**Proiect(The Forgotten War)**

**Autor(Nejneriu Mihai)**

**Povestea Jocului:**

Intr-un alt univers departat, foarte asemanator cu al nostru, unde totul se petrecea normal, personajul nostru Adi era un Student normal la facultate. Intr-o zi a primit o scrisoare dubioasa. Fiind ingrijorat, si-a anuntat prietenii lui. Coincidenta facea ca toti au primit aceeasi scrisoare. “Veniti la prapastia din nord, soarta lumii depinde de voi!” S-au adunat toti sa discute despre acest lucru si au ajuns la concluzia ca trebuie sa mearga acolo si sa verifice. Odata ce au ajuns acolo, prapastia s-a luminat. Cand s-au aplecat toti sa se uite, au fost atrasi inauntru. Toti au fost teleportati intr-o pestera a dragonilor. Fiecare a fost teleportat intr-o locatie diferita. Ingrijorat, Adi exploreaza pestera. Dupa cateva minute, intampina un personaj deosebit. Iselda. Iselda este o vrajitoare extrem de cunoscuta. Acum 2000 de ani, ea a fost cea care a salvat omenirea de la distrugere si a reusit sa il faca pe Regele Dragon sa se retraga. Aceasta il informa pe Adi ca Regele Dragon a detectat prezenta lui si ca va face tot ce poate in speranta de a il opri. Iselda ii spune ca a fost transportata aici dupa marele razboi, impreuna cu echipa ei, si ca si-a pierdut puterite cand a ajuns aici. Tot echipamentul ei a fost impartit in toata pestera. Ca sa scape, trebuie sa sacrifice un numar de suflete, si este dispusa sa ofere anumite beneficia in schimbul acestora. Ii ofera lui Adi o amuleta speciala si o sabie. Amuleta are puterea de a extrage sufletele din inamici si il teleporteaza pe Adi inapoi la Iselda in caz de pericol. Aceasta ii mai spune ca toti prietenii lui, impreuna cu grupul ei, au fost capturati si sunt tinuti la nivele diferite in pestera. Pentru a scapa de aici, Adi este nevoit sa il bata pe Regele Dragon. Pe parcurs, el va fi oprit de prietenii Regelui Dragon. Dupa ce i-a infrant pe toti, inclusive Regele Dragon, adevaratul antagonist isi va face aparitia, Iselda. Ea este cea care i-a chemat pe baieti dar nu cu scopul de a salva lumea, ci cu scopul de a o ajuta pe ea sa scape. Povestea din spate este acum dezvaluita, faptul ca Iselda a obtinut o putere foarte mare care a scuturat tot universul. Regele Dragon cand a simtit asta, a pornit la razboi, impreuna cu prietenii ei, ca sa o opreasca pe aceasta. Cand grupul de prieteni a intrat in pestera, Regele Dragon a incercat sa ii opreasca deoarece credea ca noi de fapt suntem de partea ei (ceea ce este adevarat deoarece noi nu stiam adevaratul motiv). Dupa ce Regele Dragon este infrant, Iselda ii foloseste sufletul, impreuna cu toate sufletele obtinute ca sa sparga bariera impotriva ei la intrarea in pestera. Scopul jocului s-a transformat in a o opri pe Iselda. Dupa ce aceasta este infranta, sufletele la toti “boss-ii” se intorc inapoi, si ei te vor ajuta sa o infrangi pe aceasta. In final, Iselda va fi inlocuita cu forma de om a Regelui Dragon, in loc de suflete, caracterul principal va obtine banuti si in loc sa omoare inamicii in calea lui, acesta doar ii lasa inconstienti. Ultimul boss devine inapoi Regele dragon si inainte de aceasta batalie, va fi un cinematic care va arata cum Regele Dragon se transforma in forma lui originala, zambind, si totul se termina cu bine.

**Prezentare joc:**

Jocul va avea un singur tip de joc, story line, unde scopul este de a scapa din pestera cu toti prietenii tai.

La fiecare final de nivel, se va afla o batalie mai dificila decat cele anterioare.

Fiecare pestera va avea cel putin 2 drumuri care duc mai departe si ofera beneficii diferite.

**Reguli joc:**

Personajul principal va ataca cu tasta “Space” si va fugi cu tasta “Shift”. Pentru a deschide meniul de upgrade-uri de la Iselda se va apasa tasta “E”. Pentru movement se vor apasa tastele “W” “A” “S” “D”.

Jocul este format din 4 nivele. Fiecare nivel are:

-Safe map-ul care este prima harta intalnita in fiecare nivel. Aceasta nu contine inamici si aici apare Iselda. Dupa aceasta urmeaza 4 harti generate complet aleatoriu, inclusiv inamicii din aceasta. Dupa toate acestea urmeaza Boss map-ul care este o harta predefinita care contine 4 pereti si mai multi inamici decat anterior. Dupa boss map, urmeaza nivelul urmator.

Dupa fiecare nivel, vor aparea tipuri noi de inamici. In primul nivel va aparea numai “Drake”. La al doilea va aparea “Fast Drake”, la al treilea apare “Slow Drake” si la ultimul apare “Ghost Drake”.

Dupa infrangerea tuturor inamicilor, recompensa va aparea si dupa revendicarea acesteia, se vor deschide usile. Fiecare iesire va avea desenata pe aceasta cate un power up care reprezinta recompensa la nivelul urmator.

Iselda este un NPC care iti va oferi posibilitatea de a cumpara upgrade-uri.

