



การพัฒนาวิทยาศาสตร์และเทคโนโลยี เพื่อเตรียมการสู่อนาคต



ผู้ช่วยศาสตราจารย์ ดร.สันติ นุราษ

Asst.Prof.Dr.Santi Nuratch, Ph.D.

santi.inc.kmutt@gmail.com

Embedded Computing and Control Laboratory
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี KMUTT

EEC

ระเบียงเศรษฐกิจพิเศษภาคตะวันออก
เชื่อมโลก ให้ไทยแล่น



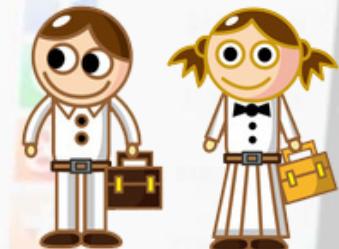
Eastern Economic Corridor (EEC)



Special Economic Zone (SEZ) for the Future of ASEAN
เขตพื้นที่เศรษฐกิจพิเศษ ภาคตะวันออก เพื่ออนาคตของอาเซียน

โครงการ EEC มีเป้าหมายยกระดับพื้นที่เขตเศรษฐกิจภาคตะวันออกให้
กลายเป็น “World-Class Economic Zone” รองรับการลงทุน
อุตสาหกรรม Super Cluster และอุตสาหกรรมเป้าหมายของประเทศไทย เพื่อ
เป็นกลไกขับเคลื่อนเศรษฐกิจในอีก 20 ปีข้างหน้า

EEC



ระเบียงเศรษฐกิจพิเศษภาคตะวันออก
เชื่อมโลก ให้ไทยแล่น

Who am I?, What are my research fields?



ผู้ช่วยศาสตราจารย์ ดร.สันติ นุราษ

Asst.Prof.Dr.Santi Nuratch, Ph.D.

santi.inc.kmutt@gmail.com

Embedded Computing and Control Laboratory (ECC Lab.)

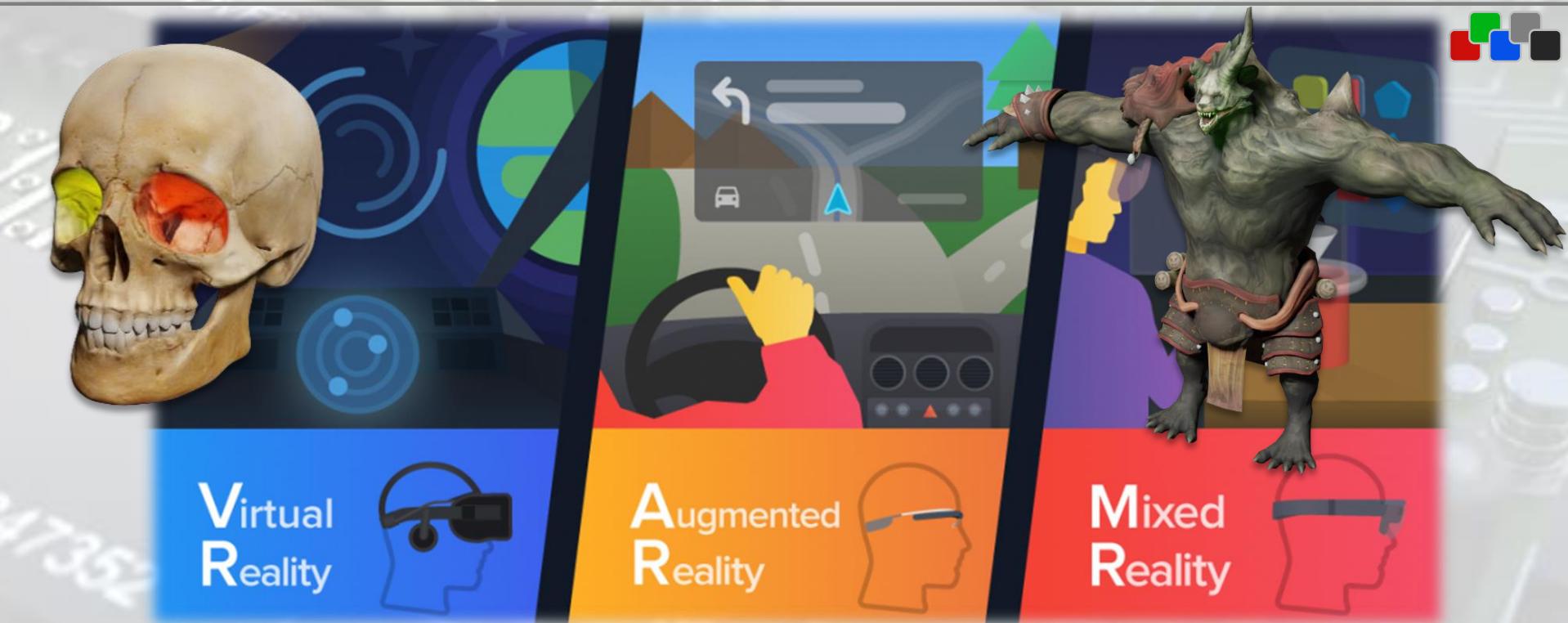
Department of Control System and Instrumentation Engineering (INC)

King Mongkut's University of Technology Thonburi (KMUTT), THAILAND

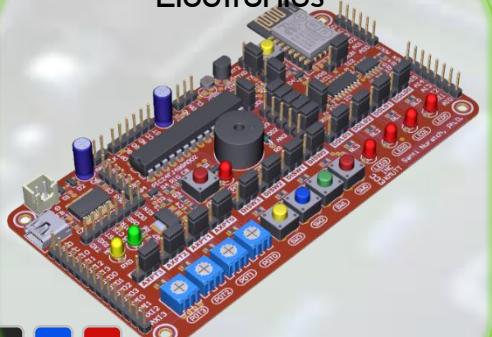


- ▶ Embedded Systems, Real-time Systems, Wireless Control and Monitoring
- ▶ Digital Signal Processing, Real-time Signal Processing, Active Filter
- ▶ Speech Signal Processing, Speech Recognition, Speaker Recognition, Speech Enhancement
- ▶ Artificial Intelligence, Machine Learning, Deep Learning, Pattern Recognition
- ▶ System Dynamics (Modelling and Control)
- ▶ Power Electronics (Modelling and Digital Control, Digital and Analog Circuits)
- ▶ Industrial Automation and Instrumentation Systems
- ▶ Internet of Things (IoT), Industry 4.0, VR, AR, MR, and Computer Technologies
- ▶ Industrial Software Design and Implementation (Embedded, Com/Web-based applications)

สิ่งที่จะได้เรียนรู้กันในวันที่ 1 และวันที่ 2



Electronics



Microcontroller



User Interface



Programming

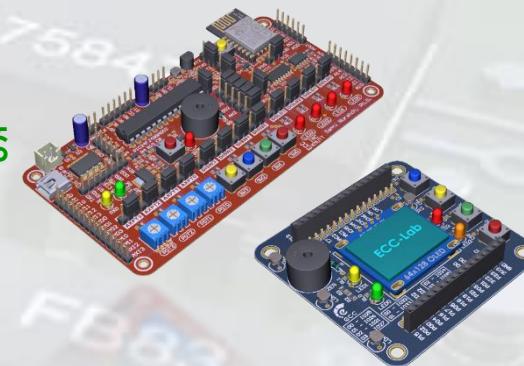




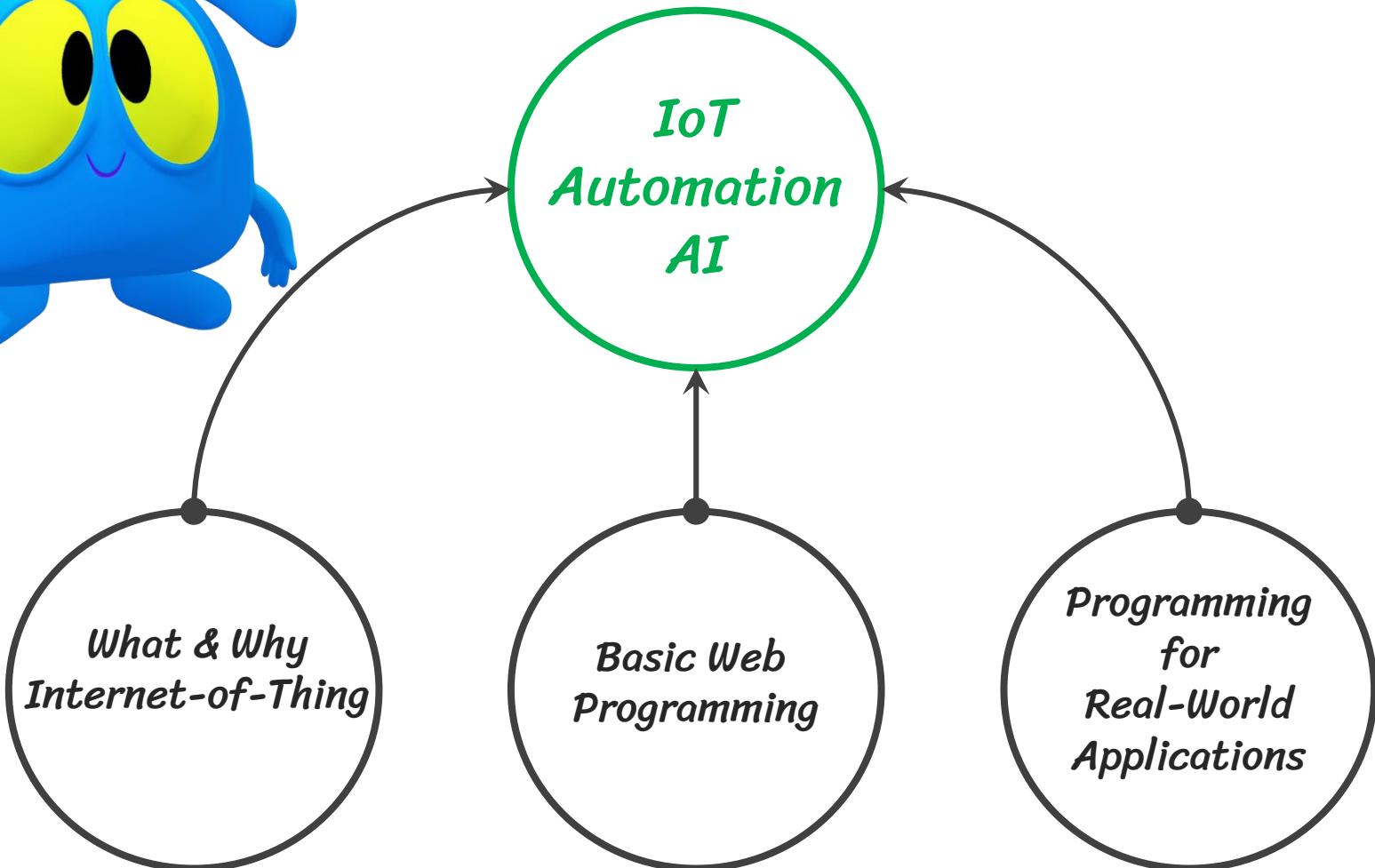
รู้จัก เข้าใจ ประยุกต์ ต่อยอด

เทคโนโลยีใหม่ เอไอ วิوار์ เอเวอร์ เกมส์ แอนด์ เมมชั่น พร้อมการประยุกต์ใช้งานในปัจจุบันและอนาคต

