Osztrogonácz Dóra
UHI5AX

Farm játék

Felhasználói dokumentáció

Játék célja

A játékos ebben a játékban kipróbálhatja a gazda szerepet, igazgathatja a farmját, növényeket ültethet és learathat. A játék célja a növények vásárlása, termesztése majd eladása. A játéknak alapvetően nincs vége, a felhasználó bármikor kiléphet és később folytathatja a farmolást.

Menü

A felhasználót a program elindításakor egy menü fogadja, amelyben a következő opciók közül tud választani:

1. új játék
2. játék betöltése
3. kilépés

Az „új játék” opció

Ebben az opcióban a játékos elkezdheti a saját gazdálkodását, először a megjelenő ablakban beírja a nevét majd a „Save” gombra kattintva elmenti. Ha sikeres volt, akkor megjelenik a játékablak és kezdetben 2 db termékeny földdel rendelkezik (zöldként jelennek meg), illetve 2 növény maggal (Inventoryban található). Ezután már a felhasználón áll, hogyan alakítja a farmját, aminek állapotát majd el tudja majd menteni.

A „játék betöltése” opció

Ha már játszott a játékkal, akkor később visszalépve a programba nem kell újra kezdenie. Ebben az opcióban megtalálja a korábbi mentését és be tudja tölteni ezt, így ugyanott folytathatja a kertészkedést. Egyszerre csak egy játékállás van elmentve, mindig a legutóbbi.

„Kilépés”

Kiléphet a felhasználó a programból.

Játék menete

Az új játék opcióban a kezdeti állapotot (2 mag, 2 föld) megkapott játékos elkezdheti a farmját művelni, első sorban érdemes elültetni a magokat. A magok öntözés hatására megnőnek és learathatóak. Ezután a játékos a növényt eladhatja, pénzt szerezhet vele. A pénzből újabb magokat vehet a boltban vagy a földjeit bővítheti. A játékosnak nincs végtelen tárhelye, de eladhatja a magokat a boltban, ha már megtelt a tároló.

Tevékenységek a játékban:

A felhasználó minden műveletet egérrel tud kiválasztani és kattintással véglegesíteni azt.

- új mag vásárlása
- új növény ültetése
- öntözés (ennek hatására megnő)
- learatás
- növény eladása
- új föld vásárlása

Osztrogonácz Dóra
UHI5AX

Játék működése/jelzések értelmezése

A játék ablakban különböző gombok és jelzések vannak ezek magyarázata következik.

A játék ablak legfelső sorában balról jobbra olvasva megjelenik az elején megadott **játékos neve**, majd az aktuális pénz és a „Save and go back to the Main Menu” gomb.

A **pénz** értéke a különböző interakciók során változik, eladás esetén nő az értéke, míg vásárlás esetén csökken.

Save and go back to the Main Menu gomb, avagy **Mentés gomb**: fontos a játék csak akkor kerül elmentésre, ha a játék befejezésekor ezzel visszalépünk.

Bal oldalt található az Inventory és Shop gombok.

Inventory gomb: Ide kattintva egy felugró kisablak jelenik meg, ami mutatja a játékos tárhelyének a tartalmát, megjelenik a mag azonosítója és neve. Alapvetően a tárhely mérete 30.

Shop gomb: kattintás után szintén egy felugró ablak fogadja a játékost, ahol a megjelenő magok közül kattintással vásárolhat, ha olyan magot szeretne megvásárolni, amit már tartalmaz a tárhelye, akkor a játék erre figyelmezteti. A felugró ablak jobb felső sarkában található az eladás gomb, ahol a listában megjelenő azonosítók közül kiválaszthatja a felhasználó az eladni kívánt magot és a „Sell” gombra kattintva eladja. A gomb használata során a játékos pénze és tárhelyének tartalma változik a vásárlás és eladás folyamatoknak megfelelően. A bolt funkció tehát lehetővé teszi, hogy a játékos új magokat vásároljon vagy eladjon.

