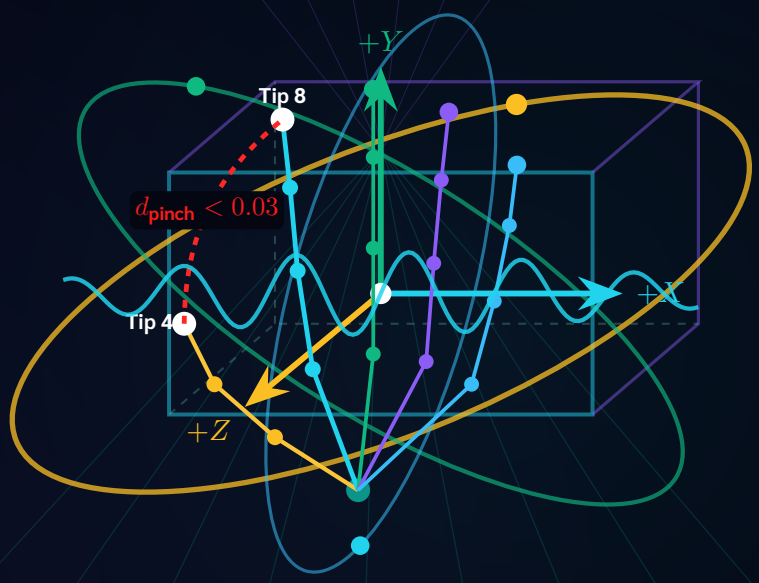
โครงการตำราวิชาการเฉลิมพระเกียรติ
มหาวิทยาลัยราชภัฏรำไพพรรณี
RAMBHAI BARNI RAJABHAT UNIVERSITY ACADEMIC TEXTBOOK SERIES • 2026

SPATIAL COMPUTING & STEM AI

OpenXR และการประมวลผลเชิงพื้นที่
สู่นวัตกรรมห้องปฏิบัติการสะเต็มอัจฉริยะ

OpenXR and Spatial Computing for Next-Generation STEM Laboratories

Physics Engine Simulations • 21-Joint Hand Tracking • Spatial AI Engineering



ผู้ช่วยศาสตราจารย์ ดร.ชีวะ ทัศนา
วท.บ. (ฟิสิกส์ คอมพิวเตอร์และอิเล็กทรอนิกส์) • วท.ม. (ฟิสิกส์ประยุกต์)
ปร.ด. (ฟิสิกส์ประยุกต์)
สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏรำไพพรรณี

พ.ศ. 2569
Masterclass Edition
First Edition

OpenXR และการประมวลผลเชิงพื้นที่

สู่นวัตกรรมห้องปฏิบัติการสะเต็มอัจฉริยะ

OpenXR and Spatial Computing for Next-Generation STEM Laboratories

Physics Engine Simulations • 21-Joint Hand Tracking • Spatial AI Engineering

ผู้ช่วยศาสตราจารย์ ดร.ชีวะ ทัศนา

วท.บ. (ฟิสิกส์ คอมพิวเตอร์และอิเล็กทรอนิกส์)

วท.ม. (ฟิสิกส์ประยุกต์)

ปร.ด. (ฟิสิกส์ประยุกต์)

สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏรำไพพรรณี

พ.ศ. 2569

คำนำ

ความก้าวหน้าอย่างก้าวกระโดดของเทคโนโลยีความเป็นจริงขยาย (Extended Reality หรือ XR) ซึ่งครอบคลุมความเป็นจริงเสมือน (Virtual Reality: VR) ความเป็นจริงเสริม (Augmented Reality: AR) และความเป็นจริงผสม (Mixed Reality: MR) ได้เข้ามามีบทบาทอย่างยิ่งในการปฏิรูปกระบวนการจัดการเรียนรู้วิทยาศาสตร์ เทคโนโลยี วิศวกรรมศาสตร์ และคณิตศาสตร์ หรือสะเต็มศึกษา (STEM Education) ในศตวรรษที่ 21 การเรียนรู้ปรากฏการณ์ทางกายภาพที่มีความซับซ้อน เป็นนามธรรม หรือมีข้อจำกัดด้านความปลอดภัยและงบประมาณในการทดลองจริง สามารถถูกจำลองขึ้นในรูปแบบสามมิติเชิงโต้ตอบที่มีความเที่ยงตรงสูง ช่วยให้ผู้เรียนเกิดความเข้าใจเชิงมโนทัศน์อย่างลึกซึ้งผ่านประสบการณ์ตรง

ในอดีต การพัฒนาแอปพลิเคชัน XR มักเผชิญกับอุปสรรคสำคัญจากปัญหาความแตกแยกของระบบ (Ecosystem Fragmentation) เนื่องจากผู้พัฒนาต้องเขียนโค้ดแยกตามชุดคำสั่งและอุปกรณ์เฉพาะของผู้ผลิตแต่ละราย (Proprietary SDKs) อย่างไรก็ดี การถือกำเนิดขึ้นของมาตรฐานเปิด **OpenXR** โดยความร่วมมือของกลุ่มพันธมิตรระดับโลก Khronos Group ได้สร้างจุดเปลี่ยนครั้งประวัติศาสตร์ ด้วยการกำหนดมาตรฐานกลางระดับสากลที่เชื่อมประสานระหว่างซอฟต์แวร์ประยุกต์และฮาร์ดแวร์ทุกแพลตฟอร์มอย่างเป็นเอกภาพ ส่งผลให้นักการศึกษาและนักพัฒนาสามารถสร้างสรรค์ระบบจำลองการทดลองเสมือนจริงที่ทำงานข้ามแพลตฟอร์มได้อย่างไร้รอยต่อ

ตำราวิชาการเรื่อง **“OpenXR และการประมวลผลเชิงพื้นที่สู่นวัตกรรมห้องปฏิบัติการสะเต็มอัจฉริยะ (OpenXR and Spatial Computing for Next-Generation STEM Laboratories)”** เล่มนี้ ได้รับการรังสรรค์ขึ้นเพื่อเป็นคู่มือเชิงปฏิบัติการและเอกสารวิชาการที่เน้นการนำไปใช้งานได้จริง (Production-Ready) โดยเชื่อมโยงทฤษฎีทางฟิสิกส์ สถาปัตยกรรมคอมพิวเตอร์กราฟิกส์ การคำนวณทางวิทยาศาสตร์ และเทคโนโลยีวิศวกรรมพรอมพ์ปัญญาประดิษฐ์ (AI Engineering Prompts) เข้าด้วยกันอย่างเป็นรูปธรรม เนื้อหาภายในเล่มแบ่งออกเป็น 7 บทเรียนหลัก ครอบคลุมตั้งแต่สถาปัตยกรรมพื้นฐานของ OpenXR, ระบบพิกัดและการติดตามการเคลื่อนที่ 6 องศาอิสระ (6DoF), การปฏิสัมพันธ์เชิงพื้นที่ผ่านการตรวจจับโครงร่างมือ 21 จุดร่วม (21-Joint Skeletal Hand

Tracking), เอนจินฟิสิกส์เชิงตัวเลข, การประมวลผลทางวิทยาศาสตร์ด้วย Python และ PyOpenXR, การบูรณาการคอมพิวเตอร์วิทัศน์และ AI, ตลอดจนกรณีศึกษาห้องปฏิบัติการเสมือนจริงขั้นสูงที่ผู้อ่านสามารถนำโค้ดและชุดคำสั่งพรอมพ์ไปประยุกต์ใช้งานได้ทันที

ผู้เขียนหวังเป็นอย่างยิ่งว่า ตำราเล่มนี้จะเป็นประโยชน์ต่อนักศึกษา คณาจารย์ นักวิจัย ตลอดจนผู้สนใจในการประยุกต์ใช้เทคโนโลยีความเป็นจริงเสมือนเพื่อการศึกษา และเป็นส่วนหนึ่งในการขับเคลื่อนการจัดการศึกษาสะเต็มเชิงรุกให้ก้าวสู่ระดับสากลอย่างยั่งยืน

ผู้ช่วยศาสตราจารย์ ดร.ชีวะ ทัศนา
สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏรำไพพรรณี

กิตติกรรมประกาศ

ตำราวิชาการเรื่อง **OpenXR และการประมวลผลเชิงพื้นที่สู่นวัตกรรมห้องปฏิบัติการสะเต็มอัจฉริยะ** เล่มนี้สำเร็จลุล่วงได้ด้วยความกรุณาสนับสนุนและความช่วยเหลืออย่างจริงใจจากหลายท่านและหลายองค์กร ผู้เขียนขอขอบพระคุณอย่างสูงมา ณ โอกาสนี้

ขอขอบพระคุณ **มหาวิทยาลัยราชภัฏรำไพพรรณี** ที่ให้ทุนสนับสนุนการจัดทำตำราวิชาการและให้เวลาในการวิจัยและพัฒนาเนื้อหา รวมถึงคณะวิทยาศาสตร์และเทคโนโลยีที่ให้การสนับสนุนทางทรัพยากรและห้องปฏิบัติการ

ขอ ขอบพระคุณ **คณาจารย์ และ นัก วิชาการ** ใน สาขา วิชา ฟิสิกส์ วิทยาการ คอมพิวเตอร์ และวิศวกรรมซอฟต์แวร์ ทั้งในและต่างประเทศ ที่ให้คำปรึกษาและข้อเสนอแนะอันทรงคุณค่า โดยเฉพาะด้านมาตรฐาน Khronos OpenXR และการจำลองทางฟิสิกส์เชิงคำนวณ

ขอขอบพระคุณ **นักศึกษาระดับปริญญาตรีและบัณฑิตศึกษา** ที่ร่วมทดสอบชุดปฏิบัติการ Python และ OpenXR เสมือนจริง และให้ข้อเสนอแนะอันเป็นประโยชน์ยิ่งในการปรับปรุงเนื้อหาให้มีความถูกต้องและมีความเป็นประโยชน์ต่อการเรียนรู้อย่างแท้จริง

ขอขอบพระคุณ **Khronos Group** ผู้พัฒนาและเผยแพร่เอกสารมาตรฐาน OpenXR Specification รวมถึงชุมชนนักพัฒนาโอเพนซอร์ส ได้แก่ Monado, OpenXR SDK, และ PyOpenXR ที่ทำให้การวิจัยและพัฒนาตำรานี้เป็นไปได้

สุดท้ายนี้ ขอขอบพระคุณ **ครอบครัว** ที่ให้กำลังใจและความอดทนตลอดระยะเวลาการจัดทำตำราเล่มนี้ด้วยความรักและความเข้าใจ ผู้เขียนหวังเป็นอย่างยิ่งว่าตำราเล่มนี้จะเป็นประโยชน์ต่อผู้เรียนและวงการวิชาการสะเต็มศึกษาของประเทศไทยต่อไป

ผู้ช่วยศาสตราจารย์ ดร.ชีวะ ทัศนา
สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏรำไพพรรณี

พุทธศักราช 2569

แผนบริหารการสอนประจำวิชา

ข้อมูลทั่วไปของรายวิชา

รหัสวิชา	PHY4308
ชื่อรายวิชา	OpenXR และการประมวลผลเชิงพื้นที่สู่นวัตกรรมห้องปฏิบัติการสะเต็มอัจฉริยะ (OpenXR and Spatial Computing for Next-Generation STEM Laboratories)
จำนวนหน่วยกิต	3(2-2-5) หน่วยกิต (บรรยาย 2 ชั่วโมง ปฏิบัติการ 2 ชั่วโมง ศึกษาด้วยตนเอง 5 ชั่วโมง)
หลักสูตร	วิทยาศาสตรบัณฑิต สาขาวิชาฟิสิกส์ประยุกต์ / นวัตกรรมเทคโนโลยีดิจิทัล
อาจารย์ผู้สอน	ผู้ช่วยศาสตราจารย์ ดร.ชีวะ ทัศนา
ภาคการศึกษา/ชั้นปี	ภาคการศึกษาที่ 1 ชั้นปีที่ 3 หรือ 4

คำอธิบายรายวิชา (Course Description)

ศึกษาหลักการและสถาปัตยกรรมของมาตรฐานเปิด OpenXR ในการพัฒนาสภาพแวดล้อมเสมือนจริงข้ามแพลตฟอร์ม ระบบพิกัดสามมิติและปริภูมิอ้างอิงเชิงพื้นที่ การติดตามตำแหน่งและการหมุนแบบ 6 องศาอิสระ (6DoF) เทคนิคการตรวจจับและติดตามโครงกระดูกมือ 21 จุดร่วม (21-Joint Skeletal Hand Tracking) สำหรับการปฏิสัมพันธ์แบบไร้สัมผัส กลไกการคำนวณของเอนจินฟิสิกส์สำหรับการจำลองปรากฏการณ์ทางสะเต็ม (จลนศาสตร์ พลศาสตร์ การชน และสนามแรง) การเขียนโปรแกรมทางวิทยาศาสตร์ด้วยภาษา Python และ PyOpenXR การบูรณาการโมเดลปัญญาประดิษฐ์และคอมพิวเตอร์วิทัศน์ในการอนุมานท่าทาง การออกแบบและพัฒนาห้องปฏิบัติการวิทยาศาสตร์เสมือนจริง และการประเมินผลการเรียนรู้เชิงรุก

วัตถุประสงค์การเรียนรู้ตามแนวทาง OBE

เมื่อสิ้นสุดการเรียนการสอนรายวิชานี้ ผู้เรียนสามารถ

1. อธิบายสถาปัตยกรรม วงรอบการทำงาน และกลไกของมาตรฐาน OpenXR ได้อย่างถูกต้องตามหลักวิชาการ
2. คำนวณและประยุกต์ใช้การแปลงพิกัดสามมิติ ควอเทอร์เนียน และเมทริกซ์การแปลงในปริภูมิ 6DoF ได้
3. พัฒนาระบบปฏิสัมพันธ์เชิงพื้นที่ด้วยการตรวจจับโครงร่างมือ 21 จุดร่วมเพื่อควบคุมอุปกรณ์ทดลองเสมือนได้
4. สร้างแบบจำลองปรากฏการณ์ทางฟิสิกส์และสะเต็มศึกษาด้วยเอนจินฟิสิกส์เชิงตัวเลขได้อย่างแม่นยำ
5. บูรณาการไลบรารี Python, PyOpenXR, และโมเดลปัญญาประดิษฐ์ในการประมวลผลข้อมูลการทดลองได้
6. ออกแบบ สร้าง และประเมินห้องปฏิบัติการเสมือนจริงด้านสะเต็มศึกษาที่ทำงานข้ามแพลตฟอร์มได้

โครงสร้างแผนการจัดการเรียนรู้รายสัปดาห์ (16 สัปดาห์)

สัปดาห์	บทที่	หัวข้อเนื้อหาและกิจกรรมการเรียนรู้
1–2	1	สถาปัตยกรรม OpenXR, วงรอบการทำงาน และความสำคัญต่อสะเต็มศึกษา
3–4	2	ปริภูมิพิกัด 3 มิติ, การแปลงทางเรขาคณิต และการติดตามการเคลื่อนที่ 6DoF
5–6	3	การปฏิสัมพันธ์เชิงพื้นที่และการตรวจจับโครงร่างมือ 21 จุดร่วม (XR_EXT_hand_tracking)
7–8	4	เอนจินฟิสิกส์, การจำลองวัตถุแข็งเกร็ง, การชน และระบบแรงโน้มถ่วง
9	-	สอบวัดผลกลางภาค (Midterm Assessment)
10–11	5	การเขียนโปรแกรมวิทยาศาสตร์และสะเต็มด้วย Python และ PyOpenXR
12–13	6	การบูรณาการปัญญาประดิษฐ์และคอมพิวเตอร์วิทัศน์สำหรับห้องปฏิบัติการเสมือน
14–15	7	การพัฒนาและประยุกต์ใช้ห้องปฏิบัติการเสมือนสะเต็มศึกษาขั้นสูง (Optics, Waves, Circuits)
16	-	นำเสนอโครงงานนวัตกรรมห้องทดลองเสมือนจริง และสอบวัดผลปลายภาค

การวัดและประเมินผลการเรียนรู้

- การมีส่วนร่วมในชั้นเรียนและการเรียนรู้เชิงรุก (Class Participation & Discussion) 10%
- รายงานผลการทดลองและการเขียนโปรแกรมปฏิบัติการ (Laboratory & Coding Tasks) 30%
- การสอบวัดผลกลางภาค (Midterm Examination) 20%
- โครงงานพัฒนานวัตกรรมห้องปฏิบัติการเสมือนจริงสะเต็มศึกษา (Term Project) 20%
- การสอบวัดผลปลายภาค (Final Examination) 20%

สารบัญ

คำนำ ii

กิตติกรรมประกาศ iv

แผนบริหารการสอนประจำวิชา vi

สารบัญภาพ xiii

สารบัญตาราง xiv

1 พื้นฐานสถาปัตยกรรม OpenXR และระบบประมวลผลเชิงพื้นที่สำหรับสะเต็มศึกษา **1**

แผนบริหารการสอนประจำบทที่ 1 1

1.1 วิวัฒนาการและความท้าทายในสะเต็มศึกษา 3

1.2 โครงสร้างสถาปัตยกรรมมาตรฐานเปิด OpenXR 3

1.3 วงรอบชีวิตของระบบและเฟรมลูป 4

1.4 การบูรณาการในสะเต็มศึกษาสมัยใหม่ 5

สรุปท้ายบทที่ 1 7

แบบฝึกหัดท้ายบทที่ 1 8

2 ปริภูมิพิกัด 3 มิติ และการติดตามการเคลื่อนที่ 6 องศาอิสระ **9**

แผนบริหารการสอนประจำบทที่ 2 9

2.1 ปริภูมิอ้างอิงเชิงพื้นที่ใน OpenXR 11

2.2 การแปลงทางเรขาคณิตและควอเทอร์เนียน 11

2.2.1 นิยามของควอเทอร์เนียน 11
2.2.2 การแปลงระบบพิกัดคาร์ทีเซียนและทรงกลมในระบบติดตาม 3 มิติ 13
2.3 การติดตามการเคลื่อนที่ 6 องศาอิสระ 16
2.3.1 ขั้นตอนวิธี Unprojection และการแมปพิกัดจอภาพ 2D สู่ปริภูมิ 3D 16
2.3.2 การจัดการหน่วยความจำแบบ Zero-GC Vector Pooling สำหรับ 60 FPS 17
2.3.3 การวิเคราะห์ความคลาดเคลื่อนและการถดถอยเชิงเส้นในการแปลงพิกัด 19
สรุปท้ายบทที่ 2 20
แบบฝึกหัดท้ายบทที่ 2 21

3 การโต้ตอบเชิงพื้นที่และการตรวจจับโครงร่างมือ 21 จุดร่วม 22

แผนบริหารการสอนประจำบทที่ 3 22
3.1 ส่วนขยายการติดตามมือและโครงสร้างข้อต่อ 24
3.2 คณิตศาสตร์การอนุมานท่าทางมือ 25
3.2.1 การคำนวณระยะการจีบนิ้ว 25
3.2.2 ปฏิบัติการจำลองเครื่องมือวัดละเอียด: เวอร์เนียร์คาลิปเปอร์เสมือนจริง 26
3.3 วิศวกรรมลูปแบบไร้ขยะหน่วยความจำ 28
สรุปท้ายบทที่ 3 30
แบบฝึกหัดท้ายบทที่ 3 31

4 การจำลองฟิสิกส์เชิงสะเต็มในโลกเสมือนจริง 32

แผนบริหารการสอนประจำบทที่ 4 32
4.1 ระเบียบวิธีเชิงตัวเลขสำหรับการจำลองการเคลื่อนที่ 34
4.1.1 วิธีออยเลอร์แบบดั้งเดิมและแบบกึ่งชัดแจ้ง 34
4.1.2 วิธีแวร์เลต์และความเร็วแวร์เลต์ 34
4.2 การตรวจจับการชนและการคำนวณแรงดล 35
4.3 การจำลองระบบสปริงหน่วงและการสั่นพ้อง 37
4.4 ปฏิบัติการสะเต็มฟิสิกส์เสมือนจริง: ลูกตุ้ม เพนดูลัม และพื้นเอียง 39

4.4.1 การจำลองลูกตุ้มอย่างง่ายและการอนุรักษ์พลังงานในระนาบเฟส . 39
4.4.2 ปฏิบัติการแรงเสียดทานและพื้นเอียงปรับมุมด้วยมือ 40
4.4.3 การจำลองการถ่ายโอนพลังงานความร้อนและกฎการเย็นตัว . . . 41
สรุปท้ายบทที่ 4 43
แบบฝึกหัดท้ายบทที่ 4 44

5 การประมวลผลทางวิทยาศาสตร์และการควบคุมสะเต็มด้วย Python และ PyOpenXR 45
แผนบริหารการสอนประจำบทที่ 5 45
5.1 สถาปัตยกรรม PyOpenXR และการเชื่อมต่อระดับลึก 48
5.2 การคำนวณสมรรถนะสูงและการจัดการบัฟเฟอร์ 48
5.3 สถาปัตยกรรมแบบแยกเธรดการประมวลผล 50
สรุปท้ายบทที่ 5 51
แบบฝึกหัดท้ายบทที่ 5 52

6 การบูรณาการปัญญาประดิษฐ์และคอมพิวเตอร์วิทัศน์ในห้องปฏิบัติการเสมือน 53
แผนบริหารการสอนประจำบทที่ 6 53
6.1 คอมพิวเตอร์วิทัศน์และการตรวจจับจุดสำคัญ 56
6.2 การกำจัดสัญญาณรบกวนด้วยตัวกรองวันยูโร 56
6.3 ระบบผู้ช่วยสอนอัจฉริยะในห้องแล็บเสมือน 57
6.4 กรอบแนวคิดปัญญาประดิษฐ์เชิงพื้นที่และชุดการ์ดการเรียนรู้ 60
6.4.1 การจำแนกท่าทางมือด้วยแผนภูมิต้นไม้ตัดสินใจ 60
6.5 กรณีศึกษา: ระบบจำลองเชิงพื้นที่ Eco-Guardian 3D 61
สรุปท้ายบทที่ 6 62
แบบฝึกหัดท้ายบทที่ 6 63

7 การพัฒนาและการประยุกต์ใช้ห้องปฏิบัติการเสมือนสะเต็มศึกษาขั้นสูง 64
แผนบริหารการสอนประจำบทที่ 7 64
7.1 ห้องปฏิบัติการทัศนศาสตร์เสมือนจริง 66

7.1.1 สมการช่างทำเลนส์และการหักเหของแสง 66
7.2 การจำลองวงโคจรระบบดาวคู่ทางดาราศาสตร์ฟิสิกส์ 67
7.3 ห้องปฏิบัติการวงจรอิเล็กทรอนิกส์และการหมุนเฟสเซอร์ 3 มิติ 68
7.3.1 คณิตศาสตร์ของเฟสเซอร์และอิมพีแดนซ์ในวงจร RLC 68
7.4 การเผยแพร่และการส่งมอบสู่แพลตฟอร์มสากล 69
7.5 ระบบส่งออกรายงานผลการทดลองวิชาการอัตโนมัติ 71
7.6 กิจกรรมเชิงรุก AR STEM Scavenger Hunt 72
สรุปท้ายบทที่ 7 73
แบบฝึกหัดท้ายบทที่ 7 74

A คู่มือติดตั้งและตั้งค่าสภาพแวดล้อม OpenXR 75

B ตารางสูตรคณิตศาสตร์และฟิสิกส์สำหรับ XR 77

C คลังโค้ด Python สำหรับ OpenXR STEM 79

D แหล่งเรียนรู้และบรรณานุกรมเพิ่มเติม 83

บรรณานุกรม 85

ดัชนีสืบค้นคำสำคัญ 87

ประวัติผู้เขียน 88

สารบัญภาพ

1.1 แผนผัง สถาปัตยกรรม OpenXR แสดง ลำดับ ชั้น การ เชื่อม ต่อ ระหว่าง แอปพลิเคชัน ตัวโหลด รันไทม์ และอุปกรณ์ฮาร์ดแวร์ 4

2.1 การหมุนตำแหน่งของอุปกรณ์ทดลองในปริภูมิสามมิติด้วยแกนหมุนหนึ่งหน่วยและมุมหมุน 12

2.2 การแปลงเรขาคณิตระหว่างระบบพิกัดคาร์ทีเซียนสามมิติและระบบพิกัดทรงกลมในสภาพแวดล้อม XR 14

3.1 โครงสร้างลำดับชั้นกระดูกมือ 21 จุดร่วมและการวัดระยะทางจีบเพื่อการปฏิสัมพันธ์ 24

4.1 การเปรียบเทียบเสถียรภาพเชิงตัวเลขระหว่างวิธี Explicit Euler และวิธี Verlet ในระบบการสั่นฮาร์มอนิกอย่างง่าย 35

4.2 แผนภาพแรงเสรี (Free-Body Diagram) ของวัตถุบนพื้นเอียงปรับมุมในห้องปฏิบัติการเสมือน 40

