

# În inima Regatului

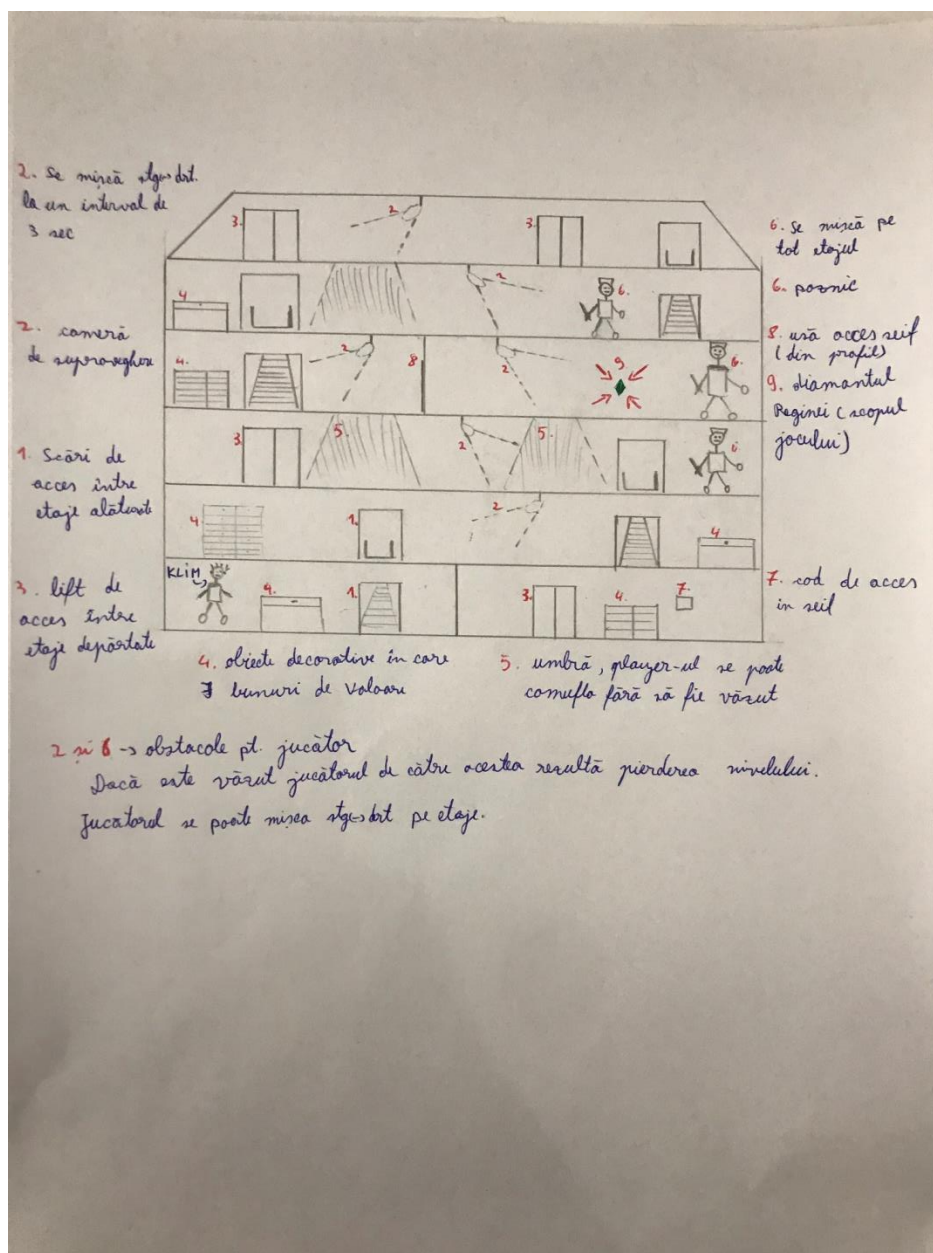
Savin Damian

1212B

## Proiectarea contextului:

Jocul este situat în epoca contemporană undeva într-o regiune numită Tammora, condusă de regele Visarion. Klim (personajul principal al jocului) lucrează în calitate de gardian al regelui, această ocupație fiind doar una temporară, întrucât el plănuiește să-și trădeze regele încercând să pună mâna pe cele trei diamante ale reginei Raissa, ascunse fiecare în trei palate diferite. Auzind fără să vrea discuția regelui cu unul dintre cei mai apropiați sfetnici ai săi, în care se dezvăluie detalii despre diamante și locațiile lor mai exacte din cele trei castele, Klim concepe un plan bine pus la punct, astfel încât să poată fura diamantele și să fugă din regat. Acesta nu o poate face de unul singur, are nevoie de un ajutor de încredere, ai vrea să-l ajuți pe Klim să fure diamantele și să fugă din regat?





