

Meșterul Grigore
Petrisor Eduard-Gabriel

1212A

1. Context	2
2. Gameplay	2
3. Continut	3
4. Niveluri/Hărți	5
5. Interfața Grafică	5
6. Architectural Design Document.....	8
6.1. Pachetul Entity.....	8
6.2. Pachetul Input.....	8
6.3. Pachetul Graphics.....	9
6.4. Pachetul Map.....	9
6.5. Pachetul Objects.....	10
6.6. Pachetul GameWindow.....	10
6.7. Pachetul DataBase.....	11
7. Sabloane de Proiectare.....	11
8. Despre.....	12
9. Gallery.....	13
10. Bibliografie	16

1. Contextul Jocului:

În 2016 compania producătoare de software C.A.P. Industries este înființată de către Eduard the Carpathian Thunder. Compania urmând să aibă o creștere foarte rapidă și în mai puțin de un an ajungând cel mai mare provider de software pe plan global. Motivul acestei creșteri este unul simplu, Eduard și-a folosit stăpânirea asupra electricității pentru a oferi limbajelor de programare folosite de către companie o formă de percepție de sine. Acest lucru a permis dezvoltarea unei relații de colegialitate între Limbaje și Programatori. Relație care a devenit fundația companiei și a permis producerea de programe și actualizarea programelor deja existente de 10 ori mai rapid față de competitori.

În 2032 Eduard the Carpathian Thunder moare de bătrânețe la vârstă de 96 de ani eveniment ce a cauzat limbajele de programare să se răscăle. Acum este misiunea lui Grigore, fiul lui Eduard și singurul său discipol să vină în companie și să readucă limbajele la comportamentul lor uzual.

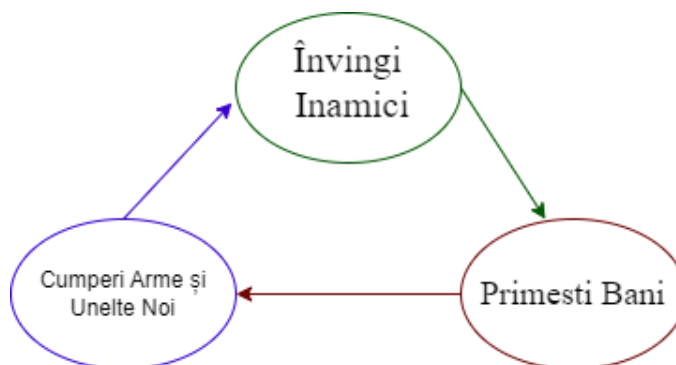
2. Gameplay:

a. Controale:

- i. W - mergi înainte
- ii. S - mergi înapoi
- iii. A - mergi la stânga
- iv. D - mergi la dreapta

b. Descriere

- i. Jucătorul va putea alege un caracter înainte de a începe o hartă.
 - 1. Vor fi disponibile 3 caractere, dintre care 1 va fi disponibil încă de la început iar celelalte 2 vor fi deblocate pe parcursul jocului.
 - 2. Fiecare caracter va avea o armă și o caracteristică specifică.
- ii. După selectarea caracterului jucătorul va putea alege harta pe care să joace.
 - 1. Vor fi disponibile 3 hărți, a doua hartă va fi deblocată după completarea primei hărți, iar a treia va fi deblocată după completarea celei de-a doua hărți
 - 2. Fiecare hartă va fi mai dificilă decât harta anterioară.
- iii. O dată ce un nivel începe wave-uri de inamici vor ataca jucătorul. Fiecare inamic învins va oferi jucătorului o anumită cantitate de bani.
- iv. Folosind banii obținuți jucătorul va putea cumpăra noi arme și unelte ce îl vor asista în completarea nivelului.
- v. După 15 minute va apărea un Boss, o dată ce acesta este învins nivelul se va sfârși.



c. Reguli:

- i. Jucătorul poate doar să se miște și să aleagă ce dorește să cumpere.
- ii. Armele trag automat.
- iii. Jucătorul poate avea în posesie maxim 3 arme și 3 unelte.
- iv. Magazinul cu arme și unelte va apărea o dată la x minute în timpul unui nivel și va oferi jucătorului o selecție de 3 arme și unelte aleatorii pe care să le cumpere. Jucătorul poate alege să plece din magazin fără să cumpere nimic.
- v. Cât timp jucătorul este în contact cu un inamic, acesta va suferi în fiecare secundă o cantitate de damage egală cu valoarea atacului inamicului.
- vi. Jucătorul va putea distruge anumite obiecte prezente pe hartă pentru a recupera o cantitate predefinită de viață.

3. Continut:

a. Personaje:

- i. Grigore M.



Caracteristica: Fiu de Magnat → toate armele și uneltele din magazin sunt cu 20% mai ieftine.

- ii. Weng W.



Caracteristica: *Nu a fost stabilită încă*

- iii. Thomas S.



