

# PROTOTÍPUS KONCEPCIÓJA

40 – ZETA

Konzulens:  
**KOVÁCS BOLDIZSÁR**

## Csapattagok

Alpek Dávid Zsolt  
Csia Klaudia Kitty  
Litaveczi Marcell  
Marton Judit  
Ruskó Eszter

C31X0F  
HA5YCV  
IPHJNB  
M0MYIM  
H8IBRC

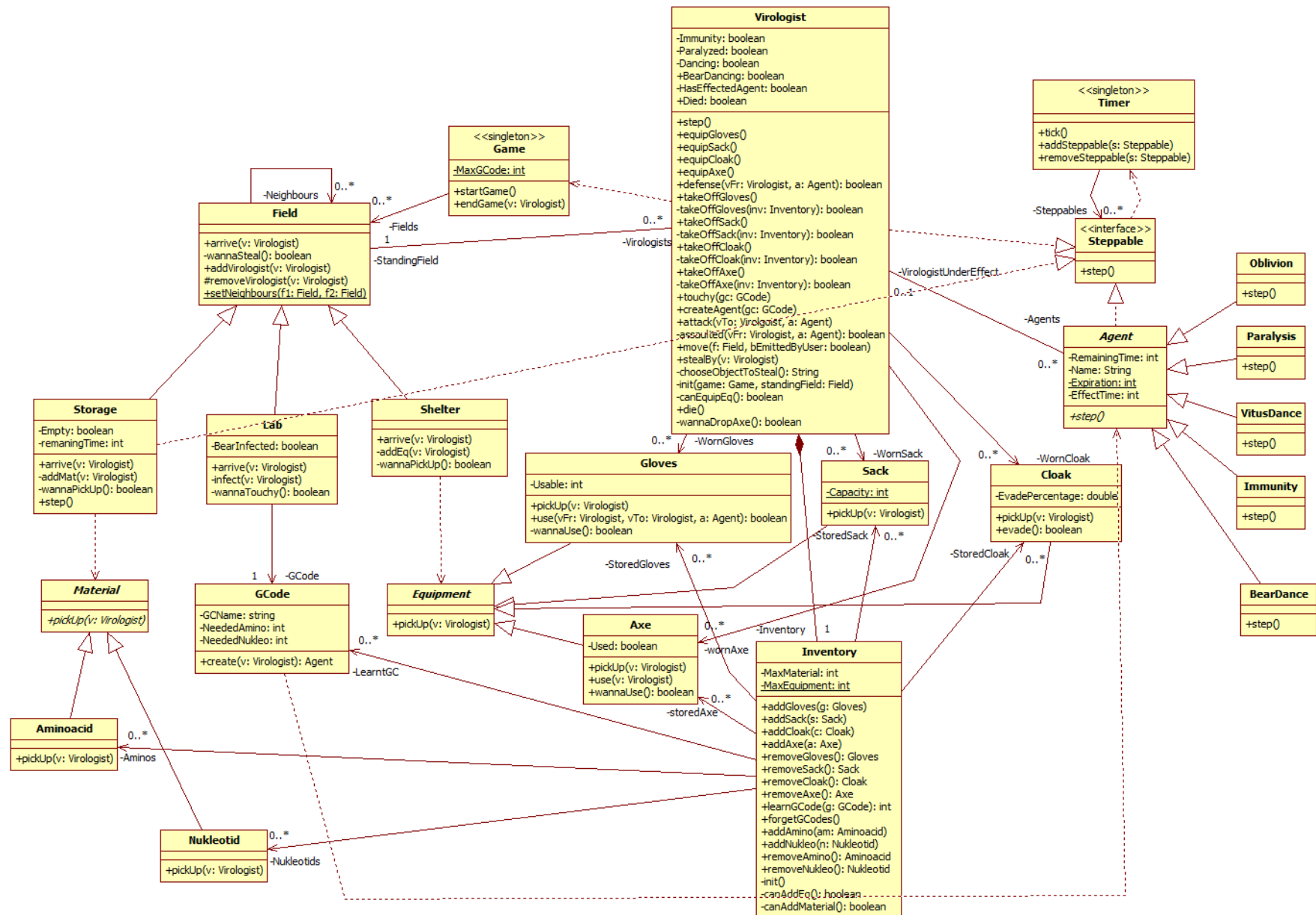
alpek.david.zsolt@gmail.com  
kitkat@sch.bme.hu  
marcell.litaveczi@gmail.com  
judit.marton@edu.bme.hu  
eszter@rusko.hu

2022. 04. 11.

## 7. Prototípus koncepciója

### 7.0 Változás hatása a modellre

#### 7.0.1 Módosult osztálydiagram



## 7.0.2 Új vagy megváltozó metódusok

### Storage

*void arrive()*: Itt mostantól figyelembe fogjuk venni azt is, hogy abban az esetben, hogyha medvevírussal fertőzött játékos lép a Storage-ba, akkor azt a raktárat ürítse ki, magyarul az adott raktárból egy körig nem vehető fel anyag, amit egy időzítő segítségével késleltetünk.

*void step()*: Ha a raktár üres, akkor van egy időzítője amely egy körig megakadályozza, hogy onnan anyagot tudjon felvenni bármelyik virológus. Ezt az időzítőt lépteti a függvény.

*void setRemaningTime(time: int)*: beállítja a remainingTime attribútumot a paraméterben kapott értékre.

### Lab

*void infect(Virologist v)*: Abban az esetben, ha a virológus olyan laborba érkezik, amely meg van fertőzve medvevírussal, úgy a beérkezett virológus is megfertőződik vele.

### Game

*void endGame(Virologist v)*: Akkor is véget tud érni egy játékosnak a játék, ha megfertőződik medvevírussal, illetve, ha megölik balta/fejsze használat segítségével.

### Virologist

*void die()*: Ez akkor hívódik meg, amennyiben a játékosra sikeresen elhasznál egy fejszét egy másik játékos.

*void equipAxe()*: Ez a metódus a fejsze felvételénél hívódik meg, mikor a virológus magára akarja venni.

*void takeOffAxe()*: Amikor a fejsze elhasználandó, vagy ellopják, ez a metódus hívódik meg.

*boolean takeOffAxe(Inventory inv)*: Amikor a játékos vissza szeretné helyezni a fejszét a tárolójába, akkor ez a metódus hívódik meg.

*void setBearDancing(b: boolean)*: beállítja a BearDancing attribútumot a paraméterben kapott értékre.

*void wannaDropAxe()*: *boolean* : Megkérdezi a játékost, hogy el akarja-e dobni a fejszét.

*void CanEquipEq()*: *boolean*: Megadja, hogy a virológus tud-e felvenni még felszerelést.

### Inventory

*void addAxe()*: Ez akkor hívódik meg, amikor a játékos a tárolójában szeretné elhelyezni a baltát.

*void removeAxe()*: Ez metódus hívódik meg, amikor a játékos fel szeretné venni a baltát, vagy éppen kilopják tőle azt.

### Axe

*void pickUp()*: Ez meghívódik, amikor a játékos felvesz egy fejsze típusú eszközt az óvóhelyről.

*void use()*: Ez az eszköz használat kérését követően hívódik meg. Használni egy másik játékosra lehet, és sikeres támadással végezni lehet a másik játékoskal, ezzel befejezni neki az adott játékot.

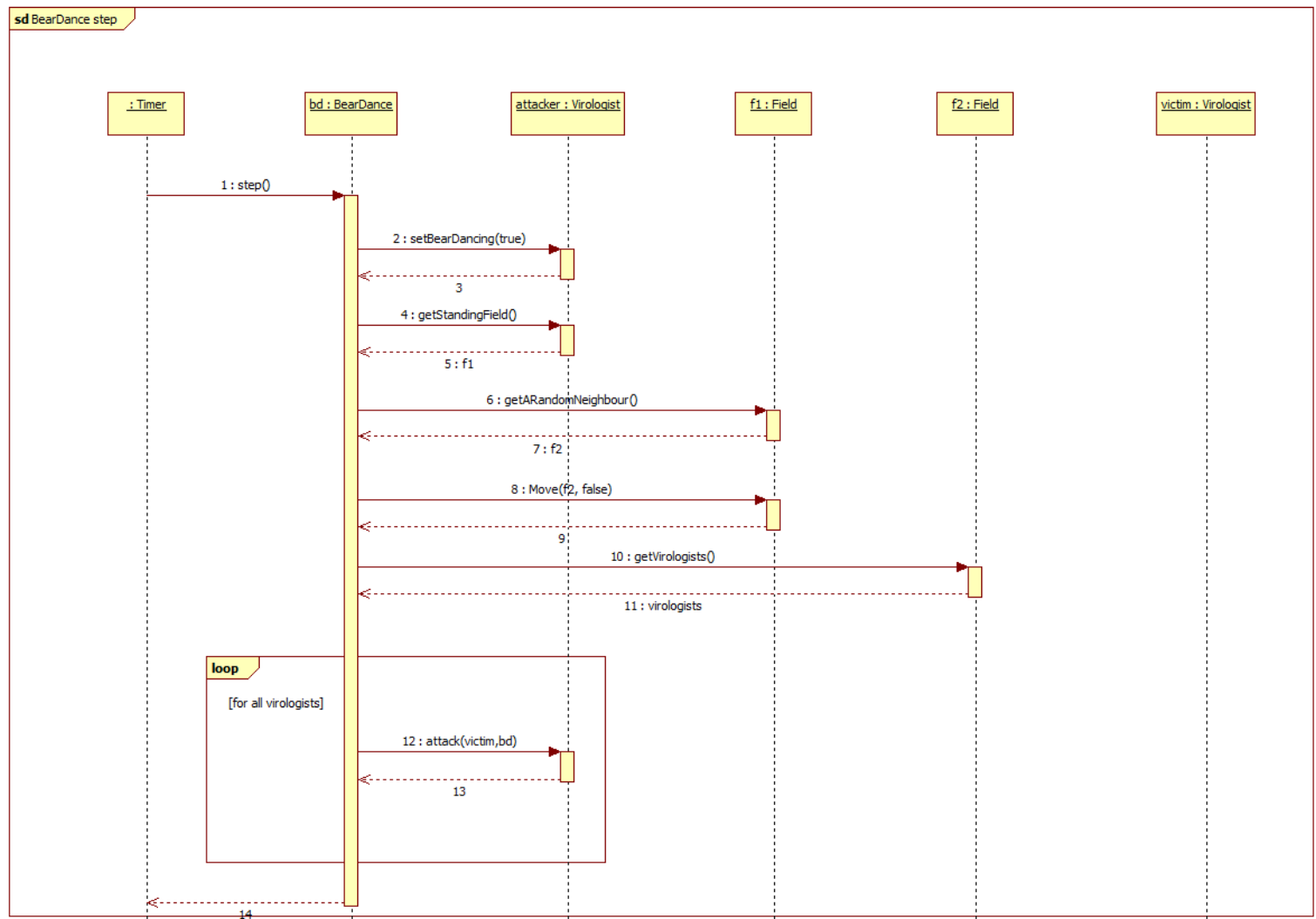
*void wannaUse()*: *boolean*: Megkérdezi a játékost, hogy szeretné-e használni a fejszét.

### BearDance

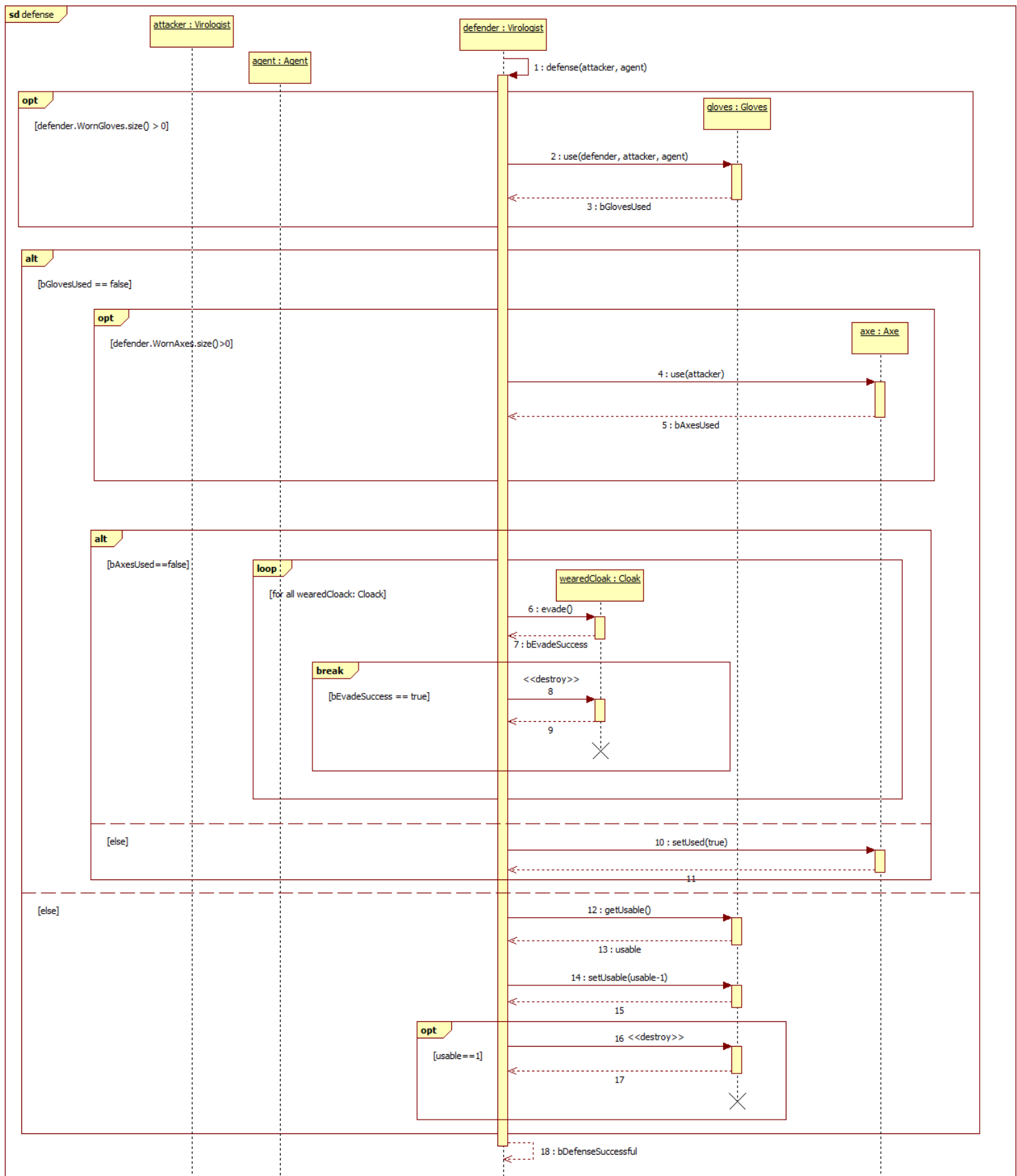
*void step()*: A medvetánc ágens léptetésével foglalkozó metódus.

