

Tartalom

2.	Követelmény, projekt, funkcionalitás	11
2.1	Bevezetés	11
2.1.1	Cél	11
2.1.2	Szakterület	11
2.1.3	Definíciók, rövidítések	11
2.1.4	Hivatkozások	11
2.1.5	Összefoglalás	11
2.2	Áttekintés	11
2.2.1	Általános áttekintés	11
2.2.2	Funkciók	11
2.2.3	Felhasználók	12
2.2.4	Korlátozások	13
2.2.5	Feltételezések, kapcsolatok	13
2.3	Követelmények	13
2.3.1	Funkcionális követelmények	13
2.3.2	Erőforrásokkal kapcsolatos követelmények	15
2.3.3	Átadással kapcsolatos követelmények	15
2.3.4	Egyéb nem funkcionális követelmények	16
2.4	Lényeges use-case-ek	16
2.4.1	Use-case leírások	16
2.4.2	Use-case diagram	19
2.5	Szótár	19
2.6	Projekt terv	20
2.7	Napló	22
3.	Analízis modell kidolgozása	25
3.1	Objektum katalógus	25
3.1.1	Anyag - Resource	25
3.1.2	Aminosav – AminoAcid	25
3.1.3	Amnéziavírus - AmnesiaVirus	25
3.1.4	Ágens - Agent	25
3.1.5	Bénítóvírus - StunVirus	25
3.1.6	Felszerelés - Inventory	25
3.1.7	Genetikai kód - GeneticCode	25
3.1.8	Hatás - Effect	25
3.1.9	Játék (kontroller) - Controller	25

3.1.10	Kesztyű - Glove.....	25
3.1.11	Laboratórium - Laboratory	25
3.1.12	Mező - Tile	26
3.1.13	Nukleotid - Nucleotid.....	26
3.1.14	Óvóhely - Safehouse	26
3.1.15	Pálya - Map	26
3.1.16	Raktár - Warehouse	26
3.1.17	Szabad terület - EmptyTile.....	26
3.1.18	Vakcina - Vaccine	26
3.1.19	Védőfelszerelés - Equipment	26
3.1.20	Védőköpeny - ProtectiveCloak	26
3.1.21	Virológus - Virologist	26
3.1.22	Vitustáncvírus - VitusDanceVirus	26
3.1.23	Zsák – Bag	26
3.2	Statikus struktúra diagramok - Osztály diagram	27
3.3	Osztályok leírása.....	28
3.3.1	Agent	28
3.3.2	AmnesiaVirus	29
3.3.3	Bag	29
3.3.4	Collectable.....	29
3.3.5	Controller	30
3.3.6	Effect	30
3.3.7	EmptyTile.....	31
3.3.8	Equipment	31
3.3.9	GeneticCode	32
3.3.10	Glove	33
3.3.11	Inventory	33
3.3.12	Laboratory	34
3.3.13	Map.....	35
3.3.14	ProtectiveCloak	35
3.3.15	Resource	36
3.3.16	ResourceType.....	36
3.3.17	Safehouse	36
3.3.18	StunVirus.....	37
3.3.19	Tile	37
3.3.20	Vaccine	38
3.3.21	Virologist.....	39

3.3.22	VitusDanceVirus	39
3.3.23	Warehouse	40
	Szekvencia diagramok.....	41
3.4	State-chartok	51
3.5	Napló	52
4.	Szkeleton tervezése	55
4.1	A szkeleton modell valóságos use-case-ei.....	55
4.1.1	Use-case diagram	55
4.1.2	Use-case leírások.....	56
4.2	A szkeleton kezelői felületének terve, dialógusok	62
4.2.1	Szkeleton program felépítése	62
4.2.2	Szkeleton program kimenete	62
4.2.3	Szkeleton program bemenete	62
4.3	Szekvencia diagramok a belső működésre	63
4.3.1	moveToEmptyTile	63
4.3.2	moveToLaboratory	63
4.3.3	moveToSafehouse	64
4.3.4	moveToWarehouse.....	64
4.3.5	pickUpBag.....	65
4.3.6	pickUpCloak.....	65
4.3.7	pickUpGlove	66
4.3.8	pickUpResource	66
4.3.9	learnGeneticCode	67
4.3.10	craftStunVirus	67
4.3.11	craftAmnesiaVirus	68
4.3.12	craftVitusDanceVirus.....	68
4.3.13	craftVaccine	68
4.3.14	useAmnesiaVirus	69
4.3.15	useStunVirus	69
4.3.16	useVaccine	70
4.3.17	useVitusDanceVirus.....	70
4.3.18	robVirologist	71
4.3.19	forgetCodesViaAmnesiaVirus	72
4.3.20	enlargeTheBag	73
4.3.21	stunnedPlayerMissesTurn	74
4.3.22	playerVitusDances	75
4.4	Kommunikációs diagramok.....	76

4.4.1	moveToEmptyTile	76
4.4.2	MoveToLaboratory	76
4.4.3	moveToSafehouse	77
4.4.4	moveToWarehouse.....	77
4.4.5	pickUpBag.....	78
4.4.6	pickUpCloak.....	78
4.4.7	pickUpGlove	79
4.4.8	pickUpResource	79
4.4.9	learnGeneticCode	80
4.4.10	craftStunVirus	80
4.4.11	craftAmnesiaVirus	81
4.4.12	craftVitusDanceVirus.....	81
4.4.13	craftVaccine	82
4.4.14	useAmnesiaVirus	82
4.4.15	useStunVirus	83
4.4.16	useVaccine	83
4.4.17	useVitusDanceVirus.....	84
4.4.18	robVirologist	84
4.4.19	forgetCodesViaAmnesiaVirus	85
4.4.20	enlargeTheBag	85
4.4.21	stunnedPlayerMissesTurn	86
4.4.22	playerVitusDances	86
4.5	Napló	87
5.	Szkeleton tervezése	90
5.1	A szkeleton modell valóságos use-case-ai.....	90
5.1.1	Use-case diagram	90
5.1.2	Use-case leírások.....	91
5.2	A szkeleton kezelői felületének terve, dialógusok	97
5.2.1	Szkeleton program felépítése	97
5.2.2	Szkeleton program kimenete	97
5.2.3	Szkeleton program bemenete	97
5.3	Szekvencia diagramok a belső működésre	98
5.3.1	moveToEmptyTile	98
5.3.2	moveToLaboratory	98
5.3.3	moveToSafehouse	99
5.3.4	moveToWarehouse.....	99
5.3.5	pickUpBag.....	100

5.3.6	pickUpCloak.....	100
5.3.7	pickUpGlove	101
5.3.8	pickUpResource	101
5.3.9	learnGeneticCode	102
5.3.10	craftStunVirus	102
5.3.11	craftAmnesiaVirus	103
5.3.12	craftVitusDanceVirus	103
5.3.13	craftVaccine	103
5.3.14	useAmnesiaVirus	104
5.3.15	useStunVirus	104
5.3.16	useVaccine	105
5.3.17	useVitusDanceVirus.....	105
5.3.18	robVirologist	106
5.3.19	forgetCodesViaAmnesiaVirus	107
5.3.20	enlargeTheBag	108
5.3.21	stunnedPlayerMissesTurn	109
5.3.22	playerVitusDances	110
5.4	Kommunikációs diagramok.....	111
5.4.1	moveToEmptyTile	111
5.4.2	MoveToLaboratory	112
5.4.3	moveToSafehouse	112
5.4.4	moveToWarehouse.....	113
5.4.5	pickUpBag.....	113
5.4.6	pickUpCloak.....	114
5.4.7	pickUpGlove	114
5.4.8	pickUpResource	115
5.4.9	learnGeneticCode	115
5.4.10	craftStunVirus	116
5.4.11	craftAmnesiaVirus	116
5.4.12	craftVitusDanceVirus.....	117
5.4.13	craftVaccine	117
5.4.14	useAmnesiaVirus	118
5.4.15	useStunVirus	118
5.4.16	useVaccine	119
5.4.17	useVitusDanceVirus.....	119
5.4.18	robVirologist	120
5.4.19	forgetCodesViaAmnesiaVirus	120

5.4.20	enlargeTheBag	121
5.4.21	stunnedPlayerMissesTurn	121
5.4.22	playerVitusDances	122
5.5	Napló	123
6.	Protótípus koncepciója	126
6.0	Változás hatása a modellre	126
6.0.1	Módosult osztálydiagram	126
6.0.2	Új vagy megváltozó metódusok	127
6.0.3	Szekvencia-diagramok	128
6.1	Protótípus interface-definíciója	133
6.1.1	Az interfész általános leírása	133
6.1.2	Bemeneti nyelv	133
6.1.3	Kimeneti nyelv	134
6.2	Összes részletes use-case	136
6.3	Tesztelési terv	138
6.4	Tesztelést támogató segéd- és fordítóprogramok specifikálása	140
6.5	Napló	141
7.	Részletes tervezet	143
7.1	Osztályok és metódusok tervezet	144
7.1.1	Agent	144
7.1.2	AmnesiaVirus	145
7.1.3	Axe	145
7.1.4	Bag	146
7.1.5	BearDanceVirus	146
7.1.6	ConsoleView	147
7.1.7	Controller	148
7.1.8	Effect	149
7.1.9	EmptyTile	150
7.1.10	Equipment	150
7.1.11	GeneticCode	151
7.1.12	Glove	152
7.1.13	InfectedLaboratory	152
7.1.14	Inventory	153
7.1.15	Laboratory	155
7.1.16	Map	155
7.1.17	ProtectiveCloak	156
7.1.18	Resource	156

7.1.19	Safehouse	158
7.1.20	StunVirus.....	158
7.1.21	Tile	159
7.1.22	UsableEquipment	160
7.1.23	Vaccine	160
7.1.24	View	161
7.1.25	Virologist.....	161
7.1.26	VitusDanceVirus	163
7.1.27	Warehouse.....	164
7.2	A tesztek részletes tervei, leírásuk a teszt nyelvén	165
7.2.1	Kesztyű használata	166
7.2.2	Balta használata.....	168
7.2.3	Védőköpeny hatása	169
7.2.4	Zsák hatása	171
7.2.5	Sikeres lopás.....	172
7.2.6	Sikertelen lopás	174
7.2.7	Medvevírus elkapása	175
7.2.8	Genetikai kód megtanulása és ágens készítése.....	176
7.2.9	Genetikai kód megtanulása és ágens készítése.....	177
7.2.10	Az utolsó genetikai kód megtanulása és a játék megnyerése.....	177
7.2.11	Amnéziavírus hatása	179
7.2.12	Vitustáncvírus alkalmazása és hatása.....	180
7.2.13	Vakcina alkalmazása és hatása.....	181
7.3	A tesztelést támogató programok tervei	182
7.4	Napló	183
8.	Prototípus beadása	185
8.1	Fordítási és futtatási útmutató.....	185
8.1.1	Fájllista	185
8.1.2	Fordítás	188
8.1.3	Futtatás	188
8.2	Tesztek jegyzőkönyvei	189
8.2.1	Kesztyű használata	189
8.2.2	Balta használata.....	189
8.2.3	Védőköpeny Hatása.....	189
8.2.4	Zsák Hatása	190
8.2.5	Sikeres lopás.....	190
8.2.6	Sikertelen lopás	191

8.2.7	Medvevírus elkapása	192
8.2.8	Genetikai kód megtanulása és ágens használata	192
8.2.9	Genetikai kód megtanulása és ágens használata	192
8.2.10	Az utolsó genetikai kód megtanulása és a játék megnyerése	192
8.2.11	Amnézia vírus hatása	192
8.2.12	Vakcina védelme	192
8.2.13	VitusTánc vírus hatása	193
8.3	Értékelés	193
8.4	Napló	193
9.	Grafikus felület specifikációja.....	194
9.1	A grafikus interfész	194
9.2	A grafikus rendszer architektúrája.....	195
9.2.1	A felület működési elve.....	195
9.2.2	A felület osztály-struktúrája	196
9.3	A grafikus objektumok felsorolása	196
9.3.1	Controller	196
9.3.2	GraphicsView	197
9.3.3	IObservable	197
9.3.4	ObservableAgent	198
9.3.5	ObservableEquipment	198
9.3.6	ObservableResource	198
9.3.7	ObservableTile	199
9.3.8	ObservableVirologist	199
9.3.9	ObservableGeneticCode	200
9.4	Kapcsolat az alkalmazói rendszerrel	200
9.5	Napló	204
10.	Grafikus változat beadása.....	205
10.1	Fordítási és futtatási útmutató.....	205
10.1.1	Fájllista	205
10.1.2	Fordítás és telepítés	207
10.1.3	Futtatás	207
10.2	Értékelés	208
10.3	Napló	209
11.	2 Összefoglalás.....	210
11.1	A projektre fordított összes munkaidő.....	210
11.2	• Projekt összegzés.....	211
11.2.1	Mit tanultak a projektból konkrétan és általában?	211

11.2.2 Mi volt a legnehezebb és a legkönnyebb?.....	211
11.2.3 Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?	211
11.2.4 Ha nem, akkor hol okozott ez nehézséget?	211
11.2.5 Milyen változtatási javaslatuk van?	211
11.2.6 Milyen feladatot ajánlanak a projektre?	211
11.2.7 Egyéb kritika és javaslat.....	212

2. Követelmény, projekt, funkcionálitás

52 – service_unavailable

Konzulens:
Micskei Zoltán

Csapattagok

Fekete Sámuel	GJ8J3A	feketesamu@gmail.com
Száraz Dániel	GT5X34	szdani2012@gmail.com
Jurásek Jónás	PH4QFK	jurasekjonas12@gmail.com
Domonkos Ádám	CWGYWC	domonkosadam01@gmail.com
Kászonyi Zsombor Marcell	DCE2Q1	kaszonyirelli@gmail.com

2022.02.23.

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

A dokumentum célja bemutatni a leendő projekt felépítését és a programozói döntéseket.

2.1.2 Szakterület

Többjátékos szórakoztató játékprogram.

2.1.3 Definíciók, rövidítések

pl. = például

UML = unified modelling language

stb = és a többi

2.1.4 Hivatkozások

iit.bme.hu

github.com

office.com

onedrive.com

plantuml.com

2.1.5 Összefoglalás

A továbbiakban a részletezett döntéseket mutatjuk be. Összeszedtük a program követelményeit, funkciót, majd meghatároztuk a use-case-eket. Egy projektindító megbeszélésen megterveztük a program felépítését és működését. Megbeszéltük a csapattagok felelősségeit, illetve a használni kívánt eszközöket is összeszedtük. Kialakítottuk a közös munkához elengedhetetlen rendszereket, hogy közösen tudjunk dolgozni.

2.2 Áttekintés

2.2.1 Általános áttekintés

A játék használatához mindenki egyetlen számítógépre lesz szükség. A kész játékot valamilyen formában letölve már lehet is használni, nem kell foglalkozni semmilyen egyéb hálózatkezeléssel és külső adatbázissal. (Azaz a 2.3.2-es pontban leírtakon kívül másra nem lesz szükség.)

A játék belső működése független a játék megjelenésétől. A játék működését egy belső "Kontroller" irányítja. A játékban különféle mezők vannak, különböző szerepekkel, ezekre tudnak a játékosok lépni. A játékosok a mezőkkel, valamint a többi játékossal tudnak akciókat kezdeményezni. Ilyen akciók lehetnek a lépések, ágensek használata, stb... (több a 2.2.2 végén.).

2.2.2 Funkciók

Egy pusztító biológiai katasztrófában mindenki elvesztette a látását. A városban virológusok kóborolnak és gyógymódot kutatnak.

A különféle vírusok genetikai kódja egy-egy laboratórium falára van felkarcolva. Ahhoz, hogy egy virológus a genetikai kódot megismerje, el kell jutnia az adott laboratóriumba, és le kell

tapogatnia a genetikai kódot. Ez alapján lehet majd vagy vakcinát, vagy magát a vírust előállítani.

Egy már megismert kód alapján a vírus vagy a vakcina (közös nevükön: ágens) létrehozható, de ehhez a virológusnak a szükséges mennyiséggű aminosavval és nukleotiddal (közös néven: anyag) kell rendelkeznie. Az aminosavak és a nukleotidok különféle raktárakban szedhetők össze, de mindenki csak korlátos mennyiséget vehet fel egy akcióval, és korlátos mennyiséget hordhat belőlük magánál. Ha a begyűjtött anyag mennyisége eléri ezt a korlátot, akkor többet már nem tud magához venni. Egy raktárban minden csak egy fajta anyag gyűjthető, de a raktárak nem tudnak kifogni az anyagból.

Egy virológus az előállított ágenst rövid időn belül felhasználhatja: vagy saját magára, vagy egy másik virológusra kenheti, de csak akkor, ha a kenést végző virológus meg tudja érinteni a másikat. Egy virológus akkor tud megérteni egy másikat, ha egy mezőn állnak. A felkent ágensek csak adott ideig hatásosak, az idő letelte után elbomlanak, hatásuk megszűnik.

Sokféle ágens létezik. Van olyan (vitustáncvírus), amelyik vitustáncot okoz: az áldozat kontrollálatlanul, véletlenszerű mozgással kezd el haladni. Van olyan (vakcina), amely megvéd attól, hogy más virológusok egyes ágensei hatással legyenek az ágens hatása alatt álló virológusra. Van olyan ágens (bénítóvírus), amely megbénít, így amíg az ágens hatása tart, az áldozat nem tud semmit csinálni (lebénül). Van olyan (amnéziavírus), amely hatására az áldozat elfelejti a már megismert genetikai kódokat.

A virológusok a vándorlás során védőfelszereléseket is gyűjthetnek. A védőfelszerelések a városban vannak szétszórva óvóhelyeken. Az óvóhely nem fogy ki a védőfelszerelésekkel. Egy felszerelés megszerzéséhez a virológusnak a megfelelő óvóhelyre kell bemennie, és a védőfelszerelést fel kell vennie. A felszerelések csak azt a virológust védi, aki viseli őket. A felszerelések hatása addig tart, amíg a virológus viseli őket. Egyszerre azonban maximum 3 felszerelés viselhető. A védőfelszerelésektől csak úgy lehet megszabadulni, ha azt egy másik virológus veszi el.

Sokféle védőfelszerelés létezik. Van védőköpeny, amely az ágenseket 82,3%-os hatásfokkal tartja távol. Van zsák, amely megnöveli a virológus anyaggyűjtő képességét. Van kesztyű, amellyel a felkent ágens a kenőre visszadobható.

A virológusok a vándorlásuk során találkozhatnak egymással. Találkozáskor elmehetnek egymás mellett, ágenst kenhetnek a másik virológusra, vagy, amíg a másik virológus lebénult állapotban van, elvehetik a másik anyagkészletét és felszerelését. Ilyenkor az anyagkészletből annyit visz el, amennyi belefér a korlátba, a többi a lebénult virológusnál marad. Ezenkívül 1 választott védőfelszerelést vehet el, ha még tud felvenni újat.

A játékot az a virológus nyeri, aki legelőször megtanulja az összes fellelhető genetikai kódot. A játéktér eltérő oldalszámú sokszögekből álló rácsot alkot, a virológusok ennek mezőin (szabad terület, raktár, óvóhely, laboratórium stb.) lépkedhetnek. Egy virológus minden csak a saját mezőjét észleli teljesen, ezenkívül a szomszédos mezőkre vezető utat tudja kitapogatni.

A játék körökre osztott. Egy körben egy virológus maximum 2 akciót hajthat végre. Akciónak számít a szomszédos mezőre való átlépés, genetikai kód leolvasása, ágens létrehozása, ágens alkalmazása, felszerelés felvétele, anyag gyűjtése, más virológustól lopás, illetve a passzolás.

2.2.3 Felhasználók

Nagyon fiatal gyerekekben és nagyon idős emberekben minden korosztálynak ajánljuk, mivel a játék nagyban épít a felhasználók memoriájára. Fontos, hogy mivel a felhasználók karakterei vakok, ezért csak nagyon minimális információt látnak a teljes pályáról, viszont az adott mezőn, amin állnak, arról minden tudnak. Ha valaki tehát képes megjegyezni a mezőket, amiket már járt, nagy előnyre lehet szert.

Mivel a játékot számítógépen lehet majd játszani, ezért a felhasználóknak rendelkezniük kell egy minimális tudással az adott szoftverekről (pl. a játék) és az adott hardverekről (pl. az egér).

2.2.4 Korlátozások

Legalább 1.8 verziójú JDK telepítése szükséges a számítógépre.

2.2.5 Feltételezések, kapcsolatok

- iit.bme.hu → A feladat szövege.
- github.com → A közös fejlesztéshez használt verziókezelő platform.
- office.com, onedrive.com → A közös dokumentáció készítéshez használt weboldalak.
- plantuml.com → UML diagramok készítéséhez használt weboldal.

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Kommunikáció
FK01	A különféle vírusok genetikai kódja egy-egy laboratórium falára van felkarcolva.	Bemutatás	alapvető	Feladat	Genetikai kód megismerése	
FK02	Játékos le tudja tapogatni az adott helyen található genetikai kódot.	Bemutatás	alapvető	Feladat	Genetikai kód megismerése	
FK03	Genetikai kód alapján lehet vakcinát vagy vírust előállítani.	Kiértékelés	alapvető	Feladat	Ágens készítés	
FK04	Ágens létrehozásához megfelelő mennyiségi anyaggal kell rendelkezni.	Kiértékelés	alapvető	Feladat	Ágens készítés	
FK05	Anyagok raktárakból szedhetőek össze.	Bemutatás	fontos	Feladat	Anyag felvétele	
FK06	Felvehető anyagok mennyisége korlátos.	Kiértékelés	fontos	Csatlakoztatás	Anyag felvétele	
FK07	Elkészített ágensek rövid ideig használhatóak.	Bemutatás	fontos	Feladat	Ágens alkalmazása, Idő műlása	
FK08	Ágensek felhasználhatóak másik virológusra, vagy saját magára.	Bemutatás	alapvető	Feladat	Ágens alkalmazása, Idő műlása	

FK09	A virulóbus a saját mezőjén tud tájékozódni.	Bemutatás	opcionális	Csapat	Lépés	
FK10	Egy virológus meg tud érinteni egy másikat, ha egy mezőn állnak	Bemutatás	fontos	Csapat	Másik játékos meglopása	
FK11	A virológusok a vándorlás során védőfelszereléseket is gyűjthetnek	Bemutatás	fontos	Feladat	Védőfelszerel és felvétele	
FK12	A védőfelszerelések szét vannak szórva óvóhelyeken	Kiértékelés	fontos	Feladat	Védőfelszerel és felvétele	
FK13	A felszerelések csak azt a virológust védi, aki viseli őket	Bemutatás	alapvető	Feladat	Védőfelszerel és felvétele	
FK14	Az óvóhely nem fogy ki a védőfelszerelések ből	Bemutatás	fontos	Csapat	Védőfelszerel és felvétele	
FK15	Egyszerre maximum 3 felszerelés viselhető.	Kiértékelés	fontos	Feladat	Védőfelszerel és felvétele	
FK16	A játékot az a virológus nyeri, aki legelőször megtanulja az összes fellelhető genetikai kódot.	Kiértékelés	alapvető	Feladat	Genetikai kód megismerése	
FK17	Egy körben egy virológus maximum 2 akciót hajthat végre.	Kiértékelés	opcionális	Csapat	Lépés, Genetikai kód megismerése, Ágens készítés, Ágens alkalmazása, Anyag felvétele, Védőfelszerel és felvétele, Másik játékos meglopása	
FK18	Egy virológus minden csak a saját mezőjét észleli teljesen,	Bemutatás	alapvető	Csapat	Lépés	

	ezenkívül a szomszédos mezőkre vezető utakat tudja kitapogatni.					
--	---	--	--	--	--	--

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
EKK01	Kell egy számítógép, a hozzá tartozó alap hardverekkel (monitor, billentyűzet, egér), egyéb extra hardverre nincs szükség.	Bemutatás	alapvető	csapat	Természetesen a számítógép rendelkezzen operációs rendszerrel is.
EKK02	Java és JDK	Bemutatás	alapvető	csapat	Célszerű az általunk használt verziókat használni.
EKK03	Git	Bemutatás	opcionális	csapat	Csak a fejlesztéshez szükséges

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
ÁKK01	A szoftverből egy futtatható jar fájl készül	Bemutatás	alapvető	csapat	
ÁKK02	A szoftverhez dokumentáció is tartozik.	Bemutatás	alapvető	csapat	
ÁKK03	A szoftverhez használati kézikönyv is tartozik. Ez a szoftver használatát írja le.	Bemutatás	alapvető	csapat	

2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
NFK01	A működéshez szükséges fő függvényekhez UNIT tesztek lesznek elérhetők, ezzel biztosítva a stabil, hibamentes működést.	Bemutatás	alapvető	csapat	
NFK02	A felhasználó rendelkezik minimális tapasztalattal szoftverek alkalmazásában	Bemutatás	alapvető	csapat	
NFK03	Stabil szoftver, ezt a tesztekkel tudjuk biztosítani.	Kiértékelés	alapvető	csapat	

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Idő múlása
Rövid leírás	A játékban az idő akkor telik, amikor egy kör véget ér. Az ágensek állapota változhat, amikor új kör kezdődik.
Aktorok	Kontroller
Forgatókönyv	<ol style="list-style-type: none"> Minden kör végén az ágensek lejárata és hatásuknak ideje (amennyiben egy játékoson van) csökken. <p>2/A. Egy alkalmazott ágens hatása X kör után eltűnik. 2/B. Egy elkészített ágens X kör után lejár és eltűnik.</p>

Use-case neve	Lépés
Rövid leírás	A játékosok mindenkor szomszédos mezőkre tudnak lépni.
Aktorok	Játékos
Forgatókönyv	<ol style="list-style-type: none"> Ha a játékoson olyan ágenst alkalmaztak, ami vitustáncot okoz, akkor véletlenszerűen lép az egyik szomszédos mezőre a körében. Ha a játékos le van bénulva, akkor nem tud lépni. Egyébként a játékos választja ki, hogy marad a jelenlegi mezőn vagy hogy melyik szomszédos mezőre lép. Szomszédos mezőre lépés egy akciónak számít.

Use-case neve	Genetikai kód megismerése
---------------	---------------------------

Rövid leírás	A játékosok laboratórium mezőkön genetikai kódokat tanulhatnak, ami alapján később ágenst készíthetnek.
Aktorok	Játékos
Forgatókönyv	<p>1/A. Ha a játékos laboratórium mezőn áll, és még nem tanulta meg az itt lévő genetikai kódot, akkor választhatja a genetikai kód megtanulása akciót, ha van még akciója a körben.</p> <p>1/B. Az akció elvégzése után a genetikai kódot megjegyzi a játékos, és amíg azt el nem felejt (egy ágens hatására) addig készíthet ez alapján ágenst.</p> <p>2. Egyébként a játékos ezen a mezőn nem tanulhat kódot.</p>

Use-case neve	Ágens készítés
Rövid leírás	A játékosok a megismert genetikai kódok alapján készíthetnek vírust vagy vakcinát.
Aktorok	Játékos
Forgatókönyv	<ol style="list-style-type: none"> Ágens készítés is egy akció, tehát a játékosok csak akkor hajthatják végre, ha van még akciójuk a körben. Csak akkor tudják megtenni, ha ismerik az ágenshez tartozó genetikai kódot és ha van elég anyaguk az adott ágens elkészítéséhez Ha sikeresen elkészítette az ágenst, akkor anyagkészlete csökken az elkészítéshez szükséges anyaggal.

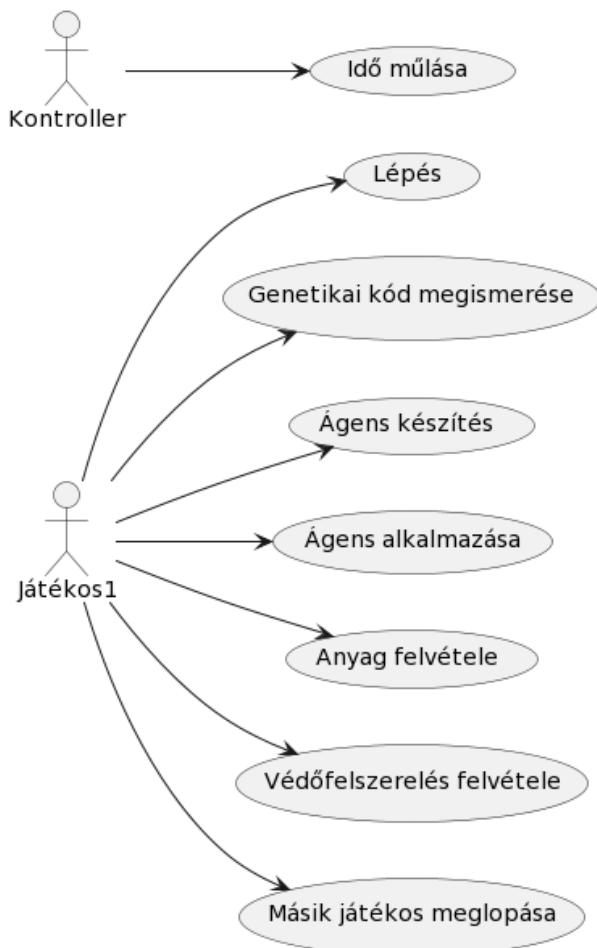
Use-case neve	Ágens alkalmazása
Rövid leírás	A játékosok az elkészített ágenst használhatják magukra vagy más játékosra.
Aktorok	Egy mezőn tartózkodó játékosok
Forgatókönyv	<ol style="list-style-type: none"> Ágens alkalmazása is egy akció, tehát a játékos csak akkor hajthatja végre, ha van még akciója a körben. A. Ha a játékos mezőjén más játékos is tartózkodik, akkor rá is alkalmazható. B. A játékos saját magára minden alkalmazhatja az ágenst. A. Ha a játékoson, akire az ágenst alkalmazzák, van olyan hatás, ami megvéd más hatástól, akkor hatástalan az ágens. B. Egyébként az ágens hatása fix ideig aktív a játékoson.

Use-case neve	Anyag felvétele
Rövid leírás	A játékosok raktár mezőkön valamilyen anyagot vehetnek fel, a raktár típusától függően.
Aktorok	Játékos
Forgatókönyv	<ol style="list-style-type: none"> Ha a játékos raktár mezőn áll, akkor választhatja az anyag felvétele akciót, ha van még akciója a körben. Ekkor fix mennyiségi anyagot kap a játékos. A játékos anyagkészlete korláatos, egy fix értéket nem haladhat meg, ennél anyagfelvétel után sem lesz több nála.

Use-case neve	Védőfelszerelés felvétele
Rövid leírás	A játékosok óvóhely mezőn felszerelést vehetnek fel.
Aktorok	Játékos
Forgatókönyv	<ol style="list-style-type: none"> Ha a játékos óvóhely mezőn áll és nincs kettőnél több felszerelés rajta, akkor választhatja a felszerelés felvétele akciót, ha van még akciója a körben. Az akció után a felszerelés hatása addig érvényes a játékosra, amíg viseli, azaz valaki el nem lopja tőle.

Use-case neve	Másik játékos meglopása
Rövid leírás	A játékosok elvehetik azon játékosok védőfelszerelését és anyagkészletét, akik velük egy mezőn és bénult állapotban vannak.
Aktorok	Egy mezőn tartózkodó játékosok
Forgatókönyv	<ol style="list-style-type: none"> Ha a játéknak a mezőjén tartózkodik egy olyan játékos, aki le van bénulva, akkor választhatja a játékos meglopása akciót, ha van még akciója a körben. Az akció hatására a támadó játékos megkapja a támadott játékos anyagkészletéből azt a maximális mennyiséget, amivel még nem lépi túl az anyagkészlet felső korlátját. Ha a támadott játéknak van egy vagy több, a támadónak pedig háromnál kevesebb védőfelszerelése, akkor a támadó választhat a támadott játékos védőfelszerelései közül egyet, amit ellop tőle. Ezzel a felszerelés hatása a támadott játékosról átkerül a támadó játékosra.

2.4.2 Use-case diagram



2.5 Szótár

Akció: A játékos által választható tevékenység, amit a virológusa végrehajt: Lépés, Kenés, Letapogatás, Lopás, Ágens készítése, Anyaggyűjtés

Anyag: Olyan gyűjthető erőforrás, ami ágensek készítéséhez szükséges.

Anyaggyűjtés: Akció, amikor a felhasználó növeli anyagkészletét.

Anyagkészlet: A virológus jelenlegi anyagmennyisége.

Aminosav: Egy anyagtípus.

Amnéziavírus: Ágens, amelynek hatására az áldozat elfelejtíti az eddig tanult Genetikai kódokat.

Ágens: vakcina vagy vírus

Ágens készítése: Akció, amikor a virológus anyagkészletének csökkentésével egy ismert genetikai kód alapján elkészít egy ágenst.

Áldozat: Virológus, akit megpróbálnak megkenni.

Bénított: Áldozat nem tud csinálni semmit.

Bénítóvírus: Ágens, amelynek hatására az áldozat megbénul.

Genetikai kód: Ez alapján lehet majd ágenst előállítani.

Hatásfok: Azt adja meg, hogy egy tevékenység mekkora eséllyel lesz sikeres.

Játék: A készítendő program.

Játékos: A felhasználók, akik részt vesznek a játékban.

Játéktér: A mezők összessége, amin a virológusok lépkedhetnek.

Kenés: Akció, amikor egy virológus elkészített ágensét használja egy megérinthető virológuson.

Kenő: Támadó virológus, aki ken.

Kesztyű: Védőfelszerelés, amellyel a felkent ágens a kenőre visszadobható.

Korlát: Maximális/minimális mennyiségek

Kör: A játékos ekkor cselekedhet.

Laboratórium: Mező, ahol genetikai kódok vannak.

Letapogatás: Akció, Amikor a virológus megtanul egy genetikai kódot.

Lépés: Mozgás egyik mezőről egy szomszédos mezőre.

Lopás: Akció, amikor egy virológus egy bénított virológustól anyagot, illetve védőfelszerelést vess el.

Megérintható: Az érintő és az érintett egy mezőn állnak.

Mező: Objektumok a pályán, ahova a játékos léphet.

Nukleotid: Egy anyagtípus.

Nyertes: Az a játékos, akinek virológusa először tanulja meg az összes genetikai kódot.

Óvóhely: Mező, ahol védőfelszereléseket lehet felvenni.

Passz: Akció, amikor a virológus nem csinál semmit.

Raktár: Mező, ahol anyagokat lehet gyűjteni.

Szabad terület: Mező, ahol nincs semmilyen erőforrás.

Találkozás: Két virológus egy mezőn tartózkodik egy időben

Vak: Elvesztette a látását.

Vakcina: Ágens, amely megvéd a többi ágens hatásától.

Vándorlás: Lépések sorozata

Védőfelszerelés: Olyan gyűjthető eszköz, ami segítséget nyújt viselőjének.

Védőköpeny: Védőfelszerelés, ami az ágenseket 82,3%-os hatásfokkal tartja távol.

Virológus: A játékban résztvevő karakter, akit a játékos irányít.

Vitustánc: az áldozat kontrollálatlanul, véletlenszerű mozgással kezd el haladni

Vitustáncvírus: Ágens, amelynek hatására az áldozat kontrollálatlanul mozog.

Zsák: Védőfelszerelés, ami növeli a virológus anyagkorlátját.

2.6 Projekt terv

A projektben február 28-ig meghatározzuk a követelményeket és a program követelményeit. Március 16-ig meghatározzuk az analízismodellt, majd elkezdjük implementálni. Március 28-ig elkészítjük a program szkeleton változatát, amiben az adatstruktúrák, és közöttük lévő kapcsolatok helyesek és teljesek. Április 25-ig továbbfejlesztjük a programot prototípus állapotba. Ekkor már egy konzolos felületen irányítható a játék. A játék logikája és a belső vezérlés interfésze végleges. Május 16-ig elkészítjük a program kész változatát, amely MVC modell szerint készült. A játék logikája megegyezik a prototípusban láthatókkal, de már grafikus felüettel jelenik meg. A végleges grafikus program elkészülte után befejezzük az egyesített dokumentációt.

Domonkos Ádám a kapcsolattartó, ō koordinálja a kiosztott munkákat és befejezésüket.

Fekete Sámuel a verziókezelés felügyelője, ō kezeli a brancheket és pull requesteket.

Jurásek Jónás a dokumentáció felügyelője, ō ellenőrzi a dokumentumok teljességet és egységét.

Száraz Dániel segít Fekete Sámuelnek a verziókezelésben és a Jurásek Jónásnak a dokumentumok ellenőrzésében.

Kászonyi Zsombor a kód minőségéért és egységességeért felel, valamint a határidőket és az egyéb elvégzett munkák minőségét ellenőrzi.

A dokumentációkat a Microsoft Word online szerkesztőjével készítjük. Azért döntöttünk emellett, mert egyből egy formázott dokumentumot kapunk, és egyszerre többen tudjuk szerkeszteni egyidőben.

A kódokat a github.com nevű weboldalon osztjuk meg. Azért esett erre a megoldásra a választásunk, mert ez az egyik leghatékonyabb verziókezelő szoftver, amivel közösen tudunk dolgozni.

2.7 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2022.02.23. 19:30	1 óra	Fekete Száraz Jurásek Domonkos Kászonyi	Értekezlet. DÖNTÉS: Körökre osztott lokális stratégiai játék DÖNTÉS: körönként max 2 akció DÖNTÉS: ágens létrehozása: nukleotid, aminosav elveszik DÖNTÉS: A kenés után eltűnik az ágens DÖNTÉS: akció: leolvasás, ágens létrehozása, ágens alkalmazása, felszerelés elvétel, felszerelés felvétele, lépés másik mezőre, passz
2022.02.23. 20:30	2 óra	Fekete	Use case leírások kezdetleges megírása
2022.02.23. 20:30	2 óra	Száraz	Felhasználók leírása, napló vázának készítése
2022.02.23. 20:30	2 óra	Jurásek	Funkciók megírása
2022.02.23. 20:30	2 óra	Domonkos	Funkcionális követelmények, Szótár, Korlátozások megírása
2022.02.25. 12:00	1 óra	Fekete	Use case leírások véglegesítése és diagram elkészítése
2022.02.25. 20:50	2 óra	Kászonyi	Átadással kapcsolatos követelmények, Egyéb nem funkcionális követelmények.

			Funkciók befejezése, Szótár
2022.02.26. 10:00	1 óra	Jurásek	
2022.02.26. 13:00	2 óra	Domonkos	Funkcionális követelmények befejezése, dokumentum szerkesztése
2022.02.26. 9:00	2 óra	Jurásek	Funkcionális követelmények ellenőrzése, a Dokumentum formázása
2022.02.27 15:30	4 óra	Száraz	Általános áttekintés megírása, Feltételezések, kapcsolatok megírása, Teljes dokumentum átnézése, ellenőrzése, nyelvtani hibák javítása, felmerülendő kérdések kijegyzetelése.
2022.02.27 17:10	3 óra	Kászonyi	Dokumentum ellenőrzése és szerkesztése, nyelvtani, elvi és fogalmi hibák javítása, értekezlet megszervezése.
2022.02.27. 18:00	1 óra	Domonkos	Dokumentum ellenőrzése, use-case diagrammok kijavítása
2022.02.27. 18:15	1 óra	Fekete	Szótár ellenőrzése
2022.02.27. 21:00	1 óra	Fekete Száraz Jurásek Domonkos Kászonyi	Értekezlet: DÖNTÉS: a dokumentum minősége megfelelő, szép, kielégítő

3. Analízis modell I.

52 – service_unavailable

Konzulens:
Micskei Zoltán

Csapattagok

Fekete Sámuel	GJ8J3A	feketesamu@gmail.com
Száraz Dániel	GT5X34	szdani2012@gmail.com
Jurásek Jónás	PH4QFK	jurasekjonas12@gmail.com
Domonkos Ádám	CWGYWC	domonkosadam01@gmail.com
Kászonyi Zsombor Marcell	DCE2Q1	kaszonyirelli@gmail.com

2022.02.23.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Anyag - Resource

Ágensek létrehozásához szükséges erőforrás. Két típusa az aminosav, illetve a nukleotid.

3.1.2 Aminosav – AminoAcid

Bizonyos ágensek létrehozásához szükséges erőforrás. Raktár ból lehet szerezni ebből az anyagból.

3.1.3 Amnéziavírus - AmnesiaVirus

Játékosok által összegyűjtött genetikai kódok elfelejtéséért felel, amikor felkenik ezt a vírust. Ő felel a saját hátralévő idejének ellenőrzéséért is.

3.1.4 Ágens - Agent

A különböző vírusok és vakcinák felkenéséért felelős, ő jelzi az áldozatnak, hogy megtámadták, és ő adja át a hatását. Felel a lejáratának idejéért, azaz ő fog jelezni, hogyha már többé nem felhasználható.

3.1.5 Bénítovírus - StunVirus

A bénító hatás kifejtéséért felelős, amikor felkenik egy virológusra. Ő felel a saját hátralévő idejének ellenőrzéséért is.

3.1.6 Felszerelés - Inventory

Egy virológus anyagkészletéért és felvett védőfelszerelések számontartásáért felelős. Megakadályozza, hogy egy virológus valamelyik anyaga túllépje a felső korlátot vagy hogy háromnál több védőfelszerelése legyen.

3.1.7 Genetikai kód - GeneticCode

Ezeket gyűjtik a virológusok, ha valaki az összeset összegyűjtötte, akkor megnyerte a játékot.

3.1.8 Hatás - Effect

Egy alkalmazott ágens vagy viselt védőfelszerelés hatásának kifejtéséért, vagy ellenőrzéséért felel.

3.1.9 Játék (kontroller) - Controller

A játék vezérléséért felelős. Ő felel a játékosok sorrendben sorakerüléséért, illetve annak ellenőrzéséért, hogy egy hatás ezt nem akadályozza ezt meg.

3.1.10 Kesztyű - Glove

Ágensek visszadobásáért felelős, amikor a virológusra fel akarnak kenni egy vírust.

3.1.11 Laboratórium - Laboratory

Genetikai kódok leolvasásáért, megtanulásáért felelős.

3.1.12 Mező - Tile

A pálya ilyen elemeket tartalmaz. Felelős a virológus tájékozatásáért, mik találhatóak a mezőn. Pl. Anyag, genetikai kód, másik virológus, védőfelszerelés. Ismeri a szomszédait, és a rajta lévő varázslókat.

3.1.13 Nukleotid - Nucleotid

Bizonyos ágensek létrehozásához szükséges erőforrás. Raktár ból lehet szerezni ebből az anyagból.

3.1.14 Óvóhely - Safehouse

A virológusok felszerelését tudja bővíteni védőfelszereléssel.

3.1.15 Pálya - Map

A mezők betöltéséért, létrehozásáért és tárolásáért felelős.

3.1.16 Raktár - Warehouse

A védőfelszerelések kiosztásáért felelős. Ő helyezi be az ott található védőfelszerelést a varázsló felszerelései közé.

3.1.17 Szabad terület - EmptyTile

Egy olyan mező, amin nincs semmi.

3.1.18 Vakcina - Vaccine

Az ő felelőssége a virólógrusra érkező ágensek megakadályozása, amennyiben aktív, illetve a hatásából hátralévő idő ellenőrzése.

3.1.19 Védőfelszerelés - Equipment

Ő felel a hatásának érvényesítéséért.

3.1.20 Védőköpeny - ProtectiveCloak

Ő felel a sorsolásáért, hogy sikerül-e megvédeni a virológust, illetve az eredmény jelzéséért, végrahajtásáért.

3.1.21 Virológus - Virologist

Játékost szimbolizáló osztály, ő felel a lépéseiért, felszereléséért, illetve ő tud különböző akciókat végrehajtani.

3.1.22 Vitustáncvírus - VitusDanceVirus

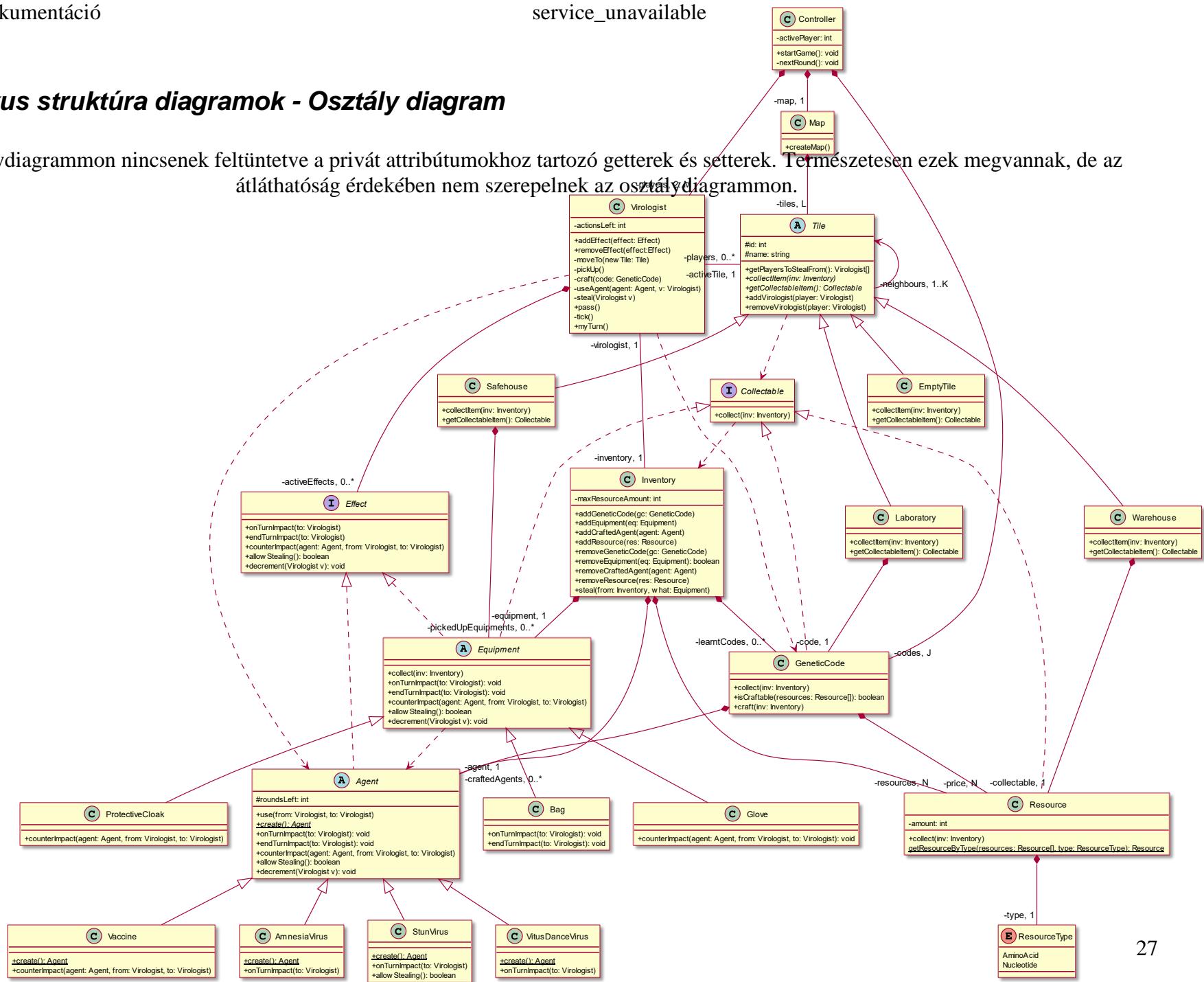
A vitustánc hatásának kifejtéséért felelős, amikor felkenik egy virológusra. Ő felel a saját hátralévő idejének ellenőrzéséért is.

3.1.23 Zsák – Bag

A virológus maximális anyagmennyiségeinek megnöveléséért felelős védőfelszerelés. Amikor birtokolja ezt a felszerelést egy virológus, akkor lép életbe a hatása.

3.2 Statikus struktúra diagramok - Osztály diagram

Az osztálydiagramon nincsenek feltüntetve a privát attribútumokhoz tartozó getterek és setterek. Természetesen ezek megvannak, de az átláthatóság érdekében nem szerepelnek az osztálydiagramnon.



3.3 Osztályok leírása

3.3.1 Agent

- **Felelősség**
A különböző vírusok és vakcinák absztrakt ōsosztálya. Ez reprezentálja az elkészített, de még nem alkalmazott, illetve az alkalmazott, aktív ágenseket is. Körönként csökken a `roundsLeft` tagváltozója, amikor eléri a nullát, törlődik.
- **Ősosztályok**
-
- **Interfészek**
 - Effect
- **Attribútumok**
 - **protected int roundsLeft:** hány kör múlva jár le a hatása, vagy hány körig alkalmazható még.
- **Metódusok**
 - **abstract Agent create():** Visszaad egy másolatot önmagáról. Az osztály minden leszármazottjának úgy kell megvalósítania, hogy a leszármazott típust adja vissza. Amikor a játékos egy genetikai kód szerint létrehoz egy ágenset, akkor a kód `Agent` mezőjében lévő `Agent`-en hívódik meg ez a metódus, az eredménye pedig a játékos `Inventory`-jának `craftedAgents` változójában tárolódik el.
 - **use(Virologist from, Virologist to):** Egy `Virologist(from)` elkészített `Agent`jét rákeni egy `Virologist(to)`.
 - **void onTurnImpact(Virologist to):** Nem csinál semmit, így a játékos körében semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
 - **void counterImpact(Agent agent, Virologist from, Virologist to):** Nem csinál semmit, így a kenés elleni védekezésben semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
 - **void endTurnImpact(Virologist to):** Nem csinál semmit, így a játékos körének végén semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
 - **bool allowStealing():** Hamissal tér vissza, így azoknak a felszereléseknek, akik explicit nem engedélyezik, hogy lopjanak a játékostól, akin aktívak, nem kell felüldefiniálnia.
 - **void decrement(Virologist v):** Csökkenti a `roundsLeft` attribútum értékét eggyel. Ha eléri a nullát, akkor megpróbálja kivenni magát a paraméterként kapott `Virologist activeEffect`-jéből és az `Inventory craftedAgents`-éből, de csak az egyikben volt benne, tehát csak az egyikből fog sikerülni.

3.3.2 AmnesiaVirus

- **Felelősség**

Amelyik játékosra alkalmazzák ezt a vírust, az elfelejti az összes eddig megtanult geneticCode-ot.

- **Ősosztályok**

- Agent

- **Interfészek**

-

- **Attribútumok**

- **protected int roundLeft:** Az ágens hátralévő idejét adja meg.

- **Metódusok**

- **Agent create():** Létrehoz egy AmnesiaVirus-t és ezt visszaadja a hívónak, aki aztán ezt el tudja tárolni.
- **void OnTurnImpact(Virologist to):** A kiválasztott Virologist ismert genetikai kódjait kiüríti.

3.3.3 Bag

- **Felelősség**

Olyan védőfelszerelés, aminek hatására a Virologist kétszer annyi Resource-t tud eltárolni minden fajtából.

- **Ősosztályok**

- Equipment

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **void onTurnImpact(Virologist to):** A Virologist-nak megkétszerezi a maxResourceAmount értékét.
- **void endTurnImpact(Virologist to):** A Virologist-nak megfelezi maxResourceAmount értékét.

3.3.4 Collectable

- **Felelősség**

Az összes összegyűjthető dologhoz tartozó interface. Felelőssége, hogy a gyűjthető dolgok összegyűjtésükkor elhelyezzék magukat a Virologist inventory-jában.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

void collect(Inventory inv): Az adott összegyűjthető dolgot belerakja a paraméterben átadott *Inventory*ba. minden gyűjthető dolognak definiálnia kell, hogy hova szeretné magát rakni.

3.3.5 Controller

- **Felelősség**

A játék menedzselése a háttérben, egymás utáni körök, pálya legenerálását kezdeményezi.

- **Ősosztályok**

-

- **Interfész**

-

- **Attribútumok**

- **Virologist players:** Az aktuális játékban résztvevő virológusok.
- **GeneticCode codes:** Az aktuális játékban használt összes genetikai kód.
- **Map map:** Az aktuális játékhoz tartozó pálya.
- **int activePlayer:** megadja a soron lévő játékos helyét a tömbben

- **Metódusok**

- **void startGame():** Elindítja a játékot, inicializálja a szükséges változókat.

3.3.6 Effect

- **Felelősség**

A hatásokat kényszeríti ki, hogy minden eshetőségre készítsék el a saját reakciójukat, még akkor is, ha ők egy adott dologba nem avatkoznak bele.

- **Ősosztályok**

-

- **Interfész**

-

- **Attribútumok**
 -

- **Metódusok**

- **void onTurnImpact(Virologist to):** Az olyan hatások használják, amik a játékos körében az akciók végkimenetelét változtathatják.
- **void counterImpact(Agent agent, Virologist from, Virologist to):** Az olyan hatások használják, amik a mások körében rájuk kent ágensekre hatnak.
- **void endTurnImpact(Virologist to):** Az olyan hatások használják, amiknek a játékos körének végén semmisé kell tenniük a játékos körére vonatkozó történéseket.
- **bool allowStealing():** Az adott hatás engedélyezi-e, hogy lopjanak attól a játékostól, aki aktív. Ha hamissal tér vissza, az nem azt jelenti, hogy megtiltja a lopást, hanem hogy ő nem engedélyezi, de attól még egy másik aktív hatás engedélyezheti.
- **void decrement():** Csökkenti az *Effect* élettartamát.

3.3.7 EmptyTile

- **Felelősség**

- A sima mezőket reprezentáló osztály, Bármennyi játékos rá tud lépni, azonban nincs rajta semmilyen *Collectable* interfész megvalósító, a *Virologist* által felvehető objektum.

- **Ősosztályok**

- Tile

- **Interfészek**
 -

- **Attribútumok**
 -

- **Metódusok**

- **void collectItem(Virologist player):** A mezőn nincs felvehető objektum, ezért a függvény lefut, de a játékos felszerelése nem bővül.
- **Collectable getCollectableItem():** A mezőn nem található *Collectable* interfész megvalósító objektum, ezért *null*-val tér vissza.

3.3.8 Equipment

- **Felelősség**

A felszereléseket reprezentáló absztrakt osztály.

- **Ősosztályok**
 -

- **Interfészek**

- Collectable

- Effect
- **Attribútumok**
 -
- **Metódusok**
 - **void onTurnImpact(Virologist to):** Nem csinál semmit, így a játékos körében semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
 - **void counterImpact(Agent agent, Virologist from, Virologist to):** Nem csinál semmit, így a kenés elleni védekezésben semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
 - **void endTurnImpact(Virologist to):** Nem csinál semmit, így a játékos körének végén semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
 - **bool allowStealing():** Hamissal tér vissza, így azoknak a felszereléseknek, akik explicit nem engedélyezik, hogy lopjanak a játékostól, akin aktívak, nem kell felülfeldefiniálnia.
 - **collect(Inventory inv):** Elrakja az *Inventory* osztály *pickedUpEquipments* változójába a felszerelést, ha az még elfér ott.
 - **void decrement(Virologist v):** Nem csinál semmit, mert az *Equipment*-nek nincs élettartama.

3.3.9 GeneticCode

- **Felelősség**

A játékosok ilyeneket gyűjthetnek laboratórium mezőkön és ezekből tudnak ágenseket készíteni, ha rendelkeznek a szükséges mennyiségű anyaggal. Az a játékos nyeri a játékot, aki először összegyűjt az összes *GeneticCode*-ot.
- **Ősosztályok**
 -
- **Interfészek**
 - Collectable
- **Attribútumok**
 - **const Resource[] Price** - A genetikai kódból *Agent* készítéséhez szükséges anyagmennyiségek. *Resource*-ok tömbje, mert akár több különböző anyag is kellhet a készítéshez.
 - **const Agent agent** - A kódból készíthető ágens.
- **Metódusok**
 - **void collect(Inventory inv):** A paraméterként kapott *Inventory*-n meghívja az *addGeneticCode()* függvényt, ezzel eltárolja annál a játékosnál, hogy ő megtanulta ezt a kódot. Egyszerre több játékosnál is el lehet tárolva.
 - **bool isCraftable(Resource[] resources):** Akkor ad igazat, ha a paraméterként kapott *resources* tömb minden anyagából van legalább annyi, mint *price*-ban.

- **void craft(Inventory inv):** *inv*-en meghívja a `removeResource()` metódust `price` minden elemével, majd az `addCraftedAgent()` függvényt az `agent create()` metódusának visszatérési értékével. Tehát levonja a virológustól az elhasznált anyagot, majd odaadja az elkészült ágenst. Feltételezi, hogy a virolónak van elegendő anyaga, tehát csak akkor hívandó, ha `isCraftable()` igazzal tért vissza.

3.3.10 Glove

- **Felelősség**
Olyan védőfelszerelés, ami megvédi az áldozatot az Ágens hatásától, és ezzel együtt a támadó játékosra pedig rákeni az ágenst.
- **Ősosztályok**
 - Equipment
- **Interfészek**
 -
- **Attribútumok**
 -
- **Metódusok**
 - **void counterImpact(Agent agent, Virologist from, Virologist to):** A megkent `Virologistról (to)` leveszi, a támadó `Virologistra (from)` pedig átrakja a kent `Agent` hatását.

3.3.11 Inventory

- **Felelősség**
Egy virológus összes anyagát, genetikai kódját, ágensét, illetve védőfelszerelését tárolja, illetve ezeket kezeli, hozzáadja az újakat, illetve törli az elvesztetteket.
- **Ősosztályok**
 -
- **Interfészek**
 -
- **Attribútumok**
 - **private Agent[] craftedAgents:** a virológus összes ágensét tartalmazó lista
 - **private GeneticCode[] learntCodes:** a virológus összes megtanult genetikai kódját tartalmazó lista
 - **private Equipment[] pickedUpEquipments:** a virológus összes felvett védőfelszerelését tartalmazó lista
 - **private Resource[] resources:** a virológus összes anyagát tartalmazó lista

- **private int maxResourceAmount:** A maximálisan összegyűjthető anyagok mennyisége
- **private Virologist virologist:** A felszerelés tulajdonosát tárolja
- **Metódusok**
 - **void addGeneticCode(GeneticCode gc):** Hozzáad egy Genetikai kódot a *learntCodes* listához.
 - **void addEquipment(Equipment eq):** Hozzáad egy felszerelést a *pickedUpEquipments* listához.
 - **void addCraftedAgent(Agent agent):** Hozzáad egy új ágenst a *craftedAgents* listához.
 - **void addResource(Resource res):** Hozzáad egy új anyagot a *resources* listához.
 - **void removeGeneticCode(GeneticCode gc):** Eltávolítja a megadott genetikai kódot virológustól.
 - **bool removeEquipment(Equipment eq):** A *pickedUpEquipments* listából eltávolítja a *removedEquipment*-et, ha az ott van. Ha nincs, hamissal tér vissza, egyébként igazzal.
 - **void removeCraftedAgent(Agent agent):** *craftedAgents* listából törli *agent*-et.
 - **void removeResource(Resource res):** A *resources* lista azon eleméből, melynek típusa megegyezik *res* típusával, levon annyit, ami *res amount* mezőjében van.
 - **void steal(Inventory from, Equipment what):** *from Resource*-ait odaadja a *resources* tömbnek, majd levonja *from*-ból. Ha *what* nem *null*, akkor kiveszi *from*-ból, és berakja *pickedUpEquipments*-be.

3.3.12 Laboratory

- **Felelősség**

A játékban található Laboratórium mezőket reprezentáló osztály. A rajta található genetikai kódok átadásáért felelős a rajta álló játékosok számára.
 - **Ősosztályok**
 - Tile
 - **Interfészek**
 -
- 2. Attribútumok**
- **-GeneticCode code:** A Laboratórium mezőn található, felvehető genetikai kódot tárolja
 - **Metódusok**
 - **void collectItem(player: Virologist):** A mezőről felvehető *Collectable* interfészt megvalósító *GeneticCode* objektumot átadja a paraméterként kapott játékos (*Virologist*) számára.

- **Collectable getCollectableItem()**: Visszaadja a mezőn található `Collectable` interfész megvalósító `GeneticCode` objektumot.

3.3.13 Map

- **Felelősség**

A játékot alkotó pálya tárolásáért felelős osztály. Tárolja a mezőket, amik a közvetlen szomszédoch ismerete miatt lefedi az egész pályát. Ő felelős a pálya beolvasása és inicializálásáért.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **Tile[L] tiles**: A pálya mezői

- **Metódusok**

- **void createMap()**: Betölti a pályát a `tiles` attribútumába. Létrehozza a pályaelemeket és szomszédságokat.

3.3.14 ProtectiveCloak

- **Felelősség**

Olyan védőfelszerelés, amely az áldozat virológusokat 82.3% eséllyel védi meg a rájuk kent ágensek hatásától. Ilyenkor az ágens elhasználódik, és hatását senki nem kapja meg.

- **Ősosztályok**

- Equipment

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **void counterImpact(Agent agent, Virologist from, Virologist to)**: Sorsol egy számot 1-től 1000-ig. Ha a szám nem nagyobb, mint 823, megkent virológusról levelez a kent ágens hatását.

3.3.15 Resource

- **Felelősség**

Egy adott anyagból egy bizonyos mennyiséget reprezentál.

- **Ősosztályok**

-

- **Interfészek**

- Collectable

- **Attribútumok**

- **int amount:** hány egység anyagot reprezentál az objektum
- **const ResourceType type:** milyen típusú anyagot reprezentál az objektum

- **Metódusok**

- **static Resource getResourceByType(Resource[] resources, ResourceType type):**
A *resources* tömbben megkeresi az első olyan elemet, amelyik *type* típusú, és azt visszaadja. Olyan *Resource*]-on érdemes használni, amiben minden típusból egy van (ilyen például az *Inventory* *resources* mezője vagy a *GeneticCode* *price* mezője).
- **void collect(Inventory inv):** A paraméterként kapott *Inventory*-ba berak annyi anyagot, amennyi ebben az objektumban éppen van.

3.3.16 ResourceType

- **Felelősség**

A játékban szereplő anyagtípusok.

- **Lehetséges értékek**

- **AminoAcid**
- **Nucleotide**

3.3.17 Safehouse

- **Felelősség**

A játékban szereplő óvóhely pályaelem reprezentációjáért felelős osztály. A mezőn található Collectable interfészt megvalósító Equipment objektum felvételéért felelős.

- **Ősosztályok**

- Tile

- **Interfész**

-

- **Attribútumok**

- **Equipment equipment:** A Safehouse-ban lévő equipment.

- **Metódusok**

- **void collectItem(Virologist player):** A mezőről felvehető `Collectable` interfészt megvalósító `Equipment` objektumot átadja a paraméterként kapott játékos (`Virologist`) számára.
- **Collectable getCollectableItem():** Visszaadja a mezőn található `Collectable` interfészt megvalósító `Equipment` objektumot.

3.3.18 StunVirus

- **Felelősség**

Amelyik virológusra alkalmazzák azt a következő x körre megállítja akciók végrehajtásában.

- **Ősosztályok**

- Agent

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **Agent create():** Létrehoz egy `StunVirus`-t és ezt visszaadja a hívónak, aki aztán ezt el tudja tárolni.
- **void onTurnImpact(Virologist to):** A kiválasztott `Virologist`-et lebénítja adott számú körre, így nem fog tudni akciókat végrehajtani.
- **boolean allowStealing():** Igazzal tér vissza, mert akin aktív ez az effekt, attól lehet lopni.

3.3.19 Tile

- **Felelősség**

- A pályát alkotó mezőket reprezentáló osztály. Privát tagváltozóban tárolja az egyedi azonosítóját, illetve a megjelenítési nevét. Ismeri a rajta tartózkodó `Virologist` objektumokat, illetve az adott mező közvetlen szomszédait. A mezőn található `Collectable` interfészt megvalósító objektumok listázásáért és átadásáért felel a rajta lévő játékosoknak.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**
 - **protected Virologist players:** Mezőn álló játékosok (*Virologist*) referenciáját tárolja
 - **protected Tile neighbours:** Szomszédos mezők (*Tile*) referenciáját tárolja
 - **protected int id:** Mező egyedi azonosítója
 - **protected string name:** Mező megjelenítési neve
- **Metódusok**
 - **void collectItem(Virologist player):** A mezőről felvehető *Collectable* interfész megvalósító objektumot átadja a paraméterként kapott játékos (*Virologist*) számára. Ha mezőn nincs felvehető objektum, a függvény lefut, de a játékos felszerelése nem bővül.
 - **Collectable getCollectableItem():** Visszaadja a mezőn található *Collectable* interfész megvalósító objektumot. Ha a mezőn nem található ilyen objektum, akkor *null*-val tér vissza.
 - **Virologist[] getPlayersToStealFrom():** A *players* mező elemei közül azokat adja vissza, amin van olyan *Effect*, amire az *allowStealing()* igazzal tér vissza.
 - **void addVirologist(Virologist player):** Hozzáadja az adott mezőhöz a virológust.
 - **void removeVirologist(Virologist player):** Leveszi az adott mezőről a virológust.

