

8. Részletes tervek

8.0 Változtatások

Szobák szomszédságának leírása:

- Minden szoba neve után felsoroljuk a szomszédos szobák neveit, amelyekkel a szoba közvetlenül kapcsolatban van.
- A szomszédos szobák neveit szóközzel választjuk el.
- Vessző után kiírjuk a szoba aktuális állapotát / állapotait (ragacsos, átkozott, mérgező, tiszta)
- Még egy vessző után pedig a szoba férőhelye
- A sor végén Enter van.

Példa:

- Szoba1: Szoba2 Szoba3, tiszta, 5

Itt a Szoba1-ből közvetlen út vezet a Szoba2 és Szoba3-ba.

Emberek leírása:

- Minden szoba neve után felsoroljuk az ott található oktatók és takarítók neveit.
- A neveiket szóközzel választjuk el.
- Ha egy szobában nincsenek oktatók vagy takarítók, akkor az "Emberek:" utáni sor üres marad.

Példa:

- Szoba1: Szoba2 Szoba3, tiszta, 5
- Oktatók: Oktato1 Oktato2

Itt a szobák kapcsolatán felül a Szoba1-ben tartózkodik az Oktato1 és az Oktató2.

Tárgyak leírása:

- Minden szoba neve után felsoroljuk az ott található tárgyak neveit.
- A tárgyak neveit szóközzel választjuk el.
- Ha egy szobában nincsenek tárgyak, akkor az "Tárgyak:" utáni sor üres marad.

Példa:

- Szoba1: Szoba2 Szoba3, tiszta, 5
- Oktatók: Oktato1 Oktato2
- Tárgyak: Rongy Pohar

Itt a szobák kapcsolatán és a szobába tartózkodó Oktatókon felül látható, hogy a Szoba1-ben van egy rongy és egy Pohár.

Egy Labirintus teljes mentési példája:

Szoba1: Szoba2 Szoba3, tiszta, 5 Ember: Oktato1 Oktato2 Tárgyak: Rongy Pohar Szoba2: Szoba1 Szoba3, tiszta, 3 Ember: Oktato3 Tárgyak: TVSZ Szoba3: Szoba1 Szoba2, tiszta, 4 Ember: Tárgyak:	
---	--

Ahogy látható az egyes szobákat külön sor választja el egymástól.

8.0.1 Bemeneti nyelv**play <hallgatók db> <körök száma> <Szoba>**

Leírás: Játék indítása, pálya inicializálása, az alapján, hogy mennyi hallgató fog játszani.

Opciók: Hallgatók (játékosok), és körök számának, illetve a kezdő szoba megadása.

save <file név>

Leírás: A játék ezzel a paranccsal menti el a labirintus állását egy file-ba, ahonnan majd később újra megnyitható lesz.

Opciók: Annak a file-nak a neve, ahova menteni szeretne a játékos.

ajtohasznalat <Ajto> <Ember>

Leírás: A Ember ezzel a paranccsal lesz lehetősége átmenni egyik szobából a másikba, azzal, hogy megadja azt az ajtót, amit használni szeretne.

Opciók: Ajto, a Hallgató szobájából nyíló egyik ajtó azonosítója, amit használni szeretne az Ember. Ember, aki használni szeretné.

targyfelvesz <Targy> <Ember>

Leírás: Az Ember ezzel a paranccsal lesz lehetősége felvenni egy a szobában lévő tárgyat.

Opciók: Tárgy, a kívánt tárgy, amit a Hallgató fel szeretne venni. Ember, aki fel akarja venni a tárgyat. Ember, aki fel szeretné venni.

hasznal <Targy> <Hallgató>

Leírás: A kiválasztott tárgyat használja a Hallgató, viszont ez csak Camambert, Pohár és Tranzistor tárgyakra van értelmezve, a többire több mint felesleges nyomkodni.

Opciók: Tárgy, a kívánt tárgy, amit a Hallgató használni szeretne. Hallgató, aki használni szeretné.

targyeldob <Targy><Ember>

Leírás: Az Ember ezzel a paranccsal lesz lehetősége eldobni egy a hátizsákjában lévő tárgyat.

Opciók: Tárgy, a kívánt tárgy, amit az Ember el szeretne dobni. Ember, aki el akarja dobni

infoember <Ember>

Leírás: A Játékosnak ezzel a paranccsal lesz lehetősége az Ember ál lekérdezni.

Opciók: Ember, akinek a tárgyait szeretnénk megtudni.

infoszoba <Szoba>

Leírás: A Játékosnak ezzel a paranccsal lesz lehetősége a szoba adatait lekérdezni.

Opciók: Szoba, aminek az adatait szeretnénk megtudni.

szobafeloszt <Szoba>

Leírás: A Játékosnak ezzel a paranccsal lesz lehetősége egy szobát kettéosztani.

Opciók: Szoba, amit ketté szeretnénk osztani.

szobaösszevon <Szoba> <Szoba>

Leírás: A Játékosnak ezzel a paranccsal lesz lehetősége két szobát összevonni.

Opciók: A két szoba, amiket össze szeretnénk vonni.

load <palyanev>

Leírás: Pálya betöltése a megadott nevű file-ból, Szobákkal, Oktatókkal és tárgyakkal együtt.

Opciók: palyanev, Melyik file-t olvassa be a pálya inicializálásakor.

random <bool>

Leírás: Beállítja a tesztelés módját, hogy véletlenszerűséget vagy determinisztikus lefutást használjon.

Opciók: bool: Engedélyezi vagy letiltja a véletlenszerűséget. Ha "true", akkor a program véletlenszerű funkcióit használja, ha "false", akkor determinisztikus módban fut.

reset

Leírás: Eredeti állapotba hozza a programot.

Opciók: -

tick <léptetes>

Leírás: Idő léptetése egy egységgel.

Opciók: léptetés, Léptetések száma amivel múlni fog az "idő".

8.0.2 Kimeneti nyelv

play:

a palya legeneralva <hallgatok db> fore.

kezdő szoba: <Szoba>

a játék elindult, <körök száma> korig tart.

save:

a palya a <file név> fájlba lett elmentve.

szobák:

<Szoba1>

szomszédok: <Szomszédos szobák listája vesszővel elválasztva>

tárgyak: <Szoba1-ben levő tárgyak vesszővel elválasztva>

emberek: <Szoba1-ben levő oktatók vesszővel elválasztva>

állapot: <Szoba1 állapota>

.

.

.

<SzobaN>

szomszédok: <Szomszédos szobák listája vesszővel elválasztva>

tárgyak: <SzobaN-ben levő tárgyak vesszővel elválasztva>

emberek: <SzobaN-ben levő tárgyak vesszővel elválasztva>

állapot: <SzobaN állapota>

ajtóhasznalat:

Aleset1:

<Ember>: atleptel az <Ajtó> ajton.

Aleset2:

<Ember>: az <Ajtó> ajtó egyirányú és nem tudsz rajta átmenni.

Aleset3:

<Ember>: a szomszéd szoba megtelt. nem fersz be.

tárgyfelvesz:

Aleset1:

<Ember>: az inventoryba tetted a <Tárgy> tárgyat.

Aleset2:

<Ember>: az inventoryd teli van, nem fér bele a <Tárgy> tárgy.

Aleset3:

<Ember>: a szoba ragacsos a <Tárgy> tárgy nem vehető fel.

hasznal:

Aleset1:

<Hallgató>: a <Tárgy> tranzisztort parosítja a másik taskában lévő tranzisztorral.

Aleset2:

<Hallgató>: a <Tárgy> tranzisztor parjához teleportalunk.

Aleset3:

<Hallgató>: a <Tárgy> pohár megved a támadások előtt, és eldobod a <Tárgy> tárgyat.

Aleset4:

<Hallgató>: a <Camambert> elgázítja a <Szoba> szobát.

targyeldob:

Aleset1:

<Ember>: az inventorydbol a foldre dobtad a <Tárgy> tárgyat.

Aleset2:

<Ember>: az inventoryd ures, nem dobhatsz el tárgyat.

infoember:

Aleset1:

<Ember> tárgyai: < Tárgyak listája, vesszővel elválasztva>

itt van: <Jelenlegi szoba>

Aleset1.1:

ajult.

Aleset1.2:

eber.

Aleset2:

nincs ilyen ember <Ember>

infoszoba:

Aleset1:

a szoba allapota: <Szoba állapot>

a szobaban levo tárgyak: < Tárgyak listája vesszővel elválasztva>

a szobában levo oktatók: <Oktatók listája vesszővel elválasztva>

a szobabol nyilo ajtok: <Ajtó listája vesszővel elválasztva>

Aleset2:

nincs ilyen szoba <Szoba>

szobafeloszt:

uj szoba jott létre: <Új szoba>

szobaosszevon:

<Szoba> es <Szoba> ossze lett vonva.

Load:

a <Pályanév> jatek betoltve.

szobak:

<Szoba1>

szomszedok: <Szomszédos szobák listája vesszővel elválasztva>

targyak: <Szoba1-ben levő tárgyak vesszővel elválasztva>

emberek: <Szoba1-ben levő oktatók és vagy takarítók vesszővel elválasztva>

allapot: <Szoba állapot>

<Üres sor>

.

.

.

<SzobaN>

szomszedok: <Szomszédos szobák listája vesszővel elválasztva>

targyak: <SzobaN-ben levő tárgyak vesszővel elválasztva>

emberek: <SzobaN-ben levő oktatók és vagy takarítók vesszővel elválasztva >

allapot: <SzobaN állapot>

random:

Aleset1:

mostantól determinisztikus a játék.

Aleset2:

mostantól véletlenszerű a játék.

reset:

a játék visszaállt a kiinduló állapotba.

tick:

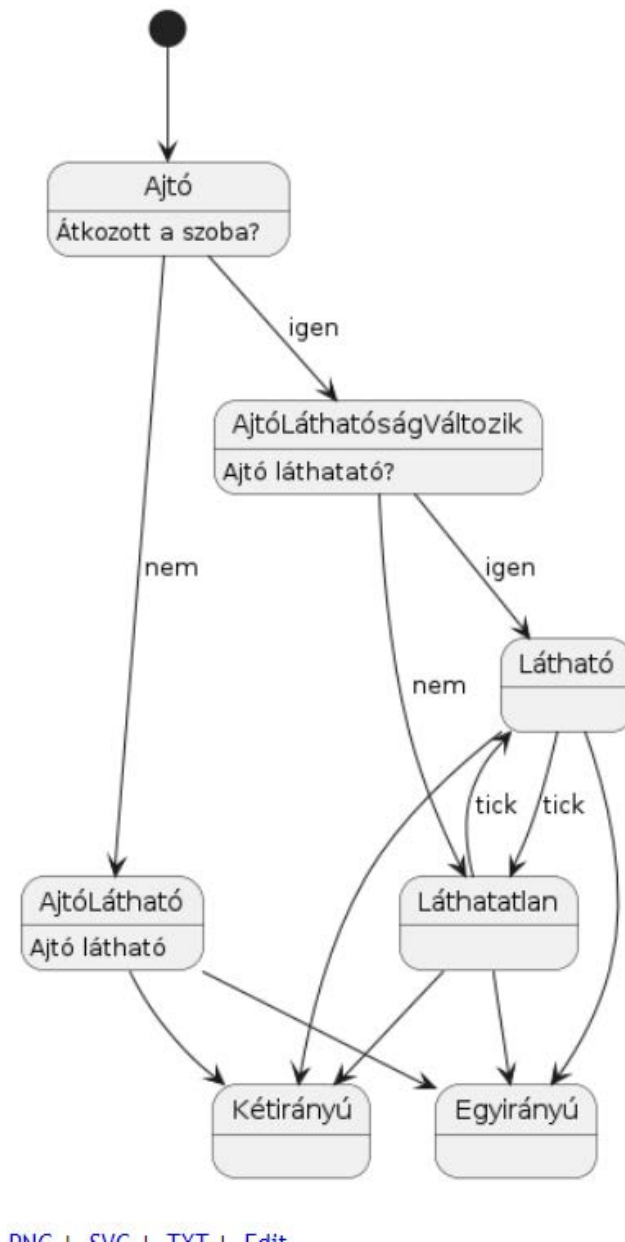
Aleset1:

a játéknak vége, az idő lejárt.

8.1 Osztályok és metódusok tervei.**8.1.1 Ajtó**

- **Felelősség**

Az Ajtó osztály felelős az egyes szobákat összekötő ajtók kezeléséért és működtetéséért a labirintusban. Feladata az ajtók állapotának nyomon követése, valamint annak ellenőrzése, hogy nyitható-e a kért irányba az ajtó. Az Ajtó osztály biztosítja a játékosok számára az átjutást a különböző szobák között azáltal, hogy lehetővé teszi az áthaladást az ajtókon keresztül.



- **Őszosztályok**

Nincs

- **Interfészek**

Nem valósít meg interface-t.

