

Részletes tervek

35 – *pupakok*

Konzulens:
Koczó Attila

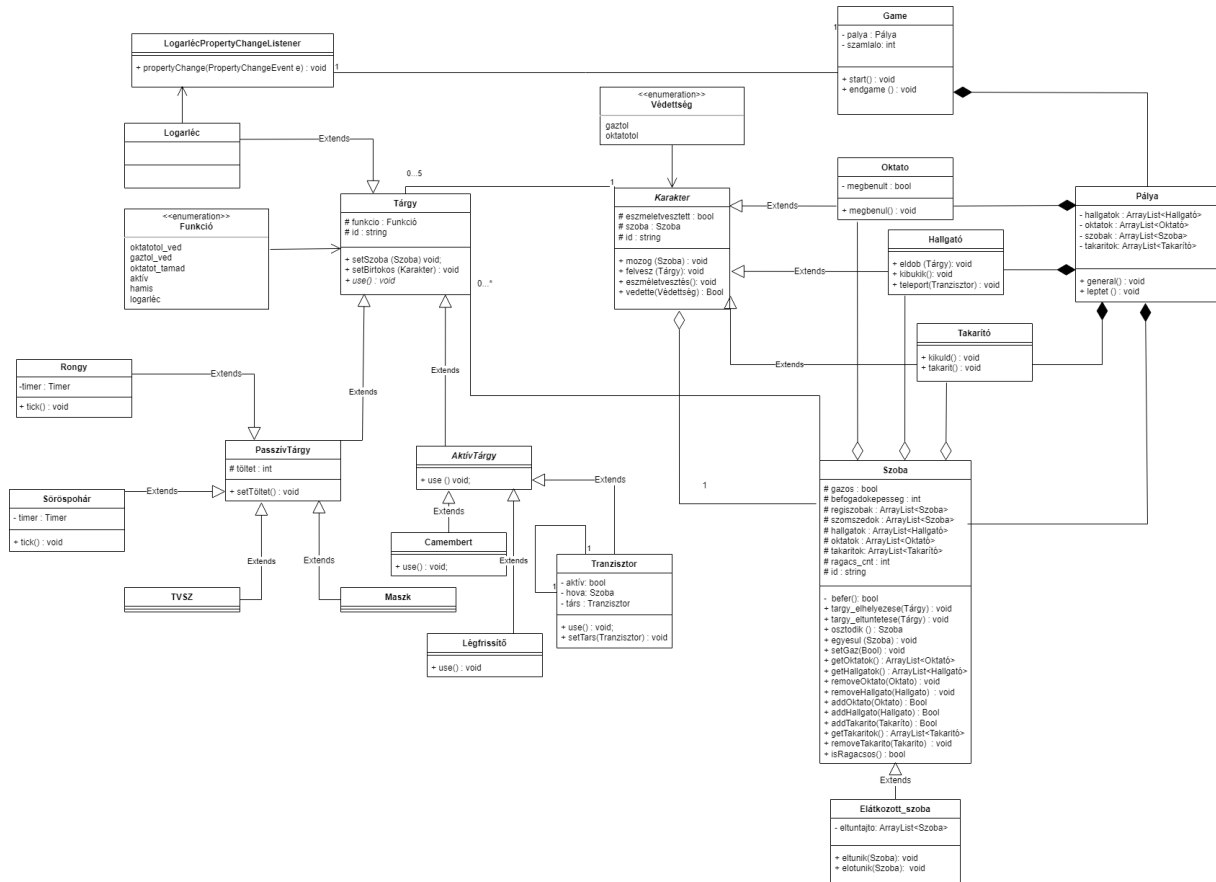
Csapattagok:

Tarsoly Levente	E1IK75	tarsolyl@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.04.15

8. Részletes tervek

8.0 Javítások



8.0.1 Add_character parancs

add_character <milyen karakter> <melyik szobába> <karakter id>

Leírás: Hozzáadunk a játékhoz egy új karaktert.

Opciók:

- <milyen karakter> hallgató hozzáadása esetén, 'hallgato', oktató hozzáadása esetén 'oktato', és takarító hozzáadása esetén 'takarito' az elvárt paraméter.
- <melyik szobába> A karakter melyik szobában kerül elhelyezésre
- <karakter id> Az elhelyezendő karaktert azonosító string, ami a karakter id attribútuma is lesz egyben.

8.0.2 Spawn_item parancs

spawn_item

Leírás: Hozzáadunk a játékhoz különböző tárgyakat.

Opciók:

- <targy> <szoba> <targy id> a paraméterben kapott tárgy típusból elhelyez egyet a paraméterül kapott szobában, a tárgy id értéke lesz a tárgy id attribútumának értéke

