## 1. Parancsok listája – Felülvizsgálat

***A 11. számú use-case kapcsán***

**CREATE\_MUSHROOMBODY** Mushroomody\_Name Tecton\_Name

**Leírás:** A gombatest létrejön egy tektonon (céltekton).

**Opciók:** Az első paraméter meghatározza a létrehozandó gombatestet, a második a céltektont.

***A 12. számú use-case kapcsán***

**GROW\_MUSHROOMBODY** Mushroomody\_Name Tecton\_Name

**Leírás:** A gombatest létrejön és rákerül egy tektonra (céltekton).

**Opciók:** Az első paraméter meghatározza a létrejövő gombatestet, a második a céltektont.

***A 13. számú use-case kapcsán***

**PUT\_SPORE** Spore\_TypeSpore\_Name Tecton\_Name

**Leírás:** Egy adott típusú spóra rákerül egy tektonra (céltekton).

**Opciók:** Az első paraméter meghatározza a spóra típusát, a második a nevét, a harmadik a céltektont.

***A 14. számú use-case kapcsán***

**EJECT\_SPORES** Mushroomody\_Name Tecton\_Name

**Leírás:** A kiválasztott gombatest valamennyi spórája rákerül egy tektonra (céltekton).

**Opciók:** Az első paraméter meghatározza a gombatestet, a második a céltektont.

***A 15. számú use-case kapcsán***

**DEACTIVATE** Mushroomody\_Name

**Leírás:** A kiválasztott gombatest elpusztul (inaktívvá válik).

**Opciók:** A paraméter meghatározza a gombatestet.

***Új parancs***

**ADD\_SPORE** Mushroomody\_Name Spore\_TypeSpore\_Name

**Leírás:** A rendszer a kiválasztott gombatesthez meghatározott típusú spórát rendel.

**Opciók:** Az első paraméter meghatározza a gombatestet, a második a spóra típusát, a harmadik a spóra nevét.

***Új parancs***

**SET\_REMAININGEJECTS** Mushroomody\_Name RemainingEjects\_Count

**Leírás:** A rendszer a kiválasztott gombatesthez meghatározott számú hátralévő spórakilövést rendel.

**Opciók:** Az első paraméter meghatározza a gombatestet, a második a hátralévő spórakilövések számát.

## 2. A tesztek részletes tervei, leírásuk a teszt nyelvén

[A tesztek részletes tervei alatt meg kell adni azokat a bemeneti adatsorozatokat, amelyekkel a program működése ellenőrizhető. Minden bemenő adatsorozathoz definiálni kell, hogy az adatsorozat végrehajtásától a program mely részeinek, funkcióinak ellenőrzését várjuk és konkrétan milyen eredményekre számítunk, ezek az eredmények hogyan vethetők össze a bemenetekkel. A tesztek leírásakor az előző dokumentumban (proto koncepciója) megadott szintakszist kell használni.]

### TSZ/1. számú teszteset – Gombatest sikeres növesztése FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton) gombafonál által

* **Leírás**

Gombafonál sikeresen növeszt gombatestet olyan FertileTectonon, amely nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton.

(Megjegyzés: SustainingTecton, MultiLayeredTecton és AridTecton esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + gombatest sikeres növesztése az ehhez szükséges feltételekkel: a FertileTectonon mint céltektonon van legalább 3 db spóra és nincs gombatest
  + a létrejött gombatest spóráinak meghatározása
  + a létrejött gombatest megmaradt spórakilövései számának beállítása
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton f1

SET\_BREAKTIMER f1 5

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 f1

PUT\_SPORE SpeedSpore s1 f1

PUT\_SPORE SpeedSpore s2 f1

PUT\_SPORE SpeedSpore s3 f1

//Act

GROW\_MUSHROOMBODY mb1 f1

SET\_REMAININGEJECTS mb1 3

ONTURNBEGIN f1

ONTURNBEGIN m1

//Assert

STATE f1

STATE m1

STATE mb1

* **Elvárt kimenet**

f1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

s1

s2

s3

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = false

location Tecton = f1

growTimer int = 0

mb1: MushroomBody

remainingEjects int = 3

location Tecton = f1

mushroomSpores List<Spore> = {

}

### TSZ/2. számú teszteset – Gombatest spórahiány miatti sikertelen növesztése FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton) gombafonál által

* **Leírás**

Gombafonál sikertelenül kísérel meg gombatestet létrehozni olyan FertileTectonon, amely nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton, és amelyen nem található elegendő spóra.

