

การเขียนโปรแกรมสำหรับเด็ก เวอร์ชั่น 1.0 26.3.2555

สมชาย พัฒนาชวนชม

สงวนลิขสิทธิ์ พ.ศ. 2555 โดย สมชาย พัฒนาชวนชม

ลิขสิทธิ์หนังสืออยู่ภายใต้เงื่อนไขสัญญาอนุญาต ครีเอทีฟคอมมอนส์ 3.0 (CC-BY-NC-SA)

ให้เผยแพร่ โดยอ้างอิงแหล่งที่มา ห้ามนำไปใช้เพื่อการค้า และให้อนุญาตต่อไปแบบเดียวกัน



This work is licensed under a <u>Creative Commons Attribution-NonCommercial-ShareAlike</u> <u>3.0 Unported License</u>.

บทน้ำ

ทำไมต้องส่งเสริมให้เด็กหัดเขียนโปรแกรม

เป็นที่ทราบกันดีว่า ในปัจจุบันนี้คอมพิวเตอร์มีบทบาทอย่างมากในชีวิตประจำวันของเรา อีกทั้ง เทคโนโลยีสารสนเทศและการสื่อสารยังช่วยส่งเสริมความก้าวหน้าอย่างมาก ในศาสตร์ทุกแขนง และสาขาอาชีพ ส่งผลให้ในศตวรรษที่ 21 มีความต้องการเพิ่มศักยภาพของเด็ก ให้มีความรู้ด้าน คอมพิวเตอร์และเทคโนโลยีสูงขึ้น และไม่ควรเป็นผู้ใช้เพียงอย่างเดียว แต่ต้องสามารถใช้ ความคิดสร้างสรรค์เพื่อสร้างผลงานใหม่ๆ ออกมาได้ด้วย

การฝึกเขียนโปรแกรมเป็นเครื่องมือสำคัญที่ใช้ส่งเสริมศักยภาพในการแก้ปัญหา ด้วย กระบวนการคิดวิเคราะห์ คิดอย่างเป็นระบบ และคิดสร้างสรรค์ ที่ช่วยให้ผู้เรียนแสดงออกทาง ความคิด ผ่านไปยังโปรแกรมผลงานที่สร้างขึ้น แล้วสื่อสารไปยังผู้ใช้โปรแกรมเหล่านั้น

ในนานาประเทศ ตลอดหลายทศวรรษที่ผ่านมามีความพยายามอย่างต่อเนื่อง ที่จะส่งเสริมการ เขียนโปรแกรมให้กับเด็ก ๆ แต่ส่วนใหญ่ยังไม่บรรลุเป้าหมาย เนื่องจากโปรแกรมภาษาที่ใช้ ยังคงซับซ้อนเกินความสามารถของผู้เรียน อย่างไรก็ตามงานวิจัยในเรื่องโปรแกรมภาษาที่ เหมาะสมกับเด็ก หรือผู้ที่ไม่มีประสบการณ์การเขียนโปรแกรมมาก่อน ยังดำเนินมาตลอด และ ในช่วง 4-5 ปีที่ผ่านมา โปรแกรมภาษา Scratch จากทีมงานวิจัย Media Lab MIT ได้รับการ เผยแพร่ และเป็นที่ยอมรับอย่างกว้างขวาง ด้วยความง่ายของโปรแกรมภาษา และเน้นการเขียน โปรแกรมผ่านสื่อหลากหลายชนิด ทำให้เด็ก ๆ สนุกสนานและสนใจกับการเรียนรู้ได้อย่าง ยาวนาน

เด็กได้อะไรจากการเขียนโปรแกรม

กระบวนการเขียนโปรแกรม ส่งเสริมให้เด็กใช้ความคิดวิเคราะห์ คิดอย่างเป็นระบบ คิดเซิง ตรรกะ และคิดสร้างสรรค์ เพื่อแก้ปัญหาทางคอมพิวเตอร์ ด้วยการสร้างโปรแกรมผลงาน ซึ่งเริ่ม จากการพัฒนาทางความคิด ถ่ายทอดความคิดสู่การลงมือปฏิบัติอย่างเป็นขั้นตอน แก้ไขปัญหา เฉพาะหน้าเมื่อมีอุปสรรค และเมื่อทำกระบวนการนี้ซ้ำๆ ก็จะเกิดความซำนาญและความมั่นใจ ในแนวคิดและความสามารถของตัวเอง

การนำเสนอผลงานให้ผู้อื่นได้ทดลองใช้งาน พร้อมเปิดโอกาสรับฟ[ั]งคำวิจารณ์ ข้อสงสัย และ ข้อเสนอแนะ เป็นการเพิ่มทักษะการฟ[ั]งและการยอมรับ การใช้เหตุผลในการอธิบาย และยัง สามารถนำคำแนะนำมาปรับปรุงแก้ไขผลงานตัวเองให้ดียิ่งขึ้นได้

การแลกเปลี่ยนผลงานและความคิดเห็นกับผู้อื่น ทำให้ผู้เรียนได้มุมมอง ประสบการณ์ และ เทคนิคการเขียนโปรแกรมใหม่ ๆ สามารถนำมาต่อยอดเชื่อมโยงกับผลงานตัวเองได้ และยังทำ ให้ผู้เรียนเข้าใจในสิ่งที่ตัวเองได้เรียนรู้มา และสิ่งใดบ้างที่ต้องเรียนรู้เพิ่มเติม

เมื่อผู้เรียนมีพื้นฐานการเขียนโปรแกรมแล้ว การทำงานเป็นทีมเพื่อแก้ปัญหาที่ใหญ่ขึ้น เป็นการ ส่งเสริมทักษะการบริหารและการวางแผนเพื่อแบ่งงานกันทำภายในกลุ่ม และทักษะการสื่อสาร ภายในกลุ่มซึ่งสำคัญมากในการทำงานให้สำเร็จตามเป้าหมาย

การสร้างผลงานยังสามารถเชื่อมโยงการใช้ความรู้ในศาสตร์อื่นๆ ผ่านผลงานที่หลากหลาย เช่น การจำลองทางวิทยาศาสตร์ การเล่าเรื่องแอนิเมชั่น การแสดงผลงานศิลปะและงานดนตรี แม้กระทั่งเกมส่งเสริมการเรียนรู้ เป็นต้น การสร้างผลงานเป็นการเรียนรู้แบบผสมผสานแบบ หนึ่ง ที่จะทำให้เด็กมีความสนใจและความเข้าใจในศาสตร์ที่เชื่อมโยง ในเชิงลึกมากยิ่งขึ้น

ทำไมต้องใช้ Scratch

โปรแกรม Scratch เป็นทั้งเครื่องมือและโปรแกรมภาษา พัฒนาโดยทีมงาน Media Lab MIT มี กลุ่มเป้าหมายเป็นเด็กๆ หรือผู้เริ่มต้นหัดเขียนโปรแกรม วัตถุประสงค์ของทีมพัฒนา คือการ ส่งเสริมผู้เรียนให้ใช้ความคิดสร้างสรรค์ ผ่านการใช้สื่อต่างๆ ไม่ว่าจะเป็นข้อความ ภาพ เสียง หรือ ภาพเคลื่อนไหว ด้วยเครื่องมือที่ไม่ซับซ้อน และมาพร้อมกับตัวโปรแกรม Scratch เพื่อ สร้างโปรเจกต่างๆ ตามจินตนการของผู้เรียน อาทิเช่น การเล่าเรื่อง การสร้างแอนิเมชั่น การ จำลอง แม้กระทั่งการเขียนเกมก็ทำได้

Scratch เหมาะสำหรับผู้เริ่มต้นในการเขียนโปรแกรม เนื่องจากวิธีการเขียนโปรแกรม ทำได้โดย การต่อบล็อกคำสั่งเพื่อสร้างโปรแกรมสคริปต์ คล้ายกับการต่อเลโก้ บล็อกที่ต่อด้วยกันได้เท่านั้น ที่จะอนุญาตให้ต่อกันได้ การใส่ข้อมูลในบล็อกก็มีการตรวจเช็คเพื่อป้องกันการใส่ข้อมูลผิดพลาด ทำให้ไม่เกิดข้อผิดพลาดในการเขียนโปรแกรม การเขียนโปรแกรมจึงเป็นเรื่องง่าย ทำให้ผู้เรียน สนใจเรียนรู้ ออกแบบ และสร้างโปรเจกต่างๆ อย่างสนุกสนาน ด้วยความคิดสร้างสรรค์และ จินตนาการ

การฝึกเขียนโปรแกรมภาษา Scratch ยังใช้แนวคิด หลักการเขียนโปรแกรม และแนวทางปฏิบัติ ในการพัฒนาโปรแกรม เช่นเดียวกับที่ใช้ในโปรแกรมภาษาอื่นๆ จึงเป็นการปูพื้นฐานและเตรียม ความพร้อม ในการเรียนคอมพิวเตอร์ระดับต่อไปที่ซับซ้อนยิ่งขึ้น

ความคิดสร้างสรรค์เป็นกระบวนการทางสังคม ดังนั้นทีมพัฒนา Scratch ได้สร้างเว็บไซต์ http://scratch.mit.edu เป็นเครือค่ายชุมชนผู้สนใจใน Scratch ที่มีผู้ใช้งานอยู่ทั่วโลก เพื่อเป็น แหล่งความรู้ข้อมูลข่าวสาร เป็นที่แลกเปลี่ยนโปรเจกผลงาน แลกเปลี่ยนความคิดเห็น และ มุมมองใหม่ ๆ ในเครือข่ายชุมชน

ถึงแม้ Scratch จะถูกออกแบบมาสำหรับเด็ก แต่ศักยภาพของมันก็สูงพอที่ผู้ใหญ่ โดยเฉพาะครู สามารถเรียนรู้และใช้ Scratch ได้อย่างหลากหลาย โดยเฉพาะเป็นเครื่องมือส่งเสริมการสอน อย่างสร้างสรรค์ได้ เช่น ออกแบบโจทย์ทดสอบที่ไม่น่าเบื่อ สร้างแบบจำลองเพื่อเพิ่มความเข้าใจ หรือทำสื่อการสอนในวิชาต่างๆ เป็นตัน

้ความคาดหวังเมื่อเด็กได้เรียนรู้ Scratch จากหนังสือเล่มนี้

การเขียนโปรแกรมภาษา Scratch สามารถเพิ่มเติมทักษะและความรู้ ให้กับผู้เรียน ด้าน คอมพิวเตอร์ เทคโนโลยีสารสนเทศ และการสื่อสาร เพื่อเตรียมความพร้อมสำหรับการเรียนใน ระดับที่สูงขึ้น ซึ่งผู้เรียนต้องเรียนรู้เชิงลึก ต้องมีกระบวนการคิดอย่างสร้างสรรค์ เป็นผู้สร้าง มากกว่าเป็นผู้ใช้เพียงอย่างเดียว

การเขียนโปรแกรมเป็นกิจกรรม ที่ส่งเสริมการใช้ความคิดวิเคราะห์อย่างเป็นเหตุเป็นผล และคิด อย่างเป็นระบบ เพื่อแก้โจทย์ป ัญหาที่ซับซ้อน ส่งเสริมการใช้ความคิดสร้างสรรค์ และ จินตนาการ ในการออกแบบและการสร้างโปรแกรมผลงาน นอกจากนั้นการนำเสนอผลงานและ การแลกเปลี่ยนความคิดเห็นระหว่างผู้เรียน ยังเป็นการเพิ่มพูนความรู้ ประสบการณ์ และ มุมมองใหม่ๆ รวมทั้งทักษะที่ใช้ในการสื่อสารกับผู้อื่นด้วย

การสร้างโปรแกรมผลงาน เป็นการฝึกผู้เรียนให้เลือกใช้สื่อที่มีอยู่หลากหลายได้อย่างเหมาะสม สามารถเชื่อมโยงความรู้ในศาสตร์อื่นๆ เข้ากับโปรแกรมผลงาน ในลักษณะของสื่อรูปแบบผสม ทำให้เพิ่มมิติและขีดความสามารถในการเรียนการสอนได้เป็นอย่างดี

อ่านหนังสือเล่มนี้อย่างไร

หนังสือแบ่งออกเป็นสามส่วนหลักๆ ส่วนแรกแนะนำตัวโปรแกรม Scratch และการใช้งาน เครื่องมือต่างๆ ส่วนที่สองแนะนำวิธีการเขียนโปรแกรมภาษา Scratch แนวคิดและหลักการ เขียนโปรแกรม ส่วนที่สามเรียนรู้เทคนิคการเขียนโปรแกรมแบบต่างๆ จากโปรเจกที่มาพร้อม กับโปรแกรม Scratch โดยแต่ละโปรเจกที่ถูกเลือกมาในแต่ละหมวด เป็นตัวอย่างเพื่อให้ผู้เรียน ได้ศึกษา และสามารถศึกษาตัวอย่างที่เหลือเพิ่มเติมได้ด้วยตนเอง

การฝึกเขียนโปรแกรมผ่านโปรเจก ผู้เรียนควรศึกษาทำความเข้าใจภาพรวมของโปรเจก เข้าถึง ความสัมพันธ์ระหว่างตัวละครต่างๆกับเวที เรียนรู้การใช้บล็อกคำสั่งเพื่อทำงานให้ได้ตาม เป้าหมาย เมื่อมีความเข้าใจดีแล้ว ผู้เรียนควรต่อยอดโปรเจก หาไอเดียใหม่ๆ ใส่เข้าไปแล้ว เขียนโปรแกรมเพิ่มเติม

การอ่านแบบเรียงลำดับ เหมาะสำหรับผู้ที่ไม่มีประสบการณ์การเขียนโปรแกรมมาก่อนเลย ส่วน คนที่มีประสบการณ์การเขียนโปรแกรมมาบ้าง อาจอ่านเร็ว ๆ ในส่วนแรกเพื่อทำความรู้จักกับ โปรแกรม Scratch ก่อน จากนั้นเรียนรู้เทคนิคต่าง ๆ ในการเขียนโปรแกรมภาษา Scratch ใน ส่วนที่สองและสาม

ผู้เรียนไม่ควรศึกษาบล็อกคำสั่งที่ละคำสั่ง แต่อย่างน้อยควรเข้าใจการแบ่งหมวดของบล็อกต่างๆ ควรเริ่มจากการศึกษาตัวปัญหา พยายามหาแนวทางของคำตอบ ทดสอบแนวทางนั้นด้วยการ เขียนโปรแกรม อาจเลียนแบบโปรแกรมที่ได้ศึกษามาก่อนหน้านี้ก็ได้ เมื่อผู้เรียนมีประสบการณ์ การเขียนโปรแกรมมากขึ้น จะเข้าใจการใช้งานบล็อกต่างๆ มากขึ้นตามไปด้วย อย่างไรก็ตาม เมื่อเกิดความสงสัยในการใช้งานบล็อกต่างๆ ด้านท้ายของหนังสือได้รวบรวมการใช้งานบล็อก คำสั่งต่างๆ ไว้สำหรับอ้างอิง

สารบัญ

| ทนำ | 3 |
|-----------------------------|-----|
| ไรแกรม Scratch | 9 |
| ทูลบาร์และเมนูบาร์ | 11 |
| ตัวละครและเวที | 21 |
| พื้นที่สำหรับเขียนสคริปต์ | 35 |
| ชุดตัวละครและภาพฉากหลังเวที | 38 |
| เสียงตัวละครและเสียงประกอบ | 41 |
| บล็อกคำสั่ง | 44 |
| ารเขียนโปรแกรมใน Scratch | 55 |
| การสร้างสคริปต์ | 56 |
| หลักการในการเขียนโปรแกรม | 65 |
| ปรเจกใน Scratch | 79 |
| โปรเจก Aquarium | 81 |
| โปรเจก MadLibs | 92 |
| โปรเจก SimpleCircuit | 106 |
| โปรเจก GardenSecret | 124 |
| โปรเจก PianoMachine | 140 |
| โปรเจก FishChomp | 157 |
| ารใช้งานบล็อกคำสั่ง | 169 |
| หล่งอ้างอิง | 185 |

โปรแกรม Scratch

หน้าตาของโปรแกรม Scratch

ก่อนอื่นมาทำความรู้จักหน้าตาโปรแกรม Scratch กันก่อน โปรแกรม Scratch นั้นเต็มไปด้วย สีสันสวยงาม โดยมีเวทีสำหรับการแสดงจากตัวละครต่างๆ ซึ่งแต่ละตัวละครรวมทั้งเวทีเอง ต่าง ก็มีบทบาทหน้าที่ของตัวเองที่ถูกเขียนไว้ในสคริปต์ สคริปต์สร้างได้ด้วยการเขียนโปรแกรม ที่ สร้างจากบล็อกคำสั่งในหมวดต่างๆ นำมาประกอบเรียงกันเข้าเป็นกลุ่มๆ



องค์ประกอบหลักๆ ในโปรแกรม Scratch จากรูปสามารถแบ่งออกเป็นสัดส่วนได้ดังนี้

- 1. ทูลบาร์และเมนูบาร์
- 2. เวทีสำหรับตัวละคร
- 3. รายการตัวละครและเวที
- 4. ข้อมูลของเวทีหรือตัวละครที่ถูกเลือก
- พื้นที่สำหรับเขียนโปรแกรมสคริปต์ ชุดตัวละคร (ฉากหลังเวที) และเสียงประกอบ ของ เวทีหรือตัวละครที่ถูกเลือก
- 6. บล็อกคำสั่งหมวดต่างๆ ซึ่งแยกตามสีแตกต่างกัน
- 7. บล็อกคำสั่งต่างๆ ในหมวดที่ถูกเลือก

ທູຄນາຮໍ່ແລະເນ**ນູ**ນາຮໍ່

เนื่องจากความต้องการให้ Scratch ใช้งานง่าย ไม่ซับซ้อน ทูลบาร์จึงมีได้หลายแห่งตามความ เหมาะสม รวมทั้งอยู่รวมกับเมนูบาร์ด้วย

ทูลบาร์

ทูลบาร์ประกอบไปด้วย ปุ่มต่างๆ สำหรับงานที่ใช้บ่อยๆ ทำให้สะดวกในการใช้งาน เพียงคลิกที่ ปุ่มเมื่อต้องการทำงานนั้นๆ



การบันทึกโปรเจกป[ั]จจุบัน ทำได้อีกวิธี คือเลือกเมนู File คลิกเลือก Save

การแชร์โปรเจกไปยังเว็บไซต์ Scratch ทำได้อีกวิธี คือเลือกเมนู Share คลิกเลือก Share This Project Online…

การเปลี่ยนภาษา



สำหรับคนที่ไม่ถนัดภาษาอังกฤษ Scratch รองรับภาษาอื่นๆ ได้หลายภาษา รวมทั้งภาษาไทย ด้วย วิธีการเปลี่ยนภาษาทำได้โดย คลิกปุ่มลูกโลก จะเห็นรายการภาษาต่างๆ ให้เลือก

ถ้าไม่พบภาษาไทย หรือภาษาอื่นที่ต้องการ ให้คลิก more... เพื่อดูภาษาอื่นๆ เพิ่มเติม คลิก เลือก "ไทย" เพื่อเลือกภาษาไทย

หน้าตาโปรแกรมจะเปลี่ยนเป็นภาษาไทยเกือบทั้งหมด ยกเว้นชื่อและข้อความที่เขียนเป็น ภาษาอังกฤษตั้งแต่แรก



การใช้เครื่องมือ

คลิกเลือกเครื่องมือที่ต้องการ ตัวชี้เมาส์จะเปลี่ยนเป็นรูปของเครื่องมือนั้น จากนั้นสามารถคลิก บนตัวละคร ชุดแต่งกาย เสียง บล็อกคำสั่ง หรือสคริปต์ ที่รองรับเครื่องมือนั้นๆ



ถ้าต้องการยกเลิกการใช้เครื่องมือ ให้คลิกพื้นที่อื่นๆ ของหน้าจอคอมพิวเตอร์ ที่ไม่ใช่ ตัวละคร ชุดแต่งกาย เสียง บล็อกคำสั่ง หรือสคริปต์

15

โหมดของเวที

ปุ่มปรับเปลี่ยนโหมดของเวที อยู่มุมบนขวาของโปรแกรม สามารถเลือกเปลี่ยนให้เหมาะสมกับ งานที่กำลังทำอยู่



โหมดการนำเสนอ (Presentation mode) คลิกปุ่มนี้เมื่อต้องการนำเสนอโปรเจกเต็มหน้า จอคอมพิวเตอร์ ถ้าต้องการออกจากโหมดการนำเสนอ กดคีย์ Esc



โหมดการดู (View mode) เหมาะสำหรับตอนออกแบบและเขียนโปรแกรมสคริปต์ พื้นที่สีขาว แสดงขนาดของเวที คลิกปุ่มระหว่างสองปุ่มนี้ เพื่อสลับระหว่างเวทีเล็กกับเวทีใหญ่ คลิกปุ่มแรก เลือกเวทีเล็ก เพื่อให้ได้พื้นที่ในการเขียนสคริปต์มากขึ้น คลิกปุ่มที่สองเลือกเวทีใหญ่ เพื่อดูการ ทำงานบนเวทีได้ชัดเจนยิ่งขึ้น ปุ่มที่ถูกเลือกปจจจุบันจะมีกรอบสีเทาเข้มล้อมอยู่ (ปุ่มเวทีเล็ก)

เมนูบาร์

เมนูบาร์แบ่งการทำงานต่างๆ ของ Scratch เป็นหมวดหมู่ อยู่ร่วมกับปุ่มเครื่องมือบางส่วน

SCRATCH 🕀 🗟 🏠 File Edit Share Help

8 4 53 XK

เมนู File

| File Edit Share | Help |
|-----------------|------|
| New | |
| Open | |
| Save | |
| Save As | |
| Import Project | |
| Export Sprite | |
| Project Notes | |
| Quit | |

ในเมนู File มีเมนูย่อย ใช้จัดการกับโปรเจก Scratch ได้หลายอย่าง โดยคลิกเลือก

- New สร้างโปรเจกใหม่
- Open... เปิดโปรเจกเก่า
- Save บันทึกโปรเจกปจจุบัน
- Save As... บันทึกโปรเจกปจจจบันในชื่อใหม่
- Import Project... นำเข้าตัวละครและฉากเวทีจากโปรเจกอื่นๆ มารวมกับโปรเจก ปัจจุบัน
- Export Sprite... ส่งออกไฟล์ของตัวละครที่ถูกเลือกอยู่ เพื่อใช้ในโปรเจกอื่น
- Project Notes... เขียนอธิบายเกี่ยวกับโปรเจก เช่นคำสั่งในการใช้งานโปรเจก
- Quit ออกจากโปรแกรม Scratch

เมนู Edit

File Edit Share Help Undelete Start Single Stepping Set Single Stepping... Compress Sounds... Compress Images... Show Motor Blocks

ในเมนู Edit มีเมนูย่อยดังนี้

- Undelete กู้ บล็อก สคริปต์ ตัวละคร ชุดละคร หรือเสียง กลับมาจากที่ถูกลบออกไป
- Start Single Stepping แสดงการทำงานทีละขั้น แต่ละบล็อกที่กำลังทำงานจะถูกไฮไลท์ มีประโยชน์ในการหาข้อผิดพลาดของโปรแกรม และช่วยให้นักเขียนโปรแกรมมือใหม่ เข้าใจลำดับขั้นตอนการทำงานของโปรแกรม
- Set Single Stepping... ใช้กำหนดความเร็วการทำงานใน 1 ขั้นตอนของโปรแกรม
- Compress Sounds... บีบอัดข้อมูลเสียงเพื่อลดขนาดของไฟล์โปรเจก แต่การบีบอัด อาจลดคุณภาพของเสียง
- Compress Images... บีบอัดข้อมูลภาพเพื่อลดขนาดของไฟล์โปรเจก แต่การบีบอัดอาจ ลดคุณภาพของภาพ
- Show Motor Blocks เพิ่มบล็อกมอเตอร์ในบล็อกหมวดการเคลื่อนที่ สามารถใช้บล็อก มอเตอร์ร่วมกับ LEGO® Education WeDo ได้ เพื่อควบคุมการทำงานมอเตอร์ ที่ เชื่อมต่อกับคอมพิวเตอร์

เมนู Share

| File | Edit | Share | Help |
|------|------|--------------------|--|
| | | Share 1 Go To 9 | This Project Online Scratch Website |

ในเมนู Share มีเมนูย่อยดังนี้

- Share This Project Online... สำหรับแชร์โปรเจกปจจจุบันไปยังเว็บไซต์ของ Scratch
- Go To Scratch Website... ไปยังเว็บไซต์ของ Scratch (scratch.mit.edu)

เว็บไซต์ Scratch มีข้อมูลที่น่าสนใจหลายๆ อย่างเกี่ยวกับ Scratch สามารถดาวน์โหลด โปรแกรม Scratch ได้ที่นี้ รวมทั้งโปรเจกที่ผู้อื่นแชร์ไว้มากมาย

| SCRAT | home pr | ojects galleries suppo | ort forums about | Language ^a |
|------------------|--------------------------|------------------------|-----------------------|-----------------------|
| imagine • progra | m • share | Signup for an account | | search |
| | | | | |
| Create and | share your ov | vn interactive | | |
| stories da | mes music ar | nd art | | _ |
| stones, ga | mes, music, ai | | set this effect to | a 1 |
| Chee | ck out the 2,335,641 pro | jects | change x by pict | Im 10 to 60 |
| from | around the world! | | change Will effect | by (90) |
| To c | reate your own projects | : | wait 0.2 secs | |
| | | | set while effect to I | 0 |
| 86 6 | Download Scratch | | | |
| | | | | |
| Featured Projec | ts | See mo | Collab Cam | P |
| 0.0 | Met contrib | | n | Collaborate |
| a s anthing | | ANPL | COLLAB | with other |
| + + + + h = - | | | CAMP | Scratchers at |
| A 1 | | | | Collab Camp to |
| the an | | | create musi | c mashups. |
| Map-It | Launch your | Valentine's Day | | Learn more |
| by | own | by JoanofArc | - | |
| UnprovenTheore | by <u>sennajnuet</u> | | Scratch Day | y |

19

การแชร์โปรเจก

ถ้าต้องการแชร์โปรเจกป[ั]จจุบันที่กำลังสร้างอยู่ไปยังเว็บไซต์ของ Scratch คลิกเมนู Share เลือก Share This Project Online...

| and the second sec | Your Scratch website login na | mei | | |
|--|--|---------------|--|--|
| | kidsangsan | | | |
| Came Over | Password: | Create accoun | | |
| Gaine Over | ****** | _ | | |
| New Martha Martha | Project name: | | | |
| | Delicious Fish | | | |
| Tags: | Project notes: | | | |
| Animation Music Art Simulation Game Story Hore tags: | INSTRUCTIONS Level 1 (20 seconds): Click the Green Flag to start. Move the mouse to control the big fish. Get 1 point for eating a goldfish. Level 2 (30 seconds): Eating Goldfish1 gets 2 point bonus, but eating the small green fish loses 3 points. HOW I MADE THIS * This project extends the FishChomp project * Add scores, time, levels, more fishes, | | | |

การแชร์โปรเจก ต้องมี Login และ Password ถ้ายังไม่มี สมัครได้โดยคลิก Create account ซึ่ง จะเปิดเว็บไซต์ Scratch ออกมา คลิก Sign up เพื่อสมัครสมาชิก จากนั้นใส่ข้อมูลต่างๆ ที่ เกี่ยวกับโปรเจก เช่น ชื่อโปรเจก (Project name) รายละเอียดของโปรเจก (Project notes) และ สามารถคลิกกล่องหน้า Compress sounds and images เพื่อบีบอัดข้อมูลไฟล์เสียงและไฟล์ รูปภาพให้มีขนาดเล็กลง

เมนู Help

| File | Edit | Share | Help |
|------|------|-------|--|
| | | | Help Page Help Screens About Scratch |

เมนู Help ใช้สำหรับขอความช่วยเหลือในการใช้งาน Scratch มีเมนูย่อยดังนี้

- Help page... นำไปสู่เว็บไซต์ของ Scratch ที่มีคู่มือ เอกสารอ้างอิง บทเรียน และ คำถามคำตอบที่พบบ่อยๆ
- Help screens... นำไปสู่เว็บไซต์ของ Scratch ที่อธิบายการใช้งานบล็อกต่างๆ
- About Scratch... ให้ข้อมูลเกี่ยวกับ Scratch

เมนูอื่น ๆ

ถ้าคลิกขวาที่ตัวละคร เวที บล็อก หรือพื้นที่ว่างบางแห่ง จะมีเมนูปรากฏขึ้นมา ให้เลือกใช้งานที่ เกี่ยวข้องกับสิ่งที่คลิกนั้นๆ

ตัวละครและเวที

เวที

เวที (Stage) เป็นสถานที่ที่ให้ตัวละครใช้แสดง มีภาพฉากหลัง (Background) ที่สามารถเปลี่ยน ได้ มีขอบเขตที่แน่นอน โดยแต่ละตำแหน่งบนเวที ถูกกำหนดด้วยค่า x ในแนวนอน และ y ใน แนวตั้ง ที่มุมล่างขวามีตำแหน่งพิกัดของเมาส์แสดงอยู่



ด้านบนซ้ายของเวทีแสดงชื่อของโปรเจก (3 FishChomp) ด้านบนขวามีปุ่มธงเขียว 🛤 ใช้ สำหรับเริ่มต้นทำงานสคริปต์ในโปรเจก ส่วนปุ่มแปดเหลี่ยมสีแดง ● ใช้สำหรับหยุดการ ทำงานสคริปต์ทั้งหมด



เมื่อคลิกขวาบนฉากเวที จะมีเมนูปรากฏให้เลือกดังนี้

- grab screen region for new sprite ตีกรอบพื้นที่ฉากหลังแล้วนำมาใช้เป็นตัวละคร ใหม่
- save picture of stage... บันทึกภาพทั้งหมดที่ปรากฏอยู่บนเวที ณ ขณะนั้น แล้วเก็บ ลงในไฟล์ภาพ

ขนาดของเวทีและตำแหน่งต่าง ๆ บนเวที



เวทีมีขนาด 480 หน่วยในแนวนอน (แกน x) และ 360 หน่วยในแนวตั้ง (แกน y) การบอก ตำแหน่งใด ๆ บนเวทีจะอยู่ในรูปพิกัด (x, y) โดยค่า x และ y นั้น จะอ้างอิงจากตำแหน่ง ศูนย์กลาง (0, 0) ซึ่งอยู่ตรงกลางเวที ดังนั้นมุมบนล่างทั้งซ้ายขวาจะมีตำแหน่งตามที่แสดงไว้ใน รูป

ภาพฉากหลังที่จะแสดงบนเวทีได้ จะมีขนาดไม่เกินกว่าขนาดของเวที (480 x 360) ถ้าภาพฉาก หลังที่จะใช้มีขนาดใหญ่กว่า โปรแกรม Scratch จะลดขนาดของภาพนั้นตามสัดส่วนโดย อัตโนมัติ เพื่อให้พอดีกับขนาดของเวที

รายการตัวละครและเวที



ตัวละคร (Sprite) คือสิ่งที่สร้างขึ้นมาเพื่อใช้แสดงบทบาทหน้าที่บนเวที ตามวัตถุประสงค์ที่ได้ ออกแบบไว้ ส่วนเวที มีฉากหลังแสดงภาพบรรยากาศ และสิ่งแวดล้อมในสถานที่หนึ่งๆ

