

# บอทคลิกอัตโนมัติผ่านอีมูเลเตอร์

Automatic Click Bot via Emulator Game



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสยาม

ปีการศึกษา 2565

บอทคลิกอัตโนมัติผ่านอีมูเลเตอร์

Automatic Click Bot via Emulator Game

สรวิชัย สุขผล

วิทยา ชั้นพรหม

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสยาม

ปีการศึกษา 2565

..... ประธานกรรมการสอบปริญญาานิพนธ์


(อาจารย์ สรายุทธ อินทรเสมา)

..... กรรมการสอบ

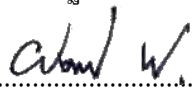
(อาจารย์ภูติษฐ์ วงศ์เจตจันทร์)

..... กรรมการผู้ทรงคุณวุฒิ

(พลอากาศโท ดร.พาหรัณ สงวนนโกชัย)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์

(พันตรี ดร.นรณัฐ สงวนศักดิ์โยธิน)

..... คณบดีคณะวิศวกรรมศาสตร์


(ผู้ช่วยศาสตราจารย์ ดร.ขงยุทธ นารายณ์)

ชื่อโครงการ : บอทคลิกอัตโนมัติผ่านอีเมลเตอร์  
 หน่วยกิต : 2 หน่วยกิต  
 ผู้จัดทำ : นาย สรวิษฐ์ สุขผล 6204000012  
 นาย วิทยา ชันพรม 6323000002  
 อาจารย์ที่ปรึกษา : อาจารย์ สรายุทธ อินทรเสมา  
 ระดับการศึกษา : ปริญญาตรี  
 ภาควิชา : วิศวกรรมคอมพิวเตอร์  
 คณะ : วิศวกรรมศาสตร์  
 ภาคการศึกษา/ ปีการศึกษา : 1/2565

#### บทคัดย่อ

โครงการนี้นำเสนอบอทคลิกเกมสล็อตอัตโนมัติผ่านอีเมลเตอร์ ซึ่งเป็นเกมที่เล่นผ่านสมาร์ตโฟน โดยเปิดเกมผ่านโปรแกรมจำลอง ซึ่งเขียนโค้ดด้วยภาษาไพทอน ร่วมกับชุดคำสั่งโอเพ่นซีวีในการจับภาพ และคำสั่งการคลิกอัตโนมัติตามที่ต้องการ โดยมีการเขียนอัลกอริทึมสำหรับการจับภาพธนบัตร การจับภาพกากบาทในตัวเกม และคลิกอัตโนมัติเมื่อเจอรูปภาพนั้น ซึ่งโปรแกรมจะทำงานตั้งแต่เปิดเกมจนถึงการทำงานต่างๆภายในเกมตามที่ได้กำหนดไว้ รวมถึงการส่งเข้าเกมให้ใหม่หากเกมมีปัญหา โดยสามารถเปิดได้หลายหน้าจอ โดยไม่ทำการแย่งเมาส์จากผู้ใช้งาน โครงการนี้ถูกออกแบบและพัฒนาให้ทำงานสำหรับหน้าจอกอมพิวเตอร์ขนาดร้อยเปอร์เซ็นต์ และขนาดหน้าต่างของอีเมลเตอร์ให้มีขนาด 900:1600 โดยผลการทดลองโปรแกรมเห็นได้ว่าโปรแกรมทำงานได้อย่างดีตั้งแต่การเข้าเกมสัจจนไปถึงการทำงานภายในเกม และสามารถแยกจอการทำงานได้ตามจุดประสงค์ที่วางไว้

คำสำคัญ: บอทคลิกอัตโนมัติ , ภาษาไพทอน , อีเมลเตอร์ , การจับภาพโดยใช้โอเพ่นซีวี

อาจารย์ที่ปรึกษา.....

(อาจารย์ สรายุทธ อินทรเสมา)

..... 6 / ๗๕, 256๕ .....

**Project Title** : Automatic Click Bot via Emulator Game  
**Credits** : 2 credits  
**By** : Mr. Sorawit Sukphon 6204000012  
Mr. Witaya Khanprom 6323000002  
**Advisor** : Mr. Sarayut Intarasema  
**Degree** : Bachelor of Engineering  
**Department** : Computer Engineering

**Semester / Academic year** : 1/2021

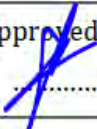
### Abstract

This project proposed an automatic, emulator-based smartphone game clicking system on a Python-language emulator. OpenCV was used for image recognition, and automatic clicking commands were written. In addition, an algorithm for banknote recognition, cross symbol recognition, and automatic clicking of a picture was written. The program starts at the beginning of the game, and carries out commands as instructed, including retrying the game in case of problems. This program can be opened as multiple instances without user mouse input, and was designed to work 100% for computer screens and the emulator screen size of 900x1600. Experiments showed that the program performed satisfactorily from the beginning to the game operation, and could be split into multiple instances per the objectives.

**Keywords:** autoclicking bot, Python, emulator, OpenCV Image Recognition

Advisor \_\_\_\_\_ 

(Mr. Sarayut Intarasema)

Approved by  
  
.....

## กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์ สรายุทธ อินทรเสมา อาจารย์ที่ปรึกษา ซึ่งมีความกรุณาเสียสละเวลาเพื่อให้ความรู้ และคำแนะนำตลอดไปจนถึงการทำโครงการ

ขอขอบพระคุณ คณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสยาม ตลอดไปจนถึงผู้เกี่ยวข้องทุกท่านที่ไม่ได้กล่าวมา ณ ที่นี้

คณะผู้จัดทำ

นาย วิทยา ชันพรม

นาย สรวิชญ์ สุขผล



## สารบัญ

	หน้า
บทคัดย่อ (ภาษาไทย)	ข
บทคัดย่อ (ภาษาอังกฤษ)	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญรูป	ฉ
<b>บทที่ 1 บทนำ</b>	
1.1 ที่มาของโครงการ	1
1.2 คำสำคัญ	1
1.3 วัตถุประสงค์ในการทำโครงการ	2
1.4 ขอบเขตความสามารถของโครงการ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 เครื่องมือ เครื่องวัด และอุปกรณ์ที่ใช้	2
1.7 แผนการดำเนินงาน	3
<b>บทที่ 2 ทฤษฎีที่เกี่ยวข้อง</b>	
2.1 ไพทอน (Python)	4
2.2 โอเพ่นซีวี (OpenCV)	6
2.3 Autoit	8
2.4 Photoshop	10
2.5 Visual Studio Code	11
2.6 โปรแกรม NOX	12
2.7 Package/Module	15
2.8 ตัวเกมมือถือแห่งความสุข	20
2.9 Line notify	22
<b>บทที่ 3 การวิเคราะห์และการออกแบบ</b>	
3.1 การออกแบบระบบ	26
3.2 ภาพรวมการทำงานของโปรแกรม	27
3.3 การออกแบบลำดับขั้นตอนการทำงานของบอททั้งหมด	28

สารบัญ (ต่อ)

	หน้า
3.4 คำสั่งที่จำเป็นในการสร้างบอท	29
3.5 การออกแบบการทำงานของบอทในแต่ละส่วน	32
3.6 การออกแบบการตั้งค่าหน้าจอของโปรแกรมจำลอง	39
3.7 คุณลักษณะของเครื่องคอมพิวเตอร์ที่ใช้ทำบอท	41
3.8 ออกแบบการจับภาพเข้า linenotify	42
3.9 การออกแบบส่งภาพและข้อความเข้าไลน์	43
3.10 การออกแบบการส่งภาพและข้อความเข้า line notify	44
3.11 การแปลงไฟล์เป็น .exe	45
<b>บทที่ 4 ขั้นตอนการทดลอง</b>	
4.1 การทดลองคำสั่งเชื่อมต่อ ADB และ อิมูเรเตอร์	47
4.2 การทดลองคำสั่งการแสดงค่าอิมูเรเตอร์เก็บไว้ในรูปเพื่อแสดงค่าทีละตัว	47
4.3 การทดสอบนำค่าของอิมูเรเตอร์เข้าไปในฟังก์ชัน	48
4.4 ตรวจสอบการอยู่น้ำหลัก	49
4.5 ตรวจสอบรูปไม้กวาดและรูปกระดาด	50
4.6 ตรวจสอบรูปเงิน	51
4.7 ตรวจสอบรูปยื่นยัน	52
4.8 เลื่อนหน้าจอเปลี่ยนไปหน้าครัว	53
4.9 การตรวจสอบการอยู่น้ำครัว	55
4.10 ตรวจสอบรูปจานอาหารที่ 1 และ ที่ 2 พร้อมรูปจานในอ่างล้างจาน	56
4.11 เลื่อนหน้าจอเปลี่ยนไปหน้าครัว	58
4.12 ตรวจสอบรูปแมว	59
4.13 ตรวจสอบหน้ารับรางวัล	61
4.14 รับรางวัลครั้งที่ 1	62
4.15 กดรับกล่อง	63
4.16 กดรับของรางวัลครั้งที่ 2	64
4.17 กดปิดหน้าต่างรับรางวัล	65
4.18 ทดสอบการรีเซ็ตเกมตามเวลาที่กำหนด	66

## สารบัญ (ต่อ)

	หน้า
4.19 ทดสอบคำสั่งการหาชื่อแอปพลิเคชัน	67
4.20 ทดสอบการกดเข้าเกมเอง	68
4.21 แยกการทำงานหลายหน้าจอ	69
4.22 ทดสอบการทำงานบอทโดยทับหน้าจอและไม่ดึงเมาส์	70
4.23 ทดสอบการเชื่อมต่อ line notify	71
4.24 ทดสอบการส่งภาพและข้อความเข้า line notify	72
4.25 ทดสอบการรันตัวบอทผ่านไฟล์ exe	73
4.26 ทดสอบประสิทธิภาพหาเงินในเกมระหว่างใช้บอทเล่นเกมกับเล่นด้วยมือ อย่างละ 1 ชั่วโมง	74
4.27 ทดสอบการใช้บอทกับโปรแกรมจำลองที่ขนาดไม่ตรงกับโปรแกรมบอท	76
<b>บทที่ 5 สรุปผลและข้อเสนอแนะ</b>	
5.1 สรุปผลการดำเนินโครงการ	77
5.2 ปัญหาการดำเนินโครงการ	77
5.3 ข้อเสนอแนะ	77
บรรณานุกรม	78
ภาคผนวก	79
ภาคผนวก ก ขั้นตอนการติดตั้งโปรแกรมจำลอง และ Visual Studio Code	80
ประวัติผู้จัดทำ	92



## สารบัญตาราง

	หน้า
ตารางที่ 1.1 การวางแผนการดำเนินงาน	3
ตารางที่ 3.1 คุณลักษณะของเครื่องคอมพิวเตอร์ที่ใช้ทำบอท	41



## สารบัญรูป

	หน้า
รูปที่ 2.1 เปรียบเทียบการเขียน Code Python และ Java	5
รูปที่ 2.2 โอเพนซีวี	6
รูปที่ 2.3 คำสั่งติดตั้ง OpenCv-Python	7
รูปที่ 2.4 ติดตั้ง Opencv ใน cmd	7
รูปที่ 2.5 ติดตั้ง Opencv เรียบร้อย	7
รูปที่ 2.6 Autoit	8
รูปที่ 2.7 การหาค่าสีและตำแหน่งของรูปภาพ	9
รูปที่ 2.8 ค่าสีและตำแหน่งของรูปภาพ	9
รูปที่ 2.9 Photoshop	10
รูปที่ 2.10 Visual studio code	11
รูปที่ 2.11 โปรแกรม Nox	12
รูปที่ 2.12 เว็บไซต์โหลด Nox	13
รูปที่ 2.13 ติดตั้ง Nox	13
รูปที่ 2.14 ติดตั้ง Nox สำเร็จ	13
รูปที่ 2.15 หน้าต่าง โปรแกรม Nox	14
รูปที่ 2.16 Package/Module Numpy	15
รูปที่ 2.17 คำสั่งติดตั้ง Numpy	15
รูปที่ 2.18 คำสั่งการนำ Numpy มาใช้	15
รูปที่ 2.19 Package/Module Pillow	16
รูปที่ 2.20 คำสั่งติดตั้ง Pillow	16
รูปที่ 2.21 คำสั่งการนำ Pillow มาใช้	16
รูปที่ 2.22 Package/Module Pure python adb	17
รูปที่ 2.23 คำสั่งติดตั้ง Pure python adb	17
รูปที่ 2.24 คำสั่งการนำ Module Pure python adb มาใช้	17
รูปที่ 2.25 Package/Module Multipricess	18
รูปที่ 2.26 คำสั่งติดตั้ง Multipricess	18
รูปที่ 2.27 Package/Module Auto-py-to-exe	19

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 2.28 คำสั่งติดตั้ง Auto-py-to-exe	19
รูปที่ 2.29 การแปลงไฟล์เป็น .exe	19
รูปที่ 2.30 ตัวเกมหม้อไฟแห่งความสุข	20
รูปที่ 2.31 ภาพตัวอย่างภายในเกม 1	20
รูปที่ 2.32 ภาพตัวอย่างภายในเกม 2	21
รูปที่ 2.33 ภาพตัวอย่างภายในเกม 3	21
รูปที่ 2.34 Linenotify	22
รูปที่ 2.35 ขั้นตอนที่ 1 ของการการออกโทเค็น line notify	22
รูปที่ 2.36 ขั้นตอนที่ 2 ของการการออกโทเค็น line notify	23
รูปที่ 2.37 ขั้นตอนที่ 3 ของการการออกโทเค็น line notify	23
รูปที่ 2.38 ขั้นตอนที่ 4 ของการการออกโทเค็น line notify	23
รูปที่ 2.39 ขั้นตอนที่ 5-6-7 ของการการออกโทเค็น line notify	24
รูปที่ 2.40 ขั้นตอนที่ 8 ของการการออกโทเค็น line notify	24
รูปที่ 2.41 เชื่อมต่อ line notify เรียบร้อยแล้ว	25
รูปที่ 3.1 ภาพรวมการทำงานของระบบตอนรันโปรแกรม	27
รูปที่ 3.2 บล็อกไดอะแกรมการทำงานของบอท	28
รูปที่ 3.3 การทำงานในหน้าหลัก	33
รูปที่ 3.4 บล็อกไดอะแกรมการทำงานของบอทในหน้าหลักหารูปบิล	34
รูปที่ 3.5 บล็อกไดอะแกรมการทำงานของบอทในหน้าหลักหารูปไม้กวาด	34
รูปที่ 3.6 บล็อกไดอะแกรมการทำงานของบอทในหน้าหลักหารูปเงินและรูยีนยัน	34
รูปที่ 3.7 บล็อกไดอะแกรมการทำงานของบอทในหน้าหลักหารูปแมวและรับรางวัล	34
รูปที่ 3.8 การเลื่อนไปหน้าครัวและกลับมาหน้าหลัก	35
รูปที่ 3.9 การทำงานในหน้าครัว	36
รูปที่ 3.10 การรีเซตเกม	37
รูปที่ 3.11 การทำงานในส่วนยืนยันข้อตกลงและกดเข้าร้าน	38
รูปที่ 3.12 การทำงานหลายหน้าจอ	38
รูปที่ 3.13 ตั้งค่าขนาดหน้าจอโปรแกรมจำลอง	39

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.14 ขนาด 900X1600 ของโปรแกรมจำลอง	40
รูปที่ 3.15 ตั้งค่าเฟรมเรตโปรแกรมจำลอง	41
รูปที่ 3.16 การแคปรูปและเซฟลงไฟล์เดอร์	42
รูปที่ 3.17 บล็อกไดอะแกรมการทำงานของบอท	43
รูปที่ 3.18 การส่งรูปและข้อความเข้า linotify	44
รูปที่ 3.19 การแปลงไฟล์เป็น .exe	46
รูปที่ 4.1 ผลการทดลองคำสั่ง เชื่อมต่อ ADB และ อิมูเรเตอร์	47
รูปที่ 4.2 การแสดงค่าของอิมูเก็บไว้ในรูป	47
รูปที่ 4.3 การแสดงค่าลิตรในฟังก์ชัน	48
รูปที่ 4.4 ไดอะแกรมการเชื่อมต่อ ADB และ EMU	48
รูปที่ 4.5 การตรวจสอบการอยู่หน้าหลัก	49
รูปที่ 4.6 ตรวจสอบรูปไม้กวาดและรูปกระดาษ	50
รูปที่ 4.7 ตรวจสอบรูปเงิน	51
รูปที่ 4.8 ตรวจสอบรูปยืนยัน	52
รูปที่ 4.9 เลื่อนหน้าจอเปลี่ยนไปหน้าครัว	53
รูปที่ 4.10 หน้าครัว	54
รูปที่ 4.11 การตรวจสอบการอยู่หน้าครัว	55
รูปที่ 4.12 รูปจานอาหารที่ 1 และ ที่ 2	56
รูปที่ 4.13 รูปอ่างล้างจาน	57
รูปที่ 4.14 เลื่อนหน้าจอเปลี่ยนไปหน้าครัว	58
รูปที่ 4.15 ตรวจสอบรูปแมว	59
รูปที่ 4.16 การหาค่าตำแหน่งรูปแมว	60
รูปที่ 4.17 การหาค่าสีรูปแมว	60
รูปที่ 4.18 ตรวจสอบหน้ารับรางวัล	61
รูปที่ 4.19 รับรางวัลครั้งที่ 1	62
รูปที่ 4.20 กดรับกล่อง	63
รูปที่ 4.21 รับรางวัลครั้งที่ 2	64

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.22 กดปิดหน้ารับรางวัล	65
รูปที่ 4.23 การรีเซ็ตเกม	66
รูปที่ 4.24 ทดสอบคำสั่งการหาชื่อแอปพลิเคชัน	67
รูปที่ 4.25 ทดสอบการกดเข้าเกมอัตโนมัติ	68
รูปที่ 4.26 แยกการทำงานหลายหน้าจอ	69
รูปที่ 4.27 การทำงานบอทโดยทับหน้าจอและไม่ดึงเมาส์	70
รูปที่ 4.28 ทดสอบการเชื่อมต่อ line notify	71
รูปที่ 4.29 ทดสอบการส่งภาพและข้อความเข้า line notify	72
รูปที่ 4.30 ทดสอบการรันตัวบอทผ่านไฟล์ .exe	73
รูปที่ 4.31 ทดสอบประสิทธิภาพหาเงินในเกม 1 ชั่วโมง ด้วยผู้เล่น	74
รูปที่ 4.32 ทดสอบประสิทธิภาพหาเงินในเกม 1 ชั่วโมง ด้วยการใช้บอทเล่น	75



# บทที่ 1

## บทนำ

### 1.1 ที่มาของโครงการ

ในปัจจุบัน โลกแห่งเกมออนไลน์ไม่ใช่เพียงแค่เล่นเพื่อสนุกเท่านั้น แต่ยังสามารถหารายได้จากการเล่นได้อีกด้วย ไม่ว่าจะเป็นเกมสลิปสองหางออนไลน์ ที่สามารถใช้บอทไปหารายได้จากการพาเวลดได้ หรือเกมMIRM ที่ใช้บอทปล่อยให้สามารถหาของภายในเกมเองได้ และสามารถนำของที่ได้จากในเกมไปขายต่อเป็นเงินจริงได้ และยังมีเกมอีกมากที่สามารถหาเงินจากการเล่นเกมได้ ไม่ว่าจะเป็นเกมบอมคริปโต เกมROX เป็นต้น

การเล่นเกมที่เป็นเกมแนวเก็บเวลหรือเกมแนวที่ใช้เงินจริงลงไปเล่น และเกมที่ต้องมีการเฝ้าจอตลอดเวลา ทำให้ผู้เล่นไม่สามารถเล่นได้ตลอด 24 ชั่วโมง หรือบางวันก็ไม่ได้มีเวลาว่างมากพอที่จะเล่นเกม เลยทำให้ผู้เล่นเสียผลประโยชน์จากการเล่นเกม หากเป็นเกมแนวคริปโตที่ต้องใช้เงินจริงลงไปเล่นก็จะทำให้ผู้เล่นคืนทุนได้ช้าหรือบางทีก็อาจขาดทุน ดังนั้นเวลาในการเล่นเกมที่เฝ้าจอตลอด 24 ชั่วโมง จึงเป็นสิ่งสำคัญมาก และเป็นปัจจัยสำคัญในการเล่นเกมที่ทำให้ตัวละครเติบโตได้ไว หรือเงินที่เราลงทุนไปได้ทุนคืนไวขึ้น

จากปัญหาในการเล่นเกมที่เฝ้าจอสามารถเฝ้าจอได้ เพราะบอทสามารถทำงานได้ตลอด 24 ชั่วโมง และมีการทำงานที่หลากหลาย ไม่ว่าจะเป็นการเชื่อมต่องेमอีกครั้งเมื่อเกมหลุด ระบบที่จะช่วยป้องกันการหยุดการทำงานของเครื่องคอมพิวเตอร์ การสั่งให้เม้าส์ของผู้เล่นนั้นไปกดตามจุดต่างๆภายในเกมตามที่ต้องการได้ หรือบอทสามารถทำงานได้หลายหน้าจอบ ระบบคร่าวๆที่กล่าวมาทำให้ผู้เล่นสามารถหาเงินได้มากขึ้น สะดวกสบายในการเล่นเกมที่เฝ้าจอ และทำให้ผู้เล่นสามารถพัฒนาตัวละครของตัวเองได้ไวขึ้น

โดยภาษาที่ใช้ในการพัฒนาบอทนั้น จะเป็นภาษาไพทอนเนื่องจากภาษาไพทอนนั้นมีหลักการเขียนที่ง่ายไม่ซับซ้อน และสามารถเข้าใจได้ง่าย เป็นภาษาที่เหมาะสมสำหรับผู้สนใจในการนำมาพัฒนาบอท อีกทั้งไพทอนยังมีไลบรารีมากมายให้ใช้งานทำให้สะดวกมากๆในการเขียนบอทมากขึ้น

### 1.2 คำสำคัญ (Keyword) สำหรับโครงการ

บอทคลิกอัตโนมัติผ่านอิมูเลเตอร์ / การจับภาพโดยใช้โอเพ่นซีวี

Python / Visual Studio Code / Automatic Bot Game 24 Hours

### 1.3 วัตถุประสงค์ในการทำโครงการ

- 1.3.1 เพื่อศึกษาการพัฒนาบอทเกมด้วยภาษาไพทอน
- 1.3.2 เพื่อศึกษาแนวทางการเขียนโปรแกรมอโต้คลิกนำไปประยุกต์ใช้ลักษณะอื่น เช่น การเทรดหุ้น
- 1.3.3 เพื่อให้โปรแกรมบอททำงานได้ตลอด 24 ชั่วโมง
- 1.3.4 เพื่อให้ผู้เล่นเกมได้รับผลประโยชน์จากการเล่นเกมมากที่สุด

### 1.4 ขอบเขตความสามารถของโครงการ

- 1.4.1 ตัวบอทสามารถทำงานได้หลายหน้าจอเมื่อเปิดเกมเดียวกันพร้อมกัน
- 1.4.2 สามารถเข้าเกมได้เองเมื่อมีการหลุดออกจากเกม
- 1.4.3 ตัวบอทจะไม่แย่งเมาส์จากผู้ใช้งาน
- 1.4.4 บอทสามารถค้นหาภาพและสีภายในเกมได้อย่างถูกต้องและแม่นยำ
- 1.4.5 มีการแจ้งเตือนทาง Line notify โดยจะส่งเป็นรูปภาพและข้อความ
- 1.4.6 มีระบบ Refresh อัตโนมัติ ช่วยป้องกันการหยุดการทำงานของเครื่องคอมพิวเตอร์
- 1.4.7 ตัวบอทสามารถเชื่อมต่อกับโปรแกรมจำลอง Nox ได้
- 1.4.8 ตัวบอทสามารถทำงานได้หากมีการซ้อนทับหน้าจอกัน
- 1.4.9 ตัวเกมต้องมีขนาดหน้าจอ 900 X 1600 และขนาดหน้าจอของคอมพิวเตอร์ต้องตั้งเป็น 100% บอทถึงจะสามารถทำงานได้
- 1.4.10 บอทสามารถทำงานผ่านไฟล์ที่แปลงเป็น .exe ได้

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 ทำให้ตัวละครเกมของผู้เล่นพัฒนาเองได้อัตโนมัติ
- 1.5.2 สามารถลดเวลาในการเล่นเกมเพื่อไปทำกิจกรรมอย่างอื่นได้
- 1.5.3 ได้ความรู้เกี่ยวกับการเขียนบอทด้วยภาษาไพทอน

### 1.6 เครื่องมือ, เครื่องวัด และอุปกรณ์ที่ใช้

1.6.1 คอมพิวเตอร์ตั้งโต๊ะ	1	เครื่อง
1.6.2 โปรแกรม Visual Studio Code	1	ชุด
1.6.3 ตัวเกม หม้อไฟแห่งความสุข	1	ชุด
1.6.4 Autoit	1	ชุด
1.6.5 ตัวโปรแกรมจำลอง Nox	1	ชุด
1.6.6 Adobe Photoshop	1	ชุด

## 1.7 แผนการดำเนินงาน

ตารางที่ 1.1 การวางแผนการดำเนินงาน

เลขที่	หัวข้องาน	%	2565												
			ม.ค.	ก.พ.	มี.ค.	เม.ษ.	พ.ค.	มิ.ย	ก.ค.	ธ.ค.	ก.ย.				
1	ศึกษาข้อมูลเกี่ยวกับโครงการ	8.00	■												
		8.33	▨												
2	นำเสนองานและรูปแบบโครงการ	8.00	■												
		8.33	▨												
3	ศึกษาโครงสร้าง Visual Studio Code และหลักการทำงานของ python	8.00			■										
		8.33			▨										
4	ศึกษาโครงสร้างและหลักการทำงานของ OpenGL	4.00	■												
		4.17			▨										
5	ศึกษาข้อมูลตัวโปรแกรมจำลองและcodeในการเชื่อมต่อโปรแกรม	16.00				■									
		12.51			▨										
6	ออกแบบการทำงาน	12.00				■									
		12.51			▨										
7	ติดตั้งโปรแกรมที่ต้องใช้และเขียนโค้ดบน Visual Studio Code	12.00				■									
		8.33			▨										
8	ทดสอบการทำงานของโปรแกรม	12.00				■									
		12.51			▨										
9	ปรับปรุงและแก้ไขข้อผิดพลาดเกี่ยวกับตัว Code	12.00				■									
		12.51			▨										
10	จัดทำเล่มปริทัศน์ฉบับร่าง	8.00				■									
		12.51				▨									



ตามแผนงาน



ตามงานจริง



## บทที่ 2

### เอกสารและทฤษฎีที่เกี่ยวข้อง

เนื้อหาบทนี้เป็นเอกสาร ทฤษฎีที่เกี่ยวข้องเพื่อใช้ในการปฏิบัติงานในโครงการสร้างบอทคลิกอัตโนมัติผ่านอิมูเลเตอร์ ได้ศึกษาค้นคว้าหาเอกสารการออกแบบโปรแกรมขั้นตอนการสร้างโปรแกรมและความรู้เกี่ยวกับการสร้างบอทเกมอโต้คลิกด้วยภาษาไพทอน

#### 2.1 ไพทอน (Python)

ภาษาโปรแกรม Python[3] คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้ เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ใน ส่วนของการแปลงชุดคำสั่งที่เราเขียนให้เป็นภาษาเครื่อง Python มีการทำงานแบบ Interpreter คือ เป็นการแปลงชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตาม ที่ต้องการ นอกจากนี้ภาษาโปรแกรม Python ยังสามารถที่จะนำไปใช้ในการเขียนโปรแกรมได้ หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึง ทำให้มีการนำไปใช้กันอย่างแพร่หลายในหลายองค์กรใหญ่ระดับโลก เช่น Google, YouTube, Instagram, Dropbox และ NASA เป็นต้น ภาษาโปรแกรมไพทอนนี้เองก็ยังสามารถใช้ได้หลากหลาย Platform ไม่ว่าจะเป็น Microsoft Windows, Unix, Linux และ Mac OS เป็นต้น ดังนั้นขอบเขตของ การพัฒนาโปรแกรมจึงกว้าง ทำให้เป็นที่นิยมอย่างมากและถูกใช้งานอย่างกว้างขวาง ตัวอย่างที่ พอจะรู้จักอย่างเช่น เกมส์ The Sims 4 ที่ นำ Python ไปสร้าง mod หรือนอกเหนือจากการพัฒนาใน ด้าน Data Science แล้วยังถูกใช้ในเรื่องของ AI ที่วิเคราะห์รูปภาพต่างๆ ด้วย

ความสามารถของ Python ยังมีประโยชน์สำหรับการเขียนโค้ดฝั่งเซิร์ฟเวอร์ เนื่องจากมี ไลบรารีจำนวนมากที่ประกอบด้วยโค้ดที่เขียนไว้ล่วงหน้าสำหรับฟังก์ชันแบ็คเอนด์ที่ซับซ้อน นักพัฒนายังใช้เฟรมเวิร์ก Python ที่หลากหลายซึ่งมีเครื่องมือที่จำเป็นทั้งหมดเพื่อสร้างเว็บแอปพลิเคชันได้เร็วขึ้นและง่ายขึ้นอีกด้วย ตัวอย่างเช่น นักพัฒนาสามารถสร้างโครงสร้างเว็บแอปพลิเคชันได้ภายในไม่กี่วินาที เนื่องจากไม่จำเป็นต้องเขียนขึ้นใหม่ทั้งหมด จากนั้นนักพัฒนาสามารถ ทดสอบได้โดยใช้เครื่องมือทดสอบของเฟรมเวิร์ก โดยไม่ต้องพึ่งพาเครื่องมือทดสอบภายนอก

เมื่อเปรียบเทียบ Python กับภาษาอื่นๆอย่างเช่น Java ถือว่า Python เป็นภาษาที่ง่ายที่สุดใน การเรียนรู้ เป็นการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ที่มีการใช้งานง่าย

ประกอบกับมีตัวแปรและ ไลบรารีที่ถูกออกแบบมาให้ดี ความกระชับและง่ายกว่า แคม การใช้ code และการเขียน script ก็สั้น จนสามารถประหยัดเวลาได้มาก

ตัวอย่างง่ายในการเขียนภาษาโปรแกรม ให้มีการแสดงออกมาทางหน้าจอว่า “hello world!”



