



**COLEGIUL NAȚIONAL “UNIREA” BRAȘOV**

**LUCRARE PENTRU ATESTAREA  
COMPETENȚELOR PROFESIONALE**

# **Grafuri -Graphy-**

**Profesor îndrumător:  
Stoian Elena-Florica**

**Nume elev:  
Popa Cătălin  
Clasa a XII-a E**

**2022**

## 1. Introducere

Aplicația Graphy este destinată elevilor care sunt dispuși să învețe teoria grafurilor într-un mod plăcut și interactiv, ea având o direcție artistică care tinde spre jocurile video retro. Acest stil artistic a fost ales datorită limitărilor din punct de vedere al performanței impuse de limbajul de programare C++ când vine vorba de grafică. Aplicația poate fi folosită și de elevii doritori să interacționeze cu partea mai puțin plăcută a programului, codul sursă, Graphy fiind open-source. Grafurile și arborii sunt printre cele mai importante structuri de date care se folosesc în programare, fiind utilizate în orice domeniu al programării, de la jocuri video la baze de date și rețele sociale. Astfel, înțelegerea și stăpânirea lor este pasul pe care orice persoană care tinde spre o carieră în programare trebuie să îl facă.



## 2. Argumentarea temei alese

Tema a fost aleasă datorită pasiunii mele pentru informatică și dorinței de a înțelege toate tehnicile folosite într-o aplicație de acest gen.

Astfel, folosind toate cunoștințele dobândite legate de programare am ales să creez o aplicație care, pe lângă faptul că și-a atins scopul principal, m-a ajutat să devin un programator mai bun.

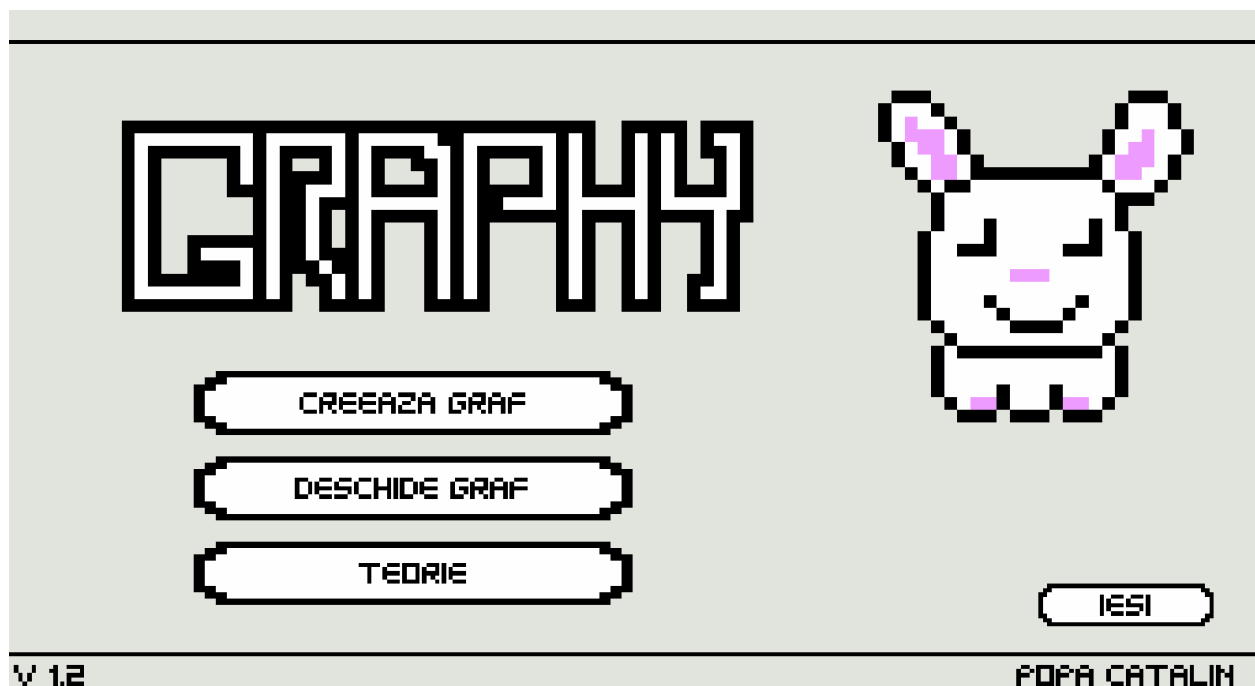
## 3. Prezentarea generală a aplicației

Aplicația se deschide cu ecranul principal de unde utilizatorul poate intra în orice alt ecran al aplicației:

- De editare al grafului (orientat sau neorientat)
- De deschidere sau ștergere al unui graf deja salvat
- De prezentare a teoriei referitoare la cele două tipuri de grafuri,

sau poate ieși din aceasta.

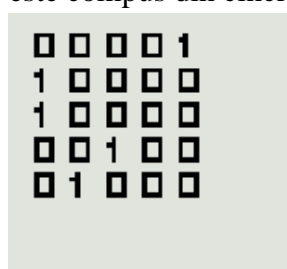
Pe acest ecran se găsește logo-ul aplicației realizat manual și mascota acesteia, un iepure animat în două cadre. În subsolul ecranului se afla versiunea curentă a aplicației și numele dezvoltatorului ei.



Primul buton ne introduce în ecranul de creare a noului graf. În acesta putem selecta tipul grafului și numele acestuia.



După ce a fost introdus numele grafului și selectat tipul acestuia urmează ecranul de editare, care este compus din cinci elemente.



Primul element: locul în care este afișată matricea de adiacență care se actualizează în timp real. În cazul în care matricea are mai mult de șapte elemente și nu mai e vizibilă complet, se poate apăsa tasta TAB pentru a o afișa integral.

GRAF ORIENTAT:  
 SE NUMESTE GRAF ORIENTAT SAU DIGRAF O PERECHE ORDONATA DE MULTIMI  
 NOTATA  $G=(V, U)$ , UNDE:  
 $V$  ESTE O MULTIME FINITA SI NEVIDA ALE CAREI ELEMENTE SE NUMESC  
 NODURI SAU VARFURI.  
 $U$  ESTE O MULTIME DE PERECHEI ORDONATE DE ELEMENTE DISTINCTE DIN  
 $V$  ALE CAREI ELEMENTE SE NUMESC ARCE.

1/9

Al doilea element: spațiul în care este afișată teoria referitoare la graful selectat în ecranul anterior. Definițiile se află într-o rotație care este actualizată la fiecare șapte secunde, dar dacă utilizatorul dorește să ajungă la o definiție mai înaintată poate apăsa săgețile de pe tastatură.

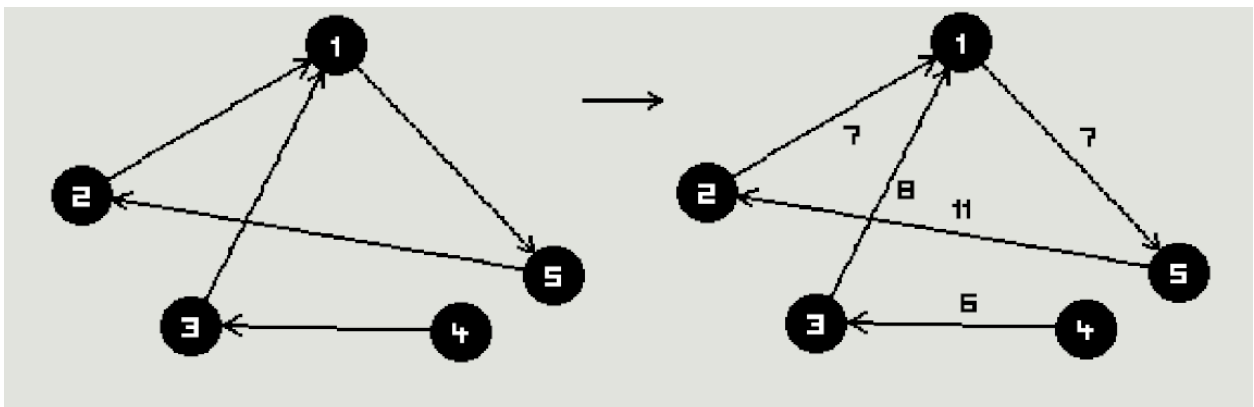
NOD: 1  
 ID (IN MATRICE): 0  
 GRAD INTERIOR: 2  
 GRAD EXTERIOR: 1

