

## 3. Analízis modell kidolgozása I.

10 – Extra

Konzulens:

Dobos-Kovács Mihály

### Csapattagok

**Draskóczy Dóra Boglárka**

Halász Máté

Molnár Martina

Pigler András

Serfőző Dávid

**CTWF8V**

ZINVFE

BFCYOE

XERZQ0

F05MHO

**ddb.dora@gmail.com**

matejjhalasz@gmail.com

1011molnarmartina@gmail.com

andras.pigler@gmail.com

serdavid3010@gmail.com

2024. 03. 04.

## 3. Analízis modell kidolgozása

### 3.1 Objektum katalógus

#### 3.1.1 Labirintus

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

#### 3.1.2 Karakter

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

#### Oktató (Enemy)

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 (Player)

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.

#### 3.1.3 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 veszítik 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 befogadóképességgel 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.

#### 3.1.4 Tárgyak

Mind az oktató, mind a hallgató fel tudja venni ezeket a szobákban. Különleges és különféle erőkkal 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.

### 3.1.5 Logarléc

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

### 3.1.6 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.

### 3.1.7 Szent Söröspohár

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

### 3.1.8 TVSZ denevérbőrre nyomtatott példányai

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

### 3.1.9 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.

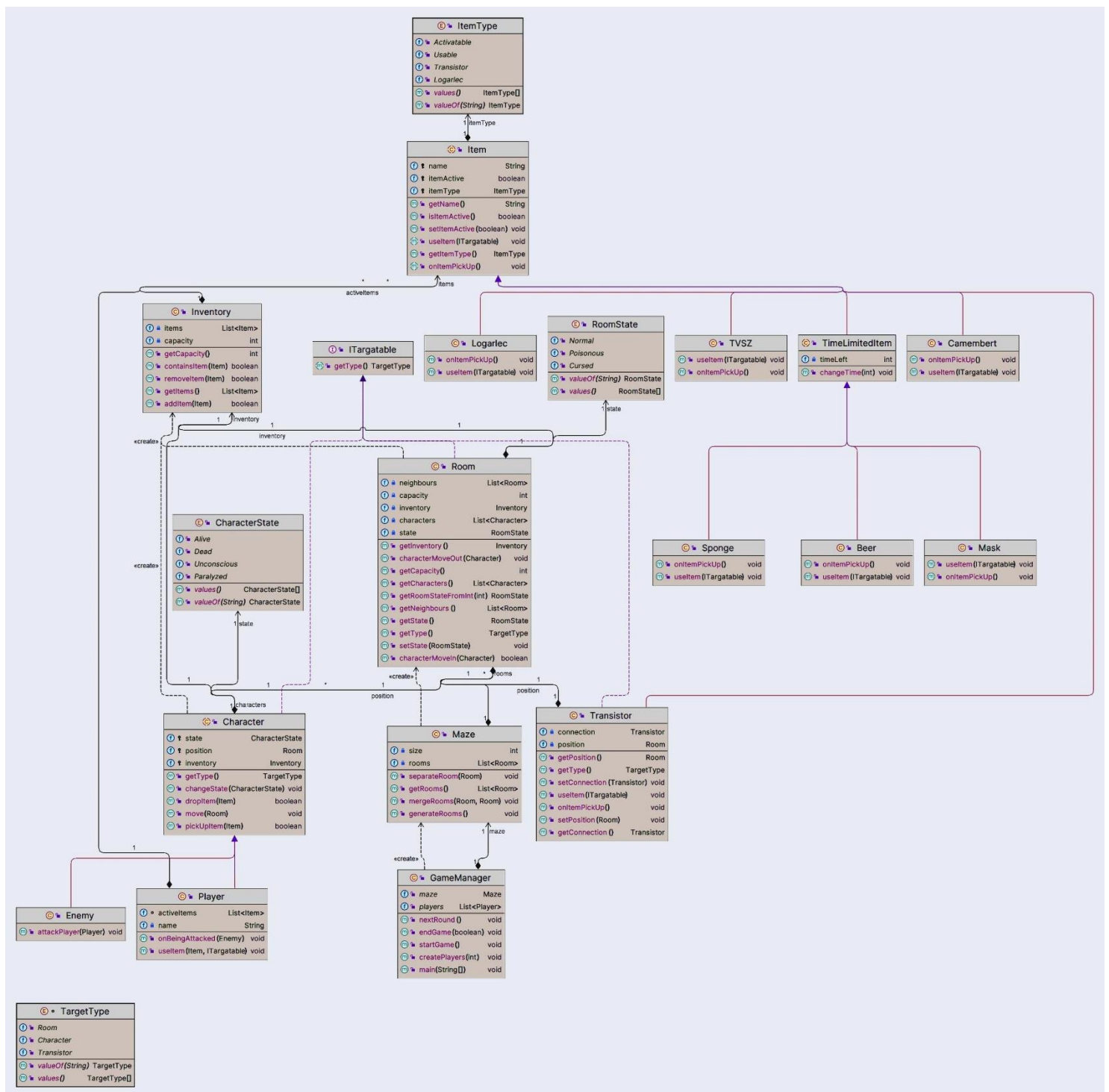
### 3.1.10 Dobozolt káposztás camembert

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.

### 3.1.11 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.

## 3.2 Statikus struktúra diagramok



Az ábra a dokumentum végén darabokra bontva is megtalálható a könnyebb áttekinthetőség és olvashatóság érdekében.