**Personajele Jocului:**

-Adi:

Personajul principal, nimic special

 -Normal Drake:

Primul inamic intalnit in joc.

Are viteza considerata normala, dimensiune normala si viata normala.

-Fast Drake:

Prima aparitie se afla in nivelul 2.

Este mult mai rapid decat cel normal, are mai putina viata, este mai micut, dar are atacul mai puternic

-Slow Drake:

Prima aparitie se afla in nivelul 3.

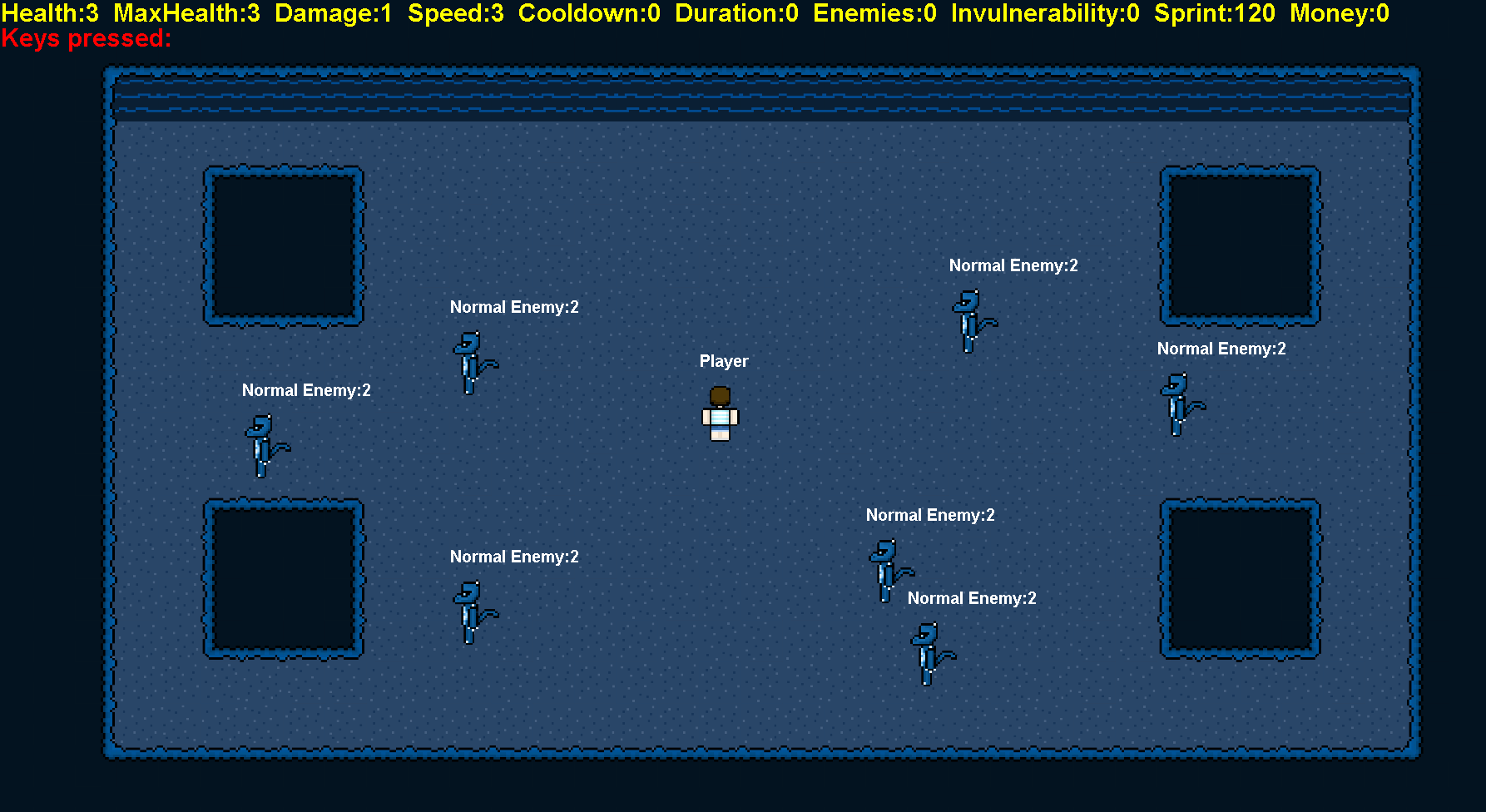
Este mult mai incet decat cel normal, dar are dimensiunea mai mare, are mai multa viata si are atacul EXTREM de puternic. Un singur atac te poate infrange!

