

# Vincent's Legacy: Echoes of Darkness

Rotariu Tudor-Liviu

Grupa 1210B

## Povestea jocului:

Protagonistul jocului este Vincent, un Tânăr vrăjitor care abia și-a descoperit de curând puterile și este încă neinițiat în tainele magiei. Acesta trăiește în Tărâmul Întunecat în care magia neagră domină și peste care domnește Vrăjitoarea cea Rea, Desdemona. Înfricoșătoarea vrăjitoare Desdemona a aflat dintr-o veche profeție că un erou neînfricat va răzbi peste Tărâmul Întunecat și îi va pune capăt domniei. Într-o tentativă de a-și proteja imperiul malefic, ea a rostit un blestem puternic care a sustras puterile magice ale locuitorilor și a împiedicat nașterea celor binecuvântați cu daruri magice. Totuși, într-o casă săracă dintr-un colț al ținutului, se născu Vincent, un copil al căruia destin era scris și asupra căruia blestemul n-a avut efect.

Vincent crește și, la 16 ani, în timp ce hoinărea prin pădurea din spatele casei, fiind furios și având emoții intense, aprinsese fără să realizeze o grămadă de frunze uscate. Atunci realizase că are puteri magice, spre mirarea lui, a familiei și a unicei sale prietene, Prințesa Helena, care era chiar fiica vitregă a Desdemonei.

Vrăjitoarea cea Rea, aflând de la servitorii ei credincioși împrăștiati pe întreg regatul, că Vincent are puteri magice, realiză că despre el vorbea profeția. Desdemona se înfurie, și zboară până în sătucul în care trăiește Vincent și, sub privirea îngrozită a fiicei sale, lovește de trei ori cu bastonul ei magic în pământ, lucru care îl transformă pe băiat într-un schelet și dispără răzând malefic.

Helena, fiind la rândul ei o vrăjitoare исcusită și un maestru al poțiunilor, îl informă pe Vincent de existența unei poțiuni care îi poate recăpăta forma umană. Însă, pentru aceasta Vincent trebuie să se aventureze pe tărâmuri periculoase, în căutarea ingredientelor necesare.

## Mecanica jocului:

Jocul va începe cu meniul principal, din care jucătorul poate alege dacă să înceapă un nou joc, să selecteze un progres deja existent sau să părăsească jocul.

Dacă jucătorul alege LOAD GAME, va fi trimis la nivelul cel mai ridicat la care a ajuns în acel gameplay. De exemplu, dacă jucătorul a murit la nivelul al doilea și a ieșit în meniul principal, în load game va fi stocat nivelul al doilea, până când va reuși să treacă la nivelul al treilea.

Jocul este finalizat și câștigat în momentul în care jucătorul a trecut prin toate cele trei niveluri și a colectat minim 20 de obiecte la fiecare din ele. În momentul în care Vincent colectează poțiunea, redenevine om iar jucătorul termină jocul.



La începerea unui nou joc, jucătorul va fi introdus în universul fictiv prin redarea poveștii, sub forma unor imagini. Acestea se vor derula la click-ul jucătorului. De asemenea, finalul va consta tot dintr-o serie de cutscene-uri, care vor prezenta deznodământul poveștii.



Desdemona, the Evil Witch, reigns over the Dark Realm. Here, dark magic dominates, and there seems to be no salvation.



Desdemona learns from a prophecy that a young wizard from her kingdom will dethrone her and bring about her demise. The witch casts a curse, preventing the birth of babies with magical powers.



Many years later, in a poor family, in a humble house, Vincent is born, the boy foretold by the prophecy, a baby with magical powers that no one knew about.



At the age of 16, while strolling through the forest with his only friend, Helena, intense emotions cause him to magically ignite a pile of dry leaves, much to everyone's surprise.



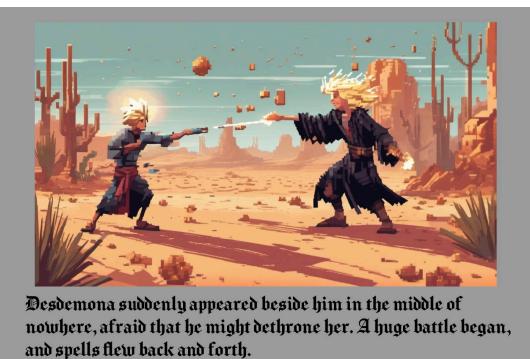
Desdemona, upon learning from her faithful servants about Vincent's powers, becomes furious and flies to the boy's village. She firmly strikes the ground three times with her staff, transforming him into a skeleton.



Helena was none other than Desdemona's stepdaughter, whom she hated with all her heart. Helena promises Vincent that she will help him by preparing a magical potion, for which she will need a few additional ingredients.



The second Vincent's lips touched the magic potion, he regained human form, to his and Helena's amazement. However...



Desdemona suddenly appeared beside him in the middle of nowhere, afraid that he might dethrone her. A huge battle began, and spells flew back and forth.



Vincent, now human again and regaining his powers at their true intensity, sends the most powerful spell ever known. He strikes Desdemona right in the heart, and she disintegrates into thin air, leaving behind only a macabre scream.



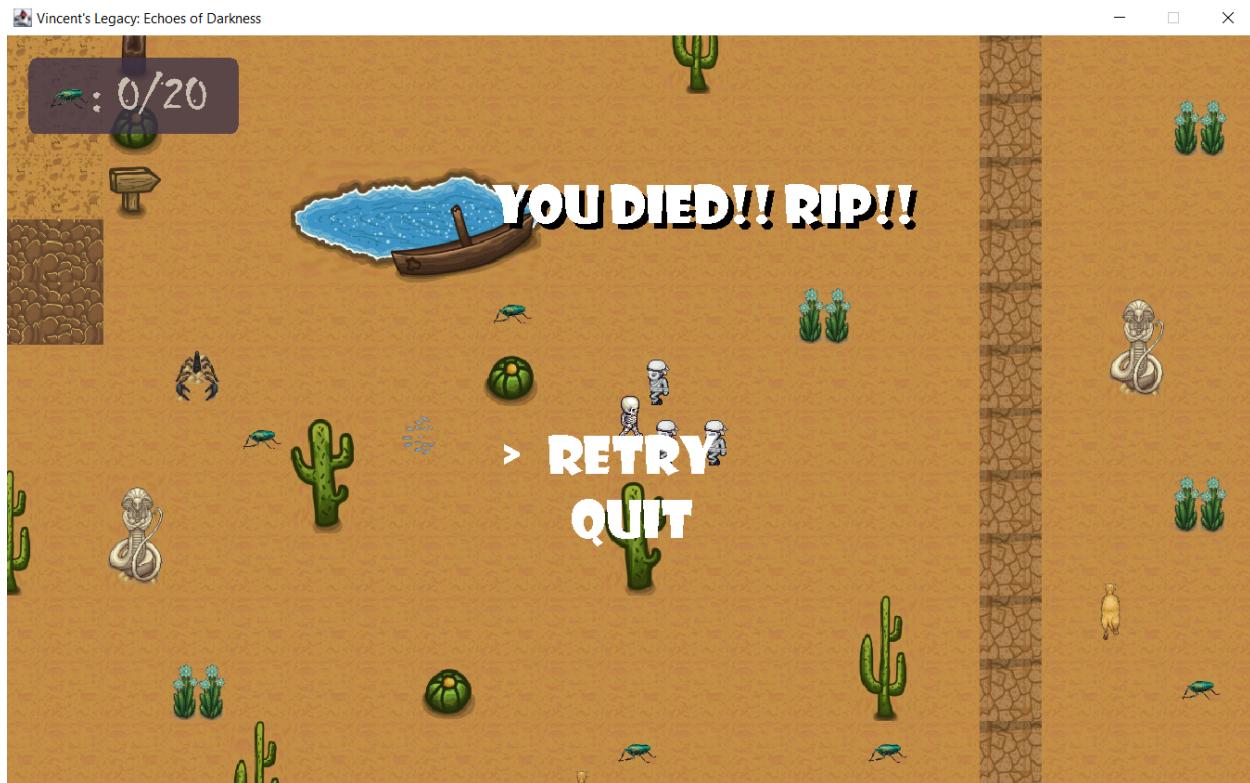
Finally! Vincent has defeated Desdemona. The Dark Land has been destroyed, and peace and serenity have been restored to the universe! All was well at last.

Jocul va fi unul singleplayer, de tip actiune/puzzle, întrucât protagonistul trebuie să se băta cu anumiți inamici, dar scopul principal al jocului este de a colecta toate obiectele cerute.

Deși jocul este unul 2D, proiecția grafică este una trei sferturi ( $\frac{3}{4}$ ), dând iluzia unui fals 3D. Caracterul se poate mișca pe hartă sus-jos-stânga-dreapta. Puterea lui specială, vrăjitor fiind, este că poate arunca vrăji (tasta spațiu). Inamicii lui pot, de asemenea, să arunce vrăji, cu excepția dovlecilor de la primul nivel care îți scad viața doar dacă îi atingi. Protagonistul are un health bar, care scade la impactul cu vrăjile inamici.

