



ู Code Their Dreams ปั้นโปรแกรมเมอร์รุ่นเยาว์

คู่มือเรียนรู้การเงียนโปรแกรมด้วย





สารบัญ

หัวข้อ	หน้า
● เกี่ยวกับ Scratch	1
 ส่วนประกอบของ Scratch 	1
 การกำหนดคำสั่งให้กับเหตุการณ์ 	26
 การกำหนดตัวแปรและการคำนวณ 	39
 การสร้างเงื่อนไขและการวนซ้ำ 	53
 การออกแบบและสร้างโปรแกรม 	70



เกี่ยวกับ Scratch

Scratch ทำให้เราสามารถเขียนโปรแกรมที่สามารถตอบโต้ได้ในหลายรูปแบบ ทั้งการสร้างเรื่องราว เกมส์ แอนิเมชั่น และอื่นๆอีกมากมาย ผ่านลิงค์นี้ http://scratch.mit.edu และสามารถแชร์สิ่งที่เราสร้างขึ้นมากับเพื่อนๆในสังคม ออนไลน์ของเราได้

Scratch ทำให้เราได้คิดอย่างสร้างสรรค์ มีเหตุผล เป็นระบบ และสามารถ ทำงานร่วมกันได้ดีขึ้น ซึ่งเป็นทักษะที่ควรจะมีในศตวรรษที่ 21

Scratch เป็นโครงการของ Lifelong Kindergarten Group ที่ MIT Media Lab ซึ่งเปิดให้คนทั่วไปใช้ได้โดยไม่เสียค่าใช้จ่าย



ส่วนประกอบของ Scratch

- 1. Menu Bar แถบเมนูด้านบนสุดของหน้าต่าง Scratch ประกอบไปด้วย
 - 💮 สำหรับเปลี่ยนภาษา โดยโปรแกรม Scratch มีภาษาไทย ให้ เลือกได้ด้วย



- เมนู File ประกอบไปด้วยเครื่องมือย่อย
 - "New" ใช้สำหรับสร้างไฟล์ Scratch ใหม่
 - "Upload from your computer" สำหรับอัพโหลด Scratch จาก คอมพิวเตอร์ของเรา
 - "Download to your computer" สำหรับดาวน์ โหลด Scratch ลงเครื่องของตัวเอง

File	T	Edit 🔻	Tips	About
	New			
	Uplo	oad from	your c	omputer
	Dov	vnload to) your c	omputer
	Reve	ert		

- "Revert" สำหรับย้อนไฟล์ปัจจุบันไปที่จุดเริ่มต้น



เมนู Edit ประกอบด้วยเครื่องมือย่อย

- "Undelete" ใช้สำหรับการนำสิ่งที่เพิ่งลบไปกลับมา คล้ายๆคำสั่ง Undo
- "Small stage layout" ใช้สำหรับการย่อหน้าต่าง Stage ให้เล็ก ลง เพื่อทำให้หน้าต่าง Script Area ใหญ่ขึ้น เห็น Block ที่เรา สร้างได้กว้างขึ้น ซึ่งเราสามารถเลือกที่เครื่องมือนี้อีกครั้งเพื่อทำ ให้หน้าต่าง Stage กลับมามีขนาดเท่าเดิม
- "Turbo mode" ใช้สำหรับการแสดงผลโปรแกรมเร็วขึ้น จากการ ลดระยะเวลาหน่วงระหว่าง Block ลง เราสามารถทำให้กลับสู่ โหมดปกติโดยกดที่ปุ่มนี้เหมือนเดิม
- Tips ใช้สำหรับเรียกดูหน้าจอทางด้านขวา สำหรับเรียนรู้วิธีการใช้
 Scratch ด้วยตนเอง step-by-step



- About ใช้สำหรับไปที่หน้าจอ about ของ Scratch (<u>https://scratch.mit.edu/about/</u>)
- 3. Sprite List

เป็นส่วนสำหรับการเก็บตัวละคร ชิ้นส่วนต่างๆในโปรแกรมของเรา โดยเรา สามารถเลือกเพิ่มตัวละครและชิ้นส่วนต่างๆรวมถึงฉากหลังได้จากส่วนนี้



- ส่วนสำหรับการแสดงฉากหลังปัจจุบัน การเพิ่มฉากหลัง และการ set ฉาก หลังใหม่
- ส่วนนี้เราสามารถเพิ่มฉากหลังใหม่ได้ 4 วิธี ได้แก่ การเลือกจากรูปที่มีมาให้ จากโปรแกรม การวาดเองโดยโปรแกรมการวาดสำเร็จรูปที่มีให้ การเลือก จากรูปในเครื่องของเรา และการถ่ายรูปผ่านกล้องทันที



- ส่วนตัวละครและชิ้นส่วนต่างๆของโปรแกรมที่ถูกเพิ่มเข้ามาจะมาอยู่ตรง ส่วนนี้
- ส่วนเพิ่มตัวละครใหม่สามารถทำได้ 4 วิธี ได้แก่ การเลือกตัวละครจากที่มี อยู่ในโปรแกรม การวาดเองโดยใช้โปรแกรมสำเร็จรูปที่มีให้ การเลือกตัว ละครจากภาพในเครื่องของเรา และการถ่ายภาพตัวละครผ่านกล้องเรา ทันที

4. Blocks Palette

เป็นส่วนของคำสั่งที่เราจะนำไปประกอบกันเป็นโปรแกรมแบ่งออกเป็น หมวดหมู่ต่างๆตามประเภทการใช้งาน ประกอบด้วย 2 ส่วนใหญ่ๆ คือ ส่วน หมวดหมู่ด้านบน และส่วนตัวคำสั่งด้านล่าง



สคริปต์	Costumes	เสียง
การเคอี รูปร่าง เสียง ปากกา ข้อมูล	อ <mark>นที่</mark> เหตุก ควบห กำลัง โอเป More	าารณ์ คุม งรับรู้ lอร์เรเตอร์ e Blocks
เคลือน 10 หมุน (*) หมุน (*) หมนน (*) หมูน (*) หมนน (*) หมูน (*) หมนน (*) หมูน (*) หมนน (*) หมน (*) หม) กัาว 15 ธงศา 15 ธงศา 15 ธงศา 15 ธงศา 15 ธงศา 15 ธงศา 10 γ: 0 3/31/11 v 10 γ: 0 11 0	Ο γ: Ο
สั้งรุงโนงกาก	ารหมุ่น ด้าย-ขวา	

ส่วนด้านบน แบ่งออกเป็น 3 เมนู ได้แก่

 สคริปต์ ใช้เลือกหาคำสั่งเพื่อนำมาต่อกันเป็นโปรแกรม แบ่งออกเป็น 10 หมวดด้วยกัน แต่ละหมวดก็จะมี Block คำสั่ง การกระทำที่แตกต่างกัน ดังนี้



สคริปต์	Costumes		เสียง
การเคลื่	้อนที่	เหตุก	ารณ์
รูปร่าง		ควบค	าม
เสียง		กำลัง	เร้บรู้
ปากกา		โอเป	ວຣ໌ເຣເທວຣ໌
ข้อมูล		More	Blocks

 <u>หมวดการเคลื่อนที่</u> เราสามารถทำให้ตัวละครเคลื่อนไหวได้ตามที่เรา ต้องการ ประกอบไปด้วย







 <u>หมวดการรูปร่าง</u> เราสามารถเปลี่ยนแปลงตัวละครและฉากให้เหมาะกับ สถานการณ์ได้ ประกอบด้วย