ตัวละครและเวทีที่ใช้ในแต่ละโปรเจก จะปรากฏอยู่ในรายการใต้เวที เปรียบเสมือนห้องแต่งตัว ของตัวละครกับห้องเก็บฉากเวที โดยมีเส้นแบ่งกั้นอยู่ ห้องเก็บฉากเวทีจะอยู่ซ้ายมือ ส่วนห้อง แต่งตัวจะอยู่ด้านขวามือ แต่ละตัวละครรวมทั้งเวที จะมีชื่อกำกับอยู่ใต้ภาพ ตัวละครหรือฉาก เวทีที่ถูกเลือกจะมีกรอบสีฟ้าล้อมรอบอยู่

เมื่อคลิกขวาที่ตัวละครในรายการ จะมีเมนูปรากฏให้เลือกดังนี้

- show เพื่อให้ตัวละครปรากฏตัวบนเวที
- export this sprite ส่งออกเป็นไฟล์เพื่อเก็บข้อมูล ชุด และสคริปต์ของตัวละครนี้
- duplicate ทำซ้ำตัวละครนี้
- delete ลบตัวละครนี้



เมื่อคลิกขวาบนตัวละครที่อยู่บนเวที จะมีเมนูปรากฏให้เลือกดังนี้

- grab screen region for new costume ตีกรอบพื้นที่ฉากหลังแล้วนำมาใช้เป็นชุดตัว ละครใหม่
- export this sprite ส่งออกเป็นไฟล์เพื่อเก็บข้อมูล ชุด และสคริปต์ของตัวละครนี้
- duplicate ทำซ้ำตัวละครนี้
- delete ลบตัวละครนี้
- resize this sprite ปรับขนาดของตัวละครนี้
- rotate this sprite หมุนตัวละครนี้

การสร้างตัวละครใหม่

ในการสร้างตัวละครใหม่ ๆ เพิ่มเข้ามาในโปรเจกสามารถทำได้หลายวิธี โดยการเลือกคลิกปุ่ม เครื่องมือ 3 ปุ่มนี้





วิธีที่ง่ายที่สุดในการเพิ่มตัวละครใหม่ในโปรเจก คือคลิกที่ปุ่มนี้ โปรแกรม Scratch จะสุ่มตัว ละคร จากคลังตัวละครที่มีอยู่ในโปรแกรมมาให้ วิธีนี้ได้ตัวละครมาอย่างรวดเร็ว แต่ไม่รู้ว่าจะได้ ตัวละครใดมา



แต่ถ้าต้องการเลือกตัวละครเอง ทำได้โดยคลิกปุ่มนี้ โปรแกรมจะแสดงคลังตัวละครในหมวด ต่างๆ มาให้เลือก เช่นในกรณีนี้ ผู้ใช้เลือกโฟล์เดอร์หมวดสัตว์ (Animals) โดยดับเบิ้ลคลิกที่โฟล์ เดอร์หรือคลิกเลือกโฟล์เดอร์แล้วคลิกปุ่ม OK

| New Sprite | | | | |
|-----------------------------|----------|----------------|---------|-----------|
| Computer | Costumes | 2 £ 💽 | | |
| kant Desktop Čostumes | Animals | Fantasy | Letters | People |
| | Things | Transportation | | |
| | <u></u> | | | OK Cancel |

ในโฟล์เดอร์หมวดสัตว์ มีตัวละครให้เลือกมากมาย ถ้าต้องการเลือกตัวละคร เช่น ปู (crab2) ให้ ดับเบิ้ลคลิกได้เลย หรือคลิกเลือกที่ปูแล้วคลิกปุ่ม OK



สังเกตว่า ตัวละครบางตัวจะมีสคริปต์มาด้วย เช่น มด (Crawling Ant) มี 4 สคริปต์ ถ้าเลือกมด ก็จะได้สคริปต์มาด้วย



วิธีสุดท้ายถ้าต้องการสร้างตัวละครด้วยตัวเอง Scratch ก็มีโปรแกรมสำหรับวาดรูปมาให้ด้วย โดยการคลิกปุ่ม จะได้โปรแกรมวาดรูป (Paint Editor)

เมื่อวาดรูปเสร็จแล้ว คลิกปุ่ม OK เพื่อบันทึกรูปเป็นตัวละครใหม่ ถ้าไม่ต้องการรูปที่วาดมา คลิก ปุ่ม cancel



สิ่งที่สำคัญของการสร้างตัวละครด้วยวิธีนี้คือ การกำหนดจุดศูนย์กลางของชุดตัวละคร ซึ่ง นอกจากใช้บอกตำแหน่งพิกัดของตัวละครแล้ว ยังเป็นจุดอ้างอิงสำหรับการเคลื่อนที่ การหมุน และการสลับทิศ

เครื่องมือวาดรูปแถวที่ 1



- พู่กัน : สำหรับวาดอย่างอิสระด้วยสีพื้นหน้าปจจุบัน บริเวณเครื่องมือทางเลือกสามารถปรับ เลือกขนาดของพู่กันได้
- ยางลบ : สำหรับลบอย่างอิสระ บริเวณที่ถูกลบจะโปร่งแสง บริเวณเครื่องมือทางเลือก สามารถปรับเลือกขนาดของยางลบได้
- ถังเทสี : สำหรับเทสีบนพื้นที่ที่ต่อเนื่องกัน ด้วยสีเดียวหรือเฉดสี บริเวณเครื่องมือทางเลือก สามารถเลือกรูปแบบการเทสีได้ (สีเดียว เฉดสีแนวนอน เฉดสีแนวตั้ง หรือ เฉดสีแนวรัศมี) เฉดสีเกิดจากการผสมสีจากสีพื้นหน้าไปยังสีพื้นหลัง
- สี่เหลี่ยม : วาดรูปสี่เหลี่ยมทึบหรือกรอบสี่เหลี่ยม ด้วยสีพื้นหน้า บริเวณเครื่องมือทางเลือก สามารถเลือกรูปแบบของสี่เหลี่ยมได้ (สี่เหลี่ยมทึบหรือกรอบสี่เหลี่ยม) ในกรณีของกรอบ สี่เหลี่ยม ขนาดของเส้นกำหนดได้จากขนาดของพู่กัน
- วงรี: วาดรูปวงรีทึบหรือกรอบวงรี ด้วยสีพื้นหน้า บริเวณเครื่องมือทางเลือกสามารถเลือก รูปแบบของวงรีได้ (วงรีทึบหรือกรอบวงรี) ในกรณีของกรอบวงรี ขนาดของเส้นกำหนดได้ จากขนาดของพู่กัน

เครื่องมือวาดรูปแถวที่ 2



- เส้นตรง : วาดเส้นตรงด้วยสีพื้นหน้า บริเวณเครื่องมือทางเลือก สามารถปรับเลือกขนาด ของพู่กันได้ ถ้ากด Shift + ลากเส้น จะบังคับเส้นเป็นเส้นแนวนอนหรือแนวตั้ง
- ข้อความ : ใส่ข้อความเข้ากับรูปวาด บริเวณเครื่องมือทางเลือกสามารถปรับเลือกแบบอักษร และขนาดของตัวอักษรได้ แต่ละชุดตัวละครสามารถใช้ข้อความได้เพียงหนึ่งข้อความ
- เครื่องมือใช้เลือกพื้นที่ : ตีกรอบพื้นที่สี่เหลี่ยม จากนั้นสามารถลากย้ายสี่เหลี่ยมนี้ไปยัง ตำแหน่งใหม่ได้ (กดคีย์ Del เพื่อลบสิ่งที่เลือกในสี่เหลี่ยม กดคีย์ Shift + Del หรือ Shift + backspace เพื่อลบสิ่งอื่นทั้งหมดที่อยู่นอกกรอบสี่เหลี่ยม) คลิกนอกกรอบสี่เหลี่ยมเพื่อ ยกเลิกกรอบสี่เหลี่ยมปัจจุบัน
- ตราประทับ : เลือกพื้นที่สี่เหลี่ยมครอบคลุมในสิ่งที่ต้องการทำซ้ำ จากนั้นคลิกในบริเวณอื่น
 เพื่อทำซ้ำ ยกเลิกการทำซ้ำด้วยการคลิกเครื่องมือใด ๆ ก็ได้
- หลอดหยด : ใช้ปลายหลอดหยดเพื่อเลือกสีสำหรับสีพื้นหน้า โดยสามารถเลือกได้จากจานสี หรือบริเวณภายในผืนผ้าใบ ถ้าต้องการเลือกสีที่อยู่นอกบริเวณผืนผ้าใบ ให้คลิกหลอดหยด ในผืนผ้าใบ แล้วลากออกไปข้างนอกผ้าใบเพื่อเลือกสีที่ต้องการ

31

เครื่องมืออื่น ๆ

- คลิกบริเวณสีปจจุบันแต่ละครั้ง จะสลับสีระหว่างสีพื้นหน้ากับสีพื้นหลัง คลิกสีในจานสีเพื่อ เลือกสีใหม่สำหรับสีพื้นหน้า ถ้าต้องการเลือกสีสำหรับสีพื้นหลัง กดคีย์ Shift + คลิกสีในจาน สี
- คลิกเลือกจานสีทางเลือก เพื่อเปลี่ยนจานสีเป็นจานสีแบบต่อเนื่อง
- คลิกปุ่มกำหนดจุดศูนย์กลาง เพื่อเลือกตำแหน่งในรูป ให้เป็นจุดศูนย์กลางของการหมุนเมื่อ ตัวละครถูกหมุนบนเวที
- คลิกปุ่มซูมภาพ เพื่อขยายหรือย่อผืนผ้าใบ ถ้าซูมภาพเกิน 100% แถบเลื่อนภาพใน แนวนอนและแนวตั้งจะปรากฏ เพื่อใช้สำหรับเลื่อนดูภาพในผืนผ้าใบ การซูมภาพไม่ได้ เปลี่ยนขนาดของภาพ
- คลิกปุ่มมาตราส่วน (หดหรือขยาย) เพื่อเปลี่ยนขนาดรูปในผืนผ้าใบ หรือรูปในส่วนที่เลือกไว้ ถ้ากดคีย์ Shift + คลิกปุ่มมาตราส่วน จะสามารถใส่ค่าขยายเป็นเปอร์เซ็นต์ได้ (100 % คือ มาตราส่วนดั้งเดิม)
- คลิกปุ่มหมุน (ตามเข็มหรือทวนเข็มนาพิกา) เพื่อหมุนรูปในผืนผ้าใบ หรือรูปในส่วนที่เลือก
 ไว้ ถ้ากดคีย์ Shift + คลิกปุ่มหมุน จะสามารถใส่ค่าองศาได้
- คลิกปุ่มสลับด้าน (แนวนอนหรือแนวตั้ง) เพื่อสลับด้านของรูปในผืนผ้าใบ หรือรูปในส่วนที่ เลือกไว้
- คลิกปุ่มนำเข้า เพื่อเปิดรูปภาพจากไฟล์ และใส่ไว้บนผืนผ้าใบ
- คลิกปุ่มล้างผ้าใบ เพื่อล้างรูปทั้งหมดจากผืนผ้าใบ
- ถ้าเกิดข้อผิดพลาดระหว่างวาดรูป คลิกปุ่มยกเลิก เพื่อยกเลิกสิ่งที่เพิ่งทำไป แต่ถ้าเปลี่ยนใจ ต้องการนำสิ่งที่เพิ่งยกเลิกไปกลับมา คลิกปุ่มทำซ้ำ

ข้อมูลของตัวละคร

เมื่อคลิกเลือกตัวละครในรายการ หรือดับเบิ้ลคลิกตัวละครบนเวที จะเห็นข้อมูลส่วนตัวของตัว ละครนั้น



ตัวละครมีส่วนประกอบอยู่หลายอย่าง เช่น ชื่อ (Creature6) ทิศทางการเคลื่อนที่ (104) ตำแหน่งบนเวที (x: -166 y: 125) สีของปากกา (สีฟ้า) สคริปต์ (Scripts) ชุดตัวละคร (Costumes) เสียง (Sounds) รูปแบบในการหมุน และความสามารถในการเคลื่อนที่ด้วยการใช้ เมาส์ลากในโหมดนำเสนอหรือบนเว็บไซต์ Scratch แม่กุญแจปิดหมายถึงไม่สามารถทำได้ คลิก แม่กุญแจเพื่อปลดล็อก ให้สามารถทำได้

ส่วนประกอบบางอย่างอาจมีหรือไม่มีก็ได้ เช่น สคริปต์และเสียง แต่องค์ประกอบที่ต้องมีคือ ชื่อ ทิศทางการเคลื่อนที่ ตำแหน่งบนเวที ซึ่งจะถูกกำหนดโดยอัตโนมัติถ้าไม่ระบุตอนสร้างตัวละคร ส่วนชุดตัวละคร ซึ่งบ่งบอกหน้าตา และการมีอยู่ของตัวละครนั้นๆ ต้องมีในตอนสร้างตัวละคร

ทิศทางการเคลื่อนที่



ทิศทางบนเวทีจะถูกระบุเป็นองศา ระหว่าง -180 ถึง 180 องศา ทิศ 0 ชี้ไปด้านบน หมุนตาม เข็นนาพิกาเป็นทิศบวก หมุนทวนเข็นนาพิกาเป็นทิศลบ ทิศ -180 กับ 180 องศาชี้ไปด้านล่าง เหมือนกัน แต่มาจากคนละด้าน (ตามรูป)

เมื่อนำรูปปลามาใส่กรอบทิศทางจะได้ดังรูป เส้นสีน้ำเงินเป็นตัวบอกทิศทางการเคลื่อนที่ป[ั]จจุบัน คือ 104 องศา การเปลี่ยนทิศทางทำได้โดยคลิกที่ตัวปลาแล้วหมุนตามทิศที่เราต้องการ สังเกต ว่าเส้นบอกทิศทางสีน้ำเงินและตัวเลขบอกทิศองศาจะเปลี่ยนตามขณะหมุนด้วย

ข้อมูลของเวที

34



เมื่อคลิกเลือกเวทีในรายการ หรือดับเบิ้ลคลิกพื้นที่ว่างบนเวที จะเห็นข้อมูลส่วนตัวของเวที

เวทีมีส่วนประกอบน้อยกว่าตัวละครมาก เนื่องจากเวทีเคลื่อนที่ไม่ได้ จึงไม่จำเป็นต้องมีตำแหน่ง ทิศทางการเคลื่อนที่ รูปแบบในการหมุน และความสามารถในการเคลื่อนที่ด้วยการใช้เมาส์ลาก เวทีวาดรูปบนตัวเองไม่ได้ จึงไม่มีปากกาสำหรับวาดรูป แต่สามารถลบรูปบนเวทีได้ ในแต่ละโป รเจกมีเวทีเดียว จึงมีชื่อเดียว (Stage) และไม่สามารถเปลี่ยนชื่อได้

อย่างไรก็ตามเวทีสามารถมีสคริปต์เป็นของตัวเอง มีภาพฉากหลังได้หลายภาพ และมีเสียง ประกอบเวทีได้

พื้นที่สำหรับเขียนสคริปต์

สคริปต์สำหรับตัวละคร

เมื่อเลือกตัวละครแล้ว (เช่น hungry fish) คลิกแทป Scripts จะเห็นพื้นที่สำหรับเขียนสคริปต์



สคริปต์คือชุดคำสั่งสำหรับตัวละครหรือเวที เพื่อให้ทำงานตามที่ออกแบบโปรแกรมไว้

สคริปต์ของตัวละครหนึ่งๆ มีได้หลายสคริปต์ (2 สคริปต์ในตัวอย่างนี้) หรือไม่มีเหลือก็ได้ สคริปต์หนึ่งๆ ประกอบไปด้วยบล็อกคำสั่งมาเรียงต่อกันเป็นกลุ่ม บางบล็อกสามารถอยู่ข้างใน หรือซ้อนอยู่บนบล็อกอื่นได้ 36

เมื่อคลิกขวาที่บล็อกใดๆ บนสคริปต์ จะมีเมนูปรากฏดังนี้

- help เพื่อข้อความช่วยเหลือ ดูว่าบล็อกนั้นทำงานอย่างไร
- duplicate ทำซ้ำบล็อก โดยบล็อกที่จะถูกทำซ้ำ จะมาตั้งแต่บล็อกที่ถูกคลิกรวมไปถึง บล็อกที่ตามมาด้วย

แต่ถ้าคลิกขวาบล็อกบนสุดของสคริปต์ ซึ่งเป็นตัวแทนของทั้งสคริปต์ จะมีเมนูเพิ่มอีกอย่างคือ delete คลิกเพื่อลบบล็อกทั้งสคริปต์
สคริปต์สำหรับเวที

สคริปต์สำหรับเวทีก็คล้ายๆ กับของตัวละคร เพียงแต่ว่าบล็อกที่สามารถใช้งานได้จะลดลง โดยเฉพาะบล็อกหมวดการเคลื่อนที่นั้นไม่มีเลย เนื่องจากเวทีไม่สามารถเคลื่อนที่ได้นั้นเอง



เมื่อคลิกขวาบนพื้นที่ว่าง จะมีเมนูให้เลือกดังนี้

- clean up เพื่อจัดระเบียบการใช้พื้นที่ของแต่ละสคริปต์ให้เป็นระเบียบเรียบร้อย
- save picture of script เพื่อบันทึกสุคริปต์ทั้งหมดให้เป็นไฟล์ภาพ
- add comment จะปรากฏพื้นที่สีเหลือง สามารถบันทึกข้อความเกี่ยวกับสคริปต์ที่กำลัง เขียนอยู่ เพื่อใช้เตือนความจำในภายหลัง



ชุดตัวละครและภาพฉากหลังเวที



คลิกแทป Costumes ของ hungry fish เพื่อดู แก้ไข หรือเพิ่มเติมชุดของตัวละคร ซึ่งในตัวอย่าง นี้มี 2 ชุด ชุดที่ใส่อยู่ป[ั]จจุบัน จะมีกรอบสีฟ้าล้อมอยู่ สามารถใช้เมาส์คลิกชุดอื่น เพื่อเปลี่ยนชุด ให้ตัวละครบนเวทีได้ ชุดแต่ละชุดเกิดจากรูปภาพ ซึ่งมีส่วนประกอบดังนี้ (สำหรับชุดป[ั]จจุบัน)

- ชื่อชุด (open-mouth)
- ขนาดของภาพ (214 x 111)
- ขนาดไฟล์ของภาพ (2 KB)
- ลำดับชุดกำกับอยู่ที่มุมซ้ายบน (1) ซึ่งสามารถสับเปลี่ยนได้ด้วยการลากชุดขึ้นลง ไป ตำแหน่งที่ต้องการ
- ปุ่ม Edit เพื่อแก้ไขภาพในโปรแกรมวาดรูป (Paint Editor)
- ปุ่ม Copy เพื่อทำซ้ำชุดนี้
- 🔹 ปุ่มลบซุด 🔯 เมื่อไม่ต้องการซุดนี้แล้ว

39

คลิกขวาที่ชุดใด จะมีเมนูให้เลือกดังนี้

- turn into new sprite จะสามารถสร้างตัวละครใหม่จากชุดนี้ได้
- export this costume จะสามารถส่งออกชุดนี้เพื่อเก็บไว้ในไฟล์ได้

เมื่อเลือกเวที แล้วคลิกแทป Backgrounds ของเวที เพื่อดู แก้ไข หรือเพิ่มเติมฉากหลัง ซึ่งมี ส่วนประกอบคล้ายกับชุดตัวละคร เพียงแต่ว่าภาพของฉากหลังมีขนาดไม่เกิน 480 x 360 ตาม ขนาดของเวที



การสร้างชุดละครหรือฉากหลัง

การสร้างชุดละครหรือฉากหลังทำได้หลายวิธี เมื่อคลิกแทป Costumes หรือแทป Backgrounds จะเห็น 3 ปุ่มสำหรับสร้างชุดตัวละครหรือฉากหลัง

Paint Import Camera

- คลิกปุ่ม Paint เพื่อวาดชุดใหม่ โดยใช้โปรแกรมวาดรูป (Paint Editor)
- คลิกปุ่ม Import เพื่อโหลดชุดที่มีมากับโปรแกรม Scratch (ชุดอย่างเดียว ไม่มีสคริปต์)
- คลิกปุ่ม Camera ถ้าเครื่องคอมพิวเตอร์มีกล้อง คลิกปุ่มรูปกล้องหรือกดแป้นเว้นวรรค เพื่อถ่ายรูป เมื่อถ่ายเสร็จคลิกปุ่ม Done ก็จะได้รูปออกมา เป็นชุดถัดไปของตัวละครที่ เลือกปัจจุบัน หรือภาพฉากหลังถัดไปของเวที



นอกจากนั้นยังสามารถลากรูปภาพ จากเว็บไซต์ โฟล์เดอร์ หรือเดสก์ท็อป มาใส่ในรายการของ ชุดตัวละครหรือภาพฉากหลังได้เลย

เสียงตัวละครและเสียงประกอบ



คลิกแทป Sounds ของ hungry fish เพื่อฟัง แก้ไข หรือเพิ่มเติมเสียงให้กับตัวละคร ตัวอย่างนี้มี เพียงเสียงเดียวในรายการ เสียงไม่ได้เกิดขึ้นตลอดเวลา แต่จะส่งเสียงเมื่อถูกเรียกใช้งานเท่านั้น

เสียงแต่ละเสียงมีส่วนประกอบดังนี้

- ชื่อ (chomp)
- ระยะเวลา (0:00:01)
- ขนาดไฟล์ (31 KB)
- ปุ่มเปิดเสียง 区 สำหรับทดสอบเสียง
- ปุ่มปิดเสียง 🔳
- ปุ่มลบเสียง 🖾 ถ้าไม่ต้องการเสียงนี้แล้ว

เมื่อคลิกขวาที่เสียงจะมีเมนู export this sound ให้เลือก เพื่อส่งออกเสียงนี้เก็บไว้ในไฟล์

การสร้างเสียง

คลิกแทป Sounds จะเห็น 2 ปุ่มสำหรับสร้างเสียง ให้กับตัวละครหรือเวที



คลิกปุ่ม Record จะได้โปรแกรมบันทึกเสียง (Sound Recorder) ออกมา เพื่อบันทึกเสียงจาก ผู้ใช้ ผ่านไมโครโฟนของเครื่องคอมพิวเตอร์ โปรแกรมบันทึกเสียงมีแถบบันทึกเสียง และเวลาที่ ใช้ไปในการบันทึกเสียงกำกับอยู่ด้วย



การบันทึกเสียงก็คล้ายกับเครื่องเสียงทั่ว ๆ ไป คลิกปุ่มวงกลมสีแดงเริ่มบันทึกเสียง เมื่อบันทึก เสร็จ คลิกปุ่มหยุดเสียง (สี่เหลี่ยม) จากนั้นสามารถคลิกปุ่มเล่นเสียง (สามเหลี่ยม) เพื่อทดสอบ เสียงที่บันทึกมา ถ้าไม่พอใจเสียงที่ได้มา สามารถบันทึกใหม่ได้ ด้วยการคลิกปุ่มบันทึกเสียงอีก ครั้ง บันทึกซ้ำกี่ครั้งก็ได้จนกว่าจะพอใจ ถ้าพอใจกับเสียงที่บันทึกแล้ว คลิกปุ่ม OK เพื่อนำไปใช้ งานต่อไป แต่ถ้าไม่ต้องการเสียงที่บันทึกแล้ว ก็คลิกปุ่ม Cancel เพื่อยกเลิกการบันทึกเสียง

คลิกปุ่ม Import เพื่อนำเข้าไฟล์เสียง ซึ่งไฟล์นั้นอาจบันทึกด้วยโปรแกรมบันทึกเสียงใน โปรแกรม Scratch หรือโปรแกรมบันทึกเสียงอื่นๆ หรือจะใช้ไฟล์เสียงที่มีอยู่ในโปรแกรม Scratch ก็ได้ เมื่อคลิกปุ่ม Import จะได้หน้าต่าง Import Sound แสดงโฟล์เดอร์เสียงในหมวด ต่างๆ ที่อยู่ในโปรแกรม Scratch

| Import Sound | |
|-----------------------------------|---|
| Computer Computer | Sounds |
| Desktop Č Desktop Sounds | Human Instruments Music Loops Percussion Vocals |
| | OK Cancel |

เมื่อคลิกเลือกโฟล์เดอร์ เช่น Animal จะเห็นไฟล์เสียงมากมาย เมื่อคลิกที่ไฟล์ใดจะได้ยินเสียง นั้นทันที แต่ถ้าต้องการนำเข้าไฟล์ใด ให้ดับเบิ้ลคลิกที่ไฟล์นั้น หรือคลิกเลือกไฟล์นั้น แล้วคลิก ปุ่ม OK

| Import Sound | |
|-----------------|--|
| Computer | Animal |
| kant Desktop | Bird Cat Cricket Crickets Dog1 Dog2 |
| Sounds | Duck Goose Horse HorseGallop Kitten Meow Owl |
| | OK Cancel |

บล็อกคำสั่ง

บล็อกคำสั่ง เป็นส่วนประกอบหลักของสคริปต์ ใช้สำหรับสั่งให้ตัวละครหรือเวทีทำงานตาม วัตถุประสงค์ของคำสั่งนั้นๆ

Scratch แบ่งบล็อกคำสั่งเป็นหมวดต่างๆ 8 หมวด มีสีต่างกัน ตามลักษณะหน้าที่ของแต่ละ หมวด เมื่อคลิกที่บล็อกหมวดใด จะเห็นคลังบล็อกคำสั่งในหมวดนั้น โดยบล็อกคำสั่งที่ปรากฏ จะขึ้นกับว่า ณ ขณะนั้น เลือกตัวละครหรือเวทีอยู่

| Motion | Control |
|--------|-----------|
| Looks | Sensing |
| Sound | Operators |
| Pen | Variables |

บล็อกคำสั่งสำหรับตัวละครกับเวทีบางหมวด อาจมีความแตกต่างกันบ้าง เนื่องจาก ความสามารถที่แตกต่างกัน เช่น เวทีเคลื่อนที่ไม่ได้ ดังนั้นจึงไม่มีบล็อกคำสั่งในหมวดการ เคลื่อนที่สำหรับเวทีเลย

คลังบล็อกคำสั่งสำหรับตัวละคร



บล็อกหมวดการเคลื่อนที่ (Motion) เกี่ยวกับการเคลื่อนที่ และการย้ายตำแหน่งของตัวละคร บล็อกหมวดรูปร่าง (Looks) เกี่ยวกับรูปร่างของตัวละคร เช่นสี ขนาด การแสดงคำพูด และ ความคิด รวมถึงเอฟเฟคต่างๆ



บล็อกหมวดเสียง (Sound) ใช้สำหรับเปิดเสียงดนตรีชนิดต่างๆ ผู้ใช้อาจอัดเสียงพูดด้วยตัวเอง แล้วนำมาใส่ในตัวละคร เหมือนกับการพากย์เสียง

บล็อกหมวดปากกา (Pen) ใช้สำหรับวาดรูปหรือประทับรูป (ของตัวละคร) บนเวที

| Motion Control Looks Sensing Sound Operators Pen Variables |
|---|
| when A clicked |
| when space key pressed |
| wait 1 secs |
| repeat 10 |
| broadcast got-mev |
| when I receive got-me |
| if |
| if else |
| wait until |
| stop script |
| |

บล็อกหมวดควบคุม (Control) ใช้ในการบังคับทิศทางการทำงานของสคริปต์เช่น การทำงานซ้ำ การตัดสินใจ การติดต่อสื่อสารกับตัวละครอื่นๆ และที่สำคัญใช้กำหนดเริ่มต้นและหยุดการ ทำงานของสคริปต์

| Motion Control Looks Sensing Sound Operators Pen Variables |
|--|
| touching ? touching color ? color is touching ? ask What's your name? and wai |
| mouse x mouse y mouse down? |
| key space pressed? distance to |
| <pre>timer x position of goldfish 1 loudness loudness loud?</pre> |
| slider sensor value |
| |

บล็อกหมวดกำลังรับรู้ (Sensing) ใช้รับสัญญานต่างๆ ที่มีต่อตัวละครหรือผู้ใช้งาน เช่น การชน กันของตัวละคร การกดคีย์ของผู้ใช้งาน



| Motion Control Looks Sensing Sound Operators Pen Variables | |
|---|--|
| | |
| pick random 1 to 10 | |
| and or | |
| join hello world letter 1 of world | |
| ength of world | |
| (sqrt of 10) | |

บล็อกหมวดโอเปอร์เรเตอร์ (Operators) เกี่ยวกับการคำนวณทางคณิตศาสตร์ ตรรกะ และการ จัดการตัวอักษร และข้อความ kidsangsan.com สมชาย พัฒนาชวนชม



บล็อกหมวดตัวแปร (Variables) ใช้ในการสร้างตัวแปรเพื่อเก็บข้อมูลเอาไว้ใช้ภายหลัง ตัวแปรที่ สร้างสามารถใช้ส่วนตัวของแต่ละตัวละคร หรือใช้ร่วมกันกับตัวละครทั้งหมดรวมทั้งเวทีด้วย

เมื่อสร้างตัวแปรแล้ว เช่น ตัวแปร score และตัวแปรลิสต์ level จะมีบล็อกสำหรับจัดการตัวแปร เพิ่มขึ้นมา

ถ้าคลิกกล่องหน้าตัวแปร จะปรากฏเครื่องหมายถูกภายในกล่อง (คลิกอีกครั้งเพื่อยกเลิก) เพื่อ ทำให้ตัวแปรนี้แสดงอยู่บนเวที

คลั่งบล็อกคำสั่งสำหรับเวที

| Motion Control | Motion Control |
|--|---|
| Looks Sensing | Looks Sensing |
| Sound Operators | Sound Operators |
| Pen Variables | Pen Variables |
| Stage selected: No motion blocks | Clear |
| Motion Control | Motion Control |
| Looks Sensing | Looks Sensing |
| Sound Operators | Sound Operators |
| Pen Variables | Pen Variables |
| switch to background backgrour next background background # change color effect by 25 set color effect to 0 clear graphic effects | when clicked when space key pressed when Stage clicked wait 1 secs |

เวทีไม่สามารถเคลื่อนที่ได้ ทำให้บล็อกคำสั่งหมวดการเคลื่อนที่ไม่มีเลย ส่วนหมวดอื่นๆ ก็มี บล็อกคำสั่งน้อยกว่าของตัวละคร เช่น หมวดปากกามีแต่บล็อกลบเท่านั้น เนื่องจากตัวละคร เท่านั้นที่วาดรูปได้ หมวดรูปร่างจะมีบล็อกคำสั่งเกี่ยวกับฉากหลังเท่านั้น ส่วนหมวดควบคุม มี บล็อกเหมือนกันหมด ยกเว้นบล็อก เมื่อเวทีถูกคลิก "when Stage clicked" (ดัดมาให้ดูเฉพาะ ส่วนที่ต่างกัน)

| Motion Control |
|--------------------------------|
| Looks Sensing |
| Sound Operators |
| Pen Variables |
| |
| ask What's your name? and wait |
| |
| |
| mouse x |
| mouse y |
| mouse down? |
| |
| key space pressed? |
| |
| reset timer |
| timer |
| x position of Sprite1 |
| |
| loudness) |
| loud? |
| |
| slider sensor value |
| sensor button pressed 2 |
| |
| |
| |
| |
| |
| |
| |

บล็อกหมวดรับรู้ ไม่มีบล็อกการรับรู้ทางสัมผัส และการรับรู้ระยะทาง

ส่วนบล็อกหมวดโอเปอรเรชั่น หมวดเสียง และหมวดตัวแปรเหมือนกันกับของตัวละคร

53

บล็อกยังถูกแบ่งตามรูปร่างและลักษณะการวางต่อกัน ซึ่งมีอยู่ 6 แบบดังนี้

1. Hat blocks



มี 4 บล็อกอยู่ในหมวดควบคุม เป็นบล็อกเริ่มต้นของทุกๆ สคริปต์ ด้านบนจะโค้งมนคล้าย หมวก ด้านล่างมีส่วนนูนยื่นออกมาสำหรับต่อกับบล็อกที่ตามมา

