

COLLEGE N COLLEAGUES

Készítette:

Balázs Ede

Bíró Apor

Lőrinczi Mátyás

Számítástechnika III. év

Vezető tanár: Szántó Zoltán

1. Bevezető.....	3
2. Projekt célja.....	3
3. Követelmények.....	4
3.1 Felhasználói követelmények.....	4
3.2 Rendszerkövetelmények.....	6
3.2.1 Funkcionális követelmények.....	6
3.2.2 Nem-funkcionális követelmények.....	7
4. Tervezés.....	7
4.1 Architektúra.....	7
4.2 Modulok leírása.....	9
4.2.1 Fontosabb őssosztályok.....	9
4.3 Managelés.....	10
5. Alkalmazás működése.....	11
5.1 Főmenü.....	11
5.2 New Game.....	11
5.3 Load Game.....	12
5.4 Quit.....	12
5.5 Exploration.....	13
5.6 Fight.....	15
6. Összegzés.....	16
6.1 További fejlesztési lehetőségek.....	16

1. Bevezető

Miután hárman eldöntöttük, hogy egy csapatot fogunk alkotni sokat agyaltunk azon, hogy milyen témájú alkalmazást készítsünk. Mindhárman az Informatika más-más területein voltunk tapasztaltabbak. Mátyás beágyazott informatikával foglalkozott inkább, Apor webfejlesztéssel, Ede pedig videójátékfejlesztéssel. Végül a játékfejlesztés mellett tettük le a voksunkat, mivel átlagot véve ez tetszett meg a legjobban mindannyiunknak.

Az appról röviden:

Az appot amit fejlesztettünk C++ nyelvben, a Simple DirectMedia Layer(SDL) ingyenes multiplatformos, nyílt forráskódú multimédiás szoftverkönyvtár használatával írtunk, adatbázis használata nélkül. Mivel ezt a nyelvet már líceum óta tanuljuk, ezért nem akadtunk fel igazán egyetlen problémán sem, a kódolás elég gördülékenyen ment. Mindhárman más-más, a kedvenc fejlesztői környezetekben dolgoztunk, Apor CLion-ban, Mátyás Visual Studio-ban, Ede pedig Linux Vim-ben. Mivel az SDL multiplatformos, ezért az app ugyanúgy futtatható Windows-on, Linux-on és macOS-en is.

