

## 8. Részletes tervek

23 – totoro

Konzulens:

Haragos Gergő Viktor

### Csapattagok

Bohus János  
Ducz Ákos  
Galambos Csaba  
Horváth Avarka  
Végvári Péter

XYOVFZ	janos.bohus1@gmail.com
GC1RTE	akos.ducz@gmail.com
BVBQKN	galambos.csaba05@gmail.com
OODMXV	avarkahorvath@gmail.com
AW3JH1	vegvaripeter21@gmail.com

## 8. Részletes tervek

### 8.1 Változtatások.

- JSON helyet, java binárisba fog történni a serializáció.
- Identifiable osztály, amiből mindenki leszarmazik.

### 8.2 Osztályok és metódusok tervei.

#### 8.2.1 Identifiable

- **Felelősség**

Egyedi azonosítót biztosító absztrakt osztály.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **private final int id:** egyedi azonosító.

- **Metódusok**

- **public int getId():** visszaadja az azonosítót.

#### 8.2.2 Player

- **Felelősség**

Absztrakt osztály, ebből származnak le a játékosok

- **Ősosztályok**

Identifiable

- **Interfészek**

- **Attribútumok**

- **private int score:** A játékos pontszáma

- **Metódusok**

- **public void addScore(int i):** hozzáad i-t a játékos pontszámához.
- **public int getScore():** visszadja a játékos pontszámát
- **public void endTurn():** játékos befejezi a körét

### 8.2.3 Effect

- **Felelősség**

A rovarokra érvényes hatások öse. Minden spóra egy effectet tárol, ha a rovar megeszi akkor lesz érvényes rá.

- **Ősosztályok**

Identifiable

- **Interfészek**

- **Attribútumok**

- **-int duration:** Tárolja, hogy hány körig van még érvényben az effect.

- **Metódusok**

- **+ abstract apply(insect: Insect):void** Ez a függvény fogja az effectet alkalmazni a rovaron.
- **+abstract remove(insect: Insect):void** Ez a függvény törli az effectet a rovarról.
- **+decreaseDuration():void** : Csökkenti a durationt.

### 8.2.4 ParalyzeEffect

- **Felelősség**

A ParalyzeEffect osztály célja, hogy egy rovar ideiglenesen megbénítson. A hatás alatt álló rovar nem tud mozogni, spórát enni vagy gombafonalat vágni.

- **Ősosztályok**

Identifiable → Effect → ParalyzeEffect

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **Metódusok**

- **+apply(insect: Insect): void:** A hatás alkalmazása a célzott rovarra. Beállítja, hogy a rovar bénult állapotba kerüljön (ne tudjon aktív műveleteket végrehajtani).
- **+remove(insect: Insect): void:** Törli a bénított hatást az adott rovarról.

### 8.2.5 FastEffect

- **Felelősség**

A FastEffect osztály célja, hogy egy rovar ideiglenesen felgyorsítson. A hatás alatt álló rovar többet tud mozogni.

- **Ősosztályok**

Identifiable → Effect → FastEffect

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **Metódusok**

- **+apply(insect: Insect): void:** A hatás alkalmazása a célzott rovarra. Beállítja, hogy a rovar gyorsult állapotba kerüljön
- **+remove(insect: Insect): void:** A rovar sebességét visszaállítja alaphelyzetbe, törli az effectet.

### 8.2.6 ClawParalyzeEffect

- **Felelősség**

A ClawParalyzeEffect osztály célja, hogy egy rovar csáprágóját ideiglenesen megbénítsa. A hatás alatt álló rovar nem tud, spórát enni, gombafonalat vágni.

- **Ősosztályok**

Identifiable → Effect → ClawParalyzeEffect

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **Metódusok**

- **+apply(insect: Insect): void:** A hatás alkalmazása a célzott rovarra. Beállítja, hogy a rovar csáprágója lebénuljon.
- **+remove(insect: Insect): void:** Eltörli az effectet a rovarról.

### 8.2.7 SplitEffect

- **Felelősség**

A SplitEffect osztály célja, hogy egy rovar csáprágóját ideiglenesen megbénítsa. A hatás alatt álló rovar nem tud, spórákat enni, gombafonalat vágni.

- **Össztályok**

Identifiable → Effect → SplitEffect

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **Metódusok**

- **+apply(insect: Insect): void:** A hatás alkalmazása a célzott rovarra. Létrejön egy új rovar.
- **+remove(insect: Insect): void:** Eltörli az effectet a rovarról.

### 8.2.8 SlowEffect

- **Felelősség**

A SlowEffect osztály célja, hogy egy rovar ideiglenesen lelassítsa. A hatás alatt álló rovar kevesebbet tud mozogni.

- **Össztályok**

Identifiable → Effect → SlowEffect

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **Metódusok**

- **+apply(insect: Insect): void:** A hatás alkalmazása a célzott rovarra. Fele annyit fog tudni a rovar mozogni.
- **+remove(insect: Insect): void:** Eltörli az effectet a rovarról.

### 8.2.9 IStem

- **Felelősség**

Gombatestek interfésze. Ez felel azért, hogy legyen add és remove függvénye azoknak az osztályoknak akik használják a gombatesteket.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **+add(mushroomStem: MushroomStem): boolean:** Új gombatest növesztése ezzel lehetséges.
- **+remove(mushroomStem: MushroomStem): boolean:** Gombatest törlése ezzel lehetséges, ha meghal

### 8.2.10 Mushroomer

- **Felelősség**

Gombász játékos osztálya. Ez az osztály tárolja a gombákat, irányít mindent amit a gombász tud.

- **Ősosztályok**

Identifiable → Player

- **Interfészek**

IStem, ISpore, IThread

- **Attribútumok**

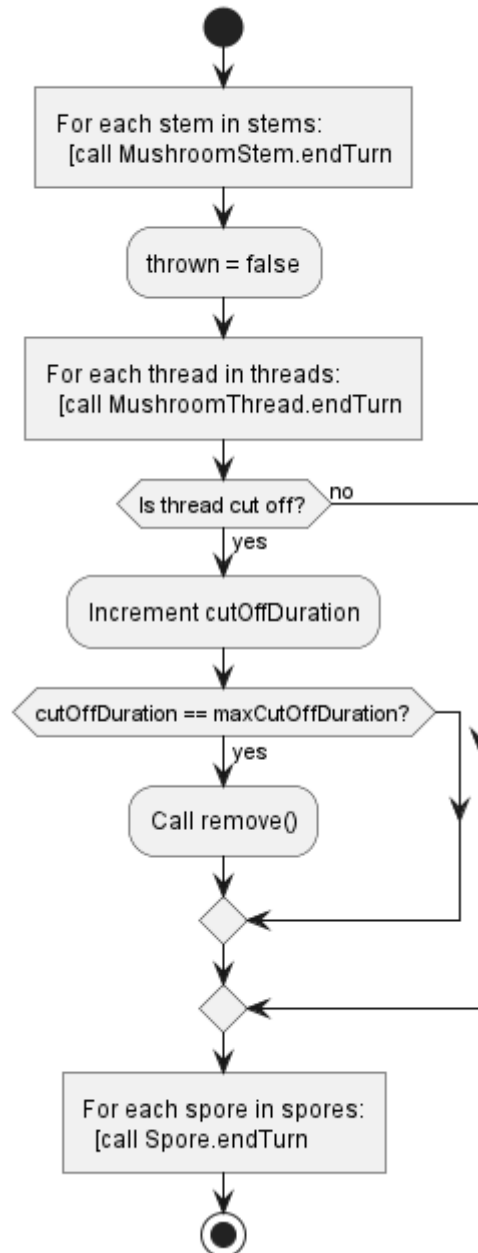
- **-spores: List<Spore>:** Tárolja a spórákat amiket lerakott a gombász
- **-stems: List<MushroomStem>:** Gombász gombatestjei
- **-threads: List<MushroomThread>:** Gombász gombafonaljai

- **Metódusok**

- **add(MushroomThread): boolean:** Hozzáad egy MushroomThreadet a listához.
- **add(Spore): boolean:** Hozzáad egy Spórát a listához.
- **add(MushroomStem): boolean:** Hozzáad egy gombatestet a listához.
- **eatWith(MushroomThread, Insect): boolean:** Kiválasztja melyik fonallal eszi meg melyik rovar.