2. Stack blocks

เป็นบล็อกส่วนใหญ่ที่ทำงานหลักๆ สามารถเรียงต่อกันได้ ใน Scratch มี 75 บล็อก ด้านบน จะเว้าส่วนด้านล่างจะนูนทำให้มันวางด้านบน ด้านล่าง หรือระหว่างกลางบล็อกอื่นได้

3. Boolean blocks



เป็นบล็อกที่คืนค่าคำตอบถูกหรือผิด มีทั้งหมด 13 บล็อก อยู่ในบล็อกหมวดโอเปอร์เรชั่น และบล็อกหมวดกำลังรับรู้ รูปร่างของบล็อกจะเป็นหกเหลี่ยมไม่มีส่วนยื่นหรือเว้า ใช้ซ้อนอยู่ ข้างในบล็อกอื่นอีกที ส่วนใหญ่ใช้แสดงเงื่อนไขใน C blocks หรือ Boolean blocks ด้วย กันเอง 4. Reporter blocks

เป็นบล็อกที่ใช้คืนค่าได้ทั้งตัวเลขหรือตัวอักษร มีทั้งหมด 26 บล็อก โดยไม่รวมบล็อกที่เกิด จากการสร้างตัวแปรหรือลิสต์ในแต่ละครั้ง รูปร่างของบล็อกจะโค้งมนไม่มีส่วนยื่นหรือเว้า ใช้ ซ้อนอยู่ข้างในบล็อกอื่นอีกที ส่วนใหญ่ใช้ใน Stack blocks ที่ต้องการค่าสำหรับข้อมูลเข้า นอกจากนั้นยังใช้ใน C blocks และ Boolean blocks หรือแม้กระทั่งใน Report blocks ด้วย กันเองก็มี

5. C blocks



มีทั้งหมด 6 บล็อกอยู่ในบล็อกหมวดควบคุม เป็นบล็อกรูปร่างคล้ายตัว "C" ใช้ครอบบล็อก อื่นๆ เพื่อให้ทำงานตามเงื่อนไขหรือทำงานซ้ำหลายๆ รอบ รูปร่างของบล็อกภายในตัว "C" มีส่วยนูนเป็นจุดเริ่มต้นสำหรับการเชื่อมต่อกับบล็อกที่อยู่ภายใน ภายนอกตัว "C" มีส่วนเว้า ด้านบนและส่วนนูนด้านล่าง ทำให้บล็อกเหล่านี้ต่อด้านบนหรือด้านล่างกับบล็อกอื่นได้อีก

6. Cap blocks

มีอยู่ 2 บล็อกในหมวดควบคุม ใช้สำหรับหยุดการทำงานสคริปต์ ส่วนบนเว้าส่วนล่างเรียบ ทำให้บล็อกเหล่านี้เป็นบล็อกสุดท้ายในสคริปต์ หรือบล็อกสุดท้ายใน C blocks การเขียนโปรแกรมใน Scratch



การเขียนโปรแกรมใน Scratch คือการเขียนสคริปต์ให้ ตัวละครแต่ละตัวรวมทั้งเวทีทำงาน รวมกัน โดยที่ตัวละครแต่ละตัวรวมทั้งเวที สามารถมีได้หลายสคริปต์หรือไม่มีสคริปต์เลยก็ได้

สคริปต์ประกอบจากบล็อกคำสั่งเพื่อทำงาน ตามที่ออกแบบโปรแกรมไว้ รูปด้านบนเป็นสคริปต์ ของตัวละครปลา (hungry fish) ที่มีอยู่ 2 สคริปต์ แถบสีขาวที่ล้อมรอบสคริปต์แรก หมายถึง สคริปต์นี้กำลังทำงานอยู่

การสร้างสคริปต์

การเขียนโปรแกรมสคริปต์ เป็นการนำบล็อกมาเรียงต่อกันอย่างมีความหมาย หลักการและ ความเข้าใจในการเขียนโปรแกรม จะทำให้ผู้เรียนเขียนสคริปต์ได้ถูกต้องตามเป้าหมาย

การเขียนสคริปต์ให้กับตัวละครหรือเวที เริ่มต้นด้วยการเลือกตัวละครหรือเวทีก่อน จากนั้นคลิก แทปสคริปต์ ซึ่งเป็นพื้นที่สำหรับเขียนสคริปต์

การนำบล็อกมายังพื้นที่เขียนสคริปต์ เริ่มจากเลือกหมวดของบล็อกก่อน จากนั้นจะเห็นคลัง บล็อกคำสั่งในหมวดนั้นด้านล่าง ลากบล็อกที่ต้องการไปยังพื้นที่เขียนสคริปต์ ถ้าไม่ต้องการ บล็อกใด ๆ ที่อยู่ในพื้นที่เขียนสคริปต์แล้ว ก็ลากมันกลับมายังพื้นที่ของคลังบล็อกคำสั่งได้ หรือ คลิกขวาที่บล็อกนั้นแล้วเลือก delete

ข้อมูลสำหรับบล็อกคำสั่ง

บล็อกคำสั่งจะทำงานหรือทำหน้าที่อย่างใดอย่างหนึ่ง โดยที่บางบล็อกจะทำงานได้นั้นต้องรับ ข้อมูลเพิ่มเติม บางบล็อกสามารถคืนข้อมูลได้ บางบล็อกสามารถทั้งรับทั้งคืนข้อมูล ทั้งนี้ขึ้นอยู่ กับหน้าที่และวัตถุประสงค์ของแต่ละบล็อก

ข้อมูลที่ใช้ในบล็อกมีหลายชนิด ได้แก่ ตัวเลข ตัวอักษร สี ข้อมูลที่อยู่ในตัวแปร ข้อมูลรายการ เช่น คีย์บอร์ด โน๊ตดนตรี ประเภทเครื่องดนตรี ตัวละครที่สร้างขึ้น ฯลฯ

การใส่ข้อมูลในบล็อก

บางบล็อกจะมีแถบสีขาว สามารถใส่ข้อความหรือตัวเลขได้ เช่น บล็อกเคลื่อนที่ 🚾 ம steps ต้องการค่าตัวเลขจำนวนก้าวในการเคลื่อนที่ ถ้าต้องการเปลี่ยนค่านี้ ให้คลิกที่แถมสีขาว จากนั้น พิมพ์แก้ไขข้อมูล แถบสีขาวนี้ยังสามารถรับค่าที่คืนมาจากบล็อกอื่นได้ ด้วยการใส่บล็อกนั้นลง ในแถบสีขาวโดยตรง เช่น ค่าตำแหน่ง x ที่ได้จากบล็อก < position

บางบล็อกสามารถเลือกค่าได้จากเมนูรายการแบบดึงลง เช่น ^{set instrument to} **Iจ** โดยคลิกที่ สามเหลี่ยมจะเห็นรายการ จากนั้นคลิกเลือกรายการที่ต้องการ

โดยปกติ Scratch จะป้องกันการใส่ข้อมูลไม่ตรงกับชนิดข้อมูลที่บล็อกต้องการ ถ้าใส่ไม่ตรงจะใช้ ค่าโดยปริยาย (Default) ที่กำกับไว้แทน เช่น ถ้าบล็อกต้องการตัวเลข แต่พยายามใส่ตัวอักษร แทน ก็จะได้เลขค่าโดยปริยายที่กำกับไว้ แต่ถ้าไม่มีค่าใดกำกับเลย ก็จะเป็น 0 แทน ถ้าบล็อก ต้องการข้อมูลตัวอักษร ตัวเลขสามารถใส่ได้ เพราะถือว่าเป็นตัวอักษรด้วย ถ้าไม่ใส่เลยก็ได้ เพราะช่องว่างถือว่าเป็นอักษรว่างนั้นเอง ส่วนข้อมูลในรายการจะไม่มีปฏิหาเนื่องจากใช้วิธีคลิก เลือกรายการ

ลำดับการทำงานของบล็อกที่ซ้อนกัน

ปกติบล็อกในสคริปต์จะทำงานจากบนลงล่าง ถ้าเจอบล็อกควบคุมใด ๆ บล็อกที่อยู่ภายในก็จะ ทำงานจากบนลงล่างเหมือนกัน เมื่อบล็อกมีบล็อกอื่นซ้อนกันหลายชั้น จะเริ่มทำงานจากบล็อก ด้านในสุดก่อน เปรียบเสมือนการใส่วงเล็บซ้อนกันในทางคณิตศาสตร์ ซึ่งลำดับการทำงานจะทำ จากวงเล็บในสุดก่อน ถ้าอยู่ในระดับเดียวกัน จะทำจากซ้ายไปขวา เช่น เงื่อนไขจากบล็อกโอ เปอร์เรชั่นที่ซ้อนกันตามรูปนี้ สามารถเขียนในรูปแบบวงเล็บได้ดังนี้



((battconnected = 1) and ((Switch = 1) and (bulbconnected =1)))

สคริปต์จะตรวจสอบข้อเท็จจริงจากเงื่อนไข Switch = 1 ก่อน ถัดไปตรวจสอบ bulbconnected = 1 ตามด้วยการตรวจสอบข้อเท็จจริงจากเงื่อนไข battconnected = 1

จากนั้นหาข้อเท็จจริงจากโอเปอร์เรชั่นตรรกะและ (ขวา) ระหว่างข้อเท็จจริง Switch = 1 และ bulbconnected = 1 คำตอบที่ได้นำมาหาข้อเท็จจริงสุดท้ายด้วยโอเปอร์เรชั่นตรรกะและ (ซ้าย) กับคำตอบที่ได้จากเงื่อนไข battconnected = 1

ความช่วยเหลือการใช้บล็อก

ใน Scratch มีบล็อกมากมาย จะรู้ได้อย่างไรว่ามันทำงานอย่างไร นอกจากคู่มือที่อ่านได้จาก เว็บไซต์ หรือไฟล์คู่มือ PDF แล้ว ในตัวโปรแกรม Scratch เองก็มีวิธีการใช้งานฉบับย่อให้ด้วย โดยการคลิกขวาบล็อกที่ต้องการรู้การทำงานของมัน ซึ่งจะเป็นบล็อกที่อยู่ในสคริปต์ หรือบล็อก ที่อยู่ในคลังบล็อกคำสั่งก็ได้ จากนั้นเลือก Help

ตัวอย่างเช่น Help ของบล็อกคำสั่งเพื่อเปลี่ยนภาพฉากหลังเวทีภาพถัดไป (next background) จะได้รูปเหมือนข้างล่างนี้



ในหน้าต่าง Help จะอธิบายหลักการใช้งาน และวัตถุประสงค์ของบล็อกนี้ พร้อมทั้งตัวอย่างการ ใช้งานให้ดูด้วย

การต่อบล็อก

บล็อกที่สามารถเรียงต่อกันได้ เมื่อลากบล็อกเข้าไปใกล้ๆ กับบล็อกเป้าหมาย จะเห็นแถบสีขาว ขึ้น เป็นการบอกว่าบล็อก 2 บล็อกนี้สามารถต่อกันได้ เมื่อวางบล็อกลง บล็อกจะต่อได้สนิทกับ บล็อกเป้าหมาย



บล็อกที่ต่อกันไม่ได้ จะไม่เห็นแถบสีขาวนี้ และไม่สามารถบังคับให้ต่อกันได้ ถ้าวางบล็อกลงไป บล็อกจะอยู่เหลื่อมกับบล็อกเป้าหมาย แต่จะไม่ต่อกัน



โดยทั่วไป รูปร่างของบล็อกจะบงบอกอยู่แล้วว่าบล็อกสามารถต่อกันได้หรือไม่ ลองสังเกตจาก สองตัวอย่างที่ผ่านมา การใช้แถบสีขาว เป็นตัวช่วยอีกอย่างหนึ่ง เพื่อลดความยากในการเขียน โปรแกรม และป้องกันการเกิดข้อผิดพลาดในไวยากรณ์ของโปรแกรมภาษา ซึ่งเกิดขึ้นบ่อยๆ ใน โปรแกรมภาษาอื่น

การซ้อนบล็อก

บล็อกสามารถซ้อนกันได้หลาย ๆ ชั้น การซ้อนบล็อกเข้าไปข้างในของบล็อกอื่น ก็คล้ายกับการ ต่อบล็อก คือเมื่อลากบล็อกเข้าไปใกล้ ๆ บล็อกที่ต้องการจะซ้อนบล็อกภายใน จะเห็นแถบสีขาว ล้อมรอบบริเวณที่สามารถวางบล็อกได้ เป็นการบอกว่า ตำแหน่งนี้สามารถวางบล็อกได้ บล็อกที่ วางไม่ได้จะไม่เห็นแถบสีขาวนี้ และไม่สามารถบังคับให้วางได้ ถ้าวางไปมันก็จะลอยอยู่ด้านบน แต่ไม่เข้าไปด้านในบล็อก



การนำบล็อกออก

การนำบล็อกที่ไม่ต้องการออกจากสคริปต์ ทำได้โดยคลิกเลือกบล็อกแล้วลากออกมาจากสคริปต์ โดยที่ตำแหน่งของบล็อกในสคริปต์จะทำให้การดึงบล็อกออกมาได้ไม่เหมือนกัน เช่นตัวอย่าง ด้านล่างนี้

- (1) เป็นสคริปต์เริ่มต้น
- (2) ถ้าบล็อกที่ต้องการเอาออกอยู่ล่างสุดของสคริปต์ หรือบล็อกล่างสุดที่อยู่ใน C บล็อก บล็อกนั้นบล็อกเดียวจะถูกดึงออกมา
- (3) แต่ถ้าบล็อกนั้นมีบล็อกอื่นตามมาติดกัน บล็อกที่ตามมาจะถูกดึงออกมาด้วย
- (4) ถ้าดึง C บล็อกออกมา บล็อกที่อยู่ภายใน C บล็อกนั้น (และบล็อกอื่นที่ต่อตาม C บล็อก ด้วย) ก็จะถูกดึงออกตามมาด้วย



การรันสคริปต์



การรันสคริปต์คือการสั่งให้สคริปต์ทำงาน ทำได้สองวิธี คือ รันสคริปต์เฉพาะตัวละครหรือเวที กับ รันสคริปต์ทั้งโปรเจก (ทุกตัวละครและเวที)

การรันสคริปต์ของตัวละคร (หรือเวที) ใช้ทดสอบการทำงานของสคริปต์ของตัวละครนั้นๆ (หรือ เวที) ทำได้ด้วยการดับเบิ้ลคลิกที่บล็อกใดก็ได้ในสคริปต์นั้น สคริปต์ใดทำงานอยู่ จะมีเส้นสีขาว ล้อมรอบสคริปต์นั้น ในตัวอย่างนี้ ตัวละคร hungry fish มีสคริปต์กำลังทำงานพร้อมกัน 2 สคริปต์

สคริปต์แต่ละสคริปต์จะทำงานจากบนลงล่าง ถ้าสคริปต์ทำงานเพียงรอบเดียว สคริปต์จะหยุด การทำงานเองเมื่อบล็อกสุดท้ายทำงาน แต่ถ้าสคริปต์ทำงานหลายรอบ ก็จะทำงานซ้ำจนกว่าจะ ครบรอบตามที่กำหนดไว้ ก่อนที่จะหยุดทำงาน แต่ถ้าสคริปต์ทำงานตลอดเวลา (ในบล็อก forever) สคริปต์จะทำงานซ้ำไปเรื่อย ๆ และถ้าต้องการหยุดทำงานสคริปต์นั้น ทำได้ด้วยการ ดับเบิลคลิกที่บล็อกใดก็ได้ของสคริปต์นั้นอีกครั้ง 64

การรันสคริปต์ทั้งโปรเจกของทุกตัวละครและเวทีพร้อมกัน คือการคลิกปุ่มธงเขียว 🎘 ที่อยู่มุม ขวาบนของเวที ทุกสคริปต์ของทุกตัวละครและเวที ที่เริ่มตันสคริปต์ด้วยบล็อกธงเขียว

when Acticked จะทำงานพร้อมกัน และถ้าต้องการหยุดการทำงานทั้งโปรเจก ให้คลิกปุ่ม แปดเหลี่ยมสีแดง • ถัดจากปุ่มธงเขียว

หลักการในการเขียนโปรแกรม

หลักการและแนวคิดในการเขียนโปรแกรมมีมากมาย บางภาษาคอมพิวเตอร์ก็มีหลักการและ แนวคิดเหมือนกันบ้าง ต่างกันบ้าง ดังนั้นความเข้าใจในหลักการและแนวคิดที่มีใน Scratch สามารถนำไปประยุกต์ใช้ได้อีก ในการเรียนภาษาคอมพิวเตอร์อื่นๆ ต่อไป

หลักการและแนวคิดที่มีใน Scratch บางอย่างต้องเอาออก เพราะอาจยากเกินไปสำหรับเด็ก แต่ เพียงพอต่อการทำโปรเจกได้มากมาย หลักการและแนวคิดดังกล่าวมีดังนี้

- Sequence การทำงานแบบเรียงลำดับ
- Control flow การควบคุมลำดับการทำงาน มีสองแบบ คือแบบวนซ้ำ (Loop) และแบบมี เงื่อนไข (Condition)
- Parallelism การทำงานแบบคู่ขนาน
- Operation: การประมวณผลอักขระข้อความ (String) หรือการคำนวณทางคณิตศาสตร์ (Logic, Mathematical functions)
- Input การส่งข้อมูลจากผู้ใช้ให้กับโปรแกรมขณะทำงาน
- Message การส่งสารระหว่างตัวละครต่างๆ และเวที
- Event and event handling เหตุการณ์และการตอบสนองต่อเหตุการณ์นั้นๆ
- Synchronization การประสานและส่งต่อการทำงาน
- user interaction การติดต่อและโต้ตอบระหว่างผู้ใช้กับโปรแกรม
- User interface design การออกแบบวิธีการและช่องทางการสื่อสารระหว่างโปรแกรม กับผู้ใช้
- Variable and structured variable ตัวแปร และตัวแปรที่มีโครงสร้าง

การทำงานแบบเรียงลำดับ (Sequence)

การสร้างโปรแกรมใน Scratch ผู้เขียนโปรแกรมต้องคิดอย่างเป็นระบบ เกี่ยวกับลำดับขั้นตอน การทำงาน เพื่อใช้ในการสร้างสคริปต์ โดยการนำบล็อกคำสั่งมาเรียงต่อกันอย่างมีความหมาย ตามเป้าหมายที่วางไว้

ตัวอย่างเช่น การให้ตัวละครวาดรูปสี่เหลี่ยมหนึ่งรูป จะประกอบไปด้วยการทำงานย่อยหลายๆ คำสั่งด้วยกัน นั้นคือ การจรดปากกาลงเตรียมวาดรูป การเคลื่อนที่ไปข้างหน้า 50 ก้าว จากนั้น เปลี่ยนทิศทางทวนเข็มนาพิกาไป 90 องศา แล้วก็เคลื่อนที่ไปอีก 50 ก้าว เปลี่ยนทิศทางทวน เข็มนาพิกาอีก 90 องศา ทำแบบนี้เพิ่มอีก 2 รอบ ก็จะได้การวาดรูปสี่เหลี่ยมหนึ่งรูป จากนั้นจึง ยกปากกาขึ้น



จะเห็นได้ว่าการเขียนสคริปต์ให้ทำงานอะไรสักอย่าง มักจะเริ่มจากภาพรวมหรือเป้าหมายก่อน จากนั้นแจกแจงงานเป็นลำดับขั้นตอน สร้างงานย่อยๆ และประกอบเข้าด้วยกัน เพื่อให้งานใหญ่ นั้นสำเร็จตามเป้าหมาย

67

การควบคุมลำดับการทำงาน (Control flow)

งานบางอย่างง่ายพอที่จะทำแบบเรียงลำดับ ลำดับการทำงานของสคริปต์จึงตรงไปตรงมา ทำงานเรียงลำดับจากบล็อกบนสุด ลงไปจนถึงบล็อกล่างสุด แต่งานบางอย่างมีความซับซ้อน มากขึ้น การทำงานแบบเรียงลำดับอาจไม่เพียงพอ หรือไม่มีประสิทธภาพ

ดังนั้นจึงมีการควบคุมลำดับการทำงาน อีกสองลักษณะเพื่อเพิ่มขีดความสามารถในการเขียน โปรแกรม นั้นก็คือ การควบคุมลำดับการทำงานแบบวนซ้ำ (Loop) และการควบคุมลำดับการ ทำงานแบบมีเงื่อนไข (Condition) โดยใช้บล็อกทำซ้ำและบล็อกเงื่อนไข ที่มีอยู่หลายแบบให้ เลือกใช้ในบล็อกหมวดควบคุม

การควบคุมลำดับทำงานแบบวนซ้ำ (Loop)

การควบคุมลำดับการทำงานแบบวนซ้ำ เป็นการทำซ้ำงาน (บล็อกที่อยู่ภายในบล็อกทำซ้ำ) เหมือนเดิมหลาย ๆ ครั้ง เช่น การให้ตัวละครวาดรูปสี่เหลี่ยมนั้น ส่วนหนึ่งของงานนี้ คือให้ตัว ละครเคลื่อนที่เป็นสี่เหลี่ยม ซึ่งมีลักษณะการทำงานซ้ำกัน คือเคลื่อนที่ไปข้างหน้า 50 ก้าว จากนั้นเปลี่ยนทิศทวนเข็มนาพิกาไป 90 องศา ทำแบบนี้ซ้ำกัน 4 รอบจะได้รูปสี่เหลี่ยม ถึงแม้ การเขียนแบบเรียงลำดับจะตรงไปตรงมา แต่โปรแกรมจะยาวเกินไป ถ้าใช้การควบคุมลำดับการ ทำงานแบบวนซ้ำ จะได้สคริปต์ใหม่ที่สั้นลงและอ่านทำความเข้าใจได้ง่ายกว่าดังนี้

| pen down | |
|-------------------|--|
| repeat 4 | |
| move 50 steps | |
| turn 👌 90 degrees | |
| | |
| pen up | |
| <u> </u> | |

การควบคุมลำดับการทำงานแบบวนซ้ำ มีหลายลักษณะ ตามแต่ละชนิดของบล็อกควบคุม เช่น



การทำงานซ้ำที่ไม่กำหนดจำนวนรอบ คือทำไปตลอดจนกว่าจะจบโปรแกรม



การทำงานซ้ำที่กำหนดจำนวนรอบที่แน่นอน เช่น 10 รอบ



การทำงานซ้ำที่ขึ้นกับเงื่อนไข โดยก่อนจะทำซ้ำ ต้องตรวจสอบเงื่อนไขก่อน ว่ายังเป็นจริงอยู่ หรือไม่ ถ้าเงื่อนไขเป็นจริงจึงจะทำซ้ำ หรือถ้าเงื่อนไขเป็นจริงจึงหยุดทำซ้ำ

การควบคุมลำดับการทำงานแบบมีเงื่อนไข (Condition)

เงื่อนไขถูกใช้ในการควบคุมทิศทางการทำงาน โดยการตัดสินใจที่จะทำงานหรือไม่ทำ หรือต้อง เลือกทำงานอย่างใดอย่างหนึ่ง ขึ้นอยู่กับเงื่อนไขเป็นตัวกำหนด เช่น



ถ้าเงื่อนไขเป็นจริง จะทำงานที่อยู่ภายในบล็อกนี้



ถ้าเงื่อนไขเป็นจริง จะทำงานที่อยู่ภายในส่วนบน (ใต้ if) แต่ถ้าเป็นเท็จ จะทำงานที่อยู่ภายใน ส่วนล่าง (ใต้ else)

เงื่อนไขยังใช้ควบคุมการทำงานรวมกับการควบคุมแบบอื่นอีก เช่น



ถ้าเงื่อนไขเป็นจริง จะทำซ้ำงานที่อยู่ภายในบล็อกนี้



ถ้าเงื่อนไขเป็นจริง จะหยุดทำซ้ำงานที่อยู่ภายในบล็อกนี้

wait until 🔵

ถ้าเงื่อนไขเป็นเท็จ จะหยุดการทำงานชั่วคราว จนกว่าเงื่อนไขจะเป็นจริงจึงจะทำงานในลำดับ ต่อไป

โอเปอร์เรชั่น (Operation)

การเขียนโปรแกรมต้องอาศัยการคำนวณไม่มากก็น้อย โอเปอร์เรชั่นเป็นเครื่องมือที่ใช้ในการ คำนวณ หรือการจัดการกับข้อความตัวอักษรที่ใช้ในโปรแกรม

โอเปอร์เรชั่นใน Scratch เป็นโอเปอร์เรชั่นพื้นฐาน แต่ก็เพียงพอต่อการทำโปรเจก โอเปอร์เรชั่น ที่ซับซ้อนสามารถสร้างได้ ด้วยการประกอบโอเปอร์เรชั่นพื้นฐานเข้าด้วยกันตามหลัก คณิตศาสตร์

โอเปอร์เรชั่นพีชคณิตและฟงัก์ชันต่างๆ

โอเปอร์เรชั่นตรรกะ

โอเปอร์เรชั่นจัดการตัวอักษรและข้อความ

join _____length of _____letter __ of ___

พื้นที่สีขาวภายในบล็อกหมายถึงผู้ใช้สามารถใส่ข้อมูลเข้าไปได้จากคีย์บอร์ด หรือใส่บล็อกก็ได้ พื้นที่สีขาวทรงมนรับค่าเป็นตัวเลขและรับบล็อกทรงมน พื้นที่ทรงหกเหลี่ยมรับได้เฉพาะข้อมูลที่ เป็นจริงหรือเท็จจากบล็อกทรงหกเหลี่ยมเท่านั้น ส่วนพื้นที่สีขาวทรงสี่เหลี่ยมรับข้อมูลที่เป็น ตัวอักษร ทำให้รับตัวเลขก็ได้ (ถือว่าเป็นตัวอักษรด้วย) รับบล็อกทรงมนและบล็อกทรงหก เหลี่ยมก็ได้ด้วย

การทำงานแบบคู่ขนาน (Parallelism)

ใน Scratch ประกอบไปด้วยตัวละครต่างๆ รวมทั้งเวที ซึ่งสามารถทำงานพร้อมๆ กันได้ ณ ขณะหนึ่งๆ เรียกการทำงานแบบนี้ว่าการทำงานแบบคู่ขนาน ซึ่งใกล้เคียงกับชีวิตจริงของคนเรา ที่เกิดการทำงานหลายๆ อย่างพร้อมกันไปในเวลาเดียวกัน เช่น ในงานปาร์ตี้เต้นรำที่มีผู้คน มากมาย บางคนก็พูดคุยกัน บางคนก็กำลังดื่มกินอาหาร และหลายๆ คนก็กำลังเต้นไป พร้อม กับเสียงเพลงที่เปิดจากเวที

ใน Scratch การทำงานแบบคู่ขนาน เป็นการทำงานจากหลาย ๆ สคริปต์พร้อมกัน แต่ละสคริปต์ นั้นอาจมาจากตัวละครเดียวกันก็ได้ หรือจากหลาย ๆ ตัวละคร รวมทั้งเวทีก็ได้ สคริปต์เหล่านี้ถูก กระตุ้นให้ทำงานพร้อมกัน ณ ขณะหนึ่ง ๆ แต่ไม่จำเป็นต้องเริ่มหรือจบสคริปต์พร้อมกัน เช่นเมื่อ ผู้ใช้คลิกธงเขียวเริ่มต้นโปรแกรม สคริปต์ที่ตอบสนองต่อการคลิกธงเขียวทุกสคริปต์จะเริ่มต้น ทำงานพร้อมกัน แต่ขณะที่สคริปต์เหล่านั้นทำงานอยู่ บางสคริปต์กำลังถูกกระตุ้นจากผู้ใช้ เช่น การกดคีย์หรือการคลิกตัวละคร ให้เริ่มทำงาน หรือถูกกระตุ้นจากสคริปต์อื่นด้วยการส่งสาร

บล็อกที่ควบคุมให้สคริปต์เริ่มทำงาน (พร้อมกัน) ได้จะอยู่บนสุดของสคริปต์ ซึ่งมีหลายแบบ เช่น

| when 🛤 clicked | ตอบสนองเมื่อผู้ใช้คลิกธงสีเขียวเริ่มต้นโปรแกรม |
|--------------------------|--|
| when space v key pressed | ตอบสนองเมื่อผู้ใช้กดคีย์ (space) ต่างๆ บนคีย์บอร์ด |
| when Sprite1 clicked | ตอบสนองเมื่อผู้ใช้คลิกตัวละครใดๆ บนเวที |
| when I receive | ตอบสนองเมื่อได้รับสารจากตัวละครหรือเวที ที่เป็นผู้ส่งสาร |

72

การสื่อสารภายในและภายนอกของโปรแกรม

ขณะโปรแกรมทำงาน ผู้ใช้สามารติดต่อสื่อสารกับโปรแกรมได้ โดยการติดต่อหรือให้ข้อมูลกับ โปรแกรม ภายในโปรแกรมเอง ตัวละครต่างๆ และเวที สามารถติดต่อสื่อสารกัน ผ่านการส่งสาร หรือแชร์ข้อมูลระหว่างกัน

การติดต่อสื่อสารกันทำให้เกิดเหตุการณ์ขึ้น อย่างไรก็ตาม เหตุการณ์ที่เกิดขึ้นอาจเป็นการ สื่อสารทางเดียว ถ้าอีกฝ่ายไม่รับรู้ถึงเหตุการณ์นั้น ดังนั้นต้องมีการตรวจจับรับรู้ถึงเหตุการณ์ที่ สนใจ และเหตุการณ์ที่ถูกตรวจจับได้เท่านั้นจึงจะมีการตอบสนองต่อเหตุการณ์นั้น

การติดต่อและให้ข้อมูลจากผู้ใช้ (Input)

ขณะโปรแกรมทำงาน ผู้ใช้สามารถติดต่อหรือให้ข้อมูลใหม่ ๆ กับโปรแกรมได้ โดยทำได้หลาย ทาง เช่น การคลิกด้วยเมาส์ การกดคีย์บอร์ด การเปลี่ยนระดับเสียง ผ่านการพูดใส่ไมโครโฟน การปรับค่าข้อมูลของตัวแปร ผ่านการเลื่อนสไลส์เดอร์ เป็นต้น

การส่งสารระหว่างตัวละครและเวที (Message)

เวทีและตัวละครต่างๆ สามารถติดต่อสื่อสารกันได้ ด้วยการส่งสารระหว่างกัน โดยเริ่มจากการ สร้างสารขึ้นมาก่อน จากนั้นตัวละครหรือเวทีสามารส่งสารออกไปได้ ด้วยการกระจายสารซึ่งไม่ ระบุผู้รับ เวทีหรือตัวละครใดที่สนใจรับสารนี้ ก็สามารถตรวจจับสารนี้ และตอบสนองต่อสารที่รับ ได้
การเกิดเหตุการณ์ (Events)

ในขณะที่โปรแกรมกำลังทำงาน มีเหตุการณ์เกิดขึ้นมากมายในโปรแกรม ทั้งจากการติดต่อและ โต้ตอบระหว่างโปรแกรมกับผู้ใช้ ระหว่างตัวละครกับเวที หรือระหว่างตัวละครด้วยกันเอง

เหตุการณ์ที่เกิดจากผู้ใช้ เช่น ผู้ใช้คลิกธงเขียวเริ่มต้นโปรแกรม ผู้ใช้คลิกตัวละคร ผู้ใช้พิมพ์ ข้อความตอบคำถามจากตัวละคร เหตุการณ์ที่เกิดจากตัวละครหรือเวที เช่น การชนขอบเวที ของตัวละคร การชนหรือการสัมผัสกันเองระหว่างตัวละคร การส่งสารระหว่างตัวละคร เป็นต้น