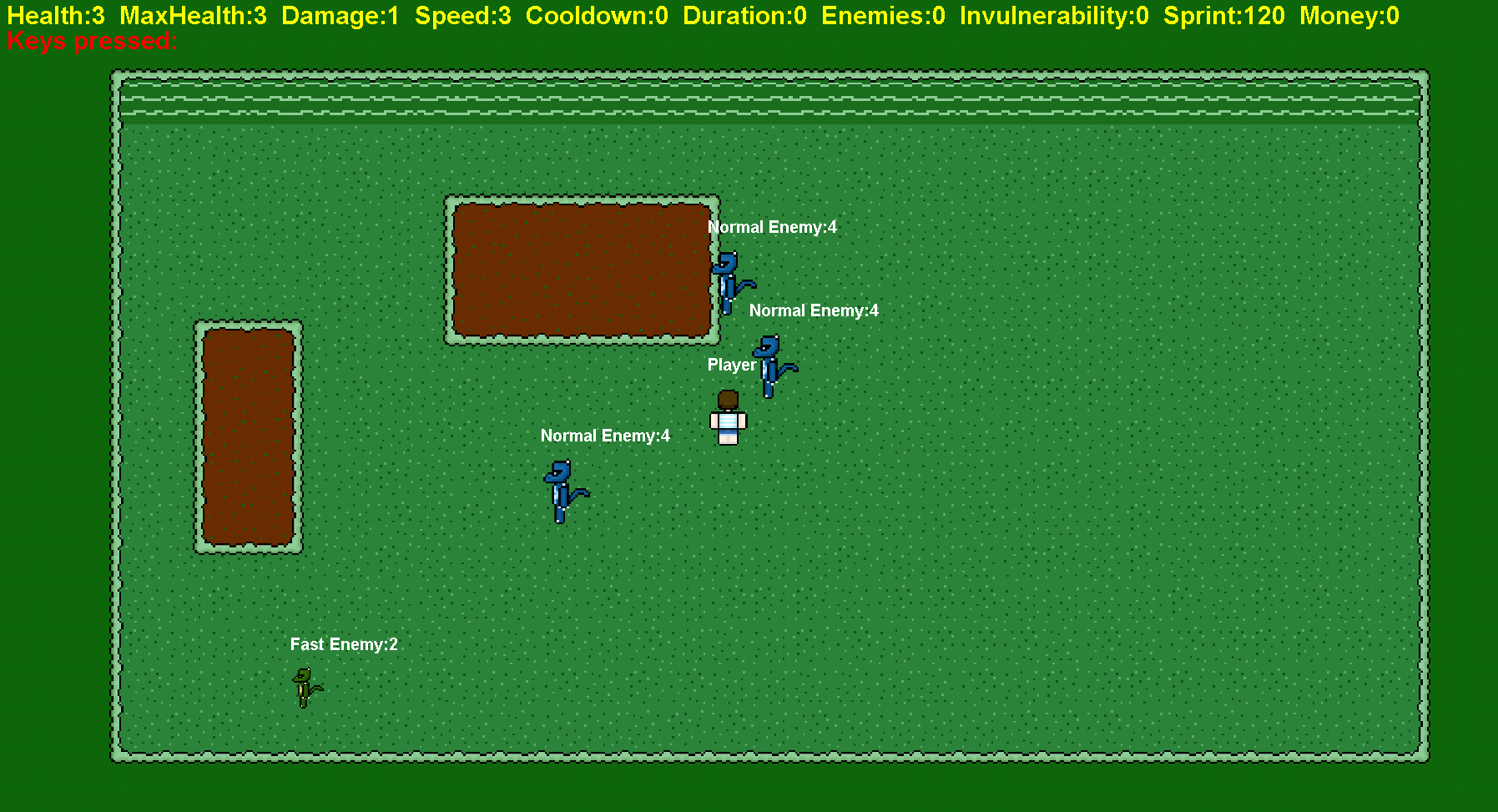
-Ghost Drake:

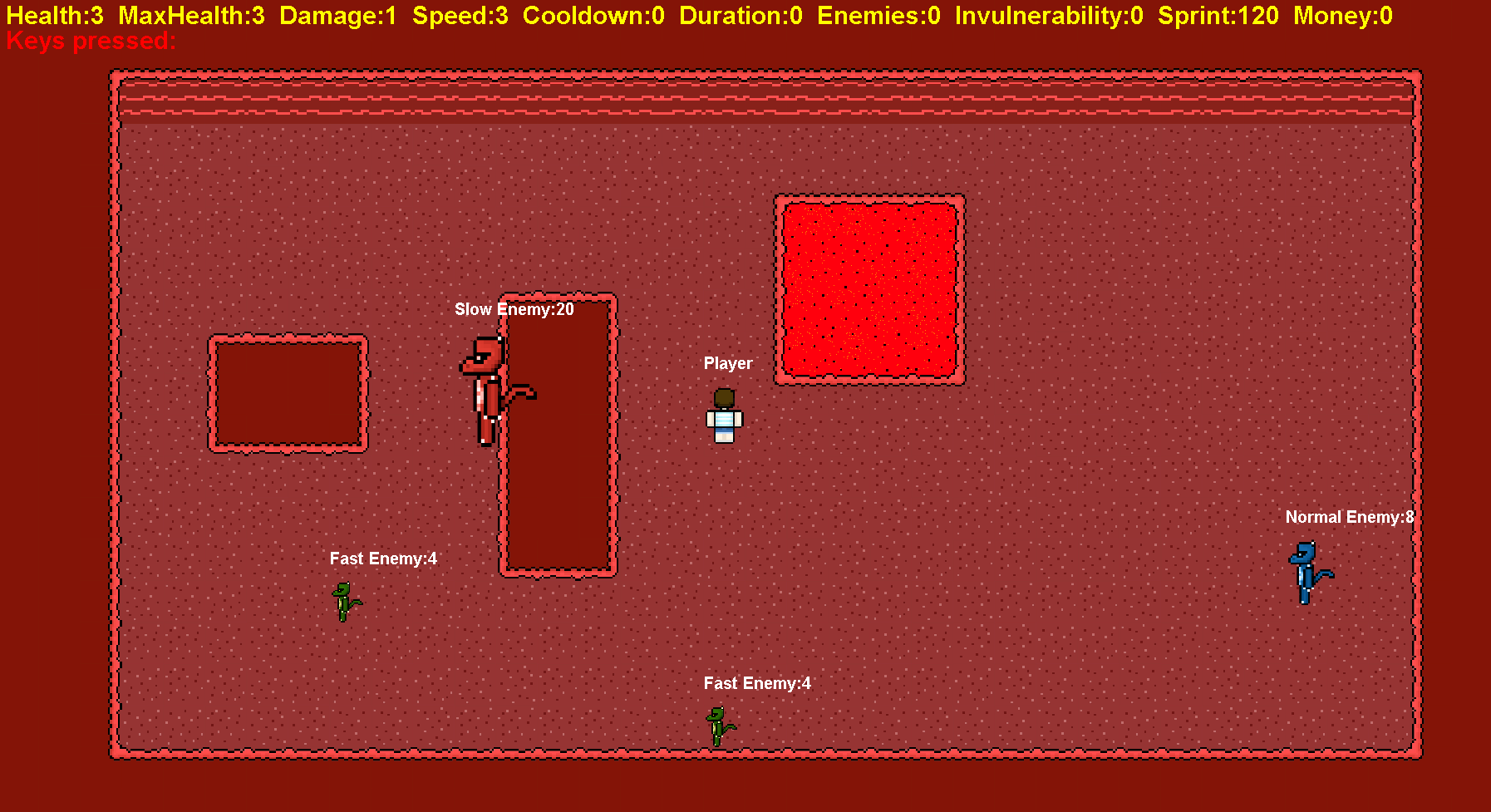
Prima aparitie se afla in nivelul 4.

