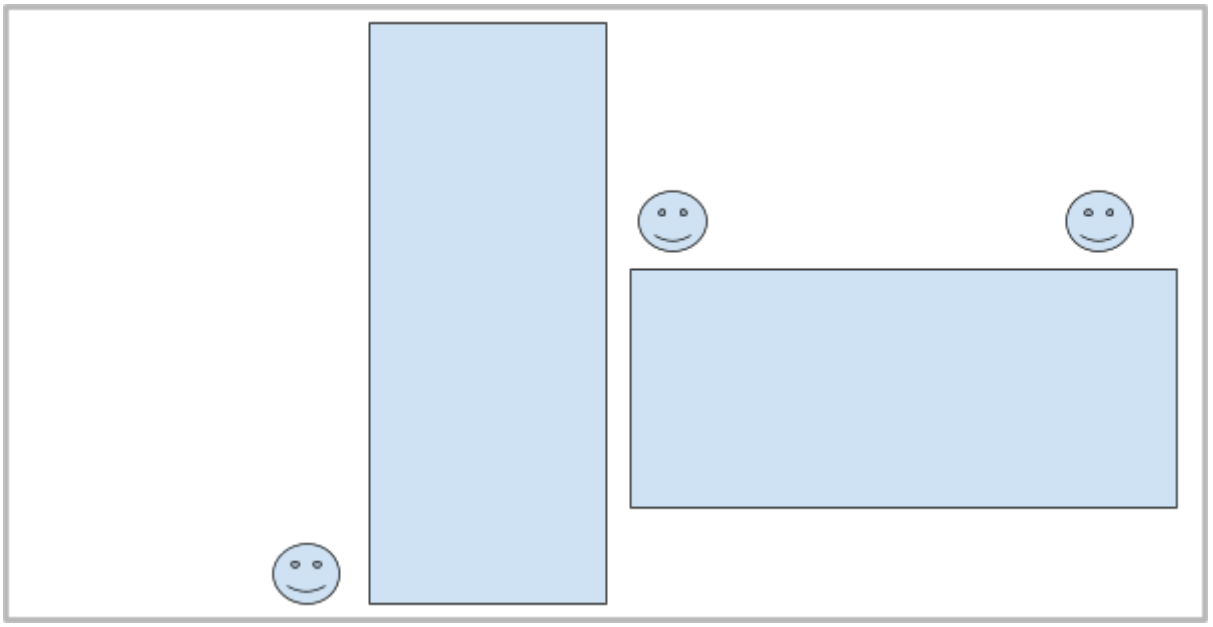
ชื่อบูธ: Software booth
ตำแหน่งบูธ: E1

- Project Programming Methodology - น้องไนน์ (บอส)
- Project Programming Methodology - บีม (บอส)
- AI
    - ให้น้องวาดตัวเลข แล้ว predict https://github.com/supakornPanya/Neural_Vision
    - Project Palm
- ให้ความรู้ software

สรุป: 2 progmeth + 1 ai proj

+ หาแผ่นความรู้เพิ่มในบูธ



Flow ตอนน้องเข้ามาจะทำอะไรบ้าง

Step 1 : ชวนน้องลองวาดตัวเลขบนหน้าจอ (Number CNN)
Step 2: ให้น้องเลือกเล่นเกม 1 เกมจาก 2 เกม และอธิบายการเล่น (Sumo Must Grow: ถ้าชอบแนว Action สนุกๆ,The Roguelike RPG: ถ้าชอบแนววางแผน)
Step 3 : บรีฟเส้นทางการเรียน Software ตั้งแต่ปี 1-3 ให้ข้อมูลการเรียนจริงว่าต้องเจอวิชาอะไรบ้าง (Agile, Database, OS, Security) + ชวนน้องคุยว่าอยากรู้วิชาไหนเป็นพิเศษคนละ 1วิชา ละก็ตอบคำถามที่น้องสงสัย
Step 4 : ภารกิจ ให้น้องๆ วาดเลขใน Number CNN เลขใดก็ได้ ให้ตรง เกิน 90%
ให้เล่นทุกคน คนละ 1 รอบน้าา

