

4. Analízis modell kidolgozása 2.

66 – otthonmeglefordult

Konzulens:
Szőke Máté

Csapattagok

Kovács Ákos
Horesnyi Olivér
Hulej Attila
Kaszala Kristóf
Lőrincz Zoltán

H7FTHG
D7DBWE
VTZ65K
S9XEU5
BUXM3K

akoskovacs1980@gmail.com
horesnyi.oliver@freemail.hu
alittahu@gmail.com
kaszalakristof1@gmail.com
zoltan9811@gmail.com

2019.03.04.

Tartalomjegyzék

4.	Analízis modell kidolgozása	4
4.1	Objektum katalógus	4
4.1.1	Armchair.....	4
4.1.2	ChocolateMachine.....	4
4.1.3	ChocolatePanda.....	4
4.1.4	Entrance.....	4
4.1.5	Exit	4
4.1.6	GamblerPanda	4
4.1.7	GameMachine	4
4.1.8	Hole	4
4.1.9	LazyPanda	4
4.1.10	Map.....	4
4.1.11	Orangutan	4
4.1.12	Panda	5
4.1.13	Player.....	5
4.1.14	Tile	5
4.1.15	Wardrobe	5
4.1.16	WeakTile	5
4.2	Statikus struktúra diagramok	6
4.3	Osztályok leírása.....	7
4.3.1	<i>Animal</i>	7
4.3.2	Armchair.....	8
4.3.3	ChocolateMachine.....	8
4.3.4	ChocolatePanda.....	9
4.3.5	Entrance.....	9
4.3.6	Exit	10
4.3.7	Field.....	11
4.3.8	GamblerPanda	11
4.3.9	Game	12
4.3.10	GameMachine	13
4.3.11	ITickable.....	13
4.3.12	LazyPanda	14
4.3.13	Orangutan	15
4.3.14	Panda	16
4.3.15	Player.....	17
4.3.16	Timer	17
4.3.17	Wardrobe	18
4.3.18	WeakTile	18
4.4	Szekvencia diagramok	19
4.4.1	Armchair standUp	19
4.4.2	Armchair wantToSitDown	20
4.4.3	ChocolateMachine ticks	21
4.4.4	ChocolatePanda jumps	22
4.4.5	Entrance enterToField	22
4.4.6	GamblerPanda gotScared	23
4.4.7	GameMachine ticks.....	24
4.4.8	LazyPanda steppedOnField.....	25
4.4.9	LazyPanda steppenOnWeakTile	26
4.4.10	Orangutan dies.....	27

4.4.11	Orangutan pull	28
4.4.12	Orangutan removePandaFromLine	29
4.4.13	Orangutan step Exit	30
4.4.14	Orangutan step Field	30
4.4.15	Orangutan step Wardrobe.....	31
4.4.16	Orangutan step WeakTile	31
4.4.17	Orangutan steppedOn Wardrobe	32
4.4.18	Orangutan steppedOnExit	33
4.4.19	Orangutan steppedOnField.....	34
4.4.20	Orangutan steppedOnWeakTile	35
4.4.21	Panda dies.....	36
4.4.22	Panda hitByOrangutan	37
4.4.23	Panda step Field	38
4.4.24	Panda step Wardrobe.....	38
4.4.25	Panda step WeakTile	39
4.4.26	Panda steppedOnField.....	40
4.4.27	Panda steppedOnWardrobe	41
4.4.28	Panda steppedOnWeakTile	42
4.4.29	Panda ticks.....	43
4.4.30	Timer ticks.....	43
4.5	State-chartok	44
4.5.1	ChocolataMachine whistling Stm	44
4.5.2	WeakTileStm.....	44
4.6	Napló	45

4. Analízis modell kidolgozása

4.1 Objektum katalógus

4.1.1 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.

4.1.2 ChocolateMachine

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

4.1.3 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.

4.1.4 Entrance

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

4.1.5 Exit

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

4.1.6 GamblerPanda

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

4.1.7 GameMachine

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

4.1.8 Hole

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

4.1.9 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.

4.1.10 Map

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

4.1.11 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.

4.1.12 Panda

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

4.1.13 Player

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

4.1.14 Tile

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

4.1.15 Wardrobe

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

4.1.16 WeakTile

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



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

4.3 Osztályok leírása

4.3.1 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.
- **Game game:** A játékot kezelő objektum.

- **Metódusok**

- **void die():** Az állat meghal és eltűnik a pályáról.
- **void step(Field to):** Az állat a megadott mezőre lép.
- **void steppedOnField(Field f):** Az állat csempére lépett.
- **void steppedOnWeakTile(WeakTile wt):** Az állat gyenge csempére lépett.
- **void steppedOnChocolateMachine(ChocolateMachine cm):** Az állat csokiautomatára lépett.
- **void steppedOnArmchair(Armchair am):** Az állat fotelra lépett.
- **void steppedOnExit(Exit ex):** Az állat kijáratra lépett.
- **void steppedOnGameMachine(GameMachine gm):** Az állat játékautomatára lépett.
- **void steppedOnEntrance(Entrance ent):** Az állat bejáratra lépett.
- **void steppedOnWardrobe(Wardrobe w):** Az állat szekrényre lépett.
- **void HitByOrangutan(Orangutan o):** Az állattal ütközik egy orángután.
- **void HitByPanda(Panda p):** Az állattal ütközik egy panda.
- **void gotScared():** Az állat megijed.
- **void jump():** Az állat ugrik.

4.3.2 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):** Meghívja az állat SteppedOnArmchair() metódusát.
- **void standUp():** Ha van állat a fotelen, akkor véletlenszerűen léptetheti az állatot a szomszédos mezők valamelyikére.
- **void tick():** Meghívja a standUp() metódust.
- **void wantToSitDown(LazyPanda lp):** Áthelyezi a LazyPandát erre a mezőre.

4.3.3 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**

- **boolean whistling:** Attól függően igaz vagy hamis az értéke, hogy sípol-e az automata
- **int tickCounter:** Számolja, hogy hány tick telt el a sípolás kezdete óta.

- **Metódusok**

- **void accept(Animal a):** Meghívja az állat SteppedOnChocolateMachine() metódusát.
- **void tick():** Ha nem sípol az automata, véletlenszerűen elkezd sípolni, vagy nem. Ha sípol, ugrasztja a szomszédos mezőkön lévő állatokat és növeli a tickCountert. Ha a tickCounter eléri a 10-et, abbamarad a sípolás és nullázódik a számláló.

4.3.4 ChocolatePanda

- **Felelősség**

Ha meghallja a csokiautomata sípolását, ugrik egyet.

- **Ősosztályok**

Animal → Panda

- **Interfészek**

ITickable

- **Attribútumok**

-

- **Metódusok**

- **void jump():** A panda ugrik egyet, ezzel meghívja az animalJumped() metódust azon a mezőn, amin áll.

4.3.5 Entrance

- **Felelősség**

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

- **Ősosztályok**

Field

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **void accept(Animal a):** Meghívja az állat SteppedOnEntrance() metódusát.
- **void enterToField(Orangutan o):** Elhelyezi az orángutánt az adott mezőn (az orángután fieldje az entrance objektum lesz, az entrance animalja pedig az orángután).

4.3.6 Exit

- **Felelősség**

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

- **Ősosztályok**

Field

- **Interfészek**

-

- **Attribútumok**

- **Entrance entrancePair:** A kijárat ebben az attribútumban tárolja a hozzá tartozó bejáratot, ahol a kijáraton átlépő orángutának be kell majd jönnie.
- **Game game:** A játékot kezelő 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 játékban az orángután pandáinak számával csökkenti az aktuális pandák számát a game-en keresztül. A pandák számának megfelelően pontozza az orángutánt irányító játékost a game-en keresztül. Törli a kivitt pandákat az orángután láncából továbbá leveszi őket a Timer listájáról és a hozzájuk tartozó mezőkről (null-ra állítja a mezők animal referenciáját). Meghívja az entrancePair enterToField() metódusát az átvett orángutánt paraméterként átadva.

4.3.7 Field

- **Felelősség**

Egy pályabeli csempét reprezentál és őssosztálya a különböző pályaelemeknek. Állatok léphetnek rá és el is tárolja, hogy mely állat tartózkodik rajta. Ismeri a szomszédos mezőket is.

- **Őssosztályok**

-

- **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.
- **void animalJumped():** Egy állat hívja meg, aki ugrott az adott mezőn.
- **void wantToSitDown(LazyPanda lp):** Egy LazyPanda hívja meg, aki le akar ülni az adott mezőn.

4.3.8 GamblerPanda

- **Felelősség**

Ha meghallja a játékgép csilingelését, megijed.

- **Őssosztályok**

Animal → Panda

- **Interfészek**

ITickable

- **Attribútumok**

-

- **Metódusok**

- **void gotScared():** A panda megijed, emiatt ha egy láncba tartozik, akkor elengedi a többi panda kezét (removePandaFromLine() metódus).

4.3.9 Game

- **Felelősség**

Elindítja és leállítja a játékot. Kezeli a játékosokat és a pontjaikat.

- **Össztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **Player[1..2] players:** A játékosokat tároló tömb.
- **int pandaNum:** A pályán lévő pandák aktuális száma.
- **Timer timer:** A játék időzítő objektuma.
- **int timeFromStart:** A játék indulása óta eltelt idő.

- **Metódusok**

- **void start():** Elindítja a játékot.
- **void endGame():** Leállítja a játékot.
- **void decreasePandaNum(int i):** Csökkenti az aktuális pandák számát i-vel.
- **void increaseTime():** Növeli az eltelt időt eggyel.
- **void addPoints(int p, Orangutan o):** Az adott orángutánhoz tartozó játékosnak p pontot ad.

4.3.10 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 csilingel-e az automata.
- **int tickCounter:** A csilingelés kezdete óta eltelt idő.

- **Metódusok**

- **void accept(Animal a):** A játékgép nem fogad egyetlen állatot sem, a metódus azonnal visszatér.
- **void tick():** Ha nem csilingel az automata, véletlenszerűen elkezd csilingelni, vagy nem. Ha csilingel, ugrasztja a szomszédos mezőkön lévő állatokat és növeli a tickCounter-t. Ha a tickCounter eléri a 10-et, abbamarad a csilingelés és nullázódik a számláló.

4.3.11 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.

4.3.12 LazyPanda

- **Felelősség**

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

- **Össztályok**

Animal → Panda

- **Interfészek**

ITickable

- **Attribútumok**

- **int energy:** A panda energiája. Az értéke minimum 0, maximum 2.
- **boolean sitting:** Attól függően igaz vagy hamis az értéke, hogy ül-e a lusta panda.

- **Metódusok**

- **void steppedOnField(Field f):** Ha a paraméterben átadott mezőn van állat, meghívja annak HitByPanda() metódusát. Ha nincs állat, átlép a mezőre. Ezután, ha a panda energiája 0, megpróbál leülni a szomszédos mezőkön.
- **void steppedOnWeakField(WeakTile wt):** Ha a paraméterben átadott mezőn van állat, meghívja annak HitByPanda() metódusát. Ha nincs állat, átlép a mezőre. Ha a gyenge csempe életeréje 0, a panda meghal, a die() metódusa hívódik meg. Ezután, ha a panda energiája 0, megpróbál leülni a szomszédos mezőkön.
- **void steppedOnWardrobe(Wardrobe w):** Meghívja a wardrobe teleportPanda() metódusát, önmagát átadva paraméterként. Ezután, ha a panda energiája 0, megpróbál leülni a szomszédos mezőkön.

Orangutan

- **Felelősség**

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

- **Össztályok**

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 és az összes utána következőt a láncból. Ehhez minden panda vezetőjét null-ra állítja és hozzáadja a Tickable-ök listájához a Timer-ben.
- **void addPanda(Panda p):** Hozzáadja a pandát a láncához.
- **void die():** Meghívja a pull()-t, majd mindent pandát kivesz a láncból. Null-ra állítja mezője állatát.
- **void HitBy(Orangutan o):** Az orángután nem ütközhet orángutánnal, azonnal visszatér.
- **void HitBy(Panda p):** A panda nem mehet neki orángutánnak, azonnal visszatér.
- **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.
- **void steppedOnExit(Exit ex):** Meghívja az exit orangutanExited() metódusát.
- **void steppedOnField(Field f):** Ha a paraméterben átadott mezőn van állat, meghívja annak HitByOrangutan() metódusát. Ha nincs állat, átlép a mezőre és meghívja a pull() metódust.
- **void steppedOnWardrobe(Wardrobe w):** Meghívja a wardrobe teleportOrangutan() metódusát, önmagát átadva paraméterként.
- **void steppedOnWeakTile(WeakTile wt):** Ha a paraméterben átadott mezőn van állat, meghívja annak HitByOrangutan() metódusát. Ha nincs állat, átlép a mezőre. Ha a gyenge csempe életeréje 0, a panda meghal, a die() metódusa hívódik meg.
- **void tick():** A panda a szomszédok közül véletlenszerűen lép az egyik mezőre.

4.3.13 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 die():** A game-en keresztül csökkenti az aktuális pandák számát 1-gyel. Ha van vezetője, akkor kiveszi magát a sorból (removePandaFromLine()). Kiveszi magát a Timer listájából és null-ra állítja a mezője Animal referenciáját.
- **void HitBy(Orangutan o):** Ha a panda hozzá volt már csatolva egy lánchoz, akkor abból a láncból eltávolítja (removePandaFromLine()). Hozzáadja az ütköző orángután láncának elejére, eközben helyet cserél az orángutánal.
- **void HitBy(Panda p):** A panda nem mehet neki az orángutának, azonnal visszatér.
- **void steppedOnField(Field f):** Ha a paraméterben átadott mezőn van állat, meghívja annak HitByPanda() metódusát. Ha nincs állat, átlép a mezőre.
- **void steppedOnWardrobe(Wardrobe w):** Meghívja a wardrobe teleportPanda() metódusát, önmagát átadva paraméterként.
- **void steppedOnWeakTile(WeakTile wt):** Ha a paraméterben átadott mezőn van állat, meghívja annak HitByPanda() metódusát. Ha nincs állat, átlép a mezőre. Ha a gyenge csempe életeréje 0, a panda meghal, a die() metódusa hívódik meg. Ezután, ha a panda energiája 0, megpróbál leülni a szomszédos mezőkön.
- **void tick():** A szomszédos mezők közül véletlenszerűen kiválaszt egyet és oda lépteti a pandát.