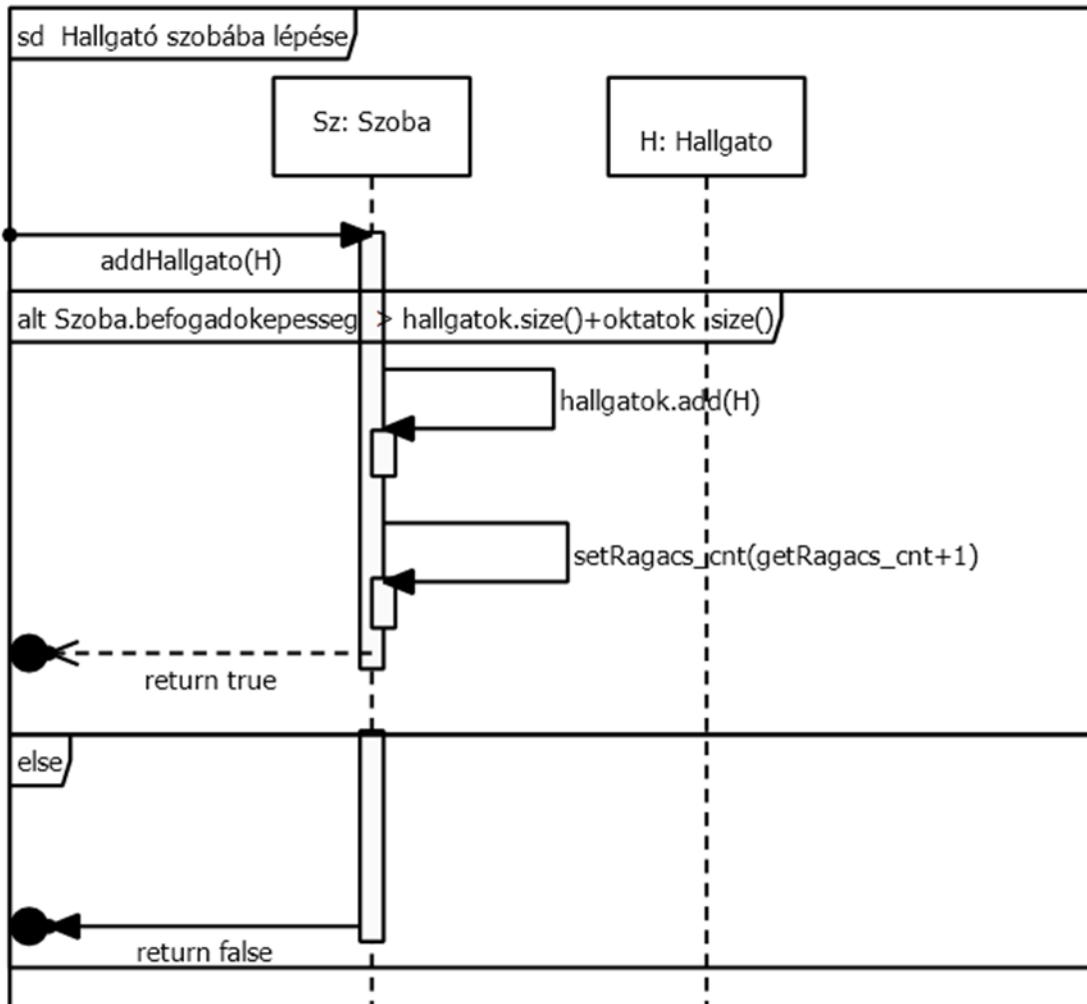
8.0.3 Connect parancs

connect <szoba1><szoba2>

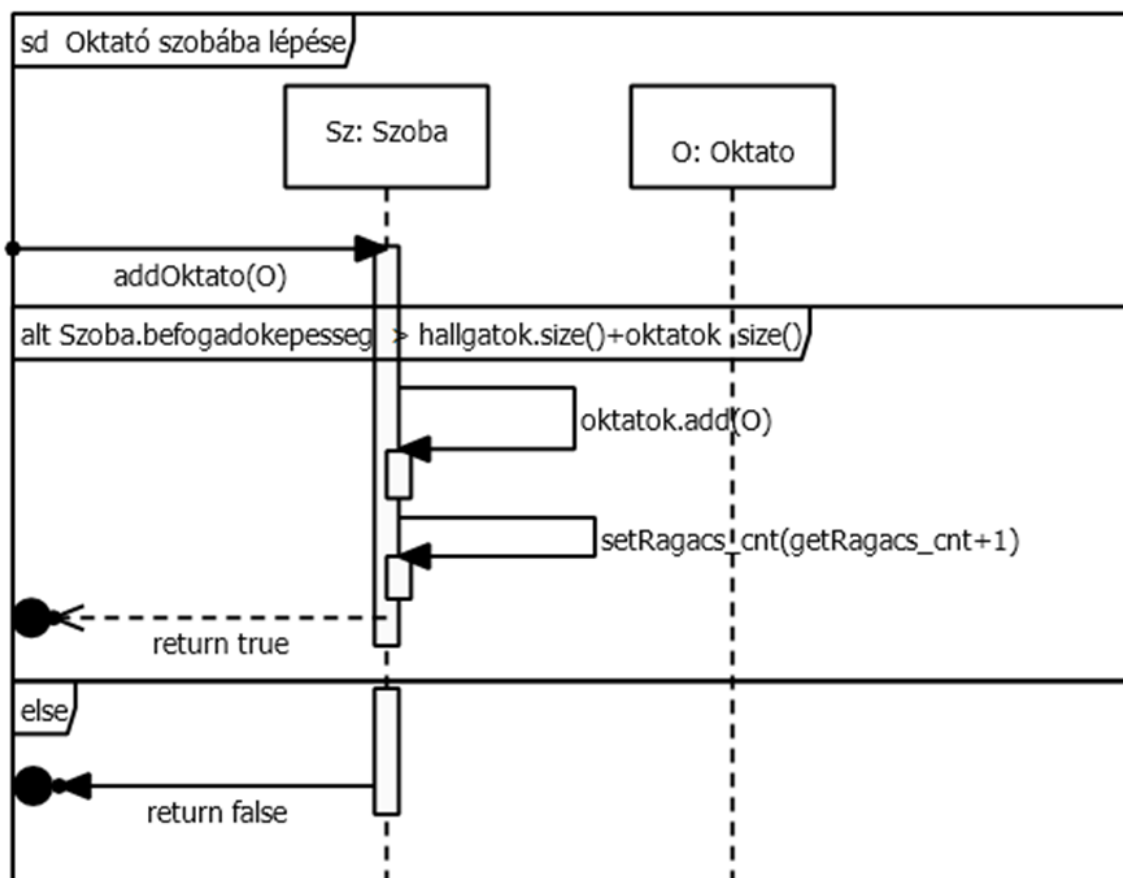
Leírás: A paraméterből kapott első szobából a második felé kerül egy ajtó

Opciók: -

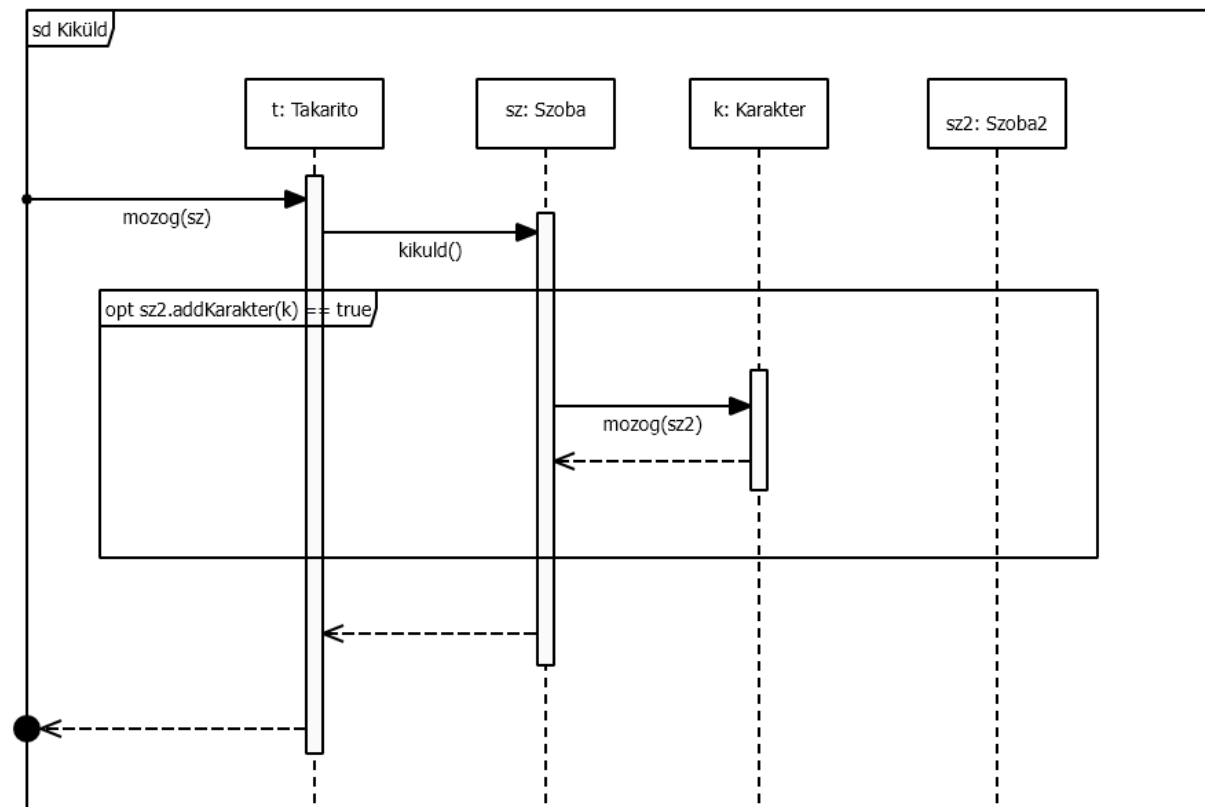
8.0.4 Add Hallgató



8.0.4 Add Oktató



8.0.5 Kiküld



A fenti példában a Karakter bármilyen játékbeli karakternek megfeleltethető, mivel mindegyikre ugyanúgy működik, illetve a kiinduló szobában tartózkodó összes karakterre értendő. Ugyanígy a Szoba2, a kiinduló szoba egy random szomszédját kívánja megjeleníteni, amelybe befér még az adott karakter. Ha nem fér be további szomszédot keresünk hasonló random módon

8.1 Osztályok és metódusok tervei.

8.1.1 Szoba

- **Felelősség:**

A játékot felépítő szobák megvalósításáért felelős ez az osztály. A ki-be lépő karaktereket, tárgyak elhelyezését és eltüntetését, illetve a szobák osztódását és összeolvadását valósítja meg.

- **Ősosztályok:**

-

- **Interfészek:**

-

- **Attribútumok:**

- # hallgatók: ArrayList<Hallgato>
Lista a szobában aktuálisan tartózkodó hallgatók számára.
- # oktatók: ArrayList<Oktato>
Lista a szobában aktuálisan tartózkodó oktatók számára.

- # takaritok: ArrayList<Takarito>
Lista a szobában aktuálisan tartózkodó takarítók számára.
- # ragacs_cnt: int
Számláló a szoba ragacsosságának nyilvántartására.
- # id: string
Minden szobához tartozik egy egyedi azonosító, hogy könnyen meg lehessen különböztetni két szobát egymástól.
- # palya: Palya
A szobákat tároló pályához férhetünk hozzá, ezen az attribútumon keresztül.
- #gazos : bool
A szoba gázosságát tartja számon.
- #befogadokepesseg: int
A szobába beférő karakterek számát tárolja.
- #regiszobak: ArrayList<Szoba>
Ha egyesült, az eredeti szobák adatait tartja nyilván.
- #szomszedok: ArrayList<Szoba>
A szoba szomszédjait tárolja.
- #targyak: ArrayList<Targy>
A szobában levő tárgyakat tartalmazza.

● **Metódusok:**

- + getHallgatok(): ArrayList<Hallgato>
Visszadaja a hallgatok attribútum értékét.
- + setHallgatok(ArrayList<Hallgato>): void
A paraméterként kapott listát állítja be a hallgatok attribútum értékének.
- + getOktatok(): ArrayList<Oktato>
Visszadaja az oktatok attribútum értékét.
- + setOktatok(ArrayList<Oktato>): void
A paraméterként kapott listát állítja be az oktatok attribútum értékének.
- + getTakaritok(): ArrayList<Takarito>
Visszaadaja a takaritok attribútum értékét.
- + setTakaritok(ArrayList<Takarito>): void
A paraméterként kapott listát állítja be a takaritok attribútum értékének.
- + getRagacs_cnt(): int
Visszaadaja a ragacs_cnt attribútum értékét.
- + setRagacs_cnt(int): void
A paraméterként kapott számot állítja be ragacs_cnt attribútum értékének.
- + getid(): string
Visszaadaja az id attribútum értékét.
- + setid(string): void
A paraméterként kapott stringet állítja be az id attribútum értékének.
- + removeHallgato(Hallgato): void
Abban az esetben, ha egy hallgató kilép a szobából, akkor hívódik meg. Ebben a metódusban az egyetlen utasítás az, hogy a hallgatok attribútumból eltávolítsuk a szobát elhagyó hallgatót.
- + removeOktato(Oktato): void

Abban az esetben, ha egy oktató kilép a szobából, akkor hívódik meg. Ebben a metódusban az egyetlen utasítás az, hogy az oktatók attribútumból eltávolítsuk a szobát elhagyó oktatót.

- + removeTakarito(Takarito): void

Abban az esetben, ha egy takarító kilép a szobából, akkor hívódik meg. Ebben a metódusban az egyetlen utasítás az, hogy a takarítók attribútumból eltávolítsuk a szobát elhagyó takarítót.

- + addHallgato(Hallgato): bool

A metódus feladata lekezelni egy hallgató szobába lépésének kísérletét.

Rövid pszeudokód:

Ha a hallgató befér a szobába:

hallgatók attribútumhoz hozzáadjuk az adott hallgatót

visszatérési érték igaz

egyébként:

visszatérési érték hamis

- + addOktato(Oktato): bool

A metódus feladata lekezelni egy oktató szobába lépésének kísérletét.

Rövid pszeudokód:

Ha az oktató befér a szobába:

oktatók attribútumhoz hozzáadjuk az adott oktatót

visszatérési érték igaz

egyébként:

visszatérési érték hamis

- + addTakarito(Takarito): bool

A metódus feladata lekezelni egy takarító szobába lépésének kísérletét.

