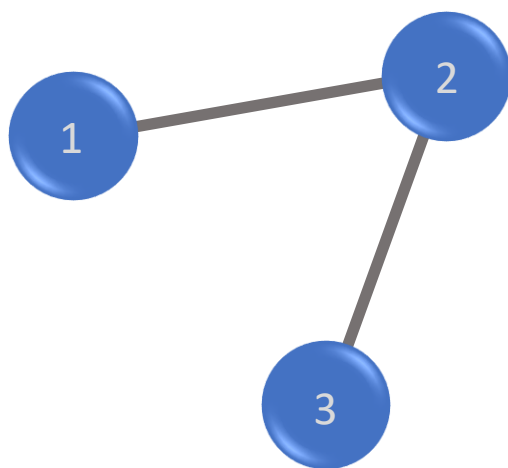


Graph Viewer

O aplicatie desktop cross platform



Proiect realizat de:
Panait Ana-Maria

Colegiul Național de Informatică „Tudor Vianu”

Contents

| | |
|------------------------------------|---|
| 1. Introducere..... | 3 |
| 2. Arhitectura aplicației..... | 3 |
| Tehnologii folosite..... | 3 |
| Implementarea aplicației | 3 |
| 3. Interfață..... | 4 |
| 4. Conținut..... | 7 |
| 5. Originalitate și inovație | 8 |

1. Introducere

GraphViewer este o aplicație desktop, cross platform, ce are ca scop aprofundarea teoriei grafurilor și a algoritmilor corespunzători, prin posibilitatea vizualizării acestora în timp real pe graful construit de utilizator. Acesta poate construi graful prin introducerea listei de muchii/listei de adiacență/matricei de costuri/matricei de adiacență sau folosind mouse-ul. Aplicația are și teorie despre grafuri disponibilă offline care facilitează învățarea indiferent de disponibilitatea semnalului, existând posibilitatea de a-ti verifica cunoștințele noi dobândite cu teste de evaluare, la care utilizatorul primește feedback.

Ideea aplicației a venit din pasiunea mea pentru informatică și pentru că la școală tocmai am făcut capitolul de grafuri și am simțit nevoia de o modalitate mai rapidă de a desena grafurile, pentru că, mai ales în probleme mai complicate, un desen clar și ușor de realizat ajuta la rezolvare. De asemenea, mai ales la început, mi-a fost greu să rețin toate conceptele noi legate de grafuri și m-ar fi ajutat să am o aplicație ca aceasta unde să am toată teoria structurată clar.

2. Arhitectura aplicației

Tehnologii folosite

Aplicația GraphViewer este implementată în Python cu ajutorul framework-ului Kivy. Am ales Python pentru că este ușor de folosit, facilitează dezvoltarea rapidă a aplicațiilor și minimizează numărul de linii de cod. Am ales Kivy, deoarece este ușor de folosit, intuitiv și cross platform, aplicația putând fi rulată pe Linux, Windows, OS X, Android, IOS, cu același cod. Aplicația are comportamentul dorit momentan pe Linux și Windows.

Implementarea aplicației

Aplicația folosește clase pentru a implementa graful, una pentru noduri, una pentru muchii și una pentru managementul grafului. Acolo se țin instanțele celor două clase menționate mai sus și sunt implementate metode de adăugare/ștergere de noduri/muchii cât și de update și interpretare a datelor date de utilizator și tot ce are legătura cu graful.