În partea de sus a ecranului, apare o casetă care va indica numărul de obiecte colectate până la acel moment, cât și una care indică viața protagonistului.

În momentul morții lui Vincent, va apărea meniul de Game Over, de unde jucătorul are 2 opțiuni: RETRY, prin care va relua jocul cu viață maximă din punctul în care se află, sau QUIT, de unde va ieși în meniul principal.



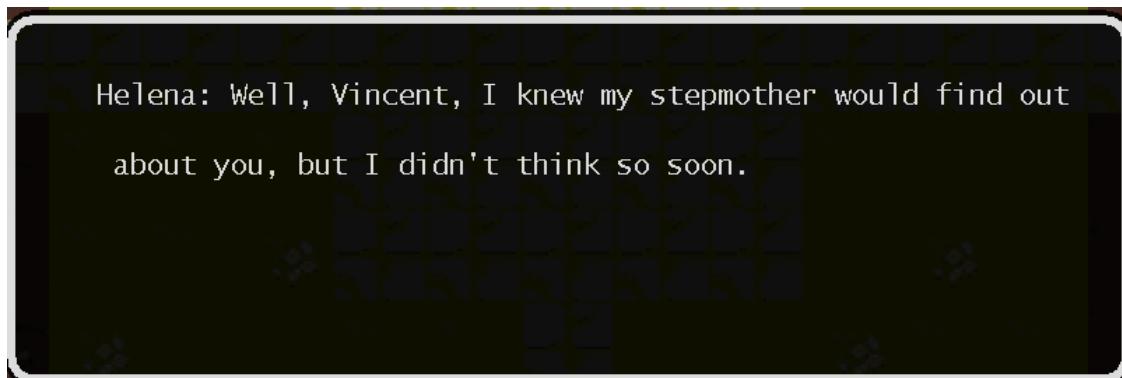
#### Instrucțiuni:

Jucătorul va folosi tastele W/A/S/D pentru a deplasa personajul pe hartă, tasta SPACE pentru a arunca o vrajă spre un adversar iar pentru a interacționa cu un NPC de pe hartă este necesară apăsarea tastei E pentru deschiderea casetei de dialog. Acestea se vor derula tot prin apăsarea tastei E, aceeași tasta fiind folosită și pentru închiderea modului dialog. Tasta P este folosită pentru intrarea în meniul de pauză, meniu care va afișa atât viața curentă a protagonistului cât și poziția lui pe hartă, prin intermediul unei baze de date. Datele din bază se sterg în mod automat atâtla deschiderea jocului, cât și la trecerea la un nou nivel.



### Casete de dialog:

La interacțiunea MC-ului cu NPC-urile, se va afișa o casetă de dialog de forma aceasta:



### Personaje:

#### 1. Vincent - Main Character

Vincent este Tânărul vrăjitor transformat într-un schelet. Conform profetiei, este singurul care o poate opri pe Desdemona, Vrăjitoarea cea Rea. Fiind vrăjitor, chiar și sub forma de schelet, puterea lui este de face vrăji. Astfel, Vincent aruncă vrăji care lovesc inamicii făcându-i să le scadă viața. De asemenea, Vincent însuși are un health bar, care scade pe măsură ce este lovit de inamici. La final, după ce bea poțiunea magică, își recapătă forma umană. Scopul lui pe parcursul jocului este să adune un număr de obiecte la fiecare nivel.

#### 2. Desdemona - final boss

Desdemona este Vrăjitoarea cea Rea, care, speriată de profetie, îl metamorfozează pe Vincent într-un schelet. Aceasta este învinsă, într-un final, de Vincent iar Tânărul Întunecat este înfrânt.

### 3. Helena - prietena lui Vincent

Helena este fiica vitregă a Desdemonei, pe care o urăște în secret. Luptă, împreună cu Vincent, pentru detronarea ei și căderea Tărâmului Întunecat. Este pricopută la potiuni, și cu ajutorul ei Vincent își va recăpăta înfățișarea umană.

### 4. Inamici

La fiecare nivel vor exista diversi inamici, vrăjiți de Desdemona pentru a-i face treaba murdară, și anume să-l opreasă pe Vincent din a colecta ingredientele necesare transformării lui în om.

#### a. Dovlecii

La primul nivel, prin cimitir mișună dovleci fermecați, care se mișcă caotic și îi scad viața lui Vincent la contactul cu acesta.

#### b. Goblinii

La nivelul al doilea, Desdemona a trimis niște goblini să apere jungla magică. Aceștia aruncă proiectile de foc, care îi pot răni pe Vincent.

#### c. Mumile

Cel de-al treilea și ultimul nivel aduce în scenă mumiile, entități fermecate care aruncă cu bucăți de sticlă și sunt vrăjite de Desdemona să îi adulmece urma lui Vincent și să îl urmărească, cu scopul de a-l ucide.

## Nivel 1:

Nivelul începe cu protagonistul, Vincent, sub înfățișarea unui schelet, în mijlocul hărtii, lângă Helena. Ei se află într-un cimitir. La contactul cu Helena (apăsarea tastei E), aceasta va începe să poarte un dialog cu protagonistul.

Helena: „Ei bine, Vincent, bănuiam că mama mea vitregă va afla de existența ta, dar nu credeam că va fi atât de curând.”

Vincent: „Ce putem face?”

Helena: „În primul rând, trebuie să te transformăm din nou în om. Nu o poți învinge ca schelet. Partea bună e că manualul meu de potiuni are tot ce avem nevoie. Doar că...”

Vincent: „Doar că ce?”

Helena: „Ne lipsesc câteva ingrediente esențiale. Trebuie să călătoresc prin trei lumi diferite pentru a le obține.

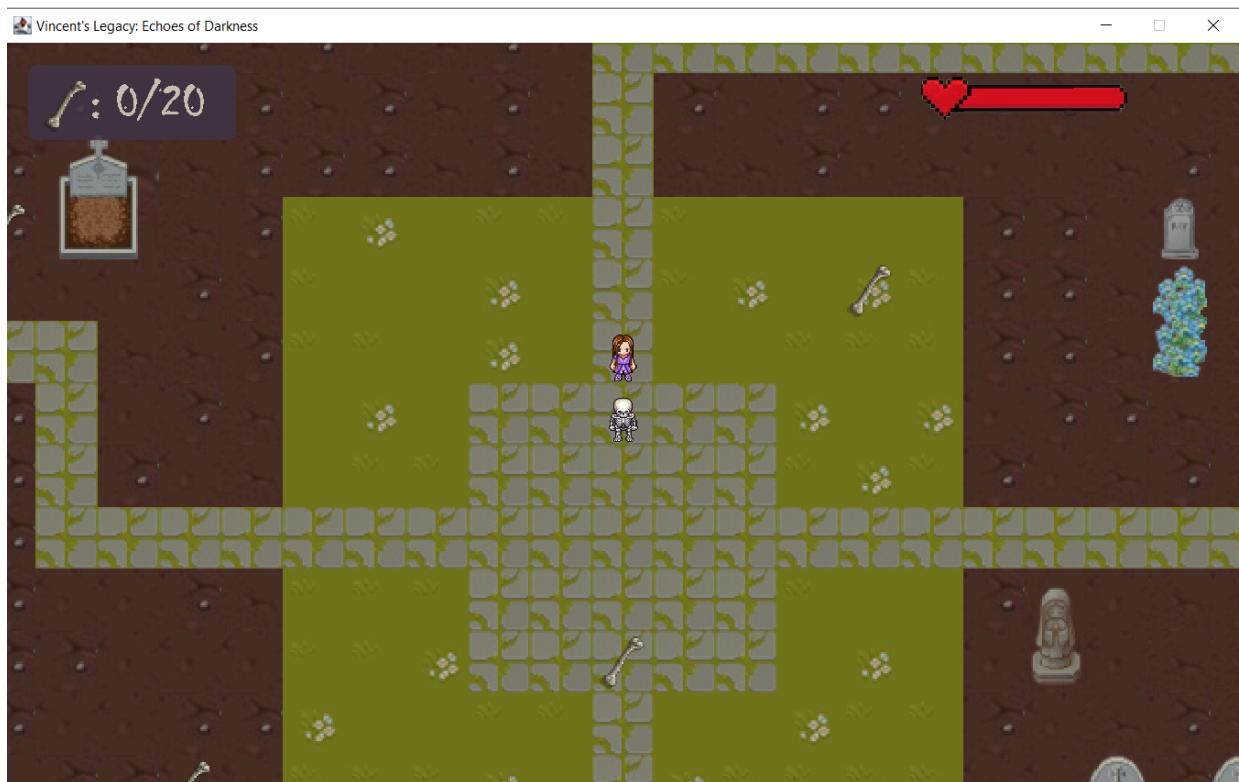
Vincent: „La ce te referi?”

