

## Titlu proiect: **The Garfield Escape**

Autor: **SFICHI ALIN-IONUȚ**

Grupa: 1208A





### **Povestea jocului:**

Pisoiul Garfield (poreclit așa pentru că avea obiceiul de a fi răsfățat și de a primi totul pe tavă) trebuie să-și convingă familia că se poate descurca și singur să poată avea grijă de el. Pentru a demonstra capacitățile sale, acesta trebuie să reușească să adune un număr de peștișori într-un timp limită și să ajungă cu aceștia înapoi acasă. Deși nu i se pare o misiune dificilă, Garfield își dă seama că nu e de glumă atunci când află că traseul pe care trebuie să-l parcurgă are forma unui labirint. Pentru că e doar un pisoi, el are nevoie de puțin ajutor pentru a-și îndeplini misiune și a-și recâștiga încrederea familiei. Ajută-l pe Garfield să le dovedească celorlalți că el poate fi și independent.

### **Prezentare:**

Există un singur jucător care trebuie să adune recompensele, să se ferească de inamici și de obstacole și să ajungă la capătul labirintului înainte ca timpul să se scurgă, pentru a putea trece la nivelul următor. Pot exista mai multe trasee prin care se poate parcurge labirintul. Jucătorul va trebui să aleagă drumul optim, pentru a reuși să rezolve toate provocările.

### **Reguli joc:**

Jocul implică deplasarea pisoiului în interiorul labirintului, iar pentru asta se vor folosi tastele direcționale: , , , , pentru deplasare sus, jos, stanga, dreapta.

Jucatorul este capturat dacă intră pe teritoriul inamicului și este atins de acesta.

În cazul obstacolelor (capcanelor) de pe traseu, odată atinse, va fi redusă semnificativ viteza de deplasare a pisoiului, acest fapt adăugând un grad în plus de dificultate.

Câștigarea jocului presupune parcurgerea tuturor nivelelor, iar câștigarea fiecărui nivel constă în parcurgerea labirintului și colectarea numărului de recompense indicat în timpul alocat.

Înfrângerea va fi declarată fie atunci când se scurge timpul înainte ca jucatorul să ajungă la finish, fie atunci când este capturat de inamic. După o înfrângere, jocul va putea fi reluat de la linia de start a primului nivel.

Mutările imposibile sunt reprezentate de deplasarea pe zonele alocate zidurilor labirintului.

### **Personajele jocului:**

- **Garfield** este protagonistul. El este descris ca fiind pisoiul ambițios, ce își dorește din toată inima să dărâme orice prejudecată și să-și demonstreze aptitudinile. Acesta poate lua mai multe înfățișări, în funcție de preferințele jucătorului.

- **Odie** (inamicul) este al doilea personaj, acesta fiind cățelul care așteaptă cu nerăbdare să prindă pisoiul.

## **Tabla de joc:**

### **Componente active:**

- peștii (recompensele ce trebuie adunate, aceștia vor dispărea de pe table de joc atunci când personajul le va atinge și vor adăuga 3 secunde în plus la timp)

### **Componente pasive:**

- iarba / suprafața pe care se mișcă jucătorul (fără coliziuni)
- ziduri / pereții labirintului (cu coliziuni, jucătorul “se ciocnește” de ziduri și nu își poate continua deplasarea în direcția respectivă)
- capcane (pânze de păianjen, existente pe tabla de joc începând cu nivelul 2, acestea vor face ca în momentul când sunt atinse de jucător, viteza de deplasare a acestuia să scadă considerabil.

### **Structura tablei de joc:**

- vedere ortogonală
- labirintul este situat pe o tablă în forma unui dreptunghi și va conține dale reprezentative pentru zidurile ce alcătuiesc pereții labirintului, respectiv pentru zona în care se deplasează personajul (iarba). De asemenea , vor exista dale pentru recompense , cat și pentru capcanele întâlnite de-a lungul traseului.
- în partea de sus a tablei de joc va fi afișat nivelul la care s-a ajuns, timpul rămas pentru finalizarea probei , numărul peștișorilor adunați precum și butoanele de opțiuni.



### Mecanica jocului:

- Scopul protagonistului este de a parcurge labirintul în timpul alocat, adunând recompensele necesare de pe traseu și evitând capcanele și inamicii.
- Abilitățile sale constau în deplasarea pe table de joc (în labirint) și colectarea obiectelor (peștișorilor) necesare
- Nu este posibilă deplasarea în zonele alocate pereților labirintului
- Este impusă colectarea unui număr de obiecte existente, în caz contrar trecerea la un alt nivel fiind imposibilă
- Obstacolele întâlnite pot fi evitate prin simpla alegere a unui traseu spre finish care să nu le conțină. Acestea sunt :
  - **de natură pasivă** (pânză de păianjen) , ele având rolul de a reduce din timpul alocat nivelului, fără a influența în vreun fel deplasarea caracterului.
  - **de natura activă** (cățelul Odie): odată ce protagonistul intră pe teritoriul cățelului și îl atinge, acesta va fi prins, iar nivelul va trebui reluat.
- Principala resursă ce trebuie gestionată este timpul; există o durată de timp impusă pentru fiecare nivel (acest fapt constituind și una din dificultăți)
- Nivelele sunt parcurse succesiv: nu există posibilitatea alegerii

