

# Laborator 1

---

## 1 Obiective

Obiectivul acestui laborator este de a descrie pe scurt biblioteca SDL și de a exemplifica dezvoltarea unei aplicații bazate pe aceasta.

## 2 Crearea unei ferestre folosind SDL

Biblioteca SDL (Simple DirectMedia Layer) gestionează accesul la hardware-ul grafic (prin biblioteci OpenGL și Direct3D), precum și audio, tastatură și mouse (independent de sistemul de operare folosit). Această bibliotecă suportă mai multe sisteme de operare precum Windows, Mac OS X și Linux. Pe parcursul acestui laborator vom folosi limbajul de programare C++ pentru dezvoltarea aplicațiilor, dar există implementări SDL disponibile și pentru alte limbaje (cum ar fi C # sau Python).

Pentru a crea o fereastră SDL sunt necesari următorii pași:

1. Inițializați biblioteca SDL apelând funcția [SDL\\_Init\(\)](#) cu argumentul [SDL\\_INIT\\_VIDEO](#), deoarece folosim numai subsistemul video al SDL.

```
SDL_Init(SDL_INIT_VIDEO);
```

2. Creați o fereastră utilizând funcția [SDL\\_CreateWindow\(\)](#). Argumentele sunt titlul ferestrei, poziția, lățimea, înălțimea și unele flag-uri (reprezentând opțiuni - de exemplu, pentru a crea o fereastră pe tot ecranul sau o fereastră redimensionabilă).

```
SDL_CreateWindow("SDL Hello World Example", SDL_WINDOWPOS_UNDEFINED,  
SDL_WINDOWPOS_UNDEFINED, WINDOW_WIDTH, WINDOW_HEIGHT, SDL_WINDOW_SHOWN |  
SDL_WINDOW_ALLOW_HIGHDPI);
```

Înainte de a închide aplicația, trebuie să dealocăm toate resursele create:

1. DISTRUGEȚI fereastra apelând funcția [SDL\\_DestroyWindow\(\)](#) și oferind ca argument pointerul către fereastră.

```
SDL_DestroyWindow(window);
```

2. Apelați funcția [SDL\\_Quit\(\)](#) care este responsabilă pentru curățarea tuturor subsistemelor inițializate (pentru exemplul nostru folosim doar subsistemul video).

```
SDL_Quit();
```

### 3 SDL Surface

Pentru a desena ceva pe ecran avem nevoie de un obiect “canvas” (pânză). În SDL, un “canvas” poate fi reprezentat de **SDL\_Surface** sau **SDL\_Texture**. **SDL\_Surface** este folosit în redarea de tip software, în schimb **SDL\_Texture** este utilizat în redarea de tip hardware, diferența principală fiind localizarea buffer-elor de date (pixeli). Pe parcursul acestui laborator vom folosi **SDL\_Surface**, deoarece vom încerca să implementăm pipeline-ul grafic de la zero.

În SDL, o suprafață este o structură de date care conține o colecție de pixeli. Fiecare fereastră are atașată o astfel de suprafață și, în plus, putem crea noi suprafețe și aplica unele operații pe acestea (de exemplu, putem încărca o imagine pe o suprafață și copia conținutul acesteia în fereastră). [SDL Surface](#) stochează formatul pixelilor, dimensiunea (lățimea și înălțimea), un pointer la datele efective ale pixelilor și alte informații relevante. Vom folosi o reprezentare a pixelilor pe 32 de biți. În această situație stocăm 1 octet per canal (roșu, verde, albastru și alfa).

Dacă dorim să desenăm un dreptunghi, trebuie să specificăm poziția de pornire a dreptunghiului (coordonatele x și y ale colțului stânga-sus) și dimensiunea (lățimea și înălțimea). Pentru a specifica o culoare folosim funcția **SDL\_MapRGB()** utilizând formatul pixel al suprafeței și valorile de culoare roșu, verde și albastru. Funcția **SDL\_FillRect()** va actualiza suprafața dreptunghiului folosind culoarea specificată.

```
SDL_Rect rectangleCoordinates = {100, 100, 200, 200};  
Uint32 rectagleColor = SDL_MapRGB(windowSurface->format, 255, 0, 0);  
SDL_FillRect(windowSurface, &rectangleCoordinates, rectagleColor);
```

Pentru fiecare canal de culoare specificăm valorile cuprinse între 0 și 255. În tabelul următor se regăsesc câteva exemple de culori care se pot obține folosind diferite combinații de valori ale celor trei canale.

