

**โครงงานวิศวกรรมคอมพิวเตอร์**

**เรื่อง**

**เกมฝึกทักษะโปรแกรมมิ่ง**

**Coder**

จัดทำโดย

|  |
| --- |
| 1. นาย จิรพัชร์ เหลืองรุ่งเกียรติ 5930300143 2. นาย อติวิชญ์ ม่วงศร 5930300968 |

ได้รับพิจารณาเห็นชอบโดย

|  |  |
| --- | --- |
| อาจารย์ที่ปรึกษาโครงงาน | …………………………………………………………………………………………… |
|  | ( อ.ดร.อดิศักดิ์ สุภีสุน ) |
| กรรมการโครงงาน | …………………………………………………………………………………………… |
|  | ( อาจารย์ กรวิทย์ ออกผล ) |
| กรรมการโครงงาน | …………………………………………………………………………………………… |
|  | ( ผศ.ดร.อนันต์ บรรหารสกุล ) |

**ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ศรีราชา**

**มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา**

**ปีการศึกษา 2562**

**เกมฝึกทักษะโปรแกรมมิ่ง**

**Coder**

|  |  |
| --- | --- |
| **ผู้จัดทำ** | 1. นาย จิรพัชร์ เหลืองรุ่งเกียรติ 5930300143 2. นาย อติวิชญ์ ม่วงศร 5930300968 |

|  |  |
| --- | --- |
| **อาจารย์ที่ปรึกษา** | อาจารย์ อดิศักดิ์ สุภีสุน |

|  |  |
| --- | --- |
| **หัวหน้าสาขาวิศวกรรมคอมพิวเตอร์** | อาจารย์ ประวิทย์ ชุมชู |

**คำสำคัญ (Key Words)**

* + **เกมฝึกเขียนโปรแกรม**
  + **คอมไพเลอร์เบื้องต้น**
  + **เกมคอมไพเลอร์**

**หลักการและเหตุผล**

เนื่องจากการศึกษาผลสัมฤทธิ์โครงการเตรียมความพร้อมทางด้านคอมพิวเตอร์ของนักศึกษาใหม่ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยกรุงเทพ พบว่ามีนิสิตสอบผ่านวิชาโปรแกรมมิ่งร้อยละ 20.75 และเหตุผลที่มีนิสิตสอบตกเพราะว่าไม่เข้าใจบทเรียนและการสอนในชั้นเรียนไม่มีความดึงดูด[1] เราจึงคิดว่าจะใช้เกมสอนการโปรแกรมเนื่องจากเกมมีความสนุก และจะทำให้ผู้เล่นสามารถเข้าใจบทเรียนของการโปรแกรมมิ่งได้มากขึ้น ประโยชน์ที่จะได้รับจากโครงงานนี้คือผู้เล่นจะได้ความรู้เกี่ยวกับการโปรมแกรมมิ่งมากขึ้นและได้รับความสนุกสนานในการเล่นเกม

**วัตถุประสงค์**

1. เพื่อให้ผู้ที่ต้องการศึกษาโปรแกรมมิ่งมีทักษะทางด้านตรรกะโปรแกรมมิ่งเพิ่มขึ้น

2. เพื่อให้นิสิต นักศึกษาที่มีความสนใจศึกษาต่อหรือกำลังศึกษาแต่ยังขาดทักษะพื้นฐานด้านโปรแกรมมิ่งได้เพิ่มทักษะด้านโปรแกรมมิ่งมากขึ้น

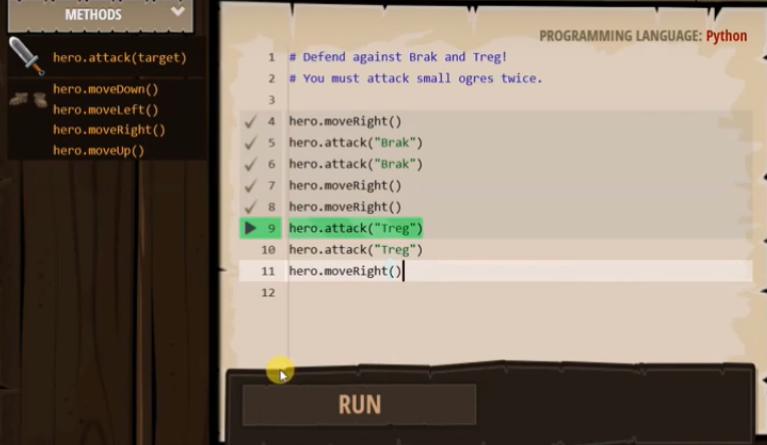
3. เพื่อให้เกิดการเรียนรู้อย่างมีประสิทธิภาพที่ดียึ่งขึ้นระหว่างผู้เรียนกับผู้สอน

**ปัญหาหรือประโยชน์เป็นเหตุผลให้พัฒนาระบบ**

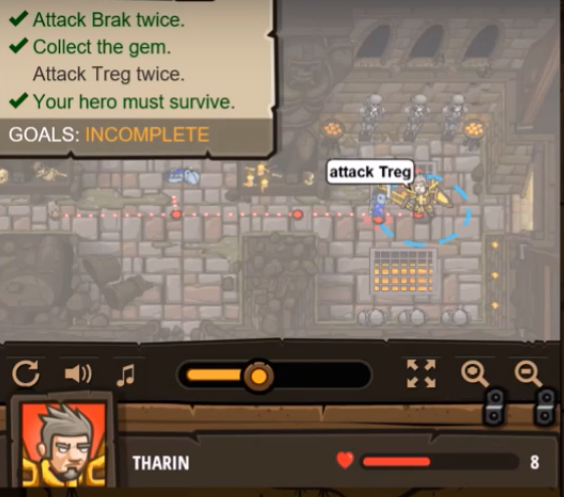
เนื่องจากประโยชน์ที่ได้รับจากการศึกษาการเขียนโค้ดโปรแกรมมิ่งพร้อมกับการสร้างความสนุกให้กับผู้เล่น นอกจากสามารถทำให้ผู้เล่นได้เรียนรู้ฝึกทักษะด้านโปแกรมมิ่งเพิ่มเติมแล้วยังช่วยให้ผู้เล่นมีทัศนคติที่ดีกับการเขียนโปรแกรมมิ่งอีกด้วย อีกทั้งยังลดปัญหาความเข้าใจที่ไม่ตรงกันระหว่างผู้สอนและผู้เรียนอีกด้วย ซึ่งทำให้สามารถลดเวลาถาม-ตอบในช่วงที่ไม่เข้าใจของผู้เรียนและเพิ่มเวลาในการเรียนรู้อย่างอื่นอีกด้วย