### Game sprite:



### Descrierea fiecărui nivel:

- **NIVEL 1:** jucătorul trebuie să adune peștișorilor și să ajungă la capătul labirintului înainte de scurgerea timpului
- **NIVEL 2:** pe lângă peștișorii din labirint, vor mai apărea unele capcane ce vor trebui evitate ; în cazul în care jucătorul se deplasează pe zona alocată capcanei, viteza acestuia va scădea iar șansele de a ajunge la capătul labirintului vor fi tot mai mici
- **NIVEL 3:** într-o zonă a traseului va exista și un inamic: atunci când protagonistul intră pe teritoriul acestuia și îl atinge, va fi capturat

### **Descrierea meniului:**

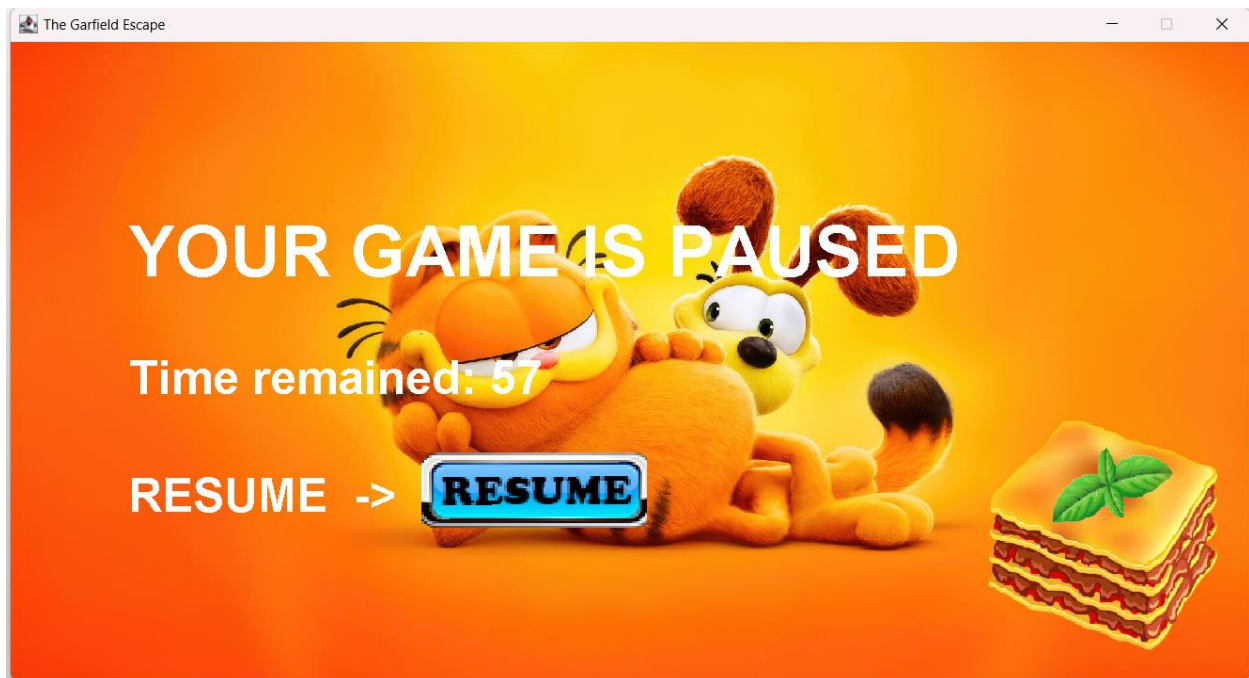
- Va exista butonul 'START' , care va fi apăsat pentru începerea jocului
- Butonul 'LOAD' va încărca ultima salvare a jocului
- Opțiunea 'HELP' va afișa pe ecran regulile jocului (nivelului la care s-a ajuns)
- Va exista posibilitatea de a ieși din joc prin intermediul opțiunii 'EXIT', în acel moment salvându-se progresul jocului în baza de date.
- Opțiunea 'PAUSE' va oferi posibilitatea de a opri jocul până la momentul când se va debifa această opțiune
- Va exista opțiunea de a activa sau dezactiva muzica de fundal, acest lucru realizându-se în meniul de setări ( 'SETTINGS' )