Caracteristica: *Nu a fost stabilită încă*

b. Inamici

- i. Universali: C, C++, Java



- ii. Specifici Harta 1: Python, JavaScript, CobraSabe(Boss harta 1)



Sprite Boss: *Nu a fost stabilit încă*

- iii. Specifici Harta 2: Rust, TypeScript, Mircea Regele Crab(Boss harta 2)



Sprite Boss: *Nu a fost stabilit încă*

- iv. Specifici Harta 3: Gandacul Doliului(Boss harta 3)

Sprite Boss: *Nu a fost stabilit încă*

c. Arme:

- i. **VIM**

- Damage de baza mic
- Va trage similar unei mitraliere
- Viteza de atac a acestei arme va crește cu y la fiecare x secunde

- ii. **NotePad**

- Nu are damage, în schimb are o șansă de x% să elimine toți inamicii vizibili pe ecran o dată la fiecare y secunde

- iii. **VS**

- Damage de baza mare
- Viteza de atac mică
- Extra dmg împotriva C/C++

- iv. **VS Code**

- Damage de baza mediu
- Viteza de atac medie
- Creeaza o zonă de damage în jurul Jucătorului. Zona va crește cu x% la fiecare y inamici învinși.

- v. **IntelliJ**

- Damage de baza mare
- O dată la x secunde spawnază aleatoriu pe hartă o mică zonă de damage, zona dispare după y secunde.
- După z inamici învinși zona va crește în mărime cu v%.

d. Unelte:

- i. Keyboard -> Jucătorul se va mișca mai rapid pe hartă
- ii. Processor -> Scade Cooldown-ul de la NotePad și IntelliJ
- iii. Extra Monitors -> Oferă un proiectil în plus pentru VS și IntelliJ
- iv. Debugger -> Crește Damage-ul pentru toate armele mai puțin NotePad și VIM
- v. SSD -> Crește viteza de atac pentru VIM, VS și VS Code

4. Niveluri/Hărți:

a. Harta 1 → Câmpia Șarpelui

Harta nu va prezenta restricții asupra Jucătorului, acesta putând să se miște cât de mult dorește în orice direcție. Inamicii la rândul lor vor putea apărea din oricare direcție. Harta va arăta similar unei câmpii.

b. Harta 2 → Oceanul Crabului

Harta va restricționa abilitatea Jucătorului de a se mișca în sus și în jos aceste direcții fiind blocate. Inamicii vor apărea atât din partea stânga a hărții cât și din cea dreapta. Harta va arăta similar unui ocean subteran partea de sus fiind rocă iar partea de jos fiind fundul oceanului. Pe această Hartă inamicii vor primi un bonus de extra 25% Viața și Atac față de prima Hartă.

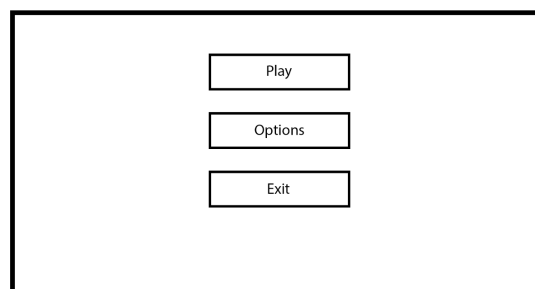
c. Harta 3 → Temnița Gândacului

Harta va fi similară unui culoar infinit deoarece Jucătorul se va putea mișca cât de mult dorește în sus sau jos însă direcțiile stânga și dreapta vor fi blocate de pereți. Inamicii vor apărea atât din partea de sus a hărții cât și din cea de jos. Harta va arăta similar unei temnițe.

Pe această Hartă inamicii vor primi un bonus de extra 50% Viața și Atac față de prima Hartă.

5. Interfața Grafică:

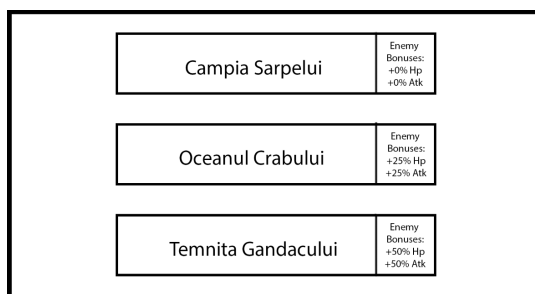
Când jocul este deschis Jucătorul va fi întâmpinat de un meniu simplu ce va conține 3 opțiuni: *Play*, *Options*, *Exit*. Opțiunea de *Exit* va închide jocul.



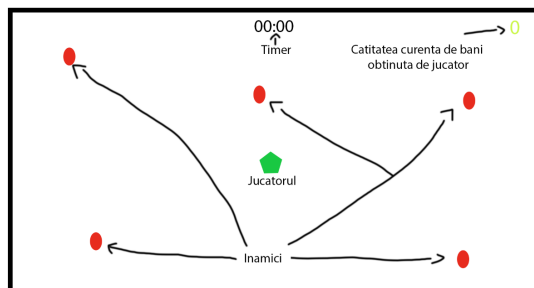
Dacă Jucătorul alege opțiunea de *Play* acesta va fi condus la un nou meniu de unde poate selecta caracterul cu care dorește să joace.



