

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 și un renderer utilizând funcția [SDL_CreateWindowAndRenderer\(\)](#). Argumentele sunt titlul ferestrei, lățimea, înălțimea, unele flag-uri (reprezentând opțiuni - de exemplu, pentru a crea o fereastră pe tot ecranul sau o fereastră redimensionabilă) și obiectele în care vor fi stocate referințele către fereastra creată și renderer.

```
SDL_CreateWindowAndRenderer("SDL Hello World Example", WINDOW_WIDTH,  
WINDOW_HEIGHT, SDL_WINDOW_HIGH_PIXEL_DENSITY, &window, &renderer);
```

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

1. DISTRUGEȚI renderer-ul apelând funcția [SDL_DestroyRenderer\(\)](#) și oferind ca argument pointerul către renderer

```
SDL_DestroyRenderer(renderer);
```

2. DISTRUGEȚI fereastra apelând funcția [SDL_DestroyWindow\(\)](#) și oferind ca argument pointerul către fereastră.

```
SDL_DestroyWindow(window);
```

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

```
SDL_Quit();
```

3 SDL Renderer

SDL_Renderer este o structură care gestionează procesul de rasterizare, legată de un **SDL_Window**. Un obiect de tip **SDL_Renderer** poate să țină evidența setărilor de rasterizare.

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);
```

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 structura **SDL_Color** utilizând valorile de culoare roșu, verde, albastru și alfa. Funcția **SDL_RenderFillRect()** va desena dreptunghiului folosind culoarea setată anterior.

```
SDL_FRect rectangleCoordinates = {100, 100, 200, 200};
SDL_Color rectagleColor { 255, 0, 0, 255 };
SDL_SetRenderDrawColor(renderer,
    rectagleColor.r, rectagleColor.g, rectagleColor.b, rectagleColor.a);
SDL_RenderFillRect(renderer, &rectangleCoordinates);
```

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_PollEvent\(\)](#) se verifică dacă există și se procesează următorul eveniment din coadă.

4.1 Evenimente de apăsare a mouse-ului

Putem verifica dacă butonul stânga al mouse-ului a fost apăsător folosind următorul fragment de cod. Mai întâi verificăm ca tipul de eveniment să fie **SDL_EVENT_MOUSE_BUTTON_DOWN** și apoi ca butonul apăsător să fie **SDL_BUTTON_LEFT**. Funcția **SDL_GetMouseState()** preia coordonatele poziției curente a mouse-ului (x și y).

```
if(currentEvent.type == SDL_EVENT_MOUSE_BUTTON_DOWN)
    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 apăsător în timp ce mouse-ul se mișcă utilizând următorul fragment de cod. Mai întâi verificăm ca tipul evenimentului să fie **SDL_EVENT_MOUSE_MOTION** și apoi ca butonul apăsător pentru să fie **SDL_BUTTON_LEFT**.

```
if(currentEvent.type == SDL_EVENT_MOUSE_MOTION)
{
    SDL_MouseButtonFlags mouseButtons = SDL_GetMouseState(nullptr, nullptr);
    if( mouseButtons & SDL_BUTTON_MASK(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_EVENT_KEY_DOWN** și apoi procesăm tastele dorite.

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

        case SDLK_R:
            //process R key
            break;
        ...
        default:
            //default process
            break;
    }
```

5 Lectură suplimentară

- Instalarea SDL3 pe diferite platforme – <https://wiki.libsdl.org/SDL3/Tutorials/FrontPage>
- Tutoriale SDL3 - <https://lazyfoo.net/tutorials/SDL3/index.php>

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).