การตอบสนองต่อเหตุการณ์ (Event handling)

เมื่อเหตุการณ์เกิดขึ้น เหตุการณ์ที่ได้รับความสนใจเท่านั้น ที่จะมีการตอบสนองต่อเหตุการณ์ นั้นๆ การตอบสนองต่อเหตุการณ์นั้น ต้องอาศัยการตรวจจับ หรือการรับสารว่าเหตุการณ์นั้นๆ เกิดขึ้นหรือยัง

ใน Scratch ใช้บล็อกควบคุมรูปทรงหมวก เป็นตัวกำหนดว่า เหตุการณ์ไหนได้รับความสนใจ ส่วนสคริปต์ใต้บล็อกควบคุมดังกล่าว เป็นงานที่ตอบสนองต่อเหตุการณ์นั้นๆ

เหตุการณ์จะไม่มีความหมายเลย ถ้าไม่มีสคริปต์ใดคอยตัวจับเหตุการณ์เหล่านั้น เช่น ถ้าไม่มี

สคริปต์ใดเลยตรวจจับและตอบสนองการคลิกธงเขียว ^{[when} A clicked] จากผู้ใช้ โปรแกรมที่ เขียนมา จะไม่สามารถเริ่มต้นทำงานได้เลย

การประสานงาน (Synchronization)

การทำงานของตัวละครและเวที อาจเกิดขึ้นพร้อมๆ กันในลักษณะคู่ขนาน (Parallelism) ที่ต่าง คนต่างทำงานของตนไป แต่ลักษณะการทำงานอีกแบบหนึ่ง คือการทำงานร่วมกัน ที่มีลำดับ ขั้นตอนการทำงานระหว่างกัน บางครั้งต้องคอยรับงานจากคนอื่น บางครั้งต้องส่งงานให้คนอื่น ทำต่อ ทุกคนต้องทำงานประสานกัน ให้เป็นจังหวะเดียวกัน เพื่อให้งานใหญ่สำเร็จลุร่วงไปได้

การประสานงานให้เป็นจังหวะเดียวกัน คล้ายกับการส่งไม้ในการวิ่งผลัด นักวิ่งคนแรกส่งไม้ให้ นักวิ่งคนที่สองวิ่งต่อ จากนั้นส่งไม้ให้นักวิ่งคนถัดไปวิ่งต่อ จนกระทั้งนักวิ่งคนสุดท้ายวิ่งเข้าเส้น ชัย

การประสานงานระหว่างตัวละครและเวทีใน Scratch ใช้วิธีการสร้างสาร (Message) เพื่อใช้ สื่อสารระหว่างกัน เปรียบเสมือนไม้ผลัด เมื่อต้องการส่งสารเพื่อส่งต่อการทำงานระหว่างกัน ก็ จะส่งกระจายสารออกไป (Broadcast) ผู้รับที่สนใจในสารนี้ ก็จะรับสารแล้วทำงานต่อไป เปรียบเสมือนการส่งไม้ผลัด ให้คนที่รับไม้วิ่งต่อไปนั้นเอง

การสร้างและส่งกระจายสารออกไป ทำได้ด้วยบล็อกส่งกระจาย **broadcast** การส่งกระจาย สารทำให้เกิดเหตุการณ์ขึ้น เวทีหรือตัวละครใด ที่สนใจรับสารนี้ สามารถทำได้โดยการเริ่มต้น สคริปต์ด้วยบล็อก **when I receive** งานที่ตามมาต่อจากบล็อกนี้ คือการตอบสนองต่อ เหตุการณ์ที่ได้รับสารนี้

การสร้างสารใหม่มาใช้เอง การส่งกระจายสารทำให้เกิดเหตุการณ์ขึ้น การตอบสนองต่อสารที่ สนใจ ทำให้เวทีและตัวละครต่างๆ สามารถสื่อสารระหว่างกันได้ และสามารถส่งต่อการทำงาน ให้สอดคล้องกันได้

การโต้ตอบระหว่างผู้ใช้กับโปรแกรม (User interaction)

การโต้ตอบระหว่างผู้ใช้กับโปรแกรม เป็นกระบวนการต่อเนื่อง ที่ผู้ใช้ติดต่อสื่อสารกับโปรแกรม ด้วยการให้ข้อมูลในรูปแบบต่างๆ ซึ่งทำให้เกิดเหตุการณ์ในโปรแกรม ตามลักษณะการให้ข้อมูล นั้น เมื่อโปรแกรมรับรู้และตรวจจับเหตุการณ์ที่เกิดขึ้น ก็สามารถตอบสนองต่อเหตุการณ์นั้นๆ ได้ตามที่ได้เขียนโปรแกรมเอาไว้

การโต้ตอบระหว่างผู้ใช้กับโปรแกรม เป็นเหตุการณ์แบบหนึ่งที่อาจเกิดเมื่อใดก็ได้ เกิดชั่วครั้ง ชั่วคราวก็ได้ เกิดอย่างต่อเนื่องก็ได้ หรือไม่เกิดเลยก็ได้ ขึ้นกับการออกแบบและชนิดของ โปรแกรม เช่น โปรแกรมแอนิเมชั่น ที่ผู้ใช้อาจดูอย่างเดียว ไม่โต้ตอบกับโปรแกรมเลย โปรแกรมตอบโจทย์คณิตศาสตร์ ที่ผู้ใช้ต้องตอบคำถามกับโปรแกรม ด้วยการคลิกเมาส์ หรือกด คีย์เลือกคำตอบ หรือโปรแกรมเกม ที่ผู้ใช้ต้องโต้ตอบกับโปรแกรมตอดเวลา เป็นต้น

การออกแบบช่องทางการติดต่อระหว่างโปรแกรมกับผู้ใช้ (User interface design)

โปรแกรมรับรู้การติดต่อ เมื่อได้รับข้อมูลจากผู้ใช้ แต่จะตอบสนองเฉพาะสิ่งที่โปรแกรมสนใจ ซึ่ง ต้องถูกเขียนโปรแกรมไว้ล่วงหน้ามาแล้ว เช่น เขียนโปรแกรมให้รับรู้และตอบสนองต่อการกด คีย์ x ถ้าผู้ใช้กดคีย์ x โปรแกรมก็จะตอบสนองตามที่ได้เขียนโปรแกรมไว้ แต่ถ้าผู้ใช้กดคีย์ y ซึ่ง โปรแกรมไม่รับรู้การกดคีย์ y จึงไม่ทำอะไรตอบสนอง

การรับรู้และตอบสนองต่อการกดคีย์ x เป็นการสร้างช่องทางให้ผู้ใช้ติดต่อสื่อสารกับโปรแกรม ในขณะที่การกดคีย์ y ไม่ได้เป็นช่องทางติดต่อระหว่างผู้ใช้กับโปรแกรม

นอกจากผู้ใช้ จะสามารถติดต่อโปรแกรมผ่านอุปกรณ์ต่าง ๆ ของคอมพิวเตอร์ (เมาส์ คีย์บอร์ด ไมโครโฟน) ได้แล้ว การสร้างตัวละครขึ้นมา เพื่อเป็นอีกช่องทางหนึ่งให้ผู้ใช้ติดต่อกับโปรแกรม โดยเฉพาะ ก็สามารถทำได้ โดยตัวละครเหล่านี้จะอยู่กับที่ คล้ายปุ่ม พร้อมจะทำงาน เมื่อผู้ใช้ คลิกที่ตัวละครเหล่านี้

ตัวแปร (Variables)

ตัวแปรเป็นที่พักของข้อมูล อ้างอิงด้วยชื่อของตัวแปร สามารถเปลี่ยนแปลงแก้ไขข้อมูลได้ หรือ เรียกดูข้อมูลในภายหลังได้ ตัวแปรข้อมูลภายใน ที่พร้อมใช้งานได้เลย มีหลายตัวแปร เช่น ตัว แปรเก็บข้อมูลคำตอบจากผู้ใช้ อารwer ตัวแปรเก็บค่าตำแหน่งของตัวละคร **(x position**) (y position) เป็นต้น

ตัวแปรยังถูกสร้างขึ้นมาใหม่ได้ เมื่อมีข้อมูลที่ต้องใช้ต่อเนื่องในโปรแกรม เช่น เมื่อต้องการเก็บ คะแนนในเกมระหว่างการเล่น ก็สามารถสร้างตัวแปร **score** มาเก็บคะแนนได้ ตัวแปรที่สร้างขึ้น ใหม่ยังกำหนดได้ว่า ให้ใช้เฉพาะกับตัวละครที่สร้างตัวแปรนี้ หรือให้ทุกตัวละครใช้ตัวแปรนี้ ร่วมกันได้

ตัวแปรยังสามารถลดความซับซ้อนของการคำนวณยาว ๆได้ การคำนวณที่ใช้บ่อย ๆ ในหลาย ๆ แห่งในโปรแกรม สามารถคำนวณเพียงครั้งเดียว แล้วเก็บผลลัพธ์ไว้ในตัวแปร จากนั้นนำไปใช้ อีกได้เลยไม่ต้องคำนวณใหม่

ลิสต์ (List)

ลิสต์เป็นตัวแปรที่ใช้เก็บข้อมูลหลายๆ อย่างเข้าด้วยกันอย่างเป็นระบบ ข้อมูลลักษณะเดียวกัน ควรเก็บไว้ที่เดียวกันให้เป็นระบบ ทำให้ง่ายต่อการใช้งาน

ลักษณะของข้อมูลที่เก็บรวมกันอย่างเป็นระบบ มีให้เห็นทั่วไปใน Scratch เช่น การเก็บชุดตัว ละคร ภาพฉากหลังของเวที รายการที่ให้เลือกในบล็อกคำสั่งต่างๆ เช่น ประเภทของเอฟเฟค ตัวโน๊ตดนตรี ชนิดเครื่องดนตรี ทิศการเคลื่อนที่ เป็นต้น

ลิสต์เหมือนตัวแปร สามารถสร้างขึ้นได้ เช่น ข้อมูลชื่อของอาวุธต่างๆ ในเกม แทนที่จะเก็บ แยกกันในแต่ละตัวแปร ซึ่งจะเพิ่มขึ้นเรื่อยๆ เมื่อมีอาวุธใหม่เพิ่มขึ้น ทำให้ยากต่อการจดจำและ การใช้งานตัวแปรเหล่านั้น แต่ถ้าเก็บในลิสต์ จะมีเพียงตัวแปรลิสต์อาวุธตัวแปรเดียว ข้อมูลชื่อ ของอาวุธจะถูกเก็บไว้ภายในรายการของลิสต์แทน

โปรเจกใน Scratch

การศึกษาการเขียนโปรแกรมภาษา Scratch ในหนังสือเล่มนี้ เริ่มจากการหัดใช้งานตัวโปรแกรม ก่อนในเบื้องต้น จากนั้นเริ่มเรียนรู้ส่วนประกอบต่างๆ เพื่อสร้างโปรเจก พร้อมกับเรียนรู้แนวคิด หลักการและเทคนิคต่างๆ ในการเขียนโปรแกรม

ในส่วนนี้เป็นการศึกษาการเขียนโปรแกรมผ่านโปรเจกต่างๆ ที่เสร็จเรียบร้อยแล้ว เป็นวิธีการที่ เริ่มจากภาพรวมของโปรเจกก่อน ผู้เรียนสามารถเห็นเป้าหมายและผลลัพธ์ของโปรเจกเป็น อย่างไร แล้วแต่ละส่วนประกอบของโปรเจกถูกสร้างอย่างไร เรียนรู้การเขียนสคริปต์ในแต่ละตัว ละครและเวที เพื่อทำให้โปรเจกสมบรูณ์ได้อย่างไร

ทุกโปรเจกที่จะเรียนรู้ต่อไปนี้ มาพร้อมกับโปรแกรม Scratch ทำให้แน่ใจได้ว่า ถ้าผู้เรียนติดตั้ง โปรแกรม Scratch แล้ว จะสามารถเรียนรู้ทุกโปรเจกจากหนังสือเล่มนี้ได้ โปรเจกใน Scratch แบ่งเป็นหมวดหมู่ที่มีเป้าหมายที่ชัดเจน แต่ละหมวดมีหลายโปรเจก เมื่อศึกษาโปรเจกหนึ่งใน แต่ละหมวดแล้ว โปรเจกที่เหลือในหมวดนั้น ก็สามารถเรียนรู้เพิ่มเติมได้เอง เมื่อศึกษา หลากหลายโปรเจกแล้ว จะเห็นศักยภาพโดยรวมของ Scratch ผู้เรียนสามารถเรียนรู้เพิ่มเติมใน เว็บไซต์ Scratch ซึ่งมีโปรเจกมากมายจากผู้ใช้ทั่วโลกได้อีก และควรนำความรู้ที่ศึกษามา ไป ต่อยอด ประยุกต์ใช้ ในการสร้างโปรเจกใหม่อย่างสร้างสรรค์ด้วยตัวเอง

กระบวนการทำงานเพื่อสร้างโปรเจกใน Scratch

การสร้างโปรเจกใน Scratch ก็คล้ายกับการสร้างละครเวทีเรื่องหนึ่ง ซึ่งประกอบไปด้วยเวที สำหรับการแสดง และตัวละครต่างๆ เป็นสำคัญ โดยที่ตัวละครแต่ละตัวก็จะมีบทบาทของมันเอง เมื่อละครเปิดฉากแต่ละตัวละครก็ออกสู่เวที ดำเนินเรื่องราวจากต้นจนจบ

ผู้สร้างโปรเจกก็เปรียบเสมือนผู้สร้างละคร ควบคุมการสร้างทั้งหมด ตั้งแต่เขียนเค้าโครงของ ละคร การดำเนินเรื่อง การจัดเวทีสถานที่ เลือกนักแสดง เขียนบท และกำกับการแสดงบนเวที นอกจากนั้นยังเป็นผู้จัดเสื้อผ้า ฉากเวที ทำเสียงประกอบ และรายละเอียดอื่นๆ อีกมากมาย ดู เหมือนจะยุ่งยากมากมายเกินกว่าเด็กๆ จะทำได้ แต่เชื่อหรือไม่ทีมพัฒนา Scratch ทำเรื่อง ทั้งหมดให้เป็นเรื่องง่าย เป็นขั้นเป็นตอน ท้าทายจินตนาการของพวกเขา

การสร้างโปรเจก เป็นกระบวนการทำซ้ำ ที่เริ่มจากส่วนเล็ก ๆ ในแต่ละตัวละครและเวที ขยาย ใหญ่ขึ้น จนเป็นโปรเจกที่สมบรูณ์ ขณะเขียนโปรแกรม ต้องคอยตรวจสอบความถูกต้องของ โปรแกรมอย่างสม่ำเสมอ เพื่อหาข้อผิดพลาดแล้วทำการแก้ไข เมื่อมีความชำนาญมากขึ้น ผู้เรียนจะเริ่มสังเกตเห็นว่า บางส่วนของโปรแกรมมีความเหมือนหรือคล้ายกัน ทั้งจากตัวละคร เดียวกันหรือต่างตัวละครกัน แทนที่จะเขียนใหม่ตั้งแต่ต้น ผู้เรียนสามารถนำส่วนของโปรแกรม นั้น มาใช้ใหม่ได้อีก หรือนำมาปรับเปลี่ยนให้เข้ากับความต้องการ ทำให้ลดเวลาในการเขียน โปรแกรมได้

โปรเจก Aquarium

การเปิดโปรเจก

เริ่มต้นโปรเจกแรก ด้วยวิธีการเปิดโปรเจกตัวอย่างในโปรแกรม Scratch โดยคลิกที่เมนู File คลิกเลือก Open ... หน้าต่าง Open Project จะปรากฏ พร้อมกับโปรเจกมากมายที่มาตอน ติดตั้งโปรแกรม จะมีปุ่มโฟล์เดอร์อยู่ด้านซ้ายสุด ตรงกลางเป็นโฟล์เดอร์ย่อยหรือไฟล์โปรเจก ต่างๆ ด้านขวามือจะเป็นรายละเอียดของโปรเจกที่เลือก



เพื่อเปิดโปรเจก Aquarium ให้คลิกปุ่มโฟล์เดอร์ Examples (1) จากนั้นจะเห็นโฟล์เดอร์ย่อยใน หมวดต่างๆ ให้เลือกโฟล์เดอร์ Animation ซึ่งจะแสดงโปรเจกต่างๆ ในหมวดนี้ สังเกตซื่อโฟล์ เดอร์ล่าสุด (Animation) ปรากฏอยู่ที่ (2) เมื่อคลิกเลือก Aquarium (3) จะเห็นรายละเอียดของ โปรเจกทางขวามือ (4) เป็นหน้าตาของโปรเจกนี้ (5) เป็นข้อมูลผู้สร้างโปรเจก วิธีใช้งาน แนวคิดในการสร้างโปรเจกนี้ และแนวคิดต่อยอด คลิกปุ่ม OK เพื่อเปิดโปรเจก



ภาพรวมของโปรเจก

โปรเจก Aquarium เป็นการสร้างภาพเคลื่อนไหวหรือแอนิเมชั่นในอควาเรียม การทำงานหลัก ๆ ของปลา ปลาหมึก และแมงกระพรุน คือ การว่ายน้ำไปมาแหวกว่ายผ่านกอสาหร่ายในตู้กระจก แต่เพื่อให้ดูสมจริง เมื่อชนกระจก (กรอบของเวที) พวกมันจะว่ายกลับ และให้ดูเป็นธรรมชาติ มากยิ่งขึ้น พวกมันแต่ตัวจะต้องมีการเปลี่ยนทิศการเคลื่อนที่บ้างบางเวลา โดยที่ทุก ๆ หนึ่ง วินาทีพวกมันมีโอกาสหันกลับทิศ 180 องศา โอกาสการกลับทิศของพวกมันแต่ละตัวมีค่าไม่ เท่ากัน เช่น พวกปลาต่าง ๆ จะมีโอกาสกลับทิศอยู่ 1 ใน 6 และถ้าสังเกตที่ฉากหลังจะเห็น ฟองอากาศลอยขึ้น เนื่องจากการเปลี่ยนแปลงของฉากหลังนั้นเอง

ตัวละครและฉากเวที

โปรเจก Aquarium มีปลาชนิดต่างๆ อยู่ 5 ตัว ปลาหมึก แมงกระพรุน อย่างละ 1 ตัว และ สาหร่าย 3 กอ ฉากหลังเป็นพื้นทรายกับฟองอากาศในน้ำ



ตัวละครใน Aquarium มีอยู่หลายตัว ถ้าสนใจในตัวละครตัวใดตัวหนึ่ง ต้องเลือกมันก่อน เช่น ต้องการเลือกแมงกระพรุน ทำได้โดยการคลิกที่แมงกระพรุนที่อยู่ในรายการตัวละคร หรืออีกวิธี หนึ่งคือการดับเบิ้ลคลิกที่แมงกระพรุนในเวที ตัวละครที่ถูกเลือก (แมงกระพรุน) จะมีกรอบสีฟ้า ล้อมรอบอยู่

ตัวละครที่ถูกเลือก (แมงกระพรุน) จะสามารถเขียนโปรแกรมสคริปต์ได้ในแทป Scripts เพิ่มหรือ ลดชุดละครได้ในแทป Costumes หรือใส่เสียงได้ในแทป Sounds

การเลือกเวทีทำได้คล้ายกับตัวละคร คือ คลิกเวทีในรายการตัวละครกับเวที หรือดับเบิลคลิก พื้นที่ว่างบนเวที เมื่อเลือกเวทีแล้ว จะปรากฏสีฟ้าล้อมรอบเวทีเช่นกัน



เวทีสามารถเขียนโปรแกรมสคริปต์ได้ในแทป Scripts เพิ่มหรือลดฉากหลังเวทีได้ในแทป Backgrounds หรือใส่เสียงได้ในแทป Sounds

สคริปต์ตัวละครและเวที

สคริปต์ของปลา

สคริปต์ของปลาชนิดต่างๆ (Creature2 – Creature6) จะเหมือนกันหมด ดังนั้นดูสคริปต์ของ ปลา Creature5 เป็นตัวแทนของปลาที่เหลือ



สคริปต์แรกทำให้ปลาเคลื่อนที่ไปเรื่อย ๆ ในตู้ปลา และเมื่อชนผนัง (ขอบของเวที) ก็จะเปลี่ยนทิศ ทางการเคลื่อนที่ ด้วยการสะท้อนกลับออกมา สคริปต์เริ่มเมื่อผู้ใช้คลิกธงสีเขียวเริ่มต้นทำงาน สคริปต์ ตามด้วยบล็อกควบคุมตลอดเวลา (forever) เพื่อให้ปลาว่ายน้ำไปเรื่อย ๆ ปลาจะ เคลื่อนที่ (move 1 steps) ไปช้า ๆ (ทีละ 1) และตรวจสอบตัวเองว่า ชนกรอบของเวทีหรือยัง ถ้า ชนมันจะสะท้อนกลับ (if on edge, bounce) ส่วนสคริปต์ที่สองทำให้ปลาดูเป็นธรรมชาติมากขึ้น ด้วยการเปลี่ยนทิศ ว่ายกลับไปกลับมา เหมือนอยู่ในตู้ปลาจริงๆ สคริปต์นี้เริ่มทำงานพร้อมกับสคริปต์แรกเมื่อผู้ใช้คลิกธงเขียว การ เปลี่ยนทิศเกิดขึ้นได้ตลอดเวลา ดังนั้นจึงใช้บล็อกตลอดเวลาควบคุมการทำงาน การเปลี่ยนทิศ ขึ้นอยู่กับเงื่อนไข (บล็อกถ้า if) จากการสุ่มตัวเลข (pick random) ตั้งแต่ 1 ถึง 6 ถ้าเลขสุ่มได้ เท่ากับ 1 เงื่อนไขจะเป็นจริง แสดงว่าปลามีโอกาสในการกลับทิศอยู่ 1 ใน 6 เมื่อเงื่อนไขเป็นจริง ปลาจะหันกลับ 180 องศา (turn 180 degrees) การตรวจสอบเงื่อนไขเพื่อการเปลี่ยนทิศ จะ เกิดขึ้นทุกๆ 1 วินาที โดยใช้บล็อกคอย (wait 1 sec) ในการจับเวลา

สคริปต์ของปลาหมึก

สคริปต์ของปลาหมึก (Creature7) คล้ายกับของปลามาก สคริปต์แรกนั้นเหมือนกับสคริปต์แรก ของปลาเลย ส่วนสคริปต์ที่สองก็แทบจะเหมือนสคริปต์ที่สองของปลาอีกเหมือนกัน เพียงแต่ โอกาสในการกลับทิศอยู่ที่ 1 ใน 3 (pick random 1 to 3 = 1)

| Creature7 |
|--|
| when clicked forever move 1 steps if on edge, bounce when clicked forever |
| if pick random 1 to 3 = 1 |
| turn (7 180) degrees |
| wait 1 secs |

สคริปต์ของแมงกระพรุน



สคริปต์ของแมงกระพรุนมีอยู่ 3 สคริปต์ แต่ก็คล้ายกับสคริปต์ของปลาเช่นกัน สคริปต์แรกเป็น การใช้เอฟเฟคเพื่อทำให้ตัวเองโปร่งแสง โดยการใช้บล็อกเอฟเฟคล่องหน (ghost) ด้วยค่าที่ 35 (set ghost effect to 35)

สคริปต์ที่สองนั้นเหมือนกับสคริปต์แรกของปลาเลย ส่วนสคริปต์ที่สามก็คล้ายกับสคริปต์ที่สอง ของปลาอีกเหมือนกัน เพียงแต่โอกาสในการกลับทิศอยู่ที่ 1 ใน 10 (pick random 1 to 10 = 1)

สคริปต์ของสาหร่าย



หน้าที่ของสาหร่าย คือ ทำให้ตู้ปลาดูมีมิติลึกเข้าไป ด้วยการกำหนดตำแหน่งในแนวลึกของตู้ ปลา (เลเยอร์ Layer) ให้กับสาหร่ายแต่ละกอ เพื่อทำให้ดูเหมือนว่าสาหร่ายอยู่ลึกคนละตำแหน่ง ในตู้ปลา สคริปต์ของสาหร่ายแต่ละกอเริ่มตันเมื่อผู้ใช้คลิกธงเขียว สาหร่าย plant1 จะเคลื่อนไป ที่เลเยอร์หน้าสุดก่อน (go to front) จากนั้นย้ายถอยหลังไป 2 เลเยอร์ (go back 2 layers) สาหร่าย plant2 ทำคล้ายกัน แต่ถอยหลังไป 4 เลเยอร์ ส่วนสาหร่าย plant3 ก็ทำเช่นกัน แต่ ถอยหลังด้วยการสุ่มค่าระหว่าง 1 ถึง 7

เลเยอร์ใน Scratch เป็นการกำหนดตำแหน่งความลึกแบบง่าย ๆ ไม่ซับซ้อนอะไร เลเยอร์ที่มี ตัวเลขมาก ก็จะอยู่ลึกเข้าไปด้านในของจอคอมพิวเตอร์ การกำหนดเลเยอร์ทำให้ตัวละครเสมือน อยู่ในโลกสามมิติ ตัวละครจะมีการเหลื่อมล้ำกัน ตัวละครที่อยู่เลเยอร์ด้านหน้า (ตัวเลขน้อย) ก็ จะบังตัวละครที่อยู่เลเยอร์ลึกลงไป (ตัวเลขมาก)

สคริปต์ของเวที

สคริปต์เวทีส่วนใหญ่มักจะเกี่ยวกับฉากหลังของเวที (Backgrounds) สคริปต์ของเวทีในโปรเจก Aquarium ถูกเขียนขึ้นเพื่อให้เห็นการเคลื่อนไหวของฟองอากาศตลอดเวลา โดยการเปลี่ยนฉาก หลังของเวที



จากสคริปต์จะเห็นว่า เมื่อผู้ใช้คลิกธงเขียวแล้ว ฉากหลังจะเปลี่ยนเป็นภาพถัดไปที่ละภาพ (next background) สลับวนกันไป (ใช้บล็อกตลอดเวลา forever) ตลอดทุกๆ 1 วินาที (จับเวลาด้วย บล็อก wait) จนกว่าโปรแกรมจะหยุด การเปลี่ยนแปลงของฉากเป็นการแสดงแอนิเมชั่นง่ายๆ เพื่อทำให้เห็นฟองอากาศลอยขึ้นตลอดเวลา

บล็อก next background ใช้สำหรับเปลี่ยนภาพฉากหลังตามลำดับของภาพ เช่น ในกรณีนี้ มี ฉากหลัง 3 ภาพ และป[ั]จจุบันแสดงภาพที่ 1 ทุกครั้งที่ใช้บล็อก next background ภาพที่จะ แสดงถัดไปคือภาพที่ 2 ภาพถัดไปคือภาพที่ 3 จากนั้นจะวนกลับไปภาพที่ 1 อีกครั้ง ลำดับภาพ จะเป็นเช่นนี้ไปเรื่อย ๆ เมื่อยังใช้คำสั่งนี้อยู่

แอนิเมชั่นจากการเปลี่ยนภาพฉากหลัง



ฉากเวทีสามารถทำแอนิเมชั่นแบบง่าย ๆ ด้วยการเปลี่ยนภาพฉากหลัง เช่นในกรณีนี้ มีภาพฉาก หลังอยู่ 3 ภาพ ที่แตกกันในส่วนของฟองอากาศ เมื่อเปลี่ยนภาพฉากหลังทีละภาพ ต่อเนื่องกัน ไป จะเห็นฟองอากาศลอยขึ้นตลอดเวลา

เนื่องจากมีภาพที่ใช้แสดงแอนิเมชั่นอย่างจำกัด เพื่อแลกกับการใช้ข้อมูลที่น้องลง ทำให้การ เปลี่ยนแปลงภาพเกิดขึ้นรวดเร็วเกินไป จนดูไม่ทัน ดังนั้นต้องหน่วงเวลาการเปลี่ยนแปลงภาพ ด้วยบล็อก wait ซึ่งเป็นคำสั่งหยุดทำงานชั่วคราว ตามระยะเวลาที่กำหนด

การต่อยอดโปรเจก Aquarium

การศึกษาโปรเจกที่มีอยู่แล้ว ทำให้เราได้เรียนรู้การเขียนโปรแกรม จากมุมมองของคนอื่น ได้ฝึก อ่านโปรแกรม และแนวคิดเบื้องหลังของโปรเจกนั้นๆ วิธีนี้ยังช่วยให้มือใหม่หัดเขียนโปรแกรม เริ่มต้นหัดเขียนโปรแกรมด้วยการต่อยอดจากโปรเจกเดิม โดยไม่ต้องเริ่มจากศูนย์ คำแนะนำ ต่อไปนี้เป็นตัวอย่าง เพื่อดูว่าจะต่อยอดโปรเจกนี้ได้อย่างไรบ้าง

- สังเกตว่า เหล่าปลา ปลาหมึก และแมงกระพรุน ต่างเคลื่อนที่ไม่หยุดเลย ทำอย่างไรให้ มันหยุดบ้าง เคลื่อนที่บ้าง
- พวกมันเคลื่อนที่ด้วยความเร็วเท่ากันหมด ดูแล้วง่วงนอน ทำอย่างไรให้มันเคลื่อนที่ด้วย ความเร็วต่างกัน หรือแต่ละตัวสามารถใช้ความเร็วที่ต่างกันได้
- เพิ่มความหลากหลายในการเคลื่อนที่ของปลา แทนที่จะใช้การสุ่มเพื่อเปลี่ยนทิศ ลองให้ พวกมันเคลื่อนที่ไปพร้อม ๆ กับการเปลี่ยนทิศ โดยการใช้บล็อกเปลี่ยนทิศทางคู่กับ บล็อกเคลื่อนที่ ทดลองเปลี่ยนค่าต่าง ๆ ดูว่าค่าไหนที่ทำให้ปลาเคลื่อนที่เป็นธรรมชาติ มากขึ้น

จะเห็นว่าการปรับเปลี่ยนโปรแกรมใน Scratch นั้นทำได้ง่าย และที่สำคัญเห็นผลได้แทบทันที ทำให้การเรียนเขียนโปรแกรมใน Scratch ไม่ยากเหมือนภาษาอื่น

โปรเจก MadLibs

โปรเจก MadLibs เป็นการเล่าเรื่อง (Story telling) ที่มีการโต้ตอบกับผู้ใช้ ก่อนจะเข้าสู่การเล่า เรื่อง เริ่มต้นเปิดโปรเจก MadLibs ด้วยการคลิกเมนู File เลือก Open จากนั้นคลิกปุ่ม Examples เลือกโฟล์เดอร์ Stories เลือกโปรเจก MadLib

ภาพรวมโปรเจก

MadLibs เป็นการเล่าเรื่อง ที่มีผู้หญิงเป็นผู้ดำเนินเรื่อง เริ่มต้นด้วยการถามตอบกับผู้ใช้ จากนั้น ข้อมูลคำตอบจากผู้ใช้จะถูกส่งต่อให้กับวาพ 2 ตัว นำออกมาเล่าเรื่องตลกผูกกับคำตอบที่ได้มา วาพทั้งสองตัวมีการส่งต่อบทบาทและบทพูดระหว่างกัน พร้อมกับแสดงแอนิแมชั่น เสียง ประกอบ และเอฟเฟคบนฉากหลัง เพื่อเพิ่มสีสรรให้กับการเล่าเรื่อง



