

Veszprémi Szakképzési Centrum Ipari Technikum

**Szakképzés neve: Szoftverfejlesztő**

**OKJ száma: 54 213 05**

# SZAKDOLGOZAT

## CastleLand Videójáték

Cseh Balázs  
témavezető

Dömök Martin  
13.C



Veszprém, 2022

## Tartalom

1. Bevezetés.....	3
1.1 Bevezetés.....	3
1.2 A tower defense játékok alap koncepciója.....	3
2. Felhasználói dokumentáció.....	4
2.1 A játékban elérhető funkciók.....	4
2.2 Hardver és szoftverkövetelmények.....	6
2.3 Telepítés menete .....	7
2.4 Egy tipikus futtatás kinézete .....	8
2.5 Ellenfelek típusai .....	12
2.6 Tornyok/védelmi egységek típusai .....	14
3. Fejlesztői dokumentáció .....	15
3.1 Témaválasztás indoklása .....	15
3.2 Fejlesztői környezet .....	16
3.3 Főbb algoritmusok.....	18
3.4 Továbbfejlesztési lehetőségek.....	24
3.5 Tesztelés.....	25
3.6 Összegzés.....	26
3.6 Források, szakirodalom .....	27
3.7 Licenszek, felhasznált assetek.....	30

# 1. Bevezetés

## 1.1 Bevezetés

A szakdolgozatom témája egy 3D-s, felülnézetes játékprogram. Műfaját tekintve egy klasszikus Tower Defense játék. Bizonyára ön is rendelkezik egy kedves emlékekkel legalább egy tower defense játékról. Ez a műfaj a kétezres években élte fénykorát. Én kiskoromban rengeteg időt töltöttem a világhálón megtalálható játékok kijátszásával, és talán ehhez a játék műfajhoz köthetőek a legkedvesebb emlékeim. Még ma is élénken él bennem annak az emléke, amikor a nővérem az y8 nevezetű weboldalon megmutatta ennek a játéktípusnak a sajátosságait, és játékmenetét. Mivel ez számomra egy kedves emlék emiatt döntöttem úgy, hogy szeretnék egy saját játékot fejleszteni ebben a témakörben. A játék komplexitása miatt inkább az asztali alkalmazás mellett döntöttem a webalkalmazással szemben. A szakdolgozatom megvalósításához különböző szoftvereket vettem igénybe, melyek közül a legfontosabbak például a Unity, mely a játékmotort adja, a Microsoft Visual Studio, melyben a C# nyelven íródott szkripteket készítettem el, illetve a GIMP (GNU Image Manipulation Program), melyben a szükséges képállományokat szerkesztettem.

## 1.2 A tower defense játékok alap koncepciója

A műfaj lényege, hogy a játék pályáján megvan határozva egy előre megadott útvonal, amelyet követve az ellenfeleknek kell érniük annak végpontját. A játékos célja ezt megakadályozni a lehető leghatásosabb módszerrel. A védekezésre általában többfajta eszköz/egység áll rendelkezésre a játékosnak, amelyek különféle erősséggel rendelkeznek. Ezen védelmi eszközök útvonal menti elhelyezésével lehet elpusztítani az ellenséges egységeket. Hogy a játék ne legyen túl egyszerű az ellenfelek csoportosan támadnak, és azoknak megsemmisítésével lehet valamilyen valutát gyűjteni, amelyből újabb védelmi vonalat tudunk létrehozni. Mikor egy csoport megsemmisül hamarosan egy újabb kör indul amire a játékosnak fel kell készülnie. Minden egyes körben az ellenfelek száma nő, vagy azoknak sebessége esetleg életereje, így fokozatosan nehezedik a felhasználó számára a játék.

## 2. Felhasználói dokumentáció

### 2.1 A játékban elérhető funkciók

- Új játék kezdése:
  - Egy új játékmenet indítása a játék elejéről. Az „Escape” billentyű megnyomásával elérhető a szünet menü, itt található egy „Új játék” opció, melynek segítségével a menübe való kilépés nélkül indítható egy új játékmenet.
- Szüneteltetés:
  - A játékállásának megfagyasztása, szüneteltetése. Ezt az „Escape” billentyű megnyomásával teheti meg a játékos, miután belépett egy aktív játékmenetbe, ilyenkor megjelenik egy ennek megfelelő menü. Itt található egy „Folytatás” opció melyre kattintva a menü eltűnik és a játék folytatódik.
- Védelmi egységek elhelyezése:
  - A játékosnak lehetősége van a felhasználói interfészen keresztül védelmi egységek vásárlására, majd azokat a kedve szerint elhelyezni a játék pályáján. Az erre alkalmas felhasználói felület a képernyő alján található, itt kattintással tudja kiválasztani a számára megfelelő egységet, miután ez megtörtént egy üres mező fölé húzva a kurzort a játék jelzi, hogy az elhelyezhető-e ott vagy sem, ha elhelyezhető, akkor a mező zöld színű lesz, ha ez viszont nem tehető meg akkor pirossá változik. Ha az elhelyezés sikeresen megtörténik, akkor a játék levonja az egység árát.
- Beállítások módosítása:
  - A menüben elérhető beállítások menüponton belül a játék fő hangerejének módosítása, felbontás kiválasztása, teljes képernyős mód változtatása, megjelenési grafika módosítása.

## 2. Felhasználói dokumentáció

- Védelmi egységek eladása:
  - Játék közben lehetőség nyílik egy egységre kattintva azt eladni, ha a felhasználó rákattint egy egységre akkor az egység fölött megjelenik egy gomb melynek segítségével az egységet eladhatjuk és annak értékének felét visszkapjuk.
- Az idő múlásának mértékét megváltoztatni:
  - Aktív játékmenet közben a játékos az „N” billentyű segítségével lassíthat, míg az „M” billentyű megnyomásával gyorsíthat egy bizonyos mértékig az idő múlásán. Ezzel szemben a „P” betű megnyomásával a játékban teljesen megáll az idő, ennek funkcionalitása, hogy a felhasználó úgy tud egységeket elhelyezni, hogy nem kell aggódnia, hogy egy ellenséges egység elérné a várát.

