

# ชื่อโครงการ (Title)

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

## บทคัดย่อ (Abstract)

โครงการนี้มุ่งพัฒนาวิธีการสร้าง “รอยแตก (cracks)” บนพื้นผิวสามมิติให้ดูเป็นธรรมชาติและมีความสมูอย่างสมเหตุสมผล เรานำแนวคิดเชิงกระบวนการ (procedural) ที่รวม Voronoi/Cellular noise, Poisson-disk sampling, และ ตัวแบบการเติบโตของรอยแตกตามเส้นแรงเค้นโดยประมาณ (stress-biased growth) มาสร้างเครือข่ายรอยแตก จากนั้นแมปรอยแตกสู่พื้นผิวด้วยฟังก์ชันระยะทาง/Geodesic distance บนเมช พร้อมแสดงผลด้วยเทคนิควีสดู (normal/parallax mapping) เพื่อให้เกิดความรู้สึกและเงาที่สมจริง เกณฑ์ประเมินประกอบด้วยความเร็ว (FPS), เวลาอัปเดตรอยแตก, และความสมจริงเชิงโครงสร้างของกราฟรอยแตกเทียบกับการกระจายตัวตามธรรมชาติ คาดหวังผลเป็นเดโมแบบอินเตอร์แอคทีฟและซอร์สโค้ดเปิดเผยบน GitHub ภายใน 7 สัปดาห์

## ความสำคัญและที่มาของปัญหา (Introduction & Background)

พื้นผิวที่แตกกร้าวเป็นองค์ประกอบสำคัญของความสมจริงในงานกราฟิกส์ ตั้งแต่พื้นคอนกรีตไปจนถึงวัสดุเซรามิก โจทย์คือสร้างรอยแตกที่ “สมแต่สมเหตุสมผล” ไม่เป็นลายซ้ำ และสะท้อนพฤติกรรมธรรมชาติ โครงการนี้ผสานแนวคิดเชิงกระบวนการ (Voronoi/Cellular noise, Poisson-disk sampling) กับแบบจำลองการเติบโตอิงแรงเค้นโดยประมาณ แล้วฉายรอยแตกสู่พื้นผิวของวัตถุ

## งานวิจัย/งานที่เกี่ยวข้อง (Related Work)

- Cellular/Worley/Voronoi และ Perlin noise ฐานสร้างขอบเขต/ความหยวนของลายแตก
- แบบจำลองการแตกในธรรมชาติ (เช่น รูปแบบในโคลแห่ง/สีแตกลายงา): ให้สมบัติทางสถิติเช่นการกระจายความยาวรอยแตก/มุมสาขา
- Geodesic distance บนเมช และ Poisson-disk sampling: ช่วยกำหนดจุดกำเนิดและการเติบโตของรอยแตกอย่างสม่ำเสมอแต่ไม่เป็นลายซ้ำ
- Parallax/Normal mapping: ยกระดับการรับรู้ความรู้สึกของร่องแตกโดยไม่ต้องเพิ่มโพลีกอนมากเกินไป

# คำถามการวิจัย (Research Questions)

1. วิธีที่เริ่มจากจุดกระจายหลาย ๆ จุด แล้วให้รอยแตกค่อย ๆ แผ่ขยายตามทิศที่น่าจะเกิดขึ้นจริง จะทำให้ลวดลายดูเป็นธรรมชาติกว่าวิธีแบ่งพื้นที่แบบทั่วไปหรือไม่?
2. การวางลายรอยแตกให้ไหลไปตามรูปทรงของวัตถุ ช่วยลดการบิดเบี้ยวของลายเมื่อเทียบกับการฉายจากระนาบธรรมดาได้มากน้อยเพียงใด?
3. พื้นผิวชนิดใดที่เหมาะสมกับงานวิจัยนี้มากที่สุด

# วิธีการดำเนินการ (Proposed Methodology)

## เครื่องมือและเทคโนโลยี

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

## ขั้นตอนอัลกอริทึมโดยสรุป

1. **Seed & Sampling:** สุ่มเมล็ดด้วย Poisson-disk บนโดเมน UV หรือบนเมช (ด้วยการประมาณ geodesic) เพื่อหลีกเลี่ยงการกระจุกตัว
2. **Voronoi Skeleton + Jitter:** สร้างเส้นขอบ Voronoi เป็น “โครงรอยแตกหลัก” แล้วใส่ jitter จาก Perlin/Worley noise เพื่อหลีกเลี่ยงเส้นตรงเกินไป
3. **Stress-Biased Growth:** คำนวณฟิลด์ “แรงเค้นโดยประมาณ” จากความโค้งพื้นผิว, ระยะจากจุดกระแตก, หรือทิศทางแรง ใช้เป็นน้ำหนักให้รอยแตกเติบโต/แตกกิ่งในทิศที่สมเหตุสมผล
4. **Crack Refinement:** ลบกิ่งตัน/กิ่งสั้นผิดธรรมชาติ, ทำ smoothing มุมแตกเล็กน้อย, และสร้าง crack width profile ที่แคบลงเมื่อไกลจากจุดกำเนิด
5. **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, เวลาอัปเดตกราฟรอยแตก/สร้างแผนที่ระยะทาง, หน่วยความจำ
- ความสมจริงเชิงโครงสร้าง: เปรียบเทียบกับรูปแบบธรรมชาติ
- การประเมินโดยผู้ใช้: ให้ผู้สังเกตให้คะแนนความสมจริง/ความหลากหลาย

## แผนการดำเนินงาน (Timeline: 7 สัปดาห์)

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

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

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

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

**สัปดาห์ 5:** แสดงผลแบบ normal และ parallax mapping; โปรไฟล์ประสิทธิภาพ

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

**สัปดาห์ 7:** ทำเดโมวิดีโอ, เขียนรายงานสรุปผล/วิเคราะห์, จัดเตรียมสไลด์นำเสนอ

## ผลที่คาดว่าจะได้รับ (Expected Outcomes)

- โปรแกรมเดโมแบบอินเตอร์แอคทีฟ (Windows/Linux) และซอร์สโค้ดบน GitHub
- ชุดพรีเซตพารามิเตอร์ สำหรับสร้างรอยแตกหลากหลายสไตล์
- รายงานผลการทดลอง: ประสิทธิภาพ, ตัวอย่างภาพก่อน/หลัง, และการวิเคราะห์สถิติของรอยแตก

## บรรณานุกรม (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 Policarpo, 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.