## Proiectarea sistemului:

### Comenzi:



Stă pe loc



Merge



Atacă



Revendică bunuri

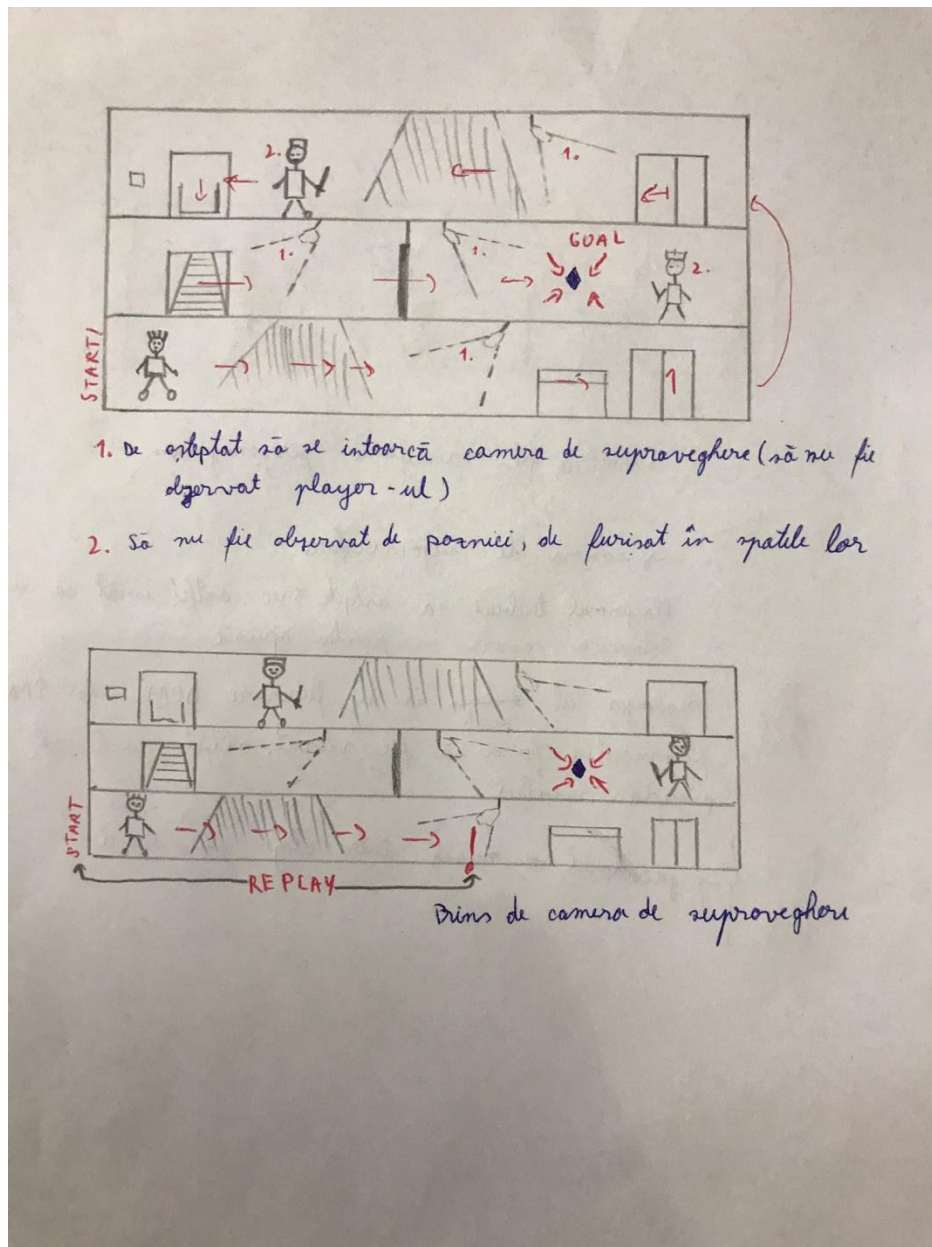


Aruncă proiectile

**Descrierea jocului:** Jucătorul urmează un traseu bine stabilit, format din etajele palatului, presărat cu obstacole, strecurându-se pe lângă camerele de supraveghere și pe lângă paznici.

Scopul jucătorului este să ajungă în seiful reginei și să pună mâna pe unul dintre cele mai valoroase diamante ale acesteia.

Jucătorul va primi un feedback, după fiecare nivel, în funcție de câte bunuri a mai găsit prin castel și în funcție de timpul pentru realizarea nivelului.



## Proiectare conținut:

Klim este protagonistul și jucătorul-personaj.



Paznicul este unul dintre obstacolele lui Klim.



## Animații:



**Prezentare joc:** Este un joc single player în care jucătorul se află într-un palat plin de obstacole, prin care trebuie să treacă neobservat folosindu-și abilitatea de agilitate și de camuflaj. Acesta câștigă când fură toate cele trei diamante.

**Temă joc:** Tema principală a jocului este aventura, întrucât le oferă jucătorilor o experiență plină de suspans și descoperire pe măsură ce progresează în poveste și înfruntă provocările întâlnite pe parcurs.

## Instrucțiuni și Reguli:

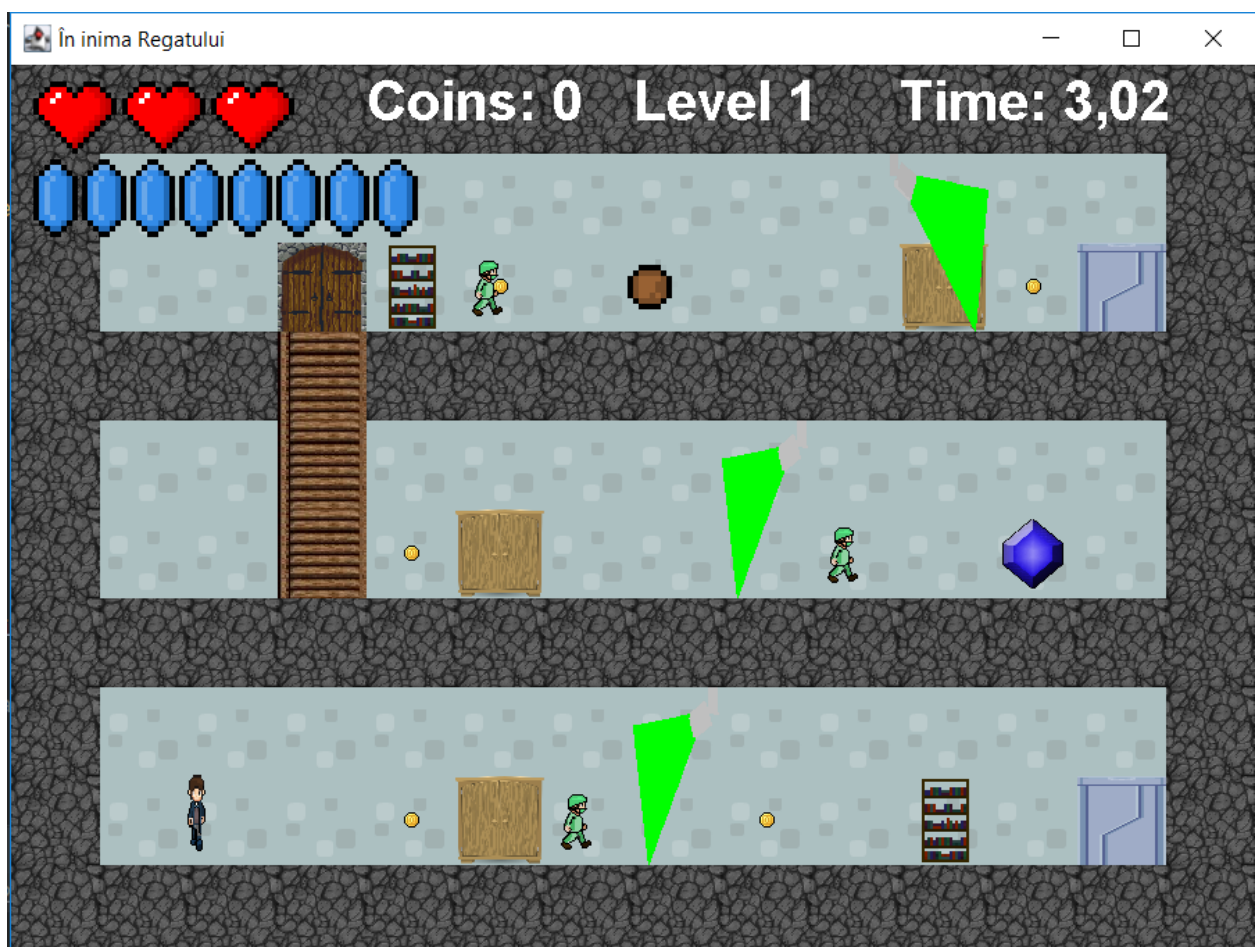
- Jucătorul controlează personajul prin apăsarea săgeților. Fiind într-un mediu terestru, săgețile corespund mișcărilor sus, jos, stânga, dreapta.
- Player-ul poate lovi paznici, aruncând în ei cu proiectile, apăsând tasta S, provocându-le daune, întrucât li se scade din viață. Ca să fie eliminat paznicul, sunt necesare trei lovituri cu proiectile.
- Player-ul are un număr limitat de proiectile, care se actualizează în funcție de cerințele nivelului. Dacă sunt terminate proiectilele în nivelul curent, player-ul nu mai primește proiectile în plus până la terminarea nivelului.
- Player-ul poate ataca și de aproape paznicul, folosindu-și sabia, apăsând tasta „Space”. În acest caz, sunt necesare șase lovituri de sabie pentru a elimina paznicul.
- Paznicul, la rândul lui, aruncă cu proiectile într-un mod aleatoriu, indiferent de pozițiile player-ului pe harta de joc. Dacă este nimerit, jucătorului i se scade din viață (acesta



având maxim patru vieți, în funcție de nivel). Sunt necesare trei/patru lovituri pentru eliminarea player-ului.