2. Projekt célja

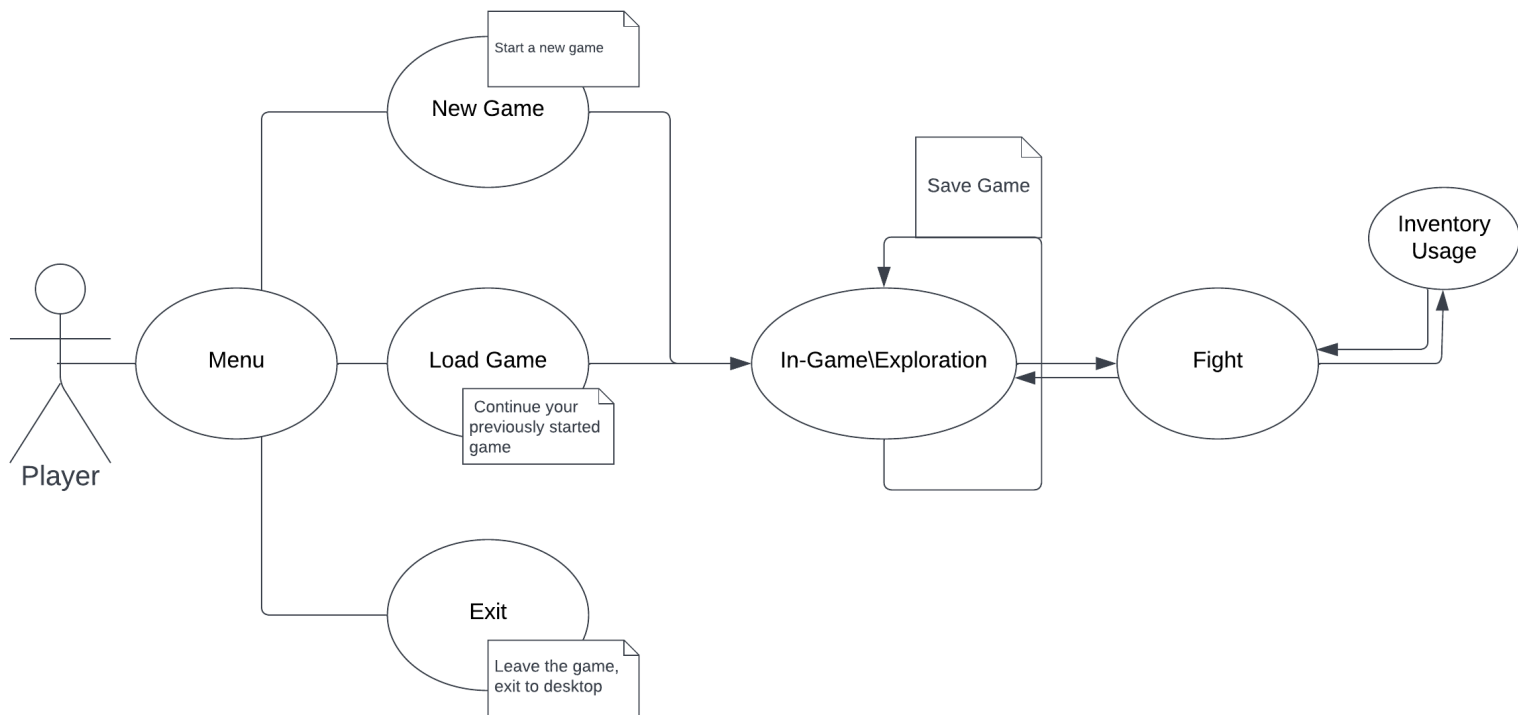
A projekt célja egy 2D-s felülnézetes játék, amely alapvetően két játékmóddal, felfedezéssel(Exploration) és harccal(Fighting) rendelkezik, amelyek összefüggésben vannak egymással. A felhasználónak a felfedezés során a célja az, hogy karakterével a pályán(Map) mozogva tárgyakat szerezzen be, amelyek eltárolódnak az leltárban(Inventory). A begyűjthető tárgyak lehetnek fegyverek(Weapon, Throwable), páncélok(Armor), varázslatok(Spell) és főzetek(Potion). A harc során pedig az a célja a felhasználónak, hogy a megszerzett tárgyakat felhasználva, párbajban legyőzze ellenfelét azáltal, hogy “megöli”, vagyis segítségükkel 0-ig sebezze az életét(HP). A játék célja az, hogy legyőzzük a végső ellenfelet, a “Boss”-t, ekkor érünk a játék végére.

3. Követelmények

3.1 Felhasználói követelmények

- A játékhoz a felhasználónak szüksége van egy személyi számítógépre (Windows, Linux, macOS), illetve egy egérre és egy billentyűzetre.
- A forráskód letöltése után a futtatáshoz nem szükséges folyamatos internet kapcsolat, ugyanis minden adat lokálisan lesz eltárolva a lemezen.
- A felhasználó az app elindítása után a főmenüben találja magát, ahol az egérrel navigálhat a különböző gombok között, amelyek “New Game”, “Load Game”, “Leave Game”
- A felhasználó a “New Game”-re klikkelve a “Character Creation” UI-ban találja magát, ahol szintén az egérrel választja ki a karaktert amellyel játszani szeretne
- A felhasználó felfedezés játékmódban a W-A-S-D gombokkal mozoghat karakterével, illetve az egérrel nyomhatja le a “Save Game” gombot
- A felhasználó harc játékmódban, illetve az inventory-ban a csak az egérrel navigál

A 3.1.1-es ábrán a Use-Case diagram található. Egyszerűen bemutatja a felhasználó szemszögéből a lehetséges funkciókat



