**4. Analízis modell kidolgozása**

24 – beta

Konzulens:

Ferencz Endre

Csapattagok

|  |  |  |
| --- | --- | --- |
| Ködöböcz Ferenc | Z3GPHW | ferko1276@gmail.com |
| Sándorfi Richárd | BR8P1W | ricsike15@gmail.com |
| Koltai Kadosa Márton | EEGDPX | kadosa.koltai@gmail.com |
| Jenei Pál | HJTX4V | jenei.pal95@gmail.com |
| Győri Kristóf | HV0R9S | gykristof0206@gmail.com |

2018.03.05.

1. **Analízis modell kidolgozása**
   1. ***Objektum katalógus***
      1. **Box**

Ezeket kell a dolgozóknak(**Player**) a megadott célhelyekre(**StorageArea**) eljutattniuk, hogy pontot kapjanak érte.

A **Box**-ok mezőkön állnak és egy-egy **Box** teljesen elfoglalja azt a mezőt amin áll (nem lehet mellette a mezőn másik **Box**, vagy **Worker**).

A **Box**-okat el lehet tolni egyik celláról a másikra, de mindig csak szomszédos cellákra.

A **Box** tudja, hogy melyik **Player** tolta el.

A **Box**-ok egymást is képesek tolni. A **Box**-okat lehet kapcsolók(**Switch**) bekapcsolására is használni, úgy, hogy a ládát arra a mezőre toljuk ahol a kapcsoló(**Switch**) van.

* + 1. **Floor**

Egy padlóelemet jelképez amelyre rá lehet lépni, ennek vannak különböző specializált változatai(**StorageArea**,**Switch**,**Trapdoor**) . Referenciát tárol a rajta tartózkodó elemről.

* + 1. **Player**

A játékos reprezentációja a játékban. A játékos mozgatja a Player-t, ládákat tol vele adott célhelyekre annak érdekében, hogy minél több pontot gyűjtsön így össze. A játékos meg tud halni, ha lyukra lép, vagy pedig, ha rátolnak egy ládát úgy, hogy nem tud rálépni a tolás irányában a következő cellára(mert pl. fal van ott).

* + 1. **Wall**

Ez jelképezi a raktár falait, illetve oszlopait a játékban. A falak egyfajta Fieldek, így abból származnak le. A falra nem lehet rálépni.

* + 1. **Hole**

Egyfajta Field amelyre, ha bármi rákerül azt eltünteti a pályáról (Player meghal, Box eltűnik).

* + 1. **Trapdoor**

Ezek olyan padlóelemek amelyekhez kapcsoló tartozik. Alapértelmezetten padlónak tűnik, de ha a Switch-ére ládát tolnak lyukká változik, ameddig a kapcsolón marad a láda.

* + 1. **Switch**

Kapcsolót jelképez a játékban, amely egy TrapDoor-hoz csatlakozik, ha Box-ottolnak rá lyukká változtatja a hozzá kapcsolódó TrapDoor-t.

* + 1. **StorageArea**

Ha erre ládát tolnak 1 pontot ad annak a játékosnak, aki a ládát rátolta, ha a láda elmozdul a pont elveszik és mindig annak adja a pontot, aki legutoljára tolt rá ládát. Referenciát tárol a játékosról, aki utoljára rátolta a ládát.