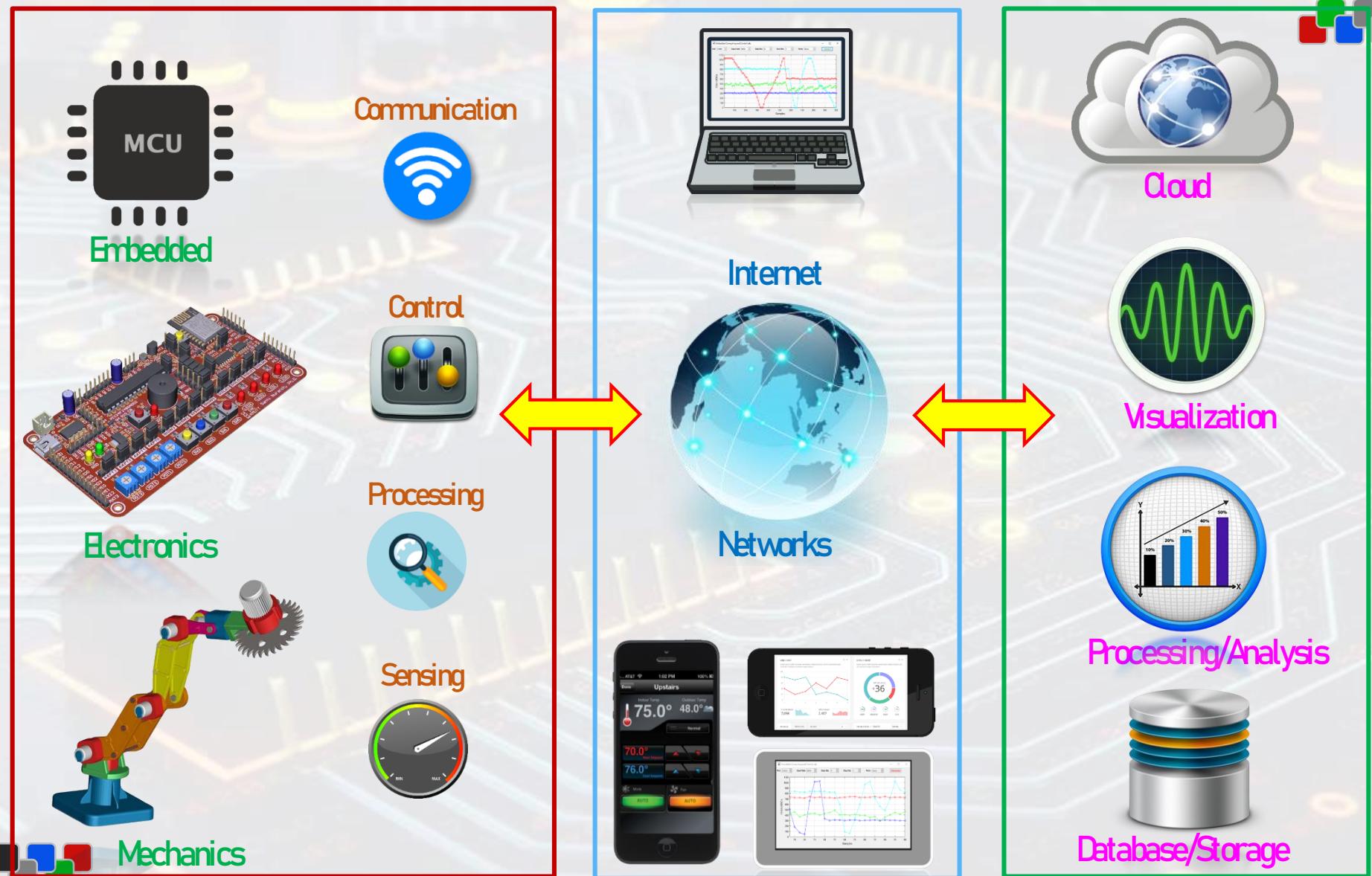
- ไมโครคอนโทรลเลอร์และระบบอินเตอร์เฟซต่อฟรีซ์
- การเขียนโปรแกรมควบคุมพิวเตอร์ควบคุมการทำงานของไมโครคอนโทรลเลอร์
- การเขียนโปรแกรมควบคุมพิวเตอร์ควบคุมระบบอินเตอร์เฟซต่อฟรีซ์
- การออกแบบยูสเซอร์อินเตอร์เฟส (UI) สำหรับระบบอินเตอร์เฟซต่อฟรีซ์
- ควบคุมพิวเตอร์กราฟิกส์แบบ 2 และ 3 มิติ
- การเขียนโปรแกรมควบคุมการแสดงผลในระบบ 3 มิติ
- การหล่อรวมของควบคุมพิวเตอร์กราฟิกส์และฟีลิกส์
- รู้จักกับระบบวิوار์ เอเวอร์ และการประยุกต์ใช้งาน
- การเขียนโปรแกรมสร้างเกมส์และแอปพลิเคชัน
- การเขียนโปรแกรมควบคุมหุ่นยนต์และระบบอัตโนมัติ
- เอไอ และการประยุกต์ใช้งานในปัจจุบันและอนาคต



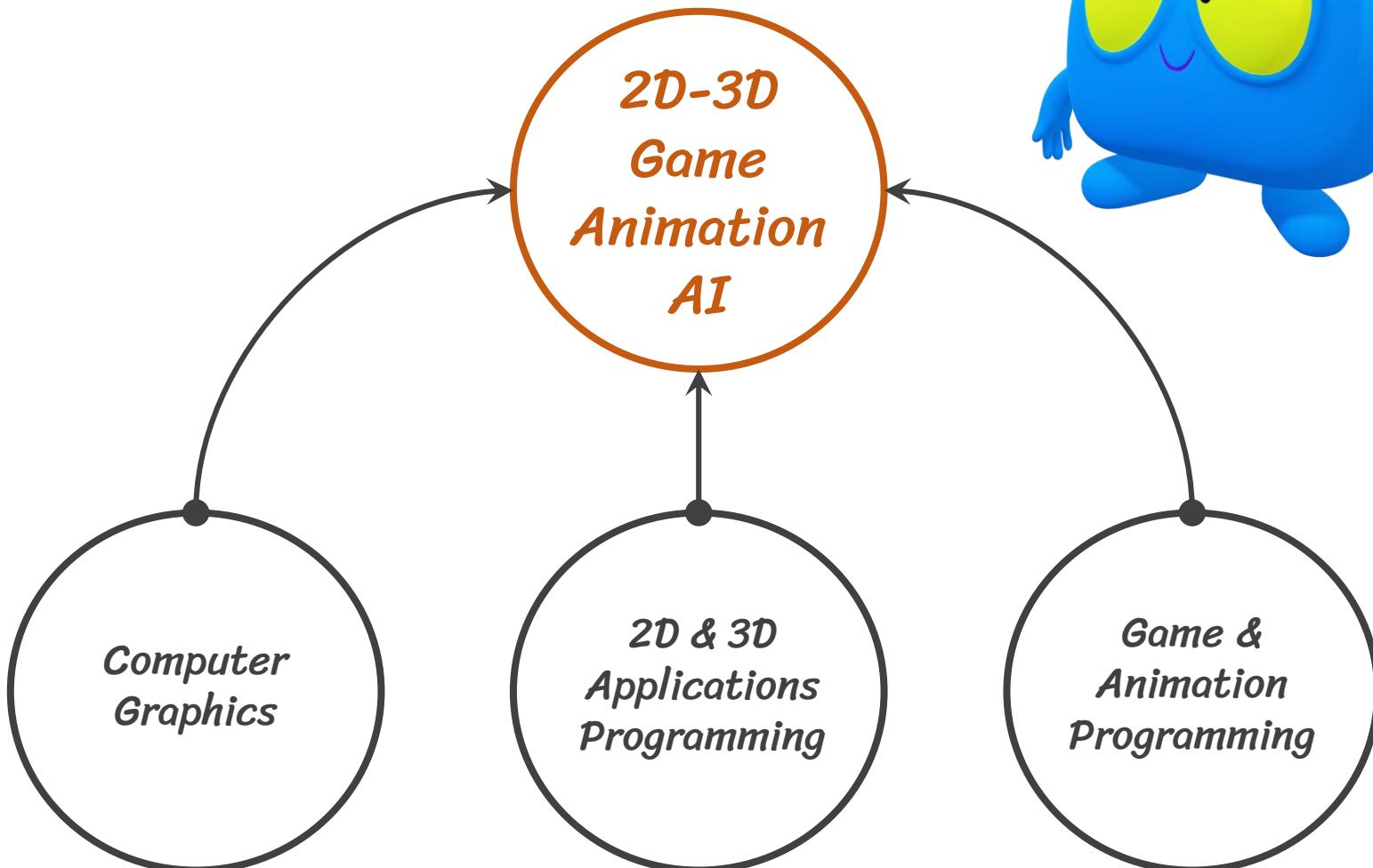
Day #1



อิเล็กทรอนิกส์ อินเทอร์เน็ตอອฟริงส์ และการเขียนโปรแกรม



Day #2



ເກມສໍ ແລ້ວນິເນ້ນ ແລະ ການເຂີຍບໂປຣແກຣມ



เครื่องมือที่ใช้ในการเรียนรู้

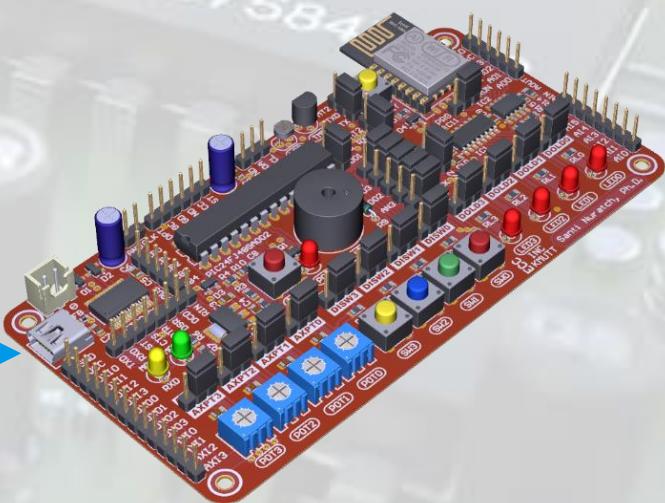


คอมพิวเตอร์ (Windows)



- พัฒนาโปรแกรม
- ดูผลการทำงานของโปรแกรม
- เว็บเซิร์ฟเวอร์
- บอร์ดเซิร์ฟเวอร์/ไ/drเวอร์
- อื่นๆ

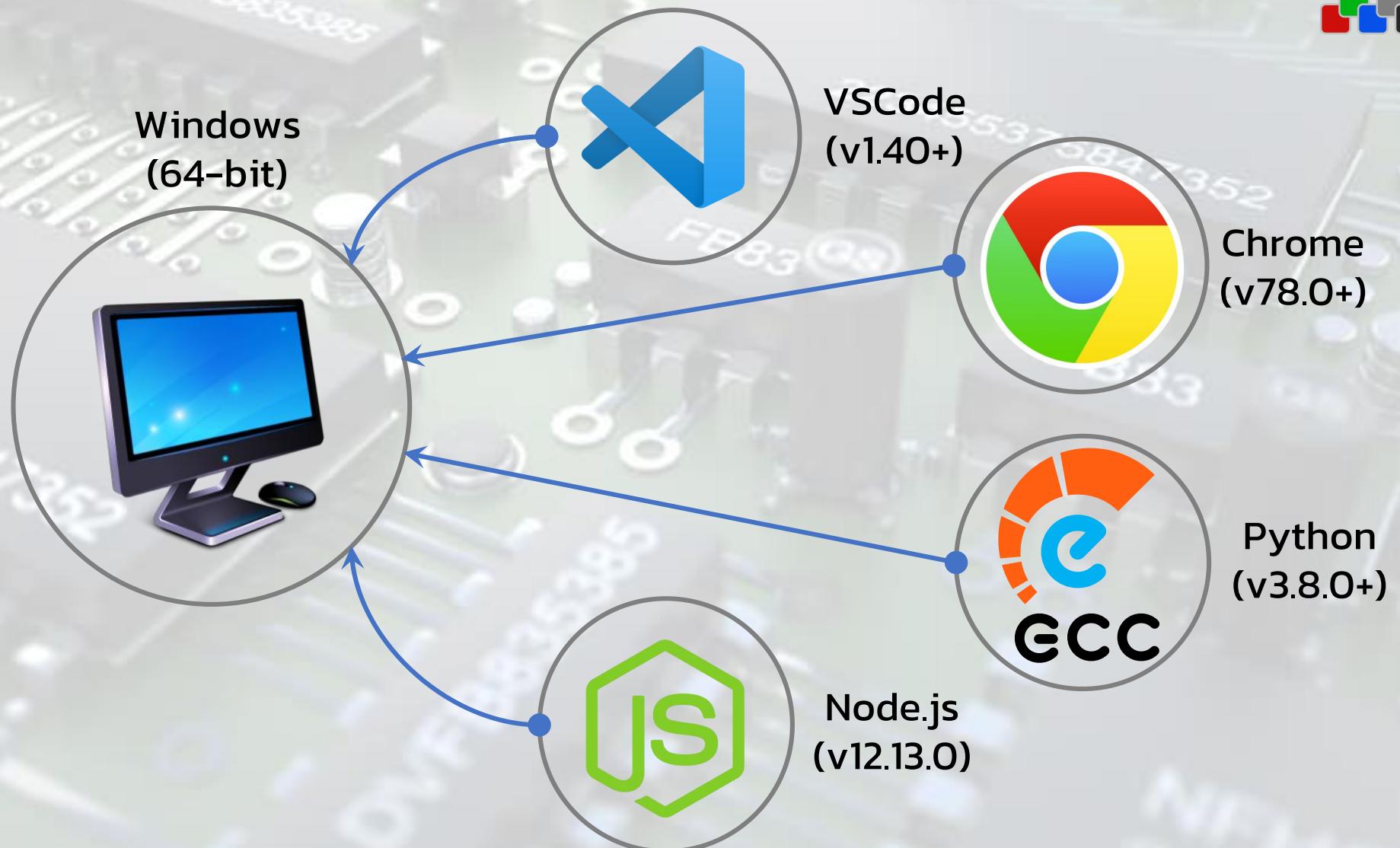
บอร์ดไมโครคอนโทรลเลอร์



- หลอดแอลอีดี (แสง)
- สวิตซ์ (ตรวจจับการกด)
- แอลเดอาร์ (ตัวรับแสง)
- บัซเชอร์ (กำเนิดเสียง)
- อื่นๆ



