ÖSSZEFOGLALÁS

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

2022. 05. 18.

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

Jelen dokumentum a „Zeta” csapat „A világtalan virológusok világa” implementációjával kapcsolatos általános tudnivalókat foglalja össze.

2.1.2 Szakterület

A készítendő szoftver célja a Szoftver projekt laboratórium tárgy oktatói által megálmodott „A világtalan virológusok világa” című játék megvalósítása, az oktatók és a játék fejlesztőinek minél nagyobb megelégedésére és élményére koncentrálva. Másodlagos célkitűzés a gondolkodtató játékok újabb példányának létrehozása, igények kielégítése.

2.1.3 Definíciók, rövidítések

ill.: illetve
ld.: lásd
pl.: például

2.1.4 Hivatkozások

Szoftvertechnológia órai jegyzet és előadásdiák
Programozás alapjai 3 órai jegyzet és előadásdiák

2.1.5 Összefoglalás

A továbbiakban részletesebben is ismertetésre kerülnek a készítendő szoftver sajátosságai, mind a leendő felhasználók, mind a fejlesztők szempontjából, a számukra érdekes aspektusokat részletezve (pl.: funkciók, ill. követelmények), valamint a szoftver megvalósításával kapcsolatos információk, tervek.

2.2 Áttekintés

2.2.1 Általános áttekintés

A program struktúrája alapvetően két részből áll: a pálya elemeiből és a pályán lévő dolgokból. A pálya elemei alatt sokszögek értendők, mely lehet szabad terület, raktár, óvóhely vagy laboratórium. A pályán lévő dolgok lehetnek játékosok, védőfelszerelések, anyagok, ill. genetikai kódok. Természetesen a pályán való mozgáshoz a pályaelemeknek és a pályán lévő dolgoknak kommunikálniuk kell egymással, ehhez mindkettő egy interfészt valósít meg. A felhasználó a játékban szereplő munkások közül egyet irányíthat, így részt vehet a játékban.

2.2.2 Funkciók

A szoftver egy olyan számítógépes játék, amellyel egyszerre több játékos tud játszani egymás ellen. A játék világa egy város lesz, ahol a játékosok vak virológus karakterek bőrébe bújhatnak.

A pálya egy sokszög alapú generált pálya, mely éleivel csatlakozik össze a szomszédos mezővel. A pálya maga egy várost jelenít meg, melyen különböző típusú mezők találhatóak. A szabad területen nem található semmi. A raktárakban aminosavak és nukleotidok (közös nevükön anyag) gyűjthetőek, melyek szükségesek az ágensek legyártásához. Ezekből a játékos csak korlátozott mennyiséget gyűjthet össze. A játékosnak továbbá eszköz gyűjtésére is lehetősége van, melyeket az óvóhelyeken tud aktiválni magán. Az eszközök különféle képességekkel bírnak, melyek között szerepel a tárolókapacitás bővítése (zsák), az ágenseket távol tartó eszköz, mely 82.3%-os hatásfokkal működik (köpeny), illetve az ellenfélre támadásnál ágenst visszakenő eszköz (kesztyű). Azonban a játékos egyszerre maximum 3 felszerelést viselhet. A laboratóriumokban a játékosok genetikai kódokat gyűjthetnek össze, melyek a

későbbiekben szükségesek lesznek az ágensek legyártásához. Egy genetikai kódot a játékosnak elég egyszer megtanulnia. Ha egy játékos rendelkezik a megfelelő genetikai kóddal és a megfelelő mennyiségű anyaggal ágenst (mely lehet vírus vagy vakcina is, de a továbbiakban mi ezt egynek vesszük, mert nincs semmilyen eltérő viselkedésük vagy tulajdonságuk, leszámítva a nevüket) gyárthat le. Egy ágens, az alapanyagoktól függően, különböző képességekkel bír, mely a játékosokra van hatással bizonyos ideig. Van olyan ágens, amely randomizált mozgást okoz az ellenfélnek (vitustánc), van, amelyik megbénítja az ellenfelet és teljesen cselekvőképtelenné teszi (bénító), van, amelyik emlékezetet töröl, így az ellenfél minden eddig megtanult kódot elfelejt és végül van a védő ágens, amely megvédi a játékost más játékosok rossz ágenseitől (immunitás).

A játék nehézségét a többi játékos adja meg, mely miatt szükséges a játékban az eszközök és az ágensek használata.

A játékmenet véget ér, ha bármelyik játékos sikeresen begyűjt/letapogat minden pályán létező genetikai kód közül egyet.

2.2.3 Felhasználók

A felhasználóknak a szoftver használatához nincs szükségük különösebb előképzettségre. A programot egyszerre többen is használhatják.

2.2.4 Korlátozások

A programnak egy játéktól elvárható mértékben stabilan és helyesen kell működnie: nem fagyhat le, nem léphet ki a felhasználó akarata nélkül és a követelményekben meghatározott módon kell játszania és a felhasználókat játszani engednie.

2.2.5 Feltételezések, kapcsolatok

Az előadások anyagait és a jegyzeteinket a projekt szakszerűbb átgondolásához használtuk.

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case
R00	A játékos irányítja a virológust a városban.	Bemutatás	Alapvető	Feladatkiírás	Mozgás, Város megtekintése, Város létrehozása
R01	A virológus látja a mezőjével szomszédos mezőket, azok tartalmát.	Bemutatás, Kiértékelés	Opcionális	Csapat	Mozgás, Város megtekintése
R02	A virológus nem tud mozogni, ha le van bénulva.	Bemutatás	Alapvető	Feladatkiírás	Mozgás
R03	A labort tartalmazó mezőkön találhatóak genetikai kódok.	Bemutatás	Alapvető	Feladatkiírás	Város megtekintése, Város létrehozása

R04	Az óvóhelyeken védőfelszerelések vannak (kesztyű, zsák köpeny).	Bemutatás	Alapvető	Feladatkiírás	Város megtekintése, Város létrehozása
R05	A raktárak tartalmazzák az aminosavakat, illetve nukleotidokat, különböző mennyiségben.	Bemutatás	Alapvető	Feladatkiírás	Város megtekintése, Város létrehozása
R06	A virológus egy körben 1 mezőnyit léphet.	Bemutatás	Opcionális	Csapat	Mozgás
R07	Egy mezőn egyszerre több virológus tartózkodhat.	Bemutatás, Kiértékelés	Fontos	Csapat	Mozgás
R08	A város eltérő oldalszámú sokszögekből álló rácsot alkot.	Bemutatás	Alapvető	Feladatkiírás	Mozgás, Város megtekintése, Város létrehozása
R09	A szabad terület egy „üres” mező.	Bemutatás	Alapvető	Feladatkiírás	Város megtekintése, Város létrehozása
R10	Egy mezőn pontosan egy szabad terület / raktár / óvóhely / laboratórium lehet.	Bemutatás	Alapvető	Csapat	Város megtekintése, Város létrehozása
R11	Egy ismert kód alapján a játékos egy ágenst tud készíteni.	Bemutatás, Kiértékelés	Alapvető	Feladatkiírás	Ágens generálás
R12	Ágens készítéshez megfelelő mennyiségű aminosavval és nukleotiddal kell rendelkeznie a virológusnak.	Bemutatás	Alapvető	Feladatkiírás	Ágens generálás
R13	Minden ágens más mennyiségű anyagot használ fel készítéskor.	Bemutatás	Opcionális	Csapat	Ágens generálás

R14	A felejtés egy olyan ágens, amelytől az áldozat elfelejti a már megismert genetikai kódokat.	Bemutatás	Alapvető	Feladatkiírás	Ágens generálás, Kenés
R15	Egy ágens csak egy adott időkorláton belül használható fel, azután eltűnik.	Bemutatás, Kiértékelés	Alapvető	Feladatkiírás, Csapat	Ágens vezérlése
R16	A kesztyűvel rendelkező virológusnak rövid időn belül választania kell, hogy visszadobja-e az ágenst a kenőre.	Bemutatás, Kiértékelés	Opcionális	Csapat	Ágens vezérlése
R17	Ha a kesztyűvel rendelkező virológus a rendelkezésre álló idő alatt nem választ, hogy vissza szeretné-e dobni az ágenst, akkor nem dobja vissza.	Bemutatás, Kiértékelés	Opcionális	Csapat	Ágens vezérlése
R18	A virológus egy ágenst ken egy másik virológusra, vagy önmagára.	Bemutatás	Alapvető	Feladatkiírás	Kenés
R19	A kenéshez a virológusnak meg kell tudnia érinteni az áldozatot.	Bemutatás	Alapvető	Feladatkiírás	Kenés
R20	Az érintés elsődleges feltétele az, hogy a 2 virológus egy közös mezőn áll.	Bemutatás, Kiértékelés	Fontos	Csapat	Kenés
R21	A virológus a kenés során elhasználja az eszköztárából a felkent ágenst.	Bemutatás, Kiértékelés	Opcionális	Csapat	Kenés, Ágens vezérlése
R22	Miközben a virológusra egy ágens hatással van, minden más, újonnan felkent ágens hatástalan marad.	Bemutatás, Kiértékelés	Opcionális	Csapat	Kenés

R23	A felkent ágensek adott ideig hatnak.	Bemutatás, Kiértékelés	Alapvető	Feladatkiírás	Kenés, Ágens vezérlése
R24	Sokféle ágens létezik, amelyek különböző hatással rendelkeznek.	Bemutatás	Alapvető	Feladatkiírás	Ágens vezérlése
R25	A vitustánc ágens hatása alatt álló virológus kontrollálhata tlan, véletlenszerű mozgással halad.	Bemutatás	Alapvető	Feladatkiírás	Kenés, Ágens vezérlése, Mozgás
R26	Az immunitás ágens megvédi a virológust attól, hogy más ágensek hatással legyenek rá.	Bemutatás	Alapvető	Ágens viselkedésének forrása: Feladatkiírás Immunitás megnevezés forrása: Csapat	Kenés, Ágens vezérlése
R27	A bénító ágens alatt álló virológus semmilyen tevékenységet (mozgás, ágens létrehozás).	Bemutatás, Kiértékelés	Alapvető, Fontos, Opcionális	Ágens viselkedésének forrása: Feladatkiírás Bénító megnevezés forrása: Csapat	Kenés, Ágens vezérlése, Mozgás
R28	A virológus, melyre ágenst kennek, a kenés során különböző, az általa viselt eszközökkel védheti magát.	Bemutatás	Alapvető	Feladatkiírás	Kenés, Gyűjtés
R29	Egyszerre egy virológus maximum 3 különböző eszközt viselhet.	Bemutatás	Alapvető	Feladatkiírás	Gyűjtés
R30	A védőköpeny 82.3%-os hatásfokkal védi meg viselőjét az ágensek hatásaitól.	Bemutatás	Alapvető	Feladatkiírás	Kenés, Gyűjtés
R31	A kesztyűvel az ágens a kenőre visszadobható.	Bemutatás	Alapvető	Feladatkiírás	Kenés, Gyűjtés

R32	Amennyiben a kenő is rendelkezik kesztyűvel, ő már nem dobhatja vissza a rá visszadobott ágenst, csak egyszeri visszadobás lehetséges.	Bemutatás, Kiértékelés	Opcionális	Csapat	Kenés, Gyűjtés
R33	A játékos a laborokban le tudja tapogatni a genetikai kódokat, melyeket ezáltal begyűjti.	Bemutatás	Alapvető	Feladatkiírás	Gyűjtés
R34	Minden kódot csak egyszer kell megtanulnia/begyűjtenie a játékosnak. Felejtőágens esetén ez inicializálódik.	Bemutatás, Kiértékelés	Alapvető	Feladatkiírás	Gyűjtés, Kenés
R35	A játékos képes kétféle anyag gyűjtésére is, melyeket elraktároz magánál, ameddig fel nem használja őket ágenskészítésre.	Bemutatás	Alapvető	Feladatkiírás	Gyűjtés, Ágensgenerálás
R36	A játékos csak fix mennyiségű anyagot tárolhat magánál. (Amekkora az éppeni tárolókapacitása).	Bemutatás, Kiértékelés	Fontos	Feladatkiírás, Csapat	Gyűjtés
R37	Egy különleges eszköz segítségével (zsák) meg tudja növelni a tárolókapacitását.	Bemutatás	Alapvető	Feladatkiírás	Gyűjtés
R38	A játékos az óvóhelyeken tudja csak begyűjteni (és ezáltal fel is venni/aktiválni) az új eszközöket.	Bemutatás, Kiértékelés	Opcionális	Csapat	Gyűjtés, Város megtekintése, Mozgás
R39	A begyűjtött eszközök automatikusan aktiválódnak a karakteren.	Bemutatás, Kiértékelés	Opcionális	Csapat	Gyűjtés

R40	Ha a játékos egy lebénult játékossal találkozik, akkor tárolókapacitásától függően begyűjtheti tőle (ellophatja) a másik játékos által gyűjtött dolgokat.	Bemutatás, Kiértékelés	Fontos	Feladatkiírás	Gyűjtés, Mozgás
R41	Az anyagok raktárakban gyűjthetőek be.	Bemutatás	Alapvető	Feladatkiírás	Gyűjtés
R42	Egy raktárban csak egyféle anyagtípus található meg.	Bemutatás, Kiértékelés	Fontos	Csapat	Gyűjtés

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R100	A program Java nyelven készül.	Bemutatás	Alapvető	Feladatkiadó	
R101	Az elkészült programkódnak fordíthatónak kell lennie Java SE Developer Kit 8-on.	Kiértékelés, bemutatás	Fontos	Feladatkiadó	
R102	A termékhez ajánlott operációs rendszer a Windows 10.	Bemutatás	Opcionális	Csapat	Bár az elkészült termék a programnyelv választása miatt elvileg crossplatform, az egyszerűség kedvéért, csak Windowsra garantáljuk a helyes működést.

R103	A szoftver fő beviteli eszközei az egér és a billentyűzet.	Bemutatás	Alapvető	Csapat	

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás
R200	Minden alapvető követelménynek teljesülnie kell.	Bemutatás	Fontos	Feladat
R201	A szoftver futtatásához szükséges a Java Runtime Environment 8 megléte.	Bemutatás	Alapvető	Feladat

2.4 Lényeges use-case-ek 2.4.1 Use-case leírások

Use-case neve	Mozgás
Rövid leírás	A játékos irányítja a karakter a városban.
Aktorok	Felhasználó
Főforgatókönyv	**1.** Szomszédos mezőre lép a játékos.
Alternatív forgatókönyv	**1. A** Ha a játékos le van bénulva, akkor nem tud mozogni, ameddig le nem jár az effekt róla.

Use-case neve	Kenés
Rövid leírás	A játékos egy ágenst használ.
Aktorok	Felhasználó
Főforgatókönyv	**1.** A játékos az előállított ágenst meghatározott időn belül felhasználja egy másik játékosra vagy saját magára hatással lesz.
Alternatív forgatókönyv	**1. A 1.** Ha a másik játékost immunitás védi, így a kenés hatástalan lesz rá.

Alternatív forgatókönyv	**1. A 2.** Ha a másik játékost kesztyű védi, akkor az a játékos eldöntheti, hogy elfogadja-e az adott ágenst vagy sem.
Alternatív forgatókönyv	**1. B** Ha immunitás védi a játékost, így a kenés hatástalan lesz rá.
Alternatív forgatókönyv	**1. C** Ha a játékos le van bénulva, nem tudja ezt az akciót végrehajtani.
Alternatív forgatókönyv	**1. D** Ha a játékos egy másik ágens hatása alatt áll, akkor az új ágens hatástalan lesz.

Use-case neve	Ágensgenerálás
Rövid leírás	A játékos a megszerzett anyagokból létrehoz egy ágenst.
Aktorok	Felhasználó
Főforgatókönyv	**1.** Ha a játékosnak megvan a megfelelő mennyiségű anyaga meg az ágenshez megfelelő kód, akkor elkészíti az ágenst.
Alternatív forgatókönyv	**1. A** Ha nincs meg a megfelelő mennyiség valamelyik kritériumból, akkor nem tudja elkészíteni az ágenst.
Alternatív forgatókönyv	**1. B** Ha a játékos le van bénulva, nem tudja ezt az akciót végrehajtani.

Use-case neve	Gyűjtés
Rövid leírás	Gyűjthető dolgok eltárolása.
Aktorok	Felhasználó
Főforgatókönyv	**1.** A játékos begyűjti a gyűjthető dolgokat.
Alternatív forgatókönyv	**1. A 1.** A játékos letapogatja a laborban a genetikai kódot.
Alternatív forgatókönyv	**1. A 2.** Ha a játékos már ismeri az adott kódot, akkor nem lesz újra eltárolva nála.
Alternatív forgatókönyv	**1. B 1.** A játékos összegyűjti az adott anyagot.
Alternatív forgatókönyv	**1. B 2.** Ha a játékos nem rendelkezik már megfelelő mennyiségű tárolókapacitással, akkor az anyagot nem tudja begyűjteni.
Alternatív forgatókönyv	**1. C 1.** A játékos begyűjti az adott eszközöket és automatikusan aktiválódik is rajta. Minden eszköztípusból csak egyet tárolhat el.

Alternatív forgatókönyv	**1. C 2.** Ha a játékosnál megvan már a három eszköz, akkor nem vehet fel többet.
Alternatív forgatókönyv	**1. D** Ha a játékos egy lebénult játékossal találkozik, akkor elveheti tőle az eszközöket, anyagokat.
Alternatív forgatókönyv	**1. E** Ha a játékos le van bénulva, nem tud gyűjtögetni semmit se.

Use-case neve	Város megtekintése
Rövid leírás	A játékos látja a várost.
Aktorok	Felhasználó
Főforgatókönyv	**1.** A játékos megnézheti, hogy melyik és milyen mezőn áll éppen, milyen gyűjthető dolgok vannak nála és milyen effektek és eszközök hatnak rá.

Use-case neve	Ágens vezérlése
Rövid leírás	Általános visszaszámláló, amely körönként egy egységet csökkent az adott értékéből.
Aktorok	Rendszer
Főforgatókönyv	**1.** Az ágens létrehozásakor körönként elkezdi csökkenteni a számlálója értékét. **2.** Ha az ágens aktiválódik egy játékoson, elindul a számlálója. 3. Ha a játékosra egy ágenst kennek, de már rendelkezik kesztyűvel, akkor a kenést követően elindul egy számláló, amely idő alatt el kell döntenie, hogy elfogadja a kenést vagy sem.
Alternatív forgatókönyv	**1. A** Ha ez az érték lejár az adott ágens megsemmisül.
Alternatív forgatókönyv	**2. A** Ha ez az érték lejár az adott effekt lekerül a játékosról.
Alternatív forgatókönyv	**3. A** Ha ez az érték lejár, és a játékos nem döntött, akkor a kapott ágens automatikusan aktiválódik a játékoson.

Use-case neve	Város létrehozása
Rövid leírás	A játék indításáért és inicalizásáért felelős use-case.
Aktorok	Rendszer
Főforgatókönyv	**1.** Létrehozza a várost és minden benne lévő elemet.

2.4.2 Use-case diagram

2.5 Szótár

Akció: cselekvés, effekt (ld. Effekt) rátétele a másik játékos (ld. Játékos) karakterére (ld. Karakter)

Aktiválódik: azt jelenti, hogy egy adott eszköz (ld. Eszköz) vagy ágens (ld. Ágens) speciális képessége (pl. immunitás ld. Immunitás)

Anyag: egy olyan fizikai dolog, amelyet maga a karakter (ld. Karakter) tud gyűjteni a játék (ld. Játék) során, ebből készülnek el az ágensek (ld. Ágens), szinonima: gyűjthető dolog

Ágens: a játék (ld. Játék) által definiált létrehozható vírus vagy vakcina, melyet a karakterek fel tudnak használni

Áldozat: egy olyan játékos (ld. Játékos) karaktere (ld. Karakter), aki ágens (ld. Ágens) hatása alá került

Billentyűzet: maga a játékos (ld. Játékos) egyik fizikai beviteli lehetősége

Csapat: játék (ld. Játék) készítői, fejlesztői

Effekt: ez egy hatás, amely akkor következik be, mikor valaki aktivál (ld. Aktiválódik) egy ágenst (ld. Ágens), többféle típusa is van, szinonima: képesség

Egér: maga a játékos (ld. Játékos) egyik fizikai beviteli lehetősége

Elkészít: ha a karakter (ld. Karakter) rendelkezik megfelelő mennyiségű anyaggal (ld. Anyag) és a megfelelő kóddal (ld. Kód) akkor ezek kombinálásával létrejön az ágens (ld. Ágens)

Eszköz: egy olyan fizikai dolog, amelyet maga a karakter (ld. Karakter) tud gyűjteni a játék (ld. Játék) során, ebből készülnek el az ágensenk (ld. Ágens), szinonima: felszerelés, gyűjthető dolog

Eszköztár: az eszközöket (ld. Eszköz) és anyagokat (ld. Anyag) itt tudja a játékos (ld. Játékos) karaktere (ld. Karakter) tárolni, amíg fel nem használj azokat, szinonima: raktár

Érintés: egyik játékos (ld. Játékos) karaktere (ld. Karakter) kapcsolatba kerül a másik játékos karakterével és akciót (ld. Akció) hajt rajta végre

Felhasználó: az ember, aki használja a programot

Felszerelés: ld. Eszköz

Gyűjtés: ez egy olyan cselekvés, amellyel a játékos (ld. Játékos) a saját kis tárolójába helyezi a pályán (ld. Pálya) található eszközt (ld. Eszköz) vagy anyagot (ld. Anyag), szinonima: eltevés, raktározás, letapogatás (kód esetében)

Hatás: ld. Effekt

Hatóidő: ameddig az ágens (ld. Ágens) effektje (ld. Effekt) hat egy karakterre (ld. Karakter)

Időkorlát: egy általun megadott fix időintervallum, mely, ha eléri a nullát valami változás következik be

Immunitás: játékosra (ld. Játékos) fix ideig nem hat más játékos ágense (ld. Ágens)

Játék: ld. Program

Játékos: a virológusokat irányító ember, maga a program felhasználója (ld. Felhasználó) lesz

Karakter: a játékos (ld. Játékos) által irányított gépi lény, szinonima: virológus

Kenés: amikor az egyik játékos (ld. Játékos) a másik játékosra egy effektet (ld. Effekt) helyez rá.