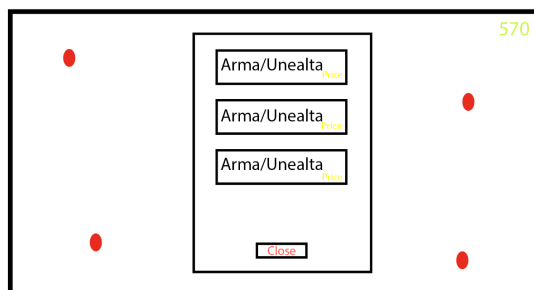
După alegerea unui caracter Jucătorul va fi condus la un nou meniu de unde poate alege harta pe care dorește să joace.



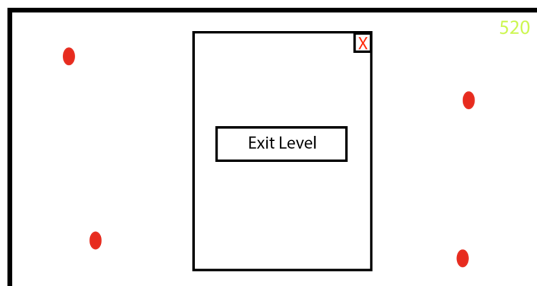
După alegerea unei Hărți jocul va începe:



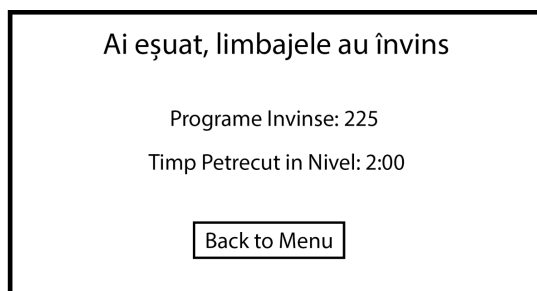
O data la x minute petrecute în nivel, în fața jucătorului va apărea un meniu în care va avea la dispoziție o selecție de 3 Arme și Unealte aleatorii pe care le poate cumpăra. Inamicii și Timer-ul nu se vor mișca cât timp acest meniu este activ.



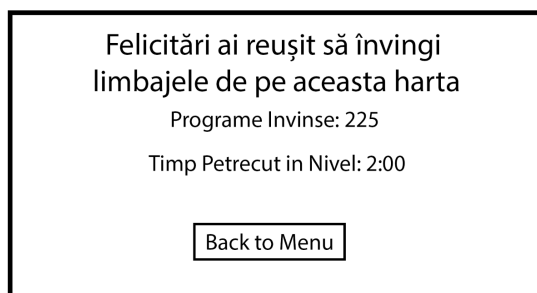
Dacă Jucătorul apasă tasta Esc în cadrul nivelului atunci se va deschide un meniu cu o singură opțiune *Exit Level*. Cat timp meniul este deschis inamicii și timer-ul nu se vor mișca. Meniul poate fi închis prin o a doua apăsare a tastei Esc sau prin apăsarea butonului *x* din stanga sus a meniului. Opțiunea de *Exit Level* va trimite Jucătorul înapoi la meniul principal.



Dacă jucătorul este învins de inamici acesta va fi întâmpinat de un meniu ce îi va arăta în partea de sus “Ai eșuat, limbajele au învins”, urmat de numărul de inamici învinși, timpul petrecut în nivel și opțiunea de întoarcere la meniul principal.



Dacă jucătorul învinge inamicii acesta va fi întâmpinat de un meniu ce îi va arăta în partea de sus “Felicitări ai reușit să învingi limbajele de pe aceasta hartă”, urmat de numărul de inamici învinși, timpul petrecut în nivel și opțiunea de întoarcere la meniul principal.