- **Attribútumok**

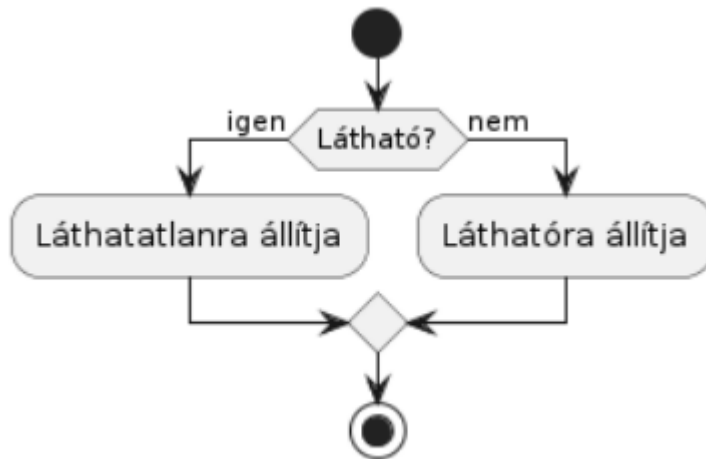
- **egyikoldal:** Tárolja az ajtó egyik oldalán lévő szobát, -, Szoba
- **masikoldal:** Tárolja az ajtó másik oldalán lévő szobát, -, Szoba
- **lathato:** Tárolja hogy jelenleg látható-e az ajtó, -, boolean
- **merreNyilik:** Tárolja hogy melyik irányba nyílik az ajtó, -, Map<Szoba, Boolean>
- **nev:** Az objektum nevét (pl. a1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String

Metódusok

- **Ajto(Szoba egyikOldal, Szoba masikOldal, String nev):** Létrehoz egy kétirányú ajtót a megadott két szoba között, a megadott névvel, +
- **String toString():** Visszaad egy stringet a következő formában: “név” :Ajto. (pl. e :Ajto), +
- **Void setMerreNyilik(boolean b1, boolean b2):** A metódus beállítja a megadott adatok alapján, hogy melyik irányba nyílik az ajtó, +
- **Void hasznal(Ember e):** A metódus először ellenőrzi hogy látható-e az ajtó és nyílik-e a szomszéd szobába, ha nem akkor visszatér bármilyen történés nélkül. Másik esetben, az embert eltávolítja a jelenlegi szobából és átlépteti a szomszédosba, +



- **Boolean getLathatosag():** A metódus visszaadja az ajtó láthatóságát, +
- **Void setLathatosag(boolean a):** A metódus beállítja az ajtó láthatóságát a megadott állapotra, +
- **Void lathatosagValtoztass():** A metódus az ajtó láthatóságát az ellenkezőjére változtatja, +



- **Szoba getSzomszed(Szoba sz):** A metódus visszaadja a megadott szoba azon szomszédját amely ezzel az ajtóval van összekötve, +

8.1.2 Ember

- **Felelősség**

Az ember absztrakt osztály reprezentálja a játékos, illetve nem játékos karaktereket

- **Ősosztályok**

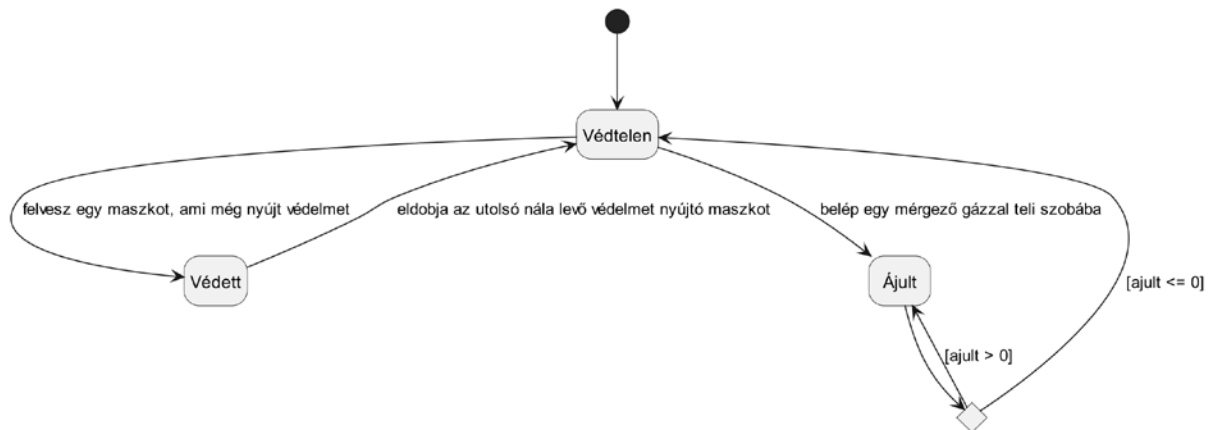
Nincs

- **Interfészek**

TárgyVisitor, Időzített

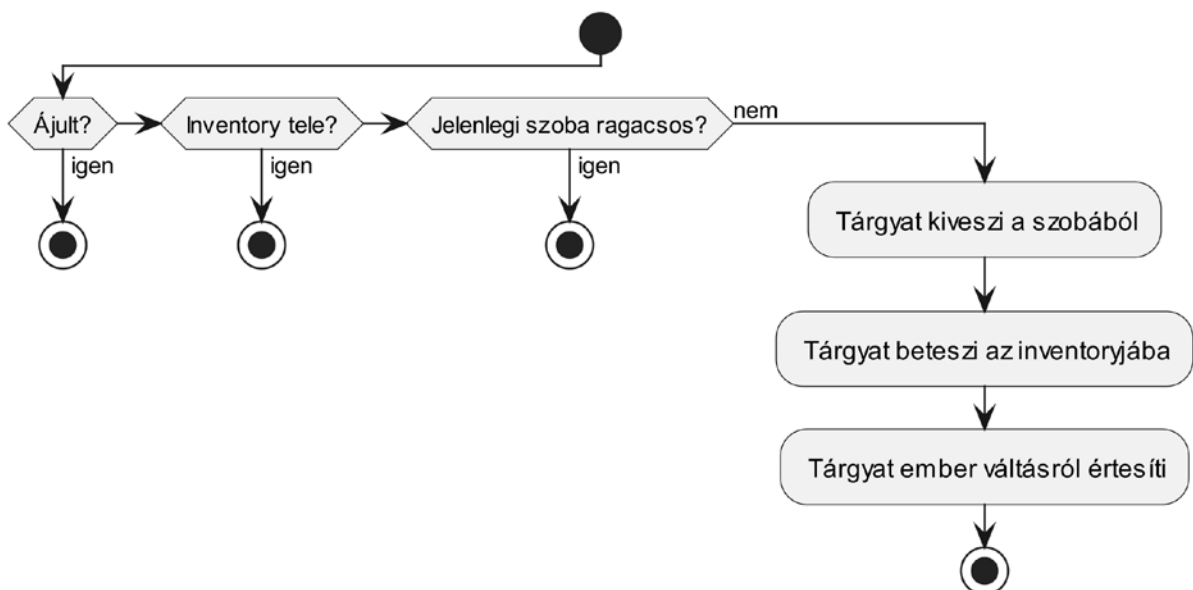
- **Attribútumok**

- **inventory:** a karakternél található tárgyakat tárolja, #, List<Targy>
- **jelenlegiSzoba:** a szobát tárolja, amiben a karakter éppen tartózkodik, #, Szoba
- **nev:** az objektum nevét (pl. e1) tárolja, csak a prototípusnál és a szkeletonnál használatos, #, String
- **gazEllenVedett:** tárolja, hogy a karakter a jelenlegi körben védett-e a mérgező gázzal szemben (vagyis van-e nála Maszk), -, boolean
- **ajult:** tárolja, hogy a karakter hány körig cselekvésképtelen még -, int

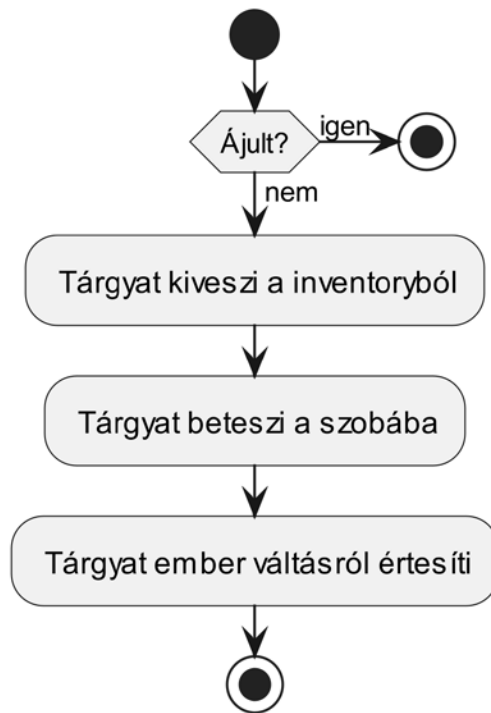


• Metódusok

- **Ember(String nev):** Létrehoz egy új Ember-t a paraméterként megadott névvel., +
- **String toString():** Visszaad egy stringet a következő formában: "név" :Ember. (pl. e :Ember), +
- **void tárgyatFelvesz(Targy t):** A metódus először ellenőrzi, hogy az Ember ájult-e, inventoryja tele van-e és a jelenlegi Szobája ragacsos-e. Ha ezek közül bármelyik feltétel teljesül, akkor visszatér anélkül, hogy bármi is történne. Egyébként kiveszi a jelenlegi szobából a t tárgyat azt hozzáadja az Ember inventoryjához, és szól a tárgynak, hogy új emberhez került., +

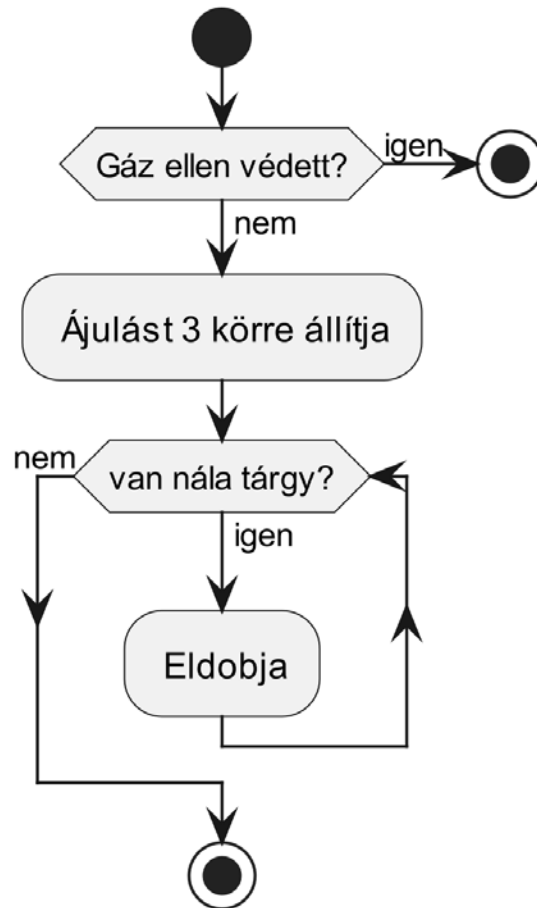


- **void tárgyatEldob(Targy t):** a metódus először ellenőrzi, hogy az Ember ájult-e, ha nem, akkor kiveszi a t tárgyat az Ember inventoryjából, azt lerakja az Ember jelenlegi szobájába és szól a tárgynak, hogy éppen nincs gazdája., +

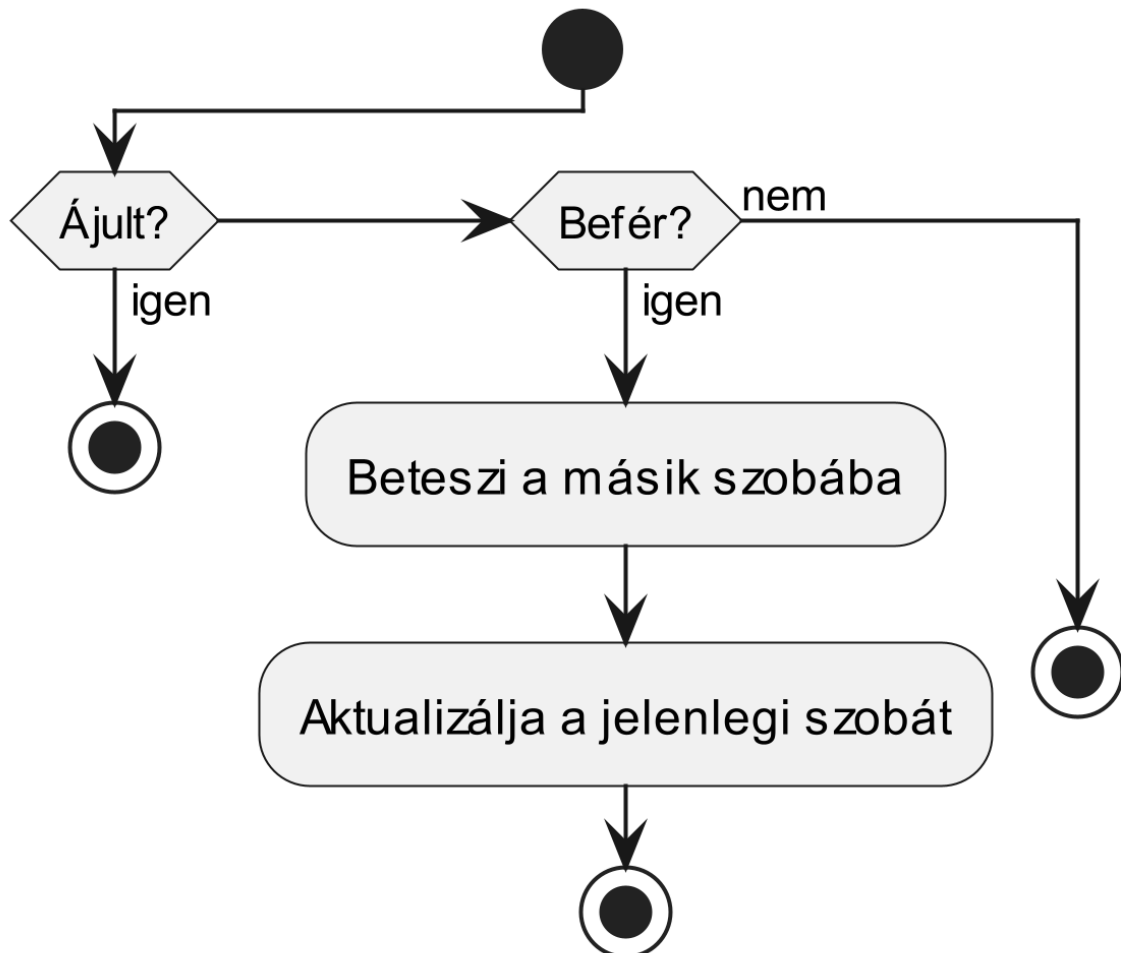


- **void visit(Maszk m):** a metódus leellenőrzi, hogy az m maszknak lejárt-e a védideje. Ha nem akkor az Emberen beállítja a mérgező gáz elleni védettséget igazra, különben hamisra., +
- **void visit(Legfrissito l):** a metódus hatására az Ember eldobja a nála levő légfrissítőt, +

