# Laborator 11

#### 1 Objective

Această lucrare de laborator evidențiază o tehnică pentru îndepărtarea suprafețelor ascunse utilizând algoritmul z-buffer.

### 2 Fundament teoretic

Determinarea suprafețelor vizibile și ascunse este o problemă fundamentală în grafica asistată de calculator. De obicei, este extrem de ineficient să procesăm obiecte care nu vor apărea niciodată în imaginea finală.

Am văzut deja o tehnică pentru îndepărtarea suprafețelor ascunse în lucrarea de laborator cu numărul 9, și anume *Back Face Culling*, care îndepărtează primitivele aflate în partea din spate a obiectelor opace din pipeline-ul grafic. Algoritmul z-buffer elimină primitive care ar apărea în spatele altor obiecte opace.

Z-buffer-ul, cunoscut și ca buffer de adâncime, este o matrice 2D având aceleași dimensiuni ca framebuffer-ul (imaginea raster finală) care păstrează, pentru fiecare pixel, distanța de la centrul de proiecție la suprafața care a dat culoarea finală a pixelului. Figura 1 prezintă un exemplu al conținutului framebuffer-ului și al z-buffer-ului atunci când se trasează un triunghi roșu paralel cu axa  $\mathbb{Z}$ , situat la o adâncime de z=-5.

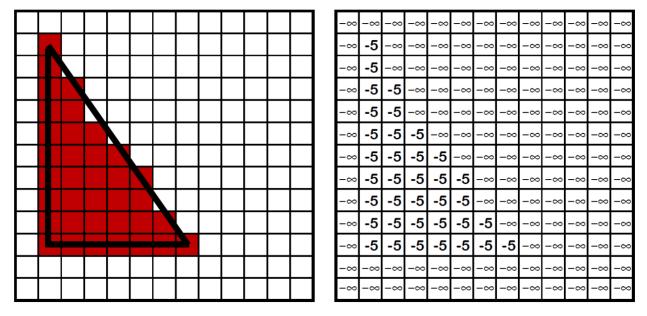


Figure 1 – Frame-buffer și z-buffer pentru desenarea unui triunghi roșu cu z=-5

Când o nouă primitivă este rasterizată, adâncimea fiecăruia dintre pixelii candidați este comparată cu valoarea de adâncime existentă în z-buffer. Pixelul final este actualizat numai dacă valoarea pixelului candidat este mai aproape de centrul de proiecție decât valoarea aflată deja în buffer. Dacă pixelul este actualizat, valoarea adâncimii existente în z-buffer este înlocuită cu adâncimea noului pixel.

De exemplu, la trasarea unui triunghi galben paralel cu axa Z și situat la o adâncime de z=-7, dacă nu există primitive procesate anterior, framebuffer-ul rezultat și z-buffer-ul sunt cele ilustrate în Figura 2.

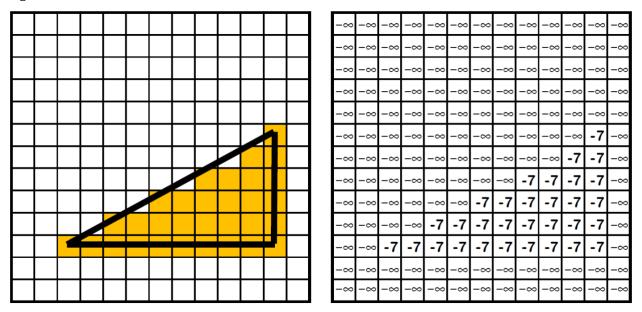


Figure 2 - Frame-buffer și z-buffer pentru desenarea unui triunghi galben z=-7

Dacă triunghiul galben este rasterizat după triunghiul roșu, atunci conținutul framebuffer-ului și al z-buffer-ului sunt ilustrate în Figure 3.

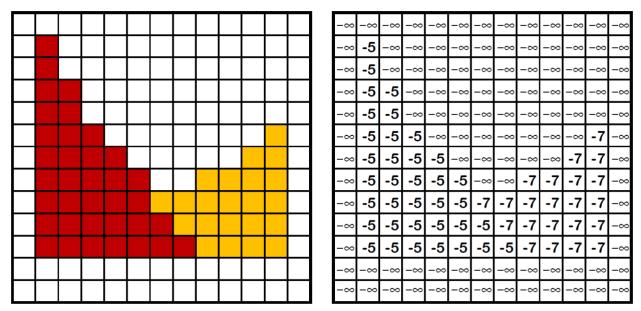


Figure 3 - Frame-buffer și z-buffer pentru desenarea unui triunghi roșu cu  $z=-5\,$  și a unui triunghi galben  $z=-7\,$ 

Cea de-a doua primitivă procesată actualizează z-buffer-ul doar pe pozițiile pixelilor pentru care adâncimea candidatului este mai aproape de centrul proiecției decât adâncimea existentă. Celelalte poziții rămân neschimbate.

## 3 Colorarea iepurelui pe bază de adâncime

Pentru a obține o mai bună înțelegere a importanței îndepărtării suprafețelor ascunse, vom folosi o schemă de colorare bazată pe adâncime pentru iepurele din laboratorul 9. Algoritmul va calcula fiecare culoare de vârf pe baza coordonatei sale inițiale Z (în spațiul obiect) folosind formula:

$$Color_{RGB} = (depthCoeff, depthCoeff, depthCoeff),$$

unde

$$depthCoeff = 255 * (1 - \frac{vertex_z - min_z}{max_z - min_z})$$

 $Vertex_z$  este coordonata Z a vârfului current,  $max_z$  and  $min_z$  sunt valorile Z maxime și minime dintre toate vârfurile iepurelui.

Rezultatul ar trebui să semene cu imaginea din Figure 4.

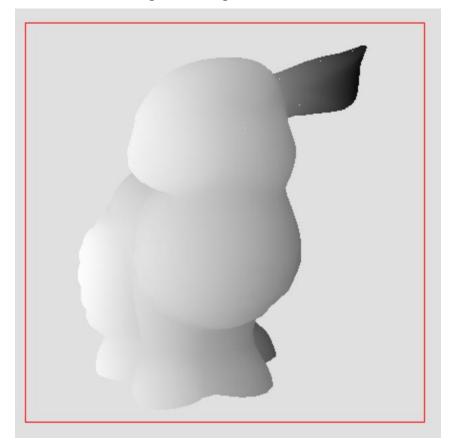


Figure 4 - Colorare pe bază de adâncime

#### 4 Temă

- Aplicați schema de colorare pe bază de adâncime pentru iepurele rasterizat în laboratorul 10.
- Extindeți funcția de rasterizare a triunghiului din laboratorul 9 pentru a include un algoritm de eliminare a suprafețelor ascunse bazat pe z-buffer.

•	Comutați interactiv între rasterizarea cu/fără z-buffer, folosind tastatura. impactul complet al algoritmului, asigurați-vă că opriți Back Face Culling.	Pentru a	a vedea