Tartalomjegyzék

[1. Bevezetés, a probléma rövid ismertetése 2](#_Toc100267543)

[2. Felhasználói dokumentáció 3](#_Toc100267544)

[2.1 A program általános specifikációja 3](#_Toc100267545)

[2.2 Rendszerkövetelmények 4](#_Toc100267546)

[2.2.1 Hardver, szoftver követelmények 4](#_Toc100267547)

[2.3 A program telepítése 6](#_Toc100267548)

[2.4 A program használatának részletes leírása 10](#_Toc100267549)

[2.4.1 Főmenü felépítése 10](#_Toc100267550)

[2.4.2 Beállítások menü felépítése 10](#_Toc100267551)

[2.4.3 Játék megállítása 11](#_Toc100267552)

[2.4.4 Karakter menü 13](#_Toc100267553)

[2.4.5 Irányítás, vizuális elemek, játékmechanikák 13](#_Toc100267554)

[3. Fejlesztői dokumentáció 17](#_Toc100267555)

[3.1 Témaválasztás indoklása 17](#_Toc100267556)

[3.2 Az alkalmazott fejlesztői eszközök 18](#_Toc100267557)

[3.2.1 Unity Hub 2.4.5 18](#_Toc100267558)

[3.2.2 Unity 2020.3.20f1 18](#_Toc100267559)

[3.2.3 Visual Studio 2022 17.0.5 18](#_Toc100267560)

[3.2.4 GIMP 18](#_Toc100267561)

[3.3 Adatmodell leírása 18](#_Toc100267563)

[3.4 Tesztelési dokumentáció 24](#_Toc100267564)

[3.5 Továbbfejlesztési lehetőségek 25](#_Toc100267565)

[3.6 Irodalomjegyzék, forrásmegjelölés 26](#_Toc100267566)

[4. Összegzés 28](#_Toc100267567)

# 1. Bevezetés, a probléma rövid ismertetése

A projektem témája egy 2D-s, felülnézetes játékprogram. Műfaját tekintve szerepjáték, ezen belül leginkább a dungeon crawler alműfajba tartozik. A záródolgozatom megvalósításához különböző szoftvereket vettem igénybe, melyek közül a legfontosabbak például a Unity, mely a játékmotort adja, a Microsoft Visual Studio, melyben a C# nyelvű szkripteket írom, a GIMP (GNU Image Manipulation Program), mely a tileset-ek és sprite-ok elkészítésében és a Unity Asset Store-ból letöltött asset-ek szerkesztésében segít, illetve a menü hátterét is ezzel készítettem, és a Microsoft PowerPoint, melynek segítségével az intro videó készült. A grafikus reprezentálás tile-based (rácsrendszer-alapú) megjelenítéssel történik, melyet a Unity beépített rácsrendszere segít megvalósítani.

A játék történetének középpontjában egy testvérpár áll. Az egyik testvért elfogták, és egy szörnyekkel teli börtön mélyére zárták, a mi feladatunk pedig az, hogy kiszabadítsuk a fogságból. A történet rövid bevezetéséhez egy cutscene-t (rövid jelenet) hívok segítségül, mely a játék első indításakor játszódik le. Ennek során szembesül a játékos a történet részleteivel. A játék egy, a mi világunkhoz hasonló kitalált világban játszódik, egy középkorhoz hasonló korban.

# 2. Felhasználói dokumentáció

## 2.1 A program általános specifikációja

A szerepjáték, másnéven RPG (role-playing game) az asztali, akár csak tollal és papírral játszható szerepjátékokból ered, melyek akár az 1970-es évek előtti eredetűek. Az első számítógépes szerepjátékok az 1970-es évek közepe-vége, 1980-as évek eleje felé bukkantak fel. Ilyen volt például az Ultima vagy a Wizardry, melyek 1981-ben jelentek meg. Régebben ezek a 2D-s szerepjátékok meglehetősen népszerűnek minősültek, ám napjainkban is gyakori ez a játékstílus, 3D-s megvalósításban. Rengeteg mai játékot inspiráltak ezek a régi szerepjátékok, mint például a Dragon Quest vagy Final Fantasy széria.

A játékom ezen belül is egy dungeon crawler, mely típusú játékok fő jellegzetessége, ahogy a név is megmutatja, egy kazamata bejárása. Ez egyértelműen abban nyilvánul meg az én projektemben, hogy a bejárható játéktér egy kazamatában helyezkedik el.

Az ilyen stílusú játékok fő elemei közé tartoznak a következők:

* a középpontban valamilyen történet áll
* nagy, nyitott pályák felfedezése (RPG)/ a játéktér egy kazamata (dungeon crawler)
* küldetések teljesítése
* különböző tárgyak, fegyverek, tapasztalati pontok (XP) gyűjtése
* a karakter képességeinek, szintjének növelése a gyűjtött tapasztalati pontok, vagy valamilyen más erőforrás által

Ezen felsorolt elemek közül az én játékom is tartalmaz többet. Egy kazamatát járhatunk be, és a karakter szintjét és fegyverét az ellenségek legyőzésével szerzett tapasztalati pontokkal fejleszthetjük. Annak ellenére, hogy a pálya jellege miatt eleve meg van kötve a játékos keze, azt részben mégis tetszőleges módon fedezhetjük fel.

A 2D-s játékok fő különbsége a más típusú megjelenítést használó játékokkal szemben a kamera pozíciója, jellege. A kamera nézetét tekintve 3 fő kameranézet lehetséges: izometrikus, felülnézetes, oldalnézetes. A projektem felülnézetes kamerapozíciót használ. Megvalósítás szempontjából is jelentősen egyszerűbb a kamera kezelése, mint egy 3D-s játék esetében, mivel egyáltalán nem, vagy csak ritkán kell a kamera mozgatásával foglalkozni. A játékos egyetlen, rögzített pontból látja a játékteret, a kamera pedig a karakter mozgását követi, így a belátható tér mérete is rögzített.

A játékprogram egy 2D-s, egyjátékosos, felülnézetes dungeon crawler szerepjáték. A játék teljesítéséhez be kell járnunk a pályát és ellenfelekkel kell megküzdenünk.

A képen képernyőkép látható

Automatikusan generált leírásAz ellenfelekkel történő harc során a játékos veszíthet az életerő-pontjaiból. Ha az életereje eléri a nullát, a játékos meghal, és visszakerül az előző mentési pontra. Ezeket a mentési pontokat a pályán elhelyezett, földön látható bekerített „tüzek” (bonfire) jelzik.

