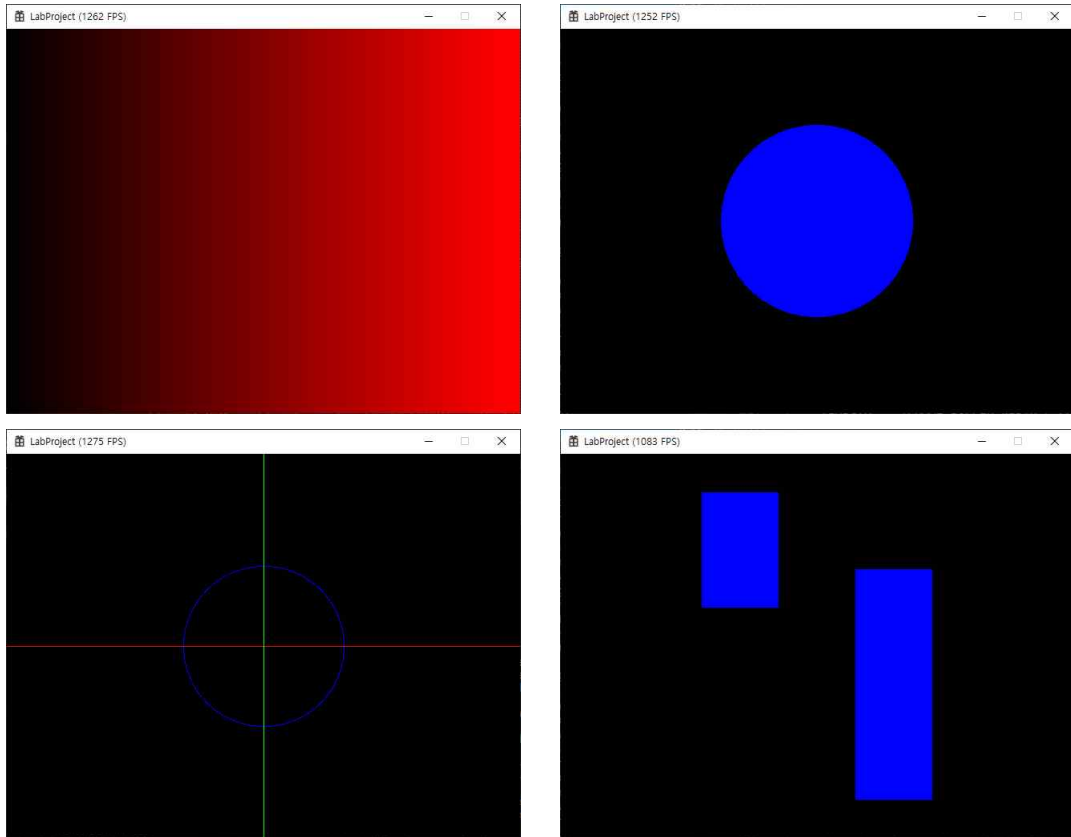


## □ 예제 프로그램 6: LabProject05-01

LabProject05 프로젝트를 기반으로 여러 가지 픽셀 셰이더를 생성해 본다.



### ① 새로운 프로젝트의 생성

먼저 새로운 프로젝트 LabProject05-01을 생성하자. “LabProjects” 솔루션을 열고 솔루션 탐색기에서 마우스 오른쪽 버튼으로 『솔루션 LabProjects』를 선택하고 메뉴에서 『추가』, 『새 프로젝트』를 차례로 선택한다. 그러면 『새 프로젝트 대화상자』가 나타난다. 그러면 프로젝트 이름 “LabProject05-01”을 입력하고 『확인』을 선택한다.

### ② LabProject05-01.cpp 파일 수정하기

이제 “LabProject05-01.cpp” 파일의 내용을 “LabProject05.cpp” 파일의 내용으로 바꾸도록 하자. “LabProject05.cpp” 파일의 내용 전체를 “LabProject05-01.cpp” 파일로 복사한다. 이제 “LabProject05-01.cpp” 파일에서 “LabProject05”를 “LabProject05-01”로 모두 바꾼다. 그리고 “LABPROJECT05”을 “LABPROJECT0501”로 모두 바꾼다.

③ "Scene.cpp" 파일을 다음과 같이 수정한다.