A játéktér legnagyobb területét a földek foglalják el, ezekhez több funkció is tartozik.

A föld (=Soil) színeinek jelentése:

- **szürke**: a föld nem ültethető, meg kell venni először
- **zöld**: a föld már megvásárlásra került, ültethető
- **barna**: egy mag elültetve lett a földbe
- **sárga:** a növény megnőtt, learatható állapotban van

Egy földre kattintva újabb felugró ablak jelenik meg, amelyben az állapottól függően elérhetőek az **interakciós gombok**. (Buy, Sell, Plant, Water, Harvest)

- **szürke**: megvásárlás gomb elérhető (Buy)
- **zöld:** eladható a föld (Sell) vagy ültethető mag, amit a listából kiválaszthat majd a Plant gombra kattintva elültethet (Plant). Ültetés után a mag kikerül a játékos tárhelyéből.
- **barna:** az elültetett növényt meg lehet öntözni (Water)
- **sárga:** learatható (Harvest), learatáskor a learatott növényért pénz kap a játékos

Osztrogonácz Dóra
UHI5AX

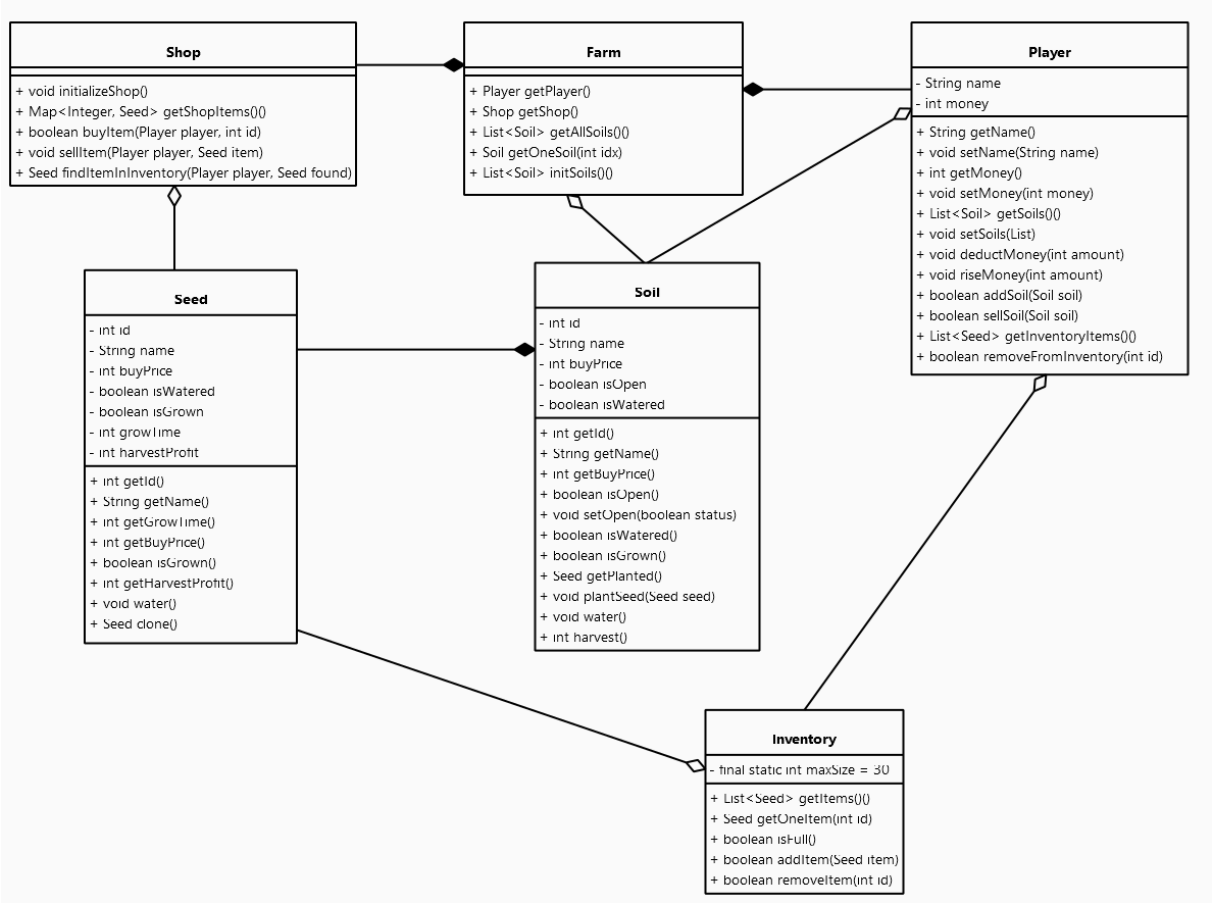
Programozói dokumentáció

Megoldási terv:

A játék megvalósításának alapja a Swing GUI. Objektumorientáltan készítem el a különböző játékelemeket és a hozzátartozó funkciókat.
Játék osztályok: Farm, Játékos, Föld, Mag, Tárhely, Bolt
Megjelenítés: MyFrame, MainMenu, AddPlayerNameFrame, NewGameFrame, Message, ErrorMessage
ezekhez Swing „eszközök”: JFrame, JPanel, JToolBar, JLabel, JButton, JDialog, JComboBox
Mentés/Betöltés: NewGameFrameSaver osztály, LoadGame osztály

Megoldás részletezése

Játék osztályok



1. ábra: UML diagram a Játék osztályokhoz

Farm: Ez az osztály összefogja az összes komponenst és megvalósít egy farmot.
Adattagok: NewGameFrame, Player, Shop, List<Soil> allSoils
A konstruktorában megkülönböztettem az új játék létrehozását és a játék betöltését, mert a játék betöltésekor nem kell már beállítani a kezdő adatokat (2 föld és 2 mag).
Az adattagjai privátak és **Getter metódusokat** is készítettem hozzájuk. Illetve egy földeket inicializáló függvényt is készítettem, ami visszaadja az elkészített összes földet egy ArrayList-ben: **public List<Soil> initSoils();**

