

COLLEGE N COLLEAGUES

Készítette:

Balázs Ede

Bíró Apor

Lőrinczi Mátyás

Számítástechnika III. év

Vezető tanár: Szántó Zoltán

Tartalomjegyzék

1. Bevezető.....	3
2. Projekt célja.....	3
3. Követelmények.....	4
3.1 Felhasználói követelmények.....	4
3.2 Rendszerkövetelmények.....	5
3.2.1 Funkcionális követelmények.....	5
3.2.2 Nem-funkcionális követelmények.....	6
4. Tervezés.....	6

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 átlagban 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 segítségével 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), Leave Game(Játék elhagyása).
 - New Game: A “Character Creation” oldalra továbbít
 - Load Game: Egy külső fájlt beolvasva betölti a legutóbbi elmentett játékot, hogy ezt a felhasználó folytathassa

- Leave Game: 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” úgymond 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

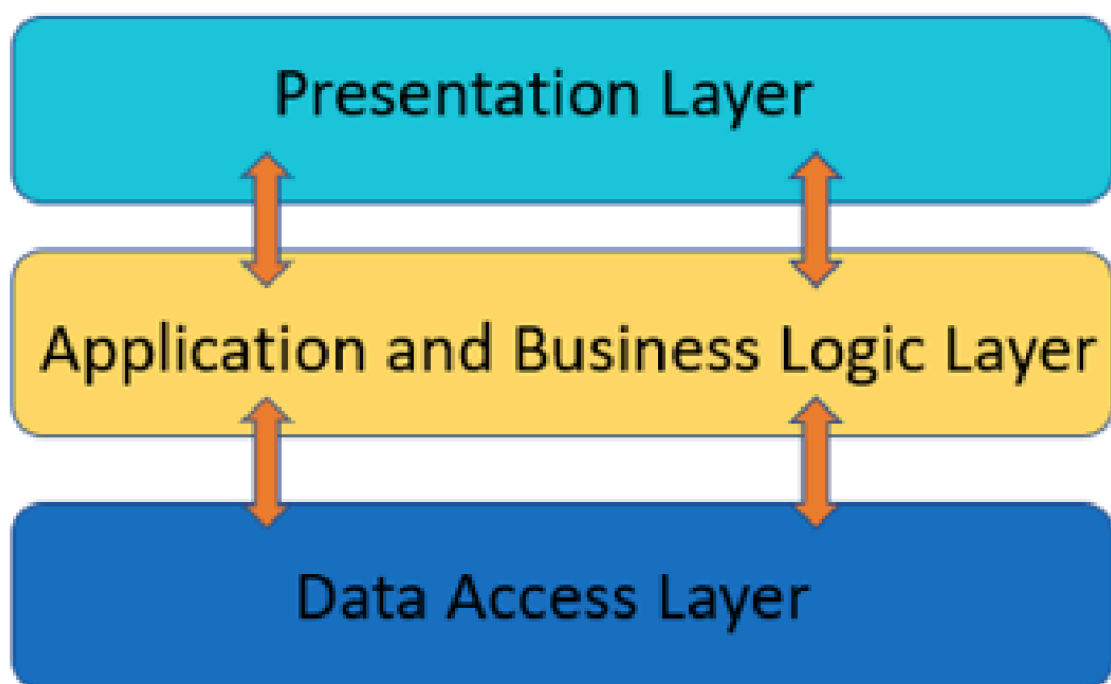
- MacOS X 10.5 vagy ennél újabb
- C++98 (ISO/IEC 14882:1998) vagy ennél újabb kompilátor
- Simple DirectMedia Layer(SDL) 2.0.0 vagy ennél újabb szoftverkönyvtár
- 700 MHz órajel frekvenciás processzor
- 10 MB szabad tárhely
- 50 MB RAM