*1. ábra*

Egy játék hangulatához, atmoszférájához rengeteget tesz hozzá a színvilág és a háttérzenék, hangeffektek megválogatása. Próbáltam úgy összeválogatni a hangeffekteket, a háttérzenét és a színpalettát, hogy tükrözze főhősünk elveszettségét és nyomasztó hangulatot, érzést keltsen a játékosban. Főhősünk egyedül van egy szörnyekkel teli kazamatában.

A pálya alapját egy lilás barna szín és annak több árnyalata alkotja. A lilát már a mesékben is a „gonosz” színének mutatták be, sok főgonosz ruhatárában volt felfedezhető lila öltözék. A többi szín a pálya alapjához alkalmazkodik, nincsenek rikító színek, a színpaletta egymáshoz illeszkedő színekből áll, egyik szín sem tűnik ki a többi közül.

A zenék, hangeffektek is összhangban vannak ezzel, elősegítik a nyomasztó atmoszféra megteremtését. Mind a menüben, mind a játékmenet során hallhatunk háttérzenét, és ha főellenségbe botlunk, akkor ez a zene megváltozik, jelezve, hogy nagyobb harc, akadály áll előttünk, nem pedig egy átlagos ellenfél.

## 2.2 Rendszerkövetelmények

### 2.2.1 Hardver, szoftver követelmények

Minimális követelmények: Processzor: Intel Core 2 Duo E6320 (2\*1866) vagy

ekvivalens

Memória: 1 GB

Operációs rendszer: Microsoft Windows 7+

Grafika: DirectX 9+ kompatibilis videókártya

Tárhely: 100 MB szabad hely

Ajánlott követelmények: Processzor: Intel Core 2 Duo E6320 (2\*1866) vagy

ekvivalens

Memória: 2 GB

Operációs rendszer: Microsoft Windows 7+

Grafika: DirectX 9+ kompatibilis videókártya

Tárhely: 250 MB szabad hely

## 2.3 A program telepítése

A játékprogram telepítéséhez egyetlen fájlra van szükségünk, ez az „installer.exe”. Ha megtalálható a fájl a számítógépünkön, azt az erre történő dupla kattintással indíthatjuk el. Ez megkezdi a telepítés folyamatát. A telepítő csak angol nyelven érhető el. A telepítő alapértelmezetten létrehoz egy parancsikont a Start Menüben, ezt a beállítást nem írhatjuk felül.

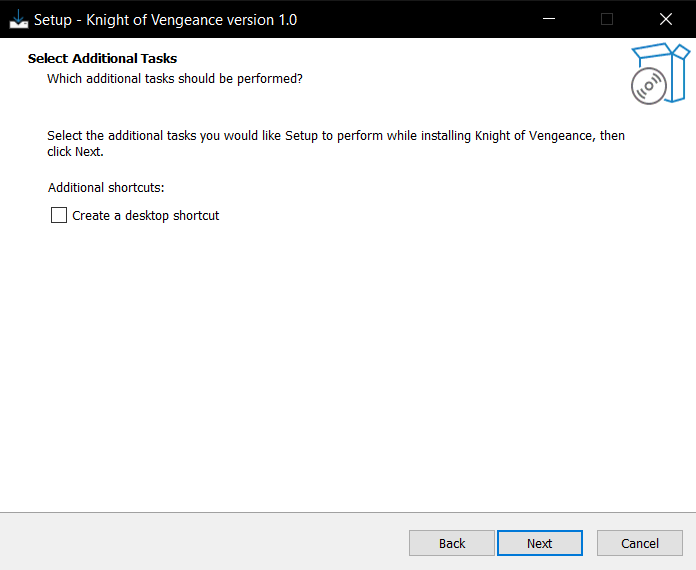
Miután engedélyeztük a rendszergazdai jogosultságokat a telepítő számára a felugró ablakban, az első képernyőn a telepítési mappa kiválasztása fogad minket. Az alapértelmezett mappa a rendszermeghajtónk „Programfájlok (x86)” mappája. Itt kerül létrehozásra egy „Knight of Vengeance” nevű mappa, ahová a játék fájljai kerülnek.

Emellett láthatjuk még itt a játék telepítéséhez szükséges szabad lemezterületet is.

A képen szöveg látható

Automatikusan generált leírásHa nincs különösebb okunk arra, hogy megváltoztassuk a telepítés mappáját, hagyjuk meg az alapértelmezett mappát, és kattintsuk a „Next” feliratú gombra a folytatáshoz.

*2. ábra*

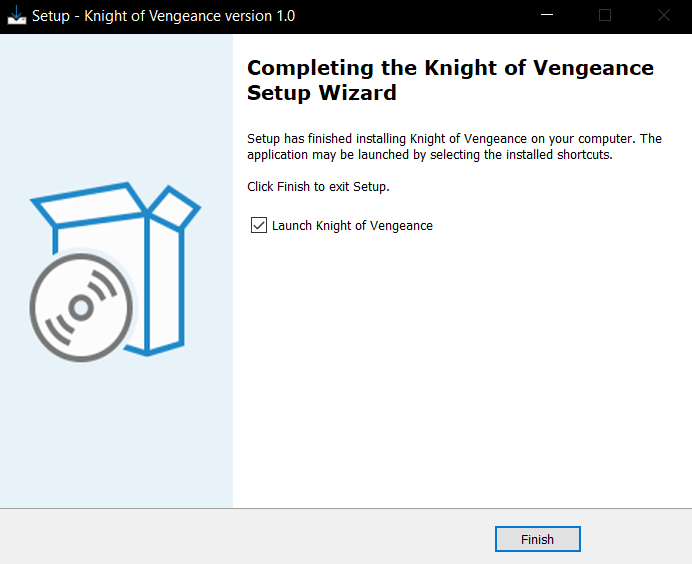
A következő képernyőn egy jelölőnégyzetet láthatunk, amivel meg tudjuk adni, hogy szeretnénk-e, ha a telepítő létrehozna nekünk egy parancsikont az asztalunkra. Alapértelmezetten ez üres. Folytatás előtt ezt bejelölhetjük, majd kattinthatunk megint a „Next” gombra.

*3. ábra*

A képen szöveg látható

Automatikusan generált leírásEzután egy ellenőrző képernyő fogad minket. Itt látható a választott telepítési könyvtár. Ha változtatni kívánunk ezen, azt a „Back” gombra kattintva tehetjük meg, mely egyesével léptet vissza a képernyőkön. Ha ellenőriztük, hogy megfelel nekünk ez a beállítás, az „Install” gombra kattintva telepíthetjük a szoftvert.