## Osztályok leírása

**3.2.1 Beer**

- **Felelősség**  
Egy tárgyat valósít meg. Védettséget nyújt a hallgatóknak oktatókkal szemben.
- **Ősosztályok**  
Item → TimeLimitedItem
- **Interfészek**  
-
- **Asszociációk**  
-
- **Attribútumok**  
-
- **Metódusok**
  - **void onItemPickUp():**  
Meghívódik, amikor a tárgyat felvették.
  - **void useItem(ITargetable iTarget):**  
Akkor hívódik, amikor egy hallgató használni szeretné a tárgyat.

**3.2.2 Camambert**

- **Felelősség**  
Egy tárgyat valósít meg. Megbénítja a vele egy szobában lévő, védettséggel nem bíró karaktereket.
- **Ősosztályok**  
Item
- **Interfészek**  
-
- **Asszociációk**  
-
- **Attribútumok**  
-
- **Metódusok**
  - **void useItem():**  
Akkor hívódik, amikor egy hallgató használni szeretné a tárgyat.
  - **void onItemPickUp():**  
Meghívódik, amikor a tárgyat felvették.

**3.2.3 Character**

- **Felelősség**  
Absztrakt ősosztály, a hallgatóknak és az oktatóknak.
- **Ősosztályok**  
-
- **Interfészek**  
ITargetable
- **Asszociációk**
  - **asszociáció:** Túloldali szereplő: Inventory.
  - **aggregáció:** Egy az egyhez kapcsolat. Túloldali szereplő: CharacterState.

- **aggregáció:** Egy az egyhez kapcsolat. Túloldali szereplő: Room  
A szoba tárol egyetlen karakterekből álló listát.
- **Attribútumok**
  - **private CharacterState state**
  - **private Room position**
  - **private Inventory inventory**
- **Metódusok**
  - **void pickUp(Item item):**  
A karakterek ezzel vehetnek fel egy-egy tárgyat.
  - **boolean dropItem(Item item):**  
Ezzel tesznek le egy-egy tárgyat a karakterek.
  - **void move(Room r):**  
Az egyes karakterek szobák között mozognak. A paraméterként kapott szobába kerül át az adott karakter.
  - **void changeState(CharacterState chState):**  
Megváltoztatja a karakter állapotát a 4 állapot valamelyikére.
  - **TargetType getType():**

### 3.2.4 CharacterState

- **Felelősség**  
Nyilvántartja, hogy az egyes karakterek élnek-e még, eszméletüknél vannak-e, vagy esetlegesen meg vannak-e bénítva.
- **Ősosztályok**  
-
- **Interfészek**  
-
- **Asszociációk**
  - **aggregáció:** Egy az egyhez típusú kapcsolat. Túloldali szereplő: Character.
- **Attribútumok**
  - **public enum Alive**
  - **public enum Dead**
  - **public enum Unconscious**
  - **public enum Paralyzed**
- **Metódusok**
  - **CharacterState[] values():**
  - **CharacterState valueOf(String value):**  
Megadja karakter állapját.

### 3.2.5 Enemy

- **Felelősség**  
Karaktert valósít meg, a játékon belül a rendszer irányítja.
- **Ősosztályok**  
Item
- **Interfészek**  
-

- **Asszociációk**
  -
- **Attribútumok**
  -
- **Metódusok**
  - **void attackPlayer(Player p):**  
Amennyiben egy szobában van egy Player-rel, és nincs megbénítva, továbbá a Player nincs védve vele szemben, megöli a hallgatót.

### 3.2.6 GameManager

- **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**
  - **asszociáció:** Túloldali szereplő: Maze
  - **aggregáció:** Túloldali szereplő Maze. Egy az egyhez kapcsolat.  
A játék irányítója egyetlen labirintussal rendelkezik, ebben épülnek fel a szobák és jönnek létre a játék elemei.
- **Attribútumok**
  - **public Maze maze**
  - **public List<Player> players**
- **Metódusok**
  - **void nextRound():**  
Lépteti a játék menetét egyik játékosról a másikra.
  - **void endGame(boolean end):**  
A játék végén hívodik meg, paraméterként megkapja, hogy a játékosok nyertek-e vagy sem.
  - **void startGame():**  
Elindítja a játékot, felépíti a pályát (maze), feltölti tárgyakkal (Item), elhelyezi a hallgatókat a szobákban.
  - **void createPlayers(int player):**  
Megalkotja a játékosokat, mind az oktatókat, mind a hallgatókat.
  - **void main(String[] str)**

### 3.2.7 Inventory

- **Felelősség**  
A karakterek, illetve szobák tárhelyét valósítja meg.
- **Ősosztályok**
  -
- **Interfészek**
  -

- **Asszociációk**
  -
- **Attribútumok**
  - **private List<Item> items:**  
A tárhelyben lévő tárgyakat listázza.
  - **private int capacity:**  
A tárhely kapacitását határozza meg.
- **Metódusok**
  - **int getCapacity():**  
Megadja, a tárhely kapacitását.
  - **boolean containsItem(Item i):**  
Megmondja, hogy van-e az adott tárgyból a tárhelyen.
  - **boolean removeItem(Item i):**  
Eltávolítja a tárgyat a tárhelyből.
  - **List<Item> getItems():**  
Visszaadja, hogy mely tárgyak szerepelnek az adott tárhelyen.
  - **boolean addItem(Item i):**  
Beteszt egy újabb tárgyat a tárhelybe amennyiben nem lépi túl ezzel a kapacitást.

### 3.2.8 Itargetable

- **Felelősség**

Interfész, amely jelölésként szolgál a tárgyak által interaktálható objektumokra.
- **Ősosztályok**
  -
- **Interfészek**
- **Asszociációk**
  - **asszociáció:** Túloldali szereplő: Characters
  - **asszociáció:** Túloldali szereplő: Transistor
  - **asszociáció:** Túloldali szereplő: Room
- **Attribútumok**
  -
- **Metódusok**
  - **TargetType getType():**

### 3.2.9 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**
  - **aggregáció:** a túloldali szereplő: ItemType. Egy az egyhez kapcsolat.
- **Attribútumok**
  - **private ItemType itemType:** Tárolja az egyes tárgyak típusát.
  - **private String name:** A tárgyak neve.