รายละเอียดแต่ละ Project

Sumo Must Grow - Progmeth Project

Idea

เกม Action Casual ที่ผสมผสานความเพลินสไตล์ *Fruit Ninja* เข้ากับเกมเพลย์แบบควบคุมตัวละคร (Top-down/Arcade) คอนเซปต์หลักคือ **"You are what you eat"** ผู้เล่นต้องเดินเก็บอาหารดีเพื่อทำคะแนนและอัปเกรดตัวเอง แต่ต้องระวัง "อาหารเสีย" ที่เกมจะสุ่มเกิดมาใกล้ๆ เพื่อหลอกล่อให้เราเผลอหยิบ เป็นเกมที่นอกจากจะสนุกเร้าใจแล้ว ยังแฝงข้อคิดเตือนใจเรื่องโภชนาการในชีวิตจริง ที่บางครั้งเราก็อาจเผลอเลือกสิ่งที่ไม่ดีต่อสุขภาพได้ง่ายๆ

วิธีการเล่น

- **การควบคุมพื้นฐาน:** บังคับตัวละครเดินเก็บของด้วยปุ่ม WASD หรือลูกศร
- **ระบบพลังชีวิตและสารพิษ (HP & Toxicity):** ผู้เล่นจะมีหัวใจทั้งหมด 3 ดวง และมีหลอด "อาหารเสีย" 3 หลอด หากเผลอกินอาหารเสียครบ 3 ครั้ง จะถูกหักหัวใจ 1 ดวง เมื่อหัวใจเหลือ 0 คือ Game Over
- **ระบบขยายร่าง (Growth System):** ยิ่งคะแนนเยอะ ตัวยิ่งใหญ่! เมื่อสะสมแต้มถึง 50, 100, 300, 500 และ 800 ตัวละครจะขยายขนาดขึ้นทีละ 0.5 เท่า ทำให้วงการเก็บอาหารกว้างขึ้น
- **สกิลพุ่งตัว (Speed Boost):** จ่าย 50 คะแนนแล้วกดปุ่ม 'P' เพื่อเร่งความเร็วการเดิน 2 เท่า เป็นเวลา 3 วินาที (คูลดาวน์ 10 วินาที) เอาไว้พุ่งฉกอาหารหรือหนีศัตรู
- **ระบบศัตรูตัวป่วน:** จะมี AI คอยวิ่งแย่งเก็บอาหารดี (แต่ไม่โจมตีเรา) ผู้เล่นต้องตัดสินใจแบบ Trade-off ว่าจะปล่อยมันไปแย่งของ หรือจะยอมสละหัวใจตัวเอง 1 ดวงเพื่อจัดการศัตรูตัวนี้ทิ้ง

ฟีเจอร์เด็ด

- **Intuitive UX/UI (เข้าถึงง่ายแบบไม่ต้องสอน):** ออกแบบ User Experience มาอย่างใส่ใจ ทำให้ผู้เล่นใหม่ที่เพิ่งเข้าเกมสามารถเข้าใจระบบทั้งหมดและเริ่มเล่นได้ทันทีโดยแทบไม่ต้องอ่านคู่มือ
- **Juicy Feedback & Visual Cues (การตอบสนองที่ชัดเจนและมีชีวิตชีวา):**
    - ทุกการกดปุ่มหรือแอคชันจะมี Feedback ตอบสนองทั้งภาพและเสียงอย่างเต็มอิ่ม
    - มี Visual Alert ชัดเจน เช่น ตอนเสียเลือด, กินอาหารเสีย หรือเมื่อสกิลบูสต์ความเร็วหมด ตัวละครจะกะพริบเพื่อแจ้งเตือนสถานะให้ผู้เล่นรู้ทันที
- **Dynamic Pause & Tension UX (รายละเอียดที่ยกระดับอารมณ์เกม):**
    - ระบบ Pause ที่ไม่แข็งทื่อ เมื่อกดหยุดเกม ไอเทมต่างๆ จะยังลอยขยับไปมา (Animation) ทำให้เกมดูมีมิติและไม่รู้สึกว่าเกมค้าง
    - ระบบบิ้วอารมณ์เมื่อเข้าสู่ช่วงวิกฤต (เลือดเหลือ 1 ดวงสุดท้าย) Status Bar จะสั่นเต้นขยับได้ เพื่อสร้างความตระหนักรู้และความเร้าใจให้ผู้เล่นระมัดระวังตัวขั้นสุด