- **void ajulas():** a metódus először ellenőrzi, hogy védett-e az Ember a mérgező gázzal szemben, ha igen akkor azonnal visszatér. Különben 3 körre állítja be az ájulást és eldobja az Ember összes tárgyát., +

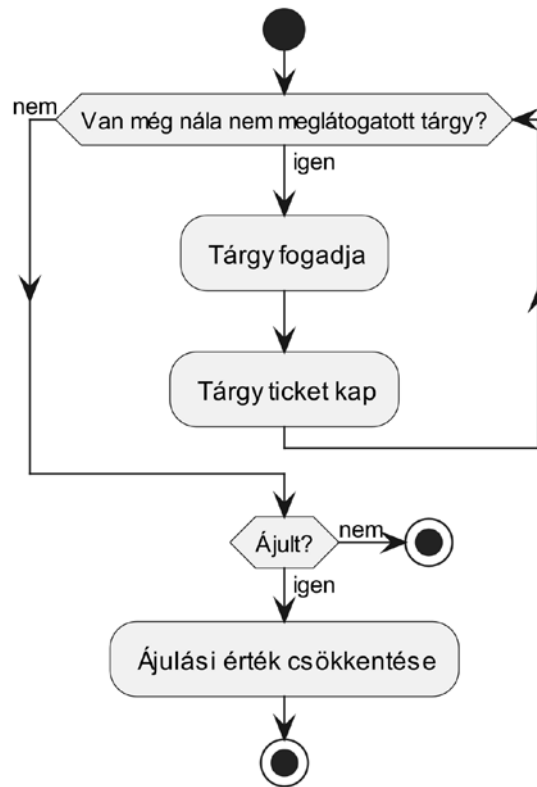


- **void masikSzobabaLep(Szoba sz):** a metódus ellenőrzi, hogy az Ember ájult-e és befér-e a paraméterként kapott szobába. Ha ájult, vagy nem fér be, akkor visszatér minden további nélkül. Egyébként az Embert beteszi a paraméterként kapott sz szobába, a jelenlegi szobáját is átállítja erre és értesíti a tárgyakat, hogy új szobába kerültek., +



- **void kilepSzobajabol():** a metódus kiveteti az Embert a jelenlegi szobájából., +
- **void targyatHasznal(Tárgy t):** a metódus hatására az Ember a paraméterként t tárgyat használja., +
- **Szoba getJelenlegiSzoba():** A metódus visszaadja az Ember jelenlegi szobáját.
- **List<Targy> getItems():** A metódus visszaadja az Ember inventoryjában tárolt tárgyak listáját., +
- **boolean getAjult():** A metódus visszaadja, hogy az Ember éppen ájult-e., +
- **abstract boolean inventoryTeleE():** A metódus itt nem, hanem minden leszármazottban kerül kötelezően implementációra, mivel nem állapítható meg általánosan, hogy egy Embernek éppen tele van-e az inventoryja, mert az Oktató például csak 1 tárgyat hordozhat magánál, míg a Hallgató 5-öt.
- **void rongyotElszenved:** A metódus nem csinál semmit, mert nem általánosan megállapítható, hogy mi történjen a rongyot elszenvedő emberrel. (Egyébként az Oktató osztály definiálja felül, ott értelmezhető a rongy elszenvedése)., +

- **void tick():** A metódus hatására minden az Ember inventoryjában található tárgy fogadni fogja az Embert, majd tickelésre kerülnek ezek is. Ezen kívül ha az Ember éppen ájult, akkor eggyel csökkenti az ájulása értékét., +



8.1.2 Labirintus

- **Felelősség**

A Labirintus osztály felelős a játéktérület létrehozásáért és kezeléséért. Ez az osztály tartalmazza a labirintusban található szobákat és irányítja azok osztódását és egyesülését. A Labirintus osztály feladata a szobák hálózatának kezelése.

- **Ősosztályok**

Nincs

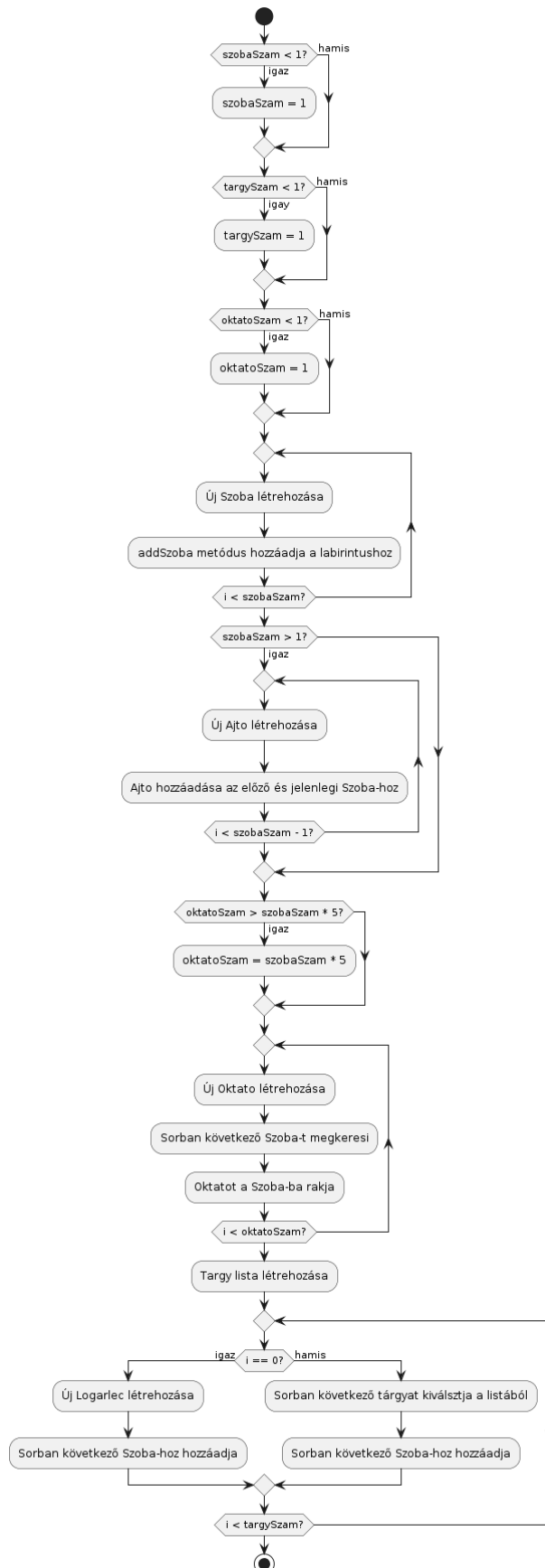
- **Interfészek**

Időzített

- **Attribútumok**

- **Timeleft:** A játékból hátralevő időt tartja számon, -, int
- **Nev:** az objektum nevét (pl. 11) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String
- **Szobak:** A labirintust alkotó szobák listáját tárolja, -, List<Szoba>

- **Metódusok**
- **Labirintus(String l):** Létrehoz egy új Labirintus-t a paraméterként megadott névvel., +
- **String toString():** Visszaad egy stringet a következő formában: “név” :Labirintus. (pl. l :Labirintus), +
- **void addSzoba(Szoba sz):** A metódus a labirintust alkotó szobákhoz hozzáadja a paraméterként kapott szobát., +
- **void setTimeLeft(int t):** A metódus beállítja a játékból hátralévő időt a megadott paraméter alapján, +
- **void szobaFeloszt(Szoba szoba):** A metódus, a megadott szobát kettéosztja. Az új szoba az eredetiben tartózkodó embereket leszámítva megegyezik az eredetivel, +
- **void szobakOsszevon(Szoba sz1, Szoba sz2):** Meghívja a szoba egyesít metódusát ami összevonja a két szobát, ajtajaikat és a benne lévő tárgyakat, +
- **void jatekNyert():** A metódus kiírja ha a játéknak vége és a hallgatók nyertek, +
- **void jatekVeszt():** A metódus kiírja ha a játéknak vége és a hallgatók vesztek, +
- **List<Szoba> getSobak():** A metódus visszatér a labirintust alkotó szobák listájával, +
- **void init(int szobaSzam, int oktatoSzam, int targySzam):** A metódus, a megadott adatok alapján a létrehoz egy labirintust és inicializálja a játékot, +



- **void tick():** A metódus, a játékból hátralevő időt csökkenti, +
- **void szobaKivesz(Szoba sz):** A metódus, a paraméterként megadott szobát eltávolítja a labirintusból, +

8.1.3 Szoba

- **Felelősség**

A labirintust szobák alkotják. Egy szoba tárolja a benne tartózkodó Embereket. Felelőssége számontartani, ha adott Hallgatók immunisak bizonyos Oktató ellen, mert védelmi tárgyat használtak. Továbbá tárolja az ajtóit, amik által a többi szobához kapcsolódik. Állapota lehet átkozott, ekkor ajtóit eltűnnek. Lehet ragacsos, ekkor a bent található és bent letett tárgyakat nem lehet felvenni. Lehet gázos, ekkor mérgezett bent a levegő és védelem szükséges az ájulás elkerüléséhez.

- **Össztályok**

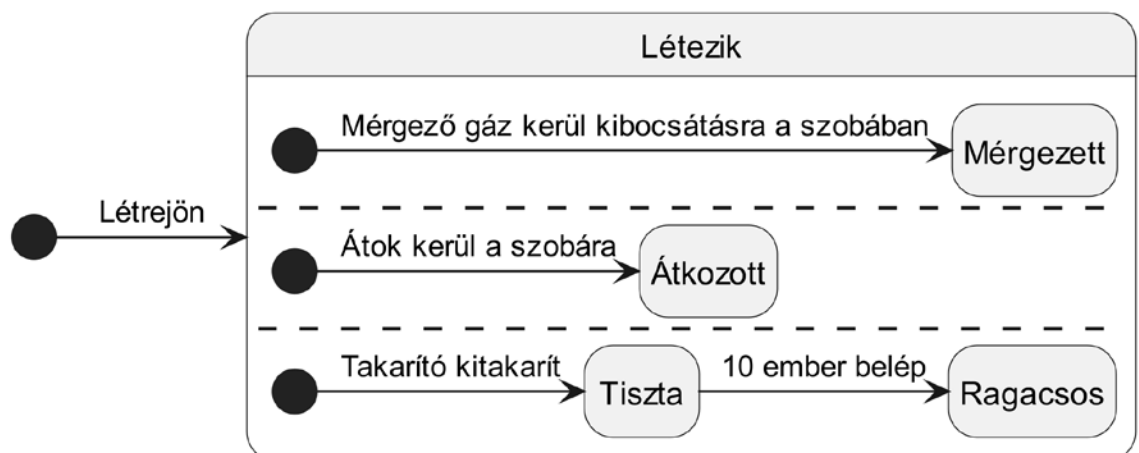
Nem leszármazott osztály.

- **Interfészek**

Az Idozitett interfacet valósítja meg.

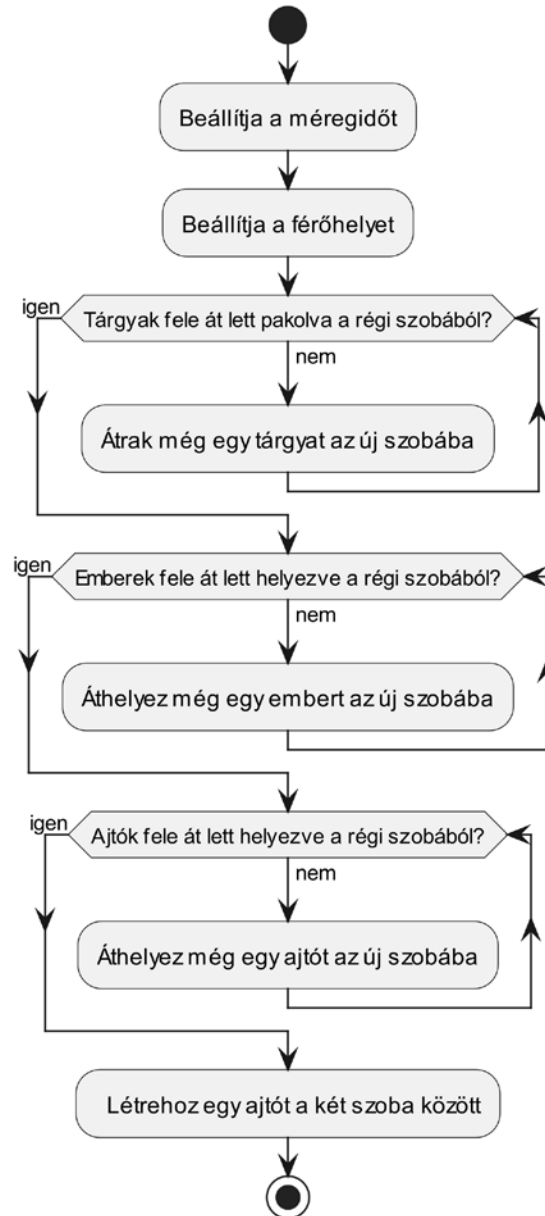
- **Attribútumok**

- **átkozott**, Tárolja, hogy a szoba átkozott állapotban van-e, -, boolean
- **immunisok**, Tárolja az adott Oktató mely Hallgatókat nem támadhatja, mert azok immunitást szereztek ellene., -, Map<Oktato, List<Hallgato>>
- **meregIdo**, Meghatározza még mennyi ideig mérgezett az adott szoba, ha nem elmúló mérge, akkor értéke -1., - int
- **nev**, Szoba neve, leginkább teszteléshez és bemeneti, kimeneti nyelvhez., -, String
- **ragacsos**, Tárolja, hogy a szoba jelenleg ragacsos állapotban van-e., -, boolean
- **ragacsIdozito**, Meghatározza, mennyi ideig ragacsos még az adott szoba., -, int
- **ferohely**, Meghatározza, hány Ember tartózkodhat egyidejűleg a Szobában., -, int
- **bentlevok**, Tárolja a Szobában bent tartózkodó embereket., -, List<Ember>
- **ajtok**, Tárolja a Szoba ajtajait., -, List<Ajto>