Al treilea element: spațiul în care sunt afișate caracteristicile ultimului nod selectat de către utilizator cu mouse-ul. Primul câmp reprezintă numele nodului, care nu trebuie să fie unic și poate fi schimbat. Al doilea este id-ul atribuit de către program nodului, acesta este unic și imposibil de modificat de către utilizator. El reprezintă numărul de ordin al nodului din interiorul matricei de adiacență. Următoarele două câmpuri indică gradul interior și exterior al nodului.

☐ PONDERI  
☐ DIJSKTRA

Al patrulea element: locul în care sunt prezentate cele două opțiuni referitoare la modul de vizualizare al grafului. Prima opțiune este aceea de a vedea distanța dintre noduri, iar a doua activează generarea celui mai scurt drum dintre două noduri selectate de către utilizator.

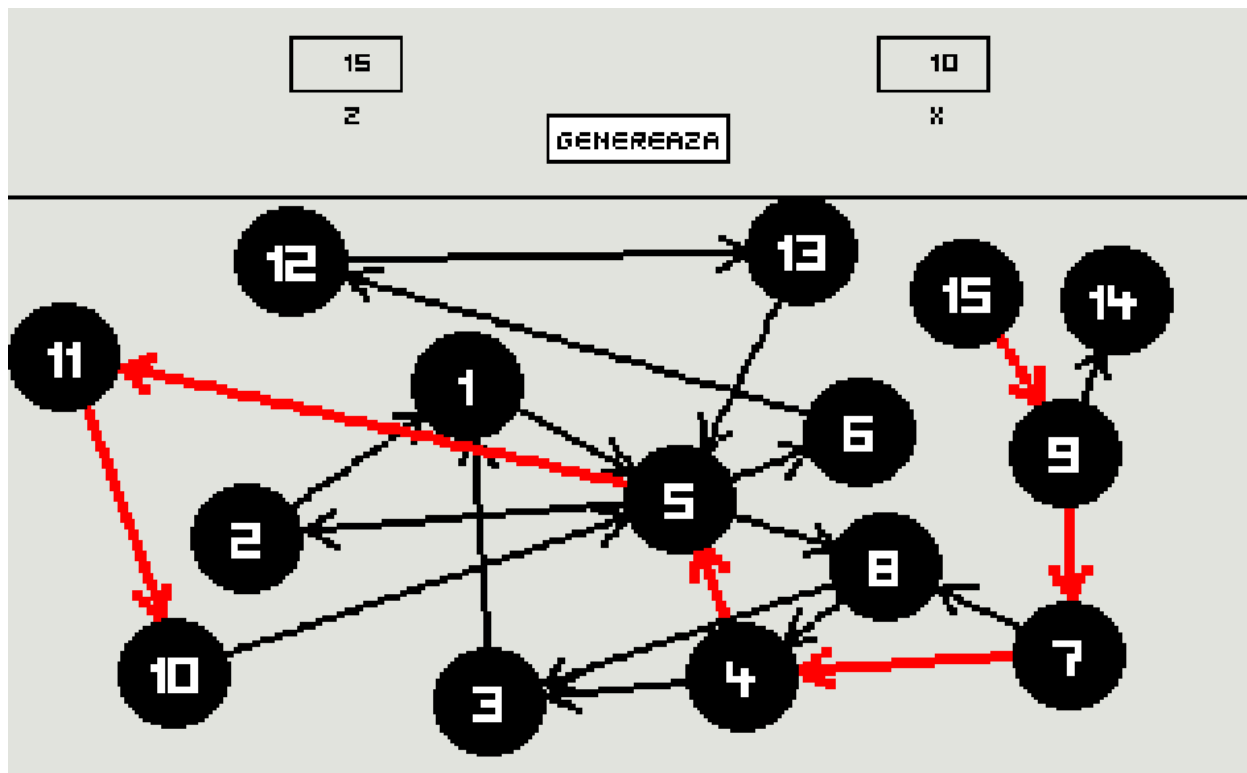
Înainte și după bifarea căsuței marcată "Ponderi":