## 2.2 Hardver és szoftverkövetelmények

Minimális gépigény:

- Processzor: Intel Core i5-3570K vagy azzal egyenértékű
- Videókártya: Nvidia GeForce GTX 1050 2 GB
- Tárhely: 150 MB szabad tárhely
- Memória: 150 MB szabad memória
- Operációs rendszer: Windows 10

Ajánlott gépigény:

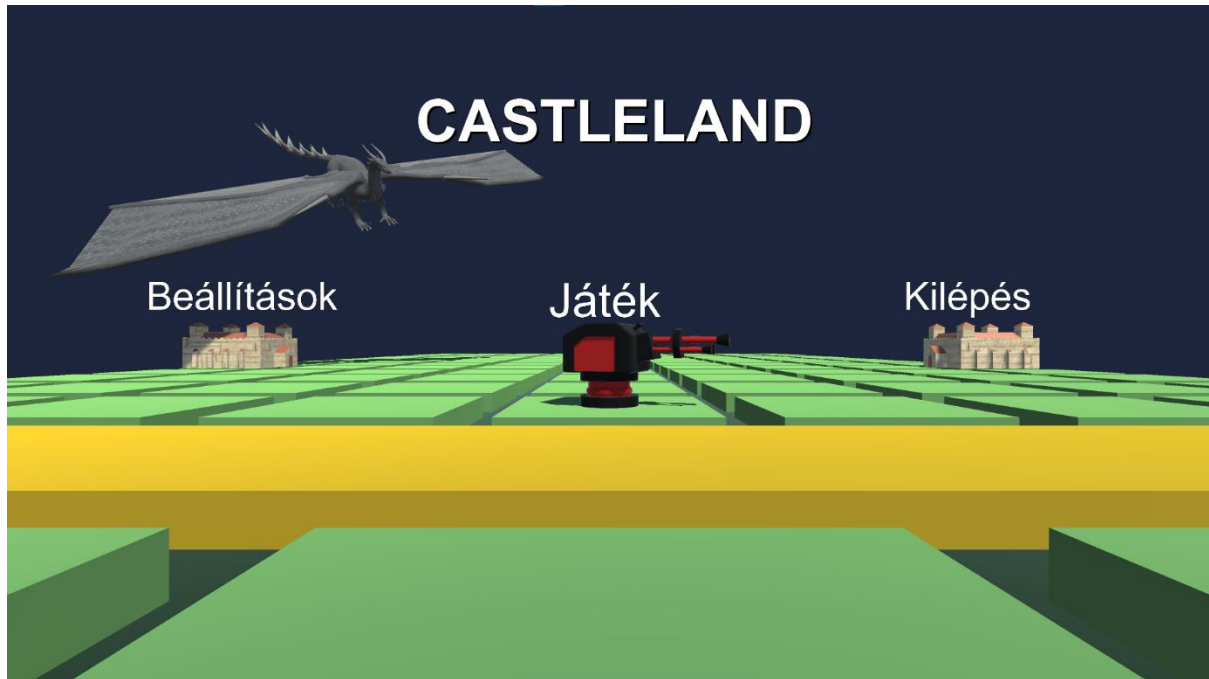
- Processzor: Intel Core i5-7400 vagy azzal egyenértékű
- Videókártya: Nvidia GeForce GTX 1650
- Tárhely: 150 MB szabad tárhely
- Memória: 150 MB szabad memória
- Operációs rendszer: Windows 10

### 2.3 Telepítés menete

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 megkezdí 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. Ha nincs különösebb okunk rá, hogy ezt megváltoztassuk, hagyjuk meg az alapértelmezett mappát, és kattintsuk a „Next” feliratú gombra a folytatáshoz. Amennyiben viszont megszeretnénk változtatni a tallózás ikonra kattintva ezt megtehetjük. Itt válasszuk ki az általunk preferált telepítési mappát, majd kattintsunk a „Next”-re. 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 katinthatunk megint a „Next” gombra. Ezután 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, ismét a „Next” gombbal léphetünk tovább. Ezután az „Install” gombra kattintva telepíthetjük a szoftvert. Itt 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 telepítésből. Ha a folyamat során bármikor meg szeretnénk szakítani a telepítést, azt a „Cancel” gombra kattintva tehetjük meg.

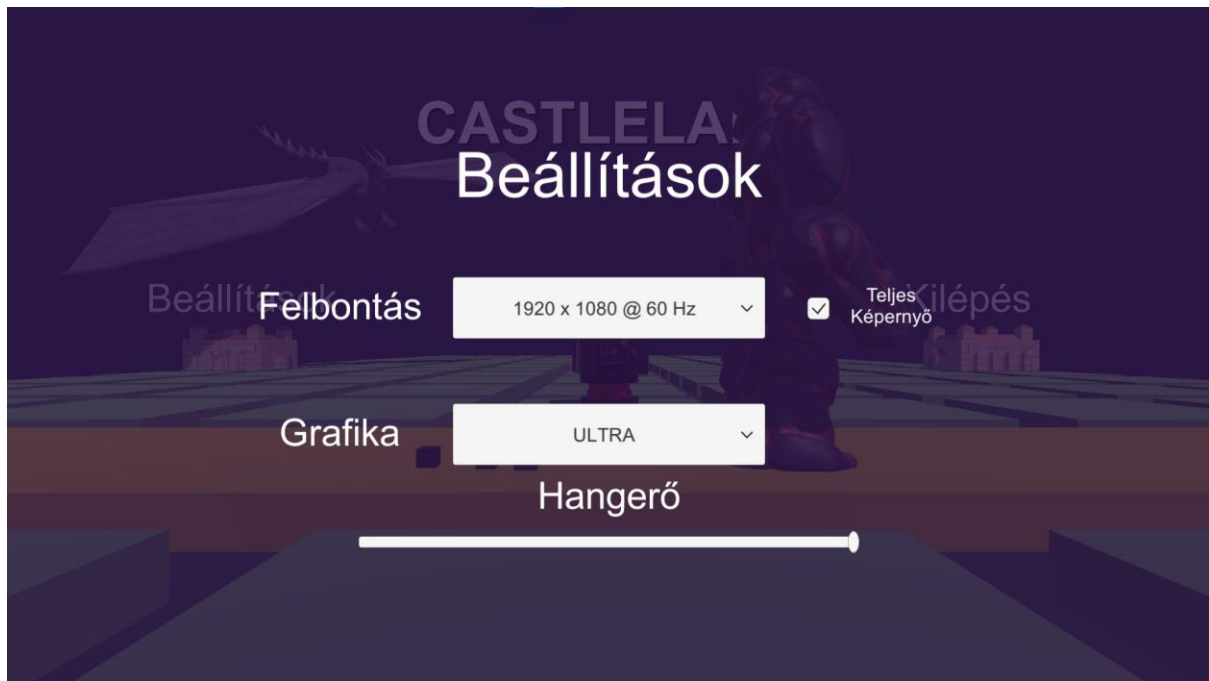
### 2.4 Egy tipikus futtatás kinézete

A játék indításakor egy fő menü fogad minket, három opcióval: „Játék”, „Beállítások”, „Kilépés”. A „Játék” gombra kattintva a felhasználó egy új játékmenet indul.

