# 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.

### InsectImpl

#### Felelősség

Egy rovarral kapcsolatos adatok keze

#### Interfészek

Insect, InsectMover, SporeEater, TurnBeginSubscriber

#### Attribútumok

* **-location: Tecton**

A tekton, amin a rovar van

* **-maxMoves: int**

Egy körön belüli maximális lépéseinek száma

* **-remainingMoves: int**

Az aktuális körben maradt lépések száma

* **-sporesEaten: int**

Megevett spórák száma

* **-effectTimer: int**

Ha van spórából származó állapot a rovaron, ez a visszaszámláló, hogy mikor jár le ennek az ideje

* **-state: InsectState**

Az aktuális állapota a rovarnak

* **-splitNum: int**

Az szakadások száma

#### Metódusok

* +**Insect(t: Tecton):** Insect konstruktora. Az insect a t tectonra fog létrejönni.

Alapértelmezett értékek:

-maxMoves: 2

-remainingMoves: maxMoves

-sporesEaten: 0

-effectTimer: 0

-state: Normal

* **+Tecton getLocation():** visszaadja a location-t
* **+setLocation(Tecton t):** beállítja a location-t
* **+int getMaxMoves():** visszaadja a MaxMoves-t
* **+setMaxMoves(i: int):** beállítja a maxMoves-t
* **+int getRemainingMoves():** visszaadja a remainingMoves-t
* **+setRemainingMoves(i: int):** beállítja a remainingMoves-t
* **+int getSporesEaten():** visszaadja a sporesEaten-t
* **+setSporesEaten(i: int):** beállítja a sporesEaten-t
* **+int getEffectTimer():** visszaadja az effectTimer-t
* **+setEffectTimer(i: int):** beállítja az effectTimer-t
* **+InsectState getState():** visszaadja a state-t
* **+setState(newState: InsectState):** beállítja a state-t
* **+int getSplitNum():** visszaadja a splitNum-ot
* **+setSplitNum(i: int):** beállítja a splitNum-ot
* **+cutMycelium():** elvágja az első gombafonalat a tectonján – meghívja a tektonon a cutMycelium() műveletet
* **+eatSpore(): megpróbálja** megenni a legelső spórát a tectonján - meghívja a tektonon a eatSpore(i: Insect) műveletet, ahol i önmaga lesz
* **+move (t: Tecton**): megpróbál elmenni a t tectonra – meghívja a tektonon a moveInsect(i: Insect) műveletet, ahol i önmaga lesz
* **+onTurnBegin():** a játékos körének elején történő dolgoknak létezik;

Pszeudokód:

Procedure onTurnBegin():

IF effectTimer BIGGER THAN 0 begin

effectTimer = effectTimer – 1

IF effectTimer IS SMALLER THAN OR EQUAL TO 0 begin

this.setState(InsectState.Normal)

this.setMaxMoves(2)

end

end

setRemainingMoves(getMaxMoves())

end procedure

* **+beSlow():** Slow állapotba állítja a rovart, és beállítja a maxMoves-t 1-re
* **+beFast():** Fast állapotba állítja a rovart, és beállítja a maxMoves-t 3-ra
* **+preventCut():** CannotCut állapotba állítja a rovart
* **+beStunned():** Stun állapotba állítja a rovart, és beállítja a maxMoves-t 0ra
* **+split():** Kettészakitja a rovart (létrejön egy új a tectonján)
* +**runAway():** elmenekül egy, véletlenszerűen kiválasztott, alkalmas tektonra

Pszeudokód:

Procedure runAway():

Set<Tecton> available

Queue<Tecton> queue

Set<Tecton> visited

ENQUEUE getLocation() INTO queue

ADD getLocation() TO visited

WHILE queue IS NOT EMPTY begin

Tecton current = DEQUEUE from queue

Boolean hasMycelium

hasMycelium = current.hasMycelium()

IF hasMycelium IS TRUE begin

ADD current TO available

End

FOR EACH neighbour IN location.getNeighbours() begin

IF (ADD neighbour TO visited) IS TRUE begin

ENQUEUE neighbour INTO queue

end

end

end

IF available IS EMTPY begin

return

end

Tecton selectedTecton = NULL

int item = RANDOM int FROM BETWEEN 0 and (SIZE OF available)

int i = 0

FOR EACH tecton IN available begin

IF i IS EQUAL TO item begin

selectedTecton = tecton

end

end

IF selectedTecton IS NOT NULL) begin

setLocation(selectedTecton)

