**7. Prototípus koncepciója**

66 – otthonmeglefordult

Konzulens:

Szőke Máté

Csapattagok

|  |  |  |
| --- | --- | --- |
| Kovács Ákos | H7FTHG | [akoskovacs1980@gmail.com](mailto:akoskovacs1980@gmail.com) |
| Kaszala Kristóf | S9XEU5 | kaszalakristof1@gmail.com |
| Lőrincz Zoltán | BUXM3K | zoltan9811@gmail.com |
| Horesnyi Olivér | D7DBWE | horesnyi.oliver@freemail.hu |
| Hulej Attila | VTZ65K | alittahu@gmail.com |

2019.03.31.

Tartalomjegyzék

[7. Prototípus koncepciója 3](#_Toc4843514)

[7.0 Változás hatása a modellre 3](#_Toc4843515)

[7.0.1 Módosult osztálydiagram 3](#_Toc4843516)

[7.0.2 Új vagy megváltozó metódusok 5](#_Toc4843517)

[7.0.2.1 Új metódusok: 5](#_Toc4843518)

[7.0.2.2 Módosult metódusok: 5](#_Toc4843519)

[7.0.3 Szekvencia-diagramok 6](#_Toc4843520)

[7.0.3.1 Orangutan release pandas – Új diagram 6](#_Toc4843521)

[7.0.3.2 Orangutan hitBy orangutan – Új diagram 7](#_Toc4843522)

[7.0.3.3 Orangutan steppedOnField – Módosult diagram 8](#_Toc4843523)

[7.0.3.4 Orangutan steppedOnWeaktile – Módosult diagram 9](#_Toc4843524)

[7.0.3.5 Orangutan steppedOnExit – Módosult diagram 10](#_Toc4843525)

[7.0.3.6 Orangutan steppedOnWardrobe – Módosult diagram 11](#_Toc4843526)

[7.0.3.7 Panda hitByOrangutan – Módosult diagram 12](#_Toc4843527)

[7.1 Prototípus interface-definíciója 13](#_Toc4843528)

[7.1.1 Az interfész általános leírása 13](#_Toc4843529)

[7.1.2 Bemeneti nyelv 13](#_Toc4843530)

[7.1.2.1 Általános vezérlés utasításai 13](#_Toc4843531)

[7.1.2.2 Létrehozó és beállító utasítások 14](#_Toc4843532)

[7.1.2.2.1 Panda és pandatípusok 14](#_Toc4843533)

[7.1.2.2.1.1 Panda 14](#_Toc4843534)

[7.1.2.2.1.2 LazyPanda 14](#_Toc4843535)

[7.1.2.2.1.3 GamblerPanda 14](#_Toc4843536)

[7.1.2.2.1.4 ChocolatePanda 14](#_Toc4843537)

[7.1.2.2.2 Orángután 15](#_Toc4843538)

[7.1.2.2.3 Állatokra vonatkozó parancsok 15](#_Toc4843539)

[7.1.2.2.4 Mezők 16](#_Toc4843540)

[7.1.2.2.4.1 Field 16](#_Toc4843541)

[7.1.2.2.4.2 Weaktile 16](#_Toc4843542)

[7.1.2.2.4.3 Game machine 16](#_Toc4843543)

[7.1.2.2.4.4 Chocolate machine 17](#_Toc4843544)

[7.1.2.2.4.5 Armchair 17](#_Toc4843545)

[7.1.2.2.4.6 Entrance 17](#_Toc4843546)

[7.1.2.2.4.7 Exit 17](#_Toc4843547)

[7.1.2.2.4.8 Wardrobe 18](#_Toc4843548)

[7.1.2.2.4.9 Minden mezőre vonatkozó parancsok 18](#_Toc4843549)

[7.1.2.2.5 Egyéb osztályok 19](#_Toc4843550)

[7.1.2.2.5.1 Timer 19](#_Toc4843551)

[7.1.2.2.5.2 Game 19](#_Toc4843552)

[7.1.2.2.5.3 Player 19](#_Toc4843553)

[7.1.2.3 Pálya leírása 20](#_Toc4843554)

[7.1.2.3.1 Prototípus esetén 20](#_Toc4843555)

[7.1.2.3.2 Grafikus felület esetén 21](#_Toc4843556)

[7.1.3 Kimeneti nyelv 23](#_Toc4843557)

[7.2 Összes részletes use-case 23](#_Toc4843558)

[7.3 Tesztelési terv 23](#_Toc4843559)

[7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása 23](#_Toc4843560)

[7.5 Napló 24](#_Toc4843561)

# Prototípus koncepciója

[A prototípus program célja annak demonstrálása, hogy a program elkészült, helyesen működik, valamennyi feladatát teljesíti. A prototípus változat egy elkészült program kivéve a kifejlett grafikus interfészt. Ez a program is parancssorból futtatható és karakteres ernyőkezelést alkalmaz. Az ütemezés, az aktív objektumok kezelése megoldott. A business objektumok - a megjelenítésre vonatkozó részeket kivéve - valamennyi metódusa a végleges algoritmusokat kell, hogy tartalmazza. A megjelenítés és működtetés egy alfanumerikus képernyőn vezérelhető és követhető, ugyanakkor a vezérlés fájlból is történhet és a megjelenítés fájlba is logolható, ezzel megteremtve a rendszer tesztelésének lehetőségét. Különös figyelmet kell fordítani a parancssori interfész logikájára, felépítésére, valamint arra, hogy az mennyiben tükrözi és teszi láthatóvá a program működését, a beavatkozások hatásait.]