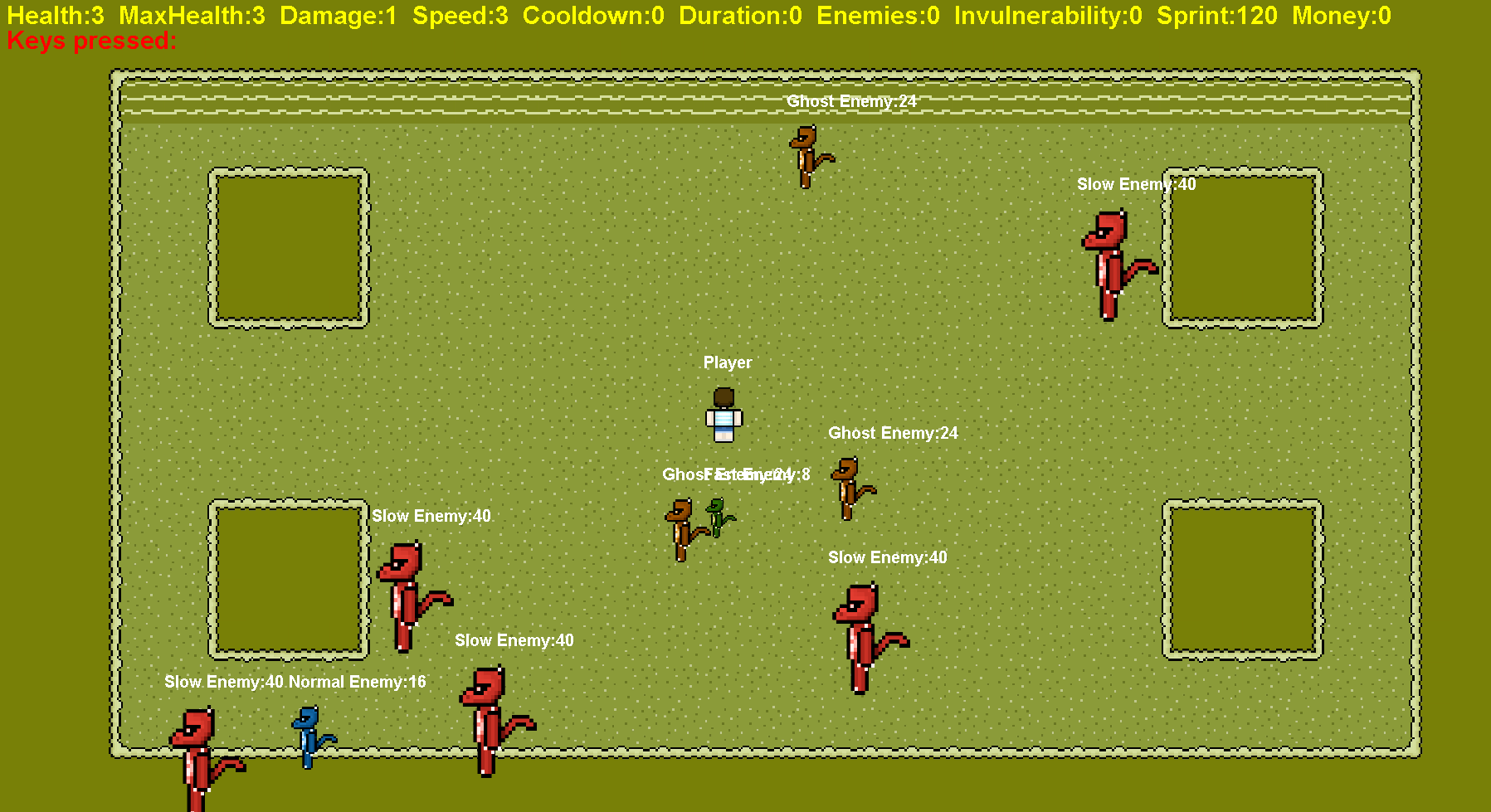
Exact ca si Normal Drake, doar ca are puterea de a trece prin perete.

