Aranara Flower Garden
Galatanu Marco
1207B

1 pct oficiu pentru aspect, respectare cerințe

Povestea jocului:(1 pct)

"Aranara Flower Garden" este un joc de tip puzzle, creat special pentru a testa abilitatile jucatorilor de a găsi cele mai bune soluții pentru probleme complexe. Personajul principal al jocului este Traveler, un calator curajos, care ajunge pe planeta Teyvat și se intalneste cu un trunchi de copac fermecat numit Todd.

Todd ii da lui Traveler o sarcina importanta: să planteze semintele unei flori vitale pentru populația planetei Teyvat. Dar, exista o problema: zona în care trebuie sa fie plantate semintele este foarte mare, iar Traveler trebuie sa găsească o soluție pentru a acoperi întreaga zona fără a trece de doua ori prin același pătrați.

Prezentare joc:(1 pct)

"Aranara Flower Garden" este un joc distractiv și educativ, care pune la încercare abilitățile de rezolvare a puzzle-urilor ale jucatorilor, dar și creativitatea si imaginatia lor in crearea unor gradini virtuale unice.

Reguli joc:(1 pct)

1. Scopul jocului este de a planta toate semintele din zona de plantare fără a trece de două ori prin aceeași zonă.
2. Jucătorul începe într-un colț al zonei de plantare și trebuie să exploreze împrejurimile pentru a găsi locurile unde trebuie plantate semintele.
3. Jucătorul poate merge în toate direcțiile, cu excepția zonei prin care a trecut deja.
4. Fiecare zonă de plantare are o anumită formă și dimensiune, iar jucătorul trebuie să găsească calea corectă pentru a ajunge la fiecare zonă.
5. Dacă jucătorul se blochează și nu mai poate merge înainte, poate apăsa un buton de resetare pentru a începe din nou din colțul de pornire, dar pierde un punct de HP.
6. Jocul este câștigat atunci când toate semințele sunt plantate în toate zonele de plantare.

Personajele jocului:(1 pct)

-Traveler este personajul principal. Scopul lui este de a îndeplini sarcinile date de creaturile magice din Teyvat (planeta pe care acesta a ajuns).

-Todd este un trunchi de copac magic. Rolul lui este de a furniza sarcinile jucătorului.

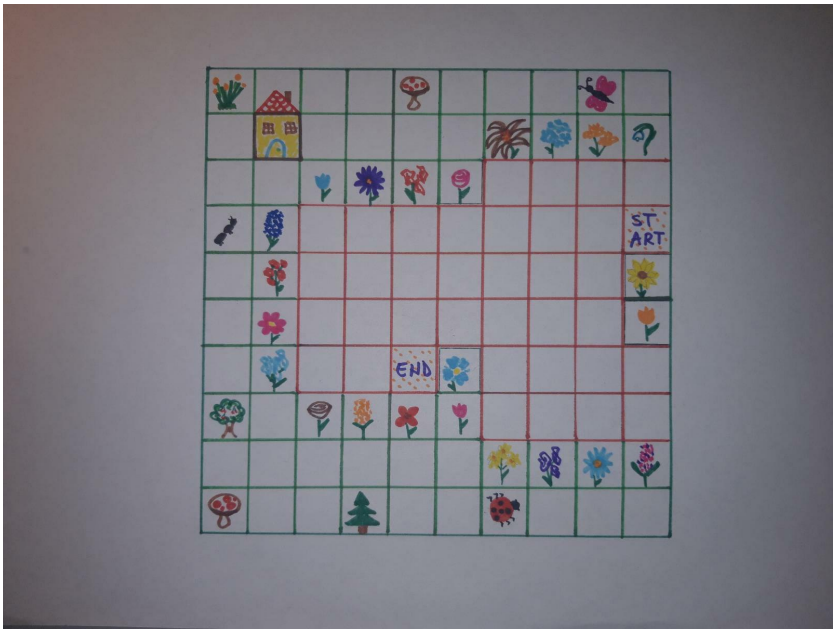
Tabla de joc (2pct):

Tabla de joc este formată din dale de diferite dimensiuni - 16x16, 32x32 sau 64x64 pixeli dispuse într-un model pătratic de dimensiune 10x10 sau alte variante.

Componentele pasive ale tablei sunt reprezentate de texturile de iarbă, sol, pietre și copaci care definesc aspectul general al nivelului de joc.

Componentele active ale tablei includ terenurile speciale unde se pot planta semințe și unde se pot crește și îngriji plantele, care adesea deblochează noi zone de plantare. Plantele speciale pot fi de diverse culori și dimensiuni și pot fi colectate pentru a debloca noi niveluri.


