# Szkeleton tervezése

## A szkeleton modell valóságos use-case-ei

[A szkeletonnak, mint önálló programnak a működésével kapcsolatos use-case-ek. ]

### Use-case diagram

### Use-case leírások

[Minden use-case-hez külön]

|  |  |
| --- | --- |
| **Use-case neve** | Tektontörés |
| **Rövid leírás** | A tekton eltörik |
| **Aktorok** | Tesztelő |
| **Alapállapot** | Létezik A és B tekton, ezek szomszédok. Az A tektonon van M Mycelium és I rovar, a B tektonon meg MB MushroomBody |
| **Bemenet** | A köv. metódus hivódik meg: A.onTurnBegin() |
| **Kommunikációs diagram** | x. kommunikációs diagram. |
| **Forgatókönyv** | 1. A tekton jelzést kap hogy új kör kezdődött 2. Elvágodik a rajta lévő összes fonál. 3. Jelet küld a szomszédos tektonjainak, hogy nézzék meg hogy a rajtuk levő fonalaknak van e ellátásuk 4. Jelet küld az összes rajta lévő vonalnak, hogy meneküljenek el 5. Létrehozza az új (letört) tecton |
|  | Tesztelő  =onTurnBegin() => A: FertileTecton  A: FertileTecton  =delete() => M:Mycelium  =myceliaCheckSustain() => B: FertileTecton  =runAway() => I: Insect  =Create() => newt: FertileTecton  =addNeighbour(A: FertileTecton) => newt: FertileTecton |

|  |  |
| --- | --- |
| **Use-case neve** | Rovar sikeres mozgása |
| **Rövid leírás** | I Rovar áll az A tektonon. Utasitást kap hogy menjen B-re |
| **Aktorok** | Tesztelő |
| **Alapállapot** | Létezik A és B tekton, ezek szomszédok. A tektonon van egy I rovar. B tektonon van Mycelium |
| **Bemenet** | A köv. metódus hivódik meg: I.move(B) |
| **Kommunikációs diagram** | x. kommunikációs diagram. |
| **Forgatókönyv** | 1. I rovar mozgási szándékot jelez B tecton felé. 2. B megvizsgálja, hogy tud e őrá mozogni I : szomszédos e I location tektonjával, és van e rajta Mycelium 3. B jelzi A tecton felé hogy I elment róla. 4. B beállitja magának I-t, mint rajta lévő rovar. 5. B visszajelez I-nek hogy mozgása sikeres, ezáltal állitsa be saját helyét B-re és csökkentse a remainingMoves attribútumát egyel |
|  | Tesztelő  =move(B: FertileTecton) => I: Insect  I: Insect  =moveInsect(I: Insect, A: FertileTecton) => B: FertileTecton  B: FertileTecton  =distance(B: FertileTecton) => A: FertileTecton  <= distance: int =A: FertileTecton  =removeOccupant(I: Insect) => A: FertileTecton  =setLocation(B: FertileTecton) => I: Insect |

|  |  |
| --- | --- |
| **Use-case neve** | Rovar sikertelen mozgása – Nem Szomszédosak a tektonok |
| **Rövid leírás** | I Rovar áll az A tektonon. Utasitást kap hogy menjen B-re |
| **Aktorok** | Tesztelő |
| **Alapállapot** | Létezik A és B tekton. A tektonon van egy I rovar. B tektonon van Mycelium |
| **Bemenet** | A köv. metódus hivódik meg: I.move(B) |
| **Kommunikációs diagram** | x. kommunikációs diagram. |
| **Forgatókönyv** | 1. I rovar mozgási szándékot jelez B tecton felé. 2. B megvizsgálja, hogy tud e őrá mozogni I : szomszédos e I location tektonjával, és van e rajta Mycelium 3. Elbukik a szomszédsági feltételen |
|  | Tesztelő  =move(B: FertileTecton) => I: Insect  I: Insect  =moveInsect(I: Insect, A: FertileTecton) => B: FertileTecton  B: FertileTecton  =distance(B: FertileTecton) => A: FertileTecton  <= distance: int =A: FertileTecton |

