

Galaktikus Labirintus

Kiss Roland, Vámosi Dávid, Pálfi Dorián, Pungor Sebestyén, Scheuer Patrik

A Modern Team csapata

Modern szoftverfejlesztési eszközök

GKNB_INTM006

2024. október



**SZÉCHENYI
EGYETEM**
UNIVERSITY OF GYŐR

Tartalomjegyzék:

Leírás	3
Játékmenet	3
Játék mechanika	4
Pontgyűjtés és Jutalmak	4
Végső összecsapás	4
Történet	4
Követelmények	6
Használati esetek	6
Táblázatos felsorolás	7
Használati eset diagram	8
A program főbb strukturális elemeinek vázlatos azonosítása	9
Program viselkedés analízálása	9
Labirintus tárolási szerkezete	10
Labirintus előállításának módja	10
Adatfolyam diagram készítése a fájlműveletek jelölésével	11
Egyéb elemek és részletek	11

Leírás

A **Galaktikus Labirintus** egy hullám-alapú túlélő játék, ahol a játékos egy űrhajót irányít, és bolygóról bolygóra haladva különböző kihívásokkal néz szembe. Minden bolygón egy manuálisan generált labirintus várja a játékos, ahol az ellenséges hullámok egyre erősebbek és egyre többen lesznek, ahogy a küldetés előrehalad. A cél, hogy túlélj minden hullámot, megoldj minden labirintust, és felszabadítsd a galaxis minden bolygóját az ellenségek alól.

Játékmenet:

- **Űrhajó irányítása és bolygók közti utazás:**

A játékos egy űrhajót irányít, amely képes bolygóról bolygóra utazni. Minden bolygón egyedi labirintus és ellenséges hullámok várják. A játékos a bolygó felszínére érkezve belép a labirintusba, ahol navigálnia kell az akadályok és ellenségek között.

- **Bolygók és labirintusok:**

Minden bolygón egyedi, véletlenszerűen generált labirintus található. A labirintusok különböző környezeti elemekkel és nehézségi fokozatokkal rendelkeznek, így minden bolygón új kihívásokat tapasztal a játékos. A labirintusok egyedi kialakításai változatos játékmenetet biztosítanak, legyen szó sivatagos, dzsungeles vagy jéggel borított bolygókról.

- **Ellenséges hullámok (waves):**

Minden bolygón több ellenséges hullám érkezik, amelyek egyre erősebbek és nagyobb számban támadnak. Az ellenségek különböző képességekkel rendelkeznek: néhány gyors és kicsi, mások lassúak, de erősebbek. Minden hullám végén van egy erősebb "boss" ellenség, amelynek legyőzése szükséges a bolygó megtisztításához.

- **Akadályok:**

A labirintus tele van akadályokkal, amelyeket a játékosnak ki kell kerülnie vagy meg kell semmisítenie, hogy elérje a célját. Ezek lehetnek például energiamezők, zárt ajtók, amelyek kulcsokat igényelnek, vagy mozgó csapdák.

- **Bolygók közti utazás:**

Miután a játékos megtisztított egy bolygót az ellenségektől, visszatér az űrhajójába, és tovább utazhat a következő bolygóra, ahol egy újabb labirintus és ellenséges hullámok várják.

- **Fejlesztések és erősítők:**

Minden bolygón lehetőség van különféle fejlesztésekre, mint például új fegyverek, pajzsok vagy sebességnövelők megszerzésére. Az űrhajó fegyverzete és felszerelése folyamatosan javítható, így felkészültebb lesz az egyre nehezebb bolygók kihívásaira.

- **Végső cél:**

A végső cél, hogy minden bolygót megtisztíts a galaxisban, legyőzve az összes ellenséges hullámot és a bolygók uralkodó „boss” ellenségeit. Minden bolygó után a játékos jutalmat kap, amellyel fejlesztheti űrhajóját, hogy felkészüljön a következő kihívásra.