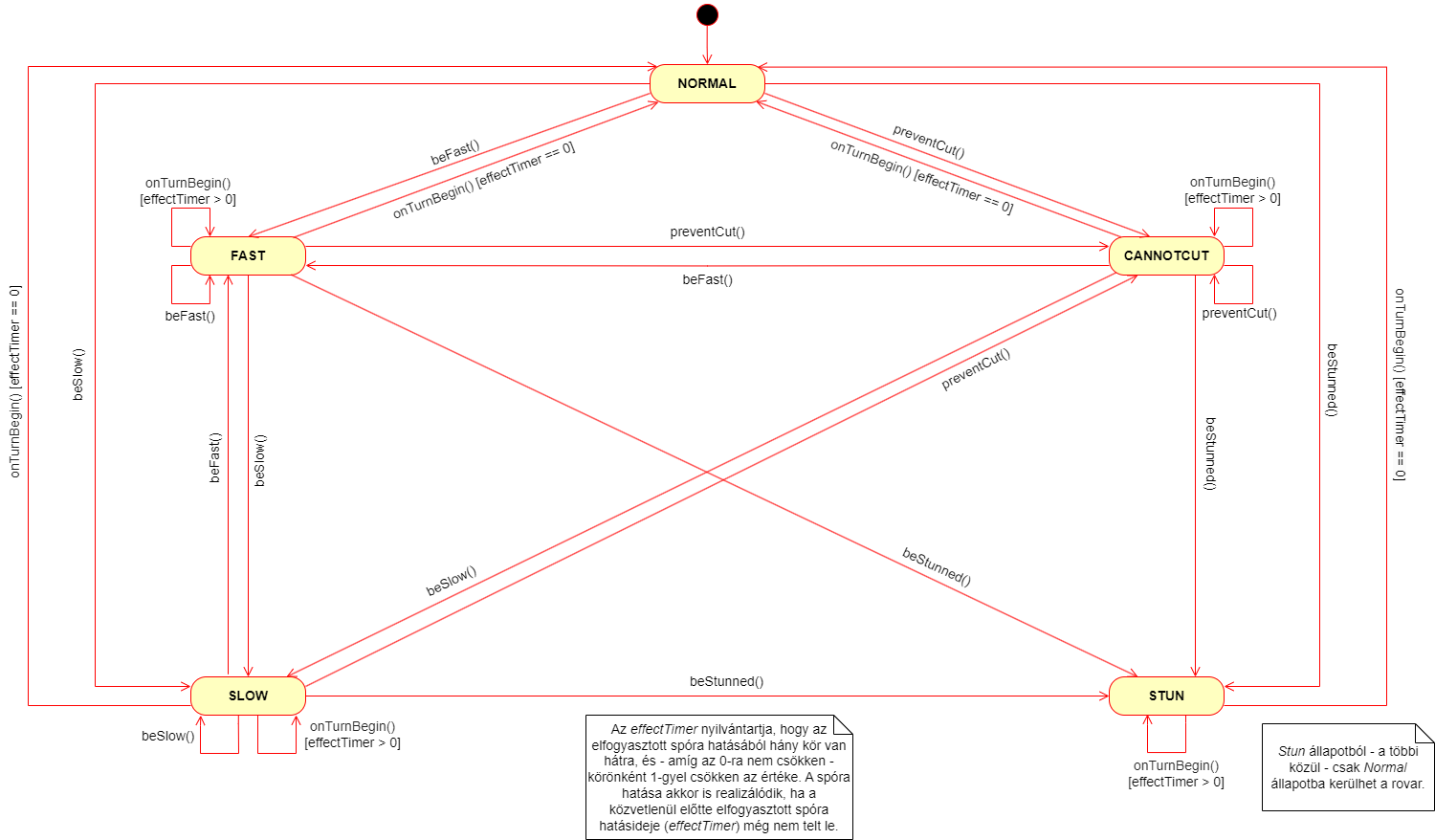
selectedTecton.addOccupant(this)

end

end procedure

* **+die():** ha Stun állapotban van,meghal (eltűnik a tektonjáról, és a location-je is null lesz, de mint object, megmarad)

#### Állapot Diagramm



### SplitSpore

#### Felelősség

Ha megette egy rovar, kettészakítja ezt.

#### Interfészek

Spore

#### Metódusok

* **+eatSpore(i: Insect):** az i rovar megette a spórát, ez meg fogja hívni a rovaron a Split() parancsot, hogy szakadjon el

### StunSpore

#### Felelősség

Ha megette egy rovar, lebénítja ezt.