|  |  |
| --- | --- |
| **Use-case neve** | Rovar sikertelen mozgása – Nincs Mycelium a céltektonon |
| **Rövid leírás** | I Rovar áll az A tektonon. Utasitást kap hogy menjen B-re |
| **Aktorok** | Tesztelő |
| **Alapállapot** | Létezik A és B tekton, ezek szomszédok. A tektonon van egy I rovar. |
| **Bemenet** | A köv. metódus hivódik meg: I.move(B) |
| **Kommunikációs diagram** | x. kommunikációs diagram. |
| **Forgatókönyv** | 1. I rovar mozgási szándékot jelez B tecton felé. 2. B megvizsgálja, hogy tud e őrá mozogni I : szomszédos e I location tektonjával, és van e rajta Mycelium 3. Elbukik a Mycelium feltételen |
|  | Tesztelő  =move(B: FertileTecton) => I: Insect  I: Insect  =moveInsect(I: Insect, A: FertileTecton) => B: FertileTecton  B: FertileTecton  =distance(B: FertileTecton) => A: FertileTecton  <= distance: int =A: FertileTecton |

|  |  |
| --- | --- |
| **Use-case neve** | Rovar általi gombafonál elvágás |
| **Rövid leírás** | I rovar elvágja a location (A) tektonján lévő első fonalat. |
| **Aktorok** | Tesztelő |
| **Alapállapot** | Létezik A és B tekton, ezek szomszédok. A tektonon van M Mycelium és I rovar, B tekton A-val szomszédos, van rajta MB MushroomBody |
| **Bemenet** | A köv. metódus hivódik meg: A.cutMycelium() |
| **Kommunikációs diagram** | x. kommunikációs diagram. |
| **Forgatókönyv** | 1. A tekton jelet kap, hogy rajta fonalvágás történik 2. Elvágja a rajta lévő legelső fonalat 3. Mivel utolsó fonál vágódott el, jelet küld a rajta levő rovaroknak hogy meneküljenek el |
|  | Tesztelő  =cutMycelium() => A: FertileTecton  A: FertileTecton  =cut() => M2: Mycelium  =runAway() => I: Insect |

|  |  |
| --- | --- |
| **Use-case neve** | Rovar elmenekülése |
| **Rövid leírás** | A rovar jelet kap hogy el kell menekülnie tectonjáról. Random kiválaszt egy megfelelő tectont, és odamenekül |
| **Aktorok** | Tesztelő |
| **Alapállapot** | Van A, B és C tekton. A és C illetve C és B szomszédosak. A-n van egy I Insect, B-n van Mycelium |
| **Bemenet** | A köv. metódus hivódik meg: I.runAway() |
| **Kommunikációs diagram** | x. kommunikációs diagram. |
| **Forgatókönyv** | 1. I jelet kap hogy menekülnie kell 2. Random kiválaszt egy tectont és megnézi hogy tud e rá menekülni (ez riggelve van):    1. Először kiválasztja C-t, de elbukik a feltétel, miszerint kell rajta Mycelium legyen    2. Kiválasztja a B-t, amire tud menni, mert van rajta Mycelium 3. Szól B-nek hogy rajta less mostmár 4. Beállitja helyének B-t |
|  | Tesztelő  =runAway() => I: Insect  I: Insect  =hasMycelium() => C: FertileTecton  <= foundTecton: bool =C: FertileTecton  =hasMycelium() => B: FertileTecton  <= foundTecton: bool =B: FertileTecton  =addOccupant(I: Insect) => B: FertileTecton |

|  |  |
| --- | --- |
| **Use-case neve** | Gombatest sikeres Spóra kilövés Szomszédosra |
| **Rövid leírás** | A tectonon MB gombatest, B A-val szomszédos tectonra kilövi a spóráit |
| **Aktorok** | Tesztelő |
| **Alapállapot** | Létezik A,B,C és D tektonok, ezek szomszédosak a következő képpen: A-B-C-D. A-n van egy MB MushroomBody |
| **Bemenet** | A köv. Metódus hivódik meg: MB.ejectSpores(B) |
| **Kommunikációs diagram** | x. kommunikációs diagram. |
| **Forgatókönyv** | 1. MB jelet kap hogy lője ki spóráit B-re 2. A kiszámolja hogy B milyen messze van tőle 3. Eldönti hogy tud e lőni oda (mennyi lövése maradt, milyen messze van) 4. Tud oda lőni, szóval B jelet kap MBtől, hogy az átadott spórákat adja hozzá a rajta lévő spórák listájához. |
|  | Tesztelő  =ejectSpores(B: FertileTecton) => MB: MushroomBody  MB: MushroomBody  =distance(B: FertileTecton) => A: FertileTecton  <= distance:int =A: FertileTecton  =transferSpores(mushroomSpores) => B: FertileTecton |