Játék Mechanika:

- **Navigáció:**
A játékos a nyílbillentyűkkel vagy WASD gombokkal irányítja az űrhajót és mozog a labirintusokban.
- **Harci rendszer:**
Az űrhajó különböző típusú fegyvereket használhat, mint például lézerágyú vagy rakéták. A fegyverek fejleszthetők a bolygón található erőforrások gyűjtésével.
- **Ellenséges mesterséges intelligencia:**
Az ellenségek reagálnak a játékos cselekedeteire, üldözik, támadják, és egyre intelligensebben lépnek fel, ahogy haladsz előre a játékban.
- **Hullámok nehézsége:**
Az ellenségek hullámai fokozatosan nehezednek, mind az ellenségek számában, mind a stratégiákban, amelyeket alkalmaznak.

Pontgyűjtés és Jutalmak:

- **Pontgyűjtés:**
Minden legyőzött ellenség és megoldott labirintus után a játékos pontokat szerez, amelyek egy globális high score táblázaton jelennek meg.
- **Jutalmak:**
A játékos különböző erőforrásokat gyűjthet, amelyekkel fejlesztheti az űrhajót: erősebb fegyverek, gyorsabb mozgás, nagyobb páncél.

Végső összecsapás:

Miután a játékos minden bolygót megtisztított, egy végső bolygóra érkezik, ahol a legnagyobb ellenséges erők és a végső boss vár rá, amelynek legyőzése felszabadítja az egész galaxist.

Történet

A hatalmas csillaghajó csendben lebegett az űr sötét végtelenségében, amikor hirtelen vészjelzés érkezett. Aszteroida vihar közeledett, és a hajó minden irányból fenyegető kőóriásokkal nézett szembe. A kapitány, egy veterán pilóta, gyors manőverezéssel próbálta kikerülni a pusztító meteorzáport, de tudta, hogy egy elkerülhetetlen találat biztosan be fog következni. Ahogy az első aszteroida eltalálta a hajót, vészesen zuhanni kezdtek egy bolygó felé, amit korábban észleltek a radaron.

A hajó nagy robajjal csapódott a felszínbe, egy idegen, barátságtalan világon. Még ki sem szálltak a roncsból, amikor megjelent az első hullám – furcsa, szörnyűséges idegen lények tűntek fel, akik egyenesen a hajó felé tartottak. A legénység fegyverei készen álltak, de a harc nem volt könnyű: a hullámok egyre sűrűbben érkeztek, mind erősebb ellenségekkel. Minden újabb támadással a legénység egy ládához futhatott, amelyből erősebb fegyvereket és életmentő felszerelést választhattak, de a döntés gyors és végleges volt.

Tudták, hogy nem lehet megállás – a hullámok jöttek, és csak idő kérdése volt, hogy megérkezzen a végső ellenség: az idegenek brutális fővezére, egy szörnyű boss. Ahogy a föld megremegett a hatalmas lény lépteitől, a csapat felkészült a végső összecsapásra. Egyetlen céljuk volt: megölni a szörnyet, mielőtt ők maguk esnek el.

Az ütközet véres és könyörtelen volt, de végül, amikor az idegen fővezér a földre rogyott, tudták, hogy sikerült. A harcnak vége volt, de az ára iszonyatosan magas. A túlélők lassan visszatértek a hajó roncsaihoz, felkészülve arra, hogy hazatérjenek – ha egyáltalán hazatérhetnek.

Követelmények

(Minimális oszlopok: egyedi azonosító (ID), megnevezés, leírás, prioritás)

Legalább 5, de legfeljebb 10 követelményt kell táblázatban összegyűjteni, ahol minden követelményhez tartozik egy egyedi azonosító (ID), megnevezés, leírás, valamint egy prioritási szint.)

ID	Név	Prioritás	Leírás
K01	Aszteroida mező	Magas	A játék első felében, amely a világűrben játszódik, lehessen aszteroidákat kerülgetni
K02	Bolygó felszín	Magas	A bolygón egy nyílt terepen lehessen harcolni az ellenséggel
K03	Mozgás	Magas	Az űrhajót, valamint később a karaktert lehessen mozgatni hogy ki tudja kerülni az akadályokat vagy az ellenségeket
K04	Interakció a felvehető tárgyakkal	Közepes	A játékos tudjon interaktálni a pályán megjelenő power-up-okkal, erősítőkkal, amivel könnyebb lesz a továbbjutás
K05	Környezeti interakció	Magas	Például az aszteroidákkal való ütközés során az űrhajó zuhanjon le
K06	Történet vonalának követése	Magas	A játékos a történet haladásával tudjon a játékban haladni, az aszteroidáktól kezdve a végső bossig bezárólag

Használati esetek

(A használati eset diagram: UML használati eset diagram, annak szabályainak megfelelően)

A használati esetek felsorolása táblázatban történik, és emellett UML használati eset (Use Case) diagramot is készíteni kell. Ez bemutatja, hogy a rendszer hogyan viselkedik különböző felhasználási esetekben.)

