

# Signed Distance Functions

# Cuprins

Scurtă introducere

Funcții de bază

Derivarea funcției pentru cutii

Exemple și aplicații

# Signed distance functions

- un *SDF* este o *funcție* ce returnează *distanța* de la un punct la o suprafață
- se cheamă *signed* deoarece returnează o distanță  $d$  unde
  - $d > 0$ , când punctul se află în exteriorul suprafeței
  - $d < 0$ , când punctul se află în interiorul suprafeței

## Funcții de bază ale funcțiilor de bază

- lungimea (magnitudinea) unui vector

```
float length(float2 v)
{
    sqrt(v.x * v.x + v.y * v.y);
}
```

- vectorul care pleaca din  $A$  în  $B$  este obținut prin  $B - A$ 
  - ceea ce înseamnă ca putem calcula distanța de la  $A$  la  $B$  prin funcția *length* de mai sus

```
ab = length(B - A);
```

## Funcții de bază

```
float signed_dst_to_circle(float2 p, float2 center, float radius)
{
    return length(p - center) - radius;
}
```

```
float signed_dst_to_box(float2 p, float2 center, float2 size);
```

- alte forme mai mult sau mai puțin complicate precum
  - plan, cilindru, capsulă, con, torus, prismă
  - *SDF*-urile se pot combina între ele pentru a obține forme mai complexe

## Derivarea pentru *unsigned\_dst\_to\_box*

```
float unsigned_dst_to_box(float2 p, float2 center, float2 size)
{
    float2 anchor = abs(p) - (center + size / 2.0);
    float unsigned_dst = length(max(anchor, 0.0));
    return unsigned_dst;
}
```

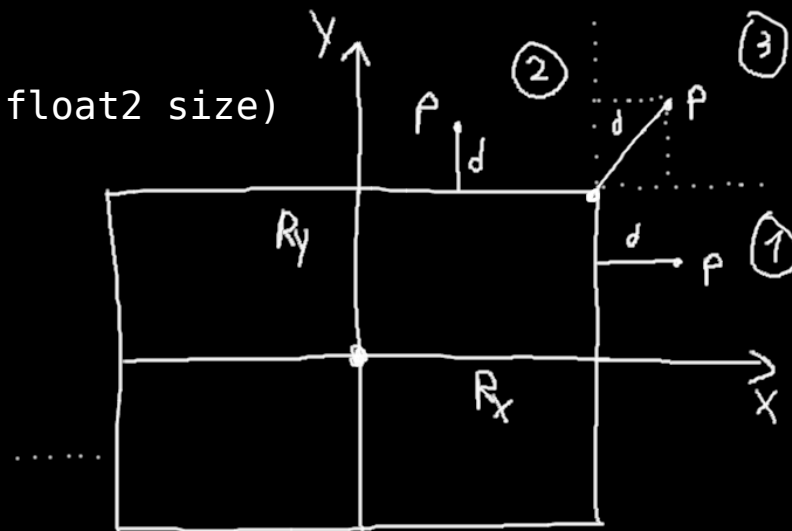
*pe tablă...*

## Derivarea pentru *unsigned\_dst\_to\_box*

```
float unsigned_dst_to_box(float2 p, float2 center, float2 size)
{
    float2 anchor = abs(p) - (center + size / 2.0);
    float unsigned_dst = length(max(anchor, 0.0));
    return unsigned_dst;
}
```

$$d_e = \sqrt{\max(|p_x - R_x|, 0)^2 + \max(|p_y - R_y|, 0)^2}$$

$$d_e = \text{length}(\max(\text{abs}(p) - R, 0))$$



$$\textcircled{1} \quad d = p_x - R_x$$

$$\textcircled{2} \quad d = p_y - R_y$$

$$\textcircled{3} \quad d = \sqrt{(p_x - R_x)^2 + (p_y - R_y)^2}$$

## Derivarea pentru *signed\_dst\_to\_box*

```
float signed_dst_to_box(float2 p, float2 center, float2 size)
{
    float2 anchor = abs(p) - (center + size / 2.0);
    float unsigned_dst = length(max(anchor, 0.0));
    float dst_inside_box = max(min(anchor, 0.0));
    return unsigned_dst + dst_inside_box;
}
```