### 7.0.3 Szekvencia-diagramok

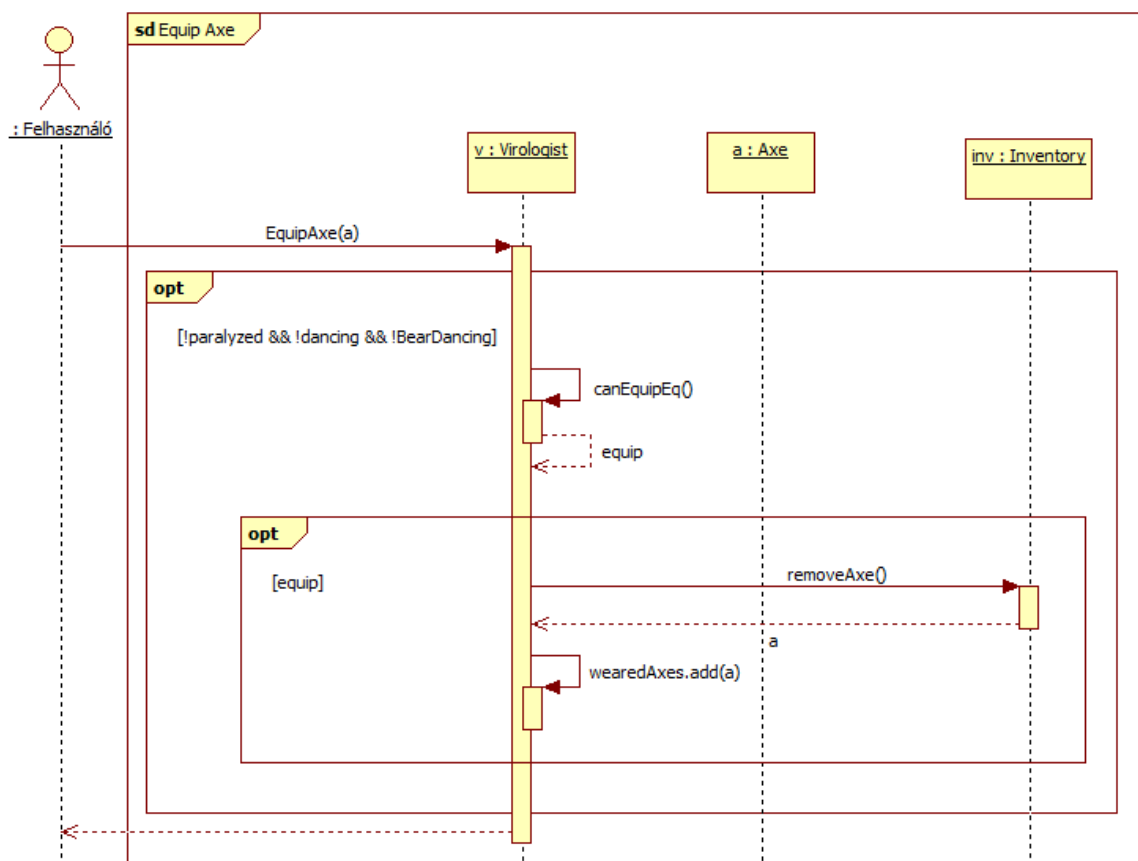
- 7.0.3.1 Bear dance step



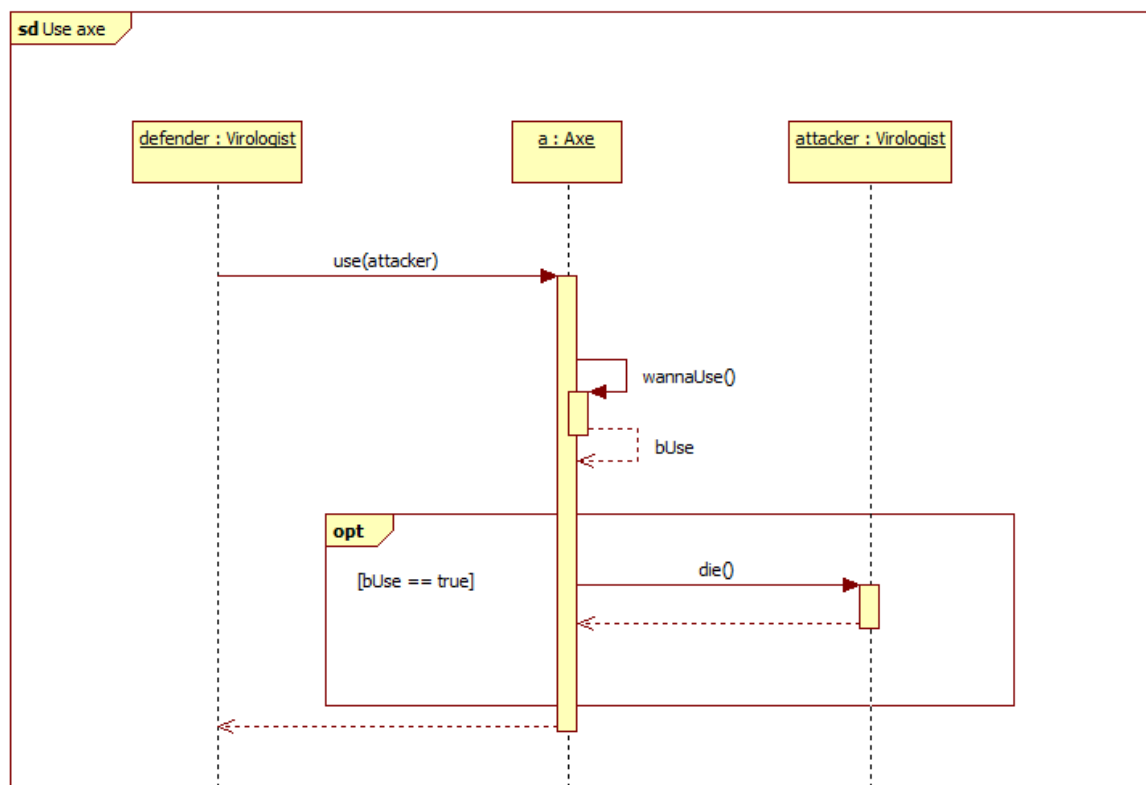
- 7.0.3.2 Defense



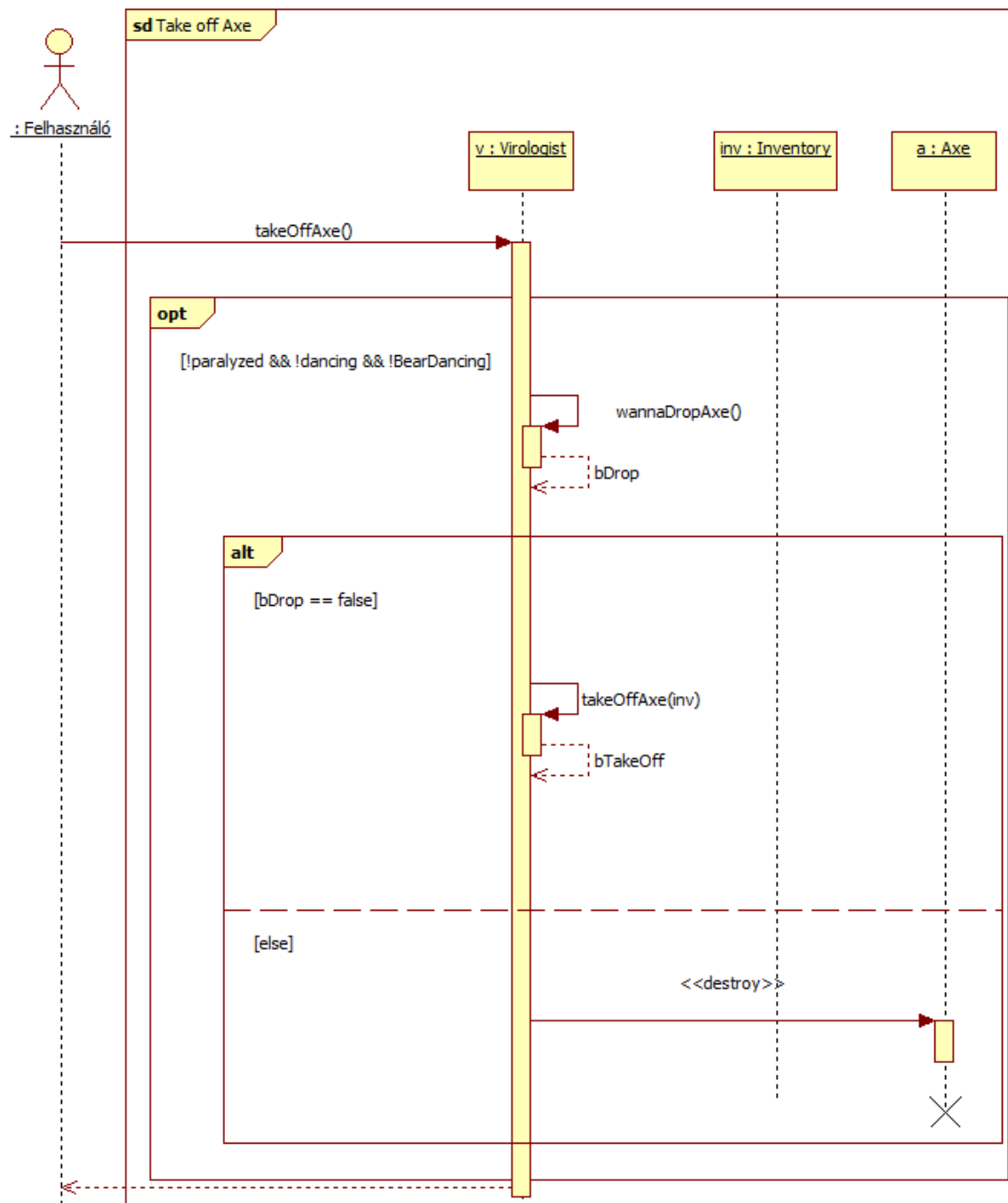
- 7.0.3.3 Equip Axe



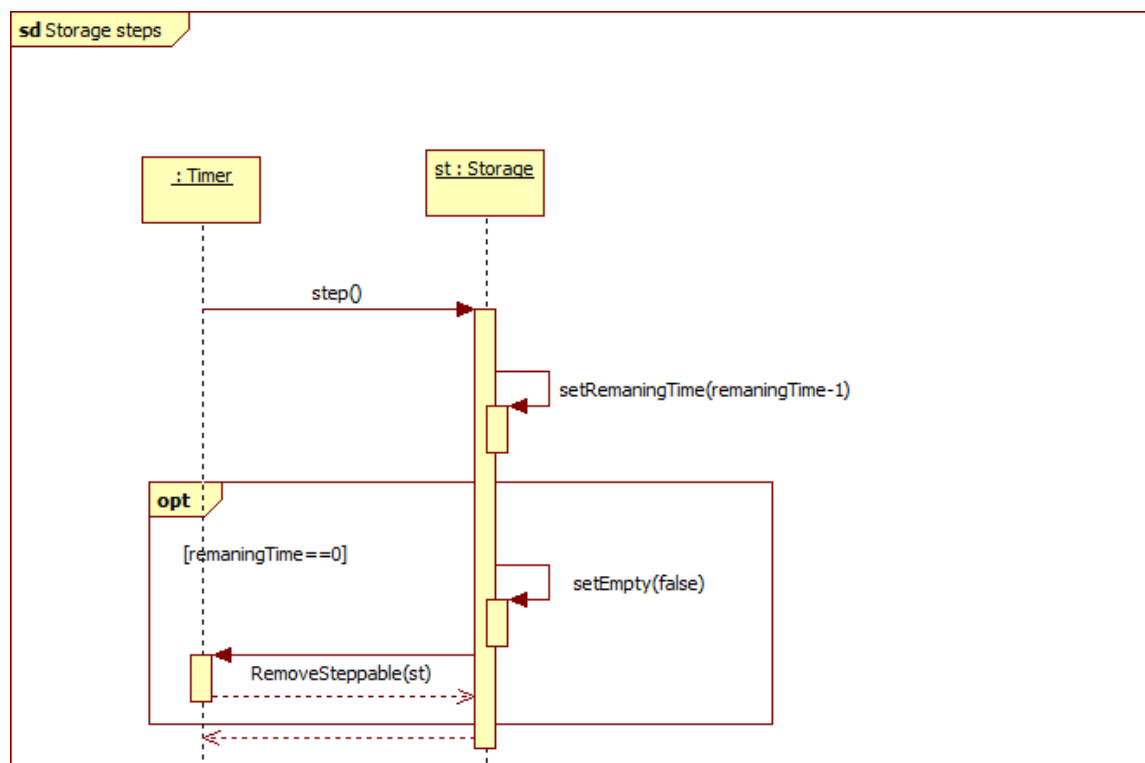
- 7.0.3.4 Use Axe



- 7.0.3.5 Take off Axe

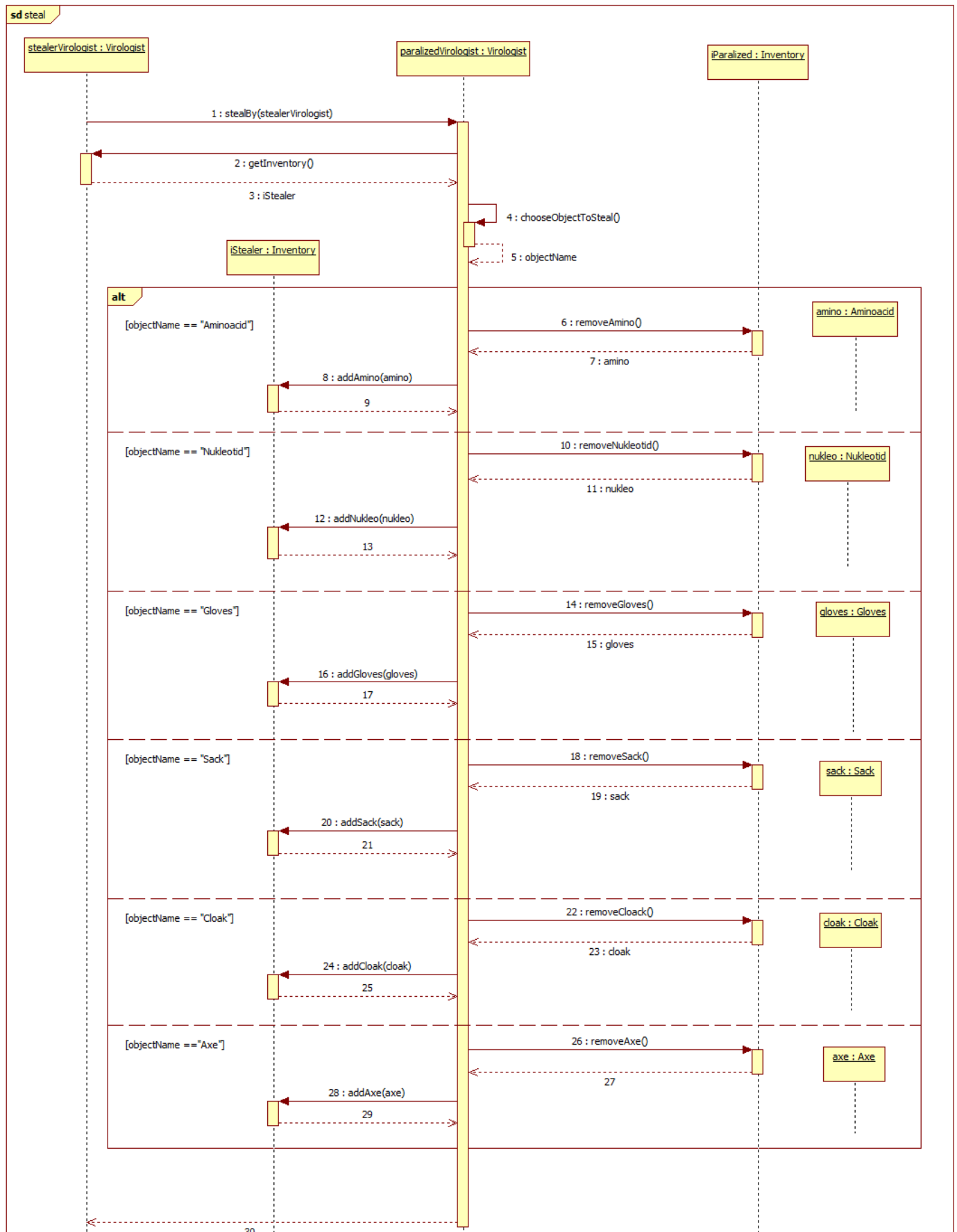


- **7.0.3.6 Storage steps**

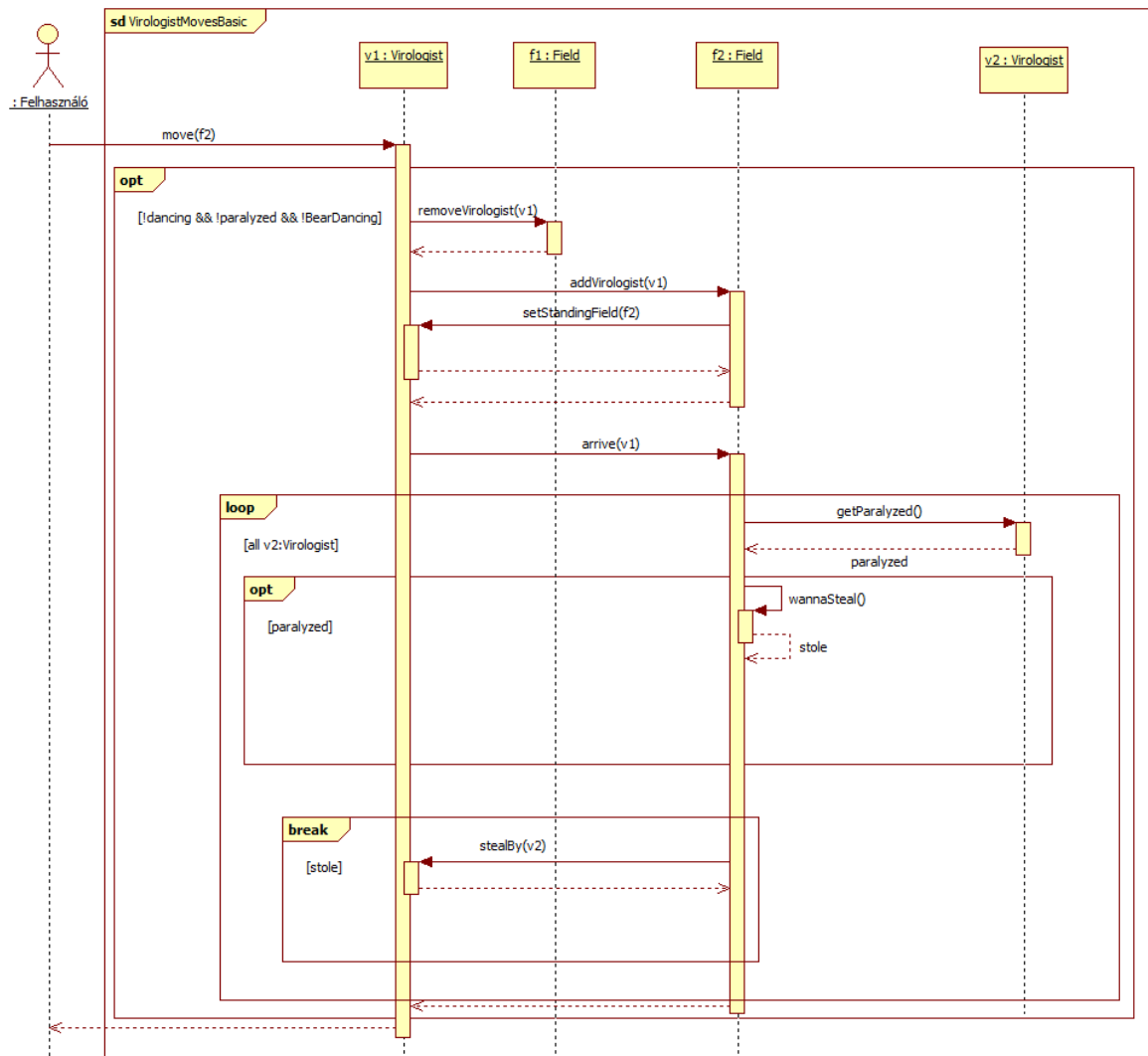




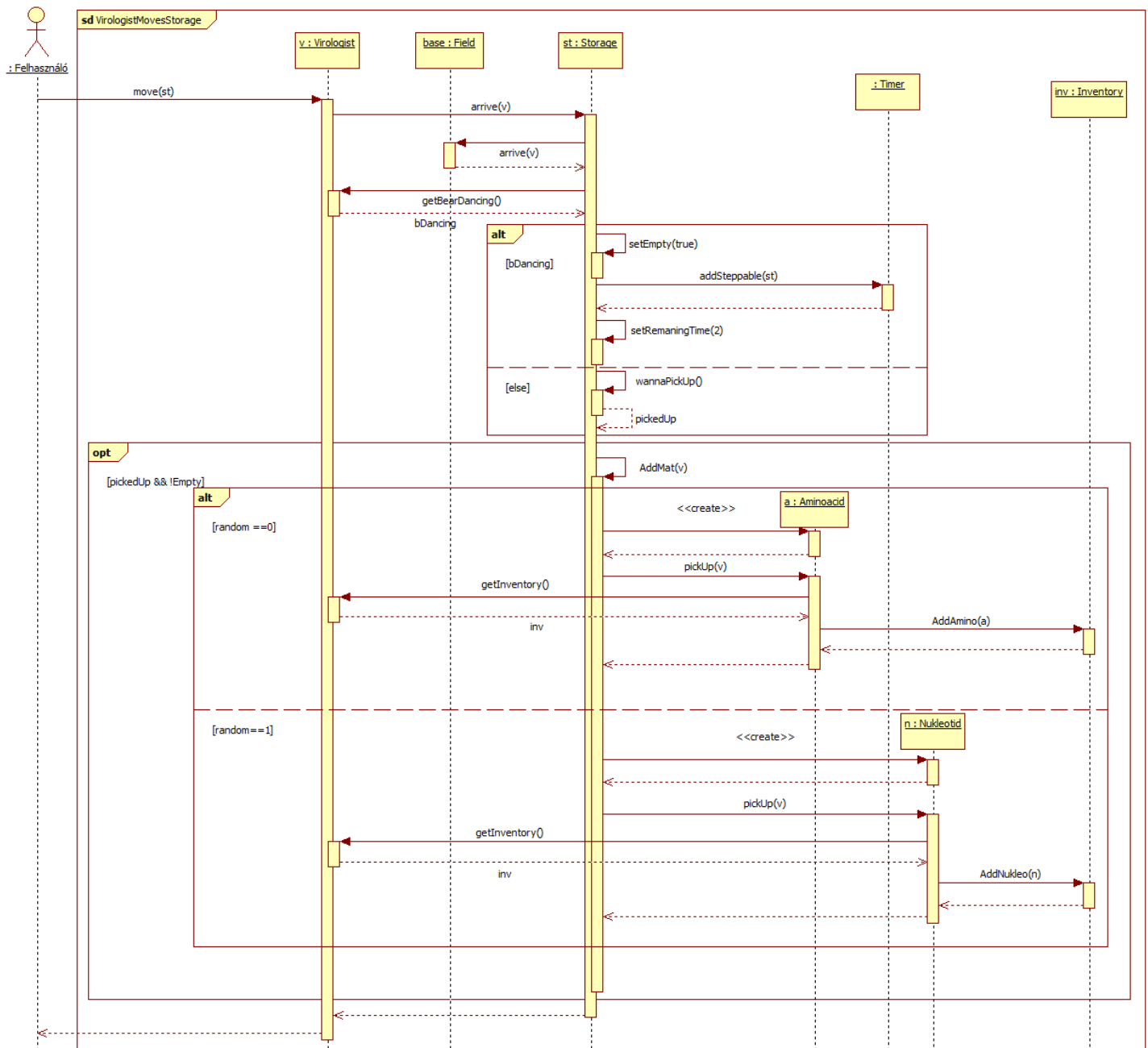
- 7.0.3.7 Stealing



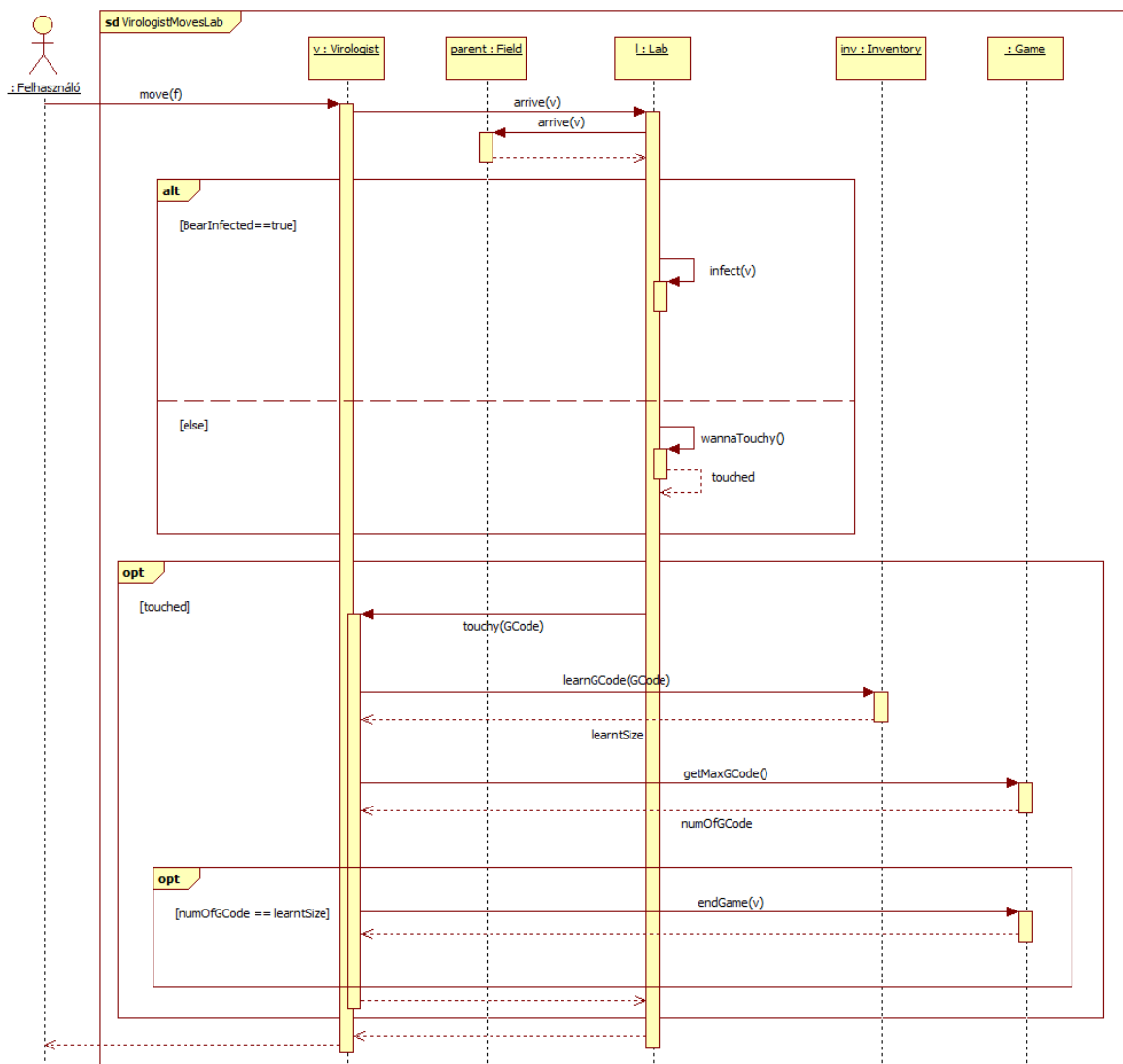
- 7.0.3.8 Virologist moves to Basic field



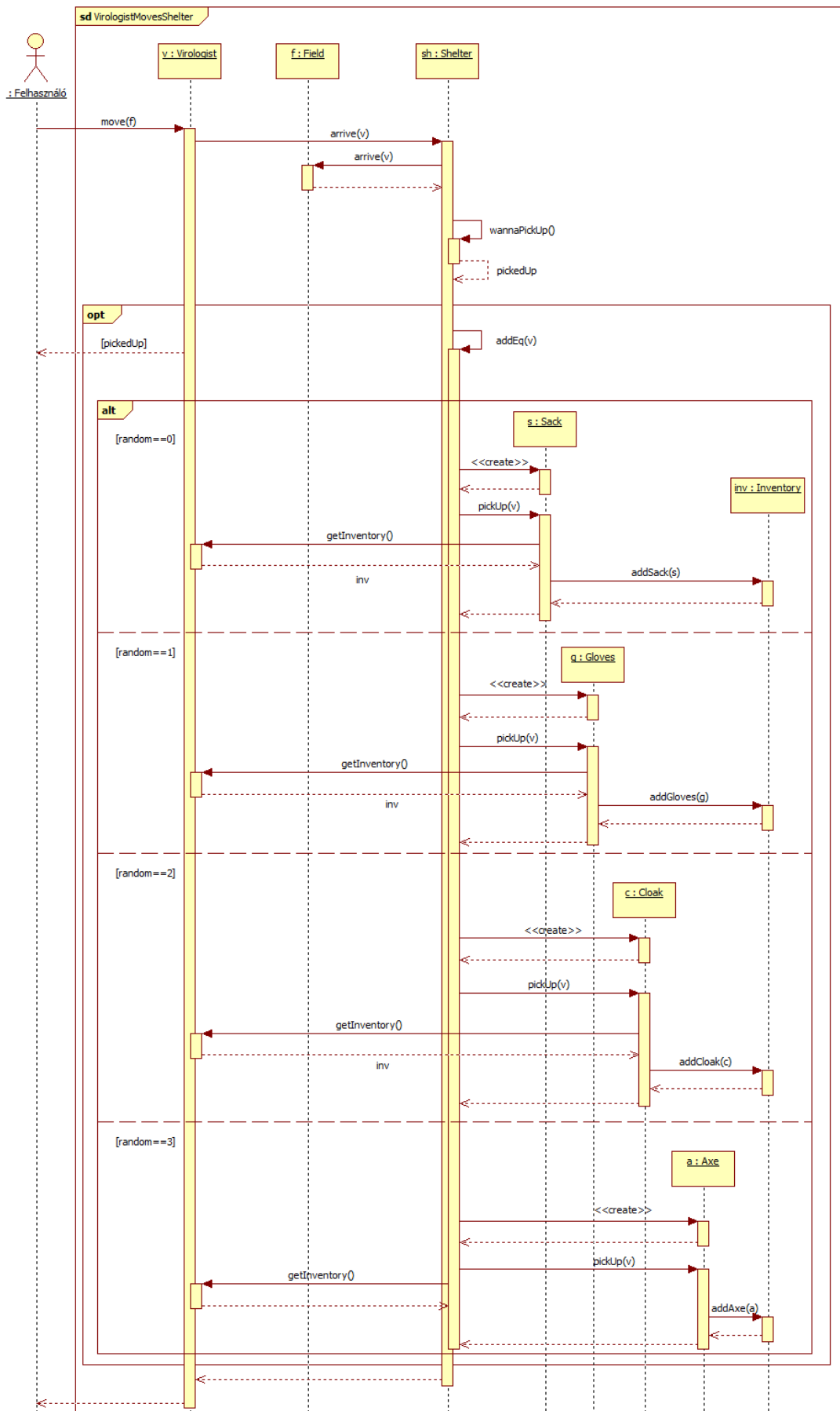
- 7.0.3.9 Virologist moves to Storage field



- 7.0.3.10 Virologist moves to Lab field



- 7.0.3.11 Virologist moves to Shelter field



## 7.1 Prototípus interface-definíciója

### 7.1.1 Az interfész általános leírása

A házi feladatunk jelenlegi állásában a program lefordulását, futtatását követően parancssoron keresztüli input beadással működik. Azonba, hogy ne kelljen minden inputot mi magunknak beadni, azaz a felhasználónak, előre elkészítettünk a program által könnyedén beolvasható fájlokat, melyeket képes beolvasni, és az alapján tovább dolgozni. Ezen fájlok megtalálhatóak lesznek majd a következő házi leadásánál a mappában.

