ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยเกษตรศาสตร์

418341: สภาวะแวดล้อมการทำงานคอมพิวเตอร์กราฟฟิกส์

ภาคต้น 2551

วันที่ 1 ตุลาคม 2551 เวลา 13.00 น. - 16.00 น.

ข้อสอบปลายภาค

ชื่อ _____ รหัส _____

1. ข้อสอบนี้มีทั้งหมด 10 หน้า
2. เวลาทำข้อสอบ**สามชั่วโมง**
3. เขียนคำตอบลายมือสวยๆ ตัวอักษรตัวโตๆ **ถ้าผมอ่านลายมือคุณไม่ออก ผมจะไม่ให้คะแนน**
4. กรุณาปิดอุปกรณ์สื่อสารทุกชนิดขณะทำข้อสอบ

ข้อ 1 (เต็ม 10)	**ข้อ 2 (เต็ม 25)**	**ข้อ 3 (เต็ม 20)**	**ข้อ 4 (เต็ม 25)**	**ข้อ 5 (เต็ม 20)**	**รวม (เต็ม 100)**

ข้อ 1

[ข้อย่อยละ 1 คะแนน] ข้อความต่อไปนี้ถูกหรือผิด จงวาดวงกลมล้อมรอบ “ถูก” หน้าข้อความที่ถูกต้อง และวาดวงกลมล้อมรอบ “ผิด” หน้าข้อความที่ผิด

ถูก	ผิด	ก. เราสามารถเขียนโปรแกรมควบคุม raster operations ด้วยภาษา Cg ได้
ถูก	ผิด	ข. Vertex shader จะต้องส่งข้อมูล normal ใน clip space เป็นข้อมูลออกเสมอ
ถูก	ผิด	ค. Fragment shader สามารถรับตำแหน่งของ vertex ใน clip space เป็นข้อมูลเข้าได้
ถูก	ผิด	ง. การใช้ environment map จำลองการสะท้อนแสงของวัตถุจะทำให้ไม่เห็นภาพสะท้อนของตัววัตถุเองบนพื้นผิวของวัตถุนั้น
ถูก	ผิด	จ. หน่วยความจำที่ใช้เพิ่มเติมในการทำ mipmap จะมีขนาดไม่มากกว่าขนาดของ texture ตั้งต้น
ถูก	ผิด	ฉ. z-buffer algorithm อาจไม่สามารถช่วยให้คำนวณสีของ fragment ต่างๆ ได้ถูกต้องถ้าหากในฉากมีวัตถุโปร่งแสงอยู่ด้วย
ถูก	ผิด	ช. ใน OpenGL เราสามารถทำ mipmap กับภาพที่มีขนาด 1000 x 1000 พิกเซลได้
ถูก	ผิด	ซ. ใน vertex program อันหนึ่ง พารามีเตอร์ float4 inPosition : POSITION และพารามีเตอร์ out float4 outPosition : POSITION ต่างเป็นตำแหน่งของ vertex ใน clip space
ถูก	ผิด	ฌ. ถ้า A เป็น matrix ที่ใช้แปลงจาก object space ไปยัง world space และให้ n เป็น normal ที่จุดๆ หนึ่งบนวัตถุใน object space แล้ว normal ที่จุดเดียวกันใน world space มีค่าเท่ากับ An
ถูก	ผิด	ญ. เราสามารถแก้ไข aliasing ที่เกิดขอบของรูปทรงได้ด้วยการใส่ low pass filter บนรูปทรงนั้น

ข้อ 2

[ข้อย่อยละ 5 คะแนน] จงอธิบายคำศัพท์เหล่านี้คร่าวๆ ด้วยประโยค 3-4 ประโยค **กรุณาเขียนสวย ๆ ตัวอักษรตัวโต ๆ**