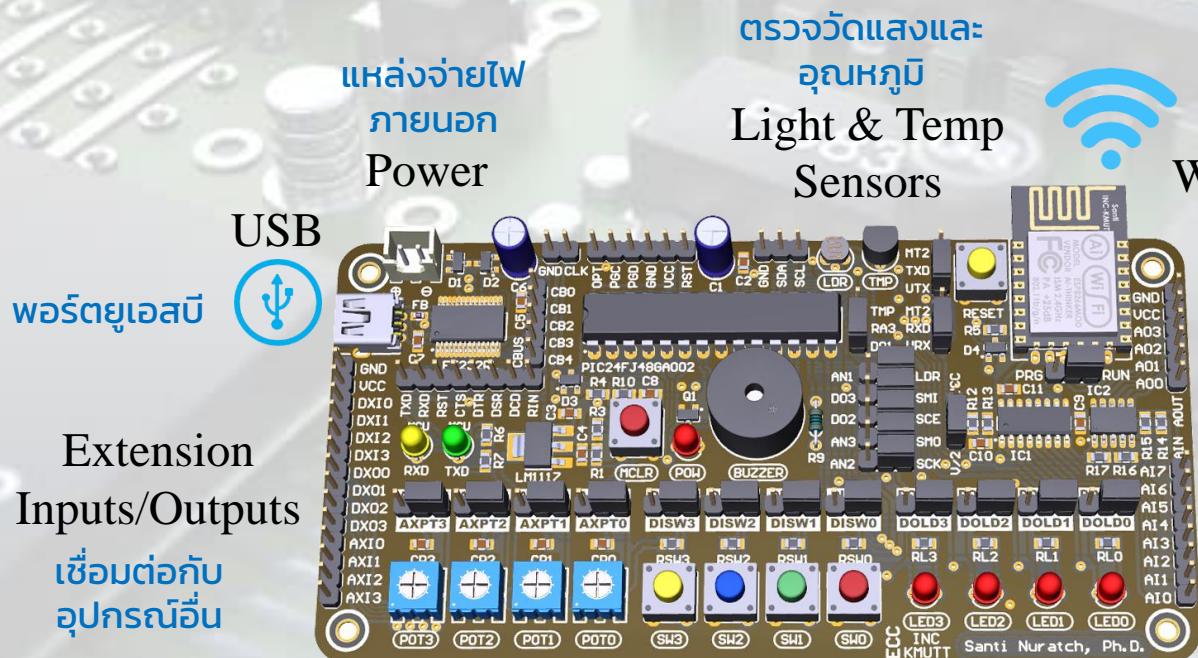
ซอฟแวร์ที่ใช้ในการพัฒนาโปรแกรม



ฮาร์ดแวร์ (บอร์ดไมโครคอนโทรลเลอร์)



บอร์ดไมโครคอนโทรลเลอร์ที่ออกแบบมาสำหรับการเรียนการสอนด้านสมองกล ผังตัว และการเขียนโปรแกรมสำหรับระบบอัตโนมัติ รวมไปถึงระบบอินเทอร์เน็ต ซอฟต์แวร์



Extention
Inputs/Outputs
เชื่อมต่อภายนอก
อุปกรณ์อื่น

Analog
Inputs

Digital
Inputs

Digital
Outputs

ตัวต้านทานประค่า
ใช้ปรับสัญญาณไฟฟ้า

สวิตซ์แบบ
กดติดปล่อยดับ

หลอดแอลอีดี

เชื่อมต่อระบบ
อินเทอร์เน็ต

WiFi

Analog
Outputs

ส่งสัญญาณไฟฟ้า
ให้อุปกรณ์อื่น

Analog
Inputs

รับสัญญาณไฟฟ้า
จากอุปกรณ์อื่น



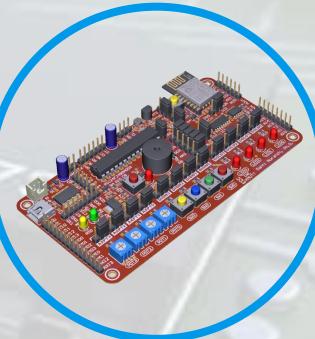
ระบบที่เราจะได้เรียนกัน



เว็บบราวเซอร์



เซิร์ฟเวอร์



ไดร์เวอร์

ไมโครคอนโทรลเลอร์



ยูสเซอร์



ไฟล์โปรแกรม



โปรแกรมเมอร์

เครื่องมือใช้เขียนโปรแกรม





ไฟล์และโฟลเดอร์
จัดเก็บไว้อย่างไร?



เขียนโปรแกรม
อย่างไร?



ดูผลการทำงาน
อย่างไร?



เขียนโปรแกรมให้ทำงานบนเบราว์เซอร์



กำหนดเนื้อหาและรูปแบบของหน้าเว็บ

HTML



กำหนดลักษณะการแสดงผลให้สวยงาม

CSS



กำหนดการทำงาน (เป็นสมองของเว็บ)

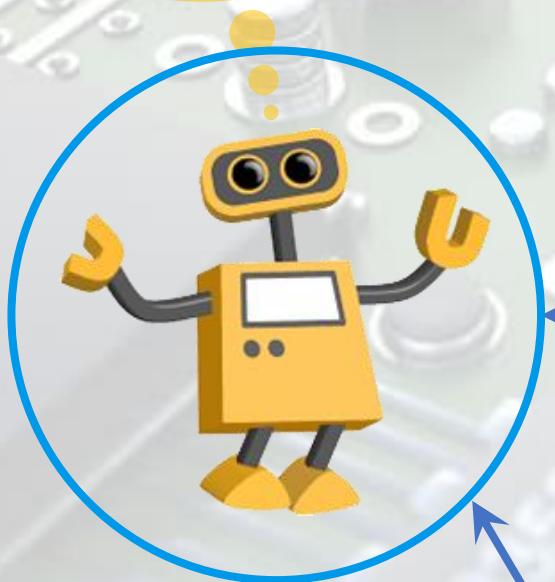
JS



คุยกันมากขึ้น เข้าใจกันมากขึ้น

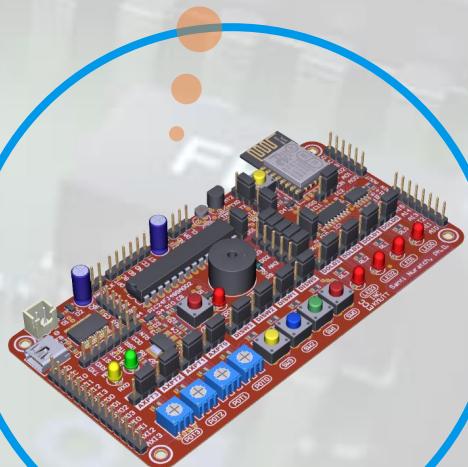


บอกรู้ว่า
Computer เดี่ยวฉันจะ
ไปสั่งการบอร์ดให้เอง



```
ai.buz.beep()  
ai.led.on(0)
```

ฉันทำงานได้หลายอย่าง
เลยนะ แต่ฉันจะทำตาม
เรอคนเดียวนะ AI



บอกรู้บอร์ดสั่งเสียงหน่อย
บอกรู้บอร์ดให้เปิดไฟหมายเลข 0 ให้ก



จัวงมวงคីយំបន្តកានកំរប



Curly brackets

{

}

Brackets

(

)

Square brackets

[

]

Quote

“ ”

Single Quote

‘ ’

Comma

,

Colon

:

Semi colon

;

Dot

.

Control

Ctrl

Alternate

Alt

Delete

Delete

Backspace

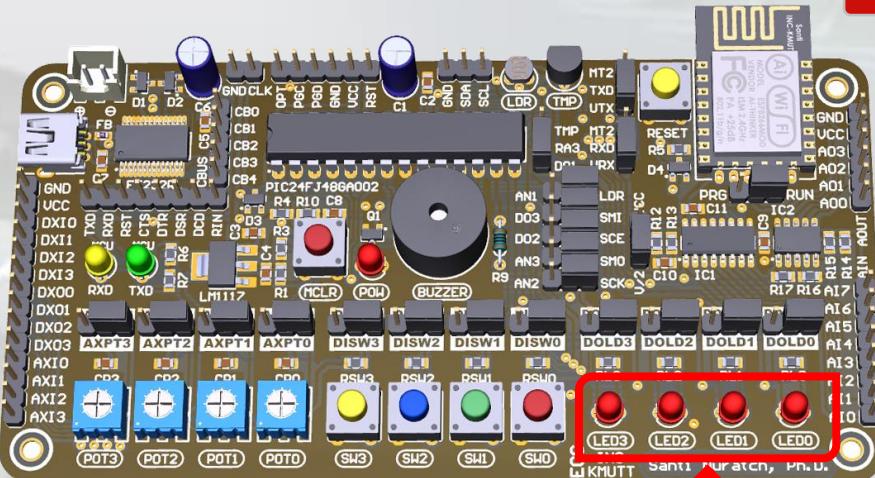
Back

Shift

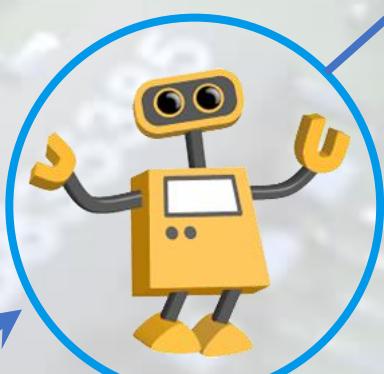
Shift



หลอดแอลอีดี (LED: Light Emitting Diode)



```
ai.led.on(LED.ID0)
ai.led.off(LED.ID0)
ai.led.inv(LED.ID0)
ai.led.wrt("0101")
...
```



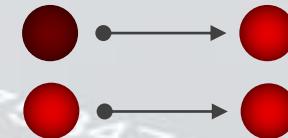
ชื่อ	หมายเลข	ไอเดีย (ID)
LEDO	0	LED.ID0
LED1	1	LED.ID1
LED2	2	LED.ID2
LED3	3	LED.ID3

พังก์ชันควบคุมแอลอีดี



พังก์ชันควบคุมสถานะ LED ให้ติด (on)

```
ai.led.set(id);  
ai.led.on(id);
```



พังก์ชันควบคุมสถานะ LED ให้ดับ (off)

```
ai.led.clr(id);  
ai.led.off(id);
```



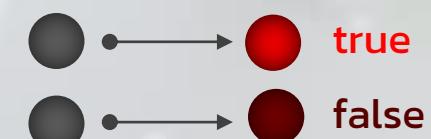
พังก์ชันควบคุมสถานะ LED ให้สลับสถานะ (toggle)

```
ai.led.inv(id);  
ai.led.toggle(id);
```



พังก์ชันควบคุมลوجิก (true/false) ของ LED

```
ai.led.control(id, logic);  
ai.led.ctrl(id, logic);
```



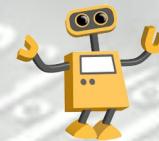
ควบคุมสถานะ LED คือการเขียนข้อมูลเพื่อควบคุมให้ LED ติด หรือ ดับ

ตรวจสอบสถานะ LED คือการอ่านข้อมูลจาก LED ว่า ณ เวลานั้น LED ติดหรือดับ





พิงก์ชันควบคุมสถานะของ LED ทั้งสี่ตัว (เขียนข้อมูลไปยัง LED)



```
ai.led.write("string-01");
ai.led.write([Array-01]);
ai.led.wrt("string-01");
ai.led.wrt([Array-01]);
ai.led.write(number0-15);
ai.led.wrt(number0-15);
```

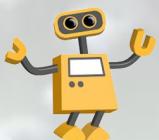
Note:

"string-01" คือข้อมูลแบบ string ที่ประกอบไปด้วย 0 และ 1 เช่น "1010" หรือ "1100"

[Array-01] คือข้อมูลแบบ array ที่ประกอบไปด้วยตัวเลข 0 และ 1 เช่น [1,0,1,0] หรือ [1,1,0,0]

number0-15 คือข้อมูลตัวเลขที่มีค่า 0-15

พิงก์ชันตรวจสอบสถานะของ LED



```
ret = ai.led.get(id);
ret = ai.led.req(id);
```



ret.val == 1



ret.val == 0

ควบคุมสถานะ LED คือการเขียนข้อมูลเพื่อควบคุมให้ LED ติด หรือ ดับ

ตรวจสอบสถานะ LED คือการอ่านข้อมูลจาก LED ว่า ณ เวลานั้น LED ติดหรือดับ

พิงก์ชันควบคุมและตรวจสอบแล็ปอีดี



ตัวอย่างการควบคุม LED ให้ติด (on)

```
ai.led.set(LED.ID0); // LED0 ติด  
ai.led.set(0); // LED0 ติด  
ai.led.on(LED.ID0); // LED0 ติด  
ai.led.on(0); // LED0 ติด
```

LED บนบอร์ดมีทั้งหมด 4 ตัว คือ LED0, LED1, LED2 และ LED3 การอ้างถึง LED เหล่านี้สามารถใช้ตัวเลข 0, 1, 2 และ 3 หรือ LED.ID0, LED.ID1, LED.ID2 และ LED.ID3 ได้ตามลำดับ

ตัวอย่างการควบคุม LED ให้ดับ (off)

```
ai.led.clr(LED.ID0); // LED0 ดับ  
ai.led.clr(0); // LED0 ดับ  
ai.led.off(LED.ID0); // LED0 ดับ  
ai.led.off(0); // LED0 ดับ
```

ชื่อ	หมายเลข	ໄວดี
LED0	0	LED.ID0
LED1	1	LED.ID1
LED2	2	LED.ID2
LED3	3	LED.ID3

ตัวอย่างการควบคุม LED ให้สลับสถานะ

```
ai.led.toggle(LED.ID0); // LED0 สลับสถานะ  
ai.led.toggle(0); // LED0 สลับสถานะ  
ai.led.inv(LED.ID0); // LED0 สลับสถานะ  
ai.led.inv(0); // LED0 สลับสถานะ
```

toggle หรือ **inverse** คือการสลับสถานะ เช่น ถ้า ณ ตอนนี้ LED ติดอยู่ คำสั่งนี้จะทำให้ LED ดับ หรือ ถ้า ณ ตอนนี้ LED ดับอยู่ คำสั่งนี้จะทำให้ LED ติด



ตัวอย่างการเขียนข้อมูลไปยัง LED กั้ง 4 ตัว

```
ai.led.write ("0101");      // LED0 และ 2 ติด  
ai.led.wrt("0101");        // LED0 และ 2 ติด  
ai.led.write ([0,1,0,1]);  // LED0 และ 2 ติด  
ai.led.wrt([0,1,0,1]);    // LED0 และ 2 ติด
```



ตัวอย่างการตรวจสอบสถานะ (on/off) ของ LED

```
ai.led.get(LED.ID0);          // อ่านสถานะ LED0  
ai.led.get(0);                // อ่านสถานะ LED0  
  
ai.led.req(LED.ID0);          // อ่านสถานะ LED0  
ai.led.req(0);                // อ่านสถานะ LED0
```

LED บนบอร์ดมีกั้งหมด 4 ตัว คือ LED0, LED1, LED2 และ LED3 การอ้างถึง LED เหล่านี้สามารถใช้ตัวเลข 0, 1, 2 และ 3 หรือ LED.ID0, LED.ID1, LED.ID2 และ LED.ID3 ได้ตามลำดับ

ชื่อ	หมายเลข	ไอเดีย
LED0	0	LED.ID0
LED1	1	LED.ID1
LED2	2	LED.ID2
LED3	3	LED.ID3



ลองฝึกเขียนโปรแกรมควบคุมแอลอีดี



คิดเอง เขียนเอง เก่งขึ้นแน่นอน

อยากให้ LED2 ติดตัวเดียว ต้องเขียนโปรแกรมอย่างไร?