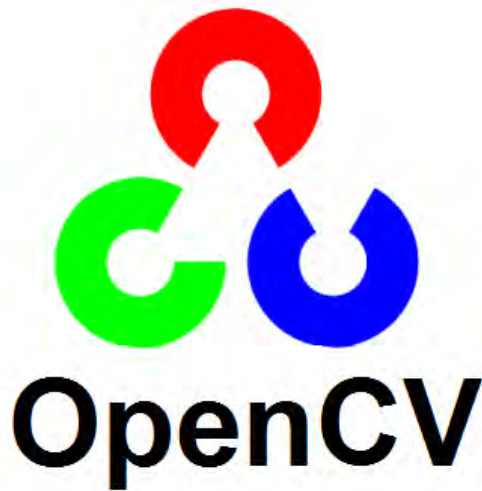
รูปที่ 2.1 การเปรียบเทียบการเขียน Code Python และ Java

เปรียบเทียบภาษา Java และ ภาษา Python เห็นได้ว่า Python มีความสั้นกระชับในการเขียน Code ออกมา และหากมีข้อผิดพลาดเกิดขึ้น โครงสร้างที่ออกมาในรูปแบบไม่ซับซ้อนเกินไปนี้ก็จะทำให้ ง่ายที่จะระบุและ จัดการถึงข้อผิดพลาด ทำให้ประหยัดเวลาลง

#### ระบบอัตโนมัติด้วยสคริปต์ Python

ภาษาการเขียนสคริปต์คือภาษาการเขียนโปรแกรมที่ทำให้งานที่มนุษย์ทำตามปกติเป็นไป โดยอัตโนมัติ โปรแกรมเมอร์จึงใช้สคริปต์ Python อย่างแพร่หลายเพื่อทำให้งานประจำวันหลาย อย่างดังต่อไปนี้เป็นไปโดยอัตโนมัติ

1. การเปลี่ยนชื่อไฟล์จำนวนมากพร้อมกัน
2. การแปลงไฟล์เป็นไฟล์ประเภทอื่น
3. การลบคำที่ซ้ำกันในไฟล์ข้อความ
4. การดำเนินการทางคณิตศาสตร์ขั้นพื้นฐาน
5. การส่งข้อความอีเมล
6. การดาวน์โหลดเนื้อหา
7. การดำเนินการวิเคราะห์พื้นฐาน
8. การค้นหาข้อผิดพลาดในหลายไฟล์



รูปที่ 2.2 โอเพนซีวี

## 2.2 โอเพนซีวี (OpenCV)

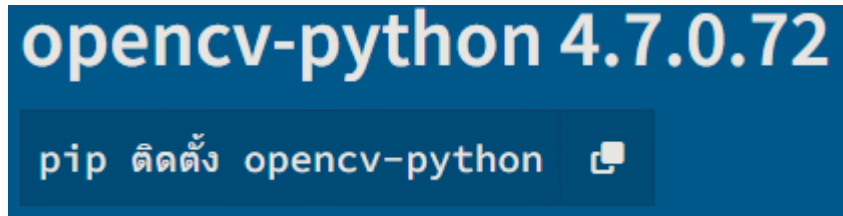
ปัจจุบัน OpenCV (Open source Computer Vision) เป็นไลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) โดยส่วนใหญ่จะมุ่งเป้าไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision) เดิมทีแล้วถูกพัฒนาโดย Intel แต่ภายหลังได้รับการสนับสนุนโดย Willow Garage ตามมาด้วย Itseez (ซึ่งต่อมาก็ก่อตั้งโดย Intel) OpenCV เป็นไลบรารีแบบข้ามแพลตฟอร์ม (Cross-Platform) และใช้งานได้ฟรีภายใต้ลิขสิทธิ์ของ BSD แบบโอเพ่นซอร์ส (Open-Source BSD License) OpenCV ยังสนับสนุนเฟรมเวิร์กการเรียนรู้เชิงลึก (Deep Learning Frameworks) ได้แก่ TensorFlow, Torch/ PyTorch และ Caffe

ความสามารถหลักๆ ของเจ้า OpenCV ก็คือสามารถช่วยให้ผู้ใช้จัดการวิดีโอหรือรูปภาพ เพื่อให้เกิดการเปลี่ยนแปลงกับรูปภาพ อย่างเช่น ทำให้ภาพชัดขึ้น ทำให้เบลอ ลดสัญญาณรบกวน (noise) ในรูปภาพ จากแหล่งที่มาของรูปภาพต่างๆ ไม่ว่าจะเป็น webcam ภาพถ่าย วิดีโอ หรือกล้องวงจรปิด

นอกจากการปรับปรุงภาพแล้วนั้น ยังมีการใช้เทคนิคการประมวลผลภาพ ที่ทำให้รู้จักกับวัตถุนั้นๆ อาจจะเคยได้ยินตัวอย่างมาบ้าง เช่น การแบ่งประเภท Object ว่าเป็น หมา แมว คน รถยนต์ เป็นต้น ไปจนถึงการนำไปใช้งานที่ซับซ้อนมากขึ้น เช่น การจดจำใบหน้าของคน อย่างที่เคยได้เจอในหนัง Action ของฮอลลีวูดที่จะมีตำรวจตามหาคนร้ายจากกล้องวงจรปิด

## การติดตั้ง OpenCV-Python

- เข้าไปที่ลิงค์ <https://pypi.org/project/opencv-python/> และกด Copy ที่ pip ติดตั้ง OpenCv-Python



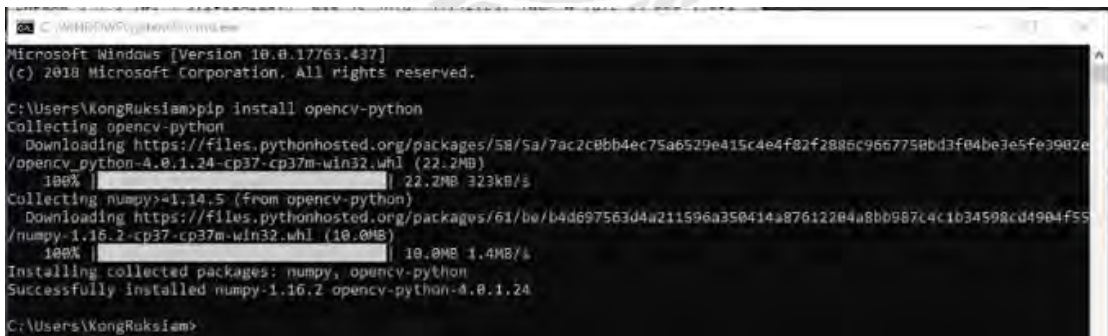
รูปที่ 2.3 คำสั่งติดตั้ง OpenCv-Python

- เปิด Command Prompt (cmd) และใส่คำสั่ง pip install Opencv-python เพื่อเป็นการติดตั้งตัว opencv-python



รูปที่ 2.4 ติดตั้ง Opencv ใน cmd

- ติดตั้งเรียบร้อยแล้ว



รูปที่ 2.5 ติดตั้ง Opencv เรียบร้อย



รูปที่ 2.6 Autoit

### 2.3 Autoit

AutoIt3 ย่อมาจาก AutoIt v 3 Launcher ได้รับการพัฒนาโดย Jonathan Bennett และเปิดตัวในปี 1999 ปัจจุบัน โปรแกรมนี้อยู่ระหว่างการพัฒนาโดยทีม AutoIt

AutoIt v3 เป็นภาษาสคริปต์ที่ออกแบบมาสำหรับการกดแป้นพิมพ์อัตโนมัติการจำลองการเคลื่อนไหวของเมาส์และการควบคุมหน้าต่าง AutoIt เป็นภาษาสคริปต์พีแรว์ที่คล้ายกับ BASIC โปรแกรมสามารถใช้ในการทำให้ Windows GUI เป็นอัตโนมัติและดำเนินงานด้านสคริปต์ทั่วไปและงานประจำ โปรแกรมสนับสนุนการแสดงผลที่ซับซ้อนฟังก์ชันผู้ใช้รูปและอื่น ๆ อีกมากมายให้ผู้ใช้จัดการหน้าต่างและกระบวนการโต้ตอบกับการควบคุมหน้าต่างมาตรฐานและรวบรวมสคริปต์เป็นปฏิบัติการแบบสแตนด์อโลนกับ GUI ทำเอง

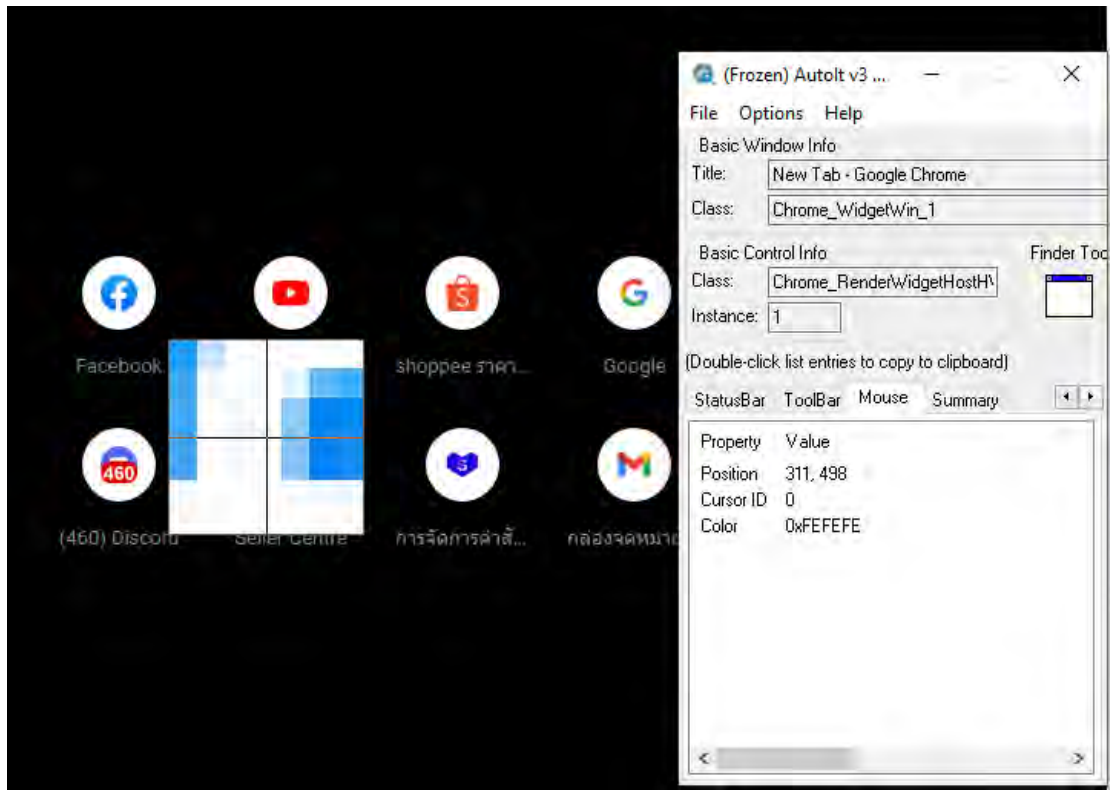
ตัวอย่างสิ่งที่โปรแกรม Autoit สามารถทำได้นั้น เช่น การทำงานแบบเดิมๆซ้ำๆ เปิดโปรแกรม เข้าสู่ระบบ Login พิมพ์ข้อความ ตรวจสอบหน้าต่างโปรแกรม เช็ค่าพิกัด เช็คสี การตั้งเงื่อนไข เพื่อเข้าสู่คำสั่งถัดไป โดยสิ่งที่จะทำนี้อาจเรียกว่า ai หรือ Bot

#### ตัวอย่างการหาค่าสีและตำแหน่งของรูปภาพของ Autoit

นอกจากตัว Autoit ยังเป็นโปรแกรมที่เอาไว้เขียนโปรแกรมเกี่ยวกับการทำ ai หรือ bot แล้ว ยังมีตัวของ Autoit V3 ที่เอาไว้หาค่าสีและตำแหน่งของสิ่งต่างๆที่เราต้องการหาอีกตัวและยังมีฟังก์ชันการทำงานในการหาชื่อตัวโปรแกรมที่เราใช้เพื่อที่เราจะได้นำไปเขียนโค้ดได้

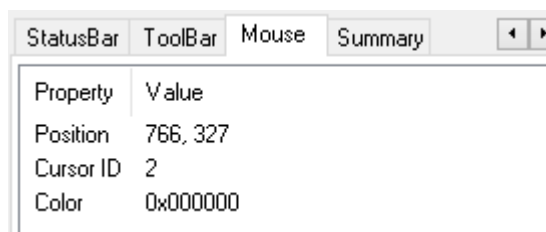
## ตัวอย่างการหาค่าสีและตำแหน่งด้วย Autoit V3

เปิดโปรแกรม Autoit V3 ขึ้นมาแล้วนำ Finder Tool ลากไปยังตำแหน่งที่ต้องการหาในที่นี่จะหาตำแหน่งของโลโก้ facebook



รูปที่ 2.7 การหาค่าสีและตำแหน่งของรูปภาพ

เลื่อนเมนูไปที่ Mouse จะมีตำแหน่ง Position และมีค่า Value 766,327 คือค่าตำแหน่งของโลโก้ facebook และมีค่า Color Value 0x000000 คือค่าสีของโลโก้ facebook โดยทั้ง 2 ตำแหน่งนี้จะถูกนำมาใช้ในการเขียน ai เพื่อนหารูปภาพที่เราต้องการให้โปรแกรมจับ



รูปที่ 2.8 ค่าสีและตำแหน่งของรูปภาพ





รูปที่ 2.9 photoshop

## 2.4 Photoshop

Photoshop เป็นโปรแกรมที่มีชื่อเสียงที่เอาไว้ตกแต่งภาพ หลายคนคงรู้จักกับโปรแกรม Photoshop ที่ใช้สำหรับตกแต่งภาพถ่ายและภาพกราฟฟิก ได้อย่างมีประสิทธิภาพ ไม่ว่าจะเป็นงานด้านสิ่งพิมพ์ นิตยสาร และงานด้านมัลติมีเดีย อีกทั้งยังสามารถ retouching ตกแต่งภาพและสร้างภาพ ซึ่งกำลังเป็นที่นิยมสูงมาก

### ความสามารถพื้นฐานของ Adobe photoshop

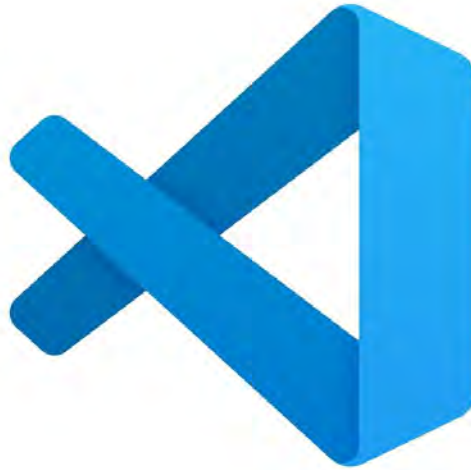
1. ตกแต่งหรือแก้ไขรูปภาพได้
2. ตัดต่อภาพบางส่วน หรือที่เรียกว่า crop ภาพ
3. เปลี่ยนแปลงสีของภาพ จากสีหนึ่งเป็นอีกสีหนึ่งได้
5. มีการแบ่งชั้นของภาพเป็น Layer สามารถเคลื่อนย้ายภาพได้เป็นอิสระต่อกัน
6. การทำ cloning ภาพ หรือการทำภาพซ้ำในรูปภาพเดียวกัน

### ข้อดีของ โปรแกรม Adobe Photoshop

1. ใช้งานได้ง่ายสำหรับมือใหม่
2. แก้ไขตกแต่งรูปภาพได้หลากหลายแบบ

### ข้อเสียของ โปรแกรม Adobe Photoshop

1. มีราคาแพง
2. กินทรัพยากรเครื่องสูง



รูปที่ 2.10 visual studio code

## 2.5 Visual Studio Code

Visual Studio Code หรือ VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขโค้ด และปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ที่ต้องการความเป็นมืออาชีพ

ซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานโปรแกรมข้ามแพลตฟอร์ม Visual Studio Code รองรับการใช้งานทั้งบนตัว Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาย

### ข้อดีของ Visual Studio Code

1. Meet IntelliSense รองรับการใช้สีเพื่อให้อ่านโค้ดง่ายขึ้น (Syntax Highlighting) รวมถึงการคาดเดาคำที่ถึง Dev กำลังจะพิมพ์ (Autocomplete)
2. Debugging รองรับการ Debug โค้ดภายในตัวโปรแกรมสามารถ Launch โปรเจกขึ้นมาแล้ว debug ด้วย breakpoint, call stacks และที่สำคัญมี Command/Console Prompt ภายในตัวอีกด้วย
3. Git นี่ก็คือข้อดีที่ถือว่าเบ็ดเสร็จได้ในโปรแกรมเดียวกัน มี Version Control ภายในตัวเองเลย จากที่ผมดูก็ถือว่าครบเครื่องเรื่อง git เลยทีเดียว





รูปที่ 2.11 โปรแกรม Nox

## 2.6 โปรแกรม Nox

ก่อตั้งขึ้นโดยทีมนักพัฒนาที่อาศัยอยู่ในฮ่องกง มุ่งเน้นไปที่โปรแกรมจำลองแอปมือถือต่างๆ เป็นหลัก และได้รับการต้อนรับจากผู้ใช้งานทั่วโลกเป็นอย่างดี มีผู้ใช้งานกว่า 150 ล้านคนใน 150 ประเทศ เปิดให้บริการมากกว่า 20 ภาษา และยังสามารถเตรียมพร้อมแก้ปัญหาระบบดิจิทัลให้ดีที่สุดสำหรับผู้ใช้งาน และสร้างการเชื่อมต่อกับ Android Windows และ Mac การกิจของเราคือ เปิดแพลตฟอร์มที่จะผสมผสานการโฆษณาแบบดิจิทัล และการโฆษณาแบบดั้งเดิมเข้าด้วยกัน และการเผยแพร่ซอฟต์แวร์ระบบ Android

Nox App Player คือ โปรแกรมจำลองระบบปฏิบัติการ Android ซึ่งเป็นระบบปฏิบัติการบนโทรศัพท์มือถือ ให้ขึ้นมาอยู่บนคอมพิวเตอร์ PC ความพิเศษ คือ ปล่อยให้โหลดใช้งานได้แบบฟรีๆ ในการนำมาใช้งานจริง ปราศจากปัญหาใดๆ เลย เนื่องจากทางผู้พัฒนา Nox App Player ใฝ่ใจในทุกรายละเอียด ใฝ่เข้ามาครบ ไม่ว่าจะเป็นเรื่องของ Hard Ware ซึ่งได้รับการปรับแต่ง ให้รองรับการใช้งานในหน่วยประมวลผล X86 หรือ แม้แต่หน่วยประมวลผลจาก AMD ก็สามารทำได้ ซึ่ง Nox App Player ก็ทำงานได้ดีเช่นเดียวกัน

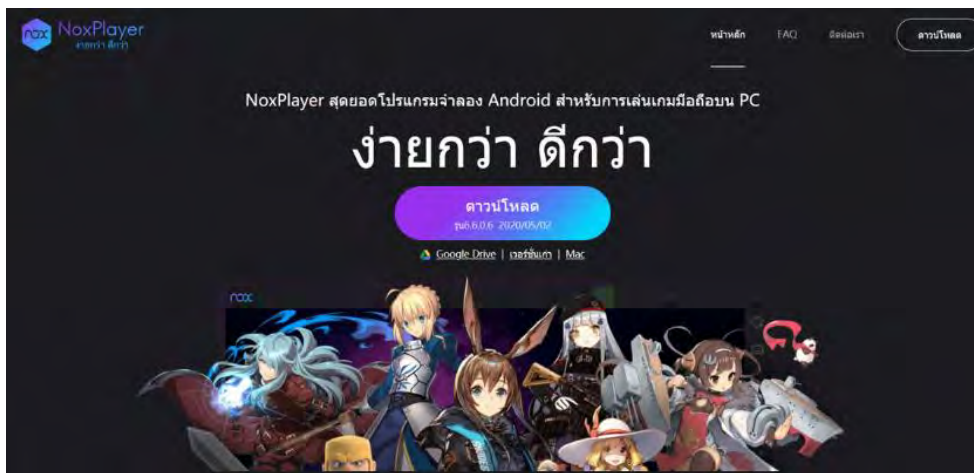
### คุณสมบัติของโปรแกรม Nox App Player

1. มีหน้าต่างที่เรียบง่ายไม่ยุ่งยาก
2. มีการตั้งค่า Keyboard สามารถทำให้ควบคุมการเล่นเกมนได้ง่าย เหมาะทั้งการใช้งานและเล่นเกม
3. ระบบเสถียร โหลดสิ้น แม้จะเล่นเกมกราฟฟิกสูงและเกมสเปคสูง
4. พื้นฐานมาจากระบบ Android 4.2

5. รองรับภาษาไทยอันนี้สำคัญ และสนับสนุนหลากหลายภาษา
6. สนับสนุน Application และเกมทุกรูปแบบ
7. สามารถแคปภาพหน้าจอ ( Screenshot ) ที่เทพไปกว่านั้นสามารถบันทึก VDO หน้าจอได้อีกด้วย

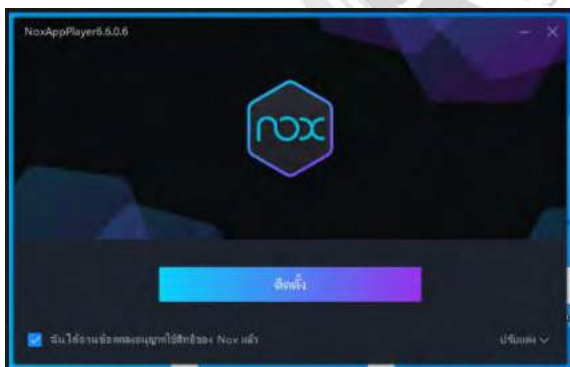
### ขั้นตอนการดาวน์โหลดและการติดตั้ง

เข้าไปที่เว็บ <https://th.bignox.com/> เพื่อดาวน์โหลดโปรแกรม

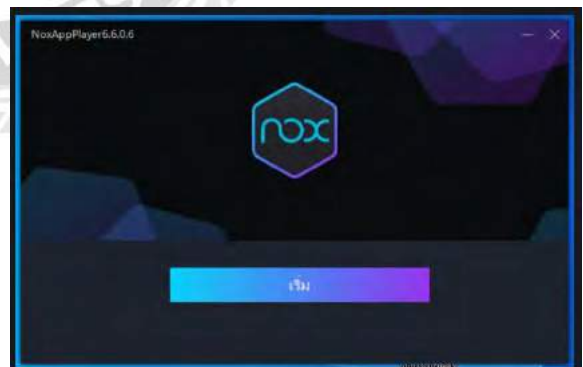


รูปที่ 2.12 เว็บไซต์ดาวน์โหลด Nox

โหลดเสร็จเปิดโปรแกรมขึ้นมา และทำการกดติดตั้งรอนจนติดตั้งเสร็จ 100% ก็สามารถใช้งานได้เลย



รูปที่ 2.13 ติดตั้ง Nox



รูปที่ 2.14 ติดตั้ง Nox สำเร็จ

## การใช้งาน Nox

1. มีคู่มือสอนการใช้งานเบื้องต้น ตัวโปรแกรมจะเป็นการจำลองประเภทอิมูเลเตอร์ (Emulator) ของระบบปฏิบัติการ Android Kitkat 4.4.2 ของเรามาเครื่องหนึ่ง โดยไม่จำเป็นต้องใช้ซิมและฟังก์ชันโทรออกแต่อย่างใด ส่วนความแรงก็ขึ้นอยู่กับคอมพิวเตอร์ของเราอีกที
2. ซึ่งถึงแม้ว่าจะเป็นคอมพิวเตอร์ที่ไม่ได้เป็นระบบสัมผัส แต่การควบคุมง่ายไม่ได้ต่างอะไรจาก Smartphone เลยแม้แต่น้อย (แถมยังดูเหมือนง่ายกว่าด้วย) พวกปุ่มบังคับจะถูกตั้งมาให้อัตโนมัติสำหรับเกมยอดเยี่ยม แต่หากเราไม่ถนัดก็สามารถตั้งค่าเพิ่มเติมได้ ส่วนจะหมุนเครื่องหรือต้องใช้เซ็นเซอร์อะไร ตัวโปรแกรมสามารถจำลองจัดให้ได้หมด
3. และเนื่องจากเป็นโปรแกรมจำลองจริง ๆ ดังนั้นจึงจำเป็นที่จะต้องใช้เหมือน Smartphone แทบทุกอย่าง ตั้งแต่การดาวน์โหลดแอปพลิเคชันผ่านทาง Play Store และจำเป็นต้องใช้ Google Account (Gmail) ในการเชื่อมต่อ ส่วนพวกไฟล์อะไรก็สามารถโอนถ่ายหากันได้ผ่านคอมพิวเตอร์โดยตรง และถ้าต้องการ Root ก็ทำได้ทันที
4. ไม่สามารถยกคอมพิวเตอร์ของเราไปไหนมาไหนสะดวกแบบมือถือ ดังนั้นพวกเกมหรือแอปพลิเคชันที่ต้องใช้ Location เราก็สามารถจำลองบน Google Maps ได้โดยตรง นอกจากนี้ยังสามารถใช้งานมัลติทาสกิ้ง เพื่อเปิดใช้งานหลายหน้าจอ เอาไว้สำหรับพวกเกมที่ต้องการปาร์ตี้แล้วเพื่อนไม่มีเวลา แต่หากใครอยากบันทึกวิดีโอก็สามารถทำได้เช่นเดียวกัน



รูปที่ 2.15 หน้าต่างโปรแกรม Nox

## 2.7 Package/Module



รูปที่ 2.16 Package/Module Numpy

### 2.7.1 Numpy

NumPy เป็นชื่อของ library ที่ใช้ในการคำนวณทางคณิตศาสตร์ในภาษา Python ซึ่งภายในถูกเขียนด้วยภาษา C จึงทำงานได้เร็วและมีประสิทธิภาพ โดย NumPy มีความสามารถในการจัดการกับอาร์เรย์หลายมิติและข้อมูลแบบเมทริกซ์