### **Implementare:**

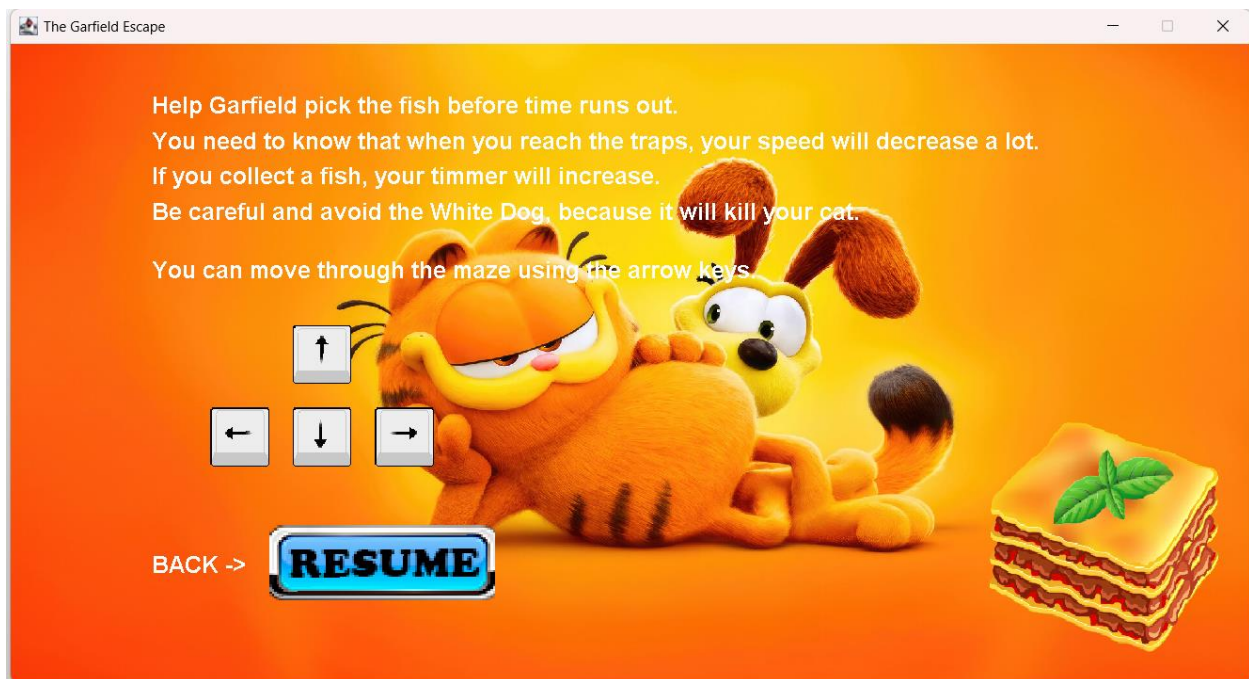


- Meniu Principal -

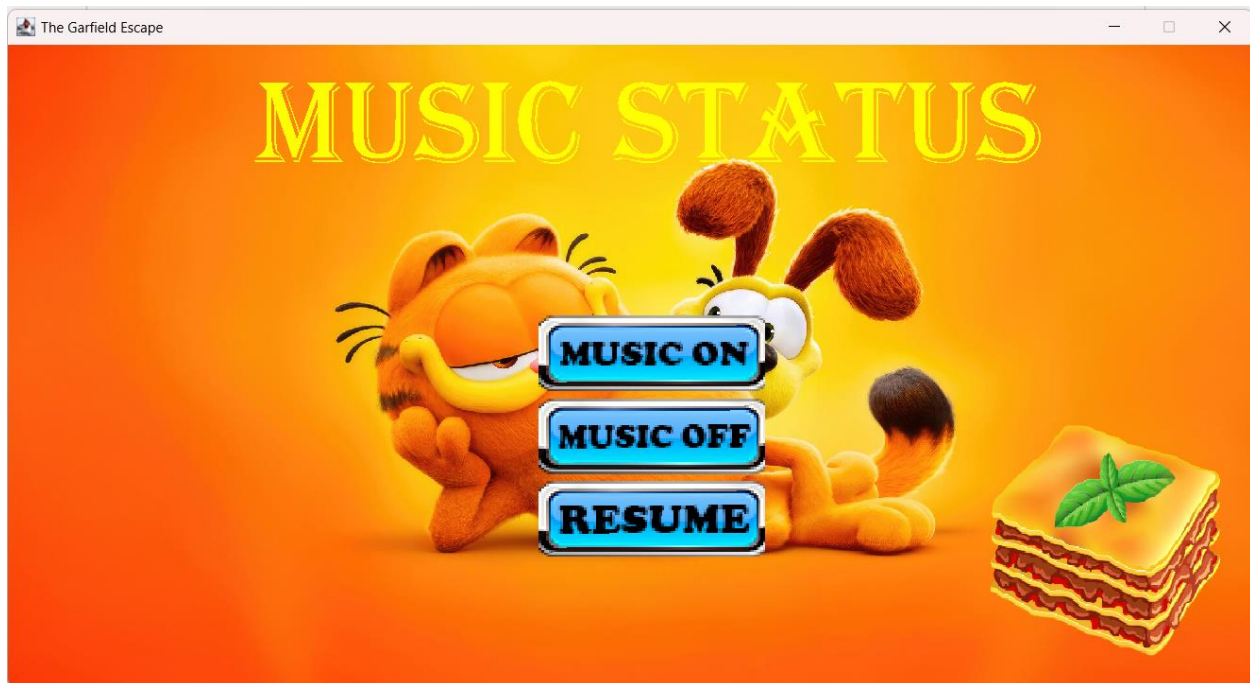




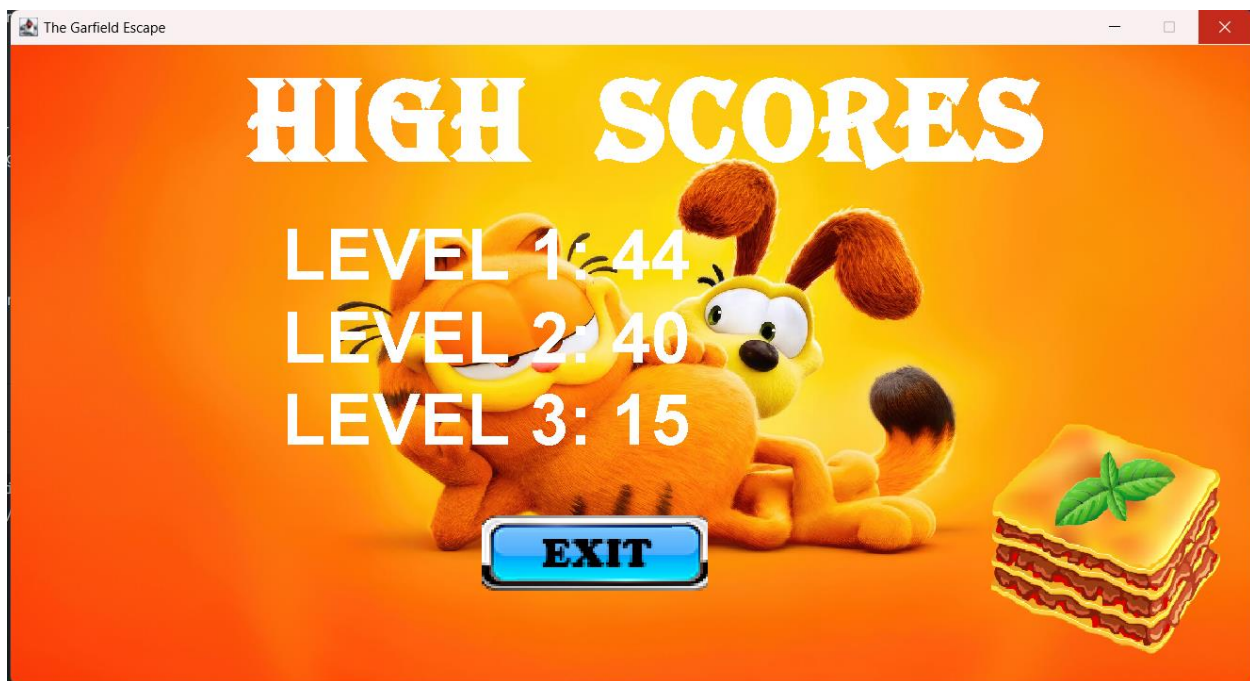
- Ecranul de pauză –



- Ecranul de ajutor –



- Setările -

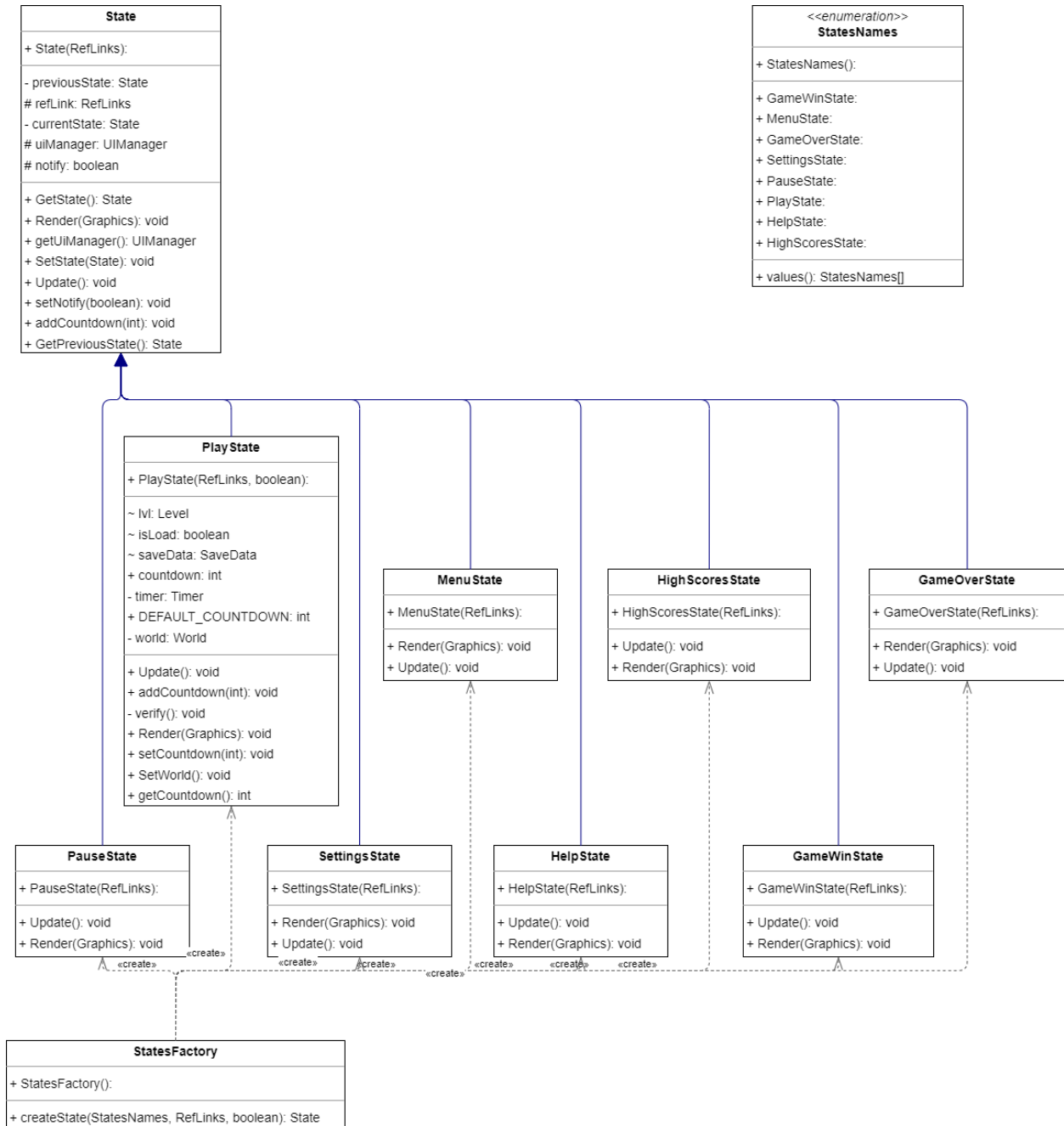


- Cele mai bune scoruri -

# 1. Șabloane de proiectare

Au fost utilizate 3 șabloane de proiectare ( Abstract Factory, Singleton și Composite ):

- Abstract Factory este utilizat pentru implementarea noțiunii de fabrică de States : în funcție de specificațiile trimise la Factory, sunt generate Stăro corespunzătoare( tipul lor, dacă se încarcă o salvare).