*4. ábra*

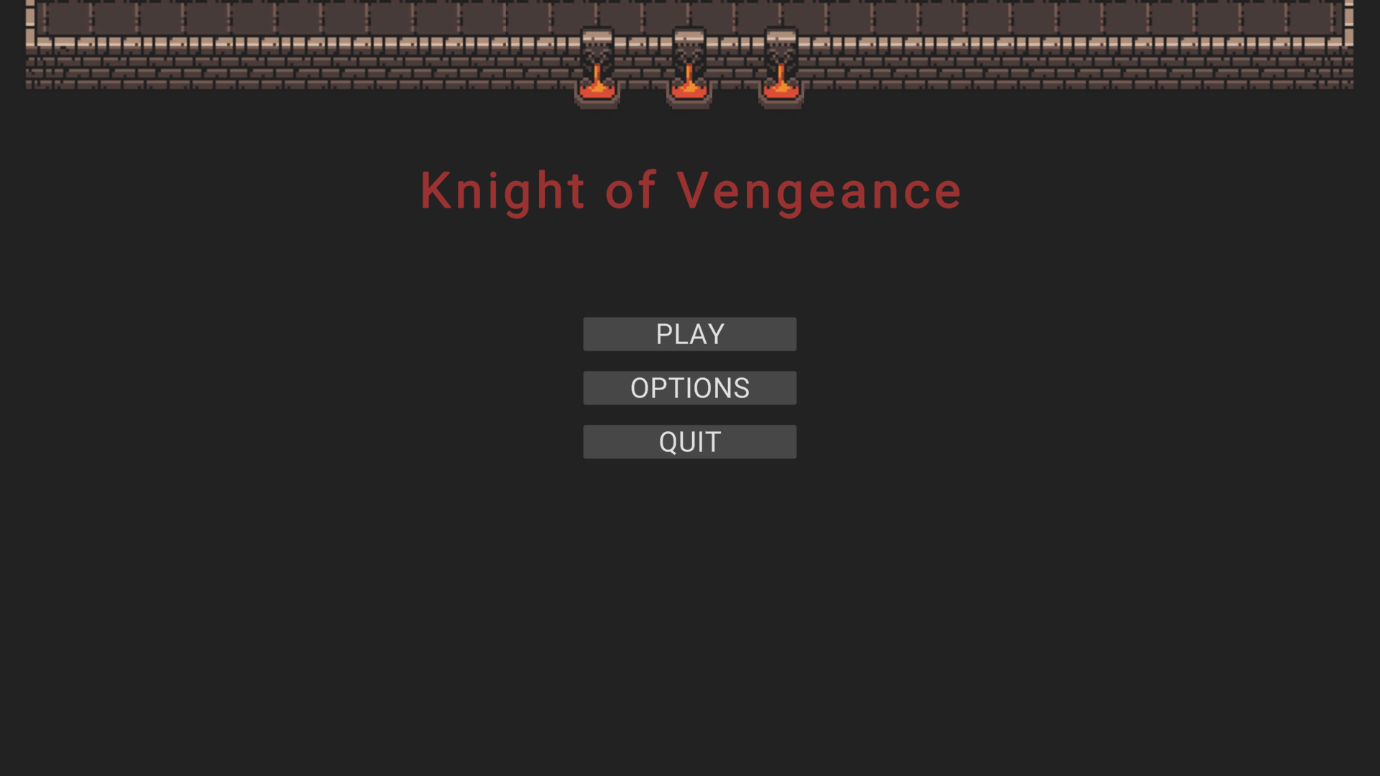
Ezután egy végső képernyő fogad minket, ahol újból egy jelölőnégyzetet láthatunk. Ezzel azt adhatjuk meg, hogy a telepítés végeztével elinduljon-e a játék. Ez alapértelmezetten be van pipálva. Ha ezt is megadtuk, akkor a „Finish” gombra kattintva léphetünk ki a telepítőből.

*5. ábra*

Ha a folyamat során bármikor meg szeretnénk szakítani a telepítést, azt a „Cancel” gombra kattintva tehetjük meg.

## 2.4 A program használatának részletes leírása

### 2.4.1 Főmenü felépítése

Ha elindítottuk a játékot, egy egyszerű, angol nyelvű menü fogad minket, 3 opcióval. Ezek a „Play”, „Options” és „Quit”.

*6. ábra*

A „Play” gombra történő kattintás után betöltődik a játék. Ha nincs meglévő mentésünk, még nem játszottunk a játékkal, akkor egy „tutorial”, bevezető szekció fogad minket, melynek segítségével elsajátíthatjuk a játék irányítását..

Amennyiben játszottunk már, a legutóbbi mentési pontunkra töltődünk be, maximum életerő-pontokkal, és újjáélednek a pályán lévő ellenségek.

A legalsó, „Quit” feliratú gomb bezárja a játékot.

### 2.4.2 Beállítások menü felépítése

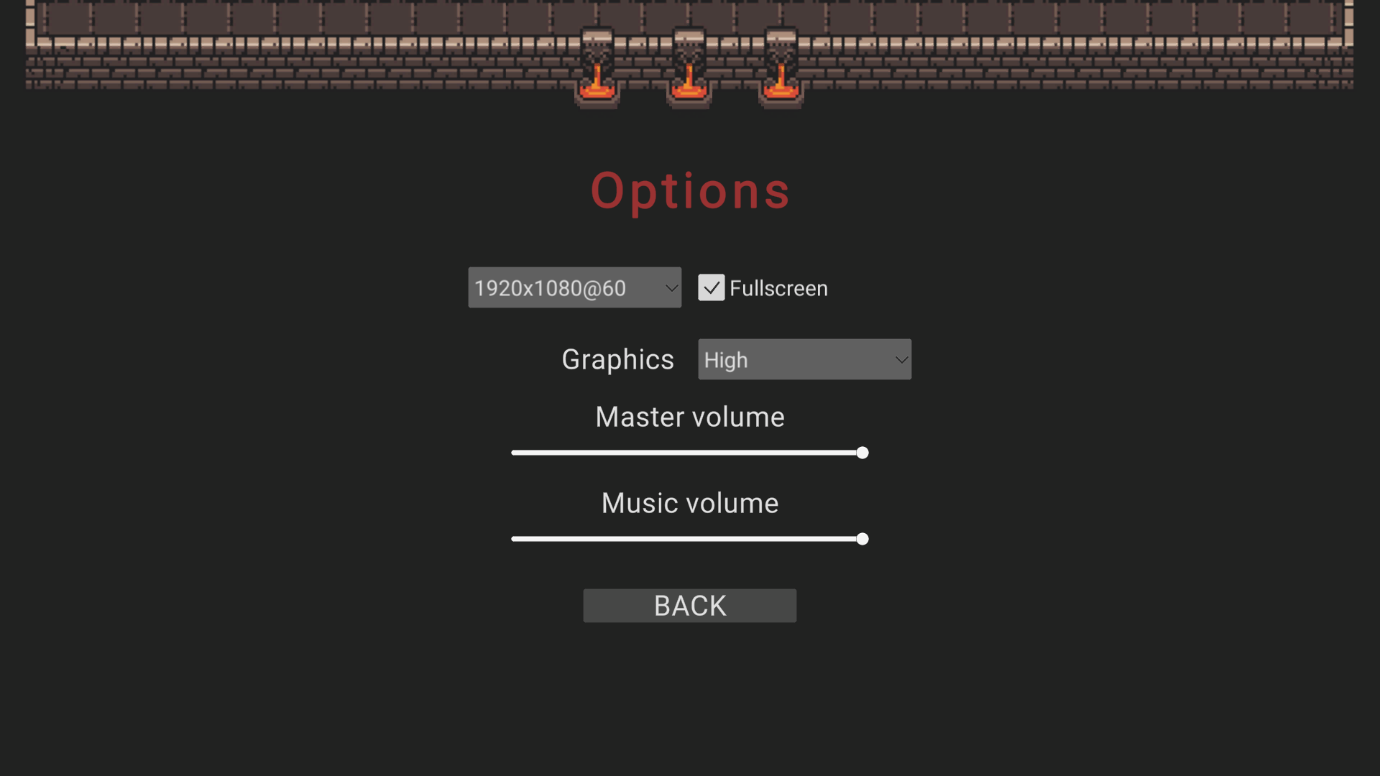
A főmenü második opciójára, az „Options”-re kattintva öt egyszerű beállítást láthatunk. Az első beállítás egy lenyíló lista, melyben a képernyőnk által támogatott felbontások láthatóak, mellettük az adott felbontás képfrissítési gyakoriságával. Kompatibilitási okok miatt minden támogatott felbontás, illetve egy adott felbontás minden elérhető képfrissítési változata elérhető itt, de a legjobb játékélmény érdekében ajánlott a legmagasabb beállítást használnunk.