4.3.14 Player

- **Felelősség**

Tárolja a játékos pontját és orángutánját.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

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

- **Metódusok**

-

4.3.15 Timer

- **Felelősség**

A Timer osztály felel a játék időzítetttségéért. Össze gyűjti 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.

4.3.16 Wardrobe

- **Felelősség**

A szekrény is egy mezőtípus, átjáróként viselkedik. Egy szekrénybe belépve a párján jövőnk 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 steppedOnWardrobe() metódusát, önmagát adva paraméterként
- **void teleportOrangutan(Orangutan o):** Áthelyezi az orángutánt a párjára, amennyiben azon nem áll semmilyen más állat. Az orángután előző mezőjére húzza az orángután pandáit.
- **void teleportPanda(Panda p):** Ha a pandának nincs vezetője (nincs láncba fűzve) és a szekrény párján semmilyen más állat nem áll, akkor áthelyezi a pandát a párjára.
-

4.3.17 WeakTile

- **Felelősség**

Ha rálépnek, csökken az élete. Ha az élete 0-ra csökken, eltörik és „lyuk válik belőle”.

- **Ősosztályok**

Field → Tile

- **Interfészek**

-

- **Attribútumok**

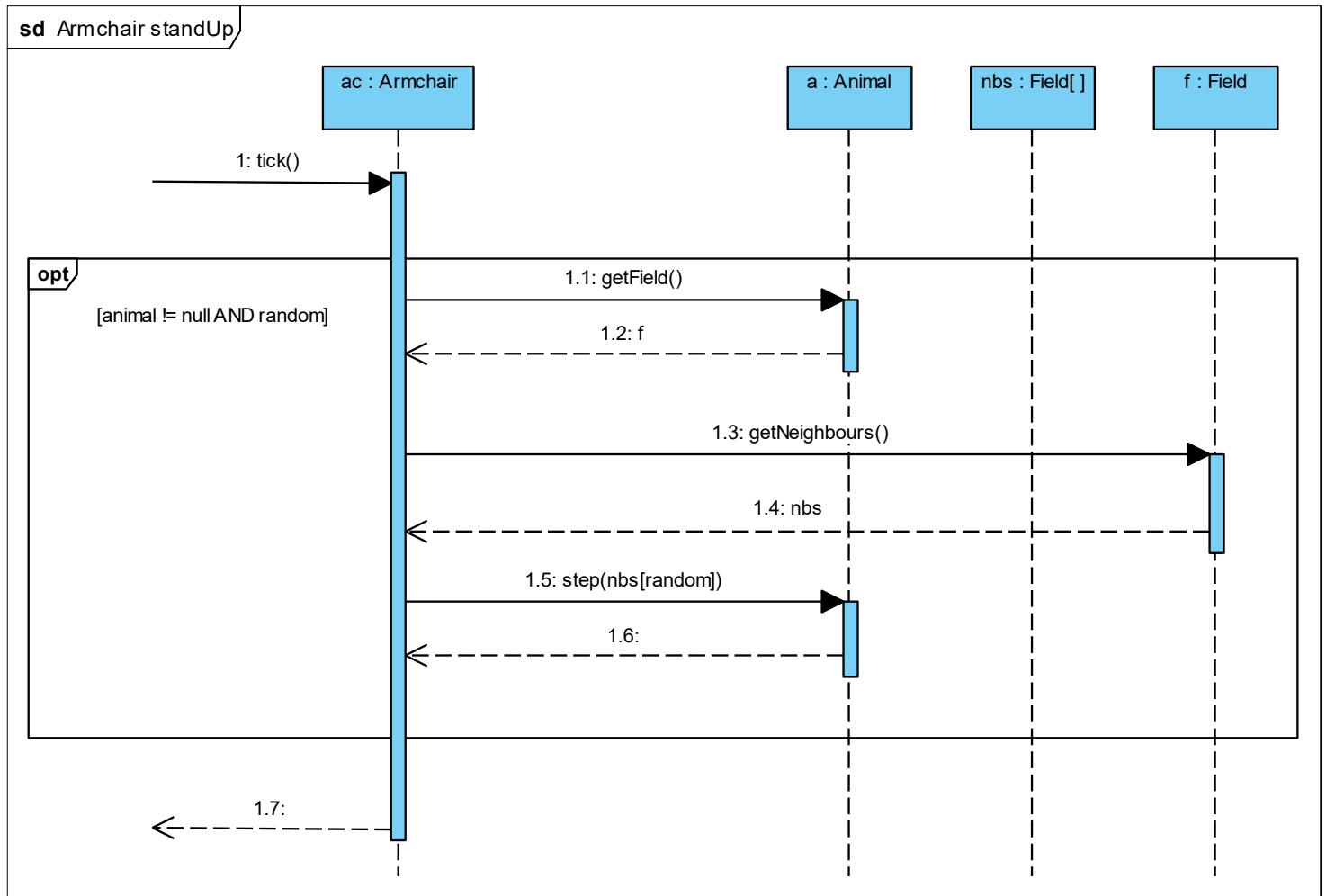
- **int life:** A gyenge csempe életereje.

- **Metódusok**

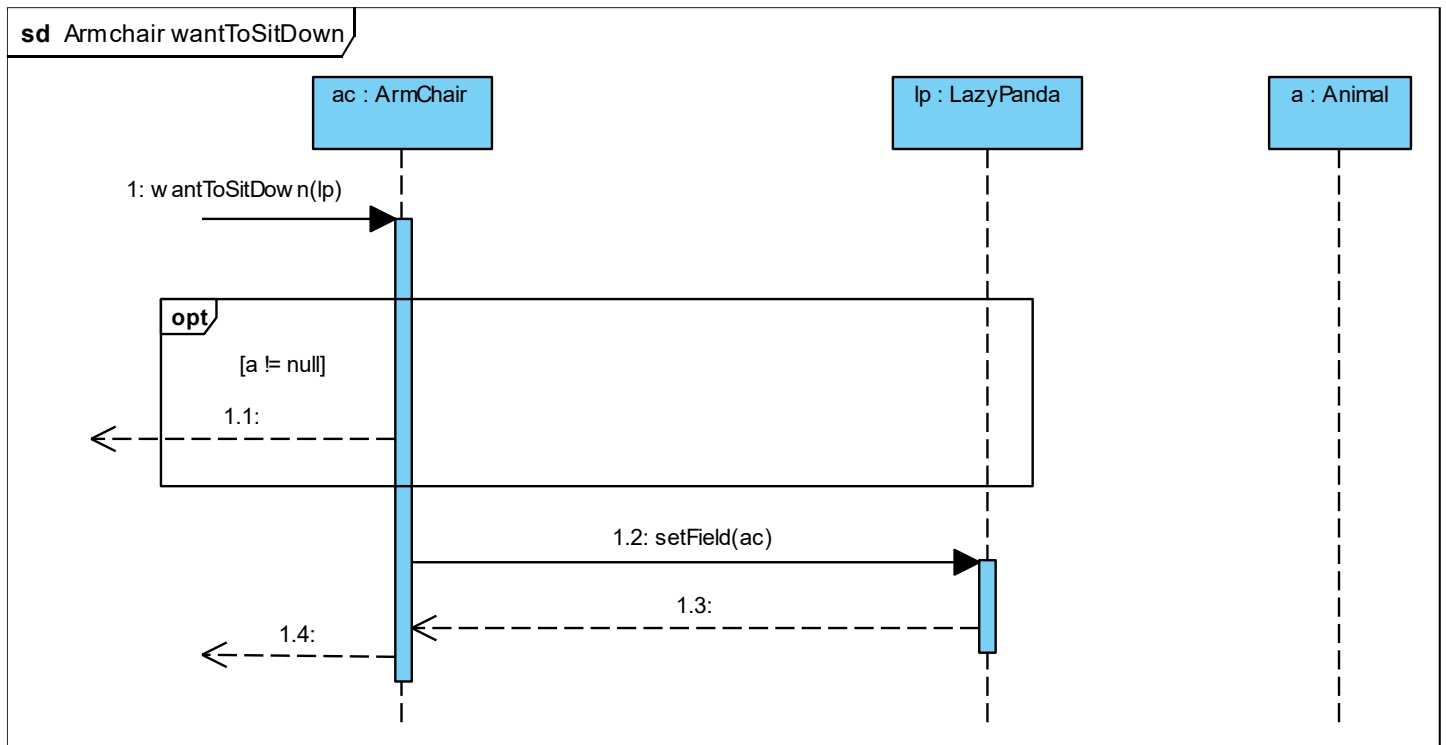
- **void accept(Animal a):** Meghívja az állat steppedOnWeakTile() metódusát.