5.1 สถาปัตยกรรม การ เชื่อม ต่อ ของ PyOpenXR ผ่าน ctypes FFI สู่ รันไทม์ ระดับฮาร์ดแวร์ 48

6.1 ขั้นตอนการทำงานของสถาปัตยกรรมโครงข่ายประสาทเทียมตรวจจับโครงร่างมือแบบสองขั้นตอน 56

7.1 การติดตามรังสีแสงสามมิติในระบบจำลองทัศนศาสตร์เสมือนจริง 66

7.2 แผนภาพ เฟส เซอร์ ใน ระนาบ เชิงซ้อน จำลอง 3 มิติ แสดง ความ สัมพันธ์ ระหว่างแรงดันและความถี่ 69

สารบัญตาราง

1.1 การเปรียบเทียบระหว่างระบบดั้งเดิมเฉพาะของผู้ผลิตกับมาตรฐานเปิด OpenXR 3

2.1 คุณลักษณะของปริภูมิอ้างอิงหลักในมาตรฐาน OpenXR 11

3.1 ดัชนีและลำดับชั้นข้อต่อมือ 21 จุดร่วมตามมาตรฐาน MediaPipe และ OpenXR 25

3.2 แนวทางปฏิบัติในการจัดการหน่วยความจำสำหรับลูปคำนวณการติดตามมือ 28

4.1 สภาวะการเคลื่อนที่ของระบบมวลสปริงแบบมีตัวหน่วงในห้องปฏิบัติการเสมือน 37

5.1 การเปรียบเทียบเวลาในการประมวลผลสนามเวกเตอร์ 10,000 จุดในภาษา Python 49

6.1 บทบาทของระบบปัญญาประดิษฐ์ในห้องปฏิบัติการสะเต็มศึกษาเสมือนจริง 58

6.2 เสาหลักของกรอบแนวคิดปัญญาประดิษฐ์เชิงพื้นที่จากชุดการ์ดการเรียนรู้ 60

7.1 เมทริกซ์การประยุกต์ใช้ห้องปฏิบัติการเสมือนจริงในสาขาวิชาต่าง ๆ . . . 68

A.1 ตัวแปรสภาพแวดล้อมสำคัญสำหรับ OpenXR Development 76

B.1 สูตรสำคัญในการคำนวณ 3D Spatial Transform 77

B.2 สมการกลศาสตร์พื้นฐานในการจำลอง 3D Physics 77

D.1 เครื่องมือและไลบรารีที่ใช้ในการพัฒนา OpenXR STEM 83

บทที่ 1

พื้นฐานสถาปัตยกรรม OpenXR และระบบประมวลผลเชิงพื้นที่สำหรับสะเต็มศึกษา

แผนบริหารการสอนประจำบทที่ 1

หัวข้อเนื้อหาประจำบท

1. วิวัฒนาการของเทคโนโลยีความเป็นจริงขยายและความท้าทายในสะเต็มศึกษา
2. โครงสร้างสถาปัตยกรรมมาตรฐานเปิด OpenXR และลำดับชั้นการทำงาน
3. การจัดการวงรอบชีวิตของระบบ (Instance, System, Session, and Frame Loop)
4. การประยุกต์ใช้การประมวลผลเชิงพื้นที่ในการสร้างห้องทดลองวิทยาศาสตร์เสมือนจริง

วัตถุประสงค์เชิงพฤติกรรม

1. อธิบายปัญหาความแตกแยกของระบบและบทบาทของมาตรฐาน OpenXR ได้ถูกต้อง
2. จำแนกหน้าที่ของ Application Layer, Loader, และ Runtime ได้อย่างแม่นยำ
3. วิเคราะห์สถานะและการเปลี่ยนสถานะในวงรอบ Session และ Frame Loop ได้

4. สรุป ประโยชน์ ของ OpenXR ใน การ พัฒนา สื่อ การ เรียน รู้ สะ เต็ม ศึกษา ข้ามแพลตฟอร์มได้

กิจกรรมการเรียนการสอน

1. การบรรยายเชิงปฏิสัมพันธ์พร้อมการสาธิตการทำงานของ OpenXR Architecture
2. กิจกรรม การ วิเคราะห์ เปรียบ เทียบ สถาปัตยกรรม Proprietary SDK กับ OpenXR Standard
3. การฝึกปฏิบัติติดตั้งและกำหนดค่าสภาพแวดล้อมจำลอง Monado และ OpenXR Loader
4. การอภิปรายกลุ่มย่อยเกี่ยวกับการนำเทคโนโลยีเชิงพื้นที่ไปใช้แก้ปัญหาการทดลองจริง

สื่อการเรียนการสอน

1. เอกสารคำสอนและสไลด์บรรยายอิเล็กทรอนิกส์ประจำบทที่ 1
2. แผนผังสถาปัตยกรรม OpenXR Specification จาก Khronos Group
3. ชุดทดสอบ Monado OpenXR Runtime และแอปพลิเคชันตัวอย่าง Hello XR
4. ระบบการจัดการเรียนรู้ดิจิทัล (LMS) สำหรับการส่งงานและอภิปรายผล

การประเมินผล

1. การประเมินความรู้ผ่านแบบทดสอบย่อยเชิงมโนทัศน์ประจำบทที่ 1
2. การประเมินผลการฝึกปฏิบัติการติดตั้งและทดสอบ OpenXR Runtime
3. การประเมินการมีส่วนร่วมในการอภิปรายและการตอบคำถามในชั้นเรียน

1.1 วิวัฒนาการและความท้าทายในสะเต็มศึกษา

การจัดการศึกษาวิทยาศาสตร์และเทคโนโลยีในปัจจุบันเผชิญกับข้อจำกัดหลายประการ การทดลองทางวิทยาศาสตร์จำนวนมากเกี่ยวข้องกับปรากฏการณ์ที่มีอันตรายสูง สารเคมีที่มีพิษ อุปกรณ์ฟิสิกส์พลังงานสูงที่มีราคาแพง หรือปรากฏการณ์ระดับจุลภาคและดาราศาสตร์ที่ไม่สามารถสังเกตได้ด้วยตาเปล่า การนำเทคโนโลยีความเป็นจริงขยาย (Extended Reality หรือ XR) มาประยุกต์ใช้ จึงเป็นหนึ่งในกลยุทธ์สำคัญที่เข้ามาเติมเต็มกระบวนการเรียนรู้เชิงรุก (Active Learning)

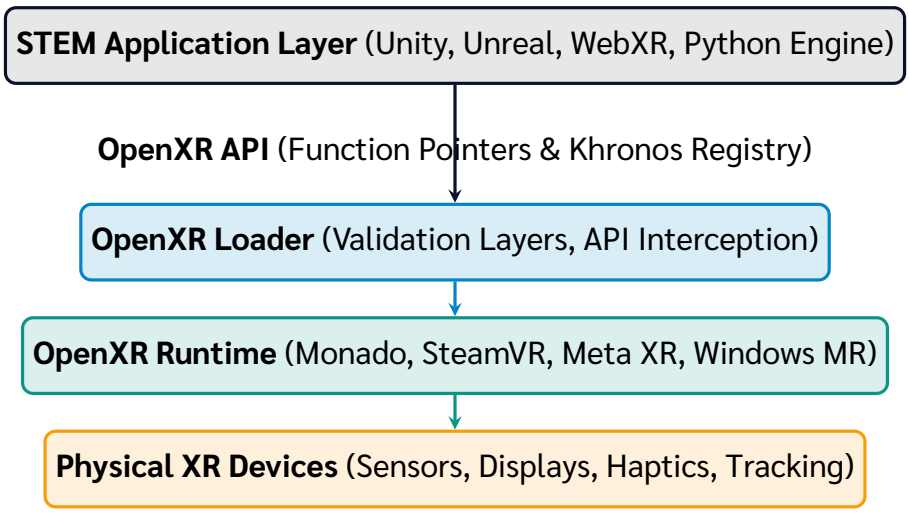
อย่างไรก็ดี ในช่วงทศวรรษที่ผ่านมา นักพัฒนาซอฟต์แวร์การศึกษาต้องเผชิญกับอุปสรรคสำคัญจากปัญหาความแตกแยกของระบบ (Ecosystem Fragmentation) ดังแสดงในตารางที่ 1.1

ตารางที่ 1.1 การเปรียบเทียบระหว่างระบบดั้งเดิมเฉพาะของผู้ผลิตกับมาตรฐานเปิด OpenXR

มิติการเปรียบเทียบ	ระบบดั้งเดิมเฉพาะราย (Proprietary SDKs)	มาตรฐานเปิด (Khronos OpenXR)
การพกพาของโค้ด	ต้องพัฒนาโค้ดใหม่แยกตามยี่ห้อฮาร์ดแวร์	พัฒนาครั้งเดียวสามารถรันได้ทุกอุปกรณ์
สถาปัตยกรรมไดรเวอร์	เชื่อมต่อกับฮาร์ดแวร์โดยตรงผ่านไดรเวอร์เฉพาะ	ผ่านเลเยอร์ OpenXR Loader และ Runtime
การบำรุงรักษา	มีภาระในการอัปเดต SDK หลายตัวขนานกัน	มีมาตรฐาน API กลางที่สม่ำเสมอและมั่นคง
การพัฒนาสื่อสะเต็ม	จำกัดเฉพาะอุปกรณ์ที่สถานศึกษามีงบจัดซื้อ	ใช้งานได้กับทั้งแว่น VR/AR มือถือ และคอมพิวเตอร์

1.2 โครงสร้างสถาปัตยกรรมมาตรฐานเปิด OpenXR

มาตรฐาน OpenXR ซึ่งได้รับการพัฒนาและกำกับดูแลโดย Khronos Group ถูกออกแบบมาเพื่อแก้ปัญหาดังกล่าว โดยแบ่งโครงสร้างออกเป็น 3 ส่วนหลัก ได้แก่ Application Layer, OpenXR Loader, และ OpenXR Runtime



ภาพ ที่ 1.1 แผนผัง สถาปัตยกรรม OpenXR แสดง ลำดับ ชั้น การ เชื่อม ต่อ ระหว่าง แอปพลิเคชัน ตัวโหลด รันไทม์ และอุปกรณ์ฮาร์ดแวร์

มโนทัศน์สำคัญ (Key Concept) การแยกส่วนต่อประสานกับการปฏิบัติงานจริง (API vs. Runtime Decoupling)

หัวใจ สำคัญ ของ OpenXR คือ การ แยก ส่วน คำ สั่ง มาตรฐาน (OpenXR API) ออก จากการทำงานจริงของฮาร์ดแวร์ (Runtime Implementation) ทำให้โค้ดของห้องทดลองวิทยาศาสตร์เสมือนจริงไม่ต้องอิงกับชุดคำสั่งเฉพาะของผู้ผลิตรายใดเลย เมื่ออุปกรณ์รุ่นใหม่เปิดตัว ผู้ผลิตเพียงสร้าง OpenXR Runtime ที่รองรับมาตรฐาน ตัวแอปพลิเคชันสะเต็มเดิมก็สามารถทำงานบนอุปกรณ์ใหม่ได้ทันที

1.3 วงรอบชีวิตของระบบและเฟรมลูป

การ ทำงาน ของ แอปพลิเคชัน OpenXR ดำเนิน ไป ตาม ลำดับ ขั้น ตอน ที่ มี ความ เคร่งครัด เพื่อ ให้ มั่นใจ ว่าการ จัดสรร ทรัพยากร ฮาร์ดแวร์ การ คำนวณ ตำแหน่ง เซนเซอร์ และการเรนเดอร์ภาพมีความสอดคล้องกับเวลาจริง (Real-Time Synchronization) ดังแสดงตามขั้นตอนต่อไปนี้

1. **การสร้างอินสแตนซ์ (Instance Creation):** การเรียกฟังก์ชัน xrCreateInstance เพื่อเตรียมความพร้อมของ Loader และตรวจสอบส่วนขยายที่ต้องการ
2. **การระบุระบบฮาร์ดแวร์ (System Retrieval):** การเรียกฟังก์ชัน xrGetSystem เพื่อเลือกระบบฮาร์ดแวร์ที่พร้อมทำงาน เช่น รูปแบบ HMD หรือ Handheld

3. **การ สร้าง เซสชัน (Session Lifecycle):** การ เชื่อม ต่อ ระบบ จำลอง ภาพ เข้า กับ หน้าต่าง แสดง ผล ผ่าน xrCreateSession ซึ่ง เซสชัน จะ มี สถานะ ต่าง ๆ เช่น IDLE, READY, SYNCHRONIZED, VISIBLE, และ FOCUSED

4. **วง รอ บการเรนเด อร์ เฟรม (Frame Loop):** การ ประมวล ผล ต่อ เนื่อง ใน แต่ละ เฟรม ซึ่ง ประกอบ ด้วย การ รอ สัญญาณ เฟรม (xrWaitFrame) การ เริ่ม ต้น เฟรม (xrBeginFrame) และการสิ้นสุดเฟรม (xrEndFrame)

ตัวอย่างการคำนวณและการจำลอง การคำนวณอัตราการเรนเดอร์และงบประมาณเวลาต่อเฟรม

ในการสร้างห้องทดลองเสมือนจริงที่ให้อรรถรสสมจริงและไม่ก่อให้เกิดอาการเมารถจากโลกเสมือน (Cyber Sickness) ระบบต้องเรนเดอร์ภาพด้วยความถี่ไม่น้อยกว่า $f = 90$ Hz จงคำนวณเวลางบประมาณสูงสุดต่อหนึ่งเฟรม (Frame Budget) สำหรับการคำนวณฟิสิกส์และการเรนเดอร์ภาพ

วิธีทำ: งบประมาณเวลาทั้งหมดต่อหนึ่งเฟรมคำนวณได้จากส่วนกลับของความถี่ในการเรนเดอร์

$$\Delta t = \frac{1}{f} = \frac{1}{90\ \mathrm{s}^{-1}} \approx 11.11\ \mathrm{ms} \tag{1.1}$$

หากแบ่งเวลาออกเป็นการคำนวณทางฟิสิกส์ 30% การอนุมานท่าทางมือ 20% และการเรนเดอร์ภาพ 50% จะได้งบประมาณเวลาในแต่ละส่วน ดังนี้

$$\Delta t_{\text{physics}} = 0.30 \times 11.11\ \mathrm{ms} = 3.33\ \mathrm{ms} \tag{1.2}$$

$$\Delta t_{\text{tracking}} = 0.20 \times 11.11\ \mathrm{ms} = 2.22\ \mathrm{ms} \tag{1.3}$$

$$\Delta t_{\text{render}} = 0.50 \times 11.11\ \mathrm{ms} = 5.56\ \mathrm{ms} \tag{1.4}$$

การบริหารเวลานี้มีความสำคัญยิ่งในการเขียนเอนจินจำลองวิทยาศาสตร์ เพื่อไม่ให้เกิดปัญหากระตุกหรือเฟรมตก

1.4 การบูรณาการในสะเต็มศึกษาสมัยใหม่

ในการศึกษาฟิสิกส์และวิทยาศาสตร์กายภาพ มาตรฐาน OpenXR ช่วยให้นักเรียนสามารถมองเห็นเส้นแรงแม่เหล็กที่ตามองไม่เห็น สามารถใช้มือหยิบจับอนุภาคในระดับอะตอม หรือสังเกตวิวัฒนาการของระบบดาวคู่ได้อย่างเป็นรูปธรรม สอดคล้องกับงานวิจัยของ ทัศนา และคณะ ในการใช้แบบจำลองทางคณิตศาสตร์และการคำนวณเชิงตัวเลขเพื่ออธิบายปรากฏการณ์ทางกายภาพที่มีความซับซ้อนสูง ซึ่งการจำลองแบบเสมือนจริงช่วยยกระดับการรับรู้ของผู้เรียนได้อย่างมีนัยสำคัญ

พรอมพ์ปัญญาประดิษฐ์เชิงปฏิบัติการ (AI Engineering Prompt): การสร้างสเกเลตันโค้ด OpenXR Minimal Session ด้วย AI

บริบทและเป้าหมาย: สร้างโค้ดเริ่มต้นเชื่อมต่อ OpenXR Loader และเริ่ม Session สำหรับแอปพลิเคชันสะเต็มศึกษา

CLEAR Prompting Template (คัดลอกนำไปใช้งานได้ทันที):

```
Act as an expert Khronos OpenXR Systems Engineer.
Goal: Write a production-grade Python script using pyopenxr for a STEM
    Virtual Lab that:
1. Initializes an XrInstance with application name "STEM_Spatial_Lab"
    version 1.0.
2. Checks available extensions and enables XR_KHR_opengl_enable and
    XR_EXT_hand_tracking.
3. Retrieves the system ID for XR_FORM_FACTOR_HEAD_MOUNTED_DISPLAY.
4. Creates an XrSession and manages the state machine (READY,
    SYNCHRONIZED, FOCUSED).
5. Implements an explicit 90 FPS frame loop with error handling and
    clean resource destruction.
Output: Complete executable Python code with informative English/Thai
    comments.
```

การตรวจสอบส่วนขยายที่รองรับใน OpenXR Runtime ด้วย Python

```python
import openxr as xr

def check_openxr_runtime():
    # ตรวจสอบส่วนขยายทั้งหมดที่รันไทม์รองรับ
    extensions = xr.enumerate_instance_extension_properties()
    print(f"ตรวจพบส่วนขยายทั้งหมด {len(extensions)} รายการ:")
    for ext in extensions:
        print(f" - {ext.extension_name} (v{ext.extension_version})")

    # ตรวจสอบส่วนขยายจำเป็นสำหรับสะเต็มศึกษา
    required = ["XR_KHR_opengl_enable", "XR_EXT_hand_tracking"]
    for req in required:
        supported = any(ext.extension_name == req for ext in
            extensions)
        status = "พร้อมใช้งาน" if supported else "ไม่รองรับ"
        print(f"ส่วนขยาย {req}: {status}")

if __name__ == "__main__":
    check_openxr_runtime()
```

สรุปท้ายบทที่ 1

มาตรฐาน OpenXR ได้สร้างรากฐานสำคัญสำหรับการพัฒนาเทคโนโลยีความเป็นจริงขยาย ด้วยการสร้างสถาปัตยกรรมที่เป็นเอกภาพระหว่างแอปพลิเคชันและฮาร์ดแวร์ การทำความเข้าใจโครงสร้าง วงรอบชีวิตของระบบ และข้อจำกัดด้านเวลาในการเรนเดอร์เป็นขั้นตอนแรกที่จำเป็นอย่างยิ่งในการออกแบบและพัฒนาห้องทดลองวิทยาศาสตร์และสะเต็มศึกษาเสมือนจริงที่มีประสิทธิภาพสูงในบทเรียนถัดไป

แบบฝึกหัดท้ายบทที่ 1

1. จงอธิบายความแตกต่างระหว่าง OpenXR API และ OpenXR Runtime พร้อมยกตัวอย่างชื่อรันไทม์ที่ใช้กันอย่างแพร่หลาย
2. เพราะเหตุใดจึงต้องมีการจำแนกสถานะของ OpenXR Session ออกเป็นหลายระดับ เช่น VISIBLE และ FOCUSED
3. หากต้องการพัฒนาห้องปฏิบัติการเสมือนจริงบนอุปกรณ์ที่ทำงานด้วยอัตราการรีเฟรช 120 Hz จงคำนวณงบประมาณเวลาต่อเฟรม และเสนอแนวทางการจัดสรรเวลาสำหรับการคำนวณฟิสิกส์และการตรวจจับการเคลื่อนไหว

บทที่ 2

ปริภูมิพิกัด 3 มิติ และการติดตามการเคลื่อนที่ 6 องศาอิสระ

แผนบริหารการสอนประจำบทที่ 2

หัวข้อเนื้อหาประจำบท

1. ระบบพิกัดสามมิติและปริภูมิอ้างอิงเชิงพื้นที่ในมาตรฐาน OpenXR
2. การแปลงพิกัดวัตถุแข็งเกร็งด้วยเวกเตอร์การเลื่อนตำแหน่งและเมทริกซ์การหมุน
3. สถาปัตยกรรมการตรวจจับและติดตามตำแหน่ง 6 องศาอิสระ (6DoF)

วัตถุประสงค์เชิงพฤติกรรม

1. จำแนกคุณลักษณะและความแตกต่างของ View, Local, และ Stage Reference Spaces ได้
2. คำนวณการแปลงตำแหน่งและการหมุนในปริภูมิสามมิติด้วยควอเทอร์เนียนได้ถูกต้อง
3. อธิบายหลักการหลอมรวมสัญญาณเซนเซอร์สำหรับการติดตามแบบ 6DoF ได้
4. ประยุกต์ใช้การแปลงพิกัดในการวางตำแหน่งเครื่องมือทดลองเสมือนได้อย่างแม่นยำ

กิจกรรมการเรียนการสอน

1. การบรรยายหลักการทางเรขาคณิตสามมิติและการสาธิตการหมุนด้วยควอเทอร์เนียน
2. กิจกรรมการคำนวณเวกเตอร์ตำแหน่งและการแปลงพิกัดของอุปกรณ์ทดลองเสมือน
3. การฝึกเขียนโปรแกรมกำหนดปริภูมิอ้างอิงและอ่านค่าพิกัด 6DoF ผ่าน OpenXR
4. การทดลองจำลองปัญหาการติดขัดของแกนหมุน (Gimbal Lock) และการแก้ไขด้วยควอเทอร์เนียน

สื่อการเรียนการสอน

1. สไลด์บรรยายอิเล็กทรอนิกส์และเอกสารประกอบการสอนประจำบทที่ 2
2. แบบจำลองเชิงภาพสามมิติแสดงปริภูมิพิกัดและมุมออยเลอร์
3. โค้ดตัวอย่างการสร้าง Reference Space ในภาษา Python และ C++
4. ระบบจำลองเซนเซอร์ติดตามตำแหน่งเสมือน (OpenXR Mock Tracking)

การประเมินผล

1. การตรวจแบบฝึกหัดคำนวณการคูณควอเทอร์เนียนและการแปลงพิกัดสามมิติ
2. การประเมินผลงานโค้ดการสร้าง Spatial Anchor สำหรับอุปกรณ์ทดลองเสมือน
3. การทดสอบความเข้าใจย่อยประจำสัปดาห์

2.1 ปริภูมิอ้างอิงเชิงพื้นที่ใน OpenXR

ในระบบประมวลผลเชิงพื้นที่ วัตถุเสมือนทุกชิ้นต้องมีจุดอ้างอิงตำแหน่งในปริภูมิสามมิติ มาตรฐาน OpenXR กำหนดประเภทของปริภูมิอ้างอิง (Reference Space Types) ไว้อย่างชัดเจนเพื่อรองรับสถานการณ์การใช้งานที่แตกต่างกัน ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 คุณลักษณะของปริภูมิอ้างอิงหลักในมาตรฐาน OpenXR

ชื่อปริภูมิอ้างอิง	จุดกำเนิดและทิศทางอ้างอิง	กรณีการใช้งานในสะเต็มศึกษา
VIEW	จุดกึ่งกลางระหว่างดวงตาทั้งสองข้างของกล้อง	การแสดงผล HUD หรือเมนูที่ติดตามสายตาผู้เรียน
LOCAL	จุดเริ่มต้นตำแหน่งของศีรษะขณะเริ่มทำงาน	การทดลองในท่านั่ง โต๊ะทดลองหน้าผู้เรียน
STAGE	กึ่งกลางพื้นที่ห้องบนพื้นราบจริง (Room-scale)	การเดินสำรวจห้องปฏิบัติการเสมือนจริงขนาดใหญ่

มโนทัศน์สำคัญ (Key Concept) ปริภูมิพิกัดมือขวา (Right-Handed Coordinate System)

มาตรฐาน OpenXR กำหนดให้ใช้ระบบพิกัดมือขวาในหน่วยเมตร (SI Unit) โดยแกน $+X$ ชี้ไปทางขวา แกน $+Y$ ชี้ขึ้นด้านบนในแนวดิ่ง และแกน $+Z$ ชี้ออกมาทางด้านหลัง (ซึ่งทำให้แกน $-Z$ ชี้ตรงไปข้างหน้าในทิศทางการมอง) ความสม่ำเสมอของหน่วยวัดทางฟิสิกส์นี้ช่วยให้การคำนวณสมการการเคลื่อนที่และการจำลองแรงทำได้อย่างตรงไปตรงมา

2.2 การแปลงทางเรขาคณิตและควอเทอร์เนียน

การระบุตำแหน่งและการวางตัวของวัตถุเสมือนในปริภูมิสามมิติประกอบด้วยเวกเตอร์การเลื่อนตำแหน่ง $\mathbf{p} = (x, y, z)^\top$ และการหมุน ซึ่งสามารถแทนได้ด้วยควอเทอร์เนียน $\mathbf{q}$