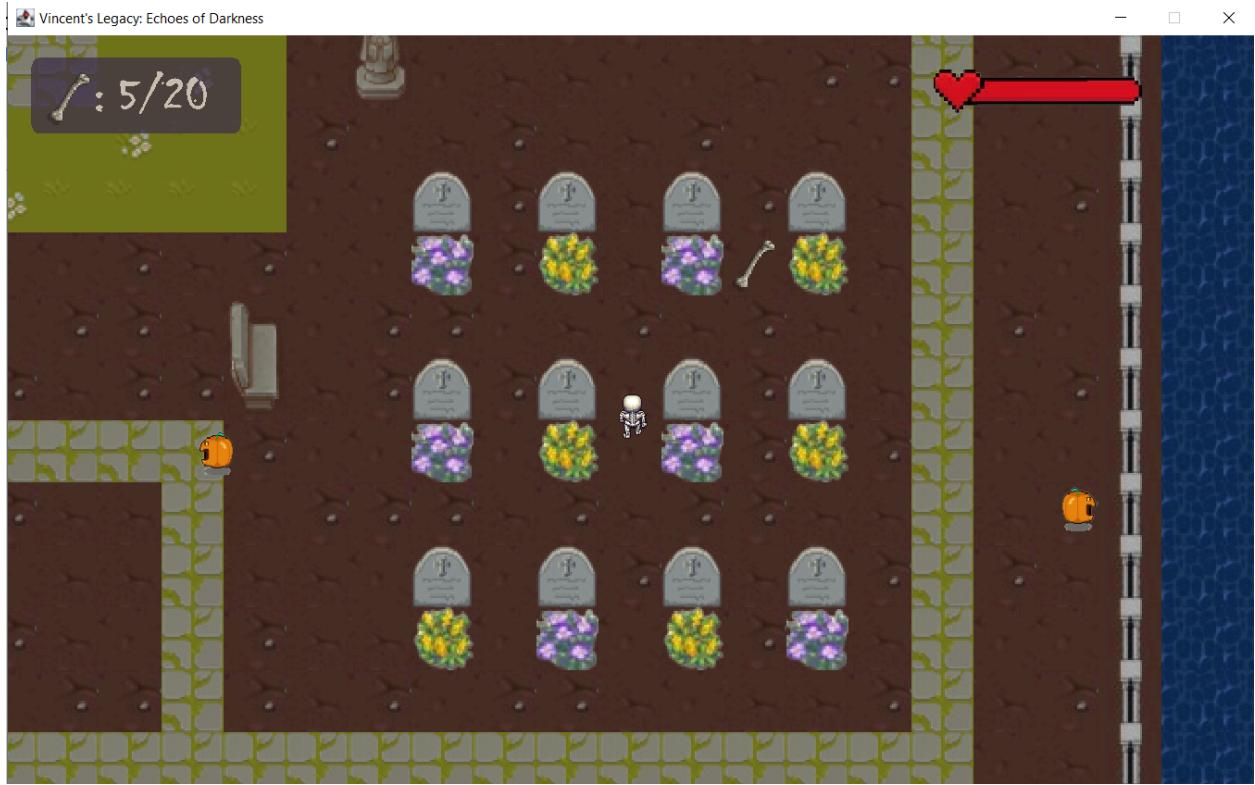
Helena: „Probabil îți se va părea ciudat dar... am nevoie de 20 de oase din cimitir, pentru a stabili o legătură cu lumea de dincolo și a-ți realiza regenerarea. Întoarce-te la mine când le ai pe toate.”

Primul nivel este unul de acomodare pentru jucător cu mecanica jocului, aşa că singurii inamici din acest nivel sunt niște pumpkins, care sunt distruiți dintr-o singură vrajă de-a lui Vincent. Pot provoca scăderea vieții lui Vincent doar dacă îl ating. Pe hartă sunt plasate 23 de oase, iar jucătorul trebuie să colecționeze minim 20, apoi să se întoarcă la Helena. În partea din stânga sus există un contor, pentru a arăta numărul de oase colectate. Va urma un dialog de genul:

Helena: „Ce bine, te-ai întors! Și mi-ai adus ce ți-am cerut! Voi deschide un portal către următoarea dimensiune din care trebuie să te aventurezi!”

Un portal se spawnează lângă cei doi, iar la contactul cu acesta, nivelul 1 se va încheia.

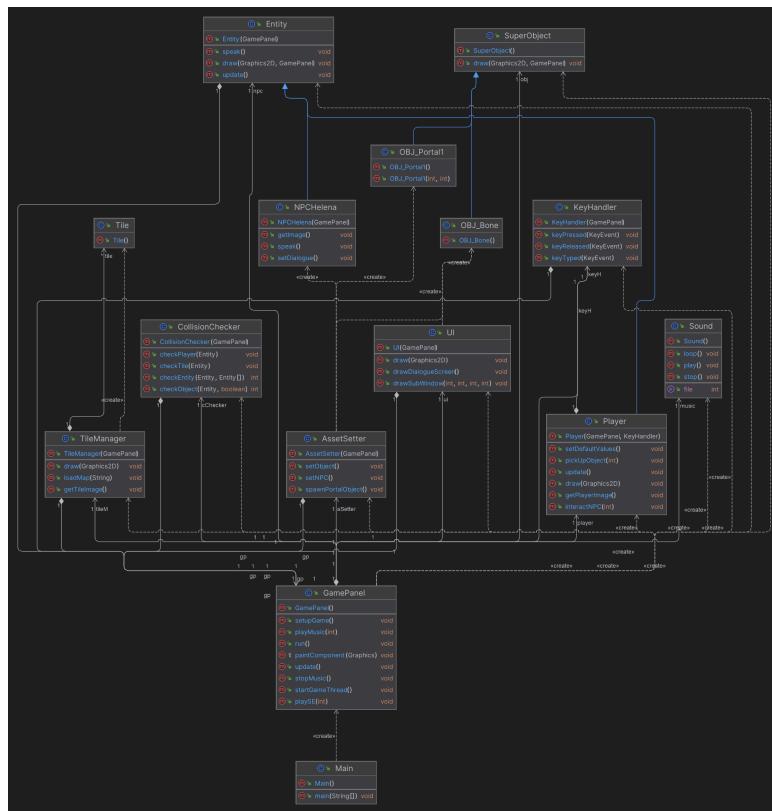




De colectat la nivelul 1:



Diagrama UML în urma finalizării etapei 1:



## Nivel 2:

Helena îl teleportează pe Vincent într-o junglă. Aici îl informează că, pentru poțiune, îi sunt necesare 20 de nuci de cocos, care cresc doar în adâncul junglei. Înamicii din acest nivel sunt niște goblini verzi, creați de Vrăjitoarea Malefică, care mișună prin junglă.

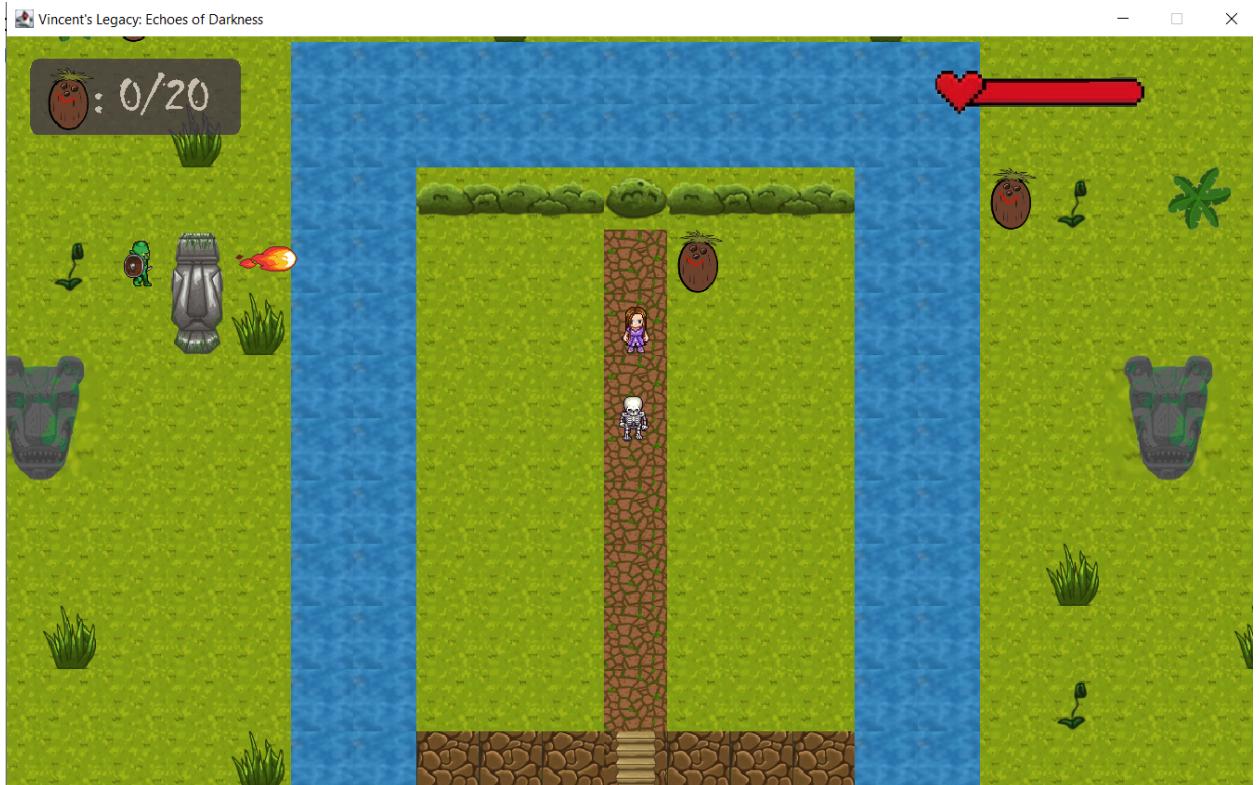
Față de nivelul precedent, se aduce un plus. Pe hartă este situat un NPC care îți aduce un side quest. Este vorba de un arheolog, îngrijorat că a pierdut în junglă cheia care deschide un templu străvechi incaș. Arheologul îl roagă pe MC să găsească cheia rătăcită prin jungla deasă și să i-o înapoieze. Jucătorul poate, însă, termina nivelul și fără să îndeplinească acest side quest. Dacă, însă, îl duce la bun sfârșit, arheologul îi oferă o recompensă. Va avea loc un dialog între cei doi, arheologul va face o glumă cum că Vincent este slab precum un schelet și îi oferă o felie de pizza. Aceasta îi maximizează health bar ul lui Vincent.