A vizsgált rendszer: a Játék maga

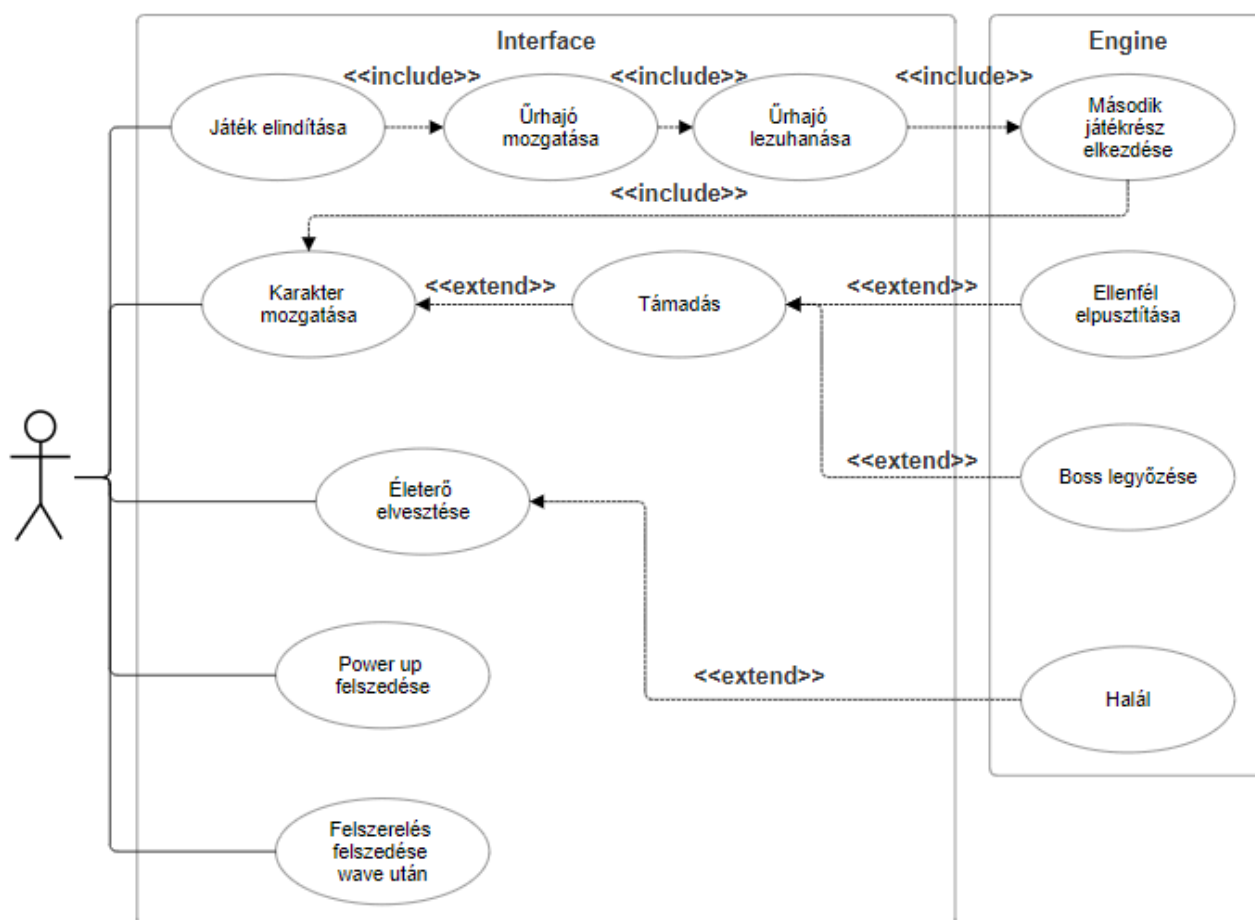
#	Use case neve	Prioritás	Leírás
	Játék elindítása	Magas	-
	Űrhajó mozgatása	Magas	Az űrben, amikor az akadály kikerülős rész van, az űrhajó irányítása
	Űrhajó lezuhanása	Közepes	Az űrhajót mindenképpen eltalálja egy aszteroida, zuhanjon le és vegye kezdetét a játék második fele ahol a földi ellenségekkel kell wave-eken keresztül harcolnia
	Második játékrész kezdése	Magas	A bolygó felszínén játszódó játékrész elindulása
	Karakter mozgatása	Magas	A zuhanás után a bolygón a főhőst mozgatjuk
	Támadás	Magas	A karakter valamilyen módon képes legyen megtámadni az ellenséget
	Ellenfél elpusztítása	Magas	A karakter képes legyen elpusztítani az ellenségeket
	Életerő elvesztése	Magas	Sérülés esetén a karakter veszítsen életerőt
	Halál	Magas	Az összes élet elvesztése után a karakter meghal és az adott szint vagy wave újraindul, a korábbi állapotok szerint (fejlesztések, életek száma stb.)
	Power up felszedése wave közben	Alacsony	A játék nem áll meg, csak erősebb lesz a karakter
	Power up vagy felszerelés kiválasztása minden wave után	Alacsony	Egy ládából választhatunk felszerelést, illetve power up-ot, amely segítségünkre lesz a következő összecsapások során
	Boss, főellenfél legyőzése	Magas	A kisebb ellenségekhez hasonlóan a boss-t is lehessen legyőzni
	Játék vége	Magas	A boss után a játék záródjon