- Singleton este utilizat atât pentru noțiunea de ‘NIVEL’ al jocului , cât și pentru clasa care implementează o bază de date SQL:

a) LEVEL : Singleton este utilizat în ideea că nu pot exista 2 sau mai multe niveluri care sa fie jucate simultan; de aceea, pe tot parcursul jocului există o singură instanță a clasei Level.

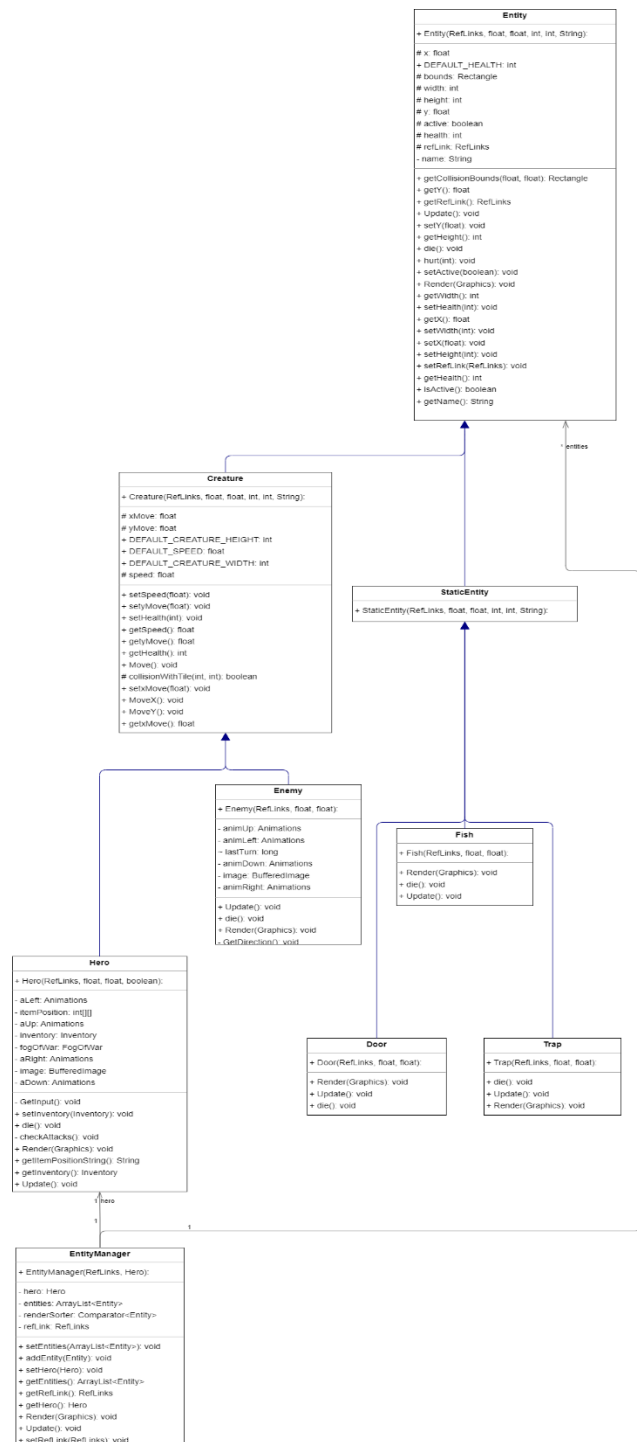
Level
- Level():
- instance: Level - levelNr: int - changeLevel: boolean
+ getInstance(): Level + incLevel(): void + getLevelNr(): int + Render(Graphics): void + setLevel(int): void + setChangeLevel(boolean): void + isChangeLevel(): boolean

b) SQL: Singleton este utilizat deoarece este necesară o singură clasă pentru gestionarea bazei de date și, implicit, o singură bază de date pentru întregul joc.

SQL
- SQL():
- c: Connection - instance: SQL
+ addAudioStatus(String, int): void + addScore(int, int): void + getMusicStatus(): int + getSoundStatus(): int + getInstance(): SQL + closeConnection(): void + deleteAUDIO(): void + getPlayStatus(): SaveData + addPlayStatus(int, int, int, int, int, int, String, String, String): void + getHighScore(int): int + deleteSCORES(): void + isEmpty(): boolean



- Șablonul composite: este utilizat în diferitele clase manager, cum ar fi clasa EntityManager, care conține o colecție de Entități Entity (de exemplu Hero, Enemy, Button, Door, care sunt clase ce extind clasa Entity), aceasta implementând operații comune (Update(), Render()) pentru toate elementele care fac parte dintr-o instanță a clasei.



```

public void Update(){
    Iterator<Entity> it= entities.iterator();
    while(it.hasNext())
    {
        Entity e=it.next();
        e.Update();

        /// daca viata entitatii respective s-a
        if(!e.isActive())
            it.remove();
    }

    entities.sort(renderSorter);
}

/*! \fn public void Render(Graphics g)
    \brief Desenarea starii curente starii c
    \param g Contextul grafic in care se real

public void Render(Graphics g){

    for(Entity e:entities){
        e.Render(g);
    }
}

```

## 2. Baza de Date

A fost implementată o bază de date pentru stocarea Stării curente a jocului ( dacă este apăsat butonul EXIT în momentul gameplay-ului propriu zis), pentru încărcarea jocului din acea bază de date, cât și pentru afișarea celor mai bune recorduri (scores) din fiecare nivel, în starea HIGH SCORES.