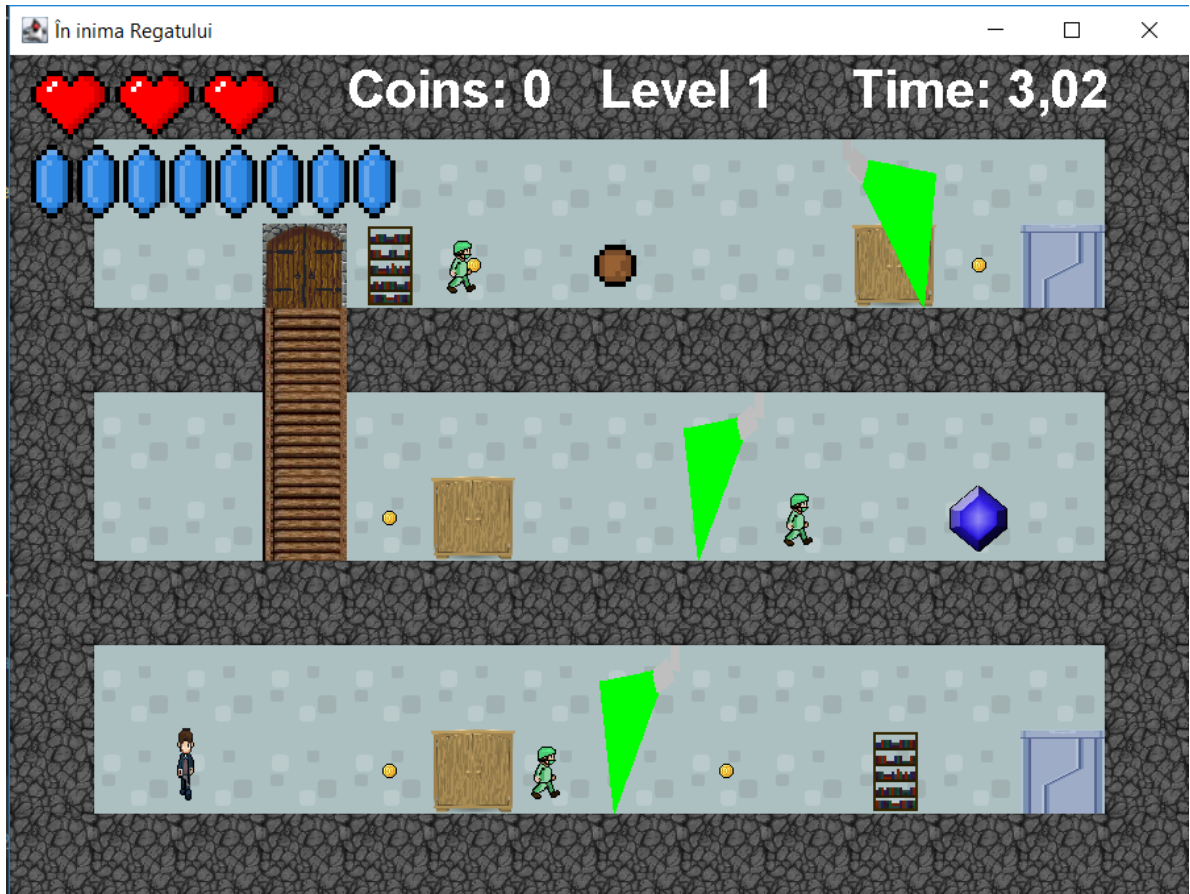
- Paznicul mai are o abilitate ascunsă. Dacă atinge jucătorul, sau este atins de către acesta (fără proiectile), paznicul provoacă daune player-ului, întrucât i se scade din viață. Dacă paznicul este atins de trei ori de către jucător, acesta este eliminat.
- Paznicul se mișcă doar stânga-dreapta la un interval stabilit de patru secunde.
- Jucătorul poate revendica bunurile ascunse (bănușii) prin palat doar trecând pe lângă ele, nefiind necesară apăsarea unei taste anume.
- Jucătorul trebuie să treacă de camerele de supraveghere, care se mișcă stânga-dreapta într-un mod aleatoriu. Dacă este atins de cameră, atunci nivelul este pierdut.

