**3. Analízis modell kidolgozása 1.**

66 – otthonmeglefordult

Konzulens:

Szőke Máté

Csapattagok

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

2019.02.25.

Tartalomjegyzék

[3. Analízis modell kidolgozása 4](#_Toc1991777)

[3.1 Objektum katalógus 4](#_Toc1991778)

[3.1.1 Armchair 4](#_Toc1991779)

[3.1.2 ChocolateMachine 4](#_Toc1991780)

[3.1.3 ChocolatePanda 4](#_Toc1991781)

[3.1.4 Entrance 4](#_Toc1991782)

[3.1.5 Exit 4](#_Toc1991783)

[3.1.6 GamblerPanda 4](#_Toc1991784)

[3.1.7 GameMachine 4](#_Toc1991785)

[3.1.8 Hole 4](#_Toc1991786)

[3.1.9 LazyPanda 4](#_Toc1991787)

[3.1.10 Map 4](#_Toc1991788)

[3.1.11 Orangutan 4](#_Toc1991789)

[3.1.12 Panda 5](#_Toc1991790)

[3.1.13 Player 5](#_Toc1991791)

[3.1.14 Tile 5](#_Toc1991792)

[3.1.15 Wardrobe 5](#_Toc1991793)

[3.1.16 WeakTile 5](#_Toc1991794)

[3.2 Statikus struktúra diagramok 6](#_Toc1991795)

[3.3 Osztályok leírása 7](#_Toc1991797)

[*3.3.1* *Animal* 7](#_Toc1991798)

[3.3.2 Armchair 8](#_Toc1991799)

[3.3.3 ChocolateMachine 8](#_Toc1991800)

[3.3.4 ChocolatePanda 9](#_Toc1991801)

[3.3.5 Entrance 9](#_Toc1991802)

[3.3.6 Exit 10](#_Toc1991803)

[*3.3.7* *Field* 11](#_Toc1991804)

[3.3.8 GamblerPanda 11](#_Toc1991805)

[3.3.9 Game 12](#_Toc1991806)

[3.3.10 GameMachine 12](#_Toc1991807)

[3.3.11 Hole 13](#_Toc1991808)

[3.3.12 ITickable 13](#_Toc1991809)

[3.3.13 LazyPanda 14](#_Toc1991810)

[3.3.14 Map 15](#_Toc1991811)

[3.3.15 Orangutan 16](#_Toc1991812)

[3.3.16 Panda 17](#_Toc1991813)

[3.3.17 Player 17](#_Toc1991814)

[3.3.18 Tile 18](#_Toc1991815)

[3.3.19 Timer 18](#_Toc1991816)

[3.3.20 Wardrobe 19](#_Toc1991817)

[3.3.21 WeakTile 19](#_Toc1991818)

[3.4 Szekvencia diagramok 20](#_Toc1991819)

[3.4.1 Animal steppedOn Hole 20](#_Toc1991820)

[3.4.2 ChocolateMachine isBeeping 20](#_Toc1991821)

[3.4.3 ChocolateMachine ticks 21](#_Toc1991822)

[3.4.4 ChocolatePanda isEffectNearby 22](#_Toc1991823)

[3.4.5 ChocolatePanda jumps 23](#_Toc1991824)

[3.4.6 ChocolatePanda steppedOn WeakTile 24](#_Toc1991825)

[3.4.7 Entrance enterField 25](#_Toc1991826)

[3.4.8 GamblerPanda gotScared 26](#_Toc1991827)

[3.4.9 GamblerPanda isEffectNearby 27](#_Toc1991828)

[3.4.10 GamblerPanda steppedOn Tile 28](#_Toc1991829)

[3.4.11 GamblerPanda steppedOn WeakTile 29](#_Toc1991830)

[3.4.12 GameMachine isRinging 30](#_Toc1991831)

[3.4.13 GameMachine ticks 30](#_Toc1991832)

[3.4.14 LazyPanda isEffectNearby 31](#_Toc1991833)

[3.4.15 LazyPanda sitDown 32](#_Toc1991834)

[3.4.16 LazyPanda steppedOn Tile 33](#_Toc1991835)

[3.4.17 LazyPanda steppedOn Wardrobe 34](#_Toc1991836)

[3.4.18 LazyPanda steppedOn WeakTile 35](#_Toc1991837)

[3.4.19 LazyPanda ticks 36](#_Toc1991838)

[3.4.20 Map addPointToPlayer 37](#_Toc1991839)

[3.4.21 Map ticks 38](#_Toc1991840)

[3.4.22 Orangutan dies 39](#_Toc1991841)

[3.4.23 Orangutan HitBy Orangutan 40](#_Toc1991842)

[3.4.24 Orangutan HitBy Panda 40](#_Toc1991843)

[3.4.25 Orangutan pull 41](#_Toc1991844)

[3.4.26 Orangutan removePandaFromLine 42](#_Toc1991845)

[3.4.27 Orangutan steppedOn Armchair 42](#_Toc1991846)

[3.4.28 Orangutan steppedOn ChocolateMachine 43](#_Toc1991847)

[3.4.29 Orangutan steppedOn Entrance 43](#_Toc1991848)

[3.4.30 Orangutan steppedOn Exit 44](#_Toc1991849)

[3.4.31 Orangutan steppedOn GameMachine 45](#_Toc1991850)

[3.4.32 Orangutan steppedOn Tile 46](#_Toc1991851)

[3.4.33 Orangutan steppedOn Wardrobe 47](#_Toc1991852)

[3.4.34 Orangutan steppedOn WeakTile 48](#_Toc1991853)

[3.4.35 Orangutan steps 49](#_Toc1991854)

[3.4.36 Panda dies 50](#_Toc1991855)

[3.4.37 Panda HitBy Orangutan 51](#_Toc1991856)

[3.4.38 Panda HitBy Panda 52](#_Toc1991857)

[3.4.39 Panda steppedOn Armchair 52](#_Toc1991858)

[3.4.40 Panda steppedOn ChocolateMachine 53](#_Toc1991859)

[3.4.41 Panda steppedOn Entrance 53](#_Toc1991860)

[3.4.42 Panda steppedOn Exit 54](#_Toc1991861)

[3.4.43 Panda steppedOn GameMachine 54](#_Toc1991862)

[3.4.44 Panda steppedOn Tile 55](#_Toc1991863)

[3.4.45 Panda steppedOn Wardrobe 56](#_Toc1991864)

[3.4.46 Panda steppedOn WeakTile 57](#_Toc1991865)

[3.4.47 Panda steps 58](#_Toc1991866)

[3.4.48 Panda ticks 59](#_Toc1991867)

[3.4.49 Timer ticks 60](#_Toc1991868)

[3.4.50 Wardrobe canEnter 61](#_Toc1991869)

[3.4.51 WeakTile decreaseLife 62](#_Toc1991870)

[3.5 Napló 63](#_Toc1991871)

# Analízis modell kidolgozása

## Objektum katalógus

### Armchair

A játékban a fotel lesz, egy mezőtípus, erre tud majd leülni a lusta panda. Egyszerre csak egy panda tud rajta ülni, amikor az kipihente magát, feláll a fotelből.

### ChocolateMachine

A csokiautomata egy mezőtípus, a csokipandára van hatással. Időnként sípol egyet, ettől a szomszédos csokipandák megijednek.

### ChocolatePanda

A csokipandák a pandák minden tulajdonságával rendelkeznek, de félnek a csokiautomatától, ha meghallják a sípolást, ugranak egyet. Ha gyenge csempén ugranak, akkor törnek egyet rajta.

### Entrance

A bejáraton egy mezőtípus, ezen jön be az orángután, miután a kijáraton kiment.

### Exit

A kijárat egy mezőtípus, ezen kell kivinni az orángutánnak a pandákat. A kijáraton való áthaladáskor a játékos a pandák száma után pontot kap.

### GamblerPanda

Játékfüggő panda, a pandák minden tulajdonságával rendelkezik, de függőségére való tekintettel fél a játékgép csilingelésétől. Ha ilyet hall, akkor elengedi a mögötte álló pandát.

### GameMachine

Játékgép egy mezőtípus, ami néha csilingel, ezzel hatással van a környező játékfüggő pandákra.

### Hole

A lyuk egy mezőtípus, gyenge csempe helyén lesz, ha az teljesen eltört. Ha egy állat (panda vagy orángután) lyukba lép, meghal.

### LazyPanda

Lusta panda, aki a pandák minden tulajdonságával rendelkezik, de ha fáradt (elfogyott az energiája) és van a szomszédságában fotel, leül rá. Miután ez energiája feltöltődött, feláll, és továbbmegy.

### Map

A játékban résztvevő dolgokat foglalja össze, az időt is kezeli.

### Orangutan

Az orángután a játékos által irányítható karakter, neki kell elkapnia a pandákat. Ehhez egy pandának kell nekimennie, ezáltal befűzi a pandát a vezetett sor elejére.

### Panda

A pandákat kell elkapni az orángutánoknak. A pandának alapvetően semmi különleges tulajdonsága nincs, képesek véletlenszerűen mozogni a mezőkön.

### Player

Egy játékost reprezentál. Nyilvántartja a játékos orángutánját és a pontjait.

### Tile

A csempe egy mezőtípus, a panda és az orángután is ráléphet. Sokszög alakú.

### Wardrobe

A szekrény is egy mezőtípus, átjáróként viselkedik. Egy szekrénybe belépve a párján jövünk ki, ezért a pálya két távolabbi pontját köti össze.

### WeakTile

A gyenge csempe egy csempefajta, amire ha rálépnek csökken az élettartama. Ha teljesen elfogy az „élete” akkor eltörik, és helyén lyuk keletkezik.

## Statikus struktúra diagramok

A privát attribútumokhoz get/set metódusok tartoznak, ezeket az átláthatóság érdekében nem jelöltük.

## 

## Osztályok leírása

### *Animal*

#### Felelősség

Egy általános állat, meg tud halni, lépni tud és más állatok nekimehetnek.

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

* **Field field**: A mező amin az állat áll.
* **double angle**: A szög amerre az állat néz, fokban. A 0 fok, ha az állat felfele néz.
* **Timer timer**: A pályához tartozó időzítő.
* **Map map:** A pályához tartozó map.

#### Metódusok

* **void die()**: Az állat meghal és eltűnik a pályáról.
* **void step(Field to)**: Az állat a megadott mezőre lép.
* **void steppedOn(Tile t)**: Az állat csempére lépett.
* **void steppedOn(WeakTile wt)**: Az állat gyenge csempére lépett.
* **void steppedOn(ChocolateMachine cm)**: Az állat csokiautomatára lépett.
* **void steppedOn(Armchair am)**: Az állat fotelra lépett.
* **void steppedOn(Exit ex)**: Az állat kijáratra lépett.
* **void steppedOn(GameMachine gm)**: Az állat játékautomatára lépett.
* **void steppedOn(Entrance ent)**: Az állat bejáratra lépett.
* **void steppedOn(Wardrobe w)**: Az állat szekrényre lépett.
* **void HitBy(Orangutan o)**: Az állattal ütközik egy orángután.
* **void HitBy(Panda p)**: Az állattal ütközik egy panda.

### Armchair

#### Felelősség

A játékban a fotel lesz, egy mezőtípus, erre tud majd leülni a lusta panda. Egyszerre csak egy panda tud rajta ülni, amikor az kipihente magát, feláll a fotelből.

#### Ősosztályok

Field

#### Interfészek

ITickable

* **Attribútumok**

**-**

#### Metódusok

* **void accept(Animal a):** A fotel nem fogad egyetlen állatot sem, a metódus azonnal visszatér.

### ChocolateMachine

#### Felelősség

A csokiautomata egy mezőtípus, a csokipandára van hatással. Időnként sípol egyet (bizonyos tick-ekre), ettől a szomszédos csokipandák megijednek.

#### Ősosztályok

Field

#### Interfészek

ITickable

#### Attribútumok

* **WeakTile[1..\*] weakTiles:** A csokiautomata körül lévő gyenge csempéket tároló tömb
* **boolean beeping:** Attól függően igaz vagy hamis az értéke, hogy sípol-e az automata

#### Metódusok

* **boolean isBeeping()**: Lekérdező metódus a beeping attribútumhoz
* **void removeWeakTile(WeakTile wt):** Eltávolít egy gyenge csempét.
* **void accept(Animal a):** A csokiautomata nem fogad egyetlen állatot sem, a metódus azonnal visszatér.
* **void tick():** Véletlenszerűen vagy állít a beeping értékén, vagy nem.

### ChocolatePanda

#### Felelősség

Ha meghallja a csokiautomata sípolását ugrik egyet.

#### Ősosztályok

Animal → Panda

#### Interfészek

ITickable

#### Attribútumok

* **ChocolateMachine[0..\*] chocolateMachines:** A pályán lévő csokiautomaták.

#### Metódusok

* **void jump()**: A panda ugrik egyet, ezzel csökkentve az alatta levő csempe életét eggyel.
* **boolean isEffectNearby()**: Attól függően ad vissza igaz/hamis értéket, hogy van-e a panda közelében sípoló automata (összehasonlítja a tárolt automatákat a szomszédos mezőkkel).

### Entrance

#### Felelősség

A bejárat egy mezőtípus. Ezen jön be az orángután, miután a kijáraton kiment.

#### Ősosztályok

Field

* **Interfészek**

**-**

* **Attribútumok**

**-**

#### Metódusok

* **void accept(Animal a):** A bejárat nem fogad egyetlen állatot sem, a metódus azonnal visszatér.
* **void enterToField(Orangutan o):** Elhelyezi az orángutánt az adott mezőn (az orángután fieldje az entrance obejktum lesz, az entrance animalja pedig az orángután).

### Exit

#### Felelősség

A kijárat egy mezőtípus, ezen kell kivinni az orángutánnak a pandákat. A kijáraton való áthaladáskor a játékos a pandák száma után pontot kap. A kivitt pandák időzítettségének megvonása az exit felelőssége.

#### Ősosztályok

Field

* **Interfészek**

**-**

#### Attribútumok

* **Entrance entrancePair:** A kijárat ebben az attribútumban tárolja a hozzá tartozó bejáratot, ahol a kijáraton átlépő orángutánnak be kell majd jönnie.
* **Map map:** Az adott játékhoz tartozó map. Az exit ezen keresztül éri el a játékos pontozását.
* **Timer timer:** A játékhoz tartozó időzítő objektum.

#### Metódusok

* **void accept(Animal a):** Meghívja az állat megfelelő steppedOn() metódusát, önmagát adva paraméterként.
* **void orangutanExited(Orangutan o)**: Lekérdezi a kilépő orángután pandáinak listáját. A pandák számának megfelelően pontozza az orángutánt irányító játékost. A kivitt pandákat leveszi a Timer listájáról és leveszi őket a hozzájuk tartozó mezőkről (null-ra állítja a mezők animal referenciáját).

### *Field*

#### Felelősség

Absztrakt ősosztálya a különböző mezőtípusoknak, egy pályabeli mezőt reprezentál. Állatok léphetnek rá és el is tárolja, hogy mely állat tartózkodik rajta. Ismeri a szomszédos mezőket is.

#### Ősosztályok

-

* **Interfészek**

**-**

#### Attribútumok

* **Field[1..\*] neighbours**: a mező szomszédai
* **Animal a**: a mezőn tartózkodó állat. Ha nincs állat a mezőn, az értéke null.

#### Metódusok

* **Field getNeighbourByAngle(double angle)**: Visszaadja a szomszédjai közül azt a mezőt, amely a paraméterben átadott szöghöz tartozik
* **void accept(Animal a):** Fogadja a paraméterben átadott állatot. Hívja az állat megfelelő (a mező típusától függő) metódusát.

### GamblerPanda

#### Felelősség

Ha meghallja a játékgép csilingelését, megijed.

#### Ősosztályok

Animal → Panda

#### Interfészek

ITickable

#### Attribútumok

* **GameMachine[0..\*] gameMachines**: A pályán lévő játékgépek.

#### Metódusok

* **void gotScared()**: A panda megijed, emiatt elengedi a többi panda kezét.
* **boolean isEffectNearby()**: Attól függően ad vissza igaz/hamis értéket, hogy van-e a panda közelében csilingelő automata (összehasonlítja a tárolt automatákat a szomszédos mezőkkel).

### Game

#### Felelősség

Tárolja a pályát, elindítja és leállítja a játékot.

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

* **map: Map**: A játék pályája.

#### Metódusok

* **void start()**: Elindítja a játékot.
* **void endGame()**: Leállítja a játékot.

### GameMachine

#### Felelősség

Játékgép egy mezőtípus, ami néha csilingel (bizonyos tick-ekre), ezzel hatással van a környező játékfüggő pandákra.

#### Ősosztályok

Field

#### Interfészek

ITickable

#### Attribútumok

* **boolean ringing:** Attól függően igaz vagy hamis az értéke, hogy sípol-e az automata.

#### Metódusok

* **void accept(Animal a):** A játékgép nem fogad egyetlen állatot sem, a metódus azonnal visszatér.
* **boolean isRinging():** A ringing attribútum értékének lekérdezésére szolgáló metódus.
* **void tick():** Véletlenszerűen vagy állít a ringing értékén, vagy nem.

### Hole

#### Felelősség

A lyuk egy mezőtípus, gyenge csempe helyén lesz, ha az teljesen eltört. Ha egy állat (panda vagy orángután) lyukba lép, meghal.

#### Ősosztályok

Field

#### Interfészek

-

* **Attribútumok**

**-**

#### Metódusok

* **void accept(Animal a):** Meghívja az állat die() metódusát.

### ITickable

#### Felelősség

Az ITickable egy interfész, amelyet minden olyan osztály megvalósít, amelynek bizonyos (véletlenszerű vagy meghatározott) időközönként kell csinálnia valamit.

#### Ősosztályok

-

#### Metódusok

**void tick():** A bizonyos időközönként elvégzendő dolog.

### LazyPanda

#### Felelősség

Ha elfogy az energiája és van szomszédos mezőn fotel, akkor leül.

#### Ősosztályok

Animal → Panda

#### Interfészek

ITickable

#### Attribútumok

* **Armchair[0..\*] armchairs**: A pályán lévő fotelek.
* **int energy**: A panda energiája. Az értéke minimum 0, maximum 2.

#### Metódusok

* **void sitDown(Armchair ac)**: A lusta panda elengedi a többi panda kezét és leül a fotelbe.
* **Armchair isEffectNearby()**: Visszaad egy fotelt, ami a lusta panda közelében van (összehasonlítja a tárolt foteleket a szomszédos mezőkkel). Null-t ad vissza, ha nincs egyezés.

### Map

#### Felelősség

Tárolja a pályán lévő mezőket, játékosokat, pandák számát, a pályához tartozó időt és időzítőt. Pontot oszt ki a játékosoknak.

