



Universitatea Tehnică „Gheorghe Asachi” Iași
Facultatea de Automatică și Calculatoare Iași
Specializarea: Tehnologia Informației
Disciplina: Proiectarea interfețelor utilizator



EVENTIA

Coordonator,

Prof. Marius Gavrilescu

Studenti,

Bahnaru Alexandru Georgian, Grupa 1409A

Cobzariu Ștefan Alexandru, Grupa 1409A

Postolache Malina, Grupa 1409A

Iași, 2024

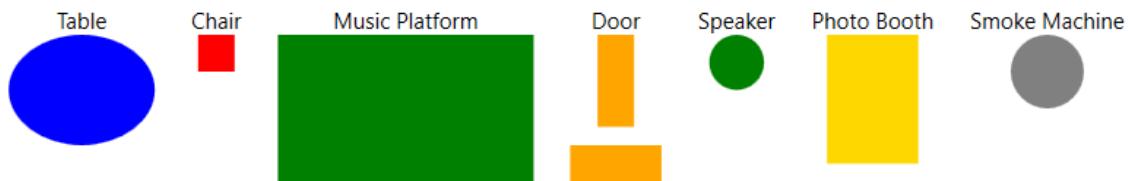
1. Obiectivele Projectului

Proiectul se concentrează pe dezvoltarea unei aplicații grafice interactive care să permită utilizatorilor să creeze, manipuleze și salveze layout-uri vizuale pe un Canvas. Aplicația include funcționalități de drag-and-drop pentru obiecte geometrice, aliniere automată a acestora, precum și operații de manipulare a obiectelor prin intermediul butoanelor.

2. Prezentarea Aplicației

Mălina- Funcționalități legate de Drag & Drop și Crearea Formelor Pe Canvas

Am implementat funcționalitatea de **drag & drop**, care permite utilizatorilor să selecteze și să mute obiecte geometrice predefinite (masă, scaun, platformă muzicală, ușă și difuzor) pe suprafața unui **Canvas**. Această funcționalitate este esențială pentru personalizarea și organizarea elementelor într-un spațiu vizual interactiv. Fiecare formă este reprezentată grafic prin elemente **Rectangle** sau **Ellipse**, iar sistemul permite adăugarea lor sub formă de copii.

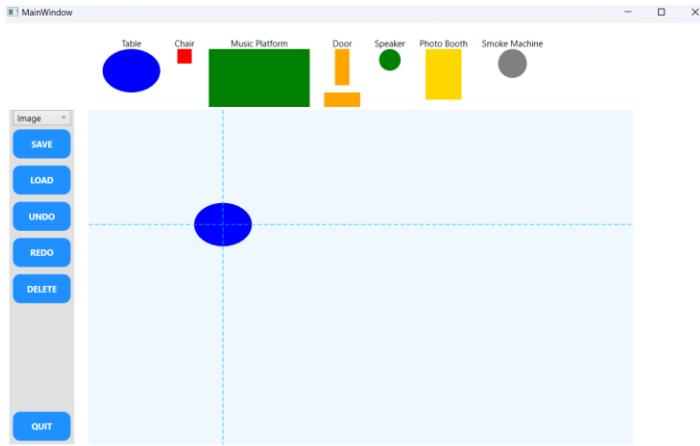


1. Crearea formelor prin copiere

Când utilizatorul dă click pe o formă din meniul de sus, aplicația creează o copie a acesteia și o plasează pe Canvas.

Poziția inițială este calculată în funcție de poziția mouse-ului utilizatorului în momentul selecției.