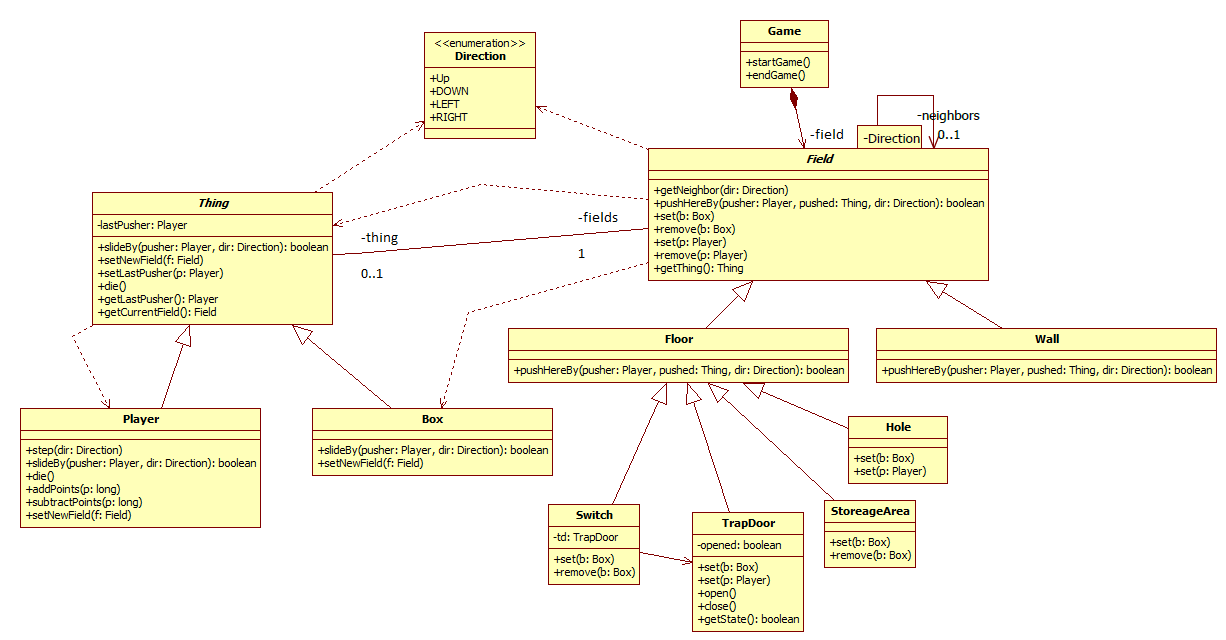
* + 1. **Game**

Ez az objektum felelős a játék logikájáért, elindítja a játékot, ha nincs több mozgatható láda, vagy csak egy játékos maradt megállítja azt, illetve tárolja a különböző pályákat.

* + 1. **Warehouse**

Egy adott pálya reprezentációja, **Field**-ek mátrixát tartalmazza

* 1. ***Statikus struktúra diagramok***

******

* 1. ***Osztályok leírása***
     1. **Box**
        + **Felelősség**

Az előtte lévő dolgot kell mozgásra vagy csúszásra kérnie és tovább adni neki azt a játékost, aki miatt elmozdult a doboz.

* + - * **Ősosztályok**

Thing

* + - * **Interfészek**
      * **Attribútumok**
* **-lastPusher: Player**: ebben az objektumban tárolódik annak a játékosnak a referenciája, aki a dobozt utoljára megtolta.
  + - * **Metódusok**
* **bool slideBy(Player pusher, Dir d)**: rekurzívan hívja az előtte lévő dolgokat ellenőri, hogy tud e a következő mezőre csúszni a doboz.
* **void setNewField(Field new)**: átregisztrálja magát az új mezőre.
* **void die()**: amikor ezt a függvényt meghívják, akkor a doboz eltűnik a játékpályáról.
* Player getLastPusher(): visszaadja annak a játkosnak a referenciáját, aki utoljára megtolta a dobozt.
* **void setLastPusher(p: Player):** beállítja a lastPusher objektumot.
* **void action(s: Switch):** meghívja a paraméterben kapott objektumreferencia interact() függvényét, ami a váltó állapot váltásáért felelős.
  + 1. **BoxFloor**
       - **Felelősség**

Egy absztrakt osztály, ami egy olyan padlómezőt definiál, amire, ha dobozt tolunk speciális műveletek hajtódnak végre.

* + - * **Ősosztályok**

Field Floor

* + - * **Interfészek**
      * **Attribútumok**
      * **Metódusok**
* **void remove(Box b)**: eltávolítja a dobozt erről a padlómezőről.
* **void set(Box b)**: rácsatol egy dobozt a padlómezőre.
* **boolean pushHereBy(Player p, Box t, Dir d)**: megpróbálja tovább léptetni a rajta lévő dolgot, ha sikerül ide csatolja a dobozt.
  + 1. **Direction**
       - **Felelősség**

Ez az enumeráció egy mező (Field) lehetséges szomszédos irányait reprezentálja: fel (UP), le (DOWN), balra (LEFT) vagy jobbra (RIGHT).

