

## 12. Teljes Dokumentum

48 – smooth\_set

Konzulens:  
Ludmány Balázs

### Csapattagok

Kis Máté Levente	H0OTLX	ghzman029@gmail.com
Kobál Sándor	C4W3BP	alexkobal90@gmail.com
Kiss Előd	U479JN	k.elod98@gmail.com
Kecskés Péter	XZ9I7F	kecskespeti@gmail.com
Martonosi Máté	QOT4DC	mate.martonosi99@gmail.com

2019.05.16.

## 2. Követelmény, projekt, funkcionalitás

### 2.1 Bevezetés

#### 2.1.1 Cél

A jelen dokumentum elsődleges célja a „Pandapláza” feladat leírásában az egyes nem tisztán specifikált elemek egyértelműsítése, illetve a legalapvetőbb implementációs kérdések taglalása.

#### 2.1.2 Szakterület

A projekt végeredményeként egy kisebb méretű, egyszemélyes játékszoftvert szeretnénk. Szoftverünk elsősorban a PC felhasználókat célozza meg.

#### 2.1.3 Definíciók, rövidítések

IIT - Irányítástechnikai és Informatikai tanszék

JavaSE - Java, Standard Edition

#### 2.1.4 Hivatkozások

A dokumentum készítése közben a Szoftvertechnológia tárgyban tanultakra illetve a feladat eredeti szövegére hivatkozunk.

<https://www.iit.bme.hu/targyak/BMEVILAB02/feladat>

#### 2.1.5 Összefoglalás

A lentiekben egy átfogó áttekintést nézünk a projektről, melynek során leírjuk a játék részletes működését, figyelve arra, hogy ezt bármely, a dokumentumot olvasó személy megértse. Ezen kívül belekezdünk a fejlesztői oldalról is lényeges dolgokba, illetve kitekintést teszünk a jövőbeli tervekre, lépésekre.

## 2.2 Áttekintés

### 2.2.1 Általános áttekintés

Egyéb fontosabb alrendszer a Java futtatókörnyezeten kívül nincs. A szoftver telepített könyvtárában hordozza a futáshoz szükséges file-jait, mint hang és kép elemek.

### 2.2.2 Funkciók

Az állatkertből kiszöktek a pandák, és a szomszédos bevásárlóközpont egyik emeletén ütöttek tanyát, ahol véletlenszerűen kószálnak. A feladat a pandákat összeszedő orángutánok irányítása és ezáltal minél több panda összegyűjtése.

Az emeleten a pandák és az orángutánok csempéken mozognak, a csempék élei csatlakoznak egymáshoz, egy élhez csak egy másik csempe éle csatlakozhat. A csempék változó élszámú szabályos sokszögek. Egy csempén csak egy állat állhat. Vannak gyengébb csempék, amelyekre, ha húsznál többször lépnek rá, eltörnek. Ha egy állat gyenge csempére lép és a csempe eltörik, akkor lezuhan és meghal (a mögötte sorba fűzött pandák elengedik egymást). Ezután a csempe újra megjelenik hasonló tulajdonságokkal. Egy orángután lezuhanása nagy pont veszteséggel, egy panda lezuhanása kisebb pont veszteséggel jár.

Az emeleten vannak tárgyak is, a tárgyakat nem lehet elmozdítani. A tárgyak szintén csempén helyezkednek el, de az ilyen csempék biztosan nem gyengék. Tárgyak közül van szekrény, fotel, játékgép, csokiautomata. A játékgép véletlen időközönként csilingel, a csokiautomata véletlen időközönként sípol. A hangokat csak a tárgy csempéjéhez csatlakozó csempéken lehet hallani. A szekrénybe be lehet lépni. Ha egy állat belép egy szekrénybe, akkor egy másik, az emeleten elhelyezett, szekrényből jön ki. A szekrények össze vannak kötve egymással, tehát egy szekrénybe belépve egyértelműen egy másik szekrényből jön ki, de a szekrények összeköttetései nincsenek jelölve, a játékos magától fedezi fel.

Az orángutánoknak egymás után körkörösén adhatjuk meg, hogy merre lépjenek tovább. Egy körben egy orángután pontosan egy csempényt léphet. Ha a játékos lépett minden orángutánnal, utána a szabadon lévő pandák is lépnek egy véletlenszerűen kiválasztott irányba. Ha egy orángután egy összeláncolt panda sorába ütközik, akkor befűzi az új pandát, de az új panda mögött befűzött pandák leszakadnak. Az orángutánok egy módon tudnak pandát fogni. Az orángután nekimegy a pandának, a panda befejezi kaotikus mozgását. Az orángután és a panda helyet cserélnek. A panda ezután követni fogja az orángutánt. A pandák láncba szerveződnek olyan módon, hogy az orángután nekimegy egy új pandának, helyet cserélnek, az új pandát befűzi maga mögé a láncba, az új panda pedig befűzi maga mögé az eddigi pandákból alkotott láncot. Ha az orángután nem tud továbblépni (pl. tárgyak állják el az útját) akkor egy helyben marad, a mögötte befűzött pandák pedig szétszóródnak.

A pandáknak különböző tulajdonságaik vannak, de egy panda csak egy tulajdonsággal rendelkezhet. Van, amelyik egy játékgéppel szomszédos csempén állva, csilingelés következtében megijed, és elengedi az előtte lévő állat mancsát. Ennek következtében a megijedt panda, és a mögötte állók is elengedik egymást, és folytatják természetes kaotikus mozgásukat az emeleten. Van amelyik a szomszéd csempén lévő csokiautomata sípolását követően ugrik egyet, ezáltal, ha éppen egy gyenge csempén áll, akkor a csempe élettartama csökken eggyel. Ebben az esetben, ha a gyenge csempe már csak egy lépést bír ki, és az ugrás után eltörik, akkor a panda lezuhan, a mögötte lévő pandák elengedik egymást és folytatják természetes kaotikus mozgásukat. Van, amelyik egy szomszéd csempén elhelyezkedő fotel mellett elhaladva elfárad, és elengedi az előtte haladó panda mancsát és beül a fotelbe. Eközben a mögötte haladó pandák is elengedik egymást, és folytatják kaotikus mozgásukat. Ha egy ilyen tulajdonsággal rendelkező panda beül egy fotelbe, akkor ezt követően a fotelben marad véletlen mennyiségű ideig, amíg a panda a fotelben ül, az orángután nem mehet neki, és nem fűzheti be a láncba. Miután lejárt az idő a panda feláll a fotelből, és folytatja kaotikus mozgását az emeleten. Ha a panda nem tud felállni (minden szomszédos csempe foglalt), akkor vár még egy lépés erejéig és a fotelben marad. Minden fotelnek megvan a saját várakozási ideje, egy játék alatt egy fotelben töltött várakozási idő nem változik.

Az emeleten két speciális tárgy van, a bejárat és a kijárat. A játék elején a bejáratból lépésenként egy-egy orángután megy ki, addig, amíg az emeleten a mennyiségük nem lesz megfelelő. Az orángután feladata, hogy a pandákat a kijáratához vezesse. Az orángután eléri a kijáratot és eltűnik az emeletről. A pandák lánc folytatja az útját a kijárat felé. Ha egy panda áthaladt a kijáraton, eltűnik az emeletről. Minden, a kijáratához eljutott, pandáért a játékos pontot kap. Miközben a pandák haladnak a kijáraton keresztül, ugyanúgy történhet váratlan esemény (pl. egy panda leül a kanapéra és emiatt elengedi a sorban előtte álló mancsát). Ha a pandák lánc végig ment a kijáraton, a pályára lép egy orángután a bejáraton keresztül. A játéknak vége van, ha az emeleten nincs több panda, vagy az összes orángután lezuhant.

A pályán nem lehet zsákutca, a bejárat és a kijárat csempéjén kívül. A zsákutca egy olyan csempét jelöl, amelyhez csak egy másik csempe csatlakozik. A bejárat és a kijárat nem számít zsákutcának, mivel nem csempe.

### 2.2.3 Felhasználók

A játékot egyszerre egy játékos játszhatja, a játék nem követel meg semmi előképzettséget felhasználótól.

### 2.2.4 Korlátozások

A játékot PC-n lehet játszani, a játékhoz egér szükséges. A játéktól elvárjuk a stabil működést, nem állhat le vagy fagyhat le indokolatlanul. A játék csakis a követelmények által meghatározott módon működhet.

### 2.2.5 Feltételezések, kapcsolatok

<https://www.iit.bme.hu/targyak/BMEVIIAB02/feladat> - A feladat szövegének megismeréséhez.

## 2.3 Követelmények

### 2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	UseCase	Komment
F100	A játék körökre osztott módon halad	kiértékelés, bemutatás	alapvető	Saját	Control Game	
F110	A játékos után, a játék logika jön a lépésben.	kiértékelés, bemutatás	alapvető	Saját	Control Game	
F120	A játék addig halad, amíg van a pályán élő panda.	kiértékelés, bemutatás	alapvető	Saját	Control Game	
F200	A pályán csak orángutánok és pandák mozognak.	kiértékelés, bemutatás	alapvető	IIT	Control Game	
F210	A pályán található minden pandát játék logika irányít.	kiértékelés, bemutatás	alapvető	IIT	Control Pandas	
F211	A pályán található orángutánokat a játékos irányítja.	kiértékelés, bemutatás	alapvető	IIT	Move Orangutan	Maximum 3 játékos
F220	A pályán található minden állat és	kiértékelés, bemutatás	alapvető	IIT	View Game	

	tárgy valamilyen csempén helyezkedik el és csak ezeken közlekedhet.					
F300	A pályán minden csempének van szomszédja, ami szintén csempe.	kiértékelés, bemutatás	alapvető	IIT	View Game	
F310	Az állatok a tartalmazó csempének csak a szomszédjaira léphet.	kiértékelés, bemutatás	alapvető	IIT	Move Orangutan, Control Pandas	
F320	A csempén csak egy tárgy helyezkedhet el.	kiértékelés, bemutatás	alapvető	Saját	Control Game	
F321	A csempén egyszerre csak egy állat lehet.	kiértékelés, bemutatás	alapvető	Saját	Control Game	
F322	A csempéken véletlen helyezkedik el szekrény, fotel, játékgép, csokiautomata, kijárat és bejárat.	kiértékelés, bemutatás	fontos	IIT	Control Game	
F330	A csempe lehet törékeny vagy nem törékeny.	kiértékelés, bemutatás	fontos	IIT	Control Game	
F331	A törékeny csempének élete van, mielőtt eltörik.	kiértékelés, bemutatás	fontos	Saját	Control Game	20 élete van pontosan
F332	Az eltört csempére lépve meghalnak az állatok.	kiértékelés, bemutatás	alapvető	IIT	Move Orangutan, Control Pandas	Eltűnnek a pályáról
F400	A szekrények a pályán össze vannak kötve egymással.	kiértékelés, bemutatás	fontos	IIT	Control Game	
F410	A szekrényeken csak állatok utazhatnak az abba való belépéssel.	kiértékelés, bemutatás	alapvető	IIT	Move Orangutan, Control Pandas	
F420	A játékgép csilingel	kiértékelés, bemutatás	fontos	IIT	Control	

	időközönként.				Game	
F421	A csokiautomata sápol véletlen időközönként.	kiértékelés, bemutatás	fontos	IIT	Control Game	
F422	Minden hanghatás csak az azt körbevevő mezőkön és azon a mezőn hallható.	kiértékelés, bemutatás	alapvető	IIT	Control Game	
F430	A bejáraton csak az orángután jön be akkor, ha kiment korábban a kijáraton.	kiértékelés, bemutatás	alapvető	Saját	Move Orangutan, Control Game	
F431	A kijáraton kilépve az orángután megvárja a kezét fogó pandákat, hogy beérkezzenek, addig pedig nincs a pályán.	kiértékelés, bemutatás	alapvető	Saját	Control Pandas	
F432	A bejáraton az orángután az utolsó bevezetett panda után megy csak ki.	kiértékelés, bemutatás	fontos	IIT	Move Orangutan, Control Game	
F500	Az orángután minden megérintett pandát maga mögé sorakoztat fel, ami nem ül fotelben.	kiértékelés, bemutatás	alapvető	IIT	Move Orangutan	
F501	Az orángután, ha egy már meglévő sorban álló pandát fog meg, akkor azt a pandát kiragadja a sorába.	kiértékelés, bemutatás	fontos	Saját	Move Orangutan	
F502	Az orángután másik orángutánt nem tud megfogni, nincs interakció közöttük	kiértékelés, bemutatás	alapvető	Saját	Control Game	Panda pandával sem lép interakcióba alapvetően.

F510	A panda, ha megérinti egy orángután, akkor azt követi és elengedi az eddig fogott bármi állatot/állatokat.	kiértékelés, bemutatás	alapvető	IIT	Control Pandas	
F511	A panda követi a kezébe adott másik állatot, ha nincs olyan tulajdonsága, hogy megzavarodik, vagy nem engedi el a kezét az előtte lévő.	kiértékelés, bemutatás	alapvető	IIT	Control Pandas	
F520	Ha a panda kezét elengedi az előtte lévő, akkor ő is elengedi a mögötte lévőét.	kiértékelés, bemutatás	alapvető	IIT	Control Pandas	
F530	A pandák csak egy tulajdonsággal rendelkezhetnek.	kiértékelés, bemutatás	fontos	IIT	Control Pandas	
F531	Panda tulajdonság lehet, hogy megzavarodik a mellette lévő játékgép csilingelése miatt elengedi a kezeket.	kiértékelés, bemutatás	fontos	IIT	Control Pandas	
F532	Panda tulajdonság lehet, hogy megzavarodik a mellette lévő csokiautomata sípolásától és csökkenti a helyén lévő törékeny csempének életét eggyel.	kiértékelés, bemutatás	fontos	IIT	Control Pandas	
F533	Panda tulajdonság lehet, hogy megzavarodik a mellette lévő foteltől és beleül.	kiértékelés, bemutatás	fontos	IIT	Control Pandas	

F540	Minden fotelnél megvan, hogy mennyi időt tölthet benne a panda.	kiértékelés, bemutatás	fontos	Saját	Control Game	Ez az idő random fotelenként
F600	Minden célba vitt panda növeli a játékos pontszámát.	kiértékelés, bemutatás	fontos	Saját	Control Game	Egy ponttal növeli
F610	Minden lezuhant panda kisebb pontlevonással jár	kiértékelés, bemutatás	fontos	Saját	Control Game	Egyet vonnak le
F620	A lezuhanó orángután nagy pont veszteséggel jár	kiértékelés, bemutatás	fontos	Saját	Control Game	

### 2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
P1	A szoftver JavaSE 1.8-ban fordítható.	kiértékelés	alapvető	Saját	
P2	A fejlesztés Windows környezetben történik minden fejlesztő esetében.	kiértékelés	alapvető	Saját	
P3	A forráskód online Github felületen található.	kiértékelés	alapvető	Saját	

### 2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
-----------	--------	------------	-----------	--------	---------



<i>D1</i>	A szoftver interface angol nyelvű.	<i>kiértékelés, bemutatás</i>	<i>alapvető</i>	<i>Saját</i>	
<i>D2</i>	A szoftver tartalmazza az esetleges kép és hanganyagot is, de ezt nem a futtatható állományba illesztve.	<i>kiértékelés</i>	<i>alapvető</i>	<i>Saját</i>	
<i>D3</i>	A szoftver futásához szükséges a Java futtatókörnyezet.	<i>kiértékelés</i>	<i>alapvető</i>	<i>Saját</i>	

### 2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
<i>O1</i>	A szoftver test-casekkel védett módon fejlesztett.	<i>kiértékelés</i>	<i>alapvető</i>	<i>Saját</i>	
<i>O2</i>	A szoftver 13. héten készen bemutatásra kerül felhasználói dokumentációval.	<i>kiértékelés, bemutatás</i>	<i>alapvető</i>	<i>Saját</i>	
<i>O3</i>	A szoftver mellett készül szkeleton proto és grafikus változat.	<i>kiértékelés, bemutatás</i>	<i>alapvető</i>	<i>Saját</i>	

## 2.4 Lényeges use-case-ek

Move Orangutan, View Game, Control Pandas, Control Game

### 2.4.1 Use-case leírások

<i>Use-case neve</i>	<i>Move Orangutan</i>
----------------------	-----------------------

<b>Rövid leírás</b>	<i>A játékos egy adott orángutánt irányít a pályán.</i>
<b>Aktorok</b>	<i>Player</i>
<b>Forgatókönyv</b>	<i>1. A pályán található orángutánokat a játékos irányítja</i>
<b>Alternatív forgatókönyv</b>	<i>1.A.1. Az orángután csak közvetlen szomszédságában lévő csempékre léphet.</i>
<b>Alternatív forgatókönyv</b>	<i>1.B.1. Ha az orángután olyan csempére lép, amelyen áll már egy panda, azt befűzi maga mögé egy láncba.</i>
<b>Alternatív forgatókönyv</b>	<i>1.B.1.A.1. Az orángután minden megérintett pandát maga mögé sorakoztat fel, ami nem ül fotelben</i>
<b>Alternatív forgatókönyv</b>	<i>1.B.1.A.2. Az orángután, ha egy már meglévő sorban álló pandát fog meg, akkor azt a pandát kiragadja a sorába.</i>
<b>Alternatív forgatókönyv</b>	<i>1.C.1. Ha az orángután eltört csempére lép, meghal.</i>
<b>Alternatív forgatókönyv</b>	<i>1.D.1. Ha az orángután szekrénybe lép, egy másik szekrényből lép ki a pályán.</i>
<b>Alternatív forgatókönyv</b>	<i>1.E.1. Ha az orángután kimegy a kijáraton, akkor a bejáratnál jelenik meg.</i>
<b>Alternatív forgatókönyv</b>	<i>1.E.1.A.1. Az orángután csak azután jelenik meg a bejáraton, miután a láncába tartozó utolsó panda is elhagyta a pályát.</i>

<b>Use-case neve</b>	<i>View Game</i>
<b>Rövid leírás</b>	<i>A játékos látja a pályát, rajta a szereplőket.</i>
<b>Aktorok</b>	<i>Player</i>
<b>Forgatókönyv</b>	<i>1. A rendszer kirajzolja a pályát és szereplőkkel 2. A játékos megtekinti a pályát és a szereplőket</i>

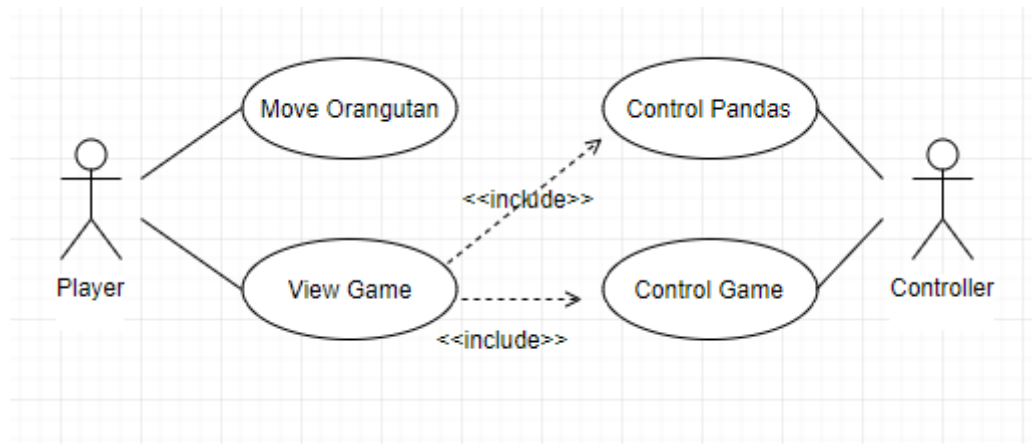
<b>Use-case neve</b>	<i>Control Pandas</i>
<b>Rövid leírás</b>	<i>A rendszer meghatározza a pályán lévő pandák viselkedését.</i>
<b>Aktorok</b>	<i>Controller</i>

<b>Forgatókönyv</b>	<i>1. A pályán található minden pandát játék logika irányít</i>
<b>Alternatív forgatókönyv</b>	<i>1.A.1. A panda csak közvetlen szomszédságában lévő csempékre léphet</i>
<b>Alternatív forgatókönyv</b>	<i>1.B.1. Ha a panda eltört csempére lép meghal</i>
<b>Alternatív forgatókönyv</b>	<i>1.C.1. Ha a panda szekrénybe lép, egy másik szekrényből lép ki a pályán.</i>
<b>Alternatív forgatókönyv</b>	<i>1.D.1. A panda, ha megérinti egy orángután, akkor azt követi és elengedi az eddig fogott bármi állatot/állatokat.</i>
<b>Alternatív forgatókönyv</b>	<i>1.D.1.A.1. A panda követi a kezébe adott másik állatot, ha nincs olyan tulajdonsága, hogy megzavarodik, vagy nem engedi el a kezét az előtte lévő.</i>
<b>Alternatív forgatókönyv</b>	<i>1.D.1.A.2. Ha a panda kezét elengedi az előtte lévő, akkor ő is elengedi a mögötte lévőét.</i>
<b>Alternatív forgatókönyv</b>	<i>1.E.1. Ha a pandát felvezető orángután kimegy a kijáraton, onnantól kezdve a panda csak a kijárat felé lép, amíg ki nem ér rajta.</i>
<b>Alternatív forgatókönyv</b>	<i>1.F.1. A pandák csak egy tulajdonsággal rendelkezhetnek.</i>
<b>Alternatív forgatókönyv</b>	<i>1.F.1.A.1. Panda tulajdonság lehet, hogy megzavarodik a mellette lévő játékgép csilingelése miatt elengedi a kezeket.</i>
<b>Alternatív forgatókönyv</b>	<i>1.F.1.A.2. Panda tulajdonság lehet, hogy megzavarodik a mellette lévő csokiautomata sípolásától és csökkenti a helyén lévő törékeny csempének életét egygel.</i>
<b>Alternatív forgatókönyv</b>	<i>1.F.1.A.3. Panda tulajdonság lehet, hogy megzavarodik a mellette lévő foteltől és beleül.</i>

<b>Use-case neve</b>	<i>Control Game</i>
<b>Rövid leírás</b>	<i>A játék lefolyását vezérli.</i>
<b>Aktorok</b>	<i>Controller</i>
<b>Forgatókönyv</b>	<i>1. A játék körökre osztott módon halad, egyszerre csak egy állat léphet</i>
<b>Alternatív forgatókönyv</b>	<i>1.A.1. A játékos után a játék logika jön a lépésben</i>

<i>Alternatív forgatókönyv</i>	<i>1.B.1. A pályán mozgó állat csak a panda és az orángután</i>
<i>Alternatív forgatókönyv</i>	<i>1.C.1. A játék addig halad, amíg minden állatfajból van még élő.</i>
<i>Alternatív forgatókönyv</i>	<i>1.D.1. Egy csempén csak egy tárgy vagy állat helyezkedhet el</i>
<i>Alternatív forgatókönyv</i>	<i>1.D.1.A.1. A csempéken véletlen helyezkedik el szekrény, fotel, játékgép, csokiautomata, kijárat és bejárat.</i>
<i>Alternatív forgatókönyv</i>	<i>1.D.1.A.2. A csempe lehet törékeny vagy nem törékeny.</i>
<i>Alternatív forgatókönyv</i>	<i>1.D.1.A.2.A.1. A törékeny csempének élete van, mielőtt eltörik (20)</i>
<i>Alternatív forgatókönyv</i>	<i>1.E.1. A szekrények a pályán össze vannak kötve egymással</i>
<i>Alternatív forgatókönyv</i>	<i>1.F.1. A játékgép csilingel és a csokiautomata sípol véletlen időközönként</i>
<i>Alternatív forgatókönyv</i>	<i>1.F.1.A.1. Minden hanghatás csak az azt körülvevő mezőkön és azon a mezőn hallható</i>
<i>Alternatív forgatókönyv</i>	<i>1.G.1. Minden fotelnél megvan, hogy mennyi időt tölthet benne a panda</i>
<i>Alternatív forgatókönyv</i>	<i>1.H.1. Minden célba vitt panda növeli a játékos pontját</i>
<i>Alternatív forgatókönyv</i>	<i>1.I.1. Ha egy panda meghal, akkor pontlevonás történik</i>
<i>Alternatív forgatókönyv</i>	<i>1.J.1. Ha egy orángután meghal az nagy pontlevonással jár</i>
<i>Alternatív forgatókönyv</i>	<i>1.K.1. Az orángután másik orángutánt nem tud megfogni, nincs interakció közöttük</i>

## 2.4.2 Use-case diagram



## 2.5 Szótár

**állat** – panda vagy orángután

**állatkert** – Háttértörténet része, továbbiakban nem használjuk.

**bejárat** - Speciális tárgy. Itt jelennek meg az orángutánok egymás után a játék kezdetén, és innen jön ki az orángután, ha az általa vezetett pandákból álló lánc végighaladt a kijáraton.

**bevásárlóközpont** - Háttértörténet része, továbbiakban nem használjuk.

**csatlakozó csempe** – lásd: szomszédos csempe

**csempe** – A játék emeletei csempékből épülnek fel, amelyeken az állatok mozoghatnak illetve tárgyak helyezkedhetnek el. A csempék mind valamilyen szabályos sokszög alakúak. Egy csempén csak egyvalami állhat, kivételt a fotel jelenti, mert ezen ülhet legfeljebb egy panda.

**csempe élete** - lásd: csempe élettartama

**csempe élettartama** - Gyenge csempéknél van értelmezve. Egy gyenge csempe, amelyre még nem léptek vagy megjavult ennyi lépést/ugrást bír ki mielőtt (újra) eltörne. Egy nem gyenge csempe élettartama végtelen.

**csilingel** – Játékgép által keltett hanghatás, amelyet csak a játékgép csempéjének szomszédos csempéin elhelyezkedő pandák hallanak meg. Ha a panda rendelkezik olyan tulajdonsággal, hogy a csilingeléstől megijed, akkor ha láncban volt, elengedi a láncban előtte álló mancsát.

**csokiautomata** – Tárgy. Véletlen időközönként sípol.

**elengedi az utána álló mancsát** - Ha egy láncban egy állat elengedi a mögötte álló mancsát, akkor az így leváló tagban (=az eredeti láncban az állat mögött álló pandák) a pandák elengedik egymást és folytatják a kaotikus mozgásukat.

**elengedi az előtte álló mancsát** - Ha egy láncban ha egy panda elengedi az előtte álló mancsát, akkor az adott panda és az eredeti láncban mögötte lévő pandák is elengedik egymást és folytatják a kaotikus mozgásukat.

**elengedik egymást** - Egy lánc leváló tagja már nem alkot egységet így a benne lévő egyes pandák elengedik egymás mancsát és folytatják kaotikus mozgásukat.

**elfáradás** - Némely panda rendelkezik azzal a tulajdonsággal, hogy fotelt látva valamely szomszédos csempén elfárad. Ha ilyenkor valamely szomszédos fotel üres akkor beleül. Ha a panda esetlegesen láncban volt akkor elengedi az előtte álló mancsát.

**eltörik** - lásd: gyenge csempe

**eltört csempe** – lásd: gyenge csempe

**emelet** – A játék színtere.

**fotel** – Tárgy. Egyes pandák elfáradnak a szomszédságában.

**gyenge (gyengébb) csempe** – Olyan csempe, amelyre ha húsznál többször lépnek (ugrást is beleértve) eltörik. Ilyenkor a törést okozó, rálépő/ráugró állat lezuhan és meghal. Gyenge csempén biztos nincs tárgy. Az eltört csempe a kör végeztével megjavul és hasonló tulajdonságokkal bír, mint előtte.

**játékgép** – Tárgy. Véletlen időközönként csilingel.

**játékos** - Az orángutánok irányítója, célja minél több panda összegyűjtése és kijáraton kivitele.

**játék vége** – Akkor következik be, ha az emeleten nincsen panda vagy ha az összes orángután lezuhant.

**kaotikus mozgás** - A nem láncban lévő pandák mozgása kiszámíthatatlan, nem követ semmilyen rendszert (*persze a csempe által meghatározott szabályokat a pandák is betartják*).

**kijárat** - Speciális tárgy. Itt tudja kivinni az összegyűjtött pandáit az orángután.

**kör** - Egy kör során minden állat lép legfeljebb egyszer.

**lánc** - Egy orángutánból és az általa sorba befűzött pandákból áll. Veszthet méretéből, ha valamelyik állat lezuhan, ha egy panda leül vagy ha egy (nem feltétlenül különböző) orángután befűzi a saját láncába a lánc valamelyik pandáját.

**láncba befűz** - Ha egy orángután egy olyan csempére lép, ahol egy panda áll, akkor a pandával helyet cserél és megfogja a mancsát. Ha az orángutánnak már volt egy láncnyi összegyűjtött pandája akkor az a láncot a frissen megfogott panda másik mancsába adja. Ha a láncba fűzni kívánt panda már egy lánc tagja (akár ugyanazé) akkor az orángután befűzi mint új panda, de az új panda eredeti láncában a panda mögött állók elengedik egymást.

**leül** - lásd: elfáradás

**lezuhan** – Esemény, ami akkor következik be ha egy állat a lépésével/ugrásával maga alatt eltör egy csempét. Mindkét állat esetében a játékos pontot veszít. Orángután esetében valamivel többet. Ha minden panda vagy orángután lezuhan akkor a játéknak vége. Ha egy lánc tagja zuhan le, akkor a hozzá képest hátrébb lévő pandák (már ha van ilyen) leszakadnak. A lezuhanó állatok véglegesen kikerülnek a játékból.

**lépés** - Lépés során egy állat a csempéjéről egy szomszédos csempére lép.

**meghal** – lásd: lezuhan

**megijed** - lásd: csilingel

**megjavult csempe** - lásd: gyenge (gyengébb) csempe

**mező** – lásd: csempe

**orángután** - A játékos közvetlen irányítása alatt lévő állatok. El tudja kapni a pandákat, akiket ezután láncba fűzi.

**összegyűjtött pandák** - lásd: lánc

**összekötött szekrények** – X szekrény össze van kötve Y szekrénnel ha egy állat X szekrénybe belépve Y szekrényből jön ki. Minden szekrény össze van kötve egy másikkal. A játékos csak kitapasztalás útján jöhet rá ezekre az XY összeköttetésekre.

**panda** - Állat, amelyet a játékos közvetlen nem tud irányítani, mozgása *kaotikus*, hacsak nincs egy láncban egy orángutánnal.

**panda tulajdonságok** - A játékban minden panda rendelkezik pontosan egy tulajdonsággal a három pandákra jellemző tulajdonság közül. Ezen tulajdonságokról bővebben az *elfáradás*, *csilingel* illetve *sípol* bejegyzéseknél olvashatunk.

**pálya** – lásd: emelet

**pont** - Szám, amely azt határozza meg, hogy a játékos milyen hatásokkal gyűjti össze a pandákat. Ha sok állat veszik oda (=zuhan le) a gyűjtögetés során akkor a pontszám alacsony.

**sípol** – Csokiautomata által keltett hanghatás, amelyet csak a játékgép csempéjének szomszédos csempéin elhelyezkedő pandák hallanak meg. Némely panda rendelkezik azzal a tulajdonsággal, hogy sípolást hallva ugrik egyet.

**sor** - lásd: lánc

**szekrény** – Egy állat egy adott szekrénybe belépve az emelet egy másik, pontosan meghatározott szekrényében találja magát.

**szomszédos csempe** – Két csempe szomszédos ha van közös élük.

**tárgy** – Nem mozgatható. Fajtái: szekrény, fotel, játékgép, csokiautomata, bejárat, kijárat.

**törékeny csempe** - lásd: gyenge (gyengébb) csempe

**ugrás** – Egyes pandák képesek rá a csokiautomatát meghallva. Felfogható egy saját mezőre lépésnek.

**zsákutca** - Olyan csempe, amire csak egy darab csempéről lehet rálépni, illetve ugyanazon a csempén keresztül lehet kilépni.

## 2.6 Projekt terv

A projekt végrehajtásához 5 fő fog rendelkezésre állni a csapatban, akik közt az elvégzendő feladatokat közös megegyezés alapján, egyenletesen osztjuk el, nincsen tehát egy kinevezett vezér a csapatban aki egyedül döntene a felmerülő kérdésekben, csapatszinten dolgozunk.

A projekt végrehajtásának alapvető eszközei a gyűlések, amelyekről célszerűen legfeljebb egy ember hiányozhat.

Az alábbi lista tartalmazza az elkövetkezendő határidőket, amiket be kell tartani:

2019.02.18.	Követelmény, projekt, funkcionalitás - beadás
2019.02.25.	Analízis modell kidolgozása 1. - beadás
2019.03.04.	Analízis modell kidolgozása 2. - beadás
2019.03.11.	Szkeleton tervezése - beadás
2019.03.25.	Skeleton - beadás és a forráskód herculesre való feltöltése
2019.04.01.	Prototípus koncepciója - beadás
2019.04.08.	Részletes tervek - beadás
2019.04.15.	Prototípus készítése, tesztelése
2019.04.23.	Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése
2019.04.29.	Grafikus felület specifikációja - beadás
2019.05.06.	Grafikus változat készítése
2019.05.13.	Grafikus változat - beadás és a forráskód herculesre való feltöltése

2019.05.15.	Összefoglalás - beadás és feltöltés
-------------	-------------------------------------

A fontosabb gyűléseken személyesen jelenik meg mindenki közös megegyezés alapján, azonban a gyakoriság, és a flexibilitás céljából Skype-on keresztül is lezajlanak a gyűlések.

A csapatmunkát, és a kommunikációt elősegíti a személyes ismeretség, valamint további eszközök, mint a Skype, illetve a Facebook. Minden csapattag rendelkezik saját lappal, így senki nincs rászorulva külső erőforrásra.

Közös megegyezés alapján a közösen szerkesztett dokumentumokat, forráskódokat a GitHub, valamint a Google Drive segítségével osztjuk meg egymással. (Míg az előbbi inkább a programkódok megosztását segíti elő, utóbbit a beszámolók, dokumentumok megosztása, és közös szerkesztése céljából használjuk.) Ide mindenki névvel ellátva tölti fel a saját feladatát, módosításait amiről tájékoztatja a többi csapattagot is.

## 2.7 Napló

Kezdet	Időtartam	Résztevők	Leírás
2019.02.14. 19:00	0,5 óra	Martonosi, Kobál, Kis	<b>Értekezlet:</b> Feladatok kiosztása a tagok között
2019.02.15. 14:00	0,5 óra	Kis	<b>Tevékenység:</b> (2.1) Dokumentum bevezetőjének megírása
2019.02.15. 16:00	1,5 óra	Kiss	<b>Tevékenység:</b> (2.6) Projektterv kidolgozása
2019.02.15. 19:00	1,5 óra	Kobál	<b>Tevékenység:</b> (2.2.2) Követelmény szövegének megírása
2019.02.16. 15:30	0,5 óra	Kecskés	<b>Tevékenység:</b> (2.3) Követelmények kidolgozása papíron
2019.02.16. 16:00	1 óra	Kecskés, Kis, Kiss, Kobál, Martonosi	<b>Értekezlet:</b> Tisztáztuk a követelményeket. Megbeszéltünk egy gyűlést Feb 17. 20:00. Létrehoztuk a közös Drive-ot
2019.02.16. 17:00	1,5 óra	Kecskés	<b>Tevékenység:</b> (2.3) Követelmények befejezése, Drive-ba illesztése
2019.02.16. 18:00	1,5 óra	Kobál	<b>Tevékenység:</b> (2.2.1) Követelmény szövegének



			pontosítása és kiegészítése
2019.02.17. 10:00	3 óra	Martonosi	<b>Tevékenység:</b> (2.4) Use-case-ek megírása, fedlap elkészítése
2019.02.17. 17:00	2 óra	Kis	<b>Tevékenység:</b> (2.5) Szótár elkészítése
2019.02.17. 20:00	1 óra	Kecskés, Kis, Kiss, Kobál, Martonosi	<b>Értekezlet:</b> Véglegesítettük a dokumentumot, kijavítottuk a hibákat, tisztáztuk a megválaszolatlan kérdéseket. Kiss kinyomtatja a dokumentumot, és elviszi Feb 18.-án leadásra. Megbeszéltük, hogy ezentúl a személyes naplót egy erre a célra létrehozott táblázatban vezetjük.
2019.02.17. 21:00	1 óra	Kecskés	<b>Tevékenység:</b> Napló összegyűjtése és dokumentum véglegesítése

### 3. Analízis modell kidolgozása

#### 3.1 Objektum katalógus

##### 3.1.1 Armchair

A pályán véletlenszerűen elszórva fotelek is találhatóak, amelyek ugyanúgy statikus elemek a pályán, tehát nem lehet rájuk lépni, átugrani, eltolni. Ha egy panda a közvetlen szomszédságába lép, akkor a többi állat kezét elengedve, beleül és elalszik (ha hajlamos erre). Ha egy panda beleül a fotelbe, akkor onnan nem áll fel, amíg le nem telik az adott fotelre jellemző, véletlenszerű várakozási idő. Ilyenkor még egy orángután sem tudja felkelteni.

##### 3.1.2 BrokenTile

A BrokenTile már egy sérült csempe. Ennek van élete, és speciális panda el tudja törni. Ennek a pandának az ugrása képes meggyengíteni a csempét. A már gyengített csempe képes eltörni bármilyen rajta lépés hatására. Minden lépés az ugrás után eggyel csökkenti az életét az ilyen objektumnak.

A sérült csempe amikor eltörik, akkor szól a szomszédjainak, hogy már nem lehet ráállni. Egy eltörő csempe megöli a rajta található állatot is.

##### 3.1.3 Entry

A bejárat egy tárgy, nem mozgítható, és nem törékeny csempén van. Az emeleten csak egy bejárat lehet. A játék elején a bejáratból lépésenként egy-egy orángután megy ki, addig, amíg az emeleten a mennyiségük nem lesz megfelelő. Miután a pandák sora végig ment a kijáraton, egy orángután megjelenik a bejáratból. A bejáratba nem lehet belépni.

##### 3.1.4 Exit

A kijárat egy tárgy, nem mozgítható, és nem törékeny csempén van. A kijáratnak az emelethez csatlakozó külön dedikált csempéje van. Az emeleten csak egy kijárat lehet. A kijáraton keresztül az orángután átvezetheti a pandákból láncolt sort. Ha az orángután belépett a kijáraton, egy időre eltűnik a pályáról. Minden a kijáraton áthaladt pandáért a játékos pontot kap. Ha egy panda átment a kijáraton, eltűnik az emeletről. A kijáratba csak orángután, és az utána vezetett lánc léphet be.

##### 3.1.5 GameMachine

A játékgép hasonlóan a csokiautomatához, szintén egy csempét foglal el, rálépni, átugrani, eltolni nem lehet. Egy pályán belőle is több lehet. A játékgép véletlenszerű időközönként csilingel, amitől a szomszédságában álló pandák megijedhetnek (ha hajlamosak erre), amitől elengedik a többi állat kezét.

##### 3.1.6 Jumping Panda

Speciális tulajdonsággal bíró panda, amely a sípolástól ugrik. Feladata, hogy értesítse a csempét, amelyen ugrott (=ismét rálépett).

##### 3.1.7 Lazy Panda

A játékban szereplő pandák közül az egyik olyan fajta, aminek a tulajdonsága van, hogy lusta. A lustaság azt jelenti, hogyha a játékban szereplő fotelek közül elhalad az egyik

mellett, akkor leül oda megpihenni bizonyos ideig (ha láncba volt fűzve, elengedi az előtte álló mancsát).

### **3.1.8 Orangutan**

A játékos által irányított karakterek, amelyek a pandákat fűzik láncba maguk mögött. Törött csempére lépve lezuhanak. Mindegyik orángután lezuhanása esetén a játék véget ér. Az orángutánok képessége és célja a kijáraton át kivezetni a láncba fűzött pandákat.

### **3.1.9 Panda**

Alapértelmezetten a gépek által irányított karakterek. Láncba fűzve követik az őket irányító orángután mozgását. A játékos a kijáraton átvezetve őket pontot kap értük. Törött csempére lépve lezuhan. Ha az emeleten elfogynak a pandák a játék véget ér.

### **3.1.10 RegularTile**

A RegularTile egy általános csempe. Nem rendelkezik élettal, tehát nem tudja panda eltörni sem. Csak ilyen csempén lehet tárgy.

### **3.1.11 Scary Panda**

Speciális tulajdonsággal bíró panda, amely a csilingeléstől megijed. Felelőssége, hogy miután meghallotta a csilingelést elengedje a láncban előtte álló mancsát és utasítsa a láncban mögötte álló pandákat, hogy tegyenek hasonlóan.

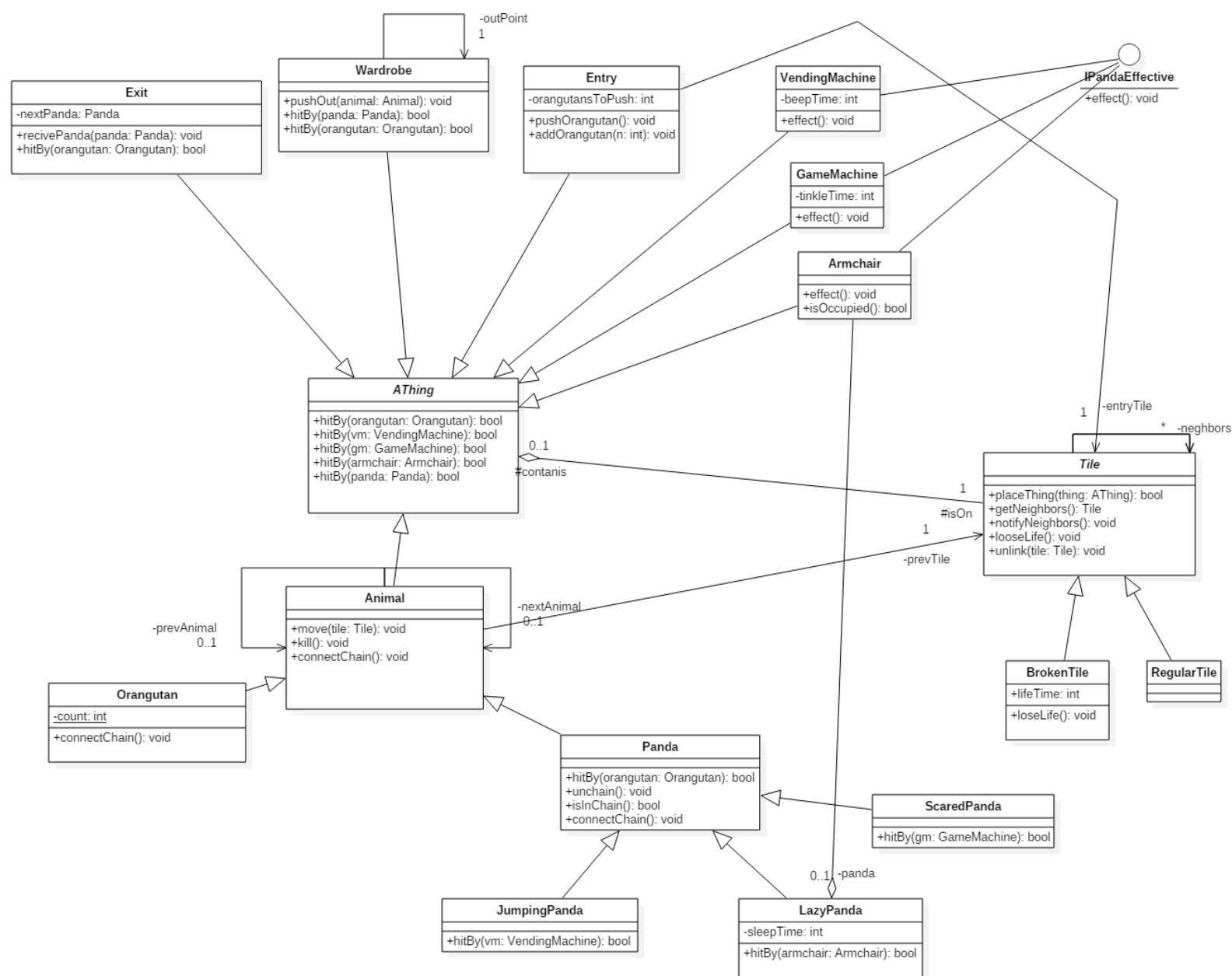
### **3.1.12 VendingMachine**

A csokiautomata egy statikus elem a pályán. Pontosan egy csempén helyezkedik el, nem lehet sem rálépni, sem átugrani sem eltolni. A pályán véletlenszerűen helyezkedhetnek el, egy pályán akár több is. Legfőbb feladata, hogy véletlenszerű időközönként sípol, ezáltal a közvetlen szomszédságában lévő csempéken álló pandákat megijesztheti (ha hajlamosak erre), amik ijedtükben ugranak egyet.

### **3.1.13 Wardrobe**

A szekrény egy tárgy, ahogy más tárgyakat sem, a szekrényt sem lehet elmozdítani, sem pedig ráállni az általa elfoglalt csempére. Alatta nem törékeny csempe van. A szekrénybe be lehet lépni. Ha egy állat belép egy szekrénybe, akkor egy másik az emeleten elhelyezett, szekrényből jön ki. A szekrények össze vannak kötve egymással, tehát az állat egy szekrénybe belépve egyértelműen egy másik szekrényből jön ki, de a szekrények összeköttetései nincsenek jelölve, a játékos magától fedezi fel.

### 3.2 Statikus struktúra diagramok



### 3.3 Osztályok leírása

#### 3.3.1 Animal

- **Felelősség**

A játékban lévő állatokat általánosan reprezentáló abstract osztály. Rendeződhetnek láncokba, lezuhanhatnak, illetve távozhatnak az emeletről a kijáraton.

- **Ősosztályok**

- **AThing**

- **Interfészek**

-

- **Attribútumok**

- **previousAnimal: Animal:** Az állat előtt álló állat. Ha nincs akkor null.
- **nextAnimal: Animal:** Az állat után álló állat (csak Panda lehet egyébként). Ha nincs akkor null.
- **prevTile: Tile:** Az állat előző lépésben elfoglalt pozíciója.

- **Metódusok**

- Fenti attribútumokhoz setter/getter
- **move(t: Tile): void:** Az argumentumként kapott t csempére lépteti az állatot.
- **kill(): void:** Megöli az adott állatot, azaz végérvényesen eltávolítja a játékból. Ha a lezuhanó állat láncban volt, akkor megtöri a láncot.
- **connectChain(): void:** A Pandalánc összeszervezése a feladata a függvénynek.

#### 3.3.2 Armchair

- **Felelősség**

Foteleket kezelő osztály, ha egy panda a szomszédságába kerül, és hajlamos az elalvásra, akkor jelzi számára, hogy aludjon el.