End
Start


ST
ART
END

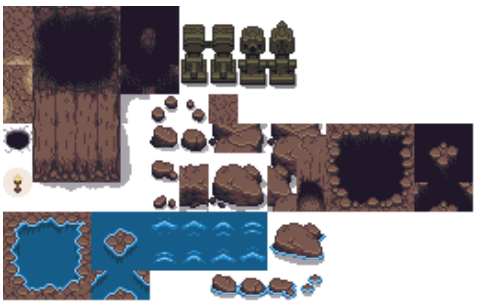
Mecanica jocului (1 pct)

În unul dintre colțurile ecranului vor fi prezente diferite informații, cum ar fi viața jucătorului sau timpul pe care-l mai are la dispoziție.

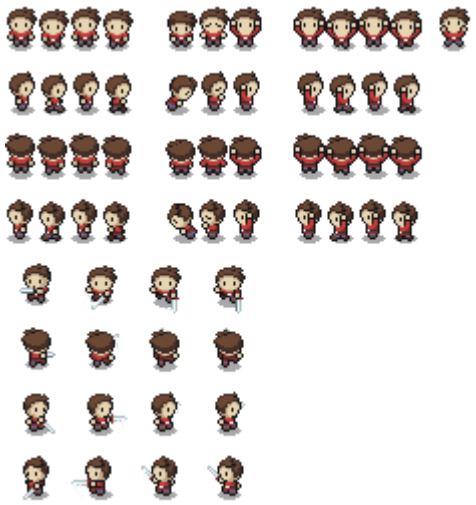
Unele zone ale tablei pot fi blocate inițial și pot fi accesate doar după ce jucătorul a colectat plantele speciale necesare pentru a debloca aceste zone. Aceste zone noi pot conține alte tipuri de plante speciale, care adaugă noi puzzle-uri pentru jucător.

Jucătorul trebuie să planteze semințele pe terenurile active fără a trece de două ori prin aceeași zonă a tablei. Acesta trebuie să planifice cu atenție mișcările și să găsească calea corectă pentru a planta și îngriji plantele în timp ce explorează și deblochează noi zone ale tablei de joc.

Game sprite (1 pct)




AaBbCcDdEeFfGgHhIiJjKkLlMm 012
NnOoPpQqRrSsTtUuVvWwXxYyZz 345
678
9
.,!¡?¿#_-♥:;'"



1 pct pentru:
Descriere fiecare nivel

Nivelul 1: Gradina pitorească

In acest nivel, jucătorii trebuie sa planteze semintele florilor în jurul unei case pitorești aflate la marginea pădurii. Zona este limitata de un gard în jurul casei și un munte în spatele acesteia. Resursele grafice disponibile sunt iarba, gardul, casa și muntele.

Nivelul 2: Parcul împărătesc

In acest nivel, jucătorii se vor concentra pe plantarea florilor în jurul unei frumoase fantani din mijlocul unei grădini în aer liber. Acest nivel este mult mai mare decat primul nivel și are mai multe obstacole, cum ar fi copacii și un castel în apropiere. Jucătorii trebuie sa-si folosească abilitățile de gandire logica și creativitatea pentru a găsi cea mai buna soluție. Resursele grafice disponibile sunt apa, iarba, copacii, fantana si castelul.