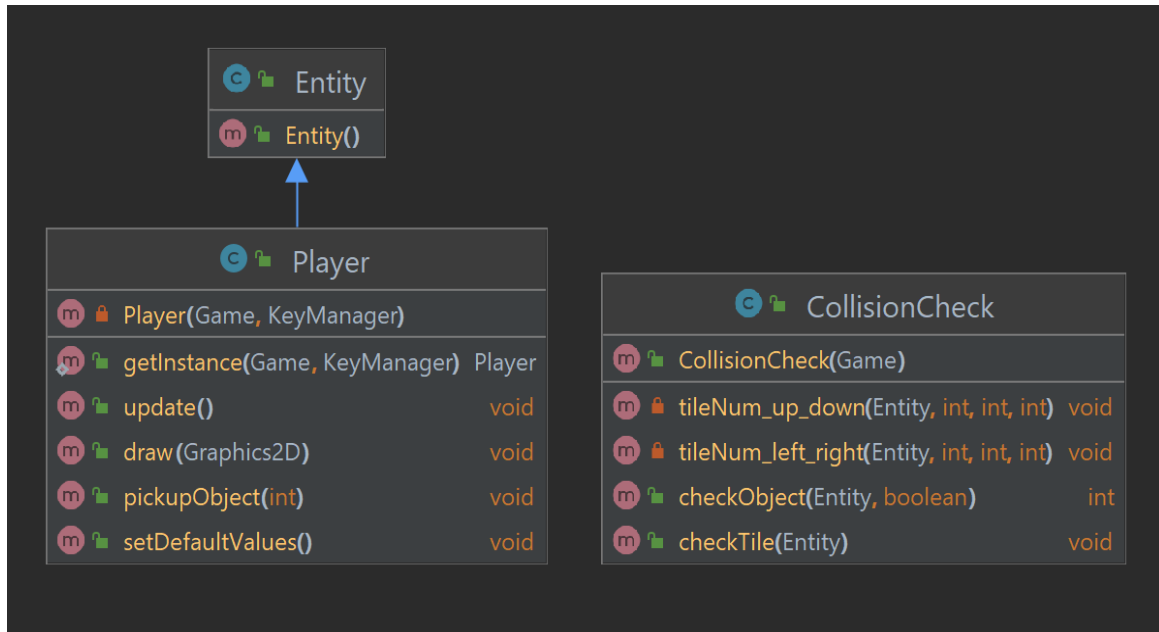


6. Architectural Design Document:

Jocul este împărțit în multiple pachete fiecare implementand o functionalitate diferită a acestuia:

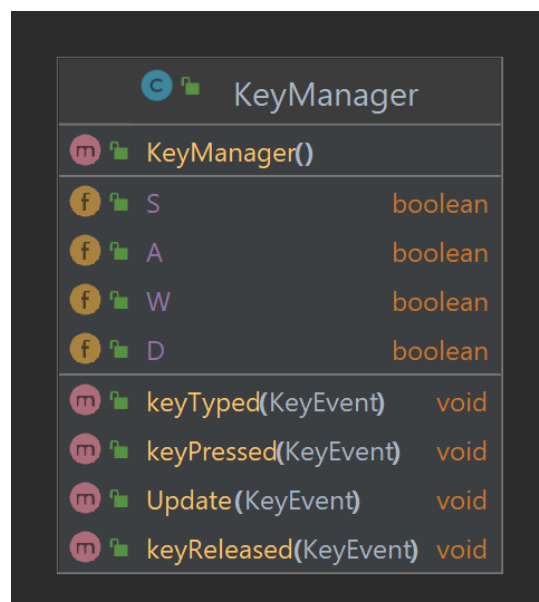
a. Pachetul Entity:

Acest pachet contine atat logica principala pentru coliziunea dintre Tiles, Entități și Obiecte cât și funcțiile pentru updatarea și desenarea Player-ului. Clasa Player a fost implementată folosind șablonul de proiectare singleton pentru a preveni existența a multiple obiecte de tip Player.



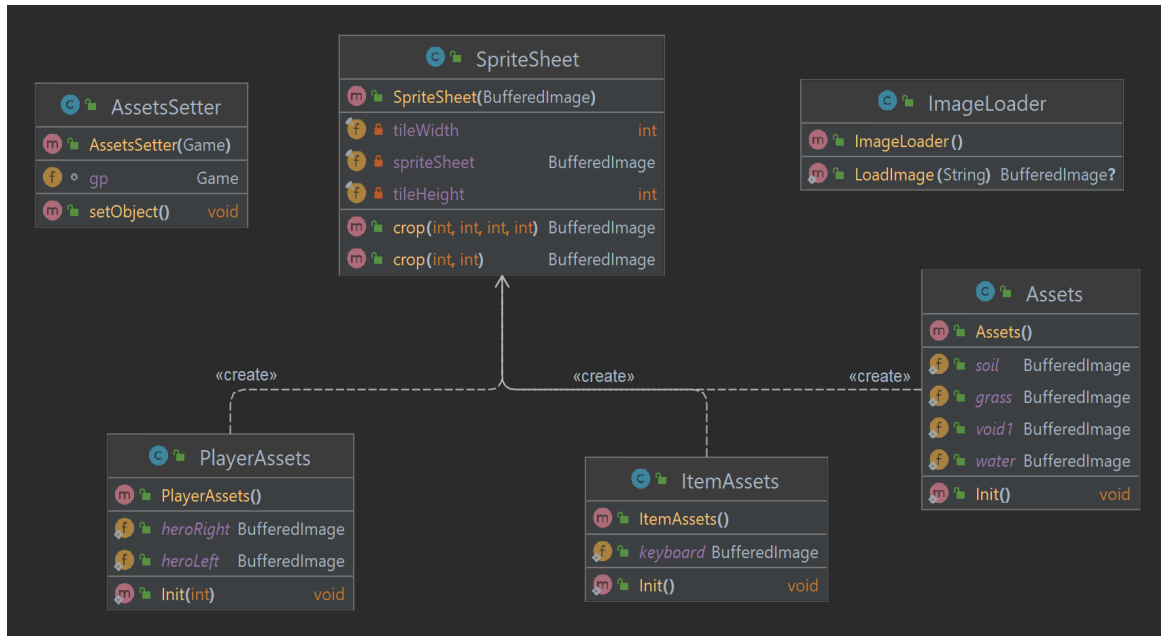
b. Pachetul Input:

Acest pachet contine doar clasa KeyManager care reține dacă o tasta a fost apasata sau nu.



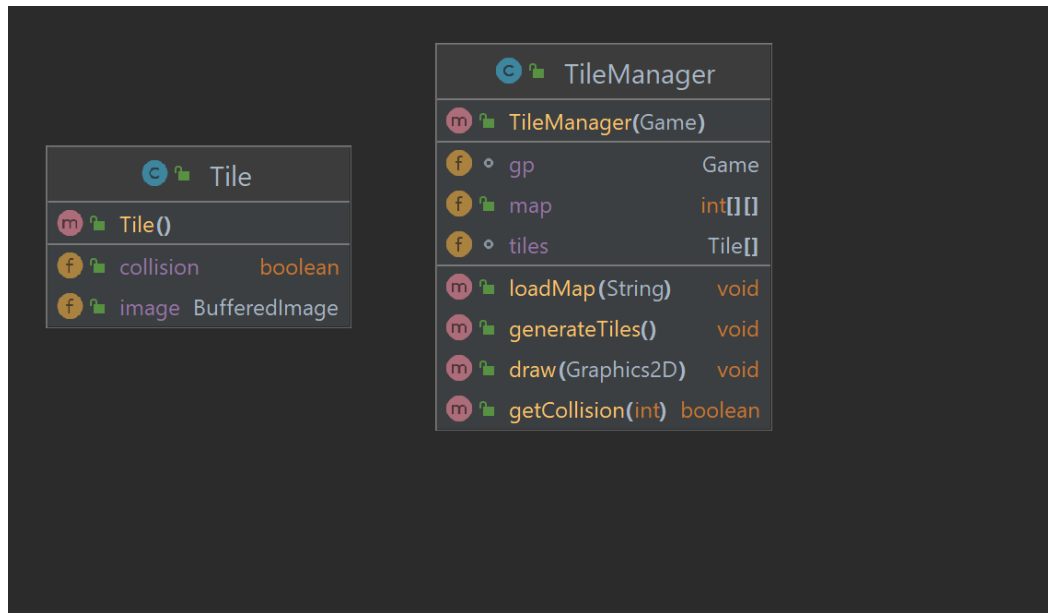
c. Pachetul Graphics:

Acest pachet contine logica necesara încărcării imaginilor folosite pentru Entități, Iteme și Tile-uri și procesării acestora;



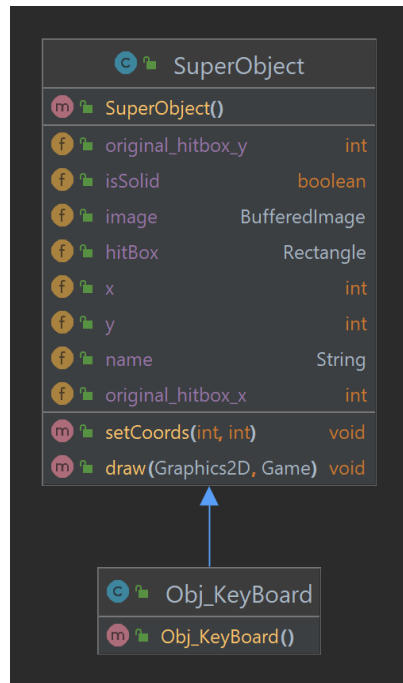
d. Pachetul Map:

Acest pachet contine logica folosită pentru citirea matricii corespunzătoare mapei curente, desenarea mapei bazată pe valorile din matrice și implementarea camerei.