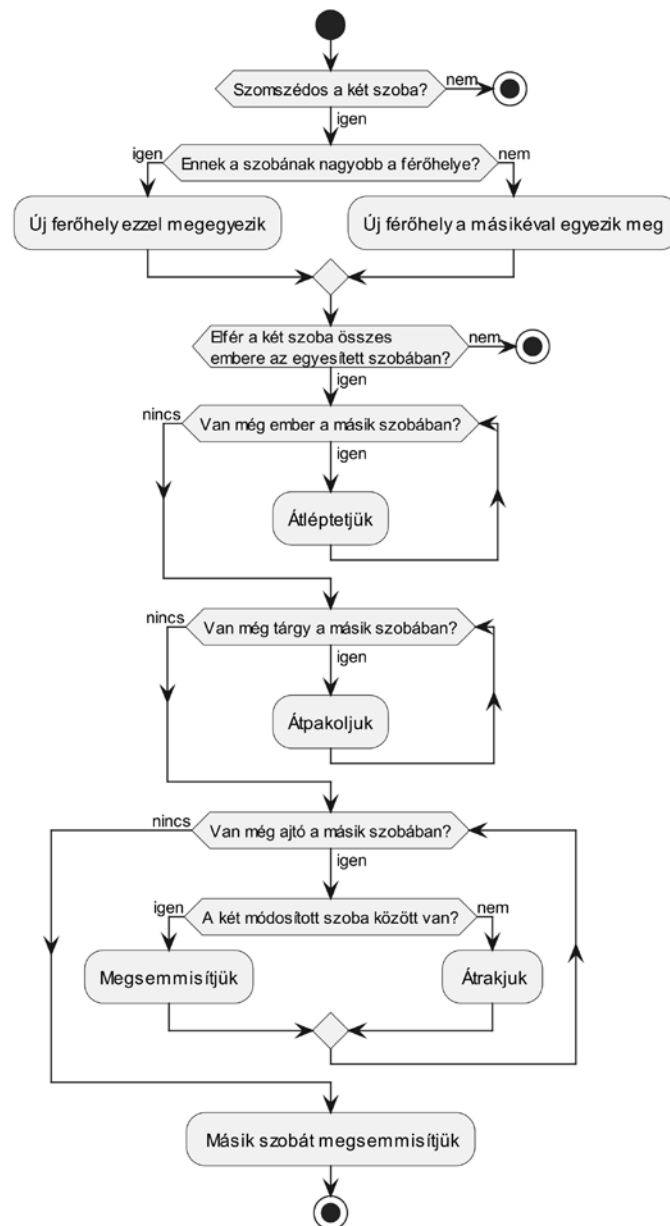
• Metódusok

- **Szoba(Szoba másikSzoba):** Létrehoz egy új szobát, úgy hogy a tulajdonságaik megegyeznek, de az emberek, tárgyak és ajtók megfelelődnek a kettő között. Emellett létrehoz egy új ajtót a két szoba közé., +



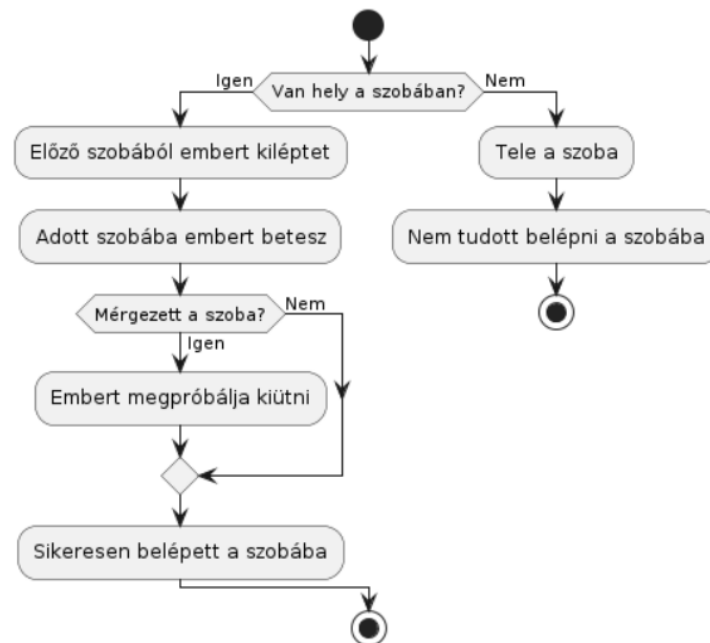
- **void tick():** Időzített interface metódusa, idő mérésére és történések validálására, láthatóság: +.

- **void egyesit(Szoba sz1):** Egyesíti this-t és a paraméterként kapott sz1 Szobát, áthelyezi a bent tartózkodó embereket és az ajtókat, a kettő között tartózkodó ajtót eltávolítva, láthatóság: +.



- **String toString():** A Szoba objektum stringgé alakítására használatos, hogy szép és olvasható legyen a kimeneti nyelvben és tesztekben, láthatóság: +.
- **boolean ragacsosE():** Visszaad egy igaz vagy hamis értéket annak függvényében, hogy az adott Szoba ragacsos vagy sem, láthatóság: +.
- **void removeItem(Targy t1):** Eltávolítja a Szobából a t1 Tárgyat, láthatóság: +.
- **List<Targy> getItems():** Visszaadja, hogy milyen Tárgyak találhatóak a Szobában, láthatóság: +.
- **void addItem(Targy t1):** Betesz a t1 Tárgyat a szobába, láthatóság: +.
- **int getFerohely():** A szoba férőhelye kérdezhető le vele, azaz a maximális kapacitása, egész szám, láthatóság: +.

- **void removeAjto(Ajto a1):** Eltávolítja az a1 ajtót a Szoba ajtóinak közül, láthatóság: +.
- **void addAjto(Ajto a1):** Beteszi az a1 ajtót a Szoba ajtóinak közé, láthatóság: +.
- **void setMeregIdo(int i):** Beállítja, hogy mennyi ideig legyen mérgezett az adott szoba az **int i** paraméter alapján, láthatóság: +.
- **List<Ajto> getAjtok():** Visszaadja az Ajtókat, amelyekkel a Szoba rendelkezik, láthatóság: +.
- **boolean mergezoE():** Visszaad egy igaz vagy hamis értéket annak függvényében, hogy az adott Szoba mérgező vagy sem, láthatóság: +.
- **boolean emberBetes(Ember e1):** Beteszi e1 Embert a szobába, ha lehetséges, láthatóság: +.



- **void emberKivesz(Ember e1):** Eltávolítja e1 Embert a szobából, láthatóság: +.
- **List<Ember> getEmberek():** Visszaadja a Szobában tartózkodó Emberek listáját, láthatóság: +.
- **void immunitastAd(Oktato o1, Hallgato h1):** Amennyiben a támadás ellen immunitást adó védekezés történt, feljegyzi a h1 Hallgató o1 Oktatóval szembeni védettségét, láthatóság: +.
- **void setTiszta(boolean tiszta):** A Szoba tisztaságát a paraméterként kapott változóra állítja, láthatóság: +.

8.1.4 Tárgy

· Felelősség

A Tárgy osztály felelős a tárgyak használatát biztosító interface megvalósításáért.

- **Ősosztályok**

- **Interfészek**

Nincs

- **Attribútumok**

Nincs

- **Metódusok**

- **void hasznal():** ezt a metódust írják felül az interfacet megvalósító osztályok, alapértelmezetten nem csinál semmit, +
- **void SetPar(Tranzisztor):** ezt írja felül a tranzisztor osztály, többi tárgynál nem csinál semmit, +
- **void accept(TargyVisitor):**
- **void SzobaValtasrolErtesit(Szoba):** ez értesíti az adott tárgyat, ha másik szobába került át ha felülírja ezt a metódust, alapértelmezetten nem csinál semmit, +
- **void hasznal(Oktato):** ezt írják felül az olyan tárgyak amelyeket oktatók ellen használnak, alapértelmezetten nem csinál semmit, +
- **void EmberValtasrolErtesit(Ember):** ez értesíti az adott tárgyat, ha másik emberhez került át ha felülírja ezt a metódust, alapértelmezetten nem csinál semmit, +

8.1.5 Camembert

- **Felelősség**

A Camembert osztály felelős a mérgező gáz ideiglenes kibocsátásáért a labirintus, egy a játékos által kiválasztott szobájába.

- **Ősosztályok**

- **Interfészek**

Megvalósítja a Targy interfacet

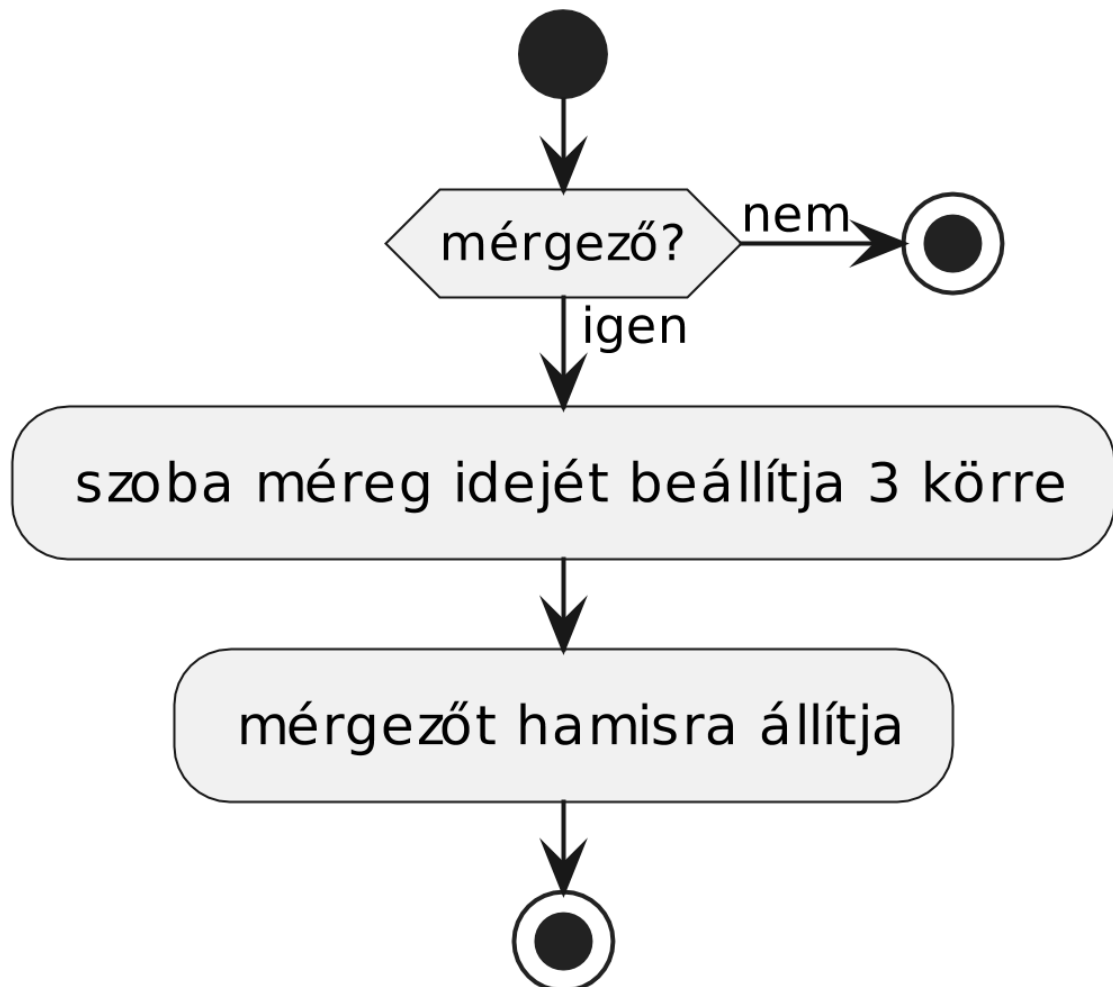
- **Attribútumok**

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String
- **szoba:** A szoba amelyben a tárgy tartózkodik, -, Szoba
- **mergezo:** tárolja mérgező gázt bocsát e még az objektum, -, boolean

- **Metódusok**

- **String toString():** Visszaad egy stringet a következő formában: “név” :Camembert., +

- **void hasznal():** ha mérgező meghívja a szobán amiben van a SetMeregido(int) függvényt egy adott értékkel pl. 3-mal és a mérgező értéket hamisra állítja, ha nem mérgező már akkor nem csinál semmit, +



- **void szobaValtasrolErtesit(Szoba):** átállítja a szoba értékét a megadott értékre, +

8.1.6 HamisLec

· Felelősség

Az osztály felelős a hamis logarléc tárgyak kezeléséért, amelyek a játékosok átverését szolgálják, nem történik semmi ha felveszi egy Ember

• Ősosztályok

Nincs

· Interfészek

Megvalósítja a Tárgy interfacet

- **Attribútumok**

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String

- **Metódusok**

- **String toString():** Visszaad egy stringet a következő formában: “név”
:HamisLec., +

8.1.7 HamisMaszk

- **Felelősség**

Az osztály felelős a hamis maszk tárgyak kezeléséért, amelyek a játékosok átverését szolgálják, nem védi meg az Embereket a gázos szobákban, de azt hihetik hogy van náluk maszk így veszélybe sodorva őket

- **Ősosztályok**

Nincs

- **Interfészek**

Megvalósítja a Tárgy interfacet

- **Attribútumok**

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String

- **Metódusok**

- **String toString():** Visszaad egy stringet a következő formában: “név”
:HamisMaszk., +

8.1.8 HamisTVSZ

- **Felelősség**

Az osztály felelős a hamis maszk tárgyak kezeléséért, amelyek a játékosok átverését szolgálják, nem védi meg a Hallgatókat az Oktatók ellen

- **Ősosztályok**

Nincs

- **Interfészek**

Megvalósítja a Tárgy interfacet

- **Attribútumok**

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String

- **Metódusok**

- **String toString():** Visszaad egy stringet a következő formában: “név”
:HamisTVSZ., +

