

คู่มือการเรียนรู้การสอน หลักสูตรการเขียนโปรแกรมเบื้องต้น



คำนำ

ทักษะการเขียนภาษาโค้ดเป็นทักษะที่มีความต้องการสูงมากในตลาดแรงงานในยุคเศรษฐกิจดิจิทัล และยังเปิดประตูสู่โอกาสแห่งอาชีพ และการพัฒนาของประเทศ เพราะคอมพิวเตอร์ช่วยสร้างนวัตกรรมใหม่ๆ

ไมโครซอฟท์ เป็นหนึ่งในองค์กรระดับโลกด้านเทคโนโลยีที่ให้ความสำคัญกับกิจกรรมการเขียนภาษาโค้ดอย่างต่อเนื่องทั้งในระดับโลก ระดับภูมิภาค และในประเทศไทย ที่ผ่านมา บริษัทไมโครซอฟท์ (ประเทศไทย) จำกัด ได้มีการจัดกิจกรรมส่งเสริมการเขียนโค้ดตั้งแต่เบื้องต้น อบรมการเขียนโค้ดอย่างง่ายโดยไม่จำเป็นต้องมีความรู้พื้นฐานในการเขียนโค้ด

คู่มือการเขียนโปรแกรมเบื้องต้นนี้ เป็นส่วนหนึ่งของโครงการ Microsoft YouthSpark ที่มีวัตถุประสงค์ในการส่งเสริมการเรียนรู้วิทยาศาสตร์คอมพิวเตอร์ด้วยกิจกรรมพัฒนาทักษะพื้นฐานการเขียนโค้ดแก่เด็กและเยาวชนไทยประกอบด้วยกิจกรรมหลากหลาย เลือกรวมมาเพื่อเตรียมความพร้อมด้านตรรกะและทักษะการแก้ปัญหา สร้างเสริมทักษะการเขียนโปรแกรมพื้นฐาน ช่วยจุดประกายความคิดสร้างสรรค์ และสร้างแรงบันดาลใจผ่านเกมส์ เนื้อหาของกิจกรรมได้รวบรวมจาก Code.org และ <https://www.touchdevelop.com/> ซึ่งเป็นแหล่งเรียนรู้ออนไลน์ฟรีสำหรับผู้สนใจพัฒนาทักษะด้านการเขียนโค้ด คู่มือนี้จัดทำขึ้นเพื่อให้คุณครู ผู้ปกครอง ชมรมต่างๆ หรือแม้แต่ศูนย์การเรียนรู้ชุมชน สามารถนำไปจัดกิจกรรมการสอนและขยายผลได้ง่ายซึ่งเหมาะสำหรับผู้เรียนอายุตั้งแต่ 6 ปีขึ้นไป

คู่มือนี้รวบรวมและพัฒนาโดยคณะทำงานหลายทีม ได้แก่ Good Factory, Saturday School และอื่นๆ ด้วยทุนสนับสนุนจากไมโครซอฟท์

กิจกรรมภาพรวม

กิจกรรม	ระยะเวลา
กิจกรรมอุ่นเครื่อง: สร้างแรงบันดาลใจผ่านวิดีโอ #WeSpeakCode in Thailand	5-10 นาที
กิจกรรม 1: คอมพิวเตอร์รอบตัวเรา Computer Around You	15-20 นาที
กิจกรรม 2: เขียนโปรแกรมกระดาษกราฟ Graph Paper Programming	15-20 นาที
กิจกรรม 3: เรียนหลักการเขียนโปรแกรมผ่านเกมส์ Hour of Code	50-60 นาที
กิจกรรม 4: ทดลองเขียนโปรแกรมด้วยตนเอง Flatverse โดยใช้ Microsoft Touch Develop	50-60 นาที

กิจกรรมอุ่นเครื่อง

#WeSpeakCode in Thailand



#WeSpeakCode

ระยะเวลา

5 - 10 นาที

อุปกรณ์

คอมพิวเตอร์ที่เชื่อมต่อระบบอินเทอร์เน็ต

ลำโพง

โปรเจ็คเตอร์

จุดประสงค์

เพื่อจุดประกายและสร้างแรงบันดาลใจให้แก่ผู้เรียน และเตรียมพร้อมเข้าสู่บทเรียน

ขั้นตอนการจัดกิจกรรม

1. เปิดวิดีโอจากลิงค์:

https://www.youtube.com/watch?v=iwxazWl_nQI

2. ผู้สอนถามความเห็นจากผู้เรียนสะท้อนว่าได้รับแรงบันดาลใจจากวิดีโอเมื่อสักครู่นี้อย่างไรบ้าง เปิดโอกาสให้ผู้เรียนแบ่งปันเรื่องราว ประสบการณ์ส่วนตัวที่เกี่ยวข้องเพื่อแลกเปลี่ยนเรียนรู้กับชั้นเรียน

กิจกรรม 1

คอมพิวเตอรืรอบตัวเรา



ระยะเวลา

15 - 20 นาที

อุปกรณ์

กระดาษ Post-it ไม่จำกัดจำนวน

ปากกามาร์คเกอร์ คนละ 1 แท่ง

จุดประสงค์

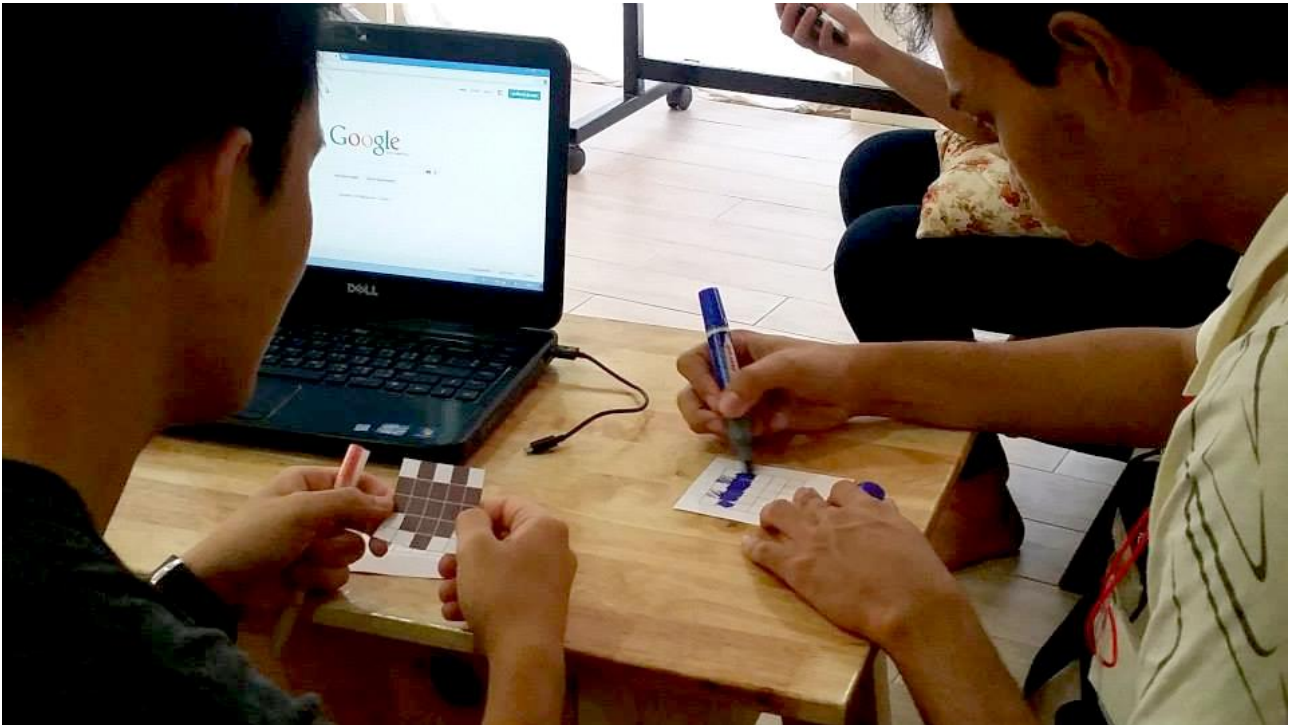
เพื่อให้ผู้เรียนทำความเข้าใจกับคอมพิวเตอร์ในชีวิตประจำวัน สามารถแยกแยะและชี้เฉพาะหน้าที่ของคอมพิวเตอร์ในสิ่งของเครื่องใช้รอบ ๆ ตัวได้

ขั้นตอนการจัดกิจกรรม

1. ผู้สอนแบ่งผู้เรียนเป็นกลุ่มย่อย กลุ่มละ 4-5 คน
2. แจกกระดาษ Post-it ให้กลุ่มละ 1-2 เล่ม และปากกามาร์กเกอร์คนละ 1 แท่ง
3. ผู้สอนยกตัวอย่างสิ่งของรอบ ๆ ตัว และอธิบายประกอบ เช่น เครื่องคิดเลข มีคอมพิวเตอร์ช่วยในการคำนวณ เป็นต้น
4. ให้ผู้เรียนระดมความคิด พุดคุยแลกเปลี่ยนในกลุ่มย่อย และเขียนไอเดียลงในกระดาษ Post-it โดยให้แต่ละกลุ่มคิดสิ่งของเครื่องใช้ในชีวิตประจำวันที่มีคอมพิวเตอร์เป็นส่วนประกอบ
จับเวลา 10 - 15 นาที
5. ผู้สอนให้ผู้เรียนแต่ละกลุ่มแลกเปลี่ยนเรียนรู้กับเพื่อนๆ ยกตัวอย่างไอเดียที่น่าสนใจ ให้ผู้เรียนอธิบายให้ฟัง
6. หากผู้สอนมีตัวอย่างอื่นๆ ที่อาจไม่มีใครนึกถึง ลองยกตัวอย่างให้ผู้เรียนฟัง หรืออาจเป็นการต่อยอดจากไอเดียที่เสนอมมาแล้ว

กิจกรรม 2

เขียนโปรแกรมกระดาษกราฟ



ระยะเวลา

15 - 20 นาที

อุปกรณ์

กระดาษกราฟว่าง คนละ 1 ใบ กระดาษกราฟโจทย์ คนละ 1 ใบ
ปากกามาร์คเกอร์ คนละ 1 แท่ง(ดูภาคผนวก)

จุดประสงค์

เพื่อให้ผู้เรียนมีความรู้เกี่ยวกับระบบการทำงาน การใช้คำสั่ง และเข้าใจความสัมพันธ์และ
บทบาทของโปรแกรมเมอร์และคอมพิวเตอร์

ขั้นตอนการจัดกิจกรรม

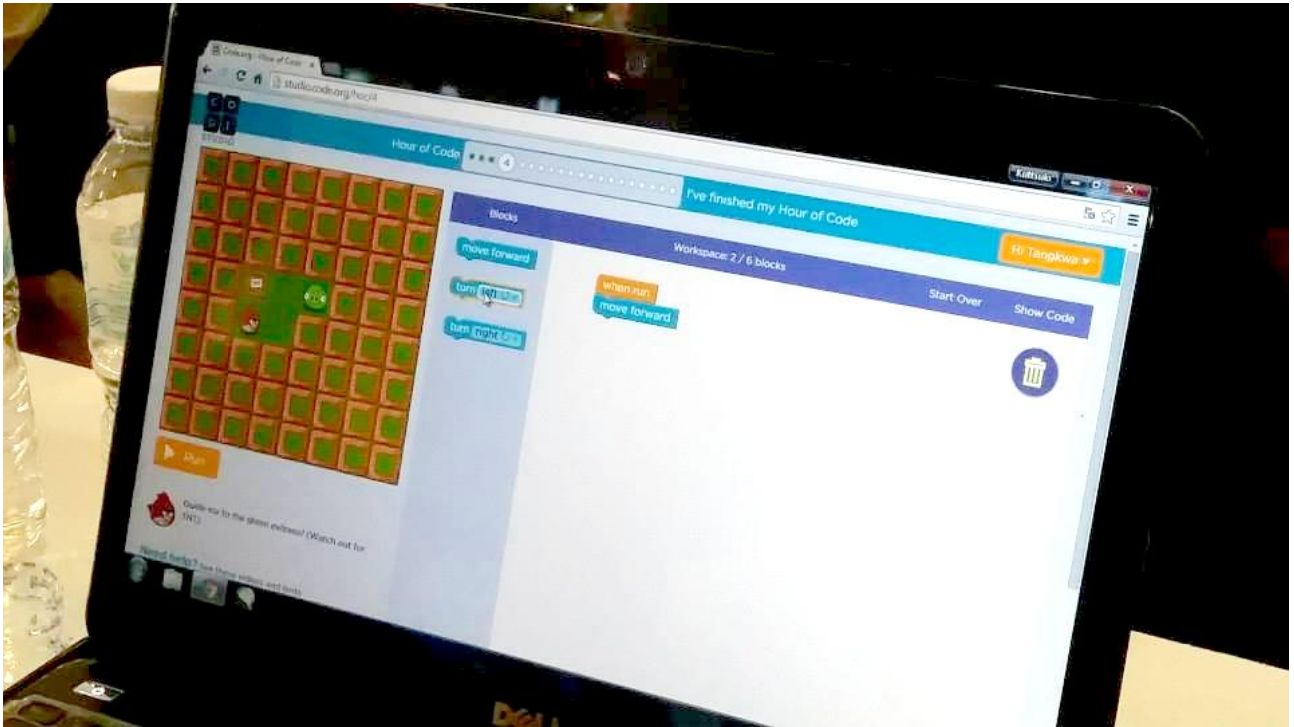
1. แจกกระดาษกราฟว่าง กระดาษกราฟโจทช์ และ ปากกามาร์คเกอร์ให้กับผู้เรียนทุกคน
2. ให้ผู้เรียนจับคู่กัน โดย ก.เป็นโปรแกรมเมอร์ (ผู้ออกคำสั่ง) และ ข.เป็นคอมพิวเตอร์ (ผู้รับคำสั่ง)
3. คำสั่งของโปรแกรมเมอร์มีทั้งหมด 3 คำสั่ง ได้แก่
 เลื่อนไปทางขวา ขึ้นบรรทัดใหม่ ฝนกระดาษ
โดยคำสั่งจะเริ่มจากช่องขวามือบนสุดก่อนเสมอ
4. จับเวลา 3 - 5 นาที เริ่มต้นการทำงานโดยโปรแกรมเมอร์เป็นผู้สั่งให้คอมพิวเตอร์ทำตามคำสั่ง ทั้ง 3 แบบ ตามโจทช์ที่กำหนด
5. เมื่อหมดเวลา ให้นักเรียนสลับบทบาทกันจับเวลา 3 - 5 นาที
6. ชวนผู้เรียนแลกเปลี่ยนความคิดเห็น ถามเพื่อให้ผู้เรียนได้สะท้อนและแบ่งปันความรู้สึกในการทำหน้าที่ทั้งบทบาทโปรแกรมเมอร์และคอมพิวเตอร์

กิจกรรม *Graph Paper Programming*

อ้างอิงมาจากคู่มือ *Unplugged Activity* พัฒนาโดย *code.org*

กิจกรรม3

เรียนหลักการเขียนโปรแกรมผ่านเกมส์



ระยะเวลา

50 - 60 นาที

อุปกรณ์

คอมพิวเตอร์ หรือแท็บเล็ต 1 เครื่องต่อผู้เรียน 1-2 คน

อินเทอร์เน็ต

โปรเจคเตอร์ (สำหรับผู้สอน)

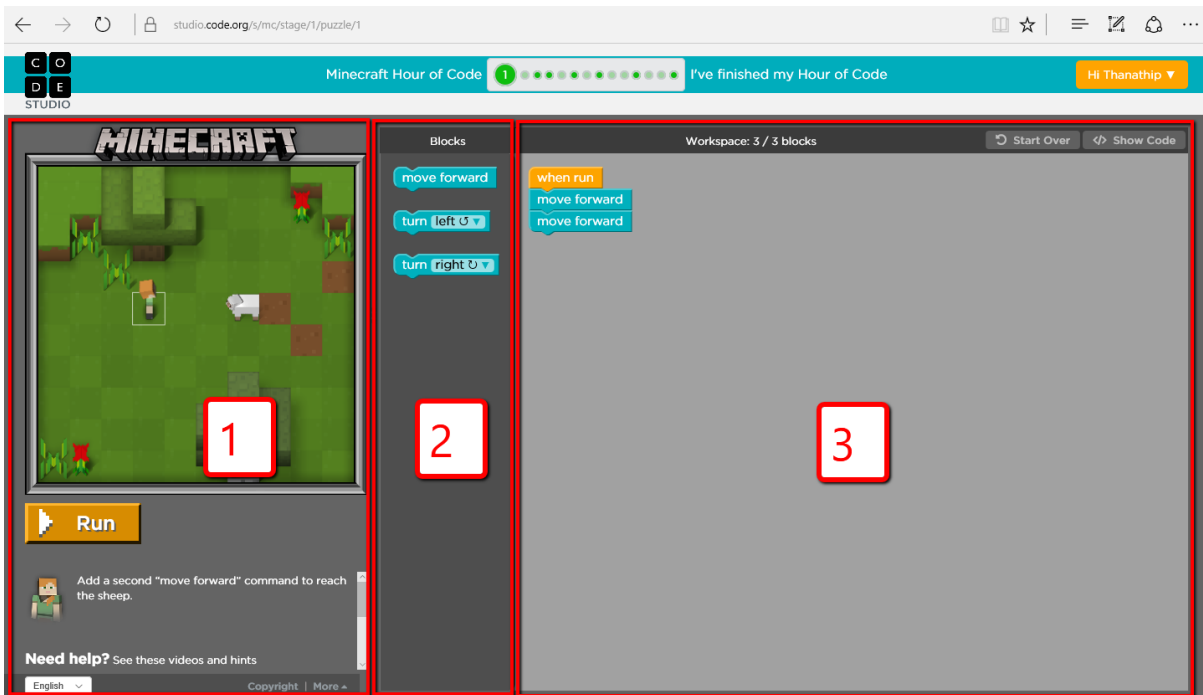
จุดประสงค์

เพื่อให้ผู้เรียนเข้าใจตรรกะพื้นฐานของการเขียนโปรแกรมเบื้องต้น ผ่านหลักสูตรเกมส์ Hour of Code พัฒนาโดย code.org

ขั้นตอนการจัดกิจกรรม

ใน Code.org คุณจะรู้พื้นฐานของวิทยาการคอมพิวเตอร์โดยการเขียนโปรแกรมให้ Alex หรือ Steve เคลื่อนไหวผ่าน โลก Minecraft ที่ได้จำลองขึ้นมา การเขียนโปรแกรมแบบดั้งเดิม มักจะอยู่ในรูปแบบของข้อความ แต่วันนี้เราจะใช้ "Blockly" ระบบซึ่งใช้บล็อกภาพที่คุณสามารถลากและวาง ในการเขียนโปรแกรม

ในสถานะจำลองต่อไปนี้เป็น สถานะที่คุณกำลังสร้างโค้ด Javascript แนวคิดที่คุณจะได้เรียนรู้คือ สิ่งที่นักโปรแกรมคอมพิวเตอร์ (Programmer) ใช้ทุกวันและเป็นรากฐานของวิทยาการคอมพิวเตอร์ก่อนที่คุณจะเริ่มต้น คุณจะต้องเลือกตัวละครของคุณ เช่น Alex ลองสร้างโค้ดสำหรับโปรแกรม ที่จะช่วยให้ Alex เคลื่อนไหวไปรอบๆ หน้าจอขอให้สนุกกับ Code.org



โดยหน้าจอกของคุณจะถูกแบ่งออกเป็นสามส่วนหลัก คือ

1. Play space หน้าจอด้านซ้าย เป็นพื้นที่เล่นเกมใน Minecraft ที่ซึ่ง โปรแกรมของคุณจะทำงาน ด้านล่างจะเป็น Instructions หรือคำแนะนำสำหรับการโปรแกรมแต่ละระดับที่จะเขียนไว้ที่ด้านล่างนี้
2. Tool box พื้นที่ตรงกลางนี้เป็นกล่องเครื่องมือและแต่ละบล็อกเหล่านี้ เป็นคำสั่งที่จะส่งผลต่อการกระทำของ Alex
3. Work Space พื้นที่สีขาวที่อยู่ด้านขวาเรียกว่า พื้นที่ทำงานและนี่คือบริเวณที่เราจะสร้างโปรแกรม

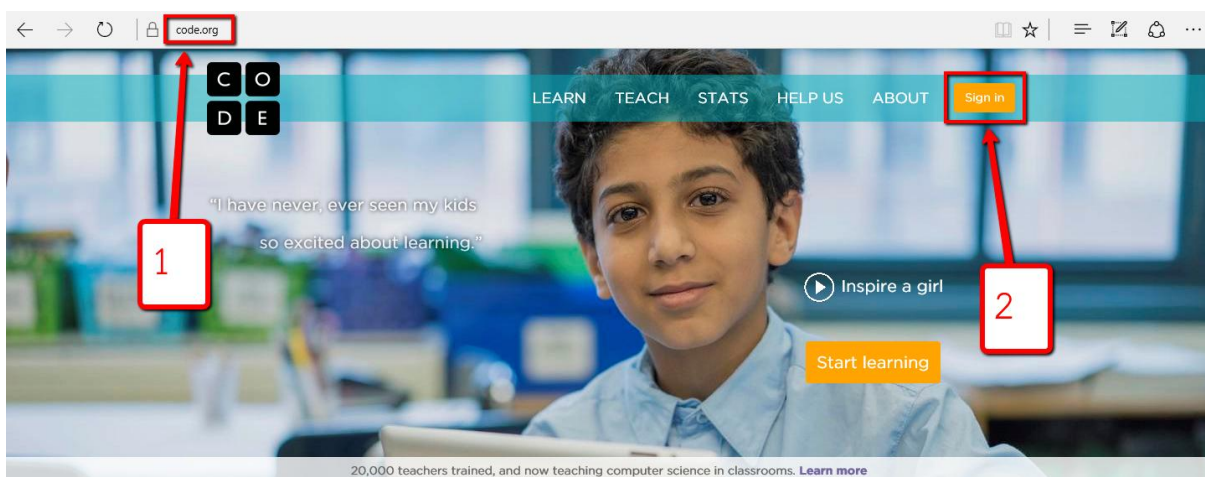
จะเกิดอะไรขึ้น ถ้าเราลากบล็อก `moveForward()`; ไปยังพื้นที่ทำงาน แล้วคลิก Run
 ผลที่ได้ : Alex เคลื่อนที่ไปข้างหน้าหนึ่งช่องในเส้นตาราง และหากเราต้องการที่จะทำอะไรบางอย่างหลังจากที่ Alex เคลื่อนไหวไปข้างหน้าในพื้นที่หนึ่งหน่วย เราสามารถเพิ่มบล็อกอื่นๆในโปรแกรมของเราได้อีก เช่นเลือกบล็อก `turnRight()`; และจะลากไปไว้ใต้บล็อก `moveForward()`; ในพื้นที่ทำงานจนเส้นสีส้ม ปรากฏขึ้น จากนั้นวางลงและบล็อกทั้งสองบล็อกจะรวมเข้าด้วยกัน เมื่อเรากดเรียกใช้งานอีกครั้ง Alex จะทำตามคำสั่งที่ซ้อนกันจากบนลงล่างในพื้นที่ทำงานของเรา

หากคุณต้องการที่จะลบบล็อก เพียงลากบล็อกนั้นออกมาจากกอง กลับไปที่กล่องเครื่องมือ หากต้องการยกเลิกการเปลี่ยนแปลงและกลับไปดูว่าระดับได้เริ่มต้นอย่างไร ให้ใช้ปุ่ม Start over ที่มุมขวาบนของพื้นที่ทำงานอีกหนึ่งข้อคือ คุณจะเห็นรูปสามเหลี่ยมเล็กๆ ในบล็อกคำสั่งสีเขียวซ้าย ขวาเวลาที่คุณเห็นรูปสามเหลี่ยมเหล่านี้ หมายความว่า คุณสามารถเลือกตัวเลือกที่ตรงกันข้ามกับคำสั่งปัจจุบันได้ที่นี้ เรามาเริ่มต้นลงมือเขียนโค้ดกันเลย!!!

ลงทะเบียนเป็นสมาชิกCode.org

1.เปิดไปยังเว็บไซต์ Code.org

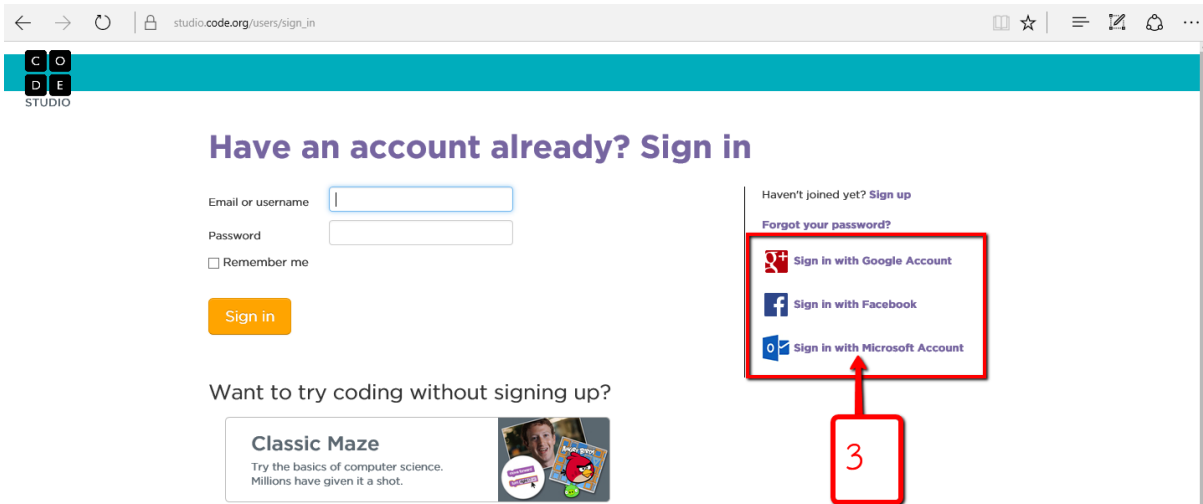
2.คลิกที่ปุ่ม Sign in



หมายเหตุ: หากมีตัวเลือกในการตั้งค่าภาษา ให้เลือก English แล้วคลิกที่ปุ่มSubmit

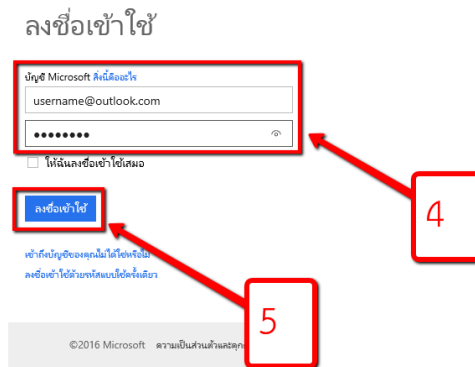


3.คลิกเลือก Sign in with Microsoft Account

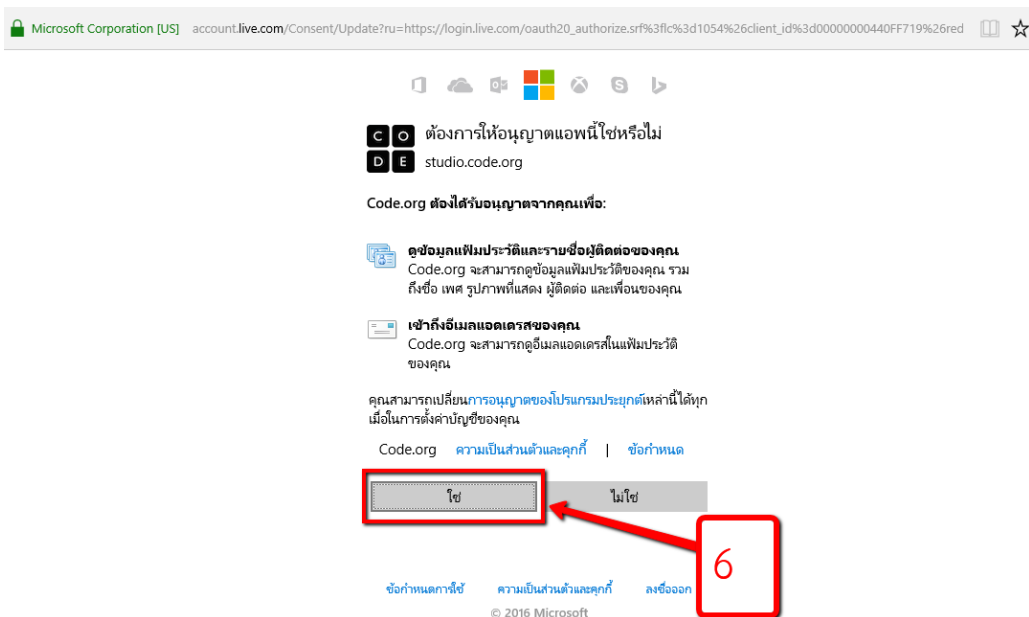


4.ลงชื่อเข้าใช้ด้วย email และ password ของคุณ

5.คลิกที่ปุ่ม ลงชื่อเข้าใช้



6.คลิกที่ปุ่ม ใช่ เพื่ออนุญาตให้แอปเข้าถึงข้อมูลของคุณ



7.ระบุอายุและเพศของคุณ

8.คลิกที่ปุ่ม Sign up

studio.code.org/users/sign_up

Student sign up for Code.org

Sign up for an account to track your progress. Your email is primarily used to allow you to log in and for password recovery. For more information, see our [Privacy Policy](#).