ก. Aliasing

ข. Blending

ค. Chromatic Dispersion

๔. Fragment Shader

๗. Shadow Mapping

ข้อ 3

[20 คะแนน] จงวาดวงกลมล้อมรอบชื่ออัลกอริทึมที่สมบัติในคำถามต่อไปนี้เป็นจริง (แต่ละข้อสามารถวงกลมได้มากกว่าหนึ่งอัลกอริทึม หรือจะไม่วงเลยก็ได้)

ข้อ 3.1

เห็นเงาของวัตถุบนตัวของวัตถุเอง

วาดภาพวัตถุลงพื้นระนาบ shadow mapping z-pass shadow volume z-fail shadow volume

ข้อ 3.2

มีการสร้างรูปทรงหลายเหลี่ยมเพิ่มเติมจากโมเดลที่ศิลปินสร้างมาให้

วาดภาพวัตถุลงพื้นระนาบ shadow mapping z-pass shadow volume z-fail shadow volume

ข้อ 3.3

ความถูกต้องของอัลกอริทึมขึ้นอยู่กับว่า “ตา” อยู่ในเงาหรือไม่

วาดภาพวัตถุลงพื้นระนาบ shadow mapping z-pass shadow volume z-fail shadow volume

ข้อ 3.4

ทำให้เงามี aliasing

วาดภาพวัตถุลงพื้นระนาบ shadow mapping z-pass shadow volume z-fail shadow volume

ข้อ 3.5

ถ้าเงามีขนาดใหญ่ขึ้นก็จะเสียเวลาประมวลผลมากขึ้น

วาดภาพวัตถุลงพื้นระนาบ shadow mapping z-pass shadow volume z-fail shadow volume

ข้อ 3.6

สามารถสร้างเงามัวได้

วาดภาพวัตถุลงพื้นระนาบ shadow mapping z-pass shadow volume z-fail shadow volume

ข้อ 4

จงเติม vertex program และ fragment program ซึ่งเขียนในภาษา Cg ในข้อ 4.1 และ 4.2 ให้สมบูรณ์ เพื่อให้สีวัตถุดังกำหนดต่อไปนี้

> ถ้าเวกเตอร์ normal ที่ normalized แล้วใน ***WORLD SPACE*** ของ fragment หนึ่งมีค่าเท่ากับเวกเตอร์ (x, y, z) โดยที่ $-1 \leq x, y, z \leq 1$ แล้ว ให้ fragment นั้นมีสี:
>
> - $R = (x + 1)/2$
> - $G = (y + 1)/2$
> - $B = (z + 1)/2$

ในช่องว่างที่กำหนดให้

ข้อ 4.1

[15 คะแนน] Vertex Program

```
void main(float4 inPosition : POSITION,
          float4 inNormal : NORMAL,
          uniform float4x4 mvp : state.matrix.mvp,
          uniform float4x4 modeling,
          uniform float4x4 modelingInverseTranspose,
          out float4 outPosition : POSITION,
          out float3 texCoord0 : TEXCOORD0)
{









}
```

โดยที่

`mvp` เป็น uniform parameter ที่บรรจุผลคูณของ projection matrix และ modelview matrix

`modeling` เป็น uniform parameter ที่บรรจุ modeling matrix

`modelingInverseTranpose` เป็น uniform parameter ที่บรรจุ inverse transpose ของ modeling matrix

ข้อ 4.2

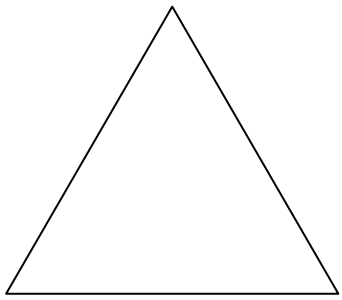
[10 คะแนน] Fragment Program

```
void main(float3 texCoord0 : TEXCOORD0,
          out float3 color : COLOR)
{









}
```

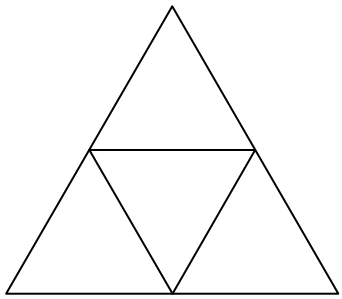
ข้อ 5

[20 คะแนน] Sierpiński triangle เป็นแฟรกตัล (รูปที่พอเอาแว่นขยายส่องดูแล้วเห็นเป็นลักษณะเหมือนกับตอนไม่ได้ใช้แว่นขยายดู) รูปแบบหนึ่ง เพื่อความง่าย เราจะแบ่ง Sierpiński triangle ออกเป็น “ชั้น” โดยที่

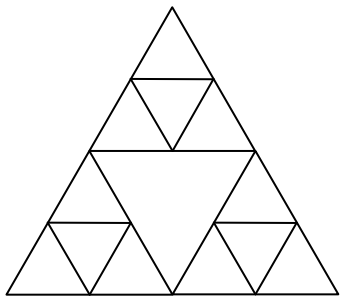
- Sierpiński triangle ชั้นที่ 0 เป็นสามเหลี่ยมด้านเท่าธรรมดา



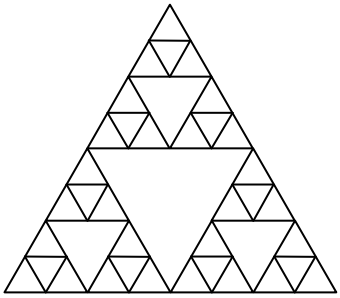
- Sierpiński Triangle ชั้นที่ 1 เกิดจากการเอา Sierpiński triangle ชั้นที่ 0 ที่ย่อส่วนลงสองเท่ามาเรียงกันตามรูปข้างล่างนี้



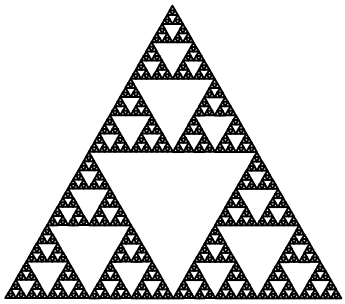
- Sierpiński Triangle ชั้นที่ 2 เกิดจากการเอา Sierpiński triangle ชั้นที่ 1 ที่ย่อส่วนลงสองเท่ามาเรียงกันตามรูปข้างล่างนี้



- Sierpiński Triangle ชั้นที่ 3 เกิดจากการเอา Sierpiński triangle ชั้นที่ 2 ที่ย่อส่วนลงสองเท่ามาเรียงกันตามรูปข้างล่างนี้



และ Sierpiński Triangle ชั้นที่ k เกิดจากการเอา Sierpiński triangle ชั้นที่ k-1 ที่ย่อส่วนลงสองเท่ามาเรียงกันตามรูปแบบที่เห็นข้างบน และข้างล่างนี้คือ Sierpiński triangle ประมาณชั้นที่ 8



จงเขียนฟังก์ชัน void drawSierpinski(int k) เพื่อวาดรูป Sierpiński Triangle ชั้นที่ k สำหรับจำนวนเต็ม k ≥ 0 ใดๆ โดยที่ Sierpiński Triangle นี้จะต้องมีขอบรอบนอกสุดเป็นรูปสามเหลี่ยมด้านเท่าที่มีความยาวด้านละสองหน่วย และมีจุดยอดมุมอยู่ที่จุด $\left(1,-\frac{1}{\sqrt{3}}\right)$, $\left(-1,-\frac{1}{\sqrt{3}}\right)$, และ $\left(0,\frac{2}{\sqrt{3}}\right)$

เพื่อความง่าย คุณสามารถเรียกใช้ฟังก์ชัน void drawTriangle() เพื่อวาดสามเหลี่ยมด้านเท่าดังกล่าวได้

ใบ้: ให้เขียนโปรแกรมแบบ recursive

```
void drawSierpinski(int k)
{









}
```