8.1.9 Legfrissito

- **Felelősség**

Az osztály felelős a légfrissito tárgy kezeléséről, amely megszünteti a Szobákban található mérges gázt

- **Ősosztályok**

Nincs

- **Interfészek**

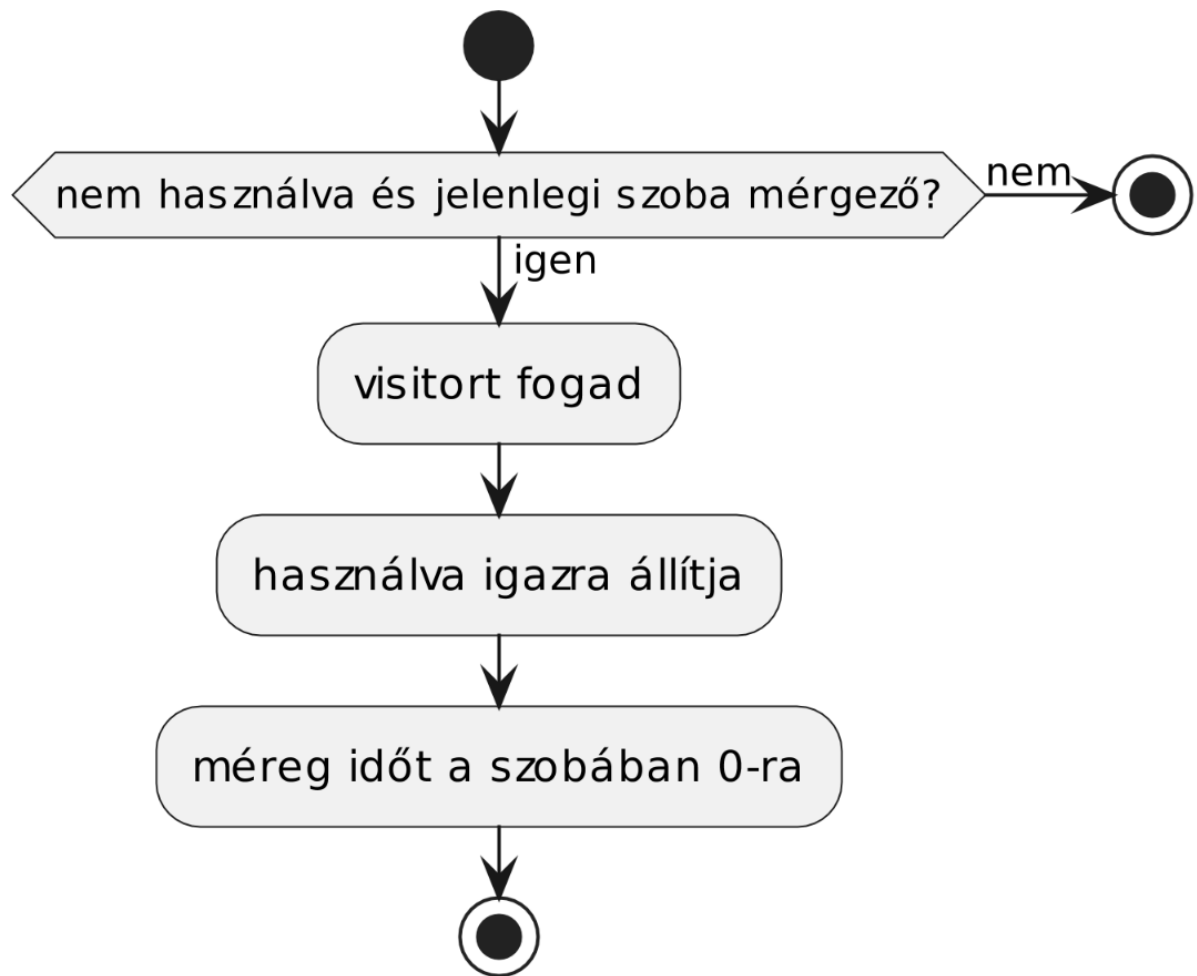
Megvalósítja a Tárgy interfacet

- **Attribútumok**

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String
- **hasznalva:** tárolja, hogy használták e már az adott objektumot, -, boolean
- **jelenlegiSzoba:** a szobát tárolja amelyben a tárgy jelenleg tartózkodik, -, Szoba

- **Metódusok**

- **String toString():** Visszaad egy stringet a következő formában: “név”
:Legfrissito., +
- **void SzobaValtasrolErtesit(Szoba):** átállítja a jelenlegiSzoba értékét a megadott értékre., +
- **void accept(TargyVisitor):** egy visitort fogad, ha a tárgy még nem volt használva, és a tárgy jelenlegi szobája mérgező, akkor a visitor meglátogatja a tárgyat, és beállítja a tárgy használva állapotát igazra és az aktuális szobában beállítja a méregidőt nullára. Ha a tárgy már használva van, vagy nem mérgező a szoba, akkor nem történik semmi., +



8.1.10 Logarlec

- **Felelősség**

A Logarléc osztály felelős a játék céljának reprezentálásáért. A játékosok feladata a Logarléc felkutatása és megszerzése a labirintusban annak érdekében, hogy nyerjenek

- **Ősosztályok.**

Nincs

- **Interfészek**

Megvalósítja a Tárgy interfacet

- **Attribútumok**

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szeptetonnál használatos, -, String

- **Metódusok**

- **String toString():** Visszaad egy stringet a következő formában: “név” :Logarléc., +
- **void accept(TargyVisitor):** fogadja a visitort, +

8.1.11 Maszk

- **Felelősség**

A Maszk osztály felelős a játékosok védelméért a mérgező gázokkal teli szobákban. A maszk egy adott ideig tudja csak megakadályozni a mérges gázok belélegzését.

- **Össztályok**

Nincs

- **Interfészek**

Megvalósítja a Tárgy interfacet

- **Attribútumok**

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String
- **vedIdo:** A maszk védelméből hátra lévő időt tárolja, -, int

- **Metódusok**

- **String toString():** Visszaad egy stringet a következő formában: “név” :Maszk., +
- **void accept(TargyVisitor):** ha védelmi idő nagyobb mint 0, fogadja a visitort és csökkenti a védelmi időt, +
- **int getVedIdo():** visszaadja a védelmi időt, +

8.1.12 Pohar

- **Felelősség**

A Pohár osztály felelős azért, hogy tárolja és kezelje a söröspohár tulajdonságait és funkcióit a labirintusban, mint például használható-e a tárgy védelem nyújtásra az oktatókkal szemben

- **Össztályok**

Nincs

- **Interfészek**

Megvalósítja a Tárgy interfacet

- **Attribútumok**

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String
- **vedIdo:** A pohár védelméből hátra lévő időt tárolja, -, int
- **tulaj:** Az embert tárolja akinél a tárgy van, -, Ember

- **Metódusok**

- **String toString():** Visszaad egy stringet a következő formában: “név” :Pohar., +
- **void hasznal():** a tárgy használata védelemre, +
- **void emberValtasrolErtesit(Ember):** átállítja a tulaj értékét a megadott értékre, +
- **void accept(TargyVisitor):** fogadja a visitort, +

- **boolean használhatóE():** visszaadja hogy a vedIdo nagyobb e mint 0, +
- **void tick():** idő léptetése, csökken a védelmi idő, +

8.1.13 Rongy

· Felelősség

Az osztály felelős a nedves táblatörlő rongy kezeléséért a játék során. Lehet őket használni védekezésre oktatók ellen.

• Ősosztályok

Nincs

· Interfészek

Megvalósítja a Tárgy interfacet

· Attribútumok

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String
- **meddigNedves:** A rongy védelméből hátra lévő időt tárolja, -, int
- **jelenlegiSzoba:** A szoba amelyben a tárgy tartózkodik, #?, Szoba

· Metódusok

- **String toString():** Visszaad egy stringet a következő formában: “név” :Rongy., +
- **void hasznal(Oktato):** a függvény minden a szobában tartózkodó emberre meghívja a rongyelszenved() függvényt, +
- **void szobaValtasrolErtesit(Szoba):** átállítja a jelenlegiSzoba értékét a megadott értékre, +
- **void accept(TargyVisitor):** fogadja a visitort, +
- **boolean használhatóE():** visszaadja hogy a meddigNedves nagyobb e mint 0, +
- **void tick():** idő léptetése, csökken a meddigNedves, +
- **getMeddigNedves():** visszaadja a meddigNedvest, +

8.1.14 TVSZ

· Felelősség

Ez az osztály felelős a TVSZ kezeléséért és tulajdonságaiért a játékban, védelem nyújtása és annak fentállásának ellenőrzésért

• Ősosztályok

Nincs

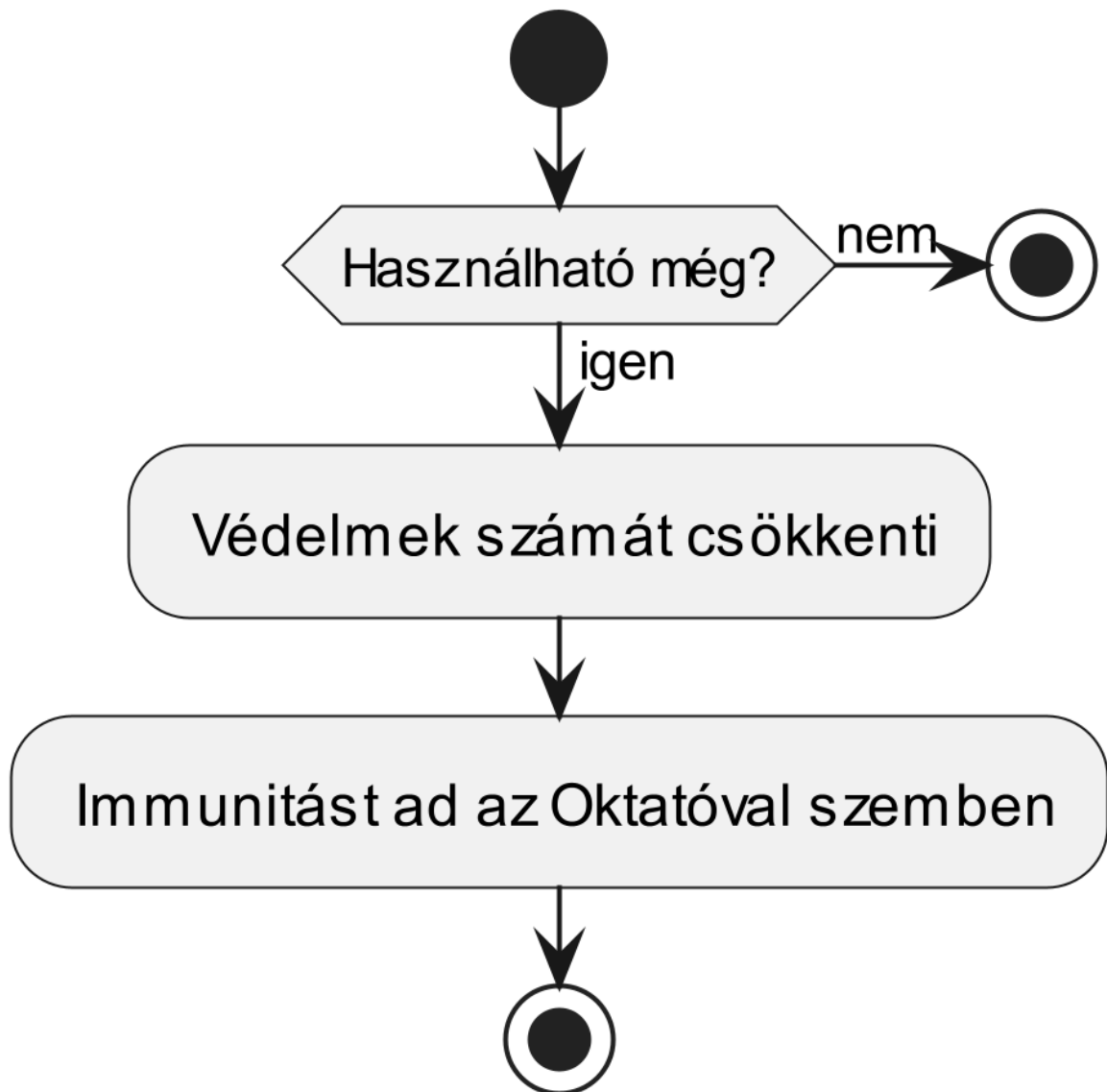
· Interfészek

Megvalósítja a Tárgy interfacet

· Attribútumok

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String

- **vedelmekSzama:** tárolja hány védelem van hátra, -, int
- **jelenlegiSzoba:** A szoba amelyben a tárgy tartózkodik, -, Szoba
- **Metódusok**
 - **String toString():** Visszaad egy stringet a következő formában: “név” :TVSZ., +
 - **void hasznal(Oktato):** ha van védelem hátra akkor csökkenti a védelmek számát és a jelenlegiSzoban meghívja az immunitastAd() függvényt., +



- **void szobaValtasrolErtesit(Szoba):** átállítja a jelenlegiSzoba értékét a megadott értékre, +
- **void accept(TargyVisitor):** fogadja a visitort, +
- **boolean használhatóE():** visszaadja hogy a vedelmekSzama nagyobb e mint 0, +

8.1.15 Tranzisztor

• Felelősség

A Tranzisztor osztály felelős a játékosok általi mozgás segítéséért. Segítésével a

játékosok képesek összekapcsolni két különböző szobát, hogy gyorsabban mozoghassanak a labirintusban.

