**RÉSZLETES TERVEK**

40 – ZETA

Konzulens:

KOVÁCS BOLDIZSÁR

Csapattagok

|  |  |  |
| --- | --- | --- |
| Alpek Dávid Zsolt | C31X0F | alpek.david.zsolt@gmail.com |
| Csia Klaudia Kitti | HA5YCV | kitkat@sch.bme.hu |
| Litavecz Marcell | IPHJNB | marcell.litavecz@gmail.com |
| Marton Judit | M0MYIM | judit.marton@edu.bme.hu |
| Ruskó Eszter | H8IBRC | eszter@rusko.hu |

2022. 04. 13.

# Részletes tervek

## Osztályok és metódusok tervei

### Equipment

#### Felelősség

Az óvóhelyen megszerezhető eszközöket reprezentáló absztrakt osztály.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

* **String ID**: Az Equipment objektum egyedi azonosítóját tárolja. **Láthatóság:** -

#### Metódusok

* **void pickUp(Virologist v)**: absztrakt metódus. **Láthatóság:** +
* **String getID()**: Visszaadja az ID attribútum értékét. **Láthatóság:** +

### Gloves

#### Felelősség

Speciális típusú felszerelés, amellyel a virológus visszadobhatja a rákent ágenst a támadóra. Viszont ezt csak meghatározott alkalommal használhatja. Továbbá a kesztyű óvóhelyről inventoryba való felvételéért felelős.

#### Ősosztályok

Equipment

#### Interfészek

Nincs

#### Attribútumok

* **int Usable**: Tárolja, hogy kesztyű hány alkalommal használható. **Láthatóság:** -

#### Metódusok

* **void pickUp(Virologist v)**: Hozzáadja a kesztyűt a kapott virológus inventoryjának kesztyű listájához. **Láthatóság:** +
* **void use(Virologist vFr, Virologist vTo, Agent a)**: Meghívja a megtámadott virológus támadás függvényét a támadó virológussal és a használt ágenssel. **Láthatóság:** +
* **boolean wannaUse()**: Megkérdezi a játékost, hogy akarja-e használni a kesztyűt. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** +
* **void setUsable(int NumOfTimes)**: Beállítja a Usable attribútum értékét a paraméterként megkapott értékre. **Láthatóság:** +
* **int setUsable(int NumOfTimes)**: Visszaadja a Usable attribútum értékét. **Láthatóság:** +

### Sack

#### Felelősség

Speciális típusú felszerelés, amellyel a virológus inventoryjának maximális anyagtárolóképessége megnő. Továbbá a zsák óvóhelyről inventoryba való felvételéért felelős.

#### Ősosztályok

Equpment

#### Interfészek

Nincs

#### Attribútumok

* **int Capacity**: Tárolja, hogy a zsák mennyivel növeli meg a maximális anyagtárolóképességet, csak olvasható. **Láthatóság:** -

#### Metódusok

* **void pickUp(Virologist v)**: Hozzáadja a zsákot a kapott virológus inventoryjának zsák listájához. **Láthatóság:** +
* **int getCapacity()**: Visszaadja a Capacity attribútum értékét. **Láthatóság:** +

### Cloak

#### Felelősség

Speciális típusú felszerelés, amely egy adott százalékos aránnyal megakadályozza, hogy a köpenyt viselő virológusra hatással legyen a felkent ágens. Továbbá a köpeny óvóhelyről inventoryba való felvételéért felelős.

#### Ősosztályok

Equipment

#### Interfészek

Nincs

#### Attribútumok

* **double EvadePercentage**: Tárolja, hogy a köpeny milyen százalékkal védi ki a felkent ágens hatását. **Láthatóság:** -

#### Metódusok

* **void pickUp(Virologist v)**: Hozzáadja a köpenyt a kapott virológus inventoryjának köpeny listájához. **Láthatóság:** +
* **boolean evade()**: EvadePercentage százalékkal igazat, egyébként hamisat ad vissza. **Láthatóság:** +

### Axe

#### Felelősség

Speciális típusú felszerelés, amellyel a virológus meg tud ölni egy másik virológust. Továbbá a fejsze óvóhelyről inventoryba való felvételéért felelős.

#### Ősosztályok

Equipment

#### Interfészek

Nincs

#### Attribútumok

* **boolean Used:** Tárolja, hogy a fejszét használták-e már. **Láthatóság:** -

#### Metódusok

* **void pickUp(Virologist v)**: Hozzáadja a fejszét a kapott virológus inventoryjának fejsze listájához. **Láthatóság:** +
* **void use(Virologist vTo)**: Meghívja a kapott virológus die metódusát és igazra állítja a Used attribútumot. **Láthatóság:** +
* **boolean wannaUse():** **()**: Megkérdezi a játékost, hogy akarja-e használni a fejszét. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** +
* **boolean getUsed():** Visszaadja a Used attribútum értékét. **Láthatóság:** +

### Inventory

#### Felelősség

A virológus által óvóhelyről, raktárból, laborból megszerzett eszközöket, anyagokat, genetikai kódokat tárolja, minden típust egy külön listában. Azok az eszközöket, amelyeket a virológus levett magáról szintén a megfelelő eszköz listába kerül vissza. Illetve nyilvántartja, hogy maximum hány anyag és mennyi eszköz lehet az inventoryban.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

* **int maxMaterial**: Tárolja, hogy maximum mennyi eleme lehet az anyaglistáknak összesen. **Láthatóság:** -
* **int maxEquipment**: Tárolja, hogy maximum mennyi eleme lehet az eszközlistáknak összesen, csak olvasható. **Láthatóság:** -
* **ArrayList** **<Gloves> storedGloves**: Tárolja a virológus által óvóhelyről felvett kesztyűket, és azokat, amelyeket a virológus levett magáról. **Láthatóság:** -
* **ArrayList** **<Sack> storedSacks**: Tárolja a virológus által óvóhelyről felvett zsákokat, és azokat, amelyeket a virológus levett magáról. **Láthatóság:** -
* **ArrayList** **<Cloak> storedCloaks**: Tárolja a virológus által óvóhelyről felvett köpenyeket, és azokat, amelyeket a virológus levett magáról. **Láthatóság:** -
* **ArrayList** **<Axe> storedAxes**: Tárolja a virológus által óvóhelyről felvett fejszéket, és azokat, amelyeket a virológus levett magáról. **Láthatóság:** -
* **ArrayList** **<GCode> learntGCode**: Tárolja a virológus által laborból megtanult genetikai kódokat, minden tárolt kód különböző. **Láthatóság:** -
* **ArrayList** **<Aminoacid> aminos**: Tárolja a virológus által raktárból felvett aminosavakat. **Láthatóság:** -
* **ArrayList** **<Nukleotid> nukleotids**: Tárolja a virológus által raktárból felvett nukleotidokat. **Láthatóság:** -

#### Metódusok

* **void init()**: Inicializálja a 7 darab tárolt listát, és beállítja a maxMaterial-t 20-ra. **Láthatóság:** -
* **void setMaxMaterial(int maxMaterial)**:  
  Ha a jelenlegi maxMaterial nagyobb mint a kapott maxMaterial, akkor először ki kell számolni a túlcsordulást. Azaz, hogy a tárolt anyaglisták hosszainak összegee és a kapott maxMatrial különbsége (túlcsordulás = aminos lista hossza+ nukleotids lista hossza – maxMaterial). Ha a túlcsordulás nagyobb, mint nulla, akkor ellenőrizni kell, hogy az anyaglisták közül melyik a hosszabb és abból egy ciklus segítségével egyesével törlni kell annyi elemet amennyi a túlcsordulás volt.  
  Egyébként a maxMaterial értékét beállítja a kapott értékre.  
  **Láthatóság:** +
* **int getMaxMaterial()**: Visszaadja a maxMaterial értékét. **Láthatóság:** +
* **void addGloves(Gloves g)**: Ha a kapott Gloves objektum nem null és felvehető eszköz az inventoryba, akkor a tárolt kesztyűk listához hozzáadja a g-t. **Láthatóság:** +
* **void addSack(Sack s)**: Ha a kapott Sack objektum nem null és felvehető az eszköz az inventoryba, akkor a tárolt zsákok listához hozzáadja az s-t. **Láthatóság:** +
* **void addCloak(Cloak c)**: Ha a kapott Cloak objektum nem null és felvehető eszköz az inventoryba, akkor a tárolt köpenyek listához hozzáadja a c-t.  
  **Láthatóság:** +
* **void addAxe(Gloves a)**: Ha a kapott Axe objektum nem null és felvehető eszköz az inventoryba, akkor a tárolt fejszék listához hozzáadja az a-t. **Láthatóság:** +
* **Gloves removeGloves()**: Ha tárolt kesztyűk lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként nulllal tér vissza. **Láthatóság:** +
* **Sack removeSack()**: Ha a tárolt zsákok lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként nulllal tér vissza. **Láthatóság:** +
* **Cloak removeCloak()**: Ha a tárolt köpenyek lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként nulllal tér vissza. **Láthatóság:** +
* **Axe removeAxe()**: Ha a tárolt fejszék lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként nulllal tér vissza. **Láthatóság:** +
* **int learnGCode(GCode g)**: Ha a kapott GCode objektum nem null és a genetikai kód még nincs benne a megtanult genetikai kódok listájában, akkor hozzáadja ehhez a listához a g-t. Majd visszatér a megtanult genetikai kódok listájának hosszával. **Láthatóság:** +
* **void forgetGCode()**: Kiüríti a megtanult genetikai kódok listáját. **Láthatóság:** +
* **void addAmino(Aminoacid am)**: Ha a kapott Aminoacid objektum nem null és felvehető anyag az inventoryba, akkor hozzáadja am-t az aminosavak listájához. **Láthatóság:** +
* **void addNukleo(Nukleotid n)**: Ha a kapott Nukleotid objektum nem null és felvehető anyag az inventoryba, akkor hozzáadja n-t a nukleotidok listájához. **Láthatóság:** +
* **Aminoacid removeAmino()**: Ha az aminosavak lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként null-al tér vissza. **Láthatóság:** +
* **Nukleotid removeNukleo()**: Ha a nukleotidok lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként null-al tér vissza. **Láthatóság:** +
* **int getStoredAminoCount()**: Visszaadja az aminosavak lista hosszát. **Láthatóság:** +
* **int getStoredNukleotidCount()**: Visszaadja a nukleotidok lista hosszát.  
  **Láthatóság:** +
* **boolean canAddEq()**: Megadja, hogy felvehető-e eszköz az inventoryba. Ha a tárolt eszközlisták hosszainak összege kisebb, mint a maxEquipment (tárolt kesztyűk lista hossza + tárolt zsákok lista hossza + tárolt köpenyek lista hossza + tárolt fejszék lista hossza < maxEquipment), akkor igazzal tér vissza. Egyébként hamissal. **Láthatóság:** -
* **boolean canAddMaterial()**: Megadja, hogy felvehető-e anyag az inventoryba. Ha a tárolt anyaglisták hosszainak összege kisebb, mint a maxMaterial (aminosavak lista hossza + nukleotidok lista hossza < maxMaterial), akkor igazzal tér vissza. Egyébként hamissal. **Láthatóság:** -

### Material

#### Felelősség

A raktárból megszerezhető anyagokat reprezentáló absztrakt osztály.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void pickUp(Virologist v)**: absztrakt metódus. **Láthatóság:** +

### Aminoacid

#### Felelősség

Speciális típusú anyag, amely szükséges ágens készítéshez. Az aminosav raktárból inventoryba való felvételét végzi.

