**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Informatikatudományi Intézet**

**Programozáselmélet és Szoftvertechnológia Tanszék**

Szakdolgozat címe

Szerző: Témavezető:

Név Név

Programtervező informatikus BSc. beosztás, titulus

**Budapest / Szombathely, 2023**

Ide kerül a hivatalos témabejelentő lap.

Tartalomjegyzék

[1. Bevezetés 1](#_Toc87521353)

[2. Felhasználói dokumentáció 2](#_Toc87521354)

[3. Fejlesztői dokumentáció 3](#_Toc87521355)

[4. Összefoglalás és további fejlesztési lehetőségek 4](#_Toc87521356)

[5. Irodalomjegyzék 5](#_Toc87521357)

[6. Melléklet 6](#_Toc87521358)

# 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 |  |  |

Architektúrális döntések

Videójátékfejlesztés esetén két ismert engine a Unity és Unreal Engine (<https://gamefromscratch.com/game-engine-popularity-in-2024/>, 2024.06.22). A társasjáték nem igényel 3D grafikát, ezért a kezdőbarátabb (<https://www.incredibuild.com/blog/unity-vs-unreal-what-kind-of-game-dev-are-you>, 2024.06.22) Unity mellett döntöttem. Az engine miatt C# a fejlesztés nyelve. Az architektúra ECS (Entity Component System), szintén Unity miatt (<https://unity.com/dots>, 2024.06.22.)

ECS alapok:

<https://github.com/Unity-Technologies/EntityComponentSystemSamples>, 2024.10.05

Kártyalogika

alap:

How to make a card game – Unity Tutorial 2022 (<https://www.youtube.com/watch?v=C5bnWShD6ng>) 2024.10.05

Teljesítmény javítás:

TODO: continue from 2nd video at https://github.com/Unity-Technologies/EntityComponentSystemSamples

job-ok használata: unity-ben a nem főszálakat job-ok segítségével használhatjuk. Ezekre is lehet delegálni feladatokat, releváns lehet ha AI túl sokáig gondolkodna. Akkor lehetne használni például igazán, ha a döntéshez használt heurisztikus értéke egymástól függetlenül kiszámolható heurisztikus értékek [gyors többváltozós függvénye (pl. lin. komb.)] lenne.

<https://docs.unity3d.com/Manual/JobSystem.html>, 2024.10.05

burst compilation – compilation technika job-okkal, javíthat teljesítményen (<https://docs.unity.cn/Packages/com.unity.burst@1.8/manual/index.html>), 2024.10.05

assembly definition/references – scriptek egymástól elkülönülő assembly-kbe csoportosítása. Az assembly-knek előre megadjuk, melyiknek mire van referenciája. Így módosításkor csak az adott assembly-t, és az erre referáló assembly-ket kell újracompile-olni. Alapesetben mindennel tenné ezt meg a unity, tehát sok script esetén feltűnő ennek az eszköznek az alkalmazása  
(<https://docs.unity3d.com/Manual/ScriptCompilationAssemblyDefinitionFiles.html>), 2024. 10. 05.

scriptableobject

adattároló osztály, akkor hasznos, amikor ugyanaz a „prefab” többször van generálva. Ekkor ahelyett, hogy minden példány külön lemásolja az azonos adatot, a scriptableobject-re referálhatnak a példányok a tényleges adatért. Módosítók, próbatételek, eszközök, varázslatok esetén releváns lehet.

<https://docs.unity3d.com/Manual/class-ScriptableObject.html>, 2024.10.07

icomponentdata

interfész, ami a komponensek adat részéért felelős. Hasznos ECS architektúrában az adatok és viselkedések szétválasztásáért

<https://copilot.microsoft.com/sl/eaA2EpblAbc> , 2024.10.07

systems/SystemState

system a legnagyobb logikai egység az ECS-ben, és érdemes általában az adott világ (várhatóan csak 1 lesz, az alapértelmezett) system-jeinek a kezeléséhez (query, component type handles) a SystemState attribútumaikat használni. Ezzel lehet megfelelő szekvenciális működést biztosítani system-ek között is. Amint megvan mik lesznek nálam a system-ek, itt lép életbe a koncepció

<https://youtu.be/k07I-DpCcvE?si=MIB8dZeAwWFy21OM>, 2024.10.07

baking

ha kódolás közben a teljesítmény nagyon gáz, akkor erről érdemes lehet utánaolvasni

<https://docs.unity3d.com/Packages/com.unity.entities@1.0/manual/baking-overview.html>, 2024. 10. 07.

com.unity.collections

Ha bármely konténer használata nagyon lassú, érdemes megfontolni ennek a package-nek az importálását

<https://docs.unity3d.com/Packages/com.unity.collections@2.5/manual/index.html>, 2024. 10. 07.