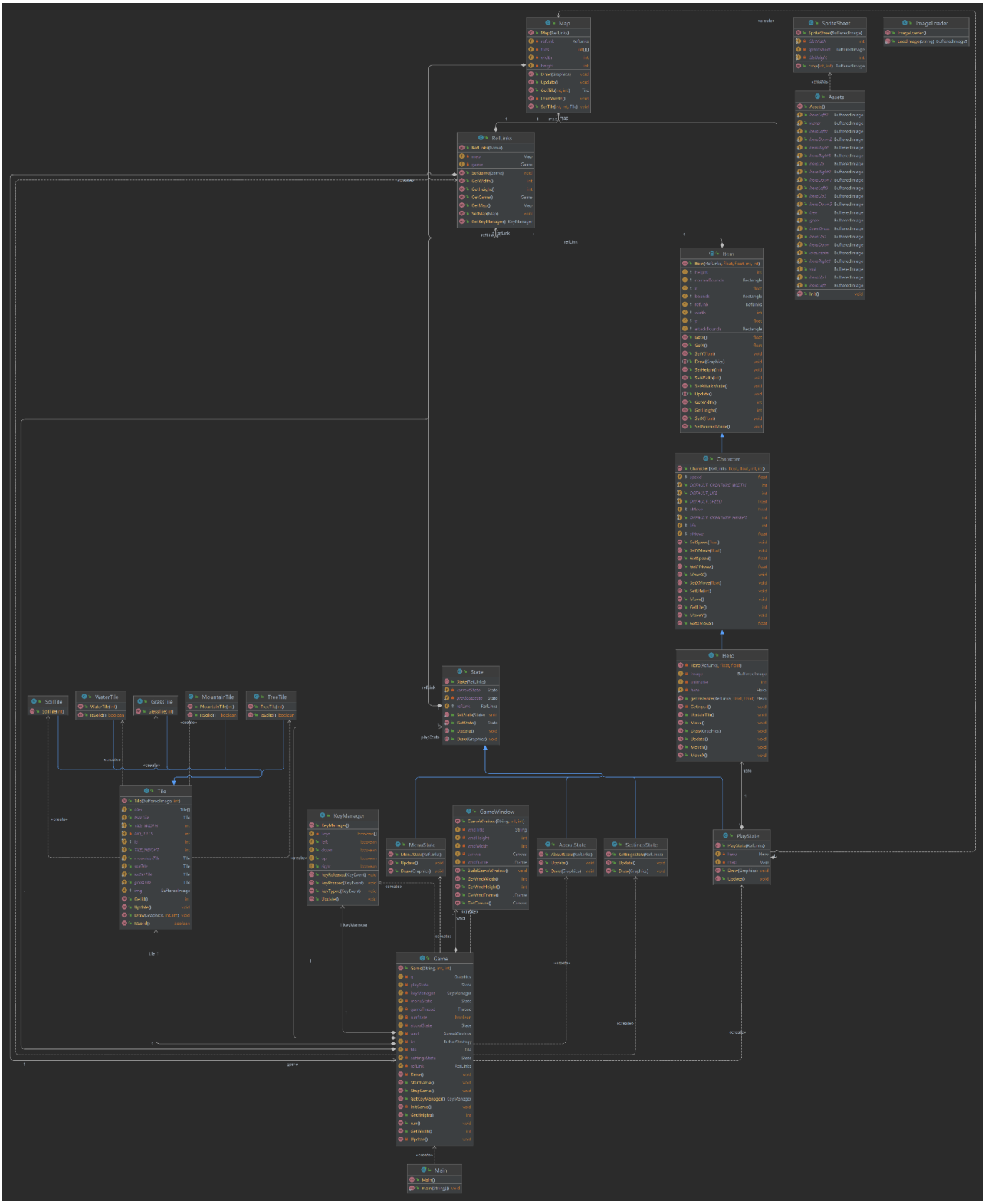
Nivelul 3: Lacul învăluit de munți

In acest nivel, jucătorii trebuie sa planteze semintele florilor în jurul unui lac mare situat la baza unui munte înalt. Zona este limitata de apă și de munte, ceea ce face dificila

plantarea florilor in unele zone. Resursele grafice disponibile sunt apa, iarba, muntele si copacii, ceea ce face acest nivel unul dintre cele mai complexe și provocatoare nivele din joc.

Etapa 2:

Diagrama UML



Descriere:

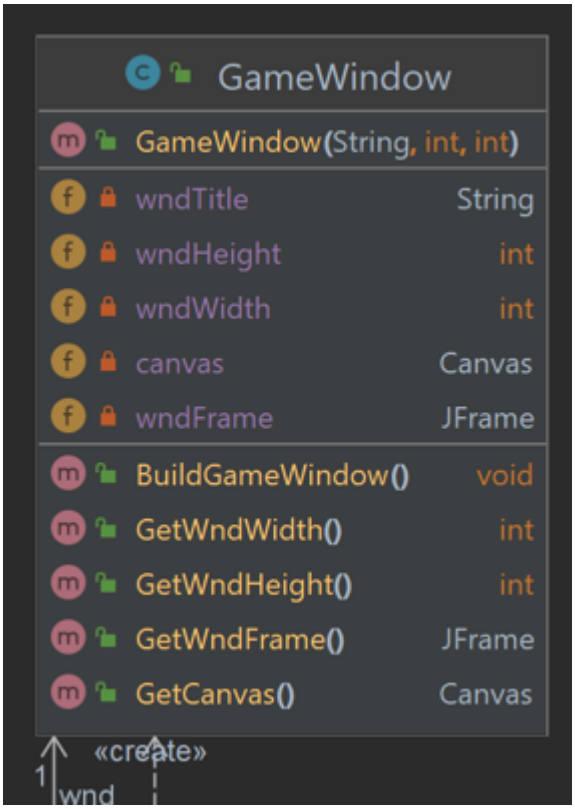
-pachetul PaooGame contine toate pachetele (GameWindow, Graphics, Input, Items, Maps, States, Tiles) dar si clasele:

Main (clasa in care se creeaza obiectul de tip Game si se apeleaza metoda StartGame()).

