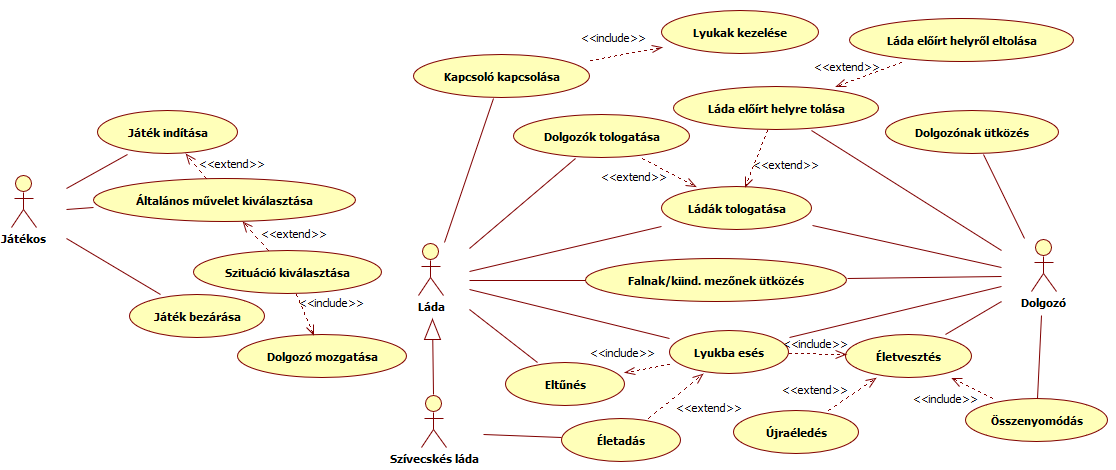
# Szkeleton tervezése

## A szkeleton modell valóságos use-case-ei

### Use-case diagram



### Use-case leírások

|  |  |
| --- | --- |
| **Use-case neve** | Falnak vagy kiindulási mezőnek ütközés |
| **Rövid leírás** | Az aktor falnak vagy kiindulási mezőnek ütközik. |
| **Aktorok** | Dolgozó, Láda |
| **Forgatókönyv** | 1. A dolgozó falra vagy nem saját kiindulási mezőre próbál lépni, ekkor ez nem sikerül neki, helyben marad. 2. A ládát falnak vagy kiindulási mezőre próbálják tolni, ekkor ez nem sikeres, helyben marad. |

|  |  |
| --- | --- |
| **Use-case neve** | Életvesztés |
| **Rövid leírás** | Valamilyen hatás miatt a dolgozó életet veszít. |
| **Aktorok** | Dolgozó |
| **Forgatókönyv** | 1. Ha a dolgozónak nem ez volt az utolsó élete, akkor ezután eggyel kevesebb élete lesz és átkerül a kiindulási mezőjére. 2. Ha a dolgozónak ez volt az utolsó élete, akkor ezután nulla élete lesz, lekerül a pályáról és ebben a játékban már nem vesz többé részt. |

|  |  |
| --- | --- |
| **Use-case neve** | Dolgozónak ütközés |
| **Rövid leírás** | A dolgozó másik dolgozónak ütközik. |
| **Aktorok** | Dolgozó |
| **Forgatókönyv** | 1. A dolgozó közvetlenül olyan mezőre próbál lépni, ahol egy másik dolgozó áll, ekkor a lépés sikertelen és a dolgozó helyben marad. |

|  |  |
| --- | --- |
| **Use-case neve** | Játék indítása |
| **Rövid leírás** | A játékos elindítja a játékot. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | 1. A játékos elindítja a játékot. |

|  |  |
| --- | --- |
| **Use-case neve** | Általános művelet kiválasztása |
| **Rövid leírás** | A játékos kiválaszt egy általános műveletet. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | 1. A játékos a menüből kiválaszt egy általános műveletet. |

|  |  |
| --- | --- |
| **Use-case neve** | Szituáció kiválasztása |
| **Rövid leírás** | A játékos kiválaszt egy szituációt. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | 1. A játékos kiválaszt egy szituációt az általános műveleten belül. |

|  |  |
| --- | --- |
| **Use-case neve** | Dolgozó mozgatása |
| **Rövid leírás** | A játékos mozgat egy dolgozót. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | 1. A játékos a kiválasztott szituációban mozgatja a hozzá tartozó dolgozót. |

|  |  |
| --- | --- |
| **Use-case neve** | Játék bezárása |
| **Rövid leírás** | A játékos bezárja a játékot. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | 1. A játékos bezárja a játékot. |

|  |  |
| --- | --- |
| **Use-case neve** | Kapcsoló kapcsolása |
| **Rövid leírás** | Egy láda mozgásával a kapcsoló állapotot vált (aktív/inaktív). |
| **Aktorok** | Láda |
| **Forgatókönyv** | 1. A kapcsoló mezőre egy láda kerül, így a kapcsoló aktív állapotba kerül. 2. A kapcsoló mezőről lekerül az eddig rajta lévő láda, így a kapcsoló inaktív állapotba kerül. |

|  |  |
| --- | --- |
| **Use-case neve** | Dolgozók tologatása |
| **Rövid leírás** | Az aktor egy dolgozót egy szomszédos mezőre tol. |
| **Aktorok** | Láda, Dolgozó |
| **Forgatókönyv** | 1. A dolgozó megfelelő irányban lévő szomszéd mezője üres, ilyenkor egy láda egyszerűen megtolja a dolgozót, aki ennek hatására a szomszédos mezőre kerül.    1. A dolgozó megfelelő irányban lévő szomszéd mezője már foglalt, ilyenkor az aktor megtolja a dolgozót, aki ennek hatására megtolja a szomszédos mezőn lévő entitást. 2. A dolgozó megfelelő irányban lévő szomszéd mezője üres, ilyenkor egy közvetve tolt dolgozó egyszerűen megtolja a dolgozót, aki ennek hatására a szomszédos mezőre kerül.    1. A dolgozó megfelelő irányban lévő szomszéd mezője már foglalt, ilyenkor az aktor megtolja a dolgozót, aki ennek hatására megtolja a szomszédos mezőn lévő entitást. |

|  |  |
| --- | --- |
| **Use-case neve** | Ládák tologatása |
| **Rövid leírás** | Az aktor egy ládát egy szomszédos mezőre tol. |
| **Aktorok** | Láda, Dolgozó |
| **Forgatókönyv** | 1. A láda megfelelő irányban lévő szomszéd mezője üres, ilyenkor az aktor egyszerűen megtolja a ládát, ami ennek hatására a szomszédos mezőre kerül. 2. A láda megfelelő irányban lévő szomszéd mezője már foglalt, ilyenkor az aktor megtolja a ládát, ami ennek hatására megtolja a szomszédos mezőn lévő entitást. |

|  |  |
| --- | --- |
| **Use-case neve** | Lyukba esés |
| **Rövid leírás** | Az aktor egy aktív lyuk mezőre kerül. |
| **Aktorok** | Láda, Dolgozó |
| **Forgatókönyv** | 1. A láda egy aktív lyuk mezőre kerül, ezáltal beleesik és eltűnik. 2. A dolgozó egy aktív lyuk mezőre kerül, ezáltal beleesik és elveszít egy életet. |

|  |  |
| --- | --- |
| **Use-case neve** | Eltűnés |
| **Rövid leírás** | A Láda eltűnik aktív lyuk mezőre kerülve. |
| **Aktorok** | Láda |
| **Forgatókönyv** | 1. A láda a lyukba esik, így lekerül a játéktérről. |

|  |  |
| --- | --- |
| **Use-case neve** | Életadás |
| **Rövid leírás** | A dolgozó életet kap. |
| **Aktorok** | Szívecskés Láda |
| **Forgatókönyv** | 1. A dolgozó egy szívecskés ládát aktív lyuk mezőre tol, ezáltal kap egy életet. 2. A dolgozó egy láncban lévő szívecskés ládát aktív lyuk mezőre tol, ezáltal kap egy életet. 3. A dolgozó aktivál egy szívecskés láda alatt lévő lyukat, aminek hatására a láda leesik és a dolgozó kap egy életet. |