❶ Render() 함수를 다음과 같이 수정한다.

```
void CScene::Render(ID3D12GraphicsCommandList *pd3dCommandList)
{
    //그래픽 루트 시그니처를 설정한다.
    pd3dCommandList->SetGraphicsRootSignature(m_pd3dGraphicsRootSignature);
    //파이프라인 상태를 설정한다.
    pd3dCommandList->SetPipelineState(m_pd3dPipelineState);
    //프리티미브 토폴로지(삼각형 리스트)를 설정한다.
    pd3dCommandList->IASetPrimitiveTopology(D3D_PRIMITIVE_TOPOLOGY_TRIANGLELIST);
    //정점 6개를 사용하여 렌더링한다.
    pd3dCommandList->DrawInstanced(6, 1, 0, 0);
}
```

④ "Shaders.hlsl" 파일 수정하기

❶ "Shaders.hlsl" 파일의 정점 셰이더 VSMain() 함수를 다음과 같이 수정한다.

```
//정점 셰이더를 정의한다. 렌더 타겟 사각형 전체를 삼각형 2개로 표현한다.
float4 VSMain(uint nVertexID : SV_VertexID) : SV_POSITION
{
    float4 output = (float4)0;

    if (nVertexID == 0) output = float4(-1.0f, +1.0f, 0.0f, 1.0f);
    else if (nVertexID == 1) output = float4(+1.0f, +1.0f, 0.0f, 1.0f);
    else if (nVertexID == 2) output = float4(+1.0f, -1.0f, 0.0f, 1.0f);

    else if (nVertexID == 3) output = float4(-1.0f, +1.0f, 0.0f, 1.0f);
    else if (nVertexID == 4) output = float4(+1.0f, -1.0f, 0.0f, 1.0f);
    else if (nVertexID == 5) output = float4(-1.0f, -1.0f, 0.0f, 1.0f);

    return(output);
}
```

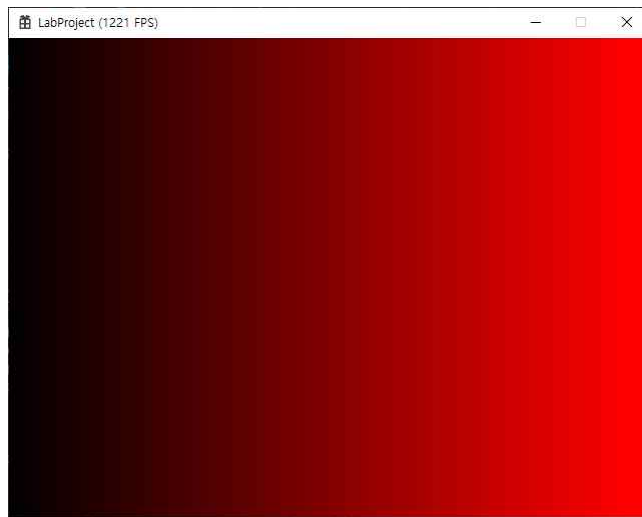
❷ "Shaders.hlsl" 파일의 픽셀 셰이더 PSMain() 함수를 다음과 같이 여러 가지로 수정해 본다.

```
#define FRAME_BUFFER_WIDTH      640.0f
#define FRAME_BUFFER_HEIGHT    480.0f
```

//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

```
float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);
    cColor.r = input.x / FRAME_BUFFER_WIDTH;

    return(cColor);
}
```

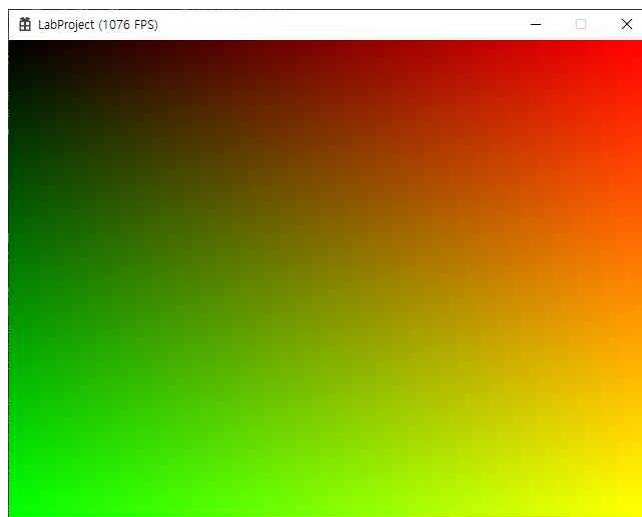


//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

```
float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

    cColor.r = input.x / FRAME_BUFFER_WIDTH;
    cColor.g = input.y / FRAME_BUFFER_HEIGHT;

    return(cColor);
}
```