Game (clasa principala a intregului proiect. Implementeaza Game-Loop (Update -> Draw))

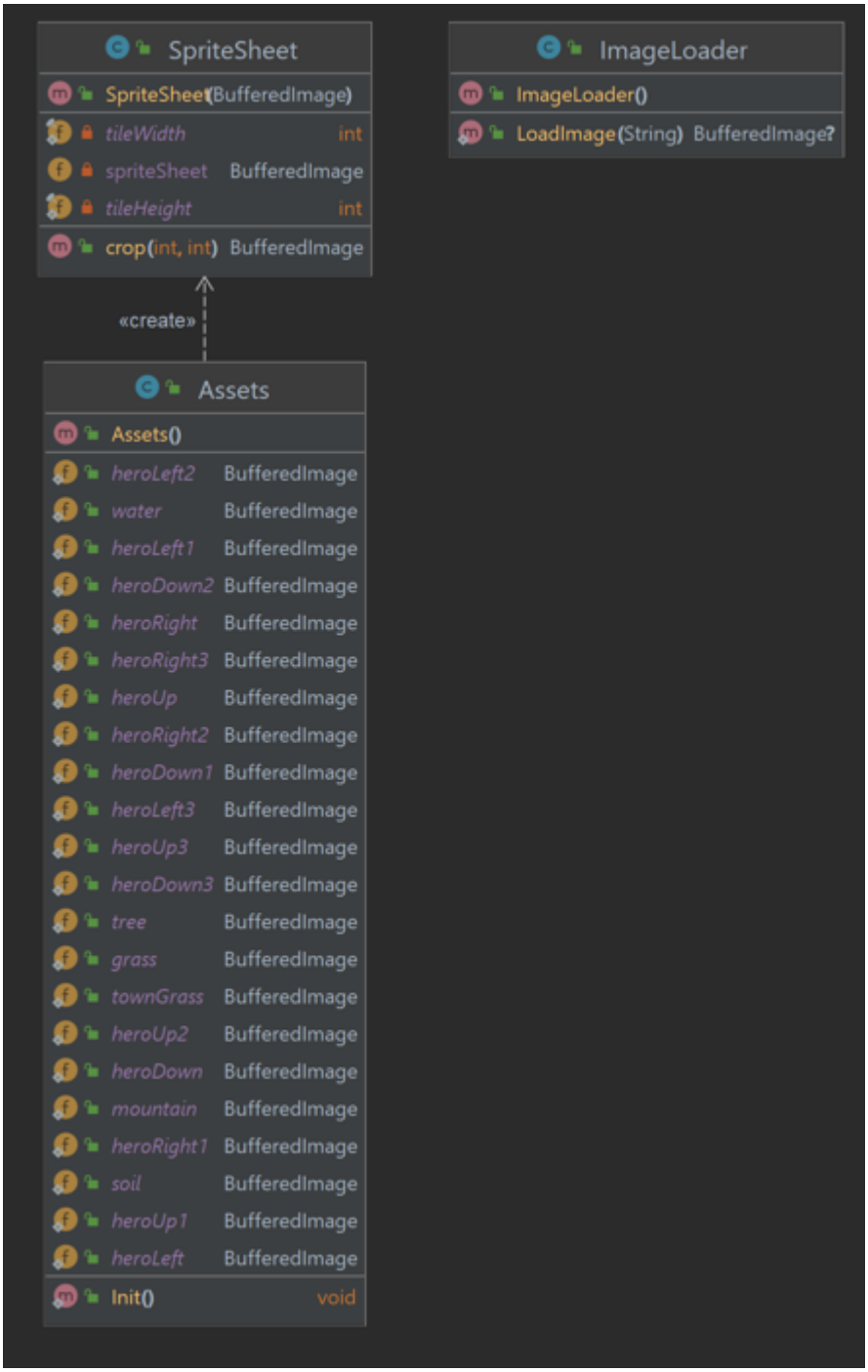
Reflinks (retine o serie de referinte ale unor elemente pentru a fi usor accesibile).

