

# IN SEARCH OF TREASURES

**AUTOR: Danalache Emanuel**

**Grupa: 1210B**

**Povestea jocului:** Căpitanul Emory, un pirat temut de pe insula Sonovra, a descoperit o corabie părăsită în derivă. Pe aceasta, a găsit o hartă veche care indica existența unor comori neexplorate pe insula sa natală. Cu inima plină de nerăbdare și dorința arzătoare de aventură, Emory hotărăște să pornească în căutarea comorilor. Alături de echipajul său, căpitanul își pregătește corabia și pleacă către orizontul nesfârșit, hotărât să descopere secretele insulei și bogățiile ascunse. Astfel începe o călătorie plină de pericole și aventuri, în care Emory și camarazii săi trebuie să-și pună la încercare curajul și priceperea pentru a ajunge la destinație și a găsi comorile legendarului tezaur.

**Prezentarea jocului:** IN SEARCH OF TREASURES este un joc offline, single player 2D, acțiunea petrecându-se pe insula Sonovra. Scopul jocului este ca jucătorul să-l controlează pe Emory, personajul principal, de-a lungul a trei nivele de dificultate oarecum diferită în căutarea comorilor ascunse. Acesta trebuie să termine fiecare nivel într-un timp mai mic ca cel prestabilit astfel încât să câștige.

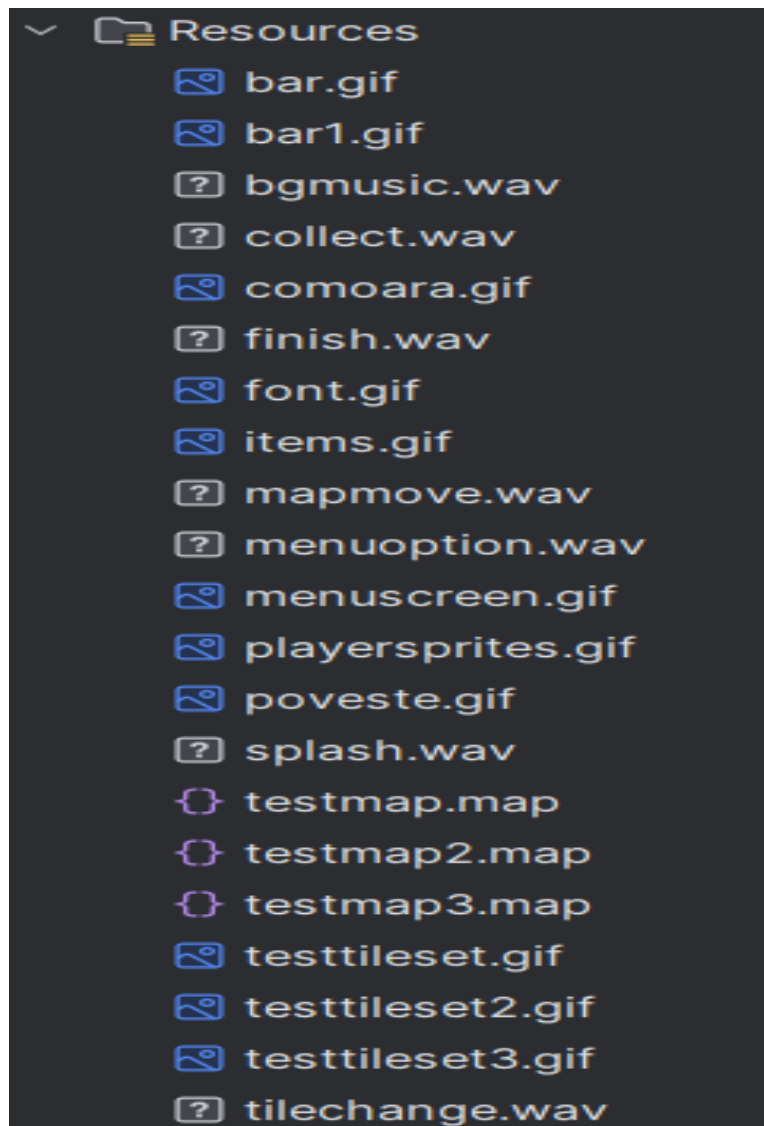
**Reguli joc:** Emory trebuie controlat pe toată harta nivelului, pentru a căuta și "colecta" comorile. Acesta va trebui să găsească la un moment dat o cheia și o bărcuța pentru a putea naviga pe apă, cât și pentru a deschide ușile încuiate. Dacă nu va găsi comorile într-un timp util, nivelul va fi pierdut și va trebui reluat din nou. La terminarea fiecărui nivel, playerul va primi un punctaj și dacă va termina toate cele 3 nivele va avea punctajul maxim.