Aplicația are trei pachete: „graph” în care se află tot ce este legat de implementarea internă a grafului, „gui” în care se află fișierele legate de interfața grafică și „globals” în care țin variabile globale care sunt necesare pentru a transmite date între celelalte două pachete. De asemenea, pe lângă aceste pachete, există și trei foldere cu resursele necesare rulării aplicației: „fonts” în care se află un fișier de tip ttf pentru încărcarea fontului necesar simbolurilor speciale care se găsesc în partea de teorie a aplicației,

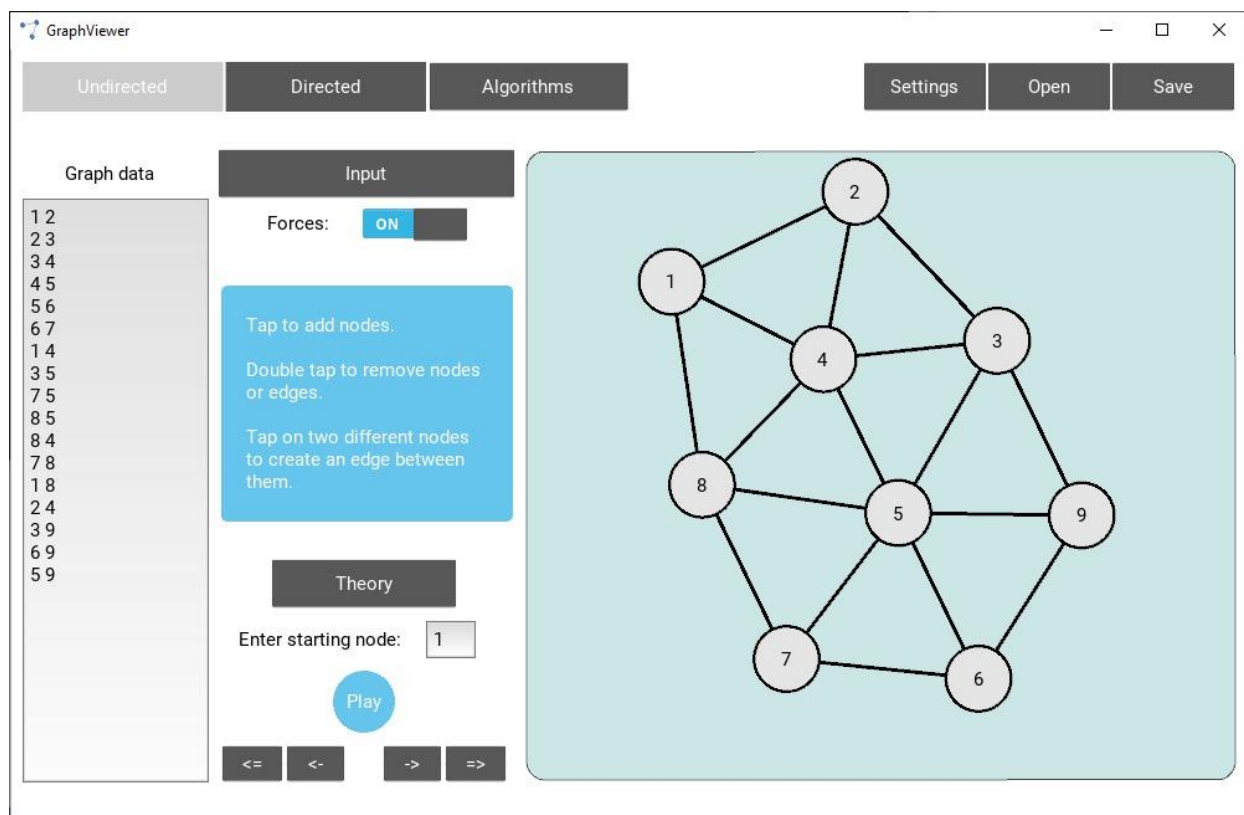
„images” în care sunt stocate imaginile și icon-urile folosite în interiorul aplicației și „theory” în care se găsesc fișierele text din care este încărcată teoria.

Aplicația a fost testată funcție cu funcție pe măsură ce a fost dezvoltată și apoi a fost încercată de cunoștințe pentru a primi feedback.

În dezvoltarea aplicației s-a folosit sistemul de versionare Git pentru a ușura menținerea codului. Repository-ul aplicației este hostat pe <https://github.com/anamariapanait10/GraphViewer>.