* + 1. **Field**
       - **Felelősség**

Absztrakt osztály tárolja azokat a dolgot (thing), amelyek előfordulhatnak a raktárban, illetve ismeri a szomszédos mezőket. Az ebből leszármazó osztályok fogják azokat az elemeket leírni, amelyeket a raktár pályán vannak, mint mező elem. Ha egy dolog kerül rá, tovább csúsztatja a rajta már előtte lévő dolgot.

* + - * **Ősosztályok**
      * **Interfészek**
      * **Attribútumok**
* **-thing: Thing:** éppen a mezőn lévő dolog referenciája.
* -**neighbors[Direction]: Field:** Tárolja a szomszédos mezőket a lehetséges irányokba.
  + - * **Metódusok**
* **Field getNeighbor(Direction d)**: a d irányban lévő szomszédos mezőt adja vissza.
* void enterHere(Player p, Direction d)
* pushHereBy(Player p, Thing t, Direction d)
  + 1. **Floor**
       - **Felelősség**

Fogadja az érkező dolgokat. Mikor rálépnek, akkor tovább tolja az itt található dolgot, ha ez lehetséges.

* + - * **Ősosztályok**

Field

* + - * **Interfészek**
      * **Attribútumok**
      * **Metódusok**
* **void enterHere(Player p, d)**: erre a mezőre lép és eltolja az itt található dolgot.
* **void remove(Thing t)**: eltávolítja a dolgot a mezőről.
* **void set(Thing t)**: rácsatolja a dolgot a mezőre.
* **bool pushHereBy(Player p, Thing t, Direction d)**: megpróbálja tovább léptetni a rajta lévő dolgot, ha sikerül ide csatolja a dobozt.
  + 1. **Game**
       - **Felelősség**

A játék elindításáért és befejezéséért felelős.

* + - * **Ősosztályok**
      * **Interfészek**
      * **Attribútumok**
* **players**: **Player[1..\*]:** a játékosokat tárolja.
* **fields**: **Field[1..\*]:** a pályán lévő mezők halmaza.
  + - * **Metódusok**
* **void startGame():** inicializálja és elindítja a játékot.
* **void endGame():** a játékot befejezi.
  + 1. **Hole**
       - **Felelősség**

Azokat a dolgokat, amik valamilyen módon erre a mezőre kerülnek, azokat megöli.

* + - * **Ősosztályok**

Field → Floor

* + - * **Interfészek**
      * **Attribútumok**
      * **Metódusok**
* **void set(Thing t)**: rácsatolja a mezőre a dolgot, amit ide érkezik, de ezzel együtt meg is meg is öli.
  + 1. **Player**
       - **Felelősség**

Egy pályán lévő játékos. Pontjainak a számlálására és a pályán való mozgásért felelős.

* + - * **Ősosztályok**

Thing

* + - * **Interfészek**
      * **Attribútumok**
* **-points: long:** a gyűjtött pontok száma
  + - * **Metódusok**
* **bool slideBy(Player p, Direction d)**: rekurzívan ellenőrzi, hogy tud e a következő mezőre csúszni. Ha nem tud csúszni megöli magát.
* **void die()**: véget vet a játéknak, a játékost megöli.
* **void step(Direction d):** egy adott irányba lépteti a játékost.
* **void addPoints(long l):** pontot ad magának.
* **void substractPooints(long l):** pontot von le magától.
  + 1. **StoreageArea**
       - **Felelősség**

Egy célhelymező feladata, hogy aki erre a helyre tolt doboz, azt ponttal jutalmazza és ha eltolnak innen egy dobozt, akkor a megfelelő embertől vonja le a pontot.

* + - * **Ősosztályok**

Field Floor BoxFloor

* + - * **Interfészek**
      * **Attribútumok**
      * **Metódusok**
* **void set(Box b):** rácsatolja a dobozt önmagára, valamint annak a játékosnak pontot ad, aki ide tolta a dobozt.
* **void remove(Box b):** eltávolítja a dobozt önmagáról és levonja annak a játékosnak a pontját, aki előtte a dobozt ide tolta.
  + 1. **Switch**
       - **Felelősség**

Amikor dobozt tolnak ide, akkor a hozzátartozó csapóajtaját ki kell nyitnia és hogy ha nincs rajta doboz, akkor be kell zárnia.

