

Laboratori de Gràfics

Sessió 5

Interpolació per fragment

- Tot el que s'interpola per cada fragment (coords x,y,z, coords de textura s,t, out's definits per l'usuari) es calcula al **centre del pixel** corresponent. Per tant:

`fract(glFragCoord.x)` serà 0.5

`fract(glFragCoord.y)` serà 0.5

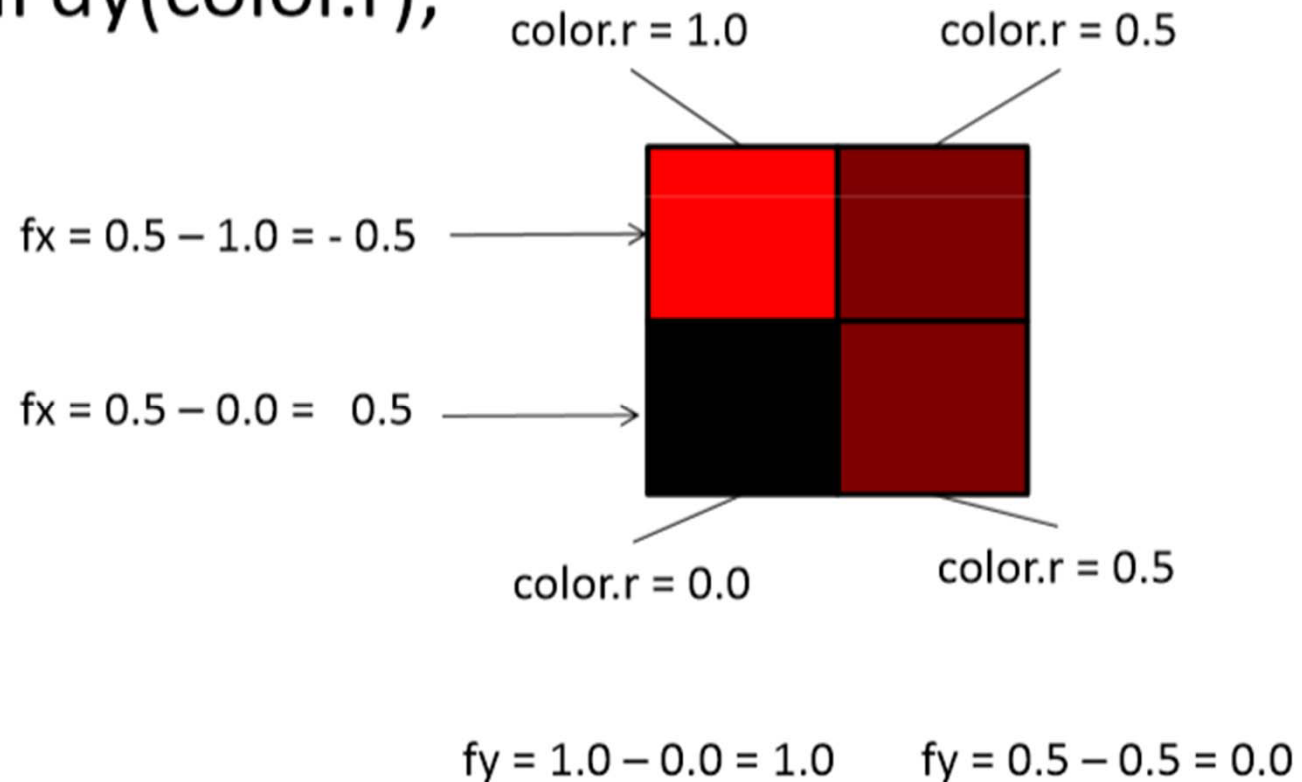
- En algunes versions de GLSL, és pot eliminar aquest offset redeclarant `gl_FragCoord`