(Megjegyzés: SustainingTecton, MultiLayeredTecton és AridTecton esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + gombatest sikertelen növesztése a vonatkozó feltételekkel: a FertileTectonon mint céltektonon nincs legalább 3 db spóra és van gombatest
  + az objektumok állapotában nem következik be változás
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton f1

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 f1

PUT\_SPORE SpeedSpore s1 f1

PUT\_SPORE SpeedSpore s2 f1

//Act

GROW\_MUSHROOMBODY mb1 f1

ONTURNBEGIN f1

ONTURNBEGIN m1

//Assert

STATE f1

STATE m1

* **Elvárt kimenet**

f1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

s1

s2

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = false

location Tecton = f1

growTimer int = 0

### TSZ/3. számú teszteset – Gombatest sikertelen növesztése gombafonál által olyan FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton), amelyen már van gombatest

* **Leírás**

Gombatest sikertelen növesztése gombafonál által olyan FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton), amelyen már van gombatest.

(Megjegyzés: SustainingTecton, MultiLayeredTecton és AridTecton esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + gombatest sikertelen növesztése a vonatkozó feltételekkel: a FertileTectonon mint céltektonon van gombatest és legalább 3 db spóra
  + az objektumok állapotában nem következik be változás
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton f1

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 f1

CREATE\_MUSHROOMBODY mb1 f1

PUT\_SPORE SpeedSpore s1 f1

PUT\_SPORE SpeedSpore s2 f1

PUT\_SPORE SpeedSpore s3 f1

//Act

GROW\_MUSHROOMBODY mb2 f1

ONTURNBEGIN f1

ONTURNBEGIN m1

//Assert

STATE f1

STATE m1

* **Elvárt kimenet**

f1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

s1

s2

s3

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = false

location Tecton = f1

growTimer int = 0

mb1: MushroomBody

remainingEjects int = 3

location Tecton = f1

mushroomSpores List<Spore> = {

}

### TSZ/4. számú teszteset – Gombatest sikertelen növesztése gombafonál által SemiFertileTectonra

* **Leírás**

Gombafonál sikertelenül kísérel meg gombatestet létrehozni SemiFertileTectonon, amelyen van legalább 3db spóra (és nincs rajta gombatest).

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + gombatest sikertelen növesztése a vonatkozó feltételekkel: a SemiFertileTectonon mint céltektonon van legalább 3 db spóra
  + az objektumok állapotában nem következik be változás
* **Bemenet**

// Arrange

CREATE\_TECTON SemiFertileTecton f1

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 f1

PUT\_SPORE SpeedSpore s1 f1

PUT\_SPORE SpeedSpore s2 f1

PUT\_SPORE SpeedSpore s3 f1

//Act

GROW\_MUSHROOMBODY mb1 f1

ONTURNBEGIN f1

ONTURNBEGIN m1

//Assert

STATE f1

STATE m1

* **Elvárt kimenet**

f1: SemiFertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

s1

s2

s3

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = false

location Tecton = f1

growTimer int = 0

### TSZ/5. számú teszteset – Gombatest sikeres spórakilövése a gombatest elhelyezkedése szerinti tektonnal szomszédos FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton)

* **Leírás**

Gombatest sikeresen kilövi a spóráit a gombatest elhelyezkedése szerinti tektonnal szomszédos FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton).

(Megjegyzés: SustainingTecton, MultiLayeredTecton, AridTecton és SemiFertileTecton mint céltektonok esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + gombatest sikeres spórakilövése a vonatkozó feltételekkel: gombatest az utolsó, azaz a harmadik kilövése előtt csak valamely szomszédjára lőhet spórát
  + a gombatest spóráinak száma a kilövés következtében 0-ra csökken
  + a kilőtt spórákat a továbbiakban a céltekton tartja nyilván
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton f1

CREATE\_TECTON FertileTecton {f1} f2

CREATE\_MUSHROOMBODY mb1 f1

ADD\_SPORE mb1 SpeedSpore s1

ADD\_SPORE mb1 SpeedSpore s2

ADD\_SPORE mb1 SpeedSpore s3

SET\_REMAININGEJECTS mb1 3

//Act

EJECT\_SPORES mb1 f2

ONTURNBEGIN f1

ONTURNBEGIN f2

ONTURNBEGIN mb1

//Assert

STATE f1

STATE f2

STATE mb1

* **Elvárt kimenet**

f1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f2

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

f2: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f1

f3

}

myceliumCapacity int = 0

spores Queue<Spore> = {

s1

s2

s3

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 2

location Tecton = f1

mushroomSpores List<Spore> = {

}

### TSZ/6. számú teszteset – Gombatest sikeres spórakilövése olyan FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton), amely a gombatest elhelyezkedése szerinti tektonnal szomszédos tekton szomszédja

* **Leírás**

Gombatest sikeres, összesen a harmadik (utolsó) spórakilövése olyan FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton), amely a gombatest elhelyezkedése szerinti tektonnal szomszédos tekton szomszédja.