 <u>หมวดเสียง</u> ทำให้เราเล่นเสียงได้ตามประเภทและจังหวะที่เราต้องการ ประกอบด้วย







4. <u>หมวดปากกา</u> สามารถวาดเส้นตามการเคลื่อนที่ของตัวละคร



 หมวดข้อมูล มีการเก็บข้อมูลและการเรียกใช้ข้อมูล มี 2 รูปแบบ ประกอบด้วย

สร้างตัวแปร	ตัวแปรใหม่	
สร้างรายการ	ชื่อตัวแปร:	
	ตกลง ยกเลิก	
รายการใหม่		
รายการชื่อ:		
● สำหรับตัวละครทั้งหมด 🔿 สำหรับตัวละครนี้เท่านั้น		
ตกลง ยกเลิก		

 การสร้างตัวแปร คือตัวเก็บค่าที่เราสามารถเรียกมาใช้ได้ตลอด สมมุติเราสร้างตัวแปรชื่อ x





การสร้างรายการ ใช้สำหรับการต้องการเก็บค่าหลายๆค่าเป็นลำดับ สมมุติให้ชื่อรายการ L





<u>หมวดเหตุการณ์</u> เป็นหมวดที่เกี่ยวข้องกับการกระทำเมื่อมีเหตุการณ์ ตามที่กำหนด ประกอบด้วย





 หมวดควบคุม ช่วยกำหนดการไหลของคำสั่งให้ทำคำสั่งใดตอนไหน ทำซ้ำ หรือ ไม่ทำเมื่อไหร่ ประกอบด้วย





8. <u>หมวดกำลังรับรู</u>้ คำสั่งในหมวดนี้จะเป็นเงื่อนไขและค่าแสดงการรับรู้ ต่างๆของตัวละคร ประกอบด้วย





<u>หมวดโอเปอร์เรเตอร์</u> ใช้สำหรับการดำเนินการทางคณิตศาสตร์ ตรรกศาสตร์ และดำเนินการกับข้อความ ประกอบด้วย





10. <u>หมวด More Blocks</u> ใช้สร้าง Block คำสั่งเพิ่มเติม และเพิ่มส่วนขยาย



- ส่วน Make a Block คือการสร้าง Block คำสั่งใหม่ที่เราต้องการ โดย การสร้างชุด Block คำสั่ง ช่วยให้ไฟล์โปรแกรมมีขนาดเล็กลง และ เมื่อต้องการใช้ Block คำสั่งใหม่ จากตรงไหนของโปรแกรม เรา สามารถเรียกได้เลยโดยไม่ต้องลาก Block คำสั่งย่อยๆมาสร้าง หน้าตาการสร้าง Block คำสั่งใหม่ ประกอบด้วย
 - 1. ส่วนการสร้างหน้าตา Block ใหม่

	New Block	
💌 ตัวเลือก		
	เพิ่มการป้อนตัวเลข:	
	เพิ่มการป้อนข้อความ:	
	Add boolean input:	
	เพิ่มป้ายข้อความ:	ข้อความ
	Run without screen refres	h
	ຕກລາ ຍກເລີກ	



ตัวอย่าง การสร้าง Block สำหรับการหาค่าเลขยกกำลัง (power)



ส่วนของการประกอบคำสั่งเพื่อใช้งานตามที่เราต้องการ
 ตัวอย่าง ชุดคำสั่งสำหรับการยกกำลัง n1 ด้วยเลขชี้กำลัง n2
 แล้วเก็บผลลัพธ์ไว้ที่ตัวแปร x เริ่มชุดคำสั่งโดย กำหนดค่าเริ่มต้น
 ของตัวแปร x เป็น 1 แล้วทำซ้ำคูณค่า n1 จำนวน n2 รอบ ทุก
 ครั้งที่คูณ จะเก็บผลไว้ที่ตัวแปร x



วิธีการเรียกใช้ก็สามารถเรียกโดยใส่ค่าตัวเลข 2 ตัว เป็นเลขฐาน และเลขชี้กำลัง





ตัวอย่างเป็น 3 ยกกำลัง 3 ส่วนคำตอบจะอยู่ในตัวแปร x ดังนั้น หลังจากที่กำหนดคำสั่งให้ตัวละครพูดค่า x ออกมาเป็นเวลา 2 วินาที จะได้



เพิ่มส่วนขยาย เป็นการเลือกเครื่องมือติดต่อเสริมเข้าไปให้กับ
 Scratch โดยแต่ละเครื่องมือจะมีชุดคำสั่งที่แตกต่างกัน





ตัวอย่างชุดคำสั่งของ PicoBoard มีเซนเซอร์และปุ่มกดอยู่ที่อุปกรณ์ Block คำสั่งของอุปกรณ์นี้ก็จะล้อกับตัวอุปกรณ์ มีคำสั่งเมื่อเกิดเหตุการณ์ ปุ่มถูกกด ได้แก่ A B C และ D เชื่อมต่อ จะทำตามคำสั่งที่กำหนด หรือทำ การตรวจสอบเซนเซอร์แสงหรือเสียงว่ามากกว่าหรือน้อยกว่าค่าที่กำหนด ให้ดำเนินการต่ออย่างไร



ตัวอย่าง LEGO WeDo 2.0 จะมีมอเตอร์เข้ามาเกี่ยวข้องกับระบบ สามารถ สร้างโปรแกรมเพื่อควบคุมการทำงานของมอเตอร์ และตรวจสอบ สถานภาพของมอเตอร์ได้





2. Costumes

ตัวละครหนึ่งตัวจะมีลักษณะที่แตกต่างกันตามบริบท ตัวอย่าง แมวส้มตัวนี้ มี 2 Costumes สังเกตได้ว่าแมวสองตัวนี้เมื่อเราเรียกใช้ สลับกันจะเป็นเหมือนแมวกำลังเดินอยู่ Costumes ในตัวละครมีประโยชน์ ในการทำให้ตัวละครสามารถแสดงหน้าตาท่าทางหรือคำพูดที่ต่างกันตาม ฉากหรือบริบทที่ต่างกันตามที่เรากำหนด





เราสามารถเพิ่มเติมลักษณะอื่นเข้าไปโดยเพิ่มเติม ลักษณะจากตัวละครเดิม ให้คลิกขวาที่ Costume ตัวละครที่เราต้องการ แล้วเลือก "ทำซ้ำ"



เราสามารถวาดเส้นเพิ่มเติมตามที่เราต้องการด้วย เครื่องมือสำเร็จรูปของ Scratch ในการทำให้ตัวละครมี Costume ที่เรา ต้องการ ตัวอย่างการสร้าง Costume ของแมวส้มให้มีลักษณะมุ่งมั่น หรือ โกรธมากขึ้น อาจนำไปใช้เมื่อต้องต่อสู้





กรณีเป็นตัวละครตัวอื่นก็จะมีจำนวน Costume ที่โปรแกรมมีให้ ไม่เท่ากัน และมีลักษณะอารมณ์ท่าทางที่นำไปใช้ได้แตกต่างกัน ตัวอย่าง เจ้าฉลามน้อยตัวนี้ มีถึง 3 Costumes ด้วยท่าทางและอารมณ์ที่ต่างกัน



