

Pamacsok

Szoftvertechnológia házi feladat

Bogár Dániel

I07XOI

1 A FELADAT LEÍRÁSA

A Szaturnusz légkörében különböző fajtájú pamacsok lebegnek egymással összekapcsolódva, kolóniát alkotva. Ha két szomszédos pamacs egyforma fajtájú, akkor néha egy új pamacsot állítanak elő, amelynek fajtája teljesen véletlenszerű, és nem feltétlenül egyezik meg egyik szülő fajtájával sem. Az új pamacsnak kezdetben csak a szülei a szomszédjai, a szülők szomszédossága pedig megszűnik. A pamacsoknak bármennyi szomszédja lehet.

A pamacsokon különböző élőlények élnek, amelyek a pamacsokon tudnak mozogni. Vannak atkák, amelyek kiszívják a pamacsok erejét. Ha egy pamacs ereje elfogy, a pamacs elpusztul, a pamacs szomszédjai összekapcsolódnak, az addig a pamacson álló élőlények a szomszéd pamacsokra kerülnek. Vannak hangyák, amelyek meg tudják növelni egy pamacs erejét, de ha egy pamacs túltöltődik, a pamacs osztódik, a két új pamacs egymás szomszédja lesz, az eredeti szomszédokat pedig egymás közt szétosztva megöröklik. Vannak katicák, amelyek a velük egy pamacson álló atkákat és hangyákat tudják megenni.

A pamacsokon általában akármennyi élőlény állhat. Van olyan fajta pamacs, amelyik spontán osztódik, ha a rajta lévő élőlények száma meghaladja az 5-öt. Van olyan pamacs fajta, amelyik a rálépő lényeket megmérgezi. Ettől az atka elpusztul, a hangya elveszíti a pamacserő-növelő képességét. Van olyan pamacs, ami véletlenszerűen a rajta álló lényeket megrázza. Ettől az atka 5 pamacson kell átlépjen ahhoz, hogy újra képes legyen erőt szívni. A hangyát a rázás nem zavarja. A katicák a méregre immunisak, a rázástól viszont megbénulnak, amíg újabb rázás nem éri őket.

A játékosok feladata a katicák irányítása. A játék célja, hogy minél nagyobb pamacskolónia jöjjön létre.

2 FUNKCIONÁLIS KÖVETELMÉNYEK

2.1 ELSŐDLEGES KÖVETELMÉNYEK

Azonosító	Leírás	Use-case
G01	A Szaturnusz légkörében különböző fajtájú pamacsok lebegnek egymással összekapcsolódva, kolóniát alkotva.	View Athmosphere
G02	Szomszédos, ugyanolyan típusú pamacsok, új pamacsának előállítása.	Control Mumps Manage Athmosphere
G03	A pamacsokon különböző élőlények élnek, amelyek a pamacsokon tudnak mozogni.	Control Mumps Control LadyBug Control Ants Control Ticks
G04	Vannak atkák, amelyek elszívják a pamacs erejét.	Control Ticks Control Mumps
G05	Ha egy pamacs ereje elfogy, a pamacs elpusztul, a pamacs szomszédjai összekapcsolódnak, az addig a pamacson álló élőlények a szomszéd pamacsokra kerülnek.	Control Mumps
G06	Vannak hangyák, amelyek meg tudják növelni egy pamacs erejét	Control Mumps

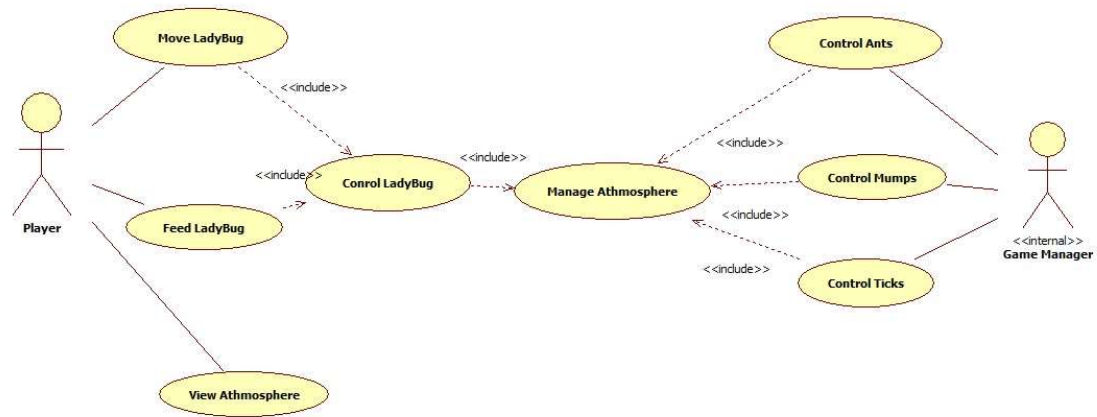
		Control Ants
G07	Ha egy pamacs túltöltődik, a pamacs osztódik, a két új pamacs egymás szomszédja lesz, az eredeti szomszédokat pedig egymás közt szétosztva megöröklük.	Control Mumps
G08	Vannak katicák, amelyek a velük egy pamacson álló atkákat és hangyákat tudják megenni.	Control LadyBug Move LadyBug Feed LadyBug Control Ants Control Ticks
G09	A pamacsokon általában akármennyi élőlény állhat.	Control Mumps Control Ants Control Ticks Control LadyBug
G10	Van olyan fajta pamacs, amelyik spontán osztódik, ha a rajta lévő élőlények száma meghaladja az 5-öt	Control Mumps Manage Athmosphere
G11	Van olyan pamacs fajta, amelyik a rálépő lényeket megmérgezi. Ettől az atka elpusztul, a hangya elveszíti a pamacserő-növelő képességét. A katicák a méregre immunisak.	Control Mumps Control Ticks Control Ants Control LadyBug
G12	Van olyan pamacs, ami véletlenszerűen a rajta álló lényeket megrázza. Ettől az atka 5 pamacson kell átlépjen ahhoz, hogy újra képes legyen erőt szívni. A hangyát a rázás nem zavarja. A katicák a rázásra megbénulnak, amíg újabb rázás nem éri őket.	Control Mumps Control Ticks Control LadyBug Control Ants
G13	A játékosok feladata a katicák irányítása	Move LadyBug Feed LadyBug
G14	A játék célja, hogy minél nagyobb pamacskolónia jöjjön létre.	View Athmosphere

2.2 TOVÁBBI KÖVETELMÉNYEK

Azonosító	Leírás	Use-case
G15	Az atmoszférában lebegő pamacsok egy gráfot alkotnak	Manage Athmosphere

3 USE-CASE-EK

3.1 USE-CASE DIAGRAM



3.2 USE-CASE LEÍRÁSOK

Cím	Move LadyBug
Leírás	A játékos mozgatja a katicát pamacsról, pamacsra
Aktorok	Player
Főforgatókönyv	1. A játékos megad egy pamacsot, amire a katica mozogni fog

Cím	Feed LadyBug
Leírás	A játékos meg tud etetni egy élőlényt (Hangya vagy Atka) a katicával
Aktorok	Player
Főforgatókönyv	1. A játékos megad egy élőlényt (Hangya vagy Atka), amit a katica meg fog enni

Cím	View Atmosphere
Leírás	A játékos megtekinti az atmoszférát
Aktorok	Player
Főforgatókönyv	1. A rendszer kirajzolja az atmoszféra aktuális állapotát 2. A játékos megtekinti az atmoszféra aktuális állapotát

Cím	Control LadyBug
Leírás	A rendszer a katicát teszi különböző effektek alá.
Aktorok	Game Manager
Főforgatókönyv	1. Ha egy katicát megrázza egy pamacs, megbénítja. 2. Ha egy katicát megmérgezik, nem történik semmi, mert immúnis rá.