อยากให้ LED1 และ LED3 ติดเพียงสองตัว ต้องเขียนโปรแกรมอย่างไร?

อยากให้ LED0 และ LED2 ติดเพียงสองตัว ต้องเขียนโปรแกรมอย่างไร?



รู้จักกับปุ่มกด (Button)



Click Me

แสดงข้อความอะไร (text)

สีอะไร (className)

เมื่อถูกคลิก ให้ทำอะไร (callback)

{ text, className, callback }



Button

ui.createButton

```
{  
  text: "สวัสดี",  
  className : "สี",  
  callback : ชื่อฟังก์ชัน  
}
```



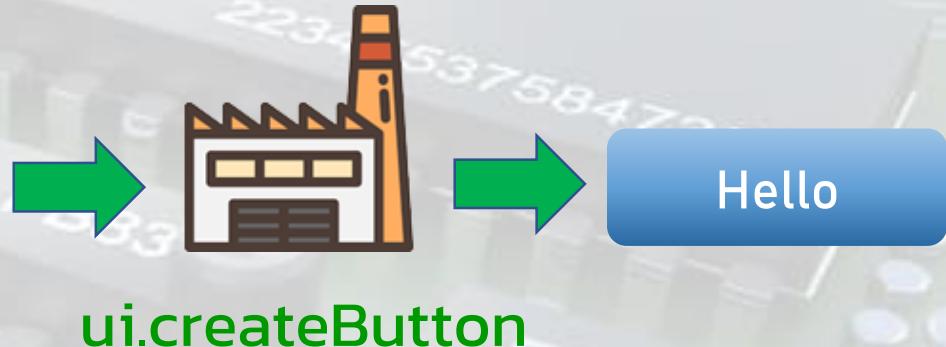
สวัสดี



ຝຶ່ງກໍ່ຂັບສົ່ງປຸມກດ (Button)

{

```
text: "Hello",  
className: "info",  
callback: function( ){ }  
}
```



ฟังก์ชันสร้างปุ่มกด (Button)

สร้างปุ่มกดแบบไม่กำหนดคุณสมบัติ (ใช้ค่าเริ่มต้น)

```
let button1 = ui.createButton();
```

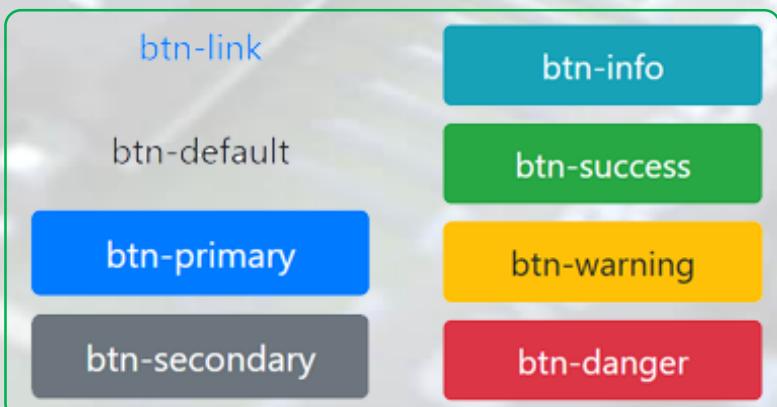
ปุ่มกด (button) ออกแบบมาให้ใช้เม้าส์คลิก การคลิกจะเป็นการสั่งให้โปรแกรมทำงานในฟังก์ชันที่กำหนดไว้ เช่น เมื่อคลิกแล้วแอล็อกอีเมลฯติด

สร้างปุ่มกดให้มีข้อความ Welcome

```
let button2 = ui.createButton({text: "Welcome"});
```

สร้างปุ่มกดให้มีข้อความ Goodbye และมีสีพื้นสีแดง

```
let button3 = ui.createButton({text: "Goodbye",
                               className: "danger"});
```



เพื่อความรวดเร็ว ลองรันโปรแกรมนี้

```
[ "link", "default", "primary", "secondary",
  "info", "success", "warning", "danger"]
  .map(c => ui.createButton({ text: c, className: `${c}` }));
```

รูปแบบการแสดงผล (className) ของปุ่มกด มีหลายแบบ

การเขียนโปรแกรมจะเร็วขึ้น เมื่อใช้เวลาฝึกฝนมากขึ้น

ปุ่มกดและกำกับดูผู้ใช้งาน

```
{  
    text: "Do It",  
    className: "btn-success",  
    callback: worker  
}
```



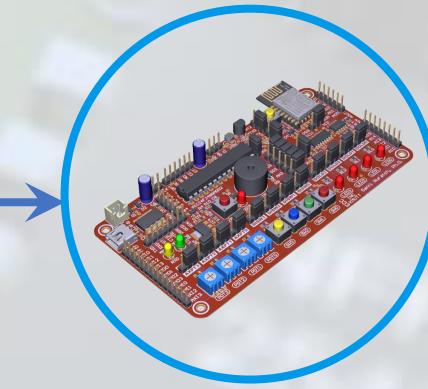
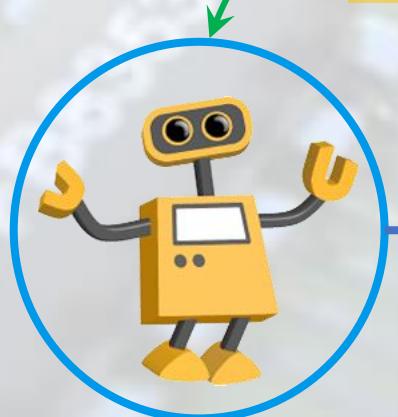
worker



JS



```
function worker() {  
    // ทดสอบ()  
    ai.led.on(LED.ID0)  
}
```



การสร้างปุ่มกดและกำหนดฟังก์ชันให้ทำงาน



สร้างปุ่มกดและกำหนดฟังก์ชัน (callback function) ให้ทำงานเมื่อถูกคลิก

```
function main() {  
    ui.createButton({ text: "Toggle LED0", className: "success", callback: action});  
}  
  
function action() {  
    ai.led.toggle(LED.ID0);  
}
```

สร้างปุ่มกดและกำหนดฟังก์ชันแบบไม่มีชื่อ (anonymous callback function)

```
ui.createButton({ text: "Toggle LED1",  
    className: "warning",  
    callback: function () { ai.led.inv(LED.ID1) }});
```

สร้างปุ่มกดและกำหนดฟังก์ชันแบบแอกซ์เริร์โว (arrow callback function)

```
ui.createButton({ text: "Toggle LED3", className: "danger",  
    callback: () => ai.led.inv(LED.ID3) });
```

ภาษาคอมพิวเตอร์มีความหลากหลาย และยึดหยุ่นสูงเหมือนภาษาคน ผลของการใช้พิດรูปแบบ ย่อมาให้ผลก็ พิດพลาด พุดพิด คนฟังก์เข้าใจพิด เบี้ยนໂປຣແກຣມพິດ ຄອມພິວເຕອຣົກໍຖ້າກໍານານພິດ (หรือไม่ทำงาน)



เรียนให้มาก อ่านให้มาก และต้องฝึกทำให้มากยิ่งกว่า

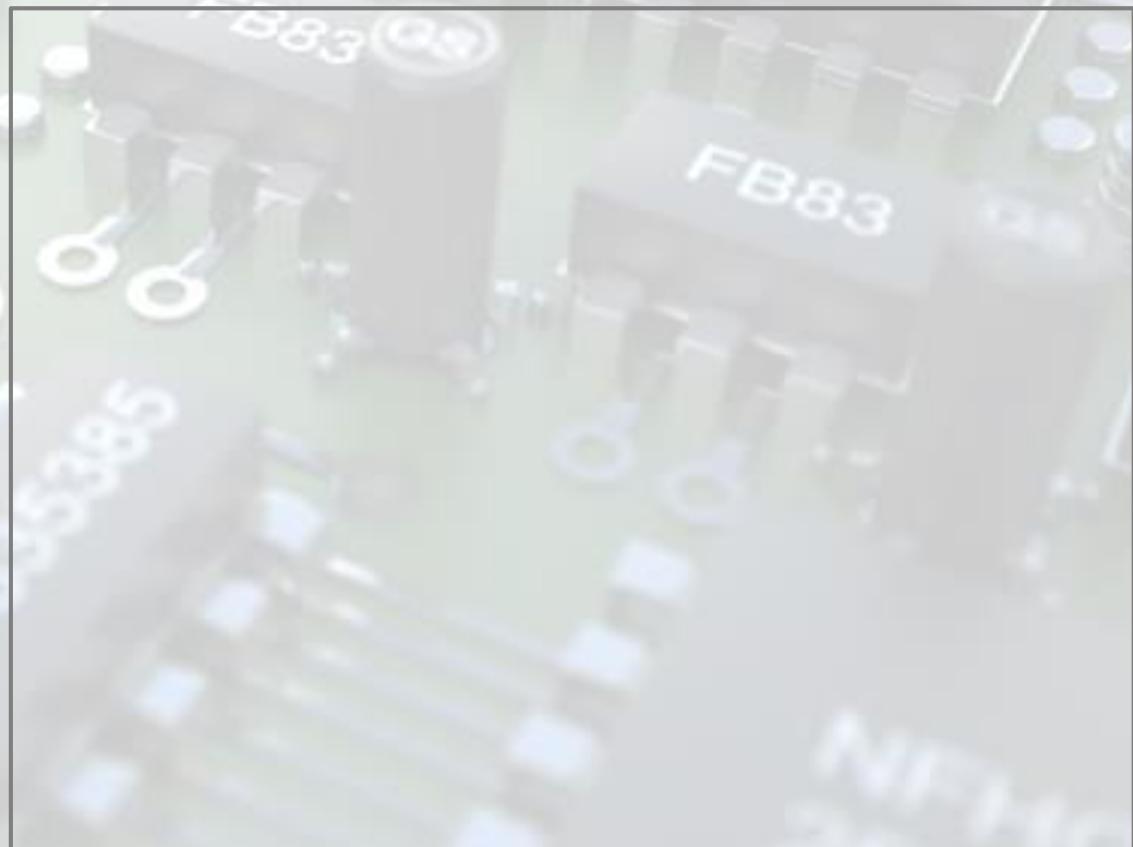
สร้างปุ่มกด จำนวน 4 ปุ่ม ตามภาพ เมื่อปุ่มเหล่านี้ถูกคลิก จะทำการกลับสถานะของแอลอีดี

Toggle LED0

Toggle LED1

Toggle LED2

Toggle LED3



ฟังก์ชันสร้าง Panel/Container/Card)



สร้าง Panel แบบไม่มี text (ไม่มีข้อความ Title)

```
let panel1 = ui.createPanel();
```



สร้าง Panel แบบมี text (มีข้อความ Title)

```
let panel2 = ui.createPanel({  
    text: "Container/Card2"  
});
```

Container/Card2

สร้าง Panel แบบมีสีพื้นหลัง(ส้ม) และกำหนดสีของข้อความ (นำเงิน)

```
let panel3 = ui.createPanel({  
    text: "Container/Card3",  
    className: "warning text-primary"  
});
```

Container/Card3



การสร้างและปรับแต่ง Container

การทำงานได้ทุกต้องจะมีคู่กันกับความสวยงาม
(ความคิดสร้างสรรค์ สำคัญไม่แพ้กักษะด้านอื่น)