**งานที่เกี่ยวข้อง**

**code combat**



ภาพที่ 1 ตัวอย่างการพิมพ์โค้ดของเกม code combat

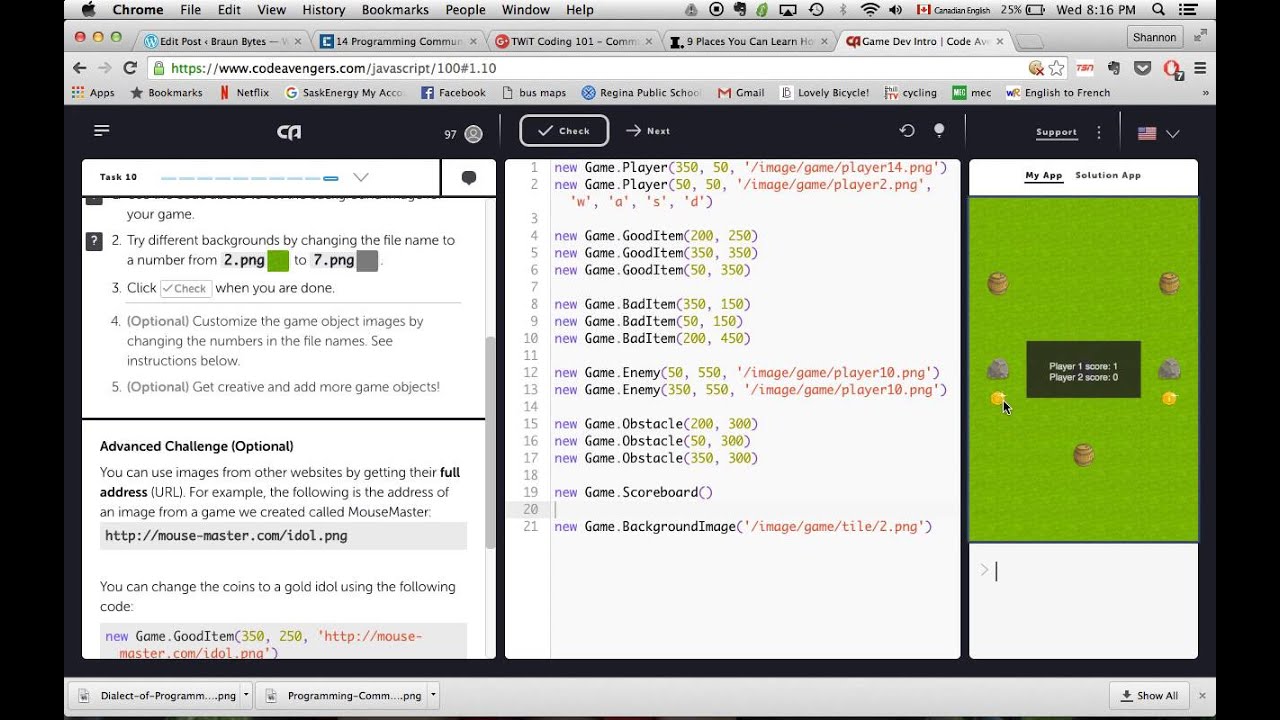


ภาพที่ 2 แสดงการดำเนินการของโค้ดที่พิมพ์บนหน้าเกม

จากภาพที่ 1 ตัวเกมมีหน้าต่างเพื่อให้ผู้เล่นสามารถเขียนโค้ดได้ และแสดงคำสั่งที่ผู้เล่นสามารถใช้ได้ในด่านนั้น จากภาพที่ 2 ภายในเกมผู้เล่นต้องควบคุมตัวละครให้ผ่านด่านไปได้ โดยจะแบ่งเป็นด่านซึ่งแต่ละด่านจะทำให้ผู้เล่นได้ใช้กระบวนการคิดในการพิมพ์โค้คว่าควรใช้คำสั่งไหนในการเอาชนะเงื่อนไขในแต่ละด่าน โดยต้องคำนึงถึงจุดหมายที่ตัวละครต้องไป และบางด่านต้องคำนึงถึงโจทย์ที่เกมระบุไว้

ข้อดี: สามารถดูได้ว่าสิ่งที่ตัวละครกำลังทำคืออะไร และจะมีกล่องข้อความแสดงสถานะขึ้นมาด้านบนตัวละครว่าตัวละครทำอะไรอยู่บริเวณบนหัวตัวละครจากภาพที่ 2

**code avenger**

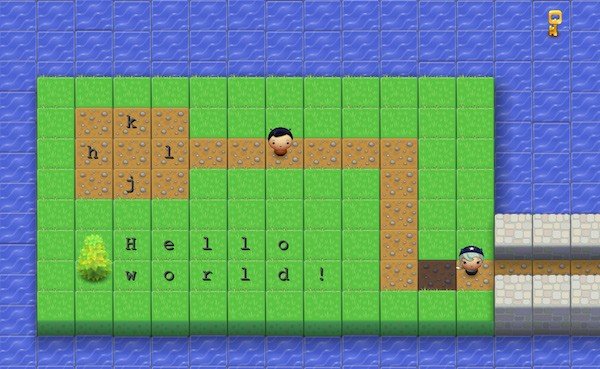


ภาพที่ 3 ตัวอย่าง ของเกม code avenger

เกมสอนให้ผู้เล่นได้เรียนรู้จากโค้ดที่มีให้อยู่แล้ว จากภาพที่ 3 ในหน้าต่างตรงกลางผู้เล่นจะมีส่วนของโค้ดที่เกมเขียนไว้อยู่แล้ว เกมต้องการให้ผู้เล่นปรับแต่งโค้ด โดยมีแนวทางจากโค้ดต้นแบบที่ จากภาพที่ 3 หน้าต่างทางขวาไว้เพื่อดูว่าการทำงานของโค้ดเป็นอย่างไร ผู้เล่นสามารถเลือกระดับการเรียนรู้ก่อนเข้าเรียนได้

ข้อดี: สามารถดูได้ว่าสิ่งที่ตัวละครกำลังทำ ผ่านไปแล้วกี่เปอร์เซ็นต์ของโค้ดที่เขียน