**Tabla de joc:**

**Model: **

****

****

****

**- Componente passive:**

Toata actiunea este petrecuta in pestera. Primul nivel este tipul de pestera clasica. Va exista mereu un perete la marginea nivelelor care, bine inteles, vor bloca accesul in afara hartii. Exista 2 tipuri de pereti, cei normali si cei colorati.

**- Componente Active:**

Componentele active din jocul meu vor fi inamicii care vor veni constant dupa tine. Dupa infrangerea lor, vor aparea puterile speciale care te vor face mai puternic.

**- Structura tablei de joc:**

In partea de sus sunt toate informatiile necesare jucatorului. Health reprezinta viata lui curenta, MaxHealth presupun ca se intelege din nume, Damage la fel, Speed de asemenea. Cooldown reprezinta perioada de timp in care jucatorul NU poate ataca. Duration inseamna cat timp atacul jucatorului este activ. Duration se va face 0 atunci cand Enemies ajunge la 0. Acesta reprezinta cati inamici mai poate ataca jucatorul nostru cu atacul respectiv. Invulnerability reprezinta cat timp jucatorul este invulnerabil dupa ce a fost lovit. Sprint reprezinta valoare de sprint ramasa. Daca este peste 0, caracterul poate fugi. Money reprezinta cate suflete a strans ca sa poate cumpara upgrade-uri de la iselda.

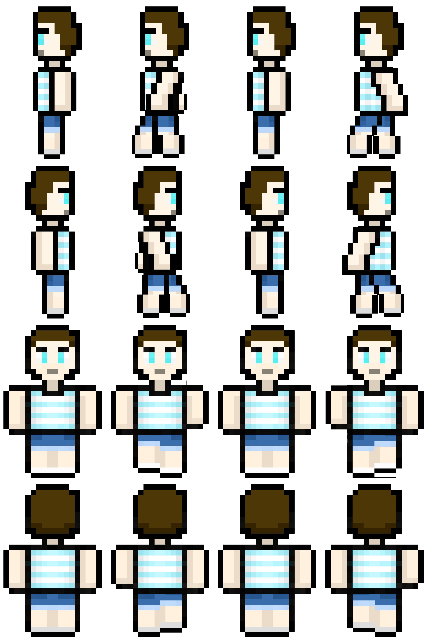
**Mecanica jocului:**

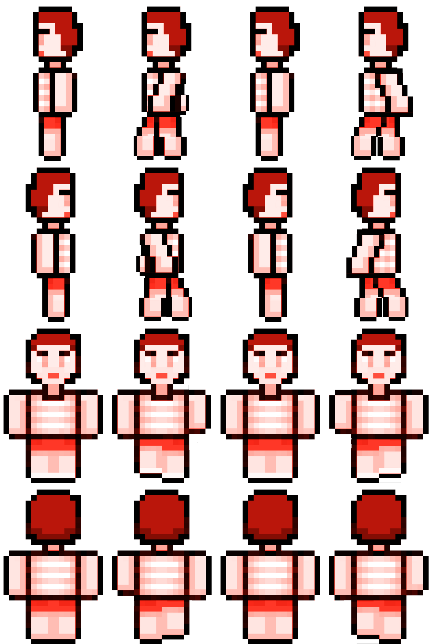
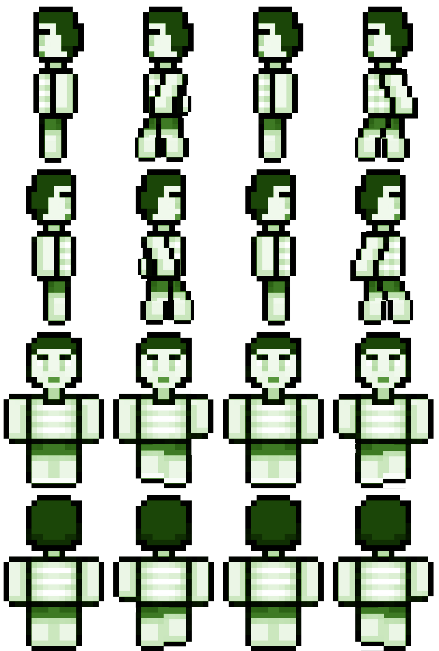
Personajul principal are 2 puteri. Sprint si Attack. Poate apasa Sprint doar atunci cand valoarea Sprint este mai mare ca 0 si aceasta ii va dubla viteza de miscare. Sprint-ul se va regenera doar atunci cand tasta specifica nu este apasata. La apasarea tastei de Attack, caracterul se va colora verde si atunci cand atinge un inamic, acesta va primi damage si valoarea de la Enemies va scadea cu unul.Jucatorul primeste bani dupa ce omaora un inamic.

