**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Informatikatudományi Intézet**

**Programozáselmélet és Szoftvertechnológia Tanszék**

Szakdolgozat címe

Szerző: Témavezető:

Márton Gergő Dr. Cserép Máté

Programtervező informatikus BSc. egyetemi adjunktus

**Budapest, 2025**

Ide kerül a hivatalos témabejelentő lap.

Tartalomjegyzék

[1. Bevezetés 1](#_Toc186028460)

[2. Felhasználói dokumentáció 2](#_Toc186028461)

[3. Fejlesztői dokumentáció 3](#_Toc186028462)

[3.1. Elemzés 3](#_Toc186028463)

[3.2. Felhasználói esetdiagramok 4](#_Toc186028464)

[3.3. Programszerkezet / Architektúra 4](#_Toc186028465)

[3.4. Architektúra rétegei 6](#_Toc186028466)

[3.4.1. View 6](#_Toc186028467)

[3.4.2. Model 6](#_Toc186028468)

[3.4.3. Controller 6](#_Toc186028469)

[3.4.4. Service 6](#_Toc186028470)

[4. Összefoglalás és további fejlesztési lehetőségek 7](#_Toc186028471)

[5. Irodalomjegyzék 8](#_Toc186028472)

[6. Melléklet 9](#_Toc186028473)

# Bevezetés

A lap mérete és a margók ebben a sablonban megfelelően vannak beállítva. A betűtípus adott (Calibri) és a 12-es betűméret is. Oldalszámozás a tartalomjegyzék után kezdődik és a dokumentum végéig folytatódik. A dolgozat fő fejezetcímei adottak és mindig új oldalon kezdődnek.



1. ábra: Galaxis útikalauz stopposoknak könyv

Használjátok az automatikus számozást az ábrákhoz. Túl sok ábra esetén érdemes ábrajegyzéket beszúrni a dokumentumban az Irodalomjegyzék után. A folyó szövegben hivatkozzatok minden egyes ábrára és táblázatra is kereszthivatkozással (például így: 1. ábra, vagy táblázat esetén így: 1. táblázat).

1. táblázat: Példatáblázat

|  |  |  |
| --- | --- | --- |
| asdf |  |  |
| 1234 |  |  |

# Felhasználói dokumentáció

# Fejlesztői dokumentáció

## Elemzés

* A játéknak csak és kizárólag többjátékos módja lesz, éppen ezért szükséges egy (fő)menü, ahol a játékos kiválaszthatja, hogy szeretne-e egy saját játékot **létrehozni**, vagy **csatlakozni** szeretne valakihez.
* Mivel ezek a partik – a társasjátékok világához képest – viszonylag *rövid ideig* tartanak, ezért mentés és betöltés funkciót nem fog az alkalmazás támogatni.
* Egy parti menetével kapcsolatban az elvárások: egy parti körökből, ezek a körök pedig az alábbi játékfázisokból állnak: **Parancs** (Command Phase), **Mozgás** (Movement Phase), **Harc** (Fighting Phase), illetve, amennyiben lehetséges **Közelharc** (Melee Phase). Ezeket a fázisokat a játékosok egymás után – nem szimultán módon – fogják végrehajtani.
  + **Parancs fázis**: A játékosok kapnak 1-1 „*Command*” Pontot, ebből tudnak plusz egységeket lehelyezni.
  + **Mozgás fázis**: Az egyes egységek mozgatása. A távolság a különböző egységekbe lesz beépítve.
  + **(Közel)Harci fázis**: A játékos eldöntheti, hogy melyik egységével melyik ellenfelet/ellenfeleket szeretné megtámadni. Figyelembe kell venni a támadások kódba égetett ***hatótávját***, majd *kockadobással* eldönti a játék, hogy a támadás sikeres-e/mennyi sebzést fog bevinni.
* A játékosok segítése érdekében amikor kiválasztunk egy egységet, megjelenítjük annak az *információs kártyáját*, mely leírja az egység különböző statisztikáit.
* Egy partinak akkor lesz **vége**, ha az egyik játékos egységei *elfogynak*, és nem is tud lehelyezni újakat, vagy ha egy játékos elég ideig *birtokolta* a pályán elhelyezett pontok többségét / közül mindent.

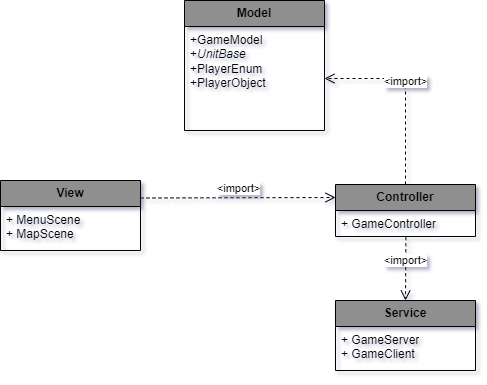
Megjegyzés: egyes funkciók elgondolása / megvalósítása még a fejlesztés során változhat, ha a tesztek végeredménye nem kielégítő játékélmény szempontjából. (Ilyenek például az 1-1 Command Pont a körök elején)

## Felhasználói esetdiagramok

2. Diagram: felhasználói esetdiagram egy játék menetéhez

1. Diagram: felhasználói esetdiagram új játék készítéséhez

## Programszerkezet / Architektúra

Az alkalmazást a(z) **Unity játékmotor** segítségével fogom megvalósítani. Architekturális felépítését tekintve a program az **MVC** (Model-View-Controller) architektúrában lesz implementálva. A **Nézet** rétegbe fognak tartozni a Jelenetek (*Scene*), és ez fogja fogadni az inputokat. A **Vezérlő** réteg fog felelni a játék vezérléséért. A Nézet továbbítja neki a kapott inputokat, és ezek alapján *vezérli a Model-t*, esetleg néhány speciális esetben direktfogja *frissíteni a Nézetet*. A **Model** fogja tartalmazni a játék fő logikáját.

3. Diagram: Csomagdiagram az architektúra szemléltetéséhez

## Architektúra rétegei

### View

### Model

### Controller

# Összefoglalás és további fejlesztési lehetőségek

# Irodalomjegyzék

# Melléklet