- **endTurn(): void:** Kör vége, ekkor hajtódik végre a spórák, fonalak, gombák

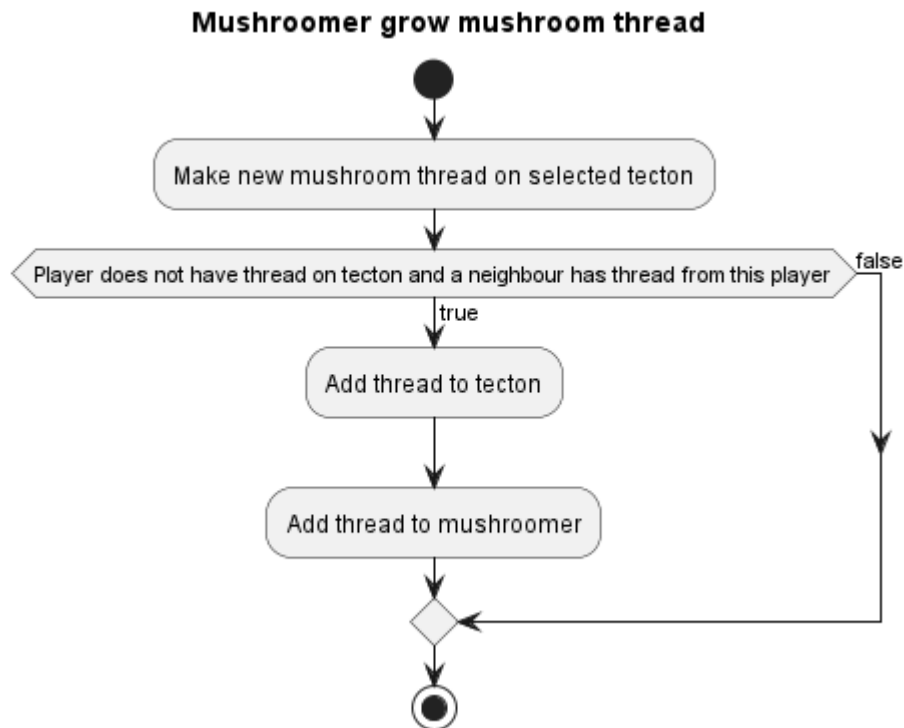
### Mushroomer End Turn Sequence



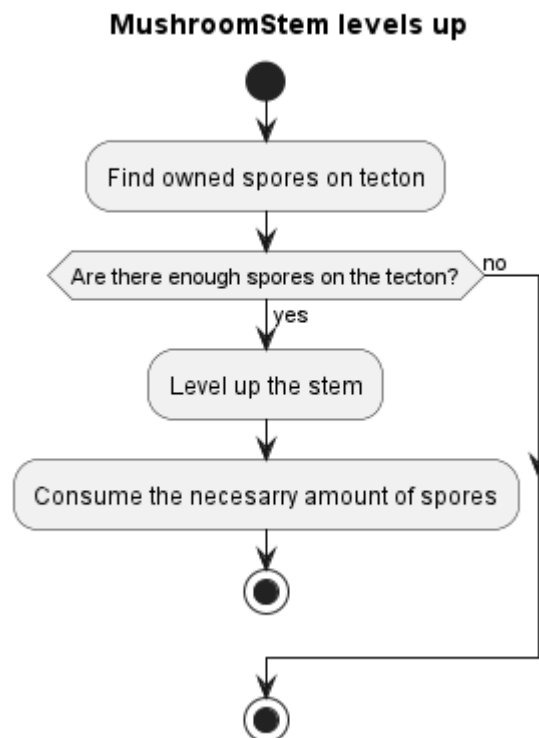
tulajdonságai.

- **getSpores(Tecton): List<Spore>:** Getter függvény a spórákhoz.

- **growMushroomthread(Tecton): boolean:** Növeszt egy fonalat, ellenőrzi, hogy lehet-e oda növeszteni.

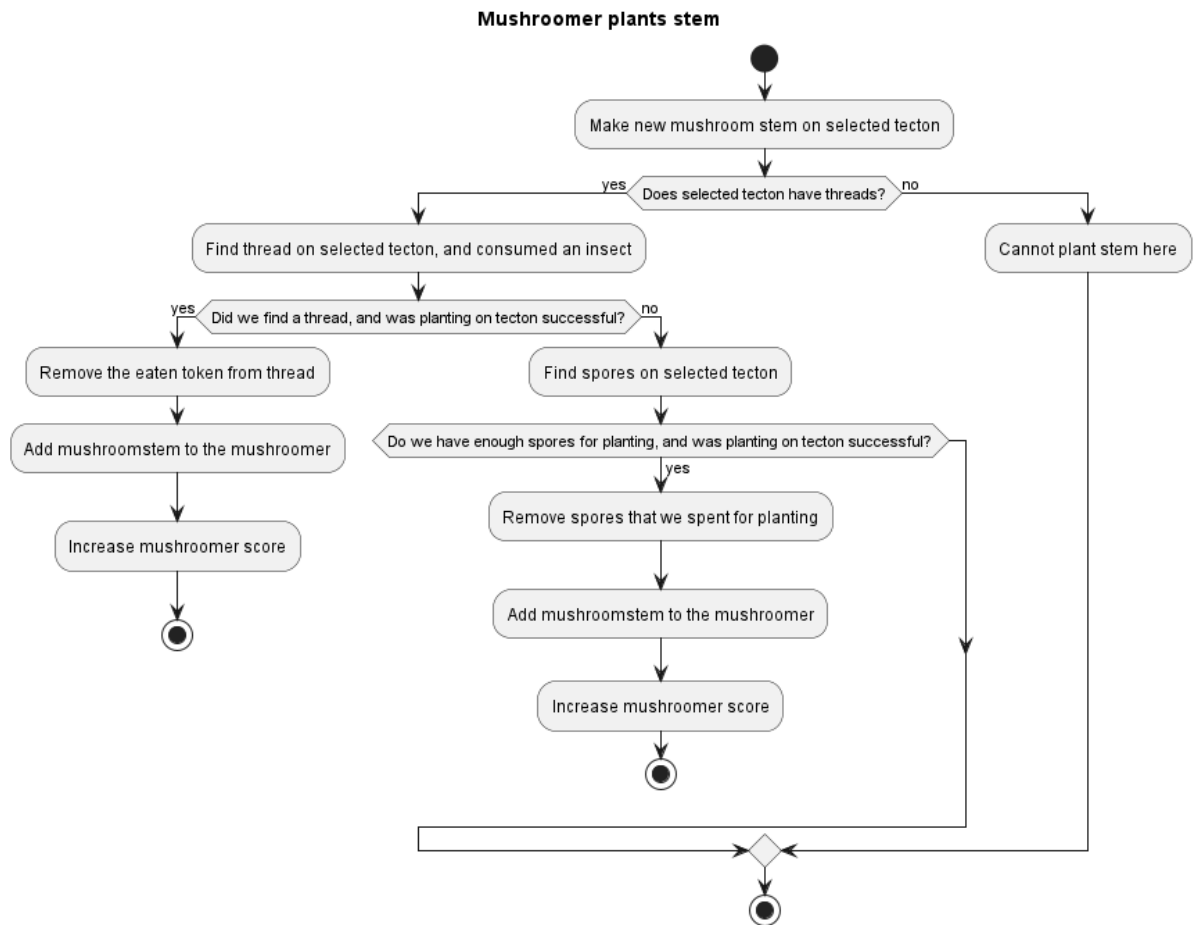


- **hasThread(Tecton): boolean:** Megnézi hogy van e fonal az adott tectonon.
- **levelUp(MushroomStem): Boolean:** Szintet kép a gomba.