După ce aduce îi nucile de cocos, Helena deschide un nou portal spre o a treia lume.

De colectat la nivelul 2:



Screenshot-uri din joc:





### Nivel 3:

Vincent și Helena se teleportează într-un deșert. Aici are loc un alt dialog între cei doi. Helena îl trimite pe Vincent să colecteze 20 de scarabei vrăjiți, pentru protecție împotriva maleficului, și îl asigură că e ultimul ingredient necesar. Înamicii din acest nivel sunt mumii, care îl urmăresc pe Vincent când intră în raza lor de acțiune și îl atacă folosind bucăți de sticlă.

De asemenea, și aici jucătorul poate îndeplini un side quest. În mijlocul pustietății aride, un negustor beduin disperat îl abordează pe Vincent. Acesta îi povestește tragedia sa: cele trei cămile ale sale au dispărut într-o furtună de nisip. Negustorul îl roagă pe Vincent să recupereze cămilele și să i le aducă înapoi. Cele trei cămile se mișcă pe hartă, iar Vincent trebuie să le caute și să le colecteze. Ca și la nivelul anterior, jucătorul nu este constrâns să finalizeze side quest ul, dar dacă o face, primește o recompensă care îi restaurează health bar ul. Negustorul îl vede pe Vincent obosit și îi oferă o ceașcă de cafea.

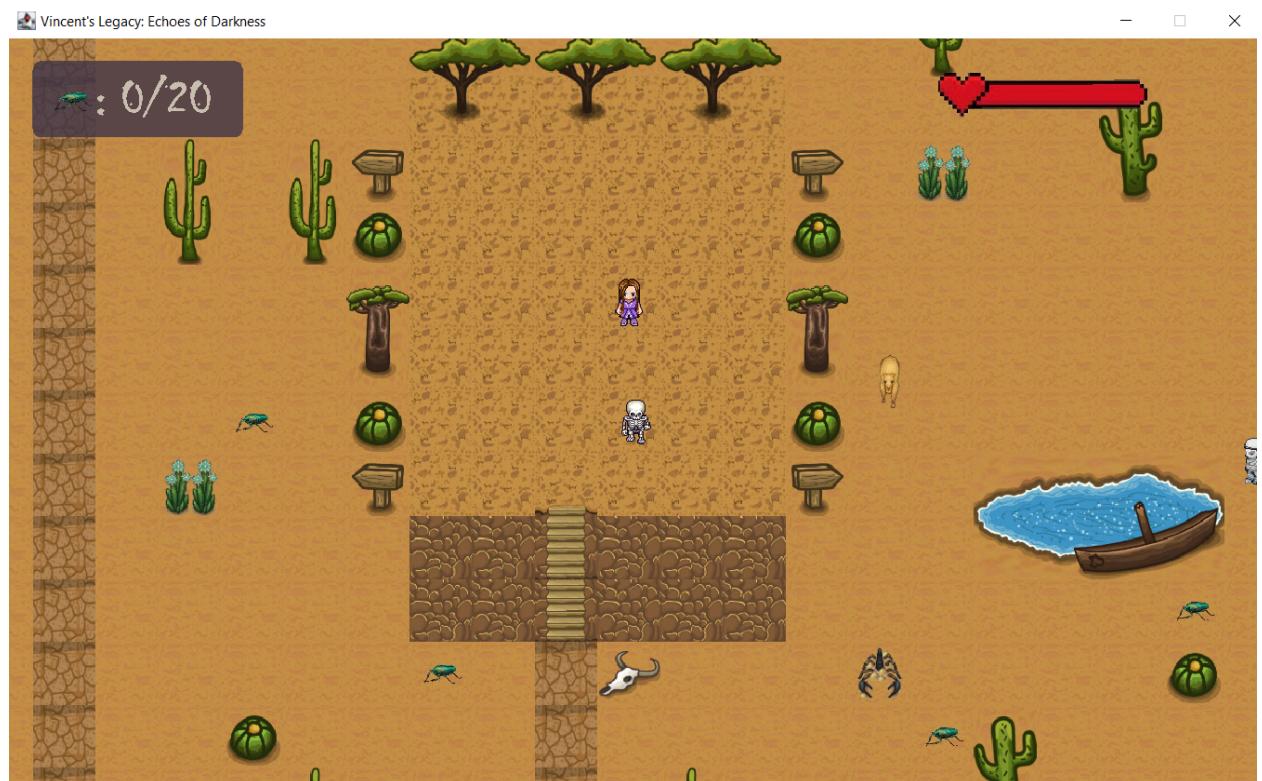
După ce colectează cei 20 de scarabei și îi aduce Helenei, aceasta finalizează poțiunea care apare lângă ei. Jucătorul trebuie să o ia, fapt care îl va transforma pe Vincent din nou în om.

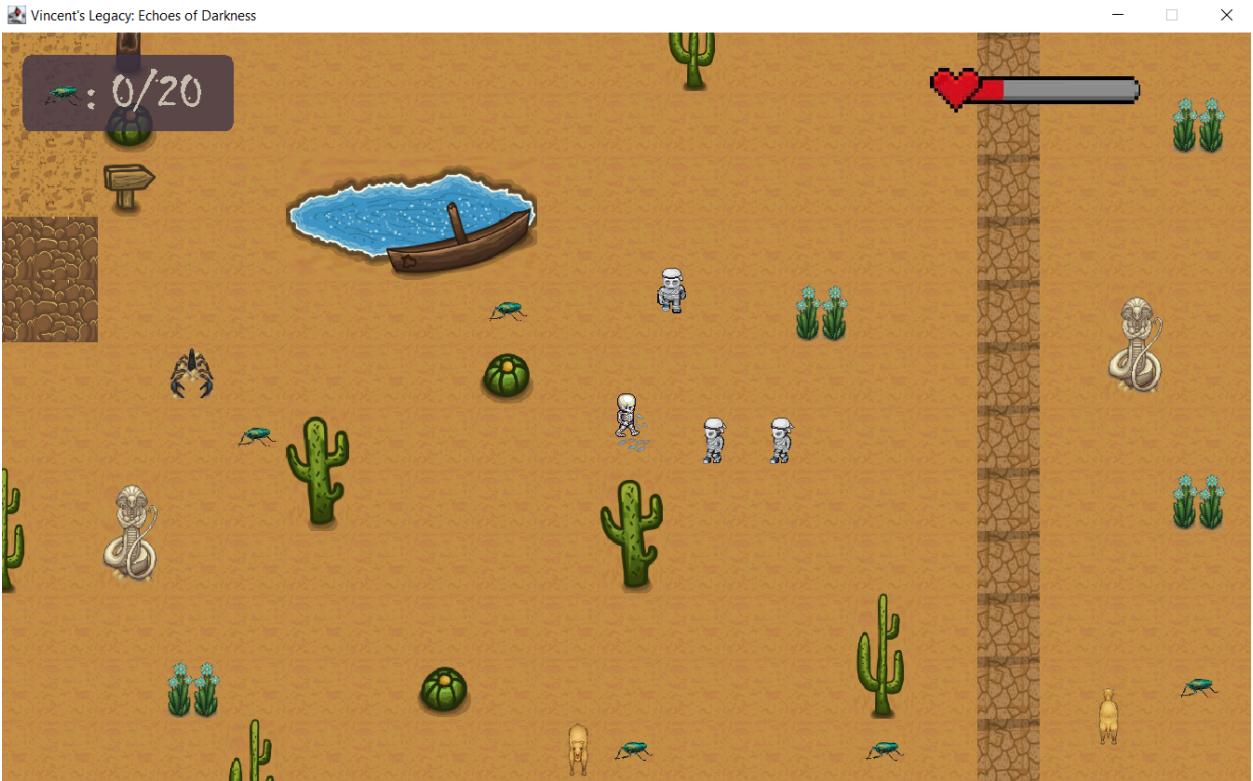
Jocul reintră într-un cutscene, care prezintă transformarea lui Vincent din nou în om, și apariția Desdemonei. Cei doi se dueleză, iar Vincent aruncă o vrajă care o nimicește pe vrăjitoare. Tânărul Întunecat este învins, iar Vincent și Helena pleacă împreună fericiți.