`layout(pixel_center_integer) in vec4 gl_FragCoord;`

dFdx, dFdy - exemple

```
float fx = dFdx(color.r);
```

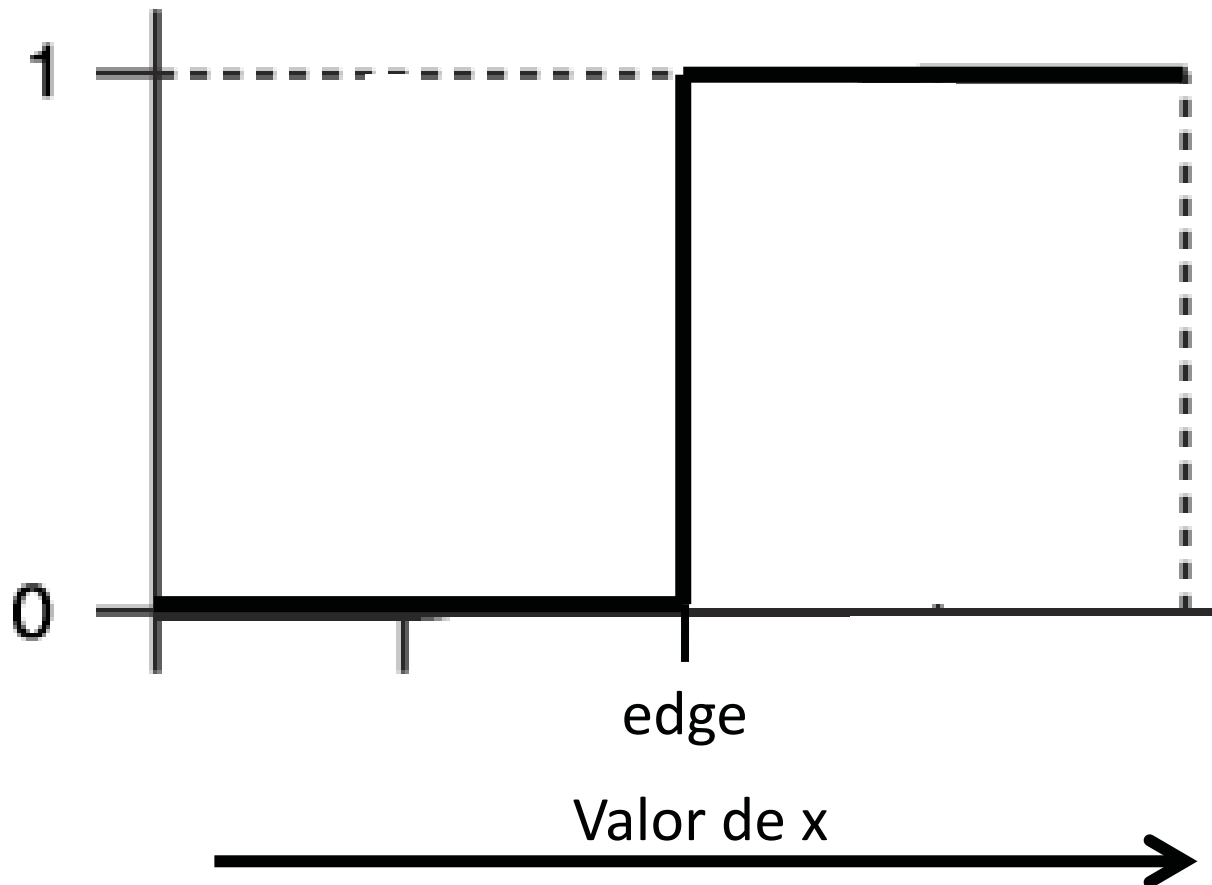
```
float fy = dFdy(color.r);
```