4. Tervezés

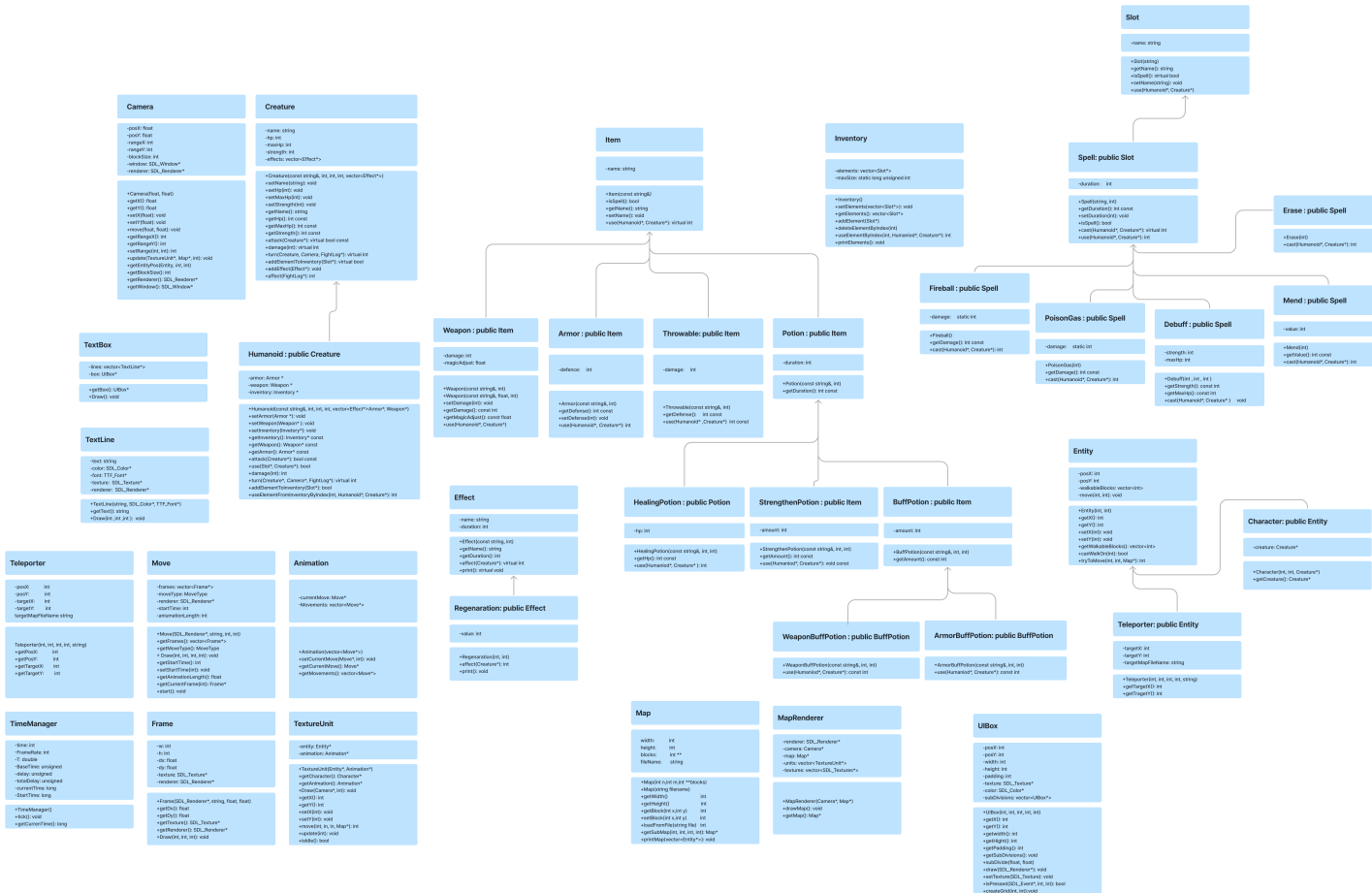
Miután eldöntöttük, hogy játékot szeretnénk készíteni, eljött az ideje azon is gondolkodni, hogy a feladatokat miként osszuk fel. Végül a háromrétegű szoftver architektúra(Three Layered Architecture) alapján kezdtünk neki a játék lefejlesztésének. Egyrészt azért eme architektúra mellett döntöttünk, mivel hogy, ahogy a nevéből is adódik, három rétegből építi fel a programot. Ez azért volt nekünk kedvező, mert mivel hogy hárman vagyunk, így pont rétegenként el tudtuk osztani három felé a feladatokat.