ตัวละครและฉากเวที

ตัวละครมีเพียงผู้หญิงเป็นผู้ดำเนินเรื่อง วาพตัวใหญ่และวาพตัวเล็กเป็นผู้เล่าเรื่อง และฉากเวที เป็นวิวทิวทัศน์

สคริปต์ตัวละครและเวที

สคริปต์ของผู้ดำเนินเรื่อง

สคริปต์ของผู้หญิงซึ่งเป็นผู้ดำเนินเรื่อง เริ่มต้นด้วยการพูดเกริ่นก่อนจะถามตอบกับผู้ใช้ คำตอบ จากผู้ใช้จะถูกเก็บไว้ แล้วส่งต่อให้วาพ 2 ตัวเป็นผู้เล่าเรื่องต่อไป



สคริปต์เริ่มเมื่อผู้ใช้คลิกธงสีเขียว เป็นการเริ่มต้นการเล่าเรื่อง โดยผู้ดำเนินเรื่องได้พูดเกริ่นสาม ประโยค ๆ ละ 2 วินาที ด้วยบล็อกพูด (say) จากนั้นเป็นการถามตอบกับผู้ใช้ โดยถามผู้ใช้ด้วยบล็อกถาม (ask) แล้วรอคำตอบจากผู้ใช้ เมื่อ ผู้ใช้พิมพ์คำตอบและกดคีย์ Enter แล้ว คำตอบจะถูกเก็บไว้ที่ตัวแปรภายในชื่อ answer

เนื่องจากทุกครั้งที่มีการถามตอบ คำตอบใน answer จะถูกแทนที่ด้วยคำตอบใหม่ล่าสุดเสมอ เพราะฉนั้นถ้าต้องการนำแต่ละคำตอบมาใช้ภายหลัง ต้องสร้างตัวแปรใหม่ มาเก็บค่าคำตอบนั้น แทน

โปรเจกนี้สร้างตัวแปรมา 4 ตัว คือ action name thing1 และ thing2 เพื่อเก็บแต่ละคำตอบของ ผู้ใช้ เมื่อได้คำตอบทั้งหมดแล้ว ก็จะเริ่มการเล่าเรื่อง ด้วยการส่งสาร start story ให้วาพ 2 ตัว ออกมาเล่าเรื่องผูกกับคำตอบที่ได้รับมาในตัวแปรทั้งสี่

การถามตอบกับผู้ใช้

การโต้ตอบระหว่างผู้ใช้กับโปรแกรม มีหลายวิธี เช่น การคลิกเมาส์ การกดคีย์บอร์ด การใช้เสียง เป็นต้น ในโปรเจกนี้ผู้ดำเนินเรื่องมีการถามตอบกับผู้ใช้ คำถามจะปรากฏบนเวที ผู้ใช้ตอบผ่าน คีย์บอร์ด โปรแกรมจะรอรับคำตอบ และรอจนกว่าผู้ใช้จะกดคีย์ Enter หรือคลิกเครื่องหมายถูก คำตอบที่ได้จะถูกเก็บไว้ในตัวแปร

ต้องการเก็บค่าในตัวแปร answer ในตัวแปร name โดยใช้บล็อก ^{set <u>name</u> to <u>answer</u> ตัว แปร name เป็นตัวแปรหนึ่งในรายการตัวแปรที่สร้างโดยผู้ใช้}

ความแตกต่างในการถามตอบระหว่างตัวละครกับเวที

ถ้าคำถามมาจากตัวละคร คำถามจะถูกพูดจากตัวละคร และจะปรากฏกรอบให้ผู้ใช้พิมพ์เพื่อ ตอบคำถาม ด้านล่างของเวที



แต่ถ้าคำถามมาจากเวที คำถามจะปรากฏในกรอบการถามตอบ ด้านล่างของเวที



การสร้างตัวแปร

96

้วิธีสร้างตัวแปรนั้นให้เลือกบล็อกหมวดตัวแปร จากนั้นคลิกปุ่มสร้างตัวแปร (Make a variable)



คำถามที่ผู้ดำเนินเรื่องถาม ต้องการ 4 คำตอบ ดังนั้นต้องสร้างตัวแปรมา 4 ตัว

ตัวแปรต้องมีชื่อ เพื่อเรียกใช้ภายหลัง ชื่อตัวแปรทั้ง 4 นั้น คือ action name thing1 และ thing2 เมื่อสร้างตัวแปรแล้ว จะเกิดบล็อกใหม่สำหรับตัวแปรทั้ง 4 ในบล็อกหมวดตัวแปร โดยด้านหน้า ของบล็อกเหล่านี้ จะมีกล่องสี่เหลี่ยมที่สามารถคลิกเลือกได้ ถ้าคลิกเลือกจะมีเครื่องหมายถูก ปรากฏ และจะทำให้ตัวแปรนี้แสดงค่าป[ั]จจุบันของมันบนเวที

การกำหนดค่าให้ตัวแปร ให้ใช้บล็อกตั้งค่า (set) โดยเลือกชื่อตัวแปรที่ต้องการก่อน (จาก รายการตัวแปรทั้งหมดที่ถูกสร้างขึ้น) จากนั้นกำหนดค่าให้มัน ซึ่งค่านั้นอาจเป็นตัวเลขหรือ ตัวอักษร ที่พิมพ์เข้าไปตรงๆ ในช่องว่าง หรือใช้ค่าที่อยู่ในบล็อกตัวแปรก็ได้ เช่น บล็อกตัวแปร answer

การแชร์ข้อมูล

ขณะที่สร้างตัวแปร ยังสามารถกำหนดตัวแปรให้เป็นตัวแปรส่วนตัวได้ โดยคลิกเลือก For this sprite only เพื่อใช้ในตัวละครตัวที่สร้างตัวแปรนี้เท่านั้น หรือกำหนดตัวแปรให้เป็นตัวแปร ส่วนรวมได้ โดยคลิกเลือก For all sprites เพื่อให้ทุกตัวละคร รวมทั้งเวที สามารถใช้ตัวแปรนี้ได้

| | ? | |
|-----------------|------------------------|--|
| Variable name? | | |
| name | | |
| For all sprites | ○ For this sprite only | |
| ОК | Cancel | |

ถ้าให้ทุกตัวละครและเวทีใช้ตัวแปรได้ ตัวแปรที่สร้างขึ้นใหม่ จะปรากฏอยู่ในบล็อกหมวดตัวแปร ของทุกตัวละคร รวมทั้งเวทีด้วย วิธีนี้เป็นการแชร์ข้อมูลระหว่างตัวละครต่าง ๆ และเวที เหมือน ในโปรเจกนี้ ตัวแปรที่ใช้เก็บคำตอบนั้น เป็นตัวแปรส่วนรวมที่ผู้ดำเนินเรื่องสร้างขึ้น แล้ววาพ สองตัวสามารถเอาไปใช้ได้ขณะเล่าเรื่อง

สคริปต์ของวาฬตัวใหญ่และวาฬตัวเล็ก

วาพตัวใหญ่และวาพตัวเล็ก จะเล่าเรื่องสลับกันไปมา จากคำตอบที่ได้รับจากผู้ดำเนินเรื่อง เพื่อ เพิ่มสีสันในการเล่าเรื่อง วาพทั้งสองจะแสดงแอนิเมชั่นไปด้วย

เพื่อให้เห็นภาพการส่งต่อการทำงานระหว่างวาพทั้งสองตัว จึงมาดูสคริปต์ของวาพทั้งสองตัวไป พร้อม ๆ กัน



เมื่อผู้ใช้คลิกธงเขียว วาพทั้งสองตัวจะซ่อนตัวด้วยบล็อกซ่อน (hide) จากนั้นเมื่อวาพทั้งสองตัว ใด้รับสาร start story จากบล็อกรับสาร (when I receive start story) วาพตัวใหญ่จะเปลี่ยนชุด เป็น whale basic วาพทั้งสองตัวจะถูกกำหนดวางไว้นอกเวทีด้วยบล็อกไปยังตำแหน่ง (x, y) สังเกตตำแหน่งของวาพทั้งสองตัว ว่าเกินขอบเวทีแล้ว (กรอบเวทีมีขนาดที่ -240 <= x <= 240, -180 <= y <= 180) จากนั้นวาพทั้งสองตัวก็เปิดตัวออกมา (show) พร้อมกับเคลื่อนที่ไปยัง ตำแหน่งที่กำหนดไว้ภายใน 1 วินาทีพร้อมกัน (glide 1 secs to x: y:) ก่อนที่วาพตัวใหญ่จะส่ง สาร story 1 ออกมา เพื่อเริ่มตันเล่าเรื่อง

การสนทนาระหว่างวาพทั้งสองตัวจะสลับกันพูด โดยเมื่อวาพตัวหนึ่งพูดจบ ก็จะส่งสารบอกวาพ อีกตัวหนึ่งให้พูดต่อ โดยเริ่มจากวาพตัวใหญ่ก่อน เพราะเป็นผู้รับสาร story 1 จากการส่งสารถึง ดัวเอง



เมื่อวาพตัวใหญ่ได้รับสาร story 1 จะเริ่มสคริปต์ด้วยการเปลี่ยนชุด (switch to costume) เป็น whale happy1 จากนั้นพูดหนึ่งประโยคก่อนจะแสดงแอนิเมชั่นดีใจ แล้วพูดต่อ โดยนำข้อมูลใน thing1 มารวมในคำพูดด้วย โดยใช้บล็อกเชื่อมคำ (join) แล้วส่งสาร story 2 เพื่อให้วาพตัวเล็ก ได้พูดต่อ



เมื่อวาพตัวเล็กได้รับสาร story 2 ก็จะพูดตอบวาพตัวใหญ่ โดยนำข้อมูลใน action มารวมใน คำพูดด้วย โดยใช้บล็อกเชื่อมคำ (join) แล้วส่งสาร story 3 เพื่อให้วาพตัวใหญ่ได้พูดต่อ เนื่องจากข้อมูลใน action อยู่ระหว่างประโยค จึงต้องเชื่อมคำสองครั้ง โดยใช้บล็อกเชื่อม (join) ซ้อนกัน

แอนิเมชั่นจากการเปลี่ยนชุดตัวละคร

ระหว่างการสนทนาของวาพทั้งสองตัว วาพตัวใหญ่แสดงแอนิเมชั่นดีใจ โดยใช้เทคนิคการ เปลี่ยนชุด (switch to costume) สลับไปมาระหว่างชุด whale happy2 กับชุด whale happy1 จำนวน 3 ครั้งภายในบล็อกควบคุมทำซ้ำ (repeat)

สังเกตว่าในแอนิเมชั่นจะมีบล็อกคอย (wait) หลายแห่ง ทั้งนี้เพราะแอนิเมชั่นจะเกิดขึ้นเร็วไป มองไม่ทัน จึงต้องหน่วงเวลาด้วยบล็อกคอยในแต่ละครั้งที่เปลี่ยนชุด

| te what scripts Costu | le 11 5 y: -128 direction: 90 mes Sounds |
|--------------------------|---|
| New costume: Paint | Import Camera |
| | 153x77 2 KB Edit Copy X |
| 2 | whale happy1 147x77 2 KB Edit Copy X |
| ° | whale happy2 147x96 2 KB Edit Copy X |
| 4 | whale mad 1 150x77 2 KB Edit Copy X |
| 5 | whale mad2 150x120 3 KB Edit Copy X |
| 6 | whale mad3 150x106 2 KB Edit Copy X |

วาพตัวใหญ่ยังสามารถแสดงแอนิเมชั่นได้อีกหลายอารมณ์ (เฉย โกธร) ทำให้มีหลายชุดมาก เพื่อแสดงในแต่ละอารมณ์ ดังนั้นการตั้งชื่อชุดให้สอดคล้องกับอารมณ์ จะทำให้ง่ายต่อการ นำมาใช้ทำแอนิเมชั่น



เมื่อวาพตัวใหญ่ได้รับสาร story 3 จะพูดตอบวาพตัวเล็ก แล้วเปลี่ยนชุดเป็น whale basic จากนั้นแสดงแอนิเมชั่นโกธร โดยการเปลี่ยนชุดระหว่าง whale mad1 กับ whale mad2 กับ whale mad3 สลับกับการคอย 0.2 วินาที ทั้งหมด 3 ครั้งภายในบล็อกควบคุมทำซ้ำ (repeat) จากนั้นเปลี่ยนชุดเป็น whale basic อีกครั้ง แล้วพูดต่อโดยนำข้อมูลใน thing2 มารวมในคำพูด ด้วย ก่อนส่งสาร story 4 เพื่อให้วาพตัวเล็กได้พูดต่อ



เมื่อวาพตัวเล็กได้รับสาร story 4 ก็จะพูดตอบวาพตัวใหญ่หนึ่งประโยค แล้วแสดงแอนิเมชั่นด้วย วิธีหมุนตัวมันเอง โดยใช้บล็อกเลี้ยว (turn) ทีละ 36 องศา สลับกับการคอย 0.01 วินาที ทำซ้ำ 10 รอบภายในบล็อกควบคุมทำซ้ำ (repeat) ก็จะหมุนครบ 360 องศา 1 รอบพอดี จากนั้นพูด ต่อโดยนำข้อมูลใน name มารวมในคำพูดด้วย โดยใช้บล็อกเชื่อมคำ (join) ซ้อนกัน ก่อนจะพูด ปิดประโยคสุดท้ายของการเล่าเรื่อง

การส่งและรับสาร

จะทำอย่างไรให้ตัวละครหรือเวที สื่อสารโต้ตอบกัน รู้จังหวะเวลาที่ตัวเองจะต้องพูดต้องแสดง ใน Scratch การส่งต่อการทำงานระหว่างตัวละคร ใช้วิธีการส่งสารออกไป โดยอาศัยชื่อของสารที่ แตกต่างกัน เป็นตัวส่งไม้ให้ระหว่างกัน เมื่อตัวละครหรือเวทีได้รับสารที่สนใจ ก็จะรู้ว่า ได้เวลาที่ จะต้องเริ่มทำงานต่อแล้ว

การส่งสารจะใช้บล็อกส่งกระจาย (broadcast) พร้อมกับเลือกสารที่มีอยู่ในรายการ ถ้าต้องการ สร้างสารใหม่ ทำได้ง่ายๆ โดยคลิกสามเหลี่ยมชี้ลงในบล็อกส่งกระจาย คลิก new... แล้วใส่ชื่อ สารลงไป เช่น ต้องการสร้างสารใหม่ชื่อ story 5

| proadcast start story ? | start story story 1 story 2 story 2 |
|----------------------------|--|
| Message name: | story 4 |
| story 5 | new |
| OK Cancel | |

ตัวละครทุกตัวรวมทั้งเวทีจะได้รับสารที่ส่งออกมาทุกสาร ไม่ว่าใครจะเป็นคนส่ง แต่เลือก ตอบสนองกับสารที่ตัวเองสนใจเท่านั้น ด้วยการใช้บล็อก ^(when I receive) เริ่มต้นสคริปต์ใหม่ โดยสามารถเลือกสารที่สนใจได้จากรายการ และเมื่อตัวละครหรือเวทีได้รับสารที่สนใจ ก็จะ ตอบสนองด้วยการเริ่มทำงานในสคริปต์นั้น

สคริปต์ของเวที

ฉากเวทีในโปรเจกนี้น่าสนใจ เพราะมีการใส่ลูกเล่นทั้งภาพและเสียง โดยใส่เอฟเฟคให้ภาพฉาก หลังของเวที ด้วยบล็อกตั้งค่าเอฟเฟค (set effect to) ซึ่งมีเอฟเฟคหลายแบบ แต่ในกรณีนี้ใช่ค่า ความสว่าง (brightness)



เมื่อผู้ใช้คลิกธงเขียว เอฟเฟคทุกอย่างจะถูกกำจัดไปก่อน โดยบล็อกล้างเอฟเฟค (clear graphic effects) เพื่อตั้งค่าเอฟเฟคทุกอย่างไปที่ค่าเริ่มต้น (ค่าเอฟเฟคมีค่าระหว่าง -100 ถึง 100 โดยที่ ค่า 0 คือค่าดั้งเดิมก่อนใส่เอฟเฟค) จากนั้นตั้งค่าความสว่างที่ -60 (set brightness effect to -60) ซึ่งจะทำให้ภาพฉากหลังมืดลง

เมื่อได้รับสาร start story ใช้บล็อกเปลี่ยนค่าเอฟเฟคความสว่าง (change effect by) ในบล็อก ควบคุมทำซ้ำ (repeat) เพื่อเปลี่ยนค่าความสว่างที่ละ 6 ทำซ้ำ 10 ครั้ง ให้ได้ค่าความสว่างรวม 60 วิธีนี้จะทำให้ภาพฉากหลังค่อย ๆ สว่างขึ้นจนกลับไปยังภาพฉากหลังดั้งเดิม

ในขณะเดียวกัน เมื่อได้รับสาร start story ก็เปิดเสียงประกอบ madlibs ขึ้นมา โดยใช้บล็อกเล่น เสียงจนจบ (play sound madlibs until done)

การต่อยอดโปรเจก MadLibs

- ฝึกการใช้ตัวแปรลิสต์ เก็บข้อมูลการถามตอบระหว่างผู้ดำเนินเรื่องกับผู้ใช้ แทนการเก็บ ข้อมูลแยกจากกันในหลายๆ ตัวแปร
- นอกจากการใช้ตัวหนังสือในการเล่าเรื่องแล้ว Scratch สามารถอัดเสียงผ่านไมโครโฟน แล้วนำมาเปิดได้ ดังนั้นการใช้เสียงพูดในการเล่าเรื่องเพิ่มเติม จะทำให้เรื่องน่าสนใจ มากขึ้น
- เพื่อใช้สื่อที่ Scratch มีให้อีกอย่าง คือการถ่ายภาพผู้ใช้จากกล้อง การเล่าเรื่องน่าจะ สนุกมากยิ่งขึ้น ถ้าบอกให้ผู้ใช้ถ่ายรูปตัวเองจากกล้องในคอมพิวเตอร์ จากนั้นเล่าเรื่อง ผูกกับรูป

สคริปต์ด้านล่างนี้ เป็นตัวอย่างสคริปต์ถามผู้ใช้ให้ถ่ายรูปของตัวเอง โดยเริ่มจากซ่อน ตัวเองก่อน (hide) เมื่อถูกถามให้ถ่ายรูป ให้คลิกที่แทป Costumes คลิกปุ่ม Camera คลิกปุ่มกล่องถ่ายรูปเพื่อถ่ายรูป คลิกปุ่ม Done จบการถ่ายรูป จากนั้นคลิกปุ่ม เครื่องหมายถูกบนเวที เพื่อรันสคริปต์ต่อ รูปที่พึ่งถ่ายไปจะเป็นชุดตัวละครปัจจุบันโดย อัตโนมัติ โดยปกติรูปที่ถ่ายจะมีขนาด 320 x 240 อาจใหญ่เกินไป ให้ใช้บล็อกกำหนด ขนาดให้เล็กลงได้ (set size to 30 %) จากนั้นจึงแสดงรูปออกมา (show)



โปรเจก SimpleCircuit

โปรเจก SimpleCircuit เป็นการจำลอง (simulation) แผงวงจรไฟฟ้าแบบง่ายๆ เริ่มด้วยการเปิด โปรเจก SimpleCircuit โดยเลือก Open ในเมนู File จากนั้นคลิกปุ่ม Examples เลือกโฟล์เดอร์ Simulations เลือกโปรเจก SimpleCircuit

ภาพรวมของโปรเจก

SimpleCircuit เป็นการจำลองแผงวงจรอนุกรมอย่างง่าย โดยต่อหลอดไฟ แบตเตอรี และสวิตช์ เข้าด้วยกัน ถ้าแผงวงจรต่อโดยสมบรูณ์แล้ว เมื่อเอาเมาส์วางเหนือสวิตช์ หลอดไฟก็จะสว่างขึ้น นอกจากภาพของอุปกรณ์ต่าง ๆ การจำลองนี้ยังสามารถแสดงรูปสัญลักษณ์ของลายวงจรไฟฟ้า และอุปกรณ์แต่ละตัว



ตัวละครและเวที

SimpleCircuit มีตัวละครที่เป็นอุปกรณ์เพียง 3 ตัว คือ สวิตช์ แบตเตอรี หลอดไฟ พร้อมกับ คำสั่งบอกวิธีใช้งาน บนฉากหลังเวทีที่เป็นแผงวงจรไฟฟ้าแบบอนุกรม

สคริปต์ตัวละครและเวที

สคริปต์ของใบคำสั่ง

ใบคำสั่งแสดงวิธีการใช้ดังนี้ คลิกธงเขียวเริ่มต้นการจำลอง ลากอุปกรณ์แต่ละตัวเชื่อมกับ แผงวงจร เปิดหลอดไฟโดยใช้เมาส์วางบนสวิตช์ กด space bar เพื่อดูแบบสัญลักษณ์ของ แผงวงจร



สคริปต์ของใบคำสั่ง เริ่มเมื่อผู้ใช้คลิกธงเขียว ก็จะแสดงวิธีใช้งานก่อนเริ่มการจำลอง จากนั้น วิธีใช้งานจะหายไป โดยใช้เอฟเฟคล่องหน (set ghost effect to 100)

วิธีใช้งาน

เมื่อผู้ใช้คลิกธงเขียว อุปกรณ์ทั้งหมดจะมาอยู่ตรงกลาง เพื่อให้ผู้ใช้นำอุปกรณ์ไปวางในแผงวงจร เอง และเมื่อกด space bar จะเห็นสัญลักษณ์สีดำของลายวงจรและอุปกรณ์ ทั้งบนเวทีและใน รายการตัวละคร



การวางอุปกรณ์ไม่ขึ้นกับตำแหน่งเพราะเป็นวงจรอนุกรมอย่างง่าย ขอเพียงให้อุปกรณ์แตะขั้ว แผงวงจรเป็นใช้ได้ แต่เพื่อความสวยงามการวางให้ขั้วของอุปกรณ์กับแผงวงจรตรงกันจะดีมาก รูปถัดไปแสดง (1) การวางที่ถูกต้องสวยงาม (2) เมื่อเอาเมาส์ชี้ที่สวิตช์ หลอดไฟจะสว่างทันที (3) เมื่อกด space bar จะเห็นสัญลักษณ์ทางไฟฟ้าสีดำ


แต่ถ้าวางอุปกรณ์แค่ให้ส่วนหนึ่งส่วนใดแตะขั้วแผงวงจรแล้ว การจำลองยังคงถูกต้องอยู่ แต่ขัด ความรู้สึกไปหน่อย ทั้งนี้เกิดจากการเขียนโปรแกรม รูปถัดไปแสดงให้เห็นว่า (1) แม้ขั้วจะถูก วางไม่ตรงตำแหน่ง (2) หลอดไฟก็ยังทำงานได้ (3) เมื่อกด space bar จะเห็นสัญลักษณ์ทาง ไฟฟ้าที่อยู่ผิดตำแหน่ง



สคริปต์ของเวที

ฉากหลังเวทีเป็นแผงวงจรอนุกรมอย่างง่าย มีขั้วต่ออุปกรณ์ 3 แห่ง ตัวขั้วมีสีเหลืองกับสีฟ้าเพื่อ ใช้ในการเชื่อมต่อกับอุปกรณ์ แผงวงจรมี 2 รูปแบบ แบบปกติที่มีสีสันสวยงาม กับแบบ สัญลักษณ์ทางไฟฟ้าที่เป็นสีดำ



สคริปต์ของแผงวงจรเริ่มเมื่อผู้ใช้คลิกธงเขียว งานหลักของแผงวงจรทำแค่สลับรูปแบบ เมื่อผู้ใช้ กดคีย์ space bar โดยใช้บล็อกควบคุมถ้ามิฉะนั้น (if-else) ในการกำหนดการแสดงรูปแบบ โดย ตรวจสอบว่าผู้ใช้กด space bar หรือยัง (key space pressed?) ถ้าใช่ ให้แสดงภาพฉากหลัง เป็นรูปแบบสัญลักษณ์ (Symbols) มิฉะนั้น (else) แสดงรูปแบบสีสันปกติ (Blank Circuit)

แต่เนื่องจากผู้ใช้จะกดคีย์ space bar เมื่อไหรก็ได้ ดังนั้นการสลับรูปแบบที่กล่าวมา ต้องอยู่ใน บล็อกควบคุมตลอดเวลา (forever) เพื่อตรวจสอบการกดคีย์ตลอดเวลา

สคริปต์ของแบตเตอรี



แบตเตอรีมีสองชุด ชุดแรกเป็นรูปแบตเตอรีเหมือนจริง ชุดที่สองเป็นรูปสัญลักษณ์ทางไฟฟ้า ของแบตเตอรี สังเกตแม่กุญแจที่ต่อจากชื่อ Battery จะเปิดอยู่ หมายความว่า แบตเตอรีสามารถ ใช้เมาส์ลากไปมาได้ขณะโปรแกรมทำงานในโหมดนำเสนอ หรือบนเว็บไซต์ของ Scratch



สคริปต์แรกของแบตเตอรี คล้ายกับแผงวงจร คือทุกครั้งที่ผู้ใช้กด space bar ให้แสดงรูป สัญลักษณ์ของแบตเตอรี มิฉะนั้นแสดงรูปจริงของแบตเตอรี โดยใช้บล็อกควบคุมถ้ามิฉะนั้น (ifelse) ในการกำหนดการแสดงรูปแบบ และอยู่ภายในบล็อกควบคุมวนซ้ำตลอดเวลา (forever) เพื่อรับการตรวจสอบตลอดเวลา



สคริปต์ที่สองของแบตเตอรี เป็นการตรวจสอบว่าแบตเตอรีต่อกับแผงวงจรแล้วหรือยัง สคริปต์ เริ่มเมื่อผู้ใช้คลิกธงเขียว จากนั้นกำหนดขนาดเริ่มต้นของแบตเตอรีเท่าของจริงก่อน ด้วยบล็อก ตั้งค่าขนาด (set size to 100%) ทั้งนี้เพราะขนาดมีผลต่อการเชื่อมเข้าแผงวงจร

จากนั้นแบตเตอรีจะมาอยู่ตรงกลางแผงวงจร (go to x:-4 y:-35) เพื่อให้ผู้ใช้นำมันไปวางใน แผงวงจรเอง วิธีการหาตำแหน่งบนเวทีเพื่อใช้ในสคริปต์ทำได้โดยลากอุปกรณ์ เช่นแบตเตอรี ไปยังตำแหน่งที่ต้องการก่อน จากนั้นสังเกตตำแหน่ง x กับ y ในบริเวณข้อมูลของแบตเตอรี เอา ค่า x กับ y ที่ปรากฏ ไปใช้ในบล็อกไปยัง (go to x: y:)

ถัดไปเป็นการตรวจสอบการเชื่อมต่อแบตเตอรีกับแผงวงจร ตลอดการใช้งาน ดังนั้นการ ตรวจสอบจะอยู่ภายในบล็อกควบคุมตลอดเวลา (forever) ในการตรวจสอบนั้นใช้บล็อกถ้า มิฉะนั้น (if-else) โดยตรวจดูว่าแบตเตอรีสัมผัสกับสีเหลืองและสีฟ้า (ซึ่งเป็นสีของขั้วใน แผงวงจร) หรือยัง ถ้าใช่ ให้ตั้งค่าตัวแปร battconnected เท่ากับ 1 (ต่อกับแผงวงจรแล้ว) ถ้า ยังให้ตั้งค่าเป็น 0 (ยังไม่ได้ต่อกับแผงวงจร)

สคริปต์ของหลอดไฟ

หลอดไฟมีสามชุด ชุดแรกเป็นรูปหลอดไฟติด ชุดที่สองเป็นรูปหลอดไฟดับ รูปที่สามเป็นรูป สัญลักษณ์ทางไฟฟ้าของหลอดไฟ



สังเกตแม่กุญแจที่อยูถัดจากชื่อ Bulb จะเปิดอยู่ หมายความว่า หลอดไฟสามารถใช้เมาส์ลากไป มาได้ขณะโปรแกรมทำงานในโหมดนำเสนอ หรือบนเว็บไซต์ของ Scratch



สคริปต์แรกของหลอดไฟ เป็นการตรวจสอบว่าหลอดไฟต่อกับแผงวงจรแล้วหรือยัง คล้ายๆ กับ แบตเตอรี สคริปต์เริ่มต้นเมื่อผู้ใช้คลิกธงเขียว จากนั้นกำหนดขนาดเริ่มต้นของหลอดไฟเท่าของ จริงก่อน ด้วยบล็อกตั้งค่าขนาด (set size to 100%) ทั้งนี้เพราะขนาดมีผลต่อการเชื่อมเข้า แผงวงจร

ถัดไปเป็นการตรวจสอบการเชื่อมต่อหลอดไฟกับแผงวงจร ภายใต้การควบคุมของบล็อก ตลอดเวลา (forever) การตรวจสอบจะใช้บล็อกถ้ามิฉะนั้น (if-else) โดยตรวจดูว่าหลอดไฟ สัมผัสกับสีเหลืองและสีฟ้า (ซึ่งเป็นสีของขั้วในแผงวงจร) หรือยัง ถ้าใช่ ให้ตั้งค่าตัวแปร balbconnected เท่ากับ 1 (ต่อกับแผงวงจรแล้ว) ถ้ายังให้ตั้งค่าเป็น 0 (ยังไม่ได้ต่อกับแผงวงจร) สคริปต์ที่สอง จะคอยตรวจสอบการกด space bar และการเปิดสวิตช์จากผู้ใช้ โดยเริ่มสคริปต์ เมื่อผู้ใช้คลิกธงเขียว จากนั้นกำหนดขนาดเริ่มต้นของหลอดไฟเท่าของจริงก่อน ด้วยบล็อกตั้งค่า ขนาด (set size to 100%) จากนั้นหลอดไฟจะมาอยู่ตรงกลางแผงวงจร (go to x:-111 y:-54) เพื่อให้ผู้ใช้นำมันไปวางในแผงวงจรเอง



ถัดไปเป็นการตรวจสอบการกดคีย์ space bar และการเปิดสวิตช์จากผู้ใช้ ซึ่งเกิดเมื่อใดก็ได้ ดังนั้นต้องตรวจสอบภายในบล็อกควบคุมตลอดเวลา (forever) การตรวจสอบจะทำ 2 ชั้น คือ ตรวจสอบการกดคีย์ก่อน (key space pressed?) ในบล็อกควบคุมถ้ามิฉะนั้น (if-else) ถ้าคีย์ถูก กดจะเปลี่ยนชุด เป็นสัญลักษณ์ทางไฟฟ้าของหลอดไฟ ถ้าไม่ได้กดคีย์ (else) จะตรวจสอบอีก ครั้งว่า หลอดไฟติดหรือไม่ โดยที่หลอดไฟจะติดได้ ก็ต่อเมื่อทุกอุปกรณ์เชื่อมต่อแผงวงจรและ สวิตช์ถูกเปิด

สวิตช์จะเปิดได้ มันต้องถูกเชื่อมต่อก่อน ดังนั้นตรวจสอบแค่สวิตช์เปิดอย่างเดียว (Switch=1) ก็ พอ การเชื่อมต่อของสวิตช์ก็ไม่จำเป็นต้องตรวจสอบก็ได้

ดังนั้นถ้าทุกอุปกรณ์เชื่อมต่อแผงวงจรและสวิตช์ถูกเปิด ((battconnected=1) and ((Switch=1) and (bulbconnected=1))) หลอดไฟจะติด โดยการเปลี่ยนชุด (switch to costume bulbon) มิฉะนั้นแล้วหลอดไฟจะดับ (switch to costume bulboff)

battconnected = 1 and Switch = 1 and bulbconnected = 1

การตรวจสอบเงื่อนไขหลายๆ เงื่อนไขพร้อมกัน (battconnected=1, Switch=1, bulbconnected=1) ทำได้โดยใช้โอเปอร์เรชั่นตรรกะและ (and) and ช่วยในการ ประมวณผล ว่าสุดท้ายแล้วเงื่อนไขได้จริงหรือเท็จ โดยที่ทุกเงื่อนไขต้องเป็นจริงทั้งหมดจึงจะให้ ตรรกะสุดท้ายเป็นจริง