* + - * **Ősosztályok**

Field Floor BoxFloor

* + - * **Interfészek**
      * **Attribútumok**
* **-trapDoor: TrapDoor**: a kapcsolómezőhöz tartozó csapóajtóra referencia.
  + - * **Metódusok**
* **void interact(Box b):** a trapDoor invert() függvényét hívja meg, amivel átváltja a trapDoor állapotát.
* **void set(Box b):** elhelyez egy doboz a mezőre és kinyitja a trapDoort.
* **void remove(Box b):** eltávolítja a dobozt mezőről és bezárja a trapDoort.
* **TrapDoor getTrapDoor():** trapDoor lekérdezésére való.
  + 1. **Thing**
       - **Felelősség**

A dolgok ősosztálya, ami eltárolja, hogy éppen melyik mezőn van az adott dolog.

* + - * **Ősosztályok**
      * **Interfészek**
      * **Attribútumok**
* **field: Field**: az a mező, amin éppen a dolog van.
  + - * **Metódusok**
* **void die():** megöli a dolgot.
* **Field getField():** visszaadja azt a mezőt, amin jelenleg áll.
* **void setNewField(f: Field):** beállítja az új mezőt.
* **boolean slideBy(Player p, Direction d):** rekurzívan ellenőrzi, hogy tud e a következő mezőre csúszni.
  + 1. **TrapDoor** 
       - **Felelősség**

Az csapóajtóra kerülő dolgok kezelésére szolgál.

* + - * **Ősosztályok**

Thing

* + - * **Interfészek**
      * **Attribútumok**
* **-opened: boolean:** azt tárolja, hogy a csapóajtó éppen nyitva van-e vagy sem.
  + - * **Metódusok**
* **void set(Thing t):** egy dolgot a csapóajtóra helyez, ami ha nyitva van, akkor a dolog meghal.
* **void invert():** a csapóajtó állapotának a változtatására szolgál.
* **boolean getState():** visszaadja a csapóajtó jelenlegi állapotát.
  + 1. **Wall**
       - **Felelősség**

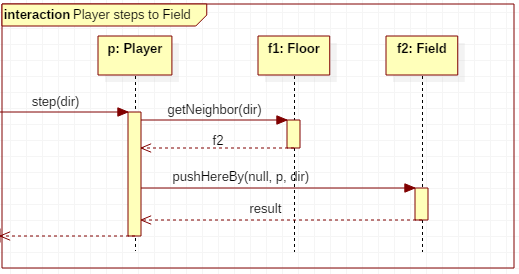
Nem engedi magára a dobozt és a játékost, ha ide próbál csúszni. Ha a játékos ide próbál lépni akkor nem hal meg.

* + - * **Ősosztályok**

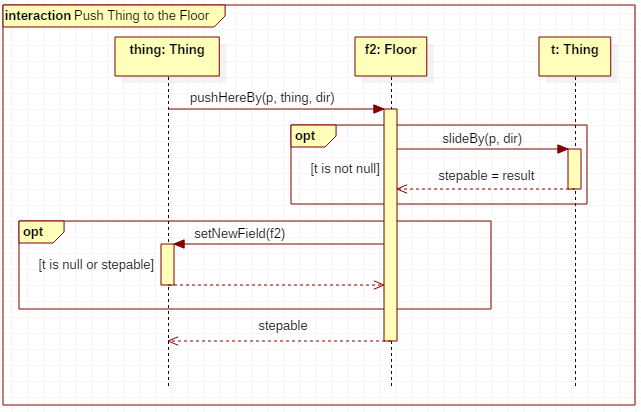
Field

* + - * **Interfészek**
      * **Attribútumok**
      * **Metódusok**
* **void enterHere(Player p, Direction d)**: nem engedi idelépni a játékost.
* **bool pushHereBy(Player p, Thing b, Dir d)**: nem engedi ide tolni a dolgot, mindig false értékkel tér vissza, tehát jelzi, hogy ide nem lehet senkit sem tolni.
  1. ***Szekvencia diagramok***