Rövid pszeudokód:

Ha a takarító befér a szobába:

takarítók attribútumhoz hozzáadjuk az adott takarítót

visszatérési érték igaz

egyébként:

visszatérési érték hamis

- + deleteHallgato(Hallgato): void

A metódus feladata az, hogy abban az esetben, ha kibukik a hallgató a szobában, ne csak a szoba hallgatói nyilvántartásából, hanem az pálya hallgatókról vezetett listájából is törlődjön.

Rövid pszeudokód:

removeHallgato(kibukott hallgató) függvény meghívása

Kibukott hallgató eltávolítása a pálya hallgatókról vezetett listájából.

Annak ellenőrzése, hogy maradt-e még hallgató a pályán:

ha nem maradt, pályának jelezni, hogy fejezze be a játékot.

- -befér(): bool

A függvény lényege, hogy megvizsgálja, befér-e a karakter a szobába

Rövid pszeudokód:

Ha befogadokepesseg - (hallgatók lista mérete + oktatók lista mérete + takarítók lista mérete) \geq 1

visszatérési érték hamis

Egyébként

visszatérési érték igaz

- +isGazos(): bool

Visszaadja, hogy egy szoba gázos-e

- +setGaz(bool): void
A paraméterként kapott érték lesz a szoba gázossága
- +getBefogadokepesség(): int
Visszaadja a szoba befogadóképességét
- +setBefogadokepesség(int): void
A paraméterként kapott érték lesz a szoba befogadóképessége
- #getRegiszobak(): ArrayList<Szoba>
Visszaadja a szoba egyesülése előtti szobák állapotát, ha nem egyesült, akkor egy üres listát
- #setRegiszobak(ArrayList<Szoba>): void
A paraméterként kapott lista lesz a szoba egyesülés előtti szobák állapota
- +getSzomszedok(): ArrayList<Szoba>
Visszaadja a szoba szomszéd szobáit
- +setSzomszedok(ArrayList<Szoba>): void
A paraméterként kapott lista lesz a szoba szomszédjainak listája
- +addSzomszed(Szoba): void
A paraméterként kapott érték hozzáadódik a szoba szomszédjainak listájához
- +removeSzomszed(Szoba): void
A paraméterként kapott érték törlődik a szoba szomszédjainak listájából
- +getTargyak(): ArrayList<Targy>
Visszaadja a szobában található tárgyak listáját
- +setTargyak(ArrayList<Targy>): void
A paraméterként kapott lista lesz a szoba tárgyainak listája
- +targy_elhelyezese(Targy) : void
A paraméterként kapott tárgyat lehelyezi a szobába
Rövid pszeudokód:
targy szoba attribútuma ez a szoba lesz
szoba tárgyainak listájába adódik a tárgy
- +targy_eltuntetese(Targy) : void
A paraméterként kapott tárgy kikerül a szoba tárgyainak listájából
- +osztodik () : Szoba
A szoba szétosztódik a két eredeti szobájává
Rövid pszeudokód:
Ha a regiszobak lista nem üres
Létrejön szoba1 és szoba2, a regiszobak listában levőszobák tulajdonságaival
Az eredeti szobából kikerülnek a tárgyak és átkerülnek az eredeti szobába azoknak a regiszobak tárgyak listái szerint
Az eredeti szoba törlődik a szomszédjai listájából, és bekerülnek az új szobák
Az eredeti szoba törlődik a pályáról
Az új szobák a pályához adódnak
- +egyesul (Szoba) : void
A paraméterként kapott szobával egyesül a szobával
Rövid pszeudokód:
Létrejön az új egyesült szoba
új szoba regiszobak listájába adódik a paraméterből kapott szoba és ez a szoba
Ha valamelyik régi szoba gázos, vagy elátkozott volt, ez is azzá válik

Minden tárgyat elhelyez az új szobába a tárgy_elhelyezese függvényvel

A régi szobák minden szomszédját az új szobának a szomszédjai közé veszi, és a régi szobákat törli a szomszédok szomszédjai közül, és az új szobát beleteszi a szomszédjaik listájába

A régi szobákat kiveszi a pályáról

Az új szobát pedig felveszi a pályára

- +isRagacsos() : bool

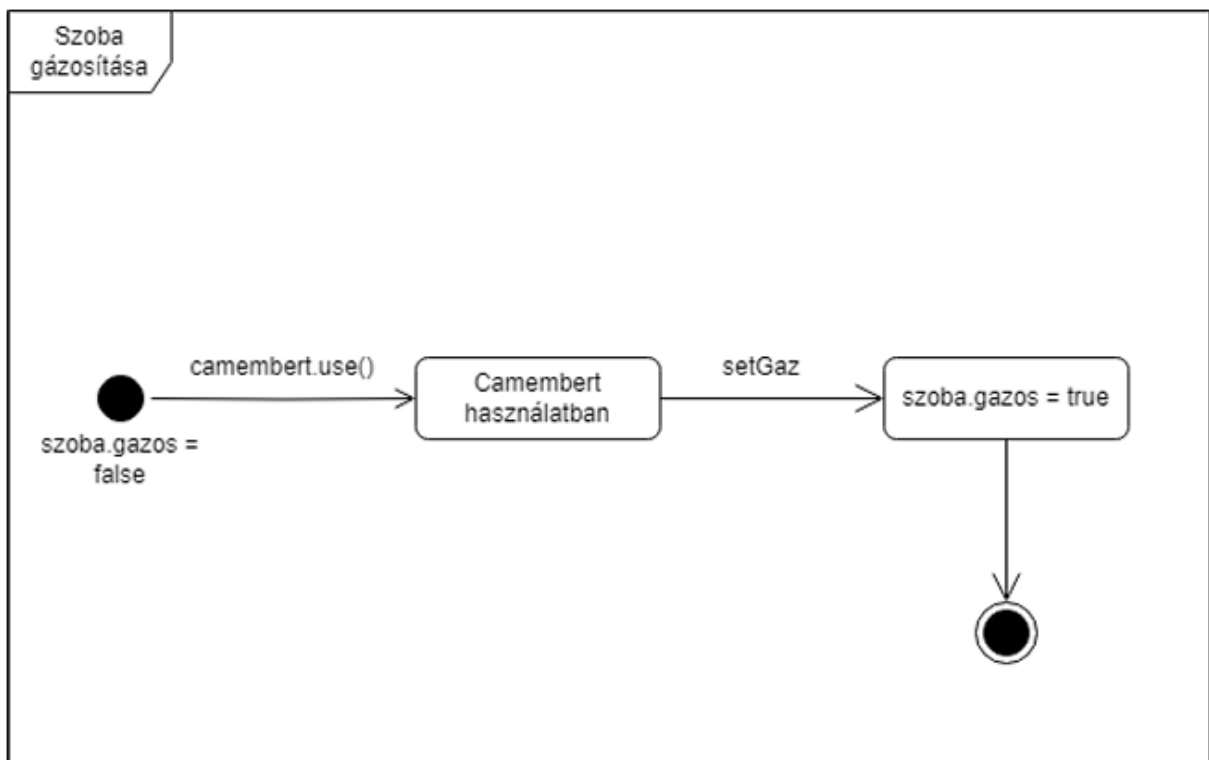
Visszaadja, hogy a szoba ragacsos-e

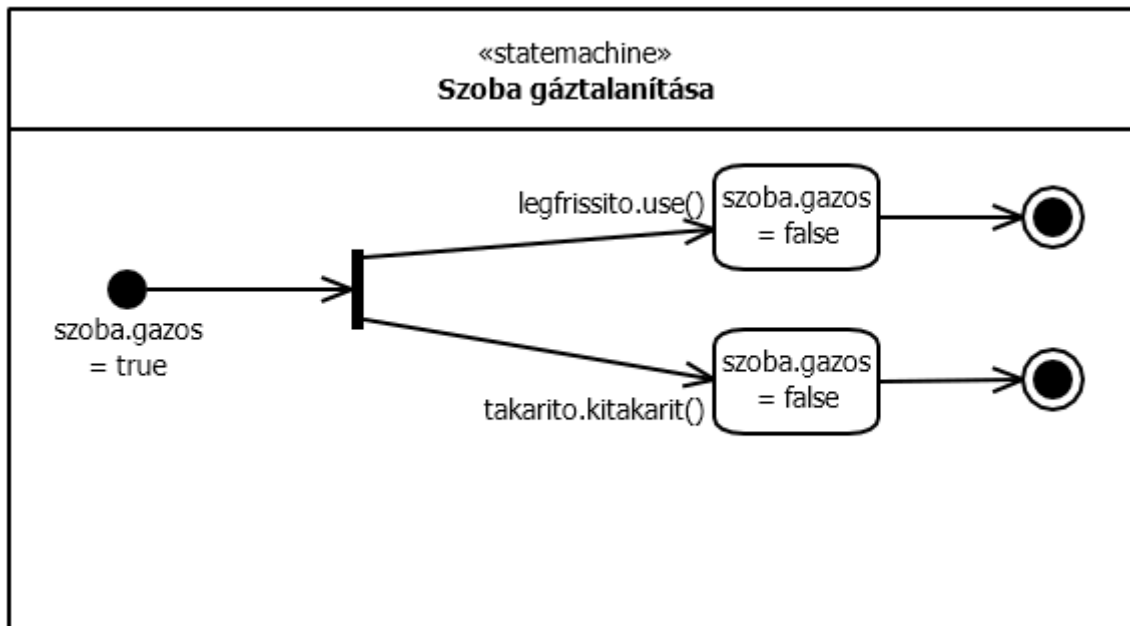
Rövid pszeudokód:

Ha a ragacsnt \geq mint a megadott ragacsossághoz kellő szám
visszatérési érték igaz

Egyébként

visszatérési érték hamis





8.1.2 Elátkozott_Szoba

- **Felelősség:**

Az elátkozott szobák megvalósításáért felelős ez az osztály. A szoba ajtajainak eltűnéséért és előtűnéséért felelős.