#### Ősosztályok

Material

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void pickUp(Virologist v)**: Ha a kapott virológus nincs lebénulva, nem vitustáncol és nem medvetáncol, akkor elkéri a virológustól az inventoryját és meghívja az inventory addAmino metódusát önmagát átadva. **Láthatóság:** +

### Nukleotid

#### Felelősség

Speciális típusú anyag, amely szükséges ágens készítéshez. A nukleotid raktárból inventoryba való felvételét végzi.

#### Ősosztályok

Material

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void pickUp(Virologist v)**: Ha a kapott virológus nincs lebénulva, nem vitustáncol és nem medvetáncol, akkor elkéri a virológustól az inventoryját és meghívja az inventory addNukleo metódusát önmagát átadva. **Láthatóság:** +

### GCode

#### Felelősség

A genetikai kódot reprezentáló osztály. Tárolja, hogy milyen ágens, milyen anyagok felhasználásával készíthető el belőle. Elkészíti az ágenst a virológusnak.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

* **String GCName**: Tárolja, hogy milyen ágens készíthető el ebből az objektumból. **Láthatóság:** -
* **int NeededAmino**: Tárolja, hogy mennyi aminosav szükséges az ágens elkészítéséhez. **Láthatóság:** -
* **int NeededNukleo**: Tárolja, hogy mennyi nukleotid szükséges az ágens elkészítéséhez. **Láthatóság:** -

#### Metódusok

* **String getGCName()**: Visszaadja, hogy milyen ágens készíthető ebből a genetikai kódból. **Láthatóság:** +
* **Agent create(Virologist v)**: Ha a virológus nincs lebénulva, nem vitustáncol és nem medvetáncol, akkor először elkéri a virológustól az inventoryját. Majd, ha az inventoryban tárolt anyagok mennyisége nagyobb vagy egyenlő, mint a szükséges anyagok (tárolt aminosav mennyiség >= szükséges aminosav és tárolt nukleotid mennyiség >= szükséges nukleotid), akkor egy-egy ciklus segítségével kiveszi az inventoryból a szükséges mennyiséget mind a két anyagból. Ezután egy switch case szerkezettel a GCName alapján elkészíti a megfelelő ágens objektumot és ezzel visszatér.

Egyébként nulllal tér vissza.  
**Láthatóság:** +

### Field

#### Felelősség

Ez az osztály az összes mező ősosztály és sima mezőként működik. Tárolja a szomszédos mezőket és az itt tartózkodó virológusokat. Kezeli a virológus megérkezését a mezőre.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

* **ArrayList** **<Field> Neighbours**: Tárolja a szomszédos mezőket. **Láthatóság:** -
* **ArrayList** **<Virologist> Viros**: Tárolja a mezőn tartózkodó virológusokat. **Láthatóság:** -
* **String ID**: A field objektum egyedi azonosítóját tárolja. **Láthatóság:** -

#### Metódusok

* **void setNeighbours(Field f1, Field f2)**: Ha egyik Field objektum se null, akkor az f1 szomszéd listájához hozzáadja f2-t és az f2 szomszéd listájához hozzáadja f1-t. **Láthatóság:** +
* **ArrayList** **<Virologist> getVirologists()**: Visszaadja a mezőn tartózkodó virológusok listáját. **Láthatóság:** +
* **ArrayList** **<Field> getNeighbours()**: Visszaadja a mező szomszédainak listáját. **Láthatóság:** +
* **void arrive(Virologist v)**: Ha kapott virológus nincs lebénulva, nem vitustáncol és nem medvetáncol, akkor egy ciklussal megnézi, hogy a többi itt található virológus közül le van-e bénulva valamelyik. Ha talál egyet akkor megkérdezi a felhasználót, hogy meg akarja-e lopni (wannaSteal). Ha a válasz igen akkor meghívja a bénult virológus stealBy függvényét és átadja magát, majd kilép ciklusból. **Láthatóság:** +
* **boolean wannaSteal()**: Megkérdezi a játékost, hogy akar-e lopni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -
* **void addVirologist(Virologist v)**: Hozzáadja a kapott Virologist objektumot az itt tartózkodó virológusok listájához. **Láthatóság:** +
* **void removeVirologist(Virologist v)**: Eltávolítja a kapott Virologist objektumot az itt tartózkodó virológusok listájából. **Láthatóság:** +
* **ArrayList** **<Virologist> getVirologists()**: Visszaadja a Viros listát. **Láthatóság:** +
* **String getID()**: Visszaadja az ID attribútum értékét. **Láthatóság:** +

### Storage

#### Felelősség

Speciális típusú mező, amelyen anyagokat szedhet fel az ide érkező virológus.

#### Ősosztályok

Field

#### Interfészek

Steppable

#### Attribútumok

* **boolean Empty**: Tárolja, hogy a raktár éppen üres-e vagy nem. Hamisra van inicializálva. **Láthatóság:** -
* **int remnaingTime**: Tárolja, hogy a raktár, hány körön keresztül lesz üres. **Láthatóság:** -

#### Metódusok

* **void arrive(Virologist v)**: Először meghívja az ősosztály arrive metódusát és átadja neki v-t. Ezután meghívja a v getBearDancingjét, ha ez igazzal tér vissza akkor beállítja az Empty attribútumot igazra, az AddSteppable-nek átadva magát bekerül a léptethető objektumok közé és beállítja a remaningTime attribútumot 2-re. Egyébként meghívja a wannaPickUp metódust. Ha ez igazzal tért vissza és az Empty hamis, akkor meghívja az addMat metódust és átadja neki v-t. **Láthatóság:** +
* **void addMat(Virologist v)**: Generál egy random egész számot 0 és 1 között. Majd egy switch case szerkezetben a 2 esetet szétválasztva más és más típusú anyagot hoz létre és meghívja az adott anyagnak megfelelő pickUp metódust átadva neki v-t. (0 esetén aminosavat hoz létre, 1 esetén nukleotidot és így tovább). **Láthatóság:** -
* **boolean wannaPickUp()**: Megkérdezi a játékost, hogy akar-e anyagot felvenni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -
* **void step()**: Csökkenti a remaningTime-ot egyel. Ha a remaningTime nulla, akkor beállítja az Empty attribútumot hamisra, majd meghívja a Timer RemoveSteppable függvényét és átadja önmagát.

### Shelter

#### Felelősség

Speciális típusú mező, amelyen eszközöket szedhet fel az ide érkező virológus.

#### Ősosztályok

Field

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void arrive(Virologist v)**: Először meghívja az ősosztály arrive metódusát és átadja neki v-t. Majd meghívja a wannaPickUp metódust, ha ez igazzal tér vissza, akkor meghívja az addEq metódust és átadja neki v-t. **Láthatóság:** +
* **void addEq(Virologist v)**: Generál egy random egész számot 0 és 3 között és egy switch case szerkezetben a 4 esetet szétválasztva más és más típusú eszközt hoz létre és meghívja az adott eszköznek megfelelő pickUp metódust átadva neki v-t.. (pl. 0 esetén zsákot hoz létre, 1 esetén kesztyűt és így tovább). **Láthatóság:** -
* **boolean wannaPickUp()**: Megkérdezi a játékost, hogy akar-e eszközt felvenni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -

### Lab

#### Felelősség

Speciális típusú mező, amelyen genetikai kódokat ismerhet meg az ide érkező virológus.

#### Ősosztályok

Field

#### Interfészek

Nincs

#### Attribútumok

* **GCode gc**: Tárolja a laborban megtanulható genetikai kód objektumot. **Láthatóság:** -

#### Metódusok

* **void arrive(Virologist v)**: Először meghívja az ősosztály arrive metódusát és átadja neki v-t. Majd meghívja a wannaTouchy metódust, ha ez igazzal tér vissza, akkor meghívja a Virológus touchy metódusát és átadja neki gc-t. **Láthatóság:** +
* **boolean wannaPickUp()**: Megkérdezi a játékost, hogy akar-e genetikai kódot letapogatni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -

### InfectedLab

#### Felelősség

Speciális típusú labor, amelyen a virológus megfertőződhet medvetánccal.

#### Ősosztályok

Field → Lab

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void arrive(Virologist v)**: Először meghívja az ősosztály arrive metódusát. A v getImmunity metódus hívás megadja, hogy a virológus immunis-e. Ha immunis akkor visszatér az arrive. Ezután a v getWornCloak metódussal elkéri a v által viselt köpenyek listáját. Ha ezen listának a hossza nagyobb, mint nulla, akkor egy ciklussal végig megy a kapott listán és minden Cloak objektum evade-jét meghívja. Ha az egyik evade igazzal tér vissza akkor törli az objektumot a listából és visszatér az arrive. Majd meghívja a v setBearDancing metódusát true értékkel, létrehoz egy új BearDance objektumot és a Timer AddSteppable-jének átadva v-t bekerül a léptethető dolgok közé. **Láthatóság:** +

### Agent

#### Felelősség

Az elkészíthető ágensek ősosztálya. Tárolja az ágens nevét, azon virológust akin hat, az ágens hatásának időtartamát, az időtartamot, ameddig felhasználható az ágens és fennmaradó időt. Az ágensek hatásait kezeli.

#### Ősosztályok

Nincs

#### Interfészek

Steppable

#### Attribútumok

* **int RemaningTime**: Tárolja, hogy hány kör után fog megszűnni az ágens. **Láthatóság:** -
* **String Name**: Tárolja az ágens nevét. **Láthatóság:** -
* **int Expiration**: Tárolja, hogy hány körig használható fel az ágens. **Láthatóság:** -
* **int EffectTime**: Tárolja, hogy hány körig hat a felkent ágens. **Láthatóság:** -
* **Virologist VirologistUnderEffect**: Tárolja, hogy melyik virológusra hat az ágens. **Láthatóság:** -

#### Metódusok

* **void step()**: absztrakt metódus. **Láthatóság:** +
* **void setVirologistUnderEffect(Virologist v)**: Beállítja a VirologistUnderEffectattribútumot v-re. **Láthatóság:** +
* **void getEffectTime()**: Visszaadja az EffectTime attribútum értékét. **Láthatóság:** +
* **void setRemaningTime(int time)**: Beállítja a RemaningTime attribútum értékét time-ra. **Láthatóság:** +
* **void getRemaningTime()**: Visszaadja a RemaningTime attribútum értékét. **Láthatóság:** +

### Oblivion

#### Felelősség

Speciális típusú ágens, melynek hatására a virológus elfelejti az összes genetikai kódot amit eddig megtanult.