### Proiectarea interfeței cu utilizatorul:



## Proiectare niveluri:

### Nivelul 1:

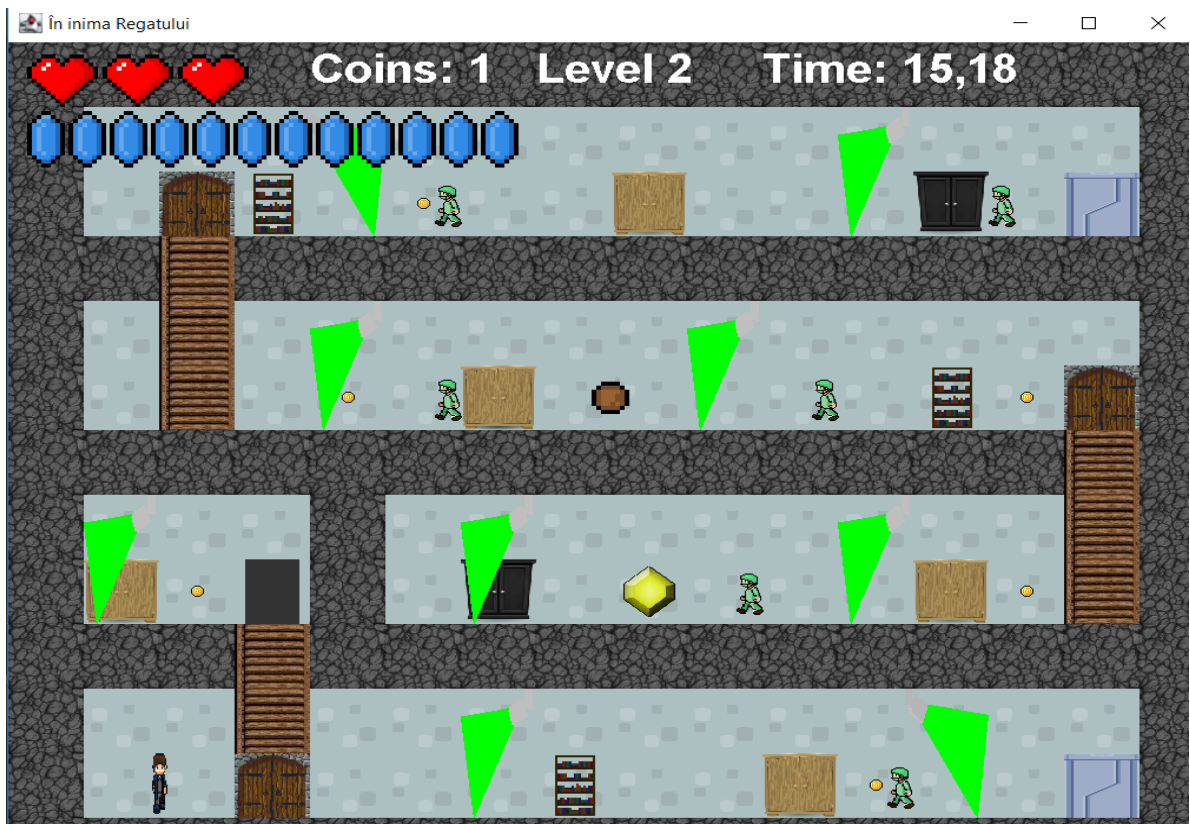


### Descriere niveluri:

**Nivel 1** ne aduce în primul castel al Reginei, unde se află ascuns în seif primul ei diamant. Dacă ai reușit să ajungi cu bine în seif, acolo se află un tunel care duce la al doilea castel, către nivelul 2.

**Nivel 2** ne aduce în al doilea castel al Reginei, unde se află ascuns în seif al doilea ei diamant. Dacă ai reușit să ajungi cu bine în seif, și să aduni un număr minim de 10 bănuți, acolo se află un tunel care duce la al treilea castel, către nivelul 3.

**Nivel 3** se petrece în castelul principal al Reginei. Pentru a câștiga trebuie să treci de paznici și de camerele de supraveghere și să colectezi un număr minim de 25 de bănuți.



## Game sprite:

Klim este protagonistul și jucătorul-personaj.



Paznicul este unul dintre obstacolele lui Klim.



Cameră de supraveghere:



Proiectile:



## Game tile-uri:



## Obiecte:



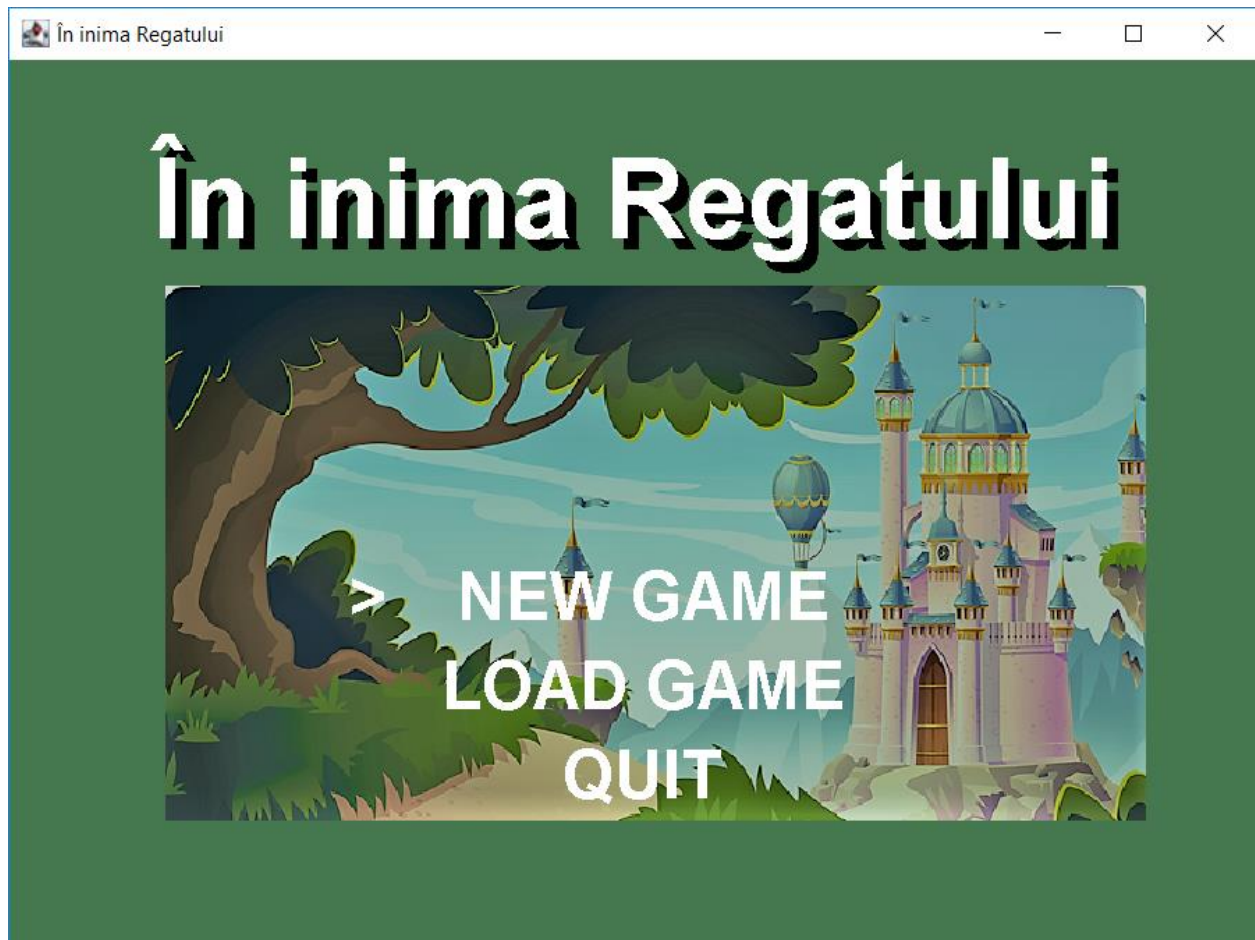


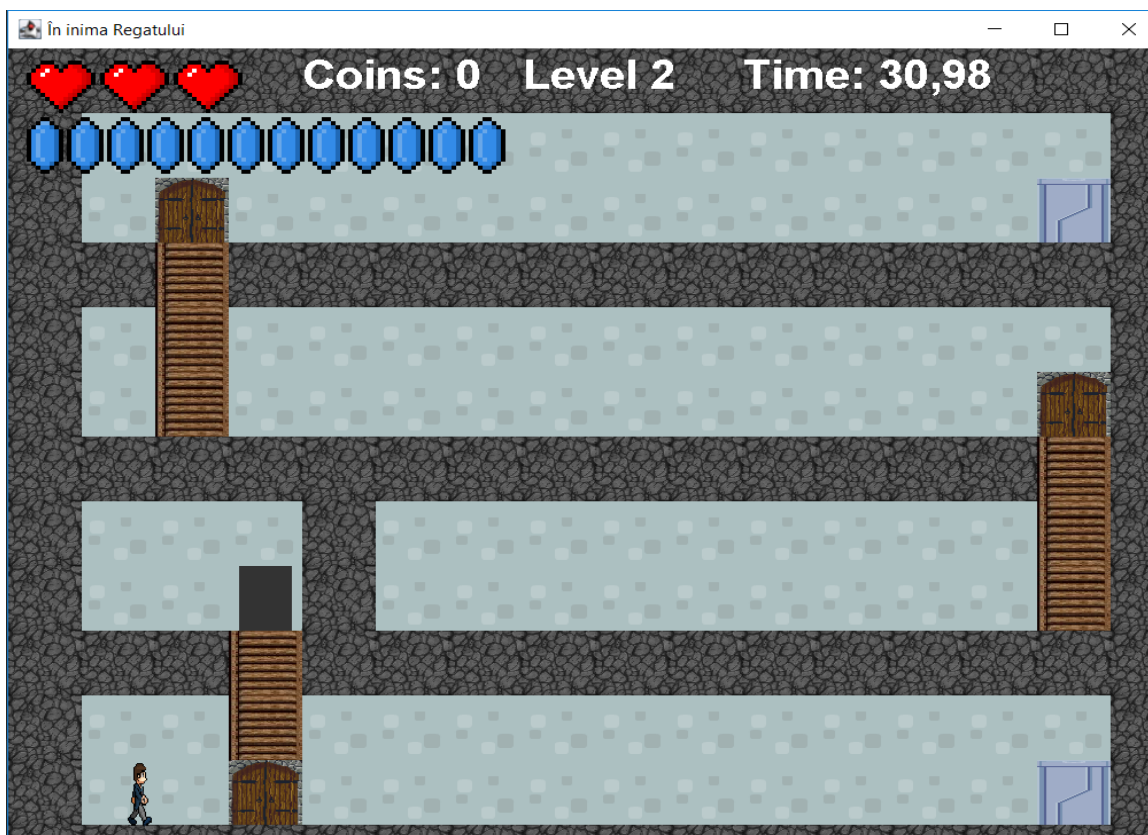
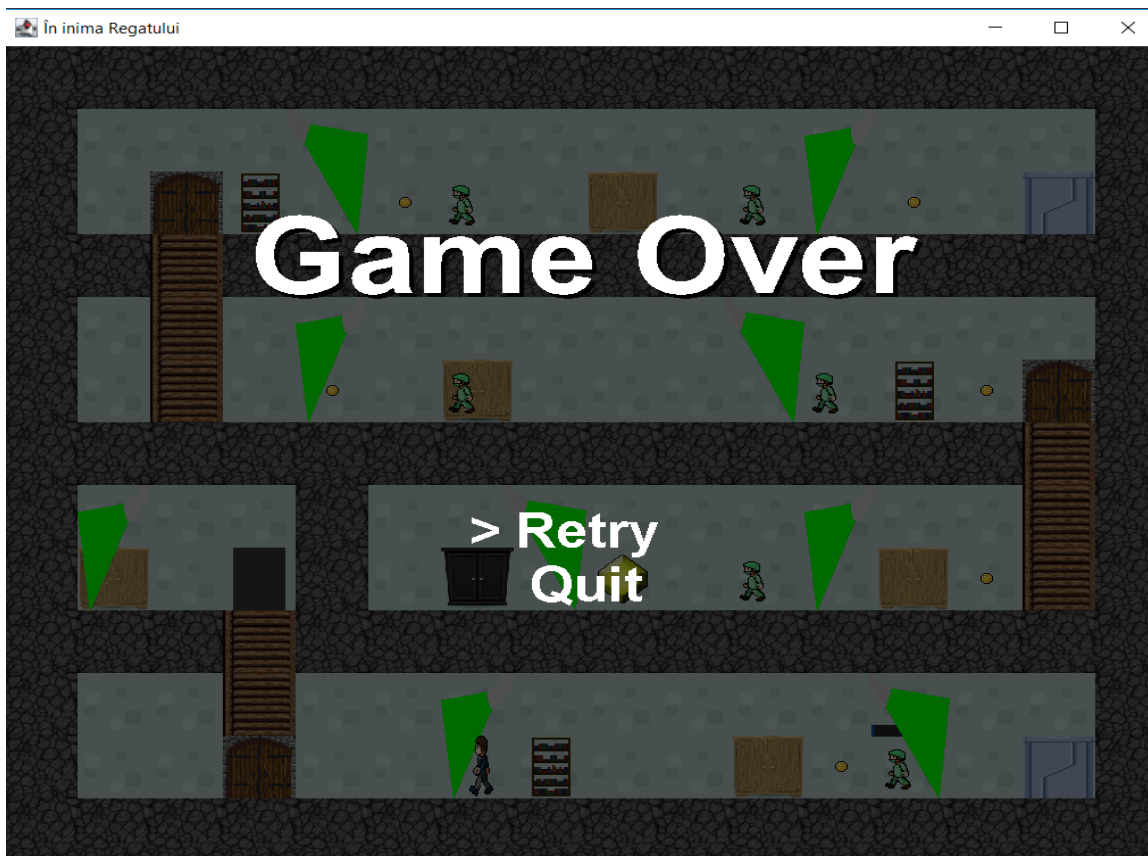
## Bibliografie:

- <https://opengameart.org/>
- <https://letsmakegames.org/resources/art-assets-for-game-developers/>
- <https://www.aseprite.org/>