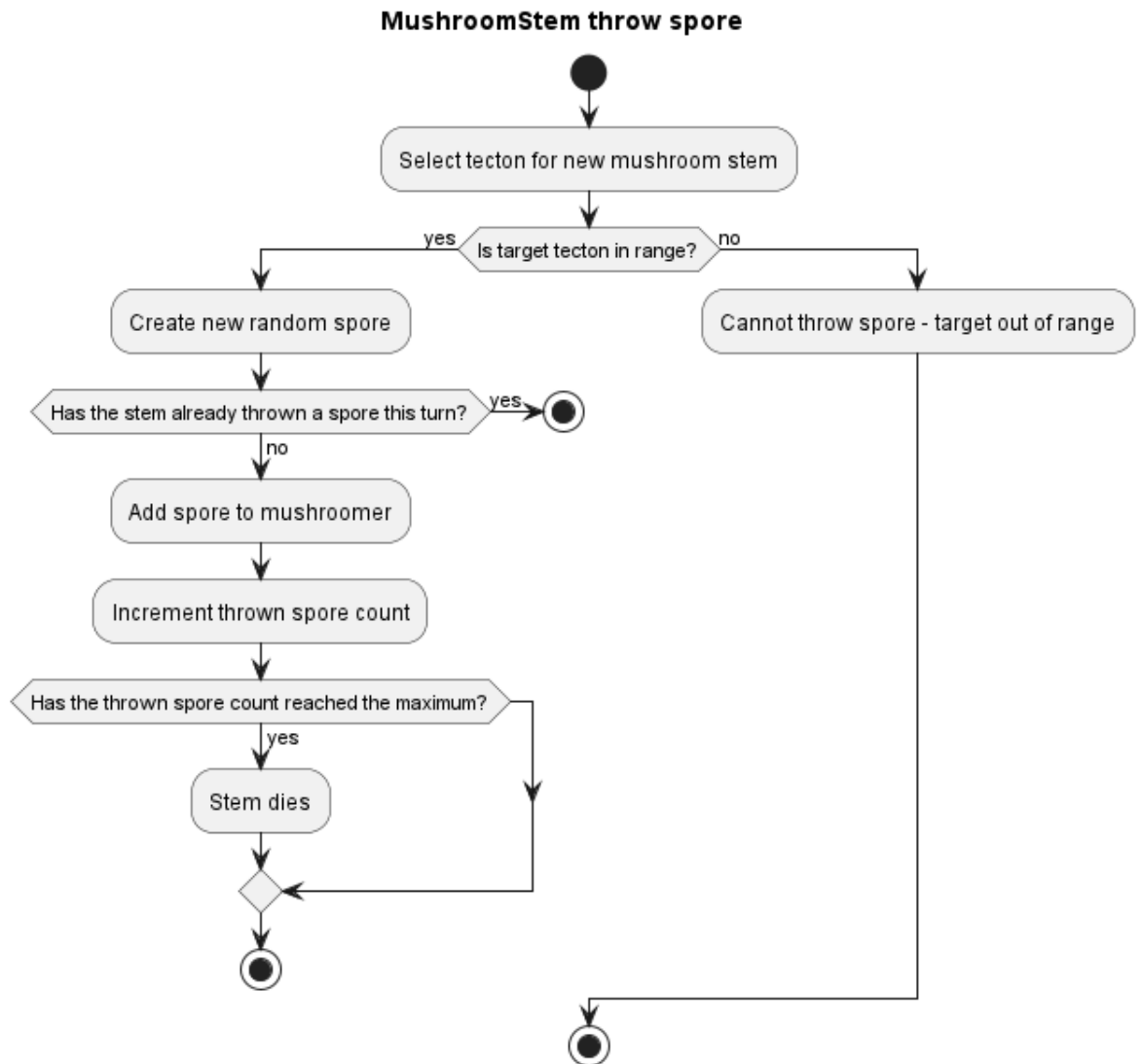


- **plantMushroomstem(Tecton): Boolean:** Növeszt egy gombatörzset, ha lehetséges.



- **remove(Spore): boolean:** Kitörli a spórát a listából.
- **remove(MushroomStem): boolean:** Kitörli a gombatestet a listából.
- **remove(MushroomThread): boolean:** Kitörli a gombafonalat a listából.

- **throwSpore(MushroomStem, Tecton): Boolean:** Dob egy általunk kiválasztott spórát az adott tectonra.



### 8.2.11 ISpore

- **Felelősség**

Spórák interfésze. Ez felel azért, hogy legyen add és remove függvénye azoknak az osztályoknak akik használják a spórákat.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **+add(spore: Spore):boolean:** Ezzel a függvénnyel lehet egy spórát lerakni, ez egy interfész függvénye, nincs megvalósítva.
- **+remove(spore: Spore):boolean:** Ezzel lehet kitörölni egy spórát, ez egy interfész függvénye, nincs megvalósítva.

### 8.2.12 Spore

- **Felelősség**

Ez a spórák ősszálya, ezt az osztályt tárolják. Absztrakt

- **Ősosztályok**

Identifiable → Entity

- **Interfészek**

- **Attribútumok**

- **private nutrition:** tápérték, amit megkap a rovarász ha az egyik rovára megeszi.
- **private effect:** hatás, amit ki fog fejteni a rovarra, ami megeszi.

- **Metódusok**

- **public Effect getEffect():** Visszaadja a hatását.
- **public int getNutrition():** Visszaadja a tápértékét.

### 8.2.13 ClawParalyzingSpore

- **Felelősség**

Olyan spóra, ami hogy ha megeszik, lebénítja a rovar csáprágóját.

- **Ősosztályok**

Identifiable → Entity → Spore

- **Interfészek**

- **Attribútumok**

- **private nutrition:** tápérték, amit megkap a rovarász ha az egyik rovara megeszi.
- **private effect:** hatás, amit ki fog fejteni a rovarra, ami megeszi. ClawParalyzeEffect.

- **Metódusok**

- **public Effect getEffect():** Visszaadja a hatását.
- **public int getNutrition():** Visszaadja a tápértékét. ClawParalyzeEffect.

### 8.2.14 ParalyzingSpore

- **Felelősség**

Olyan spóra, ami hogy ha megeszik, lebénítja a rovar.

- **Ősosztályok**

Identifiable → Entity → Spore

- **Interfészek**

- **Attribútumok**

- **private nutrition:** tápérték, amit megkap a rovarász ha az egyik rovara megeszi.
- **private effect:** hatás, amit ki fog fejteni a rovarra, ami megeszi. ParalyzeEffect.

- **Metódusok**

- **public Effect getEffect():** Visszaadja a hatását.
- **public int getNutrition():** Visszaadja a tápértékét. ParalyzeEffect.