3.1.1 Use-Case diagram

3.2 Rendszerkövetelmények

3.2.1 Funkcionális követelmények

- Főmenü: Az az oldal, amelyik az app futtatása után alaphoz megjelenik a felhasználónak a képernyőjén. Lényegében egy UI, amely három gombbal rendelkezik: New Game(Új játék), Load Game(Játék folytatása), Quit(Játék elhagyása).
 - New Game: A “Character Creation” oldalra továbbít
 - Load Game: Egy külső fájlt beolvassza betölti a legutóbbi elmentett játékot, hogy ezt a felhasználó folytathassa
 - Quit: Leállítja az app futását, amely a játék elhagyásához vezet
- Karakter készítés(Character Creation) felület: Szintén egy UI, ahol két, egy női vagy egy férfi karakter közül választhatjuk ki, hogy melyikkel szeretnénk játszani. A karakterek kiválasztását egy gomb jelzi, melyek lenyomása után az app az felfedezés játékmódba továbbítja a felhasználót, így kezdődik el a játék.
- Felfedezés(Exploration) játékmód: Az egyik “fele” úgy mond a játéknak, a kiválasztott karakterünkkel mozoghatunk a “Map”-on(pályán), különböző tárgyakat gyűjtve, amelyeket a harc játékmódban használhatunk. E játékmód során a játékos akármelyik véletlenszerű pillanatban rátalálhat egy ellenfélre, ekkor átkerül a másik játékmódba.
- Harc(Fighting) játékmód: A felfedező játékmódtól különböző felület. Itt a már az összegyűjtött, az “Inventory”-ban(leltárban) eltárolt tárgyakat használhatja fel a játékos, hogy párbajban legyőzze ellenfelét, azáltal, hogy 0 “HP”-ig(életig) sebzi őt, megelőzve azt, hogy az ellenfél győzze le őt hamarabb. Ha legyőzte ellenfelét, a játékos visszakerül a felfedezés játékmódba. Viszont ha a játékos elveszítette a párbajt, akkor a főmenüben találja magát, ahol betöltheti a legutoljára elmentett játékát, hogy újrapróbálhassa.
- Leltár(Inventory): Szintén egy UI, ahova harc játékmódba tud belépni a felhasználó. Ugyebár ezen az UI-n keresztül vannak megjelenítve az eltárolt tárgyak. Ezeket a tárgyakat a játékos ezen az

oldalon felhasználhatja, törölheti vagy kicserélheti. Amint pedig végzett visszanavigálhat oda ahol abbahagyta a játékot.

3.2.2 Nem-funkcionális követelmények

Az alkalmazást legalább a következő rendszertulajdonságokkal lehet futtatni:

- Operációs rendszer:
 - Windows XP vagy ennél újabb
 - Linux 2.6 vagy ennél újabb
 - MacOS X 10.5 vagy ennél újabb
- C++98 (ISO/IEC 14882:1998) vagy ennél újabb kompilátor
- Simple DirectMedia Layer(SDL) 2.0.0 vagy ennél újabb szoftverkönyvtár
- 700 MHz órajel frekvenciás processzor
- 10 MB szabad tárhely
- 50 MB RAM