You can [browse the various stages and puzzles](#), but you will need to sign up to save your progress.

9,027,940 students have already signed up.

We need some additional information to continue signing you up.

- Age is required

Display Name:

Email:

Age:

Gender (optional):

I agree to be bound by the Code.org [Terms of Service](#). If I am under 13 years of age, I confirm by submitting this form that I have received my parent or legal guardian's permission to use the Code.org services.

For users under 13, email addresses are not stored in a form that allows us to contact them.

Are you a teacher?

Already signed up? [Sign in](#)

Forgot your password?

[Sign in with Google Account](#)

[Sign in with Facebook](#)

[Sign in with Microsoft Account](#)

เริ่มต้นเรียนรู้การเขียนโปรแกรมผ่านเกม Minecraft

9.คลิก Try now ที่เกม Minecraft เพื่อเริ่มทดลองเรียนรู้เรื่อง Code

studio.code.org

The Hour of Code for All Ages

Star Wars
Learn to program droids, and create your own Star Wars game in a galaxy far, far away.
[Try now](#)

Minecraft
Explore a Minecraft world through code.
[Try now](#)

Frozen
Let's use code to join Anna and Elsa as they explore the magic and beauty of ice.
[Try now](#)

Classic Maze
Try the basics of computer science. Millions have given it a shot.
[Try now](#)

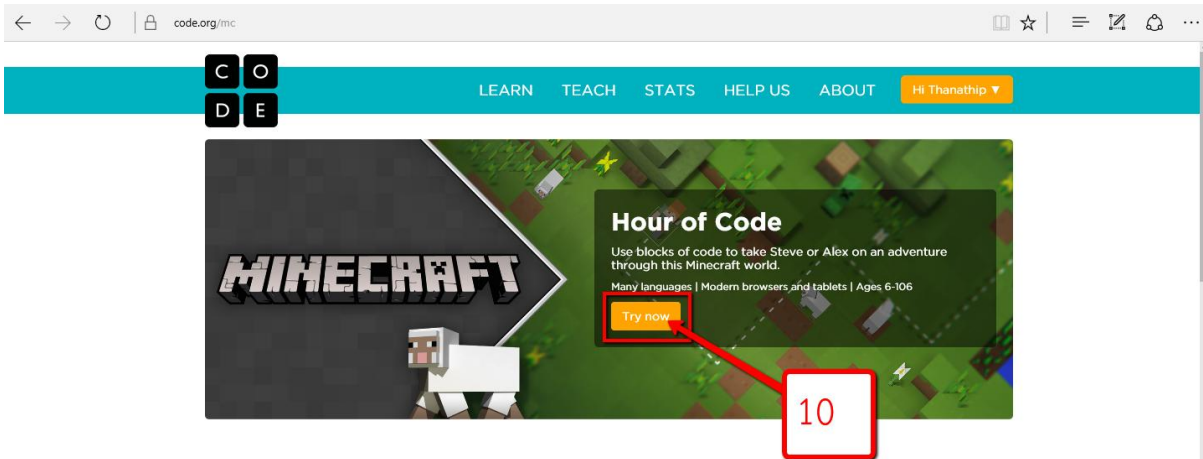
Flappy Code
Wanna write your own game in less than 10 minutes? Try our Flappy Code tutorial!
[Try now](#)

Infinity Play Lab
Use Play Lab to create a story or game starring Disney Infinity characters.
[Try now](#)

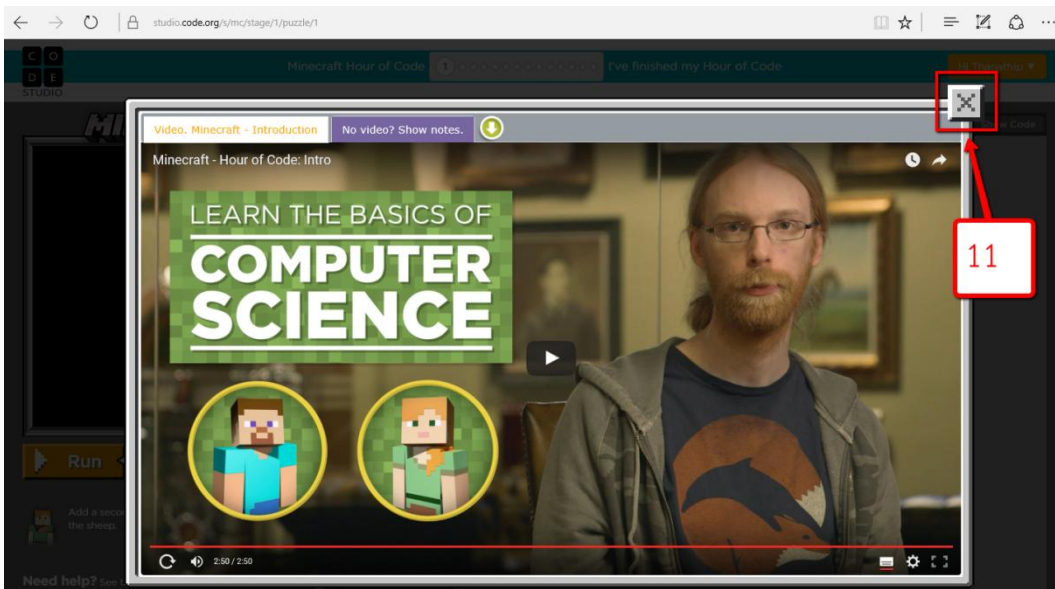
Play Lab
Create a story or make a game with Play Lab!
[Try now](#)

Artist
Draw cool pictures and designs with the Artist!
[Try now](#)

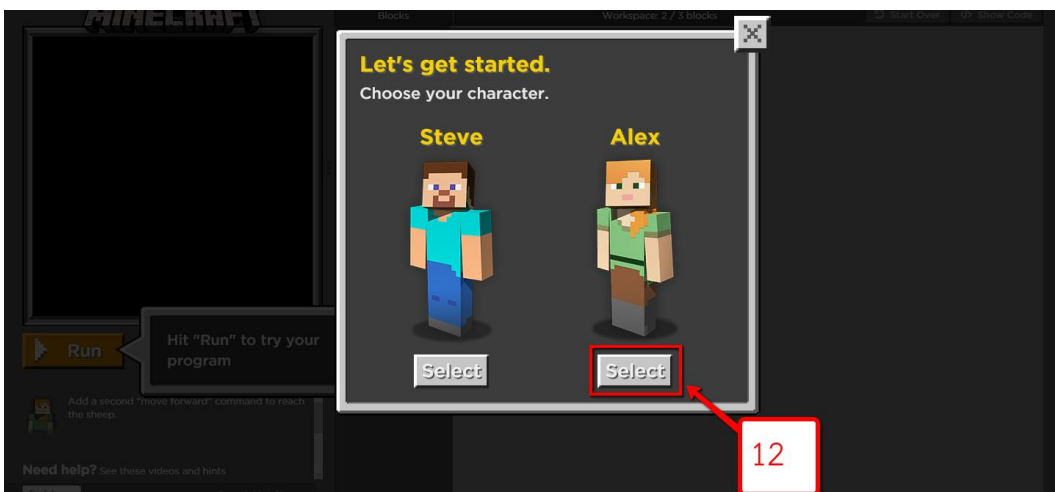
10.คลิกที่ปุ่ม Try now



11.จะมีวิดีโอแนะนำการใช้งาน ให้ดูจนจบและคลิกที่ปุ่ม ปิดวิดีโอ

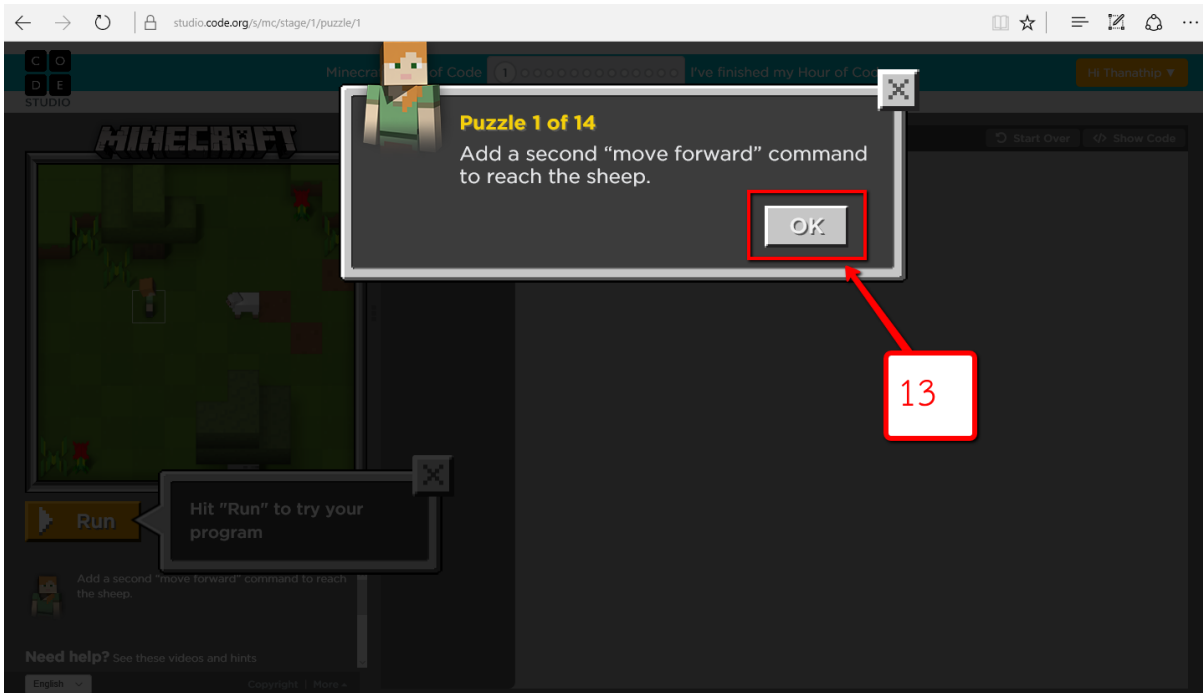


12.คลิกเลือกตัวละครของคุณ เช่น Alex



13. จะมีคำสั่งในการเล่นเกมให้อ่านทำความเข้าใจแล้วคลิกที่ปุ่ม OK

[ให้เพิ่มคำสั่ง “move forward” คำสั่งที่สองลงไป เพื่อเดินไปให้ถึงตัวแกะ]



14.คลิกลากคำสั่ง move forward ไปวางต่าคำสั่งที่มีอยู่

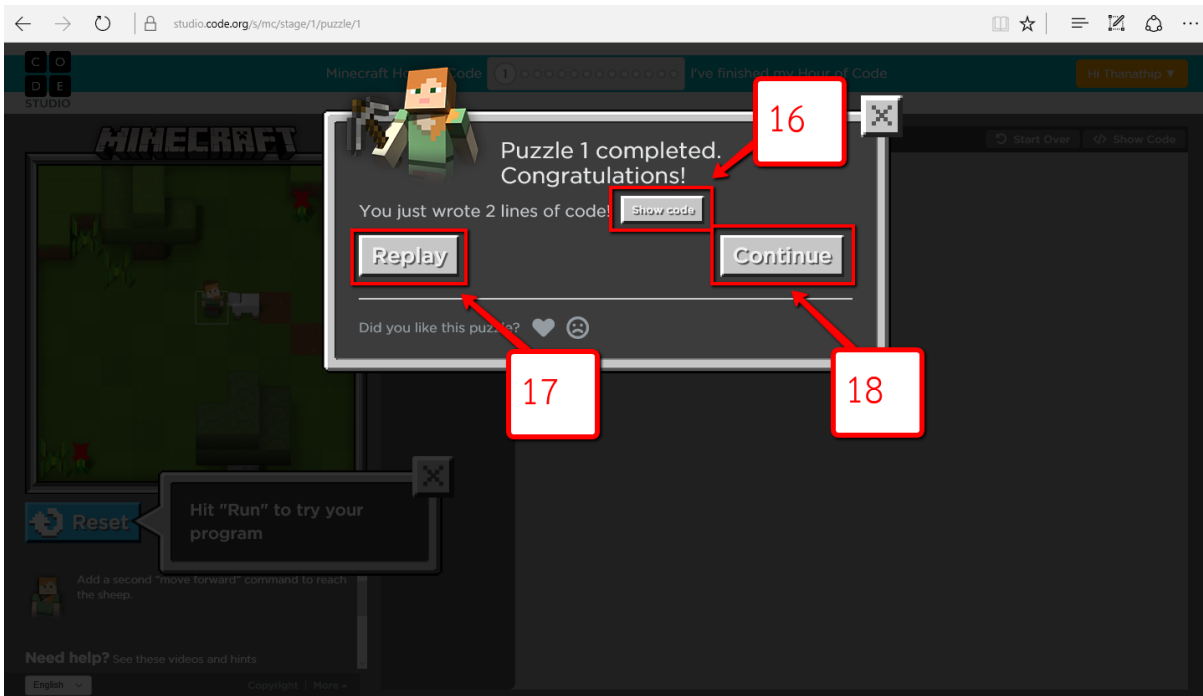
15.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง



16.คลิกที่ปุ่ม Show code เพื่อดูโค้ดคำสั่งที่เพิ่งใช้งาน

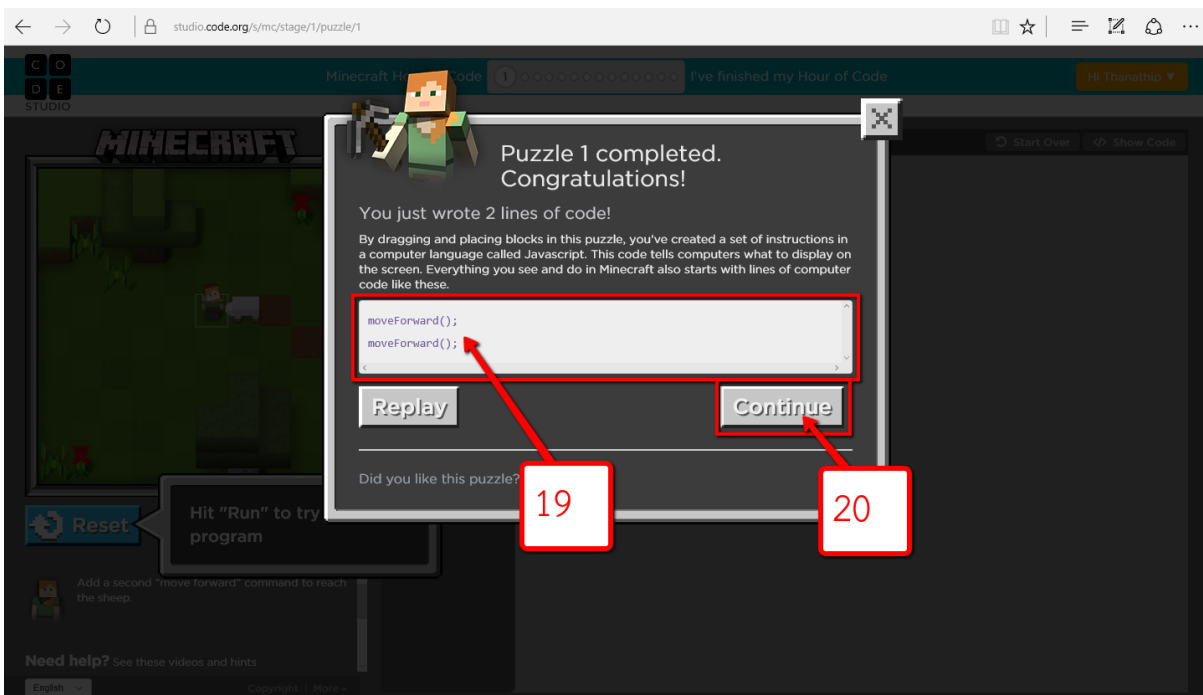
17.คลิกที่ปุ่ม Replay เพื่อกลับไปเล่นเกมอีกครั้ง

18.คลิกที่ปุ่ม Continue เพื่อเล่นเกมในด่านต่อไป



19.หน้าจอแสดงชุดคำสั่งที่เพิ่งใช้งาน

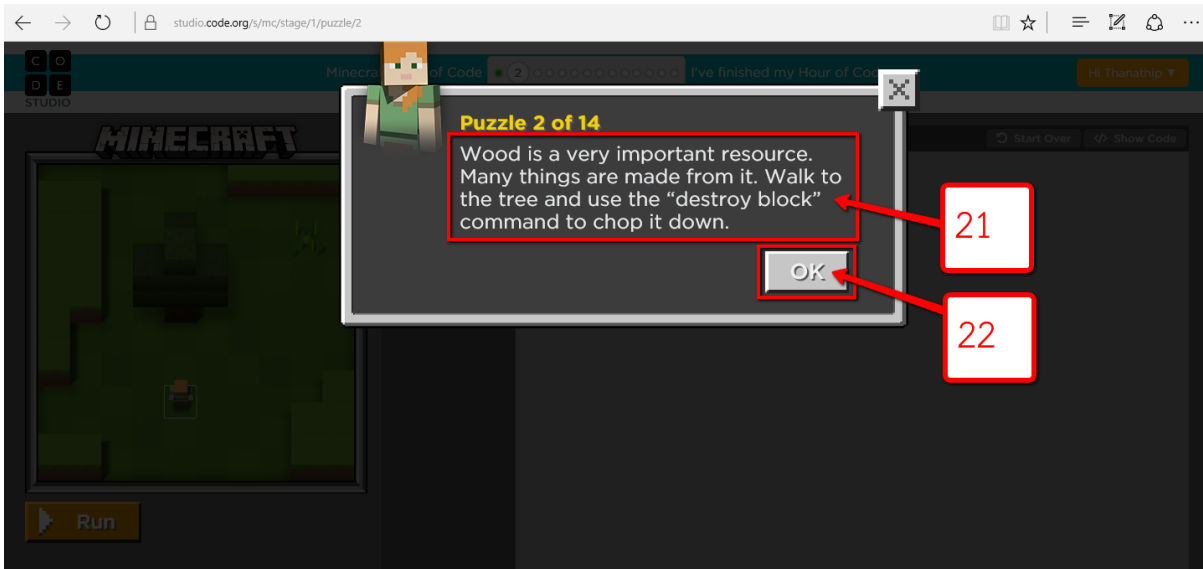
20.คลิกที่ปุ่ม Continue



21.อ่านคำสั่งสำหรับด่านต่อไป

[ไม้เป็นทรัพยากรที่มีค่า หลายๆ อย่างผลิตขึ้นมาจากไม้, ให้เดินไปยังต้นไม้แล้วใช้คำสั่ง “destroy block” เพื่อตัดต้นไม้]

22.คลิกที่ปุ่ม OK

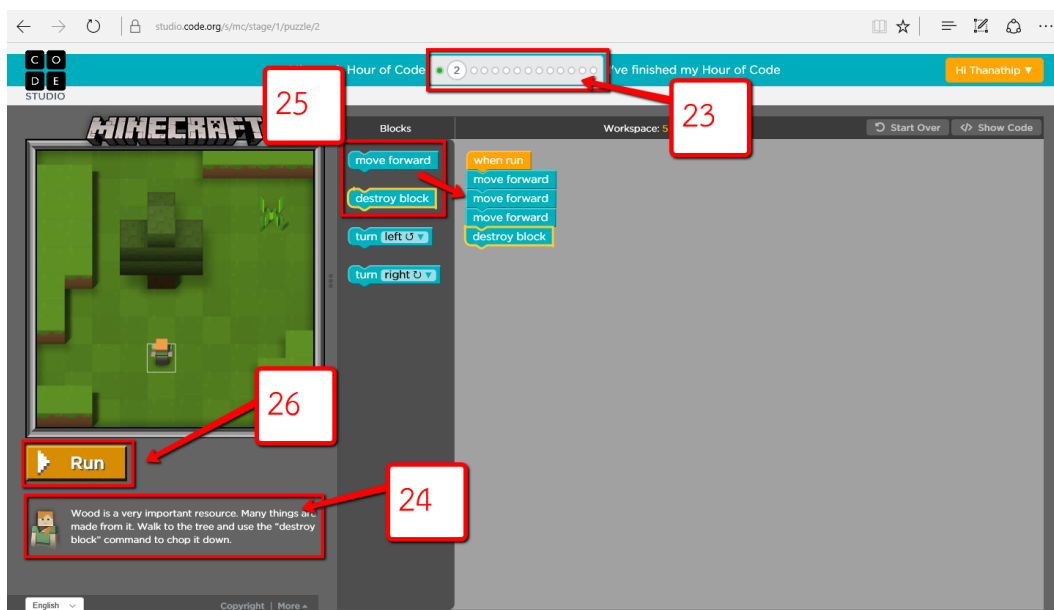


23.แถบแสดงจำนวนด่านในการเล่นเกม

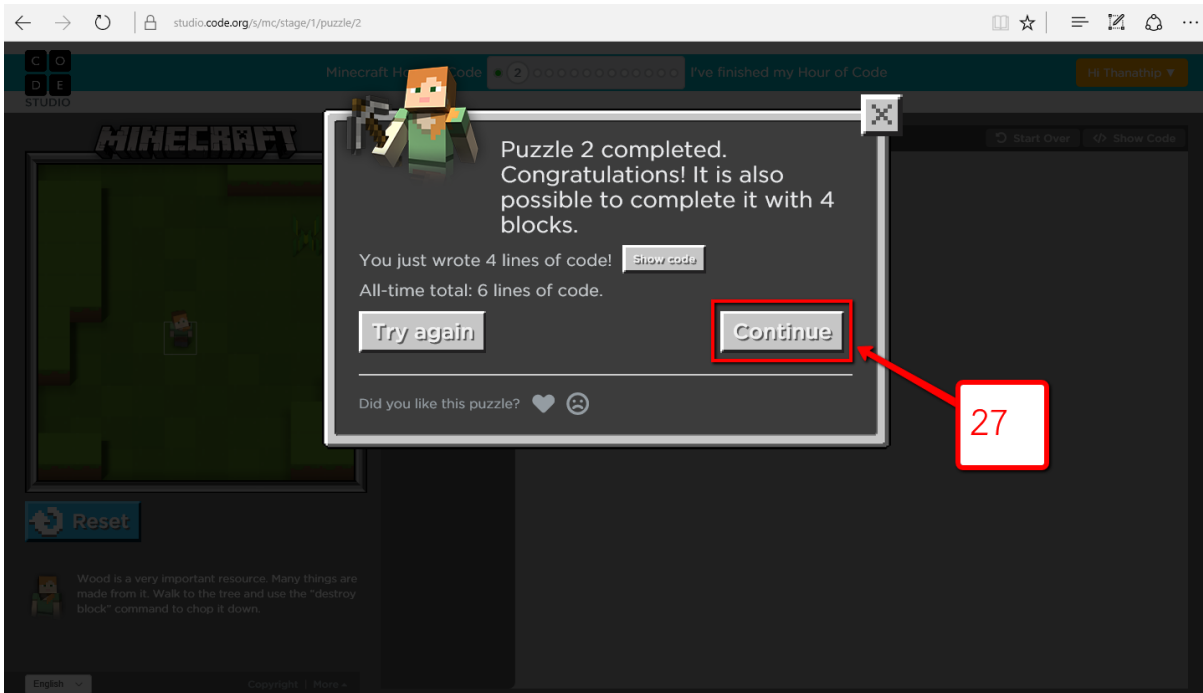
24.คำสั่งในการเล่นเกมแต่ละด่าน

25.ชุดคำสั่งที่ใช้ในการเล่นเกม

26.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง

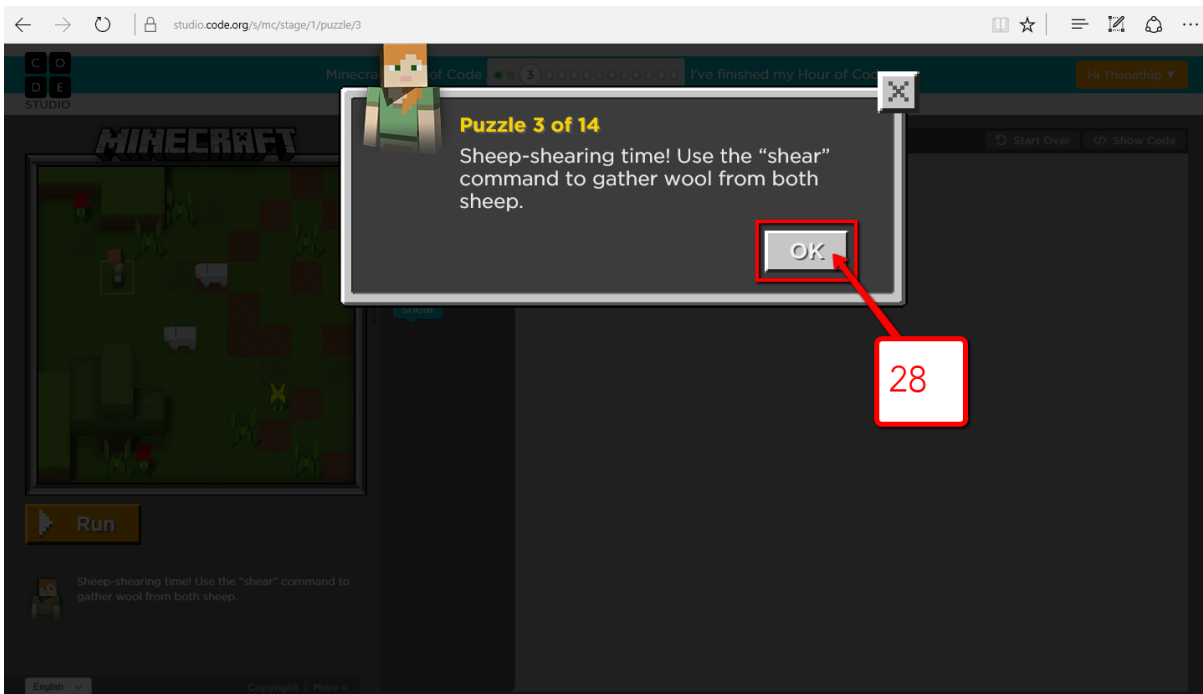


27.คลิกที่ปุ่ม Continue เพื่อไปยังด่านต่อไป



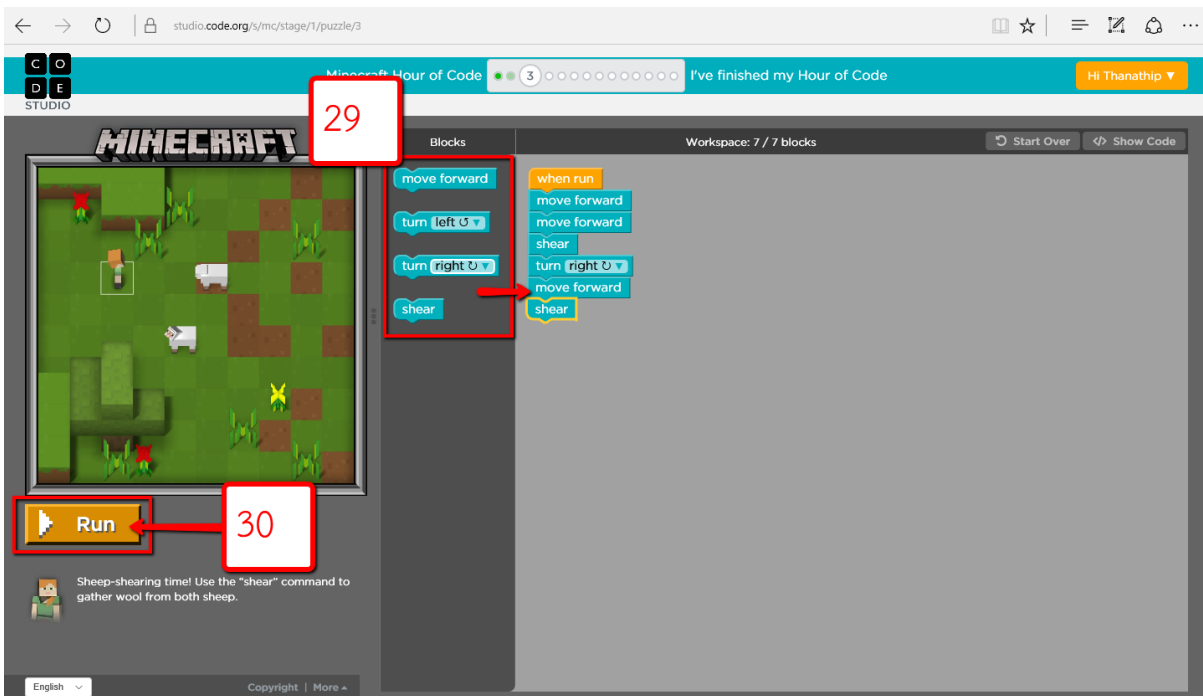
28.อ่านคำสั่ง แล้วคลิกที่ปุ่ม OK

[ถึงเวลาที่จะต้องตัดขนแกะแล้ว ใช้คำสั่ง “shear” เพื่อตัดขนแกะทั้งสองตัว]