• Ősosztályok

Nincs

• Interfészek

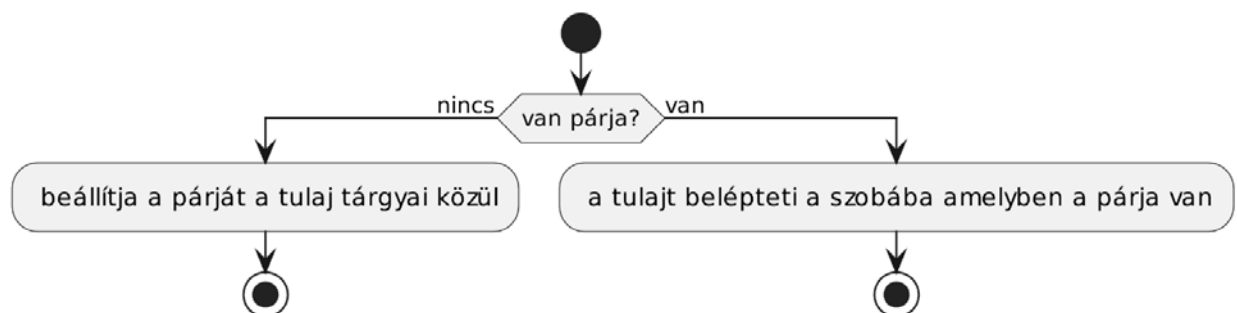
Megvalósítja a Tárgy interfacet

• Attribútumok

- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, -, String
- **par:** a tranzisztor objektum amivel párosítva lett, -, Tranzisztor
- **jelenlegiSzoba:** A szoba amelyben a tárgy tartózkodik, -, Szoba
- **tulaj:** Az embert tárolja akinél a tárgy van, -, Ember

• Metódusok

- **String toString():** Visszaad egy stringet a következő formában: “név” :Tranzisztor., +
- **void hasznal():** Ha nincs párja a függvény párosítja a tulajnál lévő másik Tranzisztor objektummal, ha pedig már van párja, a tulajdonost oda teleportálja a `masikSzobabaLep(Szoba)` függvény meghívásával a tulajon a `par GetSzoba()` értékével, +



- **void szobaValtasrolErtesit(Szoba):** átállítja a jelenlegiSzoba értékét a megadott értékre, +
- **void emberValtasrolErtesit(Ember):** átállítja a tulaj értékét a megadott értékre, +
- **Szoba getSzoa():** visszaadja a jelenlegiSzoba értékét
- **void setPar(Tranzisztor):** beállítja a par értékét a megadott objektumra, továbbá a megadott objektum par attribútuma értékét is beállítja a jelenlegi objektum értékére, így egymás párjai lesznek, +

8.1.16 Takarító

- **Felelősség**

A takarító osztály felelős a takarító karakterek játékbeli reprezentációjáért. A takarító megszünteti a mérgező gázt a szobában ahová belép.

- **Össosztályok**

Ember -> Takarító

- **Interfészek**

Nincs

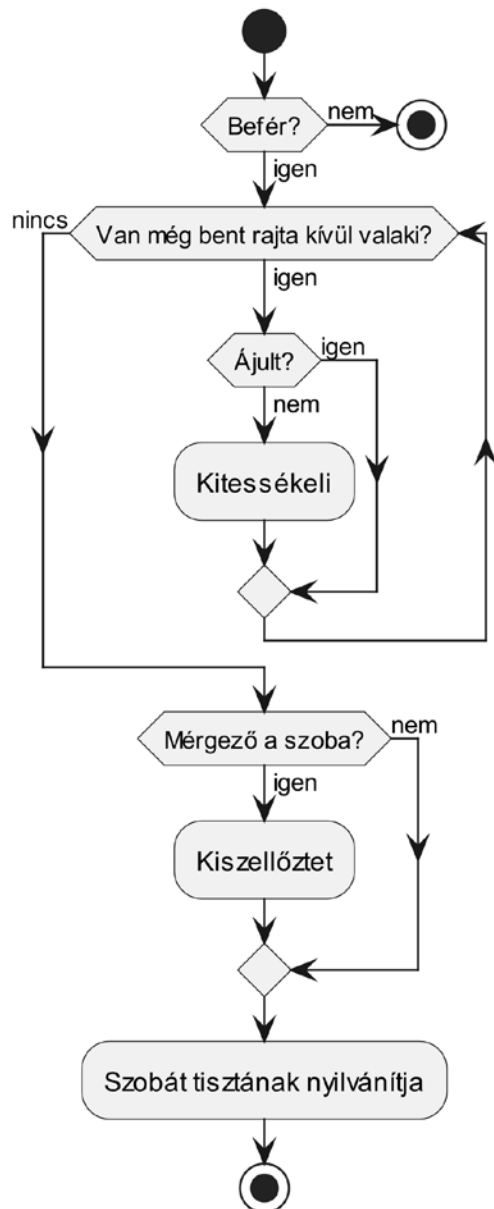
- **Attribútumok**

Attribútumait az Ember őssosztálytól örökli.

- **inventory:** a karakternél található tárgyakat tárolja, #, List<Targy>
- **jelenlegiSzoba:** a szobát tárolja, amiben a karakter éppen tartózkodik, #, Szoba
- **nev:** az objektum nevét (pl. t1) tárolja, csak a prototípusnál és a szkeletonnál használatos, #, String

- **Metódusok**

- **Takarito(String nev):** Létrehoz egy Takarítót a paraméterként kapott névvel., +
- **String toString():** Visszaad egy stringet a következő formában: “név” :Takarito. (pl. t :Takarito)., +
- **boolean inventoryTeleE():** Mindig hamissal tér vissza, hiszen a Takarító nem tud tárgyakat magával hordozni., +
- **void masikSzobabaLep(Szoba sz):** A metódus megpróbálja betenni a Takarítót a paraméterként kapott sz Szobába, ha ez nem sikerül (mert a Szoba már tele van) akkor visszatér. Egyébként a Takarítót beteszi a paraméterként kapott sz szobába és a jelenlegi szobáját is átállítja erre. Ezután kitessékeli a nem ájult embereket a szobából és ha mérgező a szoba, akkor kiszellőztet, ezzel megszüntetve a mérgező gázt a szobában. A takarítás végezetével a szobát tisztának nyilvánítja., +



8.1.17 Hallgató

- **Felelősség**

A Hallgato osztály felelős a játékos karakterének kezeléséért a labirintusban, beleértve a tárgyak felvételét és letételét, valamint a szobák közötti mozgást.

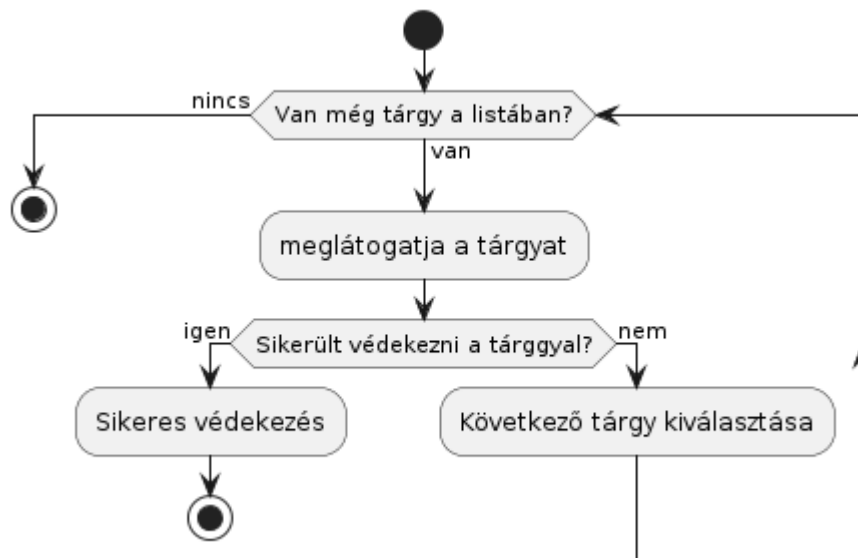
- **Össztályok**

Ember

- **Interfészek**

Nincsenek interfészek, amelyeket a Hallgato osztály megvalósít.

- **Attribútumok**
- **tudVedekezni:** Logikai érték, amely azt jelzi, hogy a hallgató tud-e védekezni, - , boolean
- **életbenVan:** Logikai érték, amely azt jelzi, hogy a hallgató életben van-e, - , boolean
- **Metódusok**
- **boolean getElembenVan():** Visszaadja, hogy a hallgató életben van-e, +
- **void setElembenVan(boolean életbenVan):** Beállítja a hallgató életbenVan állapotát, +
- **void visit(Logarlec logarlec):** A függvény meglátogatja a megadott logarlécet és a meghívja a labirintus játékNyert függvényét amivel véget vet a játéknak, +
- **void visit(TVSZ tvsz):** A függvény meglátogatja a megadott TVSZ-t és megvédi a hallgatót a szükség esetén ha a tvsz még működőképes, +
- **void visit(Pohar pohar):** A függvény meglátogatja a megadott poharat és megvédi a hallgatót a szükség esetén ha a pohár még nem járt le, +
- **void visit(Rongy rongy):** A függvény meglátogatja a megadott rongyot és megvédi a hallgatót a szükség esetén, +
- **boolean inventoryTeleE():** Ellenőrzi, hogy a hallgató hátizsákja tele van-e, +
- **void tamadasElszenved(Oktato oktato):** A függvény alkalmazza az oktató támadásából eredő hatásokat a hallgatóra, majd ellenőrzi, hogy van-e védelme ezekkel szemben,



+
meghal(): Ha a hallgató meghal akkor beállítja az életbenVan attribútumot hamisra, -
void tick(): Frissíti a hallgató állapotát a játékmenet előrehaladtával, +

8.1.18 Oktató

- **Felelősség**

Az Oktato osztály felelős a nem játékos karakterek kezeléséért a labirintusban, különösen azáltal, hogy megnehezíti a játékos dolgát a játék során.

- **Ősosztályok**

Az Oktato osztály az Ember ősosztályból származik.

- **Interfészek**

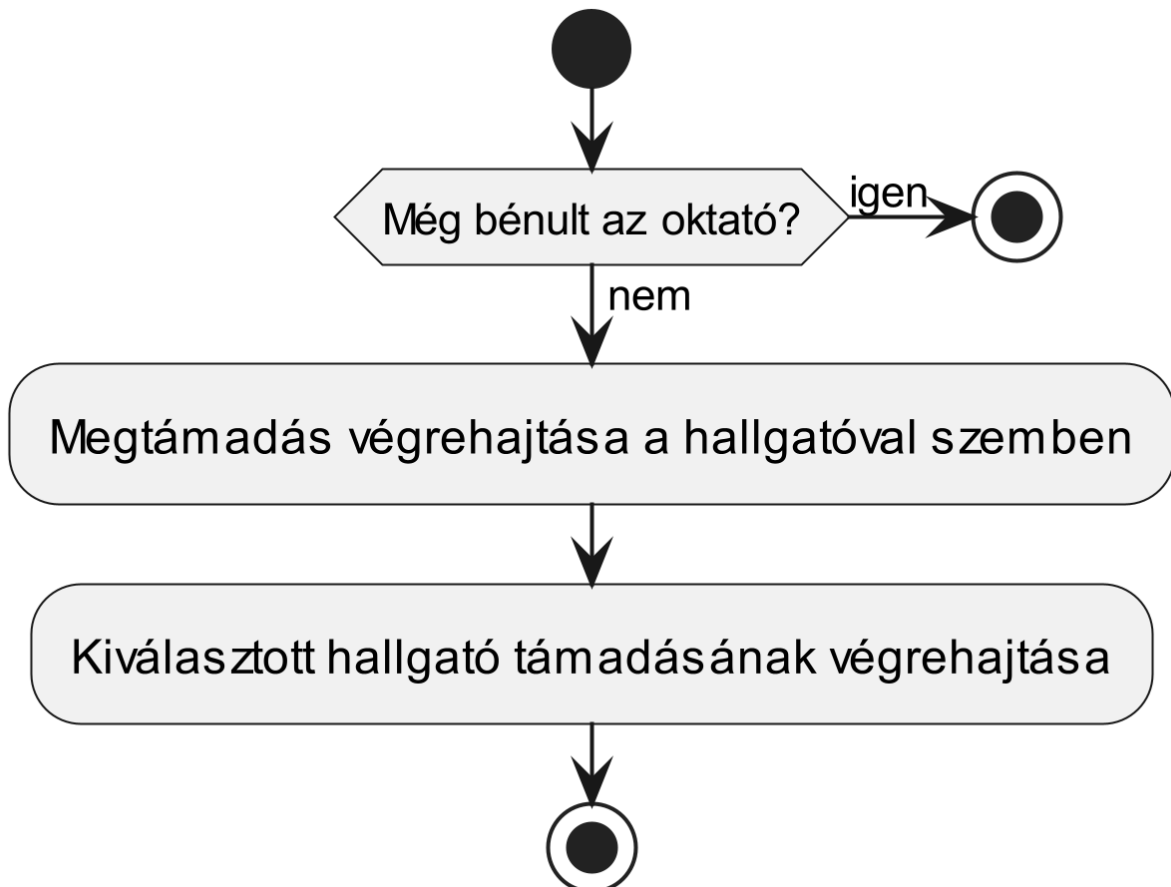
Nincsenek interfészek, amelyeket az Oktato osztály megvalósít.