Cím	Control Ants
Leírás	A rendszer irányítja a hangyákat
Aktorok	Game Manager
Főforgatókönyv	1. A rendszer megmondja a hangyának, hogy merre mozogjon, és ő ezt megteszi. 2. A rendszer annak a pamacsnak az életerejét tudja növelni, amelyiken áll. 3. Egy hangya egy pamacs hatása alá kerül.
Alternatív forgatókönyv	3.A. Ha egy hangyát egy megmérgezik, megvonják az életerő növelő képességétől. 3.B. Ha egy hangyát megrázzák, akkor nem történik semmi.

Cím	Control Mumps
Leírás	A rendszer számontartja, és kezeli a pamacsokat, a rajta álló élőlényekkel
Aktorok	Game Manager
Főforgatókönyv	<ol style="list-style-type: none"> 1. Ha egy Pamacs életereje lecsökken 0 alá, akkor a Pamacs megszűnik, a szomszédok összekapcsolódnak és a rajta lévő élőlények szétoszlanak. 2. Ha egy Pamacs életereje a maximális életereje fölé megy, akkor a pamacs osztódik. 3. Ha egy Pamacs és a szomszédja ugyanolyan típusúak, akkor véletlenszerűen tudnak egy új pamacsot generálni. 4. A pamacs képes használni a képességét, amennyiben van, és használja ezt a kezelt élőlényeken.
Alternatív forgatók	<ol style="list-style-type: none"> 4.A. A pamacs osztódó típusú 4.A.1. Ha a pamacson 5nél több élőlény van, akkor osztódik. 4.B. A pamacs mérgező típusú 4.B.1. Megmérgezi az összes rajta álló élőlényt. 4.C. A pamacs rázó típusú 4.C.1. Véletlenszerűen megrázza az összes rajta álló lényt.

Cím	Control Ticks
Leírás	A rendszer irányítja, és kezeli az atkákat.
Aktorok	Game Manager
Főforgatókönyv	<ol style="list-style-type: none"> 1. A rendszer megmondja az atkának, hogy merre mozogjon, és ő ezt megteszi 2. Az atka képes sebezni azt a pamacsot, amelyiken éppen van. 3. Egy atka a pamacs hatása alá kerül.
Alternatív forgatókönyv	<ol style="list-style-type: none"> 3.A. Ha egy atkát megmérgeznek, elpusztul. 3.B. Ha egy atkát megráznak, akkor 5 pamacson kell átlépjen, hogy újra használja a képességét.

Cím	Manage Athmosphere
Leírás	Az atmoszférában lévő elemek kezelése
Aktorok	Game Manager
Főforgatókönyv	<ol style="list-style-type: none"> 1. Az atmoszférában számontartja a pamacsokat, és kezeli azok képességeit.

4 STRUKTURÁLIS LEÍRÁS

4.1 AZ OSZTÁLYOK LEÍRÁSA

4.1.1 Mump

Felelősségek

Ez az osztály valósítsa meg a pamacsot, és a rajta élő élőlényeket kezeli. Ennek a leszármazottai valósítsák meg a sajátos viselkedést.

Attribútumok

#healt: int	A pamacs jelenlegi élet pontja számban
#maxHealth: int	A pamacs maximális élete
#creatures: Creature[]	A pamacsokon lakó élőlények egy tömbben
#name: String	A pamacs neve, ez később kelleni fog a szaporodásnál
#node: Node	Az a csomópont a gráfban, ahol a pamacs éppen van

Metódusok

+addCreature(Creature)	hozzáad egy élőlényt a creatures listához
+removeCreature(Creature)	eltávolítja a creature listából a paraméterként kapott lényt, amennyiben az a lista része
+removeAllAnt()	eltávolítja a creatures listából az összes hangyát
+removeAllTick()	eltávolítja a creatures listából az összes atkát
+action()	ez a függvény képviseli a pamacs képességét, a leszármazott osztályok implementálják
+damage(int)	a paraméterként kapott mennyiséget leveszi az életerőből
+addHealth(int)	a paraméterként kapott mennyiséget hozzáadja az életerőhöz

4.1.2 Tick

Felelősségek

Ez az osztály valósítsa meg az atka élőlényt, a Creature osztály leszármazottja

Attribútumok

-damage:int	az a mennyiség, amennyi sebzést tud okozni a pamacsnak
-stepsFromLastShake:int	számolja, hogy hány pamacson lépett át, azóta, amióta utoljára megrázták

Metódusok

+action()	a pamacsnak meghívja a damage(int) függvényét a damage tagváltozójával
+shake()	az adott élőlény megrázása, beállítja a helyes tagváltozóértékeket
+poision()	az adott élőlény megmérgezése, beállítja a helyes tagváltozóértékeket

4.1.3 Ant

Felelősségek

Ez az osztály valósítsa meg a hangya élőlényt, a Creature osztály leszármazottja.

Attribútumok

-healingPoints:int	ezzel az értékkel, mennyiséggel tudja növelni a hangya a pamacsot, amin rajta van
-healingEnabled: bool	logikai érték, amely megadja, hogy lehet ez adott hangyával életet növelni

Metódusok

+action()	meghívja a pamacsnak (annak, amin van) az addHealth(int) függvényét a healingPoints tagváltozóval
+shake()	az adott élőlény megrázása, beállítja a helyes tagváltozóértékeket
+poision()	az adott élőlény megmérgezése, beállítja a helyes tagváltozóértékeket

4.1.4 LadyBug

Felelősségek

Ez az osztály valósítsa meg a katica élőlényt, a Creature osztály leszármazottja.

Attribútumok

-shook: bool	logikai érték, amely megadja, hogy az adottkatica rázás hatása alatt van, vagy sem
--------------	--

Metódusok

+action()	az adott pamacson(amin van) lévő összes hangyát és atkát törli a pamacs creatures listájából (meghívja a removeAllAnt() és removeAllTick() függvényeket)
+shake()	az adott élőlény megrázása, beállítja a helyes tagváltozóértékeket
+poision()	az adott élőlény megmérgezése, beállítja a helyes tagváltozóértékeket

4.1.5 Creature

Felelősségek

A pamacson lévő élőlények őssosztálya

Attribútumok

#mump: Mump	Az a pamacs, amelyen ez az élőlény szerepel
-------------	---

Metódusok

#move(Node)	Ezt az élőlényt a paraméterként megadott csúcsra mozgatja
-------------	---

#action()	speciális tevékenység, ezt implementálják ez az osztály leszármazottjai
#shake()	az adott élőlény megrázása, beállítja a helyes tagváltozóértékeket
#poision()	az adott élőlény megmérgezése, beállítja a helyes tagváltozóértékeket

4.1.6 Splitter

Felelősségek

Ez az osztály valósítja meg azt a pamacsfajtát, amelyik spontán osztódik, a Mump osztály leszármazottja.

Attribútumok

-creatureNumber: int	az adott pamacson lévő élőlények száma
-name: String	az pamacs neve, a szaporodásnál jön elő

Metódusok

+action()	az osztódást valósítja meg
+addCreature(Creature)	felülírja az őosztály addCreature(Creature) függvényét, azzal a feltétellel, hogyha így több, mint 5 élőlény lenne rajta, akkor spontán osztódik

4.1.7 Poisonous

Felelősségek

Ez az osztály valósítja meg azt a pamacsfajtát, amelyik megmérgezi a rajtan állókat, a Mump osztály leszármazottja.