De colectat la nivelul 3:



Screenshot-uri din nivelul 3:





Descrierea claselor utilize:

## 1. Main

Clasa Main servește ca punctul de intrare al aplicației de joc. Aceasta initializează fereastra principală a jocului utilizând JFrame și configurându-l pentru a nu putea fi redimensionat și pentru a avea un titlu specific ("Vincent's Legacy: Echoes of Darkness"). Metoda principală (main) folosește un singleton pentru a obține instanța ferestrei, adaugă un GamePanel în fereastră, ajustează dimensiunea ferestrei la conținutul său și poziționează fereastra în centrul ecranului. În final, metoda pornește jocul prin apelarea metodelor setupGame și startGameThread din GamePanel, ceea ce initializează elementele jocului și începe bucla principală a jocului.

## 2. Singleton

Clasa Singleton implementează un pattern Singleton pentru a asigura că există o singură instanță a ferestrei principale (JFrame) în aplicație. Constructorul clasei este privat pentru a preveni instantierea directă. Metoda getInstance returnează instanța unică a ferestrei, creând-o dacă nu a fost deja creată. Aceasta metodă este sincronizată pentru a asigura siguranța firelor în contextul accesului concurrent. Astfel, Singleton garantează că aplicația va folosi o singură fereastră principală pe durata execuției.

## 3. GamePanel

Clasa GamePanel extinde JPanel și implementează interfața Runnable, fiind responsabilă pentru gestionarea logicii și a interfeței grafice a jocului. Ea definește dimensiunile plăcilor și ecranului, precum și diferite stări ale jocului, cum ar fi starea de titlu, joc, dialog, poveste, game over și pauză. Metodele principale includ setupGame, care initializează obiectele, NPC-urile și monștrii, și startGameThread, care pornește firul principal al jocului. Metoda run implementează bucla principală a jocului, actualizând logica jocului și reîmprospătând ecranul la fiecare cadru pentru a atinge 60 FPS. paintComponent este folosită pentru a desena grafica jocului pe ecran, inclusiv plăcile, jucătorul, NPC-urile, obiectele și UI-ul. Clasa utilizează diverse componente pentru a controla diferite aspecte ale jocului, cum ar fi TileManager pentru gestionarea plăcilor, KeyHandler pentru input-ul de la tastatură, CollisionChecker pentru verificarea coliziunilor și AssetSetter pentru plasarea obiectelor în joc. În plus, GamePanel integrează și funcționalități pentru gestionarea sunetului, prin metodele playMusic, stopMusic și playSE. De asemenea, clasa interacționează cu o bază de date SQLite pentru a salva și încărca stările jocului, utilizând metode precum insertData, getData, deleteAllData și resetAutoIncrement.

## 4. TileManager

Clasa TileManager gestionează gestionarea și desenarea tile-urilor din joc în cadrul hărții. Această clasă primește o referință către GamePanel pentru a accesa informații despre dimensiunile și configurația jocului. La nivelul membrilor, TileManager

deține un tablou de obiecte Tile pentru a reprezenta diversele tipuri de tile-uri din joc și o matrice mapTileNum care conține numere pentru a indica tipurile de tile-uri din harta jocului. Metoda getTileImage() este utilizată pentru a încărca imaginile asociate cu fiecare tip de tile din fișierele sursă, folosind ImageIO pentru a citi imaginile din resursele jocului. Metoda loadMap(String filePath) este folosită pentru a încărca harta jocului dintr-un fișier text specificat, unde fiecare număr reprezintă un tip de tile. Metoda draw(Graphics2D g2) este responsabilă pentru desenarea hărții pe ecran. Aceasta parcurge matricea de numere asociate tile-urilor și desenează doar tile-urile vizibile pe ecran, având în vedere poziția și dimensiunea jucătorului pentru a determina ce porțiune a hărții trebuie desenată. În concluzie, TileManager este responsabil pentru gestionarea și desenarea eficientă a hărții jocului, asigurând că doar tile-urile vizibile sunt desenate pe ecran în funcție de poziția și acțiunile jucătorului.

## 5. Tile

Clasa Tile este folosită în cadrul TileManager pentru a reprezenta diferitele tipuri de tile-uri din joc. Fiecare obiect Tile este asociat cu o imagine specifică și poate avea sau nu o proprietate de coliziune, în funcție de configurarea hărții și a mecanicilor de joc. În cadrul TileManager, fiecare element din tabloul de tile-uri este o instanță a clasei Tile, reprezentând o unitate fundamentală a hărții jocului.

## 6. KeyHandler

Clasa KeyHandler implementează interfața KeyListener pentru a gestiona evenimentele legate de tastatură în joc. Ea monitorizează diferite taste apăsate și le interpretează în funcție de starea curentă a jocului. Printre variabilele de stare se numără upPressed, downPressed, leftPressed, rightPressed, EPressed și shotKeyPressed, care indică dacă anumite taste sunt apăsate sau nu. În metoda keyPressed, se verifică starea jocului și se reacționează corespunzător la tastele apăsate. De exemplu, în starea de titlu, tastele W și S sunt folosite pentru navigarea în meniu, iar Enter pentru selectarea unei opțiuni. În starea de joc, tastele W, A, S, D sunt utilizate pentru mișcare, iar tasta SPACE pentru a trage. În starea de pauză, tasta P este folosită pentru a comuta între starea de pauză și starea de joc. Metoda keyReleased se ocupă de marcarea tastelor ca fiind eliberate atunci când acestea sunt ridicate. Comentariile din cod indică și alte funcționalități care pot fi activate sau dezactivate în funcție de nevoile jocului.

## 7. Entity

Clasa Entity este o componentă esențială în joc, având rolul de a gestiona toate entitățile din lumea jocului, inclusiv jucătorul și NPC-urile. În cadrul acestei clase sunt stocate informații cruciale precum poziția entității pe harta jocului (worldX și worldY),

viteza de deplasare, direcția actuală de mișcare și starea de viață (life). În plus, clasa Entity se ocupă de detectarea și gestionarea coliziunilor dintre entități și elementele mediului înconjurător. Pentru aceasta, folosește arii de coliziune și de atac pentru a verifica intersecțiile cu alte entități, obiecte și obstacole din joc. Prin intermediul metodelor precum checkCollision() și update(), clasa actualizează starea entității în fiecare cadru al jocului, luând decizii legate de mișcare și interacțiune. De asemenea, clasa Entity se ocupă de desenarea entităților pe ecran în poziția corespunzătoare, astfel încât acestea să fie vizibile pentru jucător. Gestionează animațiile și direcțiile de desenare pentru a crea o experiență de joc fluidă și realistă. Prin intermediul metodelor specifice, cum ar fi speak() pentru dialogurile NPC-urilor sau searchPath() pentru căutarea unei căi de la entitatea curentă la o destinație dorită, clasa Entity îndeplinește diverse funcționalități cruciale pentru experiența de joc și funcționarea corectă a întregului sistem de joc.

## 8. Player

Clasa Player este o subclasă a clasei Entity și reprezintă jucătorul din joc. Această clasă conține logica specifică pentru mișcarea și interacțiunea jucătorului în lumea jocului. Proprietatea KeyHandler keyH este utilizată pentru a gestiona intrările de la tastatură asociate jucătorului. Constructorul clasei Player primește o referință către GamePanel și KeyHandler pentru a gestiona acțiunile jucătorului în cadrul jocului. Metoda setDefaultValues() inițializează valorile implicate ale jucătorului, cum ar fi poziția inițială, viteza și direcția. Metoda getPlayerImage() încarcă imaginile asociate cu mișcarea jucătorului în diferite direcții (sus, jos, stânga, dreapta) din resursele jocului. Metoda update() este responsabilă pentru actualizarea stării jucătorului la fiecare cadru de joc. Ea gestionează intrările de la tastatură pentru mișcare și interacțiunea cu obiectele și NPC-urile din lumea jocului. Metoda draw() desenează jucătorul pe ecran în funcție de direcția și animația curentă. Metodele pickUpObject(int i) și interactNPC(int i) permit jucătorului să interacționeze cu obiectele și NPC-urile din joc, respectiv. Această clasă reprezintă o parte esențială a implementării logicii jocului, permitând jucătorului să se deplaceze în lumea jocului și să interacționeze cu elementele sale.