#### การติดตั้ง

สามารถติดตั้ง NumPy ได้ง่ายผ่าน package installer ของ Python ด้วยคำสั่ง

```
pip install numpy
```

รูปที่ 2.17 คำสั่งติดตั้ง Numpy

#### การนำ Numpy มาใช้

เวลาจะใช้งานก็ import เข้ามาใช้งานได้เหมือนตัวอย่างด้านล่างนี้เลย

```
import numpy as np
a = np.array([1,2,3])
print(a)          # [1 2 3]
print(type(a))   # <class 'numpy.ndarray'>
```

รูปที่ 2.18 คำสั่งการนำ Numpy มาใช้

#### ประโยชน์ของ Numpy

1. การสร้าง Array
2. สร้าง Array ที่มีข้อมูลภายใน
3. ทำให้ค่าใน Array อยู่ในช่วงที่กำหนด
4. ค้นหาตำแหน่งของข้อมูลที่ผ่านมาเงื่อนไข



รูปที่ 2.19 Package/Module Pillow

### 2.7.2 Pillow

โมดูล Pillow ซึ่งเป็นด้าน image processing and graphics capabilities หรือโมดูลจัดการและประมวลผลรูปภาพบน Python ซึ่งใน Python มีโมดูลด้านนี้ที่ชื่อว่า Python Imaging Library (PIL) ซึ่งรองรับแต่ตัว Python 2 ในเวลานี้ จึงมีคนได้นำ Fork PIL มาพัฒนาเป็นโมดูล Pillow ให้รองรับทั้ง Python 2 และ Python 3

#### การติดตั้ง

สามารถติดตั้ง Pillow ได้ง่ายผ่าน package installer ของ Python ด้วยคำสั่งด้านล่าง

```
pip install Pillow
```

รูปที่ 2.20 คำสั่งติดตั้ง Pillow

#### การนำ Pillow มาใช้

เวลาจะใช้งานก็ import เข้ามาใช้งานได้เหมือนตัวอย่างด้านล่างนี้เลย

```
>>> from PIL import Image
>>> im = Image.open("60_1.jpg")
>>> print(im.format, im.size, im.mode)
JPEG (450, 346) RGB
>>> im.show()
```

รูปที่ 2.21 คำสั่งการนำ Pillow มาใช้

#### โมดูล Pillow ทำอะไรได้บ้าง

1. จัดเก็บรูปภาพ (Image Archives)
2. ประมวลผลรูปภาพ (Image Processing) เช่น





## Pure python adb

รูปที่ 2.22 Package/Module Pure python adb

### 2.7.3 Pure python adb

pure-python-adb เป็นโมดูลใช้เพื่อสื่อสารกับเซิร์ฟเวอร์ adb เป็นหนึ่งในเครื่องมือ ช่วยให้ระบบสามารถสื่อสารกับอุปกรณ์ Android ที่เชื่อมต่อผ่าน USB หรือโปรแกรมจำลอง

#### การติดตั้ง

สามารถติดตั้ง Pure python adb ได้ง่ายผ่าน package installer ของ Python ด้วยคำสั่งด้านล่าง

```
$pip install -U pure-python-adb
```

รูปที่ 2.23 คำสั่งติดตั้ง Pure python adb

#### การนำ Pure python adb มาใช้

เวลาจะใช้งานก็ import เข้ามาใช้งานได้เลยตัวอย่างการเชื่อมต่อเซิร์ฟเวอร์กับ adb

```
from ppadb.client import Client as AdbClient
# Default is "127.0.0.1" and 5037
client = AdbClient(host="127.0.0.1", port=5037)
print(client.version())
```

รูปที่ 2.24 คำสั่งการนำ Module Pure python adb มาใช้

#### โมดูล Pure python adb ทำอะไรได้บ้าง

Pure python adb มีหน้าที่เชื่อมต่อกับเซิร์ฟเวอร์ adb และรับเวอร์ชัน



## Multiprocess

รูปที่ 2.25 Package/Module Multiprocess

### 2.7.4 Multiprocess

Multiprocess คือ การที่โปรเซสเซอร์ตั้งแต่ 1 ตัวขึ้นไปทำงานพร้อมๆ กันในเวลาเดียวกัน ซึ่งหมายถึง โปรเซสเซอร์ต่างๆ สามารถประมวลผลคำสั่งไปพร้อมๆ กันได้

ปกติเวลาที่รันโปรแกรมในไพธอนโปรแกรมจะทำตามคำสั่งตามลำดับที่ละขั้นตอนโดยตลอดโปรแกรมถือว่าเป็นงานงานเดียวที่รันใน CPU ตัวเดียว

ปกติแล้วคอมพิวเตอร์มีความสามารถที่จะทำงานหลายๆอย่างในเวลาเดียวกันได้อยู่แล้ว แต่ปัญหาคือหากเขียนโปรแกรมไพธอนด้วยวิธีทั่วไปยังวิ่งกี่รันได้ทีละงานในเวลาเดียว หากอยากรันหลายงานก็ต้องมาสั่งหลายๆครั้งเอง เช่นเปิดหน้าคอมมานด์ไลน์มาหลายๆอันแล้วรันไปทีละอัน แต่แบบนี้ต้องมาคอยเปิดหลายอัน อีกทั้งหากทำแบบนี้ก็จะเป็นการรันคนละโปรแกรม ไม่สามารถเอามาเชื่อมโยงกันได้โดยตรง

#### การติดตั้ง

สามารถติดตั้ง Multiprocess ได้ง่ายผ่าน package installer ของ Python ด้วยคำสั่งด้านล่าง

```
$ pip install multiprocess
```

รูปที่ 2.26 คำสั่งติดตั้ง Multiprocess

#### ข้อดีของระบบ Multiprocessing

ระบบหลายตัวประมวลผลนั้นทำให้เกิดประสิทธิภาพในการทำงานและประสิทธิผลที่เพิ่มขึ้น เพราะบางครั้งการทำงานของโปรแกรมอาจต้องการใช้ทรัพยากรอื่นๆ ถ้าหากใช้ตัวประมวลผลเดียวก็จะทำให้ต้องรอนกว่าตัวประมวลผลนั้นจะว่าง จึงสามารถใช้ทรัพยากรได้



# Auto-py-to-exe

รูปที่ 2.27 Package/Module Auto-py-to-exe

## 2.7.5 Auto-py-to-exe

เป็น Package/Module ที่ใช้สำหรับแปลงไฟล์ที่เราเขียนเสร็จแล้วให้กลายเป็น .exe เพื่อจะ  
ได้ทำการรันโปรแกรมได้สะดวกโดยไม่ต้องไปรันบน VSCode

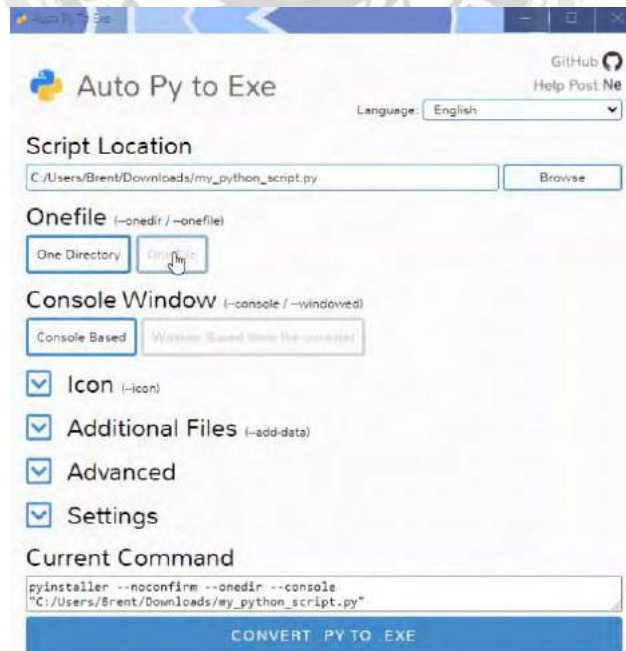
### การติดตั้ง

สามารถติดตั้ง Auto-py-to-exe ได้ง่ายผ่าน package installer ของ Python ด้วยคำสั่งด้านล่าง

```
$ pip install auto-py-to-exe
```

รูปที่ 2.28 คำสั่งติดตั้ง Auto-py-to-exe

### ตัวอย่างการนำไฟล์ไปแปลงเป็น .exe



รูปที่ 2.29 การแปลงไฟล์เป็น .exe





รูปที่ 2.30 ตัวเกมหม้อไฟแห่งความสุข

## 2.8 ตัวเกมหม้อไฟแห่งความสุข

ร้านหม้อไฟแห่งความสุข (My Hotpot Story) เกมจำลองการเป็นเจ้าของร้านหม้อไฟจำลอง ธุรกิจร้านอาหาร ที่กำลังได้รับความนิยมในบ้านเราพอสมควร โดยรูปแบบของเกมจะเป็นเกมจำลองว่าเรามาเปิดร้านอาหาร และให้เราจัดการบริหาร เก็บเงินขยายร้านและทำสิ่งต่างๆที่พัฒนาร้าน เกมจะเป็นเกมที่เล่นง่ายเข้าใจง่าย สามารถทำตามคำแนะนำไปได้เรื่อยๆ

ดาวน์โหลดเกม : ตัวเกมมีให้ดาวน์โหลดทั้งระบบ Android และ iOS (iPhone)

### คุณสมบัติของเกม

1. คุณสามารถเพลิดเพลินกับวิธีการทางธุรกิจที่หลากหลายได้อย่างอิสระ
2. สัมผัสความสนุกในการเปิดร้านหม้อไฟและเพลิดเพลินกับอาหารอร่อยทุกประเภท โดยเฉพาะอาหารจีน
3. ตกแต่งร้านและสร้างร้านอาหารหม้อไฟของคุณเอง



รูปที่ 2.31 ภาพตัวอย่างภายในเกม 1

## ภาพภายในเกม



รูปที่ 2.32 ภาพตัวอย่างภายในเกม 2



รูปที่ 2.33 ภาพตัวอย่างภายในเกม 3

## 2.9 Line notify



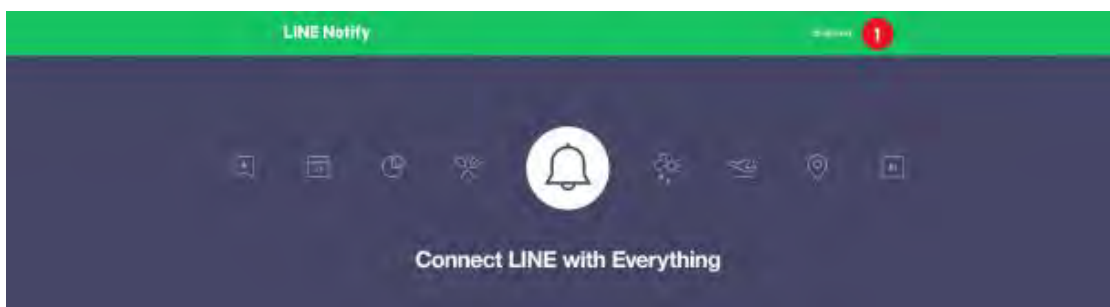
รูปที่ 2.34 linenotify

Line Notify คือ บริการที่ Line ได้เตรียมไว้ให้ในรูปแบบของ API ให้กับนักพัฒนานั้นสามารถนำไปใช้ต่อยอด พัฒนาโปรเจกต์ ที่มีความต้องการส่งข้อความในการแจ้งเตือนเข้าไปยังบัญชีส่วนตัวของเรา หรือกลุ่มได้

Line notify ใช้เพื่อแจ้งสถานะการออนไลน์ไปอีกระบบปลายทางได้ จึงทำให้สามารถส่งข้อความแจ้งเตือนจากบริการต่างๆ หรืออุปกรณ์ใดๆ ก็ตาม ที่สามารถเชื่อมต่อกับ internet และสามารถเชื่อมด้วย http post มายัง Account ของเราได้ ซึ่งการใช้งานโดยรวมของ Line notify จะมีรูปแบบดังนี้ คือ เริ่มแรกเลย ต้องไปสร้าง token ของ account ในระบบของ Line ก่อน จากนั้นเก็บ token นี้เอาไว้ แล้วเมื่อต้องการที่จะส่งข้อความแจ้งเตือนต่าง ๆ จะใช้ token นี้เพื่อส่งข้อความแจ้งเตือน ผ่านทาง http post

### 2.9.1 การออกโทเค็น line notify

1. ไปที่ <https://notify-bot.line.me/th/>



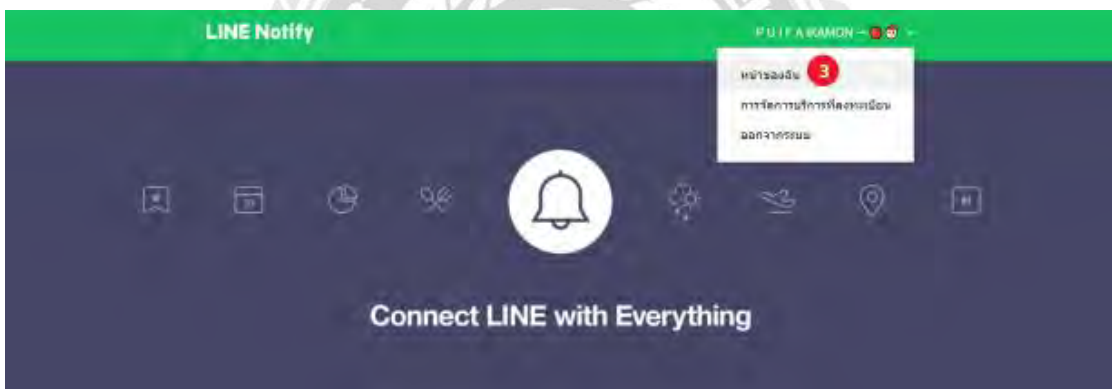
รูปที่ 2.35 ขั้นตอนที่ 1 ของการการออกโทเค็น line notify

- กรอก Email และ Password ของ Line เพื่อเข้าสู่ระบบ



รูปที่ 2.36 ขั้นตอนที่ 2 ของการการออกโทเค็น line notify

- เมื่อเข้าสู่ระบบแล้ว ให้คลิกชื่อ Line ที่มุมขวาบน คลิกหน้าของมัน



รูปที่ 2.37 ขั้นตอนที่ 3 ของการการออกโทเค็น line notify

- เข้ามาแล้วให้คลิกที่ปุ่ม ออก Token

### ออก Access Token (สำหรับผู้พัฒนา)

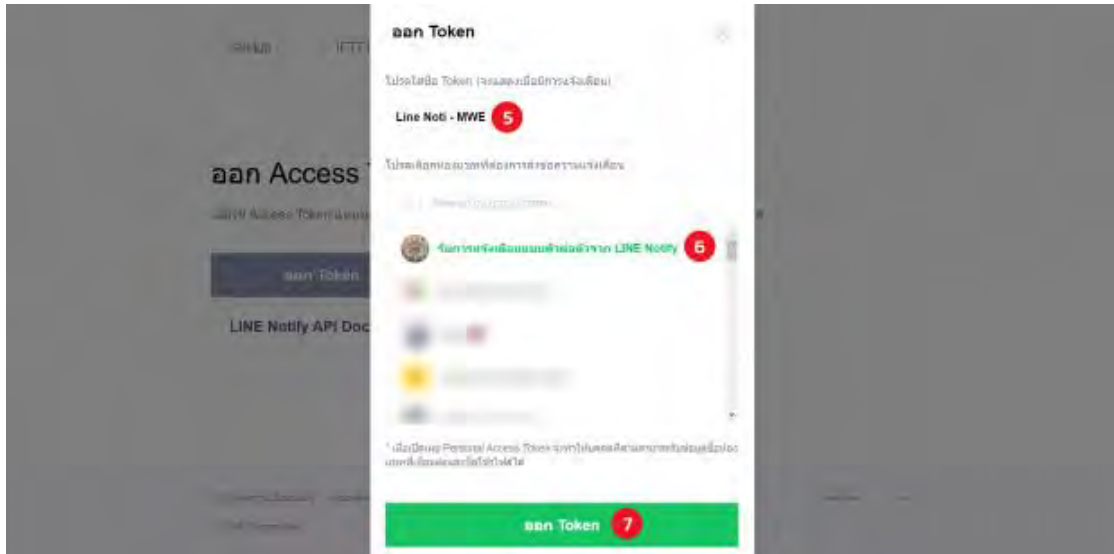
เมื่อได้ Access Token แบบบุคคล จะสามารถส่งคำขอร้องเพื่อใช้โต้ตอบกับแอปพลิเคชันบนเว็บไซต์



[LINE Notify API Document](#)

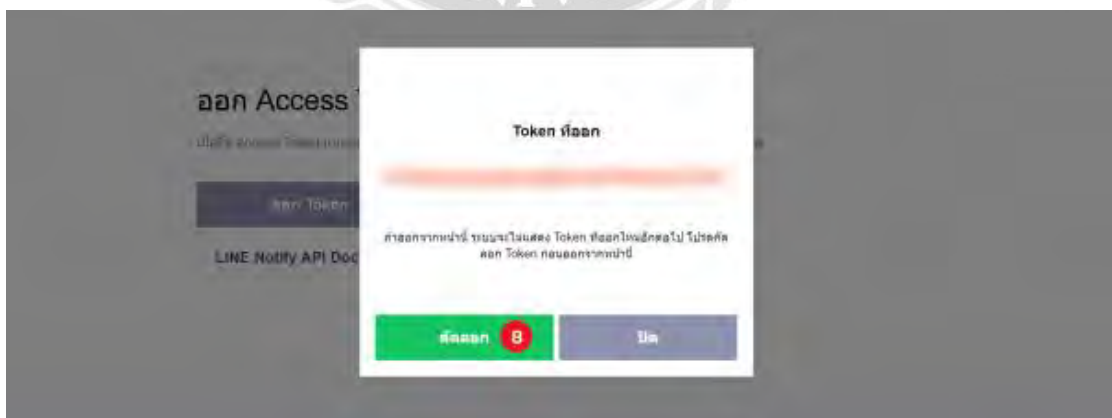
รูปที่ 2.38 ขั้นตอนที่ 4 ของการการออกโทเค็น line notify

5. กำหนดชื่อ Token เวลาแสดงแจ้งเตือน
6. เลือกแชทที่ต้องการส่งข้อความแจ้งเตือน จากตัวอย่างจะเป็นการแจ้งเตือน แบบส่วนตัว จาก Line Noti ในกรณีที่ร้านค้ามีการจัดการหลายคน ก็ให้เลือกการกลุ่มแชท เพิ่มเติมคือ เมื่อเลือกกลุ่มแชทและออก Token แล้ว ให้ดึง Line Noti เข้ากลุ่ม
7. กด คัดลอก Token เอาไว้ เพื่อนำไปเชื่อมต่อกับเว็บไซต์ของเรา



รูปที่ 2.39 ขั้นตอนที่ 5-6-7 ของการการออกโทเค็น line notify

8. กด คัดลอก Token เอาไว้ เพื่อนำไปเชื่อมต่อกับเว็บไซต์ของเรา จากนั้นกดปิด จะเห็นได้ว่าเราได้มีการเชื่อมต่อ Line Notify แล้ว



รูปที่ 2.40 ขั้นตอนที่ 8 ของการการออกโทเค็น line notify





รูปที่ 2.41 เชื่อมต่อ line notify เรียบร้อยแล้ว



## บทที่ 3

### การวิเคราะห์และการออกแบบ

ในบทนี้เป็นส่วนหนึ่งของการวิเคราะห์และออกแบบ โดยจะอธิบายถึงการออกแบบโครงสร้างของโปรแกรมบอท ลำดับการทำงานบอท ของโครงการที่ได้ทำ โดยมีรายละเอียดในการวิเคราะห์และออกแบบดังต่อไปนี้

#### 3.1 การออกแบบระบบ

ในโครงการนี้จะทำการสร้างโปรแกรมบอทในการเล่นเกมอัตโนมัติ โดยใช้ภาษาไพทอน ในการเขียนโปรแกรมร่วมกับโมดูลอีกหลายตัว เช่น โอเพนซีวีในการหาภาพ Pure python adb ในการเชื่อมต่ออุปกรณ์จำลอง และคำสั่งอโตคลิกเพื่อให้เมาส์ทำการคลิกตำแหน่งบนหน้าจอเราเอง โดยอัตโนมัติเมื่อเจอภาพหรือสีที่ถูกต้อง

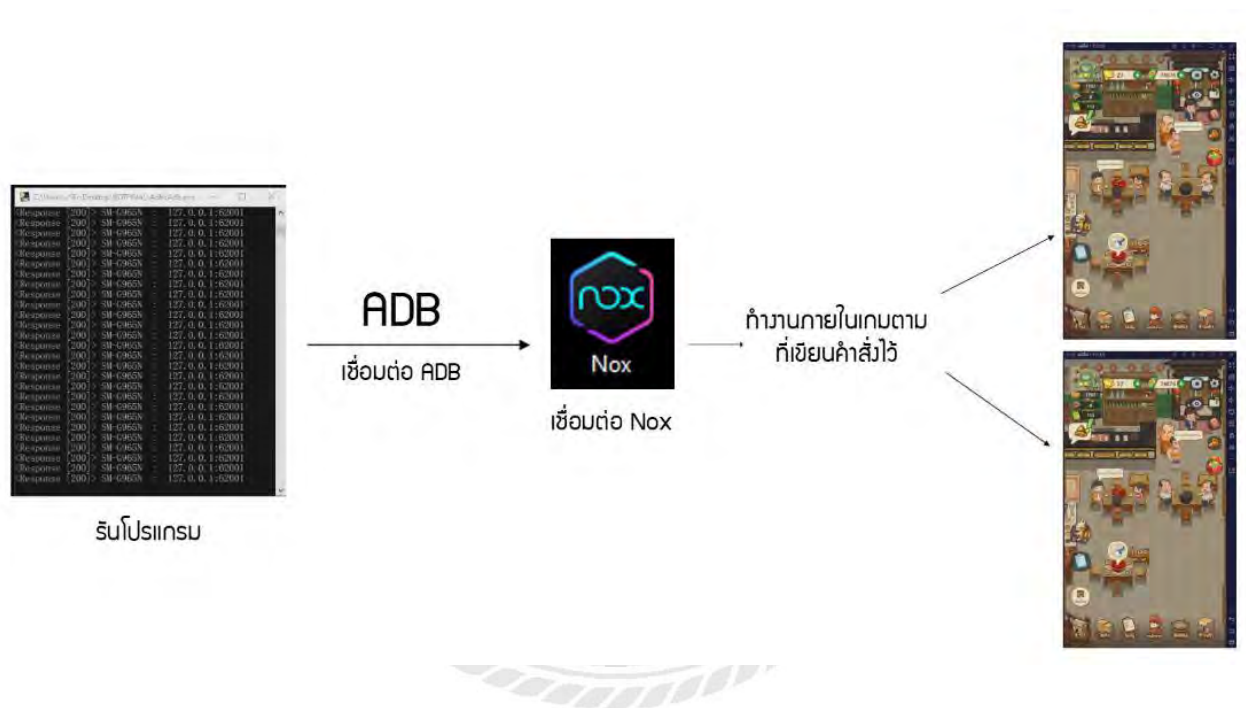
##### 3.1.1 โมดูลหรือไลบรารีและภาษาที่ใช้ในการออกแบบระบบซอฟต์แวร์

ในการออกแบบระบบซอฟต์แวร์จะมีดังต่อไปนี้

1. ภาษาไพทอนใช้ในการเขียนโปรแกรมทั้งหมด
2. NumPy ใช้สำหรับการคำนวณทางคณิตศาสตร์
3. Open cv ใช้สำหรับจัดการกับรูปภาพทั้งหมด
4. pure python adb ใช้ในการเชื่อมต่อกับอุปกรณ์จำลอง
5. Pillow ใช้ในการจัดการและการประมวลผลรูปภาพ
6. multi process ใช้ในการให้บอทสามารถทำงานได้หลายหน้าจอในเวลาเดียวกัน โดยแต่ละหน้าจอจะแยกการทำงานของตัวเองและบอทจะไม่ทำงานมั่ว
7. Auto py to exe ใช้ในการแปลงไฟล์ที่เขียนเสร็จแล้วเป็นนามสกุล .exe

### 3.2 ภาพรวมการทำงานของระบบตอนรันโปรแกรม

ตัวบอทเมื่อเขียนเสร็จและมีการรันขึ้นมาจะทำการเชื่อมต่อกับ โปรแกรมจำลองที่เอาไว้ในเปิดเกม และตัว ADB ที่เป็นเหมือนตัวกลางให้โปรแกรมบอทคุยกับคอมพิวเตอร์รู้เรื่อง และตัวบอทจะทำงานตามคำสั่งที่มีการเขียนไว้หากมีหลายจอบอทก็จะแยกและรู้เองว่าจอบไหนต้องทำอะไร และบอทจะไม่แย่งเมาส์จากผู้ใช้งาน โดยบอทจะทำการจำลองเมาส์ของมันเองขึ้นมาใหม่โดยไม่ยุ่งกับเมาส์ของผู้ใช้งานจริง



รูปที่ 3.1 ภาพรวมการทำงานของระบบตอนรันโปรแกรม





จากรูป 3.2 สามารถอธิบายการทำงานของบอทได้ดังนี้ คือ เมื่อมีการกดเริ่มทำงานของตัวบอท ตัวบอทจะกดเข้าเกม และ กดยืนยันข้อตกลงในการเล่นเกมน จากนั้นทำการกดเข้าไปในร้านอาหารและเช็คว่ายู่ในร้านหรือไหมหากอยู่ให้ทำตามคำสั่งต่อไปนี้คือ 1.กดรูปออเดิร์ฟ 2.กดรูปไม้ปัดฝุ่น 3.กดรูปเงิน-ยืนยัน 4.กดรูปแมวกวักหากมีการเรืองแสงและรับของภายในทั้งหมด หากกดทั้งหมดเสร็จแล้วหรือไม่เจอรูปทั้ง 4 ข้อ ที่กล่าวมาเลยให้ทำการสไลด์ไปหน้าห้องครัวและเช็คแล้วเช็คว่ายู่หน้าห้องครัวหรือไม่หากอยู่ให้ทำตามคำสั่งต่อไปนี้คือ 1.กดรูปปลาซี 2.กดรูปจานอาหาร 3.กดรูปอ่างล้างจานเพื่อทำการล้างจาน หากกดทั้งหมดเสร็จแล้วหรือไม่เจอรูปทั้ง 3 ข้อ ที่กล่าวมาเลยให้ทำการสไลด์กลับไปหน้าร้านอาหารเพื่อนำคำสั่งเดิมวนซ้ำเรื่อยๆ และให้ทำการรีเซ็ตเกมทุกๆ 1 ชั่วโมงไม่ต่อนั้นจะอยู่หน้าไหนก็ตาม และเริ่มกดเข้าเกมใหม่และทำกระบวนการซ้ำไปเรื่อยๆ

### 3.4 คำสั่งที่จำเป็นในการสร้างบอท

#### 3.4.1 การทำงานหลายจอแยกกัน

จาก code ด้านล่าง บรรทัดที่ 1 คือการกำหนดให้บอทมีการทำงานแต่ละหน้าต่างแยกกัน โดยต้องเขียน code ไว้ได้ main code การทำงานแยกหน้าต่างถึงจะใช้งานได้ บรรทัดที่ 2 เมื่อเราแปลงไฟล์เป็น .exe บางครั้งถ้า code ที่ไม่ได้อยู่ภายใต้ main อาจเกิด error ได้เลยมีการเขียนดักไว้กันเกิดการ error

```
บรรทัดที่ 1   if __name__ == '__main__':
บรรทัดที่ 2   freeze_support()
```