Backpack Adventure - Progmeth Project

Idea

เกมแนว Roguelike Turn-based RPG ที่ผสานกลิ่นอายและได้รับแรงบันดาลใจจาก *Backpack Hero* โดยมีหัวใจหลักคือ "การบริหารจัดการพื้นที่กระเป๋า" ผู้เล่นต้องเอาชีวิตรอดในดันเจี้ยนด้วยการจัดเรียงไอเทมในพื้นที่จำกัดให้เกิดประสิทธิภาพสูงสุดเพื่อนำไปใช้ต่อสู้

วิธีการเล่น

- **ตะลุยดันเจี้ยน:** ผู้เล่นจะต้องเดินทางสำรวจไปทีละห้อง เพื่อตามหาบรรไดหรือทางลงไปยังดันเจี้ยนชั้นถัดไปเรื่อยๆ

- **ระบบต่อสู้ (Turn-based):** เมื่อปะทะศัตรู จะเข้าสู่โหมดสลับเทิร์น ผู้เล่นสามารถโจมตี ป้องกัน หรือใช้ฟังก์ชันต่างๆ ได้โดยการ "คลิกเลือกใช้ไอเทม" ที่จัดเตรียมไว้ในกระเป๋า
- **ระบบรางวัล (Loot & Manage):** หลังจบการต่อสู้ในแต่ละห้อง เกมจะสุ่มดรอปไอเทมใหม่ๆ ให้ผู้เล่นเลือกตัดสินใจเก็บเข้ากระเป๋าเพื่ออัปเกรดความแข็งแกร่ง
- **เป้าหมายสูงสุด:** บุกตะลุยลงไปให้ถึงชั้นสุดท้าย และปราบ Final Boss ให้สำเร็จเพื่อเคลียร์เกม

ฟีเจอร์เด็ด

- **Synergy Inventory System (ระบบจัดกระเป๋าประสานพลัง):** จุดขายหลักของเกมคือไอเทมแต่ละชิ้นจะมีเงื่อนไขการทำงานที่ส่งผลกระทบ (Interact) ต่อกันและกัน
- **Strategic Placement:** การวางไอเทมไม่ใช่แค่การหาที่ว่างใส่ให้พอดี แต่ต้องคิดเชิงกลยุทธ์ การจัดวางไอเทมบางชนิดให้ "ติดกัน" หรืออยู่ในตำแหน่งที่สอดคล้องกัน จะเป็นการสร้าง Synergy ช่วยบัฟสเตตัสให้ไอเทมเหล่านั้นทรงพลังขึ้นอย่างมหาศาล ทำให้ผู้เล่นสนุกไปกับการแก้ปริศนาการจัดกระเป๋าในทุกๆ รอบที่ได้ไอเทมใหม่

Neural Vision - Interactive AI Visualization Project

Idea

โปรเจกต์ Interactive AI ที่จะเปลี่ยน "กล่องดำ" (Black Box) ของระบบ Deep Learning ให้กลายเป็นภาพที่จับต้องได้ เปิดโอกาสให้ผู้ใช้งานได้วาดตัวเลขด้วยลายมือตนเอง และเฝ้าดูการประมวลผลของโครงข่ายประสาท (Neural Network) แบบสดๆ ผ่านหน้าจอ โดยระบบจะไม่ได้แสดงแค่คำตอบ แต่จะแสดงให้เห็นถึง "น้ำหนักความสำคัญ" และ "กระบวนการตัดสินใจ" ของ AI ในทุกๆ ชั้นเลเยอร์

วิธีการเล่น

- **Drawing Phase**: ผู้เล่นวาดตัวเลข 0–9 ลงบนกระดาน Digital Canvas ขนาด 28*28 พิกเซล
- **Configuration**: ผู้เล่นสามารถเลือกปรับเปลี่ยนจำนวนชั้นของเลเยอร์ (Hidden Layers) และฟังก์ชันกระตุ้น (Activation Function) เช่น ReLU หรือ Sigmoid เพื่อเปรียบเทียบผลลัพธ์และความแม่นยำของ AI ในแต่ละรูปแบบ
- **Neural Inference**: เมื่อกดปุ่มทำนาย ระบบจะนำข้อมูลพิกเซลที่วาดไปผ่านโมเดล TensorFlow.js เพื่อประมวลผลในเบราว์เซอร์ทันที
- **Observation**: สังเกตผลการทำนายให้ได้ 90%++