Attribútumok

-name: String	a pamacs neve, a szaporodásnál jön elő
---------------	--

Metódusok

+action()	megmérgezi a rajta álló élőlényeket
-----------	-------------------------------------

4.1.8 Shaker

Felelősségek

Ez az osztály valósítja meg azt a pamacsfajtát, amelyik megrázza a rajta állókat, a Mump osztály leszármazottja.

Attribútumok

-name:String	a pamacs neve, a szaporodásnál jön elő
--------------	--

Metódusok

+action()	megrázza a rajta álló élőlényeket
-----------	-----------------------------------

4.1.9 Node

Felelősségek

A pamacsokból készít egy gráf csúcsot

Attribútumok

-node:Mump	a pamacs, amely a gráf csúcsát jelképezi
-neighbours: Node[]	az adott pamacs szomszédjai
-saturn: Saturn	az a Saturn típusú változó, amiben ez a csúcs szerepel

Metódusok

+createChild(Node)	egy másik Node-al szaporít egy újabb pamacsot
+addNeighbours(Node[])	a paraméterként kapott csúcsokat hozzáadja a neighbours listához
+addNeighbour(Node)	a paraméterként kapott csúcsot hozzáadja a neighbours listához
+removeNeighbour(Node)	a paraméterként kapott csúcsot törli a neighbours listából
+removeNeighbours(Node[])	a paraméterként kapott csúcsokat törli a neighbours listából
+distributeCreatures(Node[])	a paraméterként kapott csúcsok között elosztja az ezen pamacson lévő élőlényeket
+split()	meghívja a pamacs split() függvényét
+getNeighbours(): Node[]	visszaadja azokat a csúcsokat, amivel ez a csúcs szomszédos
+getHalfNeighbours(): Node[]	visszaadja azokat a csúcsoknak a felét, amivel ez a csúcs szomszédos
+addCreatures(Creature[])	meghívja a pamacs addCreature(Creature) függvényét a paraméterként kapott élőlények összes elemére
+getCreatures(): Creature[]	meghívja, és visszaadja a pamacs getCreatures() függvényét
+getHalfCreatures(): Creature[]	meghívja, és visszaadja a pamacs getCreatures() függvényeként visszaadott élőlények felét
+removeCreature(Creature)	meghívja a paraméterként kapott élőlénnel a pamacs removeCreature(Creature) függvényét
+removeCreatures(Creature[])	meghívja a paraméterként kapott élőlények összes elemével a pamacs removeCreature(Creature) függvényét
+addCreature(Creature)	meghívja a pamacs addCreature(Creature) függvényét a paraméterként kapott értékkel
+getSaturn(): Saturn	visszaadja azt a Szaturnuszt, amin éppen ez a csúcs van
+getNode()	visszaadja azt a pamacsot, ami a csúcsot jelképezi
+getMumpName(): String	meghívja és visszaadja a node getName() függvényét

4.1.10 Saturn

Felelősségek

A szaturnusz légkörében keringő pamacsokat tárolja (a gráf csúcsait)

Attribútumok

-mumps: Node[]	a gráf csúcsai
----------------	----------------

Metódusok

+removeNode(Node)	a paraméterként kapott Node-t eltávolítja a listából
addNode(Node)	hozzáadja a paraméterként kapott Node-t a listához

4.1.11 Mover

Felelősségek

Egy interfész, ami minden olyan dolgot reprezentál, amely időben lépni tud.

Metódusok

+move(Node)	a mozgás metódus, implementálni kell
-------------	--------------------------------------

4.1.12 Timer

Felelősségek

Periodikus időzítés a játékban.

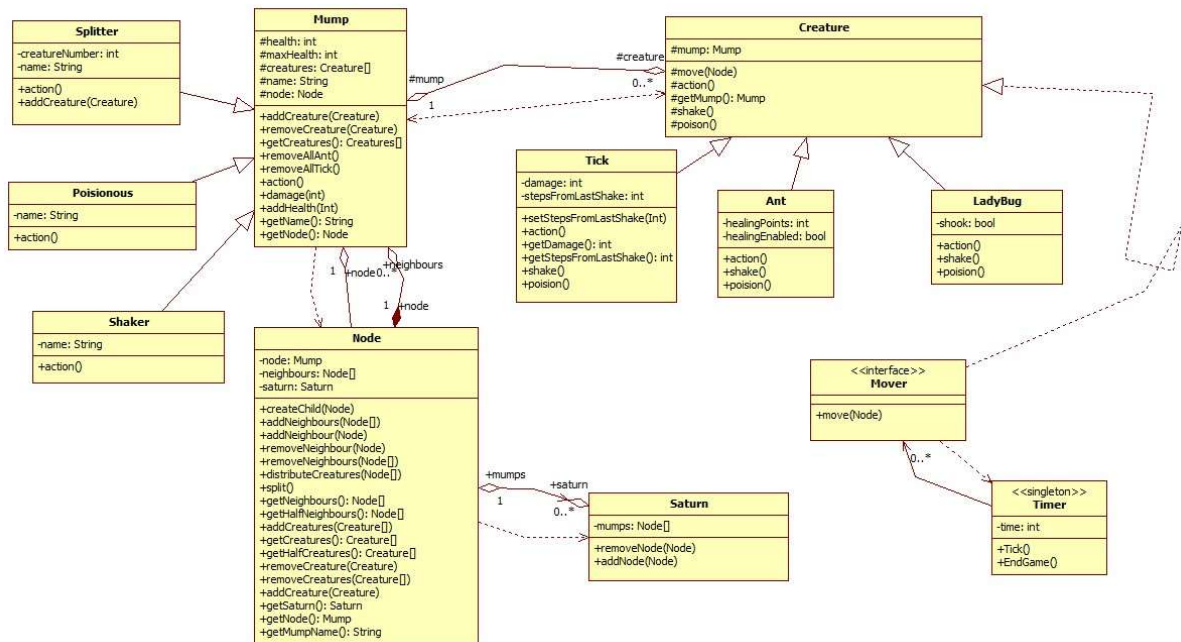
Attribútumok

-steppable: Mover[]	A léphető dolgok
---------------------	------------------

Metódusok

+Tick()	Minden léptethető dolog léptetése
+addSteppable(Mover)	Új léptethető dolog hozzáadása
+removeSteppable(Mover)	Léptethető dolog törlése