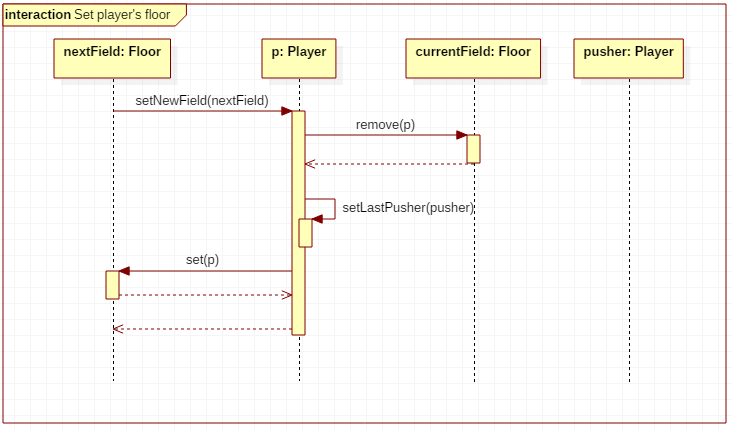
***4.4.1. A játékos egy mezőre lép***



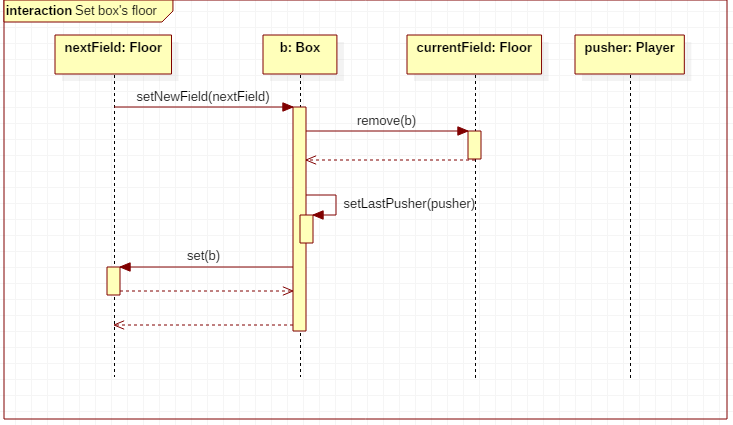
***4.4.2. A dolgot egy padló mezőre kerül***



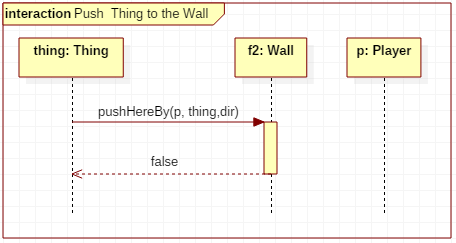
***4.4.3. Játékos egy padlóra helyezése***



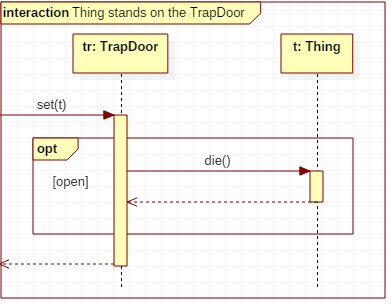
***4.4.4. A doboz egy padlóra helyezése***



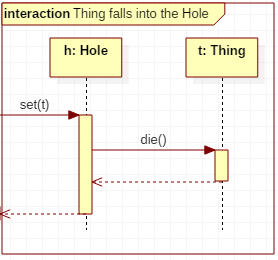
***4.4.5. Egy dolgot a falhoz kerül***



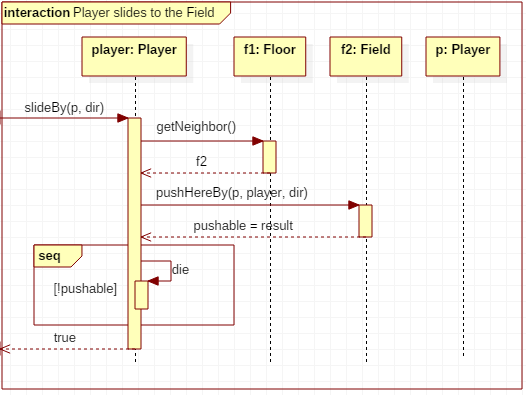
***4.4.6. A dolog rálép a csapóajtóra***