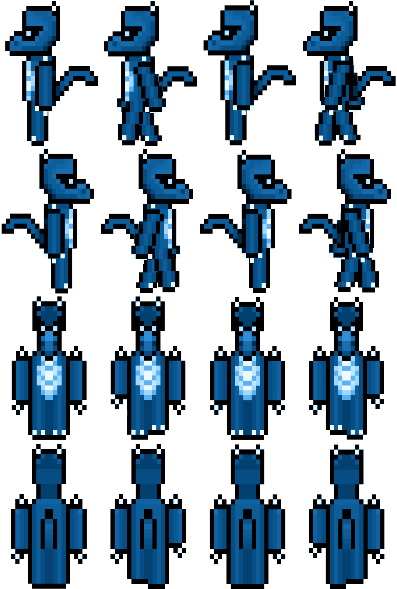
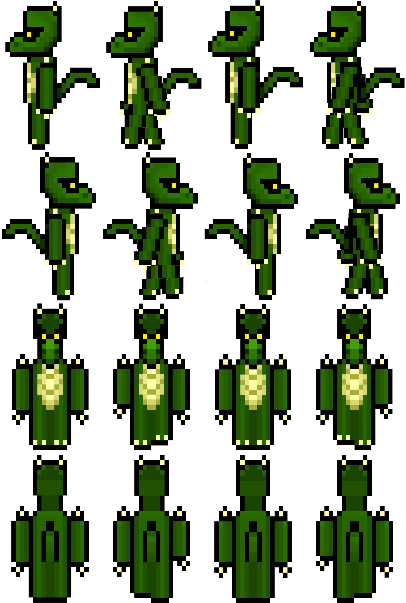
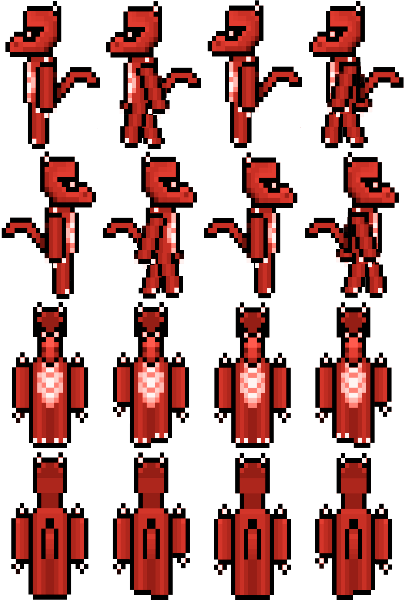
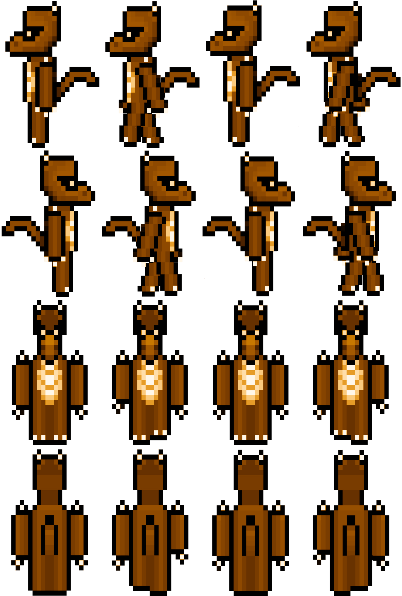
Scopul este de a omora toti inamicii din camera, de a revendica premiul si de a ajunge la finalul jocului. In total sunt 4 nivele , fiecare avand cate 6 harti.

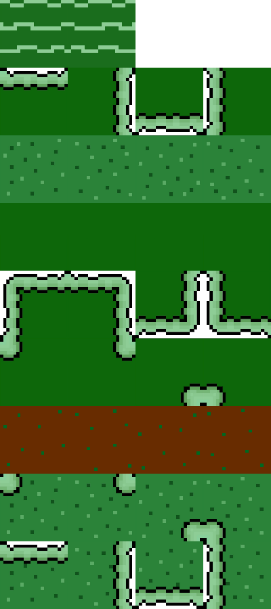
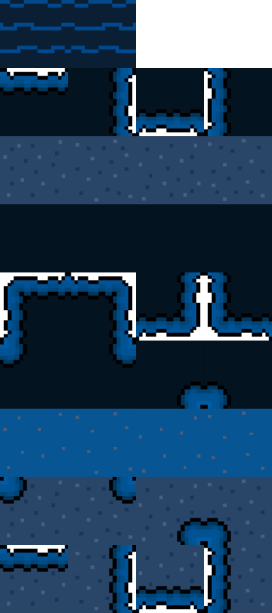
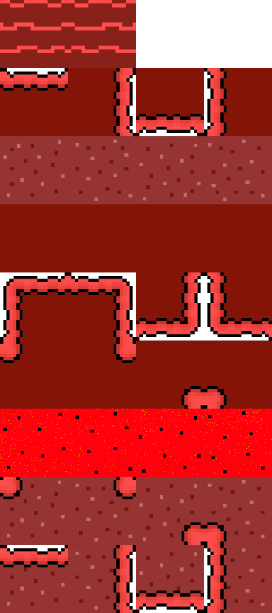
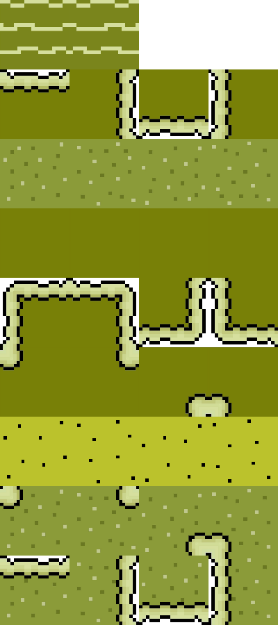
**Sprite-uri folosite:**

**Pentru caracterul principal:**

****

****

**Pentru inamici:**

**Pentru harta:**

**Pentru pickups:**

**Descriere fiecare nivel:**

Primul nivel este pestera normala.   
Al doilea nivel este o pestera de tip jungla.

Al treilea nivel va fi o pestera plina cu lava.

Al patrulea nivel a va fi o pestera tip desert.

**Descriere meniu:**

Meniul va avea optiunile de “Start new Run” care va pastra nivelele pentru power-upuri cumparate de la Iselda, dar va incepe de la primul nivel. “Start new Game” va reseta si nivelele mentionate anterior. “Continue” va pastra nivelele , toate pickup-urile obtinute si va continua de la ultimul nivel terminat.