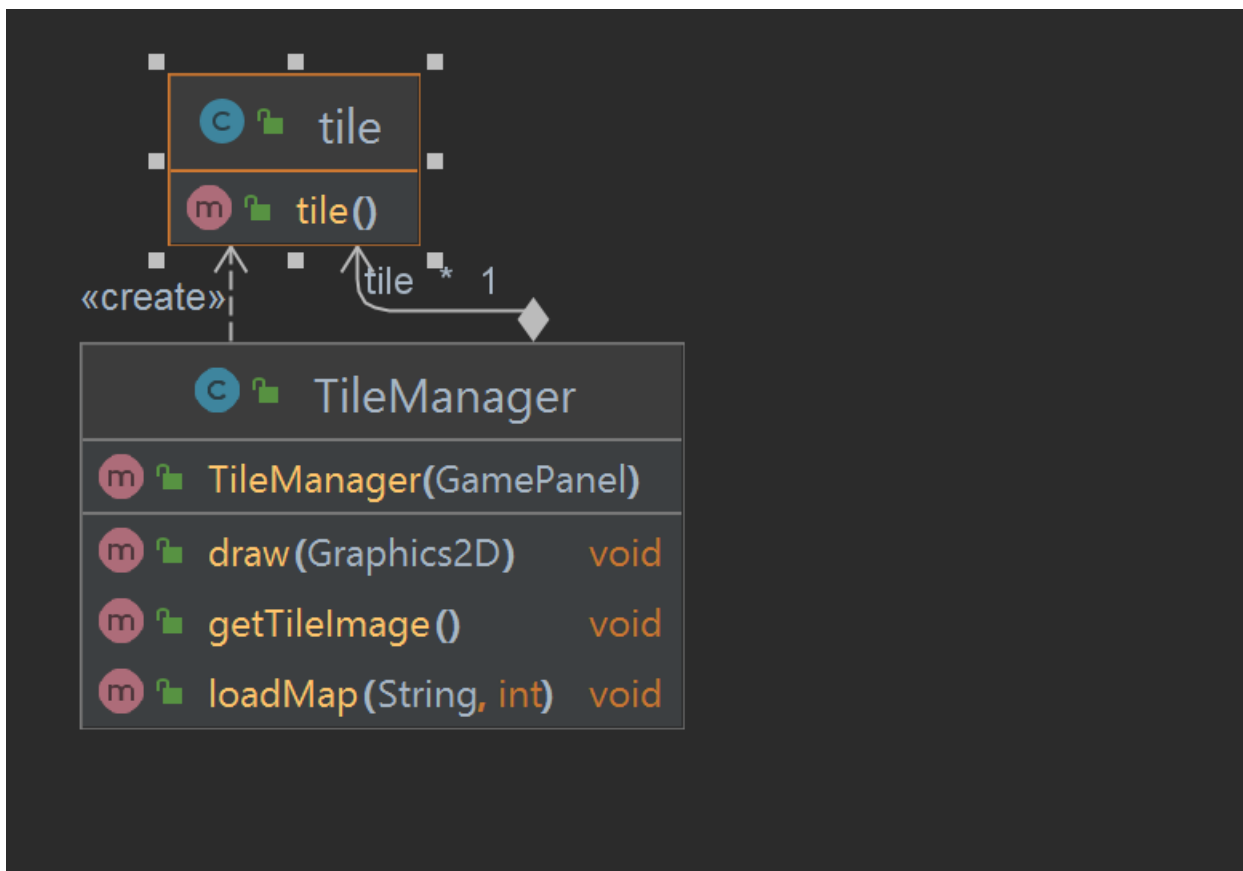
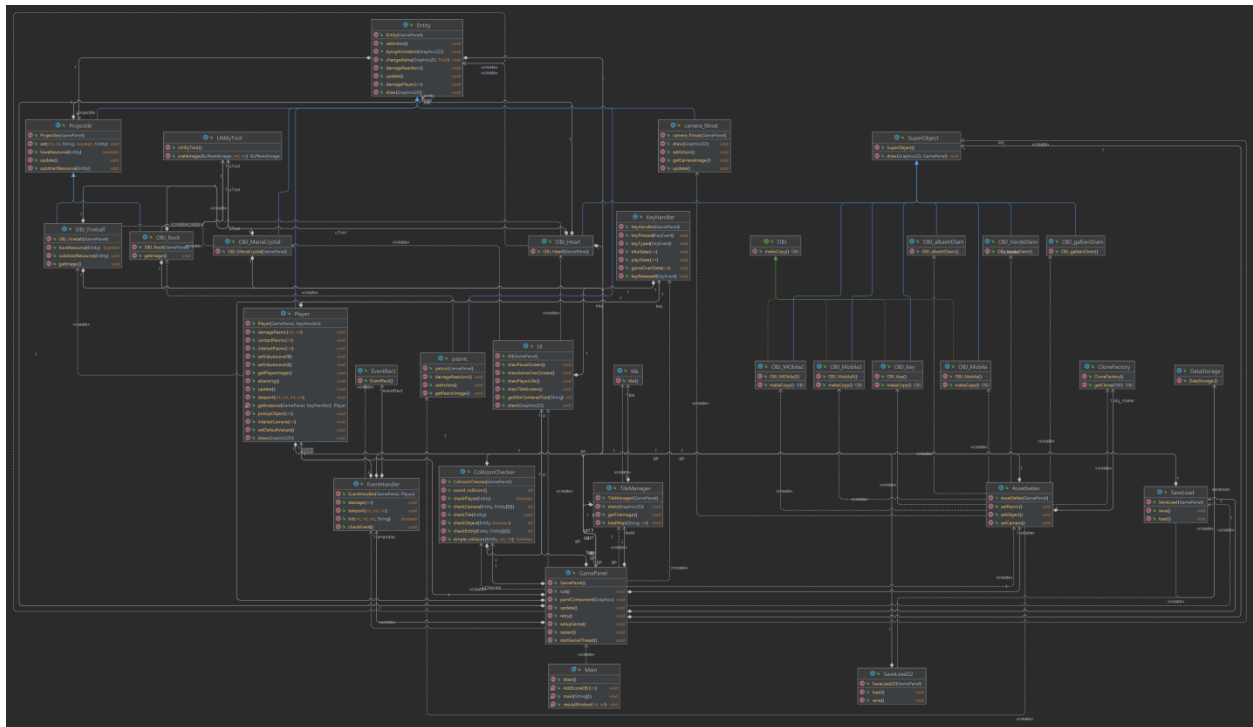
## Descriere meniu:

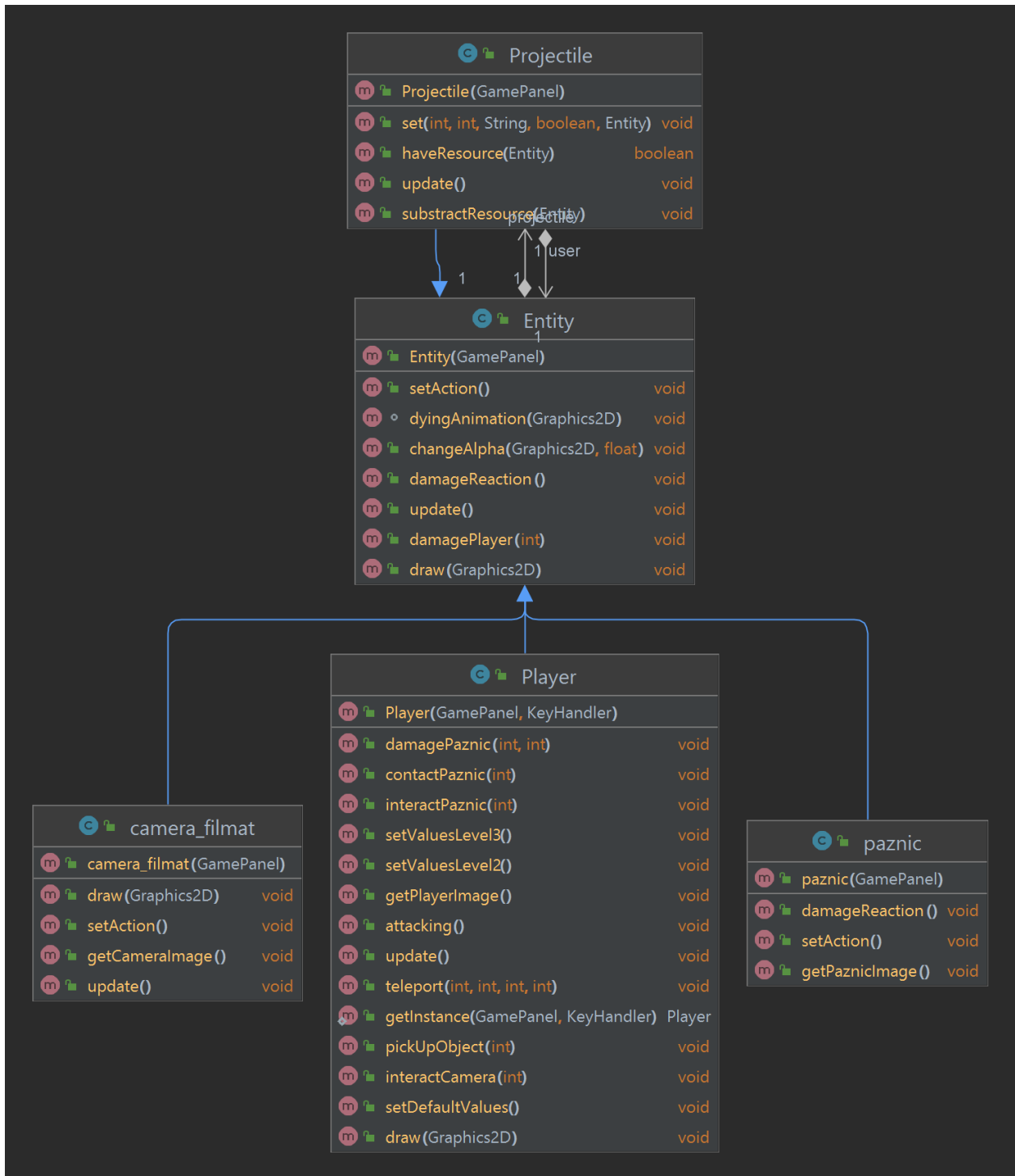
Jocul se deschide cu un meniu. Meniul va avea 3 butoane: New game, Load Game, Exit. „New game” va începe un nou joc.”Load Game” va încărca jocul în stadiul rămas anterior(se salvează automat jocul după trecerea fiecărui nivel). „Exit” va opri jocul indiferent în ce stadiu se afla.