|  |  |
| --- | --- |
| **Use-case neve** | Lyukak kezelése |
| **Rövid leírás** | Lyuk aktiválása illetve deaktiválása. |
| **Aktorok** | Láda |
| **Forgatókönyv** | 1. Egy láda egy kapcsolóra kerül, aminek hatására a kapcsolóhoz tartozó lyukak aktívvá válnak. 2. Egy láda lekerül egy kapcsolóról, aminek hatására a kapcsolóhoz tartozó lyukak inaktívvá válnak. |

|  |  |
| --- | --- |
| **Use-case neve** | Újraéledés |
| **Rövid leírás** | A dolgozó életvesztést követően újraéled. |
| **Aktorok** | Dolgozó |
| **Forgatókönyv** | 1. A dolgozó életet veszít, és ha nem az volt az utolsó élete, újraéled és a saját kiindulási mezőjére kerül. |

|  |  |
| --- | --- |
| **Use-case neve** | Összenyomódás |
| **Rövid leírás** | A dolgozót összenyomják, aminek hatására életet veszít. |
| **Aktorok** | Dolgozó |
| **Forgatókönyv** | 1. A dolgozót falnak tolják, aki ezáltal összenyomódik és életet veszít. 2. A dolgozót nem saját kiindulási mezőre tolják, aki ezáltal összenyomódik és életet veszít. |

|  |  |
| --- | --- |
| **Use-case neve** | Láda előírt helyre tolása |
| **Rövid leírás** | A ládát egy előírt helyre tolják. |
| **Aktorok** | Dolgozó |
| **Forgatókönyv** | 1. A ládát egy dolgozó közvetlenül vagy láncban egy előírt helyre tol, ezáltal pontot szerez. |

|  |  |
| --- | --- |
| **Use-case neve** | Láda előírt helyről eltolása |
| **Rövid leírás** | A ládát eltolják egy előírt helyről. |
| **Aktorok** | Dolgozó |
| **Forgatókönyv** | 1. A ládát egy dolgozó közvetlenül vagy láncban eltol egy előírt helyről, ezáltal a rátolásért pontot kapó játékos pontot veszít. |

## A szkeleton kezelői felületének terve, dialógusok

A szkeleton szoftver alapvetően egy konzolos menürendszer, melyben a fontosabb szenáriók játszhatóak végig. A menüelem általában meghatároz egy általánosabb műveletet (például: Dolgozó mozgatása), ezen belül konkrétabb szituációk választhatók ki. Például:

What do you want to do?

1 - Move Worker

2 - Push Crate

3 - Create Own Sequence (Dev. Mode)

Ha például az első lehetőséget választjuk ki:

More specifically?

 1.1 - Move Worker On Empty Floor

 1.2 - Move Worker On Empty Hole

 1.3 - Move Worker On Empty Target

 1.4 - Move Worker On Empty Switch

 1.5 - Move Worker On Wall

 1.6 - Move Worker On Spawn

 1.7 - Worker Pushes Worker

 1.8 - Worker Pushes Worker In Chain On Wall

 1.9 - Worker Pushes Worker In Chain On Spawn

Esetleg ha a másodikat:

More specifically?

 2.1 - Worker Pushes Crate On Empty Floor

 2.2 - Worker Pushes Crate On Empty Hole

 2.3 - Worker Pushes LifeCrate On Empty Hole

 2.4 - Worker Pushes Crate On Empty Target

 2.5 - Worker Pushes Crate On Empty Switch

 2.6 - Worker Pushes Crate On Wall

 2.7 - Worker Pushes Crate On Spawn

 2.8 - Worker Pushes Crate From Switch

 2.9 - Worker Pushes Crate From Target

Ezek már konkrét szituációk, melyeket kiválasztva láthatjuk a lefutó interakciókat/függvényeket. Például az elsőből az ötödiket kiválasztva megnézhetjük mi történik akkor, ha a munkás megpróbál falra lépni:

1.5 Move Worker On Wall

-> Worker::move(dir):

  -> Entity::step(firstPusher, dir):

        -> Field::getNeighbourField(dir):

        neighbor <- Field::getNeighbourField(dir)

        -> Field::isEmpty():

        true <- Field::isEmpty()

        -> Worker::visit(firstPusher, iv):

               -> Field::visitByWorker(firstPusher, w):

            false <- Field::visitByWorker(firstPusher, w)

        false <- Worker::visit(firstPusher, iv)

    false <- Entity::step(firstPusher, dir)

<- Worker::move(dir)

A szoftvernek van egy ún. developer módja, amikor saját szituációt gyárthatunk. Ekkor minden egyes dolgot magunknak kell kiválasztani. Például azon szituáció legyártása, mikor egy játékos egy másikat próbál közvetlen eltolni (amit nálunk nem lehet):

What direction to move? [L - Left, R - Right, U - Up, D - Down] : R

-> Worker::move(dir):

    -> Entity::step(firstPusher, dir):

        -> Field::getNeighbourField(dir):

            What kind of field is the neighbour? [F - Floor, W - Wall, S - Spawn, T - Target, H - Hole] : F

        neighbor <- Field::getNeighbourField(dir)

        -> Field::isEmpty():

            Is an entity already on this field? [Y - Yes, N - No] : Y

            What kind of entity? [W - Worker, C - Crate, L - LifeCrate] : W

        false <- Field::isEmpty()

       -> Worker::push(firstPusher, pushed, dir):

            -> Worker::pushByWorker(firstPusher, pusher, dir):

            false <- Worker::pushByWorker(firstPusher, pusher, dir)

        false <- Worker::push(firstPusher, pushed, dir)

   false <- Entity::step(firstPusher, dir)

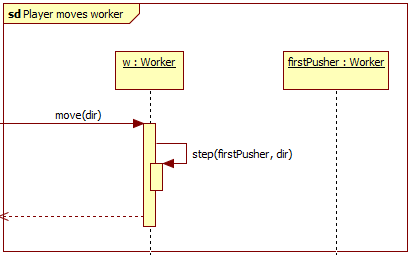
<- Worker::move(dir)

Ekkor a lefutás végén megkapjuk a lefutás közben beírt betű-szekvenciát (jelen esetben ez: RFYW). Az automatikus verzió lényegében önállóan adja a programnak ezt a bemenetet, így ezt kimásolva könnyen adhatunk hozzá új szituációt, lényegében egy új menüelemhez rendeljük hozzá a betűsorozatot.

## Szekvencia diagramok a belső működésre

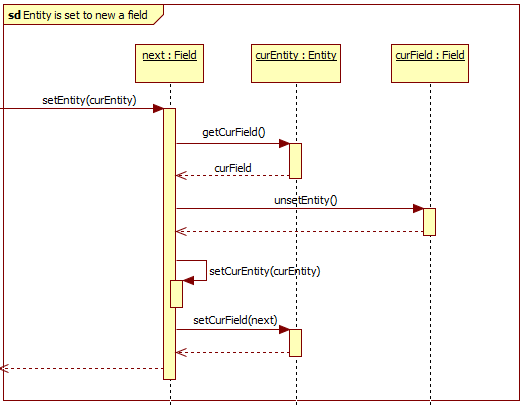
Megjegyzés: Mivel az UML eszköz nem támogatja a szabványos create jelölést, így az metódushívásként van modellezve.