ลองสร้าง Graphics User Interface (GUI) ตามภาพ และมีการทำงานที่ถูกต้อง

LED0 & LED1 Control Panel

Toggle LED0

Toggle LED1

LED2 & LED3 Control Panel

Toggle LED2

Toggle LED3

```
let panel1 = ui.createPanel({
  text: "LED0 & LED1 Control Panel",
  className: "danger text-warning"
});

new ui.createButton({
  text: "Toggle LED0",
  parentId: panel1.getBodyId(),
  className: "info",
  callback: () => ai.led.inv(LED.ID0)
});

new ui.createButton({
  text: "Toggle LED1",
  parentId: panel1.getBodyId(),
  className: "success",
  callback: () => ai.led.inv(LED.ID1)});
```

ก่อนไปต่อ ลองสร้าง Panel/Container สีเบี้ยงด้วยตัวเองก่อน

พิngก์ชันการสร้างสัญญาณเสียง Beep



พิngก์ชันการสร้างเสียง Beep

```
ai.buz.beep();
```

พิngก์ชันการสร้างเสียง Beep แบบกำหนดความถี่

```
ai.buz.beep(frequency);
```

พิngก์ชันการสร้างเสียง Beep แบบกำหนดความถี่ และ ความยาว

```
ai.buz.beep(frequency, period);
```

frequency คือตัวเลขกำหนดความถี่ของสัญญาณเสียง มีหน่วยเป็นเฮิรตซ์ (Hz)

period คือตัวเลขกำหนดความยาวของสัญญาณเสียง มีหน่วยเป็น 1/1000 วินาที (mS)

สัญญาณเสียงเกิดจากการสั่นของสิ่งต่างๆ ความถี่ของสัญญาณเสียงจะแปรผันตรงกับความเร็วในการสั่น ความถี่สูงจะได้เสียงแหลม ความถี่ต่ำจะได้เสียงกุ้ง



พิ้งก์ชันการสร้างสัญญาณเสียง Beep



ตัวอย่างการควบคุมสัญญาณเสียงโดยปุ่มกด

Beep Sound Control Panel

High Frequency

Low Frequency

อยากได้สัญญาณเสียงโน้ตดนตรี
ໂດ ແ ພ່າ
ต้องปรับโปรแกรมอย่างไร

```
let soundH = 4000;
let soundL = 1000;

let panel1 = ui.createPanel({
  text: "Beep Sound Control Panel",
  className: "bg-gray text-dark"
});

ui.createButton({
  text: "High Frequency",
  className: "btn-danger",
  parentId: panel1.getBodyId(),
  callback: () => ai.buz.beep(soundH)
});

ui.createButton({
  text: "Low Frequency",
  className: "btn-warning",
  parentId: panel1.getBodyId(),
  callback: () => ai.buz.beep(soundL)
});
```



ฟังก์ชันจัดการเวลา

ฟังก์ชันสร้าง Interval กำหนดให้ทำงานเป็นช่วงๆ ตามเวลาที่กำหนด

```
setInterval(callback, intervalTime);
```

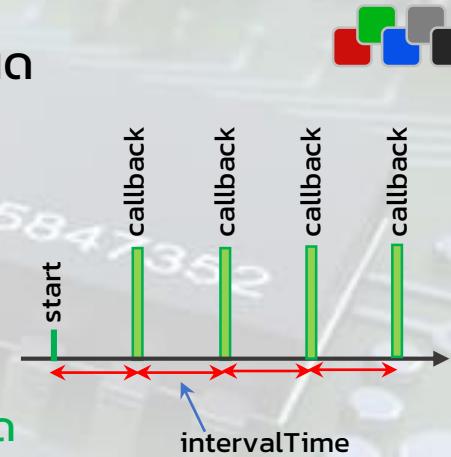
ฟังก์ชันยกเลิก Interval

```
clearInterval(intervalObj);
```

callback คือชื่อของฟังก์ชันที่ต้องการให้ทำงานทุกๆ เวลาที่กำหนด

intervalTime คือตัวเลขกำหนดระยะเวลา (หน่วยเป็น mS)

intervalObj คือชื่อของตัวแปรที่สร้างมาจากฟังก์ชัน setInterval()



ฟังก์ชันสร้าง Timeout กำหนดให้รอตามเวลาที่กำหนดแล้วค่อยทำงาน (ทำครั้งเดียว)

```
setTimeout(callback, delayTime);
```

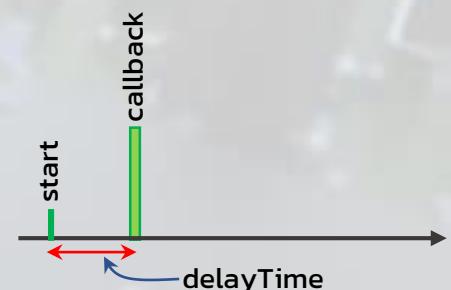
ฟังก์ชันยกเลิก Timeout

```
clearTimeout(timeoutObj);
```

callback คือชื่อของฟังก์ชันที่ต้องการให้ทำงานเมื่อรอนานตามเวลาที่กำหนด

delayTime คือตัวเลขกำหนดระยะเวลา (หน่วยเป็น mS)

timeoutObj คือชื่อของตัวแปรที่สร้างมาจากฟังก์ชัน setTimeout()



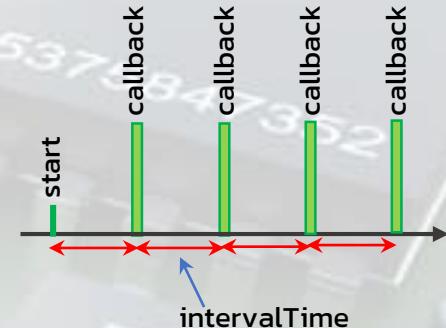
การใช้งาน Interval



โปรแกรมไฟกระพริบดวงเดียว

```
let ti = setInterval(toggle, 500);

function toggle() {
    ai.led.toggle(LED.ID0);
}
```



โปรแกรมไฟกระพริบสลับกันสองดวง

```
ai.led.write("0000");
ai.led.on(LED.ID0);

var ti = setInterval(toggle, 500);

function toggle() {
    ai.led.toggle(LED.ID0);
    ai.led.toggle(LED.ID3);
}
```

LED ดับทุกตัว

LEDO ติด

สร้าง Interval ให้ทำงานทุก 0.5 วินาที

ทำงานทุก 0.5 วินาที เรียกโดย ti

สลับสถานะของ LED0

สลับสถานะของ LED3



การใช้งาน Interval



โปรแกรมไฟกระพริบควบคุมการทำงานด้วยปุ่มกด (START/STOP)

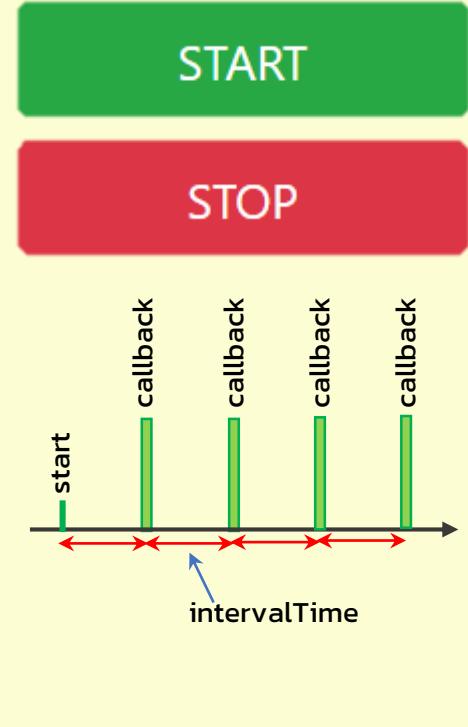
```
ui.createButton({ text: "START", className: "success", callback: start });
ui.createButton({ text: "STOP", className: "danger", callback: stop });

let ti;

function start() {
    clearInterval(ti);
    ti = setInterval(toggle, 500);
}

function stop() {
    clearInterval(ti);
}

function toggle() {
    ai.led.toggle(LED.ID0);
}
```

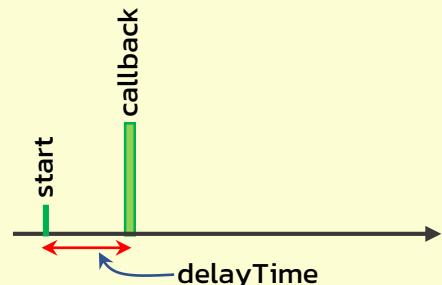


การใช้งาน Timeout

โปรแกรมเปิด/ปิด อัตโนมัติตามเวลาที่กำหนด

```
ui.createButton({  
    text: "START",  
    className: "btn-success",  
    callback: start  
});  
  
let to;  
  
function start() {  
    ai.led.write(0);  
    ai.led.on(LED.ID3)  
    setTimeout(stop, 2000);  
}  
  
function stop() {  
    ai.led.off(LED.ID3);  
    clearTimeout(to);  
}
```

START



การสร้างและใช้งานลูกบิด (Knob)

การสร้างและใช้งานลูกบิด (Knob) เพื่อควบคุม LED

```
ui.createKnob({
    min: 0,
    max: 3,
    callback: led_action1 //เปลี่ยนเกียบกับ led_action2
});

function led_action(id) {
    ai.led.write(0);
    ai.led.on(id);
}

function led_action2(id) {
    ai.led.write(1 << id);
}
```

0	0	0	1	$1 \ll 0$
0	0	1	0	$1 \ll 1$
0	1	0	0	$1 \ll 2$
1	0	0	0	$1 \ll 3$

บ่อยครั้ง การแก้ปัญหาที่มีประสิทธิภาพ จะมาจากการความรู้เบื้องต้น ที่คุณก้าวไปมาก่อนข้าม!



ฟังก์ชันสร้างลูกบิด (Knob) และการใช้งาน



การสร้างและใช้ลูกบิด (Knob) เพื่อเปลี่ยนความถี่เสียง และความยาวเสียง

```
let period = 100;

let knobF = ui.createKnob({
    min: 500,
    max: 5000,
    callback: beep_action
});
let knobP = ui.createKnob({
    min: 100,
    max: 2000,
    hue: 70,
    callback: time_action
});

function beep_action(freq) {
    ai.buz.beep(freq, period);
}

function time_action(time) {
    period = time;
}
```



พิงก์ชันตรวจจับการกดสวิตซ์



โปรแกรมตรวจจับการกดสวิตซ์ และสร้างสัญญาณเสียงเมื่อสวิตซ์ถูกกด

```
ai.psw.sub(PSW.ID0, psw0_action);
ai.psw.sub(PSW.ID1, psw1_action);
```

```
function psw0_action() {
    ai.buz.beep(1000)
}
```

```
function psw1_action() {
    ai.buz.beep(4000)
}
```





โปรแกรมควบคุมการติดดับของ LED เมื่อสวิตซ์ถูกกด

```
ai.psw.sub(PSW.ID2, psw2_action);
ai.psw.sub(PSW.ID3, psw3_action);

function psw2_action() {
    ai.led.toggle(LED.ID2)
}

function psw3_action() {
    ai.led.toggle(LED.ID3)
}
```





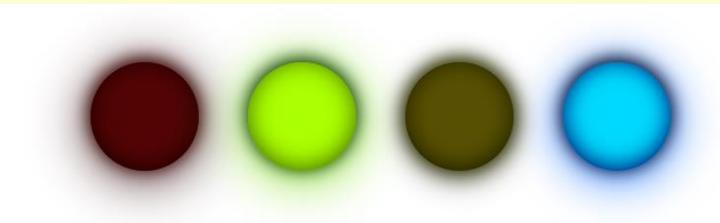
โปรแกรมตรวจจับการกดสวิตซ์ถูกและแสดงผลสถานะด้วย LED Indicator

```

let led0 = ui.createLed({
  className: 'red'
})
let led1 = ui.createLed({
  className: 'green'
})
let led2 = ui.createLed({
  className: 'yellow'
})
let led3 = ui.createLed({
  className: 'blue'
})

ai.psw.sub(PSW.ID0, () => led3.flash(500));
ai.psw.sub(PSW.ID1, () => led2.flash(500));
ai.psw.sub(PSW.ID2, () => led1.flash(500));
ai.psw.sub(PSW.ID3, () => led0.flash(500));

```

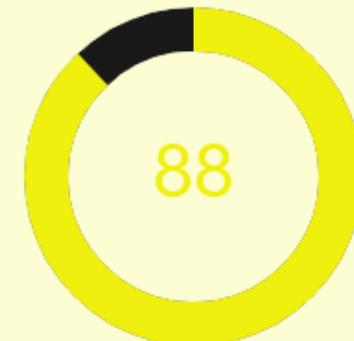
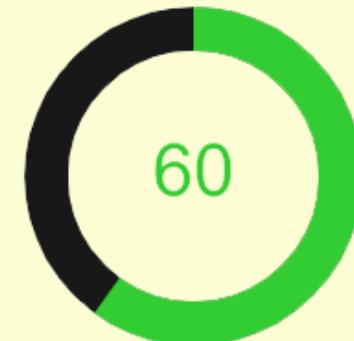


พิงก์ชันอ่านค่าแรงดันไฟฟ้า



โปรแกรมอ่านค่าแรงดันไฟฟ้าจาก POT และ LDR

```
let gauge1 = ui.createGauge({  
    hue: 120  
});  
let gauge2 = ui.createGauge({  
    hue: 60  
});  
  
ai.adc.sub(ADC.ID0, adc => {  
    gauge1.setValue(adc.val);  
})  
  
ai.adc.sub(ADC.ID1, adc => {  
    gauge2.setValue(adc.val);  
})
```



โปรแกรมนับวัตถุ จากการตัดผ่านแสง



โปรแกรมอ่านค่าแรงดันไฟฟ้าจาก POT และ LDR

```
let gauge = ui.createGauge({
    hue: 44
})
let led = ui.createLed({
    className: 'red'
})

let counter = 0;
ai.adc.sub(ADC.ID1, light => {
    if (light.dir == "dec") {
        ai.buz.beep();
        ai.led.flash(LED.ID0);
        led.flash(500)
        counter = counter + 1;
        gauge.setValue(counter);
    }
})
```



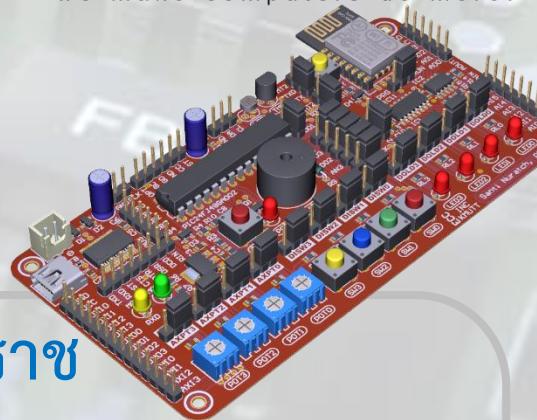
It's time to think and code!



the MORE
YOU PRACTICE
THE BETTER
YOU GET



THANK YOU!