3. เสียง

ในเมนูเสียงของแต่ละตัวละครจะมีไม่เหมือนกันขึ้นอยู่กับ ลักษณะเฉพาะของตัวละครนั้น

ตัวอย่าง เสียงของกระดิ่ง Bells





มีไฟล์เสียงที่ชื่อว่า xylo1 ความยาวเกือบ 11 วินาที





เราสามารถเพิ่มเสียงที่ต้องการเข้ามาใหม่ได้ 3 รูปแบบ ได้แก่ เพิ่มเสียงจาก ไฟล์ที่มากับโปรแกรม บันทึกเสียงใหม่เอง และนำมาจากไฟล์เสียงที่อยู่ใน เครื่องเรา



5. Scripts Area



ส่วนสำหรับใส่สคริปต์ เราสามารถลาก Block จากส่วน Block Palette มาประกอบกันใส่ในส่วน Scripts Area นี้ เพื่อทำให้ตัวละครและฉากทำงาน ตามที่เราต้องการ

6. Tip box

เป็นส่วนช่วยให้ผู้ใช้ใหม่สามารถเริ่มเรียนรู้ความสามารถของโปรแกรม Scratch ได้อย่างง่ายเป็นขั้นตอน แบ่งตามหมวดหมู่ไว้เรียกใช้งานได้ง่าย และ มีตัวอย่างเกมส์ เรื่องราว แอนิเมชั่น และวิดีโอเพลง ไว้ให้ผู้ใช้สามารถทำตาม เพื่อเกิดความคิดสร้างสรรค์ต่อยอดได้ ซึ่งถึงแม้ว่าจะเป็นผู้ใช้งานไประยะหนึ่ง แล้วแต่หลงลืมบางการใช้งานโปรแกรมก็สามารถเลือกดูส่วนที่ต้องการได้ สะดวกจากตัวเลือกที่มีให้เหล่านี้



8	8	🗵 🔒 All Tips
Step-by-Step	Step-by-Step	Step-by-Step How To Blocks
Follow these tutorial project.	Learn more about ho your project.	Dig deeper here to find out more about each block, or use the ? button in the editor to click on individual blocks to learn more.
Getting	 Effects Play with effects 	Motion +
& NA_ Animate	Create interactive Interact with a mi	Looks +
🧨 Make It	ත් Animation	Sound +
Make M	Switch from happ Make a sprite tall	Pen +
	Make a sprite wa	Data +
Race to	686 Games	Events +
Hide-an	Use arrow keys to Bounce around	Control +
	Collide Keep score	Sensing +
Create	Soccer remix tips	Operators +
Let's Da	口 Stories	More Blocks +
	JJ Music	Extensions +

ภาพหน้าตา Tip

Box ทั้ง 3 ส่วน



การกำหนดคำสั่งให้กับเหตุการณ์

ในชีวิตประจำวัน เรามีการตั้งโปรแกรมตัวเองในการตอบโต้กับสิ่งรอบข้างที่ แตกต่างกันโดยไม่รู้ตัว เช่น การที่เราโดนแดด เรานำมือหรือสิ่งของขึ้นมาบังทิศทางที่ แดดส่อง เราเจอคนพูดเสียงดังใส่ เรารู้สึกตกใจ และพยายามเดินออกจากบริเวณนั้น หรือนักกีฬาวิ่งแข่ง 400 เมตร ที่อยู่ในท่าเตรียมพร้อม พอได้ยินเสียงสัญญาณเริ่มวิ่ง นักกีฬาจึงเริ่มออกตัววิ่ง เป็นต้น

ในการสร้างโปรแกรม เราจะต้องเจอกับเหตุการณ์หลายอย่างที่แตกต่างกันจาก ผู้ใช้ โดยเหตุการณ์เหล่านั้นผู้สร้างโปรแกรมจะเป็นคนกำหนดว่า เมื่อเกิดเหตุการณ์ แบบไหนให้ทำอะไร

ตัวอย่าง การสร้างเหตุการณ์เมื่อต้องการควบคุมตัวละครให้ไปตามทิศทาง ซ้าย ขวา บน ล่าง โดยการกดปุ่มลูกศรทิศทาง





โดยในโปรแกรม Scratch มีเหตุการณ์ให้เราเลือกใช้ได้หลายแบบ ได้แก่

- 1. เมื่อธงเขียวถูกคลิก
- 2. เมื่อปุ่มคีย์ที่เรากำหนดถูกกด
- เมื่อตัวละครที่เราใส่สคริปต์ไว้ถูกกด
- 4. เมื่อฉากหลังเราถูกเปลี่ยนเป็นฉากหลังที่กำหนด
- เมื่อความดัง ระยะเวลา หรือการเคลื่อนไหวของภาพในวิดีโอที่กำลังจับอยู่มี ค่ามากกว่าที่กำหนด
- 6. เมื่อตัวละครที่เราเลือกได้รับข้อความที่กำหนด
- 1. เมื่อธงเขียวถูกคลิก



โปรแกรมจะทำงานเมื่อธงเขียวบริเวณด้านบนขวาของฉากถูกคลิก

ตัวอย่างการใช้งาน เมื่อคลิกที่ธงเขียว ให้เครื่องบินบินผ่าน







2. เมื่อปุ่มคีย์ที่เรากำหนดถูกกด





โดยเราสามารถเลือกปุ่มคีย์ได้หลายปุ่ม ได้แก่ ลูกศรทั้งสี่ทิศทาง พื้นที่ ว่าง (spacebar) ตัวอักษรภาษาอังกฤษทั้ง 26 ตัว (A - Z) ตัวเลข 0-9 และ การคลิกที่ปุ่มใดก็ได้

ตัวอย่าง เมื่อกด spacebar (พื้นที่ว่าง) ให้เล่นเสียงเครื่องดนตรี และแสดง ข้อความว่า "โดนเคาะ"



โดยการใส่สคริปต์ตามนี้



อีกหนึ่งตัวอย่าง เป็นตัวอักษร A





เมื่อเรากดที่ปุ่ม 'a' บน keyboard ตัวอักษรจะตัวใหญ่ขึ้น



ด้วยสคริปต์นี้



เมื่อตัวละครที่เราใส่สคริปต์ไว้ถูกกด





การใช้เมาส์เพื่อคลิกไปที่ส่วนต่างๆของฉาก โดยเฉพาะเกมส์ที่ ต้องใช้เมาส์เป็นสิ่งหลักในการเล่น จะต้องใช้ Block เหตุการณ์นี้อย่าง แน่นอน



้ตัวอย่าง เมื่อเจ้าแมวส้มตัวนี้ถูกคลิก จะแสดงผลออกมาว่าถูกคลิก

ด้วยสคริปต์หน้าตาแบบนี้



อีกหนึ่งตัวอย่าง เป็นผู้ชายชื่อ Calvrett กำลังนั่งอยู่





เมื่อถูกคลิก ทำการเปลี่ยนท่าและแสดงคำพูด



ด้วยสคริปต์ดังนี้



4. เมื่อฉากหลังเราถูกเปลี่ยนเป็นฉากหลังที่กำหนด





บางครั้งในการเล่าเรื่องราว หรือ สร้างเกมส์ การเปลี่ยนฉากเพื่อให้ผู้ใช้รู ว่าเรากำลังอยู่ในฉากที่จะมีวิธีการใช้หรือเรื่องราวที่แตกต่างกัน เช่น เมื่อ ฉากเปลี่ยนเป็นทะเล ให้เริ่มเล่นเสียงคลื่น หรือเมื่อฉากอยู่ในสนามรบ ให้มี ทหารค่อยๆเดินเข้ามาในฉาก เป็นต้น

