







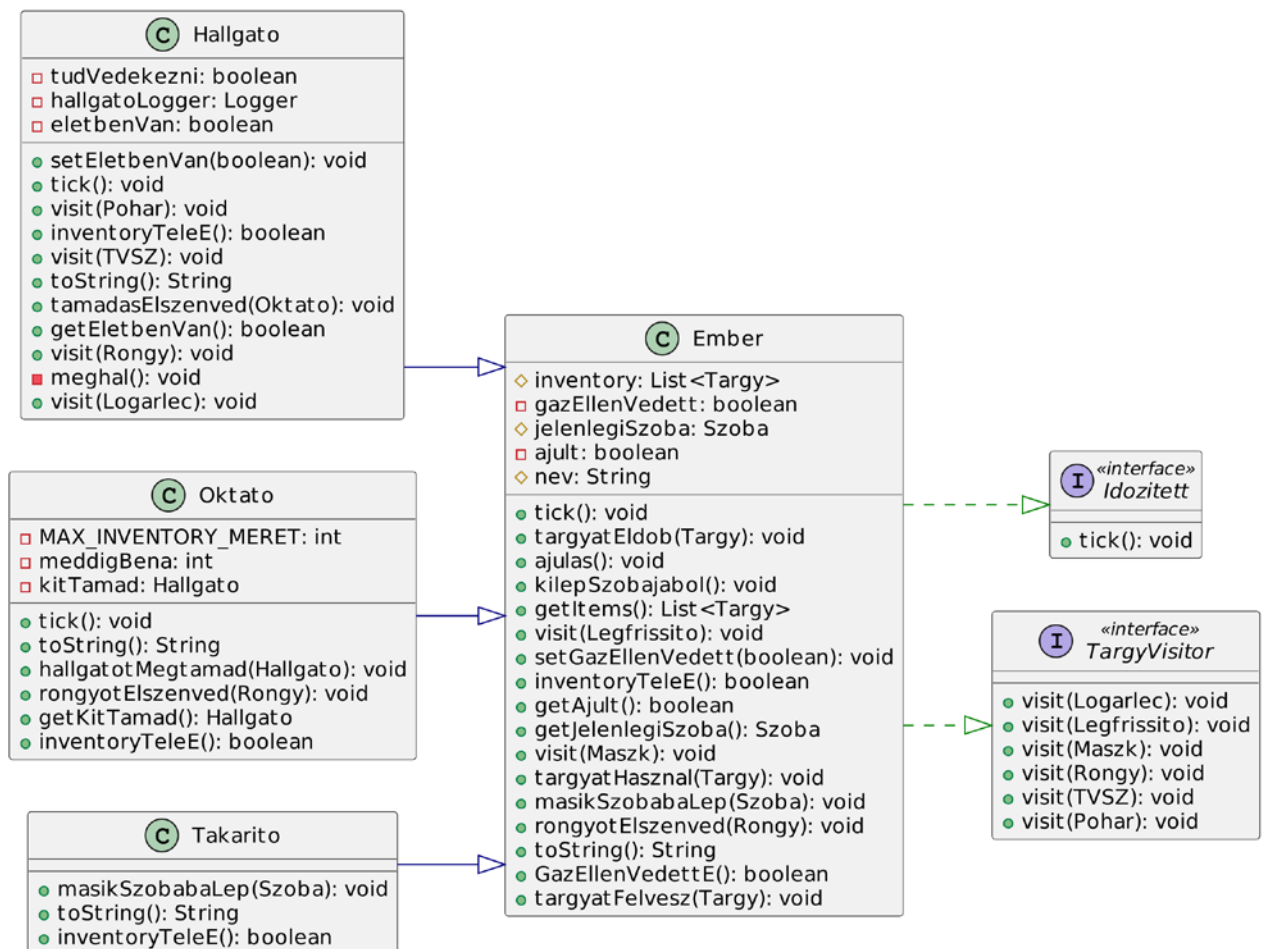
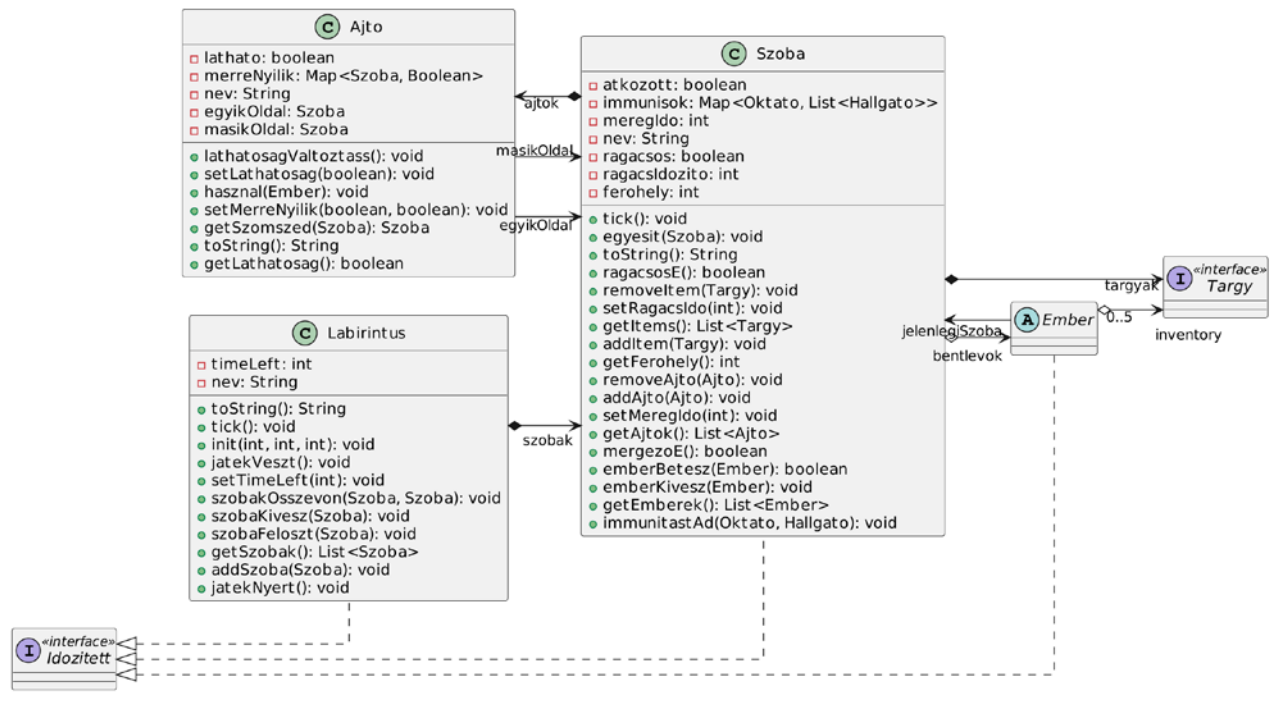
7. Prototípus koncepciója