#### Interfészek

Spore

#### Metódusok

* **+eatSpore(i: Insect):** az i rovar megette a spórát, ez meg fogja hívni a rovaron a beStunned() parancsot, hogy kerüljön Stun állapotba

### PreventCutSpore

#### Felelősség

Ha megette egy rovar, ez ne tudjon fonalat vágni.

#### Interfészek

Spore

#### Metódusok

* **+eatSpore(i: Insect):** az i rovar megette a spórát, ez meg fogja hívni a rovaron a PreventCut() parancsot, hogy kerüljön CannotCut állapotba

### SpeedSpore

#### Felelősség

Ha megette egy rovar, legyen több lépése

#### Interfészek

Spore

#### Metódusok

* **+eatSpore(i: Insect):** az i rovar megette a spórát, ez meg fogja hívni rajta a beFast() parancsot, hogy nagyobb legyen a remainingMoves-ja, és kerüljön Fast állapotba.

### SlownessSpore

#### Felelősség

Ha megette egy rovar, legyen kevesebb lépése.

#### Interfészek

Spore

#### Metódusok

* **+eatSpore(i: Insect):** az i rovar megette a spórát, ez meg fogja hívni rajta a beSlow() parancsot, hogy kisebb legyen a remainingMoves-ja, és kerüljön Slow állapotba.

## 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.]

Kiadható parancsok:

**CREATE\_INSECT tecton\_name insect\_name**

**Leírás:** A rovar létrejön és rákerül az argumentumként megadott céltektonra, ha ezen van gombafonál

**Megjegyzég:**

Alapértelmezett értékek:

-maxMoves: 2

-sporesEaten: 0

-effectTimer: 0

-state: Normal

**Opciók:** Argumentumok: A céltekton, ahova létrejönne, az objektum neve

**MOVE insect\_name tecton\_name**

**Leírás:** A rovar átmenne az argumentumként megadott céltektonra

**Opciók:** Argumentumok: A rovar, amelyik mozogna;

A céltekton, ahova mozogna

**EAT insect\_name**

**Leírás:** A rovar megeszi az első spórát a tektonján

**Opciók:** Argumentumok: A rovar, amelyik enne

**CUT insect\_name**

**Leírás:** A rovar elvágja az első gombafonalat a tektonján

**Opciók:** Argumentumok: A rovar, amelyik vágna

### Rovar létrehozása és letevése

* **Leírás**

Létrehozunk egy rovarat egy tektonra. Beállitja a létrehozott rovar tektonját arra amelyikre létrehozták, és ennek az occupants listájára hozzáadja az újonnan létrehozott rovart.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Új rovar létrejött-e jó változókkal, tekton-on rajta van-e*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
SET\_BREAKTIMER ft1 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
STATE ft1  
STATE m1  
STATE i1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

i1

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer = 0

deathTimer = -1

i1: Insect

location Tecton = ft1

maxMoves int = 2

remainingMoves int = 2

sporesEaten int = 0

effectTimer int = 0

state InsectState = NORMAL

### Rovar mozgatása

* **Leírás**

A rovar mozog egy, a tektonjával szomszédos, tektonra

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Rovar tektonja megváltozott-e, eredeti tektonról eltűnt-e, új tektonon rajta van-e. . Rovarnak 1-e a remainingMoves-ja*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]ű

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
CREATE\_TECTON FertileTecton ft2  
ADD\_NEIGHBOUR ft1 ft2  
SET\_BREAKTIMER ft1 5  
SET\_BREAKTIMER ft2 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
CREATE\_MYCELIUM Mycelium m2  
ADD\_MYCELIUM\_TO\_TECTON m2 ft2  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
MOVE i1 ft2  
STATE ft1  
STATE ft2  
STATE m1  
STATE m2  
STATE i1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

}

ft2: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m2

}

occupants List<Insect> = {

i1

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer = 0

deathTimer = -1

m2: Mycelium

growing boolean = false

location Tecton = ft2

growTimer = 0

deathTimer = -1

i1: Insect

location Tecton = ft2

maxMoves int = 2

remainingMoves int = 1

sporesEaten int = 0

effectTimer int = 0

state InsectState = NORMAL

### Rovar sikertelen mozgatása nem-szomszédos tektonra

* **Leírás**

A rovar mozogna egy, a tektonjával nem szomszédos, tektonra, ami nem sikerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Rovar tektonja megmaradt-e a régi, nem változott-e a két tekton. Rovar megtartja-e a 2 remainingMoves-t*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]ű

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
CREATE\_TECTON FertileTecton ft2  
SET\_BREAKTIMER ft1 5  
SET\_BREAKTIMER ft2 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
CREATE\_MYCELIUM Mycelium m2  
ADD\_MYCELIUM\_TO\_TECTON m2 ft2  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
MOVE i1 ft2  
STATE ft1  
STATE ft2  
STATE m1  
STATE m2  
STATE i1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

i1

}

ft2: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m2

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer = 0

deathTimer = -1

m2: Mycelium

growing boolean = false

location Tecton = ft2

growTimer = 0

deathTimer = -1

i1: Insect

location Tecton = ft1

maxMoves int = 2

remainingMoves int = 2

sporesEaten int = 0

effectTimer int = 0

state InsectState = NORMAL

### Rovar sikertelen mozgatása olyan tektonra, amelyen nincs gombafonál

* **Leírás**

A rovar mozog egy, a tektonjával szomszédos, tektonra, de nem sikerül, mert nincs rajta gombafonál

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Rovar tektonja megmaradt-e a régi, nem változott-e a két tekton. Rovar megtartja-e a 2 remainingMoves-t*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]ű

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
CREATE\_TECTON FertileTecton ft2  
ADD\_NEIGHBOUR ft1 ft2  
SET\_BREAKTIMER ft1 5  
SET\_BREAKTIMER ft2 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
MOVE i1 ft2  
STATE ft1  
STATE ft2  
STATE m1  
STATE i1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

i1

}

ft2: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer = 0

deathTimer = -1

i1: Insect

location Tecton = ft1

maxMoves int = 2

remainingMoves int = 2

sporesEaten int = 0

effectTimer int = 0

state InsectState = NORMAL

### Rovar általi spóraevés következtében kettészakadás

* **Leírás**

A rovar megeszik egy SplitSpore-t a tektonján, aminek a hatására kettészakad

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatására létrejön-e a második rovar; mindkettő rovarnak 0 lesz-e a remainingMoves*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
SET\_BREAKTIMER ft1 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
PUT\_SPORE SplitSpore splits1 ft1  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
EAT i1  
STATE ft1  
STATE m1  
STATE i1  
STATE i1-1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

i1

i1-1

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer = 0

deathTimer = -1

i1: Insect

location Tecton = ft1

maxMoves int = 2

remainingMoves int = 0

sporesEaten int = 1

effectTimer int = 0

state InsectState = NORM AL

i1-1: Insect

location Tecton= ft1

maxMoves int = 2

remainingMoves int = 0

sporesEaten int = 0

effectTimer int = 0

state InsectState = NORMAL

### Rovar általi spóraevés következtében Slow állapotba kerülés

* **Leírás**

A rovar megeszik egy SlownessSpore-t a tektonján, aminek a hatására Slow állapotba kerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatása beállitódik-e a rovaron: Slow state, 1 maxMoves, 0 remainingMoves*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
SET\_BREAKTIMER ft1 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
PUT\_SPORE SlownessSpore slows1 ft1  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
EAT i1  
STATE ft1  
STATE m1  
STATE i1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

i1

}

m1: Mycelium

growing boolean = false

location int = ft1

growTimer int = 0

deathTimer int = -1

i1: Insect

location Tecton= ft1

maxMoves int = 1

remainingMoves int = 0

sporesEaten int = 1

effectTimer int = 3

state InsectState = SLOW

### Rovar általi spóraevés következtében Fast állapotba kerülés

* **Leírás**

A rovar megeszik egy SpeedSpore-t a tektonján, aminek a hatására Fast állapotba kerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatása beállitódik-e a rovaron: Fast state, 3 maxMoves, 0 remainingMoves*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
SET\_BREAKTIMER ft1 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
PUT\_SPORE SpeedSpore speeds1 ft1  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
EAT i1  
STATE ft1  
STATE m1  
STATE i1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

i1

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer int = 0

deathTimer int = -1

i1: Insect

location Tecton = ft1

maxMoves int = 3

remainingMoves int = 0

sporesEaten int = 1

effectTimer int = 3

state InsectState = FAST

### Rovar általi spóraevés következtében PreventCut állapotba kerülés

* **Leírás**

A rovar megeszik egy spórát a tektonján, aminek a hatására PreventCut állapotba kerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatása beállitódik-e a rovaron: CannotCut state, 0 remainingMoves*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
SET\_BREAKTIMER ft1 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
PUT\_SPORE PreventCutSpore prevents1 ft1  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
EAT i1  
STATE ft1  
STATE m1  
STATE i1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