### 7.1.2 Bemeneti nyelv

#### Parancs: 0

**Leírás:** A **CreateParalysis** teszt létrehoz egy game és field objektumot, majd ezeket paraméterként átadva hívja meg a virológus konstruktorát. Ezután egy GCodot (Paralysis) is létrehoz, amit a virológus megtanul. A virológus megpróbál létrehozni egy Paralysis ágenszt.

#### Opciók:

1. **Empty (bemenet: e):** A virológus nem rendelkezik megfelelő nyersanyagmennyiséggel (material), így az ágens létrehozása meghiúsul.
2. **Fill (bemenet: f):** A virológus inventory-ját az ágens létrehozása előtt feltöltöttük a megfelelő mennyiségű nyersanyaggal (material), így az ágens sikeresen létrejön.

#### Parancs: 1

**Leírás:** **CreatingVitusDance:** A teszt létrehoz egy Game (g), Field (f), Virologist (v) és egy GCode (gc)(VitusDance) objektumot. Ezen kívül létrehoz 3 AminoAcid és 4 Nukleotid objektumot, amit a v felszereléséhez ad, hogy így, azok felhasználásával létrehozza a Vitustánc ágenszt.

#### Opciók: -

#### Parancs: 2

**Leírás:** **CreatingImmunity:** A teszt létrehoz egy g Game, egy f Field, egy v Virologist és egy gc GCode (Immunity) objektumot. Ezen kívül létrehoz 3 AminoAcid és 4 Nukleotid objektumot, amit a v felszereléséhez ad, hogy így azok felhasználásával létrehozza az Immunitás ágenszt.

