**3. Analízis modell kidolgozása I.**

10 – Extra

Konzulens:

Dobos-Kovács Mihály

Csapattagok

|  |  |  |
| --- | --- | --- |
| **Draskóczi Dóra Boglárka** | **CTWF8V** | **ddb.dora@gmail.com** |
| Halász Máté | ZINVFE | matejjhalasz@gmail.com |
| Molnár Martina | BFCYOE | 1011molnarmartina@gmail.com |
| Pigler András | XERZQ0 | [andras.pigler@gmail.com](mailto:andras.pigler@gmail.com) |
| Serfőző Dávid | F05MHO | [serdavid3010@gmail.com](mailto:serdavid3010@gmail.com) |

2024. 03. 04.

# Analízis modell kidolgozása

## Objektum katalógus

### Labirintus

Ez lesz a játék színtere. Szobákból épül fel, melyeket ajtók kapcsolnak össze egymással, felépítése időnként változhat. A játékosok itt játszák le a teljes játékot.

### Karakter

Eltérő tulajdonságokkal és célokkal rendelkező játékos, aminek döntéseit vagy a felhasználók, vagy a gép határozza meg. Képes tárgyakat felvenni és letenni.

### Oktató

Célja, hogy akadályozza a hallgatókat a Logarléc megszerzésében. Véletlenszerűen mozog a szobák között és véletlenszerűen tudja a szobákban levő tárgyakat felvenni/letenni. Ha egy vagy több hallgatóval kerül egy szobába és azok nem tudnak védekezni ellene valamilyen tárggyal, akkor az ott lévő hallgatók elveszítik a játékot.

### Hallgató

A felhasználó irányítja, célja a Logarléc megtalálása és felvétele adott időn belül. Tud szobák között mozogni, képes a bennük lévő tárgyakat felvenni, azokat – a megengedett mennyiségig – tárolni, illetve használni/aktiválni. Ha oktatóval találkozik és nincsen nála használható tárgy, elveszti a játékot.

### Szoba

A játék szobákból épül fel, ezek közt mozoghatnak a hallgatók és oktatók. Egy-egy szobából legalább egy, de esetenként sok másik szobába is nyílhat ajtó. A szobákban különféle tárgyak lehetnek. Vannak tulajdonságaik, amelyek közt van olyan ami mindegyik szobára érvényes - ilyen például a befogadóképességük -, és vannak olyanok melyek csak egyesekre igazak: mérges gázos vagy elátkozott szoba.

Mérges gázos szobába belépő hallgatók és oktatók egy rövid időre eszméletüket vesztik és a náluk lévő tárgyakat elejtik. Ha van náluk *FFP2-es maszk* nevezetű tárgy (3.1.11), az megvédi őket az ájulástól.

Elátkozott szobák szomszédaihoz vezető ajtók időközönként eltűnnek/megjelennek. A szobák szomszédai ekkor nem változnak meg.

A szobák képesek egyesülni és osztódni, de az előbbi csak már szomszédos szobák közt lehetséges. Szobák egyesülésekor mindkét szoba tulajdonságával és szomszédaival az új szoba is rendelkezni fog, befogadóképessége pedig a nagyobb befogadóképességel rendelkező szobáéval lesz azonos.Szoba osztódásakor, a két szoba szomszédos lesz egymással és mindkét szoba rendelkezni fog az eredeti tulajdonságaival. A szomszédokat és a bennük lévő tárgyakat a szobák véletlenszerűen fogják megkapni.

### Tárgyak

Mind az oktató, mind a hallgató fel tudja venni ezeket a szobákban. Különleges és különféle erőkkel bírnak, melyekkel befolyásolni tudják környezetüket vagy elősegíthetik a hallgatókat és hátráltatják az oktatókat céljaik elérésében. Általában aktiválni lehet őket, hogy fel lehessen képességüket használni. Egyes tárgyaknak ’lejárati idejük’ is van, aminek letelte után elveszítik adottságukat és eltűnnek.

### Logarléc

A játék során a hallgatók célja megtalálni ezt az ereklyét. Amint felvették, megnyerték a játékot.