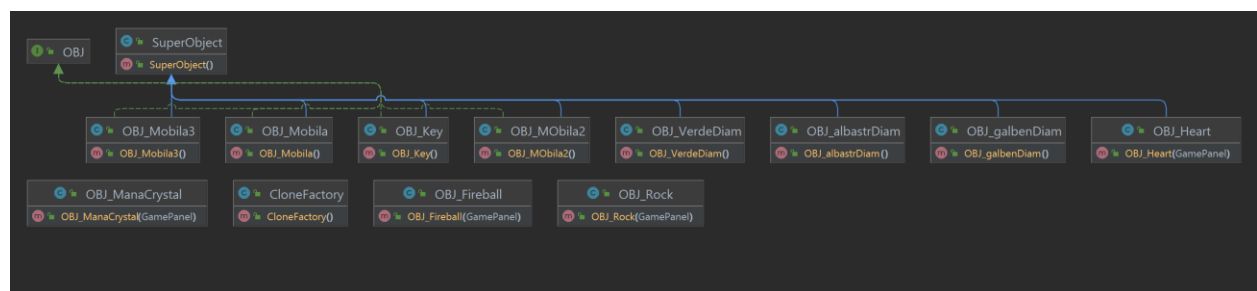
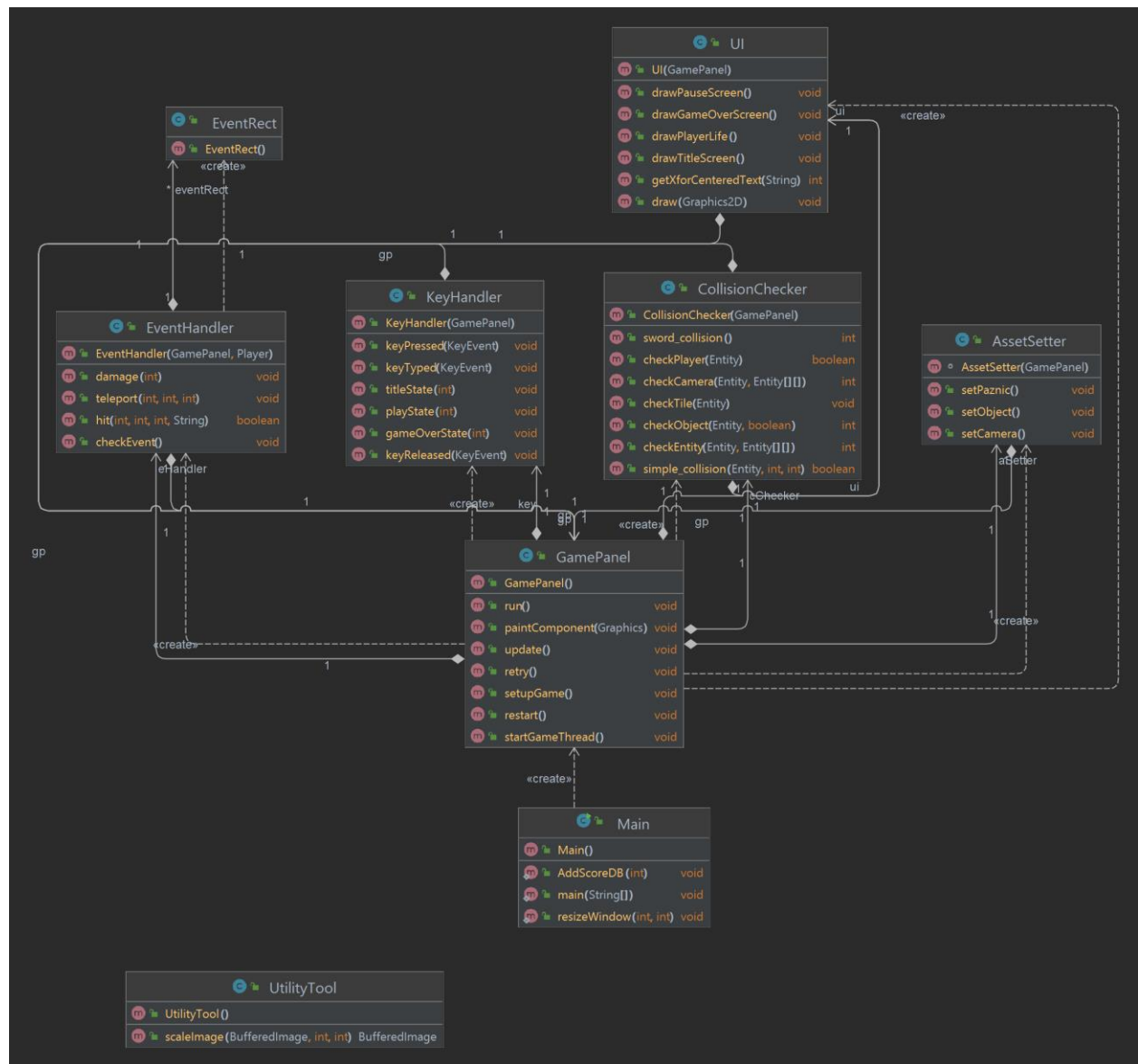