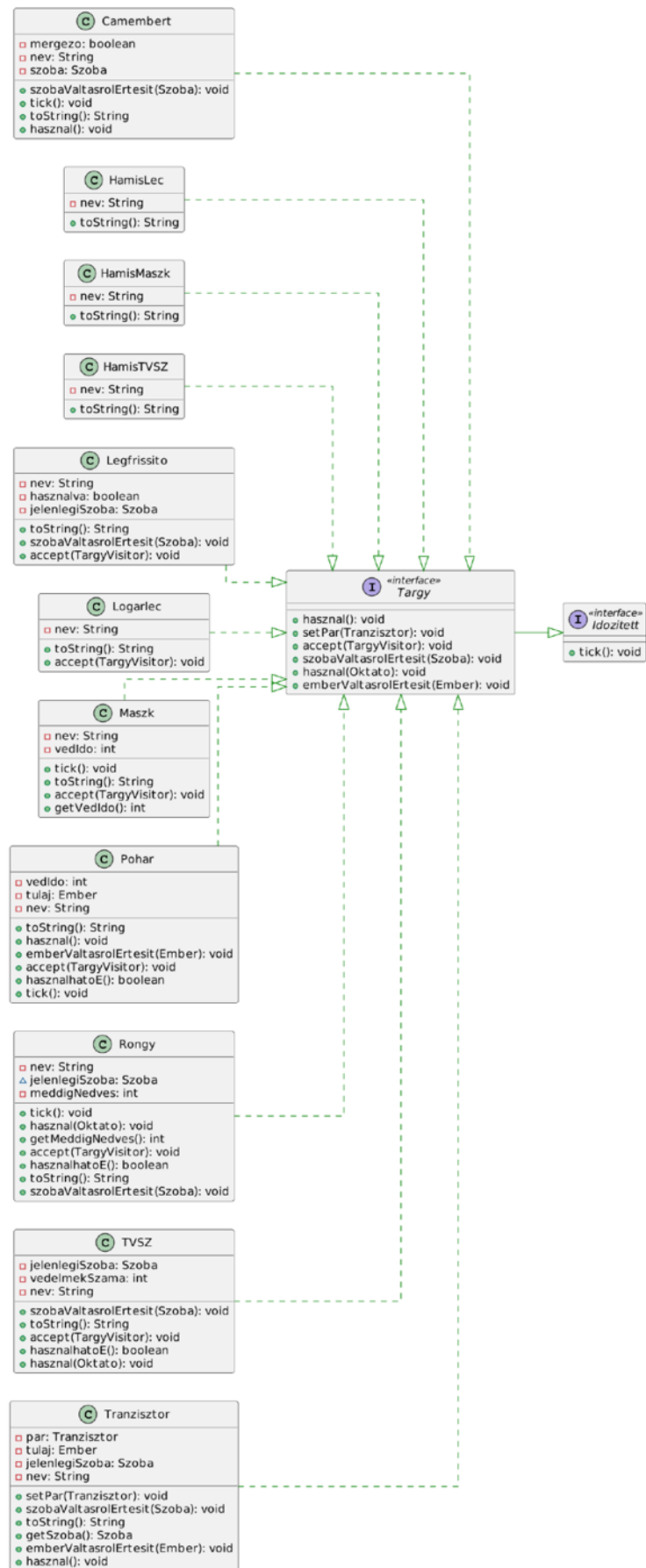
7.0 Változás hatása a modellre

7.0.1 Módosult osztálydiagram

Az interfaceknél default metódusok lettek bevezetve, az adott interfacet implementáló osztályoknál ezek csak akkor vannak jelölve, ha felüldefiniálják az adott metódust.

Jelmagyarázat		
Attribútum	Metódus	
		private
		protected
		public





7.0.2 Új vagy megváltozó metódusok

A változtatás miatti módosításokat a leírásban aláhúzással, az új metódusokat a nevük aláhúzásával, az újonnan létrehozott osztályokat pedig dőlt betűvel jelöltük. A toString() metódusokat nem részletezzük.