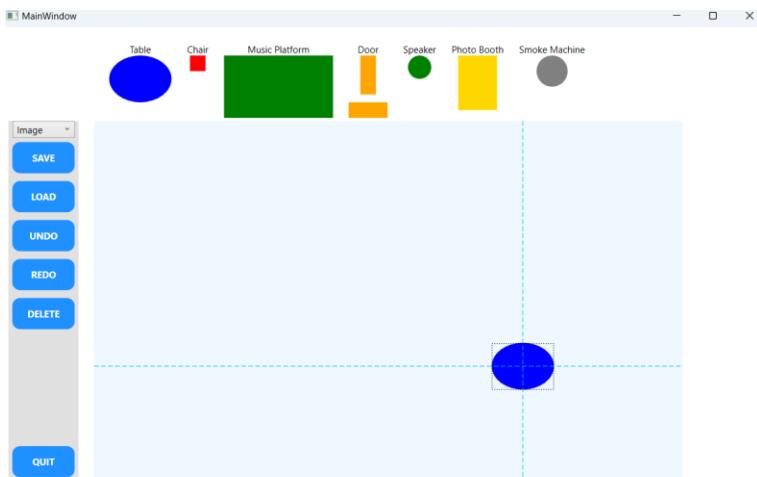
Toate copiile sunt salvate și gestionate eficient în cadrul aplicației, iar ele pot fi manipulate independent de forma originală.



2. Interacțiunea și mutarea formelor

După ce copia este creată, utilizatorul poate să mute forma cu mouse-ul. Mișcarea este restricționată astfel încât obiectul să nu poată ieși din zona definită de **Canvas**.

Pozitia obiectului este actualizată continuu în timpul mișcării pentru a reflecta coordonatele mouse-ului.



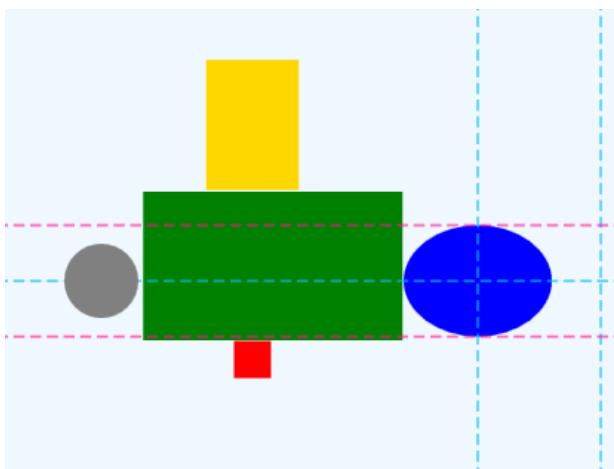
3. Finalizarea mutării formelor

Eliberarea mouse-ului finalizează procesul de mutare a obiectului.

4. Restricții de poziționare

Obiectele nu pot depăși marginile Canvasului. Poziționarea este limitată automat, iar forma este ajustată să rămână în interiorul acestuia.

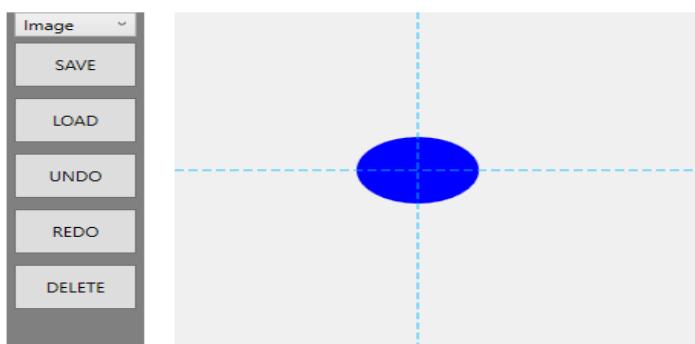
O altă restricție importantă este aceea că obiectele nu pot fi plasate unele peste altele. Fiecare element trebuie să aibă propriul spațiu delimitat.



