

# ชื่อโครงการ (Title)

- ภาษาไทย: “การสร้างรอยแตกบนพื้นผิวสามมิติแบบเป็นธรรมชาติด้วยวิธีเชิงกระบวนการ”
- English: “Procedural Generation of Natural Cracks on 3D Surfaces”

## บทคัดย่อ (Abstract)

โครงการนี้มุ่งพัฒนาวิธีการสร้าง “รอยแตก (cracks)” บนพื้นผิวสามมิติให้ดูเป็นธรรมชาติและมีความสุ่มอย่างสมเหตุสมผล เรานำแนวคิดเชิงกระบวนการ (procedural) ที่รวม Voronoi/Cellular noise, Poisson-disk sampling, และ ตัวแบบการเติบโตของรอยแตกตามเส้นแร้งเค้นโดยประมาณ (stress-biased growth) มาสร้างเครือข่ายรอยแตก จากนั้นแมปรอยแตกสู่พื้นผิวด้วยพิกัดพิเศษ/Geodesic distance บันเมซ พร้อมแสดงผลด้วยเทคนิค瓦斯ด (normal/parallax mapping) เพื่อให้เกิดความลึกและเงาที่สมจริง เกณฑ์ประเมินประกอบด้วยความเร็ว (FPS), เวลาอัปเดตรอยแตก, และความสมจริงเชิงโครงสร้าง ของกราฟรอยแตกเทียบกับการกระจายตัวตามธรรมชาติ คาดหวังผลเป็นเดโมแบบอินเตอร์แอคทีฟและชอร์สโค้ดเปิดเผยบน GitHub ภายใน 7 สัปดาห์

## ความสำคัญและที่มาของปัญหา (Introduction & Background)

พื้นผิวที่แตกร้าวเป็นองค์ประกอบสำคัญของความสมจริงในงานกราฟิกส์ ตั้งแต่พื้นคอนกรีตไปจนถึงวัสดุเซรามิก โจทย์คือสร้างรอยแตกที่ “สุ่มแต่สมเหตุสมผล” ไม่เป็นลายซ้ำ และสะท้อนพฤติกรรมธรรมชาติ โครงการนี้ผสานแนวคิดเชิงกระบวนการ (Voronoi/Cellular noise, Poisson-disk sampling) กับแบบจำลองการเติบโตอิงแร้งเค้นโดยประมาณ และจายรอยแตกสู่พื้นผิวของวัตถุ

## งานวิจัย/งานที่เกี่ยวข้อง (Related Work)

- Cellular/Worley/Voronoi และ Perlin noise** ฐานสร้างขوبนเขต/ความหยาบของลายแตก
- แบบจำลองการแตกในธรรมชาติ (เช่น รูปแบบในโคลแหนง/สีแตกลายงา): ให้สมบัติทางสติ๊ติเช่น การกระจายความเยาว์รอยแตก/มุมสาขา
- Geodesic distance บันเมซ และ Poisson-disk sampling**: ช่วยกำหนดจุดกำเนิดและการเติบโตของรอยแตกอย่างสม่ำเสมอแต่ไม่เป็นลายซ้ำ
- Parallax/Normal mapping**: ยกระดับการรับรู้ความลึกของร่องแตกโดยไม่ต้องเพิ่มโพลิกอนมากเกินไป

# คำถามการวิจัย (Research Questions)

- วิธีที่เริ่มจากจุดกระจายหลาๆ ๆ จุด และให้รอยแตกค่อยๆๆ แผ่ขยายตามทิศที่น่าจะเกิดขึ้นจริง จะทำให้ลวดลายดูเป็นธรรมชาติกว่าวิธีแบ่งพื้นที่แบบทั่วไปหรือไม่?
- การวางแผนรอยแตกให้ใกล้ไปตามรูปทรงของวัตถุ ช่วยลดการบิดเบี้ยวของลายเมื่อเทียบกับการกระจายจากระนาบธรรมชาติได้มากน้อยเพียงใด?
- พื้นผิวนี้มีตัวอย่างใดที่เหมาะสมกับงานวิจัยนี้มากที่สุด

# วิธีการดำเนินการ (Proposed Methodology)

## เครื่องมือและเทคโนโลยี

- ภาษา/ไลบรารีหลัก:** C++, OpenGL, GLFW (หน้าต่าง/อินพุต), GLAD (โหลดฟังก์ชัน), GLM (เวกเตอร์/เมตริกซ์), stb\_image / tinyobjloader (โหลดไฟล์โมเดล)
- Shader:** GLSL (vertex/fragment; optional tessellation)