Ember:

- **void tárgyatFelvesz(Tárgy t):** Ez a metódus lehetővé teszi, hogy az ember felvegyen egy tárgyat a szobából, ha nem ájult, van hely az inventoryjában és nem ragacsos a szoba. Paraméterként megkapja a felvett tárgyat, ami átkerül az azt tartalmazó szobából az ember inventoryjába.
- **void visit(Légfrissítő l):** A metódus hatására az ember eldobja a légfrissítőt.

Légfrissítő:

- **void accept():** Fogadja a látogatót. Ha még nem volt használva és mérgező a szoba, akkor méregteleníti a szobát.
- **void szobaValtasrolErtesit(Szoba sz):** Átállítja a légfrissítő jelenlegi szobáját a paraméterként kapott szobára.

Pohár:

- **void hasznal():** A tárgyat használó ember eldobja valamelyik tárgyát.
- **void emberValtasrolErtesit(Ember e):** Átállítja a pohár tulajdonosát a paraméterként kapott emberre.

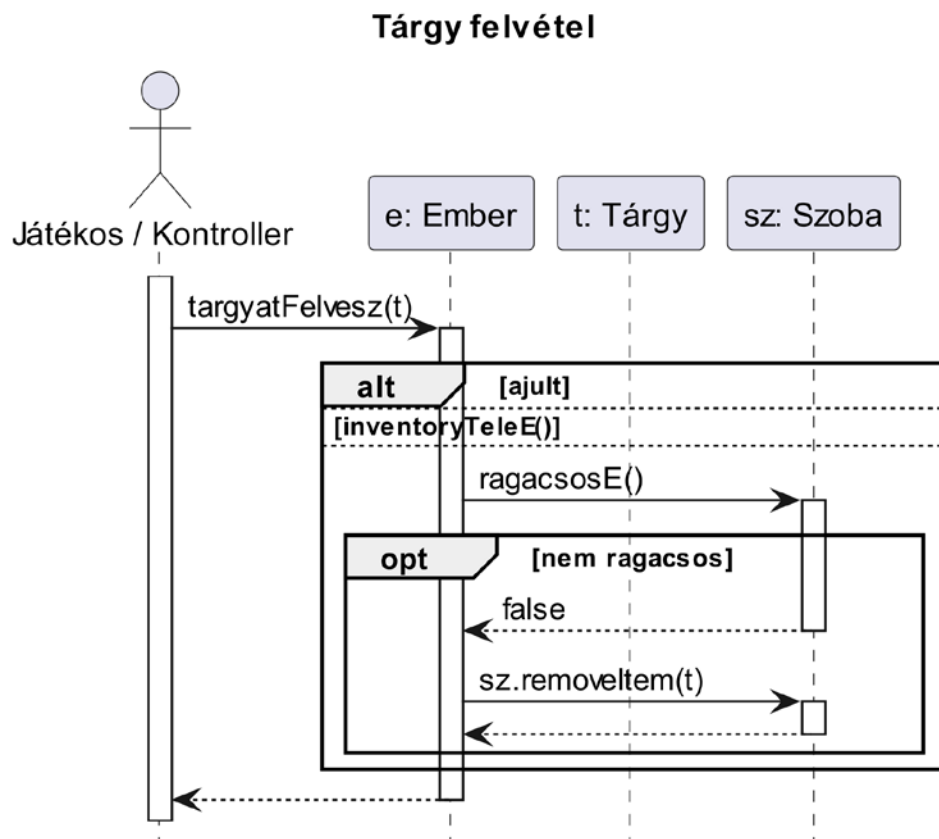
Szoba:

- **boolean ragacsosE():** visszaadja, hogy a szoba ragacsos e.
- **void tick():** Ha mérgező a szoba, akkor mindegyik bentlevő embert megpróbálja ájult állapotba hozni. Ha már volt takarítva a szobában (vagyis a ragacs időzítő nagyobb 0-nál) akkor egyet levon a ragacsossá válásig hátralevő időből. Ha lejár az időzítő, akkor a szoba ragacsossá válik.
- **void setRagacsIdo(int i):** Beállítja a ragacs időzítőt a paraméterül kapott egész számra.

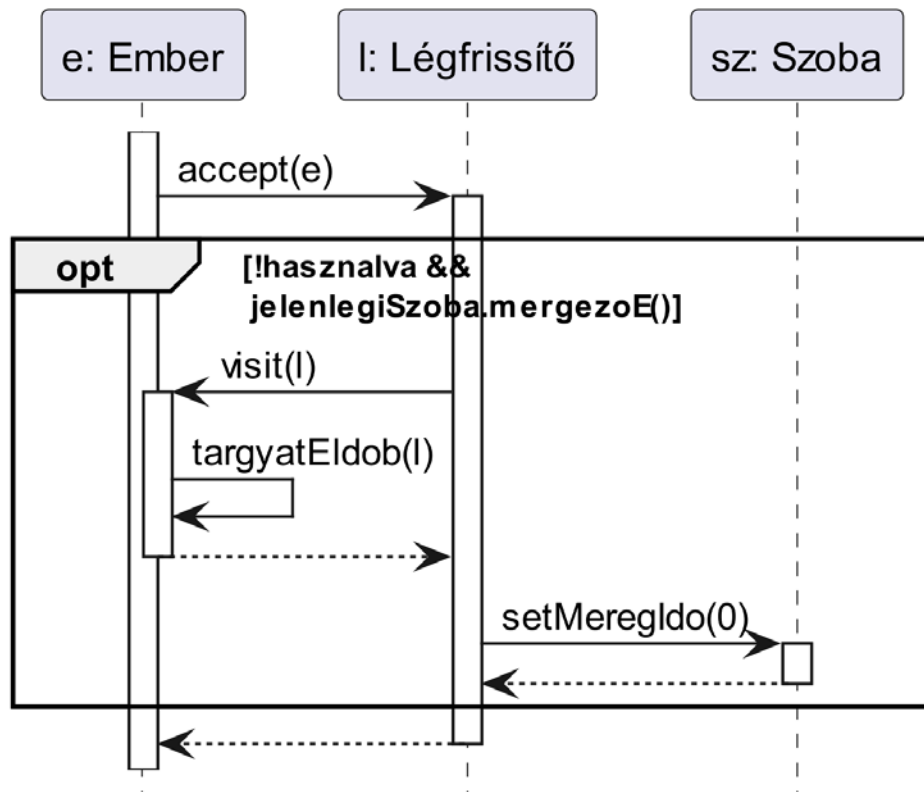
Takarító:

- **boolean inventoryTeleE():** Mindig igazat ad vissza, mivel a takarító nem vehet fel tárgyakat
- **void masikSzobabaLep(Szoba sz):** Ha a takarító nem fér be a paraméterként kapott szobába, akkor semmi nem történik. Egyébként bekerül a paraméterként kapott szobába és a jelenlegi szobája is erre módosul. Ezek után a bent levő embereket kitessekel a szobából, majd, ha mérgező a szoba akkor ki is szellőzteti azt, ami ennek hatására nem mérgez tovább.

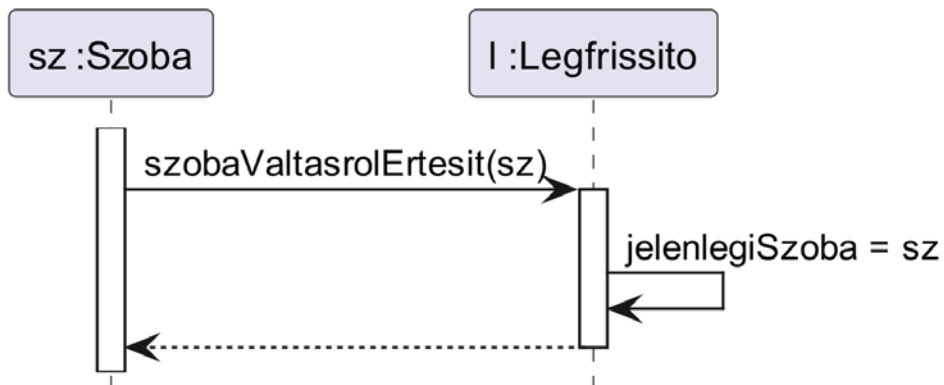
7.0.3 Szekvencia-diagramok

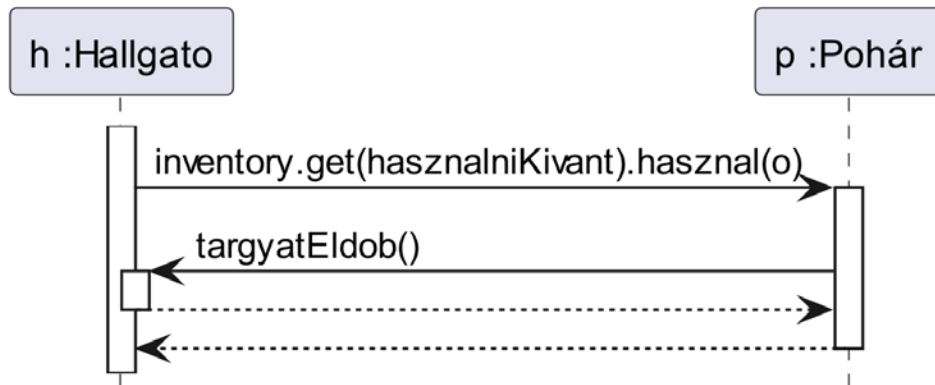
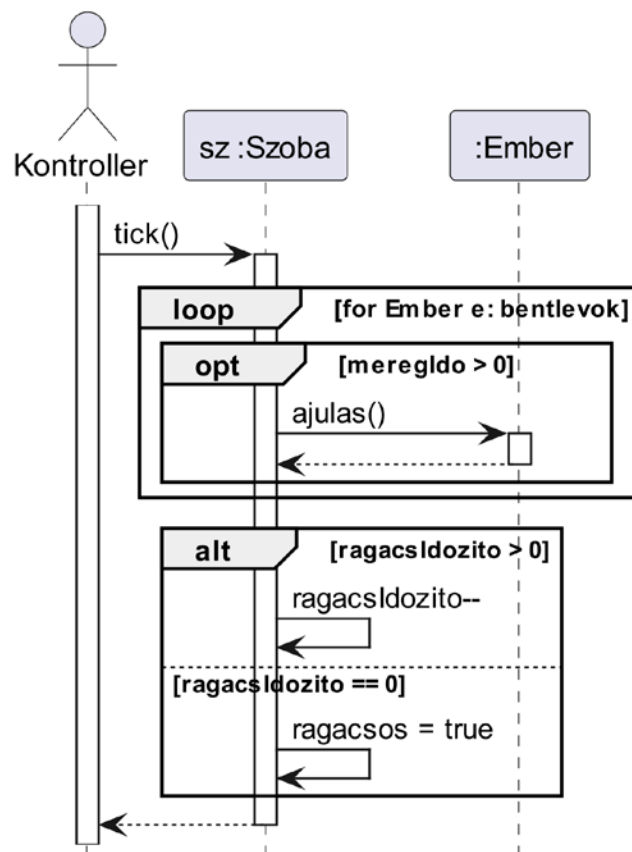