#### 3.4.2 การเชื่อมต่อ adb

จาก code ด้านล่าง บรรทัดที่ 1 คือ การแสดงคำว่า 'ADB Connect' เพื่อเช็คว่ายากตอนนี้กำลังจะมีการเชื่อมต่อกับตัว adb บรรทัดที่ 2 เป็นคำสั่งสำหรับการเชื่อมต่อกับ adb

```
บรรทัดที่ 1   print( 'ADB Connect' )
บรรทัดที่ 2   tool.start_adb()
```

### 3.4.3 การเชื่อมต่อ อีมูเรเตอร์

จาก code ด้านล่าง บรรทัดที่ 1 คือ การแสดงคำว่า 'Emurator Connect' เพื่อเช็คว่าการเชื่อมต่อจะมีการเชื่อมต่อกับตัว Emurator บรรทัดที่ 2 เป็นคำสั่งสำหรับการเชื่อมต่อกับ Emurator โดยจะมีการประกาศตัวแปร devices ไว้เพื่อนำไปใช้งานต่อ และเลข 5037 คือ เลข port ของตัว Emurator

```
บรรทัดที่ 1    print( ' Emurator Connect' )
บรรทัดที่ 2    devices = tool.connect_emu( 5037 )
```

### 3.4.4 สั่งให้บอททำงาน

จาก code ด้านล่าง บรรทัดที่ 1 คือ การเอาตัวแปร device มาเก็บค่า list จากหัวข้อ 3.4.3 บรรทัดที่ 2 เมื่อรันบอทจะมีการแจ้งบอกว่า 'Bot Start : ' พร้อมบอก ip และชื่อเครื่องของเครื่องนั้นที่เราได้มีการรันบอทอยู่ บรรทัดที่ 3 เป็นคำสั่งสำหรับให้บอทเริ่มทำงานหลายหน้าจอ

```
บรรทัดที่ 1    for device in devices :
บรรทัดที่ 2    print( 'Bot Start : ' , device.shell("getprop
                ro.product.model").strip() , ' : ' , device.serial )
บรรทัดที่ 3    Process( target=start_bot_1 , args=(device,) ).start()
```

### 3.4.5 การประกาศฟังก์ชันสำหรับเขียนบอท

จาก code ด้านล่าง บรรทัดที่ 1 คือ การประกาศฟังก์ชันสำหรับเขียนบอทมีชื่อว่า start\_bot\_1

```
บรรทัดที่ 1    def start_bot_1 ( device ):
```

### 3.4.6 ชื่อแอปพลิเคชัน

จาก code ด้านล่าง บรรทัดที่ 1 คือ การหาชื่อแอปเก็บไว้ในตัวแปร app เพื่อนำค่าไปใช้ในการเปิดตัวเกมและรีเซ็ตตัวแอปพลิเคชัน

```
บรรทัดที่ 1    app = 'com.lxqd.myhotpotstroy.abroad'
```

### 3.4.7 รีเซ็ตพร้อมเปิดตัวเกม

จาก code ด้านล่าง บรรทัดที่ 1 คือ การเปิดตัวเกมโดยจะแยกเปิดทีละหน้าจอและรีเซ็ตตัวแอปพลิเคชัน

```
บรรทัดที่ 1    print( 'Open Game : ', device.shell("getprop
                ro.product.model").strip() , ': ', device.serial )
                tool.reset_app( device , app)
```

### 3.4.8 การกำหนดเกณฑ์สภาพ

จาก code ด้านล่าง บรรทัดที่ 1 คือ การกำหนดความเหมือนของรูปภาพเนื่องจากภาพอาจมีเอฟเฟ็คเรืองแสงเลยทำให้ไม่เหมือนภาพที่แคปไว้ 100 เปอร์เซ็นเลยต้องมีการกำหนดเปอร์เซ็นความเหมือนที่ต่ำลงมาเพื่อให้โปรแกรมสามารถหาภาพได้เจอ

```
บรรทัดที่ 1    threshold_set = 0.90
```

### 3.4.9 การแสดงรูปภาพ

จาก code ด้านล่าง บรรทัดที่ 1 คือ การแสดงรูปภาพที่บอทกำลังทำงานอยู่เพื่อตรวจสอบว่าบอทมีการจับเจอรูปภาพนั้นจริงๆและง่ายต่อการแก้ไขบอท

```
บรรทัดที่ 1    showimg = True
```

### 3.4.10 จำนวนการคลิก

จาก code ด้านล่าง บรรทัดที่ 1 คือ การกำหนดจำนวนครั้งในการคลิก

```
บรรทัดที่ 1    lv_click = 1
```

### 3.4.11 เฌดสี

จาก code ด้านล่าง บรรทัดที่ 1 คือ การกำหนดความเหมือนของเฌดสีเนื่องจากภาพอาจมีเอฟเฟ็คเรืองแสงเลยทำให้ไม่เหมือนภาพที่บันทึกไว้ 100 เปอร์เซ็นเลยต้องมีการกำหนดความเหมือนให้เพี้ยนไปให้เกิน 5 ระดับ เฌดสี เพื่อให้โปรแกรมสามารถหาภาพได้

```
บรรทัดที่ 1    shade = 5
```

### 3.4.12 จับเวลา

จาก code ด้านล่าง บรรทัดที่ 1 คือ การกำหนดเวลาดีเลย์ก่อนจะถึงส่งต่อไปเป็นหน่วยวินาที

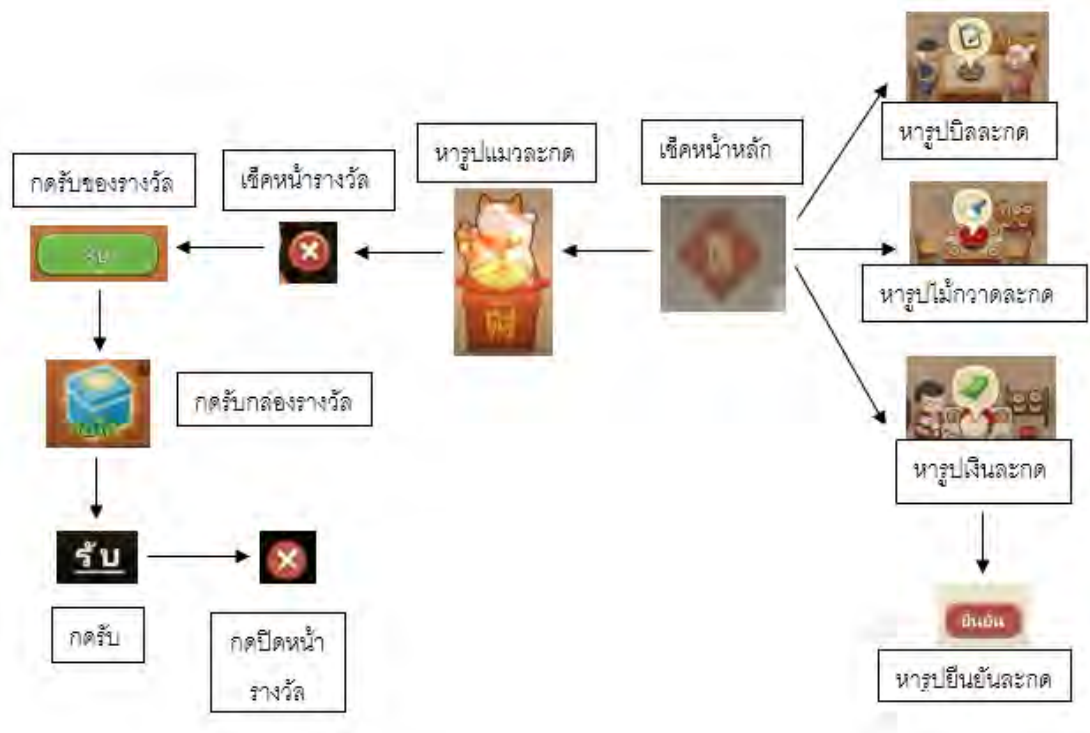
```
บรรทัดที่ 1    time_count = autoclick.timeset( 3600 )
```

## 3.5 การออกแบบการทำงานของบอทในแต่ละส่วน

การสร้างบอทในโครงการนี้การเขียน code ให้บอททำงานจะแบ่งออกเป็น 6 ส่วน คือ

1. การทำงานในหน้าหลัก
2. การทำงานหน้าครัว
3. การรีเซตเกมพร้อมกดเข้าเกม
4. การทำงานนอกเหนือจากหน้าหลักและหน้าครัว
5. การทำให้บอททำงานหลายหน้าจอ และไม่แย่งเมาส์ผู้เล่น
6. การทำให้บอทส่งข้อมูลข้อความและรูปภาพแจ้งเตือนไปทางไลน์

## 3.5.1 การทำงานในหน้าหลัก



รูปที่ 3.3 การทำงานในหน้าหลัก

จากรูปที่ 3.3 การทำงานบอทจะเริ่มจากการตรวจภาพภาษาจีนที่อยู่ในกรอบแดงเพื่อเช็คคำตอบตอนนี้บอทกำลังอยู่ในหน้าหลักจริงๆและค่อยทำคำสั่งต่อไปนี้

1. บอทจะทำการหารูปปิดหากเจอรูปปิดบอทจะทำการกด
2. บอทจะทำการหารูปไม้กวาดหากเจอรูปไม้กวาดบอทจะทำการกด
3. บอทจะทำการหารูปเงินหากเจอรูปเงินบอทจะทำการกดและหลังจากกดรูปเงินจะแสดงรูปยืนยันขึ้นมาบอทก็จะทำการตรวจหาอีกทีและกดยืนยันหากไม่เจอรูปเงินก็จะไม่สามารถกดรูปยืนยันได้
4. บอทจะทำการหารูปแมวหากเจอรูปแมวบอทจะทำการกด และเช็ครูปกากบาทเพื่อให้แน่ใจว่ามีการอยู่หน้ารับรางวัลแล้วเพื่อไม่ให้บอทนั้นเกิดการดำเนินงานที่มั่ว และบอทจะทำการหารูปคำว่ารับที่อยู่ในกรอบสีเขียวและกดและกดรูปกล่องกล่องสีฟ้าอีกทีที่พอกดในกล่องจะมีรางวัลให้รับโดยบอมก็จะกดคำว่ารับและกดกากบาทออกจากหน้ารับรางวัล

โดยการทำงานของบอททั้ง 4 ขั้นตอนนี้หากบอทจับเจอภาพไหนก่อนหรือมีภาพไหนขึ้นมาก่อนบอทก็จะทำงานในรูปรูปนั้นก่อน

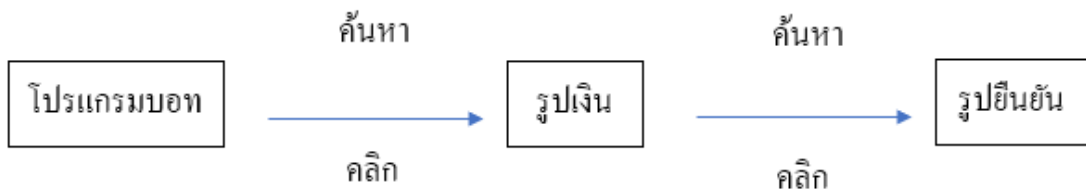
## 3.5.2 การทำงานของบอทในหน้าหลัก



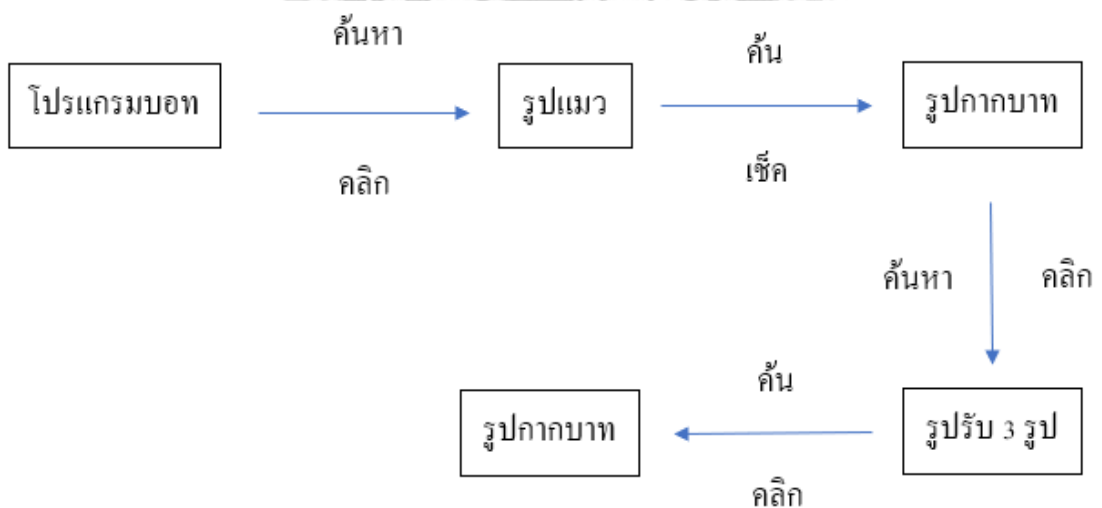
รูปที่ 3.4 บล็อกไดอะแกรมการทำงานของบอทในหน้าหลักหารูปบิล



รูปที่ 3.5 บล็อกไดอะแกรมการทำงานของบอทในหน้าหลักหารูปไม้กวาด



รูปที่ 3.6 บล็อกไดอะแกรมการทำงานของบอทในหน้าหลักหารูปเงินและรูปยืนยัน



รูปที่ 3.7 บล็อกไดอะแกรมการทำงานของบอทในหน้าหลักหารูปแมวและรับรางวัล

## 3.5.2 การทำงานในหน้าครัว



รูปที่ 3.8 การเลื่อนไปหน้าครัวและกลับมาหน้าหลัก

จากรูปที่ 3.8 จะทำงานต่อจากรูป 3.3 พอทำงานหน้าหลักเสร็จหมดทุกอย่างและไม่มี การทำงานอะไรต่อบทจะทำการสไลด์ไปหน้าครัวโดยเริ่มคลิกค้างไว้ที่ตำแหน่งที่ 1 ของหน้าหลักและทำการลากเมาส์ไปทางขวาลากไปที่ตำแหน่งที่ 2 ของหน้าหลักหน้าจอก็จะเปลี่ยนไปเป็นหน้าครัว





รูปที่ 3.9 การทำงานในหน้าครัว

จากรูป 3.9 เมื่อมีการสไลด์จากหน้าหลักไปหน้าครัวบอทจะทำการเช็ครูปหม้อเพื่อเป็นการยืนยันว่าตอนนี้อยู่น้ำคร่าแล้วบอทก็จะเริ่มทำงานคือ

1. ค้นหารูปหม้อถ้าเจอบอทก็จะทำการคลิก
2. ค้นหารูปฝาซีถ้าเจอบอทก็จะทำการคลิก
3. ค้นหารูปจานถ้าเจอบอทก็จะทำการคลิก

โดยบอทหากเจอรูปภาพไหนก่อนก็จะทำงานรูปนั้นก่อนและหากบอทไม่เจอรูปภาพทั้ง 3 รูปภาพเลยบอทก็จะทำการสไลด์กลับไปหน้าหลัก ตามรูปที่ 3.4

## 3.5.3 การรีเซ็ตเกม



รูปที่ 3.10 การรีเซ็ตเกม

```

บรรทัดที่ 1   time_count = autoclick.timeset( 3600 )
บรรทัดที่ 2   If autoclick.timeout( time_count ) :
บรรทัดที่ 3   tool.reset_app( device , app)
บรรทัดที่ 4   time_count = autoclick.timeset( 3600)

```

จากรูปที่ 3.10 คือการรีเซ็ตเกมเมื่อเวลาครบ 1 ชม และรูปที่ 3.10 เมื่อทำการรีเซ็ตแล้วเกมจะดึงมาหน้าโปรแกรมตอนเข้าเกมตามรูปการรีเซ็ตเกม

บรรทัดที่ 1 คือการกำหนดเวลารีเซ็ตหน่วยเป็นวินาที

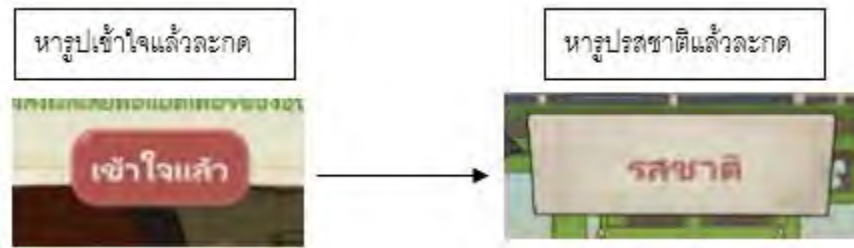
บรรทัดที่ 2 คือการตรวจสอบว่าถึงเวลาที่กำหนดแล้วไว้แล้วให้ไปทำบรรทัดที่ 3 ต่อ

บรรทัดที่ 3 คือการรีเซ็ตแอป โดยมีค่า device และชื่อ app เขียนกำกับไว้ว่าให้ทำงานกับ

แอปไหนพร้อมทั้งกดแอปนั้นด้วยเพื่อเข้าไปในแอป

บรรทัดที่ 4 คือการตั้งเวลาให้ทำการจับเวลาใหม่อีกครั้งเมื่อรีเซ็ตเสร็จแล้ว

### 3.5.4 การทำงานนอกเหนือจากหน้าหลักและหน้าครัว



รูปที่ 3.11 การทำงานในส่วนยืนยันข้อตกลงและกดเข้าร้าน

จากรูป 3.11 จะต่อเนื่องมาจากรูป 3.10 คือ เมื่อมีการรีเซ็ตเกมและกดเข้าเกมเรียบร้อยแล้ว รูปที่ 3.11 คือเมื่อกดเข้ามาตัวเกมจะมีให้ยืนยันข้อตกลงต่างๆในการจะเล่นเกมบอทก็จะทำการตรวจหาภาพที่เขียนว่า เข้าใจแล้วและคลิกให้ ต่อมาเมื่อกดแล้วจะเข้ามาอยู่หน้าร้าน ภายในเกมบอทจะหาคำว่ารสชาติเมื่อเจอแล้วก็จะคลิกและเข้าไปทำงานตามรูปที่ 3.11

### 3.5.5 การทำงานหลายหน้าจอ

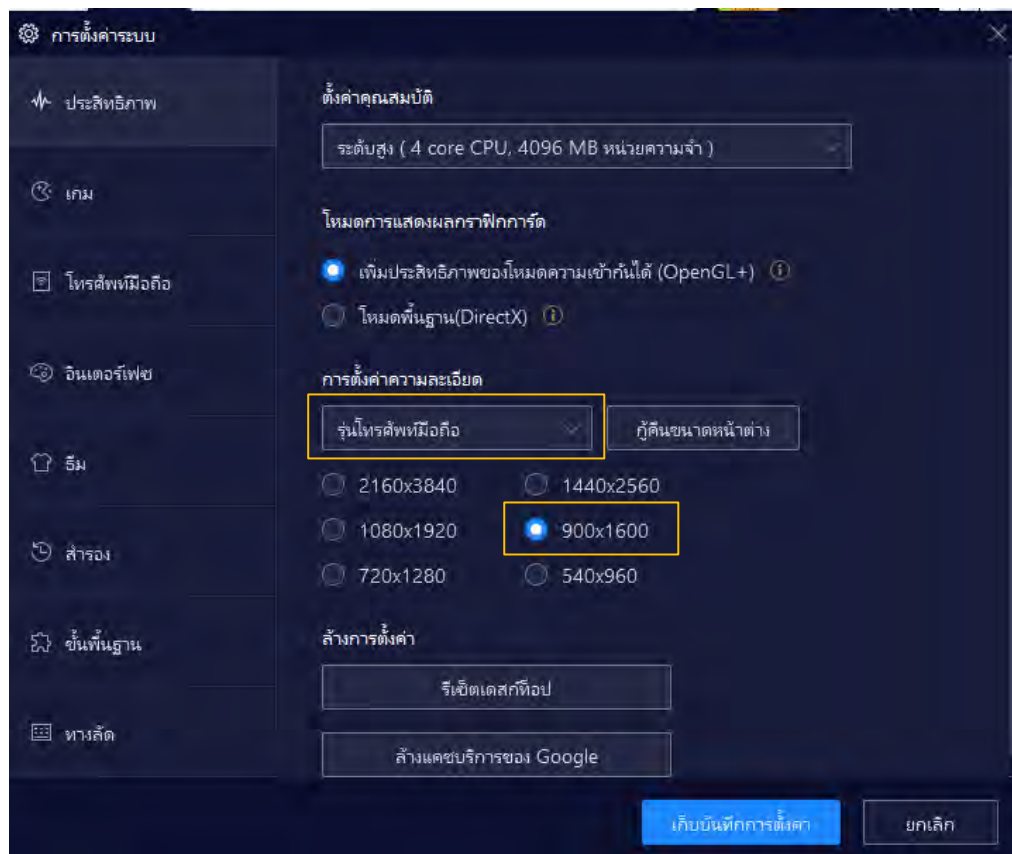


รูปที่ 3.12 การทำงานหลายหน้าจอ

บรรทัดที่ 1 `Process( target=start_bot_1 , args=(device,) ).start()`

จากรูปที่ 3.12 คือการเปิดหน้าจอหลายหน้าจอโดยหลักการทำงานของหลายหน้าจอ คือตาม code บรรทัดที่ 1 โดยใช้โมดูล Processing และใส่ฟังก์ชันชื่อ start\_bot\_1 ลงไปในฟังก์ชัน start\_bot\_1 จะมีการทำงานของบอททั้งหมดอยู่ และใส่ args=(device,) คือ device คือ การเชื่อมต่ออิมูเลเตอร์ โดยถ้ามีหลายหน้าจอบอทก็จะเชื่อมต่อให้หมดทุกหน้าจอ และสุดท้ายคือคำสั่ง start() คือคำสั่งให้บอทเริ่มทำงาน โดยจะทำงานภายใต้ for loop คือมันจะทำงานไปเรื่อยๆโดยไม่มีรอบในการทำ

### 3.6 การออกแบบการตั้งค่าหน้าจอของโปรแกรมจำลอง



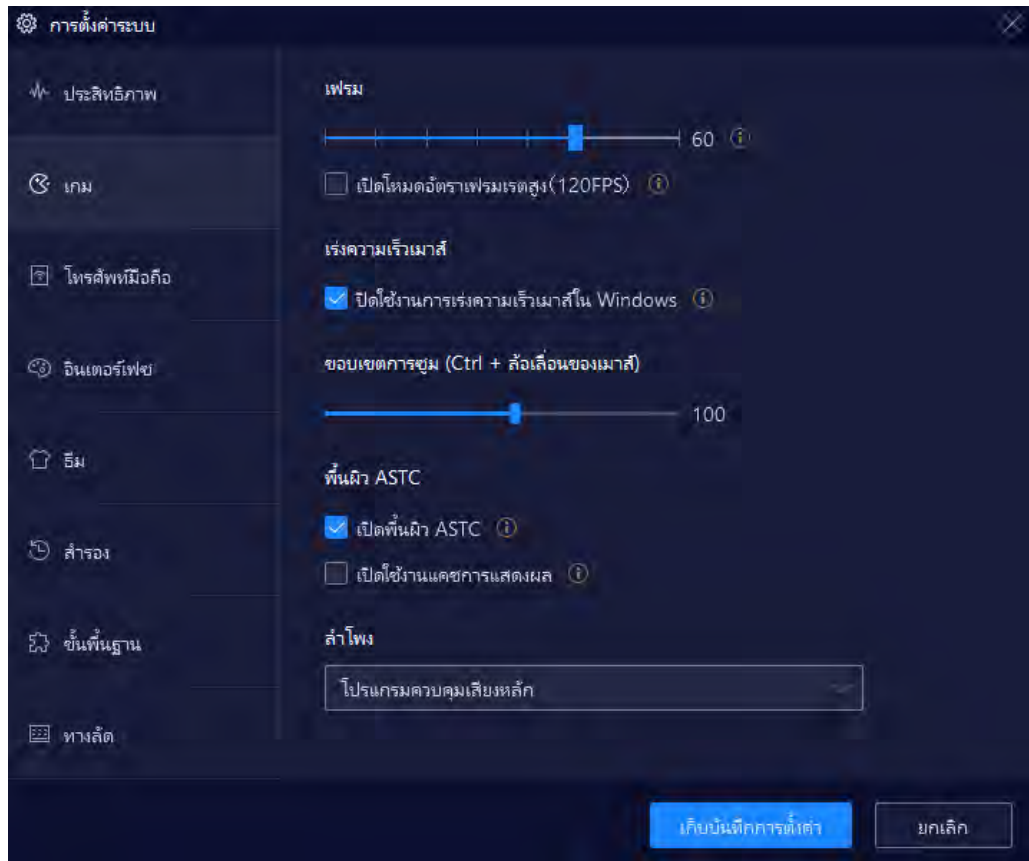
รูปที่ 3.13 ตั้งค่าขนาดหน้าจอโปรแกรมจำลอง

จากรูปที่ 3.13 คือการตั้งค่าหน้าโปรแกรมจำลองโดยต้องตั้งค่าให้เป็นแบบรูปแบบมือถือเพราะเกมนั้นเล่นในมือถือ และตั้งหน้าจอของโปรแกรมจำลองให้มีขนาด 900 X 1600 จะได้รูปดังภาพ 3.14



รูปที่ 3.14 ขนาด 900X1600 ของโปรแกรมจำลอง

ขนาดของโปรแกรมจำลองจำเป็นจะต้องเป็นขนาดเดิมทุกครั้งเมื่อทำการรันบอทและถ้าเปิดหลายจอก็ต้องตั้งค่าให้เหมือนกันด้วยทุกจอไม่งั้นขนาดรูปภาพจะผิดเพี้ยนและโปรแกรมจะหารูปภาพไม่เจอทำให้บอทไม่สามารถทำงานได้



รูปที่ 3.15 ตั้งค่าเฟรมเรตโปรแกรมจำลอง

จากรูปที่ 3.15 คือการตั้งค่าเฟรมเรตของโปรแกรมจำลองโดยโรงงานนี้จะตั้งไว้ที่ 60 FPS และหากเปิดหลายหน้าจอก็ต้องตั้งค่าให้เท่ากันด้วยไม่งั้นตัวบอทอาจจะทำงานผิดพลาดได้

### 3.7 คุณลักษณะของเครื่องคอมพิวเตอร์ที่ใช้ทำบอท

ตารางที่ 3.1 คุณลักษณะของเครื่องคอมพิวเตอร์ที่ใช้ทำบอท

คุณลักษณะ	รายละเอียด
หน้าจอคอม	1920 X 1080
CPU	I7-10700
การ์ดจอ	GTX 1080 TI
Ram	32 GB
Window	Win 10



