

**Titlu proiect(“La umbrele”)**  
**Autor (Irina Rotaru)**  
**Grupa(1208A)**

**Povestea jocului:**

Jocul este situat pe o plajă cu ieșire la Marea Mediterană numita “La umbrele” și urmărește personajul principal Sandy, care este o pisică. Într-o după-amiază specială în care familia ei a decis să facă o ieșire pe plajă ca să se joace împreună și să se bucure de soare, o furtună a aparut de nicaieri, și au fost nevoiți să plece. Lui Sandy îi e frica de apă, asa că, în timp ce familia ei pleca, a rămas în urma și s-a pierdut. El nu se mai poate întoarce pe plajă din cauza dezastrului intamplat, aşa că ramane încercarea pisicii de a o parcurge pe toată și a se reîntâlni cu familia ei. Totuși, ea trebuie să treacă printr-o serie de încercări, deoarece apă a inundat o gramada de locuri și îi raman doar fasii de nisip pe care poate merge, iar umbrelele rămase pe plajă îi pun întrebări de cultură generală. Norocul ei face ca pe plajă, familia ei se juca un joc de acest gen, deci să sperăm că va ști și ea să raspunda. Dacă va reuși să raspunda corect la întrebările primite și să iasă din plajă ce o sperie, ea se va reuni cu familia ei și vom avea o familie fericită.

**Prezentare joc:**

Mod single player în care jucătorul trebuie să parcurgă o hartă și să acumuleze răspunsuri corecte pentru a putea câștiga. El începe de la nivelul 1, care este mai ușor, și dacă îl parcurge cu succes, trece automat la nivelul 2. Dacă termină și al doilea nivel, câștiga jocul și se întoarce la meniu.

**Reguli joc:**

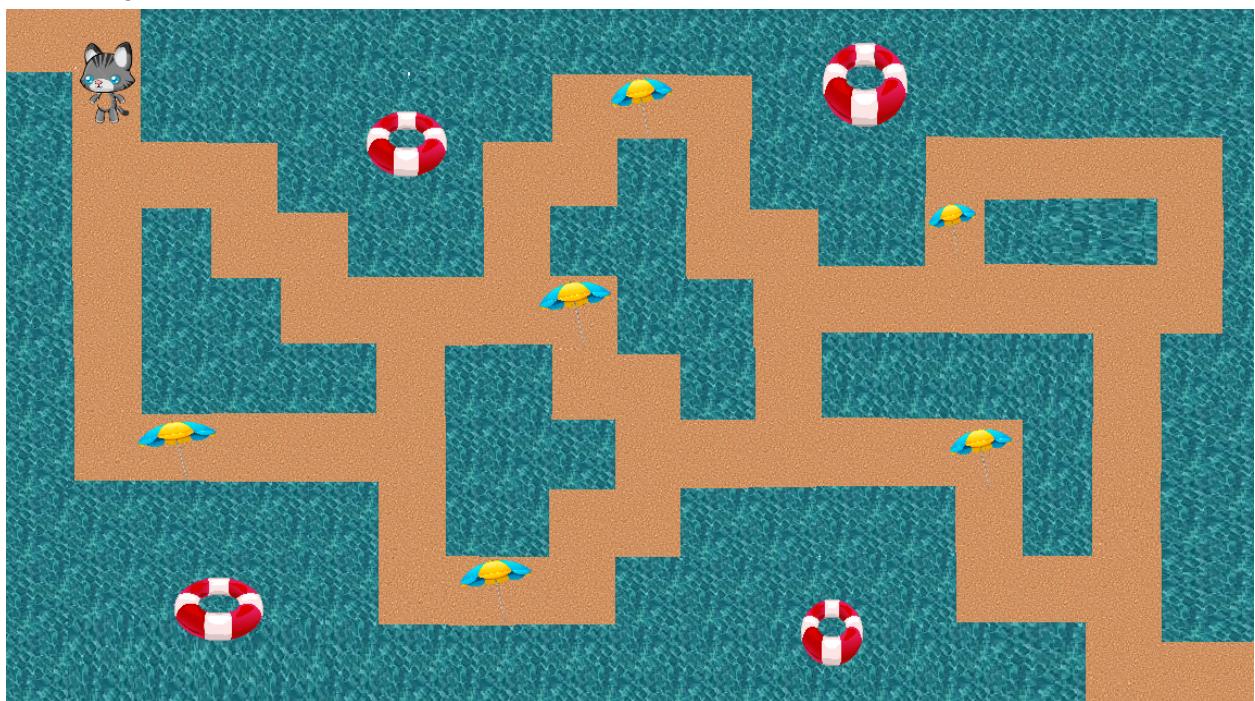
Jocul implica parcurgerea tablei de joc până la ieșire. Pe drum, personajul trebuie să treacă pe la umbrele pentru a acumula 3 răspunsuri corecte. Dacă are un număr mai mare sau egal cu 3 de răspunsuri corecte, mai trebuie doar să ajungă la capatul tablei pentru a câștiga, iar dacă are mai puține răspunsuri, pierde. Pe tabla vor fi 6 umbrele la care poate merge, se poate răspunde o singura dată la ele. Dacă câștiga, trece și la nivelul 2 unde trebuie să facă același lucru, dar cu 4 răspunsuri corecte la întrebări mai grele.