(Megjegyzés: SustainingTecton, MultiLayeredTecton, AridTecton és SemiFertileTecton mint céltektonok esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + gombatest sikeres spórakilövése a vonatkozó feltételekkel: gombatest az utolsó, azaz a harmadik kilövése előtt csak valamely szomszédjára lőhet spórát. Fejlettnek az utolsó kilövése során minősül, amely abban nyilvánul meg, hogy ekkor a szomszédja szomszédjára is tud lőni
  + a gombatest a harmadik kilövését követően elpusztul (inaktívvá válik)
  + a kilőtt spórákat a továbbiakban a céltekton tartja nyilván
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton f1

CREATE\_TECTON FertileTecton {f1} f2

CREATE\_TECTON FertileTecton {f2} f3

CREATE\_MUSHROOMBODY mb1 f1

ADD\_SPORE mb1 SpeedSpore s1

ADD\_SPORE mb1 SpeedSpore s2

ADD\_SPORE mb1 SpeedSpore s3

SET\_REMAININGEJECTS mb1 1

//Act

EJECT\_SPORES mb1 f3

ONTURNBEGIN f1

ONTURNBEGIN f2

ONTURNBEGIN f3

ONTURNBEGIN mb1

//Assert

STATE f1

STATE f2

STATE f3

STATE mb1

* **Elvárt kimenet**

f1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f2

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

f2: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f1

f3

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

f3: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f2

}

myceliumCapacity int = 0

spores Queue<Spore> = {

s1

s2

s3

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 0

location Tecton = f1

mushroomSpores List<Spore> = {

}

### TSZ/7. számú teszteset – Gombatest spórahiány miatti sikertelen spórakilövése a gombatest elhelyezkedése szerinti tektonnal szomszédos FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton)

* **Leírás**

Gombatest spórakilövést kísérel meg az elhelyezkedése szerinti tektonnal szomszédos FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton), amely azonban sikertelen, tekintettel arra, hogy a gombatestnek nincsen spórája.

(Megjegyzés: SustainingTecton, MultiLayeredTecton, AridTecton és SemiFertileTecton mint céltektonok esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + gombatest sikertelen spórakilövése a vonatkozó feltételekkel: gombatest az utolsó, azaz a harmadik kilövése előtt csak valamely szomszédjára lőheti ki meglévő spórát
  + az objektumok állapotában nem következik be változás
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton f1

CREATE\_TECTON FertileTecton {f1} f2

CREATE\_MUSHROOMBODY mb1 f1

SET\_REMAININGEJECTS mb1 3

//Act

EJECT\_SPORES mb1 f2

ONTURNBEGIN f1

ONTURNBEGIN f2

ONTURNBEGIN mb1

//Assert

STATE f1

STATE f2

STATE mb1

* **Elvárt kimenet**

f1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f2

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

f2: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f1

f3

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 3

location Tecton = f1

mushroomSpores List<Spore> = {

}

### TSZ/8. számú teszteset – Gombatest sikertelen spórakilövése olyan FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton), amely a gombatest elhelyezkedése szerinti tektonnal szomszédos tekton szomszédja

* **Leírás**

Gombatest sikertelenül kísérel meg spórakilövést olyan FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton), amely a gombatest elhelyezkedése szerinti tektonnal szomszédos tekton szomszédja, mert nem ez a gombatest harmadik (összességében az utolsó) spórakilövése.

(Megjegyzés: SustainingTecton, MultiLayeredTecton, AridTecton és SemiFertileTecton mint céltektonok esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + gombatest sikertelen spórakilövése a vonatkozó feltételekkel: gombatest az utolsó, azaz a harmadik kilövése előtt csak valamely szomszédjára lőhet spórát. Fejlettnek az utolsó kilövése során minősül, amely abban nyilvánul meg, hogy ekkor a szomszédja szomszédjára is tud lőni
  + az objektumok állapotában nem következik be változás
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton f1

CREATE\_TECTON FertileTecton {f1} f2

CREATE\_TECTON FertileTecton {f2} f3

CREATE\_MUSHROOMBODY mb1 f1

ADD\_SPORE mb1 SpeedSpore s1

ADD\_SPORE mb1 SpeedSpore s2

ADD\_SPORE mb1 SpeedSpore s3

SET\_REMAININGEJECTS mb1 3

//Act

EJECT\_SPORES mb1 f3

ONTURNBEGIN f1

ONTURNBEGIN f2

ONTURNBEGIN f3

ONTURNBEGIN mb1

//Assert

STATE f1

STATE f2

STATE f3

STATE mb1

* **Elvárt kimenet**

f1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f2

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

f2: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f1

f3

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

f3: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f2

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 3

location Tecton = f1

mushroomSpores List<Spore> = {

s1

s2

s3

}

### TSZ/9. számú teszteset – Gombatest sikertelen spórakilövése olyan FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton), amely a gombatest elhelyezkedése szerinti tekton harmadik szomszédja