#### Ősosztályok

-

#### Interfészek

ITickable

#### Attribútumok

* **Player[1..2] players**: A pályához tartozó játékosok.
* **Field[1..\*] fields:** A pályát alkotó mezők.
* **int timeFromStart**: A pálya kezdetétől eltelt idő.
* **Timer timer**: A pályához tartozó időzítő.
* **int pandaNum**: A pályán tartózkodó pandák száma.

#### Metódusok

* **void tick()**: Meghívódik az increaseTime().
* **void addPointToPlayer(int point, Orangutan o)**: Az orángutánhoz tartozó játékos pontszámát növeli.
* **void decreasePandaNum(int i)**: Csökkenti a pandák eltárolt számát 1-gyel.
* **void increaseTime()**: Növeli az eltelt időt 1-gyel.

### Orangutan

#### Felelősség

A hozzá tartozó játékos inputjára mozog a pályán, pandákat gyűjtöget.

#### Ősosztályok

Animal

#### Interfészek

-

#### Attribútumok

* **Panda[0..\*] pandas**: A pandák, amiket láncként húz maga után.
* **Field prevField**: Az a mező, ahol az orángután a legutolsó lépése előtt állt.

#### Metódusok

* **void removePandaFromLine(Panda p)**: Eltávolít egy pandát a láncból.
* **void addPanda(Panda p)**: Hozzáadja a pandát a lánchoz.
* **void pull(Field f):** Az orángután húzza maga után a láncában lévő többi pandát. A paraméterként kapott mező az orángután lépés előtti mezője, ide fog lépni a sorban lévő első panda.
* **int getIndexOf(Panda p)**: Visszaadja egy panda indexét a láncból.
* **void step(Field to)**: Beállítja a prevField-et, majd lép a paraméterben átadott mezőre. Meghívja a pull függvényt, paraméterként a prevField-et adva.
* **void HitBy(Orangutan o)**: Az orángután nem ütközhet orángutánnal, azonnal visszatér.
* **void HitBy(Panda p)**:A panda nem mehet neki orángutánnak, azonnal visszatér.

### Panda

#### Felelősség

Egy általános panda. Véletlenszerűen mozog a pályán egészen addig, amíg be nem fűzi a láncába egy orángután vagy meg nem hal.

#### Ősosztályok

Animal

#### Interfészek

ITickable

#### Attribútumok

* **Orangutan leader**: Az orángután, amelyik láncában a panda benne van.

#### Metódusok

* **void tick()**: Ha nincs orángutánja, véletlenszerűen lép egy szomszédos mezőre.
* **void HitBy(Orangutan o)**: Ha a panda hozzá volt már csatolva egy lánchoz, akkor abból a láncból eltávolítja. Hozzáadja az ütköző orángután láncának elejére, eközben helyet cserél az orángutánnal.
* **void HitBy(Panda p)**:A panda nem mehet neki az orángutánnak, azonnal visszatér.
* **void tick()**:A szomszédos mezők közül véletlenszerűen kiválaszt egyet és oda lépteti a pandát.

### Player

#### Felelősség

Tárolja a játékos pontját.

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

* **int points**: A játékos az adott pályán megszerzett pontja.
* **Orangutan orangutan**: A játékos orángutánja.

#### Metódusok

* **void addPoints(int p)**: Növeli a játékos pontszámát.

### Tile

#### Felelősség

Ráléphet egy állat.

#### Ősosztályok

Field

#### Interfészek

-

#### Attribútumok

-

#### Metódusok

* **void accept(Animal a)**: Kezeli az állatot, aki a csempére szeretne lépni.

### Timer

#### Felelősség

A Timer osztály felel a játék időzítettségéért. Összegyűjti az összes ITickable interfészt megvalósító objektumot és folytonosan hívja az ő tick() metódusaikat a játék kezdetétől a játék végéig.

#### Ősosztályok

Field

* **Interfészek**

**-**

#### Attribútumok

* **ITickable[] tickables:** Az összegyűjtött ITickable-ök tömbje.

#### Metódusok

* **void addTickable(ITickable t):** Hozzáad egy ITickable-t a gyűjteményhez.
* **void removeTickable(ITickable t):** Kivesz egy bizonyos ITickable-t a gyűjteményből.

### Wardrobe

#### Felelősség

A szekrény is egy mezőtípus, átjáróként viselkedik. Egy szekrénybe belépve a párján jövünk ki, ezért a pálya két távolabbi pontját köti össze.

#### Ősosztályok

Field

* **Interfészek**

**-**

#### Attribútumok

* + **Wardrobe pair:** A szekrény ebben az attribútumban tárolja a párját.

#### Metódusok

* **void accept(Animal a):** Meghívja az állat megfelelő steppedOn() metódusát, önmagát adva paraméterként
* **void teleport(Orangutan o):** Áthelyezi az orángutánt a párjára. Az orángután előző mezőjére húzza az orángután pandáit.
* **void teleport(Panda p):** Ha a pandának nincs vezetője (nincs láncba fűzve), akkor áthelyezi a pandát a párjára.
* **boolean canEnter():** Megvizsgálja, hogy a párján tartózkodik-e állat.

### WeakTile

#### Felelősség

Ha rálépnek, csökken az élete. Ha az élete 0-ra csökken, eltörik.

#### Ősosztályok

Field → Tile

#### Interfészek

-

#### Attribútumok

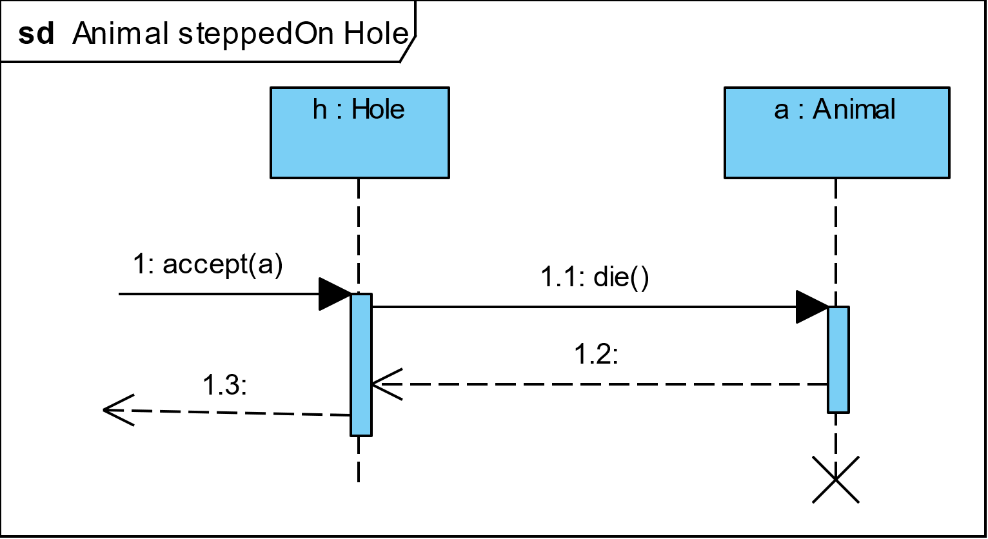
* **int life**: A gyenge csempe életereje.
* **ChocolateMachine[0..\*] cms:** A pályán lévő csokiautomaták.
* **Map map:** A játékhoz tartozó map.

#### Metódusok

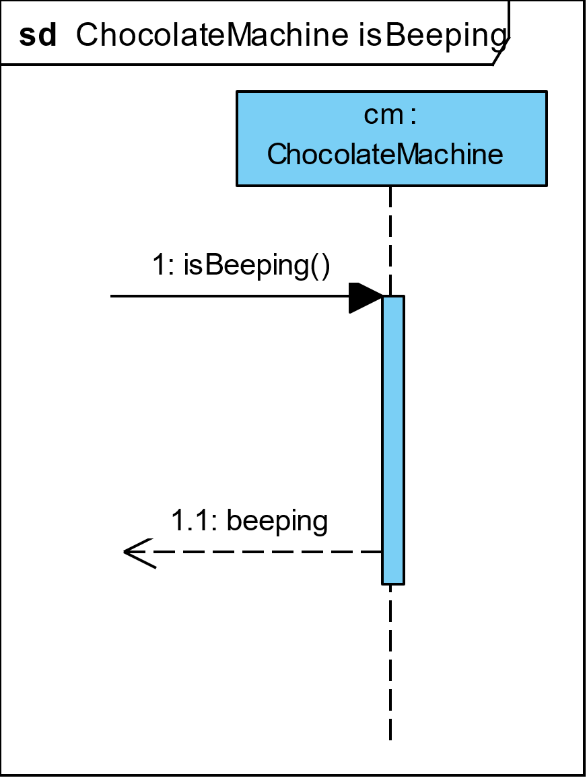
* **void decreaseLife()**: Csökkenti a csempe életerejét 1-gyel.
* **void break():** Eltörik a csempe és egy lyuk lesz a helyén. Kezeli a csokiautomaták által tárol weakTile tömböket is (törli magát belőlük). A szomszédos mezőkben átállítja az eddigi gyenge csempét lyukra.