### A játékos lépteti a dolgozót



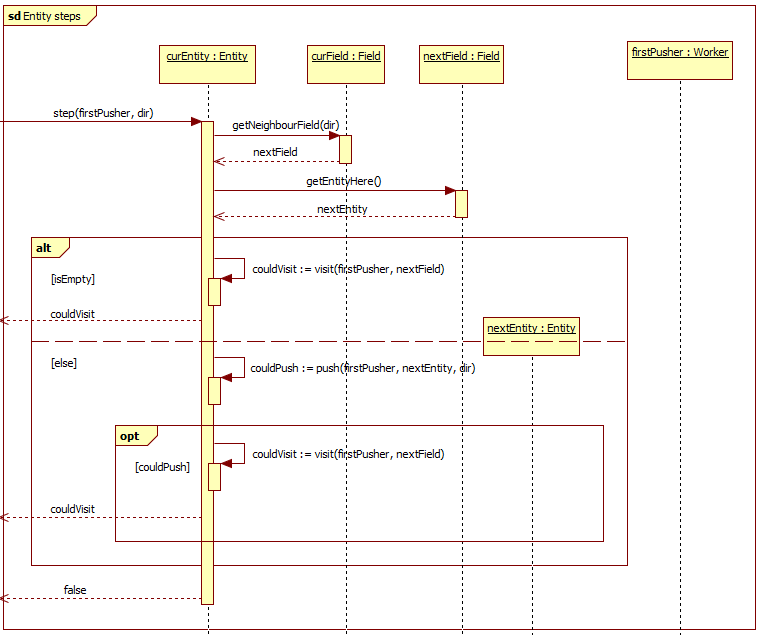
*Ilyenkor megpróbálunk ellépni az adott irányban.*

### Entitás új mezőre lép



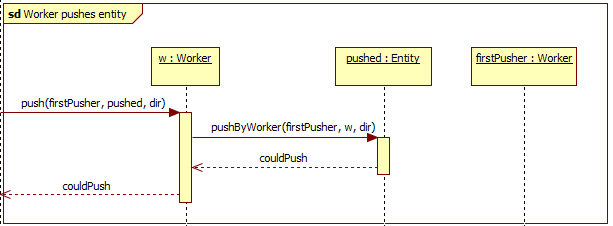
*A belépéskor adminisztráljuk mind az entitás mezőre vonatkozó, mind a mező entitásra vonatkozó referenciájának megváltozását.*

### Entitás lép



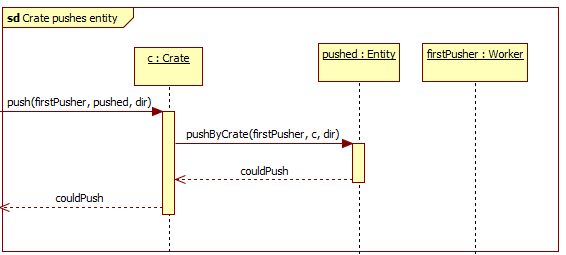
*Léptetéskor két dolgot kell ellenőriznünk. Elsőnek, hogy a célmező egyáltalán képes-e fogadni minket (pl.: Spawn csak a saját játékosát fogadja be), és ha ez fennáll, akkor azt, hogy jelenleg szabad-e, ugyanis, ha másik entitás már ott tartózkodik, akkor meg kell próbálnunk eltolni. Ha sikerült a tolás, vagy eleve üres volt, akkor odalépünk, egyébként helyben maradunk.*

### Dolgozó entitást tol



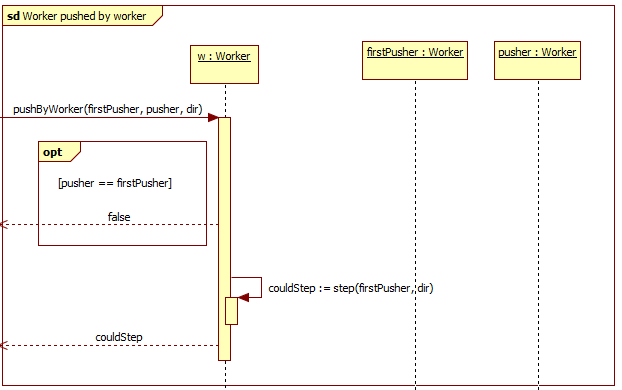
*Mini “Visitor” -> Dinamikussal típussal való hívás miatt szükséges függvény.*

### Láda entitást tol



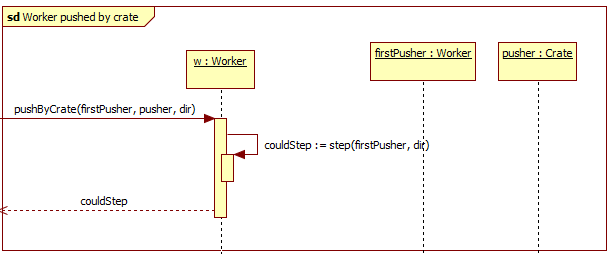
*Mini “Visitor” -> Dinamikussal típussal való hívás miatt szükséges függvény.*

### Dolgozót dolgozó tol



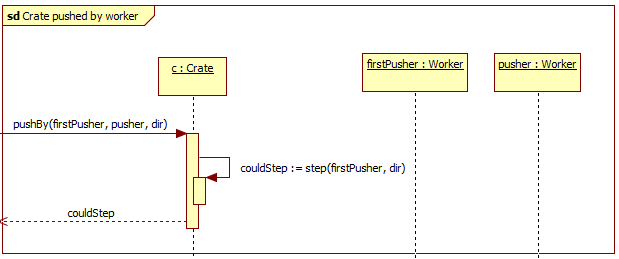
*Dolgozó dolgozót csak akkor tolhat, ha láncba van, ezért ellenőrizzük, hogy aki itt tol az nem az első-e. Ha az első akkor, vissza térünk hammissal, ezzel megtagadva a tolást, ha nem akkor megpróbálunk ellépni a megfelelő irányba és ennek sikerességével térünk vissza.*

### Dolgozót láda tol



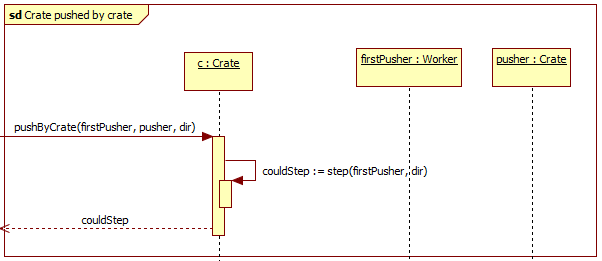
*Mivel láda mindig képes dolgozót tolni, ezért megpróbálunk ellépni a megfelelő irányba és ennek sikerességével térünk vissza.*

### Ládát dolgozó tol



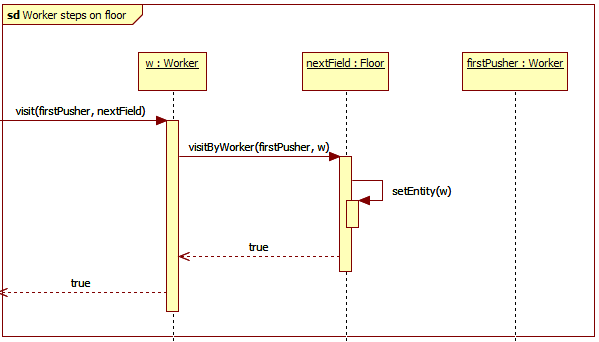
*Mivel dolgozó mindig képes ládát tolni, ezért megpróbálunk ellépni a megfelelő irányba és ennek sikerességével térünk vissza.*

### Ládát láda tol



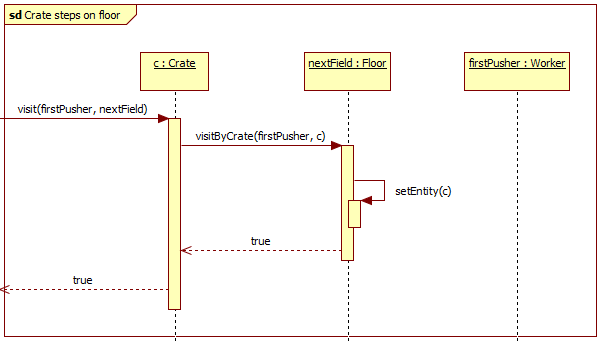
*Mivel láda mindig képes ládát tolni, ezért megpróbálunk ellépni a megfelelő irányba és ennek sikerességével térünk vissza.*