## **Personajele jocului:**

- Emory: Un pirat iscusit, personajul principal. Este în căutarea comorilor îngropate pe insulă.

## Game sprite-uri:

Toate resursele de care m-am folosit:



- Harta:



- Comoara:



- Personaje:

Piratul Emory:



### **Mecanica jocului:**

- Jocul are trei dificultăți (high, medium, low), diferența dintre ele fiind timpul care este alocat pentru a câștiga. Timpul este crescător.
- Victoria se obține atunci când se găsesc toate comorile de pe harta nivelului respectiv, având un contor descrescător pentru comorile rămase. Intersecția jucătorului cu comorile duce la "colectarea" acestora.
- Tastele W, A, S, D sau săgețile sunt pentru a se deplasa caracterul. Tasta space este pentru a deschide ușile.

### **Harta:**

Componente pasive:

- "pereții": rama hărții;
- "podeaua": pe care merge Emory;

Componente active:

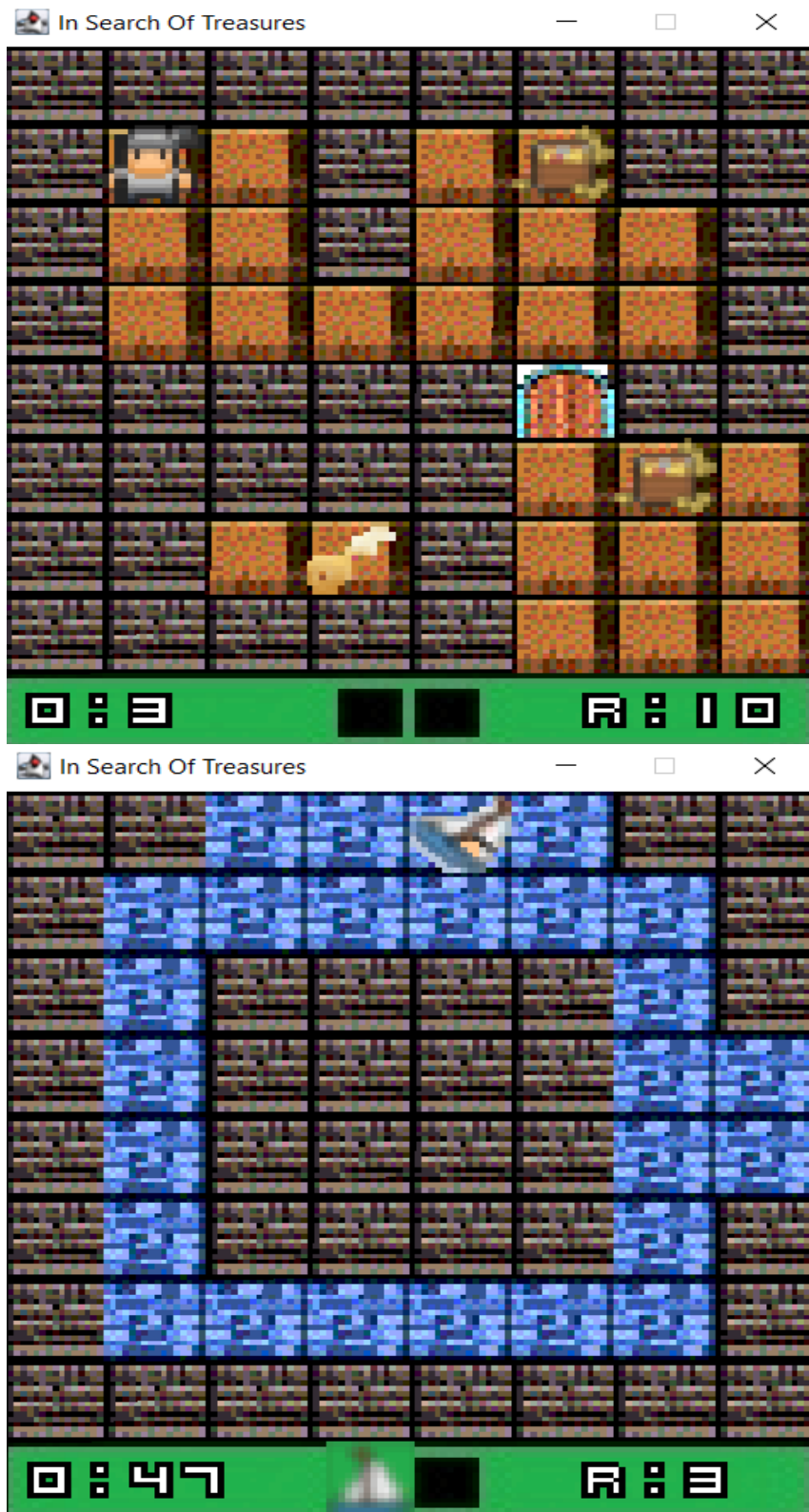
- cheia: pentru a deschide ușile (pentru nivelele 2,3);
- barca: pentru a merge pe apă (pentru nivelele 2,3);
- comorile: pe care trebuie să le colecteze