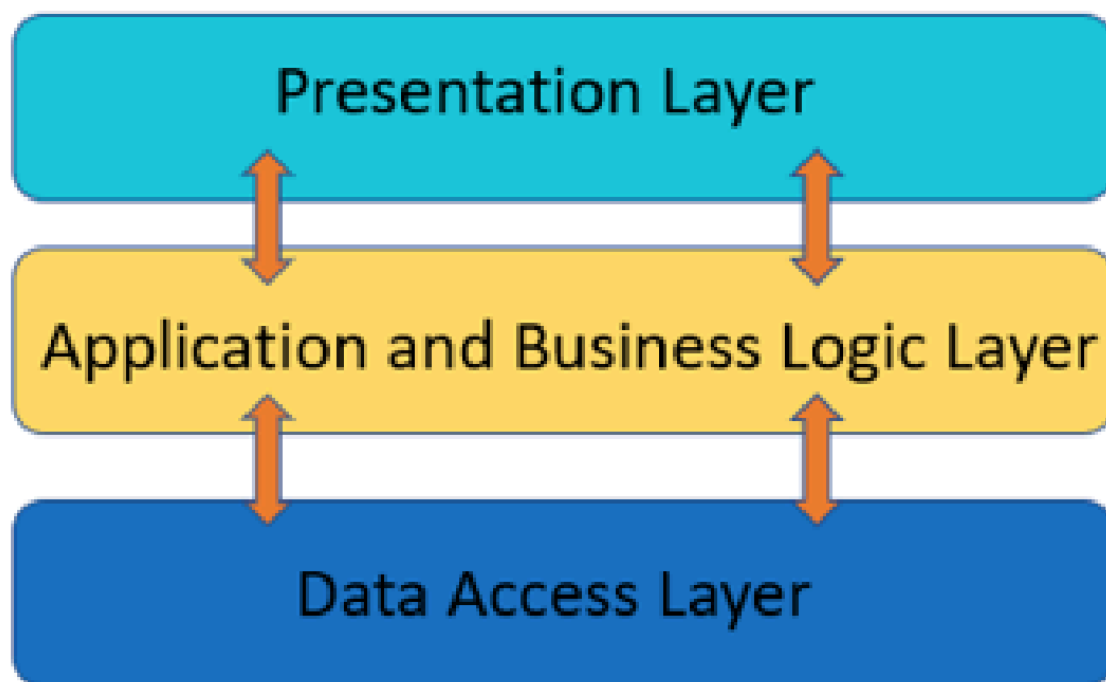
4. Tervezés

4.1 Architektúra

Miután eldöntöttük, hogy játékot szeretnénk készíteni, eljött az ideje azon is gondolkozni, hogy a feladatokat miként osszuk fel. Végül a háromrétegű szoftver architektúra(Three Layered Architecture) alapján kezdtünk neki a játék lefejlesztésének. Azért eme architektúra mellett döntöttünk, mivel hogy, ahogy a nevéből is adódik, három rétegből építi fel a programot. Egyrészt ez azért volt nekünk kedvező, mert mivel hogy hárman vagyunk, így pont rétegenként el tudtuk osztani három felé a feladatokat. Másrészt ezt az architektúrát használva, bizonyos feladatokat a többi réteg haladásától függetlenül tudtunk implementálni.

A szoftver architektúra három rétege:

- Adat réteg(Data Access Layer): Az adatokat, az adatok szerkezetét definiáló réteg. Mivel, hogy C++-ban dolgoztunk, ide sorolhatjuk az osztályokat.
- Működési logika réteg(Business Logic Layer): A program működését, háttérben mozgó függvényeit definiáló réteg. Ez a réteg határozza meg, hogy a háttérben logikailag mi történik a játékban.
- Vizuális/Bemutató réteg(Presentation Layer): A játék felhasználó szemmel látható részét, az úgynevezett “front-end”-et definiáló réteg. Ide tartoznak a játék textúrái, többek között például a map és a karakter kinézetei is.



