

Záródolgozat

Paré Attila Sándor

2022

Spacium

Videójáték készítése Godot motor használatával.

Tartalomjegyzék

Bevezető	4
Felhasználói dokumentáció	5
Ismertető	5
A játék indítása	5
Minimális Rendszer követelmények:	6
Főmenü	6
Fiókadatok módosítása	7
Bejelentkezés	8
Regisztráció	8
Mentett adatok	10
Fejlesztői dokumentáció	11
Felhasznált eszközök	11
Godot játékmotor	11
Programozási nyelv	11
Godot definíciók	12
Scenes (jelenetek)	12
Nodes	12
Scene tree (jelenet fa)	13
Signals (jelek)	13
Program architektúrája	14
Használati eset diagram	15
Jelenetek	16
Menü gombok	17
Fiókkezelő ablak	18
Beállítások ablak	20
A játék jelenet	22
A nap:	23
A hajók	24
A bolt	25
A hajók menüpont	26
Adatok tárolása	27
Összegzés	28
Továbbfejlesztési lehetőségek	28
Ábrák Jegyzéke	29

Bevezető

A videojáték egy kétdimenziós erőforrás kezelő játék, amelynek témája az úr. Emiatt a téma miatt választottam a játéknak a Spatium nevet, ami latinul azt jeleni, hogy úr. A videojáték a Godot 3.4.4 motor Mono verzióját használja és C# (ejtsd: C Sharp) programozási nyelven vettem igénybe.

A játék a casual játék („alkalmi játék”) műfajának alfajába tartozik a resource management („erőforrás-kezelő”).

A Casual játékok általában szórakoztató és egyszerű játékmenet biztosítanak ehhez társul még a könnyen használható felhasználói kezelőfelület, amit a felhasználó akár egy egér gombbal is lehet irányítani. Rövid játékmenetet biztosít, amit akár munkaszünetekben, tömegközlekedésen vagy várakozás közben is lehet játszani. Emellé társul a nyugtató Lo-Fi zene hogy erősítse a stresszoldó hatását a műfajnak.

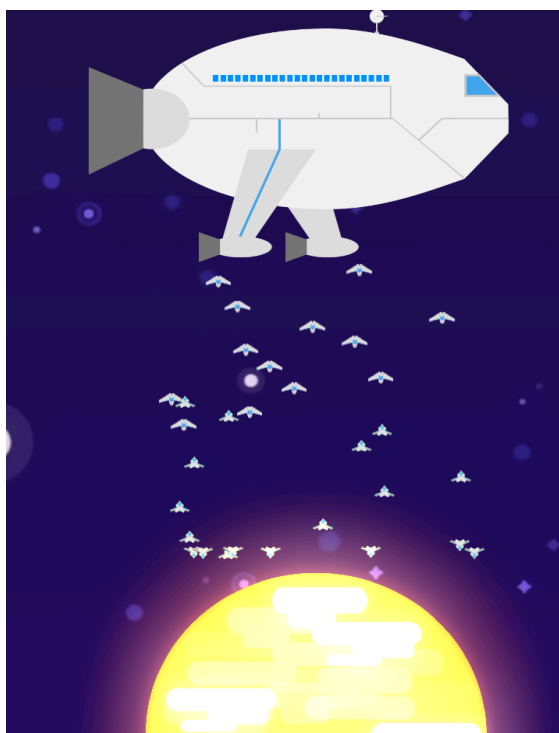
Az erőforrás kezelő alműfaj, ahol különböző erőforrásokat tudsz gyűjteni (például: pénz, energia, nyersanyagok, terület, emberi erőforrás, tokenek) abból a célból, hogy minél hatékonyabban és minél több mennyiségben lehessen erőforrást szerezni.

Mikor arra jutott a választás, hogy melyik videojáték motort használjam akkor, az Unity vagy Godot Mono között volt választásom. A videojáték motor egy olyan szoftver fejlesztői környezet, amiknek a funkciói elősegítik a videojáték gyorsabb fejlesztését, a fő funkciói közé tartozik a kettő vagy háromdimenziós grafikai megjelenítés, fizikai motor, animációk kezelése és készítése, hangok kezelése vagy akár a mesterséges intelligencia készítése például „Ellenséges robotok a játékos ellen”. Mindkét motor egyszerű használata és funkciói megfelelték volna ehhez a projekthez. De végül a Godot Mono-ra esett a választásom mivel, a motor kisebb méretű, a fejlesztői felület könnyebben kezelhető és jobban preferálom.

Felhasználói dokumentáció

Ismertető

A játék egy kétdimenziós erőforrás kezelő játék. Ez egy egyszemélyes játék, ahol a játékos célja, hogy minél több energiát szerezzen ezt úgy tudja megtenni, hogy az anyahajóval naptól napig ugrál. Mivel a ugrás rengeteg energiába kerül és túl sok idő lenne egy hajóval összegyűjteni a megfelelő mennyiségű energiát, a játékos készít különböző fajta hajókat. Miután elér egy energiaszintet akkor ugrik az anyahajó a következő csillaghoz. Minden egyes ugrás több és több energiába kerül a sérült reaktor miatt. 5 különböző fajta nap van mindegyiknek saját tulajdonságaival, amit ki kell tapasztalnia a felhasználónak. A játéknak akkor van vége, ha a felhasználó meglátogatja mind az 5 napot, de utána még lehet folytatni a játékot.



1. ábra Játékmenet

A játék indítása

A játékot fel kell telepíteni a „Setup.exe” futtatásával. Miután kiválasztotta a felhasználó a telepítési helyet a felhasználó kérheti hogy, készítsen a telepítő parancsikont az asztalon. Várnunk kell míg feltelepíti a szükséges modulokat. Sikeres telepítés esetén dupla kattintva a „Spacium.exe” parancsikonra akkor elindul a játék.

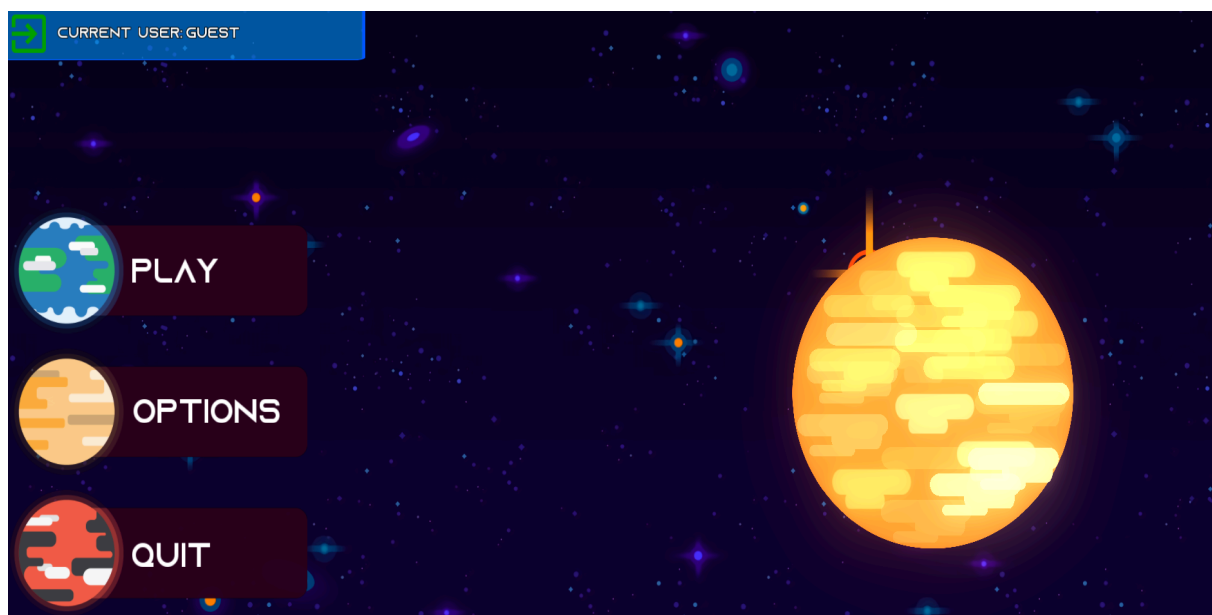
Minimális Rendszer követelmények:

- Operációs rendszer: Windows 8+
- Videó kártya: támogatja a DirectX 9 csomagot
- Processzor: 64x architektúra és SSL2 támogatja
- Memória: minimum 2GB
- Tárhely: 600MB+

Főmenü

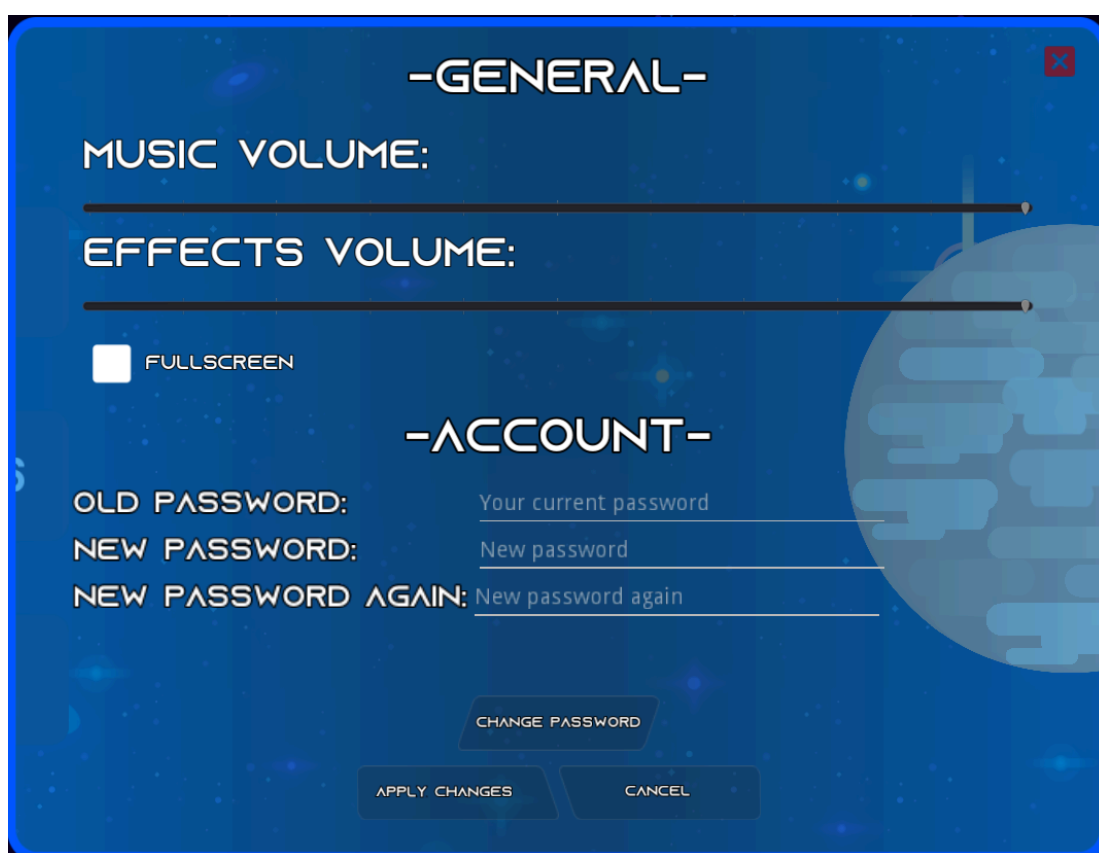
Indítás után a játék a főmenübe irányítja a játékost, ahol 4 gomb található.

A „Play” gombra kattintva tovább lép a játékba, ha a felhasználó be van jelentkezve, ha nincs bejelentkezve, akkor megjelenik a bejelentkezés és regisztrációs ablak. Az „Options” gombra kattintva megjelenik a beállítások, ahol az alapbeállításokat és a felhasználói fiókot tudjuk módosítani. A „Quit” gombra kattintva az alkalmazás bezárul.



Beállítások

A beállítás ablak két részből áll, általános („General”) beállítások és fiók („Account”) beállításból. Az általános beállításokban található 2 db csúszka, amivel a zene („Music”) és a hangeffektusoknak („Sound effects”) a hangerejét lehet csökkenteni vagy növelni. Emellett még található egy jelölőnégyzet, amivel azt lehet váltani, hogy a játék teljes képernyős vagy ablakos módban legyen. A fiókbeállításokban lehetősége van megváltoztatni a jelenleg használt felhasználó fiók jelszavát. A jelenlegi jelszó („Current password”) beviteli mezőbe kell beírni a jelenleg használt jelszót. A új jelszó („New password”) és a új jelszó újra mezőbe („New password again”) be kell írni a új jelszót a megváltoztatásához.



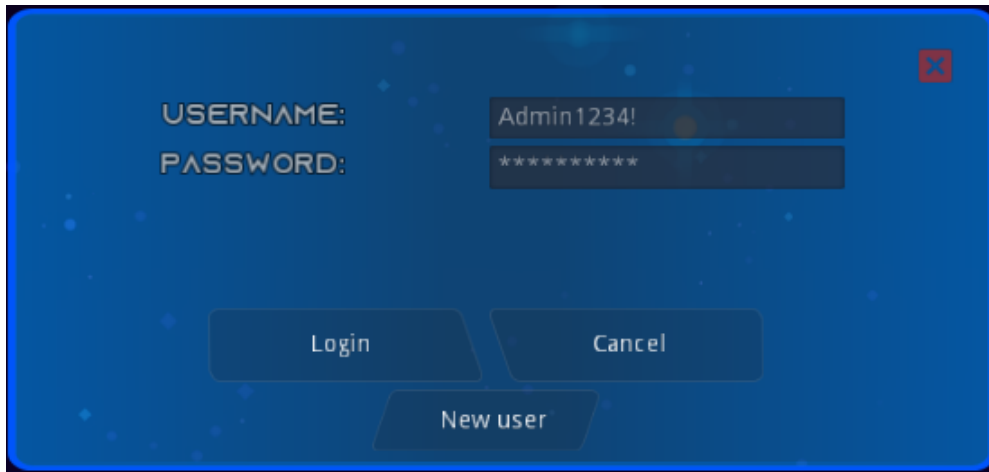
3. ábra Beállítások

Fiókadatok módosítása

A beállításokban található 3 beviteli mezőbe jelenlegi jelszót, új jelszó, új jelszó újra be kell, írni megfelelő adatokat majd ezek után rá kell kattintani a jelszó változtatás („Change password”) gombra.

Bejelentkezés

A bal felső sarokban lévő jelentkezés ikonra nyomva megjelenik a fiókkezelés ablak, ahol be lehet jelentkezni vagy új fiókot készíteni. Bejelentkezéshez be kell írni a játékos felhasználónevét („Username”) és jelszavát („Password”) ezek után rá kell nyomni a jelentkezés gombra („Login”). Ha sikeres volt a jelentkezés akkor egy üzenet fog megjelenni a képernyő közepén.

The image shows a login interface with a dark blue background featuring a subtle pattern of light blue stars and dots. In the top right corner, there is a small red square icon with a white 'X'. The interface includes two input fields: the first is labeled 'USERNAME:' and contains the text 'Admin1234!'; the second is labeled 'PASSWORD:' and contains ten asterisks '*****'. Below these fields are three buttons: 'Login' and 'Cancel' are positioned side-by-side, and 'New user' is centered below them. All text and buttons are in a light blue or white color.

4. ábra Bejelentkezési felület

Regisztráció

A fiókkezelés ablakban a új felhasználó („New user”) gombra kattintva átvált a fiók-létrehozó módba. Itt be kell írni az Email címet, felhasználónevet, jelszót, jelszót megerősítést. Ezek után megnézi hogy foglalt-e a felhasználónév és megegyeznek a jelszavak. Sikeres jelentkezés esetén megjelenik egy üzenet hogy „Sikeres regisztráció!” ellenkező esetben pedig „Sikertelen regisztráció: [hibakód]!” jelenik meg.