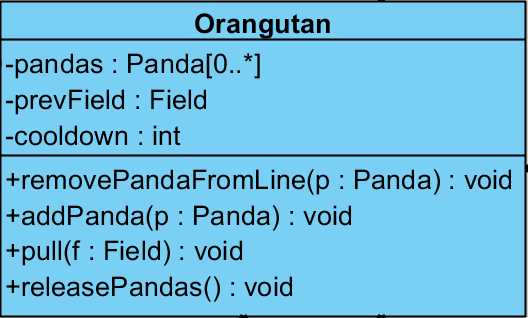
## Változás hatása a modellre

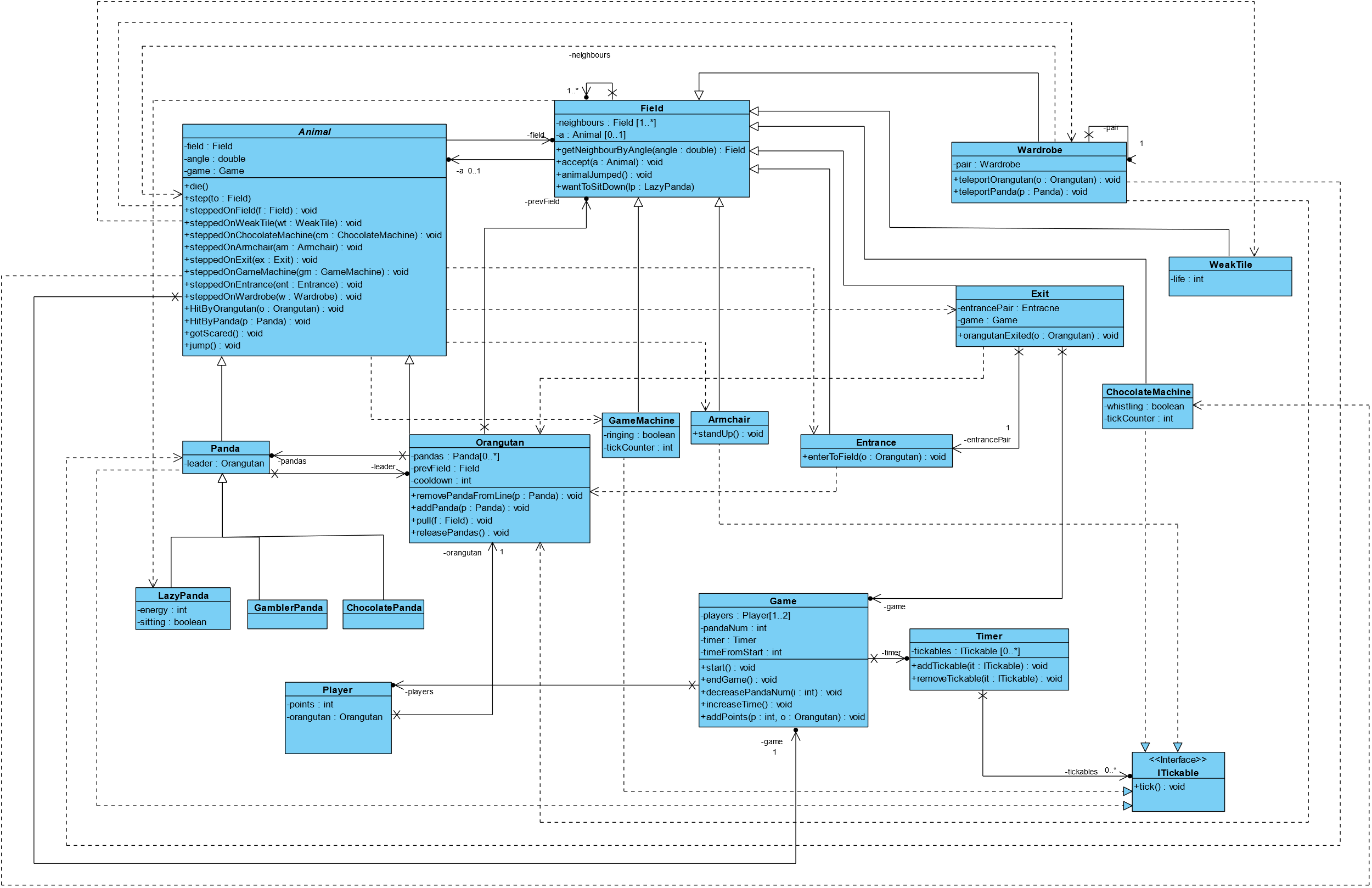
* Az orángutánok el tudják engedni a pandákat. Ilyenkor az érintett sor felbomlik.
* Az orángutánok elrabolhatják egymás pandáit. Ennek menete, hogy ha egy orángután (o1), akinek nincs pandája, nekimegy egy másik orángutánnak (o2), akkor helyet cserélnek, és o1 megszerzi o2 panda-sorát. Ekkor o2-nek 3 lépést kell tennie, mielőtt újra pandát foghat vagy másik orángutántól rabolhat.

### Módosult osztálydiagram

* Orangutan osztályban új metódus: releasePandas() : void
* Orangutan osztályban új privát tagváltozó: cooldown : int

A változás csak az Orangutan osztályt érintette, de a következő oldalon megtalálható a teljes osztálydiagram is.





### Új vagy megváltozó metódusok

#### Új metódusok:

* releasePandas()

**Osztály:** Orangutan  
**Leírás:** Felszabadítja az orangutan pandáit.

* hitByOrangutan(Orangutan o)  
  **Osztály:** Orangutan

**Leírás:** Legyen az az orangután akinek nekimennek o2, aki nekimegy o2-nek pedig o1. Ekkor ha o1-nek nincs pandája és lejárt a 3 lépéses cooldown-ja is, akkor o1 helyet cserél o2-vel. Ezután o1 megkapja o2 pandáit, és o2 cooldown-ja felveszi a 3 értéket.

#### Módosult metódusok:

* steppedOnField(Fieldf)  
  **Osztály:** Orangutan  
  **Leírás:** Mivel lépés történhet, csökkenteni kell a cooldown attribútum értékét.
* steppedOnWeaktile(Weaktile wt)  
  **Osztály:** Orangutan  
  **Leírás:** Mivel lépés történhet, csökkenteni kell a cooldown attribútum értékét.
* steppedOnExit(Exit e)  
  **Osztály:** Orangutan  
  **Leírás:** Mivel lépés történhet, csökkenteni kell a cooldown attribútum értékét.
* steppedOnWardrobe(Wardrobe w)  
  **Osztály:** Orangutan  
  **Leírás:** Mivel lépés történhet, csökkenteni kell a cooldown attribútum értékét.
* hitByOrangutan(Orangutan o)  
  **Osztály:** Panda