			le
	Játék újraindítása	Alacsony	A felhasználó újraindítja a játékot a menüből

A rendszer aktorai: felhasználó

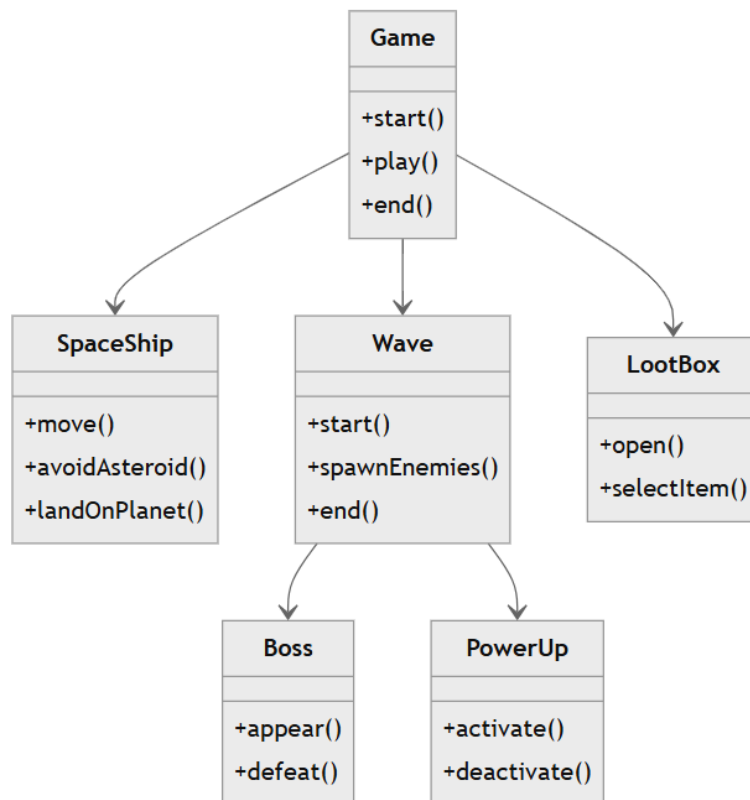
Az aktor a felhasználó, tehát azok az entitások, akik vagy amik kölcsönhatásba lépnek a rendszerrel.