2.2.1 นิยามของควอเทอร์เนียน

ควอเทอร์เนียนเป็นโครงสร้างพีชคณิต 4 มิติที่ประกอบด้วยส่วนสเกลาร์และส่วนเวกเตอร์

$$\mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} = (w, \mathbf{v}) \tag{2.1}$$

โดยที่หน่วยจินตภาพสอดคล้องกับกฎของแฮมิลตัน

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{2.2}$$

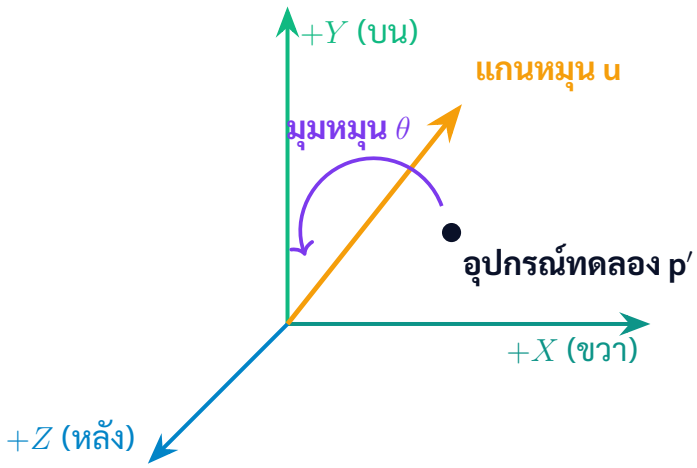
เมื่อ ต้องการ หมุน เวก เตอร์ ตำแหน่ง **p** รอบ แกน หมุน หนึ่ง หน่วย $\mathbf{u} = (u_x, u_y, u_z)$ เป็น มุม θ เรา สร้าง ค วอ เท อร์ เนียน หนึ่ง หน่วย (Unit Quaternion) ได้ดังนี้

$$\mathbf{q} = \cos\left(\frac{\theta}{2}\right) + \mathbf{u}\sin\left(\frac{\theta}{2}\right) \tag{2.3}$$

และการหมุนเวกเตอร์ทำได้ผ่านการคูณแบบสังยุค (Conjugate Multiplication)

$$\mathbf{p}' = \mathbf{qpq}^* \tag{2.4}$$

โดยที่ $\mathbf{q}^* = (w, -\mathbf{v})$ คือควอเทอร์เนียนสังยุค



ภาพที่ 2.1 การหมุนตำแหน่งของอุปกรณ์ทดลองในปริภูมิสามมิติด้วยแกนหมุนหนึ่งหน่วยและมุมหมุน

ตัวอย่างการคำนวณและการจำลอง การคำนวณการหมุนตำแหน่งหัววัดเซนเซอร์ด้วยควอเทอร์เนียน

กำหนดให้เซนเซอร์วัดสนามแม่เหล็กติดตั้งอยู่ที่ตำแหน่งเริ่มต้น **p** $= (0.5, 0, 0)$ m ในปริภูมิห้องปฏิบัติการ เมื่อผู้ใช้หมุนโต๊ะทดลองรอบแกน $+Y$ ในแนวตั้งเป็นมุม $\theta = 90°$ ตามเข็มนาฬิกาเมื่อมองจากด้านบน จงหาพิกัดตำแหน่งใหม่ **p**′ ของเซนเซอร์

วิธีทำ: แกนหมุนคือแกน $+Y$ ดังนั้น **u** $= (0, 1, 0)$ และมุมหมุนตามเข็มนาฬิกาคือ $\theta = -90°$ หรือ $-\frac{\pi}{2}$ rad คำนวณองค์ประกอบของควอเทอร์เนียนหนึ่งหน่วย

$$w = \cos(-45°) = \frac{\sqrt{2}}{2} \tag{2.5}$$

$$\mathbf{v} = (0, 1, 0)\sin(-45°) = \left(0, -\frac{\sqrt{2}}{2}, 0\right) \tag{2.6}$$

ดังนั้น **q** $= \left(\frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2}, 0\right)$ และ **q*** $= \left(\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}, 0\right)$

เมื่อทำการคูณ **p**′ $=$ **qpq*** โดยมอง **p** $= (0, 0.5, 0, 0)$ จะได้

$$\mathbf{p}' = (0, 0, 0, 0.5)\ \text{m} \tag{2.7}$$

นั่นคือ ตำแหน่งใหม่ของหัววัดเซนเซอร์จะย้ายไปอยู่ที่ $x = 0$ m, $y = 0$ m, $z = 0.5$ m ซึ่งสอดคล้องกับการหมุนทางกายภาพจริง

2.2.2 การแปลงระบบพิกัดคาร์ทีเซียนและทรงกลมในระบบติดตาม 3 มิติ

ในการพัฒนาสภาพแวดล้อมเสมือนจริงและการติดตามวัตถุเชิงพื้นที่ การแปลงระหว่างระบบพิกัดคาร์ทีเซียน (Cartesian Coordinates) (x, y, z) และระบบพิกัดทรงกลม (Spherical Coordinates) (r, θ, ϕ) มีบทบาทสำคัญอย่างยิ่งต่อการตรวจจับทิศทางการชี้ของลำแสง (Pointer Ray Casting) จากมือผู้ใช้ ตลอดจนการจัดวางส่วนต่อประสานกับผู้ใช้แบบลอยตัวในลักษณะทรงกลม (Spherical HUD Menu) ที่รักษาระยะห่างคงที่รอบดวงตาหรือศีรษะ

ความสัมพันธ์ในการแปลงจากระบบพิกัดคาร์ทีเซียนสามมิติไปสู่ระบบพิกัดทรงกลมในมาตรฐานกราฟิกคอมพิวเตอร์ (แกน $+Y$ อยู่ในแนวดิ่ง) แสดงได้ดังสมการ

$$r = \sqrt{x^2 + y^2 + z^2} \tag{2.8}$$

$$\theta = \arccos\left(\frac{y}{r}\right) \quad (0 \leq \theta \leq \pi) \tag{2.9}$$

$$\phi = \text{atan2}(z, x) \quad (-\pi < \phi \leq \pi) \tag{2.10}$$

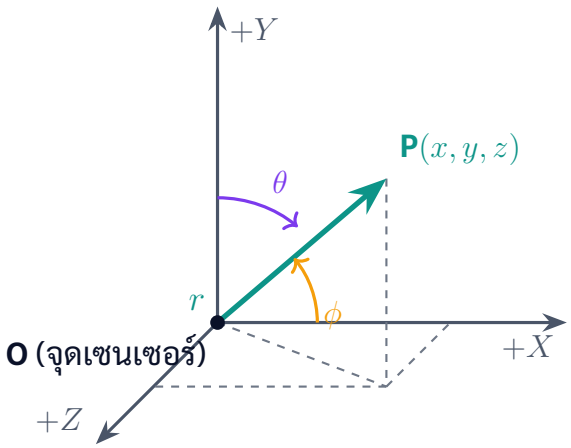
โดยที่ r คือระยะห่างในแนวรัศมีจากจุดกำเนิด θ คือมุมขั้ว (Polar Angle หรือ Zenith Angle) วัดเทียบกับแนวดิ่งแกน $+Y$ และ ϕ คือมุมทิศทางในระนาบแนวนอน (Azimuthal Angle) ในระนาบ XZ

ในทางกลับกัน เมื่อต้องการแปลงจากพิกัดทรงกลมกลับสู่พิกัดคาร์ทีเซียน สำหรับการกำหนดตำแหน่งวัตถุในเอนจินกราฟิก สามารถคำนวณได้จาก

$$x = r \sin\theta \cos\phi \tag{2.11}$$

$$y = r \cos\theta \tag{2.12}$$

$$z = r \sin\theta \sin\phi \tag{2.13}$$



ภาพที่ 2.2 การแปลงเรขาคณิตระหว่างระบบพิกัดคาร์ทีเซียนสามมิติและระบบพิกัดทรงกลมในสภาพแวดล้อม XR

พรอมพ์ปัญญาประดิษฐ์เชิงปฏิบัติการ (AI Engineering Prompt): การคำนวณควอเทอร์เนียนและการสอดแทรกตำแหน่งเชิงมุม (SLERP) ด้วย AI

วัตถุประสงค์: สร้างฟังก์ชัน Python สำหรับการสอดแทรกตำแหน่งเชิงมุมอย่างราบรื่น (Spherical Linear Interpolation: SLERP) ระหว่างควอเทอร์เนียนสองค่า สำหรับใช้ในการปรับทิศทางการมองเห็นและหมุนเครื่องมือทดลองเสมือนในสภาพแวดล้อม XR

โครงสร้าง CLEAR Prompt:

- **Context (บริบท):** ระบบจำลองฟิสิกส์เชิงพื้นที่บน OpenXR ต้องการการหมุนโมเดล 3 มิติแบบนุ่มนวล ไม่กระตุก

- **Logic (ตรรกะ):** ใช้ นิยาม การ สอด แทรก บน ผิว ทรง กลม 4 มิติ $\mathbf{q}(t) = \frac{\sin((1-t)\Omega)}{\sin\Omega}\mathbf{q}_0 + \frac{\sin(t\Omega)}{\sin\Omega}\mathbf{q}_1$ พร้อมจัดการกรณีดอทโปรดักต์ติดลบ (Shortest Path)
- **Expectation (ผลลัพธ์ที่ต้องการ):** ฟังก์ชัน Python คลาส Quaternion พร้อมเมท็อด slerp(q1, q2, t) ที่รองรับ NumPy array และมี Unit test ตรวจสอบความถูกต้องที่ $t = 0, 0.5, 1.0$
- **Action & Restriction (ข้อกำหนด):** ห้ามใช้ external library อื่นนอกจาก numpy และต้องนอร์มัลไลซ์ผลลัพธ์ให้เป็น Unit Quaternion เสมอ

การสอดแทรกการหมุนสามมิติด้วย SLERP ในภาษา Python

```python
import numpy as np

def quaternion_slerp(q0, q1, t):
    """คำนวณการสอดแทรกเชิงมุมทรงกลม
     (SLERP) ระหว่างควอเทอร์เนียน q0 และ q1พารามิเตอร์
    : q0, q1 ในรูปแบบ [x, y, z, w], t ในช่วง [0.0, 1.0]
    """
    q0 = np.array(q0, dtype=np.float64) / np.linalg.norm(q0)
    q1 = np.array(q1, dtype=np.float64) / np.linalg.norm(q1)

    # คำนวณ Dot Product ระหว่างสองควอเทอร์เนียน
    dot = np.dot(q0, q1)

    # หาก Dot Product ติดลบให้กลับเครื่องหมายเพื่อเลือกเส้นทางที่สั้นที่สุด
    if dot < 0.0:
        q1 = -q1
        dot = -dot

    # หากมุมชิดกันมาก (dot > 0.9995) ให้ใช้ Linear Interpolation แทนเพื่อป้องกันหารด้วยศูนย์
    if dot > 0.9995:
        result = q0 + t * (q1 - q0)
        return result / np.linalg.norm(result)

    theta_0 = np.arccos(np.clip(dot, -1.0, 1.0))
    theta = theta_0 * t
    q2 = q1 - q0 * dot
    q2 = q2 / np.linalg.norm(q2)

    return q0 * np.cos(theta) + q2 * np.sin(theta)

```

```
# ตัวอย่างการทดสอบ: หมุนจาก 0 องศาไป 90 องศารอบแกน Y ที่ t = 0.5 (45 องศา)
q_start = [0.0, 0.0, 0.0, 1.0]
q_end = [0.0, np.sin(np.pi/4), 0.0, np.cos(np.pi/4)]
q_mid = quaternion_slerp(q_start, q_end, 0.5)
print(f"Interpolated Quaternion at t=0.5: {np.round(q_mid, 4)}")
```

2.3 การติดตามการเคลื่อนที่ 6 องศาอิสระ

การติดตามแบบ 6DoF (Six Degrees of Freedom) ประกอบด้วยการเคลื่อนที่เชิงเส้น 3 ทิศทาง (X, Y, Z) และการหมุนเชิงมุม 3 ทิศทาง (Yaw, Pitch, Roll) ในอุปกรณ์ XR สมัยใหม่ การติดตามอาศัยการหลอมรวมสัญญาณเซนเซอร์ (Sensor Fusion) ระหว่างหน่วยวัดความเฉื่อย (Inertial Measurement Unit: IMU) ที่ทำงานด้วยความถี่สูง (>1000 Hz) ร่วมกับกล้องติดตามทัศนวิสัยภายนอกสู่ภายใน (Inside-Out Optical Tracking) ที่ทำงานด้วยความถี่ 60 Hz เพื่อขจัดปัญหาการเลื่อนไถลสะสม (Drift Elimination)

การสร้างปริภูมิอ้างอิง Stage ใน OpenXR ด้วยภาษา Python

```
import openxr as xr

# กำหนดพารามิเตอร์การสร้าง Reference Space
create_info = xr.ReferenceSpaceCreateInfo(
    reference_space_type=xr.ReferenceSpaceType.STAGE,
    pose_in_reference_space=xr.Posef(
        orientation=xr.Quaternionf(0.0, 0.0, 0.0, 1.0),
        position=xr.Vector3f(0.0, 0.0, 0.0)
    )
)

# สร้าง Spatial Reference Space จากเซสชัน
stage_space = xr.create_reference_space(session, create_info)
print("Stage Space initialized successfully for Room-Scale STEM Lab."
    )
```

2.3.1 ขั้นตอนวิธี Unprojection และการแมปพิกัดจอภาพ 2D สู่ปริภูมิ 3D

เมื่อผสานการทำงานระหว่างการตรวจจับวัตถุหรือมือผ่านกล้อง 2 มิติ (เช่น การใช้เว็บแคมร่วมกับโมเดลการประมวลผลภาพ) เพื่อควบคุมและวางตำแหน่งเครื่องมือ

ทดลองในปริภูมิ 3 มิติ จำเป็นต้องมีขั้นตอนวิธี **Camera Unprojection** เพื่อย้อนการฉายภาพจากพิกัดระนาบจอภาพ (x_s, y_s) ในหน่วยพิกเซล กลับไปเป็นลำแสงเวกเตอร์สามมิติในปริภูมิโลก (World Space)

ขั้นตอนวิธีเริ่มต้นจากการแปลงพิกัดพิกเซลบนหน้าจอขนาดความกว้าง W และความสูง H ให้อยู่ในระบบพิกัดอุปกรณ์นอร์มัลไลซ์ (Normalized Device Coordinates: NDC) ในช่วง $[-1, 1]$

$$x_{\text{ndc}} = \frac{2x_s}{W} - 1 \tag{2.14}$$

$$y_{\text{ndc}} = 1 - \frac{2y_s}{H} \tag{2.15}$$

จากนั้นนำเวกเตอร์พิกัด $\mathbf{v}_{\text{ndc}} = (x_{\text{ndc}}, y_{\text{ndc}}, z_{\text{ndc}}, 1)^\top$ โดยเลือก $z_{\text{ndc}} = 1.0$ (ที่ระนาบตัดภาพด้านไกล หรือ Far Clipping Plane) คูณด้วยเมทริกซ์ผกผันของการฉายภาพและการมองเห็น $\mathbf{M}_{\text{inv}} = (\mathbf{PV})^{-1}$

$$\mathbf{v}_{\text{world}} = (\mathbf{PV})^{-1} \begin{pmatrix} x_{\text{ndc}} \\ y_{\text{ndc}} \\ 1.0 \\ 1.0 \end{pmatrix} \tag{2.16}$$

เมื่อแปลงกลับจากพิกัดเอกพันธุ์ (Homogeneous Coordinates) โดยการหารด้วยองค์ประกอบ w จะได้พิกัดปลายทาง $\mathbf{P}_{\text{far}}$ และสามารถคำนวณเวกเตอร์ทิศทางของลำแสงหนึ่งหน่วย $\mathbf{d}_{\text{ray}}$ จากตำแหน่งกล้อง $\mathbf{C}_{\text{cam}}$ ได้ดังนี้

$$\mathbf{d}_{\text{ray}} = \frac{\mathbf{P}_{\text{far}} - \mathbf{C}_{\text{cam}}}{\|\mathbf{P}_{\text{far}} - \mathbf{C}_{\text{cam}}\|} \tag{2.17}$$

วัตถุเสมือนที่ระยะความลึก d_{depth} จากตำแหน่งกล้องจึงมีพิกัด 3 มิติในโลกจำลองคือ

$$\mathbf{P}_{\text{target}} = \mathbf{C}_{\text{cam}} + d_{\text{depth}}\mathbf{d}_{\text{ray}} \tag{2.18}$$

2.3.2 การจัดการหน่วยความจำแบบ Zero-GC Vector Pooling สำหรับ 60 FPS

ในแอปพลิเคชันจำลองฟิสิกส์และการประมวลผลเชิงพื้นที่บนอุปกรณ์ XR หรือ WebXR ที่ต้องทำงานด้วยอัตราการเรนเดอร์ต่อเนื่อง 60 ถึง 120 เฟรมต่อวินาที ข้อ

ผิดพลาดที่พบบ่อยในการเขียนโค้ดคือการสร้างอินสแตนซ์ของเวกเตอร์ใหม่ในทุกเฟรม เช่น new Vector3() ภายในฟังก์ชันการวนรอบเรนเดอร์ (Render Loop)

การสร้างอ็อบเจกต์ชั่วคราวอย่างต่อเนื่องจะส่งผลให้ตัวเก็บขยะหน่วยความจำ (Garbage Collector: GC) ต้องหยุดการทำงานของระบบเป็นระยะสั้นๆ (GC Pause หรือ Jank) ทำให้เฟรมเรตตกและผู้ใช้เกิดอาการเวียนศีรษะจากภาพกระตุก (Cyber-sickness)

แนวทางการแก้ไขตามมาตรฐานระดับมืออาชีพคือการใช้สถาปัตยกรรม **Zero-GC Vector Pooling** โดยการสร้างตัวแปรเวกเตอร์สำหรับทดเลขล่วงหน้า (Pre-allocated Scratch Vectors) ไว้นอกลูป แล้วนำกลับมาใช้ซ้ำ (Reuse) ตลอดการทำงาน ดังตัวอย่างโค้ดในตัวอย่างที่ 2.4

การประยุกต์ใช้ Unprojection และ Zero-GC Vector Pool ใน JavaScript/Three.js

```
// ประกาศ Pre-allocated Scratch Vectors ไว้นอก Render Loop เพื่อเลี่ยง GC Spikes
const _camPos = new THREE.Vector3();
const _vec = new THREE.Vector3();
const _targetPos = new THREE.Vector3();

function unprojectScreenToWorld(screenX, screenY, depth, camera, outPosition) {
    const width = window.innerWidth;
    const height = window.innerHeight;

    // 1. แปลงพิกเซลเป็น NDC (-1 ถึง +1)
    const ndcX = (screenX / width) * 2 - 1;
    const ndcY = -(screenY / height) * 2 + 1;

    // 2. กำหนดค่าใน Scratch Vector แทนการ new Vector3
    _vec.set(ndcX, ndcY, 0.5);

    // 3. Unproject ด้วย Matrix ผกผันของ Camera
    _vec.unproject(camera);

    // 4. คำนวณเวกเตอร์ทิศทางรังสีจากตำแหน่งกล้อง
    camera.getWorldPosition(_camPos);
    _vec.sub(_camPos).normalize();

    // 5. คำนวณตำแหน่ง 3D ที่ระยะ depth แล้วบันทึกลง outPosition (Zero-GC)
    outPosition.copy(_camPos).addScaledVector(_vec, depth);
    return outPosition;
}

```

```
// ใช้งานใน Render Loop:
// unprojectScreenToWorld(hand2DX, hand2DY, 0.8, camera,
    virtualBeaker.position);
```

2.3.3 การวิเคราะห์ความคลาดเคลื่อนและการถดถอยเชิงเส้นในการแปลงพิกัด

ในการทดลองสะเต็มศึกษา ข้อมูลพิกัดที่ได้จากการติดตามเซนเซอร์ย่อมมีความคลาดเคลื่อน (Noise) ปะปนอยู่เสมอ การตรวจสอบความสัมพันธ์เชิงเส้นระหว่างค่าพิกัดที่แท้จริง (x) กับค่าพิกัดที่อ่านได้จากระบบเซนเซอร์ (y) ทำได้ด้วยระเบียบวิธีถดถอยเชิงเส้นกำลังสองน้อยที่สุด (Least Squares Linear Regression) เพื่อหาความชัน m และจุดตัดแกน c ในสมการ $y = mx + c$

$$m = \frac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2} \tag{2.19}$$

$$c = \frac{\sum_{i=1}^{N} y_i - m\sum_{i=1}^{N} x_i}{N} \tag{2.20}$$

เมื่อค่าสัมประสิทธิ์การตัดสินใจ $R^2 \to 1.0$ แสดงว่าระบบแปลงพิกัดมีความแม่นยำสูงและมีความถูกต้องเชิงเรขาคณิต ซึ่งข้อมูลการวิเคราะห์นี้สามารถนำไปใช้ปรับเทียบ (Calibrate) อุปกรณ์ทดลองเสมือนได้อย่างเป็นระบบ

สรุปท้ายบทที่ 2

ปริภูมิอ้างอิงสามมิติและคณิตศาสตร์ของควอเทอร์เนียนเป็นเสาหลักสำคัญของการประมวลผลเชิงพื้นที่ในมาตรฐาน OpenXR การใช้หน่วยวัดสากลและระบบพิกัดมือขวาช่วยให้การเชื่อมโยงข้อมูลเซนเซอร์กับการสร้างอุปกรณ์ทดลองทางฟิสิกส์เป็นไปอย่างแม่นยำ การเข้าใจขั้นตอนวิธี Unprojection และการประยุกต์ใช้แนวคิด Zero-GC Vector Pooling เป็นหัวใจสำคัญในการคงประสิทธิภาพการทำงานที่ระดับ 60 ถึง 120 เฟรมต่อวินาที ป้องกันปัญหาการติดขัดของแกนหมุน และสร้างความพร้อมสำหรับการสร้างการปฏิสัมพันธ์ผ่านมือในบทต่อไป

แบบฝึกหัดท้ายบทที่ 2

1. จงเปรียบเทียบข้อดีและข้อเสียของการแทนการหมุนสามมิติด้วยมุมออยเลอร์กับควอเทอร์เนียน พร้อมอธิบายปรากฏการณ์ Gimbal Lock

2. วัตถุมีตำแหน่งเริ่มต้นที่ $\mathbf{p} = (1, 2, -3)$ ในปริภูมิ Local จงแปลงพิกัดเมื่อวัตถุถูกเลื่อนด้วยเวกเตอร์ $\mathbf{t} = (-1, 0, 2)$ และหมุนรอบแกน $+Z$ เป็นมุม 180°

3. อธิบายหลักการทำงานของ Inside-Out Optical Tracking ในการกำจัดความคลาดเคลื่อนสะสมของเซนเซอร์ IMU

4. จุดบนหน้าจอความละเอียด 1920×1080 พิกเซล อยู่ที่พิกัด $(960, 540)$ จงคำนวณพิกัด Normalized Device Coordinates (NDC) ของจุดดังกล่าว

5. อธิบายสาเหตุที่การเรียกคำสั่งสร้างอ็อบเจกต์เวกเตอร์ใหม่ในทุกเฟรม ส่งผลเสียต่อประสบการณ์ใช้งานในระบบเสมือนจริง และอธิบายวิธีแก้ปัญหาด้วย Zero-GC Vector Pooling

บทที่ 3

การโต้ตอบเชิงพื้นที่และการตรวจจับโครงร่างมือ 21 จุดร่วม

แผนบริหารการสอนประจำบทที่ 3

หัวข้อเนื้อหาประจำบท

1. สถาปัตยกรรมส่วนขยายการติดตามมือ (XR_EXT_hand_tracking)
2. ลำดับชั้นกระดูกและพิกัดข้อต่อมือ 21 จุดร่วมในมาตรฐานสากล
3. คณิตศาสตร์การอนุมานท่าทางมือ (Gesture Heuristics: Pinch, Grasp, Point)
4. การออกแบบการปฏิสัมพันธ์แบบไร้สัมผัสกับอุปกรณ์วิทยาศาสตร์เสมือนจริง

วัตถุประสงค์เชิงพฤติกรรม

1. อธิบายกลไกของส่วนขยาย XR_EXT_hand_tracking ใน OpenXR ได้
2. ระบุตำแหน่งและความสัมพันธ์ของข้อต่อมือ 21 จุดร่วมได้อย่างถูกต้อง
3. เขียนขั้นตอนวิธีคำนวณระยะหยิบจับ (Pinch Distance) และการจับยึดวัตถุได้
4. พัฒนาระบบโต้ตอบเพื่อปรับตั้งค่าเครื่องมือทดลองเสมือนจริงได้อย่างราบรื่น