4.4 Szekvencia diagramok

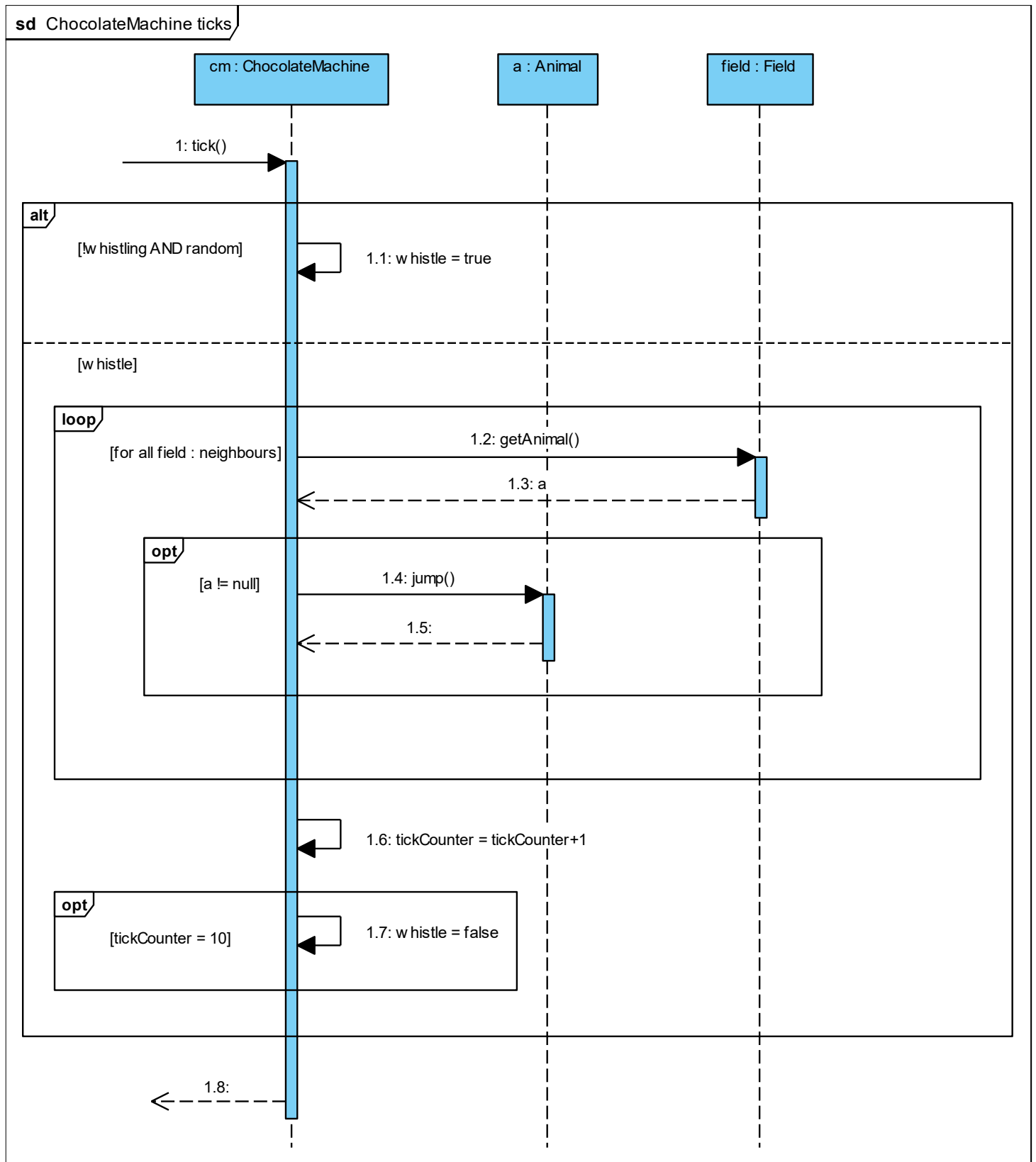
4.4.1 Armchair standUp



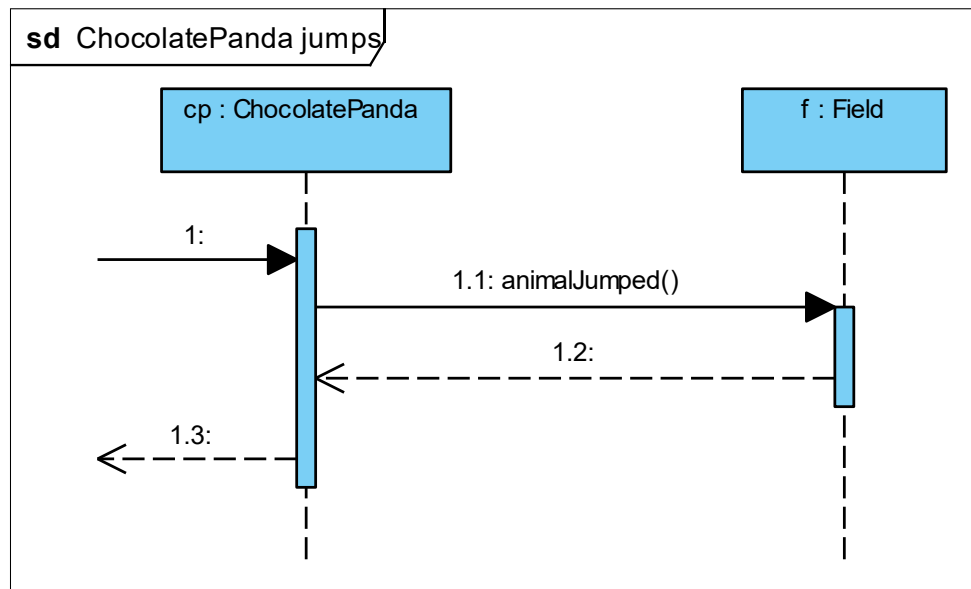
4.4.2 Armchair wantToSitDown



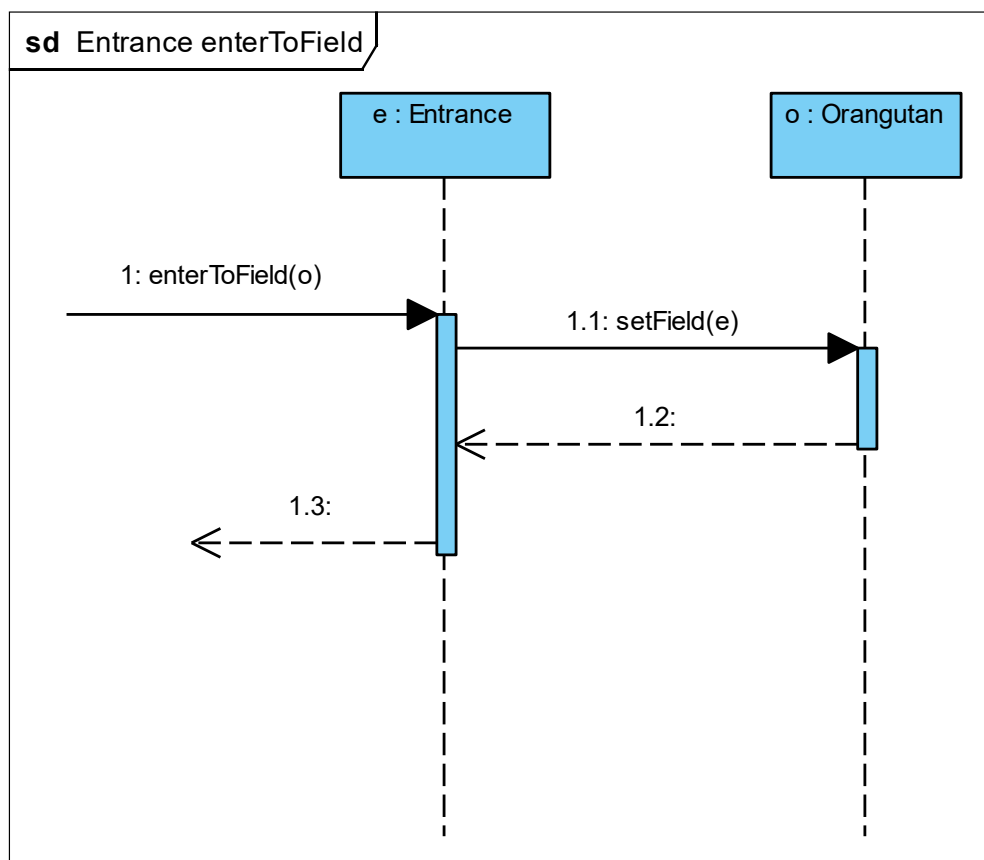
4.4.3 ChocolateMachine ticks



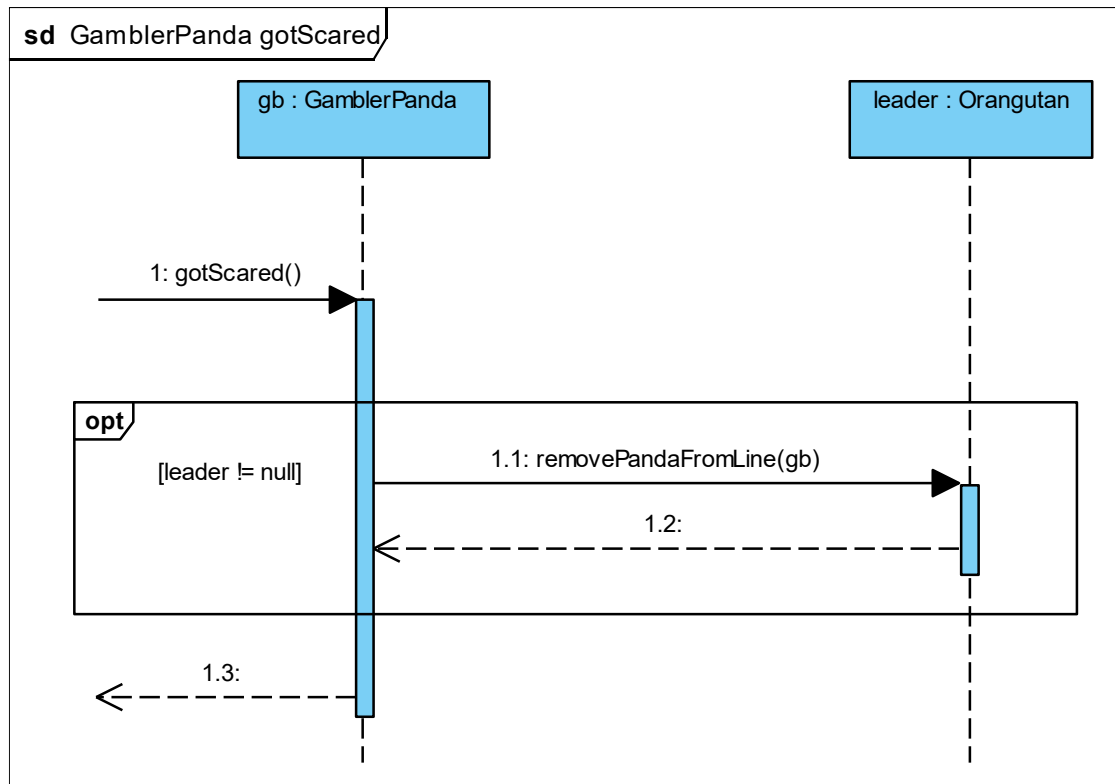
4.4.4 ChocolatePanda jumps



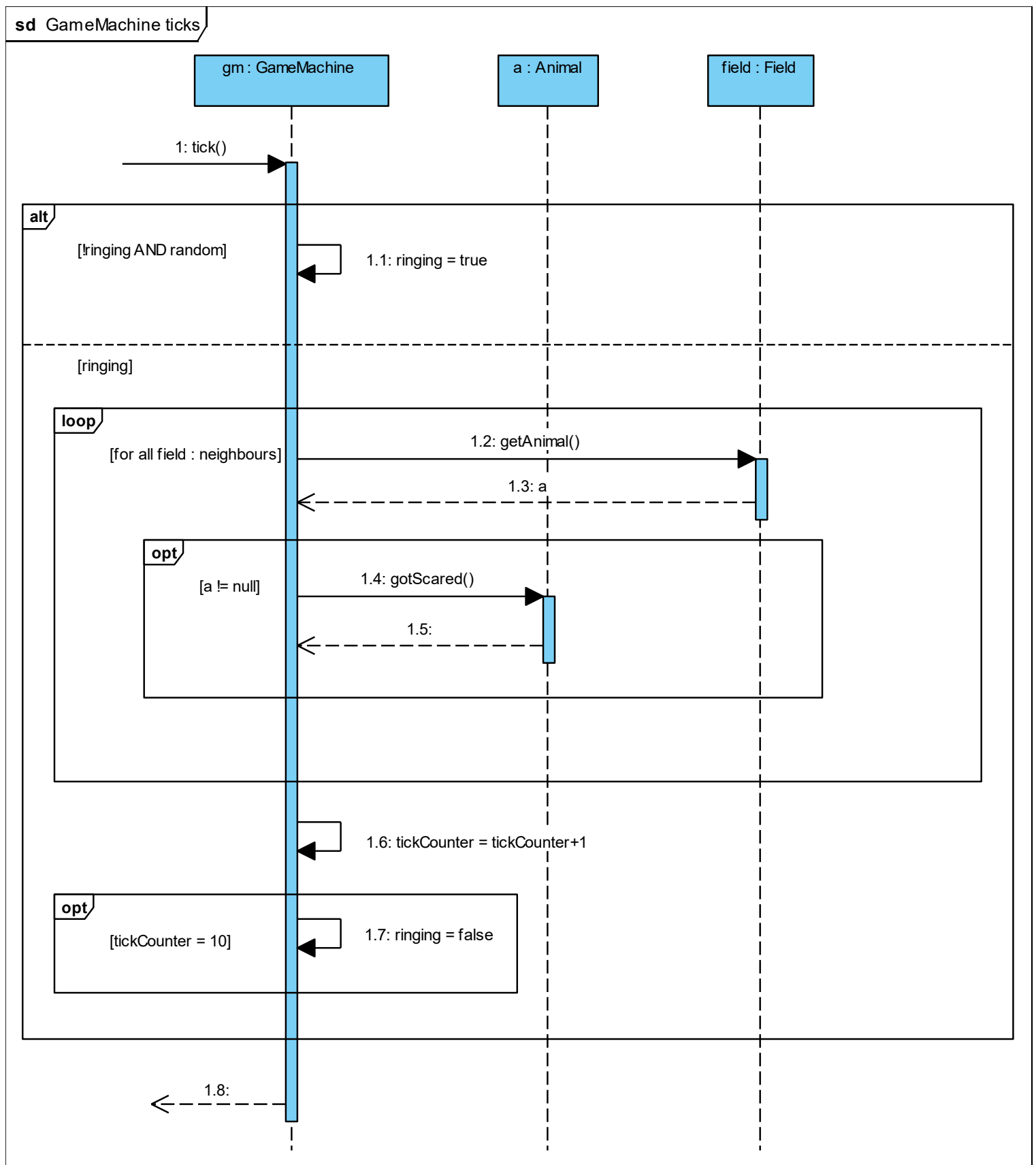
4.4.5 Entrance enterToField



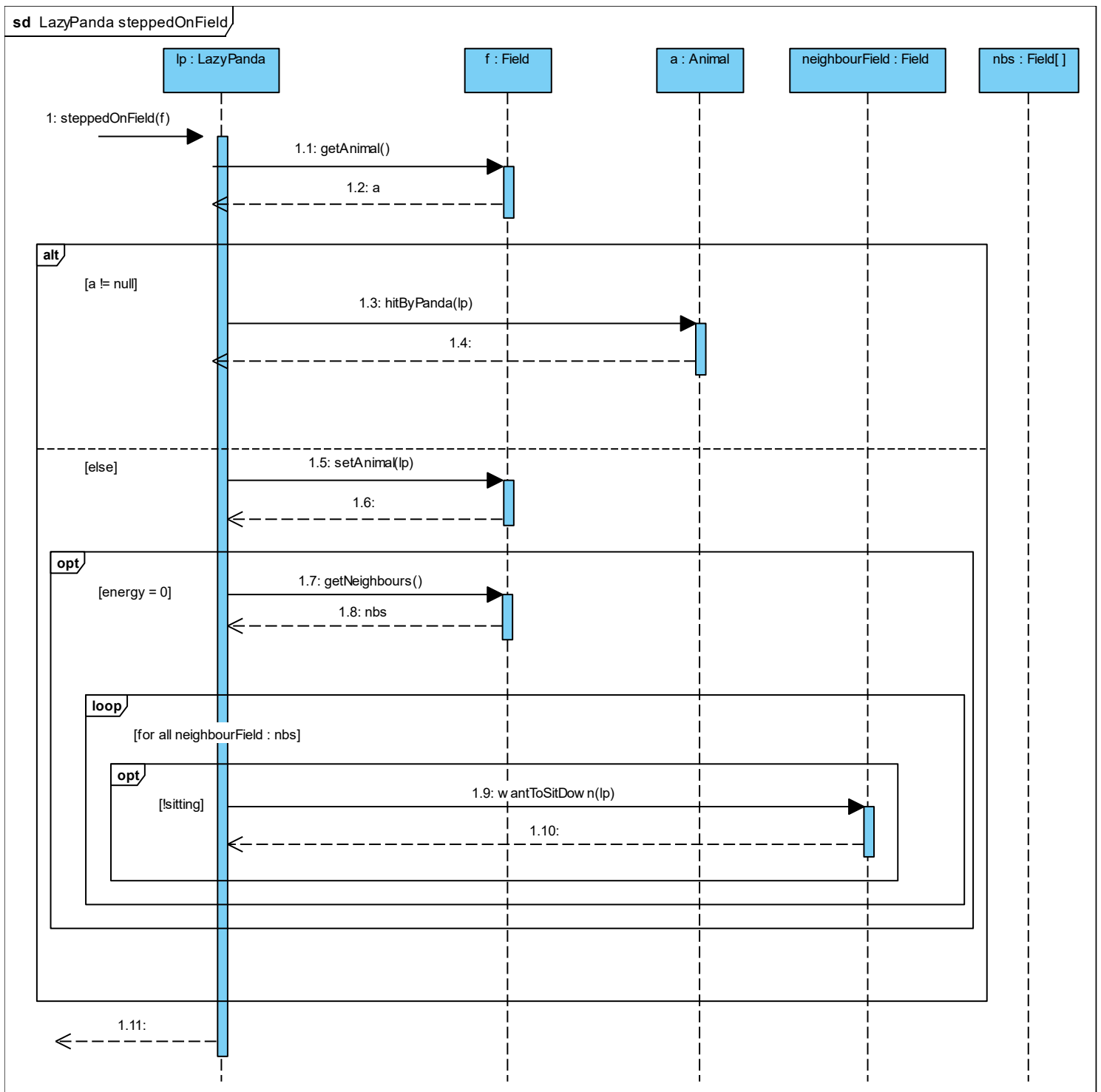
4.4.6 GamblerPanda gotScared