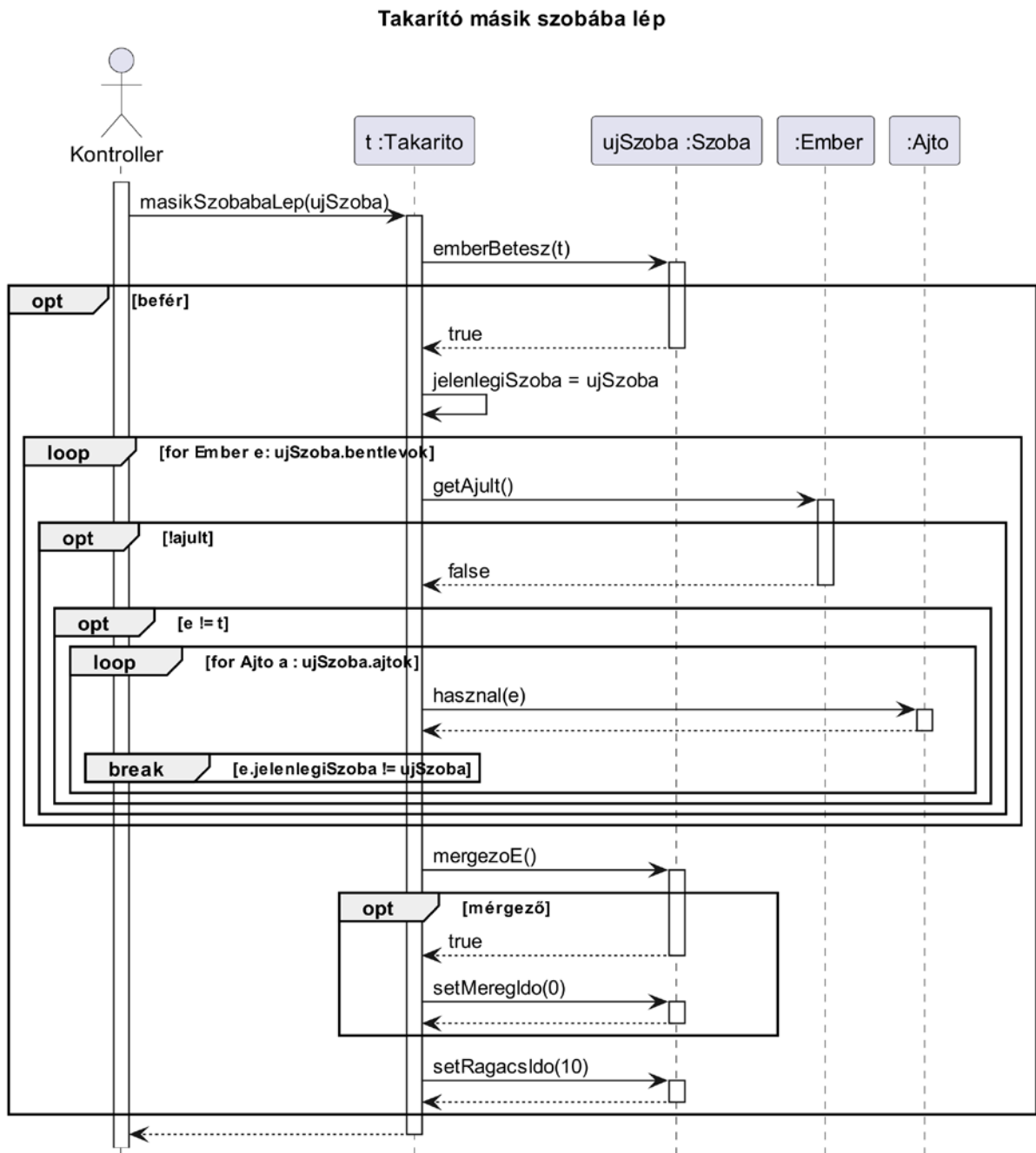
Embert a légfrissítő fogadja



Légfrissítő szobaváltásról értesül



Hallgató a Poharat használja védekezésre**Szoba ticket kap**



7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A prototípus karakteres felülete a szabványos bemenetről olvassa be a Felhasználó utasításait, illetve azok eredményét a szabványos kimenetre írja ki. A tesztelés segítésére, a pillanatnyi állapotot el lehet menteni fájlba, illetve onnan be is lehet olvasni korábbi állapotokat.

7.1.2 Bemeneti nyelv

Play <hallgatók db>

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

Opciók: Hallgatók (játékosok) számának 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>

Leírás: A hallgatónak 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 a Hallgató.

Targyfelvesz <Targy>

Leírás: A hallgatónak ezzel a paranccsal lesz lehetősége felvenni egy a szobában lévő tárgyat.

Opciók: Targy, a kívánt tárgy, amit a Hallgató fel szeretne venni.

Hasznal <Targy>

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

Opciók: Targy, a kívánt tárgy, amit a Hallgató használni szeretne.

Targyeldob<Targy>

Leírás: A hallgatónak ezzel a paranccsal lesz lehetősége eldobni egy a hátizsákjában lévő tárgyat.

Opciók: Targy, a kívánt tárgy, amit a Hallgató el szeretne dobni.

InfoHallgato<Hallgató>

Leírás: A Játékosnak ezzel a paranccsal lesz lehetősége a hallgató tárgyait lekérdezni.

Opciók: Hallgató, 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.

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".

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.
- A sor végén Enter van.