ตัวอย่าง มีเครื่องบิน กับผู้หญิงชื่อ Abby



เมื่อคลิกที่เครื่องบิน จะทำให้ฉากหลังเปลี่ยนเป็นทะเลทราย (gravel desert) แล้วเมื่อฉากหลังเปลี่ยน Abby จะเปลี่ยนชุดเป็น ชุด b





เรามาดูสคริปต์ของเครื่องบิน และสคริปต์ของ Abby กัน



สคริปต์ของเครื่องบิน



สคริปต์ของ Abby

 เมื่อความดัง ระยะเวลา หรือการเคลื่อนไหวของภาพในวิดีโอที่กำลังจับอยู่ มีค่ามากกว่าที่กำหนด




เราสามารถใช้การตรวจสอบสามสิ่งนี้เพื่อ เพื่อโปรแกรมจะเปลี่ยน สถานะหรือทำงานบางอย่างให้สอดคล้องกับสิ่งที่เปลี่ยนแปลง

ตัวอย่าง เราจะจับเวลา 10 วินาที เมื่อถึง 10 วินาทีให้แมวส้มกลับหัว



โดยมีตัวช่วยนับเวลาให้เราเห็นโดยติ๊กที่ส่วนจับเวลาในหมวดการรับรู้



และสคริปต์ที่ใส่ไปในตัวแมว





จะได้แมวตีลังกา หลังจากผ่านไป 10 วินาที



6. เมื่อตัวละครที่เราเลือกได้รับข้อความที่กำหนด

การส่งสารระหว่างตัวละครหรือสิ่งของบางอย่างจะทำให้เมื่อตัว ละครตัวหนึ่งทำอะไรบางอย่างแล้วต้องการให้ตัวละครอีกตัวหรือสิ่งของ บางอย่างทำงานต่อเนื่องจากการกระทำของตัวละครตัวนั้น เราสามารถใช้ การกระจายข้อความช่วยให้ตัวละครตัวอื่นรับรู้ถึงคำสั่งจากตัวละครนี้

ตัวอย่าง มีรถทำความสะอาดถนนคันใหญ่ กับลูกศรชี้ไปทางขวา





เมื่อคลิกที่ลูกศร รถจะเคลื่อนไปทางขวา



โดยเราใส่สคริปต์ที่มีการคุยกันได้ของสองตัวละครนี้





การกำหนดตัวแปรและการคำนวณ

เราสามารถเก็บข้อมูลตัวเลขและตัวอักษรบรรจุใส่สิ่งที่เรียกว่าตัวแปร และ สามารถใช้การคำนวณทางคณิตศาสตร์เพื่อใช้หาผลลัพธ์ในบางส่วนของโปรแกรม

<u>การสร้างตัวแปร</u>

คลิกที่สร้างตัวแปร และใส่ชื่อตัวอย่าง ชื่อ score และเลือกสำหรับตัวละคร ทั้งหมด

สร้างตัวแปร	
ตัวแปรใหม่	
ชื่อตัวแปร: score	
● สำหรับตัวละครทั้งหมด 🔿 สำหรับตัวละครนึเท่านั้น	
ตกลง ยกเลิก	

กดเช็คที่กล่องสี่เหลี่ยมหน้า Number เพื่อให้ ตัวเลขที่สุ่มมาอยู่บนหน้าจอ สำหรับดูผลที่เกิดขึ้น



เปลี่ยน Number 🔻 ที่ละ 🚺



ใช้เพื่อเปลี่ยนค่าของตัวแปรเพิ่มตามจำนวนที่กำหนดด้านหลัง เช่น เรา สามารถใช้ในการเพิ่มแต้มในเกมส์ได้ เป็นต้น



Block นี้ควรจะเป็นคำว่า "แสดงตัวแปร" แต่เข้าใจว่าคนแปรจากภาษาอังกฤษ มาผิด ใช้สำหรับแสดงตัวแปรนี้ทางหน้าจอ



ส่วน Block นี้คือการ "ซ่อนตัวแปร" จริงๆ ใช้ซ่อนตัวแปรออกจากหน้าจอ

ตัวอย่างการใช้ตัวแปรเป็นส่วนหนึ่งในการสร้างเกม เกมที่เราจะสร้างเป็นเกม บอกว่าตัวเลขที่สุ่มมาได้มีค่าน้อยหรือมาก โดย แสดงคำว่า High หาก เลขอยู่ในช่วง 4 ถึง 6 และแสดงค่า Low หากเลขที่ได้อยู่ในช่วง 1 ถึง 3 เราจะเริ่มโปรแกรมสุ่มโดย คลิกที่ปุ่มสีฟ้า





การนำตัวแปรไปใส่ไว้ในสคริปต์จุดต่างๆของโปรแกรม เริ่มตั้งแต่ส่วนปุ่มสีฟ้า จะเป็นการกำหนดค่าจากการสุ่มใส่ไว้ในตัวแปร Number







ส่วนป้ายบอก High และ Low จะน้ำค่าจากตัวแปร Number มาตรวจสอบ

<u>การคำนวณ</u>

ในการสร้างโปรแกรมหรือเกมส์ เราจะมีการคำนวณในบางส่วน เราใช้ คอมพิวเตอร์ในการคำนวณให้เราได้เลย การคำนวณทางคณิตศาสตร์และ ตรรกศาสตร์ในโปรแกรม Scratch มีให้เลือกหลายตัว ได้แก่

1. การคำนวณพื้นฐาน

การคำนวณพื้นฐานอย่างบวก ลบ คูณ หรือ หาร



โดยในช่องวงกลมสีขาวเราสามารถพิมพ์ตัวเลขใส่เข้าไป หรือ Block ที่เป็น ตัวเลข ตัวแปร หรือการคำนวณใส่เข้าไปได้เช่นกัน



ตัวอย่าง การนำ Block ตัวแปร การคำนวณ และการพิมพ์ตัวเลข ใส่เข้าไปใน Block การคำนวณการบวก



2. การสุ่มตัวเลข

เราจะใช้การสุ่มตัวเลขส่วนมากในการสร้างเกมส์



จาก Block ตัวอย่างด้านบน เราสามารถเลือกสุ่มตัวเลขจาก 1 ถึง 10 จริงๆเราสามารถแก้ไขตัวเลขตรงนี้ได้ เช่น การสุ่มเลข 1 ถึง 100 เป็นต้น

3. การเปรียบเทียบทางคณิตศาสตร์

เราใช้การเปรียบเทียบทางคณิตศาสตร์กับส่วนที่เป็นเงื่อนไข เพื่อ ตัดสินใจว่าโปรแกรมจะทำงานส่วนใดต่อ



ตัวอย่างการนำไปใช้กับส่วนการตัดสินใจของโปรแกรม หรือส่วนที่มีช่อง ให้ใส่กล่อง 6 เหลี่ยม





4. การดำเนินการทางตรรกศาสตร์



ส่วนของการตัดสินใจบางครั้งเราจะใช้การเปรียบเทียบหลายชุดร่วมกัน ในการตัดสินใจทำหรือไม่ทำชุดคำสั่ง การดำเนินการทางตรรกศาสตร์จะช่วย ให้การตัดสินใจนั้นครบถ้วนมากขึ้น ตัวอย่างการตัดสินใจในชีวิตประจำวัน เช่น

ถ้าฝนตก และ แดดออก จะเกิดรุ้งกินน้ำ กรณีนี้เงื่อนไขของเหตุการณ์คือ "ฝนตก และ แดดออก" ซึ่งขั้นด้วย "และ" ถ้าฝนไม่ตก หรือ แดดไม่ออก อย่าง ใดอย่างหนึ่ง ก็จะไม่เกิดรุ้งกินน้ำ