Functions step, smoothstep

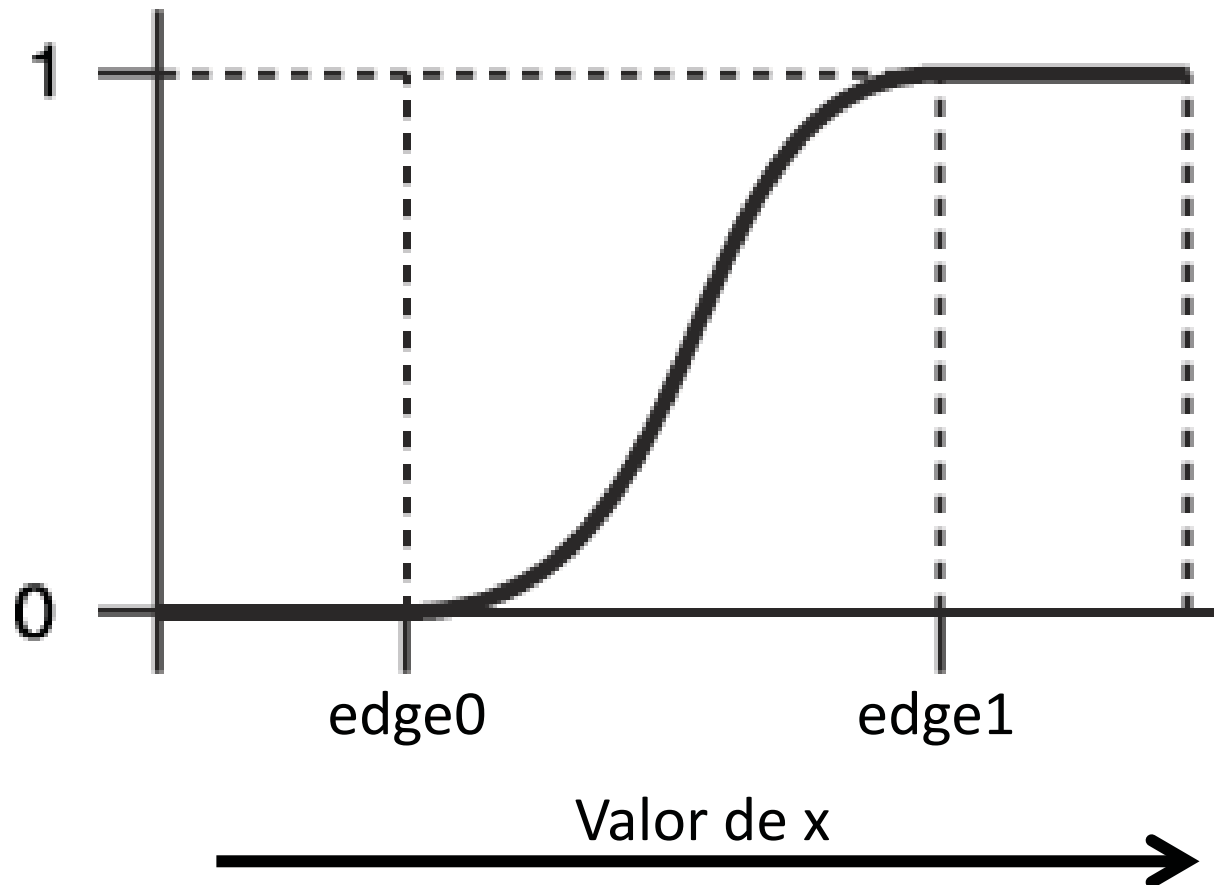
float step(float edge, float x)

$$\begin{cases} 0 & \text{if } x < \text{edge} \\ 1 & \text{otherwise} \end{cases}$$