//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

```
float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

    // cColor.rgb = distance(float2(0.0f, 0.0f), input.xy) / float2(FRAME_BUFFER_WIDTH,
```

```

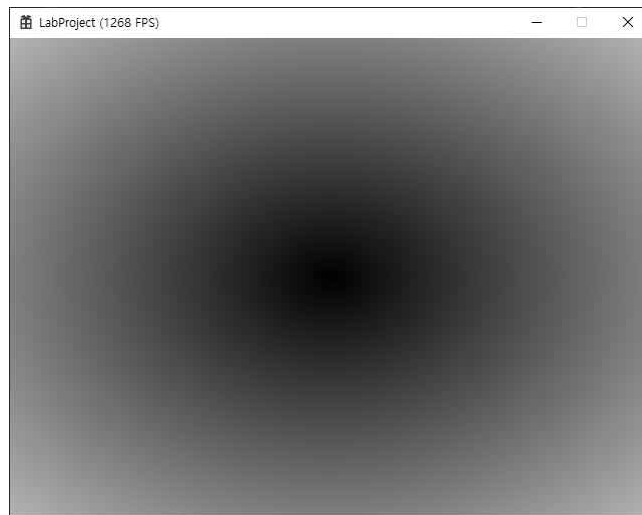
FRAME_BUFFER_HEIGHT));
// cColor.rgb = length(input.xy / float2(FRAME_BUFFER_WIDTH, FRAME_BUFFER_HEIGHT));

// cColor.rgb = distance(float2(0.0f, 0.0f), (input.xy / float2(FRAME_BUFFER_WIDTH,
FRAME_BUFFER_HEIGHT) - 0.5f)); // (0, 1) : (-0.5, 0.5)
// cColor.rgb = length(input.xy / float2(FRAME_BUFFER_WIDTH, FRAME_BUFFER_HEIGHT) -
0.5f);

    cColor.rgb = distance(float2(0.5f, 0.5f), input.xy / float2(FRAME_BUFFER_WIDTH,
FRAME_BUFFER_HEIGHT));

    return(cColor);
}

```



//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

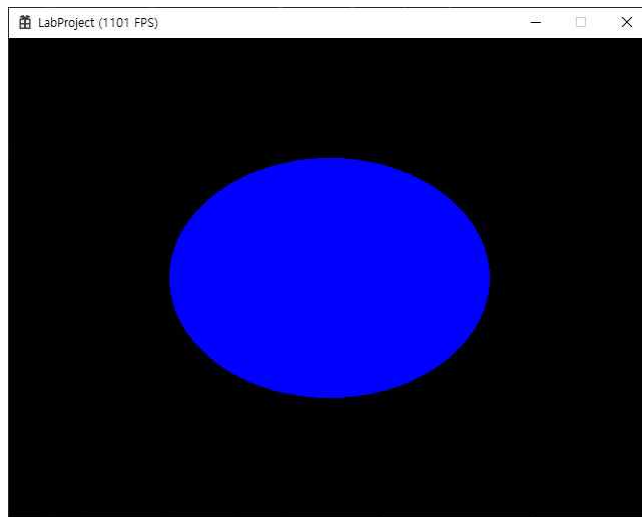
```

float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

    float fDistance = distance(float2(0.5f, 0.5f), input.xy / float2(FRAME_BUFFER_WIDTH,
FRAME_BUFFER_HEIGHT));
    if (fDistance < 0.25f) cColor.b = 1.0f;

    return(cColor);
}