### Dolgozó padlóra lép



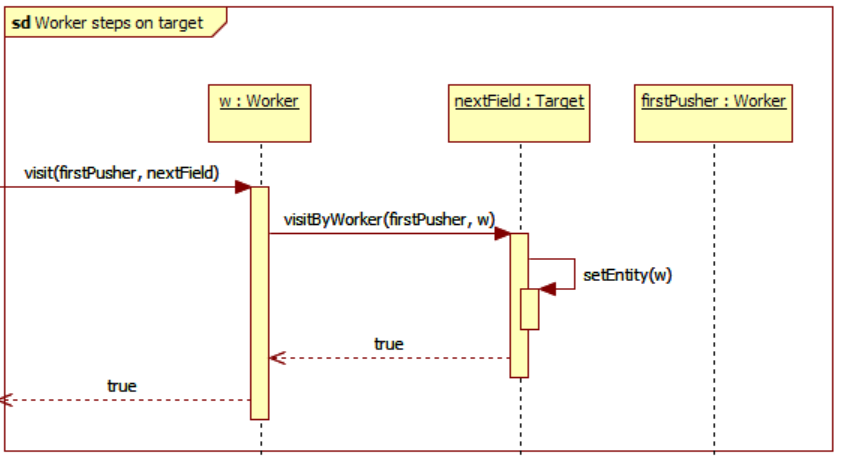
*Egy dolgozó mindig ráléphet egy padlóra.*

### Láda padlóra lép



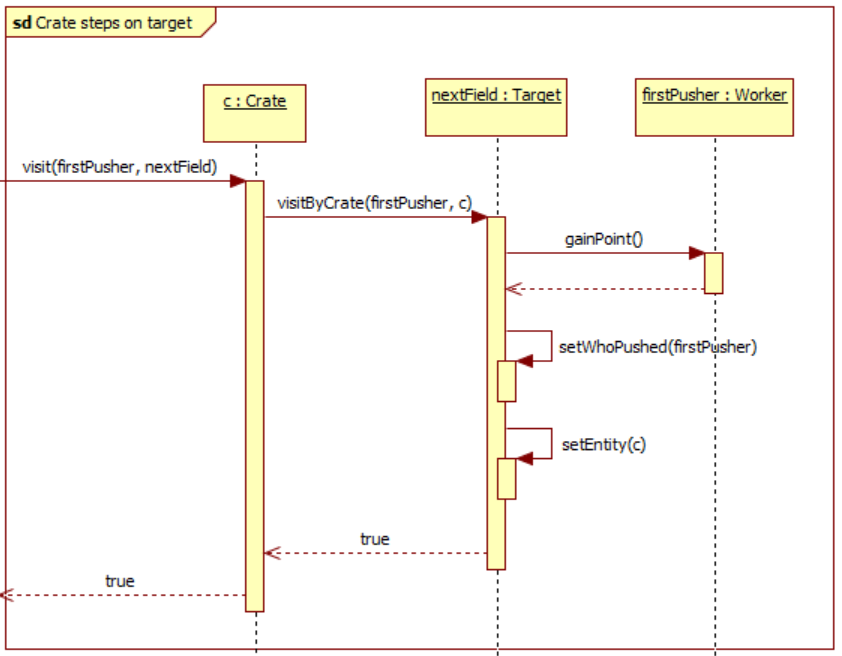
*Egy láda mindig ráléphet egy padlóra.*

### Dolgozó előírt helyre lép



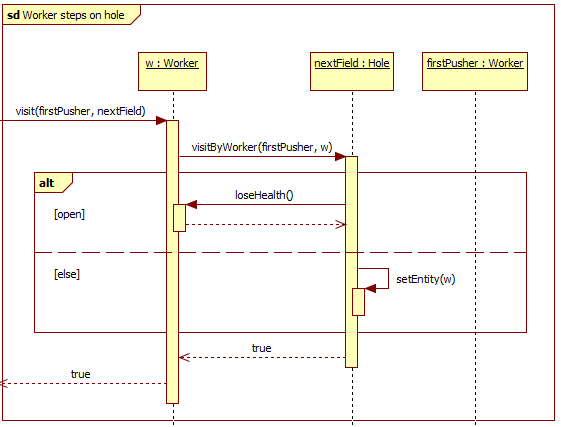
*Egy dolgozó mindig ráléphet egy előírt helyre.*

### Láda előírt helyre lép



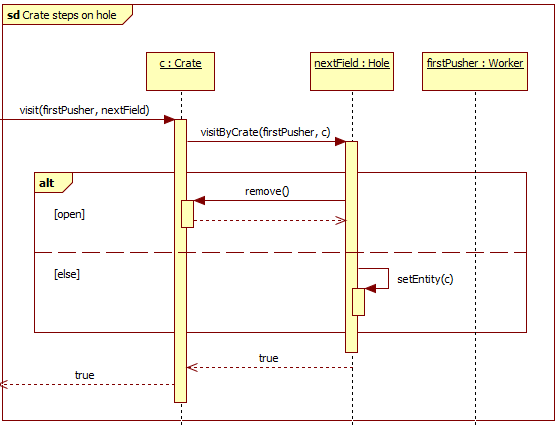
Mikor egy láda  kilép innen, a szokásos adminisztráció során elvesszük a kiosztott pontot. Tehát beléptetéskor szokásos adminisztráción kívül, két dolgot kell tennünk, kiosztani a pontot annak aki miatt idekerült a láda, valamint referenciáját eltárolni, hogy később, ha kell, eltudjuk venni a pontot.

### Dolgozó lyukra lép



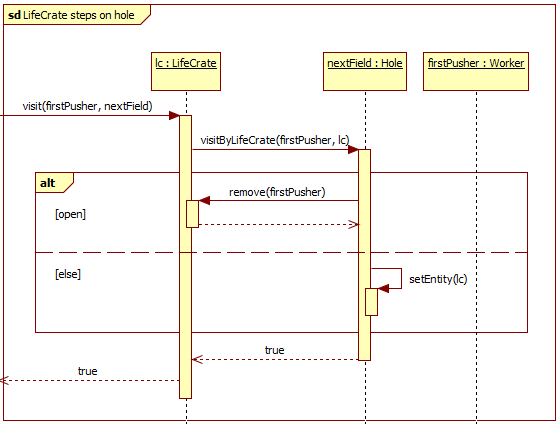
Ekkor ha nyitva van a lyuk akkor a lépés hatására életet kell veszítenie a dolgozónak, ha zárva akkor a szokásos műveleteket kell elvégezni.