**Personajele jocului:**

Sandy este personajul principal al jocului. Este o pisică ce își iubește foarte mult familia și ar merge până și la plajă pentru a petrece timp cu ei, chiar dacă îi este frica de apă. Este și foarte isteata, știind răspunsul la multe întrebări de cultură generală. Ea interacționează cu o umbrelă (nu se mișcă) ci doar declanșează niște întrebări pe baza cărora se pot acumula puncte. Imaginea ei se întoarce pe partea stângă și pe cea dreaptă în funcție de direcția de mers.



#### Tabla de joc:



Mai sus se observa tabla de joc. Personajul apare pe nisip, deoarece doar pe acolo poate merge, în jurul său sunt bucăți de apă pe care nu poate păși, e implementată coliziunea. Pe drum se observă umbrele, obiecte cu care urmează să interacționeze pe parcursul jocului. Colacii de salvare sunt elemente de decor. Trebuie să ajungă în capătul opus al tablei de joc pentru a se considera încheiat nivelul.

#### Mecanica jocului:

Jucătorul se mișcă prin apasarea tastelor up, down, right, left în direcțiile indicate. Dacă ajunge la o umbrelă și apăsa tasta enter va interacționa cu această, adică îi va apărea pe ecran

o bucată de text care e întrebarea și mai jos două răspunsuri posibile. Pentru a da răspunsul trebuie să apese una din tastele A sau B, A e primul răspuns, B e al doilea. Nu pot fi apăsatе ambele și nu se poate interacționa cu aceeași umbrelă de două ori. Dacă a răspuns corect, i se va adăuga un punct la scor. Cand a terminat de interacționat cu umbrela, trebuie să apese tastă esc pentru a ieși din întrebare. Oricând în timpul jocului, cel ce joacă poate apăsa tastă backspace pentru a trece în meniu principal al jocului, iar de acolo load îl aduce exact unde a ramas.

### **Game sprite:**

-nisip:



-apa:



-umbrela:



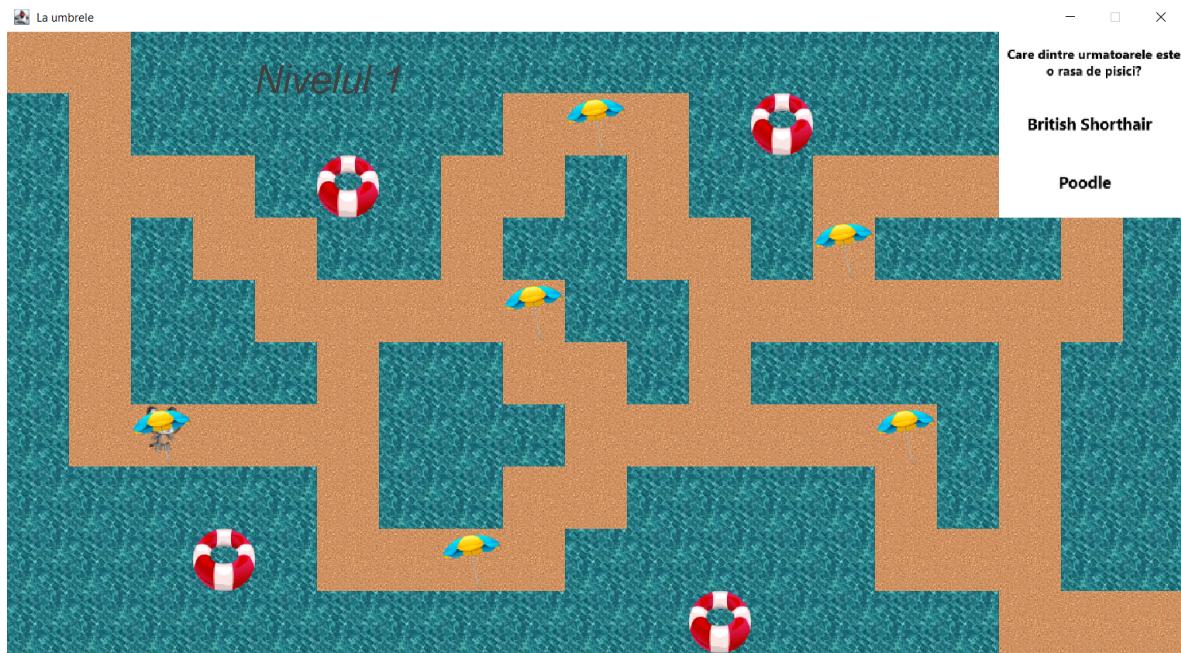
-colac de salvare:



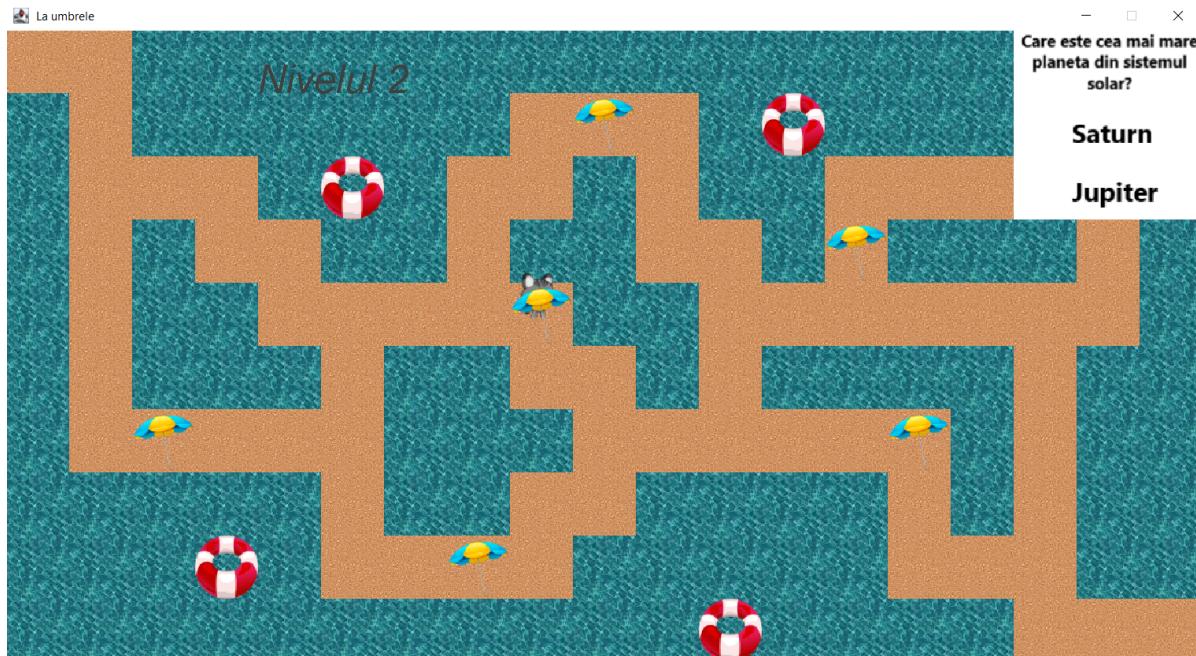
Care dintre următoarele este o rasa de pisici?	Cate continente există pe Pamant?	Cine a pictat "Cina cea de Taină"?	British Shorthair	Londra	7
Care este capitala Franței?	Care este cel mai înalt munte de pe glob?	In ce galaxie se află Pamantul?	Poodle	Paris	6
7	Makalu	Leonardo da Vinci		Andromeda	
6	Everest	Michelangelo Buonarroti		Calea Lactee	
Care este cel mai lung fluviu din Europa?	In ce an a intrat Romania în Primul Razboi Mondial?	Care este cea mai mare planetă din sistemul solar?	Cum se numește stadionul echipei Real Madrid?	Ce este soarele?	Care este cel mai rapid animal din lume?
Volga	1914	Saturn	Santiago Bernabeu	O planeta	Ghepardul
Dunarea	1916	Jupiter	Olimpico	O stea	Tigrul