Példa:

- Szoba1: Szoba2 Szoba3

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

Oktatók leírása:

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

Példa:

- Szoba1: Szoba2 Szoba3
- 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
- Oktatók: Oktato1 Oktato2
- Tárgyak: Rongy Pohar

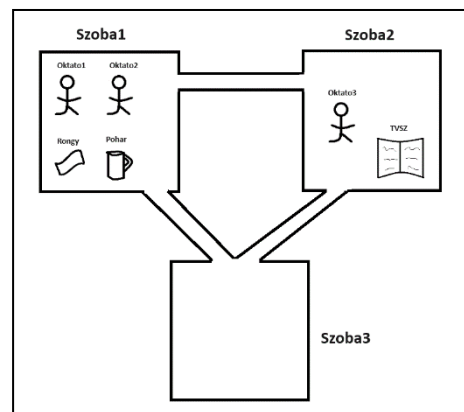
Itt a szobák kapcsolatán és a szobába tartózkodó Oktatókon felül látható, hogy a Szoba1-ben van egy rongy és egy Pohár.

Egy Labirintus teljes mentési példája:

Szoba1: Szoba2 Szoba3
Oktatók: Oktato1 Oktato2
Tárgyak: Rongy Pohar

Szoba2: Szoba1 Szoba3
Oktatók: Oktato3
Tárgyak: TVSZ

Szoba3: Szoba1 Szoba2
Oktatók:
Tárgyak:



Ahogy látható az egyes szobákat külön sor választja el egymástól.

7.1.3 Kimeneti nyelv

Fileba mentés:

A játék állása a <file név> lett elmentve.

InfoHallgató:

A hallgató tárgyai: < Tárgyak listája >

InfoSzoba:

A szoba állapota: <Szoba állapota>

A szobában lévő tárgyak: < Tárgyak listája >

A szobában lévő oktatók száma: <Oktatók db>

A szobából nyíló ajtók: <Ajtók listája>

A játékból hátralevő idő: <tick db> tick

Play:

A pálya legenerálva <hallgatók db> főre.

Játék elindult.

Szobaváltás:

Aleset1:

Átléptél a <Ajto> szobába.

Aleset2:

Az <Ajtó> ajtó egyirányú és nem tudsz rajta átmenni.

Aleset3:

A szomszéd szoba megtelt. Nem férsz be.

Tárgyfelvesz:

Aleset1:

A táskádba tetted a <Tárgy> tárgyat.

Aleset2:

A táskád teli van, nem fér bele a <Tárgy> tárgy.

Használ:

Aleset1:

A <Tárgy> tranzisztort párosítja a másik táskában levő tranzisztorral.

Aleset2:

A <Tárgy> tranzisztor párjához teleportálunk.

Aleset3:

A <Tárgy> pohár megvéd a támadások elől, és eldobod a <Tárgy> tárgyat.

Aleset4:

A <Camambert> elgázosítja a <Szoba> szobát.

Tárgyeldob:

Aleset1:

A táskádból a földre dobtad a <Tárgy> tárgyat.

Aleset2:

A táskád üres, nem dobhatsz tárgyat el.

Load:

A <Pályanév> játék betöltve.

Random:

Aleset1:

Mostantól determinisztikus a játék.

Aleset2:

Mostantól véletlenszerű a játék.

Reset:

A játék visszaállt a kiinduló állapotba.

7.2 Összes részletes use-case

Use-case neve	Pálya betöltése
Rövid leírás	A felhasználó lehetőséget kap a pálya betöltésére egy megadott fájlból, amely tartalmazza a szobákat, oktatókat és tárgyakat.
Aktorok	Felhasználó
Forgatókönyv	<ul style="list-style-type: none"> A felhasználó kiadja a Load <palyanev> parancsot. A rendszer betölti a megadott nevű fájlt, amely a pálya inicializálásához szükséges adatokat tartalmazza. A szobák, oktatók és tárgyak beolvasásra kerülnek a rendszerbe. A pálya sikeresen betöltődik és elindulhat a játék.

Use-case neve	Játék indítása
Rövid leírás	A felhasználó lehetőséget kap a játék elindítására a megadott hallgatók számával.
Aktorok	Felhasználó
Forgatókönyv	<ul style="list-style-type: none"> A felhasználó kiadja a Play <hallgatók db> parancsot, ahol <hallgatók db> a játékban résztvevő hallgatók számát jelöli. A rendszer inicializálja a pályát a megadott hallgatók számával. A játék elindul és megjeleníti a pályát a játékosok számának megfelelően.

Use-case neve	Ajtó használata
Rövid leírás	A felhasználó lehetőséget kap arra, hogy átmehessen egyik szobából a másikba, megadva az ajtót, amit használni szeretne.
Aktorok	Felhasználó
Forgatókönyv	<ul style="list-style-type: none"> A felhasználó kiadja az Ajtóhasznalat <Ajto> parancsot, ahol <Ajto> azonosítja azt az ajtót, amit használni szeretne. A rendszer ellenőrzi az ajtó használhatóságát és végrehajtja az átmenetet a megfelelő szobák között. A játékos új szobába kerül és folytathatja a játékot ott.

Use-case neve	Tárgy felvétele
Rövid leírás	A felhasználó lehetőséget kap arra, hogy felvegyen egy tárgyat a jelenlegi szobából.
Aktorok	Felhasználó
Forgatókönyv	<ul style="list-style-type: none"> • A felhasználó kiadja a Targyfelvesz <Targy> parancsot, ahol <Targy> azonosítja azt a tárgyat, amit fel szeretne venni. • A rendszer ellenőrzi, hogy a tárgy elérhető-e a jelenlegi szobában, és ha igen, akkor azt a játékos hátizsákjába helyezi (ha van elég hely). • A felhasználó folytathatja a játékot a kiválasztott tárggyal a birtokában.