The image shows a registration form with a blue background and a dark blue border. In the top right corner, there is a red square button with a white 'X' icon. The form contains four input fields with labels to their left: 'EMAIL:' with the value 'Admin1234@gmail.com', 'USERNAME:' with the value 'Admin1234!', 'PASSWORD AGAIN:' with the value '*****', and 'PASSWORD:' with the value '*****'. Below these fields are three buttons: 'Register' and 'Cancel' are side-by-side, and 'Existing user' is centered below them.

EMAIL:	Admin1234@gmail.com
USERNAME:	Admin1234!
PASSWORD AGAIN:	*****
PASSWORD:	*****

Register Cancel

Existing user

5. ábra Regisztrációs felület

Bolt

A játékban található minden egyes hajónak saját bolt felülete van ami jobb oldalt található. Ezekben a boltokban tudjuk a hajókat fejleszteni vagy több hajót építeni, hogy minél több energiát tudjunk szerezni. Minden egyes hajónak megvannak saját jó és rossz tulajdonságai. A jó tulajdonságaikat lehet fejleszteni, ilyenek például: a sebessége, vagy a szállított energia mennyisége. A jelenlegi vétel és eladási árat akkor látjuk amikor rávisszük a kurzort, bal kattintással megvesszük, jobb kattintással pedig eladjuk 25%-os veszteséggel.

- Sebesség fejlesztés („Speed upgrade”): A sebesség növelésével csökkenthetjük azt a időt, ami arra szükséges, hogy a hajó a nap és az anyahajó közti útját megtegye.
- Energia tárolás fejlesztés („Cargo upgrade”): Ez növeli a hajó energia tároló kapacitását. Ezzel növelhetjük az a energia mennyiséget amit a hajó a naptól az anyahajóig elszállít.
- Fejlesztés hely vétele („Buy upgrade slots”): A fejlesztések száma limitált. Új fejlesztés hely vételével megnövelhetjük ezt a fejlesztésszámot.
- Hajó vásárlása („Buy ship”): Itt tudunk hajókat vásárolni. Minden hajónak saját maximális darabszáma, amit az anyahajó el tud tárolni. Ha elérjük ezt a maximális értéket, akkor nem tudunk több hajót venni. Ezek után ez a gomb zárolva lesz. A hajókat eladására nincs lehetőség, ha megvettünk egy hajót akkor az árat nem lehet visszaszerezni.



6. ábra Fejlesztési felület

Játékmenet

A Játék kezdésekor láthatunk egy narancssárga napot meg a Zero-1 anyahajót, ezen kívül van 1200 energiánk és 100 meteor porunk. Oldalt látható 3 gomb, ami megnyitja a különböző hajók fejlesztési ablakait, itt 4 gombot találunk.

Ezen a menün belül lehet venni új hajót és fejlesztési helyet emelet még lehet venni gyorsaság és energia kapacitás fejlesztést is. Mindegyik hajónak megvan a saját előnye és hátránya, például a gyors hajónak („fast ship”) a sebessége nagyobb százalékkal növekedik, mint az energia kapacitása míg, a szállító hajó („cargo ship”) fejlesztései során a energia kapacitást érdemesebb fejleszteni.

A szuper hajó („super ship”) rettentően drága de 50-nél több fejlesztés fér bele, mint a gyors és szállító hajóba, valamint a fejlesztések során több előnyt is nyer. Ha a játékos elér egy energia szintet akkor a Zero-1 anyahajó tud ugrani, de az ugrásnak van hátránya, hogy minden egyes épített hajókat hátra kell hagyni hisz a súly az túl sok lenne az ugrási hajtóműnek.

Ugrás közben az anyahajó össze tud gyűjteni egy különleges erőforrást, meteor port („Meteor dust”) ebből a pórból lehet pörgetni különleges hajó kinézeteket vagy energiát is lehet belőle kinyerni. Mikor az anyahajó befejezte az ugrást akkor eljut egy új fajta naphoz. Minden egyes ugrás során több energiát kell szerezni a következő ugráshoz.

Célja a játéknak, hogy minél több pontot szerezzen, meglátogassa az összes fajta napot és kigyűjtse az összes hajókinézetet.

Mentett adatok

A felhasználó által szerzett energiát, meteor port, hajók száma és fejlesztését, megszerzett kinézeteket egy online Amazon Web Services RDS adatbázisba van elmentve. Így a játékos által szerzett eredmények nem vesznek el ha törli a játékot.

Fejlesztői dokumentáció

Felhasznált eszközök

Godot játékmotor¹

Godot egy platformfüggetlen, ingyenes és nyílt forráskódú MIT licenc alatt kiadott játékmotor. Eredetileg csak néhány amerikai cég számára lett kifejlesztve nyilvános kiadása előtt. A fejlesztőikörnyezet macOS, Windows és akár Linuxon is fut. A Godotal mobil-, PC és webes játékokat is lehet fejleszteni.

A Godot célja, hogy egy teljes körű játékfejlesztő környezetet nyújtson a felhasználók számára. Lehetővé teszi a fejlesztőknek, hogy a semmiből, csupán tartalomkészítő eszközök (zenék, képek, textúrák stb.) segítségével létrehozzanak egy játékot. A játékok alkotóelemei a kódoktól a grafikus eszközökig a számítógép fájlrendszerébe vannak lementve (adatbázis helyett). Ennek a tárolási módnak a szándeka az, hogy a játékfejlesztő csapatoknak könnyebb legyen verziókövetés használatával együtt dolgozniuk a forráskódokon.

Több platformra is telepíthető, engedélyezi a textúratömörítést, valamint a felbontás beállítását külön mindegyik platformhoz. Jelenleg a támogatott platformok a Linux, macOS, Windows, FreeBSD, OpenBSD / DragonFly BSD, Android, iOS, BlackBerry 10 és a HTML5. Ezeken felül a Windows Runtime (WinRT) és az Universal Windows Platform (UWP) is támogatott.

A Godot elnevezés Samuel Beckett Godot-ra várva című művéből származik, kifejezve a készítők azon szándékát, miszerint az motor sosem készül el, mindig lesz olyan új funkció, amire érdemes várni.

Programozási nyelv

A C# az a programozási nyelv, ami a legközvetlenebb módon tükrözi az alatta működő, minden .NET programot futtató .NET keretrendszert, valamint erősen függ is attól: nincsen nem menedzsel, natív módban futó C# program. A primitív adattípusai objektumok, a NET típusok megfelelői.

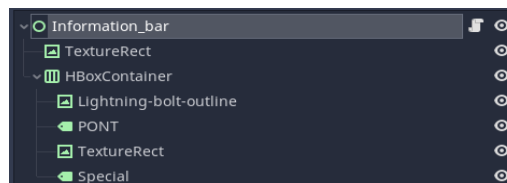
¹ [https://en.wikipedia.org/wiki/Godot_\(game_engine\)](https://en.wikipedia.org/wiki/Godot_(game_engine))

Godot definíciók

Minden játékmotor az általunk használt absztrakciók körül forog, hogy felépíthessük a videojátékunkat. Godotban, a játék egy fába rendezett node-okból halmazából áll, amiket egymásba csoportosítasz jelenetekké. Ezután ezeket a node-kat összeköthetted egymással signal-ok („jelek”) segítségével, tudnak kommunikálni.