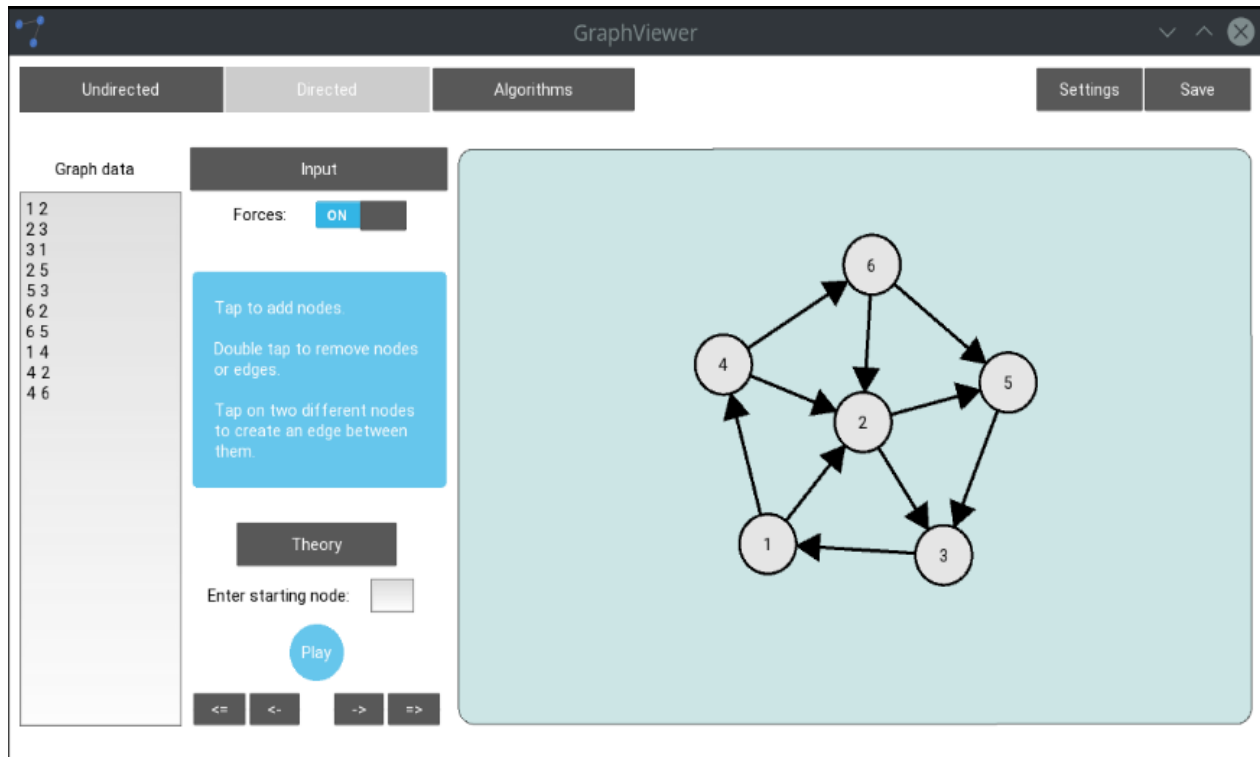
3. Interfață

Interfața cuprinde un text box, unde utilizatorul poate introduce datele pentru generarea grafului. Pentru a schimba tipul datelor de intrare se selectează butonul Input care deschide un drop down list cu cele patru posibilități: listă de muchii, listă de adiacență, matrice de adiacență, matrice de costuri. Setarea default pentru acest parametru este lista de muchii. Datele sunt validate la fiecare schimbare a textului, iar dacă utilizatorul a introdus date greșite sau formatul acestora nu este valid, textul își va schimba culoarea în roșu, iar graful nu va fi generat decât dacă datele sunt corecte.