- **private boolean itemActive:** Tárolja, hogy éppen használatban van-e az adott tárgy
- **Metódusok**
  - **void useItem():**  
A tárgyak felhasználhatók a játék során, amikor ez megtörténik, működésbe lépnek.
  - **String getName():**  
Visszaadja az adott tárgy nevét.
  - **boolean isItemActive():**  
Megmondja, hogy működésben van-e éppen az adott tárgy.
  - **void setItemActive(boolean b):**  
Aktiválja a tárgyat.
  - **ItemType getItemType():**  
Visszaadja a tárgy típusát.
  - **void onItemPickUp():**  
Egy tárgy felvételekor hívódik meg, jelzi a játék felé, hogy valamelyik karakter tulajdonába került az eszköz

### ItemType

- **Felelősség**  
A különféle felhasználású tárgyakat tudja jelölni, például, hogy aktiválható-e a tárgy (hosszabb ideig tart a hatása) vagy egyszeri használatra képes csak.
- **Össztályok**  
-
- **Interfészek**  
-
- **Asszociációk**
  - **aggregáció:** Egy az egyhez. Túloldali szereplő: Item
- **Attribútumok**
  - **public enum Activatable**
  - **public enum Usable**
  - **public enum Transistor**
  - **public enum Logarlec**
- **Metódusok**
  - **ItemType[] values():**
  - **ItemType valueOf(String s):**

#### 3.2.10 Logarléc

- **Felelősség**  
Egy tárgyat valósít meg, amint egy hallgató magához veszi a logarléce, megnyeri a játékot.
- **Össztályok**  
Item
- **Interfészek**  
-
- **Asszociációk**  
-
- **Attribútumok**  
-

- **Metódusok**
  - **void onItemPickUp():**
  - **void useItem(ITargetable):**

### 3.2.11 Mask

- **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**  
-
- **Metódusok**
  - **void useItem(ITargetable iTarget):**
  - **void onItemPickUp():**

### 3.2.12 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**
  - **private int size:**  
A szoba befogadóképessége (fő).
  - **private List<Room> rooms:**  
A pályán található szobák listája.
- **Metódusok**
  - **void mergeRooms(Room r1, Room r2):**  
Két szobát egyesít, beállítja annak tulajdonságait a két szoba megfelelő paramétereire alapján.
  - **void separateRoom(Room r):**  
Egy szobát két szomszédos szobára bont, beállítja azok tulajdonságait, szomszédait.
  - **List<Room> getRooms():**  
Visszaadja a szobák listáját.

- **void generateRooms():**  
Létrehozza a szobákat.

### 3.2.13 Player

- **Felelősség**  
A játékosok által irányítható karakter.
- **Ősosztályok**  
Character
- **Interfészek**  
ITargetable
- **Asszociációk**
  - **asszociáció:** Egy a többhöz. Túloldali szereplő: Item  
A játékosnak tárgyak állnak rendelkezésére.
  - **aggregáció:** Egy az egyhez. Túloldali szereplő: Inventory  
A tárgyakat pedig az Inventory-jában tárolja.
- **Attribútumok**
  - **List<Item> activeItems:**  
A játékosnál jelenleg aktivált tárgyak listája.
  - **private String name:**  
A játékos neve.
- **Metódusok**
  - **void onBeingAttacked(Enemy enemy):**  
Akkor hívódik meg, ha egy ellenfél megtámadta őt, itt ellenőrzi, hogy milyen védettsége van esetlegesen a támadás kivédésére vagy ha nincs semmi, kiesik a játékból.
  - **void useItem(Item i, ITargetable it):**  
A játékos ezen függvény segítségével használ tárgyakat, paraméterként a használt tárgyat, illetve a célpontot kapja meg.

### 3.2.14 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. Köztük közlekednek a karakterek.
- **Ősosztályok**  
-
- **Interfészek**  
ITargetable
- **Asszociációk**
  - **asszociáció:** Túloldali szereplő: Inventory
  - **asszociáció:** Túloldali szereplő: ITargetable
  - **aggregáció:** Egy az egyhez kapcsolat. Túloldali szereplő: RoomState
  - **asszociáció:** Túloldali szereplő: Maze  
A Maze tárolja a szobákat.
  - **aggregáció:** Túloldali szereplő: Transistor.
  - **aggregáció:** Túloldali szereplő: Character  
A karakterek szobák között mozognak, így valamelyikben állandóan benne vannak. Egy szobában több karakter is lehet

- **Attribútumok**
  - **private List<Room> neighbours:**  
A szoba szomszédait tartalmazó lista.
  - **private int capacity:**  
A szoba befogadóképessége.
  - **private Inventory inventory:**  
A szoba saját tárhelye, amiben a tárgyak vannak.
  - **private List<Character> characters:**  
A szobában tartózkodó karakterek listája.
  - **private RoomState state:**  
A szoba állapota (normál, elgázosított, stb..).
- **Metódusok**
  - **Inventory getInventory():**
  - **void characterMoveOut(Character c):**  
Kiteszi a karaktert az adott szobából.
  - **int getCapacity():**  
Lekérdezhető vele a szoba kapacitása.
  - **List<Character> getCharacters():**  
Kilistázza a szobában tartózkodó karaktereket.
  - **RoomState getRoomStateFromInt(int roomState):**  
Megadja, hogy a szoba milyen állapotban van éppen, pl. elgázosított-e.
  - **List<Room> getNeighbours():**  
Kilistázza az adott szobához tartozó szomszédokat.
  - **RoomState getState():**
  - **TargetType getType():**
  - **void setState(RoomState roomState):**  
A szoba állapotát leehet vele változtatni.
  - **boolean characterMoveIn(Character ch):**  
Egy karaktert elhelyez a szobában, visszaadja, hogy sikerült-e vagy sem. (A szoba töltöttségétől függhet.)