### Láda lyukra lép



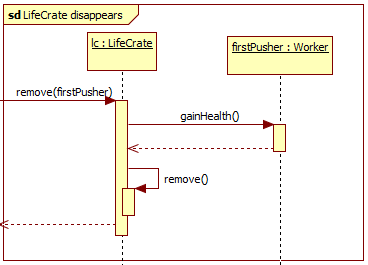
*Ekkor ha nyitva van a lyuk akkor a lépés hatására megsemmisül a láda, ha zárva akkor a szokásos műveleteket kell elvégezni.*

### Szívecskés láda lyukra lép



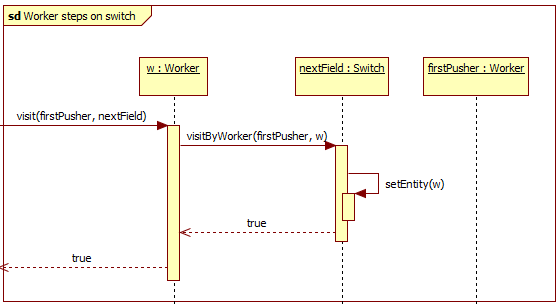
*Ekkor ha nyitva van a lyuk akkor a lépés hatására megsemmisül a láda és aki ezt okozta kap egy életet (remove függvényen belül), ha zárva akkor a szokásos műveleteket kell elvégezni.*

### Szívecskés láda eltűnik



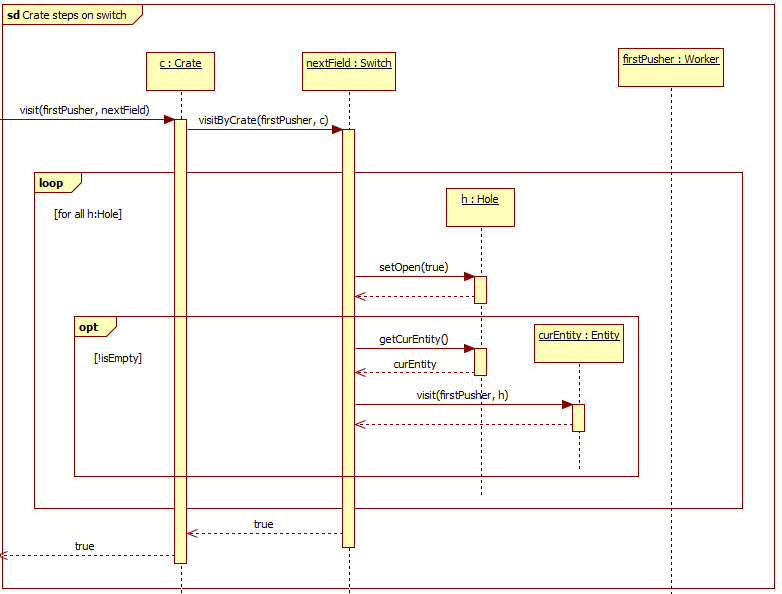
*Szívecskés láda esetén a megsemmisüléskor az azt okozó kap egy életet, majd a szokásos módon eltűnik a láda.*

### Dolgozó kapcsolóra lép



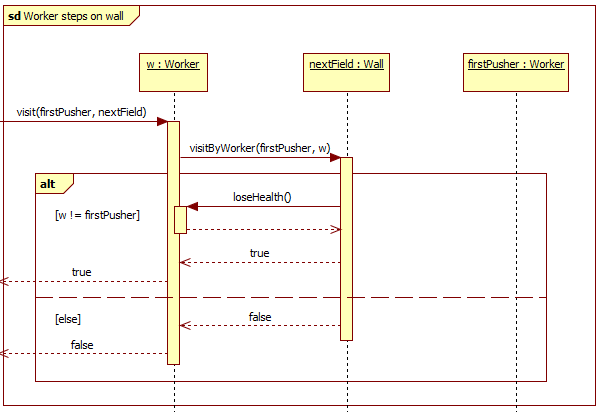
*A dolgozó nem képes kapcsolni a kapcsolót, ezért csak a szokásos lépések történnek.*

### Láda kapcsolóra lép



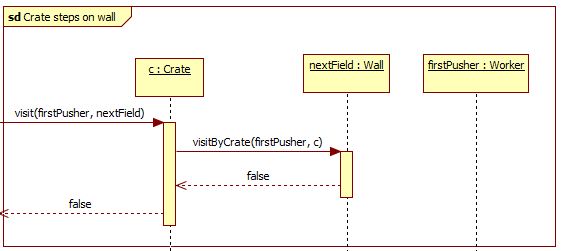
*Ha láda egy kapcsolóra lép, akkor a szokásos adminisztráción kívül, ki kell nyitni az összes hozzá tartozó lyukat, valamint az ott álló entityket le kell ejteni. Mivel a leejtést az a dolgozó okozza, aki a kapcsolóra tolta a ládát, így a leejtés megfelel annak, mintha épp beletolta volna a lyukba az ott álló entitásokat.*

### Dolgozó falra lép



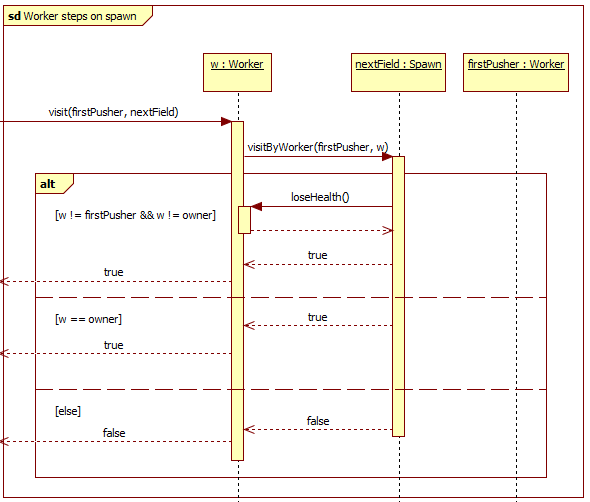
*Egy dolgozó csak akkor léphet falra, ha rátolják, ekkor életet veszít.*

### Láda falra lép



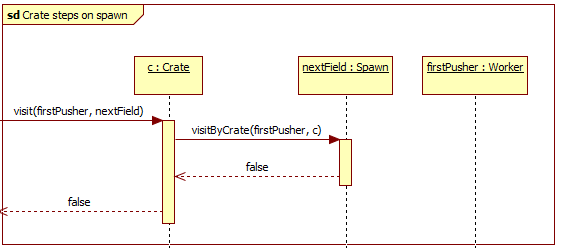
*Egy láda soha nem léphet rá egy falra.*

### Dolgozó kiindulási helyre lép



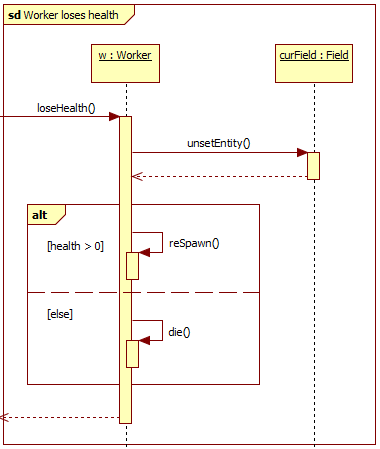
*A spawnra csak a tulajdonosa léphet rá, minden más dolgozó életet veszít a probálkozásért.*

### Láda kiindulási helyre lép



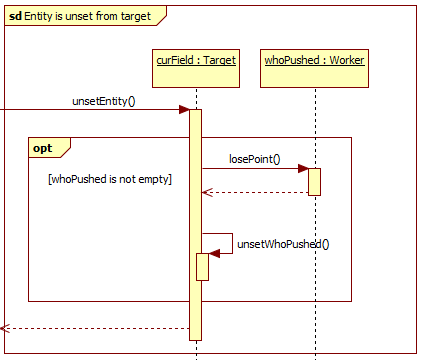
*Egy láda soha nem léphet rá egy kiindulási helyre.*

### Dolgozó életet veszít



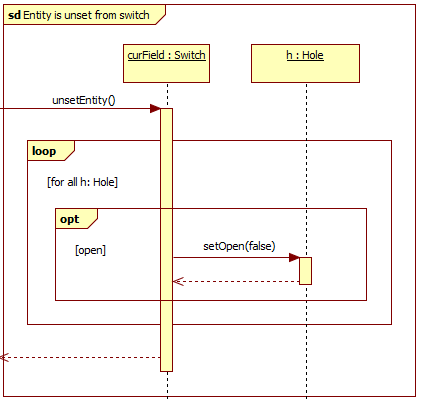
*Ha a dolgozó életet veszít akkor le kell venni a mezőről, ahol állt, csökkenteni az életeinek számát, majd az élet mennyiségének megfelelően vagy újrateremteni, vagy megölni.*

### Entitás lekerül előírt helyről



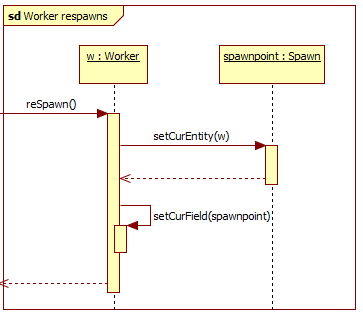
A whoPushed referencia csak akkor van beállítva, ha valaki kapott pontot, mert ládát tolt ide, ezért ebben az esetben, el kell venni tőle ezt a pontot és kinullázni a referenciát.