## ขั้นตอนอัลกอริทึมโดยสรุป

- Seed & Sampling:** สุมเมล็ดด้วย Poisson-disk บนโดเมน UV หรือบนเมฆ (ด้วยการประมาณ geodesic) เพื่อหลีกเลี่ยงการกระจัดตัว
- Voronoi Skeleton + Jitter:** สร้างเส้นขอบ Voronoi เป็น “โครงรอยแตกหลัก” และใส่ jitter จาก Perlin/Worley noise เพื่อหลีกเลี่ยงเส้นตรงเกินไป
- Stress-Biased Growth:** คำนวณฟิล์ต “แรงเห็นโดยประมาณ” จากความโค้งพื้นผิว, ระยะจากจุด ระยะห่าง, หรือทิศทางแรง ใช้เป็นหนักให้รอยแตกเติบโต/แตกกึ่งในทิศที่สมเหตุสมผล
- Crack Refinement:** ลบกิ่งต้น/กิ่งสันผิดธรรมชาติ, ทำ smoothing นุ่มนวลแล้วก็ลากเส้นใหม่ แล้วสร้าง crack width profile ที่ควบคุมโดยรูปแบบของจุดที่มีการเปลี่ยนแปลง
- Mapping สู่พื้นผิว:** เขียน crack distance map ลงเท็กซ์เจอร์ (FBO) และคำนวณ normal perturbation

## ชุดทดสอบ (Datasets / Scenes)

- รูปทรงมาตรฐาน: sphere, plane, cube, Suzanne/teapot
- ภาพอ้างอิงรอยแตกจริง สำหรับการเปรียบเทียบเชิงคุณภาพ

## เกณฑ์ประเมินผล (Evaluation)

- ประสิทธิภาพ: FPS, เวลาอัปเดตกราฟรอยแทก/สร้างแผนที่ระยะทาง, หน่วยความจำ
- ความสมจริงเชิงโครงสร้าง: เปรียบเทียบกับรูปแบบธรรมชาติ
- การประเมินโดยผู้ใช้: ให้ผู้สังเกตให้คะแนนความสมจริง/ความหลอกหลอน

## แผนการดำเนินงาน (Timeline: 7 สัปดาห์)

สัปดาห์ 1: ตั้งค่าสภาพแวดล้อม C++/OpenGL (GLFW, GLAD, GLM), โหลดเมช/เทกซ์เจอร์, สร้าง renderer พื้นฐาน

สัปดาห์ 2: โมดูล sampling (Poisson-disk), สร้าง Voronoi บนโอดเมน UV/เมช, pipeline FBO สำหรับ crack map

สัปดาห์ 3 (Milestone 1): ได้ภาพรอยแทกพื้นฐานจาก Voronoi+jitter ที่แสดงบนพื้นผิว (normal mapping)

สัปดาห์ 4: เพิ่มฟิล์ตแรงเห็นโดยประมาณ (curvature/impact-driven) และอัลกอริทึม growth/branching; ทำ smoothing/cleanup

สัปดาห์ 5: แสดงผลแบบ normal และ parallax mapping; โปรดไฟล์ประสิทธิภาพ

สัปดาห์ 6 (Milestone 2): รวมทุกโมดูลทำงานร่วมกัน เสถียร; สุม seeds หลายรูปแบบและบันทึกผล/สถิติ

สัปดาห์ 7: ทำเดโมวิดีโอ, เขียนรายงานสรุปผล/วิเคราะห์, จัดเตรียมสไลด์นำเสนอ

## ผลที่คาดว่าจะได้รับ (Expected Outcomes)

- โปรแกรมเดโมแบบอินเตอร์แอคทีฟ (Windows/Linux) และชอร์สโค้ดบน GitHub
- ชุดพรีเซ็ตพารามิเตอร์ สำหรับสร้างรอยแทกหลากหลายสไตร์
- รายงานผลการทดลอง: ประสิทธิภาพ, ตัวอย่างภาพก่อน/หลัง, และการวิเคราะห์สถิติของรอยแทก

# បរទនាណកំ (References)

1. Akenine-Möller, T., Haines, E., Hoffman, N., et al. **Real-Time Rendering** (4th ed.), CRC Press, 2018.
2. Perlin, K. “**An Image Synthesizer**,” *SIGGRAPH '85*, 1985.
3. Worley, S. “**A Cellular Texture Basis Function**,” *SIGGRAPH '96*, 1996.
4. Bridson, R. “**Fast Poisson Disk Sampling in Arbitrary Dimensions**,” *SIGGRAPH Sketch*, 2007.
5. Kimmel, R., Sethian, J.A. “**Computing Geodesic Paths on Manifolds**,” *PNAS*, 95(15), 1998.
6. Kaneko, T., et al. “**Detailed Shape Representation with Parallax Mapping**,” 2001; และ Pollicarpo, F., et al. “**Relief/Steep Parallax Mapping**,” *I3D 2005*.
7. Shorlin, K., et al. “**Crack Patterns in Drying Mud**,” *Phys. Rev. E*, 61(6), 2000.