Scenes (jelenetek)

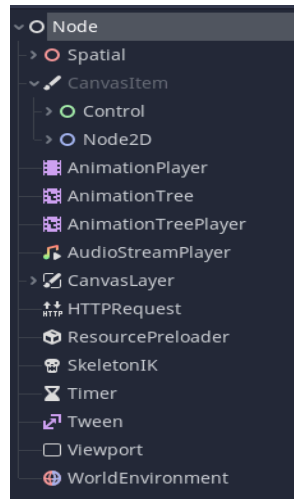
Godotban a játékot újra felhasználható jelenetekre tudjuk bontani. A jelenetek lehetnek hajók, ellenségek, a játékos, egy fegyver, a felhasználói felület, menü, egy teljes szint, minden, ami a játékban van az egy jelenet vagy, egy jelenet része. Jeleneteket be is lehet ágyazni egymásba.



7. ábra Információs sáv jelenet fája

Nodes

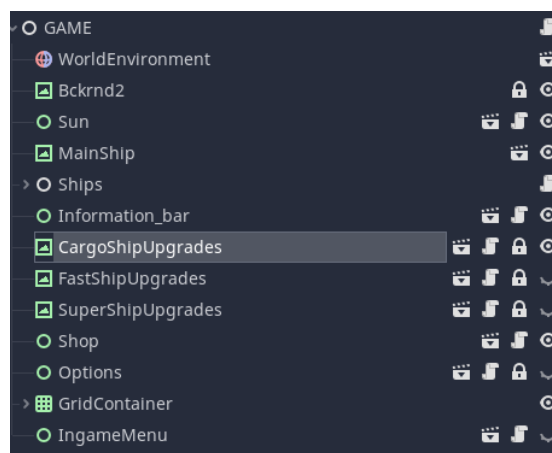
Egy jelenet az egy vagy több node-ból áll. Node-ok a játéknak a legkisebb építőelemei. Ha egy node-ot jelenetként ment el, az egyetlen node-ként jelenik meg, belső struktúrája az el van rejtve a szerkesztőben. A Godot az node-tipusok kiterjedt könyvtárát kínálja, amelyeket kombinálhat és bővíthet sokoldalúbb node-ok létrehozásához. 2D, 3D vagy felhasználói felület esetén a legtöbb esetben az effajta node-okkal dolgozunk.



8. ábra Node példa

Scene tree (jelenet fa)

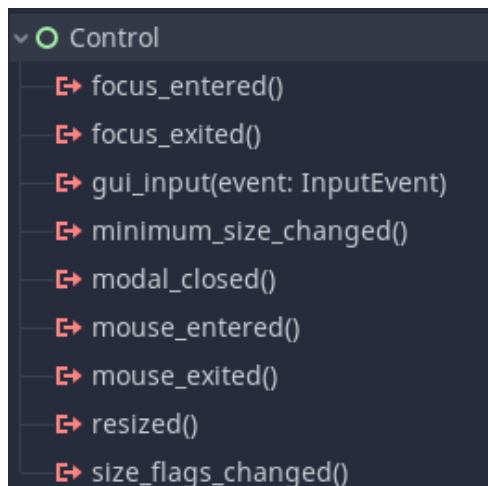
A játék összes jelenete összeáll a jelenetfában. Mivel a jelenetek a node-ok fái, a jelenetfa is a node-ok fája. De könnyebb a játékot jelenetek szerint képzelni, mivel azok karaktereket, fegyvereket, ajtókat vagy a felhasználói felületet ábrázolhatják.



9. ábra Egy általános jelenet fája

Signals (jelek)

A Node-ok jeleket bocsájtanak ki, amikor valamilyen esemény bekövetkezik. Ez a funkció lehetővé teszi a node-ok kommunikációját anélkül, hogy kódba kellene bekötni őket. Ez nagy rugalmasságot biztosít a jelenetek felépítésében. Saját jeleket is készíthetünk amit összeköthetünk funkciókkal.



10. ábra Általános jelek

Program architektúrája

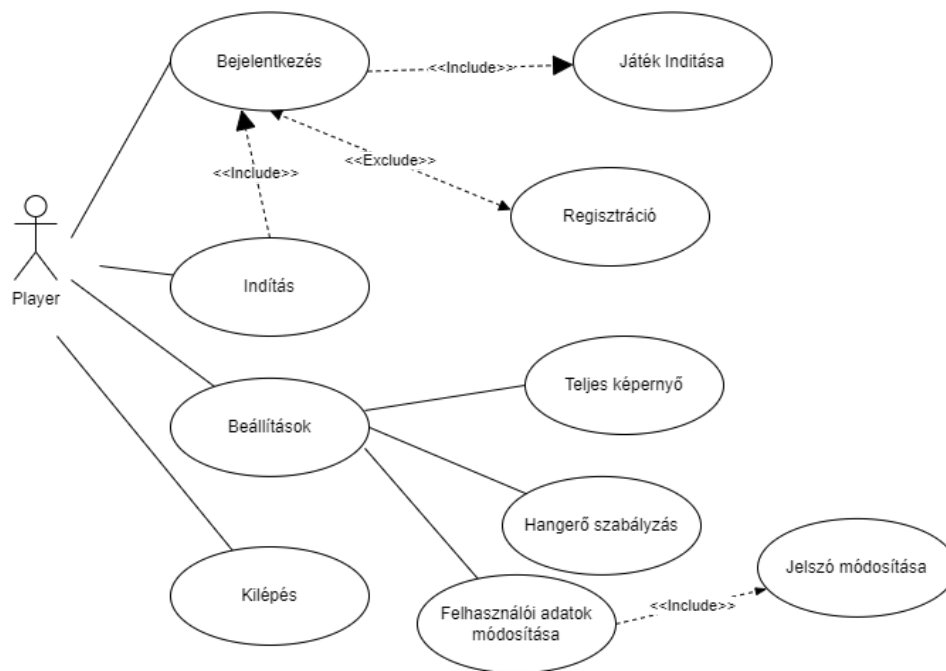
- Felhasználói felület (UI)
- Játék logika (GL)
- Adatbázis (DB)

A játékok felhasználói felületének (UI) célja, hogy lehetővé tegye a felhasználó számára, hogy a játékvilágon belül egy feladatot végrehajtson akár közvetlen bevitellel, akár egy „Heads Up Display” (HUD) művelettel. A felhasználói felület egy program része, amely lehetővé teszi az ember-számítógép interakcióját. A felhasználói felületeket az emberi pszichológia és fiziológia körütekintésével és megértésével kell megtervezni, mivel a program színe, formája és használati módja határozza meg, hogy milyen jól készül a felhasználói felület.

A játék logika (GL) magába foglalja a játék alapköveit a szkripteket, minden eleme a játéknak valami szkript alapján fut.

A adatbázis (DB) réteg az adatok tárolásával foglalkozik. A játék során elért eredmények egy külső Amazon Web Services RDS MySQL szerverre vannak elmentve.

Használati eset diagram



11. ábra Használati eset diagram

A használati eset diagram szemlélteti az alkalmazás funkcióit.

A játékos („Player”) a játék indítása után a menüben találja magát, ahol több lehetőségből választhat. Ezek a következők:

- A beállításokban („Settings”) választhat, hogy teljes képernyős módban szeretne játszani és kiválasztja a neki megfelelő hangerőt a háttérzenének meg a hangeffektusoknak. Emellett még megváltoztathatja a játékos a jelenlegi jelszavát, ha be van jelentkezve.
- Indítás („Play”) gombra kattintva megjelenik a bejelentkezés ablak, ha a felhasználó nincs bejelentkezve. Ellenkező esetben tovább lép a játékba.
- Bejelentkezés ikonra kattintva a játékos („Player”) előtt megjelenik a bejelentkezés ablak, ahol beírhatja a felhasználó nevét és jelszavát. Ha nincs felhasználói fiókja akkor átválthat regisztrációs módba, ahol be kell írnia az email címét, felhasználó nevét, jelszavát és meg kell erősíteni a jelszavát. Sikeres belépés után az indítás („Play”) gombra kattintva tovább lép a játékra. Ha jelenleg a játékos be van lépve egy felhasználói fiókba akkor a fiókkezelő ikonra rákattintva megint kiléphet a jelenleg használt fiókból.
- Kilépés („Quit”) gomb lenyomásával az alkalmazás bezáródik.