## Descriere niveluri:

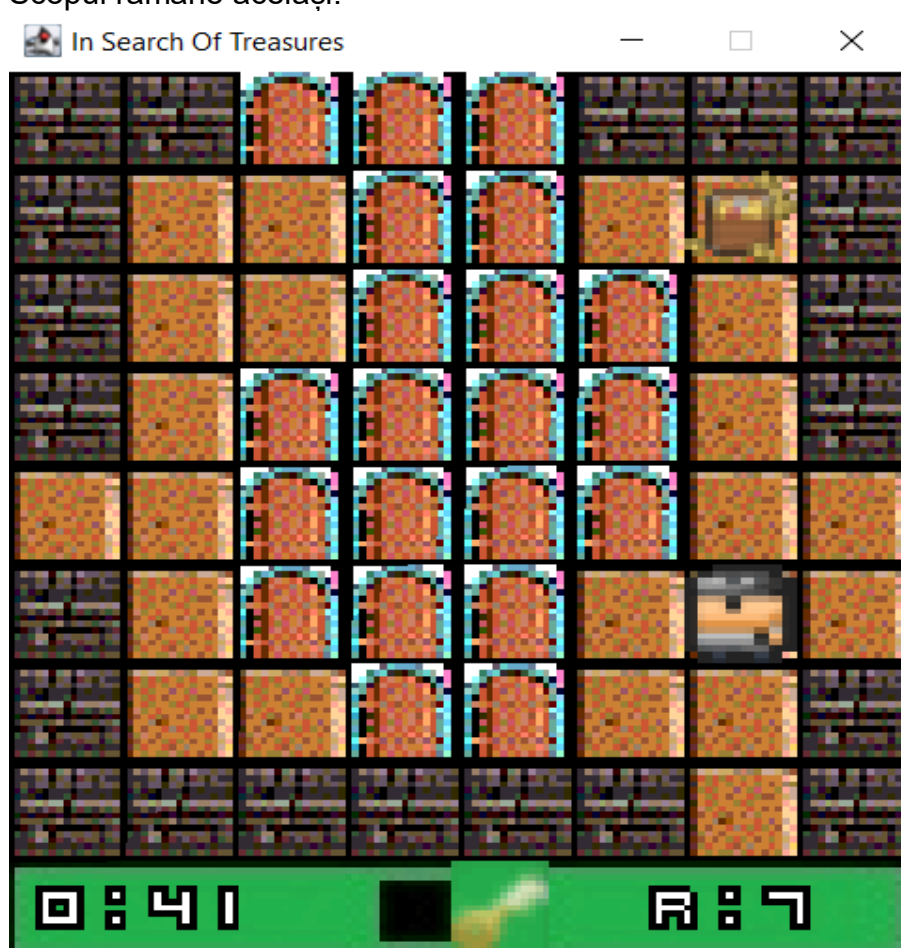
- **Nivelul 1:** Emory va parcurge toată harta în căutarea tuturor comorilor, când le va găsi pe toate va câștiga.