**vim adventure**



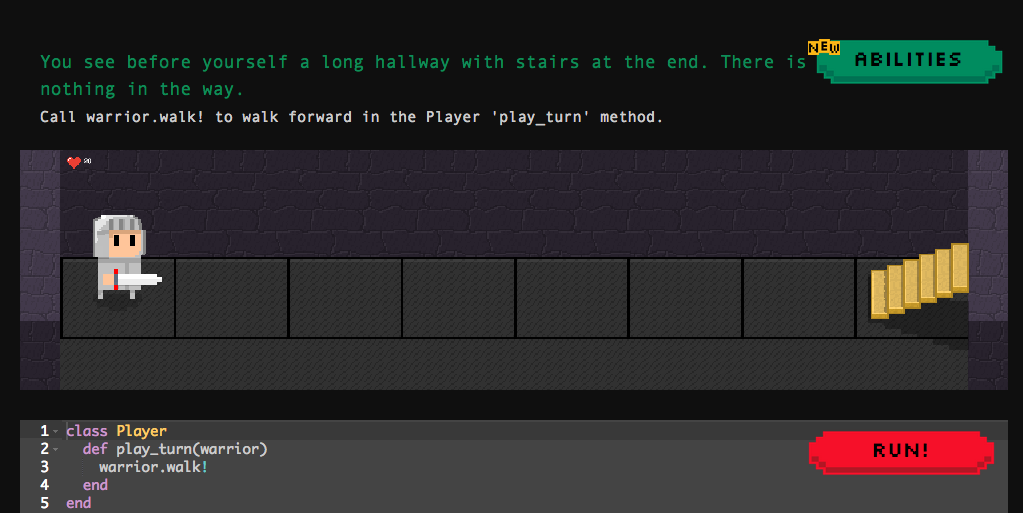
ภาพที่ 4 ตัวอย่างเกม vim adventure

ตัวเกมให้ผู้เล่นควบคุมตัวละครเพื่อให้ตัวละครสามมารถไปยังจุดหมาย หรือเส้นชัยที่เกมเป็นคนกำหนด โดยการเล่นให้ผู้เล่นบังคับตัวละครจากปุ่มคีย์บอร์ดจากภาพที่ 4 ผู้เล่นสามารถสั่งให้ตัวละครเดินไปทางขวาโดยการกดปุ่ม “L” หรือให้ตัวละครเดินขึ้นโดยการกดปุ่ม “K” ซึ่งบางปุ่มมีความสามารถที่แตกต่างกันออก โดยความสามารถนั้นผู้เล่นจะได้เพิ่มทีละอย่างเมื่อผ่านแต่ละช่วงของเกม

ข้อดี: เกมสอนเหตุผลที่ใช้ในการแก้ปัญหาต่าง ๆ เกี่ยวกับการใช้คอมพิวเตอร์

ข้อเสีย: เกมไม่ได้สอนทักษะการเขียนโค้ดผู้เล่น

**rudy warrior**

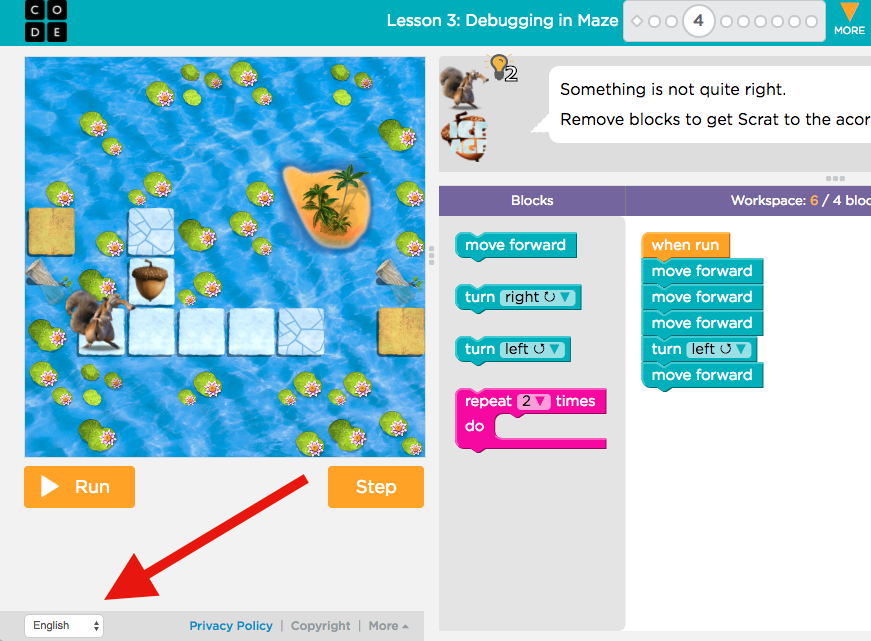


ภาพที่ 5 ตัวอย่างหน้าแสดงผลเกม และตัวอย่างโค้ดในเกม

เกมให้ผู้เล่นเขียนโค้ดเพื่อควบคุมตัวละครในเกม ให้ทำตามคำสั่งที่ผู้เล่นใส่ เกมแบ่งเป็นด่าน จากภาพที่ 5 แต่ละด่านจะมีต้นแบบของโค้ดให้ผู้เล่นเขียนโค้ดเพื่อแก้ปัญหา ลักษณะของเกมจะดำเนินในแนวระนาบเพียงอย่างเดียว

ข้อเสีย: ผู้เล่นสามารถสับสนได้หากไม่ทราบว่า Object หรือ Function คืออะไร

**code.org**



ภาพที่ 6 แสดงการทำงานของเกม code.org

ตัวเกมแบ่งออกเป็นข้อ โดยแต่ละข้อเป็นโจทย์ที่มีความคล้ายกัน คือให้ผู้เล่นเดินไปยังจุดหมายที่เกมกำหนด เกมไม่ได้ให้ผู้เล่นเขียนโค้ดภาษาคอมพิวเตอร์โดยตรง แต่ให้ผู้เล่นหยิบบล็อคที่เกมเป็นผู้กำหนดให้จากภาพที่ 6 ซึ่งแต่ละด่านนั้นเกมจะให้มาแตกต่างกัน โดยการกระทำของตัวละครจะเรียงตามลำดับบล็อคที่ผู้เล่นหยิบมาวางต่อกัน

ข้อเสีย: เกมไม่ได้ให้ผู้เล่นได้แก้ไข้โค้ดอย่างอิสระ เกมไม่ให้ผู้เล่นได้เขียนโค้ดนอกจากบล็อคที่ให้มาได้

**code monkey**



ภาพที่ 7 ตัวอย่างโค้ดของเกม code monkey

ตัวเกมจะให้ผู้เล่นควบคุมตัวละคร โดยต้องทำให้ตัวละครเดินไปเก็บกล้วยให้ได้ ถึงจะผ่านด่าน