ถ้าฝนตกจนน้ำขัง หรือ มีอุบัติเหตุรถชน รถจะติด กรณีนี้เงื่อนไขของ เหตุการณ์ คือ "ฝนตกจนน้ำขัง หรือ มีอุบัติเหตุรถชน" แต่ขั้นด้วย "หรือ" ดังนั้น ไม่ว่าฝนตกจนน้ำขัง หรือมีอุบัติเหตุรถชน อย่างใดอย่างหนึ่ง ก็จะทำให้ รถติด



ส่วน "ไม่" เป็นการทำให้ค่าความจริงเป็นเท็จ และค่าความเท็จเป็นจริง ตรงข้ามกัน



ตัวอย่างโปรแกรม การใช้ตรรกศาสตร์เข้ามาเกี่ยวข้อง

ตามสคริปต์นี้ มีเงื่อนไขเกี่ยวกับตำแหน่งของตัวละคร ต้องเป็นจริงทั้ง ตำแหน่ง x และ ตำแหน่ง y ถ้าอยู่ในเงื่อนไขแมวจะเคลื่อนที่ไปด้านหลัง 100 ก้าว





จะได้ผลลัพธ์ดังนี้



5. การดำเนินการกับข้อความ

การสร้างเรื่องราว เกมส์ หรือแอนิเมชั่น ที่มีการพูดคุย หรือสื่อสารให้ ผู้ใช้งานเข้าใจ จำเป็นต้องอาศัยข้อความที่หลากหลายและยืดหยุ่น Scratch จึงมี Block คำสั่งพื้นฐานให้กับผู้สร้างโปรแกรมเลือกใช้งานได้อย่างง่ายตาม ความต้องการ ได้แก่



- 1. การเชื่อมข้อความ
- 2. การเลือกดูตัวอักษรของคำโดยการใช้ตำแหน่งในการเลือก
- 3. การบอกค่าความยาวของข้อความ



1. การเชื่อมข้อความ

เราสามารถเชื่อมข้อความสองข้อความโดยอาศัย Block คำสั่งนี้



ตัวอย่างการใช้ควบคู่กับการรับชื่อจากผู้ใช้มาแสดงคำทักทาย







ลคริปต์ที่ใช้





 การเลือกดูตัวอักษรของข้อความโดยการใช้ตำแหน่งในการเลือก เราสามารถใช้งานโดยใส่สคริปต์ และตัวเลขแสดงตำแหน่ง



จากตัวอย่าง ตัวอักษรตัวที่ 2 ของคำว่า "English" ก็จะได้ตัว 'n'

การบอกค่าความยาวของข้อความ
ตัวอย่างเมื่อใช้สคริปต์นี้เพื่อบอกค่าความยาวของคำว่า "Thailand"
ก็จะได้ผลออกมาเป็น 8 ซึ่งก็คือจำนวนตัวอักษรของข้อความ



หลายครั้งเราอาจจะใช้การไล่ดูตัวอักษรในคำคู่กับความยาว

ตัวอย่าง รับชื่อเข้าไปแล้วให้แสดงตัวอักษรทีละตัวที่อยู่ในชื่อ





เมื่อใส่ชื่อ CDG เข้าไป แมวจะพูดออกมาเป็นตัวอักษรทีละตัวโดย count ก็จะเพิ่มขึ้นเรื่อยๆตั้งแต่ 1 ตามจำนวนตัวอักษรของคำ





ส่วนหน้าตาของสคริปต์จะยังมีส่วนที่ไม่คุ้นเคยอยู่บ้าง เราจะได้เรียนรู้

กันต่อในหัวข้อต่อไป



มาดูตัวอย่างการประยุกต์ใช้การดำเนินการทางคณิตศาสตร์มาคำนวณการหา ด้านตรงข้ามมุมฉากของพีทาโกรัส และนำการดำเนินการข้อความมาประกอบการใช้ งาน

เริ่มจากการวาดสามเหลี่ยมมุมฉาก นำตัวแปรไปวางรอเก็บค่าจากผู้ใช้ และนำ เจ้าแมวส้มไปคอยบอกค่าด้านตรงข้ามมุมฉากจากการคำนวณ





ใส่สคริปต์ให้กับตัวแมวเมื่อคลิกธงเขียวให้เริ่มทำงานโดยรับค่าความ ยาวด้าน a และ ความยาวด้าน b เพื่อนำมาคำนวณ







เสร็จแล้วเจ้าแมวก็จะบอกค่าด้านตรงข้ามมุมฉากจากการคำนวณมาให้



นี่คือสคริปต์ที่ใช้ในโปรแกรม เน้นประยุกต์ใช้ความรู้ที่ผ่านมา





การสร้างเงื่อนไขและการวนซ้ำ

<u>การสร้างเงื่อนไข</u>

ในชีวิตประจำวันเราสร้างเงื่อนไขให้กับการกระทำบางอย่างแบบไม่รู้ตัว เช่น เราตั้งเงื่อนไขของวันหยุด กับวันธรรมดา เป็นเหตุการณ์ที่ส่งผลต่อการกระทำที่ ต่างกัน โดยรูปแบบเงื่อนไขของการสร้างโปรแกรมมี 2 รูปแบบ ได้แก่ "ถ้า.. แล้ว.. มิ ฉะนั้น.." และ "ถ้า.. แล้ว.." ซึ่งแบบแรกจะเป็นการบอกว่าถ้าเงื่อนไขดังกล่าวเป็นจริง จะมีการกระทำบางอย่าง แต่ถ้าไม่ตรงตามเงื่อนไขนั้น จะกระทำอีกแบบหนึ่ง แต่แบบ ที่สองจะกระทำบางอย่างเมื่อเงื่อนไขเป็นจริงเท่านั้น ถ้าไม่เป็นจริงจะไม่มีการกระทำ อะไรที่เกิดขึ้นต่อ

ตัวอย่าง รูปแบบเงื่อนไข "ถ้า.. แล้ว.. มิฉะนั้น.."

- <u>ถ้า</u>วันนี้เป็นวันหยุด <u>แล้ว</u>ฉันจะไปเที่ยว <u>มิฉะนั้น</u>ฉันจะไปโรงเรียน

- <u>ถ้า</u>วันนี้เป็นวันพระ <u>แล้ว</u>ฉันจะออกไปใส่บาตร <u>มิฉะนั้น</u>ฉันจะนอนต่อ

ตัวอย่าง รูปแบบเงื่อนไข "ถ้า.. แล้ว.."

- <u>ถ้า</u>ช่วงนี้เป็นฤดูฝน <u>แล้ว</u>ฉันจะพกร่มออกจากบ้าน

- <u>ถ้า</u>ฝนไม่ตกเย็นวันอังคาร <u>แล้ว</u>ฉันจะซักผ้า

สำหรับโปรแกรมของเราจะมีคำสั่งที่เป็นรูปแบบคล้ายกับเรื่องราวที่ยกให้ ด้านบน ดังนี้

1. ถ้า.. แล้ว..





ตัวอย่างการใช้งาน



เมื่อรันสคริปต์นี้ โปรแกรมจะให้เราใส่ค่าคะแนนลงไป



จากนั้นเจ้าแมวจะตอบกลับมาว่าคุณได้ A ถ้าคะแนนคุณมากกว่า 79





แต่ถ้าคะแนนที่คุณใส่ไปไม่มากกว่า 79 หละ สิ่งที่แมวจะทำคือ



อยู่นิ่งๆ ไม่พูดอะไร เพราะเงื่อนไขไม่เป็นจริง



2. ถ้า.. แล้ว.. มิฉะนั้น..