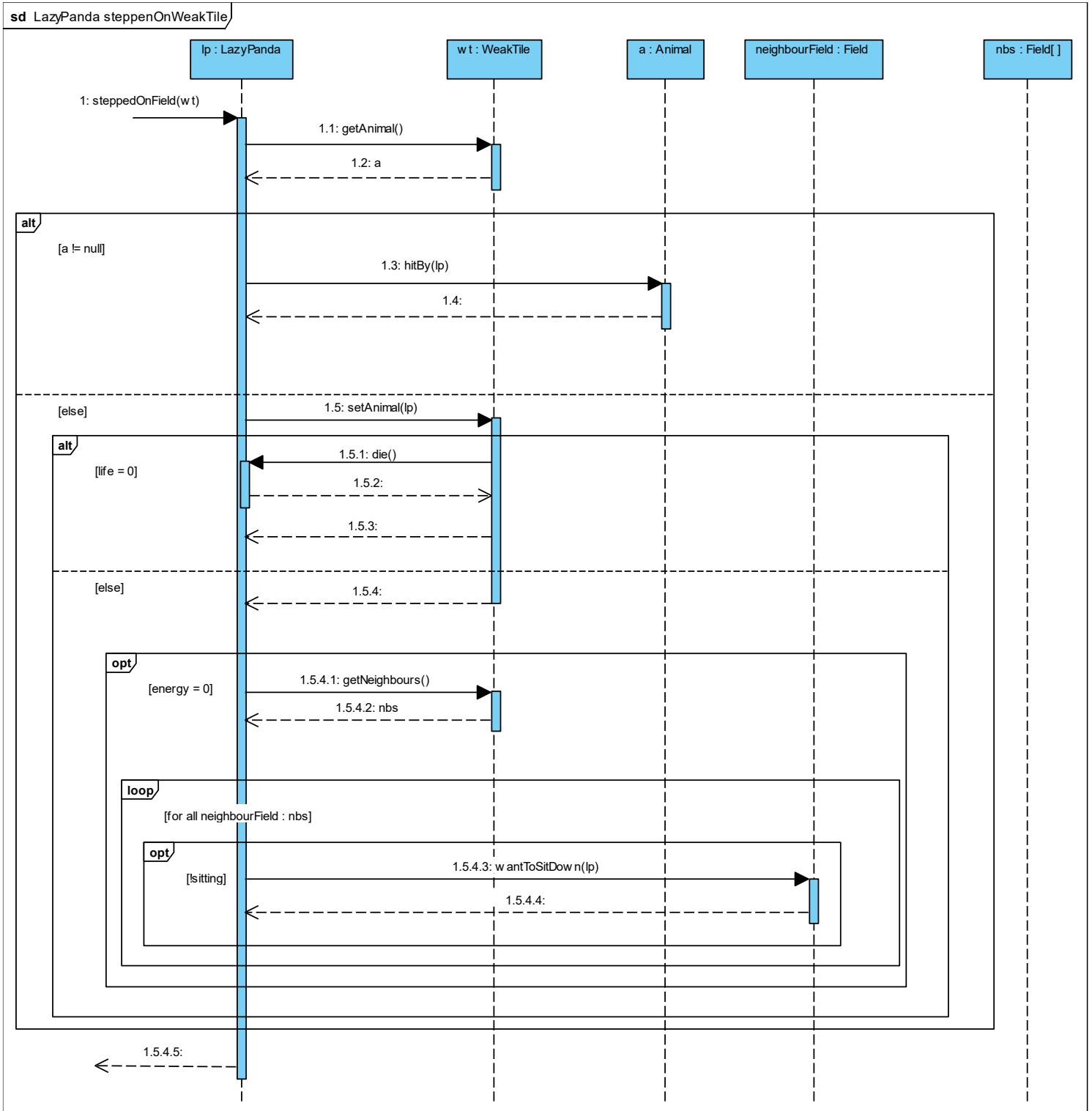
4.4.7 GameMachine ticks



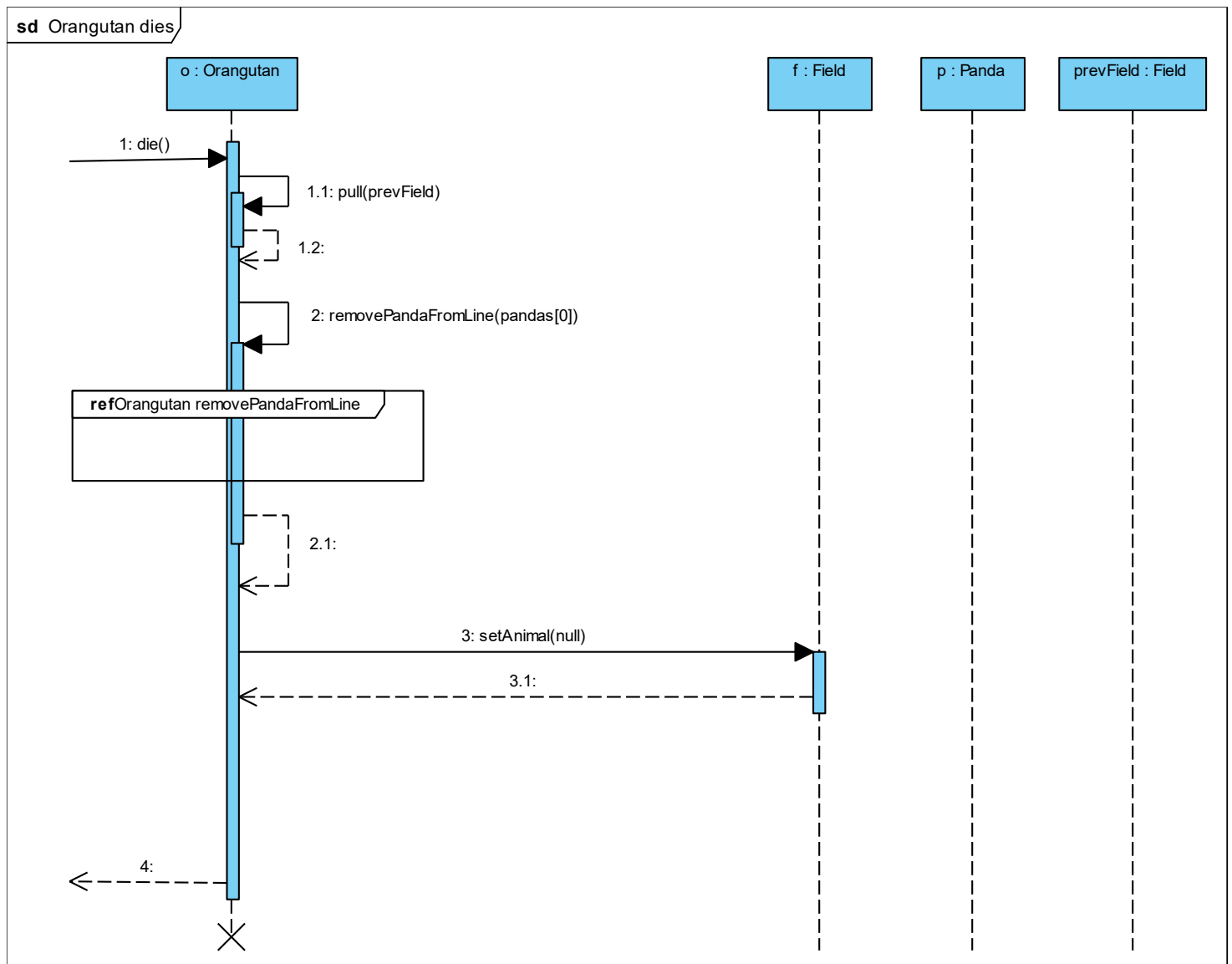
4.4.8 LazyPanda steppedOnField



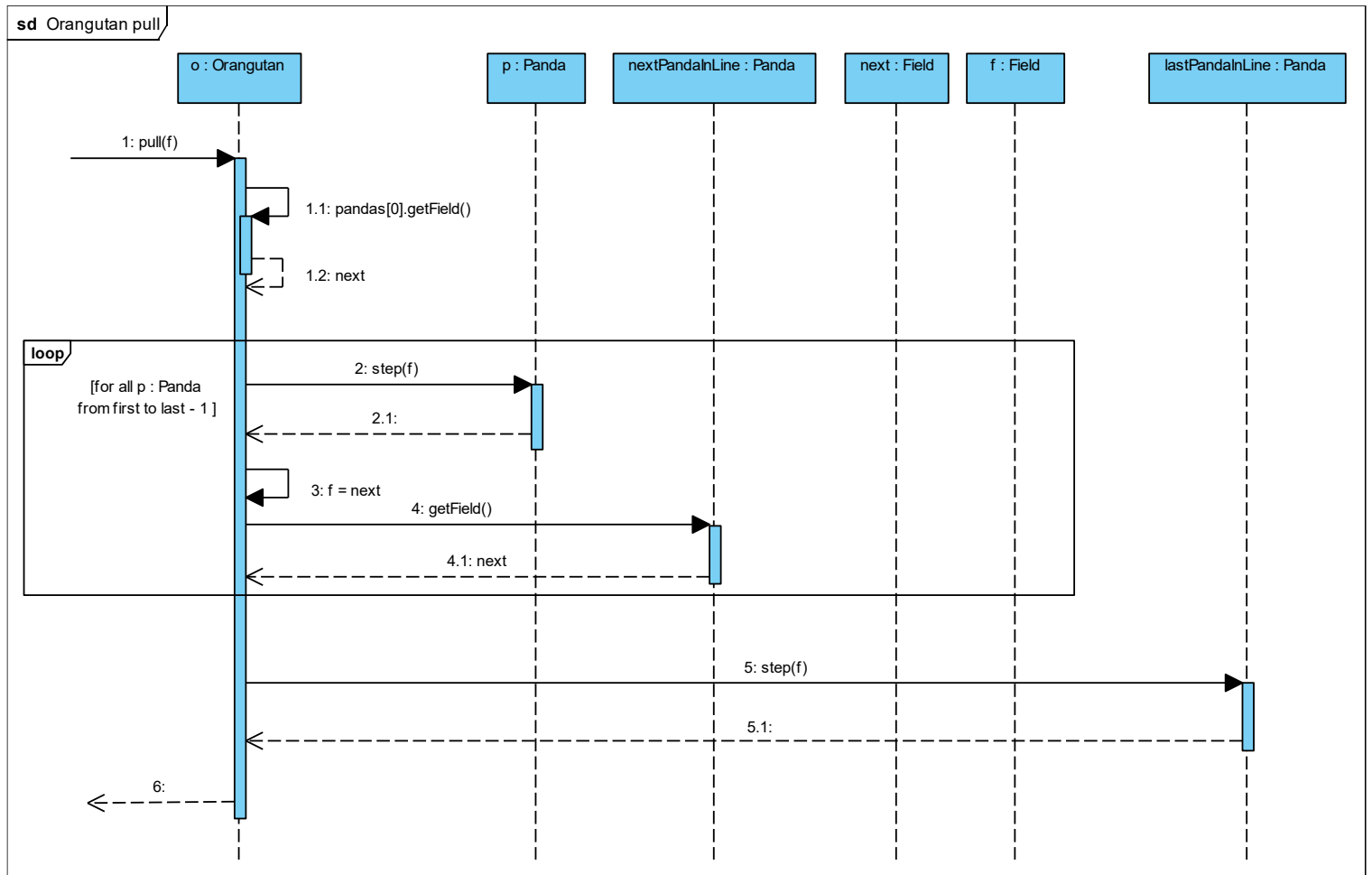
4.4.9 LazyPanda steppenOnWeakTile



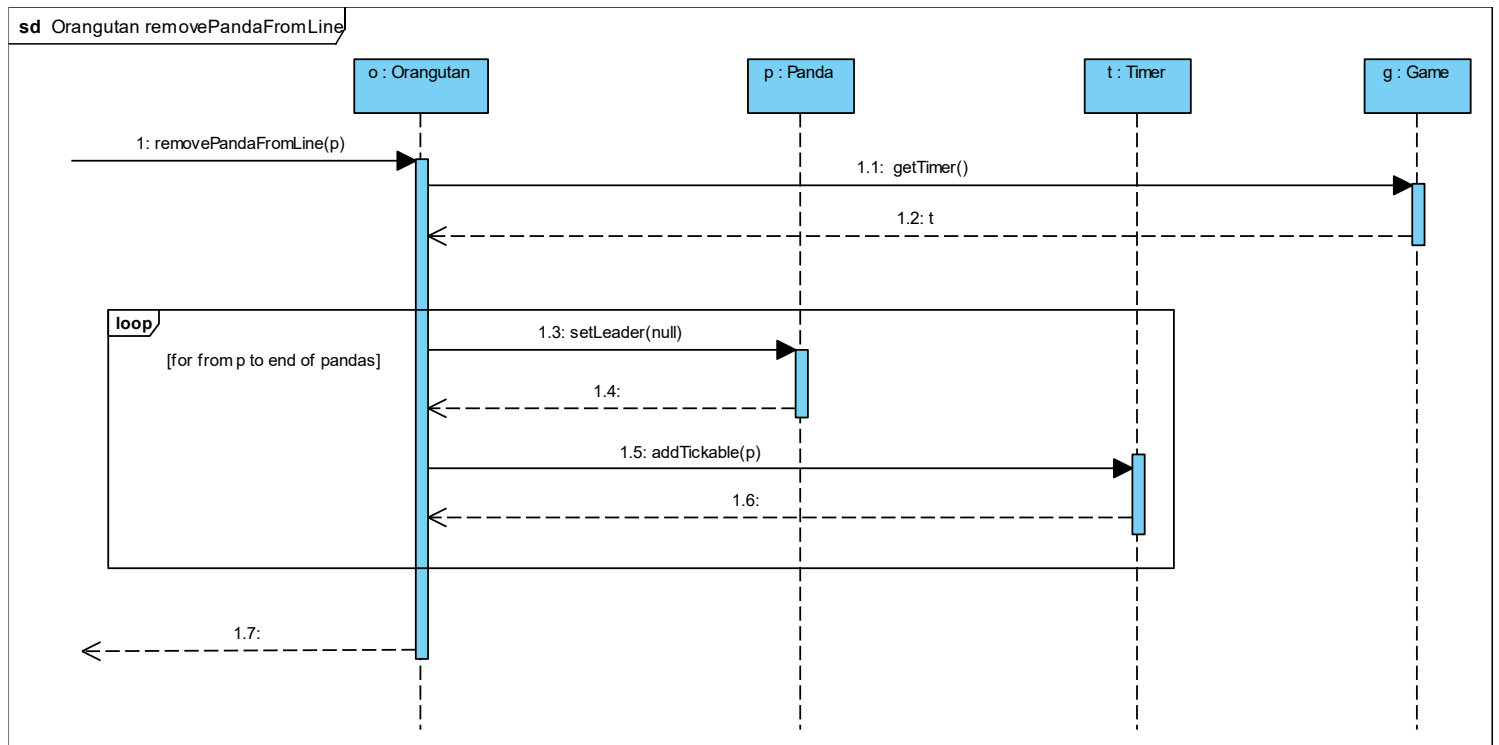
4.4.10 Orangutan dies



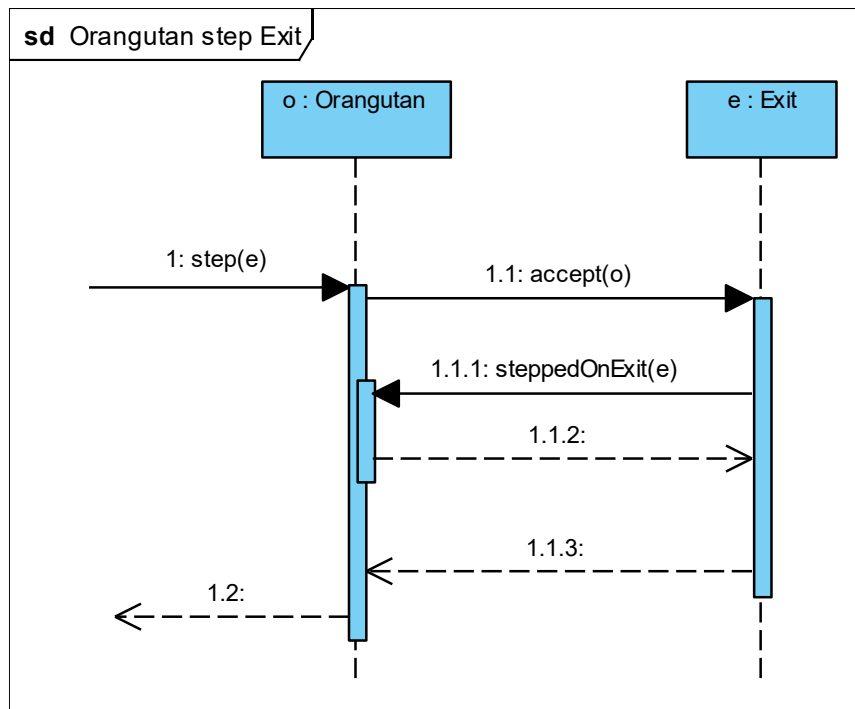
4.4.11 Orangutan pull



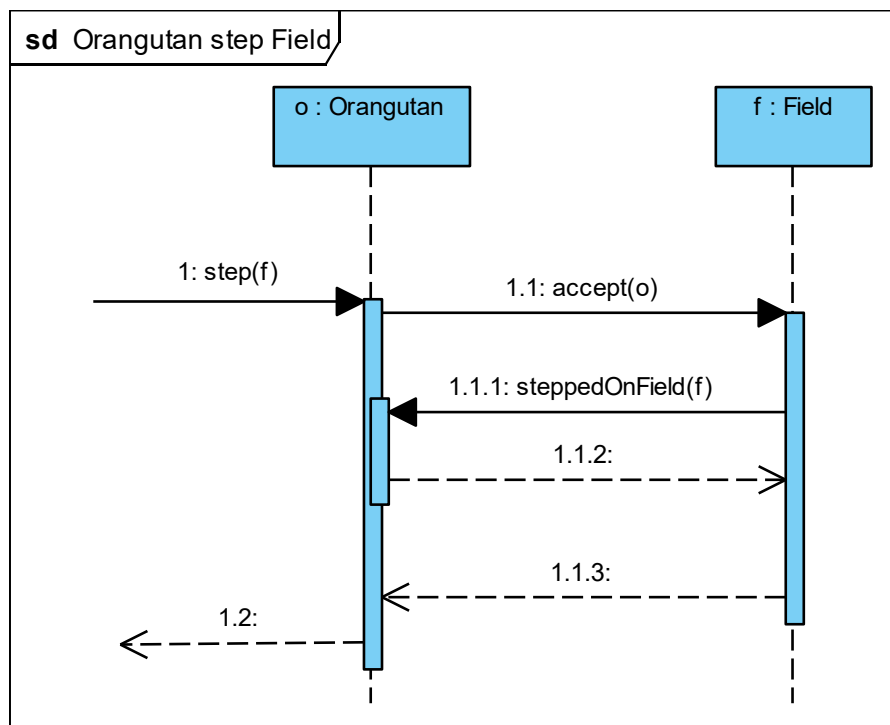
4.4.12 Orangutan removePandaFromLine



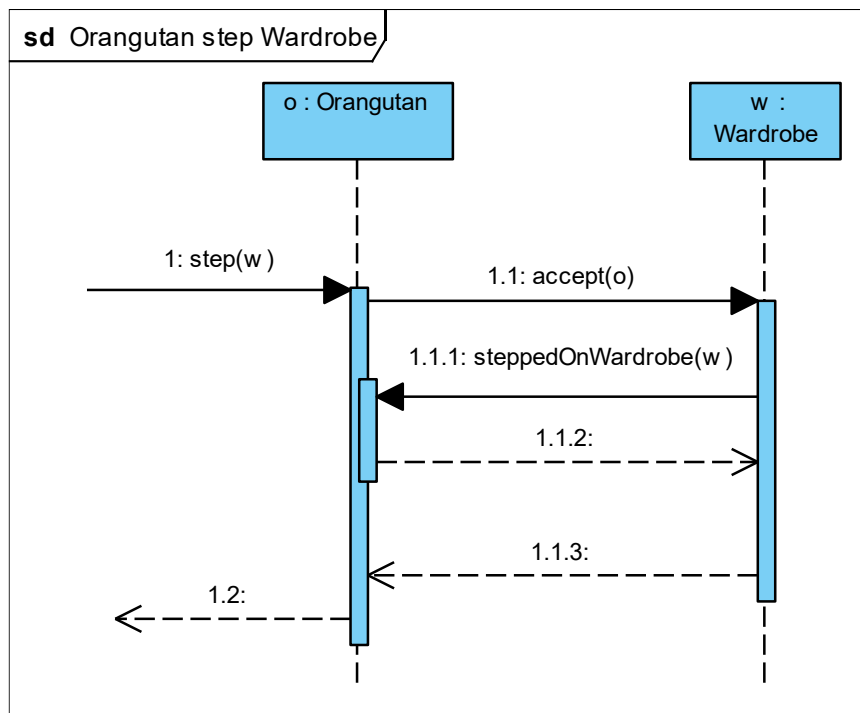