#### Ősosztályok

Agent

#### Interfészek

Steppable

#### Attribútumok

Nincs

#### Metódusok

* **void step()**: Ha a VirologistUnderEffect nem null, akkor meghívja VirologistUnderEffect getInventory függvényét, majd az Inventory forgetGCodes metódusát, beállítja a RemaningTime-t nullára és meghívja a Timer RemoveSteppable függvényét és átadja önmagát. **Láthatóság:** +

### Paralysis

#### Felelősség

Speciális típusú ágens, melynek hatására a virológus lebénul, így nem tud semmit csinálni.

#### Ősosztályok

Agent

#### Interfészek

Steppable

#### Attribútumok

Nincs

#### Metódusok

* **void step()**: Ha a VirologistUnderEffect nem null, akkor meghívja VirologistUnderEffect setParalyzed függvényét igaz értéket átadva, csökkenti a RemaningTime-t egyel. Ha a RemaningTime nulla, akkor meghívja VirologistUnderEffect setParalyzed függvényét hamis értékkel és a Timer RemoveSteppable függvényét és átadja önmagát. **Láthatóság:** +

### Immunity

#### Felelősség

Speciális típusú ágens, melynek hatására a virológus immunissá válik, tehát nem hatnak rá a mások által rákent ágensek.

#### Ősosztályok

Agent

#### Interfészek

Steppable

#### Attribútumok

Nincs

#### Metódusok

* **void step()**: Ha a VirologistUnderEffect nem null, akkor meghívja VirologistUnderEffect setImmunity függvényét igaz értéket átadva, csökkenti a RemaningTime-t egyel. Ha a RemaningTime nulla, akkor meghívja VirologistUnderEffect setImmunity függvényét hamis értékkel és a Timer RemoveSteppable függvényét és átadja önmagát. **Láthatóság:** +

### VitusDance

#### Felelősség

Speciális típusú ágens, melynek hatására a virológus lebénul, így nem tud semmit csinálni.

#### Ősosztályok

Agent

#### Interfészek

Steppable

#### Attribútumok

Nincs

#### Metódusok

* **void step()**:

Ha: VirologistUnderEffect != null, akkor

VirologistUnderEffect setDancing(igaz)

f ← VirologistUnderEffect getStandingField

Neighbours ← f getNeighbours()

chosenField ← Neighbours lista egyik eleme

chosenField move(VirologistUnderEffect,hamis)

Ha: RemaningTime = nulla, akkor

VirologistUnderEffect setDancing(hamis)

Time RemoveSteppable(ez a VitusDance objektum)

**Láthatóság:** +

### BearDance

#### Felelősség

Speciális típusú ágens, melynek hatására a virológus minden vele egy mezőn tartózkodó virológust megtámad BearDance ágenssel.

#### Ősosztályok

Agent

#### Interfészek

Steppable

#### Attribútumok

Nincs

#### Metódusok

* **void step()**:

Ha: VirologistUnderEffect != null, akkor

VirologistUnderEffect setBearDancing(igaz)

f ← VirologistUnderEffect getStandingField

infect(f)

Ha: VirologistUnderEffect getDied() = hamis, akkor

Neighbours ← f getNeighbours()

chosenField ← Neighbours lista egyik eleme

chosenField move(VirologistUnderEffect,hamis)

infect()

Ha: VirologistUnderEffect getDied() = igaz, akkor

Time RemoveSteppable(ez a BearDance objektum)

**Láthatóság:** +

* **void infect(f)**:

virologists ← f getVirologists()

Ciklus: amíg i kisebb, mint virologist lista hossza

Ha: VirologistUnderEffect != i-dik Virologist objektum, akkor

i-dik Virologist attack(ez a BearDance objektum)

Ciklus vége

**Láthatóság:** -

### Steppable

#### Felelősség

Egy interface, amely a léptethető objektumokat kezeli.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void step()**: üres metódus. **Láthatóság:** +

### Game

#### Felelősség

Létrehozza, inicializálja a mezőket, virológusokat és felépíti a pályát. Kezeli, hogy melyik virológus cselekedhet éppen és egy idő után lép a következő virológusra ezt a játék végig csinálja. Ha vége a játéknak minden virológust értesít erről majd lebontja a pályát.

#### Ősosztályok

Nincs

#### Interfészek

Steppable

#### Attribútumok

* **int MaxGCode**: Tárolja, hogy hány különböző genetikai kódot kell összegyűjtenie a virológusoknak, csak olvasható. **Láthatóság:** -
* **int activeVirologist**: Tárolja, hogy a virológusok lista hányadik eleme az éppen aktív virológus. **Láthatóság:** -
* **boolean moved**: Tárolja, hogy a soron következő virológus mozgot-e már ebben a körben, hamisra van inicializálva. **Láthatóság:** -
* **ArrayList** **<Field> fields**: Tárolja a játék összes mezőjét. **Láthatóság:** -
* **ArrayList** **<Virologist> virologists**: Tárolja a játékban resztvevő virológusokat. **Láthatóság:** -
* **Timer timer**: Tárolja, a játék során használt időzítő objektumot. **Láthatóság: -**
* **long startTime**: Tárolja, hogy a virológus mikor kezdte a körét. **Láthatóság:** -
* **int roundTime**: Tárolja, hogy a virológus köre maximum meddig tart. **Láthatóság:** -

#### Metódusok

* **int getMoved()**: Visszaadja a moved attribútum értékét. **Láthatóság:** +
* **void setMoved(boolean moved)**: Beállítja a moved attribútumot a kapott értékre. **Láthatóság:** +
* **Virologist getActiveVirologist()**: Visszaadja a virologists lista azon elemét, amely activeVirologist sorszámú helyen van. **Láthatóság:** +
* **void removeVirologist(Virologist v)**: Eltávolítja a kapott Virologist objektumot a játékban levő virológusok listájából **Láthatóság:** +
* **boolean doCommand()**: Visszatér, hogy tud-e cselekedni/cselekedett-e valamit az éppen körre kerülő játékos vagy sem. **Láthatóság:** -
* **void startGame()**: Felépíti a mezőket, magát a pályát. Létrehozza a virológusokat, azokat elhelyezi kezdeti mezőjükön, majd elindítja a játékot, azaz az első kört. **Láthatóság:** +
* **void endGame(Virologist v)**:

Ha v nem null, akkor

ciklus: amíg i kisebb, mint fields lista hossza

ciklus: amíg j kisebb, mint a virologists lista hossza

meghívja az i-dik Field j-dik Virologist objektum showEndGame-t v-vel

ciklus vége

ciklus vége

**Láthatóság:** +

* **void step()**: Ha az activeVirologist = nulla, akkor:

ciklus: s végig fut Steppables léptethető objektumokon

meghívja s step metódusát

ciklus vége

next ←hamis

startTime ← rendszer idő

ciklus: amíg rendszer idő – startTime < roundTime és !next

next ← doCommand

ciklus vége

activeVirologist ← 1+activeVirologist mod virologists lista hossza

move ← hamis

**Láthatóság:** +

### Virologist

#### Felelősség

Lehetővé teszi a felhasználó és a rendszer közötti kommunikációt. A játékos ezen az osztályon keresztül mozoghat a szomszédos mezők között, ágenst generálhat, felvehet és levehet eszközöket (amelyekből maximum 3 lehet rajta egyszerre), ellenfeleire vagy önmagára ágenst kenhet, és védekezhet egy általa indított támadás (ágens kenés) ellen. Emellett eszközöket, anyagokat lophat egy bénult virológustól.

#### Ősosztályok

Nincs

#### Interfészek

Steppable

#### Attribútumok

* **boolean Immunity**: Tárolja, hogy a virológus immunis-e. **Láthatóság:** -
* **boolean Paralyzed**: Tárolja, hogy a virológus le van-e bénulva. **Láthatóság:** -
* **boolean Dancing**: Tárolja, hogy a virológus épp vitustánc hatása alatt áll-e. **Láthatóság:** -
* **boolean BearDancing**: Tárolja, hogy a virológus épp medvetánc hatása alatt áll-e. **Láthatóság:** -
* **boolean Dead**: Tárolja, hogy a virológus meghalt-e. **Láthatóság:** -
* **ArrayList** **<Gloves> WornGloves**: A virológus által viselt kesztyűk listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik három eszközfajta listája mennyi elemből áll). **Láthatóság:** -
* **ArrayList** **<Sack> WornSacks**: A virológus által viselt zsákok listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik három eszközfajta listája mennyi elemből áll). **Láthatóság:** -
* **ArrayList** **<Cloak> WornCloaks**: A virológus által viselt köpenyek listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik három eszközfajta listája mennyi elemből áll). **Láthatóság:** -
* **ArrayList** **<Axe> WornAxes**: A virológus által viselt fejszék listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik három eszközfajta listája mennyi elemből áll). **Láthatóság:** -
* **Field standingField**: Az a mező, amelyen a virológus áll. **Láthatóság:** -
* **Inventory inventory**: Tárolja a virológus inventoryját. **Láthatóság:** -
* **Agent agents**: A virológus által készített ágensek listája. **Láthatóság:** -
* **Game game**: Az a játék, amelyen a virológus részt vesz. **Láthatóság:** -