Kód: a játékban (ld. Játék) találhatóak olyan úgynevezett genetikai kódok, melyeket a karakternek (ld. Karakter) van lehetősége gyűjteni (ld. Gyűjtés), szinonima: genetikai kód

Kör: egy körnek számít az, mikor minden játékos (ld. Játékos) elvégzi a lehetséges lépéseik számát

Lebénul: a karakter (ld. Karakter) megadott ideig nem képes cselekvésre, ebbe beleszámít a mozgás, ágensek (ld. Ágens) használata, készítése

Lép: a karakter (ld. Karakter) egy másik mezőre kerül, szinonima: mozog, mozgás

Lopás: ugyanaz a cselekvés, mint egy gyűjtésnél (ld. Gyűjtés), csak itt a játékos (ld. Játékos) nem a pályán (ld. Pálya) található mezőről (ld. Mező) szedi fel az adott gyűjthető dolgot**,** hanem egy lebénult karaktert (ld. Karakter) összegyűjtött dolgaiból tud válogatni, veheti el tőle azokat, szinonima: kifosztás, elvevés, eltulajdonítás

Megsemmisül: az adott ágens (ld. Ágens), eszközt (ld. Eszköz) többé nem lehet használni

Mező: a pályát (ld. Pálya) alkotó sokszögletű elemek, szinonima: blokk, pályaelem

Pálya: a játék (ld. Játék) helyszíne, szinonima: város

Program: a csapat által elkészített termék

Város: ld. pálya

2.6 Projekt terv

2.6.1 Ütemterv

Határidő	Feladat	Pont	Felelős
febr. 28.	Követelmény, projekt, funkcionalitás - beadás	10	Csia
márc. 7.	Analízis modell kidolgozása I. - beadás	20	
márc. 16.	Analízis modell kidolgozása II. - beadás	30	
márc. 21.	Szkeleton tervezése - beadás	20	
márc. 28.	Szkeleton elkészítése - beadás	20	
ápr. 4.	Prototípus koncepciója – beadás Szkeleton bemutatás	20	
ápr. 11.	Részletes tervek - beadás	45	
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	

2.6.2 Erőforrások

Dokumentálásra használt eszközök: Microsoft Word

Kommunikáció: Személyesen, Discord, Messenger

Modellező eszköz: WhiteStarUML

Fejlesztőeszköz: Java (Visual Studio Code)

Dokumentumok megosztása: Google Drive, Discord

Forráskód megosztása, verziókezelése: Git (Github)

2.7 Napló

Kezdet	Időtartam	Résztvevő(k)	Leírás
2022. 02. 18. 18:18	20 perc	Alpek, Csia	Tevékenység: Csapattagokkal kapcsolatfelvétel.
2022. 02. 20. 01:54	20 perc	Alpek, Csia, Litavecz	Tevékenység: Közös platformokban megegyezés.
2022. 02. 22. 12:45	1 óra	Alpek, Csia, Litavecz	Tevékenység: Közös dokumentum sablon elkészítése.
2022. 02. 23. 12:15	1,5 óra	Alpek, Csia, Marton	Tevékenység: Konzultáció konzulenssel, feladat emésztése, átbeszélése, dokumentáció bővítése. Döntés: Következő meeting időpont megbeszélése.
2022. 02. 25. 17:00	6 óra	Teljes csapat	Tevékenység: Teljes feladat közös átolvasása. Döntés: Következő nap a be nem fejezett részek külön-külön kidolgozása csapattagok között szétosztva. Amiben nem tudtunk megegyezni Marton mindannyiunk nevében levelet fogalmazott meg a konzulensünknek.
2022. 02. 26 17:00	10 perc	Teljes csapat	Megegyezés: Közös leadási határidő a külön kiosztott feladatok részére.
2022. 02. 27 10:00	2 óra	Teljes csapat	Tevékenység: A teljes első házi feladat átnézése közösen, apróbb hibák/elírások keresése még nyomtatás előtt. Döntés: Csia elvállalta, hogy hétfőn ő adja majd le a kinyomtatott papírokat, és tölti fel Herculesre a dokumentumot.
2022. 02. 28. 09:00	30 perc	Teljes csapat	Tevékenység: Utolsó hibák közös keresése, teljes dokumentum átnézése. Döntés: Mindenki által el lett fogadva a dokumentum, így leadásra kerül.

2.8 Százalékos teljesítés

Név	Százalék
Alpek	25%
Csia	25%
Litavecz	25%
Marton	25%

3. Analízis modell kidolgozása I.

3.1 Objektum katalógus

3.1.1 Játék

A **játék menedzseléséért** felelős objektum. Ez inicializálja a dolgokat a játék kezdetén, helyezi el a játékosokat a kiindulómezőre és tölti fel a raktárakat, óvóhelyeket, illetve helyezi el a genetikai kódokat a laborokba. A virológus ennek az objektumnak üzen, ha megszerezte a kellő mennyiségű genetikai kódot, így ennek az objektumnak a felelőssége lezárni a játékot, és erről értesíteni a játékosokat is.

3.1.2 Időzítő

Az **időzítések kezelé**sért felelős. Megadott időnként lépteti a *Steppable* interfészű objektumokat. Csak egy darab létezik belőle.

3.1.3 Mező

A **pálya egy egységnyi része**. A virológusok ezeken mozognak, továbbá ezeken találhatóak meg a gyűjthető elemek, mint a genetikai kódok, eszközök és anyagok. Egy mezőnek több másik szomszédja is lehet, akár különböző típusúak is.

3.1.4 Laboratórium

A mezővel megegyező tulajdonságú játékelem. A virológus itt tudja megtanulni a **genetikai kódok**at, mely később az ágens generálásához, illetve a játék megnyeréséhez szükséges.

3.1.5 Óvóhely

A mezővel megegyező tulajdonságú játékelem. A virológus itt tudja összegyűjteni és felvenni a **felszerelések**et, mint a kesztyű, a köpeny, és a zsák.

3.1.6 Raktár

A mezővel megegyező tulajdonságú játékelem. A virológus itt tudja összegyűjteni az **ágens generáláshoz szükséges anyagok**at, mint a nukleotid vagy az aminosav.

3.1.7 Virológus

Tárolja és rendelkezésre bocsátja a **tároló**t. Felveszi, illetve leveszi magáról az **eszköz**öket, a levett eszközöket belerakja az tárolóba, amennyiben van elegendő hely benne. Ha nincs elegendő hely, akkor eldobja (megsemmisíti) őket. Megtámad egy másik **virológus**t (**ágenst** ken rá), illetve védekezik az őt ért támadás során. A védekezésnél használhatja az eszközeit (max 3 eszköz, melyeket önmagán hord). Letapogatja a labor falán található **kód**ot. Ágenseket generál. Szomszédos **mezőre lép**. Eszközöket és anyagokat **lop** egy másik, lebénult virológustól.

3.1.8 Leltár

Ez egy olyan eszköz, amely a játék kezdete óta a karakternél van. Ide tudja **begyűjt**eni a játékos a **genetikai kódokat**, az **anyagokat** és az **eszközöket** is. Minden játékosnál csak egy darab ilyen található, nem ellopható.

3.1.8 Aminosav

Ez egy olyan anyagtípus, amely szükséges az **ágensek előállításához**. A játékos a raktárakban vagy esetleg más játékosoktól jut hozzá. Más különleges képessége nincs.

3.1.9 Nukleotid

Ez egy olyan anyagtípus, amely szükséges az **ágensek előállításához**. A játékos a raktárakban vagy esetleg más játékosoktól jut hozzá. Más különleges képessége nincs.

3.1.10 Vitustánc

Az előállítható **ágens**ek egyik fajtája, melyek a virológus mind magára, mind másik virológusra is elhasználhat. Hatása, hogy a **játékos elveszíti az irányítást a karakter felett**, aki véletlenszerű mozgást kezd el végezni, azaz elkezdi ezt a „táncot”, ameddig a hatás le nem jár a karakterről.

3.1.11 Immunitás

Az előállítható **ágens**ek egyik fajtája, melyek a virológus mind magára, mind másik virológusra is elhasználhat. Hatása, hogy a játékos **védettséget élvez** más ágensekkel szemben”, ameddig a hatás le nem jár.

3.1.12 Bénulás

Az előállítható **ágens**ek egyik fajtája, melyek a virológus mind magára, mind másik virológusra is elhasználhat. Hatása, hogy a játékos **mozgás- és cselekvőképtelenné válik**, ameddig a hatás le nem jár.

3.1.13 Felejtés

Az előállítható **ágens**ek egyik fajtája, melyek a virológus mind magára, mind másik virológusra is elhasználhat. Hatása, hogy a játékos **elfelejti az eddig megtanult** összes **genetikai kódot**. A hatás végleges és nem visszafordítható.

3.1.14 Kesztyű

Ez egy olyan **felszerelés** objektum, amelyet a virológus az óvóhelyen tud összegyűjteni és felvenni magára, de akár más játékostól is meg tudja szerezni. Képessége, hogy **vissza** tudja **dob**ni a rákent ágenst.

3.1.15 Köpeny

Ez egy olyan **felszerelés** objektum, amelyet a virológus az óvóhelyen tud összegyűjteni és felvenni magára, de akár más játékostól is meg tudja szerezni. Képessége, hogy 82,3%-os hatással **tartja távol** az ágenseket.

3.1.16 Zsák

Ez egy olyan **felszerelés** objektum, amelyet a virológus az óvóhelyen tud összegyűjteni és felvenni magára, de akár más játékostól is meg tudja szerezni. Képessége, hogy **meg** tudja **növelni a leltár anyaggyűjtő kapacitását**.

3.2 Statikus struktúra diagramok

A jobb áttekinthetőség érdekében pár metódust (pl. Getter-Setter) nem jelenítettünk meg rajta.

3.3 Osztályok leírása

3.3.1 Immunity

- **Felelősség**
  - Speciális típusú ágens, amely megvédi a virológust attól, hogy mások által használt ágensek hatással legyenek rá.
- **Ősosztályok**
  - Agent
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -
- **Metódusok**
  - **void step()**: Ha nem járt le az időzítő, akkor beállítja igazra azon virológus immunitás attribútumát, amelyikre hat az ágens és csökkenti az időzítőt. Ha lejárt, akkor a virológus immunitás attribútumát hamisra állítja és törli a virológusra éppen ható ágenst attribútumot.

3.3.2 Paralysis

- **Felelősség**
  - Olyan ágens, amely megbénít, így amíg az ágens hatása tart, az áldozat nem tud semmit csinálni.
- **Ősosztályok**
  - Agent
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -
- **Metódusok**
  - **void step()**: Ha nem járt le az időzítő, akkor beállítja igazra azon virológus bénultság attribútumát, amelyikre hat az ágens és csökkenti az időzítőt. Ha lejárt, akkor a virológus bénultság attribútumát hamisra állítja és törli a virológusra éppen ható ágenst attribútumot.

3.3.3 Oblivion

- **Felelősség**
  - Ezen ágens hatására az áldozat elfelejti a már megismert genetikai kódokat.
- **Ősosztályok**
  - Agent
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -

- **Metódusok**
  - **void step()**: Ha nem járt a le az időzítő, akkor elkéri azon virológus immunitás leltárját, amelyikre hat az ágens és a felejtés metódus segítségével törli az összes megismert genetikai kódot. Majd az időzítőt nullára állítja, illetve törli a virológusra éppen ható ágenst attribútumot.

3.3.4 VitusDance

- **Felelősség**
  - Ennek hatására a virológus véletlenszerűen kezd el a haladni a pályán.
- **Ősosztályok**
  - Agent
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -
- **Metódusok**
  - **void step()**: Ha nem járt le az időzítő, akkor beállítja igazra azon virológus táncolás attribútumát, amelyikre hat az ágens, csökkenti az időzítőt és egy véletlenszerűen választott szomszédos mezőre lépteti. Ha lejárt, akkor a virológus táncolás attribútumát hamisra állítja és törli a virológusra éppen ható ágenst attribútumot.

3.3.5 Agent

- **Felelősség**
  - Ennek hatására a virológus véletlenszerűen kezd el a haladni a pályán.
- **Ősosztályok**
  - -
- **Interfészek**
  - Steppable
- **Attribútumok**
  - **RemaningTime:** A fennmaradt szavatossági időt vagy a fennmaradt hatás idejét tárolja.
  - **Experiation:** A szavatossági idő, ami azt az időt jelenti ameddig az elkészített ágenst van lehetőség felhasználni.
  - **EffectTime:** A hatás ideje, ami azt az időt jelenti ameddig egy ágens hat egy virológusra.
- **Metódusok**
  - **void step()**: Absztrakt léptető metódus.

3.3.6 Steppable

- **Felelősség**
  - Egy interfész, ami minden olyan dolgot reprezentál, amely időben lépni tud.
- **Ősosztályok**
  - -
- **Interfészek**
  - -

- **Attribútumok**
  -
- **Metódusok**
  - **void step()**: Az adott lépésben végrehajtandó művelet.

3.3.7 Timer

- **Felelősség**
  - Periodikus időzítőt reprezentál a játékban, a léptethető (Steppable) dolgokat lépteti.
- **Ősosztályok**
  - -
- **Interfészek**
  - Steppable
- **Attribútumok**
  - **Steppable:** A léptethető dolgok.
- **Metódusok**
  - **void tick()**: Minden léptethető dolog léptetése.
  - **void AddSteppable(s:Steppable)**: Új léptethető dolog hozzáadása.
  - **void RemoveSteppable(s:Steppable**): Léptethető dolog törlése.

3.3.8 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**
  - -
- **Interfészek**
  - Steppable
- **Attribútumok**
  - **Immunity**: Megadja, hogy a virológus immunis-e a rákent ágenssekkel szemben.
  - **Paralyzed**: Megadja, hogy a virológus le van-e bénulva.
  - **Dancing:** Megadja, hogy a virológus épp vitustánc hatása alatt áll-e.
  - **WearedGloves:** A virológus által viselt kesztyűk listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik két eszközfajta listája mennyi elemből áll).
  - **WearedSack:** A virológus által viselt zsákok listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik két eszközfajta listája mennyi elemből áll).
  - **WearedCloak:** A virológus által viselt köpenyek listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik két eszközfajta listája mennyi elemből áll).
  - **StandingField:** Az a mező, amelyen a virológus áll.

- **Metódusok**
  - **void setImmunity(b: boolean)**: Beállítja az *Immuniy* attribútum értékét a b értékére.
  - **void setVirologistUnderEffect(a: Agent):** Beállítja a virológusra éppen ható ágens attribútum értékét a bementi paraméterére.
  - **void setDancing(b: boolean)**: Beállítja a *Dancing* attribútum értékét a bemeneti paramétere.
  - **Field getStandingField()**: Visszaadja azt a mezőt, amin a virológus éppen van.
  - **boolean defense(vFr: Virologist, a: Agent)**: Ha a virológuson lévő kesztyű listája nem üres, akkor meghívja az első kesztyű use metódusát. Amennyiben ennek visszatérési értéke *false* (vagyis a játékos nem használta a kesztyűt), akkor megnézzük a virológuson lévő köpenyek listáját. Ha a lista üres, akkor *false*-szal visszatérünk. Egyébként kivesz egy köpenyt a listából, és meghívja a köpeny *evade()* metódusát. Ha az *evade()* metódus *false*-szal tér vissza, akkor kiveszi a következő köpenyt, és megismétli rajta az előző műveletet, amíg a lista ki nem ürül. Amennyiben egy köpeny *evade()* metódusa *true*-val tér vissza, a folyamat megszakad. Végül visszaadja a legutoljára meghívott evade metódus értékét. Amennyiben a kesztyű *use()* metódusa *true*-val tér vissza (a játékos használta a kesztyűt), akkor ezzel az értékkel egyből visszatérünk a köpenyek listájának vizsgálata nélkül.
  - **void attack(vTo: Virologist, a: Agent)**: Megvizsgáljuk, hogy a virológus le van-e bénulva. Ha igen, nem történik semmi, egyébként meghívjuk a *vTo* paraméterben átadott virológus *assaulted()* metódusát, melynek visszatérési értéke jelzi, hogy sikeres volt-e a támadás. Amennyiben a támadás sikerrel járt, megvizsgáljuk, hogy a megtámadott virológus le van-e bénulva. Ha igen, és a támadó virológus tárhelye nincs tele, meghívjuk a virológus *stealBy()* metódusát.
  - **boolean assaulted(Virologist vFr, Agent a):** Megnézi, hogy a virológuson van-e bármilyen hatás. Amennyiben nincs, akkor meghívja a *defense()* metódust. Ha a *defense()* metódus *false*-szal tér vissza, inicializáljuk az ágens *remainingTime* attribútumát az *effectTime* attribútumának értékével, beállítjuk az ágens *virologistUnderEffect* attribútumát a megtámadott virológusra, majd elindítjuk az ágens működési folyamatát.
  - **void stealBy(v: Virologist)**: A felhasználó által kiválasztott "ellenséges " virológus, amennyiben az le van bénulva, meglophatja. A függvény megmutatja a másik virológus *inventory*jában, illetve magán a virológuson lévő eszközöket. A felhasználó által kiválasztott anyagokat, eszközöket a saját inventoryjába pakolja.
  - **void takeOffSack()**: Leveszi a virológuson lévő zsákot.
  - **void takeOffCloak()**: Leveszi a virológuson lévő köpenyt/köpenyeket.
  - **void takeOffGloves()**: Leveszi a virológuson léve kesztyűt/kesztyűket.
  - **Agent createAgent()**: Létrehoz egy ágenst, elindítja a létrejött ágens időzítőjét, azt elhelyezi az ágensek listájába.
  - **void equipGloves(g: Gloves)**: Hozzáad egy kesztyűt a virológuson lévő kesztyűk listájához, a kesztyűt kitörli az *inventory*-ból.
  - **void equipSacks(g: Gloves)**: Hozzáad egy zsákot a virológuson lévő zsákok listájához, a zsákot kitörli az *inventory*-ból, valamint megnöveli a maximálisan gyűjthető anyagok(nukleotid, aminosav) mennyiségét.
  - **void equipCloak(g: Gloves)**: Hozzáad egy köpenyt a virológuson lévő köpenyek listájához, a köpenyt kitörli az *inventory*-ból.

3.3.9 Inventory

- **Felelősség**
  - Ezen osztály felelőssége a virológus által összegyűjtött eszközök, anyagok tárolása, felvétele.
- **Ősosztályok**
  - -
- **Interfészek**
  - -
- **Attribútumok**
  - **MaxMaterial**: Maximálisan eltárolható anyagok száma.
  - **MaxEquipment:** Maximálisan eltárolható eszközök száma.
  - **StoredGloves:** Eltárolt kesztyűk listája.
  - **StoredSack:** Eltárolt zsákokk listája.
  - **StoredCloak:** Eltárolt köpenyek listája.
  - **LearntGC:** Már megismert kódok listája.
  - **Aminos:** Megszerzett aminosavak listája.
  - **Nukleotids:** Megszerzett nukleotidok listája.
- **Metódusok**
  - **Inventory getInventory ()**: Visszaadja a virológus által tárolt Intentory objektumot.
  - **void forgetGCodes()**: Törli a virológus GCode-kat tároló lista minden elemét.
  - **void addEqToInv(sh: Shelter)**: Az óvóhelyről elveszi a eszközt, amit a virológus fel akar venni és törli azt az óvóhelyről.
  - **void removeGloves()**: Eltávolít egy kesztyűt az inventoryból.
  - **void removeSack()**: Eltávolít egy zsákot az inventoryból.
  - **void removeCloak()**: Eltávolít egy köpenyt az inventoryból.
  - **boolean learnGCode(l: Lab)**: Leellenőrzi, hogy imeri-e a kódot, amit meg akar tanulni a virológus, ha nem, akkor hozzáadja a már megismert kódokhoz az újat. Ha már ismeri ezt a kódot, akkor nem adja hozzá. Visszatérése a leolvasás megtörténtét jelzi.
  - **void addAmino(st: Storage)**: Hozzáad egy amonisavat az inventoryhoz.
  - **void addNukleo(st: Storage)**: Hozzáad egy nukleotidot az inventoryhoz.
  - **void removeAmino()**: Eltávolít egy aminosavat az inventoryból.
  - **void removeNukleo()**: Eltávolít egy nukleotidot az inventoryból.

3.3.10 Field

- **Felelősség**
  - Üres mezőt reprezentáló osztály.
- **Ősosztályok**
  - -
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -

- **Metódusok**
  - **void arrive(v: Virologist)**: A virológust hozzáadja az új mezőhöz, amelyre lépett, és beállítja a *field* változóját az új mezőre, amely végigiterál a mezőn aktuálisan álló virológusokon.
  - **boolean wannaSteal()**: Amennyiben virológus végigiteráláskor bénult virológust talál a mezőre lépéskor, akkor meghívja ezt a metódust, ha ez igaz visszatérési értéket kap, akkor meghívja a *stealBy()* metódust is a virológusnál
  - **void addVirologist(v: Virologist)**: Hozzáadja a virológust a mező listájához.
  - **void removeVirologist(v: Virologist)**: Eltávolítja a virológust a mező listájából.
  - **Field getARandomNeighbour()**: A mező szomszédok listájából véletlenszerűen visszaad egy elemet. A szomszédok száma legalább 4 kell, hogy legyen, maximum meg az általunk megadott természetes szám, amely jelen játékunk esetében 8.

3.3.11 Lab

- **Felelősség**
  - A laboratóriumot reprezentáló osztály. Itt találhatóak a genetikai kódok, melyekből ágens készíthető.
- **Ősosztályok**
  - Field
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -
- **Metódusok**
  - **void arrive(v: Virologist)**: A virológust hozzáadja az új mezőhöz, amelyre lépett, és beállítja a *field* változóját az új mezőre, amely végigiterál a mezőn aktuálisan álló virológusokon bénult virológus után kutatva. Ez a metódus hívja meg a *wannaTouchy*-t is.
  - **boolean wannaTouchy()**: Ez a metódus kérdezi meg a felhasználót, hogy akar-e kódot letapogatni. Amennyiben igaz visszatérési értéket kap, úgy meghívódik a virológusnál a *touchy()* metódus.

3.3.12 Storage

- **Felelősség**
  - A raktára reprezentáló osztály. Itt találhatóak az anyagok, melyekből a játékos később ágenst készíthet.
- **Ősosztályok**
  - Field
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -

- **Metódusok**
  - **void arrive(v: Virologist)**: A virológust hozzáadja az új mezőhöz, amelyre lépett, és beállítja a *field* változóját az új mezőre, amely végigiterál a mezőn aktuálisan álló virológusokon bénult virológus után kutatva. Ez a metódus hívja meg a *wannaPickUp*-ot is.
  - **void removeMat()**: A *wannaPickUp()* metódust követően ez a metódus hívódik meg, amely eltávolítja a felvett anyagot a mezőről (mező listájából).
  - **void refill()**: Ez a metódus abban az esetben hívódik meg, ha a mező anyag listája üres lett. Ebben az esetben random mennyiségű és típusú anyaggal tölti fel a mező listáját.
  - **boolean wannaPickUp()**: Ez a metódus kérdezi meg a felhasználót, hogy akar-e anyagot begyűjteni. Amennyiben igaz visszatérési értéket kap, úgy meghívódik az adott anyagnál a *pickUp()* metódus.