### Descriere fiecare nivel:

La primul nivel, cel usor, jucătorul va trebui sa raspunda la 3 din 6 întrebări și ele nu vor fi de dificultate foarte grea. În poza putem observa ca personajul este la umbrela, iar întrebarea apare în colțul din dreapta sus. Se observa și nivelul ei de dificultate. Daca jucatorul a răspuns corect la minim 3 întrebări și ajunge în colțul din dreapta jos al tablei, el o sa treaca automat la nivelul următor. Daca nu, intra in meniul de Game Over.



La al doilea nivel, cel greu, jucătorul va trebui sa raspunda la 4 din 6 întrebări și ele sunt de dificultate mai ridicata. În poza putem observa ca personajul este la umbrela, iar întrebarea apare în colțul din dreapta sus. Se observa și nivelul ei de dificultate mai ridicat.



În următoarele două poze vedem ce se întampla în caz ca nivelul a fost câștigat, respectiv pierdut.



#### Descriere meniu:

Vor fi 4 butoane: new, save, load, exit. New începe un joc nou în care toate detaliile precum poziția jucătorului, nivelul și scorul sunt resetate. Save salveaza progresul de pana atunci, incarcand intr-o baza de date pozitia prin coordonate a jucătorului, x si y, de unde s-a apăsat butonul, respectiv scorul acestuia. Load încarcă jocul exact din poziția în care s-a rămas,

fără a afecta nivelul sau scorul. Exit va ieși din jocul curent. Pentru a accesa oricare dintre aceste setari, în meniu se navigheaza cu tastele up și down, iar enter alege una din opțiuni, cand cursorul din fata este pe ea. În meniu de Game Over, care apare mai sus, daca se apasa Retry, jucătorul poate încerca din nou nivelul, iar daca se apare Quit, este dus în meniu principal al jocului, aratat mai jos.



#### Diagrame de clase:

##### Main

După cum se poate observa din diagrama prezentată mai jos, această clasă apelează metoda getInstance din clasa GameSingleton pentru a accesa jocul, iar mai apoi pornește fluxul activităților prin apelul metodei StartGame() din GameSingleton.

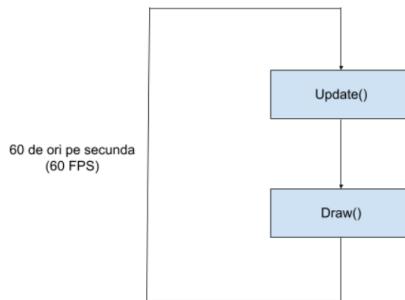
##### GameSingleton

Rolul acestei clase este de a face inițializările (de a crea instanțe pentru jucător, pentru obiecte, de a inițializa harta), dar și de a menține interfața grafică la zi cu ceea ce se întâmplă în joc. Această clasă implementează interfața Runnable pentru a avea comportamentul unui fir de execuție (thread). În momentul apelului metodei StartGame() se instantiază un obiect de tip Thread pe baza instanței curente a clasei GameSingleton.

Orice obiect de tip Thread trebuie să implementeze metoda run() care este apelată atunci când firul de execuție este pornit (start()). Această metodă run() initializează jocul prin crearea unei instanțe GameWindow, iar mai apoi controlează numărul de cadre pe secundă printr-o buclă while și pregătește noua scenă (Update()) pe care o va desena pe interfața grafică (prin Facade-Drawer-Draw()).

Metoda Update() apelează funcția update din clasa Player, care actualizează starea jocului (de exemplu: modifica poziția jucătorilor pe baza tastelor apăsate, verifică constant prin personaj coliziunea, crează diferite tile-uri).

Metoda Draw() din clasa Drawer, respectiv din Facade, va desena pe interfață grafică modificările făcute de metoda Update(). Interfața grafică este un canvas.



### GameWindow

Metoda run() a clasei GameSingleton crează o instanță a clasei GameWindow. Această clasă este responsabilă cu fereastra în care vor fi desenate obiectele pe un canvas (se poate observa că avem o metodă GetCanvas care întoarce canvasul). De asemenea, avem un obiect JFrame care permite desenarea de butoane, controale, textbox-uri, etc, dar poate conține și un canvas în care pot fi desenate diverse obiecte folosind texturi.

### Tile

Această clasă reține informații despre tile-urile din joc. În clasa GameSingleton există o matrice cu elemente de tip int ce ne spune în funcție de poziția lor pe harta dacă tile-uri sunt apă sau nisip. Se parcurge și se desenează tile-ul corespunzător valorii din matrice. De asemenea, clasa Tile mai reține și câte o instanță pentru fiecare subtip de tile (WaterTile, SandTile, QuestionTile). Are și două metode Draw() și Draw2() deoarece întrebările au dimensiuni diferite față de apă și nisip.

### WaterTile, SandTile, QuestionTile

Toate aceste clase extind clasa Tile, preluând comportamentul acelei clase sau putând să-l suprascrie folosind @Override. Constructorul acestora primește ca parametru un ID prin care se va putea identifica acel tile și apelează constructorul clasei de bază, adică al clasei Tile, care primește ca parametru un membru static al clasei Assets. Membrul static folosit este ales în funcție de tipul clasei respective.

### Assets

Conține câte un membru static de tip BufferedImage pentru fiecare tile, chiar și pentru jucător/obiecte. În acești membri sunt stocate imaginile, adică texturile acestora care vor fi desenate prin apelarea metodelor Draw() din fiecare clasa tile apelate din metoda Draw() din clasa Drawer, de Facade din GameSingleton. În acest exemplu este doar o imagine care conține toate texturile folosite, aşadar pentru a putea instanța acest obiect static trebuie să decupăm fiecare textură din acea imagine. Pentru a face asta se folosește o instanță a clasei SpriteSheet în metoda Init(). Metoda folosită din SpriteSheet este crop(x, y) sau crop2(x,y) în funcție de dimensiuni (întrebarea este de 3 ori mai lungă ca un tile clasic).

### SpriteSheet

Constructorul acestei clase primește imaginea, iar clasa conține 2 membri constanți pentru înălțimea, respectiv lățimea texturilor. Metoda crop(x, y) primește doi parametrii, x specifică pe ce coloană se află textura pe care dorim să o decupăm în imaginea cu toate texturile, iar y specifică linia. La fel pentru crop2(x,y), dar imaginea are alta lungime. Pentru a afla efectiv poziția în imagine se determină pixelii de unde încep texturile de interes prin înmulțirea liniei/coloanei cu înălțimea, respectiv lățimea texturilor. Astfel se obține colțul din stânga sus al texturii, iar pentru a afla celelalte colțuri se folosesc înălțimea, respectiv lățimea.

### ImageLoader

Înainte de a extrage fiecare textură, imaginea cu toate texturile trebuie să fie citită din memorie, iar pentru asta se folosește clasa ImageLoader cu metoda statică LoadImage(path) care primește ca parametru calea către imagine în memoria calculatorului.

### Entity

Clasa de baza ce urmează a fi extinsă. Reține informații despre imaginile folosite, direcția în care merge personajul, coliziunea acestuia, coordonatele la care se află și viteza cu care se va deplasa.

### Player, Colac, Umbrela

Player este personajul principal al jocului, el se poate mișca pe harta în funcție de coliziune. Poate interacționa cu umbrele prin a răspunde la întrebări, el oferă și răspunsuri în funcție de tastele apasate, adică A și B (CheckResponseA(), CheckResponseB()). Cu ajutorul ei desenam personajul pe harta, îl mișcăm prin taste și îi schimbăm imaginea în funcție de direcție (stanga, dreapta). Verificăm tot acolo când s-a terminat nivelul în funcție de poziția lui (sa ajungă la finalul hărții) și implementeză notiunea de scor ce duce la castigarea/ pierderea jocului. Colacul este doar un obiect de decor, nu are proprietăți speciale, nu are viteza, sta mereu pe loc.

Umbrela nu se mișcă de asemenea, dar este un obiect cu care player-ul interacționează. Vor fi mai multe astfel de obiecte desenate în diverse poziții din cadrul hărții.

### KeyHandler

Implementează clasa KeyListener, deci suprascrie metodele KeyPressed și KeyReleased. Verificăm în ea dacă tastele pe care le-am folosit au fost apasate și eliberate. Se utilizează atât pentru deplasare (sagetele up, down, left, right), cât și pentru interacție: enter, esc, A,B și pentru a ieși din joc în meniu: backspace.

### CollisionChecker

Verifică în funcție de direcția în care merge personajul și de coordonatele acestuia dacă urmează să dea de un tile de nisip sau de unul de apă. Poate merge doar pe nisip, deci în

funcție de o variabilă collisionOn verificăm dacă se poate deplasa, sau dacă va fi implementată coliziunea. Personajul nu se izbește de obiecte (umbrele), deoarece interacționează cu ele.

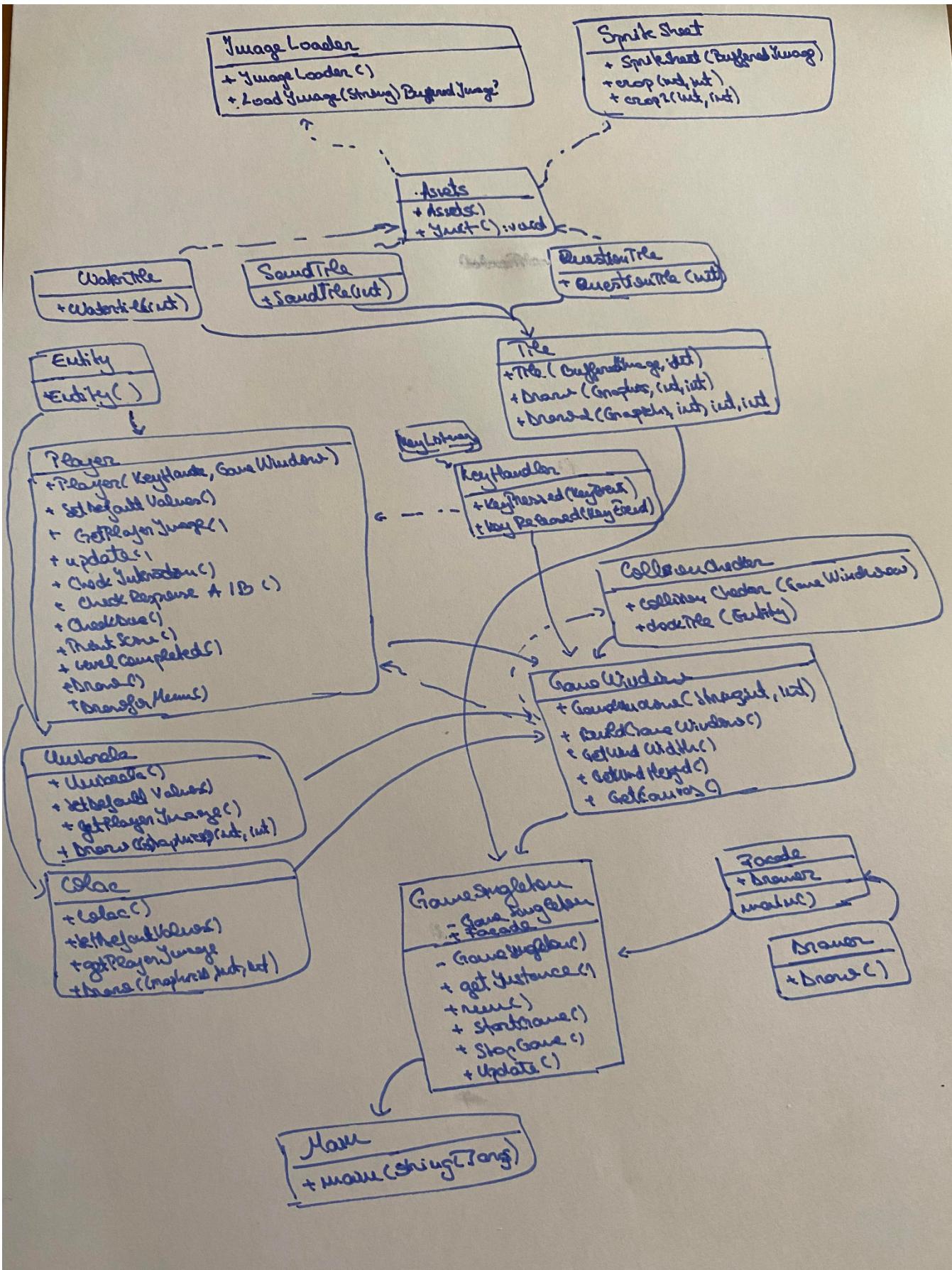
#### Facade

Implementeaza design pattern-ul cu acest nume, face mai simplă relaționarea dintre metoda de desenare și joc.