#### Metódusok

* **void setImmunity(boolean bImmun)**: Beállítja az Immunity attribútum értékét a bementi paraméterére. **Láthatóság:** +
* **void setParalyzed(boolean bParalyzed)**: Beállítja a Paralyzed attribútum értékét a bementi paraméterére. **Láthatóság:** +
* **void setDancing(boolean bDancing)**: Beállítja a Dancing attribútum értékét a bementi paraméterére. **Láthatóság:** +
* **void setBearDancing(boolean bBearDancing)**: Beállítja az BearDancing attribútum értékét a bementi paraméterére. **Láthatóság:** +
* **void setStandingField(Field f)**: Beállítja a standingField attribútum értékét a bementi paraméterére. **Láthatóság:** +
* **void die()**: Beállítja az Died attribútum értékét igazra. **Láthatóság:** +
* **boolean getImmunity()**: Visszaadja az Immunity attribútum értékét. **Láthatóság:** +
* **boolean getParalyzed()**: Visszaadja a Paralyzed attribútum értékét. **Láthatóság:** +
* **boolean getDancing()**: Visszaadja a Dancing attribútum értékét. **Láthatóság:** +
* **boolean getBearDancing()**: Visszaadja a BearDancing attribútum értékét. **Láthatóság:** +
* **Field getStandingField()**: Visszaadja a standingField attribútum értékét.  
  **Láthatóság:** +
* **Inventory getInventory()**: Visszaadja az inventory attribútum értékét. **Láthatóság:** +
* **void init(Game game, Field standingField)**: Ha a paraméterek nem null értékűek, akkor beállítja a game és standingField attribútumok értékeit a bementi paraméterekre,. Egyébként létrehoz új objektumokat. Majd beállítja az Immunity, Paralyzed, Dancing, BearDancing és Died attribútumokat hamisra. Az agents, WornGloves, WornSacks, WornCloaks, WornAxes és az inventory attribútumokat inicializálja. **Láthatóság:** +
* **void step()**:Ciklus: amíg i kisebb mint az agents lista hossza

a ← i-dik Agent

meghívja a setRemaningTime-ját a getRemaningTime-1 -el

ha: a getRemaningTime-ja = 0, akkor

agents-ből törli az i-dik elemet

különben:

agents i-dik elemét a-ra állítja

ciklus vége

**Láthatóság:** +

* **void equipGloves()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként, ha a canEquipEq igazat ad vissza, akkor meghívja az inventory removeGloves metódusát, amely egy Gloves objektummal tér vissza, ha ez nem null, akkor a WornGloves listához hozzáadja ezt. **Láthatóság:** +
* **void equipSack()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként ha a canEquipEq igazat ad vissza, akkor meghívja az inventory removeSack metódusát amely egy Sack objektummal tér vissza, ha ez nem null, akkor a WornSacks listához hozzáadja ezt, elkéri az inventory-tól, hogy mennyi a jelenlegi maxMaterial (currentMaxMaterial ← inventory getMaxMaterial) és az inventory setMaxMaterialnak átadja az elkért MaxMaterial és az adott zsák kapacitásának összegét (setMaxMaterial(currentMaxMaterial+sack getCapacity())). **Láthatóság:** +
* **void equipCloak()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként, ha a canEquipEq igazat ad vissza, akkor meghívja az inventory removeCloak metódusát, amely egy Cloak objektummal tér vissza, ha ez nem null, akkor a WornCloaks listához hozzáadja ezt. **Láthatóság:** +
* **void equipAxe()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként, ha a canEquipEq igazat ad vissza, akkor meghívja az inventory removeAxe metódusát, amely egy Axe objektummal tér vissza, ha ez nem null, akkor a WornAxes listához hozzáadja ezt. **Láthatóság:** +
* **boolean defense(Virologist vFr, Agent a)**:

bDefenseSucceed ← hamis

Ha: Dancing = igaz vagy Paralyzed = igaz, vagy BearDancing = igaz, akkor

Vissza: bDefenseSucceed

Különben, ha: a = null vagy vFr = null, akkor

bDefenseSucceed ← igaz

Vissza: bDefenseSucceed

Különben:

Ha: WornGloves lista hossza > 0, akkor

g ← WornGloves 0-dik eleme

Ha: g wannaUse() = igaz, akkor

usable ← g getUsable()

g setUsable(usable-1)

Ha usable = 1, akkor

törli a WornGloves 0-dik elemét

Különben:

g use(vFr, önmaga, a)

bDefenseSucceed ← igaz

Ha bDefenseSucceed = hamis, akkor

Ha WornAxes lista hossza > 0, akkor

a ← WornAxes 0-dik eleme

used ← a getUsed()

Ha: a wannaUse() = igaz és used = hamis, akkor

törli a WornAxes 0-dik elemét

a use(vFr)

bDefenseSucceed ← igaz

Ha bDefenseSucceed = hamis, akkor

ciklus: amíg i kisebb, mint az WornCloak lista hossza

c ← WornCloaks i-dik eleme

bDefenseSucceed ← c evade()

Ha bDefenseSucceed = igaz, akkor

törli a WornCloaks i-dik elemét

stop

ciklus vége

Vissza: bDefenseSucceed

**Láthatóság:** +

* **void takeOffGloves()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként meghívja a takeOffGloves metódust átadva neki az inventoryt. **Láthatóság:** +
* **boolean takeOffGloves(Inventory inv)**:

bFound ← hamis

Ha WornGloves lista hossza > 0, akkor

g ← WornGloves 0-dik eleme

inv addGloves(g)

bFound ← igaz

Vissza: bFound

**Láthatóság:** -

* **void takeOffCloak()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként meghívja a takeOffCloak metódust átadva neki az inventoryt. **Láthatóság:** +
* **boolean takeOffCloak(Inventory inv)**:

bFound ← hamis

Ha WornCloaks lista hossza > 0, akkor

g ← WornCloaks 0-dik eleme

inv addCloak(g)

bFound ← igaz

Vissza: bFound

**Láthatóság:** -

* **void takeOffAxes()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként, ha: a wannaDropAxe hamissal tér vissza, akkor meghívja a takeOffAxes metódust átadva neki az inventoryt. Ha wannaDropAxe igazzal tér vissza, akkor törli a WornAxes 0-dik elemét. **Láthatóság:** +
* **boolean takeOffAxes(Inventory inv)**:

bFound ← hamis

Ha WornAxes lista hossza > 0, akkor

g ← WornAxes 0-dik eleme

inv addAxes(g)

bFound ← igaz

Vissza: bFound

**Láthatóság:** -

* **void takeOffSack()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként meghívja a takeOffSack metódust átadva neki az inventoryt. **Láthatóság:** +
* **boolean takeOffSack(Inventory inv)**:

bFound ← hamis

Ha WornSacks lista hossza > 0, akkor

s ← WornSacks 0-dik eleme

inv addSack(s)

currentMaxMaterial ← inv getMaxMaterial

inv setMaxMaterial(currentMaxMaterial + s getCapacity())

bFound ← igaz

Vissza: bFound

**Láthatóság:** -

* **void createAgent(GCode gc)**:

Ha: a virológus lebénult vagy vitustáncol vagy medvetáncol, akkor

Vissza:

Ha: gc != null, akkor

a ← gc create (önmaga)

Ha: a != null, akkor

agents listához hozzáadja a-t

**Láthatóság:** +

* **void attack(Virologist vTo, Agent a)**:

Ha: Dancing = igaz vagy Paralyzed = igaz vagy Immunity = igaz vagy standingField != vTo standingField, akkor

Vissza:

Ha vTo != null és a != null, akkor

bDefenseSuccessful = vTo assaulted(önmaga, a)

Ha: bDefenseSuccessful != hamis és vTo getParalyzed(), akkor

Ha wannaSteal(), akkor

vTo stealBy(önmaga)

**Láthatóság: +**

* **boolean wannaSteal()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként megkérdezi a játékost, hogy akar-e lopni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -
* **void move(Field f, boolean bEmittedByUser)**:

Ha: Dancing = igaz vagy Paralyzed = igaz, vagy standingField szomszédai között nincsen f, akkor

Vissza:

Ha: f != null, akkor

standingField RemoveVirologist(önmage)

f addVirologist(önmaga)

standingField ← f

Ha: bEmittedByUser, akkor

f arrive(önmaga)

**Láthatóság:** +

* **void stealBy(Virologist v)**:

Ha: v = null vagy v Paralyzed = igaz, vagy v Dancing = igaz, vagy v BearDancing = igaz, akkor

Vissza:

Ha: Paralyzed = igaz, akkor,

stealerInv ← v getInventory()

objectName ← v chooseObjectToSteal

bEquippedFound ← hamis

switch(objectName)

case Aminoacid: stealerInv addAmino(inventory removeAmino)

case Nukleotid: stealerInv addNukleo(inventory removeNukleo)

case Gloves:

bEquippedFound ← takeOffGloves(inventory removeGloves())

Ha bEquippedFound =hamis, akkor

stealerInv addGloves(inventory removeGloves)

case Sack:

bEquippedFound ← takeOffSack(inventory removeSack())

Ha bEquippedFound =hamis, akkor

stealerInv addSack(inventory removeSack)

case Cloak:

bEquippedFound ← takeOffGloves(inventory removeCloak())

Ha bEquippedFound =hamis, akkor

stealerInv addCloak(inventory removeCloak)

case Axe:

bEquippedFound ← takeOffAxe(inventory removeAxe())

Ha bEquippedFound =hamis, akkor

stealerInv addAxe(inventory removeAxe) **Láthatóság:** +

* **boolean assaulted(Virologist vFr, Agent a)**:

bDefenseSucceed ← hamis

Ha: getHasEffectedAgent() = hamis és vFr != null és a != null, akkor

Ha: vFr = ez a Virologist objektum, akkor

bDefenseSucceed ← defense(vFr, a)

Ha: bDefenseSucceed = hamis, akkor

a setVirologistUnderEffect(ez a Virologist objektum)

a setRemaningTime(a getEffectTime())

Timer addSteppable(a)

Vissza: bDefenseSucceed

**Láthatóság:** -

* **String chooseObjectToSteal()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként megkérdezi a játékost, hogy milyen anyagot vagy eszközt akar lopni. És visszatér a felhasználó által beadott Stringel. **Láthatóság:** -
* **boolean canEquipEq()**: Ha a WornGloves, WornSacks, WornCloaks, WornAxes listák hosszainak összege kisebb, mint 3, akkor igazzal tér vissza. Ha nagyobb vagy egyenlő akkor hamissal. **Láthatóság:** -
* **boolean getHasEffectedAgent()**: Visszaaadja az Immunity és Dancing és Paralyzed logikai kifejezés értékét. **Láthatóság:** -
* **void showEndGame(Virologist v)**:

Ha: v != null és v != ez a Virologist objektum, akkor

Kiír: Game over! You Lost.

Különben:

Kiír: Congratulation! You won the game.

* **boolean wannaDropAxe()**:Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként megkérdezi a játékost, hogy akar-e fejszét használni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -
* **ArrayList** **<Cloak> getWornCloaks()**: Visszaadja a virologús által viselt köpenyek listáját. **Láthatóság:** +
* **void die()**: Beállítja a Died attribútumot igazra. **Láthatóság:** +

### Timer

#### Felelősség

Periodikus időt reprezentál a játékban, a léptethető (Steppable) dolgokat lépteti.