Use-case neve	Tárgy használata
Rövid leírás	A felhasználó lehetőséget kap arra, hogy használja a birtokában lévő tárgyat. Amennyiben a tárgy használata automatizált nem történik semmi .
Aktorok	Felhasználó
Forgatókönyv	<ul style="list-style-type: none"> • A felhasználó kiadja a Hasznal <Targy> parancsot, ahol <Targy> azonosítja azt a tárgyat, amit használni szeretne. • A rendszer ellenőrzi, hogy a tárgy a felhasználó hátizsákjában van-e. • Ha a tárgy felhasználható (pl. Camembert, Pohár vagy Tranzisztor), akkor a rendszer végrehajtja a megfelelő műveletet. • A felhasználó folytathatja a játékot a használt tárgy hatására bekövetkező változásokkal.

Use-case neve	Tárgy eldobása
Rövid leírás	A felhasználó lehetőséget kap arra, hogy eldobhassa a birtokában lévő tárgyat (vagy párosított tranzisztornál letegye).
Aktorok	Felhasználó
Forgatókönyv	<ul style="list-style-type: none"> • A felhasználó kiadja a Targyeldob <Targy> parancsot, ahol <Targy> azonosítja azt a tárgyat, amit el szeretne dobni. • A rendszer ellenőrzi, hogy a tárgy a felhasználó hátizsákjában van-e. • Ha a tárgy megtalálható a hátizsákban, akkor azt eltávolítja. • A felhasználó folytathatja a játékot a tárgy eldobását követően és a Szobában, ahol eldobta a tárgyat ott megjelenik az eldobott tárgy.

Use-case neve	Info a Hallgatóról
Rövid leírás	A játékos meg tudja tekinteni a megadott hallgató tárgyait/tulajdonságait ezzel a paranccsal.
Aktorok	Játékos
Forgatókönyv	<ul style="list-style-type: none"> A játékos kiadja az InfoHallgató<Hallgató> parancsot, ahol <Hallgató> azonosítja azt a hallgatót, akinek a tárgyait meg szeretné tekinteni. A rendszer megjeleníti a megadott hallgató nevét és a hozzá tartozó tárgyak listáját és esetleges tulajdonságait.

Use-case neve	Info a Szobáról
Rövid leírás	A játékos meg tudja tekinteni a megadott szoba adatait ezzel a paranccsal.
Aktorok	Játékos
Forgatókönyv	<ul style="list-style-type: none"> A játékos kiadja az InfoSzoba<Szoba> parancsot, ahol <Szoba> azonosítja azt a szobát, amelynek adatait meg szeretné tekinteni. A rendszer megjeleníti a megadott szoba nevét, a szomszédos szobákat, az ott található oktatók hallgatók és tárgyak listáját.

Use-case neve	Random
Rövid leírás	A játékos beállítja a tesztelés módját, hogy véletlenszerűséget vagy determinisztikus lefutást használjon.
Aktorok	Játékos
Forgatókönyv	<ul style="list-style-type: none"> A játékos kiadja a random <bool> parancsot, ahol <bool> egy logikai érték, ami engedélyezi vagy letiltja a véletlenszerűséget. Ha <bool> értéke true, akkor a rendszer véletlenszerű funkciókat fog használni. Ha <bool> értéke false, akkor a rendszer determinisztikus módban fog futni.

Use-case neve	Reset
Rövid leírás	A játékos az eredeti állapotba hozza a programot.
Aktorok	Játékos
Forgatókönyv	<ul style="list-style-type: none"> A játékos kiadja a reset parancsot. A program visszaállítja magát az eredeti állapotba. Minden játékbeli változó, állapot, és konfiguráció visszaáll az alapértelmezett értékeire vagy állapotára.

Use-case neve	tick
Rövid leírás	Az időt lépteti a megadott számú lépéssel.
Aktorok	Játékos
Forgatókönyv	<ul style="list-style-type: none"> A játékos kiadja a tick parancsot a megadott számú léptetéssel. A program végrehajtja az idő léptetését a megadott léptetésszám alapján.

Use-case neve	Save
Rövid leírás	A játék ezzel a paranccsal lehetőséget biztosít a labirintus aktuális állapotának mentésére egy fájlba. Ez a mentés később újra betölthető lesz.
Aktorok	Játékos
Forgatókönyv	<ul style="list-style-type: none"> • A játékos kiadja a "Save" parancsot. • A rendszer elkéri a felhasználótól a fájl nevét, ahova menteni szeretné az állapotot. • A játékos megadja a fájl nevét. • A rendszer elmenti a labirintus aktuális állapotát a megadott fájlba. • A rendszer visszaigazolja a mentés sikerességét.

7.3 Tesztelési terv

Régiék:

Teszt-eset neve	Ember felvesz egy tárgyat
Rövid leírás	Egy hallgató vagy oktató felvesz egy tárgyat az inventory-ba.
Teszt célja	A tárgy az ember inventory-ba kerül, amennyiben van elég helye.

Teszt-eset neve	Ember eldob egy tárgyat
Rövid leírás	Egy hallgató vagy oktató eldob egy tárgyat az inventory-ból.
Teszt célja	Egy tárgy egy ember inventory-jából a szobába kerül.