- **Össztályok:**

Szoba

- **Interfészek:**

-

- **Attribútumok:**

- - eltuntajto: ArrayList<Szoba>

Azon szobákat tartalmazza ez a lista, amelyekhez vezető ajtók eltűntek.

- **Metódusok:**

- # getEltuntajto(): ArrayList<Szoba>
Visszaadja az eltuntajto attribútum értékét.
- # setEltuntajto(ArrayList<Szoba>): void
A paraméterként kapott listát állítja be az eltuntajto paraméteréül.
- # addEltuntajto(Szoba): void
Hozzáadja a paraméterként kapott szobát az eltuntajto listához.
- # removeEltuntajto(Szoba): void
Eltávolítja az eltuntajto listából a paraméterként kapott szobát.
- + eltunik(Szoba): void
Az elátkozott szobának eltűnik a paraméterként kapott ajtaja

Rövid pszeudokód:

eltuntajto listába kerül a paraméterként kapott Szoba

A paraméterként kapott Szoba szomszédjai közül törlődik ez a szoba

Ennek a szomszédjai közül törlődik a paraméterként kapott Szoba

- + elotunik(Szoba): void

Az elétkozott szobának előtűnik egy paraméterként kapott eltűnt ajtaja

Rövid pszeudokód:

Ennek a szomszédjai közé adódik a paraméterként kapott Szoba

A paraméterként kapott Szoba szomszédjai közé adódik ez a szoba

eltuntajto listából törlődik a paraméterként kapott Szoba

8.1.3 Karakter

- **Felelősség:**

A játékban szereplő karakterekért felelős ez az osztály. A karakterek mozgását, tárgy felvételét és eszméletvesztését és a számontartja a karakter tárgyait, eszméletvesztettségét, azonosítóját.

- **Össztályok:**

-

- **Interfészek:**

-

- **Attribútumok:**

- # eszmeletvesztett : bool

Számontartja, hogy a karakter eszmeletvesztett-e.

- # szoba : Szoba

Számontartja, hogy a karakter, melyik szobába van éppen.

- # id : string

A karakter azonosítója, hogy meg lehessen különböztetni a többi karaktertől.

- # taska: ArrayList<Targy>

Lista a karakternél aktuálisan lévő tárgyak számára.

- **Metódusok:**

- + getEszmeletvesztett(): bool

Visszaadja, hogy a karakter eszmeletvesztett-e.

- + setEszmeletvesztett(bool): bool

Beállítja a paraméternek megfelelően az eszmeletvesztett értékét, hogy ha igaz akkor a karakter eszmeletvesztett, ha nem nem.

- + getSzoba(): Szoba

Visszaadja, hogy a karakter melyik szobába van.

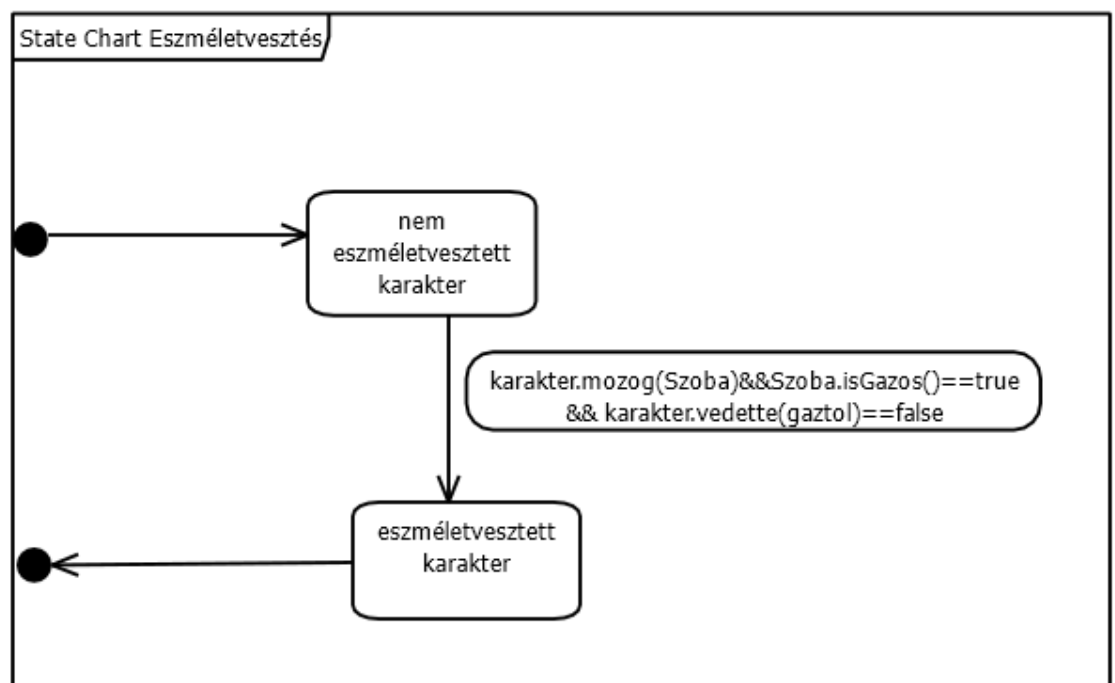
- + setSzoba(Szoba): void

Beállítja, hogy a karakter az attributumba megadott szobába van.

- + mozog (Szoba) : void

Az adott szobába helyezi a karaktert, ha befér és kiveszi az előző szobából, ahol volt, ez egy absztrakt függvény.

- + felvesz (Tárgy): void
A karakter az adott tárgyat felveszi a szobából és hozzáadja a táska listájához, ha lehetséges, ez egy absztrakt függvény.
- + eszméletvesztés(): void
Az eszméletességét a karakternek igazra állítja és az összes tárgyat, amit a karakter birtokol, elveszi a karaktertől és a szobába teszi.
Rövid pszeudokód:
meghívja setEszmeletvesztett(bool) függvényt igaz paraméterrel, majd végigmegy a táska listán és a tárgyakra meghívja setBirtokos(Karakter) függvényt és null értéket ad át neki, majd meghívja a getSzoba() szobán a tárgy_elhelyezese(Targy) függvényt a tárgyal, majd törli a táska listájából.
- + vedette(Vedettseg) : bool
Megnézi, hogy a karakter védett-e a paraméterben lévő dologtól, azzal, hogy megnézi a karakter tárgyait és ha kell használja és visszaadja, hogy védett.
Rövid pszeudokód:
Ha az attribútum Vedettseg értéke, az oktatotol van, akkor végig megy a karakter taska listáján és ha van olyan tárgya, aminek a funkcio értéke oktatot támad vagy oktatotol véd, akkor meghívja a tárgy use metódusát és visszatér igaz értékkel, ha nincs nála ilyen tárgy és azaz végigért a taska listán akkor visszatér hamis értékkel.
- + getTaska() : ArrayList<Targy>
Visszaadja a táska listát.



8.1.4 Oktato

- **Felelősség:**

A játékban szereplő oktatókért felelős ez az osztály. Az oktatók megbénulásának kezelését valósítja meg.

- **Ősosztályok:**

Karakter

- **Interfészek:**

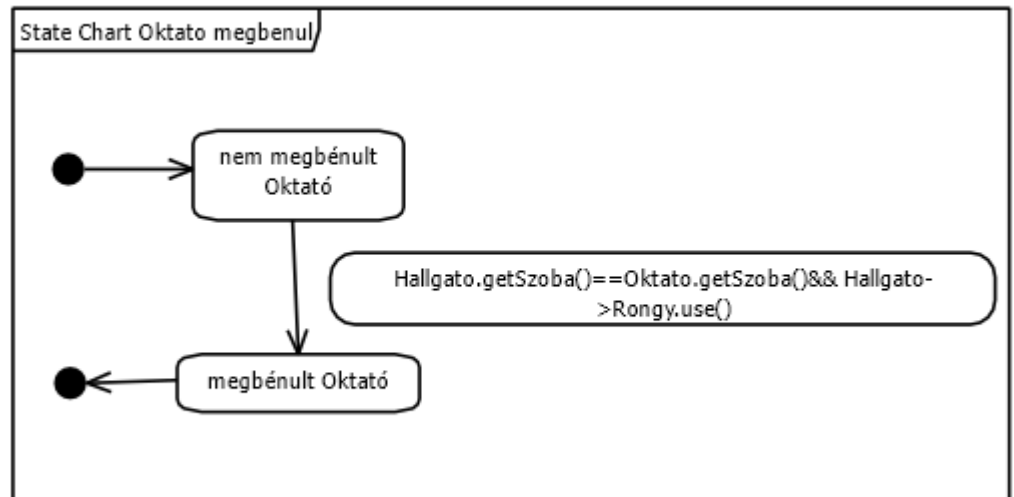
-

- **Attribútumok:**

- -megbenult: bool
Számontartja, hogy a karakter megbénult-e.