#### Ősosztályok

Nincs

#### Interfészek

Steppable

#### Attribútumok

* **Steppable Steppables**: Tárolja a léptethető dolgokat. **Láthatóság:** -

#### Metódusok

* **void tick()**: Minden léptethető dolgot léptet az előre beállított értékkel. **Láthatóság:** +
* **void addSteppable(Steppable s)**: Új léptethető dolog felvétel a Steppable listára. **Láthatóság:** +
* **void removeSteppable(Steppable s)**: A lejárt idejű léptető (tehát a léptető elérte a nullát) törlése kerül ezáltal a metódus által a Steppable listából. **Láthatóság:** +

## A tesztek részletes tervei, leírásuk a teszt nyelvén

*Csapatunk az előző házi feladatot félre értette, ezért itt pontosan definiáljuk a bemeneti és kimeneti nyelvezetet:*

1. Mező létrehozása

    input:

        create Field <mező\_objectname>

    output:

        <mező\_id>

Pl. sima mező létrehozása:

    input: create Field Field

    output: Field1

 Pl. labor létrehozása:

    input: create Field Lab

    output: Lab1

Pl. fertőzött labor létrehozása:

    input: create Field InfectedLab

    output: InfectedLab1

2. Virológus létrehozása

    input:

        create Virologist <mező\_id> <immunity> <paralyzed> <dancing> <beardancing> <aminoacid\_szám\_inventoryban> <nukleotid\_szám\_inventoryban> <kesztyű\_szám\_inventoryban> <zsák\_szám\_inventoryban> <köpeny\_szám\_inventoryban> <fejsze\_szám\_inventoryban>

    lehetséges output-ok vagy szóval elválasztva:

            <virologist\_id>

        vagy

            Virologist not created, too many material.

        vagy

            Virologist not created, too many equipment

        vagy

        Virologist not created, invalid Field Id

3. Szomszédos mezők beállítása

    input:

        setneighbor <mező\_id\_1> <mező\_id\_2>

    lehetésges output-ok vagy szóval elválasztva:

        <mező\_id\_1> neigbors: <mező\_id> <mező\_id>…….<mező\_id\_2>

        <mező\_id\_2> neigbors: <mező\_id> <mező\_id>…….<mező\_id\_1>

4.Virológus szomszédos mezőre lép

input:

move

output:

ha a virológus nem tud lépni, hibaüzenet:

Virologist paralyzed. Can not perform action. stb

Neighbors: <mező\_id> <mező\_id> ……  (ide léphet)

input:

    <választott\_mező\_id>  
    output, ha a megadott id helytelen (ekkor ismét meg kell adni bemenetként az id-t,

amíg nem valid id-t adtunk meg):

Invalid id  
Neighbors: <mező\_id> <mező\_id> ……  (ide léphet)

output, ha a megadott id helyes:

Virologist moved to <választott\_mező\_id>

    output, ha bénult virológust találtunk az új mezőn:

<virológus\_id> paralyzed. Wanna steal? (y/n)  
input, ha lopni akarunk (bármely egyéb input esetén nem lopunk):

y  
output, ha lopást választottunk:

what to steal? (Aminoacid/Nukleotid/Axe/Gloves/Cloak/Sack)  
lehetséges inputok az előző kérdés utáni zárójeles részben található, egyet kell választani közülük, a választást most <object\_name> jelöli (helytelen input esetén újra kérdez):

<object\_name>  
lehetséges output-ok vagy szóval elválasztva:

<object\_name> found, added to inventory

vagy

<object\_name> found, but inventory full, drop <object\_name>

vagy

        <object\_name> not found

output, ha Storage-ra vagy Shelter-re lépünk:

    Wanna pick up? (y/n)

input, ha felveszünk (bármely egyéb input esetén nem veszünk fel semmit):

    y

output, ha felvevést választottuk, és Storage-en állunk (csak teszt üzemmódban, éles környezetben ezt a kérdést nem tesszük fel a felhasználónak!!!):

    What to pick up? (Aminoacid/Nukleotid)

Input (a zárójelben / jellel elválasztott szavak egyikét kell megadni, helytelen input esetén újra kérdezünk):

    <object\_name>

output, ha felvevést választottuk, és Shelter-en állunk (csak teszt üzemmódban, éles környezetben ezt a kérdést nem tesszük fel a felhasználónak!!!):

    What to pick up? (Axe/Gloves/Cloak/Sack)

lehetséges output-ok vagy szóval elválasztva:

<object\_name> added to inventory

vagy

inventory full

    output, ha laborra lépünk:

        Wanna touchy?(y/n)

    input, ha tapogatunk (bármely más input esetén nem tapogatunk):

        y

    output, ha nem nyertük meg a játékot:

        Known codes: <ismert\_kódok\_száma>

    output, ha megnyertük a játékot:

        <virológus\_id> won.

    output, ha nem nyertük meg a játékot, a labor fertőzött, és nincs a virológuson védőágens vagy köpeny, vagy van köpeny, de az nem védett meg:

        BearDance infection successful

    output, ha nem nyertük meg a játékot, fertőzött mezőre lépünk, és nincs a virológuson védőágens, de van köpeny (csak teszt üzemmódban, éles környezetben ezt a kérdést nem tesszük fel a felhasználónak!!!):

        BearDance infection, cloak found, wanna defend? (y/n)

    input, ha kivédjük (bármely más bemenet esetén nem védjük ki):

        y

    output, ha a köpeny megvédett:

        BearDance infection defended, cloak lost.

5. Eszköz felvétele inventory-ból

input:

        equip <eszköz\_osztálynevének\_első\_karaktere>

lehetséges output-ok vagy szóval elválasztva:

            Equip successful.

        vagy

            Too many equipment.

        vagy

            Equipment not found in inventory

6. Eszköz levétele

input:

takeoff <eszköz\_osztálynevének\_első\_karaktere>

lehetséges output-ok vagy szóval elválasztva:

        Inventory full, drop equipment.

    vagy

        Put it to inventory? (y/n)

input, ha volt hely az inventory-ban, és vissza akarjuk tenni oda (bármely más bemenet esetén eldobjuk):

    y

output, ha visszatesszük az inventoryba:

    Equipment added to inventory.

output, ha nem tesszük vissza az inventoryban:

    Drop equipment.

7. Vezérlés átadása a következő virológusnak

input:

next

lehetséges outputok:

    amíg a soron lévő következő virológus cselekvésképtelen:

        <virologus\_id> under <ágens\_objectname>, skip step  
    ha találtunk egy cselekvőképes virológust:  
        <virológus\_id> turn

ha a kör végéről visszatérünk a kör elejére, akkor a vezérlés átadása előtt az összes steppable-re:

        <steppable\_id> step  
        <output\_of\_step\_for\_actual\_steppable>

Virologus step:

output minden ágensre:

        <ágens\_id> remaining time: <experation>

Immunity, Paralysis step

    output:

        <ágens\_id> remaining time <remaining\_effecttime>

VitusDance step

output (csak teszt üzemmódban kérdezzük hova lépjünk, éles játék során ez random, a felhasználó nem befolyásolhatja!!!):

        Select neighbor: <fied\_id> <field\_id> <field\_id>

    input (helytelen input esetén újrakérdez):

        <field\_id>

    output:

        Moved  to <field\_id>.

        <ágens\_id> remaining time <remaining\_effecttime>

BearDance step

    output:

        minden a mezőn álló virológusra, ha immunitás ágens védi:

            <virológus\_id> Beardence infection defended with Immunity.

minden a mezőn található virológusra, ha nem védi immunitás ágens, és nincs semmilyen más ágens hatása alatt:

        ha a virológuson van fejsze:

            <virológus\_id> Wanna use axe? (y/n)

        ha a virológuson van kesztyű, és nem használta a fejszét:

            <virológus\_id> Wanna use gloves? (y/n)

        ha a virológuson van köpeny (csak teszt üzemmódban kérdezzük hogy a köpeny sikeresen véd-e, éles játék során ez random, a felhasználó nem befolyásolhatja!!!):

            <virológus\_id> Wanna defend with cloak? (y/n)

input bármely eszköz    használata esetén (minden más bemenet esetén nem használjuk az adott eszközt):

        y

    output, ha használtunk eszközt:

        <virológus\_id> Beardence infection defended, <Axe used / Gloves usable: <szám> / Gloves lost / Cloak lost>

    output, ha a virológus nem áll immunity ágens hatása alatt, és nem használt eszközt, mert nem volt neki / nem akart / ágens hatása miatt cselekvőképtelen:

    <virológus\_id> BearDance infection successful.

    output, ha a virológus BearDance ágens hatása alatt áll:

    <virológus\_id> BearDance already infected.

output (csak a teszt üzemmódban kérdezzük hova lépjünk, éles játék során ez random, a felhasználó nem befolyásolhatja!!!):

        Select neighbor: <fied\_id> <field\_id> <field\_id>

Input teszt üzemmódban (helytelen input esetén újrakérdez):

        <field\_id>

    output:

        Moved  to <field\_id>.

    output:

        minden a mezőn álló virológusra, ha immunitás ágens védi:

            <virológus\_id> Beardence infection defended with Immunity.

minden a mezőn található virológusra, ha nem védi immunitás ágens, és nincs semmilyen más ágens hatása alatt:

        ha a virológuson van fejsze:

            <virológus\_id> Wanna use axe? (y/n)

        ha a virológuson van kesztyű, és nem használta a fejszét:

            <virológus\_id> Wanna use gloves? (y/n)

        ha a virológuson van köpeny (csak teszt üzemmódban kérdezzük hogy a köpeny sikeresen véd-e, éles játék során ez random, a felhasználó nem befolyásolhatja!!!):

            <virológus\_id> Wanna defend with cloak? (y/n)

input bármely eszköz    használata esetén (minden más bemenet esetén nem használjuk az adott eszközt):

        y

    output, ha használtunk eszközt:

        <virológus\_id> Beardence infection defended, <Axe used / Gloves usable: <szám> / Gloves lost / Cloak lost>

    output, ha a virológus nem áll immunity ágens hatása alatt, és nem használt eszközt, mert nem volt neki / nem akart / ágens hatása miatt cselekvőképtelen:

    <virológus\_id> BearDance infection successful.

    output, ha a virológus BearDance ágens hatása alatt áll:

    <virológus\_id> BearDance already infected.

        <ágens\_id> remaining time <remaining\_effecttime>

Oblivion step

    output:

<ágens\_id> remaining time 0

8. Ágens létrehozása

    input:

        createagent

    lehetséges output-ok vagy szóval elválasztva:

            GCode known for: <ágens\_classname> <ágens\_classname>…….

        vagy

            No GCode known, can’t create agent.

    input, ha van ismert genetikai kód (ha nem megfelelő a bemenet, újra kérdez):

        <ágens\_classname>

    lehetséges output-ok vagy szóval elválasztva:

            Not enough material.

        vagy

            Create agent successful.

