# Részletes tervek

[A dokumentum célja, hogy pontosan specifikálja az implementálandó osztályokat, beleértve a privát attribútumokat és metódusokat, ezek definícióját is.

A dokumentum második fele részletesen be kell mutassa a korábban definiált be- és kimeneti nyelv szintakszisát felhasználva, hogy mely tesztekkel lesz a prototípus ellenőrizve.]

## Osztályok és metódusok tervei.

### MyceliumImpl

#### Felelősség

A játékban a gombafonalakat reprezentáló osztály. Felelős a gombafonal növési folyamatban a növés gyorságáért, a gombafonalak elvágása esetén pedig a fonál kitörléséért és részben a többi fonál életben maradásának ellenőrzéséért is.

#### Interfészek

Mushroom, Mycelium, TurnBeginSubscriber

#### Attribútumok

* -growing: boolean

Azt ábrázolja, hogy a gombafonál éppen növekedés alatt van-e. Alapértelmezett értéke hamis. Ha a tekton úgy dönt, hogy nőhet rajta a fonál akkor igaz lesz.

* -location: Tecton

A tekton, ahol a gombafonál elhelyezkedik.

* -growTimer: int

Az idő, ami alatt a gombafonál megnő. Alapértelmezett értéke 0. Növéskor, ha a céltektonon van spóra, akkor ez az érték 1, ha nincs akkor 2 lesz.

#### Metódusok

* +Mycelium(location: FertileTecton, name: String)

Konstruktor, amely beállítja a létrehozandó gombafonál nevét és azt a tektont (céltekton), amelyen az elhelyezésre kerül. Ez a konstruktor használandó FertileTectonon, továbbá a FertileTecton valamennyi leszármazottjánál, azaz AridTecton, MultiLayeredTecton és SustainingTecton esetén.

procedure Mycelium(location: FertileTecton, name: String)

// A fonál helyét beállítja a kapott helynek

SET this.location ← location

// létrejön egy új MyceliumGrowthEvaluator aminek paraméterként beadjuk a Myceliumot

myceliumGrowthEvaluator ← CREATE MyceliumGrowthEvaluator(this)

//A fonálnövesztés feltételeinek ellenőrzésére meghívjuk a myceliumGrowthEvaluator visit

//függvényét, ami majd a location Tektonnal kommunikál

CALL myceliumGrowthEvaluator.visit(location)

END

* +Mycelium(location: SemiFertileTecton, name: String)

Konstruktor, amely beállítja a létrehozandó gombafonál nevét és azt a tektont (céltekton), amelyen az elhelyezésre kerül. Ez a konstruktor használandó SemiFertileTectonon.

procedure Mycelium(location: SemiFertileTecton, name: String)

// A fonál helyét beállítja a kapott helynek

SET this.location ← location

// létrejön egy új MyceliumGrowthEvaluator aminek paraméterként beadjuk a Myceliumot

myceliumGrowthEvaluator ← CREATE MyceliumGrowthEvaluator(this)

//A fonálnövesztés feltételeinek ellenőrzésére meghívjuk a myceliumGrowthEvaluator visit

//függvényét, ami majd a location Tektonnal kommunikál

CALL myceliumGrowthEvaluator.visit(location)

END

* +delete(): void

Kitörli a gombafonalat

* +grow(sporeCount: int): void

Ezt a jelzést a tekton fogja meghívni a gombafonálra, ha nőhet. A céltektonon lévő spóraszámtól függően módosítja a growtimert.

* +ontTurnBegin(): void

A growtimer visszaszámlálását végzi (minden körben eggyel csökken) és ha az lejárt, már nem lesz növésben a gombafonál.

* +isGrowing(): boolean

A growing attribútum getterje

* setGrowing(growing: boolean): void

A growing attribútum setterje

* +cut(): void

A gombafonál elvágódik, ezzel szól a többi gombafonálnak, hogy nézzék meg, hogy hozzá vannak-e csatlakoztatva gombatesthez

* +getLocation(): Tecton

A location attribútum getterje

* +setLocation(location: Tecton): void

A location attribútum getterje

### CarnivorousMycelium

#### Felelősség

Az alapvető gombafonál funkciókon kívül speciális feltételek között a rovarak evését és fonál növesztését megvalósító osztály.