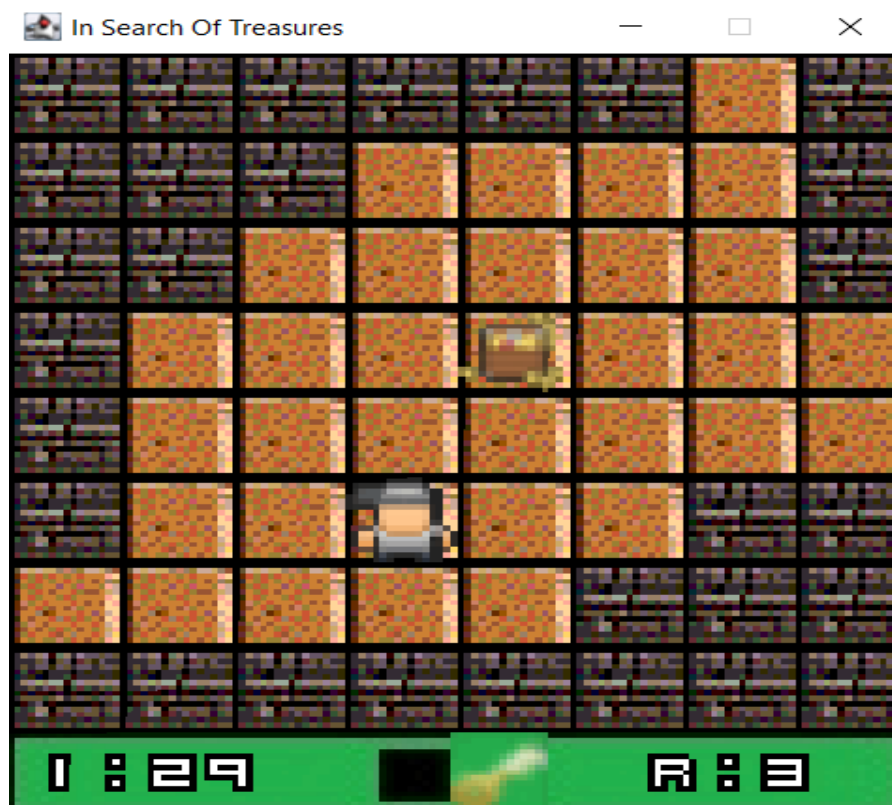


- **Nivelul 2:** în unele locuri de pe hartă îi va trebui bărcuța pentru a putea merge pe apă, dar va trebui mai întâi să o caute, apoi există unele locurile unde este o ușă și va trebui căutată cheia mai întâi pentru a intra pe ușă. Scopul rămâne același ca la nivelul 1.



- **Nivelul 3:** nivelul 1 + nivelul 2 + harta mai mare + mai multe comori de găsit.  
Scopul rămâne același.