```



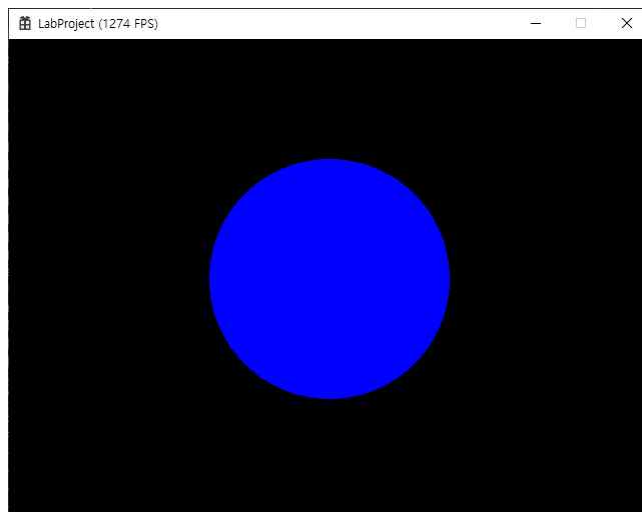
//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

```
float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

    float2 f2NDC = float2(input.x / FRAME_BUFFER_WIDTH, input.y / FRAME_BUFFER_HEIGHT) -
0.5f; // (0, 1) : (-0.5, 0.5)
    f2NDC.x *= (FRAME_BUFFER_WIDTH / FRAME_BUFFER_HEIGHT);

    // cColor.b = step(length(f2NDC), 0.25f); //step(x, y) = (x <= y) ? 1 : 0
    cColor.b = (length(f2NDC) <= 0.25f) ? 1.0f : 0.0f; //step(x, y) = (x <= y) ? 1 : 0

    return(cColor);
}
```



//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

```
float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
```

```

float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

float2 f2NDC = float2(input.x / FRAME_BUFFER_WIDTH, input.y / FRAME_BUFFER_HEIGHT) -
0.5f;
f2NDC.x *= (FRAME_BUFFER_WIDTH / FRAME_BUFFER_HEIGHT);

float fLength = length(f2NDC);
float fMin = 0.3f, fMax = 0.2f;
cColor.rgb = smoothstep(fMin, fMax, fLength);

return(cColor);
}

```



//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

```

#define HALF_WIDTH          (FRAME_BUFFER_WIDTH * 0.5f)
#define HALF_HEIGHT        (FRAME_BUFFER_HEIGHT * 0.5f)

#define EPSILON              1.0e-5f

inline bool IsZero(float fvalue) { return((abs(fvalue) <= EPSILON)); }
inline bool IsZero(float fvalue, float fEpsilon) { return((abs(fvalue) <= fEpsilon)); }
inline bool IsEqual(float fA, float fB, float fEpsilon) { return((abs(fA - fB) <= fEpsilon)); }

float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

    if ((int)input.x == (int)HALF_WIDTH) cColor.g = 1.0f;
    if ((int)input.y == (int)HALF_HEIGHT) cColor.r = 1.0f;

    float fDistance = distance((int2)input.xy, float2(HALF_WIDTH, HALF_HEIGHT));
    if (IsEqual(fDistance, 100.0f, 0.5f)) cColor.b = 1.0f;

    return(cColor);
}

```



//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

```
float Rectangle(float2 f2NDC, float fLeft, float fRight, float fTop, float fBottom)
{
    float2 f2Shape = float2(step(fLeft, f2NDC.x), step(f2NDC.x, fRight));
    f2Shape *= float2(step(fTop, f2NDC.y), step(f2NDC.y, fBottom));

    return(f2Shape.x * f2Shape.y);
}

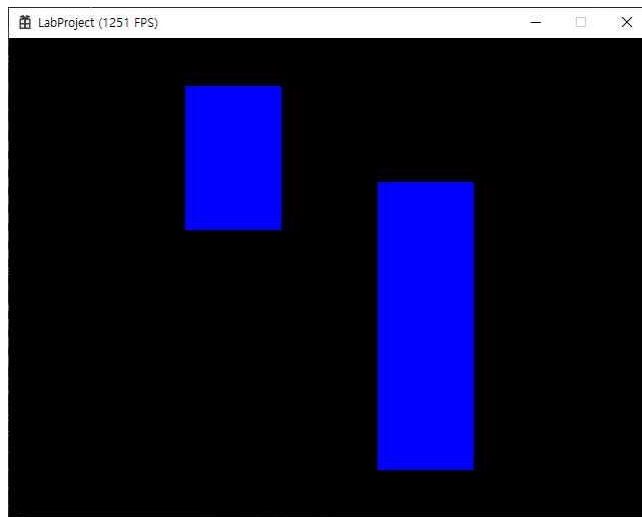
float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

    float2 f2NDC = float2(input.x / FRAME_BUFFER_WIDTH, input.y / FRAME_BUFFER_HEIGHT) -
0.5f; // (0, 1) : (-0.5, 0.5)
    f2NDC.x *= (FRAME_BUFFER_WIDTH / FRAME_BUFFER_HEIGHT);

    float2 f2Horizontal = float2(0.1f, +0.3f); //(Left, Right)
    float2 f2Vertical = float2(-0.3f, +0.3f); //(Top, Bottom)

    cColor.b = Rectangle(f2NDC, +0.1f, +0.3f, -0.2f, +0.4f);
    cColor.b += Rectangle(f2NDC, -0.3f, -0.1f, -0.4f, -0.1f);

    return(cColor);
}
```