- **Metódusok:**

- + megbenul(): void
Az megbenult az oktatóknak igazra állítja.
- + mozog(Szoba) : void
Az adott szobába helyezi az oktatót, ha befér és kiveszi az előző szobából, ahol volt és a táskájában lévő tárgyakat is frissíti.
Rövid pszeudokód:
meghívja az attribútumba megadott szobán az addOktato(Oktato) függvényt, paraméterben megadva magát.
Ha az addOktato(Oktato) függvény visszatérési értéke igaz, akkor meghívja az eddigi getSzoba()-ra a removeOktato(Oktato) függvényt majd beállítja setSzoba(Szoba) a szobát és végigmegy a táska listáját és meghívja az összes tárgyra a setSzoba(Szoba) függvényt a attribútumba az attribútumba kapott szobával.
Ha az addOktato(Oktato) függvény visszatérési értéke hamis, akkor visszatér.
- + felvesz (Tárgy): void
Az oktató az adott tárgyat felveszi a szobából és hozzáadja a táska listájához.
Rövid pszeudokód:
Az oktató hozzáadja a táskájához a tárgyat, setBirtokos(Karakter) függvényt meghívja a tárgyon ahol magát adja paraméterben, majd meghívja a szobán a tárgy_eltuntetese(Targy) függvényt a tárgyal.



8.1.5 Hallgato

- **Felelősség:**

A játékban szereplő hallgatókért felelős ez az osztály. A hallgatók kibukását, tárgy eldobását, teleportálását valósítja meg.

- **Össztályok:**

Karakter

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

- + eldob (Tárgy): void

Attribútumba kapott tárgyat elhelyezi abba a szobába, ahol tartózkodik a hallgató és kiveszi a hallgatónak a tárgyai közül.

Rövid pseudokód:

A setBirtokos(Karakter) függvényt meghívja a tárgyon ahol nullt ad paraméterben, majd meghívja a szobán a targy_elhelyezese(Targy) függvényt a tárgyal, majd törli a tárgyat a táska listájából.

- + kibukik(): void

Kiveszi a hallgatót a szobák hallgatók listájából, ezzel véget ér a hallgatónak a játék.

Rövid pseudokód:

A getSoba() szobán meghívja a deleteHallgato(Hallgato) függvényt magával és végigmegy a táska listán és meghívja rajtuk a

setSzoba(Szoba) és a setBirtokos(Karakter) függvényeket null paraméterrel.

- + teleport(Tranzisztor) : void

Az attribútumba kapott tranzisztort használja a hallgató.

Rövid pszeudokód:

Meghívja a kapott Tranziszoron a use() függvényt.

- + mozog(Szoba) : void

Az adott szobába helyezi a hallgatót, ha befér és kiveszi az előző szobából, ahol volt és a táskájában lévő tárgyakat is frissíti.

Rövid pszeudokód:

meghívja az attribútumba megadott szobán az addHallgato(Hallgato) függvényt, paraméterben megadva magát.

Ha az addHallgato(Hallgato) függvény visszatérési értéke igaz, akkor meghívja az eddigi getSzoba()-ra a removeHallgato(Hallgato) függvényt majd beállítja setSzoba(Szoba) a szobát és végigmegy a táska listáját és meghívja az összes tárgyra a setSzoba(Szoba) függvényt a attribútumba az attribútumba kapott szobával.

Ha az addHallgato(Hallgato) függvény visszatérési értéke hamis, akkor visszatér.

- + felvesz (Tárgy): void

A hallgato az adott tárgyat felveszi a szobából és hozzáadja a táska listájához, ha lehetséges.

Rövid pszeudokód:

Ha a táska lista mérete 5, akkor visszatér, ha nem akkor a hallgató hozzáadja a táskájához a tárgyat, setBirtokos(Karakter) függvényt meghívja a tárgyon ahol magát adja paraméterben, majd meghívja a szobán a tárgy_eltüntetese(Targy) függvényt a tárgyal.

8.1.6 Takarító

- **Felelősség:**

A játékban szereplő takarítók felelős ez az osztály. A takarítók takarítását és a többi karakter szobából kiküldését valósítja meg.

- **Ősosztályok:**

Karakter

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

- + kikuld() : void

A többi szobába tartozkodó karaktert áthelyezi egy másik szobába.

Rövid pszeudokód:

Addig megy a getSzoba().getSzomszedok() listán, amíg nem üres a getSzoba() szoba getHallgatok és a getOktatok listája és minden szobára Végigmegy a getSzoba() szoba a getHallgatok majd a getOktatok listáján és mindegyik karakterre meghívja a mozog függvényt az adott szomszéd szobára. Majd ha már üres lesz a Szoba getHallgatok és a getOktatok listája vagy végigér a getSzoba().getSzomszedok() listán, akkor visszatér.

- + takarit() : void

A szoba ragacs számlálóját nullára állítja.

- + mozog(Szoba) : void

Az adott szobába helyezi a takarítót, ha befér és kiveszi az előző szobából, ahol volt és a táskájában lévő tárgyakat is frissíti.

Rövid pszeudokód:

meghívja az attributumba megadott szobán az addTakarito(Takarito) függvényt, paraméterben megadva magát.

Ha az addTakarito(Takarito) függvény visszatérési értéke igaz, akkor meghívja az eddigi getSzoba()-ra a removeTakarito(Takarito) függvényt majd beállítja setSzoba(Szoba) a szobát és végigmegy a táska listáját és meghívja az összes tárgyra a setSzoba(Szoba) függvényt a attributumba az attribútumba kapott szobával.

Ha az addTakarito(Takarito) függvény visszatérési értéke hamis, akkor visszatér.

- + felvesz (Tárgy): void

A takarító az adott tárgyat felveszi a szobából és hozzáadja a táska listájához, ha lehetséges.

A takarító hozzáadja a táskájához a tárgyat, setBirtokos(Karakter) függvényt meghívja a tárgyon ahol magát adja paraméterben, majd meghívja a szobán a targy_eltuntetese(Targy) függvényt a tárgyal.

8.1.7 Tárgy

- **Felelősség:**
Az osztály a játékban megtalálható tárgyak megvalósításáért felel. Kezeli továbbá minden tárgy sajátos funkcióját a használatuk közben.
- **Össztályok:**
-
- **Interfészek:**
-
- **Attribútumok:**
 - # funkcio: Funkcio
Az adott tárgy funkciójának megfelelő enumerációs attribútum. Egyedül azt dönti el pontosan, hogy milyen tárgyról is beszélünk.
 - # id: string
A tárgyakat egyértelműen meghatározó azonosító.
- **Metódusok:**
 - + setSzoba(Szoba): void
Beállítja az adott tárgyhoz tartozó szobát, amit paraméterként kap és ahol ezek után a tárgy megtalálható lesz.
 - + setBirtokos(Karakter): void
Beállítja az adott tárgyhoz tartozó gazda karaktert, amit paraméterként kap és akinél a tárgy megtalálható lesz, amíg valamilyen módon meg nem válik tőle.
 - + use(): void
Absztrakt függvény, amely a specifikus tárgyak esetében, azok használatáért felelős.
 - + getId(): String
Getter függvény az Id attribútumra.
 - + getFunkcio(): Funkcio
Getter függvény az Funkcio attribútumra.
 - + setId(String): void
Setter függvény az Id attribútumra.

8.1.8 Funkció enumeráció

- **Felelősség:**
Egy adott tárgy funkcióját egyértelműen meghatározó felsorolás.
- **Ősosztályok:**
-
- **Interfészek:**
-
- **Attribútumok:**
 - `oktatotol_ved`
A tárgy képes megvédeni a játékost egy oktató támadásai ellen.
 - `gazitol_ved`
A tárgy képes megvédeni a játékost egy gázos szoba hatása ellen.
 - `oktatot_tamad`
A tárgy képes a játékos birtokában támadó hatással lenni oktatók ellen.
 - `aktiv`
A tárgyat aktívan képes használni egy játékos, tehát meg lehet választani a tárgy felhasználásának körülményeit.
 - `hamis`
A tárgy nem használható semmire, csupán magához venni tudja a játékos.
 - `logarlec`
A tárgy megszerzését követően a játékosok győzelmével véget ér a játék.

8.1.9 Passzív Tárgy

- **Felelősség:**
Az osztály a játékban megtalálható passzív tárgyak megvalósításáért felel. A passzív tárgyak felhasználását nem a játékos vezérli, hanem valami specifikus eseményhez vannak kötve, vagy egy adott időn belül használódnak fel.
- **Ősosztályok:**
Tárgy
- **Interfészek:**
-
- **Attribútumok:**
 - `# töltet: int`
Egyes tárgyak felhasználásának száma egy bizonyos számhoz kötött. A töltet ennek a felhasználási számnak a felügyeletéért felel.
- **Metódusok:**

- + setToltet(): void
Beállítja, hogy az adott tárgy még mennyi alkalommal használható mielőtt nem tudja többet használni a játékos.
- + getToltet(): void
Getter függvény a Toltet attribútumra.

8.1.10 Aktív Tárgy

- **Felelősség:**
Az osztály a játékban megtalálható aktív tárgyak megvalósításáért felel. Az aktív tárgyak felhasználását a játékos közvetlenül vezérelheti, eldöntheti, hogy pontosan mikor és hogyan kívánja használni azokat.
- **Ősosztályok:**
Tárgy
- **Interfészek:**
-
- **Attribútumok:**
-
- **Metódusok:**
 - + use(): void
Absztrakt függvény, amely a specifikus tárgyak esetében, azok használatáért felelős.

8.1.11 Camembert

- **Felelősség:**
Az osztály a játékban megtalálható Camembert tárgy megvalósításáért felel. A tárgy használatakor a szoba, ahol a hallgató tartózkodik, gázzal töltött állapotba kerül.
- **Ősosztályok:**
Aktív Tárgy
- **Interfészek:**
-
- **Attribútumok:**
-
- **Metódusok:**
 - + use(): void
A függvény meghívása után a szoba, ahol a játékos tartózkodik, gázzal töltődik meg.