Mellette egy jelölőnégyzetet láthatunk, „Fullscreen” felirattal. Ezzel azt állíthatjuk, hogy teljes képernyős legyen-e a játék ablaka.

Ezek alatt egy „Graphics” feliratú legördülő listát láthatunk. Ez 3 optciót tartalmaz, melyek a következők: „High”, „Medium”, „Low”. Az alapértelmezett beállítás a „High”. Ezt a legtöbb esetben nem szükséges állítanunk, mivel a játék megjelenítése meglehetősen szimpla, és nagyon kis gépigényű, de ez is kompatibilitási okok miatt található meg.

Ezalatt két csúszkát láthatunk, „Master volume” és „Music volume” feliratokkal. A „Master volume” feliratú csúszka a játék fő hangerejét állítja, ami alá a háttérzene is beletartozik, a „Music volume” viszont kizárólag a háttérzene hangerejét állítja. Ezek alapértelmezetten maximumon vannak.

Végső soron pedig egy „Back” feliratú gombot láthatunk, mellyel a főmenübe léphetünk vissza.

A beállítások menüje minden változtatást folyamatosan elment, nem szükséges külön foglalkoznunk ezzel.

*7. ábra*

### 2.4.3 Játék megállítása

A játék közben megtalálható „Pause” menüt az „Escape” billentyű lenyomásával hívhatjuk elő. Itt 4 opció fogad minket: „Resume”, „Options”, „Quit to menu”, „Quit to desktop”.

A menü megjelenésekor a játék teljesen megáll, nem tudjuk irányítani a karakterünket, nem mozognak az ellenfelek és sebződni sem tudunk.

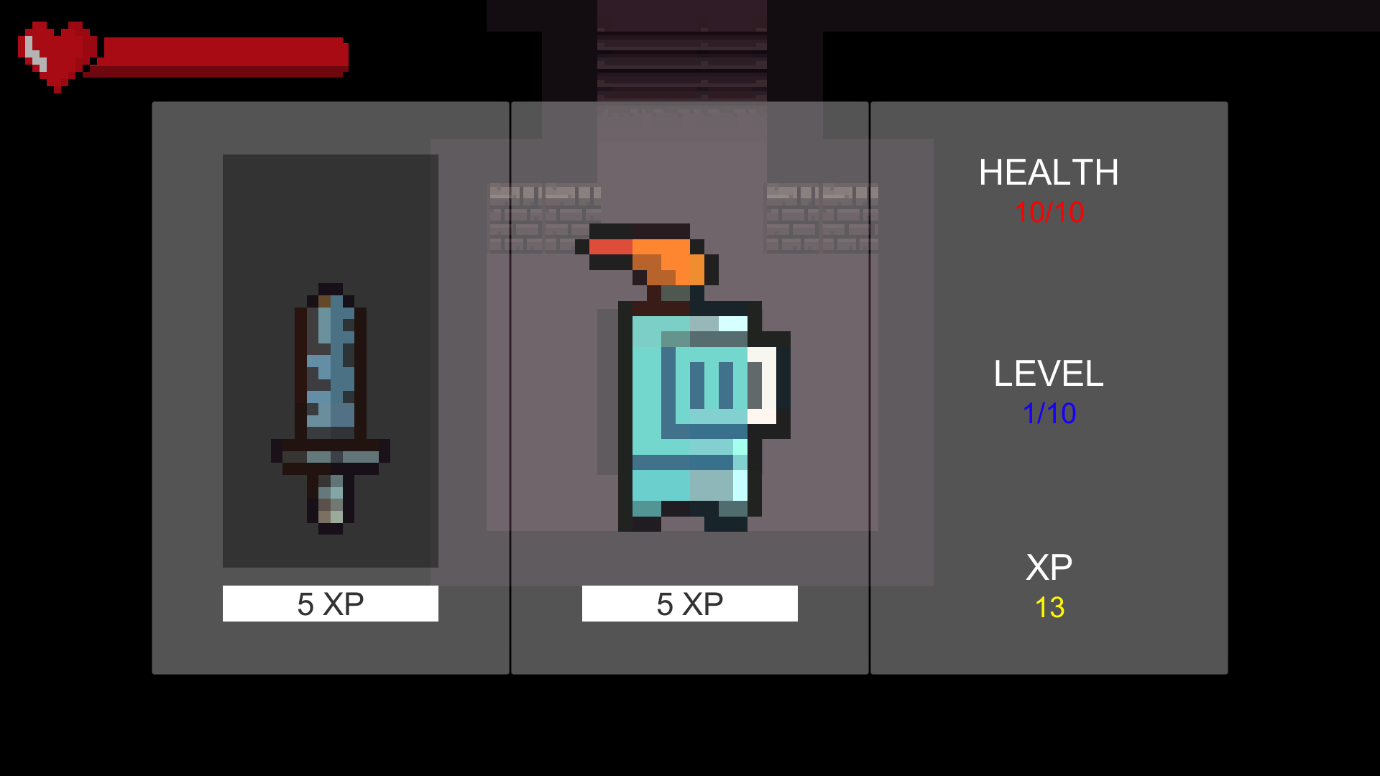
A „Resume” gomb megnyomásával bezárhatjuk a menüt, amit az „Escape” gomb újbóli megnyomásával is megtehetünk.