### Tranzisztorok

Teleportálni tudják aktiválójukat egyik szobából a másikba. A szobák szomszédossága nem releváns a működésükre. Használatuk előtt páronként csatlakoztatni kell őket, összekapcsolás nélkül nincs különleges képességük. Az összekapcsolt pár egy tagját a hallgató leteheti egy szobába, majd haladhat tovább (a tranzisztorok korlátlan ideig használhatók, így bármennyi idő eltelhet a képesség felhasználása nélkül). Ha a hallgató a nála maradó tranzisztort bekapcsolja és leteszi, akkor a másik tranzisztor szobájába kerül, a bekapcsolt tranzisztor pedig kikapcsol. Kikapcsoláskor a két tranzisztor közti kapcsolat megszűnik.

### Szent Söröspohár

A tárgy adott ideig védettséget nyújt a hallgatóknak az oktatók ellen.

### TVSZ denevérbőrre nyomtatott példányai

A tárgy három alkalommal menti meg a hallgató életét az oktatóval való találkozás során.

### Nedves táblatörlő rongy

A tárgy képessége, hogy míg ki nem szárad, teljesen megbénítja a vele egy szobában levő oktatókat, így azok sem lelket elvenni, sem mozogni, sem tárgyakat felvenni vagy eldobni nem képesek.

### Dobozolt káposztás camembert

A tárgy képessége annak környezetét befolyásolja. Kinyitása után a szoba egyik tulajdonsága (ha eddig nem volt gázos) megváltozik és gázos lesz (3.1.1). Használata után (miután ki lett nyitva), a doboz eltűnik.

### FFP2-es maszk

A tárgy képessége védelmet biztosít gázos szobák ellen. Nem kell aktiválni, ha a hallgató belép egy ilyen tulajdonsággal rendelkező szobába, automatikusan felveszi azt. Használat során azonban veszít az erejéből, amint teljesen el lett használva, eltűnik.

## Statikus struktúra diagramok

## Osztályok leírása

### Beer

#### Felelősség

Egy tárgyat valósít meg. Védettséget nyújt a hallgatóknak oktatókkal szemben.

#### Ősosztályok Item → TimeLimitedItem

#### Interfészek

Nem valósít meg interface-t.

#### Asszociációk

#### Attribútumok

#### Metódusok

* **void ProtectAgainstTeacher():**Adott ideig megvédi a hallgatót az oktatóval szemben.

### Camambert

#### Felelősség

Egy tárgyat valósít meg. Megbénítja a vele egy szobában lévő, védettséggel nem bíró karaktereket.

#### Ősosztályok

Item

#### Interfészek

#### Asszociációk

#### Attribútumok

#### Metódusok

* **void gassing(Room r):**   
  Elgázosítja a szobát, amelyben a hallgató található, aki használja.
* **void disappear()**:   
  Megszünteti a tárgyat, miután felhasználták.

### Character

#### Felelősség

Absztrakt ősosztály, a hallgatóknak és az oktatóknak.

#### Ősosztályok

#### Interfészek

#### Asszociációk

* **asszociáció:** Túloldali szereplő: szoba. Több az egyhez kapcsolat.   
  Minden karakter minden pillanatban egy szobában tartózkodik.
* **irányított asszociáció:** Túloldali szereplő: Item. Egy a többhöz kapcsolat.  
  Minden karakter legfeljebb 5 tárggyal rendelkezhet.
* **kompozíció:** Túloldali szereplő: Tranzisztor. Egy a többhöz kapcsolat.  
  Minden karakternél legfeljebb 2 tranzisztor lehet.

#### Attribútumok

* **int itemCapacity:**
* **bool unconscious:** Tárolja, hogy az adott karakter éppen eszméletlen-e vagy sem.

#### Metódusok

* **void pickUp(Item item):**   
  Absztrakt függvény. A karakterek ezzel vehetnek fel egy-egy tárgyat.
* **void putDown(Item item):**Absztrakt függvény. Ezzel tesznek le egy-egy tárgyat a karakterek.
* **void move(Room r):**Absztrakt függvény. Az egyes karakterek szobák között mozognak. A paraméterként kapott szobába kerül át az adott karakter.
* **void faint():**  
  Ha a karakter ájult állapotba kerül, ez a függvény bénítja meg, dobatja el a tárgyait.

### FFP2

#### Felelősség

Egy tárgyat valósít meg. Védelmet nyújt a hallgatóknak adott ideig, néhány alkalommal, ha van a birtokukban ilyen tárgy.

#### Ősosztályok

Item – > TimeLimitedItem

#### Interfészek

#### Asszociációk -

#### Attribútumok

* **int useAge**: Tárolja, hogy hány alkalommal képes megvédeni egy karaktert a játék hátralévő részében.