**Descrierea claselor:**

Clasa SaveManager este o clasa care se ocupa cu adaugarea si stergerea unui save file din baza de date.

Clasa CharacterList salveaza toate caracterele pentru un nivel. De asemenea ItemList salveaza toate itemele si LevelList salveaza toate nivelele din joc.

Clasele Character, Item si Tile sunt foarte asemanatoare. Toate sunt clase de baza pentru obiectele din joc. Character este pentru jucator, inamici si Iselda. Item pentru toate power-upurile din joc si Tile pentru toate tile-urile din joc, pentru a genera harta.

Interfata observer este interfata de baza pentru cei 2 observeri, PlayerSystem si DrakeSystem.

Fiecare din acestea este un observer si priveste actiunea pentru fiecare jucator. PlayerSystem se ocupa de cooldown, duration, damage, receivedamage, etc. si DrakeSystem se ocupa de DealDamage si ReceiveDamage.

Cele 2 functii comune sunt ReceiveDamage si Update. In clasa Character se va apela myObserver.update si ReceiveDamage se va apela in cazuri speciale, cum ar fi atunci cand un inamic loveste un jucator si verifica starea lui. Daca este in modul atac va apela this.receiveDamage, daca nu va apela Player.receiveDamage.

Clasa Player este de tip singleton. Contine un static Player instance, constructorul este privat, functia getInstance apeleaza constructorul daca instance este null si reset care sterge obiectul.

Clasa Collision se ocupa de coliziune.

Clasa PowerUpSystem se ocupa de informatiile legate de jucator. De exemplu el verifica ce nivel are atacul si il modifica respectiv, cate powerupuri a ridicat, ce fel de powerupuri, cate nivele are din fiecare, viata lui standard, viteza lui standard, etc.

LevelingManager este clasa care se ocupa cu meniul de level up care se deschide atunci cand se apasa tasta E aproape de Iselda si va deschide o fereastra de tip LevelingMenu.

Leveling menu este un grid de 8/2 unde prima valoare de pe fiecare coloana va fi un alt tip de grid de dimensiune 2/1. Prima valoare de pe fiecare coloana va avea butonul de Levelup si un Drawing Panel care deseneaza cate puncte are puse in fiecare abilitate, singura exceptie fiind ultima linie care contine doar butonul Confirm. A doua valoare de pe coloana va fi imaginea respectiva pentru fiecare nivel.

ControlPanel se ocupa cu butoanele din Main Menu.

EmptySpace, dupa nume, este un spatiu gol. Este folosit pentru a lasa un spatiu intre bara de sus a meniului si ControlPanel.

Clasa DataBase este de asemenea un Singleton si se ocupa de conectiunea la baza de date.

Clasa NextLevel se ocupa cu trecerea de la un nivel la altul si apeleaza functia care salveaza starea jocului.

Clasa Main doar creeaza un nou MainMenu.

Nu o sa explic ce fac clasele care existau deja in ScheletProiectPAOO.

**Baza de date:**

Baza de date este un singur tabel numit “Save” care contine o singura linie si aproximativ 20 de coloane. Coloanele vor fi ID, currentLevel, seed, 8 coloane reprezentand puterile cumparate de la iselda si alte 8 coloane reprezentand puterile gasite, currentHealth si money.

Exista si un Trigger care se executa inainte de Insert care goleste baza de date.

**Variabila Seed:**

Variabila seed mi se pare oleaca mai interesanta decat restul si am zis ca se merita sa scriu o sectiune special pentru aceasta. Ideea a fost inspirata din alte jocuri. Stiam de existenta seed-ului din alte jocuri, care reprezinta modul cum este generata harta. Dupa ce se apasa unul din cele 3 butoane din MainMenu se va deschide jocul. In acest moment TOATE nivelele, inamicii, powerup-urile, iesirile, itemele vor fi generate, in exact aceeasi ordine. TOATE se vor genera random folosind variabila numita rand de tip Random din clasa Game. Multumita acestui lucru, toate valorile se vor genera mereu in aceeasi ordine. M-am documentat putin despre clasa Random si aceasta are un camp numit seed. Toate valorile generate de clasa Random nu sunt 100% random. Sunt generate dupa un anumit seed. Eu salvez acest seed atunci cand sau save la joc si acest lucru imi garanteaza ca generarea “Random” va fi identica cu cea anterioara. La inceputul jocului, se va genera un seed random. Daca jucatorul a decis sa continue jocul precedent. Seed-ul va fi inlocuit cu cel din baza de date, si harta se va genera identic cu acel moment. Daca nu, se va pastra un seed generat random.

**Sabloane de proiectare:**

Clasa Player si clasa Database sunt de tip Singleton. Amandoua au constructor privat, o variabila statica pentru obiectul respectiv ,functia getInstance si functia reset.

Clasa PlayerSystem si DrakeSystem sunt de tip Observer. Amandoua au functia update si se apeleaza in functiile Update din obiectele atasate.