Osztrogonácz Dóra
UHI5AX

Player: egy játékos adatait tartalmazza. Konstruktorában beállítja a kapott nevet és a többi adattagot.
Adattagok: String name, int money (kezdőérték 500)**, List<Soil> soils** (amiket már megvett a játékos), **Inventory** (saját tárhely)
Gettereket és Settereket is készítettem a privát adattagokhoz.
Pénzmódosító függvények:

- **public void deductMoney(int amount):** csökkenti a pénzt a megadott mennyiséggel
- **public void riseMoney(int amount):** növeli a pénzt

Birtokolt földeket módosító függvények:

- **public boolean addSoil(Soil soil):** hozzáad egy földet a játékoshoz, ha van elég pénze, csökkenti a pénzt
- **public boolean sellSoil(Soil soil):** kivesz egy földet a játékostól és növeli a pénzt

Tárhely módosytó függvények:

- **public boolean addToInventory(Seed item):** új elem hozzáadása a tárhelyhez, ha van hely és még nincs benne tárhelyben.
- **public boolean removeFromInventory(int id):** kapott azonosítójú elem eldobása a tárhelyből, ha a tárhelyben van ilyen elem és pénz növelése

Soil: Ez az osztály valósítja meg a földhöz tartozó interakciókat, így itt a megjelenítéshez tartozó elemek is már megjelennek.
Adattagok: felugró ablak és annak a tartalma (szöveg, gombok, JComboBox), **Farm, Player, int id, String name, int buyPrice, boolean isOpen** (ha igaz, akkor a játékos már megvette a földet), **boolean isWatered** (meg lett-e öntözve), **Seed plantedSeed** (földbe elültetett aktuális mag)
A konstruktorba inicializálódnak a felugró ablak konténerei és az interakciós gombok. A föld gomb színe alapértelmezetten szürke, mert még nincs megvásárolva és meghívódik a JComboBox tartalmát frissítő függvény (**refreshTheContentOfSeeds()**). Az interakciós gombok elérhetősége beállítódik, csak a Buy gomb elérhető. Meghívódik az **actions()** függvény amiben létrejön a különböző megjelenítések formázása (**Customize()**) és a gombokhoz ActionListenerek rendelése. További adattagok inicializálása.
public void actions():