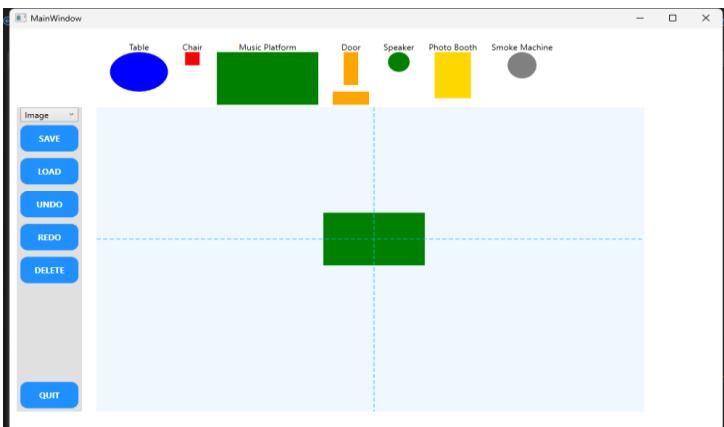
Ştefan – funcționalitățile legate de partea de aliniere a obiectelor și a liniilor de ghidaj pentru plasarea obiectelor pe suprafața Canvas.

Initial când un obiect este selectat sunt active două linii ajutătoare: o linie verticală și una orizontală ce permit plasarea formei pe suprafața Canvas și mutarea ei într-o manieră ordonată.

Plasarea unui obiect în suprafața Canvas și activarea liniilor de ghidaj:



Am restricționat suprafața în care poate fi mutată scena pe mijlocul Canvasului pentru a putea fi plasată în centrul sălii.



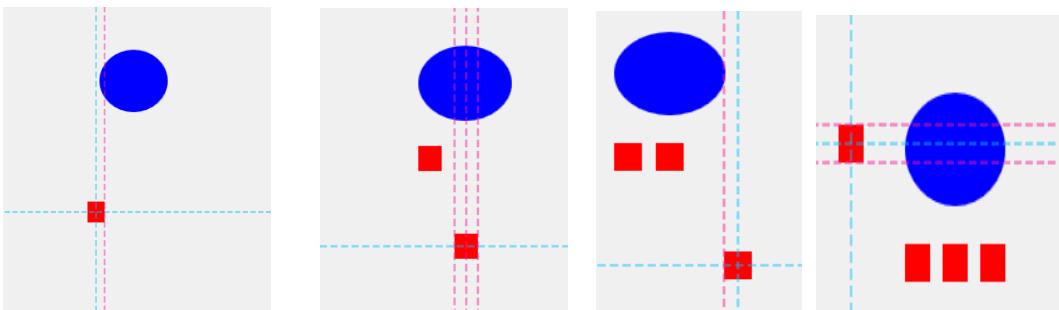
Utilizare:

Alinierea se face în momentul în care o linie de ghid de pe verticală sau orizontală detectează intersecția cu un obiect de tip masă. În funcție de poziția în care se află obiectul curent față de cel intersectat, alinierea se face în funcție de linia care intersectează obiectul și prin apasarea butonului “A” de pe tastatură.

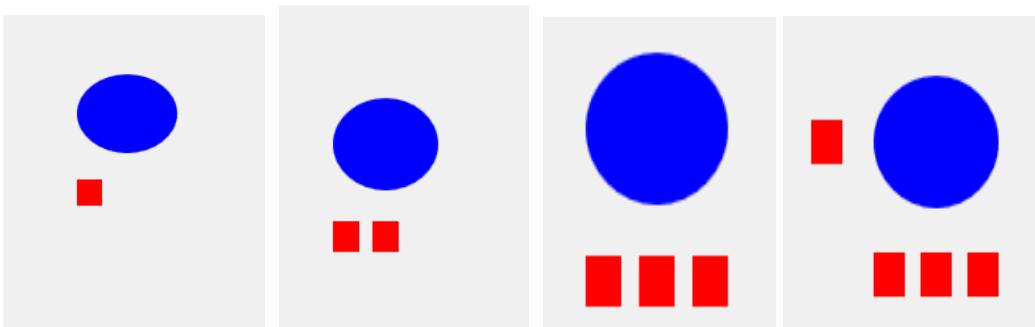
Atât pe verticală, cât și pe orizontală se află câte 3 linii de ghid: cele din mijloc sunt active tot timpul, iar cele de pe extremități doar în timpul intersecției cu un obiect. Prin detectarea intersecției cu un obiect, culoarea liniilor se modifică în roșu, iar prin dispariția intersecției, liniile de pe extremități sunt ascunse, iar cele din mijloc revin la culoarea albastru.