* **Leírás**

Gombatest sikertelenül kísérel meg spórakilövést olyan FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton, céltekton), amely a gombatest elhelyezkedése szerinti tekton harmadik szomszédja. [Azaz létezik A, B, C és D FertileTecton, amelyek a következőképpen szomszédosak (a szomszédosságot a – jelöli): A – B – C – D. (A tektonok egyéb módon nem szomszédosak egymással.) A gombatest A FertileTectonon található.]

(Megjegyzés: SustainingTecton, MultiLayeredTecton, AridTecton és SemiFertileTecton mint céltektonok esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + gombatest az utolsó, azaz a harmadik kilövése előtt csak valamely szomszédjára lőhet spórát. Fejlettnek az utolsó kilövése során minősül, amely abban nyilvánul meg, hogy ekkor a szomszédja szomszédjára is tud lőni. Érettégétől függetlenül azonban nem képes spórakilövésre az elhelyezkedése szerinti tekton harmadik szomszédjára
  + a teszt során érettnek minősülő gombatestnek az elhelyezkedése szerinti tekton harmadik szomszédja tekintetében megkísérelt spórakilövését teszteljük
  + az objektumok állapotában nem következik be változás
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton f1

CREATE\_TECTON FertileTecton {f1} f2

CREATE\_TECTON FertileTecton {f2} f3

CREATE\_TECTON FertileTecton {f3} f4

CREATE\_MUSHROOMBODY mb1 f1

ADD\_SPORE mb1 SpeedSpore s1

ADD\_SPORE mb1 SpeedSpore s2

ADD\_SPORE mb1 SpeedSpore s3

SET\_REMAININGEJECTS mb1 1

//Act

EJECT\_SPORES mb1 f3

ONTURNBEGIN f1

ONTURNBEGIN f2

ONTURNBEGIN f3

ONTURNBEGIN f4

ONTURNBEGIN mb1

//Assert

STATE f1

STATE f2

STATE f3

STATE f4

STATE mb1

* **Elvárt kimenet**

f1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f2

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

f2: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f1

f3

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

f3: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f2

f4

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

f4: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f3

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 0

location Tecton = f1

mushroomSpores List<Spore> = {

s1

s2

s3

}

### TSZ/10. számú teszteset – Elpusztult (inaktív) gombatest sikertelen spórakilövése a gombatest elhelyezkedése szerinti tektonnal szomszédos FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton)

* **Leírás**

Elpusztult (inaktív) gombatest sikertelenül próbál spórakilövést végrehajtani a gombatest elhelyezkedése szerinti tektonnal szomszédos FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton; céltekton).

(Megjegyzés: SustainingTecton, MultiLayeredTecton, AridTecton és SemiFertileTecton mint céltektonok esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + elpusztult (inaktív) gombatest semmilyen cselekvésre, így spórakilövésre sem képes (ebbe az állapotba közvetlenül a harmadik spórakilövése után kerül a gombatest, amikor már nincsen spórája)
  + az objektumok állapotában nem következik be változás
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton f1

CREATE\_TECTON FertileTecton {f1} f2

CREATE\_MUSHROOMBODY mb1 f1

DEACTIVATE mb1

//Act

EJECT\_SPORES mb1 f2

ONTURNBEGIN f1

ONTURNBEGIN f2

//Assert

STATE f1

STATE f2

STATE mb1

* **Elvárt kimenet**

f1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f2

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

f2: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

f1

}

myceliumCapacity int = 0

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 0

location Tecton = f1

mushroomSpores List<Spore> = {

}

### TSZ/11. számú teszteset – StunSpore sikeres elhelyezése FertileTectonon (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton)

* **Leírás**

Egy StunSpore sikeresen elhelyezésre kerül egy FertileTectonon (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton; céltekton).

(Megjegyzés: SustainingTecton, MultiLayeredTecton, AridTecton és SemiFertileTecton mint céltektonok, valamint PreventCutSpore, SpeedSpore és SlownessSpore esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + a spóra elhelyezése következtében a céltekton az elhelyezett spórát nyilvántartásba veszi
  + az objektumok állapota egyebekben nem változik
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton f1

//Act

PUT\_SPOREStunSpore s1 f1

ONTURNBEGIN f1

//Assert

STATE f1

* **Elvárt kimenet**

f1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 0

spores Queue<Spore> = {

s1

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}