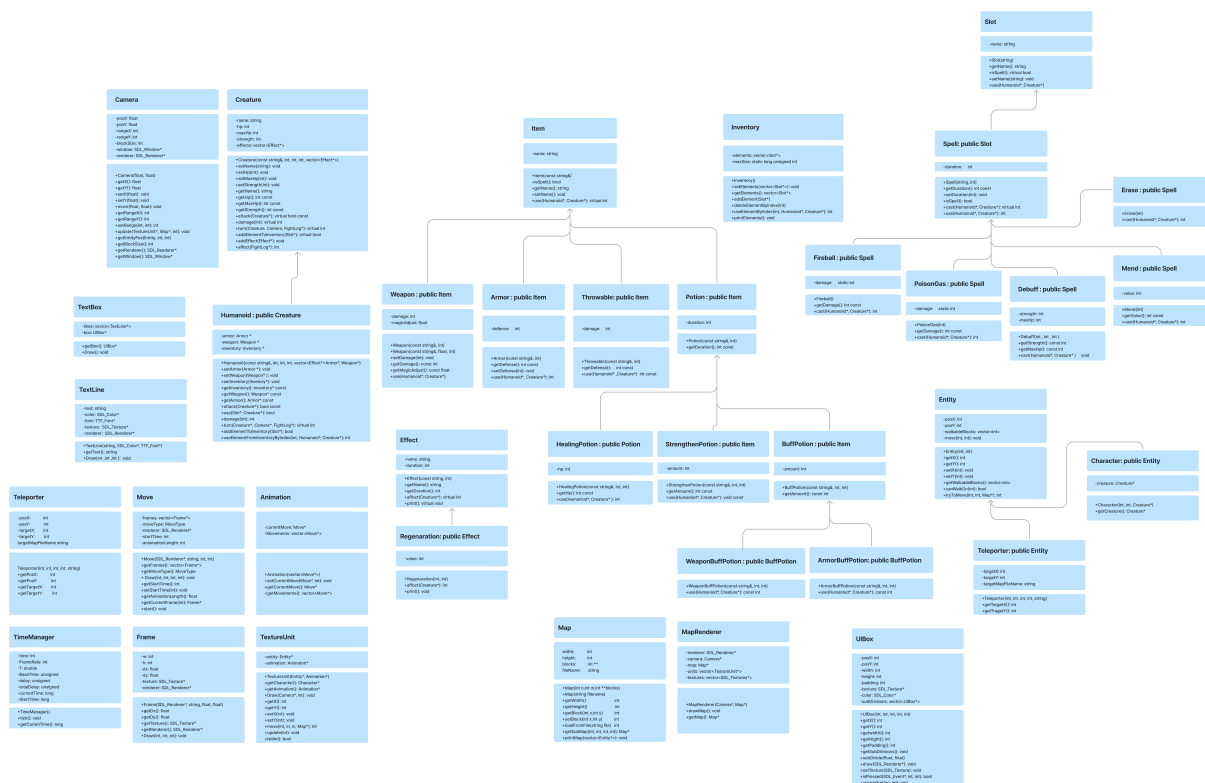
4.1.1 Háromrétegű szoftver architektúra

4.2 Modulok leírása

4.2.1 Fontosabb ősosztályok:

- Camera: Magát a játék kirajzolását kezelő osztály. Ez felel például a map azon részének a kirajzolásáért, ahol a karakterünk található.
- Creature: A karakter osztálynak az ősosztálya. Minden játékon belüli tevékenységet(pld. támadás) végző objektum(pld. ellenfél a harc játékmódban) ebbe az osztályba sorolható.
- Item: A spelleken kívül minden használható tárgy ide tartozik.
- Spell: Varázslatok, amelyeket az ellenfélen alkalmazunk.
- Slot: Ennek az osztálynak a példányaiból épül fel az inventory.

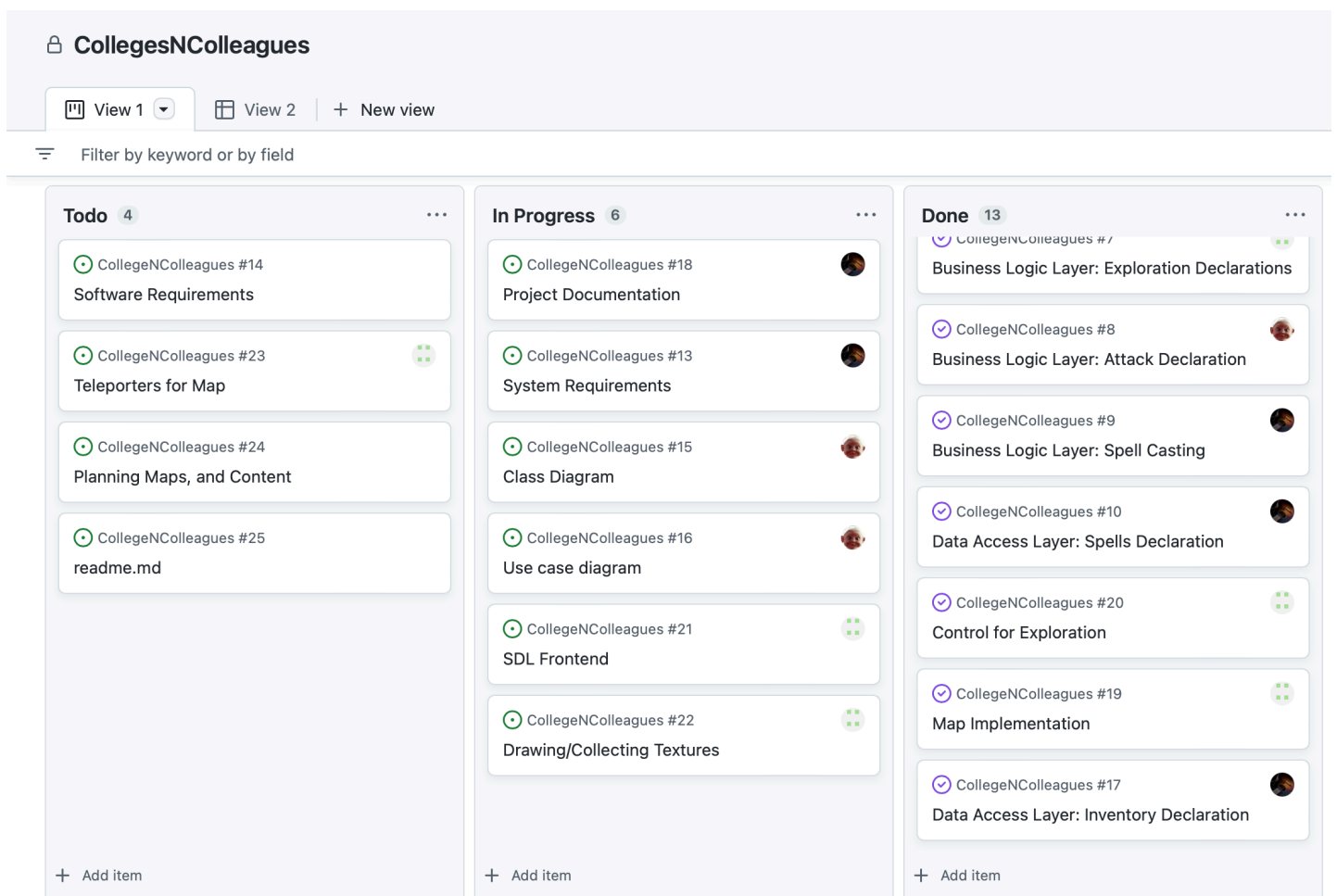
A 4.2.1-es ábrán az alkalmazás Osztálydiagramja látható. A diagram szemlélteti az objektumorientált programozási paradigma használatával fejlesztett Data Access Layer osztályait, tehát az adattagokat és a hozzájuk tartozó metódusokat.