Ca urmare a bifării celei de-a doua căsuțe al doilea element este înlocuit de elementul responsabil cu configurarea algoritmului lui Dijkstra.

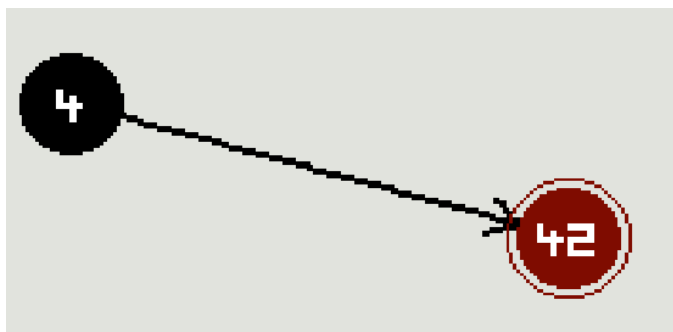
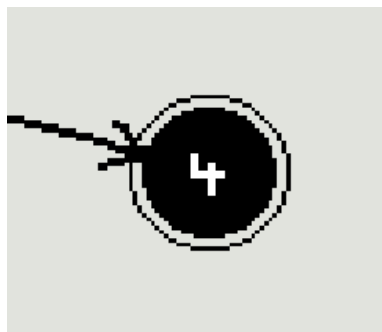


În acest spațiu se află două pătrate, în primul fiind scris numele nodului de pornire iar în al doilea numele nodului destinație. Dacă pătratul este colorat cu roșu atunci următorul nod selectat cu mouse-ul va fi pus în pătratul respectiv, când ambele sunt roșii atunci primul nod selectat se va pune în primul pătrat. Tastele z, respectiv x, șterg nodul din pătratul căruia îi este atribuit. Cea mai scurtă cale dintre cele două noduri selectate este desenată cu roșu.



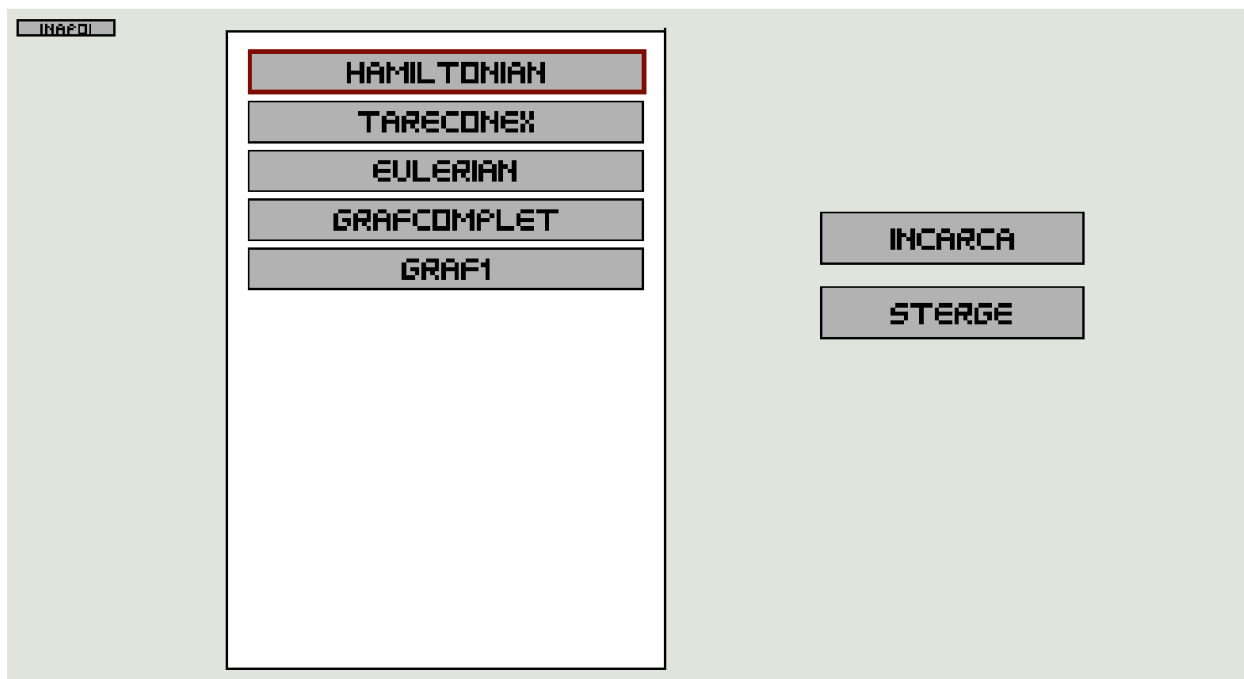
Ultimul element, al cincilea element, este spațiul principal de editare al grafului, în interiorul său utilizatorul poate adăuga sau șterge noduri și muchii, poate muta nodurile oriunde în acest spațiu și le poate schimba numele. Nodul curent selectat este încercuit pentru a oferi o mai bună vizualizare asupra acestuia:

Pentru a schimba numele nodului este necesară apăsarea tastei R, selectarea nodului a cărui nume urmează să fie schimbat, și tastarea acestuia, numele poate fi compus doar din cifre. Nodul modificat este afișat cu roșu.



După ce am terminat de editat graful îl putem salva prin apăsarea tastelor CTRL + S.

Pentru a grafurile salvate trebuie apăsăat al doilea buton de pe pagina principală, anume “Deschide graf” sau putem apăsa tasta TAB când suntem în ecranul principal. După executarea acestor comenzi se este prezentată o listă cu toate grafurile pe care le putem deschide sau șterge.



Grafurile sunt salvate într-un fișier cu extensia “.gf” . Din acest fișier sunt citite toate datele la începutul ciclului de executare al aplicației.

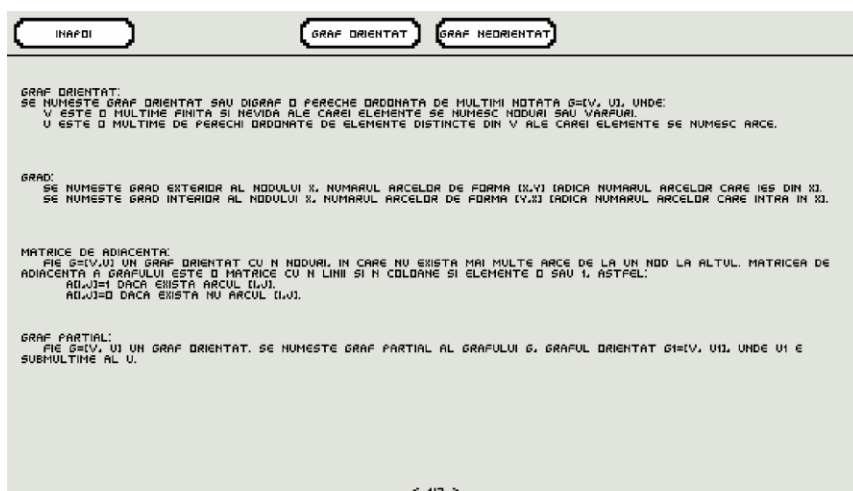
```

GRAFCOMPLET
2
5
0 1 424.981 179.884
1 2 78.0871 228.2
2 3 72.6894 107.41
3 4 209.432 25.2729
4 5 290.038 233.403
0 1 1 1 1
1 0 1 1 1
1 1 0 1 1
1 1 1 0 1
1 1 1 1 0

```

Prima linie din spațiul alocat grafului este numele, acesta e urmat de tipul grafului (1 = orientat, 2 = neorientat) și de numărul de noduri. După aceea urmează cele  $n$  noduri fiecare cu ID-ul, numele și coordonata  $x$  și  $y$  în care se găsește. La final se află matricea de adiacență folosită pentru reconfigurarea muchiilor.

Citirea se face cu ajutorul librăriei “fstream”.



În cazul în care utilizatorul dorește să învețe grafurile prin teorie, acest lucru este posibil în interiorul aplicației prin butonul de “Teorie”. Acesta ne duce pe pagina unde sunt afișate câteva noțiuni de bază referitoare la grafuri.

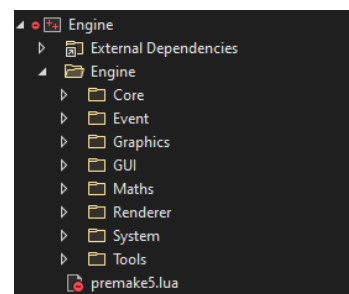
După ce utilizatorul consideră că i-au fost de ajuns grafurile, acesta poate ieși prin apăsarea butonului “Ieși” aflat în partea de jos a ecranului principal.

#### 4. Prezentarea modului în care a fost realizată aplicația

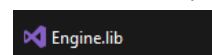
Aplicația a fost realizată în Visual Studio 2022 Community Edition, utilizând limbajul de programare C++. Acesta a fost ales datorită flexibilității oferite și a vitezei de compilare și executare a codului. Pentru reprezentarea vizuală a fost folosită librăria “windows.h”, care pune la dispoziție minimumul necesar în acest scop.

Folosind numai C++, toate sistemele aplicației au fost create de la 0, astfel mi s-a oferit flexibilitatea de a le construi exclusiv pentru Graphy, ceea ce îmi permite personalizarea acestora. Aplicația este împărțită în două proiecte în interiorul Visual Studio:

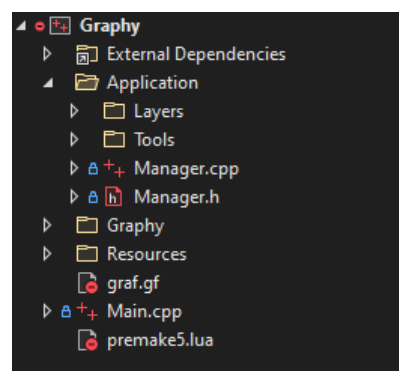
1. **Engine** (aproximativ 3000 de linii de cod), care conține tot codul scris pentru a ușura dezvoltarea aplicației propriu-zise, pune la dispoziție următoarele sisteme care se ocupa de:
  - Crearea ferestrei.
  - Citirea din fișiere externe.
  - Afișarea texturilor salvate într-un format personalizat (.spr).
  - Desenarea unor forme geometrice simple (dreptunghi, cerc, linie)
  - Afișarea scrisului dinamic.
  - Preluarea și trimiterea evenimentelor ce se produc la nivelul ferestrei sau al tastaturii și mouse-ului.
  - Crearea și interacționarea cu elementele interfeței grafice a utilizatorului.
  - Ușurarea utilizării matematicii la nivelul aplicației.
  - Schimbarea între ecranele aplicației.
  - Afișarea în consola a mesajelor de tip: info, avertisment, eroare și eroare fatală, în culorile specifice.



Acesta se compilează sub forma de librărie statică și creează un fișier de tip .lib de unde aplicația principală extrage codul deja compilat. Modul în care a fost scris permite folosirea sa și în alte proiecte.



2. **Graphy** (proiectul principal, aproximativ 2100 de linii de cod), care conține tot codul executat în interiorul aplicației și se folosește în mod exclusiv de codul scris în **Engine**. Proiectul este responsabil, în principal, cu logica din spatele aplicației și cu declararea și definirea claselor necesare (nod, muchie, graf). În interiorul lui se afla toate resursele necesare rulării aplicației.



#### 5. Bibliografie

C++ Primer Forth Edition

<http://members.chello.at/~easyfilter/bresenham.html>