În partea din stânga sus se află butoanele care modifică dacă graful introdus este orientat sau neorientat. Setarea implicită pentru acest parametru este de graf neorientat. În partea dreaptă este locul

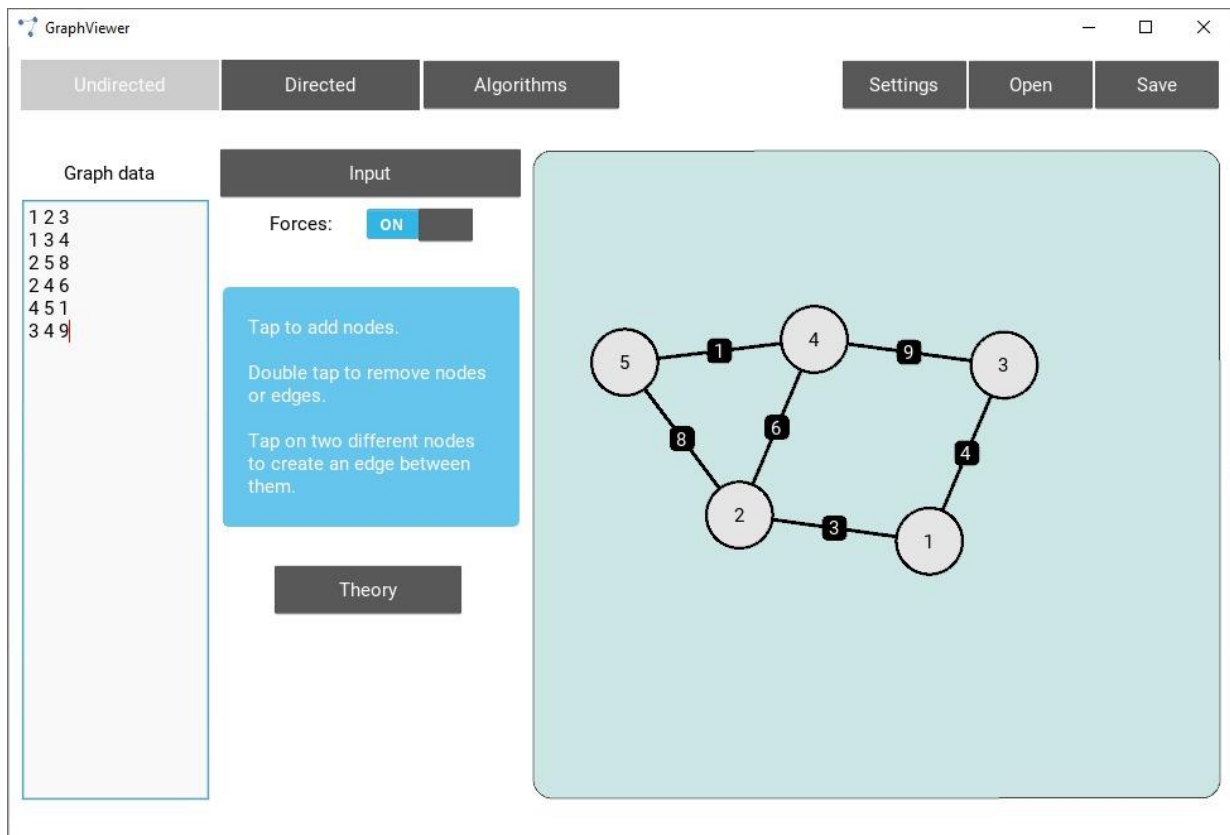
predestinat desenării grafului. Grafurile pot fi construite și folosind mouse-ul. Astfel, dacă utilizatorul dă click în spațiul de desenare, va fi generat un nou nod cu urmatorul id, iar dacă nodurile existente nu au id-ul consecutiv, atunci se va genera un nod cu cel mai mic id care lipsește. Dacă utilizatorul dă dublu-click pe un nod, atunci el se va șterge, iar dacă utilizatorul selectează două noduri se va crea muchie între ele.



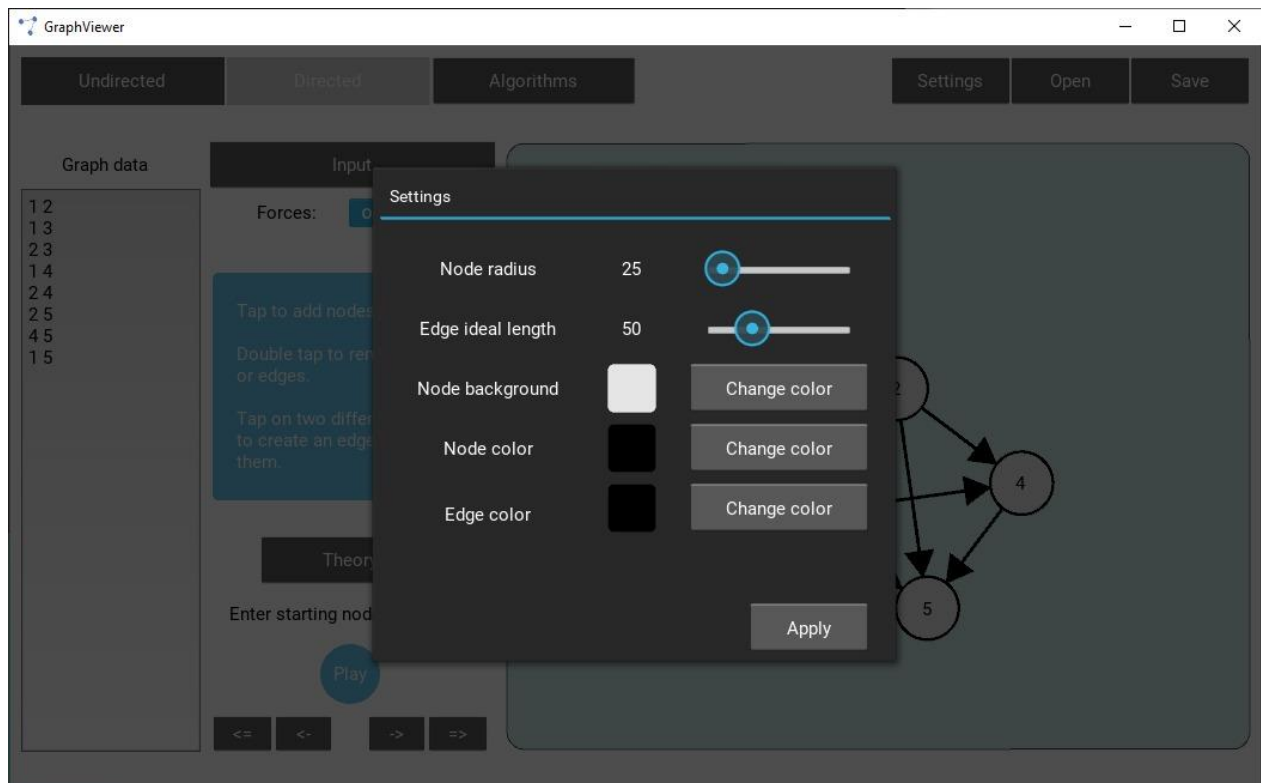
Graful este interactiv și suportă opțiunea de „drag and drop” pentru mutarea nodurilor și, de asemenea, este implementat un algoritm care calculează forțe de atracție și de respingere între noduri. Astfel, când utilizatorul trage de un nod acesta își modifică poziția, trăgând după el toate celelalte noduri cu care este conectat prin muchii, astfel încât, graful întotdeauna tinde să se ordoneze spre o „stare de echilibru” a forțelor.

În cazul în care se dorește oprirea forțelor se apasă pe switch-ul intitulat „Forces” și astfel nodurile grafului pot fi mutate cu mouseul și nu își vor mai modifica poziția în funcție de forțele dintre noduri.

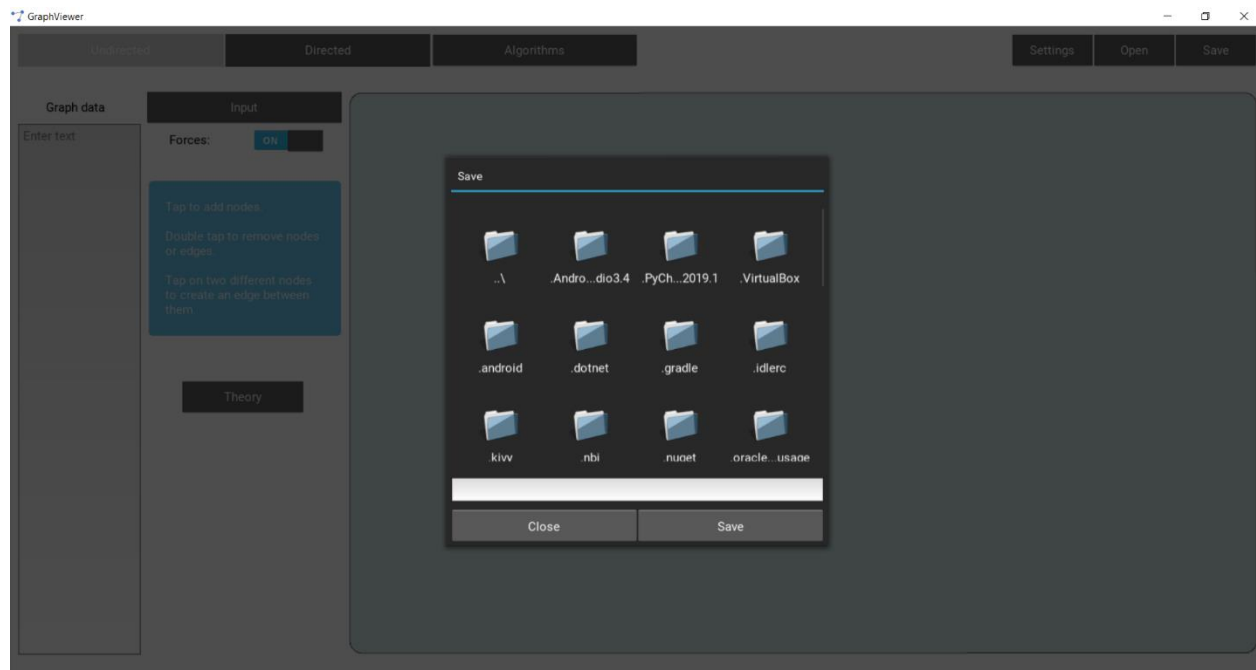
Graful are posibilitatea adăugării de costuri pe muchii, opțiune care funcționează când este selectată introducerea datelor prin lista de muchii sau matricea de costuri.



Utilizatorul poate personaliza după preferințe caracteristicile grafului precum culorile folosite (culoarea de fundal, conturul nodurilor, și culoarea muchiei), lungimea ideală a muchiilor și dimensiunea nodurilor (raza).



GraphViewer oferă posibilitatea de a salva un graf creat într-un fișier text care apoi poate fi încărcat înapoi în aplicație pentru a continua studiul pe el. Butonul Save deschide un file chooser, unde utilizatorul își poate alege calea unde va fi salvat fisierul cu datele grafului, sau poate tasta direct calea în text box-ul de sub acesta. Asemănător se întâmplă și în cazul butonului Open.