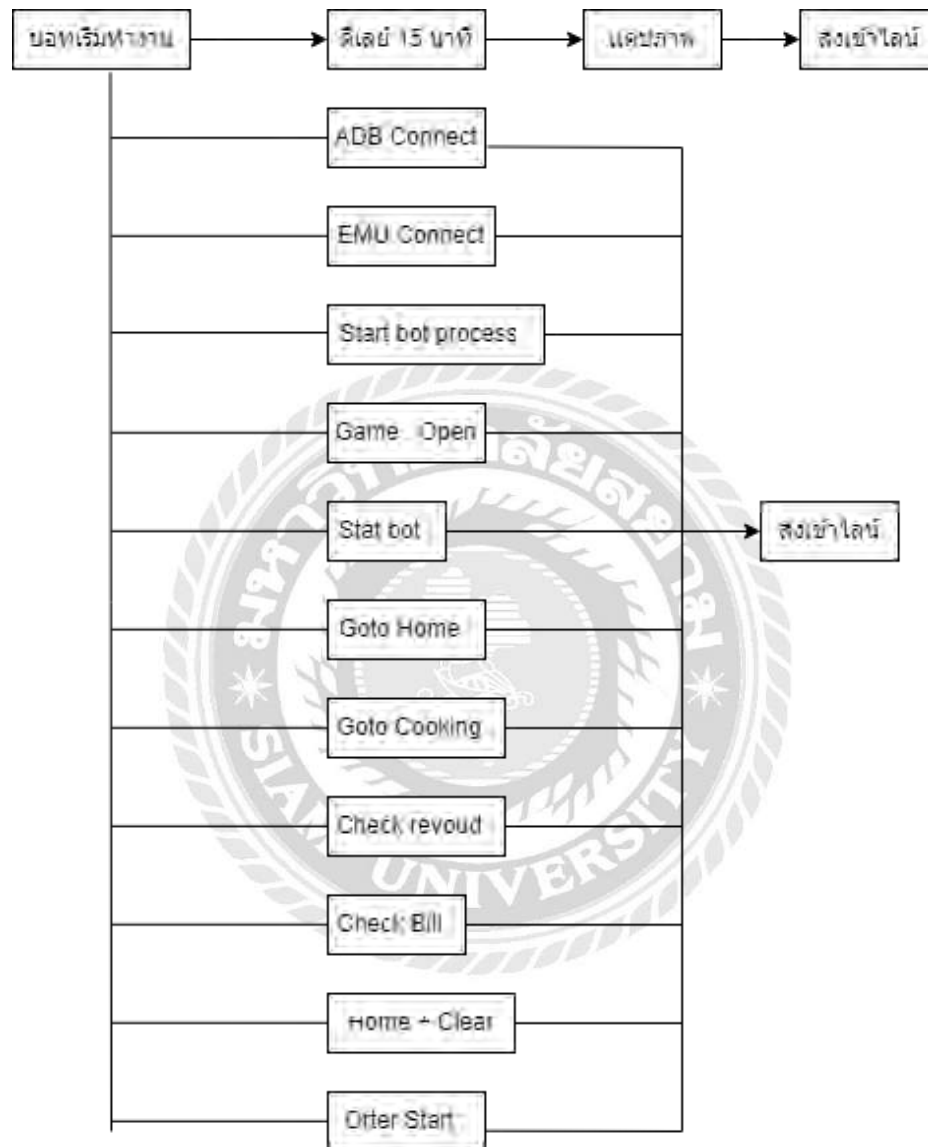
## 3.8 ออกแบบการแคปภาพเข้า linenotify



รูปที่ 3.16 การแคปรูปและเซฟลงโฟลเดอร์

การบันทึกรูปจะใช้คำสั่ง `tool.cap_img ( device , 'screen.png' )` โดยจะมีตัว `device` คือค่าโปรแกรมของโปรแกรมจำลอง และ `'screen.png'` คือการกำหนดชื่อไฟล์ เวลาเราทำการรันบอทบอททำการแคปหน้าจอตัวโปรแกรมก็จะทำการเซฟรูปนั้นลงในชื่อไฟล์ที่ตั้งไว้ให้และเซฟรูปลงในโฟลเดอร์ที่เราได้ทำการเขียนบอทไว้

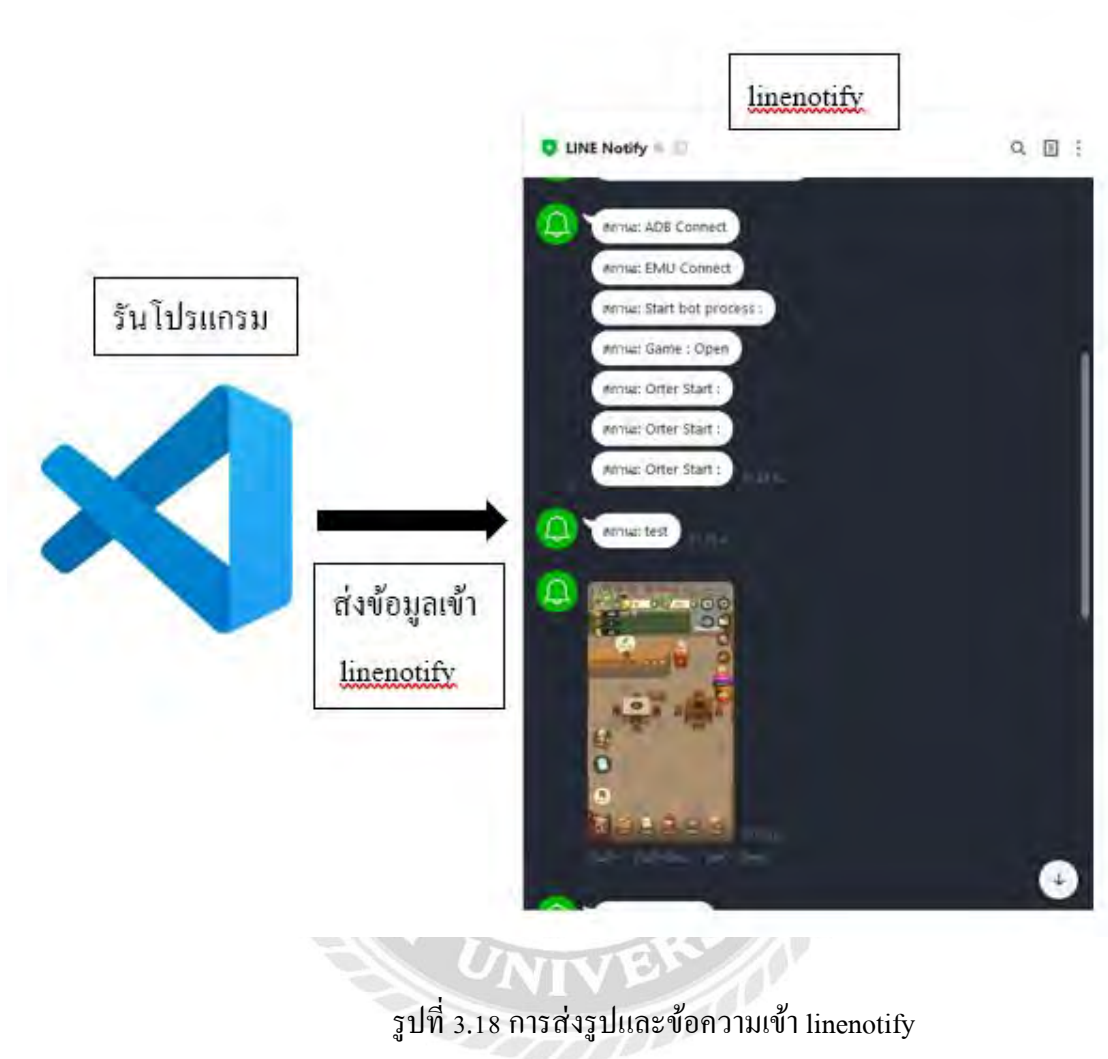
## 3.9 การออกแบบส่งภาพและข้อความเข้าไลน์



รูปที่ 3.17 บล็อกไดอะแกรมการทำงานของบอท



### 3.10 การออกแบบการส่งภาพและข้อความเข้า line notify



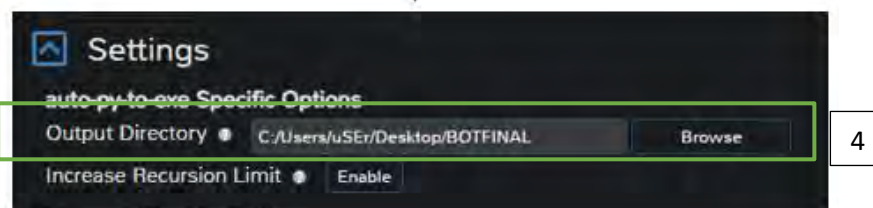
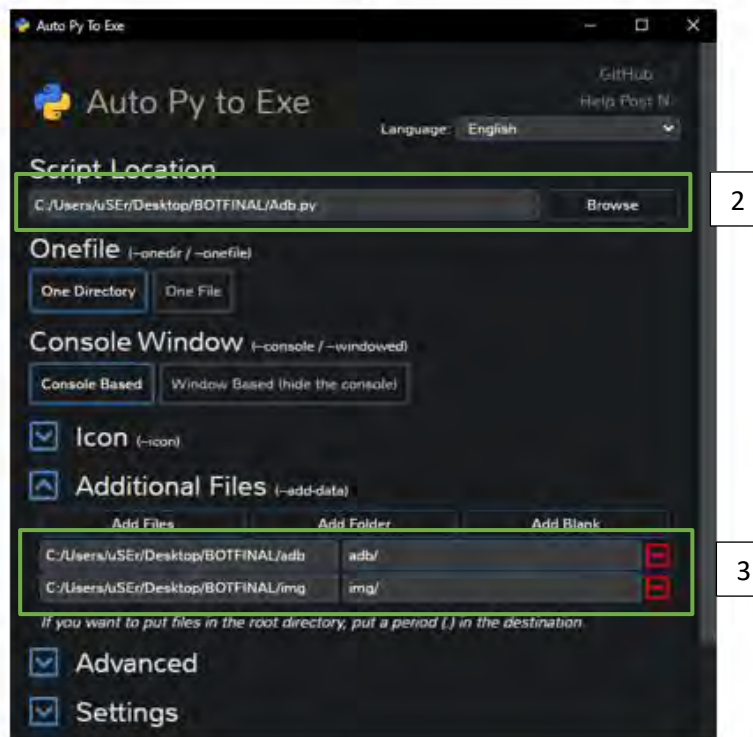
จากภาพ 3.18 คือจะเขียนโปรแกรมส่งภาพที่แคปได้จากรูป 3.18 เข้าไปทางไลน์โดยเมื่อทำการแคปรูปภาพภายในเกมเสร็จโปรแกรมจะส่งเข้าไลน์ทันที และการส่งข้อความจะมีการส่งว่าตอนนี้กำลังทำอะไรอยู่ เช่น Open คือเปิดเกมอยู่ EMU Connect คือ เชื่อมต่อ EMU แล้ว เป็นต้น

### 3.11 การแปลงไฟล์เป็น .exe

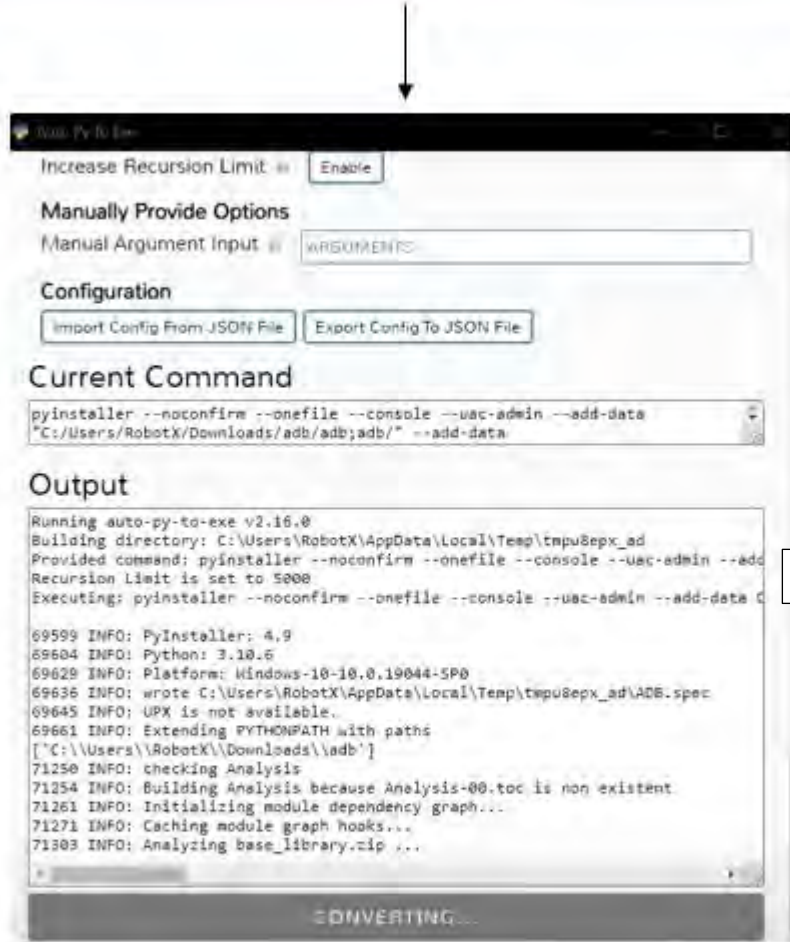
การแปลงไฟล์เป็นนามสกุล .exe เพื่อให้สามารถรัน โปรแกรมได้โดยไม่ต้องเข้าไปรันใน หน้า Visual Studio Code และสามารถส่งให้คนอื่นใช้โปรแกรมได้ด้วยโดยไม่ต้องติดตั้งอะไรเลย นอกจากตัวไฟล์ .exe

```

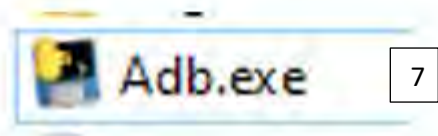
C:\Users\RobotX>auto-py-to-exe
  
```



CONVERT .PY TO .EXE 5



6



รูปที่ 3.19 การแปลงไฟล์เป็น .exe

## บทที่ 4

### การทดลอง

บทนี้เป็นการกล่าวถึงการทดลองการใช้งานของบอทเกม เพื่อให้ทราบถึงการทำงานจริงทำการทดลองการทำงานของตัวโปรแกรมที่ละฟังก์ชัน ทีละขั้นตอนว่าทำงานตรงตามที่ออกแบบไว้หรือไม่ เพื่อให้ทราบถึงข้อผิดพลาดที่เกิดขึ้นกับตัวบอท และนำข้อผิดพลาดนั้นมาทำการปรับปรุงและแก้ไขให้สามารถทำงานได้อย่างต่อเนื่อง และมีประสิทธิภาพตลอดระยะเวลาที่รันโปรแกรมอยู่

#### 4.1 การทดลองคำสั่งเชื่อมต่อ ADB และ อิมูเรเตอร์

```
ta\Local\Programs\Python\Python311\python.exe' 'c:\Users\uSER\.vscode\extensions\ms-python.python-2023.6.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '62535' '--' 'C:\Users\uSER\Desktop\BOT_หม้อไฟ\Adb.py'  
ADB Connect  
EMU Connect  
[<ppadb.device.Device object at 0x0000023F7FF25210>]  
PS C:\Users\uSER\Desktop\BOT_หม้อไฟ
```

#### รูปที่ 4.1 ผลการทดลองคำสั่งเชื่อมต่อ ADB และ อิมูเรเตอร์

จากการทดลองเมื่อทำการรันโปรแกรมที่เขียนคำสั่งเชื่อมต่อ ADB และ อิมูเรเตอร์ไว้สามารถเชื่อมต่อได้ปกติโดยถ้าเชื่อมต่อได้จะแสดงค่า ADB Connect และ EMU Connect

#### 4.2 การทดลองคำสั่งการแสดงค่า อิมูเรเตอร์เก็บไว้ในรูปเพื่อแสดงค่าที่ละตัว

```
\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\uSER\.vscode\extensions\ms-python.python-2023.6.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '53171' '--' 'C:\Users\uSER\Desktop\BOT_หม้อไฟ\Adb.py'  
ADB Connect  
EMU Connect  
<ppadb.device.Device object at 0x0000029E5C307710>  
PS C:\Users\uSER\Desktop\BOT_หม้อไฟ
```

#### รูปที่ 4.2 การแสดงค่าของอิมูเรเตอร์เก็บไว้ในรูป

จากการทดลองเมื่อทำการรันโปรแกรมที่เขียนคำสั่งเก็บค่าอิมูเรเตอร์สามารถประมวลผลค่าออกมาได้ปกติโดยจะเก็บไว้ในรูปเพื่อที่จะนำค่ามาใช้ต่อเพื่อจะมีการเปิดเกมหลายหน้าจอ

### 4.3 การทดสอบนำค่าของอิมูเรเตอร์เข้าไปในฟังก์ชัน

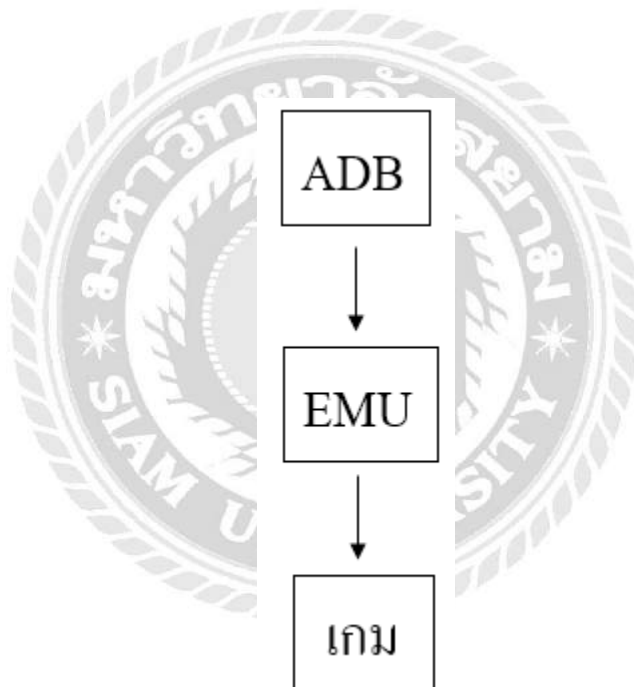
```

\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\uSER\.vscode\extensions\ms-python.
python-2023.6.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '54842' '--' 'C:\U
sers\uSER\Desktop\BOT_หม้อไฟ\Adb.py'
ADB Connect
EMU Connect
ค่าเข้าแล้ว
PS C:\Users\uSER\Desktop\BOT_หม้อไฟ>

```

รูปที่ 4.3 การแสดงค่าลิตรในฟังก์ชัน

แสดงค่าลิตรในฟังก์ชัน โดยในฟังก์ชันจะเขียนคำสั่งให้แสดงคำว่า ค่าเข้าแล้ว เมื่อมีการรันโปรแกรม เพื่อเป็นการทดสอบว่าค่าได้เข้าไปในฟังก์ชันแล้ว ที่ต้องเอาค่าของอิมูเรเตอร์เข้าไปในฟังก์ชัน เพราะต่อไปจะเขียนโปรแกรมการทำงานทั้งหมดภายใต้ฟังก์ชัน



รูปที่ 4.4 ไคอะแกรมการเชื่อมต่อ ADB และ EMU

การจะเชื่อมต่อไปยังเกมได้ต้องเชื่อมต่อ ADB ก่อนเหมือนเป็นตัวกลางเพื่อให้ตัวบอทสามารถทำงานกับตัวอิมูเรเตอร์ได้และตัวอิมูเรเตอร์ก็จะไปเชื่อมต่อกับตัวเกมอีกที

## 4.4 ตรวจสอบการอยู่หน้าหลัก



รูปที่ 4.5 การตรวจสอบการอยู่หน้าหลัก

ทำการเขียนให้บอทจับภาพโดยแคปภาพไว้และเก็บในโฟลเดอร์ที่เขียนบอท ตามรูปที่ 4.5 เพื่อเช็คว่ตอนนี้กำลังทำงานอยู่ที่หน้าหลักถ้าบอทตรวจเจอรูปภาพที่มีใน โฟลเดอร์และตรงกับ หน้าจอเกมบอทจะทำการติกรอบรูปภาพให้ตามรูปที่ 4.5

ผลการทดลอง : บอทสามารถทำงานได้ปกติ



## 4.5 ตรวจสอบรูปไม้กวาดและรูปกระดาษ



รูปที่ 4.6 ตรวจสอบรูปไม้กวาดและรูปกระดาษ

ทำการเขียนให้บอทจับภาพโดยแคปภาพไว้และเก็บในโฟลเดอร์ที่เขียนบอท ตามรูปที่ 4.6 เมื่อทำการรันบอทหากบอทเจอภาพจะทำการคลิกที่ไปรูปนั้นหากเจอหลายรูปก็จะคลิกให้หมดทุกรูปและตีกรอบสีเขียวรูปภาพที่หาเจอทั้งหมด

ผลการทดลอง : บอทสามารถทำงานได้ปกติ

## 4.6 ตรวจสอบรูปเงิน



รูปที่ 4.7 ตรวจสอบรูปเงิน

ทำการเขียนให้บอทจับภาพโดยแคปภาพไว้และเก็บในโฟลเดอร์ที่เขียนบอท ตามรูปที่ 4.7 เมื่อทำการรันบอทหากบอทเจอภาพเงินจะทำการคลิกที่ไปรูปนั้นหากเจอหลายรูปก็จะคลิกให้หมดทุกรูปและติกรอบสีเขียวรูปภาพที่หาเจอทั้งหมด

ผลการทดลอง : บอทสามารถทำงานได้ปกติ



## 4.7 ตรวจสอบรูปยืนยัน



รูปที่ 4.8 ตรวจสอบรูปยืนยัน

ต่อเนื่องจากภาพ 4.8 เมื่อกดรูปเงินเสร็จบอทจะจับภาพรูปยืนยันหากเจอก็จะทำการคลิกให้ก่อนจะถึงรูปที่ 4.8 จะต้องผ่านรูป 4.7 ก่อน

ผลการทดลอง : บอทสามารถทำงานได้ปกติ

## 4.8 เลื่อนหน้าจอเปลี่ยนไปหน้าครัว



รูปที่ 4.9 เลื่อนหน้าจอเปลี่ยนไปหน้าครัว

ทำการเขียนให้บอทให้หาตำแหน่งค่า  $x, y$  ทั้ง 2 ตำแหน่งบนหน้าจอ โดยตำแหน่งแรกคือ 1000 , 20 และ ตำแหน่งที่สองคือ 1000 , 800 เมื่อทำการรันบอทบอทจะทำการกดค้างที่ตำแหน่งแรกไว้และเลื่อนเมาส์ไปทางขวาคือตำแหน่งที่สอง เกมจะทำการเปลี่ยนไปหน้าครัวให้ โดยจะกำหนด ดีเลย์ประมาณ 1 วินาที

ผลการทดลอง : บอทสามารถทำงานได้ปกติ



รูปที่ 4.10 หน้าครัว

จากหัวข้อ 4.9 เมื่อเลื่อนจอเสร็จตัวเกมจะทำการมาที่หน้าครัวตามรูปที่ 4.10

## 4.9 การตรวจสอบการอยู่หน้าครัว



รูปที่ 4.11 การตรวจสอบการอยู่หน้าครัว

ทำการเขียนให้บอทจับภาพโดยแคปภาพไว้และเก็บในโฟลเดอร์ที่เขียนบอท ตามรูปที่ 4.11 เพื่อเช็คว่าตอนนี้กำลังทำงานอยู่ที่หน้าครัว ถ้าบอทตรวจเจอรูปภาพที่มีใน โฟลเดอร์และตรงกับหน้าจอเกมบอทจะทำการตีกรอบรูปภาพให้ตามรูปที่ 4.11

ผลการทดลอง : บอทสามารถทำงานได้ปกติ



## 4.10 ตรวจสอบรูปงานอาหารที่ 1 และ ที่ 2 พร้อมรูปงานในอ่างล้างจาน



รูปที่ 4.12 รูปงานอาหารที่ 1 และ ที่ 2

ทำการเขียนให้บอทจับภาพโดยแคปภาพไว้และเก็บในโฟลเดอร์ที่เขียนบอท ตามรูปที่ 4.12 เมื่อทำการรันบอทหากบอทเจอภาพจะทำการคลิกที่ไปรูปนั้นหากเจอหลายรูปก็จะคลิกให้หมดทุกรูปและตีกรอบสีเขียวรูปภาพที่หาเจอทั้งหมดโดยจะคลิกที่รูปงานที่ 1 ก่อนและค่อยคลิกรูปงานที่ 2

ผลการทดลอง : บอทสามารถทำงานได้ปกติ



รูปที่ 4.13 รูปร่างล้างจาน

ทำการเขียนให้บอทจับภาพโดยแคปภาพไว้และเก็บในโฟลเดอร์ที่เขียนบอท ตามรูปที่ 4.13 เมื่อทำการรันบอทหากบอทเจอภาพจะทำการคลิกที่ไปรูปนั้น โดยจะต่อจากรูปที่ 4.12 หากเจอรูปใน 4.12 จะทำงานในรูปนั้นก่อนแต่หากไม่เจอก็จะมาทำงานที่รูป 4.13 เลย

ผลการทดลอง : บอทสามารถทำงานได้ปกติ

## 4.11 เลื่อนหน้าจอเปลี่ยนไปหน้าครัว



รูปที่ 4.14 เลื่อนหน้าจอเปลี่ยนไปหน้าครัว

ทำการเขียนให้บอทให้หาตำแหน่งค่า  $x, y$  ทั้ง 2 ตำแหน่งบนหน้าจอ โดยตำแหน่งแรกคือ 1000 , 800 และ ตำแหน่งที่สองคือ 1000 , 20 เมื่อทำการรันบอทบอทจะทำการกดค้างที่ตำแหน่งแรกไว้และเลื่อนเมาส์ไปทางขวาคือตำแหน่งที่สอง เกมจะทำการเปลี่ยนไปหน้าหลักให้ โดยจะกำหนด ดีเลย์ประมาณ 1 วินาที

ผลการทดลอง : บอทสามารถทำงานได้ปกติ

## 4.12 ตรวจสอบรูปแมว

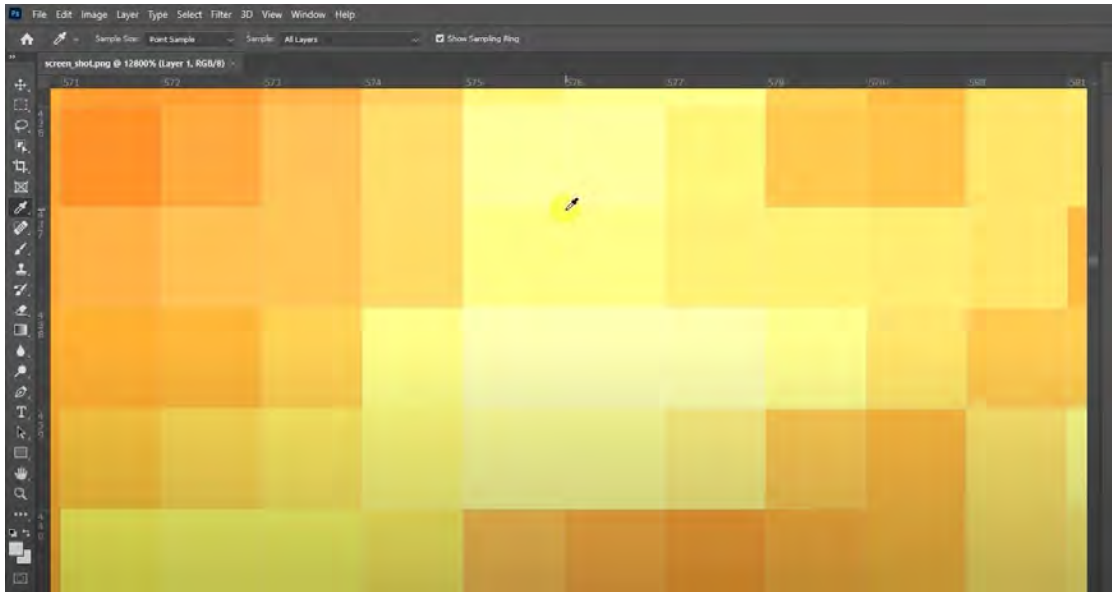


รูปที่ 4.15 ตรวจสอบรูปแมว

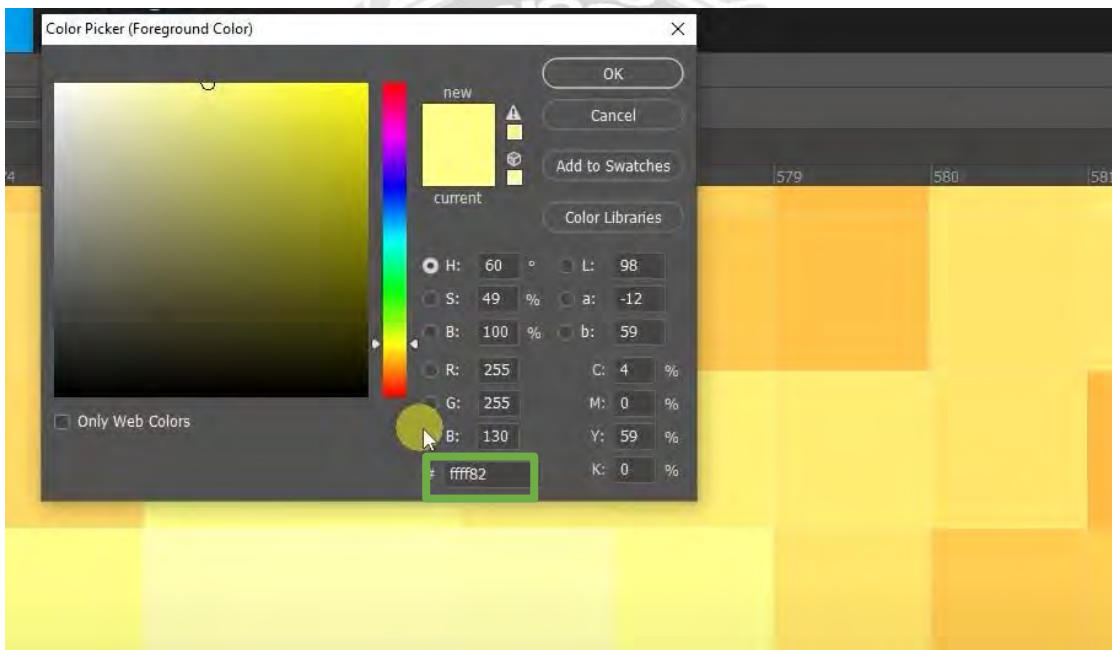
เมื่อร้านบอทบอทจะทำการหารูปแมวที่เรืองแสง หากมีการเรืองแสงบอทจะหาค่าสีของรูป และหาว่าอยู่ที่ตำแหน่งไหนบอทก็จะทำการคลิกให้