**Leírás:** Csak akkor történik meg az pandával való ütközés és befűzés, ha a cooldown értéke már lejárt.

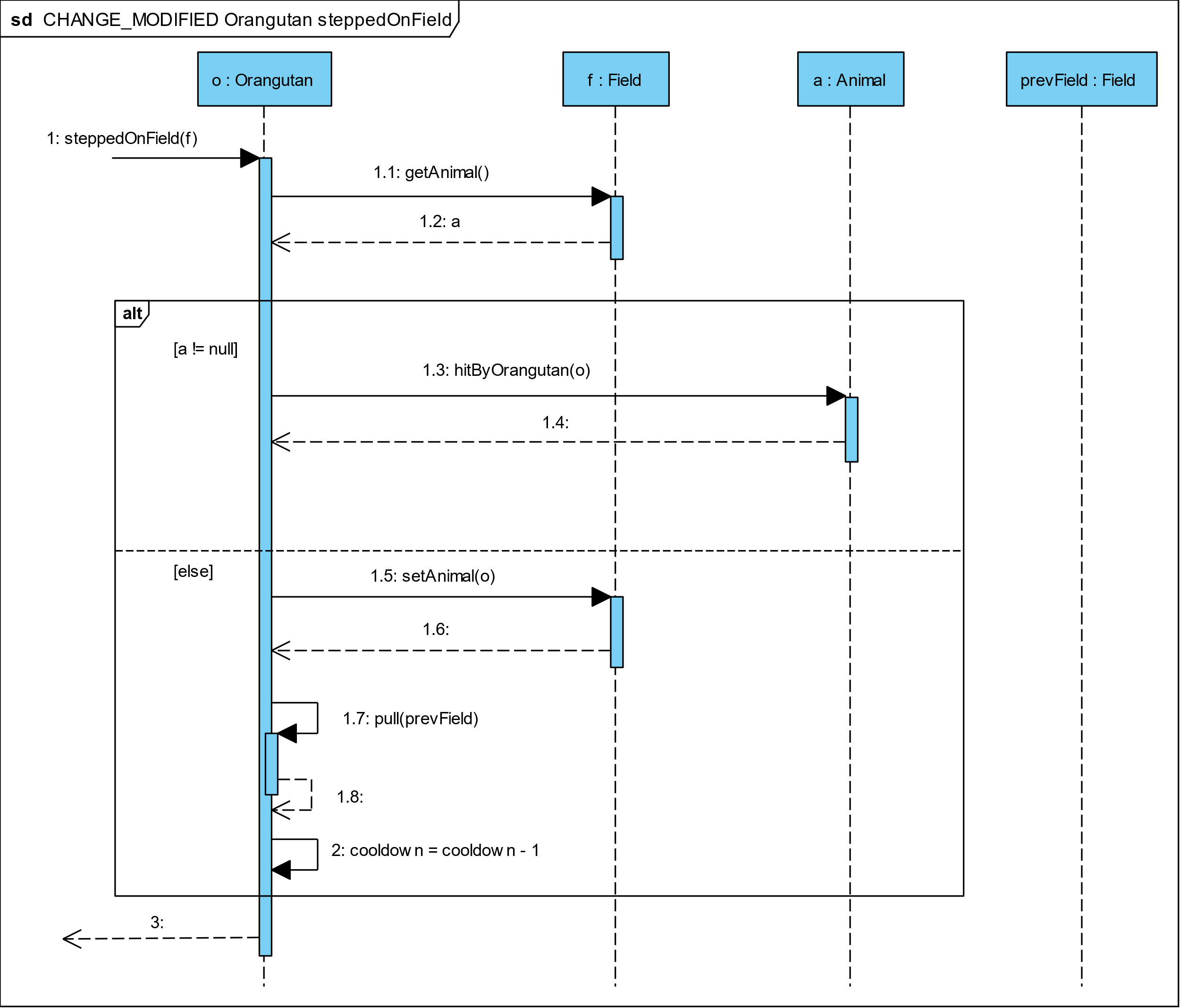
### Szekvencia-diagramok

#### Orangutan release pandas – Új diagram

#### Orangutan hitBy orangutan – Új diagram

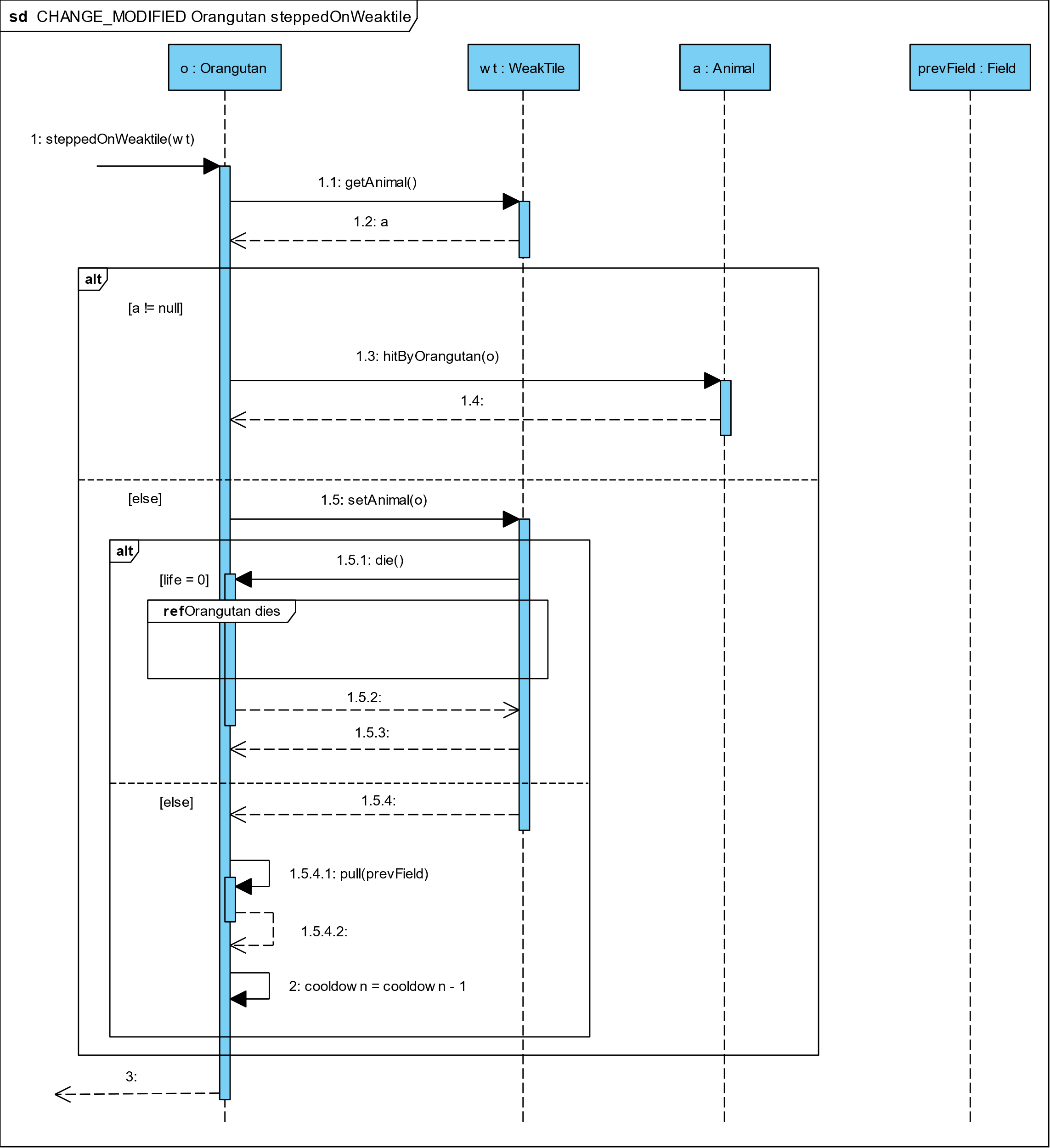
#### Orangutan steppedOnField – Módosult diagram

A módosítás (2-es sorszám) az *else* ágban található, a *pull* metódushívás után csökkenteni kell a *cooldown* értékét, mivel lépés történt.