โดยการที่ให้ผู้เล่นควบคุมลิงผ่านการเขียนโค้ด เพื่อให้ลิงแสดงการกระทำตามที่ผู้เล่นใส่ ด้วยตัวอย่างของโค้ดที่ให้ไม่ได้ซับซ้อน เช่น จากภาพที่ 7 ผู้เล่นใส่คำสั่งว่า “step 13” step คือการให้ลิงเดินหน้า หลังจากนั้นตามด้วยตัวเลข เป็นจำนวนก้าวที่ลิงจะเดินไปยังทิศทางปัจจุบันที่ลิงกำลังหันหน้าอยู่

**human resource machine**



ภาพที่ 8 ตัวอย่างเกม human resource manchin

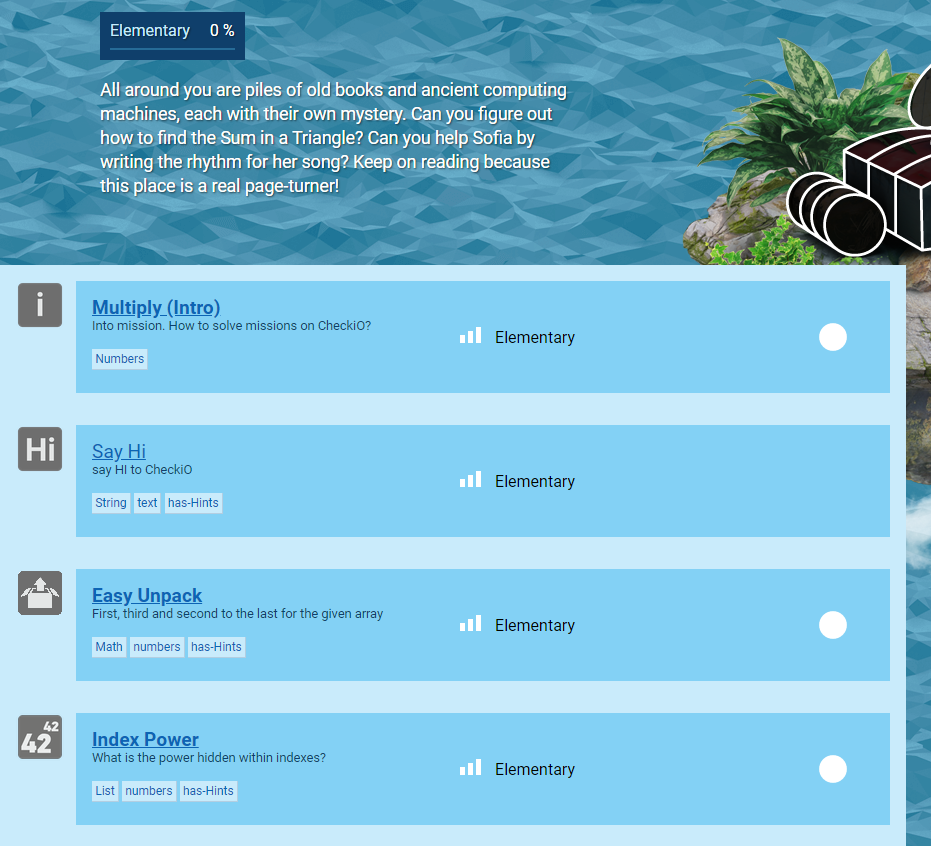
ตัวเกมแบ่งเป็นด่าน โดยแต่ละด่านจะมีโจทย์ให้ผู้เล่นทำตามโจทย์ โดยการนำกล่องตัวเลขจากช่อง IN ไปวางยังช่อง OUT โดยคำนึงถึงโจทย์ในด่านนั้น ภาพในเกมตัวละครมีความสามารถหลากหลาย เช่น ลบเลขไปหนึ่งจากเลขที่ตัวละครถืออยู่ จากภาพที่ 8 ผู้เล่นต้องหยิบบล็อคมาวางในหน้าต่าง execute เพื่อให้ตัวละครทำตามคำสั่งที่ผู้เล่นต้องการ เป็นลำดับจากบนลงล่าง ส่วนของคำสั่งที่ทำตามเงื่อนไข หรือเป็นลูปจะมีลูกศรชี้ เพื่อบอกว่าจะไปยังบรรทัดไหน ของการทำงานขั้นต่อไป

**checkio**



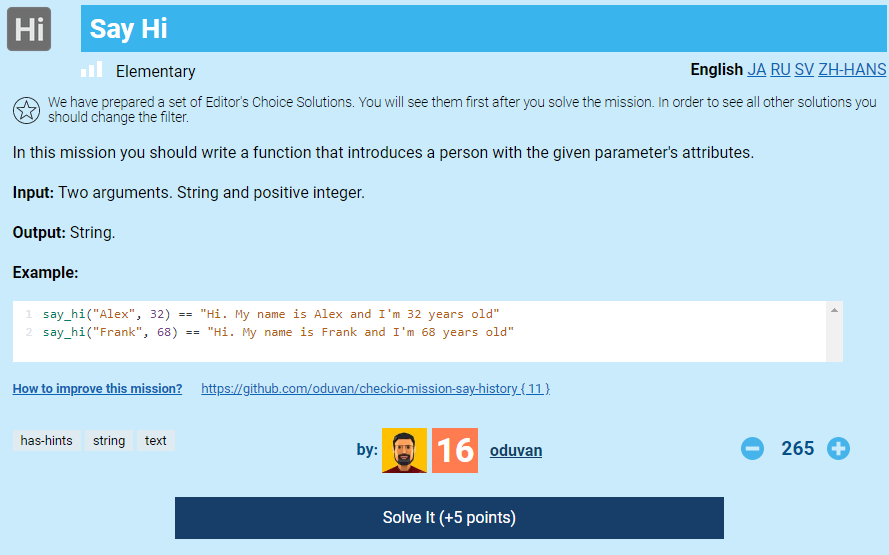
ภาพที่ 9 ตัวอย่างเกม checkio เมื่อเข้ามาในเกมเริ่มแรก

จากภาพที่ 9 เกมจะให้ผู้เล่นเดินทางไปยังเกาะต่างๆ เพื่อเรียนรู้การเขียนโปรแกรมด่านต่างๆ ตามเนื้อหาของเกาะเหล่านั้น ในช่วงเริ่มเกมผู้เล่นต้องทำการผ่านเงื่อนไขของเกมก่อนจึงจะสามารถไปเล่นในเกาะถัดไปได้ ซึ่งบางเกาะก็ต่อจากเกาะที่ล็อคอยู่เช่นกัน จึงทำให้ผู้เล่นไม่สามารถไปเกาะนั้นได้ก่อนจะผ่านเกาะก่อนหน้า