i1

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer int = 0

deathTimer int = 0

i1: Insect

location Tecton = ft1

maxMoves int = 2

remainingMoves int = 0

sporesEaten int = 1

effectTimer int = 3

state InsectState = CANNOT\_CUT

### Rovar általi spóraevés következtében Stunned állapotba kerülés

* **Leírás**

A rovar megeszik egy spórát a tektonján, aminek a hatására PreventCut állapotba kerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatása beállitódik-e a rovaron: Stun state, 0 maxMoves, 0 remainingMoves*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
SET\_BREAKTIMER ft1 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
PUT\_SPORE StunSpore stuns1 ft1  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
EAT i1  
STATE ft1  
STATE m1  
STATE i1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

i1

}

m1: Mycelium

growing boolean = false

location Tecton = ft1

growTimer = 0

deathTimer int = -1

i1: Insect

location Tecton = ft1

maxMoves int = 0

remainingMoves int = 0

sporesEaten int = 1

effectTimer int = 1

state InsectState = STUN

### Rovar általi sikertelen spóraevés

* **Leírás**

A rovar megenne egy spórát a tektonján, de nem sikerül neki, mert nincs spóra

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Minden megmarad-e úgy, ahogy volt*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
SET\_BREAKTIMER ft1 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
EAT i1  
STATE ft1  
STATE m1  
STATE i1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

m1

}

occupants List<Insect> = {

i1

}

m1: Mycelium

growing boolea = false

location Tecton= ft1

growTimer int = 0

deathTimer int = -1

i1: Insect

location Tecton = ft1

maxMoves int = 2

remainingMoves int = 2

sporesEaten int = 0

effectTimer int = 0

state InsectState = NORMAL

### Rovar általi gombafonál elvágás

* **Leírás**

A rovar elvág egy fonalat a tektonján

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Rovar tektonja megváltozott-e 2 kör eltentével, eltünt-e a fonal a tektonról, elmenekült-e a rovar*

* **Bemenet**

[a proto bemeneti nyelvén megadva (lásd előző anyag)]ű

ADD\_PLAYER Mycologist player1  
CREATE\_TECTON FertileTecton ft1  
CREATE\_TECTON FertileTecton ft2  
ADD\_NEIGHBOUR ft1 ft2  
SET\_BREAKTIMER ft1 5  
SET\_BREAKTIMER ft2 5  
CREATE\_MYCELIUM Mycelium m1  
ADD\_MYCELIUM\_TO\_TECTON m1 ft1  
CREATE\_MYCELIUM Mycelium m2  
ADD\_MYCELIUM\_TO\_TECTON m2 ft2  
CREATE\_MUSHROOMBODY mb1 ft2  
ADD\_PLAYER Entomologist player2  
CREATE\_INSECT ft1 i1  
START\_GAME  
ENDTURN  
CUT i1  
ENDTURN  
ENDTURN  
ENDTURN  
STATE ft1  
STATE ft2  
STATE mb1  
STATE m1  
STATE i1

* **Elvárt kimenet**

[a proto kimeneti nyelvén megadva (lásd előző anyag)]

ft1: FertileTecton

breakTimer int = 3

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

}

ft2: FertileTecton

breakTimer int = 3

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

m2

}

occupants List<Insect> = {

i1

}

mb1: MushroomBody

remainingEjects int = 3

location Tecton = ft2

mushroomSpores List<Spore> = {

mb1-speeds1

mb1-speeds2

mb1-speeds3

}

m1: Mycelium

growing boolean = false

location Tecton = null

growTimer int = 0

deathTimer int = 0

i1: Insect

location Tecton = ft2

maxMoves int = 2

remainingMoves int = 2

sporesEaten int = 0

effectTimer int = 0

state InsectState = NORMAL

## 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 | Bencze | Tevékenység:  Parancsok és tesztesetek átemelése, illetve tesztesetek be- és kimenetének leírása |
| 2025.04.06 22:00 | 30 perc | Bencze | Tevékenység:  Gyűlésen elhangzottak alapján, egységesítése céljából, írási hibák javítása |
| 2025.04.10 15:00 | 1 óra | Bencze | Tevékenység:  Insect osztály leírása, tesztesetek kiegészítése játékossal |
| 2025.04.10 21:00 | 30 perc | Bencze | Tevékenység:  További javítások |
| 2025.04.12 23:00 | 15 perc | Bencze | Tevékenység:  Megbeszélésen elhangzottak feljegyzése és elkezdése |
| 2025.04.13 13:00 | 3 óra | Bencze | Tevékenység:  Megbeszélésen elhangzottak folytatása, pszeudokód írás, meglévő dolgok javítása |
| 2025.04.13 21:00 | 3 óra | Bencze | Tevékenység:  Dokumentum összevágása, átnézése és véglegesítése |