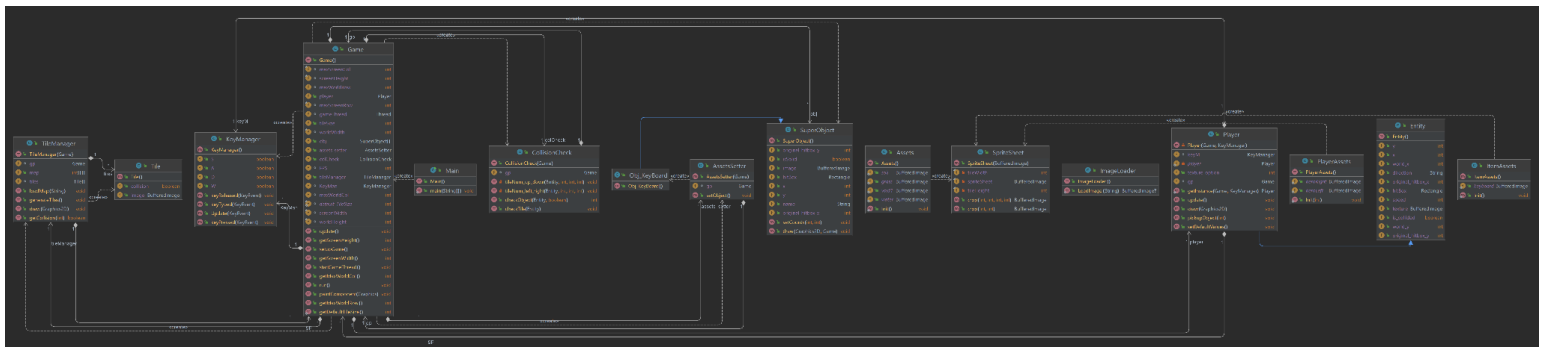
e. Pachetul Objects:

Acest pachet contine logica principala pentru generarea și desenarea obiectelor cu care Player-ul poate interactiona.



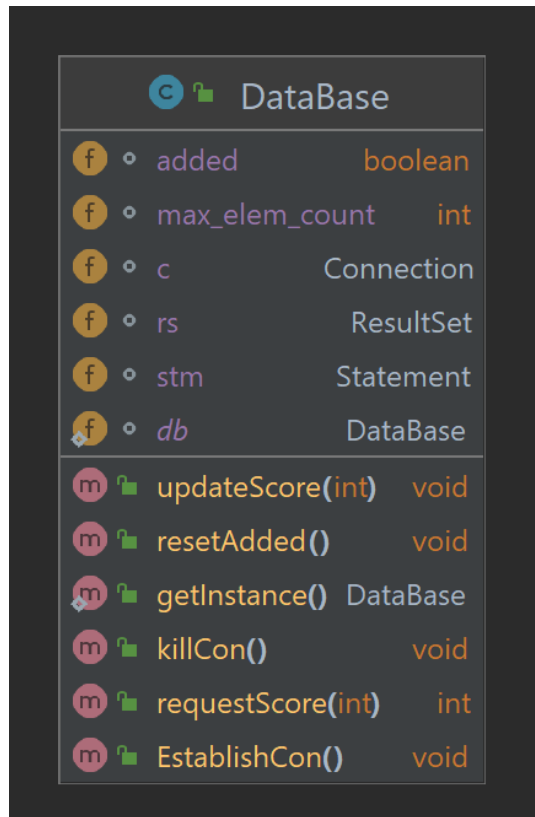
f. Pachetul GameWindow:

Acest pachet contine clasa `Game` care leaga toate celelalte pachete intre ele, în cadrul acestei clase sunt stocate instanțele pentru obiectele de tip `KeyManager`, `Player`, `AssetsSetter`, `SuperObject`, `CollisionCheck` și `TileManager`. Tot în aceasta clasa este prezent și loop-ul acestui joc implementat prin funcțiile `Run()` și `Update()`.



g. Pachetul DataBase:

Acest pachet se ocupă de procesarea datelor și introducerea acestor într-o bază de date;



7. Sabloane de Proiectare:

În acest proiect am folosit două sabloane de proiectare:

Factory:

Am folosit Factory pentru generarea de inamici specifici pentru fiecare hartă.

```
public class EnemyFactory {
    public boolean coinFlip() {
        Random random = new Random();
        return random.nextBoolean();
    }

    public Entity makeEnemy(int map_id, Game gp) {
        Entity enemy = null;

        switch (map_id) {
            case 0 -> {
                if (coinFlip())
                    enemy = new EnemyC(gp);
                else
                    enemy = new EnemyCpp(gp);
            }
            case 1 -> {
                if (coinFlip())
                    enemy = new EnemyJS(gp);
                else
                    enemy = new EnemyTS(gp);
            }
            case 2 -> {
                if (coinFlip())
                    enemy = new EnemyRS(gp);
                else
                    enemy = new EnemyPY(gp);
            }
        }

        return enemy;
    }
}
```

Singleton;