1. Aranjarea scaunelor la masa

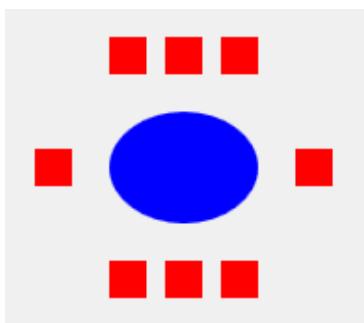
Cazurile de intersecție și alinierea: 1.



2. Efect după aliniere (Apăsarea tastei “A”)

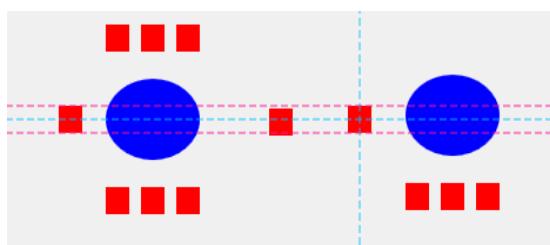


3.Alinierea tuturor scaunelor:

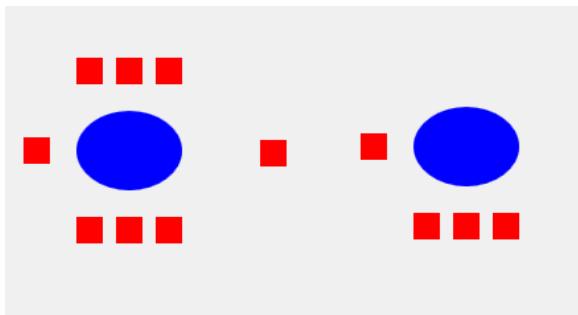


Cazul în care un obiect de tip scaun intersectează mai multe obiecte de tip masa

1.



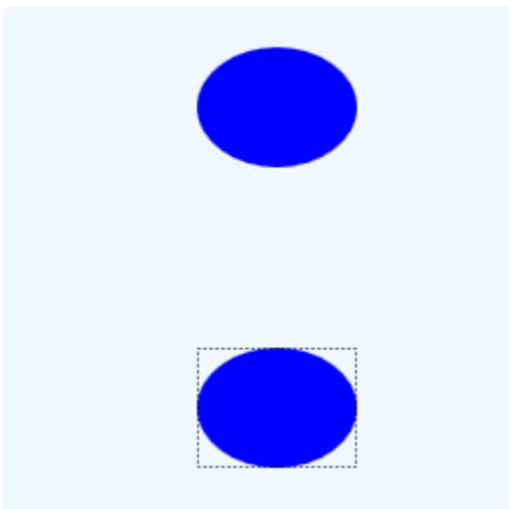
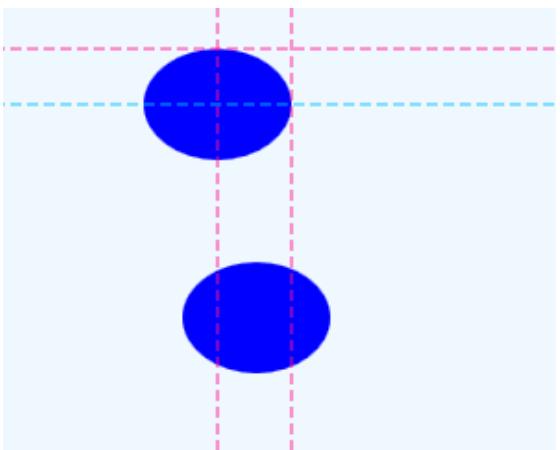
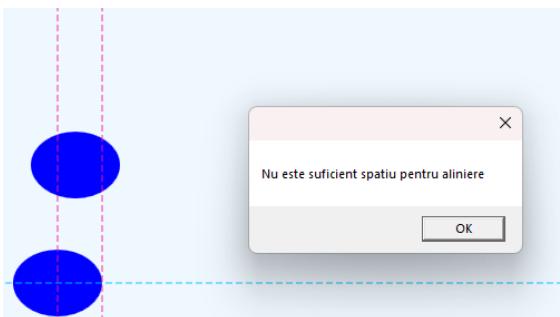
2.Efect după aliniere