8.1.12 Léghfrissítő

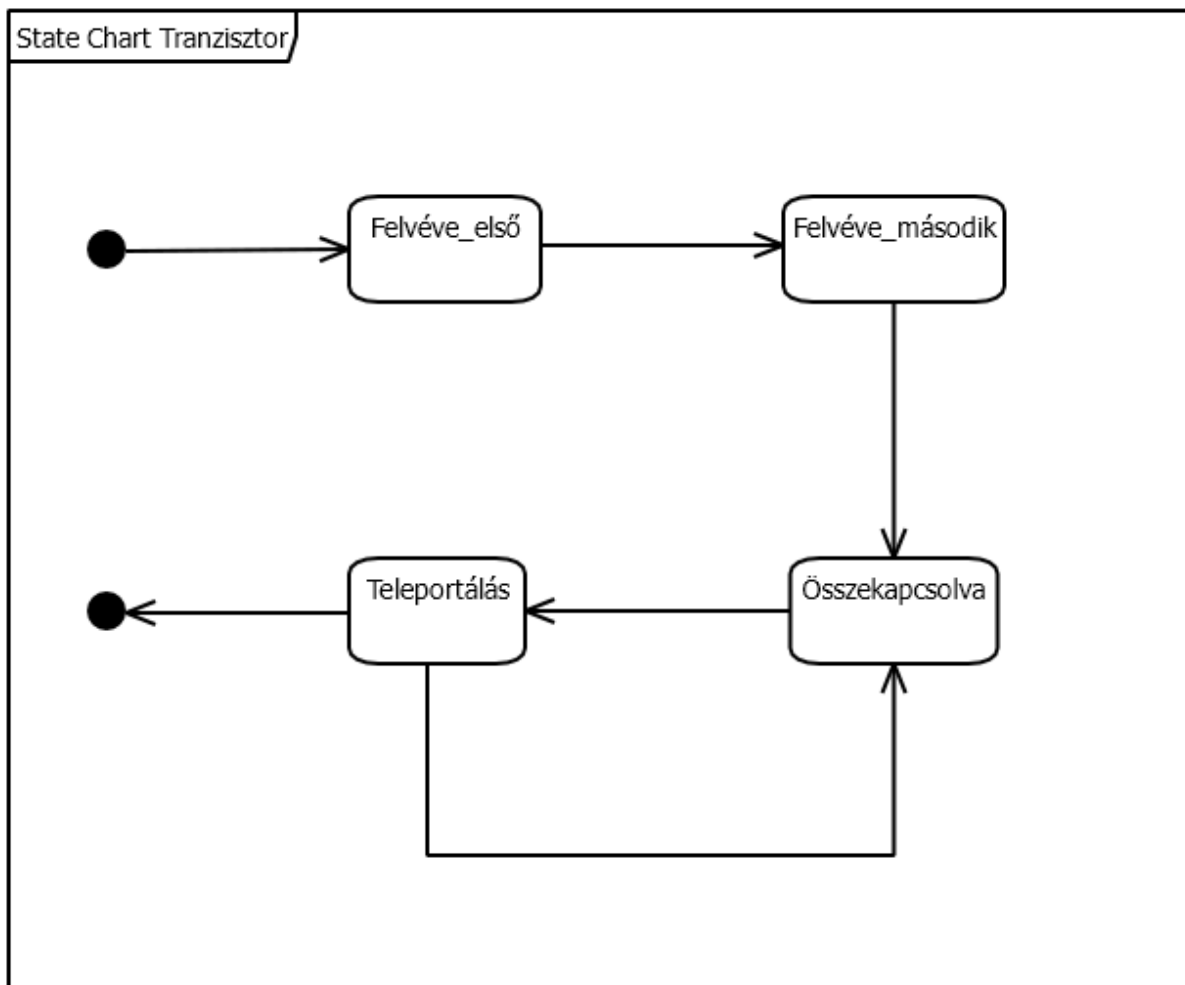
- **Felelősség:**
Az osztály a játékban megtalálható Léghfrissítő tárgy megvalósításáért felel. A tárgy használatakor, gázos szobában lerakva semlegesíti a gázhatást.
- **Össztályok:**
Aktív Tárgy
- **Interfészek:**
-
- **Attribútumok:**
-
- **Metódusok:**
 - + use(): void
A függvény meghívása után a gázos szobából, ahol a játékos tartózkodik, eltűnik a gáz.

8.1.13 Tranzisztor

- **Felelősség:**
Az osztály a játékban megtalálható Tranzisztor tárgy megvalósításáért felel. A tárgy lehetővé teszi a játékos számára, hogy a pályán két (előre meghatározott) helyszín között teleportálhasson. Mindezt két tranzisztor párosításával, majd elhelyezésükkel valósíthatja meg a felhasználó.
- **Össztályok:**
Aktív Tárgy
- **Interfészek:**
-
- **Attribútumok:**
 - – aktív: bool
Azt határozza meg, hogy a tranzisztor aktiválva van-e, vagy sem.
 - – hova: Szoba
Megadja, hogy a tranzisztor, teleportálást követően melyik szobában fogja elhelyezni a hallgatót.
 - – társ: Tranzisztor
Megadja, hogy az adott tranzisztor melyik másikkal van összekapcsolva a pályán.
- **Metódusok:**
 - + use(): void
A függvény meghívása után a játékos elteleportál abba a Szobába, ami az aktivált tranzisztor társát tartalmazza.

- + setTars(Tranzistor): void

A függvény beállítja, hogy az adott tranzistor melyik másikkal van összeköttetésben.



8.1.14 Rongy

- **Felelősség:**
Az osztály a játékban megtalálható Rongy tárgy megvalósításáért felel. A rongyot egy adott ideig tudja használni a játékos, ami idő alatt meg tudja a segítségével bénítani a vele egy szobában lévő oktatókat. Mivel ez a tárgy egy passzív tárgy, ezért, ha egy szobába kerül egy hallgató, akinél van egy rongy, egy oktatóval, akkor a tárgy hatása azonnal bekövetkezik.
- **Ősosztályok:**
Passzív Tárgy
- **Interfészek:**
-
- **Attribútumok:**
 - – timer: Timer
Meghatározza, hogy mennyi ideig használható meg a Rongy.
- **Metódusok:**
 - + tick(): void
Segít eldönteni, hogy pontosan meddig védi meg a játékost a Rongy az oktatók ellen. Amint a tárgyat magához veszi a hallgató, elindul egy Timer, ami idő alatt a tárgy használható. Ennek a Timer-nek a megfelelő nyomon követésében segít a függvény. Ha lejár az idő, akkor a tárgy eltűnik a hallgató inventory-ából.

8.1.15 Söröspohár

- **Felelősség:**
Az osztály a játékban megtalálható Söröspohár tárgy megvalósításáért felel. A söröspoharat egy adott ideig tudja használni a játékos, ami idő alatt meg tudja védeni a tárgy a szobában található oktatóktól. Mivel ez a tárgy egy passzív tárgy, ezért, ha egy szobába kerül egy hallgató, akinél van egy rongy, egy oktatóval, akkor a tárgy hatása azonnal bekövetkezik.
- **Ősosztályok:**
Passzív Tárgy
- **Interfészek:**
-
- **Attribútumok:**
 - – timer: Timer
Meghatározza, hogy mennyi ideig használható meg a Söröspohár.
- **Metódusok:**
 - + tick(): void

Segít eldönteni, hogy pontosan meddig védi meg a játékost a Söröspohár az oktatók ellen. Amint a tárgyat magához veszi a hallgató, elindul egy Timer, ami idő alatt a tárgy használható. Ennek a Timer-nek a megfelelő nyomon követésében segít a függvény. Ha lejár az idő, akkor a tárgy eltűnik a hallgató inventory-ából.

8.1.16 TVSZ

- **Felelősség:**

Az osztály a játékban megtalálható TVSZ tárgy megvalósításáért felel. A TVSZ-t három alkalommal tudja használni a játékos, ami idő alatt meg tudja védeni a tárgy a szobában található oktatóktól. Mivel ez a tárgy egy passzív tárgy, ezért, ha egy szobába kerül egy hallgató, akinél van egy rongy, egy oktatóval, akkor a tárgy hatása azonnal bekövetkezik.

- **Ősosztályok:**

Passzív Tárgy

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

-

8.1.17 Maszk

- **Felelősség:**

Az osztály a játékban megtalálható Maszk tárgy megvalósításáért felel. A Maszkot csak egy adott ideig tudja használni a játékos, ami idő alatt meg tudja védeni a tárgy egy esetleges gázos szoba hatásaitól. Mivel ez a tárgy egy passzív tárgy, ezért, ha a játékos belép egy gázos szobába, úgy azonnal felhasználásra kerül a Maszk, ha a hallgató birtokol egyet.

- **Ősosztályok:**

Passzív Tárgy

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

-

8.1.18 Pálya

- **Felelősség:**

A játéktér kezeléséért és a léptetésért felelős. Eltárolja a játékban részt vevő karaktereket, valamint, legenerálja a pályát, és lépteti a takarító, és oktató típusú karaktereket.