### Entitás lekerül kapcsolóról



*Ha üressé válik a kapcsoló mező, akkor minden hozzá tartozó lyukat be kell zárnunk.*

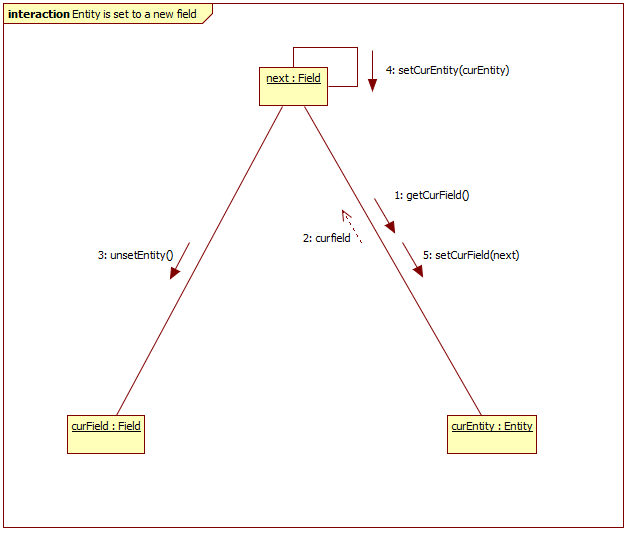
### Dolgozó respawnol



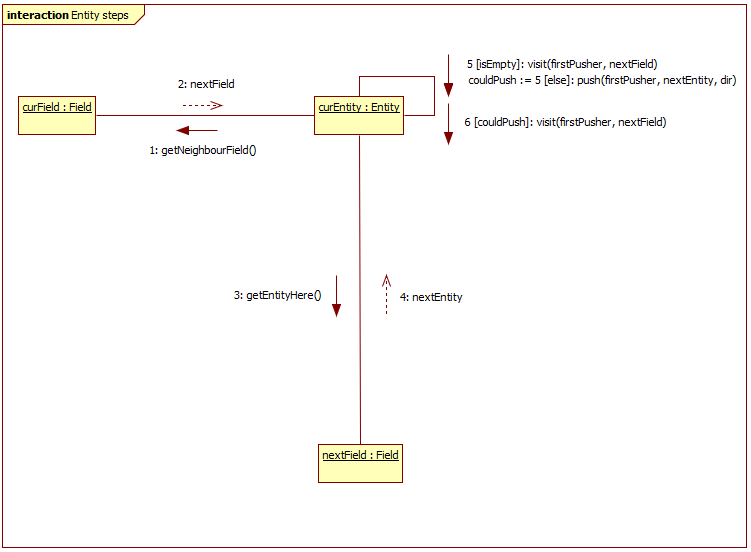
*Újrateremtődés esetén a dolgozót elhelyezzük a spanpointjára, ezt a dolgozó mező és a mező entitás referenciájának állításával tehetjük meg.*