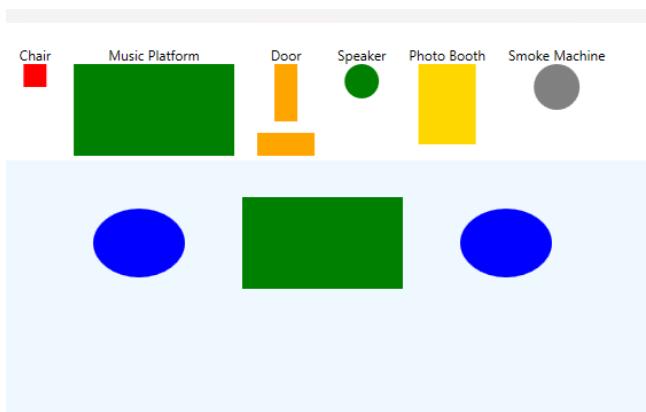
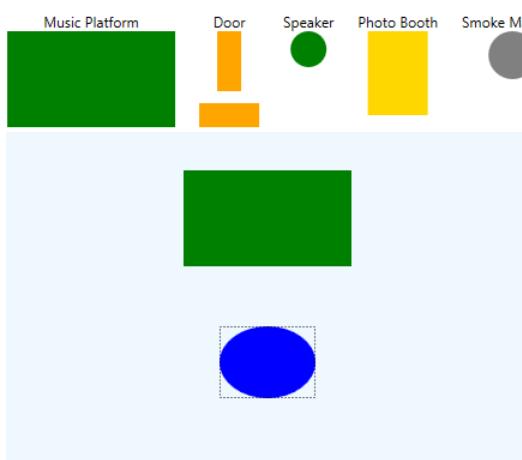
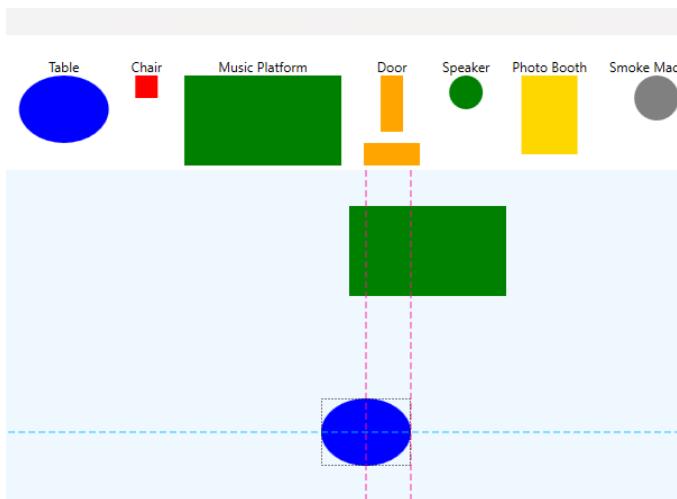
Pentru fiecare caz corespondent unei linii de ghid se returnează o listă cu obiectele intersectate și se alege obiectul cu distanța cea mai mică față de obiectul curent. În acest caz, obiectul curent va fi aliniat la obiectul intersectat cu distanța cea mai mică față de el.