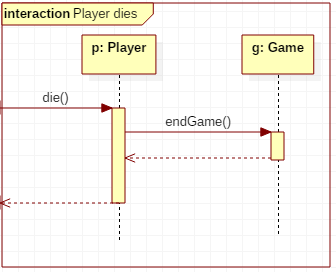
***4.4.7. A dolog a lyukba esik***



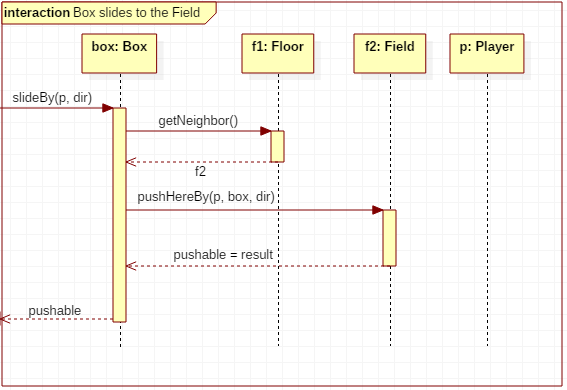
***4.4.7. A játékost egy mezőre elcsúsztatják***



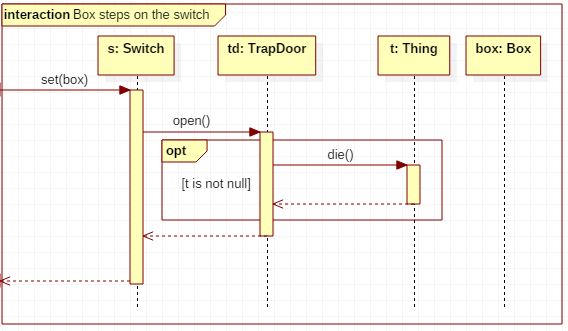
***4.4.9. A játékos meghal***



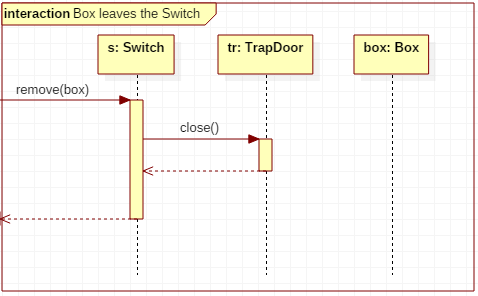
***4.4.10. A doboz egy mezőre elcsúszik***



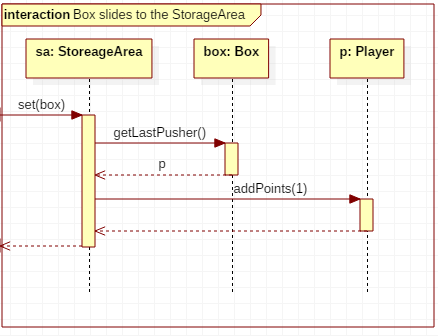
***4.4.11. Egy doboz kapcsolómezőre csúszott***



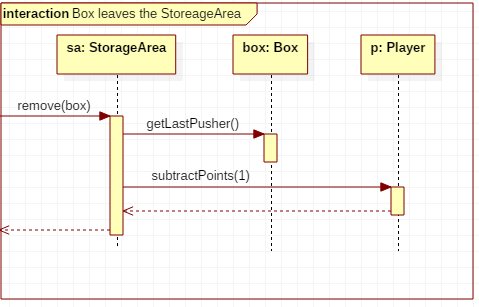
***4.4.12. Egy doboz egy kapcsolómezőről elcsúszik***



***4.4.15. A doboz egy célhelyre ér***

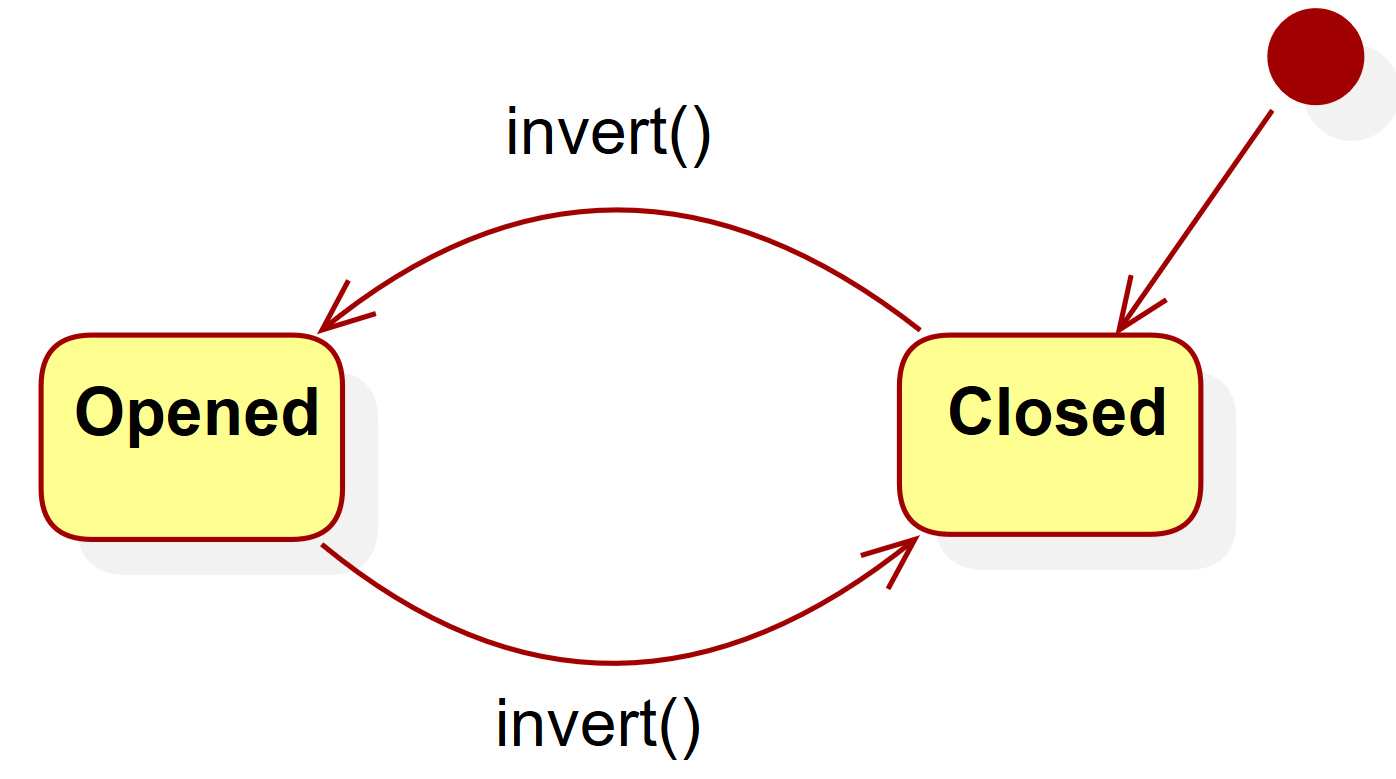


***4.4.16. A doboz egy célhelyről eltolják***



* 1. ***State-chartok***

**4.5.1. A Trapdoor állapotgépe**



* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.02. 18:00 | 2 óra | Győri  Jenei  Ködöböcz  Koltai  Sándorfi | Értekezlet.  Döntés: Ködöböcz elkészíti az osztálydiagramot, Győri a szekvencia diagramokat.  Jenei elkészíti az objektum listát |
| 2018.03.04. 13:00 | 0,5 óra | Jenei | Tevékenység: Jenei elkészíti az objektum listát |
| 2018.03.04 11:00 | 6 óra | Győri  Ködöböcz | Tevékenység: Szekvenciadiagrammok megrajzolása. Győri a Player által végrehajtott interakciókat készített el. Ködöböcz a dobozok által végrehajtantó interakciókat készítette el. |
| 2018.03.04 20:00 | 6 óra | Győri | Tevékenység: Az osztálydiagram megrajzolása és javítása. Az osztályok leírása. |
| 2018.03.05 9:00 | 2 óra | Győri | Dokumentum átnézése, formázása, hibák javítása. |