## Kommunikációs diagramok

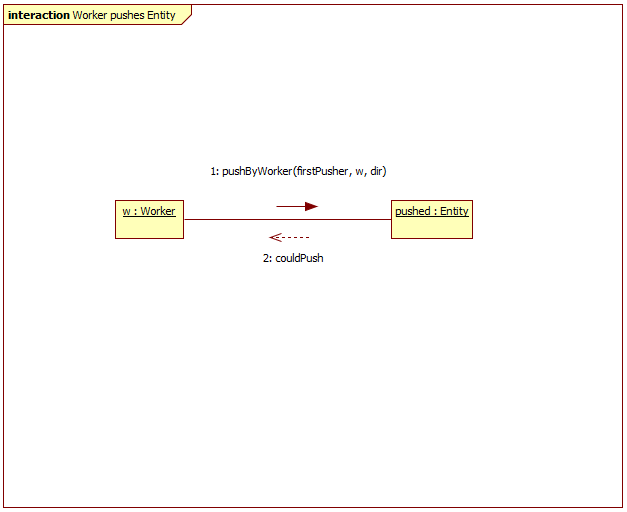
### Entitás új mezőre lép

**

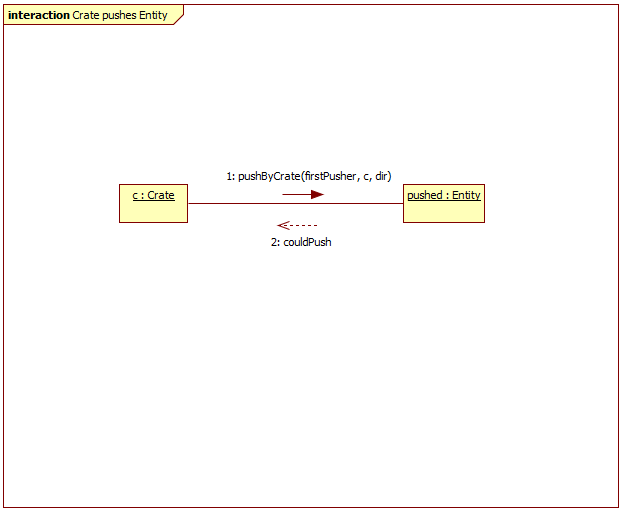
### Entitás lép

**

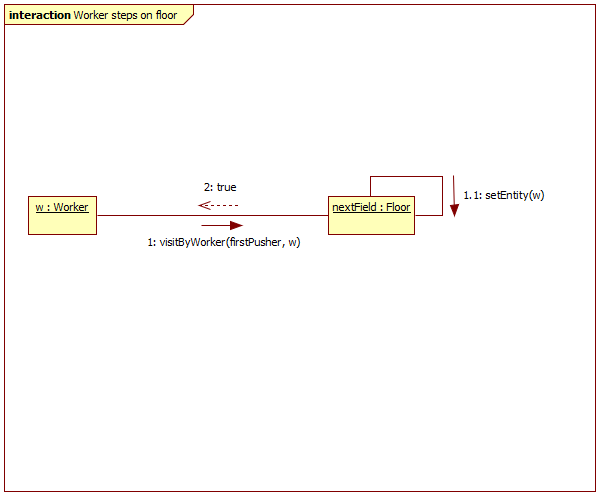
### Dolgozó entitást tol



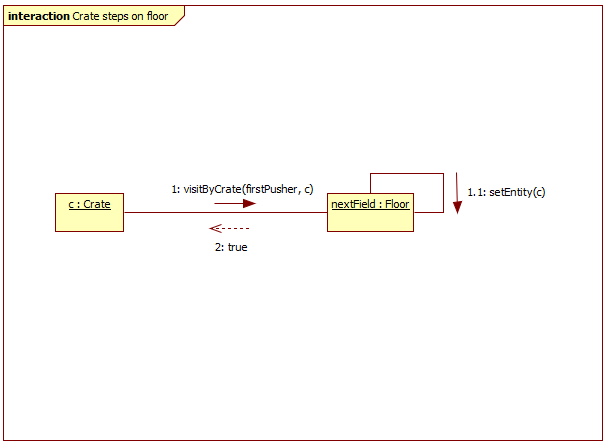
### Láda entitást tol



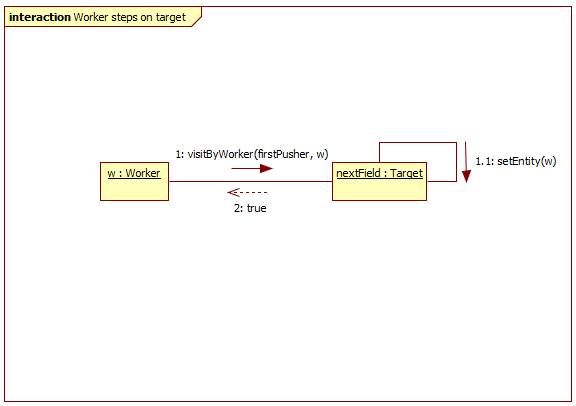
### Dolgozó padlóra lép



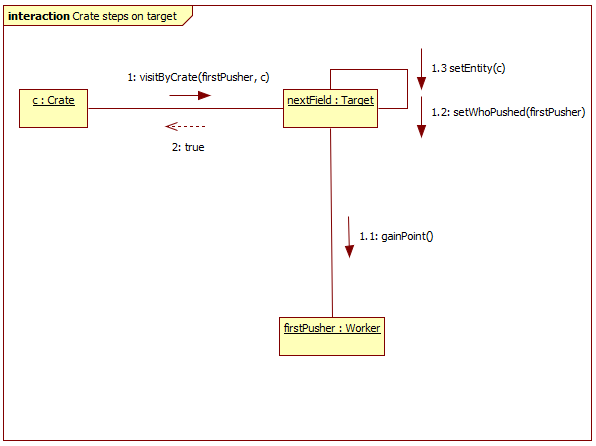
### Láda padlóra lép



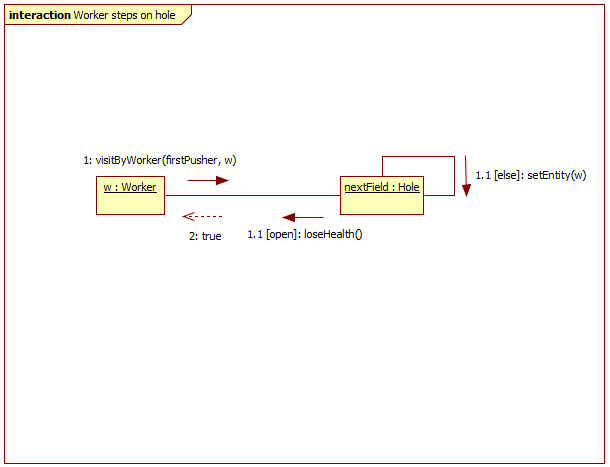
### Dolgozó előírt helyre lép



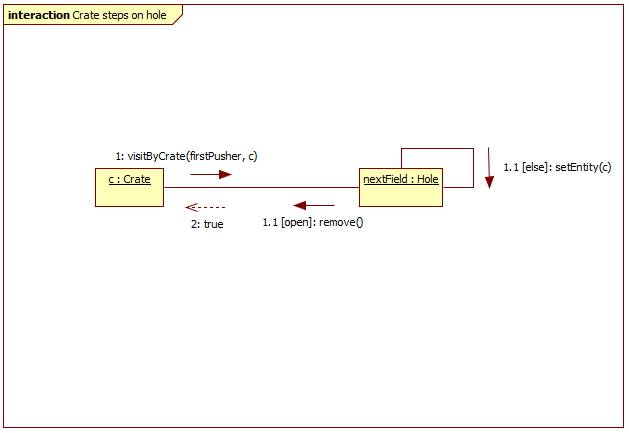
### Láda előírt helyre lép



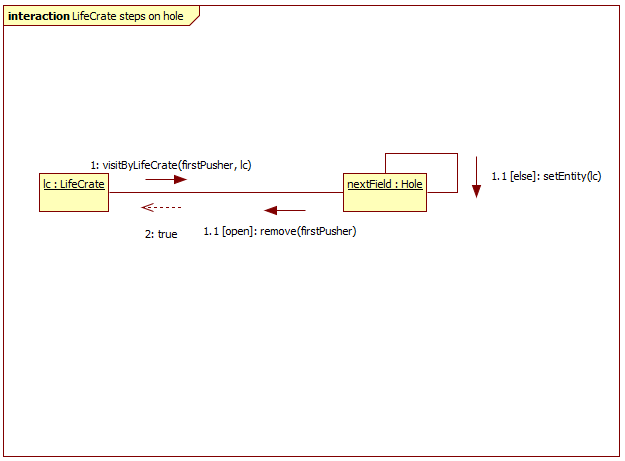
### Dolgozó lyukra lép



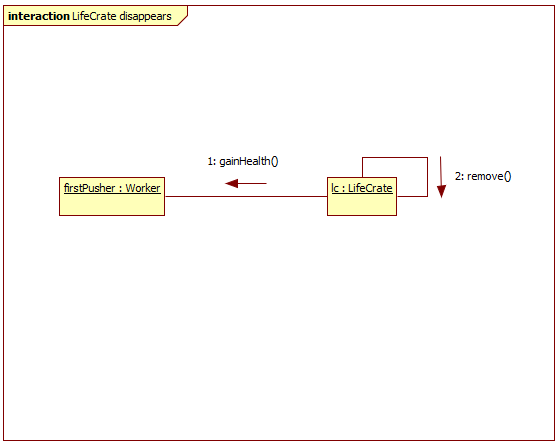
### Láda lyukra lép



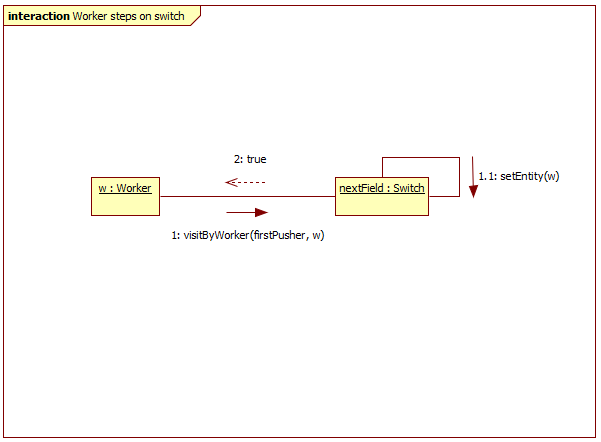
### Szívecskés láda lyukra lép



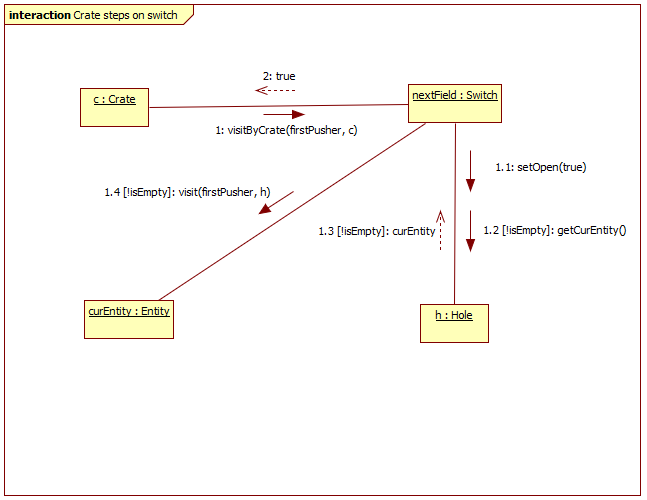
### Szívecskés láda eltűnik



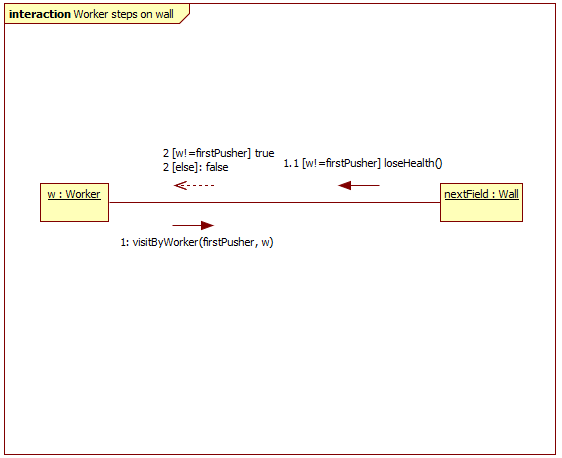
### Dolgozó kapcsolóra lép



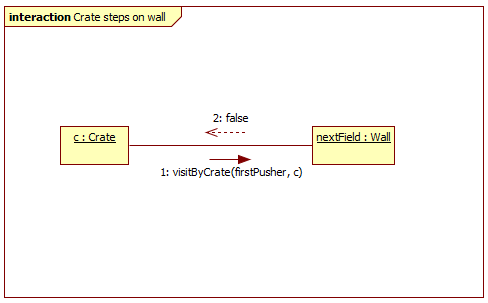
### Láda kapcsolóra lép



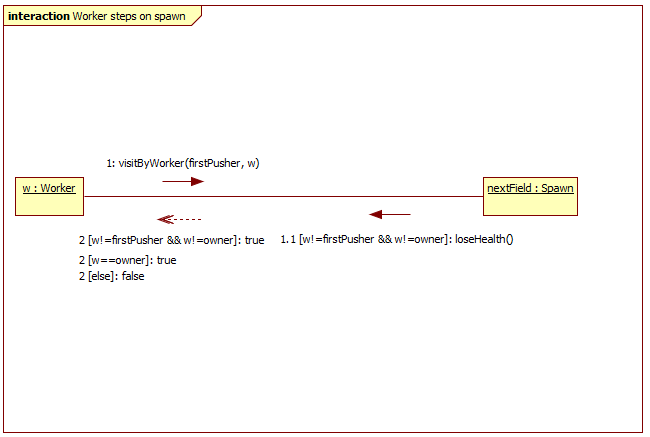
### Dolgozó falra lép



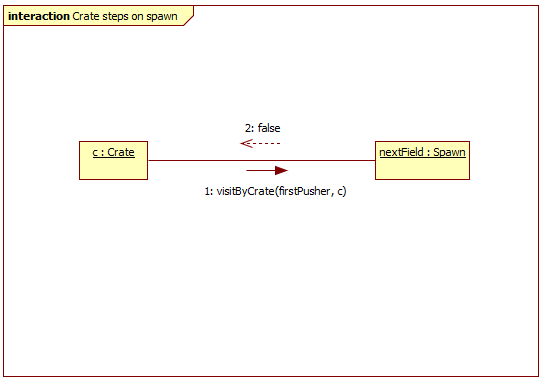
### Láda falra lép



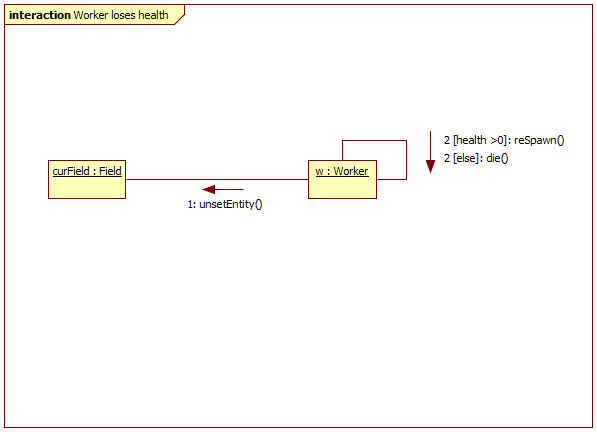
### Dolgozó kiindulási helyre lép



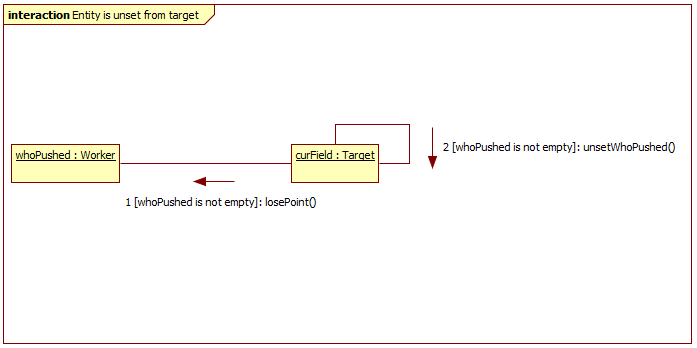
### Láda kiindulási helyre lép



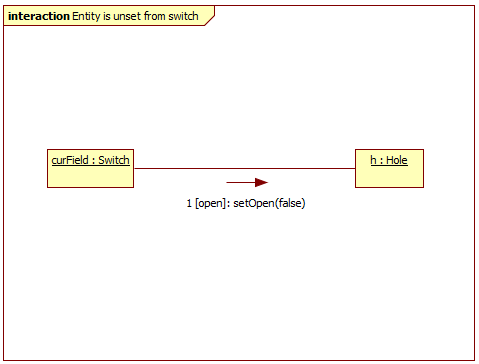
### Dolgozó életet veszít



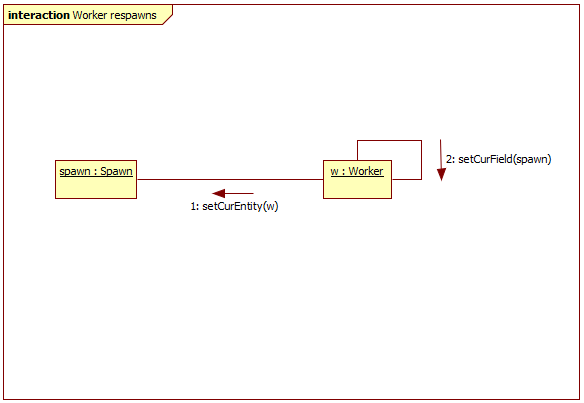