Jelenetek

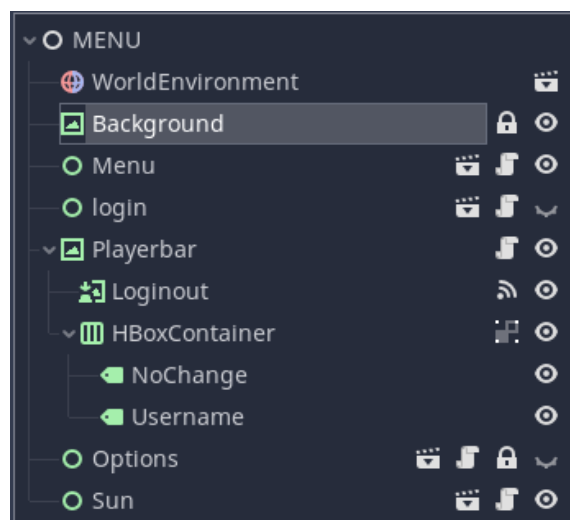
A Godotban minden egy jelenet, a jelenetek az egy node-okból álló fa.

Menü

Ez a játék fő jelenete ez az a jelenet, amit elsőnek meglát a játékos. Azt a funkciót szolgálja, hogy a felhasználó tudja kezelni a beállításokat és a fiókját. Innen tudja elindítani a játékos a játékot.

Komponensek:

- WorldEnvironment: Ez arra szolgál hogy a világító és fénylő effektust rakjon a napra és minden olyan más node-ra ahol az „self-modulate raw” érték nagyobb mint 1.
- TextureRect (Background): Ez a háttér a főmenünek, csak egy egyszerű képbeillesztés ami a képernyő nagyságára van illesztve.
- Control (Menu): Ez egy beágyazott jelenet ez tartalmazza a főmenü gombokat.
- Control (Login): Ez egy beágyazott jelenet ez tartalmazza a bejelentkezés és regisztrációs panelt.
- TextureRect (Playerbar): Egy egyszerű képbeillesztés arra szolgál, hogy a bejelentkezés gombnak legyen háttér színe, ami különbözik a fő háttérképtől.
- Control (Options): Egy beágyazott jelenet arra szolgál, hogy magában tartsa a beállításokat.
- Control (Sun): Egy beágyazott jelenet arra szolgál, hogy a napnak és annak animációi egy helyen legyenek



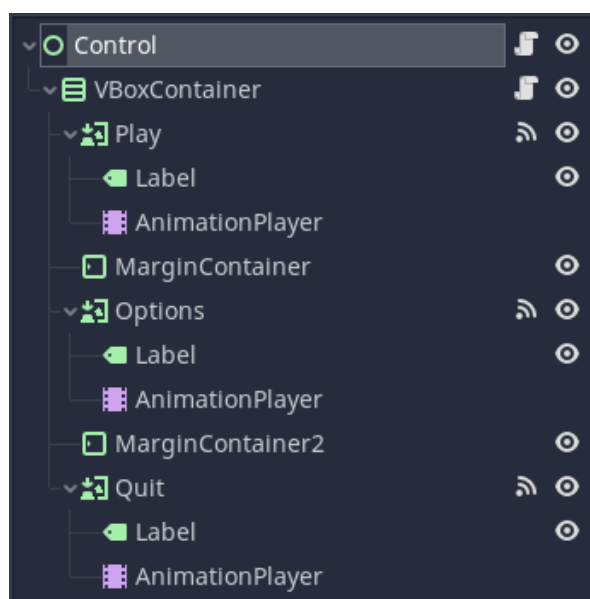
12. ábra Menü jelenet

Menü gombok

Egy saját jelenetben vannak a menügombok a könnyebb kezelés érdekében. Ez arra szolgál, hogy a menügombok saját tárolóban legyen a könnyebb elhelyezés és kód megírása érdekében.

Komponensek:

- Control: Ez a jelenetfának a gyökér node-ja, ez foglalja magába az összes gyermek node-ot. control node megmondja, hogy milyen nagyságba és hol legyen a helyzete a node-oknak amiket magába foglal.
- VBoxContainer: Ez egy vertikális táblázat, amivel ugyanolyan távolságban lesznek egymástól a gombok és így minden egyes változtatás kihat az összes gombra
- TextureButton („Play”): Saját kinézetet használ a gomb és arra szolgál, hogy elindítsa a játékot vagy megjelenítse a bejelentkező ablakot.
- TextureButton („Options”): Arra szolgál, hogy megjelenítse a beállítások ablakot.
- TextureButton („Quit”): Arra szolgál, hogy bezárja a játékot.
- Label: Egy szövegdoboz saját szöveg stílusal beállítva
- AnimationPlayer: Ebbe van beállítva, hogy ha ráviszi az egeret a felhasználó gombok animációja lejátsszódik. A AnimationPlayer key-eket használ a animáció készítéséhez. A animációk futtatásához deskriptíven keresztül küld egy jelet.
- MarginContainer: Ez arra szolgál, hogy margót állítsunk mind a négy irányba (fel, le, jobbra, balra) tetszés szerint.



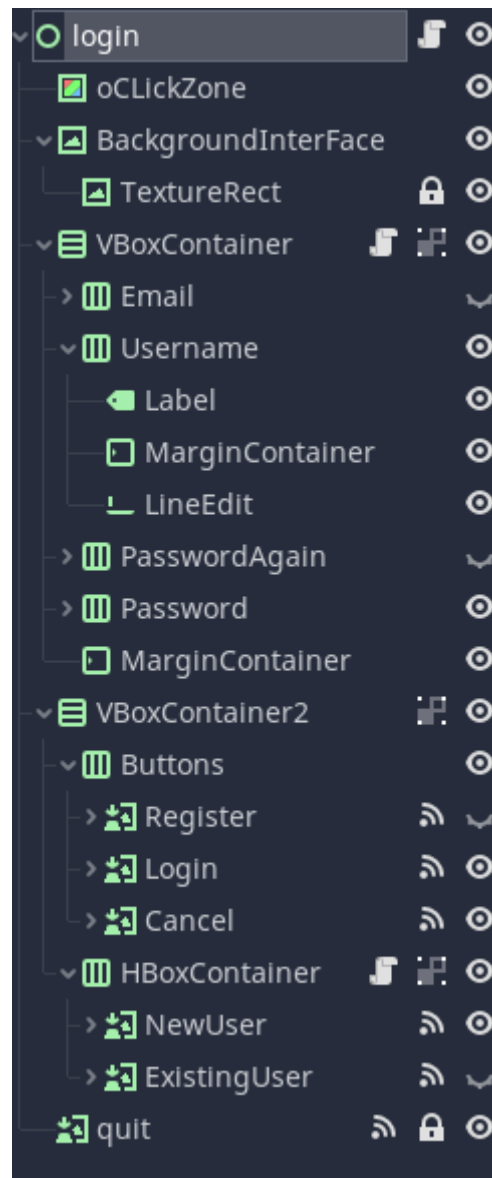
13. ábra Menü gombok jelenete

Fiókkezelő ablak

Ez a jelenet a bejelentkezés és a regisztrációs ablak szerepét egyaránt betölti. Ez a jelenet a [14] számú ábrán található.