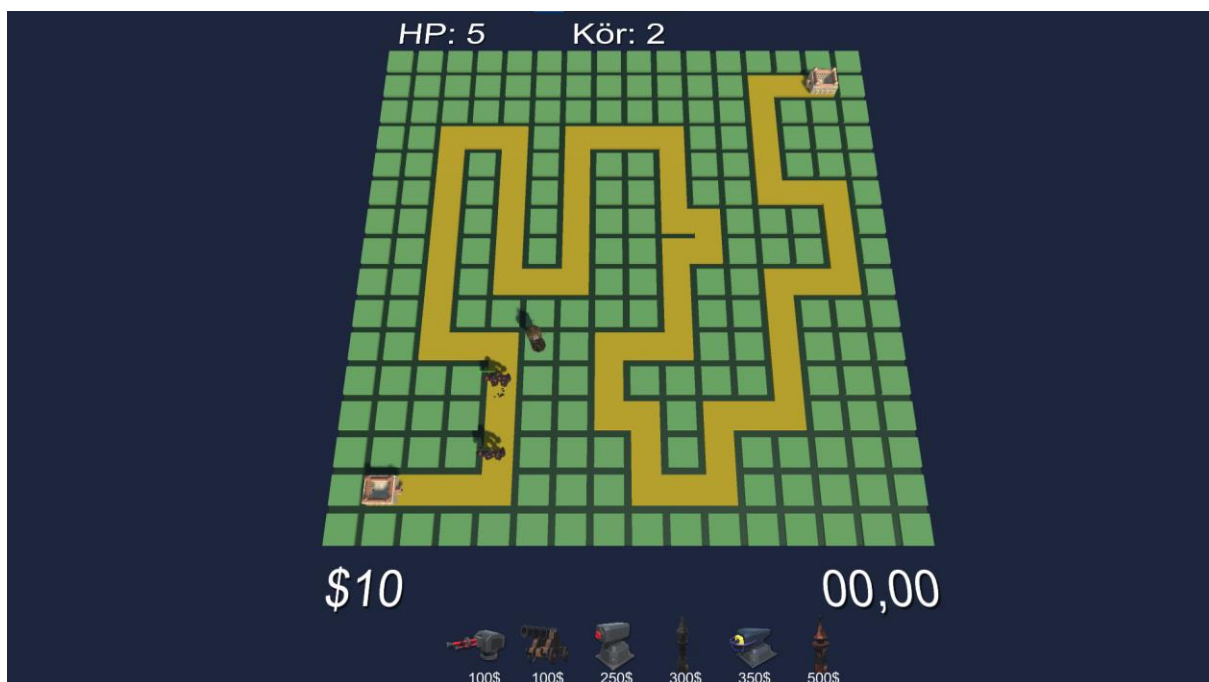


A fő menü másik opciójára, a „Beállítások” -ra kattintva három egyszerű beállítást láthatunk. Az első beállítás a „Felbontás”. Ennek segítségével a játéklablak felbontását módosíthatjuk, hogy kitöltse a teljes képernyőnket. Mellette található a „Teljes képernyő” jelölőnégyzet, mellyel beállíthatjuk, hogy ablakos, vagy teljes képernyős módban fusson a játék. Alatta egy lenyíló lista található, ahol a grafikai megjelenés minőségét állíthatjuk. Low-on az árnyékok betöltési távolsági lerövidül és az effektek minősége is romlik, viszont a teljesítmény javul. Ezután minden opció emel a betöltési távolságokon és az effektek minőségén, viszont növeli a processzor és videókártya használatot. Végül egy csúszkát láthatunk „Hangerő” felirattal, mellyel a játék hangerejét módosíthatjuk, százalékosan. Fontos, hogy az itt beállított hangerő nem csak a menüre, hanem a játékra is vonatkozik.



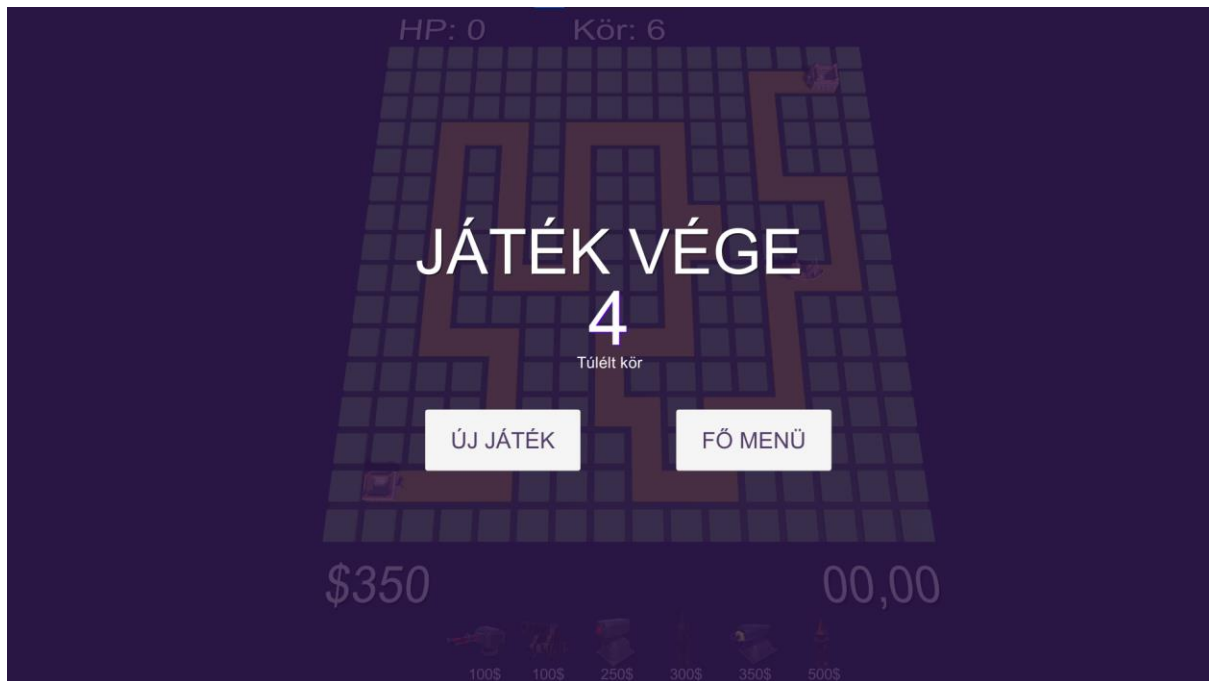


Ha a beállításokat elvégeztük az „Escape” gomb lenyomásával az előző menübe juthatunk vissza. Játék közben is hasonló menüt érhetünk el az „Escape” billentyű megnyomásával. Az itt megtalálható „Új Játék” felirat felett egy „Folytatás” gomb található, mellyel folytathatjuk a játékot, és az „Fő menü” feliratú gomb segítségével a fő menübe jutunk vissza. Amennyiben egy játékmenetet elindítunk, azonnal a játék teljes egésze a szemünk elé tárul. A játék kezelése nagyon egyszerű.