#### Interfészek

Mushroom, Mycelium, TurnBeginSubscriber

#### Attribútumok

* -growing: boolean

Azt ábrázolja, hogy a gombafonál éppen növekedés alatt van-e. Alapértelmezett értéke hamis. Ha a tekton úgy dönt, hogy nőhet rajta a fonál akkor igaz lesz.

* -location: Tecton

A tekton, ahol a gombafonál elhelyezkedik.

* -growTimer: int

Az idő, ami alatt a gombafonál megnő. Alapértelmezett értéke 0. Növéskor, ha a céltektonon van spóra, akkor ez az érték 1, ha nincs akkor 2 lesz.

#### Metódusok

* +Mycelium(location: FertileTecton, name: String)

Konstruktor, amely beállítja a létrehozandó gombafonál nevét és azt a tektont (céltekton), amelyen az elhelyezésre kerül. Ez a konstruktor használandó FertileTectonon, továbbá a FertileTecton valamennyi leszármazottjánál, azaz AridTecton, MultiLayeredTecton és SustainingTecton esetén.

procedure CarnivorousMycelium(location: FertileTecton, name: String)

// A fonál helyét beállítja a kapott helynek

SET this.location ← location

// létrejön egy új MyceliumGrowthEvaluator aminek paraméterként beadjuk a Myceliumot

myceliumGrowthEvaluator ← CREATE MyceliumGrowthEvaluator(this)

//A fonálnövesztés feltételeinek ellenőrzésére meghívjuk a myceliumGrowthEvaluator visit

//függvényét, ami majd a location Tektonnal kommunikál

CALL myceliumGrowthEvaluator.visit(location)

END

* +CarnivorousMycelium(location: SemiFertileTecton, name: String)

Konstruktor, amely beállítja a létrehozandó gombafonál nevét és azt a tektont (céltekton), amelyen az elhelyezésre kerül. Ez a konstruktor használandó SemiFertileTectonon.

procedure Mycelium(location: SemiFertileTecton, name: String)

// A fonál helyét beállítja a kapott helynek

SET this.location ← location

// létrejön egy új MyceliumGrowthEvaluator aminek paraméterként beadjuk a Myceliumot

myceliumGrowthEvaluator ← CREATE MyceliumGrowthEvaluator(this)

//A fonálnövesztés feltételeinek ellenőrzésére meghívjuk a myceliumGrowthEvaluator visit

//függvényét, ami majd a location Tektonnal kommunikál

CALL myceliumGrowthEvaluator.visit(location)

END

* +delete(): void

Kitörli a gombafonalat

* +grow(sporeCount: int): void

Ezt a jelzést a tekton fogja meghívni a gombafonálra, ha nőhet. A céltektonon lévő spóraszámtól függően módosítja a growtimert.

* +ontTurnBegin(): void

A growtimer visszaszámlálását végzi (minden körben eggyel csökken) és ha az lejárt, már nem lesz növésben a gombafonál. Ha a tektonján levő rovarok Stunned állapotban vannak megöli a rovarokat és egy gombát növesztését kezdeményezi.

* +isGrowing(): boolean

A growing attribútum getterje

* setGrowing(growing: boolean): void

A growing attribútum setterje

* +cut(): void

A gombafonál elvágódik, ezzel szól a többi gombafonálnak, hogy nézzék meg, hogy hozzá vannak-e csatlakoztatva gombatesthez

* +getLocation(): Tecton

A location attribútum getterje

* +setLocation(location: Tecton): void

A location attribútum getterje

### MyceliumGrowthEvaluator

#### Felelősség

Egy segédosztály, ami annak az eldöntésében segít, hogy egy adott tektonra nőhet-e az őt létrehozó gombafonál.