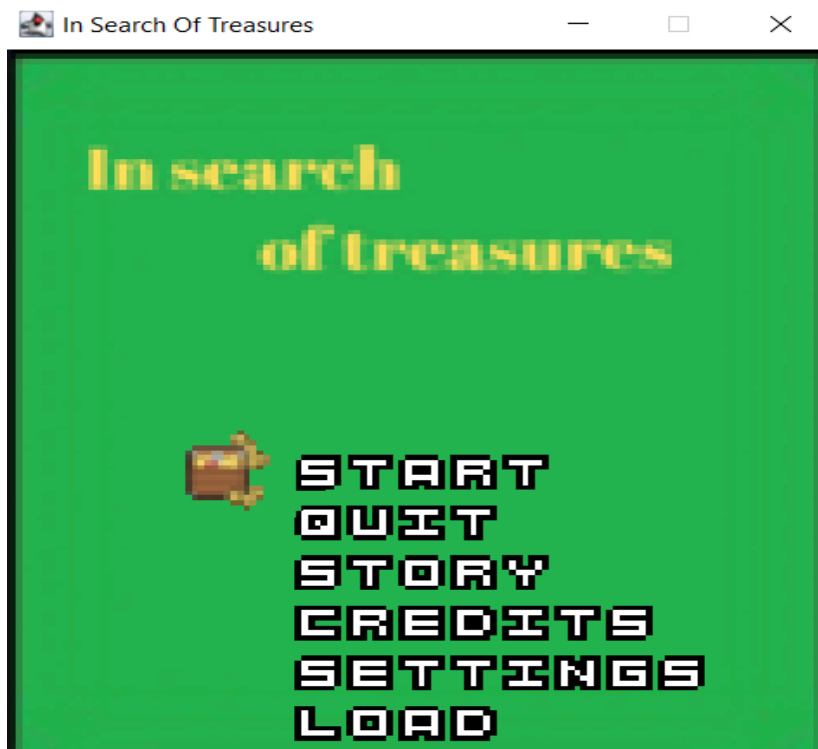




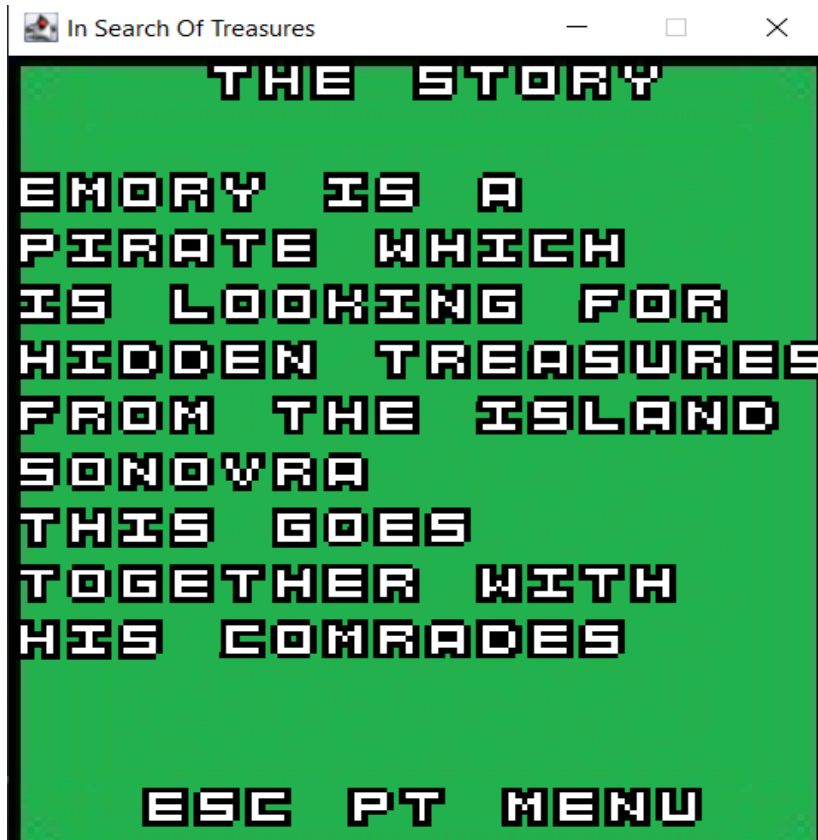
Meniul afișat pentru nivelul câștigat:



Meniul / Interacțiunea cu utilizatorul:

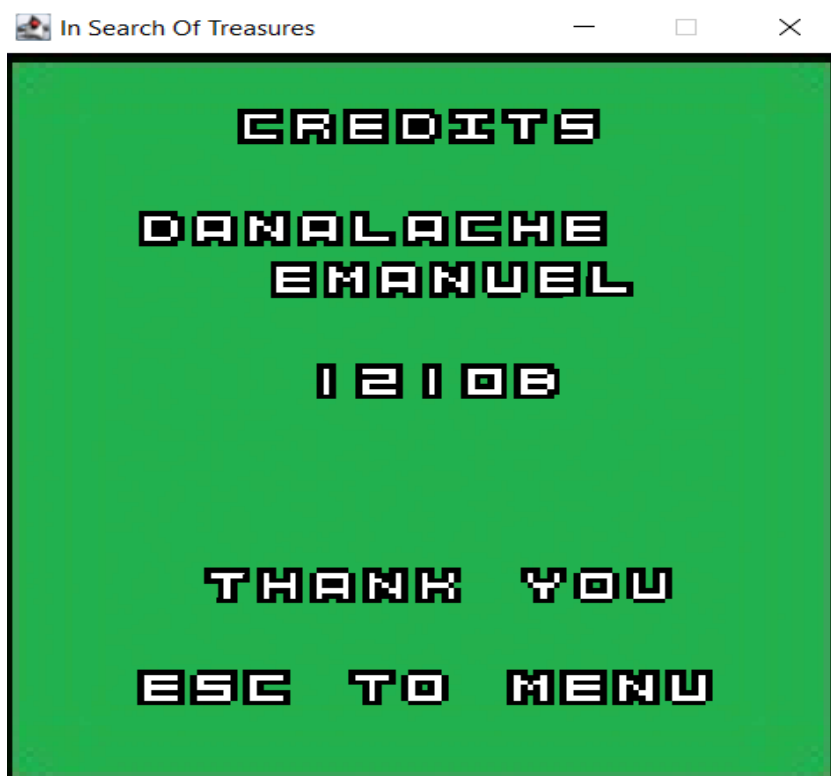


1. **Start:** începe efectiv nivelul 1
2. **Quit:** iese din joc
3. **Story:** povestea din spatele jocului