ฟีเจอร์เด็ด

- **Visualize**: การแสดงผลการทำงานของ Neural Network ละการทดลองปรับแต่ง Model Neural Network

บรีฟวิชา software

<u>ปี 1 เทอม 2</u>

- วิชา software engineer
    - เรียน onsite
    - เรียน software life cycle
    - มีการทำโปรเจคโดยต่อยอดจาก frontend, backend
        - ส่งงานเป็น sprint

        - เน้การทำงานร่วมกันโดยใช้ agile
        - เอาสองกลุ่มเดิมจาก fe,be มารวมกัน เลือก 1 ใน 2 หัวข้อแล้วทำ feature เพิ่ม 2 epic
    - มีสอบปรนัย midterm, final
    - มี lab คาบตอนบ่าย
    - เรียนไปทำไม :เรียนรู้วิธีทำงานแบบมืออาชีพ (Agile/Scrum) การบริหารจัดการโปรเจกต์ขนาดใหญ่ร่วมกับผู้อื่น
    - ต่อยอดยังไง : (Agile/Scrum) การบริหารจัดการโปรเจกต์ขนาดใหญ่ร่วมกับผู้อื่นเป็น **Project Manager (PM)** หรือ **Scrum Master**
- วิชา progmeth
    - เขียน JAVA บน intelliJ
    - มี lab ให้ส่งด้วย github
    - มีสอบ 3 รอบ
    - มีการทำโปรเจคจะเป็น game/ application + ใช้ความรู้ที่เรียนมาทั้งหมด
    - เรียนไปทำไม : ฝึกตรรกะการเขียนโปรแกรมเชิงวัตถุ (OOP) ให้ Code สะอาด และจัดการโปรเจกต์ผ่าน GitHub
    - ต่อยอดยังไง : ต่อยอดไปเป็น **Software Developer** หรือนักสร้างเกมที่เขียน Code
- วิชา frontend-backend
    - เรียน online
    - มี assignment ทุกวันทำส่งบน github
    - ไม่มีสอบ
    - มีทำโปรเจคจับกลุ่ม 3 คน
        - เป็นการทำ CRUD ระบบที่กำหนด
        - เริ่มเรียน backend ก่อนต่อด้วย frontend สมาชิกกลุ่มเป็นคนเดิมกับ backend
    - เรียนไปทำไม : รู้วิธีสร้าง "หน้าตา" (UI) และ "หัวใจ" (Server) ของเว็บไซต์ การทำระบบจัดการข้อมูล (CRUD)
    - ต่อยอดยังไง : ก้าวสู่สายงาน **Full-stack Developer** ที่สามารถสร้าง Web App ตั้งแต่ศูนย์จนจบ
- วิชา database system
    - เรียน onsite
    - มีการสอบ midterm, grader SQL, final
    - มีโปรเจคจับกลุ่ม 6 คน
        - ให้ออกแบบฐานข้อมูลของระบบตามหัวข้อที่กำหนด
        - การส่งจะมีให้ส่งเป็นช่วงๆ เรื่อยๆ
    - เรียนไปทำไม : เรียนรู้วิธีการจัดเก็บข้อมูลให้เป็นระเบียบและปลอดภัย (SQL) ไม่ให้ข้อมูลหายหรือมั่ว
    - ต่อยอดยังไง : พื้นฐานสำคัญของ **Data Architect** หรือคนดูแลระบบหลังบ้านของแอปฯ ดังๆ

<u>ปี 2 เทอม 1</u>

- วิชา introduction to data science and data engineering
    - เรียน onsite
    - มีทำ lab
    - สอบ midterm, final

- มีการทำโปรเจคจับกลุ่ม 3 คน
    - จะมีโจทย์มาให้เป็น dataset
    - ต้องมีความรู้ที่เรียนมาทั้งหมดในโปรเจค
        - datasci (ml, pipeline)
        - dataeng (FastAPI, Kafka)
        - visualization (steamlit, pydeck)
- การจัดการข้อมูลมหาศาล (Big Data) ตั้งแต่ต้นน้ำ (Pipeline) จนถึงปลายน้ำ (Visualization)
    - ต่อยอดยังไง : เป็น **Data Engineer** หรือ **Data Scientist** ที่เปลี่ยนตัวเลขให้นำไปทำเงินได้จริง