//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

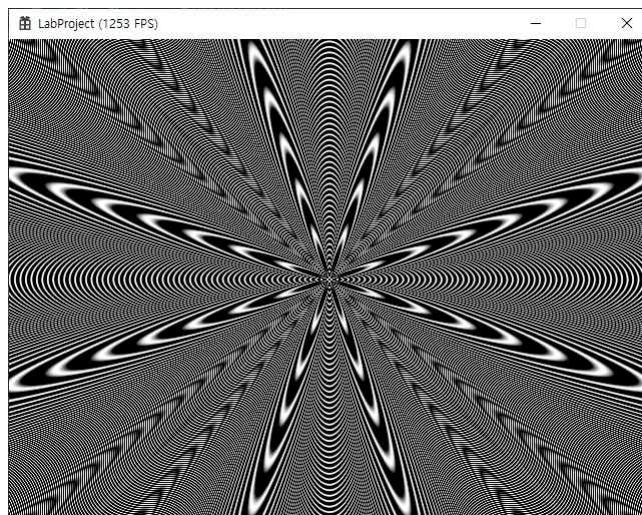
```
float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

    float2 f2NDC = float2(input.x - FRAME_BUFFER_WIDTH * 0.5f, input.y -
FRAME_BUFFER_HEIGHT * 0.5f);

    f2NDC *= 20.0f;

    float fLength = length(f2NDC);
    cColor.rgb = cos(fLength);

    return(cColor);
}
```



//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

```
float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
```



```

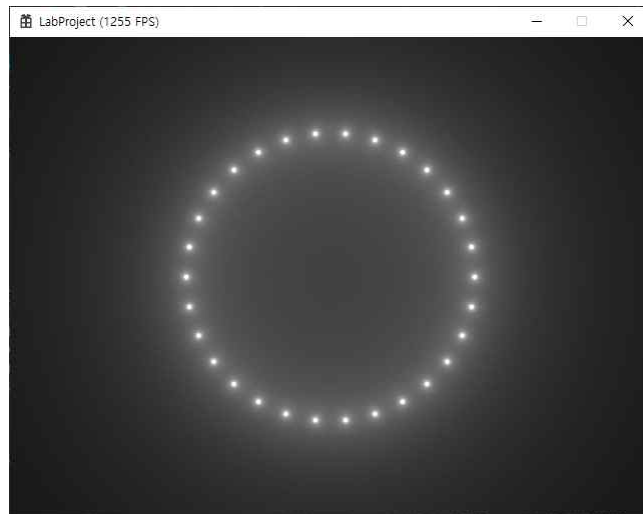
float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

float2 f2NDC = float2(input.x / FRAME_BUFFER_WIDTH, input.y / FRAME_BUFFER_HEIGHT) -
0.5f; // (0, 1) : (-0.5, 0.5)
f2NDC.x *= (FRAME_BUFFER_WIDTH / FRAME_BUFFER_HEIGHT);

float fRadius = 0.3f;
float fRadian = radians(360.0f / 30.0f);
for (float f = 0; f < 30.0f; f += 1.0f)
{
    float fAngle = fRadian * f;
    cColor.rgb += (0.0025f / length(f2NDC + float2(fRadius * cos(fAngle), fRadius *
sin(fAngle)))));
}

return(cColor);
}