Obiectul de tip player va apela Player.attach(new PlayerSystem()) si inamicii vor apela attach(new DrakeSystem())

O bucata din functia Update aflata in Character care demonstreaza faptul ca se folosesc observeri.

public void Update(int frame) {  
 currentframe++;  
 if (myObserver != null) {  
 myObserver.Update();  
 }

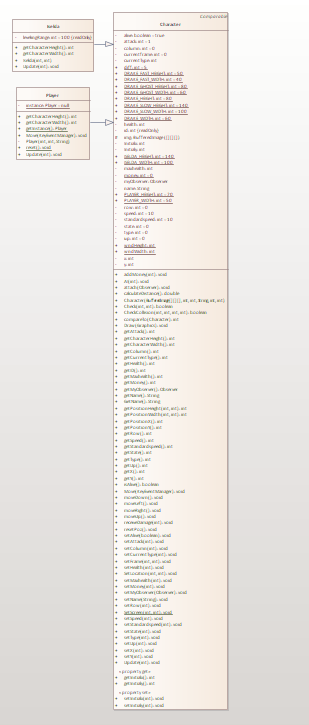
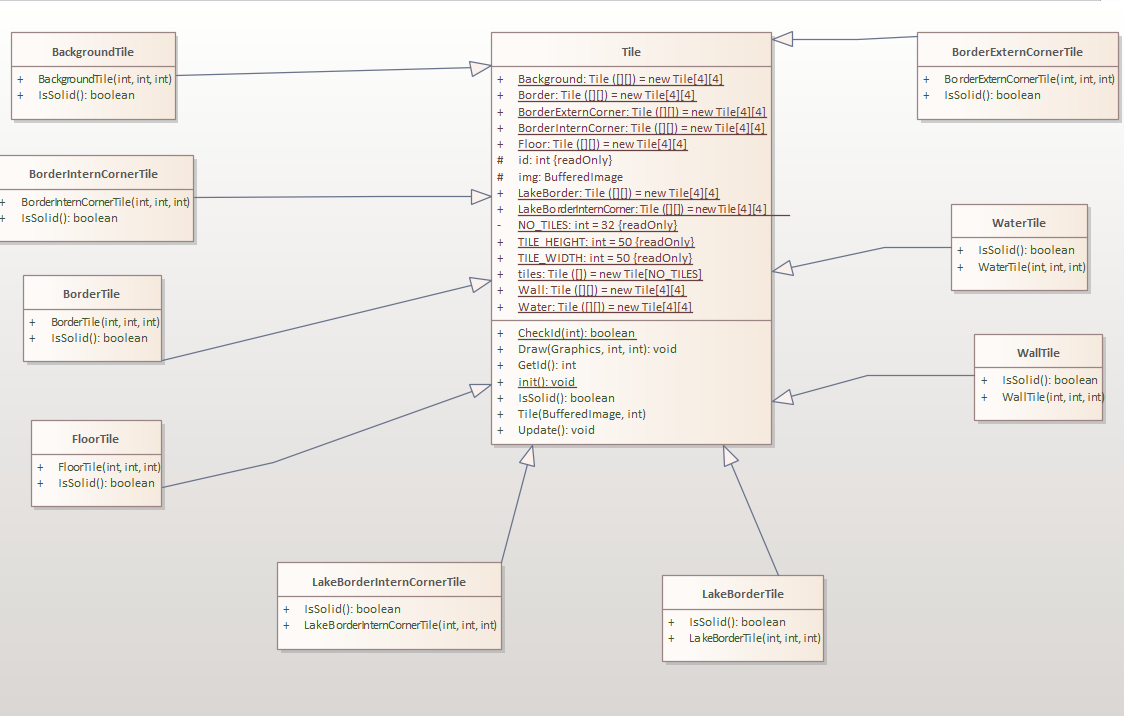
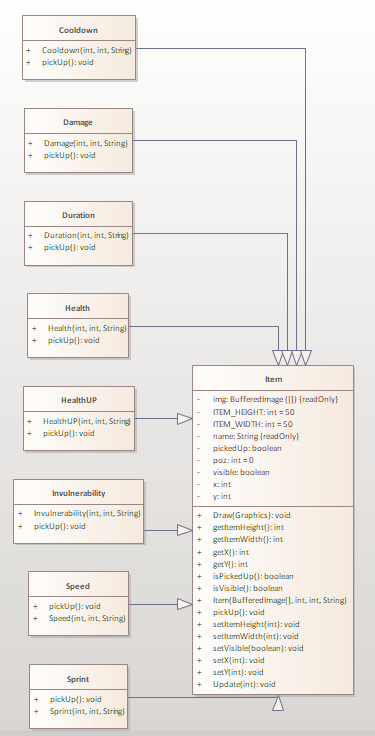
O bucata din clasa Player care demonstreaza ca este de tip Singleton si care adauga un observer la acesta.

public static Player getInstance() {  
 if (*instance* == null) {  
 *instance* = new Player(0, 0, "Player");  
 *instance*.attach(new PlayerSystem(*instance*));  
 }  
 return *instance*;  
}

**Tipul fisierelor folosite:**

Toate fisierele sunt de tip .java. Texturile sunt de tip .png si baza de date este de tip .sqlite.

**Diagramala UML**



Pentru diagrama generate de doxygen, trebuie deschis fisierul numit “DESCHIDETI PENTRU DIAGRAMA”

**Bibliografie:**

Aplicatia folosita pentru a crea sprite-urile [aici](https://www.aseprite.org/).

Am mai cautat secvente mici de cod pe stackoverflow. Nu am pastrat link pentru fiecare dintre acestea pentru ca nu influenteaza major aplicatia si nici nu sunt foarte mari secventele. Un exemplu este functia “resize” care se afla in Control Panel unde redimensioneaza o poza dupa dimensiunile primite si functia de “add text”, in acelasi loc, care adauga text la o imagine.

Am mai folosit cateva chestii de pe stackoverflow, dar nu mai tin minte exact ce si unde.