การประมวณผลทางตรรกะ

การประมวณผลทางตรรกะใน Scratch ทำได้ 3 แบบ คือใช้ ตรรกะและ (and) ตรรกะหรือ (or) ตรรกะไม่ (not) คำตอบหาได้จากตารางความจริง (Truth table) ในแต่ละแบบดังนี้



ตรรกะ และ AND

ตารางความจริงของเงื่อนไข A กับเงื่อนไข B เมื่อผ่านโอเปอร์เรเตอร์ตรรกะแล้วจะได้ผลลัพธ์ C เป็นตารางความเป็นไปได้ทั้งหมดเมื่อเงื่อนไขของ A กับ B เป็นจริง (T) หรือเป็นเท็จ (F)

ตรรกะและ (and) จะให้ผลลัพธ์เป็นจริงได้กรณีเดียว คือเงื่อนไข A และ B ต้องเป็นจริงทั้งคู่ กรณีอื่นๆ เป็นเท็จหมด

จากตารางความจริงของตรรกะและ โอเปอร์เรเตอร์ตรรกะและ จึงถูกใช้ในสคริปต์หลอดไฟ เพราะต้องการให้ทุกเงื่อนไขเป็นจริง แต่การตรวจสอบเงื่อนไขหลายๆ เงื่อนไข ต้องทำเป็นคู่ๆ เช่น ในกรณีสคริปต์หลอดไฟ เช็คสวิตช์กับหลอดไฟก่อน ผลลัพธ์ที่ได้นำไปเช็คกับแบตเตอรีอีก ที ก็จะได้คำตอบสุดท้าย

| ตรรกะ หรือ OR | | |
|-----------------|---|---|
| A OR B = C | | |
| Α | В | С |
| Т | Т | Т |
| Т | F | Т |
| F | Т | Т |
| F | F | F |
| T= จริง F= เท็จ | | |

ตรรกะหรือ (or) จะให้ผลลัพธ์เป็นเท็จได้กรณีเดียว คือเงื่อนไข A และ B ต้องเป็นเท็จทั้งคู่ กรณี อื่นๆ ให้คำตอบเป็นจริงหมด



ตรรกะไม่ (not) จะให้ผลลัพธ์สลับกับข้อเท็จจริงที่เป็นอยู่ คือจากจริงเปลี่ยนเป็นเท็จ หรือจาก เท็จเปลี่ยนเป็นจริง

สคริปต์ของสวิตช์

สวิตช์มีสามชุด ชุดแรกเป็นรูปสวิตช์ปิด ชุดที่สองเป็นรูปสวิตช์เปิด รูปที่สามเป็นรูปสัญลักษณ์ ทางไฟฟ้าของสวิตช์



สังเกตแม่กุญแจที่อยูถัดจากชื่อ Switch จะเปิดอยู่ หมายความว่า สวิตช์สามารถใช้เมาส์ลากไป มาได้ ขณะโปรแกรมทำงานในโหมดนำเสนอ หรือบนเว็บไซต์ของ Scratch



สคริปต์ของสวิตช์ดูเหมือนจะซับซ้อน เพราะทำทุกอย่างภายในสคริปต์เดียว ทำให้สคริปต์ยาว แต่จริงๆ แล้วไม่ซับซ้อน โดยเฉพาะเมื่อดูสคริปต์ของอุปกรณ์อื่นๆ มาแล้ว

สคริปต์ของสวิตช์เริ่มเมื่อผู้ใช้คลิกธงเขียว จากนั้นกำหนดขนาดเริ่มต้นของสวิตช์เท่าของจริง ก่อน ด้วยบล็อกตั้งค่าขนาด (set size to 100%) ทั้งนี้เพราะขนาดมีผลต่อการเชื่อมเข้าแผงวงจร จากนั้นสวิตช์จะมาอยู่ตรงกลางแผงวงจร (go to x:133 y:-18) เพื่อให้ผู้ใช้นำมันไปวางใน แผงวงจรเอง

ถัดไปเป็นการตรวจสอบการกดคีย์ space bar การเชื่อมต่อแผงวงจร และการเปิดสวิตช์จากผู้ใช้ ซึ่งเกิดเมื่อใดก็ได้ ดังนั้นต้องตรวจสอบภายในบล็อกควบคุมตลอดเวลา (forever) การตรวจสอบจะทำ 3 ชั้น คือตรวจสอบการกดคีย์ก่อน (key space pressed?) ในบล็อก ควบคุมถ้ามิฉะนั้น (if-else) ถ้าคีย์ถูกกดจะเปลี่ยนชุด เป็นสัญลักษณ์ทางไฟฟ้าของสวิตช์

ถ้าไม่ได้กดคีย์ (else) จะตรวจสอบชั้นที่สอง ในบล็อกถ้ามิฉะนั้น (if-else) อีกครั้งว่า สวิตช์ เชื่อมต่อแผงวงจรหรือไม่ โดยตรวจดูว่าหลอดไฟสัมผัสกับสีเหลืองและสีฟ้า (ซึ่งเป็นสีของขั้วใน แผงวงจร) หรือยัง

ถ้าใช่ ให้ตั้งค่าตัวแปร Switchconnected เท่ากับ 1 หมายความว่า สวิทช์ต่อกับแผงวงจรแล้ว ทำให้ผู้ใช้สามารถเปิดสวิทช์ได้ ดังนั้นโปรแกรมสามารถตรวจสอบการเปิดสวิทช์ต่อได้ โดย ตรวจสอบชั้นที่สาม ในบล็อกถ้ามิฉะนั้น (if-else) อีกครั้งว่า สวิตช์เปิดหรือยัง (touching mouse-pointer?) ถ้าผู้ใช้นำเมาส์มาสัมผัสที่สวิตช์ ก็จะเปลี่ยนชุดเป็น Switchon แล้วตั้งค่าตัว แปรสวิตช์เท่ากับ 1 (switch = 1) มิฉะนั้นเปลี่ยนชุดเป็น Switchoff แล้วตั้งค่าตัวแปรสวิตช์ เท่ากับ 0 (switch = 0)

ถ้าสวิตช์ยังไม่ได้เชื่อมต่อแผงวงจร (else ของบล็อกถ้ามิฉะนั้นในชั้นที่ 2) ให้ตั้งค่าตัวแปร Switchconnected เท่ากับ 0 จากนั้นเปลี่ยนชุดเป็น Switchoff แล้วตั้งค่าตัวแปรสวิตช์เท่ากับ 0 (switch = 0)

การทำงานแบบมีเงื่อนไข

122

การควบคุมลำดับการทำงานที่ขึ้นกับเงื่อนไขมีหลายแบบ คือจะทำงานถ้าเงื่อนไขเป็นจริงเท่านั้น (บล็อกถ้า if) ทำงานได้ทั้งกรณีที่เงื่อนไขเป็นจริงหรือเป็นเท็จ (บล็อกถ้ามิฉะนั้น if-else) และ สุดท้ายคือการนำบล็อกทั้งสองแบบมาซ้อนกัน (nested if) โดยนำบล็อกหนึ่งไปซ้อนในอีกบล็อก หนึ่ง ซึ่งจะซ้อนกันกี่ชั้นก็ได้ แต่มากไปก็อาจทำให้สคริปต์ดูสับสนได้ และสามารถซ้อนในส่วนที่ เงื่อนไขเป็นจริง (ใต้ if) ก็ได้ หรือในส่วนที่เงื่อนไขเป็นเท็จ (ใต้ else) ก็ได้



ในสคริปต์ของ Bulb และ Switch จะเห็นการซ้อนกันของบล็อกเงื่อนไขหลายรูปแบบ ลำดับการ ทำงานในแบบแรก (Bulb) เริ่มต้นด้วยการตรวจสอบเงื่อนไขแรกก่อน (A) ถ้าเงื่อนไข A เป็นจริง ทำงานในส่วน (1) ถ้าเงื่อนไข A เป็นเท็จ (else) ตรวจสอบเงื่อนไขที่สอง (B) ถ้าเงื่อนไข B เป็น จริง ทำงานในส่วน (2) ถ้าเงื่อนไข B เป็นเท็จ (else) ทำงานในส่วน (3)

้ลำดับการทำงานในแบบที่สอง (Switch) น่าท้าทายกว่าแบบแรก ลองฝึกทำดูจะรู้ว่าไม่ยากเลย

การต่อยอดโปรเจก SimpleCircuit

- แก้ไขการต่อวงจรให้ถูกต้อง โดยให้ขั้วไฟฟ้าต่อกันเพียงอย่างเดียว
- เมื่อสวิตซ์เปิด แบตเตอรีจะถูกใช้งาน พลังงานจะลดลงตามเวลาในการใช้งาน ดังนั้น
 เพิ่มตัวแปรพลังงานของแบตเตอรี แล้วจับเวลาการใช้งานของแบตเตอรีทุกครั้งที่
 หลอดไฟติด เมื่อแบตเตอรีหมด หลอดไฟจะต้องดับ
- เมื่อพลังงานของแบตเตอรีลดลง เสียงสว่างก็จะลดลง เพิ่มแอนิเมชันให้กับหลอดไฟ แสดงการลดแสงลงตามพลังงานที่เหลือของแบตเตอรี

โปรเจก GardenSecret

โปรเจก GardenSecret เป็นการแสดงศิลปะแบบโต้ตอบ (Interactive Art) เริ่มต้นด้วยการเปิด โปรเจก โดยเลือก Open ในเมนู File จากนั้นคลิกปุ่ม Examples เลือกโฟล์เดอร์ Interactive Art เลือกโปรเจก GardenSecret

ภาพรวมของโปรเจก

โปรเจกนี้เป็นการสร้างสวนดอกไม้ ที่ผีเสื้อเป็นผู้นำดอกไม้มาที่สวนแห่งนี้ โดยอาศัยการ ออกแบบจากผู้ใช้ เป็นผู้ชี้นำตำแหน่งการปลูกดอกไม้ ด้วยการใช้เมาส์คลิกเพื่อระบุตำแหน่ง ผีเสื้อสามารถปลูกดอกไม้ได้หลากหลายสีสัน ระหว่างการปลูกจะมีเสียงดนตรีประกอบ ทำให้ สวนแห่งนี้ดูมีชีวิตชีวา และถ้าโชคดี นางฟ้าก็จะปรากฏ บินมาเที่ยวชมสวนแห่งนี้



ตัวละครและฉากเวที

ตัวละครใน GardenSecret มีผีเสื้อ นางฟ้า กลีบดอกไม้ 5 ชนิด และใบคำสั่งในการใช้งาน ส่วน ฉากหลังเวทีเรียบง่ายด้วยสีฟ้าอ่อน

สคริปต์ของตัวละครและเวที

สคริปต์ของใบคำสั่ง

ใบคำสั่งเป็นคำแนะนำสั้นๆ ให้ผู้ใช้คลิกตำแหน่งที่ต้องการปลูกต้นไม้ในสวนแห่งนี้



สคริปต์ของใบคำสั่ง เริ่มเมื่อผู้ใช้คลิกธงเขียว จากนั้นแสดง (show) คำแนะนำการปลูกดอกไม้ และรอ (wait) 3 วินาที ก่อนที่จะหายไป (hide)



สคริปต์ของผีเสื้อ

126

ผีเสื้อเป็นผู้นำดอกไม้มาที่สวนแห่งนี้ โดยปกติมันจะซ่อนตัว และรอคำสั่งจากผู้ใช้ว่าจะให้ปลูก ดอกไม้ตรงไหน เมื่อใดที่ผีเสื้อรู้แล้วว่าจะปลูกดอกไม้ตรงไหน มันก็จะบินไปที่นั้น และเลือกว่าจะ ปลูกดอกไม้ชนิดใด ดอกไม้ที่ได้รับเลือกก็จะบานออก และถ้าดอกไหนโชคดี ก็จะมีนางฟ้าลงมา เชยชม



สคริปต์แรกของผีเสื้อเริ่มต้นเมื่อผู้ใช้คลิกธงเขียว จากนั้นซ่อนตัว (hide) แล้วทำตัวให้เล็กลง (set size to 25%) และก่อนที่มันจะวาดดอกไม้ในสวน มันจะทำให้สวนนี้ว่างเปล่าก่อน ด้วยการ ลบ (clear) ทุกอย่างที่อยู่บนเวที



สคริปต์ที่สองของผีเสื้อเริ่มพร้อมสคริปต์แรก คือเมื่อผู้ใช้คลิกธงเขียว ผีเสื้อไม่รู้ว่า ผู้ใช้จะบอก ตำแหน่งการปลูกดอกไม้เมื่อไร ดังนั้นต้องคอยในบล็อกตลอดเวลา (forever)

ผีเสื้อจะรอจนกว่า (wait until) การซี้ตำแหน่งจากเมาส์ของผู้ใช้ (mouse down?) เมื่อใดที่เมาส์ ถูกคลิก ตำแหน่งของเมาส์จะถูกเก็บไว้ ในตัวแปรภายในของ Scratch นั้นก็คือ mouse x และ mouse y ผีเสื้อจะไปยังตำแหน่ง x ของเมาส์ และ y ที่ -200 ซึ่งอยู่ด้านล่างนอกกรอบเวที ก่อน จะปรากฏตัว (show) และส่งเสียง Chimes1 โดยบล็อกเล่นเสียง (play sound) ก่อนจะเริ่มต้น วาดดอกไม้

เพื่อเพิ่มความหลากหลายสีสันของสวนแห่งนี้ ผีเสื้อตั้งค่าสีของปากกา (set pen color to) โดย สุ่มค่าสีตั้งแต่ 20 ถึง 80 (pick random 20 to 80) และเปลี่ยนเฉดสี (change pen shade by) โดยสุ่มค่าตั้งแต่ -20 ถึง 20 ต่อด้วยการกำหนดขนาดของปากกา (set pen size to) โดยสุ่มค่า ตั้งแต่ 2 ถึง 6

จากนั้นจรดปากกา (pen down) หลังจากนี้ ถ้าผีเสื้อเคลื่อนที่ไปที่ใด ก็จะวาดลายเส้นไปด้วย ดังนั้นเมื่อมันเคลื่อนที่ไปยังตำแหน่งของเมาส์ ภายใน 1 วินาที (glide 1 secs to x: mouse x y: mouse y) ก็จะเป็นการวาดก้านของดอกไม้ เมื่อมันยกปากกา (pen up) มันก็จะหยุดวาด

ผีเสื้อจะเลือกดอกไม้ เก็บไว้ที่ตัวแปร flower (set flower to) จากการสุ่มดอกไม้ที่มีอยู่ 5 ชนิด (pick random 1 to 5) แล้วส่งสาร draw-flower เพื่อให้ดอกที่ได้สุ่มเลือกไว้ วาดดอกนั้นๆ

ก่อนที่ตัวมันจะจากไป (hide) มันยังเชิญชวนให้นางฟ้ามาชมดอกไม้ ที่จะถูกสร้างขึ้นมานี้ โดย สุ่มเลขแห่งโชคมาค่าหนึ่งตั้งแต่ 1 ถึง 100 มาเก็บไว้ในตัวแปร lucky และถ้าโชคดี นางฟ้าก็จะ มาเยือน



สคริปต์ของนางฟ้า

ทุกครั้งที่ผีเสื้อปลูกดอกไม้ จะเชิญนางฟ้าให้ไปเยี่ยมชมดอกที่ปลูกใหม่ แต่นางฟ้าจะไปหรือไม่ ขึ้นอยู่กับว่าดอกนั้นจะมีโชคหรือไม่



สคริปต์ของนางฟ้า เริ่มเมื่อผู้ใช้คลิกธงเขียว ปกตินางฟ้าจะไม่ปรากฏกาย (hide) แต่นางฟ้าจะ คอยตรวจสอบอยู่ตลอดเวลา ว่าดอกไหนมีโชค ในบล็อกตลอดเวลา (forever)

ค่าความโชคดีอยู่ในตัวแปร lucky ที่ผีเสื้อสุ่มขึ้นมา นางฟ้าจะคอยจนกระทั่ง ค่าความโชคดีมีค่า มากกว่า 95 เมื่อใด (wait until lucky > 95 ค่าสุ่มสูงขนาดนี้ ต้องโชคดีจริงๆ) นางฟ้าก็จะปรากฏ กาย ณ ตำแหน่งของดอกไม้ (ตำแหน่งที่เมาส์คลิก mouse-pointer)

วิธีการเยี่ยมชมของนางฟ้า อยู่ภายใต้เงื่อนไขว่าดอกไม้ปลูก ณ ตำแหน่งใด (mouse x > 0) บน เวที ถ้าอยู่ที่ซีกตะวันออก (x > 0) นางฟ้าจะหันไปยังทิศตะวันตก (-90) มิฉะนั้นจะหันไปยังทิศ ตะวันออก (90) โดยใช้บล็อกชี้ไปยังทิศ (point to direction) เป็นตัวกำหนดการหัน

นางฟ้าจะปรากฏกาย (show) แล้วตั้งค่าความโชคดีกลับเป็น 0 ใหม่ (set lucky to 0) และจะอยู่ เพียง 1 วินาที (wait 1 secs) ก่อนจะหายตัวไป (hide)

สคริปต์ของดอกไม้

การวาดดอกไม้ให้มีความหลากหลายนั้น แทนที่จะเพิ่มชุดแต่งกาย (costumes) มากขึ้น ซึ่งเป็น วิธีการหนึ่ง แต่ไม่มีประสิทธิภาพเท่าที่ควร เนื่องจากเปลืองพื้นที่การจัดเก็บ การวาดดอกไม้ใน โปรเจกนี้ ใช้เทคนิคการวาดกลีบดอกไม้แทน นั้นก็คือการใช้กลีบดอกไม้เพียงไม่กี่แบบ แล้ว กำหนดค่าสี ความสว่าง ขนาด และการจัดเรียงตำแหน่งของกลีบดอกไม้ใหม่ ก่อนที่จะวาด ดอกไม้ ด้วยการตราประทับ (stamp) กลีบดอกไม้ไปรอบ ๆ เป็นวงกลม ซึ่งวิธีนี้สามารถสร้าง ดอกไม้ให้มีความหลากหลายได้อย่างมีประสิทธิภาพ และแสดงแอนิเมชั่นของการวาดดอกไม้ได้ ด้วย



ก่อนจะเริ่มวาดดอกไม้ ผีเสื้อเลือกดอกไม้ด้วยการสุ่มค่าเก็บไว้ในตัวแปร flower แล้วส่งสาร draw-flower ให้ดอกไม้ที่เลือกไว้วาดรูป โดยที่แต่ละดอกจะต้องตรวจสอบเองว่า ตัวเองถูกเลือก หรือไม่ เช่น ดอกที่ 1 (flower1) ก็จะตรวจสอบว่าตัวแปร flower = 1 หรือไม่ ดอกที่ 2 (flower2) ก็จะตรวจสอบว่าตัวแปร flower = 2 หรือไม่ เป็นต้น

สคริปต์แรกของดอกที่ 1 เริ่มเมื่อผู้ใช้คลิกธงเขียว จากนั้นซ่อนตัวก่อน (hide) รอการเรียกให้วาด รูป



สคริปต์ที่สองของดอกที่ 1 เริ่มเมื่อได้รับสาร draw-flower ดอกที่ 1 จะวาดดอกไม้ ก็ต่อเมื่อ ตัวเองถูกเลือก ดังนั้นการวาดรูปต้องอยู่ในบล็อกเงื่อนไขถ้า (if)

ถ้าผีเสื้อเลือกดอกที่ 1 (flower=1) ดอกที่1 จะไปยังตำแหน่งของผีเสื้อ (go to butterfly) ย้าย มาเลเยอร์ด้านหน้า (go to front) เพื่อวาดรูปทับรูปดอกอื่นๆ ที่ถูกวาดก่อนหน้านี้ รวมทั้งตัว ผีเสื้อด้วย เพื่อให้การวาดดอกไม้น่าสนใจ มันจะส่งเสียง Fall1 ประกอบด้วย

ก่อนที่จะเริ่มวาดดอกไม้ ต้องตั้งค่าต่าง ๆ ของปากกาใหม่ นั่นก็คือ สี (color) ความสว่าง (brightness) และขนาด (size) เพื่อให้เกิดความหลากหลายสีสัน



โดยที่ค่าสีมาจาก ตำแหน่ง x ของเมาส์หารด้วย 5 (mouse x / 5) ค่าความสว่างได้จากการสุ่ม ค่าตั้งแต่ -25 ถึง 25 (pick random -25 to 25) ส่วนค่าขนาดค่อนข้างซับซ้อนสักหน่อย โดยนำ ค่าตำแหน่ง y ของเมาส์ มาบวก 220 แล้วหารด้วย 3 ทำให้ดอกที่อยู่สูงจะใหญ่กว่าดอกที่อยู่ ด้านล่าง หลังจากนั้นบวกกับค่าสุ่มตั้งแต่ -5 ถึง 5

การวาดดอกไม้ ทำได้จากการตราประทับ (stamp) กลีบดอกไม้ไปรอบ ๆ เป็นวงกลม ห่างกัน ทุก ๆ 15 องศา โดยประทับ 1 ครั้ง (stamp) แล้วก็หมุนตามเข็นนาฬิกาไป 15 องศา (turn 15 degrees) เพื่อให้ได้ 1 รอบพอดี ต้องประทับจำนวน (360/15) ครั้ง ในบล็อกทำซ้ำ (repeat)

สุดท้ายวาดตรงกลางของดอกไม้ ด้วยปากกาขนาด 8 (set pen size to 8) ด้วยสีเหลือง (set pen color) โดยจรดปากกาลง (pen down) แล้วขยับปากกา 1 ก้าว (move 1 steps) จากนั้นยก ปากกา (pen up) ก็จะได้ดอกไม้แบบที่ 1 มา 1 ดอก

การคำนวณใน Scratch

การเขียนโปรแกรม คงหนีไม่พันการคำนวณและการประมวลผลข้อมูล ซึ่งพบเห็นได้เกือบจะทุก โปรเจกใน Scratch โดยข้อมูลที่สามารถนำมาคำนวณได้ คือ ตัวเลข ตัวอักษร และข้อเท็จจริง การประมวลผลข้อมูลทำได้ โดยใช้บล็อกในหมวดโอเปอร์เรชั่นดังนี้

- การคำนวณตัวเลขทั่วไป ก็มีการบวก (+) ลบ (-) คูณ (*) หาร (/) นอกจากนั้นยังมี การ สุ่มค่า (pick random) และการใช้ฟังก์ชันคณิตศาสตร์อีกมากมาย
- ส่วนการจัดการข้อมูลตัวอักษรนั้น สามารถเชื่อมคำสองคำให้เป็นคำเดียวกันได้ (join) หรือดึงอักษรใดอักษรหนึ่งมาจากคำได้ (letter of) นอกจากนั้นยังนับจำนวนอักษร ทั้งหมดในคำได้ (length of)
- สำหรับตรรกะที่ใช้ในบล็อกเงื่อนไข หาได้จากการเปรียบเทียบตัวเลข เช่น การ
 เปรียบเทียบ มากกว่า (>) น้อยกว่า (<) หรือเท่ากับ (=) ซึ่งจะคืนค่าเป็น จริงหรือเท็จ
 เมื่อมีหลายเงื่อนไข ก็ยังสามารถประมวลผลระหว่างเงื่อนไขได้จาก ตรรกะและ (and)
 ตรรกะหรือ (or) และตรรกะไม่ (not)

สิ่งสำคัญอย่างหนึ่งในการคำนวณที่กล่าวมา คือลำดับการคำนวณ ตัวอย่างเช่น ถ้าต้องการ คำนวณ 3 * 5 - 4 / 2 ในโปรแกรมภาษาอื่น จะคำนวณตามกฏลำดับการคำนวณของภาษานั้น ๆ เพื่อลดความสับสนในเรื่องลำดับการคำนวณ ใน Scratch จะคำนวณได้ที่ละคู่ เหมือนการใส่ วงเล็บ ซึ่งคนเขียนโปรแกรมต้องเป็นผู้รู้ว่า ลำดับการคำนวณเป็นอย่างไร แล้วจับคู่ให้ถูกลำดับ มิเช่นนั้นจะได้ผลลัพธ์แตกต่างกัน เช่น ((3 * 5) - (4 / 2)) = 13 หรือ ((3 * (5 – 4)) / 2) = 1.5 เป็นต้น



สคริปต์แรกของดอกที่ 2 เหมือนกับสคริปต์แรกของดอกที่ 1 ส่วนสคริปต์ที่สองของดอกที่ 2 คล้ายกับสคริปต์ที่สองของดอกที่ 1 แต่ดอกที่ 2 มีกลีบดอกอยู่ 2 ชั้น โดยจะเริ่มต้นวาดด้วยกลีบ แบบแรก (costume1) เหมือนดอกที่ 1 แล้วจึงวาดกลีบแบบที่สอง (costume2) โดยสคริปต์เริ่มเมื่อได้รับสาร draw-flower ถ้าตัวเองถูกเลือก (flower=2) ก็จะไปที่ผีเสื้อ ย้าย มาเลเยอร์หน้าสุด แล้วเปลี่ยนชุดเป็นกลีบแบบแรก (switch to costume costume1) จากนั้นส่ง เสียง Beep1

การตั้งค่าของปากกาทำเหมือนดอกที่ 1 ก่อนจะวาดกลีบแบบแรกด้วยการตราประทับ แล้วหมุน ตามเข็มนาพิกา 15 องศา จำนวน (360/15) ครั้ง ได้หนึ่งรอบพอดี

จากนั้นเปลี่ยนชุดเป็นกลีบแบบที่สอง ลดขนาดกลีบด้วยการสุ่มค่าตั้งแต่ 25 ถึง 100 (เป็น กลีบชั้นใน) จากนั้นก็ประทับด้วยกลีบแบบที่สอง โดยคราวนี้จะหมุนทวนเข็นนาพิกาทุกๆ 20 องศา จำนวน 360/20 ครั้ง ได้กลีบอีกหนึ่งชั้น



สคริปต์แรกของดอกที่ 3 เหมือนกับสคริปต์แรกของดอกที่ 1 ส่วนสคริปต์ที่สองของดอกที่ 3 คล้ายกับสคริปต์ที่สองของดอกที่ 1 โดยสคริปต์เริ่มเมื่อได้รับสาร draw-flower ถ้าตัวเองถูกเลือก (flower=3) ก็จะไปที่ผีเสื้อ ย้ายมาเลเยอร์หน้าสุด แล้วส่งเสียง Cowbell1

จากนั้นตั้งค่าของปากกาเหมือนดอกที่ 1 แล้ววาดดอกไม้ด้วยการตราประทับ โดยหมุนตามเข็ม นาพิกา 30 องศา จำนวน (360/30) ครั้ง ได้หนึ่งรอบพอดี

136



สคริปต์ของดอกที่ 4 คล้ายกับสคริปต์ของดอกที่ 3 มาก ต่างกันตรงเสียงประกอบ ที่ใช้เสียง Laser1 แทน



สคริปต์ของดอกที่ 5 ก็คล้ายกับสคริปต์ของดอกที่ 3 มากเหมือนกัน ต่างกันตรงเสียงประกอบ ที่ ใช้เสียง Gong1 แทน

138

การต่อยอดโปรเจก GardenSecret

- รูปแบบตราประทับ (stamp) แบบอื่นๆ เช่น กลีบเว้นกลีบ กลีบวนแล้วมีขนาดลดลง
 ด้วย เหมือนวงกันหอย (spiral) กลีบสี่แฉก กลีบห้าแฉก เป็นต้น
- การวาดดอกไม้ ด้วยการวาดลายเส้น จากการเคลื่อนที่ และการเปลี่ยนทิศ
- แทนที่จะเป็นดอกไม้ เปลี่ยนเป็นพลุแทน เมื่อผู้ใช้คลิกที่ใด จะมีแอนิเมชั่นขณะพลุขึ้น ระเบิด แล้วหายไป พร้อมทั้งเสียงประกอบ การออกแบบค่อนข้างยาก แต่ก็ท้าทาย เหมือนกัน โดยเฉพาะทำให้พลุ ค่อย ๆ หายไปเมื่อเวลาผ่านไป

โปรเจก PianoMachine

โปรเจก PianoMachine เป็นโปรเจกทำแอนิเมชั่นง่ายๆ เน้นที่เสียงดนตรี จึงถูกจัดอยู่ในกลุ่ม ของ Music and Dance เปิดโปรเจกด้วยการเลือก Open ในเมนู File จากนั้นคลิกปุ่ม Examples เลือกโฟล์เดอร์ Music and Dance แล้วเลือกโปรเจก PianoMachine

ภาพรวมของโปรเจก

บนเวทีมีคีย์เบียโน ซึ่งแต่ละคีย์มีเสียงต่างกัน เมื่อลูกบอลมากระทบ หรือผู้ใช้คลิกที่คีย์เบียโน คีย์นั้นจะแสดงแอนิเมชั่นง่ายๆ เสมือนว่าคีย์เบียโนถูกกดลงพร้อมกับเสียงคีย์นั้น แล้วคีย์เบียโน ก็เด้งขึ้นเอง ขณะที่โปรแกรมทำงาน ผู้ใช้สามารถลากคีย์เหล่านี้ไปวางตำแหน่งใหม่ได้ เพื่อให้ เกิดเสียงรูปแบบใหม่ๆ



บนเวทียังมีปุ่มต่างๆ (User interface) ให้ผู้ใช้ควบคุม ขณะโปรแกรมทำงาน เช่น ตอนเริ่มต้น โปรแกรม ลูกบอลจะมีอยู่ 3 ลูก เมื่อกดปุ่มจำนวนลูกบอล (ปุ่มสีเหลือง) ลูกบอลจะเพิ่มขึ้นที่ละ ลูก เมื่อถึง 6 ลูกแล้ว จะวนกลับมาที่ 1 ลูกใหม่

ปกติการเคลื่อนที่ของลูกบอล จะเคลื่อนที่ไปเรื่อย ๆ ถ้าชนขอบเวทีก็จะสะท้อนกลับ แต่ถ้าชนคีย์ เบียโนแล้ว อาจสะท้อนกลับ (rebound) หรือผ่านเลยไป (through) ทั้งนี้ขึ้นกับสถานะของปุ่ม ควบคุมการสะท้อน (ปุ่มสีน้ำเงิน)

นอกจากสองปุ่มนี้แล้ว ยังมีสไลเดอร์ที่ใช้ควบคุมความเร็วของลูกบอล และถ้าผู้ใช้คลิกที่เวที ลูก บอลทุกลูกก็จะเปลี่ยนทิศเองแบบสุ่มอีกด้วย

ตัวละครและเวที

โปรเจกนี้มีตัวละครมากถึง 26 ตัว แต่ถ้าจัดกลุ่มแล้ว จะมีใบคำสั่ง 1 ใบ ปุ่มคำสั่ง 2 ปุ่ม ลูกบอล 6 ลูก และคีย์เปียโน 17 คีย์ ส่วนฉากหลังเวทีเรียบง่ายเป็นเฉดสีฟ้าตามแนวรัศมีวงกลม

สคริปต์ของตัวละครและเวที

สคริปต์ของคำสั่ง

142

คำสั่งที่อยู่บนเวที สั่งให้ดูคำสั่งของโปรเจกนี้ใน Project Notes อีกที ซึ่งเปิดดูได้ โดยคลิกเมนู File เลือก Project Notes ... ทั้งนี้อาจเป็นเพราะว่าคำสั่งค่อนข้างยาว ไม่เหมาะที่จะใส่ไว้บนเวที วิธีนี้น่าจะดีกว่า