2.Alinierea meselor

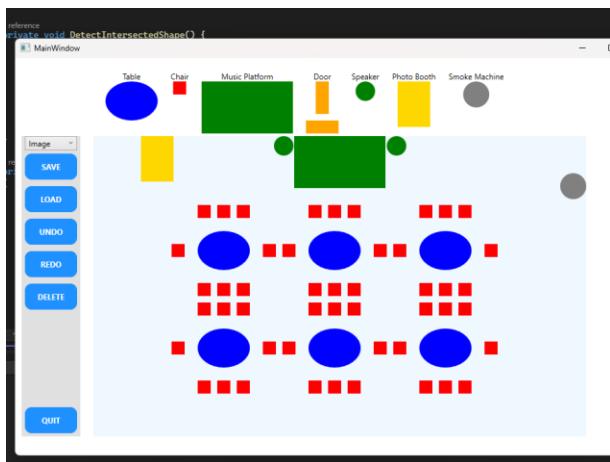
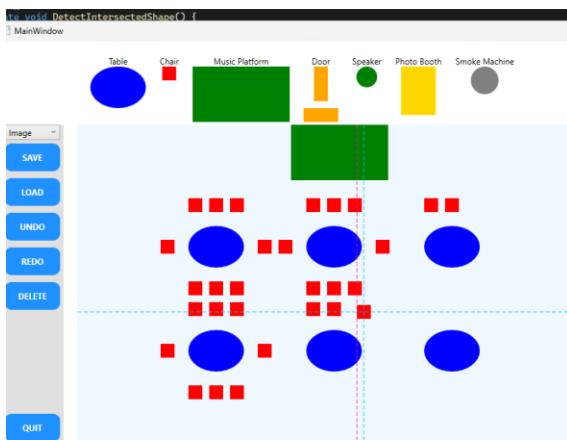
1.Cazul când nu există suficient spațiu pentru a adauga o altă masă. Între cele două mese trebuie să existe o distanță minimă ce permite plasarea a 2 scaune.