### 8.2.15 SlowingSpore

- **Felelősség**

Olyan spóra, ami hogy ha megeszik, lelassítja a rovar.

- **Ősosztályok**

Identifiable → Entity → Spore

- **Interfészek**

- **Attribútumok**

- **private nutrition:** tápérték, amit megkap a rovarász ha az egyik rovára megeszi.
- **private effect:** hatás, amit ki fog fejteni a rovarra, ami megeszi. SlowingEffect.

- **Metódusok**

- **public Effect getEffect():** Visszaadja a hatását. SlowingEffect.
- **public int getNutrition():** Visszaadja a tápértékét.

### 8.2.16 SpeedingSpore

- **Felelősség**

Olyan spóra, ami ha megeszik, lelassítja a rovar.

- **Ősosztályok**

Identifiable → Entity → Spore

- **Interfészek**

- **Attribútumok**

- **private nutrition:** tápérték, amit megkap a rovarász ha az egyik rovára megeszi.
- **private effect:** hatás, amit ki fog fejteni a rovarra, ami megeszi. FastEffect.

- **Metódusok**

- **public Effect getEffect():** Visszaadja a hatását. FastEffect.
- **public int getNutrition():** Visszaadja a tápértékét.

### 8.2.17 SplittingSpore

- **Felelősség**

Olyan spóra, amit ha megesznek, osztódásra kényszeríti a rovar.

- **Ősosztályok**

Identifiable → Entity → Spore

- **Interfészek**

- **Attribútumok**

- **private nutrition:** tápérték, amit megkap a rovarász ha az egyik rovára megeszi.
- **private effect:** hatás, amit ki fog fejteni a rovarra, ami megeszi. FastEffect.

- **Metódusok**

- **public Effect getEffect():** Visszaadja a hatását. SplitEffect.
- **public int getNutrition():** Visszaadja a tápértékét.

### 8.2.18 Entity

- **Felelősség**

Absztrakt osztály. Minden olyan objektum ebből származik le, amit birtokol egy játékos.

- **Ősosztályok**

Identifiable

- **Interfészek**

- **Attribútumok**

- **Player owner:** Az a játékos, aki birtokolja az objektumot
- **Tecton tecton:** Az entitást tartalmazó tecton.

- **Metódusok**

- **public abstract void remove()**
- **public Player getOwner():** Visszaadja az objektumot birtokló játékost.
- **public Tecton getLocation():** Visszaadja a tectont, amin az entitás van.
- **public void setLocation(Tecton location):** Beállítja az entitás tartalmazó tectont
- **public void endTurn()**

### 8.2.19 Insect

- **Felelősség**

Rovar osztály, egy rovar reprezentál.

- **Ősosztályok**

Identifiable □ Entity

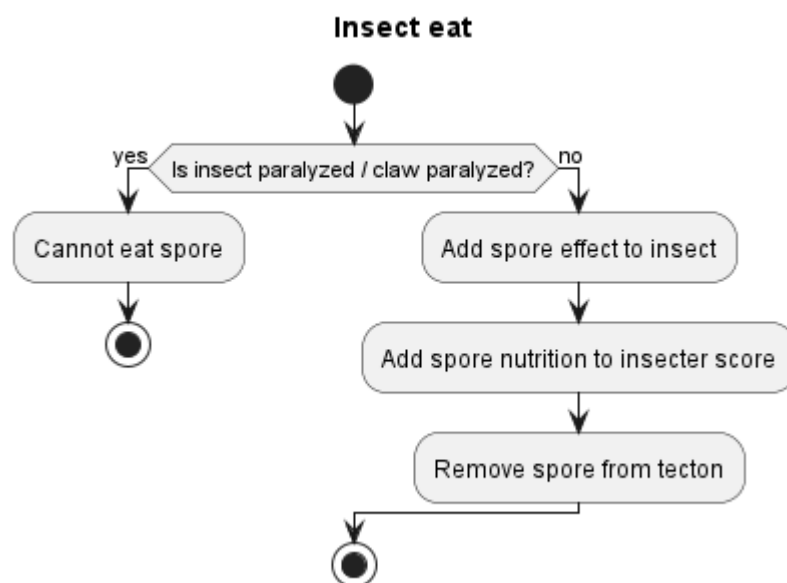
- **Interfészek**

- **Attribútumok**

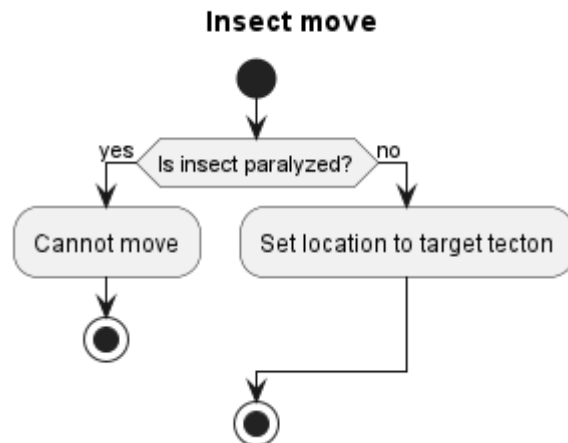
- **protected List<Effect> effects:** ilyen hatások vannak a rovaron.
- **protected boolean paralyzed:** le van-e bénulva a rovar.
- **protected boolean clawParalyzed:** le van-e bénulva a csáprágója.
- **protected int baseSpeed:** rovar alap sebessége.
- **protected double speedModifier:** rovar jelenlegi sebessége.

- **Metódusok**

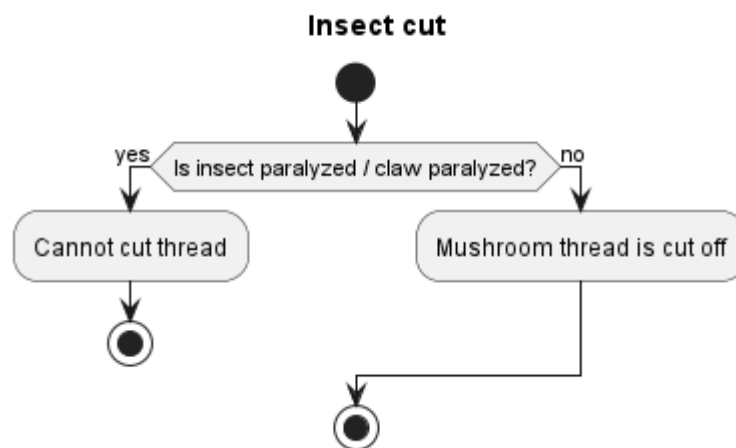
- **public boolean isParalyzed():** megállapítja, hogy a rovar bénult állapotban van-e.
- **public void setParalyzed(boolean paralyzed):** beállítja a rovar bénulási állapotát.
- **public boolean isClawParalyzed():** megállapítja, hogy a rovar csáprágója képes-e fonalat vágni.
- **public void setClawParalyzed(boolean clawParalyzed):** beállítja a karom bénulás állapotát.
- **public int getBaseSpeed():** visszaadja a rovar módosítók nélküli mozgási sebességét.
- **public double getSpeedModifier():** lekéri a sebességmódosítót.
- **public void setSpeedModifier(double speedModifier):** beállítja a sebességmódosítót.
- **public void add(Effect e):** új hatást ad a rovarhoz.
- **public void remove(Effect e):** eltávolít egy hatást a rovarról.
- **public boolean eat(Spore sp):** megpróbál elfogyasztani egy gombaspórát.