#### Interfészek

TectonVisitor

#### Attribútumok

#### -creator: Mycelium

Ezt az objektumot létrehozó gombafonál.

#### Metódusok

* + MyceliumGrowthEvaluator(mushroom: Mycelium)

Létrehozáskor meg kell adni, hogy melyik gombafonál hozta őt létre.

* +visit(tecton: FertileTecton): void

Megkér egy "Fertile" típusú tektont, hogy "fogadja be" az objektumot létrehozó gombafonalat. A tekton ezt a kérést az accept függvényében elbírálja és visszajelez a fonálnak.

* +visit(tecton: MultiLayeredTecton): void

Megkér egy "MultiLayered" típusú tektont, hogy "fogadja be" az objektumot létrehozó gombafonalat. A tekton ezt a kérést az accept függvényében elbírálja és visszajelez a fonálnak.

* +visit(tecton: AridTectton): void

Megkér egy "Arid" típusú tektont, hogy "fogadja be" az objektumot létrehozó gombafonalat. A tekton ezt a kérést az accept függvényében elbírálja és visszajelez a fonálnak.

* +visit(tecton: SemiFertileTecton): void

Megkér egy "SemiFertile" típusú tektont, hogy "fogadja be" az objektumot létrehozó gombafonalat. A tekton ezt a kérést az accept függvényében elbírálja és visszajelez a fonálnak.

* +visit(tecton: SustainingTecton): void

Megkér egy "Sustaining" típusú tektont, hogy "fogadja be" az objektumot létrehozó gombafonalat. A tekton ezt a kérést az accept függvényében elbírálja és visszajelez a fonálnak.

* +getCreator(): Mycelium

A creator attribútum getterje

### MushroombodyGrowthEvaluator

#### Felelősség

Egy segédosztály, ami annak az eldöntésében segít, hogy egy adott tektonra nőhet-e az őt létrehozó gombatest.

#### Interfészek

TectonVisitor

#### Attribútumok

#### -creator: MushroomBody

Ezt az objektumot létrehozó gombafonál.

#### Metódusok

* + MushrooomBodyGrowthEvaluator(mushroom: MushroomBody)

Létrehozáskor meg kell adni, hogy melyik gombatest hozta őt létre.

* +visit(tecton: FertileTecton): void

Megkér egy "Fertile" típusú tektont, hogy "fogadja be" az objektumot létrehozó gombatestet. A tekton ezt a kérést az accept függvényében elbírálja és visszajelez a gombatestnek.

* +visit(tecton: MultiLayeredTecton): void

Megkér egy "MultiLayered" típusú tektont, hogy "fogadja be" az objektumot létrehozó gombatestet. A tekton ezt a kérést az accept függvényében elbírálja és visszajelez a gombatestnek.

* +visit(tecton: AridTectton): void

Megkér egy "Arid" típusú tektont, hogy "fogadja be" az objektumot létrehozó gombatestet. A tekton ezt a kérést az accept függvényében elbírálja és visszajelez a gombatestnek.

* +visit(tecton: SemiFertileTecton): void

Megkér egy " SemiFertile" típusú tektont, hogy "fogadja be" az objektumot létrehozó gombatestet. A tekton ezt a kérést az accept függvényében elbírálja és visszajelez a gombatestnek.

* +visit(tecton: SustainingTecton): void

Megkér egy " Sustaining" típusú tektont, hogy "fogadja be" az objektumot létrehozó gombatestet. A tekton ezt a kérést az accept függvényében elbírálja és visszajelez a gombatestnek.