Komponensek:

- ColorRect („oClickZone”): Ez azt a cél szolgálja, hogy a menü mögött lévő képernyő elsötétüljön.
- TextureRect („BackgroundInterFace”): Ez a menüpont háttere. Ennek gyermekeként alá van rendelve még egy „TextureRect” ami a menü hátterét sötétíti meg.
- HBoxContainer (Email): Ez két fő elemet tartalmaz: az email beírására szolgáló beviteli mezőt, és a beviteli mező címkéjét.
- HBoxContainer (Username): Ez két fő elemet tartalmaz: a név beírására szolgáló beviteli mezőt, és a beviteli mező címkéjét.
- HBoxContainer (Password): Ez két fő elemet tartalmaz: a jelszó beírására szolgáló beviteli mezőt, és a beviteli mező címkéjét.
- HBoxContainer (PasswordAgain): Ez két fő elemet tartalmaz: a megerősítő jelszó beírására szolgáló beviteli mezőt, és a beviteli mező címkéjét.
- TextureButton (Register): A gomb megnyomása után egy szkript ellenőrzi a megadott adatok formátumának helyességét és, hogy található-e ilyen nevű felhasználónév az adatbázisban. Ha megadott adatok helyesek akkor a regisztráció sikeres üzenet látható.
- TextureButton (Login): A gomb megnyomása után egy szkript ellenőrzi a megadott adatok formátumának helyességét, hogy található-e ilyen nevű felhasználónév az adatbázisban, hogy a megadott jelszó a megadott névhez tartozik. Sikeres bejelentkezés esetén megjelenik egy üzenet a sikeres bejelentkezésről, ellenkező esetben visszadob egy hibaüzenetet.
- TextureButton (Cancel): A gomb megnyomása után bezáródik a fiókkezelés ablak.
- TextureButton (NewUser): Ez a gomb csak bejelentkezés módban látszódik rákattintáskor a fiókkezelő ablak átvált regisztrációs módba, megjelennek az email és a jelszó megerősítő mezők.
- TextureButton (ExistingUser): Ez a gomb csak regisztrációs módban látszódik rákattintáskor a fiókkezelő ablak átvált bejelentkezés módba, csak a felhasználónév és a jelszó mező látható.
- TextureButton (Quit): Rákattintáskor bezárja az ablakot.



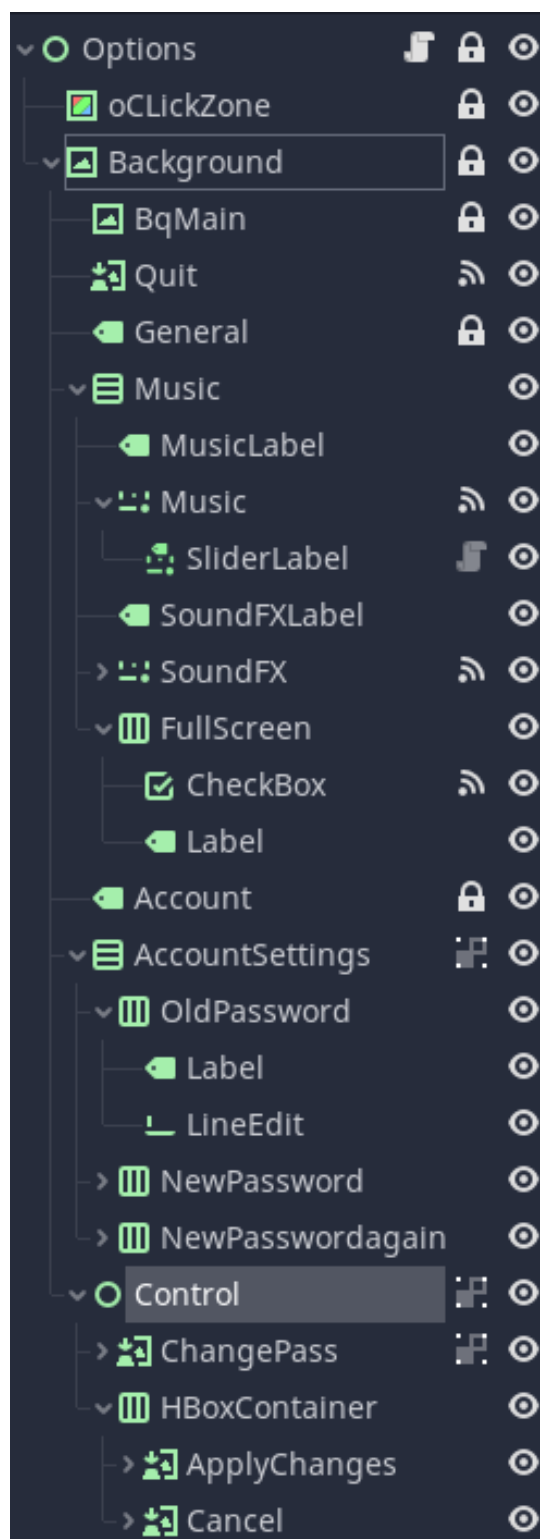
14. ábra Fiókkezelő ablak jelenet fája

Beállítások ablak

Ebben a ablakban a általános beállításokat és a fiókbeállításokat lehet beállítani amit a program lement a felhasználó számítógépére egy „config” fájlba. Ez a jelenet a [15] számú ábrán található

Komponensek:

- ColorRect (NoClickZone): Ez azt a cél szolgálja, hogy a beállítás ablak mögött lévő képernyő elsötétüljön.
- TextureRect („Background”): Ez a menüpont háttere. Ennek van alárendelve minden más node gyermekként.
- TextureButton („Quit”): Ez a gomb bezárja a beállítás ablakot.
- Slider („Music”): Ezzel a csúszkával lehet állítani a háterzene hangerejét. Ennek gyermekeként van egy saját node ami arra szolgál hogy a csúszka felé mikor rávisszük az egeret kiírja a jelenlegi értékét 0 és 100 között;
- Slider („SoundFX”): Ezzel a csúszkával lehet állítani a hang effektusok hangerejét. Ennek gyermekeként van egy saját node ami arra szolgál hogy a csúszka felé mikor rávisszük az egeret kiírja a jelenlegi értékét 0 és 100 között;
- CheckBox („Fullscreen”): Ez a jelölőnégyzet arra szolgál, hogy teljes képernyős mód és ablakos mód között lehessen váltani.
- VboxContainer („OldPassword”): Ebben a vertikális táblázat tárolóban található a beviteli mező, ahova a felhasználó jelenlegi jelszava kerül.
- VboxContainer („NewPassword”): Ebben a vertikális táblázat tárolóban található a beviteli mező, ahova a felhasználó új jelszava kerül.
- VboxContainer („NewPasswordagain”): Ebben a vertikális táblázat tárolóban található a beviteli mező, ahova a felhasználó új jelszava kerül megerősítés céljából.
- TextureButton („ChangePass”): Gomb rányomásakor ellenőrzi, hogy az összes beviteli mezőben megfelelő adatok vannak. Ha az összes adat megfelelő akkor megváltoztatja a felhasználó jelszavát.
- TextureButton („ApplyChanges”): Ezzel el tudjuk menteni a beállításokban elvégzett módosításokat, amit elment egy fájlba.
- TextureButton („Cancel”): Mentés nélkül kilép a beállításokból a fő menübe.



15. ábra Beállítás jelenet fa

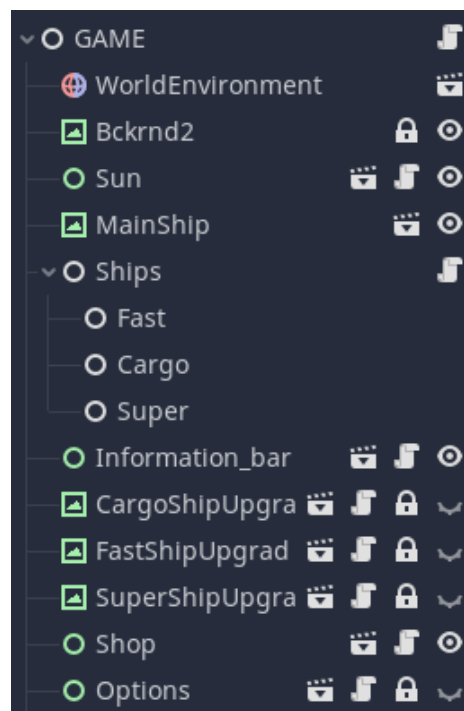
•

A játék jelenet

Ez a fő jelenet, ahol a videójátékkal játszunk. Itt találhatóak a játék fő elemei.

