**PROIECT ÎN CADRUL DISCIPLINEI POO**

-Spânzurătoarea-

***Autor:   
 Student Nistor Elena-Simona***

***Grupa***

***3122B***

# CUPRINS

[CUPRINS 2](#_Toc136174506)

[TEMA ȘI MOTIVAȚIA ALEGERII : 3](#_Toc136174507)

[DESCRIEREA PROBLEMEI: 3](#_Toc136174508)

[DESCHIDEREA PROIECTULUI ÎN CONSOLA LINUX 5](#_Toc136174509)

[ELEMENTE SPECIFICE POO: 8](#_Toc136174510)

[Cazuri excepționale , prezentarea opțiunii 2, 3 și 4 14](#_Toc136174511)

TEMA ȘI MOTIVAȚIA ALEGERII :  
Tema aleasă pentru proiectul din cadrul disciplinei POO este jocul Spânzurătoarea. Acesta presupune o interfață destul de simplă, cu o grafică la fel de simplă, ce include o mini hartă cu cele 6 nivele ale jocului , după care apariția spănzurătorii după începerea nivelului, și continuarea spânzurătorii la alegerea unei litere greșite.

Am ales proiectarea acestui joc deoarece este un joc popular și destul de cunoscut, ceea ce va face ca acest proiect să fie relevant și accesibil pentru un public larg.De asemenea un alt motiv îl reprezintă dezvoltarea abilităților cognitive, adică acest joc implică gândirea critică , analiza și sintetizarea informațiilor. Un alt motiv îl reprezintă implementarea jocului, care are o interfață simplă și intuitive, cee ace îl face ușor de utilizat și de înțeles pentru orice utilizator de orice vârstă.

Prin urmare, acest proiect poate aduce o valoare reală și practică pentru toți utilizatorii care doresc să se destindă prin a juca un joculeț de dificultate medie și ușor de utilizat.

# DESCRIEREA PROBLEMEI:

Algoritmul pentru jocul Spanzurătoarea poate fi divizat în două părți principale: inițializarea jocului și ciclul de joc. Iată o posibilă descriere a algoritmului:

1. Inițializarea jocului:

* Alege un cuvânt aleatoriu dintr-o listă prestabilită.
* Afișează utilizatorului numărul de litere din cuvânt sub forma de spații goale(underscore).
* Inițializează numărul de încercări rămase la o valoare prestabilită (de exemplu, 6).
* Inițializează lista cu litere introduse de utilizator la o listă goală.
* Odată ce o literă a fost aleasă , nu va mai putea fi introdusă din nou până la nivelul următor .

1. Ciclul de joc:

* Așteaptă până când utilizatorul introduce o literă.
* Verifică dacă litera introdusă se află în cuvântul ales.

Dacă litera introdusă este în cuvânt:

* Actualizează spațiile goale corespunzător cu litera introdusă.
* Verifică dacă utilizatorul a ghicit tot cuvântul. Dacă da, afișează un mesaj de felicitare și încheie jocul.

Dacă litera introdusă nu este în cuvânt:

* Scade numărul de încercări rămase.
* Afișează o parte a imaginii cu spânzurătoarea corespunzătoare numărului de încercări rămase.
* Verifică dacă utilizatorul a epuizat toate încercările. Dacă da, afișează cuvântul și un mesaj de înfrângere și încheie jocul.
* Adaugă litera introdusă la lista cu litere introduse de utilizator.
* Afișează starea jocului utilizatorului (spațiile goale, literele introduse, imaginea cu ghilotina).
* Repetă ciclul de joc până când utilizatorul ghicește cuvântul sau epuizează toate încercările.

# DESCHIDEREA PROIECTULUI ÎN CONSOLA LINUX

De precizat este faptul că am folosit virtual box pentru a putea lucra în Linux.