Heurisztika:

Nagyon magas:

* közvetlenül nyer a saját játékos
  + 6. kaszt megszerzése (1)
    - 6. kaszt kijátszása (1)
    - maszk kijátszása (1)
    - 6. kaszt ellopása (2)
      * varázslat (1)
      * hős kijátszás/aktiválás (2)
    - mindezeket meghiusító próbatételekre módosítás (sajátnak felfele, próbatételező ellen lefele) (3)
  + 3. szörny legyőzése (2)
    - módosítás (általában felfele)
  + ha fenti kettő közül bármelyik releváns (0)
    - dobást növelő effektusok aktiválása (1)
      * varázslat (1)
      * hős kijátszás/aktiválás (2)
    - „kevés” módosító kártya esetén kártyahúzásos effektusok aktiválása
* közvetlenül megakadályozza más játékos nyerését
  + 6. kaszt megszerzése
    - kijátszás próbatételezése
    - maszk próbatételezése
    - lopás próbatételezése (varázslat) / módosítása (hősképesség aktiválás, csak lefele)
    - fenti 3 esetben próbatételre módosítás (nyerni készülő ellen lefele, próbatételező ellen felfele)
  + 3. szörny legyőzése
    - módosítás nyeréstől el (általában lefele)

# Felhasználói dokumentáció

# Fejlesztői dokumentáció

## Funkcionális követelmények

### Játék előtt

* Játékosszám kiválasztása (2-4)
* Játék indítása

### Játék közben

#### Játéktér

* Négyzet alapú pálya, játékosok a négyzet oldalain
* Játékos saját kezében lévő kártyáit, és bárkinek a kijátszott kártyáit látja
* Játékos többiek kezében lévő kártyáinak csak a hátlapját látja
* Bal alsó sarokban szöveges leírása a megtörtént eseményeknek (innentől log)
* Kockadobás gomb log-tól jobbra
* Kör vége gomb jobb alsó sarokban
* Akciópontok száma kör vége gombtól balra
* Középen húzópakli (hátlap), dobópakli (látható), szörnypakli (hátlap), kirakott szörnyek (látható)
* reakció lehetőségkor új gomb felugrik jobb szélen arra, ha szeretnénk kiválasztott kártyával reagálni

#### Általános

#### Kártyák

* húzhatóak innen: húzópakli, dobópakli, kijátszott kártya (bárkié), kézben lévő kártya (bárkié), kirakott szörnyek
* kézben tartott kártyák sohasem feláldozhatóak, elp

##### Csapatvezető kártyák

* Minden játékos a játék elején véletlenszerűen kap egy csapatvezető kártyát, ami rendelkezik egy kaszttal és egyedi képességgel. Ez a játék végéig erősíti a játékost.
* Ez a kártya nem pusztítható, sem lopható, sem feláldozható,

##### Körön belül kijátszható kártyák

##### Körön kívül is kijátszható kártyák

##### Szörnykártyák

* Akkor támadható a lerakott szörnykártya, ha
  + a támadó játékosnak a kijátszott hőskártyái (innentől party) megfelel a szörnyön szereplő elvárásoknak
* A rajta szereplő bónusz a játék végéig erősíti a játékost aki sikeresen megtámadta. Ekkor az a szörny a játékoshoz kerül, és a szörnypakliból új kártya pótolja a szörnysereget.
* Ez a kártya nem pusztítható, sem lopható, sem feláldozható, sem eldobható

#### Játékos saját körében

* 3 akcióponttal kezdi minden játékos a körét. Kör végéig legfeljebb ennyit használhat. A nem használt pontok a kör végén elvesznek.

#### Játékos körén kívül

#### Robot játékosok

#### Játék vége

* Ha egy játékosnál bármelyik feltétel teljesül, azonnal véget ér a játék
* A nyertes neve új ablakban megjelenik
* Főmenübe visszalépés

##### 6 kaszt kijátszva

##### 3 szörny legyőzve

# Összefoglalás és további fejlesztési lehetőségek

# Irodalomjegyzék

# Melléklet