Az „Options” gomb megnyomásával elérhetjük a beállítások menüjét, mely pontosan ugyan úgy néz ki és funkcionál, mint a főmenüben. Amíg ez a menü aktív, nem tudjuk bezárni a „Pause” menüt, ha be szeretnénk zárni, előbb a „Back” gombra kell nyomnunk. *Lásd: 7. ábra*

A „Quit to menu” gomb a menübe visz minket, a „Quit to desktop” pedig teljesen bezárja a játékot. Fontos, hogy ezen gombok használatával lépjünk ki, mivel a játék ilyenkor menti csak el az összegyűjtött tapasztalati pontjainkat, szintünket stb. Ezáltal, ha nem jutunk el a következő mentési pontig, viszont ki szeretnénk lépni, az előrehaladásunk nem vész el.

*8. ábra*

### 2.4.4 Karakter menü

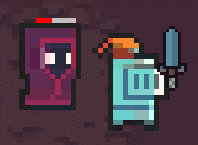
A játék során elérhető még egy menü, ez pedig a karakterünk menüje. Ezt az „I” billentyű lenyomásával érhetjük el. Ebben a menüben a fegyverünket és a karakterünket fejleszthetjük, és megtekinthetjük életerő-pontjainkat (Health), szintünket (Level) és jelenlegi tapasztalati pontjainkat (XP). A karakter, illetve a fegyver fejlesztése tapasztalati pontokba kerül, ennek értékét pedig a gombokon láthatjuk. Ha elértük a legnagyobb szintet, akkor a „MAX” felirat jelenik meg a gombokon. A játék a menü előhívása után nem áll meg, ilyenkor is tudjuk irányítani a karakterünket és tudunk támadni, sebződni.

*9. ábra*

### 2.4.5 Irányítás, vizuális elemek, játékmechanikák

Maga a játék irányítása egyszerű, könnyen érthető. A karakter mozgatása a „WASD” billentyűk használatával történik.

Támadni a „Space” billentyű lenyomásával tudunk. Minden fegyvernek különböző sebzése és visszalökési ereje van. A visszalökési erő határozza meg, hogy sikeres támadás esetén az ellenfelünket milyen távolságra lökjük el magunktól. Az ellenfelek szintén el tudnak lökni minket.

Harc közben az ellenfelek felett elhelyezkedik egy sáv, mely az életerő-pontjaikat reprezentálja, és sebződés után jelenik csak meg. Főellenséggel folytatott harc során a képernyő közepén, alul helyezkedik ez el, és sokkal több helyet is tölt ki, mivel a főellenségek életereje sokkal több. Emellett ha egy főellenséget tartalmazó szobába lépünk, onnan nem tudunk visszamenni amíg a harc nem fejeződik be, legyen az sikeres vagy sem.

*10.. ábra*

*11. ábra*

Az általunk irányított karakter is rendelkezik egy életerő-pontot jelző sávval, ez a képernyő bal felső sarkában látható.

*12. ábra*

Játék során találkozhatunk még kutakkal, melyek egy megadott mennyiségű életerő-pontot töltenek vissza, amikor hozzájuk érünk.

*13. ábra*

Emellett találkozhatunk még ládákkal is, melyekhez ha hozzáérünk, 5 és 10 közötti mennyiségű tapasztalati pontot kapunk.

*14. ábra*

A képen szöveg, képernyőkép, vektorgrafika látható

Automatikusan generált leírásÉleterő visszatöltése esetén zöld, tapasztalati pontok szerzése esetén fehér, illetve mind a karakterünk, mind az ellenfelek sebződése esetén pedig piros, a mennyiségnek megfelelő szöveg ugrik fel a karakterünk mellett.

*15. ábra*

Halál esetén egy párbeszédablak jelenik meg, ahol a „Respawn” gombra kattintva éledhetünk újra a legutóbbi mentési ponton. Ilyenkor a pálya ellenségei is újraélednek.

*16. ábra*

# 3. Fejlesztői dokumentáció

## 3.1 Témaválasztás indoklása

Mindig is szerettem videojátékokkal játszani, de ez csak egy kis része volt a motivációmnak. A döntő indok az volt, hogy sokkal közelebb áll hozzám ez a téma, mint a weblapkészítés. Legfőképpen az adatbázis-kezelés része miatt. Még ha maga a weblapkészítés nem is a gyengém, az adatbázisok terén nem vagyok olyan jártas, főleg emiatt esett a választásom a játékprogram készítésére.

A játék motorjaként a Unity Engine-re esett a választásom. Ennek több oka is volt. Először is fontos volt a szkriptek programozási nyelve. A Unity motor a C# programozási nyelvet támogatja a szkriptek írásához. Ezt a nyelvet tanuljuk már több mint 4 éve, így bízom a tudásomban. Mivel sosem foglalkoztam még játékfejlesztéssel a múltban, ezért egy ismert nyelv használata magabiztosabbá tett, és megkönnyítette a fejlesztést. Fontos volt továbbá a motor kezelhetősége, tanulhatósága. Rengeteg segítő videó található az interneten, és maga a motor is könnyen érthető, kezelhető, mégis lehet benne fejleszteni komplex játékokat.

A játék típusaként azért egy 2D-s játékot választottam, mert úgy gondoltam, hogy egy ilyen típusú játék elkészítése jelentősen könnyebb lenne, mint egy 3D-s játéké, komplexitását tekintve viszont megfelelő lenne projektem témájaként. Emellett rengeteg oktató videót találtam ebben a témakörben, és ingyenes, jó minőségű asset-eket is nagyobb számban találtam, mint 3D-s játékokhoz.

Játékmotor szempontjából eleinte szóba jött még az Unreal Engine is, ami fejlettebb, mint a Unity, főleg grafika szempontjából, viszont használata is komplexebb, és a C++ nyelvet használja szkriptelésre. A végső választásom így nem volt túl nehéz, mivel az Unreal Engine megtanulása és a C++ tudásom elmélyítése hatalmas akadályt jelentett volna számomra a határidő betartásában, és a fejlettebb grafikai megjelenítést sem tudtam volna kihasználni egy 2D-s projekt keretein belül.

A C# szkriptek szerkesztésére használt fejlesztőkörnyezetként a Microsoft Visual Studio-t választottam. Ez egyértelmű választás, mivel ebben több mint 4 év tapasztalatom volt, hiszen az oktatás során is ezt használtuk, és ingyenes környezetet biztosít.