#### Opciók: -

#### Parancs: 3

**Leírás:** **TakeOffGlovesBackToInventory:** A teszt létrehoz egy Game (g), Field (f), egy Virologist (v) és egy Gloves (glov) objektumot. Majd a virológus felszerelésébe helyezi a kesztyűt és fel is veszi, hogy utána levehesse a kesztyűt. Ezzel visszakerül a felszerelések közé a kesztyű.

#### Opciók: -

#### Parancs: 4

**Leírás:** **TakeOffCloakBackToInventory:** A teszt létrehoz egy Game (g), Field (f), egy Virologist (v) és egy Cloak (c) objektumot. Majd a virológus felszerelésébe helyezi a védőköpenyt és fel is veszi, hogy utána levehesse azt. Ezzel visszakerül a felszerelések közé a védőköpeny.

#### Opciók: -

#### Parancs: 5

**Leírás:** A **TakeOffSackBackToInventory** teszt létrehoz egy game, field és inventory objektumot. Az inventory-ban elhelyez egy Sack, és 20 Aminoacid objektumot, ezeket az inventory konstruktorának paraméterként átadva. Ezután létrehoz egy virológus objektumot a game, field, és inventory objektumokat átadva neki.

A virológus felveszi a zsákot (equipSack()). Ezután a virológus leveszi a zsákot, és visszahelyezi azt az inventory-ba (takeOffSack()).

#### Opciók:

1. **Empty (bemenet: e):** A virológus tárhelyében csak annyi nyersanyag van, hogy a zsák levételével nem kell nyersanyagot eldobnia (a zsák üres volt).
2. **Fill (bemenet: f):** Miután a virológus magára vette a zsákot (equipSack()), feltöltjük az inventory-ját még 2 darab Aminoacid objektummal - egy for ciklusban 2-szer létrehozunk egy Aminoacid-ot, majd meghívjuk annak pickUp metódusát a virológust paraméterként átadva (a zsákot feltöltjük). A virológus csak ez után veszi le a zsákot, és teszi azt vissza az inventory-ba. Így a zsák levételekor 2 Aminoacid-ot el kell dobni az inventory-ból.

**Parancs: 6**

**Leírás: TakeOffGlovesAndDrop:** A teszt létrehoz egy Game(g), Field(f), Virologist(v) és egy Gloves(g) objektumot. A v belerakja az inventoryjába a kesztyűt, utána felveszi azt, majd a teszt létrehoz 5 Sack objektumot és a v felszereléséhez adja (az inventoryba). Ezután v leveszi a kesztyűt, amit eldob, mivel nem fér bele az inventory-ba.

**Opciók: -**

**Parancs: 7**

**Leírás: TakeOffCloakAndDrop:** A teszt létrehoz egy Game (g), Field (f), Virologist (v) és egy Cloak (cloak) objektumot. A v felveszi a cloak objektumot, majd létrehoz 5 Sack objektumot, és v felszereléséhez adja, hogy az megteljen. Ezek után v leveszi a cloak objektumot, amit ezáltal eldob, mert nem fér a felszerelésébe.

**Opciók: -**

**Parancs: 8**

**Leírás: A TakeOffSackAndDrop** teszt létrehoz egy game, field és inventory objektumot. Az inventory-ban elhelyez 1 Sack, 4 Gloves és 20 Nukleotid objektumot, ezeket az inventory konstruktorának paraméterként átadva. Ezután létrehoz egy virológus objektumot a game, field, és inventory objektumokat paraméterként átadva neki. A virológus felveszi a zsákot (equipSack()). Végül létrehozunk még egy Gloves objektumot, majd a virológust paraméterként átadva meghívjuk annak pickUp metódusát. Ezáltal a virológus tárhelyében kihasználjuk az eszközök számára elérhető teljes kapacitást.

A virológus leveszi a zsákot, és eldobja azt.

**Opciók:**

1. **Empty (bemenet: e):** A virológus tárhelyében csak annyi nyersanyag van, hogy a zsák levételével nem kell nyersanyagot eldobnia (a zsák üres volt).
2. **Fill (bemenet: f):** Miután a virológus magára vette a zsákot (equipSack()), feltöltjük az inventory-ját még 2 darab Aminoacid objektummal - egy for ciklusban 2-szer létrehozunk egy Aminoacid-ot, majd meghívjuk annak pickUp metódusát a virológust paraméterként átadva (a zsákot feltöltjük). A virológus csak ez után veszi le a zsákot, és teszi azt vissza az inventory-ba. Így a zsák levételekor 2 Aminoacid-ot el kell dobni az inventory-ból.

**Parancs: 9 <ágenstípus>**

**Leírás: UseAgentOnVirologistWhoHasGlovesAndCloakAndParalyzed:** A teszt létrehoz egy Game(g), Field(f), Virologist(v1, v2), Gloves(glov), Cloak(cloak) egy bemenetnek megfelelő Immunity(i)/Oblivion(o)/VitusDance(vd)/Paralysis(p) objektumot. Hozzáadja a v2 inventory-jához a glov-ot, majd felveszi azt a v2. Ezután beállítja v2-t bénultnak. Az a-t hozzáadja a v1 ágenseihez, ezután v1 rákenni a v2-re az ágenst, ami sikerül, mivel v2 lebénult, ezért nem tud védekezni, tehát a támadás sikeres.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 10<ágenstípus>**

**Leírás: UseAgentWhileParalyzed:** A teszt létrehoz egy Game (g), Field (f), két Virologist (v1 és v2) és egy, a paraméternek megfelelő típusú Agent objektumot. A v1 állapotát bénult állapotra állítja, és az o objektumot odaadja a v1-nek, aki ezek után megpróbálja rákenni a v2-es virológusra a kapott ágenst, de bénult állapota miatt semmi se történik.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 11**

**Leírás:** A **UseParalysisOnNotImmunVirologist** teszt létrehoz egy game és field objektumot. Ezután létrehoz egy v1 és v2 virológust, mindkettőnek ugyanazon game és field objektumot konstruktorban átadva, valamint v1 konstruktorának egy Paralysis objektumot tartalmazó ágens listát is átadva. Fontos, hogy a két virológus ugyanazon mezőn álljon. A virológusok egyike sem áll ágens hatása alatt, és nincs semmilyen felszerelésük. A v1 virológus megtámadja a v2 virológust a Paralysis ágenssel. A v2 virológus lebénul.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 12**

**Leírás:** **UseImmunityOnNotImmunVirologist:** A teszt létrehoz egy Game(g), Field(f), Virologist(v1, v2), Immunity(i) objektumot. Ezután beállítja v2-t nem immunisnak, v1 megkapja i-t, majd v1 megtámadja v2-t, ami sikerül.

**Opciók:** -

**Parancs: 13**

**Leírás:** **UseOblivionOnNotImmunVirologist:** A teszt létrehoz egy Game (g), Field (f), két Virologist (v1 és v2) és egy Oblivion (o) objektumot. Ezek után a v2 Immunity tulajdonságát hamis értékre állítja. A v1 objektumhoz hozzáadja az o objektumot, ezek után a v1 megpróbálja rákenni v2-re a kapott ágens, ami sikerül neki.

**Opciók:** -

**Parancs: 14**

**Leírás:** A **UseAgentOnVirologistWhoHasCloak** teszt létrehoz egy game field, és Cloak lista objektumot egyetlen elemmel. Ezután létrehoz egy inventory objektumot, melynek átadjuk paraméterben az egy elemű Cloak listát. Majd létrehozunk egy v1 és v2 virológust, mindkettőnek ugyanazon game és field objektumot konstruktorban átadva, illetve a v2-nek az inventory objektumot, a v1-nek pedig egy Paralysis ágens tartalmazó Agent listát is átadva. A v2 virológus felveszi magára az inventory-ból a Cloak eszközt. A v1 virológus megtámadja a v2 virológust a bénító ágenssel.

**Opciók:**

1. **Defense succeed (bemenet: 1):** A köpeny sikeresen megvédi a megtámadott virológust, a v2 virológus köpenye megsemmisül, és a virológus nem lesz béna.
2. **Defense failed (bemenet: 0):** A köpeny nem védi meg a megtámadott virológust, a v2 virológus köpenye megmarad, és a virológus megbénul.

**Parancs: 15<ágenstípus>**

**Leírás:** **UseAgentOnVirologistWhoHasImmunity:** A teszt létrehoz egy Game(g), Field(f), Virologist(v1, v2), egy bemenetnek megfelelő Immunity(i)/Oblivion(o)/VitusDance(vd)/Paralysis(p) objektumot. Ezután v2-n beállítja az immunitást, hozzáadja v1-hez az ágens, majd v1 megtámadja v2-t, ami nem sikerül, mivel v2 immunis.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 16<ágenstípus>**

**Leírás:** **UseAgentOnVirologistWhoHasGloves:** A teszt létrehoz egy Game (g), egy Field (f), két Virologist (v1 és v2), egy Gloves (glov) és egy, a paraméternek megfelelő típusú Agent objektumot. Majd a v2 felszereléséhez hozzáadja a glov objektumot, a v1-hez pedig a létrehozott ágens. A v1 megpróbálja rákenni az ágens a v2-re, akinek azonban van kesztyűje. A teszt további lefolyása a felhasználói inputtól függ.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).



**Parancs: 17**

**Leírás:** A **UseAgentOnVirologistWhoHasGlovesAndCloak** teszt létrehoz egy Game Field, Gloves lista és Cloak lista objektumot, mindkét listát egy-egy elemmel. Ezután létrehoz egy inventory objektumot, melynek átadjuk paraméterben az egy elemű Gloves és Cloak listát. Majd létrehozunk egy v1 és v2 virológust, mindkettőnek ugyanazon game és field objektumot konstruktorban átadva, illetve a v2-nek az inventory objektumot, v1-nek pedig egy 1 elemű (bénító ágenszt tartalmazó) ágens listát is átadva. A v2 virológus felveszi magára az inventory-ból a Gloves, majd a Cloak eszközt. A v1 virológus megtámadja a v2 virológust a bénító ágenssel.

**Opciók:**

1. **Defense failed (bemenet: 0):** A v2 virológus nem használja a kesztyűt, és a köpeny sem védi meg őt. A v2 virológusnak minden eszköze megmarad, és lebénul.
2. **Defense using gloves (bemenet: g):** A v2 virológus használja a kesztyűt, így a kesztyű használhatóságának száma 1-el csökken. A köpenye megmarad, nem lesz béna. A v1 virológus lebénul.
3. **Defense using cloak (bemenet: c):** A v2 virológus nem használja a kesztyűt, de a köpeny megvédi őt. A v2 virológus köpenye megsemmisül. Egyik virológus sem bénul le.

**Parancs: 18<ágenstípus>**

**Leírás:** **UseAgentOnVirologistWhoHasGlovesAndIHaveToo:** A teszt létrehoz egy Game(g), Field(f), Virologist(v1, v2), Gloves(glov1, glov2) és egy bemenetnek megfelelő Immunity(i)/Oblivion(o)/VitusDance(vd)/Paralysis(p) objektumot. Ezután v2 megkapja glov2-t, v1 glov1-et, majd v1, v2 felveszik a kesztyűket, v1 megkapja az ágenszt, majd megtámadja v2-t. A támadás sikeres, mert először visszapattan v2-ről az ágens, majd v1-ről is így v2 megkapja a támadást.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 19<ágenstípus>**

**Leírás:** **UseAgentOnVirologistWhoHasGlovesAndIHaveCloak:** A teszt létrehoz egy Game (g), egy Field (f), két Virologist (v1 és v2), egy Gloves (glov) egy Cloak (cloak) és egy ágenstípusnak megfelelő típusú Agent objektumot. A v2 megkapja a glov objektumot, és fel is veszi, a v1 pedig megkapja a cloak objektumot, amit felvesz. Ezek után a v1 megkapja az ágenszt, amit megpróbál felkenni a v2-re. A teszt további lefolyása a felhasználói inputtól függ.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 20**

**Leírás:** A **UseAgentOnVirologistWhoHasGlovesAndIHaveImmunity** teszt létrehoz egy Game Field, és egy Gloves lista objektumot egy elemmel. Ezután létrehoz egy inventory objektumot, melynek átadjuk paraméterben az egy elemű Gloves listát. Majd létrehozunk egy v1 és v2 virológust, mindkettőnek ugyanazon game és field objektumot konstruktorban átadva, illetve a v2-nek az inventory objektumot, v1-nek egy egy elemű (bénító ágenszt tartalmazó) ágens listát is átadva. A v2 virológus felveszi magára az inventory-ból a Gloves eszközt. A v1 virológus megtámadja a v2 virológust a bénító ágenssel.

**Opciók:**

1. **Defense using gloves (bemenet: 1):** A v2 virológus használja a kesztyűt, így a kesztyű használhatóságának száma 1-el csökken, v2 virológus nem lesz béna. A v1 virológus sem bénul le, mivel immunitás védi.
2. **Defense fail (bemenet: 0):** A v2 virológus nem használja a kesztyűt, és lebénul.

**Parancs: 21<ágenstípus>**

**Leírás:** **SelfuseAgentWhenHasCloak:** A teszt létrehoz egy Game (g), egy Field (f), egy Virologist (v1) és egy Cloak(cloak) és egy bemenetnek megfelelő Immunity(i)/Oblivion(o)/VitusDance(vd)/Paralysis(p)

objektumot. Ezek után a v1 megkapja a cloak objektumot és fel is veszi, illetve megkapja az ágenst. Ezek után v1 felveszi a cloak objektumot és megkapja az ágenst, amit magára ken.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni a kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 22<ágenstípus>**

**Leírás: SelfuseAgentWhenHasGloves:** A teszt létrehoz egy Game (g), egy Field (f), egy Virologist (v1) és egy Gloves (glov) objektumot, illetve egy, a paraméternek megfelelő típusú Agent objektumot. Ezek után a v1 megkapja a glov objektumot és fel is veszi, illetve megkapja az ágenst. Ezek után v1 felveszi a glov objektumot és megkapja az ágenst, amit magára ken.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 23**

**Leírás: A SelfuseAgentWhenHasCloakAndGloves** teszt létrehoz egy Game, Field, Gloves lista és Cloak lista objektumot, mindkét listát egy-egy elemmel. Ezután létrehozunk egy inventory objektumot, melynek átadjuk paraméterben az egy elemű Gloves és Cloak listát. Majd létrehozunk egy virológust, a Game, Field, Inventory objektumot és egy 1 elemű (Immunity ágenst tartalmazó) ágenst listát konstruktorban paraméterként átadva. A virológus felveszi magára az inventory-ból a Gloves, majd a Cloak eszközt. Ezután megtámadja önmagát az Immunity ágenssel. A virológus az ágens hatása alá kerül.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 24<ágenstípus>**

**Leírás: SelfuseAgentWhenHasNoAgents:** A teszt létrehoz egy Game(g), Field(f) és Virologist(v1) objektumot, majd a bemenetnek megfelelő Immunity(i)/Oblivion(o)/VitusDance(vd)/Paralysis(p) ágenst magára keni.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 25<ágenstípus>**

**Leírás: SelfuseAgentWhenParalyzed:** A teszt létrehoz egy Game (g), egy Field (f), egy Virologist (v1) és egy, a paraméternek megfelelő típusú Agent objektumot. Ezek után a v1 paralyzed állapotát igaz értékre állítja, Majd hozzáadja az ágenst a v1-hez és v1 megpróbálja magára kenni. A teszt során nem történik semmi, mivel v1 bénult állapotban van.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 26**

**Leírás: A SelfuseAgentWhenHasImmunity** teszt létrehoz egy Game és Field objektumot, majd létrehozunk egy virológust, a Game és Field objektumot, valamint egy 1 elemű (VitusDance ágenst tartalmazó) ágens listát konstruktorban paraméterként átadva, illetve szintén a konstruktorban az immunitást true-ra állítva. A virológus megtámadja önmagát a VitusDance ágenssel. A virológusra az ágens nem lesz hatással, mivel már egy másik ágens (Immunity) hatása alatt áll.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 27**

**Leírás: SelfuseOblivion:** A teszt létrehoz egy Game (g), egy Field (f), egy Virologist (v1) és egy Oblivion(o) objektumot. A v1 megkapja az o ágenst, majd beadja magának.

**Opciók:** -

**Parancs: 28**

**Leírás: SelfuseVitusDance:** A teszt létrehoz egy Game (g), egy Field (f), egy Virologist (v1) és egy VitusDance (o) objektumot. A v1 megkapja az o ágenst, majd beadja magának.