### - Save/ load game:

Strategia de joc presupune nevoia unui anumit timp continuu disponibil pentru ca jucătorul să poată trece peste toate obstacolele și să colecteze toate obiectele necesare, de aici apare și nevoia salvării stării jocului la orice moment de timp și reluarea acestuia. Pentru a rezolva această problemă jucătorul trebuie să apese doar butonul EXIT din partea de sus a ferestrei de joc iar pogresul acestuia v-a fi salvat automat în baza de date. Pentru a relua jocul de la acel progress salvat anterior nu trebuie decât să apese butonul LOAD din meniul jocului.

### - High scores:

Pentru a vedea cele mai bune scoruri din joc, utilizatorul poate accesa secțiunea HIGH SCORES din meniul jocului, acolo unde vor fi afișate cele mai mari scoruri obținute vreodată pentru fiecare nivel în parte.





















### - Settings:

În funcție de preferințele utilizatorului, se poate juca având o anumită muzică de fundal. Aceasta va putea fi activată sau dezactivată din starea ‘Settings’ a jocului. Modificările aduse asupra acestui aspect vor fi salvate de fiecare dată în baza de date. La reluarea jocului se va extrage din baza de date ultima modificare făcută și se va menține și în jocul actual.

SQL
- SQL();
- c: Connection - instance: SQL
+ addAudioStatus(String, int): void + addScore(int, int): void + getMusicStatus(): int + getSoundStatus(): int + getInstance(): SQL + closeConnection(): void + deleteAUDIO(): void + getPlayStatus(): SaveData + addPlayStatus(int, int, int, int, int, int, String, String, String): void + getHighScore(int): int + deleteSCORES(): void + isEmpty(): boolean

Clasa care implementează baza de dat

Structura tabelor bazei de date este următoarea:

Name	Type
▼  Tables (3)	
▼  SAVE	
 ID	TEXT
 LEVEL	INT
 POSX	INT
 POSY	INT
 ENEMYPOSX	INT
 ENEMYPOSY	INT
 COUNTER	INT
 FISHPOSITION	TEXT
 TRAPPOSITION	TEXT
 ITEMPOSITION	TEXT
▼  SCORING	
 ID	TEXT
 LEVEL	INT
 SCORE	INT
▼  SETTINGS	
 ID	TEXT
 AUDIO	TEXT
 ON_OF	TEXT

#### **Tabela SAVE:**

În această tabelă salvăm toate informațiile necesare reluării jocului de la un anumit moment cu progresul aferent. Avem 6 coloane de tip întreg pentru: nivelul ajuns, poziția pe axa X a jucătorului, poziția pe axa Y a jucătorului, poziția pe axa X a inamicului, poziția pe axa Y a inamicului și contorul de timp. În plus mai avem și 3 coloane de tip text în care salvăm poziția peștișorilor de pe hartă, poziția capcanelor de pe hartă și poziția itemelor colectate de pe hartă.