<u>ปี 2 เทอม 2</u>

- วิชา computer network
    - เรียน online มา onsite วันที่มีแลป (ทำเป็นกลุ่ม 5 คน/เดี่ยว)
    - lab เน้นไปที่การใช้ cisco packet tracer
    - มีสอบทุกสัปดาห์โดยจะเป็นเนื้อหาที่เรียนในสัปดาห์นั้นๆ
    - มีการสอบใช้งาน cisco packet tracer ด้วย
    - เรียนไปทำไม : เข้าใจว่าอินเทอร์เน็ตทำงานยังไง คอมพิวเตอร์คุยกันได้ยังไงผ่านระบบสายและไร้สาย
    - ต่อยอดยังไง : ต่อยอดไปทาง **Network Engineer** หรือคนวางระบบ Infrastructure ของบริษัท
- วิชา operating system
    - เรียน online มา onsite วันที่มีแลป (ทำเป็นกลุ่ม 3 คน)
    - มีสอบทุกสัปดาห์โดยจะเป็นเนื้อหาที่เรียนในสัปดาห์นั้นๆ
    - เรียนไปทำไม : เข้าใจการทำงานของ CPU, Memory และระบบจัดการเครื่อง (Windows/Linux/MacOS)
    - ต่อยอดยังไง : ช่วยให้เขียนโปรแกรมที่รีดประสิทธิภาพเครื่องออกมาได้สูงสุด
- วิชา AI/ML
    - เรียน onsite
    - มีแลปตามแต่ละบทเรียน
    - สอบ midterm, final
    - เนื้อหาจะเริ่มตั้งแต่ให้รู้จัก rule-base AI ที่เป็นตอบคำถามยุคแรกจนถึง เรียน ML, Neural network, LLM
    - เรียนไปทำไม : จากกฎเกณฑ์พื้นฐาน สู่การสอนให้คอมพิวเตอร์ "คิดเองได้" จนถึงยุคของ ChatGPT (LLM)
    - ต่อยอดยังไง : ก้าวสู่สายงาน **AI Engineer** ผู้อยู่เบื้องหลังเทคโนโลยีล้ำสมัยในปัจจุบัน

<u>ปี 3 เทอม 1</u>

- วิชา Computer Security
    - เรียนไปทำไม : เรียนรู้วิธีการป้องกันการเจาะระบบ การเข้ารหัส และช่องโหว่ต่างๆ ของซอฟต์แวร์ and การจัดการทรัพยากรผ่านซอฟต์แวร์ (Virtualization) ซึ่งเป็นหัวใจของระบบ Cloud
    - ต่อยอดยังไง :เป็น **Cybersecurity Expert** (Hacker ฝ่ายดี) คอยปกป้องระบบจากผู้ไม่หวังดี and ต่อยอดสู่สายงาน **Cloud Engineer** หรือ **DevOps** ที่ดูแลแอปฯ ที่มีคนใช้เป็นล้าน

- วิชา Software-Defined Systems

เอกสาร/Project ที่เกี่ยวข้องกับบูธนี้ทั้งหมด (link)

Number CNN:
https://www.bulovic.at/cnn/
แผ่นความรู้ :
https://canva.link/f7ve8jobt27eohn
SumoMustGrow Drive
Backpack Adventure Documentation

กฎการให้คะแนน

- **20 แต้ม :** เข้าบูธและเลือกเล่นเกมฝีมือพี่ๆ (Sumo Must Grow หรือ Backpack Adventure) อย่างน้อย 1 เกม
- **10 แต้ม :** พิชิตภารกิจวาดเลขใน Number CNN ให้มีความแม่นยำ (Confidence) **เกิน 90%**
- สามารถรับคะแนนได้เพียง 1 ครั้งต่อคน (ทั้ง**ProgMeth Gamer** + **CNN**)

ผู้รับผิดชอบ

boss cedt02
zoom cedt02
august cedt03