**Opciók:** -

**Parancs: 29**

**Leírás: A SelfuseImmunity** teszt létrehoz egy Game és Field objektumot, majd létrehozunk egy virológust, a Game és Field objektumot, valamint egy 1 elemű (Immunity ágenst tartalmazó) ágens listát konstruktorban paraméterként átadva, illetve szintén a konstruktorban az immunitást true-ra állítva. A virológus megtámadja önmagát a Immunity ágenssel. A virológus az ágens hatása alá kerül.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 30**

**Leírás: SelfuseParalysis:** A teszt létrehoz egy Game (g), egy Field (f), egy Virologist (v1) és egy Paralysis(p) objektumot. A v1 megkapja az p ágenst, majd beadja magának.

**Opciók:** -

**Parancs: 31**

**Leírás: MoveWhileDancing:** A teszt létrehoz egy Game (g), két Field (f1, f2) és egy Virologist (v1) objektumot. Az f1 és f2 mezőt szomszédossá teszi. A v1 Dancing állapotát igaz értékre állítja és a v1-et átlépteti az f2 mezőre.

**Opciók:** -

**Parancs: 32**

**Leírás: A MoveToNotNeighbour** teszt létrehoz egy Game és két Field objektumot (f1 és f2). Ezután létrehoz egy virológus objektumot a Game és az f1 objektum konstruktorban való átadásával. A virológus az f2 fieldre próbál lépni.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 33**

**Leírás: VirologistMoves:** A teszt létrehoz egy Game(g), Field(f1, f2), Virologist(v1) objektumot. A v1 jelenlegi mezője f1 lesz. Beállítja f2-t f1 szomszédjának, majd v1-t f1-ről f2-re áthelyezi.

**Opciók:** -

**Parancs: 34**

**Leírás: VirologistMovesParalyzed:** A teszt létrehoz egy Game (g), két Field (f1, f2) és egy Virologist (v) objektumot. Az f1 és f2 mezőt szomszédossá teszi. Ezek után a v Paralyzed tulajdonságát igaz értékre állítja, majd átlépteti az f2 mezőre.

**Opciók:** -

**Parancs: 35**

**Leírás: A ParalyzedTouchy** teszt létrehoz egy Game és egy Lab objektumot, majd ezeket a virológus objektum létrehozásánál átadja paraméterként a konstruktorban. Ezután a teszt meghívja a virológus touchy metódusát a Laboron található genetikai kód paraméterként való átadásával.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 36<ágenstípus>**

**Leírás: NewGCTouchy:** A teszt létrehoz egy Game(g), Lab(l), Virologist(v) és a bemenetnek megfelelő Immunity/Oblivion/VitusDance/Paralysis ágenshez tartozó GCode(gc) objektumot. Lekéri v inventory-ját és v megtanulja gc-t.

**Opciók:**

1. A megtanulni kívánt ágens egy Oblivion (ágenstípus: o).
2. A megtanulni kívánt ágens egy Paralysis (ágenstípus: p).
3. A megtanulni kívánt ágens egy VitusDance (ágenstípus: v).

## 4. A megtanulni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 37**

**Leírás: OldGCTouchy:** A teszt létrehoz egy Game (g), egy Lab (l) egy Virologist (v) és egy GCode (gc) objektumot. Ezek után a v kétszer letapogatja a kódot.

**Opciók:** -

**Parancs: 38**

**Leírás: A WinGameByLastTouchy** teszt létrehoz egy Game és egy Lab objektumot, majd ezeket a virológus objektum létrehozásánál átadja paraméterként a konstruktorban. A Lab mezőn a VitusDance ágens kódja van. A virológus inventory-ját úgy hozzuk létre, hogy átadunk neki egy GCode listát, melyben szerepel az Immunity, Paralysis és Oblivion ágenseket előállító GCode. A labor letapogatásával a virológus megtanulja a genetikai kódot, és megnyeri a játékot.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 39<anyagtípus>**

**Leírás: ParalyzedPickUpMat:** A teszt létrehoz egy Game(g), Storage(st), Virologist(v) és a bemenetnek megfelelő Nukleotid(n)/Aminoacid(a) objektumot. A vt beállítja bénultra, majd megpróbálja v felvenni az anyagot, de nem sikerül, mivel v le van bénulva.

**Opciók:**

1. A felvenni kívánt anyag egy Nukleotid (anyagtípus: n).
2. A felvenni kívánt anyag egy Aminoacid (anyagtípus: a).

**Parancs: 40**

**Leírás: NukleoPickUp:** A teszt létrehoz egy Game (g), egy Storage (st) egy Virologist (v) és egy Nukleotid (n) objektumot. Ezek után a v felveszi az n objektumot.

**Opciók:** -

**Parancs: 41**

**Leírás: Az AminoPickUp** teszt létrehoz egy Game és egy Storage objektumot, majd ezeket a virológus objektum létrehozásánál átadja paraméterként a konstruktorban. Ezután létrehozunk egy Aminoacid objektumot, és meghívjuk a pickUp metódusát a virológust paraméterként átadva. Az aminoacid bekerül az Inventory-ba.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 42<anyagtípus>**

**Leírás: FullPickUpMat:** A teszt létrehoz egy Game(g), Storage(st), Virologist(v) és a bemenetnek megfelelő Nukleotid(n)/Aminoacid(a) objektumot. Lekéri v inventoryját(inv), majd hozzáad 20 anyagot, amivel betelik az inv.

**Opciók:**

1. A felvenni kívánt anyag egy Nukleotid (anyagtípus: n).
2. A felvenni kívánt anyag egy Aminoacid (anyagtípus: a).

**Parancs: 43**

**Leírás: ParalyzedPickUpEq:** A teszt létrehoz egy Game (g), egy Storage (st), egy Virologist (v) és egy Nukleotid (n) objektumot. Ezek után a v Paralyzed állapotát igaz értékre állítja, majd a v felveszi az n objektumot.

**Opciók:** -

**Parancs: 44**

**Leírás: A GlovesPickUp** teszt létrehoz egy Game és egy Shelter objektumot, majd ezeket a virológus objektum létrehozásánál átadja paraméterként a konstruktorban. Ezután létrehozunk egy Gloves objektumot, és meghívjuk a pickUp metódusát a virológust paraméterként átadva. A Gloves objektum bekerül az Inventory-ba.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 45**

**Leírás: CloakPickUp:** A teszt létrehoz egy Game(g), Shelter(sh), Virologist(v), Cloak(c) objektumokat. A v-t a sh-ra teszi, majd v beteszi c-t az inventoryjába.

**Opciók:** -

**Parancs: 46**

**Leírás: SackPickUp:** A teszt létrehoz egy Game (g), egy Shelter (sh) egy Virologist (v) és egy Sack (s) objektumot. Ezek után a v felveszi az s objektumot.

**Opciók:** -

**Parancs: 47**

**Leírás: A FullPickUpEq** teszt létrehoz egy Game, egy Shelter és egy Inventory objektumot úgy, hogy az Inventory konstruktorában egy 5 elemű Gloves listát adunk át, majd a Game-et, Shelter-t és Inventory-t a virológus objektum létrehozásánál átadja paraméterként a konstruktorban. Ezután létrehozunk egy Gloves objektumot, és meghívjuk a pickUp metódusát a virológust paraméterként átadva. A Gloves objektum nem kerül be az Inventory-ba.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 48**

**Leírás: CreatingOblivion:** A teszt létrehoz egy Game(g), Field(f), Virologist(v), GCoed(Oblivion)(gc) objektumot. A v megtanulja gc-t, majd csinál egy ágenszt a gc alapján.

**Opciók:** -

**Parancs: 49**

**Leírás: DanceWhileMove:** A teszt létrehoz egy Game (g), két Field (f1 és f2) és egy Virologus (v) objektumot. A két mezőt szomszédosnak állítja, illetve a virológus Dancing tulajdonságát igaz értékre állítja. Ezek után megpróbál v-vel átlépni f1-ről f2-re.

**Opciók:** -

**Parancs: 50**

**Leírás: A MoveToNowhere** teszt létrehoz egy Game és egy Field (f1) objektumot, majd ezeket átadja a Virologus objektumnak a konstruktora paramétereiként. Ezután meghívja a virológus move metódusát null pointerrel értékű Field (f2) paraméterrel.

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 51**

**Leírás: TakeOnOffEq:** A teszt létrehoz egy Game(g), Shelter(sh), Virologist(v), Sack(s) objektumot. Ezután lekéri a v inventoryját(inv), hozzáadja s-t inv-hez, majd v felveszi s-t aztán leveszi azt.

**Opciók:** -

**Parancs: 52**

**Leírás: StepParalyzed:** A teszt létrehoz egy Game (g), egy Field (f), egy Virologist (v1) és egy Paralysis típusú Agent (a) objektumot. Az ágenssel megfertőzi a virológust, majd ameddig tart az ágens hatása, addig lépteti az ágenszt (kifejti annak hatását a virológusra.)

**Opciók:** -

**Parancs: 53**

**Leírás: A StepVitusDance** teszt létrehoz egy Game és néhány Field objektumot (f0, f1 ....). Ezután ezeket beállítja szomszédoknak. Létrehozunk egy virológus objektumot, melynek konstruktorában átadjuk a game, és az f0 objektumokat, és a dancing paramétert true-ra állítjuk. Ezután létrehozunk egy VitusDance ágenszt, és meghívjuk a setVirologistUnderEffect metódusát a létrehozott virológus-t paraméterként átadva. Amíg az ágens getRemainingTime metódusa nem 0-val tér vissza (6 lépés), meghívjuk a step metódusát.

**Opciók:**

1. **NonRandom (bemenet: 0):** A teszt során csak 2 mezőt hozunk létre, amik egymás szomszédai. A virológus az effecttime lejártáig egyik mezőről a másikra lép.
2. **Random (bemenet: 1):** A teszt során több mezőt hozunk létre, és azt ellenőrizzük, hogy minden lépés után különböző, de szomszédos mezőn kell állnia a virológusnak.

**Parancs: 54**

**Leírás: StepOblivion:** A teszt létrehoz egy Game (g), egy Field (f), egy Virologist (v1) és egy Oblivion típusú Agent (a) objektumot. Az ágenssel megfertőzi a virológust, majd ameddig tart az ágens hatása, addig lépteti az ágenszt (kifejti annak hatását a virológusra.)

**Opciók:** -

**Parancs: 55**

**Leírás: StepImmunity:** A teszt létrehoz egy Game (g), egy Field (f), egy Virologist (v1) és egy Immunity típusú Agent (a) objektumot. Az ágenssel megfertőzi a virológust, majd ameddig tart az ágens hatása, addig lépteti az ágens (kifejti annak hatását a virológusra.)

**Opciók:** -

**Parancs: 56**

**Leírás:** A UseUnavailableAgent teszt létrehoz egy Game és egy Field objektumot, majd ezeket átadja a Virológus objektumnak a konstruktora paramétereként. Ezután a virológus Immunity ágens ken önmagára (attack metódus).

**Opciók:** - (a tesztesetnek nincs bemenete)

**Parancs: 57**

**Leírás: StealFromNonParalyzed:** A teszt létrehoz egy Game(g), Field(f), Virologist(v1, v2) objektumot. Mindkettő virológus a f-n áll. A v1 megpróbál lopni v2-től, viszont ez nem sikerül, mivel v2 nincs lebánulva.

**Opciók:** -

**Parancs: 58**

**Leírás: StealAsParalyzed:** A teszt létrehoz egy Game (g), egy Field (f) és két Virologist (v1 és v2) objektumot. A v2 objektum bénult állapotban jön létre. Ezek után a v2 megpróbálja meglopni a v1-t.

**Opciók:** -

**Parancs: 59**

**Leírás:** A StealFromPoorGuy teszt létrehoz egy Game és egy Field objektumot, majd ezeket átadja a két virológus v1, és v2 objektumnak a konstruktora paramétereként. A v1 virológus ellop egy tárgyat (anyagot vagy eszközt) v2-től.

**Opciók:**

1. StealGloves (bemenet: g): v1 virológus kesztyűt akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
2. StealSack (bemenet: s): v1 virológus zsákot akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
3. StealCloak (bemenet: c): v1 virológus köpenyt akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
4. StealAxe (bemenet: ax): v1 virológus baltát akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
5. StealAminoacid (bemenet: a): v1 virológus aminoacidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás megghiúsul.
6. StealNukleotid (bemenet: n): v1 virológus nukleotidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás megghiúsul.

**Parancs: 60**

**Leírás:** A StealFromAminoGuy teszt létrehoz egy Game és egy Field objektumot, majd ezeket átadja a két virológus v1, és v2 objektumnak a konstruktora paramétereként. A v2 konstruktorában még egy Inventory objektumot is átadunk, melyet egy elemű Aminoacid listával hozunk létre. A v1 virológus ellop egy tárgyat (anyagot vagy eszközt) v2-től.

A v1 virológus ellop egy tárgyat (anyagot vagy eszközt) v2-től.

**Opciók:**

1. StealGloves (bemenet: g): v1 virológus kesztyűt akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
2. StealSack (bemenet: s): v1 virológus zsákot akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
3. StealCloak (bemenet: c): v1 virológus köpenyt akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
4. StealAxe (bemenet: ax): v1 virológus baltát akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
5. StealAminoacid (bemenet: a): v1 virológus aminoacidot akar lopni v2-től, melyet megtalál az inventory-jában, így az aminoacid átkerül v1 inventoryjába.
6. StealNukleotid (bemenet: n): v1 virológus nukleotidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás megghiúsul.

**Parancs: 61**

**Leírás:** **StealFromNukleoGuy:** A teszt létrehoz egy Game (g), egy Field (f), két Virologist (v1 és v2), egy Nukleotid objektumot, amik közül a v2 bénultva jön létre, egy nukleotiddal a felszerelésében. Ezek után v1 megpróbálja meglopni v2-t.

**Opciók:** -

**Parancs: 62**

**Leírás:** **StealFromNukleoGuyButIAMFull:** A teszt létrehoz egy Game(g), Field(f), Virologist(v1, v2) objektumot. Ezután v1 inventoryját feltölti fullra nukleotidokkal, majd v1 megpróbál lopni v2-től, viszont ez nem sikerül mivel v1 inventoryja tele van.

**Opciók:** -

**Parancs: 63**

**Leírás:** **StealFromGuyWithGlovesEquipped:** A teszt létrehoz egy Game(g), Field(f), Virologist(v1, v2) objektumot. A v2-en vannak kesztyűk. Ezután v1 meglopja v2-t, a lopás sikeres.

**Opciók:** -

**Parancs: 64**

**Leírás:** **StealFromGuyWithGlovesInStorage:** A teszt létrehoz egy Game (g), egy Field (f), két Virologist (v1 és v2) és egy Glove objektumot. Ezek közül a v2 bénult állapotban jön létre, kesztyűvel, majd a v1 megpróbálja meglopni a v2-t.

**Opciók:** -

**Parancs: 65**

**Leírás:** A **StealFromGuyWithCloakEquipped** teszt létrehoz egy Game és egy Field objektumot, majd ezeket átadja a két virológus v1, és v2 objektumnak a konstruktora paramétereiként. A v2 konstruktorában még egy Inventory objektumot is átadunk, melyet egy elemű Cloak listával hozunk létre. A v1 virológus ellop egy tárgyat (anyagot vagy eszközt) v2-től.

**Opciók:**

1. StealGloves (bemenet: g): v1 virológus kesztyűt akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
2. StealSack (bemenet: s): v1 virológus zsákot akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
3. StealCloak (bemenet: c): v1 virológus köpenyt akar lopni v2-től, melyet megtalál rajta felvéve, így azt leveszi róla, és átrakja a saját inventory-jába.
4. StealAxe (bemenet: ax): v1 virológus baltát akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
5. StealAminoacid (bemenet: a): v1 virológus aminosavat akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás megghiúsul.
6. StealNukleotid (bemenet: n): v1 virológus nukleotidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás megghiúsul.

**Parancs: 66**

**Leírás:** A **StealFromGuyWithCloakInStorage** teszt létrehoz egy Game és egy Field objektumot, majd ezeket átadja a két virológus v1, és v2 objektumnak a konstruktora paramétereiként. A v1 virológus ellop egy tárgyat (anyagot vagy eszközt) v2-től.

**Opciók:**

1. StealGloves (bemenet: g): v1 virológus kesztyűt akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
2. StealSack (bemenet: s): v1 virológus zsákot akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
1. StealCloak (bemenet: c): v1 virológus köpenyt akar lopni v2-től, melyet a virológusra felvett tárgyként nem, de az inventory-jában megtalálja, és így a köpeny átkerül v2 inventory-jába.
3. StealAxe (bemenet: ax): v1 virológus baltát akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás megghiúsul.
4. StealAminoacid (bemenet: a): v1 virológus aminosavat akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás megghiúsul.
5. StealNukleotid (bemenet: n): v1 virológus nukleotidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás megghiúsul.

**Parancs: 67**

**Leírás: StealFromGuyWithEmptySackEquipped:** A teszt létrehoz egy Game (g), egy Field (f), és két Virologist (v1 és v2) objektumot. A v2 felszerelését megtölti aminosavakkal és nukleotidokkal, és egy üres táskával. Ezek után v2 felveszi a táskát és a paralyzed értéke igazra változik. Majd v1 megpróbál v2-től lopni.

**Opciók:** -

**Parancs: 68**

**Leírás: StealFromGuyWithStuffInSackEquipped:** A teszt létrehoz egy Game(g), Field(f) Virologist(v1, v2), Inventory(inv) objektumot. Az inv-et feltölti egy zsákkal és nukleotidokkal, inv a v2 inventoryja. A v2 felveszi a zsákot, majd hozzáad az inv-hez aminoacidokkal, v2-t beállítja bénultnak és v1 meglopja v2-t, a lopás sikeres.

**Opciók:** -

**Parancs: 69**

**Leírás: A StealFromGuyWithSackInStorage** teszt létrehoz egy Game és egy Field objektumot, majd ezeket átadja a két virológus v1, és v2 objektumnak a konstruktora paramétereként. A v2 konstruktorában még egy Inventory objektumot is átadunk, melyet egy elemű Sack listával hozunk létre. A v1 virológus ellop egy tárgyat (anyagot vagy eszközt) v2-től.

**Opciók:**

2. StealGloves (bemenet: g): v1 virológus kesztyűt akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás meghiúsul.
3. StealSack (bemenet: s): v1 virológus zsákot akar lopni v2-től, melyet a virológusra felvett tárgyként nem, de az inventory-jában megtalálja, és így a zsák átkerül v2 inventory-jába.
4. StealCloak (bemenet: c): v1 virológus köpenyt akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás meghiúsul.
5. StealAxe (bemenet: ax): v1 virológus baltát akar lopni v2-től, melyet sem rajta, sem pedig az inventory-jában nem talál meg, így a lopás meghiúsul.
6. StealAminoacid (bemenet: a): v1 virológus aminoacidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás meghiúsul.
7. StealNukleotid (bemenet: n): v1 virológus nukleotidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás meghiúsul.

**Parancs: 70**

**Leírás: A TakeOffAxe** teszt létrehoz egy game, field és inventory objektumot. Az inventory-ban elhelyez egy Axe objektumot az inventory konstruktorának paraméterként átadva. Ezután létrehoz egy virológus objektumot a game, field, és inventory objektumokat átadva neki. A virológus felveszi a baltát (equipAxe()). Ezután a virológus leveszi a baltát. A rendszer megkérdezi a felhasználót, hogy el akarja-e dobni a baltát, és a választól függően a balta megsemmisül, vagy visszakerül az inventory-ba.

**Opciók:**

1. YesToDrop (bemenet: y): A virológus nem szeretné megőrizni a baltát, a balta levételkor megsemmisül.
2. NoToDrop: (bemenet: n): A virológus szeretné megőrizni a baltát, így az visszakerül az inventory-ba.

**Parancs: 71<ágenstípus>**

**Leírás: UseAgentOnVirologistWhoHasBearDance:** A teszt létrehoz egy Game(g), Field(f), Virologist(v1, v2), BearDance(bd) és egy bemenetnek megfelelő Immunity(i)/Oblivion(o)/VitusDance(vd)/Paralysis(p) objektumot. Ezután v1 megkapja az ágenst, majd v2-n beállítja a medvetáncot. A v1 rákenni az ágenst v2-re, ami nincs hatással v2-re.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 72**

**Leírás: InfectAllVirologistsOnFieldWhenArrive:** A teszt létrehoz egy Game (g), két Field (f1 és f2), három Virologist (v1, v2 és v3), egy Oblivion típusú (o), egy Paralysis típusú (p), egy VitusDance típusú



(v), egy Immunity típusú (i) és egy BearDance típusú (b) Agent objektumot. Az f1 és f2 mezők szomszédosak, a v1 kezdetben az f1, míg a v2 és v3 kezdetben az f2 mezőn voltak. A v1-et megfertőzi a b ágens, a v2-t az o, p, és v ágens, míg a v3-at az i típusú ágens. Ezek után a v1 átlép az f2 mezőre, ekkor az f2-n levő virológusokra megpróbálja rákenni a medvevírust.

**Opciók:** -

### **Parancs: 73**

**Leírás: MoveWhileBearDance:** A teszt létrehoz egy Game és 2 Field objektumot (f1, f2). Ezután ezeket beállítja szomszédoknak. Létrehozunk egy virológus (v1) objektumot, melynek konstruktorában átadjuk a game, és az f1 objektumokat, és a BearDancing paramétert true-ra állítjuk. A virológus megpróbál egy a felhasználó által irányított lépést végrehajtani, azonban mivel BearDance hat rá, a lépés sikertelen.

**Opciók:** -

### **Parancs: 74<ágenstípus>**

**Leírás: UseAgentOnVirologistWhoHasAxe:** A teszt létrehoz egy Game (g), egy Field (f), két Virologist (v1 és v2), 1 Axe (a), és egy bemenetnek megfelelő Immunity(i)/Oblivion(o)/VirusDance(vd)/Paralysis(p) objektumot. A v1 - megkapja az ágenst. A v1 felkeni v2-re az ágenst. A teszt további kimenetele a felhasználói inputtól függ.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VirusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

### **Parancs: 75<ágenstípus>**

**Leírás: UseAgentOnVirologistWhoHasAxeAndGloves:** A teszt létrehoz egy Game (g), egy Field (f), két Virologist (v1 és v2), 1 Axe (a), egy Gloves (glov) és egy, a paraméterként megadott típusú Agent objektumot. A v1 megkapja az ágenst, a v2 pedig megkapja az a és glov objektumokat, amiket fel is vesz. Ezek után v1 megpróbálja rákenni az ágenst v2-re. A teszt további kimenetele a felhasználói inputtól függ.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VirusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

### **Parancs: 76**

**Leírás: UseAgentOnVirologistWhoHasAxeGlovesAndCloak:** A teszt létrehoz egy Game és egy Field objektumot. Ezután létrehoz egy inventory objektumot, melynek átadjuk paraméterben egy 1 elemű Axe listát, egy 1 elemű Gloves listát és egy 1 elemű Cloak listát. Majd létrehozunk egy v1 és v2 virológust, mindkettőnek ugyanazon game és field objektumot konstruktorban átadva, illetve a v2-nek az inventory objektumot, v1-nek egy elemű (bénító ágenst tartalmazó) ágens listát is átadva. A v2 virológus felveszi magára az inventory-ból az Axe, a Gloves illetve a Cloak eszközöket. A v1 virológus megtámadja a v2 virológust a bénító ágenssel.

**Opciók:**

1. **Defense using axe (bemenet: a):** A v2 virológus használja a baltát, így az kicsorbul. A v1 virológus meghal.
2. **Defense using gloves (bemenet: g):** A v2 virológus nem használja a baltát, de használja a kesztyűt, így a kesztyű használhatóságának száma 1-el csökken, v2 virológus nem lesz béna. A v1 virológus lebénul.
3. **Defense using cloak (bemenet: c):** A v2 virológus nem használja sem a baltát, sem pedig a kesztyűt, de a rajta lévő köpeny kivédi az ágens hatását, a köpeny eltűnik. Egyik virológus sem lesz béna.
4. **Defense fail (bemenet: 0):** A v2 virológus nem használja a baltát és a kesztyűt, a köpenynek nem sikerül kivédeni az ágens hatását, így az megmarad, v2 lebénul.

**Parancs: 77<ágenstípus>**

**Leírás:** **UseAgentOnVirologistWhoHasParalysis:** A teszt létrehoz egy Game(g), Field(f), Virologist(v1, v2) és egy bemenetnek megfelelő Immunity(i)/Oblivion(o)/VitusDance(vd)/Paralysis(p) objektumot. A v1 megkapja az ágenst, v2-t beállítja bénultra, majd v1 megfertőzi v2-t az ágenssel, aki nem tud védekezni, mivel le van bénulva.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 78<ágenstípus>**

**Leírás:** **UseAgentOnVirologistWhoHasVitusDance:** A teszt létrehoz egy Game (g), egy Field (f), két Virologist (v1 és v2), egy VitusDance (v) és egy, a paraméterként megadott típusú Agent objektumot. A v2 megfertőződik a v ágenssel, v1 pedig megkapja a paraméternek megfelelő ágenst. Ezek után v1 megpróbálja rákenni v2-re az ágenst, ami sikeres, mert v2 nem tud védekezni.

**Opciók:** -

**Parancs: 79**

**Leírás:** **UseAgetnetOnVirologistWhoHasBearDance:** A teszt létrehoz egy Game és egy Field objektumot. Majd létrehozunk egy v1 és v2 virológust, mindkettőnek ugyanazon game és field objektumot konstruktorban átadva, illetve a v2-nek a BearDancing paraméterét true-ra állítjuk, a v1-nek egy elemű (bénító ágenst tartalmazó) ágens listát is átadunk. A v1 virológus megtámadja a v2 virológust a bénító ágenssel. A v2 virológusra már ágens hat, így nem kapja meg a bénító ágens hatását.

**Opciók:** -

**Parancs: 80**

**Leírás:** **GetBearDanceInfectionWhenArriveToLabWhenHasImmunity:** A teszt létrehoz egy Game (g), egy Field (f), egy Lab (l), egy Virologist (v1) és egy BearDance (b) objektumot. A f és l mezőket szomszédosnak állítja és a l BearInfected tulajdonsága igaz értéket kap. A v1 l-re lép, de nem fertőződik meg, mivel immunis.

**Opciók:** -

**Parancs: 81**

**Leírás:** **GetBearDanceInfectionWhenArriveToLabWhenHasVitusDance:** A teszt létrehoz egy Game (g), egy Field (f), egy Lab (l), egy Virologist (v1), egy VitusDance (v) és egy BearDance (b) Agent objektumot. A f és l mezőket szomszédosnak állítja és a l BearInfected tulajdonsága igaz értéket kap. A v1 megfertőződik a v ágenssel. Ezek után a v1 a l mezőre lép, és megfertőződik a medvevírussal.

**Opciók:** -

**Parancs: 82**

**Leírás:** **GetBearDanceInfectionWhenArriveToLabWhenHasImmunity:** A teszt létrehoz egy Game (g), egy Field (f), egy Lab (l), egy Virologist (v1), egy VitusDance (v) és egy BearDance (b) Agent objektumot. A f és l mezőket szomszédosnak állítja és a l BearInfected tulajdonsága igaz értéket kap. A v1 nem fertőződik meg a v ágenssel, mert immunitás védi.

**Opciók:** -

**Parancs: 83<fertőzés esélye>**

**Leírás:** **GetBearDanceInfectionWhenArriveToLabWhenHasCloak:** A teszt létrehoz egy Game (g), egy Field (f), egy Lab (l), egy Virologist (v1) és egy BearDance (b), Cloak(c) objektumot. A f és l mezőket szomszédosnak állítja és a l BearInfected tulajdonsága igaz értéket kap. A v1 l-re lép a megfertőződése a bemenettől függ.

**Opciók:**

1. A v1 megfertőződik (fertőzés esélye: 1).
2. A v1 nem fertőződik meg (fertőzés esélye: 0).

**Parancs: 84**

**Leírás:** **GetBearDanceInfectionWhenArriveToLabWhenHasGloves:** A teszt létrehoz egy Game (g), egy Field (f), egy Lab (l), egy Virologist (v), egy Gloves (glov), és egy BearDance típusú Agent objektumot.

A v kezdőhelye a f. A l BearInfected tulajdonsága pedig igaz értékű. Ezek után a v megkapja a glove objektumot és fel is veszi, majd rálép az l mezőre. Ekkor megfertőződik a b ágenssel, a kesztyűjét nem tudja felhasználni védekezésre.

**Opciók:** -

#### **Parancs: 85**

**Leírás: GetBearDanceInfectionWhenArriveToLabWhenHasNoEqAndNoAgent:** A teszt létrehoz egy Game (g), egy Field (f), egy Lab (l), egy Virologist (v1) és egy BearDance (b) Agent objektumot. A f és l mezőket szomszédosnak állítja és a l BearInfected tulajdonsága igaz értéket kap. Ezek után a v1 az l mezőre lép, és megfertőződik a medvevírussal.

**Opciók:** -

#### **Parancs: 86**

**Leírás: StepBearDance:** A teszt létrehoz egy Game (g), egy Field (f), egy Virologist (v1) és egy BearDance(bd) objektumot. Az ágenssel megfertőzi a virológust, majd ameddig tart az ágens hatása, addig lépteti az ágenst (kifejti annak hatását a virológusra.)

**Opciók:** -

## 7.1.2 Kimeneti nyelv

#### **Parancs: 0**

A teszt kimenete egy lista, amely tartalmazza a virológus felhasználható ágens készletének neveit.

**Elvárt kimenet opciók:**

1. Ha a **bemenet** egy e karakter volt, akkor az elvárt **kimenet** egy üres lista
2. Ha a **bemenet** egy f karakter volt, akkor az elvárt **kimenet** egy lista, ami a "paralysis" Stringet tartalmazza, mint egyetlen elem.

Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvárt és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.

#### **Parancs: 1**

**Kimenet:** VitusDance created.

**Opciók:** -

#### **Parancs: 2**

**Kimenet:** Immunity created.

**Opciók:** -

#### **Parancs: 3**

**Kimenet:** Gloves taken off,

**Opciók:** -

#### **Parancs: 4**

**Kimenet:** Cloak taken off.

**Opciók:** -

#### **Parancs: 5**

A teszt kimenete egy lista, amely 6 darab számot tartalmaz. Az **első szám** a virológuson található zsákok listájának mérete, miután a virológus felvette az inventory-ból a zsákot. A **második szám** ezzel megegyező időpontban a virológus inventory-jában található zsákok listájának mérete. A **harmadik szám** a virológus inventory-jában található nyersanyagok (material) száma, amikor (a zsák felvétele után) opcionálisan feltöltöttük azt. A **negyedik szám** a virológuson lévő zsákok listája, miután a virológus levette magáról, és visszatette az inventory-ba a zsákot. Az **ötödik szám** a virológus inventory-jában található zsákok listájának mérete ugyanebben az időpontban. Végül a **hatodik szám** a virológus inventory-jában található nyersanyagok (material) száma, amikor (a zsák levétele után) opcionálisan már eldobásra került az inventory kapacitását meghaladó nyersanyagmennyiség.

**Elvárt kimenet opciók:**

1. Ha a **bemenet** egy e karakter volt, akkor az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **1, 0, 20, 0, 1 20**

2. Ha a **bemenet** egy **f** karakter volt, akkor az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **1, 0, 22, 0, 1, 20**

Minden más esetben a **teszt hibára fut**. Ekkor a **teszt kiírja a bemenet alapján elvárt és a valós kimenetet**. A **teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér**. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.

**Parancs: 6**

**Kimenet:** Gloves dropped.

**Opciók:** -

**Parancs: 7**

**Kimenet:** Cloak dropped.

**Opciók:** -

**Parancs: 8**

A **teszt kimenete egy lista, amely 8 darab számot tartalmaz**. Az **első szám** a virológuson található zsákok listájának mérete, miután a virológus felvette az inventory-ból a zsákot. A **második szám** a virológus inventory-jában található zsákok listájának mérete, miután az inventory-ba került még egy kesztyű a zsák virológusra való felvétele után. A **harmadik szám** ezzel megegyező időpontban az inventoryban található kesztyűk listájának mérete. A **negyedik szám** a virológus inventory-jában található nyersanyagok (material) száma, amikor (a zsák felvétele után) opcionálisan feltöltöttük azt. Az **ötödik szám** a virológuson lévő zsákok listája, miután a virológus levette magáról, és eldobta a zsákot. A **hatodik szám** a virológus inventory-jában található zsákok listájának mérete ugyanebben az időpontban. A **hetedik szám** a virológus inventory-jában található kesztyűk listájának mérete ugyanebben az időpontban. Végül a **nyolcadik szám** a virológus inventory-jában található nyersanyagok (material) száma, amikor (a zsák levétele után) opcionálisan már eldobásra került az inventory kapacitását meghaladó nyersanyagmennyiség.

**Elvárt kimenet opciók:**

1. Ha a **bemenet** egy **e** karakter volt, akkor az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **1, 0, 5, 20, 0, 0, 5, 20**
2. Ha a **bemenet** egy **f** karakter volt, akkor az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **1, 0, 5, 22, 0, 0, 5, 20**

Minden más esetben a **teszt hibára fut**. Ekkor a **teszt kiírja a bemenet alapján elvárt és a valós kimenetet**. A **teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér**. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.

**Parancs: 9 <ágenstípus>**

**Kimenet:** Attacked virologist paralyzed. Can not perform action.

**Opciók:** -

**Parancs: 10<ágenstípus>**

**Kimenet:** Virologist paralyzed. Can not perform action.

**Opciók:** -

**Parancs: 11**

A **teszt kimenete egy számokból álló lista, első eleme** megadja a **v1 virológus felhasználható ágenseit** tartalmazó listájának elemszámát, második eleme pedig 0 vagy 1 az alapján, hogy a **v2 virológus bénult-e** (1) vagy nem (0). Az **elvárt kimenet** egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **0, 1**

Minden más esetben a **teszt hibára fut**. Ekkor a **teszt kiírja az elvárt és a valós kimenetet**.

**Parancs: 12<ágenstípus>**

**Kimenet:** Attack successful.

**Opciók:** -

**Parancs: 13<ágenstípus>**

**Kimenet:** Attack successful.

**Opciók:** -

**Parancs: 14**

A **teszt kimenete egy számokból álló lista, első eleme** 0 vagy 1 értéket vehet fel annak függvényében, hogy a virológus bénult-e (0 = false, 1 = true), a **második eleme** pedig a virológuson található köpenyek listájának elemszáma. A **harmadik elem** megadja a v1 virológus felhasználható ágensei listájának elemszámát.

**Elvárt kimenet opciók:**

1. Ha a **bemenet 1** volt, az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **0, 0, 0**
2. Ha a **bemenet 0** volt, az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **1, 1, 0**

Minden más esetben a **teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvárt és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.**

**Parancs: 15<ágenstípus>**

**Kimenet:** Attacked virologist has immunity. Can not perform action.

**Opciók:** -

**Parancs: 16<ágenstípus>**

**Kimenet:** v1 attacked you with <ágenstípus>.

Wanna use Gloves? (Y/N).

**Leírás:** Megkérdezi a v2 felhasználót, hogy szeretné-e használni a kesztyűjét a védekezésre.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 17**

A **teszt kimenete egy számokból álló lista. Első eleme** a v2 virológus kesztyűje használhatóságának mennyiségét reprezentálja. Amennyiben a v2 virológus nem rendelkezik kesztyűvel, ez a szám 0. A **második eleme** a v2 virológus által viselt köpenyek számát reprezentálja. A **harmadik eleme** 0 vagy 1 az alapján, hogy a v2 virológus bénult-e (1) vagy nem (0). A **negyedik eleme** szintén 0 vagy 1, viszont ez a v1 virológus bénaságának meglétét (1) vagy hiányát (0) jelöli. Az **ötödik elem** megadja a v1 virológus felhasználható ágensei listájának elemszámát.

**Elvárt kimenet opciók:**

1. Ha a **bemenet 0** volt, az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **3, 1, 1, 0, 0**
2. Ha a **bemenet g** volt, az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **2, 1, 0, 1, 0**
3. Ha a **bemenet c** volt, az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **3, 0, 0, 0, 0**

Minden más esetben a **teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvárt és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.**

**Parancs: 18<ágenstípus>**

**Kimenet:** v1 attacked you with <ágenstípus>.

Wanna use Gloves? (Y/N).

v2 attacked you with <ágenstípus>.

Wanna use Gloves? (Y/N).

**Leírás:** Megkérdezi a v2 felhasználót, hogy szeretné-e használni a kesztyűjét a védekezésre. Ha igennel válaszol(v2), akkor megkérdezi v1 felhasználót, hogy szeretné-e használni a kesztyűjét védekezésre.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).

2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 19<ágenstípus>**

**Kimenet:** v1 attacked you with <ágenstípus>.

Wanna use Gloves? (Y/N).

**Leírás:** Megkérdezi a v2 felhasználót, hogy szeretné-e használni a kesztyűjét a védekezésre.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 20**

**A teszt kimenete egy számokból álló lista. Első eleme** a v2 virológus kesztyűje használhatóságának mennyiségét reprezentálja. Amennyiben a v2 virológus nem rendelkezik kesztyűvel, ez a szám 0. **A második eleme** 0 vagy 1 az alapján, hogy a v2 virológus bénult-e (1) vagy nem (0). **A harmadik eleme** szintén 0 vagy 1, viszont ez a v1 virológus bénaságának meglétét (1) vagy hiányát (0) jelöli. **A negyedik eleme** megadja a v1 virológus felhasználható ágensei listájának elemszámát.

**Elvart kimenet opciók:**

1. Ha **a bemenet 1** volt, az elvart kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **2, 0, 0, 0**
2. Ha **a bemenet 0** volt, az elvart kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **3, 1, 0, 0**

**Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvart és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.**

**Parancs: 21 <ágenstípus>**

**Kimenet:** Successfully used <ágenstípus> on self.

- Opciók:**
1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
  2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
  3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
  4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 22<ágenstípus>**

**Kimenet:** Successfully used <ágenstípus> on self.

**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 23**

**A teszt kimenete egy számokból álló lista. Első eleme** a virológus kesztyűje használhatóságának mennyiségét reprezentálja. Amennyiben a virológus nem rendelkezik kesztyűvel, ez a szám 0. **A második eleme** a virológus által viselt köpenyek számát reprezentálja. **A harmadik eleme** 0 vagy 1 az alapján, hogy a virológus immunitás ágens hatása alatt áll-e (1) vagy nem (0). **A negyedik elem** megadja a virológus felhasználható ágensei listájának elemszámát. **Az elvart kimenet** egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **3, 1, 1, 0**

**Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvart és a valós kimenetet.**

**Parancs: 24 <ágenstípus>****Kimenet:** Virologist has no agnets. Can not perform action.**Opciók:** -**Parancs: 25****Kimenet:** Virologist paralyzed. Can not perform action.**Opciók:** -**Parancs: 26**

**A teszt kimenete egy számokból álló lista. Első eleme 0 vagy 1 az alapján, hogy a virológus immunitás ágens hatása alatt áll-e (1) vagy nem (0). A második eleme 0 vagy 1 az alapján, hogy a virológus vitustánc ágens hatása alatt áll-e (1) vagy nem (0). A harmadik elem megadja a virológus felhasználható ágensei listájának elemszámát. Az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: 1, 0, 0**

**Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvárt és a valós kimenetet.**

**Parancs: 27****Kimenet:** Successfully used Oblivion on self.**Opciók:** -**Parancs: 28****Kimenet:** Successfully used VitusDance on self.**Opciók:** -**Parancs: 29**

**A teszt kimenete egy számokból álló lista. Első eleme 0 vagy 1 az alapján, hogy a virológus immunitás ágens hatása alatt áll-e (1) vagy nem (0). A második elem megadja a virológus felhasználható ágensei listájának elemszámát. Az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: 1, 0**

**Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvárt és a valós kimenetet.**

**Parancs: 30****Kimenet:** Successfully used Paralysis on self.**Opciók:** -**Parancs: 31****Kimenet:** Virologist is under vitusdance effect. Can not perform action.**Opciók:** -**Parancs: 32**

**A teszt kimenete azon mező neve (f1 vagy f2), amelyen a virológus áll.**

**Az elvárt kimenet: f1**

**Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvárt és a valós kimenetet.**

**Parancs: 33****Kimenet:** Successfully moved to another field.**Opciók:** -**Parancs: 34****Kimenet:** Virologist paralyzed. Can not perform action.**Opciók:** -**Parancs: 35**

**A teszt kimenete egy szám, amely a virológus által megtanult genetikai kódok listájának elemszámát reprezentálja. Az elvárt kimenet 0. Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvárt és a valós kimenetet.**

**Parancs: 36<ágenstípus>****Kimenet:** New code learnt.**Opciók:** -

**Parancs: 37**

**Kimenet:** Code already known.

**Opciók:** -

**Parancs: 38**

**Kimenet:** Game ended.

**Parancs: 39<anyagtípus>**

**Kimenet:** Virologist paralyzed. Can not perform action.

**Opciók:** -

**Parancs: 40**

**Kimenet:** Successfully picked up nucleotid.

**Opciók:** -

**Parancs: 41**

A teszt kimenete egy szám, amely a virológus inventory-jában található Aminoacid lista elemszámát reprezentálja. **Elvárt kimenet: 1.** Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvárt és a valós kimenetet.

**Parancs: 42<anyagtípus>**

**Kimenet:** Virologist's inventory full. Can not perform action.

**Opciók:** -

**Parancs: 43**

**Kimenet:** Virologist paralyzed. Can not perform action.

**Opciók:** -

**Parancs: 44**

A teszt kimenete egy szám, amely a virológus inventory-jában található Gloves lista elemszámát reprezentálja. **Elvárt kimenet: 1.** Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvárt és a valós kimenetet.

**Parancs: 45**

**Kimenet:** Cloak picked up.

**Opciók:** -

**Parancs: 46**

**Kimenet:** Sack picked up.

**Opciók:** -

**Parancs: 47**

A teszt kimenete egy lista, melynek első eleme a virológus inventory-jában található Gloves lista elemszámát reprezentálja az utolsó (6.) kesztyű felvétele előtt, a második eleme pedig ugyanezen lista elemszáma az utolsó kesztyű felvétele után. **Elvárt kimenet: 5, 5.** Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvárt és a valós kimenetet.

**Parancs: 48**

**Kimenet:** Oblivion created.

**Opciók:** -

**Parancs: 49**

**Kimenet:** Virologist is under vitusdance effect. Can not perform action.

**Opciók:** -

**Parancs: 50**

A teszt kimenete azon mező neve (f1 vagy f2), amelyen a virológus áll.



Az elvárt kimenet: f1

Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvárt és a valós kimenetet.

**Parancs: 51**

**Kimenet:** Equipment taken off.

**Opciók:**

**Parancs: 52**

**Kimenet:** Paralyzed stepped.

**Opciók:** -

**Parancs: 53**

A teszt kimenete több lista (n létrehozott mező esetén n+1 db lista), az első n lista azon mezők neveit tartalmazza, ami az n.-ként létrehozott mező szomszédai, majd az n+1. lista tartalmazza a kiinduló mező nevét, majd minden olyan mező nevét, ahova a virológus lépett.

**Elvárt kimenet opciók:**

1. Ha a **bemenet 0**, az elvárt kimenet 3 lista. Az első lista egyetlen eleme: f2, a második lista egyetlen eleme: f1, a harmadik lista rendre a következő elemeket tartalmazza: **f1, f2, f1, f2, f1, f2**

**Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvárt és a valós kimenetet.**

2. Ha a **bemenet 1**, az elvárt kimenetet úgy ellenőrizzük, hogy az adott mező szomszédossági listájában megnézzük, hogy szerepel-e a következő mező neve, illetve azt is, hogy két egymás után következő mező mindig eltérő-e. Amennyiben bármely lépésben bármely feltétel nem teljesül, a teszt hibára fut. **Két féle hibaüzenet lehetséges: 1. Step to non neighbour field detected during VitusDance. 2. No step was made during VitusDance.**

**A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.**

**Parancs: 54**

**Kimenet:** Oblivion stepped.

**Opciók:** -

**Parancs: 55**

**Kimenet:** Immunity stepped.

**Opciók:** -

**Parancs: 56**

Kimenet 0 vagy 1 az alapján, hogy a virológus immunitás alatt áll-e (1) vagy nem (0).

**Elvárt kimenet: 0**

**Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja az elvárt és a valós kimenetet.**

**Parancs: 57**

**Kimenet:** Stealing successful.

**Opciók:** -

**Parancs: 58**

**Kimenet:** Virologist paralyzed. Can not perform action.

**Opciók:** -

**Parancs: 59**

**Kimenet:** Egy számlista, a v1 és v2 virológus összes viselt eszközének listájának mérete(Axe, Gloves, Sack, Cloak), és inventory-jában található összes material (Aminoacid, Nukleotid) és Equipment (Axe, Gloves, Sack, Cloak) listájának mérete. Először v1 tulajdonságai, majd v2 tulajdonságai.

**Elvárt kimenet opciók:**

**Mivel minden bemenetre a lopás megghiúsul, ezért az elvárt kimenet listájának elemei minden esetben: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0**

**Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvárt és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején**

**input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.**

**Parancs: 60**

**Kimenet:** Egy számlista, a v1 és v2 virológus összes viselt eszközének listájának mérete (Axe, Gloves, Sack, Cloak), és inventory-jában található összes material (Aminoacid, Nukleotid) és Equipment (Axe, Gloves, Sack, Cloak) listájának mérete. Először v1 tulajdonságai, majd v2 tulajdonságai.

**Elvárt kimenet opciók:**

**1. Ha a bemenet a: A várt kimenet listájának elemei rendre: 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0**

Mivel minden más bemenetre a lopás megghiúsul, ezért az elvárt kimenet lista elemeinek várt értékei minden esetben rendre: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0

Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvárt és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.

**Parancs: 61**

**Kimenet:** Steal successful.

**Opciók:** -

**Parancs: 62**

**Kimenet:** Stealing Nukleotid unsuccessful, Inventory full.

**Opciók:** -

**Parancs: 63**

**Kimenet:** Stealing successful.

**Opciók:** -

**Parancs: 64**

**Kimenet:** Steal successful.

**Opciók:** -

**Parancs: 65**

**Kimenet:** Egy számlista, a v1 és v2 virológus összes viselt eszközének listájának mérete (Axe, Gloves, Sack, Cloak), és inventory-jában található összes material (Aminoacid, Nukleotid) és Equipment (Axe, Gloves, Sack, Cloak) listájának mérete. Először v1 tulajdonságai, majd v2 tulajdonságai.

**Elvárt kimenet opciók:**

**2. Ha a bemenet c: A várt kimenet listájának elemei rendre: 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0**

Mivel minden más bemenetre a lopás megghiúsul, ezért az elvárt kimenet lista elemeinek várt értékei minden esetben rendre: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0

Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvárt és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.

**Parancs: 66**

**Kimenet:** Egy számlista, a v1 és v2 virológus összes viselt eszközének listájának mérete (Axe, Gloves, Sack, Cloak), és inventory-jában található összes material

(Aminoacid, Nukleotid) és Equipment (Axe, Gloves, Sack, Cloak) listájának mérete.  
Először v1 tulajdonságai, majd v2 tulajdonságai.

**Elvárt kimenet opciók:**

**3. Ha a bemenet c: A várt kimenet listájának elemei rendre: 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0**

Mivel minden más bemenetre a lopás megghiúsul, ezért az elvárt kimenet lista elemeinek várt értékei minden esetben rendre: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1

Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvárt és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.

**Parancs: 67**

**Kimenet:** Steal successful.

**Opciók:** -

**Parancs: 68**

**Kimenet:** Stealing successful.

**Opciók:**

**Parancs: 69**

**Kimenet:** Egy számlista, a v1 és v2 virológus összes viselt eszközének listájának mérete (Axe, Gloves, Sack, Cloak), és inventory-jában található összes material (Aminoacid, Nukleotid) és Equipment (Axe, Gloves, Sack, Cloak) listájának mérete.  
Először v1 tulajdonságai, majd v2 tulajdonságai.

**Elvárt kimenet opciók:**

**4. Ha a bemenet c: A várt kimenet listájának elemei rendre: 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0**

Mivel minden más bemenetre a lopás megghiúsul, ezért az elvárt kimenet lista elemeinek várt értékei minden esetben rendre: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0

Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvárt és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.

**Parancs: 70**

**Kimenet:** Egy számlista, melynek első eleme megadja a virológus által viselt balták mennyiségét, második eleme megadja a virológus inventory-jában található balták mennyiségét.

**Elvárt kimeneti opciók:**

1. Ha a bemenet y, a kimenet egy lista, melynek elemei rendre: 0,0

2. Ha a bemenet n, a kimenet egy lista, melynek elemei rendre: 0, 1

Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvárt és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.

**Parancs: 71<ágenstípus>**

**Kimenet:** Attacked virologist infected with BearDance, attack unsuccessful.

**Opciók:** -

**Parancs: 72****Kimenet:** Infected v2 with BearDance.**Opciók:** -**Parancs: 73****Kimenet:** Virologist is infected with BearDance, can not perform action.**Parancs: 74<ágenstípus>****Kimenet:** Attack successful.**Opciók:** -**Parancs: 75<ágenstípus>****Kimenet:** v1 attacked you with <ágenstípus>. Wanna use Axe? (Y/N).

Wanna use Gloves? (Y/N).

**Leírás:** Megkérdezi a v2 felhasználót, hogy szeretné-e a fejszéjét használni. Ha nemmel válaszolt, akkor megkérdezi, hogy szeretné-e használni a kesztyűjét.**Opciók:**

1. A felkenni kívánt ágens egy Oblivion (ágenstípus: o).
2. A felkenni kívánt ágens egy Paralysis (ágenstípus: p).
3. A felkenni kívánt ágens egy VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

**Parancs: 76<ágenstípus>**

**A teszt kimenete egy számokból álló lista. Első eleme 0 vagy egy és azt reprezentálja, hogy a v2 virológuson lévő balta ki van-e csorbulva (1) vagy nem (0). A második eleme a v2 virológus kesztyűje használhatóságának mennyiségét reprezentálja. Amennyiben a virológus nem rendelkezik kesztyűvel, ez a szám 0. A harmadik eleme a v2 virológus által viselt köpenyek számát reprezentálja. A negyedik eleme 0 vagy 1 az alapján, hogy a v2 virológus bénító ágens hatása alatt áll-e (1) vagy nem (0). Az ötödik eleme 0 vagy 1 az alapján, hogy a v1 virológus bénító ágens hatása alatt áll-e (1) vagy nem (0). A hatodik elem megadja a v1 virológus felhasználható ágensei listájának elemszámát. A hetedik elem megadja, hogy v1 virológus halott-e (1) vagy nem (0).**

**Elvárt kimeneti opciók:**

1. Ha a **bemenet a** volt, az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **1, 3, 1, 0, 0, 0, 1**
2. Ha a **bemenet g** volt, az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **0, 2, 1, 0, 1, 0, 0**
3. Ha a **bemenet c** volt, az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **0, 3, 0, 0, 0, 0, 0**
4. Ha a **bemenet 0** volt, az elvárt kimenet egy lista, mely a következő elemeket tartalmazza, rendre egymás után: **0, 3, 1, 1, 0, 0, 0**

Minden más esetben a teszt hibára fut. Ekkor a teszt kiírja a bemenet alapján elvárt és a valós kimenetet. A teszt nem várt bemenet esetén már a teszt legelején input error hibával visszatér. Ekkor kiírja a lehetséges bemeneteket, és az észlelt bemenetet.

**Parancs: 77<ágenstípus>****Kimenet:** Attacked virologist has immunity, attack unsuccessful.**Opciók:** -**Parancs: 78<ágenstípus>****Kimenet:** Attack successful.**Opciók:**

**Parancs: 79****Kimenet:** Virologist has BearDance, infection unsuccessful.**Opciók:** -**Parancs: 80****Kimenet:** Virologist has immunity, infection unsuccessful.**Opciók:** -**Parancs: 81****Kimenet:** You are affected by BearDance.**Opciók:** -**Parancs: 82****Kimenet:** Virologist has immunity, infection unsuccessful.**Opciók:** -**Parancs: 83<fertőzés veszélye>****Kimenet:** Virologist infected with BearDance/Virologist uninfected.**Opciók:** 1. A virológus megfertőződik (fertőzés esélye: 1).  
2. A virológus nem fertőződik meg (fertőzés esélye: 0).**Parancs: 84****Kimenet:** You are affected by BearDance.**Opciók:** -**Parancs: 85****Kimenet:** You are affected by BearDance.**Opciók:** -**Parancs: 86****Kimenet:** BearDance stepped.**Opciók:** -**7.2 Összes részletes use-case**

Use-case neve	Create Map
Rövid leírás	A játék indításáért és inicializásáért felelős use-case.
Aktorok	Rendszer
Forgatókönyv	1. Létrehozza a várost és minden benne lévő elemet.

Use-case neve	Move
Rövid leírás	A játékos irányítja a karakter a városban.
Aktorok	Felhasználó
Forgatókönyv	1. Szomszédos mezőre lép a játékos.

<b>Use-case neve</b>	Equip Sack
<b>Rövid leírás</b>	A játékos felvesz magára egy adott eszközt.
<b>Aktorok</b>	Felhasználó
<b>Forgatókönyv</b>	1. Felvesz egy zsákot, amely megnöveli az anyagtároló kapacitását az <i>Inventory</i> -jának.

<b>Use-case neve</b>	Equip Cloak
<b>Rövid leírás</b>	A játékos felvesz magára egy adott eszközt.
<b>Aktorok</b>	Felhasználó
<b>Forgatókönyv</b>	1. Felvesz egy köpenyt, amely megakadályozhatja, hogy egy másik virológus ágenst kenjen rá.

<b>Use-case neve</b>	Equip Gloves
<b>Rövid leírás</b>	A játékos felvesz magára egy adott eszközt.
<b>Aktorok</b>	Felhasználó
<b>Forgatókönyv</b>	1. Felvesz egy kesztyűt, amely visszadobja az éppen rákent ágenst.

<b>Use-case neve</b>	Equip Axe
<b>Rövid leírás</b>	A játékos felvesz magára egy adott eszközt.
<b>Aktorok</b>	Felhasználó
<b>Forgatókönyv</b>	1. Felvesz egy baltát, mellyel képes megölni más játékosokat.

<b>Use-case neve</b>	Create Agent
<b>Rövid leírás</b>	A játékos elkészít egy ágenst.
<b>Aktorok</b>	Felhasználó

<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos elkészít egy adott ágens, melyhez rendelkezik a megfelelő mennyiségű alapanyaggal.</li> <li>2. Az elkészült ágens eltárolja magánál egészen felhasználásig vagy az ágens idejének lejártáig.</li> </ol>
---------------------	---

<b>Use-case neve</b>	Attacks other Virologist
<b>Rövid leírás</b>	A játékos megtámad egy másik virológust.
<b>Aktorok</b>	Felhasználó
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos kiválaszt az ellenfele közül egyet, akivel azonos mezőn tartózkodik.</li> <li>2. A játékos az általa elkészített ágensek közül választ egyet.</li> <li>3. A játékos megtámadja a már kiválasztott játékost a már kiválasztott ágenssel.</li> </ol>

<b>Use-case neve</b>	Defense
<b>Rövid leírás</b>	A játékos megpróbál kivédeni egy támadást.
<b>Aktorok</b>	Felhasználó
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos látja, hogy megtámadják, így kiválasztja a felhelyezett eszközei közül, hogy melyikkel fog védekezni a támadással szemben.</li> </ol>

<b>Use-case neve</b>	Steals from paralyzed Virologist
<b>Rövid leírás</b>	A játékos meglop egy másik játékost.
<b>Aktorok</b>	Felhasználó
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos egy olyan mezőre lép, ahol tartózkodik lebénult virológus.</li> <li>2. A játékos kiválasztja, hogy szeretne lopni a lebénult játékostól.</li> <li>3. A játékos kiválasztja, hogy miket szeretne elvenni a másik játékostól.</li> </ol>

<b>Use-case neve</b>	Picks Up Equipment
<b>Rövid leírás</b>	A játékos felvesz eszközt az óvóhelyről.
<b>Aktorok</b>	Felhasználó.

<b>Forgatókönyv</b>	1. A játékos óvóhely típusú mezőre lép. 2. A játékos elvesz az óvóhelyről eszköz(öke)t.
---------------------	--

<b>Use-case neve</b>	Picks Up Material
<b>Rövid leírás</b>	A játékos felvesz anyagot a
<b>Aktorok</b>	Felhasználó.
<b>Forgatókönyv</b>	1. A játékos raktár típusú mezőre lép. 2. A játékos elvesz a Raktárból anyago(ka)t.

<b>Use-case neve</b>	Learns GCode
<b>Rövid leírás</b>	A játékos megtanul egy genetikai kódot.
<b>Aktorok</b>	Felhasználó.
<b>Forgatókönyv</b>	1. A játékos labor típusú mezőre lép. 2. A játékos megtanulja az adott laborban található genetikai kódot.

<b>Use-case neve</b>	Infected with Beardance from Lab
<b>Rövid leírás</b>	A játékos megfertőződik medvetáncsal.
<b>Aktorok</b>	Felhasználó, Rendszer.
<b>Forgatókönyv</b>	1. A játékos labor típusú mezőre lép. 2. A labor meg van fertőződve, így a játékos is megfertőződik vele.

<b>Use-case neve</b>	Infected with Beardance from others
<b>Rövid leírás</b>	A játékos megfertőződik medvetáncsal.
<b>Aktorok</b>	Felhasználó, Rendszer.
<b>Forgatókönyv</b>	1. Egy adott mezőre olyan játékos érkezik, aki meg van fertőződve medvetáncsal. 2. A játékos nem tud sikeresen védekezni ellene, így ő is megfertőződik vele.

<b>Use-case neve</b>	Takes off Sack to Inventory
----------------------	-----------------------------



<b>Rövid leírás</b>	A játékos levesz magáról egy adott eszközt.
<b>Aktorok</b>	Felhasználó.
<b>Forgatókönyv</b>	1. A játékos kiválasztja a zsák típusú eszközt, amely már aktiválva van rajta, és eltávolítja magáról, majd elhelyezi a zsákjába.

<b>Use-case neve</b>	Takes off Sack
<b>Rövid leírás</b>	A játékosról lekerül egy adott eszközt.
<b>Aktorok</b>	Felhasználó, Rendszer.
<b>Forgatókönyv</b>	1. A játékosról lekerül az eszköz lopás által.

<b>Use-case neve</b>	Takes off Cloak to Inventory
<b>Rövid leírás</b>	A játékos levesz magáról egy adott eszközt.
<b>Aktorok</b>	Felhasználó.
<b>Forgatókönyv</b>	1. A játékos kiválasztja a köpeny típusú eszközt, amely már aktiválva van rajta, és eltávolítja magáról, majd elhelyezi a zsákjába.

<b>Use-case neve</b>	Takes off Cloak
<b>Rövid leírás</b>	A játékosról lekerül egy adott eszközt.
<b>Aktorok</b>	Felhasználó, Rendszer.
<b>Forgatókönyv</b>	1. A játékosról lekerül az eszköz lopás vagy elhasználódás által.

<b>Use-case neve</b>	Takes off Gloves to Inventory
<b>Rövid leírás</b>	A játékos levesz magáról egy adott eszközt.
<b>Aktorok</b>	Felhasználó.
<b>Forgatókönyv</b>	1. A játékos kiválasztja a kesztyű típusú eszközt, amely már aktiválva van rajta, és eltávolítja magáról, majd elhelyezi a zsákjába.

<b>Use-case neve</b>	Takes off Gloves
<b>Rövid leírás</b>	A játékosról lekerül egy adott eszközt.
<b>Aktorok</b>	Felhasználó, Rendszer.
<b>Forgatókönyv</b>	1. A játékosról lekerül az eszköz lopás vagy elhasználódás által.

<b>Use-case neve</b>	Takes off Axe to Inventory
<b>Rövid leírás</b>	A játékos levesz magáról egy adott eszközt.
<b>Aktorok</b>	Felhasználó.
<b>Forgatókönyv</b>	1. A játékos kiválasztja a balta típusú eszközt, amely már aktiválva van rajta, és eltávolítja magáról, majd elhelyezi a zsákjába.

<b>Use-case neve</b>	Takes off Axe
<b>Rövid leírás</b>	A játékosról lekerül egy adott eszközt.
<b>Aktorok</b>	Felhasználó, Rendszer.
<b>Forgatókönyv</b>	1. A játékosról lekerül az eszköz lopás vagy elhasználódás által.

<b>Use-case neve</b>	Time steps
<b>Rövid leírás</b>	Az időzítőket lépteti.
<b>Aktorok</b>	Rendszer.
<b>Forgatókönyv</b>	1. Minden felvett, időzíthető objektum értékét egyel lépteti.

<b>Use-case neve</b>	Virologist steps
<b>Rövid leírás</b>	A virológus ágenseinek idejét csökkenti.
<b>Aktorok</b>	Rendszer.

<b>Forgatókönyv</b>	1. A virológus által készített, és még raktározott ágensek idejét lépteti, ameddig azok el nem használódnak egy virológusra, vagy az időzítőjük el nem éri a nullát.
---------------------	--

<b>Use-case neve</b>	Virologist dies
<b>Rövid leírás</b>	A virológus meghal.
<b>Aktorok</b>	Felhasználó, Rendszer.
<b>Forgatókönyv</b>	1. A virológussal végeznek balta eszköz használata által.

### 7.3 Tesztelési terv

<b>Teszt-eset neve</b>	
<b>Rövid leírás</b>	
<b>Teszt célja</b>	

<b>Teszt-eset neve</b>	
<b>Rövid leírás</b>	
<b>Teszt célja</b>	

### 7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

**7.5 Ütemterv**

Határidő	Feladat	Pont	Felelős
<b>febr. 28.</b>	Követelmény, projekt, funkcionalitás - beadás	10	Csia
<b>márc. 7.</b>	Analízis modell kidolgozása I. - beadás	20	Alpek
<b>márc. 16.</b>	Analízis modell kidolgozása II. - beadás	30	Marton
<b>márc. 21.</b>	Szkeleton tervezése - beadás	20	Litavec
<b>márc. 28.</b>	Szkeleton elkészítése - beadás	20	Csia
<b>ápr. 4.</b>	Prototípus koncepciója – beadás Skeleton bemutatás	20	Alpek
<b>ápr. 11.</b>	Részletes tervek - beadás	45	
<b>ápr. 25.</b>	Prototípus készítése, tesztelése	35	
<b>máj. 2.</b>	Grafikus változat tervei – beadás Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése	30	
<b>máj. 16.</b>	Grafikus változat készítése	40	
<b>máj. 18.</b>	Egységes dokumentáció - beadás és bemutatás Grafikus változat - beadás és a forráskód herculesre való feltöltése, és teljes házi bemutatás	30	

## 7.6 Napló

Kezdet	Időtartam	Résztevő(k)	Leírás
2022.04.01 20:30	5 óra	Alpek, Csia, Litavecz, Marton	<u>Értekezlet</u> : Új feladat átbeszélése, tervek elkészítése a feladat teljesítéséhez.
2022.04.02 13:00	2 óra	Alpek, Csia, Marton, Ruskó	<u>Értekezlet</u> : Bemeneti nyelv alapjainak leszögezése.
2022.04.02 13:00	2,5 óra	Alpek, Marton, Ruskó	<u>Feladat</u> : Bemeneti nyelvhez tervek legépélése. Feladatok szétosztása tagok között.
2022.04.02 14:00	1 óra	Csia	<u>Feladat</u> : Osztálydiagram rendbe tévése, dokumentum sablon elkészítése.
2022.04.02 16:00	6 óra	Litavecz	<u>Feladat</u> : Javított szekvencia diagrammok megrajzolása.
2022.04.03 10:00	8 óra	Csia	<u>Feladat</u> : Módosult metódusok legépélése, use-case-ek megírása, interfész általános leírás elkészítése.
2022.04.03 13:00	8 óra	Alpek	<u>Feladat</u> : Bemeneti, és az azokhoz tartozó kimeneti nyelv részek elkészítése. (Darabonként mindenki 29-et)
2022.04.03	8 óra	Marton	<u>Feladat</u> : Bemeneti, és az azokhoz tartozó kimeneti nyelv részek elkészítése. (Darabonként mindenki 29-et)
2022.04.03	8 óra	Ruskó	<u>Feladat</u> : Bemeneti, és az azokhoz tartozó kimeneti nyelv részek elkészítése. (Darabonként mindenki 29-et)
2022.04.03	2 óra	Litavecz	<u>Feladat</u> : A tesztelést támogató segéd- és fordítóprogramok rész megírása.
2022.04.03 21:00	3 óra	Teljes csapat	<u>Értekezlet</u> : Aszinkron módon megbeszélni a külön elkészült feladatokat, átbeszélni az esetleges hibákat.
2022.04.04 06:00	1 óra	Csia	<u>Feladat</u> : Dokumentum véglegesítése, lezárása, naplóbejegyzések lezárása.

**7.7 Százalékos teljesítés**

Név	Százalék
Alpek	20%
Csia	20%
Litavec	20%
Marton	20%
Ruskó	20%