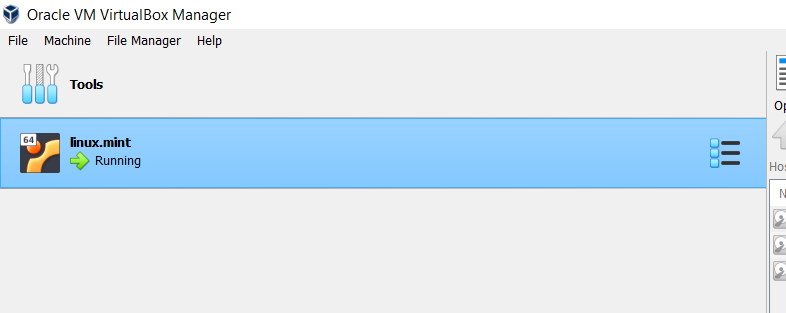


Figura 1. Virtual box și versiunea de linux instalată

Pentru a putea deschide proiectul în consola Linux se vor respecta pașii:

1. Click dreapta pe ecran

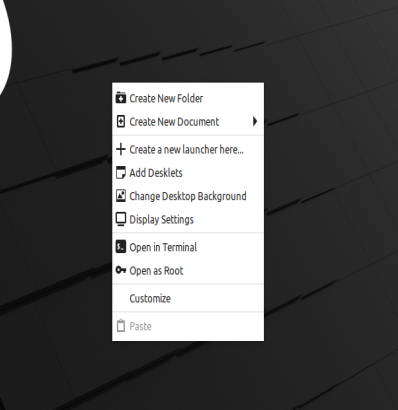


Figura 2. Pasul 1 pentru a deschide aplicația în consola Linux

2. Se alege opțiunea “Open in Terminal”:

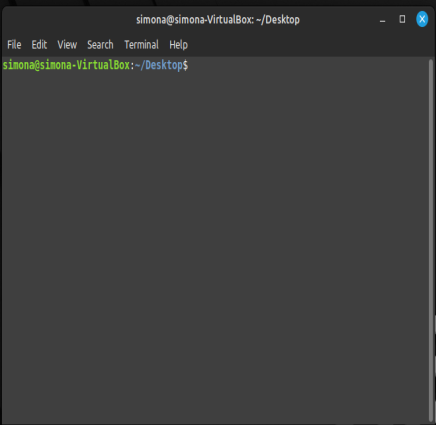


Figura 3. Deschiderea terminalului în linux

3. Se va deschide consola și vom scrie următoarele două linii în consolă:

a) ls(cu care verificăm dacă avem folderul cu proiectul curent)

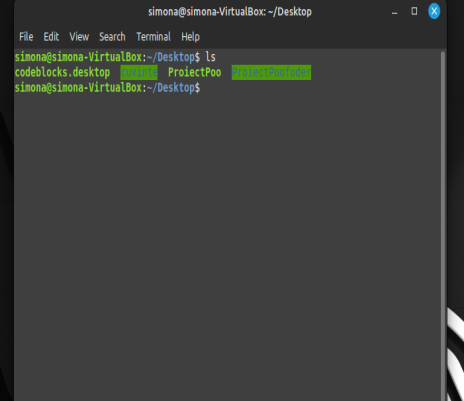


Figura 4. Prima comandă pentru a deschide proiectul.

Se observă după scrierea comenzii ls apare și folderul cu proiectul , deci îl putem

deschide.

b) ./și denumirea folderului cu proiectul , în cazul acesta “./ProiectPoo” și apăsarea tastei ENTER



Figura 5. A doua comandă pentru a deschide proiectul

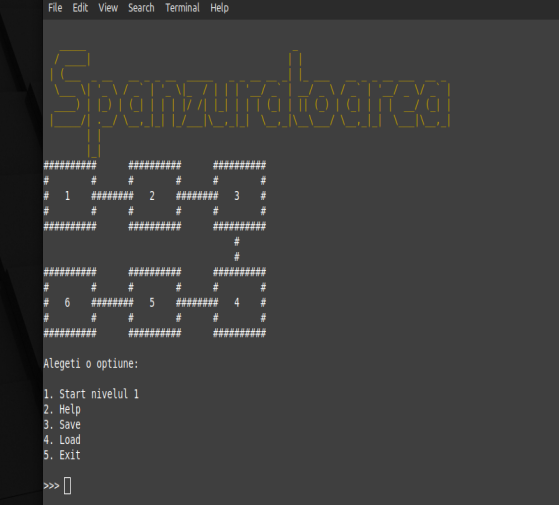


Figura 6. După apăsarea tastei ENTER

# ELEMENTE SPECIFICE POO:

Pentru a crea jocul Spânzurătoarea în C++, vom folosi POO (Programare Orientată pe Obiecte) pentru a structura și organiza codul.