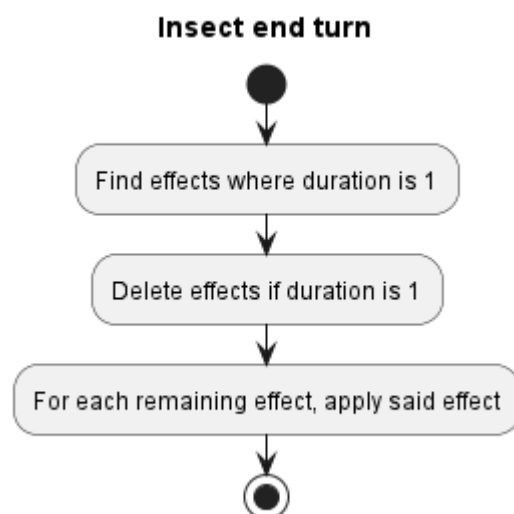
- **public boolean move(Tecton targetTecton):** megpróbál mozogni.



- **public boolean cut(MushroomThread th):** megpróbál elvágni egy gombafonalat.

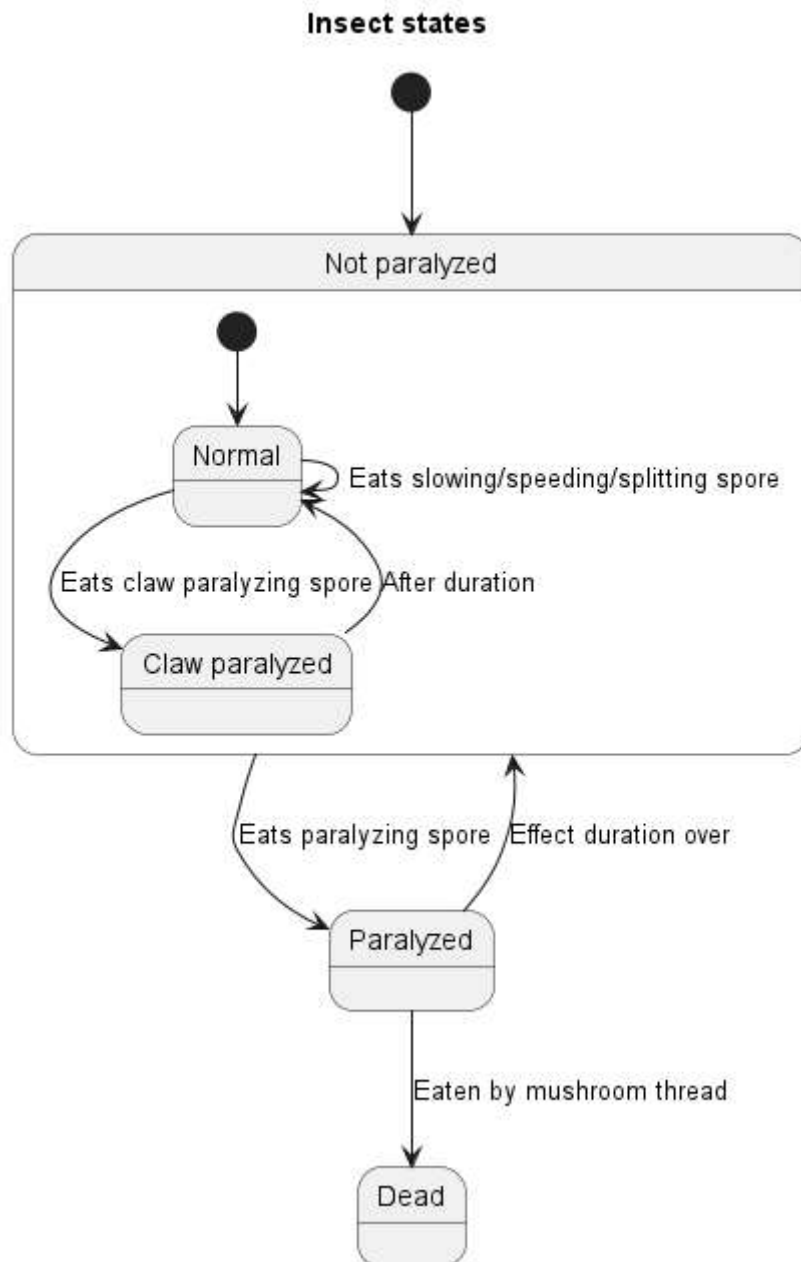


- **public void split():** rovar kettéosztja.
- **public void setLocation(Tecton location):** beállítja a rovar helyét
- **public void endTurn():** végrehajtja a kör végén szükséges folyamatokat. Törli a lejárt hatásokat, ezután újra kiértékeli az összesített hatást.



- **public void remove():** eltávolítja a rovar.





### 8.2.20 Insecter

- **Felelősség**

Rovarász osztály megvalósítása.

- **Ősosztályok**

Identifiable -> Player

- **Interfészek**

Insect

- **Attribútumok**

- **private List<Insect> insect:** rovarász által irányított rovarok.

- **Metódusok**

- **public boolean eatWith(Insect insect, Spore sp):** megesz egy spórát az adott rovarral.
- **public boolean moveWith(Insect insect, Tecton t):** egy tectonra lép az adott rovarral.
- **public boolean cutWith(Insect insect, MushroomThread th):** gombafonalat vág el az adott rovarral.
- **public void endTurn():** befejezi a körét.
- **public boolean add(Insect insect):** hozzáadja a rovar a rovarjai közé.
- **public boolean remove(Insect insect):** kiveszi a rovar a rovarjaiból.
- **public List<Insect> getInsects():** visszaadja a rovarait.

### 8.2.21 MushroomThread

- **Felelősség**

A gombászok gombafonala, tektonokat köt össze, nő, esetleg rovart eszik

- **Ősosztályok**

Identifiable -> Entity

- **Interfészek**

- **Attribútumok**

- **-bool eaten:** Tárolja, a fonal evett-e már rovar.
- **-bool connected:** Tárolja, hogy a fonal kapcsolódik-e gombatesthez.
- **-bool cutoff:** Tárolja, hogy a fonalat elvágták-e.

- **Metódusok**

- **public bool eat(insect: Insect):** Megeszik egy rovar, és beállítja az eaten változót.
- **public void remove(insect: Insect):** Eltávolítja a fonalat a játékból.
- **public void endTurn():** Csökkenti a fonal hátralévő idejét, ha el van vágva, vagy nem connected.
- **public bool hasEaten():** Visszaadja az eaten változó értékét.
- **public void setEaten(bool: Boolean):** Beállítja az eaten változó értékét.

- **public void setCutoff(bool: Boolean):** Beállítja a cutoff változó értékét.
- **public bool isConnected():** Visszaadja az eaten változó értékét.