4.4.13 Orangutan step Exit



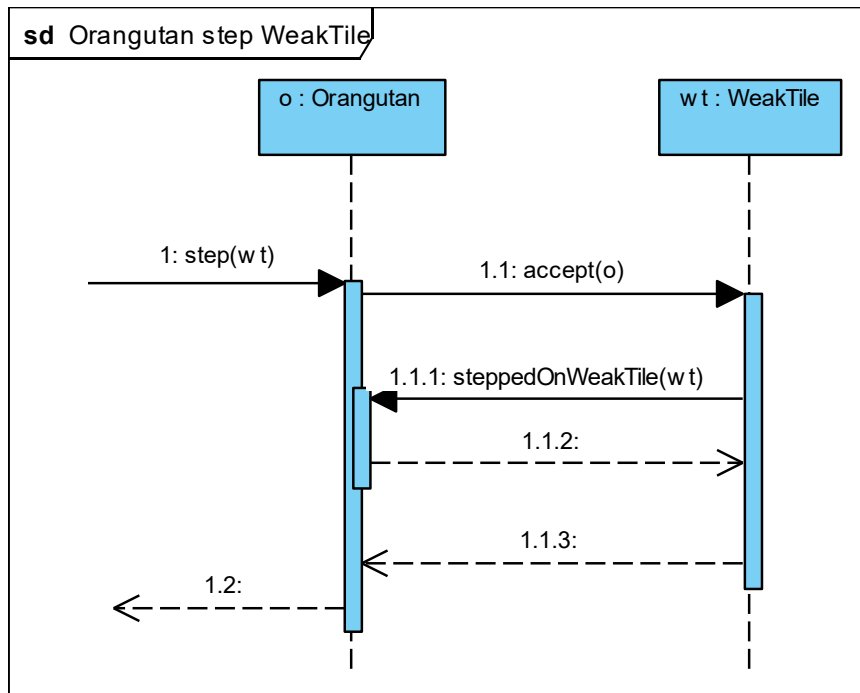
4.4.14 Orangutan step Field



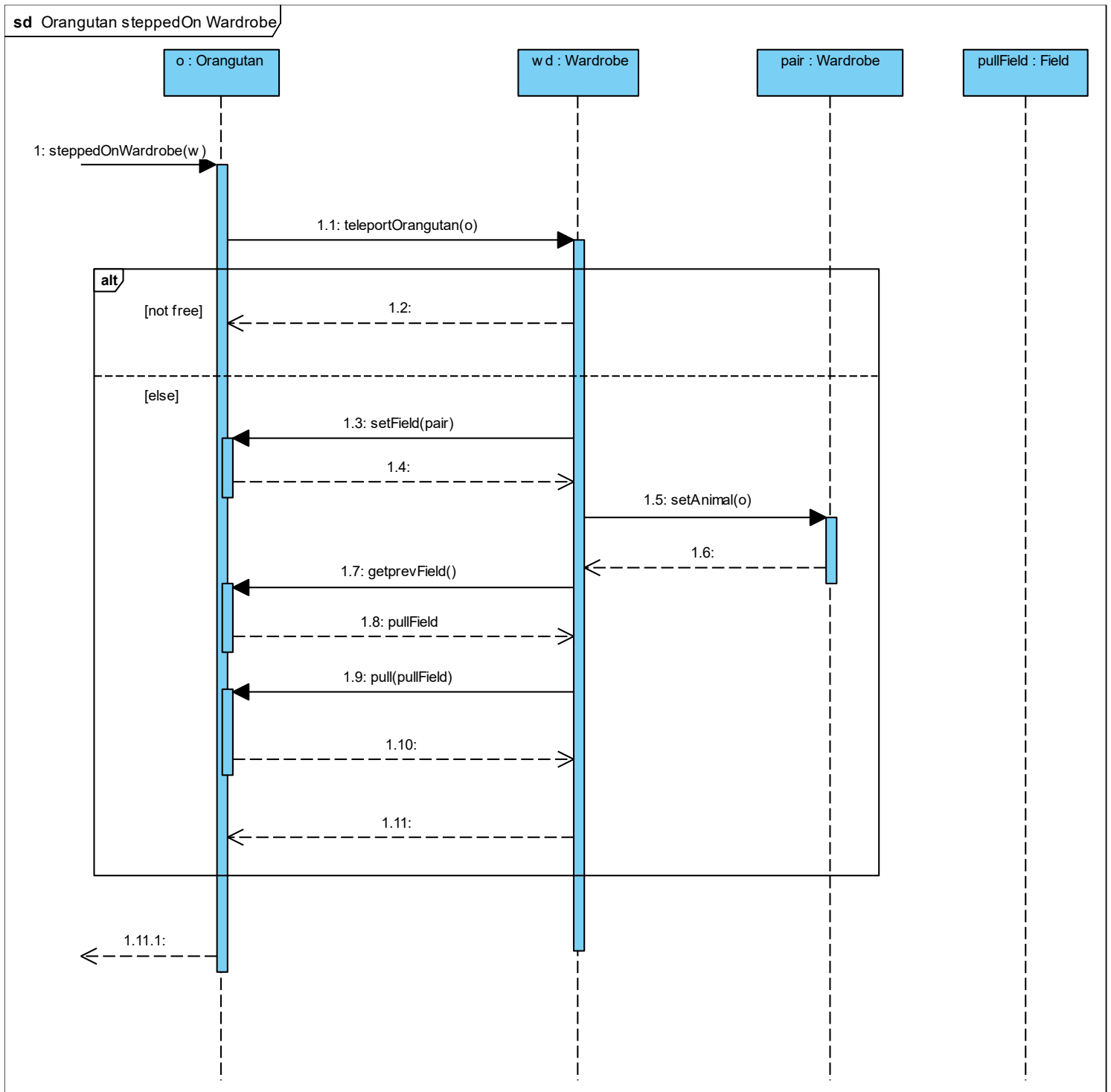
4.4.15 Orangutan step Wardrobe



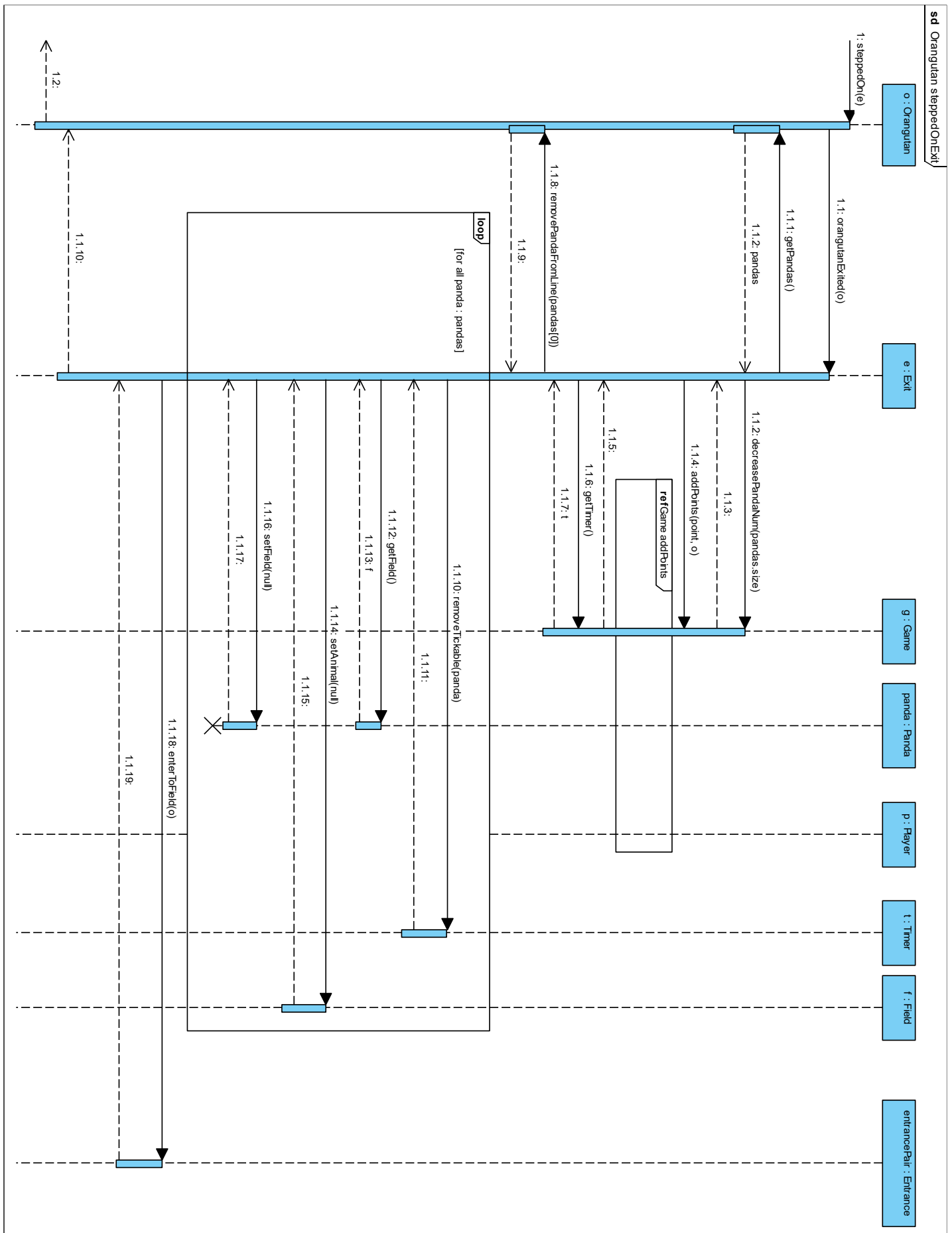
4.4.16 Orangutan step WeakTile



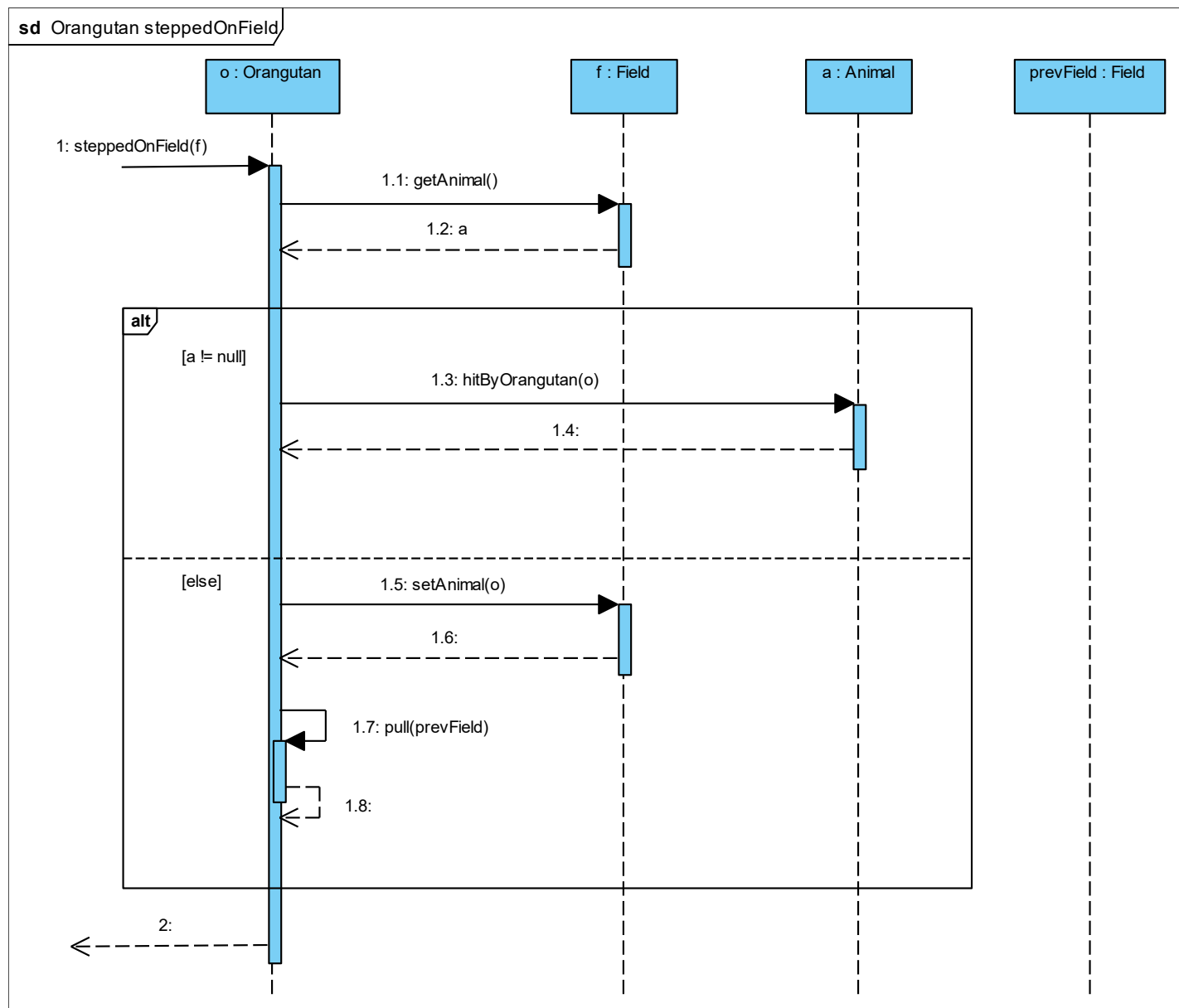
4.4.17 Orangutan steppedOn Wardrobe



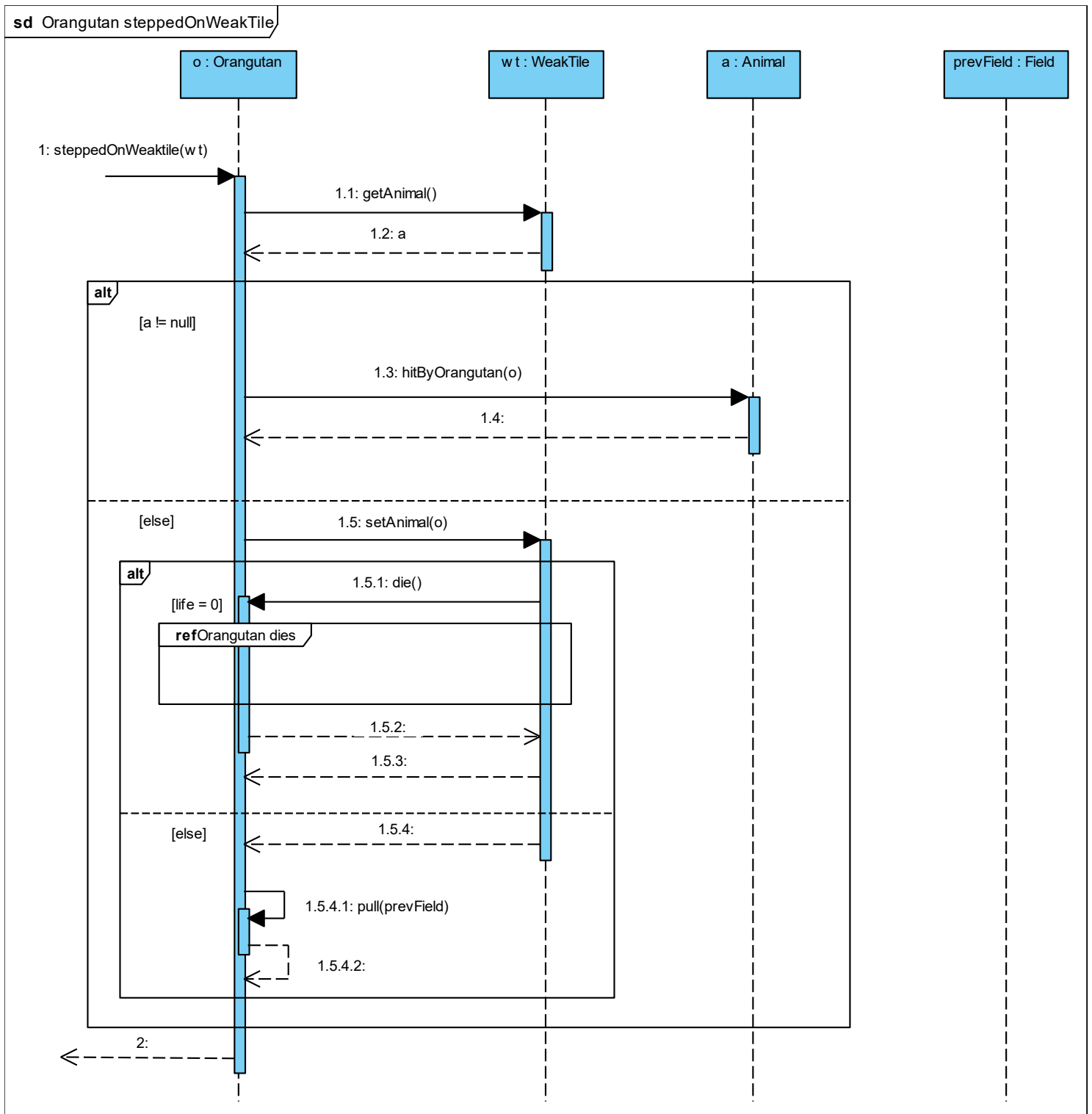
4.4.18 Orangutan steppedOnExit



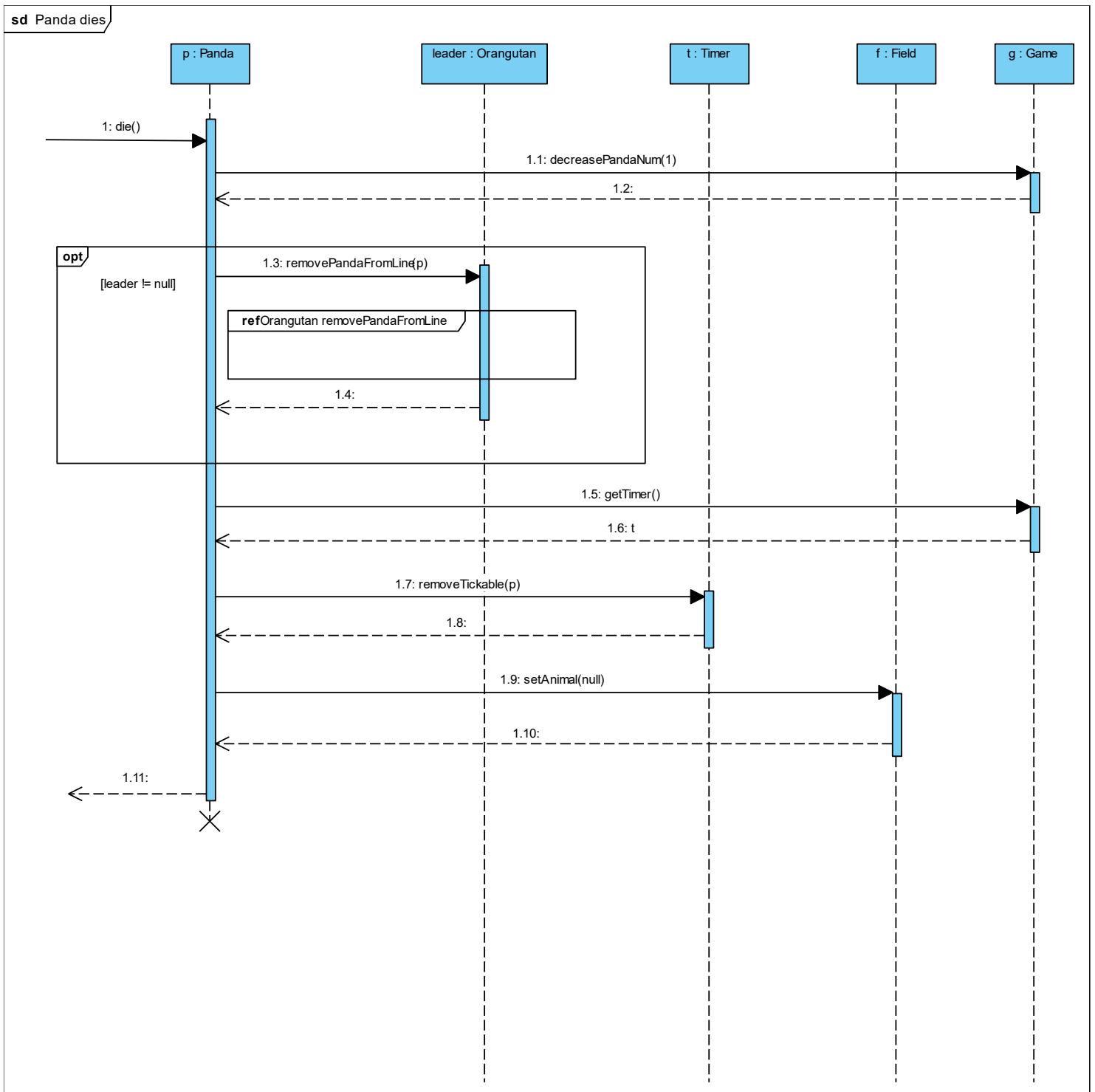
4.4.19 Orangutan steppedOnField