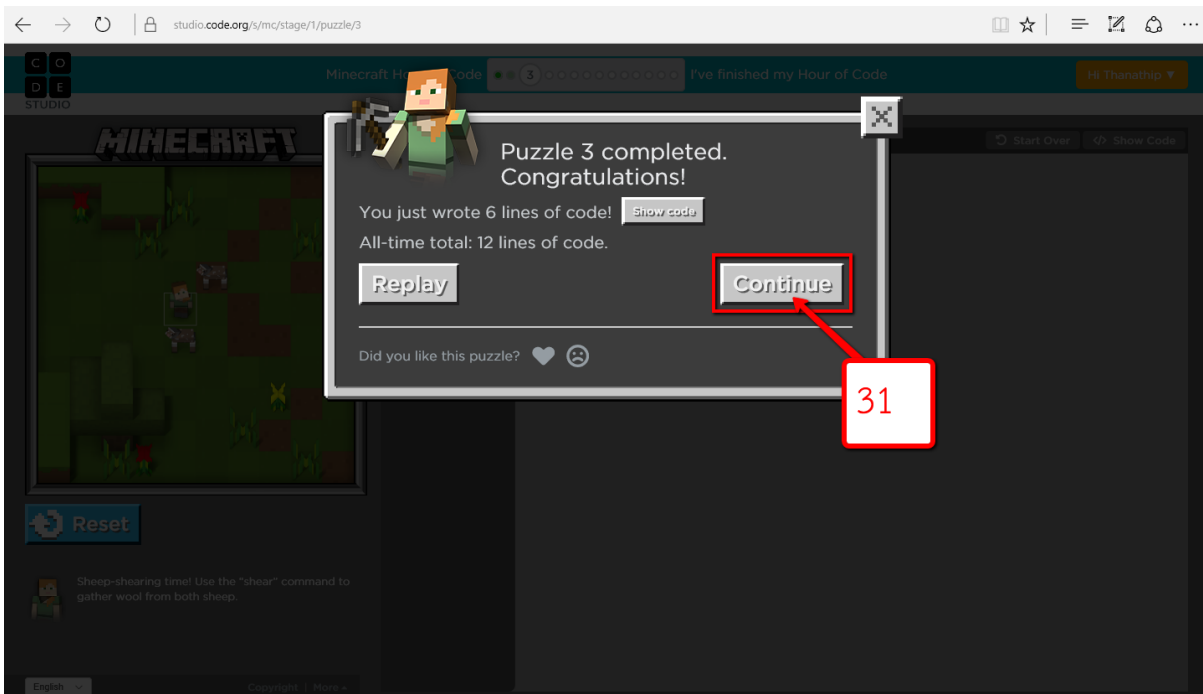


29.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

30.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง



31.คลิกที่ปุ่ม Continue

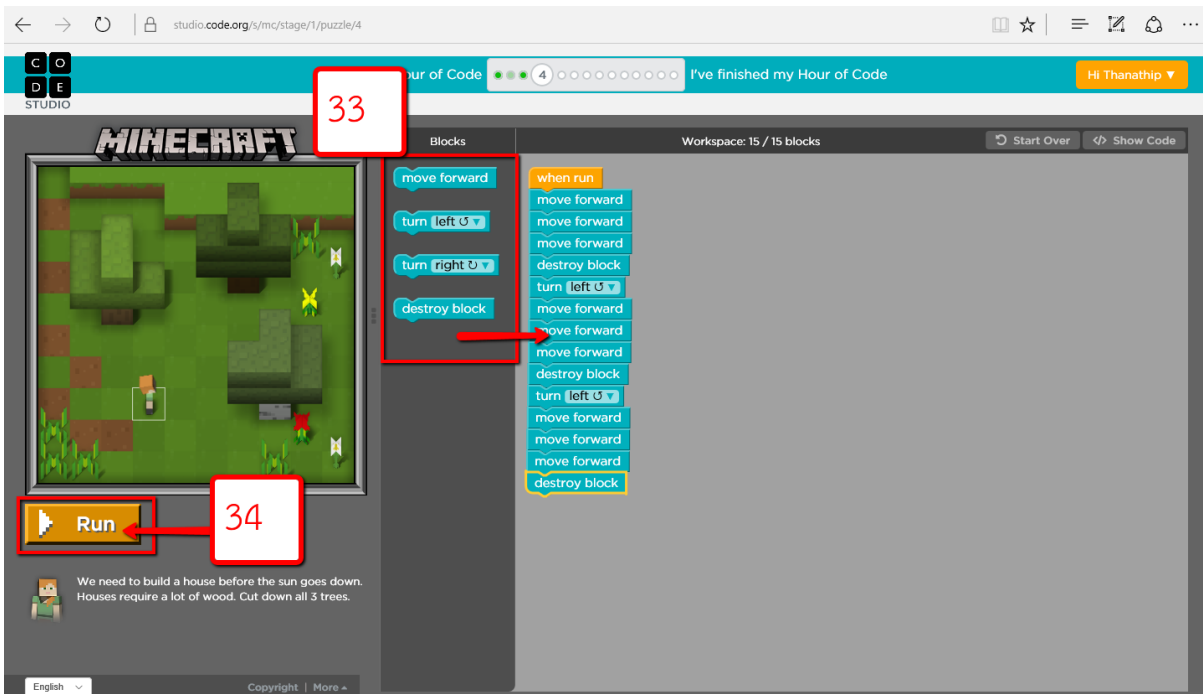


32.อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิกปุ่ม OK

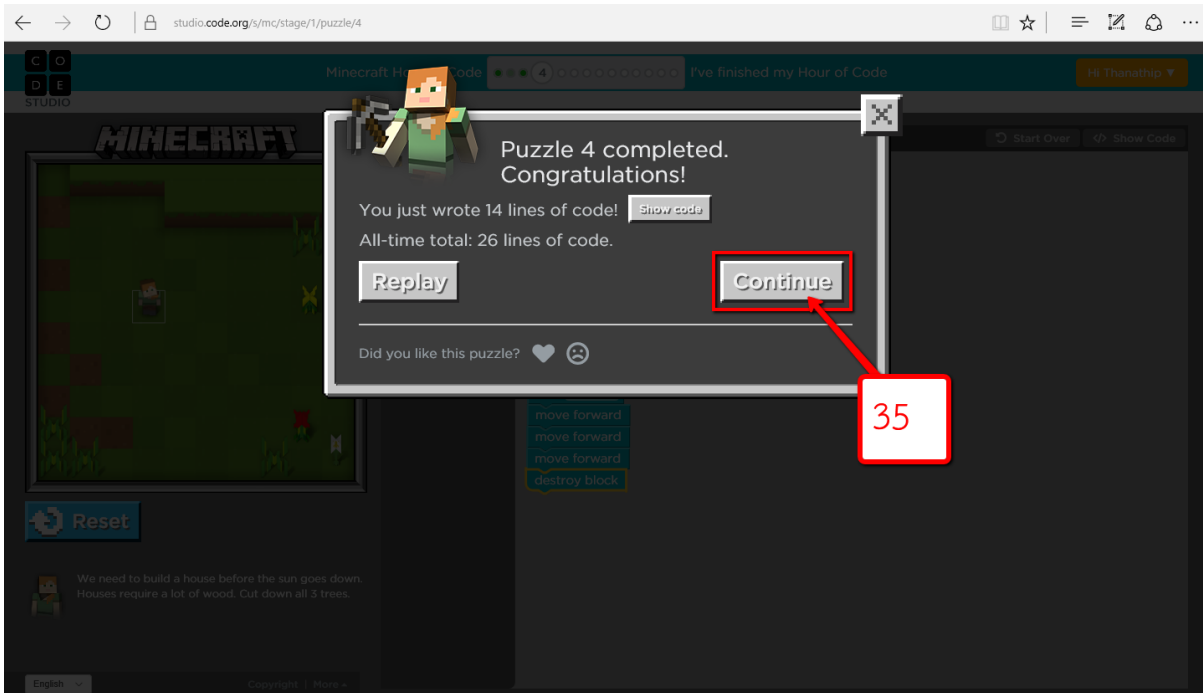


33.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

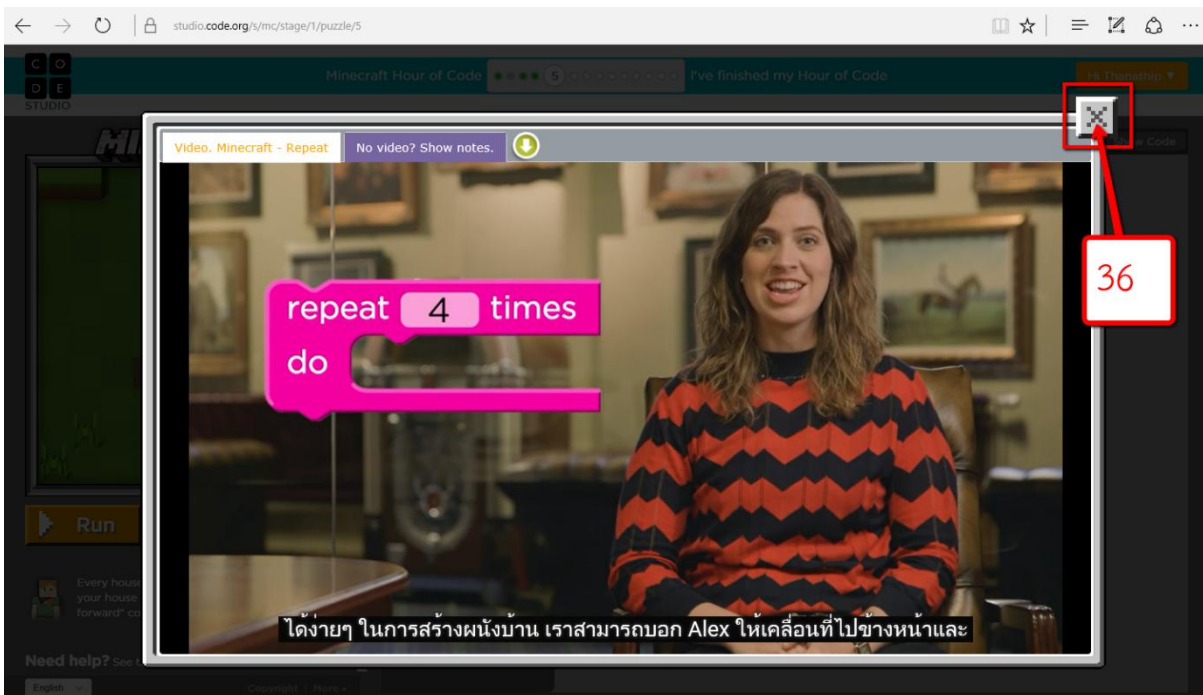
34.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง



35.คลิกที่ปุ่ม Continue



36.จะมีวิดีโอแนะนำการเล่นในระดับถัดไป ให้ดูจนจบแล้วคลิกปิดวิดีโอ

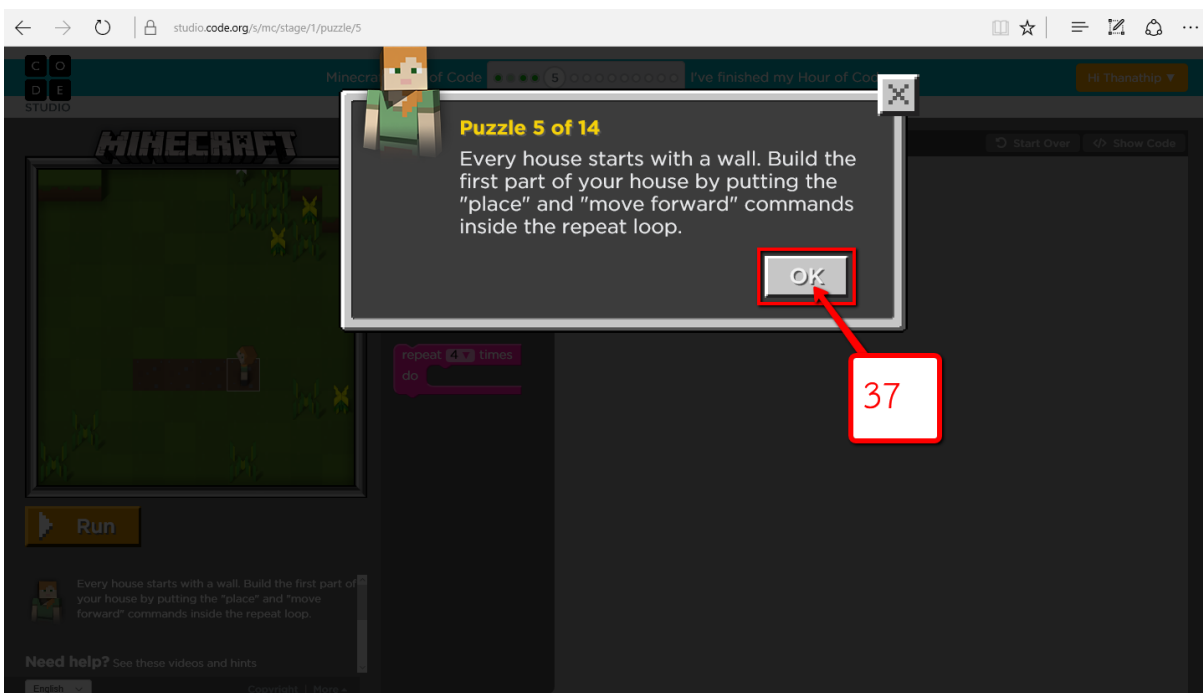


ในระดับที่ผ่านมา จำเป็นต้องใช้บล็อก `moveForward()`; จำนวนมาก และมันจะง่ายขึ้นถ้าเราเพียงแค่จะบอกคอมพิวเตอร์เพื่อดำเนินการตามคำสั่ง `moveForward()`;สี่หรือห้าครั้ง

โชคดีสำหรับเราที่คอมพิวเตอร์สามารถทำงานผ่านคำสั่งซ้ำได้ดีมาก ด้วยการสั่ง Repeat หรือวนรอบเมื่อสร้าง Minecraft เราใช้การวนรอบในการวางวัสดุเริ่มต้นทุกอย่างในการสร้าง โลกใหม่ ซึ่งสร้างได้เป็นพันๆบล็อกเรายังใช้การวนรอบในรูปแบบเล็กๆ อย่างเช่นการทำให้เท้าของ Alex ถอยกลับและไปข้างหน้าในเวลาเดินการวนรอบเป็นส่วนหนึ่งของการเขียนโปรแกรมที่มีประสิทธิภาพใกล้ค่าแล้ว ดังนั้นในระดับต่อไปอีกสองสามระดับเราจะสร้างบ้านเพื่อจะได้ได้อย่างปลอดภัย เรากำลังจะใช้บล็อกซ้ำเพื่อสร้างบ้านได้ง่ายๆในการสร้างผนังบ้าน เราสามารถบอก Alex ให้เคลื่อนที่ไปข้างหน้าและวางไม้กระดานสี่ครั้งหรือเราสามารถบอกให้เค้าเคลื่อนที่ไปข้างหน้าและวางไม้กระดานหนึ่งแผ่นแล้วใช้คำสั่งนี้และใช้บล็อกซ้ำเพื่อให้เค้าทำซ้ำหลายๆครั้งตอนนี้เราจะคลิกบนบล็อกซ้ำ และบอกเขาว่าเราต้องการให้เขาทำแบบนี้กี่ครั้งตอนนี้เรามาส่งบ้านก่อนจะถึงเวลากลางคืนกันเถอะ! ขอให้สนุกกันนะทุกคน

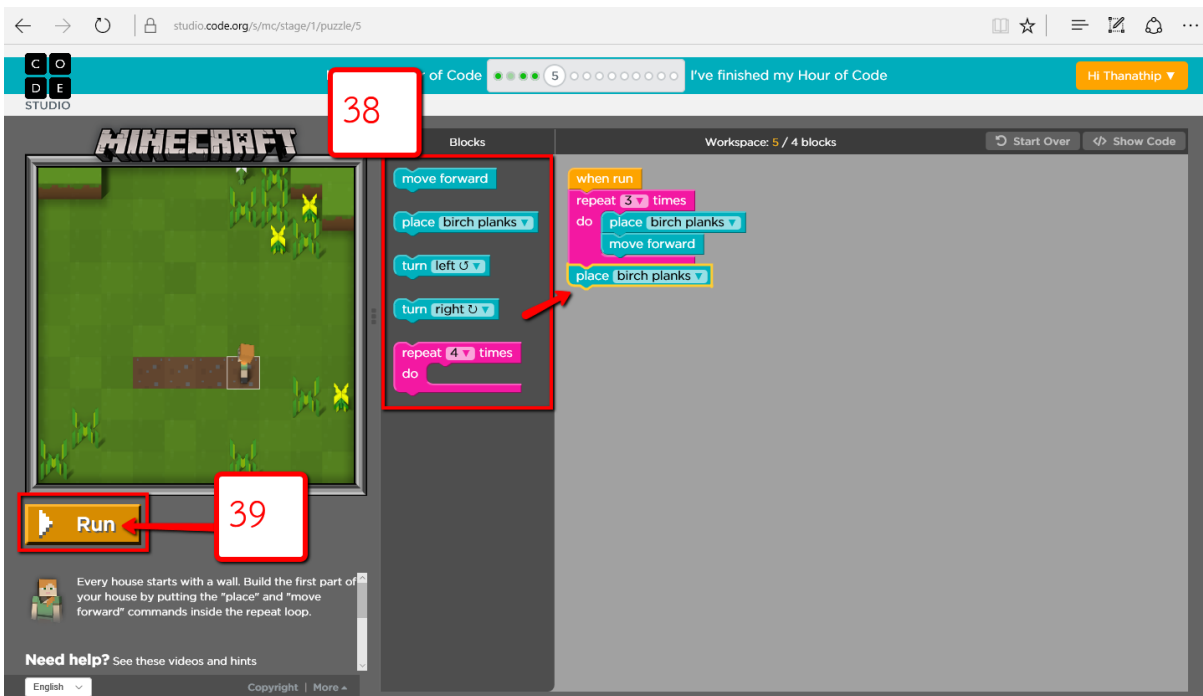
37.อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิก OK

[บ้านทุกหลังเริ่มสร้างขึ้นด้วยผนัง เริ่มสร้างส่วนแรกของบ้านด้วยการวางคำสั่ง “place” และ “move forward” ภายในกรอบคำสั่ง repeat]

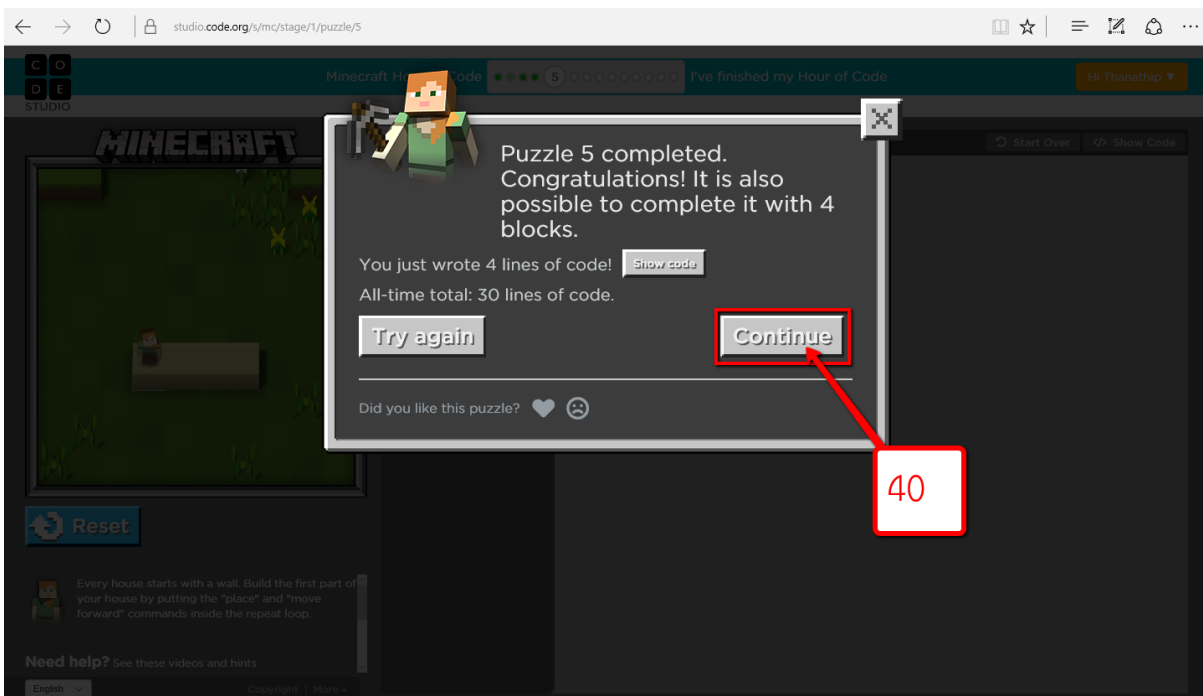


38.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

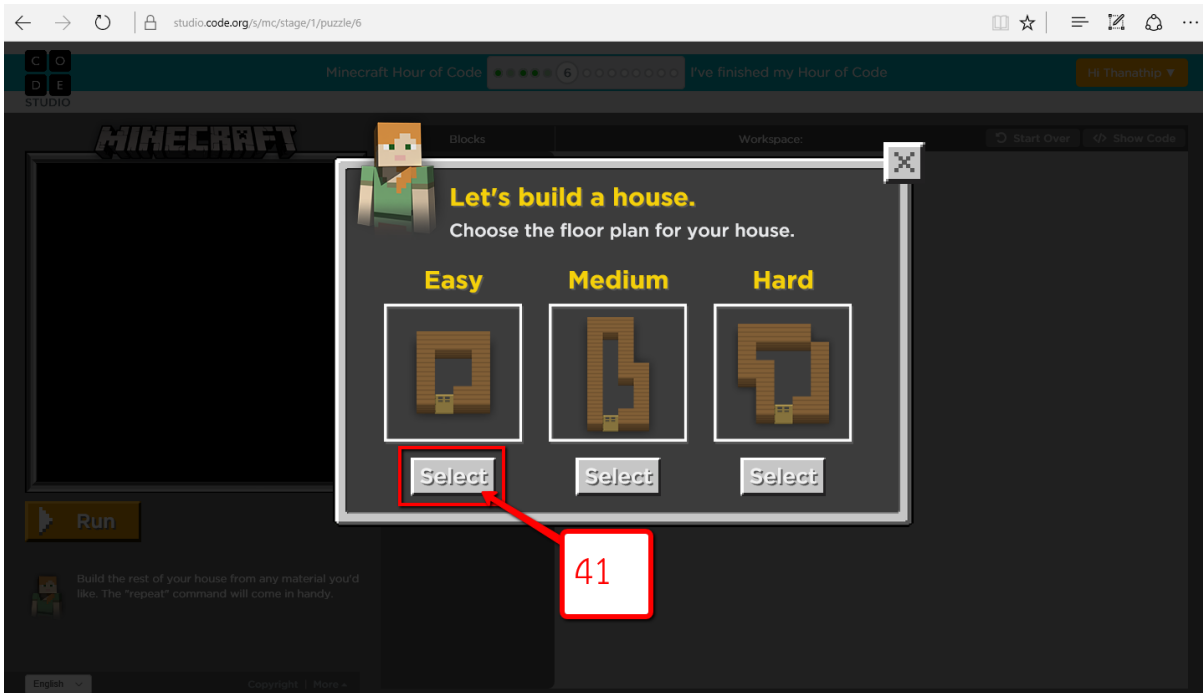
39.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง



40.คลิกที่ปุ่ม Continue

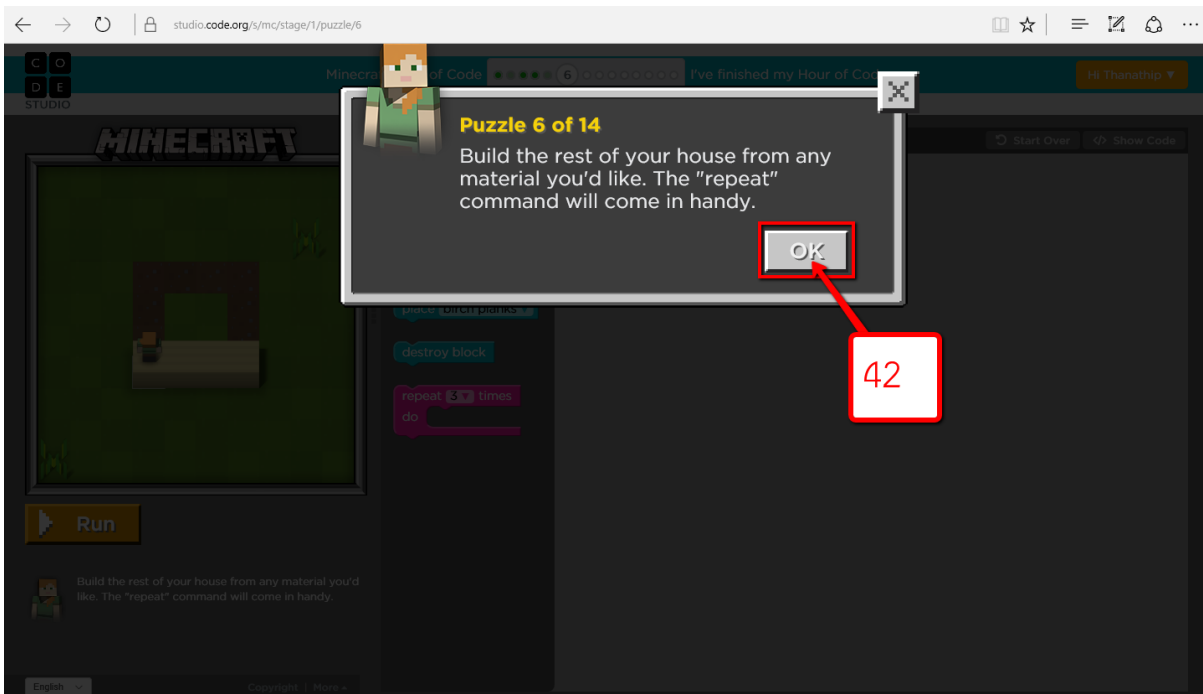


41.คลิกที่ปุ่ม Select เพื่อเลือกรูปแบบความยากง่ายในการสร้างบ้าน



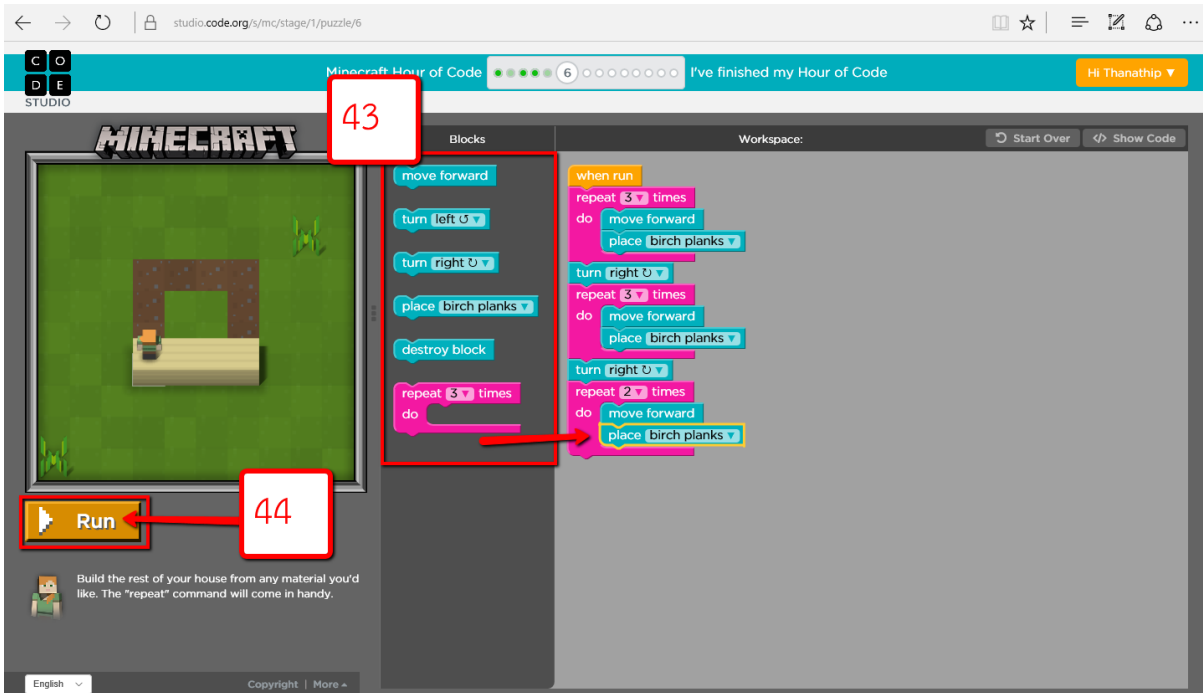
42.อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิก OK