- **soilButton.addActionListener(new ActionListener()):** A föld gomb ActionListenerje, a felugró ablak láthatóvá válik és a JComboBox tartalma frissül.
- **buySoil.addActionListener(new ActionListener()):** Az adott földhöz tartozó felugró ablakban a megvásárlás gombhoz tartozó reakció. A megvásárlás elérhető, ha a földet még nem birtokolja a játékos. Ha sikeresen hozzá lett adva, akkor az interakciós gombok közül az eladás (Sell) és az ültetés (Plant) elérhetővé válik, illetve a vásárlás (Buy) gomb nem megnyomhatóvá. Továbbá a fő SoilButton háttere zöld lesz és a játékos pénzének érteke frissül a vásárlás után. A sikeres vásárlást még egy megerősítő felugró ablak jelzi a felhasználó felé.
  Ha a vásárlás sikertelen volt, a gombok elérhetősége marad (csak Buy elérhető) és egy hiba jelző ablak ugrik fel.
- **sellSoil.addActionListener(new ActionListener()):** A földhöz tartozó felugró ablakban az eladás gombhoz tartozó reakció. A játékos földEladása metódusának meghívása. Ha sikeres volt az eladása, akkor a gombok elérhetősége változik, csak a buy elérhető és a föld gomb színe szürke lesz. A játékos pénzének értéke frissül és egy megerősítő ablak jelenik meg a felhasználó számára.
  Ha sikertelen volt a vásárlás, akkor a gombok státusza marad és egy hibaüzenetet tartalmazó ablak jelenik meg.

Osztrogonácz Dóra
UHI5AX

- **plantSeed.addActionListener(new ActionListener()):** Ha a választható elemek listája üres, akkor tud ültetni a játékos. Egyébként pedig a listából kiválasztott elemet a JComboBox-nak a getSelectedItem metódusával érem el. Majd ezt a játékos Tárhelyében megkeresem, eltárolom a földnek a plantedSeed adattagjában, majd törlöm a tárhelyből. Ezután frissítem a JComboBox tartalmát, a NewGameFrame-ben az Inventory tartalmát és a Shop-ban a JComboBoxban tárolt eladható növények tartalmát. Ez utóbbi esetben statikussá tettem őket, hogy a Soil osztályból is elérhessem ezeket a refresh/repaint függvényeket. Majd a gombok státusza változott, a sell és a water gomb elérhető, többi nem. Plusz megerősítő üzenet a felhasználónak a sikeres ültetésről és a földgomb színének megváltoztatása történt még.
- **watering.addActionListener(new ActionListener()):** Növény öntözése. A föld által eltárolt aktuálisan ültetett növény (plantedSeed) water metódusát hívja meg. Az isWatered állapota true lesz. A harvest (aratás) gomb státusza változik, ha az elültetett növény megnőtt (isGrown == true). Megnőtt növény esetén a föld színe sárga lesz, majd a felhasználót tájékoztatja a sikeres öntözésről.
- **harvestPlant.addActionListener(new ActionListener()):** Megnőtt növény learatása, profit hozzáadása a játékos pénzéhez. Az elültetett növény null értéket kap. A föld gomb háttere újra zöld. Gombok státusza ehhez megfelelően változik.