## 2. Felhasználói dokumentáció

A képernyő alján megtalálható boltból tudunk védelmi egységeket kiválasztani és ha arra elegendő pénzzel rendelkezik a játékos, akkor a pálya egyik mezőjére kattintva tudja azt lehelyezni. Ha a kiválasztás megtörtént egy mező fölé húzva a kurzort jelzi a felhasználónak a játék, hogy az elhelyezhető-e. Amennyiben elegendő pénzkerettel rendelkezik és jelenleg nincs rá építve épület akkor a mező színe élénk világos zöldé változik. Ha viszont ezen feltételek egyike nem teljesül, akkor a mező színe pirossá változik ezzel visszajelezve a játékosnak, hogy jelenleg oda nem tudja elhelyezni a kiválasztott épületet. Mindegyiknek megvan a maga erőssége és némelyik különböző ellenfelek ellen hasznosabb lehet. Később a már lehelyezett egységekre kattintva felettük megjelenik egy „Eladás” gomb melyre kattintva az egység vételárának felét visszakapja, és az megsemmisül. A képernyő valamelyik szélére húzva a kurzort vagy a WASD gombok lenyomásával, a kamera az adott iránynak megfelelően elmozdul, így teret adva a felhasználónak, hogy jobban átláthassa a világot, a szóköz gombot lenyomva pedig a kamera alaphelyzetbe állítható. Fent látható a jelenlegi körszám, illetve az életerők száma. Bal alul látható a pálya alatt a jelenlegi keretünk, amelyből egységeket tudunk vásárolni. A kör számának megfelelő mennyiségű ellenfél jön, és az ellenfelek típusai is változhatnak körönként. Az ellenfelek méretben, kinézetben, sebességben és életpont számban is különböznek. A játékos a kedvére gyorsíthatja, vagy lassíthatja a játékot bizonyos kereteken belül, hogy mindegyik ellenféllel eltudjon bánni, viszont ne legyen túl lassú sem számára a játékmenet. Amennyiben egy ellenfél eljut a mi várunkhoz (amelyet feladatunk megvédeni), akkor az az egység megsemmisül, viszont egy életerőt elveszít a játékos. Ha minden életerejét elveszti a játékos akkor a vára megsemmisül és a játéknak vége. Ilyenkor felugrik egy Game Over menü, de a játék nem áll meg, így egyfajta háttérrel biztosítva a menünek.



A menün megtalálható a túlélő körök száma, egy „Újrakezdés” gomb, illetve egy „Fő menü” gomb. Az „Újrakezdés” gombra kattintva természetesen egy teljesen új játékmenet indul, azaz az ellenséges egységek és felhasznált védelmi eszközök megsemmisülnek, a játékos pénzkerete, életeinek száma, a következő kör kezdetét jelző visszaszámláló alaphelyzetbe áll. A „Fő menü” gomb segítségével pedig a már ismerttetett kezdő képernyő fogadja a felhasználót.

## 2.5 Ellenfelek típusai

A játék szerves részét képezik a különböző ellenfelek, számuk mindig megegyezik az aktuális körszámával, és mindegyik körben megvan határozva mennyi jön a bizonyos típusú ellenfelekből. A kihívás miatt elengedhetetlen, hogy egyre többen és egyre erősebbek legyenek a játék későbbi szakaszában.

### 2.5.1 Láva gólem

A játék első ellenfele. Közepes sebességgel rendelkezik, viszont kevés életerővel, így egy ideális kezdő ellenfelet testesít meg. Egy torony is könnyen elbánik vele. Megsemmisülése esetén kevés értéket biztosít.

### 2.5.2 Hajó

A játék első második ellenfele. Közepes sebességgel rendelkezik, viszont a láva gólemnél több életerővel, így valamelyest nehezebb megsemmisíteni. Egy torony is könnyen elbánik vele, amíg egyedül van. Viszont amikor többen vannak már szükséges egy térsebző egység használata vagy több torony együttes ereje. Halála után kevéssel, de több értéket biztosít, mint a láva gólem.

### 2.5.3 Légiós

A játék harmadik ellenfele. Lassú a sebessége, melyet térsebző torony segítségével érdemes kihasználni. Több az életerője mint az előbbi ellenfeleknek, így erre érdemes figyelni. Közepes pénzmennyiség jár a sikeres likvidálásáért.

### 2.5.4 Kő gólem

A játék negyedik ellenfele, egyben a leggyorsabb. Érdemes ellene bevetni a lézervető ágyút mielőtt elérné a játékos várát. A légiósnál kevesebb míg a hajónál és láva gólemnél több életerővel rendelkezik. Közepes pénzmennyiség jár a sikeres likvidálásáért.

### 2.5.5 Fekete lovag

A játék ötödik ellenfele és egyik legnehezebben megölhetője. Viszonylag gyors mozgási sebességgel rendelkezik és az előbbi négy ellenfélénél nehezebb legyőzni. A játék ezen pontján érdemes térsebző tornyokat és lézervető ágyúval egyszerre levédekezni. Halála után a játék nagy pénzmennyiségben részesíti a játékost.

### 2.5.6 Sárkány

A játék hatodik és egyben utolsó ellenfele. A játék aktív részében egyszer tűnik fel az ötvenedik körben. Közepes sebességű viszont a többi egység életerője elenyésző az övéhez képest. A játék célja a sárkány elpusztítása, így halála esetén a játéknak vége és az erről informáló menü megjelenik. A háttérben a körök továbbmennek, így egyfajta háttérrel biztosít a menünek.

	Sebesség	Életerő
Láva gölem	20	100
Hajó	10	200
Légiós	8	500
Kő gölem	35	200
Fekete lovag	15	1000
Sárkány	10	10000

## 2.6 Tornyok/védelmi egységek típusai

### 2.6.1 Alap torony

Kis hatótávú egység mely a játék elején a leghasznosabb. Rövid hatótávja miatt kulcsfontosságú a megfelelő pozícióban történő elhelyezése. Alacsony sebzése van viszont olcsó megvásárolni.

### 2.6.2 Ágyú torony

Hatalmas hatótávolságú egység mely a játék minden pontján bevethető. Célszerű a pálya külső részein elhelyezni, mivel nagyobb a hatótávja mint a többi egységnek, hátránya a nagyon alacsony sebzés viszont alacsony áron elérhető, mely akár ideális kezdő egységnek felhasználható így.

### 2.6.3 Rakéta vető

Közepes hatótávolságú egység mely kis területű térsebzéssel rendelkezik. A játék korai szakaszában ajánlott legfőbbképpen a használata. Előnye a viszonylag magas sebzés viszont ez lassú töltési idővel, és lassú rakéta sebességgel társul. Közepes áron elérhető.

### 2.6.4 Íjász torony

Nagy hatótávolságú, gyorsan tüzelő egység. A játék közepe és vége felé ajánlott a használata. A legnagyobb másodpercenkénti sebzéssel rendelkezik egy ellenfélre vetítve. A gyors tüzelés ára a viszonylag alacsony sebzés lövedékeként és egyszerre csak egy ellenfelet tud sebezni. Közepes áron elérhető.