A használati eset diagram a felhasználó szemszögéből azonosítja a játékkal kapcsolatos interakciós lehetőségeket



Megjegyzés: A második játékrész kétféleképpen indulhat el, vagy a játékos elrontja a szintet azzal hogy túl sok sérülést szenved el és amiatt zuhan le az űrhajó, vagy pedig teljesíti a szintet, de a végén a story szerint mindenképpen eltalálja egy aszteroida és emiatt lezuhan. Emiatt nincs kettéválasztva a két esemény, mert mindenképpen lezuhan az űrhajó és a játék ugyanúgy halad tovább nem számít hogy mi okozta a balesetet.

A program főbb strukturális elemeinek vázlatos azonosítása (célszerűen UML class diagramon)



Classok

class Game: Az egész projektet összefogó class, amely azért felel, hogy a játékot egyáltalán elinduljon, játszható legyen, illetve hogy befejeződjön.

class SpaceShip: A játék főszereplőjét szimbolizáló osztály. Átala van lehetőség a játék végigjátszásához.

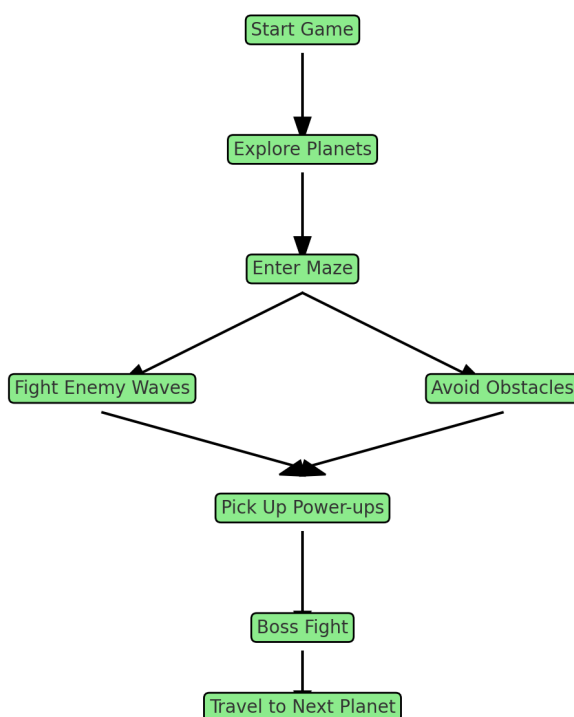
class Wave: Az adott szinten lévő ellenséghullámot kifejező osztály. Amikor a játékos belép egy szobába (szintre), ami nem a bossroom (ahol a boss spawnol), akkor minden hullámban x mennyiségű úrszörny éled, amiket a felhasználónak meg kell ölni.

class LootBox: Az az osztály, amely az adott pályán fellelhető ládákért és azok tartalmáért felel.

class Boss: Az adott szint legnagyobb szörnyét reprezentáló osztály. Őt kell legyőzni, hogy a játék tovább játszódhasson.

class PowerUp: Azok a tárgyakat leíró osztály, amiket a játékos szobákban tud találni, megfigyelni és felvenni azokat, ha szükség van rá.

Program viselkedés analízálása, folyamatábra vagy állapotgép formájában



Aktivitás diagram leírása

A játék elején karakterünk egy űrhajóban repülget az űrben, ahol is egy aszteroidamezőbe keveredik, amit nem tud elkerülni és lezuhan a legközelebbi bolygón. Ekkor belecsöppenünk a második szintre, ahol a felhasználó kiválaszthatja, hogy felkutatja a labirintust értékes lootokért, vagy megküzd az ellenséges "űrszörnyekkel". Főhősünk ezután egy szobába mehet, ahol harcolhat az adott szint főnökével. Ha sikeresen legyőzi a bosst, akkor lehetősége van átkerülni a következő bolygóra hasonló kalandokért.

[állapotdiagram helye]

Állapotdiagram leírása

[állapotdiagram után kerül leírásra]

Játék menete

Játékunk a lehetőségeket loop jellegűen ajánlja fel a felhasználó számára, aminek folyamatos teljesítése lehetővé teszi a pályák és a játék teljesítését. Ilyen lehetőségek például a következő szobába való átjutás egy sikeres bossfight után, az adott szint átkutatása, és a power-upok felvétele. Ha a user úgy dönt, hogy a következő szobába megy át, akkor a játék újra fel fogja sorolni az adott szint lehetőségeit.