*isConnected():*

*indulóTecton = this.location*

*if indulóTecton.hasMushroomSteam():*

*return true*

*látogatott = üres halmaz*

*sor = új üres sor (pl. Queue)*

*sor.push(indulóTecton)*

*látogatott.add(indulóTecton)*

*amíg sor nem üres:*

*aktuális = sor.pop()*

*ha aktuális.hasMushroomSteam():*

*return true*

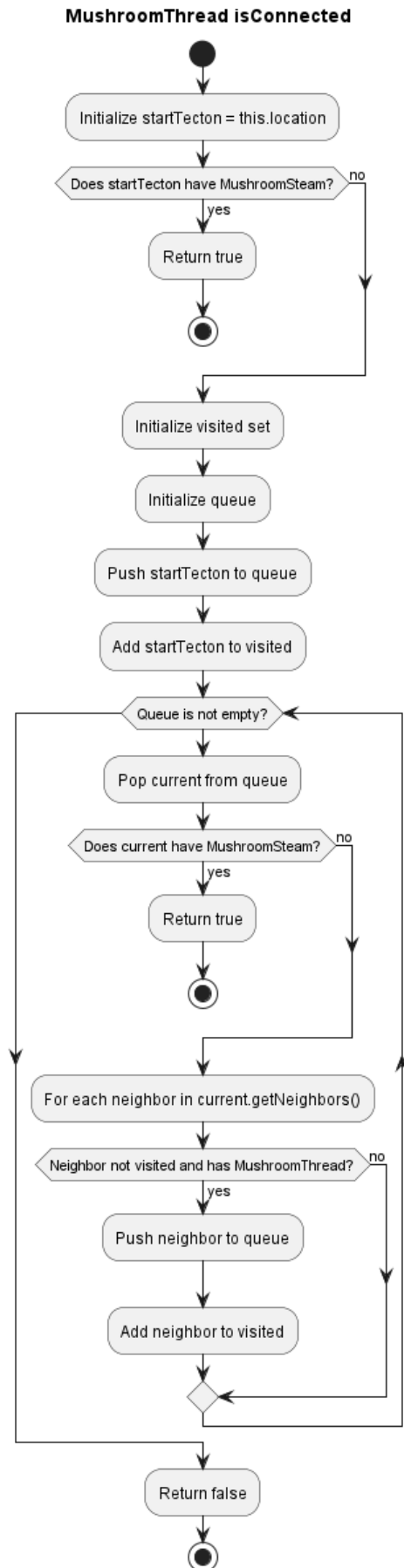
*minden szomszéd in aktuális.getNeighbors():*

*if szomszéd nincs a látogatott-ban ÉS szomszéd.hasMushroomThread():*

*sor.push(szomszéd)*

*látogatott.add(szomszéd)*

*return false*



### 8.2.22 MushroomStem

- **Felelősség**

A gombászok gombateste, spórákat lő, pontot ér és a fonalaknak szüksége van rá, hogy életben maradjon

- **Ősosztályok**

Identifiable -> Entity

- **Interfészek**

- **Attribútumok**

- **int cost:** Tárolja a gombatest növesztés költségét.
- **int maxSporeThrows:** Tárolja, hogy hány spóradobás után hal meg a gombatest.
- **int numThrownSpores:** Tárolja, hány spórát dobott a gombatest.
- **int level:** Tárolja, hanyas szintű a gombatest

- **Metódusok**

- **public bool levelUp():** Növeli a level változó értékét.
- **public void remove(insect: Insect):** Eltávolítja a fonalat a játékból.
- **public void endTurn():** Csökkenti a fonal hátralévő idejét, ha el van vágva, vagy nem connected.
- **public bool throwSpore(tecton: Tecton):** Spórát dob a megadott tektonra.
- **public int getCost():** Visszaadja a cost változó értékét.
- **public int getMaxSporeThrows():** Visszaadja a maxSporeThrows változó értékét.
- **public int getNumThrownSpores():** Visszaadja a numThrownSpores változó értékét.

### 8.2.23 Tecton

- **Felelősség**

A pálya építőelemei, a gombák és rovarok a tektonokon vannak, élnek, mozognak. . .

- **Ősosztályok**

Identifiable

- **Interfészek**

IRound, IThread, ISpore, IStem

- **Attribútumok**

- **List<spore> spores:** Tárolja a tektonon levő spórákat.
- **MushroomStem stem:** Tárolja a tektonon levő gombatestet.
- **List<tecton> neighbours:** Tárolja a tektonnal szomszédos tektonokat.
- **List<MushroomThread> threads:** Tárolja a tektonon levő gombafonalakat.
- **List<Insect> insects:** Tárolja a tektonon levő rovarokat.

- **Metódusok**

- **public List<spore> getSpores():** Visszaadja a tektonon levő spórák listáját.
- **public void remove(insect: Insect):** Eltávolítja a fonalat a játékból.
- **public void endTurn():** Csökkenti a fonal hátralévő idejét, ha el van vágva, vagy nem connected.
- **public void addNeighbour(tecton: Tecton):** Hozzáad egy szomszédot a szomszédokhoz.
- **public tecton split():** Kettétöri a tektont, és visszaadja az újat.

*split():*

```
newTecton = new Tecton()
```

```
randomNumber = random between 0 and 3
```

```
if randomNumber == 0:
```

```
    newTecton = new SingleThreadedTecton()
```

```
else if randomNumber == 1:
```

```
    newTecton = new StemlessTecton()
```

```
else if randomNumber == 2:
```

```
    newTecton = new ThreadConsumingTecton()
```

```
for each neighbor in currentTecton.neighbors:
```

```
    if random true or false:
```

```
        newTecton.addNeighbour(neighbor)
```

```
currentTecton.threads.clear()
```

```
sporesToMove = empty list
```

```
for each spore in currentTecton.spores:
```

```
    if random true or false:
```

```
        sporesToMove.add(spore)
```

```
for each spore in sporesToMove:
```

```
    spore.setLocation(newTecton)
```

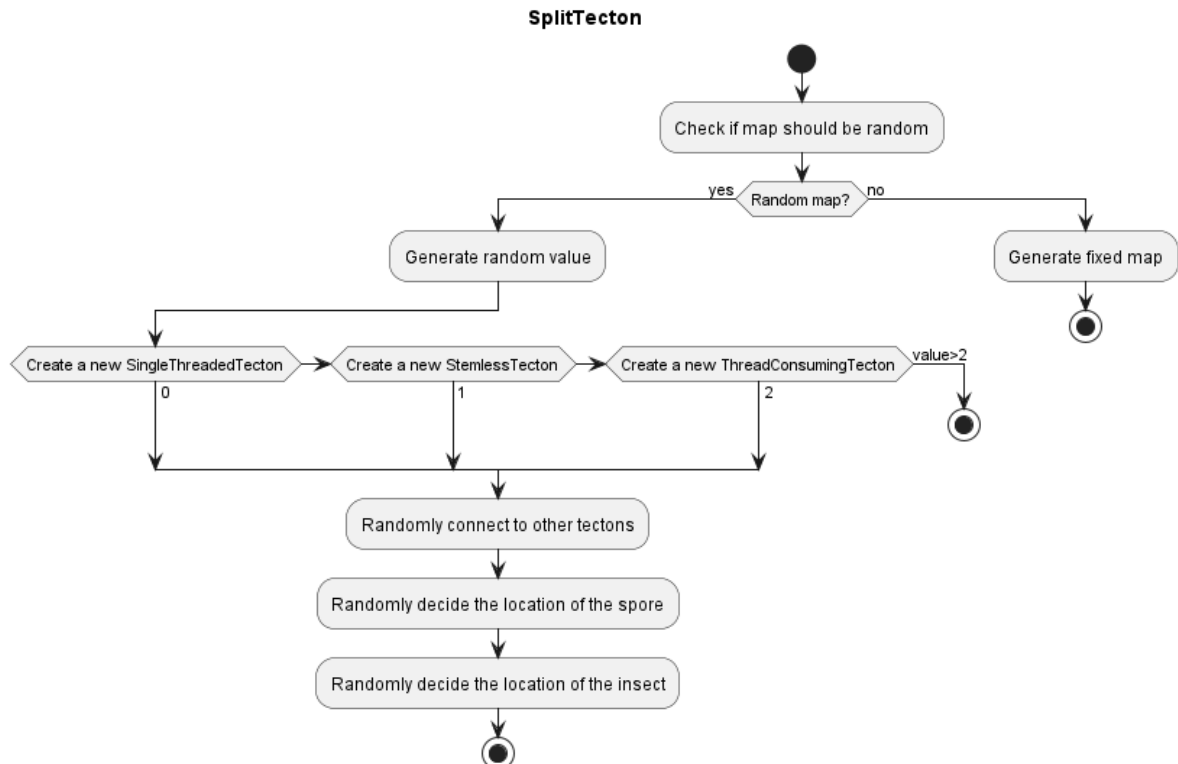
```

insectsToMove = empty list
for each insect in currentTecton.insects:
    if random true or false:
        insectsToMove.add(insect)

for each insect in insectsToMove:
    insect.setLocation(newTecton)

return newTecton