ผลการทดลอง : บอทสามารถทำงานได้ปกติ





รูปที่ 4.16 การหาค่าตำแหน่งรูปแมว



รูปที่ 4.17 การหาค่าสีรูปแมว

การหาค่าสีและตำแหน่ง ดังรูปที่ 4.16 และ 4.17 สามารถหาได้จาก โปรแกรม photoshop โดยค่าสีของแมวคือ FFFF88H และตำแหน่งของสีคือ 427 , 530

## 4.13 ตรวจสอบหน้ารับรางวัล



รูปที่ 4.18 ตรวจสอบหน้ารับรางวัล

เมื่อกดรูปแมวตามรูปที่ 4.15 แล้วจะด้งมาหน้ารับรางวัลตามรูปที่ 4.18 บอทจะตรวจสอบตำแหน่งของกากบาทเพื่อเป็นการเช็คว่าตอนนี้ได้อยู่หน้ารับรางวัลเรียบร้อยแล้ว

ผลการทดลอง : บอทสามารถทำงานได้ปกติ

## 4.14 รับรางวัลครั้งที่ 1



รูปที่ 4.19 รับรางวัลครั้งที่ 1

เมื่อตรวจเช็คแล้วว่าอยู่หน้ารับรางวัลตามหัวข้อที่ 4.18 บอทจะทำการเช็คตำแหน่งของคำว่ารับว่าอยู่ตรงไหนพอเจอแล้วบอทจะทำการกดรับของรางวัลให้โดยบอทจะเช็คค่า 2 ค่าเพื่อการทำค่าใดค่าหนึ่งไม่เจอและให้คลิกที่ค่า ['1100','450'] คือตำแหน่ง x.y

ผลการทดลอง : บอทสามารถทำงานได้ปกติ

## 4.15 กดรับกล่อง



รูปที่ 4.20 กดรับกล่อง

เมื่อกดรับของตามรูปที่ 4.19 แล้ว บอทจะทำการเช็คตำแหน่งของกล่องรางวัลว่าอยู่ตรงไหนพอเจอแล้วบอทจะทำการกดไปที่กล่องให้โดยบอทจะเช็คค่า 2 ค่าเพื่อการหาค่าใดค่าหนึ่งไม่เจอและให้คลิกที่ค่า  $[1250, 650]$  คือตำแหน่ง  $x,y$  และตัวเลข 5 วินาทีก่อนเข้าขั้นตอนต่อไป

ผลการทดลอง : บอทสามารถทำงานได้ปกติ

## 4.16 กดรับของรางวัลครั้งที่ 2



รูปที่ 4.21 รับรางวัลครั้งที่ 2

เมื่อกดไปที่กล่องตามรูปที่ 4.20 แล้ว บอทจะทำการเช็คตำแหน่งของคำว่ารับว่าอยู่ตรงไหน พอเจอแล้วบอทจะทำการกดไปที่คำว่ารับให้ โดยบอทจะเช็คค่า 2 ค่าเพื่อการหาค่าใดค่าหนึ่งไม่เจอ และให้คลิกที่ค่า [1070,445] คือตำแหน่ง x,y

ผลการทดลอง : บอทสามารถทำงานได้ปกติ



## 4.17 กดปิดหน้าต่างรับรางวัล

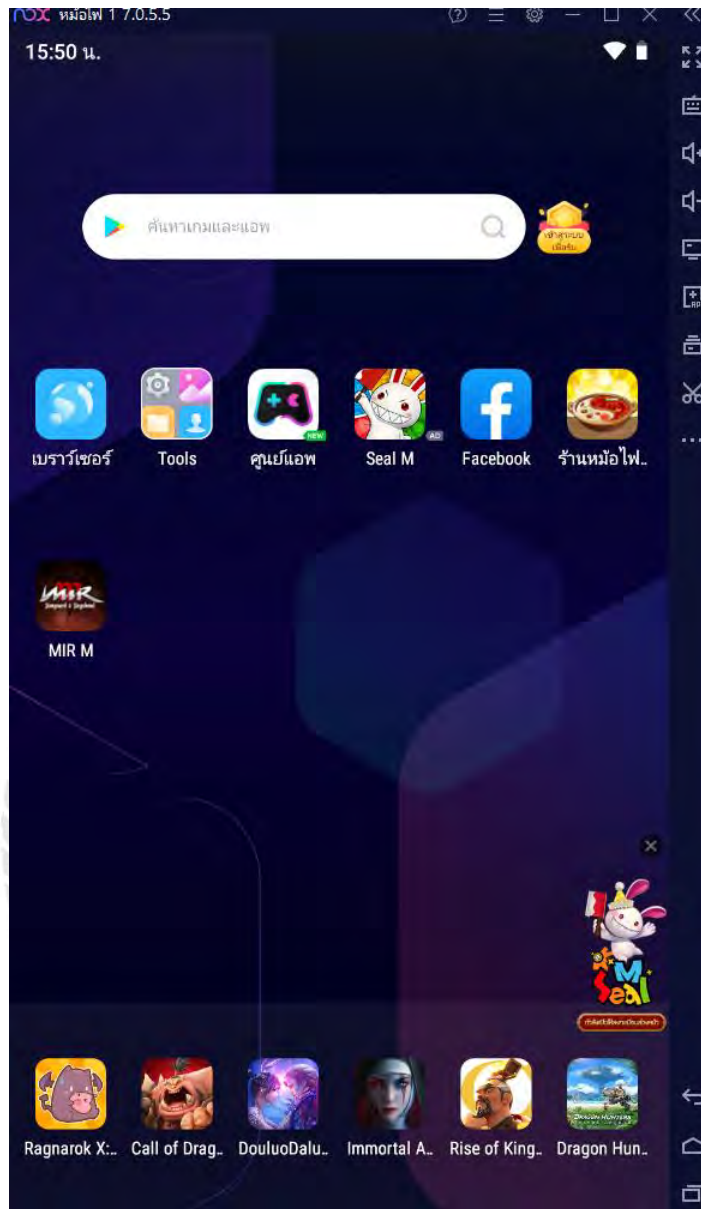


รูปที่ 4.22 กดปิดหน้าต่างรับรางวัล

เมื่อเสร็จกระบวนการตั้งแต่รูป 4.18 ถึง 4.21 แล้ว บอทจะตรวจเช็คภาพกากบาทที่ได้ทำการ  
บันทึกเก็บไว้ใน โฟลเดอร์หากรูปตรงกันบอทจะทำการคลิกให้พร้อมตีกรอบสีเหลี่ยมตรงรูปให้ด้วย

ผลการทดลอง : บอทสามารถทำงานได้ปกติ

## 4.18 ทดสอบการรีเซ็ตเกมตามเวลาที่กำหนด

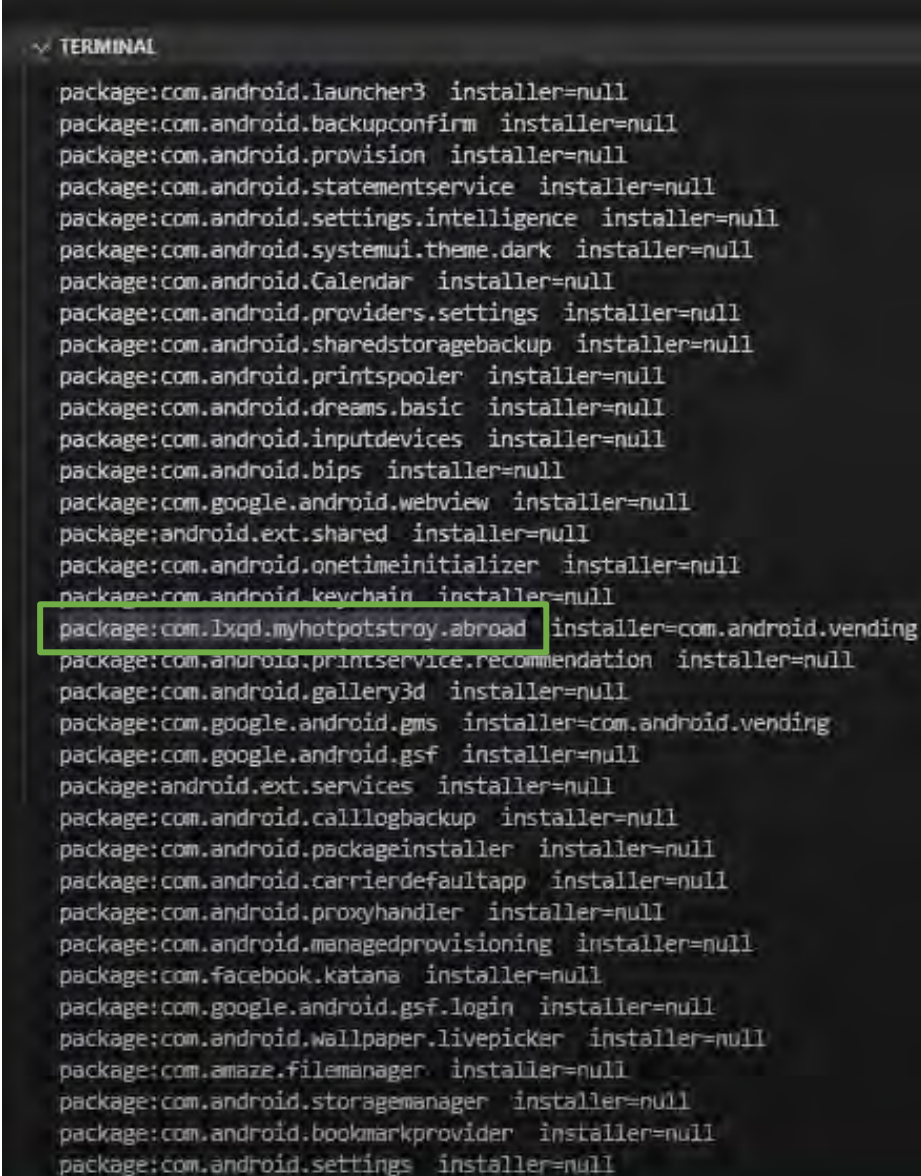


รูปที่ 4.23 การรีเซ็ตเกม

เขียนการรีเซ็ตโดยใช้คำสั่ง `time_count` และให้ตรวจเช็คชื่อแอปพลิเคชัน พร้อมเช็คเวลาที่ตั้งไว้หากครบเวลาตัวบอลจะทำการกด F5 ให้เองอัตโนมัติและดึงออกมายังหน้าต่างของตัว Nox ตามรูปที่ 4.23

ผลการทดลอง : บอทสามารถทำงานได้ปกติ

## 4.19 ทดสอบคำสั่งการหาชื่อแอปพลิเคชัน



```

✓ TERMINAL
package:com.android.launcher3 installer=null
package:com.android.backupconfirm installer=null
package:com.android.provision installer=null
package:com.android.statementservice installer=null
package:com.android.settings.intelligence installer=null
package:com.android.systemui.theme.dark installer=null
package:com.android.Calendar installer=null
package:com.android.providers.settings installer=null
package:com.android.sharedstoragebackup installer=null
package:com.android.printspooler installer=null
package:com.android.dreams.basic installer=null
package:com.android.inputdevices installer=null
package:com.android.bips installer=null
package:com.google.android.webview installer=null
package:android.ext.shared installer=null
package:com.android.onetimeinitializer installer=null
package:com.android.keychain installer=null
package:com.lxqd.myhotpotstroy.abroad installer=com.android.vending
package:com.android.printservice.recommendation installer=null
package:com.android.gallery3d installer=null
package:com.google.android.gms installer=com.android.vending
package:com.google.android.gsf installer=null
package:android.ext.services installer=null
package:com.android.calllogbackup installer=null
package:com.android.packageinstaller installer=null
package:com.android.carrierdefaultapp installer=null
package:com.android.proxyhandler installer=null
package:com.android.managedprovisioning installer=null
package:com.facebook.katana installer=null
package:com.google.android.gsf.login installer=null
package:com.android.wallpaper.livepicker installer=null
package:com.amaze.filemanager installer=null
package:com.android.storagemanager installer=null
package:com.android.bookmarkprovider installer=null
package:com.android.settings installer=null

```

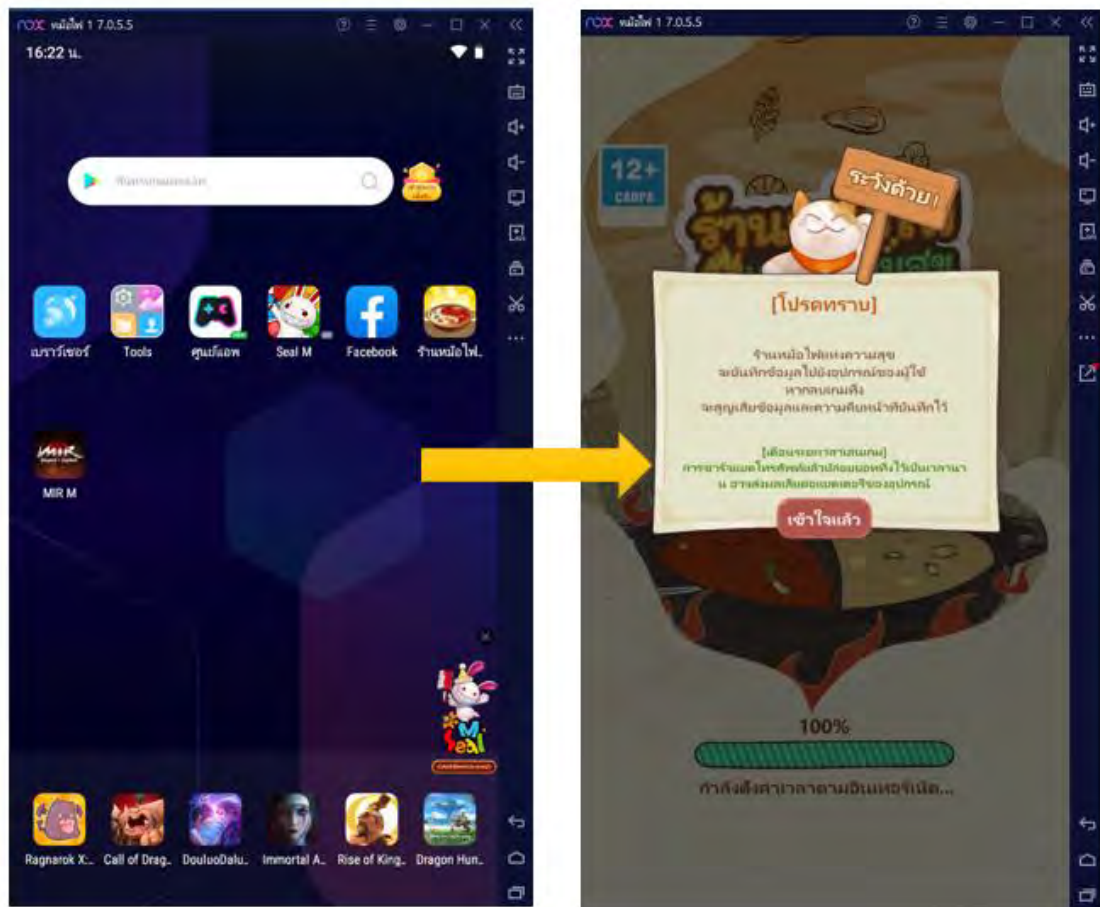
รูปที่ 4.24 ทดสอบคำสั่งการหาชื่อแอปพลิเคชัน

ทดสอบโดยการใช้คำสั่ง `input(tool.list.app(device))` เป็นคำสั่งสำหรับการหาชื่อ app ทั้งหมดที่มีอยู่ในอีมูเรเตอร์หาชื่อแอปได้ง่ายๆคือถ้าเป็นแอปที่มาจากเครื่องมันจะขึ้น null โดยเจ้าตัวเกมหม้อไฟแห่งความสุขมีชื่อว่า `com.lxqd.myhotportstroy.abroad`

ผลการทดลอง : สามารถหาชื่อแอปได้ตามปกติ



## 4.20 ทดสอบการกดเข้าเกมเอง

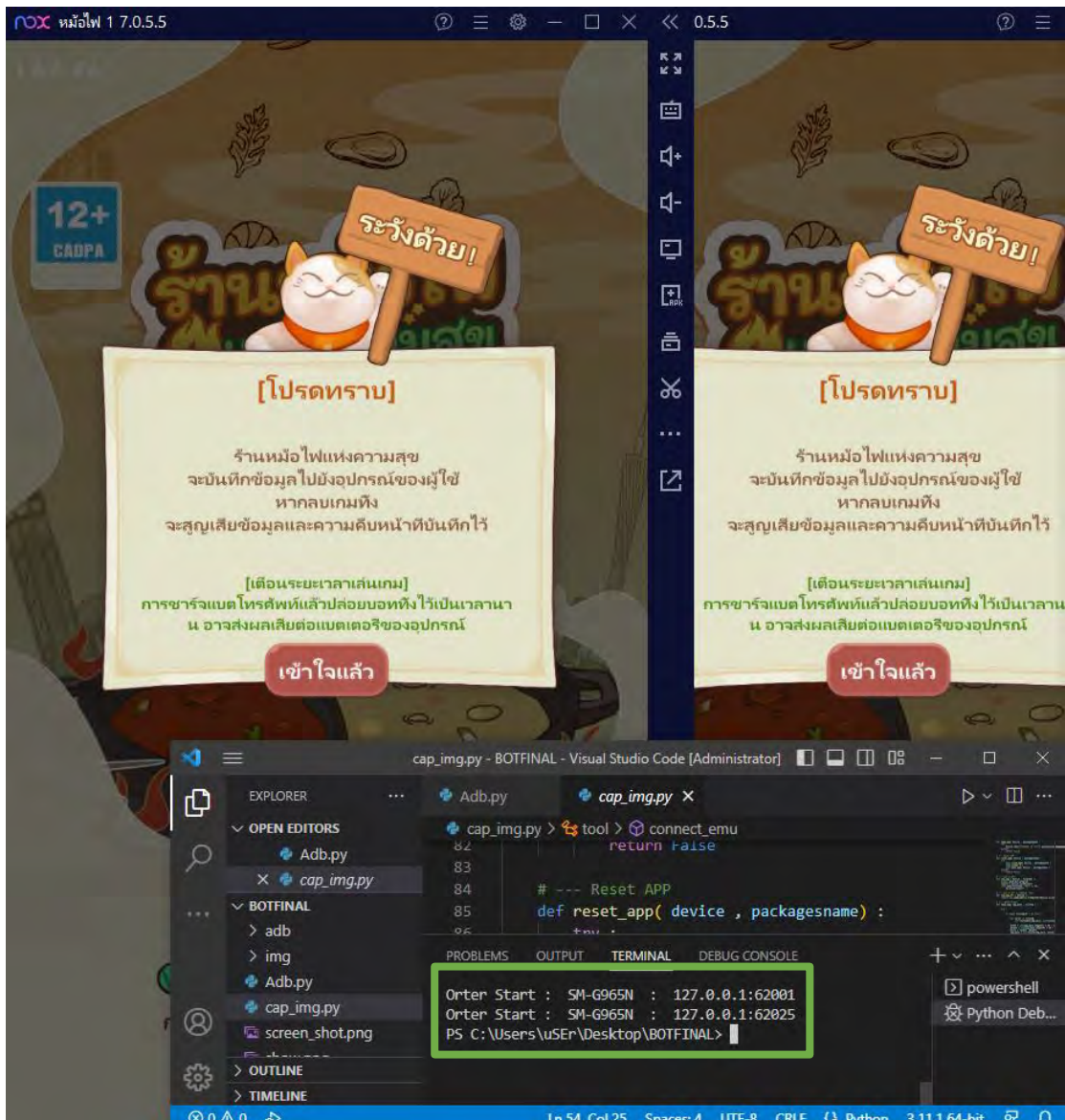


## 4.25 ทดสอบการกดเข้าเกมเอง

ทดสอบโดยการใช้คำสั่ง `tool.reset_app( device , app)` เป็นคำสั่งสำหรับการเปิดเกมเองโดยในวงเล็บเราจะใส่ device , app ซึ่งใน device จะมีค่าลิตรของตัวโปรแกรม nox อยู่ และ app จะมีค่าของตัวเกมหม้อไฟแห่งความสุขอยู่ เวลาบอทจะทำงานก็จะอ้างอิงจะ 2 ตัวแปรนี้บอทจึงสามารถเปิดตัวเกมได้อย่างถูกต้อง โดยบอทจะทำงานตอนที่ไม่ได้อยู่บนหน้าเกมก็คือทางรูปด้านซ้ายของรูปที่ 4.25 หรือเมื่อบอทมีการรีเซตเกมบอทก็จะทำการกดเข้าเกมให้เองด้วยเช่นกัน

ผลการทดลอง : สามารถรีเซตและเข้าแอปได้ตามปกติ

## 4.21 แยกการทำงานหลายหน้าจอ



รูปที่ 4.26 แยกการทำงานหลายหน้าจอ

การทำงานหลายหน้าจอเขียนโดยใช้โมดูล Process เมื่อรันโปรแกรมบอทจะแยกการทำงานออกจากกันหากมีมากกว่า 1 หน้าจอ บอทก็จะแสดงเลข IP เครื่องตามรูป 4.26

ผลการทดลอง : สามารถทำงานแยกหน้าจอได้ตามปกติ

#### 4.22 ทดสอบการทำงานบอทโดยทับหน้าจอและไม่ดึงเมาส์



รูปที่ 4.27 การทำงานบอทโดยทับหน้าจอและไม่ดึงเมาส์

การทำงานแบบทับจอมาจากตัวของโปรแกรม Emurator เองหากเขียนโปรแกรมกับตัว Emurator ก็จะสามารถรับบอทแบบทับจอได้แต่ไม่สามารถย่อจอได้ตามผลลัพธ์ดังรูป 4.27 ที่รันบอทตอนที่มีโปรแกรม VScode บังอยู่แต่บอทก็ยังทำงาน ส่วนการไม่ดึงเมาส์ คือความสามารถที่มาจากคำสั่ง Autoclick

ผลการทดลอง : สามารถหาซื้อแอปได้ตามปกติ



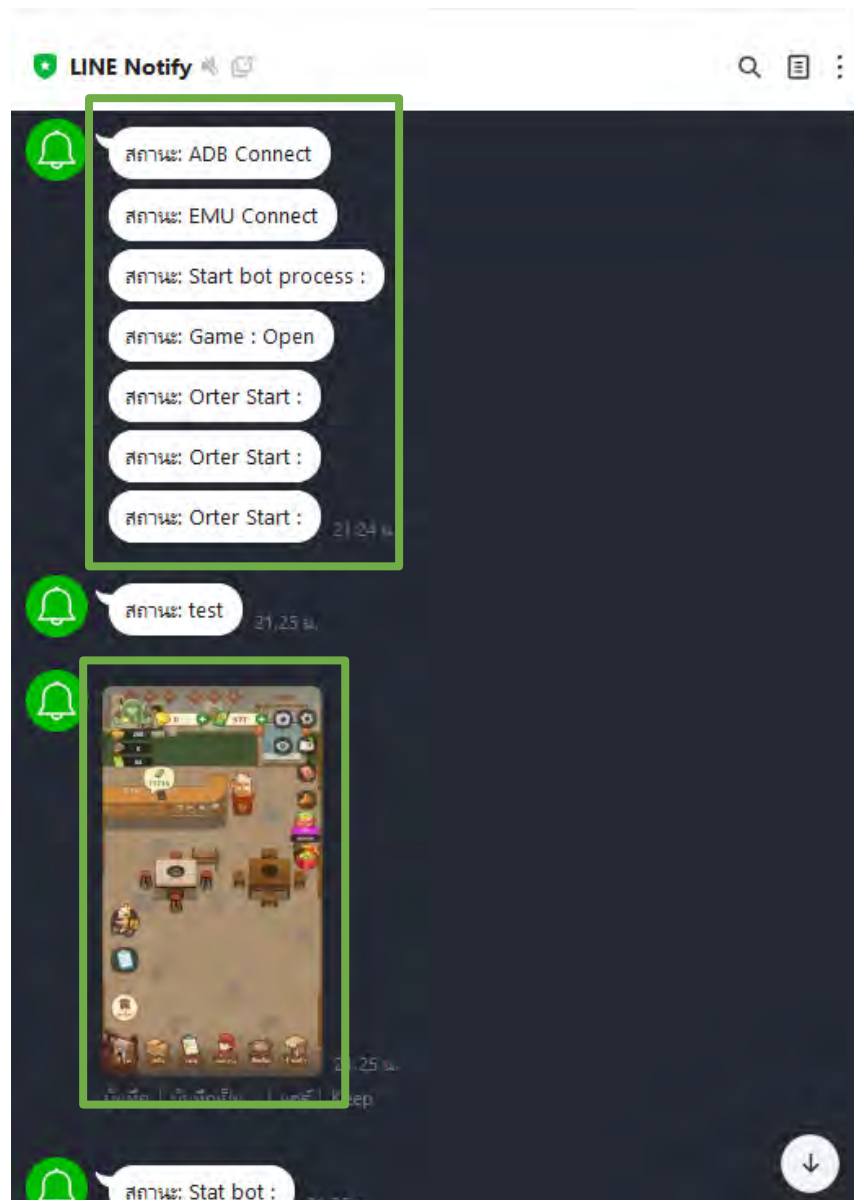
## 4.23 ทดสอบการเชื่อมต่อ line notify



รูปที่ 4.28 ทดสอบการเชื่อมต่อ line notify

จากรูปที่ 4.28 เมื่อทำการรัน code ที่ส่งข้อความเข้า line notify เสร็จแล้ว ทดสอบสามารถส่งข้อความเข้า line notify ได้ปกติโดยลองทดสอบโดยการส่งข้อความว่า “ทดสอบภาษาไทย hello”