### 2.6.5 Lézervető ágyú

Közepes hatótávolságú, folyamatosan tüzelő egység. Különlegessége abban rejlik, hogy az éppen célzott ellenfelek mozgási sebességét lelassítja, így segítve a többi egység dolgát. Remekül összekombinálható valamilyen térsebzéssel rendelkező egységgel. Hátránya az alacsony másodpercenkénti sebzés és hogy csak egy ellenfelet céloz. Magas áron elérhető

### 2.6.6 Varázsló torony

Nagy hatótávolságú egység mely nagy területű térsebzéssel rendelkezik. A játék késői szakaszában szinte nélkülözhetetlen. Előnye, hogy egyszerre kettő, három vagy akár négy ellenfelet is képes sebezni a rakéta vetőnél alacsonyabb tüzelési idővel. Ajánlott a lézervető ágyúval kombinálni.

## 3. Fejlesztői dokumentáció

### 3.1 Témaválasztás indoklása

Főként amiatt választottam a játék készítést a weblap készítéssel szemben, mivel a későbbiekben is inkább az informatika-e terén szeretnék elhelyezkedni, és tapasztalatot szerezni. A jövőm tervezésével kapcsolatban, úgy gondolom Unity megfelelő alapot tud lefektetni számomra. A fejlesztői környezettel rengeteget foglalkoztam, önmagam képzésére használtam. Különböző projekteken keresztül megtapasztaltam, hogy mire képes a Unity, és teszteltem az általa kínált funkciókat. Illetve fontos volt számomra, hogy a baráti köröm is ismerte a motort, így kölcsönösen tudtunk egymásnak segíteni, amennyiben arra szükség volt. Mint már a bevezetőben említettem kiskorom óta a mindennapjaim részét képezik a videójátékok. A Tower Defense kategóriában nagy inspirációt merítettem többek között a Plant vs Zombies játékból, melyben az egységeknek nem csak a sebzésük tér el, hanem legtöbbjük rendelkezik különböző képességekkel is. Viszont mivel a 2D játékfejlesztésben kevesebb tapasztalattal rendelkezem, így inkább a háromdimenziós játékvilág mellett döntöttem. Ez alapján meghatározta a stílust és gyakorlatilag teljesen más játékmenet jött létre a fentebb említett videójátékhoz képest, de a nálam is született egység egyedi képességgel.

## 3.2 Fejlesztői környezet

Használt programozási nyelv: #C

„A C# a C programozási nyelv legfejlettebb változata, amelyet a mai napig fejlesztésekkel látnak el. A C# a Microsoft által a .NET keretrendszer részeként kifejlesztett objektumorientált programozási nyelv. A nyelv alapjául a C++ és a Java szolgált. A cél egy olyan nyelv megalkotása volt, mellyel teljesen objektumorientált módon lehet a keretrendszer komponenseit fejleszteni.” [1]

A választásban döntő szerepet játszott az, hogy már évek óta ezt a programozási nyelvet tanulom, így sokkal jártasabb vagyok benne, és könnyebben megtudom valósítani az elképzeléseimet. Fontos volt számomra az objektumorientáltság, mivel ennek használatával a fejlesztést sokkal gyorsabban és célratörőbben elvégezhettem.

Használt fejlesztési környezet: Unity

„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. Többek között előnye, hogy a szoftver képes nagyméretű adatbázisokat kezelni, kihasználni a kölcsönhatások és animációk képességeit, előre kiszámított vagy valós idejű világítást biztosítani.,,

„Továbbá használható geometriai eszközcsomagok továbbítására, illetve viselkedési elemek hozzáadására egyes objektumokhoz. Ezek mellett a játékmotor folyamatosan megőrzi a végleges változat megjelenítését., [2]

Itt a választás számomra egyértelmű volt. Habár fejlesztési tapasztalatom nem sok volt benne, de egy rendkívül fejlett játékmotornak számít a piacon, amelyet rengeteg fejlesztő használ, így rengeteg segédanyag is elérhető hozzá. Illetve támogatja a #C programozási nyelvet és a Microsoft Visual Studio-t mint szerkesztőt. A fejlesztői környezet sok beépített, előre elkészített eszközt biztosít, amelyek megkönnyítik a fejlesztés menetét. Emellett részletes hibaüzeneteket kapunk, amennyiben hibát érzékel és információt biztosít annak pozíciójáról is. Továbbá a használata mindenki számára ingyenes, ameddig a bevétel a készítőnek nem haladja meg az éves 100.000 dollárt.



### 3. Fejlesztői dokumentáció

Használt kódszerkesztő: Microsoft Visual Studio

„A Visual Studio a Microsoft több programozási nyelvet tartalmazó fejlesztőkörnyezete, amely az évek során egyre több új programnyelvvvel bővült. Jelenleg a F#, C++, C és Visual Basic programozási nyelveket, valamint az XML-t támogatja.” [3]

Mivel az oktatás során ezzel a programmal ismerkedtem meg, így erre esett a választásom. Fontos, hogy a telepítése közben szükséges az Unityhez szükséges fejlesztési segédeszközök modul telepítése is. A kód könnyen olvashatóságát különböző színekkel való kiemelésekkel biztosítja.

Használt képszerkesztő: GIMP

„A GIMP (GNU Image Manipulation Program) egy bittérképes képszerkesztő program. Támogatja a rétegek kezelését, az átlátszóságot. A GIMP az összes fontosabb platformra (Windows, Linux, MacOS, BSD, Solaris) létezik.” [4]

A kiválasztásnál fontos volt számomra, hogy képes legyen a program több fajta képformátumot megnyitni és kezelni azokat (pl. gif, png, jpeg). Emellett beépített réteg és csatorna kezelővel rendelkezik, melynek köszönhetően a képeket könnyedén testre tudjuk szabni. Továbbá képátalakítási lehetőségek is elérhetőek benne, mint pl. a tükrözés, forgatás, méretezés. Főként a tornyok képeit készítettem el a segítségével.

Használt modellező program: Blender

„A Blender egy szabad, nyílt forráskódú, háromdimenziós grafikai program. Felhasználható animációs filmek, vizuális effektek létrehozására, használják művészek, alkalmazható 3D nyomtatásra, interaktív 3D alkalmazások és videojátékok létrehozására. A Blender számos operációs rendszerre elérhető: Microsoft Windowsra, Mac OS X-re, GNU/Linuxra, IRIX-re, Solarisra, NetBSD-re, FreeBSD-re, OpenBSD-re.” [5]

Főként a választásom amiatt esett a Blenderre, mert ingyenesen elérhető és rendelkeztem már tapasztalattal benne. Kellő részletességgel tudjuk a kívánt 3D objektumokat lerészletezni a benne megtalálható eszközökkel.