9. Ágens kenése

    input:

        attack

output (a listában mindig lesz legalább egy elem, maga a támadó virológus, hiszen önmagára is kenhet ágenst):

        Attackable: <virológus\_id> <virológus\_id> …..

    input, ha találtunk virológust (helytelen bemenet esetén újra kérdezünk):

        <virologist\_id>

    Lehetséges output-ok vagy szóval elválasztva:

            Available Agents: <agent\_classname> <agent\_classname> …..

        vagy

            No Agent available.

    Input (nem megfelelő input esetén újra kérdez):

        <agent\_classname>

    lehetséges output-ok:

ha a virológusra Immunity hat, vagy nem BearDance ágenssel támadunk, és a virológusra valamilyen másik ágens hat:

            Virologist under <agent\_classname>, cannot get effect.

    ha a virológuson van fejsze, és nem önmagára ken ágenst:

        <virológus\_id> Wanna use axe? (y/n)

    ha a virológuson van kesztyű, és nem használta a fejszét, és nem önmagára ken ágenst:

        <virológus\_id> Wanna use gloves? (y/n)

    ha a virológuson van köpeny (csak teszt üzemmódban kérdezzük hogy a köpeny sikeresen véd-e, éles játék során ez random, a felhasználó nem befolyásolhatja!!!), és nem önmagára ken ágenst:

        <virológus\_id> Wanna defend with cloak? (y/n)

input bármely eszköz    használata esetén (minden más bemenet esetén nem használjuk az adott eszközt):

        y

    output, ha használtunk eszközt:

        <virológus\_id> <Agent\_id> infection defended, <Axe used / Gloves usable: <szám> / Gloves lost / Cloak lost>

    output, ha a virológus nem áll immunity ágens hatása alatt, és nem használt eszközt, mert nem volt neki / nem akart / ágens hatása miatt cselekvőképtelen:

    <virológus\_id> <Agent\_id> infection successful.

### UseAgentOnVirologistIAmUnderEffect

#### Leírás

Egy virológus, akire hat ágens megtámad egy másik virológust, aki vele egy mezőn áll és nem hat rá semmilyen ágens, valamint a megtámadott virológus nem használ védekező eszközt. A sikeres támadás után a megtámadott virológus az ágens hatása alá kerül.

#### Tesztelt osztályok és függvények:

* + - Virologist:
* createAgent, attack, assaulted, defense
  + - GCode:
* create

#### Várható hibahelyek: -

#### Bemenet

*createField Field*

*createVirologist Field1 true false false false 1 1 0 0 0 0 Oblivion*

*createVirologist Field1 fasle false false false 0 0 0 0 0 0*

*next*

*createAgent*

*Oblivion*

*attack*

*V2*

*Oblivion*

#### Elvárt kimenet

*New Field Field1*

*New Virologist V1*

*New Virologist V2*

*V1 turn*

*GCode known for: Oblivon*

*Create agent successful*

*Attackable: V2*

*Avaible agents: Oblivion*

*Agent used, V2 under Oblivion*

### UseAgentOnVirologistWhoHasCloak

#### Leírás

A felhasználó egy ágenst ken egy másik felhasználó virológusára. A megkent virológus köpenyével kivédi a támadást. A használt ágens megsemmisül, a támadás sikertelen

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* createAgent, attack, assoulted, defense
  + - Cloak osztály:
* evade
  + - Inventory osztály:
* addCloak
  + - GCode osztály:
* create

#### Várható hibahelyek: -

#### Bemenet

createField Field

*createVirologist Field1 true false false false 1 1 0 0 0 0 Oblivion*

*createVirologist Field1 fasle false false false 0 0 0 0 1 0*

next

equip c

*createAgent*

*Oblivion*

*attack*

*V2*

*Oblivion*

#### Elvárt kimenet

*New Field Field1*

*New Virologist V1*

*New Virologist V2*

*V1 turn*

*Equip successful*

*GCode known for: Oblivion*

*Create agent successful*

*Attackable: V2*

*Avaible agents: Oblivion*

*Agent used, V2 defended with cloak, attack unsuccessful.??*

### BearDanceInfectedVirologistInfectOtherVirologistWhoHasImmunity

#### Leírás

Egy medvetánccal megfertőzött virológus olyan mezőre lép, ahol van egy másik virológus, akit immunitás véd. Megpróbálja megfertőzni medvetánccal, de a fertőzés sikertelen.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* createAgent, attack, assoulted, defense
  + - BearDance osztály:
* step

#### Várható hibahelyek: -

#### Bemenet

*createField Field*

*createField Field*

*createVirologist Field1 false false false true 0 0 0 0 0 0*

*createVirologist Field2 true false false false 0 0 0 0 0 0*

*setNeighbours Field1 Field2*

*next*

*Field2*

#### Elvárt kimenet

*New Field Field1*

*New Field Field2*

*New Viroologist V1*

*New Virologist V2*

*Field1 neighbours: Field2*

*Field2 neighbours: Field1*

*BearDance step*

*Select neighbour: Field2*

*Moved to Field2*

*V2 Beardance infection defended with Immunity.*

### UseAgentOnVirologistWhoIsUnderEffect

#### Leírás

A felhasználó egy ágenst ken egy másik felhasználó virológusára. A megtámadott virológuson van már ágens, ezért a támadás sikertelen.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* createAgent, attack, assoulted, defense
  + - GCode osztály:
* create

#### Várható hibahelyek: -

#### Bemenet

*createField Field*

*createVirologist Field1 false false false false 0 0 0 0 0 0 Oblivion*

*createVirologist Field1 false true false false 0 0 0 0 0 0*

*next*

*createAgent*

*Oblivion*

*attack*

*V2*

*Oblivion*

#### Elvárt kimenet

*New Field Field1*

*New Virologist V1*

*New Virologist V2*

*V1 turn*

*GCode known for: Oblivon*

*Create agent successful*

*Attackable: V2*

*Avaible agents: Oblivion*

*Agent used, attack unsuccessful.*

### MoveToBasicField

#### Leírás

A virológus egy sima mezőre lép.

#### Tesztelt osztályok és függvények:

* + - Virologis osztály t:
* move
  + - Field osztály:
* arrive

#### Várható hibahelyek:

* + - Virológus nem tud lépni, mert le van bénulva (Hibeüzenet:” Virologist paralised Can not perform action.”)

#### Bemenet

*createField Field*

*createField Field*

*createVirologist Field1 false false false false 0 0 0 0 0 0*

*setNeighbour Field1 Field2*

*move*

*F2*

#### Elvárt kimenet

*New Field Field1*

*New Field Field2*

*New Virologist V1*

*Field1 neighbours: Field2*

*Field2 neighbours: Field1*

*Neighbours: Field2*

*Moved to F2*

### MoveToShelter

#### Leírás

A virológus egy óvóhelyre lép, ahol felvesz egy felszerelést az inventory-jába.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* move
  + - Shelter osztály:
* arrive

#### Várható hibahelyek: -

#### Bemenet

*createField Field*

*createField Shelter*

*createVirologist Field1 false false false false 0 0 0 0 0 0*

*setNeighbour Field1 Shelter1*

*move*

*Field2*

*y*

#### Elvárt kimenet

*New Field Field1*

*New Shelter Shelter1*

*New Virologist V1*

*Field1 neighbours: Shelter1*

*Shelter1 neighbours: Field1*

*Neighbours: Shelter1*

*Virologist moved to Shelter1*

*Wanna pick up(y/n)?*

*Equipment added to inventory.*

### MoveToInfectedLab

#### Leírás

A virológus egy fertőző laborra lép.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* move
  + - InfectedLab osztály:
* arrive

#### Várható hibahelyek: -

#### Bemenet

*createField InceftedLab*

*createField Field*

*createVirologist Field1 false false false false 0 0 0 0 0 0*

*setNeighbour Field1 InfectedLab1*

*move*

*InfectedLab1*

*y*

#### Elvárt kimenet

*New Field Field1*

*New InfectedLab InfectedLab1*

*New Virologist V1*

*Field1 neighbours: InfectedLab1*

*InfectedLab1 neighbours: Field1*

*Neighbours: InfectedLab1*

*Virologist moved to InfectedLab1*

*Wanna touchy? (y/n)*

*Known codes: 1*

*BearDance infection successful.*

### NextVirologistWhoIsBearDanceInfected

#### Leírás

Vezérlés átadása medvetáncoló virológusnak.

#### Tesztelt osztályok és függvények: -

#### Várható hibahelyek: -

#### Bemenet

*createField Field*

*createVirologist Field1 false false false true 0 0 0 0 0 0*

*next*

#### Elvárt kimenet

*New Field Field1*

*New Virologist V1*

*V1 under Beardance, skip step*

### NextVirologistWhoIsUnderOblivion

#### Leírás

Vezérlés átadása a kezdő virológusnak, aki felejtés alatt van.

#### Tesztelt osztályok és függvények: -

#### Várható hibahelyek: -

#### Bemenet

*createField Field*

*createVirologist Field1 false false false false 0 0 0 0 0 0 Immunity*

*createVirologist Field1 false false false false 0 0 0 0 0 0 Oblivion*

*next*

*createAgent*

*Oblivion*

*attack*

*V1*

#### Elvárt kimenet

*New Field Field1*

*New Virologist V1*

*V1 turn*

*Oblivion step*

### A virológus ráken egy másikra egy ágenst, fejszés verzió

#### Leírás

A virológus ráken egy másik virológusra egy ágenst, akire nem hat semmilyen ágens, de rendelkezik védőfelszerelésekkel, amik közül a fejszét használja.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* createAgent, attack, assaulted, defense, die

#### Várható hibahelyek:

* + - Virológusok rossz mezőn találhatóak (nem a paraméterként megadott mezőn)
    - V2 nem tudja felvenni a felszereléseket
    - V1 nem tud ágenst létrehozni
    - V1 nem éri el V2-t
    - V1 nem tud fejszével védekezni
    - V1 fejszével védekezés után is megfertőződik
    - V1 fejszéje nem csorbul ki

#### Bemenet

*create Field Field*

*create Virologist Field1 false false false false 0 0 1 1 1 1*

*create Virologist Field1 false false false false 1 1 0 0 0 0 Paralysis*

*next*

*equip a*

*equip g*

*equip c*

*next*

*createagent*

*Paralysis*

*y*

#### Elvárt kimenet

*Field1*

*V1*

*V2*

*V1 turn*

*Equip successful.*

*Equip successful.*

*Equip successful.*

*V2 turn*

*GCode known for: Paralysis*

*Create agent successful.*

*Attackable: V1, V2*

*Available agents: Paralysis*

*V1 Wanna use axe?*

*V1 P1 defended, Axe used.*

### A virológus ráken egy ágenst egy másik virológusra, cloak nem sikerül verzió *(UseAgentOnVirologist cloak nem védi ki -> ezekben benne lehet a beardance is (védekező használja))*

#### Leírás

A virológus ráken egy másik virológusra egy ágenst, akire nem hat semmilyen ágens, de rendelkezik védőfelszerelésekkel, amik közül egyiket sem használja, a védőköpenye pedig sikertelenül védi meg.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* createAgent, attack, assaulted, defense metódusok

#### Várható hibahelyek:

* + - V1 vagy V2 nem az F1-en találhatóak
    - V2 kezdi a játékot a next paranccsal
    - V1 ágens készítése sikertelen
    - V1 nem éri el V2-t
    - V2 védekezik fejszével vagy kesztyűvel
    - V2 cloak védekezése sikeres
    - V2 cloak védekezése véletlenszerű (van olyan teszt lefutás, ahol sikeres és van olyan teszt lefutás, ahol nem)

#### Bemenet

*create Field Field*

*create Virologist Field1 false false false false 0 0 1 1 1 1*

*create Virologist Field1 false false false false 1 1 0 0 0 0 Paralysis*

*next*

*equip a*

*equip g*

*equip c*

*next*

*createagent*

*Paralysis*

*attack*

*V1*

*Paralysis*

*n*

*n*

*y*

#### Elvárt kimenet

*Field1*

*V1*

*V2*

*V1 turn*

*Equip successful.*

*Equip successful.*

*Equip successful.*

*V2 turn*

*GCode known for: Paralysis*

*Create agent successful*

*Attackable: V1, V2*

*Available agents: Paralysis*

*V1 Wanna use axe? (y/n)*

*V1 Wanna use gloves? (y/n)*

*V1 Wanna defend with cloak? (y/n)*

*V1 P1 infection successful.*

### A virológus alapanyag nélkül hoz létre ágenst

#### Leírás

A virológus megpróbál létrehozni egy ágenst a meglevő alapanyagok nélkül.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* createAgent

#### Várható hibahelyek:

* + - V1 nem ismeri a Paralysis genetikai kódot
    - V1 létrehozza az ágenst alapanyagok hiányában is
    - V1 rendelkezik alapanyagokkal a felszerelésében

#### Bemenet

*create Field Field*

*Create Virologist F1 false false false false 0 0 0 0 0 0 Paralysis*

*next*

*createagent*

*Paralysis*

#### Elvárt kimenet

*Field1*

*V1*

*V1 turn*

*GCode known for: Paralysis*

*Create agent unsuccessful: not enough materials.*

### A virológus magára ken egy ágenst

#### Leírás

A virológus ráken saját magára egy ágenst, úgy, hogy nem hat rá semmilyen ágens.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* createAgent, attack

#### Várható hibahelyek:

* + - V1 nem tudja létrehozni az Immunityt
    - V1 léterhozás után nem rendelkezik Immunityvel
    - V1 nem éri el magát, ezért nem tudja magára kenni az ágenst

#### Bemenet

*create Field Field*

*create Virologist Field1 false false false false 2 4 0 0 0 0 Immunity*

*next*

*createagent*

*Immunity*

*attack*

*V1*

*Immunity*

#### Elvárt kimenet

*Field1*

*V1*

*V1 turn*

*GCode known for: Immunity*

*Create agent successful*

*Attackable: V1*

*Available agents: Immunity*

*V1 I1 infection successful.*

### A virológus mozogni próbál bénult állapotban

#### Leírás

A virológus bénult állapotban van, amikor sorra kerül, és megpróbál lépni, de nem tud.

#### Tesztelt osztályok és függvények:

* + - Game osztály:
* next

#### Várható hibahelyek:

* + - V1 nem bénult állapotban jön létre
    - V1 lépése sikeres lesz bénult állapota ellenére

#### Bemenet

*create Field Field*

*create Virologist Field1 false true false false 0 0 0 0 0 0*

*next*

#### Elvárt kimenet

*V1 under Paralysis, skip step*

*P1 step*

*P1 remaining time 2*

*V1 under Paralysis, skip step*

*P1 step*

*P1 remaining time 1*

*V1 under Paralysis, skip step*

*P1 step*

*P1 remaining time 0*

### A virológus laborra lép

#### Leírás

A virológus egy labor mezőre lép, nem találkozik másik virológussal, letapogatja az ottani kódot, nem fertőződik meg.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* move
  + - Lab osztály:
* arrive, wannaTouchy

#### Várható hibahelyek:

* + - V1 nem tud átlépni L1-re
    - V1 nem tud L1-ben kódot tapogatni
    - V1 megfertőződik medvevírussal a laborban

#### Bemenet

*create Field Field*

*create Field Storage*

*netneighbor Field1 Storage1*

*create Virologist Field1 false false false false 0 0 0 0 0 0*

*create Virologist Storage 1 false true false false 0 0 0 0 0 0*

*next*

*move*

*Laboratory1*

*y*

#### Elvárt kimenet

*Field1*

*Laboratory1*

*Field1 neighbors: Laboratory1*

*Laboratory1 neighbors: Field1*

*V1*

*Neighbors: Laboratory1*

*Virologist moved to Laboratory1.*

*Wanna touchy? (y/n)*

*V1 won.*

### Virológus rálép egy raktárra, találkozik egy bénult virológussal

#### Leírás

A virológus rálép egy raktárra, ahol van egy bénult virológus, lop tőle, majd felvesz alapanyagot.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* move, stealBy
  + - Storage osztály:
* arrive, wannapickup, wannasteal

#### Várható hibahelyek:

* + - Virológusok nem a megfelelő helyen jönnek létre
    - A keletkező mezők nem szomszédosak
    - V1 nem tud átlépni a raktárra
    - V1 nem tud lopni V2-től, mert V2 nem bénult
    - V1 nem tud felvenni alapanyagot
    - V1 alapanyagfelvétele véletlenszerű (nem aminosavat vett fel)

#### Bemenet

*create Field Field*

*create Field Storage*

*setneighbor Field1 Storage1*

*create Virologist Field1 false false false false 0 0 0 0 0 0*

*create Virologist Storage1 false true false false 0 0 0 0 0 0*

*next*

*move*

*Storage1*

*y*

*Axe*

*y*

*Aminoacid*

#### Elvárt kimenet

*Field1*

*Storage1*

*Field1 neighbors: Storage1*

*Storage1 neighbors: Field1*

*V1*

*V2*

*V1 turn*

*Neighbors: Storage1*

*Virologist moved to Storage1.*

*V2 paralyzed. Wanna steal? (y/n)*

*What to steal? (Aminoacid/Nukleotid/Axe/Gloves/Cloak/Sack)*

*Axe not found.*

*Wanna pick up? (y/n)*

*What to pick up?*

### *next* parancs vitustáncos virológusnál

#### Leírás

A virológus vitustánc hatása alatt van, ami miatt nem jön sorra, hanem véletlenszerűen egy szomszédos mezőre lép. A teszt módban azonban megkérdezi a felhasználót, hogy melyik mezőre léptesse a vírus a virológust.

#### Tesztelt osztályok és függvények:

* + - Game osztály:
* next
  + - VitusDance osztály:
* step

#### Várható hibahelyek:

* + - V1 nincs vitustánc hatása alatt
    - V1 sorra kerül, tehát tud lépni
    - V1 lépése a vitustánc során véletlenszerű

#### Bemenet

*create Field Field*

*create Field Field*

*setneighbor Field1 Field2*

*create Virologist Field1 false false true false 0 0 0 0 0 0*

*next Field2*

#### Elvárt kimenet

*Field1*

*Field2*

*Field1 neighbors: Field2*

*Field2 neighbors: Field1*

*V1*

*V1 under VitusDance, skip step*

*Vitus1 step*

*Select neighbor: Field2*

*Moved to Field2.*

*Vitus1 remaining time 5.*

*V1 under VitusDance, skip step, stb...*

### UseAgentOnVirologistDoNotDefense

#### Leírás

Egy virológus megtámad egy vele azonos mezőn álló másik virológust, akire nem hat semmilyen másik ágens, és a védekezéshez nem használ semmilyen eszközt. A sikertelen védekezés után a megtámadott virológus az ágens hatása alá kerül.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* createAgent, attack, assaulted, defense
  + - GCode osztály:
* create

#### Várható hibahelyek

* + - Virológus létrehozásánál lehetséges hibák:
* Virologist not created, too many material.
* Virologist not created, too many equipment.
* Virologist not created, invalid Field Id.
  + - Ágens létrehozásánál lehetséges hibák:
* No GCode known, can’t create agent.
* Not enough material.
  + - Kenés során lehetséges hibák:
* No Agent available.

#### Bemenet

*create Field Field  
create Virologist Field1 false false false false 1 1 0 0 0 0 Paralysis  
create Virologist Field1 false false false false 0 0 0 0 0 0  
next  
createagent  
Paralysis  
attack  
V2  
Paralysis*

#### Elvárt kimenet

*New Field Field1  
New Virologist V1  
New Virologist V2  
V1 turn  
GCode known for: Paralysis  
Create agent successful  
Attackable: V1, V2*

*Available Agents: Paralysis  
Agent used, V2 under Paralysis*

### UseAgentOnVirologistDefenseWithGloves

#### Leírás

Egy virológus megtámad egy vele azonos mezőn álló másik virológust, akire nem hat semmilyen másik ágens, és a megtámadott virológus kesztyűvel védekezik. A támadó nem védekezik és nem áll ágens hatása alatt. Így a támadó az ágens hatása alá kerül.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* equipAxe, equipGloves, createAgent, attack, assaulted, defense
  + - GCode osztály:
* create

#### várható hibahelyek

* + - Virológus létrehozásánál lehetséges hibák:
* Virologist not created, too many material.
* Virologist not created, too many equipment.
* Virologist not created, invalid Field Id.
  + - Equip során lehetséges hibák:
* Too many equipment.
* Equipment not found in inventory.
  + - Ágens létrehozásánál lehetséges hibák:
* No GCode known, can’t create agent.
* Not enough material.
  + - Kenés során lehetséges hibák:
* No Agent available.

#### Bemenet

*create Field Field  
create Virologist F1 false false false false 1 1 0 0 0 0 Paralysis  
create Virologist F1 false false false false 0 0 1 1 0 0  
next  
next*

*equip a*

*equip g*

*next*

*createagent*

*Paralysis*

*Attack*

*V2*

*n*

*y*

#### Elvárt kimenet

*New Field F1  
New Virologist V1  
New Virologist V2*

*V1 turn*

*V2 turn*

*Equip successful  
Equip successful  
V1 turn*

*GCode known for: Paralysis*

*Create agent successful*

*Attackable: V1, V2*

V2 Wanna use axe? (y/n)

V2 Wanna use gloves? (y/n)

V2 Paralysis infection defended, Gloves usable: 2

V1 Paralysis infection successful.