4.2.1 Osztálydiagram

4.3 Managelés

Létrehoztuk ezután a Kanban Board-ot a GitHub projekten belül, amelyről egy képernyőkép a fejlesztés idejéből a 4.3-as ábrán látható. Itt átláthatóan tudtuk vezetni a feladatokat, a teendőket, a funkciókat, amelyek készen vannak vagy amelyek éppen implementációra várnak.

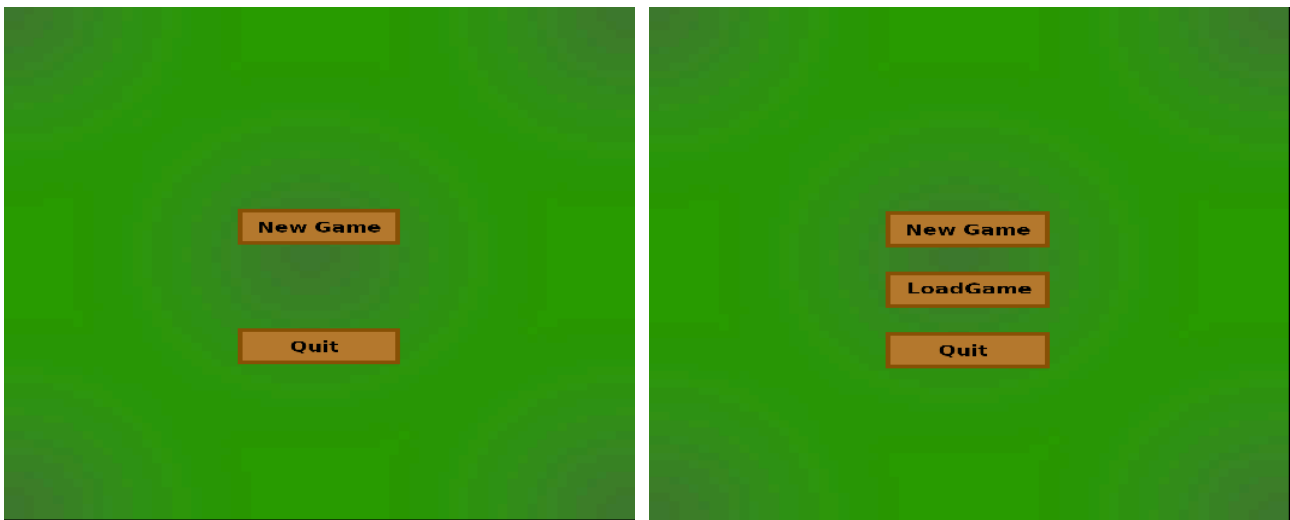


4.3.1 Kanban Board GitHub-on

5. Alkalmazás működése

5.1 Főmenü

A játék megnyitásakor a játékos egy Főmenüben találja magát. Ezen az képernyőn találunk egy "New Game", és egy "Quit" feliratú gombot. Ha már létezik elmentett játékállás, akkor az említett gombok mellett megjelenik egy harmadik gomb is, "Load Game" felirattal.