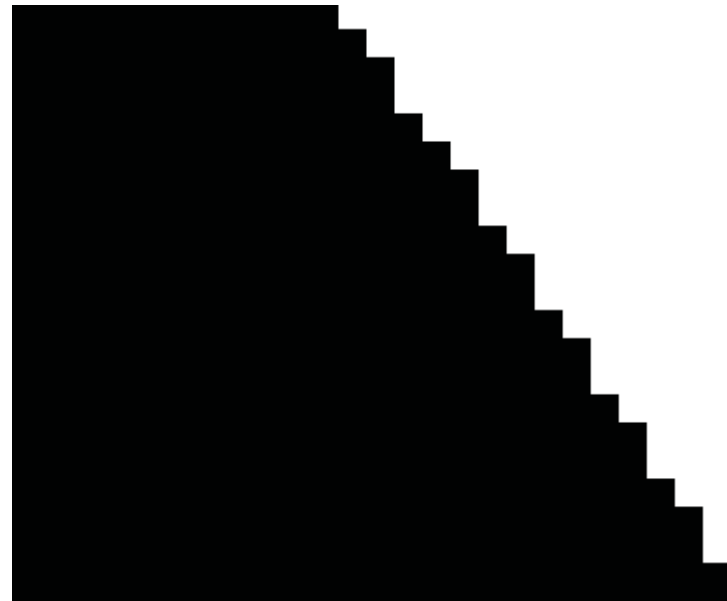
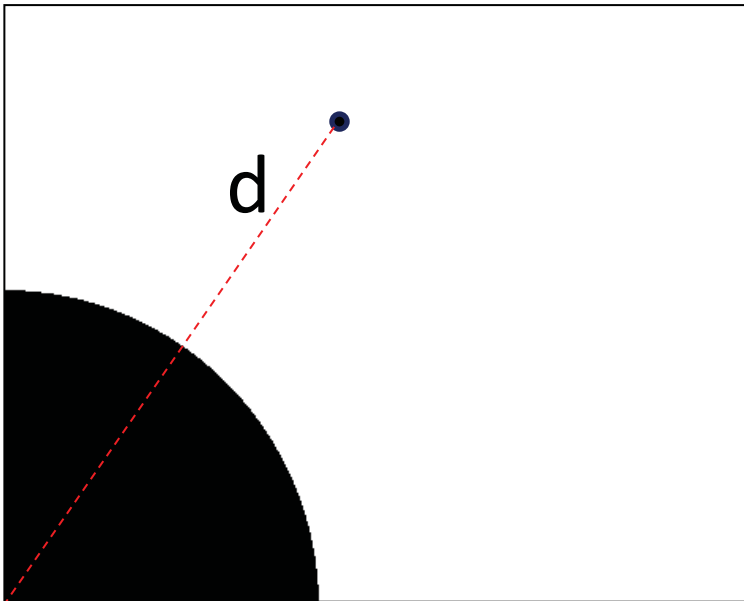
Functions step, smoothstep

float smoothstep(float edge0, float edge1, float x)



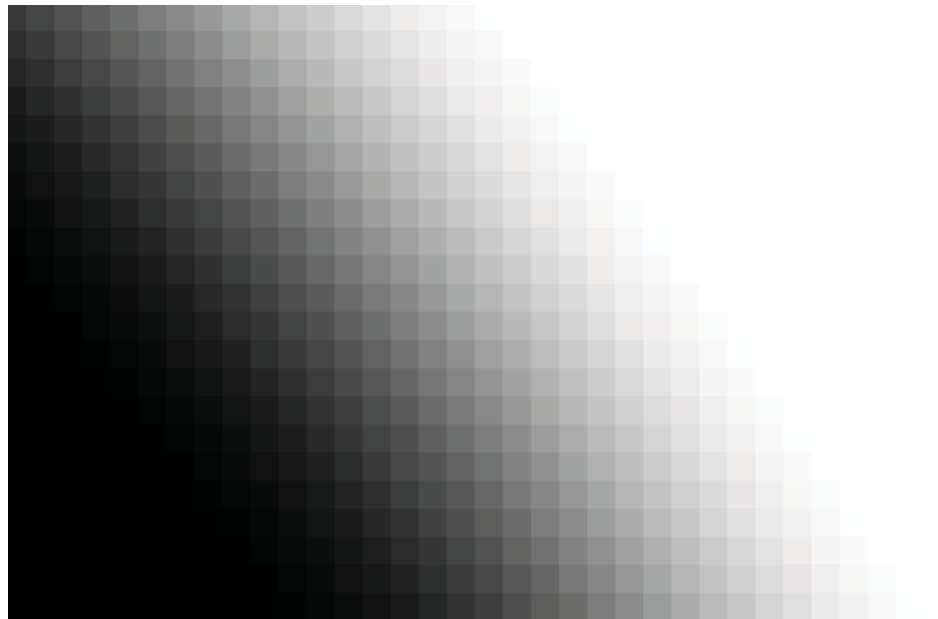
Exemple - step

```
void main() {  
    float d = length(gl_FragCoord.xy);  
    gl_FragColor = vec4(step(200, d));  
}
```