## 4.24 ทดสอบการส่งภาพและข้อความเข้า line notify

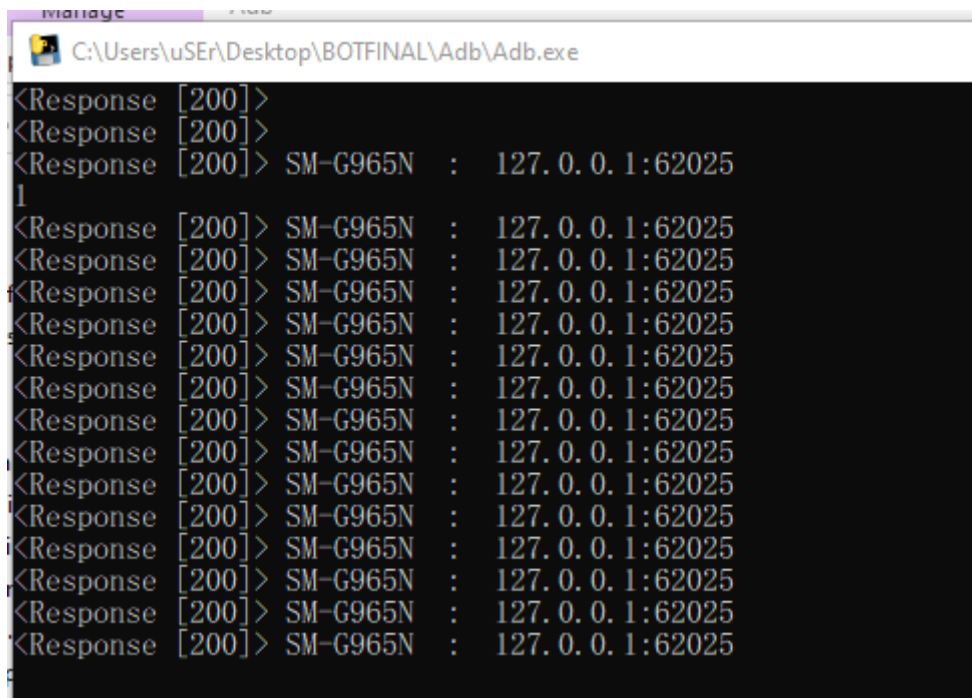


รูปที่ 4.29 ทดสอบการส่งภาพและข้อความเข้า line notify

จากรูปที่ 4.29 บอทจะส่งรูปภาพที่แคปได้จากเกมเข้าไปในไลน์เพื่อดูยอดเงินว่ายอดเงินที่ได้ได้ถึงเท่าไรแล้ว และบอทก็จะมีการส่งข้อความแจ้งเตือนการทำงานต่างๆเข้าไปในไลน์ด้วยคือเมื่อถึงการทำงานในกระบวนการๆนั้นก็จะมีข้อความแจ้งเตือนเข้าไปว่ากำลังทำอะไรอยู่

ผลการทดลอง : สามารถส่งรูปและข้อความได้ปกติ

## 4.25 ทดสอบการรันตัวบอทผ่านไฟล์ exe



```

C:\Users\user\Desktop\BOTFINAL\Adb\Adb.exe
<Response [200]>
<Response [200]>
<Response [200]> SM-G965N : 127.0.0.1:62025
|
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025
<Response [200]> SM-G965N : 127.0.0.1:62025

```

รูปที่ 4.30 ทดสอบการรันตัวบอทผ่านไฟล์ exe

จากรูปที่ 4.30 คือการทดสอบการรันโปรแกรมบอทผ่านไฟล์ .exe ไปนการดับเบิ้ลคลิกที่ไฟล์ที่ได้ทำการแปลงมาจะเห็นว่าโปรแกรมสามารถทำงานได้ปกติทุกขั้นตอนและจะแสดงเลข IP ของเครื่องนั้นๆด้วยหากเปิดหลายจอก็จะแสดงหลาย IP



## 4.26 ทดสอบประสิทธิภาพหาเงินในเกมระหว่างใช้บอทเล่นเกมกับเล่นด้วยมืออย่างละ 1 ชั่วโมง

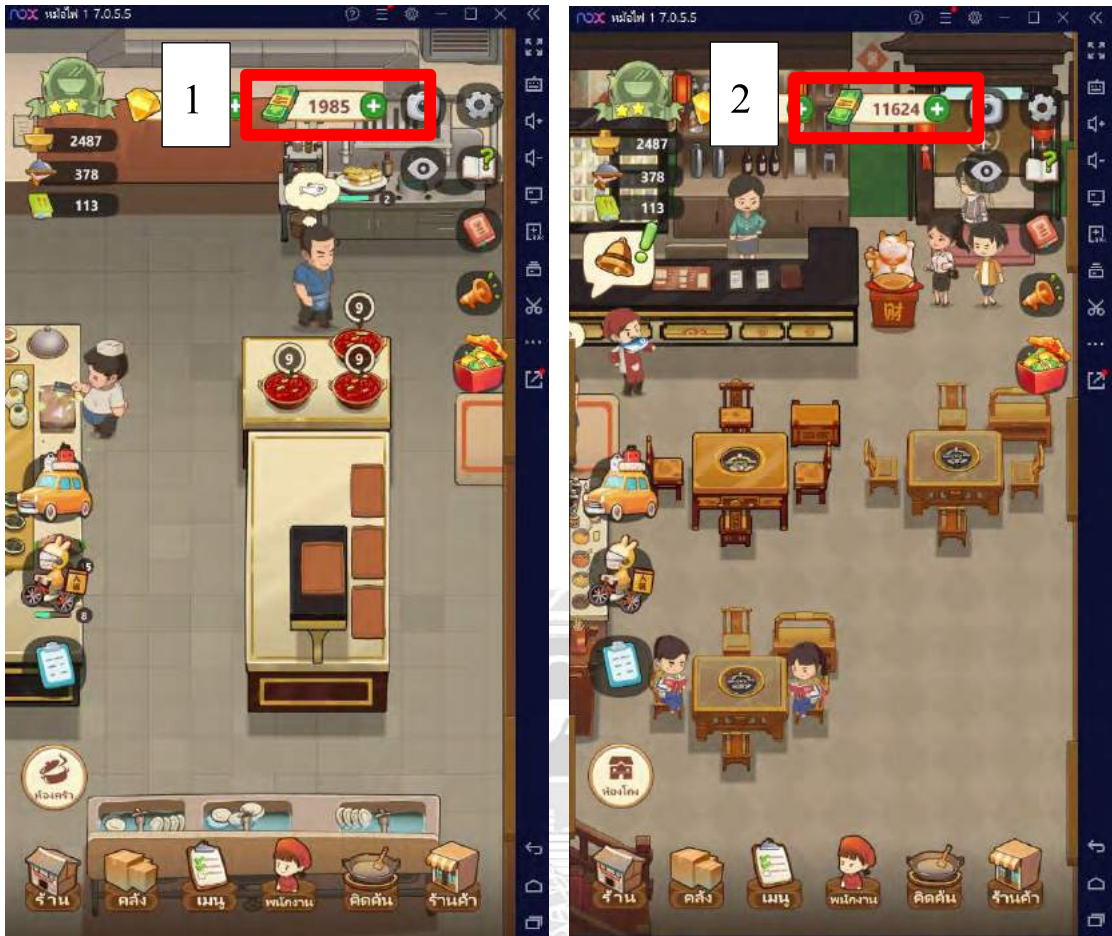


รูปที่ 4.31 ทดสอบประสิทธิภาพหาเงินในเกม 1 ชั่วโมง ด้วยการใช่มือเล่นเอง

จากการทดสอบการเล่นเกมนด้วยการใช่มือเล่น 1 ชั่วโมง

รูปที่ 1 ค่ายซ้ายมือเริ่มเล่นเกมโดยมีเงินในเกมอยู่ที่ 3512 เมื่อเล่นเกมผ่านไป 1 ชั่วโมง

รูปที่ 2 ได้เงินอยู่ที่ 10085 ได้เงินเพิ่มขึ้นมา 6573



รูปที่ 4.32 ทดสอบประสิทธิภาพหาเงินในเกม 1 ชั่วโมง ด้วยการใช้บอทเล่น

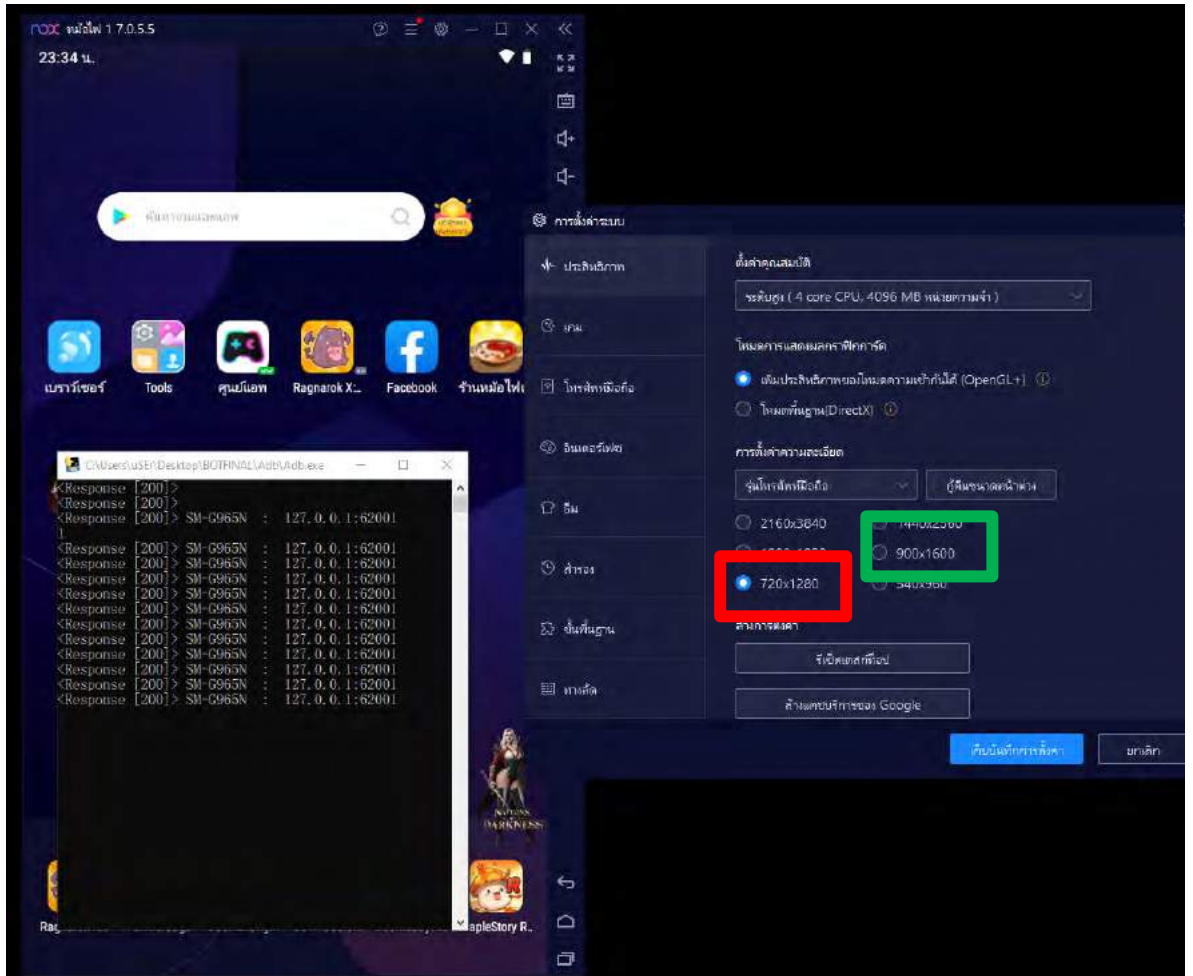
จากการทดสอบการเล่นเกมนับด้วยการใช้บอทเล่น 1 ชั่วโมง

รูปที่ 1 ค่ายซ้ายมือเริ่มเล่นเกมโดยมีเงินในเกมอยู่ที่ 1985 เมื่อเล่นเกมผ่านไป 1 ชั่วโมง

รูปที่ 2 ได้เงินอยู่ที่ 11624 ได้เงินเพิ่มขึ้นมา 9639

สรุปจากการทดลองรูปที่ 4.31 และ 4.32 สรุปได้ว่าการใช้บอทเล่นทำให้ได้ผลตอบแทนมากกว่าการเล่นเองด้วยมือ เพราะบอทสามารถทำงานได้ตลอดเวลา และทำงานได้ไวเช่นการเจอร์รูปที่เหมือนกันบอทก็จะทำการกดให้ในทีเดียว แต่การเล่นด้วยมือนั้นต้องกดทีละรูปเลยทำให้ช้ากว่าการใช้บอท

#### 4.27 ทดสอบการใช้บอทกับโปรแกรมจำลองที่ขนาดไม่ตรงกับโปรแกรมบอท



รูปที่ 4.33 ทดสอบการใช้บอทกับโปรแกรมจำลองที่ขนาดไม่ตรงกับโปรแกรมบอท

จากการทดสอบพบว่าเมื่อเปิดโปรแกรมบอท บอทสามารถหาโปรแกรมจำลองเจอแต่ไม่สามารถใช้งานได้ เนื่องจากขนาดที่กำหนดไว้ตอนเขียนโปรแกรมบอทเขียนที่ขนาด 900X1600 เมื่อนำโปรแกรมบอทมาเปิดใช้กับขนาดหน้าจออื่นที่ไม่ใช่ 900X1600 เลยไม่สามารถใช้งานได้

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

การจัดทำโครงการเรื่อง บอทคลิกอัตโนมัติผ่านอิมูเลเตอร์เกมร้านหม้อไฟแห่งความสุข  
สรุปการดำเนินงาน ปัญหา และข้อเสนอแนะในการทำงานดังต่อไปนี้

#### 5.1 สรุปผลการดำเนินโครงการ

การดำเนินงานในโครงการบอทคลิกอัตโนมัติผ่านอิมูเลเตอร์เกมร้านหม้อไฟแห่งความสุข  
สรุปผลการทดลองการใช้งานได้ดังนี้

- 5.1.1 ตัวบอทสามารถเชื่อมต่อกับตัว adb และ Emuretor ได้
- 5.1.2 บอทสามารถทำงานได้ปกติตั้งแต่กดเข้าเกมไปจนถึงการทำงานภายในเกมพร้อมทั้งการรีเซ็ตเกม
- 5.1.3 ในส่วนของการทำงานหลายหน้าจอ การให้บอททำงานแบบทัชจอ และการไม่ใช้เมาส์ร่วมกับผู้เล่นก็สามารถทำงานได้อย่างปกติและใช้ได้จริง
- 5.1.4 บอทสามารถส่งรูปภาพและขั้นตอนการทำงานต่างๆที่กำลังทำอยู่เข้าไลน์ได้

#### 5.2 ปัญหาการดำเนินโครงการ

- 5.2.1 ตัวบอทยังมีการทำงานที่ไม่เร็วพอบางที่ถึงจุดที่ต้องกดแล้วแต่เหมือนตัวบอทจะจับภาพได้ช้า
- 5.2.2 ตัวบอทสามารถใช้ได้กับขนาดหน้าจอที่กำหนดไว้เท่านั้นหากใครนำไปใช้แล้วจอเล็กกว่าหรือใหญ่กว่าที่กำหนดก็ไม่สามารถใช้ได้

#### 5.3 ข้อเสนอแนะ

- 5.3.1 บอทควรสามารถทำงานขนาดหน้าจอเท่าไรก็ได้
- 5.3.2 บอทควรที่จะสามารถทำงานได้เร็วกว่านี้
- 5.3.3 ถ้ามีการนำไปต่อยอดหรือมีการนำไปพัฒนาต่อควรทำเพิ่มในส่วนของการรันบอทแบบไม่เปิดจอเกมหรือสร้างหน้าต่าง UI และเพิ่มฟังก์ชันการทำงานเพิ่ม

## บรรณานุกรม

Amazon. (2566). *Python คืออะไร*. เข้าถึงได้จาก

<https://aws.amazon.com/th/what-is/python/>

BorntoDev School. (2562). *Amazing Bot with Autopy*. เข้าถึงได้จาก

<https://academy.borntodev.com/courses/429064/lectures/6590806>

DOANYTECH. (2565). *เชื่อมต่อ adb กับ อุปกรณ์ Android*. เข้าถึงได้จาก

<http://www.doanytech.com>

Frank Marshall. (2565). *Auto Clicker คืออะไร*. เข้าถึงได้จาก

<https://ogautoclicker.com>

Python Software Foundation. (2551). *Processing 0.52*. เข้าถึงได้จาก

<https://pypi.org/project/processing/>

Python Software Foundation. (2566). *Auto-py-to-exe 2.33.0*. เข้าถึงได้จาก

<https://pypi.org/project/auto-py-to-exe/>

ULTIMATE PYTHON. (2564). *สร้างบอททำงานแทน ใน 8 คำสั่ง*. เข้าถึงได้จาก

<https://www.ultimatepython.co/post/control-mouse-keyboard-pyautogui-python>

Yugioh2500. (2561). *รีวิว NoxPlayer โปรแกรมจำลอง Android บนคอมพิวเตอร์ PC*. เข้าถึงได้จาก

<https://ireview.in.th/review-noxplayer-android-pc/>



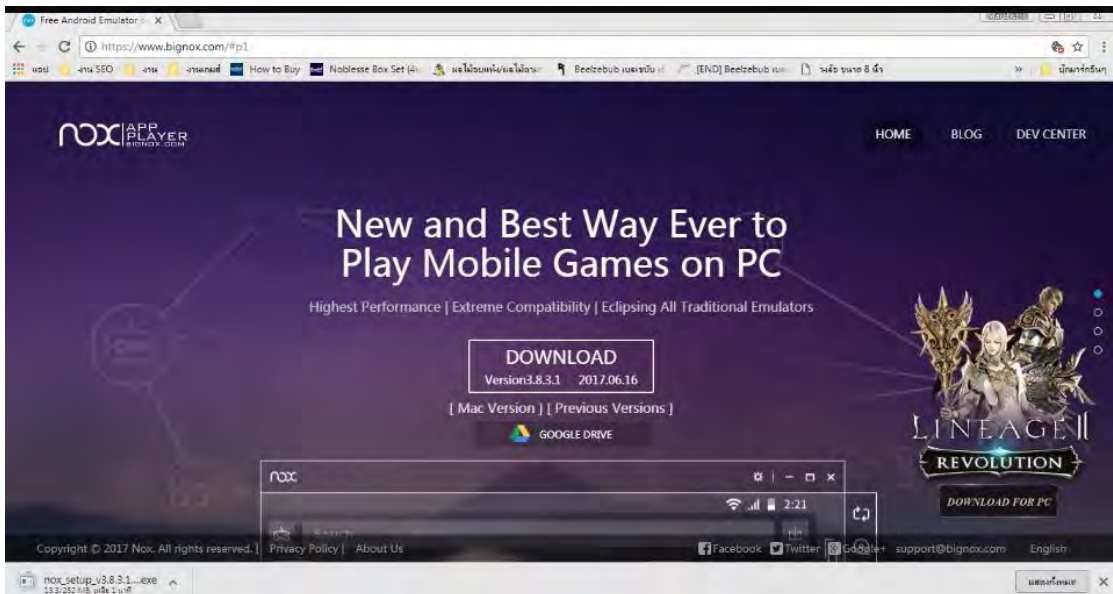
**ภาคผนวก**





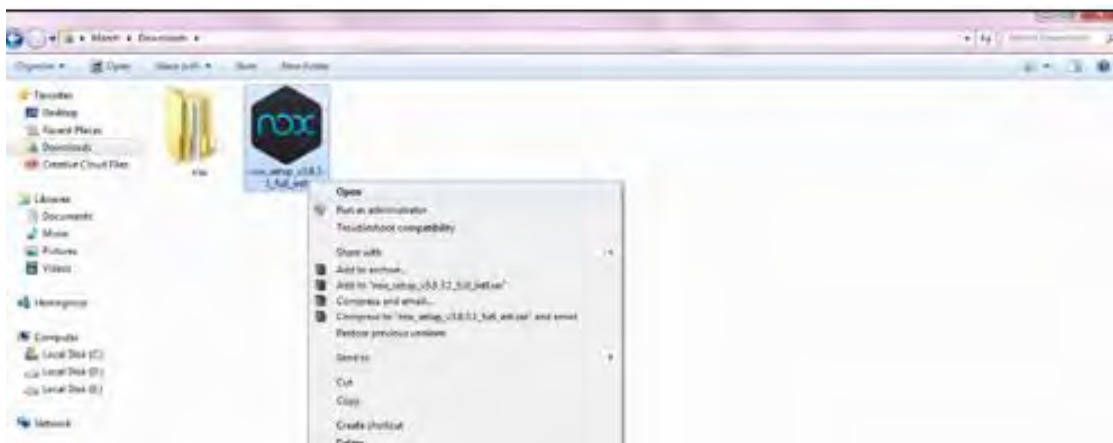
## ขั้นตอนการติดตั้ง โปรแกรมจำลอง

ขั้นที่ 1 เข้าเว็บไซต์โปรแกรม Nox Player แล้วกดดาวน์โหลดได้เลย การโหลดโดยตรงจากเว็บไซต์หลักจะช่วยป้องกันการติดไวรัส และได้โปรแกรมในเวอร์ชันล่าสุด



รูปที่ ก.1 การติดตั้งโปรแกรมจำลอง ขั้นที่ 1

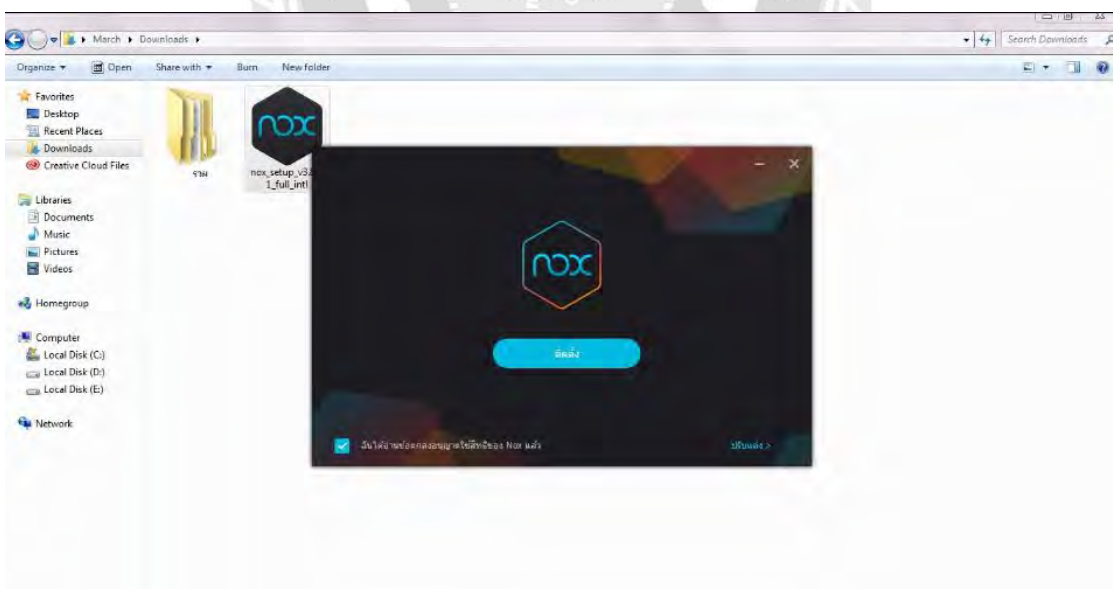
ขั้นที่ 2 เมื่อดาวน์โหลดเรียบร้อยแล้วให้กดไปหน้าที่ใช้เก็บไฟล์ กด Open จากนั้นกดรันแอดมินรอนจนขึ้นหน้าต่างติดตั้งของโปรแกรม Nox จากนั้นคลิกตรงช่อง นั้นได้อ่านข้อตกลงอนุญาตการใช้สิทธิของ Nox แล้ว กดติดตั้ง



รูปที่ ก.2 การติดตั้งโปรแกรมจำลอง ขั้นที่ 2

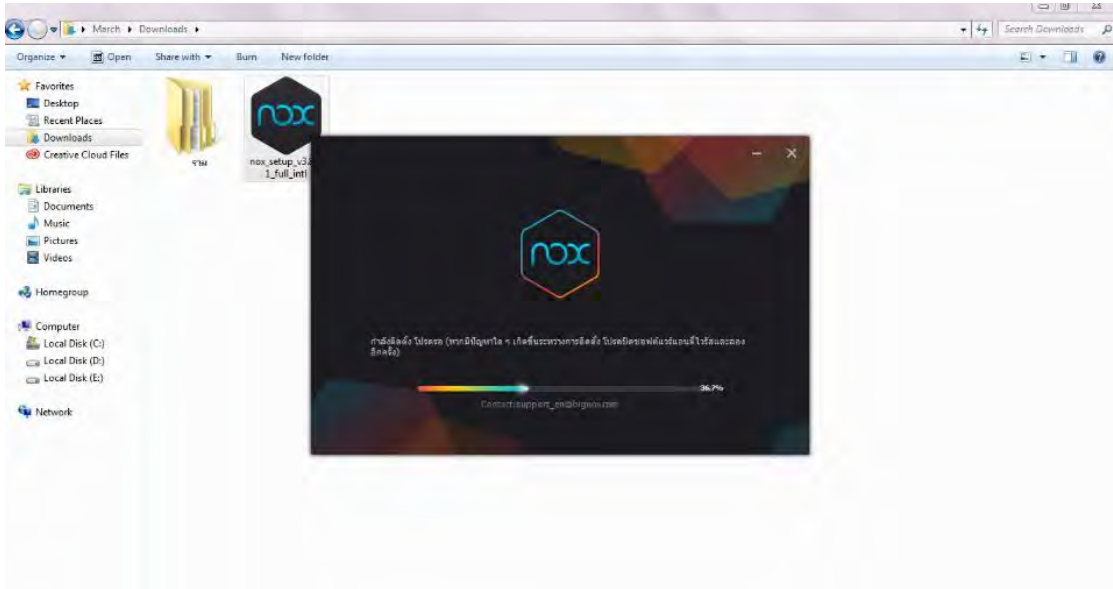


รูปที่ ก.3 การติดตั้งโปรแกรมจำลอง ขั้นที่ 2.1

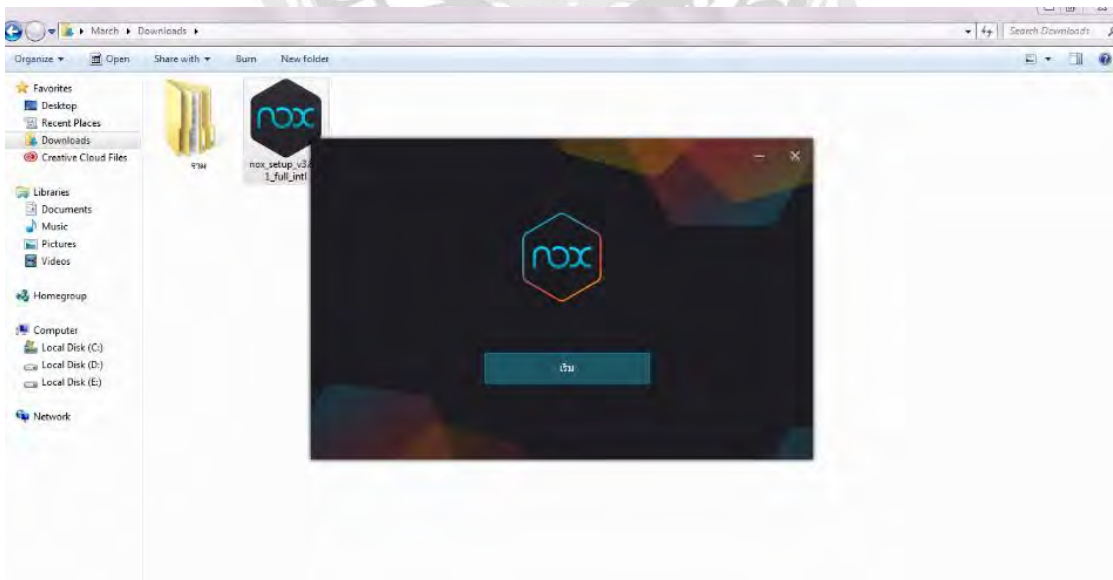


รูปที่ ก.4 การติดตั้งโปรแกรมจำลอง ขั้นที่ 2.2

ขั้นที่ 3 รองนกว่าขั้นตอนการติดตั้งจะเสร็จเรียบร้อยซึ่งใช้เวลาไม่นานประมาณ 2-3 นาทีเท่านั้น เมื่อติดตั้งเสร็จกดปุ่มเริ่มได้เลย