#### Metódusok

* **void disappear**:   
  Megszünteti a tárgyat annak utolsó felhasználása után.
* **void setMask**:   
  Átállítja a maszk védő idejét és hogy hányszor használható még fel, miután használták.
* **void protectAgainstGas():**   
  Megvédi a hallgatót adott ideig.

### GameController

#### Felelősség

A játék belső működéséért felelős, figyeli, hogy tart-e még a játék, lépteti az egyes köröket, elindítja és befejezi a játékot.

#### Ősosztályok

#### Interfészek

#### Asszociációk -

#### Attribútumok

#### Metódusok

#### void startGame(): Elindítja a játékot.

#### void endGame: Befejezi a játékot, ha a hallgatók megszerezték a logarlécet.

* **void stepRound()**: Lépteti a köröket a játék során.

### Item

#### Felelősség

Az egyes tárgyak közös ősosztálya, amelyeket a karakterek felvehetnek a játék során. Céljuk eltérő, a legtöbb védelmet biztosít a hallgatóknak valamilyen formában.

#### Ősosztályok -

#### Interfészek

#### Asszociációk

* **asszociáció**: a túloldali szereplő: karakter. Több az egyhez kapcsolat.  
  A karakterek tárgyakat vehetnek magukhoz és a játék során használhatják azokat, vagy le is tehetik.
* **asszociáció**: a túloldali szereplő: szoba. Több az egyhez kapcsolat.  
  A szobákban helyezkednek el a tárgyak, ahol a hallgatók és oktatók hozzájuk férnek.

#### Attribútumok

* **enum type**: Tárolja az egyes tárgyak típusát.

#### Metódusok

* **void use():**A tárgyak felhasználhatók a játék során, amikor ez megtörténik, működésbe lépnek.
* **void onPickUp**:   
  Egy tárgy felvételekor hívódik meg, jelzi a játék felé, hogy valamelyik karakter tulajdonába került az eszköz.

### Logarléc

#### Felelősség

Egy tárgyat valósít meg, amint egy hallgató magához veszi a logarlécet, megnyeri a játékot.

#### Ősosztályok

Item

#### Interfészek

-

#### Asszociációk

-

#### Attribútumok

#### Metódusok

### Maze

#### Felelősség

Szobákból épül fel, ő valósítja meg a játék színterét. Felelőssége biztosítani a szobák közötti átjárást, vagy éppen megszüntetni annak lelehtőségét. Ő végzi a szobák egyesítését vagy szeparálását.

#### Ősosztályok -

#### Interfészek

#### Asszociációk

* **asszociáció**: Túloldali szereplő: GameController.  
  A játékot vezérlő osztály kezeli a szobák ”labirintusát”.
* **aggregáció**: Egy a többhöz. Túloldali szereplő: Room.  
  Szobák tartoznak hozzá.

#### Attribútumok

#### Metódusok

* **void mergeRooms():**   
  Két szobát egyesít, beállítja annak tulajdonságait a két szoba megfelelő paraméterei alapján.
* **void separateRoom()**:  
  Egy szobát két szomszédos szobára bont, beállítja azok tulajdonságait, szomszédait.

### Room

#### Felelősség

A ”labirintus” szobáit megvalósító osztály. Egy-egy szobából egy vagy akár több másikba is nyílhat ajtó, ismeri a szomszédait. Ezen felül tárgyakat tárol, amiket a karakterek felvehetnek és mozgásuk során átvihetnek egy másik szobába. Közöttük közlekednek a karakterek.

#### Ősosztályok

#### Interfészek

#### Asszociációk

* **aggregáció:** Túl oldali szereplő: Maze  
  A Maze tárolja a szobákat.
* **asszociáció:** Egy a többhöz kapcsolat. Túl oldali szereplő: Room, azaz saját maga.  
  A szobák ismerik egymás szomszédjait. Egy szobának lehet több szomszédja is, de akár egyetlen egy sem.
* **asszociáció:** Egy a többhöz kapcsolat. Túl oldali szereplő: Character.  
  A karakterek szobák között mozognak, így valamelyikben állandóan benne vannak. Egy szobában több karakter is lehet.
* **aggregáció:** Több a többhöz kapcsolat. Túl oldali szereplő: Transistor.   
  E
* **asszociáció:** Egy a többhöz. Túloldali szereplő: Item.  
  Egy szobában számtalan tárgy lehet, melyeket a karakterek felvehetnek.