เรามาประยุกต์เพิ่มเติมจากตัวอย่างก่อนหน้านี้ เมื่อแมวรับค่าคะแนนไป แล้วแต่คะแนนกลับไม่มากกว่า 79 เราอยากให้แมวทำอย่างอื่น เช่น บอกว่า เราไม่ได้ A นะ เป็นต้น เราจะใช้เงื่อนไขแบบ "ถ้า.. แล้ว.. มิฉะนั้น" ลองดูตัวอย่างสคริปต์นี้ เราได้เปลี่ยนจากเงื่อนไขแบบ "ถ้า.. แล้ว

ลองดูตัวอย่างสคริปตันิ เราได้เปลี่ยนจากเงื่อนไขแบบ "ถ้า.. แล้ว .." เป็น "ถ้า.. แล้ว.. มิฉะนั้น.."

dia /	🗖 ถูกคลิก
arm W	hat's your score?
set s	core T to Annu
in 📢	score > 79 uñs
186	รุณได้ A! เป็นเวลา (5 วินาที
มิละนั้น	
199	คุณไม่ได้ A เรียใจด้วย เป็นแวลา (5) วินาที

จะทำให้เมื่อเราใส่คะแนนที่ไม่มากกว่า 79 เข้าไป จะได้ผลลัพธ์ บอกกลับมาว่าเราไม่ได้ A แทนที่จะเงียบๆไป





มากไปกว่านั้น เราสามารถเพิ่มกรณีที่มากขึ้นกว่าเดิมได้โดยใช้ เงื่อนไข "ถ้า.. แล้ว.. มิฉะนั้น.." ซ้อนกันเข้าไป

ตัวอย่างการสร้างโปรแกรมตัดเกรด ให้สามารถบอกได้ทุกเกรด ของการเรียน

ún /	อุกลลิก	
erna Wi	nat's your score? En=naa	
set sc	ore to Annau	
ān 🃢	score > 79 Lin	
1810	ลุณได้ A! เป็นเวลา 5 วินาที	
มิละนั้น		
ด้า	score > 69 min	
	ล เลลไล้ B! เป็นเวลา (5) วันวที	
20:0=1	Iu	
ត	score > 59 min	
	ซูด คุณได้ C! เป็นแวลา (5 วินาที	
Ĵ.	1=111	
	ñn (score) > 49) săn	
	ทุล คุณได้ D! เป็นเวลา 5 วินาที	
	มิละนั้น	
	ซูต <mark>สุณได้ F!</mark> เป็นเวลา (5 วันา	1
	and the second se	



ทำให้เมื่อเราใส่คะแนนเท่าไหร่ก็ตามเราก็จะสามารถรู้ได้ว่าเราได้ เกรดเท่าไหร่





<u>ุการวนซ้ำ</u>

การวนซ้ำเป็นเรื่องที่คอมพิวเตอร์สามารถทำได้ดีกว่ามนุษย์ ช่วยทุ่นแรง ได้เยอะ เพราะคอมพิวเตอร์ถนัดในการทำงานซ้ำๆที่เราตั้งคำสั่งเอาไว้โดยไม่ เหนื่อย รวดเร็วเป็นหลักมิลลิวินาที และที่สำคัญคือมีความแม่นยำมากเพราะ คอมพิวเตอร์มีปัจจัยที่มารบกวนน้อย เราเลยเลือกใช้คอมพิวเตอร์ในการ ทำงานแทนเราในบ้างครั้งที่เราต้องทำอะไรซ้ำๆเป็นจำนวนมากๆๆๆ เช่น การ คำนวณเลขจำนวนมากซ้ำๆ การหาค่ารากที่สองของจำนวนจริงที่ไม่ลงตัว การ คูณเลขจำนวนหลายหลักหลายๆครั้ง หรือการแสดงผลซ้ำๆกันหลายพันครั้ง เป็นต้น

Block สำหรับการวนซ้ำในโปรแกรม Scratch มี 3 รูปแบบ ได้แก่

1. การวนซ้ำตามจำนวนรอบที่กำหนด



2. การวนซ้ำไปเรื่อยๆไม่รู้จบ



3. การวนซ้ำจนกระทั่งเป็นไปตามเงื่อนไขที่กำหนด





การวนซ้ำตามจำนวนรอบที่กำหนด
ใช้กับโปรแกรมที่สามารถกำหนดจำนวนรอบได้เป็นตัวเลข

<u>ตัวอย่าง</u> โปรแกรมการแสดงดาวตามจำนวนที่กำหนด

้เริ่มจากการนำดาวมาใส่ไว้ด้านบนตรงกลาง



ตามด้วยตัวแปรเก็บจำนวนดาวที่จะกระจายออกมา และตัวแปร เก็บตำแหน่ง x ที่เปลี่ยนไป



เราจะนำสองสคริปต์นี้ใส่ไว้ที่ดาว เพื่อทำให้ดาวกระจายออกมา ตามจำนวนที่ใส่เข้าไป



เมือ / อุกคอิก
อาณ ภระจายลาวกี่ดวงดินะ และกอย
set Inauan to Ananu
set douvria x = to -240
กาล้า จำนวนกาว
สร้างสีวโทลนของ myself
1011 (* 15 Baffi
DIBERL drumie x 7 718= 40
when I start as a clone
ร่อน 2 วินาที ไปที่ x: สำแหน่ง x γ: -180

เมื่อเริ่มโปรแกรมด้วยการคลิกที่ธงเขียว โปรแกรมจะให้ใส่จำนวน ดาวเข้าไป



เราจะได้ดาวกระจายตัวออกมาจำนวน 4 ดวง





กรณีที่เราใส่จำนวนดาวไป 10 ดวง ดาวก็จะกระจายออกมา 10





เมื่อเปลี่ยนจำนวนดาวเป็น 50 ดาว และเปลี่ยนตำแหน่ง x ให้ เพิ่มขึ้นทีละ 10 จะได้ดาวกระจายเรียงกันออกมามากมายต่อเนื่องกัน ซึ่งถ้าเราค่อยๆกำหนดดาวทีละดวงให้กระจายออกมาเป็นชุดลักษณะนี้ ทีละดวงคงเสียเวลามาก ลองจินตนาการว่าเราจะทำให้ดาวกระจาย ออกมา 1 ล้านดวงดูสิ.. เราคงต้องใช้เวลาเป็นปีในการสร้างมันถูกไหม แต่การเขียนโปรแกรมคอมพิวเตอร์ช่วยเราได้มากในการทำอะไรซ้ำๆๆๆ กันในลักษณะนี้



2. การวนซ้ำไปเรื่อยๆไม่รู้จบ

เราลองมาทำดาวหมุนๆไปเรื่อยๆไม่รู้จบกัน โดยดาวจะเกิดเพิ่มขึ้น เรื่อยๆไม่รู้จบเช่นกัน

เริ่มจากน้ำดาวหนึ่งดวงมาไว้ตรงกลางหน้าจอ





เราจะใส่สคริปต์ 3 ก้อนนี้เข้าไปในดาว สังเกตได้ว่าทั้ง 3 ชุดคำสั่งจะมี ส่วนของการวนซ้ำตลอด อยู่ในชุดสคริปต์ เพื่อให้บางส่วนของสคริปต์ได้ ทำงานตลอดเวลา