-pachetul GameWindow contine clasa:



GameWindow: implementeaza notiunea de fereastra a jocului. Membrul wndFrame este un obiect de tip JFrame care va avea utilitatea unei ferestre grafice si totodata si cea a unui container (toate elementele grafice vor fi continute de fereastra).

-pachetul Graphics contine clasele:



Assets: (Clasa ce incarca fiecare element grafic necesar jocului)

- Assets include tot ce este folosit intr-un joc: imagini, sunete, harti etc.
- Metoda Init() initializaza referintele catre elementele grafice utilizate.

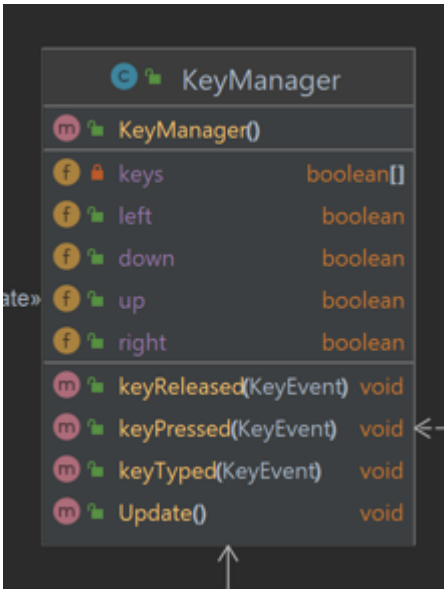
ImageLoader: (contine o metoda statica pentru incarcarea unei imagini in memorie)

- Metoda LoadImage() incarca o imagine intr-un obiect BufferedImage si returneaza o referinta catre acesta.

SpriteSheet: (retine o referinta catre o imagine formata din dale)

- Metoda crop() returneaza o dala de dimensiuni fixe (o subimagine) din sprite sheet de la adresa (x * latimeDala, y * inaltimeDala)

-pachetul Input contine clasa:



KeyManager: (gestioneaza input-ul de tastatura.)

- Clasa citeste daca au fost apasata o tasta, stabiliteste ce tasta a fost actionata si seteaza corespunzator un flag.In program trebuie sa se tina cont de flagul aferent tastei de interes. Daca flagul respectiv este true inseamna ca tasta respectiva a fost apasata si false nu a fost apasata.
- Metoda Update() este folosita in metoda Update() din clasa **Game** pentru a determina starea tastelor

-pachetul Items contine clasele:

Item: (Implementeaza notiunea abstracta de entitate activa din joc, "element cu care se poate interactiona: monstru, turn etc.".)

- Constructorul initializeaza obiectul cu anumite caracteristici (creeaza dreptunghi de coliziune pentru modul normal, creeaza dreptunghi de coliziune pentru modul de atac, iar dreptunghiul de coliziune implicit este setat ca fiind cel normal).
- Metoda abstracta Update() este destinata actualizarii starii curente.
- Metoda abstracta Draw() este destinata desenarii starii curente.
- Clasa mai contine si diferite gettere si settere pentru pozitie(x,y), caracteristici ale bounding box-ului, dar si al starii item-ului.

Character: (Defineste notiunea abstracta de caracter/individ/fiinta din joc.)

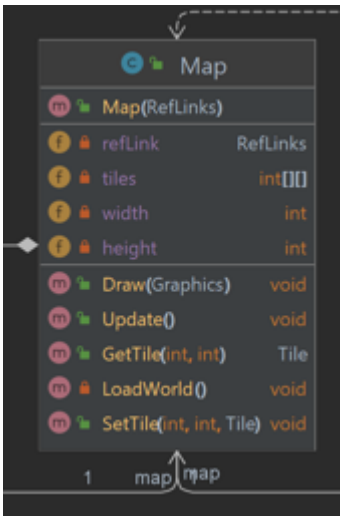
- Clasa este mostenita din clasa Item si adauga aspecte in plus cum ar fi: viata, viteza de deplasare si distanta cu care trebuie sa se miste/deplaseze in urma calculelor (in toate directiile trebuie sa se miste cu aceeasi viteza)
- Metoda Move() modifica pozitia caracterului folosindu-se de alte 2 metode MoveX() si MoveY() ce ii modifica pozitia pe axele respective.
- Ca si clasa Item, aceasta clasa contine si ea diferite settere si gettere pentru proprietatile noi adaugate

Hero: (Implementeaza notiunea de erou/player (caracterul controlat de jucator))

- Clasa este mostenita din clasa Character si adauga aspecte in plus cum ar fi: imaginea, deplasarea si dreptunghiul de coliziune.