```



- **public bool add(e: Entity):** Hozzáadja az entitást a megfelelő helyre.
- **public bool remove(e: Entity):** Eltávolítja az entitást a tektonról.
- **public List<Entity> get():** visszadja a megfelelő változó értékét.
- **public void removeUnconnectedThreads():** Csökkenti a gombatesthez nem kapcsolódó gombafonalak hátrlévő idejét.

### 8.2.24 LifeSupportTecton

- **Felelősség**

Olyan tekton, ami életben tartja a rajta lévő fonalakat, ha azok nem csatlakoznak gombatesthez.

- **Ősosztályok**

Identifiable -> Tecton

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **public void removeUnconnectedThreads():** Nem csinál semmit.

### 8.2.25 SingleThreadedTecton

- **Felelősség**

Olyan tekton, amin legfeljebb 1 gombafonal lehet.

- **Ősosztályok**

Identifiable -> Tecton

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **public bool add(msThread: MushroomThread):** Rárakja a fonalat a tektonra, és figyel rá, hogy legfeljebb 1 lehessen.

### 8.2.26 StemlessTecton

- **Felelősség**

Olyan tekton, amin nem nőhet gombatest.

- **Ősosztályok**

Identifiable -> Tecton

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **public bool add(stem: MushroomStem):** Hamissal visszatér



### 8.2.27 ThreadConsumingTecton

- **Felelősség**

Olyan tekton, amin folyamatosan halnak meg a gombafonalak.

- **Ősosztályok**

Identifiable -> Tecton

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **public void endRound():** Minden fonal életét csökkenti.

### 8.2.28 Map

- **Felelősség**

Maga a pálya, tárolja a tektonokat.

- **Ősosztályok**

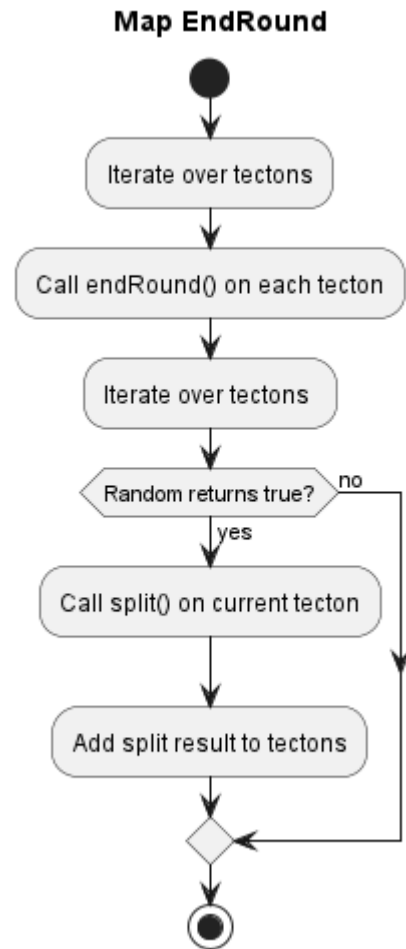
- **Interfészek**

IRound

- **Attribútumok**

- **List<Tecton> tectons:** Tárolja a pályát alkotó tektonokat.

- **Metódusok**



- **public void endRound():** Minden kör végén lefut.
- **public void generate():** Generál egy pályát.
- **public void connect(t1: Tecton, t2: Tecton):** Összeköti a két tectont (Beállítja őket egymás szomszédejának).

## 8.3 A tesztek részletes tervei, leírásuk a teszt nyelvén

### 8.3.1 Teszt: rovarok mozgása

- **Leírás**  
Rovar mozgásának tesztelése bizonyos körülmények között
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Azt teszteljük, hogy a rovar mozgatása helyesen működik-e például effektek hatása alatt is.
- **Bemenet**

*/rand -disable*

*# tesztpálya betöltése*

*# több insect van a pályán, egyesek effektek hatásai alatt*

*# mindegyik a 0-ás tectonon van alaphól, innen az 1-esre tudnak mozogni, a 4-es pedig túl távoli*

*/load testmap.map*

*/start*

*# sikeres mozgás*

*/manualtrigger !move 0 1*

*# sikertelen mozgás (elérhetetlen tecton)*

*/manualtrigger !move 0 4*

*# mozgás paralyze effekt alatt (1-es insect)*

*/manualtrigger !move 1 1*

*# mozgás slow effekt alatt (2-es insect)*

*/manualtrigger !move 2 1*

*# mozgás speed effekt alatt (3-es insect)*

*/manualtrigger !move 3 4*

- **Elvárt kimenet**

*Insect(0): location: Tecton(0) -> Tecton(1)*

*Insect(0): location: Tecton(1) -> Tecton(1)*

*Insect(1): location: Tecton(0) -> Tecton(0)*

*Insect(2): location: Tecton(0) -> Tecton(0)*

*Insect(3): location: Tecton(0) -> Tecton(4)*

### 8.3.2 Teszt: fonal elvágása

- **Leírás**  
A fonalak rovarok általi elvágását teszteljük
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Azt teszteljük, hogy a fonalvágás spórák hatása alatt is helyesen viselkedik-e, illetve hogy az elszeparált fonalak tényleg megszűnnek-e néhány körön belül.
- **Bemenet**

```
/rand -disable
# tesztpálya betöltése
# több insect van a pályán, egyesek effektek hatásai alatt
# mindegyik a 0-ás tectonon van alapból, innen az 1-esre tudnak mozogni, a 4-es pedig túl
távoli
/load testmap.map
/start
```

```
# sikertelen vágás, mivel ez az insect spóra hatása miatt nem vághat
/manualtrigger !cut 4 0
```

```
# sikeres vágás
/manualtrigger !cut 0 0
```

```
!endturn
!endturn
!endturn
!endturn
```

```
...
# néhány körön belül meg kell szűnnie a fonálnak.
!endturn
```

```
....
# néhány újabb körön belül az 1-es ID-vel rendelkező elszeparált thread is megszűnik.
```

- **Elvárt kimenet**

```
Thread(0): cutoff: false -> false
Thread(0): cutoff: false -> true
Thread(0): cutoffDuration: 0 -> 1
Turn over: next player: Player(1)
Thread(0): cutoffDuration: 1 -> 2
Turn over: next player: Player(2)
Thread(0): cutoffDuration: 2 -> 3
Turn over: next player: Player(3)
Thread(0): cutoffDuration: 3 -> 4
Turn over: next player: Player(4)
....
Thread(0): [removed]
Turn over: next player: Player(n)
....
Thread(1): [removed]
Turn over: next player: Player(n+k)
```

### 8.3.3 Teszt: spóra megevése

- **Leírás**

A spórák megevését teszteli.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt teszteljük, hogy a rovarok képesek-e megenni a spórákat, illetve hogy ekkor a megfelelő effektek érvénybe lépnek-e rajtuk.