### BearDanceOverwriteAgent

#### Leírás

Egy Virológus BearDance mozgása során megfertőz egy olyan másik virológust, aki le van bénulva. (A teszt során létrehozunk egy harmadik virológust is, hogy legyen kinek átadni a vezérlést.) Először az első virológus ágensének step metódusa hívódik meg, ennek során V2 megfertőződik, V2-re ható ágens így BearDance lesz. Majd a második virológus ágensének step metódusában így már ismét egy BearDance step hívódik meg.

#### Tesztelt osztályok és függvények:

* + - Game osztály:
* next
  + - BearDance osztály:
* step

#### várható hibahelyek

* + - Virológus létrehozásánál lehetséges hibák:
* Virologist not created, too many material.
* Virologist not created, too many equipment.
* Virologist not created, invalid Field Id.
* Ellenőrizni kell, hogy a BearDance sikeresen felülírta-e a Paralysis ágenst. Ha ez nem történik meg, az hiba.

#### Bemenet

*create Field Field*

*create Field Field*

*create Field Field*

*setneighbor Field1 Field2  
create Virologist F1 false false false true 0 0 0 0 0 0  
create Virologist F2 false true false false 0 0 0 0 0 0*

*create Virologist F3 false false false false 0 0 0 0 0 0*

*next*

*next*

*Field2*

*Field1*

#### Elvárt kimenet

*New Field Field1*

*New Field Field2*

*New Field Field3*

*Field1 neighbors: Field2*

*Field2 neighbors: Field1*

*New Virologist V1*

*New Virologist V2*

*V1 under BearDance, skip step.*

*V2 under Paralysis, skip step.*

*V3 turn*

*BearDance step  
Select neighbor: Field2*

*Moved to Field2*

*V2 BearDance infection successful.*

*BearDance step*

*V1 BearDance already infected.*

*Select neighbor: Field1*

*Moved to Field1*

*V1 under BearDance, skip step.*

*V2 under BearDance, skip step.*

*V3 turn.*

### SuccessfulCreateAgent

#### Leírás

VitusDance ágens sikeres létrehozása, ami akkor történhet meg, ha a virológus ismeri az ágenshez tartozó genetikai kódot, illetve van elég nyersanyag az inventoryjában.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* createAgent
  + - GCode osztály:
* create

#### Várható hibahelyek

* + - Virológus létrehozásánál lehetséges hibák:
* Virologist not created, too many material.
* Virologist not created, too many equipment.
* Virologist not created, invalid Field Id.
  + - Ágens létrehozásánál lehetséges hibák:
* No GCode known, can’t create agent.
* Not enough material.

#### Bemenet

*create Field Field  
create Virologist F1 false false false false 2 1 0 0 0 0 VitusDance*

*next*

*createagent*

*VitusDance*

#### Elvárt kimenet

*New Field Field1*

*New Virologist V1*

*V1 turn.*

*GCode known for: VitusDance*

*Create agent successful.*

### SelfUseAgent

#### Leírás

Egy virológus, akire kezdetben nem hat semmilyen ágens, megtámadja önmagát az immunitás ágenssel. Az ágens hatása védekezés nélkül a virológusra kerül.

#### Tesztelt osztályok és függvények:

* + - Virologist osztály:
* createAgent, attack, assaulted
  + - GCode osztály:
* create

#### Várható hibahelyek

* + - Virológus létrehozásánál lehetséges hibák:
* Virologist not created, too many material.
* Virologist not created, too many equipment.
* Virologist not created, invalid Field Id.
  + - Ágens létrehozásánál lehetséges hibák:
* No GCode known, can’t create agent.
* Not enough material.
  + - Kenés során lehetséges hibák:
* No Agent available.

#### Bemenet

*create Field Field  
create Virologist Field1 false false false false 2 4 0 0 0 0 Immunity  
next  
createagent  
Immunity  
attack  
V1  
Immunity*

#### Elvárt kimenet

*New Field Field1  
New Virologist V1   
V1 turn  
GCode known for: Immunity  
Create agent successful  
Attackable: V1*

*Available Agents: Immunity  
Agent used, V1 under Immunity*

### MoveWhenDancing

#### Leírás

A virológus Dancing közben sorra kerülne, de ágens hatása alatt nem tud cselekedni. (A teszt során egy második virológust is létrehozunk, hogy legyen kinek átadni a vezérlést.

#### Tesztelt osztályok és függvények:

* + - Game osztály:
* next

#### várható hibahelyek

* + - Hiba, ha BearDance hatása alatt álló virológus mégis sorra kerül.

#### Bemenet

*create Field Field  
create Virologist Field1 false false true false 0 0 0 0 0 0*

*create Virologist Field1 false false false false 0 0 0 0 0 0  
next*

#### Elvárt kimenet

*New Field Field1  
New Virologist V1*

*New Virologist V2   
V2 under BearDance, skip step.*

*V3 turn.*

### MoveToStorage

#### Leírás

A virológus kezdetben egy üres mezőn áll, majd egy Storage mezőre lép, ahol felvesz egy material-t.

#### Tesztelt osztályok és függvények:

* + - Field osztály:
* setNeighbour
  + - Viologist osztály:
* move
  + - Storage osztály:
* arrive, wannaPickUp, addMat

#### várható hibahelyek

* + - Szomszédos mezőre lépésnél hibalehetőség:
* Virologist paralyzed. Can not perform action.
* Virologist beardancing. Can not perform action.
* Virologist vitus dancing. Can not perform action.
  + - Anyag felvétele közben hibalehetőség:
* Inventory full.

#### Bemenet

*create Field Field*

*create Field Storage*

*setneighbor Field1 Storage1*

*create Virologist Field1 false false false false 0 0 0 0 0 0*

*next*

*move*

*Storage1*

*y*

*Aminoacid*

#### Elvárt kimenet

*New Field Field1*

*New Field Storage1*

*Field1 neighbors: Storage1*

*Storage1 neighbors: Field1*

*New Virologist V1*

*V1 turn.*

*Neighbors: Storage1*

*Virologist moved to Storage1*

*Wanna pick up? (y/n)*

*What to pick up? (Aminoacid/Nukleotid)  
Aminoacid added to inventory.*

### NextHealthyVirologist

#### Leírás

A virológus sorra kerül, megkapja a vezérlést.

#### Tesztelt osztályok és függvények:

* + - Game osztály:
* next

#### Várható hibahelyek

* + - Hiba, ha a virológus nem kerül sorra, nem kapja meg a vezérlést.

#### Bemenet

*create Field Field  
create Virologist Field1 false false true false 0 0 0 0 0 0*

*next*

#### Elvárt kimenet

*New Field Field1  
New Virologist V1*

*V1 turn.*

### NextIfParalyzed

#### Leírás

A virológus bénultság közben sorra kerülne, de Paralyzed ágens hatása alatt nem tud cselekedni. (A teszt során egy második virológust is létrehozunk, hogy legyen kinek átadni a vezérlést.

#### Tesztelt osztályok és függvények:

* + - Game osztály:
* next

#### várható hibahelyek

* + - Hiba, ha Paralyzed hatása alatt álló virológus mégis sorra kerül.

#### Bemenet

*create Field Field  
create Virologist Field1 false true false false 0 0 0 0 0 0*

*create Virologist Field1 false false false false 0 0 0 0 0 0  
next*

#### Elvárt kimenet

*New Field Field1  
New Virologist V1*

*New Virologist V2   
V1 under Paralyzed, skip step.*

V3 turn.

## A tesztelést támogató programok tervei

### Egy teszt lefutása

A tesztelés úgy fog lezajlani, hogy a program tartalmazni fog egy külön részt, amely kifejezetten a tesztekkel foglalkozik, így indítást követően az elindul, a megadott sorrendbe beolvassa a tesztfájlokat, meg beadja az első teszthez a megadott bemenetet, és elkészíti a kimenetet hozzá.

### Tesztek lefutásához program

A tesztekhez JUnit könyvtárat fogunk használni, ugyanis az lett számunkra a legkézenfekvőbb és nem kell különösebben új programot feltelepíteni, engedélyezni dolgokat a gépen.

### Elvárt és tényleges kimenet összehasonlítása

A tesztfájlok beolvasása azért szükséges, mert azokban vannak tárolva az elvárt kimenetek, így, mikor a program legenerálja azokat, össze tudjuk hasonlítani az elvárt (tesztfájlos) kimenetet a kigeneráltal. Ha a kettő azonos, akkor sikeres volt a tesztünk, ha eltérés mutatkozik, akkor meg sajnos hibára futottunk.

## Ütemterv

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

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevő(k)** | **Leírás** |

|  |  |  |  |
| --- | --- | --- | --- |
| 2022.04.07 20:30 | 5 óra | Alpek, Litavecz, Marton, Ruskó | Értekezlet: Új feladat átbeszélése, tervek elkészítése a feladat teljesítéséhez. |
| 2022.04.09 12:00 | 7 óra | Litavecz | Feladat: Az osztályok és metódusok rész megírása. |
| 2022.04.11 18:00 | 4,5 óra | Alpek, Csia, Marton, Ruskó | Értekezlet: Bemeneti nyelv alapjainak leszögezése, tesztek megírásához sablon elkészítése, feladatok szétosztása egymás közt. Csia: grafikus felülettel már előre kalkulálás, és azzal tervezni mostani kódhoz. |
| 2022.04.12. 00:00 | 6 óra | Csia | Feladat: Dokumentum összeállítása, Litavecz feladatrészének átnézése, maradék metódusrész kibővítése, hiányzó osztály elhelyezése. A 8.3-as rész és a Napló megírása. |
| 2022.04.12. 16:00 | 7 óra | Alpek | Feladat: Tesztesetek elkészítése a hozzájuk tartozó leírásokkal, illetve bemeneti/kimeneti részekkel. (Fejenként 29 db.) |
| 2022.04.12. 18:00 | 7 óra | Marton | Feladat: Tesztesetek elkészítése a hozzájuk tartozó leírásokkal, illetve bemeneti/kimeneti részekkel. (Fejenként 29 db.) |
| 2022.04.12. 15:00 | 7 óra | Ruskó | Feladat: Tesztesetek elkészítése a hozzájuk tartozó leírásokkal, illetve bemeneti/kimeneti részekkel. (Fejenként 29 db.) |
| 2022.04.13. 10:00 | 1 óra | Csia | Feladat: A dokumentum összeillesztése a maradék munkákkal, egységesítése, a dokumentum felöltése, leadása. |

## Százalékos teljesítés

|  |  |
| --- | --- |
| **Név** | **Százalék** |

|  |  |
| --- | --- |
| Alpek | 20% |
| Csia | 20% |
| Litavecz | 20% |
| Marton | 20% |
| Ruskó | 20% |