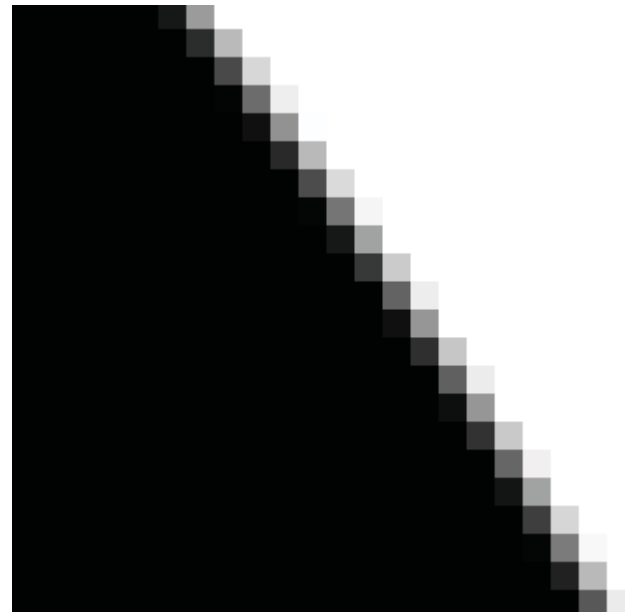
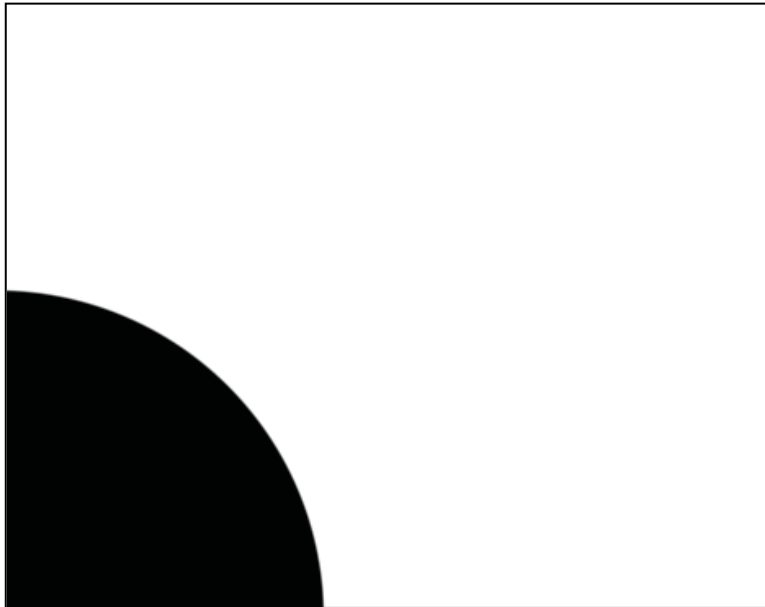
Exemple - step

```
void main() {  
    float d = length(gl_FragCoord.xy);  
    gl_FragColor = vec4(smoothstep(200-10,200+10, d));  
}
```