## 9. NPCHelena

Clasa NPCHelena extinde clasa Entity și reprezintă un NPC (non-player character) specific jocului. Acest NPC, numit Helena, are un rol important în povestea jocului și interacționează cu jucătorul în diferitele locații ale jocului. În constructorul său, sunt inițializate variabilele specifice NPC-ului și sunt încărcate imaginile necesare pentru afișarea acestuia în joc. Metoda setDialogue() stabilește dialogul dintre Helena și jucător în diferitele etape ale jocului. Dialogul variază în funcție de progresul jucătorului în obținerea anumitor obiecte sau îndeplinirea anumitor condiții. Metoda speak() este

apelată pentru a afișa dialogul pe ecran. Dialogul se adaptează la locația curentă a jucătorului și la progresul acestuia în joc. De exemplu, în timp ce jucătorul se află pe harta 0 (prima locație din joc), Helena îi va cere jucătorului să adune 20 de oase. Pe măsură ce jucătorul progresează și adună oase, dialogul se schimbă pentru a reflecta aceasta și pentru a ghida jucătorul către următoarea etapă a jocului. În mod similar, pe măsură ce jucătorul avansează în joc și ajunge în alte locații, dialogul și cerințele lui Helena se schimbă în consecință, influențând povestea și direcția jocului.

## 10. NPCArcheologist

Clasa NPCArcheologist reprezintă un alt NPC din joc, un arheolog care interacționează cu jucătorul în timpul aventurii acestuia. Asemenea celorlalte clase de NPC-uri, aceasta extinde clasa Entity și definește comportamentul și dialogul NPC-ului în diferite situații din joc. Constructorul clasei initializează variabilele specifice NPC-ului și încarcă imaginea necesară pentru afișarea acestuia în joc. Metoda setDialogue() stabilește dialogul dintre arheolog și jucător, care constă într-o serie de schimburi de replici. În acest dialog, arheologul cere ajutorul jucătorului pentru a găsi o cheie pierdută în junglă. Metoda speak() este responsabilă pentru afișarea dialogului pe ecran. În timpul interacțiunii cu jucătorul, arheologul își exprimă cererea de ajutor și, odată ce jucătorul găsește cheia, îi oferă o recompensă, în acest caz, o felie de pizza.

## 11. NPCArabMan

Clasa NPCArabMan definește comportamentul și dialogul unui personaj non-jucător (NPC) din joc, reprezentând un negustor beduin. Acest NPC interacționează cu jucătorul în timpul aventurii și îi oferă sarcini și recompense în funcție de progresul acestuia în joc. Constructorul clasei initializează variabilele specifice NPC-ului și încarcă imaginea necesară pentru afișarea acestuia în joc. Metoda setDialogue() stabilește dialogul dintre negustor și jucător, care constă într-o serie de schimburi de replici. În acest dialog, negustorul își exprimă nevoia disperată de a-și găsi camilele pierdute într-o furtună de nisip și îi cere jucătorului să le găsească și să le aducă înapoi. Metoda speak() este responsabilă pentru afișarea dialogului pe ecran și gestionarea logicii asociate. În timpul interacțiunii cu jucătorul, negustorul își exprimă nevoia și, odată ce jucătorul găsește camilele, îi oferă o recompensă, în acest caz, o ceașcă de cafea.

## 12. Projectile

Clasa Projectile reprezintă proiectile lansate în joc și gestionează comportamentul acestora în timpul jocului. Constructorul clasei initializează proiectilul și apelează constructorul clasei de bază. Metoda set() este responsabilă pentru setarea poziției, direcției și stării de viață a proiectilului. Această metodă primește coordonatele

de pe hartă, direcția în care trebuie să se deplaseze proiectilul, informații despre starea acestuia și utilizatorul care a lansat proiectilul. Metoda update() gestionează actualizarea stării proiectilului în fiecare cadru de joc. Pentru un proiectil lansat de jucător, este verificată coliziunea cu monștrii. Dacă există o coliziune, proiectilul va cauza daune monștrilor și va fi marcat ca inactiv. Pentru un proiectil lansat de un monștrul, este verificată coliziunea cu jucătorul. Dacă jucătorul nu este invincibil și există o coliziune, proiectilul va cauza daune jucătorului și va fi marcat ca inactiv. Proiectilul este apoi mutat în conformitate cu direcția sa și viața sa este redusă.

### 13. AssetSetter

În codul dat, clasa AssetSetter pare să fie responsabilă pentru inițializarea și plasarea obiectelor, monștrilor și NPC-urilor în joc. De asemenea, conține metode pentru a crea și plasa diferite tipuri de obiecte, monștri și NPC-uri pe hărți diferite. Metoda setObject este utilizată pentru a plasa obiecte pe hărțile jocului. Obiectele, cum ar fi oasele, cocoșii și scarabeii, sunt plasate la coordonate specifice pe hartă. Aceste coordonate sunt predefinite și par să fie configurate manual pentru fiecare obiect în parte, în funcție de harta pe care se află jucătorul. Metoda setNPC se ocupă de plasarea NPC-urilor pe hărți. NPC-urile, cum ar fi Helena, arheologul și arabul, sunt plasate la anumite coordonate pe hărțile corespunzătoare. Asemenea obiectelor, coordonatele NPC-urilor sunt definite manual pentru fiecare hartă în parte. Există și alte metode, cum ar fi spawnPizzaObject, spawnCoffeeObject, spawnPortalObject și spawnPotionObject, care sunt responsabile pentru plasarea unor obiecte specifice pe hărți. De exemplu, spawnPizzaObject plasează un obiect pizza pe harta cu numărul 1, la anumite coordonate. În cele din urmă, metoda setMonster este folosită pentru a plasa monștri pe hărțile jocului. Monștrii, cum ar fi dovlecii, goblinii și mumia, sunt plasați la coordonate prestabilite pe hărți. Aceste coordonate sunt stabilite manual pentru fiecare monștru în parte, în funcție de harta pe care se află jucătorul.

### 14. CollisionChecker

Clasa CollisionChecker este responsabilă pentru gestionarea verificărilor de coliziuni în cadrul jocului din GamePanel. Aceasta conține metode care verifică coliziunile între entități (Entity) și diverse elemente din joc, cum ar fi plăcile de teren (Tile), obiecte (OBJ), și chiar jucătorul (Player). Metoda checkTile(Entity entity) verifică coliziunile între o entitate și plăcile de teren din joc. Aceasta determină dacă o entitate va intra în coliziune cu plăcile de teren din jurul său în funcție de direcția în care se deplasează entitatea (up, down, left, right). Se calculează plăcile de teren pe care entitatea le-ar ocupa în urma mișcării și se verifică dacă acestea au proprietatea de coliziune. Metoda checkObject(Entity entity, boolean player) verifică coliziunile între o

entitate și obiectele din joc (OBJ). Această metodă determină dacă entitatea se ciocnește cu un obiect, iar în funcție de tipul de coliziune, poate activa anumite acțiuni sau interacțiuni. De asemenea, este utilizată pentru a determina dacă jucătorul (Player) interacționează cu un anumit obiect. Metoda `checkEntity(Entity entity, Entity[] target)` este folosită pentru a verifica coliziunile între o entitate și alte entități (Entity) din joc. Această metodă este utilă pentru a determina interacțiunile între entități, cum ar fi coliziunile între NPC-uri sau alte entități specifice. Metoda `checkPlayer(Entity entity)` este specializată pentru a verifica coliziunile între o entitate și jucătorul (Player). Aceasta determină dacă o entitate se ciocnește cu jucătorul, ceea ce poate activa anumite acțiuni sau reacții specifice. În ansamblu, clasa CollisionChecker are rolul de a gestiona și verifica interacțiunile de coliziune dintre diverse elemente din lumea jocului, contribuind la realismul și funcționalitatea acestuia. Această abordare este esențială pentru a asigura un mediu de joc interactiv și captivant pentru jucător.

## 15. UI

Clasa UI din joc este responsabilă pentru afișarea interfeței utilizatorului (UI) în cadrul GamePanel. Această clasă se ocupă de desenarea elementelor UI, cum ar fi bara de dialog, imagini, texte și ferestre de dialog. În constructorul clasei UI, sunt inițializate fonturile și imaginile necesare pentru desenarea UI-ului. Se utilizează fonturi precum "Chiller" și "Lucida Sans Typewriter" pentru a stiliza textul afișat pe ecran. De asemenea, imaginea pentru obiectul "Bone" este utilizată pentru a afișa icon-ul și numărul de oase colectate în bara de stocare. Metoda `draw(Graphics2D g2)` este folosită pentru a desena componentele UI pe ecran. Aceasta include desenarea barei de stocare a oaselor și afișarea numărului de oase colectate. În funcție de starea jocului (`gameState`), se poate afișa și fereastra de dialog utilizând metoda `drawDialogueScreen()`. Metoda `drawDialogueScreen()` este responsabilă pentru desenarea ferestrei de dialog atunci când jocul se află în starea de dialog (`dialogueState`). Aceasta afișează textul dialogului pe ecran într-o fereastră stilizată, utilizând fontul "Lucida Sans Typewriter". Textul dialogului este separat pe linii și afișat vertical în fereastra de dialog. Metoda `drawSubWindow(int x, int y, int width, int height)` este utilizată pentru a desena o fereastră de sub-componente UI, care poate fi utilizată pentru a încadra și evidenția anumite elemente UI, cum ar fi ferestrele de dialog. În ansamblu, clasa UI facilitează afișarea interfeței utilizatorului în cadrul jocului, contribuind la experiența vizuală și la interacțiunea jucătorului cu elementele din lumea jocului.