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.
- Vizuális/Bemutató réteg(Presentation Layer): A játék felhasználó szemmel látható részét, az úgymond “front-end”-et definiáló réteg. Ide tartoznak a játék textúrái, többek között például a map és a karakter kinézetei is.

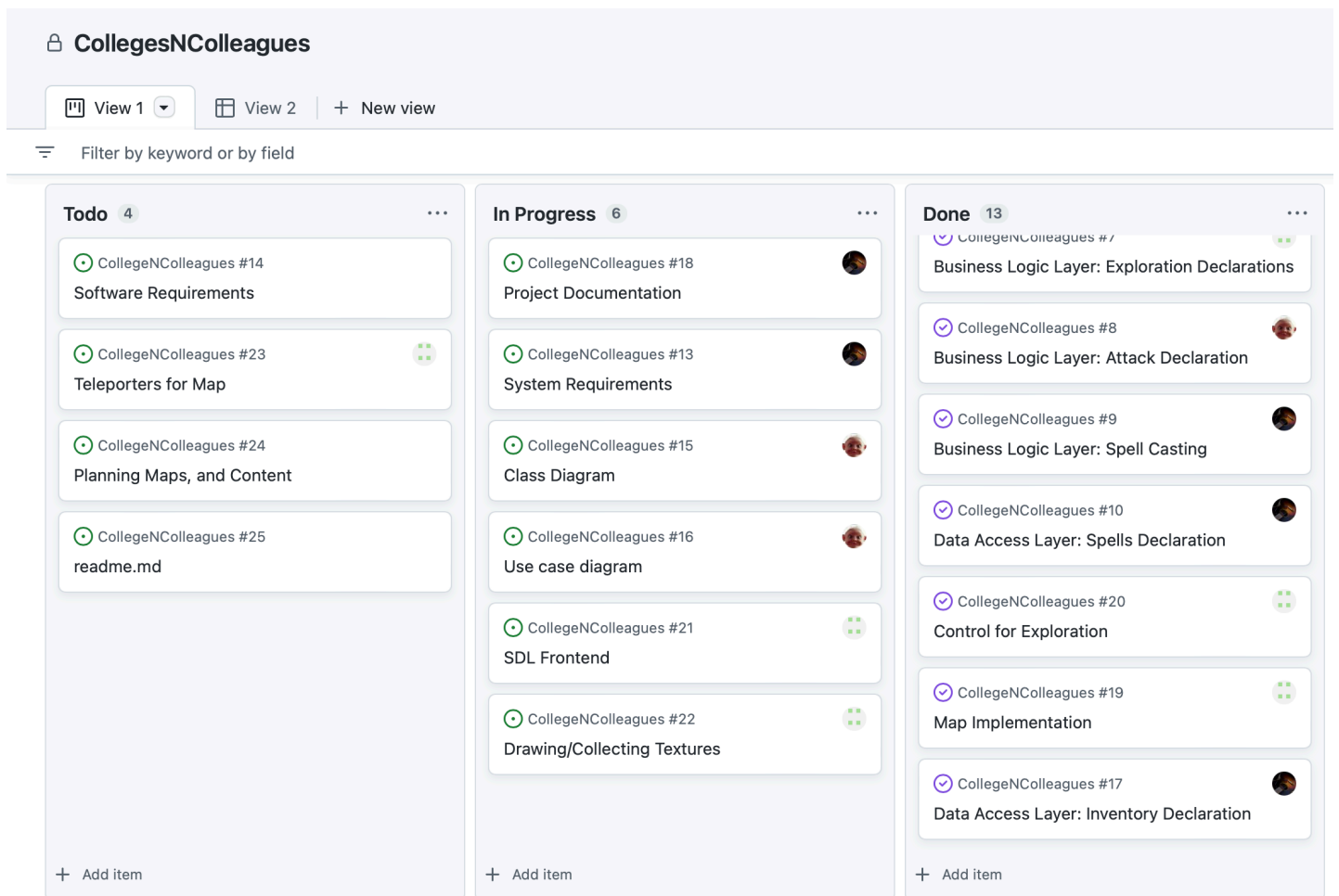


4.1 Háromrétegű szoftver architektúra



4.2 Osztálydiagram

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



4.3 Kanban Board GitHub-on

5. Alkalmazás működése