3. Alinierea mesei în jurul scenei



Plasarea și aranjarea tuturor obiectelor în sală



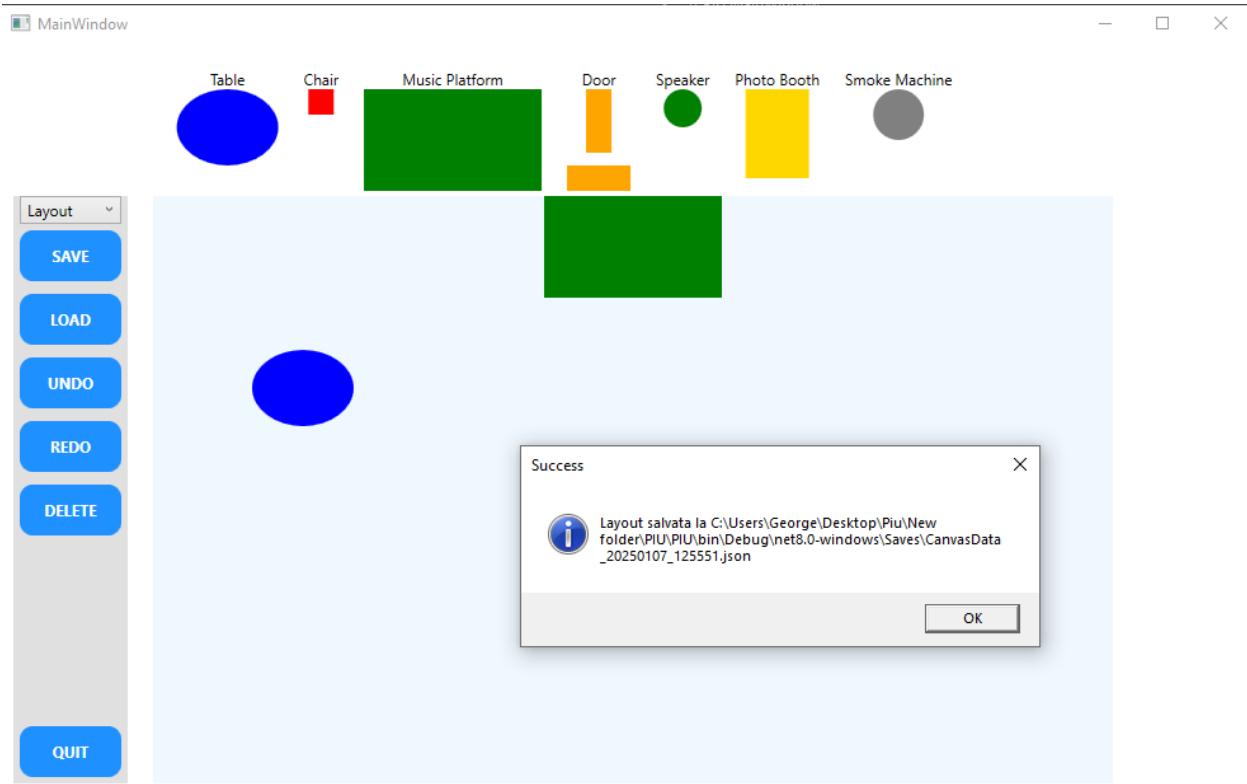
George – functionalitatele butoanelor

Butonul SAVE

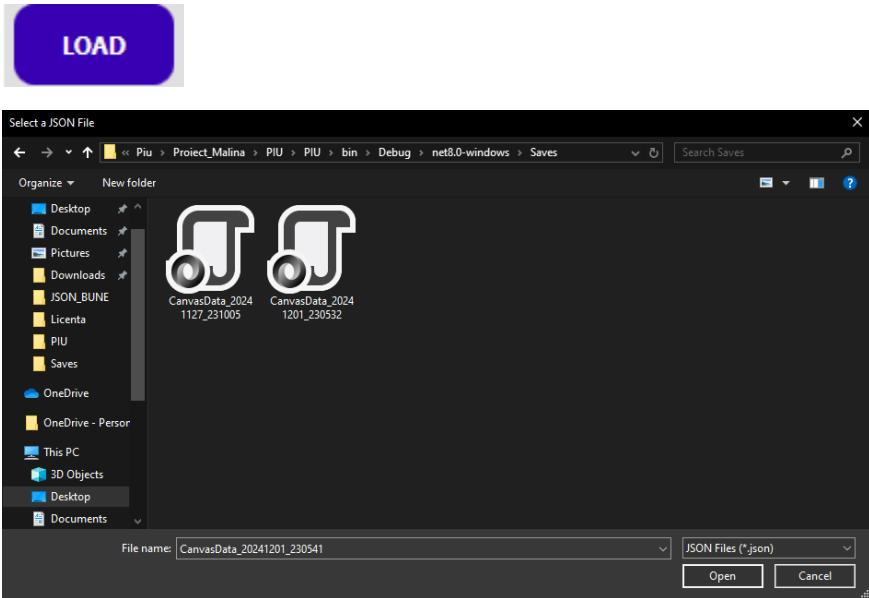


-butonul SAVE salvează layout-ul creat de utilizator în 2 modalități, imagine (salvează layout-ul creat ca PNG în fișierul save din bin-Debug-net8.0windows-Saves) sau salvează layout-ul ca un fișier JSON care conține toate obiectele create și proprietățile acestora precum – poziția, culoarea etc

-acestea sunt denumite în funcție de timestamp-ul la care au fost create pentru a fi ușor de recunoscut



Butonul LOAD



-initializează canvas-ul cu obiectele salvate într-un fișier de tip JSON pentru a putea continua editarea dintr un anumit punct

Butonul UNDO



-actionarea acestui buton duce la revenirea la acțiunea anterioara – chiar dacă acțiunea este de delete, create sau move.