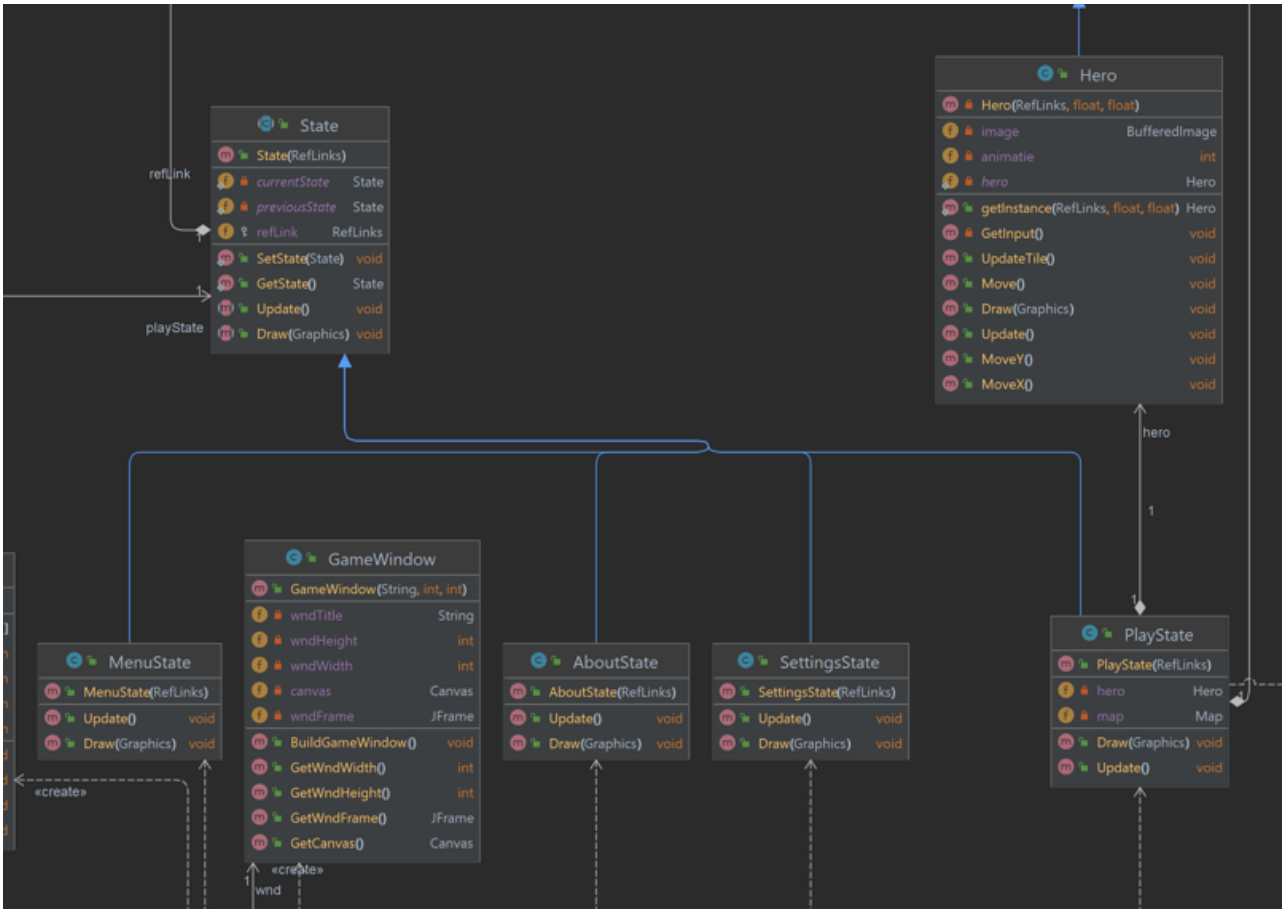
- Implementeaza un design pattern de Singleton, adica o singura instanta a eroului poate exista, acest design pattern se caracterizeaza prin constructorul privat, metoda statica getInstance ce returneaza referinta catre eroul deja creat, sau catre cel creat daca acesta era null.
- Metoda Move() suprascrie metoda Move() din clasa de baza, adaugand aspectul coliziunilor cu tile-uri si cu window-ul in care se afla eroul.
- Metoda UpdateTile() modifica tile-ul pe care calca eroul, asigurandu-se astfel ca marcheaza pe unde a trecut, pentru a nu se mai intoarce pe acelasi drum.
- Metoda Update() verifica daca a fost apasata o tasta, actualizeaza pozitia, actualizeaza harta si apoi actualizeaza imaginea jucatorului.
- Metoda Draw() deseneaza eroul in noua pozitie.