## 3.2 Az alkalmazott fejlesztői eszközök

### 3.2.1 Unity Hub 2.4.5

A Unity Hub a Unity Editor-hoz kapcsolódó szoftver. Segítséget nyújt a fiókkezelésben, licenszek aktiválásában és a projektek készítésében és rendszerezésében. Ezen a programon keresztül telepíthetjük és futtathatjuk a Unity Editor-t.

### 3.2.2 Unity 2020.3.20f1

„*A Unity egy videójáték-motor, amelyet a Unity Technologies fejleszt. A Unity segítségével háromdimenziós illetve kétdimenziós videójátékokat, ezen kívül egyéb interaktív jellegű tartalmakat lehet létrehozni, például építészeti látványterveket vagy valós idejű háromdimenziós animációkat.*”

Ezen játékmotor segítségével fejlesztettem a játékomat. A program fő részei a Unity Editor-ban, a játékmotor programjában készültek.

### 3.2.3 Visual Studio 2022 17.0.5

„*A Visual Studio a Microsoft több programozási nyelvet tartalmazó fejlesztőkörnyezete, amely az évek során egyre több új programnyelvvel bővült. Jelenleg a F#, C++, C# (ejtsd: Szí-sárp) és Visual Basic programozási nyelveket, valamint az XML-t támogatja.*”

Ezt az IDE-t (Integrated Development Environment – integrált fejlesztőkörnyezet) használtam a C# nyelvű szkriptek megírására.

### 3.2.4 GIMP

„*A GIMP (GNU Image Manipulation Program) egy bittérképes képszerkesztő program. Támogatja a rétegek kezelését, az átlátszóságot. Van némi vektorgrafikus támogatás is benne.*”

Ezt a szoftvert javarészt az ingyenesen letölthető asset-ek kis mértékű módosítására használtam, illetve a főmenü animált hátterét ennek segítségével készítettem el.

A fent említett szoftverek mindegyike ingyenes, így ezért is esett rájuk a választásom. Emellett közülük többet is használtunk az oktatás során, így volt tapasztalatom bennük.

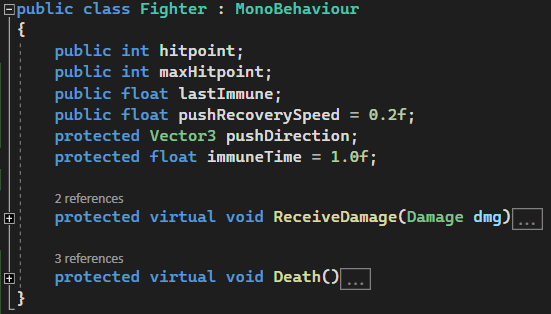
## 3.3 Adatmodell leírása

Habár a játék nem túl komplex, próbáltam úgy kialakítani a szerkezetét, hogy könnyen skálázható, moduláris legyen. Így nincsen egyetlen fő osztályom, a legfontosabb funkciók is több, kisebb osztályból tevődnek össze.

A játékom két legfontosabb komponense a főszereplő karakter, illetve az ellenfelek. Mint minden, ezek is sok kisebb osztályból tevődnek össze.

A főszereplő karakter osztálya a „Player” nevű osztály. Ez az osztály a „Mover” nevű absztrakt osztályt örökli, a „Mover” pedig a „Fighter” osztályt örökli. A játékos karaktere alá tartozik a fegyvere, mely a „Weapon” osztályt viseli, és a „Collidable” osztályt örökli.

Mivel ugyanazokból az osztályokból örököl mind a „Player”, mind az „Enemy”, ezért egyszerre írok róluk.

A „Fighter” osztály mezői és metódusai:

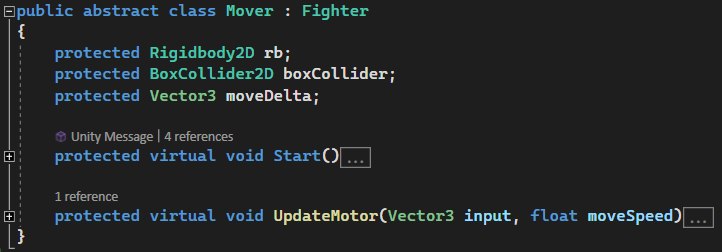
*17. ábra*

ReceiveDamage(dmg):

Ez a metódus akkor fut le, ha egy ellenfél vagy a karakterünk sebződik. A karakter esetében a fegyver OnCollide metódusán belül hívódik meg a SendMessage metódus segítségével. Ez azt jelenti, hogy ha a fegyver és az ellenfél ütközik, a fegyver üzenetet küld az ellenfélnek, hogy hívja meg magán a ReceiveDamage metódust, és paraméterként átadja a sebzést.

Maga a ReceiveDamage metódus megvizsgálja, hogy még immunisak vagyunk-e az előző ütés miatt, ha pedig nem, akkor levonja a sebzést az életerőnkből, ha pedig az eléri a nullát, akkor meghívja a Death metódust.

Death():

Ezt a metódust minden esetben felülírom a gyermekosztályokban.

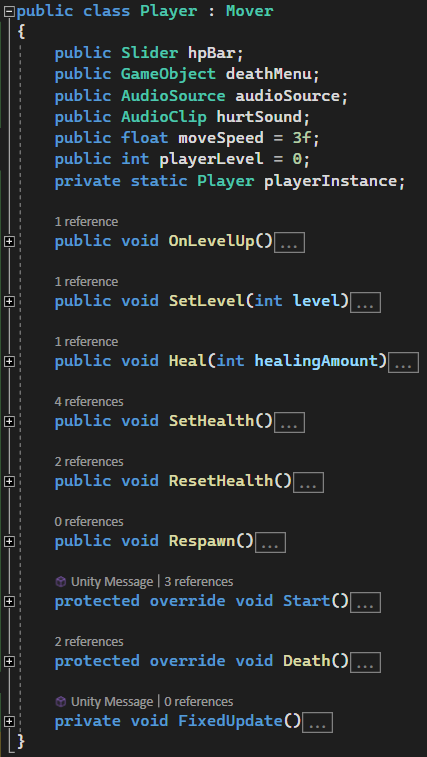
A „Mover” osztály mezői és metódusai:

*18. ábra*

Mivel ez egy absztrakt osztály, nem példányosítható, csakis örököltetésre használható. Azért választottam külön a játékos karakterétől a mozgást, mivel eredetileg írtam egy primitív útkereső algoritmust az ellenfelek számára, és a „Mover” osztály segítségével mozgattam őket is. Később a NavMesh Plus Unity bővítményre cseréltem ezt, mivel az sokkal hatékonyabban működik.