ส่วนแรกเป็นส่วนทำให้ดาวตรงกลางเริ่มหมุนโดยเปลี่ยนสีไป







ส่วนที่สองเป็นส่วนที่ใช้สร้างดาวออกมาเรื่อยๆทุกๆ 0.5 วินาที โดยทุกครั้งที่มีการสร้างดาว ให้เล่นเสียง pop ออกมาด้วย แสดงว่าทุกครั้ง ที่เราได้ยินเสียง pop แสดงว่ามีดาวเกิดขึ้น



ส่วนที่สาม เป็นส่วนที่บอกให้ดาวที่ถูกสร้างขึ้นมาใหม่เกิดใน ตำแหน่งที่ถูกสุ่มขึ้นมาและทำการหมุนและเปลี่ยนสีไปเรื่อยๆเหมือนดาว ดวงแรก



เมื่อเวลาผ่านไปช่วงหนึ่ง เราจะเห็นดวงดาวระยิบระยับกระจาย อยู่บนหน้าจอของเรามากมาย





3. การวนซ้ำจนกระทั่งเป็นไปตามเงื่อนไขที่กำหนด

ก่อนหน้านี้เราสามารถวนซ้ำได้จนไม่มีที่สิ้นสุด ซึ่งบางครั้งเราอาจจะ อยากให้การวนซ้ำนั้นวนไปได้เรื่อยๆแต่ก็อยากให้หยุดด้วยในบางเงื่อนไข เราจะสามารถทำได้โดยใช้ Block "ทำซ้ำจน.."



จากการสร้างดาวหมุนในตัวอย่างก่อนหน้าโดยการใช้ Block "วนซ้ำ ตลอด" คราวนี้ลองมาใส่เงื่อนไขการหยุดให้กับการสร้างดาว การหมุน และ การเปลี่ยนสีของดาวกันดูดีกว่า



เราจะเปลี่ยน Block จาก "วนซ้ำตลอด" เป็น Block "ทำซ้ำจน.." ก่อน แล้วใส่เงื่อนไขเกี่ยวกับเวลาเข้าไป ว่าจะให้หยุดเมื่อเวลาผ่านไปเท่าไร

โดยเงื่อนไขที่เลือกใส่ไปในตัวอย่างนี้จะเป็นเรื่องของเวลา คือ เมื่อเวลา ผ่านไปจำนวนหนึ่งตามการสุ่มจะทำให้การหมุนกับการเปลี่ยนสีหยุดลง ซึ่ง ด้วยการสุ่มเวลา ทำให้เอฟเฟคของดาวหยุดไม่พร้อมกัน ส่วนการสร้างดาว ใหม่จะหยุดสร้างก่อนที่ดาวจะหยุดหมุนและเปลี่ยนสี

มาดูสคริปต์กัน..







การออกแบบและสร้างโปรแกรม

เราได้เรียนรู้บทเรียนการเขียนโปรแกรมกันมาเต็มที่แล้ว ถึงเวลาสร้างสรรค์โปร เจคของตัวเองกันเต็มที่! แต่.. ก่อนที่เราจะเริ่มทำ มาลองศึกษาขั้นตอนกระบวนการ สร้างโปรเจค ได้แก่

- 1. สำรวจ (Explore)
- 2. วางแผน (Plan)
- 3. พัฒนา (Develop)
- 4. แบ่งปัน (Share)


1. สำรวจ (Explore)

ก่อนเริ่มขั้นตอนของการพัฒนาโปรแกรม จากแผนภาพข้างต้น จะเห็น ได้ว่าขั้นตอนของการสำรวจ จะมาแยกจากขั้นตอนอื่นๆที่วงเป็นวนรอบ เพราะเราควรทราบก่อนว่าเราอยากที่จะทำอะไรจริงๆ เรามีแรงบันดาลใจ ในการแก้ไขปัญหาอะไร สิ่งเหล่านี้จะทำให้เราพัฒนาโปรแกรมได้อย่างเต็ม ความสามารถ โดยเรามีขั้นตอนการสำรวจความสนใจด้วยคำถามเหล่านี้

- a. โปรเจคที่เรียนรู้ผ่านมาโปรเจคไหนที่ชอบที่สุด
- b. มีความสนใจในเรื่องใด (เรื่องอะไรก็ได้ในชีวิต)
- c. สิ่งใดบ้างที่เราอยากจะใช้โปรแกรมที่เราจะพัฒนาขึ้นในการ แก้ปัญหา
- การพัฒนาโปรแกรมแบบใดที่ตอบโจทย์การแก้ปัญหาของเรา เป็น เกมส์ เรื่องราว แอนิเมชั่น หรืออื่นๆ
- e. อะไรเป็นความถนัด หรือความสามารถพิเศษของเรา

คำถามด้านบนใช้สำหรับสำรวจตัวตนของเรา เมื่อเราสำรวจความ ต้องการและความสามารถของเราได้เรียบร้อย คำแนะนำถัดไปคือการหา ทีม แน่นอนว่าเราอาจจะถนัดในบางอย่าง และไม่ถนัดในบางอย่างแต่สิ่งที่ เราสนใจ อาจจะไม่ได้มีแต่เราเพียงคนเดียว เราสามารถหาเพื่อนร่วมทีมที่มี ความถนัดต่างจากเรา เพื่อมาช่วยกันแก้ไขปัญหาที่สนใจร่วมกันได้

หลังจากเราตอบคำถามด้านบน และหาเพื่อนร่วมทีมได้แล้ว ถึงเวลา คิดถึงโปรแกรมที่เราจะร่วมกันพัฒนาขึ้นมา

2. วางแผน (Plan)

ขั้นตอนการวางแผน เราจะได้คำตอบสำหรับ 4 คำถามนี้

- a. เราจะสร้างโปรแกรมอะไร
- b. มีขั้นตอนอย่างไรบ้างในการสร้างโปรแกรมนี้



- c. สิ่งที่เรามีอยู่แล้ว สำหรับโปรแกรมที่เราจะสร้าง เช่น ทีม ตัวอย่าง
 โปรแกรม ไฟล์ภาพ เป็นต้น
- d. สิ่งที่เราต้องการเพิ่มเติม เพื่อใช้ในการสร้างโปรแกรม

<u>คำถามแรก</u> เราจะสร้างโปรแกรมอะไร

วิธีที่เราแนะนำในการการหาคำตอบของคำถามนี้ เรียกว่า "การ ระดมความคิด (Brainstorm)" การระดมความคิดมีขั้นตอน ดังนี้

- ให้ทุกคนในทีมเสนอไอเดียที่อยากทำให้มากที่สุดเท่าที่เป็นไป ได้ในเวลาที่จำกัด เช่น 3 นาที เป็นต้น โดยอาจเขียนใส่ กระดาษโพสอิท แล้วพูดออกมาว่าไอเดียในการแก้คืออะไร แล้วแปะลงไปบนกระดาษหรือกระดานที่เตรียมไว้
- 2. ทีมช่วยกันจัดกลุ่มไอเดียที่มีความใกล้เคียงกันเป็นหมวดหมู่
- พูดคุยกันถึงความน่าสนใจ และความเป็นไปได้ของแต่ละไอ เดีย
- แต่ละคนเลือกไอเดียที่ชอบ 3 อันดับในใจ และเขียนอันดับ ของตนเองลงใส่โพสอิทไอเดียที่เลือกเป็นคะแนน 1-3 โดย 3 คะแนน คือไอเดียที่ชอบที่สุด
- รวมคะแนนแต่ละไอเดียและจัดอันดับไอเดียที่มีคะแนนมาก ที่สุดไปหาน้อยที่สุด
- ช่วยกันเลือกไอเดียที่ทุกคนชอบมากที่สุดอาจจะ 3-5 อันดับ แรก โดยพูดคุยถึงเหตุผลในการเลือกของแต่ละคน ซึ่งผล สุดท้ายอาจจะได้ข้อสรุป คือ การเลือกอันดับที่ 3 ก็เป็นได้