### 3. Fejlesztői dokumentáció

## 3.3 Főbb algoritmusok



Ezek az osztályok a játék központi elemeiért felelnek. Ilyen pl. új kör indítása, tornyok építése, lövés, ellenfelek mozgása és forgása.

### 3. Fejlesztői dokumentáció

#### 3.3.1 UjKor

Feladatai közé tartozik egy új kör indítása, amennyiben a pályán lévő ellenfelek elfogytak és a körök közötti időzítő lejár.

```
public Transform EllenfelEgy;  
public Transform EllenfelKetto;  
public Transform EllenfelHarom;  
public Transform EllenfelNegy;  
public Transform EllenfelOt;  
public Transform EllenfelHat;  
public Transform EllenfelHet;  
public Transform KezdoPont;
```

A képen látható változókból az első hét a különböző ellenfelek prefabjeit tárolja, míg a nyolcadik az ellenfelek kezdőpozícióját.

```
public static int ellenfelSzam;  
public static int aktualisKor;  
public static bool spawnVege;
```

A következő képen három statikus változót találhatunk, ezen változók fontosak voltak, hogy példányosítás nélkül is elérhetőek legyenek más osztályokban így szükséges volt a statikussá tételük. Az első a jelenleg pályán tartózkodó ellenfelek számát tárolja, a második az aktuális kör számát, míg a harmadik változó azt tárolja, hogy a körszámnak megfelelően minden ellenfél leéledt-e már. A spawnVege változó amiatt volt szükséges, mivel az ellenfelek leéledése között késleltetés található és ha a felhasználó sikeresen elpusztította az összes már leéledt ellenséges egységet mielőtt a körnek megfelelő számú ellenfél mennyiség leéledt, akkor az időzítő elindult és egy új kör indult, így két körnek megfelelő mennyiségű ellenfél jött rövid időn belül.

### 3. Fejlesztői dokumentáció

```
void Update()
{
    if(ellenfelSzam>0 || !spawnVege)
    {
        return;
    }

    if (idozito <= 0f)
    {
        StartCoroutine(KorInditas());
        spawnVege = false;
        idozito = korokKozottiIdo;
        return;
    }

    idozito -= Time.deltaTime;
    idozito = Mathf.Clamp(idozito, 0f, Mathf.Infinity);
    idozitoSzoveg.text = string.Format("{0:00.00}", idozito);
}
```

A képen megtalálható Update metódus ellenőrzi, hogy éppen folyamatban, van-e egy kör. Amennyiben éppen nincs folyamatban lévő kör akkor a következő kör kezdetét jelző számlálót elkezd csökkenti. Ha ez lejárt akkor meghívja a KorInditas nevezetű metódust.

```
IEnumerator KorInditas()
{
    aktualisKor++;
    ellenfelSzam = aktualisKor + 1;
    for (int i = 0; i <= aktualisKor; i++)
    {
        UjEllenfel(i);
        yield return new WaitForSeconds(0.5f);
    }
    spawnVege = true;
}
```

A fent látható funkció az újabb kör indításáért felel, eggyel növeli a körök számát, beállítja az aktuális ellenfelek számát, majd a körszámmal megegyező mennyiségű ellenfelet éleszt le az UjEllenfel metódus segítségével. Minden ellenfél leéledése között fél másodperc késleltetés található.

### 3. Fejlesztői dokumentáció

```
void UjEllenfel(int ellenfelid)
{
    if (ellenfelid<5)
    {
        Instantiate(EllenfelEgy, new Vector3(7f, 2.25f, 5f), KezdoPont.rotation);
        return;
    }
}
```

Az UjEllenfel metódus az adott ellenfél id-ja alapján egy meghatározott prefabjú ellenfelet éleszt le, a megadott kezdőhelyen annak irányával megfelelően. Fentebb csak az első öt ellenfélnek megfelelő kódrészlet található meg.

### 3. Fejlesztői dokumentáció

#### 3.3.2 Utvonalkövet

```
public float kezdosebbsseg = 10f;

[HideInInspector]
public float sebbsseg;

public Transform forogResz;
public float forgassebbsseg = 100f;

public float kezdoeletero;
public float eletero = 100;
public int ertek;

private Transform kovetkezo;
private int jelenlegi = 0;

public Image EletCsik;
```

Az osztály az ellenfelek mozgásáért, forgatásáért, sebződéséért és megsemmisüléséért felel. Az osztály tetején a felhasznált változók találhatóak meg. Az első két változó az adott ellenfél sebbségéért felel, az utána következő két változó az ellenfél forgásához szükséges adatokat tárolja. Utána az ellenfél kezdeti életereje és aktuális életereje található, ezt követi az ellenfél megsemmisítéséért járó pénzösszeg mennyisége. Az ezek alatt megtalálható két változó felel az ellenfél pozíciójának és haladási irányának meghatározásához, míg a legalul az objektum életcsikjának a képe található, amelynek kitöltöttsége a kezdeti és a jelenlegi életerejétől függ az ellenfélnek.

```
void Update()
{
    Vector3 irany = kovetkezo.position - transform.position;
    transform.Translate(irany.normalized * sebbsseg * Time.deltaTime, Space.World);

    if (Vector3.Distance(transform.position, kovetkezo.position) <= 0.5f)
    {
        UjIrandy();
    }

    sebbsseg = kezdosebbsseg;
}
```

Az Update metódus felel az ellenfél megfelelő irányba mozgatásáért. Itt fontos megjegyezni, hogy a mozgás az előre megadott ellenőrzőpontok irányába történik, ezek minden kanyarban találhatóak meg. Ha az ellenfél elér egy ellenőrző pontot akkor meghívja az UjIrandy metódust. A sebbségét az egységeknek mindig visszaállítja a kezdősebbségükre, így biztosítva, hogy ha a lézerágyú hatótávjából kiérnek, akkor újra felgyorsulnak.

### 3. Fejlesztői dokumentáció

```
void UjIrandy()
{
    if (jelenlegi >= IranyjelzokScript.iranyjelzok.Length - 1)
    {
        Megsemmisul();
        if(Valuta.HpSzam!=0)
        {
            Valuta.HpSzam--;
        }
        return;
    }
    Vector3 iranyForog = transform.position - IranyjelzokScript.iranyjelzok[jelenlegi + 1].position;
    forogResz.rotation = Quaternion.LookRotation(iranyForog);
    jelenlegi++;
    kovetkezo = IranyjelzokScript.iranyjelzok[jelenlegi];
}
```

Az UjIrandy függvény megnézi, hogy az ellenfél elérte-e a pálya végét, amennyiben elérte azt, az egység megsemmisül és csökkenti a játékos életeinek számát. Különben kiválasztja a következő ellenőrző pontot és annak megfelelő irányba forgatja az ellenfelet.

```
public void Sebzodes(float sebzesMennyiseg)
{
    eletero -= sebzesMennyiseg;

    EletCsik.fillAmount = eletero/kezdoEletero;
    if(eletero<=0)
    {
        Megsemmisul();
    }
    return;
}
```

A Sebzodes metódus a megkapott paraméternek megfelelően csökkenti az ellenfél életerejét, és az életcsíkon megjeleníti azt, ha az életerő nullánál alacsonyabbra csökken, akkor meghívja a Megsemmisul függvényt.

```
void Megsemmisul()
{
    Destroy(gameObject);
    Valuta.Penz += ertek;
    UjKor.ellenfelSzam--;
}
```

Meghívásakor az ellenséges egységet megsemmisíti, annak értékét hozzáadja a játékos pénzéhez és csökkenti a körben aktívan jelenlevő ellenfelek számát.

```
public void Lassit(float mertek)
{
    sebbeseg = kezdoSebbeseg * (1f - mertek);
}
```

A paraméterrel arányosan csökkenti az ellenfél aktuális sebbeségét.