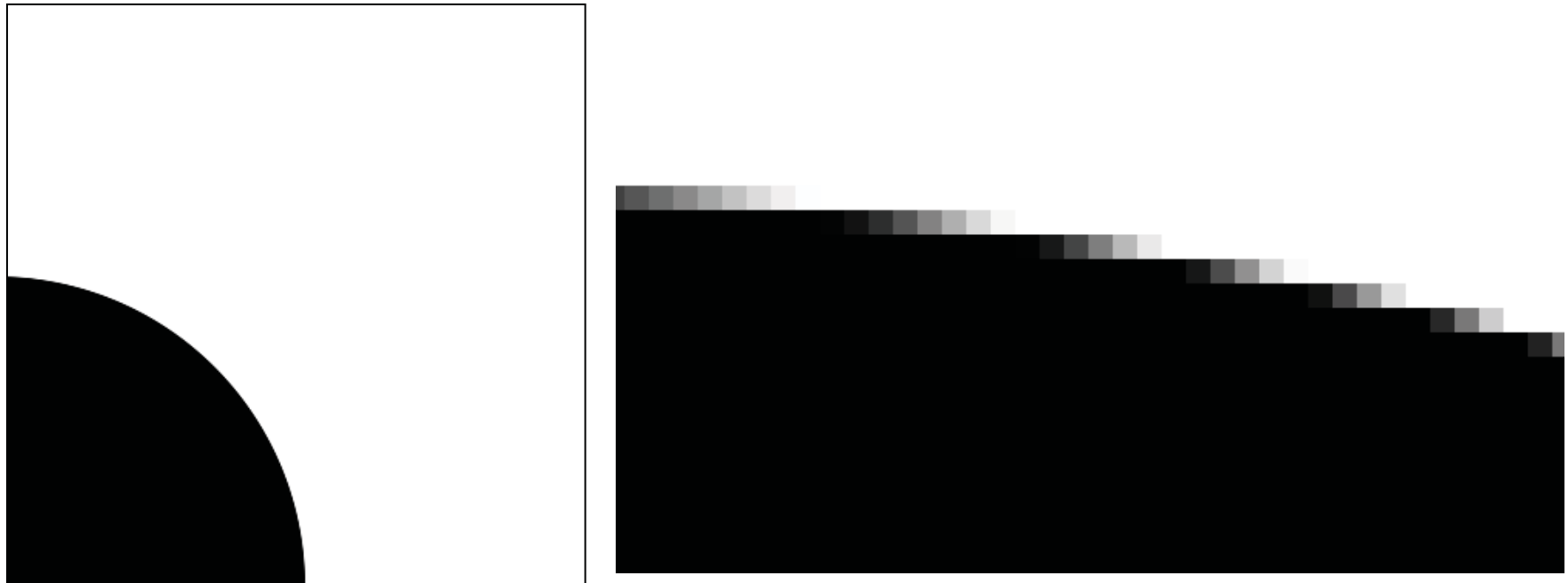
Example - smoothstep

```
void main() {  
    float d = length(gl_FragCoord.xy);  
    gl_FragColor = vec4(smoothstep(200-1,200+1, d));  
}
```



Example - smoothstep

```
void main() {  
    float d = length(gl_FragCoord.xy);  
    gl_FragColor = vec4(smoothstep(200-0.5, 200+0.5, d));  
}
```



