1. **Szkeleton tervezése**
   1. ***A szkeleton modell valóságos use-case-ei***
      1. **Use-case diagram**
      2. **Use-case leírások**

*[Minden use-case-hez külön]*

* *új játék indítása (Board: SetUp(), Game: StartGame())*
* *játékos üres szomszédos mezőre lép (Player: Move(d), Field: Accept(p), Remove(t), GetNeighbor(d))*
* *játékos közvetlenül ládát tol egy üres mezőre (Box: PushedBy(p1,p1,d), Field: Accept(b))*
* *játékos fal melletti ládát tol (Wall: Accept(b))*
* *játékos az utolsó mozgatható dobozt közvetlenül nyitott lyukba tolja (Box: PushedBy(p,t,d), Die(), Game: DecActiveBoxes(), CheckEndGame(), EndGame(), Hole: Accept(b)/2, Player: GetPoint())*
* *játékos az utolsó 3 mozgatható dobozt sorban beletolja egy nyitott lyukba - külön, hogy ha több hal meg egyszerre, akkor is bejön-e a játék vége (Box: PushedBy(p1,b,d))*
* *játékos közvetlenül csukott lyukra lép (Hole: Accept(p)/1)*
* *játékos közvetlenül nyitott lyukra lép (Hole: Accept(p)/2, Die())*
* *játékos ládát csukott lyukra tol (Hole: Accept(b)/1)*
* *játékos játékost tol (Player: PushedBy(p1,p1,d))*
* *játékos közvetve játékost tol (Player: PushedBy(p1,b,d))*
* *játékos közvetve játékost tol falnak (Wall: Accept(p))*
* *játékos megtol egy doboz/játékos/játékos/doboz sort*
* *játékos közvetlenül ládát tol pontszerző mezőre (Player: AddPoint(), PointField: Accept(b))*
* *játékos közvetlenül pontszerző mezőre lép (PointField: Accept(p))*
* *játékos lép pontszerző mezőre, amin van láda*
* *játékos már teljesített pontszerző mezőre tol ládát (azonos és különböző játékos esetére is)*
* *játékos az utolsó teljesítetlen pontszerző mezőre tol ládát*
* *játékos közvetetten ládát tol pontszerző mezőre láda-játékos-láda sorban*
* *játékos ládát tol üres switchre, a hozzá tartozó hole-on van egy láda/player*
* *játékos ládát tol foglalt switchre*
* *játékos ládát tol el foglalt switchről*

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

* 1. ***A szkeleton kezelői felületének terve, dialógusok***

*[A szkeleton által elfogadott bemenetek , valamint a szöveges konzolon megjelenő kimenetek. A kiemenet formátuma olyan kell legyen, ami alapján a működés összevethető a korábbi szekvencia-diagramokkal.]*

* 1. ***Szekvencia diagramok a belső működésre***

*[A szkeletonban implementált szekvenciadiagramok. Tipikusan egy use-case egy diagram. Ezek megegyezhetnek a korábban specifikált diagramokkal, de az egyes életvonalakat (lifeline) egyértelműen a szkeletonban példányosított objektumokhoz kell tudni kötni. Azt kell megjeleníteni, hogy a szkeletonban létrehozott objektumok egymással hogyan fognak kommunikálni.]*

* 1. ***Kommunikációs diagramok***

*[A szkeletonban, az egyes szkeleton-use-case-ek futása során létrehozott objektumok és kapcsolataik bemutatására szolgáló diagramok. Ezek alapján valósítják meg a szkeleton fejlesztői az inicializáló kódrészleteket.]*

* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018. 03. 10. | 1 óra | Nemes | Use-case-ek listájának készítése |
| 2018. 03. 10. | 1 óra | Nemes  Kovács | Értekezlet  Hole osztály SetOpen(b:boolean) és Switch osztály Remove(b:Box) metódusainak felülvizsgálata, osztálydiagram javítása (accept függvények) |
| … | … | … | … |