3.3.13 Shelter

- **Felelősség**
  - Az óvóhelyet reprezentáló osztály. Itt találhatóak a felszerelések/eszközök, melyeket később a játékos magára vehet.
- **Ősosztályok**
  - Field
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -
- **Metódusok**
  - **void arrive(v: Virologist)**: A virológust hozzáadja az új mezőhöz, amelyre lépett, és beállítja a *field* változóját az új mezőre, amely végigiterál a mezőn aktuálisan álló virológusokon bénult virológus után kutatva. Ez a metódus hívja meg a *wannaPickUp*-ot is.
  - **void removeEq(eq: Equipment):** A *wannaPickUp()* metódust követően ez a metódus hívódik meg, amely eltávolítja a felvett eszközt a mezőről (mező listájából).
  - **void refill(eq: Equipment):** Ez a metódus abban az esetben hívódik meg, ha a mező eszköz listája üres lett. Ebben az esetben random mennyiségű és típusú eszközzel tölti fel a mező listáját.
  - **boolean wannaPickUp():** Ez a metódus kérdezi meg a felhasználót, hogy akar-e eszközt begyűjteni. Amennyiben igaz visszatérési értéket kap, úgy meghívódik az adott eszköznél a *pickUp()* metódus.

3.3.14 Equipment

- **Felelősség**
  - A felszerelések/eszközök ősosztálya.
- **Ősosztályok**
  -
- **Interfészek**
  - -
- **Attribútumok**
  - -

- **Metódusok**
  - **void pickUp(v: Virologist)**: Absztrakt felvételi metódus.

3.3.15 Gloves

- **Felelősség**
  - Ez az osztály felelős a kesztyű, mint eszköz megszerzéséért. és a kesztyűk használatának a lebonyolításáért.
- **Ősosztályok**
  - Equipment
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Amikor a virológus egy óvóhelyre lép és fel akar venni egy kesztyűt onnan, akkor azt a virológus *inventory*jába teszi.

3.3.16 Cloak

- **Felelősség**
  - Ez az osztály felelős a köpeny, mint eszköz megszerzéséért és a köpenyek használatának a lebonyolításáért.
- **Ősosztályok**
  - Equipment
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Amikor a virológus egy óvóhelyre lép és fel akar venni egy köpenyt onnan, akkor azt a virológus *inventory*jába teszi.

3.3.17 Sack

- **Felelősség**
  - Ez az osztály felelős a zsák, mint eszköz megszerzéséért.
- **Ősosztályok**
  - Equipment
- **Interfészek**
  - -
- **Attribútumok**
  - **Capacity: int**: A zsák kapacitása, amivel növeli a maximum megszerezhető anyagok (nukleotid, aminosav) számát.
- **Metódusok**
  - **void pickUp(v: Virologist)**: Amikor a virológus egy óvóhelyre lép és fel akar venni egy zsákot onnan, akkor azt a virológus *inventory*jába teszi.

3.3.18 Material

- **Felelősség**
  - A raktárban található anyagok Ősosztálya.
- **Ősosztályok**
  -
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Absztrakt felvevő metódus.

3.3.19 AminoAcid

- **Felelősség**
  - Az aminosavak megszerzését irányítja.
- **Ősosztályok**
  - Material
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Indítja az aminosavak felvételét az *inventory*ba.

3.3.20 Nukleotid

- **Felelősség**
  - A nukleotidok megszerzését irányítja.
- **Ősosztályok**
  - Material
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Indítja a nukleotidok felvételét az *inventory*ba.

3.3.20 GCode

- **Felelősség**
  - A genetikai kódok felépítését tartalmazza, illetve az ágensek létrehozását irányítja.
- **Ősosztályok**
  - -
- **Interfészek**
  - -

- **Attribútumok**
  - **GCName**: A kód alapján előállítható ágens neve.
  - **UsedAmino:** Az ágens előállításához szükséges aminosavak listája.
  - **UsedNukleo:** Az ágens előállításához szükséges nukleotidok listája.
- **Metódusok**
  - **Agent create(inv: Inventory)**: Ellenőrzi, hogy az ágens létrehozásához megvan-e a megfelelő mennyiségű anyag az *inventory*ban, ha nincs, akkor visszatér: NULL. Ha megvan, akkor azokat kiveszi a virológus *inventory*jából és létrehozza a megfelelő ágenst.

3.4 Szekvencia diagramok

3.4.1 Timer steps

3.4.2 Immunity steps

3.4.3 Oblivion steps

3.4.4 VitusDance steps

3.4.5 Paralysis steps

3.4.6 Virologist moves to a basic field (*from a basic field*)

Átláthatóság érdekében feltételezzük, hogy már ki lett választva a szomszédok közül az adott típusú mező.

3.4.7 Virologist moves to a laboratory field (*from a basic field*)

Átláthatóság érdekében feltételezzük, hogy már ki lett választva a szomszédok közül az adott típusú mező.

3.4.8 Virologist moves to a shelter field (*from a basic field*)

Átláthatóság érdekében feltételezzük, hogy már ki lett választva a szomszédok közül az adott típusú mező.

3.4.9 Virologist moves to a storage field (*from a basic field*)

Átláthatóság érdekében feltételezzük, hogy már ki lett választva a szomszédok közül az adott típusú mező.

3.4.10 Refill the Storage

A kevesebb diagram átnézése, és a jobb áttekinthetőség érdekében nem jelenítettük meg két különböző diagrammon a két különböző típusú anyag elvételét, és leltárba rakását.

3.4.11 Refill the Shelter

A kevesebb diagram átnézése, és a jobb áttekinthetőség érdekében nem jelenítettük meg három különböző diagrammon a három különböző típusú eszköz elvételét, és leltárba rakását.

3.4.12 Attack

sd attack
f : Felhasználó
attacker : Virologist
attacked : Virologist
a : Agent
1 : attack(attacked, a)
opt
[attacker.Paralyzed == false]
2 : assaulted(attacker, a)
3 : hasEffectedAgent()
4 : bUnderEffect
opt
[bUnderEffect == false]
opt
[attacker != attacked]
5 : defense(attacker, a)
6 : bDefenseSuccessful
7 : bDefenseSuccessful
opt
[bDefenseSuccessful == false]
opt
[attacked.Paralyzed == true]
8 : stealBy(attacker)
9
10

3.4.13 Defense

3.4.14 Create Agent

3.4.15 Stealing

18
loop
[stolenNukleoNum > 0]
19 : removeNukleo()
20
21 : addAmino(st)
22
loop
[stolenGlovesNum > 0]
23 : removeGloves()
24
25 : addGlvoes()
26
loop
[stolenSackNum > 0]
27 : removeSack()
28
29 : addSack()
30
loop
[stolenCloakNum > 0]
31 : removeCloak()
32
33 : addCloak()
34
35

3.4.16 Gloves pick up

3.4.17 Cloak pick up

3.4.18 Sack pick up

3.4.19 Amino pick up

3.4.20 Nukleo pick up

3.4.21 Genetic code read

3.5 Állapotgépek

3.5.1 Ágens effekt visszaszámláló

3.6 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	
márc. 21.	Szkeleton tervezése - beadás	20	
márc. 28.	Szkeleton elkészítése - beadás	20	
ápr. 4.	Prototípus koncepciója – beadás Szkeleton bemutatás	20	
ápr. 11.	Részletes tervek - beadás	45	
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	

3.7 Napló

Kezdet	Időtartam	Résztvevő(k)	Leírás
2022.02.28 17:30	6 óra	Teljes csapat	Értekezlet: Osztályok, objektumok nagyvonalú meghatározása
2022.03.02 12:15	2 óra	Alpek, Csia	Konzultáción való részvétel.
2022.03.02 14:00	2 óra	Teljes csapat	Értekezlet: Osztályok és objektumok újból átbeszélése, ötletelés a konzultáción hallottak alapján.
2022.03.05 13:00	2,5 óra	Teljes csapat	Értekezlet: Az eddig tervezettek teljes elhajítása, osztályok és objektumok újból átbeszélése, megtervezése.
2022.03.05 20:30	5 óra	Teljes csapat	Értekezlet: A még aznapi értekezlet folytatása. Osztálydiagram véglegesítése.
2022.03.06 02:30	20 perc	Marton	Feladatok szétosztása az aszinkron munkához.
2022.03.06 14:00	8 óra	Litavecz	Az ágensek szekvenciadiagramjainak megtervezése, objektumleírások megadása, hozzátartozó osztályok leírása.
2022.03.06 14:00	10 óra	Marton	A virológus osztály szekvenciaidiagrammjainak elkészítése, objektumleírás megadása, hozzátartozó osztályok leírása
2022.03.06 14:00	8 óra	Csia	A mozgás és mezők szekvenciadiagramjainak megtervezése, objektumleírások megadása, hozzátartozó osztályok leírása.
2022.03.06 14:00	10 óra	Alpek	Az eszközhasználatok, gyűjtések, lopás megtervezése, objektumleírások megadása, hozzátartozó osztályok leírása.
2022.03.07 09:00	1 óra	Marton	Segítségnyújtás Csiának a dokumentum véglegesítésében.
2022.03.07 07:00	4 óra	Csia	Dokumentum egységesítése, napló bővítése, összegyűlt ötletek egybetevése.

2022.03.07 10:30	1 óra	Teljes csapat	Dokumentum teljes átnézése, apróbb hibák keresése.

3.8 Százalékos teljesítés

Név	**Százalék**
Alpek	25%
Csia	25%
Litavecz	25%
Marton	25%

3. Analízis modell kidolgozása I.

3.1 Objektum katalógus

3.1.1 Játék

A **játék menedzseléséért** felelős objektum. Ez inicializálja a dolgokat a játék kezdetén, helyezi el a játékosokat a kiindulómezőre, illetve helyezi el a genetikai kódokat a laborokba. A virológus ennek az objektumnak üzen, ha megszerezte a kellő mennyiségű genetikai kódot, így ennek az objektumnak a felelőssége lezárni a játékot, és erről értesíteni a játékosokat is.

3.1.2 Időzítő

Az **időzítések kezelé**sért felelős. Megadott időnként lépteti a *Steppable* interfészű objektumokat. Csak egy darab létezik belőle.

3.1.3 Mező

A **pálya egy egységnyi része**. A virológusok ezeken mozognak, továbbá ezeken találhatóak meg a gyűjthető elemek, mint a genetikai kódok, eszközök és anyagok. Egy mezőnek több másik szomszédja is lehet, akár különböző típusúak is.

3.1.4 Laboratórium

A mezővel megegyező tulajdonságú játékelem. A virológus itt tudja megtanulni a **genetikai kódok**at, mely később az ágens generálásához, illetve a játék megnyeréséhez szükséges.

3.1.5 Óvóhely

A mezővel megegyező tulajdonságú játékelem. A virológus itt tudja összegyűjteni és felvenni a **felszerelések**et, mint a kesztyű, a köpeny, és a zsák.

3.1.6 Raktár

A mezővel megegyező tulajdonságú játékelem. A virológus itt tudja összegyűjteni az **ágens generáláshoz szükséges anyagok**at, mint a nukleotid vagy az aminosav.

3.1.7 Virológus

Tárolja és rendelkezésre bocsátja a **leltár**at. Felveszi, illetve leveszi magáról az **eszköz**öket, a levett eszközöket belerakja az leltárba, amennyiben van elegendő hely benne. Ha nincs elegendő hely, akkor eldobja (megsemmisíti) őket. Megtámad egy másik **virológus**t (**ágenst** ken rá), illetve védekezik az őt ért támadás során. A védekezésnél használhatja az eszközeit (max 3 eszköz, melyeket önmagán hord). Letapogatja a labor falán található **kód**ot. Ágenseket generál. Szomszédos **mezőre lép**. Eszközöket és anyagokat **lop** egy másik, lebénult virológustól.

3.1.8 Leltár

Ez egy olyan eszköz, amely a játék kezdete óta a karakternél van. Ide tudja **begyűjt**eni a játékos a **genetikai kódokat**, az **anyagokat** és az **eszközöket** is. Minden játékosnál csak egy darab ilyen található, nem ellopható.

3.1.8 Aminosav

Ez egy olyan anyagtípus, amely szükséges az **ágensek előállításához**. A játékos a raktárakban vagy esetleg más játékosoktól jut hozzá. Más különleges képessége nincs.

3.1.9 Nukleotid

Ez egy olyan anyagtípus, amely szükséges az **ágensek előállításához**. A játékos a raktárakban vagy esetleg más játékosoktól jut hozzá. Más különleges képessége nincs.

3.1.10 Vitustánc

Az előállítható **ágens**ek egyik fajtája, melyek a virológus mind magára, mind másik virológusra is elhasználhat. Hatása, hogy a **játékos elveszíti az irányítást a karakter felett**, aki véletlenszerű mozgást kezd el végezni, azaz elkezdi ezt a „táncot”, ameddig a hatás le nem jár a karakterről.

3.1.11 Immunitás

Az előállítható **ágens**ek egyik fajtája, melyek a virológus mind magára, mind másik virológusra is elhasználhat. Hatása, hogy a játékos **védettséget élvez** más ágensekkel szemben”, ameddig a hatás le nem jár.

3.1.12 Bénulás

Az előállítható **ágens**ek egyik fajtája, melyek a virológus mind magára, mind másik virológusra is elhasználhat. Hatása, hogy a játékos **mozgás- és cselekvőképtelenné válik**, ameddig a hatás le nem jár.

3.1.13 Felejtés

Az előállítható **ágens**ek egyik fajtája, melyek a virológus mind magára, mind másik virológusra is elhasználhat. Hatása, hogy a játékos **elfelejti az eddig megtanult** összes **genetikai kódot**. A hatás végleges és nem visszafordítható.

3.1.14 Kesztyű

Ez egy olyan **felszerelés** objektum, amelyet a virológus az óvóhelyen tud összegyűjteni és felvenni magára, de akár más játékostól is meg tudja szerezni. Képessége, hogy **vissza** tudja **dob**ni a rákent ágenst.

3.1.15 Köpeny

Ez egy olyan **felszerelés** objektum, amelyet a virológus az óvóhelyen tud összegyűjteni és felvenni magára, de akár más játékostól is meg tudja szerezni. Képessége, hogy 82,3%-os hatással **tartja távol** az ágenseket.

3.1.16 Zsák

Ez egy olyan **felszerelés** objektum, amelyet a virológus az óvóhelyen tud összegyűjteni és felvenni magára, de akár más játékostól is meg tudja szerezni. Képessége, hogy **meg** tudja **növelni a leltár anyaggyűjtő kapacitását**.

3.2 Statikus struktúra diagramok

A jobb áttekinthetőség érdekében pár metódust (pl. Getter-Setter metódusokat) nem jelenítettünk meg a diagramon.

3.3 Osztályok leírása

3.3.1 Immunity

- **Felelősség**
  - Speciális típusú ágens, amely megvédi a virológust attól, hogy mások által használt ágensek hatással legyenek rá. Hatása 4 körön keresztül tart.
- **Ősosztályok**
  - Agent
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -
- **Metódusok**
  - **void step()**: Ha nem járt le az időzítő, akkor beállítja igazra azon virológus immunitás attribútumát, amelyikre hat az ágens és csökkenti az időzítőt. Ha lejárt, akkor a virológus immunitás attribútumát hamisra állítja és törli a virológusra éppen ható ágenst attribútumot.

3.3.2 Paralysis

- **Felelősség**
  - Olyan ágens, amely megbénít, így amíg az ágens hatása tart, az áldozat nem tud semmit csinálni. Hatása 5 körön keresztül tart.
- **Ősosztályok**
  - Agent
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -
- **Metódusok**
  - **void step()**: Ha nem járt le az időzítő, akkor beállítja igazra azon virológus bénultság attribútumát, amelyikre hat az ágens és csökkenti az időzítőt. Ha lejárt, akkor a virológus bénultság attribútumát hamisra állítja és törli a virológusra éppen ható ágenst attribútumot.

3.3.3 Oblivion

- **Felelősség**
  - Ezen ágens hatására az áldozat elfelejti a már megismert genetikai kódokat. Hatása 1 körön keresztül tart.
- **Ősosztályok**
  - Agent
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -

- **Metódusok**
  - **void step()**: Ha nem járt a le az időzítő, akkor elkéri azon virológus immunitás leltárját, amelyikre hat az ágens és a felejtés metódus segítségével törli az összes megismert genetikai kódot. Majd az időzítőt nullára állítja, illetve törli a virológusra éppen ható ágenst attribútumot.

3.3.4 VitusDance

- **Felelősség**
  - Ennek hatására a virológus véletlenszerűen kezd el a haladni a pályán. Hatása 6 körön keresztül tart.
- **Ősosztályok**
  - Agent
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -
- **Metódusok**
  - **void step()**: Ha nem járt le az időzítő, akkor beállítja igazra azon virológus táncolás attribútumát, amelyikre hat az ágens, csökkenti az időzítőt és egy véletlenszerűen választott szomszédos mezőre lépteti. Ha lejárt, akkor a virológus táncolás attribútumát hamisra állítja és törli a virológusra éppen ható ágenst attribútumot.

3.3.5 Agent

- **Felelősség**
  - Absztrakt osztály, mely keretet biztosít a különböző ágens típusok együttes kezelésére.
- **Ősosztályok**
  - -
- **Interfészek**
  - Steppable
- **Attribútumok**
  - **RemaningTime:** A fennmaradt szavatossági időt vagy a fennmaradt hatás idejét tárolja.
  - **Expiration:** A szavatossági idő, ami azt az időt jelenti ameddig az elkészített ágenst van lehetőség felhasználni.
  - **EffectTime:** A hatás ideje, ami azt az időt jelenti ameddig egy ágens hat egy virológusra.
  - **VirologistUnderEffect:** Az ágens hatása alatt álló virológust tárolja.
- **Metódusok**
  - **void step()**: Absztrakt léptető metódus.
  - **void setRemainingTime(time: int):** beállítja a remainingTime attribútumot a paraméterben kapott értékre.

3.3.6 Steppable

- **Felelősség**
  - Egy interfész, ami minden olyan dolgot reprezentál, amely időben lépni tud.
- **Ősosztályok**

  - -
- **Interfészek**
  - -
- **Attribútumok**
  -
- **Metódusok**
  - **void step()**: Az adott lépésben végrehajtandó művelet.

3.3.7 Timer

- **Felelősség**
  - Periodikus időzítőt reprezentál a játékban, a léptethető (Steppable) dolgokat lépteti.
- **Ősosztályok**
  - -
- **Interfészek**
  - Steppable
- **Attribútumok**
  - **Steppable:** A léptethető dolgok.
- **Metódusok**
  - **void tick()**: Minden léptethető dolog léptetése.
  - **void AddSteppable(s:Steppable)**: Új léptethető dolog hozzáadása.
  - **void RemoveSteppable(s:Steppable**): Léptethető dolog törlése.

3.3.8 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**
  - -
- **Interfészek**
  - Steppable
- **Attribútumok**
  - **Immunity**: Megadja, hogy a virológus immunis-e a rákent ágenssekkel szemben.
  - **Paralyzed**: Megadja, hogy a virológus le van-e bénulva.
  - **Dancing:** Megadja, hogy a virológus épp vitustánc hatása alatt áll-e.
  - **HasEffectedAgent:** Megadja, hogy a virológus jelen pillanatban ágens hatása alatt áll-e.
  - **WornGloves:** A virológus által viselt kesztyűk listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik két eszközfajta listája mennyi elemből áll).
  - **WornSack:** A virológus által viselt zsákok listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik két eszközfajta listája mennyi elemből áll).
  - **WornCloak:** A virológus által viselt köpenyek listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik két eszközfajta listája mennyi elemből áll).

- **StandingField:** Az a mező, amelyen a virológus áll.
- **Inventory:** A virológus leltára.

- **Metódusok**
  - **void setImmunity(b: boolean)**: Beállítja az *Immuniy* attribútum értékét a b értékére.
  - **void setVirologistUnderEffect(a: Agent):** Beállítja a virológusra éppen ható ágens attribútum értékét a bementi paraméterére.
  - **void setDancing(b: boolean)**: Beállítja a *Dancing* attribútum értékét a bemeneti paramétere.
  - **Field getStandingField()**: Visszaadja azt a mezőt, amin a virológus éppen van.
  - **void equipGloves(g: Gloves)**: Hozzáad egy kesztyűt a virológuson lévő kesztyűk listájához, a kesztyűt kitörli az *inventory*-ból.
  - **void equipSacks(g: Gloves)**: Hozzáad egy zsákot a virológuson lévő zsákok listájához, a zsákot kitörli az *inventory*-ból, valamint megnöveli a maximálisan gyűjthető anyagok(nukleotid, aminosav) mennyiségét 5-tel.
  - **void equipCloak(g: Gloves)**: Hozzáad egy köpenyt a virológuson lévő köpenyek listájához, a köpenyt kitörli az *inventory*-ból.
  - **boolean defense(vFr: Virologist, a: Agent)**: Ha a virológuson lévő kesztyű listája nem üres, akkor meghívja az első kesztyű use metódusát. Amennyiben ennek visszatérési értéke *false* (vagyis a játékos nem használta a kesztyűt), akkor megnézzük a virológuson lévő köpenyek listáját. Ha a lista üres, akkor *false*-szal visszatérünk. Egyébként kivesz egy köpenyt a listából, és meghívja a köpeny *evade()* metódusát. Ha az *evade()* metódus *false*-szal tér vissza, akkor kiveszi a következő köpenyt, és megismétli rajta az előző műveletet, amíg a lista ki nem ürül. Amennyiben egy köpeny *evade()* metódusa *true*-val tér vissza, a folyamat megszakad. Végül visszaadja a legutoljára meghívott evade metódus értékét. Amennyiben a kesztyű *use()* metódusa *true*-val tér vissza (a játékos használta a kesztyűt), akkor törlünk egy kesztyűt a viselt eszközök közül, és a köpenyek listájának vizsgálata nélkül true-val visszatérünk.
  - **void takeOffGloves()**: Ha a virológus nincs lebénulva, leveszi a rajta lévő kesztyűt, ha van elég hely, akkor berakja a leltárba, egyébként eldobja.
  - **void takeOffSack()**: Ha a virológus nincs lebénulva, leveszi a rajta lévő zsákot, ha van elég hely, akkor berakja a leltárba, egyébként eldobja.
  - **void takeOffCloak()**: Ha a virológus nincs lebénulva, leveszi a rajta lévő köpenyt, ha van elég hely, akkor berakja a leltárba, egyébként eldobja.
  - **Agent createAgent(g : GCode)**: Létrehoz egy ágenst (a paraméterben kapott GCode objektum create metódusának meghívásával), amennyiben ez sikeres volt (a create metódus visszatérési értéke nem null), akkor hozzáadja az ágenst a steppable objektumok közé (Timer addSteppable metódusát hívva), és ezzel elindítja az ágens időzítőjét. Elhelyezi a létrehozott ágenst az ágensek listájába.
  - **void attack(vTo: Virologist, a: Agent)**: Megvizsgáljuk, hogy a virológus le van-e bénulva. Ha igen, nem történik semmi, egyébként meghívjuk a *vTo* paraméterben átadott virológus *assaulted()* metódusát, melynek visszatérési értéke jelzi, hogy sikeres volt-e a támadás. Amennyiben a támadás sikerrel járt, megvizsgáljuk, hogy a megtámadott virológus le van-e bénulva. Ha igen, és a támadó virológus tárhelye nincs tele, meghívjuk a virológus *stealBy()* metódusát.
  - **boolean assaulted(Virologist vFr, Agent a):** Megnézi, hogy a virológuson van-e bármilyen hatás. Ha van, akkor nem történik semmi, mivel a virológusra egy időben csak egyetlen ágens hathat. Amennyiben nincs, és a virológus nem önmagára indítja a támadást (kenést), akkor meghívja a *defense()* metódust. Ha a *defense()* metódus *false*-szal tér vissza, inicializáljuk az ágens *remainingTime* attribútumát az *effectTime*

attribútumának értékével, beállítjuk az ágens *virologistUnderEffect* attribútumát a megtámadott virológusra, majd elindítjuk az ágens működési folyamatát.
- **void move(f: Field)**: A virológus átmozdul egy másik mezőre.
- **void stealBy(v: Virologist)**: A felhasználó meglophat egy másik virológust, amennyiben az le van bénulva. A függvény elkéri a paraméterben kapott virológus inventory-ját, majd meghívja a *chooseObjectToSteal()* metódust. Ezután a visszatérési értéknek megfelelően eltávolít egy eszközt vagy material-t a saját inventoryjából, és ha ez sikeres volt (volt az inventoryjában olyan eszköz / material), akkor hozzáadja azt a paraméterként kapott virológus inventoryjához. A virológus által viselt eszköz nem ellopható, csak az, ami az inventory-ban található.
- **void addAgent(a: Agent)**: Hozzáad egy ágenst az ágensek listájába.
- **void step()**: A virológus által már létrehozott, de fel nem használt ágensek lejárati idejét csökkenti, amennyiben a lejárati idő 0, akkor törli az adott ágenst a listából.
- **void touchy(l: Lab):** meghívja a virológus inventory-jának learnGCode() metódusát, amennyiben ennek visszatérése true (tanult új kódot), akkor megvizsgálja, hogy az összes kódot ismeri-e már. Ha az összes kódot megtanulta, akkor meghívja a Game objektum endGame(v:Virologist) metódusát.
- **String chooseObjectToSteal()**: A játékos kiválasztja, hogy mit szeretne ellopni, az ellopandó eszköz vagy material nevének kiválasztásával. A visszatérési érték a kiválasztott név.

3.3.9 Inventory

- **Felelősség**
  - Ezen osztály felelőssége a virológus által összegyűjtött eszközök, anyagok tárolása, felvétele.
- **Ősosztályok**
  - -
- **Interfészek**
  - -
- **Attribútumok**
  - **MaxMaterial**: Maximálisan eltárolható anyagok száma, 20 db
  - **MaxEquipment:** Maximálisan eltárolható eszközök száma, 5db
  - **StoredGloves:** Eltárolt kesztyűk listája.
  - **StoredSack:** Eltárolt zsákok listája.
  - **StoredCloak:** Eltárolt köpenyek listája.
  - **LearntGC:** Már megismert kódok listája.
  - **Aminos:** Megszerzett aminosavak listája.
  - **Nukleotids:** Megszerzett nukleotidok listája.
- **Metódusok**
  - **Inventory getInventory ()**: Visszaadja a virológus által tárolt Intentory objektumot.
  - **void addGloves(g: Gloves)**: Hozzáad egy kesztyűt az inventory kesztyű listájához, amennyiben van elég hely az inventory-ban. Ellenkező esetben eldobja azt. (Objektum megsemmisítése)
  - **void addSack(s: Sack)**: Hozzáad egy zsákot az inventory zsák listájához, amennyiben van elég hely az inventory-ban. Ellenkező esetben eldobja azt. (Objektum megsemmisítése)
  - **void addCloak(c: Cloak)**: Hozzáad egy köpenyt az inventory köpeny listájához, amennyiben van elég hely az inventory-ban. Ellenkező esetben eldobja azt. (Objektum megsemmisítése)

- **Gloves removeGloves()**: Eltávolít egy kesztyűt az inventoryból, és visszatér az eltávolított objektummal. (Ha a lista üres, akkor változatlan marad, és NULL értékkel tér vissza)
- **void removeSack()**: Eltávolít egy zsákot az inventoryból, és visszatér az eltávolított objektummal. (Ha a lista üres, akkor változatlan marad, és NULL értékkel tér vissza)
- **void removeCloak()**: Eltávolít egy köpenyt az inventoryból, és visszatér az eltávolított objektummal. (Ha a lista üres, akkor változatlan marad, és NULL értékkel tér vissza)
- **boolean learnGCode(l: Lab)**: Leellenőrzi, hogy imeri-e a kódot, amit meg akar tanulni a virológus, ha nem, akkor hozzáadja a már megismert kódokhoz az újat. Ha már ismeri ezt a kódot, akkor nem adja hozzá. Visszatérése jelzi, hogy hozzáadódott-e új elem a listához.
- **void forgetGCodes()**: Törli a virológus GCode-kat tároló listájának minden elemét.
- **void addAmino(am: Aminoacid)**: Hozzáad egy amonisavat az inventoryhoz, amennyiben van elég hely az inventory-ban. Ellenkező esetben eldobja azt. (Objektum megsemmisítése)
- **void addNukleo(n: Nukleotid)**: Hozzáad egy nukleotidot az inventoryhoz, amennyiben van elég hely az inventory-ban. Ellenkező esetben eldobja azt. (Objektum megsemmisítése)
- **void removeAmino()**: Eltávolít egy aminosavat az inventoryból, és visszatér az eltávolított objektummal. (Ha a lista üres, akkor változatlan marad, és NULL értékkel tér vissza)
- **void removeNukleo()**: Eltávolít egy nukleotidot az inventoryból, és visszatér az eltávolított objektummal. (Ha a lista üres, akkor változatlan marad, és NULL értékkel tér vissza)

3.3.10 Field

- **Felelősség**
  - Üres mezőt reprezentáló osztály.
- **Ősosztályok**
  - -
- **Interfészek**
  - Steppable
- **Attribútumok**
  - Neighbours: A mező szomszédai (Egy mezőnek 4-8 szomszédja lehet).
  - Virologist: A mezőn álló virológusok listája.
- **Metódusok**
  - **Field getARandomNeighbour()**: A mező szomszédok listájából véletlenszerűen visszaad egy elemet. A szomszédok száma legalább 4 kell, hogy legyen, maximum meg az általunk megadott természetes szám, amely jelen játékunk esetében 8.
  - **void arrive(v: Virologist)**: A virológust törli annak a field-nek a listájából, amelyen előzőleg állt (virológus field változójából kiolvasva), majd hozzáadja a saját listájához (ez az új mező, ahová a virológus lépett). Ez után beállítja a virologist *field* változóját az új mezőre (vagyis önmagára). Végül végigiterál a mezőn aktuálisan álló virológusokon, és ha bénultat talál, meghívja a wannaSteal() metódust. Amennyiben ez true-val tér vissza, meghívja a bénult virológus stealBy() metódusát.
  - **boolean wannaSteal()**: Megkérdezi a játékost, hogy szeretne-e lopni az azonos mezőn álló bénult virológustól.
  - **void addVirologist(v: Virologist)**: Hozzáadja a virológust a mező listájához.
  - **void removeVirologist(v: Virologist)**: Eltávolítja a virológust a mező listájából.

3.3.11 Lab

- **Felelősség**
  - A laboratóriumot reprezentáló osztály. Itt találhatóak a genetikai kódok, melyekből ágens készíthető.
- **Ősosztályok**
  - Field
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -
- **Metódusok**
  - **void arrive(v: Virologist)**: Meghívja az ősosztály arrive(v: Virologist) metórusát, majd a *wannaTouchy*-t is. Ha a *wannaTouchy* metódus true-val tér vissza, akkor meghívja a virológus *touchy()* metódusát is.
  - **boolean wannaTouchy()**: Ez a metódus kérdezi meg a felhasználót, hogy akar-e kódot letapogatni.

3.3.12 Storage

- **Felelősség**
  - A raktára reprezentáló osztály. Itt találhatóak az anyagok, melyekből a játékos később ágenst készíthet.
- **Ősosztályok**
  - Field
- **Interfészek**
  - Steppable
- **Attribútumok**
  - -
- **Metódusok**
  - **void arrive(v: Virologist)**: Meghívja az ősosztály arrive(v: Virologist) metórusát, majd a *wannaPickUp*-ot is. Ha a wannaPickup true-val tér vissza, akkor meghívja a
  - **void addMat(v: Virologist)**: A *wannaPickUp()* metódust követően ez a metódus hívódik meg, amely egy random számot kisorsol (annyi szám közül, ahány anyag található az aktuális játékban, jelen esetben ez 2), ehhez létrehoz egy anyagot, és annak hívja meg a megfelelő *PickUp()* metódusát.
  - **boolean wannaPickUp()**: Ez a metódus kérdezi meg a felhasználót, hogy akar-e anyagot begyűjteni. Amennyiben igaz visszatérési értéket kap, úgy meghívódik az adott anyagnál a *pickUp()* metódus.

-

3.3.13 Shelter

- **Felelősség**
  - Az óvóhelyet reprezentáló osztály. Itt találhatóak a felszerelések/eszközök, melyeket később a játékos magára vehet.
- **Ősosztályok**
  - Field
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void arrive(v: Virologist)**: Meghívja az ősosztály *arrive(v: Virologist)* metórusát, majd a *wannaPickUp*-ot is. Amennyiben ez true-val tér vissza, meghívja az *addEq* metódust.
  - **void addEq(eq: Equipment):** Kisorsol egy random számot (annyi szám közül, ahány eszköz található az aktuális játékban, jelen esetben ez 4), ehhez létrehoz egy eszközt, és annak hívja meg a megfelelő *PickUp()* metódusát.
  - **boolean wannaPickUp():** Ez a metódus kérdezi meg a felhasználót, hogy akar-e eszközt begyűjteni. Amennyiben igaz visszatérési értéket kap, úgy meghívódik az adott eszköznél a *pickUp()* metódus.

3.3.14 Equipment

- **Felelősség**
  - A felszerelések/eszközök ősosztálya.
- **Ősosztályok**
  -
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Absztrakt felvételi metódus.

3.3.15 Gloves

- **Felelősség**
  - Ez az osztály felelős a kesztyű, mint eszköz megszerzéséért. és a kesztyűk használatának a lebonyolításáért.
- **Ősosztályok**
  - Equipment
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Amikor a virológus egy óvóhelyre lép és fel akar venni egy kesztyűt onnan, akkor azt a virológus *inventory*jába teszi.
  - **use(vFr:Virologist, vTo: Virologist, a:Agent):** Először meghívja a *wannaUse()* metódust. Amennyiben ez false-al tér vissza, ugyanazzal az értékkel visszatérünk, nem történik semmi. Ellenkező esetben meghívjuk a vTo objektum attack metódusát, a vFr és a objektumot paraméterként átadva. Amikor használtuk a kesztyűt, true étrékkel térünk vissza.
  - **boolean wannaUse():** Megkérdezi a felhasználót, hogy akarja-e használni a kesztyűt, és a felhasználó válaszának megfelelő értékkel visszatér.

3.3.16 Cloak

- **Felelősség**
  - Ez az osztály felelős a köpeny, mint eszköz megszerzéséért és a köpenyek használatának a lebonyolításáért.
- **Ősosztályok**
  - Equipment
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Amikor a virológus egy óvóhelyre lép és fel akar venni egy köpenyt onnan, akkor azt a virológus *inventory*jába teszi.
  - **Void evade()**: Egy olyan algoritmus, mely 82,3%-os valószínűséggel true, 17.7%-os valószítűséggel false értéket ad vissza.

3.3.17 Sack

- **Felelősség**
  - Ez az osztály felelős a zsák, mint eszköz megszerzéséért.
- **Ősosztályok**
  - Equipment
- **Interfészek**
  - -
- **Attribútumok**
  - **Capacity: int**: A zsák kapacitása, amivel növeli a maximum megszerezhető anyagok (nukleotid, aminosav) számát (5 db).
- **Metódusok**
  - **void pickUp(v: Virologist)**: Amikor a virológus egy óvóhelyre lép és fel akar venni egy zsákot onnan, akkor azt a virológus *inventory*jába teszi.

3.3.18 Material

- **Felelősség**
  - A raktárban található anyagok Ősosztálya.
- **Ősosztályok**
  -
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Absztrakt felvevő metódus.

3.3.19 AminoAcid

- **Felelősség**
  - Az aminosavak megszerzését irányítja.
- **Ősosztályok**
  - Material
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Indítja az aminosavak felvételét az *inventory*ba.

3.3.20 Nukleotid

- **Felelősség**
  - A nukleotidok megszerzését irányítja.
- **Ősosztályok**
  - Material
- **Interfészek**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **void pickUp(v: Virologist)**: Indítja a nukleotidok felvételét az *inventory*ba.

3.3.20 GCode

- **Felelősség**
  - A genetikai kódok felépítését tartalmazza, illetve az ágensek létrehozását irányítja.
- **Ősosztályok**
  - -
- **Interfészek**
  - -
- **Attribútumok**
  - **GCName**: A kód alapján előállítható ágens neve.
  - **UsedAmino:** Az ágens előállításához szükséges aminosavak listája.
  - **UsedNukleo:** Az ágens előállításához szükséges nukleotidok listája.
- **Metódusok**
  - **Agent create(inv: Inventory)**: Ellenőrzi, hogy az ágens létrehozásához megvan-e a megfelelő mennyiségű anyag az *inventory*ban, ha nincs, akkor visszatér: NULL. Ha megvan, akkor azokat kiveszi a virológus *inventory*jából és létrehozza a megfelelő ágenst.

3.3.20 Game

- **Felelősség**
  - A játék kezdetén a mezők kialakításáért, virológusok elhelyezéséért felel, majd a játék végén értesít minden virológust arról, hogy megnyerte vagy elvesztette a játékot.
- **Ősosztályok**
  - -
- **Interfészek**
  - -
- **Attribútumok**
  - **MaxGCode**: itt tároljuk azt, hogy hányféle kód létezik, melyet a játékosoknak a győzelem érdekében mind el kell sajátítaniuk.
  - **Fields**: a játékban létező mezők listája.
- **Metódusok**
  - **startGame():** A rendszer felépíti a mezőket, létrehozza a virológusokat, azokat elhelyezi kezdeti mezőjükön.
  - **endGame(v: Virologist):** A rendszer végigiterál az összes mezőn, a mezőktől elkéri a Virologists listáját, majd az összes virológusnak megjeleníti a játék végéről szóló értesítést, illetve, hogy az adott virológus nyert-e vagy nem. Ezután minden létező objektumot töröl.

3.4 Szekvencia diagramok

3.4.1 Timer steps

3.4.2 Immunity steps

3.4.3 Oblivion steps

3.4.4 Paralysis steps

3.4.5 VitusDance steps

3.4.6 Virologist steps

3.4.7 Virologist moves to a basic field

3.4.8 Virologist moves to a laboratory field

3.4.9 Virologist moves to a shelter field

3.4.10 Virologist moves to a storage field

3.4.11 Attack

sd attack
: Felhasználó
attacker : Virologist
attacked : Virologist
a : Agent
1 : attack(attacked, a)
opt
[attacker.Paralyzed == false]
2 : assaulted(attacker, a)
3 : hasEffectedAgent()
4 : bUnderEffect
opt
[bUnderEffect == false]
opt
[attacker != attacked]
5 : defense(attacker, a)
6 : bDefenseSuccessful
7 : bDefenseSuccessful
opt
[bDefenseSuccessful == false]
8 : getParalyzed()
9 : bParalyzed
opt
[bParalyzed == true]
10 : wannaSteal()
11 : bSteal
opt
[bSteal == true]
12 : stealBy(attacker)
13
14

3.4.12 Defense

3.4.13 Steal

sd steal
stealerVirologist : Virologist
paralizedVirologist : Virologist
iParalized : Inventory
1 : stealBy(stealerVirologist)
2 : getInventory()
3 : iStealer
4 : chooseObjectToSteal()
5 : objectName
iStealer : Inventory
alt
[objectName == "Aminoacid"]
6 : removeAmino()
amino : Aminoacid
7 : amino
opt
[amino != NULL]
8 : addAmino(amino)
9
[objectName == "Nukleotid"]
10 : removeNukleotid()
nukleo : Nukleotid
11 : nukleo
opt
[nukleo != NULL]
12 : addNukleo(nukleo)
13
[objectName == "Gloves"]
14 : removeGloves()
gloves : Gloves
15 : gloves
opt
[gloves != NULL]
16 : addGloves(gloves)
17
[objectName == "Sack"]
18 : removeSack()
sack : Sack
19 : sack
opt
[sack != NULL]
20 : addSack(sack)
21
[objectName == "Cloak"]
22 : removeCloack()
cloak : Cloak
23 : cloak
opt
[cloak != NULL]
24 : addCloak(cloak)
25
26

3.4.14 Create Agent

3.4.15 Equips gloves

3.4.16 Takes off gloves

3.4.17 Equips cloak

3.4.18 Takes off cloak

3.4.19 Equips sack

3.4.20 Takes off sack

3.4.21 Genetic code read

3.5 Állapotgépek

3.5.1 Ágens effekt visszaszámláló

3.6 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	
márc. 28.	Szkeleton elkészítése - beadás	20	
ápr. 4.	Prototípus koncepciója – beadás Szkeleton bemutatás	20	
ápr. 11.	Részletes tervek - beadás	45	
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	

3.7 Napló

Kezdet	Időtartam	Résztvevő(k)	Leírás
2022.03.12 20:30	6 óra	Teljes csapat	Értekezlet: A konzultáción megbeszélt hibák átnézése. Közös ötletelés, hogy hogyan javítsuk ki ezeket.
2022.03.12 10:38	20 perc	Marton	Feladat: A következő konzultációnkig a feladatok szétosztása az emberek között, hogy kinek mi a dolga a saját határidőnkig.
2022.03.13 12:02	3 óra	Alpek	Feladat: Ágensgenerálás, illetve eszközök fel-le vételéhez szekvenciadiagram készítése, függvényleírások ennek megfelelően javítása.
2022.03.13 14:00	3 óra	Csia	Feladat: Az óvóhely és a labor mezők metódusaihoz szekvenciadiagram készítése, függvényleírások ennek megfelelően javítása. Stepper szekvenciadiagram javítása.
2022.03.13 15:34	3 óra	Litavecz	Feladat: Az üres mező és a raktár mezők metódusaihoz, illetve a virológus step() metódusához (ágens effekt léptetéséhez) szekvenciadiagram készítése, függvényleírások ennek megfelelően javítása.
2022.03.13 11:27	3 óra	Marton	Feladat: Kesztyűhasználathoz szekvenciadiagram készítése, függvényleírások javítása, defense és attack szekvenciadiagram javítása.
2022.03.14 15:00	4 óra	Teljes csapat	Értekezlet: A mindenki által beadott szekvenciadiagrammok, és a hozzájuk tartozó függvények leírásának átnézése hibák keresése végett.
2022.03.14 20:00	1 óra	Csia	Feladat: Teljes dokumentum összefésülése, képek megigazítása, közös dokumentumba helyezése, helyesírási hibák keresése, dokumentum véglegesítése.

3.8 Százalékos teljesítés

Név	Százalék
Alpek	25%
Csia	25%
Litavecz	25%
Marton	25%

5. Szkeleton tervezése

5.1 A szkeleton modell valóságos use-case-ei (I.)

5.1.1 Use-case diagram (I.)

5.1.2 Use-case leírások (I.)

Use-case neve	Equip Sack
Rövid leírás	A játékos felvesz magára egy adott eszközt.
Aktorok	User
Forgatókönyv	**1.** Felvesz egy zsákot, amely megnöveli az anyagtároló kapacitását az *Inventory*-jának.
Alternatív forgatókönyv	**1. A.** Ha a viselt eszközök száma három, akkor nem tudja ezt az akciót végrehajtani.
Alternatív forgatókönyv	**1. B.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani.

Use-case neve	Equip Cloak
Rövid leírás	A játékos felvesz magára egy adott eszközt.
Aktorok	User
Forgatókönyv	**1.** Felvesz egy köpenyt, amely megakadályozhatja, hogy egy másik virológus ágenst kenjen rá.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani.
Alternatív forgatókönyv	**1. B.** Ha a viselt eszközök száma három, akkor nem tudja ezt az akciót végrehajtani.

Use-case neve	Equip Gloves
Rövid leírás	A játékos felvesz magára egy adott eszközt.
Aktorok	User
Forgatókönyv	**1.** Felvesz egy kesztyűt, amely visszadobja az éppen rákent ágenst.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani.
Alternatív forgatókönyv	**1. B.** Ha a viselt eszközök száma három, akkor nem tudja ezt az akciót végrehajtani.

Use-case neve	Create Agent
Rövid leírás	A játékos elkészít egy ágenst.
Aktorok	User
Forgatókönyv	**1.** Elkészít egy adott ágenst.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani.
Alternatív forgatókönyv	**1. B.** Ha nincs meg minden megfelelő alapanyag az ágens elkészítéséhez (genetika kód, anyagok), akkor nem tudja ezt az akciót végrehajtani.

Use-case neve	Attacks other Virologist
Rövid leírás	A játékos megtámad egy másik virológust.
Aktorok	User
Forgatókönyv	**1.** A játékos egy általa elkészített ágenst ráken egy másik játékosra.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani.
Alternatív forgatókönyv	**1. B.** Ha a megtámadott fél már ágens hatása alatt van, akkor nem tudja ezt az akciót végrehajtani.
Alternatív forgatókönyv	**1. C.** Ha bénító ágens sikeresen célba ért, az ellenfél lophatóvá válik.

Use-case neve	Defense
Rövid leírás	A játékos megpróbál kivédeni egy támadást.
Aktorok	User
Forgatókönyv	**1.** A játékos egy, az ellenfél által készített ágenst megpróbál kivédeni az általa viselt eszközök vagy ágens segítéségével.

Use-case neve	Steals from paralyzed Virologist
Rövid leírás	A játékos meglop egy másik játékost.
Aktorok	User
Forgatókönyv	**1.** A játékos egy általa választott eszközt vagy anyagot lop ki a másik már megbénult játékos *Inventory*-jából.
Alternatív forgatókönyv	**1. A.** Ha a lopó fél *Inventory*-jában már nincs elegendő hely a lopott tárgyhoz, akkor az a tárgy megsemmisül.

Use-case neve	Moves to Storage
Rövid leírás	A játékos egy másik mezőre lép.
Aktorok	User
Forgatókönyv	**1.** A játékos egy általa kiválasztott Raktár típusú mezőre lép, ahol be tud szerezni különböző típusú anyagokat.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani. (Vitustáncnál is rá tud lépni Raktár típusú mezőre, de az nem általa van kiválasztva, hanem randomzáltan léphet rá.)
Alternatív forgatókönyv	**1. B.** Ha a mezőn lebénult játékossal találkozik, az meglophatja.
Alternatív forgatókönyv	**1. C.** Ha a játékos *Inventory*-jában már nincs elegendő hely az adott anyaghoz, akkor nem tudja ezt az akciót végrehajtani.

Use-case neve	Moves to Field
Rövid leírás	A játékos egy másik mezőre lép.
Aktorok	User
Forgatókönyv	**1.** A játékos egy általa kiválasztott üres típusú mezőre lép.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani. (Vitustáncnál is rá tud lépni üres típusú mezőre, de az nem általa van kiválasztva, hanem randomzáltan léphet rá.)
Alternatív forgatókönyv	**1. B.** Ha a mezőn lebénult játékossal találkozik, az meglophatja.

Use-case neve	Moves to Shelter
Rövid leírás	A játékos egy másik mezőre lép.
Aktorok	User
Forgatókönyv	**1.** A játékos egy általa kiválasztott Óvóhely típusú mezőre lép, ahol be tud szerezni különböző típusú eszközöket.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani. (Vitustáncnál is rá tud lépni Óvóhely típusú mezőre, de az nem általa van kiválasztva, hanem randomzáltan léphet rá.)
Alternatív forgatókönyv	**1. B.** Ha a mezőn lebénult játékossal találkozik, az meglophatja.
Alternatív forgatókönyv	**1. C.** Ha a játékos *Inventory*-jában már nincs elegendő hely az adott eszközhöz, akkor nem tudja ezt az akciót végrehajtani.

Use-case neve	Moves to Lab
Rövid leírás	A játékos egy másik mezőre lép.
Aktorok	User
Forgatókönyv	**1.** A játékos egy általa kiválasztott Labor típusú mezőre lép, ahol genetikai kódot tud megtanulni.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani. (Vitustáncnál is rá tud lépni Óvóhely típusú mezőre, de az nem általa van kiválasztva, hanem randomzáltan léphet rá.)
Alternatív forgatókönyv	**1. B.** Ha a mezőn lebénult játékossal találkozik, az meglophatja.
Alternatív forgatókönyv	**1. C.** Ha a játékos már megtanult egy adott kódot, többször nem tudja már megtanulni. (Kivéve, ha Felejtés ágens áldozatává vált, abban az esetben újra tanulhatja a kódokat.)

Use-case neve	Takes off Sack
Rövid leírás	A játékos levesz magáról egy adott eszközt.
Aktorok	User
Forgatókönyv	**1.** A játékos levesz magáról egy zsák típusú eszközt, amely levétel esetén az *Inventory*-jába kerül vissza, amely csökkenti a *MaxMaterial* attribútum értékét a zsák kapacitás értékével.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani.
Alternatív forgatókönyv	**1. B.** Ha levételkor a játékosnak nincs megfelelő kapacitása az *Inventory*-jában, akkor az adott eszköz megsemmisül levételt követően.
Alternatív forgatókönyv	**1. C.** Ha levételt követően több anyag van a játékosnál, mint amennyi lehet, akkor automatikusan törlésre kerül a többlet anyag.

Use-case neve	Takes off Cloak
Rövid leírás	A játékos levesz magáról egy adott eszközt.
Aktorok	User
Forgatókönyv	**1.** A játékos levesz magáról egy köpeny típusú eszközt, amely levétel esetén az *Inventory*-jába kerül vissza.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani.
Alternatív forgatókönyv	**1. B.** Ha levételkor a játékosnak nincs megfelelő kapacitása az *Inventory*-jában, akkor az adott eszköz megsemmisül levételt követően.

Use-case neve	Takes off Gloves
Rövid leírás	A játékos levesz magáról egy adott eszközt.
Aktorok	User
Forgatókönyv	**1.** A játékos levesz magáról egy kesztyű típusú eszközt, amely levétel esetén az *Inventory*-jába kerül vissza.
Alternatív forgatókönyv	**1. A.** Ha le van bénulva vagy vitustáncot jár, akkor nem tudja ezt az akciót végrehajtani.
Alternatív forgatókönyv	**1. B.** Ha levételkor a játékosnak nincs megfelelő kapacitása az *Inventory*-jában, akkor az adott eszköz megsemmisül levételt követően.

5.1.3 Use-case diagram (II.)

5.1.4 Use-case leírások (II.)

Use-case neve	Time steps
Rövid leírás	Az időzítőket lépteti.
Aktorok	System
Forgatókönyv	**1.** Minden felvett, időzíthető objektum értékét egyel lépteti.

Use-case neve	Virologist steps
Rövid leírás	A virológus ágenseinek idejét csökkenti.
Aktorok	System
Forgatókönyv	**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.

Use-case neve	Oblivion steps
Rövid leírás	A feledés ágens léptetését ábrázolja.
Aktorok	System
Forgatókönyv	**1.** A már felkent ágenst lépteti, amely ezidő alatt a kiható virológussal elfeledteti a már megtanult genetikai kódokat. Feledést követően az ágens hatása megszűnik a virológusról, és maga az ágens megsemmisül.

Use-case neve	VitusDance steps
Rövid leírás	A vítustánc ágens léptetését ábrázolja.
Aktorok	System
Forgatókönyv	**1.** A már felkent ágenst lépteti, amely ezidő alatt a kiható virológusra random mozgást idéz elő. Amennyiben lejár az időzítő az ágens hatása megszűnik, maga az ágens megsemmisül.

Use-case neve	Immunity steps
Rövid leírás	Az immunitás ágens léptetését ábrázolja.
Aktorok	System
Forgatókönyv	**1.** A már felkent ágenst lépteti, amely ezidő alatt megvédi más ágensek felkenésétől a virológust. Amennyiben lejár az időzítő az ágens hatása megszűnik, maga az ágens megsemmisül.

Use-case neve	Paralysis steps
Rövid leírás	A bénulás ágens léptetését ábrázolja.
Aktorok	System
Forgatókönyv	**1.** A már felkent ágenst lépteti, amely ezidő alatt megbénítja, azaz cselekvőképtelenné és lophatóvá teszi a virológust. Amennyiben lejár az időzítő az ágens hatása megszűnik, maga az ágens megsemmisül.

5.2 A szkeleton kezelői felületének terve, dialógusok

A program indításkor egy listát ír ki a konzolra, melyben előre definiált forgatókönyvek közül lehet választani. Bemenetként ekkor a forgatókönyvhöz rendelt számok egyikét várja. A szám begépelése után kiírja a forgatókönyv nevét, majd sorra az annak elvégzése alatt végrehajtott függvényhívásokkal kapcsolatos információkat, a következőképpen:

```
{tabulálás} {az objektum neve, amelyen a függvényt hívtuk} {a függvény neve}
{soremelés}
```

Az objektumok a forgatókönyvben résztvevő objektumok, a függvények ezek publikus függvényei, a tabulálás mértékét pedig az határozza meg, hogy a hívási láncban milyen mélyen hívtuk meg a függvényt. Utóbbinak megfelelően, ha pl. az `X` objektum `a()` függvény belsejében meghívja az `Y` objektum `b()` függvényt, a `b()` az `a()` alatt és annál valamivel beljebb kezdődik a konzolon.

```
X a()
      Y b()
```

Viszont ha a `c()` függvény meghívja `a()`-t, majd annak visszatérése után `b()`-t, akkor `b()` az `a()` alatt és vele azonos tabulálással lesz látható, valamint mindketten `c()` alatt és annál beljebb.

```
Z c()
      X a()
      Y b()
```

A main függvény által először hívott függvény tabulálása 0-ás. Az egyes függvényeknek esetenként felhasználói beavatkozásra lesz szükségük, ilyenkor a kimenetre kiírnak egy kérdést (pl.: Szeretnél lopni?). Miután a felhasználó meghozta a döntést, beírja azt a konzolon, a program pedig tovább futhat. Pl.:

```
      f1.arrive()
Szeretnél lopni? (I/N) I
              v1.stealBy()
```

(Megj.: A kérdést a jobb olvashatóság érdekében nem tabuláljuk. A második sor végén álló I a felhasználó válasza, melyet helytakarékossági okokból a kérdéssel egy sorba várunk.)

5.3 Szekvencia diagramok a belső működésre

5.3.1 Timer steps

5.3.2 Immunity steps

5.3.3 Oblivion steps

5.3.4 Paralysis steps

5.3.5 VitusDance steps

5.3.6 Virologist steps

5.3.7 Virologist moves to a basic field

5.3.8 Virologist moves to a laboratory field

5.3.9 Virologist moves to a shelter field

5.3.10 Virologist moves to a storage field

5.3.11 Attack

sd attack
: Felhasználó
attacker : Virologist
attacked : Virologist
a : Agent
: Timer
1 : attack(attacked, a)
opt
[attacker.Paralyzed == false]
2 : assaulted(attacker, a)
3 : getHasEffectedAgent()
4 : bUnderEffect
opt
[bUnderEffect == false]
opt
[attacker != attacked]
5 : defense(attacker, a)
6 : bDefenseSuccessful
opt
[bDefenseSuccessful == false]
7 : setVirologistUnderEffect(attacked)
8
9 : addSteppable(a)
10
11 : bDefenseSuccessful
12 : getParalyzed()
13 : bParalyzed
opt
[bDefenseSuccessful == false]
opt
[bParalyzed == true]
14 : wannaSteal()
15 : bSteal
opt
[bSteal == true]
16 : stealBy(attacker)
17
18

5.3.12 Defense

5.3.13 Steal

sd steal
stealerVirologist : Virologist
paralizedVirologist : Virologist
iParalized : Inventory
1 : stealBy(stealerVirologist)
2 : getInventory()
3 : iStealer
4 : chooseObjectToSteal()
5 : objectName
iStealer : Inventory
alt
[objectName == "Aminoacid"]
6 : removeAmino()
amino : Aminoacid
7 : amino
8 : addAmino(amino)
9
[objectName == "Nukleotid"]
10 : removeNukleotid()
nukleo : Nukleotid
11 : nukleo
12 : addNukleo(nukleo)
13
[objectName == "Gloves"]
14 : removeGloves()
gloves : Gloves
15 : gloves
16 : addGloves(gloves)
17
[objectName == "Sack"]
18 : removeSack()
sack : Sack
19 : sack
20 : addSack(sack)
21
[objectName == "Cloak"]
22 : removeCloack()
cloak : Cloak
23 : cloak
24 : addCloak(cloak)
25
26

5.3.14 Create Agent

5.3.15 Equips gloves

5.3.16 Takes off gloves

5.3.17 Equips cloak

5.3.18 Takes off cloak

5.3.19 Equips sack

5.3.20 Takes off sack

5.4 Kommunikációs diagramok

5.4.1 Timer steps

5.4.2 Immunity steps

5.4.3 Oblivion steps

5.4.4 Paralysis steps

5.4.5 VitusDance steps

5.4.6 Virologist steps

5.4.7 Virologist moves to a basic field

5.4.8 Virologist moves to a laboratory field

5.4.9 Virologist moves to a shelter field

5.4.10 Virologist moves to a storage field

5.4.11 Attack

: Felhasználó
4 : attack()
attacker : Virologist
6 [attacker.Paralyzed == false] : getHasEffectedAgent()
7 : bUnderEffect
8 [attacker.Paralyzed == false && bUnderEffect == false && attacker != attacked] : defense(attacker, a)
9 : bDefenseSuccessful
14 [attacker.Paralyzed == false && bDefenseSuccessful == false && bParalyzed == true] : wannaSteal()
15 : bSteal
5 [attacker.Paralyzed == false] : assaulted(attacker, a)
12 [attacker.Paralyzed == false] : getParalyzed()
13 : bParalyzed
16 [attacker.Paralyzed == false && bDefenseSuccessful == false && bParalyzed == true && bSteal == true] : stealBy(attacker)
1 <<create>>
<<create>>
2
attacked : Virologist
: Rendszer
10 [attacker.Paralyzed == false && bDefenseSuccessful == false] : setVirologistUnderEffect(attacker)
11 [attacker.Paralyzed == false && bDefenseSuccessful == false] : addSteppable(a)
3 <<create>>
: Timer
a : Agent

5.4.12 Defense

5.4.13 Steal

iParaziled : Inventory
amino : Aminoacid
nukleo : Nukleotid
gloves : Gloves
sack : Sack
cloak : Cloak
: Rendszer
stealerVirologist : Virologist
paralizedVirologist : Virologist
iStealer
<<create>>
1
2
3 : setInventory(iStealer)
4
5
6 : setInventory(iParalized)
7
8
9
10
11
12 : stealBy(stealerVirologist)
13 : chooseObjectToSteal()
14 : objectName
15 [objectName == "Aminoadid"] : removeAmino()
16 : amino
17 [objectName == "Aminoadid"] : addAmino(amino)
18 [objectName == "Nukleotid"] : removeNukleo()
19 : nukleo
20 [objectName == "Nukleotid"] : addNukleo(nukleo)
21 [objectName == "Gloves"] : removeGloves()
22 : gloves
23 [objectName == "Sack"] : removeSack()
24 [objectName == "Cloak"] : removeCloack(objectName == "Cloack")
25 [objectName == "Gloves"] : addGloves(gloves)
26 [objectName == "Sack"] : addSack(sack)
27 [objectName == "Cloak"] : addCloak(cloak)
28 : sack
29 : cloak

5.4.14 Create Agent

5.4.15 Equips gloves

5.4.16 Takes off gloves

5.4.17 Equips cloak

5.4.18 Takes off cloak

5.4.19 Equips sack

5.4.20 Takes off sack

5.5 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	
ápr. 4.	Prototípus koncepciója – beadás Szkeleton bemutatás	20	
ápr. 11.	Részletes tervek - beadás	45	
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	

5.7 Napló

Kezdet	Időtartam	Résztvevő(k)	Leírás
2022.03.19 10:00	5 óra	Litavecz	Feladat: A kommunikációs diagrammok részletes megismerése, megtanulása, használatának áttekintése. A főbb kommunikációs diagrammok elkészítése. A use-case diagramok elkészítése.
2022.03.20 20:30	3 óra	Teljes csapat	Értekezlet: Az eheti házi értelmezése, a feladatok szétbontása, közös átbeszélése.
2022.03.20 23:30	1 óra	Alpek, Marton	Feladat: A szkeleton átbeszélése, szekvenciadiagramok közös átnézése.
2022.03.20 23:30	1 óra	Csia, Litavecz	Feladat: A use-case-ek átnézése, a use-case-ek leírásainak elkészítése.
2022.03.21 00:30	3 óra	Alpek, Litavecz, Marton	Feladat: A még hibás szekvenciadiagramok kijavítása, minden kommunikációs diagram elkészítése.
2022.03.21 00:30	4 óra	Csia	Feladat: A dokumentum egybe rakása, a külön megoldott feladatok egységesítése egy doksiba. A hozzá tartozó, még előző háziból való szekvencia diagram kijavítása, kommunikációs diagram elkészítése.

5.8 Százalékos teljesítés

Név	Százalék
Alpek	25%
Csia	20%
Litavecz	30%
Marton	25%

6. Szkeleton elkészítése

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Agent.java	2 KB	2020.03.25 17:40	Az ágensek absztrakt főosztálya.
Aminoacid.java	1 KB	2020.03.25 17:40	Az aminosav anyagot lekezelő osztály.
Cloak.java	2 KB	2020.03.25 17:40	A köpeny eszközt lekezelő osztály.
Equipment.java	1 KB	2020.03.25 17:40	Az eszközök absztrakt főosztálya.
Field.java	4 KB	2020.03.25 17:40	A mezők absztrakt főosztálya, illetve a sima (üres) mező osztálya.
Game.java	1 KB	2020.03.25 17:40	A játék indításáért és leállításáért felelős osztály.
GCode.java	3 KB	2020.03.25 17:40	A genetikai kódot és az ágensgenerálást lekezelő osztály.
Gloves.java	2 KB	2020.03.25 17:40	A kesztyű eszközt lekezelő osztály.
Immunity.java	2 KB	2020.03.25 17:40	A sérthetetlenséget okozó ágenst lekezelő osztály.
Inventory.java	8 KB	2020.03.25 17:40	A játékosnál lévő alap tárolót és a hozzá tartozó akciókat lekezelő osztály.
Lab.java	2 KB	2020.03.25 17:40	A labor típusú mezőt lekezelő osztály.
Logger.java	1 KB	2020.03.25 17:40	A teszteléshez szükséges logoló osztály.
Material.java	1 KB	2020.03.25 17:40	Az anyagok absztrakt főosztálya.
Nukleotid.java	1 KB	2020.03.25 17:40	A nukleotid anyagot lekezelő osztály.
Oblivion.java	1 KB	2020.03.25 17:40	A feledést okozó ágenst lekezelő osztály.
Paralysis.java	1 KB	2020.03.25 17:40	A bénulást okozó ágenst lekezelő osztály.

Program.java	14 KB	2020.03.25 17:40	A főosztály, ahonnan indul a program, itt találhatóak a szenáriók is.
Sack.java	1 KB	2020.03.25 17:40	A zsák eszközt lekezelő osztály.
Shelter.java	3 KB	2020.03.25 17:40	Az óvóhely típusú mezőt lekezelő osztály.
Storage.java	3 KB	2020.03.25 17:40	A raktár típusú mezőt lekezelő osztály.
Virologist.java	18 KB	2020.03.25 17:40	A játékos karakterét és a hozzá tartozó akciókat lekezelő osztály.
VitusDance.java	1 KB	2020.03.25 17:40	A vitustáncot okozó ágenst lekezelő osztály.

6.1.2 Fordítás

A becsomagolt forrásfájlokat töltsük le a saját gépünkre, majd csomagoljuk ki őket. Navigáljunk el a kicsomagolt mappába, majd olyan IDE-ben nyissuk meg a teljes mappát (ahol az összes fájl található), ahol már van előre feltelepítve Java 11. Fordítsuk a beépített fordító segítségével.

6.1.3 Futtatás

Ha már sikerült egyszer lefordítanunk, akkor ugyanabban a környezetben ugyanezt megtehetjük futásnál is a beépített futtató segítségével. Ez akár történhet gyorsbillentyű segítségével. (Ez van, hogy F5, F9, de van, hogy F10-es gomb segítségével is történhet.

6.2 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	
ápr. 11.	Részletes tervek - beadás	45	
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	

6.3 Napló

Kezdet	Időtartam	Résztvevő(k)	Leírás
2022.03.25 17:00	1 óra	Alpek, Csia, Marton	Feladat: A következő feladat átbeszélése, részekre szedése. VM-ek elindítása, működésének letesztelése.
2022.03.25 18:00	óra	Alpek, Csia, Marton	Feladat: A szkeleton alapjainak közös megbeszélése, program osztályainak közös elkészítése.
2022.03.25 2:00	1 óra	Marton	Feladat: Feladatok kiosztása a csapattagok között. Közös leadási, összefésülési időpont megbeszélése.
2022.03.26 19:30	2 óra	Teljes csapat	Feladat: A feltöltött file-ok aszinkron átnézése.

2022.03.27. 13:00	3 óra	Teljes csapat	Értekezlet: Maradék teendő átbeszélése, megjegyzések hozzáfűzése az eddig feltöltött doksikhoz.
2022.03.27. 16:00	1 óra	Marton	Feladat: Feladatok kiosztása (szenáriók írása) a csapattagok között.
2022.03.27. 20:00	7 óra	Alpek	Feladat: Agent, Cloak fele, Equipment, Gloves fele, Immunity, Oblivion, Paralysis, Sack, VitusDance file-ok megírása belső működéssel.
2022.03.27. 20:00	13 óra	Csia	Feladat: Aminoacid, Field, GCode, Lab, Material, Nukleotid, Shelter, Storage file-ok megírása belső működéssel, hozzájuk Javadoc komentek megírása és releváns szenáriók elkészítése.
2022.03.27. 20:00	11 óra	Litavecz	Feladat: Cloak és Gloves file-ok megírása belső működéssel, hozzájuk Javadoc komentek megírása és releváns szenáriók elkészítése.
2022.03.27. 20:00	14 óra	Marton	Feladat: Game, Inventory, Logger, Program, Virologist file-ok megírása belső működéssel, hozzájuk Javadoc komentek megírása és releváns szenáriók elkészítése.
2022.03.28. 05:30	30 perc	Csia	Feladat: Dokumentum összerakása, véglegesítése.

6.4 Százalékos teljesítés

Név	Százalék
Alpek	25%
Csia	25%
Litavecz	25%
Marton	25%

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, magyarán 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 elhasználódik, vagy ellopják, ez a metódus hívódik meg.

boolean takeOffAxe(Inventory inv): Amikor a játékos vissza szeretné helyezni a fejszéjét 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 a 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 játékossal, 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 ágenst.
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 ágenst.
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 ágenst.
Opciók: -

Parancs: 3

Leírás: TakeOffGlovesBackToInventoty: 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 **TakeOffSackBackToInventoty** 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 felszereléseihez 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 rákeni 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 bénultra 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 ágenst 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 ágenst, 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 Paralisys ágenst 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 ágenst, 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 ágenst. A v1 megpróbálja rákenni az ágenst 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 ágenst 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 ágenst, 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 ágenst, 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 ágenst 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<ágenstypus>

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 létrehoz 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 enm 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 ágenst a gc alapján.
Opciók: -

Parancs: 49

Leírás: DanceWhileMove: A teszt létrehoz egy Game (g), két Field (f1 és f2) és egy Virológus (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 Virológus objektumnak a konstruktora paramétereként. 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 ágenst (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 ágenst, 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 ágenst (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 ágenst (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 ágenst 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 meghiúsul.
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 meghiúsul.
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 meghiúsul.
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 meghiúsul.
5. StealAminoacid (bemenet: a): v1 virológus aminoacidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás meghiúsul.
6. StealNukleotid (bemenet: n): v1 virológus nukleotidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás meghiúsul.

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 meghiúsul.
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 meghiúsul.
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 meghiúsul.
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 meghiúsul.
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 meghiúsul.

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 paramétereként. 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 meghiúsul.
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 meghiúsul.
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 meghiúsul.
5. StealAminoacid (bemenet: a): v1 virológus aminoacidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás meghiúsul.
6. StealNukleotid (bemenet: n): v1 virológus nukleotidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás meghiúsul.

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 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 meghiúsul.
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 meghiúsul.
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 meghiúsul.
4. StealAminoacid (bemenet: a): v1 virológus aminoacidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás meghiúsul.
5. StealNukleotid (bemenet: n): v1 virológus nukleotidot akar lopni v2-től, melyet nem talál meg az inventory-jában, így a lopás meghiúsul.

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 invhez 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 rákeni 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 ágensek, 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)/VitusDance(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 VitusDance (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 VitusDance (ágenstípus: v).
4. A felkenni kívánt ágens egy Immunity (ágenstípus: i).

Parancs: 76

Leírás: UseAgenetOnVirologistWhoHasAxeGlovesAndCloak: A teszt létrehoz egy Game és egy Field objektumot. Ezután létrehoz egy inventory objektumot, melynek átadunk 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 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: UseAgenetOnVirologistWhoHasBearDance: 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 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 glov 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

Kiemenet: 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 paralized. Can not perform action.
Opciók: -

Parancs: 10<ágenstípus>

Kimenet: Virologist paralized. 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.

Elvárt kimenet opciók:

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

Parancs: 24 <ágenstípus>

Kimenet: Virologist has no agnets. Can not perform action.
Opciók: -

Parancs: 25

Kimenet: Virologist paralized. 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 paralized. 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 paralized. 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 paralized. 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 paralized. 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 meghiúsul, 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 meghiúsul, 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, 1, 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: 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 meghiúsul, 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, 1, 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: 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 meghiúsul, 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 meghiúsul, 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** av2 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 inicalizásáért 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.

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

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

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

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

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

Forgatókönyv	1. A játékos elkészít egy adott ágens, melyhez rendelkezik a megfelelő mennyiségű alapanyaggal. 2. Az elkészült ágenst eltárolja magánál egészen felhasználásig vagy az ágens idejének lejártájáig.

Use-case neve	Attacks other Virologist
Rövid leírás	A játékos megtámad egy másik virológust.
Aktorok	Felhasználó
Forgatókönyv	1. A játékos kiválaszt az ellenfele közül egyet, akivel azonos mezőn tartózkodik. 2. A játékos az általa elkészített ágensek közül választ egyet. 3. A játékos megtámadja a már kiválasztott játékost a már kiválasztott ágenssel.

Use-case neve	Defense
Rövid leírás	A játékos megpróbál kivédeni egy támadást.
Aktorok	Felhasználó
Forgatókönyv	1. A játékos látja, hogy megtámadják, így kiválasztja a felhelyezett eszközei közül, hogy melyikkel fog védekezni a támadással szemben.

Use-case neve	Steals from paralyzed Virologist
Rövid leírás	A játékos meglop egy másik játékost.
Aktorok	Felhasználó
Forgatókönyv	1. A játékos egy olyan mezőre lép, ahol tartózkodik lebénult virológus. 2. A játékos kiválasztja, hogy szeretne lopni a lebénult játékostól. 3. A játékos kiválasztja, hogy miket szeretne elvenni a másik játékostól.

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

Forgatókönyv	1. A játékos óvóhely típusú mezőre lép. 2. A játékos elvesz az óvóhelyről eszköz(öke)t.

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

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

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

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

Use-case neve	Takes off Sack to Inventory

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

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

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

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

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

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

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

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

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

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

Forgatókönyv	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.

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

7.3 Tesztelési terv

Teszt-eset neve	
Rövid leírás	
Teszt célja	

Teszt-eset neve	
Rövid leírás	
Teszt célja	

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

7.5 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	
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	

7.6 Napló

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

7.7 Százalékos teljesítés

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

8. Részletes tervek

8.1 Osztályok és metódusok tervei

8.1.1 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:** +

8.1.2 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:** +

8.1.3 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:** +

8.1.4 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:** +

8.1.5 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:** +

8.1.6 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:** -

8.1.7 Material

- **Felelősség**

A raktárból megszerezhető anyagokat reprezentáló absztrakt osztály.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

Nincs

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

8.1.8 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:** +

8.1.9 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:** +

8.1.10 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: +

8.1.11 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:** +

8.1.12 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.

8.1.13 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:** -

8.1.14 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:** -

8.1.15 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:** +

8.1.16 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 VirologistUnderEffect attribútumot 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:** +

8.1.17 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:** +

8.1.18 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:** +

8.1.19 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:** +

8.1.20 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: +

8.1.21 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: -

8.1.22 Steppable

- **Felelősség**

Egy interface, amely a léptethető objektumokat kezeli.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

Nincs

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

8.1.23 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: +

8.1.24 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:** +

8.1.25 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:** +

8.2 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.

8.2.1 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

8.2.2 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.??

8.2.3 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.

8.2.4 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.

8.2.5 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

8.2.6 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.

8.2.7 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.

8.2.8 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

8.2.9 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

8.2.10 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.

8.2.11 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.

8.2.12 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.

8.2.13 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.

8.2.14 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

8.2.15 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.

8.2.16 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?

8.2.17 *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...

8.2.18 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

8.2.19 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.

8.2.20 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.

8.2.21 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.

8.2.22 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

8.2.23 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.

8.2.24 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.

8.2.25 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.

8.2.26 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.

8.3 A tesztelést támogató programok tervei

8.3.1 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.

8.3.2 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.

8.3.3 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.

8.4 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	

8.5 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.

8.6 Százalékos teljesítés

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

10. Prototípus beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Agent.java	3 KB	2022.03.25 17:40	Az ágensek absztrakt főosztálya.
Aminoacid.java	1 KB	2022.03.25 17:40	Az aminosav anyagot lekezelő osztály.
Axe.java	2 KB	2022.04.24 12:00	A fejsze eszközt lekezelő osztály.
BearDance.java	3 KB	2022.04.24 12:00	A medvetáncot okozó ágenst lekezelő osztály.
Cloak.java	3 KB	2022.03.25 17:40	A köpeny eszközt lekezelő osztály.
Equipment.java	1 KB	2022.03.25 17:40	Az eszközök absztrakt főosztálya.
Field.java	4 KB	2022.03.25 17:40	A mezők absztrakt főosztálya, illetve a sima (üres) mező osztálya.
Game.java	19 KB	2022.03.25 17:40	A játék indításáért és leállításáért felelős osztály.
GCode.java	3 KB	2022.03.25 17:40	A genetikai kódot és az ágensgenerálást lekezelő osztály.
Gloves.java	3 KB	2022.03.25 17:40	A kesztyű eszközt lekezelő osztály.
Immunity.java	1 KB	2022.03.25 17:40	A sérthetetlenséget okozó ágenst lekezelő osztály.
InfectedLab.java	2 KB	2022.04.24 12:00	A fertőző labor típusú mezőt lekezelő osztály.
Inventory.java	7 KB	2022.03.25 17:40	A játékosnál lévő alap tárolót és a hozzá tartozó akciókat lekezelő osztály.
Lab.java	2 KB	2022.03.25 17:40	A labor típusú mezőt lekezelő osztály.

Material.java	1 KB	2022.03.25 17:40	Az anyagok absztrakt főosztálya.
Nukleotid.java	1 KB	2022.03.25 17:40	A nukleotid anyagot lekezelő osztály.
Oblivion.java	2 KB	2022.03.25 17:40	A feledést okozó ágenst lekezelő osztály.
Paralysis.java	1 KB	2022.03.25 17:40	A bénulást okozó ágenst lekezelő osztály.
Program.java	1 KB	2022.03.25 17:40	A főosztály, ahonnan indul a program, itt találhatóak a szenáriók is.
Sack.java	1 KB	2022.03.25 17:40	A zsák eszközt lekezelő osztály.
Shelter.java	4 KB	2022.03.25 17:40	Az óvóhely típusú mezőt lekezelő osztály.
Steppable.java	1 KB	2022.04.24 12:00	A léptetőket kezelő osztály.
Storage.java	4 KB	2022.03.25 17:40	A raktár típusú mezőt lekezelő osztály.
Timer.java	2 KB	2022.04.24 22:00	Az időzítőket tárolja.
Virologist.java	17 KB	2022.03.25 17:40	A játékos karakterét és a hozzá tartozó akciókat lekezelő osztály.
VitusDance.java	3 KB	2022.03.25 17:40	A vitustáncot okozó ágenst lekezelő osztály.

10.1.2 Fordítás

A becsomagolt forrásfájlokat töltsük le a saját gépünkre, majd csomagoljuk ki őket. Navigáljunk el a kicsomagolt mappába, majd olyan IDE-ben nyissuk meg a teljes mappát (ahol az összes fájl található), ahol már van előre feltelepítve Java 11. Fordítsuk a beépített fordító segítségével. (Jelen esetben érdemes akkor ezt Eclipse-ben megnyitni és fordítani.)

10.1.3 Futtatás

Ha már sikerült egyszer lefordítanunk, akkor ugyanabban a környezetben ugyanezt megtehetjük futásnál is a beépített futtató segítségével. Ez akár történhet gyorsbillentyű segítségével. (Ez van, hogy F5, F9, de van, hogy F10-es gomb segítségével is történhet. (Jelen esetben érdemes akkor ezt Eclipse-ben megnyitni és futtatni.)

10.2 Tesztek jegyzőkönyvei

Teszteset neve	1. UseAgentOnVirologistNoAgentOnAttacker
Tesztelő neve	Ruskó Eszter
Teszt időpontja	2022.04.26 22:37
Teszt eredménye	Sikeres
Megjegyzés	A leírt tesztben bénítással támad a virológus, ami után megkérdezi a program, hogy akar-e lopni tőle. Ez a teszt dokumentálásából kimaradt.

Teszteset neve	2. UseAgentOnVirologistAgentOnAttacker
Tesztelő neve	Alpek Dávid
Teszt időpontja	2022.04.27 02:11
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja. **Kimenet:** • Game over. hozzáadva, a quit bemenet hatására.

Teszteset neve	3. UseAgentOnVirologistAxeWithDefender
Tesztelő neve	Csia Kitti
Teszt időpontja	2022.04.27 00:57
Teszt eredménye	Sikeres
Lehetséges hibaok	Defend része nem történt meg, csak végtelen ciklusba került a V1 és V2 felváltva.
Változtatások	Bementi részbe nincs megadva az eszközhasználatra való parancs.

Teszteset neve	4. UseAgentOnVirologistGlovesWithDefender
Tesztelő neve	Marton Judit
Teszt időpontja	2022.04.27 04:21
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • 3. sor create Virologist paraméterek sorrendje az előző dokumentációban fel volt cserélve, itt ez már javítva szerepel. • Attack és V2 parancsok után kell egy sor, Paralysis (ez megadja, hogy mit használunk a támadáshoz/kenéshez) • n hozzáadása a végére, ami azt adja meg, hogy nem akarunk lopni (hiszen Paralysis-sel támadtunk, így aki kesztyűvel visszadobja, és megfertőzi a támadót, lophatna tőle, hiszen lebénult) • quit hozzáadása, ami az alkalmazás futását leállítja **Kimenetek:** • A lehetséges beviteli parancsok felsorolásának hozzáadása a játék indítása után • Axe found, Gloves found, Aminoacid found, Nukleotid found sorok beszúrása, amikor valamit ki akarunk venni az inventoryból, ez az üzenet jelzi, hogy az adott dolog benne is volt az inventoryban, így ki tudtuk venni. • A támadáshoz használható ágensek kilistázása, jelenleg csak egyetlen ágensünk van elkészítve: Paralysis • infection successful és infection defended üzenetek sorrendjének felcserélése, és közben annak megkérdezése, hogy aki Paralysis-sel sikeresen infectált, az akar-e lopni.

Teszteset neve	5. UseAgentOnVirologistDefendWithCloak
Tesztelő neve	Ruskó Eszter
Teszt időpontja	2022.04.27 03:39
Teszt eredménye	Sikeres
Lehetséges hibaok	1) Hibás virológus konstruktor hívás, inputfile pontatlansága, hibás equip hívás. 2) Hiányos inputfile.

Változtatások	1) Inputfile javítása (V2 veszi fel a cloakot, nem V1). 2) Inputfile javítása (V2 használja a cloakot védekezésre).

Teszteset neve	6. UseAgentOnVirologistCloakFailed
Tesztelő neve	Alpek Dávid
Teszt időpontja	2022.04.27 04:10
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja. **Kimenet:** • Game over. hozzáadva, a quit bemenet hatására.

Teszteset neve	7. BearDanceOverWriteAgent
Tesztelő neve	Csia Kitti
Teszt időpontja	2022.04.27 04:15
Teszt eredménye	Sikeres
Lehetséges hibaok	Virologist nem kreálódott.
Változtatások	Input fájl javítása.

Teszteset neve	8. BearDanceInfectedVirologistInfectOtherVirologistWhoHasImmunity
Tesztelő neve	Marton Judit
Teszt időpontja	2022.04.27 03:22
Teszt eredménye	Sikeres

Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja **Kimenetek:** • A lehetséges beviteli parancsok felsorolásának hozzáadása a játék indítása után • Kimaradt a BearDance step. (Szerintem ezt minden ágens step-jéből lehagytuk • Select neighbor: Field2 sor, mely bekéri, hogy teszt üzemmódban melyik mezőre szeretnénk léptetni az éles környezetben random szomszédra lépő virológust. • V2 Beardance infection defended with Immunity. sor helyett Virologist under Immunity, cannot get effect. szöveget várunk • V1 under BearDance, skip step. hozzáadása, ami azért íródik ki, mert a next paranccsal pont ő kerülne sorra, de mivel nem képes cselekedni, ezért ezzel a kimenettel tovább is ugorjuk, nem várunk arra, hogy parancsokat adjon be. • V2 turn. Üzenet jelzi, hogy a V2 virológus lett aktív • Game over üzenetet a quit parancs után kapunk.

Teszteset neve	9. CreateAgentNoMaterial
Tesztelő neve	Ruskó Eszter
Teszt időpontja	2022.04.27 01:51
Teszt eredménye	Sikeres

Teszteset neve	10. CreateAgent
Tesztelő neve	Alpek Dávid
Teszt időpontja	2022.04.27 04:55
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja **Kimenet:** • Game over. hozzáadva, a quit bemenet hatására

Teszteset neve	11. UseAgentOnVirologistButVirologistUnderEffect
Tesztelő neve	Csia Kitti
Teszt időpontja	2022.04.27 02:33
Teszt eredménye	Sikeres

Teszteset neve	12. SelfUseAgent
Tesztelő neve	Marton Judit
Teszt időpontja	2022.04.27 03:10
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja **Kimenetek:** • A lehetséges beviteli parancsok felsorolásának hozzáadása a játék indítása után • Az inventory-ól minden egyes elemnél, amikor kivesszük, akkor egy Aminoacid/Nukleotid found üzenetet írunk ki. • Az Immunity elé nem tettük ki az Available Agents üzenetet • Kimaradt az Agent used, V1 under Immunity üzenet • Game over, ami a játék végén íródik ki

Teszteset neve	13. SelfUseAgentUnderEffect
Tesztelő neve	Ruskó Eszter
Teszt időpontja	2022.04.26 21:20
Teszt eredménye	Sikeres

Teszteset neve	14. MoveToBasicField
Tesztelő neve	Alpek Dávid
Teszt időpontja	2022.04.27 00:33
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja. **Kimenet:** • Game over. hozzáadva, a quit bemenet hatására.

Teszteset neve	15. MoveWhileParalyzed
Tesztelő neve	Csia Kitti
Teszt időpontja	2022.04.27 03:16
Teszt eredménye	Sikeres

Teszteset neve	16. MoveWhenDancing
Tesztelő neve	Marton Judit
Teszt időpontja	2022.04.26 23:58
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja **Kimenetek:** • A lehetséges beviteli parancsok felsorolásának hozzáadása a játék indítása után • Az előző dokumentumban rosszak voltak itt a Virologist ID-k, ez itt helyesen jelenik meg (a soron következő játékosról szóló üzenetekben) • Game over üzenet, ami a program termináláskor jelenik meg.

Teszteset neve	17. MoveToShelter
Tesztelő neve	Ruskó Eszter
Teszt időpontja	2022.04.27 03:04
Teszt eredménye	Sikeres
Lehetséges hibaok	1) Hibás inputfile. 2) A program kis-és nagybetű kezelésében lehetséges a hiba.
Változtatások	1) Inputfile javítása (tesztesetben Field1 helyett Shelter1-re lép). 2) move parancs változtatása (v.move, toLowerCase), illetve Wannapickupra válaszadás.

Teszteset neve	18. MoveToLab
Tesztelő neve	Alpek Dávid
Teszt időpontja	2022.04.27 00:41
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja. **Kimenet:** • Game over. hozzáadva, a quit bemenet hatására.

Teszteset neve	19. MoveToStorage
Tesztelő neve	Csia Kitti
Teszt időpontja	2022.04.27 03:18
Teszt eredménye	Sikeres

Teszteset neve	20. MoveToInfectedLab
Tesztelő neve	Marton Judit
Teszt időpontja	2022.04.27 04:24
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja • a create Field és create Virologist parancsoknál az előző dokumentumban kimaradt egy szóköz, az mindenképp kell, és a setneighbor parancs sem jól van írva az előző dokumentumban. • Immunity parancs, ami megadja, hogy melyik kód legyen a labor falára vésve. • next parancs hívása a move előtt, anélkül nem működik a teszt, mivel a next inicializálja az aktív játékost. **Kimenetek:** • A lehetséges beviteli parancsok felsorolásának hozzáadása a játék indítása után • A randomizált működés kiiktatása miatt teszt üzemmódban megkérdezzük, hogy milyen kódd legyen a labor falára vésve. • V1 turn. A hozzáadott next parancs kimenete • Neighbours: InfectedLab1 helyett Select neighbor: InfectedLab1 • Game over üzenet, ami a program terminálásakor jelenik meg.

Teszteset neve	21. MoveToStorageAParalyzedVirologistThere
Tesztelő neve	Ruskó Eszter
Teszt időpontja	2022.04.27 02:15
Teszt eredménye	Sikeres

Teszteset neve	22. NextWithHealthyVirologist
Tesztelő neve	Alpek Dávid
Teszt időpontja	2022.04.27 03:17
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja. **Kimenet:** • Game over. hozzáadva, a quit bemenet hatására.

Teszteset neve	23. NextWithBearDancingVirologist
Tesztelő neve	Csia Kitti
Teszt időpontja	2022.04.27 04:34
Teszt eredménye	Sikeres

Teszteset neve	24. NextVirologistWhoIsUnderVitusDance
Tesztelő neve	Marton Judit
Teszt időpontja	2022.04.27 02:45
Teszt eredménye	Sikeres
Változtatások	**Cím:** Előző dokumentumban a teszt neve: next parancs vitustáncos virológusnál **Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja • létrehozunk még egy virológust, hogy legyen kinek átadni a kontrol-t és ne kerüljünk végtelen ciklusba azzal, hogy nincs mozgásképes virológus • A next parancsot kétszer adjuk ki egymás után, hogy a Timer.tick meghívódjon a kör végén, ezután választjuk ki (randomizált működés helyett csak teszt üzemmódban), hogy hová lépjen az ágens hatása alatt kényszeresen mozgó virológus.

	Kimenetek: • A lehetséges beviteli parancsok felsorolásának hozzáadása a játék indítása után • Field1 helyett New Field Field1 kimenet, a következő sorra is igaz Field2 létrehozása után. • V1 helyett New Virologist V1, és a most hozzáadott virológusra is várjuk a kimenetet létrehozás után. • V2 turn. A skip step után átadjuk a kontrolt egy lehetséges aktív virológusnak. • Remaining time üzenet kimaradt. • A következő next parancs kimenetei hozzá lettek adva • Game over. üzenet, ami a program terminálásakor íródik ki.

Teszteset neve	25. NextWithParalyzedVirologist
Tesztelő neve	Ruskó Eszter
Teszt időpontja	2022.04.27 06:10
Teszt eredménye	Sikeres

Teszteset neve	26. NextWithOblivionedVirologist
Tesztelő neve	Alpek Dávid
Teszt időpontja	2022.04.27 06:10
Teszt eredménye	Sikeres
Változtatások	**Bemenetek:** • quit hozzáadása, ami az alkalmazás futását leállítja. **Kimenet:** • Game over. hozzáadva, a quit bemenet hatására.

Teszteset neve	27. SelfUseAgentWithAgent
Tesztelő neve	Ruskó Eszter
Teszt időpontja	2022.04.26 20:10
Teszt eredménye	Sikeres

10.3 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	Litavecz
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	

10.4 Napló

Kezdet	Időtartam	Résztvevő(k)	Leírás
2022.04.23. 14:00	3 óra	Litavecz	Feladat: A kódok átírásának megkezdése.
2022.04.24 11:00	13 óra	Teljes csapat	Megbeszélés: A feladat értelmezése, eddigi közös kódok átnézése, mindenkit érintő tesztek közös megírása.
2022.04.25 19:00	6 óra	Litavecz	Feladat: A program maradék részének megírása, tesztekre felkészítés.
2022.04.26 19:00	1 óra	Alpek, Csia, Marton, Ruskó	Megbeszélés: A tesztek megírásának szétosztása a csapattagok között.
2022.04.26 22:30	6 óra	Alpek	Feladat: A saját, kiosztott rész megírása, feltöltése a közös mappába, és a tesztek közben kialakuló hibák javítása.
2022.04.26 23:30	5 óra	Csia	Feladat: A saját, kiosztott rész megírása, feltöltése a közös mappába, és a tesztek közben kialakuló hibák javítása.
2022.04.26 22:30	16 óra	Marton	Feladat: A saját, kiosztott rész megírása, feltöltése a közös mappába, és a tesztek közben kialakuló hibák javítása.
2022.04.26 22:00	6 óra	Ruskó	Feladat: A saját, kiosztott rész megírása, feltöltése a közös mappába, és a tesztek közben kialakuló hibák javítása.
2022.04.27 17:00	1 óra	Csia	Feladat: A dokumentum összeszedése, egyberakása, maradék szöveg kitöltése.

10.5 Százalékos teljesítés

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

11. Grafikus felület specifikációja

11.1 A grafikus interfész

11.1.1 Karakter képek

Itt azok a képek találhatóak meg, amelyek több játékos esetén rajzolja ki a különböző színű karaktereket, illetve viselt eszközöknél, hogy fog kinézni a *CharacterView*-ban a karakter, és bónuszként, hogy mivé fog alakulni egy medvetánccal fertőzött karakter. A zsák nem fog látszódni egy ábrán se, ugyanis az nálunk úgy van lekezelve, hogy csak plusz számláló bővítőként van hozzáadva a karakter *Inventory*-jához, így ahhoz nem készült rajz.

Az alap karakterünk, ez lesz a kiinduló verzió.

Ha több játékos is belekerül, akkor ilyen színnel fognak megjelenni a pályán.

Alap karakterünk [balról jobbra] kesztyűvel, fejszével, majd kesztyűvel és fejszével. Ezek a verziók + a köpenyes verziók nem fognak látszódni az alap játékban, hanem kizárólag csak a *CharacterView*-ban, ugyanis alapból azt feltételezzük, hogy a karakterek vakok, így mindig csak az alap ruhát látják egymáson.

A karakterünk [balról jobbra] köpennyel, köpennyel és fejszével, köpennyel és kesztyűvel majd a legvégén mind a három eszközzel együtt.

Így néz ki egy medvetánccal fertőzött játékos, természetesen ez csak mások szemében néz ki így, ugyanis, akit megfertőztek az már csak a Game Over feliratot fogja látni. Másik játékosok szemében azonban egy folyamatosan mozgó, raktárt fosztó, véres-szemű, fertőző medvét fog látni, aki, ha megjelenik egy pályán számolhatunk a problémákkal.

11.1.2 Objektum képek

Itt olyan képek találhatóak meg, melyeket a játékos maga tud eszközölni, például az *Inventory* és *CharacterView*, majd a *SkillBar*.

A játékos *Inventory*-ja a játék kezdetében. Felső részben látható a karakter, ahogy éppen kinéz, és milyen eszközök találhatóak rajta. [Fentről lefele] látható a fejsze, köpeny, kesztyű végül a zsák helye. Az alsó részen meg látható az *Inventory*, amelynek ez a kezdeti kapacitása, és a zsákkal a maradék három szekció is feloldható majd.

Itt már láthatóak a felvett eszközök a karakterre. Egy felvett fejsze, egy köpeny és két zsák látható felvéve. Az is jól látható, hogy a felvett eszközök kihatással vannak a *CharacterView*-ban látható játékosra, illetve a zsák az *Inventory* kinézetére (azaz a kapacitásra).

Itt az *Inventory* használata látható már. Jelenleg a játékos rendelkezik egy fejszével egy zsákkal, illetve felvett már 3 db aminósavat is, mely belekerült az *Inventory*-jába. Ha az *Inventory*-ba például eszköz kerül be, akkor annak a felvételét duplaklikkel lehet megvalósítani, illetve ugyanúgy a levétel is így működik. *Inventory*-ból ki is lehet dobni dolgokat, ha már nincs elegendő hely, hiszen a már kicsorbult fejsze nem fog felhalmozódni a már táskában lévő, de még nem használt fejszére, ezért is kell ilyen sok hely.

Így néz ki a kezdetben az úgynevezett *SkillBar*, azaz, ahol az ágensek készítése, tárolása, és az épp karakterre ható ágensek mutatása a felhasználó felé folyik. Bal oldalsó körbe kerül bele az éppen aktuálisan a karakterre ható ágens, lehet az saját maga által feltett ágens, vagy mások által rátett, és lent a bal alsó sarokban jelenik majd meg, hogy hány másodperc maradt hátra még az adott ágens hatóidejéből. Majd [balról jobbra] a belső sávban látható a Vitustánc ikonja, a bénulás, felejtés, végül az immunitás. Itt már két kis rubrika is tartozik egy adott ágenshez, ami arra lesz jó majd, hogyha egy adott ágensből több is elkészült, akkor felső rubrika fogja számlálni a darabszámot, hogy jelenleg hány darab van a játékosnál, alsó rubrikába, meg elkezd visszaszámolni, hogy mennyi ideig lehet még az adott ágenst felhasználni. Mindig a legrégebben elkészült ágens került a „legtetejére a listának”. így ha az elhasználódik, akkor az alatt levő visszaszámlálója váltja fel.

Itt már egy aktív, játék közepén elképzelhető *SkillBar* látható különböző jelzésekkel. Látható, hogy jelenleg a virológusra 1 körig még hat a bénulás, illetve rendelkezik 1 darab Vitustánc támadással, amelyet még 2 körig felhasználhat, 2 darab felejtéssel, melyek közül a legrégebben elkészültet még 3 körig használhatja fel, illetve látszik még, hogy immunitásból szerzett egy genetikai kódot, viszont még nem készítette el az ágenst, melyet a jobb oldalt látható egy gomb, amin egy Erlenymeyer üvegcse van, ha arra rákattintunk, akkor a megpróbál egy ágenst legenerálni, azaz megnézi, hogy van-e megfelelő genetikai kód, illetve megfelelő mennyiségű anyag, és ha van, akkor elkészül az ágens, és hozzá a körös visszaszámláló. Ekkor felkerül annál a gombnál egy 1-es a bal felső sarokban található rubrikába.

11.1.3 Háttérképek

Itt találhatóak a mezők maguk, ugyanis nem „tile game” szerű játékot álmodtunk meg, hanem kihoztuk az egész játékot egy „click and point adventure game”-ként a szebb megjelenítés és a jobb játékélmény érdekében, továbbá nekünk is egyszerűbb volt így elkészíteni, mintsem tile. Első körben itt most csak egy mezőtípust tettünk be példának (sima default/üres mező), majd ugyanazon a mező egy, majd több játékossal (medvetánccal fertőzöttel is), *SkillBar*ral, *Inventory* plusz *CharacterView* meghívásával.

Itt látható az üres háttérkép, amely a háttér lesz az üres mezők során. Még egy kis javítás, további árnyékolások eszközölve lesznek a rendes játékba helyezés előtt.

Játék kezdetében nagyjából ilyen látvány fog tárulni a játékosok elé. Lesz egy üres *SkillBar*, egy üres karakter, tehát nem lesz még rajta semmilyen felszerelés, illetve egy üres *Inventory* is. Ha végzett az adott köri feladatával a *Next*, gombbal tud majd a következő játékosnak átadni a stafétát. Ha az ajtókra kattint, azzal tud majd helyszínt váltani, természetesen az, hogy melyik ajtó mögött mi van, azt nem tudhatja előre. A fényesen fehéren világító ajtó lesz mindig az, ahonnan érkezett. Fent a jobb felső sarokban a kis orvos ikonnal tudja előhívni a karakter-tároló panelt, ami látható a képen is jelenleg, illetve a hang gombbal tudja elnémítani vagy éppen hagyni, hogy a játékban bent lévő zene fusson vagy sem. Az x gomb segítségével tud kilépni a játékból, viszont, ha eközben még van bent más játékos, akkor nekik a játék folytatódik tovább.

Itt a képen már egy játék közbeni verziót láthatunk. Jelenleg a karakterünk le van bénulva, és a lenti szöveges szekció is szól ezzel kapcsolatban. Ilyenkor nem tudunk rányomni a *Next* gombra, mert automatikusan tovább lépteti a következő játékosra. A képen még látható, hogy van a pályán jelenleg egy medvevírussal fertőzött játékos is, illetve szintén egy, még medvevírussal nem megfertőződött játékos.

11.1.4 Menü

Itt látható egy skicc terv a játék betöltését követő menü rendszerről, és almenüikről.

A játék betöltését követően ez a három menüpont vár majd minket, amelyek viszonylag egyértelműen jelzik, hogy mi történik a kattintásukat követően. A *Start Game*-et kiválasztva elkezdődik a játék, és átvált a fentebb látható nézetre.

Abban az esetben, ha a *Settings*re kattintunk, átmegy egy másik menübe, ahol választhatunk aközül, hogy a pálya kinézetén változtatunk, a játékosok számán, vagy visszamegyünk a főmenübe. Főmenü esetében az előző képet láthatjuk majd.

Ha a *Board*ra megyünk, akkor megjelenik ez a nézet, ahol beállíthatjuk, hogy egy mezőnek hány szomszédja lehessen (magyarán milyen alakzatú legyen maga a mező). A default érték 4, a max pedig 6, ezek között nyilakkal lehet mozogni, a bevitt értéket, pedig automatikusan eltárolja. Ha elérte a minimumot, vagy maximumot, akkor a gomb szürkére vált, és nemkattinthatóvá válik. A *Back* gombbal az előző menübe térünk vissza.

Ha a *Players*re megyünk, akkor megjelenik a már előbbinél látott nézet. Itt meg lehet adni, hogy maximum hány játékos csatlakozhasson be a játékba és ezek alapján fogja a játék indítását követően legenerálni őket. A default érték 2 (tehát legalább egy ellenfél lehessen), a max pedig 4, ezek között nyilakkal lehet mozogni, a bevitt értéket, pedig automatikusan eltárolja. A *Back* gombbal az előző menübe térünk vissza.

11.1.5 Játék vége

A játék négy különböző kimenetellel végződhet.

Első eset, amikor a karaktert megölik egy fejszével és úgy ér véget a játék neki. Azonban itt az OK gomb lenyomását követően a többieknek még folytatódik a játék, így automatikusan a soron lévő következő játékost tölti be a játék.

A második eset, hogy medvetánccal fertőződik meg, ebben az esetben az adott játékos már többet nem tud játszani, de az előzővel ellentétben, itt a játékos karaktere még játékban marad, „zombi játékos” lesz belőle.

A harmadik, hogy egy játékos sikeresen összegyűjti az összes genetikai kódot. Ebben az esetben kiíródik a képernyőre, hogy x megnyerte a játékot, így a játék mindenki számára véget ér. Ilyenkor ténylegesen vége a játéknak, és az *OK* gombot felváltja a *Back to menu*, hiszen már nincs mivel játszani.

Ez abban az esetben lép fel, ha mindenki meghal, vagy az utolsó játékos is medvetánccal fertőzött lesz. Itt is hasonlóan az előző verzióban már nem *OK* gomb van.

11.2 A grafikus rendszer architektúrája

11.2.1 A felület működési elve

A játék maga egy ablakban fog végig működni a menütől az utolsó percéig a játéknak. Esetlegesen lesznek olyan panelek, amelyek választ, vagy valamilyen fajta reagálást válnak a felhasználótól (pl. mikor megtámadsz valakit, akkor a játék átvált az ő szemszögére, és akkor kap egy ilyen üzenetet, hogy megtámadtak, viszont van nálad egy köpeny, el szeretnéd használni. Ebben az esetben a felhasználónak az igen/nem válaszgombok között kell majd visszareagálnia. Illetve, ha meghívjuk a karakternézőt, akkor az egy előreugró ablakként fog működni, így azt be kell zárni a rajta levő x gomb segítségével). Az egész játékot két módon is bele lehet zárni. A menüben található *Quit Game* segítségével is, illetve az egész játék ablakán található x-gombbal. A játék maga meg van rajzolva, ahogy a háttér is, viszont átlátszó gombokat fogunk helyezni rájuk, amelyek segítik majd a játékost az interakciók tényleges végrehajtásában, így technikailag javarészt mindenhova átlátszó gombokat fogunk elhelyezni a játék szebb megjelenítése érdekében.

11.2.2 A felület osztály struktúrája

11.2.2.1 GUI osztálydiagrammja

11.2.2.2 Eredeti osztálydiagram bővítései

11.3 A grafikus objektumok felsorolása

11.3.1 Game

- **Attribútumok:**
  - **PropertyChangeSupport support**: az osztályon belül történt változások esetén értesítést küld a grafikus interface elemeinek, melyeknek frissíteniük kell a megjelenített adatait. ***Láthatóság***: -, ***Modifier***: static
  - **java.util.Timer roundTimer:** Azért felel, hogy percenként léptesse az aktív játékost. ***Láthatóság***: -, ***Modifier***: static
  - **NextPlayerTask nextPlayerTask:** Ütemezhető taszk, mely run metódusában az idő lejárta után meghívjuk a next parancsot. ***Láthatóság****: -*, ***Modifier****:* static
- **Metódusok:**
  - **void startGame(int neighborCount, int virologistCount)**: A settingsben beállított értékek alapján létrehozzuk a játék mezőit (ügyelve arra, hogy a játék megnyerhető legyen, vagyis minden fajta genetikai kód létezzen valamely labor falára vésve), azokat eltároljuk a Game.fields listában, beállítjuk a szomszédos mezőket, Létrehozzuk a virológusokat (véletlen mezőre állítva őket), azokat hozzáadjuk a Game.virologists listához. A Game.inProgress változót true-ra állítjuk, a roundTimer-re létrehozunk egy új példányt, majd a next parancs meghívásával elindítjuk a játékot. ***Láthatóság:*** *+*, ***Modifier****: static*
    void next(): Az eddigiekben is használt módszerrel beállítjuk a következő játékost. Amennyiben egy játékos cselekvésképtelen, akkor megfelelő üzenet társításával értesítjük róla a gui feliratkozott objektumait. Pl. support.firePropertyChange(„info”, null, „V1 under Paralysis, skip step.”), vagy support.firePropertyChange(„info”, null, „V1 under VitusDance, skip step.”) stb. Amikor találunk egy virológust, aki cselekvésképes, akkor pedig értesítjük róla az erre a változásra feliratkozott gui elemeket a következő metódushívásokkal: support.firePropertyChange(„activeVirologistChange”, null, Game.getActiveVirologist()), illetve a support.firePropertyChange(„fieldChanged”, null, Game.getActiveVirologist().getStandingField()). Korábban ezután a next parancsban busy wait-el dolgoztunk, mely a console-ról várt bemenetet, timeout-al, most azonban mivel már nem a console-ról olvasunk, ez a módszer nem megfelelő. Ehelyett a roundTimer és nextPlayerTask változót használjuk. Ha a nextPlayerTask nem null, akkor meghívjuk annak cancel() metódusát, valamint a roundTimer.purge() metódusát. Ez gondoskodik arról, hogy ha az idő lejárta előtt átadjuk a vezérlést, akkor már ne váltson aktív játékost az előző játékos idejének lejárta után. Ez után új példányt hozunk létre a nextPlayerTask változóba. Meghívjuk az általunk írt Timer singleton osztály startRound() metódusát, és a timer.schedule(nextPlayerTask, 10000) metódushívással újraütemezzük a nextPlayerTask által tárolt objektumot. ***Láthatóság:*** *+*, ***Modifier****: static*
  - **void endGame()**: Az összes virológuson meghívja a die függvényt, a Game inProgress változóját false-ra állítja, meghívja a roundTimer cancel metódusát, majd a support.firePropertyChange(„endGame”, null, v) metódushívással értesíti a rá feliratkozott objektumokat. ***Láthatóság:*** *+*, ***Modifier****: static*
  - **Timer getRoundTimer():** A roundTimer attribútum gettere. ***Láthatóság:*** *+*, ***Modifier****: static*
  - **void setnextPlayerTask(NextPlayerTask newTask):** A nextPlayerTask attribútum settere. ***Láthatóság:*** *+*, ***Modifier****: static*

- **NextPlayerTask getNextPlayerTask():** A nextPlayerTask attribútum gettere. ***Láthatóság:*** *+*, ***Modifier****: static*
- **void addPropertyChangeListener(PropertyChangeListener pcl)**: feliratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követésére. ***Láthatóság:*** *+*, ***Modifier****: static*
- **void removePropertyChangeListener(PropertyChangeListener pcl)**: leiratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követéséről. ***Láthatóság:*** *+*, ***Modifier****: static*

11.3.2 NextPlayerTask

- **Felelősség:**
  - Adott idő eltelte után meghívja a Game.next() parancsot, majd újraütemezi önmagát
- **Ősosztályok:**
  - java.util.TimerTask
- **Metódusok:**
  - **void run():** Meghívja a Game.next() parancsot.

11.3.3 Virologist

- **Attribútumok:**
  - **PropertyChangeSupport support**: az osztályon belül történt változások esetén értesítést küld a grafikus interface elemeinek, melyeknek frissíteniük kell a megjelenített adatait. ***Láthatóság***: -
- **Metódusok:**
  - **void equipGloves ()**: Ha bármi miatt meghiúsul a kesztyű felvétele (pl. túl sok eszköz van rajtunk), akkor support.firePropertyChange(„error”, null, errorMessage) metódushívással értesítjük a rá feliratkozott gui objektumokat. Egyébként, ha az inventoryból kivett kesztyű nem null, akkor a virológus wornGloves listájához való hozzáadás után support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság****: +*
  - **void equipSack ()**: Ha bármi miatt meghiúsul a zsák felvétele (pl. túl sok eszköz van rajtunk), akkor support.firePropertyChange(„error”, null, errorMessage) metódushívással értesítjük a rá feliratkozott gui objektumokat. Egyébként, ha az inventoryból kivett zsák nem null, akkor a virológus wornSack listájához való hozzáadás, és az inventory maxMaterial növelése után
    - support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság****: +*
  - **void equipCloak():**Ha bármi miatt meghiúsul a köpeny felvétele (pl. túl sok eszköz van rajtunk), akkor support.firePropertyChange(„error”, null, errorMessage) metódushívással értesítjük a rá feliratkozott gui objektumokat. Egyébként, ha az inventoryból kivett köpeny nem null, akkor a virológus wornCloak listájához való hozzáadás után support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság****: +*
  - **void equipAxe():**Ha bármi miatt meghiúsul a fejsze felvétele (pl. túl sok eszköz van rajtunk), akkor support.firePropertyChange(„error”, null, errorMessage) metódushívással értesítjük a rá feliratkozott gui objektumokat. Egyébként, ha az

inventoryból kivett fejsze nem null, akkor a virológus wornAxe listájához való hozzáadás után support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság**: +*

- **void takeOffGloves():** Ha volt eszköz, amit levehettünk, vagyis az if(wornGloves.size()>0) részen belül, minden művelet elvégzése után a support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság**: +*
- **void takeOffSack(): Bugfix**: Ha a zsák nem az inventory-ba kerül, de le tudjuk venni magunkról, akkor is csökkenteni kell az inventory.maxMaterial értékét. **Egyéb módosítás**: Ha volt eszköz, amit levehettünk, vagyis az if(wornSack.size()>0) részen belül, minden művelet elvégzése után a support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság**: +*
- **void takeOffCloak():**Ha volt eszköz, amit levehettünk, vagyis az if(wornCloak.size()>0) részen belül, minden művelet elvégzése után a support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság**: +*
- **void takeOffAxe():**Ha volt eszköz, amit levehettünk, vagyis az if(wornAxe.size()>0) részen belül, minden művelet elvégzése után a
    - support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság**: +*
    - **boolean assaulted(Virologist vTo, Agent a):** Ha a virológus valamilyen ágens hatása alatt van, akkor support.firePropertyChange(„error”, null, errorMessage) metódushívással értesítjük a rá feliratkozott gui objektumokat. Ha nem sikerült a védekezés, akkor a support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság**: -*
- **void createAgent(GCode gc):** Ha az ágenst sikeresen hozzáadtuk a felhasználható ágensek listájába, akkor support.firePropertyChange(„skills”, null, agents) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság**: +*
- **void setInventory():** *miután beállítottuk az inventory-t,* support.firePropertyChange(„inventory”, null, getInventory()) metódushívással értesítjük a gui feliratkozott példányait. ***Láthatóság**: +*
- **void die():** *miután elvégeztük a szükséges műveleteket,* support.firePropertyChange(„info”, null,getID() + „ died.”) metódushívással értesítjük a gui feliratkozott példányait. ***Láthatóság**: +*
- **boolean wannaDrop()**: Már nem console-on olvassuk be azt, hogy a felhasználó az inventory-ba szeretne-e tenni egy eszközt vagy eldobni, hanem egy dialogwindow-ot jelenítünk meg rá, melynek time limit-et állítunk be. A time limit-nek a
    - Timer.getRemainingTime() metódus által visszaadott értéket adjuk meg. Ha az idő lejár, a DialogWindow-ot az X gombbal bezárjuk, vagy a No gombra kattintunk, akkor a válasz nem, és a metódus false-al tér vissza. Ha a Yes gombra kattintunk, akkor a válasz igen, és a metódus true-val tér vissza. ***Láthatóság**: +*
- **boolean wannaSteal(Virologist vTo)**: Ez a metódus egy duplikáció volt, töröljük, és mindenhol, ahol ezt hívtuk, a virológus standingField-jének a wannaSteal metódusát hívjuk helyette.
- **void move(Field f, boolean bEmittedByUser):** Miután a virológust eltávolítottuk a régi mezőjéről, majd hozzáadtuk az új mezőhöz, de még az arrive meghívása előtt a support.firePropertyChange(„fieldChanged”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság**: +*

- **String chooseObjectToSteal():** Már nem console-on olvassuk be azt, hogy a felhasználó mit szeretne lopni az adott virológustól, hanem egy dialogwindow-ot jelenítünk meg rá, melynek time limit-et állítunk be. A time limit-nek a Timer.getRemainingTime() metódus által visszaadott értéket adjuk meg. A
  - DialogWindow-on van egy JComboBox, melyben ki lehet választani az ellopni kívánt Equipment-et vagy Material-t. Ha a DialogWindow-ot az X gombbal bezárjuk, akkor a metódus üres string-el tér vissza. Ha az idő lejár, vagy a DialogWindow-on az ok gombra kattintunk, akkor a ComboBox-ban kiválasztott objektumnévvel tér vissza a metódus. ***Láthatóság**: -*
- **boolean defense(Virologist vFr, Agent a):** Ha eltávolítunk egy viselt kesztyűt, vagy köpenyt, akkor a support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. Ha a Game.getActiveVirologist() nem egyenlő a this által mutatott objektummal, akkor az Axe.wannaUse és a Gloves.wannaUse előtt meghívjuk a support.firePropertyChange(„activeVirologistChange”, null, this) metódust, amivel azt jelezzük, hogy ennél a DialogWindow-nál másik játékos kell, hogy válaszoljon. ***Láthatóság:*** *+*
- **void addPropertyChangeListener(PropertyChangeListener pcl)**: feliratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követésére. ***Láthatóság:*** *+*
- **void removePropertyChangeListener(PropertyChangeListener pcl)**: leiratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követéséről. ***Láthatóság:*** *+*

11.3.4 Timer

- **Attribútumok:**
  - **java.util.concurrent.atomic.AtomicLong startTime**: az az időpont, amikor az aktív virológus sorra került ***Láthatóság:*** *-*, ***Modifier****: static*
  - **ava.util.concurrent.atomic.AtomicLong enemyStartTime**: az az időpont, amikor átadtuk a vezérlést rövidebb időre egy másik játékosnak
- **Metódusok:**
  - **void startRound()**: A funkcionalitás nem változik, csak átalakítjuk a módosított atomic változó használatához. ***Láthatóság:*** *+*, ***Modifier****: static*
  - **void startEnemyRound()**: A funkcionalitás nem változik, csak átalakítjuk a módosított atomic változó használatához. ***Láthatóság:*** *+*, ***Modifier****: static*
  - **long getEnemyRemainingTime()**: A funkcionalitás nem változik, csak átalakítjuk a módosított atomic változó használatához. ***Láthatóság:*** *+*, ***Modifier****: static*
  - **long timeSpentByEnemy():** A funkcionalitás nem változik, csak átalakítjuk a módosított atomic változó használatához. ***Láthatóság:*** *+*, ***Modifier****: static*
  - **long getRemainingTime():**A funkcionalitás nem változik, csak átalakítjuk a módosított atomic változó használatához. ***Láthatóság:*** *+*, ***Modifier****: static*
  - **void increaseRemainingTime(long time)**: Nem csak a remainingTime-ot növeljük meg a bemeneti paraméter értékével, hanem elkérjük a Game nextPlayerTask attribútumát, és meghívjuk annak cancel() metódusát. Ez után a Game roundTimer objektumát is elkérjük, és meghívjuk annak purge() metódusát. Végül új példányt hozunk létre a nextPlayerTask változóba, és a timer.schedule(nextPlayerTask, getRemainingTime()) metódushívással újraütemezzük a nextPlayerTask által tárolt objektumot. ***Láthatóság:*** *+*, ***Modifier****: static*

11.3.5 Inventory

- **Attribútumok:**
  - **PropertyChangeSupport support**: az osztályon belül történt változások esetén értesítést küld a grafikus interface elemeinek, melyeknek frissíteniük kell a megjelenített adatait. **Láthatóság**: -
- **Metódusok:**
  - **void setMaxMaterial(int maxMaterial)**: Felveszünk két új belső változót, melyekben eltároljuk, hogy mennyi aminoacid, illetve mennyi nukleotid került eldobásra, és ha bármelyik nagyobb, mint 0, akkor értesítjük az ezen osztály változásaira feliratkozott gui elemeket a support.firePropertyChange(„info”, null, msgInfo), illetve a support.firePropertyChange(„inventory”, null, this) metódushívásokkal. ***Láthatóság:*** +
  - **void addGloves(Gloves g):** Ha nem sikerül a kesztyű hozzáadása az inventory-hoz, akkor support.firePropertyChange(„error”, null, errInfo) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket, ha sikerül, akkor pedig a support.firePropertyChange(„inventory”, null, this) metódust hívjuk. ***Láthatóság:*** + **void addSack(Sack s):** Ha nem sikerül a zsák hozzáadása az inventory-hoz, akkor support.firePropertyChange(„error”, null, errInfo) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket, ha sikerül, akkor pedig a support.firePropertyChange(„inventory”, null, this) metódust hívjuk. ***Láthatóság:*** +
  - **void addCloak(Cloak c):** Ha nem sikerül a köpeny hozzáadása az inventory-hoz, akkor support.firePropertyChange(„error”, null, errInfo) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket, ha sikerül, akkor pedig a support.firePropertyChange(„inventory”, null, this) metódust hívjuk. ***Láthatóság:*** +
  - **void addAxe(Axe a):** Ha nem sikerül a fejsze hozzáadása az inventory-hoz, akkor support.firePropertyChange(„error”, null, errInfo) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket, ha sikerül, akkor pedig a support.firePropertyChange(„inventory”, null, this) metódust hívjuk. ***Láthatóság:*** +
  - **Gloves removeGloves():** Ha sikerül eltávolítani a kesztyűt, akkor a support.firePropertyChange(„inventory”, null, this) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket. . ***Láthatóság:*** +
  - **Sack removeSack():**Ha sikerül eltávolítani a zsákot, akkor a support.firePropertyChange(„inventory”, null, this) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket. . ***Láthatóság:*** +
  - **Cloak removeCloak()**: Ha sikerül eltávolítani a köpenyt, akkor a support.firePropertyChange(„inventory”, null, this) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket. . ***Láthatóság:*** +
  - **Axe removeAxe():** Ha sikerül eltávolítani a fejszét, akkor a support.firePropertyChange(„inventory”, null, this) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket. . ***Láthatóság:*** +
  - **int learnGCode(GCode g):** Ha új kódot tanultunk, akkor support.firePropertyChange(„inventory”, null, this) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket. . ***Láthatóság:*** +
  - **void forgetGCodes():** Az ismert genetikai kódok listájának kiürítése után support.firePropertyChange(„inventory”, null, this) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket. . ***Láthatóság:*** +
  - **void addAmino(Aminoacid am)**: Ha nem sikerül az Aminoacid hozzáadása az inventory-hoz, akkor support.firePropertyChange(„error”, null, errInfo)

metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket, ha sikerül, akkor pedig a support.firePropertyChange(„inventory”, null, this) metódust hívjuk. ***Láthatóság:*** +

- **void addNukleo(Nukleotid n)**: Ha nem sikerül a Nukleotid hozzáadása az inventoryhoz, akkor support.firePropertyChange(„error”, null, errInfo) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket, ha sikerül, akkor pedig a support.firePropertyChange(„inventory”, null, this) metódust hívjuk. ***Láthatóság:*** +
- **Aminoacid removeAmino():** Ha sikerül eltávolítani az Aminoacidot, akkor a support.firePropertyChange(„inventory”, null, this) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket. . ***Láthatóság:*** +
- **Nukleotid removeNukleo():** Ha sikerül eltávolítani a Nukleotidot, akkor a support.firePropertyChange(„inventory”, null, this) metódushívással értesítjük az ezen osztály változásaira feliratkozott gui elemeket. . ***Láthatóság:*** +
- **void addPropertyChangeListener(PropertyChangeListener pcl)**: feliratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követésére. ***Láthatóság:*** +
- **void removePropertyChangeListener(PropertyChangeListener pcl)**: leiratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követéséről. ***Láthatóság:*** +

11.3.6 BearDance

- **Attribútumok:**
  - **PropertyChangeSupport support**: az osztályon belül történt változások esetén értesítést küld a grafikus interface elemeinek, melyeknek frissíteniük kell a megjelenített adatait. ***Láthatóság:*** -
- **Metódusok:**
  - **void setVirologistUnderEffect(Virologist virologist)**: A metódus végén meghívjuk a support.firePropertyChange(„virologist”, null, virologist) -t, illetve a support.firePropertyChange(„gameOverByBearDance”, null, virologist) -t. ***Láthatóság:*** +
  - **void addPropertyChangeListener(PropertyChangeListener pcl)**: feliratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követésére. ***Láthatóság:*** +
  - **void removePropertyChangeListener(PropertyChangeListener pcl)**: leiratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követéséről. ***Láthatóság:*** +

11.3.7 Immunity

- **Attribútumok:**
  - **PropertyChangeSupport support**: az osztályon belül történt változások esetén értesítést küld a grafikus interface elemeinek, melyeknek frissíteniük kell a megjelenített adatait. ***Láthatóság:*** -
- **Metódusok:**
  - **void setVirologistUnderEffect(Virologist virologist)**: A metódus végén meghívjuk a support.firePropertyChange(„virologist”, null, virologist) -t. ***Láthatóság:*** +
  - **void addPropertyChangeListener(PropertyChangeListener pcl)**: feliratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követésére. ***Láthatóság:*** +
  - **void removePropertyChangeListener(PropertyChangeListener pcl)**: leiratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követéséről. ***Láthatóság:*** +

11.3.8 Oblivion

- **Attribútumok:**
  - **PropertyChangeSupport support**: az osztályon belül történt változások esetén értesítést küld a grafikus interface elemeinek, melyeknek frissíteniük kell a megjelenített adatait. ***Láthatóság:*** -
- **Metódusok:**
  - **void setVirologistUnderEffect(Virologist virologist):** A metódus végén meghívjuk a support.firePropertyChange(„virologist”, null, virologist) -t. ***Láthatóság:*** +
  - **void addPropertyChangeListener(PropertyChangeListener pcl)**: feliratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követésére. ***Láthatóság:*** +
  - **void removePropertyChangeListener(PropertyChangeListener pcl)**: leiratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követéséről. ***Láthatóság:*** +

11.3.9 Paralysis

- **Attribútumok:**
  - **PropertyChangeSupport support**: az osztályon belül történt változások esetén értesítést küld a grafikus interface elemeinek, melyeknek frissíteniük kell a megjelenített adatait. ***Láthatóság:*** -
- **Metódusok:**
  - **void setVirologistUnderEffect(Virologist virologist)**: A metódus végén meghívjuk a support.firePropertyChange(„virologist”, null, virologist) -t. ***Láthatóság:*** +
  - **void addPropertyChangeListener(PropertyChangeListener pcl)**: feliratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követésére. ***Láthatóság:*** +
  - **void removePropertyChangeListener(PropertyChangeListener pcl)**: leiratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követéséről. ***Láthatóság:*** +

11.3.10 Gloves

- **Metódusok:**
  - **boolean wannaUse():** Már nem console-on olvassuk be azt, hogy a felhasználó akarja-e használni a kesztyűt, hanem egy dialogwindow-ot jelenítünk meg rá, melynek time limit-et állítunk be. A time limit-nek a Timer.getRemainingTime() metódus által visszaadott értéket adjuk meg. Ha az idő lejár, a DialogWindow-ot az X gombbal bezárjuk, vagy a No gombra kattintunk, akkor a válasz nem, és a metódus false-al tér vissza. Ha a Yes gombra kattintunk, akkor a válasz igen, és a metódus true-val tér vissza. ***Láthatóság**: +*

11.3.11 Axe

- **Attribútumok:**
  - **PropertyChangeSupport support**: az osztályon belül történt változások esetén értesítést küld a grafikus interface elemeinek, melyeknek frissíteniük kell a megjelenített adatait. ***Láthatóság**: -*
- **Metódusok:**
  - **boolean wannaUse():** Már nem console-on olvassuk be azt, hogy a felhasználó akarja-e használni a fejszét, hanem egy dialogwindow-ot jelenítünk meg rá, melynek time limit-et állítunk be. A time limit-nek a Timer.getRemainingTime() metódus által visszaadott értéket adjuk meg. Ha az idő lejár, a DialogWindow-ot az X gombbal bezárjuk, vagy a No gombra kattintunk, akkor a válasz nem, és a metódus false-al tér vissza. Ha a Yes gombra kattintunk, akkor a válasz igen, és a metódus true-val tér vissza. ***Láthatóság**: +*
  - **void use(Virologist v):** A metódus végén meghívjuk a support.firePropertyChange(„virologist”, null, v) -t. ***Láthatóság**: +*
  - **void addPropertyChangeListener(PropertyChangeListener pcl)**: feliratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követésére. ***Láthatóság:** +*
  - **void removePropertyChangeListener(PropertyChangeListener pcl)**: leiratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követéséről. ***Láthatóság:** +*

11.3.12 GCode

- **Attribútumok:**
  - **PropertyChangeSupport support**: az osztályon belül történt változások esetén értesítést küld a grafikus interface elemeinek, melyeknek frissíteniük kell a megjelenített adatait. ***Láthatóság**: -*
- **Metódusok:**
  - **public Agent create (Virologist v):** Ha az ágens létrehozása nyersanyaghiány (Material) miatt meghiúsul, akkor support.firePropertyChange(„error”, null, errorMessage) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság:** +*
  - **void addPropertyChangeListener(PropertyChangeListener pcl)**: feliratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követésére. ***Láthatóság:** +*

- **void removePropertyChangeListener(PropertyChangeListener pcl)**: leiratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követéséről. ***Láthatóság:*** +

11.3.13 Field

- **Metódusok:**
  - **boolean wannaSteal(Virologist v):** Már nem console-on olvassuk be azt, hogy a felhasználó akar-e lopni, hanem egy dialogwindow-ot jelenítünk meg rá, melynek time limit-et állítunk be. A time limit-nek a Timer.getRemainingTime() metódus által visszaadott értéket adjuk meg. Ha az idő lejár, a DialogWindow-ot az X gombbal bezárjuk, vagy a No gombra kattintunk, akkor a válasz nem, és a metódus false-al tér vissza. Ha a Yes gombra kattintunk, akkor a válasz igen, és a metódus true-val tér vissza. ***Láthatóság:*** +

11.3.14 Lab

- **Metódusok:**
  - **boolean wannaTouchy():** Már nem console-on olvassuk be azt, hogy a felhasználó akar-e kódot letapogatni, hanem egy dialogwindow-ot jelenítünk meg rá, melynek time limit-et állítunk be. A time limit-nek a Timer.getRemainingTime() metódus által visszaadott értéket adjuk meg. Ha az idő lejár, a DialogWindow-ot az X gombbal bezárjuk, vagy a No gombra kattintunk, akkor a válasz nem, és a metódus false-al tér vissza. Ha a Yes gombra kattintunk, akkor a válasz igen, és a metódus true-val tér vissza. ***Láthatóság:*** +

11.3.15 InfectedLab

- **Attribútumok:**
  - **PropertyChangeSupport support**: az osztályon belül történt változások esetén értesítést küld a grafikus interface elemeinek, melyeknek frissíteniük kell a megjelenített adatait. ***Láthatóság:*** -

- **Metódusok:**
  - **boolean arrive(Virologist v):** Ha a virológusról eltávolítunk egy köpenyt, akkor a support.firePropertyChange(„virologist”, null, this) metódushívással értesítjük a rá feliratkozott gui objektumokat. ***Láthatóság:*** +
  - **void addPropertyChangeListener(PropertyChangeListener pcl)**: feliratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követésére. ***Láthatóság:*** +
  - **void removePropertyChangeListener(PropertyChangeListener pcl)**: leiratkoztat egy PropertyChangeListener interface-t implementáló objektum példányát az osztályban történt módosítások követéséről. ***Láthatóság:*** +

11.3.16 Storage

- **Metódusok:**
  - **boolean wannaPickup():** Már nem console-on olvassuk be azt, hogy a felhasználó akar-e nyersanyagot (Material) felvenni, hanem egy dialogwindow-ot jelenítünk meg rá, melynek time limit-et állítunk be. A time limit-nek a Timer.getRemainingTime() metódus által visszaadott értéket adjuk meg. Ha az idő lejár, a DialogWindow-ot az X gombbal bezárjuk, vagy a No gombra kattintunk, akkor a válasz nem, és a metódus false-al tér vissza. Ha a Yes gombra kattintunk, akkor a válasz igen, és a metódus trueval tér vissza. ***Láthatóság**: +*

11.3.17 Shelter

- **Metódusok:**
  - **boolean wannaPickup():** Már nem console-on olvassuk be azt, hogy a felhasználó akar-e eszközt (Equipment) felvenni, hanem egy dialogwindow-ot jelenítünk meg rá, melynek time limit-et állítunk be. A time limit-nek a Timer.getRemainingTime() metódus által visszaadott értéket adjuk meg. Ha az idő lejár, a DialogWindow-ot az X gombbal bezárjuk, vagy a No gombra kattintunk, akkor a válasz nem, és a metódus false-al tér vissza. Ha a Yes gombra kattintunk, akkor a válasz igen, és a metódus trueval tér vissza. ***Láthatóság**: +*

11.4 Kapcsolat az alkalmazói rendszerrel

11.4.1 GameOver

11.4.2 Character

11.4.3 Inventory

11.5 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	Litavecz
máj. 2.	Grafikus változat tervei – beadás Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése	30	Ruskó
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	

11.6 Napló

Kezdet	Időtartam	Résztvevő(k)	Leírás
2022.04.11 17:40	1 óra	Alpek, Csia, Ruskó	Megbeszélés: Grafikus rész alapötletek megvitatása.
2022.04.11 – 05.02	Kb. 30 óra	Csia	Feladat: Elemek, hátterek, karakterek, objektumok egyedileg megtervezése, Pixelart nevű program segítségével megrajzolásuk.
2022.05.01 11:20	1,5 óra	Csia, Ruskó	Megbeszélés: Osztálydiagram alapvető átbeszélése, alapvető ötletek egyeztetése a grafikai kinézettel kapcsolatban.
2022.05.01 13:40	2 óra 20 perc	Csia, Ruskó	Megbeszélés: További ötletek, problémák megvitatása a grafikus kinézettel kapcsolatban, osztálydiagram és állapotgépek megrajzolása a megbeszéltek szerint.
2022.05.01 16:00	3 óra	Ruskó	Feladat: Osztálydiagram leírások elkészítése.
2022.05.01 17:20	1,5 óra	Csia	Feladat: Grafikus interfész szekció összeállítása, szövegezések megírása.
2022.05.02. 17:00	6 óra	Alpek, Csia, Marton, Ruskó	Megbeszélés: Osztálydiagram teljes átszervezése, feladatok szétosztása.
2022.05.03 06:00	1 óra	Alpek	Feladat: Menü és játék vége skiccek elkészítése.
2022.0503 11:00	2 óra	Csia	Feladat: Menü és játék vége részekhez szövegezés elkészítése, A felület működési elv rész megírása.
2022.05.03 15:00	8 óra	Marton	Feladat: Osztálydiagrammok leírásainak elkészítése.
2022.05.04 01:30	3 óra	Litavecz	Feladat: Szekvenciadiagrammok megrajzolása.
2022.05.04 09:00	40 perc	Csia	Feladat: Dokumentum véglegesítése összegzése

11.7 Százalékos teljesítés

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

13. Prototípus beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
CharacterPanel.java	11 KB	2022.05.12 10:30	Ez az osztály felelős a karakter panel megjelenítésért, a karakteren történő eszközök megjelenítése.
ChooseQuestion.java	2 KB	2022.05.12 10:30	Választási lehetőség megvalósításáért felelős osztály.
CIView.java	2 KB	2022.05.12 10:30	A karakternézet és az inventory nálunk egy közös panelen jelenik meg, melyet ez az osztály jeleníti meg.
EndGame.java	2 KB	2022.05.12 10:30	A játék véget ér valamelyik játékos számára halál vagy fertőzés által, akkor ez az osztály jelez neki valamilyen módon.
GamePanel.java	14 KB	2022.05.12 10:30	Az főosztály panelje, ez felel a játék indítását követően a játék megjelenítésére.
InventoryPanel.java	8 KB	2022.05.12 10:30	Ez az osztály felelős az inventory panel megjelenítésért, az inventoryban történő változások eszközöléséért.
MainWindow.java	5 KB	2022.05.12 10:30	A főablak minden gui-s osztálynak.
MenuPanel.java	3 KB	2022.05.12 10:30	A játék indításakor egy menü tölt be, melyet ez az osztály jelenít meg.
NumberSettingDialog.java	1 KB	2022.05.12 10:30	A menüben van egy beállítások rész, ahol lehet kezelni a játékosok számát és a szomszédokét, ez hozzá a non-clickable beviteli mező.
NumberSettingsPanel.java	4 KB	2022.05.12 10:30	A fenti bevitelimezőhöz ez szolgáltatja a számok között váltogatható gombokat.
SettingsPanel.java	3 KB	2022.05.12 10:30	A beállítás menürészért felelős osztály.
SkillPanel.java	9 KB	2022.05.12 10:30	A skillsávért felelős osztály.

TextPanel.java	2 KB	2022.05.12 10:30	A játékban található egy üzenő rész is, amely a játékos tájékoztatásáért felel.
WinGame.java	2 KB	2022.05.12 10:30	Ez az osztály akkor indul el, amikor a játékot megnyeri valaki.
YesNoQuestion.java	3 KB	2022.05.12 10:30	Ha eldöntendő kérdést kap a játékos, akkor ez az osztály felel érte.
Agent.java	3 KB	2022.03.25 17:40	Az ágensek absztrakt főosztálya.
Aminoacid.java	1 KB	2022.03.25 17:40	Az aminosav anyagot lekezelő osztály.
Axe.java	3 KB	2022.04.24 12:00	A fejsze eszközt lekezelő osztály.
BearDance.java	4 KB	2022.04.24 12:00	A medvetáncot okozó ágenst lekezelő osztály.
Cloak.java	3 KB	2022.03.25 17:40	A köpeny eszközt lekezelő osztály.
Equipment.java	1 KB	2022.03.25 17:40	Az eszközök absztrakt főosztálya.
Field.java	4 KB	2022.03.25 17:40	A mezők absztrakt főosztálya, illetve a sima (üres) mező osztálya.
Game.java	26 KB	2022.03.25 17:40	A játék indításáért és leállításáért felelős osztály.
GCode.java	4 KB	2022.03.25 17:40	A genetikai kódot és az ágensgenerálást lekezelő osztály.
Gloves.java	3 KB	2022.03.25 17:40	A kesztyű eszközt lekezelő osztály.
Immunity.java	2 KB	2022.03.25 17:40	A sérthetetlenséget okozó ágenst lekezelő osztály.
InfectedLab.java	3 KB	2022.04.24 12:00	A fertőző labor típusú mezőt lekezelő osztály.
Inventory.java	11 KB	2022.03.25 17:40	A játékosnál lévő alap tárolót és a hozzá tartozó akciókat lekezelő osztály.
Lab.java	3 KB	2022.03.25 17:40	A labor típusú mezőt lekezelő osztály.
Material.java	1 KB	2022.03.25 17:40	Az anyagok absztrakt főosztálya.

NextPlayerTask.java	1 KB	2022.05.17 13:20	A következő játékosra való átléptetés.
Nukleotid.java	1 KB	2022.03.25 17:40	A nukleotid anyagot lekezelő osztály.
Oblivion.java	2 KB	2022.03.25 17:40	A feledést okozó ágenst lekezelő osztály.
Paralysis.java	2 KB	2022.03.25 17:40	A bénulást okozó ágenst lekezelő osztály.
Program.java	1 KB	2022.03.25 17:40	A főosztály, ahonnan indul a program, itt találhatóak a szenáriók is.
Sack.java	1 KB	2022.03.25 17:40	A zsák eszközt lekezelő osztály.
Shelter.java	4 KB	2022.03.25 17:40	Az óvóhely típusú mezőt lekezelő osztály.
Steppable.java	1 KB	2022.04.24 12:00	A léptetőket kezelő osztály.
Storage.java	5 KB	2022.03.25 17:40	A raktár típusú mezőt lekezelő osztály.
Timer.java	2 KB	2022.04.24 22:00	Az időzítőket tárolja.
Virologist.java	22 KB	2022.03.25 17:40	A játékos karakterét és a hozzá tartozó akciókat lekezelő osztály.
VitusDance.java	4 KB	2022.03.25 17:40	A vitustáncot okozó ágenst lekezelő osztály.

13.1.2 Fordítás

A mappa letöltését és kicsomagolását követően találtok majd a legkülső mappában egy *SeeNoLight.exe* nevezetű file-t, arra kell rákattintani, és a program elindul magától.

13.1.3 Futtatás

A mappa letöltését és kicsomagolását követően találtok majd a legkülső mappában egy *SeeNoLight.exe* nevezetű file-t, arra kell rákattintani, és a program elindul magától.

13.2 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	Litavecz
máj. 2.	Grafikus változat tervei – beadás Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése	30	Ruskó
máj. 16.	Grafikus változat készítése	40	Marton
máj. 18.	Egységes dokumentáció - beadás és bemutatás Grafikus változat - beadás és a forráskód herculesre való feltöltése, és teljes házi bemutatás	30	Csia

13.3 Napló

Kezdet	Időtartam	Résztvevő(k)	Leírás
2022.05.06 17:00	1 óra	Teljes csapat	Megbeszélés: Az első körös feladatok szétosztása a csapattagok között. Újabb határidő kitűzése.
2022.05.09 15:00	2 óra	Alpek	Feladat: A menürendszer leprogramozása, a hozzátartozó *Settings* résszel együtt.
2022.05.09 11:00	1 óra	Csia	Feladat: A *GameOver* panelek elkészítése.
2022.05.09 14:00	1 óra	Litavecz	Feladat: A karakteres inventory panel elkészítése.
2022.05.09 16:00	4 óra	Marton	Feladat: A business logic leírás alapján a változtatások bevezetése a programba.
2022.05.07 18:00	5 óra	Ruskó	Feladat: A *GamePanel* elkészítése, objektumok és alap elhelyezkedések leprogramozása, váz felépítése.
2022.05.16	3 óra	Alpek	Feladat: A menürendszer leprogramozásának elkezdése.
2022.05.14-18	40 óra	Csia	Feladat: A maradék design rajz elkészítése beleszámítva a menüket, a játék végét, az összes fieldet, skillbart, karaktereket, illetve a program összeállításában besegítés.
2022.05.14-18.	12 óra	Litavecz	Feladat: A menürendszer és a karakternézet (beleszámítva az intventoryt is) teljes leprogramozása, a program összeállításába besegítés.
2022.05.14-18	60 óra	Marton	Feladat: A business logic beleépítése a játékba, a játék végleges verziójának összeállítása.
2022.05.14-17	20 óra	Ruskó	Feladat: A skillbar működésének elkészítése, alapvető gombelhelyezkedése beletétele.
2022.05.18. 10:00	1 óra	Csia	Feladat: Dokumentumok egységesítése.

13.4 Százalékos teljesítés

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