- **Ősosztályok:**

- **Interfészek:**

-

- **Attribútumok:**

- hallgatók : ArrayList<Hallgato>

- oktatók : ArrayList<Oktato>

- szobák : ArrayList<Szoba>

- takarítók: ArrayList<Takarito>

- **Metódusok:**

- + general() : void

Ciklus amíg nem jön START szó

hallgató hozzáadása

Ciklus hallgatók száma/3ig

oktató hozzáadása

Ciklus amíg nem értünk a fájl végére

sor eltárolása

sor szétszedése space mentén

első elem a szoba

2. elem a gazos attribútuma

3. elem a elatkozott szoba-e

4. elem a befogadokepesseg attribútuma

5-7. elem a tárgy

többi elem a szoba szomszédja

Takarító hozzáadása

Karakterek lerakása a pályára

- + leptet() : void

Ciklus az oktató lista végéig

Mozgás az oktatóval egy random szobába

Takarító mozgatása

Szobák osztása

Szobák összevonása

Elátkozott ajtók eltüntetése
 Elátkozott ajtók előtüntetése

8.1.19 Game

- **Felelősség:**
 Az idő múlásáért, illetve a játék kezdéséért és befejezéséért felel. Eltárol egy pályát és egy számlálót.
- **Ősosztályok:**
- **Interfészek:**
 -
- **Attribútumok:**
 - palya : Palya
 - szamlalo : int
 - logChange : LogarlecPropertyChangeListener
- **Metódusok:**
 - + start() : void
 időzítő indítása
 pálya generálása
 - + endgame() : void
 HA az időzítő eléri a maximumot AKKOR játék vége vesztesen
 HA az utolsó hallgató is kibukott AKKOR játék vége vesztesen
 HA a LogarlécPropertyChangeListener szólt AKKOR játék vége győztesen.

8.1.20 LogarlécPropertyChangeListener

- **Felelősség:**
 Azért felel, hogy figyelje mikor lesz felvéve a logarléc. Ha ez megtörténik akkor szól a pályának, hogy vége lehet a játéknak, a hallgató nyert.
- **Ősosztályok:**
- **Interfészek:**
 -
- **Attribútumok:**
 -
- **Metódusok:**
 - + propertyChange(PropertyChangeEvent e) : void
 meghívja az endgame() függvényt

8.1.20 Logarléc

- **Felelősség:**
A logarléc tárgy osztálya. Ennek a tárgynak a felvételével lehet megnyerni a játékot.
- **Ősosztályok:**
 - Tárgy
- **Interfészek:**
 -
- **Attribútumok:**
 -
- **Metódusok:**
 -

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 Teszteset1

- **Leírás**
Egy hallgató hozzáadása egy szobához.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Hallgató hozzáadása a szobához sikeresen megtörténik-e. Várható hibajegy: nem sikerül.
- **Bemenet**
create_room szoba1
add_character hallgato szoba1 h1
- **Kimenet**
Szoba hozzáadása sikeres.
Karakter sikeresen hozzáadva.

8.2.2 Teszteset2

- **Leírás**
Egy oktató hozzáadása egy szobához.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Oktató hozzáadása a szobához sikeresen megtörténik-e. Várható hibajegy: nem sikerül.
- **Bemenet**
create_room szoba1
add_character oktato szoba1 o1
- **Kimenet**
Szoba hozzáadása sikeres.
Karakter sikeresen hozzáadva.

8.2.3 Teszteset3

- **Leírás**

- Egy takarító hozzáadása egy szobához.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Takarító hozzáadása a szobához sikeresen megtörténik-e. Várható hibajegy: nem sikerül.
- **Bemenet**
create_room szoba1
add_character takarito szoba1 t1
- **Kimenet**
Szoba hozzáadása sikeres.
Karakter sikeresen hozzáadva.

8.2.4 Teszteset4

- **Leírás**
Hallgató mozgása egyik szobából a másikba.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Hallgató mozgása egy szobából a másikba sikeresen végbemegy-e. Várható hibajegy: nem fér be.
- **Bemenet**
create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character hallgato szoba1 h1
move h1 szoba2
- **Kimenet**
Szoba hozzáadása sikeres.
Szoba hozzáadása sikeres.
Szobák összekapcsolása sikeres.
Karakter sikeresen hozzáadva.
Karakter mozgása sikeres.

8.2.5 Teszteset5

- **Leírás**
Hallgató mozgása egy olyan szobába, ahol van oktató, de a hallgatónál van söröspohár, ami megvédi.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Hallgatót sikeresen megvédi-e a söröspohár. Várható hibajegy: nem fér be a szobába, nem védi meg, és kibukik.
- **Bemenet**
create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character hallgato szoba1 h1
add_character oktato szoba2 o1
spawn_item sorospohar szoba1 s1
pick h1 s1
move h1 szoba2
- **Kimenet**
Szoba hozzáadása sikeres.
Szoba hozzáadása sikeres.
Szobák összekapcsolása sikeres.

Karakter sikeresen hozzáadva.
Karakter sikeresen hozzáadva.
Tárgy hozzáadása sikeres.
Tárgy felvétele sikeres.
Karakter mozgása sikeres.

8.2.6 Teszteset6

- **Leírás**
Oktató mozgása egyik szobából a másikba.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Oktató mozgása egy szobából a másikba sikeresen végbemegy-e. Várható hibajegy: nem fér be.
- **Bemenet**
create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character oktato szoba1 o1
move o1 szoba2
- **Kimenet**
Szoba hozzáadása sikeres.
Szoba hozzáadása sikeres.
Szobák összekapcsolása sikeres.
Karakter sikeresen hozzáadva.
Karakter mozgása sikeres.

8.2.7 Teszteset7

- **Leírás**
Takarító mozgása egyik szobából a másikba.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Takarító mozgása egy szobából a másikba sikeresen végbemegy-e. Várható hibajegy: nem fér be.
- **Bemenet**
create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character takarito szoba1 t1
move t1 szoba2
- **Kimenet**
Szoba hozzáadása sikeres.
Szoba hozzáadása sikeres.
Szobák összekapcsolása sikeres.
Karakter sikeresen hozzáadva.
Karakter mozgása sikeres.

8.2.8 Teszteset8

- **Leírás**
Hallgató mozgása egy olyan szobába, ahol van oktató, de a hallgatónál van tvsz, ami megvédi.
- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hallgatót sikeresen megvédi-e a tvsz. Várható hibajegy: nem fér be a szobába, nem védi meg, és kibukik.

- **Bemenet**
 create_room szoba1
 create_room szoba2
 connect szoba1 szoba2
 add_character hallgato szoba1 h1
 add_character oktato szoba2 o1
 spawn_item tvsz szoba1 tvsz1
 pick h1 tvsz1
 move h1 szoba2
- **Kimenet**
 Szoba hozzáadása sikeres.
 Szoba hozzáadása sikeres.
 Szobák összekapcsolása sikeres.
 Karakter sikeresen hozzáadva.
 Karakter sikeresen hozzáadva.
 Tárgy hozzáadása sikeres.
 Tárgy felvétele sikeres.
 Karakter mozgása sikeres.

8.2.9 Teszteset9

- **Leírás**
 Hallgató mozgása egy olyan szobába, ami gázos, de a hallgatónál van maszk, ami megvédi, így nem veszti el az eszméletét.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
 Hallgatót sikeresen megvédi-e a maszk a gáztól. Várható hibajegy: nem fér be a szobába, nem védi meg, és eszméletetveszt.
- **Bemenet**
 create_room szoba1
 create_room gazosszoba1
 connect szoba1 gazosszoba1
 add_character hallgato szoba1 h1
 spawn_item maszk szoba1 m1
 pick h1 m1
 move h1 gazosszoba1
- **Kimenet**
 Szoba hozzáadása sikeres.
 Szoba hozzáadása sikeres.
 Szobák összekapcsolása sikeres.
 Karakter sikeresen hozzáadva.
 Tárgy hozzáadása sikeres.
 Tárgy felvétele sikeres.
 Karakter mozgása sikeres.

8.2.10 Teszteset10

- **Leírás**
 Oktató mozgása egy olyan szobába, ami gázos, de az oktatónál van maszk, ami megvédi, így nem veszti el az eszméletét.
- **Ellenőrzött funkcionalitás, várható hibahelyek**

Oktatót sikeresen megvédi-e a maszk a gáztól. Várható hibajegy: nem fér be a szobába, nem védi meg, és eszméletetveszt.

- **Bemenet**
 create_room szoba1
 create_room gazosszoba1
 connect szoba1 gazosszoba1
 add_character oktato szoba1 o1
 spawn_item maszk szoba1 m1
 pick o1 m1
 move o1 gazosszoba1
- **Kimenet**
 Szoba hozzáadása sikeres.
 Szoba hozzáadása sikeres.
 Szobák összekapcsolása sikeres.
 Karakter sikeresen hozzáadva.
 Tárgy hozzáadása sikeres.
 Tárgy felvétele sikeres.
 Karakter mozgása sikeres.

8.2.11 Teszteset11

- **Leírás**
 Hallgató mozgása egy olyan szobába, ami gázos, és nincs nála semmilyen tárgy, ami megvédi a gáztól, így eszméletetveszt.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
 Hallgató eszméletetveszt-e. Várható hibajegy: nem fér be a szobába, nem vesz eszméletet.
- **Bemenet**
 create_room szoba1
 create_room gazosszoba1
 connect szoba1 gazosszoba1
 add_character hallgato szoba1 h1
 move h1 gazosszoba1
- **Kimenet**
 Szoba hozzáadása sikeres.
 Szoba hozzáadása sikeres.
 Szobák összekapcsolása sikeres.
 Karakter sikeresen hozzáadva.
 Karakter mozgása sikeres.

8.2.12 Teszteset12

- **Leírás**
 Oktató mozgása egy olyan szobába, ami gázos, és nincs nála semmilyen tárgy, ami megvédi a gáztól, így eszméletetveszt.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
 Oktató eszméletetveszt-e. Várható hibajegy: nem fér be a szobába, nem vesz eszméletet.
- **Bemenet**
 create_room szoba1
 create_room gazosszoba1
 connect szoba1 gazosszoba1

add_character oktato szoba1 o1
move o1 gazosszoba1

- **Kimenet**

Szoba hozzáadása sikeres.
Szoba hozzáadása sikeres.
Szobák összekapcsolása sikeres.
Karakter sikeresen hozzáadva.
Karakter mozgása sikeres.