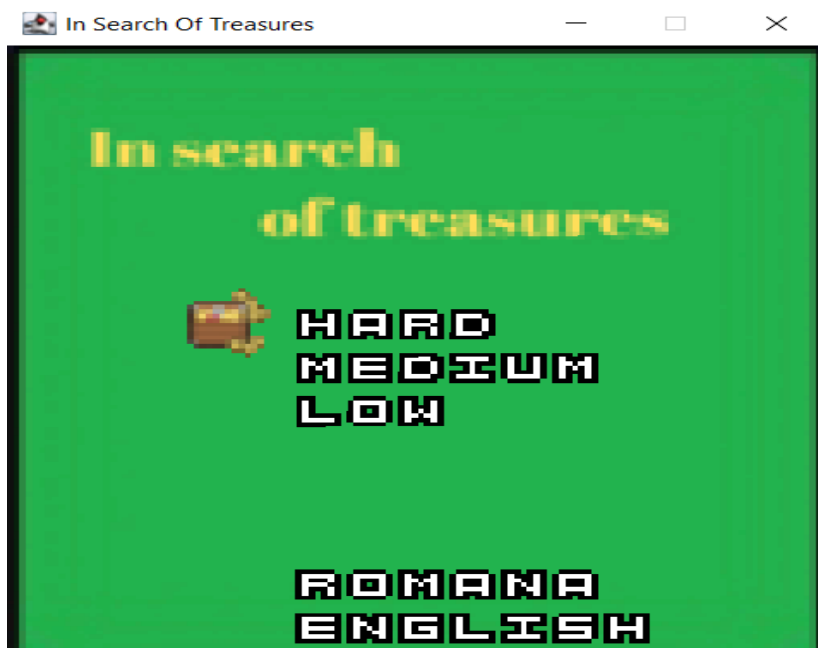




4. **Credits:** afișează numele și grupa creatorului jocului



5. **Settings:** setările pentru dificultate(hard, mediu, easy). Se poate schimba și limba din engleză în română.



**6. Load:** încarcă ultimul nivel care a fost finalizat

**Șabloane de proiectare / design pattern-uri folosite:**

- Singleton: asigură crearea unei singure instanțe dintr-o clasă și furnizează un punct global de acces la ea.
- Factory: definește o interfață pentru crearea unui obiect, dar lasă subclasele să decidă în privința instanțierii.

**Baza de date:**

Pentru conceperea acestui joc am folosit 3 baze de date.

1. Limba\_menu: conține 6 tabele din care extrag traduceri în română și engleză pentru toate stările jocului care au nevoie. Este folosită doar pentru a se citi din ea.

Table: LIMBA_POVESTE	Table: LIMBA_CREDITS																														
<table> <tr><th>Romana</th><th>Engleza</th></tr> <tr><td></td><td></td></tr> <tr><td>Povestea</td><td>The Story</td></tr> <tr><td>Emory es...</td><td>Emory is a</td></tr> <tr><td>un pirat c...</td><td>pirate wh...</td></tr> <tr><td>este în c...</td><td>is lookin...</td></tr> <tr><td>comorilor...</td><td>hidden tr...</td></tr> </table>	Romana	Engleza			Povestea	The Story	Emory es...	Emory is a	un pirat c...	pirate wh...	este în c...	is lookin...	comorilor...	hidden tr...	<table> <tr><th>Romana</th><th>Engleza</th></tr> <tr><td></td><td></td></tr> <tr><td>Credite</td><td>Credits</td></tr> <tr><td>Multumesc</td><td>Thank you</td></tr> <tr><td>ESC PT ...</td><td>ESC TO ...</td></tr> </table>	Romana	Engleza			Credite	Credits	Multumesc	Thank you	ESC PT ...	ESC TO ...						
Romana	Engleza																														
Povestea	The Story																														
Emory es...	Emory is a																														
un pirat c...	pirate wh...																														
este în c...	is lookin...																														
comorilor...	hidden tr...																														
Romana	Engleza																														
Credite	Credits																														
Multumesc	Thank you																														
ESC PT ...	ESC TO ...																														
<table> <tr><td>Table: LIMBA_CREDITS</td><td></td></tr> <tr><td></td><td></td></tr> <tr><td>Roman</td><td></td></tr> <tr><td></td><td></td></tr> <tr><td>Credite</td><td></td></tr> <tr><td>Multume</td><td></td></tr> <tr><td>ESC PT</td><td></td></tr> </table>	Table: LIMBA_CREDITS				Roman				Credite		Multume		ESC PT		<table> <tr><td>Table: LIMBA_MENU</td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td>Incepe</td><td>Start</td></tr> <tr><td>Iesire</td><td>Quit</td></tr> <tr><td>Poveste</td><td>Story</td></tr> <tr><td>Credite</td><td>Credits</td></tr> <tr><td>Setari</td><td>Settings</td></tr> </table>	Table: LIMBA_MENU						Incepe	Start	Iesire	Quit	Poveste	Story	Credite	Credits	Setari	Settings
Table: LIMBA_CREDITS																															
Roman																															
Credite																															
Multume																															
ESC PT																															
Table: LIMBA_MENU																															
Incepe	Start																														
Iesire	Quit																														
Poveste	Story																														
Credite	Credits																														
Setari	Settings																														