## Szekvencia diagramok

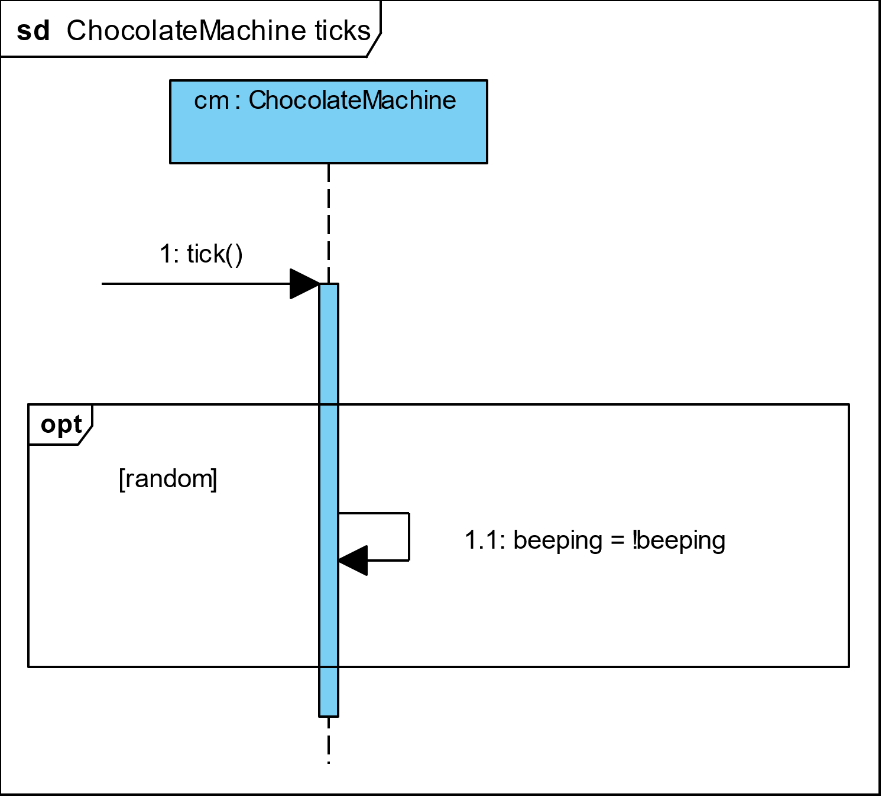
### Animal steppedOn Hole



### ChocolateMachine isBeeping



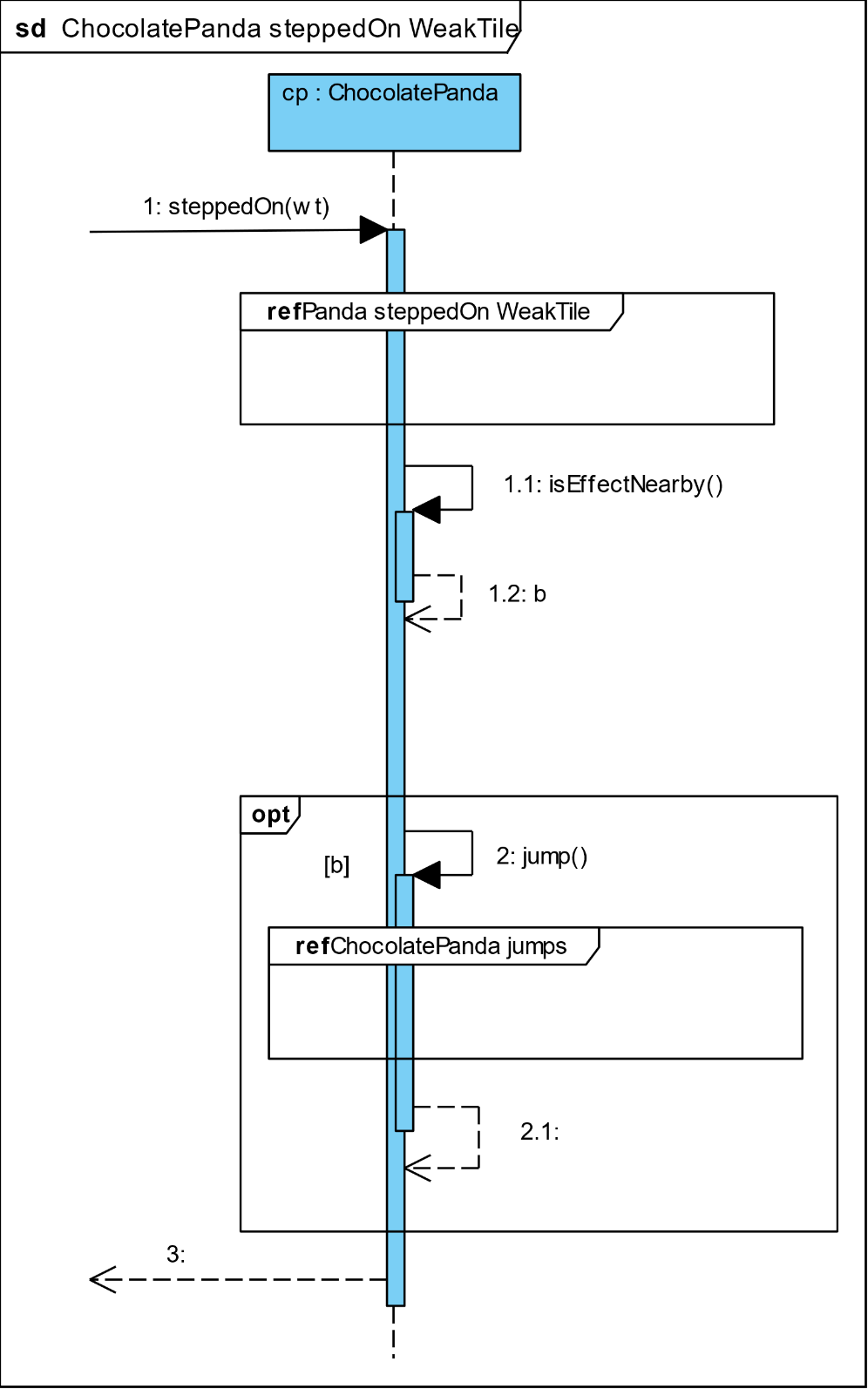
### ChocolateMachine ticks



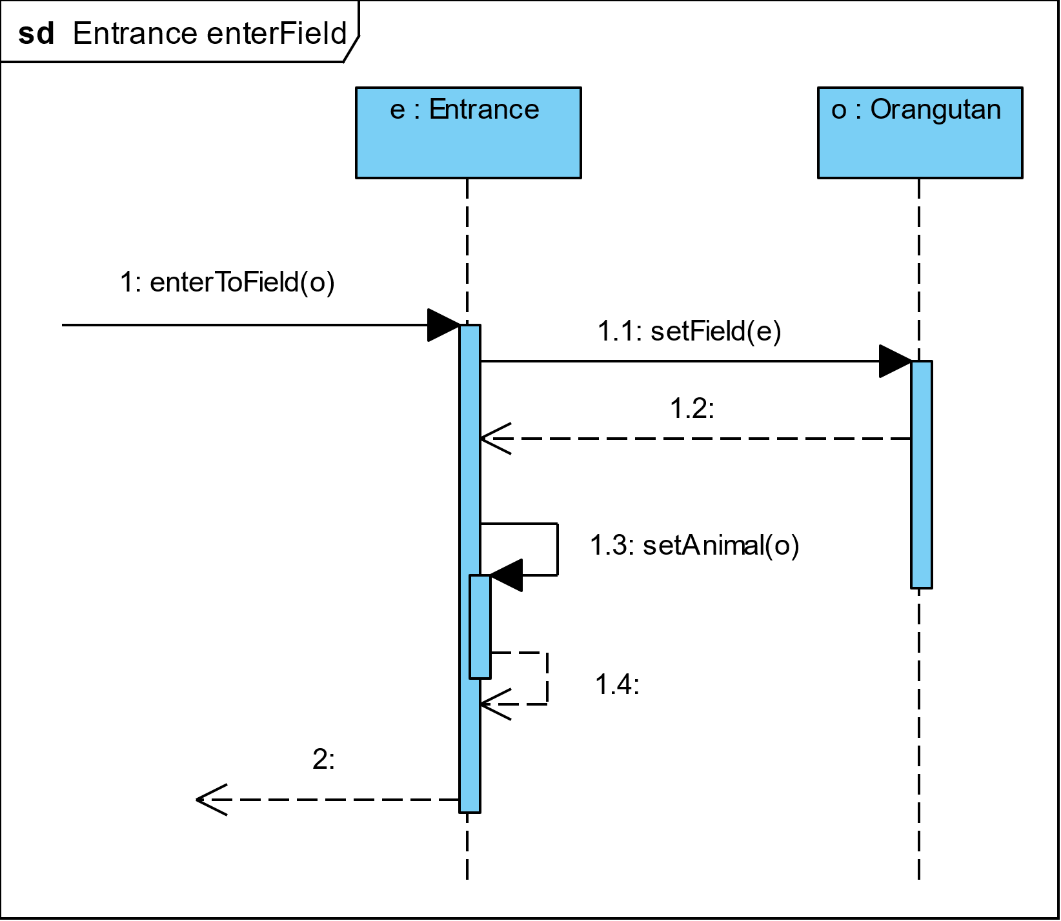
### ChocolatePanda isEffectNearby

### ChocolatePanda jumps

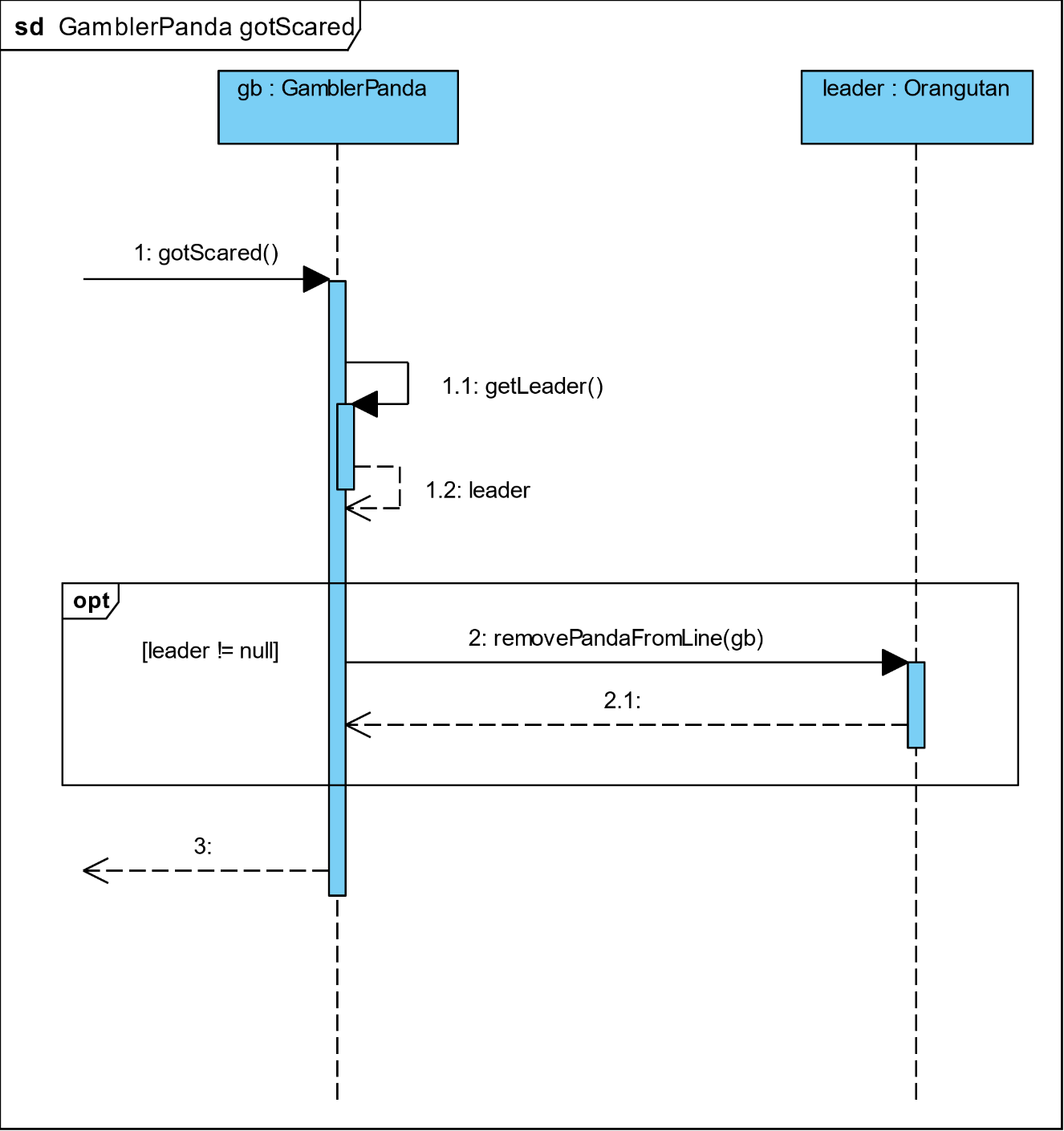
### ChocolatePanda steppedOn WeakTile



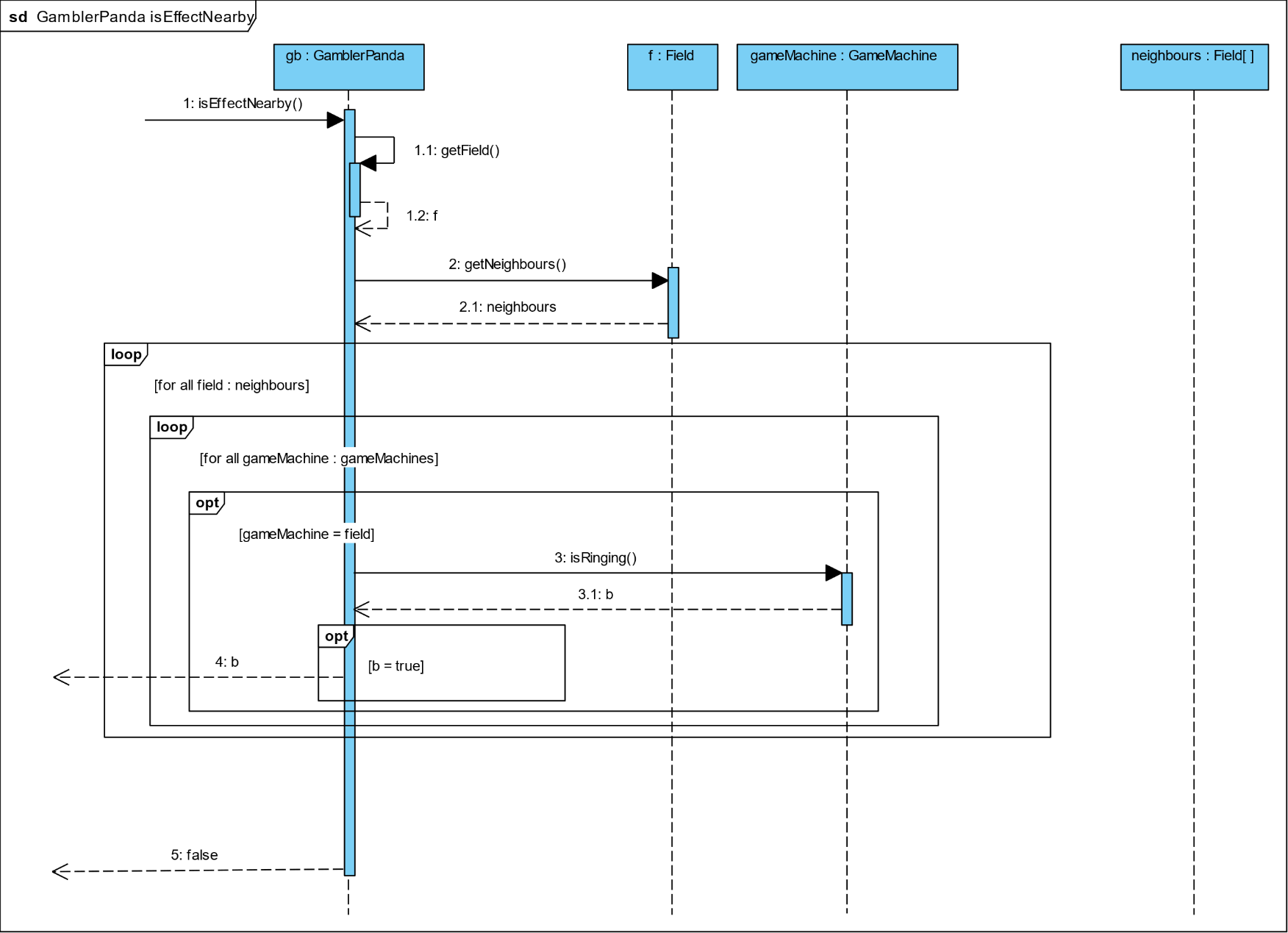
### Entrance enterField



### GamblerPanda gotScared



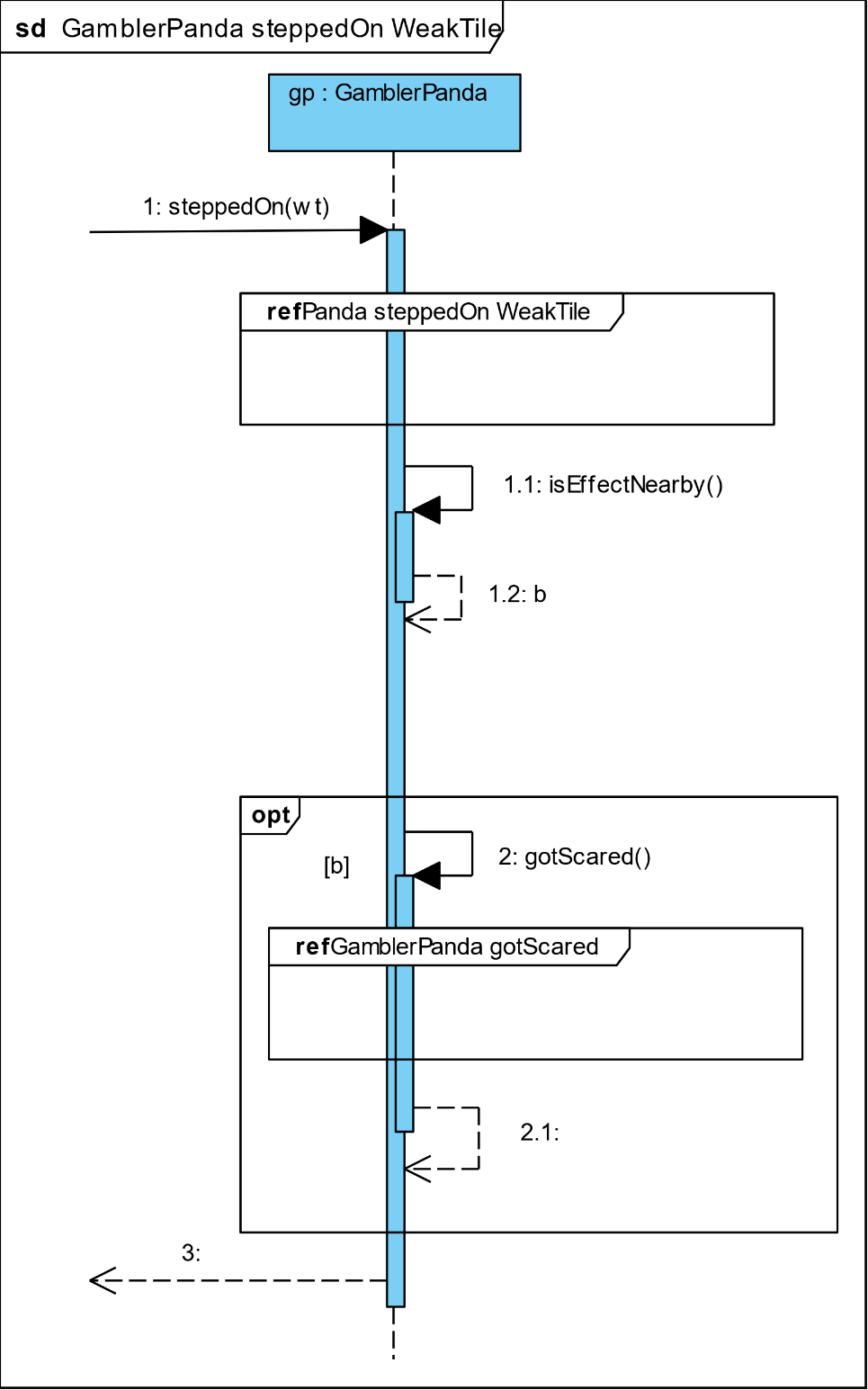
### GamblerPanda isEffectNearby



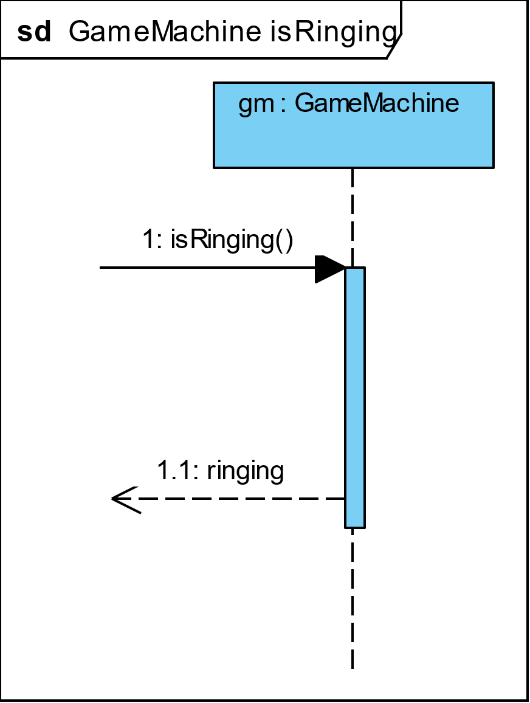