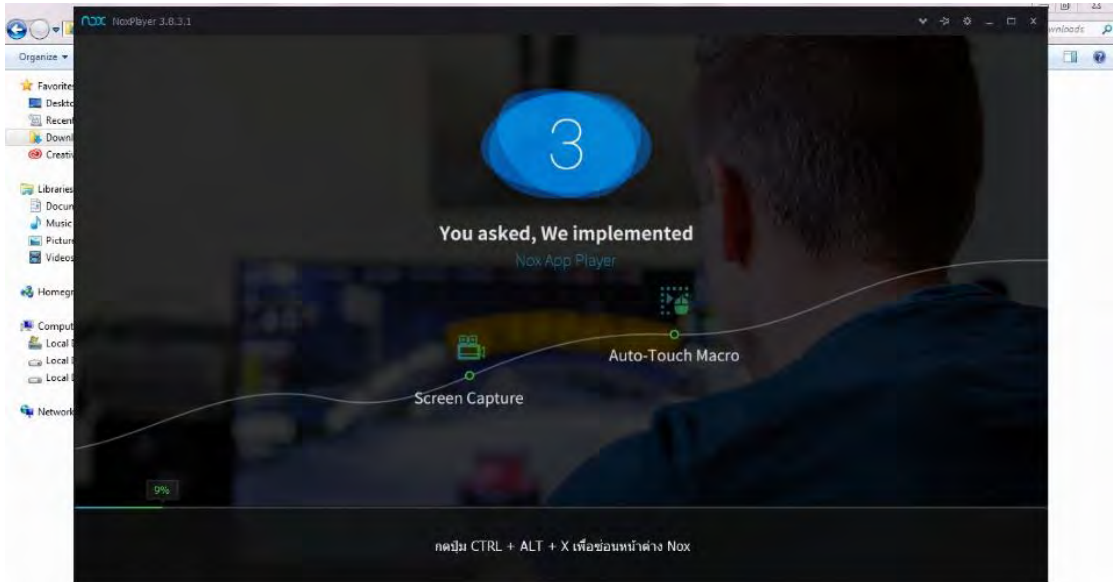


รูปที่ ก.5 การติดตั้งโปรแกรมจำลอง ขั้นที่ 3



รูปที่ ก.6 การติดตั้ง โปรแกรมจำลอง ขั้นที่ 3.1

ขั้นที่ 4 รอโปรแกรม Nox เปิดใช้งานอย่างเต็มรูปแบบ เมื่อเข้าสู่หน้าโปรแกรมเรียบร้อยแล้ว จะมีสอนเบสิกการใช้งาน กด Next ไปได้เลย



รูปที่ ก.7 การติดตั้ง โปรแกรมจำลอง ขั้นที่ 4



รูปที่ ก.8 การติดตั้ง โปรแกรมจำลอง ขั้นที่ 4.1



ขั้นที่ 5 จากนั้นก็จะเข้าสู่หน้าต่างปกติของโปรแกรม Nox สามารถเริ่มต้นใช้งานได้เลย



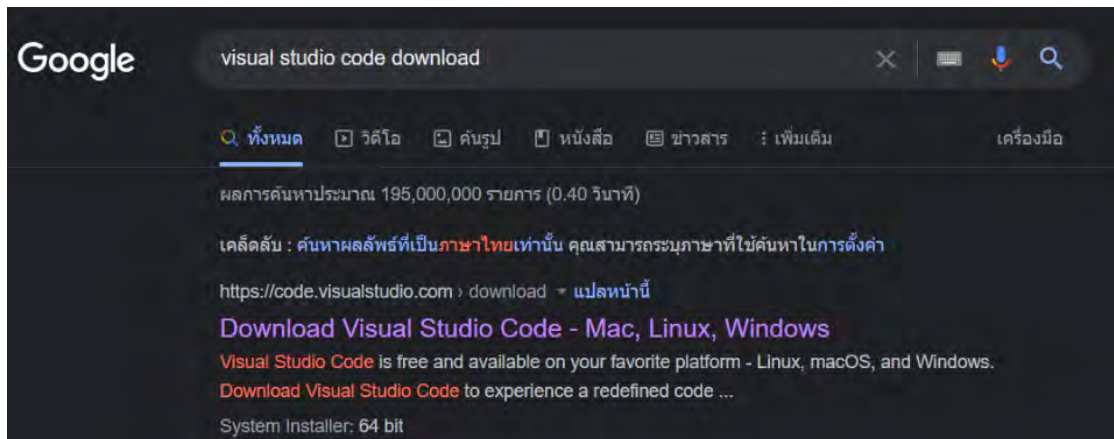
รูปที่ ก.9 การติดตั้งโปรแกรมจำลอง ขั้นที่ 5





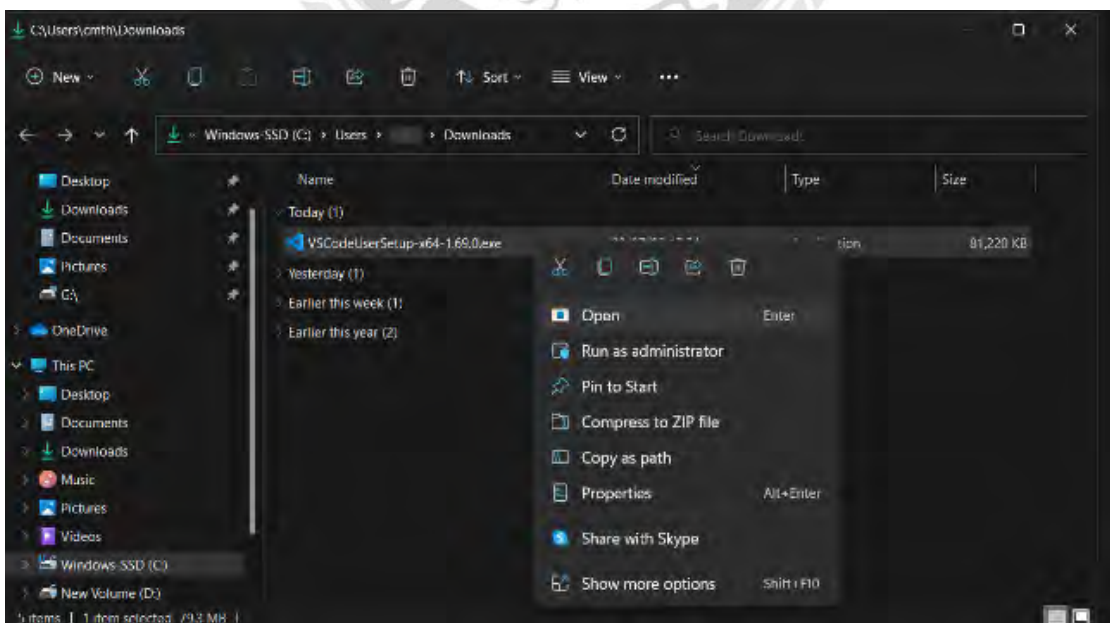
## ขั้นตอนการติดตั้ง Visual Studio Code

ขั้นที่ 1 เปิดเว็บเบราว์เซอร์ที่เราใช้งานอยู่ เช่น Chrome หรืออย่างอื่นแล้วค้นหาคำว่า visual studio code download จากนั้นให้คลิกเข้าไปที่ชื่อตามรูปภาพ หรือจะคลิกที่ลิงก์ Download Visual Studio Code - Mac, Linux, Windows นี้เพื่อดาวน์โหลดเลยก็ได้



รูปที่ ก.10 การติดตั้ง Visual Studio Code ขั้นที่ 1

ขั้นที่ 2 คลิกขวาที่ไฟล์ VSCodeUserSetup-x64-1.xx.x.exe แล้วเลือก Open หรือจะดับเบิลคลิกเข้าไปเลยก็ได้



รูปที่ ก.11 การติดตั้ง Visual Studio Code ขั้นที่ 2

ขั้นที่ 3 มี 7 ขั้นตอนย่อย

3.1 I accept the agreement และคลิก Next

3.2 คลิก Next

3.3 คลิก Next

3.4 ในส่วนนี้ผมขอแนะนำให้เลือก  ทุกช่องเพื่อให้การติดตั้งเข้ากับโค้ดในระบบปฏิบัติการของเราอย่างสมบูรณ์

Additional icon:

Create a desktop icon

Other:

Add "Open with Code" action to Windows Explorer file context menu

Add "Open with Code" action to Windows Explorer directory context menu

Register Code as an editor for supported file types

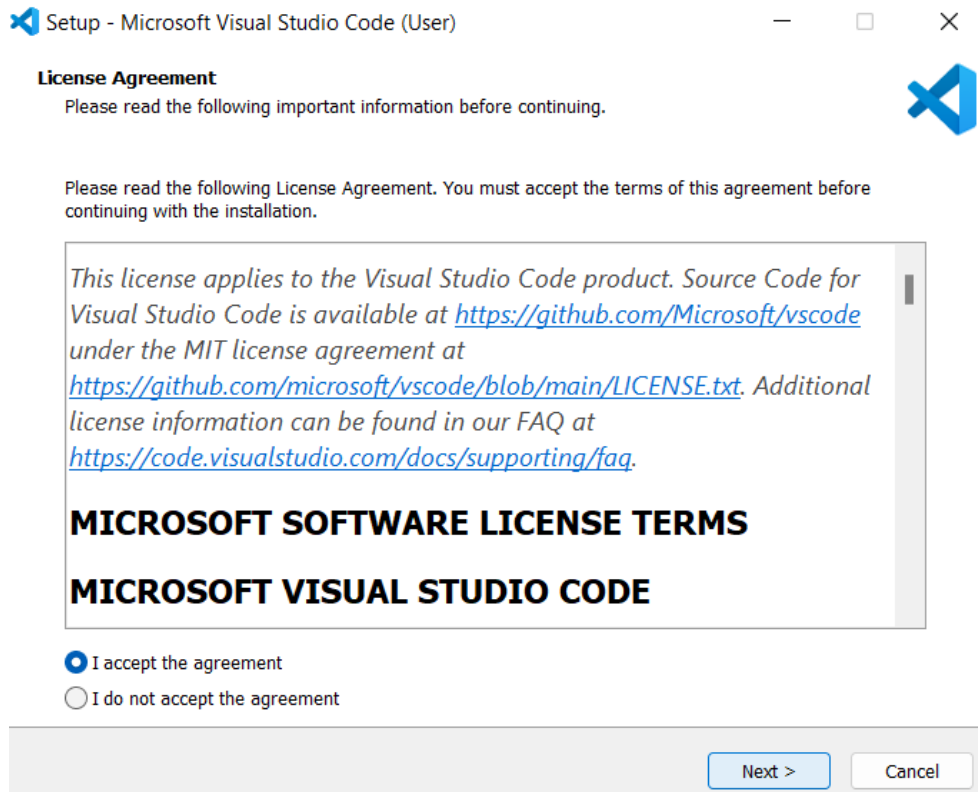
Add to PATH (requires shell restart)

จากนั้นคลิก Next

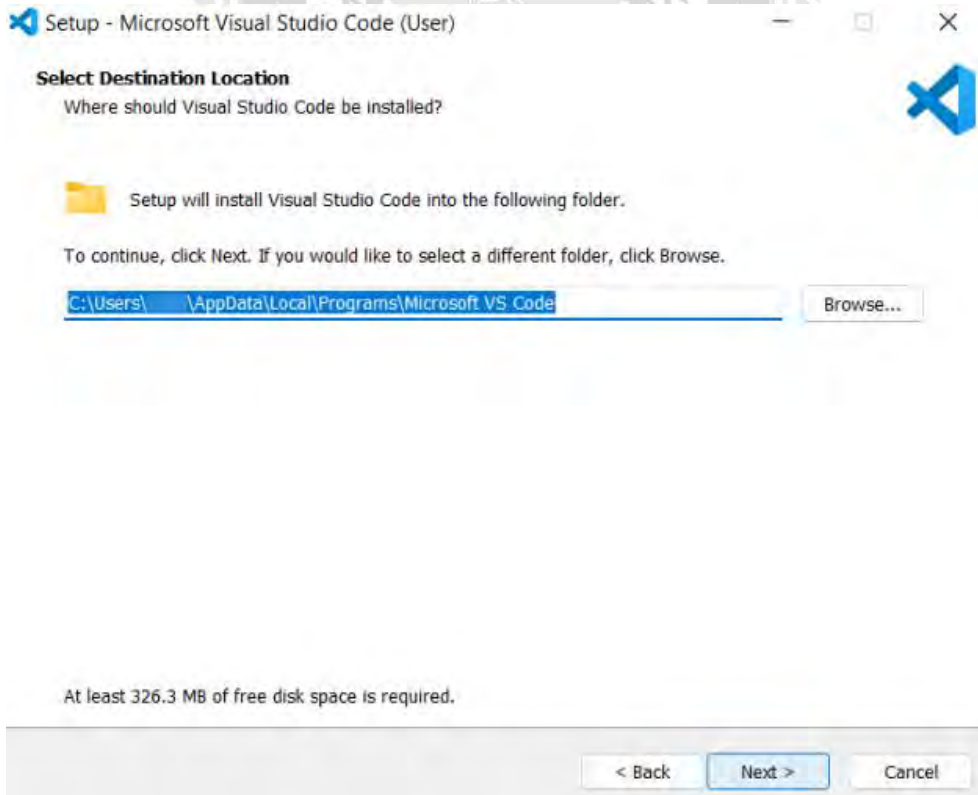
3.5 คลิก Install

3.6 รอการติดตั้งโปรแกรม

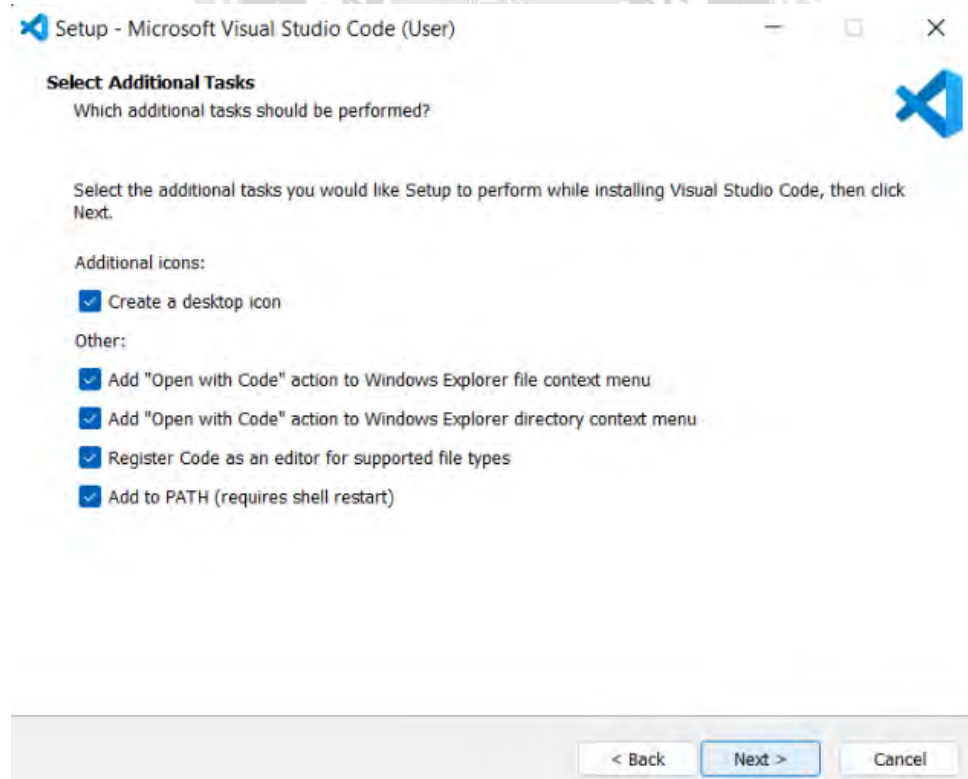
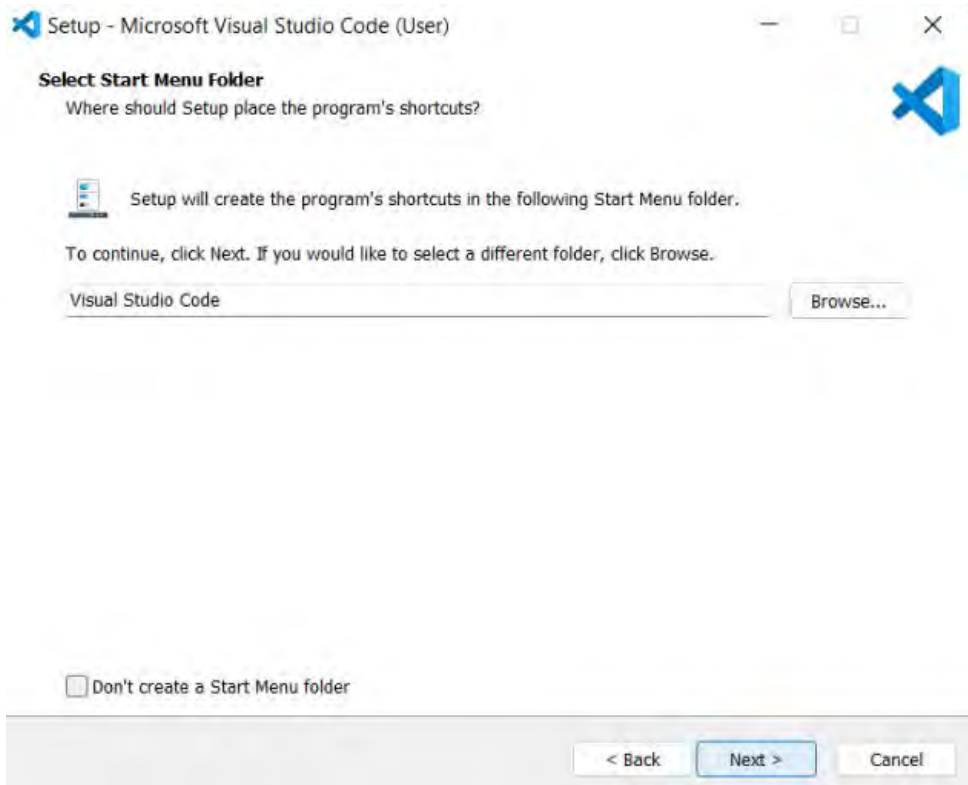
3.7 คลิก Finish

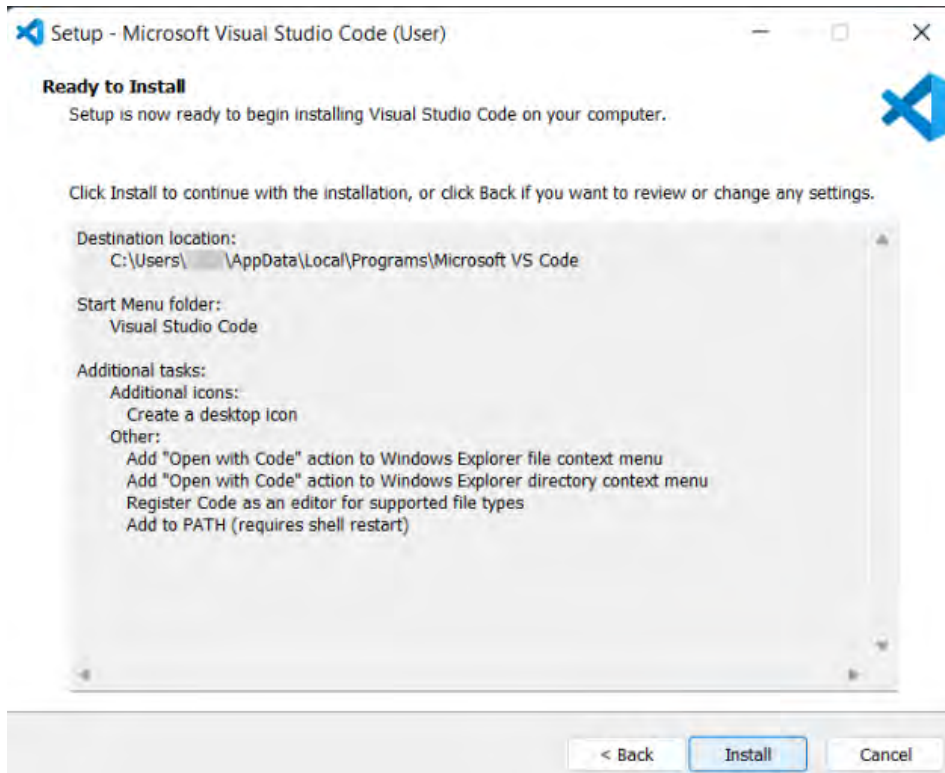


รูปที่ ก.12 การติดตั้ง Visual Studio Code รุ่นที่ 3.1

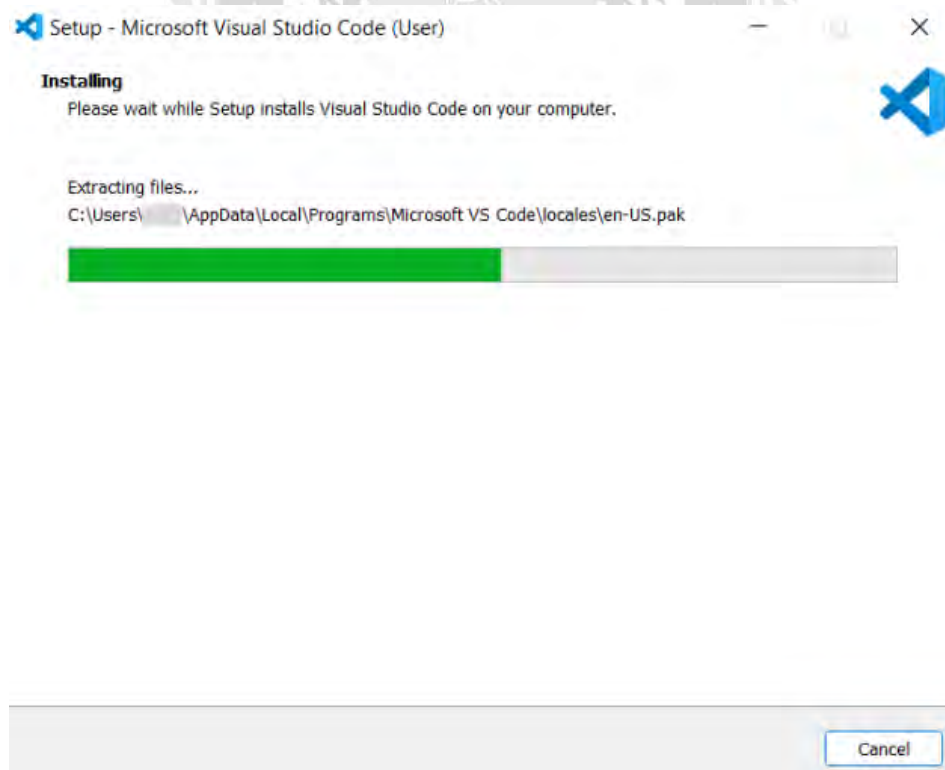


รูปที่ ก.13 การติดตั้ง Visual Studio Code รุ่นที่ 3.2

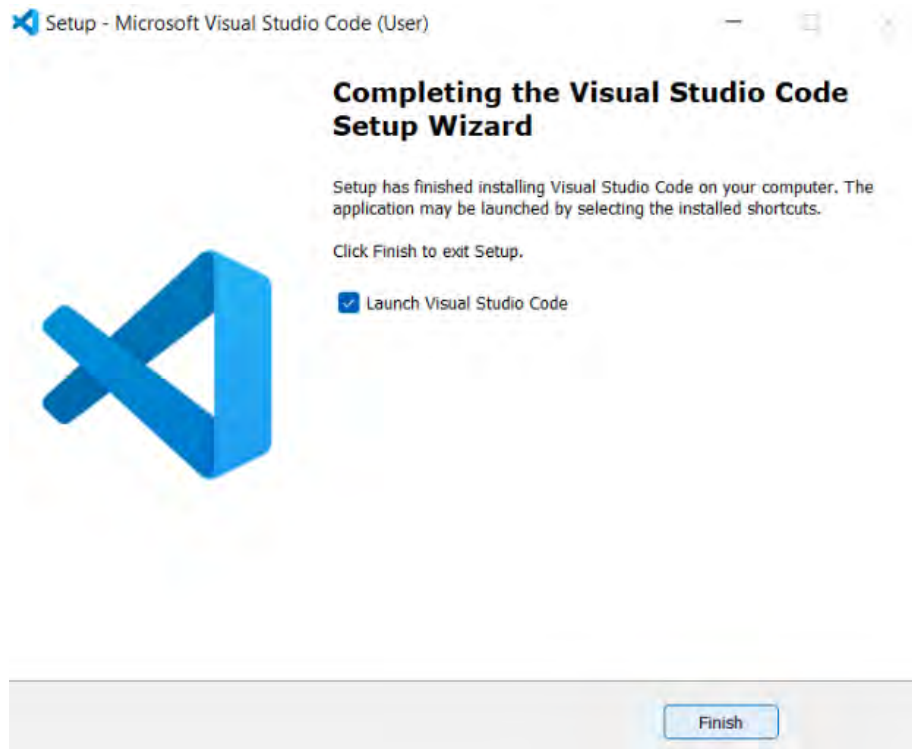




รูปที่ ก.16 การติดตั้ง Visual Studio Code รุ่นที่ 3.5

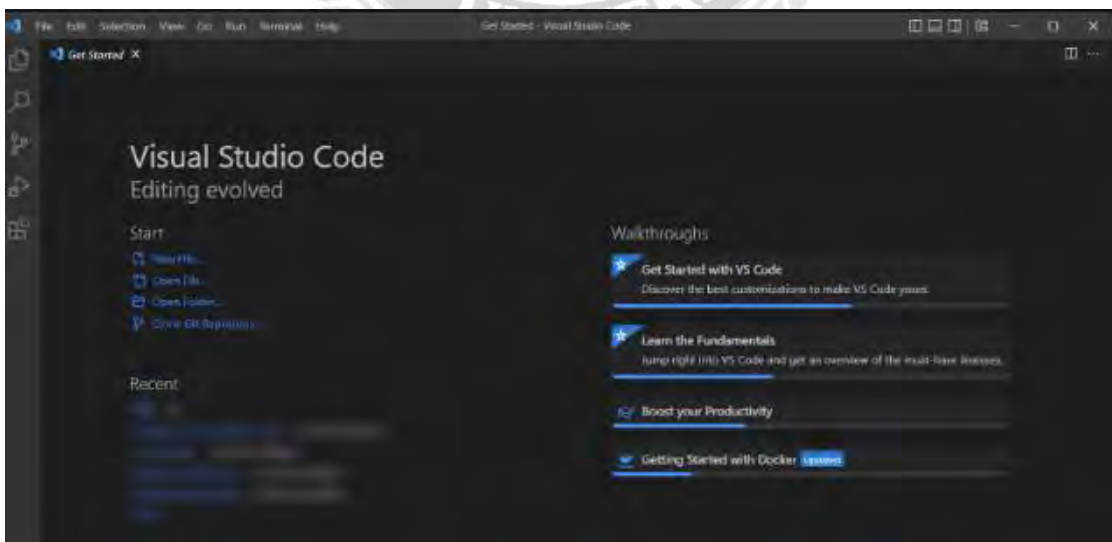


รูปที่ ก.17 การติดตั้ง Visual Studio Code รุ่นที่ 3.6



รูปที่ ก.18 การติดตั้ง Visual Studio Code ขั้นที่ 3.7

ขั้นที่ 4 เมื่อแสดงหน้าจอแบบนี้แล้ว ก็ถือว่าการติดตั้งโปรแกรม Visual Studio Code ของเราเสร็จสมบูรณ์แล้ว



รูปที่ ก.19 การติดตั้ง Visual Studio Code ขั้นที่ 4



## ประวัติผู้จัดทำ



ชื่อ-สกุล : นาย สรวิชญ์ สุขผล  
รหัสนักศึกษา : 6204000012  
ภาควิชา : วิศวกรรมคอมพิวเตอร์  
คณะ : วิศวกรรมศาสตร์  
ที่อยู่ปัจจุบัน : 172/5 หมู่5 ตำบลอ้อมน้อย  
อำเภอกระทุ่มแบน  
จังหวัดสมุทรสาคร 74130



ชื่อ-สกุล : นาย วิทยา จันพรม  
รหัสนักศึกษา : 6323000002  
ภาควิชา : วิศวกรรมคอมพิวเตอร์  
คณะ : วิศวกรรมศาสตร์  
ที่อยู่ปัจจุบัน : 120 ซอยวัดกำแพง ถนนพระราม 2  
แขวงแสมดำ เขตบางขุนเทียน  
จังหวัดกรุงเทพมหานคร 10150

