

# Laborator 7

---

## 1 Obiective

Obiectivul acestui laborator este de a folosi operațiile cu vectori și matrici definite în laboratoarele anterioare pentru a efectua diferite transformări pe o figură 2D.

## 2 Utilizarea aplicației SDL

Pentru exemplificarea grafică a transformărilor 2D, vom folosi o aplicație foarte asemănătoare cu cea din laboratorul 1, bazată pe biblioteca SDL (Simple DirectMedia Layer).

## 3 SDL\_Renderer

**SDL\_Renderer** este o structură de date care gestionează procesul de rasterizare. Aceasta este legată de un **SDL\_Window**, astfel încât rasterizarea se poate face doar în acel **SDL\_Window**. De asemenea, un obiect de tip **SDL\_Renderer** poate să țină evidența setărilor de rasterizare.

Pentru a crea un renderer legat de fereastra aplicației, vom folosi următorul cod:

```
SDL_Renderer *windowRenderer;  
  
windowRenderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED);
```

Există câteva funcții importante asociate obiectelor de tip **SDL\_Renderer**:

1. **SDL\_SetRenderDrawColor** stabilește culoarea care va fi utilizată în toate operațiile de rasterizare, până când se efectuează un alt apel la funcție.

```
SDL_SetRenderDrawColor(renderer, r, g, b, a);
```

2. **SDL\_RenderClear** șterge întreaga zonă de afișare a ferestrei utilizând culoarea activă curentă, setată anterior folosind funcția **SDL\_SetRenderDrawColor**.

```
SDL_RenderClear(renderer);
```

3. **SDL\_RenderPresent** va afișa întregul conținut rasterizat pe suprafața ecranului. Până când această funcție este apelată, procesul de rasterizare are loc într-un buffer ascuns care nu este vizibil pentru utilizator. Apelul **SDL\_RenderPresent** trebuie efectuat o singură dată, după ce toate funcțiile de rasterizare au fost apelate.

```
SDL_RenderPresent(renderer);
```

## 4 Trasarea unei linii

Trasarea unei linii utilizând SDL Renderer necesită următorii pași:

1. Definiți culoarea care urmează să fie folosită (exemplu: albastru):

```
SDL_SetRenderDrawColor(renderer, 0, 0, 255, 255);
```

2. Definiți punctele de început și de sfârșit ale liniei. Pentru aceasta vom folosi variabilele vec3:

```
vec3 P1(100, 100, 1), P2(400, 100, 1);
```

3. Desenați linia folosind obiectul renderer:

```
SDL_RenderDrawLine(windowRenderer, P1.x, P1.y, P2.x, P2.y);
```

4. Afișați conținutul obiectului renderer pe ecran:

```
SDL_RenderPresent(renderer);
```

## 5 Temă

Descărcați și rulați aplicația de pe site-ul web al laboratorului. Încercați să înțelegeți exemplul de bază și apoi extindeți aplicația cu următoarele funcționalități:

- Includeți în aplicație fișierele de implementare (.cpp).
- Definiți un dreptunghi initial având colțul din stânga-sus în  $P_1(100, 100)$  și colțul din dreapta jos în  $P_2(400, 200)$ .
- Rotiți dreptunghiul în jurul centrului său (intersecția diagonalelor) cu 10 grade în sensul acelor de ceasornic atunci când este apăsat `RIGHT_ARROW` și 10 grade în sens invers acelor de ceasornic atunci când este apăsat `LEFT_ARROW`.
- Scalați dreptunghiul având colțul din stânga sus ca referință, utilizând tastele `UP_ARROW` și `DOWN_ARROW`.