* +getCreator(): MushroomBody

A creator attribútum getterje

## 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.]

A 16. számú use-case kapcsán

****Create\_Mycelium** Mycelium\_Type Mycelium\_Name**

**Leírás:** Létrehoz egy gombafonalat

**Opciók:** A gombafonál típusa és neve.

A 17. számú use-case kapcsán

add\_mycelium\_to\_tecton Mycelium\_Name Tecton\_Name

**Leírás:** Hozzáadja a kiválasztott fonalat a kiválasztott tektonhoz

**Opciók:** A gombafonál, ami rajta lesz a tektonon és a tekton, amin lesz a fonál

A 18. számú use-case kapcsán

Grow\_mycelium Mycelium\_**Type Mycelium\_Name** Tecton\_Name

**Leírás:** Rá-nő egy gombafonál a kiválasztott tektonra

**Opciók:** A gombafonál, amit növesztünk és a tekton, amin a gombafonál lesz

### Gombafonál sikeres (lassú) növesztése

* **Leírás**

Gombafonál sikeres (lassú) növesztése gombatestből FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton)

(Megjegyzés: a növesztés hasonlóan működik, ha a tekton, ahova növesztünk SustainingTectonMultiLayeredTecton, AridTecton SustainingTecton vagy SemiFertileTecton. Akkor is hasonló a teszteset, ha gombafonálból növesztünk.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**

A gombafonál növési folyamatát, az ahhoz szükséges feltételeket és a tektonon történt változást vizsgáljuk.

Ellenőrizzük, hogy a fonál valóban rákerült-e a tektonra.

A kiválasztott FertileTectonon nincs spóra, ezért a gombafonál lassan (2 kör alatt) fog nőni.

* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

ADD\_NEIGHBOR ft1 ft2

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

//Act

START\_GAME

Grow\_mycelium Mycelium m1 ft2

ENDTURN

STATE ft2

STATE m1

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE mb1

STATE m1

**Elvárt kimenet**

ft2: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = true

location Tecton = f2

growTimer int = 1

ft1: FertileTecon

breakTimer int = 3

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 3

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

mb1: MushroomBody

reamainingEjects int = 3

location Tecton = ft1

mushroomSpores List<Spore> = {

mb1-speeds1

mb1-speeds2

}

m1: Mycelium

growing boolean = false

location Tecton = f2

growTimer int = 0

### Gombafonál sikeres gyors növesztése

* **Leírás**

Gombafonál sikeres gyors növesztése gombatestből FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton)

(Megjegyzés: a növesztés hasonlóan működik, ha a tekton, ahova növesztünk MultiLayeredTecton, AridTecton, SustainingTecton vagy SemiFertileTecton. Akkor is hasonló a teszteset, ha gombafonálból növesztünk. A céltektonon lehetne több mint egy spóra is, ez nem változtatna a működésen.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**

A gombafonál növési folyamatát, az ahhoz szükséges feltételeket és a tektonon történt változást vizsgáljuk.

Ellenőrizzük, hogy a fonál valóban rákerült-e a tektonra.

A kiválasztott FertileTectonon van egy darab spóra, ezért a gombafonál gyorsan (1 kör alatt) fog nőni.

* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

ADD\_NEIGHBOR ft1 ft2

PUT\_SPORESpeedSpore speeds1 ft2

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

//Act

START\_GAME

Grow\_mycelium Mycelium m1 ft2

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE mb1

STATE m1

**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> = {

speeds1

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

mb1: MushroomBody

reamainingEjects int = 3

location Tecton = ft1

mushroomSpores List<Spore> = {

mb1-speeds1

}

m1: Mycelium

growing boolean = false

location Tecton = f2

growTimer int = 0

### Gombafonál sikertelen növesztése gombatestből, olyan FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton), ahol már van gombafonál

* **Leírás**

A gombafonál nem nő rá a kiválasztott FertileTectonra, mert az kiválasztott FertileTecton már „tele” van fonállal (kapacitása és rajta lévő fonalak száma egyenlő).

(Megjegyzés: a növesztés hasonlóan működik, ha a tekton, ahova növesztünk MultiLayeredTecton, AridTecton, SustainingTecton vagy SemiFertileTecton. Akkor is hasonló a teszteset, ha gombafonálból növesztünk.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**

A gombafonál növési folyamatát, az ahhoz szükséges feltételeket és a tektonon történt változást vizsgáljuk.

Ellenőrizzük, hogy a fonál valóban nem került-e a tektonra.

* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

ADD\_NEIGHBOR ft1 ft2

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 ft2

//Act

START\_GAME

Grow\_mycelium Mycelium m2 ft2

ENDTURN

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE mb1

STATE m1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 3

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 3

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

mb1: MushroomBody

reamainingEjects int = 3

location Tecton = ft1

mushroomSpores List<Spore> = {

mb1-speeds1

mb1-speeds2

}

m1: Mycelium

growing boolean = false

location Tecton = f2

growTimer int = 0

### Gombafonál sikertelen növesztése gombatestből, olyan FertileTectonra (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton), ami a növést kezdeményező gombatest tektonjával nem közvetlenül szomszédos.

* **Leírás**

A gombafonál nem nő rá a kiválasztott FertileTectonra, mert az nem szomszédja a másik tektonnak, amin van a növesztést kezdeményező gombatest. Az igazi játékban nem kell kiválasztani a növést kezdeményező gombát csak azt, hogy hova növesztünk és ellenőrizzük, hogy oda tudna-e gomba növeszteni.

(Megjegyzés: a növesztés hasonlóan működik, ha a tekton, ahova növesztünk MultiLayeredTecton, AridTecton, SustainingTecton vagy SemiFertileTecton. Akkor is hasonló a teszteset, ha gombafonálból növesztünk.)

* **Ellenőrzött funkcionalitás, várható hibahelyek**

A gombafonál növési folyamatát, az ahhoz szükséges feltételeket és a tektonon történt változást vizsgáljuk.

Ellenőrizzük, hogy a fonál valóban nem került-e a tektonra.

* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

//Act

START\_GAME

Grow\_mycelium Mycelium m1 ft2

ENDTURN

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE mb1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

reamainingEjects int = 3

location Tecton = ft1

mushroomSpores List<Spore> = {

mb1-speeds1

mb1-speeds2

}

### Húsevő fonál általi rovarevés és gombatest növesztés

* **Leírás**

Új kör kezdetekor a FertileTectonon (nem SustainingTecton, nem MultiLayeredTecton és nem AridTecton) lévő húsevő fonál megeszi a rajta lévő bénult állapotban lévő rovarokat és gombatestet növeszt. (Jelen esetben a vizsgált FertileTectonon még nincs gombatest.)

(Megjegyzés: a művelet hasonlóan működik, ha a tekton, ahol a gombafonál van MultiLayeredTecton, AridTecton SustainingTecton.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Megvizsgáljuk a CarnivorousMycelium működését, ha teljesülnek az ahhoz szükséges feltételek. Valamint azt, hogy megtörténik-e az ebből következő gombatest növesztés és rovarok halála.

A FertileTectonon meg kell halni a rovaroknak és nőnie kell egy új gombatestnek.

* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

PUT\_SPOREStunSpore stuns1 ft1

ADD\_PLAYER Mycologist mycologist1

CREATE\_MYCELIUM CarnivorousMycelium cm1

ADD\_MYCELIUM\_TO\_TECTON cm1 ft1

ADD\_PLAYER Entomologist entomologist1

CREATE\_INSECT i1 ft1

//Act

START\_GAME

ENDTURN

EAT i1

ENDTURN

ENDTURN

//Assert

STATE ft1

STATE mb-ft1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb-ft1

mycelia Queue<Mycelium> = {

cm1

}

occupants List<Insect> = {

}

mb-ft1: MushroomBody

reamainingEjects int = 3

location Tecton = ft1

mushroomSpores List<Spore> = {

}

### Gombafonál elhalása AridTectonon

* **Leírás**

AridTectonon lévő fonál elpusztul, mert már 5 köre óta van ott.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Megvizsgáljuk az AridTecton többi tektontól különböző működését.

A vizsgált AridTectonon lévő fonál el kell, hogy pusztuljon.

