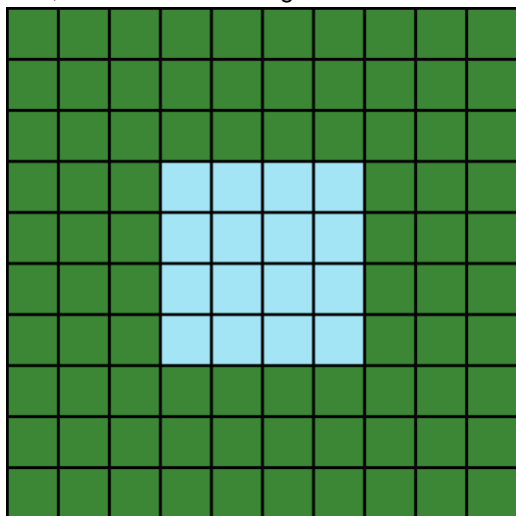


Rătuște și pisici

Se va implementa jocul cu descrierea de mai jos, urmărind cerințele. Puteți porni de la codul dat în laborator: <https://replit.com/@IrinaCiocan/x-si-0-interf-grafica-matrice#main.py> și pe care îl puteți modifica, marcând printr-un comentariu locurile în care ați modificat pentru rezolvarea cerințelor.

Opțional puteți implementa problema de la zero, dacă doriți.

Se consideră o tablă de dimensiuni $N \times N$, care va conține malul și lacul cu rătuște, cu $8 \leq N \leq 10$. Se consideră K dimensiunea malului. Malul va fi de grosime K și va reprezenta primele și ultimele K coloane, și primele și ultimele K linii. K este citit de la tastatură, cu condiția să avem $K < N/3$. Zona rămasă în centru este lacul. Pentru $K=3$, zona verde din imagine este malul și cea albastră este lacul.



Jucătorul-rătușcă este cel care începe întotdeauna. Jucătorul-rătușcă poate plasa simboluri noi doar pe lac. Jucătorul pisică poate plasa simboluri noi doar pe mal.

Ambii jucători pot opta ca în loc de a plasa un simbol, să mute un simbol. Simbolurile pot fi mutate doar pe linie sau pe coloană. Un jucător nu poate realiza mai mult de K deplasări consecutive. Dacă a făcut K deplasări consecutive, e obligat ca următoarea mutare să fie o plasare de simbol nou.

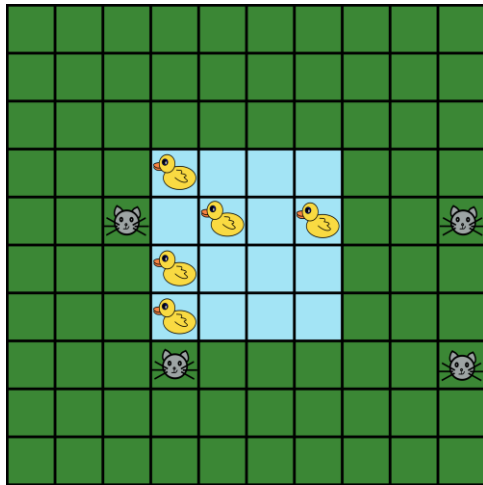
Dacă în urma unei mutări (deci deplasări de simbol, nu punere de simbol nou), jucătorul-rătușcă are pe linie sau pe coloană 3 rătuște (nu neapărat pe poziții consecutive), atunci, în cazul în care pe acea linie sau coloană se află o pisică (cu excepția cazului în care e chiar pe marginea gridului - adică prima/ultima linie respectiv prima/ultima coloană), pisica va face bândăbâc în lac și învinsă va pleca de pe tablă. E posibil ca printr-o mutare de rătușcă să se facă și o linie și o coloană cu 3 rătuște; în acest caz, vor fi capturate 2 pisici. **Rătuștele vor captura toate pisicile care îndeplinesc condițiile.**

O pisică de asemenea poate captura o rătușcă, dar tot doar în urma unei deplasări (deci nu la punerea unui simbol nou). Dacă în urma mutării, lângă pisică (la distanță Manhattan ≤ 2 , unde distanța Manhattan între două poziții 1 și 2, este $\text{abs}(\text{linie1} - \text{linie2}) + \text{abs}(\text{coloana1} - \text{coloana2})$) se află o rătușcă, pisica va captura rătușca. **Pisica nu poate captura mai mult de o rătușcă** la o deplasare, dar poate alege care rătușcă să fie capturată. Pisica este flămândă, așadar nu poate opta să nu captureze o rătușcă dacă are rătuște la distanța cerută.

În funcția de generare a mutărilor, veți verifica tipul jucătorului curent (rătușcă sau pisică) și veți genera mutările și eventualele capturi în funcție de tipul jucătorului (deci aveți 2 cazuri).

Exemplu:

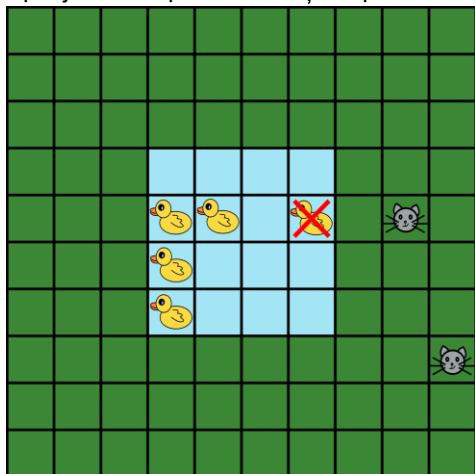
Avem tabla curentă:



Considerăm liniile și coloanele numerotate de la 0. Jucătorul rățușcă mută de pe (3,3) pe (4,3) (și capturează pisicile tăiate cu X - a capturat 2 pisici, deoarece simbolul muta se află atât pe o linie cât și pe o coloană). Pisica încercuită cu galben nu poate fi capturată deoarece e chiar pe marginea gridului. Dacă aceasta pisică era cu o poziție la stânga, ar fi fost și ea capturată.



Apoi jucătorul-pisică mută și capturează rățușca tăiată cu X):



Jocul se termină după ce unul dintre jucători a reușit să efectueze N capturi. Acel jucător este și cel care câștigă. Dacă înainte de cele N capturi, un jucător nu mai are niciun fel de mutări disponibile, pierde jocul.

Gasiti imaginile pentru simbolurile jucatorilor la:

https://drive.google.com/drive/folders/17_C4a_peVcJSrq5pnyCDSO0U2064wmpc?usp=sharing

Cerințe(4.5p+1.5 bonus - punctajul se trunchiaza la 4.5):

1. Se va citi dimensiunea tablei de la tastatura si se va genera tabla initiala desenata conform cerintei. Setati adâncimea maximă pentru arborele minimax/alpha-beta 2. **(0.25 puncte)** Celulele tablei vor fi colorate ca in imagini, iar simbolurile de pe tabla vor fi imaginile date pentru cei doi jucători. **(0.25 puncte)**
2. Scrieți o funcție care verifică dacă o mutare este validă (funcția de exemplu poate primi printre parametri, tabla veche și coordonatele noului simbol, eventual jucătorul curent (sau simbolul lui, pentru a ști cine mută) și alți parametri pe care îi considerați necesari). Folosiți această funcție pentru a verifica dacă mutarea utilizatorului e corectă.**(1 punct)**
3. Modificați funcția de generare a mutărilor conform cerințelor problemei. **(3 puncte)**
4. Modificați funcția de verificare a stării finale **(0.5 puncte)**
5. Modificați funcția de estimare a scorului astfel încât să returneze un număr cât mai mare pentru **(0.5 puncte)**
6. Afișați numărul de noduri generate pentru fiecare arbore minimax sau alpha-beta și timpul mediu de gândire al calculatorului **(0.5 puncte)**