Teszt-eset neve	Oktató támad
Rövid leírás	Az oktató megtámadja a hallgatót, ha az egy szobába kerül vele, valamilyen módon
Teszt célja	A hallgatónak vagy védekeznie kell valahogyan, vagy meghal

Teszt-eset neve	Ember gázos szoba interakció
Rövid leírás	Egy ember egy elgázosított szobába lép.
Teszt célja	Maszk hiányában az ember elájul és eldobja a tárgyait.

Teszt-eset neve	A hallgató teleport
Rövid leírás	A hallgató párosít 2 tranzisztort majd eldobja az egyiket egy szobában a másikkal pedig visszatérhet abba a szobába
Teszt célja	Az elvárt működése a tranzisztornak

Teszt-eset neve	Szoba osztódás
Rövid leírás	Egy szoba kettéosztódik.

Teszt célja	A szoba ajtóit meg a benne lévő emberek, tárgyak eloszlanak a két új szoba között.
--------------------	--

Teszt-eset neve	Szoba összevonás
Rövid leírás	A játék során 2 szoba összevonódik
Teszt célja	A szobákban lévő tárgyak, emberek, ajtók a meghatározott módon át kelljen kerüljenek az új szobába

Teszt-eset neve	Szoba elgázosítása
Rövid leírás	Egy szobában camambert használ egy hallgató.
Teszt célja	Egy eddig gázmentes szobából, gázos szoba lett, és a camambert fel lett használva.

Teszt-eset neve	Ember belép egy szobába
Rövid leírás	Egy ember az eddigi szobájából átlép egy új szobába.
Teszt célja	Amennyiben az új szobában még van hely az ember átkerül az új szobába.

Teszt-eset neve	Ajtó eltűnik/megjelenik
Rövid leírás	Az ajtó megjelenik vagy eltűnik
Teszt célja	Ha az ajtó el van tűnve nem lehet átmenni rajta

Teszt-eset neve	Ember ajtó interakció
Rövid leírás	Egy ember egy ajtót használ.
Teszt célja	Ha az ajtó nyílik ebből az irányból és létható, akkor az ember használja

Teszt-eset neve	Init
Rövid leírás	A labirintus létrehozása és feltöltése oktatókkal, tárgyakkal, szobák adatainak beállítása stb.
Teszt célja	A pálya létrehozása és a játék indítása.

Teszt-eset neve	Lejárt az idő
Rövid leírás	Ha lejár az idő vége a játéknak
Teszt célja	Vége kell legyen a játéknak

Újak:

Teszt-eset neve	Hamis tárgyfelvétel
Rövid leírás	Egy ember egy hamis tárgyat vesz fel az inventoryba.
Teszt célja	A hamis tárgy az ember inventoryjába kerül amennyiben van elég helye.

Teszt-eset neve	Hamis tárgyhasználat
Rövid leírás	Egy ember egy hamis tárgyat próbál használni.
Teszt célja	A tárgynak a sajátos tulajdonsága nem jön elő, nem történik semmi.

Teszt-eset neve	Hamis tárgydobás
Rövid leírás	Egy hallgató vagy oktató eldob egy hamis tárgyat az inventory-ból.
Teszt célja	Egy tárgy egy ember inventoryjából a szobába kerül.

Teszt-eset neve	Légfrissítő használata
Rövid leírás	A légfrissítő használata egy gázos szobában.
Teszt célja	A szoba állapota megváltozik, nem lesz többé gázos.

Teszt-eset neve	Ragacsos tárgy felvétel
Rövid leírás	Egy ember megpróbál egy ragacsos szobában felvenni egy tárgyat.
Teszt célja	A tárgyat nem sikerült felvenni mert le van ragadva.

Teszt-eset neve	A takarító üres gázos szobába lép
Rövid leírás	A takarító üres gázos szobába lép és kiszellőzteti a gázt.
Teszt célja	A szoba állapota megváltozik, nem lesz többé gázos.

Teszt-eset neve	A takarító nem üres gázos szobába lép
Rövid leírás	A takarító nem üres gázos szobába lép, kiszellőzteti a gázt és kiteszékeli az embereket egy szomszédos szobába.
Teszt célja	A szoba állapota megváltozik, nem lesz többé gázos, ragacsos lesz és nem lesz benne több ember.

Teszt-eset neve	Hamis logarléc felvétele
Rövid leírás	Hamis logarlécet vesz fel egy hallgató
Teszt célja	Nem történik semmi, a játék nem lesz megnyerve.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A tesztelést egy, a szkeletonhoz hasonló segédprogram fogja támogatni, aminek meg lehet adni tesztek és az elvárt eredményüket.

7.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.04.05. 20:00	1 óra	Ábel Bakó Csala Lévai Melegh	Konferencia
2024.04.06. 12:30	20 perc	Melegh	Ötletek áttekintése
2024.04.06. 12:30	1 óra	Melegh	Online
2024.04.06. 12:30	3 óra	Bakó	Use-casek
2024.04.06. 20:30	3 óra	Bakó	Bemeneti nyelv
2024.04.06 11:30	3 óra	Ábel	Változtatások implementálása
2024.04.06 19:00	4 óra	Ábel	Változtatások átvezetése az osztálydiagramok ra, leírásokba
2024.04.06. 12:30	3 óra	Csala	Kimeneti nyelv
2024.04.06 20:30	3 óra	Csala	Tesztelési terv
2024.04.07 11:30	2.5 óra	Csala	Szekvencia diagramok
2024.04.07 11:30	2.5 óra	Bakó	Use-casek
2024.04.07 11:00	3 óra	Ábel	Szekvencia diagramok