- **Ősosztályok**

- **AThing**

- **Interfészek**

- **IPandaEffective**

- **Attribútumok**

- **panda: Panda:** A fotelben ülő panda.

- **Metódusok**

- **isOccupied(): bool:** Visszadja, hogy az adott fotelben már ülnek-e, ha igen, az értéke true.
- **effect(): void:** Minden körben jelzi a szomszédos csempéknek, hogy ha van rajtuk panda, azt küldjék bele aludni.

### 3.3.3 AThing

- **Felelősség**

Absztrakt őszosztály, ami a tárgyakat és az állatokat reprezentálja, biztosítja a rájuk jellemző tulajdonságokat. Tudja, hogy melyik csempén helyezkedik el. El lehet helyezni csempére.

- **Őszosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **isOn: Tile:** Az a csempe, amin elhelyezkedik a tárgy, vagy állat.

- **Metódusok**

- **setTile(Tile: tile): void:** Elhelyezi a tárgyat/állatot a paraméterként átadott csempére, ha a csempe szabad, és tárgy esetén nem gyenge a csempe.
- **hitBy(<thing: AThing>): bool:** Általános implementáció, ha egy tárgy, vagy egy állat megpróbál interakcióba lépni az adott dologgal, akkor nem engedi a csempére lépni, hanem visszaküldi arra a csempére, ahonnan megpróbált lépni. False értéket ad vissza, mivel nem volt sikeres a lépés. A hitBy függvények sok helyen felül vannak írva az adott tárgy tulajdonságaitól függően.

Jelmagyarázat: <thing: AThing> - A hitBy függvény minden létező leszármazott tárgyra vagy állatra rendelkezik sajátos paraméterezéssel. Mivel mindegyik függvény ezen az absztrakciós szinten ugyan azzal az implementációval rendelkezik, ezért nem tartottuk fontosnak kiírni a magyarázatot az összes praraméterezésre. Az <thing: AThing> helyére behelyettesítendő paraméterezések:

- orangutan: Orangutan
- panda: Panda
- gm: GameMachine
- vm: VendingMachine
- armchair: Armchair

### 3.3.4 BrokenTile

- **Felelősség**

A törött csempe osztálya ez. A törött csempe képes eltörni és megölni az állatokat rajta, ha az élete nullára csökken.

- **Őszosztályok**

- **Tile**
- **Interfészek**
- 
- **Attribútumok**
  - **lifeTime: int:** A hátralévő ugrások számát tárolja. Ha nullára fogy, akkor törlik el a csempe.
- **Metódusok**
  - **loseLife(): void:** A meghívás hatására csökkenti az életét a csempének.

### 3.3.5 Entry

- **Felelősség**  
Bejáratot reprezentáló osztály. Ha van még kiküldendő orángután, akkor a kör elején kiküld egyet, majd csökkenti eggyel a kiküldendő orángutánok számát.
- **Ősosztályok**
  - **AThing**
- **Interfészek**
- 
- **Attribútumok**
  - **orangutansToPush: int:** Azon orángutánok száma, akiket még ki kell küldeni az emeletre.
- **Metódusok**
  - **pushOrangutan(): void:** Létrehoz egy orángutánt, és kihelyezi az első elérhető szomszédos csempére, majd csökkenti a kiküldendő orángutánok számát eggyel.
  - **addOrangutan(n: int): void:** Hozzáad a kiküldendő orángutánok számához egy paraméterként átadott mennyiséget.

### 3.3.6 Exit

- **Felelősség**  
Kijáratot reprezentáló osztály, ha interakcióba lép egy orángutánal, akkor eltünteti azt az emeletről. Amikor az orángután a kijárhoz vezette a pandákat, átadja a mögötte lévő pandát a kijáratnak, ezek után a kijárat lépésről lépésre befogadja a pandákat. Ha egy pandákból álló lánc végig ment a kijáraton, hozzáad egy kiküldendő orángutánt a bejárhoz.
- **Ősosztályok**
  - **AThing**
- **Interfészek**

-

- **Attribútumok**

- **nextPanda: Panda:** A következő befogadható panda.

- **Metódusok**

- **receivePanda(panda: Panda): void:** Befogadja a következő pandát, tovább mozgatja a sort, minden panda befogadásával emeli a játékos pontszámát. Ha a pandák sora véget ért, akkor a bejáraton eggyel növeli a kiküldendő orángutánok számát.
- **hitBy(orangutan: Orangutan): void:** Felülírja az őszosztály metódusát, ha egy orángután interakcióba lép a kijáratral, akkor eltünteti a pályáról, beállítja a nextPanda attribútumát, és sorba befogadja a pandákat. True-val tér vissza, ha az orángutánnak sikerült átlépni a kijáraton, ellenkező esetben false.

### 3.3.7 GameMachine

- **Felelősség**

Játékgépeket kezelő osztály, csilingeléssel megijeszti a körülötte lévő, arra hajlamos pandákat.

- **Őszosztályok**

- **AThing**

- **Interfészek**

- **IPandaEffective**

- **Attribútumok**

- **TinkleTime: int:** A játék inicializálása során megkapott random szám, ami alapján adott játékgép tudja, hogy hány körönte kell csilingelnie.

- **Metódusok**

- **effect(): void:** Ha eltelt a TinkleTime attribútumban tárolt számú kör, akkor jelzi a körülötte lévő csempéknek, hogy ijesszék meg a megfelelő pandákat.

### 3.3.8 IPandaEffective

- **Felelősség**

Egy olyan interface, ami egységes hatással rendelkező tárgyakat fog össze. Ilyen tárgy az, ami tud csengeni, sípolni vagy elfárasztani közelben lévő Pandát. Minden ilyen környező területet értesít, hogy milyen típusú jelzés érkezik.

- **Őszosztályok**

-

- **Interfészek**

-



- **Attribútumok**

-

- **Metódusok**

- **effect(): void:** Egy olyan függvény, amit meghívva az implementált osztályban megszólítja a szomszéd csempéket, hogy éppen valamilyen hatás zajlik.

### 3.3.9 JumpingPanda

- **Felelősség**

A JumpingPanda példányok sípszó hallatán ugranak egyet.

- **Ősosztályok**

**Panda**

- **Interfészek**

- -

- **Attribútumok**

- -

- **Metódusok**

- **hitBy(vm: VendingMachine): bool:** A panda meghallja a csokiautomata(vm) sípolását és ugrik egyet. True-t adunk vissza, mert JumpingPanda esetében ez mindig bekövetkezik. Ilyenkor a panda eggyel csökkenti annak a csempének az életét amin éppen áll. (Meghívja a loseLife() függvényét.)
- **hitBy(gm: GameMachine): bool:** A JumpingPanda a csilingelésre nem reagál. A függvény törzse mindösszesen egy false visszaadása, jelezve, hogy semmi nem történt.
- **hitBy(ac: Armchair): bool:** A JumpingPanda nem reagál a fotel közelségére. Visszatérünk egy false-szal és semmi mást nem csinálunk.

### 3.3.10 LazyPanda

- **Felelősség**

A játékban szereplő lusta pandák. Ezeknek tulajdonsága, hogyha elhaladnak egy fotel mellett, akkor elfáradnak és beleülnek a fotelre jellemző ideig (körig). Ez alatt az idő alatt a panda nem mozog, még egy orángután sem tudja felkelteni. Az idő letelte után a panda folytatja a rá jellemző véletlenszerű mozgását.

- **Ősosztályok**

- **Panda**

- **Interfészek**

- -

- **Attribútumok**

- **sleepTime: int:** A pandák akkor mozoghatnak ha a sleep time értékük nulla. Amennyiben ez nem teljesül, akkor az azt jelenti, hogy a panda még annyi körig alszik, amennyi az attribútum értéke.

- **Metódusok**

- **hitBy(vm: VendingMachine): bool:** Ez a fajta panda figyelmen kívül hagyja a csokiautomata(vm) hanghatásait, így ez a függvény nem több, mint egy false visszaadása.
- **hitBy(gm: GameMachine): bool:** A LazyPanda a csilingelésre nem reagál. A függvény törzse mindösszesen egy false visszaadása, jelezve, hogy semmi nem történt.
- **hitBy(ac: Armchair): bool:** Ez a függvény akkor hívódik, ha egy lusta panda elhalad egy fotel mellett. Ekkor beleül a fotelbe, sleepTime attribútumát beállítja a pandára jellemző értékre, és elalszik.

### 3.3.11 Orangutan

- **Felelősség**

A játékosok által irányított állatok, akik a pandákat fogják meg.

- **Ősosztályok**

- **Animal**

- **Interfészek**

-

- **Attribútumok**

- **cnt: static int:** A játékban még meglevő orángutánok száma.

- **Metódusok**

### 3.3.12 Panda

- **Felelősség**

Pandákat reprezentáló abstract osztály. Az egyes pandák fáradékonyak, ijedősek vagy sípszóra ugranak.

- **Ősosztályok**

- **Animal**

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **hitBy(o: Orangutan): bool:** Kezeli az esetet, amikor a panda csempéjére egy orángután akar lépni. Az orángután a pandára lépve befűzi a pandát a láncába.

- **unchain(): void:** Az állat, amin meghívjuk a függvényt elengedi az előtte álló állat mancsát, majd az megint az előtte állóét és így tovább. A függvény rekurzív működésű. A futás végeztével a leszakadt állatok már nem tudnak egymásról (previousAnimal és nextAnimal is null).
- **isInChain(): bool:** Megadja, hogy az adott állat láncban van-e. Ezt a next/previous animal értéke pontosan meghatározza: Ha mindkettő null, akkor false.

### 3.3.13 RegularTile

- **Felelősség**

Tárolja a tárgyakat és lekezeli az lépéseket, zajokat. És mindent tud, amit az őosztálya természetesen.

#### Őosztályok

- **Tile**
- **Interfészek**  
-
- **Attribútumok**  
-
- **Metódusok**
  - **loseLife(): void:** Egy üres függvény, csak a BrokenTile miatt van ott, hogy kompatibilis legyen.

### 3.3.14 ScaredPanda

- **Felelősség**
- A ScaredPanda példányai a játékgép csilingelésétől megijednek és elengedik az előttük álló állatok mancsát.
- **Őosztályok**
  - **Panda**
- **Interfészek**  
-
- **Attribútumok**  
-
- **Metódusok**
  - **hitBy(gm: GameMachine): bool:** A ScaredPanda a csilingelésre elengedi az előtte álló állat mancsát. Mivel ez mindig megtörténik a függvény igaz értékkel tér vissza.

### 3.3.15 Tile

- **Felelősség**

Absztrakt ősosztály, ami a csempéket reprezentálja és meghatározza azokat a metódusokat, amik minden csempe esetében elengedhetetlennek számítanak.

A tile abstract osztályba sorolható minden pályán található csempe, ami a mezőket alkotja.

Csempéből van több változat, mind különböző változattal rendelkezik.

Minden csempének van szomszédja és be tud fogadni maximum egy dolgot. Minden csempe értesíti a szomszédos csempéket, ha egy tárgy rajta zajt ad.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **contains: AThing:** Tárolja azt a tárgyat, vagy állatot ami a csempén található. Tárgyat vagy állatot képes így, és maximum egyet tud.
- **neighbors: Tile:** A szomszédos csempéket nyilvántartó kollekción. Ide léphetnek az állatok, és ide szól a zaj át.

- **Metódusok**

- **placeThing(thing: AThing): bool:** A csempére mozgó állatot fogadja be, ha tudja, mert szabad. (Vagy mert egyéb okok miatt lehetséges, panda befűzés, stb...)
- **notifyNeighbors(): void:** Szól a szomszéd csempéknek, hogy zaj van mellettük, vagy fotel.
- **loseLife(): void:** Ez a virtuális függvény azért van, hogy kompatibilis legyen a BrokenTile a pályával.
- **unlink(tile: Tile): void:** Törli a szomszédjai közül a kapott csempét. Törött csempe esetén hívjuk ezt a függvényt.

### 3.3.16 VendingMachine

- **Felelősség**

Csokiautomatákat kezelő osztály, sípolással megijeszti a körülötte lévő, arra hajlamos pandákat.

- **Ősosztályok**

- **AThing**

- **Interfészek**

- **IPandaEffective**

- **Attribútumok**

- **BeepTime: int:** A játék inicializálása során megkapott random szám, ami alapján adott csokiautomata tudja, hogy hány körönte kell sípolnia.

- **Metódusok**

- **effect():void**: Ha eltelt a BeepTime attribútumban tárolt számú kör, akkor jelzi a körülötte lévő csempéknek, hogy ijesszék meg a megfelelő pandákat.

### 3.3.17 Wardrobe

- **Felelősség**

Szekrényt reprezentáló osztály, ha egy állat egy szekrénnel lép interakcióba, akkor egy, ezzel a szekrénnel összekötött, másik szekrényből kerül ki. A második szekrény egy szomszédos mezőre dobja ki az első szekrénybe bekerült állatot.

- **Ősosztályok**

- **AThing**

- **Interfészek**

-

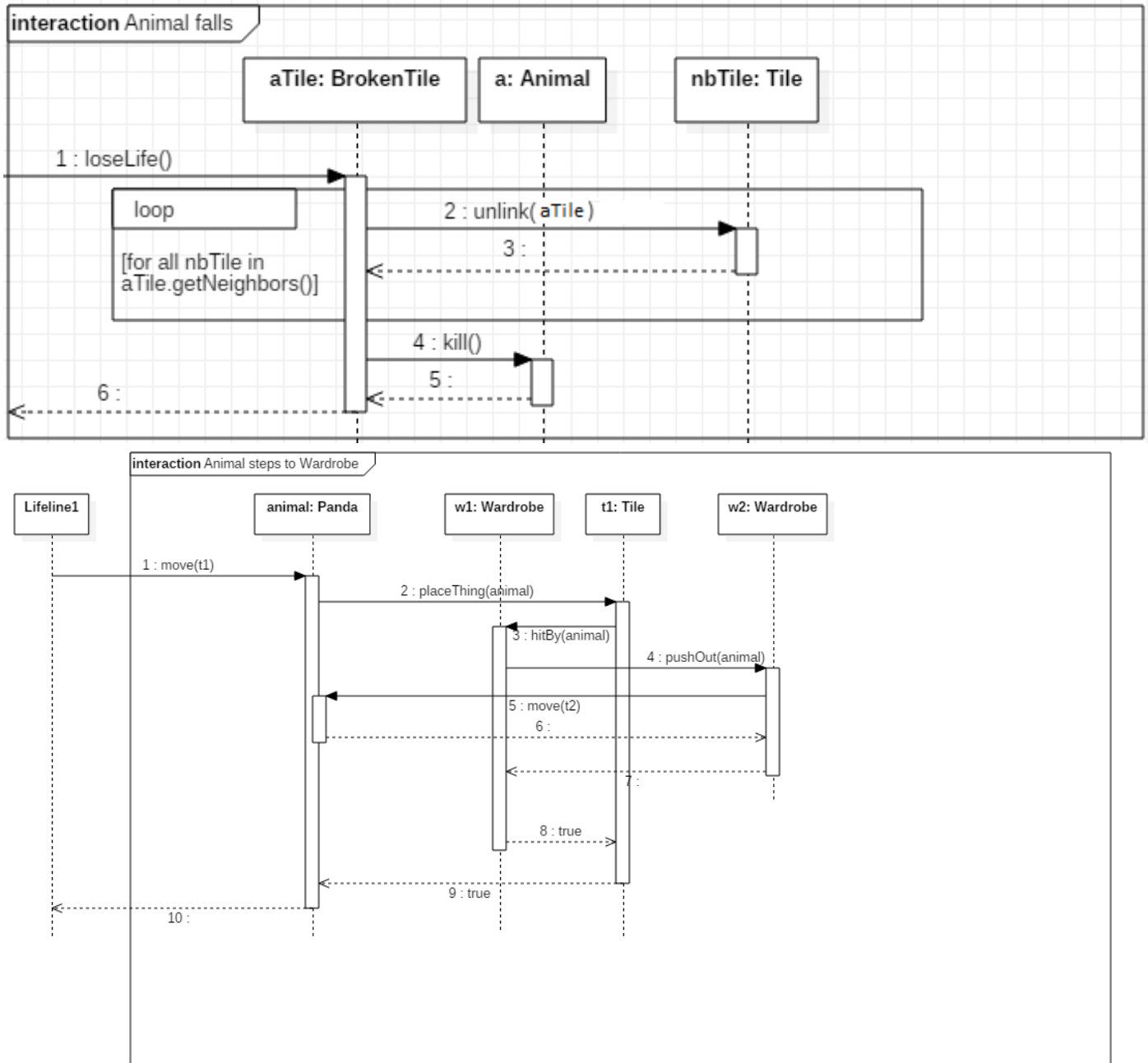
- **Attribútumok**

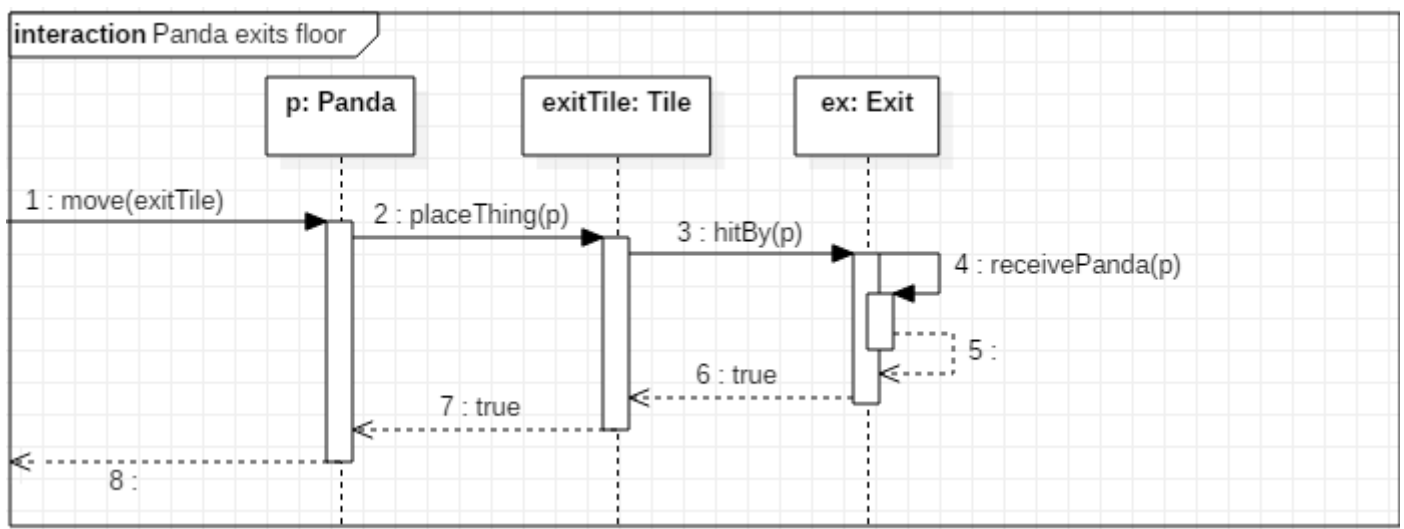
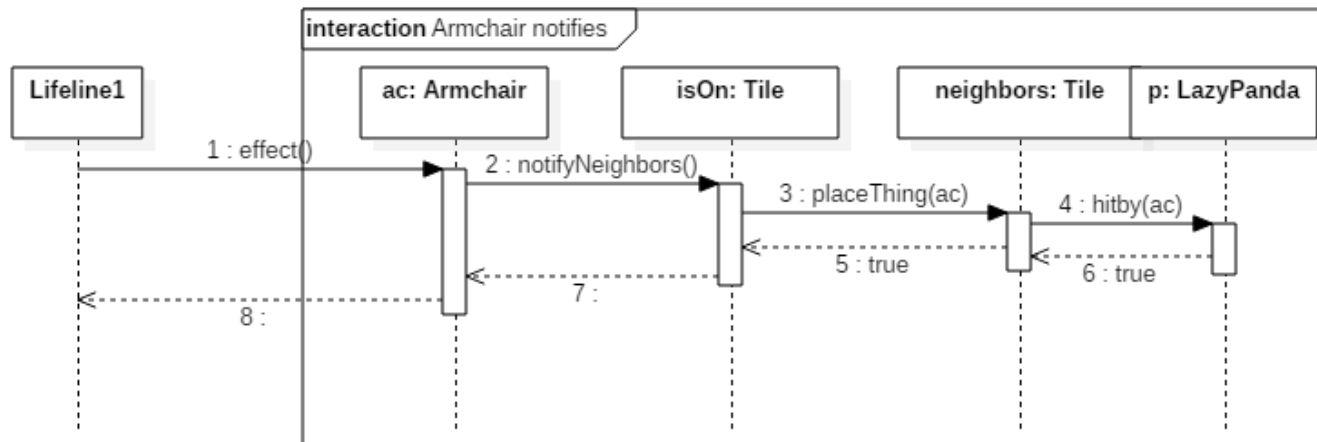
- **outPoint: Wardrobe**: Egy másik szekrény, ahonnan az állat majd kikerül.

- **Metódusok**

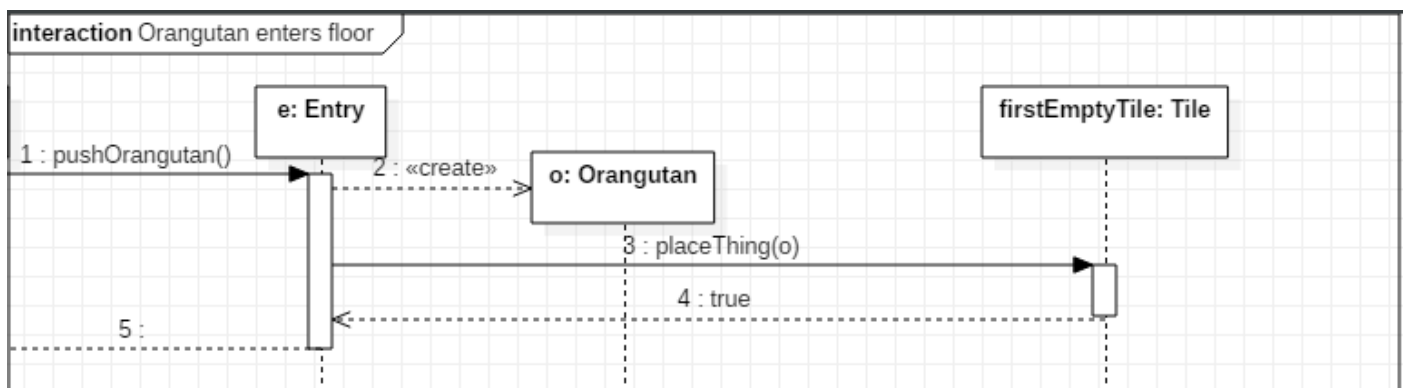
- **hitBy(orangutan: Orangutan): bool**: Felülírja az ősosztály metódusát, átadja az orángutánt egy megfelelő másik szekrénynek, hogy az majd kidobja. True-val tér vissza, ha sikerült az orángutánt átvinni a szekrények között, és elhelyezni egy csempén, ellenkező esetben false.
  - **hitBy(panda: Panda): bool**: Felülírja az ősosztály metódusát, átadja az pandát a pár szekrénynek, hogy az majd kidobja. True-val tér vissza, ha sikerült az pandát átvinni a szekrények között, és elhelyezni egy csempén, ellenkező esetben false.
  - **pushOut(animal: Animal): void**: Elhelyezi az állatot az első elérhető szomszédos mezőre.

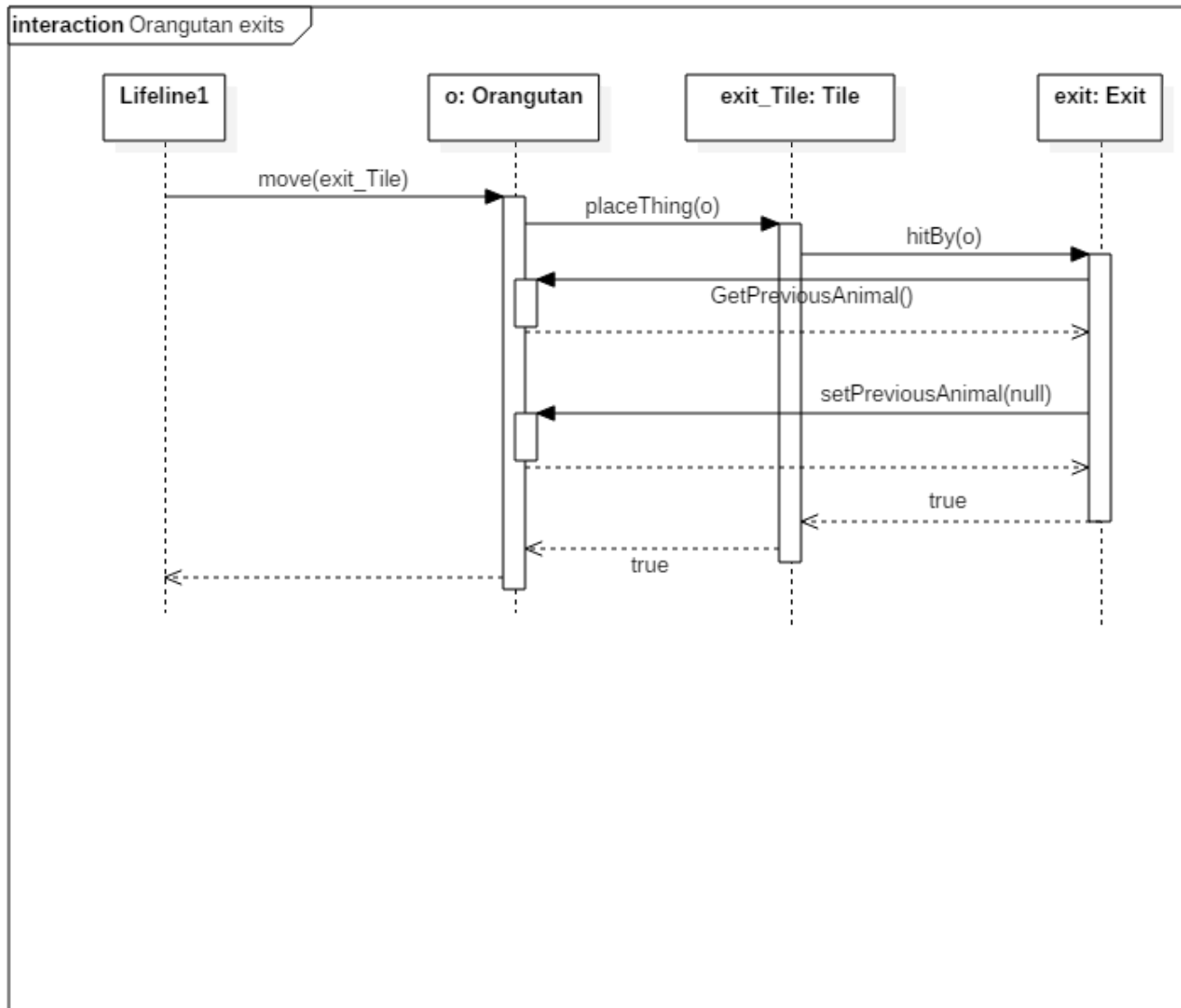
### 3.4 Szekvencia diagramok

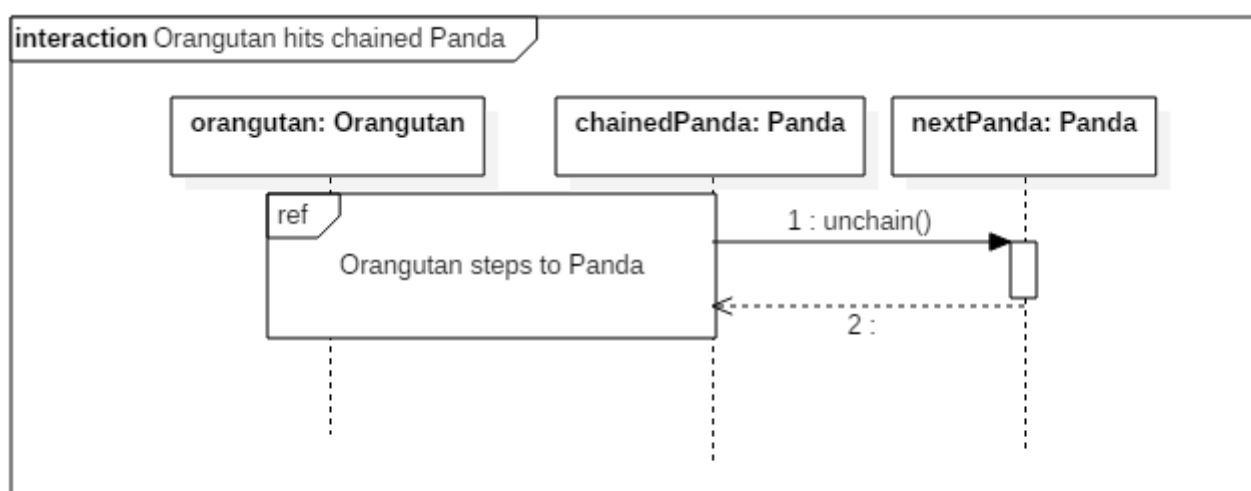


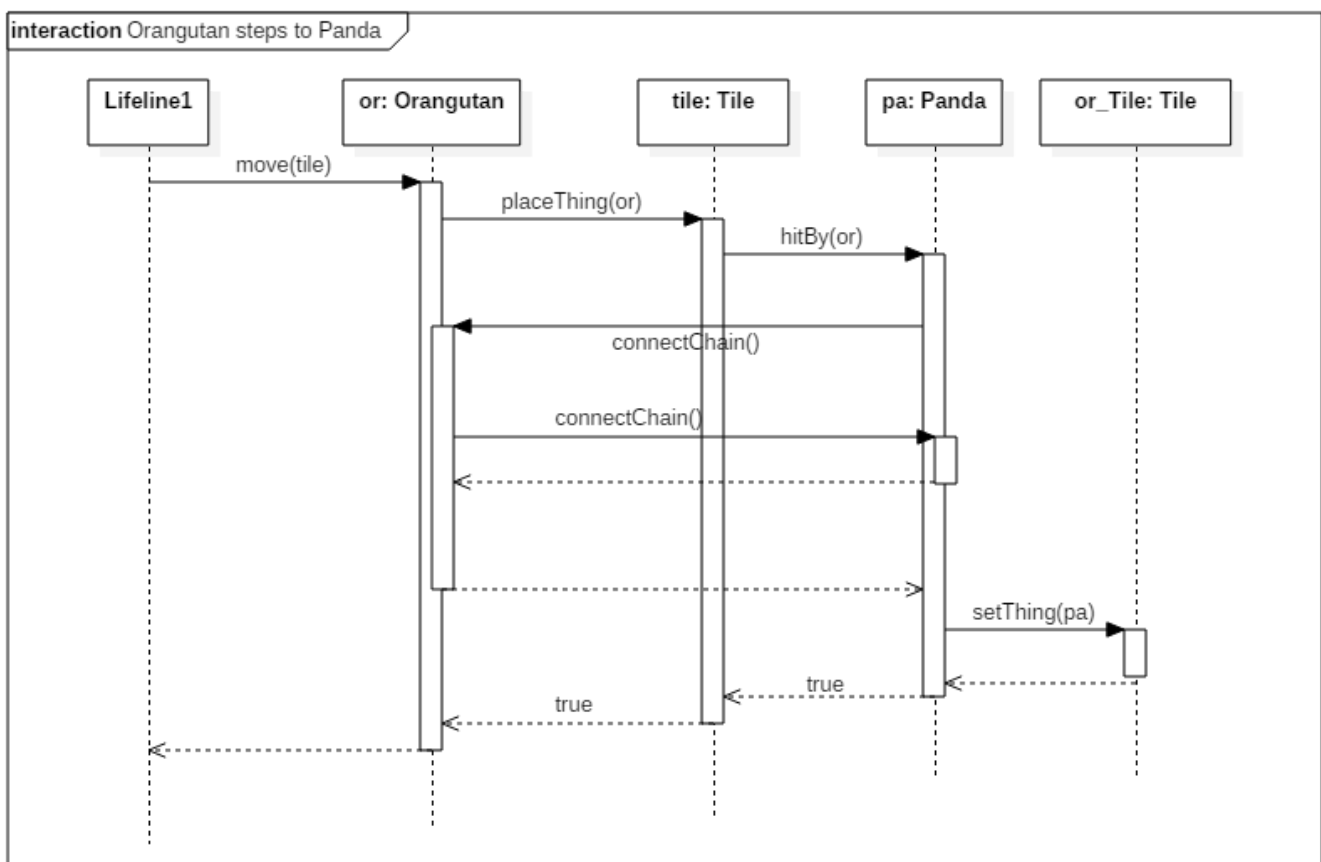
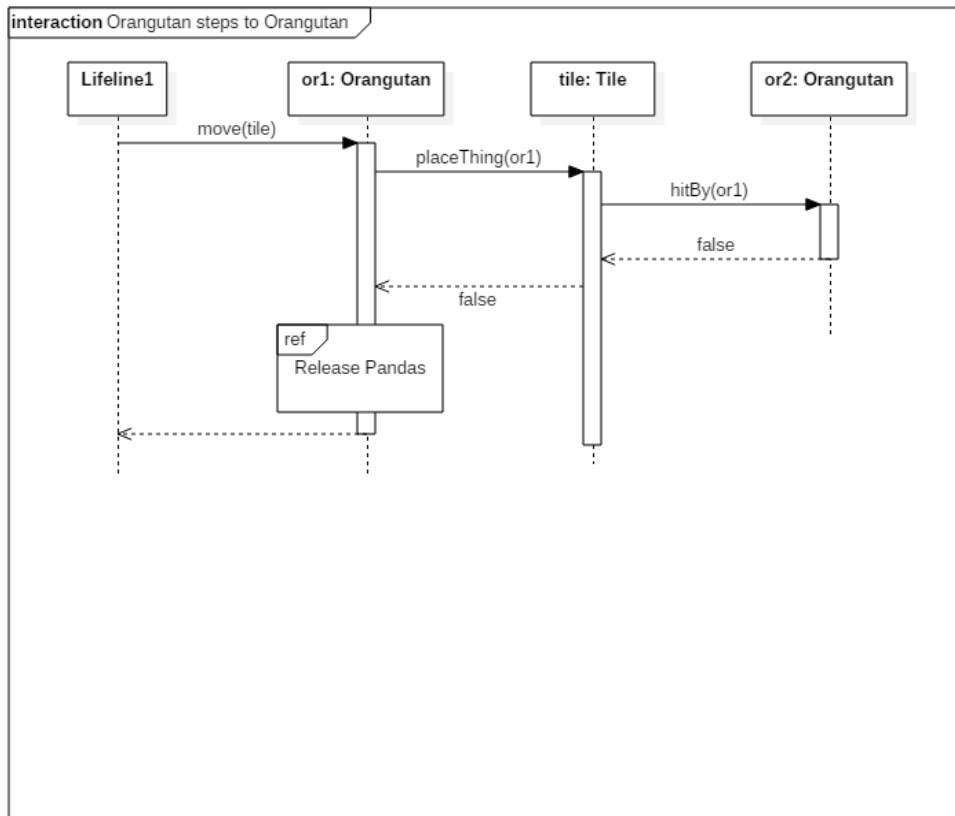


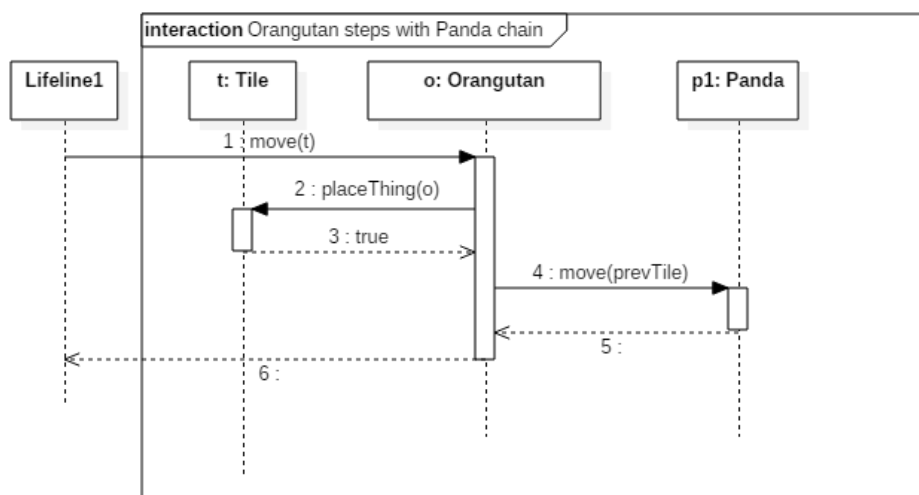
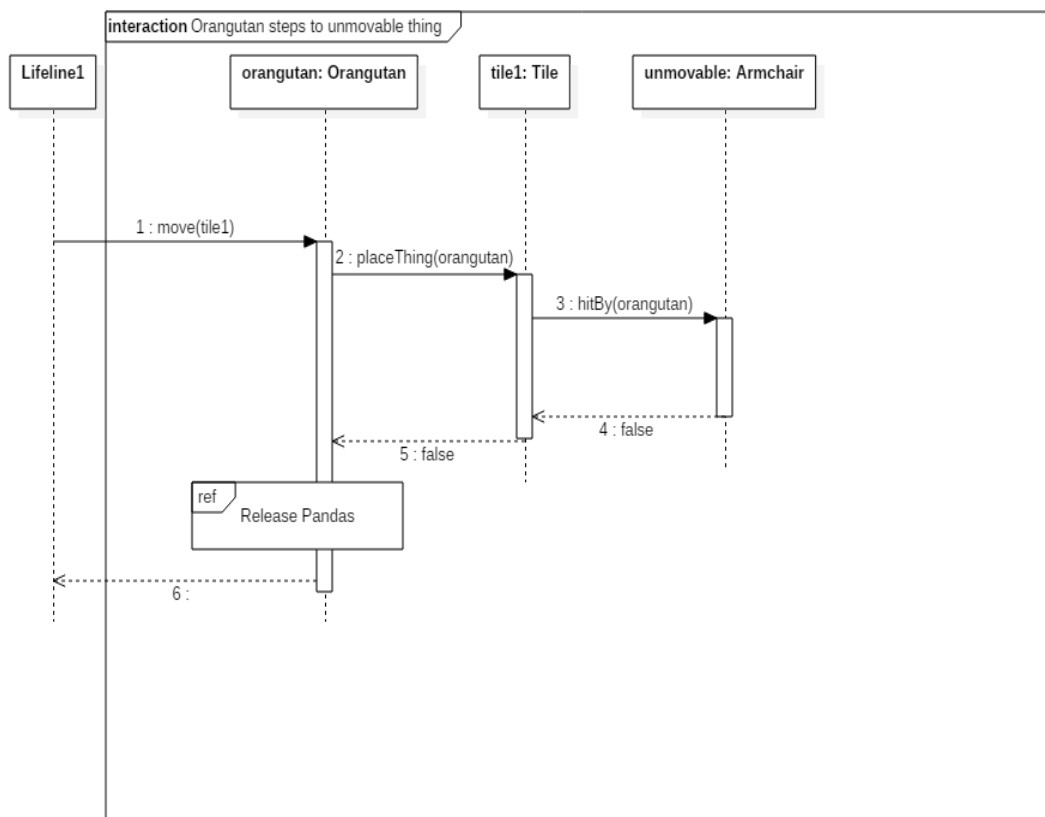


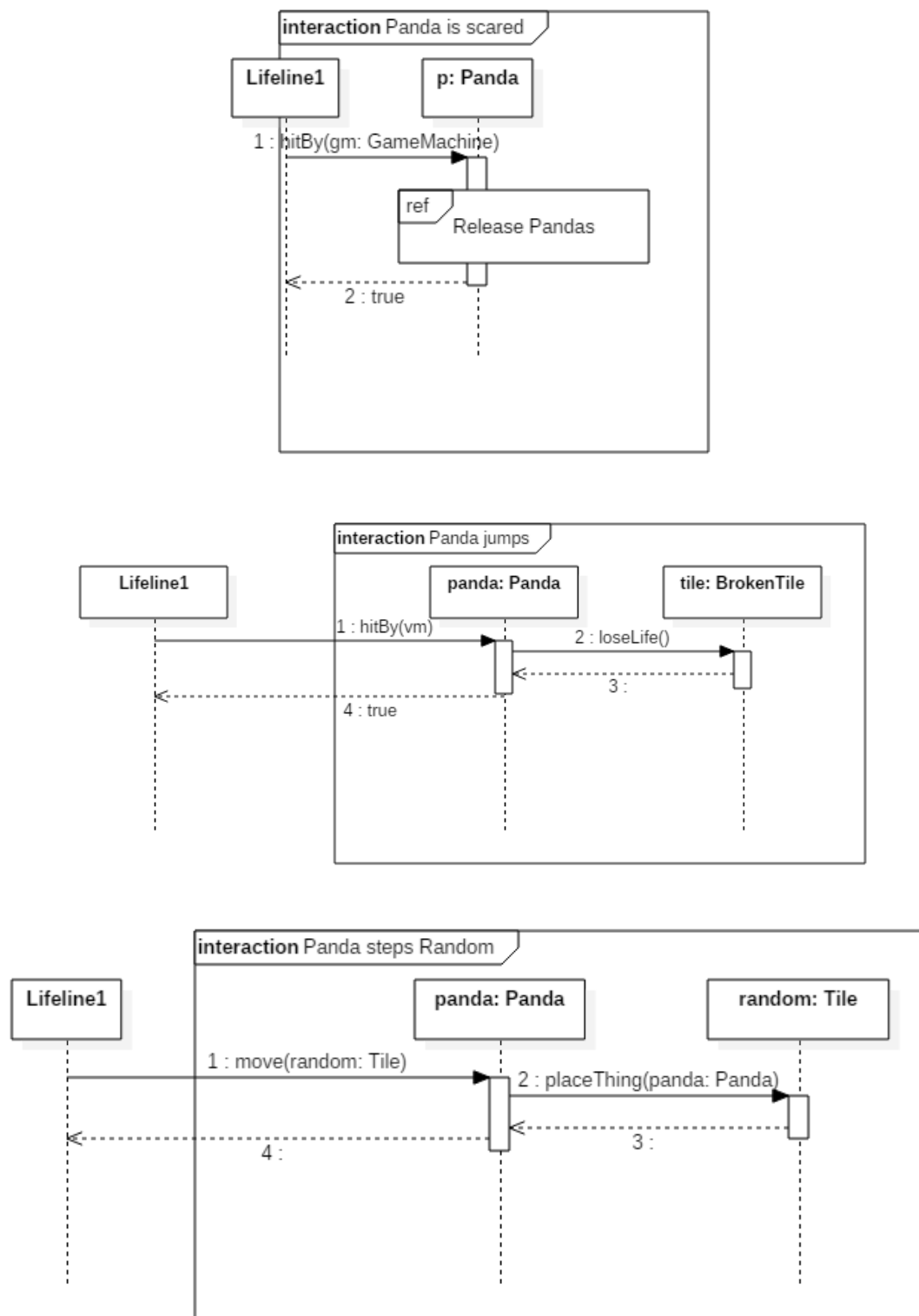


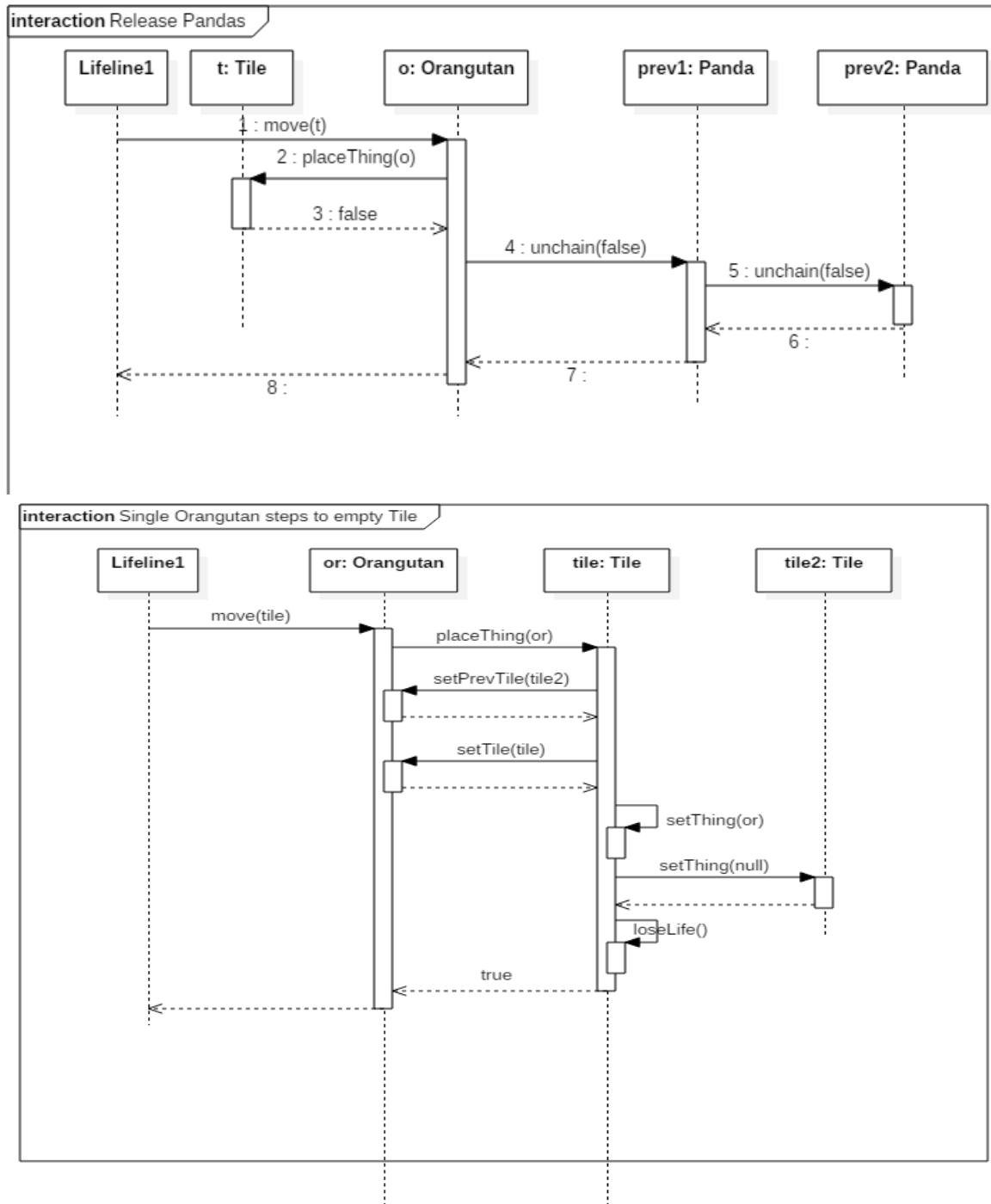


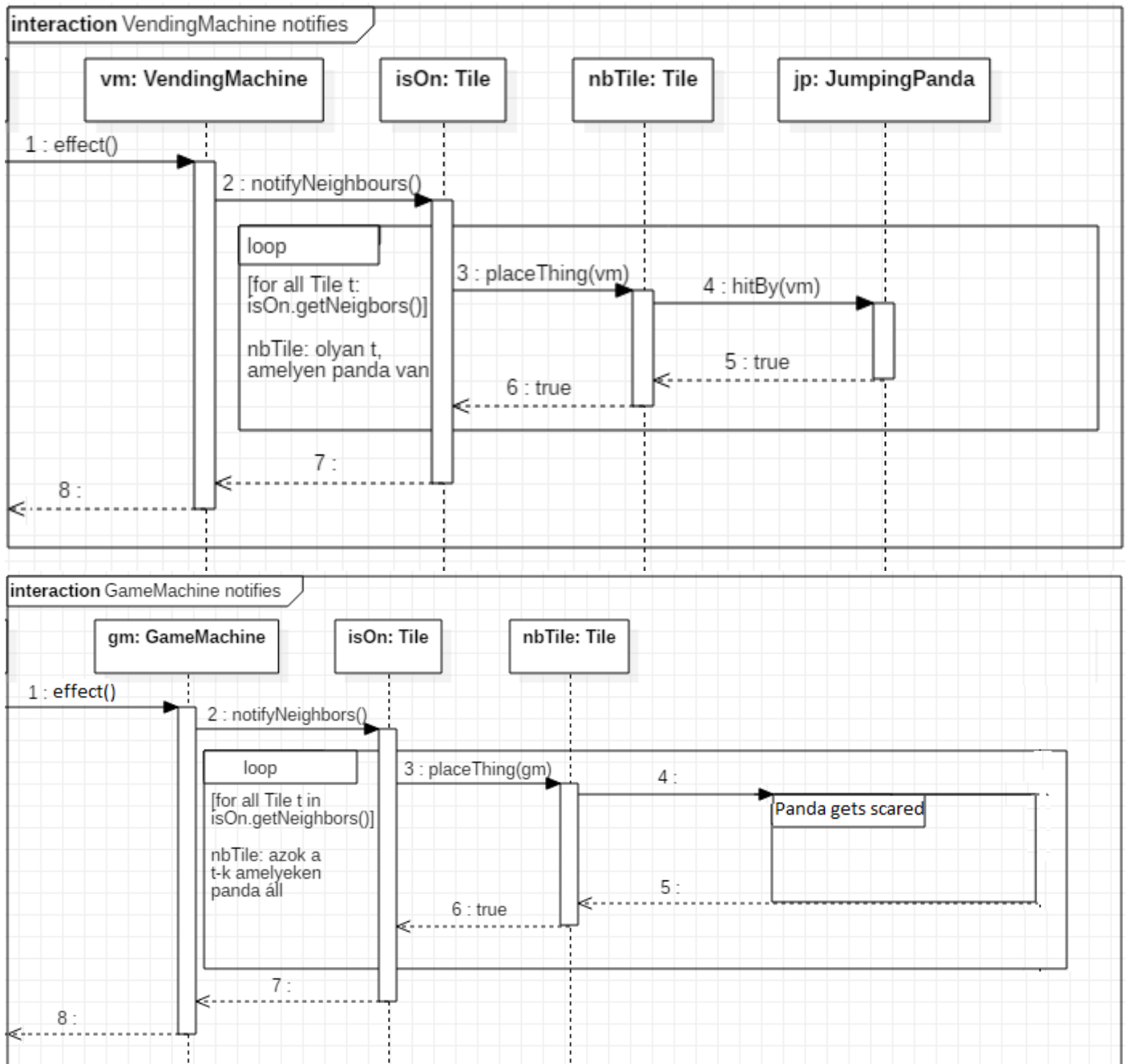




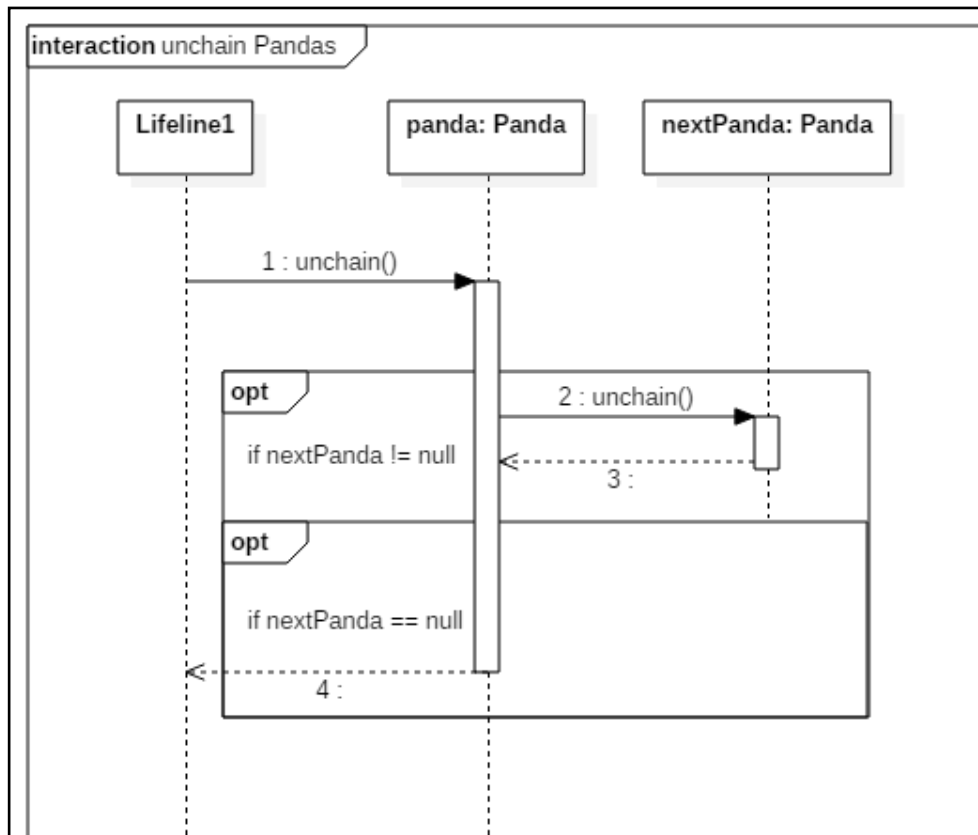












### 3.5 State-chartok

Nincsen fontosabb State-chart, amit kellene implementálni a fejlesztés jelenlegi szakaszában.