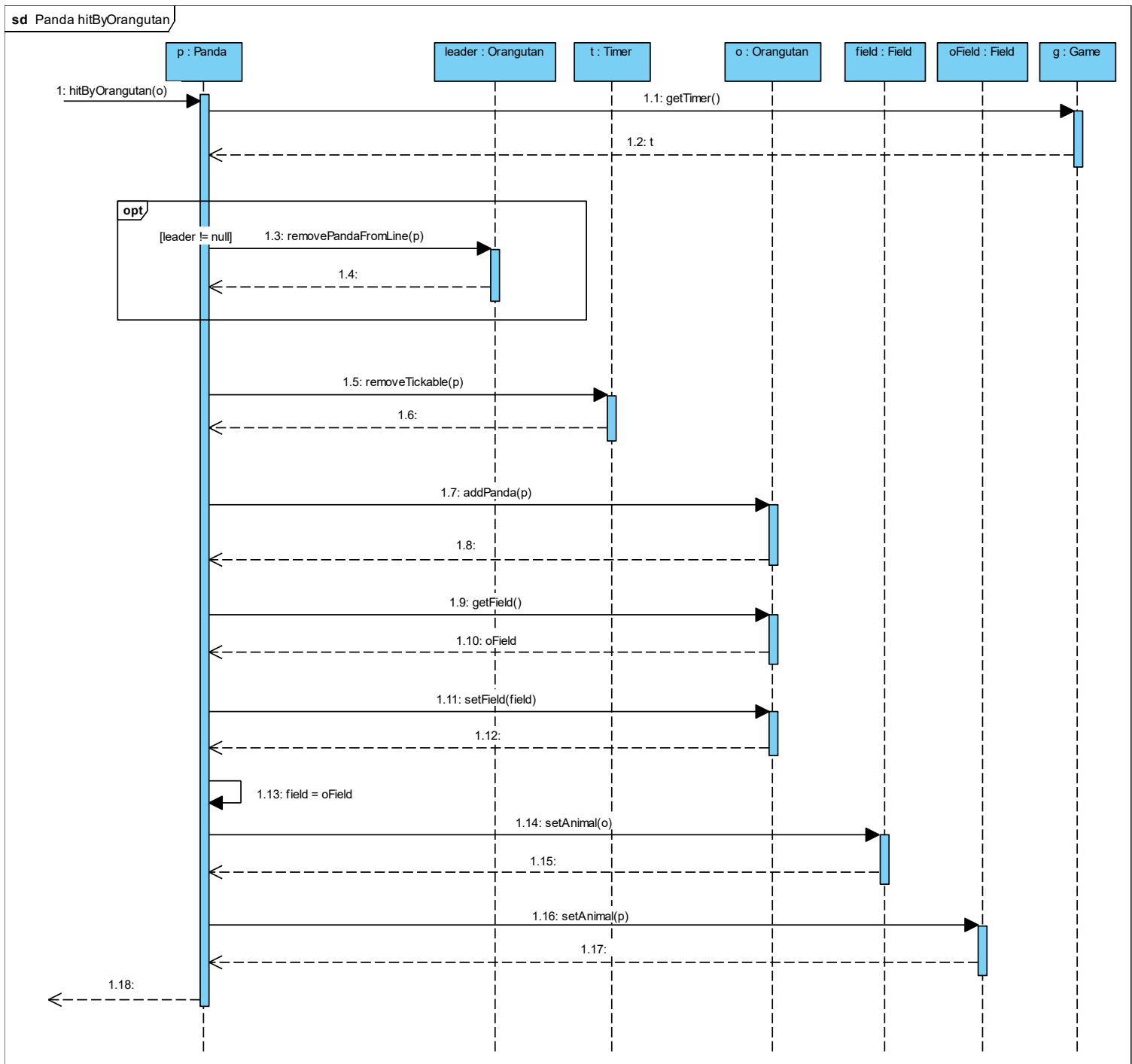
4.4.20 Orangutan steppedOnWeakTile



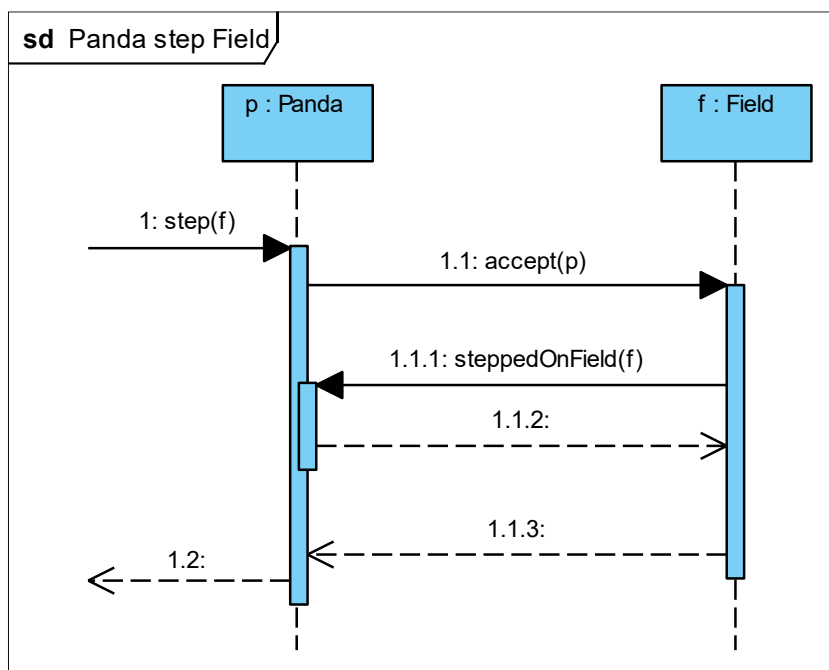
4.4.21 Panda dies



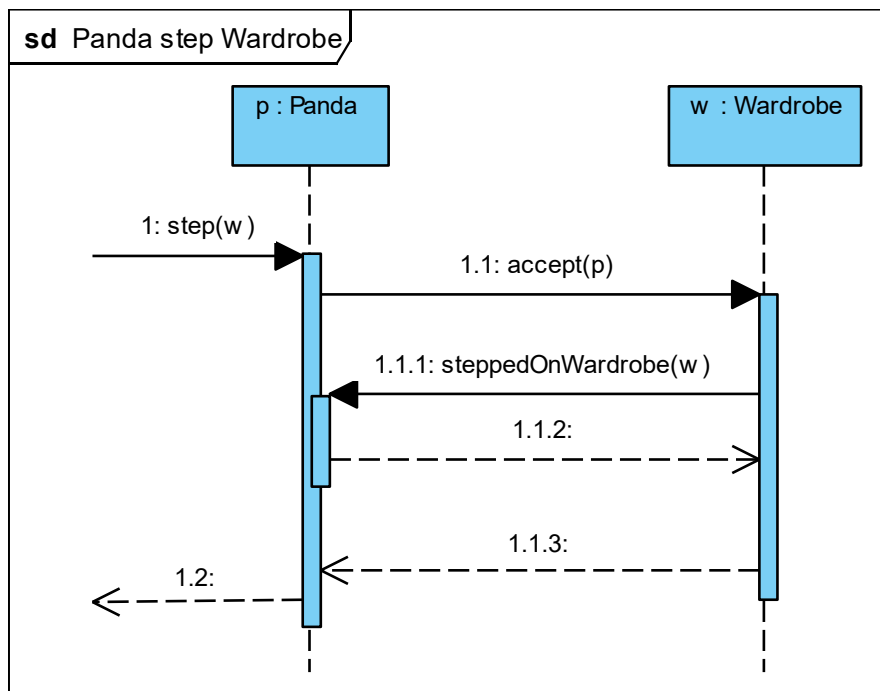
4.4.22 Panda hitByOrangutan

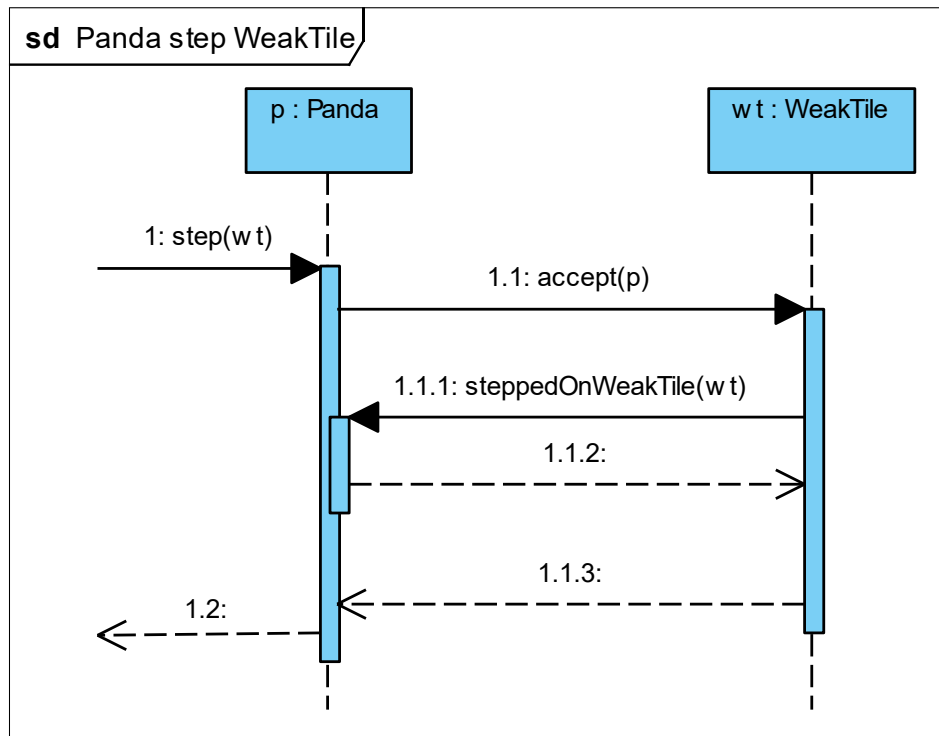


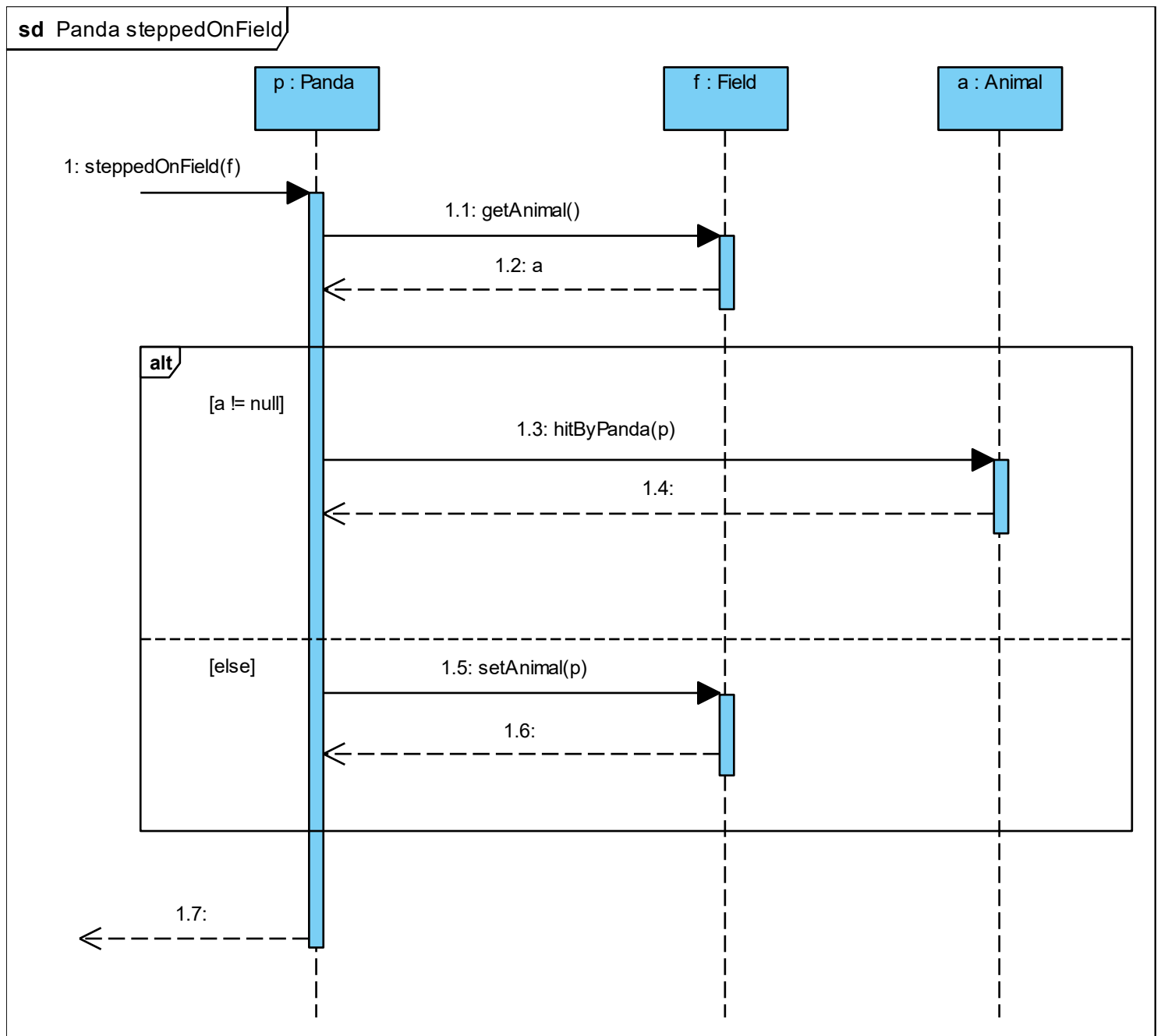
4.4.23 Panda step Field

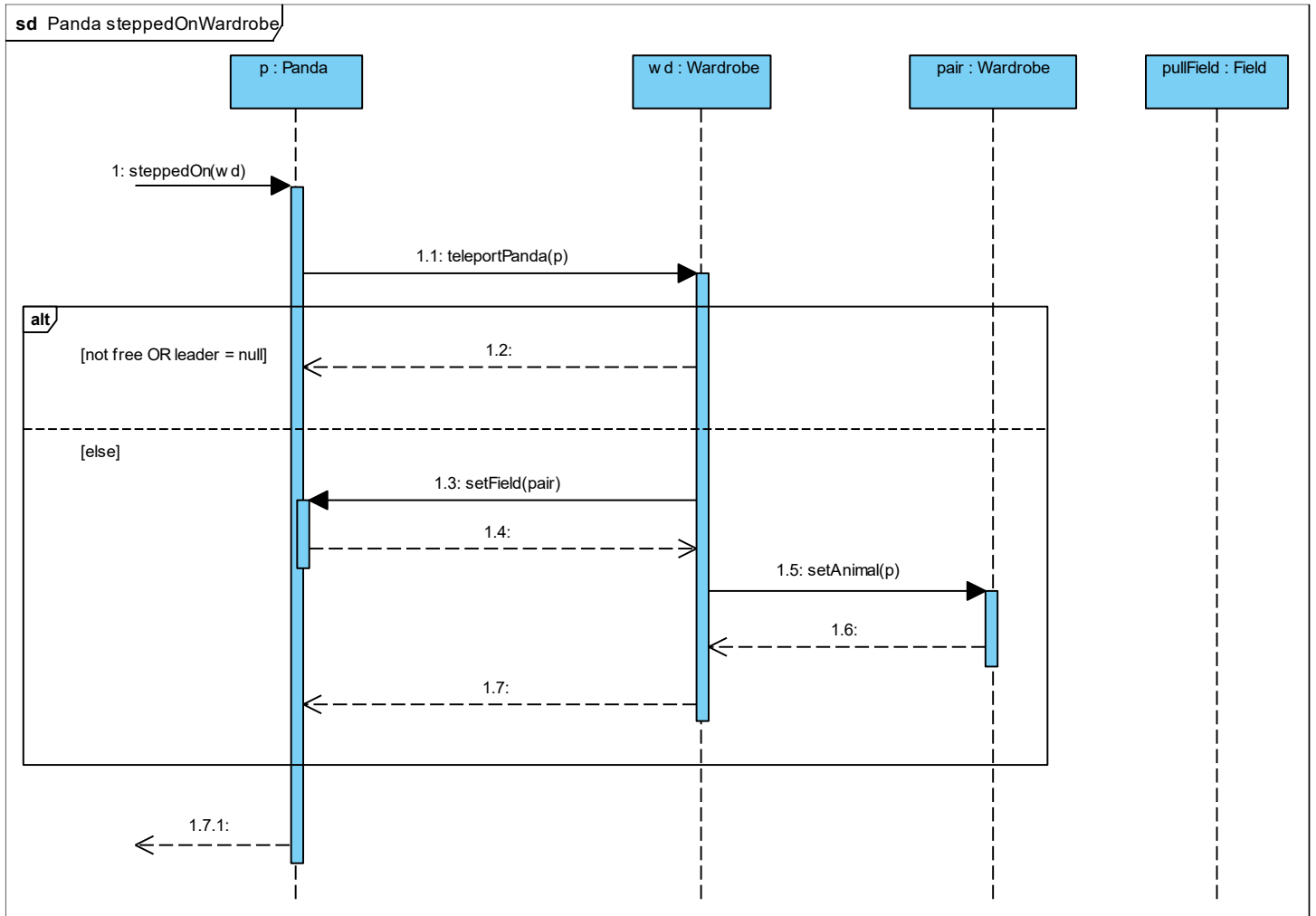


4.4.24 Panda step Wardrobe

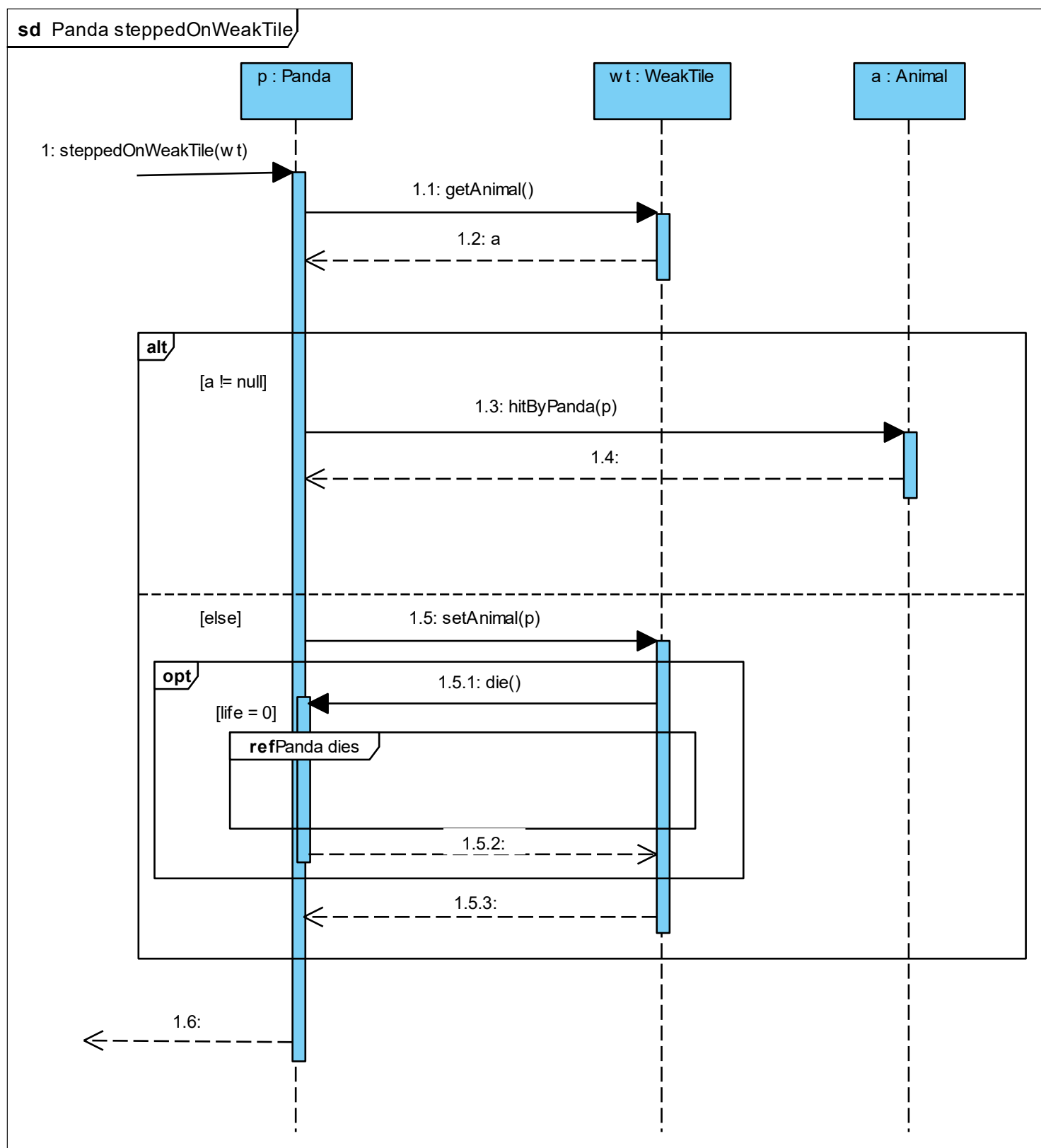


4.4.25 Panda step WeakTile