ផ្សេងៗសាស្ត្រាអារីយ៍ .ទរ.ស៊ានិ នុរាជ

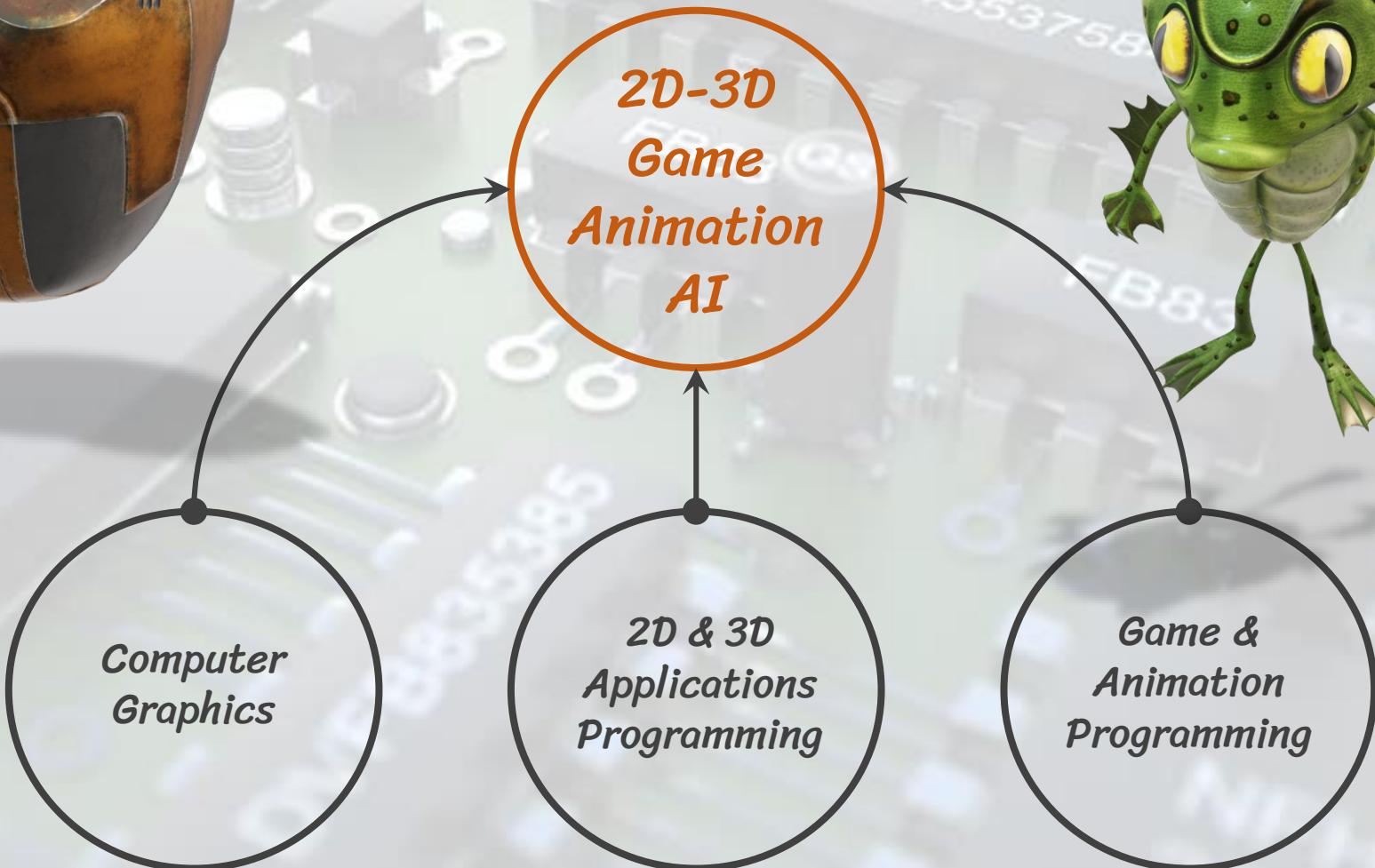
Asst.Prof.Dr.Santi Nuratch., Ph.D.

Embedded Computing and Control Lab. @ INC-KMUTT

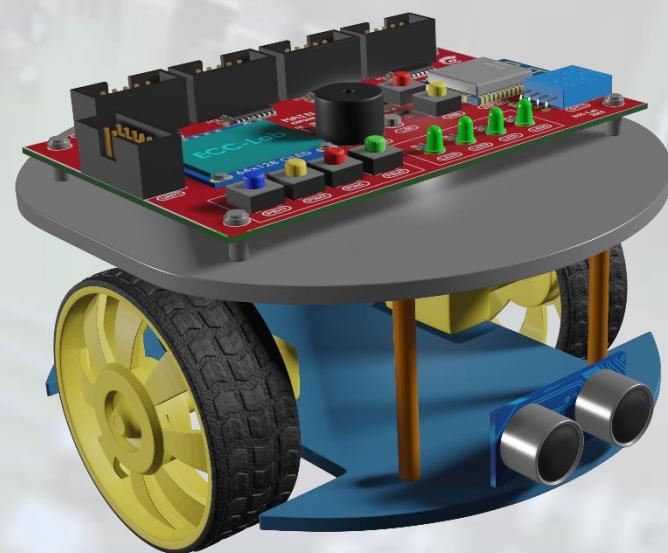
santi.inc.kmutt@gmail.com

Department of Control System and Instrumentation Engineering,
King Mongkut's University of Technology Thonburi, **KMUTT**

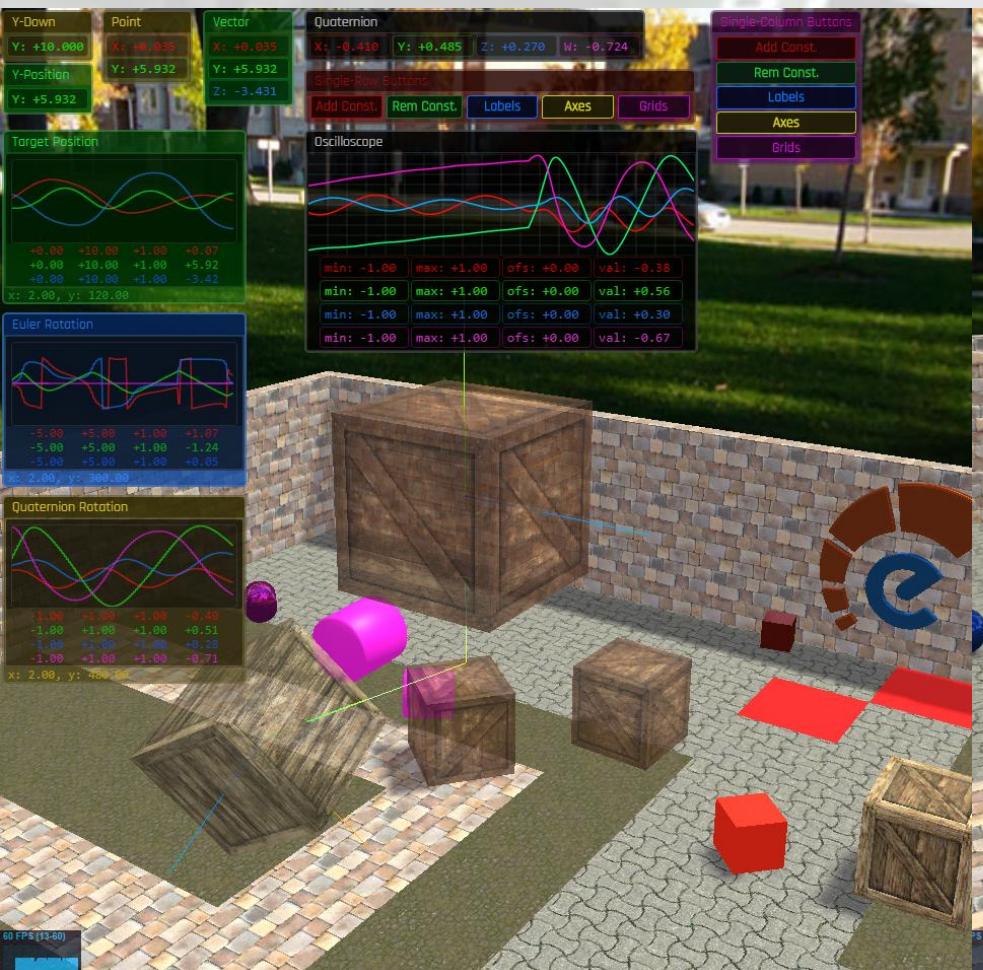
Day #2



ເຂົ້າໃນໂປຣແກຣມຄວບຄຸມແຂນກລ ມູນຍິບຕໍ່ ແລະ ເຄຣື່ອງຈັກ



ເຫັ້ນໂຍງ ກຣາຟິກສີ ຝີສັກສີ ເວົ້ວ ແລະ ແວບັນເຂັ້ນ





ไฟล์และโฟลเดอร์
จัดเก็บไว้อย่างไร?

เขียนโปรแกรม
อย่างไร?

ดูผลการทำงาน
อย่างไร?



การรันโปรแกรมและควบคุมมุมมอง



รันโปรแกรมโดยการใช้คำสั่ง `npm start`

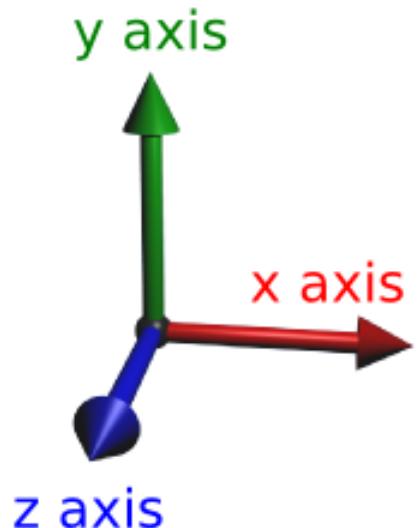
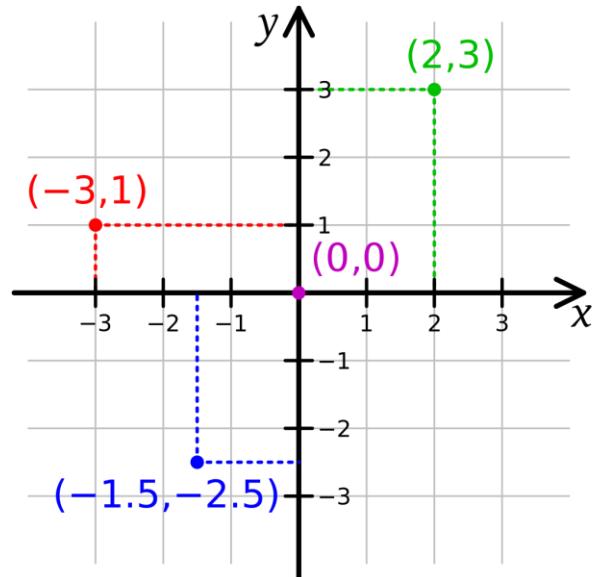
ตรวจสอบผลที่ Web-browser

การควบคุมมุมมอง

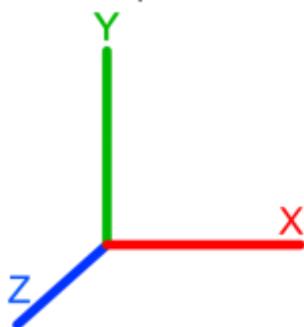
- Left-Mouse & Drag เพื่อหมุนมุมมอง (Rotate)
- Right-Mouse & Drag เพื่อเปลี่ยนตำแหน่ง (Pan)
- Middle-Mouse & Drag เพื่อขยาย (Zoom)



