำนวนกลุ่มตัวอย่าง 13 คนสามารถพัฒนาโครงงานเกมในรายวิชาได้ตามเงื่อนไขของปัญหาที่กำหนดได้ตรงเวลา กลุ่มตัวอย่างจำนวน 9 คนสามารถพัฒนาเกมได้ตรงเวลาด้วยความซับซ้อนของเกม หรือตัวโครงงานมีความแตกต่าง และความก้าวหน้าของผลงานดีกว่าข้อกำหนด ทั้งตัวปัญหาเดิมพร้อมกับรูปแบบปัญหาใหม่ที่มีการพัฒนาเพิ่มเติมเข้ามาในโครงงานเกม และอีก 8 คนไม่สามารถพัฒนาโครงงานเกมได้ตามเวลาที่กำหนด แต่มีความเข้าใจและอธิบายการทำงานของคำสั่ง ตัวแปรของภาษาโปรแกรมได้

แต่ไม่สามารถแก้ปัญหาที่กำหนดไว้ในโครงงานได้ ดังผลลัพธ์ในรูปที่ 17



รูปที่ 17. จำนวนของกลุ่มตัวอย่างแบ่งตามระดับความยากของการแก้ปัญหา

การประเมินกลุ่มตัวอย่างผู้เรียนเกี่ยวกับการแก้ปัญหา และความเข้าใจในการเขียนภาษาโปรแกรม ผ่านการใช้แนวคิดเชิงคำนวณอย่างเป็นระบบ และโปรแกรมเชิงจินตภาพเป็นสื่อการเรียนรู้ และบททวนหลังบทเรียนตลอดเวลา 5 สัปดาห์แรกในการเรียนการสอน ได้ผลลัพธ์ว่าผู้เรียนจำนวน 11 คนสามารถกำหนดปัญหา และแนวทางแก้ปัญหาใหม่ได้ด้วยตนเอง 17 คนมีทักษะการวิเคราะห์ปัญหาที่ดีขึ้น และสามารถนำรูปแบบปัญหาของโปรแกรมไปวิเคราะห์ผ่านแอปพลิเคชันโปรแกรมเชิงจินตภาพ และจากโปรแกรมเชิงจินตภาพนี้ช่วยให้กลุ่มตัวอย่างเข้าใจการทำงานของในรูปแบบฟังก์ชันของโปรแกรม



และการเขียนภาษาโปรแกรมเพิ่มขึ้นจำนวน 22 คนดังผลลัพธ์ที่ปรากฏในรูปที่ 18

รูปที่ 18. ผลลัพธ์จากแบบประเมินความก้าวหน้าของกลุ่มตัวอย่างที่มีต่อการเขียนโปรแกรม หลังจากใช้แนวคิดเชิงคำนวณอย่างเป็นระบบ ร่วมกับโปรแกรมเชิงจินตภาพเป็นสื่อการเรียนรู้เพิ่มเติมร่วมกับการออกแบบการเรียนการสอนในห้องเรียน

6.สรุปผลการทดลอง

แนวคิดเชิงคำนวณอย่างเป็นระบบเป็นแนวทางที่น่าสนใจในการนำมาทบทวนแก่ผู้เรียนภาษาโปรแกรมที่ยังไม่เข้าใจการแก้ปัญหาโปรแกรมตั้งแต่ระดับพื้นฐาน ไปจนถึงการประยุกต์ใช้งานที่ซับซ้อน ซึ่งหากนำแนวคิดเชิงคำนวณอย่างเป็นระบบมานำเสนอร่วมกับการเขียน โปรแกรมเชิงจินตภาพในรูปแบบสื่อการเรียนรู้ หรือเครื่องมือช่วยเหลือแก่ผู้เรียนจะ

สามารถเพิ่มระดับการพัฒนาทักษะของผู้เรียนให้สามารถเข้าใจ
คำสั่ง การทำงาน ไปจนถึงฟังก์ชันที่ซับซ้อนของภาษาโปรแกรม
วิเคราะห์ปัญหา ตรวจสอบรูปแบบการทำงาน และลงมือ
แก้ปัญหาที่เกิดขึ้นได้ในระยะเวลาที่สั้นลงกว่ารูปแบบการเรียน
การสอนแบบเดิม

เอกสารอ้างอิง

- [1] A. Wilson, T. Hainey and T. Conolly, "Evaluation of computer games developed by primary school children to gauge understanding of programming concepts", *European Conference on Games Based Learning, Academic Conferences International Limited*, October 2012. pp. 549
- [2] B. Poolsawas, W. Niranatlamphong and W. Sukpongthai, "Using a Game Development Platform to Improve Advanced Programming Skills", Conference: International Conference on Learning Innovation in Science and Technology (ICLIST2016), Volume 5, January 2015, pp. 58-64
- [3] J. Maloney, M. Resnick, N. Rusk, B. Silverman and E. Eastmond, "The scratch programming language and environment", *ACM Transactions on Computing Education (TOCE)*, 10(4), 16.
- [4] N. Fraser, "Google blockly-a visual programming editor", URL: <http://code.google.com/p/blockly>. Accessed Jul 2016.
- [5] A. Martinez, "Google for Education: Computational Thinking for Educators", Research at Google. On-line. Available from Internet, <http://googleforeducation.blogspot.com/2015/07/computational-thinking-for-educators.html>, accessed 6 Feb 2016.
- [6] F. Wu, "Complexity and urban simulation: towards a computational laboratory", *Geography Research Forum*, Vol. 22, February 2016, pp. 22-40.
- [7] Gamasutra., An Intro to Cellular Automation. URL: http://www.gamasutra.com/view/feature/134736/an_intro_to_cellular_automation.php. Accessed Jun 2016.
- [8] A. Phelps, C. Egert, J. Bayliss, "Games in the classroom: using games as a motivator for studying computing: part 1", *IEEE Multimedia*, April-June, 5-8, 2009

- [9] P. Petridis, I. Dunwell, D. Panzoli, S. Arnab, A. Protopsaltis, M. Hendrix and S. Freitas, "Game Engines Selection Framework for High-Fidelity Serious Applications," *International Journal of Interactive Worlds*, vol. 2012, no. 418638, June. 2012.