* **Bemenet**

// Arrange

CREATE\_TECTON AridTecton at1

SET\_BREAKTIMER ft1 6

ADD\_PLAYER Mycologist mycologist1

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 at1

//Act

START\_GAME

ENDTURN

ENDTURN

ENDTURN

ENDTURN

ENDTURN

//Assert

STATE ft1

* **Elvárt kimenet**

at1: FertileTecon

breakTimer int = 1

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

### Rovarász megpróbál a rovarral műveletet (evés, vágás, mozgás) végrehajtani, amikor már nincs több művelete

* **Leírás**

A rovar alapesetben kétszer tud mozogni és egyszer tud fonalat vágni, valamint spórát enni. Ha már vágott fonalat vagy evett spórát vagy elhasználta az összes lépését, akkor nem tud utána semmilyen műveletet végrehajtani a rovarával. Ilyenkor csak átadni képes a kört a következő játékosnak.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Nem szabad, hogy a játékos bármilyen műveletet végre tudjon hajtani, ha már nincs több művelete.

* **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

PUT\_SPORESpeedSpore speeds1 ft3

ADD\_PLAYER Mycologist mycologist1

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

ADD\_PLAYER Entomologist entomologist1

CREATE\_INSECT i1 ft1

//Act

START\_GAME

MOVE i1 ft2

MOVE i1 ft3

//innentől nem valósulnak meg

MOVE i1 ft2

EAT i1

CUT i1

ENDTURN

//Assert

STATE ft3

STATE m3

STATE i1

* **Elvárt kimenet**

ft3: FertileTecon

breakTimer int = 4

neighbours List<Tecton> = {

f2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds1

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m3

}

occupants List<Insect> = {

i1

}

m3: Mycelium

growing boolean = false

location Tecton = f3

growTimer int = 0

i1: Insect

location = ft3

maxMoves = 2

remainingMoves = 0

sporesEaten = 0

effectTimer = 0

state = Normal

### Gombász megpróbál a körében olyan műveletet végezni, amire már nincs lehetősége

* **Leírás**

A gombász a saját körében 3 fajta műveletet képes végrehajtani. Fonalat tud növeszteni összesen egyszer. Minden hozzá tartozó gombatestel képes egyszer spórát lőni. Akármennyi új gombatestet tud növeszteni (a gombatest növesztési feltételeknek megfelelően).

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ha a gombász megpróbálna valamit végrehajtani a műveletre vonatkozó korlát elérése után, nem szabad, hogy képes legyen rá.

* **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 MultiLayeredTecton mlt1

SET\_BREAKTIMER mlt1 5

ADD\_NEIGHBOUR ft1 ft3

ADD\_NEIGHBOUR ft2 ft3

ADD\_NEIGHBOUR ft1 mlt1

ADD\_NEIGHBOUR ft2 mlt1

PUT\_SPORESpeedSpore speeds1 ft3

PUT\_SPORESpeedSpore speeds2 mlt1

PUT\_SPORESpeedSpore speeds3 mlt1

PUT\_SPORESpeedSpore speeds4 mlt1

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

CREATE\_MUSHROOMBODY mb2 ft2

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 mlt1

ADD\_SPORE SpeedSpore speeds5 mb1

ADD\_SPORE SpeedSpore speeds6 mb2

//Act

START\_GAME

Grow\_mycelium Mycelium m2 ft3

EJECT\_SPORES mb1 ft3

EJECT\_SPORES mb2 ft3

GROW\_MUSHROOMBODY mb3 ft3

GROW\_MUSHROOMBODY mb4 mlt1

Grow\_mycelium Mycelium m3 mlt1

EJECT\_SPORES mb1 mlt1

ENDTURN

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE ft3

STATE mlt1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 3

neighbours List<Tecton> = {

f3

f4

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 3

neighbours List<Tecton> = {

f3

f4

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb2

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft3: FertileTecon

breakTimer int = 3

neighbours List<Tecton> = {

f1

f2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

speeds1

speeds5

speeds6

}

mushroomBody MushroomBody = mb3

mycelia Queue<Mycelium> = {

m2

}

occupants List<Insect> = {

}

mlt1: MultiLayeredTecon

breakTimer int = 3

neighbours List<Tecton> = {

f1

f2

}

myceliumCapacity int = 3

spores Queue<Spore> = {

speeds2

speeds3

speeds4

}

mushroomBody MushroomBody = mb4

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

### Összetett teszteset, amiben rovarász és gombász és is van és a játék a valósághoz hasonlóan megy.

* **Leírás**

A rovarász a rovarjával mozog vág és eszik. A gombász pedig egyszer lő ki spórát.