UpdateMotor(input, moveSpeed):

Ez a függvény egyszerűen a mozgatást kezeli. A moveSpeed a mozgási sebességet jelenti. Emellett ez a metódus kezeli még mind a játékos, mind az ellenfél visszalökését, mikor sebződnek.

A „Player” osztály mezői és metódusai:

*19. ábra*

OnLevelUp(), SetLevel(level):

Különböző szintlépésekkel kapcsolatos metódusok. A szint mentése és betöltése miatt egyszerűbb volt két külön metódusra bontani a szintlépést.

Heal(healingAmount), SetHealth(), ResetHealth():

Az életerő-pontok beállításához, visszaállításához használatos metódusok. Az életerőt visszatöltő kutak, illetve az újjáéledés miatt volt érdemes különválasztani ezeket a metódusokat.

Death():

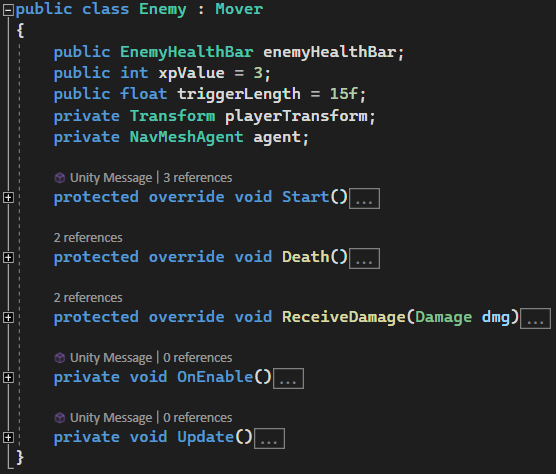
Itt kapcsolom ki a játékos objektumát, megjelenítem a halál utáni párbeszédablakot, és kikapcsolom a megfelelő menüket.

Respawn():

Visszadobja a játékost a legutóbbi mentési pontjára. A halál utáni párbeszédablak „Respawn” gombjának lenyomására fut le. Itt kezelem a többi menü engedélyezését, a játékos életerejének visszatöltését, és immunissá teszem, hogy ne sebződjön az újjáéledés pillanatában újra, a halál pillanatában elszenvedett sebzés miatt.

FixedUpdate():

Fix képfrissítéssel fut le, minden képkocka során. Itt kezelem a játékos mozgatásához szükséges billentyűk lenyomását. Azért fontos a FixedUpdate használata az Update helyett, mert ha Update metódust használnék itt, akkor a játékos mozgási sebessége függene a képfrissítési rátától.

Az „Enemy” osztály mezői és metódusai:

*20. ábra*

Death():

Törli az ellenfelet a pályáról és megadja a játékosnak a tapasztalati pontokat.

ReceiveDamage(dmg):

Meghívja a szülő osztály metódusát. Egy dologban különbözik csak, mivel meghívja még az életerő-sáv SetHealth metódusát, ami bemenetként kapja az aktuális, illetve a maximális életerő-pontokat. Így a metódus felülírása az életerő-sáv miatt szükséges.

A „Player” és az „Enemy” osztályhoz kapcsolódó további osztályok:

* Boss
* BossHealthBar
* Collidable
* EnemyHealthBar
* EnemyHitbox
* HealthBar
* Weapon

A képen képernyőkép, Téglalap, tér, Post-it cetli látható

Automatikusan generált leírás*UML osztálydiagram*

*21. ábra*

## 3.4 Tesztelési dokumentáció

A szoftver tesztelését teljes mértékben manuálisan végeztem. Minden nagyobb kódrészlet elkészültével teszteltem a szoftvert, így állítottam elő a teszteseteket. Ez volt a feketedobozos tesztelés, mivel itt csupán a program, az adott funkció helyes működését vizsgáltam. Ez annyit jelentett, hogy a Unity szerkesztőjében elindítottam a játékot, majd teszteltem, hogy az adott funkció a kívánt módon működik-e. Ez jelentette például azt, hogy a megfelelő menü előjön a megfelelő gomb lenyomására, a karakter a megfelelő műveletet hajtja végre a megfelelő gomb lenyomására stb.

Mivel fejlesztőként hozzáférésem volt a kódhoz is, hiszen én magam írtam azt, így próbáltam annak tükrében minden szélsőséges esetet is kipróbálni, minden eshetőséget tesztelni egy adott funkciónál. Ez volt a fehérdobozos része a tesztelésnek. Ilyenkor egyszerre figyeltem a kódot és a játék működését, sorról sorra próbáltam megérteni az épp futó folyamatok logikáját, a kód, a program szerkezetét, és eszerint a lehető legalaposabban tesztelni az adott kódrészletet. Erre példa lehet, hogy amikor a beállítások mentését kódoltam, több akadályba is ütköztem. Nem kerültek megfelelően mentésre az adott beállítások, illetve nem töltődtek be helyesen. Ekkor a játék éles, „build” verzióját futtattam, mellette pedig figyeltem a forráskódot, és sorról sorra próbáltam megérteni a függvényhívások sorozatát, és azt, hogy mi okozza a hibát. Ezáltal ezt sikeresen ki is tudtam javítani.

Mikor úgy gondoltam, hogy a szoftverem elérte a végleges állapotát, azt elküldtem osztálytársaknak, illetve más, informatikában kevésbé jártas ismerősöknek is. Ezt a módot nevezhetjük alfa-, illetve bétatesztnek. Ezáltal kétféle különböző szemszögből kaphattam teszteseteket: egy olyan ember szemszögéből, aki, habár nem ismeri a mögöttes kódot, valamennyire mégis belelát a játék logikájába, és egy átlagos felhasználó szemszögéből, aki semmit sem tud a játék mögöttes működéséről.

Ez azért volt hasznos, mert az átlag felhasználók valós, életszerű módon használták a szoftvert, és olyan módokon tesztelték azt, melyek nekem eszembe sem jutottak volna. Ezáltal sikerült kiszűrnöm és kijavítanom pár kisebb, egyszerűbb hibát. Ezeken felül viszont nem akadt más probléma az elkészült szoftverrel. Ez a fajta teszt segített még a „tutorial”, bevezető szekció kialakításában is, hiszen ha egy teljesen átlagos felhasználó is megérti a játék irányítását, mechanikáit, akkor tudhatom, hogy kellő részletességgel magyaráztam ezeket el.

## 3.5 Továbbfejlesztési lehetőségek

Mivel az elkészült játék csupán egy „demo”, nagyon rövid és csak egy pályából áll. A jövőben érdemes lenne kibővíteni a játékot új pályákkal.