[สร้างบ้านของคุณด้วยวัสดุใดก็ได้ตามต้องการ โดยใช้คำสั่ง “repeat”ช่วย]

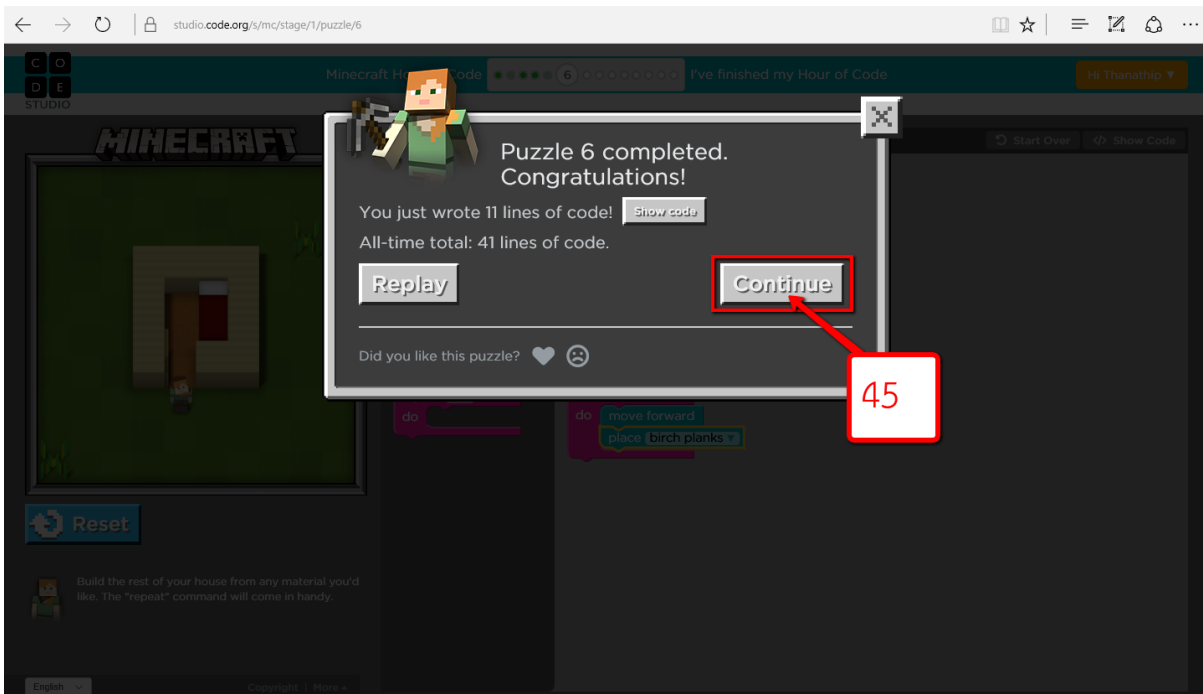


43.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

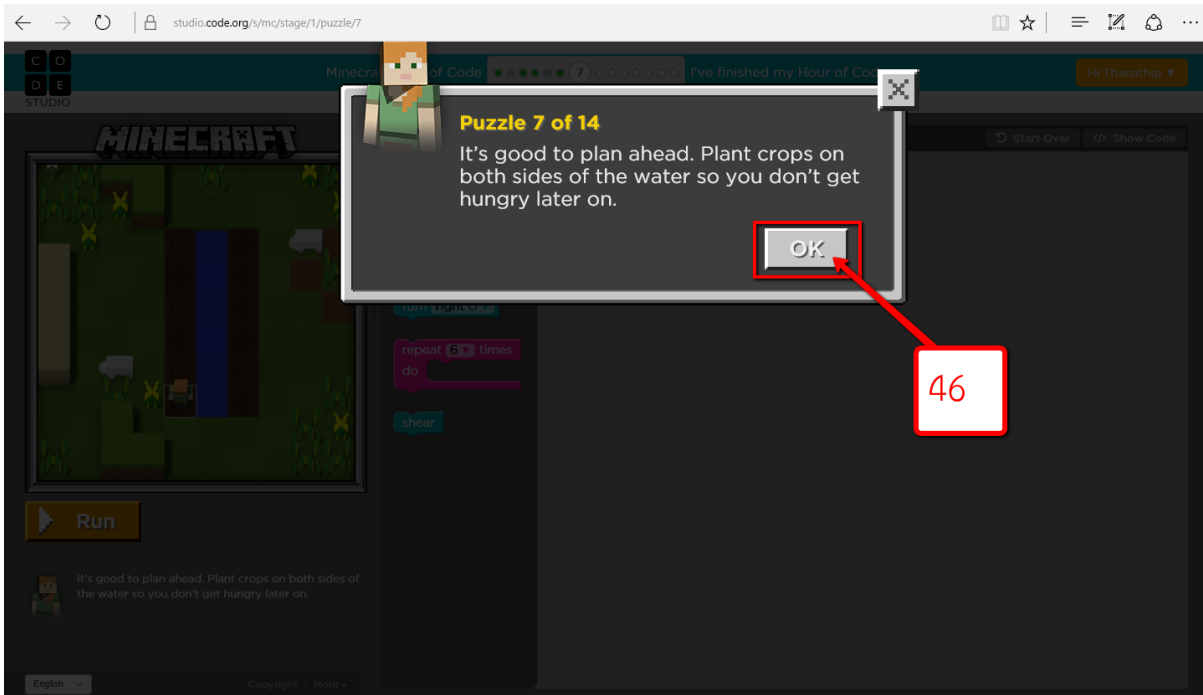
44.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง



45.คลิกที่ปุ่ม Continue

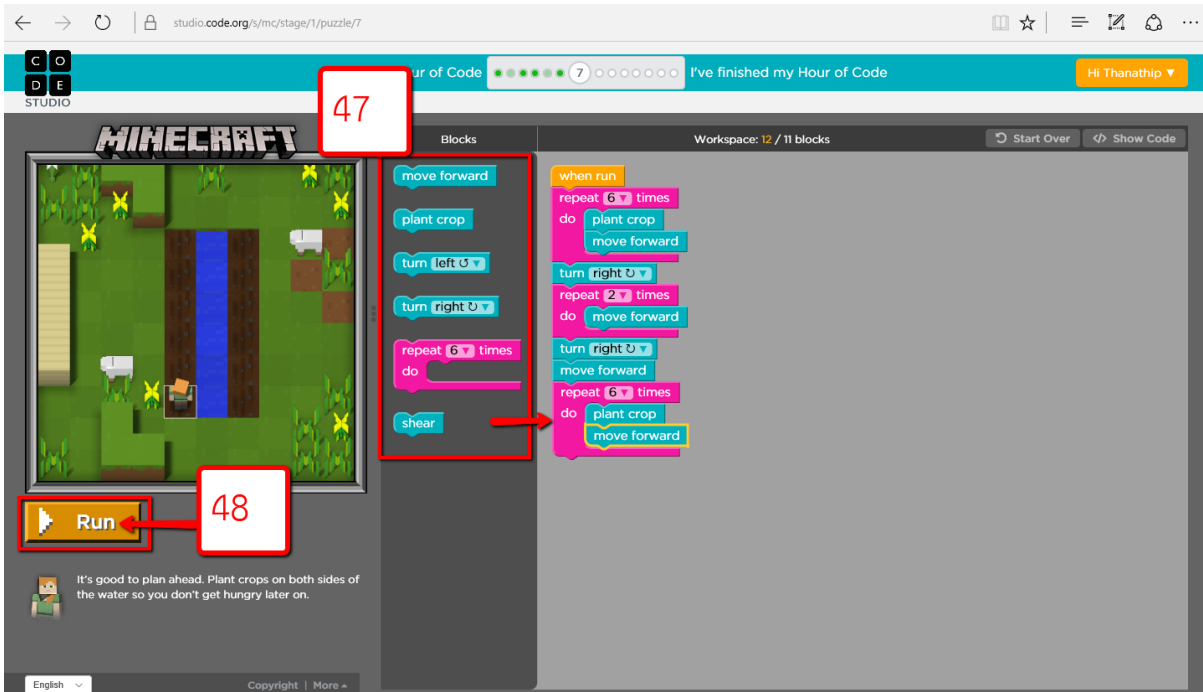


46.อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิก OK

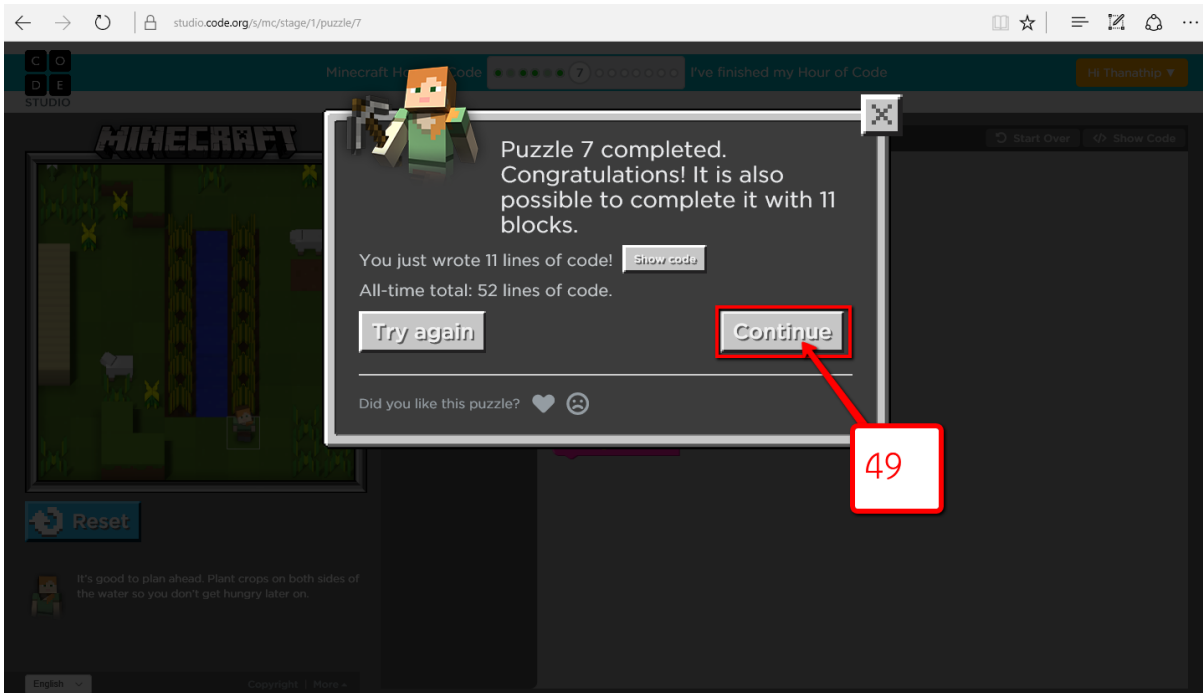


47.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

48.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง

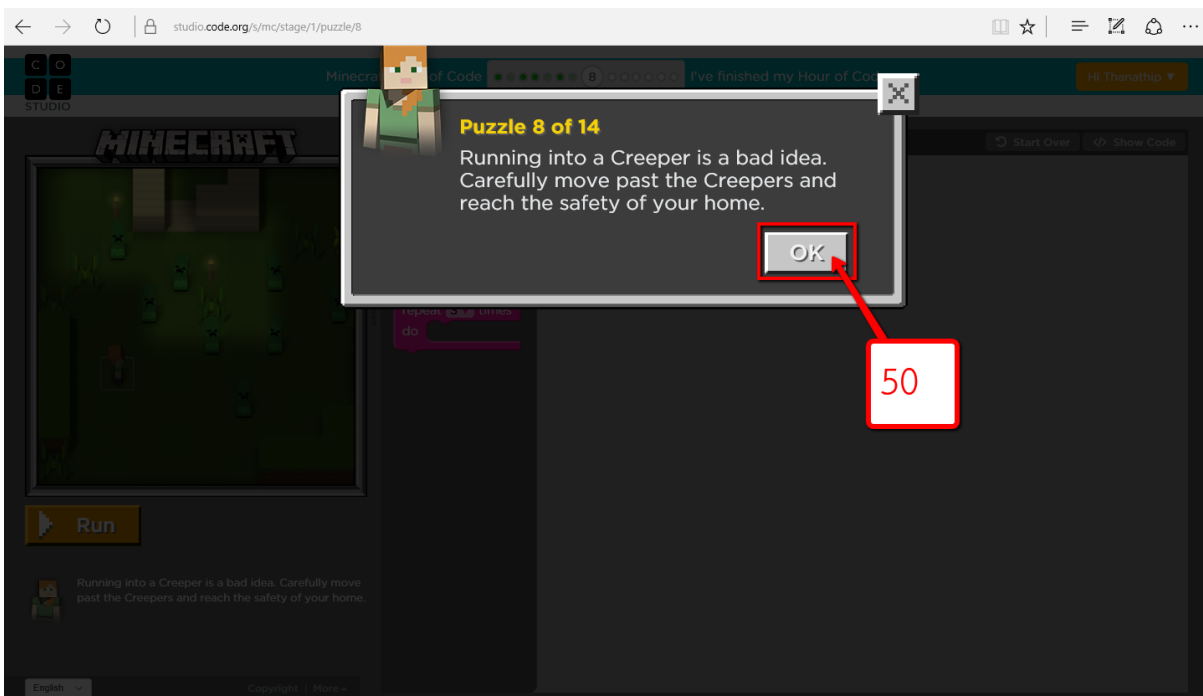


49.คลิกที่ปุ่ม Continue



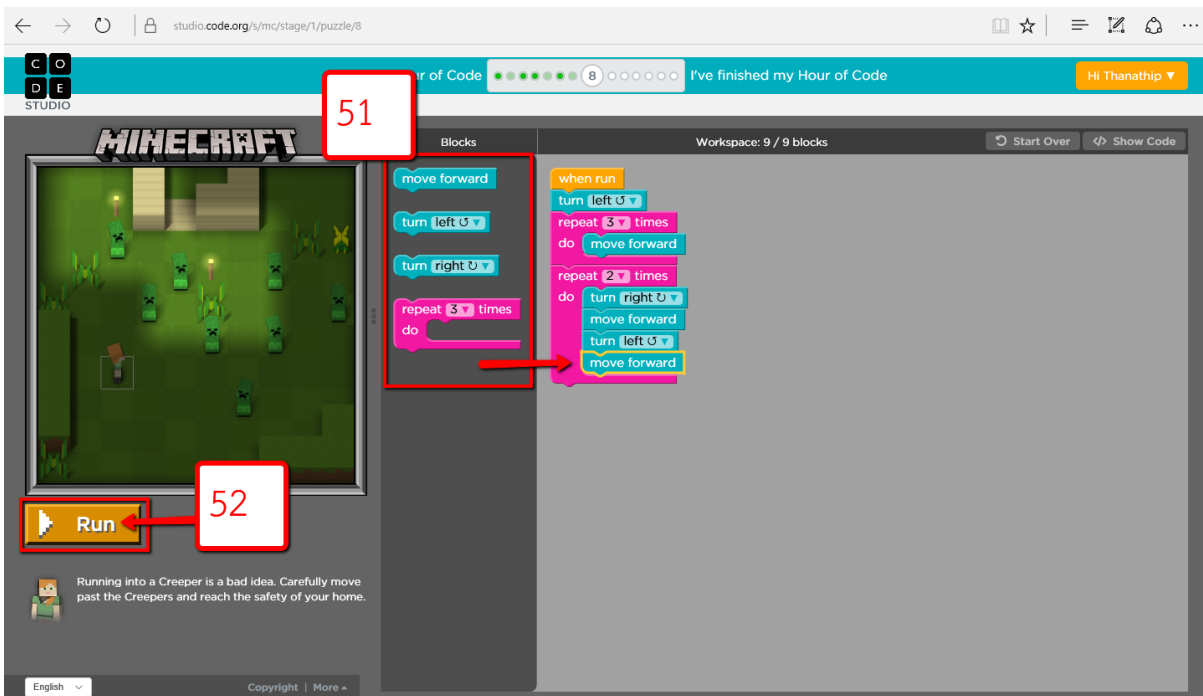
50.อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิก OK

[การวิ่งเข้าไปยัง Creeper ไม่ใช่ความคิดที่ดีแน่ๆ โปรแกรมม์ควรวางในขณะที่เคลื่อนผ่าน Creeper เพื่อไปให้ถึงบ้านอย่างปลอดภัย]

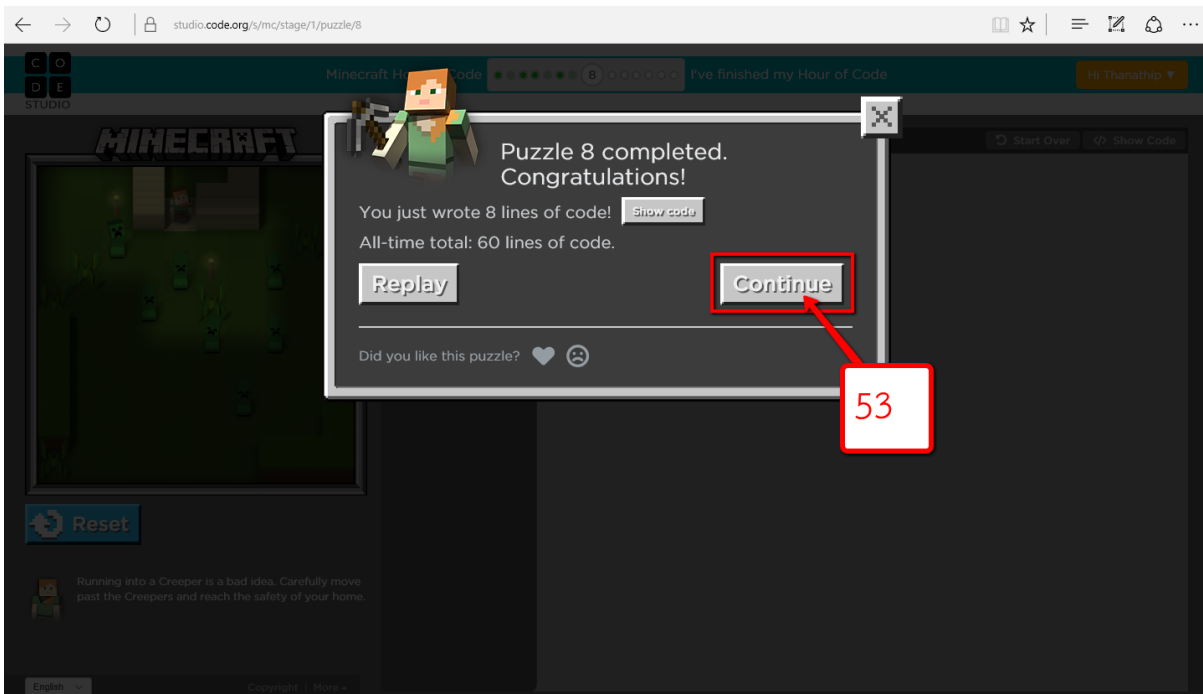


51.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

52.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง

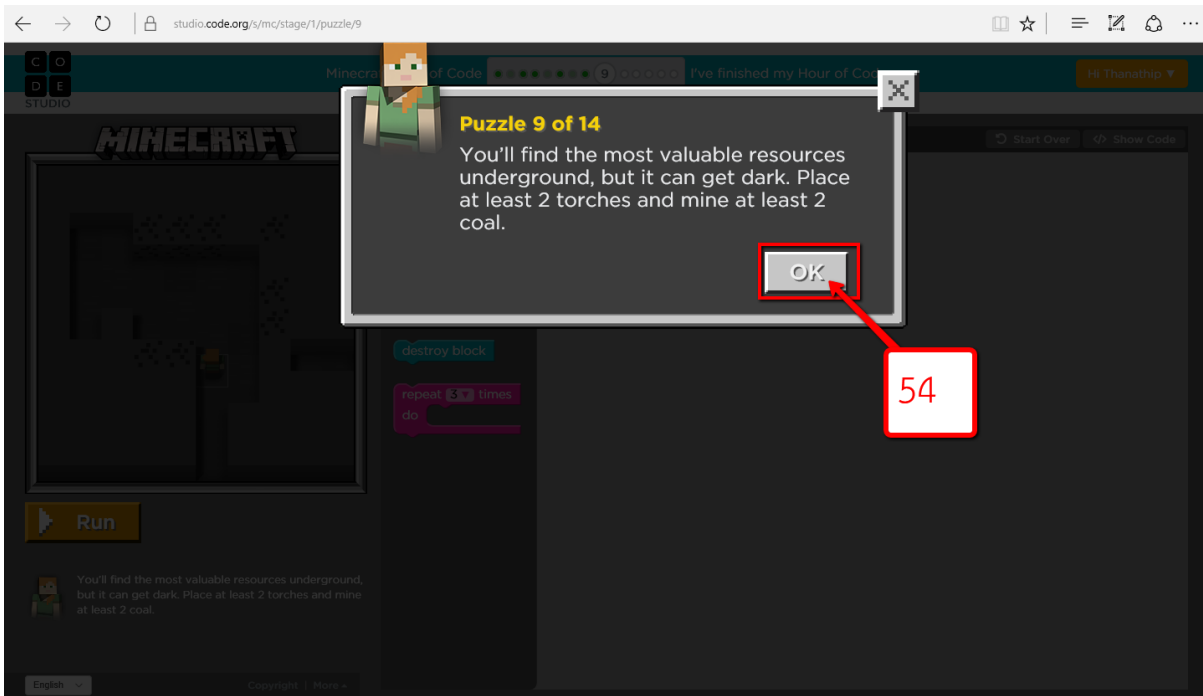


53.คลิกที่ปุ่ม Continue



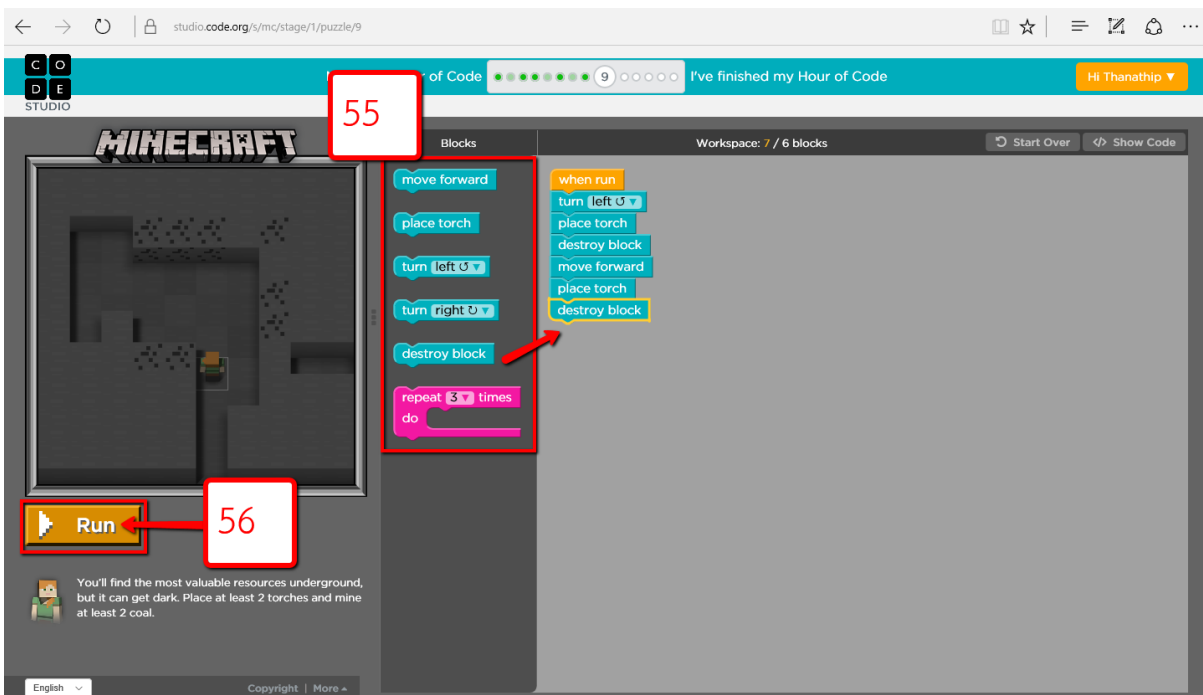
54.อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิก OK

[คุณจะพบทรัพยากรที่มีค่าอยู่ใต้ดิน, คุณต้องจุดคบเพลิงอย่างน้อยสองครั้ง และขุดแร่ถ่านหินอย่างน้อย 2 ก้อน]

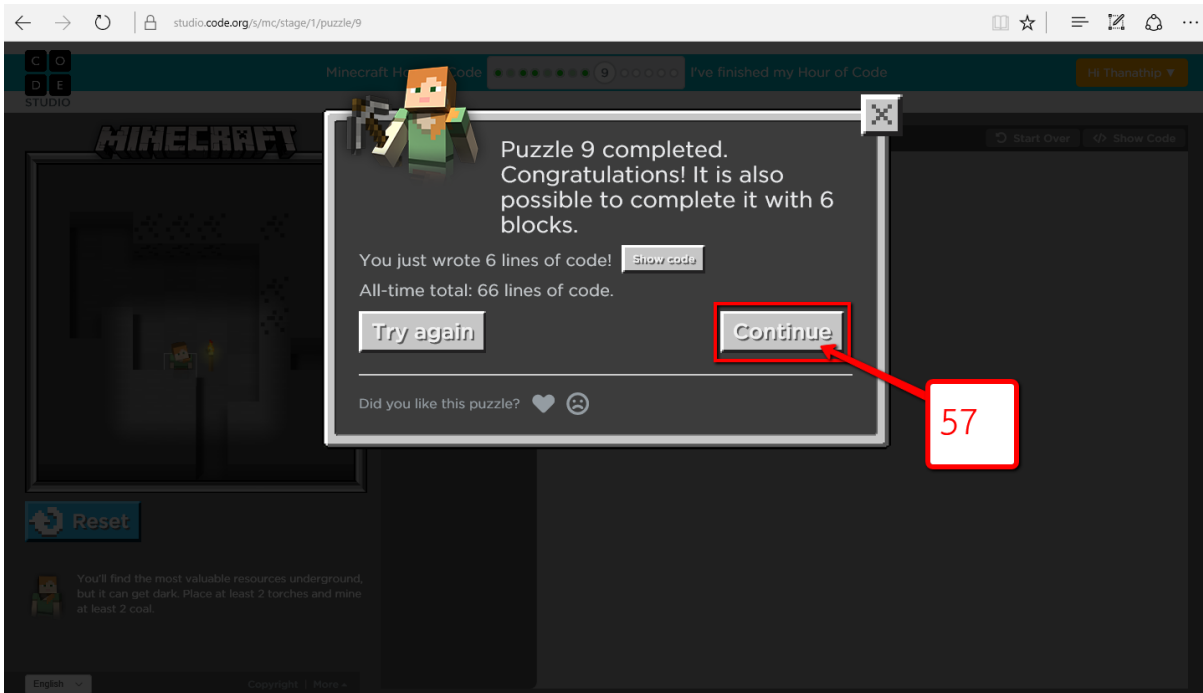


55.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

56.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง

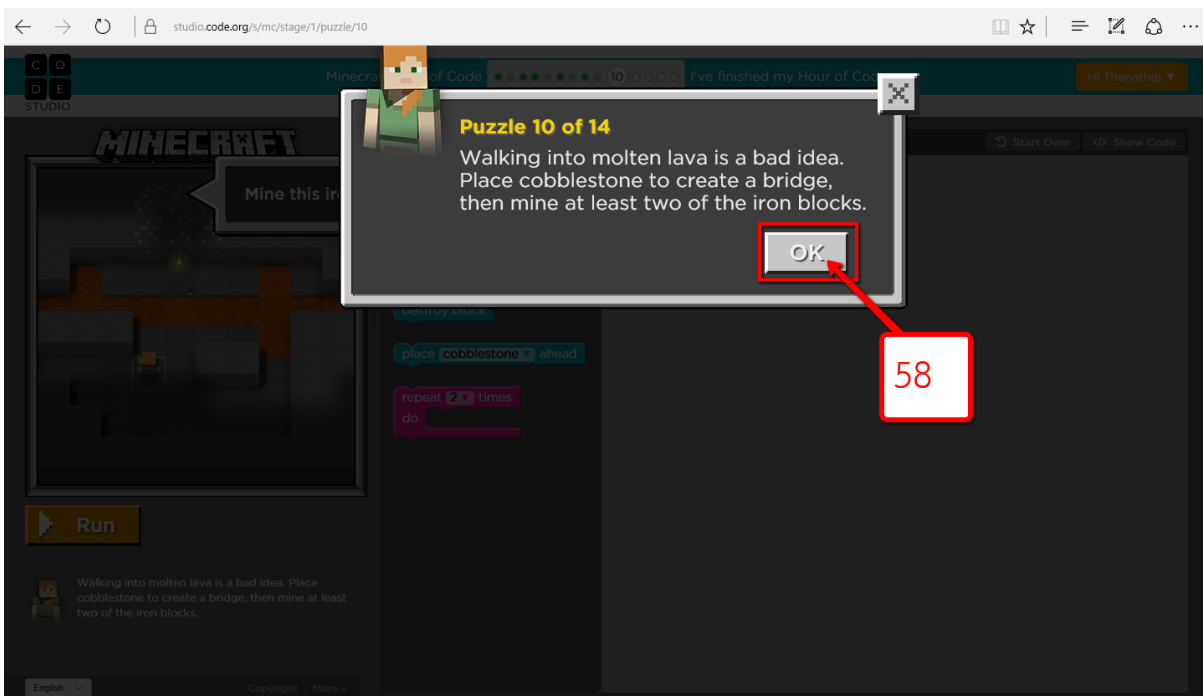


57.คลิกที่ปุ่ม Continue



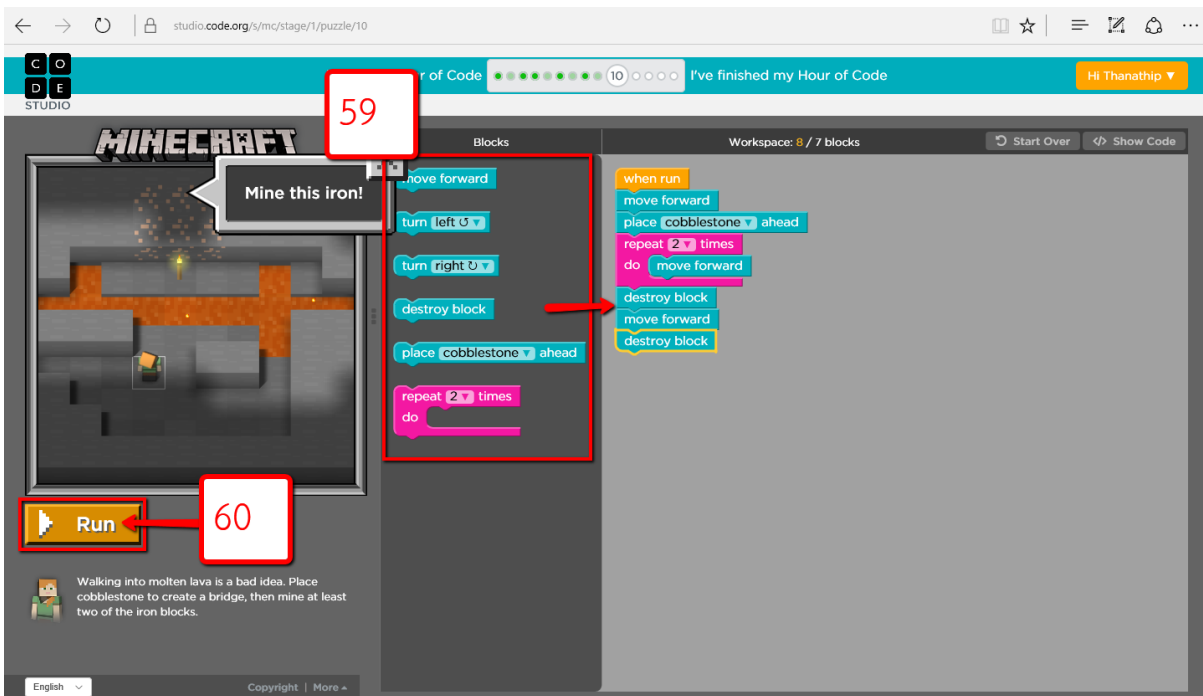
58.อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิก OK