3.3.20 Vaccine

- **Felelősség**

Olyan ágens, amely az áldozat virolágusokat megvédi a rájuk kent ágensek hatásától. Ilyenkor az ágens elhasználódik, és hatását senki nem kapja meg.
- **Ősosztályok**
 - Agent
- **Interfészek**
 -
- **Attribútumok**
 -
- **Metódusok**
 - **void counterImpact(Agent agent, Virologist from, Virologist to):** A megkent virolágusról leveszi a kent ágens hatását.
 - **static Agent create():** Létrehoz egy *Vaccine*-t és ezt visszaadja a hívónak, aki aztán ezt el tudja tárolni.

3.3.21 Virologist

- **Felelősség**

Ez az osztály reprezentálja a játékosokat a játékban. Feladatai közé tartoznak a különféle akciók, amiket a játékos a körében választhat. Ez az osztály kommunikál a felhasználóval, hogy mikor melyik akciót szeretné a játékos végrehajtani.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **int actionsLeft:** Ennyi akciója van még az adott körben a játékosnak.
- **Effect[] activeEffects:** Az összes effekt, ami ezen a virológuson aktív.
- **Tile activeTile:** A mező, amin a virológus tartózkodik.
- **Inventory inv:** A virológus felszerelése, ami tartalmazza az összes megszerzett anyagot, ágenst, genetikai kódot és védőfelszerelést.

- **Metódusok**

- **Collectable[] getCollectedItems():** Visszaadja, az eddig összegyűjtött összes gyűjthető dolgot.
- **void addEffect(Effect effect):** Hozzáad egy effektet a játékoshoz.
- **void removeEffect(Effect effect):** Levesz egy effektet a játékosról.
- **void pass():** Egy üres akció, ami egy akció felhasználásán kívül nem csinál semmit.
- **void myTurn():** Elvégzi az összes effekt *onTurnImpact()* függvényét, valamint csökkenti a hatásuk idejét eggyel. Csökkenti a létrehozott *Agent*-ek lejáratát is eggyel.

3.3.22 VitusDanceVirus

- **Felelősség**

Amelyik virolágusra alkalmazzák, az a következő x körre a körét azzal kezdi, hogy random lép kettőt.

- **Ősosztályok**

- Agent

- **Interfész**

-

- **Attribútumok**

- **protected int roundsLeft:** Az ágens hátralévő idejét adja meg.

- **Metódusok**
 - **Agent create():** Létrehoz egy *VitusDanceVirus*-t és ezt visszaadja a hívónak, aki aztán ezt el tudja tárolni.
 - **void onTurnImpact(Virologist):** A kiválasztott *Virologist* lebénítja adott számú körre, így nem fog tudni akciókat végrehajtani.

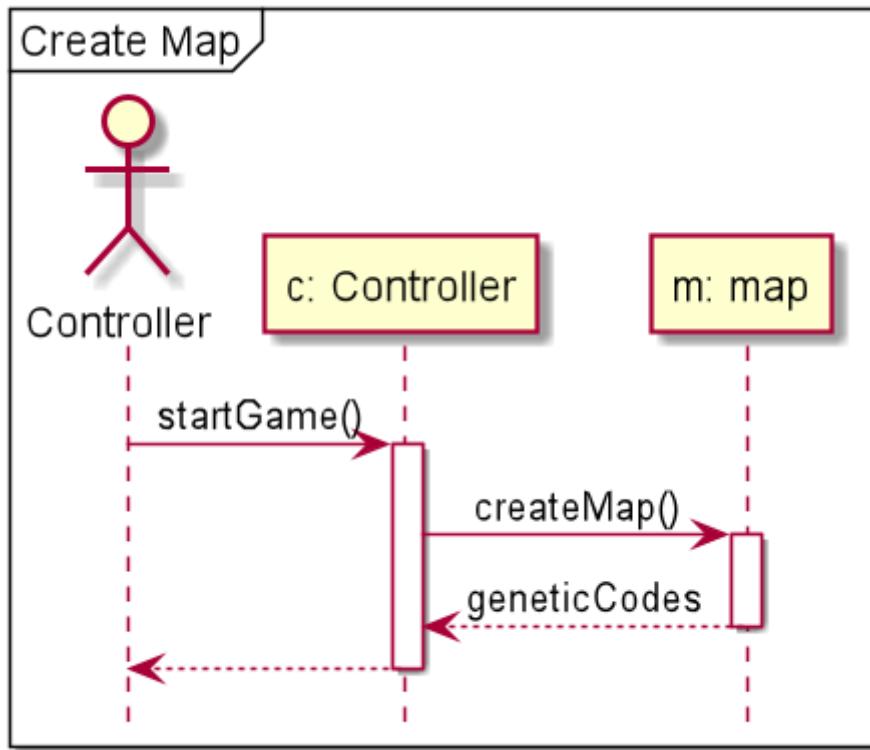
3.3.23 Warehouse

- **Felelősség**

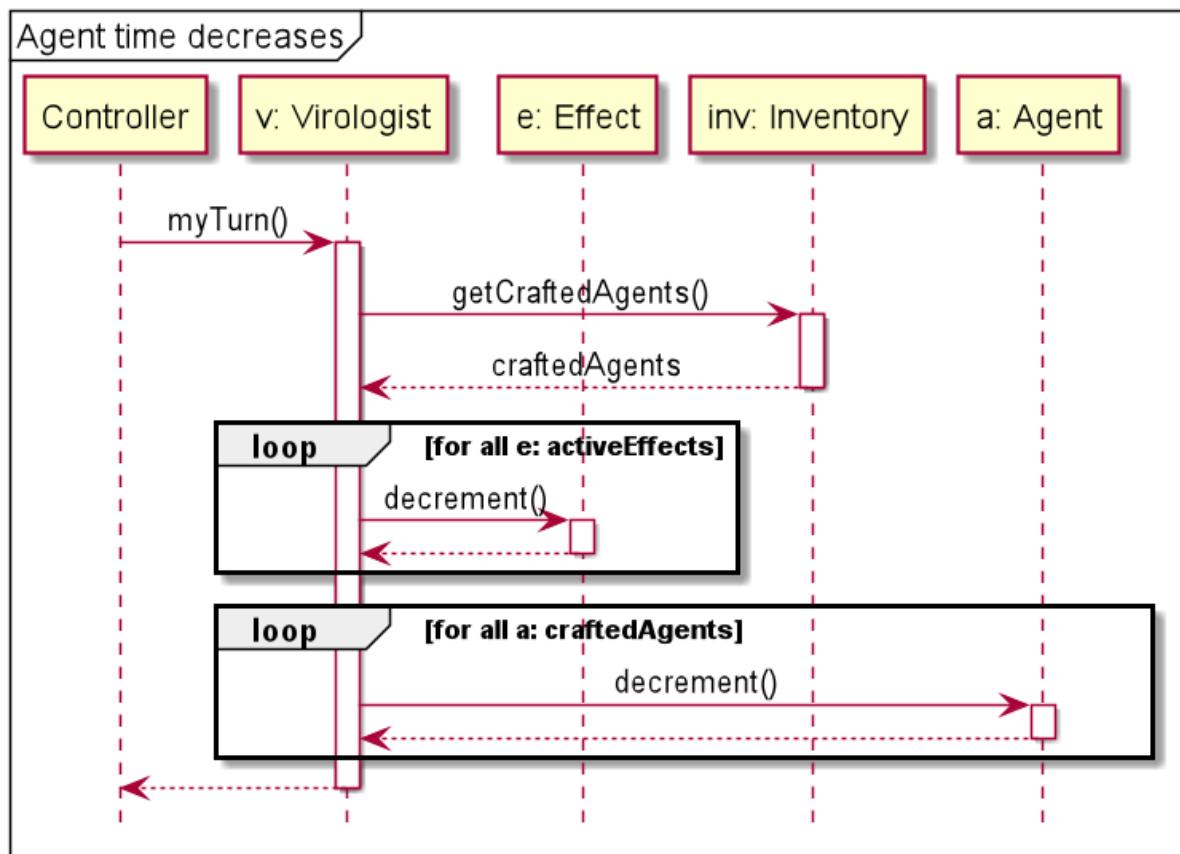
A játékban szereplő raktár pályaelem reprezentációjáért felelős. Ő felel a mezőn található *Collectable* interfészt megvalósító *Resource* objektum átadásáért a pályán tartozkodó játékosnak.
- **Ősosztályok**
 - Tile
- **Interfészek**
 -
- **Attribútumok**
 - **Resource collectable:** A Warehouse mezőn található és felvehető anyagot tárolja.
- **Metódusok**
 - **void collectItem(player: Virologist):** A mezőről felvehető *Collectable* interfészt megvalósító *Resource* objektumot átadja a paraméterként kapott *Virologist* számára.
 - **Collectable getCollectableItem():** Visszaadja a mezőn található *Collectable* interfészt megvalósító *Resource* objektumot.

Szekvencia diagramok

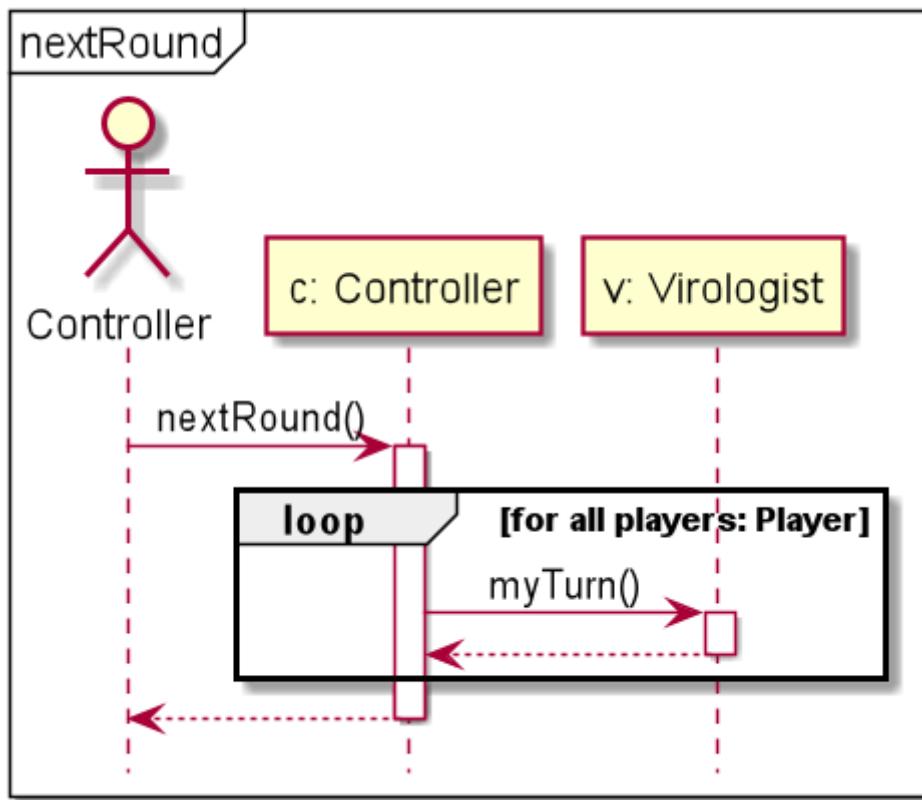
3.4.1 Controller: createMap



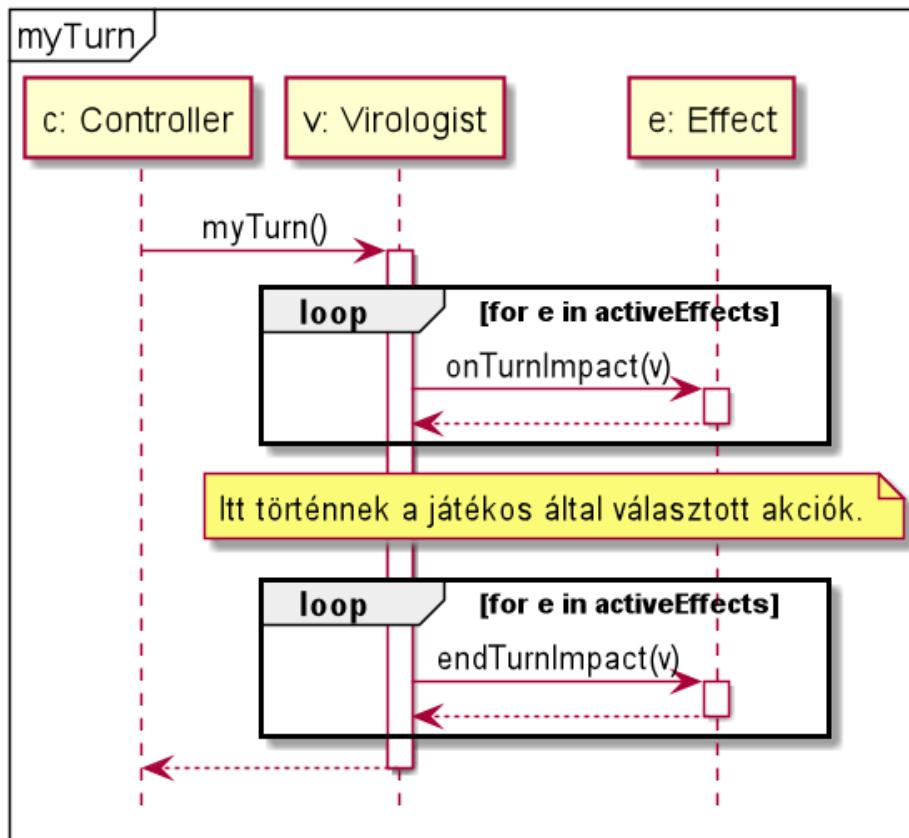
3.4.2 Controller: agentTimeDecreases



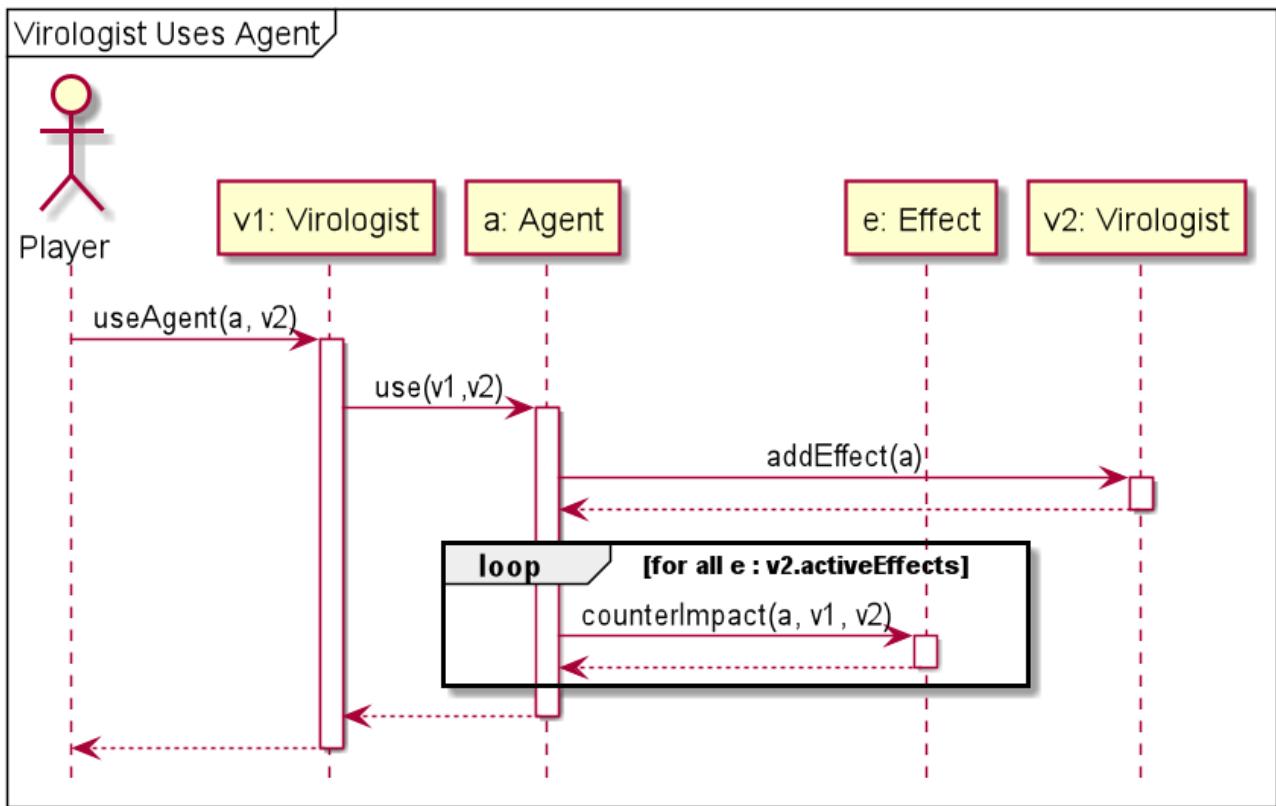
3.4.3 Controller: nextRound



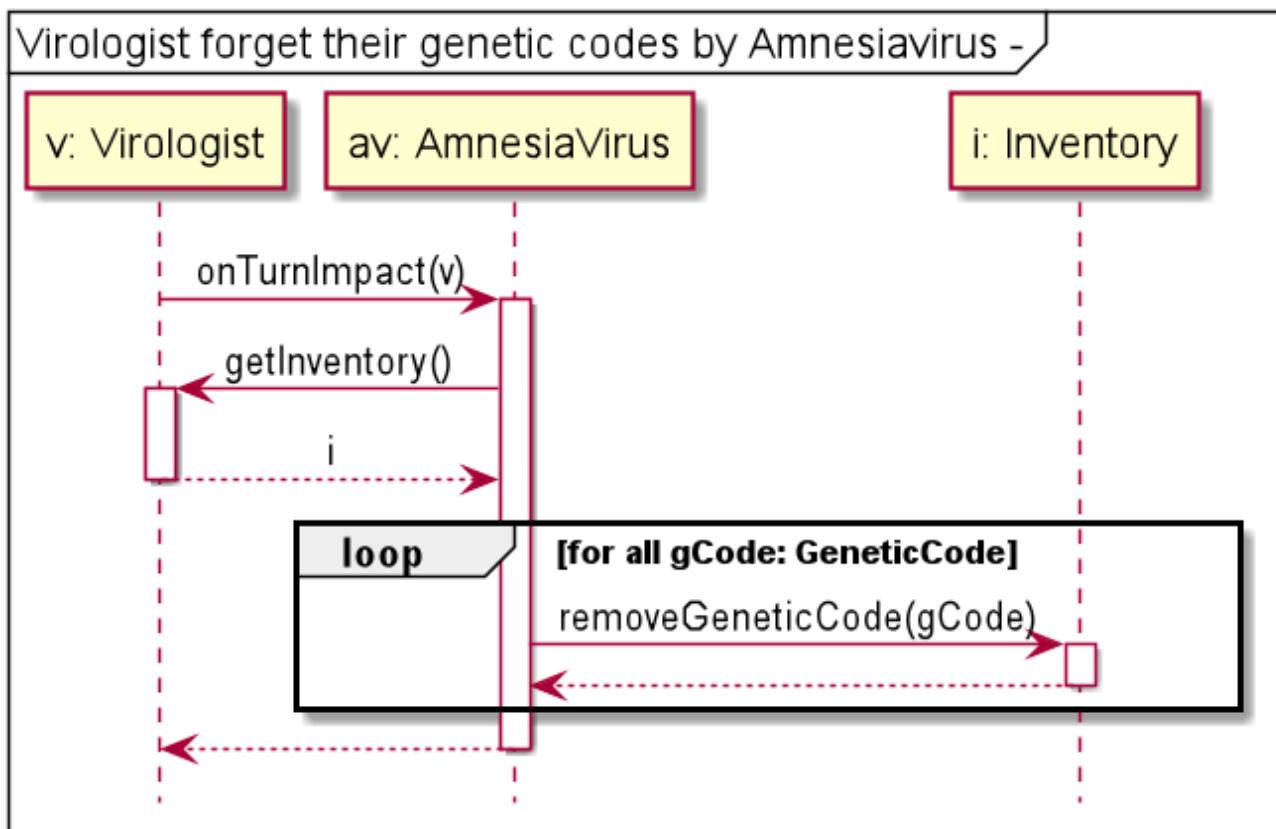
3.4.4 Player: myTurn



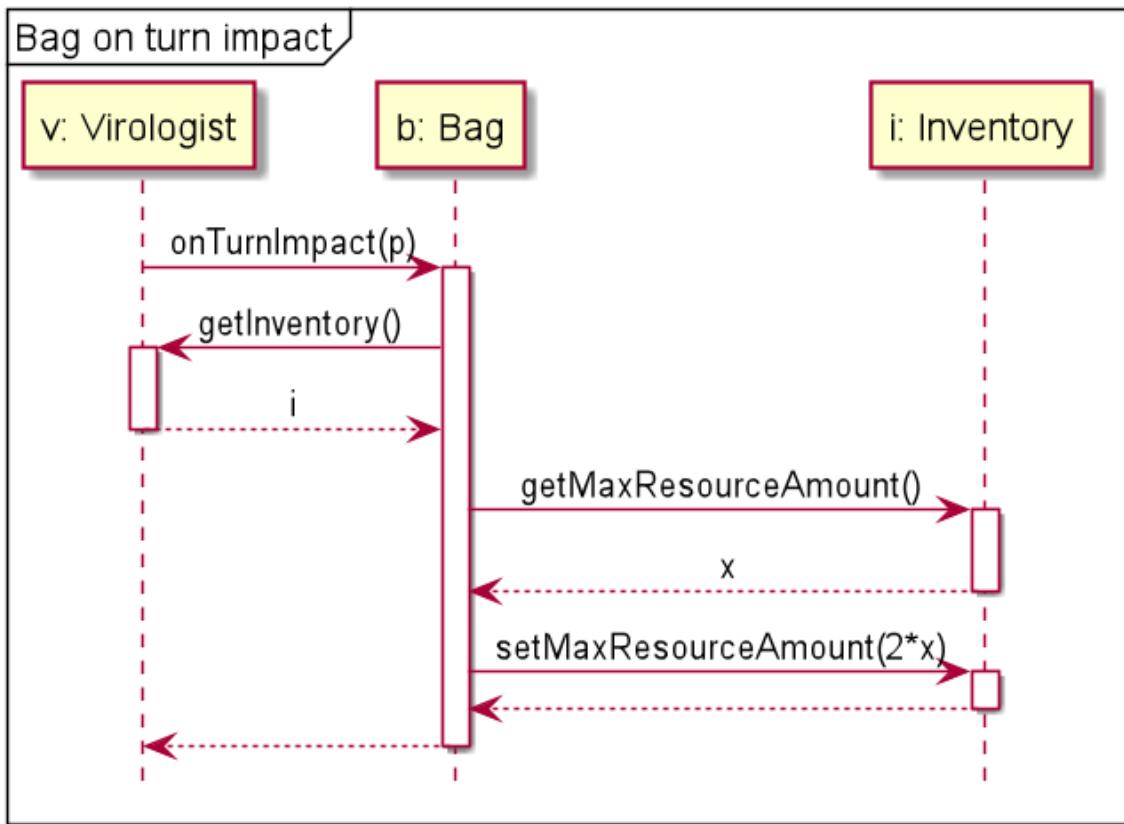
3.4.5 Player: virologistUsesAgent



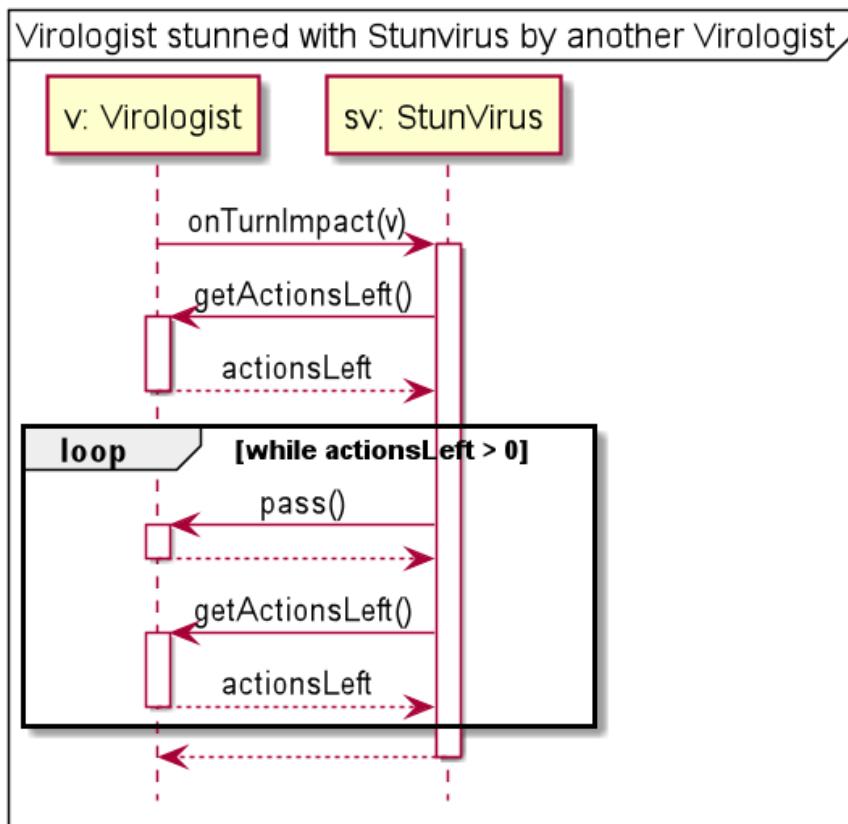
3.4.6 Virologist: amnesiaVirusOnTurnImpact



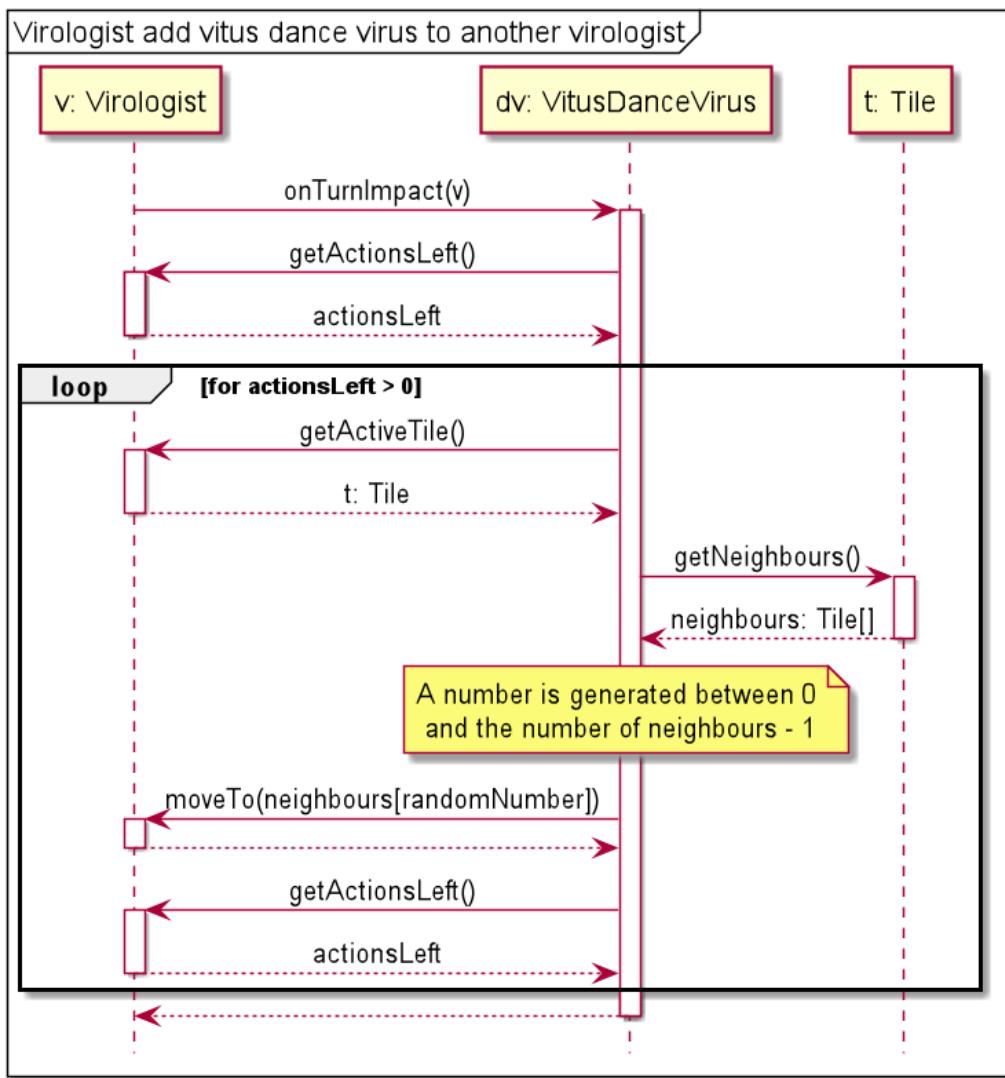
3.4.7 Virologist: bagOnTurnImpact



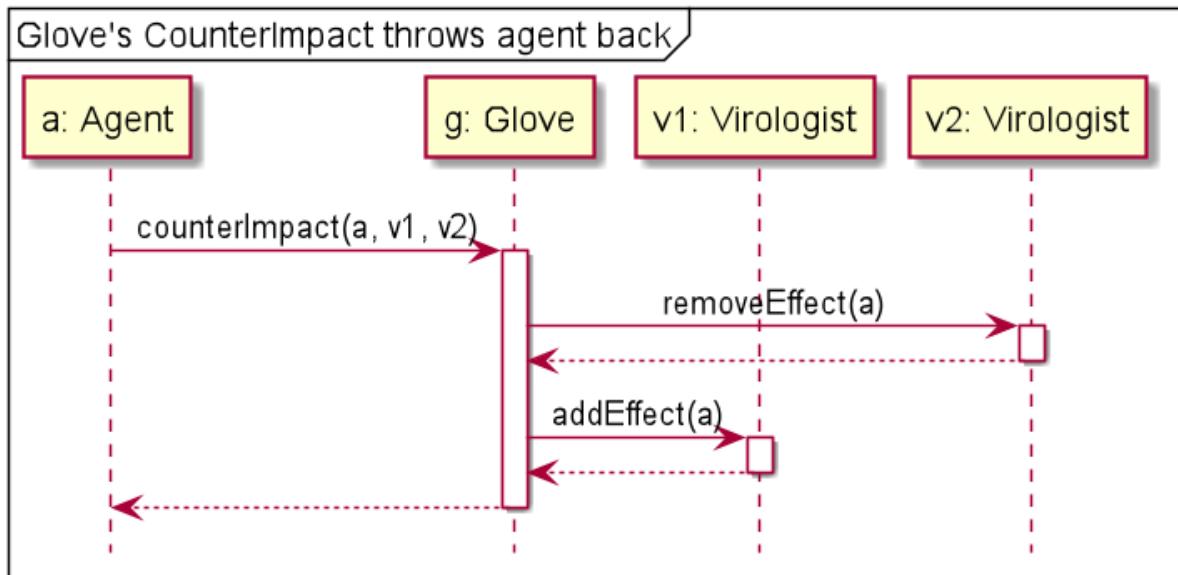
3.4.8 Virologist: stunOnTurnImpact



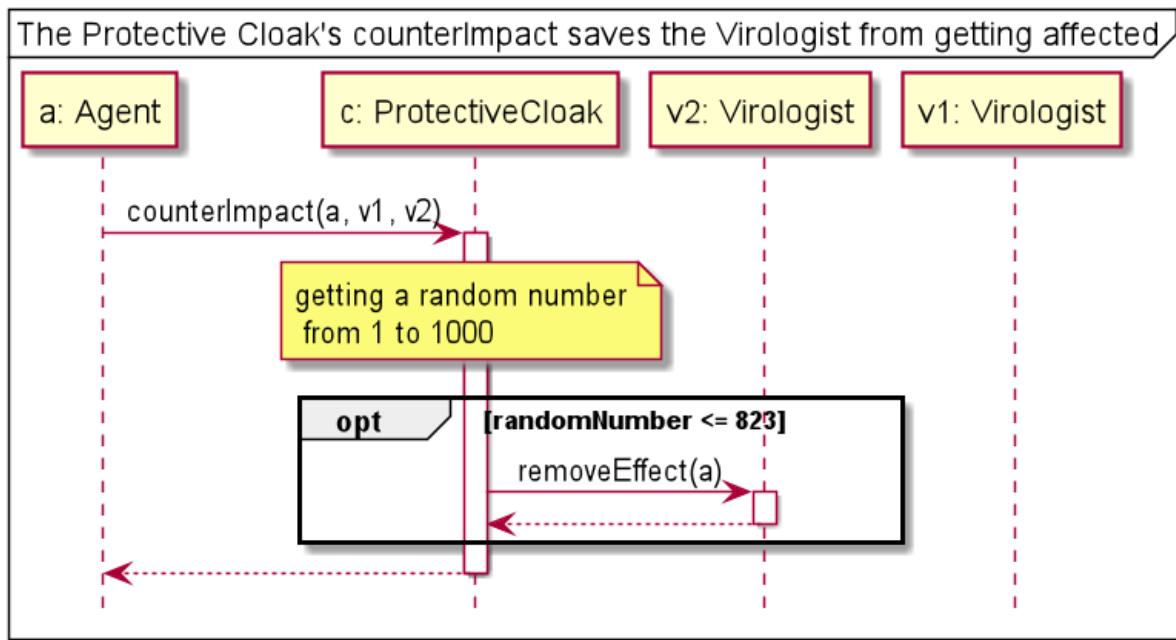
3.4.9 Virologist: vitusDanceOnTurnImpact



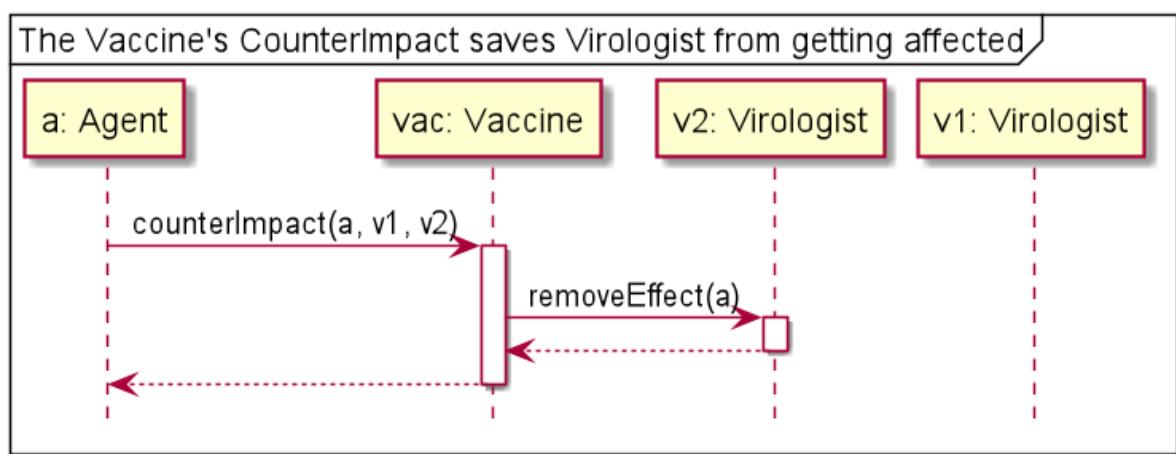
3.4.10 Agent: gloveCounterImpact



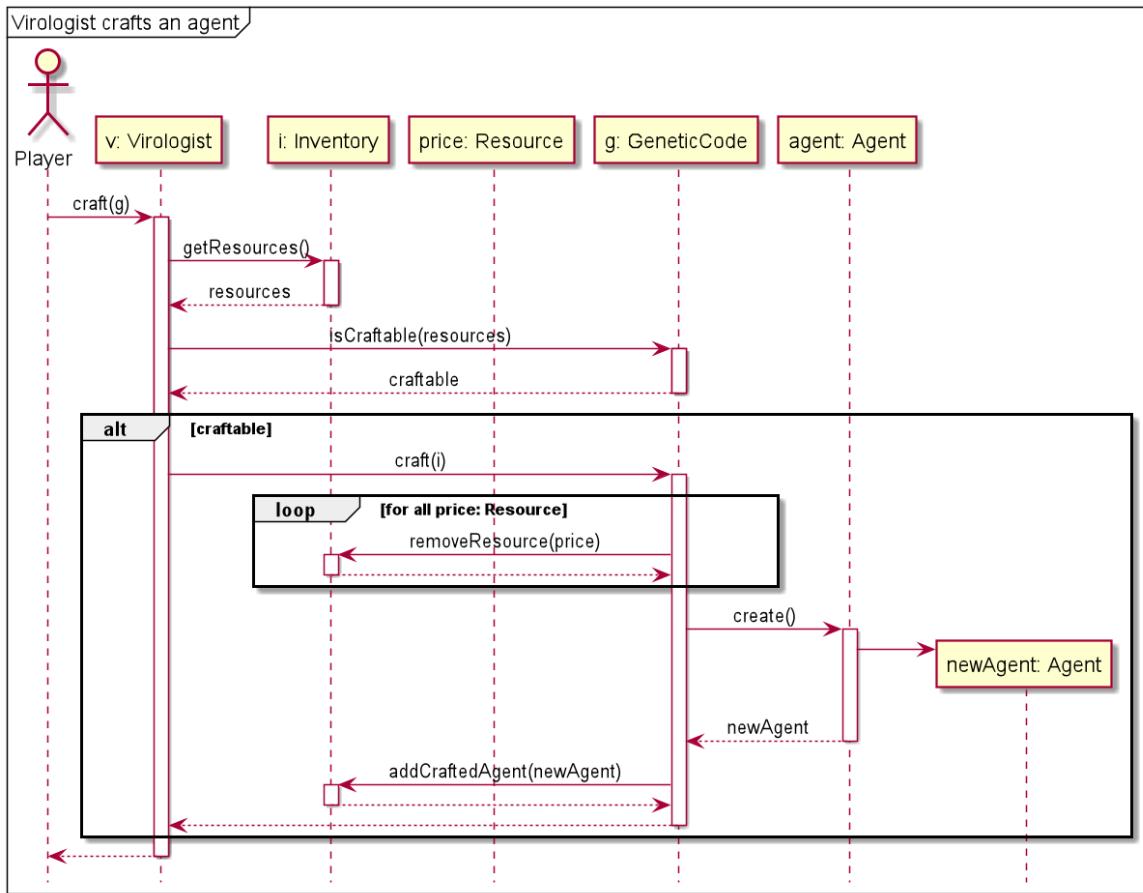
3.4.11 Agent: protectiveCloakCounterImpact



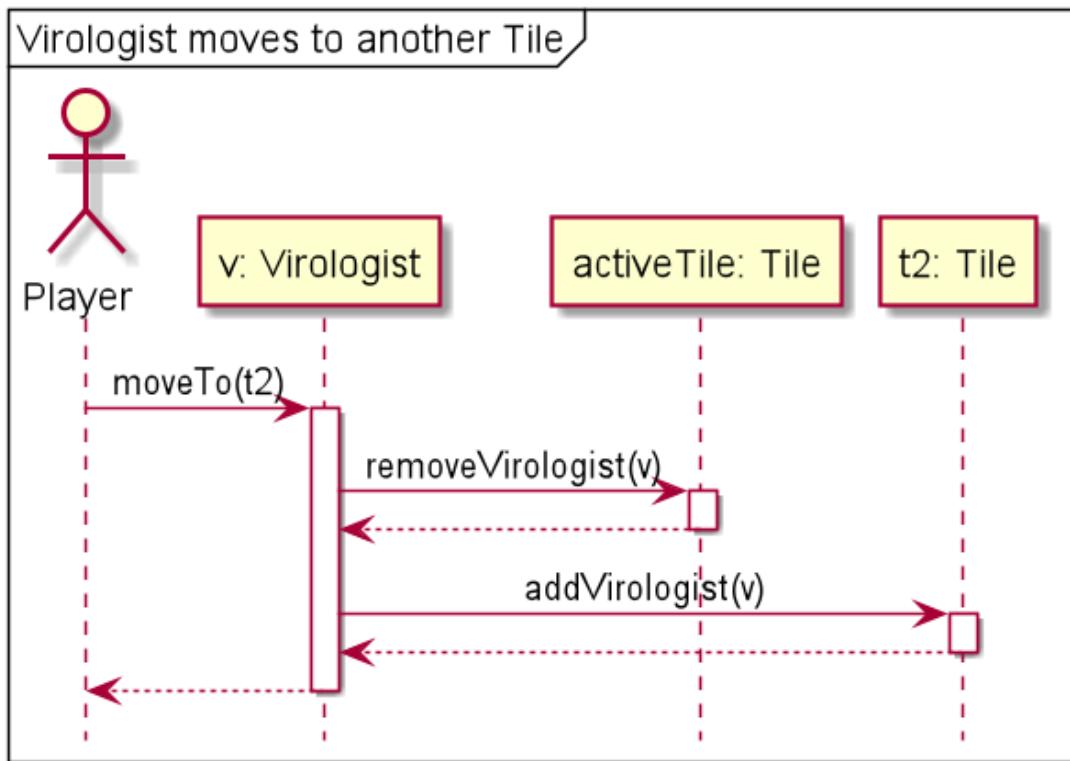
3.4.12 Agent: vaccineCounterImpact



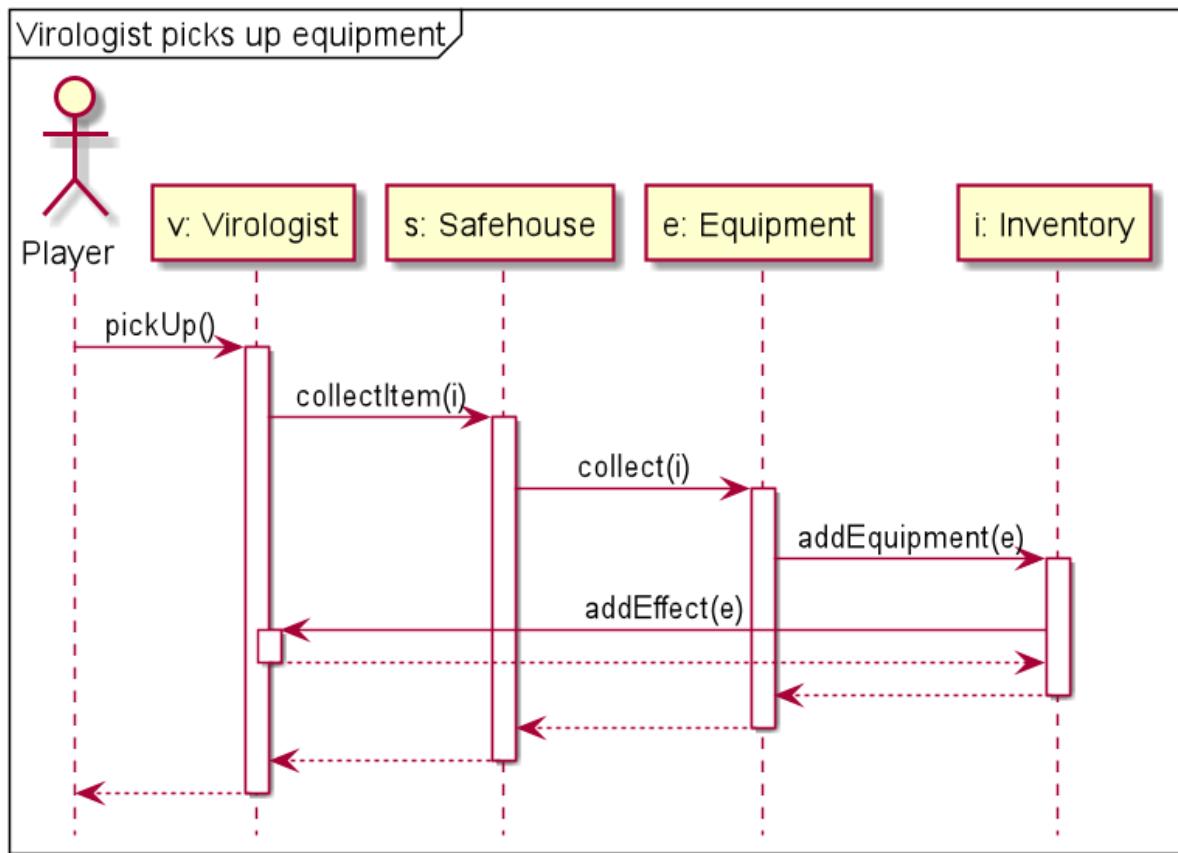
3.4.13 Player: craftAgent



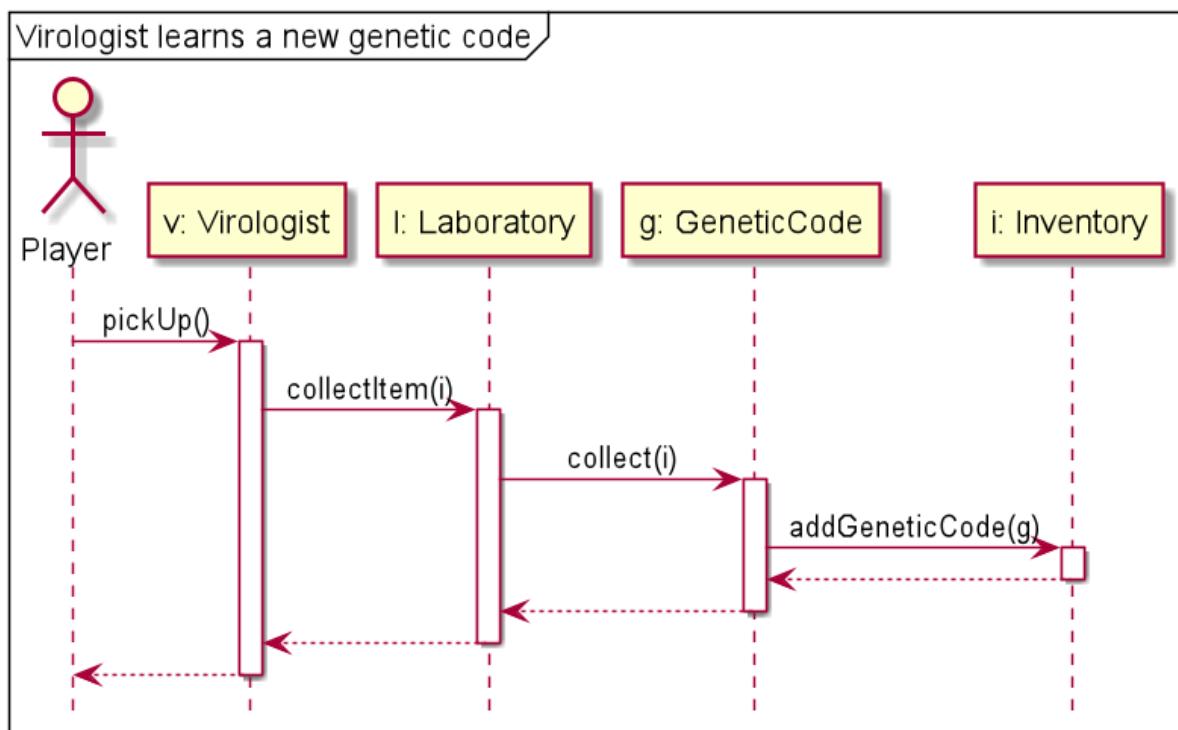
3.4.14 Player: move



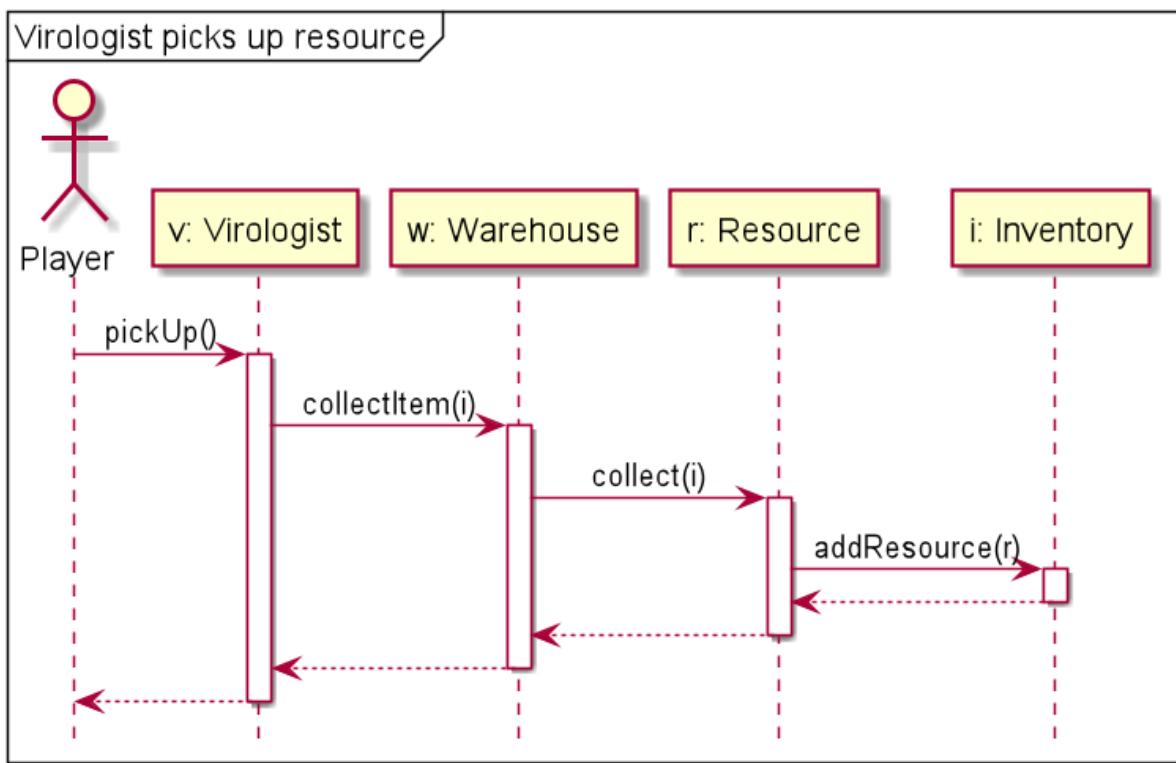
3.4.15 Player: pickUpEquipment



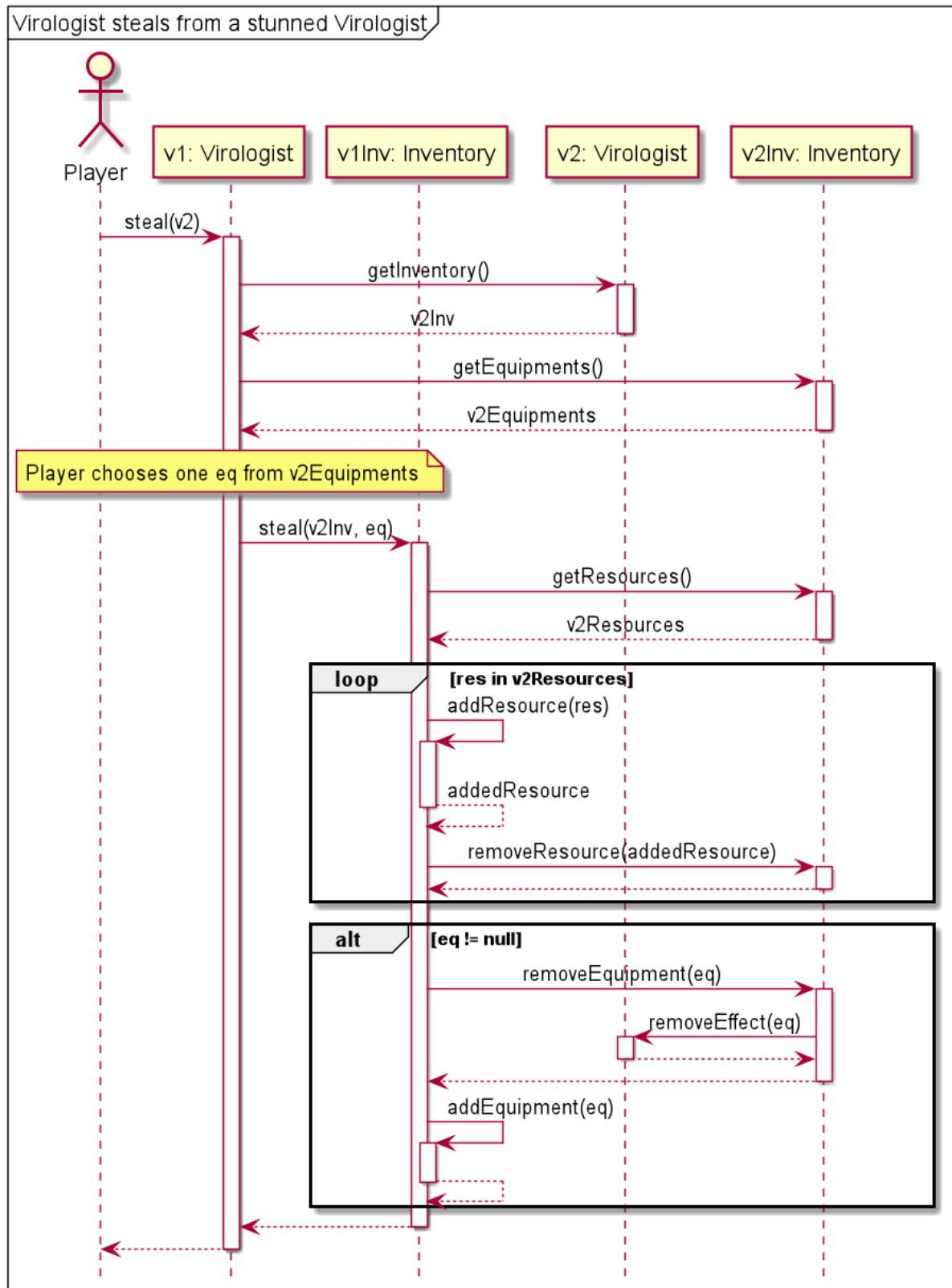
3.4.16 Player: pickUpGeneticCode



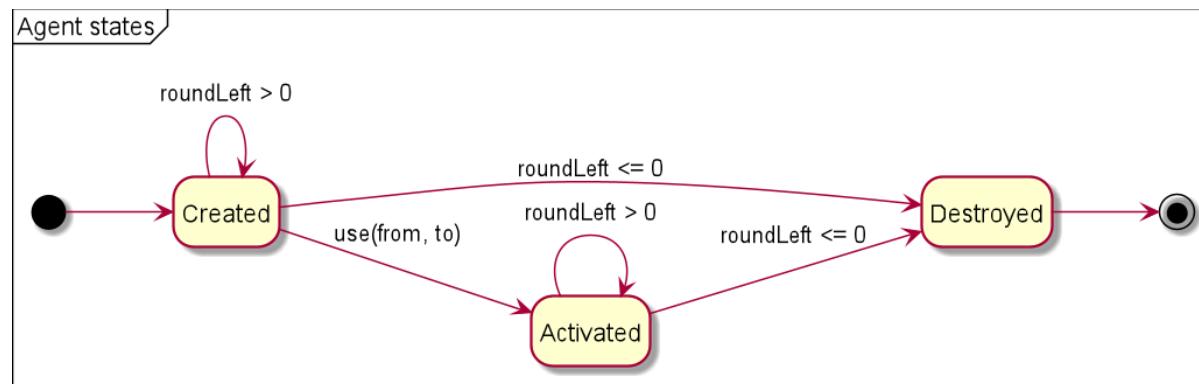
3.4.17 Player: pickUpResource



3.4.18 Player: steal



3.4 State-chartok



3.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2022.02.28. 17:00	2 óra	Fekete Száraz Jurásek Domonkos Kászonyi	Értekezlet. Döntés: A védőfelszerelések és az ágensek a hatás interfész fogják megvalósítani. Döntés: osztályok felépítése
2022.03.02. 20:00	2 óra	Domonkos	Class Diagram készítés
2022.03.02. 21:00	1 óra	Jurásek	Objektum-katalógus kifejtése
2022.03.05. 13:00	1 óra	Fekete Száraz Jurásek Domonkos Kászonyi	Értekezlet. Döntés: Szekvenciadiagramok és Osztályleírások elkészítésének felosztása. Határidő: 03.06. 20:00
2022.03.05 14:00	2 óra	Fekete	Ágens elkészítése, Genetikai kód megtanulása és Anyag felvétele szekvenciadiagrammok elkészítése.
2022.03.05. 14:00	2 óra	Jurásek	Collectable és Effect interface osztályok dokumentálása, szekvenciadiagramok tervezése
2022.03.05 15:00	4 óra	Száraz	Virológu és Inventory osztályok leírása, osztálydiagramm kiegészítése, dokumentum szerkesztése
2022.03.05 15:30	2 óra	Kászonyi	Osztálydiagramm Map, Controller, Ágensek
2022.03.05. 16:00	2 óra	Domonkos	Mozgás, Collectable objektumok felvételével kapcsolatos osztályleírások, diagrammok

2022.03.05 19:00	3 óra	Kászonyi	Szekvencia tervezése és megírása
2022.03.06 14:00	2 óra	Domonkos	Ágensek használatával kapcsolatos szekvenciadiagrammok előszítése
2022.03.06 14:00	3 óra	Fekete	Osztályleírások: GeneticCode, Resource, ResourceType
2022.03.06. 16:00	2 óra	Jurásek	4 szekvenciadiagram elkészítése, Equipment és leszármazottai, Vaccine osztályok dokumentálása
2022.03.06. 19:00	2 óra	Fekete Száraz Jurásek Domonkos Kászonyi	Értekezlet. Döntés: A hatás interface függvényei és működése.
2022.03.06 21:00	3 óra	Kászonyi	Word szerkesztése, szekvencia diagramok véglegesítése.
2022.03.06 21:00	2 óra	Száraz	Zsák és Vakcina counterImpact szekvenciadiagramok elkészítése
2022.03.06 21:00	2 óra	Domonkos	Dokumentum szerkesztése, ellenőrzése
2022.03.06 21:00	2 óra	Fekete	Dokumentum szerkesztése, ellenőrzése
2022.03.06 21:00	2 óra	Jurásek	Dokumentum szerkesztése, ellenőrzése
2022.03.06 21:00	2 óra	Száraz	Dokumentum szerkesztése, ellenőrzése

5. Szkeleton tervezése

52 – *service_unavailable*

Konzulens:
Micskei Zoltán

Csapattagok

Fekete Sámuel	GJ8J3A	feketesamu@gmail.com
Száraz Dániel	GT5X34	szdani2012@gmail.com
Jurásek Jónás	PH4QFK	jurasekjonas12@gmail.com
Domonkos Ádám	CWGYWC	domonkosadam01@gmail.com
Kászonyi Zsombor Marcell	DCE2Q1	kaszonyirelli@gmail.com

2022.02.23.

4. Szkeleton tervezése

Változtatások:

Az agent use függvényét módosítottuk (4.4.14 virologistUsesAgent), a counterImpact-ek csak akkor hívódnak meg, ha nem magára kenné fel a virológus az ágenset, ez így a játék szempontjából logikusabb, valamint az activeEffect-eket előbb kérdezi le, mielőtt hozzáadná az újat, ezt azért kellett felcserélni, hogy a Vakcinát például fel tudja kenni magára a virológus.

Továbbá a steal függvényben a felszerelések lopása történik meg először, valamint az ellopott felszerelésen lefut az onTurnImpact, hogy rögtön legyen hatása. Bekerült még egy újabb feltétel a felszerelés lopásához: csak akkor mehet végbe, ha a lopó játékosnak kevesebb, mint 3 védőfelszerelése van.

A Virologist osztálynak van egy statikus mezője, amin keresztül a Controllert tudja elérni. Erre genetikai kód megtanulása után van szükség, hogy a kontroller ellenőrizze, megnyerte-e a játékot az adott játékos.

4.1 A szkeleton modell valóságos use-case-ai

4.1.1 Use-case diagram



4.1.2 Use-case leírások

Use-case neve	moveToEmptyTile
Rövid leírás	A játékos egy üres mezőre lép.
Aktorok	Player
Forgatókönyv	<i>Kiindulási feltétel: a virológus egy üres mezőn áll.</i> 1. A játékos egy szomszédos üres mezőre lép

Use-case neve	moveToLaboratory
Rövid leírás	A játékos egy laboratórium mezőre lép.
Aktorok	Player
Forgatókönyv	<i>Kiindulási feltétel: a virológus egy üres mezőn áll.</i> 1. A játékos egy szomszédos laboratórium mezőre lép

Use-case neve	moveToSafehouse
Rövid leírás	A játékos egy óvóhely mezőre lép.
Aktorok	Player
Forgatókönyv	<i>Kiindulási feltétel: a virológus egy üres mezőn áll.</i> 1. A játékos egy szomszédos óvóhely mezőre lép

Use-case neve	moveToWarehouse
Rövid leírás	A játékos egy raktár mezőre lép.
Aktorok	Player
Forgatókönyv	<i>Kiindulási feltétel: a virológus egy üres mezőn áll.</i> 1. A játékos egy szomszédos raktár mezőre lép

Use-case neve	pickUpBag
Rövid leírás	A játékos felvesz egy zsák felszerelést egy óvóhely mezőről.
Aktorok	Player
Forgatókönyv	<i>Kiindulási feltétel: Virológus egy olyan óvóhely mezőn áll, ahol a zsák a felvehető felszerelés.</i> 1. Játékos megpróbálja felvenni a zsákat. 2/A. Ha a játékosnál még van hely, és fel tudja venni a zsákat akkor a zsák egy másolata bekerül a virológus felszerelései közé. 2/B. Ha a játékos felszerelései között már nincs hely, akkor nem tudja felvenni. 3/A. A zsák hatása érvényesül a virológuson: Megnöveli a maximális anyagmennyiséget

Use-case neve	pickUpCloak
Rövid leírás	A játékos felvesz egy védőköpeny felszerelést egy óvóhely mezőről.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Virológus egy olyan óvóhely mezőn áll, ahol a védőköpeny a felvehető felszerelés.</i></p> <p>1. Játékos megpróbálja felvenni a védőköpenyt.</p> <p>2/A. Ha a játékosnál még van hely, és fel tudja venni a védőköpenyt akkor a védőköpeny egy másolata bekerül a virológus felszerelései közé.</p> <p>2/B. Ha a játékos felszerelései között már nincs hely, akkor nem tudja felvenni.</p> <p>3/A. A védőköpeny hatása érvényessül a virológuson: 82,3% eséllyel megvédi a jövőben rákent ágensektől.</p>

Use-case neve	pickUpGlove
Rövid leírás	A játékos felvesz egy kesztyű felszerelést egy óvóhely mezőről.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Virológus egy olyan óvóhely mezőn áll, ahol a kesztyű a felvehető felszerelés.</i></p> <p>1. Játékos megpróbálja felvenni a kesztyűt.</p> <p>2/A. Ha a játékosnál még van hely, és fel tudja venni a kesztyűt akkor a kesztyű egy másolata bekerül a virológus felszerelései közé.</p> <p>2/B. Ha a játékos felszerelései között már nincs hely, akkor nem tudja felvenni.</p> <p>3/A. A kesztyű hatása érvényesül a virológuson: A jövőben rákent ágenseket visszakeni a támadóra.</p>

Use-case neve	pickUpResource
Rövid leírás	A játékos felvesz egy anyagot egy raktár mezőről.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Virológus egy raktár mezőn áll</i></p> <p>1. Játékos felveszi az anyagot</p> <p>2. Hozzáad egy adott mennyiségű anyagot a virológushoz.</p> <p>3. A virológus anyagmennyisége növekedik a felvett mennyiségű anyaggal, de nem haladja meg a maximális mennyiséget.</p>

Use-case neve	learnGeneticCode
Rövid leírás	A játékos megtanul egy genetikai kódot egy laboratórium mezőn.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel:</i> A játékos egy olyan laboratórium mezőn áll, ahol még nem tanult korábban genetikai kódot.</p> <p>1. A laboratórium mező genetikai kódja eltárolódik a virológus felszerelésében.</p> <p>2/A. Ha ez volt az utolsó genetikai kód, amit nem tanult meg ez a játékos, akkor megnyerte a játékot.</p> <p>2/B. Egyébként folytatódik a játék.</p>

Use-case neve	craftVaccine
Rövid leírás	A játékos megpróbál egy vakcinát elkészíteni.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel:</i> A virológus megtanulta a Vakcinához tartozó Genetikai kódot.</p> <p>1/A. A játékos elkészíti a vakcinát.</p> <p>1/B. A játékos virológusának nincs elég anyagkészlete, ezért nem tudja elkészíteni a vakcinát.</p>

Use-case neve	craftVitusDance
Rövid leírás	A játékos megpróbál egy vitustáncvírust elkészíteni.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel:</i> A virológus megtanulta a vitustáncvírushoz tartozó genetikai kódot.</p> <p>1/A. A játékos elkészíti a vitustáncvírust.</p> <p>1/B. A játékos virológusának nincs elég anyagkészlete, ezért nem tudja elkészíteni a vitustáncvírust.</p>

Use-case neve	craftStun
Rövid leírás	A játékos megpróbál egy bénítóvírust elkészíteni.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel:</i> A virológus megtanulta a bénítóvírushoz tartozó genetikai kódot.</p> <p>1/A. A játékos elkészíti a bénítóvírust.</p> <p>1/B. A játékos virológusának nincs elég anyagkészlete, ezért nem tudja elkészíteni a bénítóvírust.</p>

Use-case neve	craftAmnesia
Rövid leírás	A játékos megpróbál egy amnéziavírust elkészíteni.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel:</i> A virológus megtanulta az amnéziavírushoz tartozó genetikai kódot.</p> <p>1/A. A játékos elkészíti az amnéziavírust.</p> <p>1/B. A játékos virológusának nincs elég anyagkészlete, ezért nem tudja elkészíteni az amnéziavírust.</p>

Use-case neve	useAmnesiaVirus
Rövid leírás	A játékos egy virológusra felkeni az amnézia vírust.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Egy virológus, akire fel lehet kenni a vírust (lehet saját maga is).</i></p> <p>1/A. A játékos felkeni a vírust a választott virológusra.</p> <p>1/B. A játékos felkenné a vírust, de kivédi.</p> <p>1/C. A játékos felkenné a vírust, de visszakenik rá.</p>

Use-case neve	useStunVirus
Rövid leírás	A játékos egy virológusra felkeni a bénító vírust.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Egy virológus, akire fel lehet kenni a vírust (lehet saját maga is).</i></p> <p>1/A. A játékos felkeni a vírust a választott virológusra.</p> <p>1/B. A játékos felkenné a vírust, de kivédi.</p> <p>1/C. A játékos felkenné a vírust, de visszakenik rá.</p>

Use-case neve	useVaccine
Rövid leírás	A játékos egy virológusra felkeni a vakcinát.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Egy virológus, akire fel lehet kenni a vírust (lehet saját maga is).</i></p> <p>1/A. A játékos felkeni a vírust a választott virológusra.</p> <p>1/B. A játékos felkenné a vírust, de kivédi.</p> <p>1/C. A játékos felkenné a vírust, de visszakenik rá.</p>

Use-case neve	useVitusDanceVirus
Rövid leírás	A játékos egy virológusra felkeni a vitus tánc vírust.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Egy virológus, akire fel lehet kenni a vírust (lehet saját maga is).</i></p> <p>1/A. A játékos felkeni a vírust a választott virológusra.</p> <p>1/B. A játékos felkenné a vírust, de kivédi.</p> <p>1/C. A játékos felkenné a vírust, de visszakenik rá.</p>

Use-case neve	robVirologist
Rövid leírás	Az egyik játékos lop egy másik játékostól, aki le van bénulva.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: A virológus egy olyan mezőn áll, amin van legalább egy bénult játékos (önmagán kívül).</i></p> <p>1. A játékos a mezőn álló bénult játékosok közül választhat, hogy kitől lopjon.</p> <p>2/A. Ha a lopó virológusnak még kettőnél kevesebb védőfelszerelése van, és annak a játékosnak, akitől lop van legalább egy védőfelszerelése, akkor ezek közül választhat egyet, amit ellop.</p> <p>2/B. Egyébként nem tud védőfelszerelést lopni.</p> <p>3. A lopó játékos megkapja a támadott játékos anyagkészletéből azt a maximális mennyiséget, amivel még nem lépi túl az anyagkészletének felső korlátját.</p>

Use-case neve	forgetCodesViaAmnesiaVirus
Rövid leírás	Amnéziavírussal kent játékos vélegesen elfelejt az összes eddig megtanult genetikai kódját.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: A viroláguson van egy aktív amnéziavírus.</i></p> <p>1. A játékos köre elején az összes eltárolt genetikai kód törlődik.</p> <p>2. A játékos kijátszhat különböző akciókat, megy tovább a játék.</p>

Use-case neve	enlargeTheBag
Rövid leírás	Az a játékos, akinek van egy zsák az védőfelszerelései között, több anyagot tud felvenni.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: A virolágusnak van egy zsák védőfelszerelése.</i></p> <p>1. A játékos köre elején az Inventory kapacitása megkétszereződik.</p> <p>2. Lopás vagy anyagfelvételkor ezt a megnövelt korlátot kell betartania az anyag mennyiségeknek.</p> <p>3. A kör végén a kapacitás feleződik, azaz visszaáll a régi értékre.</p>

Use-case neve	stunnedPlayerMissesTurn
Rövid leírás	Bénítóvírussal kent játékos nem tud semmilyen akciót végrehajtani a körében.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: A virológuson van egy aktív bénítóvírus.</i></p> <p>1. A játékos köre elején automatikusan addig passzol, amíg el nem fogynak az akciói.</p> <p>2. A játékoson aktív többi hatás ilyenkor is érvényes, idejük fogy.</p> <p>3. A következő játékos kerül sorra.</p>

Use-case neve	playerVitusDances
Rövid leírás	Vitustánc vírussal kent játékos automatikusan véletlenszerű lépéstre használja el az összes akcióját.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: A virológuson van egy aktív vitustánc vírus.</i></p> <p>1. A játékos köre elején automatikusan addig lép egy véletlenszerű, szomszédos mezőre, amíg el nem fogynak az akciói.</p> <p>2. A játékoson aktív többi hatás ilyenkor is érvényes, idejük fogy.</p> <p>3. A következő játékos kerül sorra.</p>

4.2 A szkeleton kezelői felületének terve, dialógusok

4.2.1 Szkeleton program felépítése

A program indításakor először megjeleníti a lehetséges forgatókönyveket, ami közül a felhasználónak választania kell a program bemeneti követelményei szerint.

Van egy Initializer singleton osztály, aminek statikus metódusai végzik az összes kiírást és a bemenetek kezelését. Az egyes use-case-ekben az objektumok inicializálásakor minden objektumhoz tárol egy egyedi azonosítót, ez meg fog jelenni, amikor az objektum metódusai meghívódnak.

4.2.2 Szkeleton program kimenete

Kiiratás során az első kiírt sor a kiíró osztály neve, azonosítója és a függvény, amit hívtak rajta paraméterekkel. A program kiírja a meghívott függvény nevét. Amennyiben szükség van valami döntésre a felhasználótól, akkor bekéri a bemenetet.

A kimenetek formátuma:

Függvényhívás: [Osztálynév, Azonosító] [metódus neve(paraméterek)]

Visszatérés: return [Osztálynév, Azonosító] [visszatérési érték]

Kérdés felhasználótól: {Kérdés(y/n)}: válasz

A függvények tabulálása a következőképpen történik:

Függvényhívás után egy tabulátornyi helyel beljebb kezdődik a következő sor egészen a visszatérésig. A kérdések minden tabulálás nélkül történnek.

Pl.

```
[Virologist, v] [moveTo(tile)]
[EmptyTile, tile] [removeVirologist(v)]
return [EmptyTile, tile] [void]
[EmptyTile, newtile] [addVirologist(v)]
return [EmptyTile, newtile] [void]
return [Virologist, v] [void]
```

4.2.3 Szkeleton program bemenete

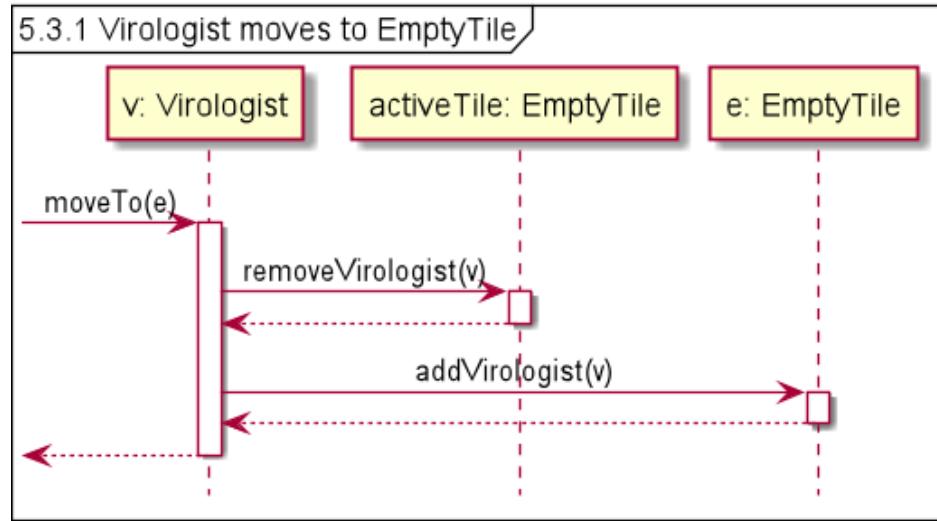
A szkeleton program az eldöntendő kérdésekre y/n bemenetet fogad el és kezel. Több válaszlehetőség esetén a program által megjelenített sorszámot fogadja el a program érvényes bemenetként.

Pl: Milyen védfelszerelései legyenek?

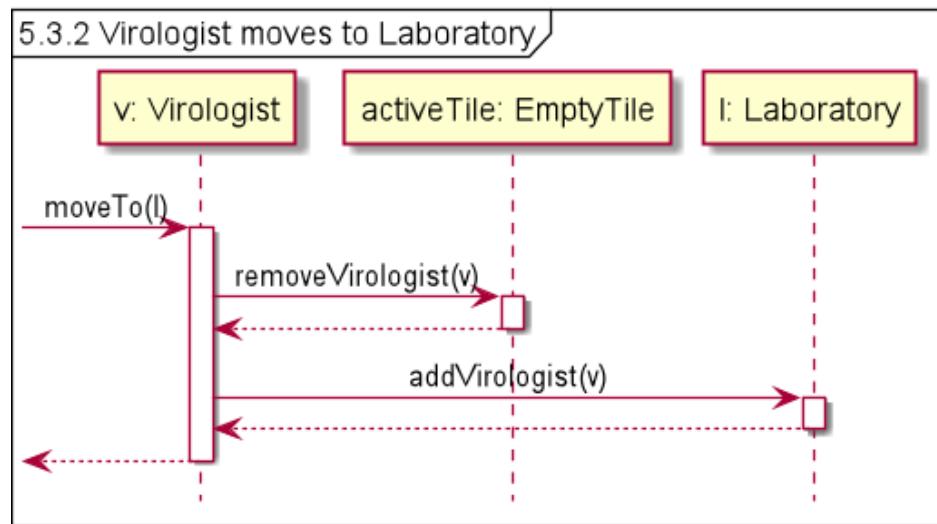
1. Glove
2. Cloak
3. both
4. neither

4.3 Szekvencia diagramok a belső működésre

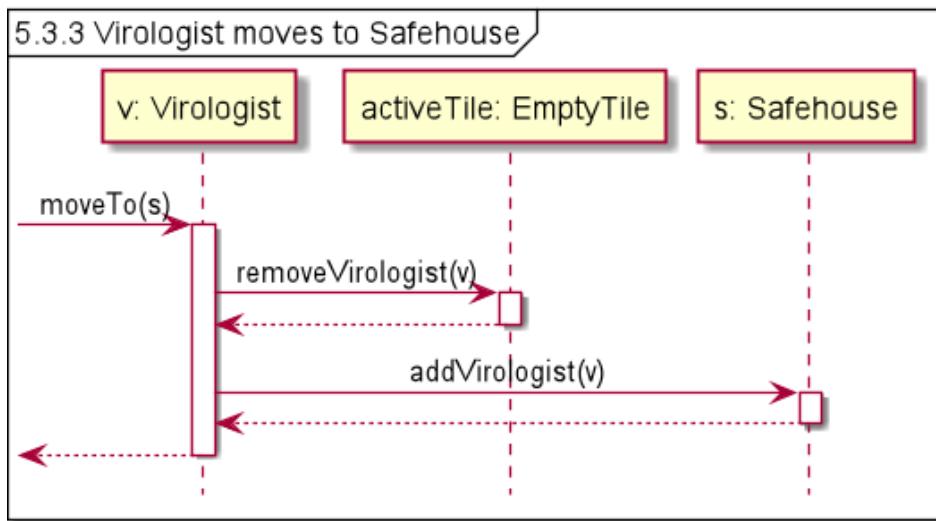
4.3.1 moveToEmptyTile



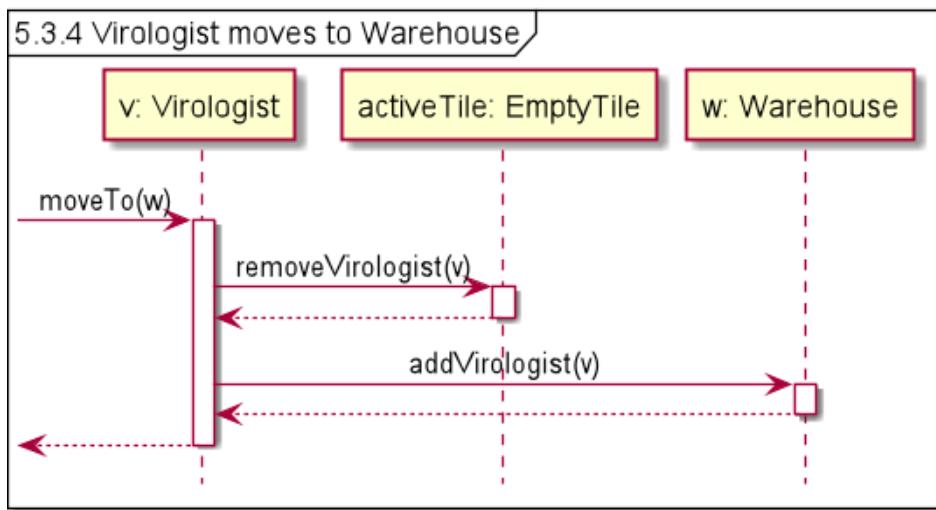
4.3.2 moveToLaboratory



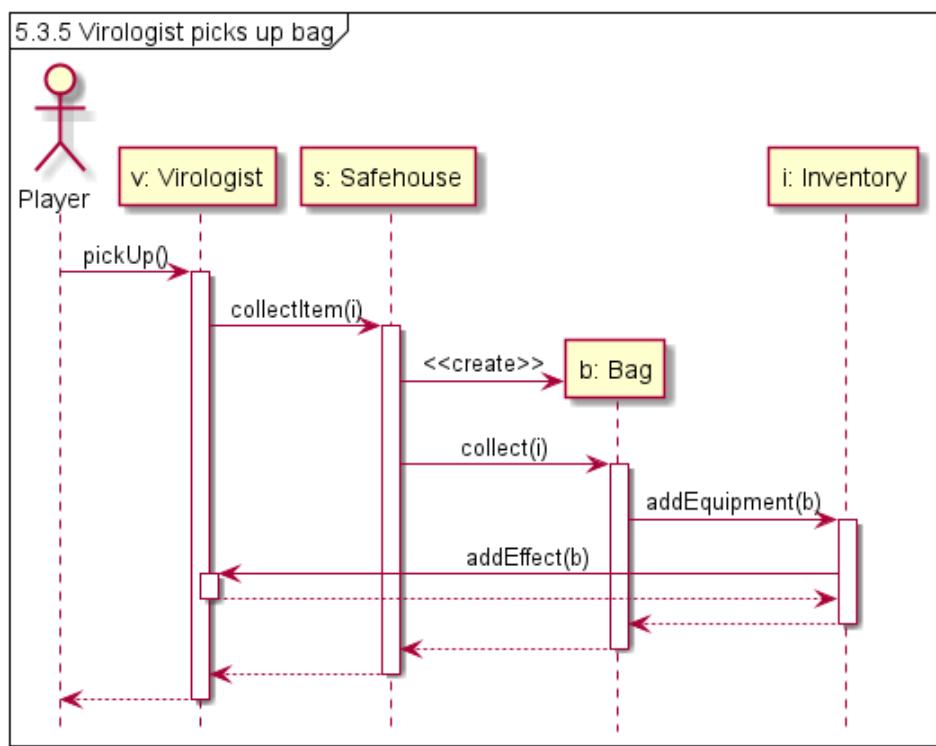
4.3.3 moveToSafehouse



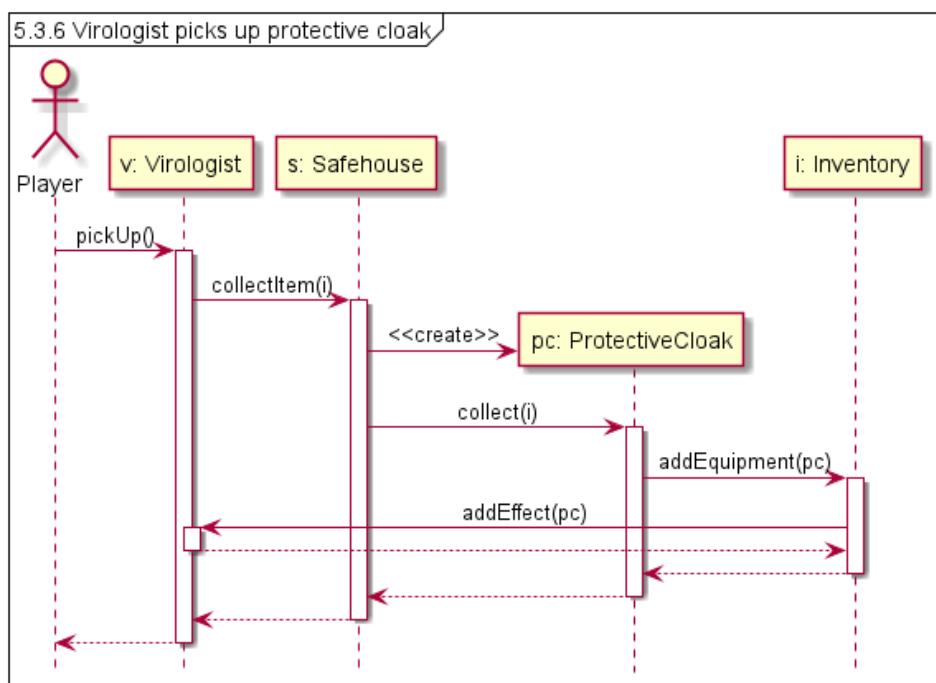
4.3.4 moveToWarehouse



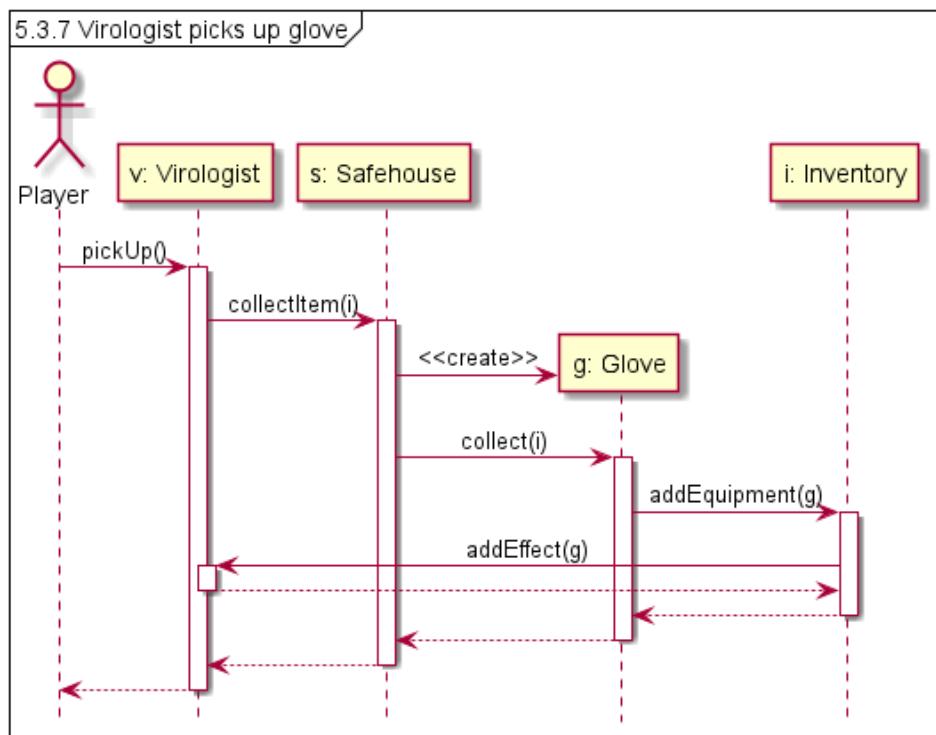
4.3.5 pickUpBag



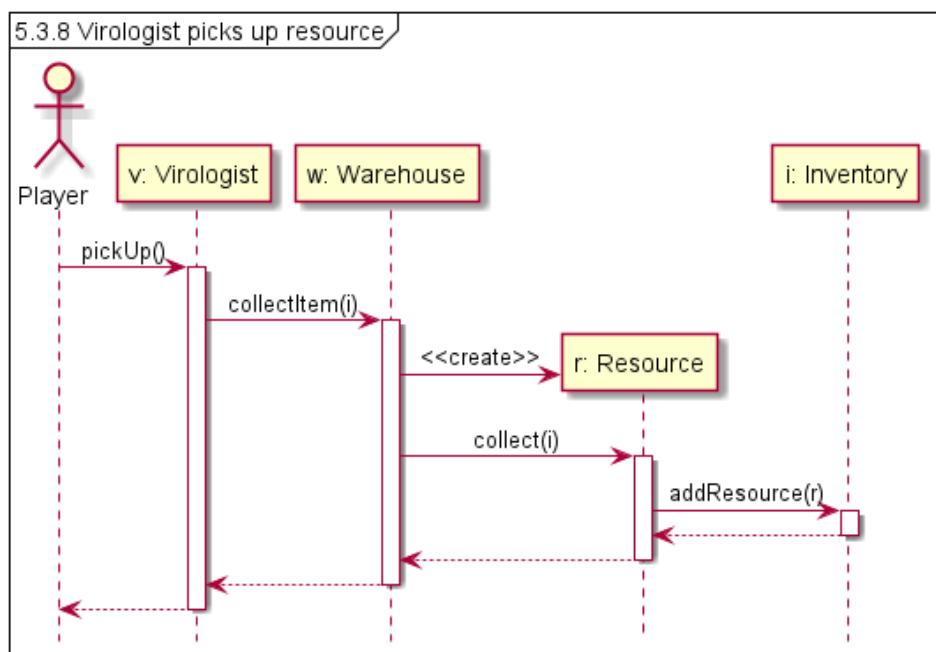
4.3.6 pickUpCloak



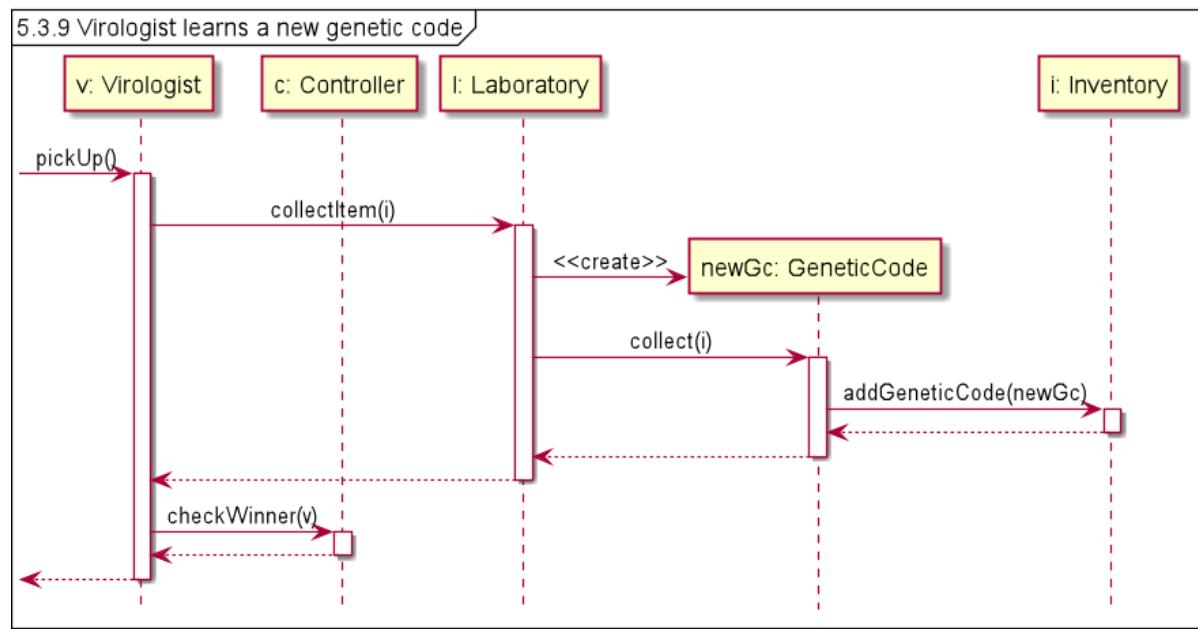
4.3.7 pickUpGlove



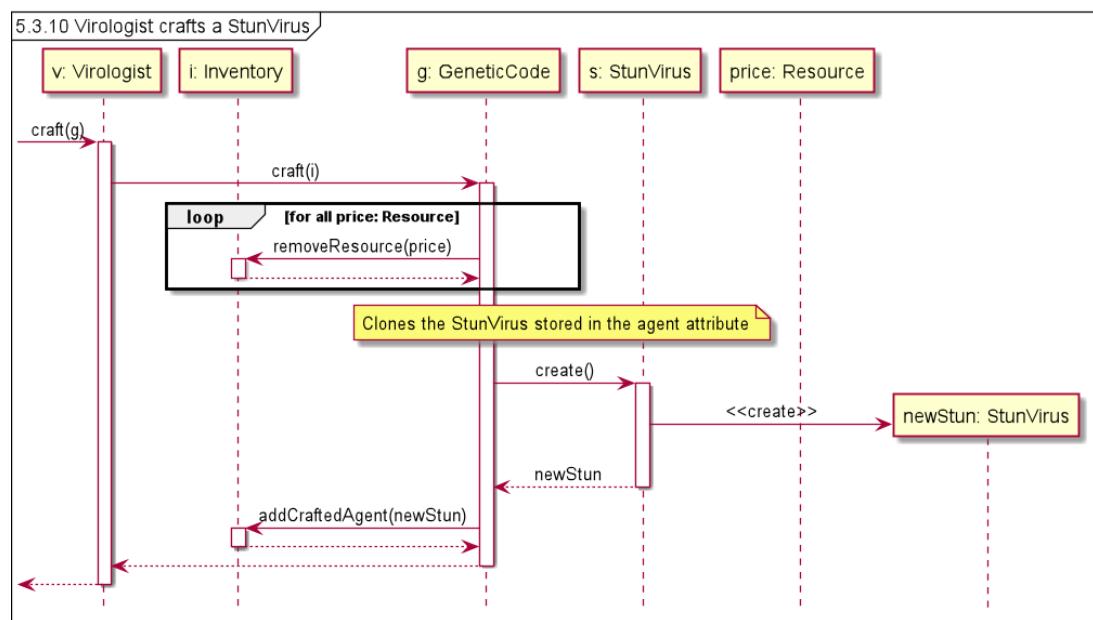
4.3.8 pickUpResource



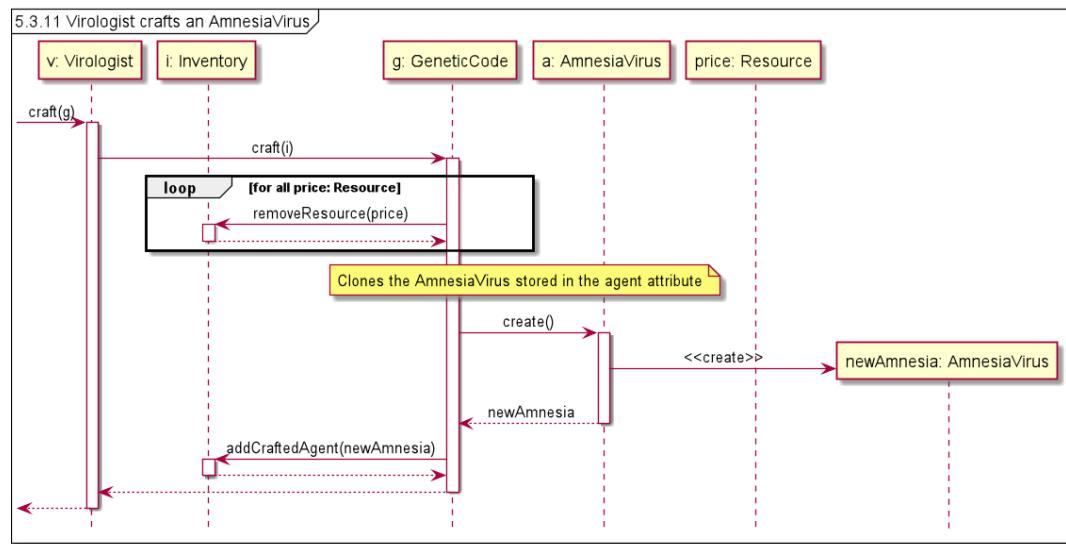
4.3.9 learnGeneticCode



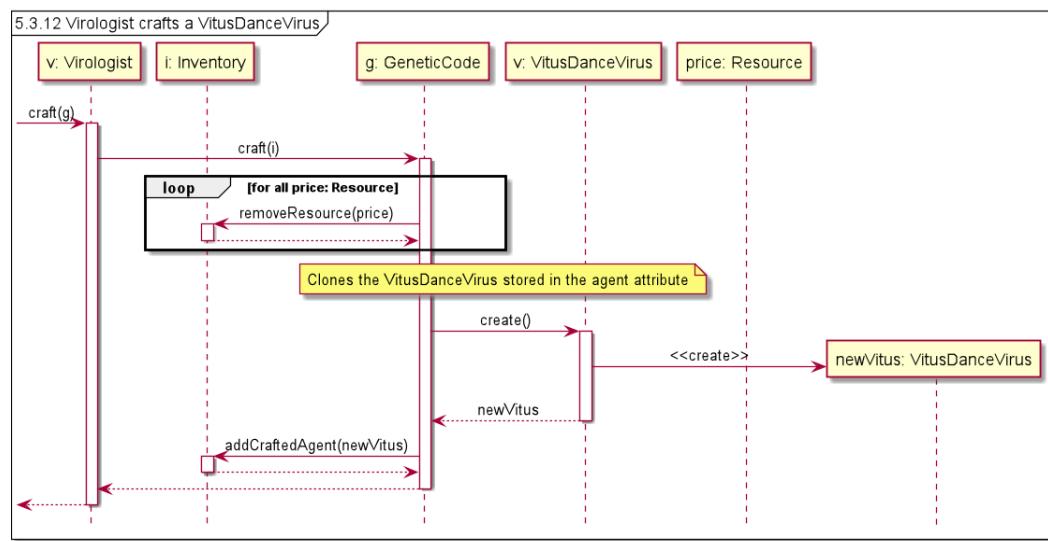
4.3.10 craftStunVirus



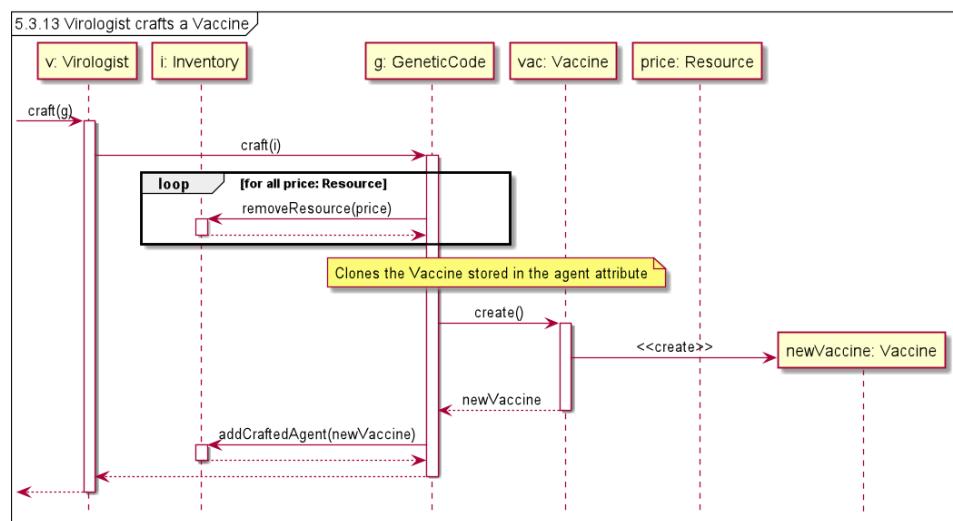
4.3.11 craftAmnesiaVirus



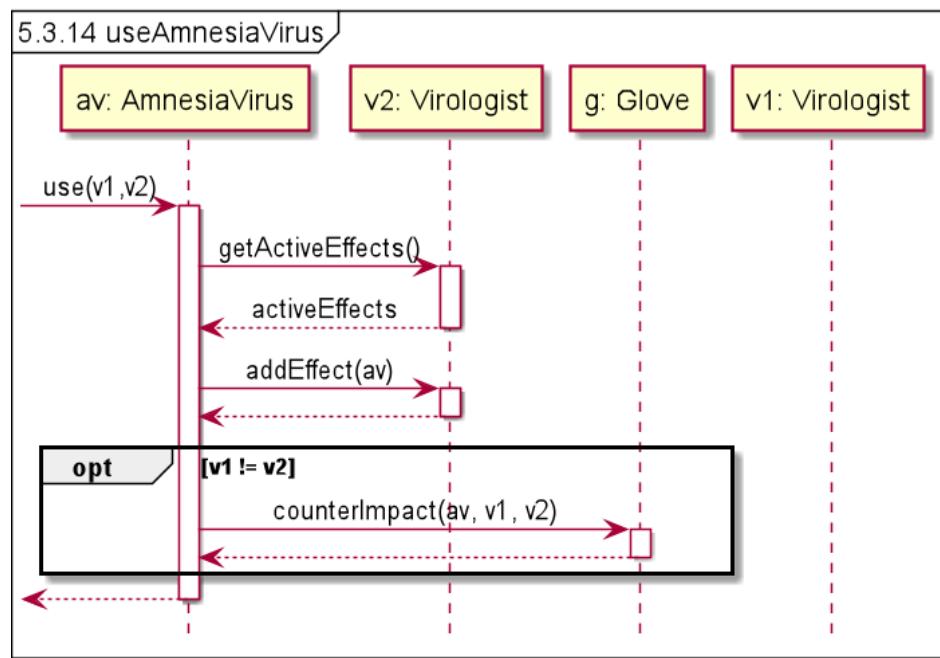
4.3.12 craftVitusDanceVirus



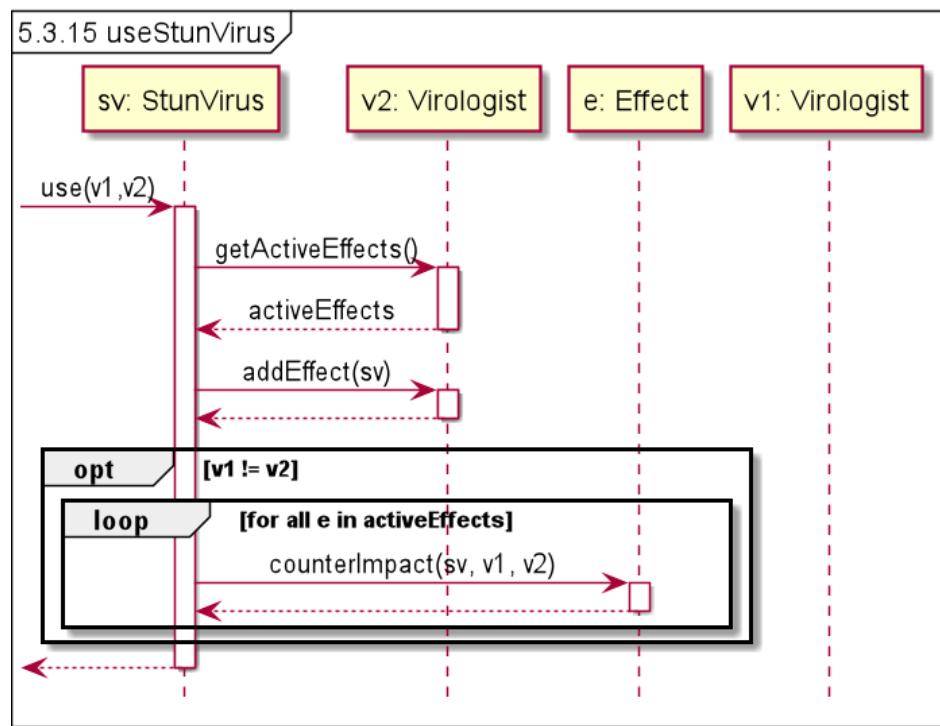
4.3.13 craftVaccine



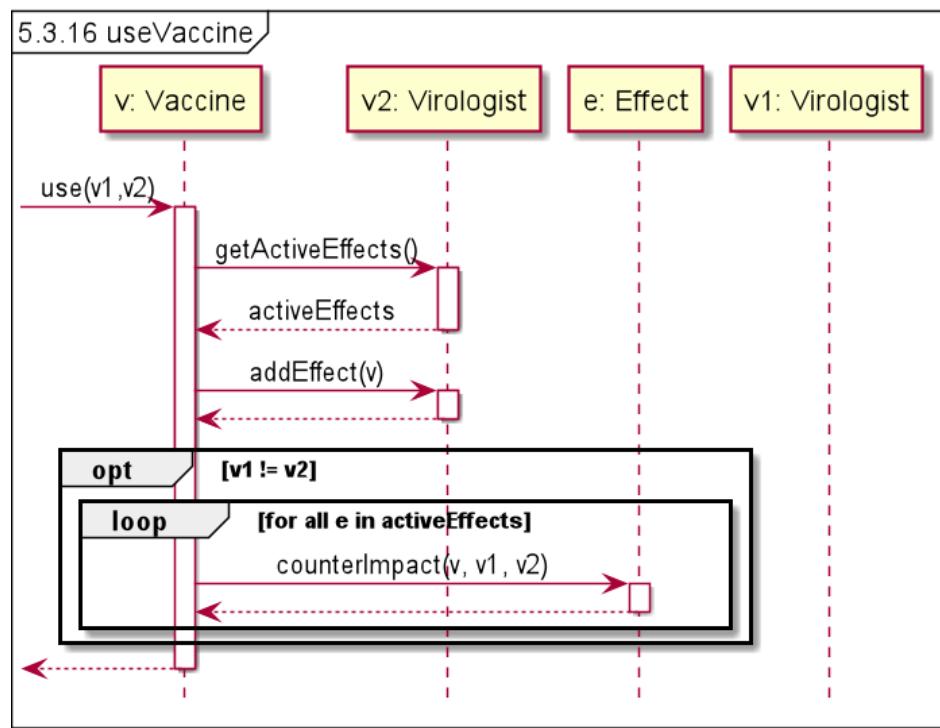
4.3.14 useAmnesiaVirus



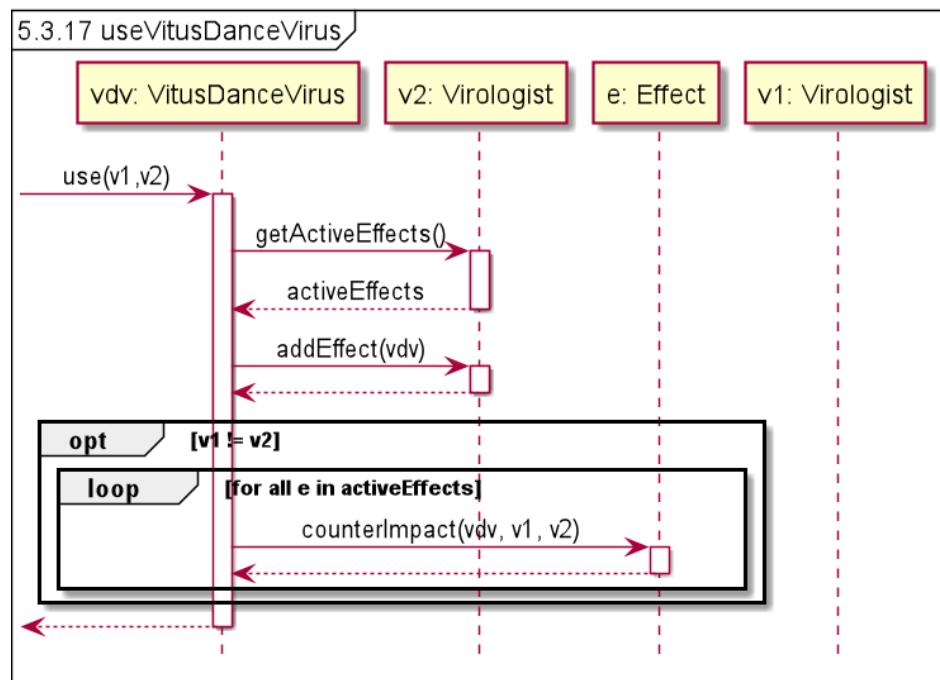
4.3.15 useStunVirus



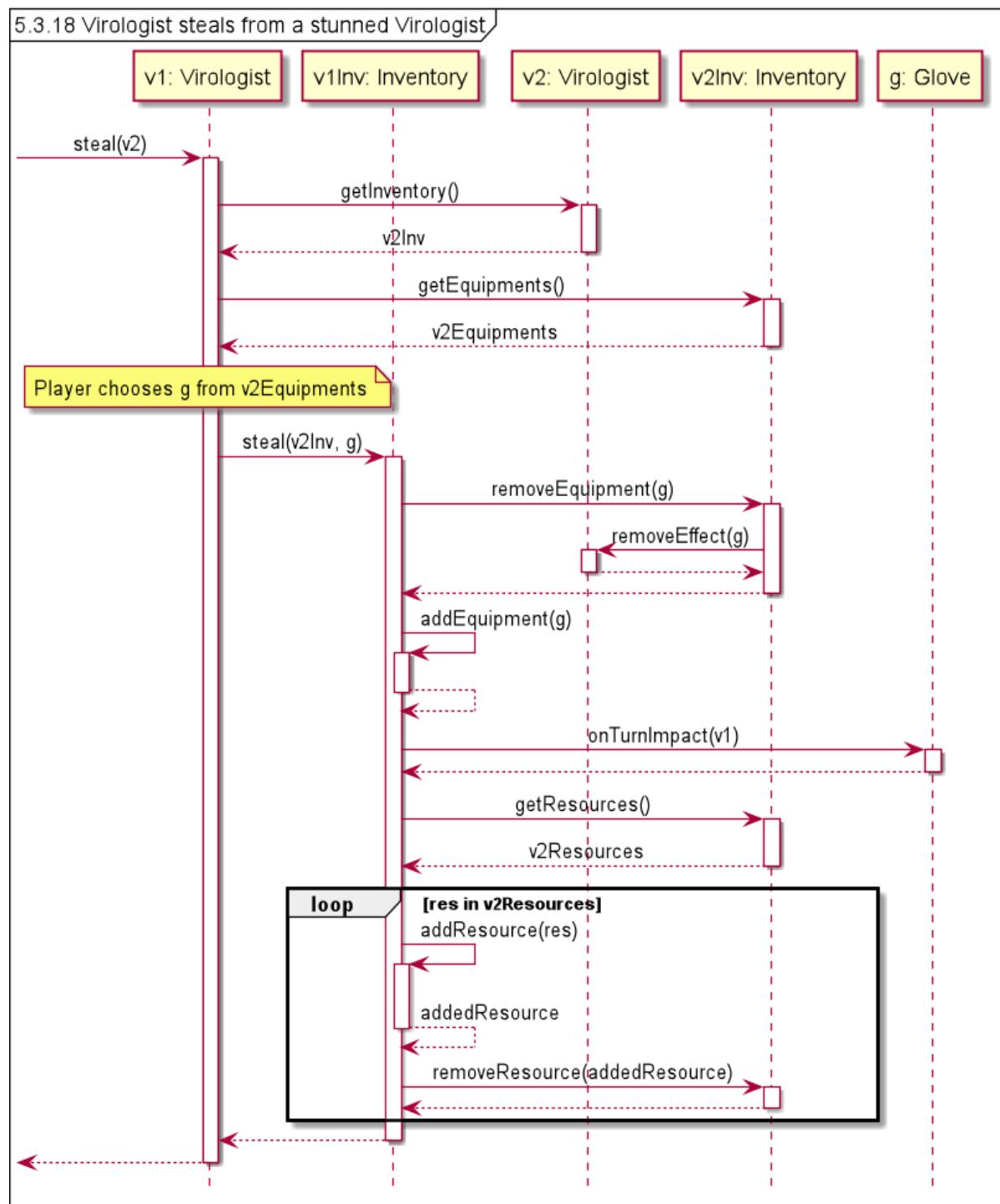
4.3.16 useVaccine



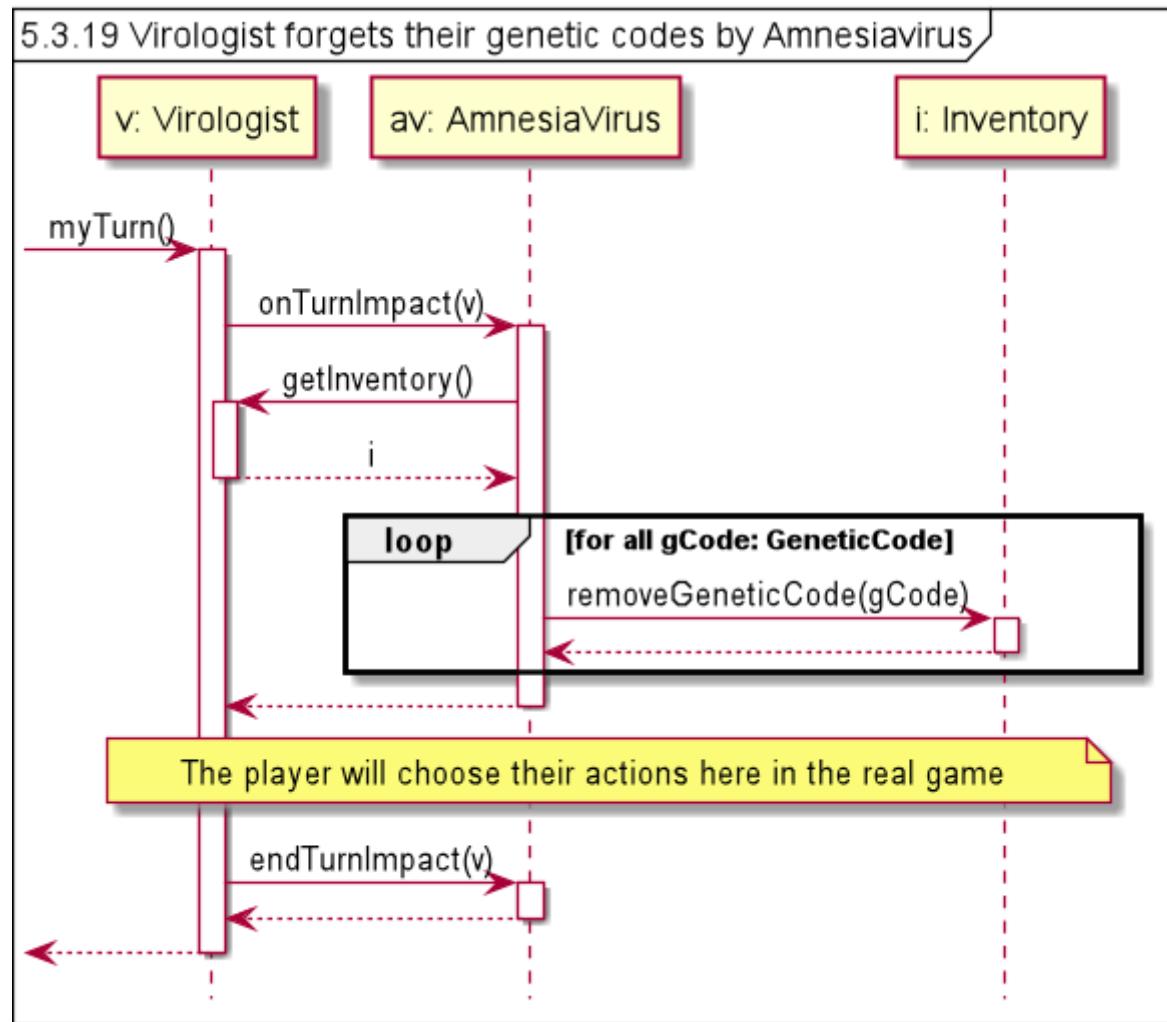
4.3.17 useVitusDanceVirus



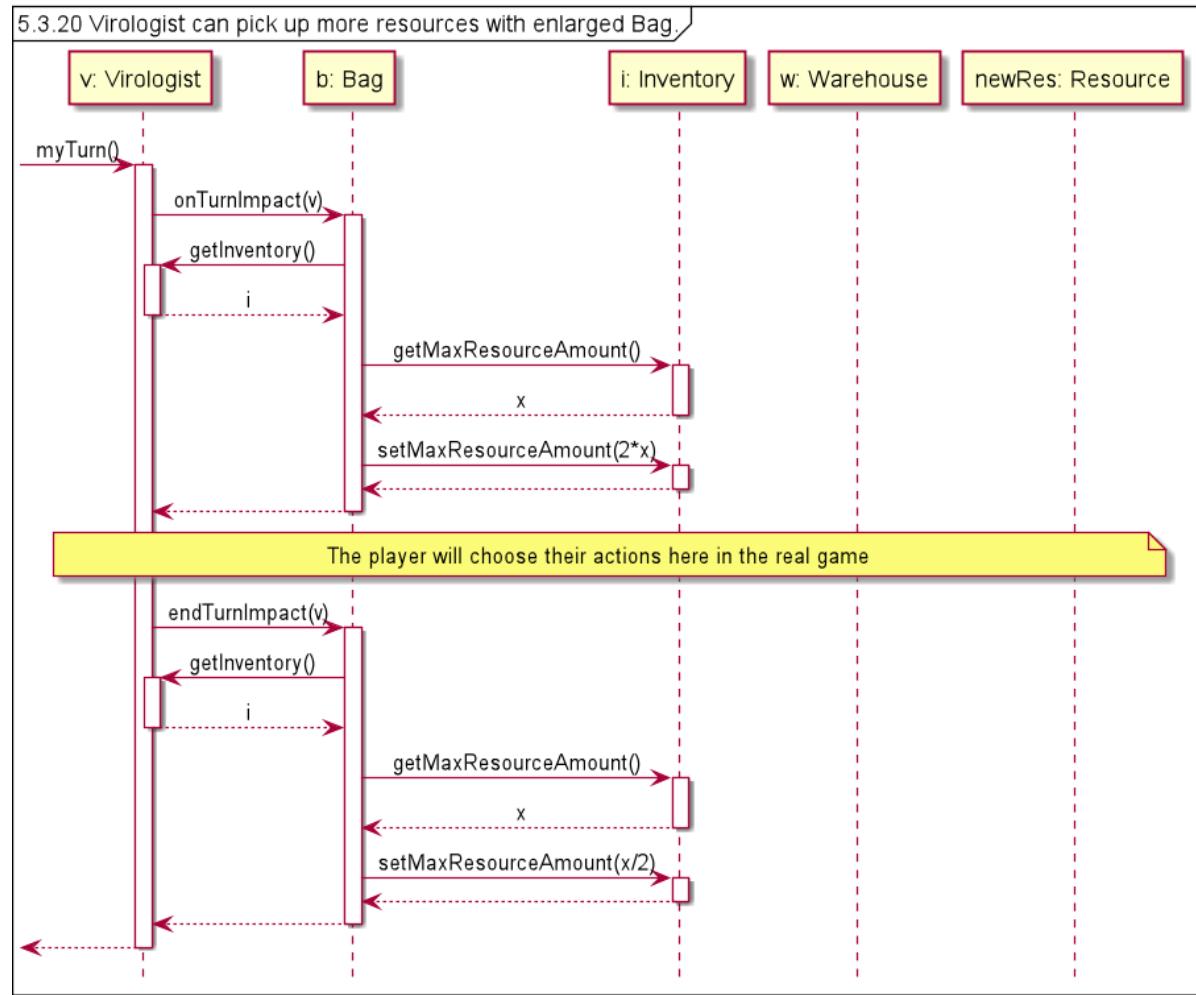
4.3.18 robVirologist



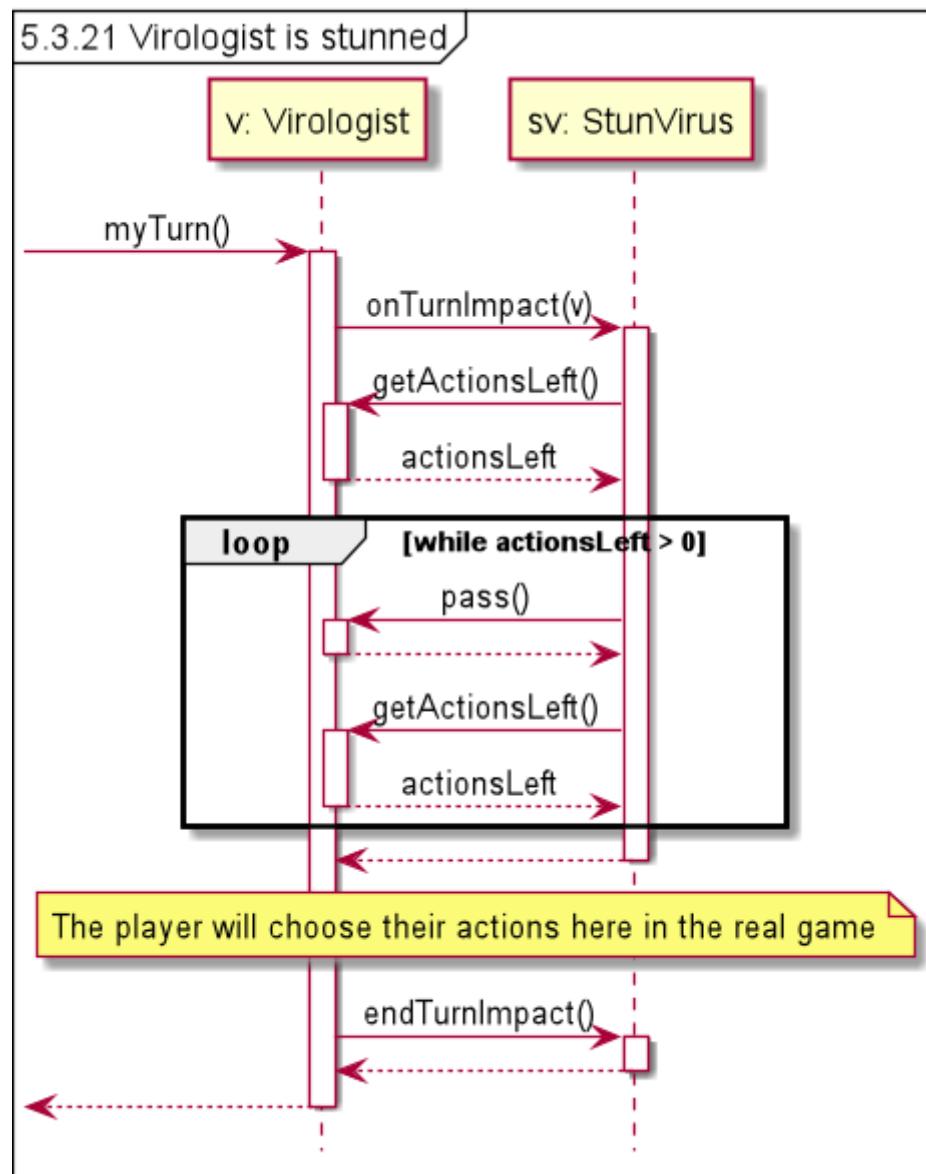
4.3.19 forgetCodesViaAmnesiaVirus



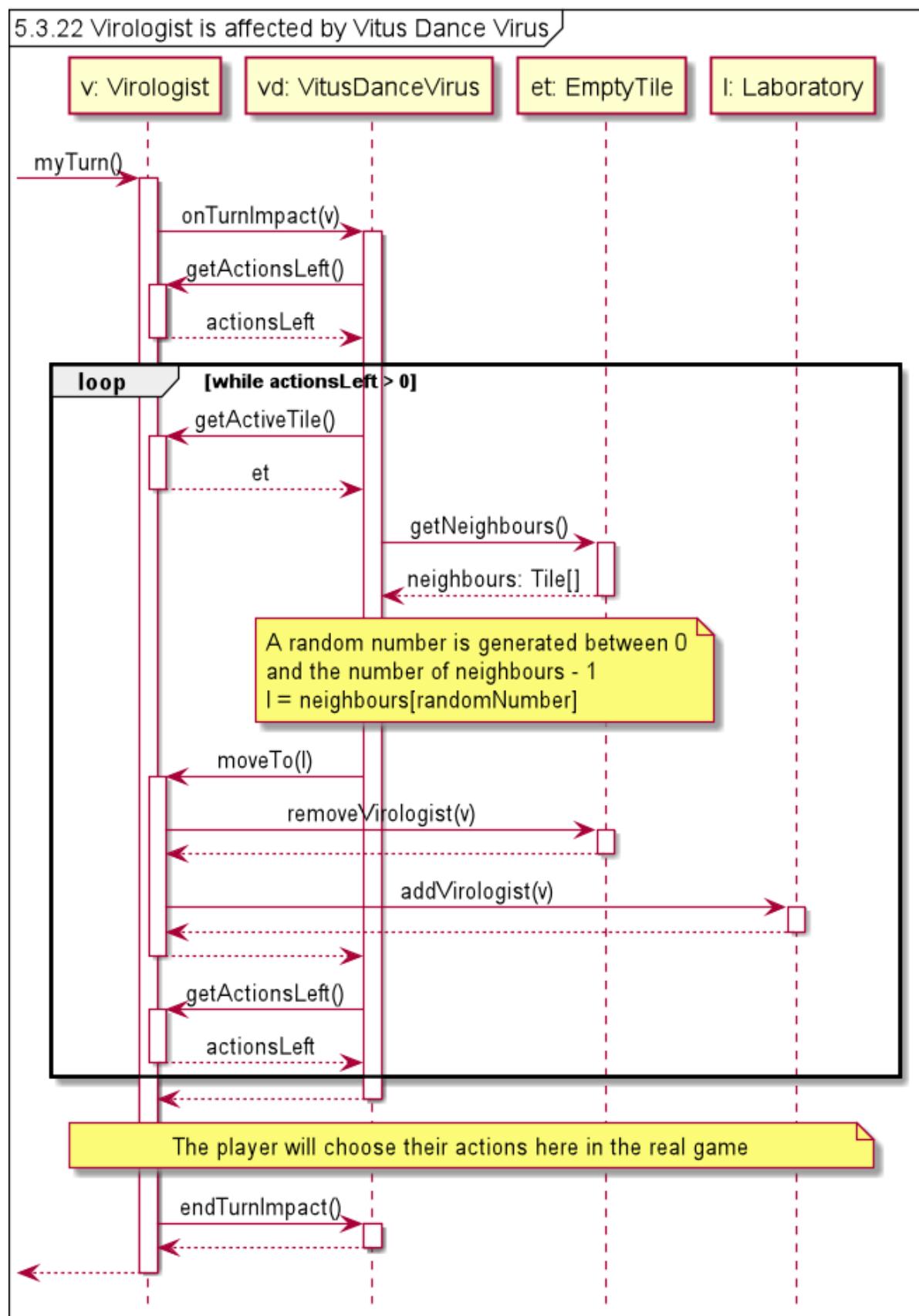
4.3.20 enlargeTheBag



4.3.21 stunnedPlayerMissesTurn

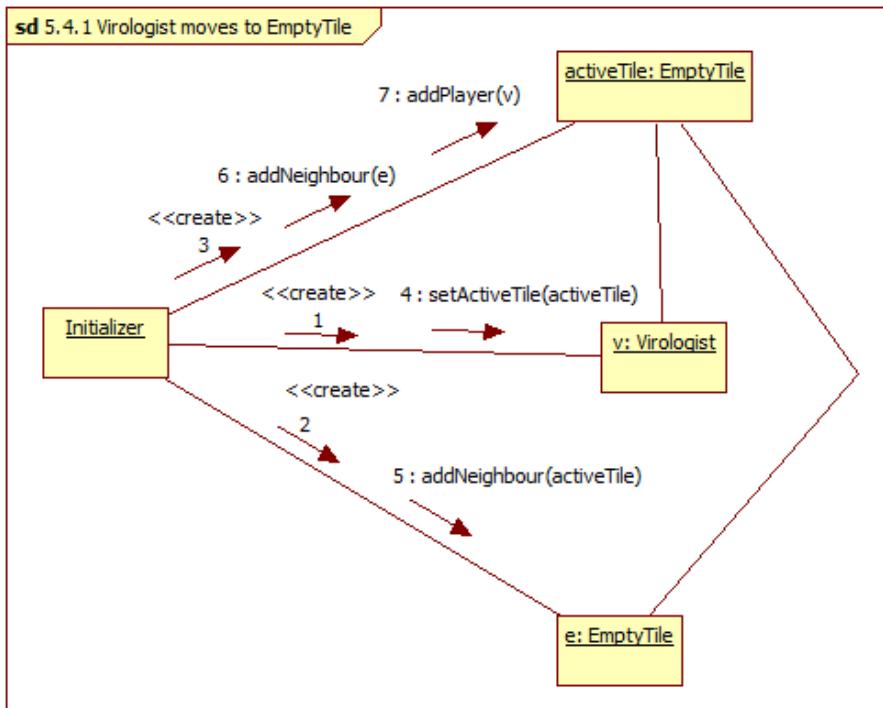


4.3.22 playerVitusDances

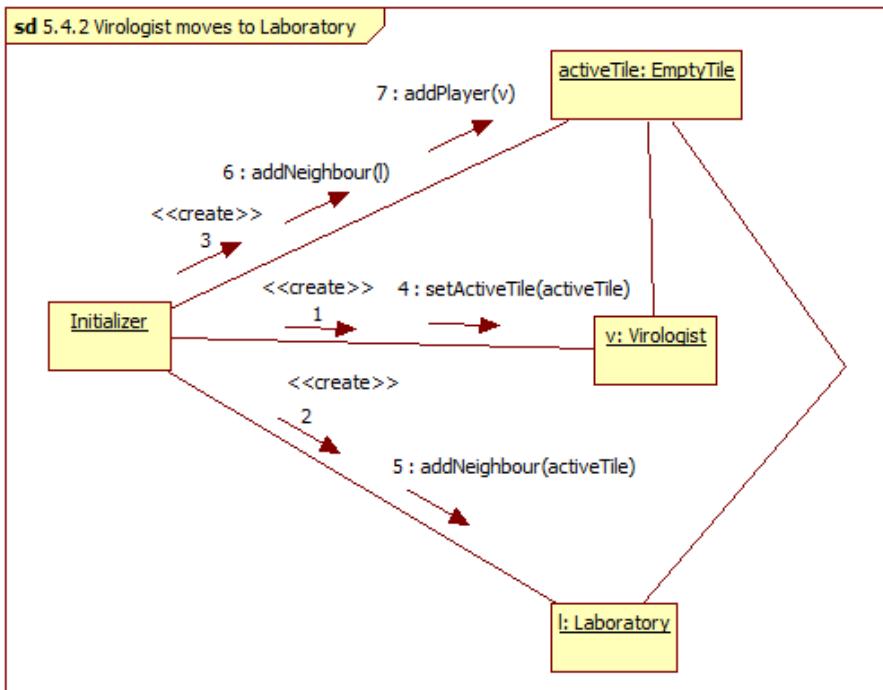


4.4 Kommunikációs diagramok

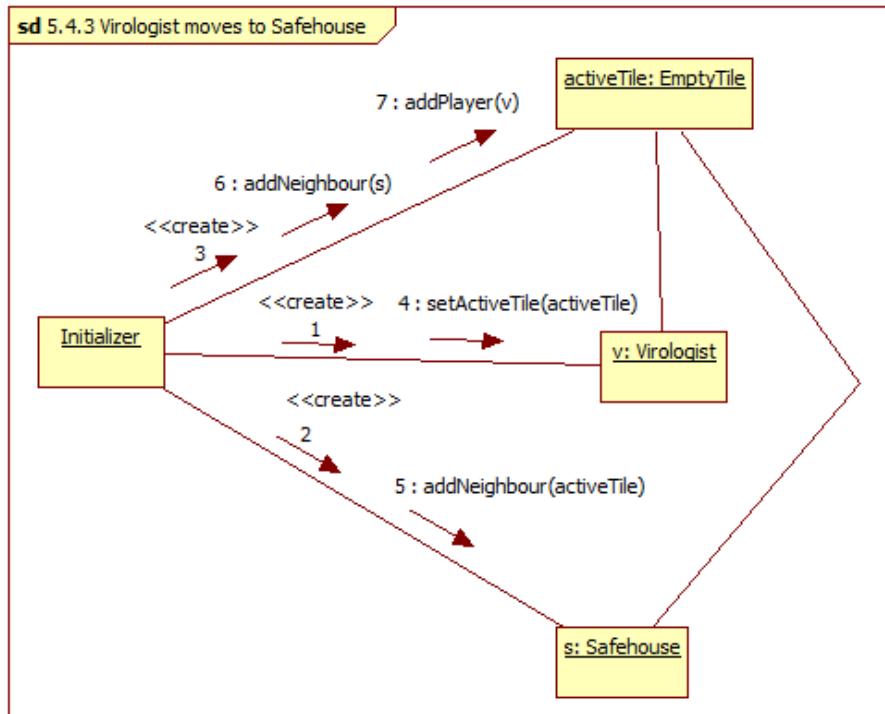
4.4.1 moveToEmptyTile



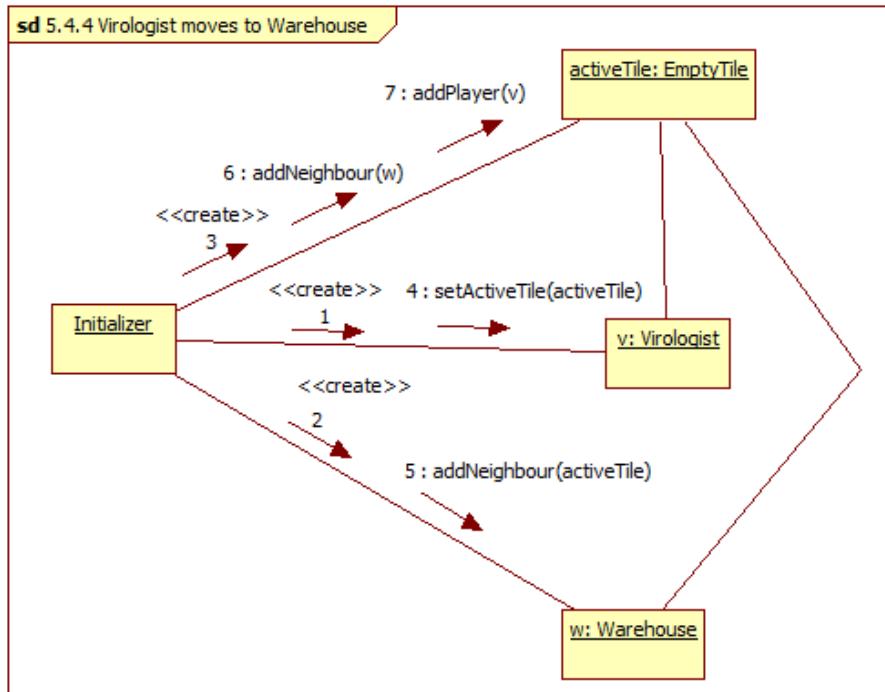
4.4.2 MoveToLaboratory



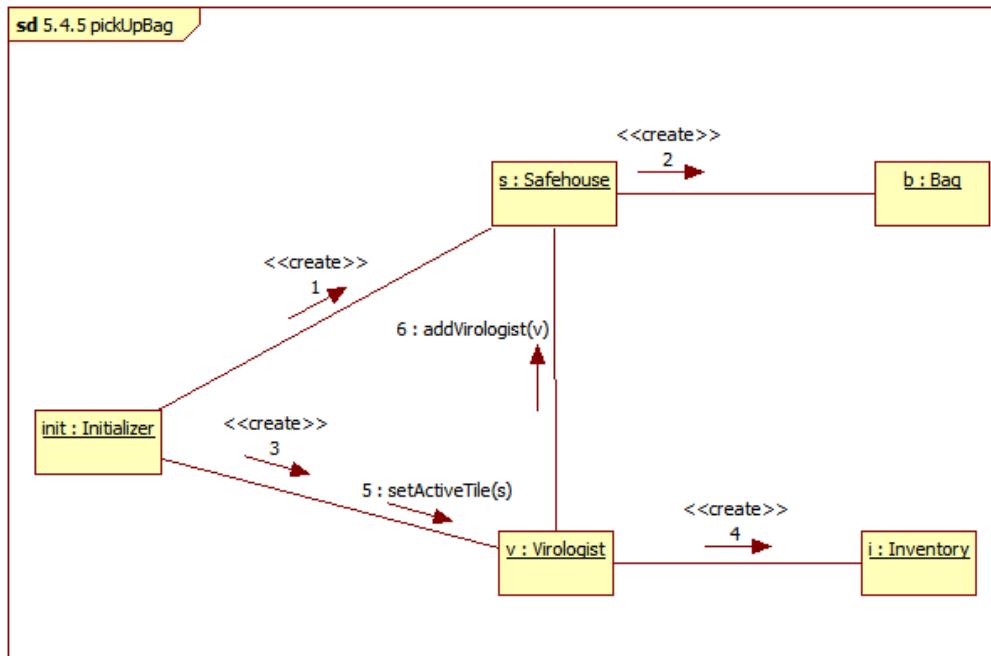
4.4.3 moveToSafehouse



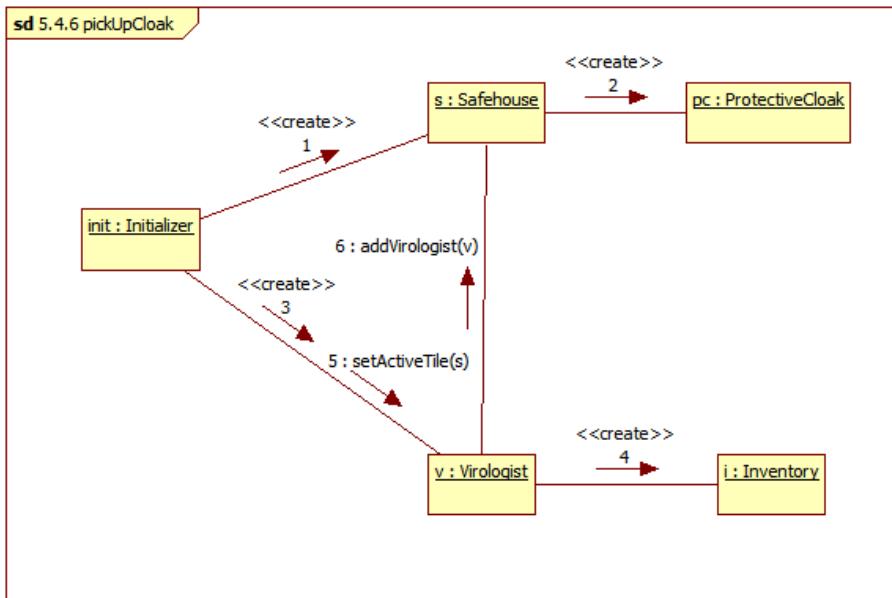
4.4.4 moveToWarehouse



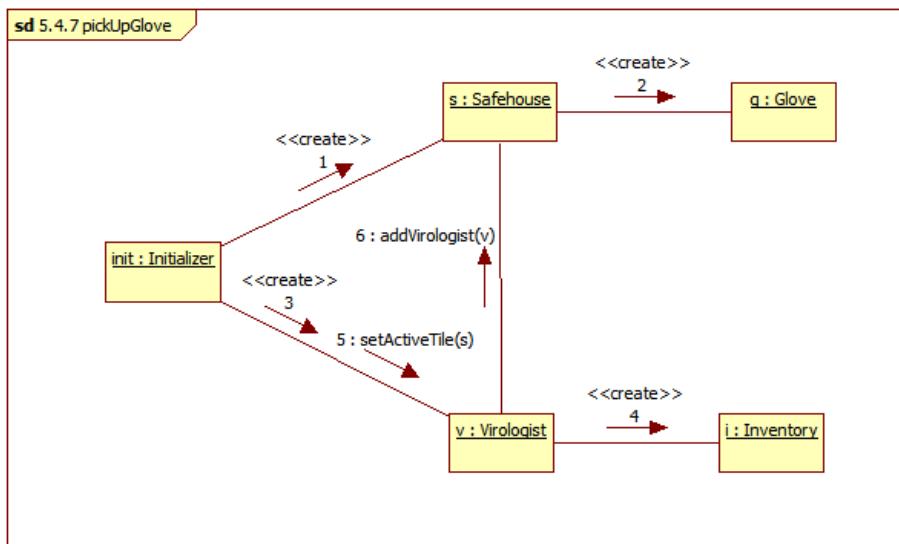
4.4.5 pickUpBag



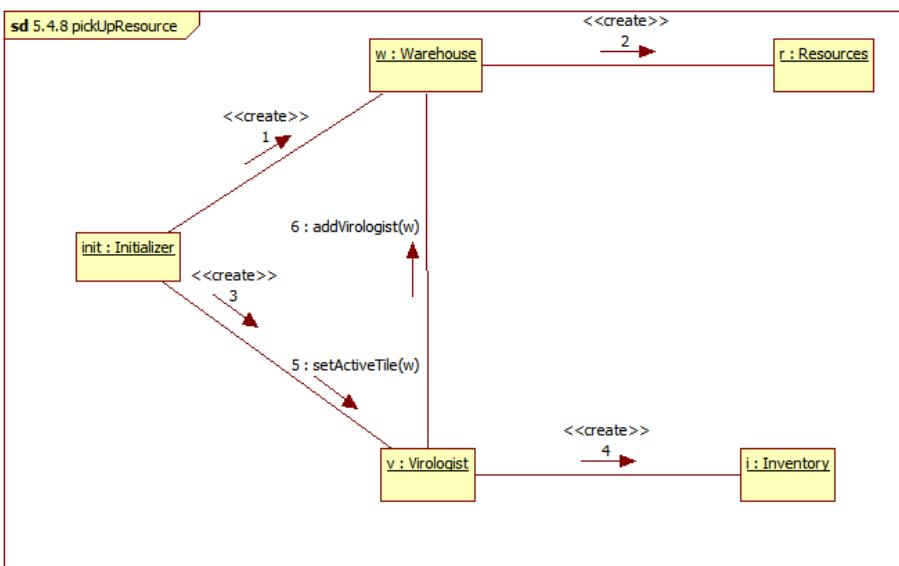
4.4.6 pickUpCloak



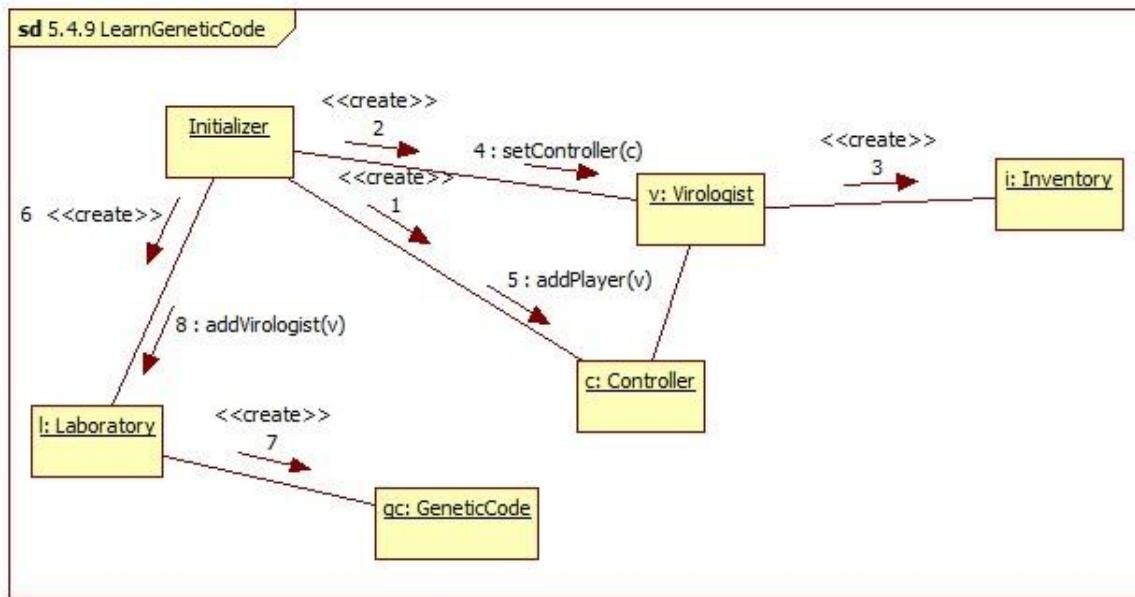
4.4.7 pickUpGlove



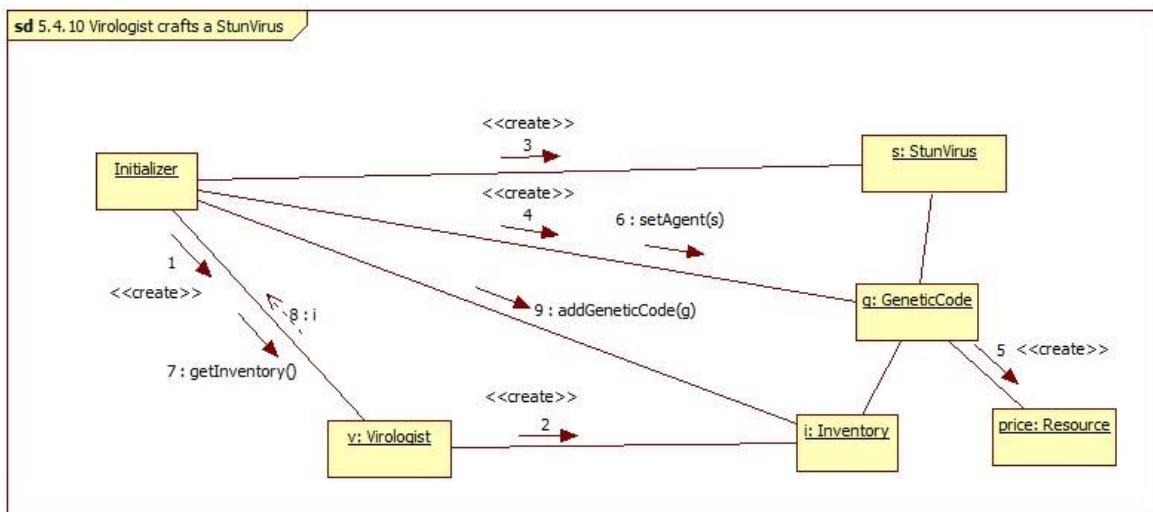
4.4.8 pickUpResource



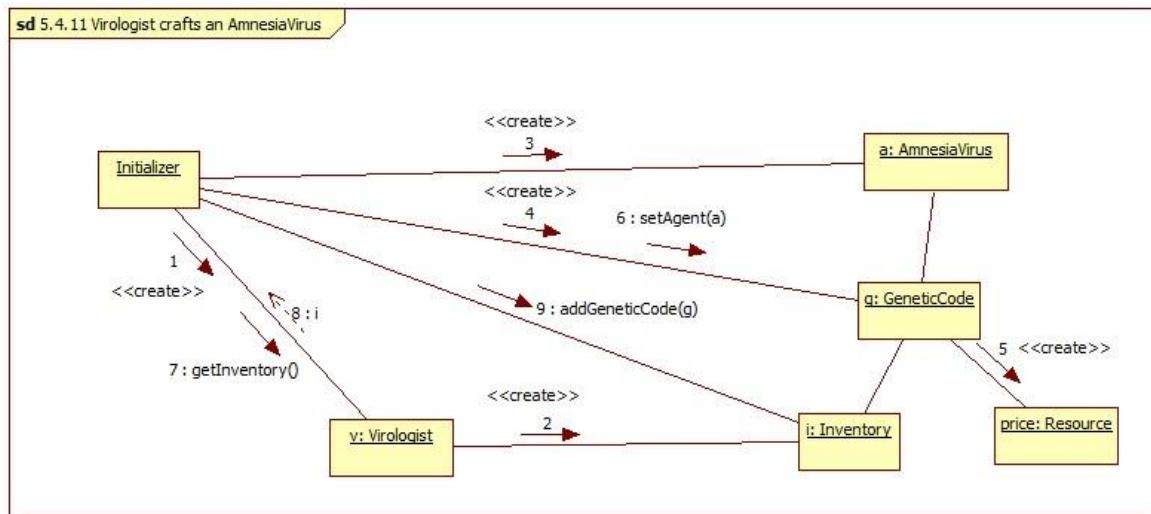
4.4.9 learnGeneticCode



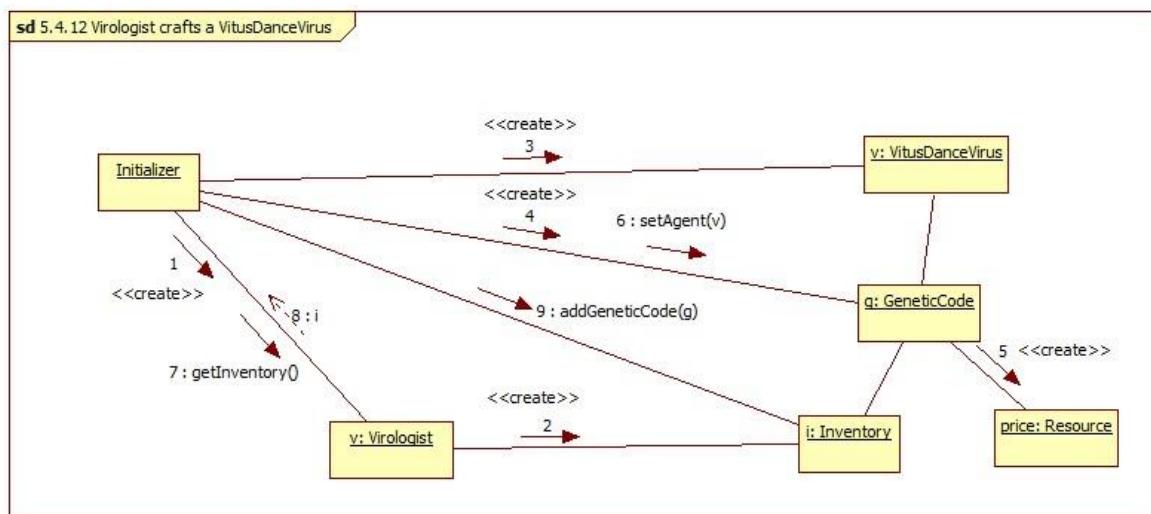
4.4.10 craftStunVirus



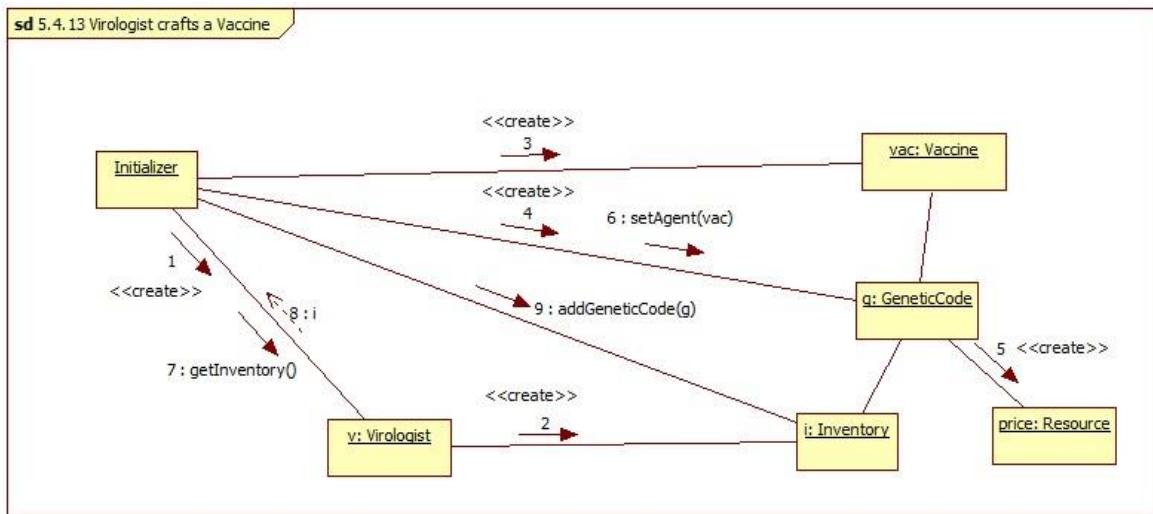
4.4.11 **craftAmnesiaVirus**



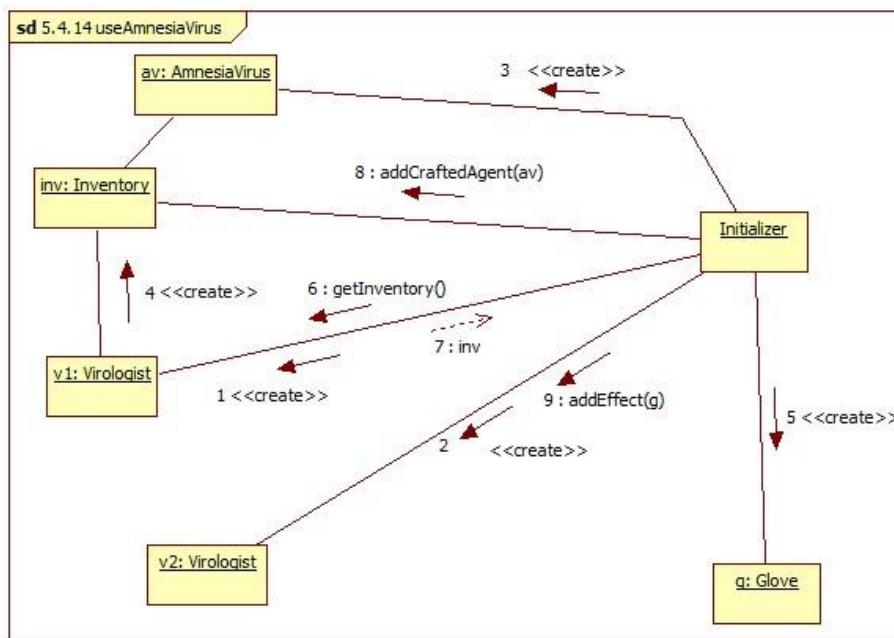
4.4.12 **craftVitusDanceVirus**



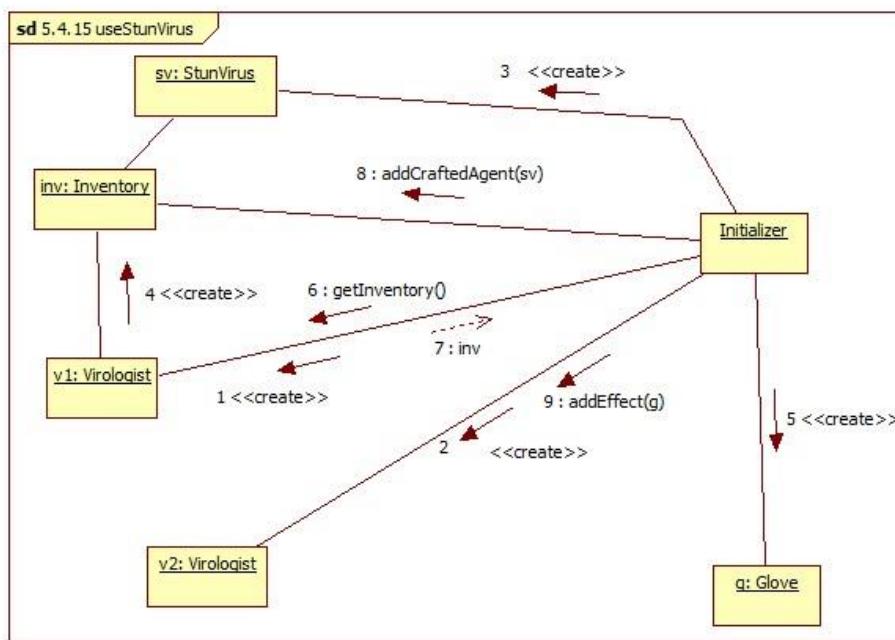
4.4.13 craftVaccine



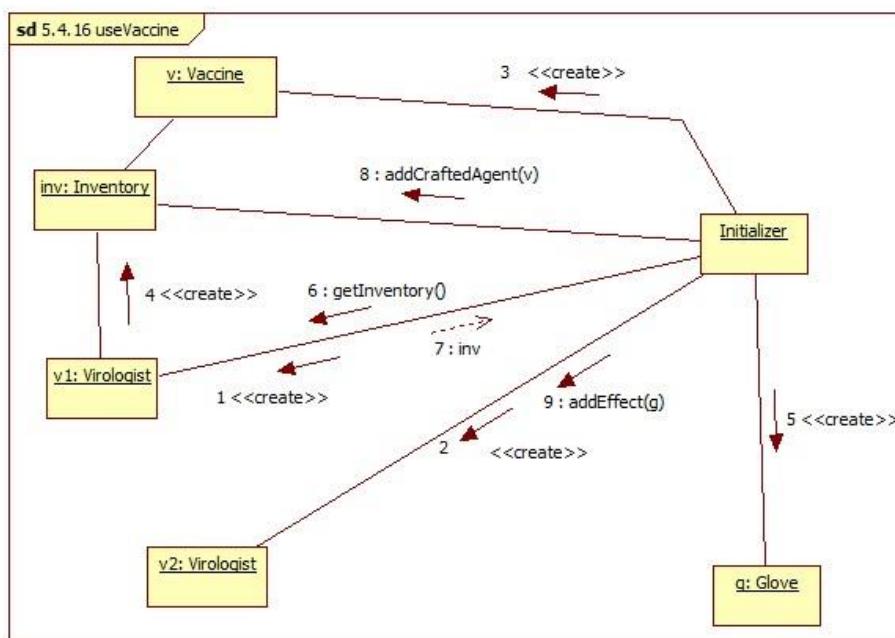
4.4.14 useAmnesiaVirus



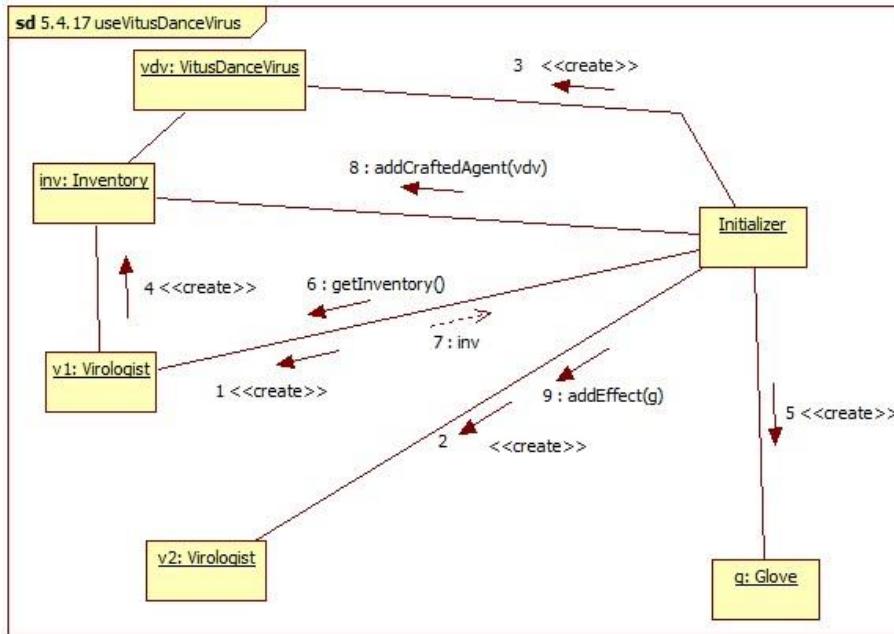
4.4.15 useStunVirus



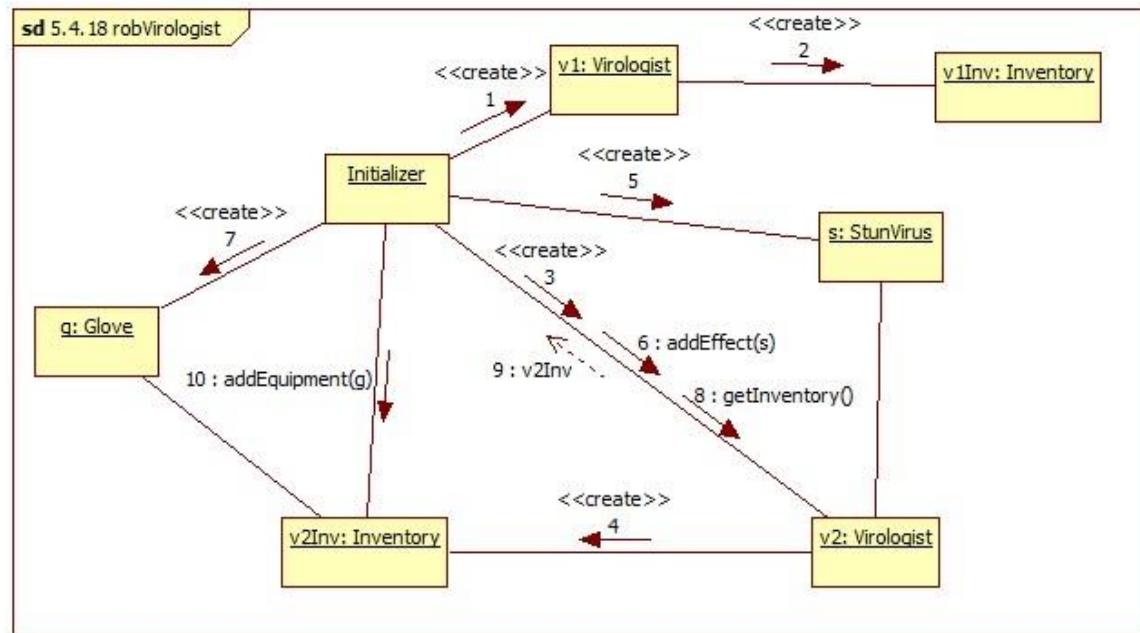
4.4.16 useVaccine



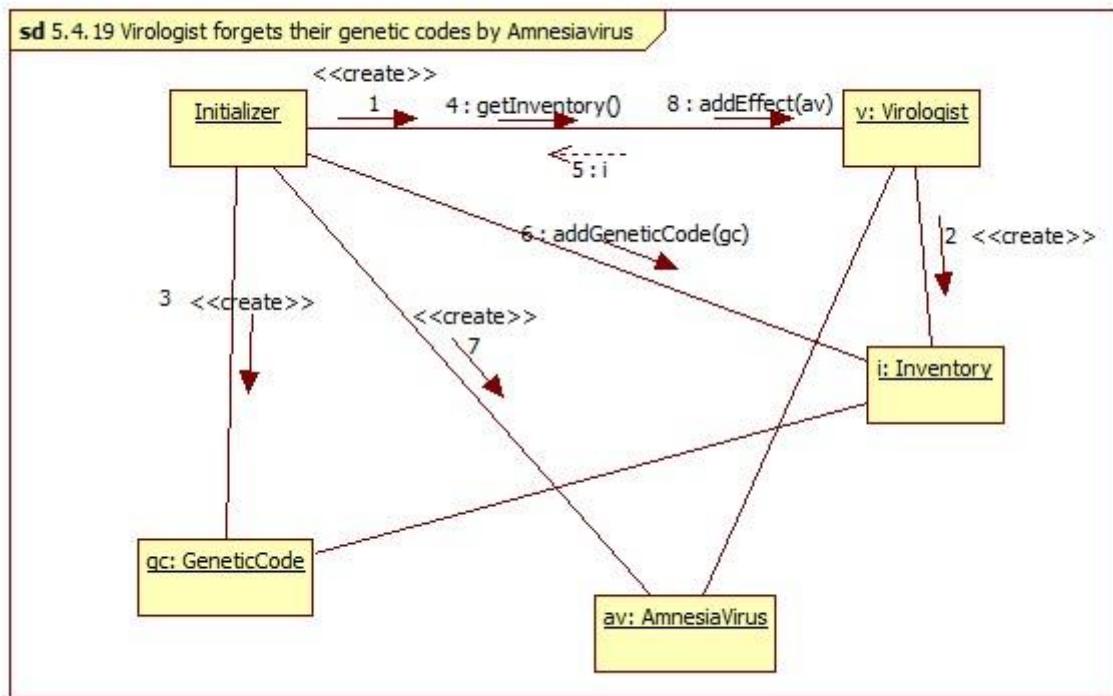
4.4.17 useVitusDanceVirus



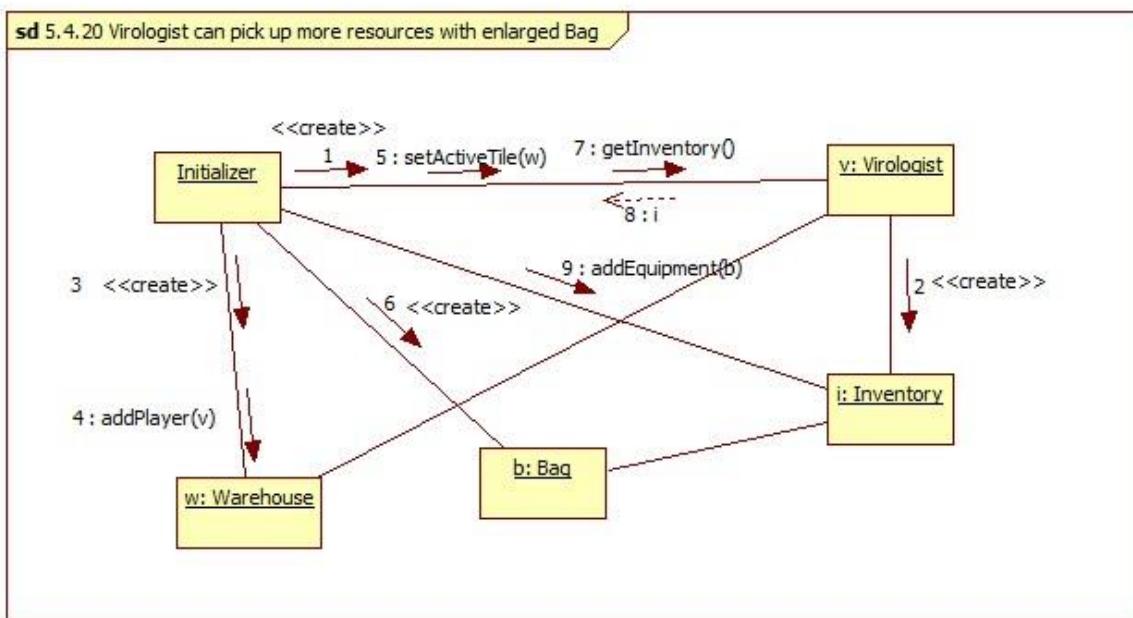
4.4.18 robVirologist



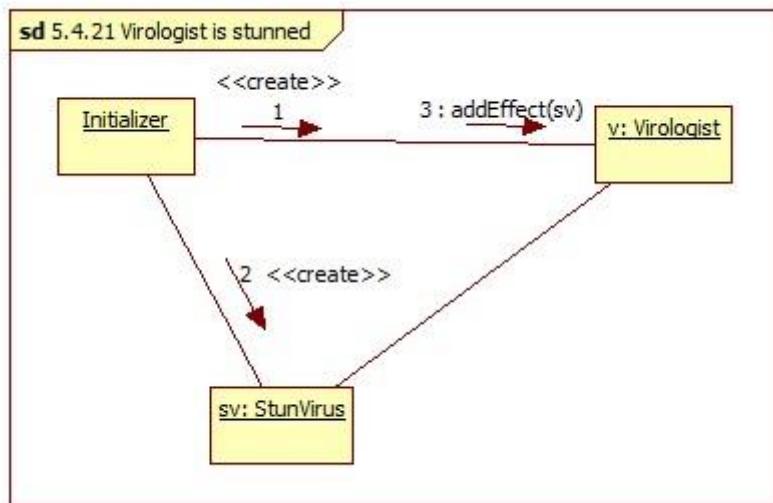
4.4.19 forgetCodesViaAmnesiaVirus



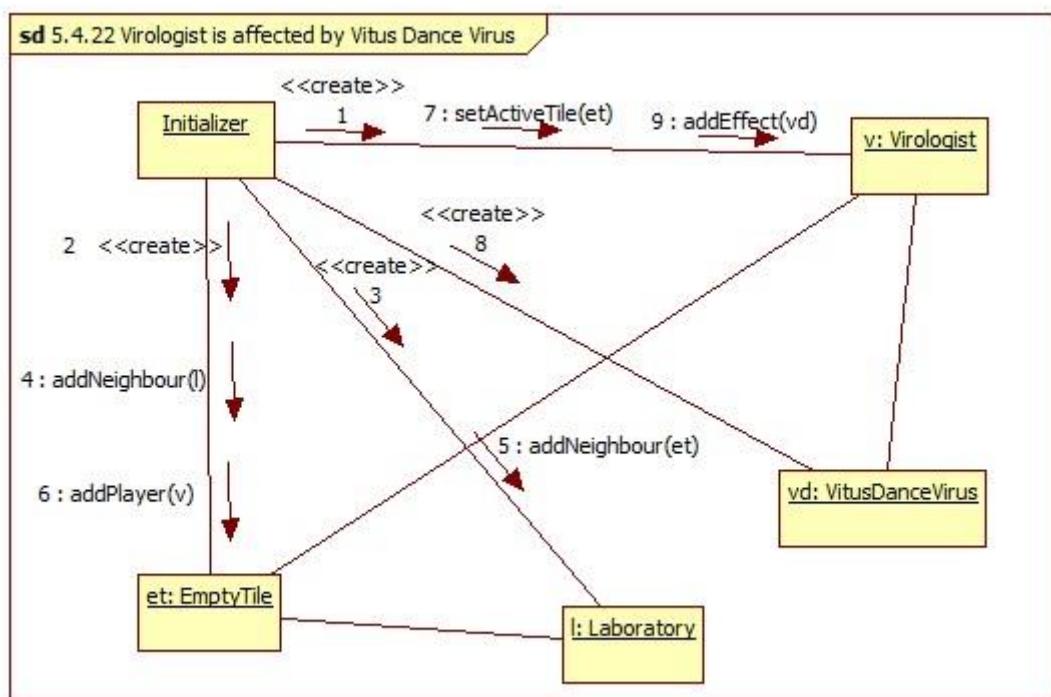
4.4.20 enlargeTheBag



4.4.21 stunnedPlayerMissesTurn



4.4.22 playerVitusDances



4.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2022.03.18. 13:00	1 óra	Fekete Kászonyi Jurásek Domonkos Száraz	Értekezlet. Döntés: Use-casek szétosztása a csapattagok között Döntés: Kommunikációs diagramok szétosztása a csapattagok között Döntés: Kommunikációs diagramok WhiteStarUML-ben
2022.03.18. 16:00	2 óra	Jurásek	4 szekvenciadiagram és kommunikációs diagram elkészítése
2022.03.19 13:00	2 óra	Száraz	4 szekvenciadiagram készítése test-case-ekhez
2022.03.19. 15:00	2 óra	Domonkos	4 use-case megírása, szekvenciadiagramok elkészítése
2022.03.19 17:00	3 óra	Kászonyi	4 use-case megírása, szekvencia diagrammok megrajzolása, kommunikációs diagrammok elkészítése.
2022.03.19. 17:00	2 óra	Domonkos	4 kommunikációs diagram elkészítése, dokumentum szerkesztése
2022.03.19 21:00	2 óra	Fekete Kászonyi Jurásek Domonkos Száraz	Értekezlet. Döntés: A kommunikációs diagramokon az inicializálásnak kell megjelennie.
2022.03.19 23:00	1 óra	Fekete	2 test-case-hez tartozó szekvenciadiagram és kommunikációs diagram elkészítése.

2022.03.20 11:00	2 óra	Fekete	4 test-case-hez tartozó szekvenciadiagram és kommunikációs diagram elkészítése, test-case leírások elkészítése.
2022.03.20 14:00	2 óra	Jurásek	5.2 véglegesítése, dokumentum szerkesztése
2022.03.20 18:00	2 óra	Domonkos	Dokumentum szerkesztése, ellenőrzése
2022.03.20 19:00	2 óra	Fekete	Dokumentum szerkesztése, átnézése
2022.03.20 19:00	2 óra	Száraz	Dokumentum szerkesztése, ellenőrzése
2022.03.20 20:00	2 óra	Kászonyi	Dokumentum átolvasása, képek beillesztése

5. Szkeleton tervezése

52 – *service_unavailable*

Konzulens:
Micskei Zoltán

Csapattagok

Fekete Sámuel	GJ8J3A	feketesamu@gmail.com
Száraz Dániel	GT5X34	szdani2012@gmail.com
Jurásek Jónás	PH4QFK	jurasekjonas12@gmail.com
Domonkos Ádám	CWGYWC	domonkosadam01@gmail.com
Kászonyi Zsombor Marcell	DCE2Q1	kaszonyirelli@gmail.com

2022.02.23.

5. Szkeleton tervezése

Változtatások:

Az agent use függvényét módosítottuk (4.4.14 virologistUsesAgent), a counterImpact-ek csak akkor hívódnak meg, ha nem magára kenné fel a virológus az ágenset, ez így a játék szempontjából logikusabb, valamint az activeEffect-eket előbb kérdezi le, mielőtt hozzáadná az újat, ezt azért kellett felcserélni, hogy a Vakcinát például fel tudja kenni magára a virológus.

Továbbá a steal függvényben a felszerelések lopása történik meg először, valamint az ellopott felszerelésen lefut az onTurnImpact, hogy rögtön legyen hatása. Bekerült még egy újabb feltétel a felszerelés lopásához: csak akkor mehet végbe, ha a lopó játékosnak kevesebb, mint 3 védőfelszerelése van.

A Virologist osztálynak van egy statikus mezője, amin keresztül a Controllert tudja elérni. Erre genetikai kód megtanulása után van szükség, hogy a kontroller ellenőrizze, megnyerte-e a játékot az adott játékos.

5.1 A szkeleton modell valóságos use-case-ai

5.1.1 Use-case diagram



5.1.2 Use-case leírások

Use-case neve	moveToEmptyTile
Rövid leírás	A játékos egy üres mezőre lép.
Aktorok	Player
Forgatókönyv	<i>Kiindulási feltétel: a virológus egy üres mezőn áll.</i> 1. A játékos egy szomszédos üres mezőre lép

Use-case neve	moveToLaboratory
Rövid leírás	A játékos egy laboratórium mezőre lép.
Aktorok	Player
Forgatókönyv	<i>Kiindulási feltétel: a virológus egy üres mezőn áll.</i> 1. A játékos egy szomszédos laboratórium mezőre lép

Use-case neve	moveToSafehouse
Rövid leírás	A játékos egy óvóhely mezőre lép.
Aktorok	Player
Forgatókönyv	<i>Kiindulási feltétel: a virológus egy üres mezőn áll.</i> 1. A játékos egy szomszédos óvóhely mezőre lép

Use-case neve	moveToWarehouse
Rövid leírás	A játékos egy raktár mezőre lép.
Aktorok	Player
Forgatókönyv	<i>Kiindulási feltétel: a virológus egy üres mezőn áll.</i> 1. A játékos egy szomszédos raktár mezőre lép

Use-case neve	pickUpBag
Rövid leírás	A játékos felvesz egy zsák felszerelést egy óvóhely mezőről.
Aktorok	Player
Forgatókönyv	<i>Kiindulási feltétel: Virológus egy olyan óvóhely mezőn áll, ahol a zsák a felvehető felszerelés.</i> 1. Játékos megpróbálja felvenni a zsákat. 2/A. Ha a játékosnál még van hely, és fel tudja venni a zsákat akkor a zsák egy másolata bekerül a virológus felszerelései közé. 2/B. Ha a játékos felszerelései között már nincs hely, akkor nem tudja felvenni. 3/A. A zsák hatása érvényesül a virológuson: Megnöveli a maximális anyagmennyiséget

Use-case neve	pickUpCloak
Rövid leírás	A játékos felvesz egy védőköpeny felszerelést egy óvóhely mezőről.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Virológus egy olyan óvóhely mezőn áll, ahol a védőköpeny a felvehető felszerelés.</i></p> <p>1. Játékos megpróbálja felvenni a védőköpenyt.</p> <p>2/A. Ha a játékosnál még van hely, és fel tudja venni a védőköpenyt akkor a védőköpeny egy másolata bekerül a virológus felszerelései közé.</p> <p>2/B. Ha a játékos felszerelései között már nincs hely, akkor nem tudja felvenni.</p> <p>3/A. A védőköpeny hatása érvényessül a virológuson: 82,3% eséllyel megvédi a jövőben rákent ágensektől.</p>

Use-case neve	pickUpGlove
Rövid leírás	A játékos felvesz egy kesztyű felszerelést egy óvóhely mezőről.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Virológus egy olyan óvóhely mezőn áll, ahol a kesztyű a felvehető felszerelés.</i></p> <p>1. Játékos megpróbálja felvenni a kesztyűt.</p> <p>2/A. Ha a játékosnál még van hely, és fel tudja venni a kesztyűt akkor a kesztyű egy másolata bekerül a virológus felszerelései közé.</p> <p>2/B. Ha a játékos felszerelései között már nincs hely, akkor nem tudja felvenni.</p> <p>3/A. A kesztyű hatása érvényesül a virológuson: A jövőben rákent ágenseket visszakeni a támadóra.</p>

Use-case neve	pickUpResource
Rövid leírás	A játékos felvesz egy anyagot egy raktár mezőről.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Virológus egy raktár mezőn áll</i></p> <p>1. Játékos felveszi az anyagot</p> <p>2. Hozzáad egy adott mennyiségű anyagot a virológushoz.</p> <p>3. A virológus anyagmennyisége növekedik a felvett mennyiségű anyaggal, de nem haladja meg a maximális mennyiséget.</p>

Use-case neve	learnGeneticCode
Rövid leírás	A játékos megtanul egy genetikai kódot egy laboratórium mezőn.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel:</i> A játékos egy olyan laboratórium mezőn áll, ahol még nem tanult korábban genetikai kódot.</p> <p>1. A laboratórium mező genetikai kódja eltárolódik a virológus felszerelésében.</p> <p>2/A. Ha ez volt az utolsó genetikai kód, amit nem tanult meg ez a játékos, akkor megnyerte a játékot.</p> <p>2/B. Egyébként folytatódik a játék.</p>

Use-case neve	craftVaccine
Rövid leírás	A játékos megpróbál egy vakcinát elkészíteni.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel:</i> A virológus megtanulta a Vakcinához tartozó Genetikai kódot.</p> <p>1/A. A játékos elkészíti a vakcinát.</p> <p>1/B. A játékos virológusának nincs elég anyagkészlete, ezért nem tudja elkészíteni a vakcinát.</p>

Use-case neve	craftVitusDance
Rövid leírás	A játékos megpróbál egy vitustáncvírust elkészíteni.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel:</i> A virológus megtanulta a vitustáncvírushoz tartozó genetikai kódot.</p> <p>1/A. A játékos elkészíti a vitustáncvírust.</p> <p>1/B. A játékos virológusának nincs elég anyagkészlete, ezért nem tudja elkészíteni a vitustáncvírust.</p>

Use-case neve	craftStun
Rövid leírás	A játékos megpróbál egy bénítóvírust elkészíteni.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel:</i> A virológus megtanulta a bénítóvírushoz tartozó genetikai kódot.</p> <p>1/A. A játékos elkészíti a bénítóvírust.</p> <p>1/B. A játékos virológusának nincs elég anyagkészlete, ezért nem tudja elkészíteni a bénítóvírust.</p>

Use-case neve	craftAmnesia
Rövid leírás	A játékos megpróbál egy amnéziavírust elkészíteni.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel:</i> A virológus megtanulta az amnéziavírushoz tartozó genetikai kódot.</p> <p>1/A. A játékos elkészíti az amnéziavírust.</p> <p>1/B. A játékos virológusának nincs elég anyagkészlete, ezért nem tudja elkészíteni az amnéziavírust.</p>

Use-case neve	useAmnesiaVirus
Rövid leírás	A játékos egy virológusra felkeni az amnézia vírust.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Egy virológus, akire fel lehet kenni a vírust (lehet saját maga is).</i></p> <p>1/A. A játékos felkeni a vírust a választott virológusra.</p> <p>1/B. A játékos felkenné a vírust, de kivédi.</p> <p>1/C. A játékos felkenné a vírust, de visszakenik rá.</p>

Use-case neve	useStunVirus
Rövid leírás	A játékos egy virológusra felkeni a bénító vírust.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Egy virológus, akire fel lehet kenni a vírust (lehet saját maga is).</i></p> <p>1/A. A játékos felkeni a vírust a választott virológusra.</p> <p>1/B. A játékos felkenné a vírust, de kivédi.</p> <p>1/C. A játékos felkenné a vírust, de visszakenik rá.</p>

Use-case neve	useVaccine
Rövid leírás	A játékos egy virológusra felkeni a vakcinát.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Egy virológus, akire fel lehet kenni a vírust (lehet saját maga is).</i></p> <p>1/A. A játékos felkeni a vírust a választott virológusra.</p> <p>1/B. A játékos felkenné a vírust, de kivédi.</p> <p>1/C. A játékos felkenné a vírust, de visszakenik rá.</p>

Use-case neve	useVitusDanceVirus
Rövid leírás	A játékos egy virológusra felkeni a vitus tánc vírust.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: Egy virológus, akire fel lehet kenni a vírust (lehet saját maga is).</i></p> <p>1/A. A játékos felkeni a vírust a választott virológusra.</p> <p>1/B. A játékos felkenné a vírust, de kivédi.</p> <p>1/C. A játékos felkenné a vírust, de visszakenik rá.</p>

Use-case neve	robVirologist
Rövid leírás	Az egyik játékos lop egy másik játékostól, aki le van bénulva.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: A virológus egy olyan mezőn áll, amin van legalább egy bénult játékos (önmagán kívül).</i></p> <p>1. A játékos a mezőn álló bénult játékosok közül választhat, hogy kitől lopjon.</p> <p>2/A. Ha a lopó virológusnak még kettőnél kevesebb védőfelszerelése van, és annak a játékosnak, akitől lop van legalább egy védőfelszerelése, akkor ezek közül választhat egyet, amit ellop.</p> <p>2/B. Egyébként nem tud védőfelszerelést lopni.</p> <p>3. A lopó játékos megkapja a támadott játékos anyagkészletéből azt a maximális mennyiséget, amivel még nem lépi túl az anyagkészletének felső korlátját.</p>

Use-case neve	forgetCodesViaAmnesiaVirus
Rövid leírás	Amnéziavírussal kent játékos vélegesen elfelejt az összes eddig megtanult genetikai kódját.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: A viroláguson van egy aktív amnéziavírus.</i></p> <p>1. A játékos köre elején az összes eltárolt genetikai kód törlődik.</p> <p>2. A játékos kijátszhat különböző akciókat, megy tovább a játék.</p>

Use-case neve	enlargeTheBag
Rövid leírás	Az a játékos, akinek van egy zsák az védőfelszerelései között, több anyagot tud felvenni.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: A virolágusnak van egy zsák védőfelszerelése.</i></p> <p>1. A játékos köre elején az Inventory kapacitása megkétszereződik.</p> <p>2. Lopás vagy anyagfelvételkor ezt a megnövelt korlátot kell betartania az anyag mennyiségeknek.</p> <p>3. A kör végén a kapacitás feleződik, azaz visszaáll a régi értékre.</p>

Use-case neve	stunnedPlayerMissesTurn
Rövid leírás	Bénítóvírussal kent játékos nem tud semmilyen akciót végrehajtani a körében.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: A virológuson van egy aktív bénítóvírus.</i></p> <p>1. A játékos köre elején automatikusan addig passzol, amíg el nem fogynak az akciói.</p> <p>2. A játékoson aktív többi hatás ilyenkor is érvényes, idejük fogy.</p> <p>3. A következő játékos kerül sorra.</p>

Use-case neve	playerVitusDances
Rövid leírás	Vitustánc vírussal kent játékos automatikusan véletlenszerű lépéstre használja el az összes akcióját.
Aktorok	Player
Forgatókönyv	<p><i>Kiindulási feltétel: A virológuson van egy aktív vitustánc vírus.</i></p> <p>1. A játékos köre elején automatikusan addig lép egy véletlenszerű, szomszédos mezőre, amíg el nem fogynak az akciói.</p> <p>2. A játékoson aktív többi hatás ilyenkor is érvényes, idejük fogy.</p> <p>3. A következő játékos kerül sorra.</p>

5.2 A szkeleton kezelői felületének terve, dialógusok

5.2.1 Szkeleton program felépítése

A program indításakor először megjeleníti a lehetséges forgatókönyveket, ami közül a felhasználónak választania kell a program bemeneti követelményei szerint.

Van egy Initializer singleton osztály, aminek statikus metódusai végzik az összes kiírást és a bemenetek kezelését. Az egyes use-case-ekben az objektumok inicializálásakor minden objektumhoz tárol egy egyedi azonosítót, ez meg fog jelenni, amikor az objektum metódusai meghívódnak.

5.2.2 Szkeleton program kimenete

Kiiratás során az első kiírt sor a kiíró osztály neve, azonosítója és a függvény, amit hívtak rajta paraméterekkel. A program kiírja a meghívott függvény nevét. Amennyiben szükség van valami döntésre a felhasználótól, akkor bekéri a bemenetet.

A kimenetek formátuma:

Függvényhívás: [Osztálynév Azonosító] [metódus neve(paraméterek)]

Visszatérés: return [visszatérési érték típusa]

Kérdés felhasználótól: {Kérdés(y/n)}: válasz

A függvények tabulálása a következőképpen történik:

Függvényhívás után egy tabulátornyi helyel beljebb kezdődik a következő sor egészen a visszatérésig. A kérdések minden tabulálás nélkül történnek.

Pl.

```
[Virologist v] [moveTo(EmptyTile tile)]
    [EmptyTile tile] [removeVirologist(Virologist v)]
    return [void]
    [EmptyTile newtile] [addVirologist(Virologist v)]
    return [void]
return [void]
```

5.2.3 Szkeleton program bemenete

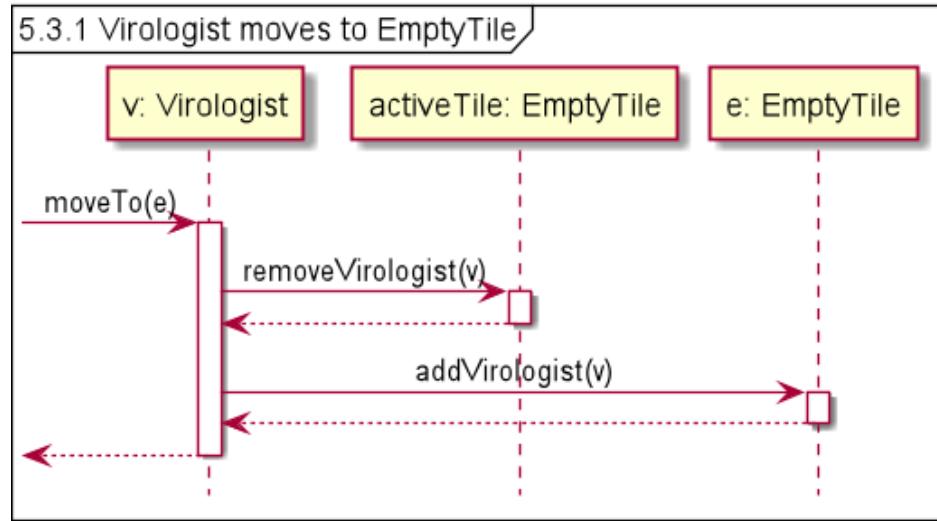
A szkeleton program az eldöntendő kérdésekre y/n bemenetet fogad el és kezel. Több válaszlehetőség esetén a program által megjelenített sorszámot fogadja el a program érvényes bemenetként.

Pl: Milyen védfelszerelései legyenek?

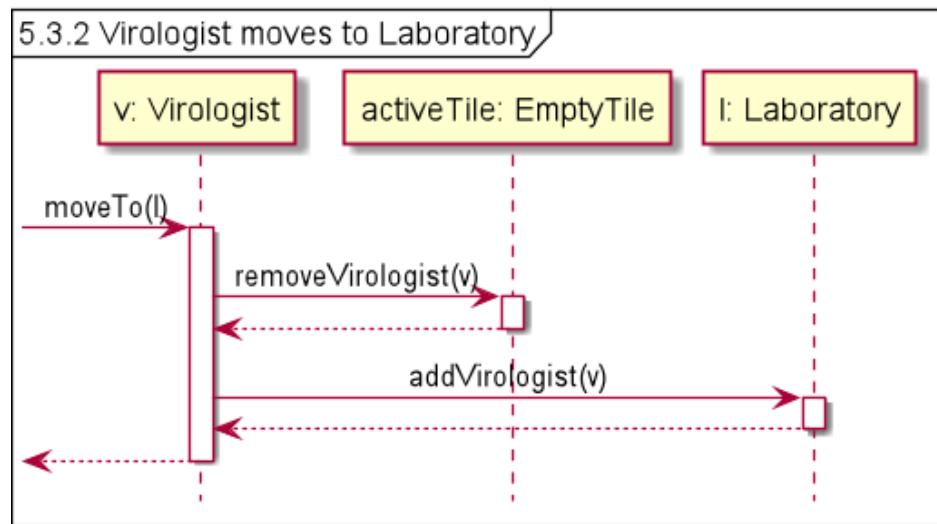
5. *Glove*
6. *Cloak*
7. *both*
8. *neither*

5.3 Szekvencia diagramok a belső működésre

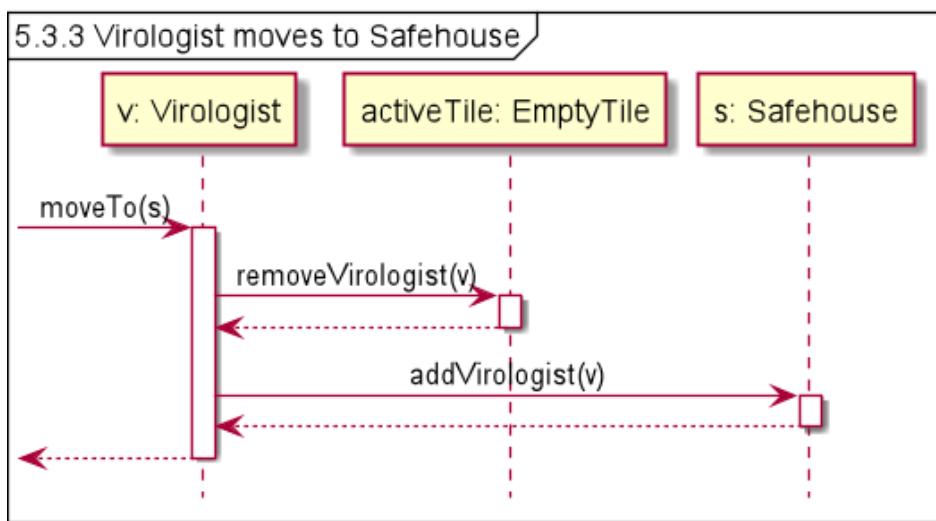
5.3.1 moveToEmptyTile



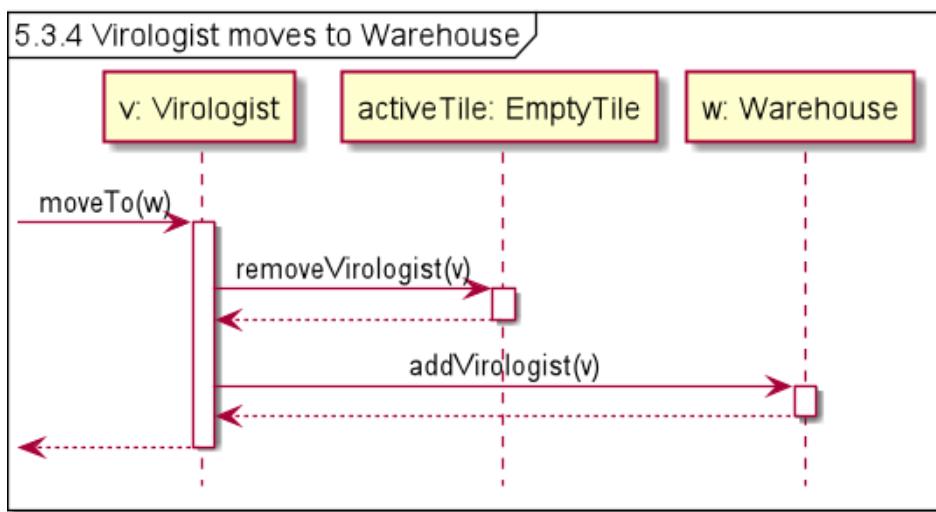
5.3.2 moveToLaboratory



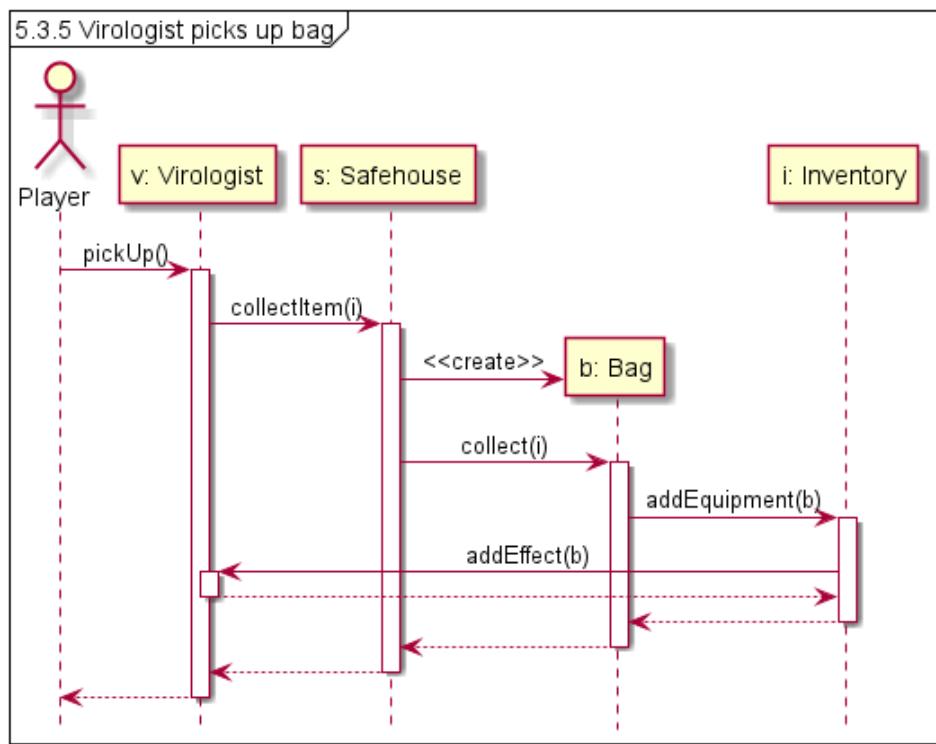
5.3.3 moveToSafehouse



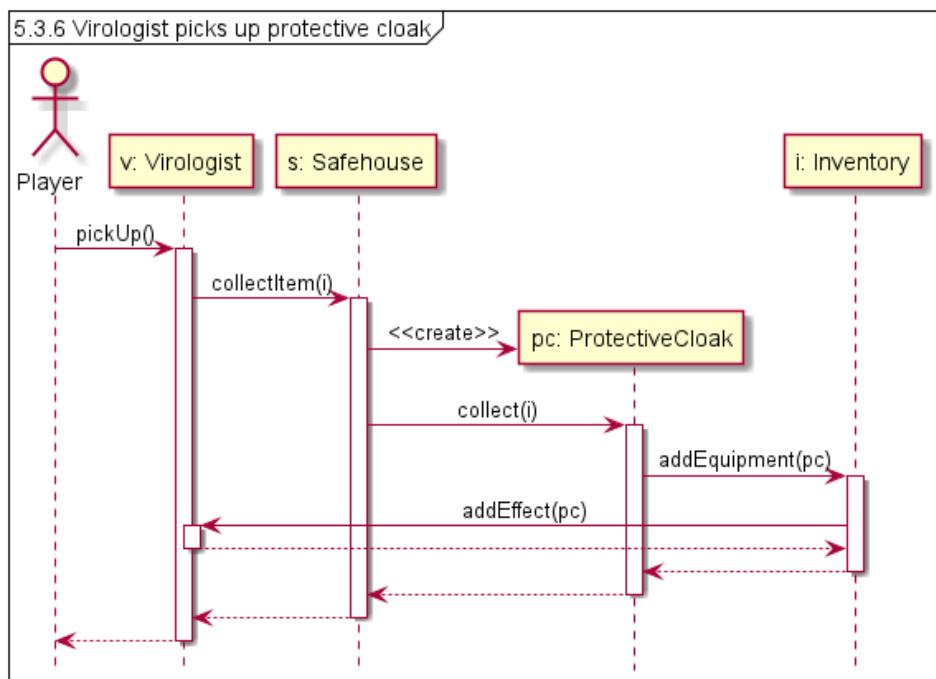
5.3.4 moveToWarehouse



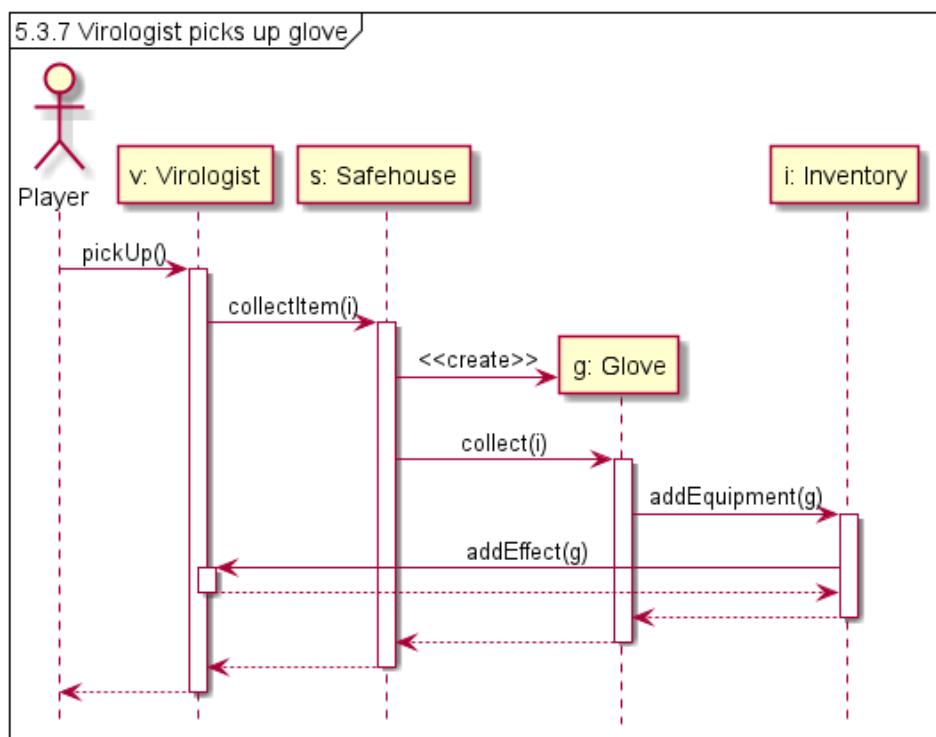
5.3.5 pickUpBag



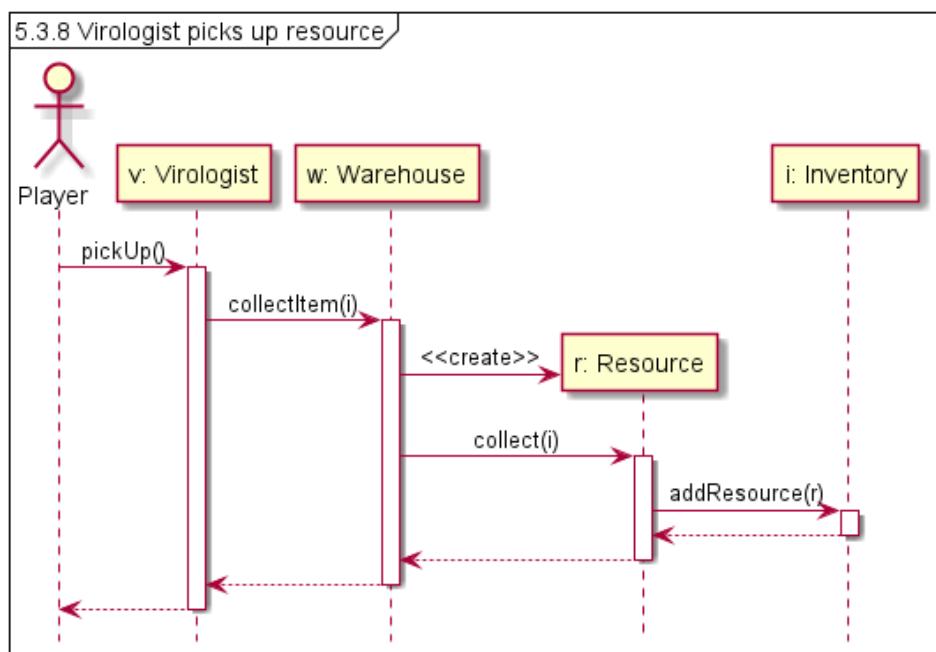
5.3.6 pickUpCloak



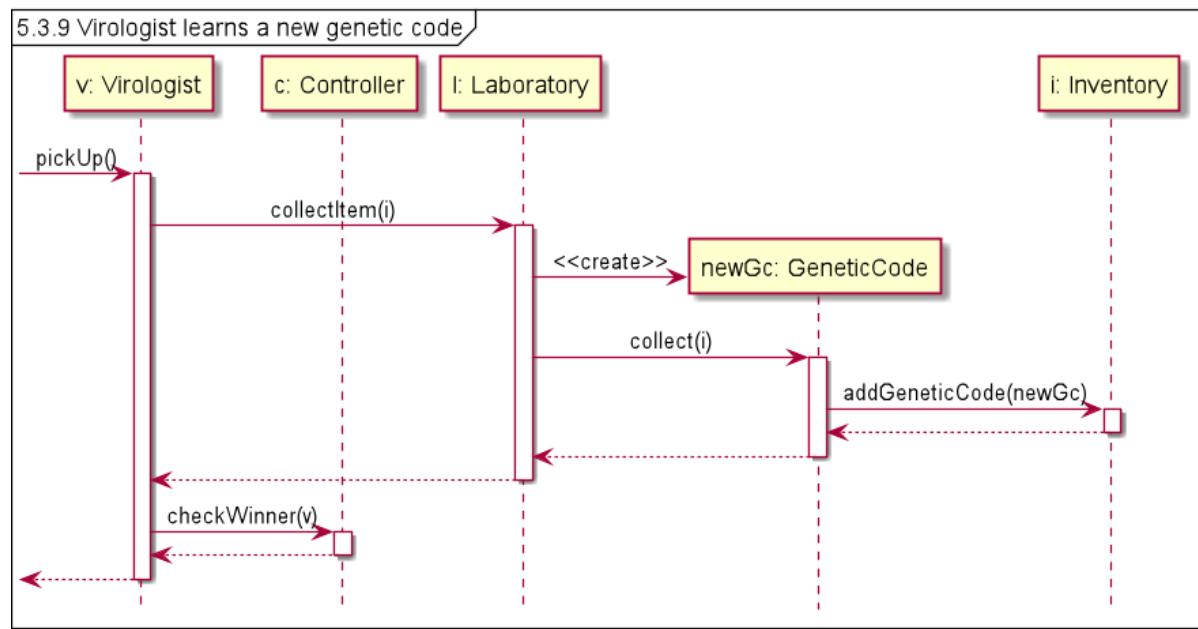
5.3.7 pickUpGlove



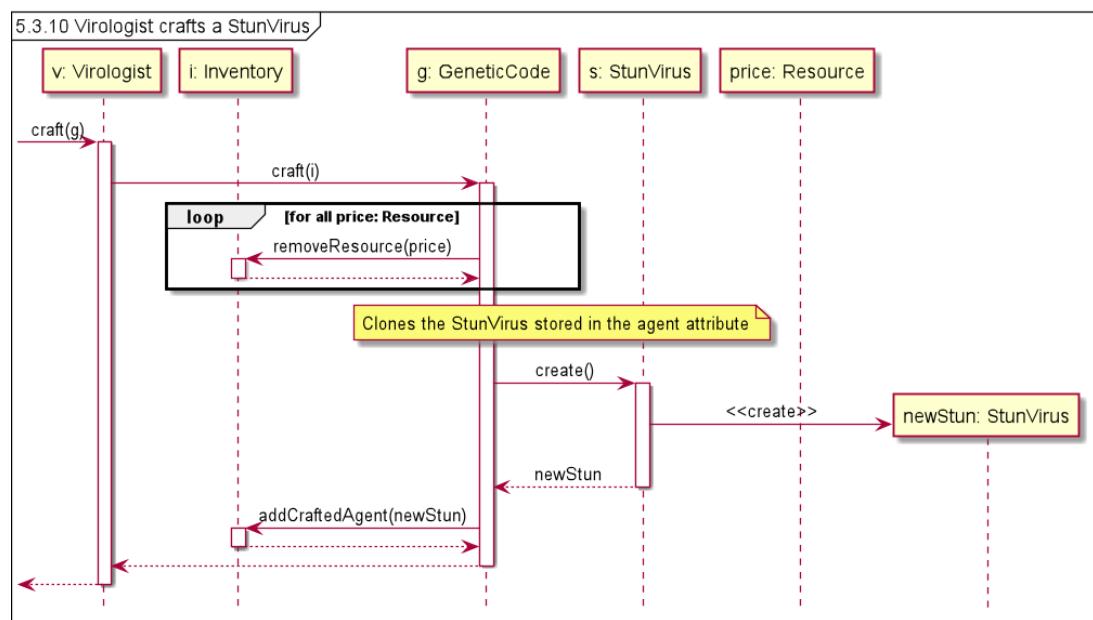
5.3.8 pickUpResource



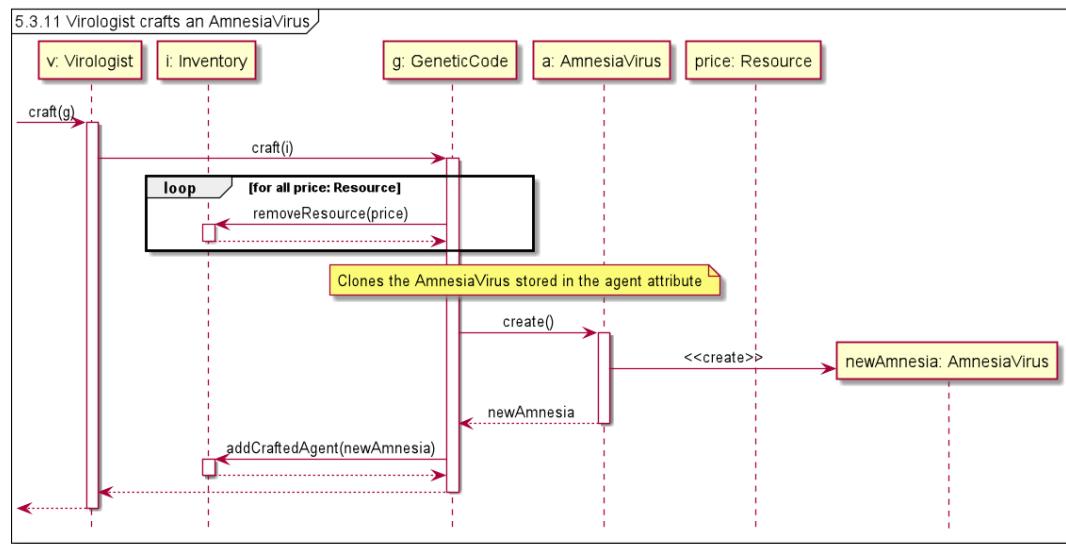
5.3.9 learnGeneticCode



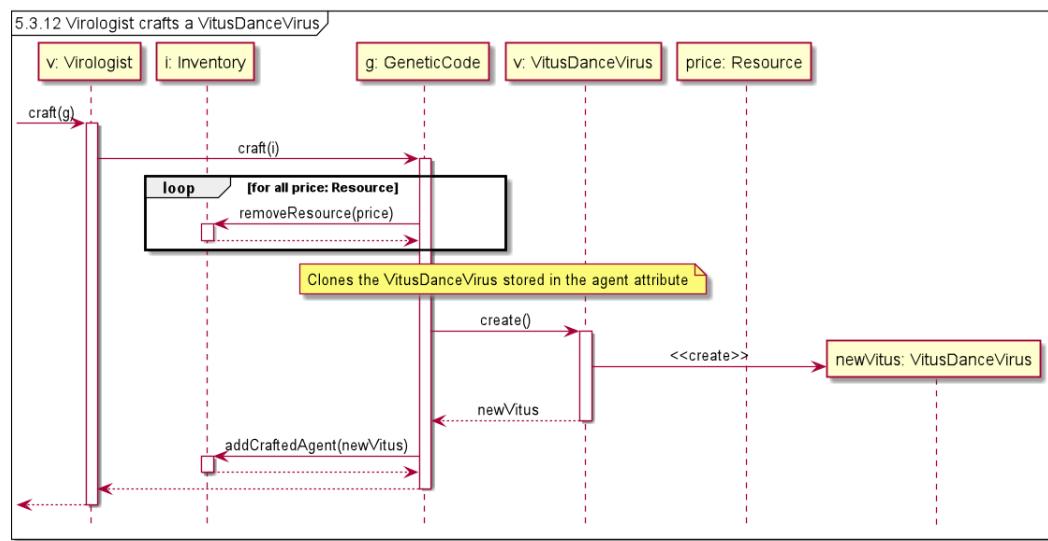
5.3.10 craftStunVirus



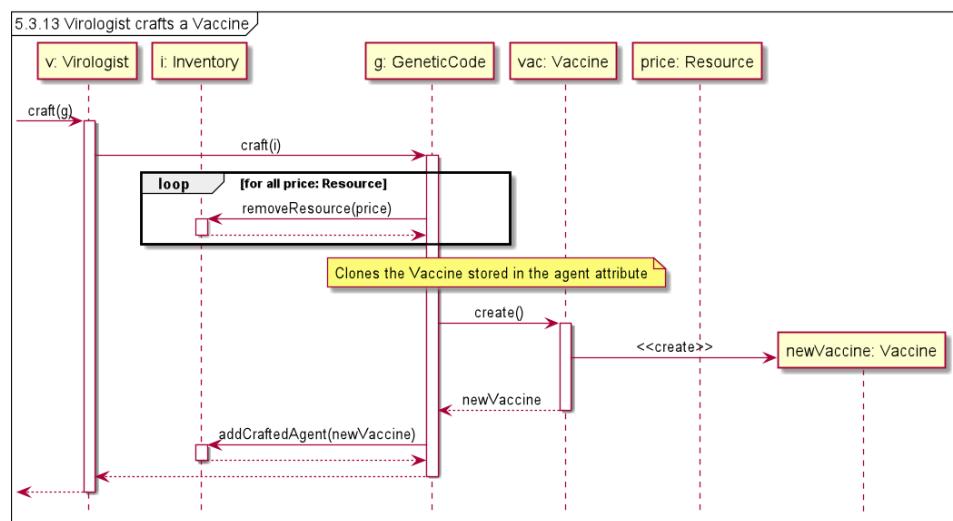
5.3.11 craftAmnesiaVirus



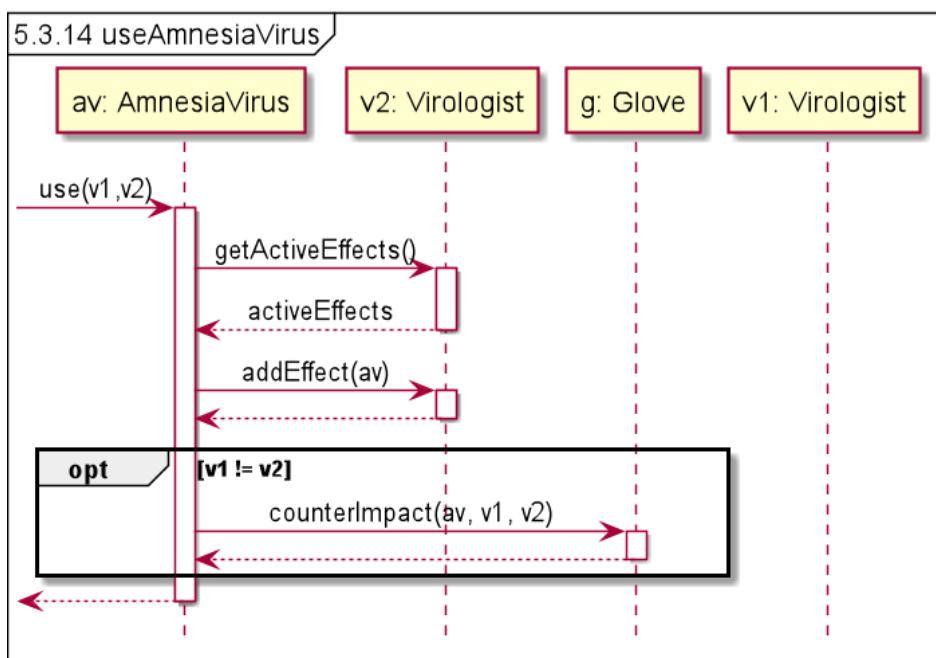
5.3.12 craftVitusDanceVirus



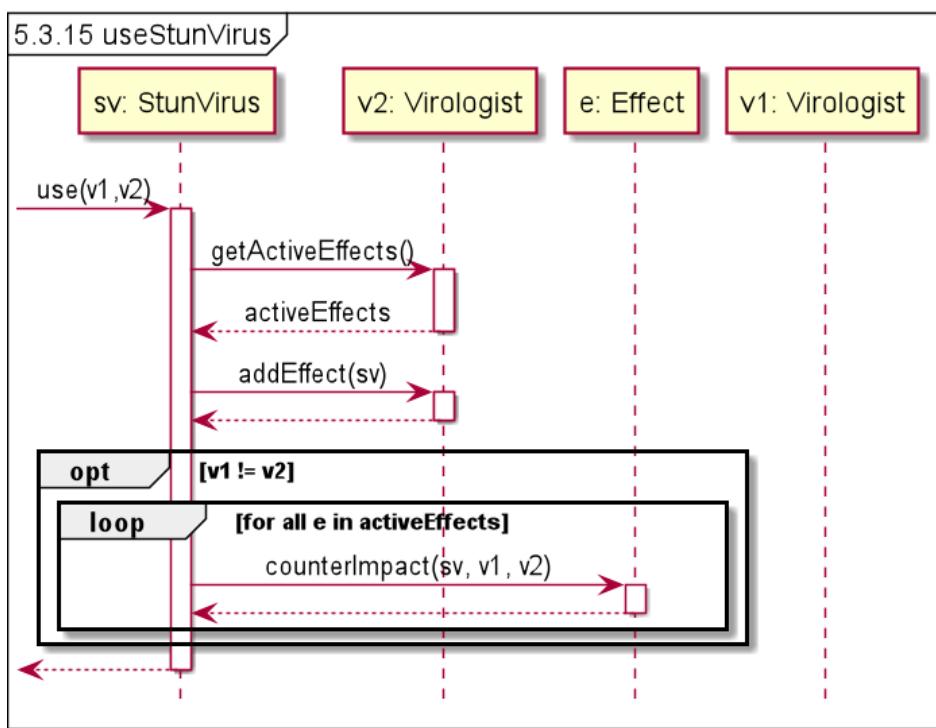
5.3.13 craftVaccine



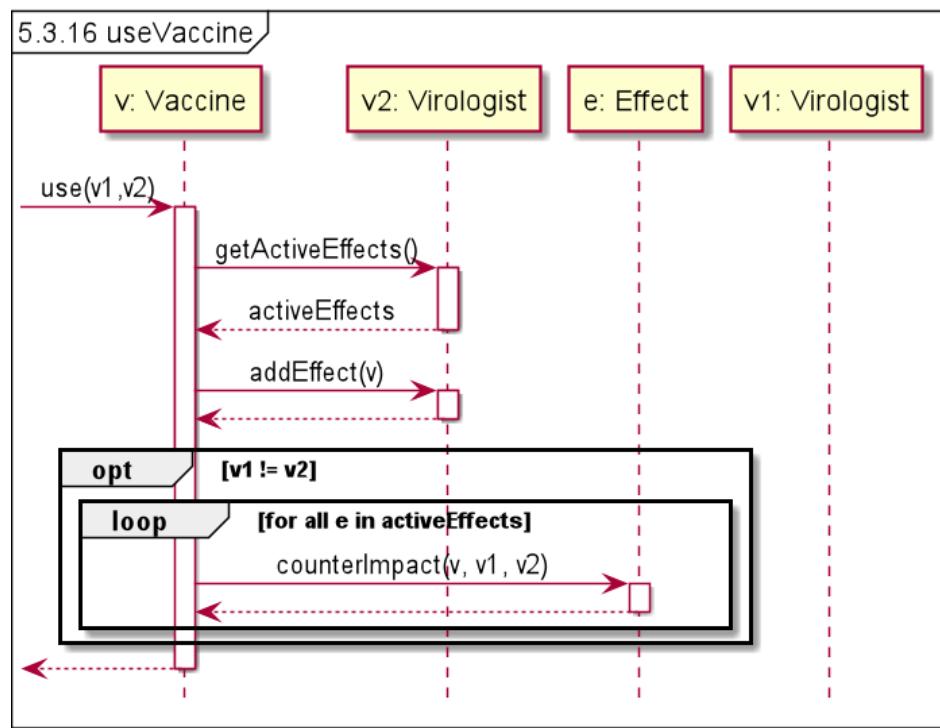
5.3.14 useAmnesiaVirus



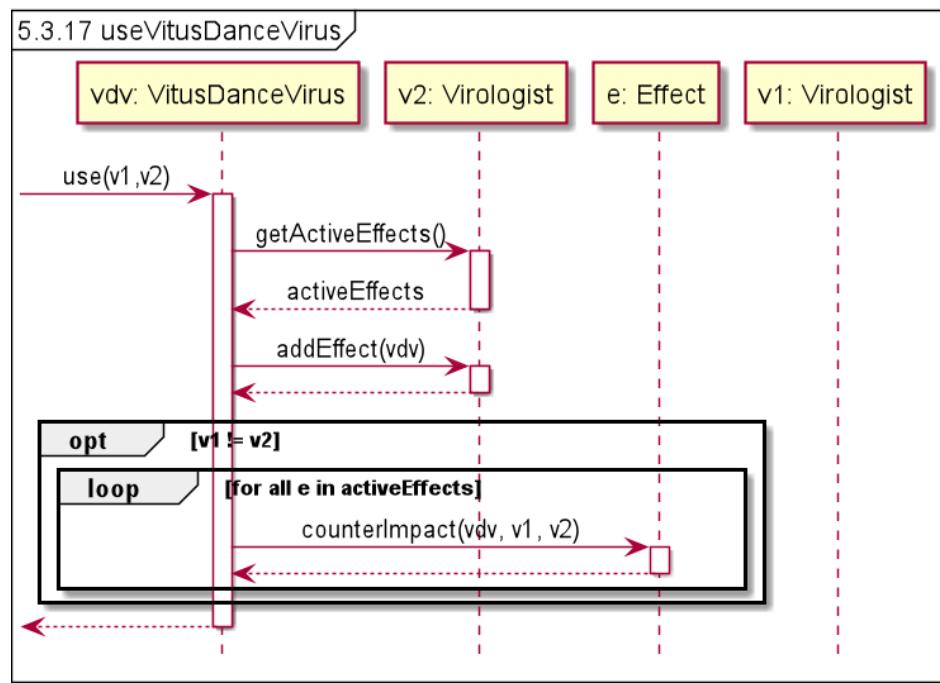
5.3.15 useStunVirus



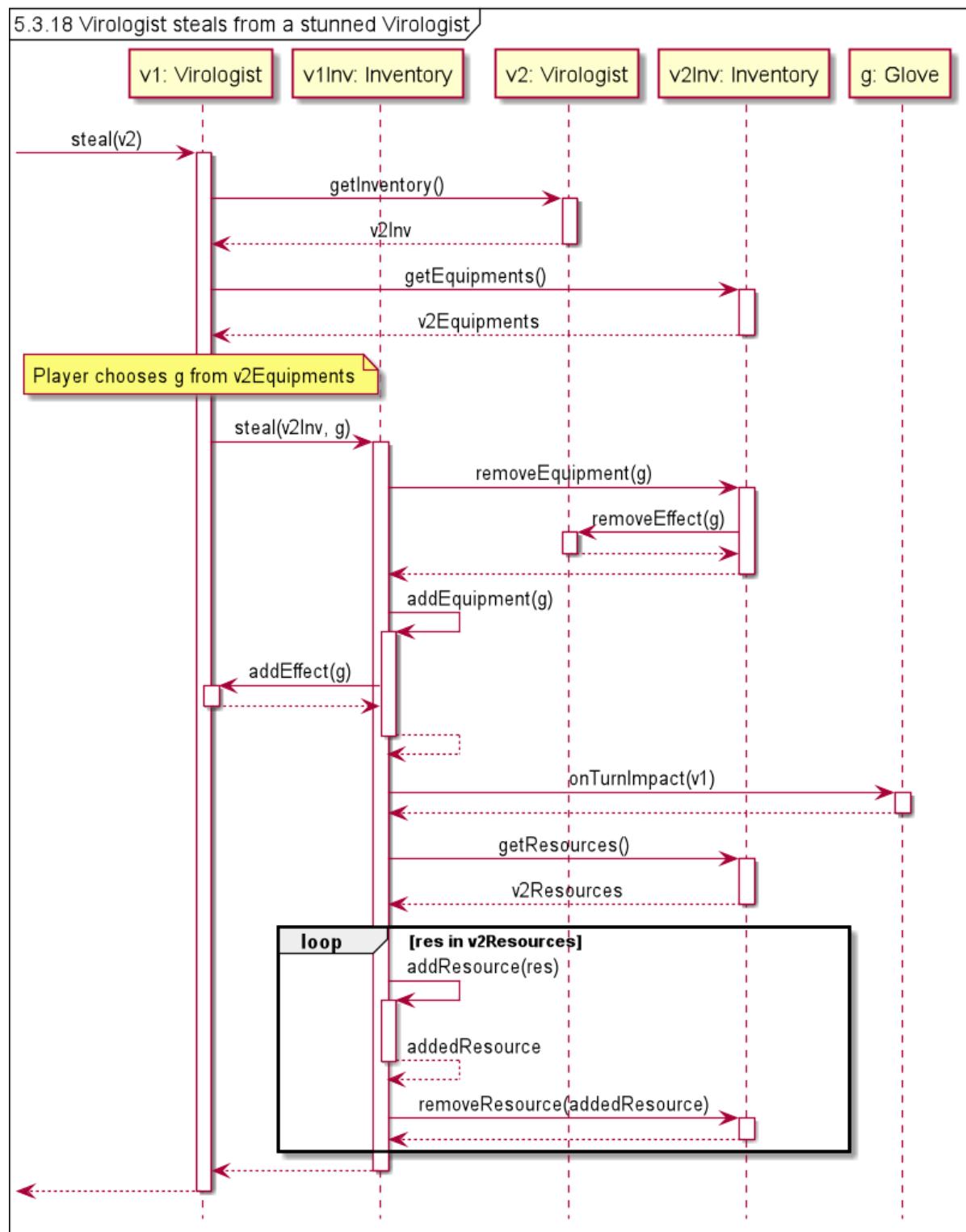
5.3.16 useVaccine



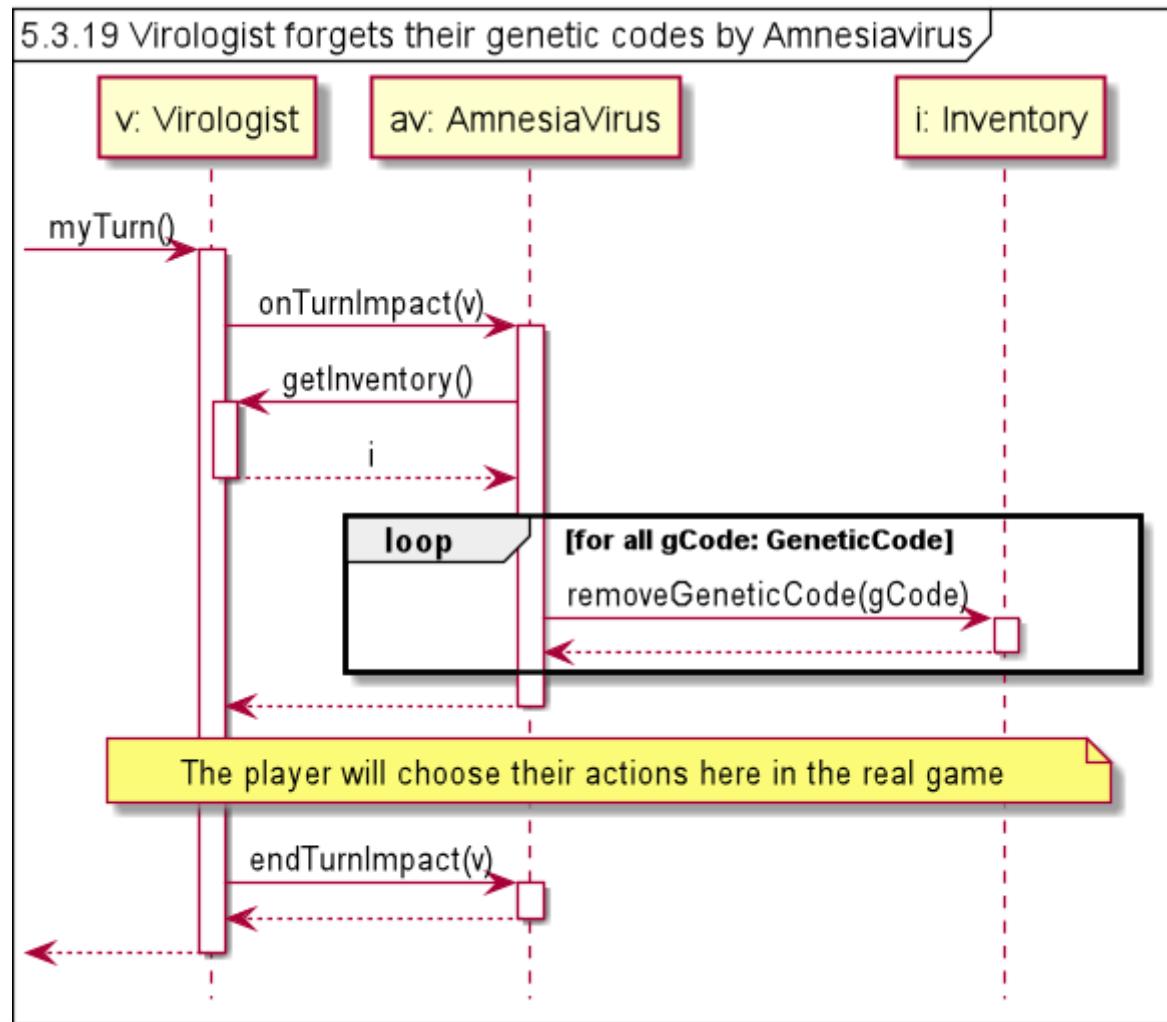
5.3.17 useVitusDanceVirus



5.3.18 robVirologist

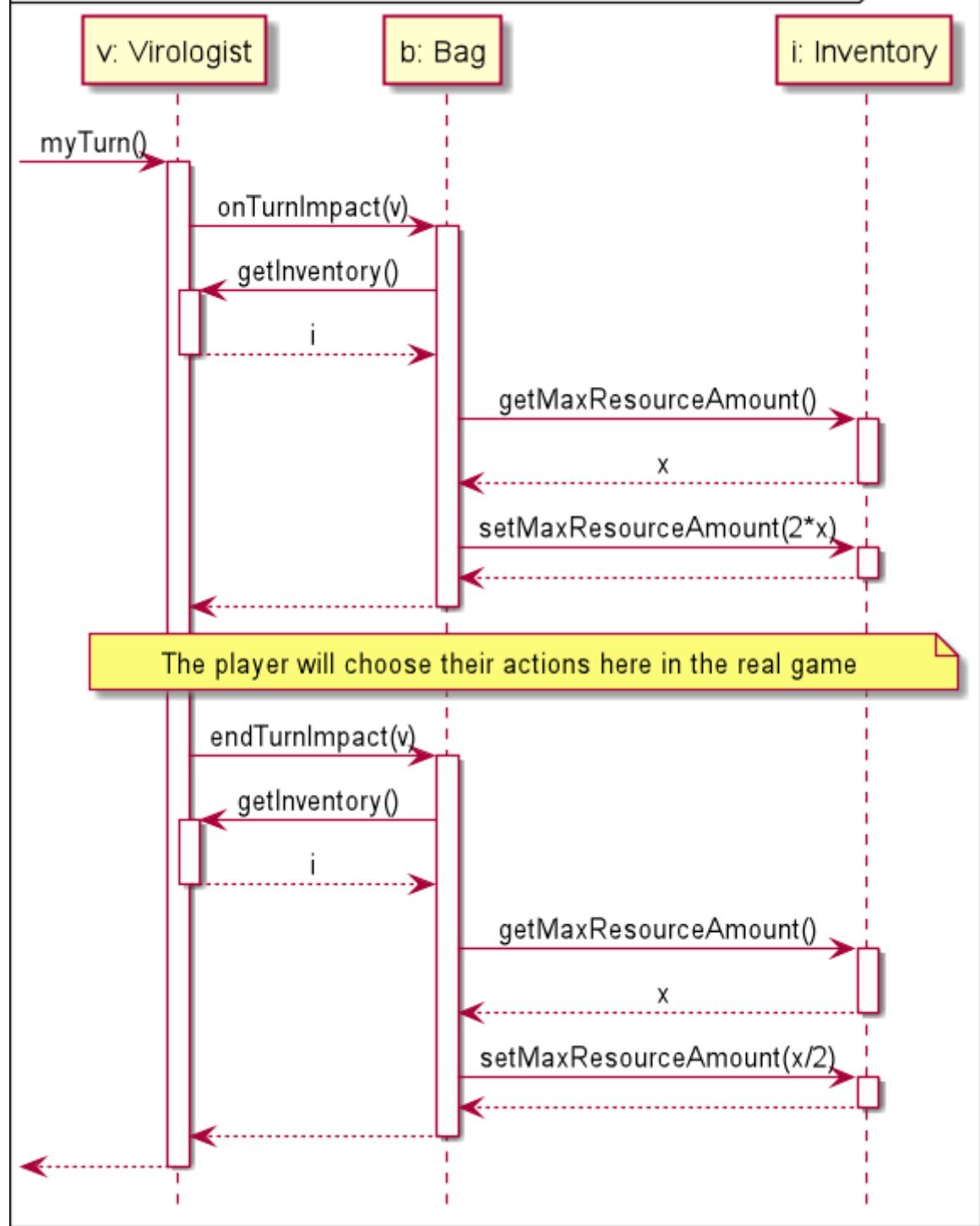


5.3.19 forgetCodesViaAmnesiaVirus

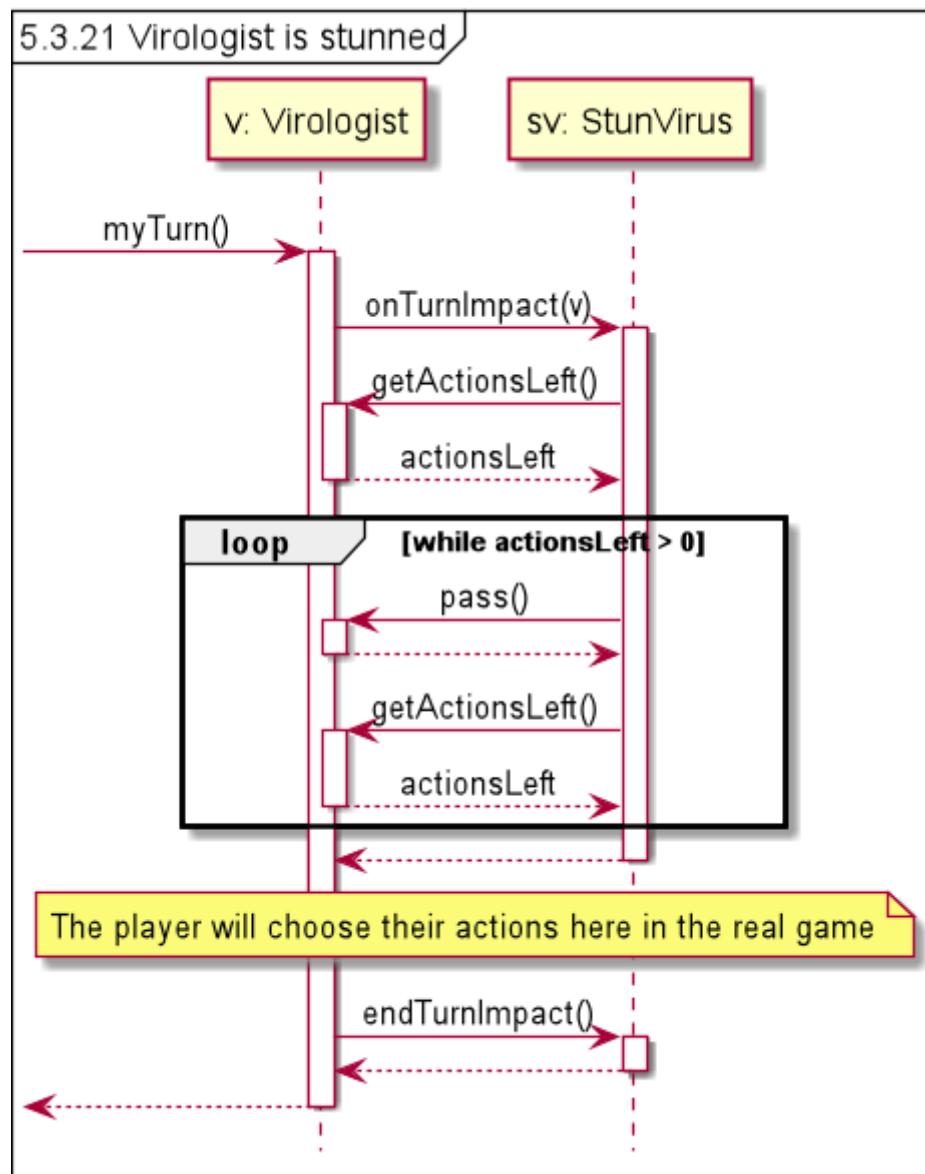


5.3.20 enlargeTheBag

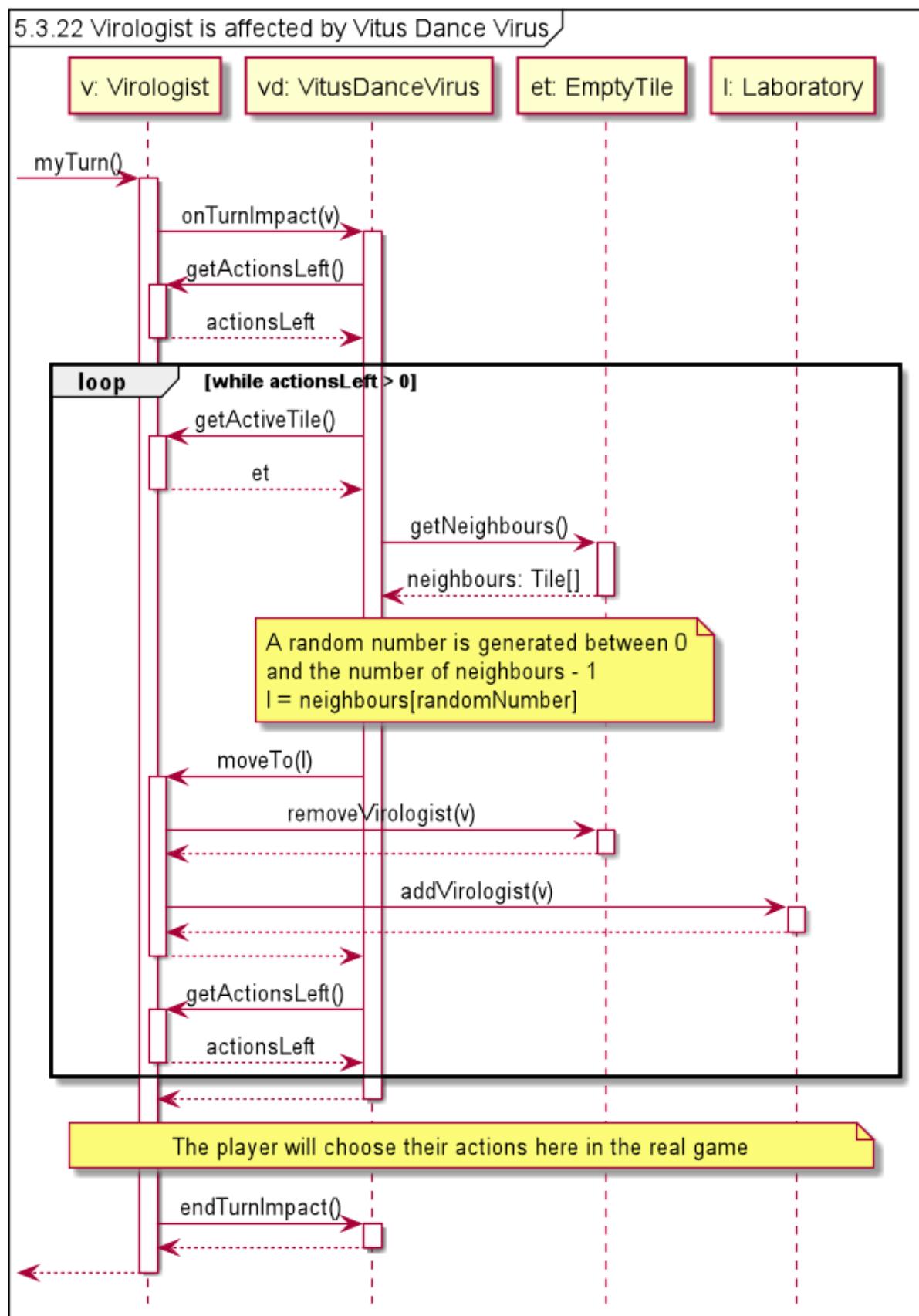
5.3.20 Virologist can pick up more resources with enlarged Bag.



5.3.21 stunnedPlayerMissesTurn

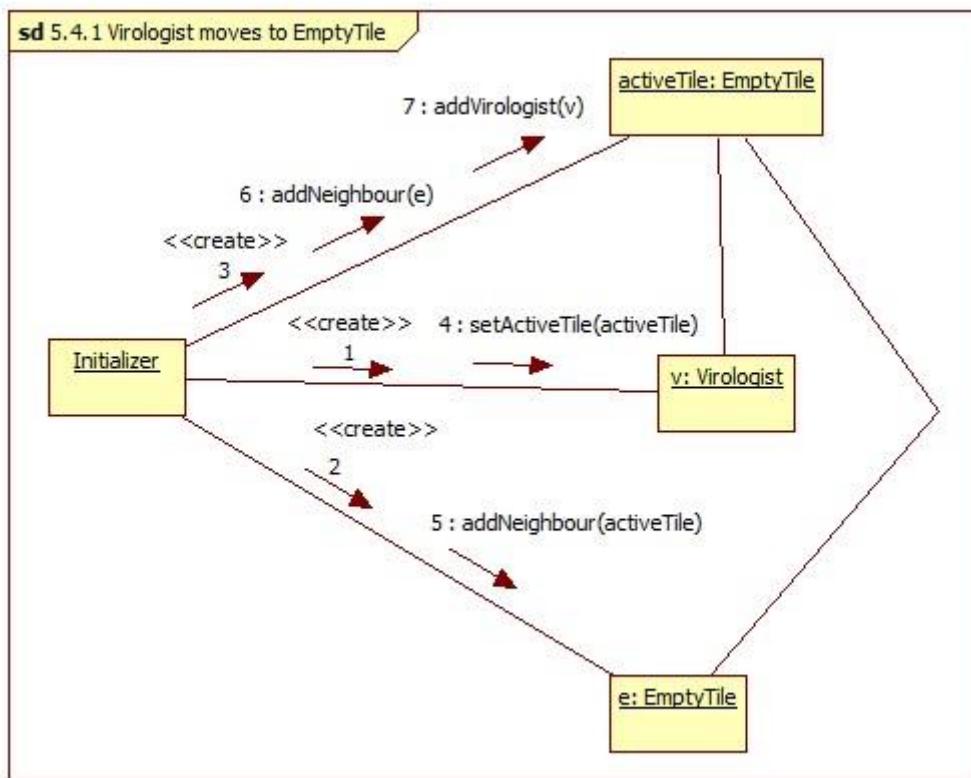


5.3.22 playerVitusDances

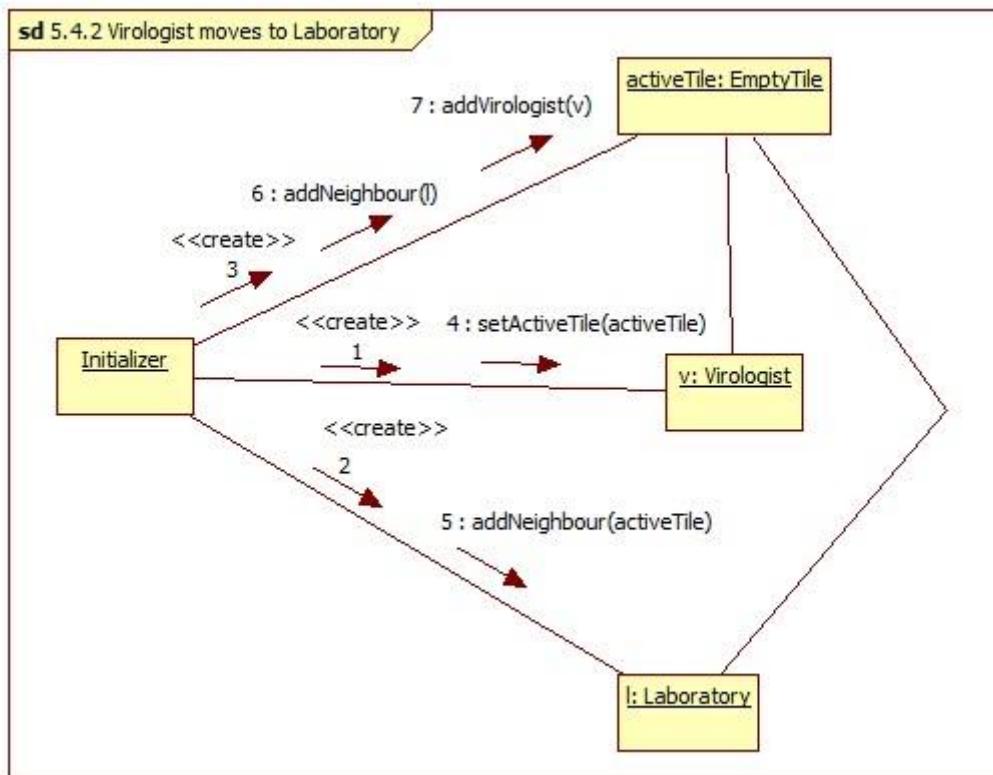


5.4 Kommunikációs diagramok

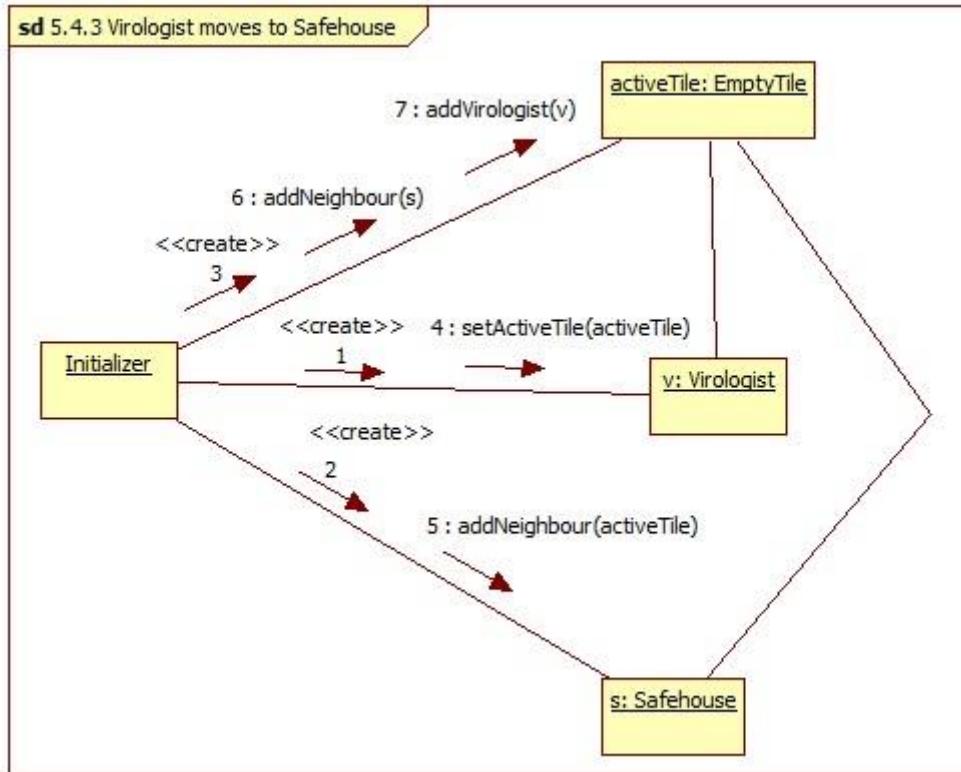
5.4.1 moveToEmptyTile



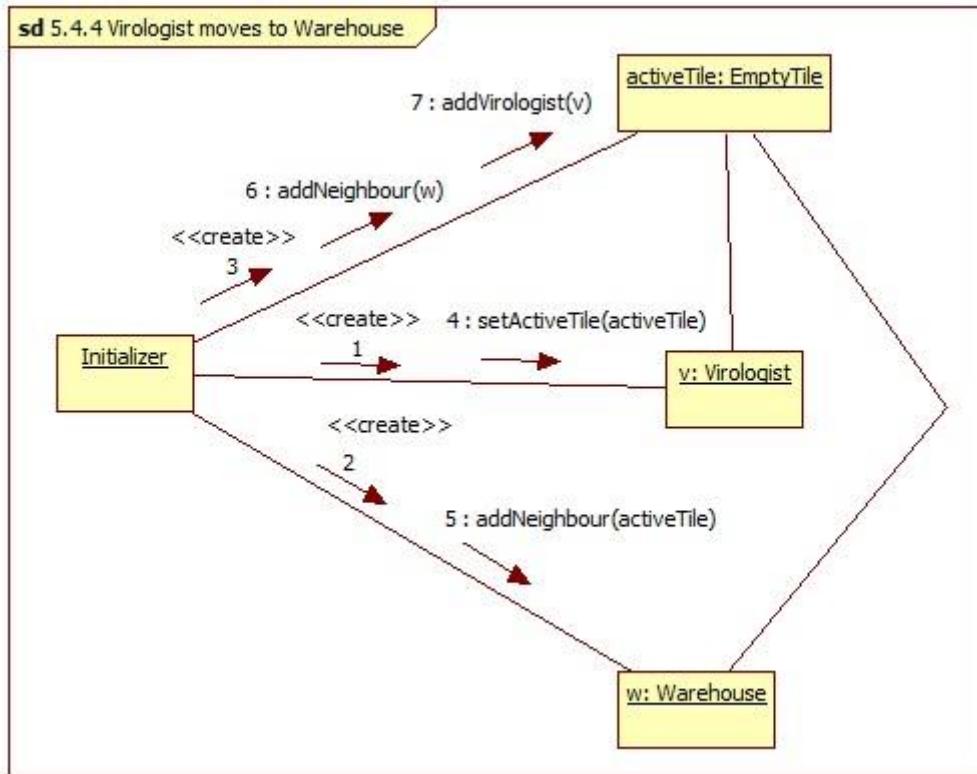
5.4.2 MoveToLaboratory



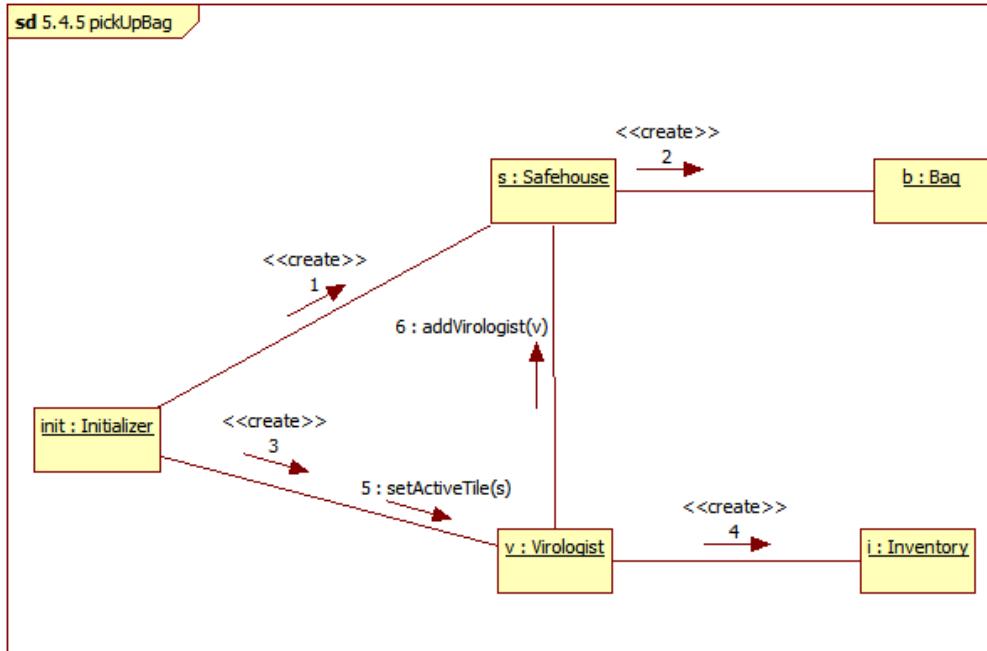
5.4.3 moveToSafehouse



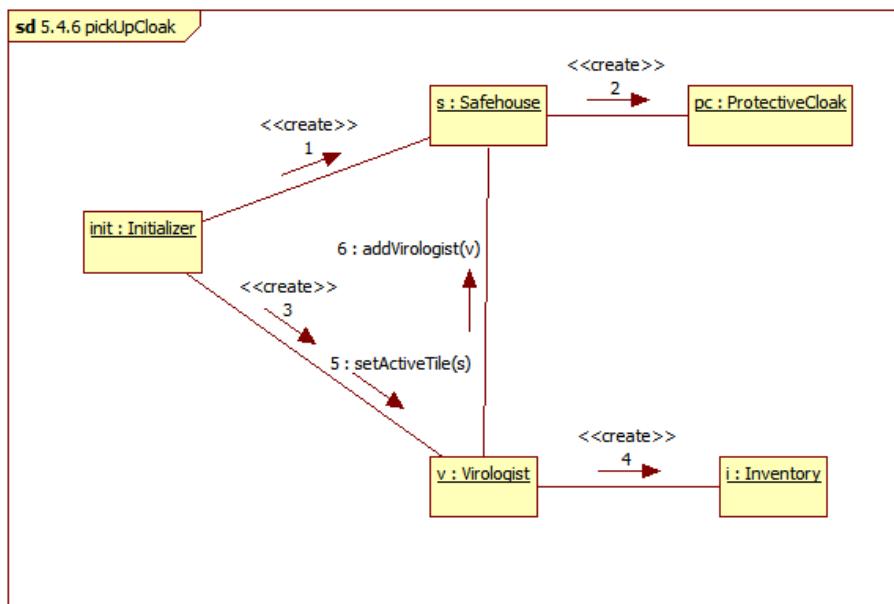
5.4.4 moveToWarehouse



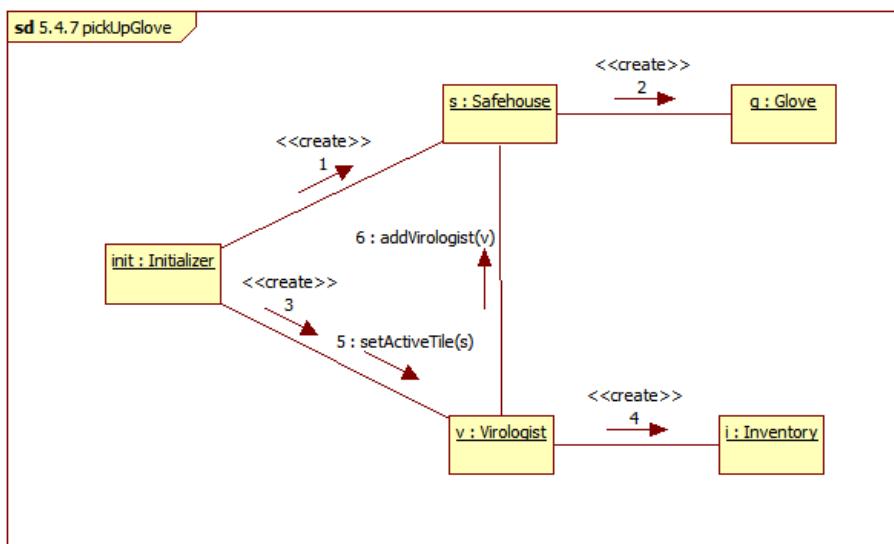
5.4.5 pickUpBag



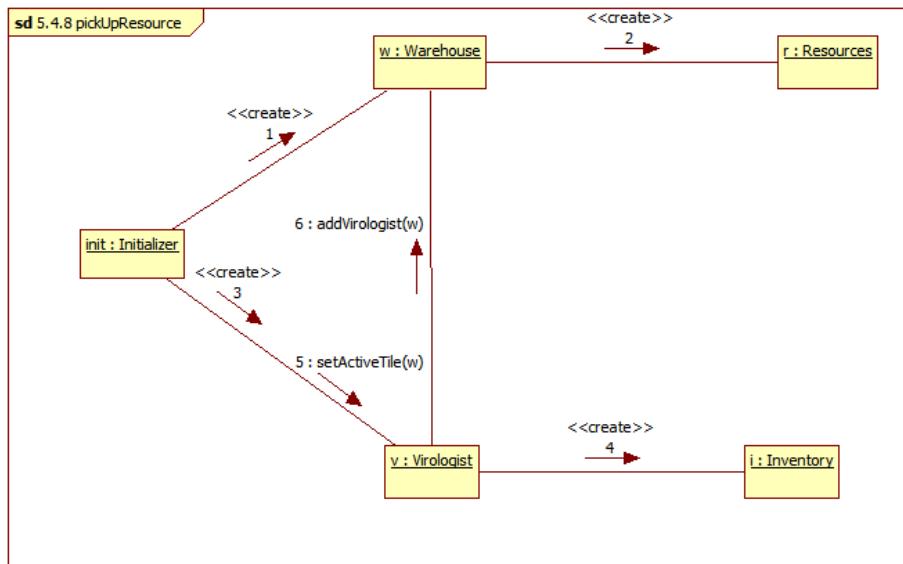
5.4.6 pickUpCloak



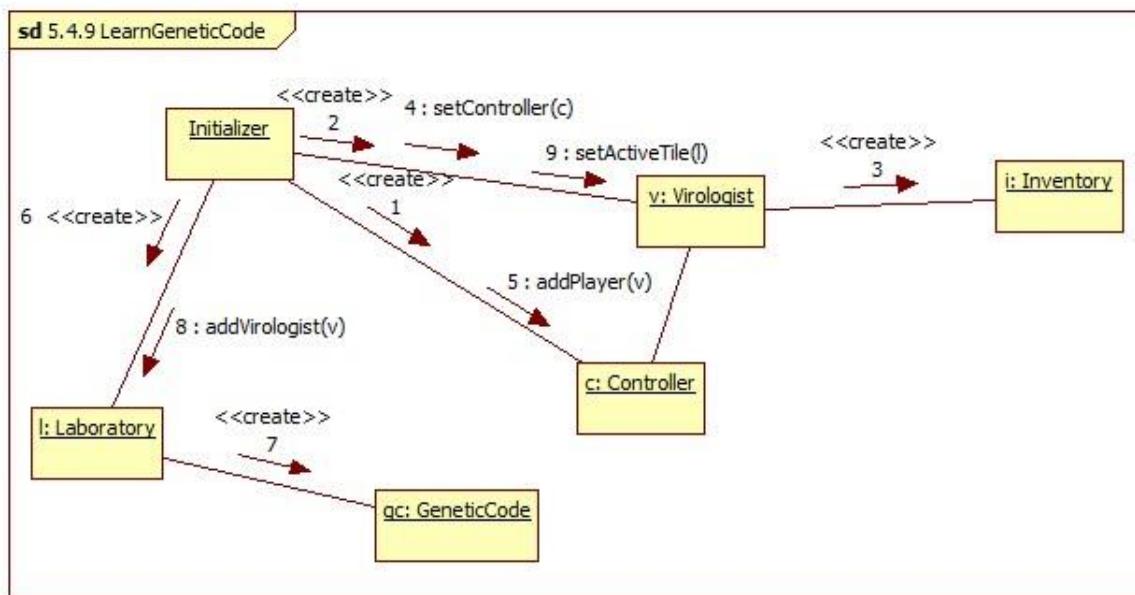
5.4.7 pickUpGlove



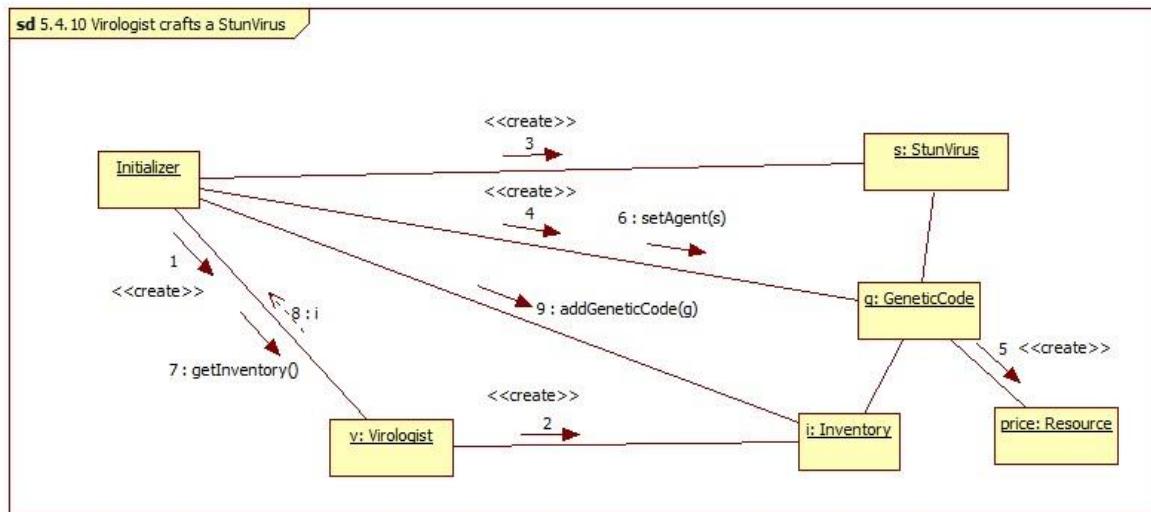
5.4.8 pickUpResource



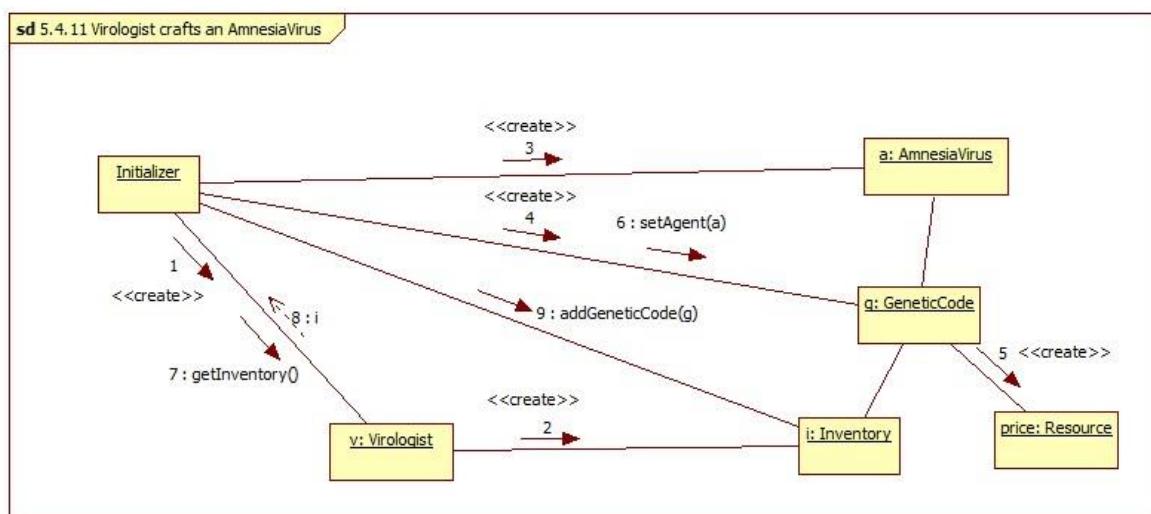
5.4.9 learnGeneticCode



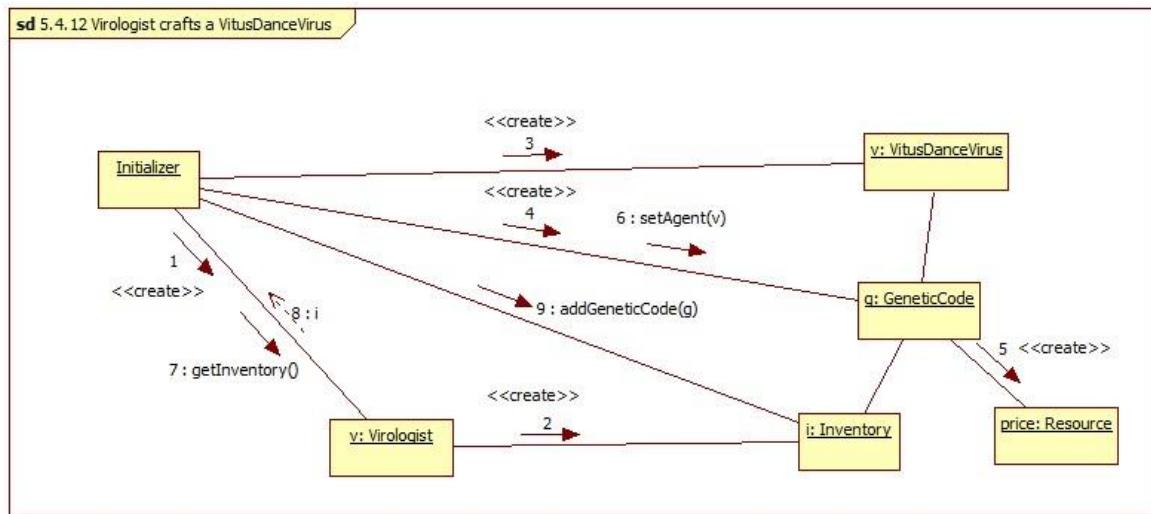
5.4.10 craftStunVirus



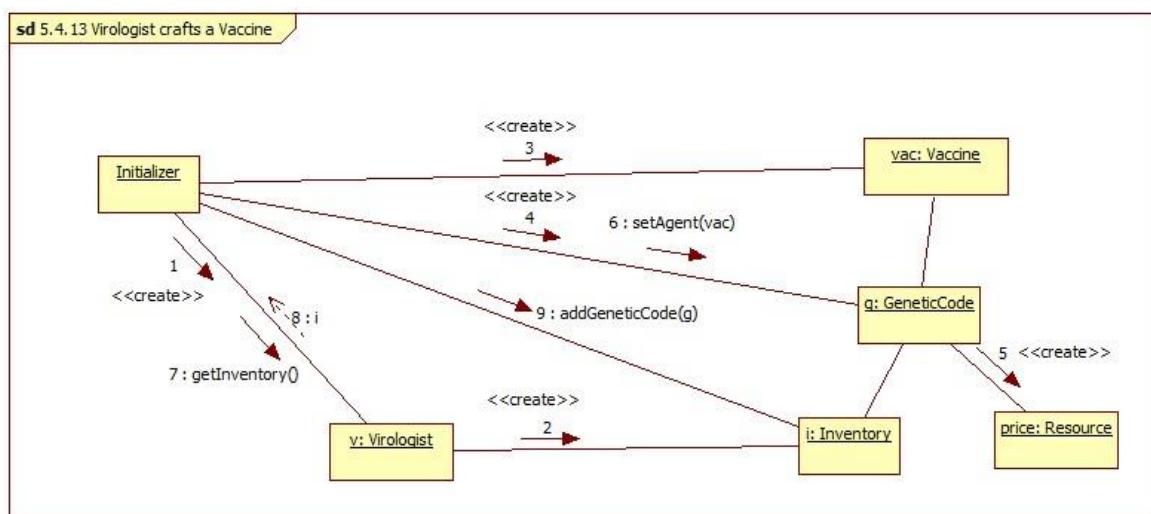
5.4.11 craftAmnesiaVirus



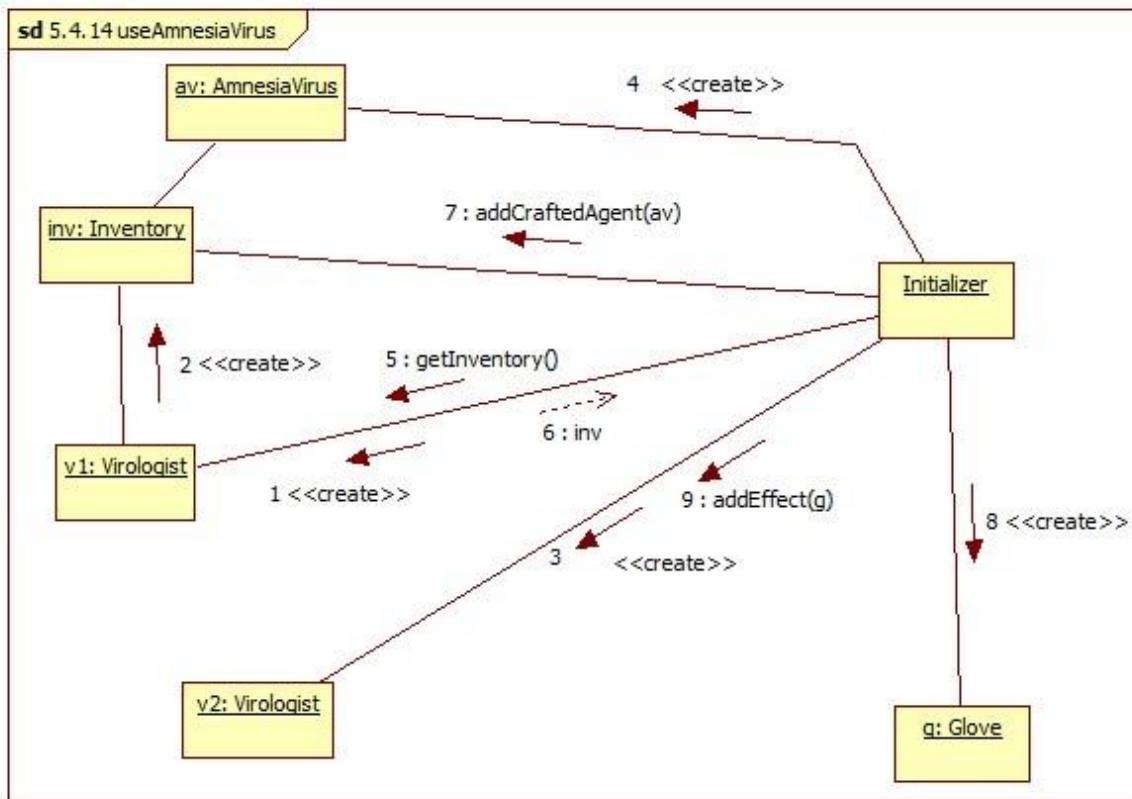
5.4.12 craftVitusDanceVirus



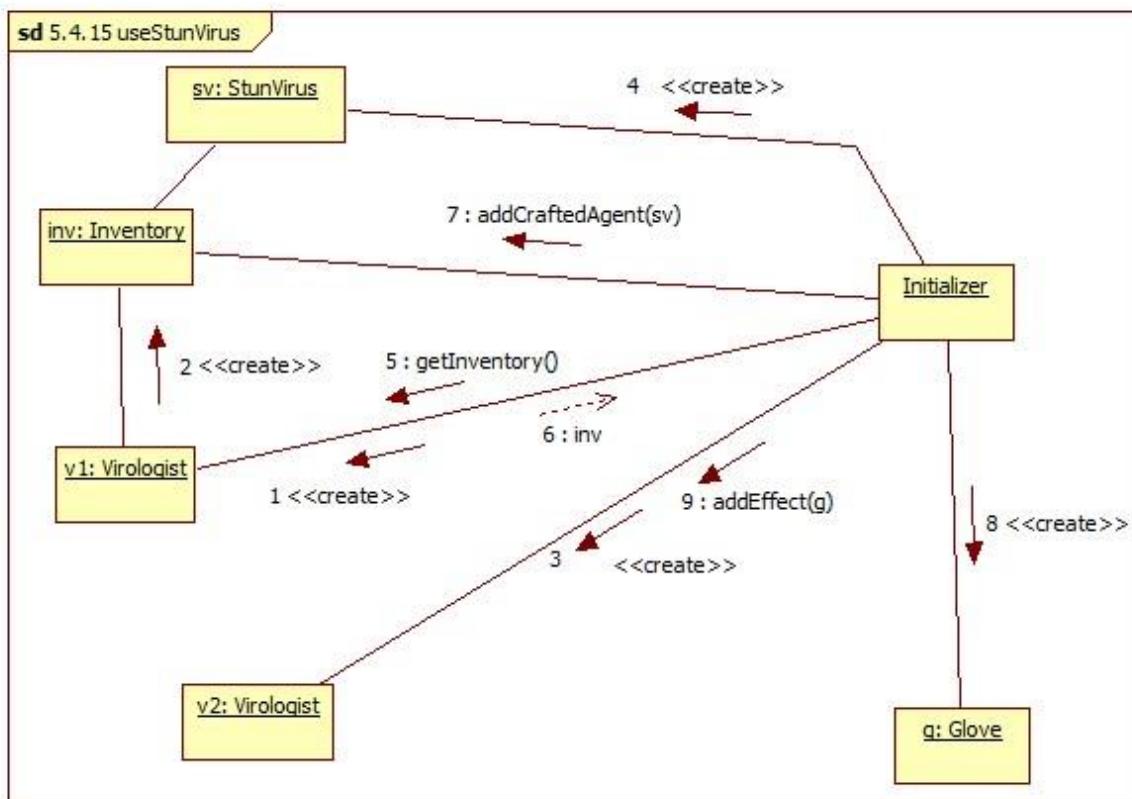
5.4.13 craftVaccine



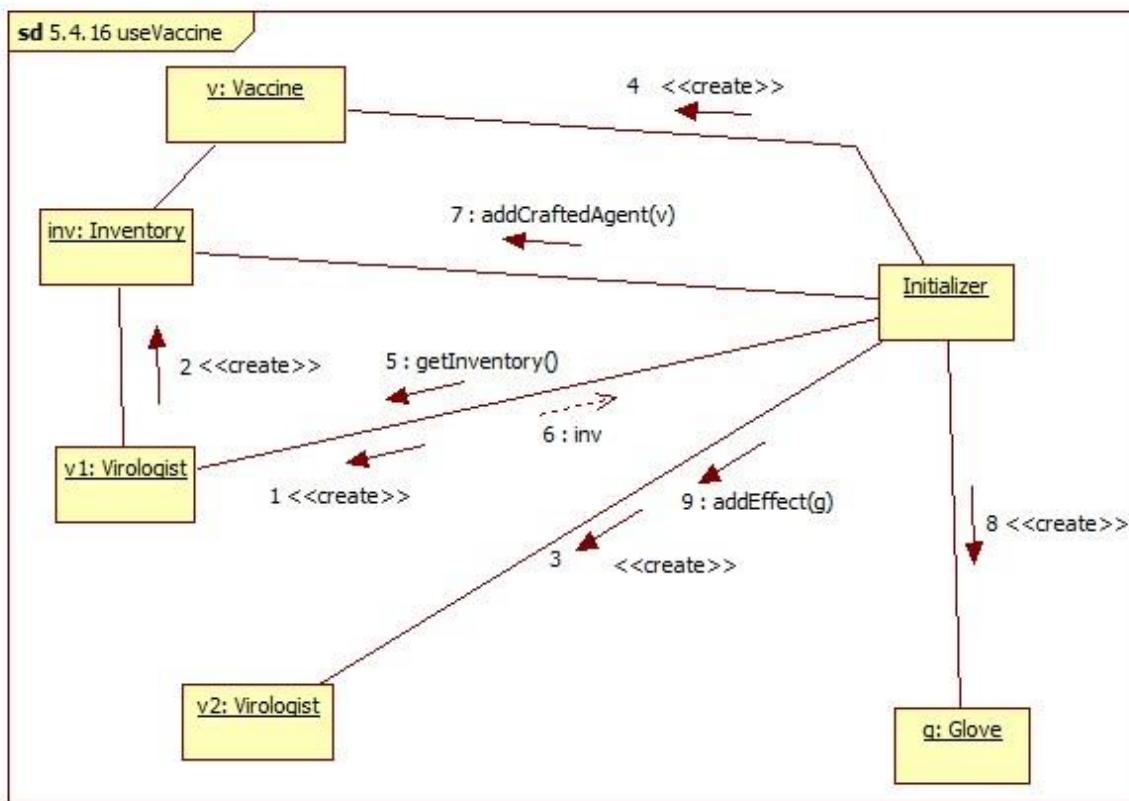
5.4.14 useAmnesiaVirus



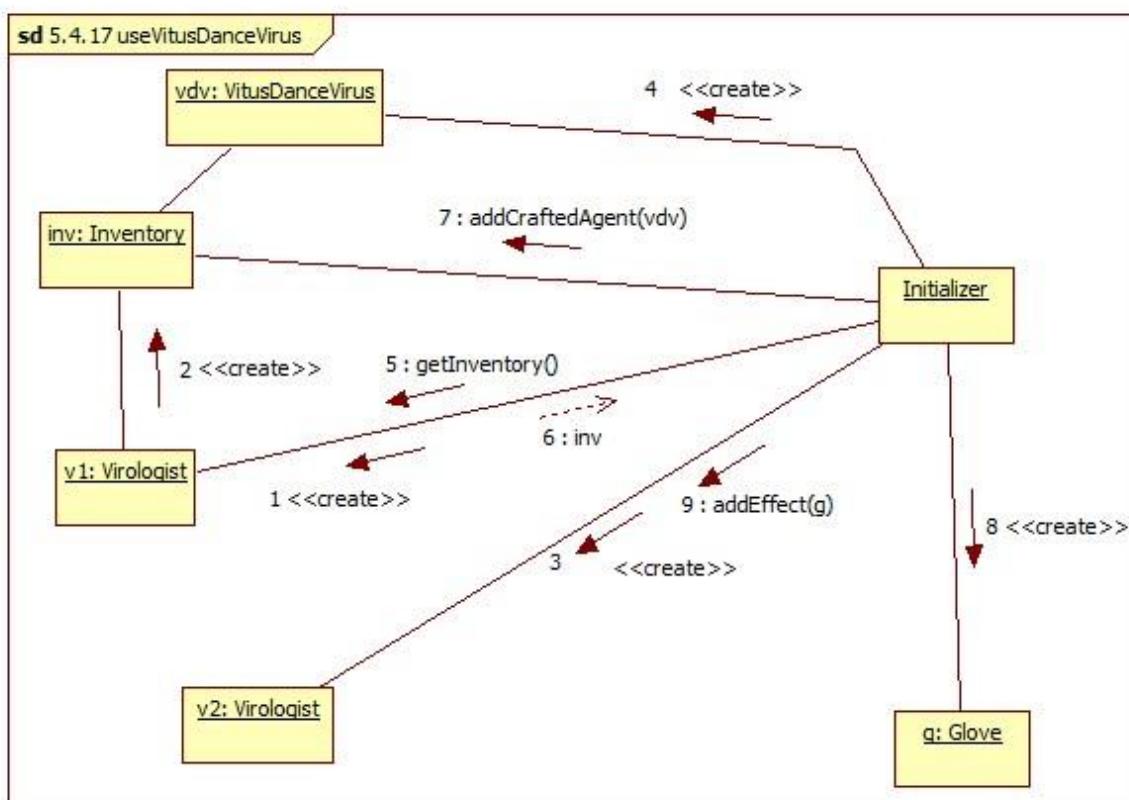
5.4.15 useStunVirus



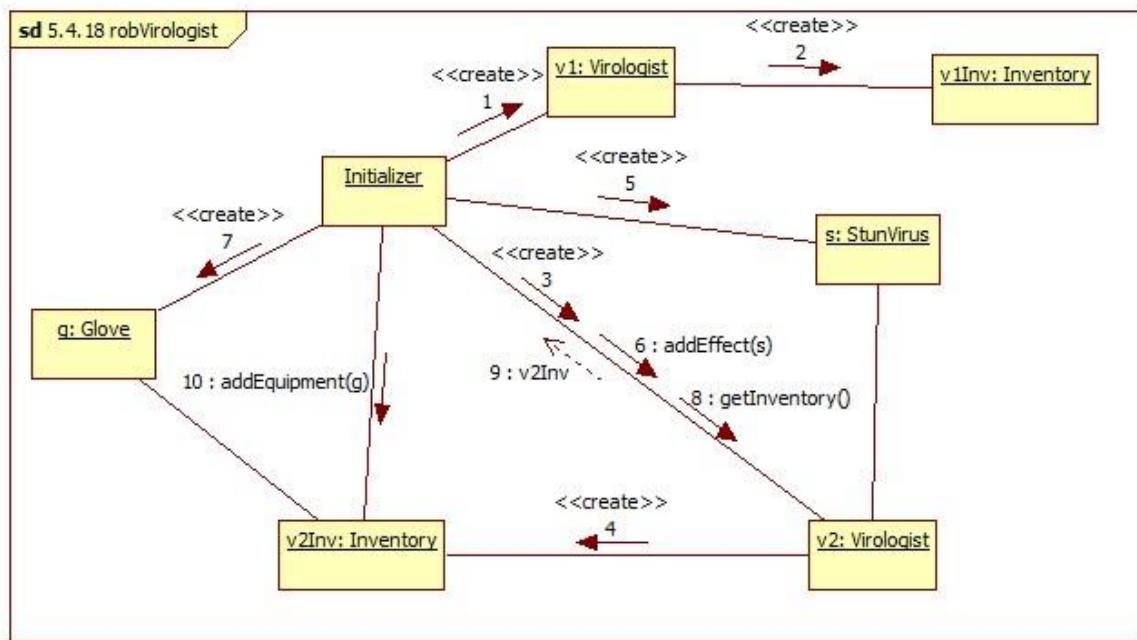
5.4.16 useVaccine



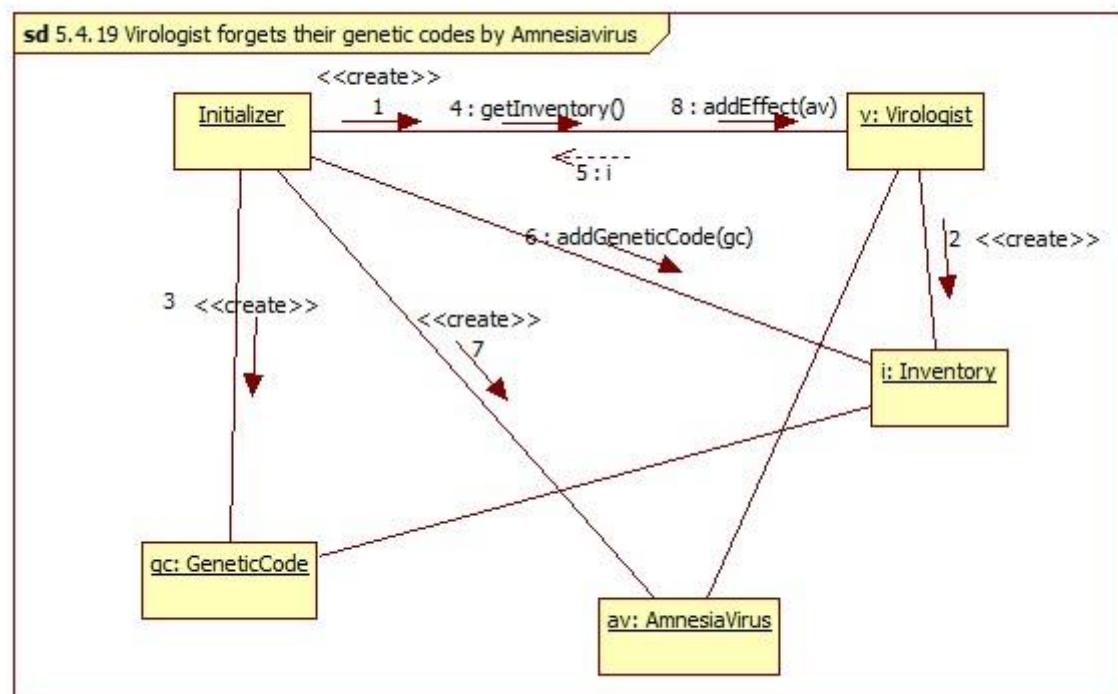
5.4.17 useVitusDanceVirus



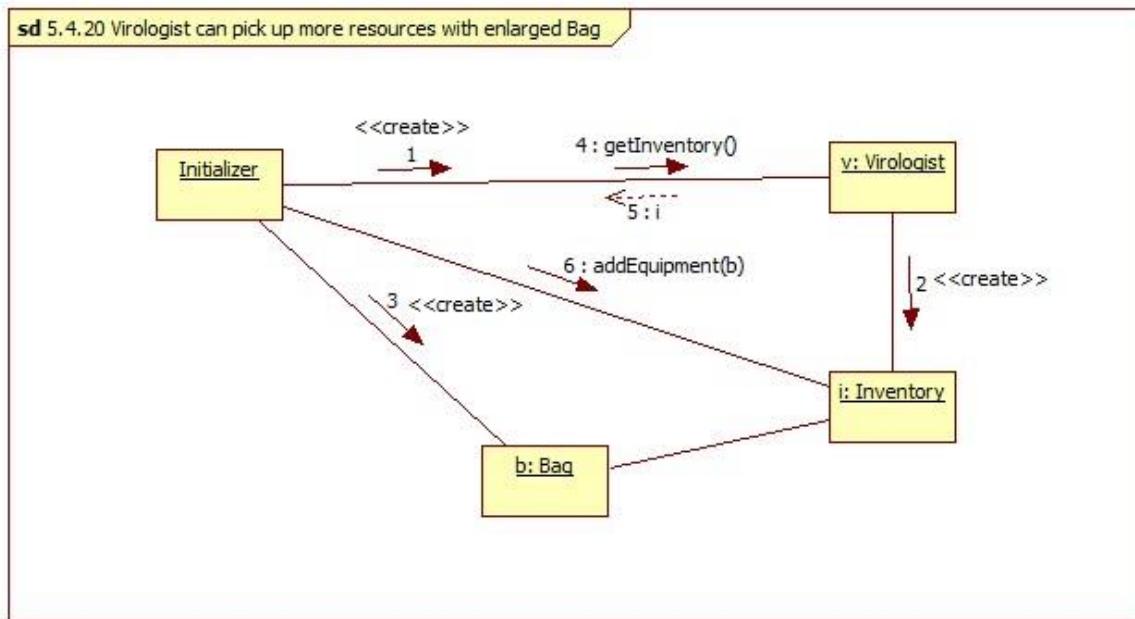
5.4.18 robVirologist



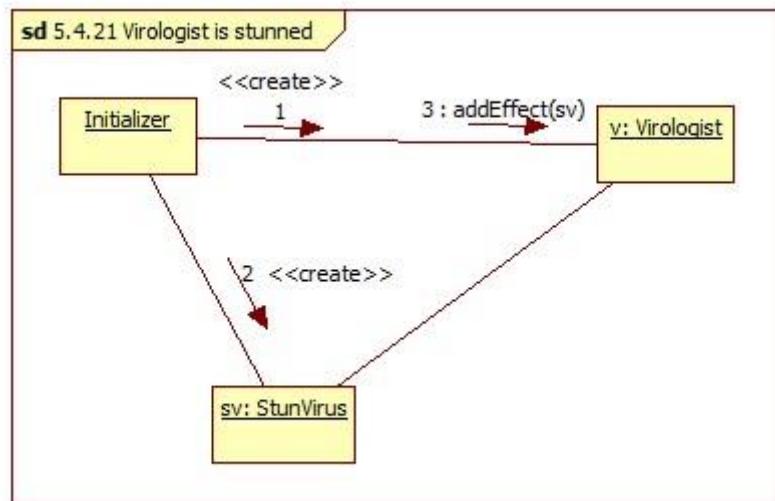
5.4.19 forgetCodesViaAmnesiaVirus



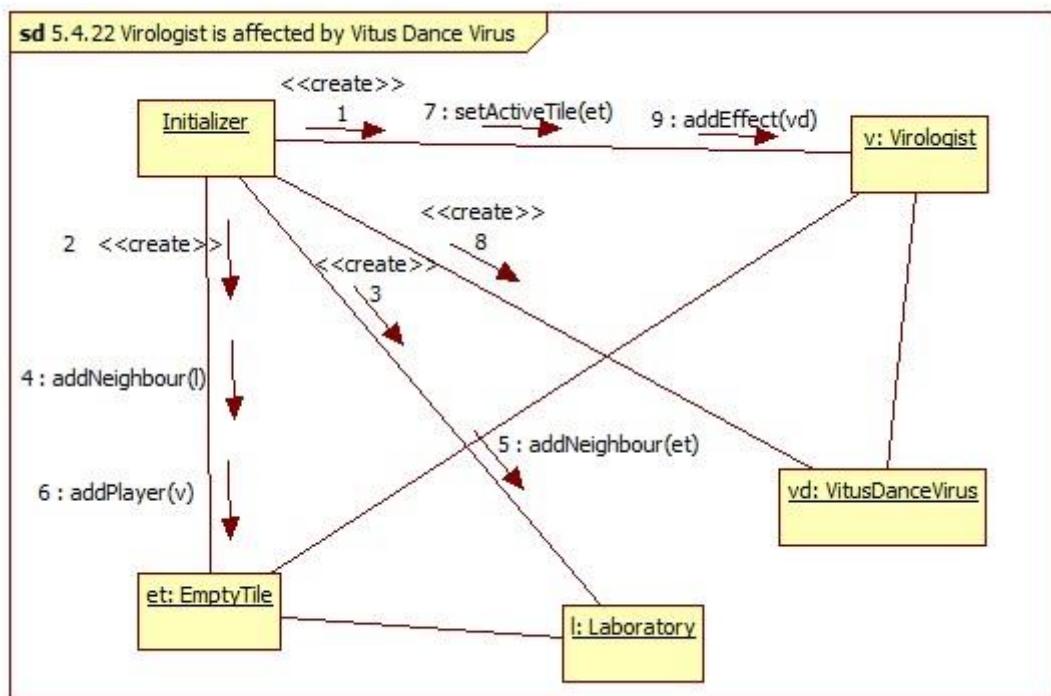
5.4.20 enlargeTheBag



5.4.21 stunnedPlayerMissesTurn



5.4.22 playerVitusDances



5.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2022.03.18. 13:00	1 óra	Fekete Kászonyi Jurásek Domonkos Száraz	Értekezlet. Döntés: Use-casek szétosztása a csapattagok között Döntés: Kommunikációs diagramok szétosztása a csapattagok között Döntés: Kommunikációs diagramok WhiteStarUML-ben
2022.03.18. 16:00	2 óra	Jurásek	4 szekvenciadiagram és kommunikációs diagram elkészítése
2022.03.19 13:00	2 óra	Száraz	4 szekvenciadiagram készítése test-case-ekhez
2022.03.19. 15:00	2 óra	Domonkos	4 use-case megírása, szekvenciadiagramok elkészítése
2022.03.19 17:00	3 óra	Kászonyi	4 use-case megírása, szekvencia diagrammok megrajzolása, kommunikációs diagrammok elkészítése.
2022.03.19. 17:00	2 óra	Domonkos	4 kommunikációs diagram elkészítése, dokumentum szerkesztése
2022.03.19 21:00	2 óra	Fekete Kászonyi Jurásek Domonkos Száraz	Értekezlet. Döntés: A kommunikációs diagramokon az inicializálásnak kell megjelennie.
2022.03.19 23:00	1 óra	Fekete	2 test-case-hez tartozó szekvenciadiagram és kommunikációs diagram elkészítése.

2022.03.20 11:00	2 óra	Fekete	4 test-case-hez tartozó szekvenciadiagram és kommunikációs diagram elkészítése, test-case leírások elkészítése.
2022.03.20 14:00	2 óra	Jurásek	5.2 véglegesítése, dokumentum szerkesztése
2022.03.20 18:00	2 óra	Domonkos	Dokumentum szerkesztése, ellenőrzése
2022.03.20 19:00	2 óra	Fekete	Dokumentum szerkesztése, átnézése
2022.03.20 19:00	2 óra	Száraz	Dokumentum szerkesztése, ellenőrzése
2022.03.20 20:00	2 óra	Kászonyi	Dokumentum átolvasása, képek beillesztése

7. Prototípus koncepciója

52 – *service_unavailable*

Konzulens:
Micskei Zoltán

Csapattagok

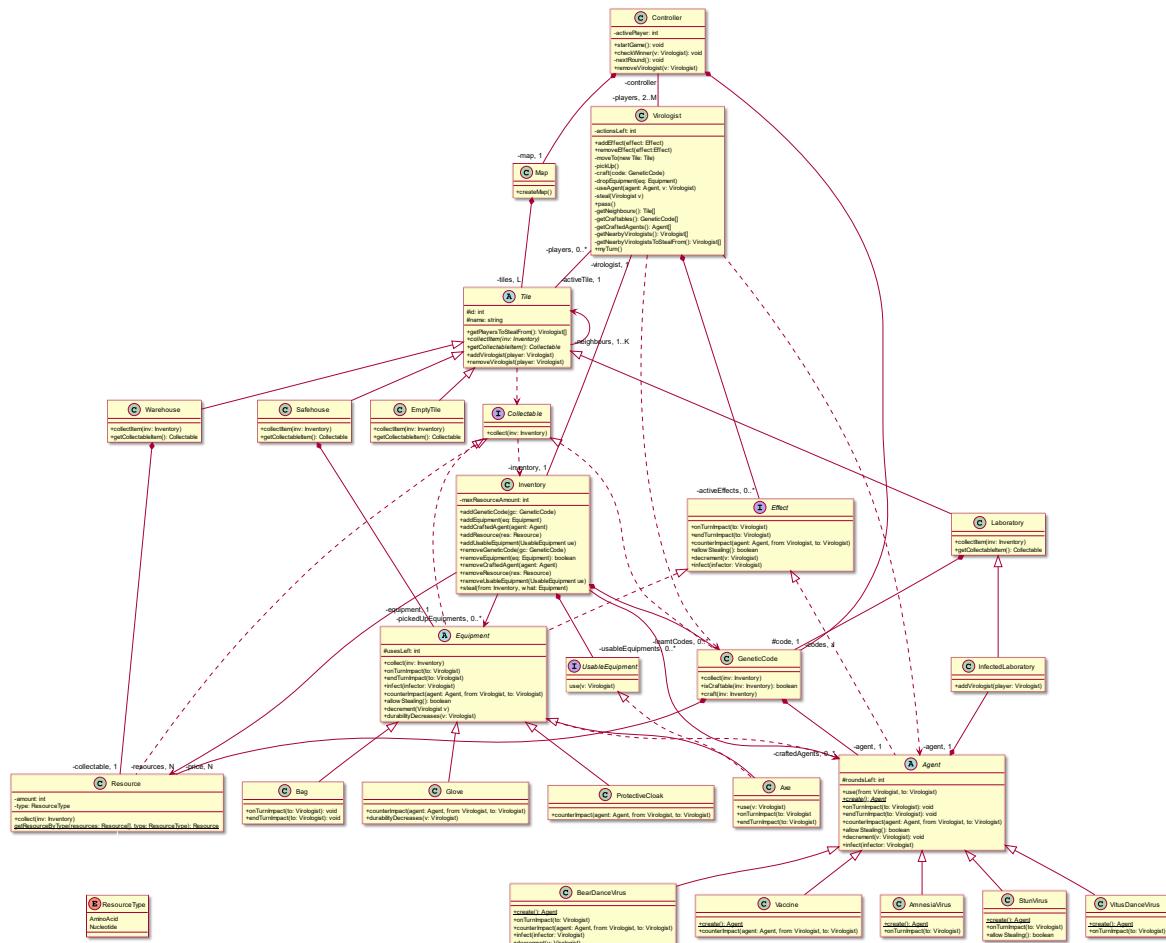
Fekete Sámuel	GJ8J3A	feketesamu@gmail.com
Száraz Dániel	GT5X34	szdani2012@gmail.com
Jurásek Jónás	PH4QFK	jurasekjonas12@gmail.com
Domonkos Ádám	CWGYWC	domonkosadam01@gmail.com
Kászonyi Zsombor Marcell	DCE2Q1	kaszonyirelli@gmail.com

2022.02.23.

6. Prototípus koncepciója

6.0 Változás hatása a modellre

6.0.1 Módosult osztálydiagram



6.0.2 Új vagy megváltozó metódusok

Új metódusok:

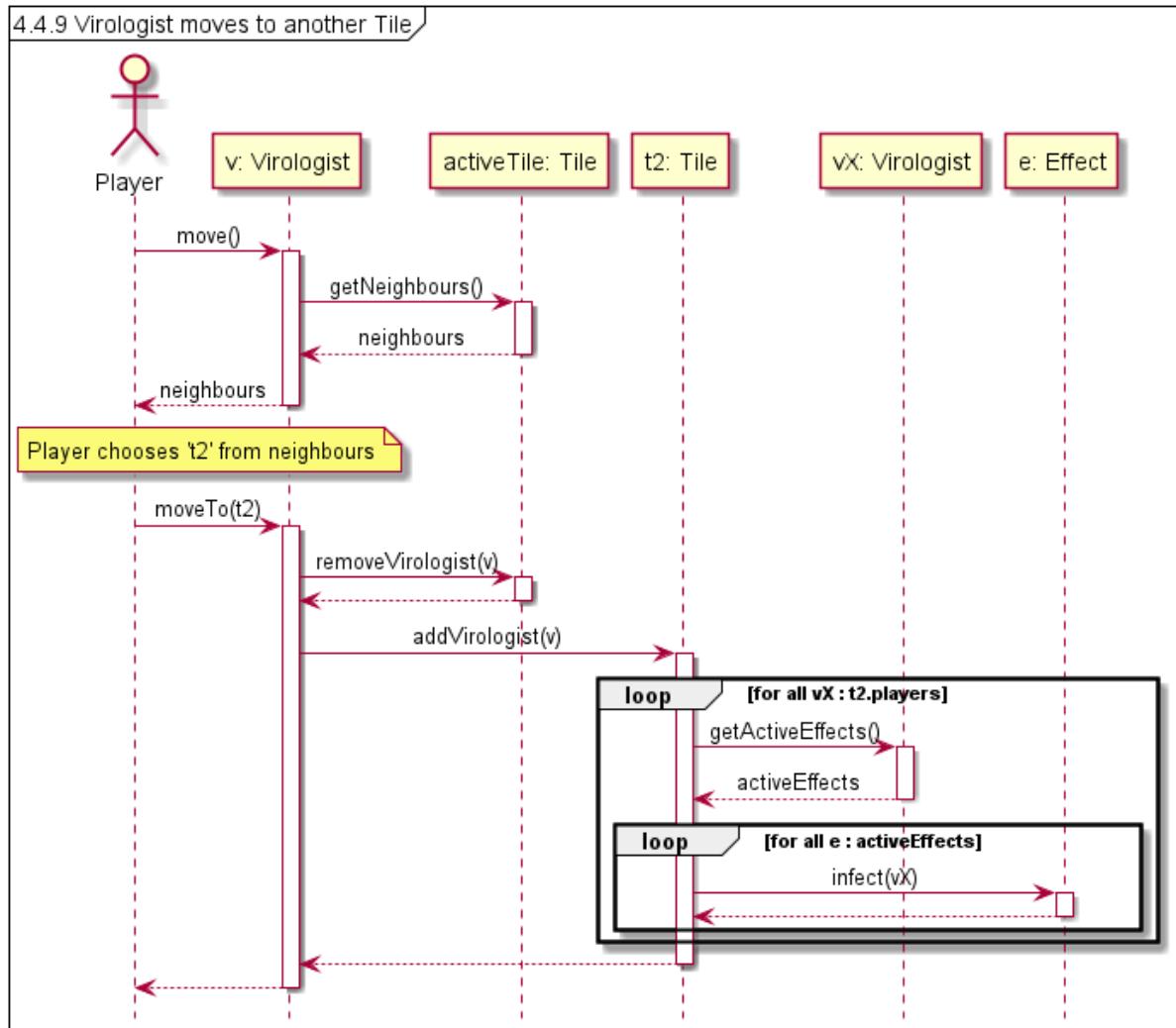
- **Controller: removeVirologist(v: Virologist):** Eltávolítja a paraméterként kapott virológust az aktív játékosok közül
- **Virologist: dropEquipment(eq: Equipment):** A játékos megsemmisít egy általa birtokolt védőfelszerelést
- **Effect: infect(infector: Virologist):** Az aktuális mezőn minden ott tartózkodó játékost megfertőz
- **Axe: +use(v: Virologist):** A paraméterként kapott játékos ellen használja a balta eszközt
- **Virologist: addUsableEquipment(UsableEquipment ue):** Hozzáadja a virológus eszközei közé a paraméterként kapott használható eszközt
- **Virologist: removeUsableEquipment(UsableEquipment ue):** Eltávolítja a virológus eszközei közé a paraméterként kapott használható eszközt
- **Equipment: durabilityDecreases(v: Virologist):** Ha az adott equipment-nél ez szükséges, akkor csökkenti a usesLeft számlálót. Az Equipment ősosztályban nem csinál semmit, ez a működés csak a Glove-ban van felüldefiniálva, ahol csökken a számláló értéke, és ha elérte a nullát, kikerül a virológus felszereléséből.

Megváltozó metódusok:

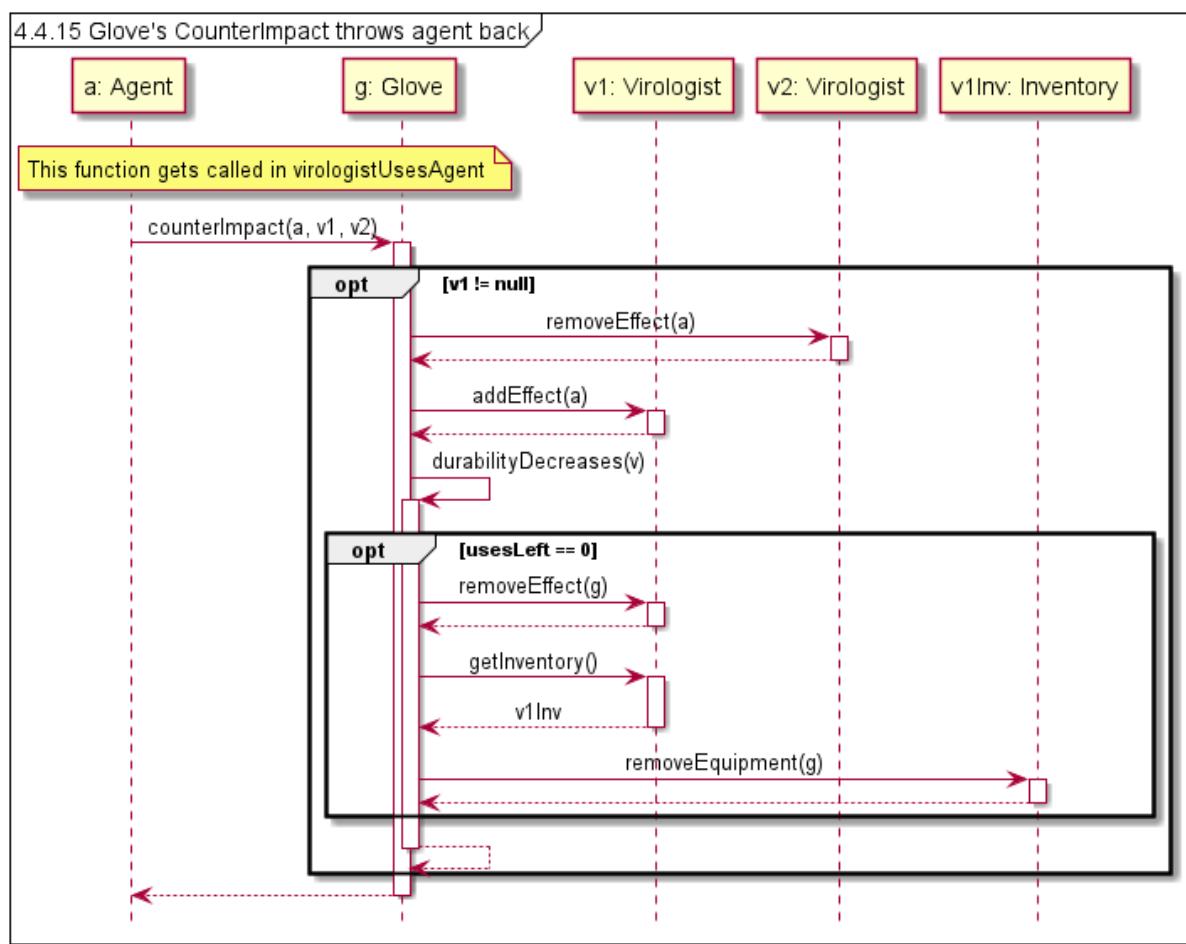
- **Tile: addPlayer(v: Virologist):** az új játékos hozzáadása után minden, a mezőn lévő játékos minden effektjén lefuttatja az infect() metódust, így ha tartózkodik a mezőn medve, akkor ő mindenkit megfertőz.
- **Glove: counterImpact(a: Agent, v1: Virologist, v2: Virologist):** Miután visszadobja az ágenst, meghívja a saját durabilityDecrease metódusát.

6.0.3 Szekvencia-diagramok

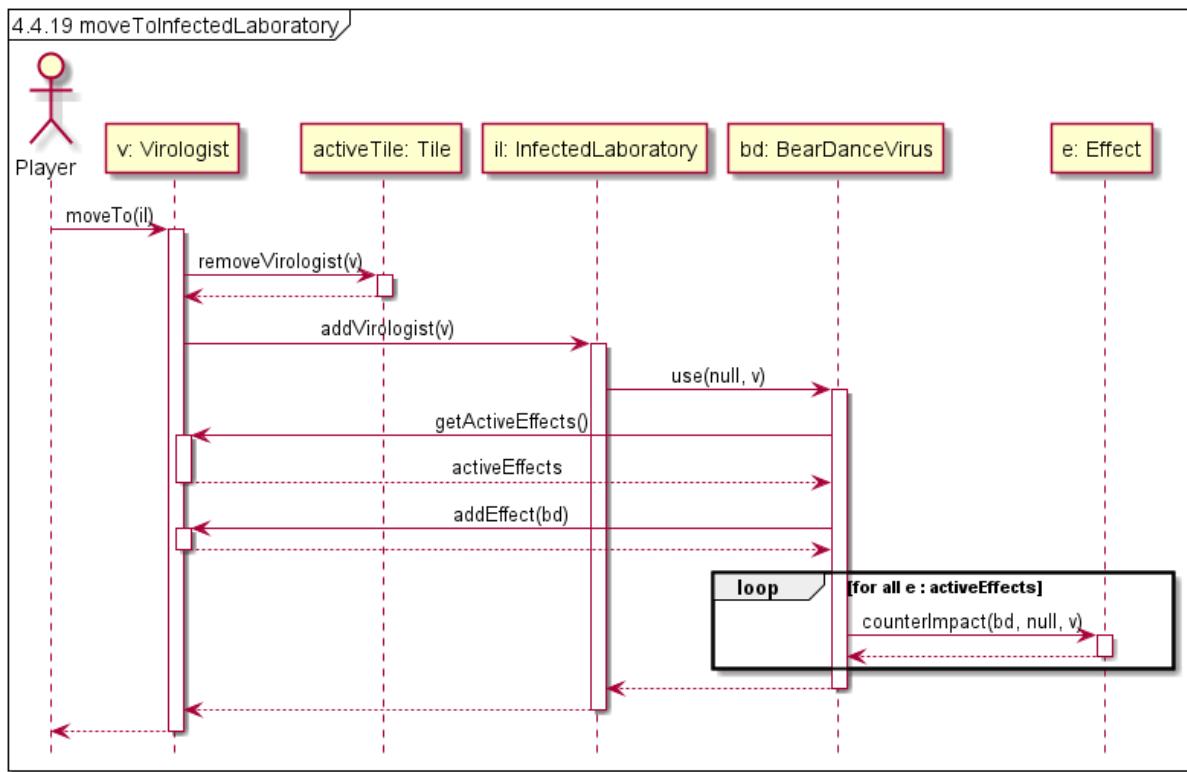
6.0.3.1 Virologist moves to another Tile



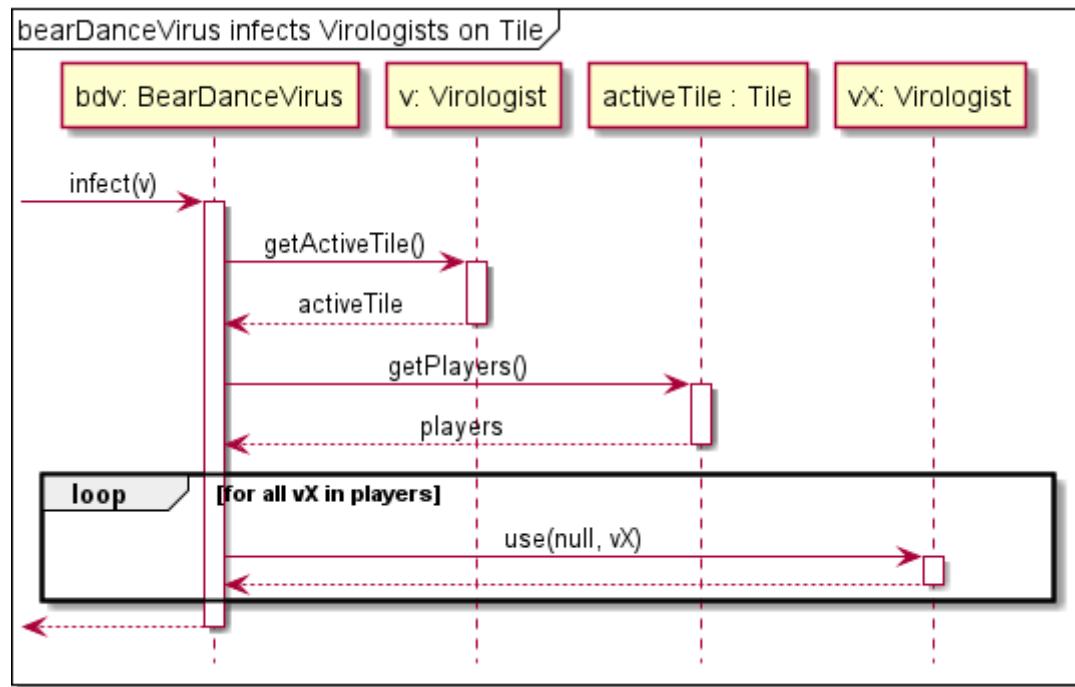
6.0.3.2 GloveCounterImpact



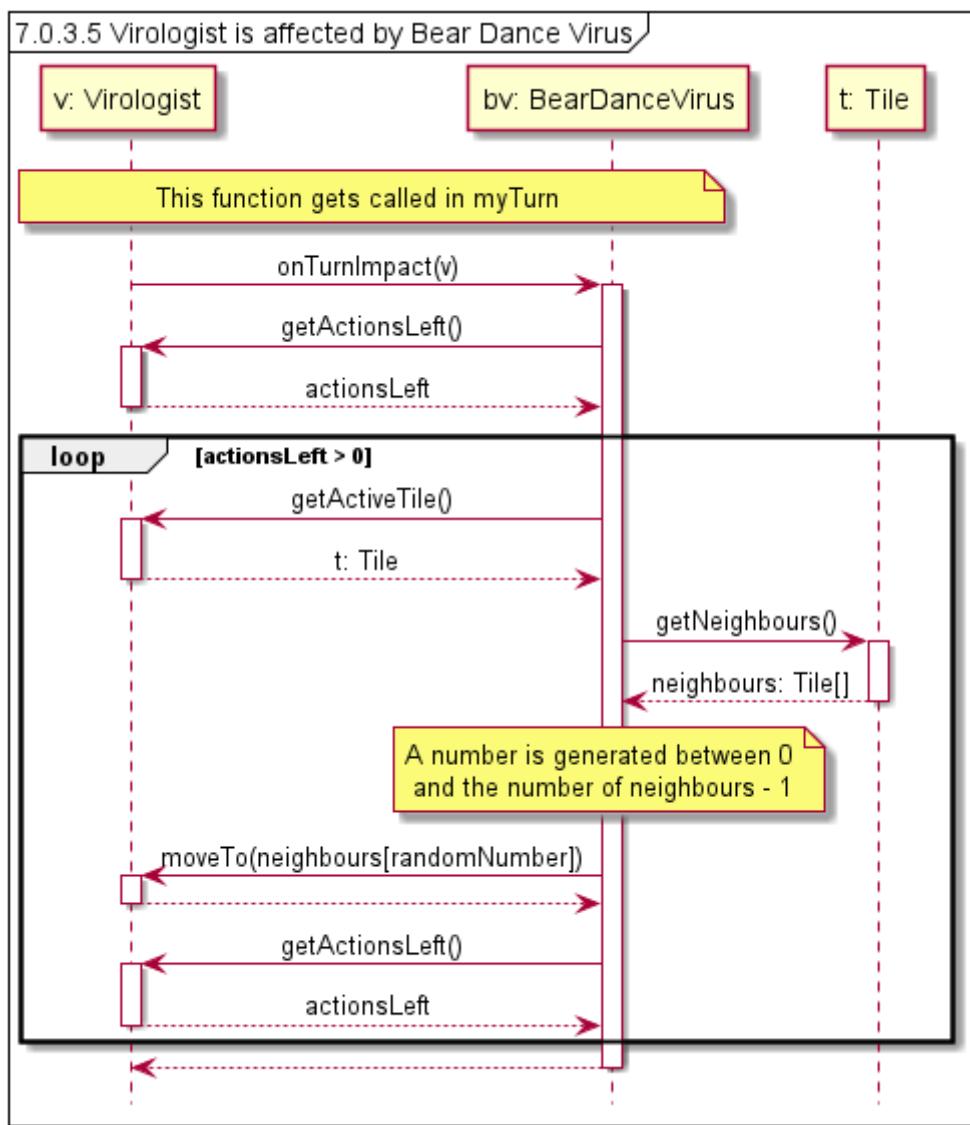
6.0.3.3 moveToInfectedLaboratory



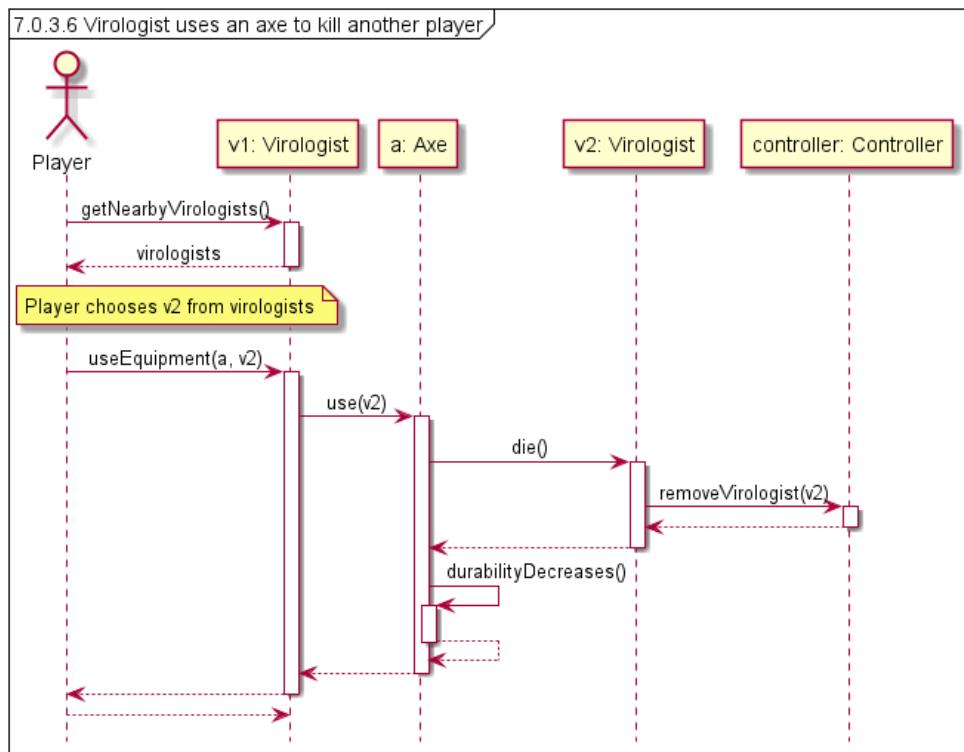
6.0.3.4 bearDanceInfect



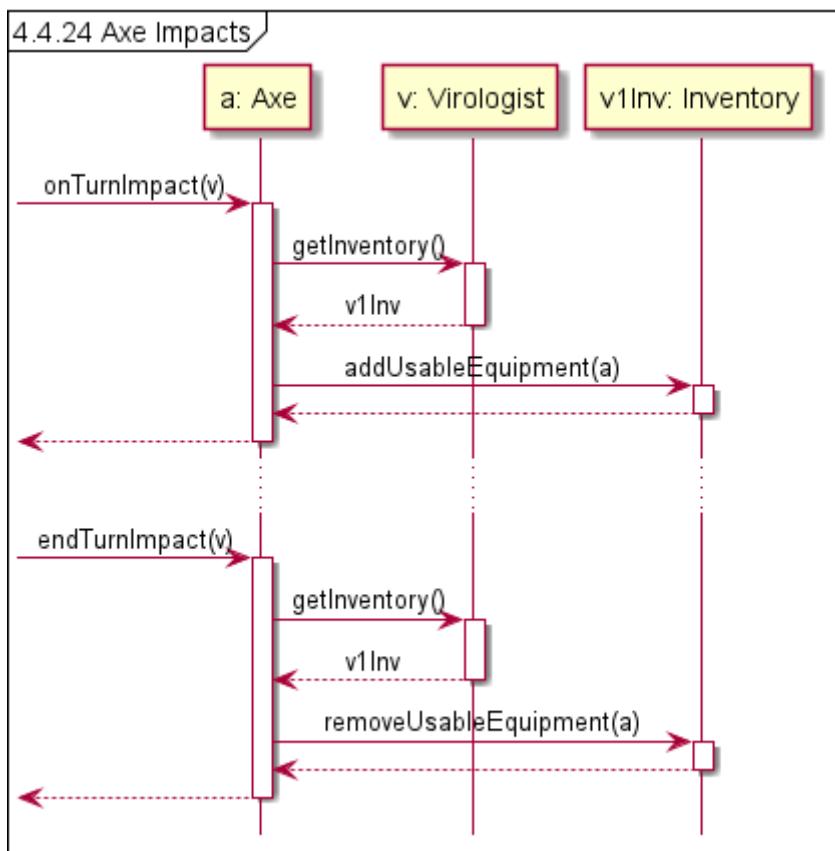
6.0.3.5 bearDanceVirusOnTurnImpact



6.0.3.6 useAxe



6.0.3.7 axeOnTurnImpact



6.1 Prototípus interface-definíciója

6.1.1 Az interfész általános leírása

Elindításkor a program egy kis köszöntő szöveget ír ki. A prototípus a parancssoron keresztül vezérelhető. Egyes parancsok az egyes akcióknak feleltethetőek meg, ezenkívül vannak a játék elindításához, illetve játékosok beállításához parancsok. A parancsok kiadásánál a kis- és nagybetű nem számít. A pálya előre elkészített, nem változtatható (Európa térképről mintázva, ezért a későbbiekben az országnevek egy mezőnélnek feleltethetőek meg). A random számok kezelésére a debug módot használjuk. Ennek indításához a program futtatásakor a parancssorban meg kell adni a ‘debug’ kifejezést argumentumnak. Ilyen módban véletlen szám sorsolásához a véletlenszerűség felülírható előre megadott elemekkel, a setnextrandom parancs segítségével.

6.1.2 Bemeneti nyelv

add

Leírás: Egy új virológust adunk a játékosokhoz, és elhelyezzük a pályán. Csak a start parancs előtt lehet kiadni. Opcionálisan megadhatjuk annak a mezőnek a nevét, ahonnan indítani szeretnénk a játékost.

Opciók: add <játékosnév> [<mezőnév>]

Pl.: add Alice

Pl.: add Bob Switzerland

start

Leírás: Elindítjuk a játékot.

Opciók: start

move

Leírás: A játékos egy szomszédos mezőre átmozog.

Opciók: move <mezőnév>

Pl.: move France

collect

Leírás: A játékos felveszi a mezőn gyűjthető dolgot.

Opciók: collect

craft

Leírás: A virológus elkészít egy ágenst.

Opciók: craft <ágensnév>

Pl.: craft AmnesiaVirus

use

Leírás: A játékos használ egy eszközt vagy ágenst egy játékoson

Opciók: use <eszköz-/ágensnév> <játékosnév>

Pl.: use Axe Bob

steal

Leírás: Egy játékos lop egy másik bénult játékestől a mezőjén.

Opciók: steal <játékosnév>

Pl.: steal Charlie

drop

Leírás: Egy játékos eldobja egy felszerelését.

Opciók: drop <felszerelésnév>

pass

Leírás: Egy játékos elpasszolja egy akcióját.

Opciók: pass

info

Leírás: A játékos, vagy mező összes információját írja ki a felhasználó számára érhető formában. Paraméter nélkül a soron lévő játékos adatait írja ki. Megadható opcionális paraméterként egy fájlnév, ekkor ebbe a fájlba íródik, egyébként a standard kimenetre.

Opciók: info [--o <objektumnév>] [--f <fájlnév>]

Pl.: info

Pl.: info --o Belgium

Pl.: info --o Charlie --f charlie_export.txt

setnextrandom

Leírás: A felhasználó beállítja a következő random sorsolás értékét. Akár előre több sorsolás értékét is megadhatja, ezek ebben a sorrendben kerülnek majd sorra. Csak debug módban használható.

Opciók: setnextrandom <érték>

Pl.: setnextrandom 2

6.1.3 Kimeneti nyelv

A parancsok közül az info ír a szabványos kimenetre, illetve az összes hibás parancs egysoros hibaüzenetet ad. Ha egy parancs jó formátumú, de nem valósítható jelenleg meg, akkor azt is üzenettel jelzi a program.

Pl.: (A dölt betű a felhasználó által kiadott parancs)

nocommand noparam

There is no such command.

add Alice

add Alice

Already existing name.

A virológusok kiírásának formátuma:

Name:

<Név>

Actions left:

<Hátralévő>/<Összes>

Tile:

<Mezőnév>

Neighbour tiles:

<Mezőnév1>

<Mezőnév2>

...

<MezőnévN>

Virologists on the same Tile:

<Név1>

...

<NévN>

Stunned Virologists:

<Név1>

...

<NévN>

Collectable:

<Gyűjthető>

Resources:

<Anyag1> <Jelenlegi mennyiség>/<Maximum>
 <Anyag2> <Jelenlegi mennyiség>/<Maximum>

Genetic codes:

<Kód1> <költség>

...

<KódN> <költség>

Usables:

<Használható1> [időtartam]

...

<HasználhatóN> [időtartam]

Equipments:

<Felszerelés1>
 <Felszerelés2>
 <Felszerelés3>

Effects:

<Hatás1> [időtartam]

...

<HatásN> [időtartam]

A Mezők kiíratásának formátuma:

Name:

<Mezőnév>

ID:

<ID>

Type:

<Mezőtípus>

Neighbours:

<Szomszéd1>
 <Szomszéd2>
 ...
 <SzomszédN>

Collectable:

<Gyűjthető>

Pl.:

info Alice

Name:

Alice

Actions left:

2/2

Tile:

Spain

Neighbour tiles:

Andorra
 France
 Portugal

Virologists on the same Tile:

Bob
 Stunned Virologists:
 none
 Collectable:
 Glove
 Resources:
 AminoAcid 50/200
 Nucleotide 100/200
 Genetic codes:
 VitusDanceVirus: 40 AminoAcid
 StunVirus: 150 Nucleotide, 20 AminoAcid
 Usables:
 AmnesiaVirus: 3 rounds left
 Axe: 1 uses left
 Equipments:
 Bag
 ProtectiveCloak
 Effects:
 Vaccine: 2 rounds left
 Bag
 ProtectiveCloak

6.2 Összes részletes use-case

Use-case neve	Add
Rövid leírás	Új virológus felvétele a játékba.
Aktorok	Controller
Forgatókönyv	Egy virológust hozzáadunk a játékhöz és elhelyezzük a pályán.

Use-case neve	Start
Rövid leírás	Elindítja a játékot
Aktorok	Controller
Forgatókönyv	Elindítja a játékot.

Use-case neve	Move
Rövid leírás	A játékos lép.
Aktorok	Player
Forgatókönyv	A játékos lép egy általa választott mezőre.

Use-case neve	Collect
Rövid leírás	A játékos felveszi az adott mezőn található gyűjthető dolgot.
Aktorok	Player
Forgatókönyv	<ul style="list-style-type: none"> A. Van felvehető anyag, ilyenkor a játékos anyagkészletéhez adódik és eggel kevesebb akciója marad. B. Van felvehető felszerelés a mezőn, ilyenkor a játékos felszereléséhez adódik és eggel kevesebb akciója marad. C. Van megtanulható genetikai kód a mezőn, ilyenkor ez a kód eltárolódik a tanult kódok között, és később ebből ágenst készíthet. A játékosnak ezután eggel kevesebb akciója marad. D. Nincsen felvehető dolog a mezőn. Ilyenkor nem történik semmi.

Use-case neve	Craft
Rövid leírás	A játékos készít egy ágenst.
Aktorok	Player
Forgatókönyv	<ul style="list-style-type: none"> 1. A játékosnak egy adott genetikai kód alapján megpróbál készíteni egy ágenst. 2./A Ha rendelkezik elég anyaggal, akkor az ágenst létrehozza és eltárolja a Player, valamint csökken az akcióinak száma 1-gyel. 2./B Ha nem rendelkezik elég anyaggal, akkor az akció sikertelenül van.

Use-case neve	Use
Rövid leírás	A játékos használ egy eszközt vagy egy ágenst.
Aktorok	Player
Forgatókönyv	<ul style="list-style-type: none"> A. A játékos rendelkezik használható eszközzel vagy ágenssel, és használja az egyiket ezek közül. Ezek után eggel kevesebb akciója marad. B. A játékos nem rendelkezik se használható eszközzel, se ágenssel, ilyenkor nem történik semmi.

Use-case neve	Steal
Rövid leírás	A játékos meglop egy bénult virológust
Aktorok	Player
Forgatókönyv	<ul style="list-style-type: none"> 1./A Az rabló játékos elveszi a bénult virológus egy felszerelését. 1./B Ha bénult virolágusnak nincs felszerelése, akkor a rabló nem tud tőle felszerelést lopni. 2. A rabló elvesz annyi anyagot a bénult virológustól, amennyit el tud tárolni.

Use-case neve	Drop
Rövid leírás	Egy játékos eldobja egy felszerelését.
Aktorok	Player
Forgatókönyv	Egy játékos eldobja egy felszerelését.

Use-case neve	Pass
Rövid leírás	A játékos passzol
Aktorok	Player
Forgatókönyv	Az aktuális játékos egyik akciója helyett passzolni tehát eggyel csökkenti akcióinak számát az adott körre.

Use-case neve	Info
Rövid leírás	Kiírja egy játékos vagy egy mező összes adatát.
Aktorok	Player
Forgatókönyv	1./A Összegyűjti a soron lévő játékos összes adatát. 1./B Összegyűjti egy megadott játékos összes adatát. 2./A Az adatokat a szabványos kimenetre írja ki. 2./B Az adatokat egy megadott fájlba írja ki.

6.3 Tesztelési terv

Teszt-eset neve	Kesztyű használata
Rövid leírás	Kesztyűvel rendelkező játékosra kent ágensek visszadobódnak a kenő virológusra. 3 kör után a késztyű megsemmisül.
Teszt célja	Az Equipment osztályból származó Glove osztály teszteli. Az eszköznek három kör után meg kell semmisülnie, így hatása sem lesz érvényes utána.

Teszt-eset neve	Lopás
Rövid leírás	Virológustól lehet lopni
Teszt célja	Virologist, StunVirus osztályt teszteli. A lebénult virológustól tudnak lopni a vele egy mezőn álló virolágusok.

Teszt-eset neve	Medvevírus elkapása
Rövid leírás	Fertőzött laboratóriumra lépve a virológus megfertőződik medvevírussal, a medvevírusos raktárakra lépve megsemmisíti a nyersanyagkészletet, a vele találkozó virolágusokat is megfertőzheti.
Teszt célja	InfectedLaboratory, BearDanceVirus osztályt teszteli. Ha valaki fertőzött mezőre lép, fertőzötté válik. Utána magától mozog és megfertőz mindenkit.

Teszt-eset neve	Balta használata
Rövid leírás	Egy játékost meg lehet ölni a baltával
Teszt célja	Axe, BearDanceVirus osztályt teszteli. A virológosukat baltával lehet megölni. Miután egy virológus meghalt, már nem vesz részt a játékban.

Teszt-eset neve	Védőköpeny védelem
Rövid leírás	Védőköpeny megvédi a viselő játékost.
Teszt célja	ProtectiveCloak osztályt teszteli. Ha egy virológus rendelkezik védőköppennnyel és az meg is védi, a rákent ágensek nem lesznek rá hatással.

Teszt-eset neve	Amnéziavírus hatása
Rövid leírás	Amnéziavírus hatására a virológus elfelejtja a megtanult genetikai kódokat
Teszt célja	AmnesiaVirus, Virologist osztályt teszteli. Ha egy virológusra rákennek egy amnéziavírust, és nincs nála ezt kivédő felszerelés, akkor elfelejtja az összes megtanult genetikai kódját.

Teszt-eset neve	Genetikai kód megtanulása és ágens elkészítése
Rövid leírás	A Virológus megtanul egy genetikai kódot, és ha van elég anyaga, akkor el is tudja készíteni belőle az ágenst.
Teszt célja	Laboratory, GeneticCode, Agent, Virologist, Resource osztályokat teszteli

Teszt-eset neve	Vakcina alkalmazása és hatása
Rövid leírás	Miután egy virológus vakcinát kent magára, más virológusok ágensei ellen megvédi, amíg le nem jár az ideje.
Teszt célja	Vaccine és Virologist osztályokban teszteli, hogy az ágens a hatását megfelelően kifejteti.

Teszt-eset neve	Vitustánc vírus alkalmazása és hatása
Rövid leírás	Miután egy virológus vitustánc vírust kent egy másik virológusra, az véletlenszerűen mozog, amíg le nem jár az ágens hatása.
Teszt célja	VitusDanceVirus és Virologist osztályokban teszteli, hogy az ágens a hatását megfelelően kifejteti.

Teszt-eset neve	Zsák hatása
Rövid leírás	Ha egy virológusnak megtelik az anyagkészlete, akkor nem tud több anyagot felvenni. Ha azonban ezután felvesz egy Zsákokat, akkor anyagkészletének a felső korlátja megduplázódik, ezután újra tud anyagot felvenni, amíg ismét be nem telik a zsákja.
Teszt célja	Virologist, Inventory, Resource, Bag osztályokat teszteli, hogy az anyagkészlet korlátai figyelembe vannak véve, de tudnak változni.

Teszt-eset neve	Az utolsó genetikai kód megtanulása és a játék megnyerése
Rövid leírás	Ha egy virolágus megtanulja az utolsó genetikai kódot, akkor a játék véget ér, ő a nyertes.
Teszt célja	Teszteli, hogy a Controller helyesen ellenőrzi-e a megtanult genetikai kódok számát.

6.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

Minden teszeset futtatásához lesz egy .bat fájl, ami futtatja a programot, majd egy .txt fájlból olvas a bemeneti nyelvben definiált parancsokat. A teszesetek minden egy vagy több info parancsal végződnek, amik egy .txt fájlba írják a tesztelt objektumok információját. Végül a .bat fájl összeveti a kimeneti fájlokat a teszeset elvárt kimenetével a Windows beépített fc (file compare) parancsával.

6.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2022.03.30.	3 óra	Domonkos Fekete Jurásek Kászonyi Száraz	Értekezlet. Döntés: Medve, az a Virológus, aki a MedveTáncVírus hatása alatt van. A Medvék immunisak minden ágensre.
2022.03.30. 21:00	2 óra	Domonkos	Teszt-esetek, megváltozott metódusok megírása
2022.03.31 14:00	2 óra	Kászonyi	Use-case-ek megírása
2022.03.31. 17:00	1 óra	Fekete	Változtatások a szekvenciákon és osztálydiagramon.
2022.04.01. 10:00	2 óra	Száraz	Use-casek kiegészítése, szekvenciák kiegészítése.
2022.04.01. 14:00	2 óra	Fekete	Tesztesetek kiegészítése, tesztelést támogató program kitalálása
2022.04.01. 17:00	4 óra	Jurásek	Prototípus interface-definíciójának kifejtése
2022.04.03 11:00	1 óra	Kászonyi	Dokumentum tartalmi ellenőrzése, javítása.
2022.04.03 19:00	2 óra	Domonkos	Dokumentum formázása, javítása

8. Részletes tervez

52 – service_unavailable

Konzulens:
Micskei Zoltán

Csapattagok

Fekete Sámuel	GJ8J3A	feketesamu@gmail.com
Száraz Dániel	GT5X34	szdani2012@gmail.com
Jurásek Jónás	PH4QFK	jurasekjonas12@gmail.com
Domonkos Ádám	CWGYWC	domonkosadam01@gmail.com
Kászonyi Zsombor Marcell	DCE2Q1	kaszonyirelli@gmail.com

2022.02.23.

7. Részletes tervezettségi részletek

Változások:

A Bemeneti nyelvhez (7.1.2) felvettünk két új parancsot.

choose

Leírás: A játék során van, hogy a felhasználó kap egy objektumlistát, és ezek közül ezzel a parancssal tud választani. minden objektum előtt lesz egy sorszám, ezt kell a parancsnak paraméterül adni.

Opciók: choose <choice number>

Pl.: choose 2

quit

Leírás: Kilép a programból, függetlenül attól, hogy van-e futó játék.

Az info parancshoz új opciót vettünk fel.

info

Leírás: A játékos, vagy mező összes információját írja ki a felhasználó számára érhető formában. Paraméter nélkül a soron lévő játékos adatait írja ki. Megadható opcionális paraméterként egy fájlnév, ekkor ebbe a fájlba íródik, egyébként a standard kimenetre. Ha nem akarunk egy adott objektumhoz tartozó összes kategóriát kiíratni, felsorolhatjuk csak azokat a kategóriákat, amelyeket szeretnénk a kimeneten látni. A sorszám a 7.1.3-ban található formátumban a megjelenés sorrendje 1-től kezdve.

Opciók: info [--o <objektumnév>] [--f <fájlnév>] [--n <kategóriszám1> <kategóriaszám2> ...]

Pl.: info

Pl.: info --o Belgium

Pl.: info --o Charlie --f charlie_export.txt 1 (csak a virológus nevét írja ki)

Pl.: info --o v1 --n 1 2 3 8 (A virológus nevét, hátralevő akcióit, mezőjét és anyagkészletét írja ki)

7.1 Osztályok és metódusok tervezési eljárásai

7.1.1 Agent

7.1.1.1 Felelősség

A különböző vírusok és vakcinák absztrakt ōsosztálya. Ez reprezentálja az elkészített, de még nem alkalmazott, illetve az alkalmazott, aktív ágenseket is. Körönként csökken a `roundsLeft` tagváltozója, amikor eléri a nullát, törlődik.

Ősosztályok

7.1.1.2 Interfészek

- Effect

7.1.1.3 Attribútumok

- **protected int roundsLeft:** hány kör műlva jár le a hatása, vagy hány körig alkalmazható még.

7.1.1.4 Metódusok

- **abstract Agent create():** Visszaad egy másolatot önmagáról. Az osztály minden leszármazottjának úgy kell megvalósítania, hogy a leszármazott típust adja vissza. Amikor a játékos egy genetikai kód szerint létrehoz egy ágenst, akkor a kód `Agent` mezőjében lévő `Agent`-en hívódik meg ez a metódus, az eredménye pedig a játékos `Inventory`-jának `craftedAgents` változójában tárolódik el.
- **use(Virologist from, Virologist to):** Egy `Virologist(from)` elkészített `Agent`jét rákeni egy `Virologist`-ra(`to`).
- **void onTurnImpact(Virologist to):** Nem csinál semmit, így a játékos körében semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
- **void counterImpact(Agent agent, Virologist from, Virologist to):** Nem csinál semmit, így a kenés elleni védekezésben semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
- **void endTurnImpact(Virologist to):** Nem csinál semmit, így a játékos körének végén semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
- **bool allowStealing():** Hamissal tér vissza, így azoknak a felszereléseknek, akik explicit nem engedélyezik, hogy lopjanak a játékostól, akin aktívak, nem kell felüldefiniálnia.
- **void decrement(Virologist v):** Csökkenti a `roundsLeft` attribútum értékét eggyel. Ha eléri a nullát, akkor megpróbálja kivenni magát a paraméterként kapott `Virologist activeEffect`-jéből és az `Inventory craftedAgents`-éből, de csak az egyikben volt benne, tehát csak az egyikből fog sikerülni.
- **void infect(Virologist v):** Nem csinál semmit. Így azok az ágensek, amik alapból nem fertőzők, nem kell implementálniuk ezt a függvényt.

7.1.2 AmnesiaVirus

- **Felelősség**

Amelyik játékosra alkalmazzák ezt a vírust, az elfelejti az összes eddig megtanult `geneticCode`-ot. Számolja a hátralévő idejét mind a virológuson, akinél effektként van jelen, mind pedig a virológus felszerelésében, ahol tárolják létrehozás után.

- **Ősosztályok**

Agent

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **Agent create()**: Létrehoz egy `AmnesiaVirus`-t és ezt visszaadja a hívónak, aki aztán ezt el tudja tárolni.
- **void onTurnImpact(Virologist to)**: A kiválasztott `Virologist` ismert genetikai kódjait kiüríti.

7.1.3 Axe

- **Felelősség**

Egy olyan védőfelszerelés, aminek használatával egy `Virologist` egy másik `Virologist`-et megölhet. Egy alkalommal lehet csak használni, ezután viszont tovább foglalja a helyet a `Virologist Inventory`-jában, ameddig az el nem dobja.

- **Ősosztályok**

- Equipment

- **Interfészek**

- UsableEquipment

- **Attribútumok**

-

- **Metódusok**

- **public void use(Virologist v)**: A paraméterben kapott `Virologist`-et kitörli a játékból, illetve a mezőről is. A `Tile`, illetve a `Controller removeVirologist()` függvényét hívja meg.
- **public Collectable cloneCollectable()**: Készít egy új `Axe` példányt, és azt adja vissza visszatérési értékében.

7.1.4 Bag

- **Felelősség**

Olyan védőfelszerelés, aminek hatására a *Virologist* kétszer annyi *Resource*-t tud eltárolni minden fajtából.

- **Ősosztályok**

- Equipment

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **public void onTurnImpact(Virologist to):** A *Virologist*-nak megkétszerezi a *maxResourceAmount* értékét a *getMaxResourceAmount ()*, illetve a *setMaxResourceAmount ()* függvények segítségével.
- **public void endTurnImpact(Virologist to):** A *Virologist*-nak megfelezi *maxResourceAmount* értékét a *getMaxResourceAmount ()*, illetve a *setMaxResourceAmount ()* függvények segítségével.
- **public Collectable cloneCollectable():** Készít egy új *Bag* példányt, és azt adja vissza visszatérési értékében.

7.1.5 BearDanceVirus

- **Felelősség**

Olyan ágens, ami *InfectedLaboratory*-ban található, és az oda belépő *Virologist*eket azonnal megtámadja. Ha egy *Virologist* elkapja, véletlenszerűen kezd lépkedni, a vele egy mezőn lévő *Virologist*-eket megpróbálja megfertőzni, és ha *Warehouse* mezőre lép, elpusztítja az ott található anyagkészletet.

- **Ősosztályok**

- Agent

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **void onTurnImpact(Virologist to):** A paraméterben kapott *Virologist* véletlenszerűen lép, majd az új mezőn meghívja az *infect()* függvényét, majd a mező *destroyCollectable()* függvényét.
- **public void infect(Virologist v):** Lekéri a paraméterben kapott *Virologist* mezőjét, majd az összes többi a mezőn tartózkodó *Virologist*-re meghívja a saját *use()* függvényét *null* és *Virologist* paraméterekkel.
- **-Collectable cloneCollectable():** Készít egy új *BearDanceVirus* példányt, és azt adja vissza visszatérési értékében.

7.1.6 ConsoleView

7.1.6.1 Felelősség

A prototípus karakteres kezelőfelülete, lekezeli a beérkező parancsokat, illetve kiírja a hibaüzeneteket, vagy az objektumok állapotát.

7.1.6.2 Ősosztályok

-

7.1.6.3 Interfészek

View

7.1.6.4 Attribútumok

-

7.1.6.5 Metódusok

- **public void menu():** A játék elkezdése előtt végrehajtható műveleteket (pl. add, start) kezeli le. “add” parancs esetén létrehoz egy új *Virologist*-et, majd a Controller *addPlayer()* metódusát hívja meg a *Virologist*tel, illetve a mező nevével paraméterként. “start” parancs hatására a Controller *gameLoop()* függvényét hívja meg.
- **public void chooseAction():** A játék közben a soron lévő játékos akciójának kiválasztását kezeli le. Az akcióknak megfeleltethető parancsok esetén (“move”, “collect”, “steal”, ...) a bejövő paraméterekhez megkeresi az aktív játékos hozzá tartozó objectjeit, és ezután meghívja a Controller az akcióhoz tartozó függvényét a megtalált paraméterekkel. Ha nem találta meg a megfelelő objektumot, vagy ha az objektumokon az akció nem hajtható végre, akkor hibaüzenetet ír ki. A “quit” parancs hatására kilép az éppen zajló játékból.
- **public int chooseOption(String[] options):** A paraméterben kapott akciókat kiírja a kimenetre, és ezután a felhasználó választhat egyet ezek közül. Visszatérési értéke a felhasználó által választott opció indexe.
- **public void gameOver(Virologist winner):** Jelzi a felhasználó számára a játék végét, és a győztes kilétét.

7.1.7 Controller

- **Felelősség**
A játék menedzselése a háttérben, egymás utáni körök kezelése, játékosok menedzselése. Ezen felül kommunikáció a `view`-val.
- **Ősosztályok**
-
- **Interfészek**
-
- **Attribútumok**
 - **-ActivePlayer:** Az éppen soron lévő játékos.
 - **-Players:** A játékosokat tároló lista.
 - **-map:** A játék pályája.
 - **-codes:** A pályán található összes genetikai kód.
 - **-endOfGame:** Vége van-e a játéknak.
 - **-view:** A felhasználói felület.
- **Metódusok**
 - **void checkWinner(Virologist v):** Ellenőrzi, hogy a kapott virológus összegyűjtötte-e az összes genetic kódot. (Összehasonlítja a codes méretét a virológus `learntCodes`-jának méretével)
 - **void gameLoop():** Amíg vége nincs a játéknak, azaz az `endOfGame` változó false, addig egy végtelen ciklusban minden körben végigmegy az összes játékoson, beállítja az `activePlayer`t, majd rajta meghívja a `startTurn()`-t, ameddig a játékosnak még van akciója, addig hívogatja a `chooseAction()`-t, majd végül az `activePlayer` `endTurn()`-jét. Ha kiér a ciklusból, akkor meghíja a `view.gameOver(activePlayer)`-t.
 - **void removeVirologist (Virologist v):** Kiszedi a kapott virológust a players-ből.
 - **Virologist getPlayerByName(String name):** Ha a players-ben szerepel name nevű játékos, akkor visszaadja, ha nem akkor Exceptiont dob.
 - **Tile getTileByName(String name):** Visszaadja a `map.getTile(name)`-et, ha null akkor Exceptiont dob.
 - **void addPlayer(Virologist v, String tileName):** Ha a kapott virológus már bent van a players-ben vagy ha a `map.getTile(tileName)` null, akkor Exceptiont dob, különben a kapott mezőhöz és a players-hez is hozzáadjuk a kapott virológust.
 - **void move(Tile t):** Ha a kapott mező szomszédos, akkor `activePlayer.move(t)`, különben Exceptiont dob.
 - **void collect():** Ha a játékos mezőjén nincs semmi felvehető, akkor Exceptiont dob, ha van, akkor `activePlayer.pickUp()` és `checkWinner(activePlayer)`.
 - **void craft(GeneticCode code):** Ha a kapott `code` craftable, akkor `activePlayer.craft(code)`, különben Exceptiont dob.

- **void use(Agent agent, Virologist v):** Ellenőrzi, hogy az `activePlayer`-nél van-e a kapott ágens és hogy vele egy mezőn áll-e a kapott virológus, ha igen, akkor `activePlayer.useAgent(agent, v)`, bármelyik nem teljesül, akkor Exceptiot dob.
- **void steal(Virologist v):** Ha a játékossal egy mezőn áll a kapott virológus, akkor az `activePlayer`-en `steal(v)`-t hív, különben Exceptiont dob.
- **void drop(Equipment eq):** Ha az `activePlayer` Inventory-jában ott van a kapott Equipment, akkor `dropEquipment(eq)`-ot hív az `activePlayer`-en, különben Exceptiont dob.
- **void pass():** `pass()`-t hív az `activePlayer`-en
- **void endGame():** `endOfGame`-et igazra állítja és az `activePlayer`-t pedig null-ra.
- **void quit():** Teljesen kilép a programból.

7.1.8 Effect

- **Felelősség**

A hatásokat kényszeríti ki, hogy minden eshetőségre készítsék el a saját reakciójukat, még akkor is, ha ők egy adott dologba nem avatkoznak bele.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **void onTurnImpact(Virologist to):** Az olyan hatások használják, amik a játékos körében az akciók végkimenetelét változtathatják.
- **void counterImpact(Agent agent, Virologist from, Virologist to):** Az olyan hatások használják, amik a mások körében rájuk kent ágensekre hatnak.
- **void endTurnImpact(Virologist to):** Az olyan hatások használják, amiknek a játékos körének végén semmisé kell tenniük a játékos körére vonatkozó történéseket.
- **bool allowStealing():** Az adott hatás engedélyezi-e, hogy lopjanak attól a játékostól, aki aktív. Ha hamissal tér vissza, az nem azt jelenti, hogy megtiltja a lopást, hanem hogy ő nem engedélyezi, de attól még egy másik aktív hatás engedélyezheti.
- **void decrement():** Csökkenti azoknak az `Effect`-eknek az élettartamát, amelyek hátralevő ideje körönként változik. Nem csökken viszont azoknak az élettartama, aminél az élettartam használat alapján változik.

7.1.9 EmptyTile

- **Felelősség**

Az üres mezőket reprezentáló osztály. Bármennyi játékos rá tud lépni, azonban nincs rajta semmilyen `Collectable` interfészt megvalósító, a `Virologist` által felvehető objektum.

- **Ősosztályok**

3. Tile

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

4. **void collectItem(Virologist player):** A mezőn nincs felvehető objektum, ezért a függvény lefut, de a játékos felszerelése nem bővül.
5. **Collectable getCollectableItem():** A mezőn nem található `Collectable` interfészt megvalósító objektum, ezért `null`-val tér vissza.

7.1.10 Equipment

- **Felelősség**

- A felszereléseket reprezentáló absztrakt osztály. `Safehouse` mezőkön lehet megszerezni. Egy `Virologist` egyszerre maximum három darabot tud belőle birtokolni.

- **Ősosztályok**

-

- **Interfészek**

- Collectable
- Effect

6. Attribútumok

7. **#int usesLeft:** A felszerelés hátralévő használatainak számát adja meg.

- **Metódusok**

- **void onTurnImpact(Virologist to):** Nem csinál semmit, így a játékos körében semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
- **void counterImpact(Agent agent, Virologist from, Virologist to):** Nem csinál semmit, így a kenés elleni védekezésben semmit nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.

- **void endTurnImpact(Virologist to):** Nem csinál semmit, így a játékos körének végén nem csináló hatásoknak nem kell definiálnia külön-külön a függvényt.
- **void infect(Virologist infector):** Nem csinál semmit, így a maguktól nem fertőző felszereléseknek nem kell implementálni ezt a függvényt.
- **bool allowStealing():** Hamissal tér vissza, így azoknak a felszereléseknek, akik explicit nem engedélyezik, hogy lopjanak az őket birtokló játékostól, nem kell felüldefiniálnia.
- **collect(Inventory inv):** Meghívja az `inv addEquipment()` metódusát saját magával, ez fogja berakni az `Inventory`-ba, ha elfér még ott.
- **void decrement(Virologist v):** Nem csinál semmit, mert az `Equipment`-nek nincs olyan élettartama, ami minden körben csökkenne.
- **void durabilityDecreases(Virologist v):** Csökkenti a `roundsLeft` attribútum értékét eggyel. Ha az eléri a 0-t, akkor meghívja a paraméterben kapott Virologus `removeEquipment` függvényét saját magát paraméterül adva.
- **abstract public Collectable cloneCollectable():** minden leszármazott készít egy új példányt saját magából, és azt adja vissza.

7.1.11 GeneticCode

- **Felelősség**
 - A játékosok ilyeneket gyűjthetnek laboratórium mezőkön és ezekből tudnak ágenseket készíteni, ha rendelkeznek a szükséges mennyiségű anyaggal. Az a játékos nyeri a játékot, aki először összegyűjt az összes `GeneticCode`-ot.
- **Ősosztályok**
 - -
- **Interfészek**
 - Collectable
- **Attribútumok**
 - **const Resource[] price:** A genetikai kódból `Agent` készítéséhez szükséges anyagmennyiség. `Resource`-ok tömbje, mert akár több különböző anyag is kellhet a készítéshez.
 - **const Agent agent:** A kódból készíthető ágens.
- **Metódusok**
 - **void collect(Inventory inv):** A paraméterként kapott `Inventory`-ban eltárolja saját maga másolatát, amit a `cloneCollectable`-el csinál.
 - **bool isCraftable(Inventory inv):** Akkor ad igazat, ha a paraméterként kapott `Inventory`-ban elég anyag áll rendelkezésre az elkészítéshez, azaz az inventoryban minden nyersanyagról legalább annyi van, mint a `price`-ban.

- **void craft(Inventory inv):** Levonja a kapott *Inventory*-ból az elhasznált anyagot (a price értékeit), majd a saját ágensének a clone-ját, amit az ágens *create()* függvényével csinál, belerakja a kapott *Inventory*-ba. Feltételezi, hogy a virológnak van elegendő anyaga, tehát csak akkor hívandó, ha *isCraftable()* igazzal tért vissza.
- **Collectable cloneCollectable():** Létrehoz egy új *GeneticCode*-ot az agent-je másolatából, amit az agent *create()* függvényével csinál. Ez után az új *GeneticCode* price-át beállítja a saját price-ára, majd return-öl az új *GeneticCode*-al.

7.1.12 Glove

- **Felelősség**
Olyan védőfelszerelés, ami megvédi az áldozatot az Ágens hatásától, és ezzel együtt a támadó játékosra pedig rákeni az ágenst.
- **Ősosztályok**
 - Equipment
- **Interfészek**
 -
- **Attribútumok**
 -
- **Metódusok**
 - **public void counterImpact(Agent agent, Virologist from, Virologist to):** A megkent *Virologistról (to)* leveszi a paraméterben kapott ágens hatását a *removeEffect()* függvényivel, a támadó *Virologistra (from)* pedig átrakja a paraméterben kapott *agent* hatását az *addEffect()* függvényivel. Ezután meghívja a saját *durabilityDecreases()* függvényét, aminek paramétere az ő birtokló *to Virologist* lesz.
 - **public Collectable cloneCollectable():** Készít egy új *Glove* példányt, és azt adja vissza visszatérési értékében.

7.1.13 InfectedLaboratory

- **Felelősség**
A játékban található olyan Laboratórium mezőket reprezentáló osztály, amelyekre lépve a játékos megfertőződik a medvevíussal, ha csak ki nem védi. A sima laboratóriumhoz hasonlóan található rajta genetikai kód, melynek megtanulása szükséges a játék megnyeréséhez.

- **Ősosztályok**
 - Laboratory
- **Interfészek**
 -

8. Attribútumok

9. **private GeneticCode code:** A Laboratórium mezőn található, felvehető genetikai kódot tárolja
10. **private Agent infectable:** Az az ágens, ami minden játékosra automatikusan felkenődik, ha erre a mezőre lép. A feladatkiírás szerint ez minden esetben *BearDanceVirus*.

- **Metódusok**

- **public void addVirologist(Virologist player):** Hozzáadja az adott mezőhöz a virológust, azaz eltárolja a *players* listában. Ezután *infectable*-ön meghívja a *use()* metódust, paraméterként *null*-t és *player*-t adván. Ezzel a játékosra rákenődik a medvetáncvírus, és azt Ő kesztyűvel nem tudja visszadobni, mert a dobó játékos helyén *null* lesz a paraméter.

7.1.14 Inventory

- **Felelősség**
Egy virológus összes anyagát, genetikai kódját, ágensét, illetve védőfelszerelését tárolja, illetve ezeket kezeli, hozzáadja az újakat, illetve törli az elvesztetteket.
- **Ősosztályok**
 -
- **Interfészek**
 -

7.1.14.1 Attribútumok

- **-Agent[] craftedAgents:** a virológus összes ágensét tartalmazó lista
- **-GeneticCode[] learntCodes:** a virológus összes megtanult genetikai kódját tartalmazó lista
- **-Equipment[] pickedUpEquipments:** a virológus összes felvett védőfelszerelését tartalmazó lista
- **-Resource[] resources:** a virológus összes anyagát tartalmazó lista
- **-int maxResourceAmount:** A maximálisan összegyűjthető anyagok mennyisége
- **-Virologist virologist:** A felszerelés tulajdonosát tárolja
- **-usableEquipments:** Az olyan felszerelések, melyeknek a viselésükön kívül van más használati módjuk is (pl.: Axe).

7.1.14.2 Metódusok

- **void addGeneticCode(GeneticCode gc):** Eltárolja a kapott Genetikai kódot a *learntCodes*-ban.

- **void addEquipment(Equipment eq):** Eltárolja a kapott felszerelést a *pickedUpEquipments*-ben és az Inventory virológusának az *addEffect()* függénnyel is átadja a kapott felszerelést, ez rakja rá a virológsra a felszerelés hatását.
- **void addCraftedAgent(Agent agent):** Eltárolja a kapott ágenst a *craftedAgents*-ben.
- **void addResource(Resource res):** A *Resource.getResourceByType()* függvényével megkeresi, hogy a saját resource-ai között van-e az adott fajtából, ha nincs, akkor létrehozza egy új resource-ot az adott típusból 0 amount-al. Utána megpróbálja jozzáadni ehhez a resourcehoz a kapott resource amount-ját az *addAmount()* függénnyel ami visszatér azza a mennyiséggel, amit hozzá lehet adni.
A visszakapott értékkel és a paraméterként kapott resource típusával létrehoz egy új *Resource*-ot és ezt return-öli.
- **void removeGeneticCode(GeneticCode gc):** Eltávolítja a kapott genetikai kódot a *learntCodes*-ból.
- **bool removeEquipment(Equipment eq):** Eltávolítja a kapott felszerelést a *pickedUpEquipments*-ból, ha az nála van, majd a *virologist*-en *removeEffect()*-et hív, szintén a felszereléssel. Ha nincs nála, hamissal tér vissza, egyébként igazzal.
- **void removeCraftedAgent(Agent agent):** Eltávolítja a kapott ágenst a *craftedAgents*-ból.
- **void removeResource(Resource res):** A *Resource.getResourceByType()* függénnyel visszakapjuk a paraméterként kapott resource típusával megegyező saját resourcunkkat és ezen hívunk egy *removeAmount()*-ot a paraméter resource amountjával.
- **void steal(Inventory from, Equipment what):** Ha a kapott *Equipment* nem null, akkor ezt a kapott *Inventory*-ból kiszedjük a *removeEquipment()*-el, azt hozzáadjuk a sajátjainkhoz az *addEquipment()*-el, majd meghívjuk az *Equipment onTurnImpact()* függvényét a saját virológusunkkal.
Ezt után lekérjük a másik *Inventory Resource*-ait a *getResources()* függénnyel, majd ezeken a *Resource*-okon végigmegyünk és mindegyiknél meghívjuk az *addResource()*-t aminek átadjuk az adott *Resource*-ot, majd ez visszaad egy mennyiséget, amennyit sikerült átvennünk, ez után ezt a mennyiséget levonjuk a másik *Inventory*-ból a *removeResource()* függénnyel.
- **void addUsableEquipment(UsableEquipment ue):** Eltárolja a kapott használható felszerelést a *usableEquipments*-ben.
- **removeUsableEquipment(UsableEquipment ue):** Eltávolítja a kapott használható felszerelést a *usableEquipments*-ból.

7.1.15 Laboratory

- **Felelősség**

A játékban található Laboratórium mezőket reprezentáló osztály. A rajta található genetikai kódok átadásáért felelős a rajta álló játékosok számára.

- **Ősosztályok**

- Tile

- **Interfészek**

-

11. Attribútumok

12. private GeneticCode code: A Laboratórium mezőn felvehető genetikai kódot tárolja

- **Metódusok**

- **public void collectItem(Inventory inv):** A mezőről felvehető `GeneticCode`-ot lemásolja a `cloneCollectable()` metódus meghívásával, majd ezen meghívja a `collect()` metódust, paraméterként továbbadva az `inv`-et. A `Collectable collect()` metódusa fogja ténylegesen eltárolni az `Inventory`-ban önmagát.
- **public Collectable getCollectableItem():** Visszaadja a mezőn található `GeneticCode`-ot.

7.1.16 Map

- **Felelősség**

A játékot alkotó pálya tárolásáért felelős osztály. Tárolja a mezőket, amik a közvetlen szomszédok ismerete miatt lefedi az egész pályát. Ő felelős a pálya és mezők inicializálásáért, azaz ő dönti el, milyen típusú mezők vannak és azokon milyen típusú tárgyat lehet felvenni.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **private ArrayList<Tile> tiles:** A pálya mezői. A `createMap()`-ben töltődik fel `Tile`-okkal.

- **Metódusok**

- **public GeneticCode[] createMap()**: Inicializálja a pályát egy beleégetett logika szerint. Létrehozza a mezőket (Európa országai), a konstruktorokban megadja a mező típusától függően, hogy milyen felvehető objektum legyen ott. Például egy Óvóhelyen minden felszerelés legyen felvehető, vagy egy genetikai kód megtanulásával minden Ágenst lehet elkészíteni. Visszatérési értékként visszaadja a kontrollernek az elkészített genetikai kódokat, ezeket kell majd a virológusoknak összegyűjteni.

7.1.17 ProtectiveCloak

- **Felelősség**

Olyan védfelszerelés, amely az áldozat virológusokat 82.3% eséllyel védi meg a rájuk kent ágensek hatásától. Ilyenkor az ágens elhasználódik, és hatását senki nem kapja meg.

- **Ősosztályok**

- Equipment

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **public void counterImpact(Agent agent, Virologist from, Virologist to)**: Sorsol egy számot 1-től 1000-ig. Ha a szám nem nagyobb, mint 823, megkent *Virologist*-ról (*to*) leveszi a paraméterben kapott ágens hatását a *removeEffect()* függvény segítségével.
- **public Collectable cloneCollectable()**: Készít egy új *ProtectiveCloak* példányt, és azt adja vissza visszatérési értékében.

7.1.18 Resource

7.1.18.1 Felelősség

Egy adott anyagból egy bizonyos mennyiséget reprezentál.

7.1.18.2 Ősosztályok

-

7.1.18.3 Interfészek

- Collectable

7.1.18.4 Attribútumok

- **private int amount**: Az anyag mennyisége
- **private final ResourceType type**: Az anyag típusa.

7.1.18.5 Metódusok

- **public void collect(Inventory inv):** A paraméterként kapott *Inventory*-hoz hozzáadja a saját klónját. A saját klónját a *cloneCollectable* függvényel hozza létre. A *Inventory*-hoz az *addResource* függvényel tudja hozzáadni.
- **public Collectable cloneCollectable():** Visszaadja a *Resource* a saját klónját. Ehhez létrehoz egy új példányt a korábbi attribútumokkal és azt adja vissza,
- **public int getAmount():** Visszaadja az *amount* attribútumot.
- **public void setAmount():** Beállítja az *amount* attribútum értékét a paraméterben kapott értékre.
- **public int getType():** Visszaadja az *type* attribútumot.
- **public void setType():** Beállítja az *type* attribútum értékét a paraméterben kapott értékre.
- **public int addAmount(int maxAmount, int addedAmount):** Hozzáadja az *amount* attribútumhoz a paraméterként kapott értéket, de nem haladja meg a paraméterként kapott max értéket. Visszaadja a sikeresen hozzáadott értéket.

```
Initialize overload := 0
Set amount := amount + maxAmount
If amount > maxAmount Then
    Set overload := amount - maxAmount
    Set amount := maxAmount
End If
Return addedAmount - overload
```
- **public void removeAmount():** Levonja a paraméterként kapott értéket az *amount* attribútumból. Ha a levonás után negatív lenne a az érték, egyenlővé teszi nullával.
- **public static Resource getResourceByType(ArrayList<Resource> resources, ResourceType type):** A tömbből visszaadja a *type* típusú. Végigjárja a paraméterként kapott listát és az összes elemének a típusát összehasonlítja a paraméterként kapott típussal. Amint megtalálta, visszatér vele. Ha a cikluson végitérülés után még nem tért vissza, *null*-lal tér vissza, mert nincs olyan típus a listában.
- **public static void initializeResourceArray(ArrayList<Resource> resources):** Létrehozza a paramétreként kapott tömbbe az összes típusú *Resource*-t. Ehhez végigiterál a lehetséges *ResourceType*-ken, és minden egyik típusból hozzáad egy nulla mennyiségű *Resource*-t a listához.

7.1.19 Safehouse

- **Felelősség**
A játékban szereplő óvóhely pályaelem reprezentációjáért felelős osztály. A mezőn található Equipment felvételéért felelős.
- **Ősosztályok**
 - Tile
- **Interfészek**
 -
- **Attribútumok**
 - **private Equipment equipment:** A Safehouse-ban lévő védőfelszerelés.
- **Metódusok**
 - **public void collectItem(Inventory inv):** A mezőről felvehető Equipment-et lemásolja a cloneCollectable() metódus meghívásával, majd ezen meghívja a collect() metódust, paraméterként továbbadva az inv-et. A Collectable collect() metódusa fogja ténylegesen eltárolni az Inventory-ban önmagát.
 - **public Collectable getCollectableItem():** Visszaadja a mezőn található Equipment-et.

7.1.20 StunVirus

7.1.20.1 Felelősség

Amelyik virológusra alkalmazzák azt a következő x körre megállítja akciók végrehajtásában. Számolja a hátralévő idejét mind a virológuson, akinél effektként van jelen, mind pedig a virológus felszerelésében, ahol tárolják létrehozás után. A bénítóvírussal fertőzött virológustól lehet lopni.

7.1.20.2 Ősosztályok

Agent

7.1.20.3 Interfészek

-

7.1.20.4 Attribútumok

-

7.1.20.5 Metódusok

- **Agent create():** Létrehoz egy StunVirus-t és ezt visszaadja a hívónak, aki aztán ezt el tudja tárolni.
- **void onTurnImpact(Virologist to):** A kiválasztott Virologist akcióját elveszi, így lebénítja adott számú körre, így nem fog tudni akciókat végrehajtani.
- **boolean allowStealing():** Igazzal tér vissza, mert akin aktív ez az effekt, attól lehet lopni.

7.1.21 Tile

- **Felelősség**

A pályát alkotó mezőket reprezentáló osztály. Privát tagváltozóban tárolja az egyedi azonosítóját, illetve a megjelenítési nevét. Ismeri a rajta tartózkodó *Virologist* objektumokat, illetve az adott mező közvetlen szomszédait. A mezőn található játékosok és *Collectable* interfész megvalósító objektumok listázásáért és átadásáért felel a rajta lévő játékosoknak.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **protected Virologist[] players:** Mezőn álló játékosok (*Virologist*) referenciáját tároló lista.
- **protected Tile[] neighbours:** Szomszédos mezők (*Tile*) referenciáját tárolja
- **protected int id:** Mező egyedi azonosítója
- **protected string name:** Mező megjelenítési neve

- **Metódusok**

- **public abstract collectItem(Inventory inv):** Absztrakt metódus. A leszármazott osztályokban a mezőről felvehető *Collectable*-t átadja a paraméterként kapott játékos (*Virologist*) számára. Ha mezőn nincs felvehető objektum, a függvény lefut, de a játékos felszerelése nem bővül.
- **public abstract Collectable getCollectableItem():** Absztrakt metódus. A leszármazott osztályokban visszaadja a mezőn található *Collectable*-t. Ha a mezőn nem található ilyen objektum, akkor *null*-lal tér vissza.
- **public Virologist[] getPlayersToStealFrom():** A *players* mező virológusai közül azokat adja vissza, akiktől lehet lopni. Végigmegy a *players* listán, és minden *Virologist*-tól elkeríti az *activeEffects* listáját, majd végigmegy minden kapott *Effect*-en, és meghívja rajtuk az *allowStealing()* metódust. Ha van olyan effekt, amin ez igazzal tér vissza, akkor ez a *Virologist* belekerül az eredménybe, egyébként nem.
- **public void addVirologist(Virologist player):** Hozzáadja az adott mezőhöz a virológust, azaz eltárolja a *players* listában. Ezután a *players* lista minden elemétől elkeríti az *activeEffects* listáját, majd végigmegy minden *Effect*-en, és meghívja rajtuk az *infect()* metódust. Ha az új játékos érkezése után van medvevírusos játékos a mezőn, akkor ő mindenkit megfertőz.
- **public void removeVirologist(Virologist player):** Leveszi az adott mezőről a virológust, azaz eltávolítja a *players* listából.
- **public void destroyCollectable():** A mezőn található *Collectable* megsemmisítésére való függvény, de itt az ősosztályban nem csinál semmit, így csak azoknak a leszármazottaknak kell felüldefiniálni, ahol megtörténhet ez.

7.1.22 UsableEquipment

7.1.22.1 Felelősség

Olyan védőfelszereléseket reprezentáló interfész, amiket lehet használni másik játékoson, nem csak a birtokló játékosra van hatásuk.

7.1.22.2 Ősosztályok

-

7.1.22.3 Interfészek

-

7.1.22.4 Attribútumok

-

7.1.22.5 Metódusok

- **void use(Virologist v):** A paraméterben megadott *v* virológusra használja a hívó a felszerelést.

7.1.23 Vaccine

7.1.23.1 Felelősség

Olyan ágens, amely az áldozat virolágusokat megvédi a rájuk kent ágensek hatásától. Ilyenkor az ágens elhasználódik, és hatását senki nem kapja meg. Számolja a hátralévő idejét mind a viroláguson, akinél effektként van jelen, mind pedig a virolágus felszerelésében, ahol tárolják létrehozás után.

7.1.23.2 Ősosztályok

Agent

7.1.23.3 Interfészek

-

7.1.23.4 Attribútumok

-

7.1.23.5 Metódusok

- **void counterImpact(Agent agent, Virologist from, Virologist to):** A megkent *Virologist*-ról leveszi a kent ágens hatását a *removeEffect()* függvény meghívásával.
- **static Agent create():** Létrehoz egy *Vaccine*-t és ezt visszaadja a hívónak, aki aztán ezt el tudja tárolni.

7.1.24 View

7.1.24.1 Felelősség

A felhasználóval való kommunikációért felelős interfész, a felhasználó cselekvéseit a controllernek továbbítja.

7.1.24.2 Ősosztályok

7.1.24.3 Interfészek

7.1.24.4 Attribútumok

7.1.24.5 Metódusok

- **public void menu():** A játék elkezdése előtt végrehajtható műveleteket (pl. add, start) kell lekezelnie.
- **public void chooseAction():** A játék közben a soron lévő játékos akciójának kiválasztását kell lekezelnie és a kapott akciót elindítania.
- **public int chooseOption(String[] options):** A paraméterben kapott akciók közül kell a felhasználónak egyet választania. A választás index-ét adja vissza.
- **public void gameOver(Virologist winner):** Jelzi a felhasználó számára a játék végét, és a győztes kilétét.

7.1.25 Virologist

- **Felelősség**

Ez az osztály reprezentálja a játékosokat a játékban. Feladatai közé tartoznak a különféle akciók, amiket a játékos a körében választhat. Ez az osztály kommunikál a Controllerrel, hogy mikor melyik akciót szeretné végrehajtani.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **int actionsLeft:** Nyilvántartja, hogy hány akciója maradt még a virológusnak az adott körben.
- **private static Controller controller:** A játékot vezérlő kontroller.
- **private Tile activeTile:** A Mező, amin a virológus éppen áll.
- **private final Inventory inventory:** A virológus felszerelése. A játék során nem változik, csak a tartalma.
- **private static ArrayList<Effect> activeEffects:** A virolágusra ható effekteket tároló lista.
- **private String name:** A játékos egyedi neve.

- **Metódusok**
 - **public void addEffect(Effect effect)**: Hozzáadja a paraméterként kapott `effect`-et a virológus `activeEffects` listájához.
 - **public void removeEffect(Effect effect)**: Eltávolítja a paraméterként kapott `effect`-et a virológus `activeEffects` listájából.
 - **public void pass()**: Olyan akció, ami csökkenti az `actionsLeft` változót, de nincs más hatása.
 - **public void startTurn()**: A játékos sorra kerülése előtt meghívja az összes `activeEffects`-ben található hatás `onTurnImpact` függvényét.
 - **public void endTurn()**: A játékos köre után lekéri az `Inventory`-ban található `craftedAgents`-ket. Ezeken meghívja a `decrement` függvényt. Meghívja az összes `activeEffects`-ben található hatás `endTurnImpact` függvényét. Visszaállítja az `actionsLeft` változót az eredeti értékre.
 - **public void moveTo(Tile newTile)**: Eltávolítja magát az `activeTile` változóban tárolt mezőről a `removeVirologist` függvénnnyel. Hozzáadja magát a paraméterben kapott mezőhöz az `addVirologist` függvénnnyel. Eltárolja a paraméterként kapott mezőt az `activeTile` attribútumban. Csökkenti a hátralévő akciókat számláló `actionsLeft` attribútumot.
 - **public void pickUp()**: Meghívja az `activeTile`-ban tárolt mezőn a `collectItem` függvényt. Ezzel bekerül a felszerelései közé a mezőn található `Collectable` objektum.
 - **public void craft(GeneticCode code)**: Létrehozza az `Agent`-et a paraméterként kapott `GeneticCode`-ból a `craft` függvénnnyel.
 - **public void useAgent(Agent agent, Virologist v)**: Felkeni a paraméterként kapott `Agent`-et a paraméterként kapott Virológusra a `use` függvénnnyel.
 - **public void dropEquipment(Equipment eq)**: Megsemmisít egy, a játéknál lévő felszerelést. Az `inventory` tagváltozón meghívja a `removeEquipment` függvényt a paraméterként kapott `Equipment`-tel. Az eldobás után ellenőrzi az anyagok mennyiségét, ha több mint az aktuális maximum, levonjuk a túlcordulás értékét az anyagból a `removeAmount` függvénnnyel.
 - **public void steal(Virologist v)**: Lop a paraméterként kapott virológustól. Lekéri a paraméterként kapott virológus felszereléseit a `getInventory` függvénnnyel. Meghívja a saját `Inventory`-án a `steal` függvényt, és átadja a kirabolt virolögustól lekért `Inventory`-t, és a lopni kívánt `Equipment`-et.
 - **public ArrayList<Tile> getNeighbours()**: Visszatér a virológus szomszédaival az `activeTile getNeighbours ()` függvényével.
 - **public ArrayList<GeneticCode> getCraftables()**: Visszatér a létrehozható `GeneticCode`-kal. Lekéri a saját `Inventory`-jának a megtanult `GeneticCode`-jait a `getLearnedCodes` függvénnnyel. Ellenőrzi az összes visszakapott `GeneticCode`-on, hogy létre tudja-e hozni az `isCraftable` függvénnnyel. A létrehozható genetikai kódokat egy listába gyűjt. Az összes kód ellenőrzése után visszatér a listával.
 - **public ArrayList<Agent> getCraftedAgents()**: Visszatér a saját `Inventory`-jában található, létrehozott `Agent`-ekkel, a `getCraftedAgents` függvénnnyel.

- **public ArrayList<Virologist> getNearbyVirologists():** Visszatér a saját mezőjén található virológusokkal. Ehhez az `activeTile` mező `getPlayers` függvényét hívja meg. A visszakapott virológusok közül kiveszi saját magát.
- **public ArrayList<Virologist> getNearbyVirologistsToStealFrom():** Visszatér a saját mezőjén található virológusokkal, akiktől tud lopni. Ehhez az `activeTile` mező `getPlayersToStealFrom` függvényét hívja meg. A visszakapott virológusok közül kiveszi saját magát.
- **public int getActionsLeft():** Visszaadja a hátralévő akciók számát tároló `actionsLeft` attribútumot.
- **public Tile getActiveTile():** Visszaadja a virológus aktuális mezőjét tároló `activeTile` attribútumot.
- **public void setActiveTile(Tile activeTile):** Beállítja a virológus aktuális mezőjét tároló `activeTile` attribútumot a paraméterként kapott mezőre.
- **public Inventory getInventory():** Visszaadja a virológus felszerelését tároló `inventory` attribútumot.
- **public ArrayList<Effect> getActiveEffects():** Visszaadja a virológusra ható effektek listáját tároló `activeEffects` listát.
- **public String getName():** Visszaadja a virológus nevét tároló `name` attribútumot.
- **public void setName(String name):** Beállítja a virológus nevét tároló `name` attribútumot a paraméterként kapott névre.
- **public static void setController(Controller controller):** Beállítja a játékvezérlőt tároló `controller` attribútumot.

7.1.26 VitusDanceVirus

7.1.26.1 Felelősség

Amelyik virológusra alkalmazzák, az a következő x körre a körét azzal kezdi, hogy random lép kettőt. Számolja a hátralévő idejét mind a virológuson, akinél effektként van jelen, mind pedig a virológus felszerelésében, ahol tárolják létrehozás után.

7.1.26.2 Ősosztályok

Agent

7.1.26.3 Interfészlek

-

7.1.26.4 Attribútumok

-

7.1.26.5 Metódusok

- **Agent create():** Létrehoz egy `VitusDanceVirus`-t és ezt visszaadja a hívónak, aki aztán ezt el tudja tárolni.
- **void onTurnImpact(Virologist):** A kiválasztott `Virologist` tett lebénítja adott számú körre, így nem fog tudni akciókat végrehajtani.

7.1.27 Warehouse

- **Felelősség**

A játékban szereplő raktár pályaelem reprezentációjáért felelős. Ő felel a mezőn található Resource objektum átadásáért a pályán tartózkodó játékosnak.

- **Ősosztályok**

- Tile

- **Interfészek**

-

- **Attribútumok**

- **private Resource collectable:** A Warehouse mezőn található és felvehető anyagot tárolja.

- **Metódusok**

- **public void collectItem(Virologist player):** A mezőről felvehető Resource-ot lemásolja a `cloneCollectable()` metódus meghívásával, majd ezen meghívja a `collect()` metódust, paraméterként továbbadva az `inv`-et. A `Collectable collect()` metódusa fogja ténylegesen eltárolni az `Inventory`-ban önmagát.
 - **Collectable getCollectableItem():** Visszaadja a mezőn található Resource objektumot, vagy `null`-t, ha azt már megsemmisítették.
 - **public void destroyCollectable():** A mezőn található Collectable megsemmisítésére való függvény, azaz a `collectable` mezőt `null`-ra állítja.

7.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

A pálya térképe a következő képen látható.



Az egyes mezők nevei megegyeznek a térképen látható országnevekkel. Az egyes országok mezői a térképen látható szomszédos országok mezőivel szomszédosak. Ezenkívül France és United Kingdom, valamint Denmark és Sweden szomszédos, ami a térképen nem látszik.

Az egyes mezők típusai és a rajtuk található felvehető dolgok:

Albania: Raktár, Nukleotid

Greece: Raktár, Aminosav

Bosnia And Herzegovina: Óvóhely, Kesztyű

Croatia: Óvóhely, Védőköpeny

Slovenia: Óvóhely, Zsák

Austria: Óvóhely, Balta

Bulgaria: Laboratórium, Bénítóvírus genetikai kódja

Montenegro: Laboratórium, Vitustáncvírus genetikai kódja

Macedonia: Vakcina genetikai kódja

Serbia: Amnéziavírus genetikai kódja

Romania: Medvevírussal fertőzött laboratórium, Vakcina genetikai kódja

A többi mező üres mező.

A bemenetekhez írtunk kommenteket // jel után, ezek nem lesznek részei a tesztfájlnak, csak a megértést segítik.

7.2.1 Kesztyű használata

- **Leírás**

Kesztyűvel rendelkező játékosra kent ágensek visszadobódnak a kenő virológusra.
3 használat után a késztyű megsemmisül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- A bénítóvírus elkészítése a genetika kódjával és nukleotiddal
- Késztyű viselés: akin a késztyű meg próbálnak megkenni egy bénítóvírusossal, arra nem kerül rá a bénítóvírus hatása
- Késztyű viselés: aki megdobja a késztyűs játékost, arra visszadobódik a bénítóvírus
- Ha elhasznál egy bénítóvírust, az kikerül a felszerelései közül
- 3 használat után a késztyű eltűnik

- **Bemenet**

add v1 BosniaAndHerzegovina

add v2 Bulgaria

add v3 Bulgaria

add v4 Bulgaria

start

//v1

pass

pass

//v2

collect

//bénítóvírus genetikai kódjának megtanulása

move Macedonia

//v3

collect

//bénítóvírus genetikai kódjának megtanulása

move Macedonia

//v4

collect

//bénítóvírus genetikai kódjának megtanulása

move Macedonia

//v1

collect

//késztyű felvétele

move Montenegro

//v2

Move Albania

collect

//nukleotid felvétele

//v3

Move Albania

collect

//nukleotid felvétele

//v4

Move Albania

collect

//nukleotid felvétele

//v1

*Move Albania
pass*

```
//v2
craft StunVirus //bénítővírus elkészítése
info --o v2 --n 1 9 10

use StunVirus v1 //bénítővírus használata v1-en
info --o v1 --n 1 12
info --o v2 --n 1 10 12

//v3
craft StunVirus //bénítővírus elkészítése
use StunVirus v1 //bénítővírus használata v1-en
info --o v1 --n 1 11

//v4
craft StunVirus //bénítővírus elkészítése
use StunVirus v1 //bénítővírus használata v1-en, itt elkopik a
                  //kesztyű, mert ez a 3. alkalom, hogy használódik
info --o v1 --n 1 11
```

- **Elvárt kimenet**

Name:

v2

Genetic codes:

StunVirus 50 Nucleotide

Usables:

StunVirus 3

Name:

v1

Effects:

Name:

v2

Usables:

Effects:

StunVirus 3

Name:

v1

Equipments:

Glove

Name:

v1

Equipments:

7.2.2 Balta használata

- **Leírás**

Egy (medvevírusos) játékost meg lehet ölni a baltával.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- játékos sikeresen felveszi a baltát és a védőköpenyt
- játékos fertőzött laboratóriumra lép és megfertőződik medvevíussal
- Egyik játékos baltával megöl egy másik játékost, aki medvevírus hatása alatt áll
- Védőköpenyes játékos nem fertőződik meg egy fertőzött laboratórium mezőre lépve

- **Bemenet**

add v1 Austria

add v2 Bulgaria

start

//v1
collect //balta felvétele
move Slovenia

//v2
pass
pass

//v1
move Croatia
collect //védőköpeny felvétele
info --o v1 --n 1 10 11

//v2

pass

pass

//v1
move Serbia
pass

//v2
pass
Move Romania //megfertőződik medvevíussal
info --o v2 --n 1 12

//v1
move Romania //nem fertőződik meg a köpeny miatt
use Axe v2 //megöli a medvevírusos virológust
info --o v1 --n 1 5 10 12

- **Elvárt kimenet**

Name:

v1

Usables:

Axe

Equipments:

ProtectiveCloak

Name:

v2

Effect:

BearDanceVirus

Name:

v1

Virologists on the same Tile:

Usables:

Effects:

7.2.3 Védőköpeny hatása

- **Leírás**

A Virológus felvesz egy védőköpenyt. Egy másik Virológus megpróbálja megkenni egy ágenssel, azonban ez nem sikerül neki.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Virológus felveszi a védőköpenyt
- Virológus megtanul genetikai kódot, és felvesz anyagot.
- Virológus elkészíti a megtanult genetikai kód alapján az anyagkészletéből egy ágenst.
- Virológus megkeni a másik Virológust az elkészült ágenssel.
- A védőköpeny megvédi a Virológust az ágenstől.
- A védőköpeny nem védi meg a Virológust az ágenstől.

- **Bemenet**

add v1 Croatia

add v2 Serbia

start

//v1

collect // védőköpeny felvétele

move Montenegro

//v2

collect // amnéziavírus genetikai kódjának megtanulása

move Kosovo

//v1

move Albania

pass

//v2

move Albania

collect // nukleotid felvétele

info --o v2 --n 1 8 9

```
//v1
pass
pass

//v2
craft AmnesiaVirus          //amnéziavírus elkészítése
info --o v2 --n 1 10
setextrandom 1               //A védőköpeny megvéden a kenéstől
use AmnesiaVirus v1          //v2 megpróbálja v1-et megkenni
info --o v1 --n 1 12

//v1
pass
pass

//v2
setextrandom 9999            //A védőköpeny ne véden meg a kenéstől
craft AmnesiaVirus
use AmnesiaVirus v1
info --o v1 --n 1 12
```

• Elvárt kimenet

Name:

v2

Resources:

Nucleotide 100/150
AminoAcid 0/150

Genetic Codes:

AmnesiaVirus 50 Nucleotide

Name:

v2

Usables:

AmnesiaVirus 3 rounds left

Name:

v1

Effects:

Name:

v1

Effects

AmnesiaVirus 3 rounds left

7.2.4 Zsák hatása

- **Leírás**

A virológus felvesz egy zsákat, aminek hatására több anyagot tud eltárolni. Ezután elkezd anyagot gyűjteni, amit így el tud tárolni. Amikor már nem tud semennyit sem gyűjteni, akkor hibaüzenet jelenik meg, és nem használja akcióját. Végül eldobja a zsákat, és így elveszíti anyagkészlete felét.

- **Ellenőrzött funkcionálitás, várható hibahelyek**

- A virológus felvesz egy Zsákat.
- A zsák hatására megnő a virológus maximális anyagmennyisége.
- A virológus vándorol.
- A virológus anyagot gyűjt.
- A virológus anyagkészlete megtelik.
- A virológus eldobja a zsákat.
- Az összes nyersanyag, ami így nem fér el a virológusnál, az elveszik.

- **Bemenet**

add v1 Slovenia

start

//v1

collect

info --o v1 --n 1 11

move Hungry

//v1

move Serbia

move Macedonia

//v1

move Greece

collect

//v1

collect

info --o v1 --n 1 8

collect

//v1

info --o v1 --n 1 8

collect

info --o v1 --n 1 2 8

drop Bag

info --o v1 --n 1 8 11

- **Elvárt kimenet**

Name:

v1

Equipments:

Bag

Name:

v1

Resources:

AminoAcid: 200/300

Nucleotide: 0/300

Name:

v1

Resources:

AminoAcid: 300/300

Nucleotide: 0/300

Can't collect more resources!

Name:

v1

Actions left:

2/2

Resources:

AminoAcid: 300/300

Nucleotide: 0/300

Name:

v1

Resources:

AminoAcid: 150/150

Nucleotide: 0/150

Equipments:

7.2.5 Sikeres lopás

- **Leírás**

Egy virológus lop egy másik virológustól. Mindketten egy mezőn állnak, a másodikon van egy aktív bénítóvírus. Az elsőnek nincsen semmilyen védfelszerelése, a másodiknak van egy kesztyűje és egy zsákja, a kesztyűt szeretné ellopni az első játékos. A második virológusnál van aminosav is, több, mint amennyi még befér az első virológushoz.

- **Ellenőrzött funkcionálitás, várható hibahelyek**

- A bénítóvírus hatása engedélyezi a lopást
- Az első játékos választhat a két védfelszerelés közül, majd a lopás megtörténik
- Az első játékos aminosav készlete nem lépi túl a maximumot
- A második játékos aminosav készlete csak annyival csökkent, amennyi befért az első játékoshoz

- **Bemenet**

add v1 Greece

add v2 Greece

start

//v1

collect

// aminosav felvétele raktárból

move Albania

info --o v1 --n 1 3

// ellenőrizzük, hogy a felvétel és a lépés megtörtént-e

//v2

collect

// aminosav felvétele raktárból

move Albania

//v1

collect

// nukleotid felvétele raktárból

move Serbia

//v2

move Montenegro

move BosniaAndHerzegovina

//v1

move Bulgaria

```

collect                                // genetikai kód megtanulása bénító vírushoz
info --o v1 --n 1 3 9                  // megtanult kód ellenőrzése

//v2
collect                                // kesztyű felvétele
info --o v2 --n 1 11 12                 // felszerelés felvételének ellenőrzése
move Croatia

//v1
craft StunVirus
info --o v1 --n 1 8 10                  // ágens elkészítésének ellenőrzése
move Serbia

//v2
move Slovenia
collect                                // zsák felvétele

//v1
move Croatia
pass

//v2
pass
move Croatia

//v1
use StunVirus v2
info --o v1 --n 1 2 6 10                // ágens használátának ellenőrzése
info --o v2 --n 1 12                    // ágens hatásának ellenőrzése
steal v2
choose 1
info --o v1 --n 1 8 11 12              // új felszerelés, anyagkészlet ellenőrzése
info --o v2 --n 1 8 11 12              // kevesebb felszerelés, anyag ellenőrzése

```

- **Elvárt kimenet**

Name:
v1

Tile:
Albania

Name:
v1

Tile:
Bulgaria

Genetic codes:
StunVirus 50 Nucleotide

Name:
v2

Equipments:
Glove

Effects:
Glove

Name:
v1

Resources:
AminoAcid 100/150
Nucleotide 50/150

Usables:

StunVirus 3

Name:

v1

Actions left:

1/2

Stunned Virologists:

v2

Usables:

Name:

v2

Effects:

StunVirus 3

Choose one:

Glove

Bag

Name:

v1

Resources:

AminoAcid 150/150

Nucleotide 50/150

Equipments:

Glove

Effects:

Glove

Name:

v2

Resources:

AminoAcid 50/150

Nucleotide 0/150

Equipments:

Bag

Effects:

Bag

7.2.6 Sikertelen lopás

- Leírás

Egy játékos megpróbál lopni egy olyan játékostól, aki ugyan egy mezőn van vele, de nincs lebénülva. A program egy hibaüzenettel jelzi neki, hogy ez nem lehetséges.

- Ellenőrzött funkcionalitás, várható hibahelyek

○ Bénítóvírus hiányában nem engedélyezett a lopás

- Bemenet

add v1 Greece

add v2 Greece

start

/v1

steal v2

- Elvárt kimenet

Invalid player as argument!

7.2.7 Medvevírus elkapása

- **Leírás**

Egy virológus egy fertőzött laboratóriumra lép, és mivel nincs ami megvédje, megfertőződik a medvevíussal. Ezután egy raktárra lép, és minden ott tartózkodó játékos is megfertőződik, de az egyikük kivédi vakcinával. A raktáron többé nem vehető fel anyag, mert a medve elpusztította.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- *InfectedLaboratory* medvevírust ad azoknak, akik odalépnek
- Véletlenszerű lépés a medvevírus hatására
- Ha egy medve lép egy mezőre, akkor az ott tartózkodó játékosok megfertőződnek.
- A medvevírus kivédhető, például vakcinával
- Ha medve lép raktárra, onnan többé nem vehető fel anyag

- **Bemenet**

add vmedve Moldova

add v2 Albania

add v3 Greece

//vmedve

pass

pass

//v2

collect

//nukleotid felvétele

move Macedonia

//v3

pass

pass

setNextRandom 0

// 0. szomszédra (Bulgária) lépjen majd a medve

setNextRandom 0

// 0. szomszédra (Görögország) lépjen majd a medve

//vmedve

pass

move Romania

// itt fertőződik meg medvével

info --o vmedve --n 1 12

// fent van-e a medvevírus

//v2

collect

// genetikai kód tanulása vakcinához

craft Vaccine

// vakcina készítése és genetikai kód

//v3

pass

pass

info --o vmedve --n 1 3

// megtörtént-e a két random lépés, tehát Görögo-ban van-e

info --o v3 --n 1 12

// v3 elkapta-e a medvét

info --o Greece --n 1 5

//elpusztította-e a medve az anyagot

//v2

use Vaccine

move Greece

info --o v2 --n 1 12 // vakcina megvédte-e v2-t a medvétől

- **Elvárt kimenet**

Name:
vmedve

Effects:
BearDanceVirus

Name:
v2

Genetic codes:
Vaccine Nucleotide 50

Usables:
Vaccine 3

Name:
vmedve

Tile:
Greece

Name:
v3

Effects:
BearDanceVirus

Name:
Greece

Collectable:
AminoAcid 0

Name:
v2

Effects:

7.2.8 Genetikai kód megtanulása és ágens készítése

- **Leírás**

A Virológus megtanul egy genetikai kódot, nincs elég anyaga, ezért nem tudja elkészíteni belőle az ágenst.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Genetikai kód megtanulása, ágens elkészítése

- **Bemenet**

*add v1 Bulgaria //Virológus hozzáadása a játékoz
start //Játék elindítása*

*collect //v1 virológus felveszi a genetikai kódot
craft StunVirus //v1 megpróbálja elkészíteni*

- **Elvárt kimenet**

Invalid player as argument!

7.2.9 Genetikai kód megtanulása és ágens készítése

- **Leírás**

A Virológus megtanul egy genetikai kódot, van elég anyaga, ezért el tudja készíteni belőle az ágenst.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Genetikai kód megtanulása, ágens elkészítése

- **Bemenet**

<i>add v1 Bulgaria</i>	//Virológus hozzáadása a játékoz
<i>start</i>	//Játék elindítása

<i>collect</i>	//v1 virológus felveszi a genetikai kódot
<i>Info --o v1 --n 1 9 10</i>	//v1 genetikai kódjainak kiiratása

<i>move Greece</i>	//v1 továbblép
<i>collect</i>	//felveszi az anyagot
<i>info --o v1 --n 1 8</i>	

<i>craft StunVirus</i>	//Elkészíti a vírust
------------------------	----------------------

<i>Info --o v1 --n 1 8 9 10</i>	//v1 adatainak kiiratása
---------------------------------	--------------------------

- **Elvárt kimenet**

<i>Name:</i>	//első kiiratás
<i>v1</i>	

<i>Genetic codes:</i>	
<i>StunVirus AminoAcid 50</i>	

<i>Usables:</i>	
-----------------	--

<i>Name:</i>	//második kiiratás
<i>v1</i>	

<i>Resources:</i>	
<i>AminoAcid 100/150</i>	

<i>Name:</i>	//harmadik kiiratás
<i>v1</i>	

<i>Resources:</i>	
<i>AminoAcid 50/150</i>	

<i>Genetic codes:</i>	
<i>StunVirus AminoAcid 50</i>	

<i>Resources:</i>	
<i>AminoAcid 50/150</i>	

<i>Usables:</i>	
<i>StunVirus 3</i>	

7.2.10 Az utolsó genetikai kód megtanulása és a játék megnyerése

- **Leírás**

Ha egy virológus megtanulja az utolsó genetikai kódot, akkor a játék véget ér, ō a nyertes.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Az utolsó genetikai kód megtanulása után aaz adott játékos megnyeri a játékot.

- **Bemenet**

<i>add v1 Greece</i>	//v1 virológus hozzáadása a játékoz
<i>add v2 Slovenia</i>	//v1 virológus hozzáadása a játékoz
<i>start</i>	//Játék elindítása
<i>//v1</i>	
<i>collect</i>	//felveszi az anyagot
<i>move Bulgaria</i>	//továbblép
<i>//v2</i>	
<i>pass</i>	
<i>pass</i>	
<i>//v1</i>	
<i>collect</i>	//felveszi a bénítóvírust (genetikai kód)
<i>move Macedonia</i>	//továbblép
<i>//v2</i>	
<i>pass</i>	
<i>pass</i>	
<i>//v1</i>	
<i>collect</i>	//felveszi a vakcinát (genetikai kód)
<i>move Serbia</i>	//továbblép
<i>//v2</i>	
<i>pass</i>	
<i>pass</i>	
<i>//v1</i>	
<i>collect</i>	//felveszi az amnéziavírust (genetikai kód)
<i>craft Vaccine</i>	//vakcina létrehozása
<i>//v2</i>	
<i>pass</i>	
<i>pass</i>	
<i>//v1</i>	
<i>move Romania</i>	//továbblép
<i>collect</i>	//felveszi a vakcinát (genetikai kód)
<i>//v2</i>	
<i>pass</i>	
<i>pass</i>	
<i>//v1</i>	
<i>move Serbia</i>	//továbblép
<i>move Montenegro</i>	//továbblép
<i>//v2</i>	
<i>pass</i>	
<i>pass</i>	
<i>//v1</i>	
<i>collect</i>	//felveszi a vitustáncot (genetikai kód)

- **Elvárt kimenet**

Player v1 has won the game!

7.2.11 Amnéziavírus hatása

- **Leírás**

A vírológusra amnéziavírust kennek és elfelejtí az összes eddig megtanult genetikai kódját.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Az amnéziavírus hatására elfelejtí a virológus a tanult genetikai kódjait.

- **Bemenet**

<i>add v1 Albania</i>	//Létrehozza az első játékost
<i>add v2 Macedonia</i>	//létrehozza a második játékost

start

<i>//v1</i>	
<i>collect</i>	//Felvezzi a nukleotidot
<i>move Kosovo</i>	//Továbblép

<i>//v2</i>	
<i>collect</i>	//Felvezzi a vakcinát
<i>move Kosovo</i>	//Továbblép

<i>//v1</i>	
<i>move Serbia</i>	//Továbblép
<i>collect</i>	//Felvezzi az Amnézia vírust

<i>//v2</i>	
<i>pass</i>	
<i>pass</i>	

<i>//v1</i>	
<i>craft AmnesiaVirus</i>	//Létrehozza az Amnézia vírust
<i>move Kosovo</i>	//Belép a másik virológus mellé

<i>//v2</i>	
<i>pass</i>	
<i>pass</i>	

info --o v2 --n 1 9

<i>//v1</i>	
<i>use AmnesiaVirus v2</i>	
<i>pass</i>	

info --o v2 --n 1 9

- **Elvárt kimenet:**

Name:

v2

Genetic Codes:

Vaccine

Name:

v2

Genetic Codes:

7.2.12 Vitustáncvírus alkalmazása és hatása

- **Leírás**

A vírológusra vitustánc vírust alkalmaznak.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Az vitustánc vírus hatására a virológus táncolni kezdd.

- **Bemenet**

add v1 Albania

//Létrehozza az első játékest

add v2 Montenegro

//létrehozza a második játékest

start

//v1

collect

//Felvezzi a nukleotidot

move Monenegro

//Továbblép

//v2

pass

//Felvezzi a vakcinát

pass

//Továbblép

//v1

collect

//Felvezzi a Vitustáncvírus genetikai kódját

craft VitusDanceVirus

//Elkészíti a Vitustáncvírust

//v2

pass

pass

//v1

use VitusDanceVirus v2

//Használja a Vitustáncvírust

SetNextRandom 4

//v2 Szerbiába lépjen tovább

SetNextRandom 4

//v2 Szerbiából Magyarországra lépjen tovább

info --o v2 --n 1 3 12

pass

//v2 nem hajt végre akciót

info --o v2 --n 1 3

- **Elvárt kimenet:**

Name:

v2

Active Tile:

Montenegro

ActiveEffects:

VitusDanceVirus

Name:

v2

ActiveTile:

Hungary

7.2.13 Vakcina alkalmazása és hatása

- **Leírás**

A vírológus vakcinát alkalmaz magára, ami megvédi egy másik virológus kenésétől. A másik virológus elkeszített ágense elveszik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- A vakcina megvédi a virológust egy másik virológus ágensétől.

- **Bemenet**

<i>add v1 Albania</i>	//Létrehozom az első játékost
<i>add v2 Albania</i>	//létrehozza a második játékost

start

//v1	
<i>collect</i>	//Felvezzi a nukleotidot
<i>move Macedonia</i>	//Továbblép

//v2	
<i>collect</i>	//Felvezzi a nukleotidot
<i>move Montenegro</i>	//Továbblép

//v1	
<i>collect</i>	//Felvezzi a Vakcina genetikai kódját
<i>craft Vaccine</i>	//Létrehozza a Vakcinát

//v2	
<i>collect</i>	//Felvezzi Vitustáncvírus genetikai kódját
<i>craft VitzusDanceVirus v1</i>	//Létrehozza a Vitustáncvírust

//v1	
<i>use Vaccine v1</i>	//Alkalmazza magára a vakcinát
<i>move Kosovo</i>	//Továbblép

info v1 1 12

//v2	
<i>move Kosovo</i>	
<i>use VitzusDanceVirus v1</i>	

info v1 1 12
Info v2 1 10

- **Elvárt kimenet:**

Name:

v1

Active Effects:

Vaccine

Name:

v1

Active Effects:

Vaccine

Name:

v2

Crafted Agents:

-

7.3 A tesztelést támogató programok tervezése

Minden teszteset futtatásához lesz egy .bat fájl, ami futtatja a programot, majd egy .txt fájlból olvas a bemeneti nyelvben definiált parancsokat. A tesztesetek során egy vagy több info parancs is lefut, ezek a standard kimenetre írnak, de ez egy fájlba lesz irányítva. A tesztesetek minden quit parancssal fejeződnek be, ezzel ér véget a programunkat futtató parancs. Végül a .bat fájl összeveti a kimeneti fájlt a teszteset elvárt kimenetével a Windows beépített fc (file compare) parancsával. Ez jelzi, ha van eltérés a két fájl között.

7.4 Napló

Kezdet	Időtartam	Részttvevők	Leírás
2022.04.04. 14:00	1 óra	Fekete Jurásek Kászonyi Domonkos Száraz	Értekezlet. Döntés: Osztályok és tesztesetek szétosztása.
2022.04.04. 17:00	2 óra	Fekete	Mezők és Map osztály attribútumai és metódusai.
2022.04.04. 19:00	2 óra	Jurásek	Equipmentek osztályleírásai
2022.04.04. 18:00	2 óra	Domonkos	Virologist, Resource osztályleírása
2022.04.04. 17:00	3 óra	Fekete	Tesztesetek be- és kimenetei
2022.04.04. 17:00	2 óra	Domonkos	Tesztesetek megírása
2022.04.04. 19:00	3 óra	Száraz	Controller, Inventory, GeneticCode osztályleírások
2022.04.07. 20:00	2 óra	Kászonyi	Ágens és típusainak leírása
2022.04.08. 10:00	2 óra	Jurásek	Tesztesetek megírása
2022.04.08. 10:00	2 óra	Kászonyi	Tesztesetek megírása
2022.04.08. 15:00	3 óra	Jurásek	A prototípus kezelői felületének kialakítása, mások munkájának ellenőrzése.
2022.04.08. 16:00	2 óra	Fekete	Mások munkájának ellenőrzése
2022.04.08. 18:00	3 óra	Száraz	Kesztyű használata, Balta használata tesztek megírása, mások munkájának ellenőrzése
2022.04.09. 16:00	2 óra	Domonkos	Mások munkájának ellenőrzése
2022.04.10. 11:00	1 óra	Kászonyi	Mások munkájának ellenőrzése
2022.04.10. 18:00	2 óra	Domonkos	Dokumentum ellenőrzése, formázása

10. Prototípus beadása

52 – *service_unavailable*

Konzulens:
Micskei Zoltán

Csapattagok

Fekete Sámuel	GJ8J3A	feketesamu@gmail.com
Száraz Dániel	GT5X34	szdani2012@gmail.com
Jurásek Jónás	PH4QFK	jurasekjonas12@gmail.com
Domonkos Ádám	CWGYWC	domonkosadam01@gmail.com
Kászonyi Zsombor Marcell	DCE2Q1	kaszonyirelli@gmail.com

2022.02.23.

8. Prototípus beadása

8.1 Fordítási és futtatási útmutató

8.1.1 Fájllista

Fájl neve	Méret (KB)	Keletkezés ideje	Tartalom
/src/agents/Agent.java	4	2022.04.22.	Ágens absztrakt űsosztály.
/src/agents/AmnesiaVirus.java	2	2022.04.22.	Amnézia vírus osztálya.
/src/agents/BearDanceVirus.java	3	2022.04.22.	Medvevírus osztálya
/src/agents/StunVirus.java	2	2022.04.22.	Bénító vírus osztálya.
/src/agents/Vaccine.java	2	2022.04.22.	Vakcina osztálya.
/src/agents/VitusDanceVirus.java	2	2022.04.22.	Vitus Tánc vírus osztálya.
/src/equipments/Axe.java	2	2022.04.22.	Balta osztálya.
/src/equipments/Bag.java	2	2022.04.22.	Hátizsák osztálya.
/src/equipments/Equipment.java	3	2022.04.22.	Védőfelszerelés absztrakt űsosztály.
/src/equipments/Glove.java	2	2022.04.22.	Kesztyű osztálya.
/src/equipments/ProtectiveCloak.java	2	2022.04.22.	Védőköpeny osztálya.
/src/equipments/UsableEquipment.java	1	2022.04.22.	Használható felszerlek interfész.
/src/main/Collectable.java	1	2022.04.22.	Gyűjthető interfész.
/src/main/ConsoleView.java	13	2022.04.22.	Parancssori nézet osztálya.
/src/main/Constants.java	1	2022.04.22.	Konstansok osztálya.
/src/main/Controller.java	9	2022.04.22.	Vezérlő osztálya.
/src/main/Effect.java	2	2022.04.22.	Hatások interfész.
/src/main/GeneticCode.java	4	2022.04.22.	Genetikai kódok osztálya.
/src/main/Inventory.java	8	2022.04.22.	Felszerelés osztálya.
/src/main/Main.java	2	2022.04.22.	A fő osztály.
/src/main/Map.java	11	2022.04.22.	Térkép osztálya.
/src/main/OutputGenerator.java	5	2022.04.22.	A parancssori kimenetet formázó osztály.
/src/main/Resource.java	4	2022.04.22.	Alapanyagok osztálya.
/src/main/ResourceType.java	1	2022.04.22.	Alapanyagtípusok enumerációja.
/src/main/SRandom.java	1	2022.04.22.	Saját Random osztály.
/src/main/View.java	1	2022.04.22.	A kinézet interfész.
/src/main/Virologist.java	9	2022.04.22.	Virológus osztálya.
/src/tiles/EmptyTile.java	1	2022.04.22.	Üres mező osztálya.
/src/tiles/InfectedLaboratory.java	2	2022.04.22.	Fertőzött laboratórium osztálya.
/src/tiles/Laboratory.java	1	2022.04.22.	Labor mező osztálya.
/src/tiles/Safehouse.java	1	2022.04.22.	Óvóhely mező osztálya.

/src/tiles/Tile.java	5	2022.04.22.	Mező absztrakt űsosztály.
/src/tiles/Warehouse.java	1	2022.04.22.	Raktár mező osztálya.
/compiled/compile.bat	1	2022.04.22.	Fordítást végző parancsok.
/test/821_input.txt	1	2022.04.22.	8.2.1 teszteset bemenetei.
/test/821_expected_output.txt	1	2022.04.22.	8.2.1 teszteset elvárt kimenete.
/test/821_run.bat	1	2022.04.22.	8.2.1 teszteset futtatható fájlja.
/test/822_input.txt	1	2022.04.22.	8.2.2 teszteset bemenetei.
/test/822_expected_output.txt	1	2022.04.22.	8.2.2 teszteset elvárt kimenete.
/test/822_run.bat	1	2022.04.22.	8.2.2 teszteset futtatható fájlja.
/test/823_input.txt	1	2022.04.22.	8.2.3 teszteset bemenetei.
/test/823_expected_output.txt	1	2022.04.22.	8.2.3 teszteset elvárt kimenete.
/test/823_run.bat	1	2022.04.22.	8.2.3 teszteset futtatható fájlja.
/test/824_input.txt	1	2022.04.22.	8.2.4 teszteset bemenetei.
/test/824_expected_output.txt	1	2022.04.22.	8.2.4 teszteset elvárt kimenete.
/test/824_run.bat	1	2022.04.22.	8.2.4 teszteset futtatható fájlja.
/test/825_input.txt	1	2022.04.22.	8.2.5 teszteset bemenetei.
/test/825_expected_output.txt	1	2022.04.22.	8.2.5 teszteset elvárt kimenete.
/test/825_run.bat	1	2022.04.22.	8.2.5 teszteset futtatható fájlja.
/test/826_input.txt	1	2022.04.22.	8.2.6 teszteset bemenetei.
/test/826_expected_output.txt	1	2022.04.22.	8.2.6 teszteset elvárt kimenete.
/test/826_run.bat	1	2022.04.22.	8.2.6 teszteset futtatható fájlja.
/test/827_input.txt	1	2022.04.22.	8.2.7 teszteset bemenetei.
/test/827_expected_output.txt	1	2022.04.22.	8.2.7 teszteset elvárt kimenete.
/test/827_run.bat	1	2022.04.22.	8.2.7 teszteset futtatható fájlja.
/test/828_input.txt	1	2022.04.22.	8.2.8 teszteset bemenetei.

/test/828_expected_output.txt	1	2022.04.22.	8.2.8 teszteset elvárt kimenete.
/test/828_run.bat	1	2022.04.22.	8.2.8 teszteset futtatható fájlja.
/test/829_input.txt	1	2022.04.22.	8.2.9 teszteset bemenetei.
/test/829_expected_output.txt	1	2022.04.22.	8.2.9 teszteset elvárt kimenete.
/test/829_run.bat	1	2022.04.22.	8.2.9 teszteset futtatható fájlja.
/test/8210_input.txt	1	2022.04.22.	8.2.10 teszteset bemenetei.
/test/8210_expected_output.txt	1	2022.04.22.	8.2.10 teszteset elvárt kimenete.
/test/8210_run.bat	1	2022.04.22.	8.2.10 teszteset futtatható fájlja.
/test/8211_input.txt	1	2022.04.22.	8.2.11 teszteset bemenetei.
/test/8211_expected_output.txt	1	2022.04.22.	8.2.11 teszteset elvárt kimenete.
/test/8211_run.bat	1	2022.04.22.	8.2.11 teszteset futtatható fájlja.
/test/8212_input.txt	1	2022.04.22.	8.2.12 teszteset bemenetei.
/test/8212_expected_output.txt	1	2022.04.22.	8.2.12 teszteset elvárt kimenete.
/test/8212_run.bat	1	2022.04.22.	8.2.12 teszteset futtatható fájlja.
/test/8213_input.txt	1	2022.04.22.	8.2.13 teszteset bemenetei.
/test/8213_expected_output.txt	1	2022.04.22.	8.2.13 teszteset elvárt kimenete.
/test/8213_run.bat	1	2022.04.22.	8.2.13 teszteset futtatható fájlja.

8.1.2 Fordítás

A compiled/compile.bat fájlon dupla kattintással létrejön a prototype.jar fájl a gyökérkönyvtárban.

A .bat fájl futtatásakor előfordulhat, hogy a Windows Defender szól, hogy veszélyes lehet a fájl futtatása. Ilyenkor kattintsunk a More info-ra, majd a a Run anyway gombra. A .bat fájl az alább olvasható két parancsot hajtja végre, amelyek nem károsak a számítógép számára.

Alternatív fordítási útmutató:

Az alábbi két parancsot kell egymás után kiadni a parancssori terminálban. A parancsok kiadása előtt el kell navigálni a projekt compiled mappájába a cd parancsal. (A fájlkezelőben az aktuális mappát jelölő eszköztárba a cmd szót begépelve, automatikusan úgy nyílik meg a paracssori terminál.). A futtatható .bat fájl pontosan ezeket a parancsokat futtatja.

```
javac -d ./src/agents/*.java ./src/equipments/*.java ./src/main/*.java ./src/skeleton/*.java ./src/tiles/*.java
jar cfe ..prototype.jar main/Main agents/*.class equipments/*.class main/*.class skeleton/*.class tiles/*.class
```

8.1.3 Futtatás

Tesztekhez tartozó fájlok

A test mappában találhatóak minden tesztesethez a bemeneti parancsok, illetve az elvárt kimenetek. A fájlok neve a következőképpen utal a tesztesetre:

XXX_input.txt → bemeni parancsokat tartalmazó fájl

XXX_expected_output.txt → elvárt kimenetet tartalmazó fájl

Az XXX helyére a teszteset három részből álló sorszámról kell helyettesíteni határoló karakterek nélkül.

Futtatás

Minden tesztesethez készült egy futtatható.bat fájl. Erre duplán kattintva elindítható az adott tesztelés, ami előállítja a teszteset kimenetét, és összehasonlítja az elvárt kimenetet tartalmazó fájllal. Az eredményt kiírja a parancssori terminálra.

A .bat fájl futtatásakor előfordulhat, hogy a Windows Defender szól, hogy veszélyes lehet a fájl futtatása. Ilyenkor kattintsunk a More info-ra, majd a a Run anyway gombra. A .bat fájl az alább olvasható két parancsot hajtja végre, amelyek nem károsak a számítógép számára.

XXX_run.bat → futtatható teszteset

XXX_output.txt → a teszteset futtatása után generált fájl, ami tartalmazza az adott bemenetre adott kimenetet

Az XXX helyére a teszteset három részből álló sorszámat kell helyettesíteni határoló karakterek nélkül.

Alternatív futtatási útmutató:

Az alábbi két parancsot kell egymás után kiadni a parancssori terminálban. A parancsok kiadása előtt el kell navigálni a projekt test mappájába a cd parancccsal. (A fájlkezelőben az aktuális mappát jelölő eszköztárba a cmd szót begépelve, automatikusan úgy nyílik meg a paracssori terminál.):

```
java -jar ../prototype.jar debug < XXX_input.txt > XXX_output.txt
```

```
fc /w XXX_output.txt XXX_expected_output.txt
```

A futtatható fájl lehetséges kimenetei:

Minden esetben kiírja az összehasonlított fájlok nevét:

Comparing files XXX_output.txt and XXX_EXPECTED_OUTPUT.TXT

Ha teszteset sikeres volt, és megegyezik a kimeneti fájl tartalma az elvártéval, a következő szöveg jelenik meg:

FC: no differences encountered

Ha nem volt sikeres a teszteset, és eltér a két fájl tartalma, akkor a két fájl eltérését írja ki.

8.2 Tesztek jegyzőkönyvei

8.2.1 Kesztyű használata

Tesztelő neve	Száraz Dániel
Teszt időpontja	2022.04.22. 12:00

8.2.2 Balta használata

Tesztelő neve	Száraz Dániel
Teszt időpontja	2022.04.22. 13:30

Tesztelő neve	Száraz Dániel
Teszt időpontja	2022.04.22. 13:00
Teszt eredménye	Minden megfelelt a vártnak, kivéve még a várt dolgok után a medvék megnyerték a játékot.
Lehetséges hibaok	Azért nyertek a medvék, mert amikor megölünk valakit baltával, akkor a megölt játékos kikerül a játékosok közül, viszont a megfertőzött játékosok közül nem vettük ki.
Változtatások	Játékos megölése esetén, ha a megölt játékos fertőzött volt, akkor a fertőzött játékosok közül is kiveszük.

8.2.3 Védőköpeny Hatása

Tesztelő neve	Jurásek Jónás
Teszt időpontja	2022.04.20. 15:00

8.2.4 Zsák Hatása

Tesztelő neve	Jurásek Jónás
Teszt időpontja	2022.04.20. 16:00

Tesztelő neve	Jurásek Jónás
Teszt időpontja	2022.04.20. 15:10
Teszt eredménye	A Zsák lekerülése után nem csökkent vissza az eredeti értékre a maximum anyagmennyiség
Lehetséges hibaok	A játékos kör végén már nem futott le az endTurnImpact-je a Zsáknak, a kikerüléshez pedig még nem került be.
Változtatások	Az Equipmentek eldobásakor lefut az endTurnImpact-juk.

8.2.5 Sikeres lopás

Tesztelő neve	Fekete Sámuel
Teszt időpontja	2022.04.18 13:00

Tesztelő neve	Fekete Sámuel
Teszt időpontja	2022.04.18 12:00
Teszt eredménye	<p>Name: v1</p> <p>Title: Albania</p> <p>Name: v1</p> <p>Title: Bulgaria</p> <p>Genetic codes: GeneticCode #6: StunVirus: 3 rounds left, [Nucleotide: 50]</p> <p>Name: v2</p> <p>Equipments: Glove: 3 uses left</p> <p>Effects: Glove: 3 uses left</p> <p>Name: v1</p> <p>Resources: AminoAcid: 100/150 Nucleotide: 50/150</p> <p>Usables: StunVirus: 3 rounds left</p> <p>Name: v1</p> <p>Actions left: 1/2</p> <p>Stunned Virologists:</p>

	<p>Usables:</p> <p>Name: v2</p> <p>Effects:</p> <ul style="list-style-type: none"> Glove: 2 uses left Bag <p>Choose one:</p> <ol style="list-style-type: none"> 1. Glove: 2 uses left 2. Bag <p>You can't steal from v2!</p> <p>Name: v1</p> <p>Resources:</p> <ul style="list-style-type: none"> AminoAcid: 100/150 Nucleotide: 50/150 <p>Equipments:</p> <p>Effects:</p> <ul style="list-style-type: none"> StunVirus: 1 rounds left <p>Name: v2</p> <p>Resources:</p> <ul style="list-style-type: none"> AminoAcid: 100/150 Nucleotide: 0/150 <p>Equipments:</p> <ul style="list-style-type: none"> Glove: 2 uses left Bag <p>Effects:</p> <ul style="list-style-type: none"> Glove: 2 uses left Bag
Lehetséges hibaok	Olyan virológusra próbáltunk bénítóvírust kenni, akinek kesztyűje volt, aki így az ágenst rögtön visszadobta, és nem tudtunk lopni tőle.
Változtatások	Át kellett alakítani a tesztet, hogy nem kesztyűt, hanem köpenyt vegyen fel v2, és ezt is lopjuk el tőle.

8.2.6 Sikertelen lopás

Tesztelő neve	Fekete Sámu
Teszt időpontja	2022.04.18 13:30

8.2.7 Medvevírus elkapása

Tesztelő neve	Fekete Sámuel
Teszt időpontja	2022.04.20 18:00

Tesztelő neve	Fekete Sámuel
Teszt időpontja	2022.04.18 14:00
Teszt eredménye	Két medvevírusos virológus örökké fertőzi egymást
Lehetséges hibaok	Egy medve virológus minden szomszédját megfertőzi, azt is, aki már fertőzött.
Változtatások	A Controller számon tartja, hogy mely virológusok fertőződtek már meg, és csak olyat lehet megfertőzni, aki eddig még nem volt megfertőződve.

8.2.8 Genetikai kód megtanulása és ágens használata

Tesztelő neve	Domonkos Ádám
Teszt időpontja	2022.04.21. 18:00

8.2.9 Genetikai kód megtanulása és ágens használata

Tesztelő neve	Domonkos Ádám
Teszt időpontja	2022.04.21. 18:10

8.2.10 Az utolsó genetikai kód megtanulása és a játék megnyerése

Tesztelő neve	Domonkos Ádám
Teszt időpontja	2022.04.21. 19:00

Tesztelő neve	Domonkos Ádám
Teszt időpontja	2022.04.21. 18:30
Teszt eredménye	[üres]
Lehetséges hibaok	Nem hívódik meg az eredményt kiíró függvény
Változtatások	Kellett egy extra break a játékoson iteráló ciklusba, hogy ha valaki megnyeri a játékot, azonnal ki is írja.

8.2.11 Amnézia vírus hatása

Tesztelő neve	Kászonyi Zsombor
Teszt időpontja	2022.04.22. 15:00

8.2.12 Vakcina védelme

Tesztelő neve	Kászonyi Zsombor
Teszt időpontja	2022.04.22. 15:00

8.2.13 VitusTánc vírus hatása

Tesztelő neve	Kászonyi Zsombor
Teszt időpontja	2022.04.22. 15:00

8.3 Értékelés

Tag neve	Tag neptun	Munka százalékban
Domonkos Ádám	CWGYWC	20%
Fekete Sámuel	GJ8J3A	20%
Jurásek Jónás	PH4QFK	20%
Kászonyi Zsombor Marcell	DCE2Q1	20%
Száraz Dániel	GT5X34	20%

8.4 Napló

Kezdet	Időtartam	Résznevők	Leírás
2022.04.13. 16:00	4 óra	Jurásek	Equipment osztályok elkészítése, ConsoleView osztály
2022.04.14. 14:00	2 óra	Jurásek	ConsoleView ellenőrzése
2022.04.12 16:00	2 óra	Kászonyi	Függvények megírása, befejezése.
2022.04.15 12:00	2 óra	Fekete	Tile és leszármazott osztályainak megírása, Map megírása
2022.04.15 16:00	2 óra	Fekete	ConsoleView osztály továbbfejlesztése, OutputGenerator osztály megírása
2022.04.18 18:00	1 óra	Fekete	Teszteleisi környezet előkészítése
2022.04.18 19:00	5 óra	Száraz	Controller, Inverntory és GeneticCode osztályok kiegészítése és megírása.

2022.04.19 12:00	3 óra	Fekete	Tesztesetek megírása, és debugolás miattuk
2022.04.19. 16:00	2 óra	Jurásek	Hibakeresés, debugolás
2022.04.21. 12:00	2 óra	Jurásek	A tesztesetek ellenőrzése
2022.04.21. 16:00	2 óra	Domonkos	Tesztesetek fájljainak előkészítése
2022.04.21. 18:00	2 óra	Domonkos	Tesztesetek során előjött hibák javítása
2022.04.22. 11:00	2 óra	Kászonyi	Tesztek megírása, ellenőrzése
2022.04.22. 12:00	2 óra	Száraz	Tesztesetek ellenőrzése, előjött hibák javítása
2022.04.22. 14:00	2 óra	Domonkos	Tesztek jegyzőkönyvei, fordítási útmutató megírása
2022.04.23 14:00	1 óra	Kászonyi	Dokumentum átnézése, kiegészítése

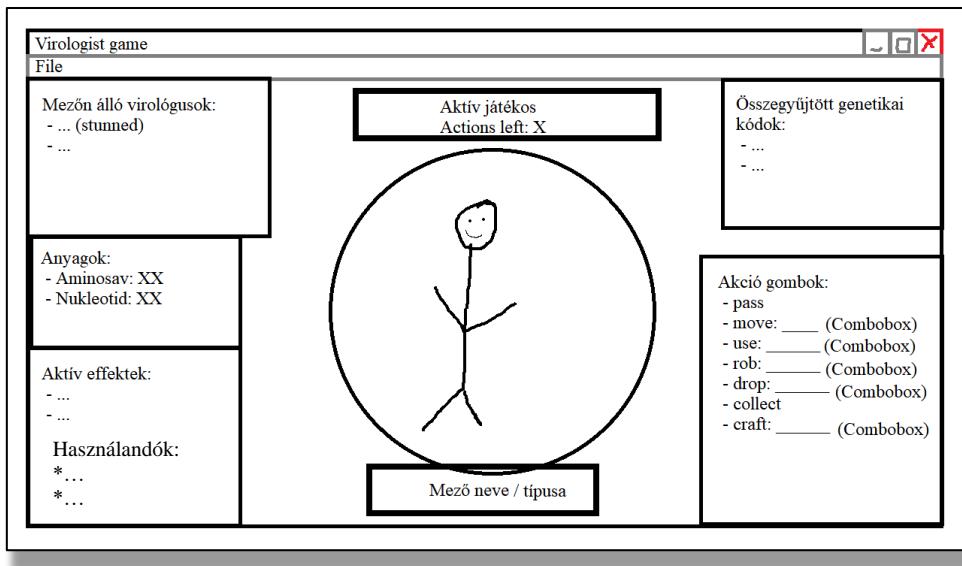
9. Grafikus felület specifikációja

9.1 A grafikus interfész

Menü kinézete:



Játék felület kinézete:



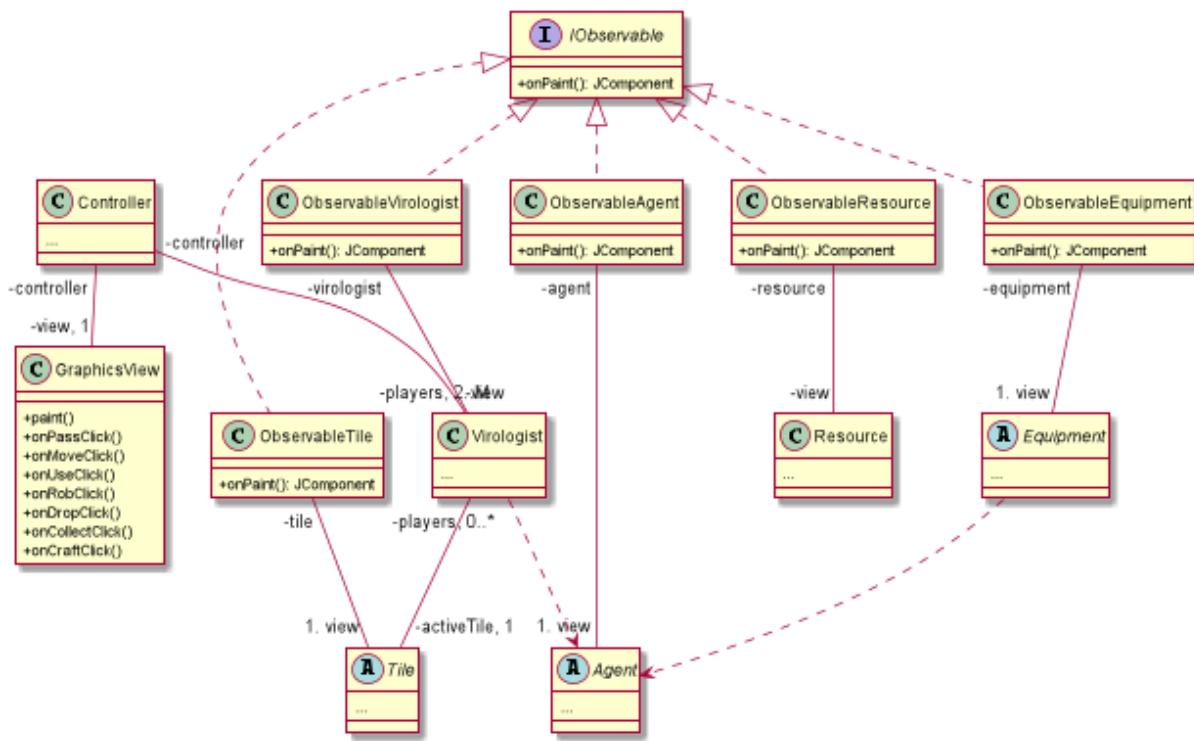
9.2 A grafikus rendszer architektúrája

MVC architektúra szerint készül a projekt. minden megjeleníthető osztályhoz tartozik egy Observable interfészt megvalósító megjelenítő osztály. Ezek az osztályok felelősek a grafikus komponensek létrehozásáért, amit a GraphicsView osztály fog megjeleníteni a képernyőn.

9.2.1 A felület működési elve

Minden a grafikus felületen meglévő objektum összekapcsolódik a játék egy objektumával, A grafikus felület kirajzoláskor a kontrollerhez fordul, majd az éppen aktuális játékosnak és birtokolt objektumainak megjelenítőjét a modellen keresztül éri el. Ezután a megjelenítést minden objektumnak a saját megjelenítő osztálya végzi. Kevert alapelvű, körök kezdetekor a megjelenítő oszálly kérdezi le az adatokat. Felhasználói interakció során, a megváltozott objektum jeleníti meg magát.

9.2.2 A felület osztály-struktúrája



9.3 A grafikus objektumok felsorolása

9.3.1 Controller

- Felelősség**
A képernyőre való megjelenítésért felelős függvény. Amikor az Ő felelőssége a megjelenítendő adatok gyűjtése, illetve aztán azok a képernyőre megjelenítése.

- Ősosztályok**

- Interfészek**

- Attribútumok**

`private GraphicsView graphicsView:` A játék megjelenítője.

- Metódusok**

9.3.2 GraphicsView

- **Felelősség**

A képernyőre való megjelenítésért felelős függvény. Amikor az Ő felelőssége a megjelenítendő adatok gyűjtése, illetve aztán azok a képernyőre megjelenítése.

- **Ősosztályok**
 -

- **Interfészek**
 -

- **Attribútumok**

private Controller controller: A controller.

- **Metódusok**

public void Paint(): A Controllertől elkéri az soron lévő játékost, és annak összes birtokolt objektumától elkéri a megjelenítőjét.

public void onPassClick(): Meghívja a Controller pass metódusát.

public void onMoveClick(): Megnézi, hogy melyik mező van a mozgáshoz tartozó ComboBoxban, és ezzel meghívja a Controller move függvényét.

public void onUseClick(): Megnézi, hogy melyik használható objektum van a használathoz tartozó ComboBoxban, és ezzel meghívja a Controller use függvényét.

public void onRobClick(): Megnézi, hogy melyik mező van a rabláshoz tartozó ComboBoxokban, és ezekkel meghívja a Controller rob függvényét.

public void onDropClick(): Megnézi, hogy melyik védőfelszerelés van az eldobáshoz tartozó ComboBoxban, és ezzel meghívja a Controller drop függvényét.

public void onCollectClick(): Meghívja a Controller collect metódusát.

public void onCraftClick(): Megnézi, hogy melyik genetikai kód van a craftoláshoz tartozó ComboBoxban, és ezzel meghívja a Controller craft függvényét.

9.3.3 IObservable

- **Felelősség**

Egy interface, amit az összes kirajzoló osztály megvalósít. Felelőssége a hozzá tartozó objektum megjelenítése.

- **Ősosztályok**
 -

- **Interfészek**
 -

- **Attribútumok**
 -

- **Metódusok**

public JComponent onPaint(): Az objektum létrehozza a megfigyelt elemről a grafikus UI elemet, amit a megjelenítésért felelős GraphicsView fog a képernyőre rajzolni.

9.3.4 ObservableAgent

- **Felelősség**

Az ágens objektumok megjelenítéséért felelős osztály.

- **Ősosztályok**

-

- **Interfészek**

IObservable

- **Attribútumok**

private Agent agent: A megjelenítendő ágens

- **Metódusok**

public void onPaint(): Az agent attribútumban tárolt ágensnek létrehozza a megjelenítendő grafikus felületet.

9.3.5 ObservableEquipment

- **Felelősség**

Az Equipment objektumok megjelenítéséért felelős osztály.

- **Ősosztályok**

-

- **Interfészek**

IObservable

- **Attribútumok**

private Equipment equipment: A megjelenítendő felszerelés.

- **Metódusok**

public void onPaint(): Az equipment attribútumban tárolt felszerelésnek létrehozza a megjelenítendő grafikus felületet.

9.3.6 ObservableResource

- **Felelősség**

Az Equipment objektumok megjelenítéséért felelős osztály.

- **Ősosztályok**
 -
- **Interfészek**
 - IObservable
- **Attribútumok**
 - private Resource resource:** A megjelenítendő anyag.
- **Metódusok**
 - public JComponent onPaint():** A resource attribútumban tárolt anyagnak létrehozza a megjelenítendő grafikus felületet.

9.3.7 ObservableTile

- **Felelősség**
 - A mezők megjelenítéséért felelős osztály.
- **Ősosztályok**
 -
- **Interfészek**
 - IObservable
- **Attribútumok**
 - private Tile tile:** A megjelenítendő mező.
- **Metódusok**
 - public JComponent onPaint():** A tile attribútumban tárolt mezőnek létrehozza a megjelenítendő grafikus felületet.

9.3.8 ObservableVirologist

- **Felelősség**
 - A Virologist objektumok megjelenítéséért felelős osztály.
- **Ősosztályok**
 -
- **Interfészek**
 - IObservable
- **Attribútumok**
 - private Virologist virologist:** A megjelenítendő virológus.

- **Metódusok**

public JComponent onPaint(): A virologist attribútumban tárolt virológusnak létrehozza a megjelenítendő grafikus felületet.

9.3.9 ObservableGeneticCode

- **Felelősség**

A GeneticCode objektumok megjelenítéséért felelős osztály.

- **Ősosztályok**

-

- **Interfészek**

IObservable

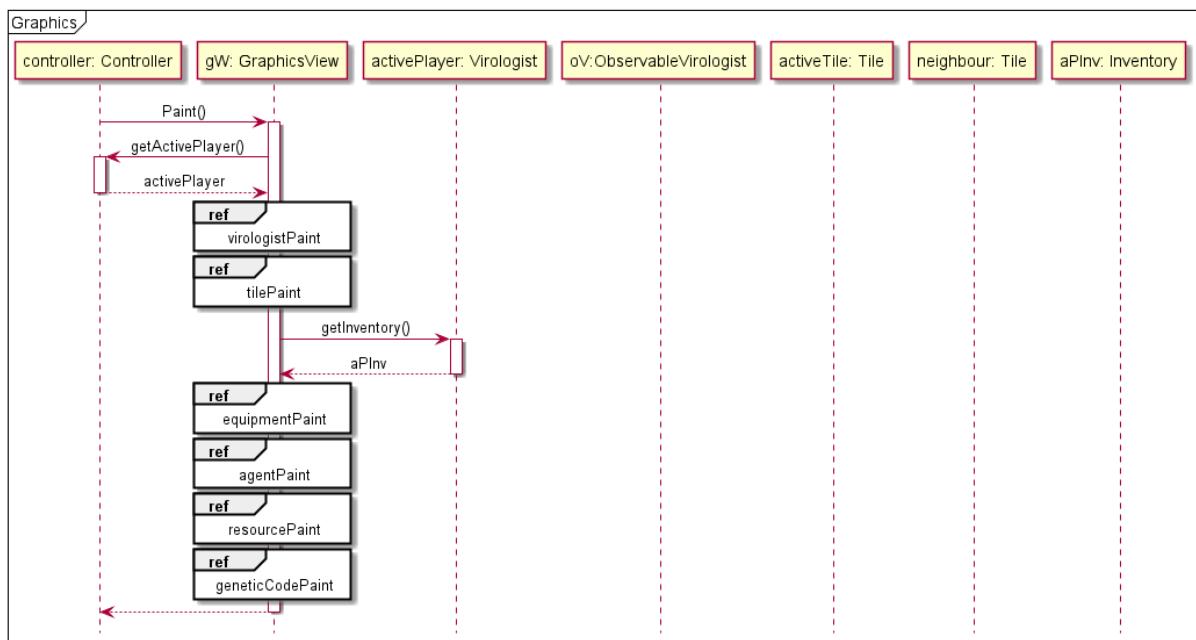
- **Attribútumok**

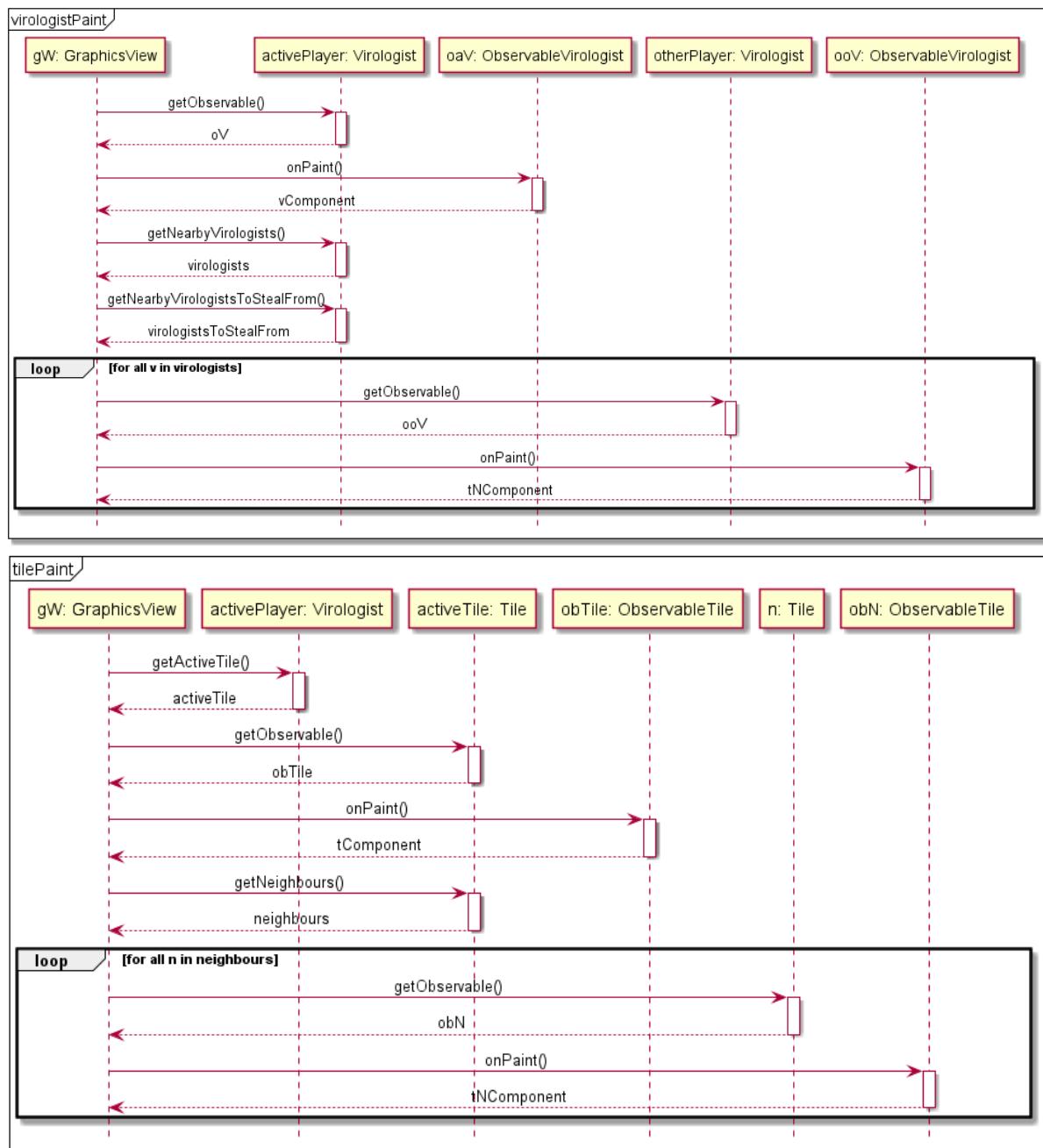
private GeneticCode code: A megjelenítendő genetikai kód.

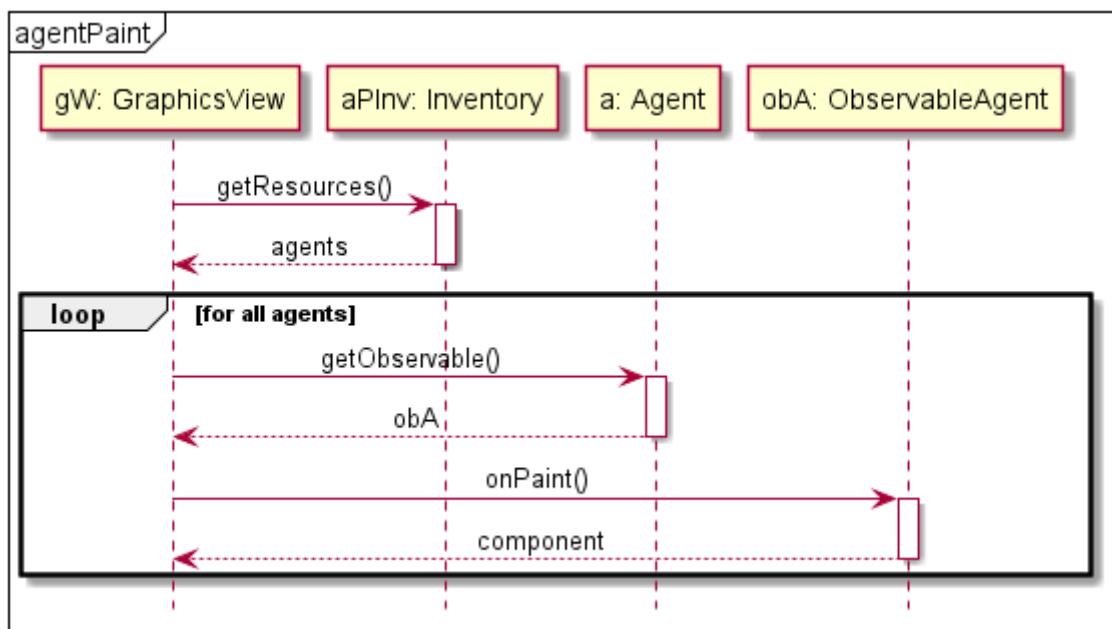
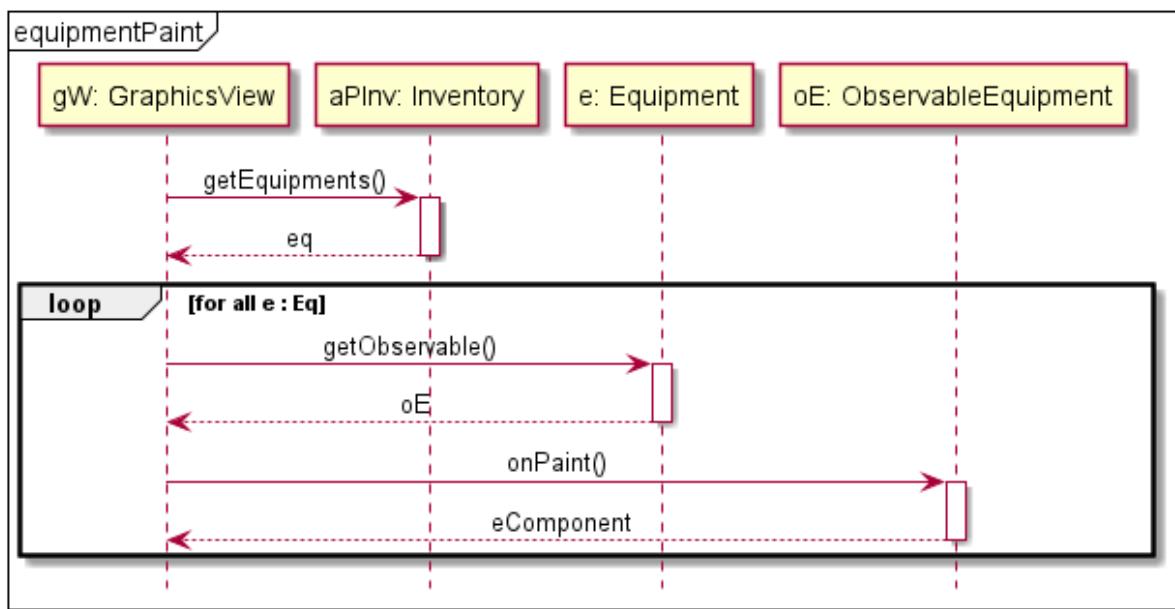
- **Metódusok**

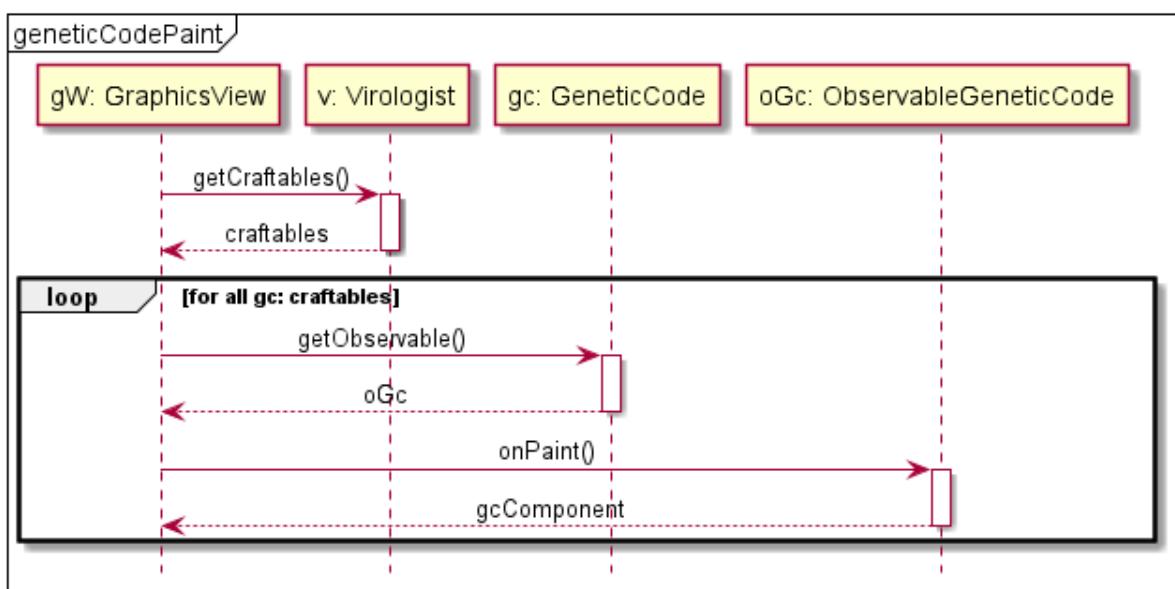
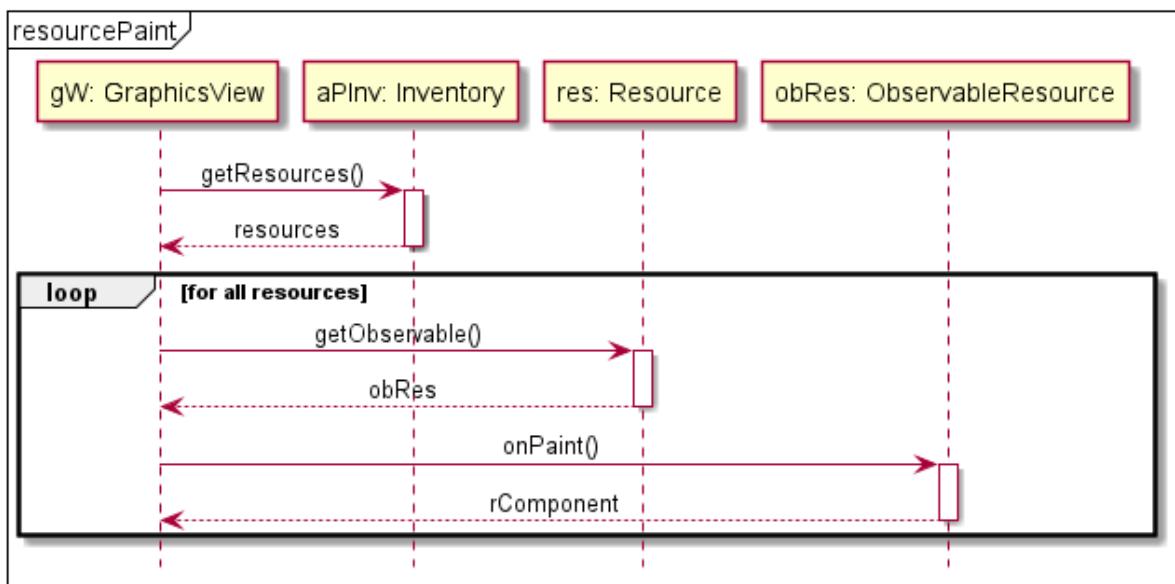
public JComponent onPaint(): A code attribútumban tárolt genetikai kódnak létrehozza a megjelenítendő grafikus felületet.

9.4 Kapcsolat az alkalmazói rendszerrel









9.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2022.04.26 18:00	1 óra	Száraz	JavaFX tesztelése
2022.04.28. 16:00	2 óra	Domonkos	Grafikus felületek tervezése
2022.04.29 10:00	2 óra	Kászonyi	Osztályleírások
2022.04.29 10:00	2 óra	Fekete	GraphicsView osztály kitalálása, megírása
2022.04.29 12:00	3 óra	Jurásek	Szekvenciadiagram készítés
2022.04.29 12:30	2 óra	Száraz	Szekvenciadiagram és osztálydiagram készítés
2022.04.29. 14:00	2 óra	Domonkos	Osztályleírások befejezése
2022.04.30. 10:00	2 óra	Jurásek	Dokumentum ellenőrzése

13. Grafikus változat elkészítése

52 - service_unavailable

Konzulens:
Micskei Zoltán

Csapattagok

Fekete Sámuel	GJ8J3A	feketesamu@gmail.com
Száraz Dániel	GT5X34	szdani2012@gmail.com
Jurásek Jónás	PH4QFK	jurasekjonas12@gmail.com
Domonkos Ádám	CWGYWC	domonkosadam01@gmail.com
Kászonyi Zsombor Marcell	DCE2Q1	kaszonyirelli@gmail.com

2022.02.23.

10. Grafikus változat beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret (KB)	Keletkezés ideje	Tartalom
agents\Agent.java	4	2022.05.15.	Ágens absztrakt űsosztály.
agents\AmnesiaVirus.java	2	2022.05.15.	Amnézia vírus osztálya.
agents\BearDanceVirus.java	3	2022.05.15.	Medvevírus osztálya
agents\StunVirus.java	2	2022.05.15.	Bénító vírus osztálya.
agents\Vaccine.java	2	2022.05.15.	Vakcina osztálya.
agents\VitusDanceVirus.java	2	2022.05.15.	Vitus Tánc vírus osztálya.
equipments\Axe.java	2	2022.05.15.	Balta osztálya.
equipments\Bag.java	2	2022.05.15.	Hátizsák osztálya.
equipments\Equipment.java	3	2022.05.15.	Védőfelszerelés absztrakt űsosztály.
equipments\Glove.java	2	2022.05.15.	Kesztyű osztálya.
equipments\ProtectiveCloak.java	2	2022.05.15.	Védőköpeny osztálya.

equipments\UsableEquipment.java	1	2022.05.15.	Használható felszerelések interfésze.
main\Collectable.java	1	2022.05.15.	Gyűjthető interfész.
main\Constants.java	1	2022.05.15.	Konstansok osztálya.
main\Controller.java	9	2022.05.15.	Vezérlő osztálya.
main\Effect.java	2	2022.05.15.	Hatások interfésze.
main\GeneticCode.java	4	2022.05.15.	Genetikai kódok osztálya.
main\GraphicsView.java	32	2022.05.15.	A grafikus megjelenítő osztálya.
main\Inventory.java	8	2022.05.15.	Felszerelés osztálya.
main>Main.java	1	2022.05.15.	A fő osztály.
main\Map.java	11	2022.05.15.	Térkép osztálya.
main\Resource.java	4	2022.05.15.	Alapanyagok osztálya.
main\ResourceType.java	1	2022.05.15.	Alapanyagtípusok enumerációja.
main\SRandom.java	1	2022.05.15.	Saját Random osztály.
main\Virologist.java	11	2022.05.15.	Virológus osztálya.
observables\IObservable.java	1	2022.05.15.	A megjeleníthető objektumok interfésze.
observables\ObservableAgent.java	1	2022.05.15.	A megjeleníthető ágens osztálya.
observables\ObservableEquipment.java	1	2022.05.15.	A megjeleníthető felszerelés osztálya.
observables\ObservableGeneticCode.java	1	2022.05.15.	A megjeleníthető genetikai kód osztálya.
observables\ObservableResource.java	1	2022.05.15.	A megjeleníthető anyag osztálya.
observables\ObservableTileName.java	1	2022.05.15.	A megjeleníthető mező nevének osztálya.
observables\ObservableTileType.java	1	2022.05.15.	A megjeleníthető mező típusának osztálya.
observables\ObservableVirologistActions.java	1	2022.05.15.	A megjeleníthető virológus akcióinak

			számának osztálya.
observables\ObservableVirologistName.java	1	2022.05.15.	A megjeleníthető virológus nevének osztálya.
observables\ObservableVirologistPicture.java	2	2022.05.15.	A megjeleníthető virológus képének osztálya.
tiles\EmptyTile.java	1	2022.05.15.	Üres mező osztálya.
tiles\InfectedLaboratory.java	2	2022.05.15.	Fertőzött laboratórium osztálya.
tiles\Laboratory.java	1	2022.05.15.	Labor mező osztálya.
tiles\Safehouse.java	1	2022.05.15.	Óvóhely mező osztálya.
tiles\Tile.java	5	2022.05.15.	Mező absztrakt ősosztály.
tiles\Warehouse.java	1	2022.05.15.	Raktár mező osztálya.
/compiled/compile.bat	1	2022.05.15.	Fordítást végző parancsok.
start.bat	1	2022.05.13.	Futtatást végző parancs.

10.1.2 Fordítás és telepítés

A compiled/compile.bat fájlon dupla kattintással létrejön a virologist-game.jar fájl a gyökérkönyvtárban.

A .bat fájl futtatásakor előfordulhat, hogy a Windows Defender szól, hogy veszélyes lehet a fájl futtatása. Ilyenkor kattintsunk a More info-ra, majd a a Run anyway gombra. A .bat fájl az alább olvasható két java parancsot hajtja végre, amelyek nem károsak a számítógép számára.

Alternatív fordítási útmutató:

Az alábbi két parancsot kell egymás után kiadni a parancssori terminálban. A parancsok kiadása előtt el kell navigálni a projekt compiled mappájába a cd parancccsal. (A fájlkezelőben az aktuális mappát jelölő eszköztárba a cmd szót begépelve, automatikusan úgy nyílik meg a paracssori terminál.). A futtatható .bat fájl pontosan ezeket a parancsokat futtatja.

```
javac -d ./src/agents/*.java ./src/equipments/*.java ./src/main/*.java ./src/tiles/*.java ./src/observables/*.java
jar cfe ..virologist-game.jar main/Main agents/*.class equipments/*.class main/*.class tiles/*.class observables/*.class
```

10.1.3 Futtatás

Ha a gyökérkönyvtárban elkészült a virologist-game.jar fájl, akkor a start.bat fájl futtatásával lehet elindítani a játékot.

A .bat fájl futtatásakor előfordulhat, hogy a Windows Defender szól, hogy veszélyes lehet a fájl futtatása. Ilyenkor kattintsunk a More info-ra, majd a a Run anyway

gombra. A .bat fájl az alább olvasható java parancsot hajtja végre, amely nem káros a számítógép számára.

Alternatív futtatási útmutató:

Az alábbi parancsot kell kiadni a parancssori terminálban. A parancs kiadása előtt el kell navigálni a projekt gyökérmapjába a cd parancssal. (A fájlkezelőben az aktuális mappát jelőlő eszköztárba a cmd szót begépelve, automatikusan úgy nyílik meg a paracssori terminál.). A futtatható .bat fájl pontosan ezt a parancsot futtatja.

java -jar virologist-game.jar

10.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Domonkos Ádám	CWGYWC	20
Fekete Sámuel	GJ8J3A	20
Jurásek Jónás	PH4QFK	20
Kászonyi Zsombor	DCE2Q1	20
Száraz Dániel	GT5X34	20

10.3 Napló

Kezdet	Időtartam	Résznevők	Leírás
2022.05.04	1 óra	Fekete Domonkos Jurásek Kászonyi Száraz	Értekezlet. Döntés: feladatok szétosztása
2022.05.06.	2 óra	Fekete	Egyes Observable osztályok megírása
2022.05.06.	2 óra	Jurásek	Observable osztályok írása.
2022.05.06	3 óra	Száraz	Observable osztályok írása, GUI menu megírása
2022.05.07.	3 óra	Domonkos	Játékkelület elrendezése, elkészítése
2022.05.07	2 óra	Kászonyi	Graphics View függvények alapvető megírása
2022.05.08.	2 óra	Fekete	GUI javítása, tesztelése
2022.05.08	2 óra	Domonkos	Többiek kódjának ellenőrzése
2022.05.08.	2 óra	Jurásek	GUI tesztelése, Menü javítása
2022.05.08	2 óra	Kászonyi	Ablak kicsinosítása, elemek elrendezése
2022.05.13.	1 óra	Fekete	Fordítás és futtatás előkészítése, dokumentálás.
2022.05.13.	1 óra	Jurásek	Átfogó tesztelés
2022.05.13	3 óra	Kászonyi	Tesztelések, bugfixek
2022.05.13	1 óra	Domonkos	Súgó megírása
2022.05.13	2 óra	Száraz	Code cleanup, GUI javítás, tesztelés

14. Egyesített dokumentáció

52 - service_unavailable

Konzulens:
Micskei Zoltán

Csapattagok

Fekete Sámuel	GJ8J3A	feketesamu@gmail.com
Száraz Dániel	GT5X34	szdani2012@gmail.com
Jurásek Jónás	PH4QFK	jurasekjonas12@gmail.com
Domonkos Ádám	CWGYWC	domonkosadam01@gmail.com
Kászonyi Zsombor Marcell	DCE2Q1	kaszonyirelli@gmail.com

2022.02.23.

11. 2 Összefoglalás

11.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Domonkos Ádám	80
Fekete Sámuel	76
Jurásek Jónás	77
Kászonyi Zsombor	74

Száraz Dániel	77
Összesen	384

- A feltöltött programok forrássorainak száma**

Fázis	KódSOROK száMA
Szkeleton	1930
Prototípus	1734
Grafikus változat	2261
Összesen	5925

11.2 • Projekt összegzés

11.2.1 Mit tanultak a projektből konkrétan és általában?

Diagram egy ‘m’ (nem mindenki tanulta meg). Ezen kívül még feladatok egyenlő elosztását, csapatmunkát, verziókezelés használatát.

11.2.2 Mi volt a legnehezebb és a legkönnyebb?

A legnehezebb a tervezés és a működés kitalálása és dokumentálása volt, a legkönnyebb a kódolás a sok tervezés után.

11.2.3 Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?

Nem.

11.2.4 Ha nem, akkor hol okozott ez nehézséget?

A tervezési fázisban többet kellet dolgozni, mint amennyi pontot értek a beadások.

11.2.5 Milyen változtatási javaslatuk van?

A feladatok nagyon repetitívnak tűnnek egy idő után, mintha többször ugyanazt kéne megesinálni egy kicsit jobban kifejtve.

11.2.6 Milyen feladatot ajánlanának a projektre?

Lovagok járják a világot, és egymással harcolnak. A lovak elfáradnak, aludniuk kell időnként. Ha egy ló teljesen elfáradt, akkor automatikusan elalszik, ekkor lovagja nem tud tovább menni, amíg a ló fel nem kel. A lovat a lovasa is le tudja fektetni aludni, ilyenkor kevesebb idő alatt kipiheni magát. A lovagok is pihenhetnek, ilyenkor A lovagok párbajozhatnak más lovagokkal, a párharc győztese véletlenszerűen kerül ki, a vesztes elveszíti élettereje egy részét. Az éppen pihenő lovagok is megtámadhatóak, ilyenkor mindenkiéppen veszít a pihenő lovag életerőt.

Különböző felszereléseket lehet szerezni. Van páncél, ami 30%-kal növeli az esélyed a viadal megnyerésére. Van nyereg, amivel a lovak nagyobb távolságot tudnak megtenni elfáradás előtt. Van sátor, amiben a lovagot nem lehet megtámadni. Akinek az összes élettereje elfogyott, az kiesik a játékból. A játék nyertese az utolsóként maradó lovag.

11.2.7 Egyéb kritika és javaslat

Lehessen más grafikus könyvtárat használni Swing helyett (pl. JavaFX...)