### 3.2.15 RoomState

- **Felelősség**  
A szobák állapotát segít kezelni.
- **Össztályok**  
-
- **Interfészek**  
-
- **Asszociációk**
  - **aggregáció:** Egy az egyhez. Túloldali szereplő: Room  
Kapcsolat célja: Nyilvántartani, hogy egy-egy szobára éppen milyen állapot áll fenn.
- **Attribútumok**
  - **public enum Normal**
  - **public enum Poisonous**
  - **public enum Cursed**
- **Metódusok**

- **RoomState valueOf(String state):**
- **RoomState[] values():**

### 3.2.16 **Sponge**

- **Felelősség**  
Egy tárgyat valósít meg, megbénítható vele egy oktató adott idő.
- **Ősosztályok**  
Item → TimeLimitedItem
- **Interfészek**  
-
- **Asszociációk**  
-
- **Attribútumok**  
-
- **Metódusok**
  - **void onItemPickUp():**  
Meghívódik, amikor egy karakter felveszi a tárgyat.
  - **void useItem(ITargetable iTarget):**  
Meghívódik, amikor egy karakter használni szeretné a tárgyat.

### 3.2.17 **TargetType**

- **Felelősség**  
A célpontokként is megadható objektumokat segít beazonosítani.
- **Ősosztályok**  
-
- **Interfészek**  
-
- **Asszociációk**  
-
- **Attribútumok**
  - **public enum Room**
  - **public enum Character**
  - **public enum Transistor**
- **Metódusok**
  - **TargetType valueOf(String str):**
  - **TargetType values():**

### 3.2.18 **TimeLimitedItem**

- **Felelősség**  
Olyan tárgyakat valósít meg, amelyek adott ideig biztosítanak valamilyen funkciót.
- **Ősosztályok**  
Item
- **Interfészek**  
-

- **Asszociációk**
  -
- **Attribútumok**
  - **int timeLeft:**  
Tárolja, hogy az adott tárgy mennyi ideig van még érvényben.
- **Metódusok**
  - **void changeTime(int ct):**  
Átállítja azt az időtartamot, ami a tárgy idejéből hátravan. Ez lehet akár pozitív és negatív irányba is.

### 3.2.19 Transistor

- **Felelősség**  
Egy olyan tárgy, amely a játékosok mobilitásához segít hozzá. Két transistor között tudnak teleportálni, ha össze vannak kapcsolva és valamelyik le van helyezve egy adott szobában.
- **Ősosztályok**  
Item
- **Interfészek**  
ITargetable
- **Asszociációk**
  - **aggregáció:** Egy az ??? Túloldali szereplő: Kapcsolat célja.
- **Attribútumok**
  - **private Transistor connection:**  
Az éppen összekapcsolt tranzisztor párja (lehet null, ha nincs).
  - **private Room position:**  
A szoba, amelyben esetlegesen aktiválva lett (szintén lehet null, ha még nem aktiválták).
- **Metódusok**
  - **Room getPosition():**
  - **TargetType getType():**
  - **void setConnection(Transistor tr):**  
Felállítja a kapcsolatot két transistor között.
  - **void useItem(ITargetable itarget):**  
Használatba lép a tárgy. Ennek három esete, ha még nincs összekapcsolva, akkor összekapcsolja egy másikkal. Ha már össze van, de még nincs lehelyezve, akkor lehelyezi. Ha már van kapcsolat és le is van helyezve, akkor teleportál.
  - **void onItemPickUp():**
  - **void setPosition(Room r):**  
Beállítja, hogy melyik szobába aktiválták a tárgyat.
  - **Transistor getConnection():**  
Visszaadja a jelenlegi párját.

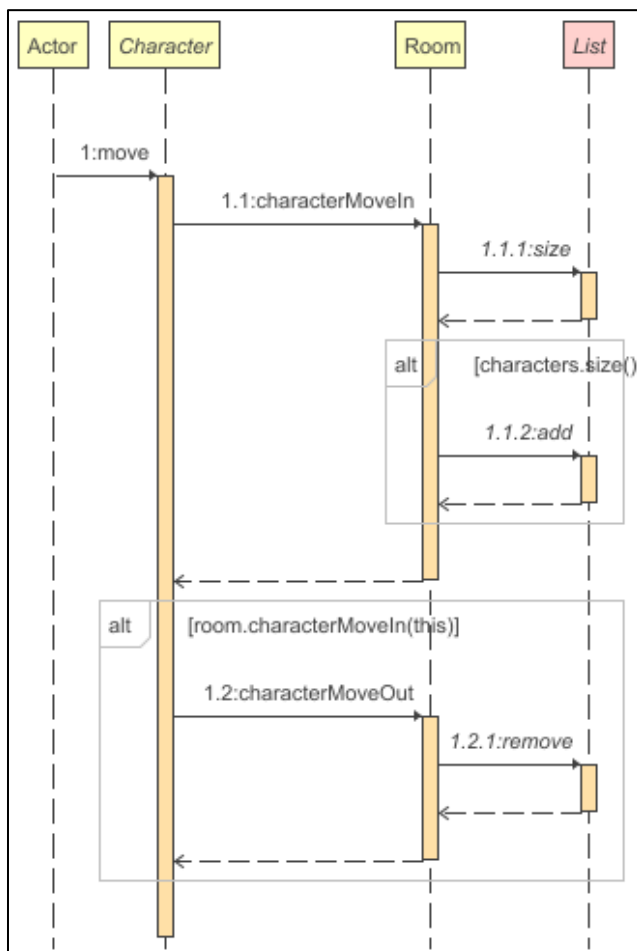
### 3.2.20 TVSZ

- **Felelősség**  
Egy tárgyat valósít meg, amely megvédi a hallgatót attól, hogy az oktató el tudja venni a lelkét, ha egy szobába kerülnek.