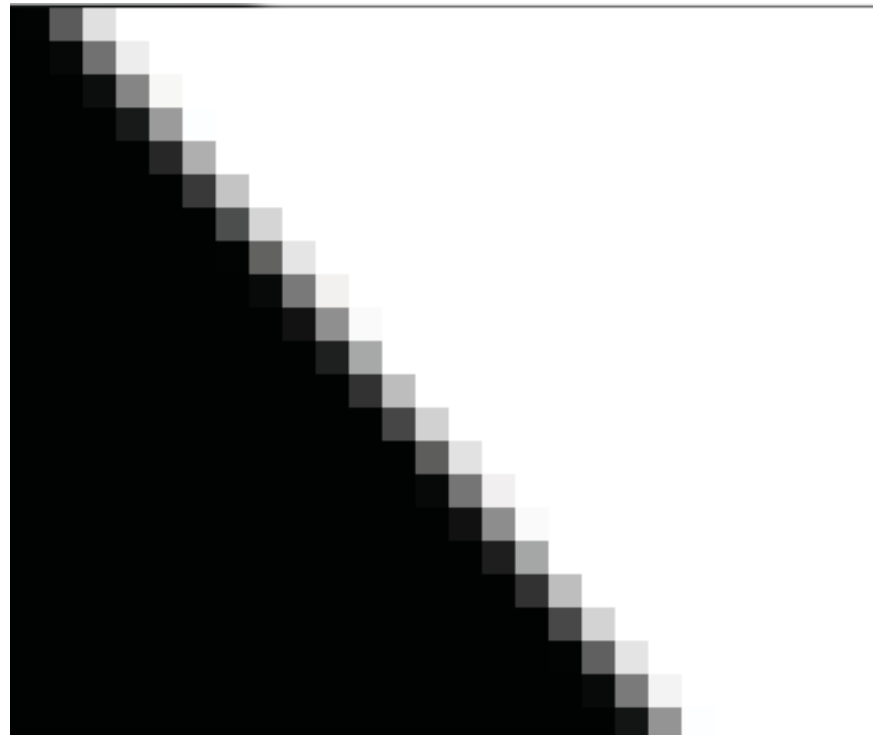
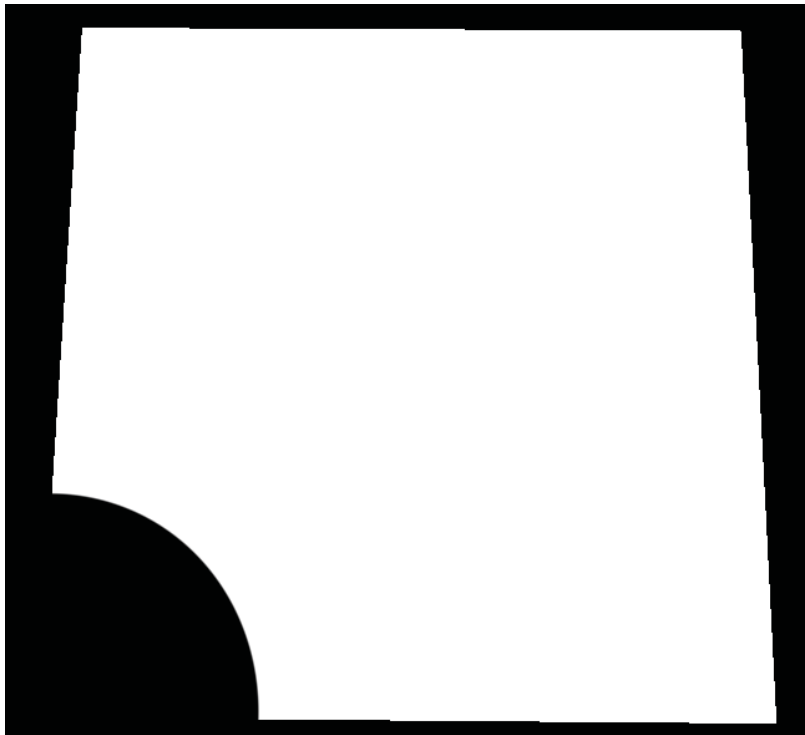
Exemple 2 - smoothstep

```
void main() {  
    float d = length(vtexCoord);  
    const float r = 0.3;  
    gl_FragColor = vec4(smoothstep(r-0.5, r+0.5, d));  
}
```



Exemple 2 – smoothstep + dFdx,dFdy

```
float width = 0.5*length(vec2(dFdx(d), dFdy(d)));  
gl_FragColor=vec4(smoothstep(r-width, r+width, d));
```



aastep (*)

```
float aastep(float threshold, float x)
{
    float width = 0.7*length(vec2(dFdx(x), dFdy(x)));
    return smoothstep(threshold-width, threshold+width, x);
}
```

(*) Patrick Cozzi, Christophe Riccio (Eds.) *OpenGL Insights*, CRC Press, 2012