-pachetul Maps contine clasa:



Map: (Implementeaza notiunea de harta a jocului.)
- Constructorul Map() incarca harta de start folosindu-se de metoda LoadWorld()
- Metoda LoadWorld() incarca din fisierul Map.txt harta, citind de pe prima linie a fisierului cate linii si cate coloane are harta, apoi cu un for citeste in tiles[x][y] valorile din fisier.

-pachetul States contine clasele:



State: (Implementeaza notiunea abstracta de stare a jocului/programului.)

- Din ea se vor deriva toate starile urmatoare: AboutState, MenuState, PlayState, SettingsState
- Metoda SetState() seteaza starea curenta a jocului, va fi folosita pentru tranzitia dintr-o stare in alta (Meniu -> Setari, Meniu -> About, Meniu -> Play)
- Metoda GetState() returneaza starea curenta a jocului, pentru a putea actualiza starea curenta din clasa Game.

AboutState: (Implementeaza notiunea de credentiale.)

- Momentan este o clasa ce urmeaza a fi implementata
- Metoda Update() va actualiza pozitia credentialelor.
- Metoda Draw() va afisa pe ecran credentialele la noua lor pozitie.

MenuState: (Implementeaza notiunea de meniu pentru joc.)

- Momentan este o clasa ce urmeaza a fi implementata
- Metoda Update() va actualiza starea in cazul in care este selectata vreo optiune din meniu
- Metoda Draw() va afisa meniul, iar daca vreo optiune este pe cale de a fi selectata, aceasta va fi afisata cu alta culoare.

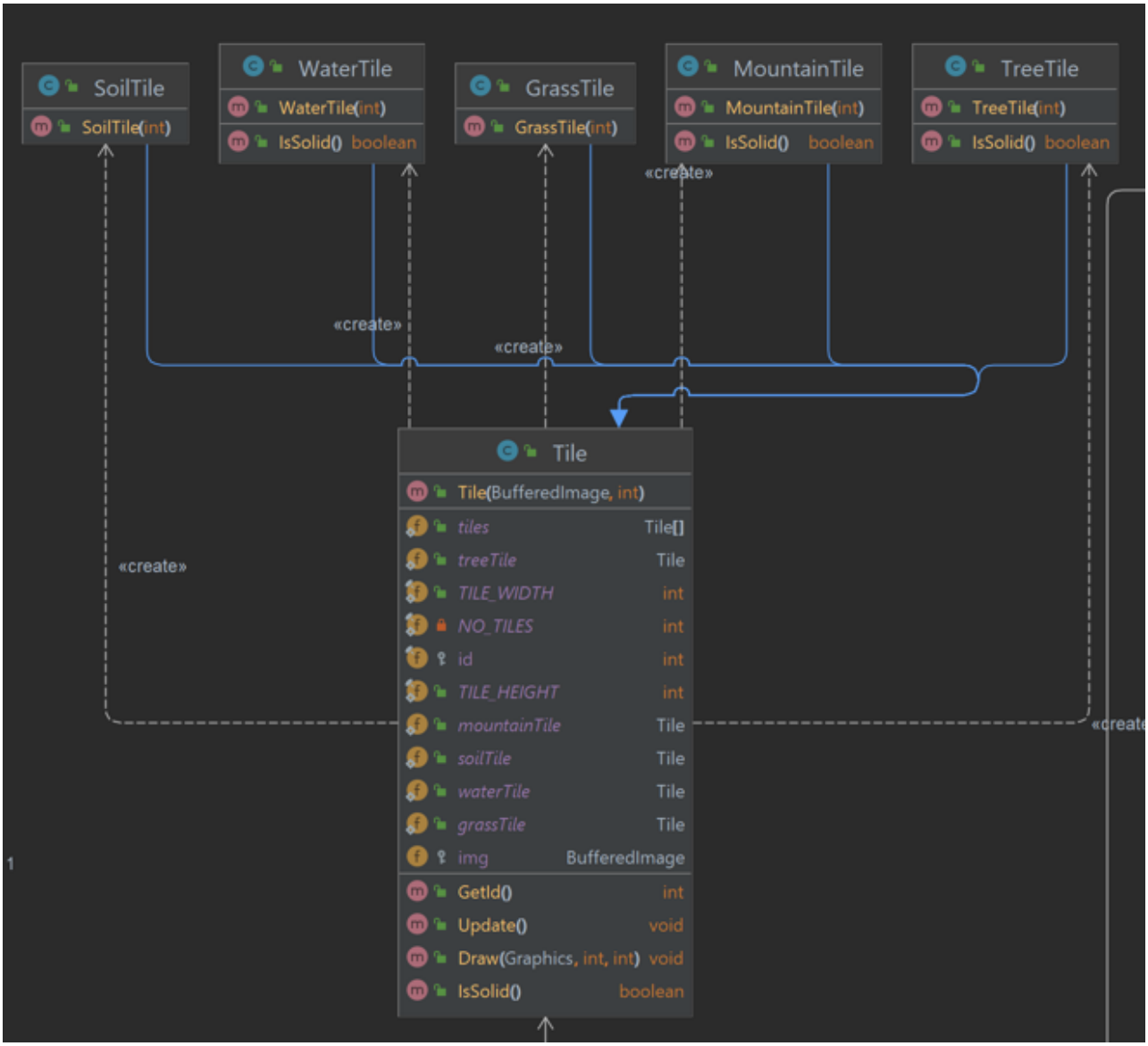
PlayState: (Implementeaza/controleaza jocul si eroul.)