* **Ellenőrzött funkcionalitás, várható hibahelyek**
* **Bemenet**

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 2

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft2 5

CREATE\_TECTON FertileTecton ft3

SET\_BREAKTIMER ft3 5

CREATE\_TECTON MultiLayeredTecton mlt1

SET\_BREAKTIMER mlt1 5

ADD\_NEIGHBOUR ft1 mlt1

ADD\_NEIGHBOUR ft2 mlt1

ADD\_NEIGHBOUR ft3 mlt1

ADD\_PLAYER Entomologist entomologist1

CREATE\_INSECT i1 ft1

ADD\_PLAYER Mycologist mycologist1

CREATE\_MUSHROOMBODY mb1 ft1

CREATE\_MYCELIUM Mycelium m1

ADD\_MYCELIUM\_TO\_TECTON m1 ft1

CREATE\_MYCELIUM Mycelium m2

ADD\_MYCELIUM\_TO\_TECTON m2 mlt1

CREATE\_MYCELIUM Mycelium m3

ADD\_MYCELIUM\_TO\_TECTON m3 mlt1

CREATE\_MYCELIUM Mycelium m4

ADD\_MYCELIUM\_TO\_TECTON m4 ft2

ADD\_SPORE SpeedSpore speeds1 mb1

//Act

START\_GAME

MOVE i1 mlt1

CUT i1

ENDTURN

EJECT\_SPORES mb1 mlt1

ENDTURN

EAT i1

ENDTURN

ENDTURN

//Assert

STATE ft1

STATE ft2

STATE ft3

STATE mlt1

STATE ft1-1

STATE mb1

STATE i1

* **Elvárt kimenet**

ft1: FertileTecon

breakTimer int = 0

neighbours List<Tecton> = {

mlt1

f1-1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecon

breakTimer int = 3

neighbours List<Tecton> = {

mlt1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m4

}

occupants List<Insect> = {

}

ft3: FertileTecon

breakTimer int = 3

neighbours List<Tecton> = {

mlt1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mlt1: MultiLayeredTecton

breakTimer int = 3

neighbours List<Tecton> = {

ft1

ft2

ft3

}

myceliumCapacity int = 3

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m3

}

occupants List<Insect> = {

i1

}

ft1-1: FertileTecon

breakTimer int = 0

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

mb1: MushroomBody

reamainingEjects int = 2

location Tecton = ft1

mushroomSpores List<Spore> = {

ft1- speeds1

ft1-speeds2

}

i1: Insect

location = mlt1

maxMoves = 3

remainingMoves = 0

sporesEaten = 1

effectTimer = 3

state = Fast

## A tesztelést támogató programok tervei

[A tesztadatok előállítására, a tesztek eredményeinek kiértékelésére szolgáló segédprogramok részletes terveit kell elkészíteni.]

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2025.04.06 16:00 | 2 óra | Guzmics | Teszt leírások elkezdése |
| 2025.04.08 18:00 | 2 óra | Guzmics | Teszt leírások folytatása |
| 2025.04.10 16:00 | 2 óra | Guzmics | Alapvető teszt leírások befejezése |
| 2025.04.11 19:00 | 2 óra | Guzmics | Összetett teszt leírások és eddigi tesztek javítása |
| 2025.04.13 13:00 | 2 óra | Guzmics | Objektumleírások készítése |

Ötlet/jegyzet:

Fonalat növeszteni összesen egyszer tud

Spórát kilőni 1szer testenként

Végtelen testet tud növeszteni

Kérdés: