## 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 inaktívvá válik.

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

***Új parancs***

**ADD\_SPORE** Spore\_TypeSpore\_Name Mushroomody\_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 ft1

SET\_BREAKTIMER ft1 5

PUT\_SPORE SpeedSpore speeds1 ft1

PUT\_SPORE SpeedSpore speeds2 ft1

PUT\_SPORE SpeedSpore speeds3 ft1

ADD\_PLAYER Mycologist mycologist1

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 ft1

//Act

START\_GAME

GROW\_MUSHROOMBODY mb1 ft1

SET\_REMAININGEJECTS mb1 3

ENDTURN

//Assert

STATE ft1

STATE m1

STATE mb1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds1

speeds2

speeds3

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer int = 0

mb1: MushroomBody

remainingEjects int = 3

location Tecton = ft1

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 ft1

SET\_BREAKTIMER ft1 5

CREATE\_MYCELIUM Mycelium m1

PUT\_SPORE SpeedSpore speeds1 ft1

PUT\_SPORE SpeedSpore speeds2 ft1

ADD\_PLAYER Mycologist mycologist1

ADD\_MYCELIUM\_TO\_TECTON m1 ft1

//Act

START\_GAME

GROW\_MUSHROOMBODY mb1 ft1

ENDTURN

//Assert

STATE ft1

STATE m1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds1

speeds2

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

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 ft1

SET\_BREAKTIMER ft1 5

PUT\_SPORE SpeedSpore speeds1 ft1

PUT\_SPORE SpeedSpore speeds2 ft1

PUT\_SPORE SpeedSpore speeds3 ft1

ADD\_PLAYER Mycologist mycologist1

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 ft1

CREATE\_MUSHROOMBODY mb1 ft1

//Act

START\_GAME

GROW\_MUSHROOMBODY mb2 ft1

ENDTURN

//Assert

STATE ft1

STATE m1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds1

speeds2

speeds3

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer int = 0

mb1: MushroomBody

remainingEjects int = 3

location Tecton = ft1

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 sft1

SET\_BREAKTIMER sft1 5

PUT\_SPORE SpeedSpore speeds1 sft1

PUT\_SPORE SpeedSpore speeds2 sft1

PUT\_SPORE SpeedSpore speeds3 sft1

ADD\_PLAYER Mycologist mycologist1

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 sft1

//Act

START\_GAME

GROW\_MUSHROOMBODY mb1 sft1

ENDTURN

//Assert

STATE sft1

STATE m1

* **Elvárt kimenet**

ft1: SemiFertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds1

speeds2

speeds3

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = false

location Tecton = sft1

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 ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

ADD\_NEIGHBOUR f1 f2

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

ADD\_SPORE SpeedSpore speeds1 mb1

ADD\_SPORE SpeedSpore speeds2 mb1

ADD\_SPORE SpeedSpore speeds3 mb1

SET\_REMAININGEJECTS mb1 3

//Act

START\_GAME

EJECT\_SPORES mb1 ft2

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE mb1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft1

ft3

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds1

speeds2

speeds3

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 2

location Tecton = ft1

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 inaktívvá válik
  + a kilőtt spórákat a továbbiakban a céltekton tartja nyilván
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

CREATE\_TECTON FertileTecton ft3

SET\_BREAKTIMER ft3 5

ADD\_NEIGHBOUR ft1 ft2

ADD\_NEIGHBOUR ft2 ft3

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

ADD\_SPORE SpeedSpore speeds1 mb1

ADD\_SPORE SpeedSpore speeds2 mb1

ADD\_SPORE SpeedSpore speeds3 mb1

SET\_REMAININGEJECTS mb1 1

//Act

START\_GAME

EJECT\_SPORES mb1 ft3

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE ft3

STATE mb1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft1

ft3

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft3: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds1

speeds2

speeds3

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 0

location Tecton = ft1

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 ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

ADD\_NEIGHBOUR ft1 ft2

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

SET\_REMAININGEJECTS mb1 3

//Act

START\_GAME

EJECT\_SPORES mb1 ft2

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE mb1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft1

ft3

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 3

location Tecton = ft1

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 ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

CREATE\_TECTON FertileTecton ft3

SET\_BREAKTIMER ft3 5

ADD\_NEIGHBOUR ft1 ft2

ADD\_NEIGHBOUR ft2 ft3

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

ADD\_SPORE SpeedSpore speeds1 mb1

ADD\_SPORE SpeedSpore speeds2 mb1

ADD\_SPORE SpeedSpore speeds3 mb1

SET\_REMAININGEJECTS mb1 3

//Act

START\_GAME

EJECT\_SPORES mb1 ft3

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE ft3

STATE mb1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft1

ft3

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft3: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 3

location Tecton = ft1

mushroomSpores List<Spore> = {

speeds1

speeds2

speeds3

}

### 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 ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

CREATE\_TECTON FertileTecton ft3

SET\_BREAKTIMER ft3 5

CREATE\_TECTON FertileTecton ft4

SET\_BREAKTIMER ft4 5

ADD\_NEIGHBOUR ft1 ft2

ADD\_NEIGHBOUR ft2 ft3

ADD\_NEIGHBOUR ft3 ft4

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

ADD\_SPORE SpeedSpore speeds1 mb1

ADD\_SPORE SpeedSpore speeds2 mb1

ADD\_SPORE SpeedSpore speeds3 mb1

SET\_REMAININGEJECTS mb1 1

//Act

START\_GAME

EJECT\_SPORES mb1 ft3

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE ft3

STATE ft4

STATE mb1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft1

ft3

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft3: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft2

ft4

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft4: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft3

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 0

location Tecton = ft1

mushroomSpores List<Spore> = {

speeds1

speeds2

speeds3

}

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

* **Leírás**

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**
  + 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 ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

ADD\_NEIGHBOUR ft1 ft2

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

DEACTIVATE mb1

//Act

START\_GAME

EJECT\_SPORES mb1 ft2

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE mb1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

remainingEjects int = 0

location Tecton = ft1

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 SplitSpore, 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 ft1

SET\_BREAKTIMER ft1 5

//Act

PUT\_SPOREStunSpore stuns1 ft1

//Assert

STATE ft1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

stuns1

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

### TSZ/12. számú teszteset – Gombatest inaktívvá válása a harmadik spórakilövését követően

* **Leírás**

Gombatest három spórakilövését követően inaktívvá válik.

Egy FertileTectonon (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton; céltekton) található gombatest mindhárom alkalommal egy darab SpeedSpore típusú spórát lő ki (funkcionálisan mindig az összes spóráját kilövi; körönként egy-egy SpeedSpore-t termel) különböző FertileTectonokra (fenti jellegű; céltekton). A kilövések során a céltekton szomszédos. A rovar tektonról-tektonra mozogva bejárja a pályát és közben nem eszik spórát, nem rág el fonalat.

(Megjegyzés: SustainingTecton, MultiLayeredTecton és AridTecton mint a gombatest elhelyezkedése szerinti tekton és céltekton, CarnivorousMycelium, valamint SplitSpore, StunSpore, PreventCutSpore és SlownessSpore esetén a teszt hasonlóképpen működik, mutatis mutandis.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + a gombatest a harmadik spórakilövését követően inaktívvá válik
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

CREATE\_TECTON FertileTecton ft3

SET\_BREAKTIMER ft3 5

CREATE\_TECTON FertileTecton ft4

SET\_BREAKTIMER ft4 5

ADD\_NEIGHBOUR ft1 ft2

ADD\_NEIGHBOUR ft1 ft3

ADD\_NEIGHBOUR ft1 ft4

ADD\_NEIGHBOUR ft2 ft3

ADD\_NEIGHBOUR ft3 ft4

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

ADD\_SPORE SpeedSpore speeds1 mb1

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 ft1

CREATE\_MYCELIUM Mycelium m2

ADD\_MYCELIUM\_TO\_TECTON m2 ft2

CREATE\_MYCELIUM Mycelium m3

ADD\_MYCELIUM\_TO\_TECTON m3 ft3

CREATE\_MYCELIUM Mycelium m4

ADD\_MYCELIUM\_TO\_TECTON m4 ft4

ADD\_PLAYER Entomologist entomologist1

CREATE\_INSECT i1 f1

//Act

START\_GAME

EJECT\_SPORES mb1 ft2

ENDTURN

MOVE i1 ft2

ENDTURN

EJECT\_SPORES mb1 ft3

STATE ft3 és elvárt kimenet az első kör után

ENDTURN

MOVE i1 ft3

ENDTURN

EJECT\_SPORES mb1 ft4

ENDTURN

MOVE i1 ft4

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE ft3

STATE ft4

STATE mb1

STATE m1

STATE m2

STATE m3

STATE m4

STATE i1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 2

neighbours List<Tecton> = {

ft2

ft3

ft4

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 2

neighbours List<Tecton> = {

ft1

ft3

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds1

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m2

}

occupants List<Insect> = {

}

ft3: FertileTecon

breakTimer int = 2

neighbours List<Tecton> = {

ft1

ft2

ft4

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds2

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m3

}

occupants List<Insect> = {

}

ft4: FertileTecon

breakTimer int = 2

neighbours List<Tecton> = {

ft1

ft3

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds3

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m4

}

occupants List<Insect> = {

i1

}

mb1: MushroomBody

remainingEjects int = 0

location Tecton = ft1

mushroomSpores List<Spore> = {

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer int = 0

m2: Mycelium

growing boolean = false

location Tecton = ft2

growTimer int = 0

m3: Mycelium

growing boolean = false

location Tecton = ft3

growTimer int = 0

m4: Mycelium

growing boolean = false

location Tecton = ft4

growTimer int = 0

i1: Insect

location = ft4

maxMoves = 2

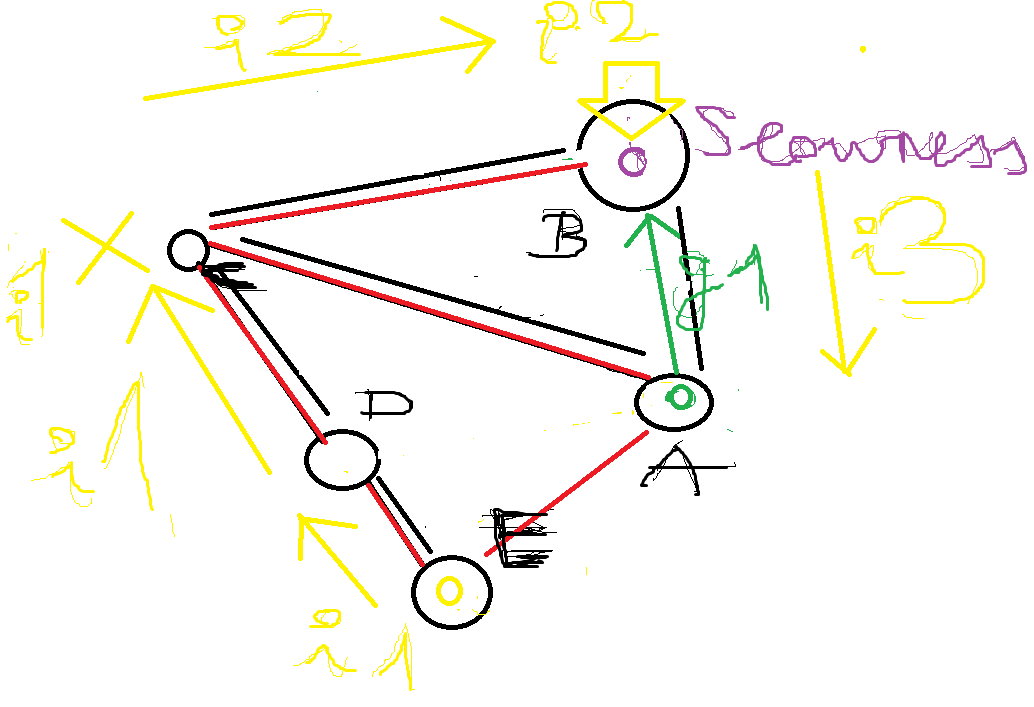
remainingMoves = 0

sporesEaten = 0

effectTimer = 0

state = Normal

### TSZ/13. számú teszteset – Rovar által elvágott gombafonál elsorvadása és az elfogyasztott spóra hatása



Ezt akartam ide példának felhozni. A fekete kör a tekton, a zöld a gombatest, a sárga a rovar, lila a spóra. A piros vonalak a meglévő fonalak, a feketék a szomszédságok. Ez a kiindulóállapot.

1. A rovar az első köre végén elvág egy gombafonalat a C tektonon állva, ezt jelzi az X. Ez két kör alatt pusztul el. Az első körben a gombatest növeszt egy fonalat a B tektonra (zölddel).
2. A rovar a B tektonra jön a 2. körben (csak egyet mozog), megeszi az ott lévő SlownessSpore-t (ezt jelzi a kettős nyíl), ettől lelassul.
3. A rovar a 3. körben az A tektonra érkezik (csak egyet mozog, de csak egyet is tudna). Ekkorra elpusztul az elrágott gombafonál.
4. Ezt már nem ábrázoltam.

Közben kilövi a spóráit a gombatest a C tektonra, ahová megérkezik a rovar a 4. körben. Addigra elmúlik a spóra hatása. Ott a rovar megeszi az egyik spórát. A végén még egy tektontörést is behozhatok akár, hogy teljessé váljon a csoda!

**Kérdések**

* A rovar kettőt mozoghat egy körben és még ehet vagy vághat. A sorrend lényeges? Lehet az h mozog, eszik, mozog? Vagy a sorrend kötött, pl. előbb kell h mozogjon, és ha eszik vagy vág, többet nem mozoghat? (Erről már volt szó, tudom, de érdemes tisztázni.)

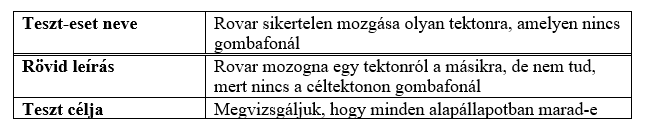
Lásd Gergőnél, tehát kettő mozog, vagy eszik vagy vág vagy egy mozog és egyet vág vagy eszik.

* Hogyan lehet A-ról B-re gombafonalat növeszteni? A szintaktikánkkal nem tudok már ide tenni. Szóval, a kódban meg kéne azt is adni, hogy melyik két tekton közé kerül a fonál.

Sehogy. Ha van három szomszédos tektonon fonál

* A rovar a C tektonon állva rág. Mit vág ő ott el? Azon a tektonon 3 gombafonál is találkozik. Ha mind a hármat, az nem lehet, mert csak egyet vághat el. Ha csak egyet, akkor meg kéne mondani, hogy melyik két tekton közöttit vágja el. Ez ellentmond annak, amit mondtatok h a fonalak a tektonokon vannak és nem tektonok között.

Ha azonban úgy van h ha van egy fonál két tektonon, akkor ezt úgy kell venni h közöttük (is) halad ez a fonál, akkor plusz feltételként vegyük bele, és a rovar mozgásába is, hogy ez csak akkor igaz, ha a két tekton szomszédos. Tehát, ebben az esetben tényleg nem lehet már fonalat tenni A és B közé, mert azzal h van fonál A-n, B-n és C-n, úgy kell venni h egymás között is megy ez a fonál, mivel ezek a tektonok kölcsönösen szomszédosak. Ennél a példánál maradva, ha A és E nem lenne szomszédos, hiába lenne mindkettőn fonál, a rovar nem tudna mozogni közvetlenül e két tekton között. Az előző leadandóban volt egy ilyen teszteset, ez pl. a szomszédosságot nem vizsgálta, csak a gombafonál meglétét:



egy tektonon bárhány lehet, a több fonál biztonsági célokat szolgálhat, más célokat nem szolgál. A legelőször megnövesztettet fogja elvágni

* Az nem lényeges h ki növesztette a fonalat? Melyik játékos? Melyik gombatest? Az az egyszerű, ha nem jelöljük.

nem számít

* A következő tesztesetet is itt ábrázolom, de akár a 12-essel is egybe lehet vonni, így legalább a 12-es is izgalmasabb lesz!

ok

* Ha a fonalak csak a tektonokon vannak és nem a tektonok között, a C tektonon egy vágás csak akkor eredményezne fonalelhalást, éspedig pl. az E-n lévő fonalét, ha A és E tekton nem szomszédos, mert akkor közöttük nem nőhetett volna fonál.

oké

* Ha a fonalak csak a tektonokon vannak és nem a tektonok között, akkor nincs értelme a MultiLayered-nek, tekintve azt is, hogy az o.katalógusból kihúztuk ezt a részt: „Új gombatestet olyan gombafonál(rész) növeszthet, amely összeköttetésben áll az eredeti gombatestével.” Ui. minden gombafonál minden gombatesthez tartozik, ha van kapcsolat (szomszédosság az érintett tektonon). Így egy gombatest a távolban is növeszthet fonalat, ha van odáig folytonosság, még ha a köztes fonalakat nem is ő növesztette!

lásd fent

tektontörés nem kell

### TSZ/14. számú teszteset – „A rovarra tényleg megfelelő ideig tartanak a státuszok miután megeszik és spórát!”