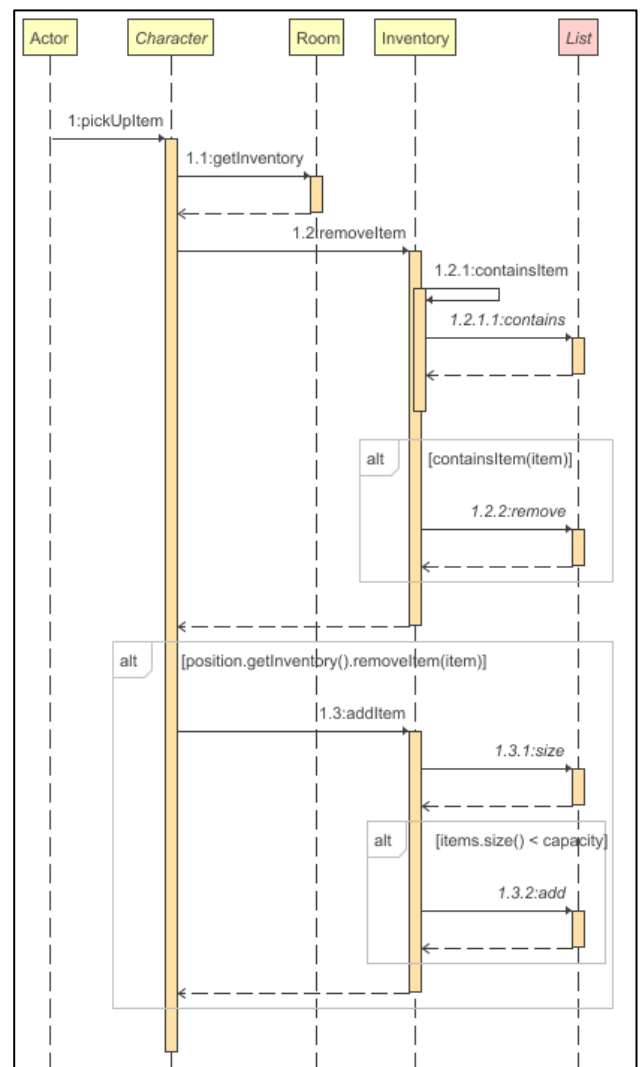
- **Össztályok**  
Item
- **Interfészek**  
-
- **Asszociációk**  
-
- **Attribútumok**  
-
- **Metódusok**
  - **void useItem(ITargetable itarget):**  
Akkor hívódik meg, ha használni szeretné egy karakter.
  - **void onItemPickUp():**  
Akkor hívódik, amikor egy karakter felveszi a tárgyat.

## Szekvencia diagramok

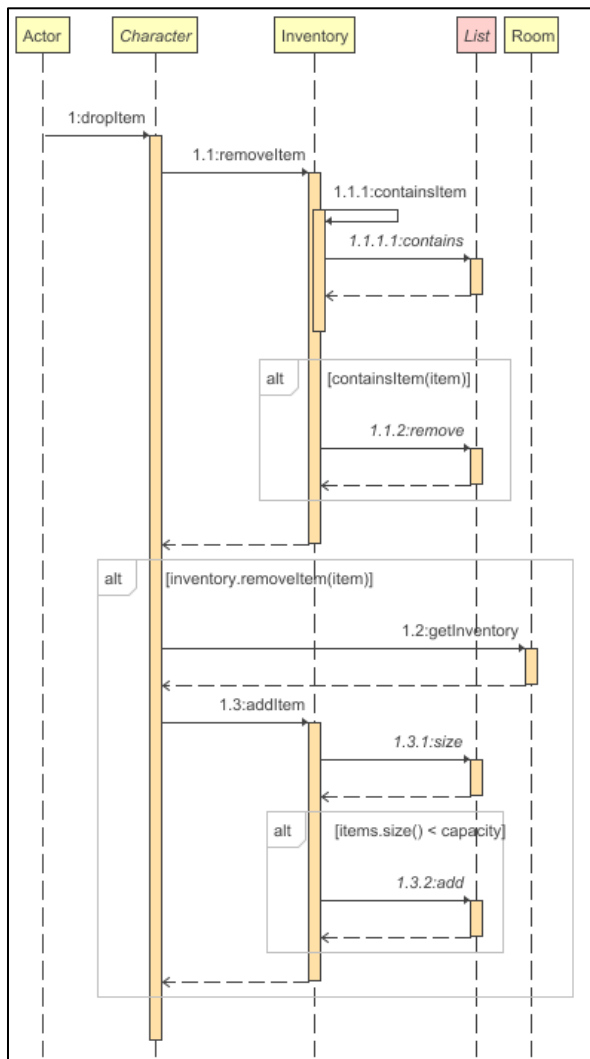
3.4.1 Character moves



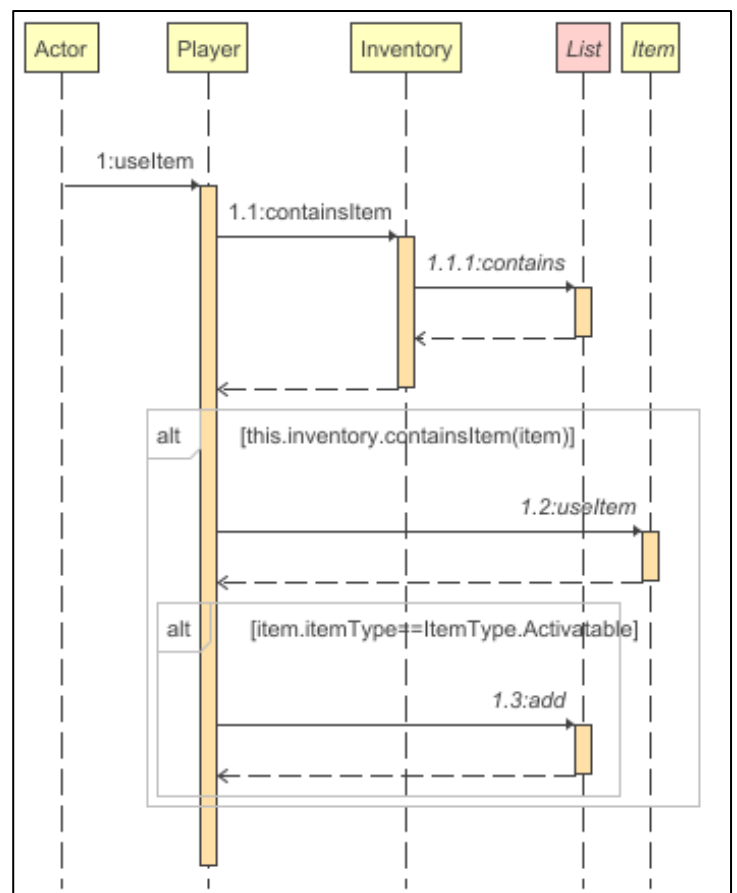
3.4.2 Character picks up item



## 3.4.3 Character drops item

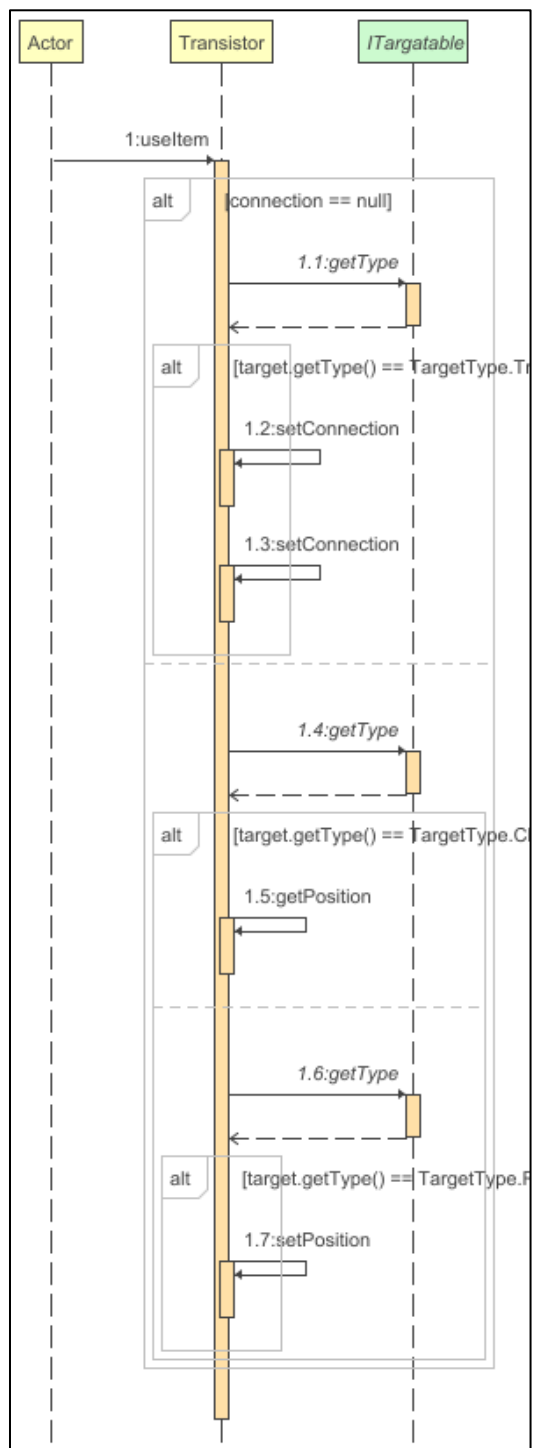


## 3.4.4 Player uses item

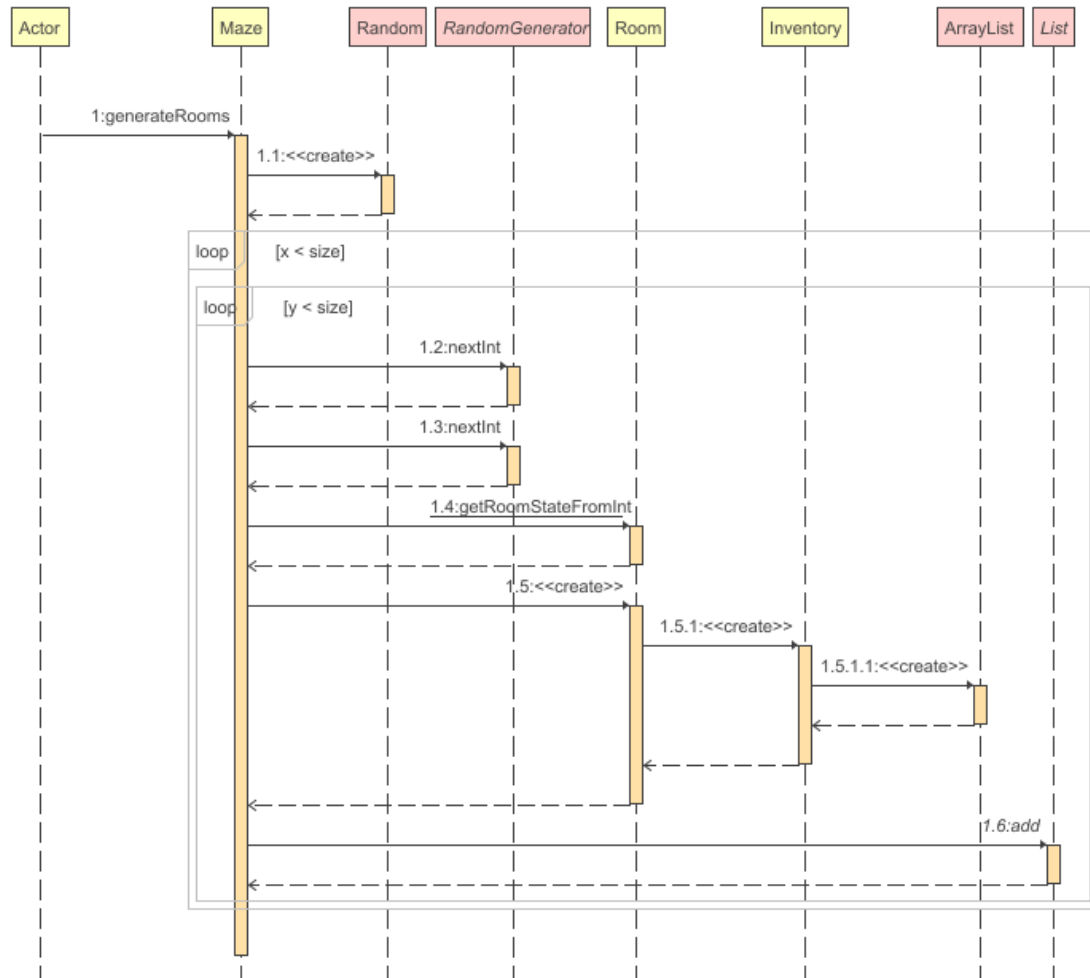




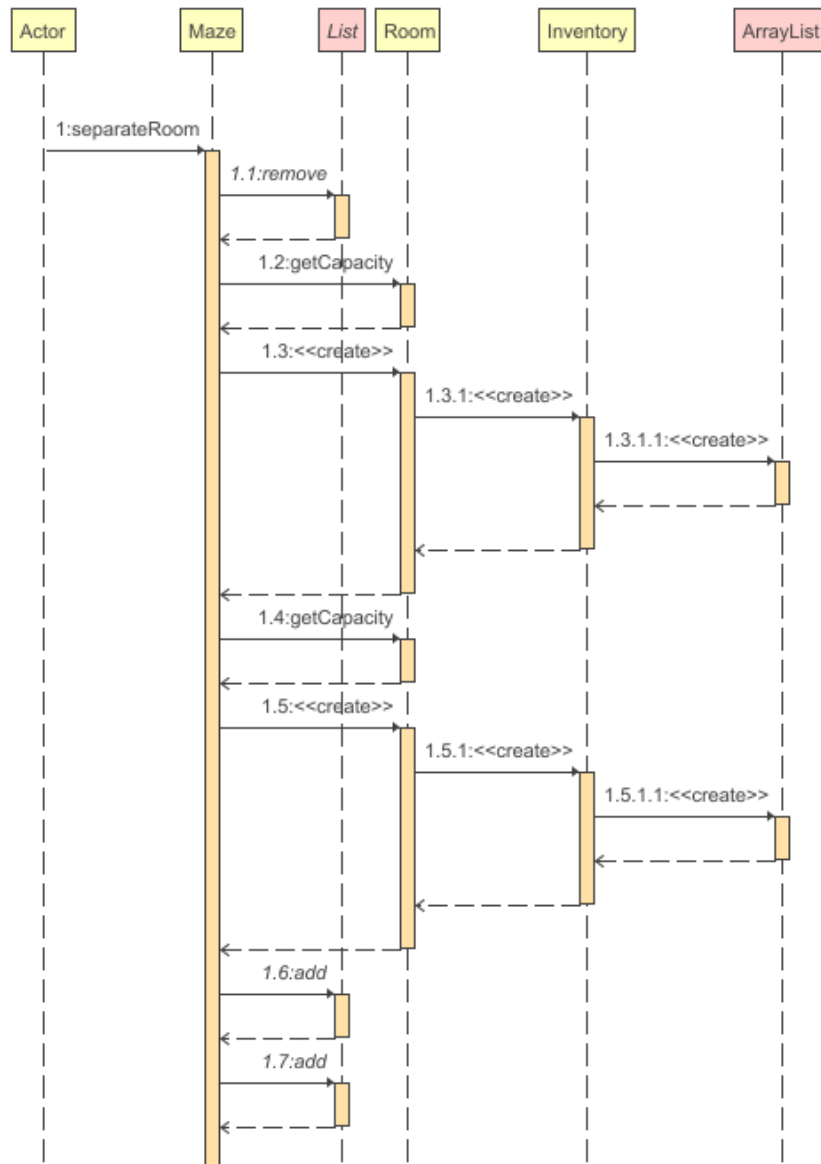
## 3.4.5 Transistor uses item



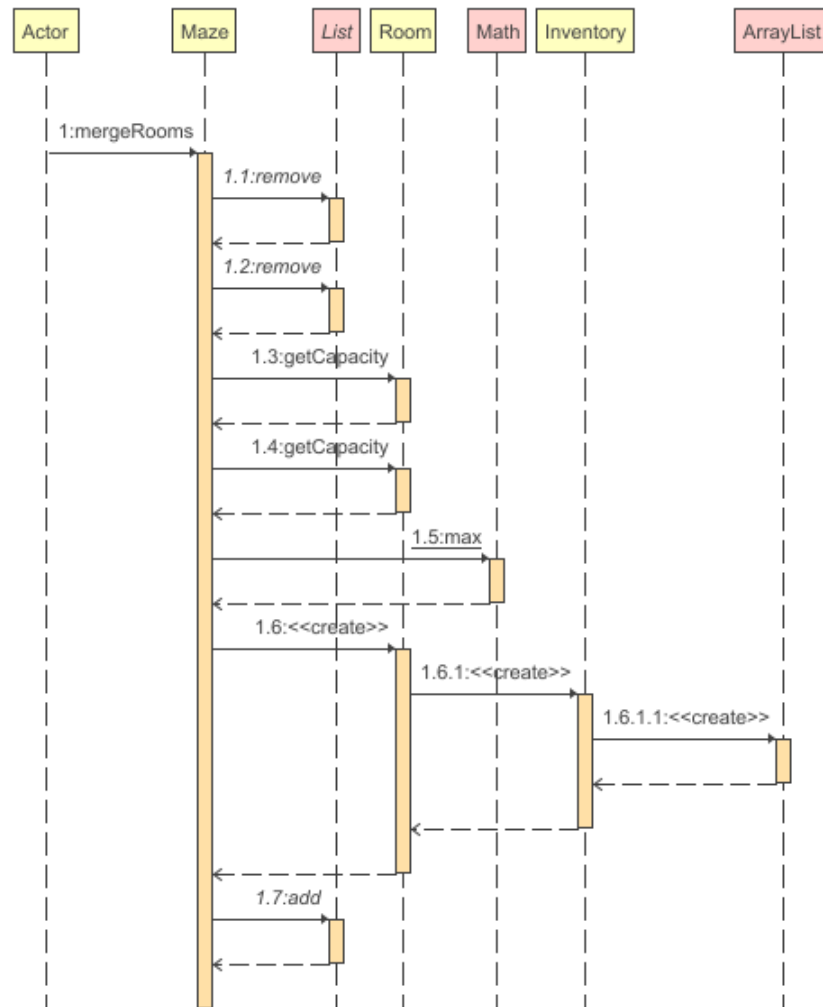
## 3.4.6 Maze generates rooms



## 3.4.7 Maze separates room



## 3.4.8 Maze merges rooms

**State-chartok**

-

**Napló**

| Kezdet                 | Időtartam     | Résztevők  | Leírás   |
|------------------------|---------------|--|--|
| 2024.02.29.<br>18:00   | 1 óra 15 perc | Draskóczy,<br>Halász,<br>Molnár,<br>Pigler,<br>Serfőző | Értekezlet.<br>Döntés:<br>Pigler elkészíti az Objektum katalógust és egy kezdetleges vázat az osztálydiagramhoz,<br>Draskóczy, Molár és Serfőző összeülnek megtervezni a statikus struktúradiagramokat,<br>Serfőző készíti majd a szekvencia diagramot.  |
| 2024.02.29.<br>20:00   | 2 óra         | Pigler   | Pigler elkészíti az Objektum katalógust és az osztálydiagram kiindulásául szolgáló vázát.  |
| 2024.03.01.<br>14:30   | 3,5 óra       | Draskóczy,<br>Molnár,<br>Serfőző                       | Értekezlet.<br>Tevékenység:<br>Az objektumkatalógus és a korábban elkészített use-case-ek alapján közösen gondolkodva elkészítik az osztálydiagramot és egyeztetik, hogy melyik osztály milyen adattagokkal, illetve függvényekkel rendelkezzen.<br>Döntés:<br>Serfőző befejezi az osztálydiagramot és belekezd a szekvenciadiagramokba. A fennmaradó kérdéseket a csapat többi tagjával átbeszéljük. Molnár kiegészíti az objektum katalógust, Draskóczy megírja az osztályok leírását. |
| 2024.03.01.<br>19:00   | 2 óra         | Molnár   | Molnár kiegészíti az objektum katalógust.  |
| 2024.03.01.<br>19:00   | 2 óra         | Serfőző  | Serfőző elkészíti a végleges osztálydiagramot.   |
| 2024.03.02.11:00       | 1 óra         | Draskóczy,<br>Halász,<br>Molnár,<br>Pigler,<br>Serfőző | Értekezlet.<br>Döntés: A még kérdéses attribútumok, függvények, illetve osztályok átbeszélése,<br>Halász elkészíti a szekvencia diagramokat.   |
| 2024.03.02.20:00       | 5 óra         | Halász   | Halász elkészíti a szekvencia diagramokat és módosításokat végez az osztálydiagramon   |
| 2024. 03.03. 8:00      | 4 óra         | Draskóczy  | Draskóczy elkészíti az osztályok leírását.   |
| 2024.03.03.17:00       | 1 óra         | Halász   | Halász kiegészíti az osztályok leírását.   |
| 2024. 03. 03.<br>17:00 | 1 óra         | Serfőző  | Serfőző letisztázza a szekvencia diagramokat.  |
| 2024. 03. 03.<br>18:00 | 1 óra         | Pigler   | Pigler átnézi az osztályok leírását.   |

|                        |       |           |   |
|------------------------|-------|-----------|---|
| 2024. 03. 03.<br>19:00 | 1 óra | Draskóczy | Draskóczy egységesíti a dokumentumot és megírja a naplót. |
|------------------------|-------|-----------|---|

