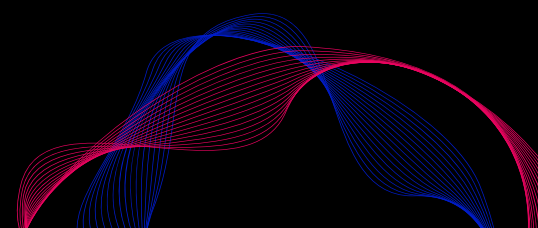




Titlu: Proiect Candy Crush

Prezentare bot Python pentru Candy Crush.

- Scop: Automatizarea jocului Candy Crush.
- Mediu de lucru: Python.
- Obiectiv: Explorarea programării prin jocuri.

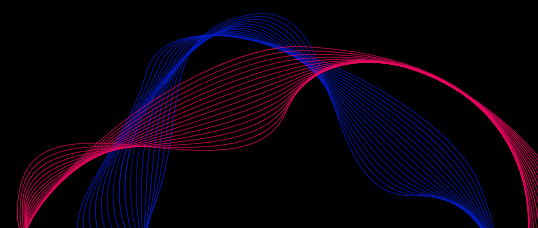




Înțelegerea funcționării programului

Cum funcționează programul Candy Crush simplificat.

- Găsește combinații: Programul caută automat grupuri de bomboane identice.
- Șterge bomboane: Elimină bomboanele care fac parte dintr-o combinație.
- Cad bomboane: Bomboanele de sus cad pentru a umple golurile.
- Apar bomboane noi: Programul adaugă automat bomboane noi în grilă.

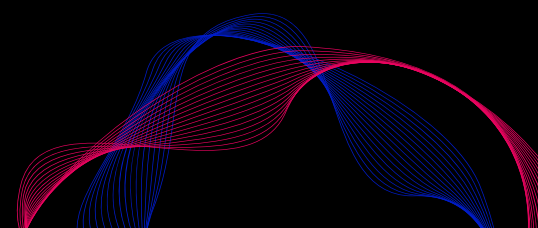




Numărarea swap-urilor valide

Un aspect crucial al botului este modul în care gestionează mișcările.

- Swap-uri Valide: Numărăm doar mișcările care conduc la formarea unei combinații.
- Cascade: Cascadele (combinații multiple dintr-o singură mișcare inițială) sunt bune, dar nu cresc numărul de swap-uri utilizate. Fiecare mișcare inițială contează ca un singur swap, indiferent de câte reacții în lanț provoacă.



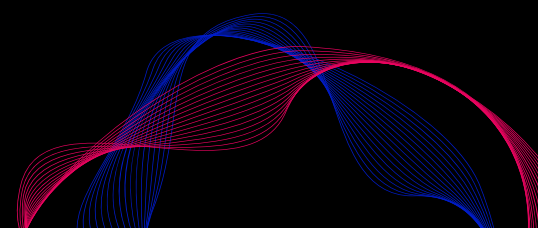


Sistemul de notare explicat

Scorul este direct legat de lungimea combinațiilor.

- 3 în linie: 20 de puncte per combinație
- 4 în linie: 30 de puncte per combinație
- ≥ 5 în linie: 40 de puncte per combinație

Cu cât combinația e mai mare, cu atât scorul crește mai repede!

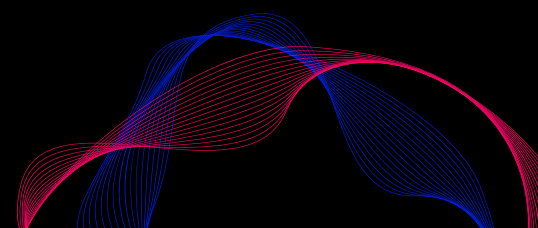




Proces de simulare

Pentru a evalua performanța botului, am efectuat o serie de simulări.

- Rulare: Am rulat 100 de jocuri complete.
- Statistici: Am colectat date despre:
 - Puncte medii obținute.
 - Numărul mediu de swap-uri efectuate.
 - Numărul mediu de swap-uri pentru jocurile care au atins 10.000 depuncte.

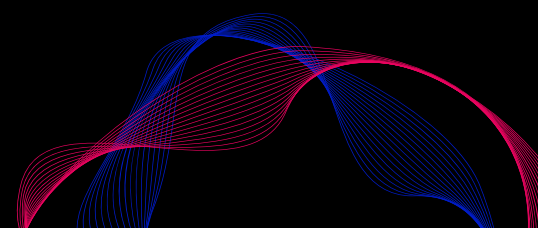




Indicatori de performanță

Indicatorii ne-au ajutat să înțelegem eficiența botului.

- Puncte Medii: 0 medie mai mare indică o strategie mai bună.
- Swap-uri Medii: Mai puține swap-uri pentru același scor sugerează o eficiență sporită.
- Swap-uri Medii (10.000p): Această metrică arată cât de eficient este botul în jocurile de succes.

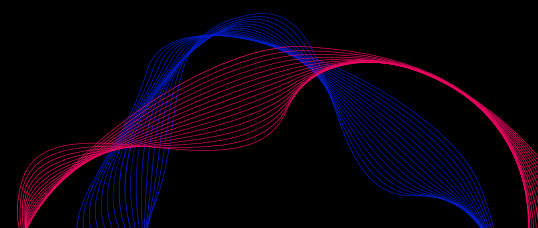




Testarea unitară

Am asigurat fiabilitatea codului prin teste unitare

- Generare Tablă: Teste pentru a verifica dacă tablele de joc sunt generate corect.
- Implementare Gravitațională: Teste pentru a ne asigura că bomboanele cad corect.
- Detecție Formațiuni: Teste pentru a verifica acuratețea detectării combinațiilor.

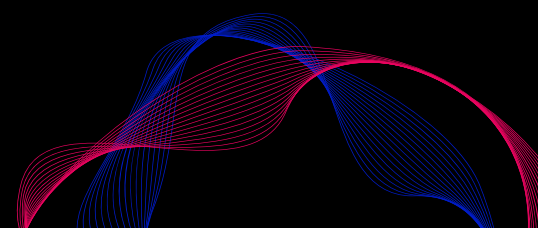




Colectarea și analiza datelor

Rezultatele au fost stocate pentru analiză detaliată.

- Format CSV: Am salvat toate datele într-un format CSV ușor de utilizat.
- Analiză: Utilizând aceste date, putem explora performanța botului, identifica punctele forte și zonele care necesită îmbunătățiri.



Probleme întâmpinate la crearea botului

Crearea botului a avut câteva dificultăți importante.

- Algoritmi rapizi: A fost greu să găsim metode rapide de a vedea combinațiile posibile.
- Reacții în lanț: Trebuia asigurat faptul că reacțiile după fiecare mutare sunt corecte.
- Testare atentă: Am verificat dacă botul funcționează bine în multe situații diferite din joc.

Îmbunătățiri și îmbunătățiri viitoare

Îmbunătățirile sunt mereu pe drum!

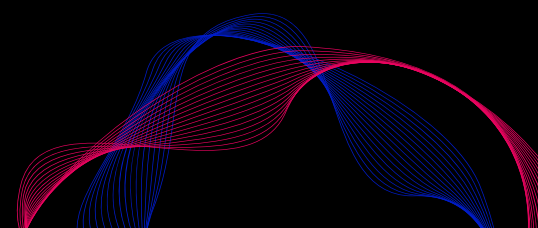
- Strategii Avansate: Implementarea unor strategii mai sofisticate pentru a maximiza scorul.
- Învățare Automată: Antrenarea botului folosind algoritmi de învățare automată pentru a se adapta și a îmbunătăți performanța în timp.
- Interfețe cu Utilizatorul: Crearea unei interfețe grafice prietenoase pentru a interacționa mai ușor cu botul.



Principalele învățăminte din proiect

Ce am învățat din această aventură?

- **Importanța Algoritmilor:** Algoritmii eficienți sunt cruciali pentru performanța optimă.
- **Testarea Riguroasă:** Testarea amănunțită este esențială pentru identificarea și corectarea erorilor.
- **Perseverența:** Rezolvarea problemelor complexe necesită răbdare și determinare.





Concluzie și întrebări și răspunsuri

În concluzie, crearea unui bot de Candy Crush în Python a fost o experiență de învățare valoroasă, plină de provocări și satisfacții.