Acest șablon a fost folosit atât la Player cât și la DataBase pentru a preveni existența mai multor obiecte de aceste tipuri.

```
public class Player extends Entity {
    private static Player player = null;

    private Player(Game gp, KeyManager keyM) {
        super(gp);
        this.keyM = keyM;

        setDefaultValues();
    }

    public static Player getInstance(Game gp, KeyManager keyM) {
        if (player == null) {
            player = new Player(gp, keyM);
        }

        return player;
    }
}
```

```
public class DataBase {
    static DataBase db = null;

    private DataBase() {
        EstablishCon();
    }

    public static DataBase getInstance() {
        if (db == null) {
            db = new DataBase();
        }

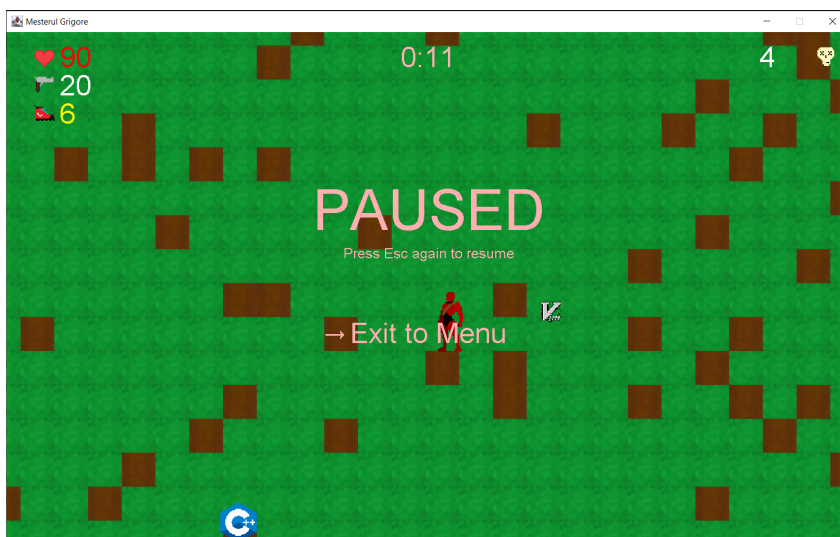
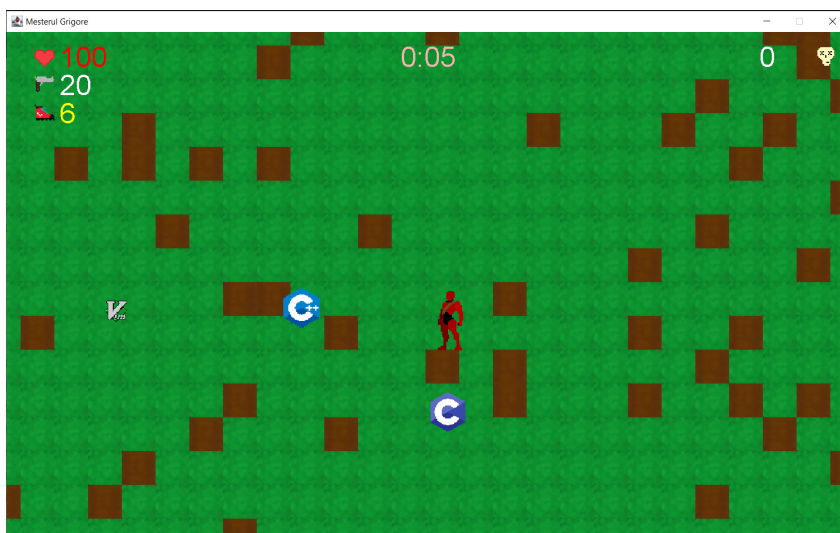
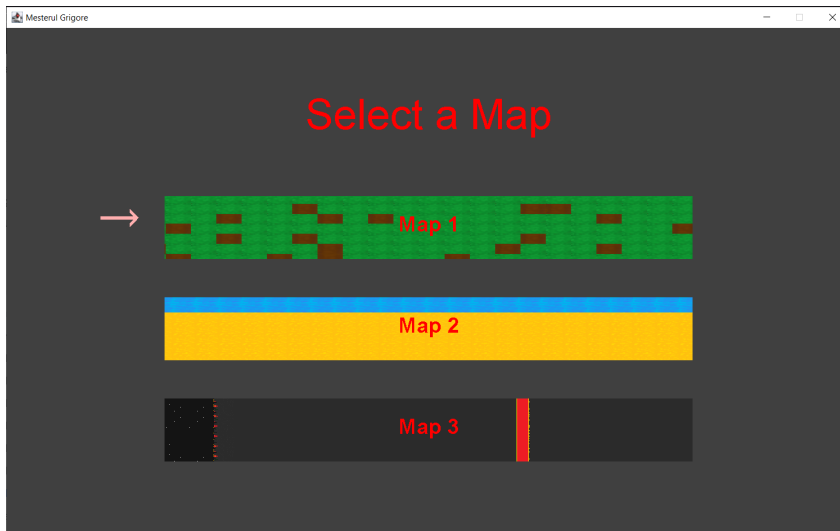
        return db;
    }
}
```