### GamblerPanda steppedOn Tile



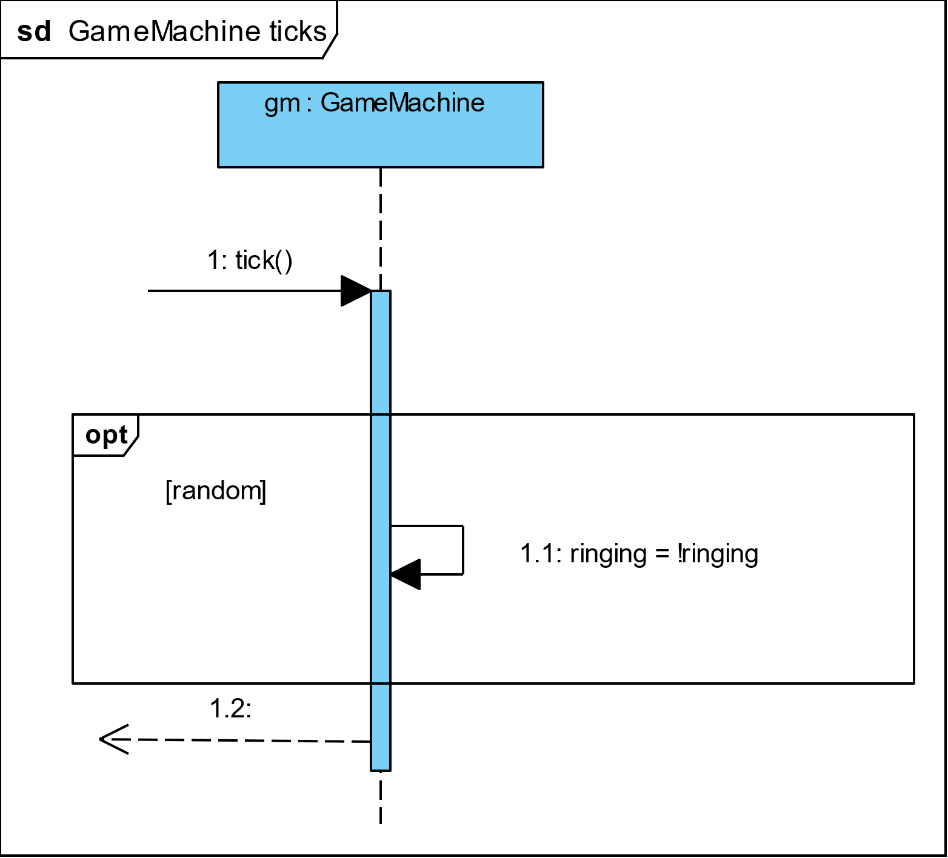
### GamblerPanda steppedOn WeakTile



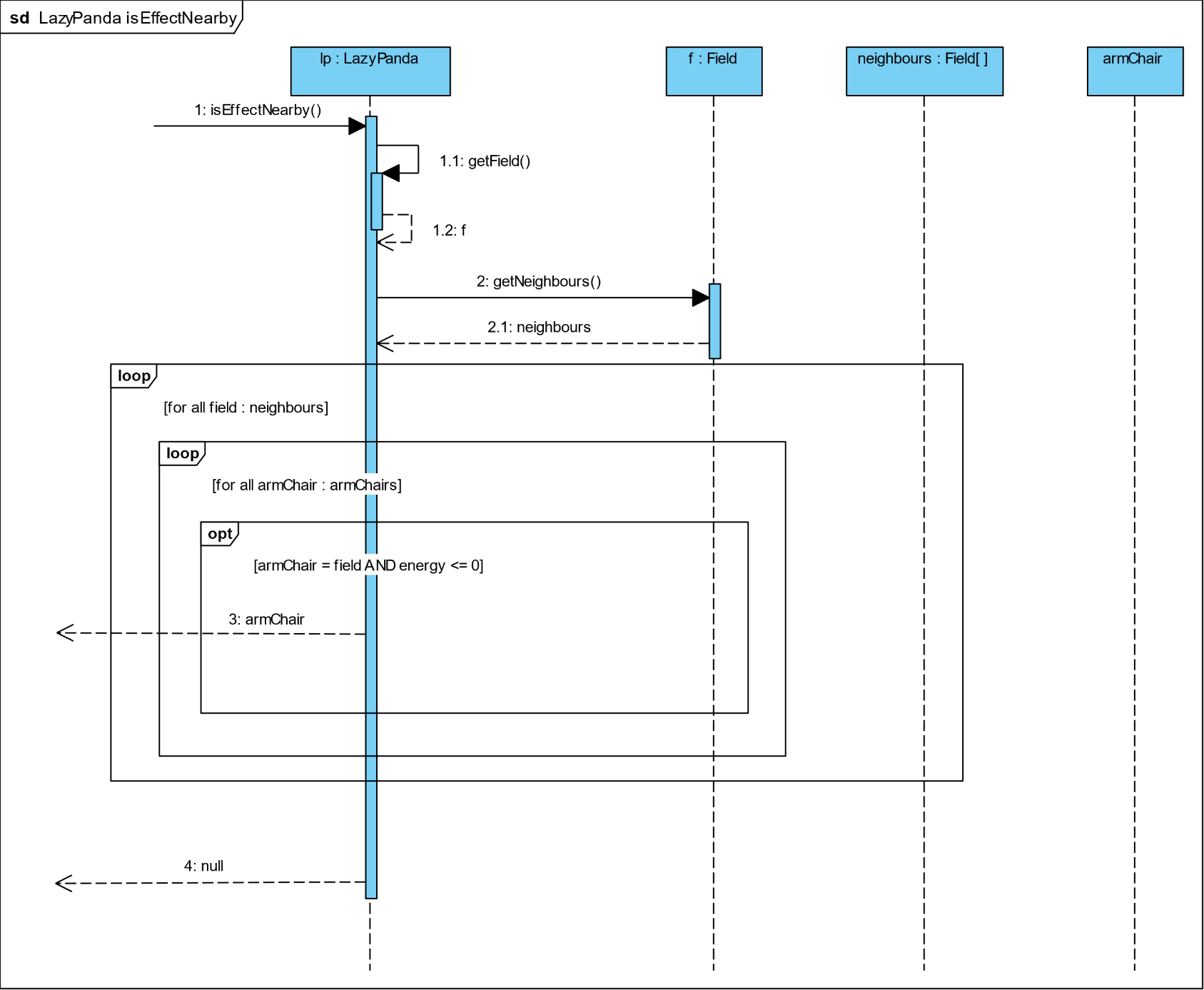
### GameMachine isRinging



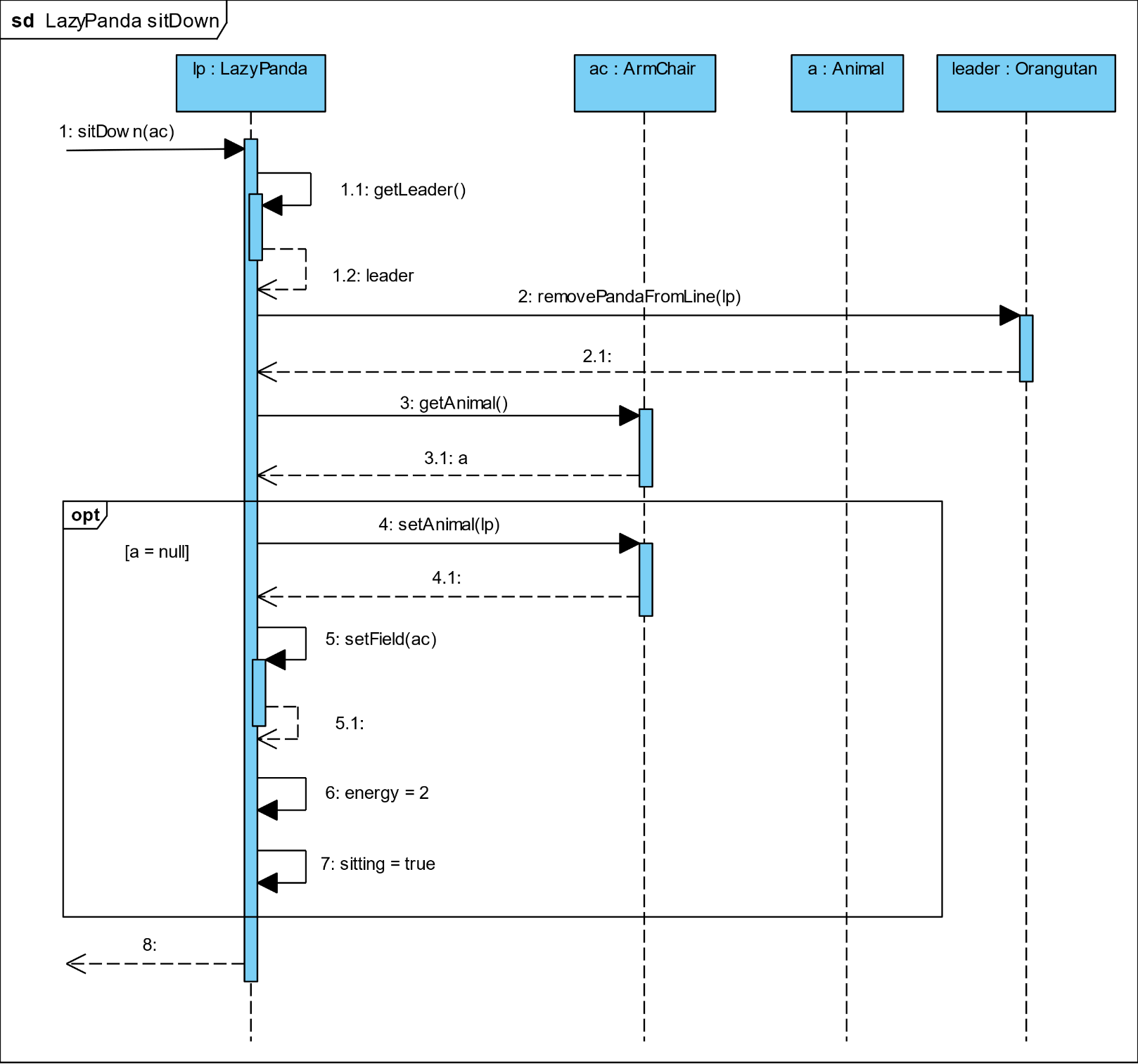
### GameMachine ticks



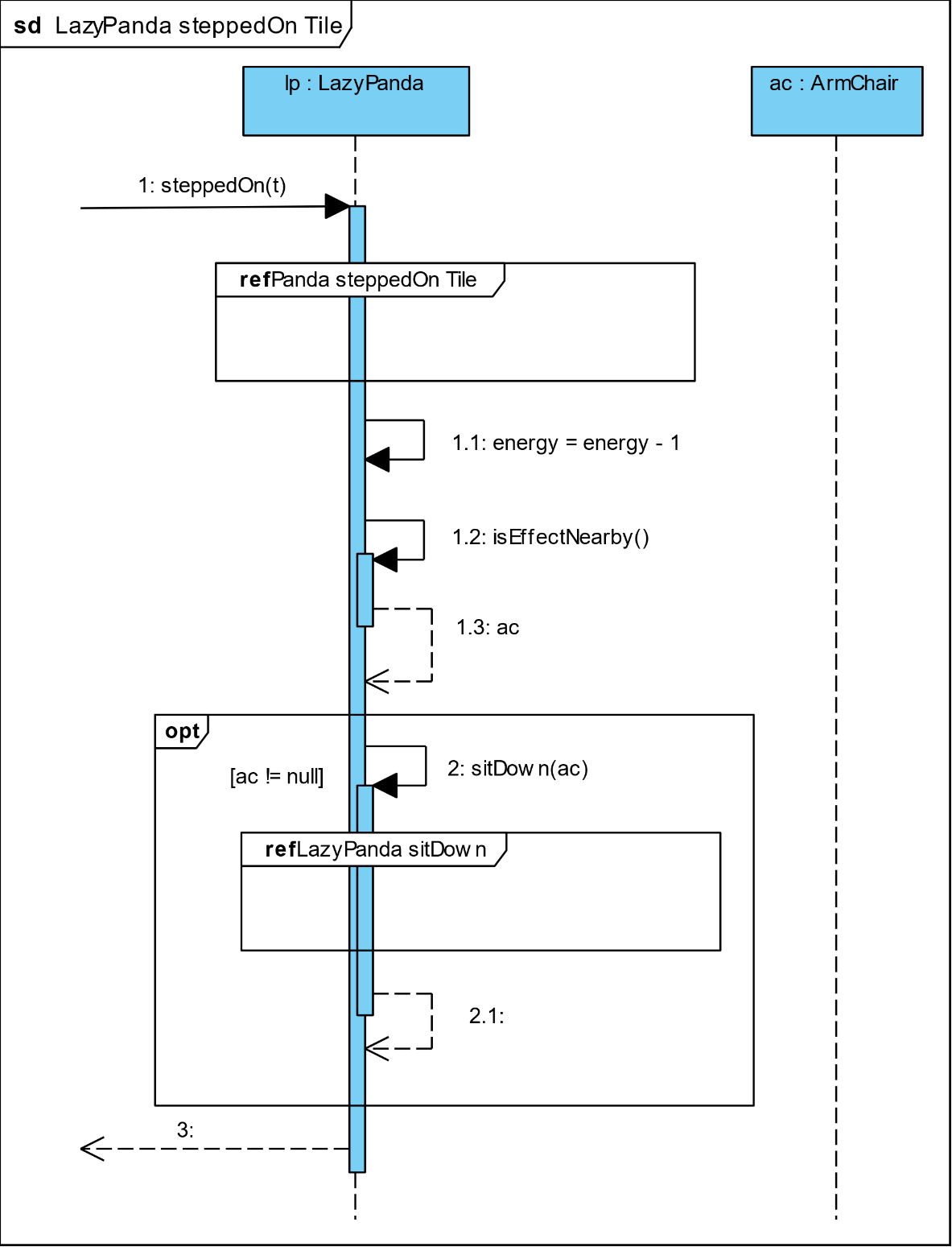
### LazyPanda isEffectNearby



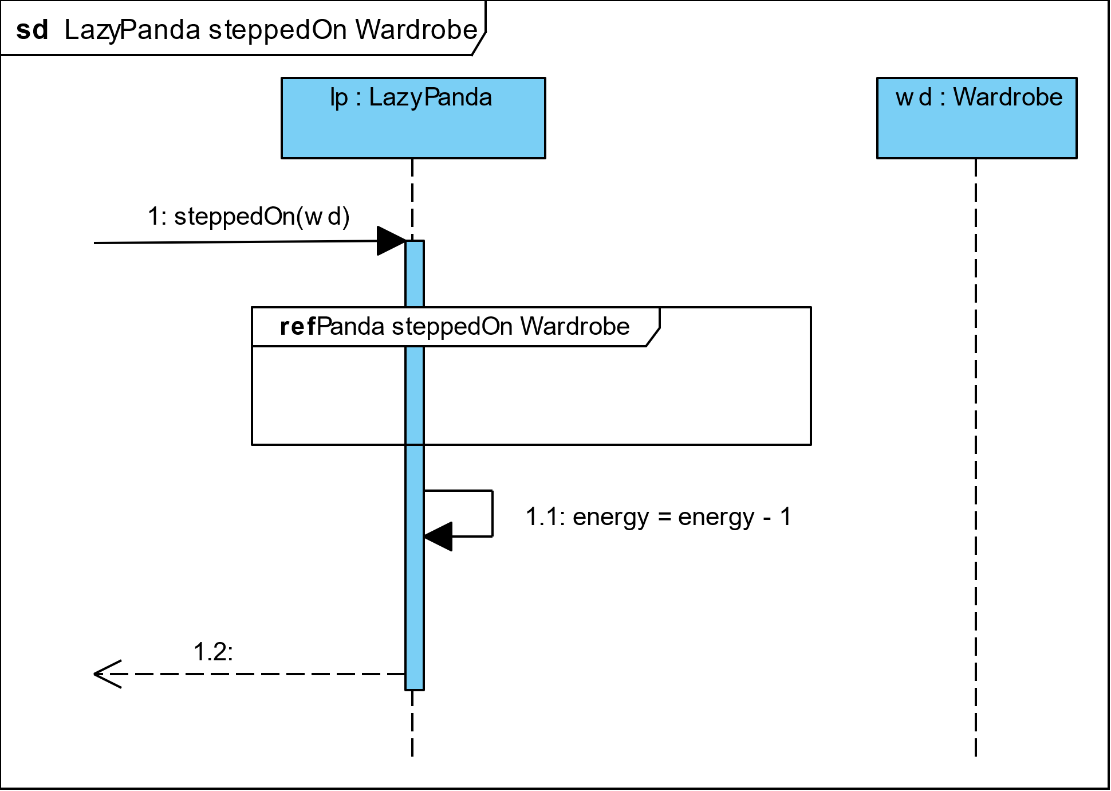
### LazyPanda sitDown



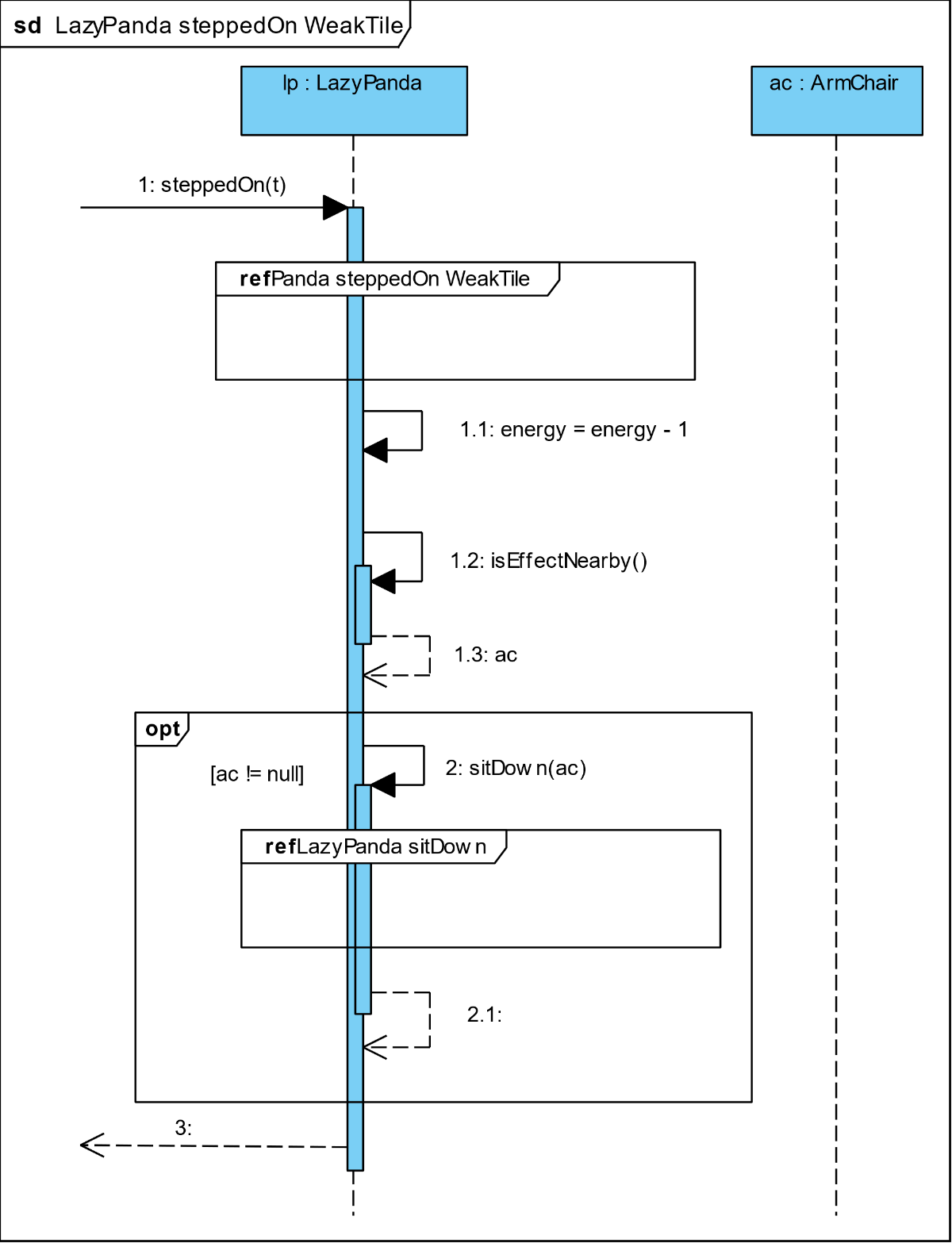
### LazyPanda steppedOn Tile



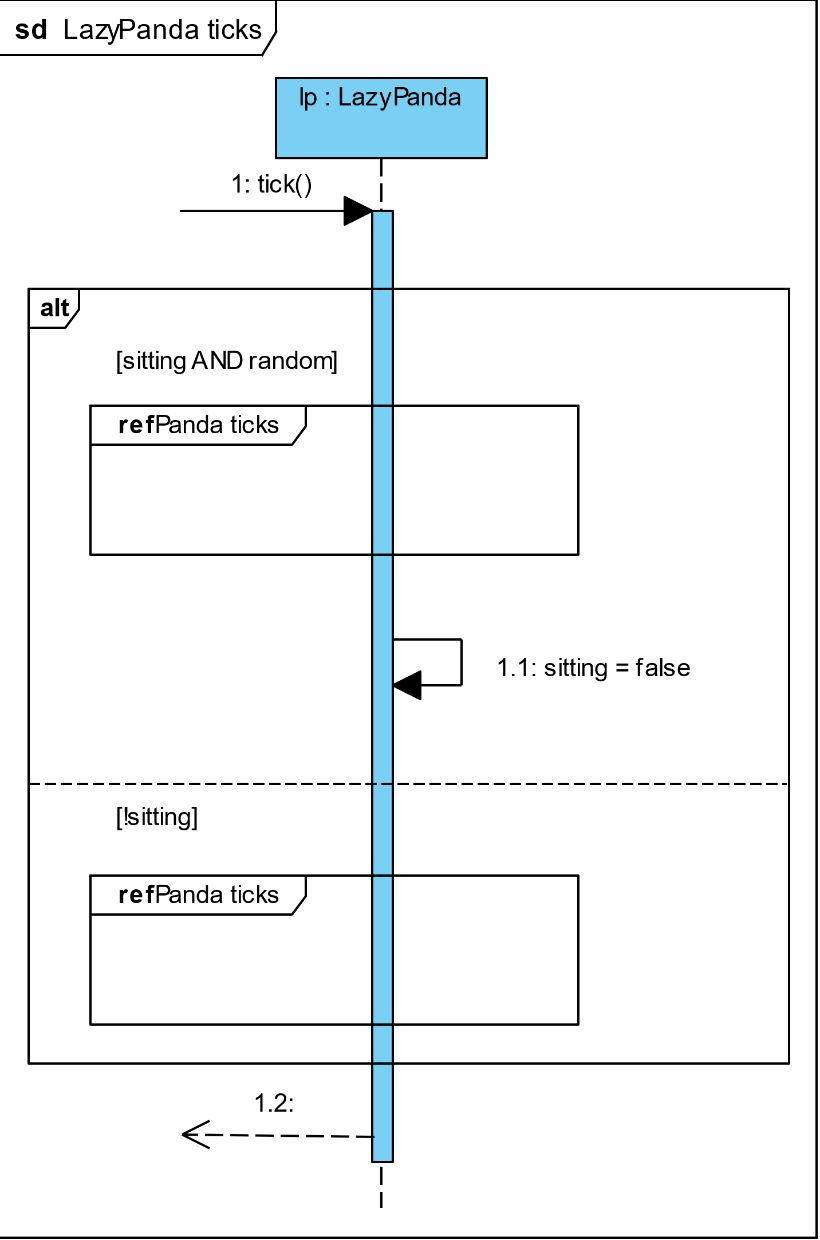
### LazyPanda steppedOn Wardrobe



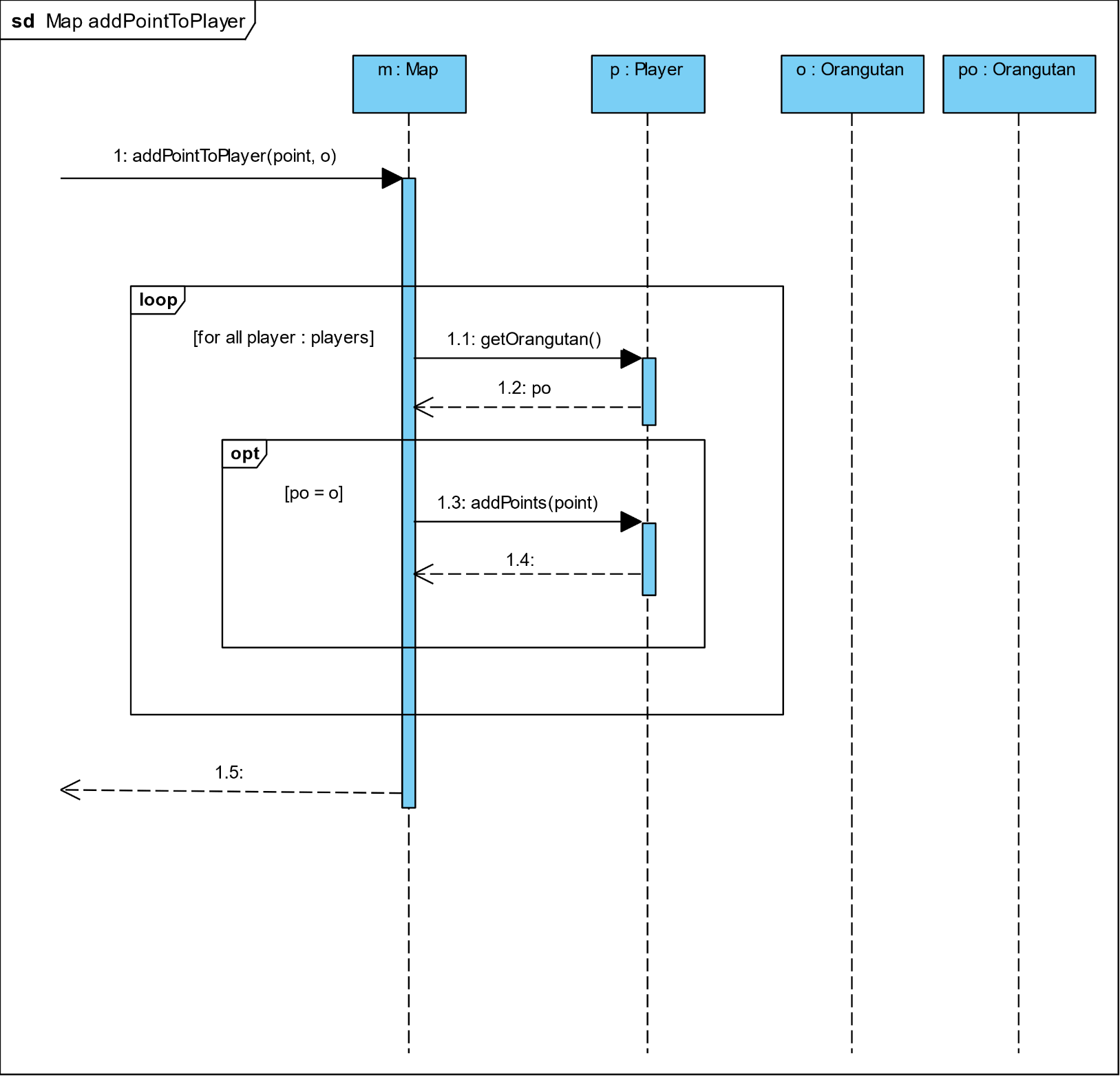
### LazyPanda steppedOn WeakTile



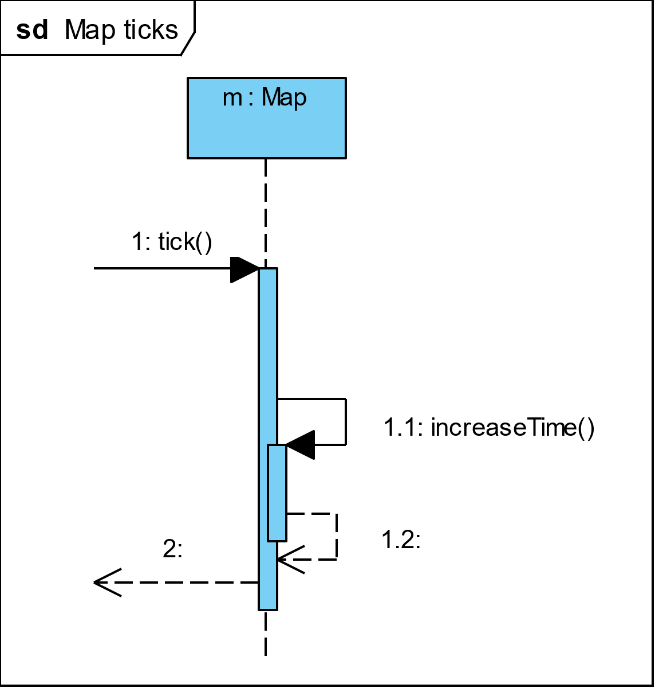