## 3.4 Továbbfejlesztési lehetőségek

Továbbfejlesztési lehetőségek találhatóak a játékban még bőven. Talán a legfontosabb továbbfejlesztési lehetőség a játék optimalizálása lenne, mivel a játék típusához képest a gépigénye túlságosan nagy. A valós idejű árnyékok lecserélése az olyan objektumoknál, amelyek a játék közben nem mozognak jó eséllyel javítana a teljesítményen. A processzor terhelését lehetne azzal csökkenteni, hogy az árnyékok pozíciójait a videokártya számíttaná ki, ez azt eredményezné, hogy a processzorhasználat sokat esne vissza, de cserébe a videokártya terhelése pár százalékkal nőne. Viszont a világítástechnika egy rendkívül összetett rendszer, amelyet viszonylag könnyű elrontani, és ez esetben nem csak a fények minősége romlik, de a teljesítmény tovább csökkenhet. Ez egy létfontosságú fejlesztés lenne, mivel a magas hardverigények behatárolják a potenciális felhasználók körét és a fejlesztési lehetőségeket.

A következő fejlesztési lehetőség az optimalizáció után az ellenfelek meganimálása lenne. Ezt eredetileg szerettem volna a játékba belerakni, viszont tovább emelné a hardverkövetelményeket, amit elszerettem volna kerülni. Viszont a játék színvonalán, élvezhetőségén sokat emelne viszonylag kevés erőfeszítéssel.

A játék színvonalán tovább emelhetne egy bevezető, amely során a játék egységei és védelmi eszközei ismertetve lennének a felhasználónak, és nem kellene a felhasználói dokumentumra támaszkodnia, amennyiben információt akar ezekről találni. Ez a bevezető akár remek alap lehet egy történethez a játéknak, így plusz célt és élvezetet adva a játékosnak, emellett bővítené a potenciális célközönséget.

További fejlesztés lehetne, hogy több pálya kerüljön be a játékba, amelyek között a felhasználó vagy az egyik pálya teljesítésével, vagy a menüben kiválasztva tudna lépkedni. A meglévő komponensekkel ezt a feladatot viszonylag könnyű lenne kivitelezni, a demó verzióba az időhiány miatt nem került be.

A több pálya értelemszerűen több ellenféltypussal és védelmi egységgel járna együtt. A játék úgy lett megírva, hogy mindkettőt gyorsan és könnyedén belehessen implementálni. Az egyetlen akadálya ennek a megfelelő modellek hiánya volt.

Ezek mellett a játékban elérhető védelmi eszközöket az eladásukhoz hasonló módon lehetne akár tovább fejleszteni, így bizonyos érték ellenében tovább növelni azok erejét. Ez több variációs lehetőséget kínálna a játékosnak a játék teljesítésére, így növelne annak az újrajátszhatóságán is akár. Ennek a funkciónak az egyetlen akadálya, hogy a fejlesztett védelmi eszközök több modellt igényelnének még, amelyeknek hiányát így is megéreztem.



## 3.5 Tesztelés

A játék elkészítésénél a funkciókat egyesével fejlesztettem le, és minden egyes elkészült funkciót többször teszteltem, mielőtt neki álltam volna a következőnek. A tervezett megvalósítandó funkciókat egy szöveges állományban lévő listán tároltam, így ha egy funkció elkészült és hibamentesnek találtam teszteléskor, akkor lekerült a listáról. Az észlelt hibákat egyből javítottam, hiszen minél nagyobb lett a program annál nehezebb lett volna őket kiszűrni. A hibákat visszakereséses elv alapján kezdtem el kijavítani, ez azt takarja, hogy azonosítottam azt a részt, ahol a probléma felmerült és lépésenként megnéztem, hogy jut el oda a program és mit, miért csinál. Mivel viszonylag rövid hosszúságú szkriptjeim vannak, így könnyű volt a kódban lévő hibák okát megtalálni. A nem kódban jelenlévő hibákat szerencsére legtöbbször a játékmotor kijelezte, és részletes információt biztosított annak hol létezik és okáról. Miután elkészült egy újabb funkció nem csak az adott funkciót igyekeztem tesztelni, hanem az addigi sikeres tesztek is megismételtem, így biztosítva azt, hogy az egyik funkció működése nem befolyásolja az addig helyesen működőket.

Amint a játék demója elkészült több ismerősömnek is odaadtam, hogy futtassák le és teszteljék. Legtöbbször pozitív visszajelzéseket kaptam egy kevés kritikával. A kritikák között olyanok szerepeltek, hogy az 1920 X 1080-nál kisebb felbontású kijelzőkön a kameraszegély mérete túl nagy volt, így a védelmi egységek kiválasztásánál a felhasználó akarata ellenére is elmozdult a kamera. Erre reagálva csökkentettem a szegély méretét, így biztosítva, hogy csak a játékos akaratával mozdulhasson el. A további észrevételek között szerepelt az, hogy a játék színvonalasabb lenne animációkkal, illetve a különböző egységeknek is lehetnének hangjai a zene mellett, ezeket sajnos nem volt időm megvalósítani. Illetve az egyik tesztelő ötlete volt az idő lassítása és gyorsítása és ez a funkció egy későbbi verzióban sikeresen belekerült a játékba.

## 3.6 Összegzés

A fejlesztés megkezdésénél azt a célt tűztem ki magam elé, hogy egy működőképes, játszható játékot fejlesszek ki. Mikor elkezdtem a projectet, akkor még viszonylag kevés tapasztalattal rendelkezttem, és bár csináltam már játék prototípust, de ezeket nem fejeztem be, ellentétben a CastleLand-del. A fejlesztés során rendkívül sok új ismeretre tettem szert, melyeket a későbbiekben is feltudok majd használni. A játékom megfelel a Tower Defense játékok koncepciójának, emellett összeségében sikeresen megvalósítottam a legtöbb elképzelésemet, így elégedett vagyok a végeredménnyel. Véleményem szerint a játék jelenlegi formájában is élvezhető, de a későbbiekben tovább tervezem fejleszteni, amennyiben a szabadidőm ezt engedni fogja, hogy elérje egy akár pénzért kiadható játék szintjét is, még akkor is, ha a kiadás maga nem szerepel a terveim között.