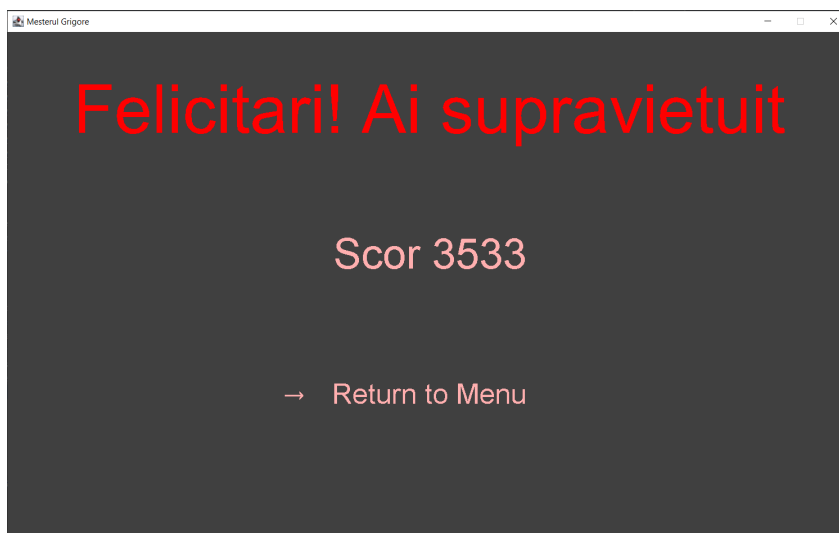
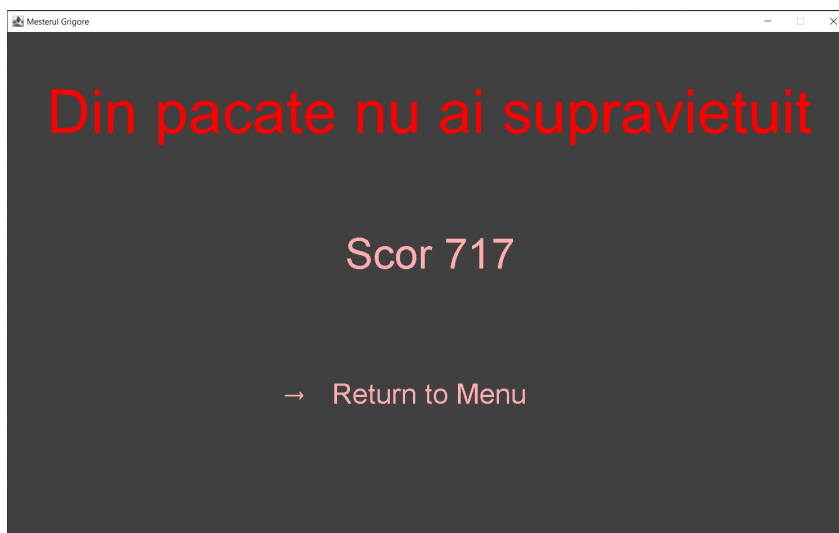
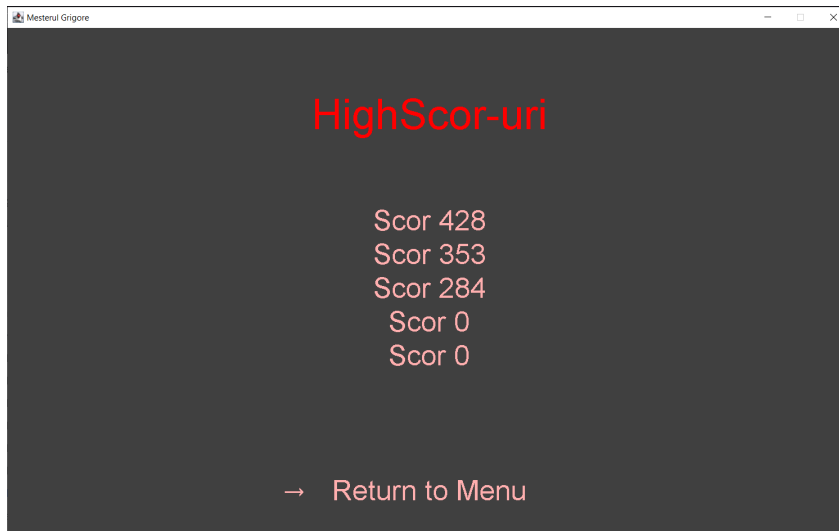
8. Despre:

În prezent condiția de câștig a jucătorului este să supraviețuiască 5 minute în cadrul unei mape. Dacă jucătorul reușește să reziste toate cele 5 minute fără să își piardă tot Hp-ul atunci nivelul se va opri deoarece jucătorul a câștigat. Dacă Hp-ul jucătorului ajunge la 0 în cadrul unui nivel atunci nivelul se va opri și se va considera că jucătorul a pierdut. În ambele cazuri jucătorul va fi întâmpinat la final de nivel de scorul obținut.

9. Gallery:







10.Bibliografie:

- Jocuri folosite drept inspirație: Vampire Survivors, Brotato
- Sprite-uri Inamici: <https://www.pngwing.com>
- Model Player: <https://sventhole.itch.io/prototype-hero-demo>