ภาพที่ 10 ตัวอย่างด่านต่างๆ ภายในเกาะของเกม checkio

จากภาพที่ 10 พอเข้าไปในเกาะจะมีโจทย์ให้ผู้เล่นได้เรียนรู้ ผู้เล่นสามารถเลือกเรียนโจทย์ไหนก่อนก็ได้ พอทำได้แล้ว เปอร์เซ็นต์ด้านบนจะเพิ่มขึ้น เพื่อบอกว่าผู้เล่นเรียนรู้จากเกาะนั้นแล้วกี่เปอร์เซ็นต์



ภาพที่ 11 ตัวอย่างโจทย์ของด่าน ภายในเกม checkio

จากภาพที่ 11 ผู้เล่นกดที่โจทย์จะมีเงื่อนไขให้ผู้เล่นแก้ปัญหาตามที่เงื่อนต้องการ จากภาพที่ 11 ผู้เล่นสามารถกดปุ่ม ‘Solve It’ เพื่อทำการแก้ไขโค้ด ตัวเกมก็จะให้ผู้เล่นแก้โค้ดตามภาษาที่ผู้เล่นได้กำหนดไว้ตอนลงทะเบียน โดยภาษาที่ผู้เล่นเลือกนั่นจะมีรูปแบบการเขียนเหมือนกับภาษาจริง

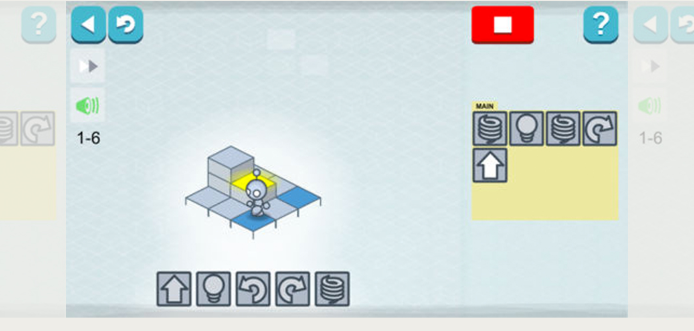
**coding game**



ภาพที่ 12 ตัวอย่างการผ่านด่านของเกม coding game

เกมมีความซับซ้อนของการเขียนโค้ดสูง และภายในเกมผู้เล่นสามารถเขียนโค้ดได้ตามความต้องการ และรูปแบบการเล่นของเกมนี้คือผู้เล่นต้องเอาชนะมอนเตอร์ที่เกมสร้างขึ้นมา จากภาพที่ 12 ภาษาที่ใช้ในการเขียนโค้ดมีรูปแบบภาษาที่เหมือนกับภาษาที่ใช้ในการเขียนโค็ดจริง บางด่านหากโค้ดที่ผู้เล่นเขียนมีช่องโห่วเกมจะไม่ทำงาน

**lightbot and lightbot jr.**



ภาพที่ 13 ตัวอย่างเกม lightbot and lightbot jr

ตัวเกมให้ผู้เล่นวางบล็อคควบคุมตัวละครให้ทำตามลงในช่อง Main ซึ่งเมื่อผู้เล่นกด Run ตัวละครจะทำตามคำสั่งเป็นลำดับตามที่ผู้เล่นวางไว้รายบรรทัดตามภาพที่ 13 ตัวเกมอธิบายการทำงานที่ตัวละครสามารถทำได้ในรูปของบล็อค เกมไม่สามารถทำงานในลักษณะของลูปได้

**ประเภทของเกม**

เกมที่กล่าวมาข้างต้นสามารถแบ่งประเภทออกได้เป็น 3 ประเภท

1. เกมฝึกโปรแกรมมิ่งแบบวางบล็อค ประกอบไปด้วยเกม code.org, human resource machine และ lightbot and lightbot jr.
2. เกมฝึกโปรแกรมมิ่งแบบโลจิกการทำงาน ประกอบไปด้วยเกม vim adventure
3. เกมฝึกโปรแกรมมิ่งแบบพิมพ์ ประกอบไปด้วยเกม coding game, checkio, code monkey, rudy warrior, code avenger, code combat



ภาพที่ 14 ตารางแสดงข้อดีข้อเสียของแต่ละเกม

จากภาพที่ 14 ผู้แต่งทำการเปรียบเทียบความอิสระของการใช้คำสั่งต่างๆ ภายในเกมของแต่ละเกม และคะแนนที่ผู้แต่งให้เป็นเพียงความคิดเห็นส่วนตัวของผู้แต่งเท่านั่นโดยการให้คะแนนจะแทนด้วยรูปมือซึ่งมีคะแนนสูงสุดอยู่ที่ 3 มือ และลดลงมาตามลำดับ

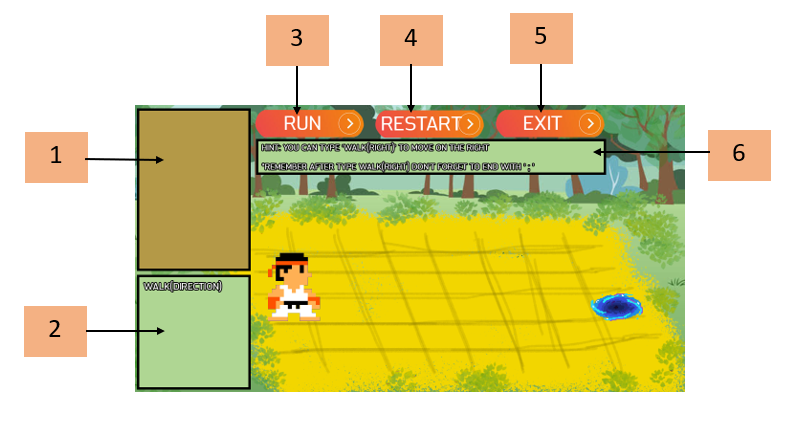
**งานวิจัยที่เกี่ยวข้อง**

จากงานวิจัยที่[1] กล่าวว่า “ผลสอบของนักเรียนที่ใช้เกมในการสอน สูงกว่าผลสอบของนักเรียนที่สอนแบบบรรยาย เพราะขาดความสนใจในเนื้อหา” และจากงานวิจัยที่[3] กล่าวว่า”นักเรียนมีผลสอบหลังเรียนสูงกว่าก่อนเรียน เพราะบทเรียนบนแท็บเล็ต คอมพิวเตอร์ มีภาพและเสียงที่เหมาะสมกับวัยของผู้เรียน” จากผลการวิจัยต่างๆในเรื่องที่เกมมีผลต่อการเรียนรู้ เมื่อผู้ทำวิจัยไปสำรวจและพบว่าจากงานวิจัยที่[7] การจะทำให้เกมมีผลการเรียนรู้ที่ดีอาจขึ้นอยู่กับการออกแบบเกมด้วย

**เป้าหมายและขอบเขตของโครงงาน**

1. พัฒนาเกมเพื่อเพิ่มทักษะด้านโปรแกรมมิ่งให้มากกว่าเดิม สามารถเรียนรู้ได้ง่าย
2. พัฒนาเกมเพื่อเสริมความรู้คำสั่งแบบเงื่อนไขคือ if-else และคำสั่งแบบลูปคือ while
3. ตัวเกมสามารถเล่นได้บนระบบ window10 เท่านั้นไม่ได้ทำงานบน web browser
4. เกมเป็นแบบ Single-player / Offline
5. เกมเป็นภาพแบบ 2D
6. เกมจะประกอบไปด้วยเวิลด์ทั้งหมด 3 เวิลด์แต่ละเวิลด์มี 8 ด่าน และด่านสอนผู้เล่น 6 ด่าน รวมทั้งหมด 30 ด่าน
7. เกมไม่รองรับการทำงานบนโทรศัพท์
8. เกมไม่รองรับให้ผู้เล่นสร้างฟังค์ชั่นได้
9. เกมไม่รองรับการกำหนดตัวแปรจากผู้เล่น
10. เกมจะประกอบไปด้วยด่านทั้งหมด 30 ด่านโดยรวมด่านสอนใช้คำสั่งให้ผู้เล่น
11. เมื่อผู้เล่นเขียนคำสั่งผิดเกมสามารถแจ้งเตือนได้
12. เกมสามารถทำการบันทึกได้

**รายละเอียดของการพัฒนา**



ภาพที่ 15 ตัวอย่างหน้าต่างของเกม

1. ส่วนที่ให้ผู้เล่นใส่คำสั่งควบคุมตัวละคร
2. ส่วนที่บอกคำสั่งที่ผู้เล่นสามารถใช้ได้
3. ปุ่มสั่งให้แสดงผลตามที่ผู้เล่นได้เขียนไป
4. ปุ่มเริ่มด่านนั่นๆใหม่
5. ปุ่มออกจากเกม
6. คำสั่งที่ให้ผู้เล่นทำตามถึงจะผ่าน

**เนื้อเรื่องย่อ**

ภายในเกมผู้เล่นสามารถพิมพ์ข้อความ ที่เป็นคำสั่งที่เกมกำหนดขึ้นมา เพื่อแสดงผลบนตัวเกมได้ ภายในเกมจะประกอบไปด้วยตัวละครของผู้เล่นและอุปสรรคในการผ่านแต่ละด่านโดยผู้เล่นต้องใช้การเขียนโปรแกรมเพื่อผ่านอุปสรรคในแต่ละด่านให้ได้ตามที่แต่ละด่านกำหนดเงื่อนไขเอาไว้ โดยเกมจะมีทั้งหมด 30 ด่านแบ่งเป็นด่านสอนผู้เล่น 6 ด่าน ส่วนอีก 24 ด่านจะแบ่งเป็นเวิลด์โดยแบ่งเป็นเวิลด์ทะเลทรายจำนวน 8 ด่าน, เวิลด์น้ำแข็ง 8 ด่าน, เวิลด์ลาวา 8 ด่าน และในเกมตัวละครของผู้เล่นสามารถแปลงร่างโดยการเก็บไอเทมเพื่อผ่านอุปสรรคต่างๆ ได้

**ความรู้พื้นฐาน เทคนิค เทคโนโลยีและเครื่องมือที่ใช้**

1. ความรู้พื้นฐาน และเทคนิค OOP

รูปแบบการทำงานของ OOP หรือ Object Oriented Programming ซึ่งมีประโยชน์ทางด้านการซ่อนข้อมูลให้อยู่ในคลาส ไม่ให้มีการรั่วไหลออกนอกคลาสหากเราไม่ได้ทำการเรียก ป้องกันการเข้าถึงข้อมูลจากบุคคลที่สาม มีการสืบทอดสามารถนำมาใช้กับตัวละครซึ่งมีความสามารถที่แตกต่างกัน แต่โดยทั่วไปตัวละคร หรือศัตรูมีความสามารถบางอย่างที่มีเหมือนกันอย่างเช่น ความสามารถในการเดิน

1. ใช้ภาษา Java ในการพัฒนา

โดยปกติจากการทำงานของภาษา Java มีความสอดคล้อง และเขียนได้ง่ายเมื่อต้องการเขียนโปรแกรมในลักษณะของ Object ซึ่งสอดคล้องกับความรู้พื้นฐานข้อก่อนหน้า

1. ความรู้พื้นฐาน และเทคนิคการทำงานของ Compiler

หลักพื้นฐานของ compiler high language programming คือการนำสิ่งที่ผู้ใช้พิมพ์มาแปลงให้เป็นภาษาเครื่องแล้วคอมพิวเตอร์จึงทำงานจากภาษาเครื่องนั้น ซึ่งจากโครงงานนี้ได้หยิบลักษะของ compiler ทีมีการแปลงภาษาระดับสูงให้เป็นภาษาที่คำสั่งภายในเกมเข้าใจ

**รายละเอียดระบบที่จะพัฒนา**

ฟังก์ชั่นที่ตัวเกมจะต้องมีโดยพื้นฐานคือ ส่วนที่นำข้อมูลเข้า ส่วนประมวลผล ส่วนการแสดงผล

**ส่วนนำข้อมูลเข้า**

หมายเลข 1 จากภาพที่ 15 ภายในเกมจะมีช่องไว้สำหรับรับอินพุตที่ผู้เล่นใส่มาได้ และปุ่ม 2 ปุ่ม คือปุ่มรัน สำหรับทำการดึงอินพุตที่ผู้เล่นใส่ไปประมวลผล และปุ่มรีเซ็ท สำหรับรีเซ็ทค่าบนเกมของด่านนั้นๆ เป็นการเริ่มต้นใหม่

**ปุ่มรัน**

หมายเลข 3 จากภาพที่ 15 เมื่อผู้เล่นกด เกมจะดึงโค้ดที่ผู้เล่นได้พิมพ์ไว้ก่อนหน้าไปประมวลผล

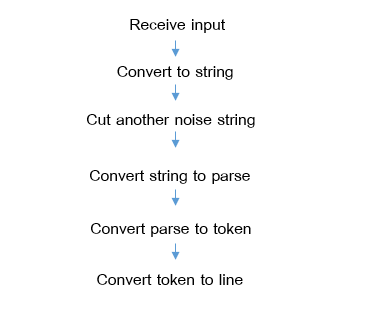
**ปุ่มรีเซ็ท**

หมายเลข 4 จากภาพที่ 15 เมื่อผู้เล่นกด เกมจะทำการรีเซ็ทค่าที่ผู้เล่นเคยได้ ในด่านนั้นๆ กลับไปเป็นค่าเริ่มต้นของด่านนั้นๆ

**ส่วนประมวลผล**

จากภาพที่ 16 อธิบายการทำงานได้ต่อไปนี้

1. เมื่อผู้เล่นทำการกดปุ่มรัน จะรับค่าสตริงจากที่ผู้เล่นพิมพ์มาประมวลผล
2. เมื่อได้ค่าแล้วจะนำคำสั่งที่ผู้เล่นพิมพ์มาตัดข้อความว่าง หรือเว้นบรรทัดออก
3. เมื่อทำการตัดข้อความที่ไม่จำเป็นแล้ว ทำการแบ่งย่อยข้อความให้เป็นตัวอักษรซึ่งตัวอักษรที่ตัดออกมาเรียกว่า Parse
4. นำ Parse แต่ละตัวมารวมเป็นคำสั่งซึ่งแต่ละคำสั่งเรียกว่า Token
5. ทำการตรวจสอบหา Operator ต่างๆ เช่น “(“, “)”, “{“, “}”, “;”
6. นำคำสั่งที่ได้จากกระบวนการข้างต้นมารวมกันให้เป็นบรรทัดเดียวเรียกว่า Line



ภาพที่ 16 ภาพแสดงระบบการรับคำสั่งจากผู้เล่น

นำ Line มาประมวลผลผ่านฟังชั่น Runable

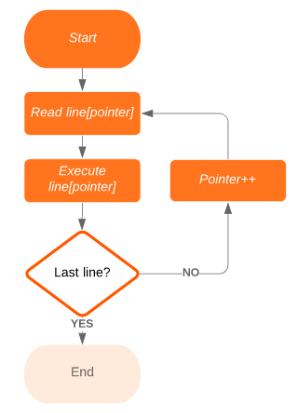
ฟังก์ชั่น Runable มีการทำงานต่อไปนี้

1. แต่ใน Line
3. เมื่อประมวลผลที่ Pointer ชี้ใน Line เสร็จจึงเลื่อน Pointer ชี้บรรทัดถัดไปถัดไป

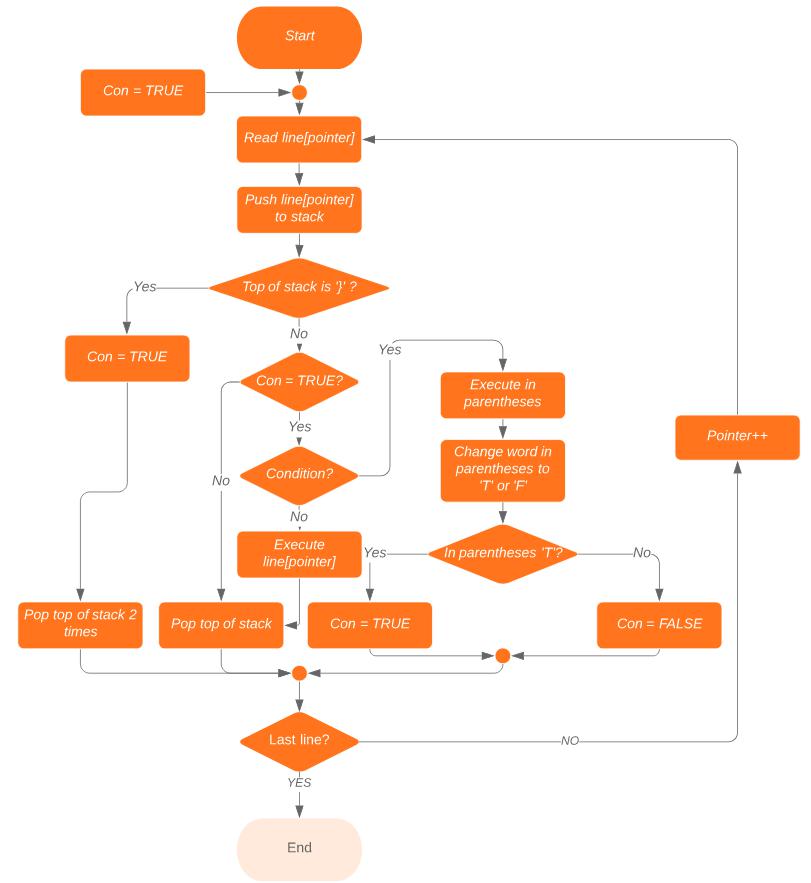
ฟังก์ชั่น Read รับพารามิเตอร์ค่าล่าสุดที่ใส่ใน Stack มาแปลงเป็น Token จากนั้นจึงเข้าสู่ฟังก์ชั่น Check

ฟังกชั่น Check รับพารามิเตอร์ Token มาเช็คว่าแต่ละ Element ของ Token มีความหมายว่าอะไร หากข้อความนั้นมีการแสดงผลจึงนำไปประมวลผลในตัวเกมต่อ

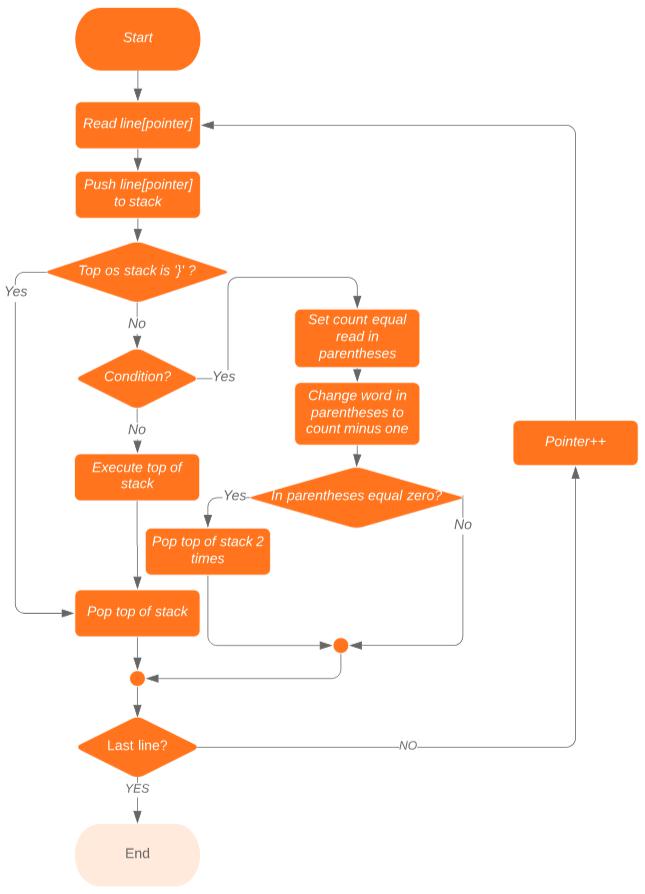
ส่วนของการทำงานของระบบแบบคำสั่งปกติจะมี flowchart ดังภาพที่ 17



ภาพที่ 17 แสดงการทำางานของระบบเมื่อเจอคคำสั่งปกติ

ส่วนของการทำงานของระบบแบบมีเงื่อนไขจะมี flowchart ดังภาพที่ 18

ภาพที่ 18 แสดงการทำงานของระบบเมื่อเจอคำสั่งแบบมีเงื่อนไข

ส่วนของการทำงานของระบบแบบวนลูปจะมี flowchart ดังภาพที่ 19

ภาพที่ 19 แสดงการทำงานของระบบเมื่อเจอคำสั่งแบบลูป

**ส่วนการแสดงผล**

ตัวเกมมีแมพที่เกิดจากการวาดทีละ Element ของ Array 2 มิติ ตัวอย่าง

mapRow1 = [“000000”]

mapRow2 = [“000010”]

mapRow3 = [“000000”]

จากตัวอย่างของโค้ดจะได้แมพที่มีขนาดเท่ากับ 6 x 3 ช่อง หลังจากนั้นตัวเกมจะมีฟังก์ชันอัพเดตซึ่งภายในจะมี for เช็ค map ในแต่ละ row ว่าเป็นเลขอะไร ในกรณีนี้ให้ 0 คือพื้นดินธรรมดาที่ตัวละครสามารถเดินได้ ให้ 1 เป็นเส้นชัย

กำหนดให้เลข 9 แทนตัวละคร เมื่อเริ่มเกมจะทำการเปลี่ยนตำแหน่ง (x, y) ใดๆใน แมพให้เป็นเลข 9 จากนั้นฟังก์ชันอัพเดตจะทำการเปลี่ยนตำแหน่งที่ (x, y) ใดๆ บนหน้าจอห้เป็นรูปตัวละคร โดยการกำหนดจุด 2 จุดบนหน้าจอ จากสูตร

จุดที่ 1 จากสมการ (y \* scale) + locationX + (padX \* x)

จุดที่ 2 จากสมการ (x \* scale) + locationY - (padY \* x) - picY + picX

scale คือ ขนาดของรูปภาพจะใหญ่กี่เท่าจากรูปจริง

locationX คือ ตำแหน่งเริ่มแรกแนวแกน x ที่จะวางรูปภาพ

locationY คือ ตำแหน่งเริ่มแรกแนวแกน y ที่จะวางรูปภาพ

padX คือ ความกว้างของรูปภาพที่จะต้องเผื่อไว้

padY คือ ความยาวของรูปภาพที่จะต้องเผื่อไว้

picX คือ ความกว้างของรูปแบบต้นฉบับ

picY คือ ความยาวของรูปแบบต้นฉบับ

**แผนการดำเนินงาน**

ตาราง 1 แผนการดำเนินงาน

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ลำดับ | กิจกรรม | ปี พ.ศ. 2562 | | | | | ปี พ.ศ. 2563 | | |
| ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. |
| 1. | พบอาจารย์ที่ปรึกษา |  |  |  |  |  |  |  |  |
| 2. | สร้างเกมต้นแบบ |  |  |  |  |  |  |  |  |
| 3. | พบกรรมการ และนำเสนอ |  |  |  |  |  |  |  |  |
| 4. | สร้างฟังค์ชั่น If และ While |  |  |  |  |  |  |  |  |
| 5. | ออกแบบเกมเพิ่มเติม |  |  |  |  |  |  |  |  |
| 6. | ปรับปรุงภาพ |  |  |  |  |  |  |  |  |
| 7. | นำเสนอโครงงาน |  |  |  |  |  |  |  |  |

# เอกสารอ้างอิง

[1]ปณิธิ เนติ นันทน์, ชลธร อริยปิติพันธ์. การศึกษาผลสัมฤทธิ์ โครงการเตรียมความพร้อมทาง ด้านคอมพิวเตอร์ของนักศึกษาใหม่คณะวิทยาศาสตร์และเทคโนโลยีมหาวิทยาลัยกรุงเทพ.

[2]ประสงค์ อุทัย และ คณะ. (2012). การ ศึกษาเปรียบเทียบผลสัมฤทธิ์ ในทางการเรียนและ ความชอบจากการสอนโดยใช้เกมกับการสอนปกติ: ในวิชาพื้นฐานคอมพิวเตอร์และเทคโนโลยี สารสนเทศ. *Veridian E-Journal, Silpakorn University (Humanities, Social Sciences and arts)*, *5*(1), 476-485

[3]อัญชลี อิน ถา, ภาสกร เรืองรอง. (2016). การพัฒนา บทเรียนบนแท็บเล็ตพีซีร่วมกับการ จัดการเรียนรู้โดยใช้เกม เรื่องการเสริมสร้างสุขภาพในวัยเด็กสำหรับนักเรียน ชั้น ประถมศึกษา ปีที่ 2. *JOURNAL OF EDUCATION NARESUAN UNIVERSITY*, *18*(4), 324-333.

[4] ธนสาร รุจิรา, ปานจิตร์ หลงประดิษฐ์, ณัฐชา เดชดำรง. (2015). เกมส่งเสริมการเรียนรู้ แบบผสมผสาน เรื่องการประกอเครื่องคอมพิวเตอร์. *Journal of Information Technology Management and Innovation*, *2*(1), 19-27.

[5] บัญญพนต์ พูลสวัสดิ์, พนมพร ดอกประโคน. (2016). Visual Programming and Computational Thinking Game. *Journal of Information Science and Technology*, *6*(2), 9-16

[6] ราตรี สุภาเฮือง,วชิระ อินทร์อุดม. (2011). การพัฒนาเกมการสอนบนเว็บเรื่องการสื่อสาร ข้อมูลสำหรับนักเรียนชั้นมัธยมศึกษาปีที่ 4 โรงเรียนขอนแก่นวิทยา ย น 2. *Journal of Education Khon Kaen University (Graduate Studies Research)*, *5*(3), 81-86

[7] Dondlinger, M.J., 2007. Educational video game design: A review of the literature. *Journal of applied educational technology*, *4*(1), pp.21-31.