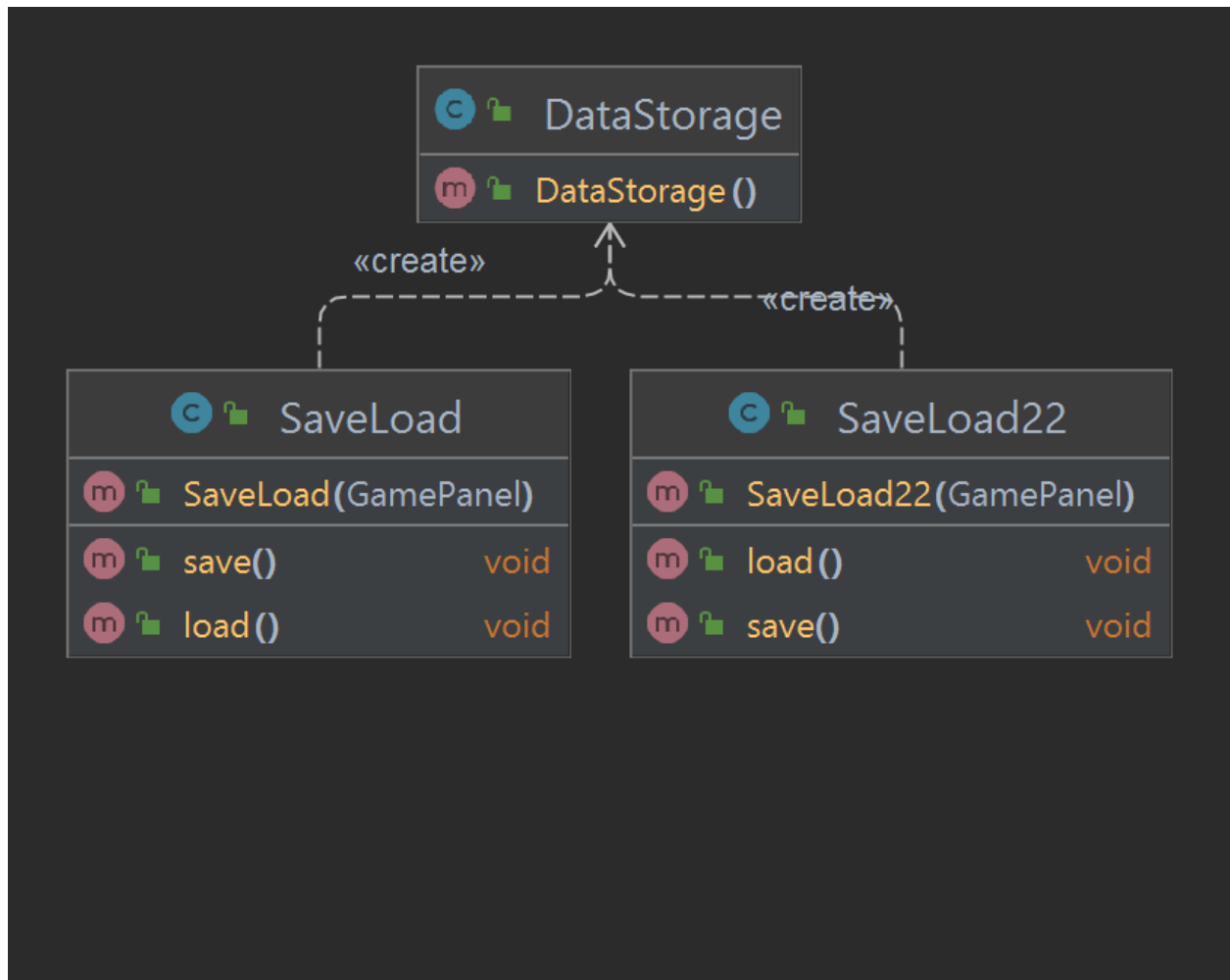
### Diagrame UML:











## Descriere clase proiect:

### Main

După cum se poate observa din diagrama prezentată mai sus, această clasă creează fereastră de început a jocului, creează o instanță de tipul `Game`, iar mai apoi pornește fluxul activităților prin apelul metodei `startGameThread()`.

### GamePanel

Rolul acestei clase este de a face inițializările (de exemplu de a crea instanțe pentru jucători/inamici, pentru a inițializa harta). Această clasă implementează interfața `Runnable` și extinde clasa `JPanel` pentru a avea comportamentul unui fir de execuție (thread). În momentul apelului metodei `startGameThread()` se instanțiază un obiect de tip `Thread` pe baza instanței curente a clasei `Game`. Orice obiect de tip `Thread` trebuie să implementeze metoda `run()` care este apelată atunci când firul de execuție este pornit (`start()`). Această metodă `run()` inițializează jocul prin crearea unei instanțe, 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 (`draw()`).

Metoda `update()` actualizează starea jocului (de exemplu: modifica poziția jucătorilor pe baza tastelor apăstate, schimbă poziția inamicilor folosind chiar tehnici de inteligență artificială, crează diferite tile-uri (dale), etc. Metoda `draw()` va desena pe interfața grafică modificările făcute de metoda `update()`. Interfața grafică este un canvas, făcând o analogie cu realitatea, poate fi considerată o pânză pentru desen, pe care sunt desenate diverse obiecte.

### **TileManager**

Această clasă reține informații despre tile-urile din joc. În clasa `GamePanel` există o instanță a clasei `Tile`, deoarece clasa `Tile` conține un vector cu obiecte tot de tipul `Tile`, așadar în acest vector vor fi stocate toate tile-urile din joc în așa fel încât acest vector poate fi parcurs și pentru fiecare element se apelează metoda `update()` și apoi `draw()`.

### **Assets**

Clasa este responsabilă de crearea și plasarea obiectelor și entităților în joc, folosind șablonul de proiectare `Prototype`.

### **UI**

Clasa este responsabilă cu realizarea interfeței și afișarea pe ecran a meniului jocului și afișarea pe ecran a stărilor jocului.

### **KeyHandler**

Clasa se ocupă de gestionarea tastelor. Aceasta verifică dacă anumite taste sunt apăstate prin suprascrierea metodelor `keyPressed(KeyEvent e)` și `keyReleased(KeyEvent e)`.

### **CollisionChecker**

Această clasă se ocupă cu implementarea principalelor coliziuni în joc. Coliziuni între player și tile-uri, player și obiecte, player și alte entități și coliziuni între entități și player.

### **EventHandler**

Clasa verifică și gestionează interacțiunile jucătorului cu diferite evenimente din joc (ex. Trecere dintre niveluri)

### **Entity**

Clasa este responsabilă cu crearea și gestionarea entităților jocului (player, paznic, camere de supraveghere, proiectile), oferă o structură comună și funcționalități generale necesare pentru interacțiunile și comportamentul acestora.

## **Player**

Aceasta conține metodele prin care se desenează și animează personajul principal.

## **SaveLoad**

Această clasă conține metoda save() care salvează într-un fișier text datele jocului, și de asemenea conține metoda load() care ne ajută să încărcăm jocul în stadiul în care l-am salvat.

## **Șabloane de proiectare:**

### **Singleton:**

Este folosit în clasele Player și GamePanel pentru a crea o instanță a obiectului “player”.

### **Prototype:**

Este folosit în clasa AssetSetter, pentru a crea clone ale obiectelor din clasele :OBJ\_Key, OBJ\_Mobila, OBJ\_Mobila2, OBJ\_Mobila3 cu scopul de a le folosi ulterior pentru instanțierea unor noi obiecte similare cu cele copiate.