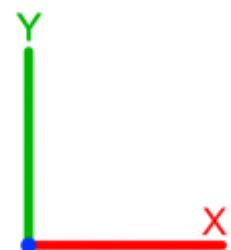
เข้าใจระบบแกนสอง และ สามมิติ



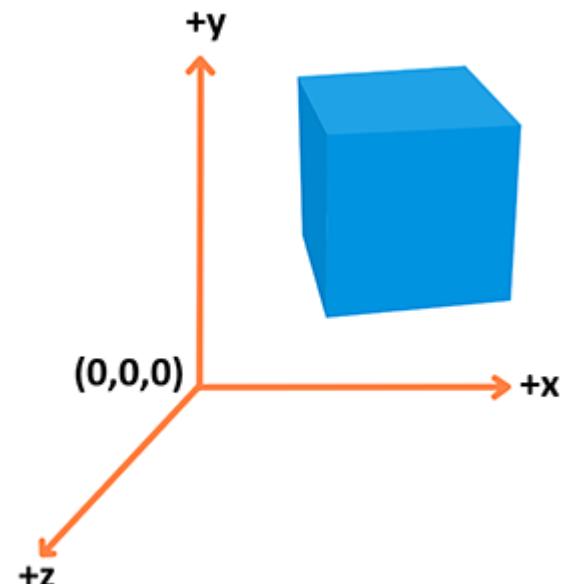
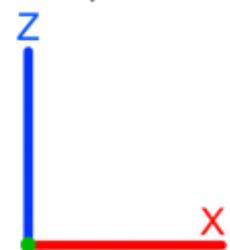
Perspective



Front view



Top view



เริ่มใช้งาน Game Engine

การเปลี่ยนตำแหน่งของวัตถุบนระบบสมมติ (x, y, z)

```
import GameEngine from '../libs/GameEngine'

new GameEngine({
    models:      ['mousey'],
    onReady:     ready,
    onLoaded:    loaded,
    onUpdate:   update
})

function ready(engine, objs) {
    engine.setCameraPosition(0, 1, 6);
}

function loaded(engine, animator, character, name) {

}

function update(engine, animator, character, name) {

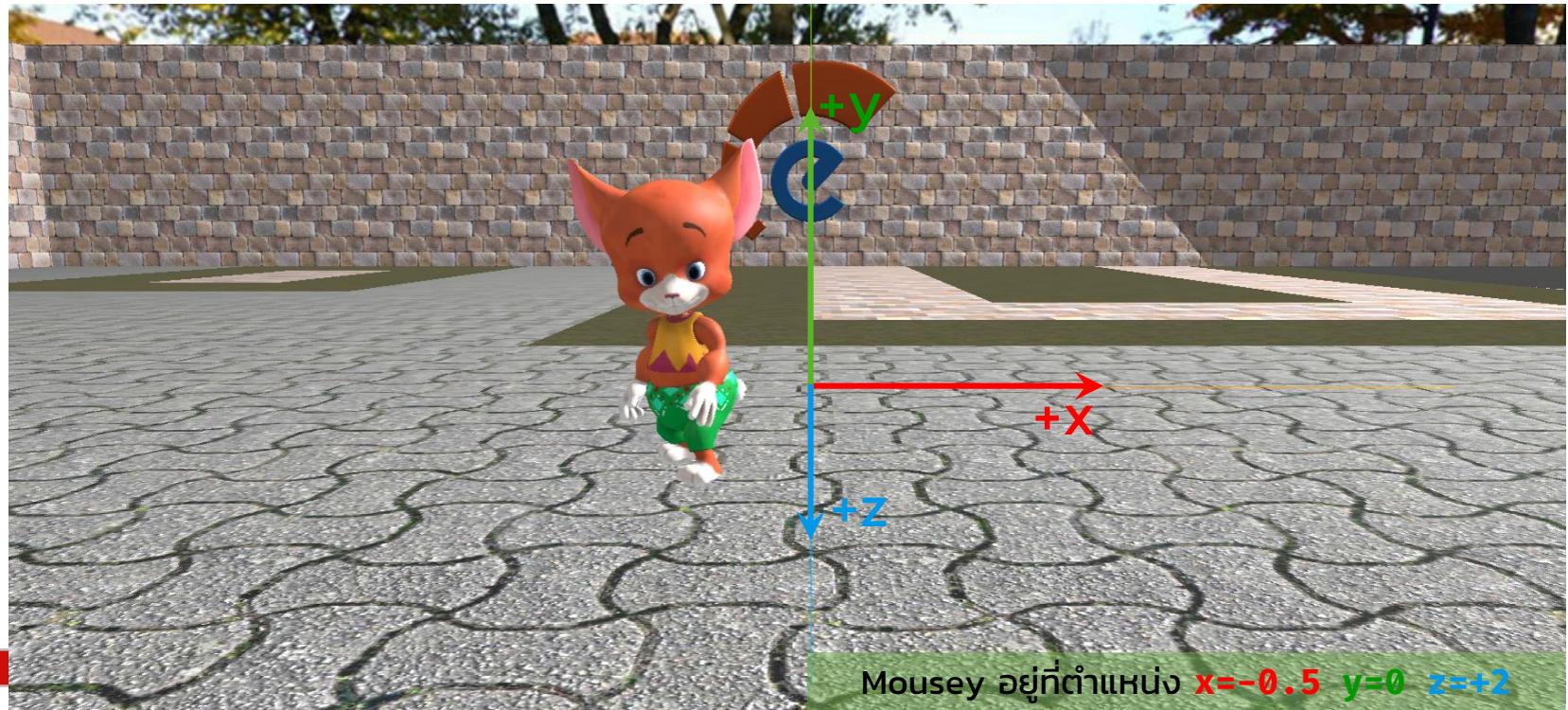
}
```



การควบคุมตำแหน่งของวัตถุ

การเปลี่ยนตำแหน่งของวัตถุในระบบสามมิติ (x , y , z)

```
function loaded(engine, animator, character, name) {  
    character.position.x = -0.5;  
    character.position.y = 0;  
    character.position.z = +2;  
}
```

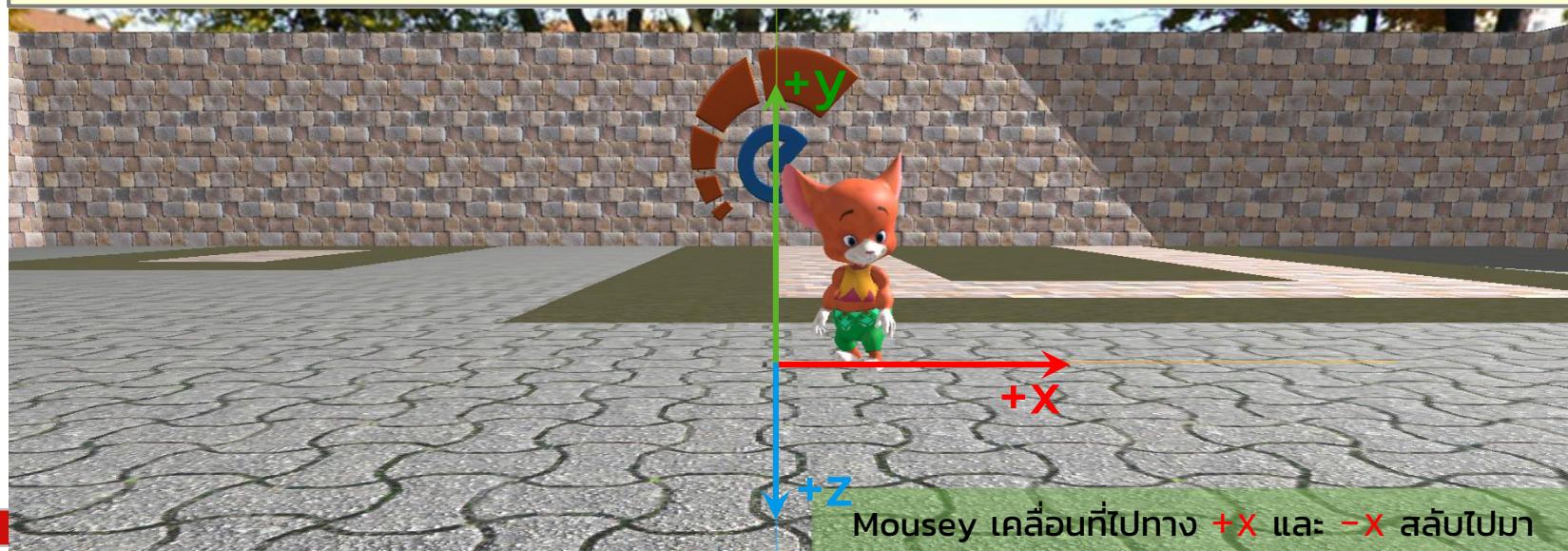


การควบคุมตำแหน่งของวัตถุแบบต่อเนื่อง

การเปลี่ยนตำแหน่งของวัตถุแบบต่อเนื่อง

```
let step = 0.02;
let dir = 1;
function update(engine, animator, character, name) {
    character.position.x += step*dir;
    character.position.y = 0;
    character.position.z = 0;

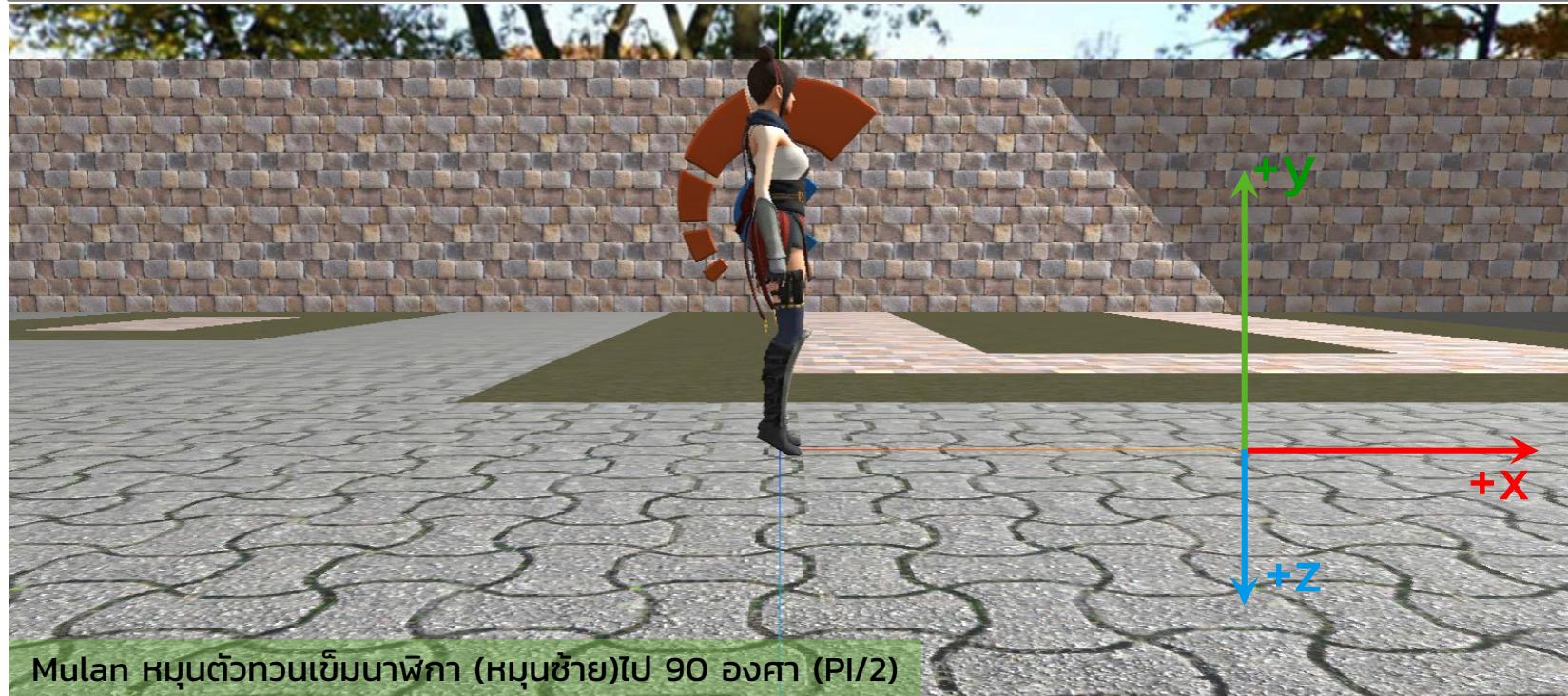
    if(Math.abs(character.position.x) > 3) {
        dir = dir * (-1);
    }
}
```



การควบคุมองค์ความรู้ของวัตถุ

การควบคุมองค์ความรู้ของวัตถุ

```
function loaded(engine, animator, character, name) {  
    character.rotation.x = 0;  
    character.rotation.y = Math.PI/2;  
    character.rotation.z = 0;  
}
```



การควบคุมองค์ความรู้ของวัตถุแบบต่อเนื่อง

การควบคุมองค์ความรู้ของวัตถุแบบต่อเนื่อง

```
let rad = 0.0;
function update(engine, animator, character, name) {
    character.rotation.x = 0;
    character.rotation.y = rad;
    character.rotation.z = 0;

    rad += 0.01;
    if(rad > 2*Math.PI) {
        rad = 0.0;
    }
}
```