8.2.13 Teszteset13

- **Leírás**

Hallgató mozgása egy olyan szobába, ahol van oktató, de a hallgatónál nincs olyan tárgy, ami megvédené őt, ezért kibukik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hallgatót kibukott-e. Várható hibajegy: nem fér be a szobába, nem bukik ki.

- **Bemenet**

create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character hallgato szoba1 h1
add_character oktato szoba2 o1
move h1 szoba2

- **Kimenet**

Szoba hozzáadása sikeres.
Szoba hozzáadása sikeres.
Szobák összekapcsolása sikeres.
Karakter sikeresen hozzáadva.
Karakter sikeresen hozzáadva.
Karakter mozgása sikeres.

8.2.14 Teszteset14

- **Leírás**

Tárgyak elhelyezésének tesztje. Mivel minden tárgyra azonosan működik, és egy nem túl bonyolult funkció tesztje, ezért csak egy tárgyra tesztelünk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tárgy elhelyezése adott szobában sikeresen megtörténik-e. Várható hibajegy: nem sikerül elhelyezni.

- **Bemenet**

create_room szoba1
spawn_item maszk szoba1 m1

- **Kimenet**

Szoba hozzáadása sikeres.
Tárgy hozzáadása sikeres.

8.2.15 Teszteset15

- **Leírás**

A paraméterként megadott fájl tartalmát használva betölti az abban leírtakat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Valóban betöltődik-e a fájl tartalma, ahogy vártuk. Várható hibajegy: Nem sikerült a betöltés művelet.

- **Bemenet**
load file1
- **Kimenet**
Fájl sikeresen betöltve.

8.2.16 Teszteset16

- **Leírás**
A paraméterként megadott fájlba kimentjük a játék jelenlegi állapotát.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Valóban megtörténik-e a fájlba mentés a helyes adatokkal, ahogy vártuk. Várható hibajegy: Nem sikerült a mentés művelet.
- **Bemenet**
save file1
- **Kimenet**
Fájl sikeresen mentve.

8.2.17 Teszteset17

- **Leírás**
Két szoba egyesülése
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeresen egyesül-e a szoba, az egyesült szoba tulajdonságai megfelelőek lesznek-e
- **Bemenet**
create_room szoba1 2
create_room szoba2 5
create_room szoba3 1
connect szoba1 szoba3
connect szoba3 szoba1
spawn_item maszk szoba2 m1
merge szoba1 szoba2
- **Kimenet**
egyesules sikeres
befogadokepesség sikeres
szomszed sikeres
targy sikeres

8.2.18 Teszteset18

- **Leírás**
Egy gázos és egy nem gázos szoba egyesítése
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeresen egyesül-e a szoba, az egyesült szoba tulajdonságai megfelelőek lesznek-e
- **Bemenet**
create_room szoba1 2
create_room szoba2 gazos 5
merge szoba1 szoba2
- **Kimenet**
egyesules sikeres
gazos egyesules sikeres

8.2.19 Teszteset19

- **Leírás**
Egy elátkozott és egy nem elátkozott szoba egyesítése
- **Ellenőrzött funkcionalitás, várható hibahelyek**

Sikeresen egyesül-e a szoba, az egyesült szoba tulajdonságai megfelelőek lesznek-e

- **Bemenet**
create_room szoba1 2
create_room szoba2 elatkozott 5
merge szoba1 szoba2
- **Kimenet**
egyesules sikeres
elatkozott egyesules sikeres

8.2.20 Teszteset20

- **Leírás**
Egy elátkozott nem gázos és egy nem elátkozott gázos szoba egyesítése
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeresen egyesül-e a szoba, az egyesült szoba tulajdonságai megfelelőek lesznek-e
- **Bemenet**
create_room szoba1 gazos 2
create_room szoba2 elatkozott 5
merge szoba1 szoba2
- **Kimenet**
egyesules sikeres
elatkozott egyesules sikeres
gazos egyesules sikeres

8.2.21 Teszteset21

- **Leírás**
Két egyesült szoba megfelelően osztódik-e
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeresen osztódik-e a szoba, visszaállnak-e az egyesülés előtti állapotukba
- **Bemenet**
create_room szoba1 2
create_room szoba2 5
create_room szoba3 1
connect szoba1 szoba3
connect szoba3 szoba1
spawn_item maszk szoba2 m1
merge szoba1 szoba2
split szoba1
- **Kimenet**
osztodas sikeres
befogadokepesseg sikeres
szomszed sikeres
targy sikeres

8.2.22 Teszteset22

- **Leírás**
Egy gázos és egy nem gázos egyesült szoba megfelelően osztódik-e
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeresen osztódik-e a szoba, visszaállnak-e az egyesülés előtti állapotukba
- **Bemenet**
create_room szoba1 2
create_room szoba2 gazos 5
merge szoba1 szoba2

- split szoba1
- **Kimenet**
osztodas sikeres
gazos osztodas sikeres

8.2.23 Teszteset23

- **Leírás**
Egy elátkozott és egy nem elátkozott egyesült szoba megfelelően osztódik-e
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeresen osztódik-e a szoba, visszaállnak-e az egyesülés előtti állapotukba
- **Bemenet**
create_room szoba1 2
create_room szoba2 elatkozott 5
merge szoba1 szoba2
split szoba1
- **Kimenet**
osztodas sikeres
elatkozott osztodas sikeres

8.2.24 Teszteset24

- **Leírás**
Egy gázos nem elátkozott és egy elátkozott nem gázos egyesült szoba megfelelően osztódik-e
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeresen osztódik-e a szoba, visszaállnak-e az egyesülés előtti állapotukba
- **Bemenet**
create_room szoba1 gazos 2
create_room szoba2 elatkozott 5
merge szoba1 szoba2
split szoba1
- **Kimenet**
osztodas sikeres
elatkozott osztodas sikeres
gazos osztodas sikeres

8.2.25 Teszteset25

- **Leírás**
Egy korábban egyesült, de már visszaosztódott szobára meghívódik az osztódás
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ebben az esetben nem kell történnie semminek, hiszen csak olyan szoba osztódhat, amely egyesült már
- **Bemenet**
create_room szoba1 2
create_room szoba2 5
merge szoba1 szoba2
split szoba1
split szoba1
- **Kimenet**
első osztodas sikeres
második nem osztodas sikeres

8.2.26 Teszteset26

- **Leírás**

- Egy tárgyat felvesz egy hallgató.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A tárgy megjelenik-e a karakternél, akkor sikeres ha nem akkor a teszt megbukott.
- **Bemenet**
create_room szoba1
spawn_item söröspohar1
add_character hallgato1
pick hallgato1 söröspohar1
- **Kimenet**
A tárgyfelvétel sikeres.

8.2.26 Teszteset27

- **Leírás**
Egy tárgyat eldob egy hallgató.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A tárgy eltűnik-e a karaktertől, akkor sikeres ha nem akkor a teszt megbukott.
- **Bemenet**
create_room szoba1
spawn_item söröspohar1
add_character hallgato1
pick hallgato1 söröspohar 1
throw hallgato1 sorospohar1
- **Kimenet**
A tárgy sikeresen eltűnt.

8.2.26 Teszteset28

- **Leírás**
Egy tárgyat használ egy hallgató.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A tárgy use függvénye helyesen hívódik-e. Ha nem hívódik meg akkor sikertelen a teszt.
- **Bemenet**
create_room szoba1
spawn_item camembert1
add_character hallgato1
pick hallgato1 camembert1
use hallgato1 camembert1
- **Kimenet**
A tárgy használata sikeres.

8.3 A tesztelést támogató programok tervei

Minden tesztesethez tartozni fog egy bemeneti fájl, ahol a parancsok lesznek az adott teszt vizsgálatához, illetve egy kimeneti fájl ahol az elvárt eredmény lesz. A load paranccsal betöltött fájlból elindítjuk a tesztesetet és a kimenetet összehasonlítjuk a kimeneti fájljal. Ha egyezik akkor kiírja, hogy SUCCESS, ha nem akkor, hogy FAIL. Ezen kívül parancssorból is indíthatók tesztek. A tesztelésre külön programot nem használunk. Lesz egy testAll parancs ami az összes tesztet futtatja.

Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.04.10. 21:30	1 óra	Csapat	Értekezlet: Döntések: Javítások megbeszélése. Feladatok kiosztása.
2024.04.12. 9:00	3 óra	Czotter	8.2.1 - 8.2.14 tesztesetek megírása. Javítások 8.0.1 és 8.0.2 elkészítése. Szoba, ElátkozottSzoba osztályleírások egy részének elkészítése.
2024.04.14. 18:00	2 óra	Tarsoly	Szoba befejezése, és az osztódás és egyesülés tesztjeinek megírása.
2024.04.14. 21:00	1 óra	Csapat	Értekezlet: Döntések, kérdések megbeszélése, részletes tervek átbeszélése.
2024.04.14. 23:30	2 óra	Németh	Tárgyakkal kapcsolatos tervek.
2024.04.14. 23:00	2,5 óra	Schulcz	Javítás és Karakterekkel kapcsolatos osztályok és metódusok tervei, állapotgépek.
2024.04.15. 11	2 óra	Hermann	Game, Pálya és Logarléc osztályok megírása, valamint a Throw, Pick, Use parancsokhoz a tesztek megírásam dokumentum szerkesztése.