- **Attribútumok**

- **MAX_INVENTORY_MERET** : privát konstans attribútum, amely meghatározza az oktató inventory-jának maximális méretét, - , int
- **meddigBena** : privát egész szám attribútum, amely tárolja, hogy meddig marad még bénult az oktató, - , int
- **kitTamad**: privát referenciaváltozó egy Hallgato objektumra, amely megmondja, hogy melyik hallgatót támadja az oktató, - , Hallgato

- **Metódusok**

- **boolean inventoryTeleE()**: publikus metódus, amely visszaadja, hogy az oktató zsebe tele van-e, +
- **void rongyotElszenved(Rongy rongy)**: publikus metódus, amely kezeli az oktatót érő rongy hatásait, és beállítja a meddigBena értékét, +
- **void tick()** : publikus metódus, amely frissíti az oktató állapotát a játékmenet előrehaladtával, +
- **void hallgatotMegtamad(Hallgato hallgato)**: Ez a publikus metódus felelős egy adott hallgató megtámadásáért az oktató által. Ha az oktató még nem bénult, akkor megtámadja a kapott hallgatót, és kezeli a támadás hatásait. Amennyiben az oktató még bénult, a metódus erről tájékoztatást nyújt, és nem hajtja végre a támadást. A metódus végül beállítja azt a hallgatót, amelyet az oktató támad., +



- **Hallgato getKitTamad():** publikus metódus, amely visszaadja a hallgatót, akit támad az oktató, +

8.1.19 TárgyVisitor

- **Felelősség**

A TárgyVisitor interface segít abban, hogy a tárgyak különbözően viselkedjenek, attól függően, hogy ki birtokolja őket.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

Nincs

- **Metódusok**

void visit(Logarlec l): A szkeleton esetén kilogolja, hogy ki látogatta meg a tárgyat, egyébként csak az interfacet implementáló osztályok implementálják a megfelelő logikát, +

void visit(TVSZ t): A szkeleton esetén kilogolja, hogy ki látogatta meg a tárgyat, egyébként csak az interfacet implementáló osztályok implementálják a megfelelő logikát, +

void visit(Pohar p): A szkeleton esetén kilogolja, hogy ki látogatta meg a tárgyat, egyébként csak az interfacet implementáló osztályok implementálják a megfelelő logikát, +

void visit(Rongy r): A szkeleton esetén kilogolja, hogy ki látogatta meg a tárgyat, egyébként csak az interfacet implementáló osztályok implementálják a megfelelő logikát, +

void visit(Maszk m): A szkeleton esetén kilogolja, hogy ki látogatta meg a tárgyat, egyébként csak az interfacet implementáló osztályok implementálják a megfelelő logikát, +

void visit(Legfrissito l): A szkeleton esetén kilogolja, hogy ki látogatta meg a tárgyat, egyébként csak az interfacet implementáló osztályok implementálják a megfelelő logikát, +

8.1.20 Időzített

- **Felelősség**

Az Időzített interface segít abban, hogy a labirintusban található különböző objektumokon az idő múlását lehessen reprezentálni.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

Nincs

- **Metódusok**

void tick(): A szkeleton esetén kilogolja, hogy tick történt, egyébként az interfacet implementáló osztályok implementálják a megfelelő logikát.,+

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 Hallgató felvesz tárgyakat

- **Leírás**

Egy hallgató felvesz tárgyakat, amik az inventoryjába kerülnek, amennyiben abban még van elég hely.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az emberek tárgyat felvevő funkcionalitása. Ebben az esetben hallgatóval próbálkozunk, de a közös logika miatt az oktató és a takarító esetében is hasonló kimenetet kell kapnunk, ami egyedül az inventoryjuk mérete miatt térhet el. Hibalehetőség lehet, hogy nem létező tárgyat próbálunk felvenni.

- **Pályaleíró:**

szoba1:,tisztá,1

emberek:

targyak:tvsz, tvsz, tvsz, tvsz, tvsz, tvsz

- **Bemenet**

load tesztpalya

play 1 1 szoba1

targyfelvesz tvsz1 hallgato1

targyfelvesz tvsz2 hallgato1

targyfelvesz tvsz3 hallgato1

targyfelvesz tvsz4 hallgato1

targyfelvesz tvsz5 hallgato1

targyfelvesz tvsz6 hallgato1

- **Elvárt kimenet**

a tesztpalya1 jatek betoltve.

szobak:

szoba1

szomszedok:

targyak:tvsz1,tvsz2,tvsz3,tvsz4,tvsz5,tvsz6

emberek:

allapot: tisztá

a palya legeneralva 1 fore.

kezdo szoba: szoba1

a jatek elindult, 1 korig tart.

hallgato1: az inventorydba tetted a tvsz1 tárgyat.

hallgato1: az inventorydba tetted a tvsz2 tárgyat.

hallgato1: az inventorydba tetted a tvsz3 tárgyat.

hallgato1: az inventorydba tetted a tvsz4 tárgyat.

hallgato1: az inventorydba tetted a tvsz5 tárgyat.

hallgato1: az inventoryd teli van, nem fer bele a tvsz6 tárgy.

8.2.2 Ember eldob egy tárgyat

- **Leírás**

Egy hallgató megpróbál eldobni egy tárgyat az inventoryjából, de nincs nála semmi.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ember tárgyeldobás funkciója. Ebben az esetben hallgatóval próbálkozunk, de a közös logika miatt az oktató és a takarító is ugyanilyen futást kell eredményezzen. Hibalehetőség lehet, hogy nem létező tárgyat próbálunk eldobni

- **Pályaleíró**

szoba1:,tiszta,1

emberek:

targyak:

- **Bemenet**

load tesztpalya

play 1 1 szoba1

targyeldob t1 hallgato1

- **Elvárt kimenet**

a tesztpalya1 jatek betoltve.

szobak:

szoba1

szomszedok:

targyak:

emberek:

allapot: tiszta

a palya legeneralva 1 fore.

kezdo szoba: szoba1

a jatek elindult, 1 korig tart.

hallgato1: az inventoryd ures nem dobhatsz el tárgyat.

8.2.3 Oktató támad

- **Leírás**

A hallgató belép egy olyan szobába, ahol éppen egy oktató tartózkodik, aki így rátámad

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az oktató hallgató elleni támadása

- **Pályaleíró**

szoba1:szoba2,tiszta,1

emberek:

targyak:

szoba2:szoba1,tiszta,2

emberek:oktato1

targyak:

- **Bemenet**

load tesztpalya

play 1 1 szoba1

ajtohasznalat ajto12 hallgato1

infoember hallgato1

- **Elvárt kimenet**

a tesztpalya jatek betolve.

szobak:

szoba1

szomszedok:szoba2

targyak:

emberek:

allapot: tiszta

szoba2

szomszedok:szoba1

targyak:

emberek:oktato1

allapot:tiszta

a palya legeneralva 1 fore.

kezdo szoba: szoba1

a jatek elindult 1 korig tart

hallgato1: atleptel az ajto12-n

nincs ilyen ember hallgato1

8.2.4 Ember gázos szoba interakció

- **Leírás**

Egy hallgató belép egy mérges gázzal teli szobába és maszk hiányában elájul.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A mérgező gáz emberekre gyakorolt élettani hatásai. Várható hiba, hogy a gáz semmilyen hatással nincs az emberekre

- **Pályaleíró**

szoba1:szoba2,tiszta,1

emberek:

targyak:

szoba2:szoba1,mergezo,1

emberek:

targyak

- **Bemenet**

load tesztpalya

play 1 1 szoba1

ajtohasznalat a12 hallgato1

infoember

- **Elvárt kimenet**

a tesztpalya jatek betolve.

szobak:

szoba1

szomszedok:szoba2

targyak:

emberek:

allapot: tiszta

szoba2

szomszedok:szoba1

tárgyak:
 emberek:
 állapot:mergezo
 a palya legeneralva 1 fore.
 kezdo szoba: szoba1
 a jatek elindult 1 korig tart
 hallgato1: atleptel az ajto12 ajton
 hallgato1 tárgyai:
 itt van: szoba2
 ajult

8.2.5 A hallgató teleportál

- **Leírás**

A hallgató párosít 2 tranzisztort, az egyiket eldobja, majd átlép egy másik szobába, ahol a nála levő tranzisztort használja.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A tranzisztor párosítási, illetve teleportálási funkcionalitása. Hiba lehet, hogy a tranzisztor párosítása, illetve a teleportálás sikertelen

- **Pályaleíró**

szoba1:szoba2,tiszta,1
 emberek:
 tárgyak:tranzisztor,tranzisztor

szoba2:szoba1,tiszta,1
 emberek:
 tárgyak:

- **Bemenet**

load tesztpalya
 play 1 1 szoba1
 tárgyfelvesz tranzisztor1 hallgato1
 tárgyfelvesz tranzisztor2 hallgato1
 hasznal tranzisztor1 hallgato1
 tárgyeldob tranzisztor1 hallgato1
 ajtohasznalat ajto12 hallgato1
 hasznal tranzisztor2 hallgato1

- **Elvárt kimenet**

a tesztpalya jatek betolte.
 szobak:
 szoba1
 szomszedok:szoba2
 tárgyak:tranzisztor1,tranzisztor2
 emberek:
 állapot: tiszta

 szoba2
 szomszedok:szoba1
 tárgyak:
 emberek:
 állapot:tiszta

a palya legeneralva 1 fore.
 kezdő szoba: szoba1
 a játék elindult 1 korrig tart
 hallgató1: az inventorydba tetted a tranzisztor1 tárgyat.
 hallgató1: az inventorydba tetted a tranzisztor2 tárgyat.
 hallgató1: a tranzisztor1 tranzisztort parositja a másik taskaban lévő tranzisztorral.
 hallgató1: az inventorydból a földre dobta a tranzisztor1 tárgyat.
 hallgató1: átleptel az ajtó12 ajton.
 hallgató1: a tranzisztor2 parjához teleportálunk

8.2.6 Szoba elgázosítása

- **Leírás**

Egy szobában camambert használ egy hallgató, aminek hatására a szobát ellepi a mérges gáz.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A camembert mérges gáz kibocsátó hatása. Hiba lehet, hogy felbontáskor a szoba nem telik meg gázzal.

- **Pályaleíró**

szoba1:,tisztá,1
 emberek:
 tárgyak:camembert

- **Bemenet**

load teszt_palya
 play 1 1 szoba1
 tárgyfelvesz camembert1 hallgató1
 használ camembert1 hallgató1
 info_szoba szoba1

- **Elvárt kimenet**

a teszt_palya játék betöltve.
 szobák:
 szoba1
 szomszédok:
 tárgyak:camembert1
 emberek:
 állapot: tisztá

a palya legeneralva 1 fore.
 kezdő szoba: szoba1
 a játék elindult 1 korrig tart
 hallgató1: az inventorydba tetted a camembert1 tárgyat.
 hallgató1: a camembert1 elgázosítja a szoba1 szobát.
 a szoba állapota:mergezo
 a szobában lévő tárgyak:
 a szobában lévő oktatok:
 a szobából nyíló ajtók:

8.2.7 Ragacsos tárgy felvétel

- **Leírás**

Egy ember megpróbál felvenni egy tárgyat a ragacsos szobából, de ez nem sikerül neki mert a tárgy le van ragadva.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A ragacsos szoba hatása a tárgyra. Hiba lehet, hogy a tárgyat sikerül felvenni, annak ellenére, hogy a szoba ragacsos

- **Pályaleíró**

szoba1:,ragacsos,1

emberek:

targyak:tvosz

- **Bemenet**

load tesztplaya

play 1 1 szoba1

targyfelvesz tvosz1 hallgato1

- **Elvárt kimenet**

a tesztplaya jatek betoltve.

szobak:

szoba1

szomszedok:

targyak:tvosz1

emberek:

allapot: ragacsos

a palya legeneralva 1 fore.

kezdo szoba: szoba1

a jatek elindult 1 korig tart

hallgato1: a szoba ragacsos a tvosz tárgy nem veheto fel.

8.2.8 Hamis logarléc felvétele

- **Leírás**

A Hallgató felvesz egy hamis logarléct, aminek hatására semmi nem történik

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Lényegében semmi, mert a hamis tárgyaknak nincsen semmilyen funkciója azon kívül, hogy a játékos becsapják. Hiba lehet, hogy mégis történik valamilyen nem várt viselkedés a tárgy birtoklásának hatására.

- **Pályaleíró**

szoba1:, tiszta, 1

emberek:

targyak: hamislec

- **Bemenet**

load tesztplaya

play 1 2 szoba1

targyfelvesz hamislec1 hallgato1

tick 1

- **Elvárt kimenet**

a tesztplaya jatek betoltve.

szobak:

szoba1

szomszedok:

targyak:hamislec1

emberek:

állapot: ragacsos
 a palya legeneralva 1 fore.
 kezdő szoba: szoba1
 a játék elindult 1 korrig tart
 hallgató1: az inventorydba tetted a hamislec tárgyat.

8.2.9 A takarító nem üres gázos szobába lép

- **Leírás**

A takarító nem üres gázos szobába lép, kiszellőzteti a gázt és kitessékeli az embereket egy szomszédos szobába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A takarító kitessékeli, illetve szellőztető funkcionalitása. Hiba lehet, hogy a takarító újult embereket is kitessékeli, illetve nem szellőztet ki.

- **Pályaleíró**

szoba1: szoba2, tiszta, 4
 emberek: takarító1
 tárgyak:

szoba2: szoba1, tiszta, 5
 emberek: oktató1, oktató2
 tárgyak: Maszk, Maszk, TVSZ

- **Bemenet**

load tesztplaya
 play 2 10 szoba2
 tárgyfelvesz maszk1 hallgató1
 tárgyfelvesz tvsz hallgató1
 tárgyfelvesz maszk2 oktató1
 infoember hallgató1
 infoember hallgató2
 infoember oktató1
 infoember oktató2
 ajtóhasználat ajtó12 takarító
 infoember hallgató1
 infoember hallgató2
 infoember oktató1
 infoember oktató2
 infoszoba szoba1
 infoszoba szoba2

- **Elvárt kimenet**

a tesztplaya játék betöltve.
 szobák:
 szoba1
 szomszédok: szoba2
 tárgyak:
 emberek:
 állapot: tiszta

 szoba2

szomszedok:szoba1
targyak:maszk1,maszk2,tvsz1
emberek:oktato1,oktato2
allapot: tiszta
a palya legeneralva 2 fore.
kezdo szoba: szoba2
a jatek elindult 10 korig tart
a palya legeneralva 2 fore.
hallgato1: az inventory-ba tetted a maszk1 tárgyat.
hallgato1: az inventory-ba tetted a tvsz tárgyat.
oktato1: az inventory-ba tetted a maszk2 tárgyat.
hallgato1 tárgyai: maszk1, tvsz
itt van: szoba2
eber
hallgato2 tárgyai:
itt van: szoba2
ajult.
oktato1 tárgyai: maszk2
itt van: szoba2
eber
oktato2 tárgyai:
itt van: szoba2
ajult.
takarito: atleptel az ajto12 ajton.
hallgato1: atleptel az ajto21 ajton.
oktato1: atleptel az ajto21 ajton.
hallgato1 tárgyai: maszk1, tvsz
itt van: szoba1
eber.
hallgato2 tárgyai:
itt van: szoba2
ajult.
oktato1 tárgyai: maszk2
itt van: Szoba1
eber.
oktato2 tárgyai:
itt van: szoba2
ajult.
a szoba allapota: tiszta
a szobaban levo targyak:
a szobaban levo oktatok: oktato1
a szobabol nyilo ajtok: ajto12
a szoba allapota: mergezo
a szobaban levo targyak:
a szobaban levo oktatok: oktato2
a szobabol nyilo ajtok: ajto21

8.2.10 A takarító üres gázos szobába lép

- Leírás

A takarító üres gázos szobába lép és kiszellőztet, aminek hatására eltűnik a mérges gáz a szobából.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A takarító szellőztető funkciója. Hiba lehet, hogy a szoba gázos marad.