[การเดินลงไปในลาวาเดือดๆเป็นความคิดที่ไม่น่าจะดี, ให้ใช้หินกรวด (cobblestone) เพื่อสร้างสะพาน แล้วค่อยขุดแร่เหล็กอย่างน้อย 2 ก้อน]

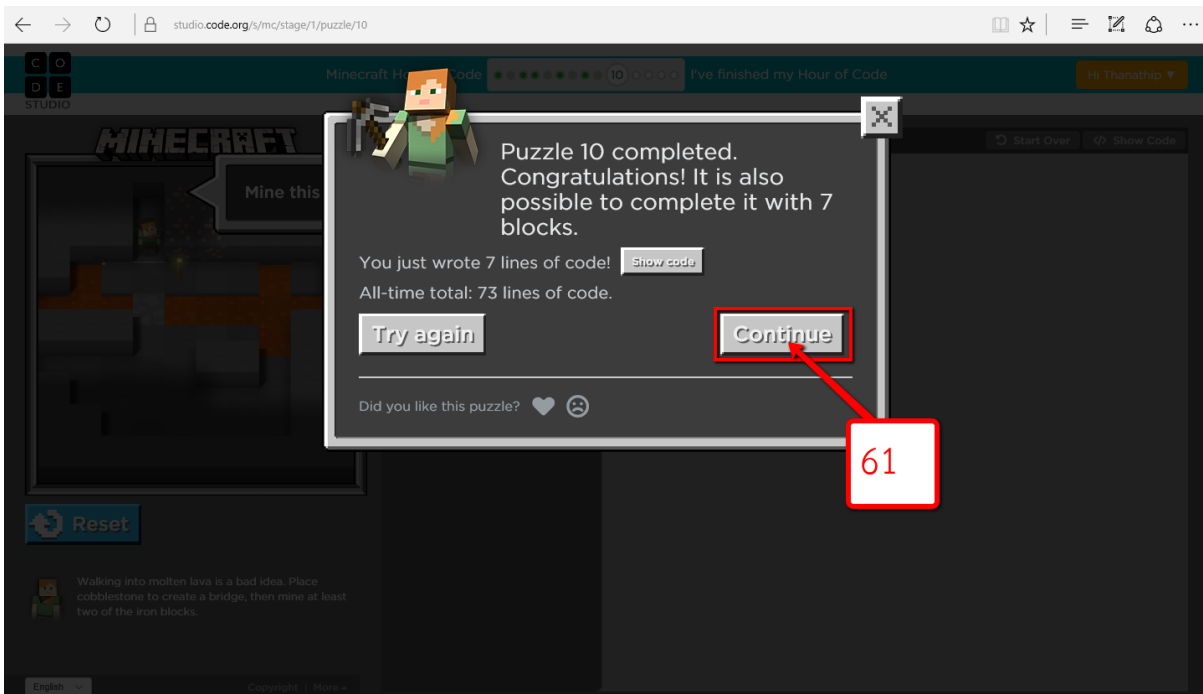


59.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

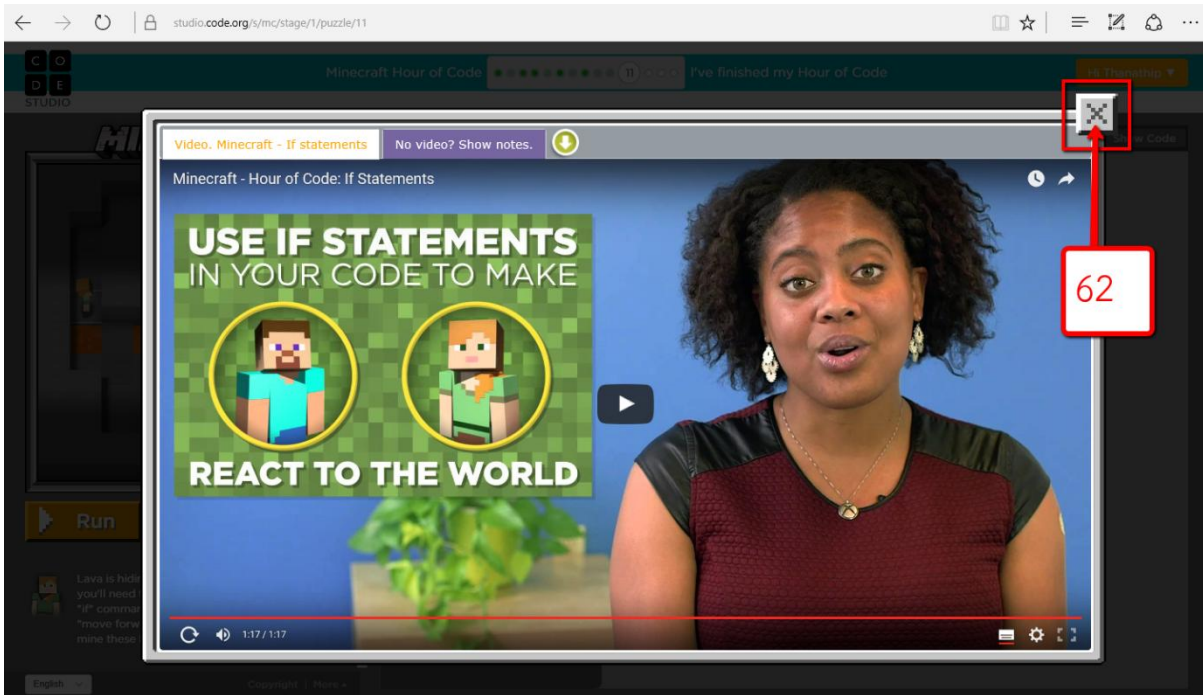
60.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง



61.คลิกที่ปุ่ม Continue



62. จะมีวิดีโอแนะนำสำหรับด่านต่อไปให้คุณจบแล้วคลิกปิดวิดีโอ

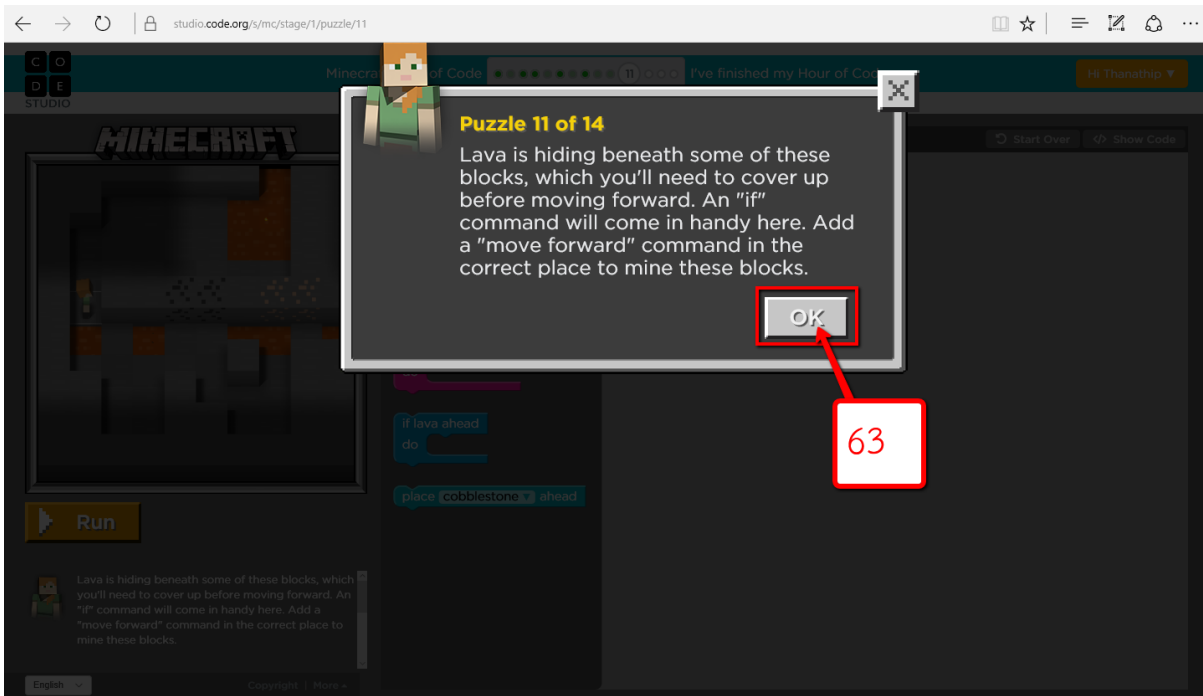


ตอนนี้เรากำลังจะได้เรียนรู้เกี่ยวกับข้อความที่มี `if` ข้อความที่มี `if` เป็นส่วนพื้นฐานของการเรียนรู้ในการเขียนโปรแกรม ข้อความนี้ช่วยคอมพิวเตอร์ในการตัดสินใจ คอมพิวเตอร์ทุกเครื่อง ใช้ข้อความที่มี `if` รวมถึงโทรศัพท์ของเรา เช่น เมื่อเราต้องการปลดล็อกโทรศัพท์ โทรศัพท์จะเรียกใช้โค้ดบางอย่าง ที่บอกว่าเราใส่รหัสผ่านถูกต้องแล้วหรือไม่ จากนั้นจึงปลดล็อกโทรศัพท์ มิฉะนั้นจะแสดงข้อความแสดงข้อผิดพลาด

คุณสามารถใช้ข้อความที่มี `if` ในโค้ดของคุณที่จะทำให้ Steve และ Alice ตอบสนองต่อสิ่งที่พวกเขาเห็นในโลกตัวอย่าง เช่น ถ้ามีหินอยู่ด้านหน้าพวกเขาก็สามารถเลี้ยวซ้ายหรือเลี้ยวขวาถ้าพวกเขาวิ่งไปเจอดินไม้ หรือในกรณีที่ เราไม่ต้องการที่จะที่จะตกลงไปในลาวา เป็นเรื่องง่ายที่จะวางแผนรับมือกับลาวา เราสามารถมองเห็นลาวาได้บนจอ แต่ลาวาที่อยู่ใต้หิน ซึ่งเรามองไม่เห็นล่ะ หลังจากที่เราละเปิดหิน เราจะต้องตรวจสอบว่า มีลาวาในบริเวณนั้นหรือไม่ ก่อนที่จะเคลื่อนไหวไปข้างหน้า หากมีลาวาที่นั่น เราต้องการที่จะวางหินที่ด้านหน้าตัวละครของเรา ก่อนที่จะเคลื่อนไหวไปข้างหน้า ด้วยวิธีนี้เราสามารถเคลื่อนไหวไปข้างหน้าได้อย่างปลอดภัย ถึงเวลาสำหรับการระเบิดแร่อื่นๆ แล้ว! และอย่าลืมที่จะใช้ข้อความที่มี `if` เพื่อคุณขั้นตอนของคุณ

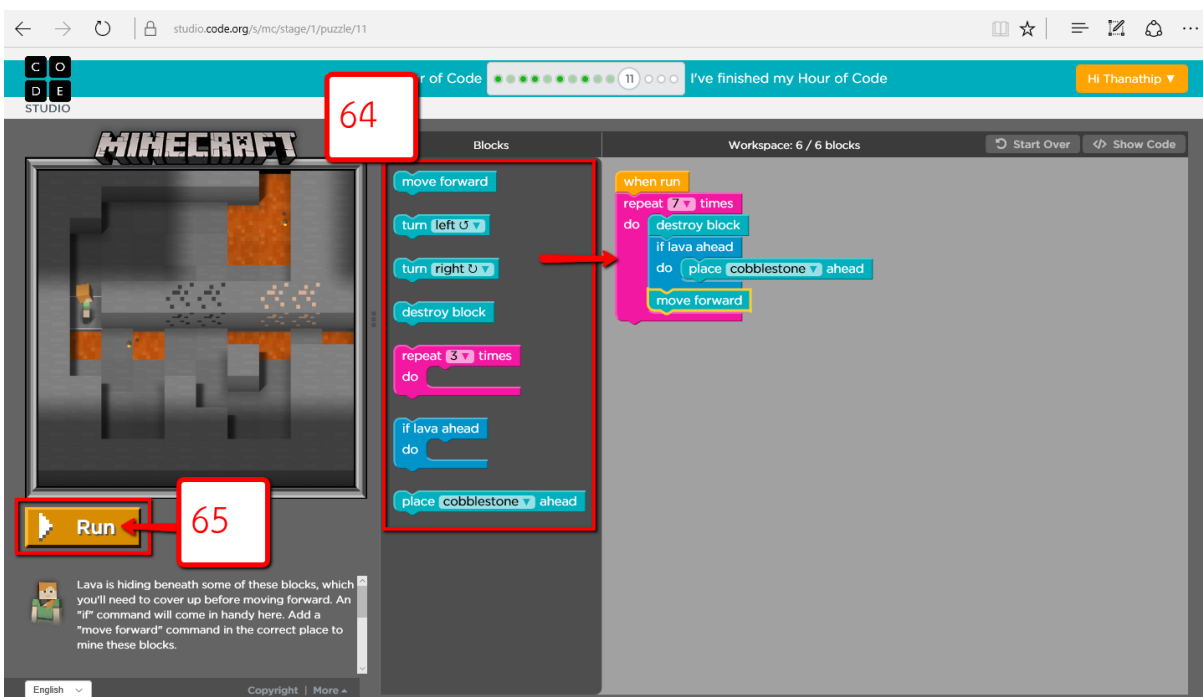
63. อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิก OK

[มีลาวาซ่อนอยู่ใต้บางบล็อก ซึ่งคุณต้องถมมันก่อนจะเดินไปข้างหน้า โดยคำสั่ง “ถ้า (if)” จะช่วยจัดการตรงจุดนี้ให้ และเพิ่มคำสั่ง “เดินไปข้างหน้า (move forward)” ในตำแหน่งที่ถูกต้องเพื่อขุดบล็อกเหล่านี้]

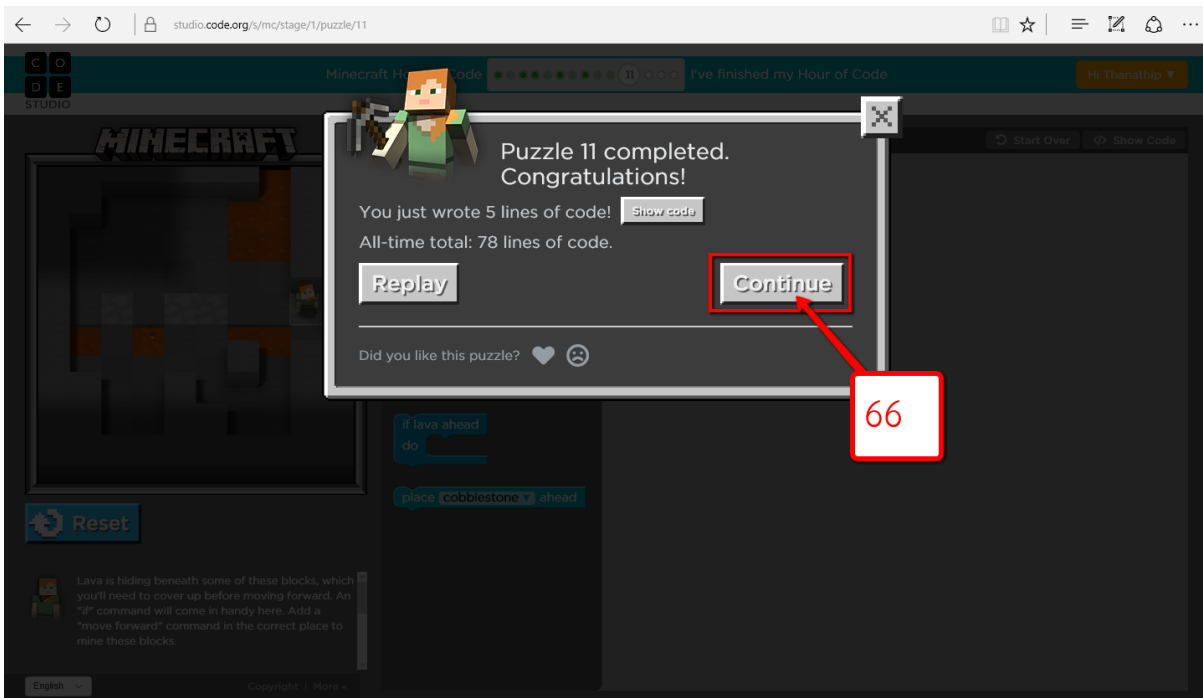


64.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

65.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง

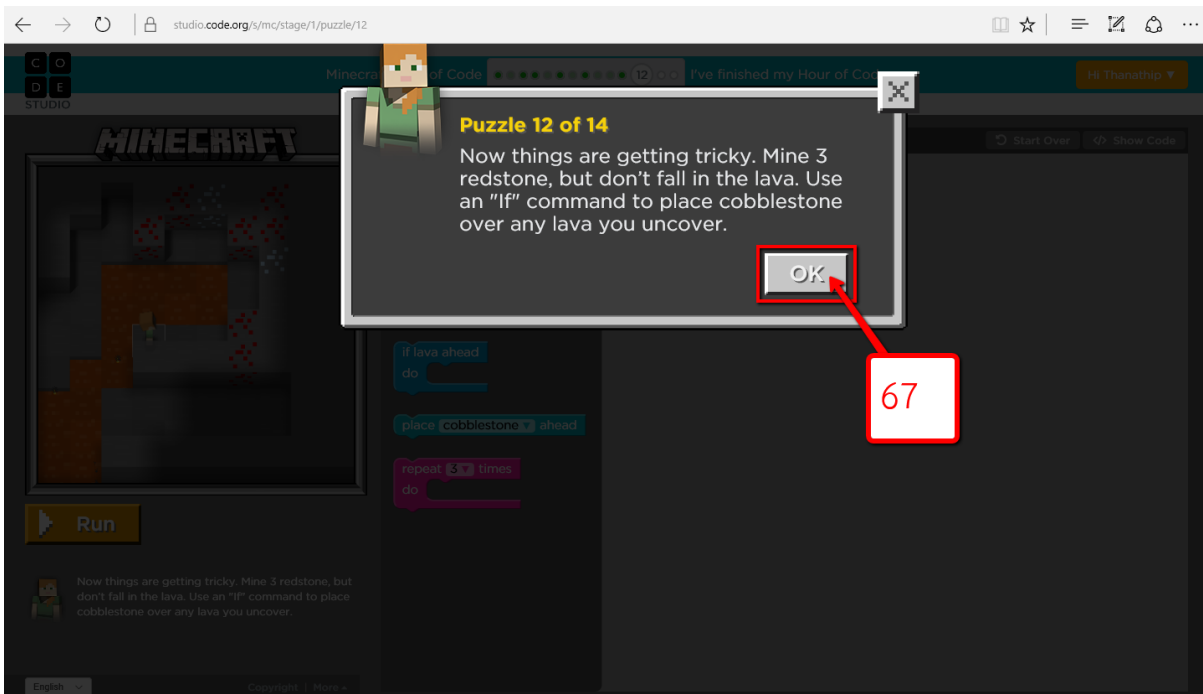


66.คลิกที่ปุ่ม Continue



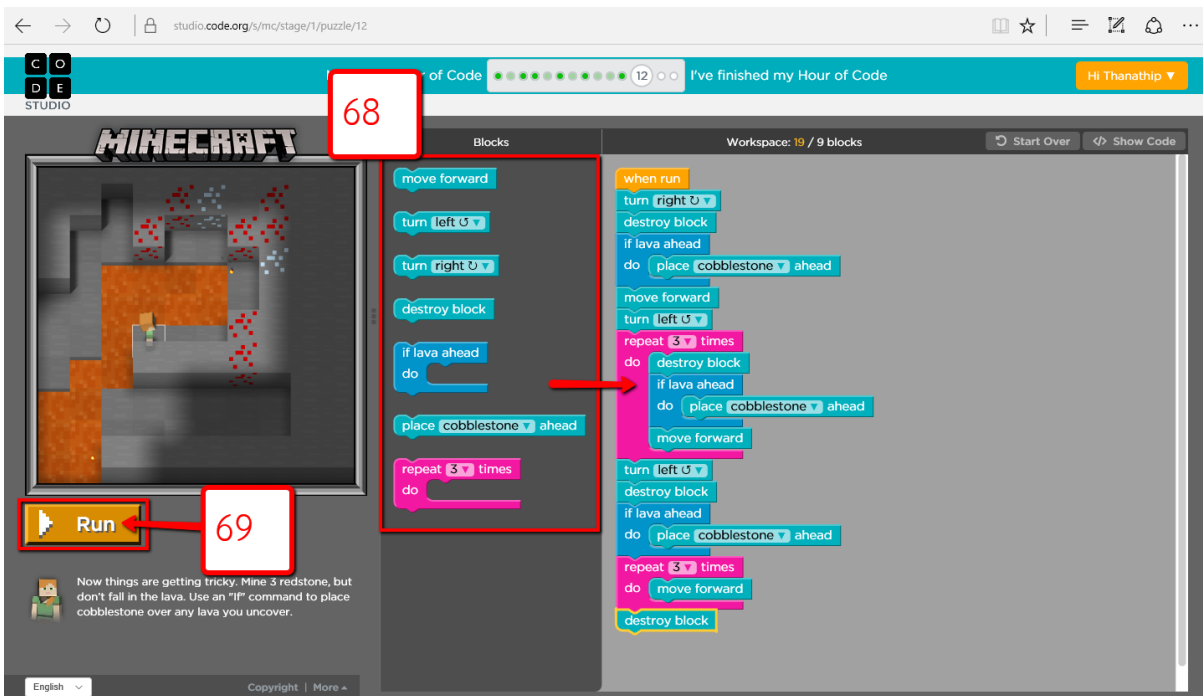
67.อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิก OK

[ตอนนี้ทุกอย่างดูจะยากขึ้น ให้ขุดพลอยสีแดง (redstone)3 ก้อน แต่ห้ามตกลงไปในลาวา ใช้คำสั่ง “ถ้า (if)” เพื่อถมหินกรวด (cobblestone) ลงในลาวาที่คุณยังไม่ได้ถม]