<u>คำถามที่เหลือ</u>

- มีขั้นตอนอย่างไรบ้างในการสร้างโปรแกรมนี้ สิ่งที่เรามีอยู่แล้ว
- สำหรับโปรแกรมที่เราจะสร้าง เช่น ทีม ตัวอย่างโปรแกรม ไฟล์
 ภาพ เป็นต้น
- สิ่งที่เราต้องการเพิ่มเติม เพื่อใช้ในการสร้างโปรแกรม

หลังจากได้ไอเดียออกมาแล้วว่าจะทำอะไร ขั้นตอนนี้ เราจะมา ช่วยกันในทีมลองเขียนขั้นตอนการสร้างโปรแกรมดูว่า เราจะทำ อะไรบ้างเป็นลำดับ เพื่อดูว่าแต่ละขั้นตอนเราต้องการอะไรบ้าง สิ่งใดมี อยู่แล้ว สิ่งใดต้องทำเพิ่ม เรามีแบบฟอร์มในการเขียนให้ตามเอกสาร หน้าถัดไปนี้





ชื่อโปรแกรม

ลองเขียนภาพคร่าวๆของการดำเนินของโปรแกรมเราด้านล่างนี้ และเขียนด้านล่างของหน้าจอว่าในแต่ละขั้นตอนอาศัยสิ่งใดบ้างที่ ใช้ในหน้านั้น



อธิบายสิ่งที่จะเกิดขึ้นในหน้านี้

อธิบายสิ่งที่จะเกิดขึ้นในหน้านี้

สิ่งที่ต้องใช้ในหน้านี้ เช่น ตัวละคร เสียง ตัวอย่างโปรเจค เป็นต้น

สิ่งที่ต้องใช้ในหน้านี้ เช่น ตัวละคร เสียง ตัวอย่างโปรเจค เป็นต้น



อธิบายสิ่งที่จะเกิดขึ้นในหน้านี้

อธิบายสิ่งที่จะเกิดขึ้นในหน้านี้

สิ่งที่ต้องใช้ในหน้านี้ เช่น ตัวละคร เสียง ตัวอย่างโปรเจค เป็นต้น

สิ่งที่ต้องใช้ในหน้านี้ เช่น ตัวละคร เสียง ตัวอย่างโปรเจค เป็นต้น



3. พัฒนา (Develop)

ขั้นตอนนี้คือขั้นตอนของการพัฒนาโปรแกรม ทุกคนในทีมจะมีหน้าที่ที่ ต่างกันในการพัฒนา สิ่งที่แนะนำในขั้นตอนนี้คือ ในทีมควรกำหนดเวลาการ มาอัพเดทงานกันขึ้นอยู่กับเวลาว่ามีมากหรือน้อย กรณีเวลามีทั้งหมด 6 ชั่วโมงในการทำโปรแกรม ทีมอาจจะมาอัพเดทข้อมูลการทำงานของสมาชิก ในทีม กันทุกๆ 1 ชั่วโมง เพื่อนำปัญหาที่แต่ละคนเจอมาหาทางแก้ไขร่วมกัน หรือจัดแบ่งหน้าที่กันใหม่ให้สำเร็จตามเวลา

คำถามแนะนำสำหรับการอัพเดทการทำงาน

- 1. อะไรเป็นส่วนที่ชอบที่สุดในการทำงานที่ผ่านมา
- 2. ส่วนใดที่ยังต้องทำต่อไป
- 3. ส่วนไหนที่คนอื่นในทีมจะต้องทำต่อจากเรา
- 4. ส่วนใดที่ต้องการความช่วยเหลือจากคนอื่นเพื่อให้งานเราไปต่อได้
- 4. แบ่งปัน (Share)

หลังจากที่เราวางแผน ออกแบบ พัฒนาได้ถึงระดับหนึ่งแล้ว เป็นโอกาส ดีที่เราจะแบ่งปันเรื่องราวการทำงานของโปรแกรมเราให้กับคนอื่นๆรู้ เพื่อรับ ความแนะนำ ติชม (Feedback) และรับฟังไอเดียการทำงานของโปรแกรม ของคนอื่นเพื่อช่วยออกความเห็นเพื่อการพัฒนาของเขาและได้รับมุมมองที่ แตกต่างสำหรับการพัฒนาโปรแกรมของเราเช่นกัน

คำถามสำหรับการให้ความเห็นกับโปรแกรมของคนอื่น

- 1. สิ่งที่ทำงานได้ดีหรือสิ่งที่เราชอบมากในโปรแกรม
- 2. สิ่งที่ยังสับสนอยู่ว่าทำงานอย่างไร แนะนำว่าควรปรับเป็นแบบใด
- 3. สิ่งที่ทำให้ดีขึ้นได้หรือสิ่งที่ไม่ควรอยู่ในโปรแกรม



หลังจากได้รับคำแนะนำติชมมาเรียบร้อย เราจะกลับไปที่ขั้นตอนของ การวางแผน (Plan) พัฒนาโปรแกรม (Develop) และกลับมาที่แบ่งปัน (Share) ใหม่ วนไปจนกระทั่งถึงเวลาที่จะแสดงให้คนอื่นเห็น (Showcase)

มาถึงจุดนี้ เราอาจจะรู้สึกกังวลหรือมีความเครียดกับการทำโปรแกรมไม่ เสร็จดีนัก แต่จริงๆแล้วเราได้รับประสบการณ์มากมายที่เป็นจุดเริ่มต้นในการ เป็นผู้พัฒนาโปรแกรม และการที่เราทำโปรแกรมมาได้ถึงขนาดนี้ก็ถือว่าสุด ยอดมากๆแล้ว

สุดท้ายสิ่งที่สร้างจะมีคุณค่ามากขึ้นเมื่อเราได้หันกลับมาถามตนเองว่า สิ่งที่เราได้สร้างสรรค์ ได้คิดอย่างเป็นระบบ ฯลฯ ส่งผลอย่างไรต่อตัวเราบ้าง ซึ่งเรามีชุดคำถามให้ตามนี้

- 1. โปรแกรมของเราคืออะไร
- 2. ไอเดียโปรแกรมของเราได้ช่วยแก้ปัญหาอย่างไร เราได้ไอเดียนี้มา อย่างไร
- 3. ขั้นตอนการพัฒนาโปรแกรมของกลุ่มเราเป็นอย่างไร
- สิ่งใดในการพัฒนาที่น่าสนใจ ท้าทาย และน่าประหลาดใจ และ ทำไมถึงเกิดความรู้สึกแบบนั้นขึ้นมา
- 5. เราได้เรียนู้รูอะไรบ้าง
- 6. สิ่งที่ชอบมากเกี่ยวกับโปรเจคนี้
- 7. อะไรที่เราอยากจะเปลี่ยน ถ้าเรากลับไปแก้ไขได้
- ความรู้สึกต่อการเขียนโปรแกรมหรือการสร้างโปรแกรมเปลี่ยนไป จากตอนแรกอย่างไร