### 3.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2019.02.20- 13:00	0.5 óra	Kecskés, Kis, Kiss, Kobál, Martonosi	<b>Értekezlet:</b> Feladatok felosztása és pontok meghatározása. <b>Döntés:</b> Kiss és Kis specifikálják az állatokat reprezentáló objektumokat és osztályokat, Martonosi és Kobál specifikálják a tárgyakat, Kecskés foglalkozik a csempékkel és az emelettel. Mindenkinek el kell készítenie a saját részét Feb 23. 14:00 határidővel. Következő értekezlet Feb 23. 14:00 kezdéssel, amikor terv szerint egyeztetünk objektumokat és osztályokat. Megtervezzük az objektumdiagramot, és felosztjuk egymás között a feladatok maradékát.
2019.02.22- 15:00	2.0 óra	Kobál	A bejárat, kijárat, szekrény és ezek interfészeinek, megőssztályainak specifikálása.
2019.02.22- 16:00	0.5 óra	Kecskés	Objektumok és Osztályok kidolgozása StarUML-lel.
2019.02.22- 17:00	1.0 óra	Kis	Orangután, és LustaPanda objektum és osztályának megírása.
2019.02.22. 20:00	1.0 óra	Martonosi	A csokiautomata, játékgép, fotel és ezek interfészeinek, megőssztályainak specifikálása
2019.02.23. 14:00	1.0 óra	Kecskés, Kiss, Kobál,	<b>Értekezlet:</b> Törékeny csempék kérdését, megbeszéltük a pandák és az orangutánok mozgását. <b>Döntés:</b> Törékeny csempék leszarmaznak a csempékből a

		Martonosi	modularitás érdekében. Az orángutánok sarokba szorítva eleresztik az összegyűjtött pandákat. Értekezlet folytatása Feb 23. 19:00 kezdéssel.
2019.02.23. 16:30	1.0 óra	Kiss	A megbeszélésen megbeszélteknek megfelelően módosítja az eddig elkészített feladatokat. Állatok kidolgozása objektum és osztálydiagrammal.
2019.02.23. 17:00	0.5 óra	Kis	Objektumlista (Pandák, LustaPanda, Ijedős Panda, Ugró Panda) elkészítése.
2019.02.23. 19:00	1.5 óra	Kecskés, Kis, Kiss, Kobál, Martonosi	<b>Értekezlet:</b> Osztálydiagram, tisztázása, a szükséges modellezési kérdések megbeszélése, és egymás között a szekvencia diagrammok elkészítésének felosztása (lásd "Szekvenciák" doksi). <b>Döntés:</b> Másnapra mindenki pótolja az elmaradt és eddig nem értett szekvencia diagramokat. Lezáró értekezlet Feb 24. 19:00 kezdéssel.
2019.02.23. 20:30	1.5 óra	Martonosi	Kiosztott szekvencia diagramok elkészítése
2019.02.23. 22:00	1.0 óra	Korbál	Orangutan steps to unmovable thing, Animal steps to Wardrobe, Orangutan steps with Panda chain és a VendingMachine notifies szekvencia diagram elkészítése.
2019.02.24. 13:00	2.0 óra	Kiss	Az 1., 2., és 3. számú szekvencia diagramok elkészítése.
2019.02.24. 16:00	2.0 óra	Kis	Szekvenciadiagramok készítése
2019.02.24. 19:00	3.5 óra	Kecskés, Kis, Kiss, Kobál, Mar- tonosi	<b>Értekezlet:</b> Szekvencia diagramok ellenőrzése, hiányok részek pótlása. Közös munka az értekezlet alatt. Bármilyen helyben felmerült kérdés helyrerakása.
2019.02.24. 22:45	2.5 óra	Kecskés	Képek és szöveg egybefűzése, egységesre formázása. A napló kiegészítése és így a végleges dokumentum összeállítása.
2019.03.02. 12:00	1.0 óra	Kis	Osztályok leírásánál ABC sorrend kialakítása, apróbb ésszerűsítések, pontosítások
2019.03.02. 16:00	0.5 óra	Kobál	AThing osztály hitBy() függvényének általános leírásának kijavítása. A visszatérési értékek az Entry, Exit és Wardrobe osztályokban hitBy() függvényeinél átírás.
2019.03.02. 17:00	1.0 óra	Kiss	Még a driveban kiírt hibák egy részének javítása (Szekvenciák).
2019.03.02. 20:00	1.5 óra	Kis	Szekvenciadiagramok kijavítása (Animal falls, G.M. notifies, V.M. notifies)
2019.03.03. 18:00	1.5 óra	Kis	Osztálydiagram pontosítások (IPandaEffective interfész hozzáadás, kisebb hibák javítása)
2019.03.03. 19:00	0.5 óra	Martonosi	A dokumentum javítása Kiss által beszéltek alapján (Szekvenciák és osztályleírások)
2019.03.03. 20:00	2.0 óra	Kecskés, Kis, Kiss, Kobál, Martonosi	<b>Értekezlet:</b> Teljes dokumentum átvizsgálata. Közös munka az értekezlet alatt. Bármilyen helyben felmerült kérdés megbeszélése.
2019.03.03. 22:30	0.5 óra	Kecskés	Formázás, és a napló kiegészítése, a végleges dokumentum összeállítása.

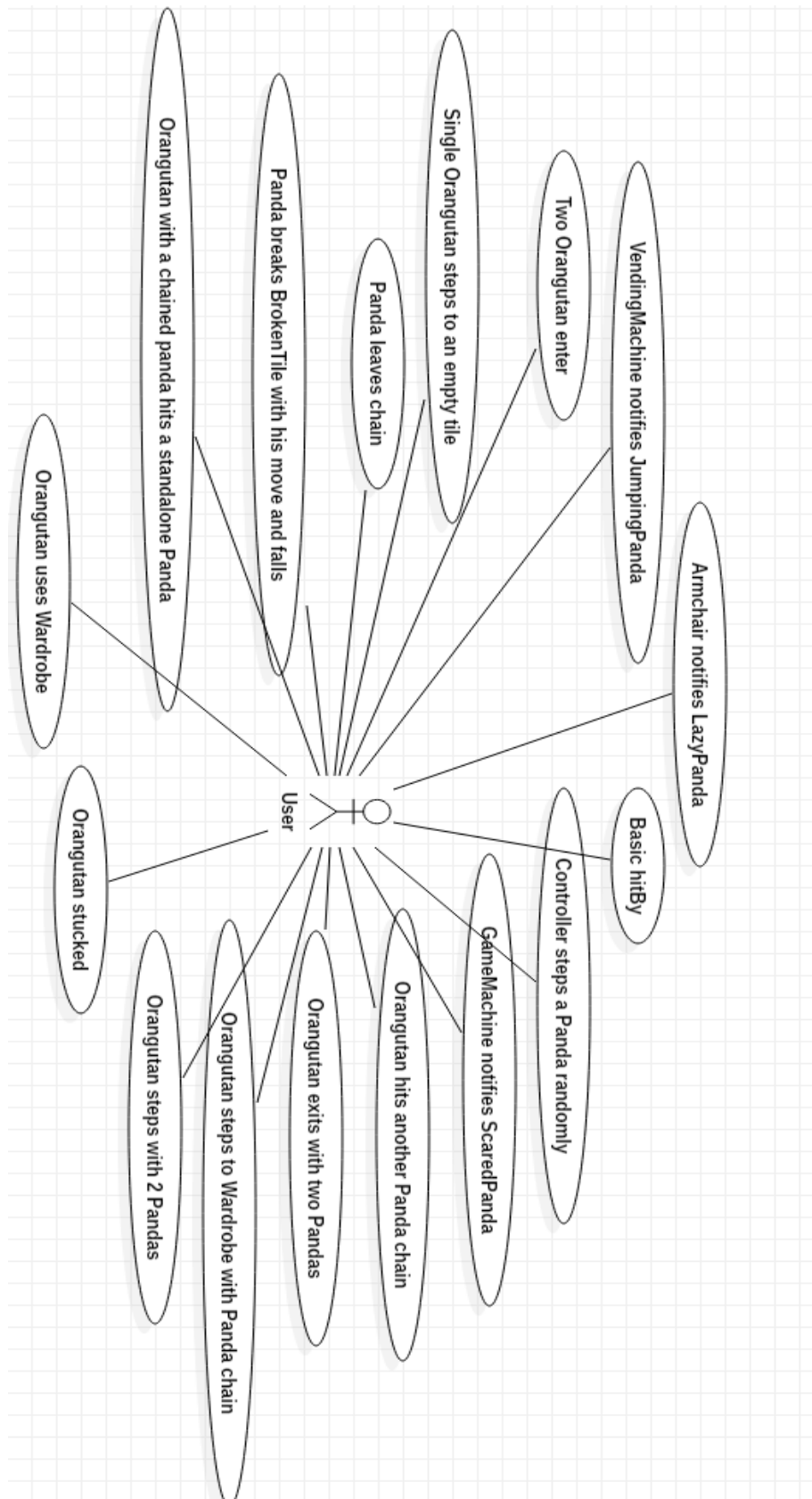
#### 4. Analízis modell tisztázása

A feladat javítva található fentebb, ezt adtuk be kétszer 3. és 4. Iterációképpen.

## 5. Szkeleton tervezése

### 5.1 A szkeleton modell valóságos use-case-ei

#### 5.1.1 Use-case diagram



### 5.1.2 Use-case leírások

<b>Use-case neve</b>	Armchair notifies LazyPanda
<b>Rövid leírás</b>	Lemodellezzük azt a folyamatot, hogy egy fotel szól egy fáradt pandának, hogy üljön belé.
<b>Aktorok</b>	User
<b>Forgatókönyv</b>	1. A program a képernyőre írja a lépéssel járó függvényhívásokat.

<b>Use-case neve</b>	Basic hitBy
<b>Rövid leírás</b>	Az AThing absztrakt őssosztályban implementált alapértelmezett hitby() függvényt hivatott szimulálni. Ez abban nyilvánul meg, hogy egy állat megpróbál rálépni egy olyan objektumra, amire nem léphet, így a helyén kell maradnia.
<b>Aktorok</b>	Controller, Player
<b>Forgatókönyv</b>	<p>1.A.1 A Controller lekérdezi az Orangutan alatt lévő csempétől a szomszédjait.</p> <p>1.A.2 A Controller átadja a szomszédok listáját a játékosnak.</p> <p>1.A.3 A Játékos kiválasztja a csempét, amire lépni szeretne.</p> <p>1.A.4 A csmpén egy VendingMachine helyezkedik el.</p> <p>1.A.5 Az Orangutan nem lép.</p>

<b>Use-case neve</b>	Controller steps a Panda randomly
<b>Rövid leírás</b>	A kör egy nélkülözhetetlen mozzanatát modellezi. Egy kör során a kontroller minden nem láncolt pandát egy véletlenszerűen sorsolt szomszédos csempére léptet(amennyiben ez lehetséges).
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	1. A konzol megjeleníti a függvényhívási fát, ami ahhoz kell, hogy a kontroller elintézzze a random szomszédos mező kiválasztását és elintézzze a lépés feladatát. A példát egy adott nem láncolt pandán mutatjuk be.

<b>Use-case neve</b>	GameMachine notifies ScaredPanda
<b>Rövid leírás</b>	A GameMachine és a ScaredPanda közti interakciót hivatott bemutatni.
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	<p>1.A.1 A Controller szól a GameMachine-nak, hogy csilingeljen</p> <p>1.A.2 A GameMachine sípol a szomszédos csempéken</p> <p>1.A.3 Az egyik szomszédos csempén van egy ScaredPanda.</p> <p>1.A.4 A ScaredPanda megijed.</p> <p>1.A.5 A ScaredPanda elengedi a sorban előtte lévő Panda-t.</p> <p>1.A.6 A ScaredPanda mögött lévő Panda-k elengedik egymást.</p>

<b>Use-case neve</b>	Orangutan exits with two Pandas
<b>Rövid leírás</b>	Lemodellezzük azt a folyamatot, hogy egy orángután kilép a kijáraton a háta mögött két pandával.
<b>Aktorok</b>	User
<b>Forgatókönyv</b>	1. A program a képernyőre írja a lépéssel járó függvényhívásokat.

<b>Use-case neve</b>	Orangutan hits another Panda chain
<b>Rövid leírás</b>	Az orángután elkapja egy másik panda lánc egyik panda tagját és befűzi maga mögé.
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	1. Megjelenik a függvényhívási fa, amely kezeli a panda megfogását és a helycserét, illetve az orángután panda láncába való befűzést.

<b>Use-case neve</b>	Orangutan steps to Wardrobe with Panda chain
<b>Rövid leírás</b>	Az orángután áthalad egy szekrényen egy mögötte lévő láncsal.
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	1. Megjelenik a függvényhívási fa, ami az orángután átlépteti a szekrény "túloldalára", ezt követően a lánc is követi az orángutánt, majd az orángután lép még egyet, hogy az első panda is kövesse őt a szekrényen túlra.

<b>Use-case neve</b>	Orangutan steps with 2 Pandas
<b>Rövid leírás</b>	Az orángután egy üres mezőre lép, miközben 2 panda van mögötte a láncban.
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	1. Megjelenik a függvényhívási fa, amely megoldja, hogy az orángután maga után húzza az egész láncot miután üres mezőre lépett.

<b>Use-case neve</b>	Orangutan stucked
<b>Rövid leírás</b>	Az orángután olyan helyzetbe kerül, hogy ahova lépni szeretne, oda a szabályok szerint nem lehetséges.
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	1. A konzol megjeleníti a függvényhívási fát, amely ahhoz szükséges, hogy a lépést megakadályozza, illetve, hogy a szabályoknak megfelelően az orángután elengedje a mögötte lévő panda láncot.

<b>Use-case neve</b>	Orangutan uses Wardrobe
<b>Rövid leírás</b>	Lemodellezzük azt a folyamatot, hogy egy orángután belép egy szekrénybe.
<b>Aktorok</b>	User
<b>Forgatókönyv</b>	1. A program a képernyőre írja a lépéssel járó függvényhívásokat.

<b>Use-case neve</b>	Orangutan with a chained panda hits a standalone Panda
<b>Rövid leírás</b>	Egy már létező lánc kibővítésének legegyszerűbb példája. A vezető orángutánt mindösszesen 1 panda követi. Az orángután a következő lépésével egy egyedülálló pandát fog bevenni a láncába.
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	1. A kimenet egy függvényhívási fa, amiben láthatóak a mozgást intéző illetve a láncban a previous/next kapcsolatokat beállító függvények.

<b>Use-case neve</b>	Panda breaks BrokenTile with his move and falls
<b>Rövid leírás</b>	Azt az eseményt modellezzük, amelyben a panda (akár orángután is lehetne) a lépésével eltör (=megsemmisít) egy BrokenTile típusú csempét és emiatt leesik.
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	1. A konzol megjeleníti a függvényhívási fát, amellyel elintézhető az eltörő csempére való lépéskor mind a csempe, mind a rálépő panda megsemmisítése. A példához adott lesz egy 1 élettartamú BrokenTile, amit a példában használt panda a lépésével eltör.

<b>Use-case neve</b>	Panda leaves chain
<b>Rövid leírás</b>	Azt szimulálja, ahogy a pandák elengedik egymást valaminek a hatására.
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	1.A.1 A Controller szól a Panda-nak, hogy lépjen ki a láncból. 1.A.2 A Panda elengedi az előtte lévő Panda-t. 1.A.3 A Panda szól a mögötte lévő Panda-nak, hogy lépjen ki a láncból.

<b>Use-case neve</b>	Single Orangutan steps to an empty tile
<b>Rövid leírás</b>	A folyamat során egy Orángután rálép egy, a szkeleton által meghatározott, üres mezőre. A jövőbeli játékban a csempét, amelyre az orángután lépne a User választja ki.
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	1. A konzol megjeleníti a függvényhívási fát, ami ahhoz kell, hogy egy magában álló orángután ráléphessen egy szomszédos üres csempére.

<b>Use-case neve</b>	Two Orangutan enter
----------------------	---------------------

<b>Rövid leírás</b>	Lemodellezzük azt a folyamatot, hogy egymás után két orángután belép a bejáraton.
<b>Aktorok</b>	User
<b>Forgatókönyv</b>	1. A program a képernyőre írja a lépésekkel járó függvényhívásokat.

<b>Use-case neve</b>	VendingMachine notifies JumpingPanda
<b>Rövid leírás</b>	A JumpingPanda interakcióját hivatott szimulálni egy VendingMachine mellett.
<b>Aktorok</b>	Controller
<b>Forgatókönyv</b>	1.A.1 A Controller szól a VendingMachine-nek, hogy sípoljon. 1.A.2 A VendingMachine sípol a mellette lévő csempéken. 1.A.3 A VendingMachine Tile-ja mindegyik szomszédján sípol. 1.A.4 A szomszédok között az egyiken van egy JumpingPanda. 1.A.5 A JumpingPanda ugrik 1.A.6 A JumpingPanda alati BrokenTile-nak csökken az élete.



## 5.2 A szkeleton kezelői felületének terve, dialógusok

A szkeleton egy konzolos alkalmazás lesz, amely indításkor kilistázza az egyes forgatókönyveket egy sorszámmal ellátva:

```
<1> Orangutan exits with 2 Pandas
<2> Orangutan with a chained Panda meets another Panda
      .
      .
      .
```

Végül a konzol kér egy sorszámot, amellyel “lefuttatható” az adott forgatókönyv (a fejlesztés jelen stádiumában ez mindösszesen a függvényhívások faszerű ábrázolását jelenti, törekedve a minél egyszerűbb összehasonlíthatóságra a szekvenciadiagramokkal).

Egy faszerkezet (=kimenet) lehetne a következő:

```
a.xxx()
  b.yyy()
  return
  c.zzz(d)
    d.www()
    return
  return
return
```

Magyarázat:

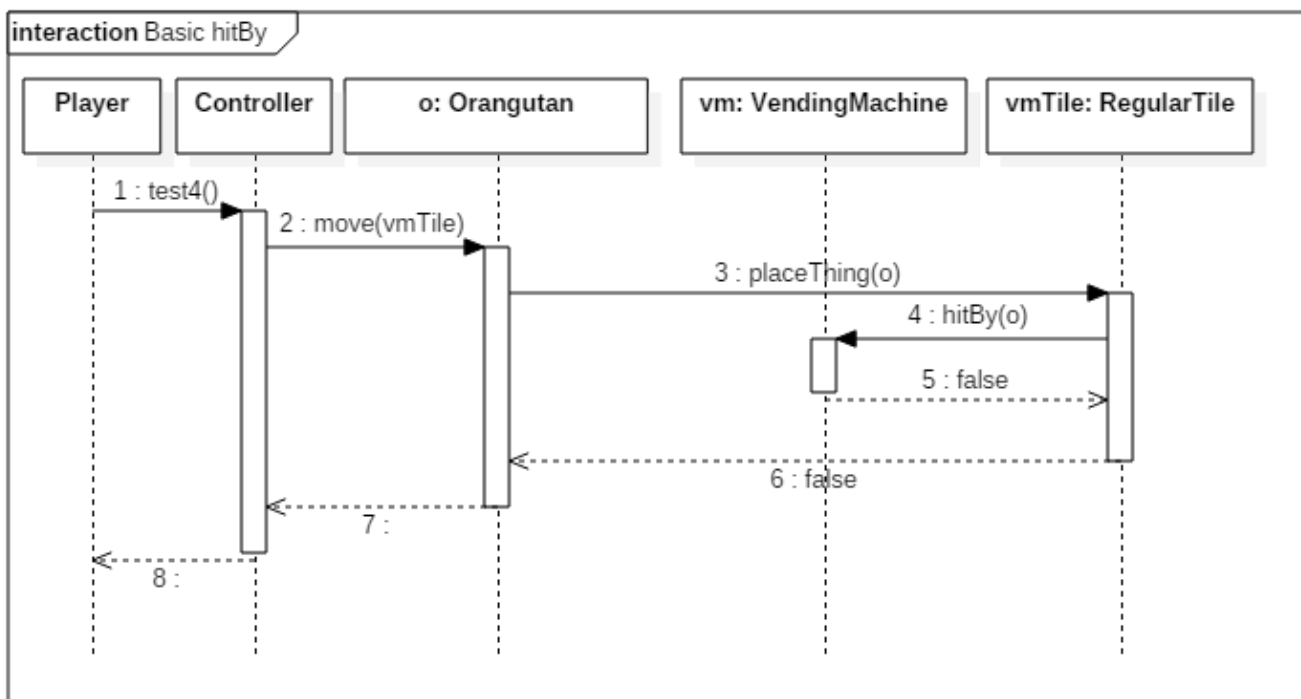
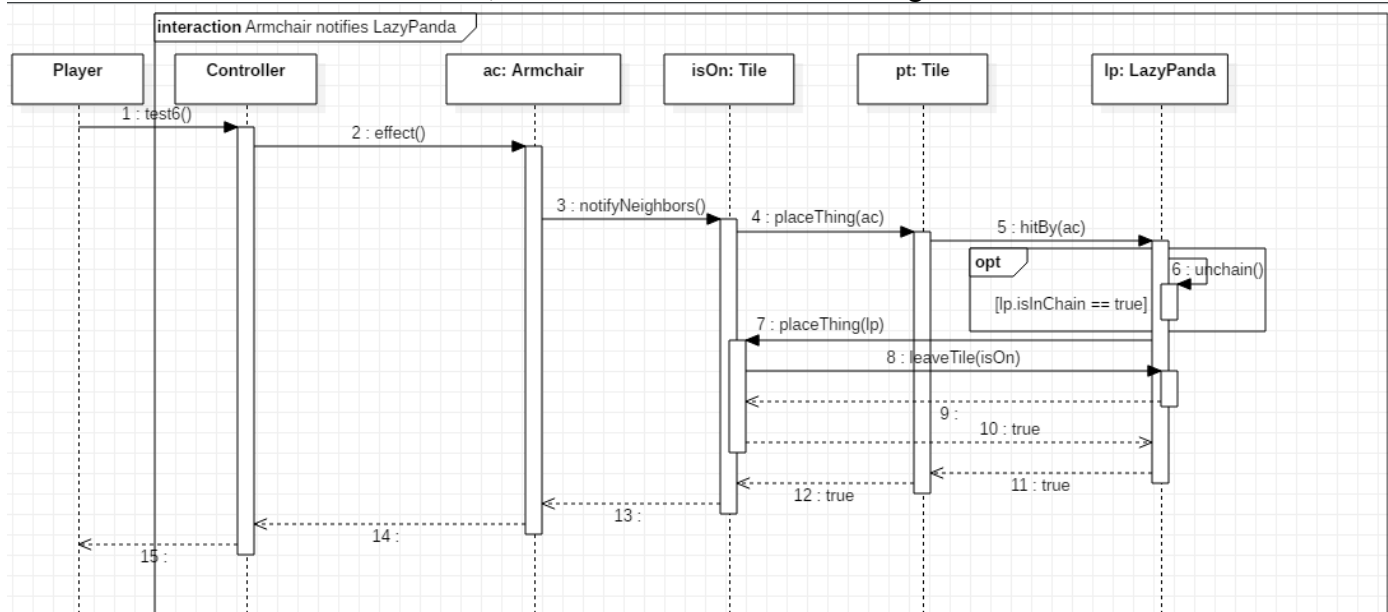
A program main függvénye meghívja az “a” objektum xxx() függvényét. Az xxx() függvény elér egy “b” objektumot, aminek meghívja az yyy() függvényét. Mivel ez további függvényhívásokat nem intéz, ezért visszatér belőle egy return utasítással. Továbbhaladva az xxx() függvényben meg van hívva a “c” objektumon a zzz() függvény “d” paraméterrel. Így pedig az előzőek alapján könnyen értelmezhető a kimenet hátralévő részei.

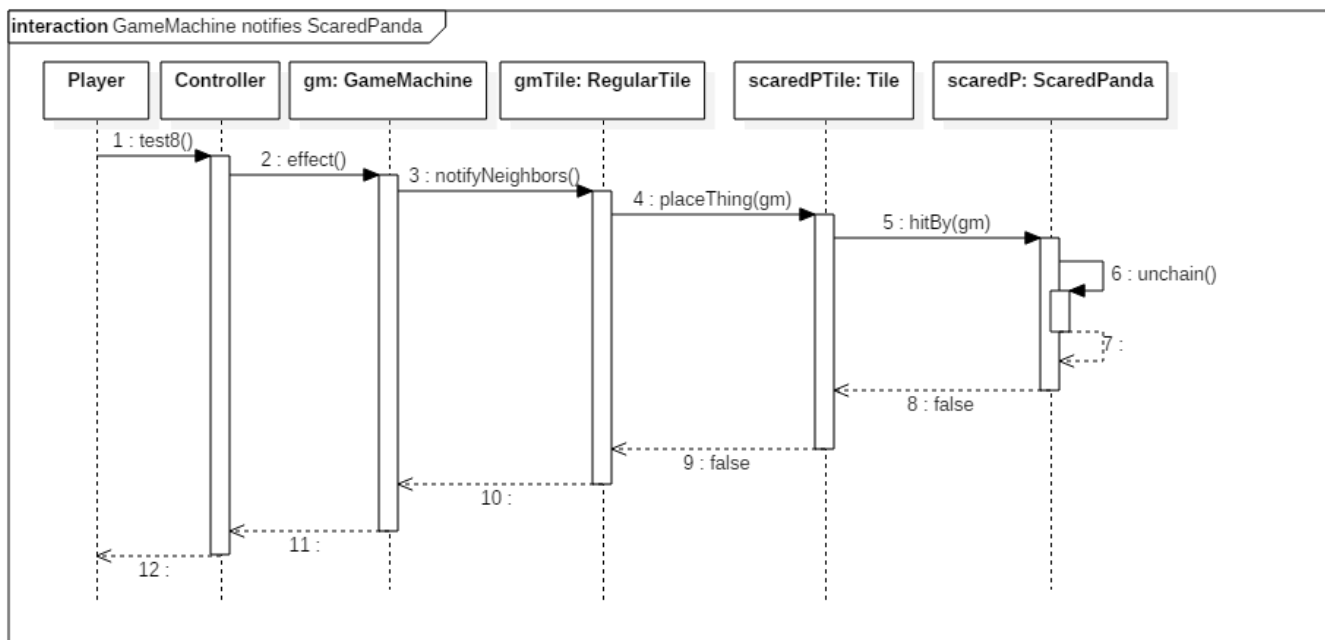
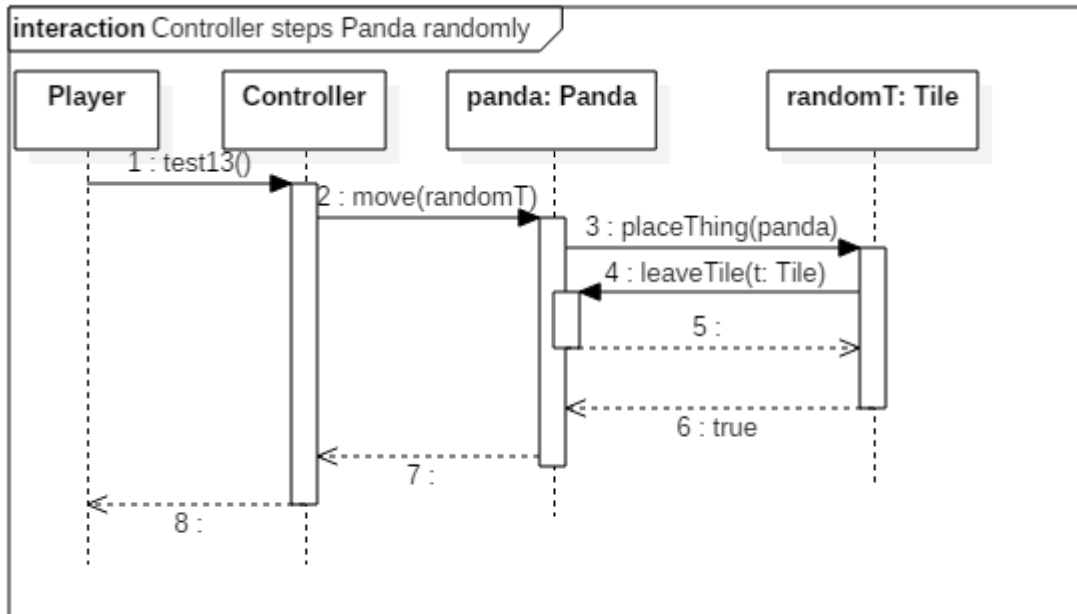
A kimenet kiírása után a program megkérdezi, hogy szeretne-e a felhasználó újabb forgatókönyvet futtatni <I/N>. Amennyiben igen, a konzol ismét kiírja a listát a lehetséges forgatókönyvekkel, ha nem akkor a program terminál.

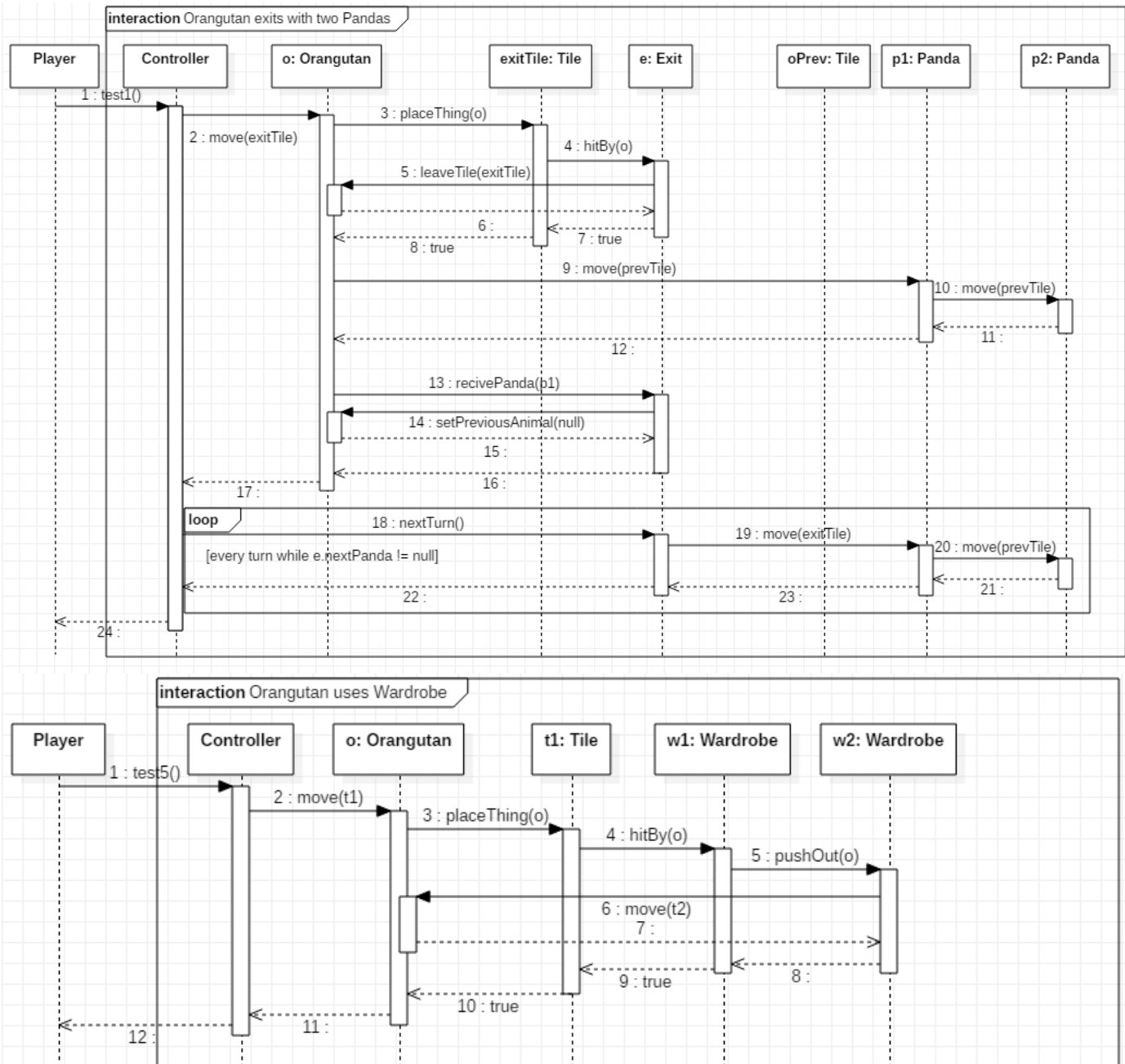
Ahhoz, hogy a szkeleton a fenti módon működhessen, el kell látni az egyes osztályokat egy public string name változóval, ami azt a nevet tárolja, amivel az adott objektumot példányosítva lett.

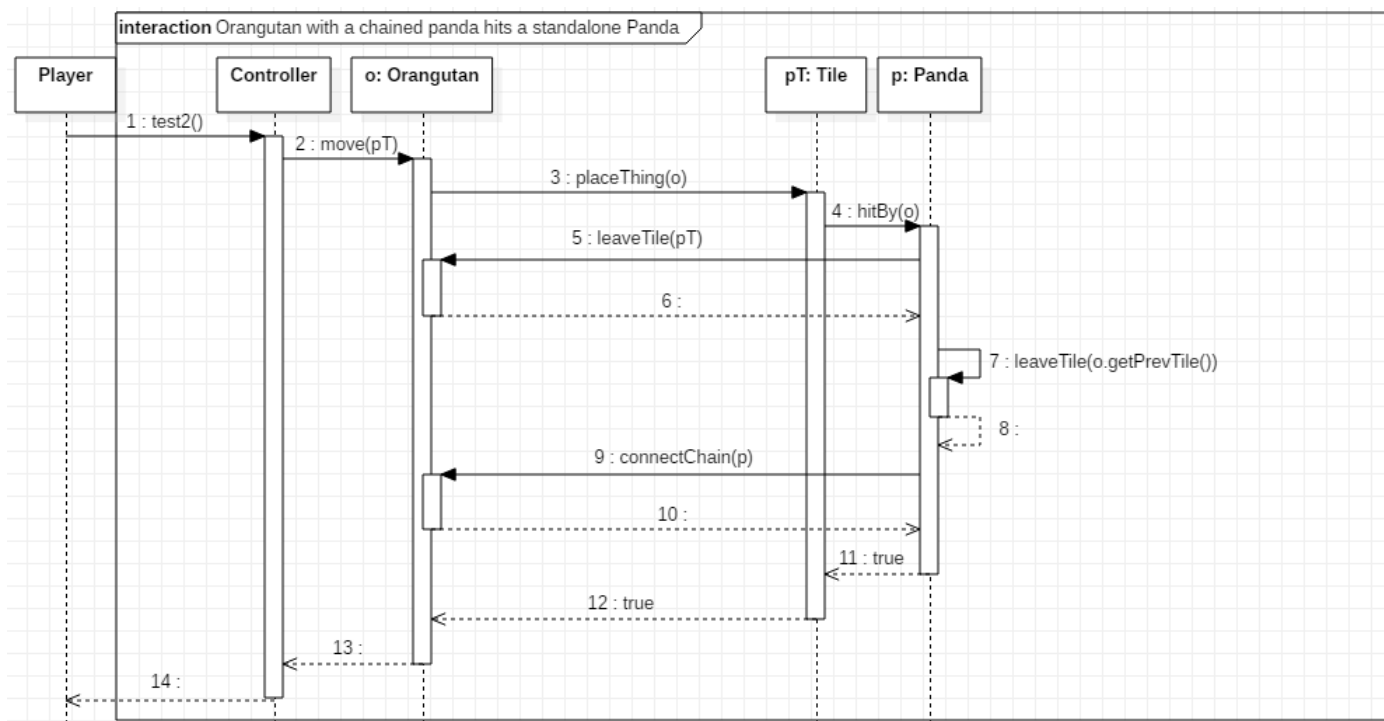
### 5.3 Szekvencia diagramok a belső működésre

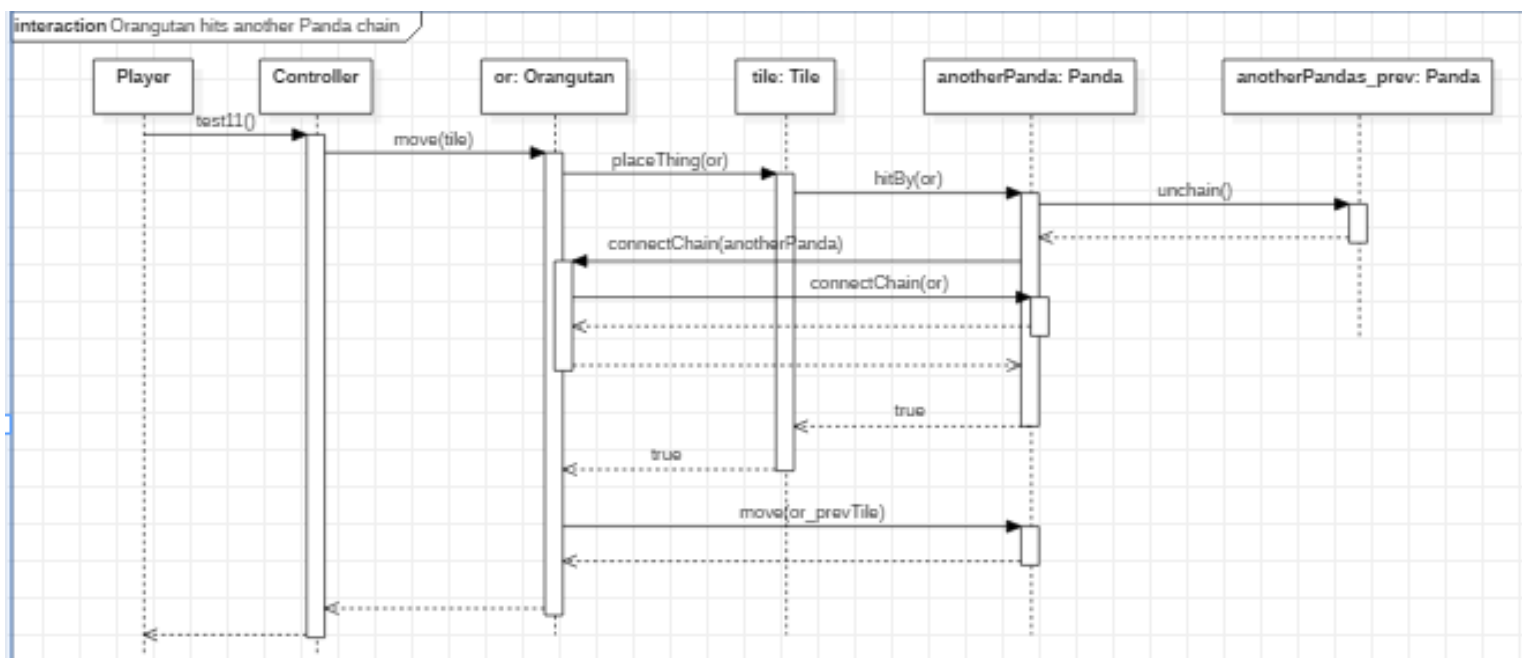
A pályákat pont akkorának vesszük, hogy elérjen rajtuk a teszt! A szekvencia diagramokat nem általánosan csináltuk, hanem az adott tesztesetnek megfelelően.