Butonul REDO



-actionarea acestui buton duce la refacerea acțiunii anterioare– chiar dacă acțiunea este de delete, create sau move.

Butonul DELETE



-actionarea acestui buton duce la ștergerea ultimului obiect apăsat.

Butonul QUIT



- actionarea acestui buton duce la inchiderea aplicatiei.

Functionalitatea de undo/redo

Aceasta funcționalitate a fost realizată folosind elementele principale:

- Clasa Action.cs, care este folosită pentru a salva datele ultimei acțiuni realizate în interiorul canvasului
- Stiva _redoActions, care este folosită pentru a salva toate acțiunile care pot fi realizate în urma apăsării butonului Redo
- Stiva _historyActions, care este folosită pentru a salva toate acțiunile care pot fi realizate în urma apăsării butonului Undo

Acțiunile realizate pot fi de mai multe tipuri: creare, ștergere, mutare, iar în cazul în care este încărcat un layout, acțiunile de creare a obiectelor vor fi „undoable”.

Pentru a putea gestiona corect acțiunile, clasa Action conține următoarele proprietăți:

- _shape - forma obiectului

- `_point` - coordonatele obiectului
- `_creation` - flag pentru crearea unui obiect
- `_deletion` - flag pentru ștergerea unui obiect
- `_undoable` - flag pentru o acțiune undoable

Astfel, în momentul în care un obiect este creat, șters sau mutat, un obiect de tip Action va fi creat și adăugat în `_historyActions`, cu flagurile specifice setate pe true, acesta fiind eligibil pentru acțiunea de undo. În momentul în care functionalitatea de undo este utilizată acest obiect de tip Action va fi mutat în `_redoActions`, devenind o acțiune care poate fi executată doar prin apăsarea butonului redo.

Load/Save

În cadrul proiectului, procesul de salvare și încărcare a unui canvas se realizează prin utilizarea unui fișier JSON care are rolul de a salva informațiile tuturor obiectelor care se află în interiorul canvasului, precum poziție, culoare.

Clasa ShapeData

Proprietățile clasei:

- `ShapeType` - tipul de forma
- `Width` - lățimea obiectului
- `Height` - înălțimea obiectului
- `Left` - valoarea proprietății Left specifică Canvasului
- `Top` - valoarea proprietății Top specifică Canvasului
- `FillColor` - valoarea culorii obiectului

Load

Procesul de load se realizează în felul următor :

- este selectat fișierul JSON din care se dorește încărcarea layout-ului
- toate obiectele din acest fișier vor fi convertite într-o listă de obiecte de tipul `ShapeData`, reprezentând toate obiectele care trebuie create în interiorul canvasului, folosind `JsonSerializer.Deserialize`
- după crearea acestor obiecte, le vor fi atribuite evenetele specifice și liniile pentru acțiunea de tip `SnapTo`

Save

Procesul de salvare se realizează în felul următor:

- ca imagine
- tot conținutul canvasului va fi salvat într-un fișier de tip png, cu numele format din data și ora la care a fost salvat, pentru a fi ușor de găsit.
- ca layout
- invers procesului de load, pentru toate obiectele din interiorul canvasului vor fi create obiecte de tipul `ShapeData` care vor fi adăugate într-o listă

-folosind JsonSerializer.Serialize, lista de obiecte va fi salvată într-un fișier de tip json

3. Tehnologii Utilizate

Aplicația este dezvoltată utilizând următoarele tehnologii:

- **C#** – limbaj de programare principal.
- **WPF (Windows Presentation Foundation)** – folosit pentru construirea interfeței grafice.
- **.NET Framework** – platforma pe care rulează aplicația.
- **XAML** – pentru definirea interfeței grafice.

4.Biblioteci necesare:

- Aplicația utilizează doar bibliotecile standard incluse în **WPF**, astfel nu necesită alte pachete externe.

5. Versiuni și Tool-uri Folate

- **Visual Studio 2022**
- **.NET 8.0 SDK**
- **Windows 10/11**