A különböző pályák új ellenfeleket és főellenségeket is tartalmazhatnának. Jelenleg a játék ellenségei nem változatosak, és a játékmenet is szimpla, egyszerű. Új ellenfelek bevezetésével a harc mechanikáit is lehetne fejleszteni. A játékos megszerezhetne új fegyvereket, illetve különböző támadásokat sajátíthatna el. A főellenségek rendelkezhetnének speciális támadásokkal, ami tovább színesítené a játékmenetet.

Különböző játékmódokat is lehetne bevezetni, mint például egy „hardcore” mód, melyben a karakterünk egyetlen élettel rendelkezik, vagyis ha meghalunk a játékban, akkor a végigjátszásunk véget ér, minden fejlődésünk elvész. Ez a tulajdonság a Roguelike játékok egyik fő eleme. Ezzel a móddal a nagyobb kihívásokat kedvelő játékosok felé tehetném vonzóbbá a játékom.

Az újbóli végigjátszások élményét is nagyban bővíteni lehetne, mivel jelenleg nincsen semmilyen értéke az újrajátszásoknak. A játék befejezése után a játékos választhatna, hogy szeretné-e újrajátszani a játékot. Ekkor a karakter megtartaná a fegyvereit és a szintjét, viszont az ellenfelek is több életerővel rendelkeznének, többet sebeznének. Ezáltal a játék eleje meglehetősen könnyű lenne, viszont ahogy haladunk előre, már nem számítana annyit a fejlett fegyverünk és magas szintünk, így újra kihívást jelentene a játék. Ehhez szükség lenne újabb fegyverekre és a nagyobb szintre lépés lehetőségére, melyek által még tovább fejleszthetnénk karakterünket, és újra befejezhetnénk a játékot. Habár egy idő után elfogyna a plusz tartalom, a kihívást gyakorlatilag a végtelenségig lehetne növelni, így színesítve az újbóli végigjátszásokat.

Feltéve, hogy az előbb említett lehetőségeket megvalósítottam, egy végső opcióm a többjátékosos mód hozzáadása lehetne. Ezt ténylegesen csak akkor lenne kifizetődő megvalósítani, ha a többi lehetőséggel már végeztem, mivel ez már teljes mértkében csak plusz tartalmat jelentene a már jól kidolgozott játékomban. Ez a mód az úgymond alkalmi, átlagos játékosok számára kedvezne, mivel ezek a játékosok gyakran csupán a barátaikkal folytatott kikapcsolódás miatt játszanak játékokkal.

## 3.6 Irodalomjegyzék, forrásmegjelölés

<https://0x72.itch.io/16x16-dungeon-tileset> (megtekintve: 2021. január 14.)

<https://0x72.itch.io/dungeontileset-ii> (megtekintve: 2021. január 14.)

<https://www.youtube.com/watch?v=b8YUfee_pzc&t=19508s> (megtekintve: 2021. január 14.)

<https://www.youtube.com/watch?v=ryISV_nH8qw&t=174s> (megtekintve: 2021. január 14.)

<https://www.youtube.com/watch?v=whzomFgjT50> (megtekintve: 2021. január 14.)

<https://www.youtube.com/watch?v=sPiVz1k-fEs&t=337s> (megtekintve: 2021. január 18.)

<https://edit.elte.hu/xmlui/bitstream/handle/10831/52979/Szakdolgozat.pdf> (megtekintve: 2021. január 22.)

<https://edit.elte.hu/xmlui/bitstream/handle/10831/40727/Enyingi%20K%C3%A1roly_szakdolgozat.pdf> (megtekintve: 2021. január 22.)

<https://hu.wikipedia.org/wiki/Unity_(j%C3%A1t%C3%A9kmotor)> (megtekintve: 2022. február 4.)

<https://hu.wikipedia.org/wiki/Microsoft_Visual_Studio> (megtekintve: 2022. február 4.)

<https://hu.wikipedia.org/wiki/C_Sharp> (megtekintve: 2022. február 4.)

<https://hu.wikipedia.org/wiki/GIMP> (megtekintve: 2022. február 4.)

<https://www.youtube.com/watch?v=BLfNP4Sc_iA> (megtekintve: 2022. április 4.)

<https://www.youtube.com/watch?v=v1UGTTeQzbo> (megtekintve: 2022. április 4.)

<https://www.youtube.com/watch?v=rtpHU1kfabI&t=81s> (megtekintve: 2022. április 4.)

<https://www.youtube.com/watch?v=SDfEytEjb5o&t=773s> (megtekintve: 2022. április 5.)

<https://www.youtube.com/watch?v=4iFNZvNzRHw&list=PL170TfEhZz-9EkGVU-vtVrOgrSsnk_ldn> (megtekintve: 2022.április 5.)

<https://www.youtube.com/watch?v=HeFd3kE8im0> (megtekintve: 2022. április 4.)

<https://www.youtube.com/watch?v=Rge79Y2WzH0&t=3s> (megtekintve: 2022. április 5.)

<https://mixkit.co/free-sound-effects/sword/> (megtekintve: 2022. április 5.)

<https://mixkit.co/free-sound-effects/hurt/> (megtekintve: 2022. április 5.)

# 4. Összegzés

Összességében elégedett vagyok a munkámmal. A kitűzött céljaim nagy részét sikerült megvalósítanom, és a tesztelés során kapott visszajelzések is pozitívak voltak.

Véleményem szerint a legtöbb fejlesztésre maga a játékmenet szorulna. Emellett a játék demó jellege miatt a történet sem bontakozik ki megfelelően.

Ettől függetlenül úgy gondolom, hogy kis továbbfejlesztés, a játékmenet színesítése és a pálya kibővítése után egy modern játék követelményeinek megfelelő, akár piacképes játékká is válhatna a munkám.

A fejlesztés során sokszor a legnagyobb kihívást nem is maga a kódolás, hanem a dokumentáció írása jelentette. Ebből is rengeteget sikerült tanulnom.

Emellett a szoftver elkészítése során is rengeteg tudással gyarapodtam. Mivel a témavezetők feladata csak úgymond a helyes ösvényre való terelés volt, így csak magamra tudtam hagyatkozni, magamtól kellett elsajátítanom a Unity játékmotor használatát. Úgy gondolom, hogy ez sikerült is, és sok tapasztalatot szereztem a játékfejlesztés terén.

Önmagam oktatása terén is sokat fejlődtem, a fent említett okok miatt. Meg kellett tanulnom, hogy hogyan tudom az adott „tutorial” videóban látottakat a saját problémámon alkalmazni.

Összességében rengeteg hasznos tudással gyarapodtam a záródolgozatom elkészítése során, melyet minden bizonnyal a jövőben is fel fogok majd tudni használni.