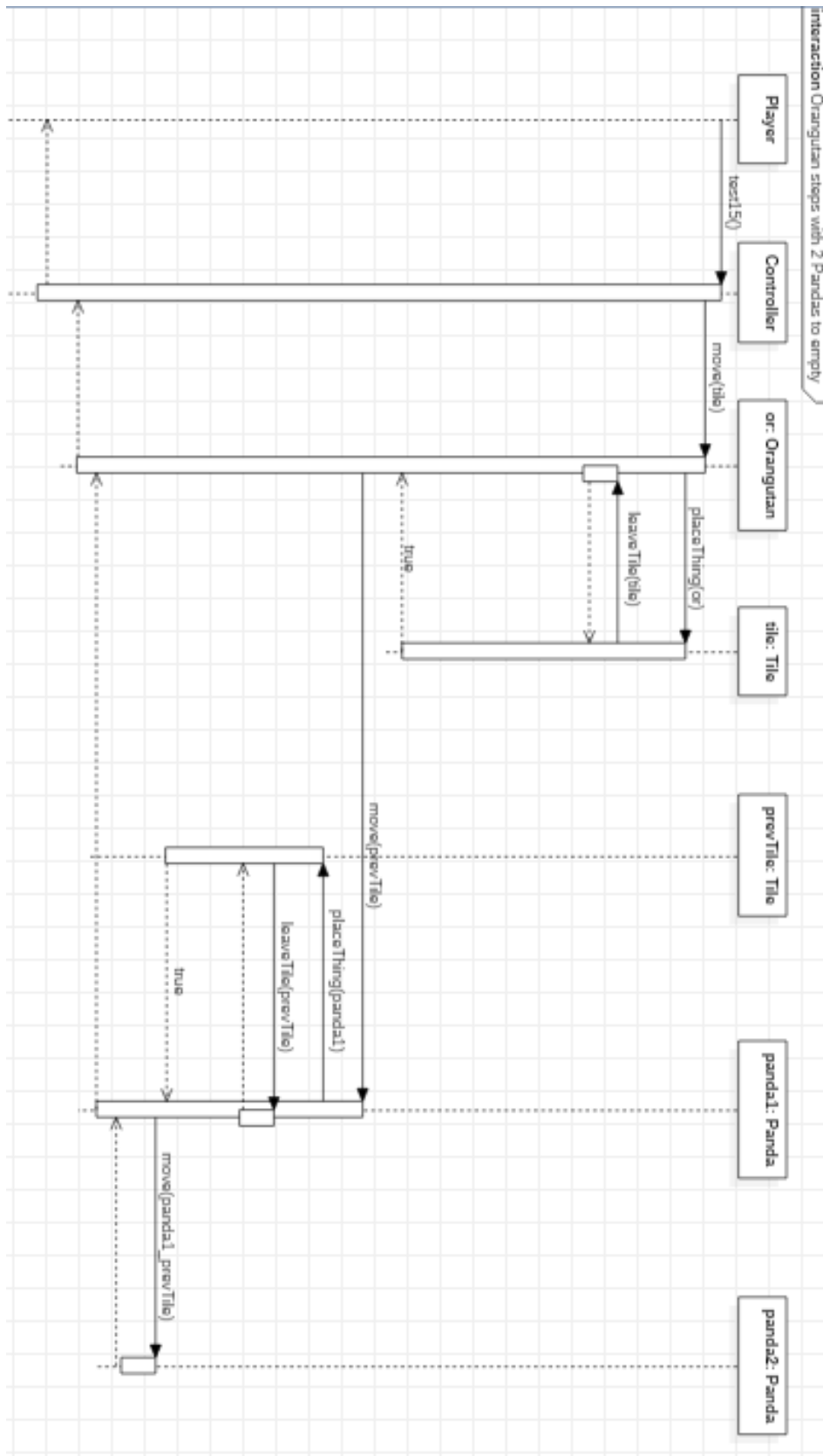






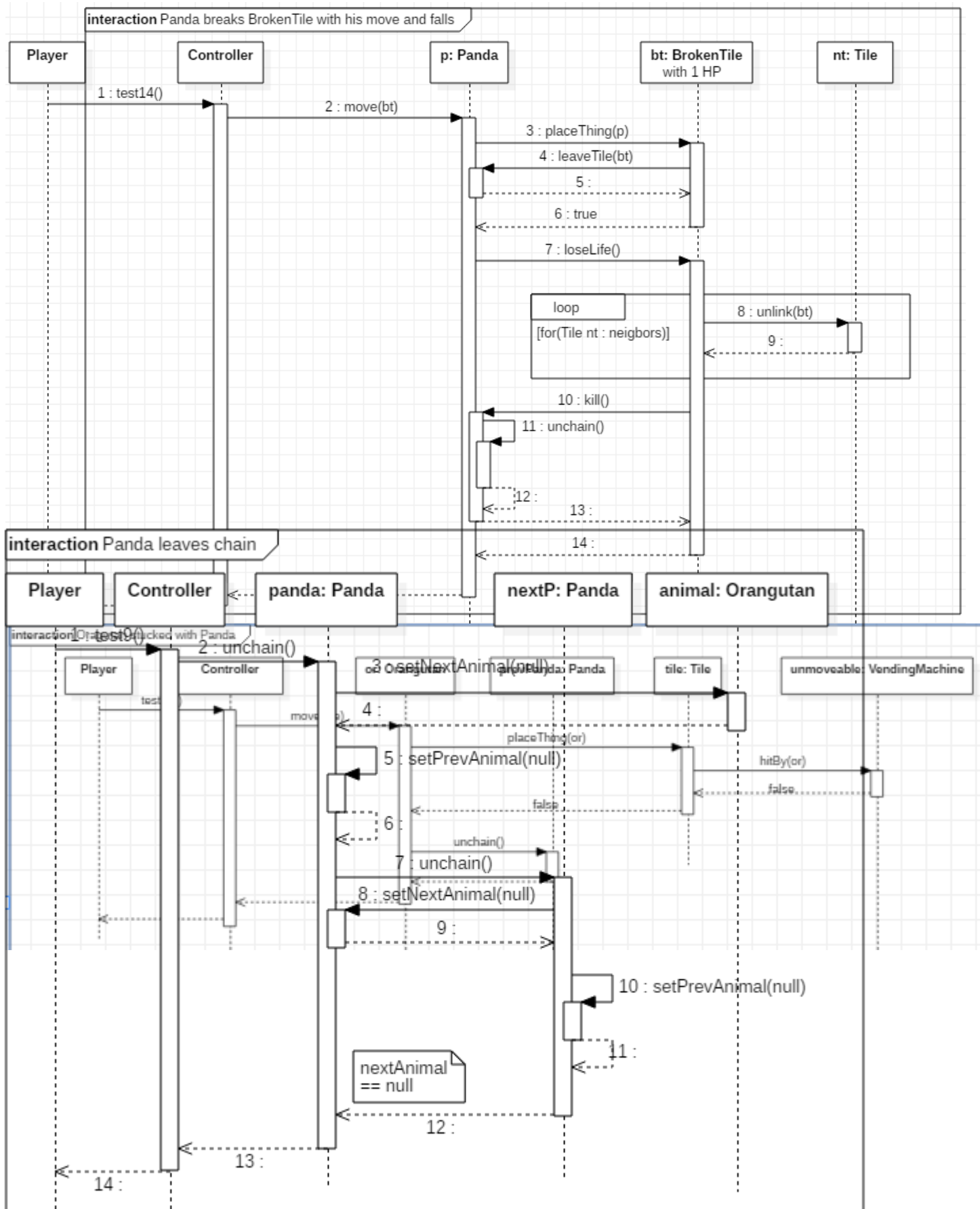


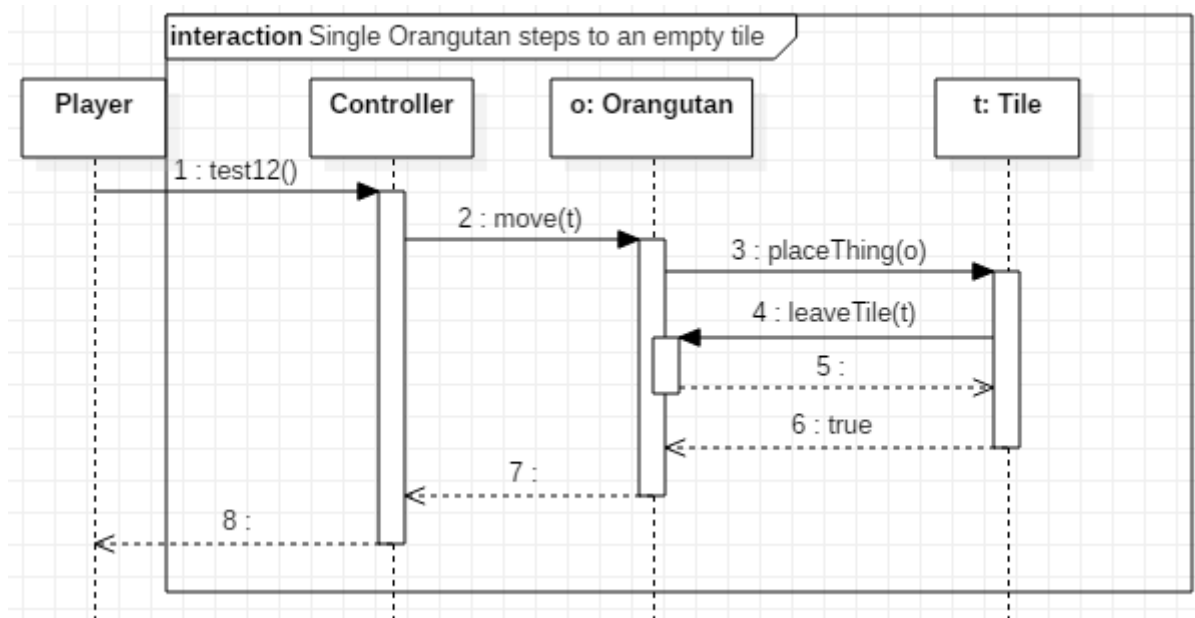


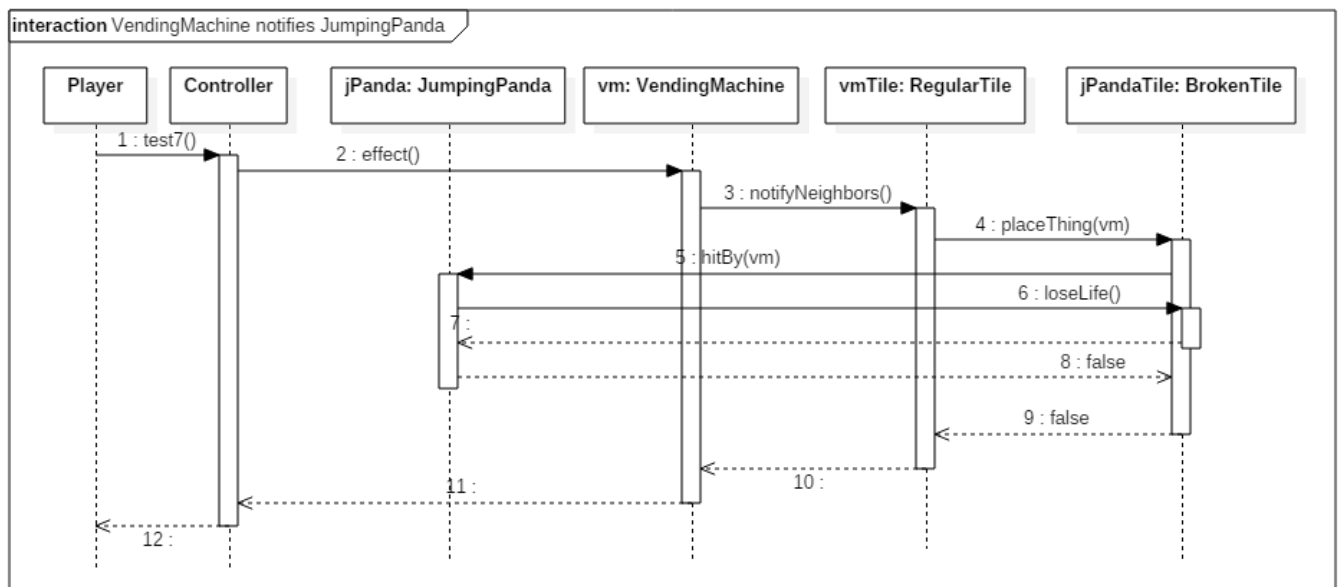
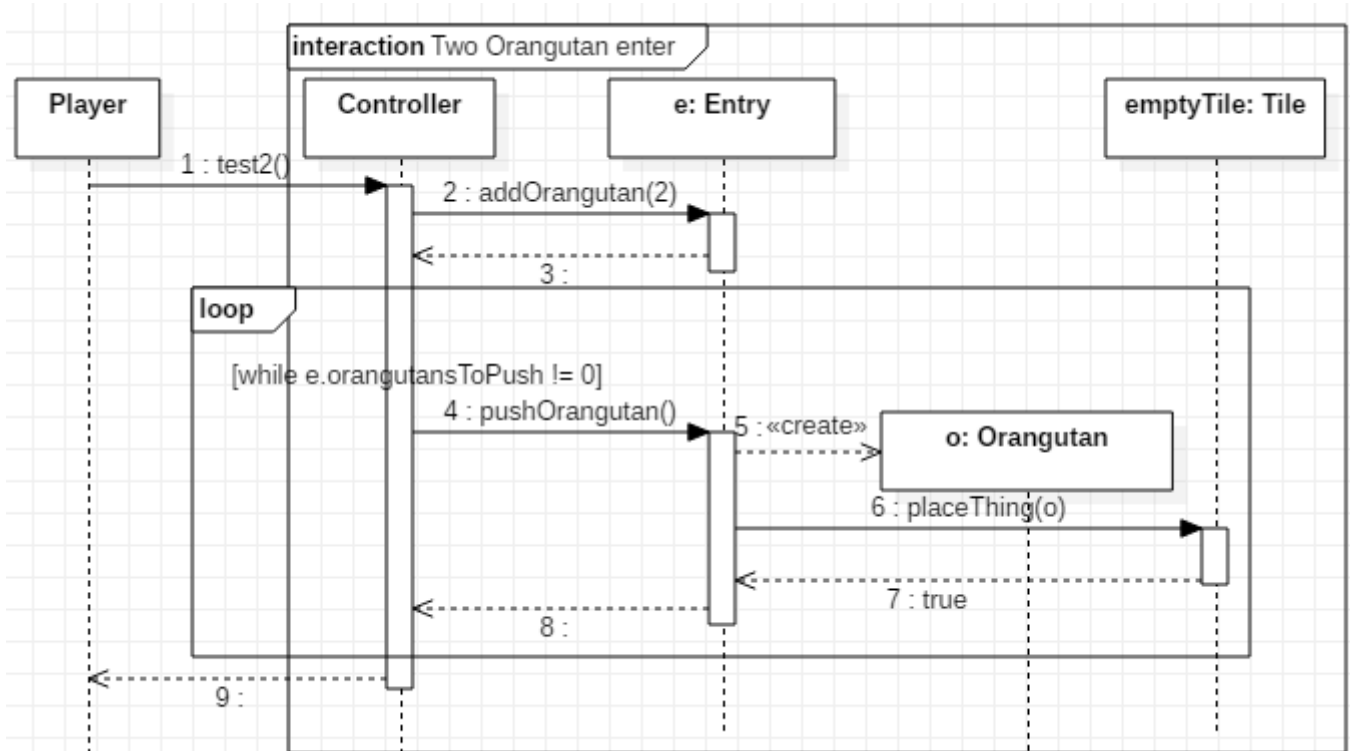




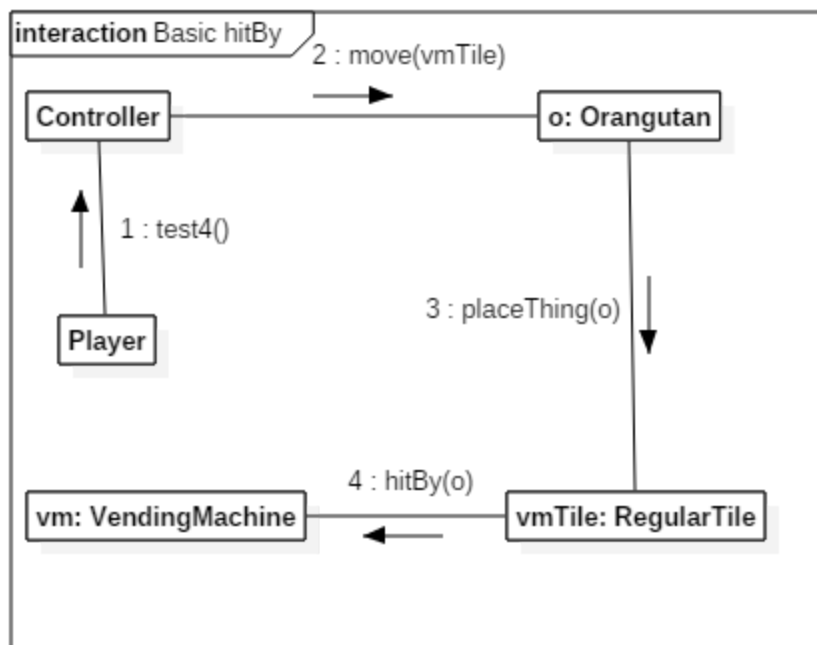
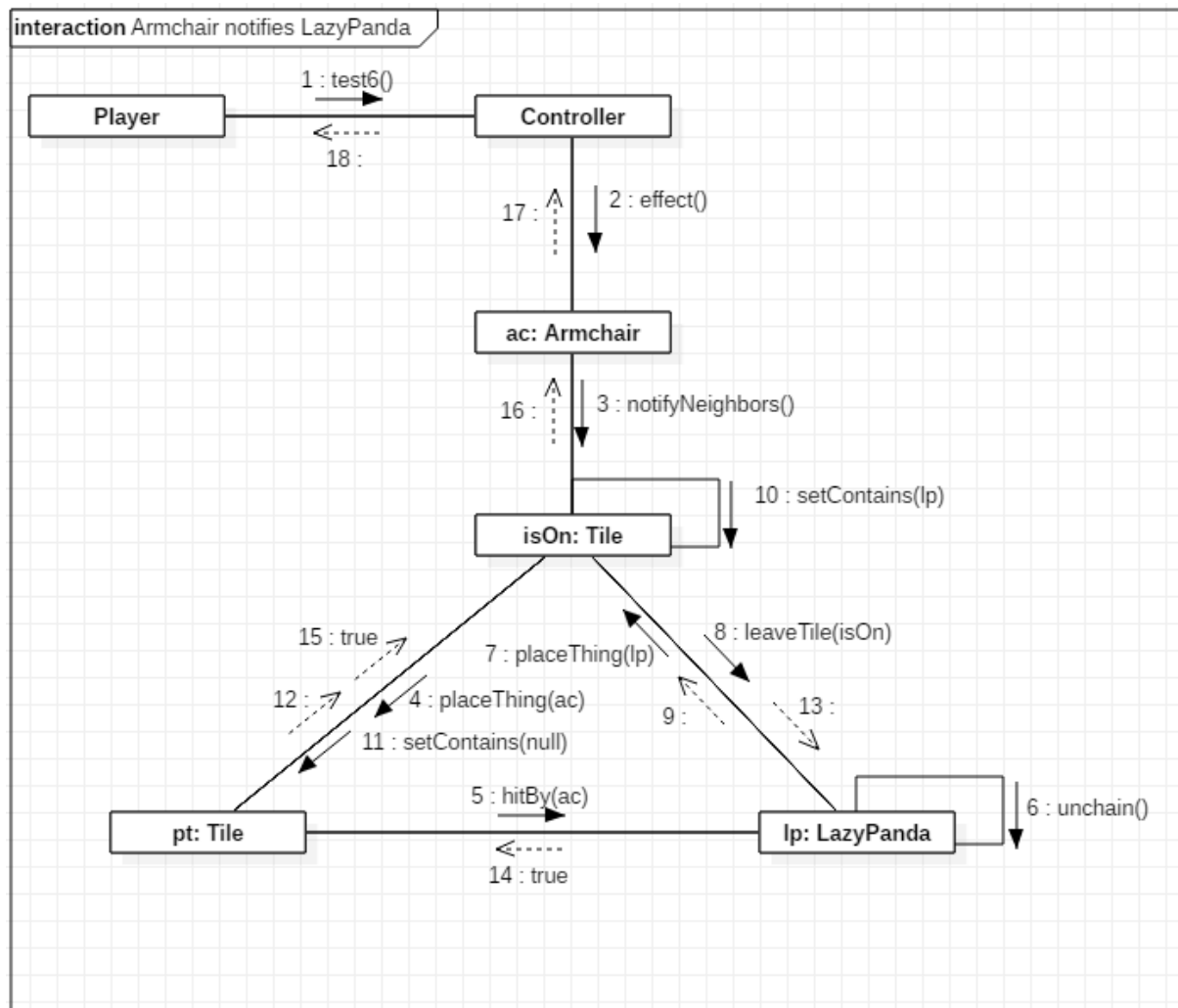


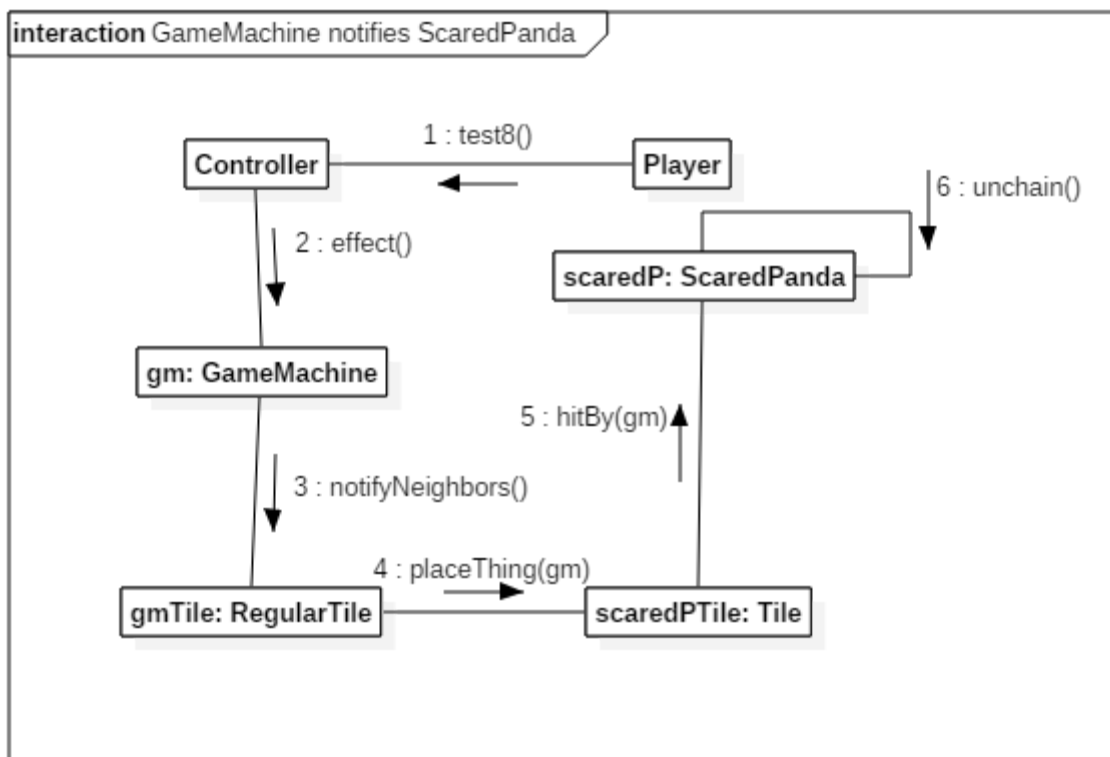
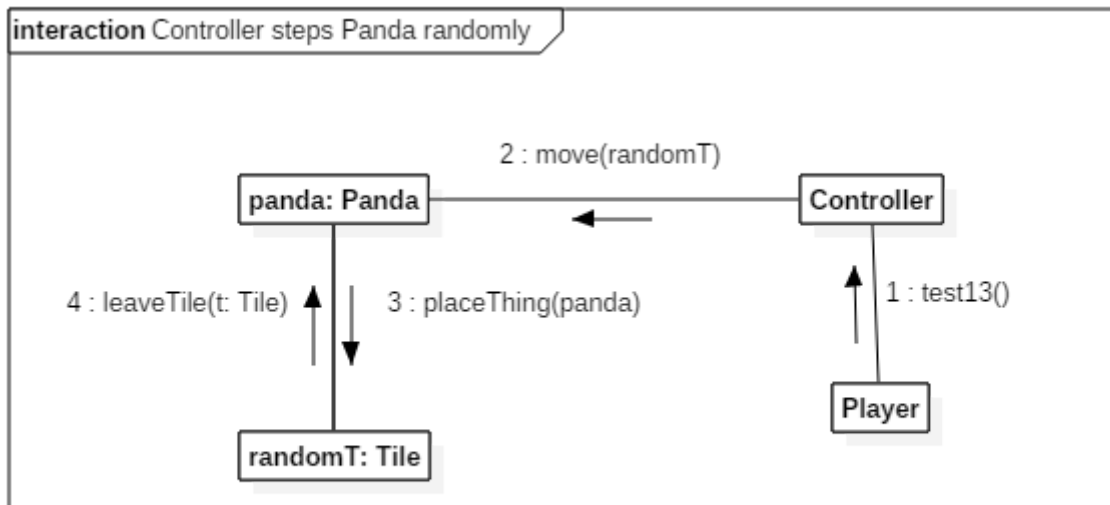


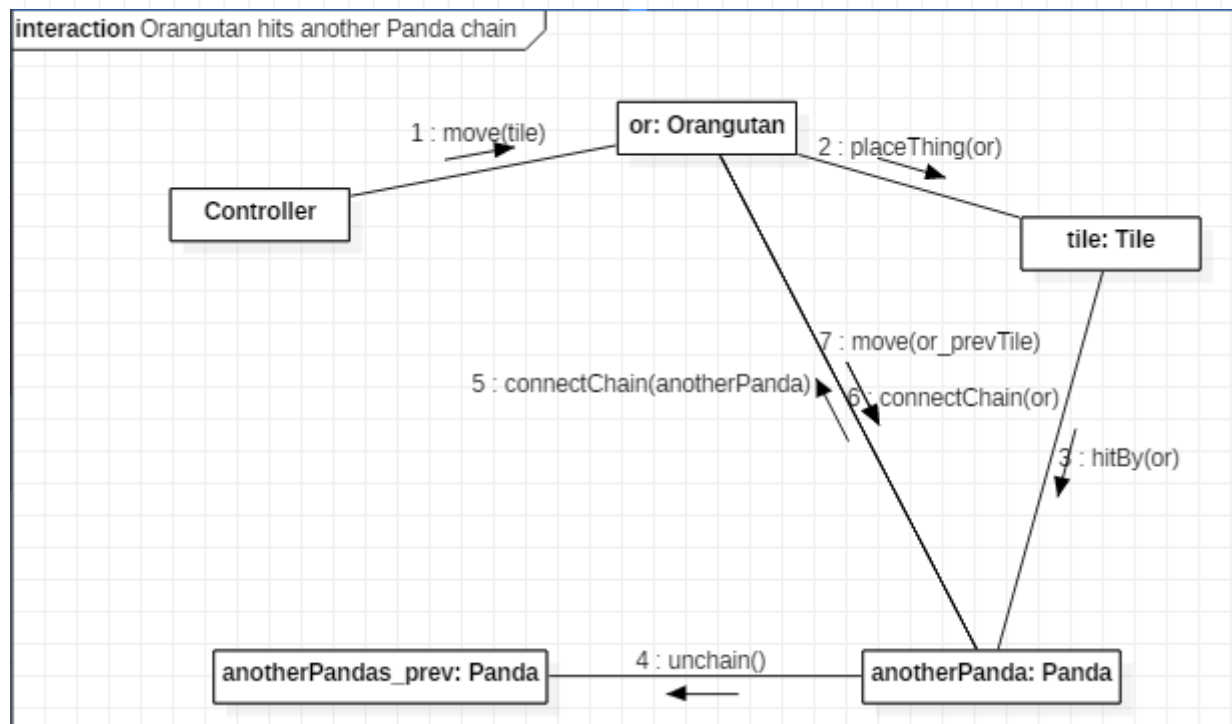
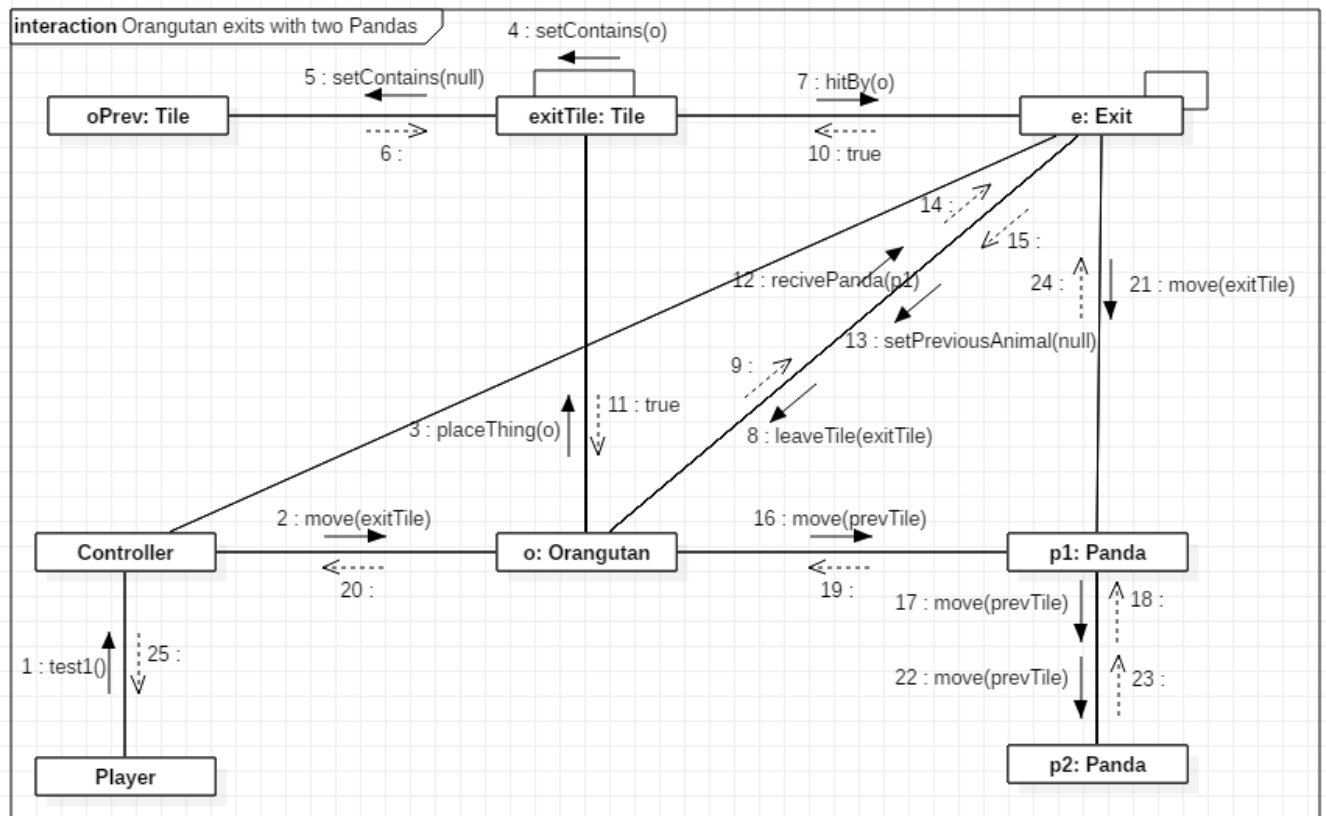


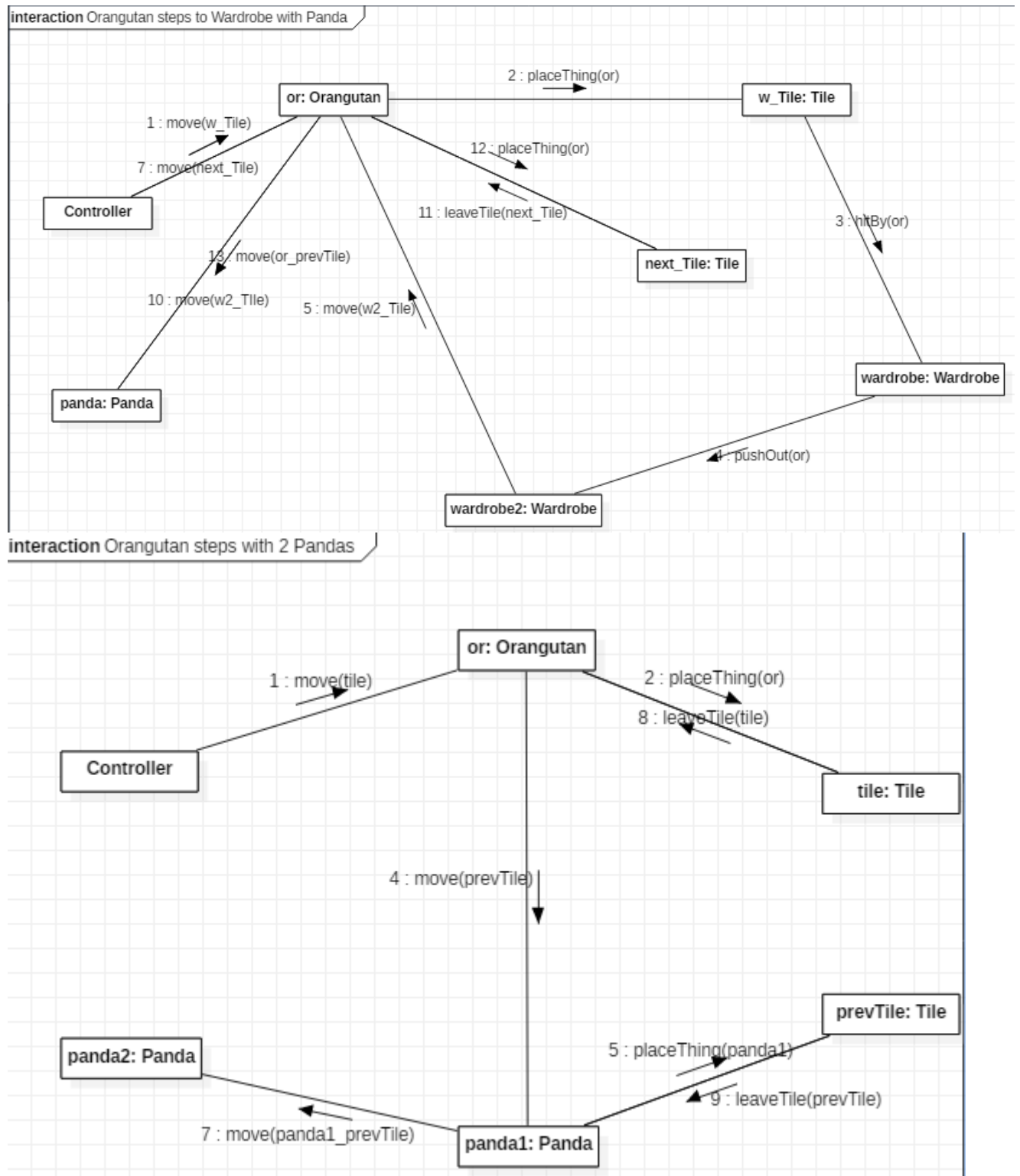


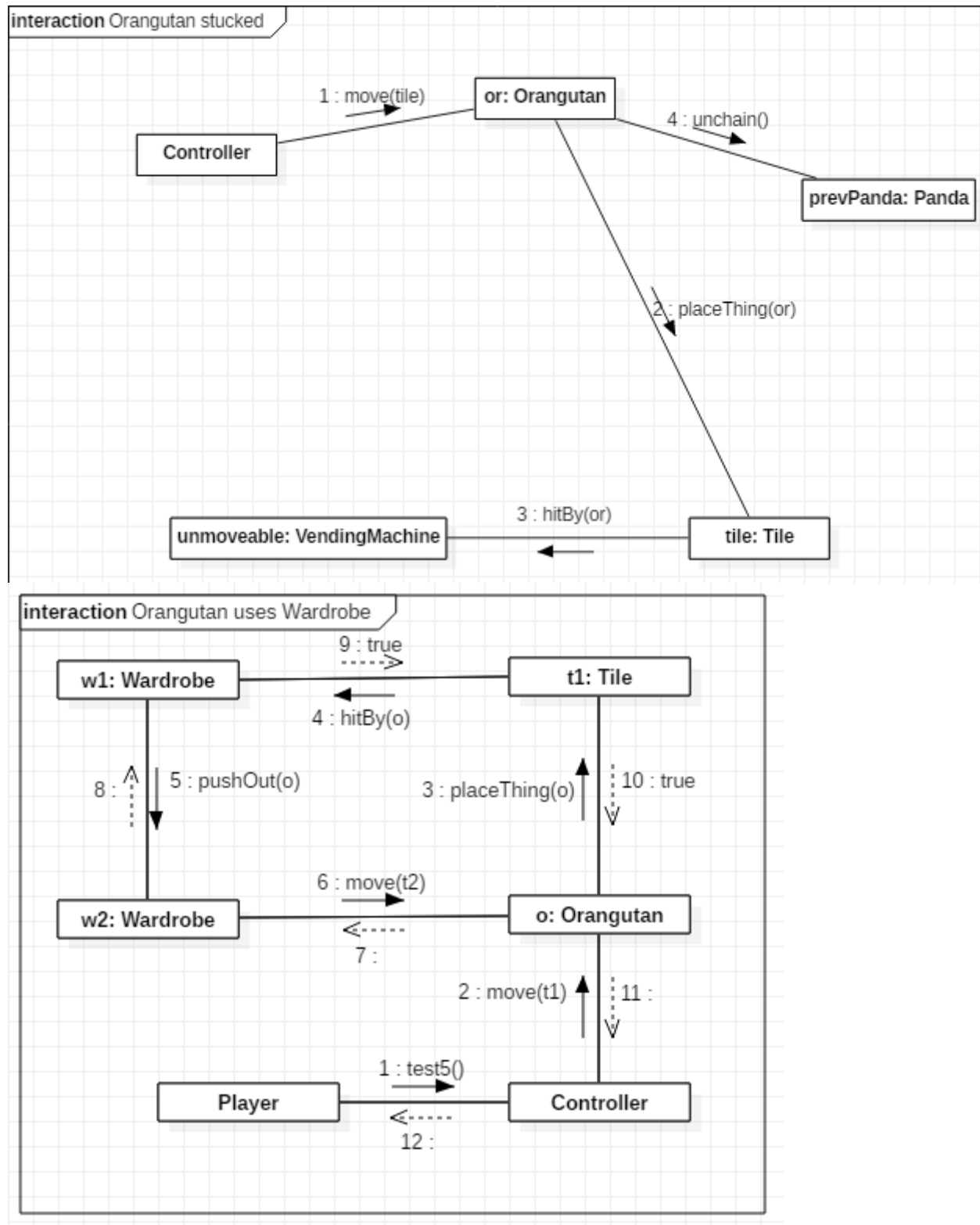
## 5.4 Kommunikációs diagramok



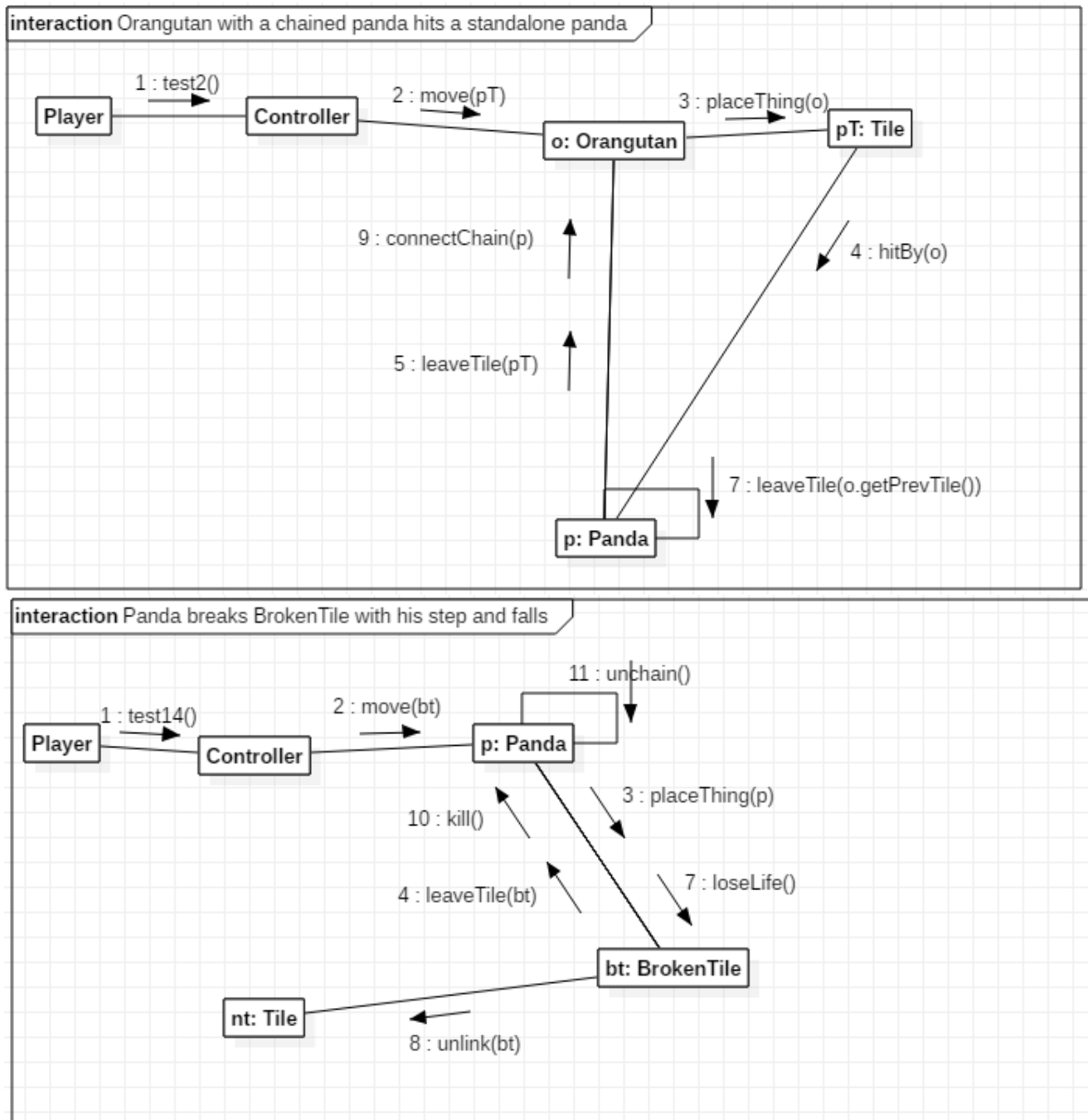


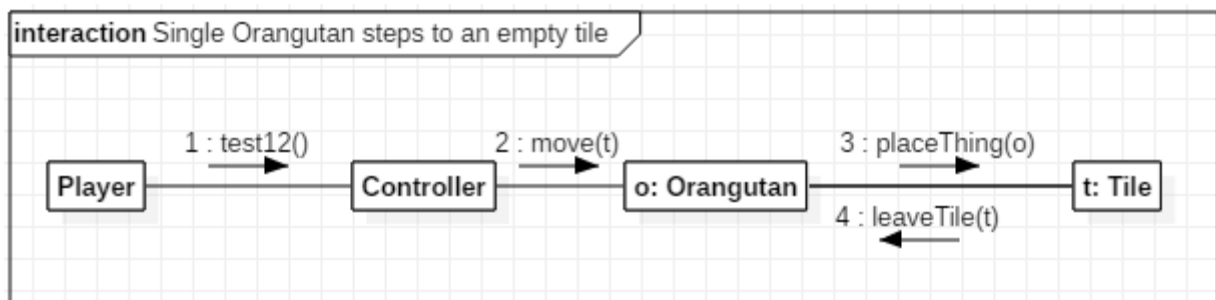
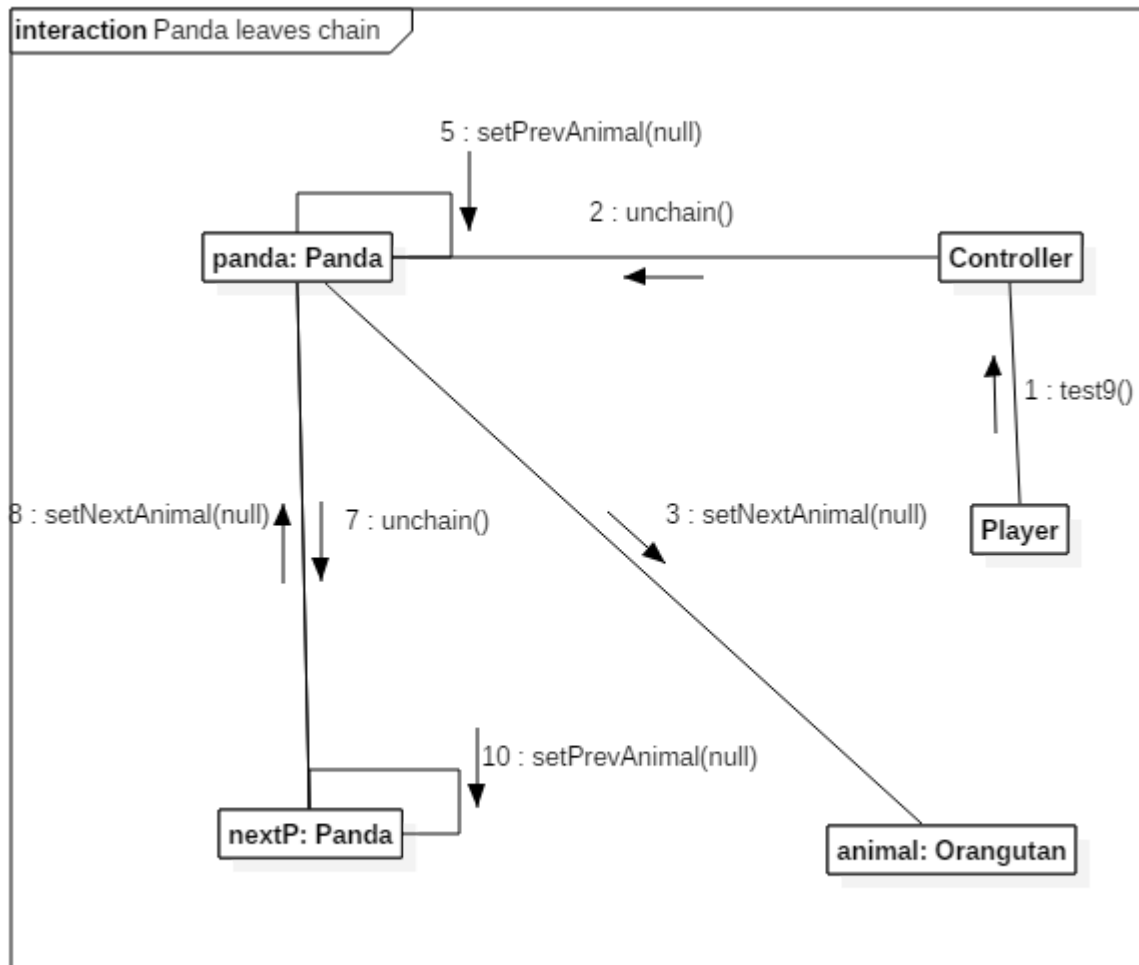