#### Drawer

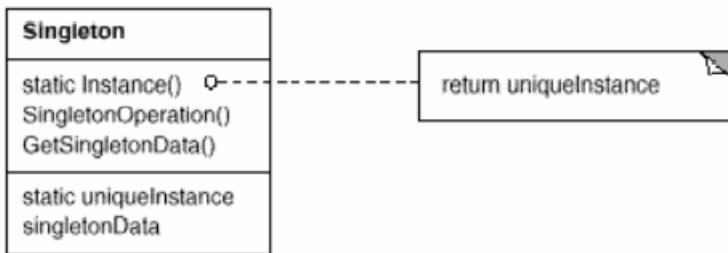
Implementeaza metoda Draw() pentru a desena elementele de pe harta.

**Mai jos se poate observa diagrama UML de clase.**

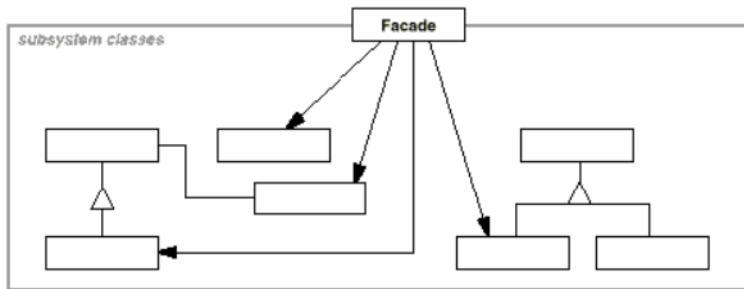


### **Sabioane de proiectare utilizate:**

Singleton - Am folosit acest şablon de proiectare deoarece asigura ca o clasa are doar o instanță și oferă un punct de vedere global asupra ei. L-am utilizat în a face clasa GameSingleton, pentru a ma asigura că este un singur joc creat și pentru că am vrut să fie global vizibilă. Trebuie să avem un singur joc și acesta să fie accesibil pentru clienți. Am denumit GameSingleton în proiect, am făcut constructorul, un singleton și metoda getInstance, pentru a putea fi folosit în Main. Mai jos am pus și structura acestui Design Pattern.



Facade - Am folosit acest şablon de proiectare pentru a avea o interfață ce face sistemul mai ușor de utilizat. Am considerat că va reduce complexitatea și dependențele dintre sisteme și mi s-a părut important să am ceva mai simplu pentru a putea accesa complexitatea programului. L-am utilizat pentru a face un Drawer, astfel încât, în funcția run, ce rulează jocul, să acceseze prin Facade metoda de a desena harta și elementele ei și să pară mai simplu, în comparație cu complexitatea funcției Draw. Mai jos am pus structura acestui design pattern.



### **Baze de date:**

Când se apăsa butonul Save din meniu, se scriu într-o bază de date poziția jucătorului (coordonatele x și y la care se află acesta pe harta) și scorul pe care îl are în momentul în care a apăsat save. Baza de date e facută în DB Browser. Mai jos este o poza cu tabelul și informațiile stocate pe el. Aceasta este și afișată în terminal dacă se apăsa Save pentru a putea fi vizualizată.

The screenshot shows the DB Browser for SQLite interface. The main window displays a table named 'Joc' with three columns: x, y, and score. The data is as follows:

	x	y	score
1	194.0	368.0	2
2	1034.0	562.0	3
3	1034.0	562.0	3
4	560.0	432.0	1
5	1034.0	524.0	5
6	1016.0	400.0	5
7	514.0	268.0	1
8	1034.0	576.0	3

An 'Edit Database Cell' dialog is open for the first row, specifically for the 'score' column. The value '1' is displayed, and the 'Mode' dropdown is set to 'Text'. A tooltip indicates the type is 'Text / Numeric' and there is '1 character(s)'. The 'Apply' button is visible.

## Bibliografie:

[https://www.123rf.com/photo\\_117609133\\_stock-vector-cat-game-sprites.html](https://www.123rf.com/photo_117609133_stock-vector-cat-game-sprites.html)

<https://opengameart.org/content/sand-texture-pack>

<https://opengameart.org/content/water-texture-pack>

<https://www.cleannpng.com/png-yellow-blue-beach-umbrella-png-clipart-17761/>

<https://www.cleannpng.com/png-lifebuoy-png-30633/>

Design Patterns. Elements of Reusable Object-Oriented Software. By Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.