#### Attribútumok

* **int maxCapacity:** A szoba befogadóképessége.
* **int capacity:** A szobában jelenleg tartózkodó karakterek létszáma.
* **bool poisonous:** Számon tartja, hogy az adott szoba éppen elgázosított-e vagy sem.
* **bool cursed:** Számon tartja, hogy az szoba elátkozott-e vagy sem, azaz, nyílnak-e belőle ajtók vagy sem.

#### Metódus

* **void remove(Character: ch)**:   
  Eltávolítja a szobában lévő játékosokat, ha azok át akarnak menni egy másik szobába, vagy teleportálnak egy tranzisztorral, vagy amennyiben kénytelenek átkerülni máshová, mert nem férnek be az adott szobába.
* **void accept(Character ch):**Beengedi a karaktereket a szobákba.

### Sponge

#### Felelősség

Egy tárgyat valósít meg, megbénítható vele egy oktató adott időre.

#### Ősosztályok

Item – > TimeLimitedItem

#### Interfészek

-

#### Asszociációk

-

#### Attribútumok

#### Metódusok

* **dry():**Amikor a táblatörlő rongy kiszárad, megsemmisíti önmagát.
* **paralyze(Teacher t):**Amikor egy hallgató felhasználja, megbénít adott időre egy oktatót.

### Student

#### Felelősség

#### Ősosztályok

Character

#### Interfészek

-

#### Asszociációk

* **Függőség:** Túloldali szereplő: Teacher  
  Egy oktató képes elvenni egy hallgató lelkét, így közvetlen hatása lehet rá a játék során.

#### Attribútumok

* **bool protectedAgainstTeacher:**Amennyiben egy hallgató felhasznál egy sörös poharat vagy a TVSZ-t, akkor védetté válik a vele egy szobában lévő oktatóval szemben.
* **bool protectedAgainstGas:**   
  Amennyiben a hallgatónál van FFP2-es maszk, védetté válik egy gázos szobában.

#### Metódusok

* **void pickUp(Item item):**A hallgató a játék során dönthet úgy, hogy felvesz egy tárgyat abban a szobában, amiben éppen tartózkodik
* **void putDown(Item item):**A felvételhez hasonlóan, le is tehet egy tárgyat egy hallgató abban a szobában, ahol éppen tartózkodik.
* **void useItem():**A hallgató felhasználhatja a nála lévő tárgyakat.
* **void move(Room r):**A játék során a hallgató mozog, ekkor átkerül egy másik szobába, mint ahol korábban volt.
* **void die():**Amennyiben a hallgató egy oktatóval kerül egy szobába, és nincs nála a védelmét szolgáló eszköz, az oktató elveszi a hallgató lelkét, ekkor ő kiesik a játékból.

### TimeLimitedItem

#### Felelősség

Olyan tárgyakat valósít meg, amelyek csak adott ideig biztosítanak valamilyen funkciót.

#### Ősosztályok

Item.

#### Interfészek

-

#### Asszociációk

-

#### Attribútumok

* **int time:** Tárolja, hogy az adott tárgy mennyi ideig van érvényben miután felhasználta egy karakter.

#### Metódusok

### Teacher

#### Felelősség

Egy karaktert valósít meg.

#### Ősosztályok

Character

#### Interfészek

-

#### Asszociációk

**Függőség:** Túloldali szereplő: Student  
Egy oktató képes elvenni egy hallgató lelkét, így közvetlen hatása lehet rá a játék során.

#### Attribútumok

* **bool paralized:** Tárolja, hogy az adott oktató ájult állapotban van-e vagy eszméleténél van.

#### Metódusok

* **void pickUp(Item item):**   
  Az oktatók is képesek tárgyakat felvenni abban a szobában, amiben tartózkodnak, ez náluk véletlenszerűen működik.
* **void putDown(Item item):**Hasonlóan a felvételhez, véletlenszerűen le is tehetnek tárgyakat az oktatók abban a szobában, ahol éppen tartózkodnak.
* **void move(Room r):**   
  Az oktatók is mozognak a szobák között, ám ez véletlenszerűen történik.
* **void kill(Student student):**Amikor az oktatók egy szobában tartózkodnak egy (vagy több hallgatóval) és nincsenek ájult vagy bénított állapotban, akkor elveszik a hallgatók lelkét.

### Transistor

#### Felelősség

Egy tárgyat valósít meg, amit a karakterek felvehetnek, kettőt összekapcsolhatnak, és teleportálhatnak vele bizonyos szobák között.

#### Ősosztályok

Item

#### Interfészek

-

#### Asszociációk

* **asszociáció:** Túloldali szereplő: egy másik tranzisztor.   
  Egy tranzisztor összekapcsolható egy másikkal.
* **aggregáció:** Nulla vagy kettő az egyhez. Túl oldali szereplő: Character  
  Egy karakternél semennyi vagy legfeljebb kettő tranzisztor lehet egyszerre.
* **aggregáció:** Nulla vagy egy a nulla vagy egyhez. Túl oldali szereplő: Room.  
  Ha letesznek egy olyan tranzisztort valamelyik szobába, amit előtte összekapcsoltak egy másik tranzisztorral, akkor annak ismernie kell a hozzákapcsolta tranzisztort és hogy melyik szobában tették le, mert oda kell majd tudnia teleportálni a hallgatónak, ha leteszi a másik tranzisztort.

#### Attribútumok

#### Metódusok

* **void connectTo(Transistor t):**Egy tranzisztor összekapcsolható egy másik tranzisztorral.

### TVSZ

#### Felelősség

Egy tárgyat valósít meg. Védelmet biztosít a hallgatóknak.

#### Ősosztályok

Item.

#### Interfészek

-

#### Asszociációk

-

#### Attribútumok

* **int useAge:** Tárolja, hogy az adott tárgy hányszor nem lett még felhasználva, mivel csupán adott számú alkalommal védhet meg egy hallgatót.

#### Metódusok

* **void protectAgainstTeacher():**Ha egy hallgató egy szobába kerül egy oktatóval, akkor megvédi a hallgatót attól, hogy az oktató el tudja venni a lelkét.

## Szekvencia diagramok

[Inicializálásra, use-case-ekre, belső működésre. Konzisztens kell legyen az előző alfejezettel. Minden metódus, ami ott szerepel, fel kell tűnjön valamelyik szekvenciában. Minden metódusnak, ami szekvenciában szerepel, szereplnie kell a valamelyik osztálydiagramon. Fontos, hogy az aktor use-case-eitől bármelyik diagramhoz el lehessen jutni a metódushívások követésével.]

## State-chartok

[Csak azokhoz az osztályokhoz, ahol van értelme. Egyetlen állapotból álló state-chartok ne szerepeljenek. A játék működését bemutató state-chart-ot készíteni tilos.]

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2024.02.29. 18:00 | 1 óra 15 perc | Draskóczi,  Halász,  Molnár,  Pigler,  Serfőző | Értekezlet.  Döntés: Pigler elkészíti az Objektum katalógust és egy kezdetleges vázat az osztálydiagramhoz,  Draskóczi, Molár és Serfőző összeülnek megtervezni a statukus struktúradiagramokat,  Serfőző készíti majd a szekvencia diagramot. |
| 2024.02.29. 20:00 | 2 óra | Pigler | Pigler elkészíti az Objektum katalógust és az osztálydiagram kiindulásául szolgáló vázát. |
| 2024.03.01. 14:30 | 3,5 óra | Draskóczi,  Molnár,  Serfőző | Értekezlet.  Tevékenység:  Az objektumkatalógus és a korábban elkészített use-case-ek alapján közösen godolkodva elkészítik az osztálydiagramot és egyeztetik, hogy melyik osztály milyen adattagokkal, illetve függvényekkel rendelkezzen.  Döntés:  Serfőző befejezi az osztálydiagramot és belekezd a szekvenciadiagramokba. A fennmaradó kérdéseket a csapat többi tagjával átbeszéljük. Molnár kiegészíti az objektum katalógust, Draskóczi megírja az osztályok leírását. |
| 2024.03.01. 19:00 | 2 óra | Molnár | Molnár kiegészíti az objektum katalógust. |
| 2024.03.01. 19:00 | 2 óra | Serfőző | Serfőző elkészíti a végleges osztálydiagramot. |
| 2024.03.02.11:00 | 1 óra | Draskóczi,  Halász,  Molnár,  Pigler,  Serfőző | Értekezlet.  Döntés: A még kérdéses attribútumok, függvények, illetve osztályok átbeszélése, Serfőző és Halász elkészíti a szekvencia diagramokat. |