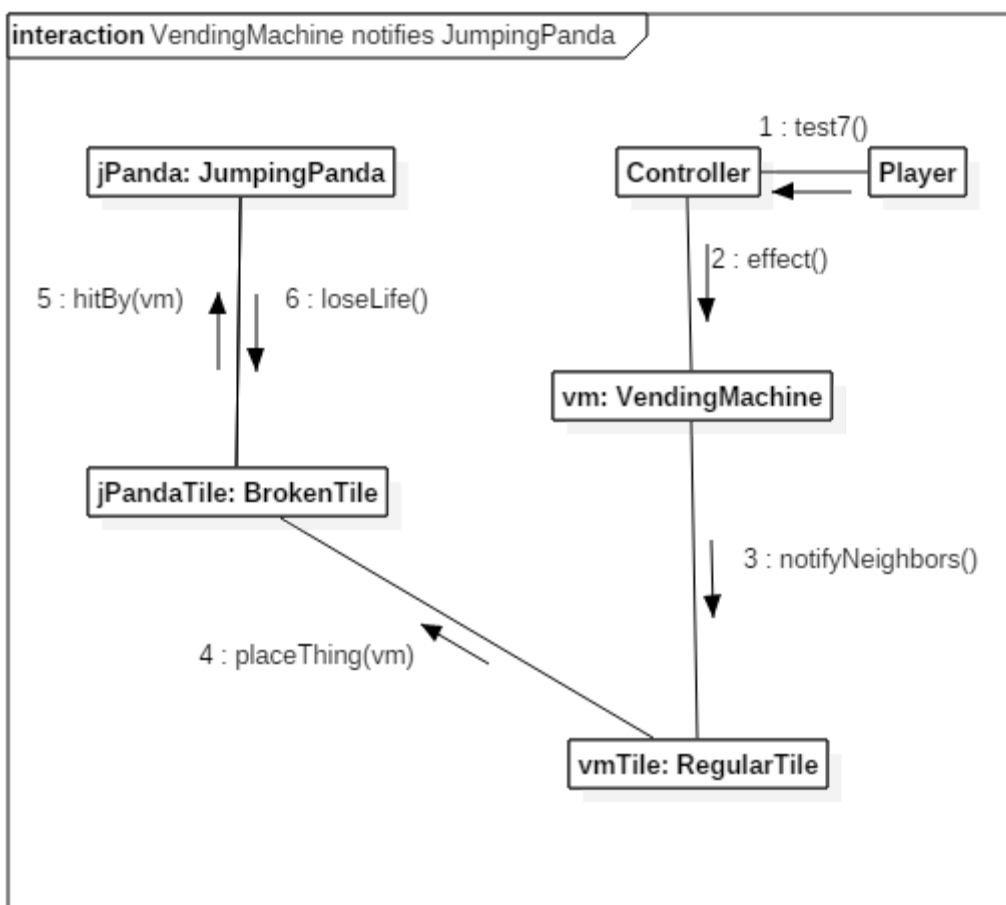
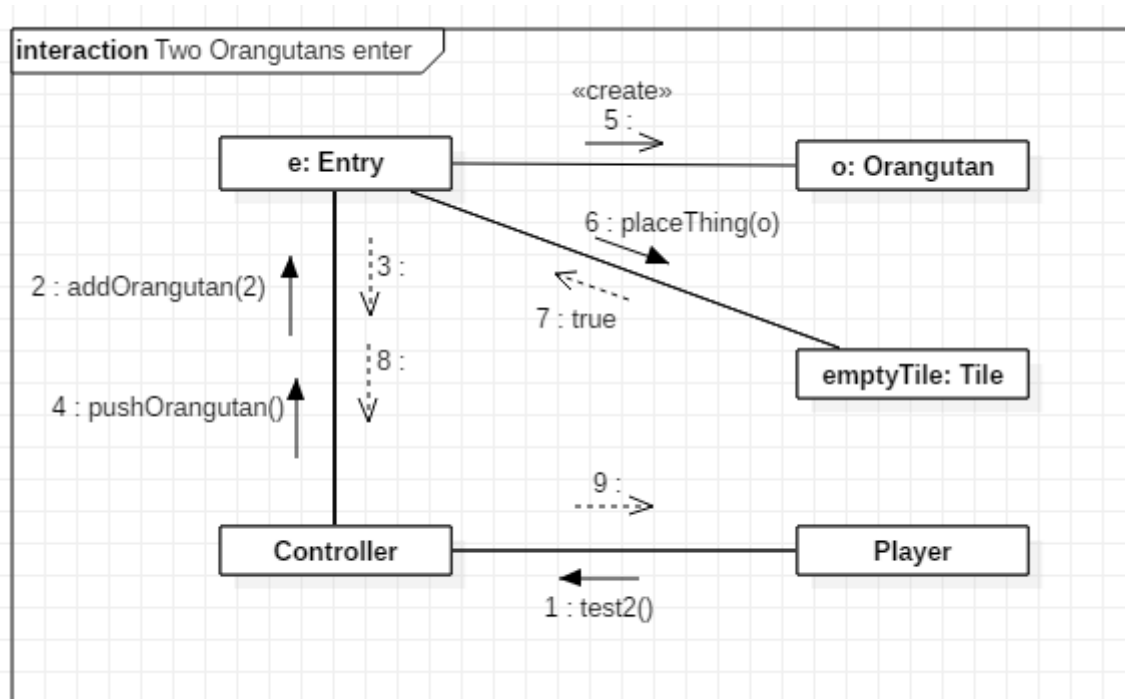












### 5.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2019.03.08. 20:00	1.5 óra	Kis	Konzolos felület megtervezése, a Call Tree-t kiíró singleton osztály megírása annak biztosítására, hogy a leírtak kivitelezhetők.
2019.03.08. 21:00	0.6 óra	Kis, Kiss, Kobál, Martonosi	<b>Értekezlet:</b> A szkeleton tervének elkészítéséhez szükséges feladatok felosztása egymás között.
2019.03.09. 11:00	1.5 óra	Kiss	Stucked Orangutan, Orangutan hit chain, Orangutan steps with 2 Pandas, és Orangutan steps to Wardrobe with Panda chain szekvenciák megcsinálása
2019.03.09. 13:30	1.0 óra	Korbál	Skeleton szekvenciadiagramok elkészítése (saját rész)
2019.03.09. 14:00	1.5 óra	Martonosi	Skeleton terveben kiosztott szekvenciadiagrammok elkészítésének befejezése, korábbi diagrammok javítása
2019.03.09. 15:30	1.0 óra	Korbál	Usecase-ek leírása és kommunikációs diagramok elkészítése (saját rész)
2019.03.09. 16:00	2.0 óra	Kis	Szekvenciadiagrammok: 2, 12, 13, 14 plusz ezekhez tartozó komm. diagramok
2019.03.09. 20:30	1.5 óra	Kiss, Kobál, Martonosi	<b>Értekezlet:</b> Elkészült szekvencia diagramok átnézése, javítások javaslata, értekezlet megbeszélve Mar 10. 18:00 kezdettel.
2019.03.10. 10:30	0.5 óra	Martonosi	Szekvenciadiagrammok javítása, kommunikációs diagrammok elkészítése
2019.03.10. 12:00	1.5 óra	Kiss	Szekvencia diagrammokhoz tartozó kommunikációs diagrammok megszerkesztése, és megbeszélte hibák javítása.
2019.03.10. 12:00	2.0 óra	Kis	Szekvenciadiagrammok: 2, 12, 13, 14 kijavítása, konkretizálása az adott forgatókönyvre
2019.03.10. 18:00	1.0 óra	Kecskés, Kis, Kobál, Martonosi	<b>Értekezlet:</b> Maradék diagramok átnézése, és minden tesztet végéigvétele
2019.03.10. 20:00	1.0 óra	Kecskés	Formázás, és a napló kiegészítése, a végleges dokumentum összeállítása.

## 6. Szkeleton beadás

### 6.1 Fordítási és futtatási útmutató

#### 6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Animal.java	2196	2019.03.24 16:47	Az állatokat gyűjtő őszosztály
Armchair.java	1339	2019.03.24 16:47	Karosszék függvényeit tároló
AThing.java	1804	2019.03.24 16:47	Minden tárgy őszosztálya
BrokenTile.java	1150	2019.03.24 16:47	Törött csempe osztálya
Controller.java	433	2019.03.24 16:47	A pandát irányítja (és később a pályát állítja össze)
Entry.java	1551	2019.03.24 16:47	Bejárat osztálya
Exit.java	1875	2019.03.24 16:47	Kijárat osztálya
GameMachine.java	1394	2019.03.24 16:47	Játékgép osztálya
IPandaEffective.java	203	2019.03.24 16:47	Az hatással rendelkező (szomszéd csempét értesítő effekteket tartalmazó) osztályok interface-e
JumpingPanda.java	641	2019.03.24 16:47	Ugráló panda osztálya
LazyPanda.java	2349	2019.03.24 16:47	Lusta panda osztálya
Main.java	13900	2019.03.24 16:47	Fő osztály, tesztesetekkel
Orangutan.java	1585	2019.03.24 16:47	Orángután osztálya
Panda.java	2590	2019.03.24 16:47	Panda osztálya
Printer.java	932	2019.03.24 16:47	A konzolra függvényeket író osztály és benne található függvények
RegularTile.java	170	2019.03.24 16:47	Nem törős csempe osztálya
ScaredPanda.java	510	2019.03.24 16:47	Ijedős panda osztálya
Tile.java	6513	2019.03.24 16:47	Csempe őszosztálya
VendingMachine.java	1404	2019.03.24 16:47	Játékautomata osztálya
Wardrobe.java	2178	2019.03.24 16:47	szekrény osztálya

#### 6.1.2 Fordítás

A program futtatásához lent kell, hogy legyen a JDK (JavaSE1.8) környezet és be kell, hogy legyen építve a környezetbe.

(Windows-vezérlőpult-rendszer-speciális rendszerbeállítások-környezeti változók – Path – szerkesztés - JDK mappájának hozzáadása a mezőbe)

[https://github.com/alexbobal/projlab\\_smooth\\_set](https://github.com/alexbobal/projlab_smooth_set)

A forráskódot a github linken zip-ből kicsomagolva az src mappában találjuk.

A futtatáshoz szükséges parancsokat ebben a mappában a következő módon kell csinálni:

```
javac Main.java
java Main
```

#### 6.1.3 Futtatás

A program indítása után egy jól leírt interface látható, amiben tesztelni úgy lehet, hogy a teszt eset számát beírjuk a console-ba (enterrel zárjuk). Ha újabb teszt esetet szeretnénk végrehajtani, akkor a kiválasztott teszt eset lefutása után felkínálja a program ezt. Ha nem akarunk többet futtatni, akkor pedig kilép szabályosan.

pl.:

14

i

2

n

## 6.2 Értékelés

A százalék az első iterációtól kezdve él. Összesen 63.5 óra egyéni munkát könyveltem el arányaiban.

Tag neve	Tag neptun	Munka százalékban
Kecskés Péter	XZ9I7F	19
Kis Máté Levente	H0OTLX	21,5
Kiss Előd	U479JN	19,5
Kobál Sándor	C4W3BP	21
Martonosi Máté	QOT4DC	19

## 6.3 Napló

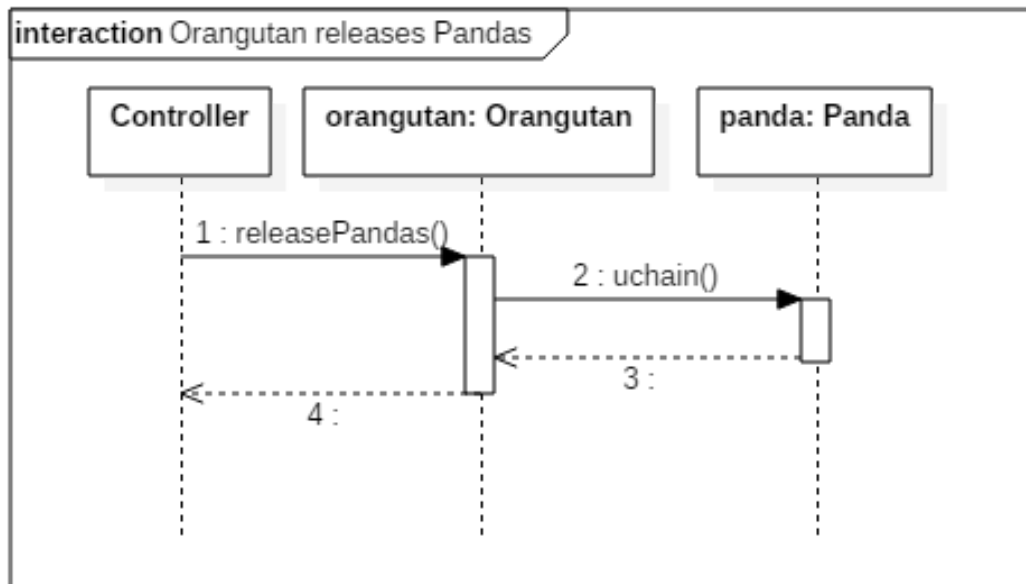
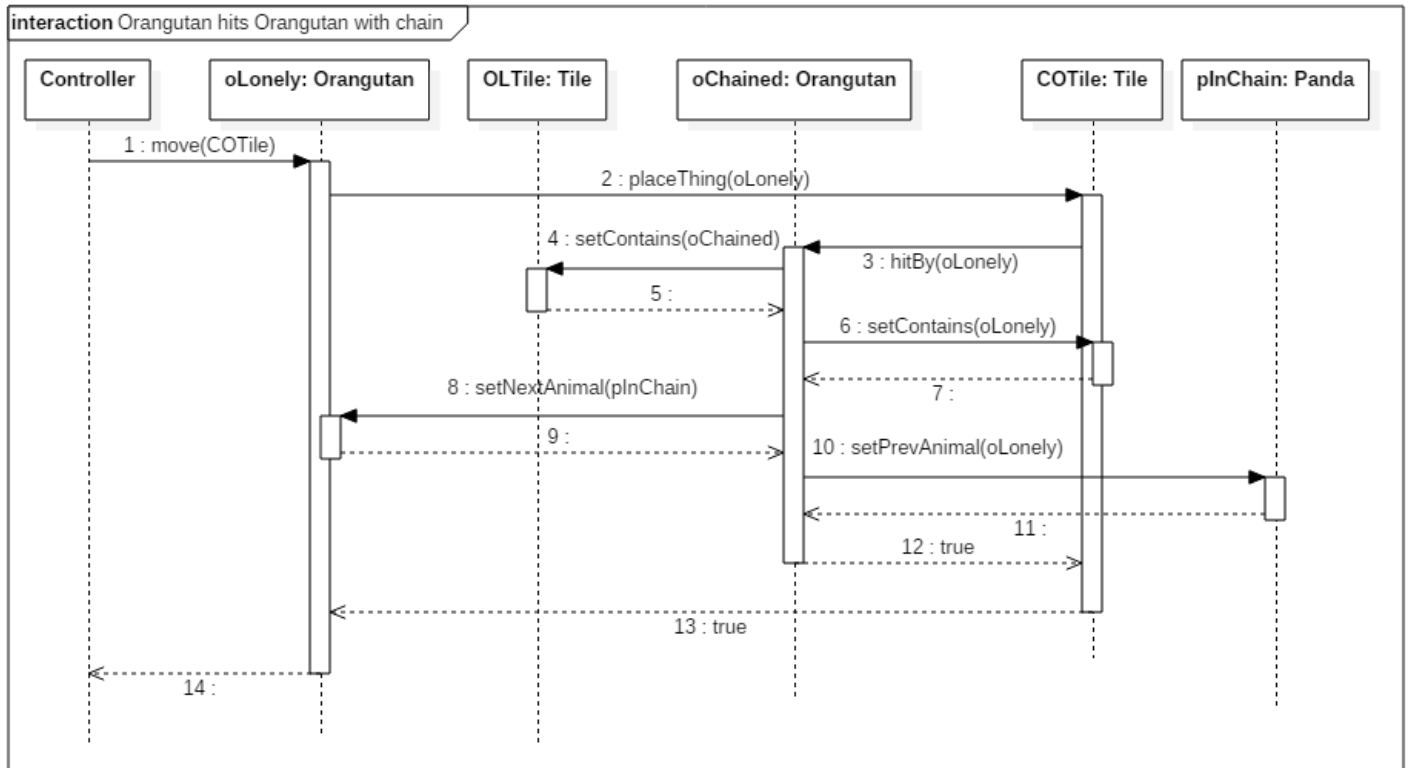
Kezdet	Időtartam	Résztevők	Leírás
2019.03.15 18:10	0.2 óra	Kis, Kobál, Martonosi	<b>Értekezlet:</b> Feladatok gyors (megvalósítandó függvények/osztályok) kiosztása hétfővégre.
2019.03.16 16:00	2.0 óra	Kiss	Az Orangutan, Panda, LazyPanda, és Animal osztályok implementálása.
2019.03.16 18:00	1.5 óra	Kobál	Wardrobe, Entry és az Exit osztályok implementálása.
2019.03.18 10:00	1.0 óra	Martonosi	Osztályok elkészítése
2019.03.18 16:00	1.5 óra	Kecskés	Eclipse github összehangolása, java.doc megjegyzések készítése és a kapott osztályok implementálása függvények leírása commitolása.
2019.03.18 22:00	2.25 óra	Kecskés, Kiss, Kobál, Martonosi	<b>Értekezlet:</b> A feladatok leellenőrzése, egységesítés a .class-oknál. Felvetődött hitby(AThing: a) problémája. Értekezlet megbeszélése Mar 19. 16:00 kezdéssel.
2019.03.19 12:00	1.0 óra	Kobál	JavaDoc kommentek megírása a Wardrobe, Entry és az Exit osztályokban
2019.03.19 16:00	1.0 óra	Kobál, Kiss, Kis, Martonosi	<b>Értekezlet:</b> Printer használatának eldöntése, hitBy() probléma elemzése. Következő gyűlésre mindenkinek pótolnia kell a JavaDoc kommenteket.
2019.03.19 17:00	1.0 óra	Martonosi	Osztály dokumentálása és javítása a többi függvénynek megfelelően

2019.03.20 13:00	0.5 óra	Kobál	placeThing() metódus implementálása minden tárgyra, notifyNeighbors() felelőség átruházása, megoldás tesztelése.
2019.03.20 18:00	0.5 óra	Kobál, Kiss, Kis, Kecskés	<b>Értekezlet:</b> Tesztek megírásának felosztása. placeThing() probléma megoldásának egyeztetése. Mar 23. határidővel mindenkinek el kell készítenie a teszteseteket, és pótolnia a hiányosságokat az osztályokban.
2019.03.20 18:30	1.0 óra	Kecskés	A megbeszélésen elhangzott rám vonatkozó hibák javítása, plusz feladatok megoldása
2019.03.21 14:00	1.0 óra	Kiss	A kijelölt tesztek elkészítése. (10, 14, 15, 16)
2019.03.21 19:00	1.0 óra	Kobál	A kijelölt tesztek elkészítése
2019.03.23 14:00	1.0 óra	Kecskés	Megkapott tesztek megírása, commitolása (elfelejtettem pusholni is)
2019.03.23 18:00	1.5 óra	Kiss, Kobál, Kecskés	<b>Értekezlet:</b> placeThing() leaveTile() move() hatásköreinek megbeszélése. Kiss implementálja a mozgást végző függvényeket, Kobál pótolja a dokumentációt, a megfelelő függvényeket hozzáadja a master repository-ba, Kecskés elkészíti a futtatási útmutatót Mar 24. határidővel, amikor újabb értekezletre kerül sor.
2019.03.24 21:00	1.5 óra	Kiss, Kobál, Kecskés	<b>Értekezlet:</b> A kijelölt feladatok ellenőrzése történt meg, és a dokumentum megbeszélése.
2019.03.24 22:30	1.0 óra	Kecskés	Formázás, és a napló kiegészítése, a végleges dokumentum összeállítása.





## 7.1.3 Szekvencia-diagramok



## 7.2 Prototípus interface-definíciója

### 7.2.1 Az interfész általános leírása

A prototípus elsődlegesen parancssoron lesz vezérelhető, de mint kényelmi funkció, engedélyezzük előre megírt pályák illetve utasítássorozatok felhasználását.

A konzolos felület némileg behatárol minket, így az áttekinthetőség érdekében a pályák kizárólag valamilyen téglalap alakot vehetnek fel, benne minden csempe szabályos 4szög alakú.

### 7.2.2 Bemeneti nyelv

A determinisztikus működés érdekében a véletlenszerű elemek (tárgyak effektjei, szabad pandák mozgása, a játék elején a szekrény összeköttetések meghatározása) csak explicit hívhatóak, a rendszer nem maga intézi.

#### *cm (create map)*

**Leírás:** Létrehoz egy megfelelő nagyságú tömböt a paraméterlistában vagy a fájlban található sor(**r**) és oszlopszám(**c**) függvényében. Az ezt követő r db c bejegyzést tartalmazó sorból pedig létrehozza a pályát és beállítja a szomszédsági viszonyokat.

A pálya megalkotása után, ki is rajzoljuk azt.

**Opciók:** cm <number of rows> <number of cols>

cm <filename.extension>

Példa: cm map.txt

map.txt tartalma:

```
4 5
* r b r x
w r a r r
v r r w r
r r g X r
```

A pálya készítésekor, csak élettelen dolgok felvételét engedjük meg. Hiszen nem tudni pl., hogy egy állat alatt milyen típusú csempe van, így a pálya megadásához több információra lenne szükségünk, ami miatt kevésbé lehetne stuktúrált a bementünk. (élettelen tárgyak alatt biztosan törhetetlen csempe van)

Jelmagyarázat.

- r - regular tile, nem eltörhető csempe
- b - broken tile, eltörhető csempe
- x - entry, bejárat
- X - exit, kijárat
- a - armchair, fotel
- v - vending machine, csokiautomata
- g - game machine, játék automata
- w - wardrobe, szekrény
- \* - lyuk (széttört csempe pl)

#### *put*

**Leírás:** Letesz egy megadott fajta állatot, megadott névvel egy megadott mezőre amennyiben ez lehetséges. Ezután megjelenítjük a pályát.

**Opciók:**

put jpanda	<animal_name>	<row>	<col>
put spanda	<animal_name>	<row>	<col>
put lpanda	<animal_name>	<row>	<col>
put orangutan	<animal_name>	<row>	<col>

A pályában így megjelenő új jelek:

o - orangutan  
l - lazy panda  
j - jumping panda  
s - scared panda

### *tie*

**Leírás:** Összeköt két szekrényt. Amelyik állat belelép az első paraméteren megadott szekrénybe, az a második paraméteren megadott szekrény által meghatározott csempén találja magát.

**Opciók:** tie <wi> <wj>

wi, a pályát sorfolytonosan olvasva az i. szekrényt jelöli.

### *chain*

**Leírás:** Láncba fűz két állatot, amennyiben ez lehetséges. (=Beállítja a prev/nextAnimal kapcsolatokat és a prevTile mezőket.)

**Opciók:** chain <animal\_name1> <animal\_name2>  
chain <a1.row> <a1.col> <a2.row> <a2.col>

Az első állat mindig az orángutánhoz közelebbi állat (vagy maga az orángután), amely mögött a láncban már nincs semmi. Az állatokat megadhatjuk a nevükkel (kis pálya kreálása után a kezdeti láncok megalkotásánál hasznos) vagy koordinátájukkal (sok mozgás után már nem fogjuk tudni fejben tartani a nevüket az egyes állatoknak)

### *move*

**Leírás:** Lépteti a megadott állatot a megadott irányba, amennyiben ez lehetséges. Sikeres léptetés után kirajzolja a pályát. A megadott irány négyféle lehet, a valódi játéktól eltérően, kihasználva, hogy minden csempe szabályos négyszög alakú, így legfeljebb 4 szomszédal rendelkezik.

**Opciók:** move <animal\_name> <w/a/s/d>  
move <a.row> <a.col> <w/a/s/d>

Az állat megadható a nevével, illetve koordinátájával az előző parancsban ismertetett elvek szerint.

### *undo*

**Leírás:** Az utolsó lépésünket semmissé teszi. Újrarajzolja a pályát.

**Opciók:** undo

### *run*

**Leírás:** Futtat egy megadott fájlban lévő parancssorozatot.

**Opciók:** run <filename.extension>

### *save*

**Leírás:** Elmenti az adott pályát, a rajta lévő állatok nélkül a megadott fájlba, úgy hogy az utána a cm paranccsal visszaolvasható legyen.

**Opciók:** save <filename.extension>

### 7.2.3 Kimeneti nyelv

A parancsok egy része (cm, put, undo, run) a (cm + put) parancsok által meghatározott jelölésrendszer szerint kirajzolja az eddigi pályánkat a konzolra.

A save parancs az állatok nélküli pályánkat kimentti egy fájlba a cm parancsnál tárgyalt struktúra szerint. (törött csempék életteremtését, illetve szekrénykapcsolatokat elfelejti)

#### A többi kimenetet a hibaüzenetek adják:

##### általános hibaüzenetek:

“No such command”: a beírt sor első szava nem azonosítható parancsként

“No such parameterizing”: a beírt sor első szava azonosítható parancsként, viszont a többi szó nem ad helyes paraméterezést.

“(x,y) Overflow”: Koordinátás megadásoknál ha x vagy y koordináta túlmutat a pályán akkor írjuk ki.

##### cm hibaüzenetei:

“No entry, missing x symbol”: Nincs bejárat.

“No exit, missing X symbol”: Nincs kijárat.

“Single instance of wardrobe is forbidden”: Csak egy szekrény van, így nem kezelhető a működése.

##### put hibaüzenetei:

“Selected target is not empty”: Az állatot egy másik állatra vagy tárgyra akartuk tenni.

##### tie hibaüzenetei:

“<wi>: No wardrobe found”: valamelyik paraméterbeli szerkény nem létezik.

“<wi>: Already tied”: Az első paraméterbeli szekrényt már csatlakoztattuk. (Egy szekrényből csak egy szekrénybe mehetünk, viszont egy szekrénybe több szekrényből is eljuthatunk)

“Self-tie is forbidden”: A szekrény saját magához akartuk csatlakoztatni.

##### chain hibaüzenetei:

“Parameter: Wrong parameterizing”: Koordinátás megadásnál, akkor fordul elő, ha a koordináták nem állatra mutatnak, vagy ha a második kettő koordináta egy orángutánt jelöl.

Állatneves megadásnál pedig akkor ha a második paraméter egy orángután neve.

“Parameter: 2nd animal is already in a chain”: A parancsban meghatározott második állat már eleve láncban volt. Szeretnénk tiltani az ilyesfajta viselkedést, mert az adott állat új láncba tétele sok függvényhívást vonna maga után, továbbá könnyen nem várt eredményeket látnánk, ha nem tartjuk pontosan fejben a láncokat, ami még kis pályák esetében is nehézkes.

“Parameter: 1st animal is not at the end of the chain”: A 2. paraméterbeli állatot nem a lánc végéhez próbáltuk meg hozzáadni.

“Animals aren't located at neighbor tiles”: Két nem szomszédos csempén lévő állatot próbáltunk láncba rendezni.

##### move hibaüzenetei:

“Object cannot be moved”: A semmit, egy tárgyat vagy egy láncolt pandát próbáltunk mozgatni.

“Cannot step to target direction”: Lyukra próbáltunk lépni vagy esetleg a pályát akartuk elhagyni.

##### undo hibaüzenetei:

“No previous operation”: Még semmit nem csináltunk, és így akarunk visszaállítani.

##### run hibaüzenetei:

“Error in line: <n>: ...”: Ha a fájlban hibás utasítást találunk, akkor kiírjuk a hibás utasítás sorszámát(hanyadik sorban volt a fájlban), majd az utasítás hibaüzenetét, ellátva a hibát közvetlen megelőző pályaállapot ábrázolásával. A háttérben a pályát visszaállítjuk a run parancs előtti állapotra, hogy a hibát kijavítva újra tudjuk futtatni az utasítássorozatot.

### 7.3 Összes részletes use-case

<b>Use-case neve</b>	OrangutanTileStep
<b>Rövid leírás</b>	Az Orángután mozgását egyik mezőről a másikra tesztelő eset.
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. Az Orángután egy olyan pályán jelenik meg, ahol csak egy irányba indulhat el.</li> <li>2. Átlép az üres mezőre(Tile) sikeresen</li> <li>3. Átlépne a következő mezőre (Tile), de azon áll egy tárgy (VendingMachine), ezért a Tile visszautasítja és azon a pozíción marad, ahol eddig volt.</li> </ol>

<b>Use-case neve</b>	OrangutanIn-OutStep
<b>Rövid leírás</b>	Az Orángután Entry és Exit interakcióját tesztelő függvény
<b>Aktorok</b>	Controller, Player
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A pályán található egy Entry, ahol egy Orángután belép.</li> <li>2. Az Orángután pedig csak egy irányba, a tőle szomszédos mezőn található Exit irányába tud menni.</li> <li>3. Az Orángután kilép a kijáraton.</li> <li>4. Az Entry újra kilép az Orángután.</li> </ol>

<b>Use-case neve</b>	OrángutánWardrobeInteraction
<b>Rövid leírás</b>	Orángután egy Wardrobe-bbal lévő interakcióját tesztelő függvény
<b>Aktorok</b>	Player, Controller
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A pályán található Orángután az előtte lévő Wardrobe-ba lép.</li> <li>2.</li> </ol>

<b>Use-case neve</b>	AnimalsDie
<b>Rövid leírás</b>	Egy Orángután és egy Panda is olyan BrokenTile-ra lép, ami pont alatta törik el, és leesnek.
<b>Aktorok</b>	Player, Controller
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. Az pályán található egy Orángután, egy Panda és egy-egy BrokenTile annyi étellel, hogy pont rálépve zuhanjon le vele az állat.</li> <li>2. Az Orángután rálép az egyik BrokenTile-ra és lezuhan.</li> <li>3. A Panda rálép a másik BrokenTilera, és lezuhan.</li> </ol>

<b>Use-case neve</b>	Orangutan2PandaInteraction
<b>Rövid leírás</b>	Az Orángután Pandákat állít maga mögé
<b>Aktorok</b>	Player, Controller
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A pályán található egy Orángután és két Panda.</li> <li>2. Az Orángután hozzáér az első Pandához, ami nincs láncban és az Orángután beállítja maga mögé.</li> <li>3. Az Orángután felvesz egy másik Pandát a sorába és közvetlen maga mögé rakja, a Panda kezébe az első Panda lesz.</li> <li>4. Az Orángután kilép egy olyan Wardrobe-on, ami össze van kötve azzal a Wardrobe-al, amin belépett.</li> </ol>

<b>Use-case neve</b>	JumpingPandaInteraction
<b>Rövid leírás</b>	A speciális JumpingPanda ugrós tulajdonságát tesztelő Use-case
<b>Aktorok</b>	Player, Controller
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A pályán található egy JumpingPanda egy BrokenTile-on, ami közvetlen egy VendingMachine közelében van.</li> <li>2. A VendingMachine sípol, amire a Panda ugrik egyet.</li> </ol>

<b>Use-case neve</b>	ScaredPandaInteraction
<b>Rövid leírás</b>	A speciális ScaredPanda tulajdonságát tesztelő Use-case
<b>Aktorok</b>	Player, Controller
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A pályán található egy láncban lévő ScaredPanda (Orángután előtte és egy sima Panda mögötte), ami közvetlen egy GameMachine közelében van.</li> <li>2. A ScaredPanda csilingel, amire a Panda megijed és elengedi a kezét.</li> <li>3. A lánc szétesik</li> </ol>

<b>Use-case neve</b>	LazyPandaInteraction
<b>Rövid leírás</b>	A speciális LazyPanda tulajdonságát tesztelő Use-case
<b>Aktorok</b>	Player, Controller
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A pályán található egy láncban lévő LazyPanda (Orángután előtte és egy másik LazyPanda mögötte), ami egy Armchair közelében van.</li> <li>2. Az első LazyPanda amint közvetlen szomszédja lesz az Armchairnak, leül és szétszedi a láncát.</li> <li>3. A második LazyPanda is a közelébe ér, de nem tud leülni, mert már foglalt.</li> <li>4. Az első LazyPanda feláll az Armchairból.</li> </ol>

## 7.4 Tesztelési terv

<b>Teszt-eset neve</b>	Program starts and makes a map
<b>Rövid leírás</b>	A program elindul és betölt egy pályát
<b>Teszt célja</b>	A programba indulás után sikeresen betöltődik a pálya

<b>Teszt-eset neve</b>	Entry generates 2 orangutans, and player steps with them
<b>Rövid leírás</b>	A bejáraton belép 2 orángután, majd a játékos lép velük
<b>Teszt célja</b>	Sikeresen létrejönnek az orángutánok, majd a játékos lép velük.

<b>Teszt-eset neve</b>	Two orangutans wandering on the floor and suddenly one orangutan sees a panda and hits it
<b>Rövid leírás</b>	A pályán lévő két orángután közül az egyik nekimegy egy pandának, ezáltal befűzi a láncába
<b>Teszt célja</b>	Az elkapott panda sikeresen bekerül az orángután láncába

<b>Teszt-eset neve</b>	Orangutan with Panda-chain go through the Wardrobe
<b>Rövid leírás</b>	Egy orángután a láncával átkel egy szekrényen.
<b>Teszt célja</b>	A panda és lánc eltűnik az egyik szekrényben, majd megjelenik a másikban

<b>Teszt-eset neve</b>	Armchair notifies some randomly stepping pandas
<b>Rövid leírás</b>	A pályán szabadon mozgó pandák közül egy fotel értesít többet is
<b>Teszt célja</b>	Az első értesített panda beleül a fotelbe, majd a többi figyelmen kívül hagyja az értesítést (mivel a fotelben már ülnek)

<b>Teszt-eset neve</b>	Randomly stepping pandas walking on a broken tile, and suddenly one of them dies
<b>Rövid leírás</b>	Szabadon mozgó pandák törött csempén sétálnak amíg az le nem szakad, és a rajta lévő panda meg nem hal
<b>Teszt célja</b>	Amikor a csempe életeréje 0-ra csökken az beszakad, az éppen rajta lévő panda meghal

<b>Teszt-eset neve</b>	Lonely Orangutan walking around a broken tile and breaks it, then the game ends
<b>Rövid leírás</b>	A pályán lévő utolsó orángután addig sétál egy törött csempén amíg az le nem szakad, az orángután meghal, majd mivel ő volt az utolsó, a játék véget ér
<b>Teszt célja</b>	Amikor a csempe életeréje 0-ra csökken az beszakad, az orángután meghal, a játék véget ér

<b>Teszt-eset neve</b>	GameMachine scares a panda
<b>Rövid leírás</b>	Egy játékgép megijeszti egy láncba fogott pandát ami emiatt elengedi a mögötte álló panda kezét.
<b>Teszt célja</b>	A játékgép jelzése után a panda elengedi a mögötte álló kezét.

<b>Teszt-eset neve</b>	Orangutan steals Pandas from an Orangutan without Panda-chain
<b>Rövid leírás</b>	Egy lánc nélküli orángután nekimegy egy másik láncal rendelkezőnek, és így ellopja a láncát
<b>Teszt célja</b>	Az ütközés után a lánc nélküli orángután megkapja a másik láncát, az eredeti orángután pedig elveszíti azt



<b>Teszt-eset neve</b>	Orangutan hits a chained Panda from another chain
<b>Rövid leírás</b>	Egy orángután nekimegy egy láncba fűzött pandának, aki ezáltal elengedi az előtte lévő kezét és a mögötte lévőkkal beáll a nekik menő orángután láncába
<b>Teszt célja</b>	A megütött panda elengedi az előtte lévő kezét, majd sikeresen csatlakozik a mögötte lévőkkal a másik orángutánhoz

<b>Teszt-eset neve</b>	WendingMachine makes panda jump
<b>Rövid leírás</b>	Egy csokiautomata megijeszt egy Pandát a szomszédos mezőn.
<b>Teszt célja</b>	A Panda ugrik egyet a csempén egyet levonva a csempe életéből.

<b>Teszt-eset neve</b>	Orangutan hits it's own Panda chain
<b>Rövid leírás</b>	Egy orángután belemegy a saját pandasorának közepébe.
<b>Teszt célja</b>	Az orángután befűzi a pandát maga mögé aminek nekiment, és az újonnan befűzött panda elengedi a mögötte álló láncot.

<b>Teszt-eset neve</b>	Orangutan exits with 2 Pandas
<b>Rövid leírás</b>	Egy orángután kimegy a kijáraton miközben 2 pandát húz maga után.
<b>Teszt célja</b>	Az orángután eltűnik a játéktérről, de a pandák láncban maradnak, a kijárat pedig megjegyzi, hogy melyik pandát kell behívja a következő körben.

<b>Teszt-eset neve</b>	Exit invites panda to come in
<b>Rövid leírás</b>	A kijárat behív egy pandát magába.
<b>Teszt célja</b>	A behívott panda eltűnik a játéktérről, de ha van még utána panda a láncban, akkor a kijárat megjegyzi, hogy kit kell következő körben behívnia.

<b>Teszt-eset neve</b>	A GameMachine notifies one Panda, and it leaves the chain with the pandas behind.
<b>Rövid leírás</b>	A Játékautomota megijeszt egy pandát egy szomszédos mezőn.
<b>Teszt célja</b>	A megijesztett panda elengedi a mögötte álló pandaláncot.

<b>Teszt-eset neve</b>	Panda breaks a tile
<b>Rövid leírás</b>	A Panda mozgásával eltör egy csempét.
<b>Teszt célja</b>	A Panda alatt eltörik a csempe, amitől a Panda lezuhan, és meghal.

<b>Teszt-eset neve</b>	Last Orangutan dies
<b>Rövid leírás</b>	Az utolsó orángután is kikerül a játékból.
<b>Teszt célja</b>	A játék véget ér, hiszen az egyik állatfajta meghalt.

<b>Teszt-eset neve</b>	Last Panda dies
<b>Rövid leírás</b>	Az utolsó Panda is kikerül a játékból.
<b>Teszt célja</b>	A játék véget ér, hiszen az egyik állatfajta meghalt.

<b>Teszt-eset neve</b>	Orangutan releases its PandaChain
<b>Rövid leírás</b>	Az orángután valamilyen okból kifolyólag elengedi a mögötte lévő pandaláncot.
<b>Teszt célja</b>	A Pandák szétszélednek és folytatják véletlenszerű mozgásukat a pályán.

<b>Teszt-eset neve</b>	Orangutan brings PandaChain over broken tile
<b>Rövid leírás</b>	Egy Orángután egy pandaláncot húz át egy törött csemple fölött.
<b>Teszt célja</b>	A mozgás során minden állat ami rálép a csempére, levon egyet annak életéből.

### 7.5 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A program batch módon (pl. PowerShell segítségével) programozható. A tesztek úgy megvalósítottak, hogy bemenetként van szöveges file, és a program a console-ra dobja ki a megoldást. Ezt össze tudjuk hasonlítani egy egyszerű összehasonlító paranccsal (mint UNIX rendszerek alatt található diff parancs, windows-on FC vagy „findstr /V /G:input.txt output.txt >DiffResult.txt”) és amennyiben egyezik a várt kimenettel, akkor sikeres a teszt (OK). Ha a teszt nem sikertelen (FAIL), akkor kijelzi a program, hogy eltérést tapasztalt a sorban.

### 7.6

**Napló**

Kezdet	Időtartam	Résztevők	Leírás
2019.03.29 19:30	1.5 óra	Kecskés, Kobál, Martonosi	<b>Értekezlet:</b> A kapott feladatok megbeszélése és összegyűjtése. Változtatások megbeszélése és javítása az osztálydiagramban.
2019.03.30 14:00	0.5 óra	Kiss, Kobál, Martonosi	<b>Értekezlet:</b> Feladatok gyors (megvalósítandó függvények/osztályok) kiosztása hétvégére.
2019.03.30 21:45	1.0 óra	Martonosi	Kiosztott tesztesetek megírása
2019.03.31 12:00	1.0 óra	Kobál	Megírásra került a "Változás hatása a modellre" dokumentum pont.
2019.03.31 13:00	1.0 óra	Kiss	A megbeszélésnek megfelelően a 12-21- es tesztesetek elkészítése
2019.03.31 18:00	0.2 óra	Kecskés, Kiss, Martonosi	<b>Értekezlet:</b> Kész feladatok ellenőrzése/véglegesítése
2019.03.31 20:00	3.5 óra	Kis	Prototípus interfészének bemeneti és kimeneti nyelvének megalkotása
2019.04.01 00:00	1.0 óra	Kecskés	A felosztott feladatok egybeolvasztása, ellenőrzése és véglegesítése naplóval a végén egy dokumentumba

## 8. Részletes tervek

### 8.1 Osztályok és metódusok tervei.

#### 8.1.1 Animal

- **Felelősség**

A játékban lévő állatokat általánosan reprezentáló abstract osztály. Rendeződhetnek láncokba, lezuhanhatnak, illetve távozhatnak az emeletről a kijáraton.

- **Ősosztályok**

- **AThing**

- **Interfészek**

-

- **Attribútumok**

- **#previousAnimal: Animal:** Az állat előtt álló állat. Ha nincs akkor null.
- **#nextAnimal: Animal:** Az állat után álló állat (csak Panda lehet egyébként). Ha nincs akkor null.
- **#prevTile: Tile:** Az állat előző lépésben elfoglalt pozíciója.

- **Metódusok**

- **+setPrevAnimal(prevAnimal: Animal) :void:** prevAnimal setter
- **+setNextAnimal(nextAnimal: Animal) :void:** nextAnimal setter
- **+setPrevTile(prevTile: Tile) :void:** PrevTile setter
- **+getPrevAnimal(): Animal:** prevAnimal getter
- **+getNextAnimal(): Animal:** NextAnimal getter
- **+move(tile: Tile): void:** Az argumentumként kapott t csempére lépteti az állatot.
- **+kill(): void:** Megöli az adott állatot, azaz végérvényesen eltávolítja a játékból. Ha a lezuhanó állat láncban volt, akkor megtöri a láncot.
- **+connectChain(): void:** A Pandalánc összeszervezése a feladata a függvénynek.
- **+leaveTile(t: Tile): void:** Amikor az állat másik csempére lép, akkor meghívott függvény
- **+unchain(): void:** Kiszedi a pandát a láncból

#### 8.1.2 Armchair

- **Felelősség**

Foteleket kezelő osztály, ha egy panda a szomszédságába kerül, és hajlamos az elalvásra, akkor jelzi számára, hogy aludjon el.

- **Ősosztályok**

- **AThing**

- **Interfészek**

- **IPandaEffective**

- **Attribútumok**
  - **-panda: LazyPanda:** A fotelben ülő panda.
- **Metódusok**
  - **+effect(): void:** Minden körben jelzi a szomszédos csempéknek, hogy ha van rajtuk panda, azt küldjék bele aludni.
  - **+isOccupied(): bool:** Visszadja, hogy az adott fotelben már ülnek-e, ha igen, az értéke true.
  - **+notifyNeighbors(): void:** Amikor aktív a fotel jelez, akkor a közeli szomszédoknak jelzi, hogy menjenek a pandák hozzá.
  - **+setPanda(p: LazyPanda): void:** berakja a pandát a fotelbe

### 8.1.3 AThing

- **Felelősség**

Absztrakt ősosztály, ami a tárgyakat és az állatokat reprezentálja, biztosítja a rájuk jellemző tulajdonságokat. Tudja, hogy melyik csempén helyezkedik el. El lehet helyezni csempére.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **#isOn: Tile:** Az a csempe, amin elhelyezkedik a tárgy, vagy állat.

- **Metódusok**

- **+getIsOn(): Tile:** megadja a csemét, ami áll az elem.
- **+setIsOn(isOn: Tile): void:** beállítja a csempét, amin áll az elem
- **+hitBy(panda: Panda): bool:** Panda ütközik ezzel az objektummal
- **+hitBy(orangutan: Orangutan): bool:** Orángután ütközik ezzel az objektummal
- **+hitBy(vm: VendingMachine): bool:** VendingMachine ütközik ezzel az objektummal
- **+hitBy(gm: GameMachine): bool:** GameMachine ütközik ezzel az objektummal
- **+hitBy(armchair: Armchair): bool:** Armchair ütközik ezzel az objektummal
- **+notifyNeighbors(): void:** Értesíti a szomszédokat

### 8.1.4 BrokenTile

- **Felelősség**

A törött csempe osztálya ez. A törött csempe képes eltörni és megölni az állatokat rajta, ha az élete nullára csökken.

- **Ősosztályok**

- **Tile**

- **Interfészek**

-

- **Attribútumok**
  - **-lifeTime: int:** A hátralévő ugrások számát tárolja. Ha nullára fogy, akkor törlik el a csempe.
- **Metódusok**
  - **+loseLife(): void:** A meghívás hatására csökkenti az életét a csempének.
  - **+BrokenTile(n: int):** konstruktor, az N a csempe élete, ahány lépést bír az egész.

### 8.1.5 Controller

- **Felelősség**  
A pályát generálja és mozgatja a pandákat.
- **Ősosztályok**  
-
- **Interfészek**  
-
- **Attribútumok**
  - **-orangutansToPush: int:** Azon orángutánok száma, akiket még ki kell küldeni az emeletre.
  - **-orangutansPoint[]: int:** Orángutánok pontja számon tartva, létrehozásuk sorrendjében
  - **-orangutans[]: Orantugan:** Az a tömb, amiben tárolva van az összes Orángután.
  - **-map: Floor:** A tényleges pálya, amin végig fut a Step(). Ez egy fileba írható osztály, amivel a teszteseteket lehet összehasonlítani egyszerűen.
- **Metódusok**
  - **+Initialization(): void:** létrehozza a (még üres) tömböket/változókat.
  - **+pushOrangutan(): void:** Létrehoz egy orángutánt, és kihelyezi az első elérhető szomszédos csempére, majd csökkenti a kiküldendő orángutánok számát eggyel.
  - **+addOrangutan(n: int): void:** Hozzáad a kiküldendő orángutánok számához egy paraméterként átadott mennyiséget.
  - **+movePandaRandomly(panda: Panda): void:** A pandákat mozgató függvény.
  - **+moveOrangutan(orangutan: Orangutan): void:** Az Orángutánt mozgató függvény.
  - **+generateMap(): void:** A pályát generáló függvény, fileból olvas be egy pályát és állítja a tárgyainak értékét
  - **+OrangutanExiting(): void:** A függvény, ami megoldja, hogy egy kijáratba lépett panda sora kijöhessen, megkapja a pandák utáni pontját, majd újra beküldje a bejáraton
  - **+Step(): void:** Az idő múlásával meghívott függvény, minden periodikusan történő eseményt ez a függvény hív meg úgy, hogy végig megy a pályán és sorrendben először az Orángutánokat engedi léptetni, majd a Pandákat. Végül pedig a pályán történő egyéb események, mint csörgés és sípolás kezdődik, ha épp eljött az ideje.

### 8.1.6 Entry

- **Felelősség**  
Bejáratot reprezentáló osztály. Ha van még kiküldendő orángután, akkor a kör elején kiküld egyet, majd csökkenti eggyel a kiküldendő orángutánok számát.
- **Ősosztályok**

- **AThing**
- **Interfészek**
  -
- **Attrubútomok**
  - **-entryTile: Tile**
  - **-orangutansToPush: int**
- **Metódusok**
  - **+Entry(entryTile: Tile):** Konstruktor, paraméterében megadtott a csempe, amire lépnek az ajtón kilépő Orángutának
  - **-pushOrangutan(): void:** Létrehoz egy orángutánt, és kihelyezi az első elérhető szomszédos csempére, majd csökkenti a kiküldendő orángutának számát eggyel.
  - **+addOrangutan(n: int): void:** Hozzáad a kiküldendő orángutának számához egy paraméterként átadott mennyiséget.
  - **+nextTurn(): void:** Lerak egy Orángutánt az ajtó mellett lévő csempére.

### 8.1.7 Exit

- **Felelősség**

Kijáratot reprezentáló osztály, ha interakcióba lép egy orángutánnal, akkor eltünteti azt az emeletről. Amikor az orángután a kijáratához vezette a pandákat, átadja a mögötte lévő pandát a kijáratnak, ezek után a kijárat lépésről lépésre befogadja a pandákat. Ha egy pandákból álló lánc végig ment a kijáraton, hozzáad egy kiküldendő orángutánt a bejáratához.
- **Ősosztályok**
  - **AThing**
- **Interfészek**
  -
- **Attribútumok**
  - **-nextPanda: Panda:** A következő befogadható panda.
- **Metódusok**
  - **+hitBy(panda: Panda): boolean:** A láncban lévő Pandát juttatja ki a kijáraton és adja hozzá pontként az Orángutánhoz.
  - **+hitBy(orangutan: Orangutan): void:** Felülírja az őszosztály metódusát, ha egy orángután interakcióba lép a kijáratral, akkor eltünteti a pályáról, beállítja a nextPanda attribútumát, és sorba befogadja a pandákat. True-val tér vissza, ha az orángutánnak sikerült átlépni a kijáraton, ellenkező esetben false.

### 8.1.8 Floor

- **Felelősség**

A pályát reprezentáló osztály. „Szerializálható” (toString-gel) és tartalmaz szomszédsági mátrixot és a tárgyak tömbjét.

- **Ősosztályok**
- 
- **Interfészek**
- 
- **Attribútumok**
  - +tileset: Tile[1..\*]: Szomszédossági mátrix
  - +things: AThing[0..\*]: A tárgyakat tároló tömb
- **Metódusok**
  - +ToString(): String: A pálya tartalmát szövegbe író függvény

### 8.1.9 GameMachine

- **Felelősség**  
Játékgépeket kezelő osztály, csilingeléssel megijeszti a körülötte lévő, arra hajlamos pandákat.
- **Ősosztályok**
  - **AThing**
- **Interfészek**
  - **IPandaEffective**
- **Attribútumok**
  - **-tinkleTime: int:** A játék során megkapott random szám, ami alapján adott játékgép tudja, hogy hány körönte kell csilingelnie. (a baseTinkleTime alapján számolt)
  - **-baseTinkleTime: int:** A játék inicializálása során megkapott random szám, ami alapján adott játékgép tudja, hogy hány körönte kell csilingelnie.
- **Metódusok**
  - **+GameMachine(i: int):** beállítja a tinkleTime-ot i értékére (az időtartam amikor a játékgép hangot ad)
  - **+effect(): void:** Ha eltelt a TinkleTime attribútumban tárolt számú kör, akkor jelzi a körülötte lévő csempéknek, hogy ijesszék meg a megfelelő pandákat.
  - **+notifyNeighbors(): void:** Értesíti a szomszédokat az effectről, hogy a játékgép hangot ad.

### 8.1.10 IPandaEffective

- **Felelősség**  
Egy olyan interface, ami egységes hatással rendelkező tárgyakat fog össze. Ilyen tárgy az, ami tud csengeni, sípolni vagy elfárasztani közelben lévő Pandát. Minden ilyen környező területet értesít, hogy milyen típusú jelzés érkezik.
- **Ősosztályok**
- 
- **Interfészek**



-

- **Attribútumok**

-

- **Metódusok**

- **+effect(): void:** Egy olyan függvény, amit meghívva az implementált osztályban megszólítja a szomszéd csempéket, hogy éppen valamilyen hatás zajlik.

### 8.1.11 JumpingPanda

- **Felelősség**

A JumpingPanda példányok sípszó hallatán ugranak egyet.

- **Ősosztályok**

Panda

- **Interfészek**

- -

- **Attribútumok**

- -

- **Metódusok**

- **+hitBy(vm: VendingMachine): bool:** A panda meghallja a csokiautomata(vm) sípolását és ugrik egyet. True-t adunk vissza, mert JumpingPanda esetében ez mindig bekövetkezik. Ilyenkor a panda egyet csökkenti annak a csempének az életét amin éppen áll. (Meghívja a loseLife() függvényét.)

### 8.1.12 LazyPanda

- **Felelősség**

A játékban szereplő lusta pandák. Ezeknek tulajdonsága, hogyha elhaladnak egy fotel mellett, akkor elfáradnak és beleülnek a fotelre jellemző ideig (kőrig). Ez alatt az idő alatt a panda nem mozog, még egy órán át sem tudja felkelteni. Az idő letelte után a panda folytatja a rá jellemző véletlenszerű mozgását.

- **Ősosztályok**

- **Panda**

- **Interfészek**

- -

- **Attribútumok**

- **-sleepTime: int:** A pandák akkor mozoghatnak ha a sleep time értékük nulla. Amennyiben ez nem teljesül, akkor az azt jelenti, hogy a panda még annyi körig alszik, amennyi az attribútum értéke.
- **-pandasRegularSleepTime: int:** A Pandára jellemző alvásidő.
- **Metódusok**
  - **+LazyPanda(n: int):** Kontruktor, amiben az n adja meg, hogy mennyi ideig alszik a Panda.
  - **+hitBy(ac: Armchair): bool:** Ez a függvény akkor hívódik, ha egy lusta panda elhalad egy fotel mellett. Ekkor beleül a fotelbe, sleepTime attribútumát beállítja a pandára jellemző értékre, és elalszik.
  - **+move(tile: Tile): void:** Amikor a panda mozog a tile paraméterként kapott mezőre. Ekkor ellenőrzi, hogy tud e mozogni (nem alszik) és egy másik csempére lépteti.

### 8.1.13 Orangutan

- **Felelősség**

A játékosok által irányított állatok, akik a pandákat fogják meg.

- **Ősosztályok**

- **Animal**

- **Interfészek**

-

- **Attribútumok**

- **+count: static int:** A játékban még meglevő orángutánok száma.

- **Metódusok**

- **+move(tile: Tile): void:** Mozgatja a pandát tile paraméterként kaptott mezőre. Ha nem tud semerre mozogni, akkor elengedi a láncát (hátha így tud magának vágni új utat)
- **+kill(): void:** Megöli a pandát és bontja a Pandaláncot.
- **+connectChain(animal: Animal): void:** Üres függvény az Animaltól örököelve. (Orángután nem tud csatlakozni lánchoz ugye, ezért üres)
- **Orangutan.hitBy(orangutan: Orangutan): bool:** A függvény felülírja az őssztály metódusát. Ha egy Orangutan nekimegy egy másik Orángutannak, akkor az előbbi elveszi az utóbbitól a Panda láncát. Az Orángutánok helyet cserélnek, és a panda lánc átlinkelődik az egyiktől a másikhoz.
- **Orangutan.releasePandas(): void:** A függvény lehetőséget ad arra, hogy az Orángután elengedje a saját pandaSorát.

### 8.1.14 Panda

- **Felelősség**

Pandákat reprezentáló abstract osztály. Az egyes pandák fáradékonyak, ijedősek vagy sípszóra ugranak.

- **Ősosztályok**

- **Animal**
- **Interfészek**
  -
- **Attribútumok**
  -
- **Metódusok**
  - **+hitBy(or: Orangutan): bool:** Kezeli az esetet, amikor a panda csempéjére egy orángután akar lépni. Az orángután a pandára lépve befűzi a pandát a láncába.
  - **+unchain(): void:** Az állat, amin meghívjuk a függvényt elengedi az előtte álló állat mancsát, majd az megint az előtte állóét és így tovább. A függvény rekurzív működésű. A futás végeztével a leszakadt állatok már nem tudnak egymásról (previousAnimal és nextAnimal is null).
  - **+isInChain(): bool:** Megadja, hogy az adott állat láncban van-e. Ezt a next/previous animal értéke pontosan meghatározza: Ha mindkettő null, akkor false.
  - **+move(tile: Tile): void:** A Panda tile paraméterű csempére mozgó függvénye.
  - **+kill(): void:** A Panda halálakor meghívott függvény. Elereszt a láncot maga mögött, ha meghal.

### 8.1.15 RegularTile

- **Felelősség**

Tárolja a tárgyakat és lekezeli az lépéseket, zajokat. És mindent tud, amit az őosztálya természetesen.

#### Őosztályok

- **Tile**
- **Interfészek**
  -
- **Attribútumok**
  -
- **Metódusok**
  -

### 8.1.16 ScaredPanda

- **Felelősség**

A ScaredPanda példányai a játékgép csilingelésétől megijednek és elengedik az előttük álló állatok mancsát.

- **Őosztályok**
  - **Panda**

- **Interfészek**

- 

- **Attribútumok**

- 

- **Metódusok**

- **+hitBy(gm: GameMachine): bool:** A ScaredPanda a csilingelésre elengedi az előtte álló állat mancsát. Mivel ez mindig megtörténik a függvény igaz értékkel tér vissza.

### 8.1.17 Tile

- **Felelősség**

Absztrakt ősosztály, ami a csempéket reprezentálja és meghatározza azokat a metódusokat, amik minden csempe esetében elengedhetetlennek számítanak.

A tile abstract osztályba sorolható minden pályán található csempe, ami a mezőket alkotja. Csempéből van több változat, mind különböző változattal rendelkezik.

Minden csempének van szomszédja és be tud fogadni maximum egy dolgot. Minden csempe értesíti a szomszédos csempéket, ha egy tárgy rajta zajt ad.

- **Ősosztályok**

- 

- **Interfészek**

- 

- **Attribútumok**

- **#contains: AThing:** Tárolja azt a tárgyat, vagy állatot ami a csempén található. Tárgyat vagy állatot képes így, és maximum egyet tud.
  - **#neighbors: Tile:** A szomszédos csempéket nyilvántartó kollekció. Ide léphetnek az állatok, és ide szól a zaj át.

- **Metódusok**

- **+setContains(thing: AThing): void:** contains setter
  - **+addNeighbor(tile: Tile): void:** Hozzáadja szomszédnak a kapott mezőt a szomszédot tároló tömbhöz a függvény.
  - **+getContains(): AThing:** contains getter függvény.
  - **+getNeighbors(): Tile[]:** Visszadja a szomszédokat tároló tömböt.
  - **+Tile():** Konstruktor, ami inicializálja a tömböt, amit tartalmaz(szomszédokat tároló tömb).
  - **+placeThing(panda: Panda): bool:** A csempére mozgó Pandát fogadja be ha tudja. Ütközteti először a mezőn álló tárggyal, ha van rajta és csak akkor fogadja be, ha az ütközés engedi ezt.
  - **+placeThing(orangutan: Orangutan): bool:** A csempére mozgó Orángutánt fogadja be ha tudja. Ütközteti először a mezőn álló tárggyal, ha van rajta és csak akkor fogadja be, ha az ütközés engedi ezt.
  - **+placeThing(vm: VendingMachine): bool:** A csempére rakott VendingMachine-t fogadja be. Ütközteti először a mezőn álló tárggyal, ha van rajta és csak akkor fogadja be, ha az ütközés engedi ezt. Különbön reagál pl. átadja az effectet.
  - **+placeThing(gm: GameMachine): bool:** A csempére rakott GameMachine-t fogadja be. Ütközteti először a mezőn álló tárggyal, ha van rajta és csak akkor fogadja be, ha az ütközés engedi ezt. Különbön reagál pl. átadja az effectet.

- **+placeThing(*armchair: Armchair*): bool**: A csempére rakott Armchair-t fogadja be. Ütközteti először a mezőn álló tárggyal, ha van rajta és csak akkor fogadja be, ha az ütközés engedi ezt. Különben reagál pl. átadja az effectet.
- **+loseLife(): void**: Ez a virtuális függvény azért van, hogy kompatibilis legyen a BrokenTile a pályával.
- **+unlink(*tile: Tile*): void**: Törli a szomszédjai közül a kapott csempét. Törött csempe esetén hívjuk ezt a függvényt.

### 8.1.18 VendingMachine

- **Felelősség**

Csokiautomatákat kezelő osztály, sípolással megijeszti a körülötte lévő, arra hajlamos pandákat.

- **Ősosztályok**

- **AThing**

- **Interfészek**

- **IPandaEffective**

- **Attribútumok**

- **-beepTime: int**: A játék során kapott szám, ami alapján adott csokiautomata tudja, hogy hány körönte kell sípolnia.
- **-baseBeepTime: int**: A játék inicializálása során megkapott random szám, csokiautomata tudja, hogy hány körönte kell sípolnia.

- **Metódusok**

- **+VendingMachine(*t: int*)**: Konstruktor, megadja *t* paraméterében, hogy milyen időközönként ad ki hangot.
- **+effect():void**: Ha eltelt a BeepTime attribútumban tárolt számú kör, akkor jelzi a körülötte lévő csempéknek, hogy ijesszék meg a megfelelő pandákat.
- **+noitfyNeighbors(): void**: Értesíti a szomszédokat az effectről, hogy a csokiautomata hangot ad.

### 8.1.19 Wardrobe

- **Felelősség**

Szekrényt reprezentáló osztály, ha egy állat egy szekrénnel lép interakcióba, akkor egy, ezzel a szekrénnel összekötött, másik szekrényből kerül ki. A második szekrény egy szomszédos mezőre dobja ki az első szekrénybe bekerült állatot.

- **Ősosztályok**

- **AThing**

- **Interfészek**

-

- **Attribútumok**

- **-outPoint: Wardrobe**: Egy másik szekrény, ahonnan az állat majd kikerül.

- **Metódusok**
  - **+setOutPoint(wd: Wardrobe): void:** outPoint setter
  - **+pushOut(animal: Animal): void:** Elhelyezi az állatot az első elérhető szomszédos mezőre.
  - **+hitBy(panda: Panda): bool:** Felülírja az ősosztály metódusát, átadja az pandát a pár szekrénynek, hogy az majd kidobja. True-val tér vissza, ha sikerült az pandát átvinni a szekrények között, és elhelyezni egy csempén, ellenkező esetben false.
  - **+hitBy(orangutan: Orangutan): bool:** Felülírja az ősosztály metódusát, átadja az orángutánt egy megfelelő másik szekrénynek, hogy az majd kidobja. True-val tér vissza, ha sikerült az orángutánt átvinni a szekrények között, és elhelyezni egy csempén, ellenkező esetben false.

## 8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

### 8.2.1 Program starts and makes a map

- **Leírás**

Betölti a pályát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Itt ellenőrizzük a program működését az emelet betöltésekor.

- **Bemenet**

```
cm test1.flr
```

- **Elvárt kimenet**

```
cm test1.flr
0  1  2  3  4  5  6  7  8  9
1  _  X  _  _  _  _  _  _  _
2  X  _  X  _  _  _  _  X  _
3  _  X  _  X  _  _  _  X  _
4  _  _  X  _  X  _  _  X  _
5  _  _  _  X  X  _  X  X  _
6  _  _  _  _  X  X  X  _  X
7  _  _  _  _  _  X  _  X  _
8  _  X  X  X  X  X  X  _  _
9  _  _  _  _  _  X  _  _  _
1:X
2:r
3:g
4:w
5:a
6:r
7:w
8:b20
9:x2
```

### 8.2.2 Entry generates two orangutans and player steps with them

- **Leírás**

Az emeleten elhelyezkedő Entry tárgy 2 Orangutant kell, hogy kitegyen. Miután kiteszi az elsőt, abban a körben a játékosnak lépnie kell az Orangutannal. A következő körben az Entry kitesz még egy Orangutant és ezzel is lépnie kell a játékosnak.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt ellenőrzi a játék azon funkcióját, hogy a játék kezdetekor az Entry kihelyezi a szükséges mennyiségű Orangutant az emeletre.

- **Bemenet**

```
cm test2.flr
start
move o1 3
move o1 4
move o2 3
```

- **Elvárt kimenet**

```

cm test2.flr
  0  1  2  3  4
  1      X      -
  2  X      X      -
  3      X      X
  4  -      X      -
1:x2
2:r
3:r
4:r
start
1:x1
2:o
3:r
4:r
make a move o1
move o1 3
1:x0
2:o
3:o
4:r
make a move o1
move o1 4
1:x0
2:o
3:r
4:o
make a move o2
move o2 3
1:x0
2:r
3:o
4:o
make a move o1

```

## 8.2.3

## 8.2.4 Two orangutans wandering on the floor and suddenly one orangutan sees a panda and hits it.

- **Leírás**

Az emeleten 2 db Orangutan és 1 db Panda helyezkedik el. Az Orangutanokat úgy léptetjük, hogy az egyik elkapja a pandát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

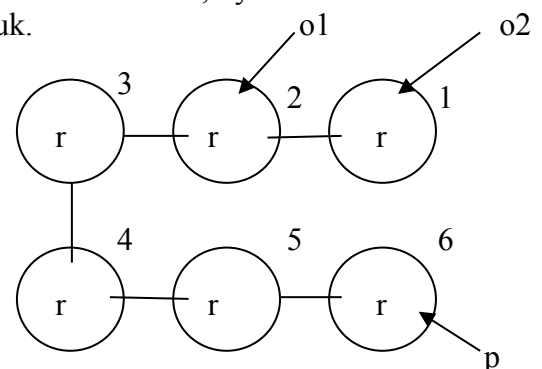
A teszt ellenőrzi a játék azon funkcióját, hogy egy Orangutan befűzhet egy pandát. Ha a panda a 2. körben az ellenkező irányba lép, akkor a kimenet különbözhet, ilyenkor a tesztet addig kell futtatni, amíg a megfelelő kimenetet nem kapjuk.

- **Bemenet**

```
cm test3.flr
put orangutan o1 1
put orangutan o2 2
put lpanda p1 6
start
move o1 1
move o2 2
move o1 4
move o2 3
move o1 5
```

- **Elvárt kimenet**

```
cm test3.flr
0 1 2 3 4 5 6
1 X - - - -
2 X - X - - -
3 - X - X - -
4 - - X - X -
5 - - - X - X
6 - - - - X -
1:r
2:r
3:r
4:r
5:r
6:r
put orangutan o1 1
1:o
2:r
3:r
4:r
5:r
6:r
put orangutan o2 2
1:o
2:o
3:r
4:r
5:r
6:r
put lpanda p1 6
1:o
2:o
3:r
4:r
5:r
6:p
start
1:o
2:o
```



```
3:r
4:r
5:r
6:p
make a move o1
move o1 3
1:o
2:r
3:o
4:r
5:r
6:p
make a move o2
move o2 2
1:r
2:o
3:o
4:r
5:r
6:p
make a move o1
move o1 4
1:r
2:o
3:r
4:o
5:p
6:r
make a move o2
move o2 3
1:r
2:r
3:o
4:o
5:p
6:r
make a move o1
move o1 5
```



```

1:r
2:r
3:o
4:p
5:o
6:r
make a move o2

```

### 8.2.5

#### 8.2.6 Orangutan with Panda chain goes through a Wardrobe

- **Leírás**

Egy Orangutan utána befűzött Pandakkal megy a pályán, majd egy Wardrobe-hoz ér, amibe belép, és a pályán elhelyezett másik Wardrobe-ból lép ki, ez után addig lép, amíg a teljes panda lánc át nem ment a Wardrobe-on.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt ellenőrzi, hogy egy Orangutan utána befűzött Pandakkal láncsal képes-e áthaladni egy Wardrobe-on

- **Bemenet**

```

cm test4.flr
put orangutan o1 2
put lpanda 11 3
put lpanda 12 4
chain o1 11
chain 11 12
start
move o1 1
move o1 4
move o1 3

```

- **Elvárt kimenet**

```

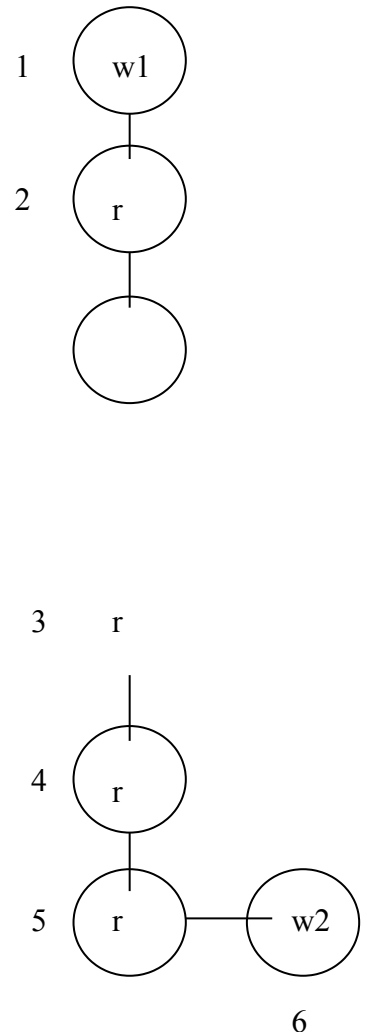
cm test4.flr
0 1 2 3 4 5 6
1  _ X  _  _  _  _
2  X  _  X  _  _  _
3  _  X  _  X  _  _
4  _  _  X  _  X  _
5  _  _  _  X  _  X
6  _  _  _  _  X  _
1:w
2:r
3:r
4:r
5:r
6:w
put orangutan o1 2
1:w
2:o
3:r
4:r
5:r
6:w
put lpanda 11 3
1:w
2:o
3:l
4:r
5:r
6:w
put lpanda 12 4
1:w
2:o

```

```

3:l
4:l
5:r
6:w
chain o1 11
chain 11 12
start
1:w
2:o
3:l
4:l
5:r
6:w
make a move o1
move o1 1
1:w
2:l
3:l
4:r
5:o
6:w
make a move o1
move o1 4
1:w
2:l
3:r
4:o
5:l
6:w
make a move o1
move o1 3

```



```
1:w  
2:r  
3:o  
4:l  
5:l  
6:w  
make a move o1
```

### 8.2.7 Armchair notifies some randomly stepping pandas

- **Leírás**

A pályán el van helyezve egy Armchair, aminek az értesítési ideje 3 lépés, az orangután 3szór kell, hogy lépjen, így 3 kör letelte után az Armchair értesíti a körülötte lévő pandákat. Az Armchair mellett található 2 panda egy JumpingPanda és egy LazyPanda. Ezek közül a LazyPanda el fog láthatólag el fog tűnni a pályáról, mivel az Armchair-ben ül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrzi a játék azon funkcióját, hogy az Armchair tárgy interakcióba tud lépni egy LazyPanda-val.

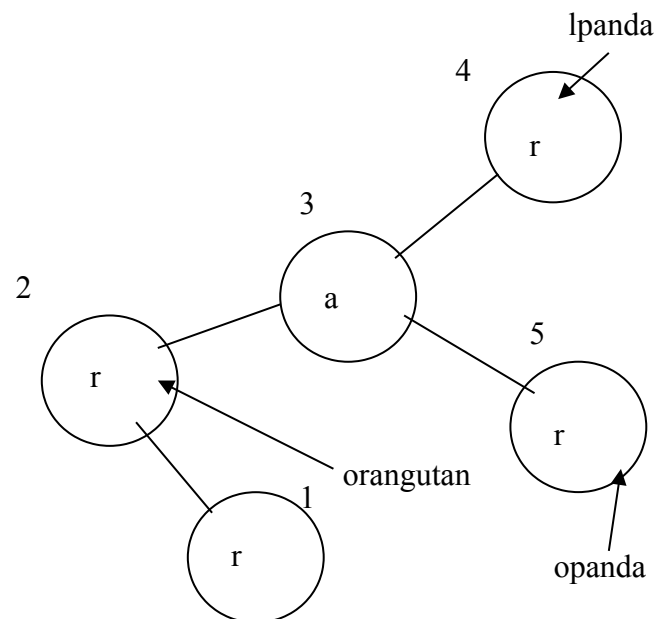
- **Bemenet**

```
cm test5.flr
put orangutan o1 2
put lpanda l1 4
put jpanda j1 5
start
move o1 1
move o1 2
move o1 1
```

- **Elvárt kimenet**

```
cm test5.flr
0 1 2 3 4 5
1 _ X _ _ _
2 X _ X _ _
3 _ X _ X X
4 _ _ X _ _
5 _ _ X _ _
```

```
1:r
2:r
3:a3
4:r
5:r
put orangutan o1 2
1:r
2:o
3:a3
4:r
5:r
put lpanda l1 4
1:r
2:o
3:a3
4:l
5:r
put jpanda j1 5
1:r
2:o
3:a3
4:l
5:j
start
1:r
2:o
3:a3
4:l
5:j
make a move o1
move o1 1
1:o
2:r
3:a2
4:l
5:j
```



```
make a move o1
move o1 2
1:r
2:o
3:a1
4:l
5:j
make a move o1
move o1 1
1:o
2:r
3:a0
4:r
5:j
make a move o1
```

### 8.2.8 Lonely Orangutan walking around a broken tile and breaks it, then the game ends

- **Leírás**

*A pályán egyetlen orangutánnal tud a játékos két csempe között lépkedni. Amint a csempe eltörik, az orangután meghal és így a játék véget ér.*

- **Ellenőrzött funkcionalitás, várható hibahelyek**

*A teszt ellenőrzi, hogy a játékos képes léptetni az orángutánt, a törött csempe élettartama tényleg csökken, ha rálépnek, illetve, ha a törött csempe eltörik, akkor az orángután tényleg meghal-e.*

- **Bemenet**

[illegible]

- **Elvárt kimenet**

cm test7.flr	2:o 11	1:r
0 1 2	move o1 1	2: 0
1 _ X	1:o	
2 X _	2:b 11	
1:r	move o1 2	
2:b 20	1:r	
put orangutan 01 2	2:B 10	
1:r	move o1 1	
2:o 20	1:o	
start	2:b 10	
1:r	move o1 2	
2:o 20	1:r	
move o1 1	2:B 9	
1:o	move o1 1	
2:b 20	1:o	
move o1 2	2:b 9	
1:r	move o1 2	
2:o 19	1:r	
move o1 1	2:B 8	
1:o	move o1 1	
2:b 19	1:o	
move o1 2	2:b 8	
1:r	move o1 2	
2:o 18	1:r	
move o1 1	2:B 7	
1:o	move o1 1	
2:b 18	1:o	
move o1 2	2:b 7	
1:r	move o1 2	
2:o 17	1:r	
move o1 1	2:B 6	
1:o	move o1 1	
2:b 17	1:o	
move o1 2	2:b 6	
1:r	move o1 2	
2:o 16	1:r	
move o1 1	2:B 5	
1:o	move o1 1	
2:b 16	1:o	
move o1 2	2:b 5	
1:r	move o1 2	
2:o 15	1:r	
move o1 1	2:B 4	
1:o	move o1 1	
2:b 15	1:o	
move o1 2	2:b 4	
1:r	move o1 2	
2:o 14	1:r	
move o1 1	2:B 3	
1:o	move o1 1	
2:b 14	1:o	
move o1 2	2:b 3	
1:r	move o1 2	
2:o 13	1:r	
move o1 1	2:B 2	
1:o	move o1 1	
2:b 13	1:o	
move o1 2	2:b 2	
1:r	move o1 2	
2:o 12	1:r	
move o1 1	2:B 1	
1:o	move o1 1	
2:b 12	1:o	
move o1 2	2:b 1	
1:r	move o1 2	

### 8.2.9 GameMachine scares a panda

- **Leírás**

A pályán egy orángutánnal és a hozzá tartozó pandalánccal tud a játékos lépkedni. Található még a pályán egy játékgép, ami 2 körönként csilingel, ezzel megijeszti a pandaláncban lévő, erre hajlamos pandát, ami ennek hatására elengedi a szomszédjait, jelen esetben az orángutánt, illetve a mögötte álló pandát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Megijesztett panda valóban elengedi-e a szomszédjai kezét.

- **Bemenet**

```
cm test8.flr
put orangutan o1 3
put spanda sp 2
put lpanda lp 1
start
move o1 4
move o1 5
move o1 6
```

- **Elvárt kimenet**

```
cm test8.flr          1:l
0 1 2 3 4 5 6 7      2:s
1 _ X _ _ _ _ _      3:o
2 X _ X _ _ _ _      4:r
3 _ X _ X _ _ _      5:r
4 _ _ X _ X _ X      6:r
5 _ _ _ X _ X _      7:g
6 _ _ _ _ X _ _      move o1 4
7 _ _ _ _ X _ _      1:r
1:r                  2:l
2:r                  3:s
3:r                  4:o
4:r                  5:r
5:r                  6:r
6:r                  7:g
7:g                  move o1 5
put orangutan o1      1:r
3                      2:r
1:r                  3:l
2:r                  4:s
3:o                  5:o
4:r                  6:r
5:r                  7:g
6:r                  move o1 6
7:g                  1:r
put spanda sp 2       2:l
1:r                  3:r
2:s                  4:s
3:o                  5:r
4:r                  6:o
5:r                  7:g
6:r
7:g
put lpanda lp 1       1:l
1:l
2:s
3:o
4:r
5:r
6:r
7:g
start
```

## 8.2.10 Orangutan steals Pandas from an Orangutan without Panda-chain

- **Leírás**

A pályán két gorillát irányíthat a játékos, melyek közül csak az egyiknek van pandalánca(egy panda). A teszt alatt az egyedüli orángután nekimegy a másik orángután láncának, és így elloppja azt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az orángután láncolt pandának ütközve tényleg el tudja-e lopni azt.

- **Bemenet**

```
cm test9.flr
put orangutan o1 3
put orangutan o2 2
put lpanda lp 1
start
move o1 5
move o2 4
move o1 7
move o2 5
move o1 8
move o2 6
```

- **Elvárt kimenet**

cm test9.flr	4:r	7:o
0 1 2 3 4 5 6 7 8	5:r	8:r
1 _ X _ _ _ _ _	6:r	move o2 5
2 X _ _ _ X _ _	7:r	1:r
3 _ _ _ X _ _ _	8:r	2:r
4 _ _ X _ X _ _	start	3:r
5 _ X _ _ _ X X _	1:l	4:l
6 _ _ _ _ X _ _	2:o	5:o
7 _ _ _ _ X _ X	3:o	6:r
8 _ _ _ _ _ X _	4:r	7:o
1:r	5:r	8:r
2:r	6:r	move o1 8
3:r	7:r	1:r
4:r	8:r	2:r
5:r	move o1 5	3:r
6:r	1:r	4:l
7:r	2:l	5:o
8:r	3:o	6:r
put orangutan o2 3	4:r	7:r
1:r	5:o	8:o
2:r	6:r	move o2 6
3:o	7:r	1:r
4:r	8:r	2:r
5:r	move o2 4	3:r
6:r	1:r	4:r
7:r	2:l	5:l
8:r	3:r	6:o
put orangutan o1 2	4:o	7:r
1:r	5:o	8:o
2:o	6:r	
3:o	7:r	
4:r	8:r	
5:r	move o1 7	
6:r	1:r	
7:r	2:r	
8:r	3:r	
put lpanda lp 1	4:o	
1:l	5:l	
2:o	6:r	
3:o		

### 8.2.11 Orangutan hits a chained Panda from another chain

- **Leírás**

A pályán két orángután található, mindkettőnek van egy egyelemű pandalánca. A teszt során az egyik orángután nekimegy a másik orángután láncának, így elrabolva azt a pandát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Lépés láncsal, befűzött panda saját láncba fűzése

- **Bemenet**

```
cm test 10.flr
put orangutan o1 2
put orangutan o2 4
put lpanda lp1 1
put lpanda lp2 3
start
move o1 6
move o2 5
move o1 8
move o2 6
move o1 9
move o2 7
```

- **Elvárt kimenet**

cm test 10.flr	3:r	7:r
0 1 2 3 4 5 6 7 8 9	4:o	8:r
1 _ X _ _ _ _ _ _	5:r	9:r
2 X _ _ _ _ X _ _	6:r	move o1 8
3 _ _ _ X _ _ _ _	7:r	1:r
4 _ _ X _ X _ _ _	8:r	2:r
5 _ _ _ X _ X _ _	9:r	3:r
6 _ X _ _ X _ X _	put lpanda lp2 3	4:l
7 _ _ _ _ _ X _ _	1:l	5:o
8 _ _ _ _ _ X _ X	2:o	6:l
9 _ _ _ _ _ _ X _	3:l	7:r
1:r	4:o	8:o
2:r	5:r	9:r
3:r	6:r	move o2 6
4:r	7:r	1:r
5:r	8:r	2:r
6:r	9:r	3:r
7:r	start	4:l
8:r	1:l	5:l
9:r	2:o	6:o
put orangutan o1 2	3:l	7:r
1:r	4:o	8:o
2:o	5:r	9:r
3:r	6:r	move o1 9
4:r	7:r	1:r
5:r	8:r	2:r
6:r	9:r	3:r
7:r	move o1 6	4:l
8:r	1:r	5:l
9:r	2:l	6:o
put orangutan o2 4	3:l	7:r
1:r	4:o	8:r
2:o	5:r	9:o
3:r	6:o	move o2 7
4:o	7:r	1:r
5:r	8:r	2:r
6:r	9:r	3:r
7:r	move o2 5	4:r
8:r	1:r	5:l
9:r	2:l	6:l
put lpanda lp1 1	3:r	7:o
1:l	4:l	8:r
2:o	5:o	9:o
	6:o	



### 8.1.1 VendingMachine makes JumpingPanda jump

- **Leírás**

A pályán el van helyezve egy VendingMachine, ami a következő körben jelezni fog a környező JumpingPanda'knak, hogy ugorjanak. A VendingMachine szomszédságába többféle pandát teszünk, mindegyiket egy-egy BrokenTile-ra.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

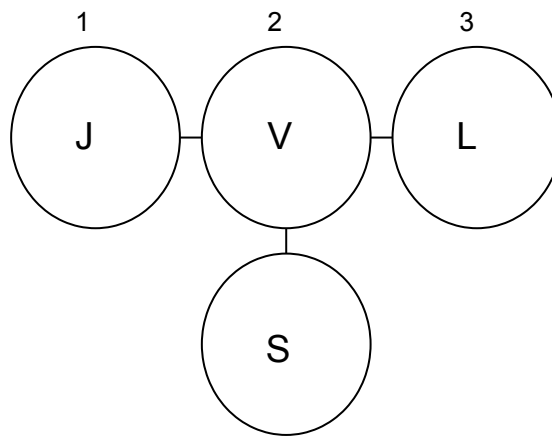
Ellenőrzi, hogy a VendingMachine jelzésére ténylegesen csak a JumpingPanda reagált-e, így csak az alatta lévő BrokenTile élete csökken eggyel. (ezt a kimeneten ténylegesen nem látjuk)

- **Bemenet**

```
cm test11.flr
put jpanda j1 1
put lpanda l1 3
put spanda s1 4
start
```

- **Elvárt kimenet**

```
cm test11.flr
  1  2  3  4
1  _  X  _  _
2  X  _  X  X
3  _  X  _  _
4  _  X  _  _
1:b
2:v
3:b
4:b
put jpanda j1 1
1:j
2:v
3:b
4:b
put lpanda l1 3
1:j
2:v
3:l
4:b
put lpanda s1 4
1:j
2:v
3:l
4:s
start
1:j
2:v
3:l
4:s
```



### 8.1.2

### 8.1.3 Orangutan hits its own Panda chain

- **Leírás**

A pálya 5 törhetetlen csempéből áll, amelyet feltöltünk 1 orángutánnal és 3 vele láncba kapcsolt LazyPandával (vagy bármilyen más pandával) 1 csempét üresen hagyva. Az orángutánt ezután ráléptetjük a láncának utolsó pandájára és lépünk még egyet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt kell látnunk, hogy a plusz egy lépésnél már nem a régi láncát húzza az orángután, hanem az újat.

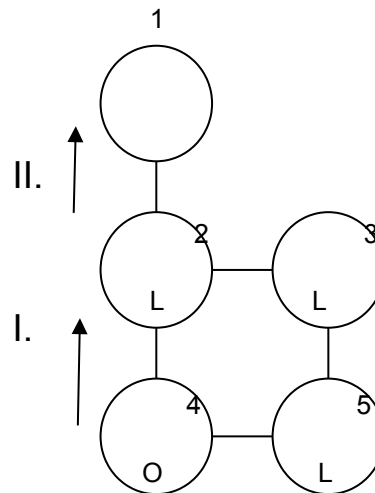
- **Bemenet**

```
cm test12.flr
put lpanda 11 2
put lpanda 12 3
put orangutan o1 4
put lpanda 13 5
chain o1 13
chain 13 12
chain 12 11
start
move o1 2
move o1 1
```

- **Elvárt kimenet**

```
cm test12.flr
  1  2  3  4  5
1  -  X  -  -  -
2  X  -  X  X  -
3  -  X  -  -  X
4  -  X  -  -  X
5  -  -  X  X  -
1:r
2:r
3:r
4:r
5:r
put lpanda 11 2
1:r
2:l
3:r
4:r
5:r
put lpanda 12 3
1:r
2:l
3:l
4:r
5:r
put orangutan o1 4
1:r
2:l
3:l
4:o
5:r
put lpanda 13 5
1:r
2:l
3:l
4:o
5:l
chain o1 13
chain 13 12
chain 12 11
start
```

9/14/2020



```
1:r
2:l
3:l
4:o
5:l
make a move o1
move o1 2
1:r
2:o
3:l
4:l
5:l
make a move o1
move o1 1
1:o
2:l
3:l
4:r
5:l
make a move o1
```

### 8.1.4

### 8.1.5 Orangutan exits with two Pandas

- **Leírás**

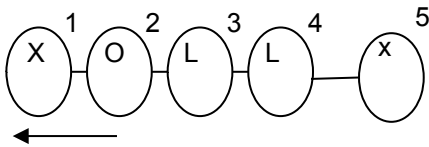
A pálya álljon 3 üres törhetetlen csempéből és egy kijáratból. A három üres csempére tegyük le az orángutánt a láncba kapcsolt két laza pandájával (vagy bármilyen más típusú pandájával). Az orángutánt ezután ráléptetjük a kijárat csempéjére.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az orángután a lépésével közelebb húzza a pandáit a kijáratához, eltűnik és felbukkan a bejáraton. Ezután a kijárat minden körben magához hívja a lánc következő pandáját és mivel semmilyen pandákat zavaró tényező nincs, így két plusz kör végeztével azt kell látnunk, hogy csak az orángutánunk van a pályán.

- **Bemenet**

```
cm test13.flr
put orangutan o1 2
```



```
put lpanda l1 3
put lpanda l2 4
chain o1 l1
chain l1 l2
start
move o1 1
move o1 4
move o1 3
```

- **Elvárt kimenet**

```
cm test13.flr
  1  2  3  4  5
1  _  X  _  _  _
2  X  _  X  _  _
3  _  X  _  X  _
4  _  _  X  _  X
5  _  _  _  X  _
1:X
2:r
3:r
4:r
5:x
put orangutan o1 2
1:X
2:o
3:r
4:r
5:x
put lpanda l1 3
1:X
2:o
3:l
4:r
5:x
put lpanda l2 4
1:X
2:o
3:l
```

```
4:l
5:x
chain o1 l1
chain l1 l2
start
1:X
2:o
3:l
4:l
5:x
make a move o1
move o1 1
1:X
2:l
3:l
4:r
5:o
make a move o1
move o1 4
1:X
2:l
```

```
3:r
4:o
5:x
make a move o1
move o1 3
1:X
2:r
3:o
4:r
5:x
make a move o1
```

### 8.1.6 Orangutan is exiting the floor with a Panda chain meanwhile a GameMachine scares one ScaredPanda in it who leaves the chain with all the Pandas behind.

- **Leírás**

A pályát 8 mezőből alkotjuk, egy bejáratból, egy kijáratból, egy játékgépből és 5 törhetetlen csempéből. A játékgép a következő kör végén jelez. A pályára lerakunk egy orángutánt a kijárat szomszédságába és 4 vele láncolt pandát (2 Scared, 1 Jumping, 1 Lazy) úgy, hogy az orángután a kijáratra lépés következtében odahúzza az egyik ScaredPandát a játékgép (VendingMachine) közelségébe.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a VendingMachine szomszédságába húzott panda és az amögött lévő pandák leszakadnak-e a láncból a jelzés hatására, azaz a következő körökben nem indulnak meg maguktól a kijárat felé, minthogy az alapesetben történne.

- **Bemenet**

```
cm test14.flr
put jpanda j1 2
put spanda s1 3
put lpanda l1 4
put spanda s2 5
put orangutan o1 6
chain o1 s2
chain s2 l1
chain l1 s1
chain s1 j1
start
move o1 7
move o1 2
```

- **Elvárt kimenet**

```
cm test14.flr
  1  2  3  4  5  6  7  8
1  _  X  _  _  _  _  _
2  X  _  X  _  _  _  _
3  _  X  _  X  _  _  _
4  _  _  X  _  X  _  X
5  _  _  _  X  _  X  _
6  _  _  _  _  X  _  X
7  _  _  _  _  _  X  _
8  _  _  _  X  _  _  _
```

```
1:x
2:r
3:r
4:r
5:r
6:r
7:X
8:v
put jpanda j1 2
1:x
2:j
3:r
4:r
5:r
6:r
7:X
8:v
put spanda s1 3
```

9/14/2020

```
1:x
2:j
3:s
4:r
5:r
6:r
7:X
8:v
put lpanda l1 4
1:x
2:j
3:s
4:l
5:r
6:r
7:X
8:v
put spanda s2 5
```

```
1:x
2:j
3:s
4:l
5:s
6:r
7:X
8:v
put orangutan o1 6
1:x
2:j
3:s
4:l
5:s
6:o
7:X
8:v
chain o1 s2
```

```

chain s2 l1
chain l1 s1
chain s1 j1
start
make a move o1
move o1 7
1:o
2:r
3:j
4:s
5:l
6:s
7:X
8:v
make a move o1
move o1 2
1:x
2:o
3:j
4:s
5:r
6:l
7:X
8:v
make a move o1

```

### 8.1.7 Panda breaks a tile with its move

- Ellenőrzött funkcionalitás, várható hibahelyek

*A mozgás valóban megvalósul-e, a csempe összetörik-e?*

- Bemenet

cm map.txt  
map.txt tartalma:

```

1 2 3 4
1 - x - -
2 x - - -
3 - - - x
4 - - x -
1:r
2:b
3:r
4:r

```

```

put jpanda pandicsek 1
put orangutan orcsi 3
  move orcsi 4
  move orcsi 3
  move orcsi 4
  move orcsi 3
  move orcsi 4
  move orcsi 3

```

```

move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3

```

```

move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3

```

```

move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4
move orcsi 3
move orcsi 4

```

- 

- **Elvárt kimenet**

cm map.txt

```

  1 2 3 4
1 - x - -
2 x - - -
3 - - - x
4 - - x -
1:r
2:b
3:r
4:r

```

```

put jpanda
pandicsek 1

```

```

1:j
2:b
3:r
4:r

```

```

put orangutan
orsci 3

```

```

1:j
2:b
3:o
4:r

```

```

move orcsi 4

```

```

1:j
2:b
3:r
4:o

```

```

1:r
2:j
3:r
4:o

```

```

move orcsi 3
1:r
2:j
3:o
4:r

```

```

1:j
2:b

```

```

3:o
4:r

```

```

move orcsi 4

```

```

1:j
2:b
3:r
4:o

```

```

1:r
2:j
3:r
4:o

```

```

move orcsi 3

```

```

1:r
2:j
3:o

```

4:r	4:o	4:r
1:j	1:r	1:j
2:b	2:j	2:b
3:o	3:r	3:o
4:r	4:o	4:r
move orcsi 4	move orcsi 3	move orcsi 4
1:j	1:r	1:j
2:b	2:j	2:b
3:r	3:o	3:r
4:o	4:r	4:o
1:r	1:j	1:r
2:j	2:b	2:j
3:r	3:o	3:r
4:o	4:r	4:o
move orcsi 3	move orcsi 4	move orcsi 3
1:r	1:j	1:r
2:j	2:b	2:j
3:o	3:r	3:o
4:r	4:o	4:r
1:j	1:r	1:j
2:b	2:j	2:b
3:o	3:r	3:o
4:r	4:o	4:r
move orcsi 4	move orcsi 3	move orcsi 4
1:j	1:r	1:j
2:b	2:j	2:b
3:r	3:o	3:r
4:o	4:r	4:o
1:r	1:j	1:r
2:j	2:b	2:j
3:r	3:o	3:r
4:o	4:r	4:o
move orcsi 3	move orcsi 4	move orcsi 3
1:r	1:j	1:r
2:j	2:b	2:j
3:o	3:r	3:o
4:r	4:o	4:r
1:j	1:r	1:j
2:b	2:j	2:b
3:o	3:r	3:o
4:r	4:o	4:r
move orcsi 4	move orcsi 3	move orcsi 4
1:j	1:r	1:j
2:b	2:j	2:b
3:r	3:o	3:r

4:o	4:r	4:o
1:r	1:j	1:r
2:j	2:b	2:j
3:r	3:o	3:r
4:o	4:r	4:o
move orcsi 3	move orcsi 4	move orcsi 3
1:r	1:j	1:r
2:j	2:b	2:j
3:o	3:r	3:o
4:r	4:o	4:r
1:j	1:r	1:j
2:b	2:j	2:b
3:o	3:r	3:o
4:r	4:o	4:r
move orcsi 4	move orcsi 3	move orcsi 4
1:j	1:r	1:j
2:b	2:j	2:b
3:r	3:o	3:r
4:o	4:r	4:o
1:r	1:j	1:r
2:j	2:b	2:j
3:r	3:o	3:r
4:o	4:r	4:o
move orcsi 3	move orcsi 4	move orcsi 3
1:r	1:j	1:r
2:j	2:b	2:j
3:o	3:r	3:o
4:r	4:o	4:r
1:j	1:r	1:j
2:b	2:j	2:b
3:o	3:r	3:o
4:r	4:o	4:r
move orcsi 4	move orcsi 3	move orcsi 4
1:j	1:r	1:j
2:b	2:j	2:b
3:r	3:o	3:r
4:o	4:r	4:o
1:r	1:j	1:r
2:j	2:b	2:j
3:r	3:o	3:r
4:o	4:r	4:o
move orcsi 3	move orcsi 4	move orcsi 3
1:r	1:j	1:r
2:j	2:b	2:j
3:o	3:r	3:o



4:r

3:r

4:o

1:j

2:b

1:r

3:o

3:r

4:r

4:o

move orcsi 4

1:j

2:b

3:r

4:o

1:r

2:j

3:r

4:o

move orcsi 3

1:r

2:j

3:o

4:r

1:j

2:b

3:o

4:r

move orcsi 4

1:j

2:b

3:r

4:o

1:r

2:j

3:r

4:o

move orcsi 3

1:r

2:j

3:o

4:r

1:j

2:b

3:o

4:r

move orcsi 4

1:j

2:b

## 8.2.12 8.1.2 Teszteset17

- Leírás

Last Orangutan dies

- Ellenőrzött funkcionalitás, várható hibahelyek

*A játékban lévő utolsó orángután is meghal. Véget ér-e a játék?*

- Bemenet

*cm map.txt*

map.txt tartalma:

```

      1 2 3 4
1 - x - -
2 x - - -
3 - - - x
4 - - x -
1:r
2:b
3:r
4:r

```

put orangutan orcsi 1

put jpanda pancsi 3

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

move orcsi 2  
move orcsi 1

```
move orcsi 2  
move orcsi 1
```

```
move orcsi 2
```

- **Elvárt kimenet**

```
cm map.txt
```

```
    1 2 3 4  
1 - x - -  
2 x - - -  
3 - - - x  
4 - - x -  
1:r  
2:b  
3:r  
4:r
```

orcsi 1	put orangutan	2:b	4:r
1:o		3:r	move orcsi 2
2:b		4:j	1:r
3:r		1:o	2:o
4:r		2:b	3:j
		3:j	4:r
pancsi 3	put jpanda	4:r	1:r
1:o		move orcsi 2	2:o
2:b		1:r	3:r
3:j		2:o	4:j
4:r		3:j	
		4:r	move orcsi 1
			1:o
		1:r	2:b
		2:o	3:r
		3:r	4:j
		4:j	
			1:o
			2:b
			3:j
			4:r
			move orcsi 2
			1:r
			2:o
			3:j
			4:r
			1:r
			2:o
			3:j
			4:j
			1:o
			2:b
			3:r
			4:j
			1:o
			2:b
			3:j
			4:r
			move orcsi 2
			1:r
			2:o
			3:j
			4:r

1:r  
2:o  
3:r  
4:j

move orcsi 1

1:o  
2:b  
3:r  
4:j

1:o  
2:b  
3:j  
4:r

move orcsi 2

1:r  
2:o  
3:j  
4:r

1:r  
2:o  
3:r  
4:j

move orcsi 1

1:o  
2:b  
3:r  
4:j

1:o  
2:b  
3:j  
4:r

move orcsi 2

1:r  
2:o  
3:j  
4:r

1:r  
2:o  
3:r  
4:j

move orcsi 1

1:o  
2:b

3:r  
4:j

1:o  
2:b

3:j

4:r

move orcsi 2

1:r  
2:o  
3:j  
4:r

1:r  
2:o  
3:r  
4:j

move orcsi 1

1:o  
2:b  
3:r  
4:j

1:o  
2:b  
3:j  
4:r

move orcsi 2

1:r  
2:o  
3:j  
4:r

1:r  
2:o  
3:r  
4:j

move orcsi 1

1:o  
2:b  
3:r  
4:j

1:o  
2:b  
3:j  
4:r

move orcsi 2

1:r

2:o  
3:j  
4:r

1:r  
2:o  
3:r  
4:j

move orcsi 1

1:o

2:b  
3:r  
4:j

1:o  
2:b  
3:j  
4:r

move orcsi 2

1:r

2:o  
3:j  
4:r

1:r  
2:o  
3:r  
4:j

move orcsi 1

1:o

2:b  
3:r  
4:j

1:o  
2:b  
3:j  
4:r

move orcsi 2

1:r

2:o  
3:j  
4:r

1:r  
2:o  
3:r

4:j	1:o	4:r
move orcsi 1	2:b	1:r
1:o	3:j	2:o
2:b	4:r	3:r
3:r	move orcsi 2	4:j
4:j	1:r	
	2:o	move orcsi 1
1:o	3:j	1:o
2:b	4:r	2:b
3:j	1:r	3:r
4:r	2:o	4:j
move orcsi 2	3:r	
1:r	4:j	1:o
2:o		2:b
3:j	move orcsi 1	3:j
4:r	1:o	4:r
	2:b	<i>move orcsi 2</i>
1:r	3:r	1:r
2:o	4:j	3:j
3:r		4:r
4:j	1:o	
	2:b	
move orcsi 1	3:j	
1:o	4:r	
2:b	move orcsi 2	
3:r	1:r	
4:j	2:o	
	3:j	
1:o	4:r	
2:b		
3:j	1:r	
4:r	2:o	
move orcsi 2	3:r	
1:r	4:j	
2:o		
3:j	move orcsi 1	
4:r	1:o	
	2:b	
1:r	3:r	
2:o	4:j	
3:r		
4:j	1:o	
	2:b	
move orcsi 1	3:j	
1:o	4:r	
2:b	move orcsi 2	
3:r	1:r	
4:j	2:o	
	3:j	

### 8.2.13 8.1.3 Teszteset18

- **Leírás**

## Last Panda dies

- **Ellenőrzött funkcionalitás, várható hibahelyek**

*A játékban lévő utolsó orángután is meghal. Véget ér-e a játék?*

- **Bemenet**

```
cm map.txt
map.txt tartalma:
```

	1	2	3	4
1	-	x	-	-
2	x	-	-	-
3	-	-	-	x
4	-	-	x	-
1:r				
2:b				
3:r				
4:r				

put jpanda  
pandicsek 1  
put orangutan  
oresi 3

	move orcsi 3
put jpanda	move orcsi 4
	move orcsi 3
put orangutan	move orcsi 4
	move orcsi 3
	move orcsi 4
move orcsi 4	move orcsi 3
move orcsi 3	move orcsi 4
move orcsi 4	move orcsi 3
move orcsi 3	move orcsi 4
move orcsi 4	move orcsi 3
move orcsi 3	move orcsi 4
move orcsi 4	move orcsi 3
move orcsi 3	move orcsi 4
move orcsi 4	move orcsi 3
move orcsi 3	move orcsi 4
move orcsi 4	move orcsi 3
move orcsi 3	move orcsi 4
move orcsi 4	move orcsi 3
move orcsi 3	
move orcsi 4	move orcsi 4
move orcsi 3	
move orcsi 4	
move orcsi 3	
move orcsi 4	

**Elvárt kimenet**

cm map.txt

1 2 3 4

1 - x - -

2 x - - -

3 - - - x

4 - - x -

1:r

2:b

3:r

4:r

put jpanda

pandicsek 1

1:j

2:b

3:r

4:r

put orangutan

orcsi 3

1:j

2:b

3:o

4:r

move orcsi 4

1:j

2:b

3:r

4:o

1:r

2:j

3:r

4:o

move orcsi 3

1:r

2:j

3:o

4:r

1:j

2:b

3:o

4:r

move orcsi 4

1:j

2:b

3:r

4:o

1:r

2:j

3:r

4:o

move orcsi 3

1:r

2:j

3:o

4:r

1:j

2:b

3:o

4:r

move orcsi 4

1:j

2:b

3:r

4:o

1:r

2:j

3:r

4:o

move orcsi 3

1:r

2:j

3:o

4:r

1:j

2:b

3:o

4:r

move orcsi 4

1:j

2:b

3:r

4:o

1:r

2:j

3:r

4:o

move orcsi 3

1:r

2:j

3:o

4:r

1:j

2:b

3:o

4:r

move orcsi 4

1:j

2:b

3:r

4:o

1:r

2:j

3:r

4:o

move orcsi 3

1:r

2:j

3:o

4:r

1:j

2:b

3:o

4:r

move orcsi 4

1:j

2:b

3:r

4:o



1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
2:j  
3:r  
4:o  
move orcsi 3

1:r  
2:j  
3:o  
4:r

1:j  
2:b  
3:o  
4:r  
move orcsi 4

1:j  
2:b  
3:r  
4:o

1:r  
3:r  
4

## 8.2.14 8.1.4 Teszteset19

- **Leírás**

Orangutan releases its PandaChain

- **Ellenőrzött funkcionalitás, várható hibahelyek**

*Az orángután elengedi a mögötte álló pandaláncot*

- **Bemenet**

cm map.txt  
map.txt tartalma:

```

1 2 3 4
1 - x - x
2 x - x -
3 - x - -
4 x - - -
1:r
2:r
3:r
4:r
```

```

put orangutan orcsi 1
put jpanda pandicsek 2
put jpanda pancser 3
put orangutan ogre 4
chain orcsi pandicsek
chain pandicsek pancser
move orcsi 4
```

- **Elvárt kimenet**

cm map.txt

```

1 2 3 4
1 - x - x
2 x - x -
3 - x - -
4 x - - -
1:r
2:r
3:r
4:r
```

put orangutan orcsi 1

```

1:o
2:r
3:r
4:r
```

pandicsek 2

put jpanda

```

1:o
2:j
3:r
4:r
```

pancser 3

```

2:j
3:j
4:r
```

put jpanda

```

1:o
```

```

ogre 4      put orangutan
            1:o
            2:j
            3:j
            4:o
            chain orcsi
pandicsek   chain pandicsek
pancser     move orcsi 4
            1:o
            2:j
            3:j
            4:o

```

### 8.2.15 8.1.5 Teszteset20

- **Leírás**  
Orangutan brings PandaChain over broken tile
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
*Egy orángután átvezet egy pandaláncot egy törékeny csempén.*
- **Bemenet**

```

cm map.txt
map.txt tartalma:
    1 2 3 4 5
    1 - x - - -
    2 x - x - -
    3 - x - x -
    4 - - x - x
    5 - - - x -
    1:r
    2:b
    3:r
    4:r
    5:r
    put orangutan orcsi 3
    put jpanda pancsi 4
    put jpanda pancser 5
    chain orangutan pancsi
    chain pancsi pancser
    move orcsi 2
    move orcsi 1

```

- **Elvárt kimenet**

orcsi 3	put orangutan	4:r	2:b
		5:r	3:o
1:r		put jpanda	4:j
2:b	pancsi 4		5:r
3:o		1:r	

	put jpanda
pancser 5	1:r 2:b 3:o 4:j 5:j
	chain orangutan
pancsi	
	chain pancsi
pancser	
	move orcsi 2
	1:r 2:o 3:j 4:j 5:r
	move orcsi 1
	1:o 2:j 3:j 4:r 5:



### 8.3 A tesztelést támogató programok tervei

A program batch módon (pl. PowerShell segítségével) programozható. A tesztek úgy megvalósítottak, hogy bemenetként van szöveges file, és a program a console-ra dobja ki a megoldást. Ezt össze tudjuk hasonlítani egy egyszerű összehasonlító paranccsal (mint UNIX rendszerek alatt található diff parancs, windows-on FC vagy „findstr /V /G:input.txt output.txt >DiffResult.txt”) és amennyiben egyezik a várt kimenettel, akkor sikeres a teszt (OK). Ha a teszt nem sikertelen (FAIL), akkor kijelzi a program, hogy eltérést tapasztalt a sorban.

### 8.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2019.04.05 20:30	1 óra	Kecskés, Kiss, Kobál	<b>Értekezlet:</b> Feladatok gyors kiosztása hétvégére. Jövőbeli kódolási/implementációs problémák megbeszélése
2019.04.06 14:00	3.0 óra	Kobál	Bemeneti és kimeneti nyelv hibáinak javítása
2019.04.06 20:00	2.0 óra	Kobál	1-5 Tesztesetek elkészítése
2019.04.07 10:00	2.0 óra	Kiss	Megcsinálva a 16-20 tesztesetek a megbeszélteknek megfelelően
2019.04.07 13:00	2.0 óra	Martonosi	Kiosztott tesztesetek implementálása
2019.04.07 15:00	0.5 óra	Kecskés, Kiss, Kis, Kobál, Martonosi	<b>Értekezlet:</b> Kész feladatok ellenőrzése
2019.04.07 15:00	1.0 óra	Kecskés	A kiosztott feladatok, Osztálydiagramok és metódusok elkészítése
2019.04.07 18:00	3.5 óra	Kis	Tesztesetek: 11/12/13/14/15 (13. és 14. összevonva)
2019.04.07 20:30	1.0 óra	Martonosi	Tesztesetek szépítése, dokumentum szerkesztése
2019.04.07 23:30	2.0 óra	Kecskés	Formázás, és a napló kiegészítése, a végleges dokumentum összeállítása.

## 9. Prototípus beadása

### 9.1 Fordítási és futtatási útmutató

#### 9.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Animal.java	2456	2019.04.24 23:06	Az állatokat gyűjtő ösosztály
Armchair.java	1421	2019.04.23 23:53	Karosszék függvényeit tároló
AThing.java	1897	2019.04.24 23:06	Minden tárgy ösosztálya
BrokenTile.java	1093	2019.04.24 23:06	Törött csempe osztálya
Controller.java	3561	2019.04.23 23:53	A pandát irányítja (és később a pályát állítja össze)
Entry.java	1649	2019.04.24 23:06	Bejárat osztálya
Exit.java	2558	2019.04.24 23:06	Kijárat osztálya
Floor.java	4093	2019.04.22 15:59	A pálya osztálya, benne tárolva a csempékkel, pandákkal, orángutánokkal
FloorMakingProgram.java	10635	2019.04.24 23:06	A teszteseteket (.flr fileokat) generáló program
GameMachine.java	1432	2019.04.22 15:59	Játékgép osztálya
IPandaEffective.java	212	2019.04.22 15:59	Az hatással rendelkező (szomszéd csempét értesítő effekteket tartalmazó) osztályok interface-e
JumpingPanda.java	805	2019.04.24 23:06	Ugráló panda osztálya
LazyPanda.java	1864	2019.04.24 23:06	Lusta panda osztálya
Orangutan.java	1985	2019.04.24 23:06	Orángután osztálya
Panda.java	2459	2019.04.24 23:06	Panda osztálya
Program.java	3718	2019.04.24 11:31:	A pályát beolvasó, állatot lehelyező/chainelő és controllert meghívó tényleges játékindító osztály
RegularTile.java	237	2019.04.22 15:59	Nem törős csempe osztálya
ScaredPanda.java	695	2019.04.24 23:06	Ijedős panda osztálya
Tile.java	6148	2019.04.24 23:06	Csempe ösosztálya
VendingMachine.java	1456	2019.04.24 23:06	Játékautomata osztálya
Wardrobe.java	2367	2019.04.22 15:59	szekrény osztálya
test1.flr	1499	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test2.flr	919	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test3.flr	671	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test4.flr	781	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test5.flr	721	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test6.flr	569	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test7.flr	827	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test8.flr	759	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test9.flr	803	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test10.flr	742	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test11.flr	637	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test12.flr	854	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test13.flr	948	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test14.flr	569	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test15.flr	583	2019.04.24 23:06	Számozott teszteset pályája szerializálva
test1.txt	378	2019.04.24 23:06	Számozott teszteset helyes kimenete
test2.txt	411	2019.04.24 23:06	Számozott teszteset helyes kimenete



test3.txt	650	2019.04.24 23:06	Számozott teszteset helyes kimenete
test4.txt	567	2019.04.24 10:51	Számozott teszteset helyes kimenete
test5.txt	431	2019.04.24 10:53	Számozott teszteset helyes kimenete
test6.txt	503	2019.04.24 11:07	Számozott teszteset helyes kimenete
test7.txt	646	2019.04.24 23:06	Számozott teszteset helyes kimenete
test8.txt	904	2019.04.24 11:17	Számozott teszteset helyes kimenete
test9.txt	1079	2019.04.24 23:06	Számozott teszteset helyes kimenete
test10.txt	375	2019.04.24 23:06	Számozott teszteset helyes kimenete
test11.txt	498	2019.04.24 11:33	Számozott teszteset helyes kimenete
test12.txt	454	2019.04.24 23:37	Számozott teszteset helyes kimenete
test13.txt	796	2019.04.24 23:06	Számozott teszteset helyes kimenete
test14.txt	223	2019.04.24 23:06	Számozott teszteset helyes kimenete
test15.txt	381	2019.04.24 23:06	Számozott teszteset helyes kimenete

### 9.1.2 Fordítás

A program futtatásához lent kell, hogy legyen a JDK (JavaSE1.8) környezet és be kell, hogy legyen építve a környezetbe.

(Windows-vezérlőpult-rendszer-speciális rendszerbeállítások-környezeti változók – Path – szerkesztés - JDK mappájának hozzáadása a mezőbe)

A forráskódot a github linken zip-ből kicsomagolva az src mappában találjuk.

A futtatáshoz szükséges parancsokat ebben a mappában a következő módon kell csinálni:

```
javac Program.java
```

```
java Program
```

### 9.1.3 Futtatás

## 9.2 Tesztek jegyzőkönyvei

### 9.2.1 Teszteset1 (Program starts and makes a map)

Tesztelő neve	Kobál Sándor
Teszt időpontja	2019.04.23 20:10

### 9.2.2 Teszteset2 (Entry generates two orangutans and player steps with them)

Tesztelő neve	Kobál Sándor
Teszt időpontja	2019.04.23 20:13

### 9.2.3 Teszteset3 (Two orangutans wandering on the floor and suddenly one orangutan sees a panda and hits it.)

Tesztelő neve	Kobál Sándor
Teszt időpontja	2019.04.23 20:18

### 9.2.4 Teszteset4 (Orangutan with Panda chain goes through a Wardrobe)

Tesztelő neve	Kecskés Péter
Teszt időpontja	2019.04.23 20:08

### 9.2.5 Teszteset5 (Armchair notifies some randomly stepping pandas)

Tesztelő neve	Kecskés Péter
Teszt időpontja	2019.04.23 20:12

**Teszteset6 (Lonely Orangutan walking around a broken tile and breaks it, then the game ends)**

Tesztelő neve	Martonosi Máté
Teszt időpontja	2019.04.23 20:10

**9.2.6 Teszteset7 (GameMachine scares a panda)**

Tesztelő neve	Martonosi Máté
Teszt időpontja	2019.04.23 20:15

**9.2.7 Teszteset8 (Orangutan steals Pandas from an Orangutan without Panda-chain)**

Tesztelő neve	Martonosi Máté
Teszt időpontja	2019.04.23 20:18

**9.2.8 Teszteset9 (Orangutan hits a chained Panda from another chain)**

Tesztelő neve	Martonosi Máté
Teszt időpontja	2019.04.23 20:20

**9.2.9 Teszteset10 (VendingMachine makes JumpingPanda jump)**

Tesztelő neve	Kis Máté Levente
Teszt időpontja	2019.04.23 20:10

**9.2.10 Teszteset11 (Orangutan hits its own Panda chain)**

Tesztelő neve	Kis Máté Levente
Teszt időpontja	2019.04.23 20:12

**9.2.11 Teszteset12 (Orangutan exits with two Pandas)**

Tesztelő neve	Kecskés Péter
Teszt időpontja	2019.04.23 20:10
Teszt eredménye	Sikertelen
Lehetséges hibaok	A kijárat controller része hibás
Változtatások	Controller javítása

**9.2.12 Teszteset13 (Orangutan is exiting the floor with a Panda chain meanwhile a GameMachine scares one ScaredPanda in it who leaves the chain with all the Pandas behind.)**

Tesztelő neve	Martonosi Máté
Teszt időpontja	2019.04.24 19:20

**9.2.13 Teszteset14 (Panda breaks a tile with its move)**

Tesztelő neve	Martonosi Máté
Teszt időpontja	2019.04.24 19:30

**9.2.14 Teszteset15 (Orangutan releases its PandaChain)**

<b>Tesztelő neve</b>	Kecskés Péter
<b>Teszt időpontja</b>	2019.04.24 19:20

### 9.3 Értékelés

A százalékok a github statok és a napló súlyozásával adódtak

Tag neve	Tag neptun	Munka százalékban
Kecskés Péter	XZ9I7F	23
Kis Máté Levente	H0OTLX	16
Kiss Előd	U479JN	15
Kobál Sándor	C4W3BP	23
Martonosi Máté	QOT4DC	23

### 9.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2019.04.13 17:00	0.5 óra	Kiss, Kobál, Martonosi	<b>Értekezlet:</b> Tesztesetek implementációjának kiosztása.
2019.04.17 19:00	3.0 óra	Kobál	A Floor osztály implementálása, szerializálás megvalósítása, példateszt elkészítése.
2019.04.18 11:00	2.0 óra	Kiss	Kiss E. megcsinálja a rá kiszabott tesztesetekhez tartozó pályákat.
2019.04.18 18:30	2 óra	Kecskés, Kobál, Martonosi	<b>Értekezlet:</b> A kapott feladatok egyesztetése, a kész kódok ellenőrzése és githubban akadt problémák elhárítása. (file exclude)
2019.04.18 20:00	1.5 óra	Kecskés	Kiszabott feladatok elkészítésének előkészítése.
2019.04.18 22:00	2.0 óra	Marotonosi	Bemenetértelmező megírása.
2019.04.19 13:00	1.0 óra	Kobál	1,2,3 szintek elkészítése.
2019.04.19 18:00	2.0 óra	Kobál	Controller implementálásának elkezdése.
2019.04.21 10:00	2.5 óra	Martonosi	Kiszabott tesztesetek megírása.
2019.04.22 10:00	1.5 óra	Martonosi	Valaki commitja miatt elcsesződött tesztesetek megjavítása.
2019.04.22 16:00	2.0 óra	Kecskés	Controller javítása.
2019.04.23 10:00	1.5 óra	Kis	Teszt 11/12/13/14
2019.04.23 20:00	4 óra	Kecskés, Kobál, Kiss, Kis, Martonosi	<b>Értekezlet:</b> A maradék kódok tesztelése és prototípus befejezése(nem sikerült).
2019.04.24 12:30	2.5 óra	Kecskés, Martonosi	Proto teljes átfutása és tesztek felének kijavítása, átdolgozása.
2019.04.24 18:30	4.5 óra	Kecskés, Martonosi	Proto délelőtt még hibásan maradt részeinek javítása, minden véglegesítése.
2019.04.24 23:30	1.0 óra	Kecskés	Dokumentum formázása véglegesítése (tesztesetek, ki és bementeti nyelv illesztése) és százalékok kiszámolása a csapatnak osztva.

### 9.4.1 Bemeneti nyelv

Az egyszerűség kedvéért az emeletet nem a játékosnak kell létrehoznia, csak ki kell választania a neki megfelelő előre megírt emeletek közül az egyiket.

#### *cm (create map)*

**Leírás:** Megkap paraméterként egy fájlt, amiben specifikálva van egy emelet leírása (csempék, szomszédjaik és a rajtuk elhelyezkedő tárgyak).

**Opciók:** cm <filename.flr>

Példa: cm map1.flr

map1.flr tartalma: Megfelel a Java API szerilizáló interfaceének.

Képernyőre kiírt tartalom fájl kiválasztása után:

0	1	2	3	4	5	6	7	8	9	10
1	—	—	—	—	X	—	—	—	X	X
2	—	—	X	—	—	—	—	X	—	X
3	—	X	—	X	—	—	X	X	X	—
4	—	—	X	—	X	X	—	—	—	—
5	X	—	—	X	—	—	—	—	—	—
6	—	—	—	X	—	—	—	—	—	—
7	—	—	X	—	—	—	—	—	—	—
8	—	X	X	—	—	—	—	—	—	—
9	X	—	X	—	—	—	—	—	—	X
10	X	X	—	—	—	—	—	—	X	—

1:w  
2:b20  
3:r  
4:r  
5:r  
6:x  
7:X  
8:w  
9:a3  
10:r

A pálya készítésekor, csak élettelen dolgok felvételét engedjük meg. Hiszen nem tudni pl., hogy egy állat alatt milyen típusú csempe van, így a pálya megadásához több információra lenne szükségünk, ami miatt kevésbé lehetne stuktúrált a bementünk. (élettelen tárgyak alatt biztosan törhetetlen csempe van)

Jelmagyarázat.

r - regular tile, nem eltörhető csempe

b<life> - broken tile, eltörhető csempe

x<orangutans> - entry, bejárat (Orángutánok kiírása sorban vesszővel tagolva)

X - exit, kijárat

a<time> - armchair, fotel

v<time> - vending machine, csokiautomata

g<time> - game machine, játék automata

w - wardrobe, szekrény

**put**

**Leírás:** Letesz egy megadott fajta állatot, megadott névvel egy megadott mezőre amennyiben ez lehetséges. Ezután megjelenítjük a pályát.

**Opciók:**

put jpanda	<animal_name>	<id>
put spanda	<animal_name>	<id>
put lpanda	<animal_name>	<id>
put orangutan	<animal_name>	<id>

A pályában így megjelenő új jelek:

o - orangutan  
l - lazy panda  
j - jumping panda  
s - scared panda

**chain**

**Leírás:** Láncba fűz két állatot, amennyiben ez lehetséges. (=Beállítja a prev/nextAnimal kapcsolatokat és a prevTile mezőket.)

**Opciók:** chain <animal\_name1> <animal\_name2>  
chain <a1\_id> <a2\_id>

Az első állat mindig az orángutánhoz közelebbi állat (vagy maga az orángután), amely mögött a láncban már nincs semmi. Az állatokat megadhatjuk a nevükkel (kis pálya kreálása után a kezdeti láncok megalkotásánál hasznos) vagy koordinátájukkal (sok mozgás után már nem fogjuk tudni fejben tartani a nevüket az egyes állatoknak)

**move**

**Leírás:** Lépteti a megadott állatot a megadott csempére, amennyiben ez lehetséges.

Sikeres léptetés után kiírja a csempék azonosítóit és a rajtuk elhelyezkedő objektumokat. A megadott csempe mindenképpen szomszédos kell, hogy legyen a jelenlegi csempével.

**Opciók:** move <animal\_name> <next\_id>

Az állat megadható a nevével, illetve a csempe azonosítójával, amin éppen áll az előző parancsban ismertetett elvek szerint.

**unchain**

**Leírás:** Az orángután elengedi az aktuális láncát önként.

**Opciók:** unchain <animal\_name>

Az orángután, amit névnek adtunk elengedi a láncát, ha volt neki.

**start**

**Leírás:** Elindítja a játékot, körönként lépteti a pandákat, és megkéri a játékost, hogy lépjen az orángutánokkal. Az Entry-ből körönként kitesz egy Orangutant az emeletre.

**9.4.2 Kimeneti nyelv**

A parancsok egy része(cm, put, undo, run) a (cm + put) parancsok által meghatározott jelölésrendszer szerint kiírja a csempék azonosítóit és a rajtuk elhelyezkedő objektumok karaktereit.

Minden kör után a Controller megkéri a játékost, hogy lépjen az orángutánokkal. Végigmegy az összes Orangutan-on és a kimenetre az alábbi üzenetet írja:

make a move <orangutan\_name>

ezek után a játékosnak lépnie kell a kiírt nevű orángutánossal.

**A Pálya tartalmának kírása minden mozgás után:**

A játék során léphetnek az orángutánok, és minden lépésük után a csempék tartalmát kijelzi a program. A kimenet olyan, mint amit a bemeneti nyelvnél már le van írva.

## 9.5 A tesztek részletes tervei, leírásuk a teszt nyelven

### 9.5.1 Program starts and makes a map

- **Leírás**

Betölti a pályát

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Itt ellenőrizzük a program működését az emelet betöltésekor.

- **Bementet**

```
cm test1.flr
```

- **Elvárt kimenet**

```
cm test1.flr
0  1  2  3  4  5  6  7  8  9
1  _  X  _  _  _  _  _  _  _
2  X  _  X  _  _  _  _  X  _
3  _  X  _  X  _  _  _  X  _
4  _  _  X  X  _  _  _  X  _
5  _  _  _  X  _  X  _  X  _
6  _  _  _  _  X  X  X  X  _
7  _  _  _  _  _  X  X  _  _
8  _  X  X  X  X  _  X  _  _
9  _  _  _  _  _  X  _  _  _
1X
2r
3g
4w
5a
6r
7w
8b20
9x[o, o]
```

### 9.5.2 Entry generates two orangutans and player steps with them

- **Leírás**

Az emeleten elhelyezkedő Entry tárgy 2 Orangutant kell, hogy kitegyen. Miután kiteszi az elsőt, abban a körben a játékosnak lépnie kell az Orangutannal. A következő körben az Entry kitesz még egy Orangutant és ezzel is lépnie kell a játékosnak.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt ellenőrzi a játék azon funkcióját, hogy a játék kezdetekor az Entry kihelyezi a szükséges mennyiségű Orangutant az emeletre.

- **Bemenet**

```
cm test2.flr
put jpanda p 5
start
move o1 3
move o1 4
move o2 3
```

- **Elvárt kimenet**

```
cm test2.flr
0  1  2  3  4  5
1  _  X  _  _  _
2  X  _  X  _  _
3  _  X  _  X  _
4  _  _  X  _  _
5  _  _  _  _  _
1x2
2r
3r
4r
5r

put jpanda p 5
1x2
2r
3r
4r
5j

start
1x2
2r
3r
4r
5j

make a move o1
move o1 3
1x1
2r
3o
4r
5j

make a move o1
move o1 4
1x0
2o
3r
4o
5j

make a move o2
move o2 3
1x0
2r
3o
4o
5j

make a move o1
```

### 9.5.3

#### 9.5.4 Two orangutans wandering on the floor and suddenly one orangutan sees a panda and hits it.

- **Leírás**

Az emeleten 2 db Orangutan és 1 db Panda helyezkedik el. Az Orangutanokat úgy léptetjük, hogy az egyik elkapja a pandát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt ellenőrzi a játék azon funkcióját, hogy egy Orangutan befűzhet egy pandát. Ha a panda a 2. körben az ellenkező irányba lép, akkor a kimenet különbözhet, ilyenkor a tesztet addig kell futtatni, amíg a megfelelő kimenetet nem kapjuk.

- **Bemenet**

```
cm test3.flr
put orangutan o1 2
put orangutan o2 1
put lpanda pl 6
start
move o1 3
move o2 2
move o1 4
move o2 3
move o1 5
```

- **Elvárt kimenet**

cm test3.flr	3o
0 1 2 3 4 5 6	4r
1 _ X _ _ _	5r
2 X _ X _ _	6l
3 _ X _ X _	
4 _ _ X _ X _	make a move o2
5 _ _ _ X _ X	move o2 2
6 _ _ _ _ X _	1r
1r	2o
2r	3o
3r	4r
4r	5r
5r	6l
6r	
	make a move o1
put orangutan o1 2	move o1 4
1r	1r
2o	2o
3r	3r
4r	4o
5r	5l
6r	6r
put orangutan o2 1	make a move o2
1o	move o2 3
2o	1r
3r	2r
4r	3o
5r	4o
6r	5l
	6r
put lpanda pl 6	
1o	make a move o1
2o	move o1 5
3r	1r
4r	2r
5r	3o
6l	4l
	5o
start	6r
1o	
2o	make a move o2
3r	
4r	
5r	
6l	
make a move o1	
move o1 3	
1o	
2r	

### 9.5.5



### 9.5.6 Orangutan with Panda chain goes through a Wardrobe

- **Leírás**

Egy Orangutan utána befűzött Pandakkal megy a pályán, majd egy Wardrobe-hoz ér, amibe belép, és a pályán elhelyezett másik Wardrobe-ból lép ki, ez után addig lép, amíg a teljes panda lánc át nem ment a Wardrobe-on.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt ellenőrzi, hogy egy Orangutan utána befűzött Pandakkal láncsal képes-e áthaladni egy Wardrobe-on

- **Bemenet**

```
cm test4.flr
put orangutan o1 2
put lpanda l1 3
put lpanda l2 4
chain o1 l1
chain l1 l2
start
move o1 1
move o1 4
move o1 3
```

- **Elvárt kimenet**

cm test4.flr	3l
0 1 2 3 4 5 6	4l
1    X    -    -    -	5r
2   X    -   X    -   -	6w
3   -   X    -   X    -	
4   -   -    X    -   X	make a move o1
5   -   -    -   X    -	move o1 1
6   -   -   -   -   X   -	1w
1w	2l
2r	3l
3r	4r
4r	5o
5r	6w
6w	
	make a move o1
put orangutan o1 2	move o1 4
1w	1w
2o	2l
3r	3r
4r	4o
5r	5l
6w	6w
put lpanda l1 3	make a move o1
1w	move o1 3
2o	1w
3l	2r
4r	3o
5r	4l
6w	5l
	6w
put lpanda l2 4	make a move o1
1w	
2o	
3l	
4l	
5r	
6w	

```
chain o1 l1
chain l1 l2
start
1w
2o
```

### 9.5.7

## 9.5.8

### 9.5.9 Armchair notifies some randomly stepping pandas

- **Leírás**

A pályán el van helyezve egy Armchair, aminek az értesítési ideje 1 lépés, az orángután 3szor kell, hogy lépjen, így minden kör letelte után az Armchair értesíti a körülötte lévő pandákat. Az Armchair mellett található egy LazyPanda. Ezek közül a LazyPanda el fog láthatólag el fog tűnni a pályáról, mivel az Armchair-ben ül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrzi a játék azon funkcióját, hogy az Armchair tárgy interakcióba tud lépni egy LazyPanda-val.

- **Bemenet**

```
cm test5.flr
put orangutan o1 2
put lpanda l1 4
start
move o1 1
move o1 2
move o1 1
```

- **Elvárt kimenet**

cm test5.flr	move o1 2
0 1 2 3 4 5	1r
1 _ X _ _ _	2o
2 X _ X _ _	3a0
3 _ X _ X X	4r
4 _ _ X _ _	5r
5 _ _ X _ _	
1r	make a move o1
2r	move o1 1
3a	1o
4r	2r
5r	3a0
	4r
put orangutan o1 2	5r
1r	
2o	make a move o1
3a	
4r	
5r	

## 9.5.10

```
put lpanda l1 4
1r
2o
3a
4l
5r
```

```
start
1r
2o
3a
4l
5r
```

```
make a move o1
move o1 1
1o
2r
3a
4l
5r
```

```
make a move o1
```

### 9.5.11 Lonely Orangutan walking around a broken tile and breaks it, then the game ends

- **Leírás**

*A pályán egyetlen orángutánnal tud a játékos két csempe között lépkedni. Amint a csempe eltörik, az orángután meghal és így a játék véget ér.*

- **Ellenőrzött funkcionalitás, várható hibahelyek**

*A teszt ellenőrzi, hogy a játékos képes léptetni az orángutánt, a törött csempe élettartama tényleg csökken, ha rálépnek, illetve, ha a törött csempe eltörik, akkor az orángután tényleg meghal-e.*

- **Bemenet**

```
cm test6.flr
put orangutan o1 2
put lpanda l1 3
start
move o1 1
move o1 2
move o1 1
move o1 2
move o1 1
move o1 2
move o1 1
move o1 2
```

- **Elvárt kimenet**

cm test6.flr	make a move o1
0 1 2 3	move o1 2
1 _ X _	1r
2 X _ _	2o
3 _ _ _	3l
1r	
2b4	make a move o1
3r	move o1 1
	1o
put orangutan o1 2	2b2
1r	3l
2o	
3r	make a move o1
	move o1 2
put lpanda l1 3	1r
1r	2o
2o	3l
3l	
start	make a move o1
1r	move o1 1
2o	1o
3l	2b1
	3l
make a move o1	make a move o1
move o1 1	move o1 2
1o	1r
2b4	2o
3l	3l
make a move o1	
move o1 2	
1r	
2o	
3l	
make a move o1	
move o1 1	
1o	
2b3	
3l	

### 9.5.12

### 9.5.13 GameMachine scares a panda

- **Leírás**

A pályán egy orángutánnal és a hozzá tartozó pandalánccal tud a játékos lépkedni. Található még a pályán egy játékgép, ami 2 körönként csilingel, ezzel megijeszti a pandalácmban lévő, erre hajlamos pandát, ami ennek hatására elengedi a szomszédjait, jelen esetben az orángutánt, illetve a mögötte álló pandát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Megijesztett panda valóban elengedi-e a szomszédjai kezét.

- **Bemenet**

```
cm test7.flr
put orangutan o1 3
put spanda sp 2
put lpanda lp 1
chain o1 sp
chain sp lp
start
move o1 4
move o1 5
move o1 6
```

- **Elvárt kimenet**

```
cm test7.flr
0 1 2 3 4 5 6
7
1 _ X _ _ _ _
- 2 X _ X _ _ _
- 3 _ X _ X _ _
- 4 _ _ X _ X _
X 5 _ _ _ X _ X
- 6 _ _ _ _ X _
- 7 _ _ _ X _ _
1r
2r
3r
4r
5r
6r
7g

put orangutan o1 3
1r
2r
3o
4r
5r
6r
7g

put spanda sp 2
1r
2s
3o
4r
5r
6r
7g

put lpanda lp 1
```

```
1l
2s
3o
4r
5r
6r
7g

make a move o1
```

```
chain o1 sp
chain sp lp
start
1l
2s
3o
4r
5r
6r
7g

make a move o1
move o1 4
1r
2l
3s
4o
5r
6r
7g

make a move o1
move o1 5
1r
2r
3l
4s
5o
6r
7g

make a move o1
move o1 6
1r
2r
3l
```

### 9.5.14

### 9.5.15 Orangutan steals Pandas from an Orangutan without Panda-chain

- **Leírás**

A pályán két orangutánt irányíthat a játékos, melyek közül csak az egyiknek van pandalánca(egy panda). A teszt alatt az egyedüli orangután nekimegy a másik orangután láncának, és így ellopja azt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az orangután láncolt pandának ütközve tényleg el tudja-e lopni azt.

- **Bemenet**

```
cm test8.flr
put orangutan o1 3
put orangutan o2 2
put lpanda lp 1
chain o1 lp
start
move o1 5
move o2 4
move o1 7
move o2 5
move o1 8
move o2 6
```

- **Elvárt kimenet**

```
cm test8.flr
0 1 2 3 4 5 6 7 8
1 _ X _ _ _ _ _ _
2 X _ _ _ X _ _ _
3 _ _ _ X _ _ _ _
4 _ _ X _ X _ _ _
5 _ X _ X _ X X _
6 _ _ _ _ X _ _ _
7 _ _ _ _ X _ _ X
8 _ _ _ _ _ _ X _
1r
2r
3r
4r
5r
6r
7r
8r

put orangutan o1 3
1r
2r
3o
4r
5r
6r
7r
8r

put orangutan o2 2
1r
2o
3o
4r
5r
6r
7r
8r

put lpanda lp 1
1l
2o
3o
4r
5r
```

```
6r
7r
8r

chain o1 lp
start
1l
2o
3o
4r
5r
6r
7r
8r
```

```
make a move o1
move o1 5
1r
2o
3l
4r
5o
6r
7r
8r

make a move o2
move o2 4
1r
2r
3l
4o
5o
6r
7r
8r
```

```
make a move o1
move o1 7
1r
2r
3r
4o
5l
6r
```

```
7o
8r

make a move o2
move o2 5
1r
2r
3r
4l
5o
6r
7o
8r

make a move o1
move o1 8
1r
2r
3r
4l
5o
6r
7r
8o

make a move o2
move o2 6
1r
2r
3r
4r
5l
6o
7r
8o

make a move o1
```

### 9.5.16

### 9.5.17 Orangutan hits a chained Panda from another chain

- **Leírás**

A pályán két orangután található, mindkettőnek van egy egyelemű pandalánca. A teszt során az egyik orangután nekimegy a másik orangután láncának, így elrabolva azt a pandát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Lépés láncsal, befűzött panda saját láncba fűzése

- **Bemenet**

```
cm test9.flr
put orangutan o1 2
put orangutan o2 4
put lpanda lp1 1
put lpanda lp2 3
chain o1 lp1
chain o2 lp2
start
move o1 6
move o2 5
move o1 8
move o2 6
move o1 9
move o2 7
```

- **Elvárt kimenet**

cm test9.flr	9r	3l
0 1 2 3 4 5 6 7 8 9		4o
1    X    -    -    -    -    -    -    -	put lpanda lp1 1	5r
2   X    -    -    -    -    X    -    -    -	1l	6o
3   -    -    -    X    -    -    -    -    -	2o	7r
4   -    -    X    -    X    -    -    -    -	3r	8r
5   -    -    -    X    -    X    -    -    -	4o	9r
6   -    X    -    -    X    -    X    X    -	5r	
7   -    -    -    -    -    X    -    -    -	6r	make a move o2
8   -    -    -    -    -    X    -    -    X	7r	move o2 5
9   -    -    -    -    -    -    -    X    -	8r	1r
1r	9r	2l
2r		3r
3r	put lpanda lp2 3	4l
4r	1l	5o
5r	2o	6o
6r	3l	7r
7r	4o	8r
8r	5r	9r
9r	6r	
	7r	make a move o1
put orangutan o1 2	8r	move o1 8
1r	9r	1r
2o		2r
3r	chain o1 lp1	3r
4r	chain o2 lp2	4l
5r	start	5o
6r	1l	6l
7r	2o	7r
8r	3l	8o
9r	4o	9r
	5r	
	6r	make a move o2
put orangutan o2 4	7r	move o2 6
1r	8r	1r
2o	9r	2r
3r		3r
4o		4l
5r	make a move o1	5l
6r	move o1 6	6o
7r	1r	7r
8r	2l	

```

8o
9r

make a move o1
move o1 9
1r
2r
3r
4l
5l
6o
7r
8r
9o

make a move o2
move o2 7
1r
2r
3r
4r
5l
6l
7o
8r
9o

make a move o1

```

### 9.5.18

### 9.5.19 *VendingMachine makes JumpingPanda jump*

- **Leírás**

A pályán el van helyezve egy VendingMachine, ami a következő körben jelezni fog a környező JumpingPandáknak, hogy ugorjanak. A VendingMachine szomszédságába másik pandát teszünk, egy szintén BrokenTile-ra. A ScaryPanda nem töri össze jelzésre a csempét.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrzi, hogy a VendingMachine jelzésére ténylegesen csak a JumpingPanda reagált-e, így csak az alatta lévő BrokenTile élete csökken eggyel. (ezt a kimeneten ténylegesen nem látjuk)

- **Bemenet**

```

cm test10.flr
put jpanda j1 1
put orangutan o1 3
put spanda s1 4
start
move o1 3
move o1 3

```

- **Elvárt kimenet**

cm test10.flr	2v1	3o
0 1 2 3 4	3b20	4s
1 _ X _ _	4b20	
2 X _ X X		start
3 _ X _ _	put orangutan o1 3	1j
4 _ X _ _	1j	2v1
1b1	2v1	3o
2v1	3o	4s
3b20	4b20	
4b20		make a move o1
	put spanda s1 4	move o1 3
put jpanda j1 1	1j	1j
1j	2v1	2v1

3o  
4s

make a move o1  
move o1 3  
1j  
2v0  
3o  
4s

make a move o1

**9.5.20**



## 9.5.21

### 9.5.22 *Orangutan hits its own Panda chain*

- **Leírás**

A pálya 5 törhetetlen csempéből áll, amelyet feltöltünk 1 orángutánnal és 3 vele láncba kapcsolt LazyPandával (vagy bármilyen más pandával) 1 csempét üresen hagyva. Az orángutánt ezután ráléptetjük a láncának utolsó pandájára és lépünk még egyet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt kell látnunk, hogy a plusz egy lépésnél már nem a régi láncát húzza az orángután, hanem az újat.

- **Bemenet**

```
cm test11.flr
put lpanda 11 2
put lpanda 12 3
put orangutan o1 4
put lpanda 13 5
chain o1 13
chain 13 12
chain 12 11
start
move o1 2
move o1 1
```

- **Elvárt kimenet**

cm test11.flr	4o
0 1 2 3 4 5	5l
1     X     -	
2 X     - X X -	chain o1 13
3 - X - - X	chain 13 12
4 - X - - X	chain 12 11
5 - - X X -	start
1r	1r
2r	2l
3r	3l
4r	4o
5r	5l
put lpanda 11 2	make a move o1
1r	move o1 2
2l	1r
3r	2o
4r	3l
5r	4l
	5l
put lpanda 12 3	make a move o1
1r	move o1 1
2l	1o
3l	2l
4r	3r
5r	4l
	5l
put orangutan o1 4	make a move o1
1r	
2l	
3l	
4o	
5r	

```
put lpanda 13 5
1r
2l
3l
```

## 9.5.23

## 9.5.24

### 9.5.25 *Orangutan exits with two Pandas*

- **Leírás**

A pálya álljon 3 üres törhetetlen csempéből és egy kijáratból. A három üres csempére tegyünk le az orángutánt a láncba kapcsolt két lazy pandájával (vagy bármilyen más típusú pandájával). Az orángutánt ezután ráléptetjük a kijárat csempéjére.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az orángután a lépésével közelebb húzza a pandáit a kijáratához, eltűnik és felbukkan a bejáraton. Ezután a kijárat minden körben magához hívja a lánc következő pandáját és mivel semmilyen pandákat zavaró tényező nincs, így két plusz kör végeztével azt kell látnunk, hogy csak az orángutánunk van a pályán.

- **Bemenet**

```
cm test12.flr
put orangutan o1 2
put lpanda 11 3
put lpanda 12 4
chain o1 11
chain 11 12
start
move o1 1
move o1 4
move o1 3
```

- **Elvárt kimenet**

cm test12.flr	start
0 1 2 3 4 5	1X
1 _ X _ _	2o
2 X _ X _	3l
3 _ X _ X _	4l
4 _ _ X _ X	5x[]
5 _ _ _ X _	
1X	make a move o1
2r	move o1 1
3r	1X
4r	2r
5x[]	3l
	4l
put orangutan o1 2	5x[o]
1X	
2o	make a move o1
3r	move o1 4
4r	1X
5x[]	2r
	3r
put lpanda 11 3	4o
1X	5l
2o	
3l	
4r	
5x[]	
	move o1 3
put lpanda 12 4	
1X	
2o	
3l	
4l	
5x[]	
chain o1 11	
chain 11 12	

## 9.5.26

### 9.5.27 *Orangutan is exiting the floor with a Panda chain meanwhile a GameMachine scares one ScaredPanda in it who leaves the chain with all the Pandas behind.*

- **Leírás**

A pályát 8 mezőből alkotjuk, egy bejáratból, egy kijáratból, egy játékgépből és 5 törhetetlen csempéből. A játékgép a következő kör végén jelez. A pályára lerakunk egy orángutánt a kijárat szomszédságába és 4 vele láncolt pandát (2 Scared, 1 Jumping, 1 Lazy) úgy, hogy az orángután a kijáratra lépés következtében odahúzza az egyik ScaredPandát a játékgép (GameMachine) közelségébe.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a GameMachine szomszédságába húzott panda és az amögött lévő pandák leszakadnak-e a láncból a jelzés hatására, azaz a következő körökben nem indulnak meg maguktól a kijárat felé, minthogy az alapesetben történne.

- **Bemenet**

```
cm test13.flr
put spanda s1 2
put lpanda l1 3
put spanda s2 4
put orangutan o1 5
chain o1 s2
chain s2 l1
chain l1 s1
start
move o1 6
move o1 7
move o1 2
```

- **Elvárt kimenet**

cm test13.flr	6r	8g
0 1 2 3 4 5 6 7 8	7w	
1 _ X _ _ _ _ _	8g	make a move o1
2 X _ X _ _ _ _		move o1 6
3 _ X _ X _ _ _	put spanda s2 4	1w
4 _ _ X _ X _ _ X	1w	2r
5 _ _ _ X _ X _ _	2s	3s
6 _ _ _ _ X _ X _	3l	4l
7 _ _ _ _ _ X _ _	4s	5s
8 _ _ _ X _ _ _ _	5r	6o
1w	6r	7w
2r	7w	8g
3r	8g	
4r		make a move o1
5r	put orangutan o1 5	move o1 7
6r	1w	1w
7w	2s	2o
8g	3l	3r
	4s	4s
put spanda s1 2	5o	5l
1w	6r	6s
2s	7w	7w
3r	8g	8g
4r		
5r	chain o1 s2	make a move o1
6r	chain s2 l1	move o1 2
7w	chain l1 s1	1w
8g	start	2o
	1w	3r
put lpanda l1 3	2s	4s
1w	3l	5l
2s	4s	6s
3l	5o	7w
4r	6r	8g
5r	7w	

```
make a move o1
```

## 9.5.28

### 9.5.29 *Panda breaks a tile with its move*

- **Ellenőrzött funkcionalitás, várható hibahelyek**

*A mozgás valóban megvalósul-e, a csempe összetörik-e?*

- **Bemenet**

```
cm test14.flr
put orangutan o 3
put lpanda l 1
start
move o 3
move o 3
```

- **Elvárt kimenet**

```
cm test14.flr
0 1 2 3
1 _ X _
2 X _ _
3 _ _ _
```

```
1r
2b1
3r
```

```
put orangutan o 3
1r
2b1
3o
```

```
put lpanda l 1
1l
2b1
3o
```

```
start
1l
2b1
3o
```

```
make a move o
move o 3
1l
2b1
3o
```

```
move o 3
```

## 9.5.30

### 9.5.31 *Orangutan releases its PandaChain*

- **Leírás**

Orangután elengedi a láncát saját akaratából.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

*Az orangután elengedi a mögötte álló pandaláncot.*

- **Bemenet**

```
cm test15.flr
put orangutan o1 2
put jpanda p1 1
chain o1 p1
start
```

```
move o1 3
unchain o1
move o1 4
```

● **Elvárt kimenet**

```
cm test15.flr
  0  1  2  3  4
  1  -  X  -  -
  2  X  -  X  -
  3  -  X  -  X
  4  -  -  X  -
1r
2r
3r
4r

put orangutan o1 2
1r
2o
3r
4r

put jpanda p1 1
1j
2o
3r
4r

chain o1 p1
start
1j
2o
3r
4r

make a move o1
move o1 3
1r
2j
3o
4r

make a move o1
unchain o1
1r
2j
3o
4r

make a move o1
move o1 4
1j
2r
3r
4o

make a move o1
```

## Grafikus felület specifikációja

### A grafikus interfész

Közös Ikonok stb.

Pálya

EndGame

A játék alatt használt képek 64x64-es felbontású átlátszó png fileban találhatóak.

Alap csempék:



RegularTile



BrokenTile

A csempéken találhatóak AThings leszármazottak:



Armchair



Entry



Exit



GameMachine



Orangutan



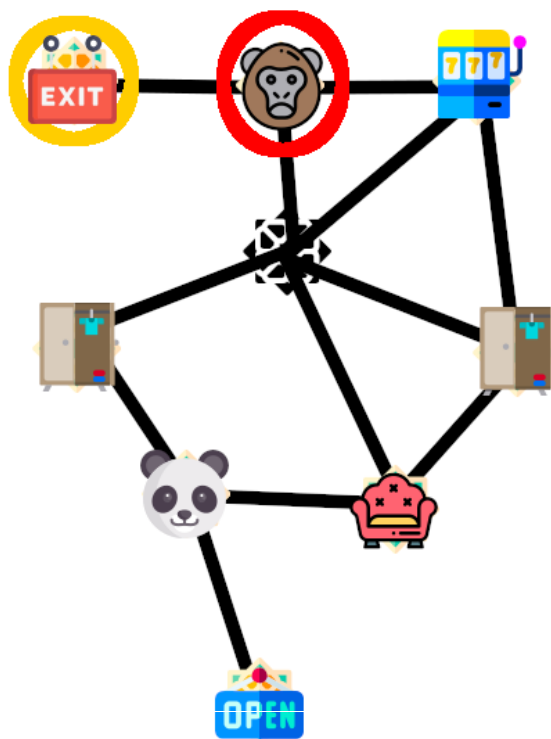
Panda



VendingMachine



Wardrobe



Egy kész pálya a játék összes elemét tartalmazva az ábrán.

A pályán körben lévő orángután az, amivel mozgunk és sárga kör, az a mező, ahova a kiválasztást jóváhagyva az orángután lépni fog. Ezt billentyűn jobbra és balra navigálással majd kiválasztással tehetjük. Játékot úgy indítunk, hogy az ablak felső bal sarkában legördülő menüből kiválasztjuk a futás gombot. A játék végéről egy felugró ablak tájékoztat az elért pontszámunkkal.

### A grafikus rendszer architektúrája

A projekt az MVC tervezési mintát valósítja meg.

A Modell a programban található Floor osztály egy példányát adja. Itt található a pálya: csempék, rajta található tárgyak, állatok. Csak adatok, és a karakterek közötti interakció megvalósított itt.

A Controller az azonos névvel ellátott osztályban van implementálva, amiben megvalósul az állatok mozgatása, illetve a tárgyak értesítése (effect-> csilingelés, sípolás...) a szomszéd csempékre. Ennek az osztálynak feladata még, hogy a felhasználói interakciókat (Orángután léptetése, játék indítás leállítása)

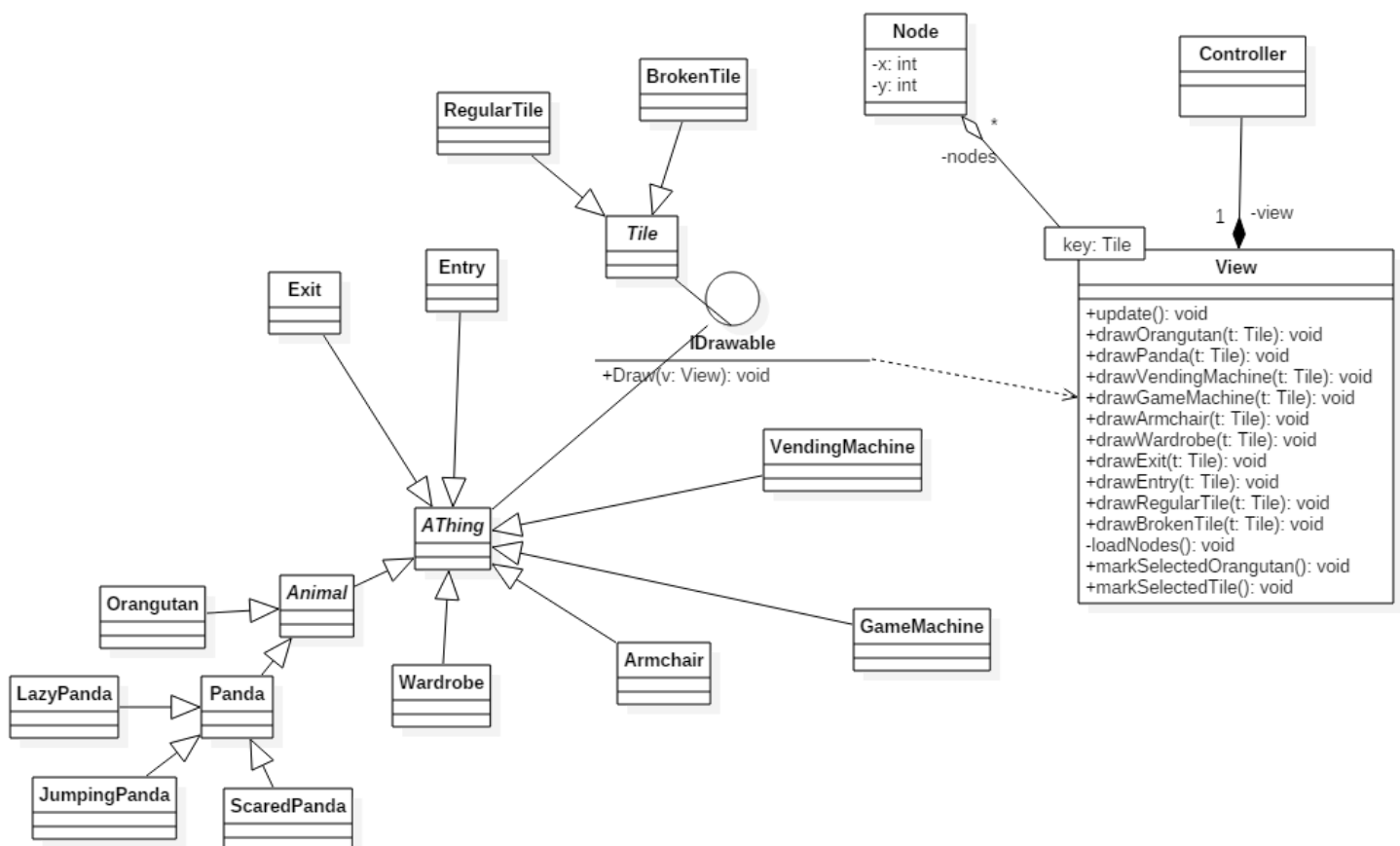
kezelése. Az program futása során minden modellben való változtatás után jelzi a View számára, hogy frissítse magát.

A View azonos nevű osztályban megvalósul a tényleges képernyőre rajzolás. Ha a Controller jelzi, hogy változás történt, akkor a View a modellt használva minden pályaelemet kirajzol az erre implementált IDrawable interface-el. Az interface függvényei visszahívják a megfelelő View függvényét és kirajzolják magukat. Ez egy Armchairban ülő állat esetén sorban így kerül egymásra képpel: RegularTile – Armchair – LazyPanda.

## A felület működési elve

A grafikus rendszerünk az MVC alapjaira fog épülni (Model-View-Controll), illetve a hibrid megoldást választottuk. A Controller osztály fogja jelezni minden lépés után a View-nak, hogy a lépés megtörtént. A View osztály csak a megjelenítésért felel, ez a két osztály az IDrawable interface-el keresztül kommunikál egymással. A View osztály update függvénye tehát a pályán lévő összes állatnak, tárgynak meghívja a Draw függvényét, amelyek újrarajzoltatják magukat.

## A felület osztály-struktúrája



## A grafikus objektumok felsorolása

### View

- **Felelősség**

Ebben az osztályban van összegyűjtve minden olyan függvény, ami a kirajzolással kapcsolatos.

- **Ósosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **-nodes: HashMap<Tile, Nodes>**: A képernyőn megjelenő csempéket reprezentáló attribútum.

- **Metódusok**

- **+update():void**: A Controller hívja meg. A meghívás számít utasításnak, hogy rajzolja ki az frissített képet a pályáról. Végig megy a pályán, és minden elemet kirajzol.
- **+drawArmchair(t: Tile): void**: Az Armchair képét rajzolja a kapott Node-ra tile alapján
- **+drawBrokenTile(t: Tile): void**: A BrokenTile képét rajzolja a kapott Node-ra tile alapján
- **+drawEntry(t: Tile): void**: Entry képét rajzolja a kapott Node-ra tile alapján
- **+drawExit(t: Tile): void**: Az Exit képét rajzolja a kapott Node-ra tile alapján
- **+drawGameMachine(t: Tile): void**: A GM képét rajzolja a kapott Node-ra tile alapján
- **+drawOrangutan(t: Tile): void**: Orángután képét rajzolja a kapott Node-ra tile alapján
- **+drawPanda(t: Tile): void**: Panda képét rajzolja a kapott Node-ra tile alapján
- **+drawRegularTile(t: Tile): void**: RegularTile képét rajzolja a kapott Node-ra tile alapján
- **+drawVendingMachine(t: Tile): void**: VM képét rajzolja a kapott Node-ra tile alapján
- **+drawWardrobe(t: Tile): void**: Wardrobe képét rajzolja a kapott Node-ra tile alapján
- **-loadNodes(): void**: Betölti a pálya elemeinek pozícióit és feltölti így a Node HashMap-jét.

### Node

- **Felelősség**

A pályán megjelenő csomópont, pozícióval rendelkezik.

- **Ósosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **-x: int**
- **-y: int**

- **Metódusok**

- **+Node(\_x: int, \_y: int)**: Értékadó konstruktor.

### IDrawable



- **Felelősség**

Interface, amit minden rajzolható objektumban implementált.

- **Ősosztályok**

-

- **Interfészek**

-

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.

## **Animal**

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.(Virtuális tagfüggvény, alosztályokban megvalósított)

## **Armchair**

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát. Meghívja a rajta található állat Draw függvényét is, ha benne ül.

## **AThing**

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát. (Implementáció miatt van benne)

## **BrokenTile**

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát. Meghívja a rajta található objektum Draw-ját, ha van valami rajta.

## **Entry**

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.

## Exit

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.

## GameMachine

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.

## Orangutan

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.

## Panda

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.

## RegularTile

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.  
Meghívja a rajta található objektum Draw-ját, ha van valami rajta.

## Tile

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.  
(Implementálás miatt van benne)

## **VendingMachine**

- **Interfészek**

IDrawable

- **Metódusok**

- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.

## **Wardrobe**

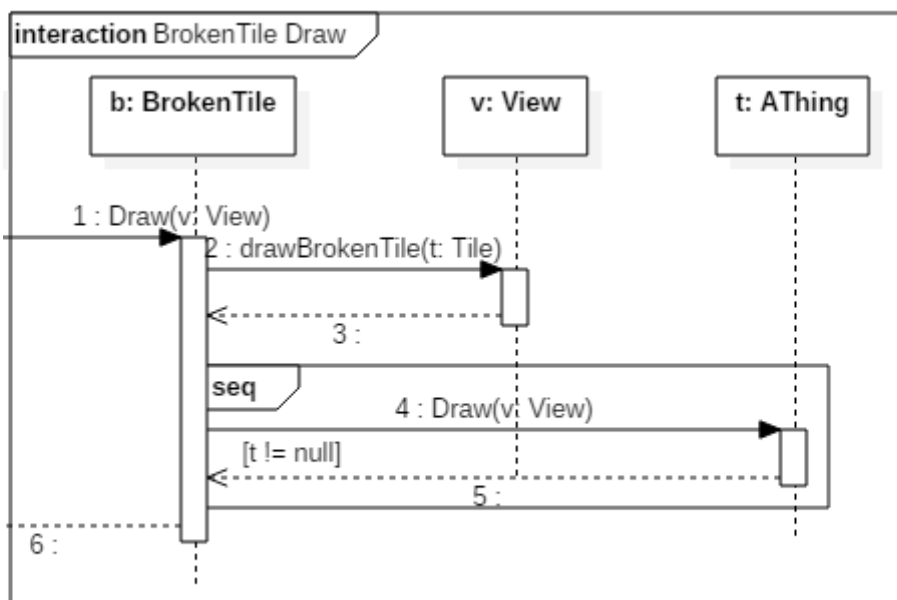
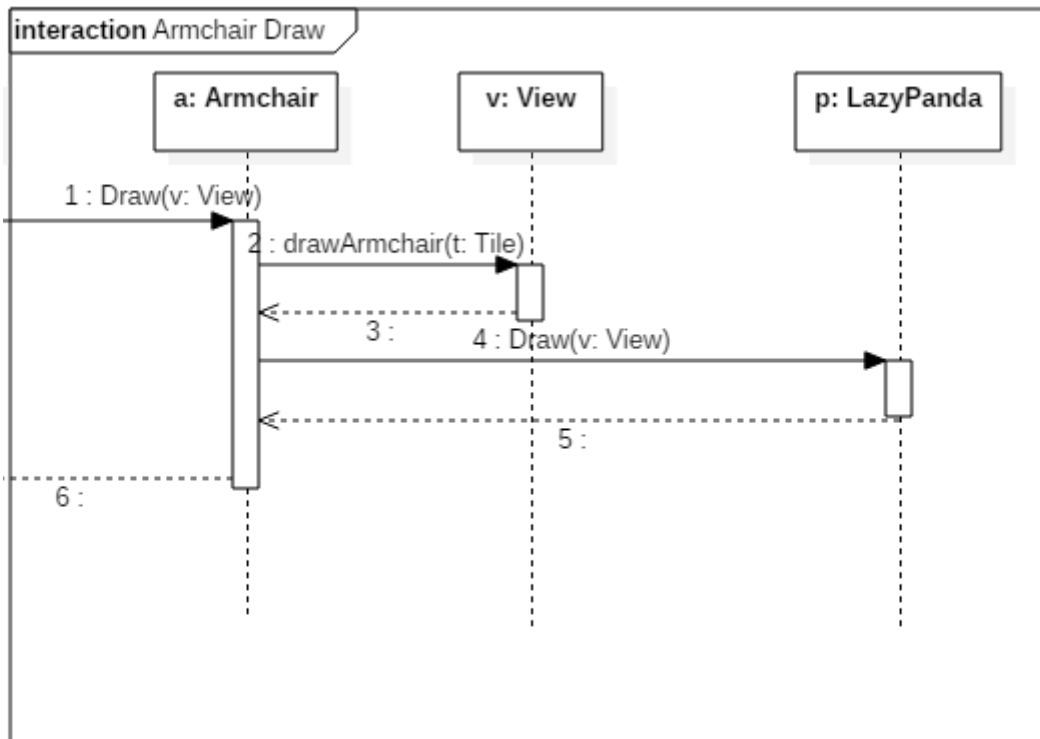
- **Interfészek**

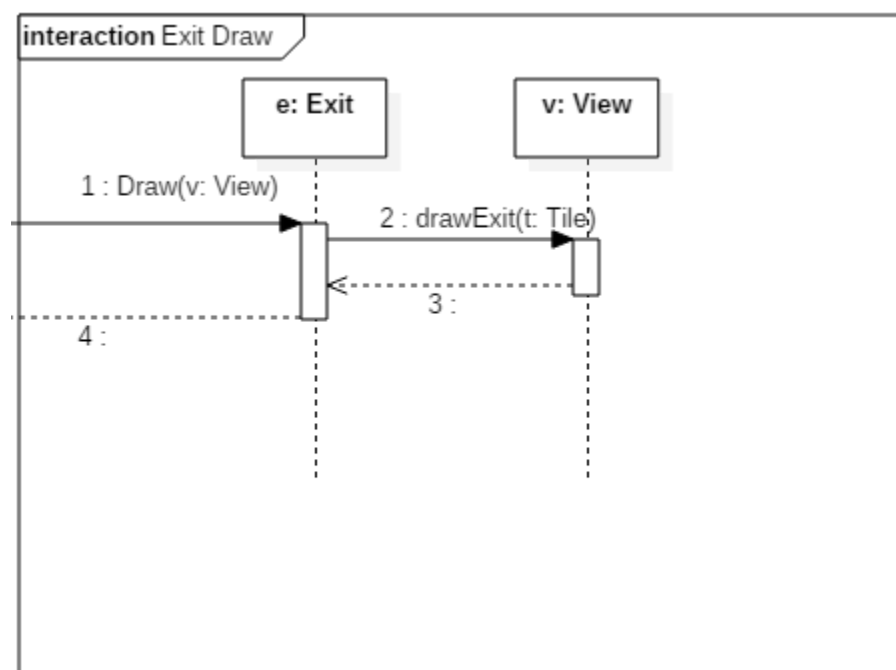
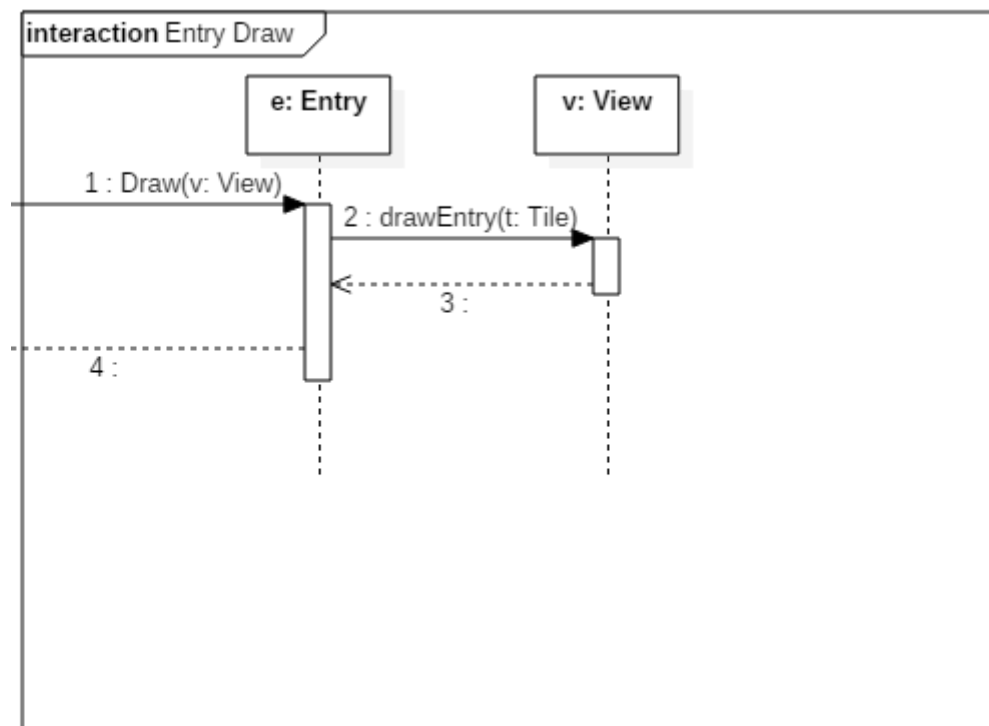
IDrawable

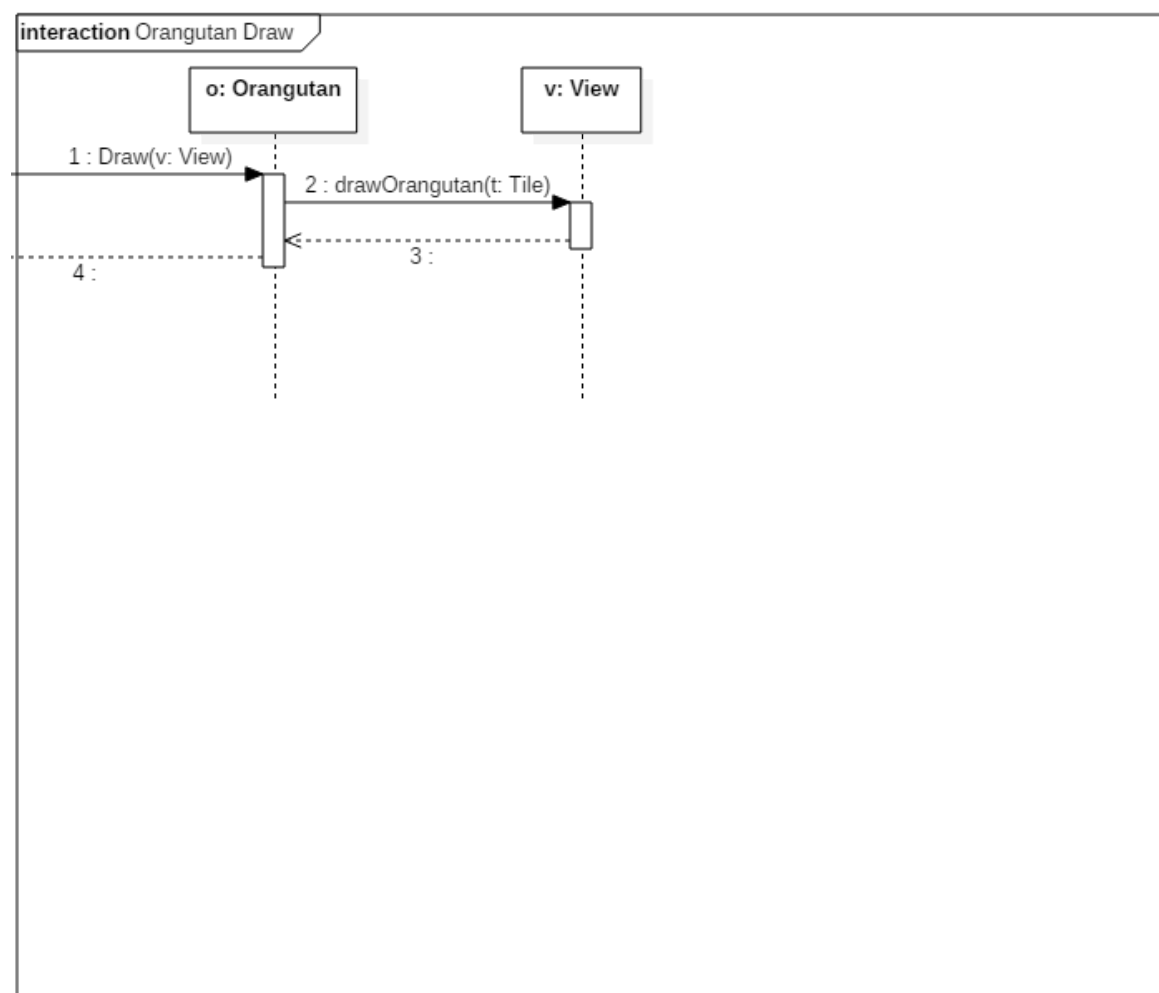
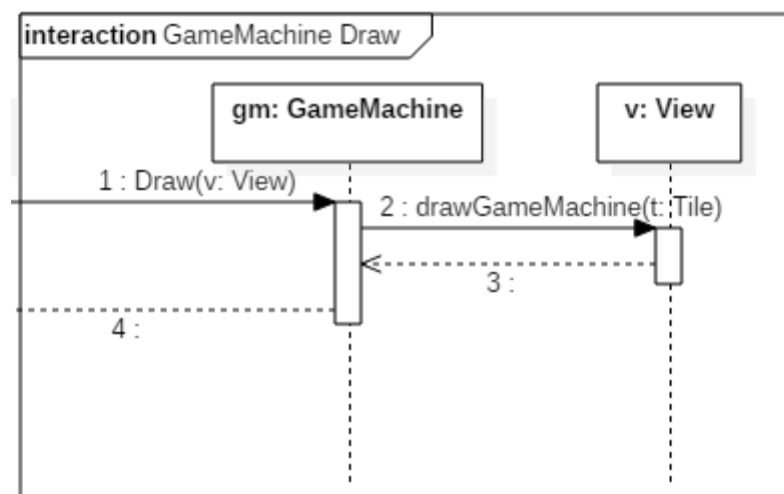
- **Metódusok**

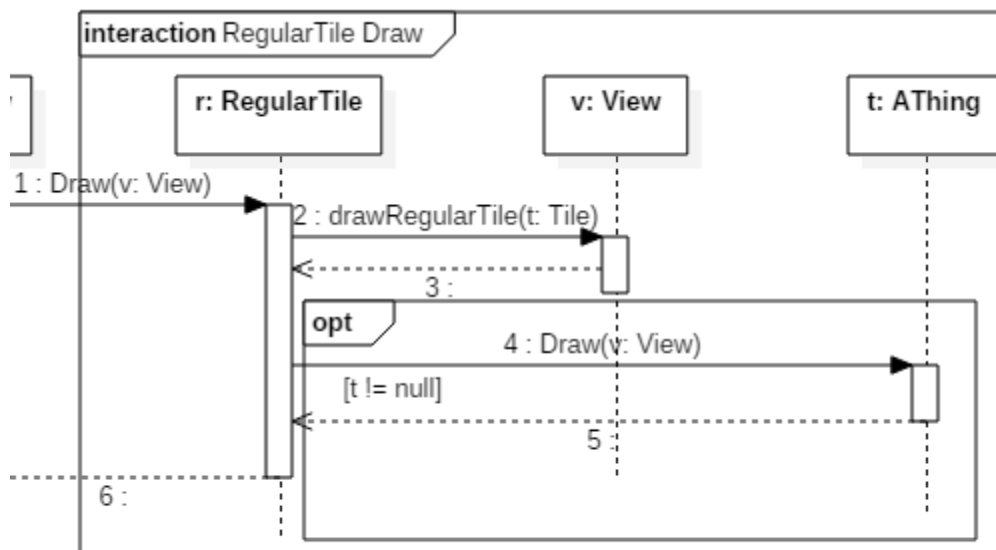
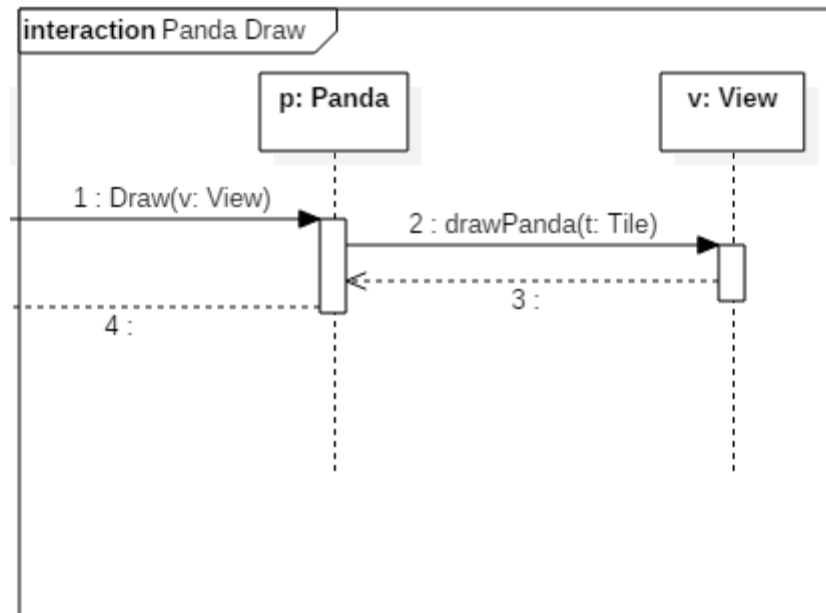
- **+Draw(v: View): void:** Az implementált objektum kirajzolja a paraméterként kapott pontra a magát.

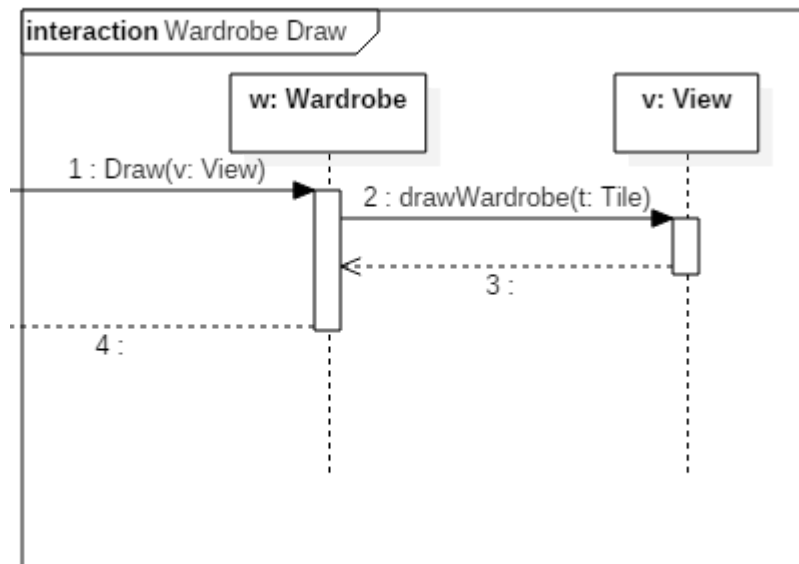
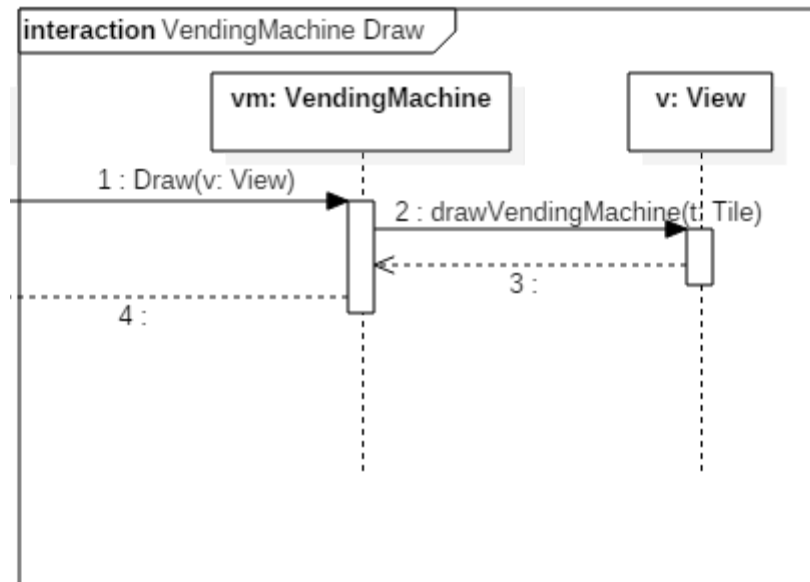
## ***Kapcsolat az alkalmazói rendszerrel***













**Napló**

Kezdet	Időtartam	Résztevők	Leírás
2019.04.27. 18:00	0.5 óra	Kecskés, Kobál, Martonosi	<b>Értekezlet:</b> A feladat közös értelmezése, gui kidolgozása közösen
2019.04.27 19:00	0.5 óra	Kecskés	Interface és MVP modell kidolgozása lapon, hogy következő értekezleten felvethető legyen egy lehetőségként
2019.04.28 10:00	1 óra	Kecskés, Kiss, Kis, Kobál, Martonosi	<b>Értekezlet:</b> A feladatok ellenőrzése és felvetődött kérdések (Interface szerepe és View-Model kapcsolat kérdése) megoldása
2019.04.28 11:00	0.5 óra	Kiss	A kiszabott rész (felület működési elve)
2019.04.28 11:00	1.0 óra	Kobál	A grafikus interface kidolgozása (képek és pályadesign)
2019.04.28 15:00	0.5 óra	Kis	Osztályleírások kidolgozása
2019.04.28 16:00	1.5 óra	Martonosi	Osztálydiagram és Szekvenciák elkészítése
2019.04.28 18:00	1.5 óra	Kecskés	A konkrét modell kidolgozása, osztályok és szekvenciák megírása
2019.04.28 23:00	1.0 óra	Kecskés	Dokumentum formázása, naplózás mindenkinek és saját rész javítása az értekezletnek megfelelően

## Grafikus változat beadása

### Fordítási és futtatási útmutató

#### Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
mainFloor.flr	3227	2019.05.13 02:10	A pálya szerializálva
mainFloor.vw	381	2019.05.13 02:10	A pálya csomópontjainak elhelyezkedése
src/Animal.java	3064	2019.05.13 02:10	Az állatokat gyűjtő ősz osztály
src/Armchair.java	2005	2019.05.13 02:10	Karosszék függvényeit tároló
src/AThing.java	2309	2019.05.13 02:10	Minden tárgy ősz osztály
src/BrokenTile.java	1691	2019.05.13 02:10	Törött csempe osztály
src/Controller.java	5921	2019.05.13 02:10	A pandát irányítja (és később a pályát állítja össze)
src/Entry.java	2283	2019.05.13 02:10	Bejárat osztály
src/Exit.java	3292	2019.05.13 02:10	Kijárat osztály
src/Floor.java	8479	2019.05.13 02:10	A pálya osztály, benne tárolva a csempékkel, pandákkal, orángutánokkal
src/FloorMakingProgram.java	2487	2019.05.13 02:10	A pályákat beállító / (.flr fileokat) generáló program
src/GameMachine.java	1795	2019.05.13 02:10	Játékgép osztály
src/IDrawable.java	328	2019.05.13 02:10	A kirajzoláshoz szükséges interface
src/IPandaEffective.java	297	2019.05.13 02:10	Az hatással rendelkező (szomszéd csempét értesítő effekteket tartalmazó) osztályok interface-e
src/JumpingPanda.java	901	2019.05.13 02:10	Ugráló panda osztály
src/LazyPanda.java	2062	2019.05.13 02:10	Lusta panda osztály
src/Node.java	823	2019.05.13 02:10	A képernyőn elhelyezkedő pontok osztály
src/Orangutan.java	2572	2019.05.13 02:10	Orángután osztály
src/Panda.java	2885	2019.05.13 02:10	Panda osztály
src/Program.java	412	2019.05.13 02:10	A programot elindító, main-t tartalmazó osztály
src/RegularTile.java	564	2019.05.13 02:10	Nem törős csempe osztály
src/ScaredPanda.java	786	2019.05.13 02:10	Ijedős panda osztály
src/Tile.java	6482	2019.05.13 02:10	Csempe ősz osztály
src/VendingMachine.java	2044	2019.05.13 02:10	Játékautomata osztály
src/View.java	20611	2019.05.13 02:10	A kirajzoláshoz tartozó osztály
src/Wardrobe.java	2572	2019.05.13 02:10	A szekrényt reprezentáló osztály
src/images/Armchair.png	2750	2019.05.13 02:10	Armchair képe
src/images/BrokenTile.png	1652	2019.05.13 02:10	BrokenTile képe
src/images/Entry.png	2103	2019.05.13 02:10	Entry képe
src/images/Exit.png	2184	2019.05.13 02:10	Exit képe
src/images/GameMachine.png	1877	2019.05.13 02:10	GameMachine képe
src/images/Orangutan.png	19006	2019.05.13 02:10	Orangutan képe
src/images/Panda.png	18473	2019.05.13 02:10	Panda képe
src/images/RegularTile.png	3041	2019.05.13 02:10	Sima csempe képe
src/images/VendingMachine.png	2692	2019.05.13 02:10	VendingMachine képe

g			
src/images/Wardrobe.png	16906	2019.05.13 02:10	Wardrobe képe

## Fordítás és telepítés

Az src mappában cmd-t indítva és a java futtatókörnyezetet implementálva:

```
javac Program.java
```

```
java Program.
```

(Ha nincs futtatókörnyezet, akkor lehet a teljes elérési úttal dolgozni a Program.java és Program paraméterek megadásakor. A parancssor ilyenkor a jdk könyvtárából kell futtatni)

(Jók ezek a labor gépek)

## Futtatás

A java Program parancs után egy cyan színű ablak jelenik meg.

A játék úgy indítható, ha File→Open gombbal kitallózzuk a mainFloor.flr-t ami a src könyvtár mellett található (a projekt fordított mappájából egyet visszalépve a gyöker könyvtár felé). A program automatikusan a mellette található .vw file-t betölti, nekünk nem kell azzal is törődni.

Amint ez megvan, látjuk a képernyőn a pálya elemeit állatok nélkül. Ha a File→Start gombra nyomunk, akkor elindul a játék, és lehet mozogni a kékkel bekarikázott állattal. A kiválasztott csempe, ahova lépnénk pink színű, váltani csempék között A és D gombbal lehet. Ha kiválasztottunk a csempét, akkor a SPACE gombbal lehet jóváhagyni és megtörténik a mozgás. A kiválasztás ideje alatt akár el is engedhetjük a kezét a mögöttünk álló pandáknak, ha R gombot nyomunk.

## Értékelés

Tag neve	Tag neptun	Munka százalékban
Kecskés Péter	XZ9I7F	25
Kis Máté Levente	H0OTLX	15
Kiss Előd	U479JN	14
Kobál Sándor	C4W3BP	25
Martonosi Máté	QOT4DC	21

## Napló

Kezdet	Időtartam	Résztvevők	Leírás
2019.05.04 11:00	1.0 óra	Kecskés, Kiss, Kis, Kobál, Martonosi	<b>Értekezlet:</b> Átértelmezésre került a View osztály, Kobál kijavítja az Entry - Exit helytelen működését, Kecskés megoldja a node-ok tárolását, Kiss adaptálja a gráfkirajzolót, Kis implementálja a View osztályt, Martonosi megoldja a mezők kiválasztásának megjelenítését. Pénteken egy újabb gyűlésre kerül sor.
2019.05.10 14:00	1.0 óra	Kecskés, Kiss, Kobál, Martonosi	<b>Értekezlet:</b> A hét közben felmerült problémák átbeszélése történt. Külön branchben javítjuk a modell-controller hibákat és a model-controller-view problémákat.
2019.05.11 11:00	1.0 óra	Kecskés, Kiss, Kis, Kobál, Martonosi	<b>Értekezlet:</b> Az átbeszélt dolgok ellenőrzése, hogy megcsinálta-e mindenki és együtt javítása
2019.05.11 13:00	3.0 óra	Kiss	A korábbi hibák az Exit, Entry, és Controller osztályokban kijavítva
2019.05.11 14:00	1.0 óra	Martonosi	A mainFloor.flr elkészítése
2019.05.11 15:00	3.5 óra	Kis	A view kidolgozása (jbutton alapon)
2019.05.11 16:00	2.0 óra	Kobál	MVC architektúra megvalósítása
2019.05.11 16:00	1.0 óra	Kecskés	A View egy lehetséges kidolgozása és deserializálás megírása

			(.vw file beolvasás Node listába)
2019.05.12 13:30	2.0 óra	Kiss	A korábbi hibák az Exit, Entry, és Controller osztályokban kijavításra kerültek külön brachben(elod_branch_entry_fix)
2019.05.12 16:00	0.5 óra	Kecskés, Kobál, Martonosi	<b>Értekezlet:</b> Megbeszélés, hogy még naaaagyon sok minden van hátra, haladni kéne, mert kifutunk az időből. Megoldás: üljünk össze hároman és fejezzük be a kódot, mert különben nem lesz semmi.
2019.05.12 16:30	6.0 óra	Kecskés	A View végleges kidolgozása: Draw .png megvalósítása, vonalak összekötése+kirajzolása, Swingworker implementálása a responzivitás érdekében
2019.05.12 16:30	8.0 óra	Kobál	A Controller singeltonná alakítása illesztése a View mellé, minden dokumentálása
2019.05.12 16:30	7.0 óra	Martonosi	A mainFloor.vw elkészítése, keylistener megírása és hozzáadása és view controllerrel összehangolása
2019.05.13 10:45	1.5 óra	Kecskés	Dokumentum formázása, apróbb javítások a projektben (serializálás javítása, Gameover dialog) és Százalékok összeszámolása mindenkinek

## Összefoglalás

### A projektre fordított összes munkaidő

*A munkaórák a naplóban megadott eredmények alapján számoltak*

Tag neve	Munkaidő (óra)
<i>Kecskés Péter</i>	39
<i>Kis Máté Levente</i>	28
<i>Kiss Előd</i>	26
<i>Kobál Sándor</i>	35,5
<i>Martonosi Máté</i>	36,5
<b>Összesen</b>	<b>165</b>

- A feltöltött programok forrásainak száma**

A kódsorok számlálásához a clock nevű github alkalmazást használtam (clock.exe grafikusbeadas.java)

Fázis	Kódsorok száma
Szkeleton	949
Prototípus	1349
Grafikus változat	1474
<b>Összesen</b>	<b>3772</b>

- Projekt összegzés**

### Mit tanultak a projektből konkrétan és általában?

Megtanultuk, hogy hogyan kell egy gyűlést megszervezni és lebonyolítani pontosan és hatékonyabban, mint előtte. Rájöttünk, hogy egy szoftver fejlesztésének menetében sokkal több lépés is van (pl. proto/skeleton kell a grafikus előtt) és a szekvencia diagramok kellenek már a projekt korábbi szakaszába is, hogy mindenki lássa, hogy a függvény-függvény kapcsolat pontosan mit is csinálna belülről.

### Mi volt a legnehezebb és a legkönnyebb?

Legnehezebb biztosan az volt, hogy a projekt megfeleljen az objektumorientált programozás minden kritériumának. Ami ezután legnehezebb volt számunkra, az a grafikus felület kirajzolásának megtervezése úgy, hogy figyelembe vegyük azt is, hogy az MVC modellt betartva be kell tartanunk az információáramlás menetét.

Legkönnyebb szerintem az alapok összerakása, mert nagyon jól befogadható játék ötletet kaptunk megvalósításra. Könnyű volt még illeszteni a projektet a megváltozott elvárásoknak (elengedi a pandát és lophatja a másik pandáját orángután). Ez valószínű azért volt könnyű, mert a projektet oo módon megtervezve és betartva a rendet virtuális/abstract osztályok, interface-ek, heterogén kollekciók használatával jó vázat lehet építeni.

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

A munkaórák nem mindig mérvadóak az idővel, mert vannak olyan tagjai a csapatnak, akiknek könnyedén megy a szekvencia és dokumentáció szerkesztése, míg a kódolás nagyon nehezen (vice versa). Ami még sokszor problémát jelentett, hogy néhány ember nem ért rá az időpontokban (külföldre utazás, vagy táborok szünetek) így nem segített úgy.

### Ha nem, akkor hol okozott ez nehézséget?

Fontos dolog, hogy a csapatunkban történt gyűlések sokszor effektívebbek voltak, mint az apróbb felosztások. Volt többször olyan, hogy gyűlések alatt egy ember osztott asztalt és a többiek segítettek diktálni és megbeszélni a dolgot, hogy hatékonyabban dolgozzunk. Tehát a gyűlések fontos szerepet játszottak az időben, mégsem egyéni munkaórának volt elkönnyelve.

Ami még a munkaóránál eltérhet a valós munkával, hogy a csapat megegyezett abban, hogy a hasznos munkát számítjuk elsődlegesen bele a naplóba, tehát amikor főlegesen dolgozott, vagy csak játszott a projekttel egy külön branch-ben ami végül nem is kellett, akkor azt nem számítjuk bele, vagy csak felezett óraszámba.

### **Milyen változtatási javaslatuk van?**

-

### **Milyen feladatot ajánlanának a projektre?**

Jövőre is valami elvontabb pályát megjelenítő játékot, mert így nem csaphatja össze a csapat a grafikus megjelenítést.

### **Egyéb kritika és javaslat**

-