418341 สภาพแวดล้อมการทำงานคอมพิวเตอร์กราฟิกส์
การบรรยายครั้งที่ 20

ประมุข ขันเงิน

pramook@gmail.com

Uniform Parameter

- คือ parameter ที่ผู้ใช้เป็นคนกำหนดให้กับ vertex program หรือ fragment program เอง
- ประกาศเป็น argument ของฟังก์ชันที่เราจะเขียน
- ใส่ reservered word “uniform” ไว้ข้างหน้า

ตัวอย่าง Fragment Program

```
void main(uniform float4 color,
                out float4 outColor : COLOR)
{
   outColor = color;
}
```

การกำหนดค่าให้ uniform parameter

- ประกาศตัวแปรประเภท CGparameter

```
CGparameter color;
```

- แล้วใช้ฟังก์ชัน cgGetNamedParameter(…) สร้าง parameter ขึ้นมาจริงๆ
  - Parameter ตัวแรกคือโปรแกรม (ตัวแปรประเภท CGprogram)
  - Parameter ตัวที่สองคือสตริงที่มีค่าเท่ากับชื่อตัวแปร
  - ตัวอย่าง

```
color = cgGetNamedParamater(
        fragmentProgram, “color”);
```

การกำหนดค่าให้ uniform parameter (ต่อ)

- เสร็จแล้วทำการเซ็ตค่าด้วย cgSetParameter[1234][idfs](…)
- ยกตัวอย่างเช่นในกรณี color

```
cgSetParameter3f(color, 1, 1, 0);
```

- หลังจากเซตค่า parameter ให้หลายๆ ตัวเสร็จแล้ว ให้เรียกฟังก์ชัน cgUpdateProgramParameter([โปรแกรม]) เพื่อสั่งให้ Cg Runtime ปรับค่า uniform parameter ให้เป็นไปตาที่เราตั้ง

ตัวอย่างการใช้

```
cgSetParameter3f(color, 1,1,0);
cgUpdateProgramParameters(fragmentProgram);
glBegin(GL_TRIANGLES);
    glVertex3f(0.5f, -0.5f, 0);
    glVertex3f(0, 0.5f, 0);
    glVertex3f(-0.5f, -0.5f, 0);
glEnd();
```

Texture Parameter

- Texture parameter อนุญาตให้ผู้เขียนโปรแกรม Cg อ่านข้อมูลจาก texture มาได้
- เราสามารถใช้ texture parameter ได้เฉพาะใน fragment program เท่านั้น
- Texture parameter จะต้องประกาศเป็น uniform parameter ของ fragment program

Texture Parameter (ต่อ)

- Texture parameter สามารถมีชนิดข้อมูลได้ 2 แบบคือ
  - sampler2D
    - มี texture coordinate (u,v) โดยที่ $0 \le u, v \le 1$
    - ขนาดของ texture ต้องเป็นเลข 2^k
    - สามารถทำ mipmap ได้
  - samplerRECT
    - มี texture coordinate (u,v) โดยที่ $0 \le u \le$ ความกว้าง และ $0 \le v \le$ ความสูง ของ texture
    - ขนาดของ texture ไม่จำเป็นต้องเป็นเลข 2^k
    - ไม่สามารถทำ mipmap ได้

Texture Parameter (ต่อ)

- เวลาจะอ่านข้อมูลจาก sampler2D ต้องอ่านโดยใช้ฟังก์ชัน tex2D(…)
  - สมมติว่า texture เป็นตัวแปรชนิด tex2D
  - สมมติว่า texCoord เป็นตัวแปรชนิด float2
  - เราสามารถอ่านข้อมูลจาก texture ณ texture coordinate texCoord ได้โดยใช้ tex2D(texture, texCoord)
  - มันจะคืนตัวแปรชนิด float4 (RGBA) มาให้
  - ตัวอย่าง: float4 color = tex2D(texture, texCoord)

Texture Parameter (ต่อ)

- เวลาจะอ่านข้อมูลจาก samplerRECT ต้องอ่านโดยใช้ฟังก์ชัน texRECT(…)
  - สมมติว่า texture เป็นตัวแปรชนิด tex2D
  - สมมติว่า texCoord เป็นตัวแปรชนิด float2
  - เราสามารถอ่านข้อมูลจาก texture ณ texture coordinate texCoord ได้โดยใช้ texRECT(texture, texCoord)
  - มันจะคืนตัวแปรชนิด float4 (RGBA) มาให้
  - ตัวอย่าง: float4 color = texRECT(texture, texCoord)

ตัวอย่าง fragment program

```
void main(float2 texCoord : TEXCOORD0,
          out float4 color : COLOR,
          uniform sampler2D texture)
{
   color = tex2D(texture, texCoord);
}
```

การตั้งค่าในภาษา C

- เราจะต้องมีการสร้างตัวแปร CGparameter เพื่ออ้างถึง texture parameter เหมือนกับในคาบที่แล้ว
- นี่เป็นเพราะว่า texture parameter เป็น uniform parameter แบบหนึ่ง

การตั้งค่าในภาษา C (ต่อ)