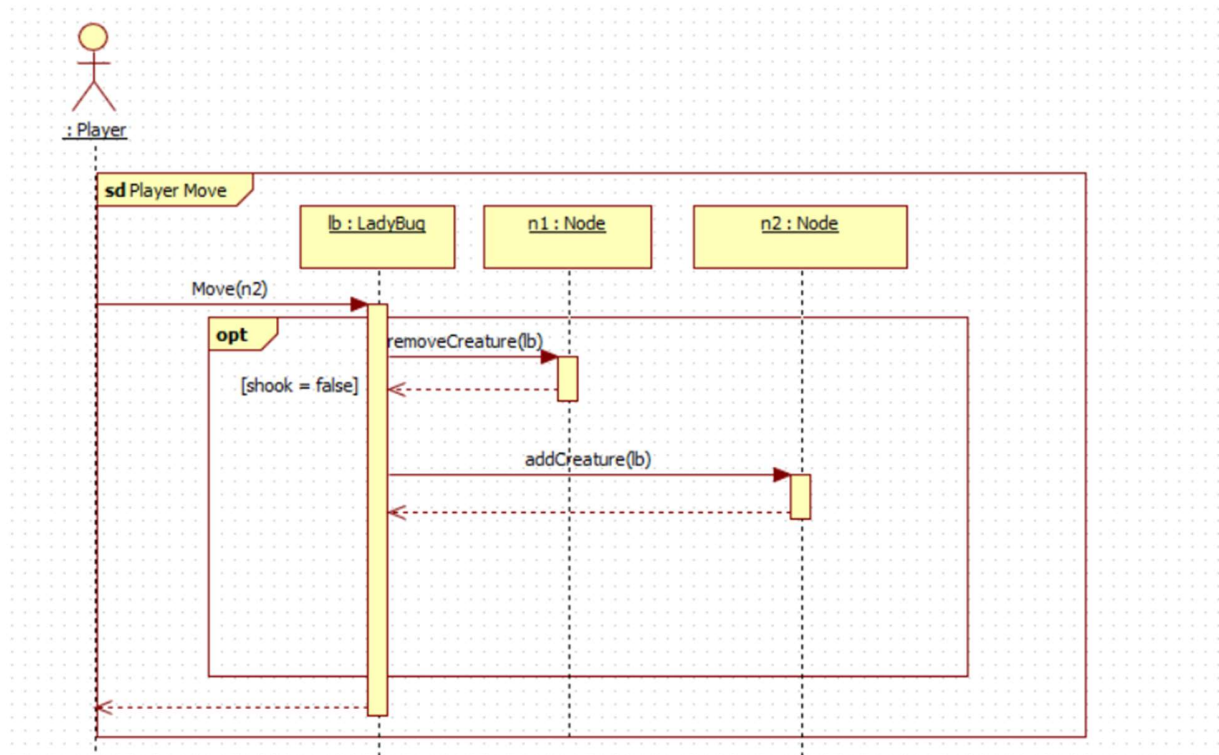
4.2SZTÁLYDIAGRAM



5 VISELKEDÉS LEÍRÁSA

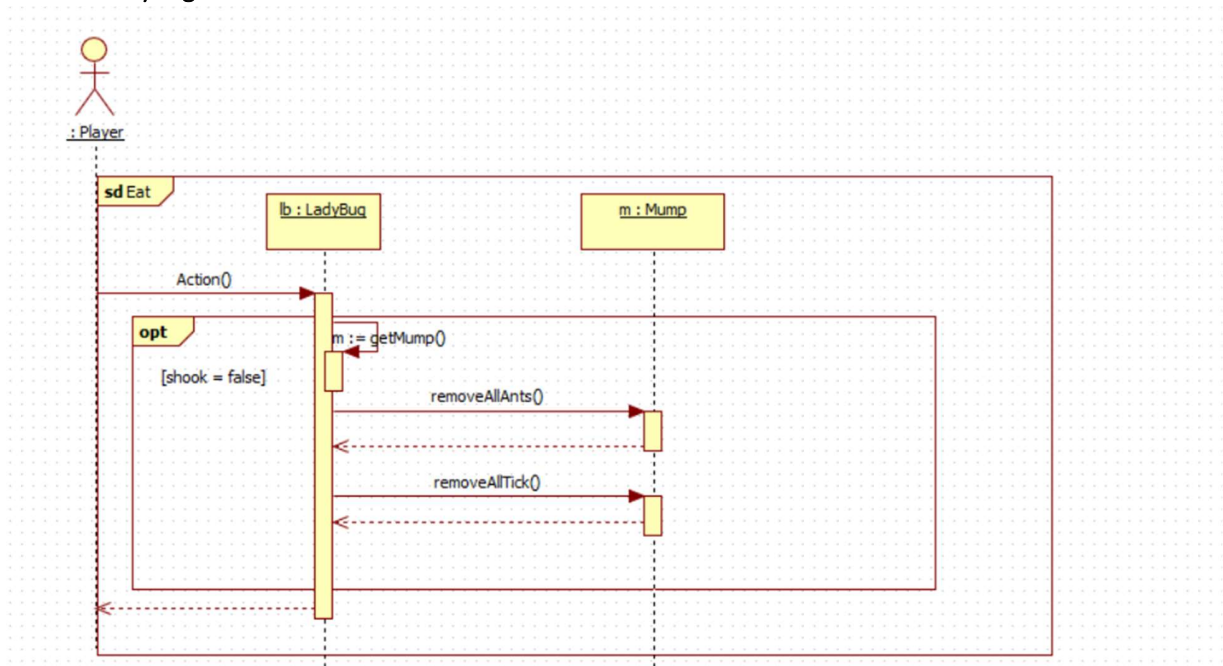
5.1 SZEKVENCIA DIAGRAMOK

5.1.1 Player Move

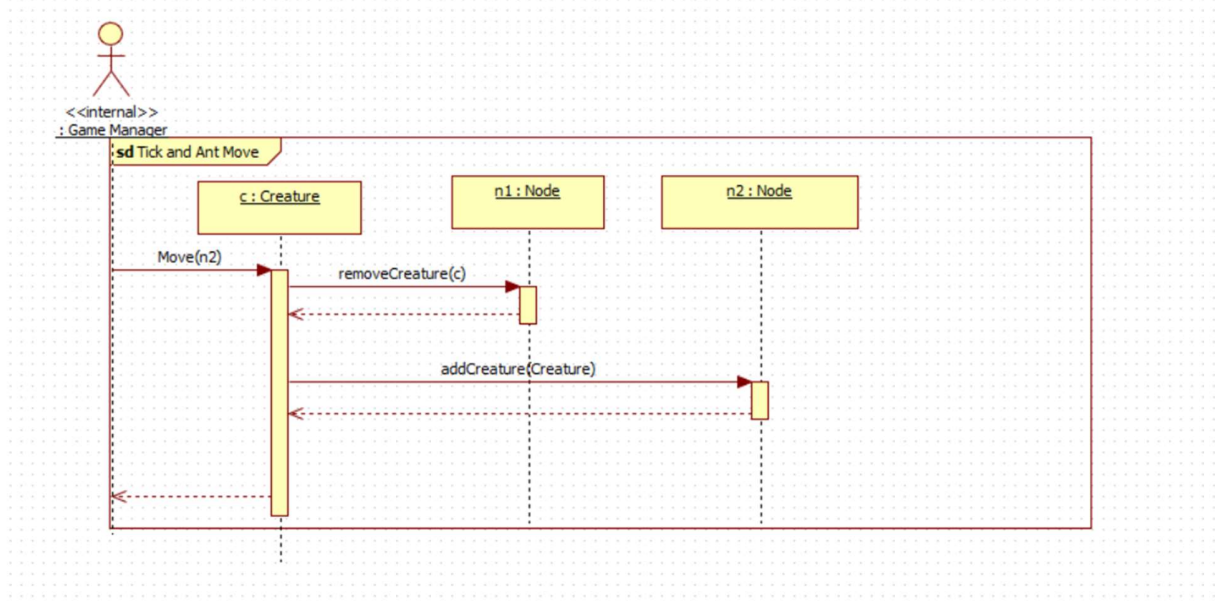


n1 – A node, amin a LadyBug jelenleg van

5.1.2 LadyBugEat

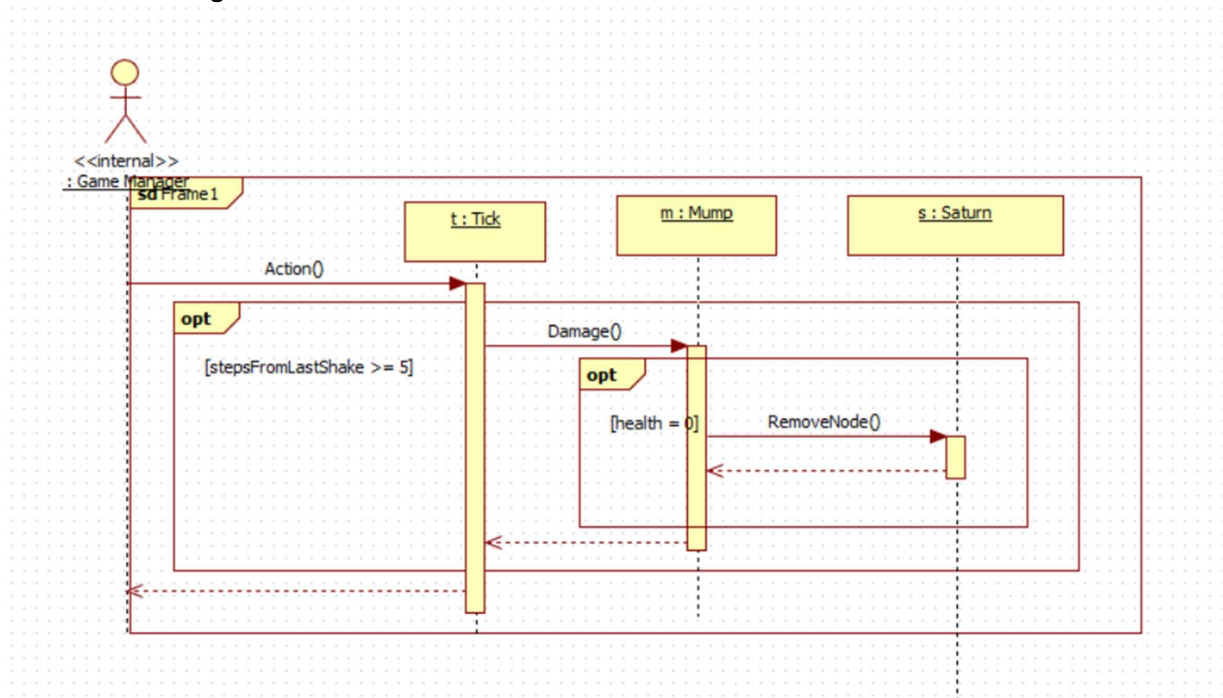


5.1.3 TickAndAntMove

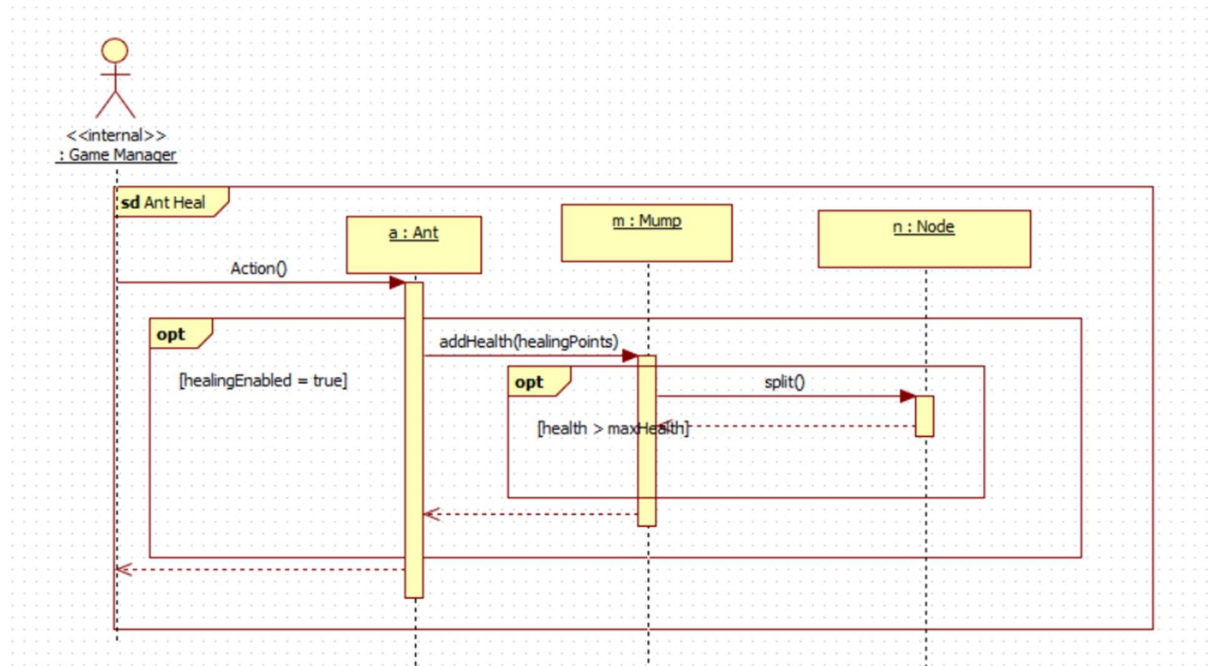


n1 – A node, amin jelenleg az élőlény van

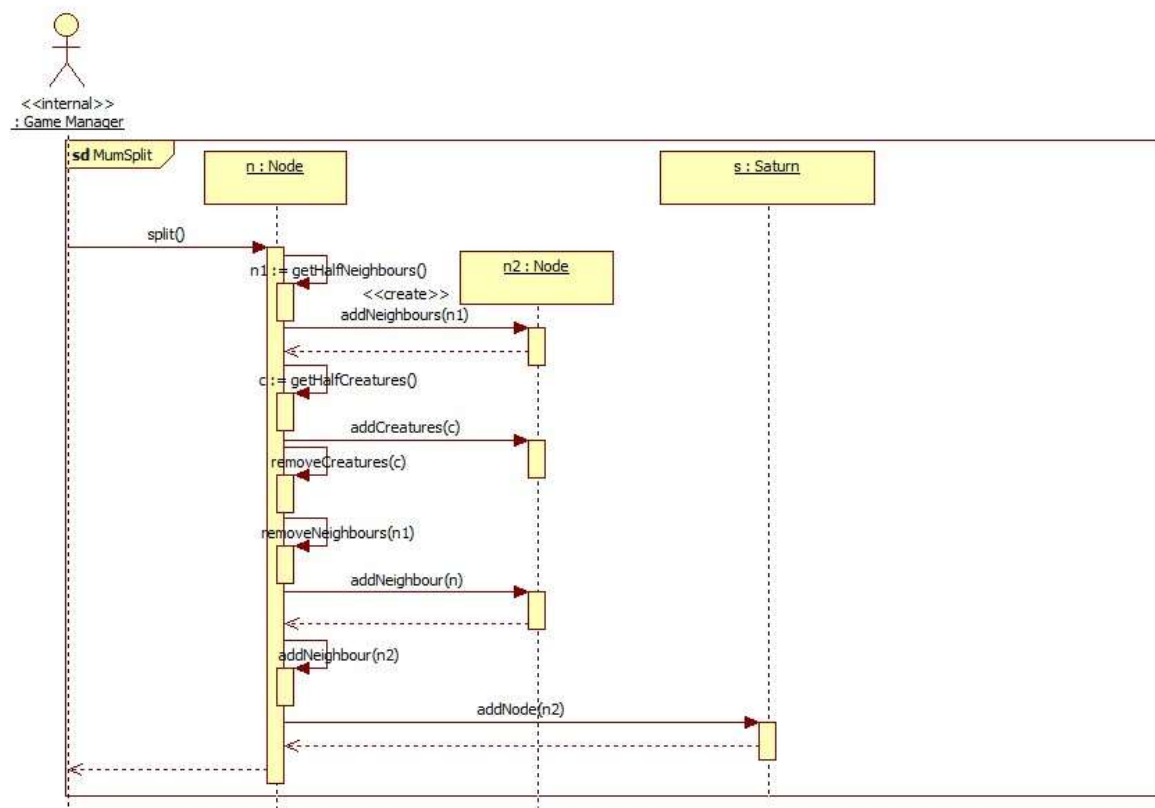
5.1.4 TickDamage



5.1.5 AntHeal



5.1.6 MumpSplit

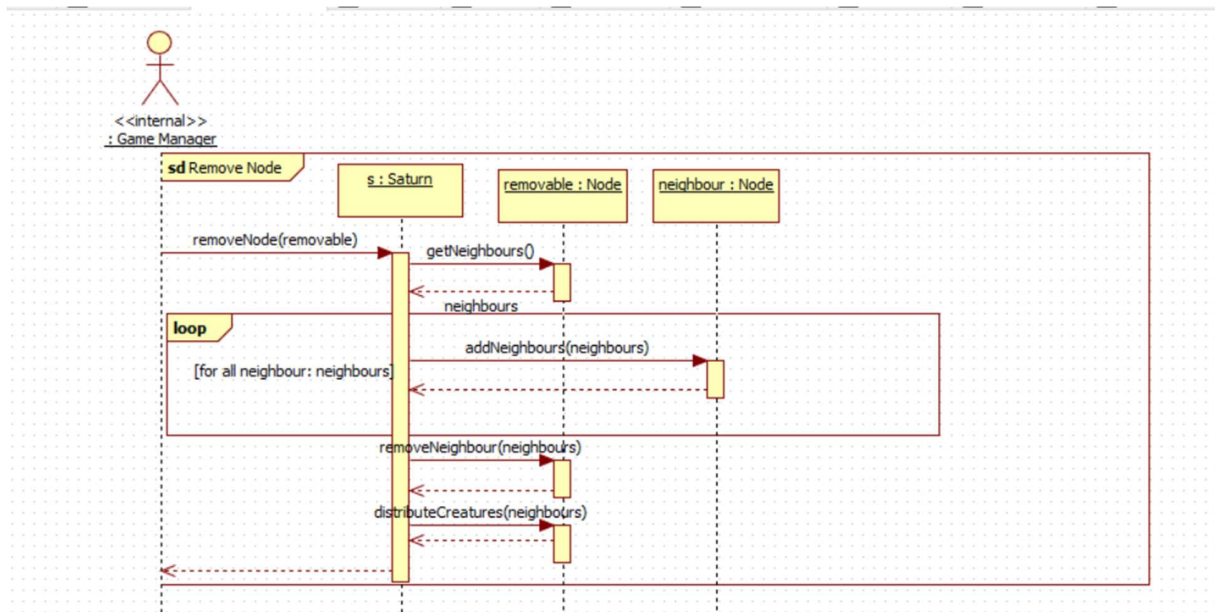


n: az eredeti Node

n2: az a node, amit létrehoz az osztódással

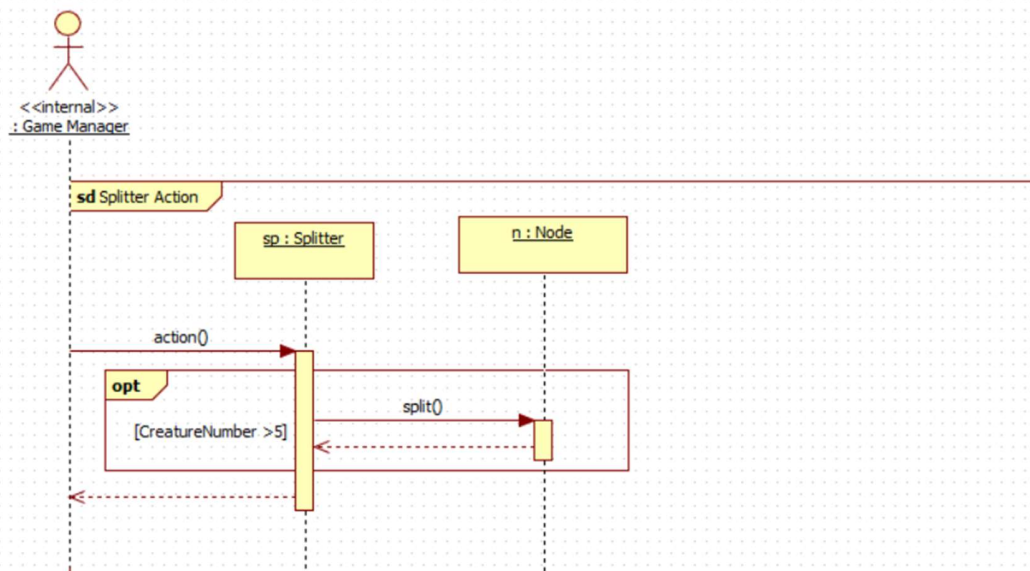
Egyszerűen elosztja a szomszédokat és az élőlényeket rajta. Először lekérdezi ezeket, majd hozzáadja az újhoz, végül eltávolítja saját magáról.

5.1.7 removeNode

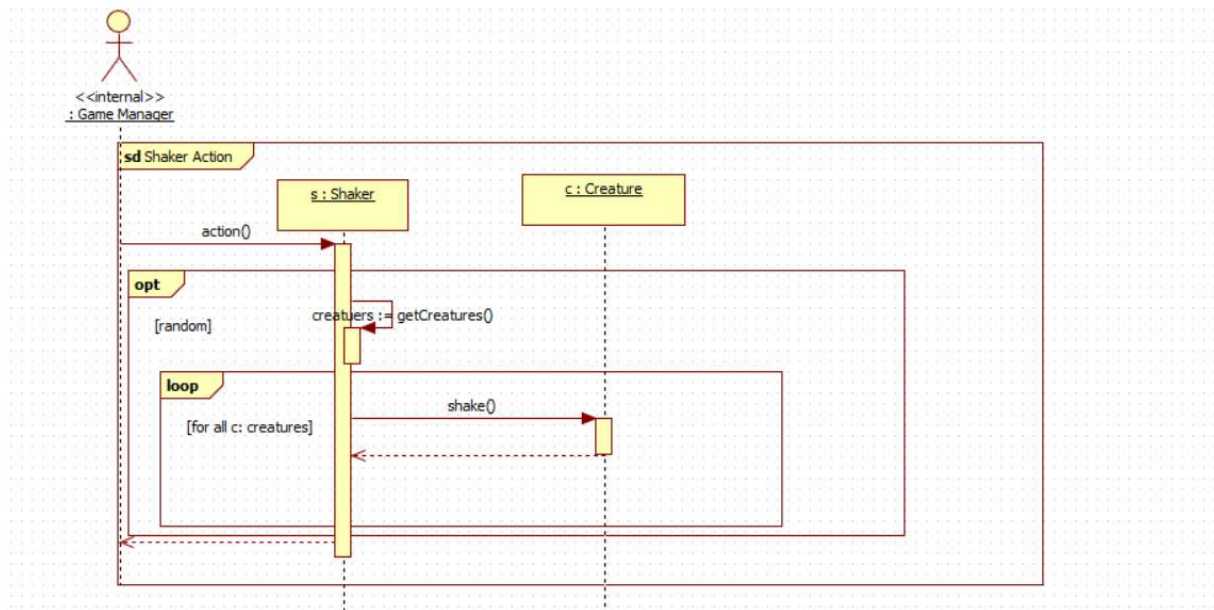


A removable node lekéri az összes szomszédját, majd ezeket a csúcsokat összeköti az összes többi szomszéddal. Végül az eredeti removable csúcsnak az összes összeköttetését törli, és szétosztja az élőlényeket a szomszédok között.

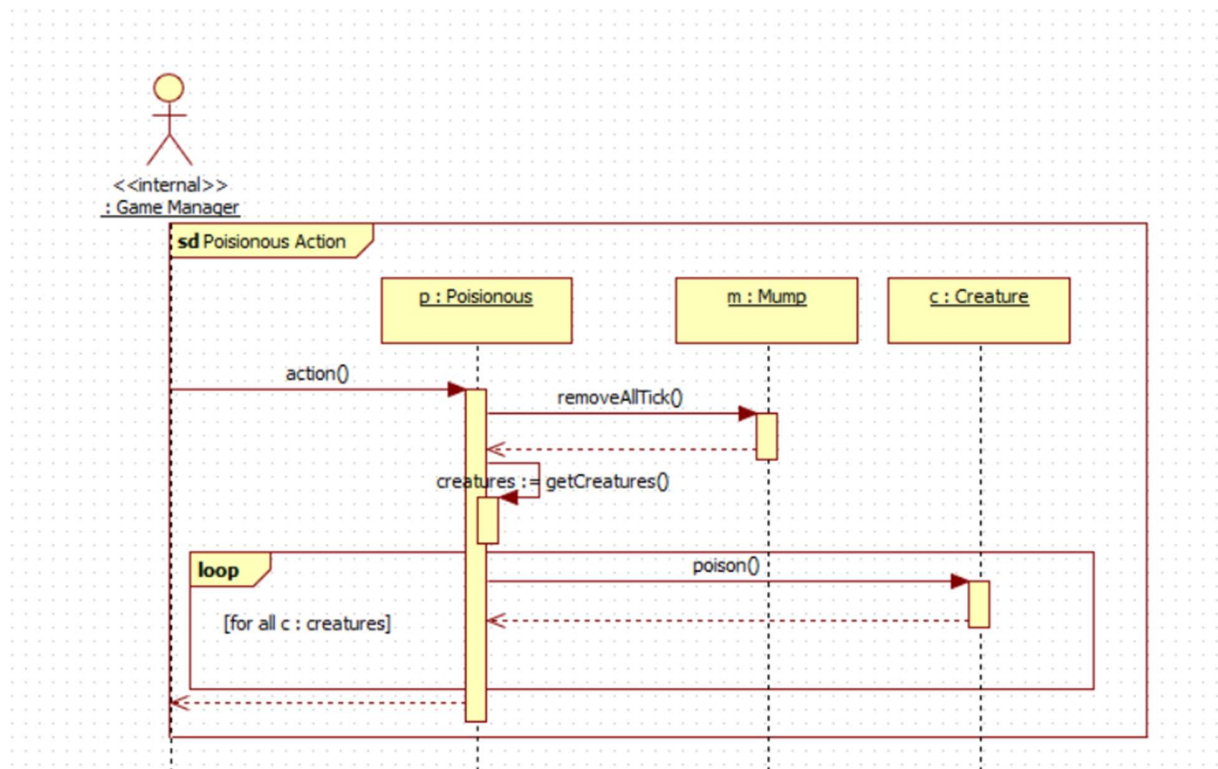
5.1.8 SplitterAction



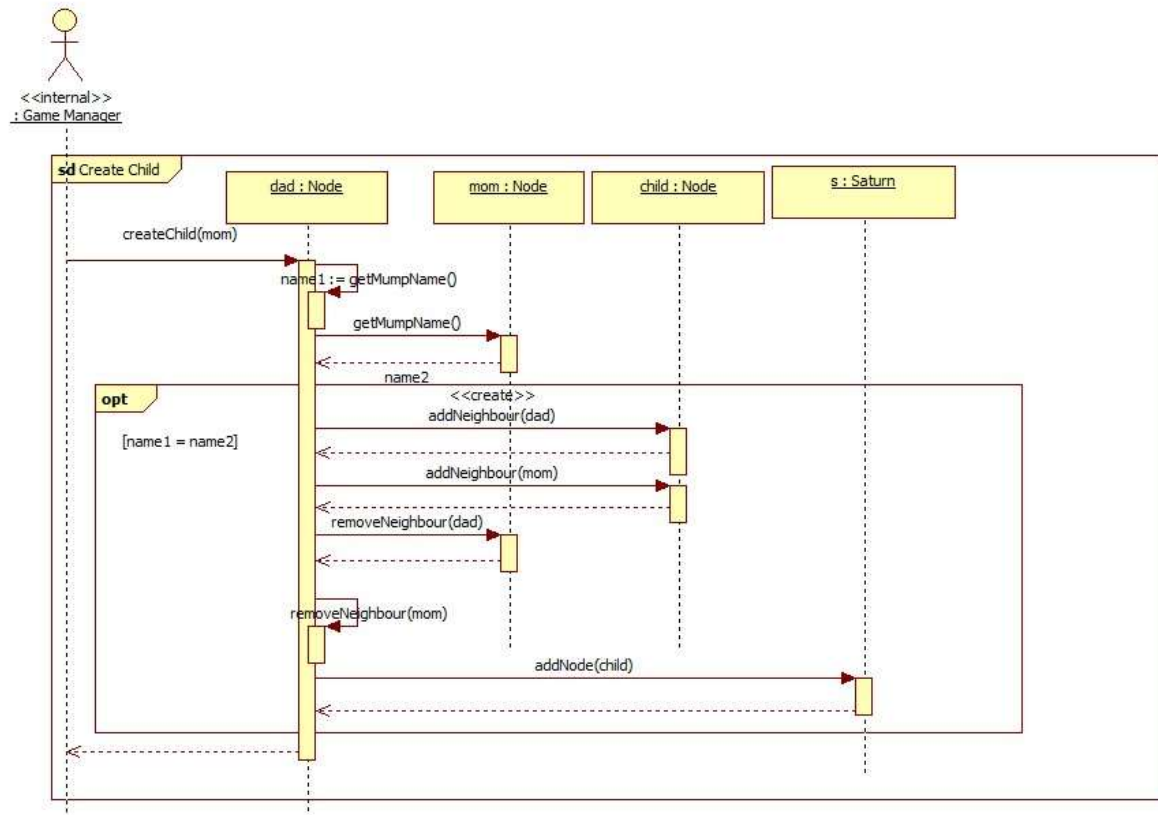
5.1.9 ShakerAction



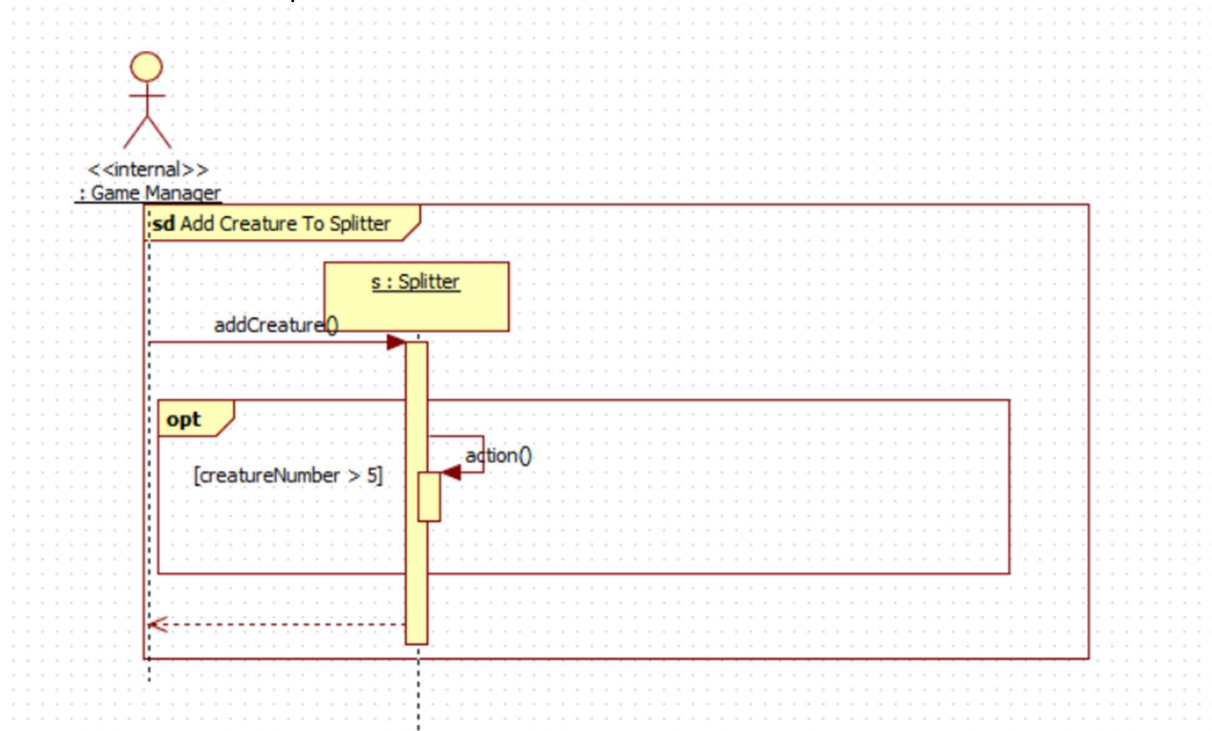
5.1.10 PoisionousAction



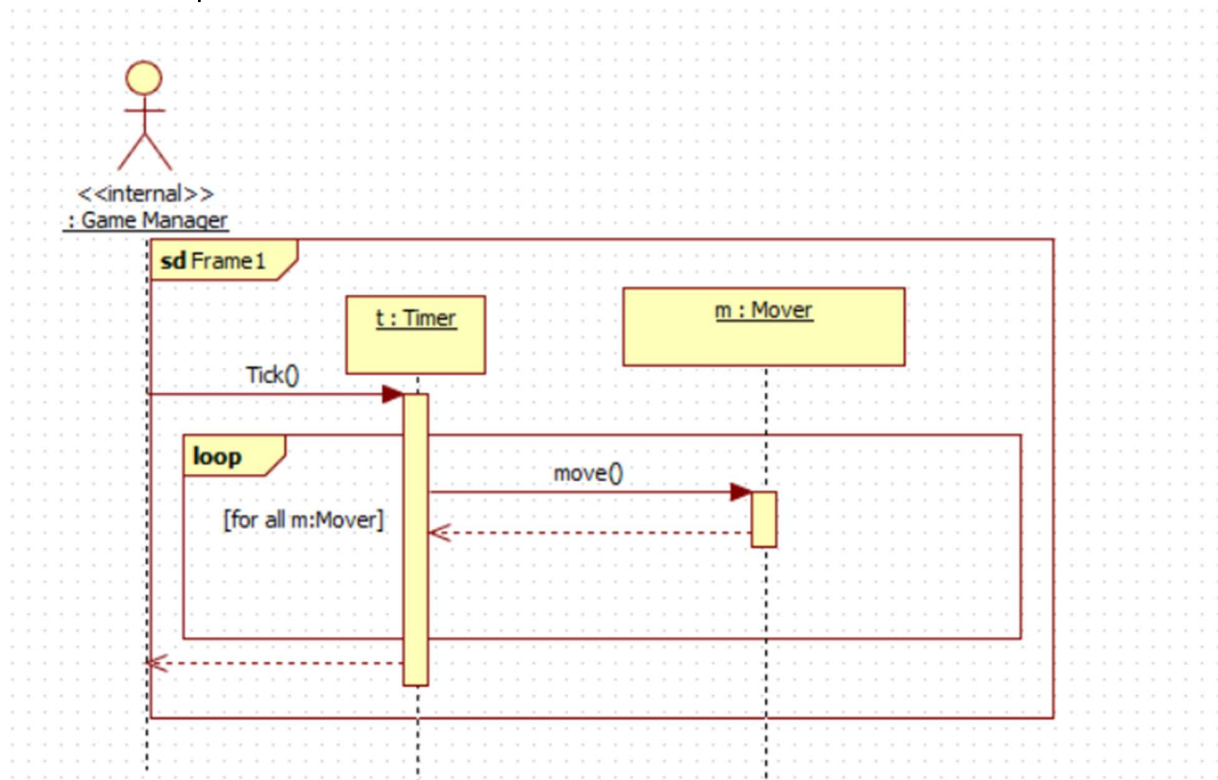
5.1.11 CreateChild



5.1.12 AddCreatureToSplitter

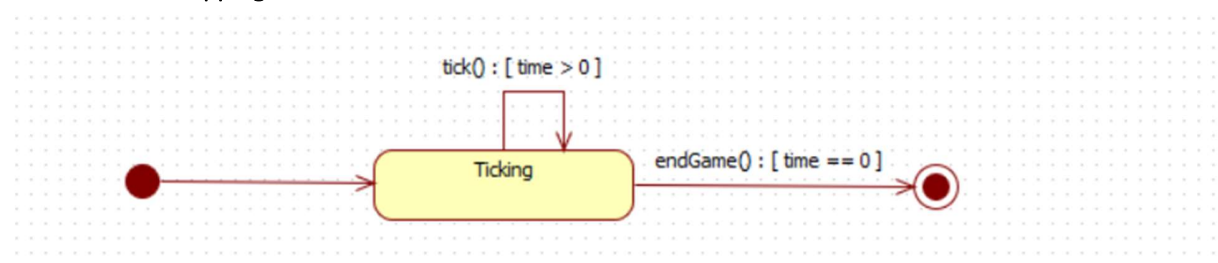


5.1.13 TimeStep



5.2 ÁLLAPOTGÉP DIAGRAMOK

5.2.1 Time Stepping



6 NAPLÓ

Kezdet	Időtartam	Elvégzett munka	Hivatkozások
2022.12.03: 14:15	2	Feladat megértése, elképzelése és osztálydiagram elkezdése	1, 4
2022.12.04: 18:30	3	osztálydiagram, majd use-case diagram elkészítése	2, 3, 4
2022.12.06 10:00	2	osztálydiagram és use-case diagram dokumentálása	3, 4
2022.12.09 13:45	7	szekvenciadiagramok készítése, osztálydiagram módosítása	4, 5
2022.12.11 19:00	5	szekvenciadiagramok készítése, osztálydiagram javítása, state-chart diagram készítése	4, 5
2022.12.12 19:00	2	Maradék feladatok naplózása	4, 5, 6

Összes elvégzett munka: 21

Modellező eszköz: WhiteStar UML

Egyéb eszközök: Microsoft Word