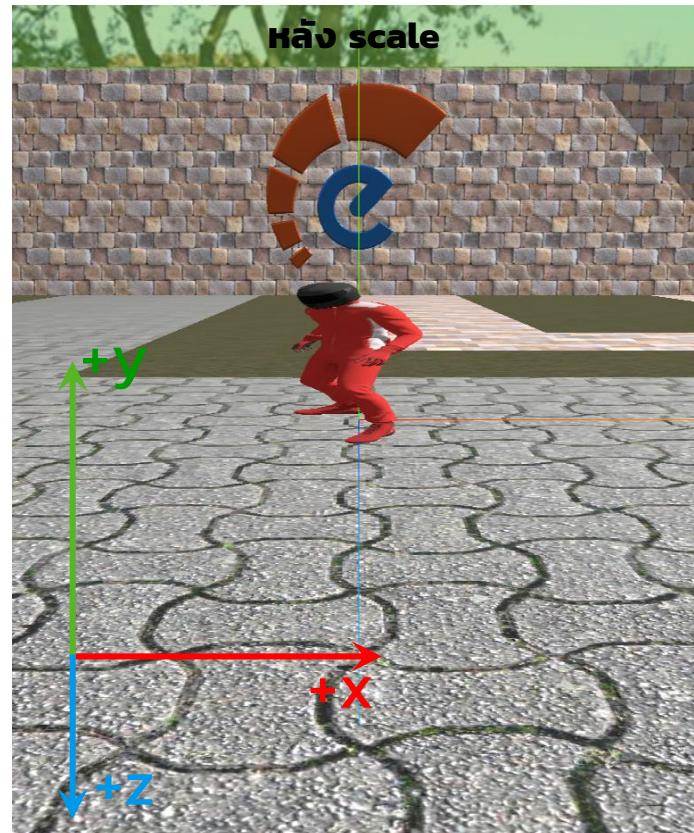
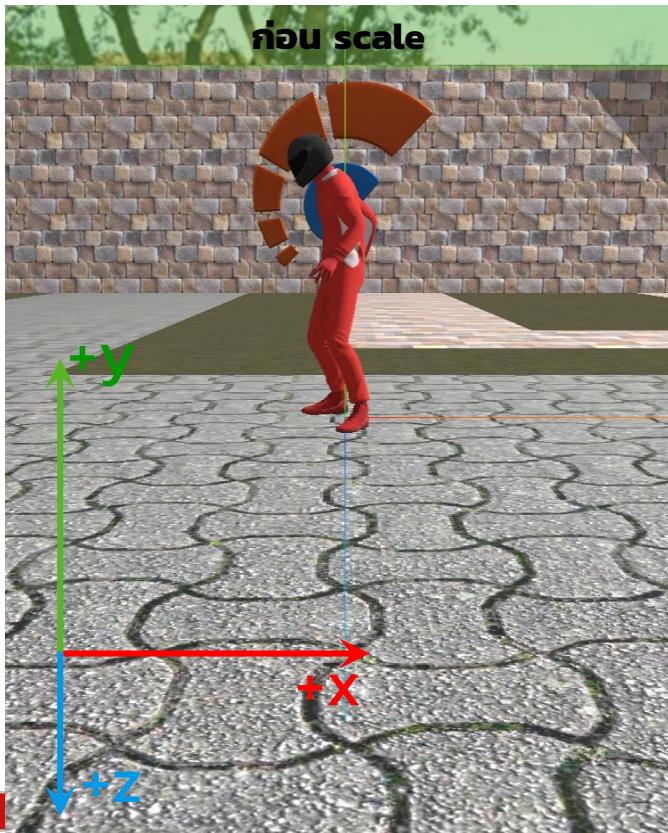


Mulan หมุนตัวทวนเข็มนาฬิกา (หมุนซ้าย) อย่างต่อเนื่อง

การควบคุมขนาดของวัตถุ

การควบคุมขนาดของวัตถุ

```
function loaded(engine, animator, character, name) {  
    character.scale.x = 1.5;  
    character.scale.y = 0.5;  
    character.scale.z = 1.5;  
}
```

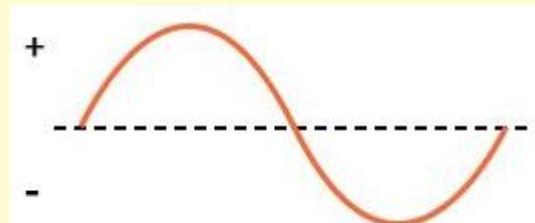


การควบคุมขนาดของวัตถุแบบต่อเนื่อง

การควบคุมขนาดของวัตถุแบบต่อเนื่อง

```
let factor = 1.0;
let alpha = 0.0;
function update(engine, animator, character, name) {
    character.scale.x = factor;
    character.scale.y = factor;
    character.scale.z = factor;

    factor = 1 + Math.sin(alpha);
    alpha += 0.01;
}
```



การสร้างปุ่มควบคุม (Buttons)

การสร้างปุ่มควบคุม (Buttons) และการประยุกต์ใช้งาน

```
function loaded(engine, animator, character, name) {  
    createButtons(engine)  
}  
let dir = 0;  
function update(engine, animator, character, name) {  
    character.rotation.y += 0.01*dir;  
}
```

```
function createButtons(engine) {  
    const btnc = WebGui.createContainer('Control Panel', 'pink', 5, 5);  
    const btnNames = ['RotateY+', 'RotateY-', 'Labels', 'Axes', 'Grids'];  
    const btnTypes = ['red', 'green', 'blue', 'yellow', 'pink'];  
    for (let i = 0; i < btnTypes.length; i++) {  
        WebGui.createButton(btnc, btnNames[i], i, btnTypes[i], buttonActions);  
    }  
    function buttonActions(btn) {  
        if (btn.id == 0) {  
            dir = +1;  
        } else if (btn.id == 1) {  
            dir = -1;  
        } else if (btn.id == 2) {  
            engine.toggleLabels();  
        } else if (btn.id == 3) {  
            engine.toggleAxes();  
        } else if (btn.id == 4) {  
            engine.toggleGrids();  
        }  
    }  
}
```



พิสิกส์ (แรงส่ง แรงด้าน กับการเคลื่อนที่)

การเปิดใช้งาน Physics Engine และการตรวจสอบการกดคีย์บอร์ด

```
function ready(engine, objs) {  
    engine.setCameraPosition(0, 1, 6);  
    engine.physics.enabled=true;  
}  
  
function loaded(engine, animator, character, name) {  
}  
  
const KEY_DELAY      = 100;  
const FORCE_SCALE   = 200;  
function update(engine, animator, character, name) {  
    if (engine.getKeyDown('f', KEY_DELAY)) {  
        engine.applyForceToRayBody(FORCE_SCALE);  
    }  
}
```



Ray-Casting & Particles

จับตำแหน่งเม้าส์ด้วย Ray-Casting และสร้าง Particles

```
function ready(engine, objs) {  
    engine.setCameraPosition(3, 6, 15);  
    engine.physics.enabled=true;  
    initParticles(engine);  
}  
let particleGroup, emitter;  
function update(engine, animator, character, name) {  
    particleGroup.tick(engine.getDeltaTime() / 1000);  
    if (engine.getRayIntersec()) {  
        let point = engine.getRayIntersec().point;  
        emitter.position.value = point  
        character.position.copy(point);  
    }  
}
```



การโหลดโมเดลเข้ามาอย่างจาก (Scene)



การโหลดโมเดลเข้ามาอย่างจาก (Scene)

```
const assetName = 'models/boxes001.gltf';
const targetName = 'CubeRedWood001';
let targetBox;
function ready(engine, objs) {
    engine.setCameraPosition(3, 6, 15);
    engine.physics.enabled=true;
    engine.loadModel(assetName).then( (model) => {
        targetBox = engine.getBodyByMeshName(targetName);
    });
}

function update(engine, animator, character, name) {
    if(engine.getKeyDown('n', 200)) {
        engine.toggleLabels();
    }
    if (engine.getKeyDown('f', 200)) {
        engine.applyForceToRayBody(500);
    }
    if(targetBox) {
        targetBox.angularVelocity.set(0, 20, 0);
    }
}
```



โหลด boxes001 เข้ามาในจาก

การโคลนโมเดลและจัดวางในตำแหน่งที่ต้องการ

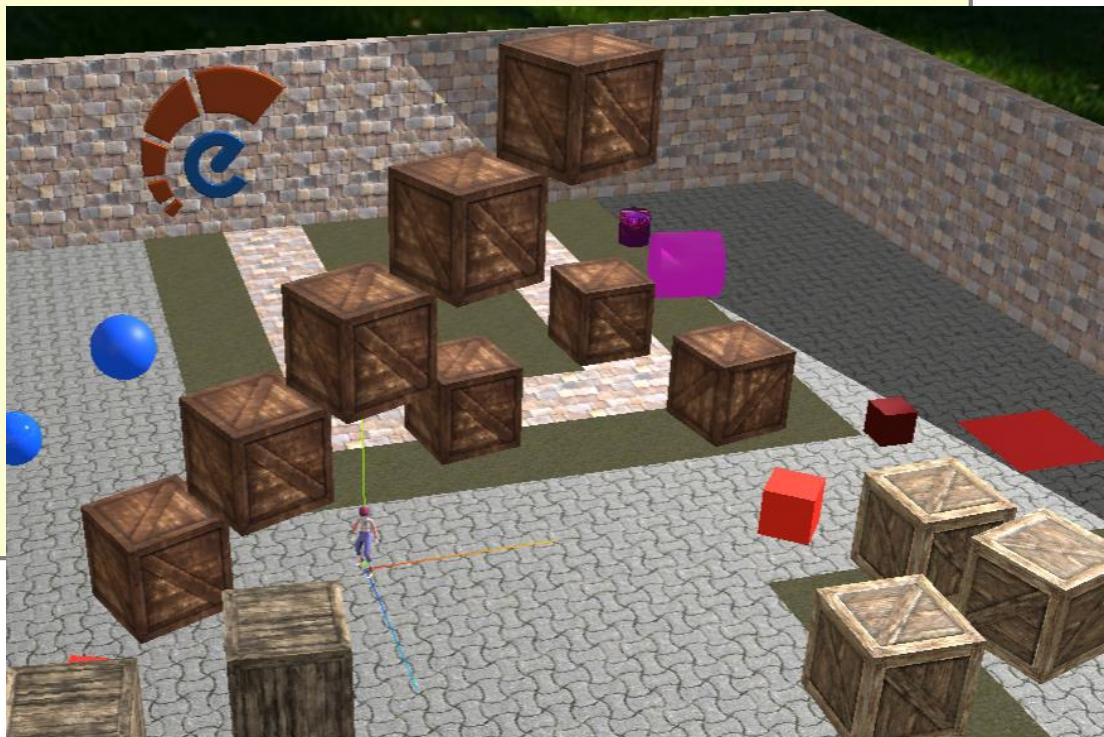


การโคลนโมเดลและจัดวางในตำแหน่งที่ต้องการ

```
const boxPositions = [
  {x: 0, y: 0, z: 0},
  {x: 2, y: 2, z: 0},
  {x: 4, y: 4, z: 0},
  {x: 6, y: 6, z: 0},
  {x: 8, y: 8, z: 0},
];

const Parent = new THREE.Object3D();
const Boxes = [];
function cloenBox(engine, org) {

  boxPositions.map(p => {
    const box = org.clone();
    box.rotation.set(0, 0, 0);
    box.position.x = p.x;
    box.position.y = p.y;
    box.position.z = p.z;
    Boxes.push(box);
    Parent.add(box);
  });
  Parent.position.set(-4, 1, 0);
  engine.getScene().add(Parent);
}
```



การโหลดตัวละครหลายตัวเข้ามาในจาก

การโหลดตัวละครหลายตัวเข้ามาในจาก และการควบคุม Animation ของเป้าหมาย

```
new GameEngine({
    models:      ['mulan', 'mousey', 'ortiz', 'doozy', 'nancy', 'pubg'],
    onReady:     ready,
    onLoaded:    loaded,
    onUpdate:   update
})

function ready(engine, objs) {
    engine.setCameraPosition(3, 6, 15);
}

function update(engine, animator, character, name) {
    if(name == 'mulan') {
        animator.playAnimationByKeyboard();
    }
}
```

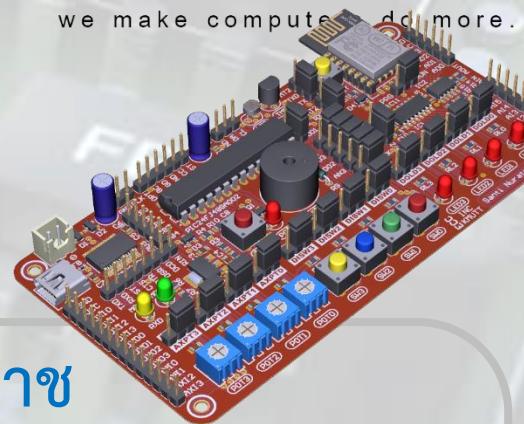
ลองกดคีย์ 0, 1, 2, 3, 4
แล้วดูท่าทางของตัวละคร



ฝึกควบคุมรถ และ แขนกล (แป้งกัน)



THANK YOU!



ผู้ช่วยศาสตราจารย์ ดร.สันติ นุราษ

Asst.Prof.Dr.Santi Nuratch., Ph.D.

Embedded Computing and Control Lab. @ INC-KMUTT

santi.inc.kmutt@gmail.com

Department of Control System and Instrumentation Engineering,
King Mongkut's University of Technology Thonburi, **KMUTT**