public void refreshTheContentOfSeeds(): frissíti a JComboBox chooseSeed tartalmát
public void init(): inicializálja a kezdő elérhető földet
Customize(): beállítja és személyre szabja a megjelenítést
Getterek és Setterek az adattagokhoz.
public void water() és public int harvest() metódusok valósítják meg a növény öntözését és aratását.
Seed: Ez az osztály valósítja meg a maghoz tartozó funkciókat.
Adattagok: int id, String name, int buyPrice, boolean isWatered, boolean isGrown,int growTime, int harvestProfit;
Getterek és Setterek az adattagokhoz. **public void water()** metódus, csökkenti a growTime-ot és ha az 0, akkor az isGrown = true értéket kap. Emellett a **clone()** metódus lemásolja a magot
Inventory:
Adattagok: maxsize = 30 és a birtokolt magokról egy ArrayList (List<Seed> items)
Metódusok: getterek és setterek, isFull() ellenőrzi tele van-e a tárhely
public boolean addItem(Seed item): ha nincs tele, akkor megkeresi tartalmazza-e már ezt a magot, ha nem akkor hozzáadja, visszaadja a hozzáadás sikerességét
public boolean removeItem(int id): ha megtalálja az elemet kitörli, visszaadja a sikerességét
Shop:
Adattagok: HashMap<Integer, Seed> shopItems
HashMapben tárolja az elemeket, az Integer a növény azonosítója, könnyű a keresés és hozzáadás a tárhelyhez vásárláskor, illetve törlés a tárhelyből eladáskor
public boolean buyItem(Player player, int id): új mag vásárlása, hozzáadódik a tárhelyhez, pénz módosul
public void sellItem(Player player, Seed item): választott mag eladása
public Seed findItemInInventory(Player player, Seed found): elem megtalálása a tárhelyben

Megjelenítés

MyFrame: a főablak, JFrame-ből származik le, tartalmaz egy MainMenu JPanel
MainMenu: JPanel, tartalmazza a kisebb elemeket, a megjelenő szöveget és a gombokat. Itt történik a gombokhoz az reakciók (actionlistenerek) hozzáadása. Új játék kezdésekor megjelenik a játékos nevét inicializáló ablak. Játék létrejön egy NewGameFrame, ahol a játékos neve „loadgame”, ezt kezelem később a NewGameFrame konstruktorában.

Osztrogonácz Dóra
UHI5AX

AddPlayerNameFrame: Játékos nevét megadó felugró ablak, a Save gombhoz hozzárendel egy ActionListenert() ami létrehozza a NewGameFramet és láthatóvá teszi.
NewGameFrame: megvalósítja a teljes játékablakot és a hozzátartozó funkciókat. Szövegek, Gombok, Felugró ablakok és egyéb megjelenítések.
Message: Felhasználó számára egy megerősítő ablak.
ErrorMessage: Felhasználó számára jelez hibát.

Mentés/Betöltés

NewGameFrameSaver:
public static void saveState(NewGameFrame frame, String fileName): a paraméterben kapott frame tartalmát kimenti a kapott fileName nevű fájlba, ebben a programban a saved.txt-be menti az adatokat a NewGameFrame-ben a Save and go back to mainMenu gombját megnyomva. A mentést egy bufferedWriter valósítja meg és azonosításként az adatok elé kiírja a típusukat.
például: a játékos neve: antal, pénz: 300, ez esetben a tárhelye üres és 3 birtokolt földje van
PlayerName: antal
PlayerMoney: 300
Soils: 0,BasicSoil,200
Soils: 1,BasicSoil,200
Soils: 2,BasicSoil,200
LoadGame:
public static void loadState(NewGameFrame frame, String fileName): a paraméterben kapott frame tartalmát fogja módosítani és a kapott fileName nevű fájlból olvas be a BufferedReader. If-ekkel különbözteti meg a kezdő sorokat és a beolvasás módját.
Minden adat beolvasása után a frame.farm adatai változnak. Ez a metódus a NewGameFrame konstruktorában hívódik meg, ha a játékos neve „loadgame”, ezt a Main Menu LoadGame gombja teszi lehetővé. Mindig a legutoljára játszott és elmentett játék adatai töltődnek be.
Legfontosabb adatok amik meghatározzák a játékot, tehát a

- játékos neve, pénze, tárhelyének tartalma és a birtokolt földek