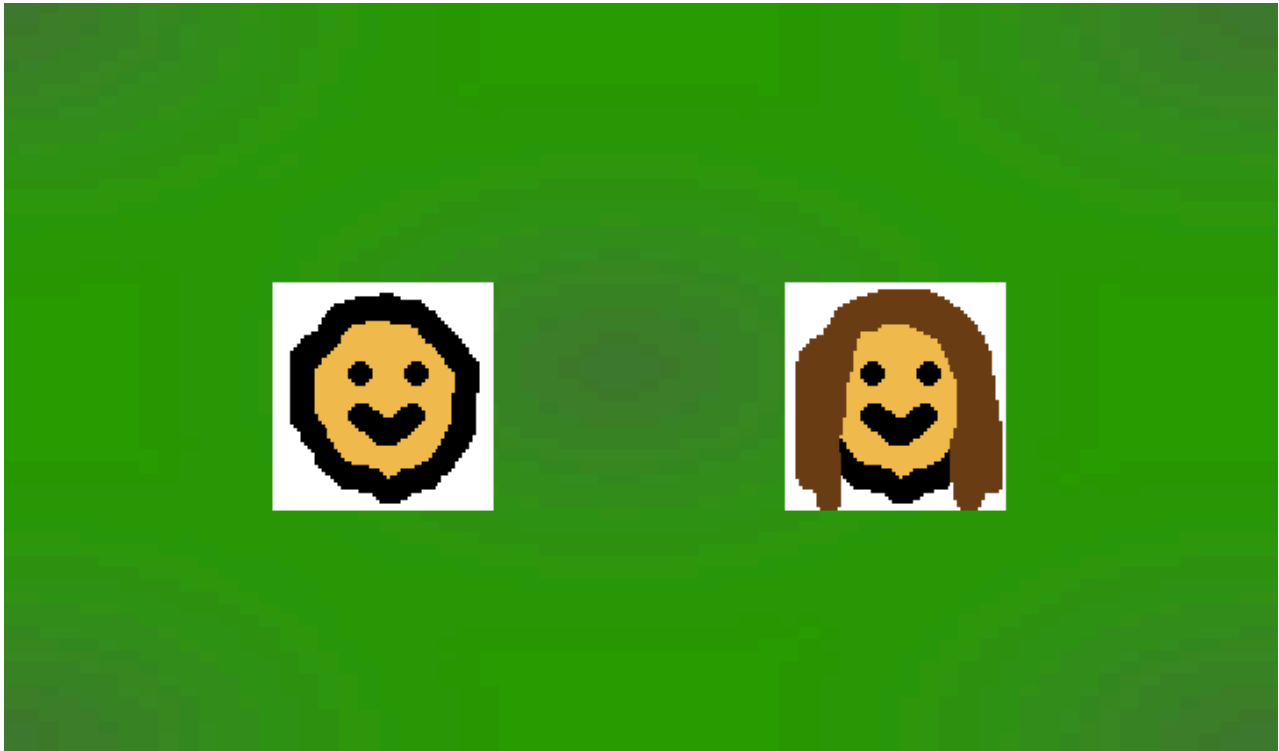


5.2 New Game

A "New Game" gomb megnyomásakor megjelenik a Character Selection Screen (Karakterválasztó képernyő), amiben a játékos választhat több karakter közül. Ez kizárólag esztétikai döntés, nincsen hatása a játékmenetre. Miután a játékos kiválasztotta a karakterét, a játék létrehozza annak a háttérbeli osztályait.