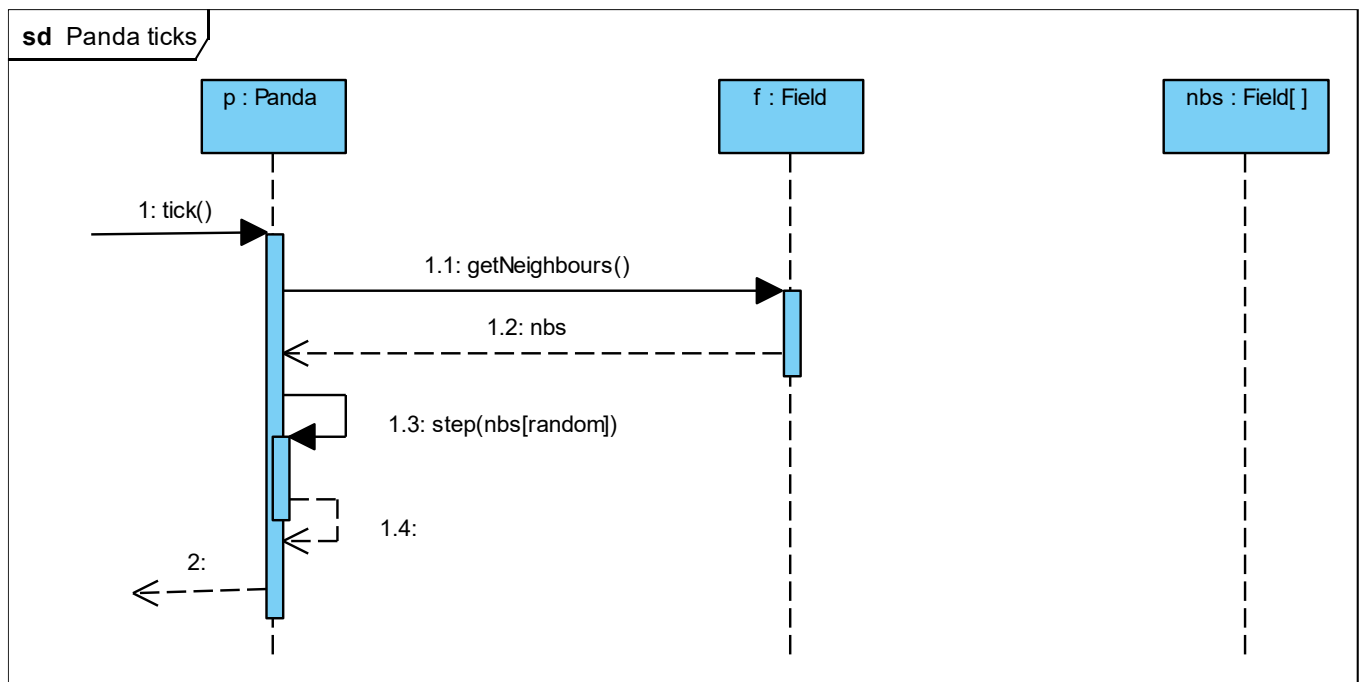
4.4.26 Panda steppedOnField

4.4.27 Panda steppedOnWardrobe

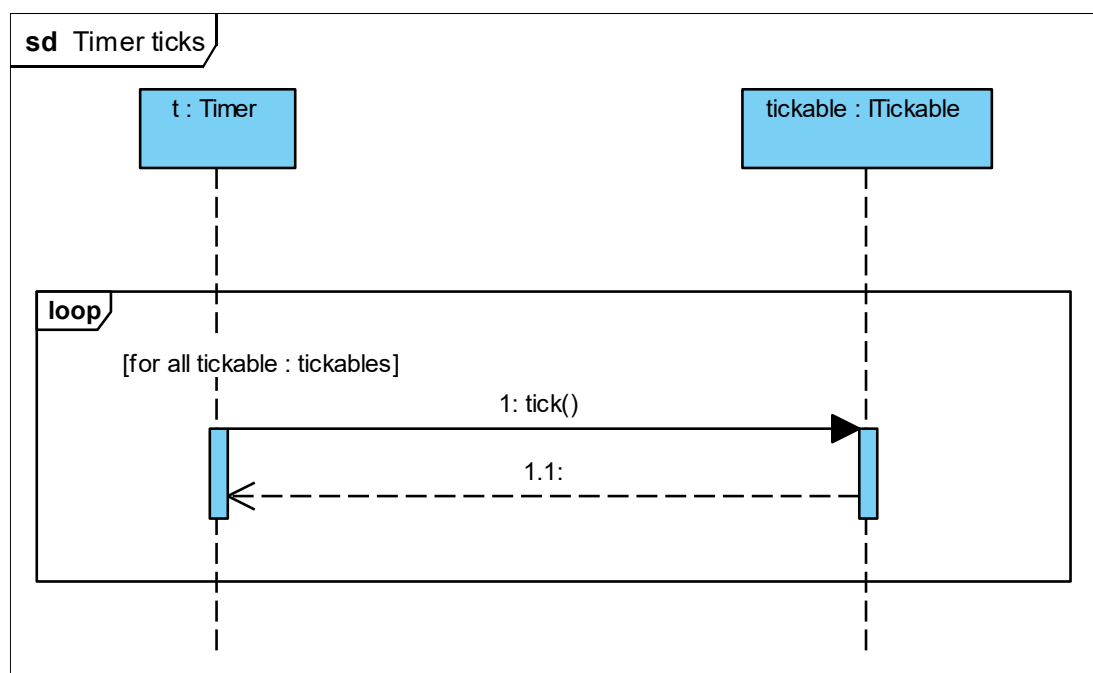
4.4.28 Panda steppedOnWeakTile



4.4.29 Panda ticks

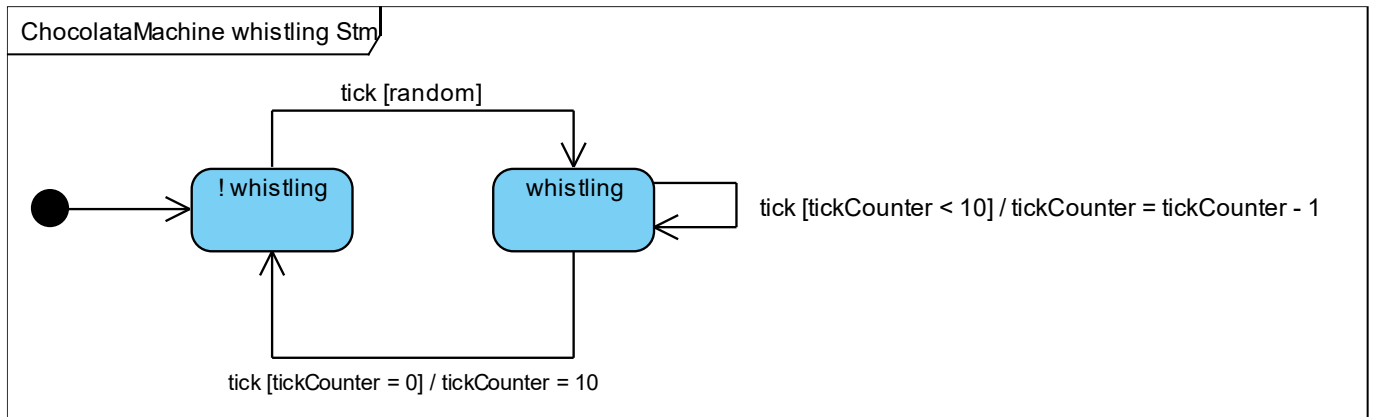


4.4.30 Timer ticks

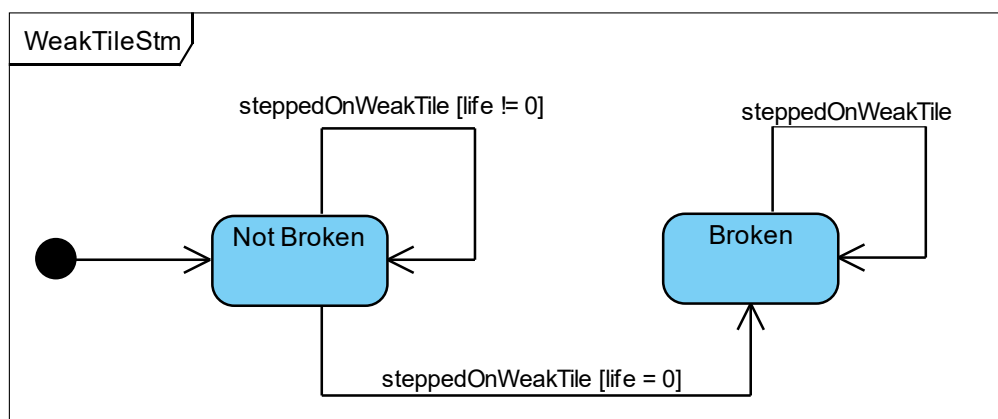


4.5 State-chartok

4.5.1 ChocolataMachine whistling Stm



4.5.2 WeakTileStm



4.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2019.02.27 - 2019.02.28. 11:00 – 01:30	2 óra 30 perc	Hulej Kaszala Kovács	Osztálydiagram javítása
2019.02.28. 13:00 – 16:00	3 óra	Horesnyi Hulej Kaszala Kovács Lőrincz	Értekezlet: Szekvenciadiagramok javítása.
2019.02.28. 20:30 – 00:00	3 óra 30 perc	Horesnyi Kaszala Kovács Lőrincz	Szekvenciadiagramok javítása.
2019.03.02. 13:30 – 15:30	2 óra	Lőrincz	Osztályleírások frissítése, kisebb javítások szekvenciákon
2019.03.02. 16:00 – 17:00	1 óra	Kovács	Szekvencia és állapot diagramok beillesztése a dokumentumba.