### LazyPanda ticks



### Map addPointToPlayer

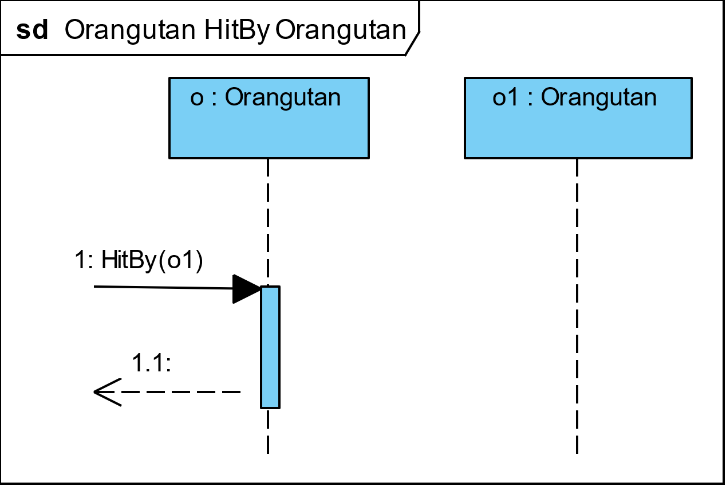


### Map ticks

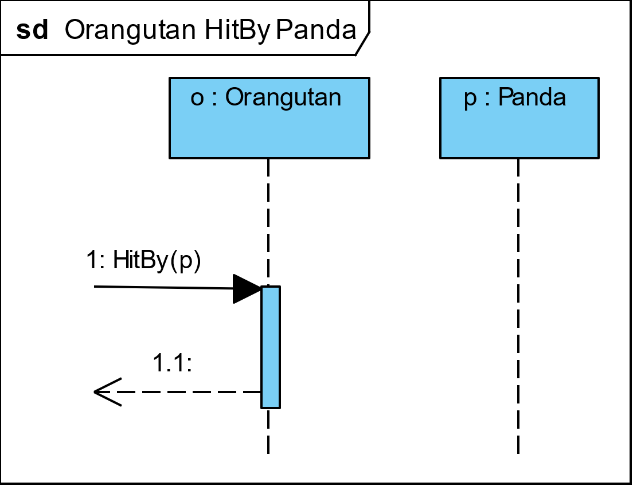


### Orangutan dies

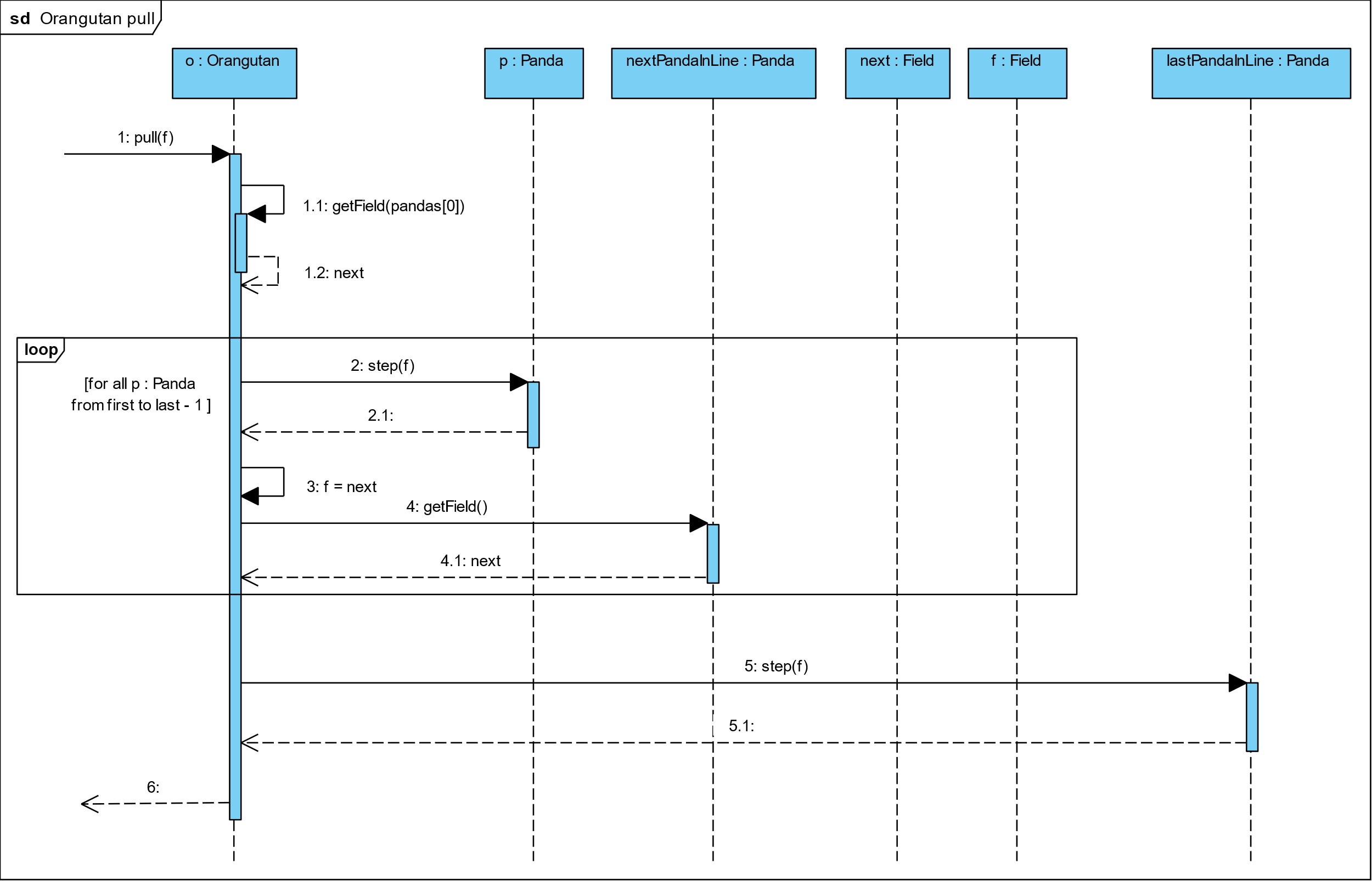
### Orangutan HitBy Orangutan



### Orangutan HitBy Panda

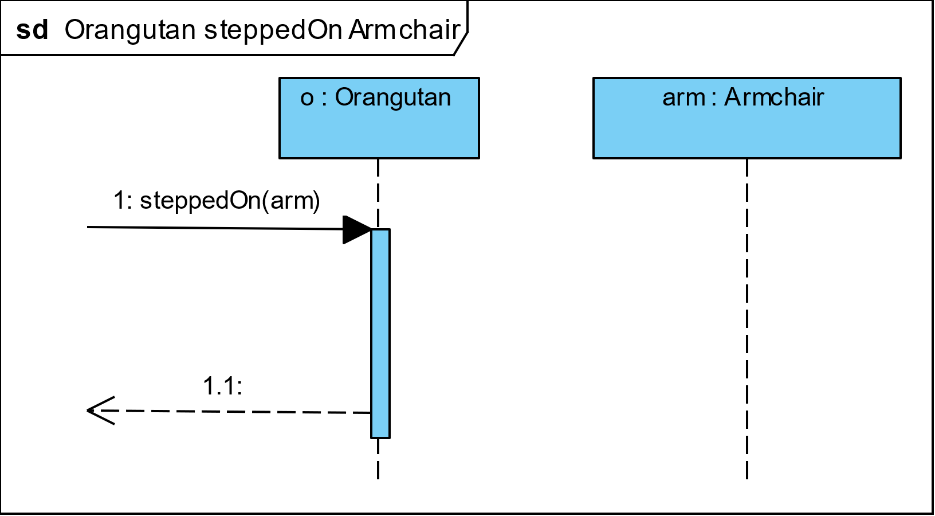


### Orangutan pull

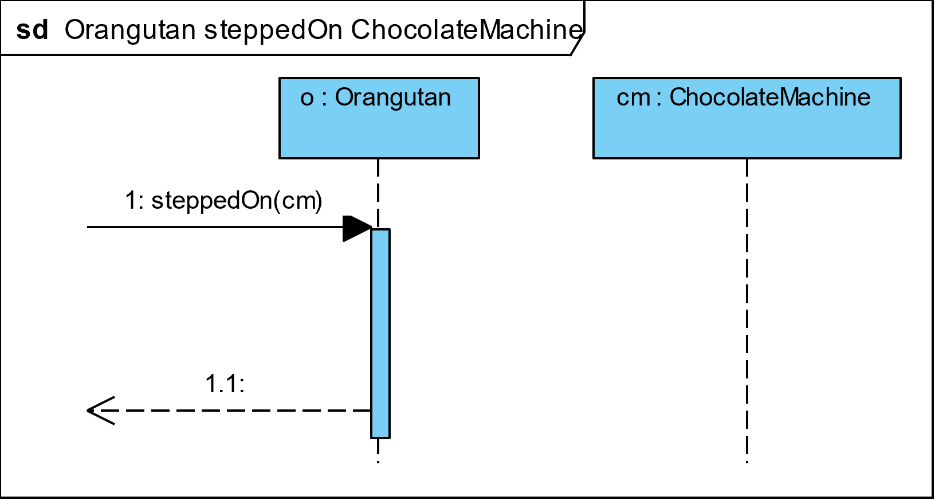


### Orangutan removePandaFromLine

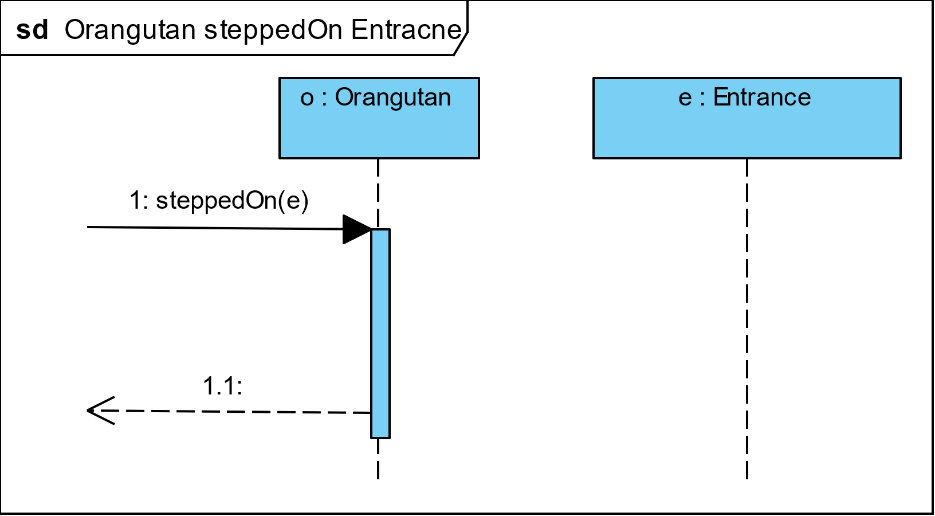
### Orangutan steppedOn Armchair



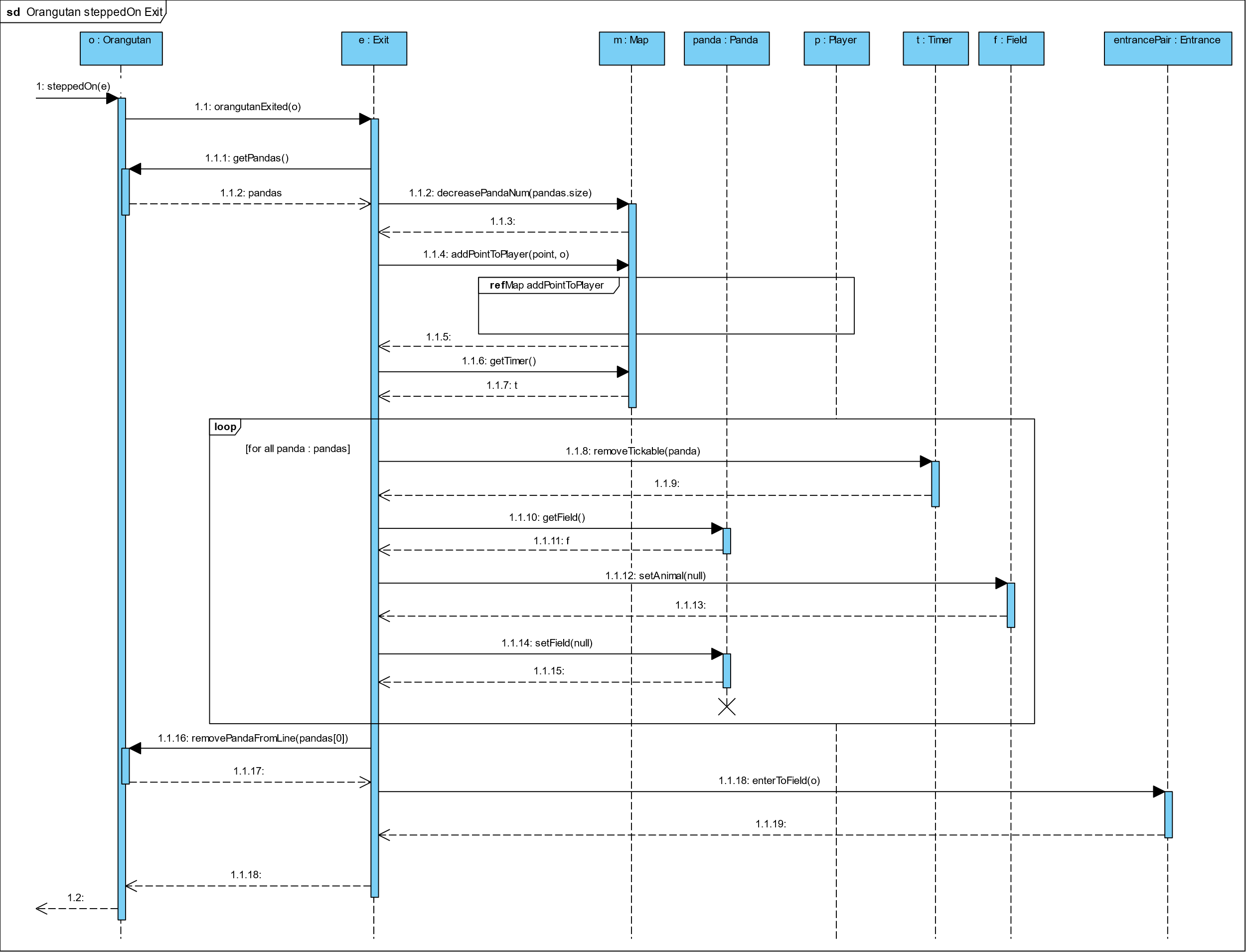
### Orangutan steppedOn ChocolateMachine



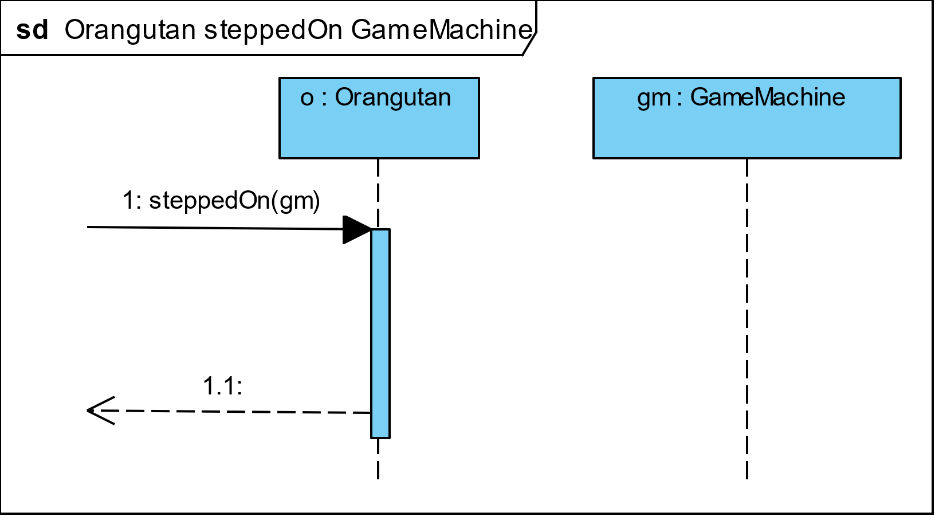
### Orangutan steppedOn Entrance



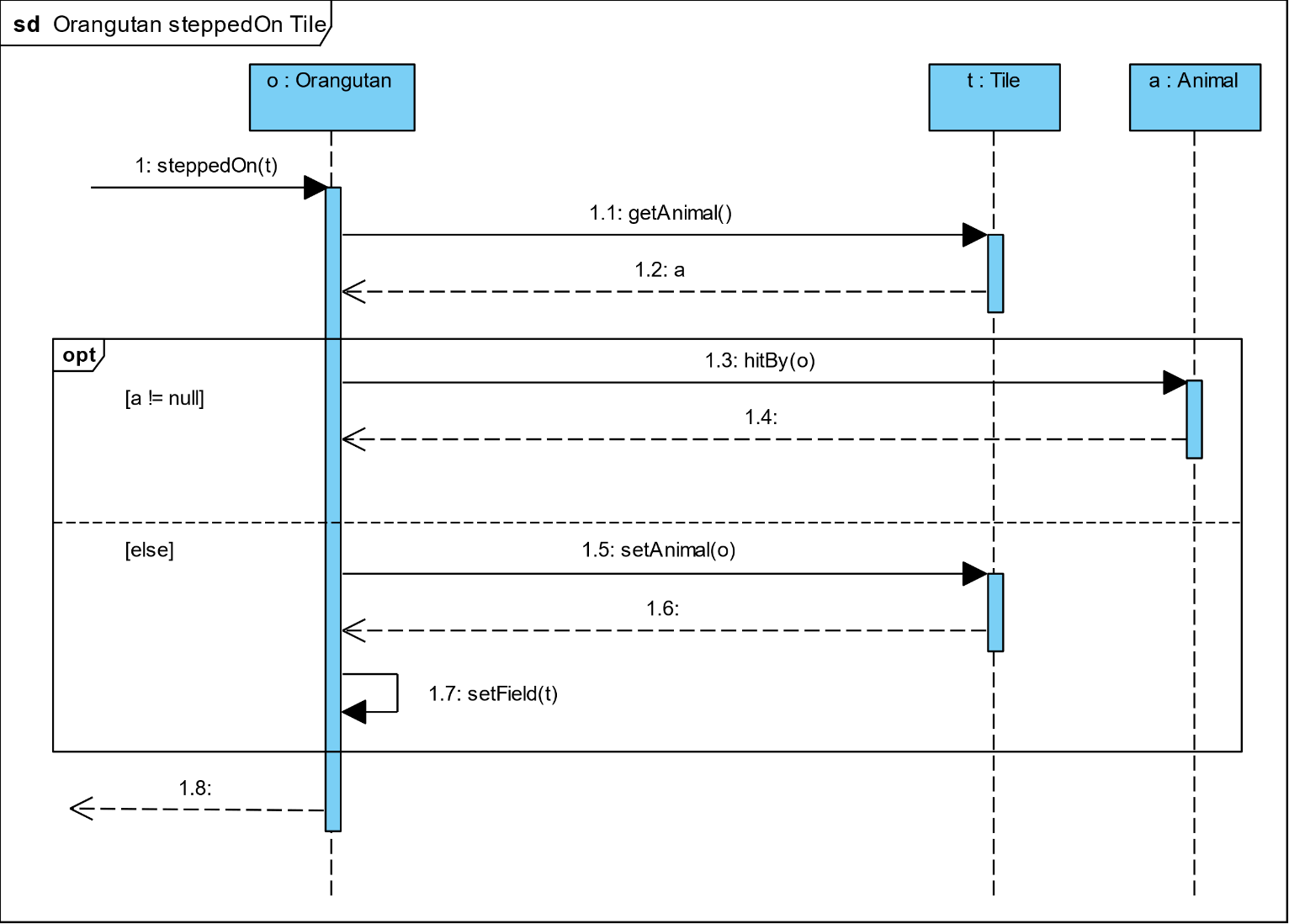
### Orangutan steppedOn Exit



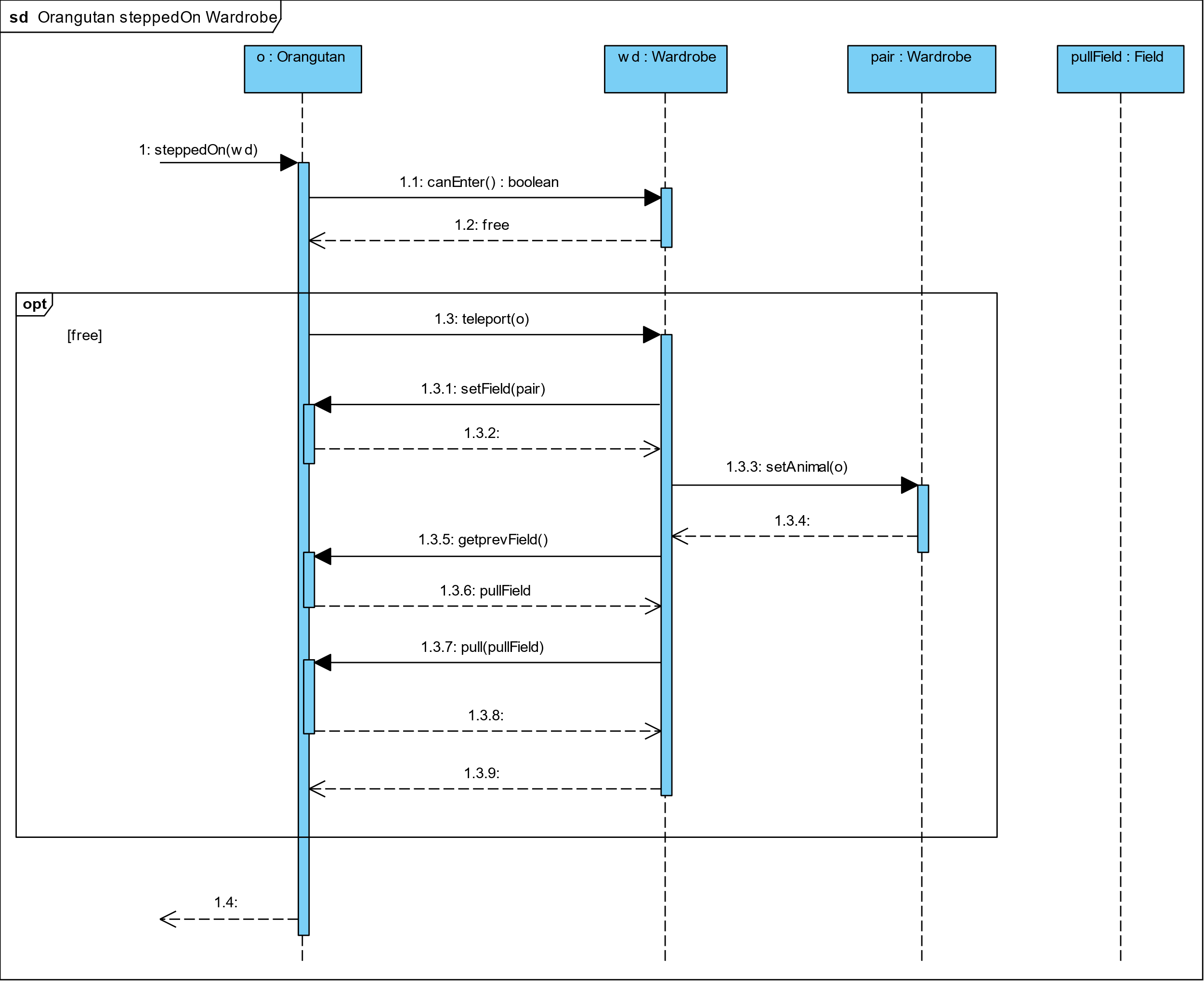
### Orangutan steppedOn GameMachine



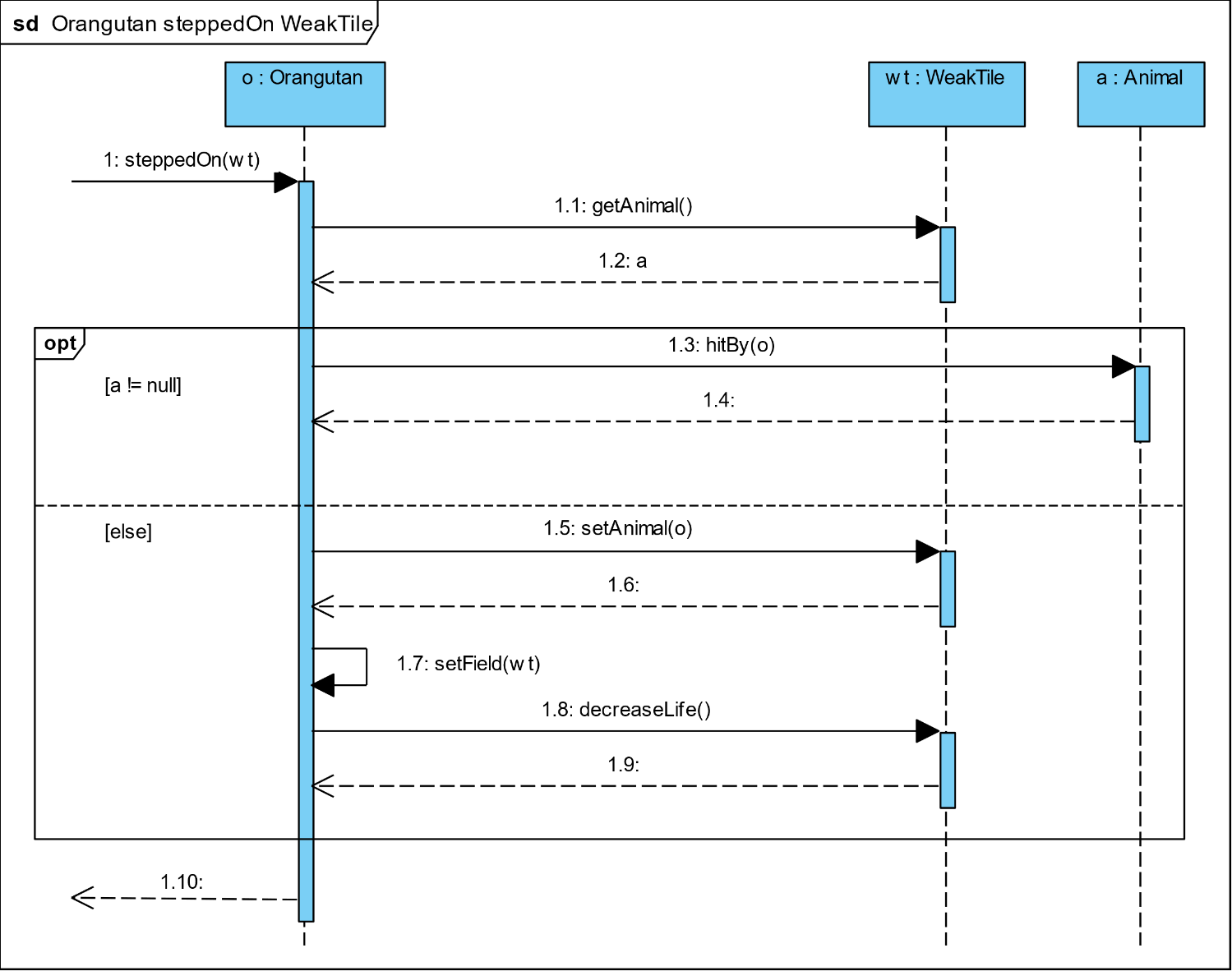
### Orangutan steppedOn Tile



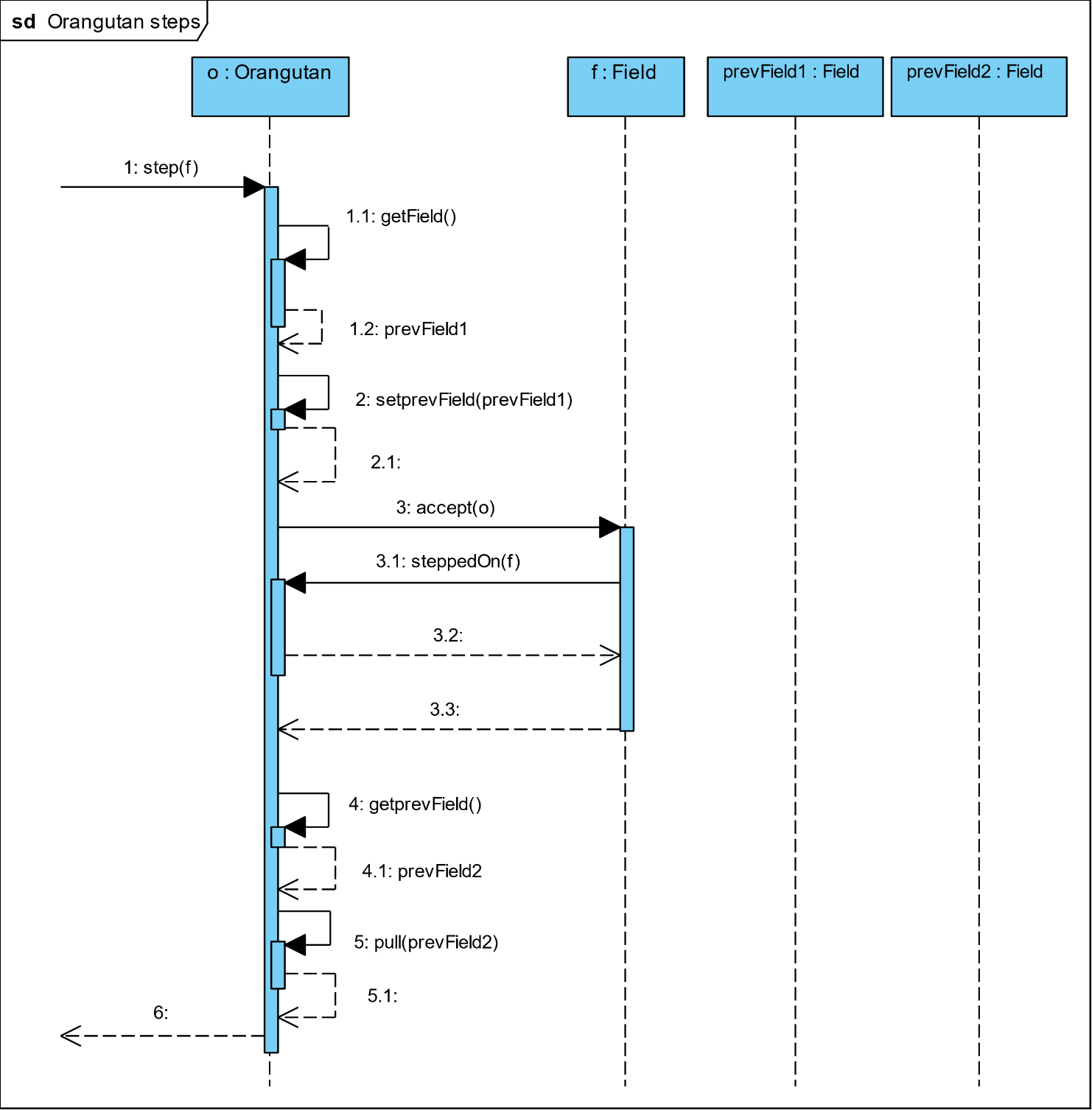
### Orangutan steppedOn Wardrobe



### Orangutan steppedOn WeakTile

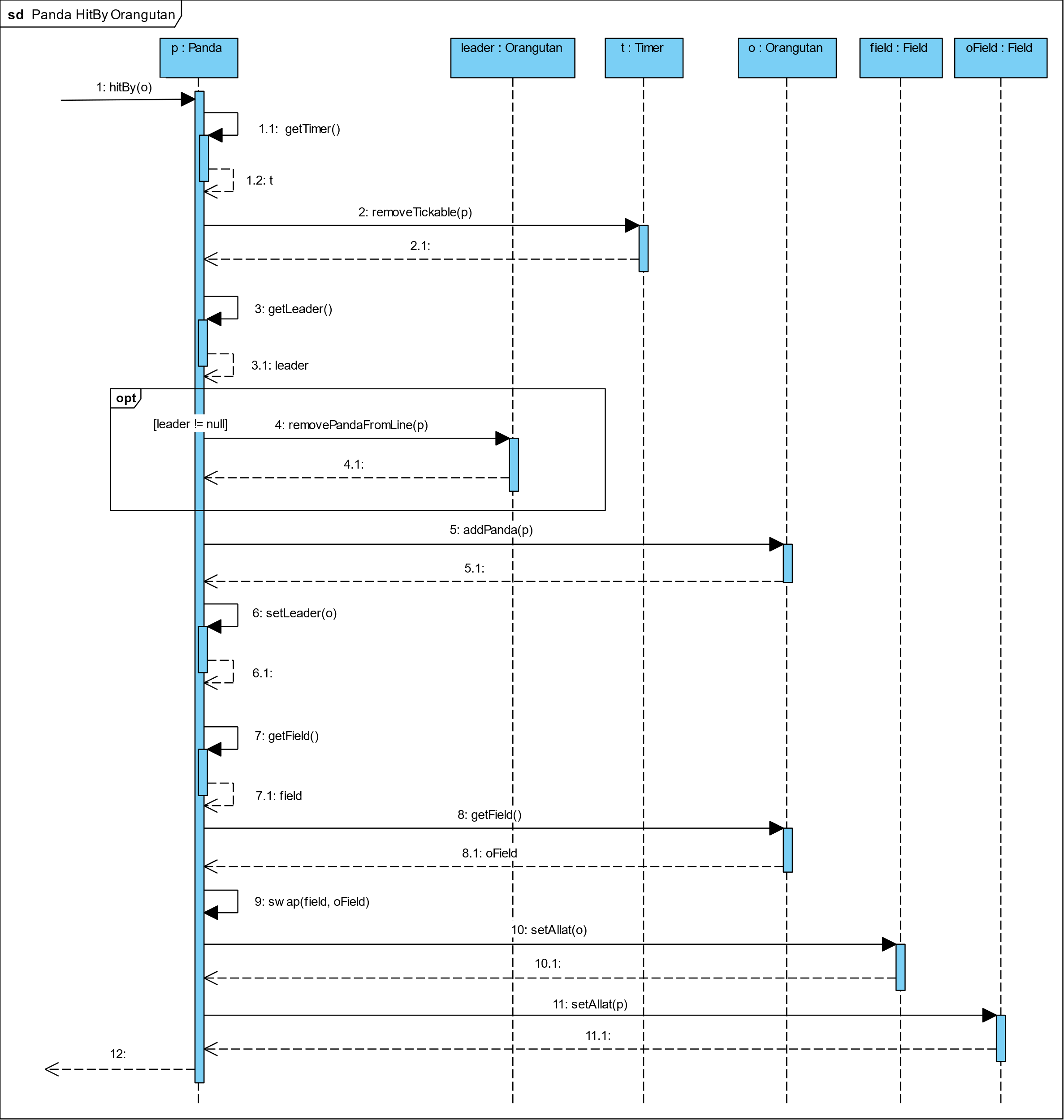


### Orangutan steps

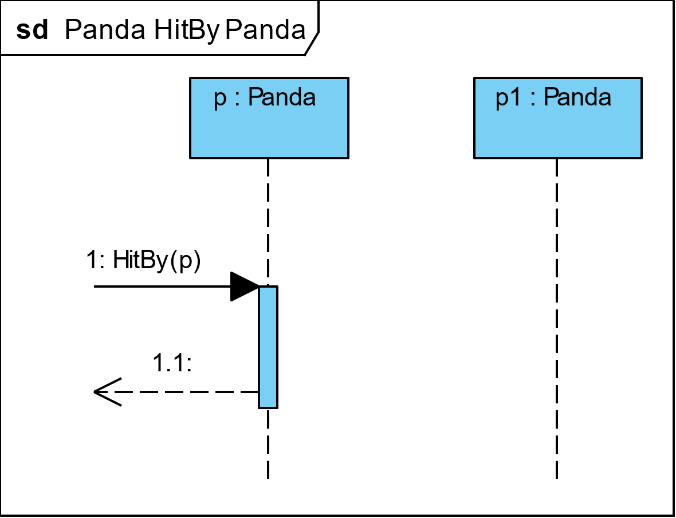


### Panda dies

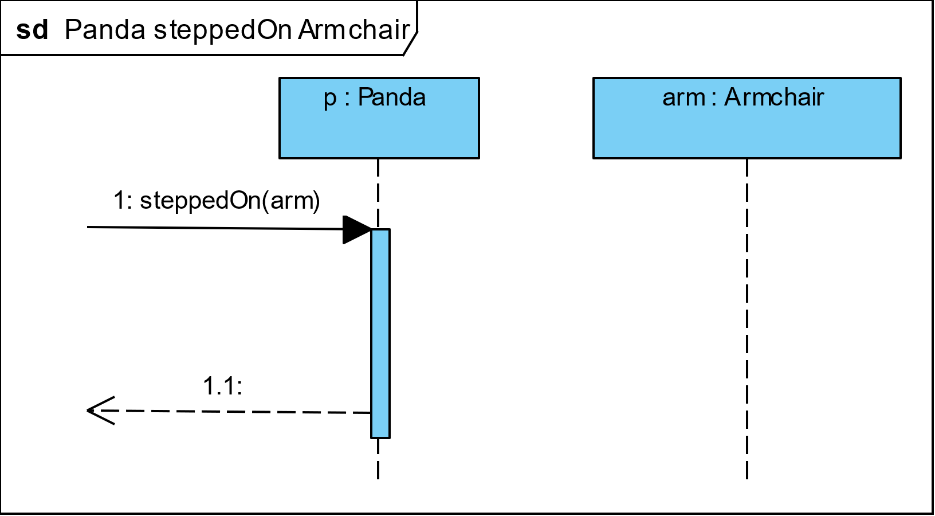
### Panda HitBy Orangutan



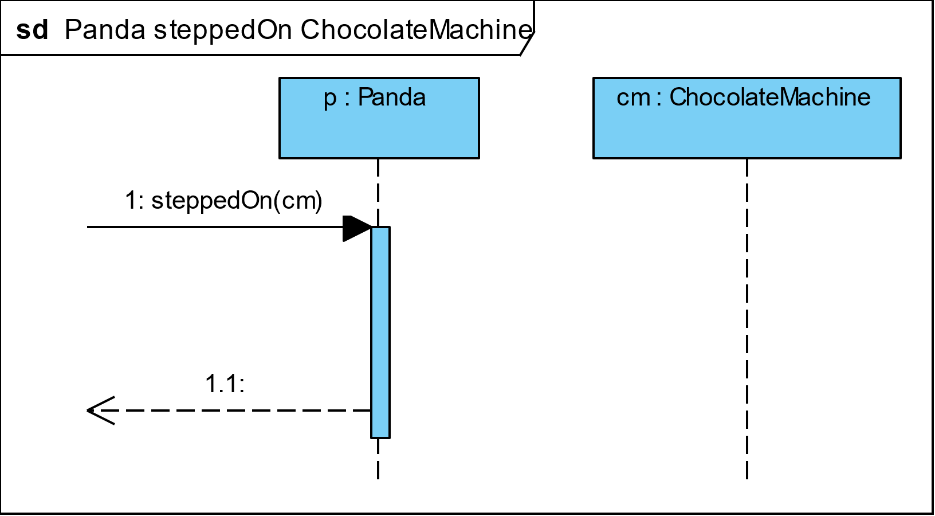
### Panda HitBy Panda



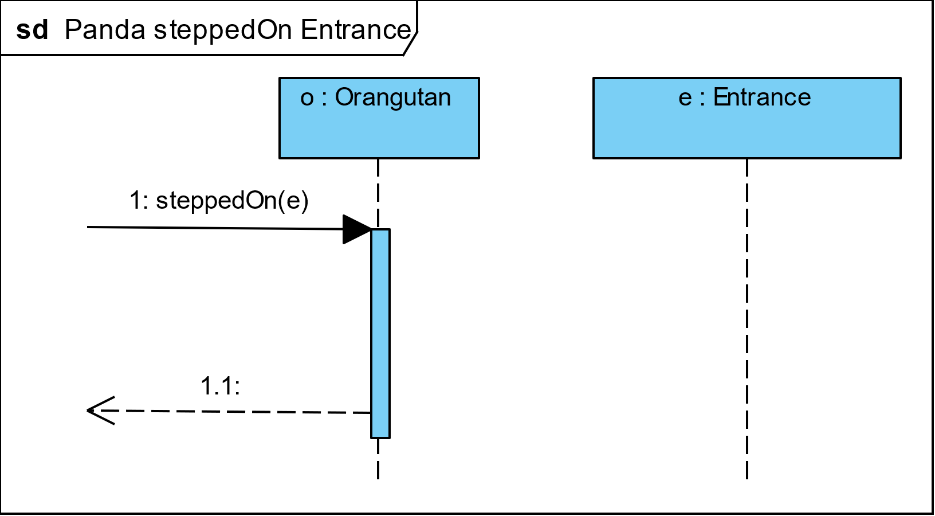
### Panda steppedOn Armchair



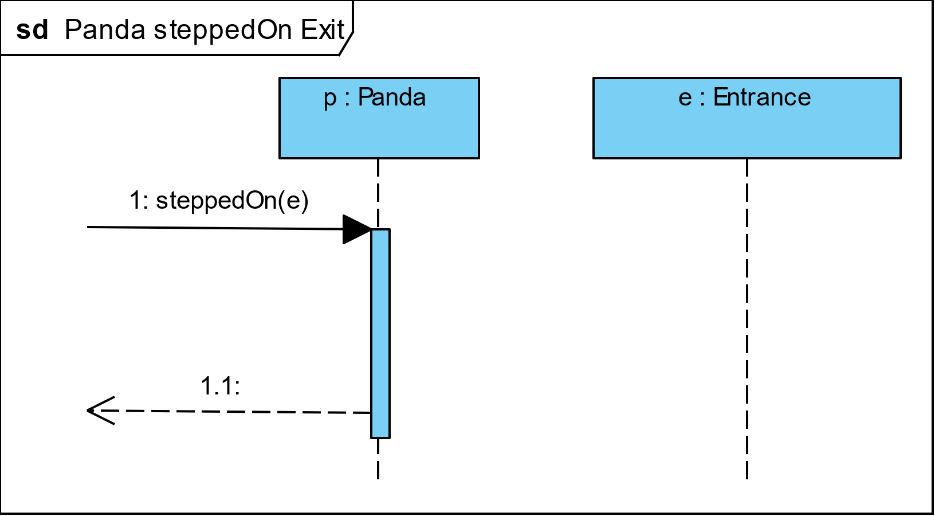
### Panda steppedOn ChocolateMachine



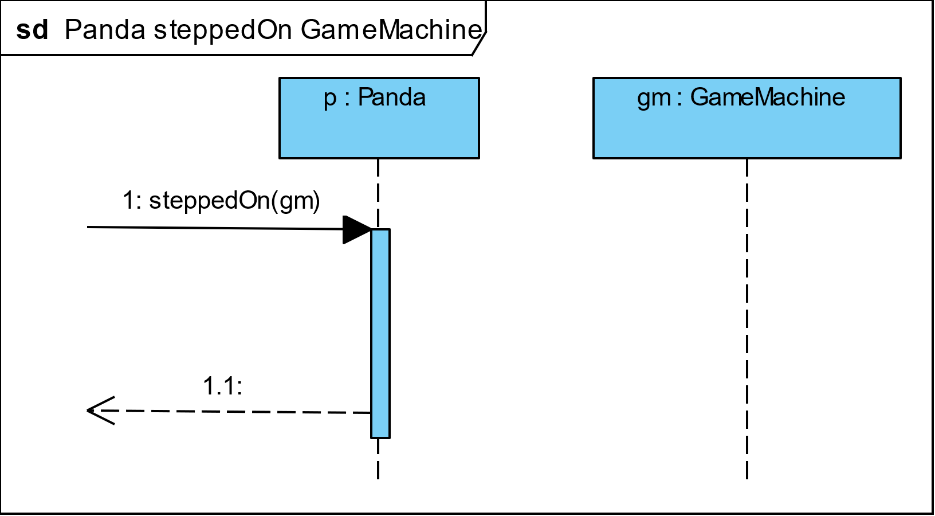
### Panda steppedOn Entrance



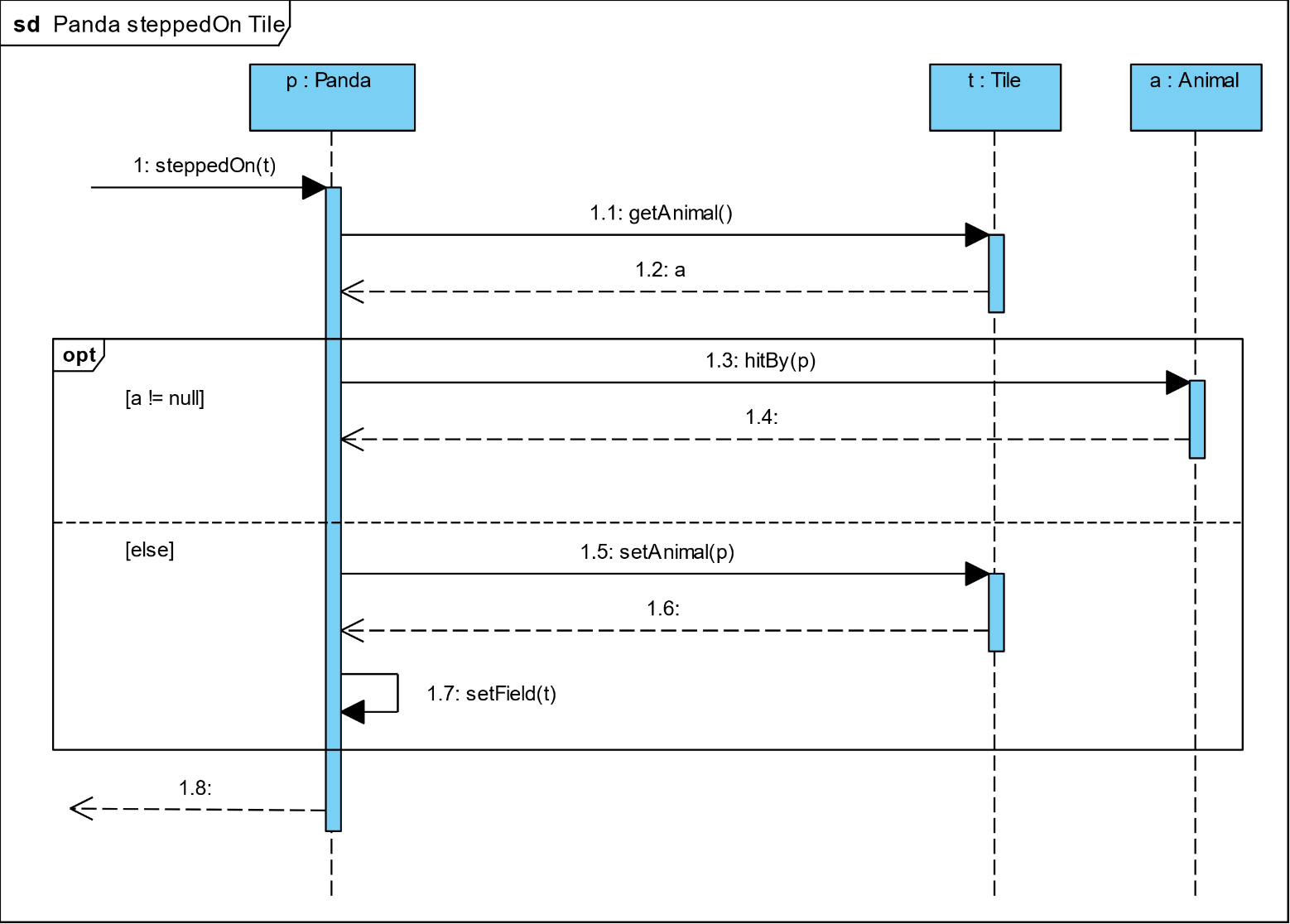
### Panda steppedOn Exit



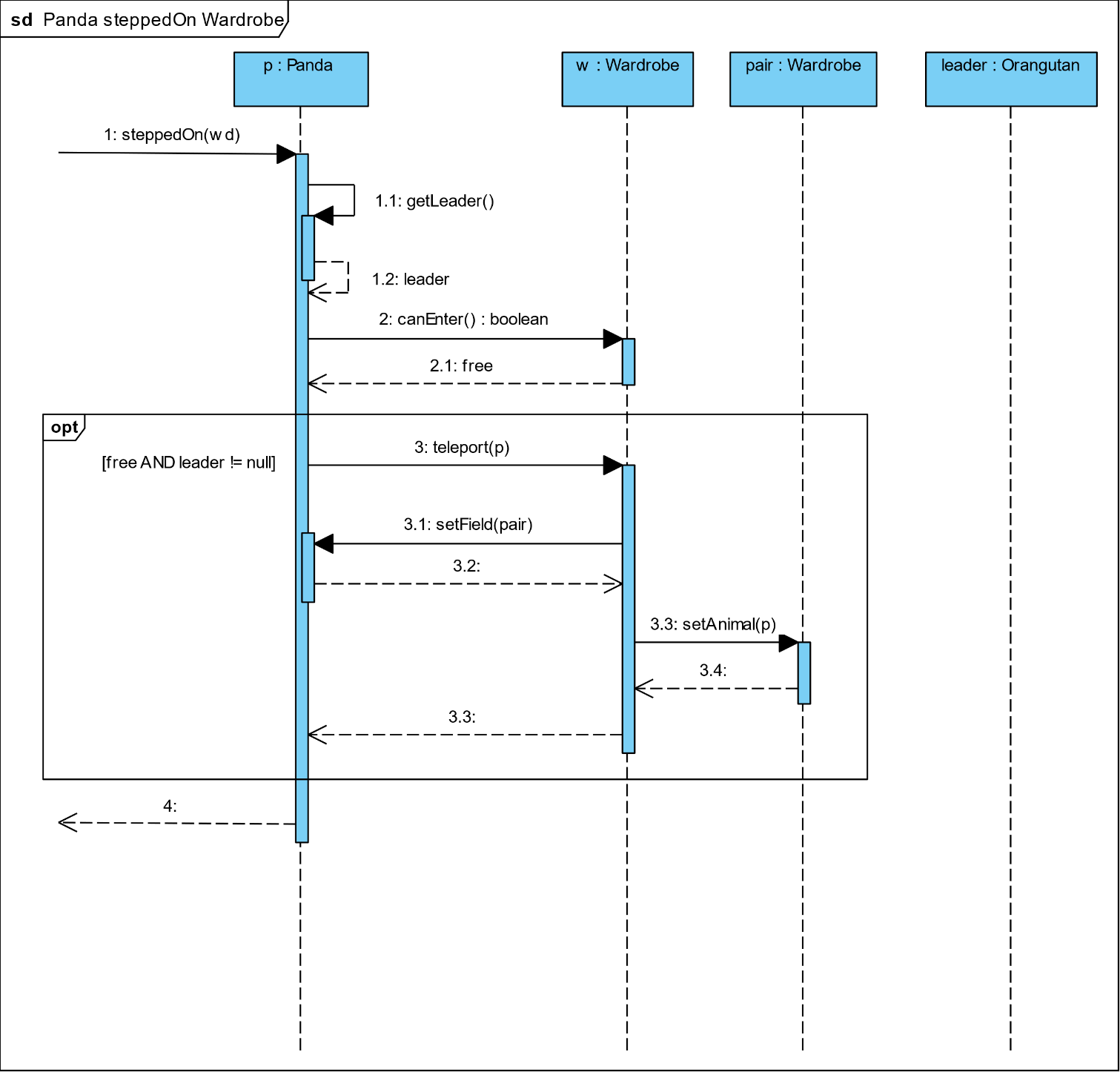
### Panda steppedOn GameMachine



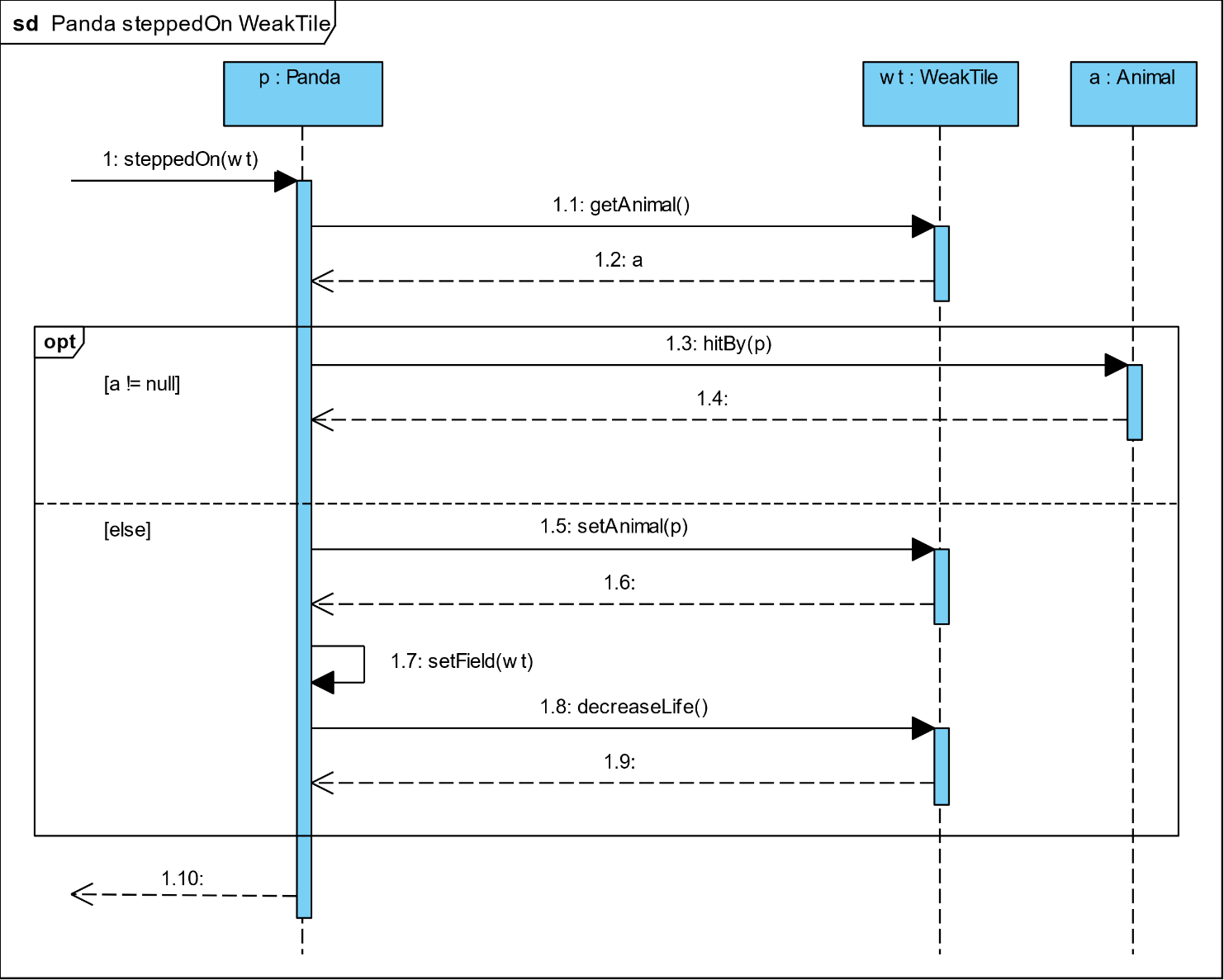
### Panda steppedOn Tile



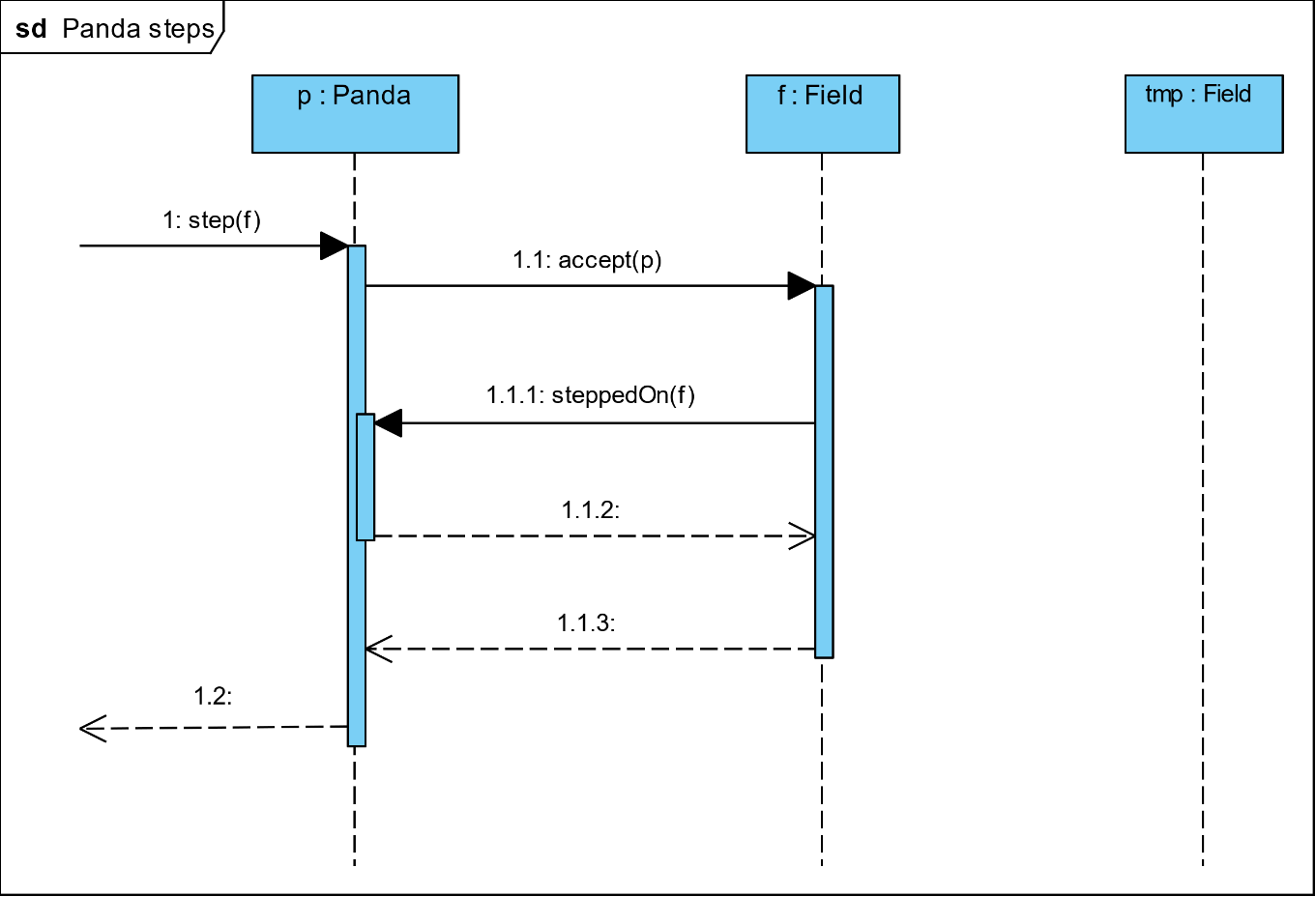
### Panda steppedOn Wardrobe



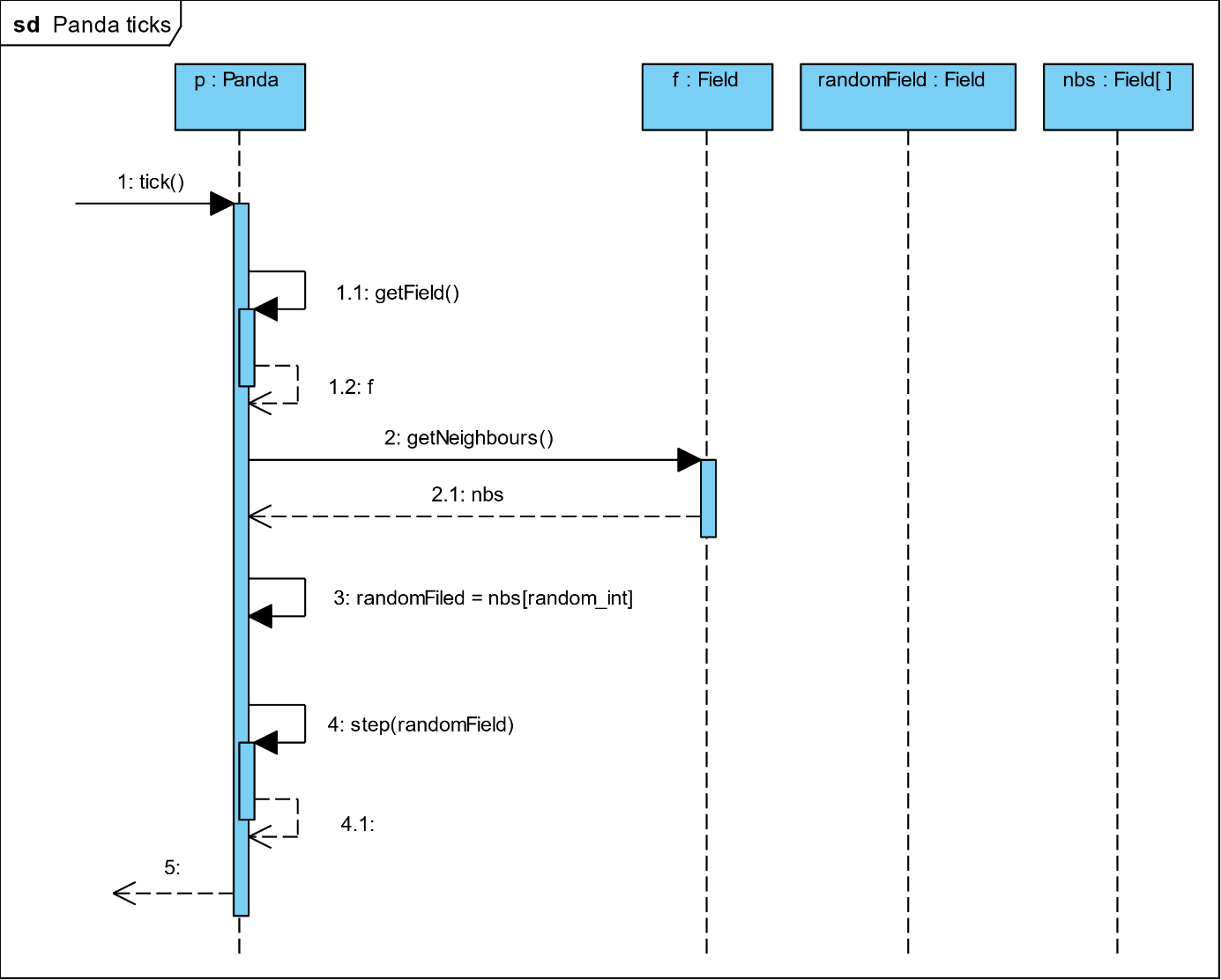
### Panda steppedOn WeakTile



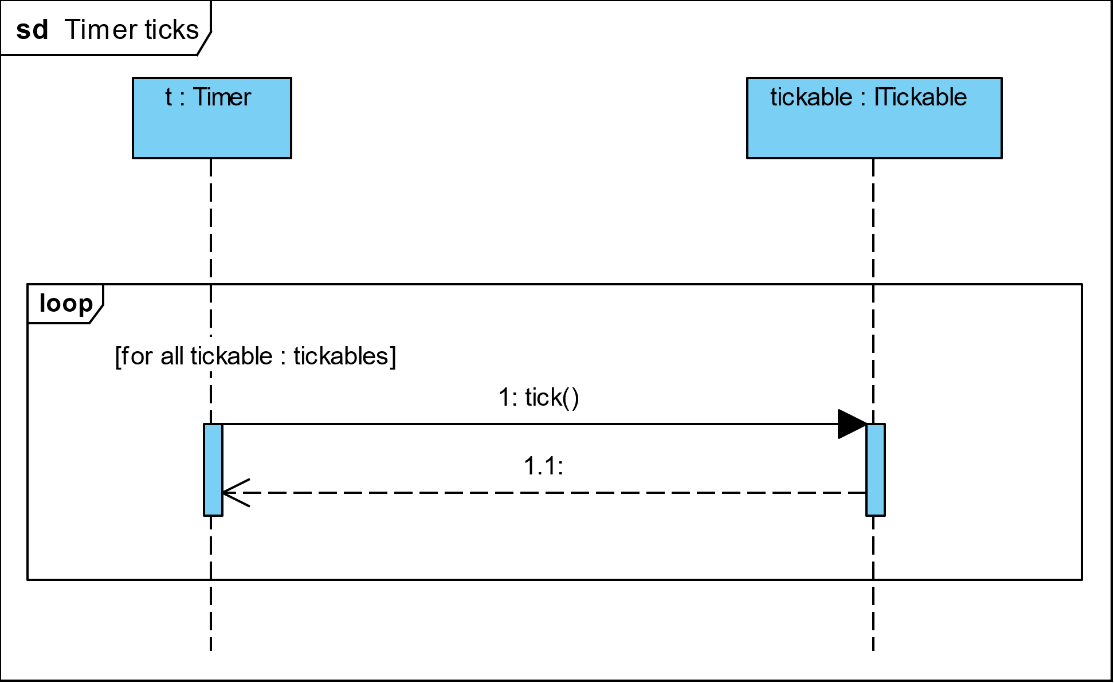
### Panda steps



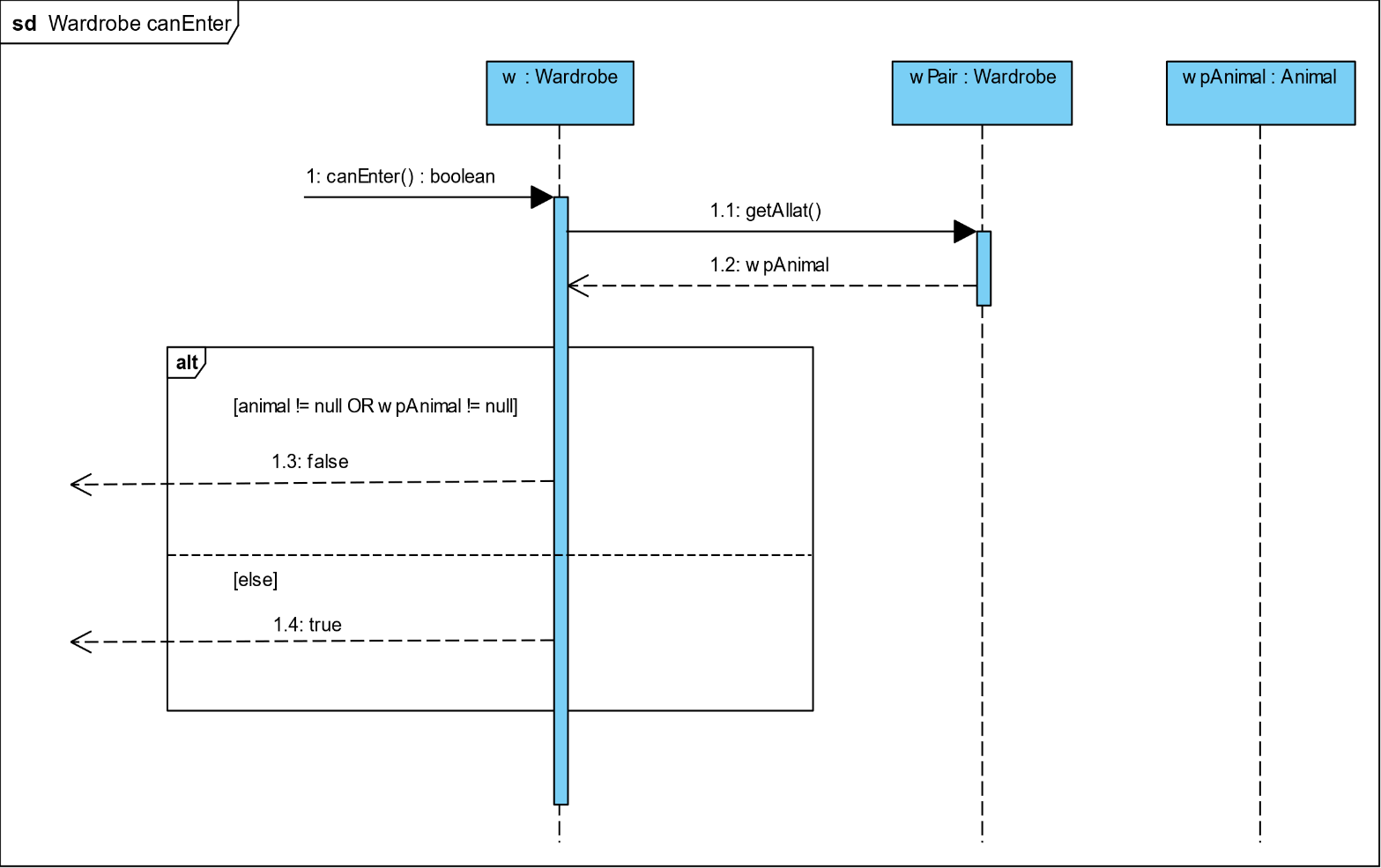
### Panda ticks



### Timer ticks



### Wardrobe canEnter



### WeakTile decreaseLife

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2019.02.17. 12:00 - 18:00 | 6 óra | Horesnyi  Hulej  Kaszala  Kovács  Lőrincz | Értekezlet:  Osztálydiagram megtervezése. Szekvenciadiagramok kezdetleges tervezése. |
| 2019.02.21. 12:15 - 16:00 | 3 óra 45 perc | Horesnyi  Hulej  Kaszala  Kovács  Lőrincz | Értekezlet:  Osztálydiagram véglegesítése, szekvenciadiagramok tervezése |
| 2019.02.23.  12:30 – 14:00 | 1 óra 30 perc | Kaszala | Osztálydiagram szerkesztése, átnézése, javítása |
| 2019.02.23. 15:00 -  2019.02.24. 02:30 | 11 óra 30 perc | Kaszala Kovács | Szekvencia diagramok tervezése, és megrajzolása, felmerülő problémák kezelése. Osztálydiagramon kisebb változtatások, a konzisztencia érdekében. |
| 2019.02.23.  18:45-21:00 | 2 óra 15 perc | Horesnyi | Osztályok leírása:  3.3.1, 3.3.4, 3.3.8, 3.3.9, 3.3.13, 3.3.14-3.3.18, 3.3.21 |
| 2019.02.23.  18:45-20:45 | 2 óra | Lőrincz | Osztályok leírása:  3.3.2, 3.3.3, 3.3.5-3.3.7, 3.3.10-3.3.12, 3.3.19, 3.3.20 |
| 2019.02.23.  21:30-21:50 | 20 perc | Lőrincz | Javítások az osztályok leírásában, formázás:  3.3.1-3.3.21 |
| 2019.02.24  11:00-13:00 | 2 óra | Lőrincz | Javítások a szekvenciadiagramokon, osztályok leírásának kiegészítése |
| 2019.02.24. 18:00 - 19:00 | 1 óra | Hulej | Dokumentum összeállítása, szekvenciák beszúrása |