- สมมติว่าเราจะใช้ fragment program ใน slide ที่แล้ว และเราใช้ตัวแปร fragmentProgram เป็นตัวแทนของ fragmentProgram นี้
- ขั้นแรกให้ประกาศตัวแปรประเภท CGparameter ชื่อ texture

  CGparameter texture;
- แล้วสั่ง

  texture = cgGetNamedParameter(fragmentProgram,
  “texture”);

การตั้งค่าในภาษา C (ต่อ)

- หลังจากนั้นให้ทำการสร้าง texture เหมือนกับทำปกติใน OpenGL (ดูการบรรยายครั้งที่ 15)
- เมื่อทำเสร็จแล้วคุณจะมีตัวแปร GLuint ไว้เก็บเลขของ texture หนึ่งตัว
- สมมติว่าตัวแปรตัวนี้คือ texID

การตั้งค่าในภาษา C (ต่อ)

- ก่อนจะทำการ render ให้เรียก

  cgGLSetTextureParameter(texture, texID);
- แล้วเรียก

  cgGLEnableTextureParameter(texture);
- แล้วจึงทำการ render ตามปกติ

ตัวอย่างโค้ด

```
void initCg()
{
    context = cgCreateContext();
    cgGLSetDebugMode( CG_FALSE );
    cgSetParameterSettingMode(context, CG_DEFERRED_PARAMETER_SETTING);

    fragmentProfile = cgGLGetLatestProfile(CG_GL_FRAGMENT);
    cgGLSetOptimalOptions(fragmentProfile);

    fragmentProgram = cgCreateProgramFromFile(context, CG_SOURCE, "fragment-program.cg“,
            fragmentProfile, "main“, NULL);
    cgGLLoadProgram(fragmentProgram);

    texture = cgGetNamedParameter(fragmentProgram, "texture");
}
```

ตัวอย่างโค้ด (ต่อ)

```
void initCg()
{
    context = cgCreateContext();
    cgGLSetDebugMode( CG_FALSE );
    cgSetParameterSettingMode(context, CG_DEFERRED_PARAMETER_SETTING);

    fragmentProfile = cgGLGetLatestProfile(CG_GL_FRAGMENT);
    cgGLSetOptimalOptions(fragmentProfile);

    fragmentProgram = cgCreateProgramFromFile(context, CG_SOURCE, "fragment-program.cg“,
            fragmentProfile, "main“, NULL);
    cgGLLoadProgram(fragmentProgram);

    texture = cgGetNamedParameter(fragmentProgram, "texture");
}
```

ให้ texture แทน uniform parameter ชื่อเดียวกันใน fragment program

ตัวอย่างโค้ด (ต่อ)

```
Gluint texID;

void initTexture()
{
    glGenTextures(1, &texID);
    glBindTexture(GL_TEXTURE_2D, texID);

            :
            :
            :

    glTexImage2D(GL_TEXTURE_2D, 0, ilGetInteger(IL_IMAGE_BPP),
    ilGetInteger(IL_IMAGE_WIDTH), ilGetInteger(IL_IMAGE_HEIGHT), 0,
            ilGetInteger(IL_IMAGE_FORMAT), GL_UNSIGNED_BYTE, ilGetData());

    ilDeleteImages(1, &image);
}
```

ตัวอย่างโค้ด (ต่อ)

สร้าง texture โดยเก็บชื่อของ texture ไว้ในตัวแปร texID

```
Gluint texID;

void initTexture()
{
    glGenTextures(1, &texID);
    glBindTexture(GL_TEXTURE_2D, texID);

            :
            :
            :

    glTexImage2D(GL_TEXTURE_2D, 0, ilGetInteger(IL_IMAGE_BPP),
    ilGetInteger(IL_IMAGE_WIDTH), ilGetInteger(IL_IMAGE_HEIGHT), 0,
            ilGetInteger(IL_IMAGE_FORMAT), GL_UNSIGNED_BYTE, ilGetData());

    ilDeleteImages(1, &image);
}
```

ตัวอย่างโค้ด (ต่อ)

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    cgGLEnableProfile(fragmentProfile);
    cgGLBindProgram(fragmentProgram);

    cgGLSetTextureParameter(texture, texID);
    cgGLEnableTextureParameter(texture);

    glBegin(GL_TRIANGLES);
        glTexCoord2f(0, 0); glVertex2f(-0.8, 0.8);
        glTexCoord2f(1, 0); glVertex2f(0.8, 0.8);
        glTexCoord2f(0.5, 1); glVertex2f(0.0, -0.8);
    glEnd();
    cgGLDisableProfile(fragmentProfile);
    glutSwapBuffers();
}
```

ตัวอย่างโค้ด (ต่อ)

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    cgGLEnableProfile(fragmentProfile);
    cgGLBindProgram(fragmentProgram);

    cgGLSetTextureParameter(texture, texID);
    cgGLEnableTextureParameter(texture);

    glBegin(GL_TRIANGLES);
        glTexCoord2f(0, 0); glVertex2f(-0.8, 0.8);
        glTexCoord2f(1, 0); glVertex2f(0.8, 0.8);
        glTexCoord2f(0.5, 1); glVertex2f(0.0, -0.8);
    glEnd();
    cgGLDisableProfile(fragmentProfile);
    glutSwapBuffers();
}
```

Set ชื่อของ texture แล้ว enable มัน