De menționat este faptul că acest proiect a fost creat și va fi rulat din sistemul de operare linux, mai exact din linia de comandă linux.

Meniul jocului va fi destul de intuitiv conform figurii 7.

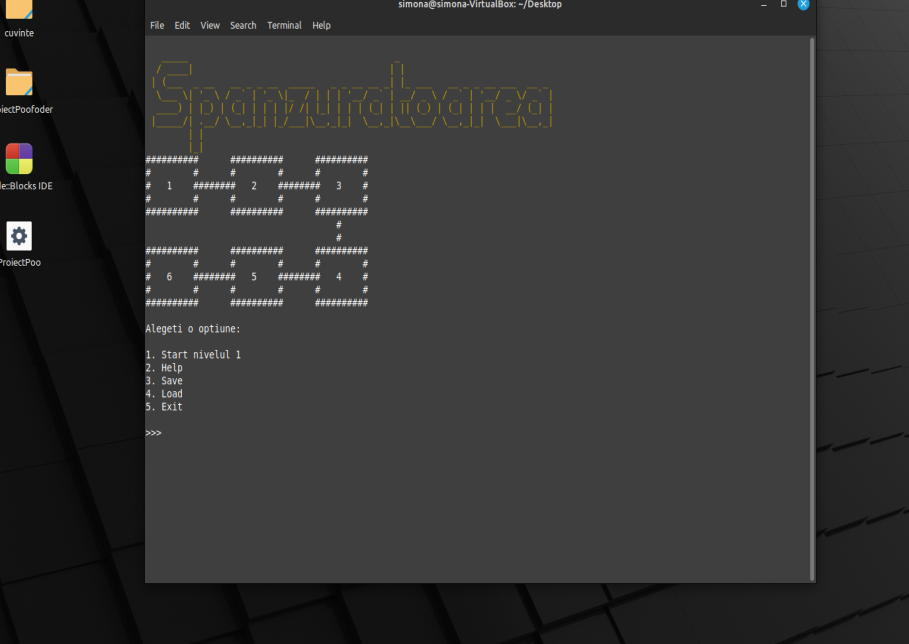


Figura 7. Meniul jocului

În continuare, vom descrie detaliat fiecare element al programului OOP:

CLASE:

1. Clasa "JocHandler": Această clasă va fi folosită pentru menținerea meniului , jocului , setărilor și statusul programului.

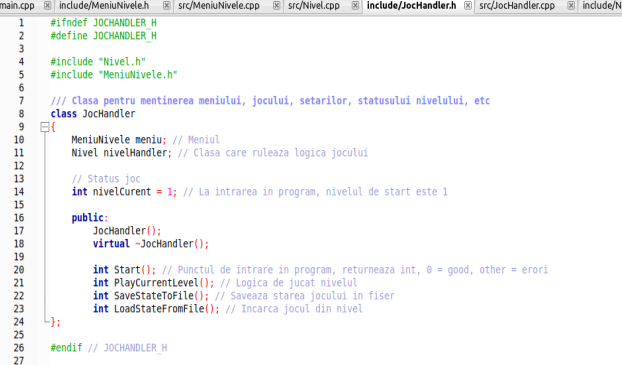


Figura 8 . Header-ul pentru Clasa JocHandler

Aici avem 4 metode folosite după cum urmează

* Start () = punctul de intrare în program, returnează int, 0=ok, altceva=eroarea
* PlayCurrentLevel() = Logica de jucat nivelul
* SaveStateToFile() = Saveaza starea jocului in fiser
* LoadStateFromFile() = încarcă jocul de la nivelul pe care l-am salvat anterior(adică dacă am rămas la nivelul 4 și am salvat, chiar dacă închid programul, la redeschiderea consolei , voi alege din meniu acestă opțiune și se va redeschide jocul de la nivelul 4).

1. Clasa “MeniuNivele": Această clasă va va afisa harta cu nivele și va citi opțiunea din meniu a utilizatorului.

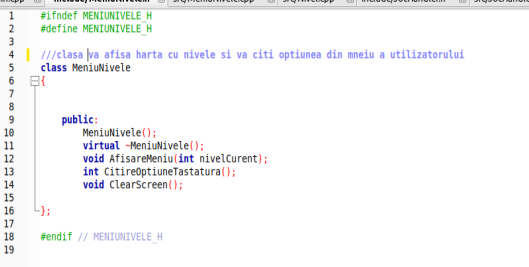


Figura 9. Header-ul pentru Clasa MeniuNivele

Aici avem 3 metode după cum urmează

* AfisareMeniu(int nivelCurent)= Funcția pentru afișarea meniului principal, cu nivelele
* CitireOptiuneTastatura() = Citim opțiunea din meniu a utilizatorului
* ClearScreen() = Ruleaza comanda de clear a terminalului

1. Clasa "Nivel": Această clasă va afișa spânzurătoarea cuvântul și decripția acestuia și va citii literele introduse de utilizator.

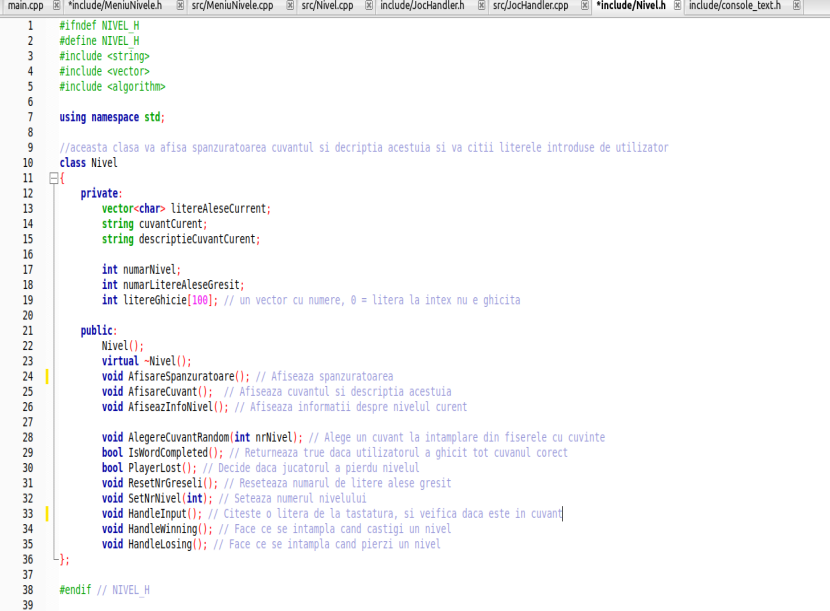


Figura 10. Header-ul pentru Clasa Nivel

Aici avem 11 metode după cum urmează

* AfisareSpanzuratoare() = afișează spânzuratoarea
* AfisareCuvant() = afișează cuvântul și descripția acestuia
* AfiseazInfoNivel() = afișează informații despre nivelul current
* AlegereCuvantRandom(int nrNivel) = alege un cuvânt la întâmplare din fișierele cu cuvinte
* IsWordCompleted() = returnează true dacă utilizatorul a ghicit tot cuvântul corect
* PlayerLost() = decide dacă jucătorul a pierdut nivelul
* ResetNrGreseli() = resetează numărul de litere alese greșit
* SetNrNivel(int) = setează numărul nivelului
* HandleInput() = citește o literă de la tastatură, și veifică dacă este în cuvânt
* HandleWinning() = face ceea ce se întâmplă când câștigi un nivel(trece la următorul nivel )
* HandleLosing() = face ceea ce se întâmplă când pierzi un nivel(dacă ești la nivelul 3 și ai epuizat toate încercările jocul se va relua de la nivelul 1)

De asemenea avem un fișier text predestinat definirii emoji-urilor apărute în meniu. Acest fișier se numește console\_text.h.

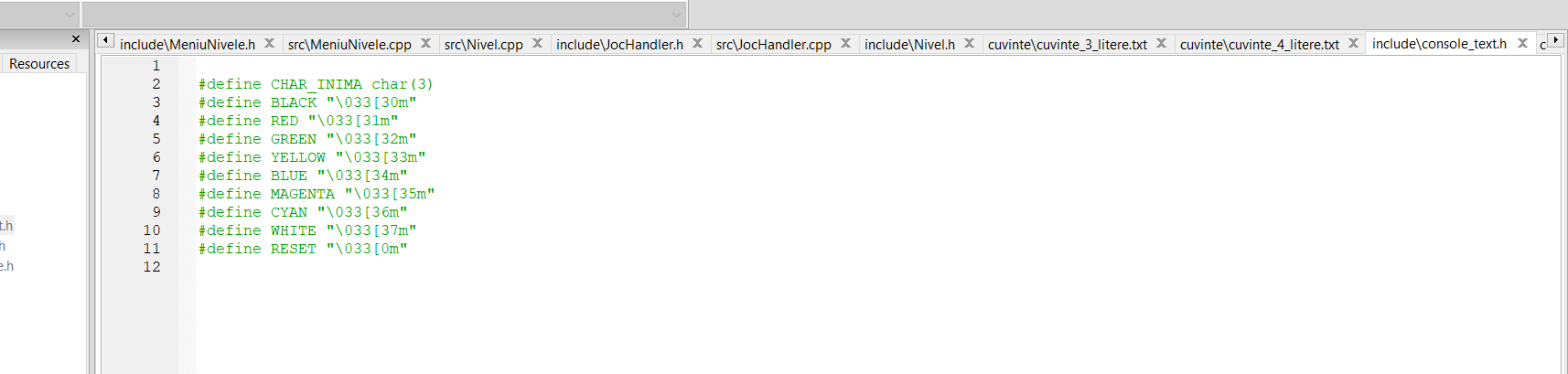


Figura 11. Fișierul pentru console\_text.h.

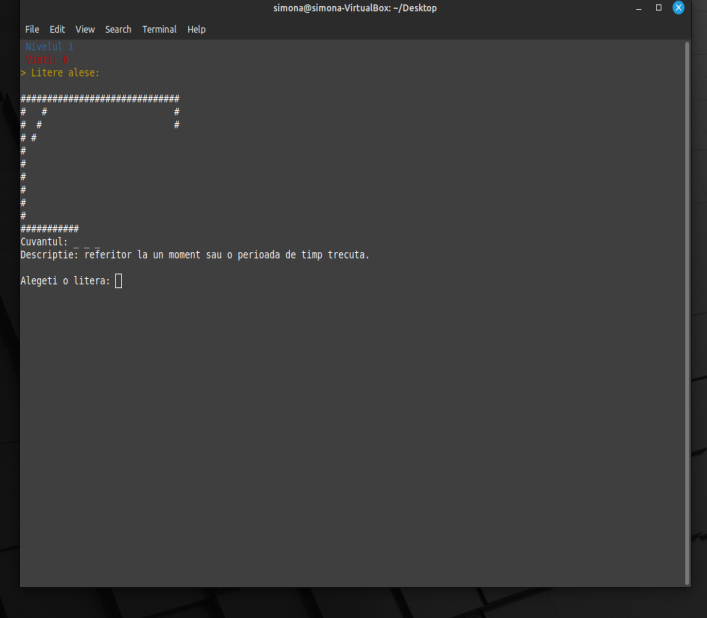


Figura 12. Afișarea în consola Linux la apăsarea opțiunii 1 și folosirea texturilor din fișierul console\_text.h

În ceea ce privește fișierele în proiect , mai sunt prezente alte 6 fișiere folosite pentru listele de cuvinte de la fiecare nivel de unde se va alege un cuvânt random alături de descripția sa. De exemplu fișierul de cuvinte pentru nivelul 1care conține numai cuvinte cu 3 litere prezentat în figura 13.

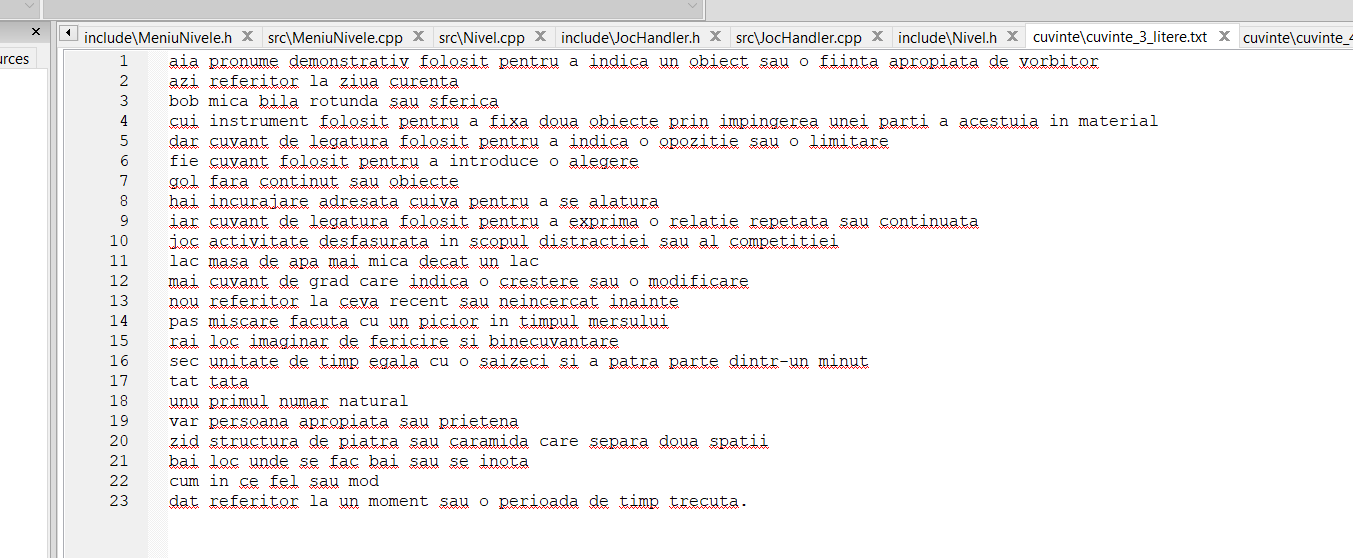


Figura 13. Fișierul de cuvinte pentru nivelul 1

# Cazuri excepționale , prezentarea opțiunii 2, 3 și 4

În cazul în care utilizatorul va pierde, in consolă se va afișa:

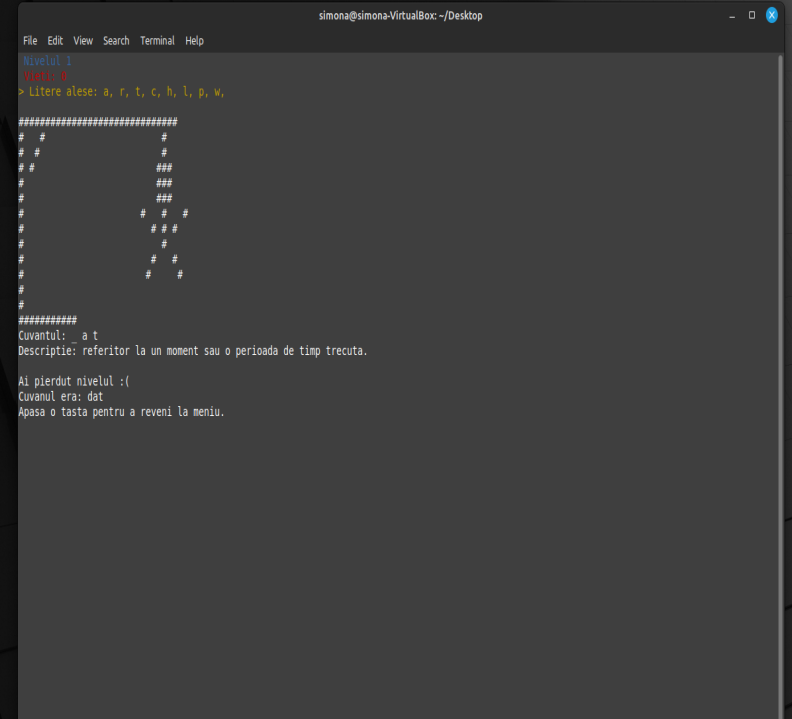


Figura 14. Mesajele afișate după ce utilizatorul a depășit numărul de încercări fără a ghici cuvântul

Se observă în colțul din stânga sus , apariția numărului de vieți (care este 0, utilizatorul a epuizat toate șansele),lista de cuvinte alese și nivelul la care este utilizatorul.

În cazul în care utilizatorul dorește să aleagă din nou o literă, nu va mai fi posibil:

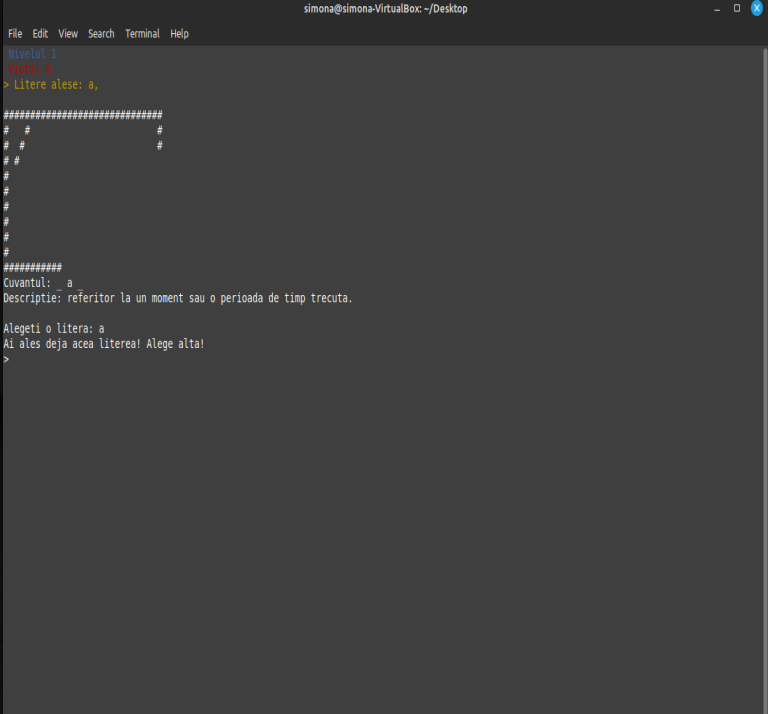


Figura 15. Interdicția de a alege de mai multe ori aceeași literă.

În cazul în care se va alege opțiunea 2 sau Help, se va afișa o descripție nu foarte lungă, dar care să ajute utilizatorul în folosirea programului creat :

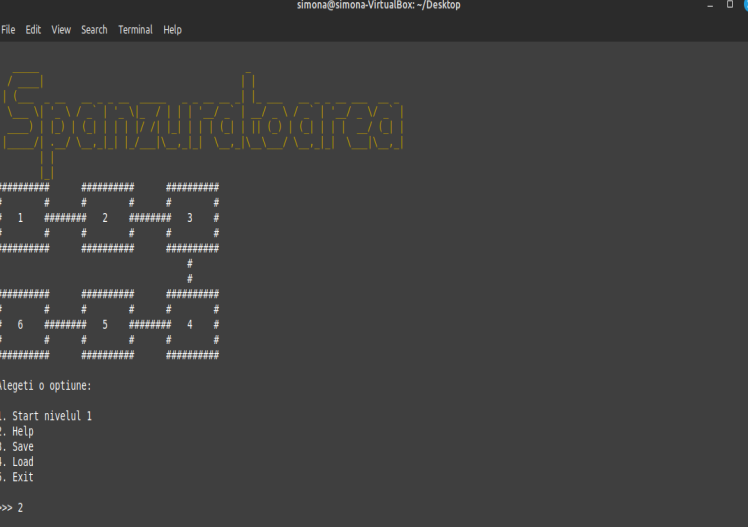


Figura 16. Alegerea opțiuni 2



Figura 12. Afișarea mesajului pentru opțiunea 2

În cazul în care se va alege opțiunea 3



Figura 13. Rezultatul opțiunii 3

În cazul în care alegem opțiunea 4 , indiferent dacă închidem consola sau nu, se va relua jocul de la nivelul salvat la opțiune 3:

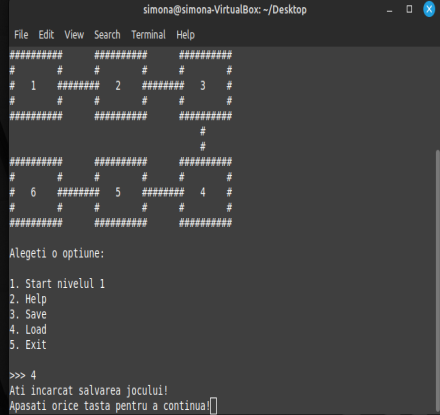


Figura 14. Rezultatul opțiunii 4