```



//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

```

float RegularPolygon(float2 f2NDC, float fSides, float fRadius)
{
    float fAngle = atan(f2NDC.y / f2NDC.x);
    float fSlices = (2.0f * 3.14159f) / fSides;

    float fShape = step(cos(floor((fAngle / fSlices) + 0.5f) * fSlices - fAngle) *
length(f2NDC), fRadius);

    return(fShape);
}

float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

    float2 f2NDC = float2(input.x / FRAME_BUFFER_WIDTH, input.y / FRAME_BUFFER_HEIGHT) -
0.5f; // (0, 1) : (-0.5, 0.5)
    f2NDC.x *= (FRAME_BUFFER_WIDTH / FRAME_BUFFER_HEIGHT);

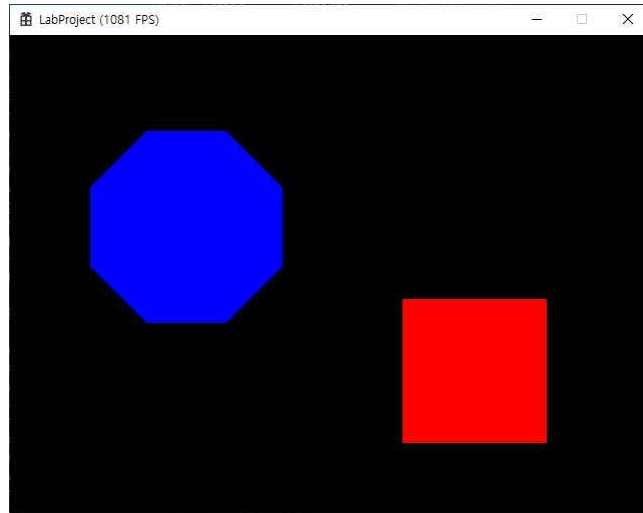
```

```

cColor.b = RegularPolygon(f2NDC - float2(-0.3f, -0.1f), 8.0f, 0.2f); //4, 6, 8, ...
cColor.r = RegularPolygon(f2NDC - float2(+0.3f, +0.2f), 4.0f, 0.15f);

return(cColor);
}

```



//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

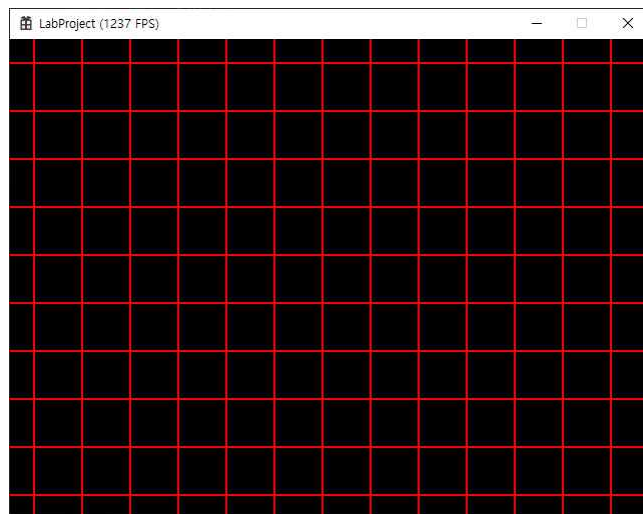
```

float4 PSMain(float4 input : SV_POSITION) : SV_TARGET
{
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);

    float x = abs(frac((input.x * 10.0f) / FRAME_BUFFER_HEIGHT) - 0.5f);
    float y = abs(frac((input.y * 10.0f) / FRAME_BUFFER_HEIGHT) - 0.5f);
    cColor.r = ((x <= 0.0125f) || (y <= 0.0125f)) ? 1.0f : 0.0f;

    return(cColor);
}

```



//픽셀 셰이더를 다음과 같이 정의하여 실행해 보면 다음과 같은 결과를 볼 수 있다.

```

float4 PSMain(float4 input : SV_POSITION) : SV_TARGET

```

```
{  
    float4 cColor = float4(0.0f, 0.0f, 0.0f, 1.0f);  
  
    float2 f2NDC = input.xy / float2(FRAME_BUFFER_WIDTH, FRAME_BUFFER_HEIGHT) - 0.5f;  
    f2NDC.x *= (FRAME_BUFFER_WIDTH / FRAME_BUFFER_HEIGHT);  
    f2NDC.xy *= 10.0f;  
  
    cColor.b = smoothstep(0.125f, 0.875f, abs(cos(length(f2NDC) * 3.14159f)));  
  
    return(cColor);  
}
```