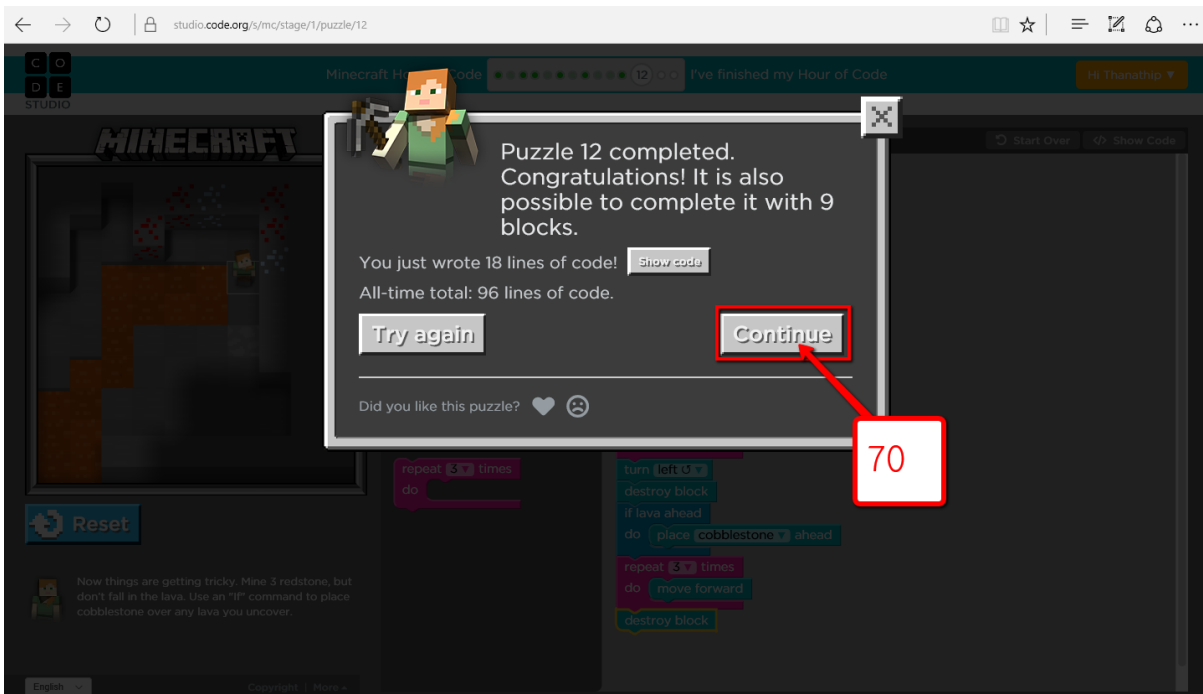


68.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

69.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง

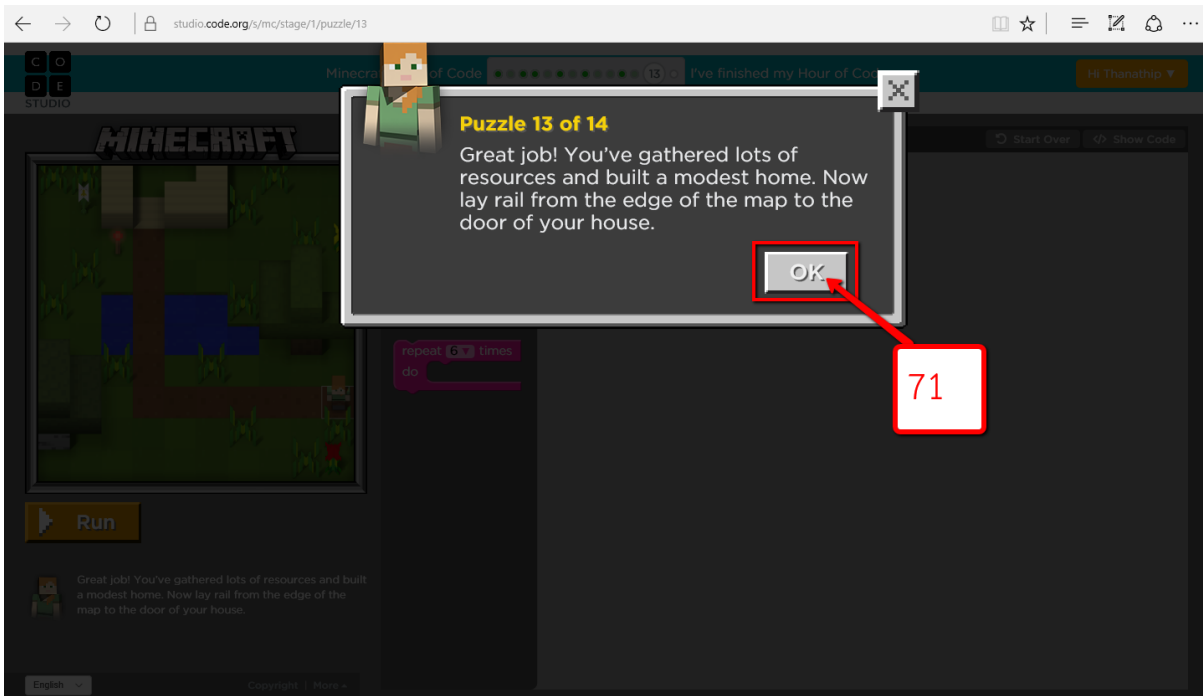


70.คลิกที่ปุ่ม Continue



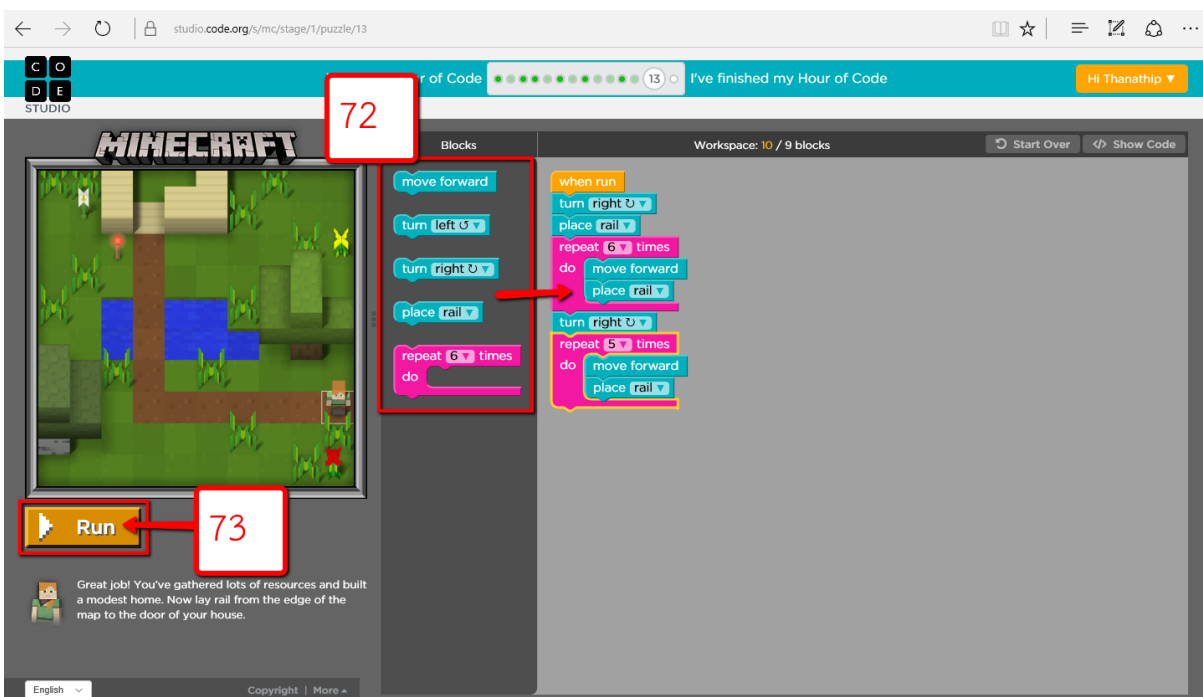
71.อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิก OK

[ทำดีมาก! คุณได้ทำการรวบรวมทรัพยากรจำนวนมาก และสร้างบ้านขนาดย่อมแล้ว ตอนนีให้คุณวางรางรถไฟจากขอบแผนที่ไปยังประตูหน้าบ้านของคุณ]

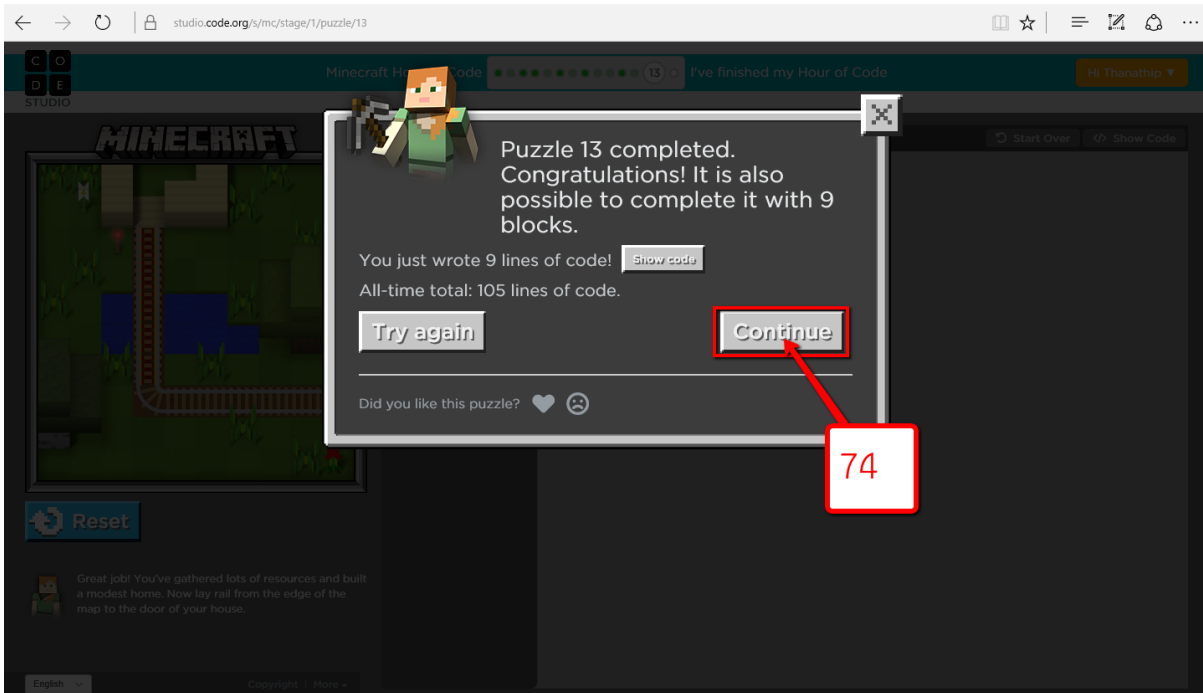


72.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

73.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง



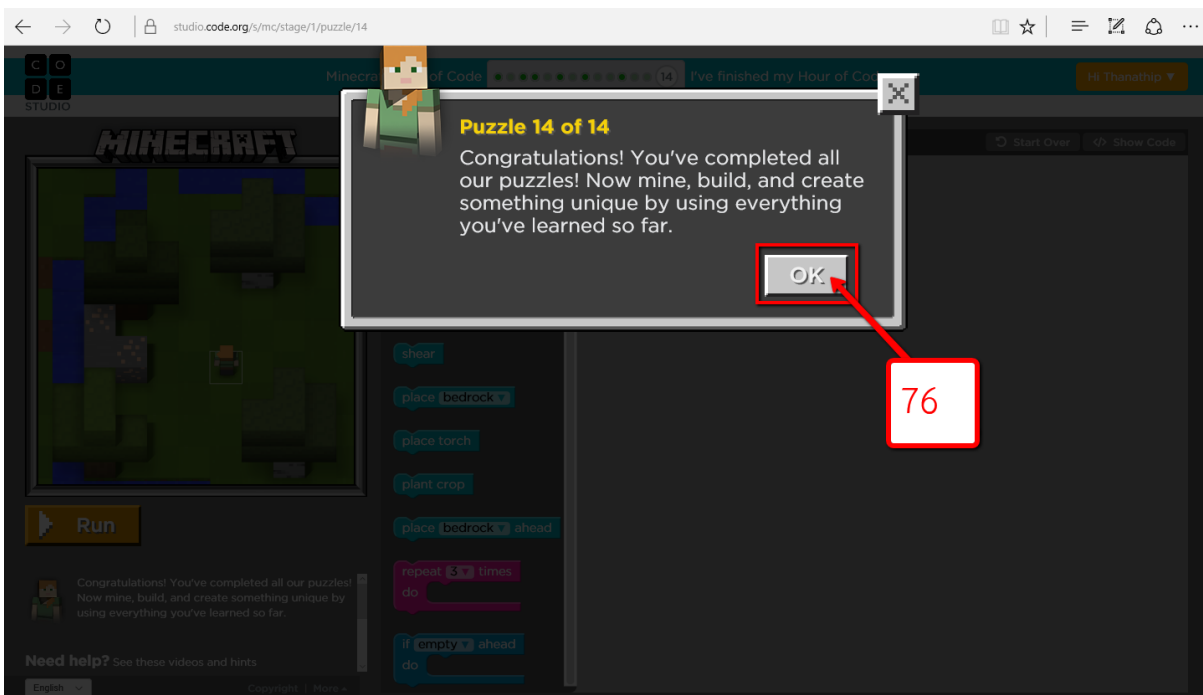
74.คลิกที่ปุ่ม Continue



75.คุณเข้าสู่ด่านสุดท้าย คุณวิดีโอจบแล้วคลิกปิด

76.อ่านคำสั่งสำหรับด่านต่อไปแล้วคลิก OK

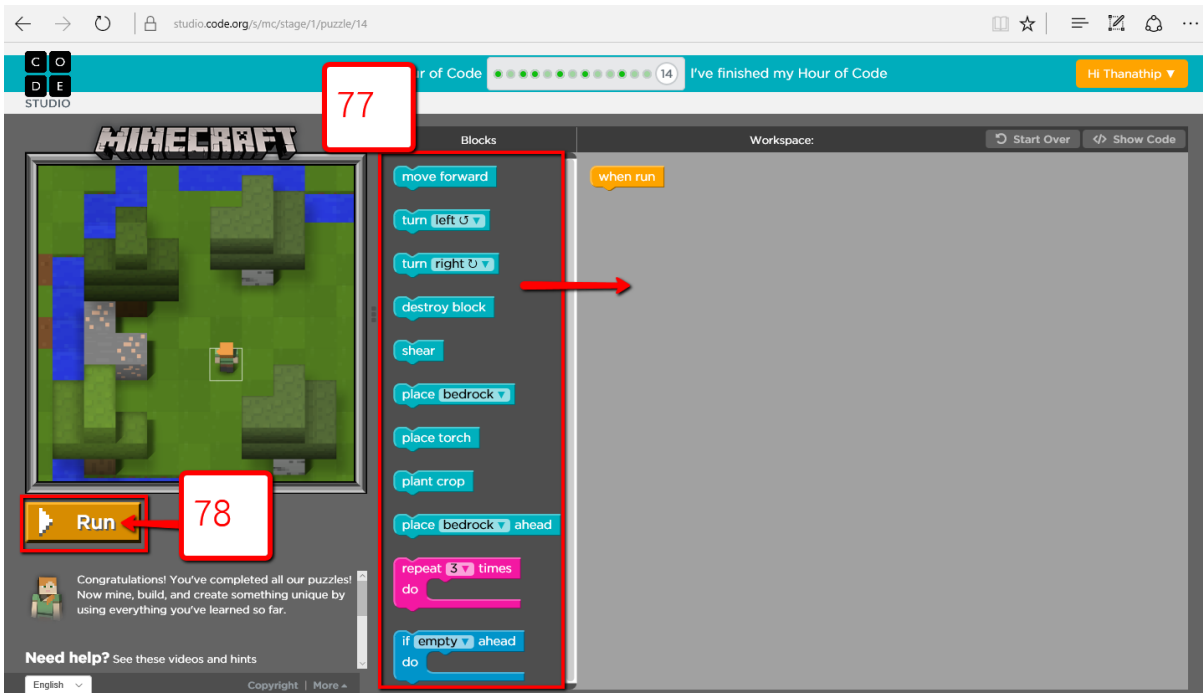
[ขอแสดงความยินดีด้วย! คุณได้ผ่านด่านทั้งหมดของเกมแล้ว และตอนนี้ให้ชุด สร้าง และ ประดิษฐ์บางสิ่งที่เป็นเอกลักษณ์เฉพาะตัวคุณ โดยใช้ทุกสิ่งที่คุณได้เรียนรู้มาทั้งหมด]



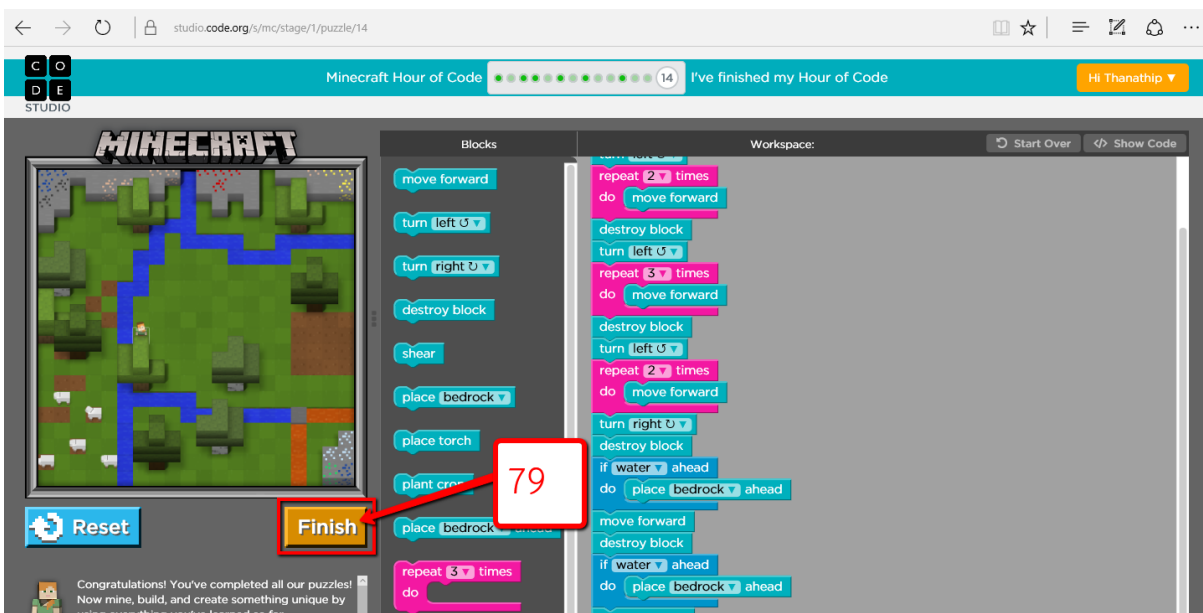
ขอแสดงความยินดี! คุณเพิ่งเรียนรู้พื้นฐานในการสร้างบล็อกในวิทยาการคอมพิวเตอร์ ตอนนี้คุณสามารถใช้บล็อกเหล่านี้เพื่อการสร้างสรรค์ตามแบบของคุณ ไม่มีคำแนะนำเพิ่มเติมและไม่มีปริศนาที่จะแก้แล้ว คุณสามารถสร้างอะไรก็ได้ตามที่คุณต้องการ คุณเลือกได้ ขอให้สนุกกับโปรแกรม

77.คลิกลากคำสั่งลงไปจัดเรียงตามต้องการ

78.คลิกที่ปุ่ม Run เพื่อดูผลของคำสั่ง



79.คลิกที่ปุ่ม Finish

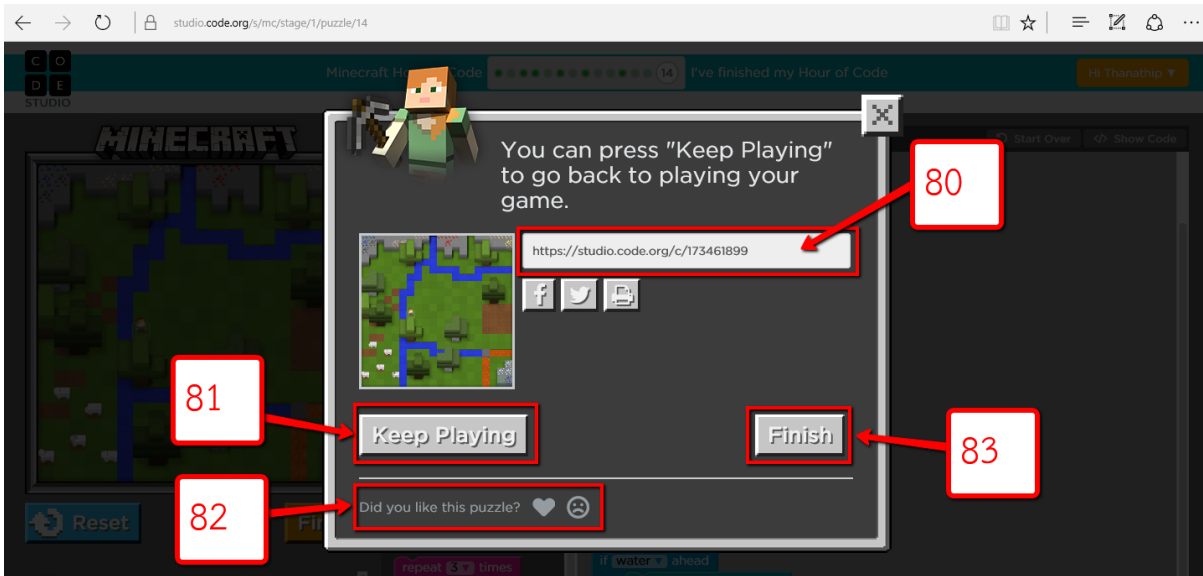


80. URL สำหรับกลับเข้ามาเล่นเกมของคุณอีกครั้ง

81.ปุ่ม Keep Playing เพื่อย้อนกลับไปเล่นเกมต่อ

82.คลิกเพื่อบอกว่าคุณชอบหรือไม่ชอบเกมนี้

83.คลิกที่ปุ่ม Finish

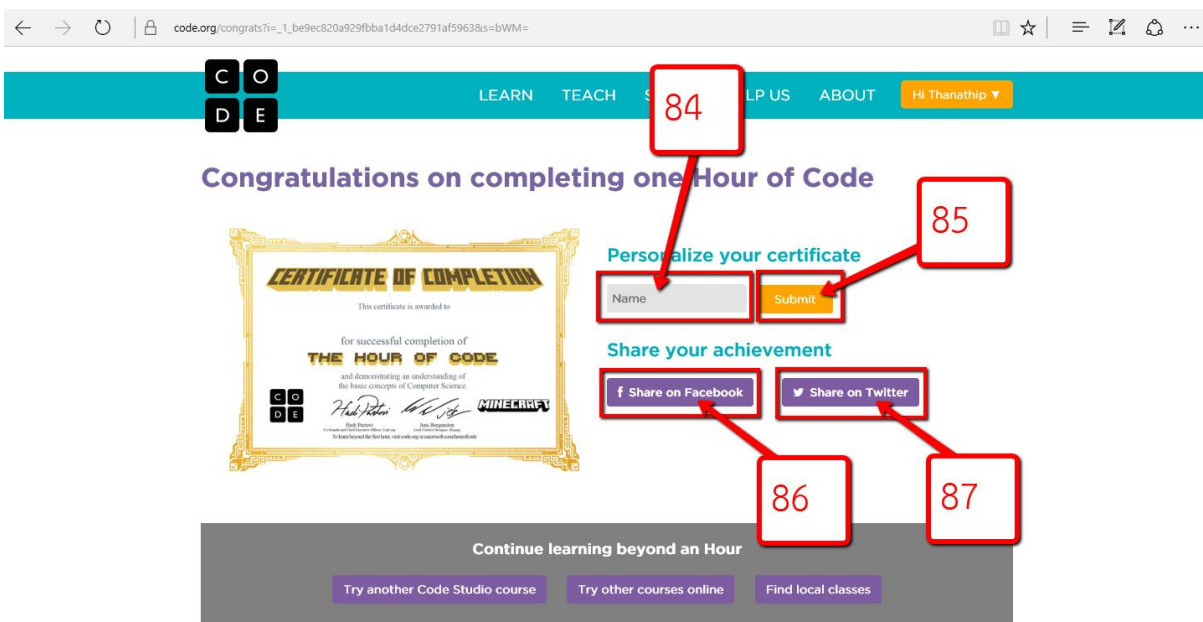


84.กรอกชื่อ และนามสกุลของคุณ

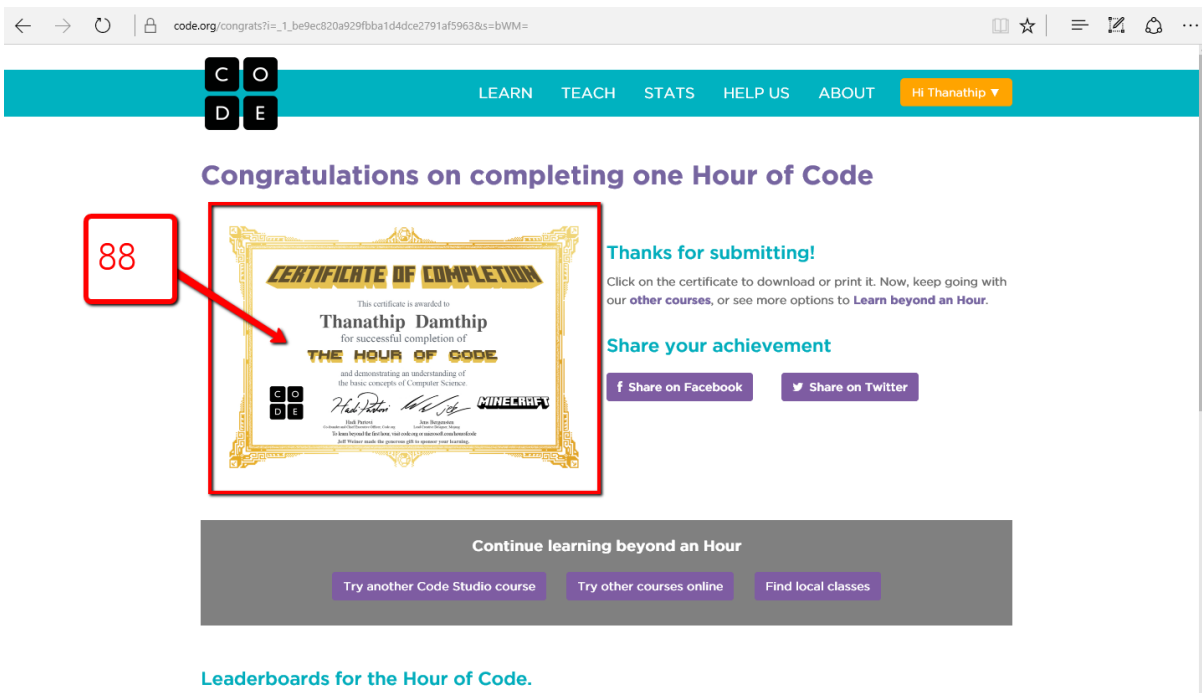
85.คลิกที่ปุ่ม Submit

86.ปุ่มแบ่งปันในประกาศไปยัง Facebook ของคุณ

87.ปุ่มแบ่งปันในประกาศไปยัง Twitter ของคุณ



88. หลังจากคลิก Submit ชื่อของคุณจะมาปรากฏบนใบประกาศ ให้คลิกบนใบประกาศ เพื่อบันทึกภาพ

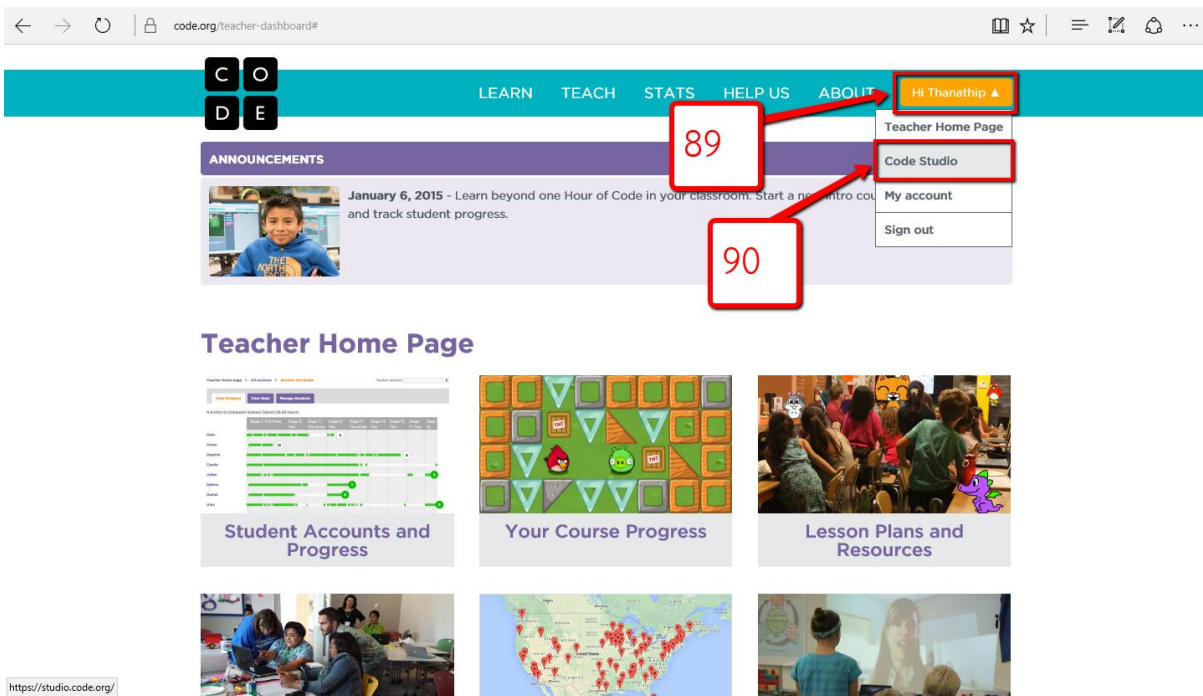


Leaderboards for the Hour of Code.

เริ่มต้นสร้างเกมของคุณเอง

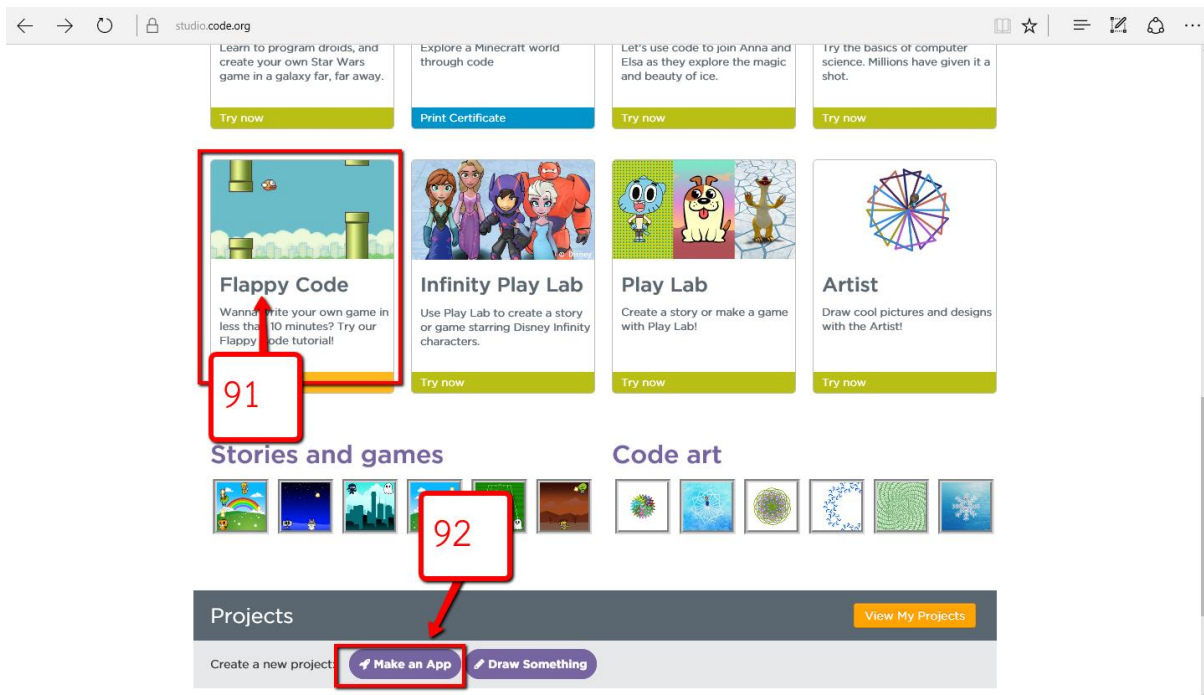
89.คลิกที่ปุ่มชื่อของคุณ

90.คลิกที่เมนู Code Studio

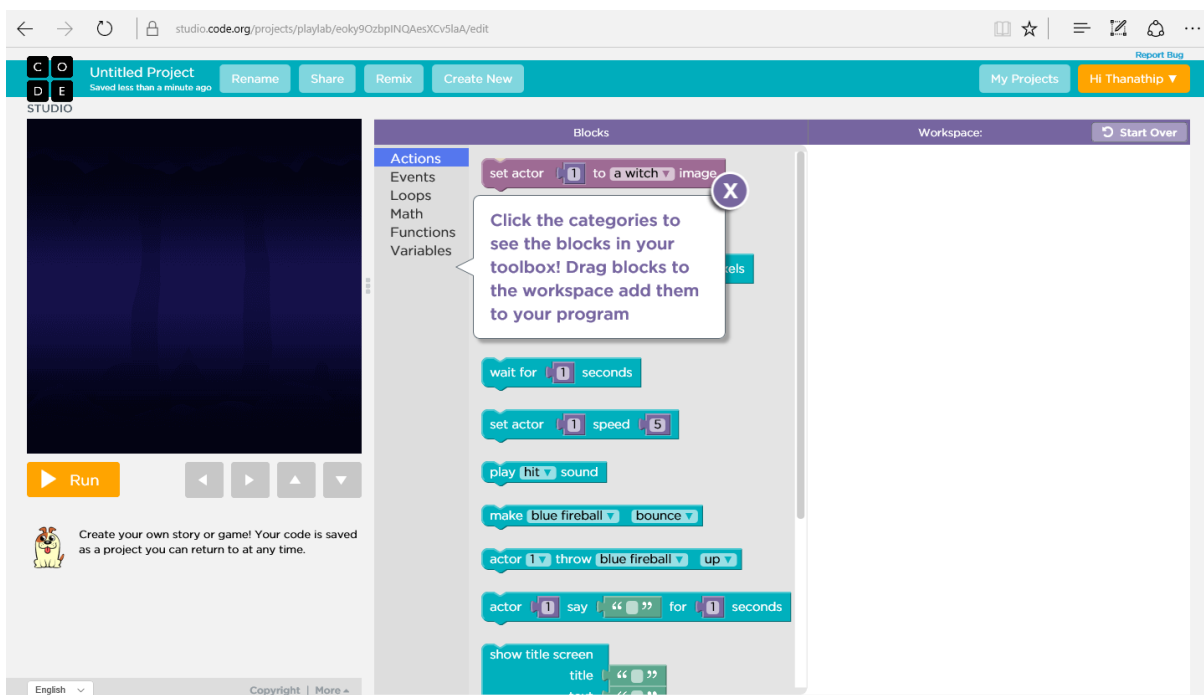


91.คุณสามารถศึกษาการเขียนโปรแกรมเพิ่มเติมจากเกม Flappy Code ได้เพื่อความเข้าใจที่มากขึ้น

92.คลิกที่ปุ่ม Make an App เพื่อเริ่มสร้างเกมของคุณเอง



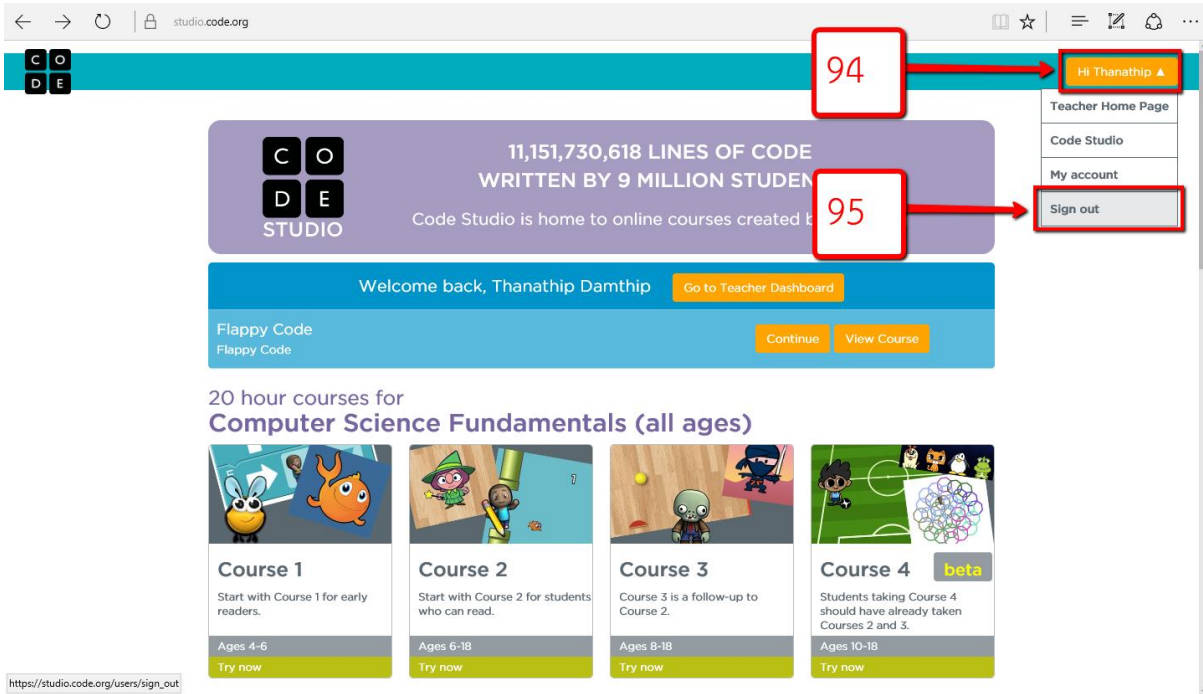
จะปรากฏหน้าต่าง Studio สำหรับสร้างเกมขึ้นมา คุณสามารถสร้างเกมโดยใช้ปุ่มคำสั่งต่าง ๆ ได้ตามจินตนาการ



การออกจากระบบ

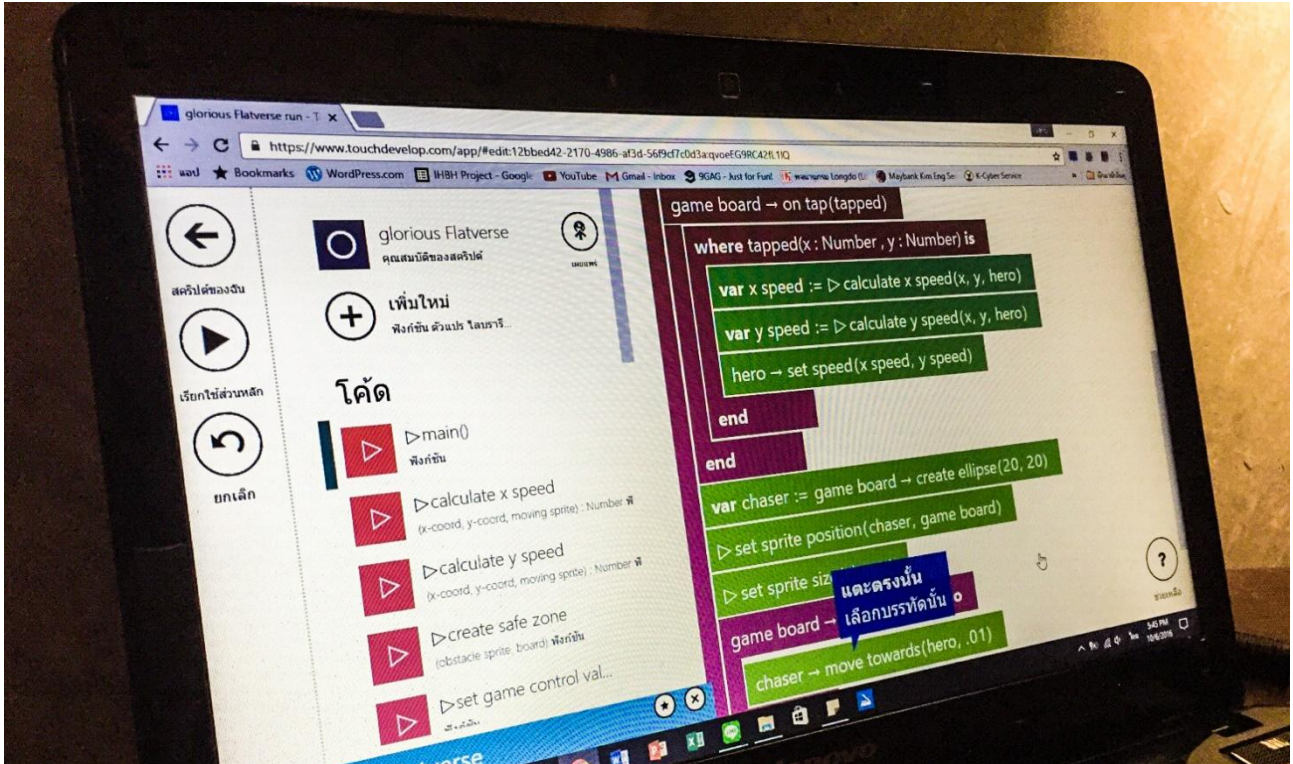
94.คลิกที่ชื่อของคุณ

95.คลิกที่เมนู Sign out



กิจกรรม 4

ทดลองเขียนโปรแกรมด้วยตนเอง



ระยะเวลา

50 - 60 นาที

อุปกรณ์

คอมพิวเตอร์ หรือแท็บเล็ต 1 เครื่องต่อผู้เรียน 1-2 คน

อินเทอร์เน็ต

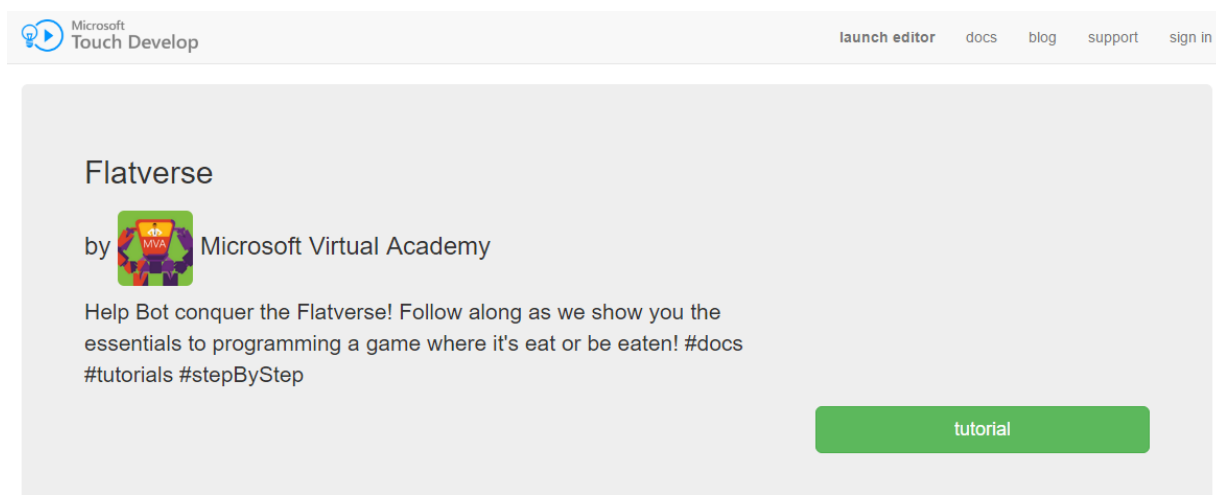
โปรเจ็คเตอร์ (สำหรับผู้สอน)

จุดประสงค์

เพื่อให้ผู้เรียนเข้าใจตรรกะพื้นฐานของการเขียนโปรแกรมเบื้องต้น ผ่านหลักสูตรเกมส์ Flatverse โดยใช้ Microsoft Touch Develop

ขั้นตอนการจัดกิจกรรม

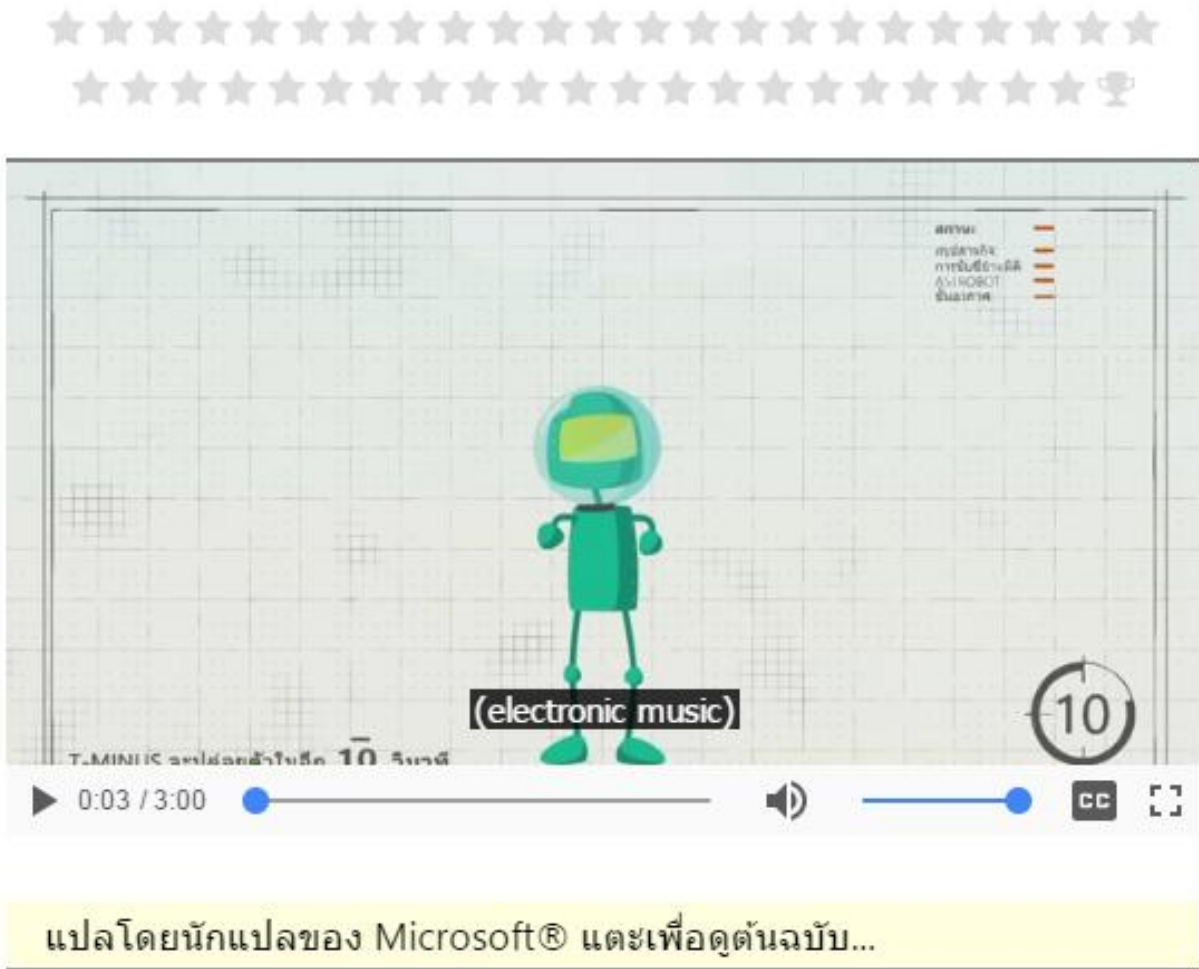
1. ผู้สอนและผู้เรียนเปิดบราวเซอร์ไปที่ <https://www.touchdevelop.com/mbahc>
2. จะพบหน้าจอเริ่มต้น Tutorial สำหรับสร้างเกมส์ Flatverse ให้กดที่ปุ่ม Tutorial เพื่อเริ่มสร้างเกมส์



The screenshot shows the Microsoft Touch Develop website. At the top left is the logo "Microsoft Touch Develop". On the right side of the header, there are links for "launch editor", "docs", "blog", "support", and "sign in". The main content area features a card for "Flatverse" by "Microsoft Virtual Academy". The card includes a small icon of a robot, the text "Help Bot conquer the Flatverse! Follow along as we show you the essentials to programming a game where it's eat or be eaten! #docs #tutorials #stepByStep", and a prominent green button labeled "tutorial".

© 2016 Microsoft [privacy and cookies](#) [legal](#)

3. หน้าจอจะแสดงวิดีโอเพื่อกระตุ้นการสร้างสรรค์แถบดาวด้านบนจะแสดงให้เห็นถึงความคืบหน้าในการเรียนเขียนโปรแกรม กด “มาเริ่มกันเถอะ” เพื่อเข้าสู่บทเรียน



มาเริ่มกันเถอะ!

4. เข้าสู่บทเรียนขั้นที่ 1 หน้าจะแสดงวิดีโอสอนการสร้างกระดานเกมส์ ผู้สอนอธิบายจุดประสงค์ของขั้นแรกตามวิดีโอ หรือจะให้ผู้เรียนดูวิดีโอจนจบแล้วจึงเริ่มอธิบายจุดประสงค์ของขั้นแรก คือแสดงการสร้างกระดานเกมส์ หลักการเขียนโปรแกรมคือการกำหนดคำสั่งให้กับคอมพิวเตอร์ โดยแต่ละภาษาจะมีหลักไวยากรณ์ของคำสั่งที่ต้องเรียนรู้ ในที่นี้เป้าหมายของขั้นแรกคือการเริ่มต้นสร้างกระดานเกมส์

🔄 game → start คือคำสั่งเริ่มเกมส์

var game board := 🔄 game → start คือการสร้างตัวแปรของกระดานเกมส์เพื่อเก็บค่า และกำหนดค่าของกระดานเกมส์ในภายหลัง

เมื่ออธิบายจุดประสงค์ของขั้นตอนแรกแล้วให้กด “มาเริ่มกันเถอะ” เพื่อเริ่มสร้างเกมส์



The screenshot shows a video player interface. At the top, there are two rows of stars; the first row has one yellow star followed by 17 grey stars, and the second row has 17 grey stars followed by a trophy icon. The video content displays the code `var game board := 🔄 game → start` on a dark blue background. Below the code, there is a Thai translation: **ตัวแปร** (ตัวแปรจะจัดเก็บข้อมูล (เช่น กระดานเกม) เพื่อเก็บไว้ใช้ในภายหลัง) and the instruction **to memorize that code.** The video player controls at the bottom show a play button, the progress bar at 0:57 / 1:31, a volume icon, and a closed captions icon.

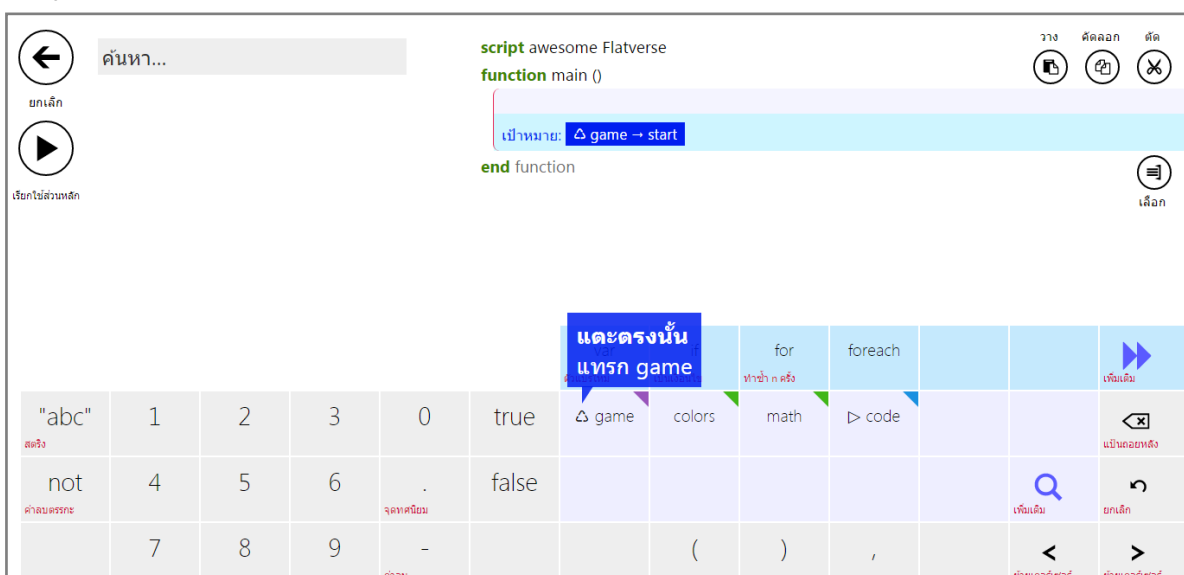
แปลโดยนักแปลของ Microsoft® และเพื่อดูฉบับ...

เริ่มกันเถอะ!

5. หน้าจอจะแสดงส่วนของการเขียนโปรแกรม โดยจะมีกล่องสีน้ำเงินที่จะเป็นตัวช่วยในการเขียนโปรแกรมทุกขั้นตอน แต่ผู้สอนควรชี้แจงให้ผู้เรียนเข้าใจในหลักการของโปรแกรม ควบคู่กับการใช้ตัวช่วยเพื่อให้การสอนมีประสิทธิภาพ ผู้เรียนคลิกตรงบรรทัดสีฟ้าซึ่งเป็นส่วนเขียนโปรแกรมที่อยู่ใน function main



6. หน้าจอจะแสดงเป็นคำสั่งขึ้นมา ซึ่ง Touch Develop จะช่วยให้ผู้เรียนใช้งานได้ง่าย โดยแสดงผลเฉพาะตัวเลือกที่เหมาะสมเท่านั้น ตัวช่วยของโปรแกรมจะบอกว่าเป้าหมายของโปรแกรมที่ต้องการคืออะไร ในที่นี้คือ เป้าหมาย: Δ game \rightarrow start ผู้เรียนคลิก Δ game เพื่อสร้างคำสั่งตามเป้าหมาย




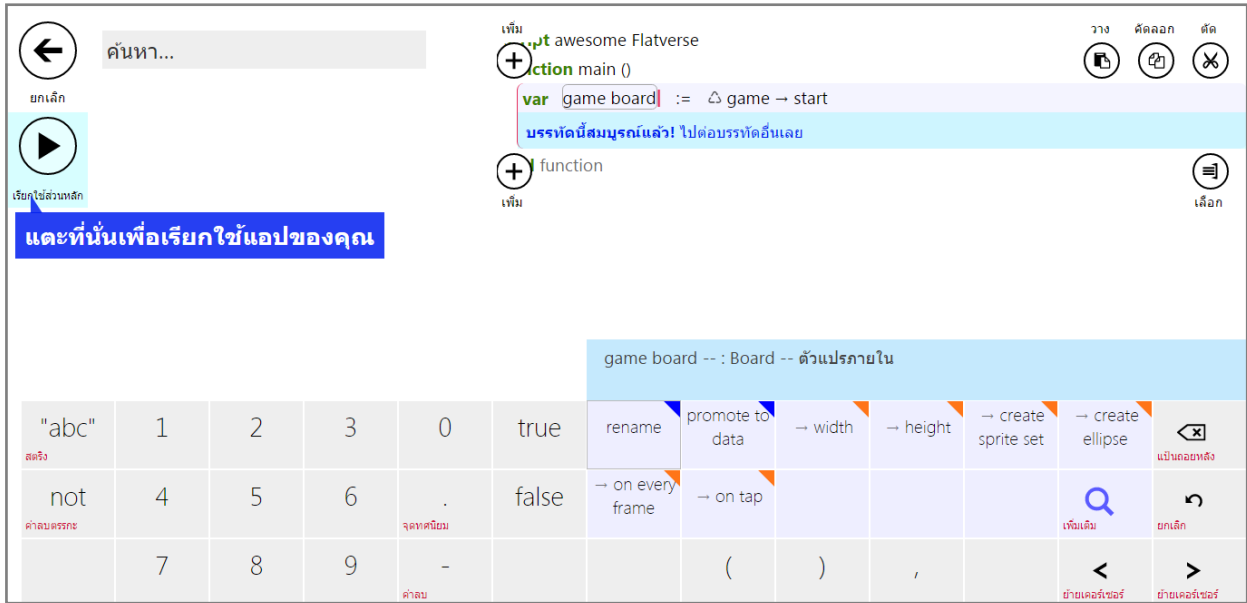
7. หน้าจอเป็นคำสั่งจะแสดงคำสั่งใหม่ที่เปลี่ยนไป ผู้เรียนเลือก `→ start` เพื่อสร้างคำสั่งตามเป้าหมาย


The screenshot shows a programming environment with a code editor and a keyboard overlay. The code editor displays a function definition for 'game' with a comment 'เป้าหมาย: △ game → start'. The keyboard overlay highlights the 'start' key with a blue callout box containing Thai text 'และตรงนั้นแทรก start'.

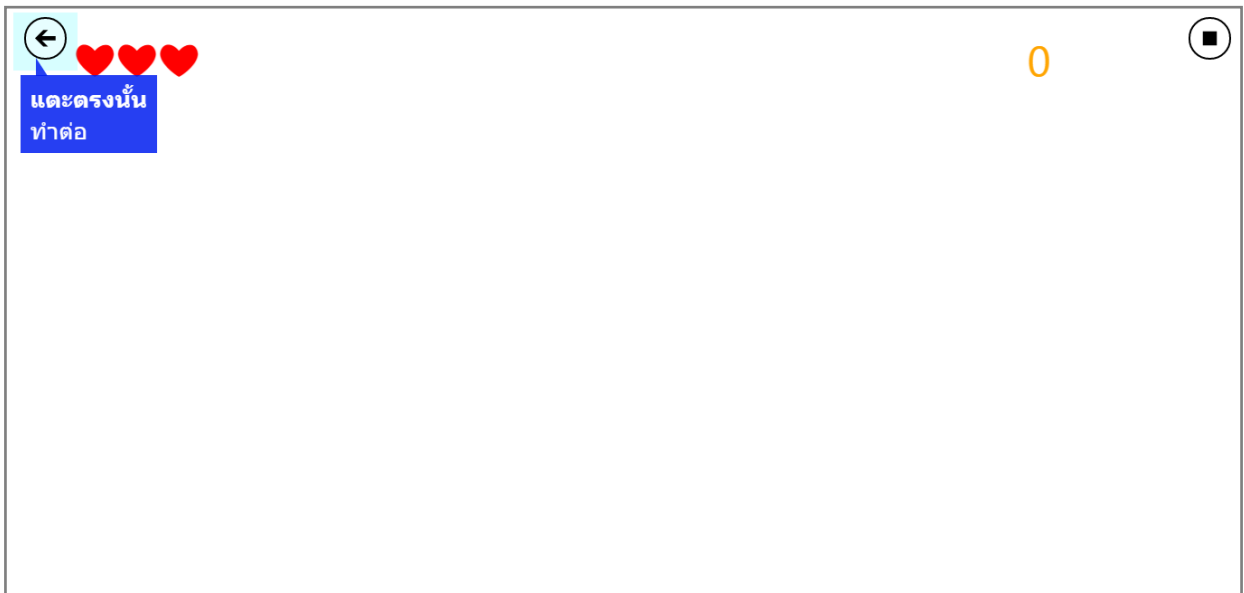
8. ผู้เรียนเลือก `store in var` เพื่อนำค่าการเริ่มต้นโปรแกรมลงไปเก็บในตัวแปร

The screenshot shows a programming environment with a code editor and a keyboard overlay. The code editor displays a function definition for 'game' with a comment 'เป้าหมาย: game board := △ game → start'. The keyboard overlay highlights the 'store in var' key with a blue callout box containing Thai text 'และตรงนั้นเก็บค่าในตัวแปร'.