Komponensek:

- GameEnviromet: Arra szolgál, hogy amelyik node-nak a „self-modulate raw” értéke az nagyobb mint egy az világítson és egyéb effektusok készítésében is segít.
- Control („Sun”): Ez a nap, amihez mennek a hajók és viszik a energiát az anyahajónak.
- TextureRect („MainShip”): Ez a Zero-1 anyahajó innen indulnak el a hajók a naphoz és ide szállítják az energiát.
- Node („Ships”): egy gyűjtő node semmi funkciója nincs azon kívül hogy, az újonnan vett gyors, szállító és szuper hajók bekerüljenek a saját gyűjtő node-jukba.
- Control („Information_bar”): Ebben a control node-ban találhatóak a jelenlegi energia mennyiség és hogy mennyi meteor porja van a játékosnak.
- TextureRect („CargoShipUpgrades”): Tárolója a szállító hajó fejlesztési menünek.
- TextureRect („FastShipUpgrades”): Tárolója a gyors hajó fejlesztési menünek.
- TextureRect („SuperShipUpgrades”): Tárolója a szuper hajó fejlesztési menünek.
- Control („Options”): Ebbe található a beállítások menü.



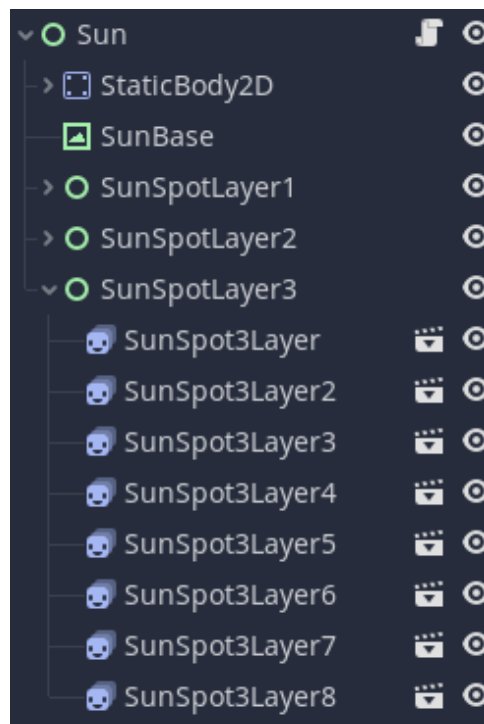
16. ábra A Játék jelenet fája

A nap:

A nap az egy összetett animált sprite, több animált sprite-ból lett összerakva, ami egy Control node-al van összekötve.

Komponensek:

- StaticBody2D: Ez adja meg statikus testét a napnak ami megmondja a hajóknak, hogy innen tovább ne menjenek és forduljanak meg.
- TextureRect („SunBase”): Ez az alap textúrája a napnak egy fehér kör, aminek a színét kódból módosul.
- Control („SunspotLayer1-3”): Ebben a control node-okban találhatóak az „AnimatedSprite” node-ok minden egyes node egymásnak a másolata és animáció egy „SpriteSheet” -ből van lejátszva. Összesen van 3 fajta animáció.



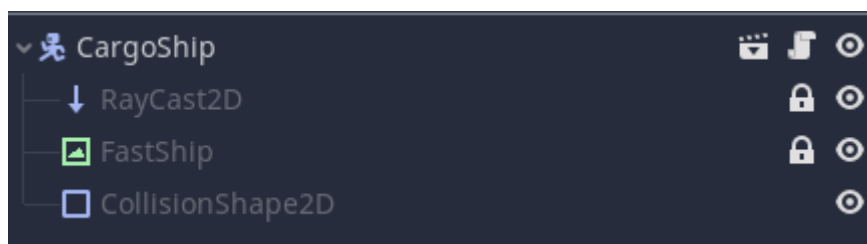
17. ábra Nap jelenet fa

A hajók

A gyors, szállító és a szuper hajók összetételében ugyanolyanok csak pár textúra van kicserélve, hogy könnyebben meg lehessen különböztetni őket.

Komponensek:

- KinematicBody2D („CargoShip”): A kinematikus testek speciális típusú testek, amelyeket a felhasználó vagy gép által vezérelt. Egyáltalán nem hat rájuk a fizika; más típusú testekhez, például karakterhez vagy merev testhez, ezek ugyanazok, mint a statikus testek. Ebben az esetben arra szolgálnak, hogy a hajók két test között mozogjanak
- RayCast2D: A RayCast egy vonalat reprezentál az eredetétől a célpozícióig. A 2D tér lekérdezésére szolgál, hogy megtalálja a legközelebbi objektumot a sugár útja mentén.
- TextureRect („ShipType”): Ez a hajónak a kinézete. Ez az egyedüli különbség a hajók jelenetei között.
- CollisionShape2D: Megadja, hogy a KinematicBody2D-nek milyen alakzata legyen lehet négyzet, kör, ellipszis vagy akármilyen 2D forma amit el lehet képzelni



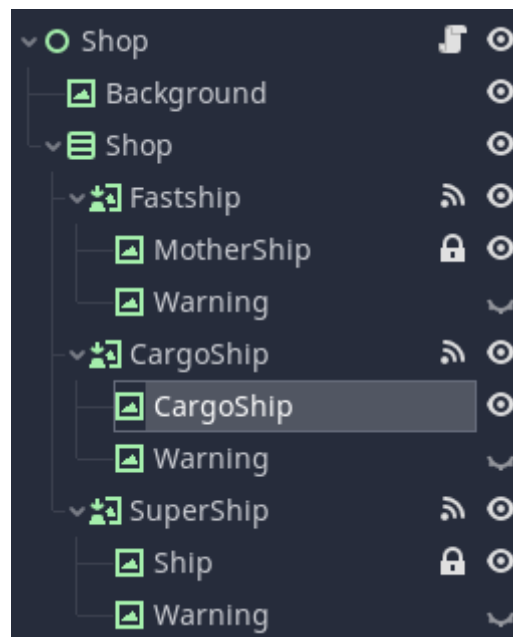
18. ábra Egy hajó jelenete

A bolt

Ebben az oldalmenüben találhatóak a gombok amik előhozzák a hajó menüket. A hajó menükben lehet vásárolni fejlesztéseket és hajót.

Komponensek:

- TextureRect („Background”): Ez a boltnak a háterre ebbe van belefoglalva az összes gomb.
- TextureButton („FastShip”): Ezzel a gombbal meg lehet nyitni a gyors („Fast”) hajónak a fejlesztési menüjét.
- TextureButton („CargoShip”): Ezzel a gombbal meg lehet nyitni a szállító („Cargo”) hajónak a fejlesztési menüjét.
- TextureButton („SuperShip”): Ezzel a gombbal meg lehet nyitni a szuper („Super”) hajónak a fejlesztési menüjét.
- TextureRect („Warning”): Ez arra szolgál, hogy jelezze a játékosnak ha, tud hajót vagy fejlesztést venni.



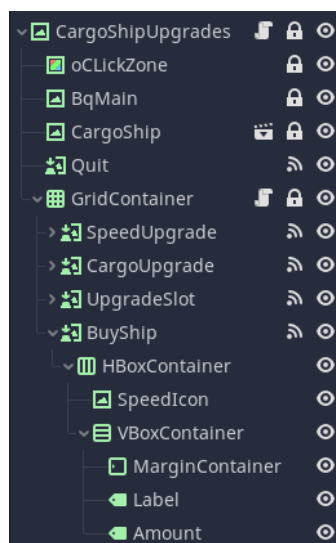
19. ábra A bolt jelenet fája

A hajók menüpont

A gyors, szállító, szuper hajó menüpont, ahol a fejlesztéseket és magát a hajót lehet megvenni az mind szerkezetileg ugyan úgy van felépítve, a szkripten kívül az egyetlen különbség a megjelenített ábra a hajóról.