ส่วนสคริปต์ของใบคำสั่งก็ง่าย ๆ คือเมื่อผู้ใช้คลิกธงเขียวแล้วมันจะหายไป ด้วยเอฟเฟคหายตัว (ghost) เท่านั้นเอง



สคริปต์ของเวที

สคริปต์ของเวที เริ่มต้นทุกครั้งที่ผู้ใช้คลิกที่ว่างบนเวที โดยใช้บล็อกควบคุมคอยตรวจสอบว่าเวที ถูกคลิกหรือยัง (when Stage clicked) ถ้าเวทีถูกคลิก เวทีจะส่งสาร new random direction ออกไป ซึ่งตัวละครใดที่รอรับสารนี้ (ลูกบอลทั้งหมด) จะถูกกระตุ้นให้ทำงานต่อไป



สคริปต์ของคีย์เปียโน

เสียงเปียโนเกิดจากการกระทบของลูกบอลกับตัวคีย์เปียโน หรือผู้ใช้คลิกตัวคีย์เปียโนโดยตรง เปียโนมีเสียงต่างกัน 17 เสียง จาก 17 คีย์ แต่ทุกคีย์ใช้หลักการเดียวกัน ทำให้สคริปต์ออกมา คล้ายกันมาก ต่างกันแค่เสียงของแต่ละคีย์เท่านั้น ดังนั้นมาดูสคริปต์ของคีย์ keyG-55 ซึ่ง สามารถเป็นตัวแทนของคีย์ที่เหลือได้เป็นอย่างดี



สคริปต์แรกของคีย์ keyG-55 เริ่มเมื่อผู้ใช้คลิกธงเขียว ตัวคีย์จะอยู่ในสภาวะที่ยังไม่ได้กด โดย เปลี่ยนชุดเป็น up สคริปต์ที่สองเริ่มเมื่อคีย์นี้ถูกคลิก (when keyG-55 clicked) จากการคลิกคีย์ นี้โดยตรงของผู้ใช้ ก็จะส่งสาร playG-55 ออกไป

สคริปต์ที่สามเริ่มเมื่อตัวมันเองได้รับสาร playG-55 ตัวคีย์จะเปลี่ยนชุดเป็น down เพื่อแสดง การกดลงของคีย์ จากนั้นเล่นเสียงตัวโน๊ตที่ 55 ครึ่งจังหวะ (play note 55 for 0.5 beats) แล้ว เปลี่ยนเครื่องแต่งกายเป็น up เป็นอันครบรอบการแสดงแอนิเมชั่นของการกดคีย์เบียโนหนึ่งครั้ง

นอกจากการรับสาร playG-55 จากตัวคีย์เองแล้ว สคริปต์นี้ยังรับสารจากลูกบอล เมื่อลูกบอล กระทบคีย์นี้ด้วย เทคนิคนี้ทำให้เขียนสคริปต์เพียงครั้งเดียวเท่านั้น สำหรับแสดงแอนิเมชั่นการ กดคีย์พร้อมเสียงของคีย์นั้นๆ
สคริปต์ของลูกบอล

ปกติลูกบอลจะเคลื่อนที่ไปมาตลอด เมื่อชนขอบเวทีก็จะสะท้อนกลับ แต่ถ้าชนกับคีย์เบียโนจะ เกิดเสียงของคีย์นั้นๆ ส่วนจะสะท้อนกลับหรือผ่านเลยไป ขึ้นกับสถานะการสะท้อน ในตัวแปร rebound ณ ขณะนั้น ลูกบอลมีอยู่ด้วยกัน 6 ลูก ตอนเริ่มต้นมีอยู่เพียง 3 ลูกเท่านั้นบนเวที แต่ สามารถเพิ่มลดได้ ด้วยปุ่มควบคุมจำนวนลูกบอล

สคริปต์ของลูกบอลนั้นค่อนข้างยาวทีเดียว มีถึง 20 สคริปต์ แต่สคริปต์ส่วนใหญ่ เป็นสคริปต์ที่ คอยตรวจสอบการชนคีย์เปียโนต่างๆ (17 คีย์) ซึ่งจะคล้ายกันหมด สคริปต์ของลูกบอลทั้ง 6 ลูก มีลักษณะคล้ายกันมากๆ ดังนั้นมาดูสคริปต์ของลูกบอล 1 ก็น่าจะเพียงพอกับการทำความเข้าใจ ลูกบอลที่เหลือ



สคริปต์แรกเริ่มเมื่อผู้ใช้คลิกธงเขียว ลูกบอลจะย้ายไปเลเยอร์ด้านหน้าสุด (go to front) เพื่อให้ เห็นลูกบอลตลอดเวลาขณะเคลื่อนที่ไปมา จากนั้นกำหนดทิศการเคลื่อนที่ (point in direction) แบบสุ่มตั้งแต่ 1 ถึง 360 องศา จากนั้นลูกบอลจะเคลื่อนที่ไปเรื่อย ๆ ภายในบล็อกควบคุม ตลอดเวลา (forever) การเคลื่อนที่จะเร็วหรือช้า ถูกกำหนดในตัวแปร speed ซึ่งมีค่าตั้งแต่ 10 ถึง 20 และผู้ใช้สามารถปรับเลื่อนได้ด้วยสไลเดอร์บนเวที จากนั้นจะตรวจสอบการชนขอบเวที ถ้าชนก็จะสะท้อนกลับ ก่อนที่จะเคลื่อนที่ต่อไปก็จะหยุดพัก (wait) 0.01 วินาที



ระหว่างที่ลูกบอลเคลื่อนที่ไป ก็จะตรวจสอบด้วยว่าตัวเองไปกระทบกับคีย์ไหนบ้าง การ ตรวจสอบนี้ทำเหมือนกันหมดทั้ง 17 คีย์ มาดูคีย์ keyC-48 ในสคริปต์ที่สองเป็นตัวอย่าง ส่วนคีย์ อื่นก็คล้ายๆ กัน ยกเว้น keyC-60 ในสคริปต์ที่สาม ที่ไม่มีการสะท้อนของลูกบอลเมื่อคีย์ถูกชน

สคริปต์ที่สองเริ่มเมื่อผู้ใช้คลิกธงเขียว การตรวจสอบนี้เกิดขึ้นเมื่อใดก็ได้ ทำให้ต้องทำในบล็อก ตลอดเวลาถ้า (forever if) ถ้าลูกบอลสัมผัสกับคีย์นี้ (touching keyC-48) ก็จะตรวจสอบต่อไปว่า ตัวแปร rebound=1 หรือไม่ ถ้าใช่ ก็จะสะท้อน โดยกำหนดทิศการเคลื่อนที่ใหม่เท่ากับ 180 + direction โดยที่ตัวแปร direction เป็นตัวแปรภายใน ที่เก็บค่าทิศการเคลื่อนที่ป[้]จจุบันของลูก บอลไว้ ทำให้ลูกบอลสะท้อนกลับ 180 องศาจากทิศเดิม จากนั้นมันจะส่งสารบอกคีย์ที่มันชน และจะรอจนกว่าแอนิเมชั่นของคีย์จบลง (broadcast playC-48 and wait) เพื่อให้เกิดการสัมผัส คีย์แค่ครั้งเดียว ลูกบอลจะรอตัวมันเองเคลื่อนออกจากคีย์นี้ก่อน โดยตรวจสอบว่าตัวมันไม่ สัมผัสกับคีย์นี้แล้ว (not tounching keyC-48)



สคริปต์ที่สี่ เริ่มเมื่อได้รับสาร new random direction จากการที่ผู้ใช้คลิกที่ว่างบนเวที ทำให้ลูก บอลเปลี่ยนทิศการเคลื่อนที่ของตัวเองแบบสุ่มตั้งแต่ 1 ถึง 360 องศา ทำให้มีโอกาสได้เสียงใหม่ เกิดขึ้น

สคริปต์ที่ห้า เริ่มเมื่อได้รับสาร number of balls changed จากการที่ผู้ใช้คลิกปุ่มควบคุมจำนวน ลูกบอล ลูกบอลแต่ละลูก จะตรวจสอบสิทธิการปรากฏตัวของตัวเอง จากตัวแปร balls เช่น เมื่อ ลูกบอล 1 (ball1) ได้รับสารนี้ จะตรวจสอบดูว่า balls > 0 หรือไม่ ลูกบอล 2 (ball2) ก็จะ ตรวจสอบดูว่า balls > 1 หรือไม่ เป็นต้น

สำหรับลูกบอล 1 ถ้า balls มากกว่า 0 ก็จะปรากฏบนเวที (show) ถ้าไม่ใช่ก็ซ่อนตัว (hide) ซึ่ง ในตอนเริ่มต้น ปุ่มควบคุมจำนวนลูกบอลได้กำหนดตัวแปร balls = 3 ทำให้ลูกบอล 1-3 ปรากฏ บนเวที

การทำซ้ำตัวละครที่มีลักษณะคล้ายกัน

ใน Scratch ตัวละครต่างชนิดกัน มักถูกสร้างของใครของมัน แต่ถ้าตัวละครเป็นกลุ่มเดียวกัน ที่ คล้ายกัน เช่น คีย์เบียโน กับ ลูกบอล ในโปรเจกนี้ แทนที่จะสร้างที่ละตัว ควรจะเริ่มจากการ สร้างต้นแบบของตัวละครก่อน แล้วทำซ้ำตัวละครจากต้นแบบอีกที จำนวนเท่าไรก็ได้ โดยคลิก ขวาที่ตัวละครต้นแบบในรายการตัวละคร จากนั้นเลือก duplicate ซึ่งตัวละครใหม่จะได้สคริปต์ และชุดมาทั้งหมดจากตัวละครต้นแบบ ต่อจากนั้นเราสามารถแก้ไขเพิ่มเติมตัวละครใหม่ได้



ในกรณีของลูกบอล จะเห็นประโยชน์อย่างมาก เพราะแต่ละลูกมี 20 สคริปต์ แล้วชุดก็เหมือนกัน หมด การทำซ้ำตัวละครจากต้นแบบ จะช่วยลดเวลาในการสร้างตัวละครใหม่อย่างมาก สังเกตว่า ทุกครั้งที่ทำซ้ำ แม้จะได้ทุกอย่างจากตัวละครต้นแบบ แต่ตัวละครที่ได้จากการทำซ้ำ จะได้ชื่อตัว ละครใหม่ ซึ่งถือว่าเป็นอิสระจากตัวละครต้นแบบ

สคริปต์ของปุ่มควบคุมจำนวนลูกบอล

ผู้ใช้สามารถเพิ่มหรือลดจำนวนลูกบอลบนเวทีได้ โดยคลิกปุ่มควบคุมจำนวนลูกบอล ทุกครั้งที่ คลิกจะเพิ่มลูกบอล 1 ลูกบนเวที จนกระทั่งครบ 6 ลูกแล้ว จะกลับมาเริ่มที่ 1 ลูกใหม่



สคริปต์แรกเริ่มเมื่อผู้ใช้คลิกธงเขียว จะกำหนดจำนวนลูกบอลเริ่มต้นเท่ากับ 3 ในตัวแปร balls (set balls to 3) แล้วเปลี่ยนชุดเป็นหมายเลข 3 จากนั้นส่งสาร number of balls changed (ให้ ลูกบอลทุกลูก)

สคริปต์ที่สองเริ่มเมื่อผู้ใช้คลิกที่ปุ่มนี้ (when balls clicked) ปุ่มจะแสดงแอนิเมชั่นสั้นๆ ให้รู้ว่า มันถูกคลิก ด้วยการลดขนาดเหลือ 75% (set size to 75%) แล้วคอย 0.3 วินาที จากนั้นจะ กำหนดค่าในตัวแปร balls ใหม่ โดยต้องการให้เพิ่มที่ละ 1 แต่เมื่อถึง 6 ให้กลับไปเริ่ม 1 ใหม่ ทำ ได้โดยใช้สูตร (1 + (balls mod 6))

เมื่อเปลี่ยนค่าในตัวแปรแล้ว จึงส่งสาร number of balls changed ให้ลูกบอลทุกลูกทราบ แล้วก็ เปลี่ยนชุดถัดไป (next costume) ก่อนที่จะเปลี่ยนขนาดกลับมาเท่าเดิม (100%)



สังเกตว่า ชุดตัวละครมีอยู่ 6 ชุด เรียงตามตัวเลขด้วยความตั้งใจ ทั้งนี้เพราะเมื่อใช็บล็อก next costume มันจะเลือกชุดถัดไปตามลำดับ เมื่อไปถึงชุดสุดท้ายแล้ว มันจะย้อนกลับไปที่ชุดแรก โดยอัตโนมัติ

การใช้โอเปอร์เรเตอร์ mod

mod เป็นการหารที่เอาเศษเป็นผลลัพท์ เช่น 3 mod 6 ก็คือ 3 หารด้วย 6 ได้ 0 เศษ 3 ผลลัพท์ ของการ mod ที่ได้เท่ากับ 3 อีกตัวอย่างหนึ่ง 8 mod 6 ก็คือ 8 หารด้วย 6 ได้ 1 เศษ 2 ผลลัพท์ ของการ mod ที่ได้เท่ากับ 2

จากสูตรการคำนวณ (1 + (balls mod 6)) ที่ใช้ในโปรเจกนี้ ทำให้การเปลี่ยนชุดเพิ่มที่ละ 1 แต่ เมื่อถึงชุดที่ 6 แล้วสามารถกลับไปเริ่มต้นที่ 1 ได้อีกครั้ง

เมื่อ balls มีค่าเท่ากับ 5, balls mod 6 ก็คือ 5 หารด้วย 6 ได้ 0 เศษ 5 ผลลัพท์ของการ mod ที่ ได้เท่ากับ 5 เมื่อ balls mod 6 เท่ากับ 5 ดังนั้น

 $1 + (balls \mod 6) = 1 + 5 = 6$

แต่เมื่อ balls มีค่าเท่ากับ 6, balls mod 6 ก็คือ 6 หารด้วย 6 ได้ 1 เศษ 0 ผลลัพท์ของการ mod ที่ได้เท่ากับ 0 เมื่อ balls mod 6 เท่ากับ 0 ดังนั้น

1 + (balls mod 6) = 1 + 0 = 1 นั้นคือสิ่งที่ต้องการ

สคริปต์ของปุ่มควบคุมการสะท้อนของลูกบอล



ปกติลูกบอลจะเคลื่อนที่ไปเรื่อย ๆ เมื่อชนขอบเวทีก็จะสะท้อนกับ แต่เมื่อชนคีย์เบียโน การ สะท้อนของลูกบอล ขึ้นอยู่กับสถานะการสะท้อนในตัวแปร rebound ที่ถูกกำหนดโดยปุ่มควบคุม การสะท้อน

สถานะการสะท้อนมีอยู่ 2 สถานะ คือ สะท้อน (rebound) จะทำให้ลูกบอลสะท้อนกลับ หรือผ่าน (through) จะทำให้ลูกบอลผ่านเลยไป สถานะป[ั]จจุบันดูได้จากชุดตัวละครนี้ ซึ่งปรากฏอยู่บนเวที



สคริปต์แรกของปุ่มควบคุมการสะท้อน เริ่มเมื่อผู้ใช้คลิกธงเขียว และเริ่มต้นด้วยสถานะสะท้อน โดยเปลี่ยนชุดเป็น rebound จากนั้นตั้งค่าตัวแปร rebound=1 (สถานะสะท้อน)

สคริปต์ที่สองเริ่มเมื่อผู้ใช้คลิกที่ปุ่มนี้ (when Rebound Button clicked) มันจะแสดงแอนิเมชั่น สั้นๆ ด้วยการลดขนาดเหลือ 75% แล้วคอย 0.3 วินาที

ทุกครั้งที่ปุ่มนี้ถูกคลิกจากผู้ใช้ ก็จะเปลี่ยนสถานะสลับไปมาระหว่างสถานะสะท้อนกับสถานะ ผ่าน ด้วยการตรวจสอบสถานะการสะท้อนป[ั]จจุบัน ว่าเป็นสถานะผ่านหรือไม่ (rebound=0) ถ้า ใช่ก็จะเปลี่ยน rebound เป็น 1 (เปลี่ยนเป็นสถานะสะท้อน) แล้วเปลี่ยนชุดเป็น rebound มิฉะนั้นแล้ว (else) จะเปลี่ยน rebound เป็น 0 (เปลี่ยนเป็นสถานะผ่าน) แล้วเปลี่ยนชุดเป็น through

จากนั้นเปลี่ยนขนาดของปุ่มกลับมาเท่าเดิม 100% เพื่อจบการแสดงแอนิเมชั่น

ระหว่างโปรแกรมทำงาน ผู้ใช้สามารถเปลี่ยนเสียงเพลงได้หลายวิธี เช่น เปลี่ยนการสะท้อนกลับ ของลูกบอลในตัวแปร rebound เปลี่ยนจำนวนลูกบอลในตัวแปร balls คลิกที่ว่างบนเวที และ สุดท้ายเพิ่มหรือลดความเร็วของลูกบอลในตัวแปร speed

ผู้ใช้สามารถปรับค่าตัวแปรความเร็ว speed ได้ด้วยสไลเดอร์บนเวที แต่ก่อนหน้านั้นต้องทำให้ ตัวแปรอยู่ในรูปแบบสไลเดอร์ และแสดงอยู่บนเวทีก่อน โดย (1) คลิกกล่องหน้าบล็อกตัวแปร speed ในบล็อกหมวดตัวแปร เครื่องหมายถูกบอกว่าตัวแปรนี้ได้แสดงอยู่บนเวทีแล้ว จากนั้น (2) คลิกขวาที่ตัวแปรนี้บนเวที จะปรากฏเมนูให้เลือกรูปแบบการแสดง คลิกเลือก slider เพื่อให้ มันสามารถเลื่อนปรับค่าได้ด้วยสไลเดอร์ ขณะโปรแกรมกำลังทำงาน นอกจากนั้นสามารถตั้งค่า ต่ำสุดสูงสุดได้ โดย (3) คลิก set slider min and max เช่นตัวแปร speed ตั้งค่าต่ำสุดที่ 10 และ ค่าสูงสุดที่ 20 เป็นต้น



การออกแบบหน้าตาโปรแกรม

การออกแบบหน้าตาโปรแกรมเป็นเรื่องสำคัญที่ไม่ควรมองข้าม โดยเฉพาะโปรแกรมที่ต้องการ โต้ตอบกับผู้ใช้ แม้เป็นโปรแกรมเล็กๆ แต่อาจใช้งานยาก ถ้าหน้าตาโปรแกรมไม่สื่อถึงการใช้ งาน ก็ไม่สามารถดึงศักยภาพของโปรแกรมออกมาได้ ผู้ใช้มีช่องทางให้ติดต่อและโต้ตอบกับ โปรแกรมได้หลายวิธี เช่น

ข้อความบนเวที เป็นการสื่อสารตรงไปยังผู้ใช้ ส่วนใหญ่เป็นคำสั่ง หรือวิธีใช้งาน ตัวแปรเมื่อ สร้างแล้ว สามารถแสดงบนเวทีได้ แต่ถ้าต้องการให้ผู้ใช้ปรับเปลี่ยนค่าของมันได้ ขณะที่ โปรแกรมทำงาน ทำได้โดยแสดงตัวแปรในรูปแบบสไลเดอร์ เหมือนตัวแปรความเร็ว (speed)

การสร้างปุ่มคำสั่ง ทำได้โดยการสร้างตัวละคร ที่รับการคลิกเมาส์จากผู้ใช้ เพื่อให้โปรแกรม ทำงานตอบสนอง เช่น เมื่อผู้ใช้คลิกที่คีย์เบียโนใดๆ ก็จะมีเสียงออกมาจากคีย์นั้นๆ แต่การ สร้างตัวละครให้เป็นปุ่มคำสั่งโดยเฉพาะก็มี เช่น ปุ่มเพิ่มหรือลดจำนวนลูกบอล และปุ่มการ สะท้อนของลูกบอล

โปรแกรมยังรับคำสั่งจากผู้ใช้ ผ่านการกดคีย์ต่างๆ บนคีย์บอร์ด เช่น คีย์ตัวอักษร a-z คีย์ตัวเลข 0-9 คีย์ลูกศร และคีย์เว้นวรรค (space bar)

นอกจากนั้น ผู้ใช้สามารถติดต่อโปรแกรมผ่านไมโครโฟน หรือผ่านอุปกรณ์ต่อพ่วง เช่น LEGO® WeDo™

การต่อยอดโปรแกรม PianoMachine

- Scratch มีเครื่องดนตรีหลายชนิดให้เลือกใช้ ลองเพิ่มหรือเปลี่ยนเครื่องดนตรีชนิดอื่น
- บันทึกการบรรเลงเพลงเบียโน แล้วเล่นซ้ำได้ การบันทึกนั้นต้องบันทึกทั้งตัวโน๊ตและ จังหวะเวลาในการเล่นตัวโน๊ตที่ถูกบันทึก
- การบันทึกต้องสร้างปุ่มเพิ่ม เพื่อให้ผู้ใช้คลิกเมื่อต้องการบันทึก ควรใช้ตัวแปรลิสต์มา เก็บบันทึกตัวโน๊ตและเวลา
- การเล่นซ้ำ เล่นเสียงจากข้อมูลในลิสต์ ขณะเล่นซ้ำ ลูกบอลควรหยุด เพื่อไม่ให้เสียงดีกัน

โปรเจก FishChomp

การทำเกมใน Scratch ไม่ยากอย่างที่คิด FishChomp เป็นตัวอย่างที่ดีในการเริ่มต้นการทำเกม ใน Scratch โดยเกมนี้เป็นเกมปลาใหญ่กินปลาเล็ก ผู้เล่นใช้เมาส์นำทางปลาใหญ่ไปกินปลาเล็ก เกมนี้ดูง่ายจนรู้สึกเหมือนว่าจะยังไม่สมบูรณ์ เพราะปลาทองที่ถูกกินจะออกมาใหม่เรื่อย ๆ ไม่มี วันจบเกม และเกมนี้ไม่มีคะแนนให้ ไม่มีข้อจำกัดเรื่องเวลา อย่างไรก็ตาม เกมนี้ให้แนวคิด สำหรับผู้เริ่มต้นในการทำเกม และยังเว้นช่องว่างให้เราขยายเกมนี้ต่อไปได้อีก

ภาพรวมของโปรเจก

เกมนี้มีปลาทองตัวเล็ก 3 ตัวเคลื่อนที่ไปเรื่อยๆ ในท้องทะเล ผู้เล่นเพียงลากเมาส์นำปลาตัวใหญ่ ไปกินเหล่าปลาทอง ตัวใดถูกกินก็จะโผล่มาอีกไม่มีวันจบ



ตัวละครและเวที

โปรเจกนี้มีตัวละครเป็นปลาสองชนิด คือปลาตัวใหญ่กับปลาทองเหมือนกัน 3 ตัว และใบคำสั่ง แสดงวิธีการเล่นเกม ซึ่งจะค้างอยู่ที่มุมบนขวาของเวทีตลอดเวลา เพราะมันไม่มีสคริปต์นั่นเอง

สคริปต์ของตัวละครและเวที

สคริปต์ของปลาทอง

ในเกม FishChomp เหล่าปลาทองจะว่ายน้ำไปมา ผู้เล่นใช้เมาส์นำทางปลาตัวใหญ่มากินปลา ทอง แต่พวกมันก็กลับมาเกิดใหม่อย่างรวดเร็ว แล้วก็ถูกกินอีกเป็นวงจรชีวิตในเกมนี้

ปลาทองมีอยู่ 3 ตัว มีลักษณะและพฤติกรรมเหมือนกันหมด ยกเว้นทิศการเคลื่อนที่ที่เริ่มต้น ต่างกัน เพราะฉะนั้นมาดูสคริปต์ของ goldfish 1 ก็พอ

| 🖲 🔥 goldfish 1 🔒 |
|---|
| 💽 🧡 x: 29 y: -72 direction: 163 |
| Scripts Costumes Sounds |
| |
| when 🚔 clicked |
| point in direction 90 |
| forever |
| move 2 steps |
| turn 🗣 pick random -20 to 20 degrees |
| if on edge, bounce |
| if color is touching |
| broadcast got-me |
| hide |
| wait 3 secs |
| go to x: -200 y: pick random -200 to 200) |
| show |
| |

สคริปต์ของปลาทองเริ่มเมื่อผู้เล่นคลิกธงเขียว ปลาทองก็จะปรากฏ (show) และหันไปยังทิศ 90 องศา จากนั้นมันจะมีพฤติกรรมซ้ำ ๆ ตลอด ทำให้ต้องอยู่ในบล็อกตลอดเวลา (forever) นั้นก็คือ เคลื่อนที่ไปที่ละนิด (2 หน่วย) เปลี่ยนทิศเล็กน้อยโดยการสุ่มตั้งแต่ -20 ถึง 20 องศา ถ้าชนขอบ เวทีก็จะสะท้อนกลับ

จากนั้นตรวจสอบว่าสีส้ม (สีของปลาทองเอง) สัมผัสกับสีน้ำเงิน (สีของปากปลาตัวใหญ่) หรือไม่ ถ้าใช่แสดงว่าปลาทองถูกกิน มันจะส่งสาร got-me ไปให้ปลาตัวใหญ่ แล้วก็หายตัวไป (hide) รอ อีก 3 วินาที (wait) มันจะโผล่ขึ้นมาใหม่ที่ตำแหน่ง x -200 และสุ่มค่า y ตั้งแต่ -200 ถึง 200

สคริปต์ของปลาตัวใหญ่

ปลาตัวใหญ่จะเคลื่อนที่ก็ต่อเมื่อผู้เล่นขยับเมาส์ มันจะว่ายน้ำตามเมาส์ และเมื่อเจอปลาทอง มัน ก็จะกินเจ้าปลาทองเท่านั้นเอง



สคริปต์แรกของปลาตัวใหญ่ เริ่มต้นเมื่อผู้ใช้คลิกธงเขียว มันจะเปลี่ยนชุดเป็น open-mouth อ้า ปากสีน้ำเงินคอยกินปลาทอง ทั้งนี้เพราะการกินปลาทอง ต้องเช็คว่าสีน้ำเงินสัมผัสสีสัมของปลา ทองหรือไม่ จากนั้นมันจะคอยตามเมาส์ตลอดเกมในบล็อกตลอดเวลาถ้า (forever if) โดย ตรวจสอบว่า ระยะตัวมันกับเมาส์ห่างกันเกินกว่า 10 หน่วยหรือไม่ (distance to mouse-pointer > 10) ถ้าใช่ ก็จะหันไปที่เมาส์ (point towards mouse-pointer) และเคลื่อนที่เข้าไปหา 3 หน่วย



สคริปต์ที่สองเริ่มเมื่อได้รับสาร got-me จากปลาทอง ขณะที่ปลาทองถูกกิน ปลาตัวใหญ่จะแสดง แอนิเมชั่นการกินด้วยการเปิดเสียง chomp (play sound chomp) จากนั้นขยับปาก 2 รอบใน บล็อกทำซ้ำ (repeat) โดยเปลี่ยนชุดปิดปาก closed-mouth แล้วคอย 0.3 วินาที ก่อนจะเปลี่ยน ชุดเปิดปาก open-mouth อีกครั้ง

การต่อยอดโปรเจก FishChomp

FishChomp เป็นเกมที่เข้าใจได้ง่าย เหมาะสำหรับผู้ที่ชื่นชอบเกม และอยากเขียนเกมตั้งแต่ เบื้องต้น ลองมาดูว่าอะไรบ้าง น่าจะทำให้เกม สนุก ท้าทายมากขึ้น

- เพิ่มปลาทอง หรือปลาชนิดอื่น
- ปลาทองออกมาตรงไหนก็ได้ของเวที และออกมาให้ไวขึ้น
- ปลาตัวใหญ่มีมิติในการเคลื่อนที่มากขึ้น เช่นสามารถหมุนได้ เคลื่อนที่เร็วขึ้น
- มีเพิ่มอีกระดับ (level 2) เพิ่มความท้าท้าย
- มีการควบคุมเกม
- มีคะแนน (score)
- มีเวลาจำกัดในการเล่นในแต่ละระดับ
- สื่อสารกับผู้เล่น
- แสดงวิธีเล่น และหายไประหว่างการเล่น
- บอกระดับ (level)
- บอกผู้เล่นเมื่อจบเกม และแสดงคะแนนต่างๆ ให้ดู

ลงมือต่อยอดเกม FishChomp

ระดับของเกม

เกมสามารถมีได้หลายระดับ แต่ละระดับมีเป้าหมายของตัวเอง มีความน่าสนใจและความท้าทาย ที่แตกต่างกัน สิ่งแวดล้อมในเกม (ตัวละครหรือภาพฉากหลัง) หรือกฎของเกม ในแต่ละระดับ ก็ อาจแตกต่างกันบ้าง เหมือนเดิมบ้างก็ได้

ปกติเกมควรเพิ่มความท้าทาย ในระดับที่สูงขึ้น การต่อยอดนี้มี 2 ระดับ โดยระดับแรกมีเพียง ปลาทองเหมือนกันหมด 4 ตัว ผู้เล่นเพียงไล่จับกินพวกมัน กินได้ 1 ตัว ได้ 1 คะแนน ส่วนระดับ ที่สอง ปลาทองตัวที่ 1 (มีหางสีน้ำเงิน) จะมีปลาสีเขียวเป็นเพื่อน คอยว่ายน้ำอยู่ใกล้ ๆ ถ้ากิน ปลาทองตัวที่ 1 จะได้โบนัส 2 คะแนน ส่วนปลาทองตัวอื่นได้ 1 คะแนนเหมือนเดิม แต่ถ้ากิน ปลาสีเขียว จะถูกหัก 3 คะแนน

การเพิ่มตัวละครในเกม

การเพิ่มปลาทองอีกซักตัวไม่ใช่เรื่องยาก เพียงคลิกขวาที่ตัวปลาทอง เลือกทำซ้ำ (duplicate) ก็ จะได้ปลาทองมาอีกตัวหนึ่งที่เหมือนกัน อย่างไรก็ตาม ถ้าปลาทองยังอยู่ระหว่างการแก้ไข ก็ไม่ ควรทำ นอกเสียจากว่าปลาทองที่ต้องการมีพฤติกรรมที่ต่างกันมาก และเพียงต้องการคุณสมบัติ บ้างประการของมันมา ซึ่งในการต่อยอดครั้งนี้มีทั้งสองแบบคือ

แบบที่หนึ่งต้องการปลาทองเหมือนกันอีกหนึ่งตัว โดยทำซ้ำปลาทองตัวที่ 3 ได้ตัวที่ 4 มา โดย ให้ทำซ้ำเมื่อสคริปต์ปลาทองตัวที่ 3 แก้ไขเสร็จเรียบร้อยแล้ว

แบบที่สองต้องการเพื่อนปลาทองตัวที่ 1 ใน level 2 เพื่อให้มันคอยว่ายน้ำอยู่ใกล้ๆ ปลาทองตัว ที่ 1 แต่ให้มันมีลักษณะคล้ายปลาตัวใหญ่ (เป็นลูกปลาตัวใหญ่) ดังนั้นจึงทำซ้ำปลาตัวใหญ่ แล้ว มาย่อขนาดให้ใกล้เคียงกับเหล่าปลาทอง แล้วเอาสคริปต์ของปลาทองมาเป็นต้นแบบ ปลาทองตัวที่ 1 ในระดับแรกเหมือนกับปลาทองตัวอื่นๆ จึงใช้ชุดเดิม แต่เพื่อให้ผู้เล่นแยกปลา ทองตัวที่ 1 ออกจากปลาทองตัวอื่นๆ ในระดับที่สอง ควรเพิ่มชุดอีกหนึ่งชุดสำหรับระดับที่สอง โดยปลาทองตัวที่ 1 จะมีหางเป็นสีน้ำเงินแทน ทำได้โดยทำซ้ำชุดเดิมก่อน จากนั้นคลิกปุ่ม edit เพื่อแก้ไขด้วยโปรแกรมวาดรูป