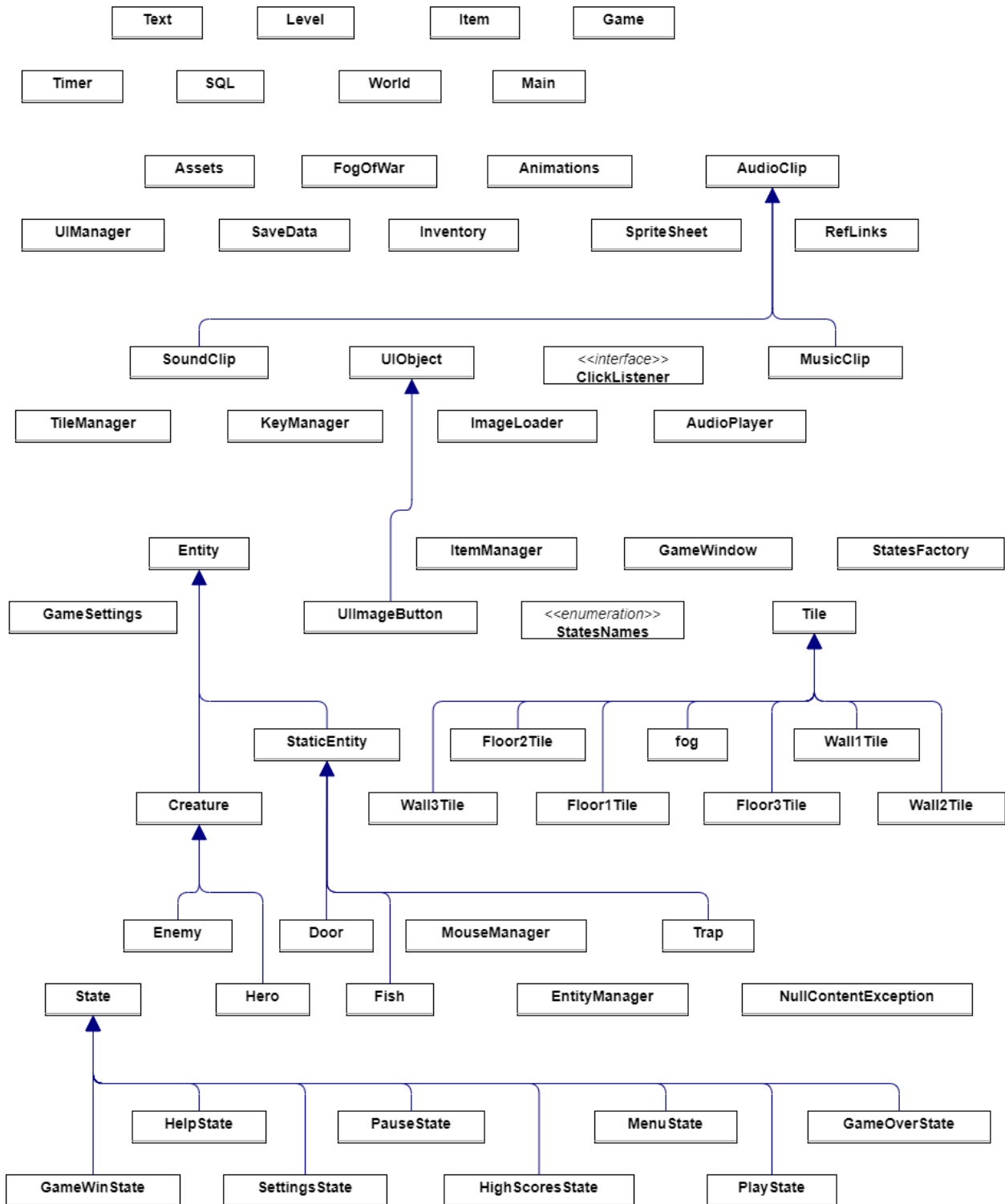
#### **Tabela SCORING:**

În această tabelă salvăm scorurilor obținute la fiecare nivel completat. Avem 2 coloane de tip întreg: numărul nivelului și scorul obținut pe baza timpului rămas.

#### **Tabela SETTINGS:**

În această tabelă setările legate de melodiile care se redau în joc. Avem 2 coloane de tip text: una pentru tipul de audio redat și una pentru oprit/pornit.

### 3. Diagrama de clase ce ilustrează moștenirea



-Diagramele de clasă par complicate deoarece conține clasa Reflink, o instanță a căreia fiind dată drept parametru în aproape toate metodele, motiv pentru care nu am afișat și dependencies

## **Scurtă explicație a funcționalității claselor:**

### **Pachetul Audio:**

**AudioClip:** Clasă abstractă care definește un clip audio printr-o referință la un obiect Clip și conține o metodă abstractă Update pentru actualizarea volumului în funcție de setările jocului.

**AudioPlayer:** Gestionează și controlează redarea clipurilor audio din joc, permițând încărcarea și redarea muzicii de fundal și a sunetelor, actualizând lista de clipuri audio.

**MusicClip:** Extinde AudioClip pentru a reprezenta muzica de fundal, oferind metode pentru a porni și opri redarea și pentru a actualiza starea în funcție de setările de muzică.

**SoundClip:** Extinde AudioClip pentru a reprezenta sunetele jocului, oferind metode pentru a activa sunetele și pentru a actualiza starea în funcție de setările de sunet.

### **Pachetul Entities:**

**Entity:** Clasă abstractă care definește o entitate generică din joc, extinsă de alte clase care definesc entități specifice, cum ar fi creaturi sau obiecte statice. Conține metode pentru actualizarea și desenarea stării curente a entității, precum și pentru gestionarea coliziunilor și a altor atribute.

**Creatures:** Extinde Entity și definește noțiunea de creatură sau individ care se poate deplasa în joc, conținând metode pentru gestionarea mișcării și detectarea coliziunilor.

**StaticEntity:** Extinde Entity și implementează noțiunea de entitate statică, cum ar fi cheia sau ușa.

**Hero:** Extinde DynamicEntity și reprezintă eroul controlat de utilizator, gestionând input-ul de la tastatură pentru a actualiza poziția eroului și animațiile pentru diferitele sale acțiuni.

**Enemy:** Extinde DynamicEntity și reprezintă inamicul din joc, având un comportament de mișcare predeterminat și detectând coliziunile cu eroul.

**Door:** Extinde StaticEntity și implementează comportamentul pentru ușă, permițând trecerea la următorul nivel atunci când eroul colectează cheia.

**Fish:** Extinde StaticEntity și implementează comportamentul pentru un peștișor, permițând eroului să îl colecteze.

**Trap:** Extinde StaticEntity și implementează comportamentul pentru unei capcane, scăzând viteza eroului.

### **Pachetul Exceptions:**

**NullContentException:** Definește o excepție personalizată pentru a semnala că un fișier are un conținut nul. Constructorul cu parametri primește un mesaj de eroare și îl transmite clasei de bază Exception.

### **Pachetul GameWindow:**



GameWindow: Implementează o fereastră pentru joc, oferind funcționalități precum crearea ferestrei, setarea dimensiunilor, afișarea în centrul ecranului și furnizarea unui canvas pe care se poate desena.

### **Pachetul Graphics:**

Animations: Definește și gestionează animațiile pentru deplasarea personajelor în joc, stocând viteza animației, imaginile specifice și gestionând actualizarea și returnarea frame-urilor curente.

Assets: Responsabilă pentru încărcarea tuturor elementelor grafice necesare jocului, utilizând o clasă auxiliară SpriteSheet pentru a încărca și croi imaginile necesare.

ImageLoader: Utilitar simplu pentru încărcarea imaginilor din fișierele sursă, folosind ImageIO.

SpriteSheet: Reprezintă o foaie de dale (sprite sheet) și furnizează metode pentru a croi imagini individuale (dale) din sprite sheet-ul principal.

FogOfWar: Reprezintă grafica specifică mecanicii de FogOfWar.

Text: Responsabilă pentru afișarea textului pe ecran într-o anumită poziție și culoare.

### **Pachetul Input:**

MouseManager: Gestionează intrările de la mouse, detectând click-urile și mișcările cursorului.