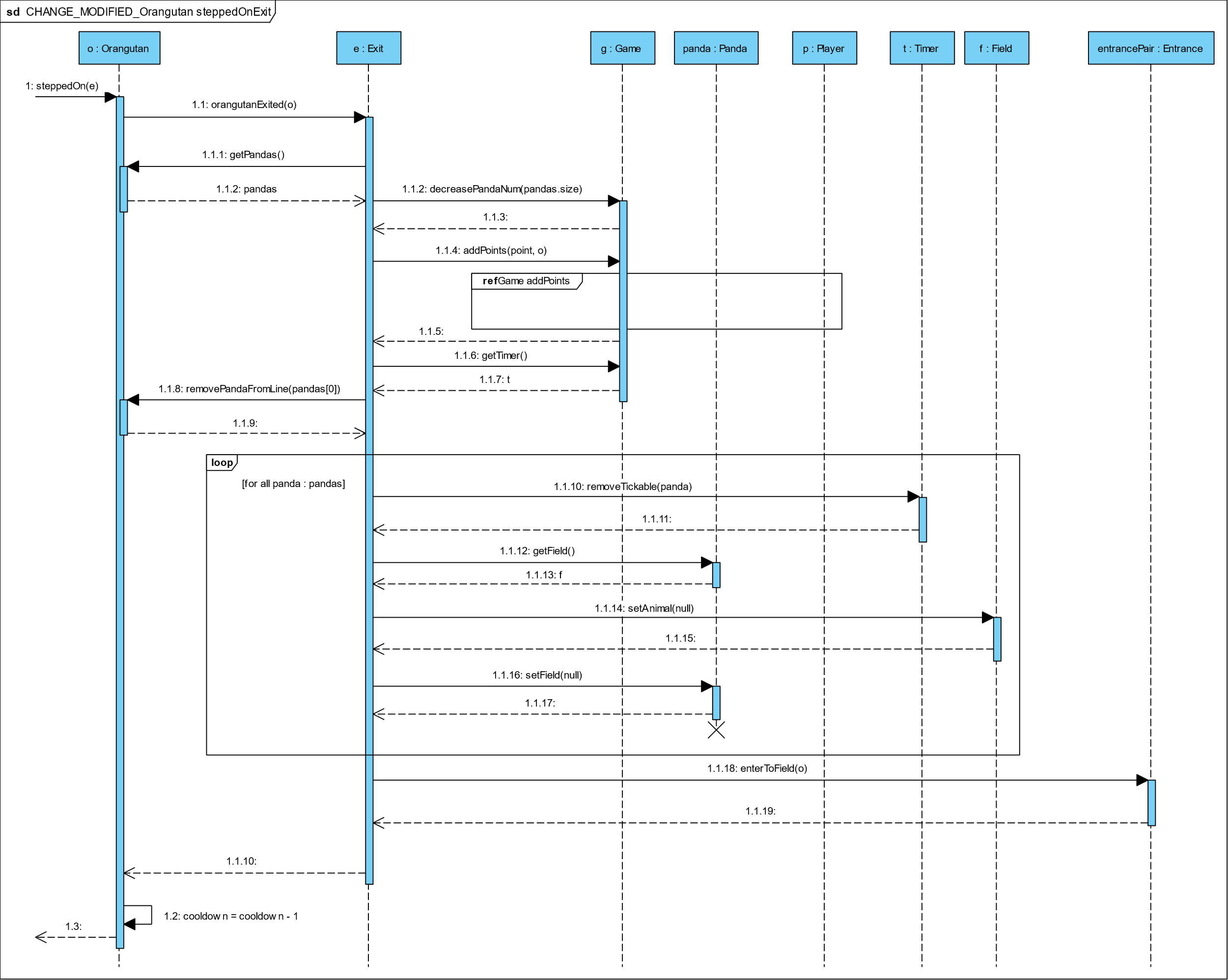


#### Orangutan steppedOnWeaktile – Módosult diagram

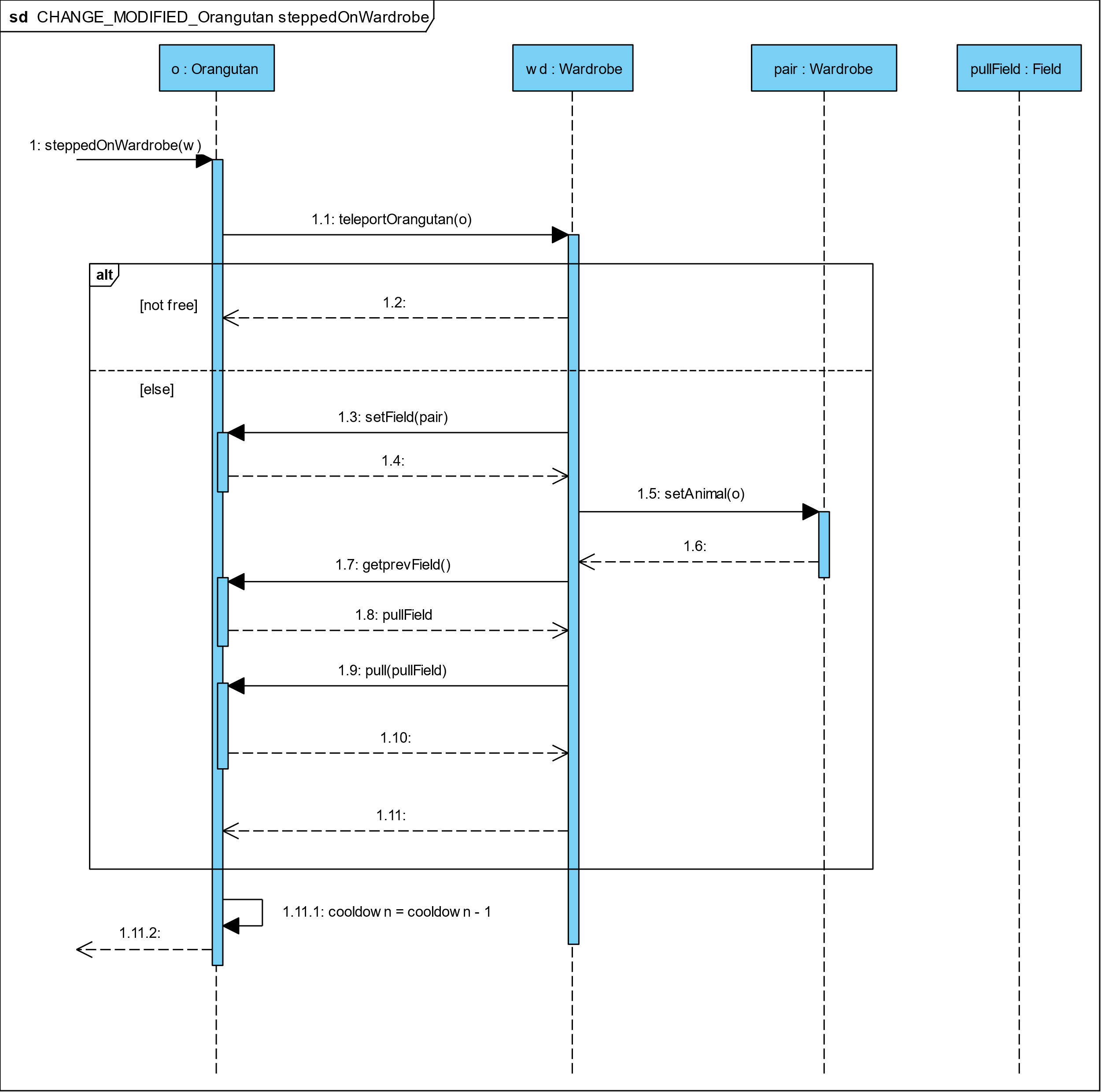
A módosítás (2-es sorszám), második alt else ágában található, a pull metódushívás után csökkenteni kell a cooldown értékét, mivel lépés történt.



#### Orangutan steppedOnExit – Módosult diagram

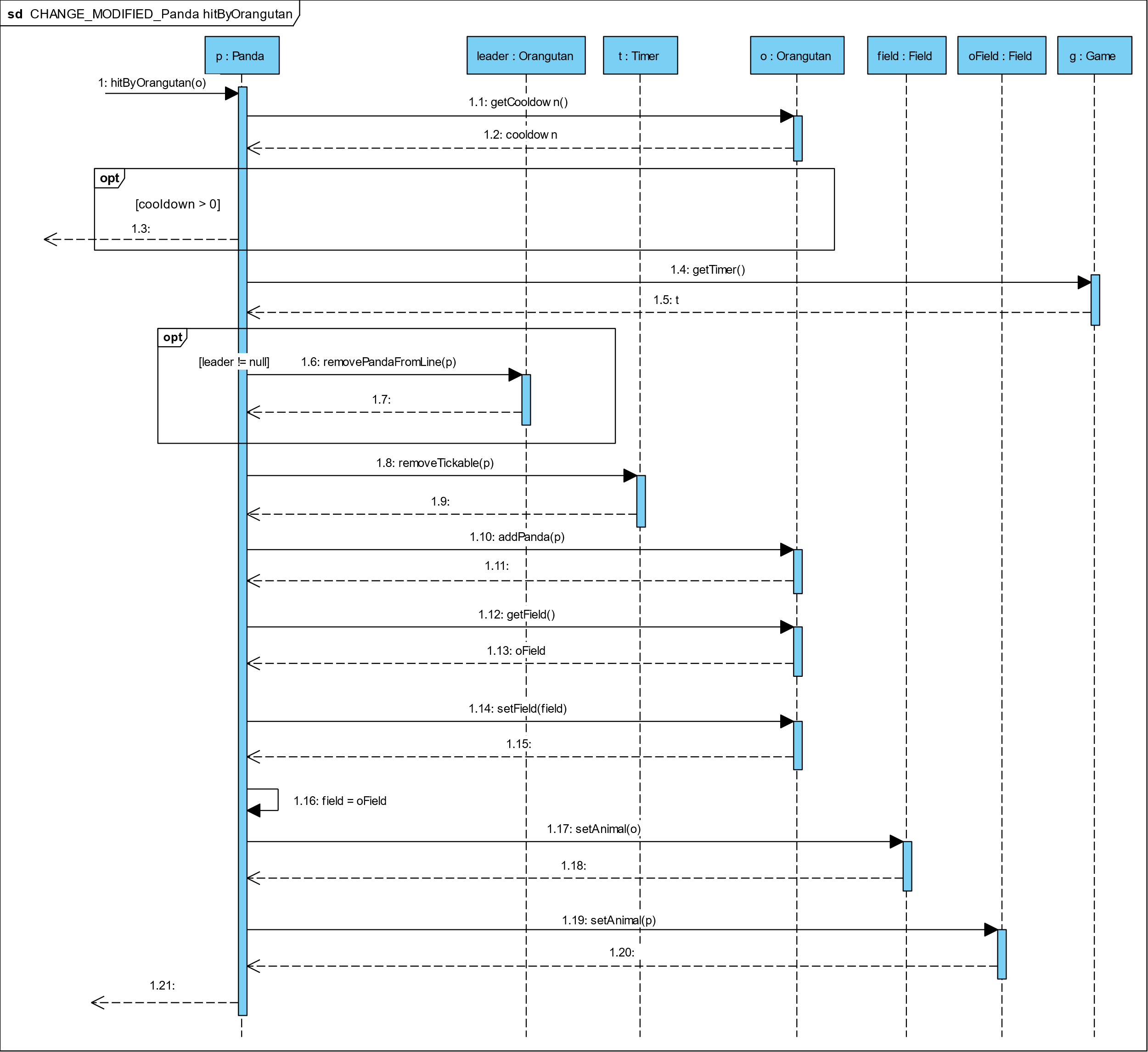
  
A módosítás (1.2-es sorszám), az orangutanExited visszatérése után látható, lépés miatt kell csökkenteni a cooldown értékét.

#### Orangutan steppedOnWardrobe – Módosult diagram

  
A változás (1.11-es sorszám) a teleportOrangutan hívás visszatérése után került be.

#### Panda hitByOrangutan – Módosult diagram

A módosítás hatása az 1.1, 1.2, 1.3 és az 1.3 visszatérést tartalmazó elágazásban valósult meg. Csak akkor kell a pandával való ütközést vizsgálni, ha már lejárt a cooldown. Ha nem járt le akkor nem kell csinálni semmit, és visszatérünk.



## Prototípus interface-definíciója

[Definiálni kell a teszteket leíró nyelvet. Külön figyelmet kell fordítani arra, hogy ha a rendszer véletlen elemeket is tartalmaz, akkor a véletlenszerűség ki-bekapcsolható legyen, és a program determinisztikusan is tesztelhető legyen.]

### Az interfész általános leírása

[A protó (karakteres) input és output felületeit úgy kell kialakítani, hogy az input fájlból is vehető legyen illetőleg az output fájlba menthető legyen, vagyis kommunikációra csak a szabványos be- és kimenet használható.]

### Bemeneti nyelv

[Definiálni kell a teszteket leíró nyelvet (szintakszis és szemantika). Külön figyelmet kell fordítani arra, hogy ha a rendszer véletlen elemeket is tartalmaz, akkor a véletlenszerűség ki-bekapcsolható legyen, és a program determinisztikusan is futtatható legyen. A szálkezelést is tesztelhető, irányítható módon kell megoldani. A programot egy adott konfigurációból is el kell tudni indítani, vagyis kell olyan parancs, amivel konkrét előre megadott állapotból indul a rendszer (pl. load).]

#### Általános vezérlés utasításai

#[string]

Leírás: Komment a bemeneti fájlokban.

Opciók:

random [on/off]

Leírás: Be- vagy kikapcsolja a véletlenszerűséget a játékban.

Opciók: on : Bekapcsolja a véletlenszerűséget.

off: Kikapcsolja a véletlenszerűséget.

load [infile.txt]

Leírás: Betölti az adott konfigurációs fájlt.

Opciók: infile.txt – txt formátumú fájlnév

loadmap [infile.txt]

Leírás: Betölti az adott pályát. Később, a grafikus felületnél lesz rá szükség, mivel ott a pályának a megadása eltér a prototípusban használttól.

Opciók: infile.txt – txt formátumú fájlnév

getstate [name]

Leírás: Kiírja az adott objektum adatait, állapotát a konzolra.

Opciók: name – Az objektum neve.

save [outfile.txt]

Leírás: Kiírja az összes objektum állapotát a megadott fájlba.

Opciók: name – Az objektum neve.

#### Létrehozó és beállító utasítások

##### Panda és pandatípusok

###### Panda

create panda [name] [game] [leader=null] [field=null]

Leírás: Létrehoz egy pandát.

Opciók: name – A panda neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

game – A játék, amiben a panda részt vesz.

leader – A panda vezetője.

Ha van neki akkor annak az orángutánnak a neve.

Ha nincs akkor lehet null értéket is írni, de kötelező valamit írni.

###### LazyPanda

create lazypanda [name] [game] [leader=null] [field=null] [energy] [sitting]

Leírás: Létrehoz egy lusta pandát.

Opciók: name – A lusta panda neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

set lp\_energy [lazypanda\_name] [int\_to\_set]

Leírás:

Opciók:

set lp\_sitting [lazypanda\_name] [true/false]

Leírás:

Opciók:

###### GamblerPanda

create gamblerpanda [name] [game] [leader=null] [field=null]

Leírás: Létrehoz egy játékos pandát.

Opciók: name – A játékos panda neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

###### ChocolatePanda

create chocolatepanda [name] [game] [leader=null] [field=null]

Leírás: Létrehoz egy csokipandát.

Opciók: name – A csokipanda neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

##### Orángután

create orangutan [name] [game] [field=null]

Leírás: Létrehoz egy orángutánt.

Opciók: name – Az orángután neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

add panda [orangutan\_name] [panda]

Leírás:

Opciók:

##### Állatokra vonatkozó parancsok

set a\_field [animal] [field]

Leírás:

Opciók:

step [animal] [field]

Leírás:

Opciók:

##### Mezők

###### Field

create field [name] [animal=null]

Leírás: Létrehoz egy csempét.

Opciók: name – A csempe neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

set f\_animal [field\_name] [animal]

Leírás: Létrehoz egy csempét.

Opciók: name – A csempe neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

###### Weaktile

create weaktile [name] [animal=null] [life]

Leírás: Létrehoz egy csempét.

Opciók: name – A csempe neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

set wt\_animal [weaktile\_name] [animal]

Leírás: Létrehoz egy csempét.

Opciók: name – A csempe neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

set life [weaktile\_name] [int\_to\_set]

Leírás:

Opciók:

###### Game machine

create gamemachine [name] [ringing=false]

Leírás: Létrehoz egy játékgépet.

Opciók: name – A játékgép neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

set gm\_tickcounter [game\_machine\_name] [int\_value]

Leírás: A játékgép tickcounter változóját állítja. A gép egy konstans tick után átmegy csilingelő állapotból zajtalan állapotba. A tickcounter tartja nyilván, hogy hány tick óta vagyunk csilingelő állapotban.

Opciók: machine\_name – A játékgép neve.

int\_value – Az tickszámláló értéke.

set gm\_ringing [game\_machine\_name] [true/false]

Leírás: A játékgép csilingelését állítja.

Opciók: true – A játékgép csilingek.

false – A játékgép nem csilingel.

###### Chocolate machine

create chocolatemachine [name] [whistling=false]

Leírás: Létrehoz egy csokigépet.

Opciók: name – A csokigép neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

set cm\_tickcounter [chocolate\_machine\_name] [int\_value]

Leírás: A csokiautomata tickcounter változóját állítja. A gép egy konstans tick után átmegy sípoló állapotból zajtalan állapotba. A tickcounter tartja nyilván, hogy hány tick óta vagyunk sípoló állapotban.

Opciók: machine\_name – A csokiautomata neve.

int\_value – Az tickszámláló értéke.

set cm\_whistling [chocolate\_machine\_name] [true/false]

Leírás: A csokigép sípolását állítja.

Opciók: true – A csokigép sípol.

false – A csokigép nem sípol.

###### Armchair

create armchair [name]

Leírás: Létrehoz egy fotelt.

Opciók: name – A fotel neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

###### Entrance

create entrance [name]

Leírás: Létrehoz egy bejáratot.

Opciók: name – A bejárat neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

###### Exit

create exit [name] [entrancepair] [game]

Leírás: Létrehoz egy kijáratot.

Opciók: name – A kijárat neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

###### Wardrobe

create wardrobe [name] [wpair]

Leírás: Létrehoz egy szekrényt.

Opciók: name – A szekrény neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

###### Minden mezőre vonatkozó parancsok

add neighbour [field\_to] [field\_to\_add]

Leírás: Adott mezőhöz adott mezőt ad hozzá szomszédként.

Opciók: field\_to – A mező, amihez hozzá akarjuk adni a szomszédot.

field\_to\_add – A hozzáadandó mező neve.

##### Egyéb osztályok

###### Timer

create timer [name]

Leírás: Létrehoz egy Timer-t.

Opciók: name – A timer neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

add tickable [timer] [tickable]

Leírás: Létrehoz egy Timer-t.

Opciók: name – A timer neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

###### Game

create game [name] [pandanum] [timer]

Leírás: Létrehoz egy játékot.

Opciók: name – A játék neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

add player [game] [player]

Leírás: Létrehoz egy Timer-t.

Opciók: name – A timer neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

###### Player

create player [name] [orangutan] [points=0]

Leírás: Létrehoz egy játékost.

Opciók: name – A játékos neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

#### Pálya leírása

##### Prototípus esetén

Mivel a prototípusban nem kell grafikusan kirajzolni a pályát, ezért a különböző csempék alakját nem kell definiálni, csak a szomszédsági kapcsolatokat kell megadni. Ez a fentebb tárgyalt parancsokkal megoldható:

# objektumok létrehozása

create timer t1

create game g 2 t1

create orangutan o1 g null

create panda p1 g null null

create panda p2 g null null

# f1 mezőn o1 áll

create field f1 o1

set a\_field o1 f1

# f2 mezőn p1 áll

create field f2 p1

set a\_field p1 f2

# f3 mezőn p2 áll

create field f3 p2

set a\_field p2 f3

# f4 mezőn nem áll senki

create weaktile wt1 null 20

# szomszédságok

# f1 <-> f2

add neighbour f1 f2

add neighbour f2 f1

#f2 <-> f3

add neighbour f2 f3

add neighbour f3 f2

#wt1 <-> f1

add neighbour wt1 f1

add neighbour f1 wt1

# orángután lépjen wt1-re

step o1 wt1

##### Grafikus felület esetén

A grafikus felülethez a pályát már úgy kell megadni, hogy az kirajzolható legyen, tehát a csempék alakja is fontos. Ehhez a csúcsok koordinátáit, és a csempék „középpontjait” (a csempe egy belső pontja, ahova majd az állat textúrája kerülhet) is tudnunk kell.

Ez az alább leírtakhoz hasonlóan fog megvalósulni, de a haladás közben apróbb javítások még történhetnek. A # karakterrel kezdődő sorok itt is kommentek.

# A koordinátákat óramutatóval ellentétes sorrendben kell #megadni. Az szomszédok kiválasztása a belső szögek alapján   
#fog történni, ezért az a pont, amihez a szögszámolást #viszonyítjuk annak kell az utolsónak lenni.

# A K karakterrel kezdődő sor adja meg a középpontot.

# Egy mező

[mező típusa]

[mező sorszáma/azonosítója/neve]

[x;y]

[x;y]

…

[x;y]

K [x;y]

# Másik mező

[mező típusa]

[mező sorszáma/azonosítója/neve]

[x;y]

…

[x;y]

K [x;y]

#szomszédságok beállítása

szomszedok

[mező\_azon1]: [sz\_azon1] [sz\_azon2] …

[mező\_azon2]: [sz\_azon1] [sz\_azon4] …

…

[mező\_azonX]: [sz\_azonY] [sz\_azonZ] …

#állatok elhelyezése

#orángután

orangutanpos

[orangután\_név1] [mező\_azon1]

[orángután\_név2] [mező\_azon2]

#pandák

pandapos random:false

[panda\_név1] [mező\_azon1]

…

[panda\_névX] [mező\_azonY]

Ha azt akarjuk, hogy a pandák helye véletlenszerű legyen:

pandapos random:true [num\_of\_pandas]

Ezután nem kel felsorolni a pandákat.

Például:

field

1

2;5

1;3

2;1

6;2

5;4

3;5

K 3;3

weaktile

2

2;6

2;5

3;5

3;6

K 2.5;5.5

armchair

3

1;5

1;3

2;5

K 1.2;4.3

szomszedok

1: 2 3

2: 1

3: 1

orangutanpos

o1 1

pandapos random:true 10

A pálya koordinátás megadásához segítségül használható például a GeoGebra program, ahol tervezhetünk egy sokszögekből álló pályát, és a koordinátákat is leolvashatjuk.

### Kimeneti nyelv

[Egyértelműen definiálni kell, hogy az egyes bemeneti parancsok végrehajtása után előálló állapot milyen formában jelenik meg a szabványos kimeneten. A program képes legyen olyan kimenetet előállítani, amellyel az objektumok állapota ellenőrizhető (pl. save). Ebben az alfejezetben is precízen definiálni kell, hogy a kimenet nyelve milyen elemekből és milyen szintakszissal áll elő.]

Az egyetlen bemeneti parancs, ami ír valamit a szabványos kimenetre az a „getstate [name]” parancs, a többi parancs nem ír ki semmit. A getstate paraméterként egy objektum nevét kapja és kiírja az objektum adattagjait a következő formátumban (a […] helyére kerülnek az egyes adattagok értékei):

**Panda**

kimenet: Panda: [name], g: [game\_n], l: [leader\_n], f: [field\_n]

* name: A panda egyedi neve.
* game\_n: A játék neve, amiben a panda részt vesz.
* leader\_n: A pandát vezető orángután neve, vagy ha nincs, akkor „null” az értéke.
* field\_n: A mező neve, amin a panda áll, „null” az értéke, ha valamiért nem lett mező hozzárendelve.

**LazyPanda**

kimenet: LazyPanda: [name], g: [game\_n], l: [leader\_n], f: [field\_n], e: [energy], s: [sitting]

* name: A lusta panda egyedi neve.
* game\_n: A játék neve, amiben a lusta panda részt vesz.
* leader\_n: A lusta pandát vezető orángután neve, vagy ha nincs, akkor „null” az értéke.
* field\_n: A mező neve, amin a lusta panda áll, „null” az értéke, ha valamiért nem lett mező hozzárendelve.
* energy: A lusta panda energiája egész számban kifejezve.
* sitting: Értéke „true”, ha a lusta panda fáradt és „false” egyébként.

**GamblerPanda**

kimenet: GamblerPanda: [name], g: [game\_n], l: [leader\_n], f: [field\_n]

* name: A játékos panda egyedi neve.
* game\_n: A játék neve, amiben a játékos panda részt vesz.
* leader\_n: A játékos pandát vezető orángután neve, vagy ha nincs, akkor „null” az értéke.
* field\_n: A mező neve, amin a játékos panda áll, „null” az értéke, ha valamiért nem lett mező hozzárendelve.

**ChocolatePanda**

kimenet: ChocolatePanda: [name], g: [game\_n], l: [leader\_n], f: [field\_n]

* name: A csokipanda egyedi neve.
* game\_n: A játék neve, amiben a csokipanda részt vesz.
* leader\_n: A csokipandát vezető orángután neve, vagy ha nincs, akkor „null” az értéke.
* field\_n: A mező neve, amin a csokipanda áll, „null” az értéke, ha valamiért nem lett mező hozzárendelve.

**Orangutan**

kimenet: Orangutan: [name], g: [game\_n], f: [field\_n], pf: [prevfield\_n], cd: [cd], p: [[p1] [p2] … [pn]]

* name: Az orángután egyedi neve.
* game\_n: A játék neve, amiben az orángután részt vesz.
* field\_n: A mező neve, amin az orángután áll, „null” az értéke, ha valamiért nem lett mező hozzárendelve.
* prevfield\_n: A mező neve, amiről az orángután a legutolsó lépésével ellépett. Ha nincs ilyen „null” az értéke.
* cd: A lépések száma, amit az orángutánnak meg kell még tennie, mielőtt újra tud lopkodni más orángutántól.
* pn: A pandák egyedi neve szóközzel elválasztva és sorrendben, akik az orángután láncában vannak (p1 panda az első a láncban, p2 a második és pn az utolsó).

**Field**

kimenet: Field: [name], a: [animal\_n], n: [[n1] [n2] … [nk]]

* name: A mező neve.
* animal\_n: Az állat neve, ami a mezőn áll, „null” az értéke, ha üres a mező.
* nk: A mezővel szomszédos mezők nevei, szóközzel elválasztva egymástól.

**WeakTile**

kimenet: WeakTile: [name], a: [animal\_n], l: [life], n: [[n1] [n2] … [nk]]

* name: A csempe neve.
* animal\_n: Az állat neve, ami a csempén áll, „null” az értéke, ha üres a mező.
* life: A csempe életereje.
* nk: A csempével szomszédos mezők nevei, szóközzel elválasztva egymástól.

**GameMachine**

kimenet: GameMachine: [name], a: [animal\_n], r: [ringing], tc: [tickcounter], n: [[n1] [n2] … [nk]]

* name: A játékgép neve.
* animal\_n: Az állat neve, ami a játékgépen áll, „null” az értéke, ha nem áll senki a játékgépen.
* ringing: Értéke „true”, ha csörög a játékgép, „false” egyébként.
* tickcounter: : Értéke a körök száma, amióta a játékgép csörög.
* nk: A mezővel szomszédos mezők nevei, szóközzel elválasztva egymástól.

**ChocolateMachine**

kimenet: ChocolateMachine: [name], a: [animal\_n], w: [whistling], tc: [tickcounter], n: [[n1] [n2] … [nk]]

* name: A csokigép neve.
* animal\_n: Az állat neve, ami a csokigépen áll, „null” az értéke, ha nem áll senki a csokigépen.
* whistling: Értéke „true”, ha sípol a csokigép, „false” egyébként.
* tickcounter: Értéke a körök száma, amióta a csokigép sípol.
* nk: A csokigéppel szomszédos mezők nevei, szóközzel elválasztva egymástól.

**Armchair**

kimenet: Armchair: [name], a: [animal\_n], n: [[n1] [n2] … [nk]]

* name: A fotel neve.
* animal\_n: Az állat neve, ami a fotelben ül, „null” az értéke, ha nem ül senki a fotelben.
* nk: A fotellel szomszédos mezők nevei, szóközzel elválasztva egymástól.

**Entrance**

kimenet: Entrance: [name], a: [animal\_n], n: [[n1] [n2] … [nk]]

* name: A bejárat neve.
* animal\_n: Az állat neve, ami a bejáraton áll, „null” az értéke, ha nem áll senki a bejáraton.
* nk: A bejárattal szomszédos mezők nevei, szóközzel elválasztva egymástól.

**Exit**

kimenet: Exit: [name], g: [game], a: [animal\_n], ep: [entrancepair], n: [[n1] [n2] … [nk]]

* name: A kijárat neve.
* game: A játék neve, amiben a kijárat szerepel.
* animal\_n: Az állat neve, ami a kijáraton áll, „null” az értéke, ha nem áll senki a kijáraton.
* entrancepair: A kijárat bejáratpárja, ahol bejön majd az orángután.
* nk: A kijárattal szomszédos mezők nevei, szóközzel elválasztva egymástól.

**Wardrobe**

kimenet: Wardrobe: [name], wp: [wpair], a: [animal\_n], n: [[n1] [n2] … [nk]]

* name: A szekrény neve.
* wpair: a szekrény párjának a neve.
* animal\_n: Az állat neve, ami a szekrényen áll, „null” az értéke, ha nem áll senki a szekrényen.
* nk: A szekrénnyel szomszédos mezők nevei, szóközzel elválasztva egymástól.

**Timer**

kimenet: Timer: [name], t: [[t1] [t2] … [tk]]

* name: Az időzítő neve.
* tk: Azok az objektumok, amikre az időzítő hatással van.

**Game**

kimenet: Game: [name], pn: [pandanum], t: [timer], tfs: [timefromstart], ps: [[p1] [p2] … [pk]]

* name: A játék neve.
* pandanum: A játékban szereplő pandák száma.
* timer: Az időzítő neve.
* timefromstart: A játék kezdete óta eltelt idő.
* pk: A játékosok nevei, akik szerepelnek a játékban, szóközzel elválasztva egymástól.

**Player**

kimenet: Player: [name], o: [orangutan], pts: [points]

* name: A játékos neve.
* orangutan: Az orángután neve, amit a játékos irányít.
* points: A játékos pontszáma.

## Összes részletes use-case

[A use-case-eknek a részletezettsége feleljen meg a kezelői felületnek, azaz a felület elemeire kell hivatkozniuk a bemeneti nyelv parancsai alapján.

Alábbi táblázat minden use-case-hez külön-külön.]

|  |  |
| --- | --- |
| **Use-case neve** |  |
| **Rövid leírás** |  |
| **Aktorok** |  |
| **Forgatókönyv** |  |

## Tesztelési terv

[A tesztelési tervben definiálni kell, hogy a be- és kimeneti fájlok egybevetésével miként végezhető el a program tesztelése. Meg kell adni magas szintű teszt forgatókönyveket. Az egyes teszteket elég informálisan, szabad szövegként leírni, tesztesetenként egy-öt mondatban. Minden teszthez meg kell adni, hogy mi a célja, a proto mely funkcionalitását, osztályait stb. teszteli. Az alábbi táblázat minden teszt-esethez külön-külön elkészítendő.]

|  |  |
| --- | --- |
| **Teszt-eset neve** |  |
| **Rövid leírás** |  |
| **Teszt célja** |  |

## Tesztelést támogató segéd- és fordítóprogramok specifikálása

[Rövid bemutatással (elvárt funkcionalitás) specifikálni kell a tesztelést támogató segédprogramokat.]

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2019.03.29  20:30 – 23:30 | 3 óra | Kovács | Diagramok rajzolása, beillesztése. Változások implementálása a kódban. Bemeneti nyelv kezdetleges definiálása. |
| 2019.03.30 10:30 – 12:30 | 2 óra | Kovács | Bemeneti nyelv átdolgozása, befejezése. |