- **Bemenet**

*/rand -disable*

*# tesztpálya betöltése*

*# több insect van a pályán, egyesek effektek hatásai alatt*

*# mindegyik a 0-ás tectonon van alaptól, innen az 1-esre tudnak mozogni, a 4-es pedig túl távoli*

*/load testmap.map*

*/start*

*# sikertelen evés, mivel nem egy tektonon vannak.*

*/manualtrigger !eat 0 0*

*# sikeres evések, effektek kipróbálása*

*/manualtrigger !eat 0 1*

*/manualtrigger !eat 0 2*

*/manualtrigger !eat 0 3*

*/manualtrigger !eat 0 4*

- **Elvárt kimenet**

*# az első sornak nincs hatása*

*Insect(0): speedModifier: 0 -> 1*

*Spore(1): [removed]*

*Insect(0): paralyzed: false -> true*

*Spore(2): [removed]*

*Insect(0): clawParalyzed: false -> true*

*Spore(3): [removed]*

*Insect(0): speedModifier: 1 -> 0*

*Spore(4): [removed]*

### 8.3.4 Teszt: gombafonal növesztése

- **Leírás**

A gombafonalak növesztését teszteli.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt teszteljük, hogy a fonalak növesztésére vonatkozó szabályok fennállnak-e.

- **Bemenet**

*/rand -disable*

*# tesztpálya betöltése*

*# több insect van a pályán, egyesek effektek hatásai alatt*

*# mindegyik a 0-ás tectonon van alaphól, innen az 1-esre tudnak mozogni, a 4-es pedig túl távoli*

*/load testmap.map*

*/start*

*# ide nem szabadna fonalat növeszteni, mert itt már van*

*!grow 0*

*# itt nincs, ide tehát szabad*

*!grow 4*

- **Elvárt kimenet**

*# az első sornak nincs hatása*

Thread(n): [created]

Tecton(4): threads: Thread(n) added

### 8.3.5 Teszt: gombatest növesztése

- **Leírás**  
A gombatestek növesztését teszteli.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Azt teszteljük, hogy a gombatestek növesztésére vonatkozó szabályok fennállnak-e.
- **Bemenet**

```
/rand -disable
# tesztpálya betöltése
# több insect van a pályán, egyesek effektek hatásai alatt
# mindegyik a 0-ás tectonon van alaphól, innen az 1-esre tudnak mozogni, a 4-es pedig túl
távoli
/load testmap.map
/start
```

```
# ide szabad növeszteni, és van is itt elegendő spóra hozzá (illetve fonal)
/manualtrigger !plant 0
```

```
# ide nem tudunk növeszteni, nincs hozzáadás spóra
/manualtrigger !plant 1
```

```
# itt fonal nincs hozzá
/manualtrigger !plant 4
```

```
# ide tudunk növeszteni, mivel a fonal evett meg korábban rovar
/manualtrigger !plant 2
```

- **Elvárt kimenet**

```
MushroomStem(0): [created]
Tecton(0): stems: MushroomStem(0) added
# a második sornak nincs hatása
# a harmadik sornak sincs
MushroomStem(1): [created]
Tecton(2): stems: MushroomStem(1) added
Thread(n): eaten: true -> false
```

### 8.3.6 Teszt: spórák dobása

- **Leírás**

A spórák dobását teszteli

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt teszteljük, hogy a spórák dobására vonatkozó szabályok fennállnak-e.

- **Bemenet**

*/rand -disable*

*# tesztpálya betöltése*

*# több insect van a pályán, egyesek effektek hatásai alatt*

*# mindegyik a 0-ás tectonon van alapból, innen az 1-esre tudnak mozogni, a 4-es pedig túl távoli*

*/load testmap.map*

*/start*

*# sikeres dobás tesztelése*

*/manualtrigger !throw 0 0*

*# sikertelen dobás, túl távoli a tecton*

*/manualtrigger !throw 0 4*

*# fejlettebb gombatest viszont tud messzebbre dobni*

*/manualtrigger !throw 1 4*

- **Elvárt kimenet**

Spore(n): [created]

MushroomStem(0): numThrownSpores: 0 -> 1

# a második sornak nincs hatása

Spore(n+1): [created]

MushroomStem(1): numThrownSpores: 0 -> 1



### 8.3.7 Teszt: kör és round vége

- **Leírás**  
A körök/roundok végét teszteli
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Megnézzük hogy a játékosok megfelelő sorrendben következnek-e egymás után, illetve hogy a round vége is megtörténik-e.

- **Bemenet**

*/rand -disable*

*# tesztpálya betöltése*

*# több insect van a pályán, egyesek effektek hatásai alatt*

*# mindegyik a 0-ás tectonon van alaptól, innen az 1-esre tudnak mozogni, a 4-es pedig túl távoli*

*/load testmap.map*

*/start*

*!endturn*

*!endturn*

*!endturn*

*!endturn*

- **Elvárt kimenet**

Turn over: next player: Player(1)

Turn over: next player: Player(2)

Turn over: next player: Player(3)

Round over

Turn over: next player: Player(0)

### **8.4 A tesztelést támogató programok tervei**

A tesztelést a dinamikus JUnit tesztekkel valósítjuk meg. Mindegyik teszteset egy külön mappában helyezkedik el a Tests mappában. A mappa neve fogja azonosítani a tesztet. Egy tesztnek tartalmaznia kell az alábbiakat:

- **input.txt:** parancssori utasításokat tartalmaz, minden sorban pontosan egyet. A teszt beolvassa a tartalmát, majd egyesével végrehajtja a benne talált parancsokat. A legelsőnek a pályát kell betöltenie.
- **output.txt:** az utasítások végrehajtásakor a program log-olja, hogy mi történik, ezek írja ki ebbe a fájlba.
- **expected.txt:** a teszt várt eredménye.

Ha az output.txt megegyezik az expected.txt-vel, akkor sikeres a teszt.

A szükséges pályák mindig tesztfuttatáskor le lesznek generálva.

**8.5 Napló**

Kezdet	Időtartam	Résztevők	Leírás
2025.04.02.	1.5 óra	Bohus Dúcz Galambos Horváth Végvári	Tevékenység: megbeszélés, előző rész hibáinak kijavítása, feladatok felosztása
2025.04.07 14:00	1.5 óra	Végvári	Tevékenység: Tesztelést támogató program írása
2025.04.09 17:00	1.5 óra	Bohus	Tevékenység: Osztályok írása
2025.04.09 17:00	1.5 óra	Horváth	Tevékenység: Osztályok írása
2025.04.10 8:00	6 óra	Horváth	Tevékenység: Aktivitás diagramok, state chartok
2025.04.10 20:30	0,5 óra	Galambos	Tevékenység: Osztályok írása
2025.04.11 12:00	1.5 óra	Bohus	Tevékenység: Osztályok írása, activity diagramok
2025.04.11 19:00	1.5 óra	Végvári	Tevékenység: Osztályok írása
2025.04.11 20:00	2 óra	Dúcz	Tevékenység: Tesztesetek elkészítése
2025.04.11 20:00	1 óra	Végvári	Tevékenység: Osztályok írása
2025.04.13 02:00	2 óra	Galambos	Tevékenység: Osztályok írása, dokumentum átnézése
2025.04.13 12:00	1.5 óra	Végvári	Tevékenység: Activity diagramok írása
2025.04.13 13:00	0.5 óra	Bohus	Tevékenység: pszeduo kód, dokumentum rendezése
2025.04.13 13:30	0.5 óra	Végvári	Tevékenység: Osztályok írása