Komponensek:

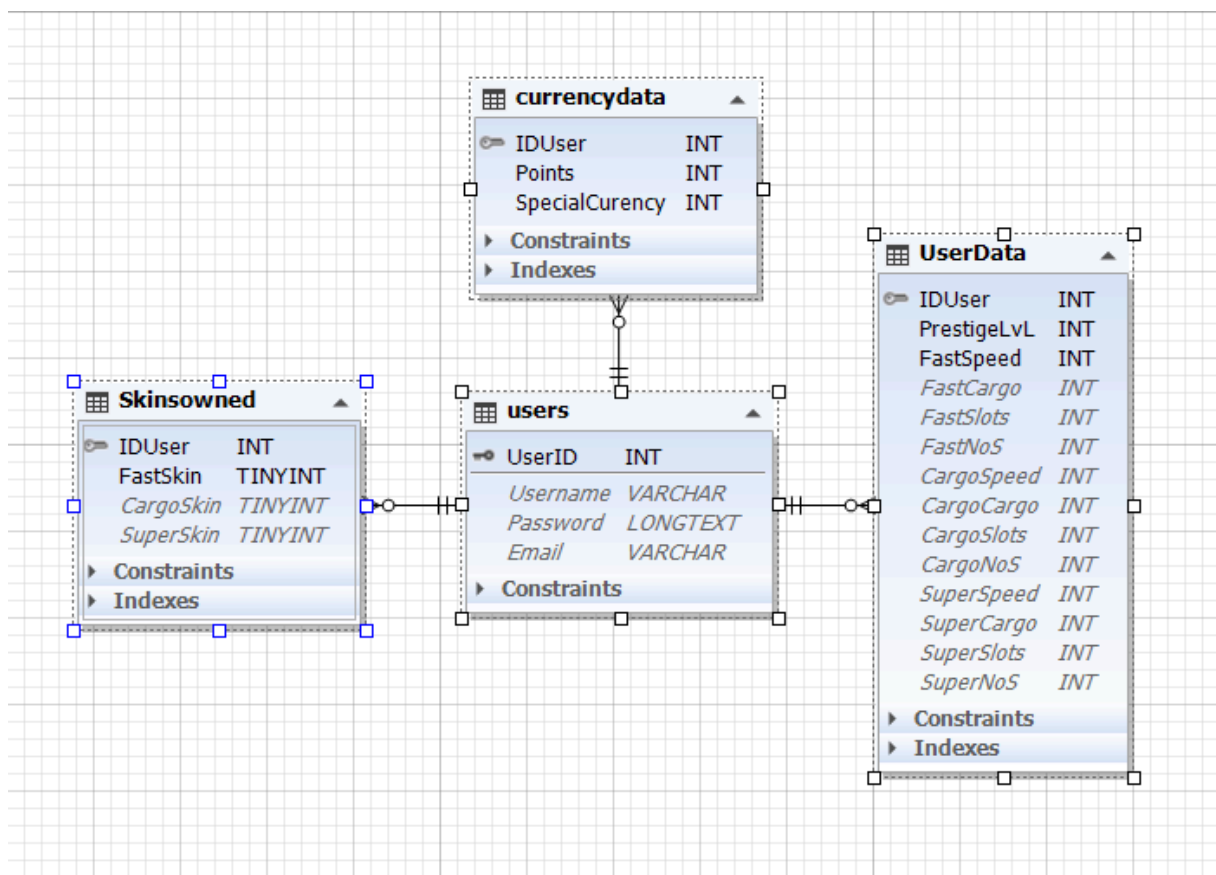
- TextureRect („ShipTypeUpgrade”): ez a fő node-ja a jelenetnek, ennek vannak alárendelve a node-ok.
- ColorRect („No_click_Area”): Egy elsötétülést készít az ablak mögött, ezen be van kapcsolva a „Show_behind_parent” emiatt jelenik meg a fő node mögött.
- TextureRect („ShipType”): Ez a ábrázolása, hogy jelenleg melyik hajó fejlesztésében vagyunk.
- TextureButton („Quit”): Erre a gombra kattintva kilépünk a bolt ablakból.
- TextureButton („SpeedUpgrade”): Ezzel a gombbal lehet hajó sebességnövelő fejlesztést venni bal kattintással, vagy eladni jobb kattintással.
- TextureButton („CargoUpgrade”): Ezzel a gombbal lehet hajó energia kapacitás növelő fejlesztést venni bal kattintással, vagy eladni jobb kattintással.
- TextureButton („UpgradeSlot”): Ezzel a gombbal lehet venni további fejlesztési helyet. Ennek van egy maximális értéke.
- TextureButton („BuyShip”): Ezzel a gombbal lehet venni a jelenleg kiválasztott hajót. Ez a gomb nem használható ha a játékos eléri a maximális hajó mennyiséget.



20. ábra Egy fejlesztési ablak jelenet fája

Adatok tárolása

A játék játszása során szerzett energiát és meteor port, valamint a megvásárolt fejlesztéseket scriptable object, azaz szkriptelhető objektumba van eltárolva futási időben, majd ezt tárolja egy internetes adatbázisba. A szkriptelhető objektumok Godotban olyan objektumok, amik adatok tárolására és saját funkciók készítésére használhatóak. Három szkriptelhető objektumot használok, az egyiket a MySQL parancsok lefuttatásához, a hajó adatok tárolására az utolsót pedig a globális paraméterek eltárolására. Az objektumokban több osztály is található. Minden adatot egy Amazon Web Services RDS MySQL szerveren van eltárolva egy külön fiókkal, aminek csak UPDATE, INSERT, SELECT jogosultsága van.



21. ábra Adatbázis diagram

Összegzés

A záródolgozat feladata az egy egyjátékos kétdimenziós videojáték, mely egy MySQL adatbázishoz csatlakozik. Az előre eltervezett és a témabejelentőben meghatározott funkciókat sikerült megvalósítani. Az technikumon szerzett tudásomat felhasználtam a fejlesztéshez, emelett sok tapasztalattal gazdagodtam játékfejlesztés területén, illetve Godot és C# ismereteimet is bővítettem. Szívesen foglalkoznék ezután is játékfejlesztéssel, hiszen egy játék készítése izgalmas és kreatív munka, de egyben kihívás is, ezért sok tapasztalat szerezhető vele. Egy játékot mindig lehet tovább fejleszteni, bővíteni, finomítani. Sok erre irányuló ötlet merült fel bennem a fejlesztés során, azonban ezekre idő hiányában nekem kevés lehetőségem volt, de szívesen megvalósítanék még sok funkciót a jövőben, vagy szívesen látnám, ha valaki kedvet kapna hozzá és továbbfejlesztené ezt a játékot. A játék lokális mentésre átírva elérhető GitHub és Itch.io platformokon MIT nyílt forráskódú licenccel így akárki hozzá írhat és fejlesztheti.

Továbbfejlesztési lehetőségek

- Többfajta hajó hozzáadása másfajta fejlesztésekkel
- További kinézetek hozzáadása
- Többfajta nap hozzáadása akár képzeletbeli, nem létező napok hozzáadása különleges attribútumokkal
- UI továbbfejlesztése

Ábrák Jegyzéke

1. ábra Játékmenet	5
2. ábra Főmenü	6
3. ábra Beállítások	7
4. ábra Bejelentkezési felület	8
5. ábra Regisztrációs felület	8
6. ábra Fejlesztési felület	9
7. ábra Információs sáv jelenet fája	12
8. ábra Node példa	12
9. ábra Egy általános jelenet fája	13
10. ábra Általános jelek	13
11. ábra Használati eset diagram	15
12. ábra Menü jelenet	16
13. ábra Menü gombok jelenete	17
14. ábra Fiókkezelő ablak jelenet fája	19
15. ábra Beállítás jelenet fa	21
16. ábra A Játék jelenet fája	22
17. ábra Nap jelenet fa	23
18. ábra Egy hajó jelenete	24
19. ábra A bolt jelenet fája	25
20. ábra Egy fejlesztési ablak jelenet fája	26
21. ábra Adatbázis diagram	27