- Character: Az osztály, ami a játékos pozícióját és mozgását kezeli felfedezés közben.
- TextureUnit: A játékos kirajzolását, textúráját és animációit kezeli felfedezés közben.
- Humanoid osztály: A Humanoid osztály felelőssége eltárolni és kezelni azokat az adatokat, amelyeknek harc közben van jelentőségük, mint például az Életerőpont, a sebzés, a leltár és a fegyverzet.
- Inventory osztály: Az Inventory osztály a játékos harc közben használható tárgyait és képességeit tartalmazza. Az osztály egy "Slot" osztályokból álló lista, amikhez szabad hozzáférés van. Egy Slot lehet egy Item, vagy egy Spell. A Spell-eket többször is lehet használni, míg

az Item-ek használat után eltűnnek a leltárból, elhasználódnak, vagy áthelyeződnek a fegyver vagy fegyverzet helyére.



5.3 Load Game

A "Load Game" gomb megnyomásakor a karakter adatait nem automatikusan generálja le, hanem egy korábbi játék elmentésekor kreált mentésfájlból szerzett adatokból generálja le. Ezekkel az adatokkal meghatározott karakter játék közben megkülönböztethetetlen egy új játék által generált karaktertől.

5.4 Quit

A "Quit" gomb megnyomásával a játékos kiléphet a játékból. Ennek a gombnak a használata mellett alkalmazható az Operációs rendszerfüggő kilépési módszer. (Például a sarokban az X megnyomása, ALT+F4, SUPER+C, stb.) A rendszerfüggő kilépés a játék bármelyik pontjában használható, viszont kilépéskor a játék el nem mentett állása elveszik.

5.5 Exploration

Exploration közben a játékos egy 2 dimenziós felülnézetes négyzetrácsos pályán találja magát. A játékosnak egész pozíciója van, mezőről mezőre tud lépni, és nem lehet félig egy mezőn és félig egy másikon. A mezők több félék lehetnek. 2 Kategória szerint lehet elkülöníteni őket.



5.5.1 Járhatóság

Egy mező lehet járható és járhatatlan. A játékos a járható mezőkön keresztül tud közlekedni. A járhatatlan mezők struktúrát adnak a pályának, és akadályozhatják a játékos haladását.



5.5.2 Biztonság

Egy mező lehet biztonságos és rizikós(például a bozót). A játékos rizikós mezőre lépve találkozhat ellenfelekkel, akik elkezdenek egy harcot (fight).



5.5.3 Save Game

Exploration közben a bal felső sarokban megjelenik egy "Save Game" - feliratú Gomb, aminek megnyomásával a játékos elmentheti a játékállását.



5.6 Fight

A játék harcrendszere körökre osztott. Egyszer az egyik fél végezhet akciót, majd a másik. A játékosnak egy körben lehetősége van támadni, vagy a leltárából használni egy Item-et, vagy egy Spell-t. A harc akkor ér véget, ha az egyik fél életerőpontja eléri a 0-át. Ha a játékos életerőpontja éri el a 0-át, a játékos veszít.



A harc egy ablakban játszódik le. Az ablak felső részében vannak a játékos lehetőségei (Támadás és Eszközhasználat), az alsó részében pedig a harcnapló(Fight Log)

A Harcnapló egy várakozási sorban eltárolt üzenetsor, amiben kronologikusan követhetőek az utolsó néhány esemény a harcból. Ilyen események a felek lépései és a körönként megjelenő időszakos hatások (mérgek, regeneráció, stb.)

6. Összegzés

6.1 További fejlesztési lehetőségek

A projekt az Objektum Orientáltságának köszönhetően bővíthető. A tervezés és fejlesztés alatt is több ötlet volt, amikkel ki lehet bővíteni a játékot.

Ötletek, amik nem kerültek be a projektbe bonyolultság és munkaidő szempontjából növekvő sorrendbe rendezve.

- Több pálya közti utazás
- Különböző területeken különböző ellenfelek megjelenése.
- Szintlépési rendszer
- Harc utáni véletlenszerű tárgytalálat (MobDrop)
- Pénz és vásárlás
- Nem játszható karakterekkel történő interakció

Az implementálatlan lehetőségek mellett a játék fejleszthető lenne textúrák, új tárgyak, varázslatok és pályafájlok készítésével. Ezek könnyen beépíthetőek, mivel a játékban egyes mechanikákhoz nem készült példa. Erre jó példa a Magic Adjustment, ami a fegyverek tulajdonsága, és meghatározza, hogy mennyit erősít egy fegyver egy adott képesség használatán.