*pe tablă...*

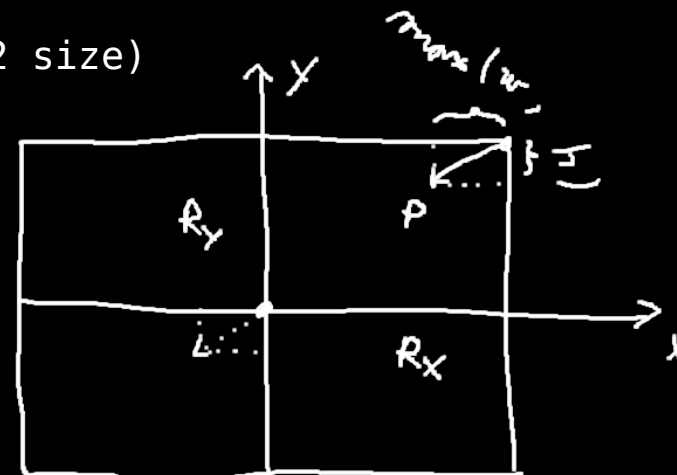


## Derivarea pentru *signed\_dst\_to\_box*

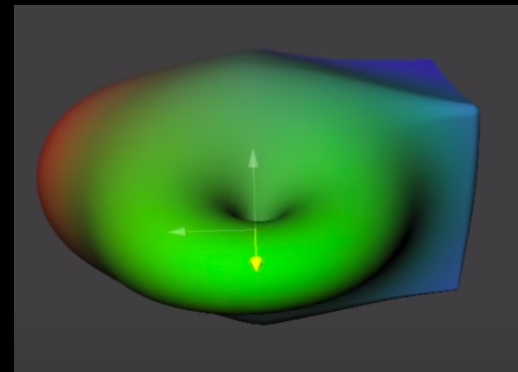
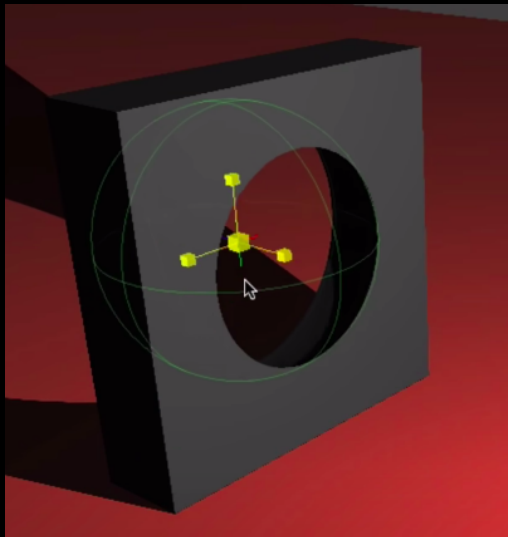
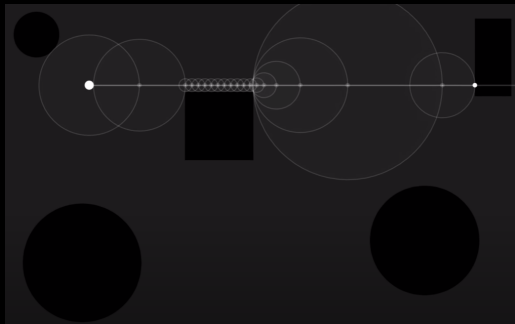
```
float signed_dst_to_box(float2 p, float2 center, float2 size)
{
    float2 anchor = abs(p) - (center + size / 2.0);
    float unsigned_dst = length(max(anchor, 0.0));
    float dst_inside_box = max(min(anchor, 0.0));
    return unsigned_dst + dst_inside_box;
}
```

$$d_i = \max(\min(\text{abs}(P) - R, 0))$$

$$d = d_e + d_i$$



# Exemple



# Aplicații

- Ray Marching
- randare la un DPI înalt a fonturilor folosind GPU pentru jocuri video (implementarea aparține Valve)
- *SDFGI* – model de iluminare globală în timp real folosit de motorul de jocuri video Godot (din versiunea 4.0)
- *GPUI* – framework UI pentru randarea tuturor elementelor pe GPU, în stilul jocurilor video
  - creat pentru a dezvolta un editor text numit *Zed* care țintește să randeze la 120 FPS

Mulțumesc pentru atenție!