- **Pályaleíró**

Szoba1:Szoba2, tiszta, 2

Emberek:

Tárgyak:

Szoba2: Szoba1, mergezo, 1

Emberek: Takarító1

Tárgyak:

- **Bemenet**

load tesztplaya

play 1 1 szoba1

infoszoba Szoba2

Tick 1

Ajtohasznalat Ajto12 Takarito1

Tick 1

InfoSzoba Szoba2

Tick 1

- **Elvárt kimenet**

a tesztplaya jatek betoltve.

szobak:

szoba1

szomszedok:szoba2

targyak:

emberek: takarito1

allapot: tiszta

szoba2

szomszedok:szoba1

targyak:

emberek:

allapot: mergezo

a palya legeneralva 1 fore.

kezdo szoba: szoba1

a jatek elindult 1 korig tart

Takarito1: Átléptél az Ajto12 ajtón

A szoba állapota: Tiszta

A szobában lévő tárgyak:

A szobában lévő oktatók száma:

A szobából nyíló ajtók: Ajto21

8.2.11 Szoba osztódás

- **Leírás**

A labirintus egy szobájának két szobára történő felosztása.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Felosztás megfelelő működése, esetleges hiba lehet az emberek és ajtók nem megfelelő szétosztása

- **Pályaleíró**

szoba1:szoba2,tiszta,1
emberek:oktato1,oktato2,oktato3
targyak:tvosz,pohar

szoba2:szoba1,tiszta,1
emberek:
targyak:

- **Bemenet**

Load teszt-palya
SzobaFeloszt Szoba1
InfoSzoba Szoba1
InfoSzoba Szoba3

- **Elvárt kimenet**

a teszt-palya jatek betoltve
szobak:

szoba1
szomszedok: szoba2
targyak:tvosz1,pohar1
emberek:oktato1,oktato2,oktato3
allapot:tiszta

szoba2
szomszedok: szoba1
targyak:
emberek:
allapot:tiszta

uj szoba jott létre: szoba3
a szoba allapota: tiszta
a szobaban levo targyak: pohar
a szobaban levo oktatok: oktato3
a szobabol nyilo ajtok: ajto13
a szoba allapota: tiszta
a szobaban levo targyak: tvosz
a szobaban levo oktatok: oktato1,oktato2
a szobabol nyilo ajtok: ajto31, ajto32

8.2.12 Szoba összevonás

- **Leírás**

A labirintus két szobája összevonásra kerül

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szobák összevonásának megfelelő működése, esetleges hiba lehet nem szomszédos szobák összevonása vagy emberek és ajtók eltűnése az éterben

- **Pályaleíró**

Szoba1:Szoba2, tiszta, 1
Emberek:
Tárgyak: TVSZ,

Szoba2: Szoba1, Szoba3, Szoba4, tiszta, 1

Emberek:

Tárgyak: Logarléc

- **Bemenet**

load tesztplan

SzobaOsszevon Szoba1 Szoba2

InfoSzoba Szoba1

InfoSzoba Szoba2

- **Elvárt kimenet**

a tesztplan jatek betöltve

szobák:

szoba1

szomszedok: szoba2

targyak:tvsl,

emberek:oktato1

allapot:tiszta

szoba2

szomszedok: szoba1

targyak:logarlec1

emberek: oktato2,oktato3

allapot:tiszta

szoba1 es szoba2 ossze lett vonva

A szoba állapota: Tiszta

A szobában lévő tárgyak: tvsl, logarlec1

A szobában lévő oktatók: oktato1,oktato2,oktato3

A szobából nyíló ajtók:

Nincs ilyen szoba szoba2

8.2.13 Ember nem fér be a szobába

- **Leírás**

Az ember át szeretne lépni egy szobába, ami tele van így nem tud belépni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az az ember aki egyedül van egy szobában megpróbál átlépni a már tele lévő szobába, ami nem valósulhat meg. Várható hibahely ha sikeresen átlép a már tele lévő szobába.

- **Pályaleíró**

Szoba2:Szoba1, tiszta, 2

Emberek: Oktató1, Oktató2

Tárgyak:

Szoba1: Szoba2, tiszta, 1

Emberek:

Tárgyak:

- **Bemenet**

load tesztplan

InfoSzoba szoba2

InfoEmber hallgato1

Ajtohasznalat a1 hallgato1

InfoSzoba szoba2

InfoEmber hallgato1

- **Elvárt kimenet**

A tesztplaya jatek betoltve.

Szobák:

Szoba2:

Szomszédok: Szoba1

Emberek: Oktató1, Oktató2

Tárgyak:

Állapot: tiszta

Szoba1:

Szomszédok: Szoba2

Emberek:

Tárgyak:

Állapot: tiszta

A szoba állapota: tiszta

A szobában lévő tárgyak:

A szobában lévő oktatók száma: 2

A szobából nyíló ajtók: 1

hallgato1 tárgyai:

Itt van: szoba1

Eber

A szomszéd szoba megtelt. Nem férsz be.

A szoba állapota:

A szobában lévő tárgyak:

A szobában lévő oktatók száma: 2

A szobából nyíló ajtók: 1

hallgato1 tárgyai:

Itt van: szoba1

Eber

8.2.14 Lejárt az idő

- **Leírás**

A logarléc megtalálására alokált idő elfogyott, a játéknak vége van.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tick metódus az egyetlen lehetséges hibaforrás

- **Pályaleíró**

szoba1:tiszta,1

emberek:

targyak:

- **Bemenet**

load tesztplaya

play 1 1 szoba1

tick 1

- **Elvárt kimenet**

A tesztpálya játék betöltve.

Szobák:

Szoba1:

Szomszédok:

Emberek:

Tárgyak:

Állapot: tiszta

A pálya legenerálva 1 főre.

Kezdő szoba: Szoba1

A játék elindult, 1 körig tart.

A játéknak vége, az idő lejárt.

8.2.15 Hamis tárgyhasználat

- **Leírás**

Egy ember egy hamis tárgyat próbál használni, de a sajátos tulajdonság, ami a valódi tárgyra jellemző nem jön elő, így nem történik semmi.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrzésre kerül, hogy hamis tárgyak tényleg nem rendelkeznek a valódi tárgyakra jellemző tulajdonsággal. Hiba lehet, hogy valamilyen csoda folytán mégis rendelkeznek a valódi tárgyra jellemző tulajdonsággal.

- **Pályaleíró**

Szoba1: , Gázos, 1

Emberek:

Tárgyak: Hamis TVSZ, Hamis Logarléc, Hamis Maszk

- **Bemenet**

load tesztpálya

play 1 2 szoba1

targyfelvesz hamistvsz1

targyfelvesz hamisllec1

targyfelvesz hamismaszk1

infoember hallgato1

tick 1

infoember hallgato1

- **Elvárt kimenet**

a tesztpálya játék betöltve

Szobák:

Szoba1:

Szomszédok:

Emberek:

Tárgyak: Hamis TVSZ, Hamis Logarléc, Hamis Maszk

Állapot: gázos

a palya legeneralva 1 fore
 kezdő szoba: szoba1
 a játék elindult 2 korig tart
 hallgató1: az inventory-ba tetted a hamistvsz1 tárgyat.
 hallgató1: az inventory-ba tetted a hamislec1 tárgyat.
 hallgató1: az inventory-ba tetted a hamismaszkl1 tárgyat.
 hallgató1 tárgyai: hamistvsz1, hamislec1, hamismaszkl1
 itt van: szoba1
 eber
 hallgató1 tárgyai:
 itt van: szoba1
 ajult

8.2.16 Léghfrissítő

- **Leírás**

A léghfrissítő használatának hatására a szobából eltűnik a mérges gáz.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A léghfrissítő mérges gázt megszüntető hatása.

- **Pályaleíró**

Szoba0:Szoba1, tiszta, 1

Emberek:

Tárgyak:

Szoba1:Szoba0, Szoba2, tiszta, 1

Emberek:

Tárgyak: Léghfrissítő

Szoba2: Szoba1, Gázos, 1

Emberek:

Tárgyak:

- **Bemenet**

Load teszt-palya

Szoba0

Szomszédok: Szoba1

Emberek:

Tárgyak:

Állapot: tiszta

InfoSzoba Szoba1

Tick 1

Ajtohasznalat Ajto1 Hallgató1

Tick 1

Tárgyfelvesz legfrissítő Hallgató1

Tick 1

InfoSzoba Szoba2

Tick 1

Ajtohasznalat Ajto2 Hallgató1

Tick1

InfoSzoba Szoba2

- **Elvárt kimenet**

A tesztpalya jatek betoltve.

Szobák:

Szoba1:

Szomszédok: Szoba0, Szoba2

Emberek:

Tárgyak: Légfrissítő

Állapot: tiszta

Szoba2:

Szomszédok: Szoba1

Emberek:

Tárgyak:

Állapot: Gázos

A szoba állapota: Tiszta

A szobában lévő tárgyak: legfrissítő

A szobában lévő oktatók száma: -

A szobából nyíló ajtók: Ajtó1, Ajtó2

Hallgató1: Átléptél az Ajtó1 ajtón

Hallgató1: Az Inventory-ba tetted a legfrissítő tárgyat.

A szoba állapota: Gázos

A szobában lévő tárgyak: -

A szobában lévő oktatók száma: -

A szobából nyíló ajtók: Ajtó2

Hallgató1: Átléptél az Ajtó2 ajtón

A szoba állapota: Tiszta

A szobában lévő tárgyak: -

A szobában lévő oktatók száma: -

A szobából nyíló ajtók: Ajtó2

8.2.17 Ember egyirányú ajtót szeretne használni a rossz irányba

- **Leírás**

Az ember át szeretne lépni egy szobába, egy egyirányú ajtón, abból az irányból ahonnan nem nyitható az ajtó.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ember megpróbál átlépni egy egyirányú ajtón, ami az ő oldaláról nem nyílik. Várható hibahely ha sikeresen átlép az ajtón.

- **Pályaleíró**

Szoba1:, tiszta, 1

Emberek:

Tárgyak:

Szoba2:Szoba1, tiszta, 1

Emberek:

Tárgyak:

- **Bemenet**

load tesztpalya

InfoSzoba szoba1

InfoSzoba szoba2

InfoEmber hallgato1

Ajtohasznalat a1 hallgato1

InfoSzoba szoba2

InfoEmber hallgato1

- **Elvárt kimenet**

A tesztpalya jatek betoltve.

Szobák:

Szoba1:

Szomszédok:

Tárgyak:

Emberek:

Állapot: tiszta

Szoba2

Szomszédok: Szoba1

Tárgyak:

Emberek:

Állapot: tiszta

A szoba állapota: tiszta

A szobában lévő tárgyak:

A szobában lévő oktatók száma: 0

A szobából nyíló ajtók: 0

A szoba állapota: tiszta

A szobában lévő tárgyak:

A szobában lévő oktatók száma: 0

A szobából nyíló ajtók: 1

hallgato1 tárgyai:

Itt van: szoba1

Eber

Az a1 ajtó egyirányú és nem tudsz rajta átmenni.

A szoba állapota:

A szobában lévő tárgyak:

A szobában lévő oktatók száma: 0

A szobából nyíló ajtók: 1

hallgato1 tárgyai:

Itt van: szoba1

Eber

8.3 8.3 A tesztelést támogató programok tervei

Egy parancssori alkalmazás fog erre a célra készülni, ami indításkor kilistázza a megadott tesztelési mappában található teszteseteket, ezek közül kell a megfelelőt kiválasztani a sorszámának begépelésével.

A teszteseteket egy erre a célra kijelölt mappában kell elhelyezni a következő formában:

<teszteset mappája>

|

-> bemenet.txt, elvart.txt, tesztpalya.txt.

A bemenet.txt-nek a teszt nyelven írt bemeneti parancsokat, az elvart.txt-nek az elvárt kimenetet a prototípus kimeneti nyelvén, a tesztpalya.txt pedig a pályaleíró nyelvvel megadott pályát kell tartalmazza.

8.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.04.13. 11:00	1 óra	Ábel Bakó Csala Lévai Melegh	Feladatok felosztása
2024.04.13. 13:00	2 óra	Melegh	osztályleírás készítése.
2024.04.13. 19:00	3 óra	Lévai	osztályleírás
2024.04.13. 10:00	3 óra	Ábel	Ember, Takarító, TárgyVisitor, Időzített osztályok leírása
2024.04.13. 10:30	1 óra	Bakó	Hallgató, oktató osztályok leírása
2024.04.13. 11:00	3 óra	Csala	osztályleírás
2024.04.13. 14:00	6 óra	Bakó	Tesztek kidolgozása/ osztályleírások befejezése
2024.04.13. 15:00	3 óra	Ábel	Tesztek írása
2024.04.14. 19:30	3 óra	Melegh	Osztályleírás, tesztesetek
2024.04.14. 21:00	4 óra	Csala	Teszt
2024.04.14 17:00	6 óra	Lévai	Oszályleírás, tesztesetek
2024.04.15. 10:00	3 óra	Ábel	Tesztek átírás
2024.04.14 10:30	2.5 óra	Csala	Tesztek átírása