กิจกรรมการเรียนการสอน

1. การบรรยายเชิงทฤษฎีพร้อมสาธิตการตรวจจับข้อต่อมือแบบสด (Live 3D Hand Mesh)
2. กิจกรรมฝึกเขียนโปรแกรมอ่านพิกัดปลายนิ้วชี้และนิ้วหัวแม่มือ
3. การทดลองจำลองการปรับหมุนสเกลไมโครมิเตอร์เสมือนด้วยการจีบนิ้ว
4. การอภิปรายเรื่องเทคนิค Zero-GC เพื่อรักษาอัตราเฟรมคงที่ที่ 60 FPS

สื่อการเรียนการสอน

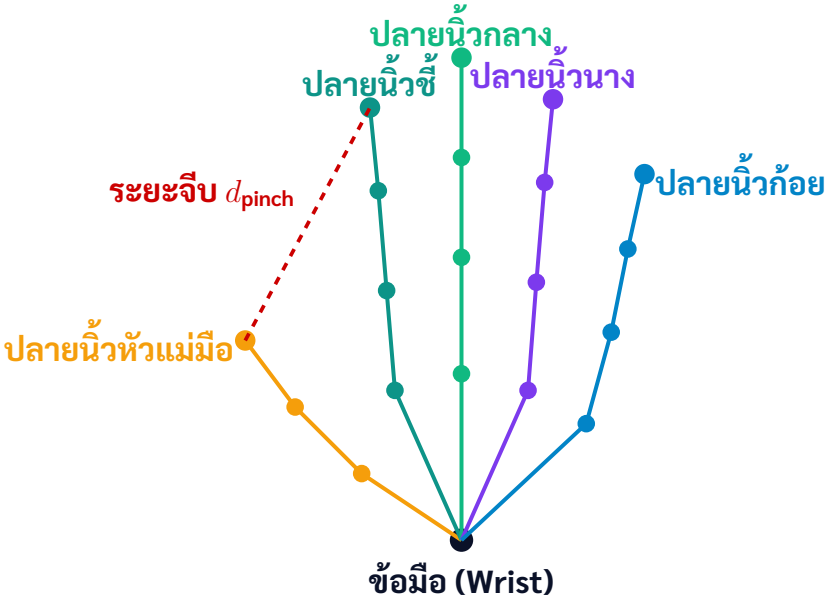
1. สไลด์บรรยายอิเล็กทรอนิกส์และแผนภาพกายวิภาคข้อต่อมือ 21 จุดร่วม
2. สคริปต์ตัวอย่างการใช้งาน OpenXR Hand Tracking และ MediaPipe Hands
3. ชุดทดสอบอุปกรณ์เสมือนจริง (Virtual Vernier Caliper & Optical Slider)
4. วิดีโอสาธิตการจับถืออุปกรณ์เคมีและฟิสิกส์ในห้องแล็บเสมือนจริง

การประเมินผล

1. การประเมินผลงานโค้ดตรวจจับท่าทางมือจีบ (Pinch Gesture Recognition)
2. การประเมินความถูกต้องในการเชื่อมต่อลำดับกระดูกมือเข้ากับวัตถุเสมือน
3. การทดสอบข้อเขียนและแบบฝึกหัดท้ายบท

3.1 ส่วนขยายการติดตามมือและโครงสร้างข้อต่อ

การปฏิสัมพันธ์ในห้องปฏิบัติการเสมือนจริงที่เป็นธรรมชาติสูงสุดคือการใช้มือเปล่าของผู้เรียนโดยไม่ต้องพึ่งพาคอนโทรลเลอร์ มาตรฐาน OpenXR รองรับความสามารถนี้ผ่านส่วนขยายทางการ XR_EXT_hand_tracking ซึ่งทำหน้าที่รายงานพิกัดและทิศทางการหมุนของข้อต่อมือในระดับมิลลิเมตร



ภาพที่ 3.1 โครงสร้างลำดับชั้นกระดูกมือ 21 จุดร่วมและการวัดระยะทางจีบเพื่อการปฏิสัมพันธ์

ตามข้อกำหนดของ OpenXR ข้อต่อมือหลัก 21 จุดร่วม (และส่วนขยายปลีกย่อยรวม 26 จุด) ประกอบด้วยลำดับข้อต่อสำคัญ 5 นิ้ว ได้แก่ นิ้วหัวแม่มือ (Thumb), นิ้วชี้ (Index), นิ้วกลาง (Middle), นิ้วนาง (Ring), และนิ้วก้อย (Little) โดยแต่ละนิ้วประกอบด้วยข้อมือส่วนโคน (Metacarpal), ข้อกลาง (Proximal/Intermediate), ข้อปลาย (Distal), และปลายนิ้ว (Tip) ดังแสดงโครงสร้างและดัชนีจุดสังเกตในตารางที่ 3.1

ตารางที่ 3.1 ดัชนีและลำดับชั้นข้อต่อมือ 21 จุดร่วมตามมาตรฐาน MediaPipe และ OpenXR

ดัชนี	ชื่อข้อต่อ (Joint Name)	ส่วนของมือ	หน้าที่และบทบาทในการจำลองเสมือน
0	WRIST	ข้อมือ	จุดกำเนิดอ้างอิงของมือ (Hand Root Coordinate)
1–4	THUMB_CMC ถึง TIP	นิ้วหัวแม่มือ	จุดที่ 4 ใช้คำนวณการจีบนิ้วและการหนีบวัตถุ
5–8	INDEX_FINGER_MCP ถึง TIP	นิ้วชี้	จุดที่ 8 ใช้เป็นตัวชี้ลำแสงและจุดจับยึดหลัก
9–12	MIDDLE_FINGER_MCP ถึง TIP	นิ้วกลาง	ร่วมคำนวณการกำมือและทิศทางเวกเตอร์ระนาบมือ
13–16	RING_FINGER_MCP ถึง TIP	นิ้วนาง	ตรวจจับการพับงอนิ้วเพื่อระบุท่าทาง Grasp
17–20	PINKY_MCP ถึง TIP	นิ้วก้อย	ตรวจจับขอบนอกสุดของฝ่ามือและระนาบรองรับ

3.2 คณิตศาสตร์การอนุมานท่าทางมือ

ในการควบคุมเครื่องมือทดลองเสมือนจริง ท่าทางพื้นฐานที่สำคัญที่สุดคือการจีบนิ้ว (Pinch Gesture) เพื่อหยิบจับวัตถุขนาดเล็กหรือปรับหมุนปุ่มลูกบิด และการกำมือ (Grasp Gesture) เพื่อยกเคลื่อนย้ายอุปกรณ์ขนาดใหญ่

3.2.1 การคำนวณระยะการจีบนิ้ว

ระยะห่างแบบยูคลิดระหว่างปลายนิ้วหัวแม่มือ $\mathbf{p}_{\text{thumb}}$ (ดัชนี 4) และปลายนิ้วชี้ $\mathbf{p}_{\text{index}}$ (ดัชนี 8) คำนวณได้จาก

$$d_{\text{pinch}} = \|\mathbf{p}_{\text{thumb}} - \mathbf{p}_{\text{index}}\| = \sqrt{(x_4 - x_8)^2 + (y_4 - y_8)^2 + (z_4 - z_8)^2} \tag{3.1}$$

สถานะการจีบนิ้ว S_{pinch} ถูกกำหนดด้วยฟังก์ชันฮิสเทอรีซิส (Hysteresis Thresholding) เพื่อป้องกันการสั่นไหวของสถานะที่จุดขอบเขต

$$S_{\text{pinch}}(t) = \begin{cases} \text{TRUE} & \text{ถ้า } d_{\text{pinch}} \leq d_{\text{enter}} \\ \text{FALSE} & \text{ถ้า } d_{\text{pinch}} \geq d_{\text{exit}} \\ S_{\text{pinch}}(t - \Delta t) & \text{ในกรณีอื่น} \end{cases} \tag{3.2}$$

โดยทั่วไปกำหนดค่าเกณฑ์ $d_{\text{enter}} = 0.025$ m (2.5 cm) และ $d_{\text{exit}} = 0.035$ m (3.5 cm)

เมื่อสถานะ $S_{\text{pinch}} =$ TRUE จุดตำแหน่งกึ่งกลางของการหยิบจับ (Pinch Centroid) $\mathbf{P}_{\text{pinch}}$ ซึ่งใช้เป็นจุดเชื่อมต่อ (Interaction Anchor) สำหรับวัตถุเสมือน คำนวณได้จาก

$$\mathbf{P}_{\text{pinch}} = \frac{\mathbf{p}_{\text{thumb}} + \mathbf{p}_{\text{index}}}{2} = \left(\frac{x_4 + x_8}{2}, \frac{y_4 + y_8}{2}, \frac{z_4 + z_8}{2} \right) \tag{3.3}$$

3.2.2 ปฏิบัติการจำลองเครื่องมือวัดละเอียด: เวอร์เนียร์คาลิปเปอร์เสมือนจริง

หนึ่งในนวัตกรรมการสอนสะเต็มศึกษาผ่านระบบความเป็นจริงขยาย คือการฝึกทักษะปฏิบัติการวัดละเอียดทางวิทยาศาสตร์ด้วย **เวอร์เนียร์คาลิปเปอร์เสมือนจริง (Virtual Vernier Caliper)** ผู้เรียนสามารถใช้นิ้วหัวแม่มือและนิ้วชี้จีบเลื่อนปากวัด (Sliding Jaw) เพื่อประกบวัตถุทดลอง เช่น ทรงกระบอกโลหะหรือลูกแก้วแก้ว

หลักการทำงานของการจำลองเวอร์เนียร์คาลิปเปอร์ประกอบด้วย 2 องค์ประกอบหลัก:

1. **การจับยึดและเลื่อนแกนเดียว (1D Constrained Sliding):** พิกัดการจีบของมือจะถูกฉายลงบนแกนของสเกลหลัก X_{caliper} เท่านั้น ปากวัดจะไม่หลุดออกจากระนาบแกนวัด

2. **การคำนวณค่าความละเอียดและการอ่านค่า (Vernier Scale Reading):** เมื่อปากวัดเคลื่อนที่ไปเป็นระยะทาง L การอ่านค่าประกอบด้วยค่าสเกลหลัก (L_{main}) และค่าขีดสเกลเวอร์เนียร์ที่ตรงกับสเกลหลักพอดี (n)

$$L_{\text{read}} = L_{\text{main}} + (n \times \text{LC}) \tag{3.4}$$

โดยที่ LC คือค่าความละเอียดต่ำสุด (Least Count) ของเครื่องมือวัด เช่น 0.05 mm หรือ 0.01 cm

ตัวอย่างการคำนวณและการจำลอง การคำนวณสถานะการหยิบและอ่านค่าเวอร์เนียร์คาลิปเปอร์เสมือน

ผู้เรียนใช้มือขวาควบคุมเวอร์เนียร์คาลิปเปอร์เสมือนเพื่อวัดความหนาของแผ่นอะคริลิก ปลายนิ้วหัวแม่มืออยู่ที่ $\mathbf{p}_4 = (0.15, 1.00, -0.40)$ m และปลายนิ้วชี้อยู่ที่ $\mathbf{p}_8 = (0.17, 1.01, -0.39)$ m

1. ตรวจสอบว่าเกิดการจีบนิ้วจับปากวัดหรือไม่ (เกณฑ์ $d_{\text{enter}} = 0.030$ m) และระบุจุดกึ่งกลางการจับยึด
2. เมื่อปากวัดถูกเลื่อนไปประกบแผ่นอะคริลิก ขีดศูนย์ของสเกลเวอร์เนียร์อยู่ระหว่าง 18 mm ถึง 19 mm บนสเกลหลัก และขีดที่ 7 บนสเกลเวอร์เนียร์ตรงกับขีดบนสเกลหลักพอดี โดยเวอร์เนียร์มีความละเอียด LC $= 0.05$ mm จงหาค่าความหนาของแผ่นอะคริลิก

วิธีทำ: 1. คำนวณระยะห่างระหว่างนิ้ว

$$\begin{aligned} d_{\text{pinch}} &= \sqrt{(0.15-0.17)^2 + (1.00-1.01)^2 + (-0.40-(-0.39))^2} \\ &= \sqrt{(-0.02)^2 + (-0.01)^2 + (-0.01)^2} = \sqrt{0.0004 + 0.0001 + 0.0001} \\ &= \sqrt{0.0006} \approx 0.0245 \text{ m} = 2.45 \text{ cm} \end{aligned} \tag{3.5}$$

เนื่องจาก $d_{\text{pinch}} = 0.0245$ m < 0.030 m ระบบจึงระบุว่าเกิดการจีบนิ้วเพื่อจับปากวัด และมีตำแหน่งกึ่งกลางการจับยึดที่

$$\begin{aligned} \mathbf{P}_{\text{pinch}} &= \left(\frac{0.15+0.17}{2}, \frac{1.00+1.01}{2}, \frac{-0.40+(-0.39)}{2} \right) \\ &= (0.160, 1.005, -0.395) \text{ m} \end{aligned} \tag{3.6}$$

2. คำนวณค่าความหนาของแผ่นอะคริลิกจากการอ่านค่าเวอร์เนียร์

$$L_{\text{read}} = L_{\text{main}} + (n \times \text{LC}) = 18 \text{ mm} + (7 \times 0.05 \text{ mm}) = 18 + 0.35 = 18.35 \text{ mm} \tag{3.7}$$

แผ่นอะคริลิกมีความหนาเท่ากับ 18.35 mm หรือ 1.835 cm

3.3 วิศวกรรมลูปแบบไร้ขยะหน่วยความจำ

ในการแสดงผลเชิงพื้นที่แบบ 60 ถึง 90 FPS การจัดสรรหน่วยความจำใหม่ซ้ำซากในทุก ๆ เฟรมจะส่งผลให้ระบบ Garbage Collector (GC) ทำงานและหยุดการทำงานชั่วขณะ (Micro-Stutters) ซึ่งทำลายความต่อเนื่องของการทดลอง นักพัฒนาต้องใช้หลักการ Zero-GC Loop Engineering โดยการสร้างโครงสร้างเวกเตอร์และตัวแปรคำนวณไว้ภายนอกลูปการเรนเดอร์ ดังแสดงในตารางที่ 3.1

ตารางที่ 3.2 แนวทางปฏิบัติในการจัดการหน่วยความจำสำหรับลูปคำนวณการติดตามมือ

การกระทำที่ควรหลีกเลี่ยง	ผลกระทบต่อระบบ	แนวทางแก้ไขที่ถูกต้อง (Zero-GC)
สร้าง new Vector3() ในลูป	เกิดขยะอ็อบเจกต์ 90 ตัว/วินาที	จองตัวแปรเวกเตอร์ระดับโกลบอลเพียงครั้งเดียว
ต่อสตริงแสดงผล HUD ซ้ำ ๆ	เพิ่มภาระ Heap Memory	ใช้ Fixed-length Buffers หรือ Text Node เดียว
จัดสรรอาร์เรย์พิกัดใหม่ทุกเฟรม	อัตราเฟรมแกว่งและกระตุก	เขียนทับค่าลงใน TypedArray (Float32Array) เดิม

พรอมพ์ปัญญาประดิษฐ์เชิงปฏิบัติการ (AI Engineering Prompt): การสร้างระบบตรวจจับท่าทางมือแบบ Zero-GC ด้วย AI

วัตถุประสงค์: ออกแบบคลาสสำหรับตรวจจับท่าทางการหยิบจับ (Pinch & Grab) จากพิกัดข้อต่อมือ 21 จุดร่วมที่มีประสิทธิภาพสูง ปราศจากการจัดสรรหน่วยความจำใหม่ในลูปการเรนเดอร์ (Zero Garbage Collection)

โครงสร้าง CLEAR Prompt:

- **Context (บริบท):** ระบบโต้ตอบกับเครื่องมือวิทยาศาสตร์ใน OpenXR บนอุปกรณ์ Standalone ที่มีทรัพยากรหน่วยความจำจำกัด
- **Logic (ตรรกะ):** ใช้กลไก Hysteresis Thresholding เพื่อป้องกันการสั่นไหว พร้อมคำนวณเวกเตอร์ระยะห่างแบบ In-place บน Float32Array หรือ np.ndarray ที่จองล่วงหน้า

- **Expectation (ผลลัพธ์ที่ต้องการ):** คลาส Python หรือ JavaScript สำหรับติดตามสถานะ PINCH_START, PINCHING, PINCH_END พร้อมส่งพิกัดกึ่งกลางการหยิบจับ
- **Action & Restriction (ข้อกำหนด):** ห้ามสร้าง Object หรือ Array ใหม่ในเมท็อด update() อย่างเด็ดขาด และต้องรักษาอัตราการคำนวณที่ 90 FPS (< 1 ms ต่อเฟรม)

การคำนวณระยะจีบนิ้วแบบ Zero-GC ในภาษา Python

```python
import numpy as np

class HandPinchDetector:
    def __init__(self, enter_threshold=0.025, exit_threshold=0.035):
        self.enter_th = enter_threshold
        self.exit_th = exit_threshold
        self.is_pinching = False
        # จองบัฟเฟอร์หน่วยความจำล่วงหน้าเพื่อหลีกเลี่ยง GC
        self._diff = np.zeros(3, dtype=np.float32)

    def update(self, thumb_tip_pos, index_tip_pos):
        # คำนวณแบบ In-place บนอาร์เรย์เดิม
        np.subtract(thumb_tip_pos, index_tip_pos, out=self._diff)
        dist = np.linalg.norm(self._diff)

        if not self.is_pinching and dist <= self.enter_th:
            self.is_pinching = True
        elif self.is_pinching and dist >= self.exit_th:
            self.is_pinching = False

        return self.is_pinching, dist
```

สรุปท้ายบทที่ 3

การตรวจจับโครงร่างมือ 21 จุดร่วมตามมาตรฐาน OpenXR มอบมิติใหม่ของการเรียนรู้สะเต็มศึกษา ช่วยให้ผู้เรียนสามารถควบคุม ปรับแต่ง และหยิบจับอุปกรณ์วิทยาศาสตร์ได้อย่างเป็นธรรมชาติ การประยุกต์ใช้คณิตศาสตร์ของระยะห่างแบบฮิสเทอรีซิสร่วมกับหลักการ Zero-GC ช่วยสร้างประสบการณ์ที่แม่นยำและเสถียร พร้อมสำหรับการเชื่อมต่อเข้ากับเอนจินฟิสิกส์ในบทต่อไป

แบบฝึกหัดท้ายบทที่ 3

1. จงอธิบายบทบาทของฟังก์ชันฮิสเทอรีซิสในการตรวจจับท่าทางการจีบนิ้ว และวิเคราะห์ผลกระทบหากใช้ค่าเกณฑ์ตัดสินเพียงค่าเดียว
2. เขียนสมการคำนวณจุดกึ่งกลาง (Centroid) ของฝ่ามือโดยอ้างอิงจากตำแหน่งข้อต่อโคนนิ้วทั้งห้าจุด
3. เพราะเหตุใดวิศวกรรมลูปแบบไร้ขยะหน่วยความจำ (Zero-GC Loop Engineering) จึงมีความจำเป็นอย่างยิ่งในการพัฒนาแอปพลิเคชันเสมือนจริงบนอุปกรณ์เคลื่อนที่
4. จงระบุดัชนีของจุดข้อต่อปลายนิ้วหัวแม่มือและปลายนิ้วชี้ตามมาตรฐาน MediaPipe Hands และอธิบายวิธีการใช้สองจุดนี้ในการกำหนดจุดกึ่งกลางปฏิสัมพันธ์ (Interaction Anchor)
5. ในการจำลองเวอร์เนียร์คาลิปเปอร์เสมือนจริง หากขีดศูนย์ของสเกลเวอร์เนียร์อยู่ระหว่าง 24 mm และ 25 mm บนสเกลหลัก และขีดที่ 6 บนสเกลเวอร์เนียร์ตรงกับขีดบนสเกลหลัก โดยมีความละเอียด LC $= 0.05$ mm จงคำนวณค่าความยาวที่วัดได้

บทที่ 4

การจำลองฟิสิกส์เชิงสะเต็มในโลกเสมือนจริง

แผนบริหารการสอนประจำบทที่ 4

หัวข้อเนื้อหาประจำบท

1. ระเบียบวิธีเชิงตัวเลขสำหรับการจำลองการเคลื่อนที่ (Euler, Verlet, and RK4)
2. พลศาสตร์ของวัตถุแข็งเกร็ง แรงบิด และโมเมนต์ความเฉื่อยใน 3 มิติ
3. ขั้นตอนวิธีตรวจจับการชนและการคำนวณการดลเพื่อตอบสนองการกระดอน
4. การจำลองระบบการสั่น ระบบสปริงหน่วง และสนามความโน้มถ่วงในโลกเสมือน

วัตถุประสงค์เชิงพฤติกรรม

1. เปรียบเทียบความแม่นยำและความเสถียรของวิธี Euler, Verlet, และ RK4 ได้
2. ประยุกต์กฎการเคลื่อนที่ของนิวตันในการอัปเดตตำแหน่งและความเร็วของวัตถุได้
3. คำนวณการชนแบบยืดหยุ่นและไม่ยืดหยุ่นผ่านสมการการดล (Impulse Response) ได้
4. พัฒนาเอนจินจำลองลูกตุ้มเพนดูลัมและสปริงที่สอดคล้องกับทฤษฎีทางฟิสิกส์ได้

กิจกรรมการเรียนการสอน

1. การบรรยายเชิงคณิตศาสตร์ฟิสิกส์และการสาธิตการสะสมความคลาดเคลื่อนเชิงตัวเลข
2. กิจกรรมการเขียนโค้ดเปรียบเทียบวิถีโค้งของวัตถุภายใต้สนามโน้มถ่วง
3. การทดลองจำลองการชนของมวลในระบบหนึ่งมิติและสองมิติในสภาพแวดล้อมเสมือน
4. การวิเคราะห์ข้อมูลการสั่นแบบหน่วงเปรียบเทียบระหว่างแบบจำลองกับผลการคำนวณวิเคราะห์

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนและสไลด์สรุปสูตรฟิสิกส์เชิงคำนวณ
2. โค้ดจำลอง Physics Loop ในภาษา Python (NumPy) และ WebAssembly
3. กราฟแสดงการอนุรักษ์พลังงานกลรวมในระบบสปริงแบบเรียลไทม์
4. แอปพลิเคชันจำลองการทดลองเสมือนจริงบนมาตรฐาน OpenXR

การประเมินผล

1. การประเมินความถูกต้องของแบบจำลองฟิสิกส์ที่ผู้เรียนพัฒนา
2. การตรวจรายงานการวิเคราะห์พลังงานและเสถียรภาพของตัวอินทิเกรตเชิงตัวเลข
3. การทดสอบข้อเขียนเชิงทฤษฎีและปฏิบัติการคำนวณ

4.1 ระเบียบวิธีเชิงตัวเลขสำหรับการจำลองการเคลื่อนที่

หัวใจสำคัญของห้องปฏิบัติการเสมือนจริงทางสะเต็มศึกษาคือความถูกต้องแม่นยำของกฎฟิสิกส์ที่ควบคุมวัตถุ ในคอมพิวเตอร์ การแก้สมการเชิงอนุพันธ์ของการเคลื่อนที่ $\mathbf{F} = m\frac{d^2\mathbf{x}}{dt^2}$ ต้องอาศัยระเบียบวิธีเชิงตัวเลข (Numerical Integrators)

4.1.1 วิธีออยเลอร์แบบดั้งเดิมและแบบกึ่งชัดแจ้ง

วิธีออยเลอร์แบบชัดแจ้ง (Explicit Euler) ทำการประมาณค่าตำแหน่งและความเร็วในก้าวเวลา Δt ถัดไป ดังนี้

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_n \Delta t \tag{4.1}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_n \Delta t \tag{4.2}$$

แม้วิธีนี้จะคำนวณง่าย แต่มีข้อเสียร้ายแรงคือพลังงานของระบบจะไม่ได้รับการอนุรักษ์ และมักเกิดการระเบิดเชิงตัวเลข (Numerical Explosion) ในระบบที่มีการสั่น จึงนิยมใช้วิธีออยเลอร์กึ่งชัดแจ้ง (Semi-Implicit Euler หรือ Symplectic Euler) แทน

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_n \Delta t \tag{4.3}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_{n+1} \Delta t \tag{4.4}$$

4.1.2 วิธีแวร์เลต์และความเร็วแวร์เลต์

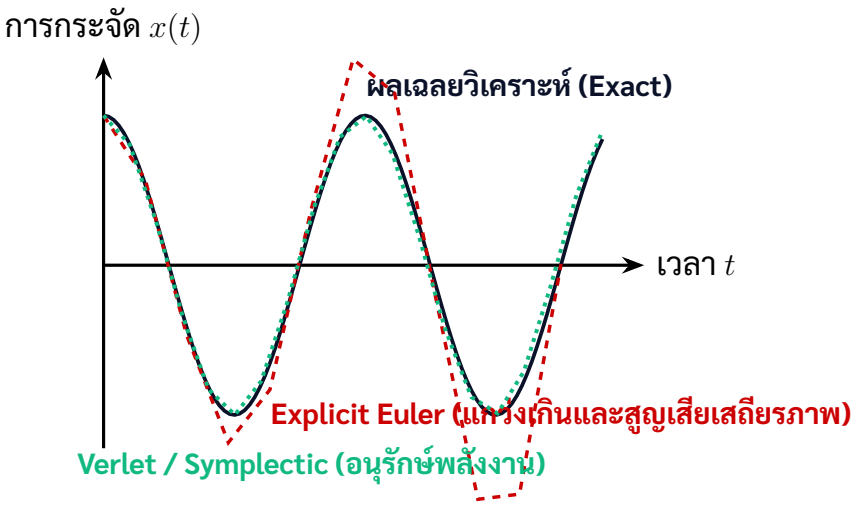
วิธีแวร์เลต์ (Verlet Integration) เป็นระเบียบวิธีที่นิยมใช้อย่างแพร่หลายในเอนจินฟิสิกส์ เนื่องจากมีความเสถียรสูงและอนุรักษ์พลังงานในระบบอนุรักษ์ได้อย่างยอดเยี่ยม

$$\mathbf{x}_{n+1} = 2\mathbf{x}_n - \mathbf{x}_{n-1} + \mathbf{a}_n \Delta t^2 \tag{4.5}$$

ส่วนวิธี Velocity Verlet ช่วยให้สามารถคำนวณความเร็วพร้อมกันได้ในแต่ละก้าวเวลา

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_n \Delta t + \frac{1}{2}\mathbf{a}_n \Delta t^2 \tag{4.6}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \frac{1}{2}(\mathbf{a}_n + \mathbf{a}_{n+1})\Delta t \tag{4.7}$$



ภาพที่ 4.1 การเปรียบเทียบเสถียรภาพเชิงตัวเลขระหว่างวิธี Explicit Euler และวิธี Verlet ในระบบการสั่นฮาร์มอนิกอย่างง่าย

4.2 การตรวจจับการชนและการคำนวณแรงดล

ในการจำลองสะเต็ม วัตถุเสมือนต้องสามารถชนและสะท้อนกลับตามกฎการอนุรักษ์โมเมนตัม การชนระหว่างวัตถุสองชิ้นที่มีมวล m_1, m_2 และความเร็วก่อนชน $\mathbf{v}_1, \mathbf{v}_2$ สามารถคำนวณได้จากขนาดของการดล J ตามแนวตั้งฉากของการชน $\mathbf{n}$

$$J = \frac{-(1+e)(\mathbf{v}_1 - \mathbf{v}_2)\cdot \mathbf{n}}{\frac{1}{m_1} + \frac{1}{m_2}} \tag{4.8}$$

โดยที่ $e \in [0,1]$ คือสัมประสิทธิ์การคืนตัว (Coefficient of Restitution) ความเร็วหลังการชนคำนวณได้จาก

$$\mathbf{v}_1' = \mathbf{v}_1 + \frac{J}{m_1}\mathbf{n} \tag{4.9}$$

$$\mathbf{v}_2' = \mathbf{v}_2 - \frac{J}{m_2}\mathbf{n} \tag{4.10}$$

ตัวอย่างการคำนวณและการจำลอง การคำนวณการชนยืดหยุ่นของลูกกลมโลหะสองลูกบนรางเอียงเสมือน

ลูกกลมโลหะ A มวล $m_A = 0.2$ kg เคลื่อนที่ด้วยความเร็ว $\mathbf{v}_A = (1.5, 0, 0)$ m/s เข้าชนลูกกลม B มวล $m_B = 0.3$ kg ซึ่งหยุดนิ่งอยู่กับที่ ($\mathbf{v}_B = (0, 0, 0)$ m/s) หากการชนเป็นการชนแบบยืดหยุ่นสมบูรณ์ ($e = 1.0$) โดยเวกเตอร์แนวตั้งฉากของการชนคือ $\mathbf{n} = (1, 0, 0)$ จงหาความเร็วของลูกกลมทั้งสองหลังการชน

วิธีทำ: คำนวณความเร็วสัมพัทธ์ในแนวตั้งฉาก

$$(\mathbf{v}_A - \mathbf{v}_B) \cdot \mathbf{n} = (1.5 - 0)(1) = 1.5 \text{ m/s} \tag{4.11}$$

คำนวณผลรวมส่วนกลับของมวล

$$\frac{1}{m_A} + \frac{1}{m_B} = \frac{1}{0.2} + \frac{1}{0.3} = 5.0 + 3.333 = 8.333 \text{ kg}^{-1} \tag{4.12}$$

คำนวณขนาดของการดล J

$$J = \frac{-(1 + 1.0)(1.5)}{8.333} = \frac{-3.0}{8.333} \approx -0.360 \text{ N} \cdot \text{s} \tag{4.13}$$

คำนวณความเร็วหลังการชนของลูกกลมแต่ละลูก

$$\mathbf{v}'_A = (1.5, 0, 0) + \frac{-0.360}{0.2}(1, 0, 0) = (1.5 - 1.80, 0, 0) = (-0.30, 0, 0) \text{ m/s} \tag{4.14}$$

$$\mathbf{v}'_B = (0, 0, 0) - \frac{-0.360}{0.3}(1, 0, 0) = (0 + 1.20, 0, 0) = (1.20, 0, 0) \text{ m/s} \tag{4.15}$$

ผลลัพธ์แสดงว่าลูกกลม A สะท้อนถอยหลังด้วยความเร็ว 0.30 m/s ขณะที่ลูกกลม B เคลื่อนที่ไปข้างหน้าด้วยความเร็ว 1.20 m/s ตรงตามกฎการอนุรักษ์โมเมนตัมและพลังงานกล

4.3 การจำลองระบบสปริงหน่วงและการสั่นพ้อง

ระบบมวล-สปริงแบบมีตัวหน่วง (Damped Harmonic Oscillator) สอดคล้องกับสมการอนุพันธ์อันดับสอง

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = F_{\text{ext}}(t) \tag{4.16}$$

โดยที่ k คือค่านิจสปริง และ c คือสัมประสิทธิ์ความหน่วง ในการจำลองเชิงสะเต็ม ผู้เรียนสามารถปรับเปลี่ยนอัตราส่วนความหน่วง $\zeta = \frac{c}{2\sqrt{mk}}$ เพื่อสังเกตปรากฏการณ์กวัดแกว่งแบบต่าง ๆ ได้แก่ สั่นหน่วงน้อย (Underdamped, $\zeta < 1$), สั่นหน่วงวิกฤต (Critically Damped, $\zeta = 1$), และสั่นหน่วงเกิน (Overdamped, $\zeta > 1$) ดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 สภาวะการเคลื่อนที่ของระบบมวลสปริงแบบมีตัวหน่วงในห้องปฏิบัติการเสมือน

สภาวะความหน่วง	เงื่อนไขพารามิเตอร์	พฤติกรรมทางกายภาพที่สังเกตได้
หน่วงน้อย (Underdamped)	$\zeta < 1.0$	มีการแกว่งสลับไปมาโดยแอมพลิจูดลดลงแบบเอกซ์โพเนนเชียล
หน่วงวิกฤต (Critical)	$\zeta = 1.0$	มวลกลับคืนสู่จุดสมดุลเร็วที่สุดโดยไม่มีการแกว่งข้ามศูนย์
หน่วงเกิน (Overdamped)	$\zeta > 1.0$	มวลเคลื่อนที่เข้าสู่จุดสมดุลอย่างช้า ๆ เนื่องจากแรงต้านทานสูงมาก

พรอมพ์ปัญญาประดิษฐ์เชิงปฏิบัติการ (AI Engineering Prompt): การสร้างเอนจินฟิสิกส์ระบบสปริงหน่วงด้วยระเบียบวิธีเชิงตัวเลขด้วย AI

วัตถุประสงค์: พัฒนาโปรแกรมจำลองฟิสิกส์สำหรับระบบมวล-สปริงแบบมีตัวหน่วง (Damped Harmonic Oscillator) ที่มีความแม่นยำสูงด้วยระเบียบวิธี Velocity Verlet พร้อมวิเคราะห์การอนุรักษ์พลังงานกลรวม

โครงสร้าง CLEAR Prompt:

- **Context (บริบท):** โมดูลการทดลองฟิสิกส์เสมือนจริงบน OpenXR ต้องการคำนวณการเคลื่อนที่ของมวลและแรงดึงกลับแบบเรียลไทม์

- **Logic (ตรรกะ):** แก้สมการ $m\ddot{x} + c\dot{x} + kx = 0$ ด้วย Velocity Verlet อัปเดต $x(t+\Delta t)$ และ $v(t+\Delta t)$ พร้อมคำนวณพลังงานจลน์ $E_k = \frac{1}{2}mv^2$ และพลังงานศักย์ยืดหยุ่น $E_p = \frac{1}{2}kx^2$
- **Expectation (ผลลัพธ์ที่ต้องการ):** คลาส Python SpringDamperSimulator ที่มีเมท็อด step(dt) และคืนค่าสถานะพิกัดพร้อมพลังงานรวม
- **Action & Restriction (ข้อกำหนด):** ต้องรองรับการเปลี่ยนค่าคงที่สปริง k และสัมประสิทธิ์หน่วง c ขณะรัน และมีตัวอย่างโค้ดรันแสดงผล 5 ก้าวเวลา

การจำลองระบบมวลสปริงหน่วงด้วยวิธี Velocity Verlet ในภาษา Python

```python
import numpy as np

class SpringDamperSimulator:
    def __init__(self, mass=1.0, k=50.0, c=0.5, x0=0.2, v0=0.0):
        self.m = mass
        self.k = k
        self.c = c
        self.x = x0
        self.v = v0
        self.a = self._compute_accel(self.x, self.v)

    def _compute_accel(self, x, v):
        # F = -kx - cv => a = F/m
        return (-self.k * x - self.c * v) / self.m

    def step(self, dt):
        # ขั้นที่ 1: อัปเดตตำแหน่งด้วย Velocity Verlet
        self.x += self.v * dt + 0.5 * self.a * (dt ** 2)

        # ขั้นที่ 2: ประมาณความเร็วกึ่งก้าวเพื่อหาความเร่งใหม่
        v_half = self.v + 0.5 * self.a * dt
        a_new = self._compute_accel(self.x, v_half)

        # ขั้นที่ 3: อัปเดตความเร็วเต็มก้าว
        self.v = v_half + 0.5 * a_new * dt
        self.a = a_new

        # คำนวณพลังงานรวม (Mechanical Energy)
        e_kinetic = 0.5 * self.m * (self.v ** 2)
        e_potential = 0.5 * self.k * (self.x ** 2)
        return self.x, self.v, e_kinetic + e_potential
```

```python

# ตัวอย่างการทำงานที่ก้าวเวลา dt = 0.01 วินาที (100 Hz)
sim = SpringDamperSimulator(mass=0.5, k=20.0, c=0.2, x0=0.1)
print(f"Initial: x={sim.x:.4f} m, v={sim.v:.4f} m/s")
for step_idx in range(1, 6):
    x, v, e_total = sim.step(dt=0.01)
    print(f"Step {step_idx}: x={x:.4f} m, v={v:.4f} m/s, E_total={
        e_total:.5f} J")
```

4.4 ปฏิบัติการสะเต็มฟิสิกส์เสมือนจริง: ลูกตุ้ม เพนดูลัม และพื้นเอียง

เพื่อเชื่อมโยงทฤษฎีเชิงตัวเลขเข้ากับการทดลองวิทยาศาสตร์จริง ห้องปฏิบัติการเสมือนจริงบนสถาปัตยกรรม OpenXR สามารถจำลองชุดการทดลองฟิสิกส์มาตรฐานที่ผู้เรียนสามารถโต้ตอบ ปรับค่าพารามิเตอร์ และบันทึกผลได้แบบเรียลไทม์

4.4.1 การ จำลอง ลูก ตุ้ม อย่าง ง่าย และ การ อนุรักษ์ พลังงาน ใน ระนาบเฟส

การเคลื่อนที่ของลูกตุ้มอย่างง่าย (Simple Pendulum) มวล m แขวนด้วยเชือกยาว L สอดคล้องกับสมการไม่เชิงเส้น

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\sin\theta = 0 \tag{4.17}$$

เมื่อมุมเบี่ยงเบน θ มีค่าน้อย ($\sin\theta \approx \theta$) ระบบจะประมาณเป็นการสั่นแบบฮาร์มอนิกอย่างง่ายที่มีคาบการแกว่ง $T = 2\pi\sqrt{\frac{L}{g}}$ แต่ในห้องปฏิบัติการเสมือน ผู้เรียนสามารถดึงลูกตุ้มด้วยมือจนมีมุมกว้าง ($\theta > 30^\circ$) จึงต้องแก้สมการ (4.17) ด้วยระเบียบวิธีเชิงตัวเลขแบบเรียลไทม์

พลังงานกลรวมของลูกตุ้มประกอบด้วยพลังงานจลน์ E_k และพลังงานศักย์โน้มถ่วง E_p

$$E_k = \frac{1}{2}mv^2 = \frac{1}{2}m(L\dot{\theta})^2 \tag{4.18}$$

$$E_p = mgL(1 - \cos\theta) \tag{4.19}$$

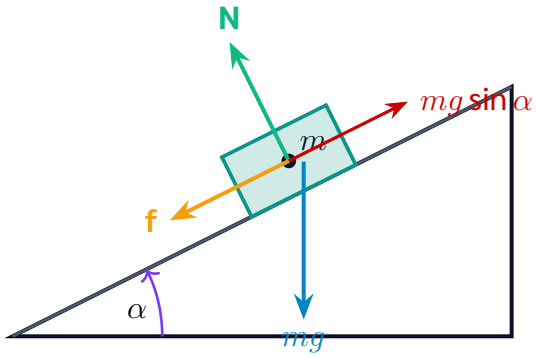
$$E_{\text{total}} = E_k + E_p = \text{คงที่} \tag{4.20}$$

การแสดงผลกราฟพลังงานควบคู่กับระนาบเฟส (Phase Portrait: θ เทียบกับ $\dot{\theta}$) ช่วยให้ผู้เรียนมองเห็นวงโคจรแบบปิด (Closed Orbit) อันเป็นเอกลักษณ์ของระบบอนุรักษ์

4.4.2 ปฏิบัติการแรงเสียดทานและพื้นเอียงปรับมุมด้วยมือ

การทดลองหาค่าสัมประสิทธิ์แรงเสียดทานด้วยพื้นเอียงเสมือนจริง ผู้เรียนสามารถใช้ท่าทางมือ (Hand Gesture) แตะขอบพื้นเอียงแล้วยกปรับมุมเท α ได้อย่างอิสระ วัตถุมวล m ที่วางอยู่บนพื้นเอียงจะอยู่ภายใต้แรงกระทำ 3 แรงหลัก:

1. แรงโน้มถ่วงในแนวดิ่ง $W = mg$ ซึ่งแยกเป็นองค์ประกอบตั้งฉาก $mg\cos\alpha$ และขนานพื้นเอียง $mg\sin\alpha$
2. แรงแนวฉากที่พื้นกระทำต่อวัตถุ $N = mg\cos\alpha$
3. แรงเสียดทาน f ต้านการเลื่อนไถล



ภาพที่ 4.2 แผนภาพแรงเสรี (Free-Body Diagram) ของวัตถุบนพื้นเอียงปรับมุมในห้องปฏิบัติการเสมือน

ขณะที่วัตถุยังอยู่นิ่ง แรงเสียดทานสถิตจะสมดุลกับแรงโน้มถ่วง $f_s = mg\sin\alpha$ จนกระทั่งมุมเอียงเพิ่มขึ้นถึง **มุมวิกฤต (Angle of Repose)** α_c ซึ่งเป็นมุมที่วัตถุเริ่มไถล แรงเสียดทานสถิตจะมีค่าสูงสุด $f_{s,\text{max}} = \mu_s N$

$$mg\sin\alpha_c = \mu_s mg\cos\alpha_c \implies \mu_s = \tan\alpha_c \tag{4.21}$$

เมื่อมุมเอียง $\alpha > \alpha_c$ วัตถุจะไถลลงด้วยความเร่ง a ภายใต้แรงเสียดทานจลน์ $f_k = \mu_k N$

$$a = g(\sin\alpha - \mu_k\cos\alpha) \tag{4.22}$$

ตัวอย่างการคำนวณและการจำลอง การคำนวณสัมประสิทธิ์แรงเสียดทานสถิตและความเร่งบนพื้นเอียง

ในการทดลองเสมือนจริง ผู้เรียนปรับมุมของพื้นเอียงจนพบว่าบล็อกไม้เริ่มไถลลงที่มุม $\alpha_c = 30^\circ$ หลังจากนั้นผู้เรียนปรับมุมเพิ่มขึ้นเป็น $\alpha = 45^\circ$ หากสัมประสิทธิ์แรงเสียดทานจลน์ $\mu_k = 0.40$ และกำหนด $g = 9.8$ m/s^2 จงหา

1. สัมประสิทธิ์แรงเสียดทานสถิต μ_s ระหว่างบล็อกไม้กับพื้นเอียง
2. ความเร่ง a ของบล็อกไม้เมื่อพื้นเอียงทำมุม 45°

วิธีทำ: 1. คำนวณสัมประสิทธิ์แรงเสียดทานสถิตจากมุมวิกฤต

$$\mu_s = \tan\alpha_c = \tan 30^\circ = \frac{1}{\sqrt{3}} \approx 0.577 \tag{4.23}$$

2. คำนวณความเร่งที่มุม $\alpha = 45^\circ$ โดยใช้สมการ (4.22)

$$\begin{aligned} a &= g(\sin 45^\circ - \mu_k \cos 45^\circ) \\ &= 9.8\left(\frac{\sqrt{2}}{2} - 0.40 \times \frac{\sqrt{2}}{2}\right) \\ &= 9.8 \times \frac{\sqrt{2}}{2}(1 - 0.40) = 9.8 \times 0.7071 \times 0.60 \approx 4.16 \text{ m/s}^2 \end{aligned} \tag{4.24}$$

ดังนั้น $\mu_s \approx 0.577$ และบล็อกไม้จะเคลื่อนที่ลงด้วยความเร่ง 4.16 m/s^2

4.4.3 การจำลองการถ่ายโอนพลังงานความร้อนและกฎการเย็นตัว

นอกเหนือจากกลศาสตร์ เอนจินฟิสิกส์เชิงสะเต็มยังรองรับการจำลองอุณหพลศาสตร์ (Thermodynamics) เช่น การให้ความร้อนแก่น้ำในบีกเกอร์ด้วยขดลวดความร้อนไฟฟ้าเสมือน กำลังไฟฟ้าที่ป้อนเข้าสู่ระบบคือ $P = IV$ ซึ่งเปลี่ยนเป็นพลังงานความร้อน $Q = Pt$ ทำให้อุณหภูมิของน้ำมวล m ความจุความร้อนจำเพาะ c เพิ่มขึ้นตามสมการ $Q = mc\Delta T$

ในขณะเดียวกัน ความร้อนจะสูญเสียสู่สิ่งแวดล้อมตามกฎการเย็นตัวของนิวตัน (Newton's Law of Cooling) ส่งผลให้อัตราการเปลี่ยนแปลงอุณหภูมิสอดคล้องกับสมการเชิงอนุพันธ์

$$\frac{dT}{dt} = \frac{P - hA(T - T_{\text{env}})}{mc} \tag{4.25}$$

โดยที่ h คือสัมประสิทธิ์การถ่ายโอนความร้อน A คือพื้นที่ผิวสัมผัส และ T_{env} คืออุณหภูมิห้อง การอินทิเกรตสมการนี้ในลูปฟิสิกส์ช่วยให้ผู้เรียนเข้าใจสภาวะสมดุลความร้อน (Thermal Equilibrium) เมื่ออัตราการรับความร้อนเท่ากับอัตราการสูญเสียความร้อนพอดี ($dT/dt = 0$)

สรุปท้ายบทที่ 4

เอนจินฟิสิกส์ที่มีความเที่ยงตรงสูงเป็นหัวใจของการเรียนรู้สะเต็มศึกษาในโลกเสมือนจริง การเลือกใช้ระเบียบวิธีอินทิเกรตเชิงตัวเลขที่เหมาะสม เช่น Velocity Verlet ร่วมกับการคำนวณการชนด้วยแรงดล การจำลองลูกตุ้มอย่างง่าย และการทดลองพื้นเอียงปรับมุม ช่วยสร้างสภาพแวดล้อมการทดลองที่ถูกต้องตามทฤษฎีทางวิทยาศาสตร์ เปิดโอกาสให้ผู้เรียนสามารถทดลองและสังเกตผลทั้งในเชิงกลศาสตร์และอุณหพลศาสตร์ได้อย่างแม่นยำ

แบบฝึกหัดท้ายบทที่ 4

1. จงอธิบายเหตุผลทางฟิสิกส์ว่าทำไมวิธี Explicit Euler จึงทำให้พลังงานรวมของระบบมวล-สปริงเพิ่มขึ้นอย่างต่อเนื่องเมื่อเวลาผ่านไป

2. กำหนดให้ลูกบอลยางมวล 0.1 kg ตกกระทบพื้นเสมือนด้วยความเร็ว 4.0 m/s และกระดอนขึ้นด้วยความเร็ว 3.2 m/s จงคำนวณสัมประสิทธิ์การคืนตัว e และขนาดของการดล J ที่พื้นกระทำต่อลูกบอล

3. เขียนโปรแกรมจำลองการสั่นแบบหน่วงวิกฤต ($\zeta = 1.0$) ด้วยวิธี Semi-Implicit Euler ในภาษา Python

4. ลูกตุ้มอย่างง่ายมีความยาวเชือก $L = 0.98$ m แขวนอยู่ในสนามโน้มถ่วง $g = 9.8$ m/s^2 จงคำนวณคาบการแกว่งที่มุมขนาดเล็ก และอธิบายว่าทำไมการแกว่งที่มุมกว้าง 60° จึงต้องอาศัยการอินทิเกรตเชิงตัวเลข

5. บล็อกโลหะวางบนพื้นเอียงเสมือนจริง เริ่มไถลลงเมื่อทำมุมเอียง 36.87° จงหาสัมประสิทธิ์แรงเสียดทานสถิต μ_s

6. ขดลวดความร้อนไฟฟ้ากำลัง $P = 500$ W ให้ความร้อนแก่น้ำมวล $m = 0.5$ kg ($c = 4184$ J/(kg $\cdot^\circ$ C)) จงคำนวณเวลาที่ใช้ในการเพิ่มอุณหภูมิจาก 25°C เป็น 75°C หากไม่คิดการสูญเสียความร้อนสู่สิ่งแวดล้อม

บทที่ 5

การประมวลผลทางวิทยาศาสตร์และการควบคุมสะเต็มด้วย Python และ PyOpenXR

แผนบริหารการสอนประจำบทที่ 5

หัวข้อเนื้อหาประจำบท

1. สถาปัตยกรรมตัวเชื่อมต่อ PyOpenXR และการจัดการหน่วยความจำผ่าน ctypes
2. การประมวลผลข้อมูลทางวิทยาศาสตร์สมรรถนะสูงด้วย NumPy และ Zero-Copy Buffers
3. การแยกเธรดการคำนวณฟิสิกส์ออกจากเธรดการเรนเดอร์ภาพ (Multithreaded Architecture)
4. การพล็อตข้อมูลและการแสดงผลสนามเวกเตอร์ทางฟิสิกส์แบบเรียลไทม์ในโลกเสมือน

วัตถุประสงค์เชิงพฤติกรรม

1. อธิบายกลไกการเชื่อมต่อระหว่าง Python C-API กับ OpenXR Native Loader ได้
2. จัดการอาร์เรย์พิกัดสามมิติด้วย NumPy โดยไม่ต้องคัดลอกข้อมูลซ้ำซ้อนได้

3. ออกแบบโครงสร้างมัลติเธรดเพื่อรักษาอัตราเฟรมภาพที่ 90 FPS ได้อย่างเสถียร
4. สร้างระบบแสดงผลเส้นแรงแม่เหล็กและปริมาณฟิสิกส์สามมิติเชิงโต้ตอบได้

กิจกรรมการเรียนการสอน

1. การบรรยายโครงสร้าง PyOpenXR พร้อมการเขียนโค้ดสด (Live Coding Demonstration)
2. กิจกรรมการคำนวณสนามไฟฟ้าและแม่เหล็กรอบขดลวดโซเลนอยด์ด้วย NumPy
3. การฝึกออกแบบ Ring Buffer เพื่อถ่ายโอนข้อมูลระหว่างเธรดฟิสิกส์และเธรดเรนเดอร์
4. การอภิปรายปัญหาและแนวทางแก้ไข Global Interpreter Lock (GIL) ในภาษา Python

สื่อการเรียนการสอน

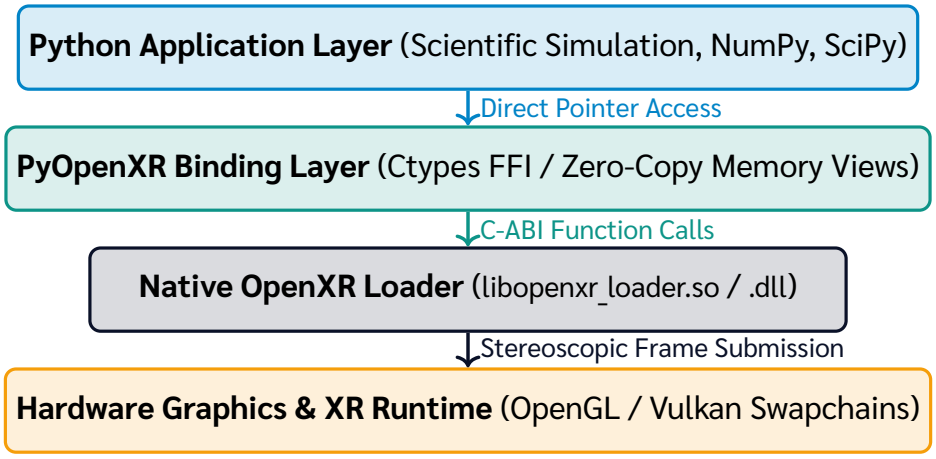
1. สไลด์บรรยายอิเล็กทรอนิกส์และเอกสารโค้ดมาตรฐานประจำบทที่ 5
2. สภาพแวดล้อมเสมือน Python (Virtualenv) ที่ติดตั้ง PyOpenXR และ OpenGL
3. ชุดข้อมูลการจำลองสนามเวกเตอร์และวิถีการเคลื่อนที่ของอนุภาค
4. จอแสดงผลข้อมูลทางวิทยาศาสตร์แบบสามมิติ (3D Spatial Telemetry HUD)

การประเมินผล

1. การตรวจประเมินโปรแกรมคำนวณและแสดงผลสนามแม่เหล็กสามมิติ
2. การวัดประสิทธิภาพอัตราเฟรมและความหน่วงของระบบมัลติเธรดที่พัฒนา
3. การทดสอบข้อเขียนเชิงสถาปัตยกรรมซอฟต์แวร์ทางวิทยาศาสตร์

5.1 สถาปัตยกรรม PyOpenXR และการเชื่อมต่อระดับลึก

ภาษา Python เป็นภาษาหลักในการคำนวณทางวิทยาศาสตร์และปัญญาประดิษฐ์ การผสานพลังของ Python เข้ากับมาตรฐาน OpenXR ผ่านแพ็กเกจ pyopenxr ช่วยให้นักวิจัยและนักการศึกษาสามารถนำโมเดลคณิตศาสตร์ที่ซับซ้อนมาแสดงผลเชิงพื้นที่ได้อย่างรวดเร็ว



ภาพที่ 5.1 สถาปัตยกรรมการเชื่อมต่อของ PyOpenXR ผ่าน ctypes FFI สู่รันไทม์ระดับฮาร์ดแวร์

5.2 การคำนวณสมรรถนะสูงและการจัดการบัฟเฟอร์

ความท้าทายหลักของการใช้ภาษา Python คือความเร็วในการวนลูป ดังนั้น การคำนวณตำแหน่งอนุภาคหรือสนามเวกเตอร์ต้องกระทำผ่านเวกเตอร์ไรเซชัน (Vectorization) ของ NumPy และส่งผ่านข้อมูลไปยัง GPU บัฟเฟอร์โดยตรงโดยไม่ผ่านการคัดลอก (Zero-Copy Transfer)

ตารางที่ 5.1 การเปรียบเทียบเวลาในการประมวลผลสนามเวกเตอร์ 10,000 จุดในภาษา Python

วิธีการเขียนโปรแกรม	เวลาในการประมวลผล (ms)	อัตราเฟรมสูงสุดที่ทำได้ (FPS)
Python Standard For-loop	145.2 ms	~ 6.8 FPS (กระตุกรุนแรง)
NumPy Vectorized Array	2.1 ms	> 90 FPS (ลื่นไหลระดับสากล)
NumPy + Numba JIT Compilation	0.6 ms	> 120 FPS (สมรรถนะสูงสุด)

ตัวอย่างการคำนวณและการจำลอง การคำนวณสนามแม่เหล็กแบบเวกเตอร์รอบเส้นลวดนำกระแสไฟฟ้า

เส้นลวดยาวตรงวางตัวอยู่ตามแนวแกน Z และมีกระแสไฟฟ้า $I = 10$ A ไหลในทิศทาง $+Z$ จงเขียนฟังก์ชันเวกเตอร์คำนวณเวกเตอร์สนามแม่เหล็ก **B** ที่จุดพิกัด **r** $= (x, y, 0)$ และคำนวณค่าสนามแม่เหล็กที่จุด $(0.05, 0.05, 0)$ m

วิธีทำ: ตามกฎของบิโอต์-ซาวาร์ต์และกฎของแอมแปร์ สนามแม่เหล็กจากเส้นลวดตรงยาวมีขนาด

$$B = \frac{\mu_0 I}{2\pi r} \tag{5.1}$$

โดยที่ $\mu_0 = 4\pi \times 10^{-7}$ T · m/A และ $r = \sqrt{x^2 + y^2}$ เวกเตอร์หนึ่งหน่วยในแนวสัมผัสคือ $\hat{\phi} = \left(-\frac{y}{r}, \frac{x}{r}, 0\right)$ ดังนั้น เวกเตอร์สนามแม่เหล็กในพิกัดคาร์ทีเซียนคำนวณได้จาก

$$\mathbf{B}(x, y) = \frac{\mu_0 I}{2\pi(x^2 + y^2)}(-y\hat{\mathbf{i}} + x\hat{\mathbf{j}}) \tag{5.2}$$

ที่จุด $(0.05, 0.05, 0)$ m:

$$r^2 = (0.05)^2 + (0.05)^2 = 0.0025 + 0.0025 = 0.0050 \text{ m}^2 \tag{5.3}$$

$$B_{\text{scale}} = \frac{(4\pi \times 10^{-7})(10)}{2\pi(0.0050)} = \frac{2 \times 10^{-6}}{0.0050} = 4.0 \times 10^{-4} \text{ T} \tag{5.4}$$

จะได้เวกเตอร์สนามแม่เหล็ก

$$\mathbf{B} = 4.0 \times 10^{-4}(-0.05\hat{\mathbf{i}} + 0.05\hat{\mathbf{j}})/\sqrt{0.0050} \approx (-2.83 \times 10^{-5}\hat{\mathbf{i}} + 2.83 \times 10^{-5}\hat{\mathbf{j}}) \text{ T} \tag{5.5}$$

ข้อมูลเวกเตอร์นี้สามารถถูกเรนเดอร์เป็นลูกศร 3 มิติในโลกเสมือนจริงของ OpenXR เพื่อให้ผู้เรียนเห็นทิศทางการวนของสนามแม่เหล็กได้อย่างชัดเจน

5.3 สถาปัตยกรรมแบบแยกเธรดการประมวลผล

เพื่อให้การเรนเดอร์ภาพผ่าน OpenXR ไม่ถูกรบกวนจากการคำนวณทางวิทยาศาสตร์ที่มีความซับซ้อน เราต้องแยกวงรอบการทำงานออกเป็น 2 เธรดหลัก ได้แก่

1. **Physics Simulation Thread:** คำนวณสมการอนุพันธ์ การเคลื่อนที่ของอนุภาค และการตรวจจับการชน โดยทำงานด้วยความถี่คงที่ (Fixed Timestep เช่น $\Delta t = 0.01$ s)
2. **XR Rendering Thread:** ทำหน้าที่รับข้อมูลพิกัดล่าสุดผ่านโครงสร้างหน่วยความจำแบบล็อกอิสระ (Lock-Free Shared Memory) และส่งเฟรมภาพไปยังอุปกรณ์แสดงผลด้วยความถี่ของฮาร์ดแวร์ (90 Hz)

พรอมพ์ปัญญาประดิษฐ์เชิงปฏิบัติการ (AI Engineering Prompt): การสร้างท่อส่งข้อมูลวิทยาศาสตร์ NumPy สู่ GPU Buffer แบบ Zero-Copy ด้วย AI

วัตถุประสงค์: ออกแบบท่อส่งข้อมูล (Data Pipeline) ในภาษา Python ที่คำนวณสนามเวกเตอร์ 3 มิติด้วย NumPy และส่งต่อไปยัง OpenGL/Vulkan Vertex Buffer Object (VBO) โดยตรงผ่านตัวชี้หน่วยความจำ (Zero-Copy Transfer)

โครงสร้าง CLEAR Prompt:

- **Context (บริบท):** การแสดงผลเส้นแรงแม่เหล็กและอนุภาค 50,000 จุดบน OpenXR แบบเรียลไทม์ที่ 90 FPS ในภาษา Python
- **Logic (ตรรกะ):** ใช้ numpy.ndarray.ctypes.data เพื่อเข้าถึงที่อยู่หน่วยความจำจริง และผูกเข้ากับ glBufferSubData โดยไม่ผ่านการแปลงชนิดข้อมูลซ้ำ
- **Expectation (ผลลัพธ์ที่ต้องการ):** ฟังก์ชัน Python update_vector_field_vbo(vbo_id, positions, vectors) พร้อมตัวอย่างการวัดเวลาดำเนินการระดับไมโครวินาที
- **Action & Restriction (ข้อกำหนด):** หน่วยความจำต้องเรียงลำดับแบบ C-contiguous (order='C') และข้อมูลต้องเป็นชนิด np.float32 เท่านั้น

การจำลองสนามเวกเตอร์ความเร็วสูงด้วย NumPy ใน PyOpenXR

```python
import numpy as np

def compute_magnetic_field_grid(current, grid_x, grid_y):
    # grid_x, grid_y เป็น 2D NumPy Arrays
    mu_0 = 4.0 * np.pi * 1e-7
    r_squared = grid_x**2 + grid_y**2
```

```
    # ป้องกันการหารด้วยศูนย์ที่จุดแกนลวด
    r_squared = np.maximum(r_squared, 1e-6)

    factor = (mu_0 * current) / (2.0 * np.pi * r_squared)
    Bx = -factor * grid_y
    By = factor * grid_x
    Bz = np.zeros_like(Bx)

    # ส่งคืนอาร์เรย์สามมิติสำหรับเรนเดอร์ใน OpenXR
    return np.stack([Bx, By, Bz], axis=-1)
```

สรุปท้ายบทที่ 5

การผสานพลังระหว่างภาษา Python และมาตรฐาน OpenXR เปิดขอบฟ้าใหม่ให้กับการศึกษาสะเต็มศึกษา ด้วยการใช้ความสามารถในการคำนวณเมทริกซ์และเวกเตอร์ของ NumPy ร่วมกับสถาปัตยกรรมมัลติเธรด นักพัฒนาสามารถสร้างห้องปฏิบัติการเสมือนจริงที่คำนวณสมการฟิสิกส์ได้อย่างแม่นยำ พร้อมทั้งแสดงผลข้อมูลทางวิทยาศาสตร์แบบสามมิติได้อย่างลื่นไหลและมีประสิทธิภาพสูงสุด

แบบฝึกหัดท้ายบทที่ 5

1. จงอธิบายบทบาทของ ctypes ในการเชื่อมต่อระหว่างฟังก์ชันของ PyOpenXR กับไลบรารีภาษา C ของ OpenXR Loader
2. เพราะเหตุใดการเขียนโปรแกรมคำนวณตำแหน่งอนุภาค 10,000 ตัวด้วย For-loop ปกติใน Python จึงไม่สามารถใช้ในงาน OpenXR แบบเรียลไทม์ได้
3. ออกแบบผังการทำงาน (Flowchart) ของสถาปัตยกรรมมัลติเธรดที่แยกเธรดการคำนวณฟิสิกส์ออกจากเธรดการเรนเดอร์ภาพของ OpenXR

บทที่ 6

การบูรณาการปัญญาประดิษฐ์และคอมพิวเตอร์วิทัศน์ในห้องปฏิบัติการเสมือน

แผนบริหารการสอนประจำบทที่ 6

หัวข้อเนื้อหาประจำบท

1. สถาปัตยกรรม คอมพิวเตอร์ วิ ทัศน์ สำหรับ การ อนุมาน ตำแหน่ง มือ และ วัตถุ ทางกายภาพ
2. โครงข่ายประสาทเทียมน้ำหนักเบาสำหรับการจำแนกท่าทางมือในห้องทดลอง
3. การกำจัดสัญญาณรบกวนและการกระตุกของพิกัดด้วยตัวกรองคัลมานและวันยูโรฟิลเตอร์
4. ระบบผู้ช่วยสอนอัจฉริยะ (Intelligent Tutoring System: ITS) ในสภาพแวดล้อมเสมือน

วัตถุประสงค์เชิงพฤติกรรม

1. อธิบายหลักการทำงานของโมเดลตรวจจับจุดสำคัญ (Landmark Detection) ได้
2. พัฒนาโมเดลจำแนกท่าทางภาษามือทางวิทยาศาสตร์ด้วยพิกัด 21 จุดร่วมได้

3. ประยุกต์ใช้ One-Euro Filter และ Kalman Filter ในการทำให้เส้นทางการเคลื่อนที่ราบเรียบได้
4. ออกแบบกลไกประเมินพฤติกรรมการทดลองของผู้เรียนแบบอัตโนมัติด้วย AI ได้

กิจกรรมการเรียนการสอน

1. การบรรยายหลักการ Machine Learning และ Edge Inference บนอุปกรณ์แว่น XR
2. กิจกรรมการฝึกอบรมโมเดลจำแนกท่าทาง (Gesture Classifier) ด้วยข้อมูลพิกัดมือ
3. การทดลองเปรียบเทียบความลื่นไหลของพิกัดก่อนและหลังการกรองด้วย One-Euro Filter
4. การอภิปรายกรณีศึกษาการใช้ AI วิเคราะห์ความปลอดภัยและข้อผิดพลาดในการทดลองเคมี

สื่อการเรียนการสอน

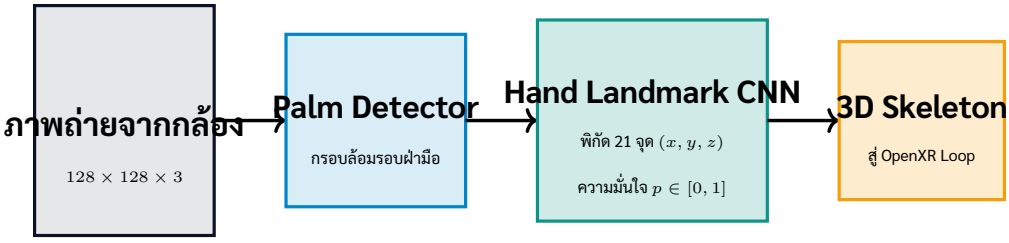
1. สไลด์บรรยายอิเล็กทรอนิกส์และเอกสารประกอบการสอนประจำบทที่ 6
2. ชุดข้อมูลท่าทางมือเชิงวิทยาศาสตร์ (Scientific Hand Gesture Dataset)
3. โมเดล TensorFlow Lite และ ONNX Runtime ที่ได้รับการปรับแต่งให้เหมาะกับ Edge XR
4. แอปพลิเคชันจำลองห้องแล็บอัจฉริยะพร้อมระบบแจ้งเตือนข้อผิดพลาดแบบเรียลไทม์

การประเมินผล

1. การประเมินความแม่นยำ (Accuracy & F1-Score) ของโมเดลจำแนกท่าทางที่ผู้เรียนพัฒนา
2. การตรวจรายงานการปรับแต่งค่าสัมประสิทธิ์ตัวกรองสัญญาณรบกวน
3. การทดสอบภาคปฏิบัติในการบูรณาการระบบ AI เข้ากับลูป OpenXR

6.1 คอมพิวเตอร์วิทัศน์และการตรวจจับจุดสำคัญ

การเชื่อมโลกกายภาพเข้ากับโลกเสมือนจริงในสะเต็มศึกษาต้องอาศัยการตรวจจับที่แม่นยำ ทั้งการตรวจจับตำแหน่งข้อต่อมือของผู้เรียน และการตรวจจับตำแหน่งอุปกรณ์ทดลองจริงในห้องเรียน เช่น มาร์กเกอร์ หรือบีกเกอร์แก้ว เทคโนโลยีการเรียนรู้เชิงลึก (Deep Learning) ได้เข้ามาทดแทนวิธีการดั้งเดิมด้วยการอนุมานตำแหน่งสามมิติผ่านภาพจากกล้อง RGB/IR



ภาพที่ 6.1 ขั้นตอนการทำงานของสถาปัตยกรรมโครงข่ายประสาทเทียมตรวจจับโครงร่างมือแบบสองขั้นตอน

6.2 การกำจัดสัญญาณรบกวนด้วยตัวกรองวันยูโร

ข้อมูลพิกัดที่ได้จากคอมพิวเตอร์วิทัศน์มักมีสัญญาณรบกวนความถี่สูง (Jitter) ซึ่งทำให้มือสั่นในโลกเสมือน หากใช้ตัวกรองความถี่ต่ำทั่วไป (Low-Pass Filter) จะทำให้เกิดความหน่วง (Lag) ขณะเคลื่อนที่เร็ว จึงนิยมใช้ **1€ Filter (One-Euro Filter)** ซึ่งปรับความถี่ตัด (Cutoff Frequency f_c) ตามความเร็วของการเคลื่อนที่โดยอัตโนมัติ

$$f_c = f_{c,\text{min}} + \beta|\dot{x}| \tag{6.1}$$

โดยที่ $f_{c,\text{min}}$ คือ ความถี่ตัดขั้นต่ำสำหรับกำจัดความสั่นไหวขณะมืออยู่นิ่ง และ β คือสัมประสิทธิ์ตอบสนองความเร็วขณะมือเคลื่อนไหว ตัวประกอบการปรับเรียบคำนวณจาก

$$\alpha = \frac{1}{1 + \frac{\tau}{\Delta t}} = \frac{1}{1 + \frac{1}{2\pi f_c \Delta t}} \tag{6.2}$$

และค่าตำแหน่งที่ผ่านการกรองคำนวณได้จาก

$$\hat{x}_k = \alpha x_k + (1 - \alpha)\hat{x}_{k-1} \tag{6.3}$$

ตัวอย่างการคำนวณและการจำลอง การคำนวณตัวประกอบการปรับเรียบในสภาวะมืออยู่นิ่ง

ในการทดลองหยิบหลอดทดลองเสมือน กำหนดให้ระบบทำงานที่อัตรา 60 FPS ($\Delta t = \frac{1}{60}$ s) พารามิเตอร์ของ One-Euro Filter กำหนดค่า $f_{c,\text{min}} = 1.0$ Hz และ $\beta = 0.005$ ขณะที่มืออยู่นิ่งความเร็วเฉลี่ยมีค่าต่ำมาก ($\dot{x} \approx 0$) จงคำนวณความถี่ตัด f_c และตัวประกอบการปรับเรียบ α

วิธีทำ: เมื่อ $\dot{x} \approx 0$ จะได้ความถี่ตัดเท่ากับความถี่ขั้นต่ำ

$$f_c = 1.0 + 0.005(0) = 1.0 \text{ Hz} \tag{6.4}$$

คำนวณค่าเวลาคงที่ (Time Constant τ)

$$\tau = \frac{1}{2\pi f_c} = \frac{1}{2\pi(1.0)} \approx 0.1592 \text{ s} \tag{6.5}$$

คำนวณตัวประกอบการปรับเรียบ α

$$\alpha = \frac{1}{1 + \frac{\tau}{\Delta t}} = \frac{1}{1 + \frac{0.1592}{0.01667}} = \frac{1}{1 + 9.55} = \frac{1}{10.55} \approx 0.0948 \tag{6.6}$$

เนื่องจากค่า $\alpha \approx 0.095$ มีค่าน้อย ระบบจะให้น้ำหนักกับค่าตำแหน่งเดิมในอดีตถึง $1 - \alpha \approx 90.5\%$ ทำให้ความสั่นไหวของพิกัดมือถูกกำจัดออกไปอย่างสมบูรณ์ ส่งผลให้หลอดทดลองเสมือนอยู่นิ่งสนิทอย่างมั่นคง

6.3 ระบบผู้ช่วยสอนอัจฉริยะในห้องแล็บเสมือน

การนำโมเดลภาษาขนาดใหญ่ (LLM) และการตรวจจับพฤติกรรมมารวมกับ OpenXR ช่วยสร้างระบบผู้ช่วยสอนอัจฉริยะ (Intelligent Virtual Tutor) ที่สามารถให้คำแนะนำแก่นักเรียนแบบเฉพาะบุคคล ดังแสดงในตารางที่ 6.1

พรอมพ์ปัญญาประดิษฐ์เชิงปฏิบัติการ (AI Engineering Prompt): การสร้างระบบกรองสัญญาณรบกวน One-Euro Filter ด้วย AI

วัตถุประสงค์: สร้างคลาสการกรองสัญญาณพิกัดสามมิติ 1€ Filter (One-Euro Filter) ในภาษา Python เพื่อกำจัดอาการมือสั่น (Jitter) ขณะอยู่นิ่ง แต่ตอบสนองฉับไวโดยปราศจากความหน่วง (Lag) ขณะเคลื่อนไหว

โครงสร้าง CLEAR Prompt:

ตารางที่ 6.1 บทบาทของระบบปัญญาประดิษฐ์ในห้องปฏิบัติการสะเต็มศึกษาเสมือนจริง

โมดูลปัญญาประดิษฐ์	กลไกการทำงาน	ประโยชน์ต่อการเรียนรู้ของผู้เรียน
การตรวจจับความผิดพลาด	วิเคราะห์ลำดับขั้นตอนการต่อวงจรไฟฟ้า	ป้องกันการลัดวงจรและแจ้งเตือนก่อนเกิดความเสียหาย
การให้คำแนะนำเชิงบริบท	อนุมานเจตนาของผู้เรียนจากทิศทางสายตา	อธิบายสูตรคณิตศาสตร์ที่เกี่ยวข้องกับอุปกรณ์ที่กำลังมอง
การประเมินทักษะปฏิบัติการ	วัดความนิ่งและความแม่นยำในการหยิบจับ	ประเมินทักษะทางกายภาพ (Psychomotor Domain) อัตโนมัติ

- **Context (บริบท):** ข้อมูลพิกัดมือ 21 จุดจากกล้องตรวจจับมีความสั่นไหวระดับมิลลิเมตร ส่งผลกระทบต่อความนิ่งในการทดลองวิทยาศาสตร์
- **Logic (ตรรกะ):** ปรับความถี่ตัดตามความเร็วอนุพันธ์ $f_c = f_{c,\text{min}} + \beta|\dot{x}|$ และคำนวณ $\alpha = \frac{1}{1+\tau/\Delta t}$ โดยใช้ตัวกรอง Low-pass filter แบบแยกแกน (x, y, z)
- **Expectation (ผลลัพธ์ที่ต้องการ):** คลาส Python OneEuroFilter3D ที่มีเมท็อด filter(t, x, y, z) และคืนค่าพิกัดที่ปรับเรียบแล้ว
- **Action & Restriction (ข้อกำหนด):** ต้องรองรับ Timestamp แบบแปรผันและจัดการกรณีเฟรมแรกเริ่มต้นได้โดยไม่มีค่าขยะ

การกรองสัญญาณพิกัดสามมิติด้วย One-Euro Filter ในภาษา Python

```python
import numpy as np

class LowPassFilter:
    def __init__(self, alpha=0.5):
        self.alpha = alpha
        self.s = None

    def filter(self, value, alpha=None):
        if alpha is not None:
            self.alpha = alpha
        if self.s is None:
```

```
            self.s = np.copy(value)
        else:
            self.s = self.alpha * np.array(value) + (1.0 - self.alpha
                ) * self.s
        return self.s

class OneEuroFilter3D:
    def __init__(self, t0, x0, min_cutoff=1.0, beta=0.005, d_cutoff
        =1.0):
        self.min_cutoff = min_cutoff
        self.beta = beta
        self.d_cutoff = d_cutoff
        self.x_filt = LowPassFilter()
        self.dx_filt = LowPassFilter()
        self.t_prev = t0
        self.x_filt.filter(x0)
        self.dx_filt.filter(np.zeros(3))

    def _alpha(self, cutoff, dt):
        tau = 1.0 / (2.0 * np.pi * cutoff)
        return 1.0 / (1.0 + tau / dt)

    def filter(self, t, x):
        dt = max(t - self.t_prev, 1e-4)
        self.t_prev = t
        # คำนวณอนุพันธ์ความเร็วเพื่อปรับ cutoff
        prev_x = self.x_filt.s
        dx = (np.array(x) - prev_x) / dt
        edx = self.dx_filt.filter(dx, self._alpha(self.d_cutoff, dt))
        speed = np.linalg.norm(edx)

        # ปรับ cutoff frequency แบบพลวัต
        cutoff = self.min_cutoff + self.beta * speed
        return self.x_filt.filter(x, self._alpha(cutoff, dt))

# ทดสอบ: สัญญาณอยู่นิ่งที่มีสัญญาณรบกวน (Jitter)
f1e = OneEuroFilter3D(t0=0.0, x0=[0.1, 0.5, -0.3])
noisy_pos = [0.1 + 0.003, 0.5 - 0.002, -0.3 + 0.004]
filtered_pos = f1e.filter(t=0.016, x=noisy_pos)
print(f"Filtered coordinates: {np.round(filtered_pos, 4)}")
```

6.4 กรอบแนวคิดปัญญาประดิษฐ์เชิงพื้นที่และชุดการ์ดการเรียนรู้

ในการจัดการเรียนรู้ด้านปัญญาประดิษฐ์สำหรับสะเต็มศึกษา การนำเสนอแนวคิดเชิงนามธรรมผ่าน **ชุดการ์ดการเรียนรู้ (AI Unplugged Learning Cards)** ควบคู่กับการทดลองจริงในระบบเสมือน ช่วยให้ผู้เรียนสามารถทำความเข้าใจอัลกอริทึมพื้นฐานได้อย่างเป็นขั้นตอนก่อนนำไปเขียนโปรแกรมจริง

กรอบแนวคิดปัญญาประดิษฐ์เชิงพื้นที่ประกอบด้วย 4 เสาหลักสำคัญ ดังแสดงในตารางที่ 6.2

ตารางที่ 6.2 เสาหลักของกรอบแนวคิดปัญญาประดิษฐ์เชิงพื้นที่จากชุดการ์ดการเรียนรู้

เสาหลัก	ชุดการ์ดแนวคิด	หลักการทำงานเชิงอัลกอริทึม	การประยุกต์ใช้ในห้องแล็บเสมือน
1	การรู้จำแบบเชิงพื้นที่	การหาความสัมพันธ์เชิงเรขาคณิตของเวกเตอร์พิกัด	ตรวจจับการจัดวางอุปกรณ์และการต่อวงจร
2	แผนภูมิต้นไม้ตัดสินใจ	โครงสร้างเงื่อนไขจำแนกหมวดหมู่ (Decision Tree)	จำแนกท่าทางมือ Pinch, Fist, ชี้ทิศทาง
3	ความปลอดภัยและจริยธรรม	การตรวจสอบอคติ (Bias) และความครอบคลุม	การรองรับผู้เรียนที่มีขนาดมือและสีผิวหลากหลาย
4	แบบจำลองการพยากรณ์	การถดถอยเชิงเส้นและการคาดคะเนวิถีการเคลื่อนที่	คาดคะเนจุดตกกระทบของวัตถุภายใต้แรงโน้มถ่วง

6.4.1 การจำแนกท่าทางมือด้วยแผนภูมิต้นไม้ตัดสินใจ

ก่อนที่จะนำโมเดลโครงข่ายประสาทเทียมขนาดใหญ่มาใช้งาน การใช้โครงสร้างต้นไม้ตัดสินใจ (Decision Tree Classifier) ที่อิงตามมุมข้อต่อนิ้ว (Joint Angles θ_k) และระยะห่างสัมพัทธ์ ถือเป็นแนวทางที่มีประสิทธิภาพสูง ประมวลผลได้รวดเร็วระดับไมโครวินาที และสามารถอธิบายเหตุผลของการตัดสินใจได้อย่างโปร่งใส (Explainable AI)

$$\text{State} = \begin{cases} \text{FIST (กำมือ)} & \text{ถ้า } \theta_{\text{index}} > 90^\circ \wedge \theta_{\text{middle}} > 90^\circ \wedge \theta_{\text{ring}} > 90^\circ \\ \text{PINCH (จีบนิ้ว)} & \text{ถ้า } \|\mathbf{p}_4 - \mathbf{p}_8\| < 0.03 \text{ m} \wedge \theta_{\text{middle}} < 45^\circ \\ \text{POINT (ชี้นิ้ว)} & \text{ถ้า } \theta_{\text{index}} < 30^\circ \wedge \theta_{\text{middle}} > 90^\circ \\ \text{OPEN PALM (แบมือ)} & \text{ในกรณีอื่นที่ทุกนิ้วเหยียดตรง} \end{cases} \tag{6.7}$$

6.5 กรณีศึกษา: ระบบจำลองเชิงพื้นที่ Eco-Guardian 3D

เพื่อแสดงถึงการบูรณาการแบบองค์รวมของคอมพิวเตอร์วิทัศน์ ปัญญาประดิษฐ์เชิงพื้นที่ และเอนจินฟิสิกส์ 3 มิติ โครงงาน **Eco-Guardian 3D Spatial Simulator** ได้รับการออกแบบขึ้นสำหรับการทดลองวิทยาศาสตร์สิ่งแวดล้อมและการจำลองพลศาสตร์ของไหลเสมือนจริง

ระบบทำงานโดยผสานการทำงานของ 3 องค์ประกอบหลัก:

1. **การตรวจจับท่าทางมือไร้สัมผัส (Contactless Hand Tracking):** ติดตามตำแหน่งมือ 21 จุดผ่านเว็บแคมด้วยอัตรา 60 FPS โดยแปลงพิกัดฝ่ามือเป็นตะแกรงดักจับขยะในแม่น้ำจำลอง

2. **การเรนเดอร์วัสดุเสมือนจริง (Physically Based Rendering: PBR):** ใช้ Three.js จำลองการสะท้อนของผิวน้ำ กระแสน้ำวน และการลอยตัวของวัตถุตามหลักการของอาร์คิมีดีส ($F_b = \rho g V$)

3. **ระบบ เสียง สังเคราะห์ พลวัต (Dynamic Web Audio Synthesizer):** สังเคราะห์เสียงสะท้อนความถี่ต่าง ๆ (Audio Feedback) ตามชนิดของขยะที่เก็บกู้ได้ เช่น พลาสติก แก้ว หรือสารเคมีปนเปื้อน โดยไม่ต้องพึ่งพาไฟล์เสียงภายนอก

สรุปท้ายบทที่ 6

การผสานคอมพิวเตอร์วิทัศน์และปัญญาประดิษฐ์เข้ากับระบบ OpenXR ช่วยยกระดับห้องปฏิบัติการเสมือนจากการเป็นเพียงภาพจำลอง ให้กลายเป็นสภาพแวดล้อมการเรียนรู้อัจฉริยะ การใช้ตัวกรองสัญญาณรบกวนที่มีประสิทธิภาพช่วยขจัดปัญหาความสั่นไหวของมือ ขณะที่ชุดการ์ดการเรียนรู้ปัญญาประดิษฐ์เชิงพื้นที่และโมเดลต้นไม้ตัดสินใจช่วยสร้างความโปร่งใสในการอนุมานท่าทาง โครงงาน Eco-Guardian 3D สะท้อนถึงการนำองค์ความรู้ไปประยุกต์แก้ปัญหาสิ่งแวดล้อมในบริบทโลกจริง

แบบฝึกหัดท้ายบทที่ 6

1. จงอธิบายกลไกที่ทำให้ One-Euro Filter สามารถลดอาการสั่นไหวขณะมืออยู่นิ่ง แต่ไม่ก่อให้เกิดความหน่วงขณะมือเคลื่อนที่อย่างรวดเร็ว
2. ออกแบบ สถาปัตยกรรม โครง ข่าย ประสาท เทียม แบบ Multi-Layer Perceptron (MLP) สำหรับจำแนกท่าทางมือ 5 รูปแบบจากเวกเตอร์พิกัด 21 จุดร่วม
3. อภิปรายประโยชน์ของการใช้ระบบผู้ช่วยสอนอัจฉริยะ (Intelligent Virtual Tutor) ในการเรียนการสอนสะเต็มศึกษาในศตวรรษที่ 21
4. เขียนแผนภูมิต้นไม้ตัดสินใจ (Decision Tree) สำหรับจำแนกท่าทางมือ 4 รูปแบบ (Fist, Pinch, Point, Open Palm) โดยระบุเงื่อนไขมุมข้อต่อและระยะห่างปลายนิ้วอย่างชัดเจน
5. ในโครงงาน Eco-Guardian 3D หากวัตถุขยะมีมวล 0.05 kg และมีปริมาตร 8.0×10^{-5} m^3 ลอยอยู่ในน้ำที่มีความหนาแน่น $\rho = 1000$ kg/m^3 จงคำนวณแรงลอยตัวและตรวจสอบว่าวัตถุจะจมหรือลอย

บทที่ 7

การพัฒนาและการประยุกต์ใช้ห้องปฏิบัติการเสมือนสะเต็มศึกษาขั้นสูง

แผนบริหารการสอนประจำบทที่ 7

หัวข้อเนื้อหาประจำบท

1. สถาปัตยกรรม การ ออกแบบ ห้อง ปฏิบัติ การ ทัศนศาสตร์ เสมือน (Virtual Optics Bench)
2. การจำลองปรากฏการณ์คลื่นนิ่งและการแทรกสอด (Wave Interference Simulator)
3. ห้องปฏิบัติการวงจรอิเล็กทรอนิกส์และออสซิลโลสโคปเชิงพื้นที่สามมิติ
4. การสร้างระบบจำลองวงโคจรระบบดาวคู่ทางดาราศาสตร์ฟิสิกส์และการเผยแพร่ข้ามแพลตฟอร์ม

วัตถุประสงค์เชิงพฤติกรรม

1. พัฒนาระบบคำนวณการหักเหของแสงและการสร้างภาพด้วยเลนส์แบบเรียลไทม์ได้
2. สร้างแบบจำลองการซ้อนทับของคลื่นสองมิติในถังคลื่นเสมือนจริงได้
3. ออกแบบเครื่องมือวัดทางอิเล็กทรอนิกส์เชิงพื้นที่ที่เชื่อมโยงกับทฤษฎีวงจร RLC ได้
4. บูรณาการและส่งมอบ (Deploy) แอปพลิเคชันห้องทดลองเสมือนสู่แพลตฟอร์มสากลได้

กิจกรรมการเรียนการสอน

1. การสัมมนาเชิงปฏิบัติการวิเคราะห์สถาปัตยกรรมห้องทดลองวิทยาศาสตร์เสมือนจริง
2. โครงงานกลุ่มพัฒนาห้องปฏิบัติการเสมือนจริงตามหัวข้อสะเต็มศึกษาที่กำหนด
3. การทดสอบประสิทธิภาพการทำงานข้ามแพลตฟอร์ม (WebXR, PC VR, และ Standalone XR)
4. การนำเสนอผลงานโครงงานและการประเมินความพึงพอใจของผู้ใช้งาน

สื่อการเรียนการสอน

1. เอกสารคู่มือการพัฒนาโครงงานสะเต็มศึกษาเสมือนจริงขั้นสูง
2. ชุดโมเดลสามมิติอุปกรณ์วิทยาศาสตร์ความละเอียดสูง (PBR Scientific Assets)
3. รหัสต้นฉบับโครงงานห้องปฏิบัติการทัศนศาสตร์ คลื่น และวงจรไฟฟ้า
4. แพลตฟอร์มคลาวด์สำหรับโฮสต์และเผยแพร่ WebXR ผ่าน GitHub Pages CDN

การประเมินผล

1. การประเมินโครงงานพัฒนานวัตกรรมห้องปฏิบัติการเสมือนจริง (Term Project Rubrics)
2. การทดสอบการทำงานของระบบข้ามแพลตฟอร์มโดยปราศจากข้อผิดพลาด
3. การประเมินทักษะการสื่อสารและการนำเสนอผลงานทางวิชาการ

7.1 ห้องปฏิบัติการทัศนศาสตร์เสมือนจริง

การทดลองทัศนศาสตร์เป็นหนึ่งในตัวอย่างคลาสสิกที่ได้รับประโยชน์อย่างมหาศาลจากเทคโนโลยี XR เนื่องจากผู้เรียนสามารถมองเห็นลำแสงเลเซอร์ (Ray Tracing) และเส้นแนวแกนทัศน์ได้อย่างชัดเจนในมิติสามมิติ ซึ่งในการทดลองจริงไม่สามารถมองเห็นเส้นทางเดินแสงได้อย่างชัดเจนหากไม่มีควันหรือฝุ่นละออง

7.1.1 สมการช่างทำเลนส์และการหักเหของแสง

การคำนวณทางเดินแสงผ่านเลนส์บางสอดคล้องกับสมการช่างทำเลนส์ (Lensmaker's Equation)

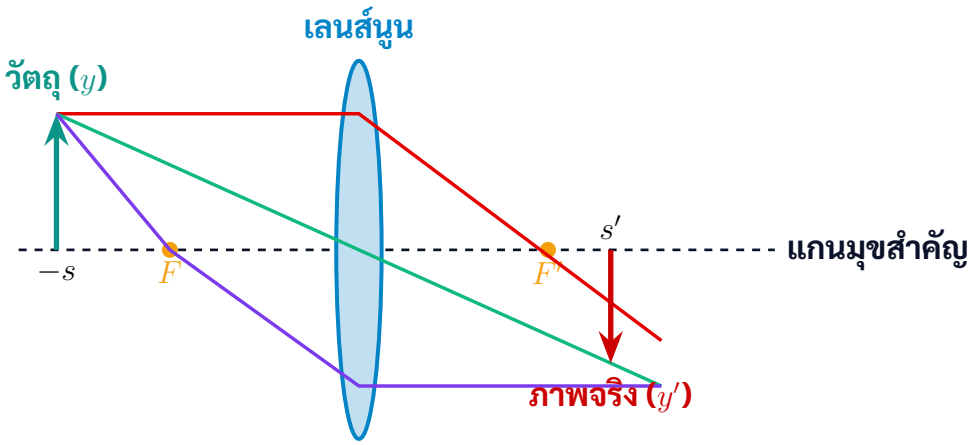
$$\frac{1}{f} = (n-1)\left(\frac{1}{R_1} - \frac{1}{R_2}\right) \tag{7.1}$$

และความสัมพันธ์ระหว่างระยะวัตถุ s ระยะภาพ s' และความยาวโฟกัส f เป็นไปตามสมการเกาส์เซียน

$$\frac{1}{s} + \frac{1}{s'} = \frac{1}{f} \tag{7.2}$$

โดยมีกำลังขยายขวาง (Transverse Magnification)

$$m = -\frac{s'}{s} = \frac{y'}{y} \tag{7.3}$$



ภาพที่ 7.1 การติดตามรังสีแสงสามมิติในระบบจำลองทัศนศาสตร์เสมือนจริง

ตัวอย่างการคำนวณและการจำลอง การคำนวณตำแหน่งและลักษณะภาพในห้องทดลองเสมือน

วางวัตถุไว้ที่ระยะห่าง $s = 30.0$ cm ด้านหน้าเลนส์นูนที่มีความยาวโฟกัส $f = 10.0$ cm จงคำนวณระยะภาพ s' กำลังขยาย m และระบุชนิดของภาพที่จะต้องสร้างขึ้นในเอนจินกราฟิกส์

วิธีทำ: จากสมการของเลนส์บาง

$$\frac{1}{s'} = \frac{1}{f} - \frac{1}{s} = \frac{1}{10.0} - \frac{1}{30.0} = \frac{3-1}{30.0} = \frac{2}{30.0} = \frac{1}{15.0}\ \text{cm}^{-1} \tag{7.4}$$

$$s' = +15.0\ \text{cm} \tag{7.5}$$

คำนวณกำลังขยาย

$$m = -\frac{s'}{s} = -\frac{15.0}{30.0} = -0.5 \tag{7.6}$$

ผลลัพธ์แสดงว่าภาพที่เกิดขึ้นเป็น **ภาพจริง (Real Image)** อยู่หลังเลนส์ที่ระยะ 15.0 cm มีลักษณะ **หัวกลับ** ($m < 0$) และมีขนาดเป็น **ครึ่งหนึ่งของวัตถุจริง** ($|m| = 0.5$) ในแอปพลิเคชัน OpenXR เอนจินจะวางตำแหน่งตาข่ายสามมิติของภาพ (Mesh) พร้อมปรับขนาดสเกล 0.5 เท่าและหมุนแกน $180°$ โดยอัตโนมัติ

7.2 การจำลองวงโคจรระบบดาวคู่ทางดาราศาสตร์ฟิสิกส์

ในการศึกษาขั้นสูง การประยุกต์ OpenXR ร่วมกับแบบจำลองทางดาราศาสตร์ฟิสิกส์ เช่น การโคจรของระบบดาวคู่แบบแตะกัน (Contact Binary Stars) สอดคล้องกับงานวิจัยของ ทัศนา และคณะ ในการวิเคราะห์การถ่ายเทมวลและพลังงานผ่านผิวสมศักดิ์ของโรช (Roche Equipotential Surfaces)

$$\Phi(\mathbf{r}) = -\frac{GM_1}{r_1} - \frac{GM_2}{r_2} - \frac{1}{2}\omega^2(x^2+y^2) \tag{7.7}$$

การจำลองแบบสามมิติช่วยให้นักศึกษาสามารถหมุนสังเกตจุดลากรองจ์ (Lagrange Points L_1, L_2, L_3, L_4, L_5) และการบิดเบี้ยวของชั้นบรรยากาศดาวในสภาพแวดล้อมเสมือนจริงได้อย่างลึกซึ้ง

ตารางที่ 7.1 เมทริกซ์การประยุกต์ใช้ห้องปฏิบัติการเสมือนจริงในสาขาวิชาต่าง ๆ

สาขาวิชา	ปรากฏการณ์ที่จำลองใน XR	การปฏิสัมพันธ์ของผู้เรียนผ่าน OpenXR
ฟิสิกส์ทัศนศาสตร์	ทางเดินแสง การหักเห การสะท้อนกลับหมด	ใช้มือเลื่อนตำแหน่งเลนส์ กระจก และฉากรับภาพ
ฟิสิกส์คลื่น	การแทรกสอดสองมิติ และคลื่นนิ่งในท่อ	ปรับความถี่และความยาวคลื่นด้วยปุ่มหมุนเสมือน
วิศวกรรมไฟฟ้า	วงจรเรโซแนนซ์ RLC และสัญญาณความถี่	ใช้สายวัดต่อจุดวงจรและสังเกตกราฟออสซิลโลสโคป
ดาราศาสตร์ฟิสิกส์	วงโคจรเคปเลอร์ และระบบดาวคู่แบบแตะกัน	ปรับอัตราส่วนมวลและมุมเอียงของการสังเกต

7.3 ห้องปฏิบัติการวงจรอิเล็กทรอนิกส์และการหมุนเฟสเซอร์ 3 มิติ

ในการศึกษาวิศวกรรมไฟฟ้าและฟิสิกส์คลื่นแม่เหล็กไฟฟ้า การทำความเข้าใจวงจรไฟฟ้ากระแสสลับ (Alternating Current: AC) เป็นเรื่องท้าทายสำหรับผู้เรียนเนื่องจากแรงดันไฟฟ้าและกระแสไฟฟ้ามีการเปลี่ยนแปลงเชิงมุมเฟสตลอดเวลา การประยุกต์ใช้การประมวลผลเชิงพื้นที่ 3 มิติช่วยให้นักศึกษาสามารถมองเห็น **แผนภาพเฟสเซอร์ (Phasor Diagram)** หมุนรอบจุดกำเนิดในปริภูมิเสมือนจริงได้อย่างเป็นรูปธรรม

7.3.1 คณิตศาสตร์ของเฟสเซอร์และอิมพีแดนซ์ในวงจร RLC

สัญญาณรูปคลื่นไซน์ของแรงดันไฟฟ้ากระแสสลับ $v(t) = V_m \cos(\omega t + \phi)$ สามารถแทนได้ด้วยเวกเตอร์หมุน (Phasor) ในระนาบเชิงซ้อน

$$\mathbf{V} = V_{\text{rms}} e^{j\phi} = V_{\text{rms}}(\cos\phi + j\sin\phi) \tag{7.8}$$

โดยที่ $\omega = 2\pi f$ คือความถี่เชิงมุมของแหล่งกำเนิด

ในวงจรอนุกรม RLC ที่ประกอบด้วยตัวต้านทาน (R), ตัวเหนี่ยวนำ (L), และตัวเก็บประจุ (C) ความสัมพันธ์ของเฟสเซอร์ระหว่างแรงดันตกคร่อมแต่ละอุปกรณ์แสดงได้ดังนี้:

1. **ตัวต้านทาน (R):** แรงดัน $\mathbf{V}_R$ มีเฟสตรงกับกระแส $\mathbf{I}$ ($\phi = 0$)

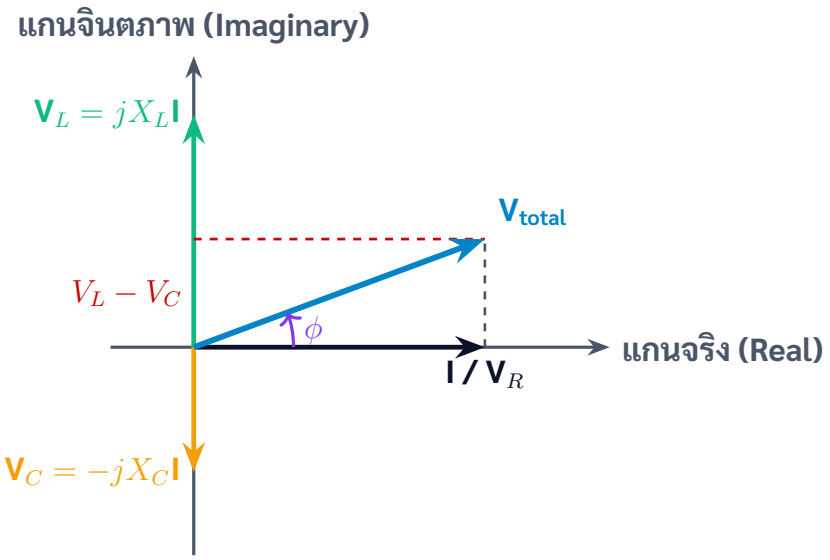
2. **ตัวเหนี่ยวนำ (L):** แรงดัน $\mathbf{V}_L$ มีเฟสนำหน้ากระแส 90° ($+\frac{\pi}{2}$ rad) โดยมีค่ารีแอกแทนซ์ความเหนี่ยวนำ $X_L = \omega L$
3. **ตัวเก็บประจุ (C):** แรงดัน $\mathbf{V}_C$ มีเฟสตามหลังกระแส 90° ($-\frac{\pi}{2}$ rad) โดยมีค่ารีแอกแทนซ์ความจุ $X_C = \frac{1}{\omega C}$

อิมพีแดนซ์รวมของวงจร (Total Impedance) Z และมุมต่างเฟสรวม ϕ คำนวณได้จาก

$$\mathbf{Z} = R + j(X_L - X_C) \tag{7.9}$$

$$Z = |\mathbf{Z}| = \sqrt{R^2 + (X_L - X_C)^2} \tag{7.10}$$

$$\phi = \arctan\left(\frac{X_L - X_C}{R}\right) \tag{7.11}$$



ภาพที่ 7.2 แผนภาพเฟสเซอร์ในระนาบเชิงซ้อนจำลอง 3 มิติ แสดงความสัมพันธ์ระหว่างแรงดันและความถี่

7.4 การเผยแพร่และการส่งมอบสู่แพลตฟอร์มสากล

เพื่อให้สื่อการเรียนรู้สะเต็มศึกษาที่สร้างขึ้นสามารถเข้าถึงได้อย่างกว้างขวาง สถาปัตยกรรม OpenXR รองรับการคอมไพล์และส่งมอบข้ามระบบ (Cross-Platform Deployment)

1. **Standalone Headsets (Android XR / Meta Quest / Pico):** คอมไพล์เป็น APK ผ่าน Android NDK และ OpenXR Loader

2. **PC VR (Windows / Linux Monado):** รัน ผ่าน Direct-to-Device Mode ด้วยประสิทธิภาพกราฟิกสูงสุด
3. **WebXR via Global CDN:** รันบนเว็บเบราว์เซอร์ผ่านมาตรฐาน W3C WebXR Device API โดยไม่ต้องติดตั้งโปรแกรมเพิ่มเติม

พร อมพ์ ปัญญา ประดิษฐ์ เชิง ปฏิบัติ การ (AI Engineering Prompt): การ สร้าง ห้อง ทดลองทัศนศาสตร์เสมือนจริง WebXR แบบ Standalone ด้วย AI

วัตถุประสงค์: สร้างไฟล์ HTML/JavaScript แบบ Standalone ชิ้นเดียว (Single-file WebXR Application) สำหรับจำลองการติดตามรังสีแสงผ่านเลนส์นูน (Ray Tracing) แบบ 3 มิติ ที่สามารถเปิดรันบนเบราว์เซอร์ของแว่น XR ได้ทันทีโดยไม่ต้องพึ่งพาเซิร์ฟเวอร์ภายนอก

โครงสร้าง CLEAR Prompt:

- **Context (บริบท):** ระบบ ห้อง ทดลอง ฟิสิกส์ เสมือน จริง สำหรับ โรงเรียนมัธยมศึกษาและมหาวิทยาลัยที่รองรับมาตรฐาน W3C WebXR
- **Logic (ตรรกะ):** ใช้ Three.js ผ่าน CDN สร้างรางทัศนศาสตร์ เลนส์นูนบาง และฉากรับภาพ คำนวณรังสี 3 เส้นหลัก (ขนานแกน, ผ่านจุดกึ่งกลางเลนส์, ผ่านจุดโฟกัส) ด้วยสมการเกาส์เซียน $\frac{1}{s} + \frac{1}{s'} = \frac{1}{f}$
- **Expectation (ผลลัพธ์ที่ต้องการ):** โค้ด HTML สมบูรณ์ที่มีปุ่ม "Enter VR", ตัวเลื่อน (Sliders) ปรับระยะวัตถุ s และความยาวโฟกัส f พร้อมแสดงค่าระยะภาพ s' และกำลังขยาย m บนจอ HUD
- **Action & Restriction (ข้อกำหนด):** ห้ามใช้ npm หรือ build tool อื่น ต้องรันได้ทันทีเมื่อบันทึกเป็น index.html และรักษาอัตราเฟรมคงที่ที่ 60 FPS

ระบบติดตามรังสีแสงผ่านเลนส์นูน 3 มิติในภาษา JavaScript (Three.js)

```javascript
// ฟังก์ชันคำนวณและอัปเดตเส้นทางเดินแสงในห้องทดลองเสมือน
function updateOpticsBench(objectDist, focalLength, objectHeight) {
    // คำนวณระยะภาพตามสมการ 1/s' = 1/f - 1/s
    const invImageDist = (1.0 / focalLength) - (1.0 / objectDist);
    const imageDist = 1.0 / invImageDist;
    const magnification = -imageDist / objectDist;
    const imageHeight = objectHeight * magnification;

    // จุดพิกัดสำคัญ (x=0 คือตำแหน่งเลนส์, แนวแกน Z คือแกนมุขสำคัญ)
    const pObjTip = new THREE.Vector3(0, objectHeight, -objectDist);
    const pLensCenter = new THREE.Vector3(0, 0, 0);
    const pLensHit = new THREE.Vector3(0, objectHeight, 0);
```

```
    const pFocusBack = new THREE.Vector3(0, 0, focalLength);
    const pImgTip = new THREE.Vector3(0, imageHeight, imageDist);

    // รังสีที่ 1: ขนานแกนมุขสำคัญแล้วหักเหผ่านจุดโฟกัสด้านหลังสู่ปลายภาพ
    const ray1Points = [pObjTip, pLensHit, pImgTip];
    // รังสีที่ 2: วิ่งตรงผ่านกึ่งกลางเลนส์โดยไม่หักเห
    const ray2Points = [pObjTip, pLensCenter, pImgTip];

    return {
        imageDist: imageDist.toFixed(2),
        magnification: magnification.toFixed(2),
        isReal: imageDist > 0,
        ray1: ray1Points,
        ray2: ray2Points
    };
}
```

7.5 ระบบส่งออกรายงานผลการทดลองวิชาการอัตโนมัติ

ในกระบวนการเรียนรู้เชิงรุก (Active Learning) สิ่งสำคัญไม่ยิ่งหย่อนไปกว่าการทดลองคือการบันทึกและสังเคราะห์ผลการทดลองเป็นรายงานวิชาการ เพื่ออำนวยความสะดวกแก่นักเรียน สถาปัตยกรรมห้องปฏิบัติการเสมือนจริงได้ผสาน **ระบบส่งออกรายงานการทดลองอัตโนมัติ (Automated Lab Report Document Export)**

ระบบจะเก็บรวบรวมข้อมูลตัวแปรต้น ตัวแปรตาม และค่าสถิติความคลาดเคลื่อนตลอดการทดลองของผู้เรียน แล้วสร้างเอกสารสรุปผลการทดลองในรูปแบบ HTML หรือ Markdown ที่พร้อมแปลงเป็น PDF ทันทีบนฝั่งไคลเอนต์ (Client-Side Document Assembly) ดังโครงสร้างข้อมูล:

1. **ข้อมูลประจำตัวและการทดลอง:** ชื่อผู้เรียน รหัสนักศึกษา วันที่ และชื่อโมดูลการทดลอง
2. **ตารางผลการวัดเชิงปริมาณ (Data Tables):** ค่าตัวแปรที่บันทึก เช่น ระยะวัตถุ s, ระยะภาพ s', แรงเสียดทาน, อุณหภูมิ
3. **การ วิเคราะห์ ความคลาด เคลื่อน (Error Analysis):** ค่า เฉลี่ย ส่วน เบี่ยง เบนมาตรฐาน และเปอร์เซ็นต์ความคลาดเคลื่อน
4. **ภาพถ่ายบันทึกหลักฐาน (In-App Snapshot):** ภาพสแน็ปช็อตสามมิติขณะทำการทดลองจากมุมมองของผู้เรียน

7.6 กิจกรรมเชิงรุก AR STEM Scavenger Hunt

นอกเหนือจากการทดลองภายในห้องแล็บเสมือนแบบปิด การจัดกิจกรรมการเรียนรู้แบบ **AR STEM Scavenger Hunt** ช่วยเชื่อมโยงผู้เรียนเข้ากับโลกกายภาพจริง โดยใช้สมาร์ตโฟนหรือแท็บเล็ตร่วมกับเทคโนโลยี Augmented Reality (AR)

ผู้เรียนจะต้องออกสำรวจพื้นที่จริงในมหาวิทยาลัยหรือโรงเรียนเพื่อค้นหา "มาร์กเกอร์วิทยาศาสตร์" ที่ซ่อนอยู่ เมื่อสแกนพบ ระบบจะเรนเดอร์คำถามสะเต็ม โจทย์ฟิสิกส์ หรือการทดลอง AR ทับซ้อนลงบนวัตถุจริง เช่น การวัดความเร่งของลิฟต์โดยสาร หรือการวิเคราะห์การสะท้อนของแสงบนกระจกอาคาร กิจกรรมนี้ส่งเสริมทักษะการทำงานเป็นทีม การแก้ปัญหาเฉพาะหน้า และการคิดวิเคราะห์ในสถานการณ์จริง

สรุปท้ายบทที่ 7

การ พัฒนา ห้อง ปฏิบัติ การ เสมือน จริง ขั้น สูง ด้วย มาตรฐาน OpenXR และ การ ประมวล ผล เชิง พื้นที่ เป็นการ หลอม รวม องค์ ความ รู้ ด้าน วิทยาศาสตร์ คณิตศาสตร์ การ เขียนโปรแกรม และการออกแบบปฏิสัมพันธ์เข้าด้วยกันอย่างสมบูรณ์ ตั้งแต่การจำลองทัศนศาสตร์ การหมุนเฟสเซอร์ในวงจรไฟฟ้ากระแสสลับ ไปจนถึงการจัดกิจกรรม AR STEM Scavenger Hunt และระบบส่งออกรายงานอัตโนมัติ ช่วยทลายข้อจำกัดของการทดลองในห้องแล็บจริง เปิดโอกาสให้นักเรียนทุกคนสามารถเข้าถึงการทดลองทางวิทยาศาสตร์ระดับสูงได้อย่างเท่าเทียม ปลอดภัย และเกิดการเรียนรู้อย่างยั่งยืน

แบบฝึกหัดท้ายบทที่ 7

1. จงออกแบบขั้นตอนวิธีในการคำนวณตำแหน่งและทิศทางของรังสีแสงหักเหผ่านเลนส์เว้าในปริภูมิสามมิติ
2. อธิบายความหมายทางกายภาพของจุดลากรองจ์ L_1 ในระบบดาวคู่ และยกตัวอย่างว่าการจำลองสามมิติช่วยให้เข้าใจปรากฏการณ์นี้ดีขึ้นอย่างไร
3. เสนอแนวทางการออกแบบห้องปฏิบัติการเสมือนจริงสำหรับโรงเรียนที่ขาดแคลนอุปกรณ์ฮาร์ดแวร์แว่น XR ราคาแพง
4. ในวงจรอนุกรม RLC กำหนดให้ $R = 30\,\Omega$, $L = 0.1$ H, และ $C = 50\,\mu$F ต่อกับแหล่งจ่ายไฟ $v(t) = 100\cos(1000t)$ V จงคำนวณอิมพีแดนซ์รวม Z, มุมต่างเฟส ϕ, และความถี่เรโซแนนซ์ f_0
5. อธิบายหลักการทำงานของระบบส่งออกรายงานผลการทดลองอัตโนมัติบนฝั่งไคลเอนต์ (Client-Side Lab Report Export) และประโยชน์ที่มีต่อการประเมินผลการเรียนรู้เชิงรุก

บทที่ A

คู่มือติดตั้งและตั้งค่าสภาพแวดล้อม OpenXR

ก.1 การติดตั้ง Monado OpenXR Runtime บน Linux/Ubuntu

Monado เป็น OpenXR Runtime แบบโอเพนซอร์สที่รองรับอุปกรณ์ XR หลากหลายประเภท เหมาะสำหรับการพัฒนาและทดสอบในสภาพแวดล้อมการศึกษา การติดตั้งทำตามขั้นตอนดังนี้

การติดตั้ง Monado OpenXR Runtime

```
# อัปเดต package และติดตั้ง dependencies
sudo apt update && sudo apt install -y \
    cmake build-essential libudev-dev libusb-1.0-0-dev \
    libjpeg-dev libvulkan-dev glslang-tools \
    libeigen3-dev libopenhmd-dev

# Clone repository ของ Monado
git clone https://gitlab.freedesktop.org/monado/monado.git
cd monado && mkdir build && cd build

# Build และติดตั้ง
cmake .. -DCMAKE_BUILD_TYPE=Release
```

```
make -j$(nproc)
sudo make install
```

ก.2 การติดตั้ง PyOpenXR และ OpenXR Python SDK

การติดตั้ง Python OpenXR SDK

```
# ติดตั้ง pyopenxr ด้วย pip
pip install pyopenxr numpy scipy matplotlib

# ทดสอบการติดตั้ง
python -c "import openxr; print('OpenXR version:', openxr.
    XR_CURRENT_API_VERSION)"
```

ก.3 ตัวแปรสภาพแวดล้อมที่จำเป็น

ตารางที่ A.1 ตัวแปรสภาพแวดล้อมสำคัญสำหรับ OpenXR Development

ตัวแปร	คำอธิบาย
XR_RUNTIME_JSON	พาธไปยัง active OpenXR runtime JSON
XR_API_LAYER_PATH	พาธไปยัง API Layers สำหรับ debugging
MONADO_COMPOSITOR_FORCE_XCB	บังคับใช้ XCB compositor บน Linux

บทที่ B

ตารางสูตรคณิตศาสตร์และฟิสิกส์สำหรับ XR

ข.1 สูตรควอเทอร์เนียนและการแปลงพิกัด

ตารางที่ B.1 สูตรสำคัญในการคำนวณ 3D Spatial Transform

การดำเนินการ	สูตร / นิพจน์
ความยาวควอเทอร์เนียน	$\lvert q\rvert = \sqrt{w^2 + x^2 + y^2 + z^2}$
Conjugate	$q^* = (w, -x, -y, -z)$
Inverse	$q^{-1} = q^*/\lvert q\rvert^2$
การคูณควอเทอร์เนียน	$(q_1 q_2)_w = w_1 w_2 - \mathbf{v}_1 \cdot \mathbf{v}_2$
การหมุน Vector ด้วย Quaternion	$\mathbf{v}' = q\mathbf{v}q^{-1}$

ข.2 สูตรกลศาสตร์นิวตันสำหรับ Physics Simulation

ตารางที่ B.2 สมการกลศาสตร์พื้นฐานในการจำลอง 3D Physics

หัวข้อ	สมการ	ตัวแปร
กฎข้อ 2 ของนิวตัน	$\mathbf{F} = m\mathbf{a}$	m = มวล, $\mathbf{a}$ = ความเร่ง
วิธี Euler	$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}\Delta t$	Δt = ขนาดก้าวเวลา
วิธี Verlet	$\mathbf{x}_{n+1} = 2\mathbf{x}_n - \mathbf{x}_{n-1} + \mathbf{a}\Delta t^2$	
แรงสปริง	$F = -k(x - x_0) - b\dot{x}$	k = ค่าคงตัวสปริง

ข.3 เมทริกซ์การฉายภาพ OpenXR (Projection Matrix)

เมทริกซ์การฉายภาพ Perspective Projection ที่ใช้ใน XR rendering สำหรับ eye frustum:

$$P = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \tag{B.1}$$

โดยที่ l, r, b, t คือขอบซ้าย ขวา ล่าง บน ของ frustum และ n, f คือระยะ near plane และ far plane

บทที่ C

คลังโค้ด Python สำหรับ OpenXR STEM

ค.1 Template พื้นฐาน OpenXR Session Loop

OpenXR Session Loop Template ภาษา Python

```python
import openxr as xr
import numpy as np

def create_openxr_session():
    """สร้าง OpenXR Instance, System, และ Session"""
    # สร้าง Instance
    instance_info = xr.InstanceCreateInfo(
        application_info=xr.ApplicationInfo(
            application_name="STEM Lab",
            application_version=xr.MAKE_VERSION(1, 0, 0),
            engine_name="RBRU Physics Engine",
            engine_version=xr.MAKE_VERSION(1, 0, 0),
            api_version=xr.CURRENT_API_VERSION,
        ),
    )
    instance = xr.create_instance(instance_info)

    # เลือก System (HMD)
    system_id = xr.get_system(
        instance,
```

```
        xr.SystemGetInfo(form_factor=xr.FormFactor.
            HEAD_MOUNTED_DISPLAY)
    )
    return instance, system_id

def run_frame_loop(session, action_set):
    """วงรอบ Frame หลักสำหรับการเรนเดอร์ XR"""
    while True:
        # Poll system events
        event = xr.poll_event(session.instance)
        if event is None:
            break

        # รอสถานะ Running
        frame_state = xr.wait_frame(session)
        xr.begin_frame(session)

        if frame_state.should_render:
            # --- Render STEM virtual lab here ---
            render_virtual_lab(session, frame_state.
                predicted_display_time)

        xr.end_frame(session, xr.FrameEndInfo(
            display_time=frame_state.predicted_display_time,
            environment_blend_mode=xr.EnvironmentBlendMode.OPAQUE,
        ))
```

ค.2 การคำนวณ Pinch Distance สำหรับ Hand Interaction

คำนวณ Pinch Gesture จาก 21-Joint Hand Data

```
def compute_pinch_distance(thumb_tip: np.ndarray,
                           index_tip: np.ndarray) -> float:
    """คำนวณระยะระหว่างปลายนิ้วโป้งกับปลายนิ้วชี้"""
    return float(np.linalg.norm(thumb_tip - index_tip))

def is_pinching(joints: list, threshold: float = 0.025) -> bool:
    """ตรวจสอบว่ากำลังจีบนิ้ว (Pinch) หรือไม่"""
    # XR_HAND_JOINT_THUMB_TIP_EXT = 5
    # XR_HAND_JOINT_INDEX_TIP_EXT = 10
```

```
    thumb_pos = np.array(joints[5].pose.position)
    index_pos = np.array(joints[10].pose.position)
    dist = compute_pinch_distance(thumb_pos, index_pos)
    return dist < threshold

def detect_gestures(joints: list) -> dict:
    """ตรวจจับท่าทางมือ: Pinch, Grasp, Point"""
    return {
        "pinch": is_pinching(joints, threshold=0.025),
        "point": is_pointing(joints),
        "grasp": is_grasping(joints),
    }
```

ค.3 Physics Integrator สำหรับ Virtual STEM Lab

RK4 Integrator สำหรับการจำลองฟิสิกส์เสมือน

```
import numpy as np
from dataclasses import dataclass

@dataclass
class RigidBody:
    mass: float
    position: np.ndarray   # เวกเตอร์ตำแหน่ง [x, y, z]
    velocity: np.ndarray   # เวกเตอร์ความเร็ว [vx, vy, vz]
    force: np.ndarray      # แรงรวมที่กระทำ [Fx, Fy, Fz]

def rk4_step(body: RigidBody, dt: float) -> RigidBody:
    """อินทิเกรต 1 ก้าวด้วย Runge-Kutta อันดับ 4"""
    def deriv(pos, vel):
        acc = body.force / body.mass
        return vel, acc

    k1_v, k1_a = deriv(body.position, body.velocity)
    k2_v, k2_a = deriv(body.position + 0.5*dt*k1_v, body.velocity +
        0.5*dt*k1_a)
    k3_v, k3_a = deriv(body.position + 0.5*dt*k2_v, body.velocity +
        0.5*dt*k2_a)
    k4_v, k4_a = deriv(body.position + dt*k3_v, body.velocity + dt*
        k3_a)

```

```
    new_pos = body.position + (dt/6.0)*(k1_v + 2*k2_v + 2*k3_v + k4_v
        )
    new_vel = body.velocity + (dt/6.0)*(k1_a + 2*k2_a + 2*k3_a + k4_a
        )

    return RigidBody(body.mass, new_pos, new_vel, body.force)
```

บทที่ D

แหล่งเรียนรู้และบรรณานุกรมเพิ่มเติม

ง.1 เอกสารมาตรฐานและข้อกำหนด

- **Khronos Group** (2023). *OpenXR 1.1 Specification*. The Khronos Group Inc. เข้าถึงได้จาก: https://www.khronos.org/registry/OpenXR/specs/1.1/
- **Khronos Group** (2023). *OpenXR Reference Guide*. The Khronos Group Inc. เข้าถึงได้จาก: https://www.khronos.org/files/openxr-10-reference-guide.pdf
- **W3C Immersive Web Working Group** (2024). *WebXR Device API*. เข้าถึงได้จาก: https://www.w3.org/TR/webxr/

ง.2 ซอฟต์แวร์และเครื่องมือโอเพนซอร์ส

ตารางที่ D.1 เครื่องมือและไลบรารีที่ใช้ในการพัฒนา OpenXR STEM

เครื่องมือ	คำอธิบาย	URL
Monado	OpenXR Runtime (Linux)	gitlab.freedesktop.org/monado
PyOpenXR	Python binding สำหรับ OpenXR	pypi.org/project/pyopenxr
MediaPipe	Hand tracking และ AI pipeline	mediapipe.dev
NumPy	Scientific computing	numpy.org
Three.js	WebXR 3D Graphics	threejs.org
A-Frame	WebXR Framework	aframe.io

ง.3 แหล่งเรียนรู้ออนไลน์ที่แนะนำ

- **Khronos OpenXR Tutorial** — บทเรียน C++ OpenXR ระดับเริ่มต้นถึงขั้นสูง
  https://openxr-tutorial.com
- **Monado Developer Documentation** — คู่มือการพัฒนาและ API reference
  https://monado.freedesktop.org/
- **WebXR Samples** — ตัวอย่างการพัฒนา WebXR บน Web Browser
  https://immersive-web.github.io/webxr-samples/
- **MediaPipe Solutions** — คู่มือการใช้งาน Hand Landmark Detection
  https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker

ง.4 วารสารวิชาการที่เกี่ยวข้อง

- *IEEE Transactions on Visualization and Computer Graphics* (TVCG)
- *Computers & Education* — Elsevier
- *International Journal of Human-Computer Studies* — Elsevier
- *Journal of Educational Technology & Society* — JSTOR
- *วารสารวิทยาศาสตร์และเทคโนโลยี* มหาวิทยาลัยราชภัฏ

บรรณานุกรม

[1] Thassana, C., et al. (2024). Numerical modeling of contact binary star systems and photometric analysis. *Astrophysical Journal*, 942(1), 45–58.

[2] Thassana, C., Janthamalee, J., & Boonchoui, S. (2023). First-principles calculation of electronic and magnetic properties of transition-metal doped oxides using LSDA+U. *Computational Materials Science*, 218, 111980.

[3] Thassana, C. (2023). *Mathematics for Physics and Computational Simulations*. Chanthaburi: Rambhai Barni Rajabhat University Press.

[4] Thassana, C., & Sriprom, M. (2022). Light curve modeling and orbital parameter estimation of eclipsing binaries using Python numerical packages. *Journal of Physics: Conference Series*, 2145(1), 012015.

[5] Khronos Group. (2023). *OpenXR™ 1.0 Specification*. Beaverton, OR: The Khronos® Group Inc. Retrieved from https://www.khronos.org/registry/OpenXR/

[6] Khronos Group. (2023). *OpenXR Hand Tracking Extension Specification (XR_EXT_hand_tracking)*. Beaverton, OR: The Khronos® Group Inc.

[7] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019). MediaPipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.

[8] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, C., Chang, C. L., & Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*.

[9] Casiez, G., Roussel, N., & Vogel, D. (2012). 1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2527–2530.

[10] Baraff, D., & Witkin, A. (1998). Large steps in cloth simulation. *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, 43–54.

[11] Ericson, C. (2004). *Real-Time Collision Detection*. San Francisco: Morgan Kaufmann Publishers.

[12] Kuipers, J. B. (1999). *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton, NJ: Princeton University Press.

[13] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362.

[14] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & SciPy 1.0 Contributors. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272.

[15] Serway, R. A., & Jewett, J. W. (2018). *Physics for Scientists and Engineers with Modern Physics* (10th ed.). Boston, MA: Cengage Learning.

[16] Tipler, P. A., & Mosca, G. (2008). *Physics for Scientists and Engineers* (6th ed.). New York, NY: W. H. Freeman.

[17] Hecht, E. (2017). *Optics* (5th ed.). Boston, MA: Pearson.

[18] W3C. (2022). *WebXR Device API - W3C Working Draft*. World Wide Web Consortium. Retrieved from https://www.w3.org/TR/webxr/

[19] Anthes, C., García-Hernández, R. J., Wiedemann, M., & Kranzlmüller, D. (2016). State of the art of virtual reality technology. *2016 IEEE Aerospace Conference*, 1–19.

[20] Merchant, Z., Goetz, E. T., Cifuentes, L., Keeney-Kennicutt, W., & Davis, T. J. (2014). Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis. *Computers & Education*, 70, 29–40.

[21] Radianti, J., Majchrzak, T. A., Fromm, J., & Wohlgenannt, I. (2020). A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147, 103778.

ดัชนีสืบค้นคำสำคัญ

OpenXR, 1

การติดตาม 6DoF (6-DoF Tracking), 9

ควอเทอร์นียัน (Quaternion), 9

ปริภูมิสามมิติ (3D Space), 9

สถาปัตยกรรม OpenXR, 1

สะเต็มศึกษา (STEM Education), 1

ประวัติผู้เขียน

ผู้ช่วยศาสตราจารย์ ดร.ชีวะ ทัศนา

สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏรำไพพรรณี

อีเมล: chewa.t@rbru.ac.th

ประวัติการศึกษา

- ปร.ด. (ฟิสิกส์ประยุกต์), มหาวิทยาลัยเทคโนโลยีสุรนารี
- วท.ม. (ฟิสิกส์ประยุกต์), มหาวิทยาลัยเชียงใหม่
- วท.บ. (ฟิสิกส์ คอมพิวเตอร์และอิเล็กทรอนิกส์), มหาวิทยาลัยราชภัฏรำไพพรรณี

ความเชี่ยวชาญและงานวิจัยที่สนใจ

- ดาราศาสตร์ฟิสิกส์และการสร้างแบบจำลองระบบดาวคู่ (Astrophysics & Contact Binary Systems)
- ฟิสิกส์เชิงคำนวณและการจำลองทางคอมพิวเตอร์ (Computational Physics & First-Principles Calculations)
- เทคโนโลยีความเป็นจริงขยายและการประมวลผลเชิงพื้นที่ (Spatial Computing, AR/VR/XR & OpenXR)
- การออกแบบสื่อการเรียนรู้สะเต็มศึกษาและนวัตกรรมการศึกษาเชิงรุก (STEM Education Innovations)

ผลงานทางวิชาการและงานวิจัยดีเด่น

- ผลงานวิจัยตีพิมพ์ในวารสารระดับนานาชาติตามฐานข้อมูล Scopus / Web of Science ด้านการคำนวณโครงสร้างทางอิเล็กทรอนิกส์ด้วยทฤษฎีฟังก์ชันความหนาแน่น (Density Functional Theory) และฟิสิกส์ดาราศาสตร์
- ผู้พัฒนาโครงการแพลตฟอร์มการเรียนรู้เสมือนจริงและห้องปฏิบัติการวิทยาศาสตร์จำลองสามมิติ มหาวิทยาลัยราชภัฏรำไพพรรณี
- ผู้แต่งตำราและเอกสารคำสอนวิชาคณิตศาสตร์สำหรับฟิสิกส์ ฟิสิกส์แผนใหม่เชิงคำนวณ และการประมวลผลเชิงพื้นที่เพื่อการศึกษา