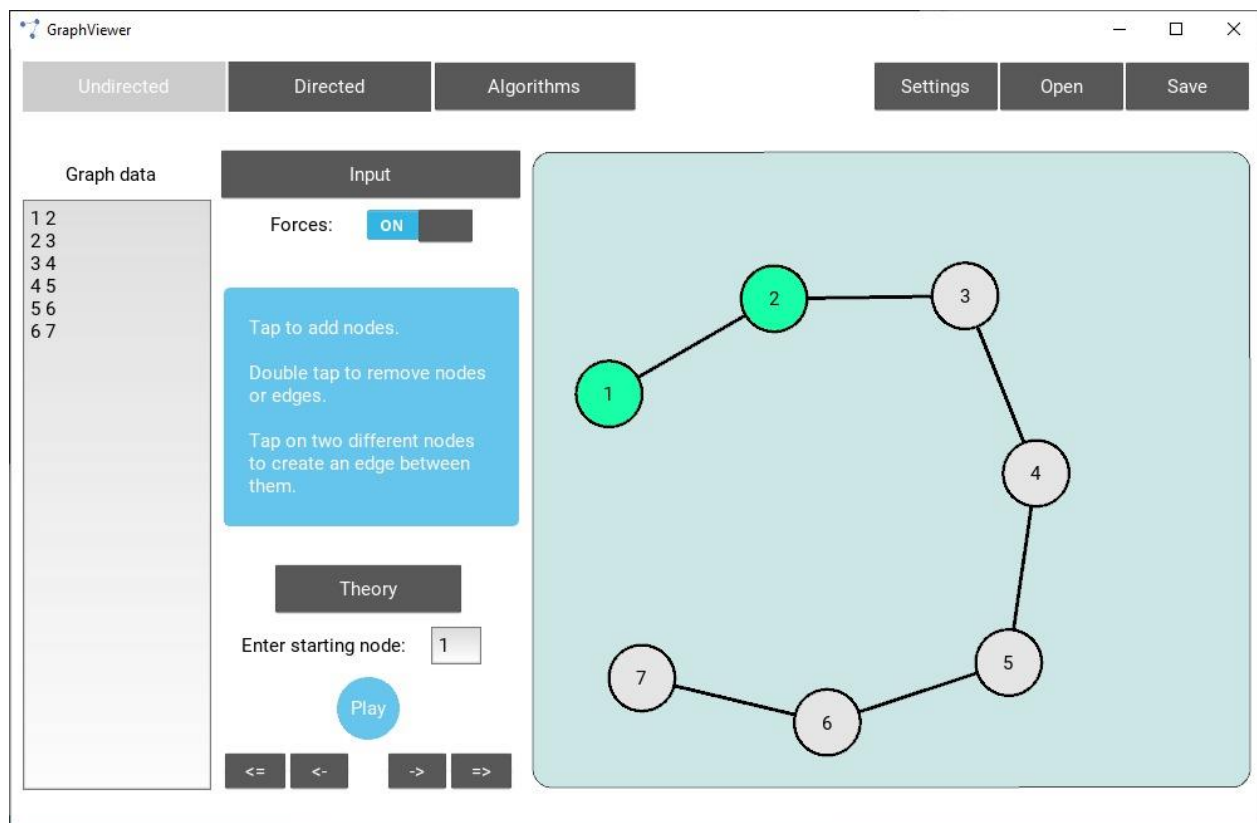


Aplicația este responsive, răspunzând la modificările de dimensiune a ferestrei. Aceasta are setată o dimensiune minimă și pornește maximizată.

4. Conținut

Pe partea de teorie, aplicația prezintă definițiile de bază ale teoriei grafurilor împreună cu exemple care au fost desenate tot cu ajutorul acestei aplicații. Utilizatorul își poate verifica cunoștințele dobândite parcurgând testele de evaluare grilă cu dificultate progresivă, pentru care primește și feedback.

De la butonul Algorithms se poate selecta algoritmul pe care utilizatorul îl poate vizualiza în timp real pe graful construit de acesta, prin colorarea nodurilor parcurse de acesta. Utilizatorul este astfel implicat în procesul de învățare, rezultând aprofundarea rapidă a teoriei.



5. Originalitate și inovație

Deși mai există aplicații dedicate grafurilor, aceasta îmbină desenarea cu mai multe moduri de introducere a datelor (atunci când utilizatorul introduce un nod sau o muchie cu mouse-ul se va actualiza și textul din editorul de text) cu noțiuni teoretice și teste de evaluare, astfel încât ea formează un pachet cu tot ce este nevoie pentru dobândirea unei baze solide în ceea ce privește teoria grafurilor.

Aplicația are avantajul că generează un graf interactiv pe care utilizatorul îl poate modifica și personaliza după preferințe astfel încât să ofere tot ajutorul de care acesta ar putea avea nevoie.