- Optiuni: conține o singură tabelă și are în componența ei sursele din care se va încărca conținutul jocului(hud, comoara, iteme, etc). Folosită în clasa Content, fiind incarcată de funcția load\_source. Doar se citește din aceasta bază de date.

Table: **OPTIUNI** ▼

LIST
/menusr...
/bar.gif
/Poveste....
/playersp...
/comoara...
/items.gif
/font.gif

3. Savegame: conține o singură tabelă. Se și citește și se scrie în această bază de date, ea având scopul de a reține ultimul level salvat la execuția jocului pentru a putea fi pornit din meniu.

Table: **SAVE\_GAME** ▼

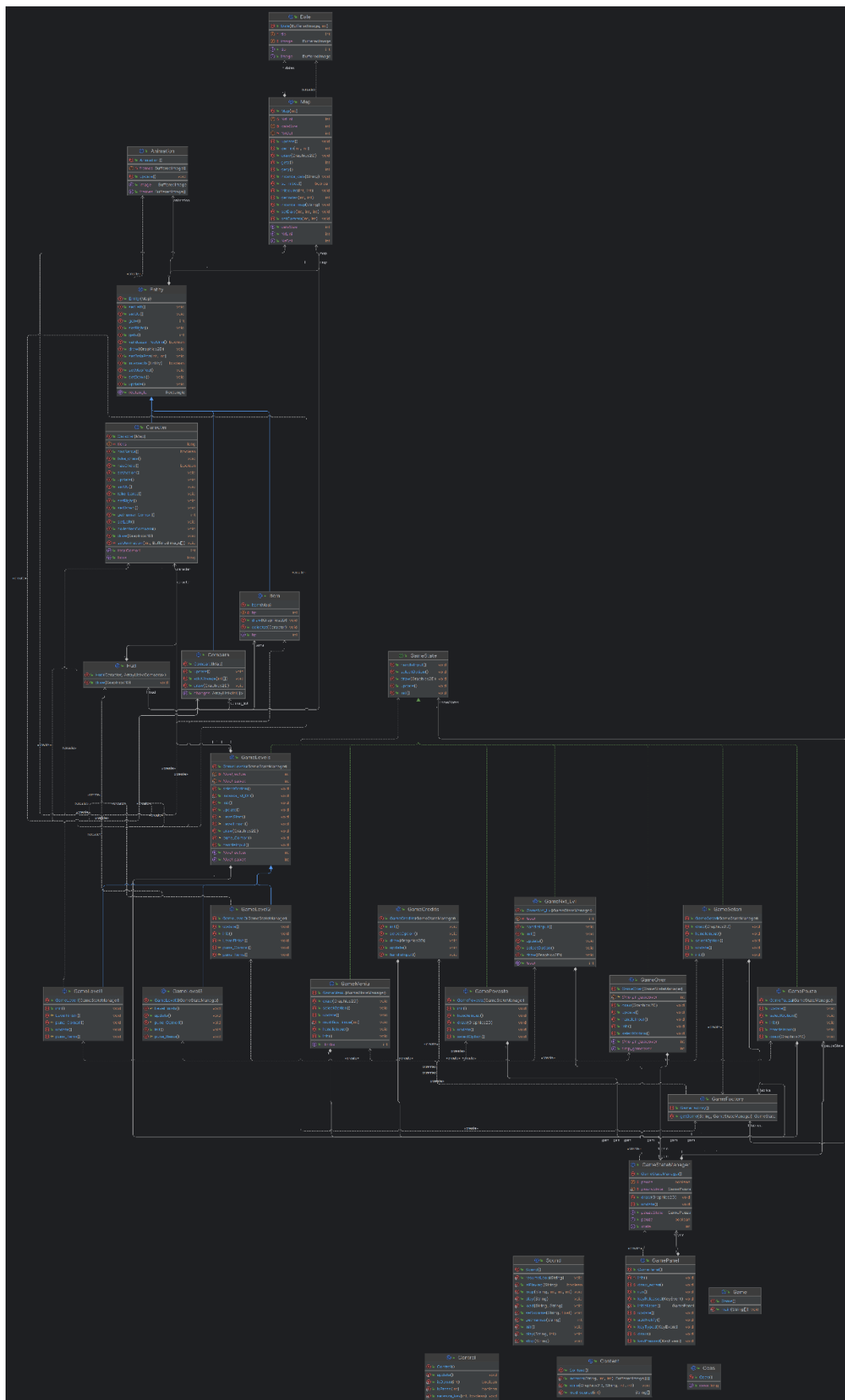
ID	LEVEL_...
1	3

## Documentația tehnică a proiectului

Componentele sistemului:

- Meniu
- GameOver
- Level1
- Level2
- Level3
- GamePauza
- GamePoveste
- GameCredits
- GameSetari

**Diagrame de clasa:**



Clasele **Item**, **Comoara** si **Caracter** vor moșteni clasa **Entity**. Caracter va fi instanțiată pentru player, comoara pentru comori, iar item pentru chei si barcă. Animation va seta modul în care se va realiza mișcarea comorilor si a playerului de-a lungul jocului. Playerul va da din picioare, iar comoara are 4 stări. Animația va avea un delay de 10, adică se va modifica o dată la 10 frame-uri. Cum jocul este setat la 60 FPS, asta înseamnă că vor fi 6 alternări ale animației într-o secundă.

### Diagrama pentru Game care implementeaza Factory:

Toate obiectele de tip Game vor fi instantiate prin intermediul **GameFactory**. Toate au la bază interfața **GameState**.

**GameLevels** este clasa de baza pentru cele trei nivele.

**GameNxt\_Lvl** este starea dintre nivele, care face managementul trecerii la nivelul urmator.

**GameOver** oferă jucătorului rank-ul obținut si timpul. Totodată, dacă jucătorul pierde jocul, tot această clasă este instanțiată.

**GameMenu** ne duce in meniul jocului. Aici vom avea setări, povestea jocului, posibilitatea de încărcare a jocului si cea de load. Totodată, tot aici vom putea închide jocul.

**GameSetari** ne permite să schimbăm dificultatea jocului, cât și limba în care acesta este prezentat. Sunt disponibile 2 limbi (română, engleză) si 3 dificultăți( high, medium, low) diferența dintre acestea fiind timpul pentru care se face game over.

**GameCredits** conține date despre mine, ca și dezvoltator. Toate aceste clase sunt legate una de alta prin faptul că ele prezintă de fapt stările jocului, astfel încat fiecare stare duce la altă stare prin anumite operatii.(de exemplu, pornirea unui joc va duce la trecerea din starea gameMenu la starea gameLvl1).

Clasele din **Manager** se ocupă cu managementul resurselor si a jocului.

**Map** va crea harta în care se desfășoară jocul(este citită dintr-un fișier text), **Sound** va adăuga jocului sunet, **Hud** va crea bara jocului care conține temporizator (care e generat de Ceas), numărul de comori rămase si itemele colectate.

### Diagramele pentru main:

Aici se implementează **SINGLETON**. Crează fereastra jocului, seteaza numărul de FPS-uri.

## Concluzii

### Respectarea cerințelor propuse:

1. **Proiect funcțional** – proiectul a fost testat pe parcursul dezvoltării pentru a mă asigura că nu întâmpin nicio eroare.
2. **Structura program si gestiunea erorilor** – erorile posibile la încărcarea datelor sunt „manageriate” prin utilizarea sintaxelor try-catch.
3. **3 nivele cu grad diferit de dificultate**: am implementat 3 nivele cu hărți mai complexe si mai mari decât cele precedente.
4. **Implementare coliziuni**: există coliziuni cu comorile, atunci când se face colectarea lor, atât cât și prin crearea dalelor de tip normal/blocat prin care se poate trece, respectiv nu se poate trece.
5. **Scor**: scorul este reprezentat de numărul de comori rămase. Este descrescător, iar în momentul în care se ajunge la 0 se consideră jocul câștigat. Totodată, rank-ul generat la finalul jocului cât și timpul obținut pot fi considerate scoruri.
6. **Strategia de joc**: strategia aleasă de mine a fost cea prezentată mai sus, adică metoda de câștig este colectarea tuturor comorilor, iar cea de pierdere este depășirea timpului setat pe nivel pentru a-l finaliza. La expirarea timpului, jocul generează game over.
7. **Setări**: limba + dificultate(un timp limită mai mic înseamnă un nivel mai greu)
8. **Save and load**: prezent.
9. **Cod comentat**: prezent