กฎของเกม

เพื่อเพิ่มมิติของเกม ควรแก้ไขเพิ่มเติมพฤติกรรมของตัวละครต่าง ๆ ในเกมบ้าง เช่น ปลาทอง ควรออกมาตรงไหนของเวทีก็ได้ ทำให้ผู้เล่นคาดเดาไม่ได้ ทำได้โดยเปลี่ยนค่าสุ่ม x และ y ใน บล็อกไปยัง (go to x: ... y: ...) ให้กระจายทั่วเวที นั้นก็คือ ค่า x อยู่ตั้งแต่ -240 ถึง 240 และค่า y อยู่ตั้งแต่ -180 ถึง 180 และเพื่อให้ปลาทองออกมาเร็วขึ้นหลังจากถูกกิน ทำได้โดยลดเวลา การคอย (wait) ลงจาก 3 มาอยู่ที่ 1 วินาที

การทำให้ปลาตัวใหญ่มีมิติในการเคลื่อนที่มากขึ้น คือสามารถหมุนได้เพื่อให้การไล่จับปลาทอง ง่ายขึ้น โดยการคลิกเลือกปุ่ม หมุนได้ (ปุ่มบน) บริเวณข้อมูลส่วนตัวของปลาตัวใหญ่ ซึ่งจาก เดิมทำได้แค่หันซ้ายขวา (ปุ่มกลาง) ส่วนการเคลื่อนที่ให้เร็วขึ้น ทำได้โดยเพิ่มระยะการเคลื่อนที่ (move) ให้มากขึ้น แต่ถ้ามากเกินอาจจะควบคุมเมาส์ไม่อยู่ ดังนั้นเพิ่มจาก 3 เป็น 5 ใน level 1 ส่วนใน level 2 มีเพื่อนปลาทองอยู่ด้วย และถ้ากินเพื่อนปลาทอง จะถูกหักแต้ม ดังนั้นไม่ควร เพิ่มความเร็วมากนัก ควรเพิ่มจาก 3 เป็น 4 ก็พอ



คะแนน

การนับคะแนนระหว่างเล่นเกมทำให้เกมดูท้าทายมากขึ้น ทำให้ผู้เล่นรู้ความสามารถของตัวเอง และสามารถเปรียบเทียบการเล่นกับผู้เล่นคนอื่นๆ ได้อีกด้วย

การใส่คะแนนบนเวที ทำได้โดยสร้างตัวแปรเพื่อเก็บคะแนนขึ้นมาก่อน เช่น score ถ้าต้องการ ให้คะแนนแสดงบนเวทีตลอดระหว่างการเล่นเกม ทำได้โดยคลิกบล็อกหน้าตัวแปรนั้น (จะเห็น เครื่องหมายถูก) นอกจากนั้นยังสามารถปรับตำแหน่งตัวแปรบนเวทีได้ด้วย โดยใช้เมาส์ลากตัว แปรไปยังตำแหน่งที่ต้องการ ตอนออกแบบโปรแกรม

การปรับเปลี่ยนคะแนน ขึ้นกับกติกาที่กำหนดไว้ นั้นคือ ในระดับแรก ถ้ากินปลาทองได้ 1 ตัว จะได้ 1 คะแนน ส่วนระดับที่สอง ถ้ากินปลาทองตัวที่ 1 จะได้ 2 คะแนน ส่วนปลาทองตัวอื่นได้ 1 คะแนนเหมือนเดิม แต่ถ้ากินปลาสีเขียว จะถูกหัก 3 คะแนน

การจับเวลา

ถ้าไม่มีเวลาจำกัดในการเล่นแต่ละระดับ ก็จะไม่รู้ว่าเกมจะจบเมื่อใด ยกเว้นแต่ว่าในแต่ละระดับมี เงื่อนไขอื่นกำหนด หรือมีเป้าหมายให้ทำ เช่น กินปลาให้ได้ 20 ตัว เป็นตัน การต่อยอดคราวนี้ กำหนดให้แต่ละระดับมีเวลาไม่เท่ากัน ระดับแรกง่ายหน่อย ใช้ 20 วินาที ระดับสองยากขึ้นมา นิดหนึ่งเพิ่มเวลาเป็น 30 วินาที



ใน Scratch ทำนาพิกาจับเวลาได้ง่าย ๆ โดยสร้างตัวแปร time ขึ้นมา แล้วตั้งค่าเวลาที่ต้องการ ให้มัน จากนั้นนับเวลาถอยหลังที่ละ 1 วินาที จนกระทั่งเป็น 0 เช่น ในระดับแรก ตั้งเวลาเป็น 20 วินาที จากนั้นตรวจสอบว่าตัวแปร time ยังมากกว่า 0 หรือไม่ ถ้าใช่ จับเวลา 1 วินาที ด้วย บล็อก wait จากนั้นลดค่าตัวแปร time ที่ละ 1 ทำอย่างนี้วนไปเรื่อย ๆ ในบล็อก forever จนกว่า time = 0 ด้วยวิธีนี้จะทำให้รู้ว่าเวลาในแต่ละระดับหมดหรือยัง ถ้าเวลาหมดลงแล้ว เกมจะดำเนิน ต่อในส่วน else เพื่อเล่นเกมในระดับต่อไป หรือจะแสดงผลการเล่นก่อนจะจบเกม

การสื่อสารกับผู้เล่น

การสื่อสารกับผู้เล่นนั้นก็สำคัญ เช่น ตั้งแต่ต้นเกมต้องบอกผู้เล่นว่าเล่นอย่างไร และระหว่างเล่น วิธีการเล่นนั้นต้องหายไป มิฉะนั้นจะรบกวนการเล่น ก่อนเข้าแต่ละระดับก็ควรบอกว่าขณะนี้จะ เล่นในระดับใด และเมื่อจบเกมก็ควรมีคะแนน หรือข้อมูลอื่นๆ ให้ดู เช่น จำนวนปลาชนิดต่างๆ ที่กินได้



โปรเจก Delicious Fish เป็นการต่อยอดโปรเจก FishChomp ที่ผู้เขียนลองทำขึ้นมา และได้ upload โปรเจกนี้ที่เว็บไซต์ http://scratch.mit.edu/projects/kidsangsan/1889679 ผู้อ่าน สามารถเข้าไปเล่นได้ หรือดาวน์โหลดเกม มาศึกษาแก้ไขปรับปรุงเพิ่มเติมได้





<u>kidsangsan</u> shared it 1 day, 23 hours ago Based on <u>SampleProject</u>...'s <u>project</u>

การใช้งานบล็อกคำสั่ง

บล็อกหมวดการเคลื่อนที่ (Motion)

| move (10) steps | เคลื่อนตัวละครไปข้างหน้าหรือถอยหลัง (ใส่เลขค่าลบ) |
|---------------------------|---|
| turn 🗘 15 degrees | หมุนตัวละครตามเข็มนาพิกา |
| turn 🐧 🔝 degrees | หมุนตัวละครทวนเข็มนาพิกา |
| point in direction 90 V | กำหนดทิศของตัวละครไปยัง (0 = บน, 90 = ขวา, 180 = ล่าง, -90 = ซ้าย) |
| point towards | กำหนดทิศของตัวละครไปยัง (เมาส์หรือตัวละครอื่น) |
| go to x: 0 y: 0 | เคลื่อนตัวละครไปยังตำแหน่ง x และ y บนเวที |
| go to | เคลื่อนตัวละครไปยังตำแหน่งของ (เมาส์หรือตัวละครอื่น) |
| glide 1 secs to x: 0 y: 0 | เคลื่อนตัวละครไปยังตำแหน่ง ด้วยระยะเวลาที่ถูกกำหนด |
| change x by 🔟 | เปลี่ยนตำแหน่ง x ของตัวละคร ตามระยะที่ถูกกำหนด |
| set x to O | ตั้งค่าตำแหน่ง x ของตัวละคร ตามค่าที่ถูกกำหนด |
| change y by 10 | เปลี่ยนตำแหน่ง y ของตัวละคร ตามระยะที่ถูกกำหนด |
| set y to O | ตั้งค่าตำแหน่ง y ของตัวละคร ตามค่าที่ถูกกำหนด |

| if on edge, bounce | กลับทิศของตัวละคร เมื่อตัวละครชนขอบเวที |
|--------------------|---|
| | |

บล็อกตัวแปร

บล็อกที่มาพร้อมกับกล่องเลือกด้านหน้า ถ้าคลิกที่กล่องเลือก จะแสดงค่าที่คืนมาจากบล็อกนั้น บนเวที

| x position | คืนค่าตำแหน่ง x ของตัวละคร (-240 to 240) |
|------------|--|
| y position | คืนค่าตำแหน่ง y ของตัวละคร (-180 to 180) |
| direction | ดื่นค่าทิศทางการเคลื่อนที่ของตัวละคร |

บล็อกสำหรับมอเตอร์

| motor on for 🔵 secs | เปิดมอเตอร์เป็นเวลาตามที่กำหนด |
|---------------------|--|
| motor on | เปิดมอเตอร์ |
| motor off | ปิดมอเตอร์ |
| motor power | ตั้งค่ากำลังของมอเตอร์ แล้วเปิดมอเตอร์ (ค่ากำลังของมอเตอร์อยู่ ตั้งแต่ 0 ถึง 100) |
| motor direction | ตั้งค่าหรือเปลี่ยนทิศการหมุนของมอเตอร์ โดยไม่ต้องเปิดมอเตอร์ (this way = ตามเข็มนาฬิกา, that way = ทวนเข็มนาฬิกา, reverse = เปลี่ยนทิศการหมุน) |

บล็อกหมวดรูปร่าง (Looks)

| switch to costume costume1 | เปลี่ยนหน้าตาตัวละคร ด้วยการเปลี่ยนชุดละคร |
|----------------------------------|---|
| next costume | เปลี่ยนชุดตัวละครถัดไป ในรายการชุดตัวละคร (ถ้าถึงชุด สุดท้ายในรายการ จะกลับไปเริ่มที่ชุดแรกใหม่) |
| switch to background background1 | เปลี่ยนหน้าตาเวที ด้วยการเปลี่ยนภาพฉากหลัง |
| next background | เปลี่ยนภาพฉากหลังถัดไป ในรายการภาพฉากหลัง (ถ้าถึงชุด สุดท้ายในรายการ จะกลับไปเริ่มที่ชุดแรกใหม่) |

บล็อกแสดงความคิดหรือพูด

| say Hello! for 2 secs | แสดงลูกโป่งคำพูดของตัวละคร ในระยะเวลาที่ถูกกำหนด |
|-----------------------|---|
| say Hello! | แสดงลูกโป่งคำพูดของตัวละคร (ลบลูกโป่งคำพูดได้ด้วยการรัน บล็อกนี้อีกครั้งโดยไม่ใส่ตัวหนังสือ) |
| think Hmm. for 2 secs | แสดงลูกโป่งความคิดของตัวละคร ในระยะเวลาที่ถูกกำหนด |
| think Hmm. | แสดงลูกโป่งความคิดของตัวละคร |

บล็อกเลเยอร์

| go to front | ย้ายตัวละครไปยังด้านหน้าของตัวละครอื่นๆ |
|------------------|--|
| go back 1 layers | ย้ายตัวละครกลับไปยังเลเยอร์ที่ถูกกำหนด เพื่อว่ามันจะสามารถ ซ่อนอยู่ด้านหลังตัวละครอื่นๆ ได้ |

บล็อกแอฟเฟ็ค

| change color veffect by (25) | เปลี่ยนเอฟเฟคบนตัวละคร ตามค่าที่ถูกกำหนด (ใช้รายการ |
|------------------------------|--|
| | เลือกแบบดึงลงเพื่อเลือกเอฟเฟค) |
| | |
| set color v effect to 0 | ตั้งค่าเอฟเฟคตามค่าที่กำหนด (ค่าเอฟเฟคอยู่ตั้งแต่ -100 ถึง |
| | 100 ค่า 0 เป็นค่าปกติที่ไม่มีเอฟเฟค บางเอฟเฟคค่าลบกับค่า |
| | บวกให้ผลเหมือนกัน) |
| | |
| clear graphic effects | ล้างเอฟเฟคทั้งหมดของตัวละคร |
| | |
| change size by 10 | เปลี่ยนขนาดของตัวละคร ด้วยค่าที่ถูกกำหนด |
| | |
| set size to 100 % | ตั้งค่าขนาดของตัวละครเป็น % ของขนาดเริ่มต้น |
| | |
| show | ทำให้ตัวละครปรากฏบนเวที |
| | |
| hide | ทำให้ตัวละครหายไปจากเวที (เมื่อตัวละครหายไป ตัวละครอื่นๆ |
| - | จะไม่สามารถตรวจจับด้วยการใช้บล็อกสัมผัสได้) |
| | |
| 1 | |

บล็อกตัวแปร

บล็อกที่มาพร้อมกับกล่องเลือกด้านหน้า ถ้าคลิกที่กล่องเลือกจะแสดงค่าที่คืนมาจากบล็อกนั้น บนเวที

| costume #) | ดื่นหมายเลขชุดละครปัจจุบัน |
|--------------|--|
| background # | คืนหมายเลขภาพฉากหลังป [ั] จจุบัน |
| size | คืนค่าขนาดของตัวละครเป็น % ของขนาดเริ่มต้น |

บล็อกหมวดเสียง (Sound)

| play sound meow | เริ่มเปิดเสียงจากรายการเลือกแบบดึงลง แล้วรันบล็อกถัดไปแม้ |
|---------------------------------|---|
| | เสียงยังเล่นอยู่ |
| | |
| play sound meow until done | เปิดเสียงแล้วรอจนเสียงหยุด ก่อนที่จะรันบล็อกถัดไป |
| stop all sounds | ปิดเสียงทั้งหมด |
| play drum (48 🗸 for (0.2) beats | เล่นเสียงกลองจากรายการเลือกแบบดึงลง ด้วยจังหวะที่กำหนด |
| play note 60 ▼ for 0.5 beats | เล่นโน๊ตดนตรีตามจังหวะที่กำหนด (ตัวเลขยิ่งสูง จะได้เสียงสูง |
| | ຕ າມ) |
| | |
| rest for 02 beats | หยุดพักเสียงตามจังหวะที่กำหนด |
| set instrument to 🚺 | เลือกชนิดเครื่องดนตรีให้ตัวละคร สำหรับใช้ในโน็ตบล็อก (แต่ละ |
| _ | ตัวละครมีเครื่องดนตรีของตัวเอง) |
| | |

| change volume by -10 | ปรับระดับเสียงของตัวละครตามค่าที่กำหนด ระดับเสียงอยู่ |
|----------------------|---|
| _ | ระหว่าง 0 ถึง 100 |
| | |
| set volume to 100 % | ตั้งค่าระดับเสียงของตัวละคร ด้วยค่าที่กำหนด |
| change tempo by 20 | เปลี่ยนจังหวะเสียงของตัวละคร ด้วยค่าที่กำหนด |
| set tempo to 60 bpm | ตั้งค่าจังหวะเสียงของตัวละคร ด้วยค่าที่กำหนด |

บล็อกตัวแปร

บล็อกที่มาพร้อมกับกล่องเลือกด้านหน้า ถ้าคลิกที่กล่องเลือกจะแสดงค่าที่คืนมาจากบล็อกนั้น บนเวที

| volume) | คืนค่าระดับเสียงของตัวละคร |
|---------|-----------------------------|
| tempo | คืนค่าจังหวะเสียงของตัวละคร |

บล็อกหมวดปากกา (Pen)

| clear | ลบรอยวาดเขียนบนเวที จากการใช้ปากกาหรือแสตมป์ |
|------------------------|--|
| pen down | จรดปากกาของตัวละคร เพื่อวาดเขียนขณะที่มันเคลื่อนที่ |
| pen up | ยกปากกาของตัวละครขึ้น เพื่อไม่ให้วาดเขียนขณะที่มันเคลื่อนที่ |
| set pen color to | ตั้งค่าสีให้ปากกา ตามค่าสีจากตัวเลือกสี ซึ่งจะทำให้ระดับสีเปลี่ยน ด้วย |
| change pen color by 10 | เปลี่ยนค่าสี ตามค่าที่กำหนด |
| set pen color to 0 | ตั้งค่าสี ตามค่าที่กำหนด (ค่าระหว่าง 0 ถึง 200 คือสีในวงล้อของ แถบสีรุ้ง เช่น pen_color = 0 คือสีแดง, pen_color = 100 คือสีน้ำ เงิน) |
| change pen shade by 10 | เปลี่ยนระดับสี ตามค่าที่กำหนด |
| set pen shade to 50 | ตั้งค่าระดับสีตามค่าที่กำหนด (pen_shade = 0 คือค่อนข้างมืด pen_shade = 100 คือค่อนข้างสว่าง ค่าที่ถูกตั้งเริ่มแรก คือ 50 เมื่อใช้ตัวเลือกสี จะได้ระดับสีจากสีที่เลือกมาด้วย) |
| change pen size by () | เปลี่ยนขนาดของเส้นปากกา ตามค่าที่กำหนด |
| set pen size to 1 | ตั้งค่าขนาดของเส้นปากกา ตามค่าที่กำหนด |
| stamp | ประทับภาพของตัวละครบนเวที |

บล็อกหมวดควบคุม (Control)

บล็อกเริ่มต้นทำงาน

| when 🛤 clicked | รันสคริปต์ เมื่อธงเขียวถูกคลิก |
|------------------------|--------------------------------------|
| when space key pressed | รันสคริปต์ เมื่อคีย์ที่กำหนดไว้ถูกกด |
| when Sprite1 clicked | รันสคริปต์ เมื่อตัวละครถูกคลิก |
| when I receive | รันสคริปต์ เมื่อได้รับสารที่สนใจ |

บล็อกทำซ้ำ

| forever | รันบล็อกที่อยู่ภายใน ซ้ำ ๆ ไปตลอด |
|-----------|---|
| repeat 10 | รันบล็อกที่อยู่ภายใน ตามจำนวนครัั้งที่กำหนด |

บล็อกเงื่อนไข

| | ตรวจสอบเงื่อนไข ถ้าเป็นจริง ทำงานบล็อกที่อยู่ภายใน |
|------|---|
| if | ตรวจสองเงื่อนไข ถ้าเป็นจริงทำงานบล็อกที่อยู่ภายในส่วนแรก (ใต้ |
| else | if) ถ้าเป็นเท็จทำงานบล็อกที่อยู่ภายในส่วนที่สอง (ใต้ else) |

บล็อกทำซ้ำแบบมีเงื่อนไข

| forever if | ตรวจสอบเงื่อนไขตลอดเวลาว่าเป็นจริงหรือไม่ ถ้าเป็นจริง ทำงาน บล็อกที่อยู่ภายใน |
|--------------|--|
| repeat until | ตรวจสอบเงื่อนไขตลอดเวลา ถ้ายังเป็นเท็จ ทำงานซ้ำบล็อกที่อยู่ ภายใน ถ้าเป็นจริง ทำงานบล็อกถัดไป |

บล็อกส่งสาร

| broadcast | ส่งกระจายสารถึงทุกตัวละครและเวที จากนั้นทำงานบล็อกถัดไป โดยไม่ต้องรอสคริปต์ที่ถูกกระตุ้นจากสารนี้ |
|----------------------|---|
| broadcast 💽 and wait | ส่งกระจายสารถึงทุกตัวละครและเวที กระตุ้นสคริปต์ที่รับสารนี้ให้ ทำงาน รอจนกระทั่งสคริปต์นั้นๆ ทำงานจนเสร็จ แล้วจึงทำงาน บล็อกถัดไป |

บล็อกหยุดการทำงาน

| wait 🚺 secs | หยุดรอคอยตามเวลาที่กำหนด จากนั้นทำงานบล็อกถัดไป |
|--------------|---|
| wait until 🔿 | หยุดรอคอยจนกระทั่งเงื่อนไขเป็นจริง จึงจะทำงานบล็อกถัดไป |
| stop script | หยุดทำงานสคริปต์ |
| stop all | หยุดทำงานสคริปต์ทั้งหมดของทุกตัวละคร |

บล็อกหมวดกำลังรับรู้ (Sensing)

| mouse down? | คืนค่าจริง ถ้าปุ่มเมาส์ถูกกด |
|-----------------------|--|
| key space pressed? | ดืนค่าจริง ถ้าคีย์ที่กำหนดถูกกด |
| touching ? | ดืนค่าจริง ถ้าตัวละครสัมผัสกับตัวละครที่กำหนด ขอบเวที หรือตัว ชี้เมาส์ (เลือกจากรายการตัวเลือกแบบดึงลง) |
| touching color ? | คืนค่าจริง ถ้าตัวละครสัมผัสกับสีที่กำหนด (คลิกที่แถบสี จากนั้น ใช้หลอดหยดเลือกสี) |
| color is touching ? | คืนค่าจริง ถ้าสีแรกที่อยู่ในตัวละครสัมผัสกับสีที่สองที่อยู่ในฉาก หลังหรือตัวละครอื่น |
| mouse x | ดื่นค่าตำแหน่ง x ของตัวชี้เมาส์ |
| mouse y | ดื่นค่าตำแหน่ง y ของตัวชี้เมาส์ |
| distance to | ดื่นค่าระยะทาง ระหว่างตัวละครกับตัวชี้เมาส์หรือตัวละครอื่น |
| x position of Sprite1 | ดื่นค่าสมบัติของตัวละครอื่น |
| reset timer | ตั้งค่าเริ่มต้นใหม่ให้นาพิกาจับเวลาที่ศูนย์วินาที |

บล็อกถามตอบ

| ask and wait | ถามคำถามบนเวที และรับคำตอบจากคีย์บอร์ดเก็บไว้ในตัวแปร (answer) ทำให้โปรแกรมต้องหยุดคอย จนกว่าผู้ใช้กดคีย์ Enter หรือคลิก เครื่องหมายถูก |
|---------------------|---|
| answer | ดืนค่าคำตอบที่อยู่ในตัวแปร answer ซึ่งทุกตัวละครสามารถนำไปใช้ได้ ถ้าคลิกที่กล่องด้านหน้า จะแสดงค่าที่คืนมาจากบล็อกบนเวที |

บล็อกตัวแปร

บล็อกที่มาพร้อมกับกล่องเลือกด้านหน้า ถ้าคลิกที่กล่องเลือกจะแสดงค่าที่คืนมาจากบล็อกนั้น บนเวที

| T timer | คืนค่าเวลาเป็นวินาที (นาพิกาจับเวลาจะทำงานตลอดเวลา) |
|---------------------------|---|
| loudness | คืนค่าระดับเสียง (จาก 1 ถึง 100) ที่ไมโครโพนของคอมพิวเตอร์ ตรวจวัดได้ |
| loud? | คืนค่าจริง ถ้าไมโครโฟนของคอมพิวเตอร์ตรวจจับระดับเสียงที่ มากกว่า 30 (ในสเกล 1 ถึง 100) |
| sensor value | ้คืนค่าจากตัวรับรู้ที่กำหนดไว้ จะใช้บล็อกนี้ได้ต้องต่ออุปกรณ์รับรู้ กับคอมพิวเตอร์ก่อน เช่น บอร์ดตัวรับรู้ของ Scratch (http://www.playfulinvention.com/picoboard.html) หรือ LEGO® WeDo™ (<u>http://www.legoeducation.com</u>). |
| sensor button pressed > ? | คืนค่าจริง ถ้าตัวรับรู้ที่ระบุถูกกด จะใช้บล็อกนี้ได้ต้องมีบอร์ดตัวรับ รู้ต่อกับคอมพิวเตอร์ก่อน (<u>http://www.playfulinvention.com/picoboard.html</u>) |

บล็อกหมวดโอเปอร์เรเตอร์ (Operators)

บล็อกจัดการข้อความ (สตริง)

| join 🔳 | ต่อข้อความทั้งสองข้อความเข้าด้วยกัน |
|---------------|---------------------------------------|
| length of | ดื่นค่าจำนวนตัวอักษรในข้อความ |
| letter 🔵 of 📘 | ลืนตัวอักษรในข้อความ ณ ตำแหน่งที่ระบุ |

บล็อกจัดการตัวเลข
181

บล็อกจัดการตรรกะ

| | ดื่นค่าจริง ถ้าค่าตัวเลขแรกน้อยกว่าค่าตัวเลขที่สอง |
|-----|---|
| | ดื่นค่าจริง ถ้าค่าตัวเลขทั้งคู่มีค่าเท่ากัน |
| | ดื่นค่าจริง ถ้าค่าตัวเลขแรกมากกว่าค่าตัวเลขที่สอง |
| and | ้ คืนค่าจริง ถ้าเงื่อนไขทั้งคู่เป็นจริง คืนค่าเท็จ ถ้าเงื่อนไขใดเงื่อนไข หนึ่งเป็นเท็จ |
| or | ดืนค่าจริง ถ้าเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริง คืนค่าเท็จ ถ้าเงื่อนไขทั้ง คู่เป็นเท็จ |
| not | คืนค่าจริง ถ้าเงื่อนไขเป็นเท็จ คืนค่าเท็จ ถ้าเงื่อนไขเป็นจริง |

บล็อกหมวดตัวแปร (Variables)

บล็อกตัวแปร

| Make a variable | คลิกปุ่มเพื่อสร้างตัวแปร พร้อมทั้งกำหนดชื่อให้ตัวแปรใหม่ และ |
|-----------------------|---|
| | ยังสามารถระบุได้ว่า จะให้ทุกตัวละครใช้ได้ (global) หรือใช้กับ |
| | เฉพาะตัวละคร (local) ที่สร้างตัวแปรนี้ จากนั้นบล็อกตัวแปร |
| | ใหม่จะปรากฏ |
| | |
| Delete a variable | ลบบล็อกทั้งหมด ที่เกี่ยวของกับตัวแปร |
| score | คืนค่าที่เก็บไว้ในตัวแปร (score ใน ณ ที่นี้) |
| change score v by 1 | เปลี่ยนค่าในตัวแปรตามปริมาณที่กำหนด ถ้ามีตัวแปรมากกว่า หนึ่งตัว ใช้รายการเลือกแบบดึงลงเพื่อเลือกตัวแปรก่อน |
| set score to 0 | ตั้งค่าตัวแปร ด้วยค่าที่กำหนด |
| show variable score v | แสดงค่าตัวแปรบนเวที |
| hide variable score | ซ่อนค่าตัวแปร ไม่ให้แสดงบนเวที |

บล็อกตัวแปรลิสต์

| | سيبقاد لأفريه محرم بواحد |
|--|---|
| Make a list | คลิกปุ่มเพื่อสร้างรายการลิสต์ พร้อมทั้งกำหนดชื่อให้ลิสต์ ใหม่ และยังสามารถระบุได้ว่า จะให้ทุกตัวละครใช้ได้ (global) หรือใช้กับเฉพาะตัวละคร (local) ที่สร้างตัวแปรนี้ |
| | จากนั้นบล็อกลิสต์ใหม่จะปรากฏ |
| Delete a list | ลบบล็อกทั้งหมดที่เกี่ยวข้องกับลิสต์ |
| my list | คืนค่าทุกรายการที่อยู่ในลิสต์ (โดยที่แต่ละรายการจะถูก แยกด้วยช่องว่าง อย่างไรก็ตาม ถ้าแต่ละรายการเป็น ตัวอักษรเดี่ยวหรือตัวเลขโดด ช่องว่างจะถูกละไว้) |
| delete 💽 of my list | ลบหนึ่งรายการหรือรายการทั้งหมดจากลิสต์ โดยสามารถ ระบุตำแหน่งรายการที่จะลบได้ จากรายการเลือกแบบดึง ลง หรือจะพิมพ์ตัวเลขตำแหน่งโดยตรง เลือก last เพื่อลบ รายการท้ายสุดของลิสต์ เลือก all เพื่อลบรายการทั้งหมด ในลิสต์ การลบรายการจะลดขนาดของลิสต์ลง |
| insert thing at 🚺 of my list | แทรกรายการไปยังตำแหน่งที่ระบุในลิสต์ โดยสามารถระบุ ตำแหน่งรายการได้ จากรายการเลือกแบบดึงลง หรือจะ พิมพ์ตัวเลขตำแหน่งโดยตรง เลือก any เพื่อแทรกรายการ ณ ตำแหน่งที่ได้จากการสุ่ม เลือก last เพิ่มรายการที่ ท้ายสุดของลิสต์ การแทรกรายการจะเพิ่มขนาดของลิสต์ |
| replace item 1 of <u>my list</u> with thing | แทนที่รายการในลิสต์ด้วยค่าที่กำหนด โดยสามารถระบุ ตำแหน่งรายการได้ จากรายการเลือกแบบดึงลง หรือพิมพ์ ตัวเลขตำแหน่งโดยตรง เลือก any เพื่อแทนที่รายการ ณ ตำแหน่งที่ได้จากการสุ่ม ขนาดของลิสต์ไม่เปลี่ยนแปลง จากการแทนที่รายการ |

| add thing to my list | เพิ่มรายการที่กำหนดต่อท้ายลิสต์ โดยที่รายการสามารถ เป็นตัวเลขหรือข้อความก็ได้ |
|--------------------------|--|
| item 1 of my list | ดืนค่าในรายการ ณ ตำแหน่งที่ระบุในลิสต์ โดยสามารถ ระบุตำแหน่งรายการได้ จากรายการเลือกแบบดึงลง หรือ พิมพ์ตัวเลขตำแหน่งโดยตรง เลือก any เพื่อคืนค่าใน รายการ ณ ตำแหน่งที่ได้จากการสุ่ม |
| length of my list | คืนค่าจำนวนรายการทั้งหมดในลิสต์ (ขนาดของลิสต์) |
| list contains thing | คืนค่าจริง ถ้าลิสต์มีค่าในรายการกับค่าที่กำหนดไว้ เหมือน ตรงกันทั้งหมด |

หมายเหตุ บล็อกที่มาพร้อมกับกล่องเลือกด้านหน้า ถ้าคลิกที่กล่องเลือกจะแสดงค่าที่คืนมาจาก บล็อกนั้นบนเวที

แหล่งอ้างอิง

ผู้เขียนได้เขียนบทความลงในบล็อกคิดสร้างสรรค์ <u>http://kidsangsan.com/</u> ของผู้เขียนเอง ซึ่ง เกี่ยวกับความคิดสร้างสรรค์ การศึกษา และการเขียนโปรแกรม โดยเริ่มต้นเขียนเกี่ยวกับ โปรแกรม Scratch อย่างจริงจังก่อน เพื่อเป็นแหล่งเรียนรู้ สำหรับการเขียนโปรแกรมสำหรับ เด็ก ๆ จากนั้นนำมารวบรวม และเรียบเรียงใหม่อีกครั้งในหนังสือเล่มนี้

ข้อมูลที่ใช้ในหนังสือเล่มนี้ยังมีแหล่งอ้างอิงหลัก ๆ จากเว็บไซต์ Scratch <u>http://scratch.mit.edu/</u> ดู่มือการใช้งาน Scratch <u>http://info.scratch.mit.edu/Support/Reference_Guide_1.4</u> และโปร เจกที่ใช้ในหนังสือเล่มนี้ เป็นโปรเจกตัวอย่างที่มาพร้อมกับโปรแกรม Scratch v1.4