## 16. EventHandler

Clasa EventHandler este responsabilă pentru gestionarea evenimentelor din joc, cum ar fi interacțiunea jucătorului cu obiectele sau trecerea între nivele. Prin intermediul

metodelor sale, clasa verifică coliziunile dintre jucător și zonele evenimentelor, precum și distanțele parcuse de jucător pentru a activa acțiuni specifice, cum ar fi teleportarea la un alt nivel.

## 17. EventRect

Clasa EventRect definește obiectele rectangulare folosite pentru gestionarea evenimentelor în joc. Aceste obiecte sunt subclase ale clasei Rectangle din biblioteca standard Java AWT și conțin informații despre poziția lor implicită, precum și despre starea evenimentului asociat. Prin intermediul acestor obiecte, jocul poate detecta și reacționa la interacțiunile jucătorului cu diferite zone din mediul său.

## 18. Sound

Clasa Sound este responsabilă de gestionarea efectelor de sunet în cadrul jocului. Aceasta utilizează funcționalitățile din Java Sound API pentru a încărca și reda fișiere audio. În constructorul clasei Sound, sunt inițializate URL-urile pentru diferite fișiere audio, cum ar fi sunetele ambientale și sunetul de colectare al obiectelor (ex.: bones.wav). Metoda setFile(int i) este utilizată pentru a încărca un fișier audio specificat prin indexul i în obiectul Clip. Această metodă obține un flux de intrare audio din URL-ul specificat și deschide obiectul Clip pentru redare.

## 19. SuperObject

Clasa SuperObject reprezintă o componentă fundamentală în structura jocului, oferind funcționalități esențiale pentru gestionarea și desenarea obiectelor pe ecran. Această clasă servește ca o bază comună pentru toate obiectele din joc, inclusiv obiecte precum oasele (OBJ\_Bone) sau portalul (OBJ\_Portal1). Atributele acestei clase includ o imagine asociată obiectului (BufferedImage image), numele obiectului (String name), un indicator de coliziune pentru gestionarea interacțiunilor solide cu alte entități (boolean collision), și coordonatele obiectului în lumea jocului (int worldX și int worldY). De asemenea, clasa definește o zonă solidă (Rectangle solidArea) care descrie aria pe care o ocupă obiectul în spațiul jocului, esențială pentru detecția coliziunilor. Metoda draw(Graphics2D g2, GamePanel gp) este responsabilă pentru desenarea obiectului pe ecran, luând în considerare poziția acestuia în lumea jocului și configurația camerei jucătorului (GamePanel). Această metodă asigură că obiectele sunt desenate doar atunci când sunt vizibile în cadrul camerei jucătorului, optimizând astfel performanța și eficiența desenării în joc.

## 20. OBJ\_Bone

## 21. OBJ\_Coffee

## 22. OBJ\_Key

- 23. OBJ\_MrCoconut
- 24. OBJ\_Pizza
- 25. OBJ\_Potion
- 26. OBJ\_Scarab

Clasele menționate extind clasa SuperObject și reprezintă un tip specific de obiect în joc. Aceste clase sunt responsabile pentru inițializarea și configurarea caracteristicilor specifice ale unui obiect de un tip specific în cadrul jocului. În constructorul clasei, se stabilește numele obiectului (name) pentru a identifica clar tipul de obiect. De asemenea, în constructor, imaginea asociată obiectului este încărcată dintr-un fișier de imagine folosind ImageIO.read().

- 27. OBJ\_Spell
- 28. OBJ\_Fireball
- 29. OBJ\_Glass

Clasele OBJ\_Glass, OBJ\_Fireball și OBJ\_Spell sunt toate subclase ale clasei Projectile și sunt responsabile pentru gestionarea proiectilelor specifice din joc. Fiecare clasă își încarcă imaginile asociate proiectilelor din fișierele de resurse PNG folosind clasa ImageIO și metoda getResourceAsStream() pentru a citi aceste fișiere. Aceste clase definesc și setează proprietăți precum viteza proiectilelor, durata lor de viață maximă, puterea de atac și starea de viață curentă. Proiectile sunt inițializate ca fiind "neactive" (alive = false) și vor fi activate și desenate pe ecran în momentul în care sunt lansate în joc.

#### 30. OBJ\_Health

Clasa OBJ\_Health este o subclasă a clasei SuperObject și are rolul de a gestiona obiectele de tip sănătate (health) din joc. Ea încarcă imaginile asociate acestor obiecte din fișierele de resurse PNG folosind clasa ImageIO și metoda getResourceAsStream() pentru a citi aceste fișiere. Obiectele de sănătate sunt reprezentate vizual prin mai multe imagini, fiecare imagine reprezentând o stare diferită a sănătății. Clasa OBJ\_Health încarcă șase imagini diferite pentru a reprezenta diferențele stări ale obiectului de sănătate, de la completă (imaginile 1 și 2) la parțial deteriorată (imaginile 5 și 6).

#### 31. MON\_Pumpkin

Clasa MON\_Pumpkin este responsabilă pentru gestionarea caracteristicilor specifice ale unui tip de monstru în joc numit "Pumpkin". Atributul actionLockCounter funcționează ca un contor pentru timpul dintre acțiunile următoare ale monstrului. Constructorul inițializează atributele monstrului, precum viteză și viață maximă, și

încarcă imaginile corespunzătoare pentru diferitele direcții de mișcare. Metoda setAction() implementează un comportament de mișcare haotică, alegând aleatoriu o direcție de mișcare la fiecare 120 de iterații.

### 32. MON\_Goblin

Clasa MON\_Goblin reprezintă un tip de monstru din joc denumit "Goblin". Constructorul clasei inițializează caracteristicile specifice ale goblinului, inclusiv viteză, viață maximă și atac. De asemenea, acesta încarcă imaginile necesare pentru animația goblinului în diferitele direcții de mișcare. Metoda setAction() controlează comportamentul goblinului, combinând mișcarea haotică cu atacul. La fiecare 120 de iterații, goblinul alege aleatoriu o direcție de mișcare (sus, jos, stânga sau dreapta). De asemenea, există o șansă mică ca goblinul să lanseze un proiectil de tip OBJ\_Fireball în direcția aleasă, dacă anumite condiții sunt îndeplinite.

### 33. MON\_Mummy

Clasa MON\_Mummy reprezintă un monstru de tip "Mumie" din joc. Constructorul clasei inițializează caracteristicile specifice ale mumiei, cum ar fi viteză, viață maximă și atac. De asemenea, creează un proiectil de tip OBJ\_Glass pentru a fi folosit în atacurile sale. Metoda setAction() este responsabilă pentru controlul comportamentului mumiei. Dacă mumia se află pe o cale către jucător (onPath este setat la true), ea îl va urmări și îl va ataca. Altfel, mumia va avea un comportament haotic, mișcându-se aleatoriu. De asemenea, există o șansă mică ca mumia să lanseze un proiectil de sticlă (OBJ\_Glass) în direcția jucătorului, dacă acesta este în apropiere și dacă proiectilul nu este deja activ.

### 34. MON\_Camel

Clasa MON\_Camel reprezintă o entitate de tip "Cămilă" din joc. Aceasta va trebui colectată de către player, și dusă NPC-ului din nivelul 3. Constructorul clasei inițializează caracteristicile specifice ale cămilei, cum ar fi viteză și viață maximă. Metoda getImage() este responsabilă pentru încărcarea imaginilor cămilei în diverse direcții de mișcare (sus, jos, stânga, dreapta). Metoda setAction() controlează comportamentul cămilei. În acest caz, cămila se mișcă haotic, alegând aleatoriu una dintre cele patru direcții (sus, jos, stânga, dreapta) la intervale regulate de timp. Această mișcare haotică este simbolizată de alegerile aleatorii de direcție la fiecare 120 de unități de timp.

### 35. Node

Clasa Node este o componentă esențială în implementarea algoritmilor de căutare a căilor în jocuri și aplicații AI. Fiecare obiect Node reprezintă o celulă sau o poziție într-o hartă sau grilă. Atributele clasei includ referința către nodul părinte, coordonatele coloanei și rândului, costurile utilizate în algoritmul A\*, precum și indicatori care arată dacă nodul este solid, deschis sau deja verificat în timpul procesului de căutare. Constructorul clasei Node initializează obiectul cu coordonatele specificate. Această clasă este fundamentală în gestionarea structurii de date necesare pentru algoritmii de căutare a căilor.

### 36. PathFinder

Clasa PathFinder este esențială în implementarea algoritmilor de căutare a căilor în jocuri și aplicații AI. Aceasta utilizează un grid de obiecte Node pentru a găsi cel mai scurt drum între două puncte pe hartă. Constructorul clasei primește o referință la GamePanel, iar metoda instantiateNodes() creează și initializează toate nodurile în grid. Metoda resetNodes() resetează starea nodurilor și alte setări atunci când este necesar. Metoda setNodes() setează nodurile de start și de final și calculează costurile acestora. Algoritmul de căutare este implementat în metoda search(), care explorează nodurile din openList și găsește drumul optim către nodul țintă. trackThePath() este utilizată pentru a urmări drumul de la nodul final înapoi la nodul de start și a-l salva în pathList. Această clasă este crucială pentru navigarea entităților pe hartă și luarea deciziilor AI.