Nume	Canal Roșu	Canal Verde	Canal Albastru	Culoare
Alb	255	255	255	
Roșu	255	0	0	
Verde	0	255	0	
Albastru	0	0	255	
Galben	255	255	0	
Negru	0	0	0	

### 4 Procesarea evenimentelor

În SDL evenimentele ar putea fi lucruri de genul apăsării unei taste de tastatură sau mișcarea mouse-ului. Toate evenimentele sunt stocate într-o coadă în ordinea în care au apărut. Folosind funcția [SDL\\_WaitEvent\(\)](#) primim următorul eveniment din coadă.

## 4.1 Evenimente de apăsare a mouse-ului

Putem verifica dacă butonul stânga al mouse-ului a fost apăsat folosind următorul fragment de cod. Mai întâi verificăm ca tipul de eveniment să fie **SDL\_MOUSEBUTTONDOWN** și apoi ca butonul apăsat să fie **SDL\_BUTTON\_LEFT**. Funcția **SDL\_GetMouseState()** primește coordonatele poziției curente a mouse-ului (x și y).

```
if(currentEvent.type == SDL_MOUSEBUTTONDOWN)
{
    if(currentEvent.button.button == SDL_BUTTON_LEFT)
    {
        SDL_GetMouseState(&mouseX, &mouseY);
    }
}
```

## 4.2 Evenimente de mișcare a mouse-ului

Putem verifica dacă butonul stânga al mouse-ului este apăsat în timp ce mouse-ul se mișcă utilizând următorul fragment de cod. Mai întâi verificăm ca tipul evenimentului să fie **SDL\_MOUSEMOTION** și apoi ca butonul apăsat pentru să fie **SDL\_BUTTON\_LEFT**.

```
if(currentEvent.type == SDL_MOUSEMOTION)
{
    if(currentEvent.button.button == SDL_BUTTON_LEFT)
    {
        SDL_GetMouseState(&mouseX, &mouseY);
    }
}
```

## 4.3 Evenimente generate de tastatură

Putem obține tasta apăsată utilizând următorul fragment de cod. Mai întâi verificăm ca tipul de eveniment să fie **SDL\_KEYDOWN** și apoi procesăm tastele dorite.

```
if(currentEvent.type == SDL_KEYDOWN)
{
    switch(currentEvent.key.keysym.sym)
    {
        case SDLK_UP:
            //process UP key
            break;

        case SDLK_R:
            //process R key
            break;

        ...

        default:
            //process other keys
    }
}
```

```
//default process  
  
break;  
  
}
```

## 5 Lectură suplimentară

- Instalarea SDL pe diferite platforme – [http://lazyfoo.net/SDL\\_tutorials/lesson01/index.php](http://lazyfoo.net/SDL_tutorials/lesson01/index.php)
- Tutoriale SDL - [http://lazyfoo.net/SDL\\_tutorials/](http://lazyfoo.net/SDL_tutorials/)

## 6 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:

- Schimbați culoarea dreptunghiului prin setarea canalelor de culoare folosind tastele R (pentru roșu), G (pentru verde) și B (pentru albastru) și modificând valoarea canalului prin apăsarea tastelor UP și DOWN.
- Afișați interactiv dreptunghiul. Primul set de coordonate (colțul stanga-sus) este preluat la evenimentul de apăsare al butonului stânga al mouse-ului și al doilea set de valori (lățimea și înălțimea dreptunghiului) este calculat la fiecare eveniment de mișcare a mouse-ului (dar numai dacă butonul din stânga este încă apăsat – desenare dreptunghi cu Drag and Drop, similară celei din utilitarul Microsoft Paint).