|  |  |
| --- | --- |
| **Use-case neve** | Gombatest sikeres Spóra kilövés szomszéd szomszédjára |
| **Rövid leírás** | A tectonon MB gombatest, C A szomszédjával szomszédos tectonra kilövi a spóráit |
| **Aktorok** | Tesztelő |
| **Alapállapot** | Létezik A,B,C és D tektonok, ezek szomszédosak a következő képpen: A-B-C-D. A-n van egy MB MushroomBody |
| **Bemenet** | A köv. Metódus hivódik meg: MB.ejectSpores(C) |
| **Kommunikációs diagram** | x. kommunikációs diagram. |
| **Forgatókönyv** | 1. MB jelet kap hogy lője ki spóráit C-re 2. A kiszámolja hogy C milyen messze van tőle 3. Eldönti hogy tud e lőni oda (mennyi lövése maradt, milyen messze van) 4. Tud oda lőni, szóval C jelet kap MBtől, hogy az átadott spórákat adja hozzá a rajta lévő spórák listájához. |
|  | Tesztelő  =ejectSpores(C: FertileTecton) => MB: MushroomBody  MB: MushroomBody  =distance(C: FertileTecton) => A: FertileTecton  <= distance:int =A: FertileTecton  =transferSpores(mushroomSpores) => C: FertileTecton |

|  |  |
| --- | --- |
| **Use-case neve** | Gombatest sikerestelen Spóra kilövés szomszéd szomszédjára |
| **Rövid leírás** | A tectonon MB gombatest, C A szomszédjával szomszédos tectonra kilőné spóráit, de nem tudja, mert nem utolsó kilövése |
| **Aktorok** | Tesztelő |
| **Alapállapot** | Létezik A,B,C és D tektonok, ezek szomszédosak a következő képpen: A-B-C-D. A-n van egy MB MushroomBody |
| **Bemenet** | A köv. Metódus hivódik meg: MB.ejectSpores(C) |
| **Kommunikációs diagram** | x. kommunikációs diagram. |
| **Forgatókönyv** | 1. MB jelet kap hogy lője ki spóráit C-re 2. A kiszámolja hogy C milyen messze van tőle 3. Eldönti hogy tud e lőni oda (mennyi lövése maradt, milyen messze van) 4. Nem tud oda lőni, mert nem utolsó lövése, C viszont szomszédjának szomszédja |
|  | Tesztelő  =ejectSpores(C: FertileTecton) => MB: MushroomBody  MB: MushroomBody  =distance(C: FertileTecton) => A: FertileTecton  <= distance:int =A: FertileTecton |

|  |  |
| --- | --- |
| **Use-case neve** | Gombatest sikerestelen Spóra kilövés nagyobb távolságra |
| **Rövid leírás** | A tectonon MB gombatest, D-re , azaz A-tól 3ra lévő tectonra kilőné spóráit, de nem tudja, mert túl messze van |
| **Aktorok** | Tesztelő |
| **Alapállapot** | Létezik A,B,C és D tektonok, ezek szomszédosak a következő képpen: A-B-C-D. A-n van egy MB MushroomBody |
| **Bemenet** | A köv. Metódus hivódik meg: MB.ejectSpores(D) |
| **Kommunikációs diagram** | x. kommunikációs diagram. |
| **Forgatókönyv** | 1. MB jelet kap hogy lője ki spóráit D-re 2. A kiszámolja hogy D milyen messze van tőle 3. Eldönti hogy tud e lőni oda (mennyi lövése maradt, milyen messze van) 4. Nem tud oda lőni, mert túl messze van |
|  | Tesztelő  =ejectSpores(D: FertileTecton) => MB: MushroomBody  MB: MushroomBody  =distance(D: FertileTecton) => A: FertileTecton  <= distance:int =A: FertileTecton |

## 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.]

## 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.]

## 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.]

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| … | … | … | … |