Sprite-uri utilizate:

1. Vincent în ipostaza de schelet - main character



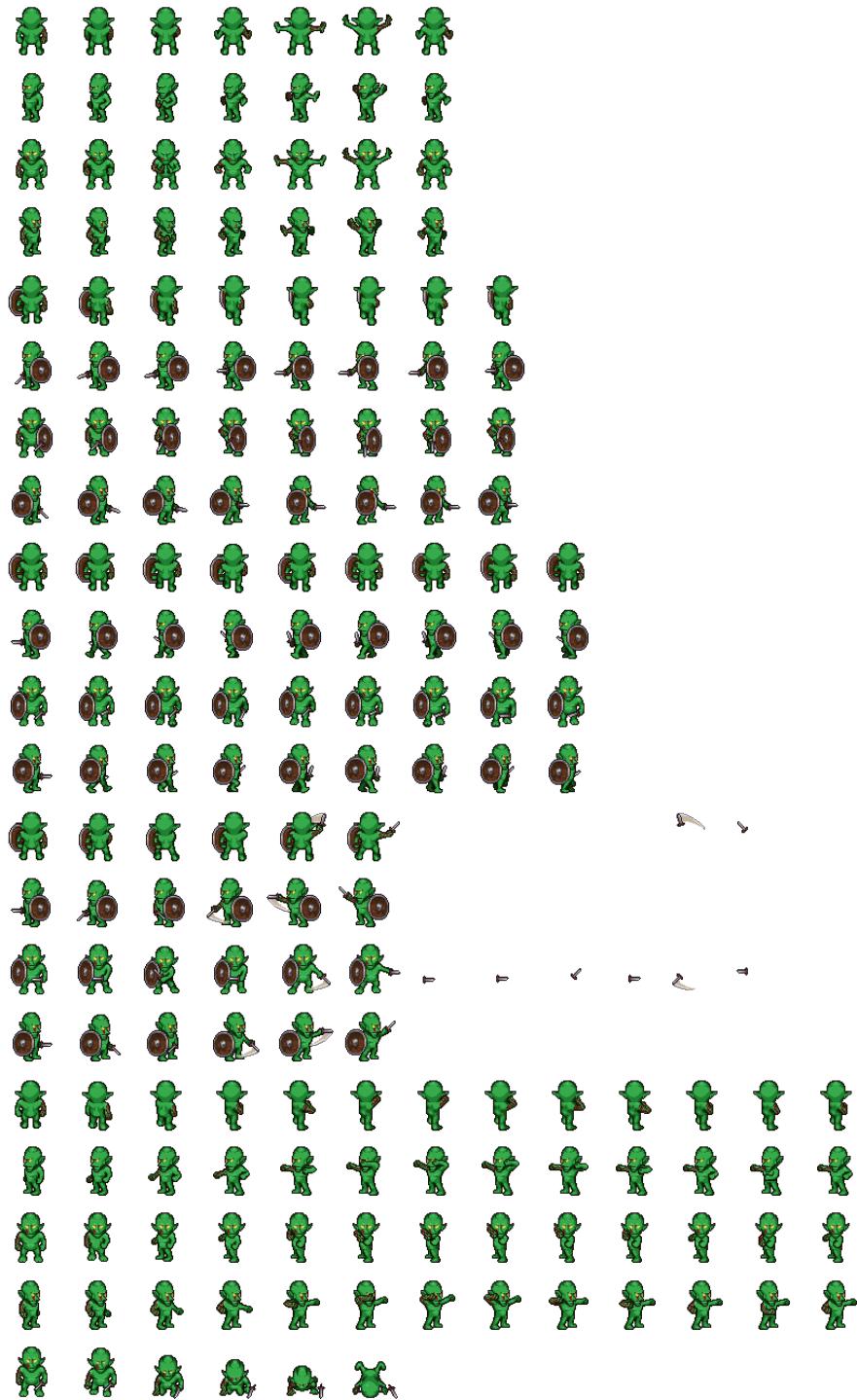
2. Helena - fiica vitregă a Desdemonei, prietena lui Vincent



3. Pumpkins - inamicul de la lvl 1



4. Goblini - inamicul de la lvl 2



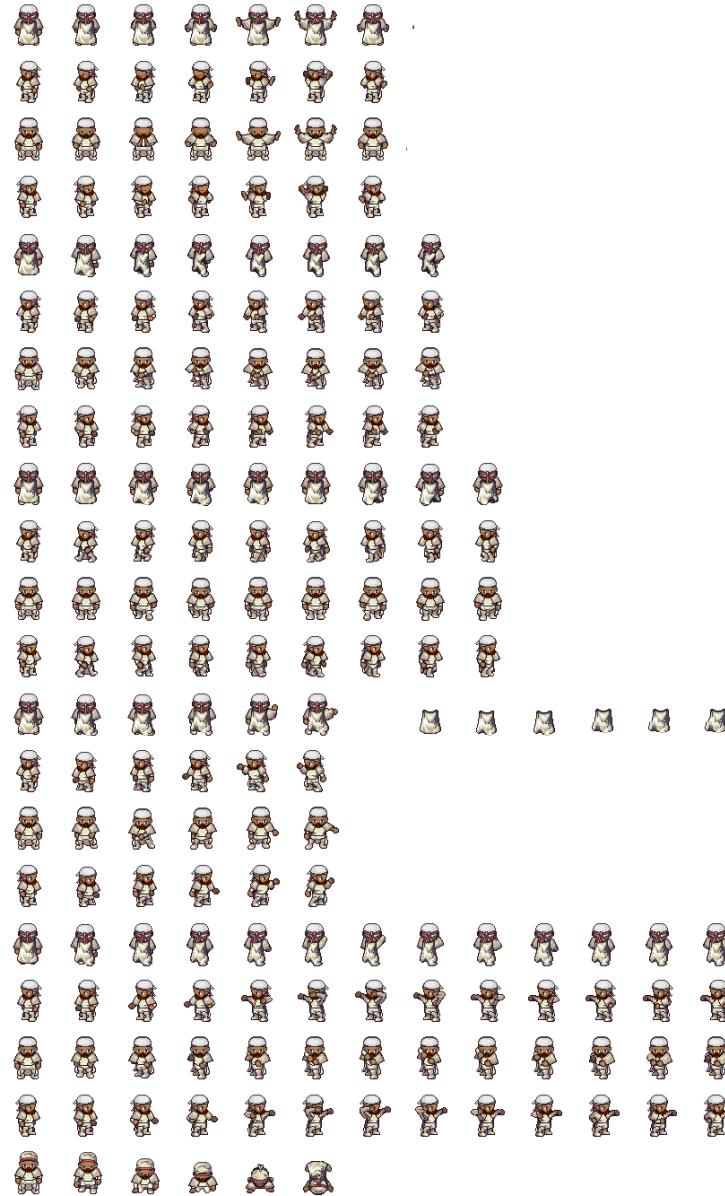
5. Arheolog - NPC de la lvl 2



6. Mumii - inamici de la lvl 3



7. Arabul - NPC de la lvl 3



## Soundtrack:

Story de la începutul jocului:

Ex: <https://www.youtube.com/watch?v=sIxPFAJN9UM>

Nivelul 1 - [https://www.youtube.com/watch?v=H14z\\_OGKPwM](https://www.youtube.com/watch?v=H14z_OGKPwM)

Nivelul 2 - sunete exotice, tribale, specifice junglei din nivelul 2

Ex: [https://www.youtube.com/watch?v=9hyk\\_pkq9\\_A](https://www.youtube.com/watch?v=9hyk_pkq9_A)

Nivelul 3 - muzică specifică Egiptului Antic

Ex: <https://www.youtube.com/watch?v=GI6dOS5ncFc>

Boss Fight

Ex: <https://www.youtube.com/watch?v=tBI63CPbW9c>

## Bibliografie:

- [https://sanderfrenken.github.io/Universal-LPC-Spritesheet-Character-Generator/#?body=Body\\_color\\_light&head=Human\\_male\\_light](https://sanderfrenken.github.io/Universal-LPC-Spritesheet-Character-Generator/#?body=Body_color_light&head=Human_male_light) - sprite generator
- <https://craftpix.net/> - sprites
- <https://elthen.itch.io/> - sprites
- <https://www.deviantart.com/> - sprites
- <https://www.youtube.com/watch?v=slyi7Nt-9LE&t=1607s> - joc vechi, de la care a pornit ideea
- [https://www.youtube.com/playlist?list=PL\\_QPQmz5C6WUF-pOQDsbsKbaBZqXj4qSq](https://www.youtube.com/playlist?list=PL_QPQmz5C6WUF-pOQDsbsKbaBZqXj4qSq) - tutorial Java 2D Game
- [https://www.demirramon.com/generators/undertale\\_text\\_box\\_generator](https://www.demirramon.com/generators/undertale_text_box_generator) - text box generator
- [www.canva.com](http://www.canva.com) - editor cu care am realizat interfață
- <https://wepik.com/ai> - AI Generator cu care am realizat imaginile pentru poveste