- Metoda Update() actualizeaza starea curenta a jocului, apeland map.Update() si hero.Update(), actualizand starea mapei si starea eroului.
- Metoda Draw() deseneaza pe ecran starea curenta a jocului, apeland la fel ca si metoda Update(), mai intai map.Draw() apoi hero.Draw().

SettingsState: (Implementeaza notiunea de setari ale jocului)

- Momentan este o clasa ce urmeaza a fi implementata
- Metoda Update() ca actualiza starea setarii respective in cazul in care este modificata vreo setare
- Metoda Draw() va afisa setarile, iar daca vreo setare este pe care de a fi modificata, aceasta va fi afișată cu alta culoare pentru a sti pe ce setare ne situăm.

-pachetul Tiles contine clasele:



Tile: (Reține toate dalele într-un vector și oferă posibilitatea regasirii după un id.)

- Metoda IsSolid() returneaza in mod implicit fals, pentru a specifica ca tile-ul nu este solid, dar clasele ce vor moșteni aceasta clasa (toate tile-urile folosite pentru hartă) vor suprascrie aceasta metoda în cazul în care vor fi solide.
- Metoda Draw() va desena dala pe ecran la coordonatele x si y din matricea de dale, înmulțite cu lățimea și înălțimea unei dale.
- Metoda GetId() returnează id-ul dalei care este folosit în matricea de dale.

GrassTile: (Abstractizeaza noțiunea de dala de tip iarba.)

- Are un constructor ce apelează constructorul clasei de baza Tile cu Assets.grass și id-ul respectiv dalei de iarba din matrice.

MountainTile: (Abstractizeaza noțiunea de dala de tip munte sau piatra.)

- Are un constructor ce apelează constructorul clasei de baza Tile cu Assets.mountain si id-ul respectiv dalei de munte din matrice.
- Suprascrie metoda IsSolid() din clasa de baza Tile pentru a returna adevărat.

SoilTile: (Abstractizeaza noțiunea de dala de tip pamant.)

- Are un constructor ce apelează constructorul clasei de baza Tile cu Assets.soil și id-ul respectiv dalei de pamant din matrice.

TreeTile: (Abstractizeaza noțiunea de dala de tip copac.)

- Are un constructor ce apelează constructorul clasei de baza Tile cu Assets.tree și id-ul respectiv dalei de copac din matrice.
- Suprascrie metoda IsSolid() din clasa de baza Tile pentru a returna adevărat.

WaterTile: (Abstractizeaza noțiunea de dala de tip apa.)

- Are un constructor ce apelează constructorul clasei de baza Tile cu Assets.water și id-ul respectiv dalei de apa din matrice.
- Suprascrie metoda IsSolid() din clasa de baza Tile pentru a returna adevărat.

Descriere meniu

Joc nou
    Dificultate
    Nivel
        Gradina pitorească
        Parcul împărătesc
        Lacul învăluit de munți
    Mod de joc
        TimeAttack
        HighScore
Jocuri salvate
    Save1 (ex: Marco_1)
    [Save2]
Setari
    Video
    Audio
    Controale
Credits
    Informații despre resursele folosite etc.

Bibliografie

https://game8.co/games/Genshin-Impact/archives/387328
https://opengameart.org/content/zelda-like-tilesets-and-sprites