## 3.6 Források, szakirodalom

### Források

[1] [https://hu.wikipedia.org/wiki/C\\_\(programoz%C3%A1si\\_nyelv\)](https://hu.wikipedia.org/wiki/C_(programoz%C3%A1si_nyelv))

(Megtekintés: 2021. 12. 12)

[2] [https://hu.wikipedia.org/wiki/Unity\\_\(j%C3%A1t%C3%A9kmotor\)](https://hu.wikipedia.org/wiki/Unity_(j%C3%A1t%C3%A9kmotor))

(Megtekintés: 2021. 12. 12)

[3] [https://hu.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://hu.wikipedia.org/wiki/Microsoft_Visual_Studio)

(Megtekintés: 2021. 12. 12)

[4] <https://hu.wikipedia.org/wiki/GIMP>

(Megtekintés: 2021. 12. 12)

[5] [https://hu.wikipedia.org/wiki/Blender\\_\(program\)](https://hu.wikipedia.org/wiki/Blender_(program))

(Megtekintés: 2021. 12. 12)

### 3. Fejlesztői dokumentáció

#### Szakirodalmak

<https://docs.unity3d.com/Manual/index.html> (Megtekintés: 2021. 12. 12)

[https://www.youtube.com/watch?v=beuoNuK2tbk&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=beuoNuK2tbk&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&ab_channel=Brackeys) (Megtekintés: 2021. 12. 12)

[https://www.youtube.com/watch?v=aFxucZQ\\_5E4&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=2&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=aFxucZQ_5E4&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=2&ab_channel=Brackeys) (Megtekintés: 2021. 12. 12)

[https://www.youtube.com/watch?v=n2DXF1ifUbU&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=3&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=n2DXF1ifUbU&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=3&ab_channel=Brackeys) (Megtekintés: 2021. 12. 12)

[https://www.youtube.com/watch?v=QKhn2kl9\\_8I&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=4&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=QKhn2kl9_8I&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=4&ab_channel=Brackeys) (Megtekintés: 2022. 02. 15)

[https://www.youtube.com/watch?v=oqidgRQAMB8&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=5&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=oqidgRQAMB8&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=5&ab_channel=Brackeys) (Megtekintés: 2022. 02. 15)

[https://www.youtube.com/watch?v=t7GuWvP\\_IEQ&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=6&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=t7GuWvP_IEQ&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=6&ab_channel=Brackeys) (Megtekintés: 2022. 02. 15)

[https://www.youtube.com/watch?v=qokYflgxU6M&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=7&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=qokYflgxU6M&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=7&ab_channel=Brackeys) (Megtekintés: 2022. 02. 15)

[https://www.youtube.com/watch?v=uv1zp7aOoOs&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=8&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=uv1zp7aOoOs&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=8&ab_channel=Brackeys) (Megtekintés: 2022. 02. 16)

[https://www.youtube.com/watch?v=IMhRDTf9pBk&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=9&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=IMhRDTf9pBk&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=9&ab_channel=Brackeys) (Megtekintés: 2022. 02. 16)

[https://www.youtube.com/watch?v=LJLWNnqAjQ4&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=10&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=LJLWNnqAjQ4&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=10&ab_channel=Brackeys) (Megtekintés: 2022. 02. 16)

[https://www.youtube.com/watch?v=pZ0QyngaQv4&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=11&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=pZ0QyngaQv4&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=11&ab_channel=Brackeys) (Megtekintés: 2022. 02. 16)

[https://www.youtube.com/watch?v=VW90z41gd8Y&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=12&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=VW90z41gd8Y&list=PLjbp7P0bEfYo0C49X3xjde791kuIdEjF&index=12&ab_channel=Brackeys) (Megtekintés: 2022. 03. 12)

### 3. Fejlesztői dokumentáció

[https://www.youtube.com/watch?v=ZapFCWu0zk0&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=13&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=ZapFCWu0zk0&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=13&ab_channel=Brackeys) (Megtekintés: 2022. 03. 12)

[https://www.youtube.com/watch?v=5vw7RYxvqeE&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=14&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=5vw7RYxvqeE&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=14&ab_channel=Brackeys) (Megtekintés: 2022. 03. 12)

[https://www.youtube.com/watch?v=aezXrDwUDak&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=15&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=aezXrDwUDak&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=15&ab_channel=Brackeys) (Megtekintés: 2022. 03. 12)

[https://www.youtube.com/watch?v=UOYBr9vFqYI&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=16&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=UOYBr9vFqYI&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=16&ab_channel=Brackeys) (Megtekintés: 2022. 03. 12)

[https://www.youtube.com/watch?v=vnqRTqfk0Lo&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=17&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=vnqRTqfk0Lo&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=17&ab_channel=Brackeys) (Megtekintés: 2022. 03. 13)

[https://www.youtube.com/watch?v=Vn26oMNNpuE&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=18&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=Vn26oMNNpuE&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=18&ab_channel=Brackeys) (Megtekintés: 2022. 03. 13)

[https://www.youtube.com/watch?v=FM6Wn1fPBjk&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=20&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=FM6Wn1fPBjk&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=20&ab_channel=Brackeys) (Megtekintés: 2022. 03. 14)

[https://www.youtube.com/watch?v=789gM9R3htc&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=21&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=789gM9R3htc&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=21&ab_channel=Brackeys) (Megtekintés: 2022. 03. 14)

[https://www.youtube.com/watch?v=UKs1qO8w7qc&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=23&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=UKs1qO8w7qc&list=PLjbp7P0bEfYo0C49X3xjde791kumIdEjF&index=23&ab_channel=Brackeys) (Megtekintés: 2022. 03. 14)

## 3.7 Licenszek, felhasznált assetek

<https://devassets.com/assets/tower-defense-assets/>

<https://brainchildpl.gumroad.com/l/rQsZt>

<https://brainchildpl.gumroad.com/l/KkiLV>

<https://sketchfab.com/3d-models/simples-arrow-e3e3d4ae579a4df999eb32a69628c772>

<https://sketchfab.com/3d-models/archer-tower-07238c64d795488e9cdf932bb51e36e5>

<http://pixabay.com/music/main-title-battle-of-the-dragons-8037/>

<http://pixabay.com/music/main-title-honor-and-sword-main-11222/>

A felhasznált modellek mindegyike vagy Creative Commons licenszet használ (forrásmegjelölés esetén ingyenesen felhasználható, megosztható), vagy személyes használatra ingyenesen felhasználható. Mivel a játék nem lesz elérhető sehol, így ezeknek a követelményeknek megfelelnek.