KeyManager: Gestionează intrările de la tastatură, detectând apăsările și eliberările de taste.

### **Pachetul Inventory:**

Pachetul Inventory gestionează obiectele colectate de către erou pe parcursul jocului. Conține clasa Inventory care are responsabilitatea de a adăuga, actualiza și reda elementele din inventarul eroului.

### **Pachetul Items:**

Pachetul Items gestionează obiectele colectabile din joc, definind clasa Item pentru fiecare obiect individual și clasa ItemManager pentru a gestiona colectiv aceste obiecte.

### **Pachetul Level:**

Level: Implementează conceptul de nivel în joc folosind Singleton Pattern, asigurând o singură instanță a nivelului la un moment dat. Gestionează numărul nivelului curent, permite schimbarea nivelului și oferă metode pentru incrementarea și redarea grafică a numărului nivelului. Constructorul este privat pentru a preveni crearea de multiple instanțe, iar metoda getInstance returnează instanța unică a clasei.

### **Pachetul SaveData:**

Pachetul SaveData se ocupă cu gestionarea datelor salvate ale jocului, inclusiv pozițiile jucătorului, inamicilor și obiectelor pe hartă. Acesta include clasa SaveData, care permite stocarea și recuperarea informațiilor despre starea curentă a jocului.

Clasa SaveData oferă o structură pentru salvarea și încărcarea stării jocului, incluzând nivelul curent, pozițiile jucătorului și ale inamicilor, numărul de obiecte colectate și pozițiile diferitelor obiecte pe hartă.

### **Pachetul Settings:**

Settings: Implementează setările jocului, oferind opțiuni pentru activarea sau dezactivarea muzicii și sunetelor. Include metode pentru a obține și seta starea curentă a muzicii și sunetelor.

### **Pachetul SQLite:**

SQL: Gestionează baza de date a jocului folosind Singleton Pattern pentru a asigura o singură instanță a clasei. Constructorul privat stabilește conexiunea cu baza de date și creează tabelele necesare pentru stocarea scorurilor și a setărilor jocului. Metodele principale includ insertPlayer și insertScore pentru inserarea datelor în tabele, și diverse metode de interogare pentru recuperarea datelor recente sau cele mai bune scoruri.

### **Pachetul States:**

State: Implementează noțiunea de stare a jocului sau programului. Conține attribute și metode esențiale pentru gestionarea stării curente a jocului, cum ar fi previousState și currentState pentru a gestiona tranzițiile între diferitele stări ale jocului, și metode abstracte Update() și Render(Graphics g) pentru actualizarea și desenarea stării curente. De asemenea, include metode pentru setarea și obținerea stării curente sau anterioare a jocului.

GameOverState, GameWinState, HelpState, HighScoresState, MenuState, PauseState, PlayState, SettingsState: Clase specifice care extind State și implementează funcționalități și interfețe grafice pentru diferitele stări ale jocului, cum ar fi afișarea ecranului de Game Over, Game Win, informații, cele mai bune scoruri, meniul principal, pauză, jocul propriu zis și setările.

### **Pachetul Tiles:**

TileManager: Gestionează toate tipurile de dale disponibile în joc, stocând o listă de dale și oferind metode pentru accesarea acestora în funcție de un identificator.

Wall1Tile, Wall2Tile, Wall3Tile, Floor1Tile, Floor2Tile, Floor3Tile: Clase care reprezintă tipuri specifice de dale. Extind clasa de bază Tile și sunt responsabile pentru inițializarea dalei cu o imagine specifică și setarea proprietăților, cum ar fi soliditatea.

Tile: Clasa de bază pentru toate tipurile de dale din joc, definind o dală ca fiind compusă dintr-o imagine și un identificator unic, împreună cu alte proprietăți, cum ar fi soliditatea.

### **Pachetul Timer:**

Timer: Gestionează cronometrul în joc. Ea stochează o întârziere specificată și oferă metode pentru a seta această întârziere, a verifica dacă aceasta a trecut și pentru a desena un indicator grafic al cronometrului.

### **Pachetul UI:**

ClickListener: Definește un comportament pentru evenimentele de click al mouse-ului în cadrul interfeței utilizatorului, conținând o singură metodă onClick().

UIObject: Clasă abstractă care definește caracteristicile și comportamentul de bază al obiectelor interfeței grafice, având o poziție (x, y), lățime și înălțime. Ele pot fi actualizate și desenate în funcție de starea lor curentă și pot avea o acțiune asociată la apăsarea mouse-ului.

UIButton: Extinde UIObject și reprezintă un buton grafic care conține o imagine și permite definirea unei acțiuni la apăsarea sa.

UIManager: Gestionează obiectele interfeței grafice, păstrând o listă de astfel de obiecte și oferind metode pentru actualizarea și desenarea lor, precum și pentru gestionarea evenimentelor de mișcare și eliberare a mouse-ului.

### **Pachetul World:**

World: Gestionează lumea din joc, ocupându-se de încărcarea și desenarea hărții, a dalelor, a entităților și a altor elemente, precum FogOfWar. De asemenea, procesează informațiile legate de nivelurile jocului și de evenimentele și stările specifice fiecărui nivel.

### **Bibliografie:**

<https://opengameart.org/>

<https://www.deviantart.com/>

<https://www.pngwing.com/>

<https://onlinepngtools.com/change-png-color>

<https://www.uhdpaper.com/2024/03/garfield-movie-garfield-odie-4k-543a.html?m=0>

<https://youtu.be/qejkF9spP28?si=gLkuRsVAYKQmeFJP>