### Entitás lekerül előírt helyről



### Entitás lekerül kapcsolóról



### Dolgozó respawnol



## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.05. 12:00 | 2 óra | LAKATOS  LENKEFI  JANI  CSANÁDY | Értekezlet, melynek döntései:  CSANÁDY és JANI elkészíti a use-case diagramot és a use-casek leírását.  LENKEFI elkészíti a kezelői felületet és dialógusok tervét.  LAKATOS elkészíti a szekvencia diagramokat.  SZAKÁLLAS elkészíti a kommunikációs diagramokat. |
| 2018.03.05. 18:15 | 1,5 óra | CSANÁDY | Use-case diagram (5.1.1) koncepciójának megtervezése. |
| 2018.03.08. 21:00 | 2,5 óra | SZAKÁLLAS | Use-case diagram (5.1.1) elkészítése. |
| 2018.03.09. 17:00 | 1 óra | JANI | Use-case leírások (5.1.2) elkészítése. |
| 2018.03.08. 20:30 | 3 óra | SZAKÁLLAS | Kommunikációs diagramok (5.4) elkészítése. |
| 2018.03.09. 18:00 | 2 óra | LAKATOS | Szekvencia diagramok (5.3) elkészítése. |
| 2018.03.10. 22:45 | 2 óra | CSANÁDY | Use-case leírások pontosítása (5.1.2). |
| 2018.03.10. 23:30 | 1,5 óra | SZAKÁLLAS | Use-case leírások pontosítása (5.1.2). |
| 2018.03.11. 9:45 | 2 óra | JANI | Dokumentum általános felülvizsgálata, konzisztencia ellenőrzése, apróbb módosítások. |
| 2018.03.11. 18:30 | 2 óra | LENKEFI | A kezelői felület és dialógusok tervének elkészítése (5.2). |
| 2018.03.11. 21:00 | 2 óra | LENKEFI | Dokumentum általános felülvizsgálata, konzisztencia ellenőrzése. |
| 2018.03.11. 23:00 | 1 óra | LAKATOS | Dokumentum véglegesítése. |