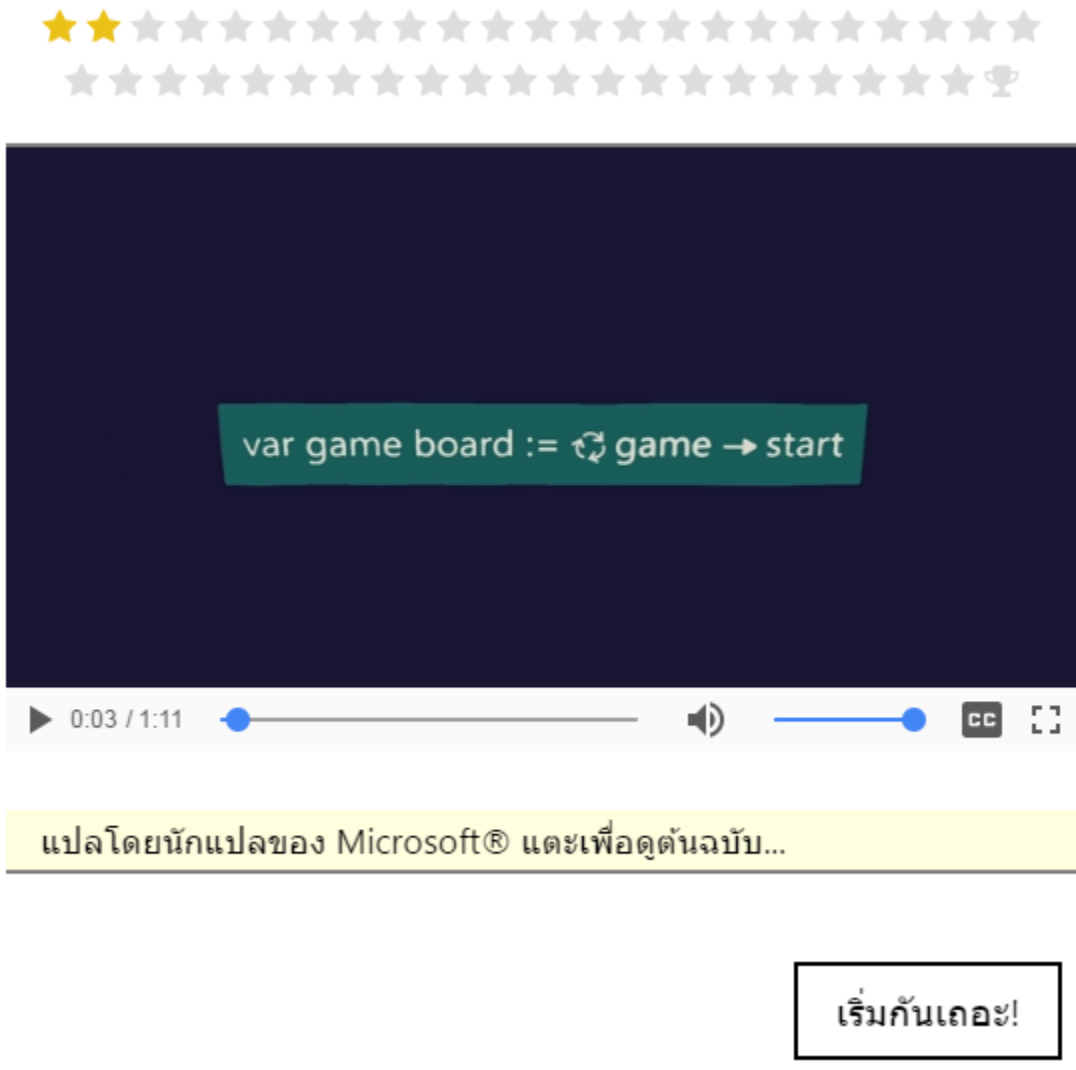
9. โปรแกรมจะแสดงผลออกมาว่า “บรรทัดนี้สมบูรณ์แล้ว” ให้กดปุ่ม  เพื่อแสดงผลลัพธ์ของโปรแกรมคำสั่งที่สร้างไว้



10. หน้าจอจะแสดงผลลัพธ์ของโปรแกรมที่เพิ่งเขียนไปในขั้นแรก เป็นหน้าจอของเกมส์ มีหัวใจแสดงชีวิตของผู้เล่นเกม และมุมขวาเป็นคะแนน กด  เพื่อทำต่อ

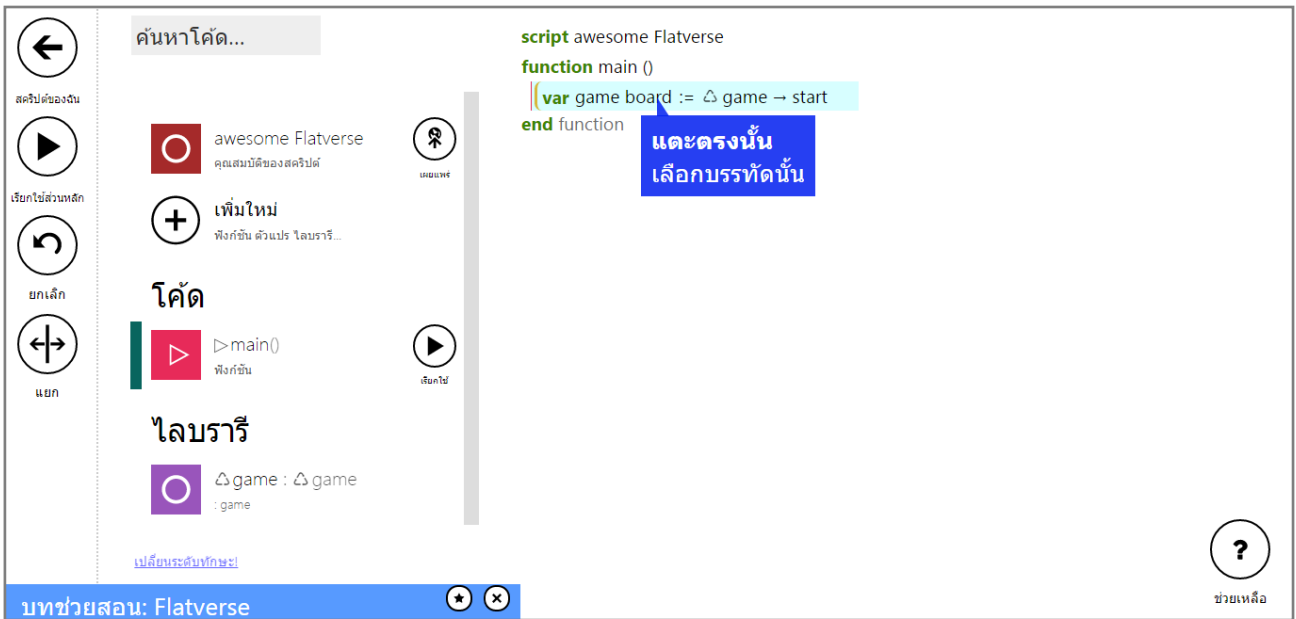


11. หน้าจอจะแสดงวิดีโอของขั้นต่อไป ผู้สอนอธิบายวิธีการสร้างเกมส์ Flatverse ไปทีละขั้นตอน และให้ผู้เรียนลองทำตาม (รายละเอียดขั้นตอนทั้งหมด ดูภาคผนวก)



The screenshot shows a video player interface. At the top, there are two rows of stars: the first row has two yellow stars and 18 grey stars; the second row has 18 grey stars and a trophy icon. The video content area is dark blue with a green box containing the code: `var game board := ↻ game → start`. Below the video is a white control bar with a play button, a progress bar showing 0:03 / 1:11, a volume icon, a volume slider, a closed captions icon, and a full screen icon. Below the control bar is a yellow bar with the text: **แปลโดยนักแปลของ Microsoft® และเพื่อจุดเน้น...**. At the bottom right, there is a button with the text: **เริ่มกันเถอะ!**

12. หน้าจอของการเขียนโปรแกรมจะมีปุ่มหลักๆ อยู่ทั้งหมด 4 ปุ่มด้านซ้ายดังนี้



- ▶ เป็นปุ่มที่แสดงผลพัทธ์ของโปรแกรมที่เขียนไว้
- ↶ เป็นปุ่มเพื่อย้อนกลับไปยังคำสั่งเก่า
- ↔ เปลี่ยนมุมมองของการเขียนโปรแกรมเป็นดังแสดงด้านล่าง คือมีหน้าจอแสดงผลพัทธ์การเขียนโปรแกรมอยู่ด้านข้าง คู่กับหน้าจอเขียนโปรแกรม



- ← เป็นปุ่มที่นำไปสู่หน้าเมนูหลัก หรือการออกจาก tutorial ดังภาพด้านล่าง ซึ่งจะมีปุ่มคำสั่งหลัก ดังนี้



เป็นปุ่มคำสั่งเพื่อจะไปเรียน Tutorial ต่อจากจุดเดิม



เป็นปุ่มคำสั่งเพื่อลองเล่นโปรแกรมที่เขียนไว้

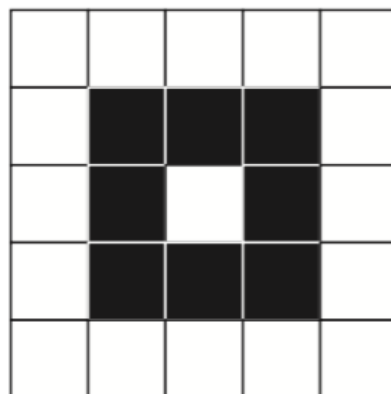
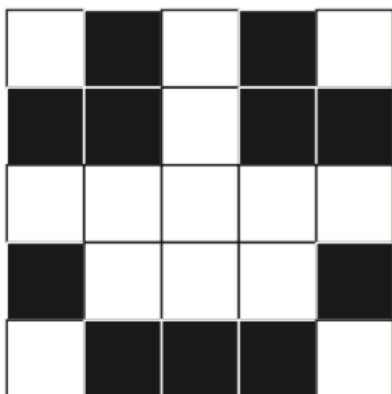
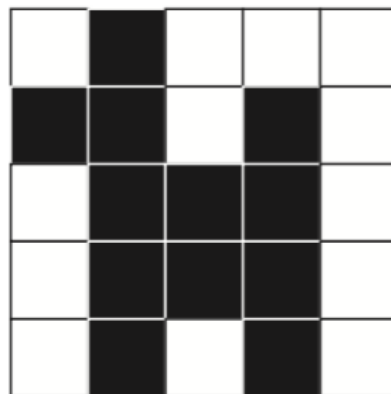
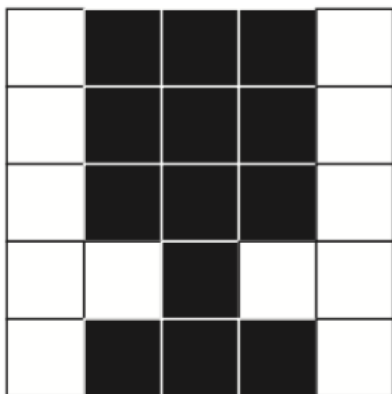
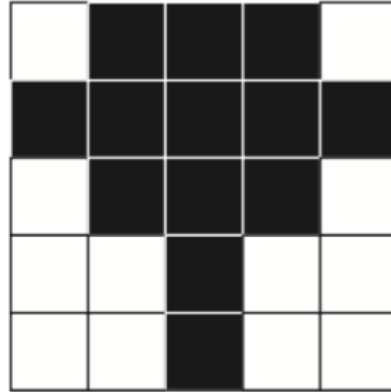
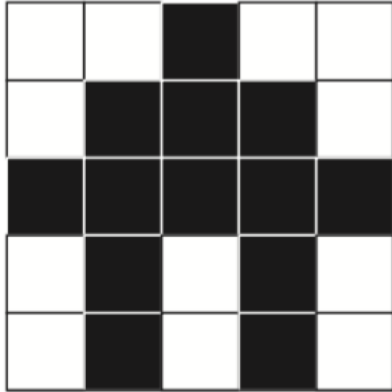


เป็นปุ่มคำสั่งที่จะเผยแพร่โปรแกรมที่สร้างไว้ให้ผู้อื่นได้ลองเล่น โดยเมนูนี้จะสามารถใช้ได้ก็ต่อเมื่อผู้เรียนเข้าสู่ระบบของ Touch Develop แล้วเท่านั้น

- เมื่อจบส่วนของการอธิบายการสร้างเกมส์ Flatverse แล้ว ผู้สอนลองท้าทายให้ผู้เรียนคิดเพิ่มเติมดังนี้

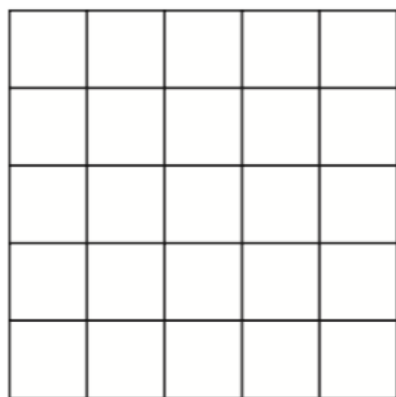
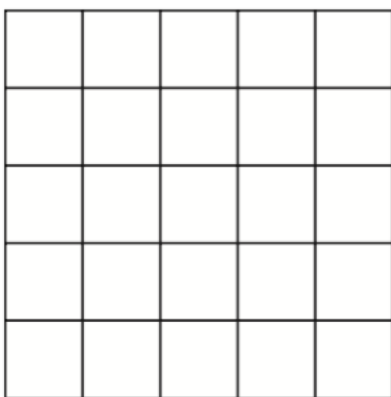
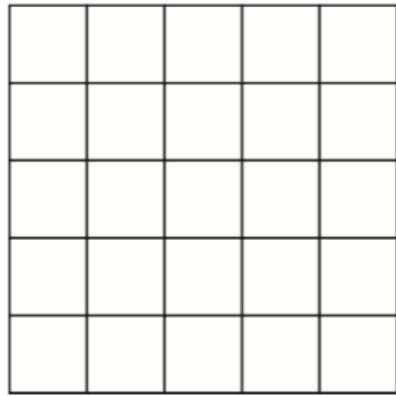
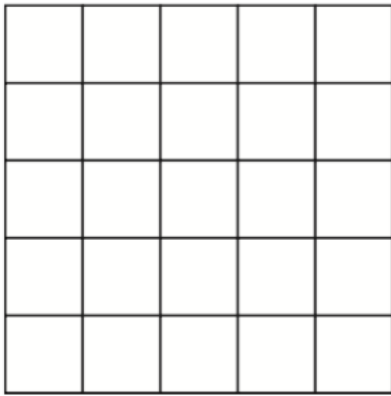
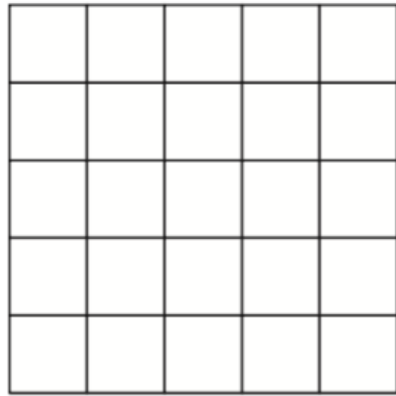
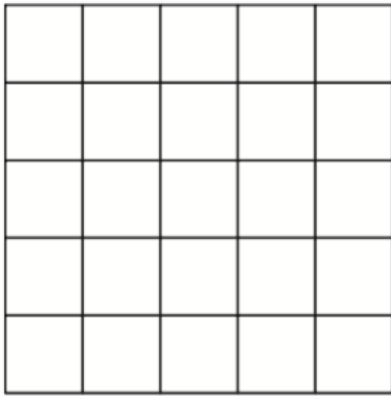
- ในเกมส์ Flatverse ที่มีอยู่นั้น สามารถปรับส่วนไหนของเกมส์เพิ่มเติมได้บ้าง เพื่อให้เกมส์มีความท้าทายยิ่งขึ้น และต้องปรับที่บริเวณใดของ Code

กิจกรรม 2: เขียนโปรแกรมกระดานกราฟ (โจทท์)



อ้างอิง: code.org

กิจกรรม 2: เขียนโปรแกรมกระดาษกราฟ



อ้างอิง: code.org

กิจกรรม 4: ทดลองเขียนโปรแกรมด้วยตนเองผ่านเกมส์ Flatverse โดยใช้ Microsoft Touch Develop

ขั้นตอนที่	การดำเนินการ	ถอดบทเรียนที่สำคัญ
1	สร้างกระดานเปล่า	
2	ปรับค่าจำนวนชีวิตเหลือ 1 ชีวิต	
3	สร้างวงกลมยานอวกาศ	รู้จัก Sprite
4	เปลี่ยนสีวงกลมยานอวกาศเป็นสีเขียว	เรียกใช้ฟังก์ชันของ Sprite
5	เปลี่ยนค่าแรงวงกลมยานอวกาศออก (ให้เป็น Flat Design)	
6	เริ่มสร้าง Event ของการ Tap หน้าจอ (เพื่อให้ยานวิ่งไปตามทิศที่แตะหน้าจอ)	รู้จัก Event Handle
7	ตั้งค่า Speed เริ่มต้นของยานเป็น 80	
8	เพิ่มการเรียกฟังก์ชันคำนวณความเร็วยานในแกน x, y โดยรับพารามิเตอร์เป็นค่า x, y ของตำแหน่งที่ผู้เล่นแตะจอ	
9	เพิ่มการเรียกฟังก์ชันที่ให้นานวิ่งในฟังก์ชัน main	เข้าใจการทำงานของฟังก์ชัน main และฟังก์ชันอื่นๆ
10	เอาการตั้งค่า Speed 80 ออกจากฟังก์ชัน main	
11	วางการตั้งค่า Speed 80 ไว้ที่ฟังก์ชัน set game control value ซึ่งเป็นฟังก์ชันรวมการตั้งค่าเกมส์ทั้งหมดแทน	
12	เพิ่มให้มีการเรียกฟังก์ชัน set game control value ในฟังก์ชัน main เพื่อให้มีการตั้งค่าที่จำเป็นเมื่อเริ่มเกมส์	
13	เพิ่มการตั้งค่า bounce on sides ให้กับเกมส์เพื่อให้ยานชนขอบจอแล้วจะกระเด็นกลับ ไม่ให้ทะลุออกนอกจอไป	
14	สร้างวงกลม Chaser (ผู้ไล่ล่ายานอวกาศ)	
15	ตั้งค่าให้ Chaser มีตำแหน่งเกิดแบบสุ่ม โดยใช้ mathrandom และให้ random ค่า x ระหว่าง 0 กับ boardwidth และค่า y ระหว่าง 0 กับ boardheight	รู้จักการใช้งานฟังก์ชันที่เกี่ยวข้องกับ Math
16	ใส่ฟังก์ชัน create safe zone เพิ่มให้กับ Chaser เพื่อป้องกันการสุ่มเกิดแล้วได้ตำแหน่งที่ทับกับยานอวกาศ	
17	ตั้งค่าให้ Chaser เพิ่มอีก คือ สุ่มสี และ เอาค่าแรงออก (Flat Design)	
18	เรียกใช้งานฟังก์ชัน set sprite position ใน main เพื่อให้ Chaser เกิดตำแหน่งสุ่มขึ้นในเกมส์จริง	
19	สร้างฟังก์ชันใหม่ขึ้นเองชื่อ set sprite size โดยรับพารามิเตอร์เป็น Sprite เพื่อเอาไว้กำหนดขนาดของ Chaser	รู้จักวิธีการสร้างฟังก์ชันขึ้นใหม่เพื่อใช้งานเองและพารามิเตอร์
20	ระบุในฟังก์ชัน set sprite size ว่า size มีค่าเป็น 20 พิกเซล	
21	เพิ่มการ random เลขระหว่าง 0,1,2 ขึ้นมา โดยถ้าขนาดเป็น 0,1 แปลว่ายานเราเกิน Chaser นั้นได้ แต่ถ้า 2 แปลว่า Chaser ใหญ่กว่ายานเรา	
22	เขียนเพิ่มว่า if random number มีค่าเป็น 0 หรือ 1 จะให้ขนาดของ Chaser ตัวนั้นมีขนาด random ตั้งแต่ 5 พิกเซล ไปจนถึงขนาดของ hero - 1 พิกเซล	รู้จักการใช้เงื่อนไข If-Else

ขั้นตอนที่	การดำเนินการ	ถอดบทเรียนที่สำคัญ
23	เขียนเพิ่ม else เพื่อจัดการกรณีสุ่มได้เลข 2 ว่า else แล้วจะให้ขนาดของ chaser ตัวนั้น random ตั้งแต่ค่าขนาด hero size ไปจนถึงขนาดของ max hero size (เป็นค่าที่อยู่ในฟังก์ชัน set game control value อยู่แล้ว)	
24	เขียนเพิ่มให้นำค่า random ขนาดที่ได้จาก if-else จากขั้นตอนก่อนหน้ามา กำหนดเป็นขนาดของparameter ของฟังก์ชัน	
25	นำฟังก์ชัน set sprite size ไปใส่ในฟังก์ชันmain เพื่อใช้งานจริง	
26	เริ่มทำให้ Chaser ไล่ตามยานได้ โดยเริ่มใส่ Event on every frame เข้าไป (เพื่อจับตาทุกเฟรมของเกม)	
27	เขียนเพิ่มให้ Chaser วิ่งเข้าหายาน โดยใช้ฟังก์ชัน move towards และระบุ parameter ของการ move towards คือ Sprite ยานอวกาศและเลขความเร็วที่ต้องการ	
28	เริ่มทำเรื่องการกินระหว่างยานกับ Chaser โดยใช้ฟังก์ชัน overlaps with เข้ามา (if chaser overlaps with hero then...)	
29	ให้เช็คเงื่อนไขเพิ่มเติมอีกว่า เมื่อเกิด overlap กันแล้ว ระหว่างChaser กับยาน ใครมีขนาดใหญ่กว่า หากยานใหญ่กว่า (โดยเปรียบเทียบค่าwidth ของทั้งคู่) ให้ผู้เล่นได้คะแนนเพิ่มเท่ากับขนาดของ Chaser ที่กินได้ พร้อมทั้งสร้าง Chaser ตัวใหม่ขึ้นมา	
30	สร้างฟังก์ชัน shrink sprite ขึ้นมา เพื่อให้ลดขนาดของยานกรณียานถูกกิน โดยรับพารามิเตอร์เป็น Sprite ที่ต้องการให้ลดขนาด	
31	เขียนให้ฟังก์ชัน shrink sprite โดย set size เป็นค่าใหม่คือhero size*hero shrinking factor (ค่า factor นี้อยู่ในฟังก์ชัน set game control value)	
32	นำฟังก์ชัน shrink sprite กลับไปใส่ในelse (ส่วนที่เป็นเงื่อนไขที่ทำเมื่อยาน ถูกกิน ต่อจากข้อ29)	
33	เพิ่มเงื่อนไขอีก โดยหากขนาดยานเล็กจนถึงค่า minimum size (ซึ่งเป็น factor ในฟังก์ชัน set game control value) ให้หักชีวิตของผู้เล่นและให้ game over	
34	เริ่มสร้างอุปสรรคบนหน้าจอโดยใช้ Sprite Set	รู้จักตัวแปรกลุ่ม (Array)
35	ทำการเพิ่ม Sprite เข้าไปใน Sprite Set และนำ Sprite จาก Sprite Set มาวาง ในหน้าจอเกมสแบบสุ่มตำแหน่ง ซึ่งเราต้องการสร้างอุปสรรคจำนวน 50 ชิ้น ดังนั้นจึงเอา for loop เข้ามาช่วย	รู้จักการใช้คำสั่งวนรอบ For Loop
36	ทำการตรวจเงื่อนไขเรื่องการชนกันของยานกับอุปสรรคเพิ่ม โดยนำfor each loop เข้ามาไล่ทำงานกับ obstacle แต่ละชิ้นใน obstacle set และตรวจสอบ เงื่อนไขการoverlap กันระหว่าง obstacle กับยาน	รู้จักการใช้คำสั่งวนรอบ For Each Loop
37	เขียนการทำงาน เมื่ออุปสรรคมีขนาดเล็กกว่ายาน (โดยการเปรียบเทียบ width ของทั้งคู่) หากเล็กกว่า ให้ผู้เล่นได้คะแนนเพิ่มตามขนาดของอุปสรรค และสร้างอุปสรรคชิ้นใหม่โดยสุ่มขนาดและตำแหน่งเกิด	

ขั้นตอนที่	การดำเนินการ	ถอดบทเรียนที่สำคัญ
38	เขียนทำงานใน else หากอุปสรรคใหญ่กว่า ให้ขนาดของยานลดลง โดยใช้ฟังก์ชัน shrink sprite ที่เคยเขียนไว้ก่อนหน้านี้	
39	เขียนฟังก์ชันให้ยานสามารถใหญ่ขึ้นได้บ้าง โดยสร้างฟังก์ชันใหม่ชื่อ grow sprite โดยมีพารามิเตอร์เป็น Sprite ชื่อว่าgrowing sprite	
40	เขียนการทำงานของฟังก์ชัน grow sprite ให้เพิ่มขนาด sprite โดยนำค่า hero grow factor จากฟังก์ชัน set game control value มาใช้ โดยให้ new hero size เท่ากับ hero size*hero grow factor	
41	นำฟังก์ชัน grow sprite ไปใช้ในทุกเงื่อนไขที่เป็นยานกินอุปสรรคหรือ Chaser	
42	เพิ่มรายละเอียด โดยตรวจสอบเพิ่มที่ Eventon every frame ว่า หาก Chaser วิ่งชนกับอุปสรรค ให้มีการbounce ตัว Chaser ด้วย	
43	ปรับให้ speed ของยานเปลี่ยนแปลงได้ตามขนาดของยาน โดยยิ่งยานมีขนาดใหญ่ขึ้น ความเร็วของยานจะยิ่งลดลง โดยเพิ่มการทำงานในส่วนเงื่อนไขกรณีที่ขนาดของยานยังคงใหญ่กว่าค่าminimum hero size อยู่ค่า hero speed factor จะถูกปรับลดลงตามขนาดของยานที่เพิ่มขึ้น	
44	เพิ่มการตรวจสอบเงื่อนไขสุดท้าย โดยตรวจสอบว่าขนาดยาน มีขนาดใหญ่เกินค่า maximum hero size ที่ตั้งค่าไว้แล้วหรือไม่ หากมีขนาดเกินแล้ว จะให้ขนาดของยานกลับมาเป็นขนาดเริ่มต้น	
45	จบการส่วนของการเขียนเกมส์ Flatverse ให้ผู้สอนท้าทายให้ผู้เรียนลองคิดการปรับเกมส์เองเพิ่มเติม ว่าสามารถปรับที่ส่วนไหนของcode ที่มีอยู่ได้บ้าง เช่น ปรับค่าmaximum hero size หรือปรับให้อุปสรรคมีสีเดียวกันหมด หลังจากนั้น ให้ท้าทายผู้เรียนให้ลองมองสิ่งรอบๆ ตัว ว่าสามารถที่จะแก้ไขโปรแกรมของสิ่งที่มีอยู่รอบๆ ตัวให้ทำงานต่างออกอย่างไรได้บ้าง	

แหล่งเรียนรู้เพิ่มเติม

สื่อการเรียนรู้	ระดับ
<u>Touch Develop</u> สอนการเขียนโปรแกรมโดยใช้ Touch Develop เป็นแหล่งความรู้ที่รวบรวมเกมส์ tutorial และตัวอย่างโปรแกรมต่างๆ	ทุกระดับ
<u>Kodu Game Lab</u> สอนการใช้ Kodu ซึ่งเป็นเว็บแอปพลิเคชันที่ใช้ภาษา Blockly (เหมือนกับ Hour of Code) เพื่อสร้างเกมส์บน PC และ Xbox	ทุกระดับ
<u>Alice</u> สื่อ 3 มิติที่สอนการเขียนโปรแกรมผ่านการสร้างเกมส์ และแอนิเมชัน	ทุกระดับ
<u>Code Studio</u> สื่อ blended learning ที่สอนเรื่องวิทยาการคอมพิวเตอร์โดยไม่ต้องใช้คอมพิวเตอร์ (unplugged)	ทุกระดับ
<u>ScratchEd</u> สอนการใช้ Scratch ซึ่งเป็นเว็บแอปพลิเคชันที่ใช้ภาษา Blockly (เหมือนกับ Hour of Code) เพื่อสร้างโปรแกรมต่างๆ เช่นเกมส์ แอนิเมชัน	ทุกระดับ

สื่อการเรียนรู้	ระดับ
<p><u>Middle School CS in Science</u> หลักสูตรวิทยาการคอมพิวเตอร์ผ่านบทเรียนวิทยาศาสตร์ ความยาว 20 บทเรียน</p>	มัธยมต้นขึ้นไป
<p><u>Middle School CS in Algebra</u> หลักสูตรวิทยาการคอมพิวเตอร์ผ่านบทเรียนพีชคณิต ความยาว 20 บทเรียน</p>	มัธยมต้นขึ้นไป
<p><u>Computer Science Fundamental</u> หลักสูตรพื้นฐานวิทยาการคอมพิวเตอร์ ความยาว 20 ชั่วโมง</p>	มัธยมต้นขึ้นไป
<p><u>Beauty and Joy of Computing</u> หลักสูตรวิทยาการคอมพิวเตอร์เบื้องต้น ยาวหนึ่งภาคการศึกษา</p>	มัธยมปลายขึ้นไป
<p><u>Code Academy</u> เว็บไซต์สอนการเขียนโปรแกรมในระดับมืออาชีพ</p>	มัธยมปลายขึ้นไป

หากมีข้อเสนอแนะ กรุณาติดต่อ

Saturday School saturdayschoolthailand@gmail.com

Good factory feedback@goodfactory.co