Játék tárolási szerkezetének fájlformátuma (séma)

(Fel kell sorolni a labirintus tárolására használt adatstruktúrákat, például térképsémákat vagy hasonlókat.)

Tilemap rendszer

- **Séma:** Unity beépített **Tilemap** rendszerét fogjuk használni a labirintus vizuális és adatbázis kezelésére. A cellák itt a Tile-ek lesznek, és azok tulajdonságait külön lehet tárolni (pl. fal vagy út).
- **Unity implementáció:** Unity Tilemap komponens használata a 2D Grid rendszerben, amely automatikusan kezeli a labirintust.

Labirintus előállításának módja

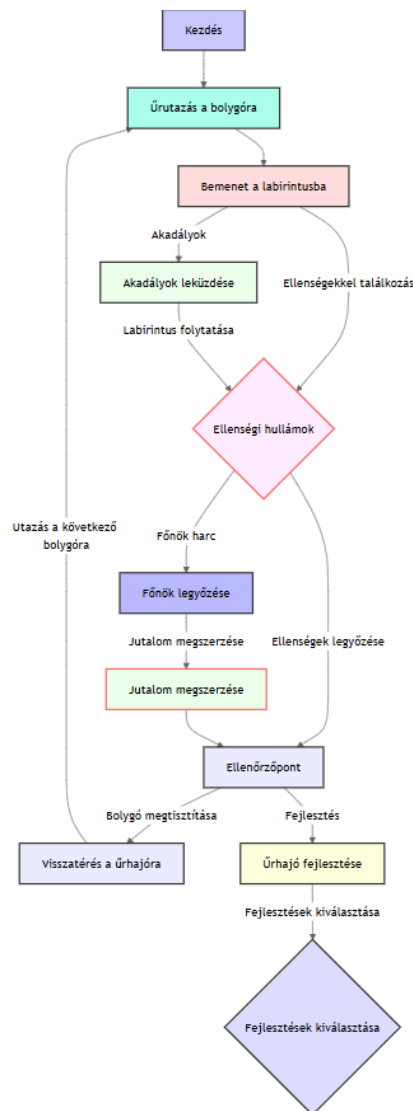
(Meg kell határozni, hogy a labirintus létrehozása véletlenszerűen generált vagy manuálisan történik. Ez a követelmény csak akkor vonatkozik rád, ha Unity alapú programot használasz.)

Mivel a játékunk egy történetközpontú úrhajós kaland, ezért a labirintusokat **manuálisan fogjuk megtervezni**. Ez lehetővé teszi, hogy a pályák illeszkedjenek a történethez, és egyedi, izgalmas helyszíneket hozhassunk létre. A manuális tervezéssel jobb lehetőségünk van narratív elemeket, rejtett történetszálakat vagy fontos eseményeket beépíteni a labirintusokba.

A történet és a pályadesign összhangja erősítheti a játék atmoszféráját, például ha egy labirintuson belül van egy rejtett szoba, ami egy kulcsfontosságú eseményt indít el. Ez segíthet mélyebb élményt nyújtani a játékosoknak.

Adatfolyam diagram készítése a fájlműveletek jelölésével ajánlott

(Adatfolyam-diagram készítése is javasolt, hogy bemutasd, hogyan áramlik az adat a rendszerben, különösen fájlműveletek során.)



Bármilyen egyéb elem vagy részlet jelentése, amely szükséges a specifikáció elkészítéséhez

(Bármilyen további részlet, amely szerinted fontos lehet a specifikációban, fel kell tüntetni.)

Elképzelés a játékról:

1. Szállunk az űrben
2. Aszteroidákat kikerüljük, de egy mindenképpen eltalál majd
3. Lezuhanunk egy bolygóra ahol jönnek a wavek
4. Wave közben instant nem kiválasztható power up-ok vagy hp
5. Wave után láda, amiből választunk valami cuccot ami segít
6. Néhány wave után jön egy boss
7. Azt megöljük és vége a játéknak

Grafikai elképzelés: