**5. Szkeleton tervezése**

26 – five\_guys

Konzulens:

Huszerl Gábor

Csapattagok

| Lekli Zsolt | JENKHQ | zsoltlekli@gmail.com |
| --- | --- | --- |
| Fejes Ábel | AF1Y1C | fejesabel@gmail.com |
| Zelch Csaba | LK0617 | zch03po@gmail.com |
| Szolár Soma | E05SC6 | somaszolar1@gmail.com |
| Toronyi Zsombor | S8F7DV | toronyizsombi@gmail.com |

2025. 03. 17.

1. **Szkeleton tervezése**
   1. ***A szkeleton modell valóságos use-case-ei***
      1. **Use-case diagram**

-

* + 1. **Use-case leírások**

Aktor: mindig a tesztelő

| **Név** | Első gombatest lehelyezése olyan tektonra, amelyre le lehet helyezni |
| --- | --- |
| **Leírás, cél** | Azt az esetet teszteli, amikor egy gombász le szeretné tenni az első gombatestjét a játék kezdetekor egy olyan tektonra, amelyre le tudja a szabályok szerint. Ez egy olyan tekton lehet, amin nincs másik gombatest és a fajtája nem AbsorbingTecton. |
| **Előfeltétel** | Lennie kell:   * t4: MultiThreadTecton - A tekton, amire le szeretné rakni a gombász a gombatestjét.   Teljesülnie kell:   * t4 SingleThreadTecton vagy MultiThreadTecton * t4-en nincs gombatest * Ha t4 SingleThreadTecton, akkor t4-en nincs gombafonál (FungalThread) |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | A tester meghívja a t4-en a **putFirstMushroom** metódust. |
| **Forgatókönyv** | * Mivel a t4 MultiThreadTecton, ezért a **putFirstMushroom** függvény csak azt ellenőrzi le, hogy van-e tektonon gombatest. (Ha SingleThreadTecton lenne, akkor azt is le kéne ellenőrizni, hogy van-e a tektonon gombafonál. AbsorbingTecton esetén a függvény szimplán false-al tér vissza). * Ha nincs, akkor létrehoz először egy fonalat, utána egy gombatestet. * A gombatest fonalának (thread attribútum) a létrehozott gombafonalat állítja be a függvény. * A fonálnak és a gombatestnek is beállítja az attribútumait a tektonnak megfelelően. * Ezek után a **putFirstMushroom** függvény true-val tér vissza. |
| **Elvárt kimenet** | A **putFirstMushroom** függvény true-val tér vissza, tehát t4 tektonon lesz egy új gombafonál és egy új gombatest. |
| **Hivatkozott szekvencia diagram** | első\_gombatest\_lehelyezése\_true |

| **Név** | Első gombatest lehelyezése egy tektonra, amelyre nem lehet lehelyezni (AbsorbingTecton) |
| --- | --- |
| **Leírás, cél** | Azt az esetet teszteli, amikor egy gombász le szeretné tenni az első gombatestjét a játék kezdetekor egy olyan tektonra, amelyre nem tudja a szabályok szerint. Ez ebben az esetben egy AbsorbingTecton. |
| **Előfeltétel** | Lennie kell:   * t5: AbsorbingTecton - A tekton, amire le szeretné rakni a gombász a gombatestjét.   Teljesülnie kell:   * t5 egy AbsorbingTecton. |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | A tester meghívja a t5-ön a **putFirstMushroom** metódust. |
| **Forgatókönyv** | Mivel a t5 tekton egy AbsorbingTecton, ezért a **putFirstMushroom** false-al tér vissza. |
| **Elvárt kimenet** | A t5 tekton változatlan marad, tehát nem kerül rá se új gombafonál, se új gombatest. |
| **Hivatkozott szekvencia diagram** | első\_gombatest\_lehelyezése\_false\_absorb |

| **Név** | Első gombatest lehelyezése egy tektonra, amelyre nem lehet lehelyezni (van gombatest a tektonon) |
| --- | --- |
| **Leírás, cél** | Azt az esetet teszteli, amikor egy gombász le szeretné tenni az első gombatestjét a játék kezdetekor egy olyan tektonra, amelyre nem tudja a szabályok szerint. Ez ebben az esetben egy olyan tekton, amelyen már van gombatest. |
| **Előfeltétel** | Lennie kell:   * t1: MultiThreadTecton - A tekton, amire le szeretné rakni a gombász a gombatestjét. * f: FungalThread - A gombafonál, amely elágazik t1-en. * m: Mushroom - A gombatest, amely a t1 tektonon helyezkedik el.   Teljesülnie kell:   * m gombatest és f fonál a t1 tektonon helyezkedik el. |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | A tester meghívja a t1-en a **putFirstMushroom** metódust. |
| **Forgatókönyv** | * Mivel a t1 MultiThreadTecton, ezért a **putFirstMushroom** függvény csak azt ellenőrzi le, hogy van-e tektonon gombatest. * Mivel van, ezért a függvény visszatérési értéke false. |
| **Elvárt kimenet** | A t1 tekton változatlan marad. |
| **Hivatkozott szekvencia diagram** | első\_gombatest\_lehelyezése\_false\_vangomba |

| **Név** | Új gombatest lerakása |
| --- | --- |
| **Leírás, cél** | Azt az esetet teszteli, amikor egy gombász le szeretne tenni egy új gombatestet egy olyan tektonra, amelyre le tudja a szabályok szerint. |
| **Előfeltétel** | Lennie kell:   * f: FungalThread - A gombászhoz tartozó gombafonál. * t3: SingleThreadTecton - A tekton, amire a gombász le szeretné helyezni egy új gombatestet. * s1: SlowingSpore, s2: NoCutSpore, s3: ParalysedSpore - A t3 tektonon lévő spórák.   Teljesülnie kell:   * t3 tektonon el kell lennie ágazva az f fonálnak. * t3 tektonon kell, hogy legyen legalább 3 olyan spóra, amely az f fonálhoz tartozik. Tehát ez esetben s1, s2 és s3-nak is az f-hez kell tartoznia. * t3 tektonon nem lehet alapból gombatest. |
| **Hivatkozott kommunikációs diagram** | Diagram 5 |
| **Bemenet** | A tester meghívja a **growMushroom**(t3) metódust f a FungalThread objektumon t3 paraméterrel. |
| **Forgatókönyv** | * A **growMushroom** először ellenőrzi, hogy t3 tektonon van-e gombatest. * Ha nincs, akkor a FungalThread lekéri a spórák listáját a célzott tektonról a **getSpores**() metódussal. * A tekton visszaküldi az elérhető spórák listáját (s1, s2, s3). * Ha mindhárom spóra ugyanazt a fonalat használja, akkor, létrejön a Mushroom objektum. * A Mushroom beállítja a pozícióját a tektonon (**setPosition**). * Továbbá beállítja a fonalát is (**setThread**). * A Mushroom elhelyezése a tektonon végbemegy a **putMushroom** függvény segítségével. * Ezek után a spórákat eltávolítja a **growMushroom** függvény (**removeSpores**). * A folyamat végén a **growMushroom** metódus visszatér true értékkel, amely megerősíti a sikeres gombatest létrehozást. |
| **Elvárt kimenet** | A t3 tektonon megjelenik az új gombatest a megfelelő attribútumokkal. |
| **Hivatkozott szekvencia diagram** | New Mushroom Creation |

| **Név** | Fejletlen gombatest spóra szórása olyan tektonra, amelyre tud |
| --- | --- |
| **Leírás, cél** | Azt az esetet teszteli, amikor egy gombász hatására egy mushroom objektum (ami fejletlen) spórát szór egy szomszédos tektonra. Ebben az esetben ez megtehető a játék szabályai szerint. |
| **Előfeltétel** | Lennie kell:   * t3:SingleThreadTecton - A cél tekton, amire a spórát a gombatest lőni szeretné. * t1:MultiThreadTecton - A tekton amin a gomba van. * m:Mushroom - A gomba ami lő. * s2:Spore - Spóra, amit a gomba kilő.   Teljesülnie kell:   * t1 és t3 szomszédosak: Benne vannak egymás neighbors listájában. * m rajta van t1-en:   + t1.mushroom=m   + m.position=t1 * m-nek van spórája, amit kilő, ez most s2:   + m.spores[0]=s2 * m fejletlen gombatest: m.state=Unevolved (alapértelmezett állapot létrehozás után) |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | A Tester meghívja az m shootSpore függvényét t3-at átadva paraméterül . |
| **Forgatókönyv** | * A **shootspore** függvény törzsében ellenőrizzük, hogy nem üres a spores. Ha üres lenne visszatérne false-al. Most nem üres, benne van s2. * A t3objektumon meghívódik a **putSpore** függvény, paraméterül megkapja az m spores listájának első elemét(s2) és az m position-ét (t1). * A **putSpore** függvény ellenőrzi, hogy a paraméterül kapott position benne van-e a t3 neighbors listájában. Most benne, ezért a függvény ráhelyezi t3 tektonra az s2 spórát, és true-val tér vissza. * Ezek után a **shootSpore** függvény is true-val tér vissza. |
| **Elvárt kimenet** | A t3 tektonon rajta lesz s2 spóra. Továbbá s2 spóra kikerült az m Mushroom spores listájából. |
| **Hivatkozott szekvencia diagram** | fejletlen\_spóraszórás\_true |

| **Név** | Fejlett gombatest spóra szórása olyan tektonra, amelyre tud |
| --- | --- |
| **Leírás, cél** | Azt az esetet teszteli, amikor egy gombász hatására egy mushroom objektum (ami fejlett) spórát szór egy olyan tektonra, ami a szomszédjának a szomszédja. Ebben az esetben ez megtehető a játék szabályai szerint. |
| **Előfeltétel** | Lennie kell:   * t2: SingleThreadTecton - A cél tekton, amire a spórát a gombatest lőni szeretné. * t1:MultiThreadTecton - A tekton amin a gomba van. * m:Mushroom - A gomba ami lő. * s4: SpeedSpore - Spóra, amit a gomba kilő. * t3: SingleThreadTecton - A tekton, ami szomszédja t2-nek és t1-nek is.   Teljesülnie kell:   * t1 és t2 nem szomszédosak, de t1 szomszédjának (t3) a szomszédja t2. * m rajta van t1-en:   + t1.mushroom=m   + m.position=t1 * m-nek van spórája, amit kilő, ez most s4:   + m.spores[0]=s4 * m fejlett gombatest: m.state=Evolved * t3-nak szomszédja t1 és t2 is. |
| **Hivatkozott kommunikációs diagram** | Diagram 5 |
| **Bemenet** | A Tester meghívja az m shootSpore függvényét t2-őt átadva paraméterül . |
| **Forgatókönyv** | * A **shootspore** függvény törzsében ellenőrizzük, hogy nem üres a spores. Ha üres lenne visszatérne false-al. Most nem üres, benne van s4. * A t2objektumon meghívódik a **putEvolvedSpore** függvény, paraméterül megkapja az m spores listájának első elemét (s4) és az m position-ét (t1). * A **putEvolvedSpore** függvény először ellenőrzi, hogy a paraméterül kapott position benne van-e a t2 neighbors listájában. Most nincs benne. * Ezek után a **putEvolvedSpore** függvény az t2 neighbors (szomszédos tektonok listája) attribútumán végig iterál, és mindegyik szomszédos tektonon meghívja a **getNeighbors** metódust. * Ellenőrzi, hogy van-e olyan tekton a t2 szomszédjai között, aminek a getNeighbors által visszaadott tekton listában benne van t1. Most t3 egy ilyen tekton. * Ekkor tudjuk, hogy t2 t1 szomszédjának a szomszédja. Így a **putEvolvedSpore** függvény ráhelyezi az s4-et t2 tektonra és true-val tér vissza. * Ezek után a **shootSpore** függvény is true-val tér vissza. |
| **Elvárt kimenet** | A t2 tektonon rajta lesz s4 spóra. Továbbá s4 spóra kikerült az m Mushroom spores listájából. |
| **Hivatkozott szekvencia diagram** | fejlett\_spóraszórás\_true |

| **Név** | Sikertelen spóraszórás, mert nem szomszédos a céltekton. |
| --- | --- |
| **Leírás, cél** | Egy Mushroom objektum megpróbál spórát lőni egy SingleThreadTecton célpontra. Mivel nem szomszédos a Mushroom tektonja a SingleThreadTecton-al, a művelet sikertelen. |
| **Előfeltétel** | Lennie kell:   * t2:SingleThreadTecton - a cél tekton * t1:MultiThreadTecton - a tekton amin a gomba van * m:Mushroom - a gomba ami lő * s2:Spore - spóra, amit a gomba kilő   Teljesülnie kell:   * t1 és t2 nem szomszédosak: Nincsenek egymás neighbors listájában. * m rajta van t1-en:   + t1.mushroom=m   + m.position=t1 * m-nek van spórája, amit kilő, ez most s2:   + m.spores[0]=s2 * m fejletlen gombatest: m.state=Unevolved (alapértelmezett állapot létrehozás után) |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | A Tester meghívja az m shootSpore függvényét t2-őt átadva paraméterül |
| **Forgatókönyv** | * A **shootspore** függvény törzsében ellenőrizzük, hogy nem üres a spores. Ha üres lenne visszatérne false-al. Most nem üres, benne van s2. * Az t2objektumon meghívódik a **putSpore** függvény, paraméterül megkapja az m spores listájának első elemét(s2) és az m position-ét - t2 tudja, hogy mit és honnan kap.   + A függvény elleneőrzi, hogy a paraméterül kapott position benne van-e az t2 neighbors listájában. Most nincs benne, ezért false-al tér vissza. * putSpore=false -> nem lehet t2-re spórát tenni m-nek. * shootSpore=false->sikertelen volt a spóra lövés. |
| **Elvárt kimenet** | A shootSpore false-al tér vissza, sikertelen volt a spóra lövés.  Nem került rá s2 t2-re. |
| **Hivatkozott szekvencia diagram** | fejletlen\_spóraszórás\_false |

| **Név** | Gombafonál sikeres elágaztatása tektonra. |
| --- | --- |
| **Leírás, cél** | Egy FungalThread ágat növeszt egy MultiThread felé. Amennyiben a MultThread egyik szomszédján van már azonos fajú FungalThread. A művelet sikeres. |
| **Előfeltétel** | Lennie kell:   * t3:SingleThreadTecton - róla ágaztatunk el * t4:MultiThreadTecton - céltekton * f:FungalThread - növeszteni kívánt fonál   Teljesülnie kell:   * t3 és t4 szomszédosak: Benne vannak egymás neigbors listáiban. * t3-on lévő fonalak közt van olyan, amilyet szeretnénk elágaztatni t4-re   + t3.thread=f   + t4-re f-et akarunk ágaztatni. |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | A Tester meghívja az f branchThread függvényét t4-et átadva paraméterül. |
| **Forgatókönyv** | * A **branchThread**-en belül meghívódik a **putThread** függvény t4-en, f átadja saját magát.   + A t4 tekton meg nézi, hogy a neigbors-ben lévő tektonokon milyen fonalak vannak. A szomszédaitól le kéri a fonál tömbjüket/fonalukat, és megnézi, hogy talál-e benne f fonállal megegyezőt. Most talál, t3 szomszédján rajta van f.   + t4 felveszi a listájába f-et * putThread=true->fel lehet tenni t4-re az f fonalat. * Visszatérve a branchThread-be f felveszi a listájába t4-et és visszatér true-val. |
| **Elvárt kimenet** | * A branchThread igazzal tér vissza. * t4 felvette a listájába f-et . * f felvette a listájába t4-et. |
| **Hivatkozott szekvencia diagram** | elágazás\_true |

| **Név** | Gombafonál sikertelen elágaztatása egyfonalas tektonra, mert már van rajta. |
| --- | --- |
| **Leírás, cél** | Egy FungalThread megpróbál ágat növeszteni egy SingleThreadTecton felé. Mivel már tartalmaz egy fonalat, és csak egyet fogadhat, a művelet sikertelen. |
| **Előfeltétel** | Lennie kell:   * f2: FungalThread - növeszteni kívánt fonál * t3:AbsorbingTecton - erről a tektonról menne a terjesztés * t2:SingleThreadTecton - céltekton * f1: FungalThread - fonál ami már van t2-őn   Teljesülnie kell:   * t3 és t2 szomszédosak: Benne vannak egymás neighbors listáiban. * t3-on rajta van f2 és f1:   + t3.threads-ben benne van f2 és f1   + f2 és f1.tectons tartalmazza t3-at * t2-őn csak f1 van rajta: hasonló mint az előbbi |
| **Hivatkozott kommunikációs diagram** | Diagram 4 |
| **Bemenet** | A tesztelő meghívja az f2 branchThread függvényét t2-őt átadva paraméterül. |
| **Forgatókönyv** | * A **branchThread**-en belül meghívódik a **putThread** függvény t2-őn, f2 átadja saját magát.   + A putThread-en belül: t2 először megnézi, hogy van-e már rajta fonál. Most f1 van rajta.   + Mivel van már rajta fonál és egyfonalas tektonokon csak egy fonál ágazhat el, ezért * putThread**=**false->t2 nem tudja elraktározni f2-őt. * branchThread**=**false->f2 nem tehető t2-re. |
| **Elvárt kimenet** | A branchThread false-al tér vissza.   * t2.thread-e nem lesz f2. (továbbra is f1) * f2.tectons nem tartalmazza t2-őt. |
| **Hivatkozott szekvencia diagram** | elágazás\_false |

| **Név** | Új spóra termelődése. |
| --- | --- |
| **Leírás, cél** | Egy gombatest generál egy spórát minden második körben.  Most ezt a viselkedést a Tester idézi elő. |
| **Előfeltétel** | Lennie kell:   * m:Mushroom - termelő gomba |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | A Tester meghívja az m generateSpore függvényét egy SpeedSpore enumot átadva paraméterként. |
| **Forgatókönyv** | * A **generateSpore** belsejében   + m létrehoz egy spórát, amelyre:     - s1:SpeedSpore - a kapott enum-nak megfelelő a Spore     - s1.thread=m.thread - s1 ctor-a megkapja m.thread-et és ezt állítja be s1.thread-nek - fonala megegyezik m fonalával   + s1-et eltárolja a spores listájában. * generateSpore=true->sikeres a spóra generálás |
| **Elvárt kimenet** | * A generateSpore igazzal tér vissza. * Létrejött s1 és ezt eltárolta m. |
| **Hivatkozott szekvencia diagram** | generate\_spore |

| **Név** | **Gombatest elpusztulása** |
| --- | --- |
| **Leírás, cél** | Egy Mushroom objektum kilövi a 10. spóráját is, ekkor a gombatestnek el kell pusztulnia |
| **Előfeltétel** | Lennie kell:   * m: gombatest, amely a 9. spóráját is már elszórta, spores tömbjében lennie kell még legalább egy spórának, amit még el tud szórni * t1 : erre a tektonra szórja az m gombatest a spóráját * t2: ezen a tektonon helyezkedik el a gombatest * f : gombafonál, amely ugyanabból a gombafajból van, mint az m gombatest * s: spóra, amelyet az m gombatest elszórhat |
| **Hivatkozott kommunikációs diagram** | Diagram 3 |
| **Bemenet** | A Tester meghívja az m gombatest shootSpore függvényét t1-et átadva paraméterül |
| **Forgatókönyv** | A shootSpore metódushívás hatására a  kilőjük az s spórát a t1 tektonra meghívva a tekton putSpore metódusát, paraméterül küldjük az s spórát, és azt a tektont is, amelyiken a gombatest található, azaz a t2-t.  Ezután növeljük a kilőtt spórák számát 1-el ha sikeres a kilövés, leellenőrizzük, hogy kilőttük-e a 10. spórát  Mivel kilőttük a 10. spórát törölnünk kell a gombatestet, ezt először a tektonról vesszük le a removeMushroom() metódushívás segítségével, majd a gombafajhoz tartózó gombafonálnál töröljük azokat a részeket a deleteUnnecessaryThreads() metódushívás segítségével, amelyek nem kapcsolódnak azonos fajhoz tartozó gombatesthez.  Egy külső tesztelő objektum lekérdi, hogy hány spórát lőttünk ki eddig a getshootedSporesCount() segítségével,  ha ez megegyezik 10-el, akkor meghívja a gombatest destruktorát. |
| **Elvárt kimenet** | A t2 tektonon az m gombatestnek el kell tűnnie, illetve meg kell szűnnie létezni. A t1 tektonon rajta kell lennie az s spórának. |
| **Hivatkozott szekvencia diagram** | gombatest\_elpusztulása |

| **Név** | **Gombatesthez nem kapcsolódó fonálrész eltávolítása** |
| --- | --- |
| **Leírás, cél** | Gombatest megszűnése, fonál felszívódása, tekton kettétörése és rovar fonálvágása során keletkezhetnek olyan fonálrészek, melyek nincsenek már kapcsolatban ugyanolyan gombafajból származó gombatesttel. Ezeket törölni kell. |
| **Előfeltétel** | Lennie kell:  t1, t2, t3, t4, t5, t6 tektonok  f gombafonál, amely elágazik a t1, t2, t3, t5, t6 tektonokon  m gombatest, amely a t2 tektonon van |
| **Hivatkozott kommunikációs diagram** | Diagram 2 |
| **Bemenet** | A Tester meghívja a gombafonál deleteUnnecessaryThreads() metódusát |
| **Forgatókönyv** | A metódushívás után sorban végigmegyünk azokon a tektonokon, amelyeken a gombafonál elágazik, és megkérdezzük, ha van-e rajtuk gombatest. Ha nincs megnézzük a sorban következő tektonra is, ha van, és ez a gombatest ugyanahhoz a gombafajhoz származik, akkor megkeressük azt a fonálrészt, amely ezzel a gombatesttel összeköttetésben van.  Ez a következő képpen történik: meghívjuk a getThreadSection(f: FungalThread) metódust arra a tektonra, amelyen a gombatest rajta van, ez visszaadja azokat a tektonokat, amelyek a gombatest tektonjával szomszédosak, és elágazik rajtuk az f gombafonál. Ezután meghívjuk ezt a metódust az így kapott tektonokra is, és ezt addig csináljuk, amig találunk új tektont. Ezeket a tektonokat behelyezzük egy listába.  Ha már nem találunk új tektonokat, akkor tovább megyünk azokon a tektonokon, amelyen a gombafonál elágazik, és megkérdezzük, ha van-e rajtuk gombatest.  Mikor ezt befejeztük, azokra a tektonokra, amelyek nem kerültek be az algoritmus által elkészített listába, de benne vannak a gombafonál által tárolt tectons listába, meghívjuk a removeThread(f) metódusukat, ezzel levesszük róluk a fonalat, majd kivesszük őket a tectons listából is, ezzel biztosítva, hogy a fonál nem lesz rajtuk elágazva. |
| **Elvárt kimenet** | A t5 és t6 tektonokon nincs az f gombafonál elágazva, mivel ez egy olyan fonálrész, amely nem kapcsolódik egy ugyanolyan fajból származó gombatesthez.  A metódushívás true értékkel térjen vissza. |
| **Hivatkozott szekvencia diagram** | gombatesthez\_nem\_kapcsolódó\_fonálrész\_eltávolítása |

| **Név** | **Tekton kettétörése** |
| --- | --- |
| **Leírás, cél** | A t3 tektonnak meghívjuk a break() metódusát, ennek hatására két új tekton kell keletkezzen. |
| **Előfeltétel** | Lennie kell:  t1, t2, t3, t4, t5 tektonok, a t3 tekton szomszédos az összes többivel  m: gombatest, amely a t1 tektonon található  f: gombafonál, amely a t1, t2, t3 tektonokon ágazik el  s: spóra, amely a t3 tektonon található  i: rovar, amely a t3 tetkonon található |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | A t3 tektonnak a Tester meghívja a break() metódusát |
| **Forgatókönyv** | A t3 tekton break() metódusa először két új tektont hoz létre. Ezeknek először be kell állítani a szomszédait. Mivel a szomszédok sorban következnek a neighbors listában, és a tekton a listában tárolt első és az n/2 (felkerekített) szomszédjánál tört el, így tudni fogjuk, hogy a két új tekton melyik tektonokkal lesz szomszédos. Ezután a t3 tektonnal szomszédos tektonoknak is beállítjuk a szomszédsági listáját, valamint kivesszük belőle a t3 tektont.  Ha van a t3 tektonon spóra, akkor ezt töröljük.  Ha van a tektonon rovar, ezt ráhelyezzük az elsőnek létrehozott tektonra, és frissítjük a rovar pozícióját.  Ha a t3 tektonon elágazott fonál, akkor erről levesszük a fonalat, majd töröljük azokat a fonálrészeket, amelyek nincsenek összeköttetésben ugyanolyan gombafajból származó tektonnal.  Végül egy külső kontroller törli a tectont, amely kettétört. |
| **Elvárt kimenet** | Létrejött a t6 és t7 tekton, illetve a t3 tekton megszűnt létezni, a bogár a t6 tektonra került, a spóra, ami a t3 tektonon volt megszűnt létezni, a keletkezett tektonoknak és a megmaradt tektonoknak szomszédsági listájuk helyesen beállítódott.  A break() metódus true-val tért vissza |
| **Hivatkozott szekvencia diagram** | tekton\_kettétörése |

| **Név** | **Tekton kettétörése, ha a tektonon gombatest van** |
| --- | --- |
| **Leírás, cél** | Egy olyan tektonnak hívjuk meg a break() metódusát, amelyen gombatest található, ekkor a tekton nem törhet ketté |
| **Előfeltétel** | Lennie kell:  t1: tekton  m: gombatest, amely a t1 tektonon van |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | A t1 tektonnak a Tester meghívja a break() metódusát |
| **Forgatókönyv** | A metódushívás után nem történik semmi, mivel a tektonon gombatest van. |
| **Elvárt kimenet** | A break() metódus false-val tér vissza |
| **Hivatkozott szekvencia diagram** | tekton\_kettétörése\_ha\_a\_tektonon\_gombatest\_van |

| **Név** | **Fonál felszívódása tektonról** |
| --- | --- |
| **Leírás, cél** | Egy AbsorbingTecton objektumnak meghívtuk az absorb metódusát, ezáltal a rajta levő fonalak fell kell szívódjanak |
| **Előfeltétel** | Lennie kell:  t3: AbsorbingTecton  f1, f2 gombafonál, amelyek elágaznak a t3 tektonon |
| **Hivatkozott kommunikációs diagram** | Diagram 4 |
| **Bemenet** | A Tester meghívja a t3 tekton absorb() metódusát |
| **Forgatókönyv** | A t3 tektonon két gombafonál ágazik el, az f1 és az f2. Mivel a t3 tektonról felszívódnak a fonalak az absorb() metódushívás során törölni kell, az f1 és az f2 fonalat a t3 tektonról, azaz, a két fonál ennek során már nem lesz elágazva a tektonon, ezért meghívódik először az f1 removeTecton(t3) metódusa, majd az f2 removeTecton(t3) metódusa, amelyek kitörlik a t3 tektont a gombafonalak tecons listájából. Ezután a deleteUnnecessaryThreads() függvények segítségével kitöröljük azokat a gombafonál részeket, amelyek már nincsenek összeköttetésben egy ugyanolyan fajból származó gombatesttel. |
| **Elvárt kimenet** | A t3 tektonon nincsenek gombafonalak elágazva, azaz a t3 threads listája üres, és az f1 és f2 tectons listája sem tartalmazza a t3-at.  Az absorb() metódus true értékkel tér vissza. |
| **Hivatkozott szekvencia diagram** | fonál\_felszívódása\_tektonról |

| **Név** | **Gombafonál elvágása** |
| --- | --- |
| **Leírás, cél** | A rovar elvágja a gombafonalat a saját tektonja, és a kiválasztott tekton közötti résen, ezáltal az összes olyan gombafonalat, amely a résen átmegy a kiválasztott tektonról töröljük. |
| **Előfeltétel** | Lennie kell:  t2, t3 tekton  f1, f2 gombafonál  f1 és f2 gombafonál is elágazik a t3 tektonon, azonban t2 tektonon csak az f1 ágazik el  i: rovar, amely a t2 tektonon található |
| **Hivatkozott kommunikációs diagram** | Diagram 4 |
| **Bemenet** | A Tester az i rovarnak meghívja a cut metódusát paraméterként a t2 tektont adva át |
| **Forgatókönyv** | A cut(t2) metódushívás után leellenőrizzük, hogy a paraméterül kapott t2 tekton szomszédos-e a t3 tektonnal, azaz azzal a tektonnal, amelyen a rovar rajta van.  Ha ez teljesül, akkor mindkét tektonról lekérjük a gombafonalakat mindkettőről külön-külön. Ha találunk olyat, amely mindkettőn elágazik, akkor ezt levesszük a t2 tektonról, majd a t2 tektont kivesszük a fonál listájából, végül pedig a deleteUnnecessaryThreads() függvényhívással kitöröljük azokat a fonálrészeket, amelyek nincsenek összeköttetésben azonos fajból származó tektonnal. |
| **Elvárt kimenet** | A t2 tektonon már nincs elágazva az f1 gombafonál.  A rovar cut metódusa true értékkel tér vissza. |
| **Hivatkozott szekvencia diagram** | gombafonál\_elvágása |

| **Név** | **Rovar lehelyezése tektonra sikeres** |
| --- | --- |
| **Leírás, cél** | A játék elején egy rovarász első lépése az lesz, hogy letegye a rovarját, amit irányítani fog a játék során egy tektonra. |
| **Előfeltétel** | Lennie kell :  t1: tecton |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | Tester meghívja a t1 tekton putFirstInsect() metódusát. |
| **Forgatókönyv** | A putFirstInstect() metódushívás után a tekton leellenőrzi magán, hogy nem foglalt, azaz nincs rajta már rovar. Ez után létrehoz egy új Rovar objektumot, aminek beállítja a pozícióját magára, illetve beállítja magán hogy az a rovar rajta van. Végül true értékkel tér vissza a hívó felé, jelezve a művelet sikerességét. |
| **Elvárt kimenet** | A t1 tektonon megjelenik az új rovar, megfelelő pozícióval. |
| **Hivatkozott szekvencia diagram** | rovarlerakás\_sikeres |

| **Név** | Rovar lehelyezése tektonra sikertelen |
| --- | --- |
| **Leírás, cél** | A játék elején egy rovarász első lépése az lesz, hogy letegye a rovarját, amit irányítani fog a játék során egy tektonra. |
| **Előfeltétel** | Lennie kell :  t2: tecton |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | Tester meghívja a t1 tekton putFirstInsect() metódusát. |
| **Forgatókönyv** | A putFirstInsect() metódushívás után a tekton önellenőrzése során arra jut, hogy már foglalt, azaz az Insect mezője nem üres. Ekkor fasle értékker tér vissza a hívó felé jelezve azt, hogy a művelet sikertelen. |
| **Elvárt kimenet** | Nem történik változás. |
| **Hivatkozott szekvencia diagram** | rovarlerakás\_sikertelen |

| **Név** | Rovar mozgása és spóraevése |
| --- | --- |
| **Leírás, cél** | A játék során egy rovarász lépései közé tartozik a mozgás lépése, ekkor a rovarász választ egy tektont amire mozogni szeretne, és a rovara átmegy oda, ha ez lehetséges.  Továbbá a már átment rovar ekkor az új tekton spórái közül megesz egyet és megkapja annak a hatását. |
| **Előfeltétel** | Kell legyen:  t2: tecton  t3: tecton  s: speedSpore  t: thread  i: insect  i Rovar t2 tektonon, s gyórsítóspóra t3 tektonon, köztük f gombafonál. |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | Tester meghívja a Rovar move(t3) metódusát. |
| **Forgatókönyv** | A move(t3) függvényhívás után a rovar meghívja t3 putInsect metódusát. A t3 megnézi, hogy t2 szomszédja és létezik -e köztük spóra. t2 igazzal való visszatérése után t3 meghívja a removeInsect metódust t2-n és beállítja magának i rovart. Igazzal tér vissza a rovar felé.  Ekkor a rovar lekérdezi t3 spóráit és a 0. elemén, azaz s-en meghívja az applyEffect metódust, aminek hatására a spóra meghívja a rovar setState metódusát és beállítódik a hatás.  A rovar igazzal tér vissza a hívó felé. |
| **Elvárt kimenet** | i Rovar t3 tektonon, gyorsított hatás a state mezőben, s már nem eleme t3 tekon spóráinak. |
| **Hivatkozott szekvencia diagram** | rovar\_mozgas\_sporaeves\_sikertelen |

| **Név** | Rovar mozgása sikertelen |
| --- | --- |
| **Leírás, cél** | A játék során egy rovarász lépései közé tartozik a mozgás lépése, ekkor a rovarász választ egy tektont amire mozogni szeretne, és a rovara átmegy oda, ha ez lehetséges.  Ebben az esetben nem lesz lehetséges fonal hiánya miatt. |
| **Előfeltétel** | Kell lennie:  t3: tecton  t4: tecton  i: Insect  i Rovar t4 tektonon, t3 szomszédos tekton t4 -gyel |
| **Hivatkozott kommunikációs diagram** | Diagram 1 |
| **Bemenet** | Tester meghívja a Rovar move(t3) metódusát. |
| **Forgatókönyv** | A move(t3) függvényhívás után a rovar meghívja t3 putInsect metódusát. t3 megnézi, hogy szomszédja -e t4 illetve, hogy van -e köztük fonal. Ebben az esetben a két tekton Threads mezőiben nincs azonos elem, ezért false értékkel tér vissza i Rovar felé, majd az szintén false értékkel tér vissza a hívó felé. |
| **Elvárt kimenet** | Nincs változás |
| **Hivatkozott szekvencia diagram** | rovar\_mozgas\_sikertelen |

| **Név** | Gombafonal elágazása olyan tektonra, ahol van spóra |
| --- | --- |
| **Leírás, cél** | Amikor a gombász egy olyan tektonra szeretné fonalát elágaztatni, ahol található spóra, akkor még egy további tektonra is onnan el tudja ágaztatni a fonalát. |
| **Előfeltétel** | Kell lennie:  t1: tekton  t2: tekton  t3: tekton  f: gombafonál |
| **Hivatkozott kommunikációs diagram** | Diagram 3 |
| **Bemenet** | Tester hívja a fonál branchThread(t1) metódusát |
| **Forgatókönyv** | A t1 tekton mivel szomszédos a t2 tektonnal, amin megtalálható az f gombafonál már, így sikeresen elágazik f gombafonál t1 tektonra. Majd lekérdezzük t1 tekton spóráit, és ha található rajta, akkor a branchThread() metódus meghívható még egy alkalommal. |
| **Elvárt kimenet** | Sikeres fonal elágazás, azaz logikai igaz visszatérési érték. |
| **Hivatkozott szekvencia diagram** | Gombafonal elágazása |

* 1. ***A szkeleton kezelői felületének terve, dialógusok***

**5.2.1 Statikus naplózó**

A kezelői felület a következő módon épül fel:

* Ahány use-case, annyi menüpont lesz a program indításakor
* A felhasználó kiválaszthatja a menüpontok közül, hogy melyik teszt-szekvencia fusson.
* A kimenet a konzolon egy általunk definiált nyelv alapján íródik ki, amely bemutatja a szekvencia hívásait, a hívási paramétereit és a szekvenciában résztvevő függvények visszatérési értékeit.
* Ez a nyelv PlantUML alapú.

A nyelv felépítése:

* **->** : metódushívás
* **- - >**: visszatérési érték
* A nyíl bal oldalán áll az az objektum, amelyik a hívó
* A nyíl jobb oldalán áll az az objektum, amelyiken a függvény meghívódik
* **:** után függvény neve metódushívás esetén (->)
* **:** után visszatérési érték visszatérés esetén (- - >)
* **participant** : a szekvencia során résztvevő objektum
  + **“ “** között az objetum neve és típusa
  + **as** után referencia erre az objektumra

**Példa**:

**Gombafonal elágazása olyan tektonra, ahol van spóra** use-case kiválasztása esetén a a konzolos kimenet:

participant "f :FungalThread" as f

participant "t1 :MultiThreadTecton" as t1

participant "t2 :MultiThreadTecton" as t2

participant "t3 :MultiThreadTecton" as t3

-> f : branchThread(t1)

f -> t1 : putThread(f)

t1 -> t2 : getThreads()

t2 --> t1 : f

t1 --> f : true

-> t1 : getSpores()

<-- t1 : spores

-> f : branchThread(t3)

f -> t3 : putThread(f)

t3 -> t1 : getThread()

t1 --> t3 : f

t3 --> f : true

<-- f : true

**5.2.2 Kérdező**

Abban az esetben, ha az adott use-case teszt esetben van állapot lekérdezés a Tester felé, amelyre a válasz Tester aktortól érkezik, akkor a program először a konzolon felteszi a kérdést (az use-case-től függő formában), és kéri a választ. Ezután írja ki a konzolra statikus naplózó a szekvencia lefutását a fent leírt formában.

**Példa:**

**(Fejletlen) gombatest spóra szórása olyan tektonra, amelyre tud** esetén a konzolos kimenet:

[Gomba állapota (e [evolved] / u [unevolved)?]

[Válasz] : u

participant "m: Mushroom" as m

participant "Tester" as t

participant "t3 :SingleThreadTecton" as t3

-> m : shootSpore(t3)

m -> t : getMushroomState()

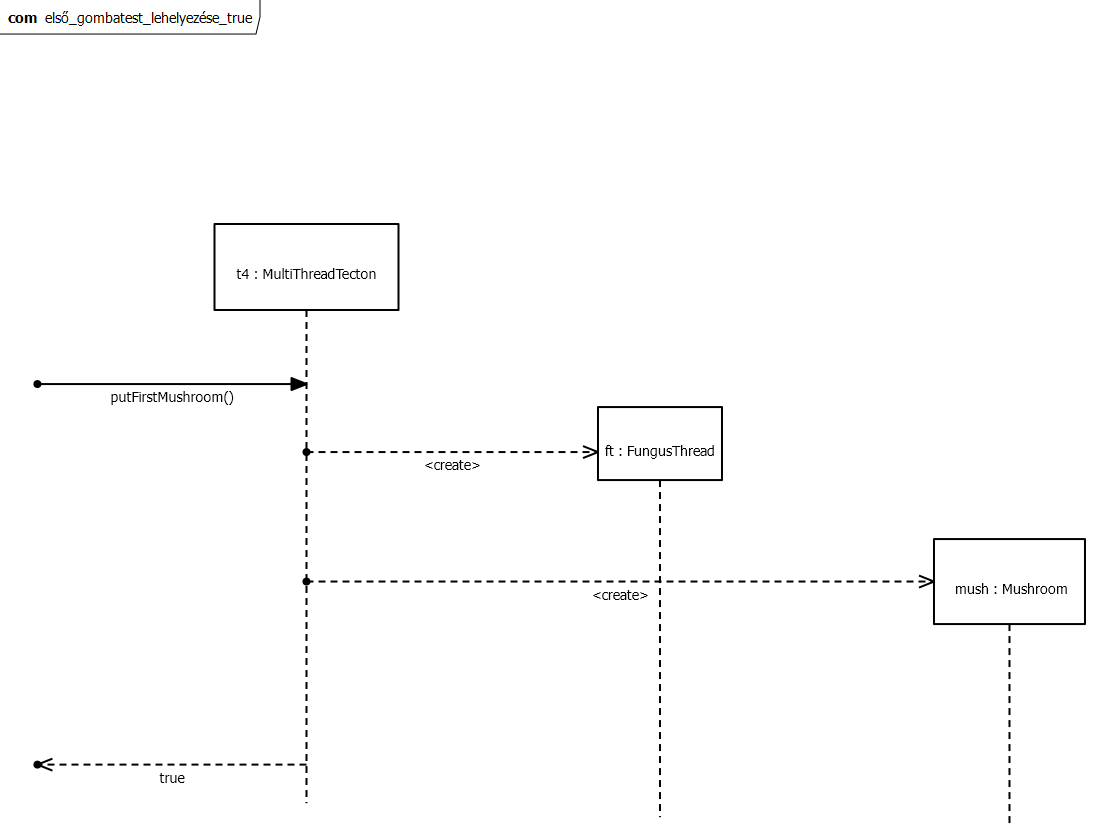
t --> m : unevolved

m -> t3 : putSpore(spores[0], position)

t3 --> m : true

<-- m : true

* 1. ***Szekvencia diagramok a belső működésre***
     1. ***első\_gombatest\_lehelyezése\_true***

******

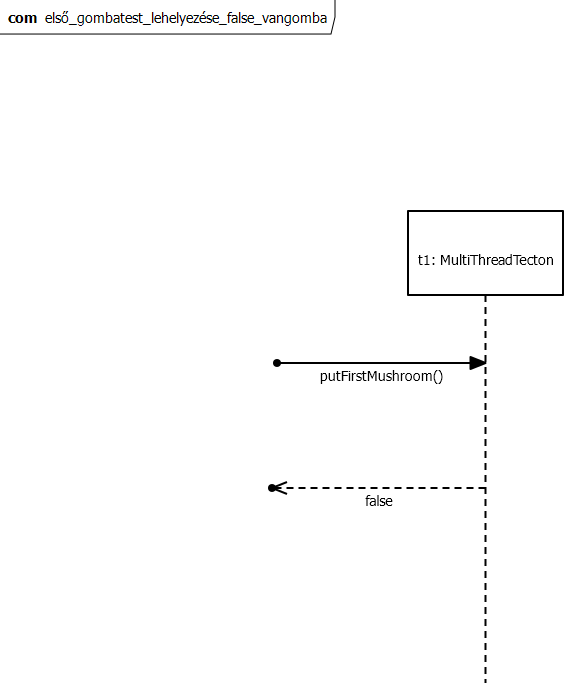
Tesztesetek amik felhasználják:

* Első gombatest lehelyezése olyan tektonra, amelyre le lehet helyezni
  + 1. **első\_gombatest\_lehelyezése\_false\_absorb**

****

Tesztesetek amik felhasználják:

* Első gombatest lehelyezése egy tektonra, amelyre nem lehet lehelyezni (AbsorbingTecton)
  + 1. **első\_gombatest\_lehelyezése\_false\_vangomba**

****

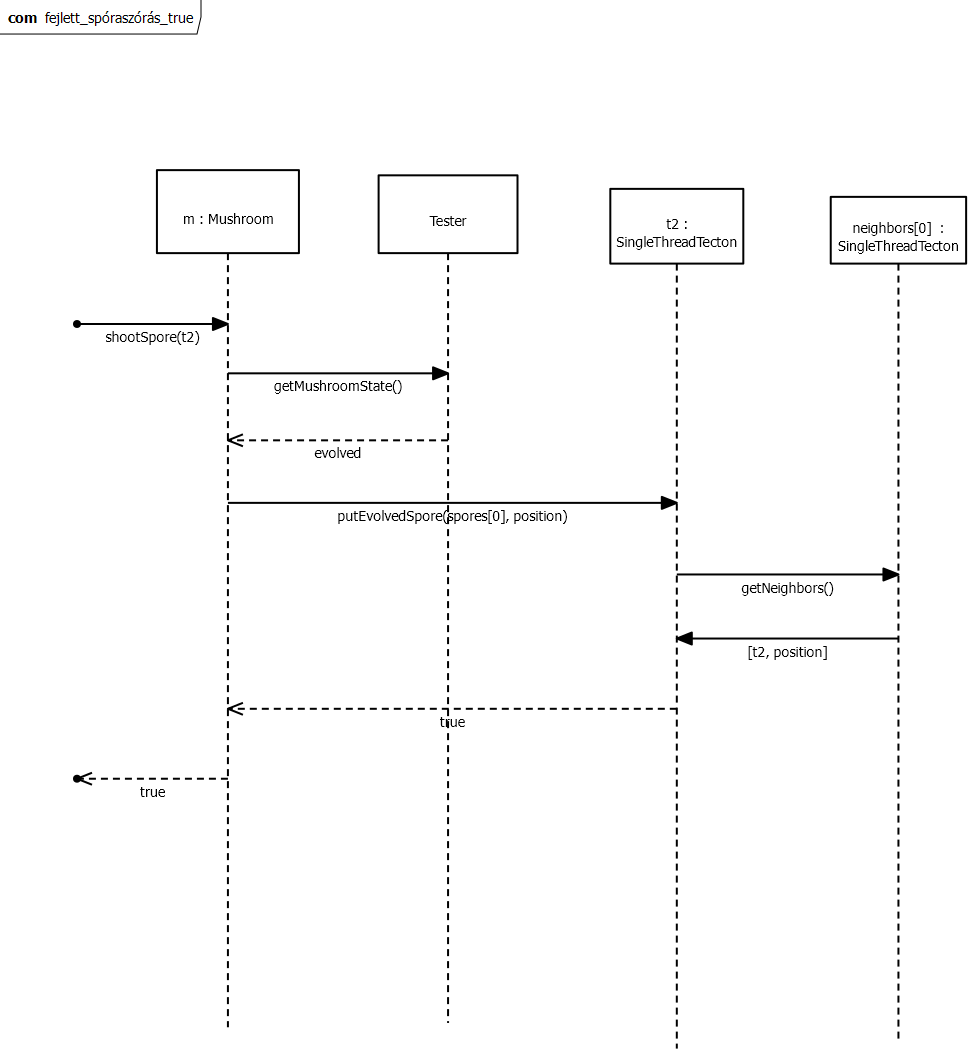
Tesztesetek amik felhasználják:

* Első gombatest lehelyezése egy tektonra, amelyre nem lehet lehelyezni (van gombatest a tektonon)
  + 1. ***fejletlen\_spóraszórás\_true***

******

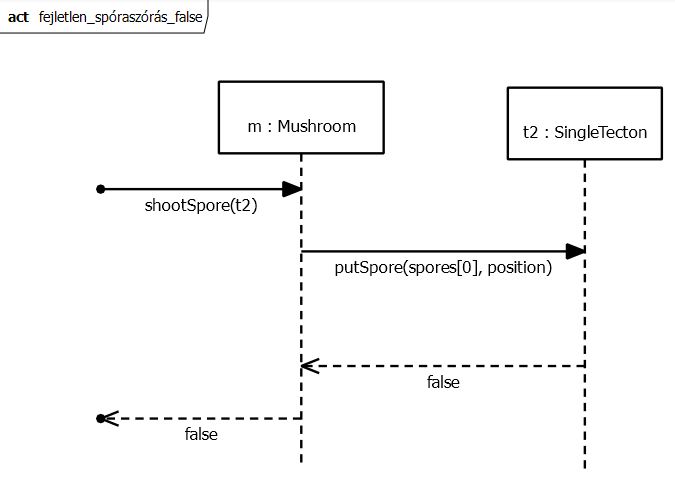
Tesztesetek amik felhasználják:

* Fejletlen gombatest spóra szórása olyan tektonra, amelyre tud.
  + 1. ***fejlett\_spóraszórás\_true***

******

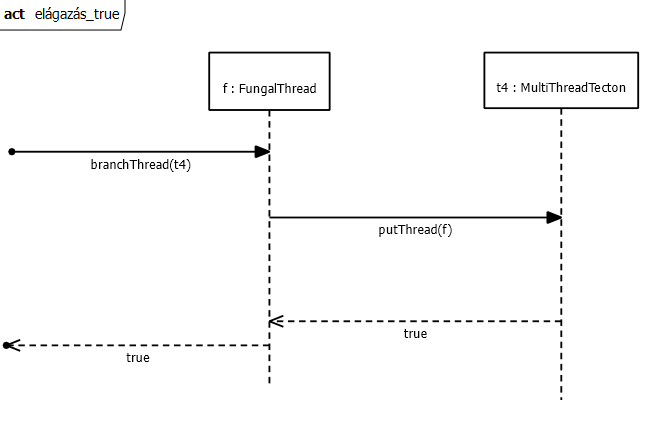
Tesztesetek amik felhasználják:

* Fejlett gombatest spóra szórása olyan tektonra, amelyre tud.
  + 1. ***fejletlen\_spóraszórás\_false***

**

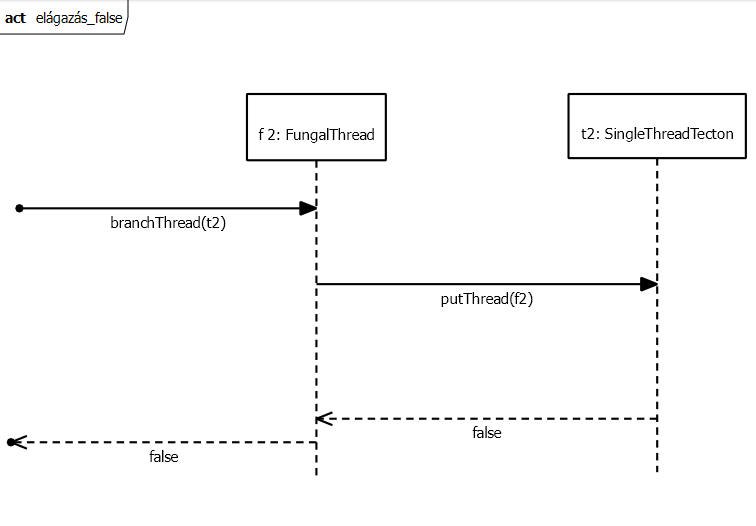
Tesztesetek amik felhasználják:

* Sikertelen spóraszórás, mert nem szomszédos a céltekton.
  + 1. ***elágazás\_true***

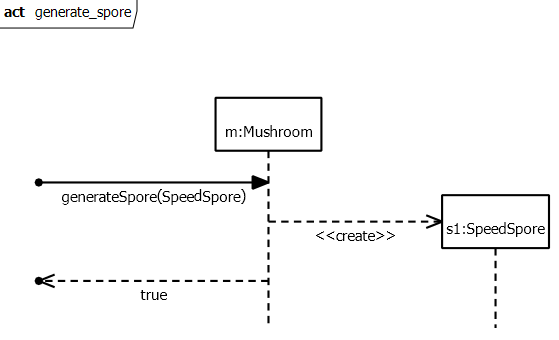


Tesztesetek amik felhasználják:

* Gombafonál sikeres elágaztatása tektonra.
  + 1. ***elágazás\_false***



Tesztesetek amik felhasználják:

* Gombafonál sikertelen elágaztatása egyfonalas tektonra, mert már van rajta.
  + 1. ***generate\_spore***

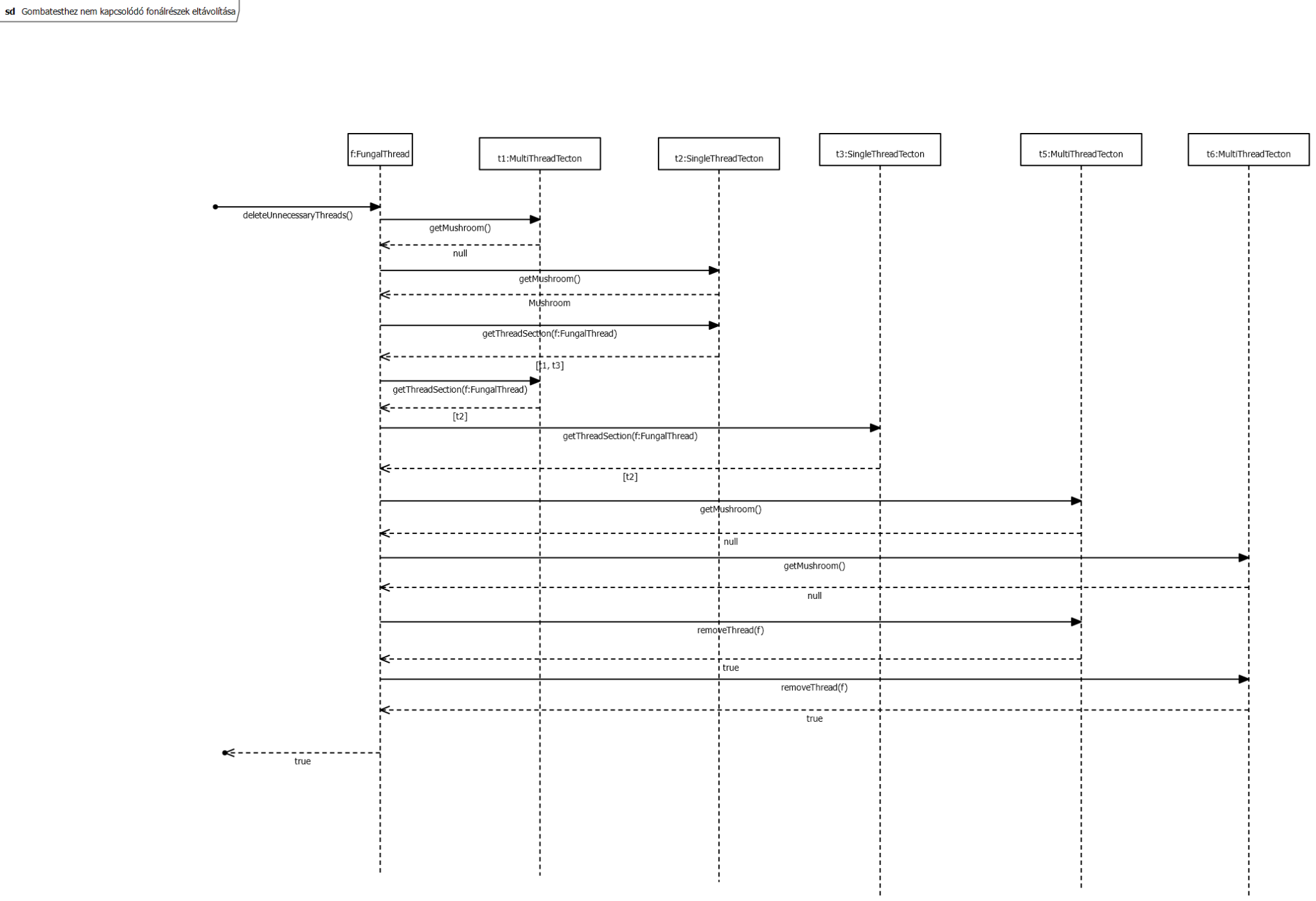
Tesztesetek amik felhasználják:

* Új spóra termelődése.
  + 1. ***gombatest\_elpusztulása***



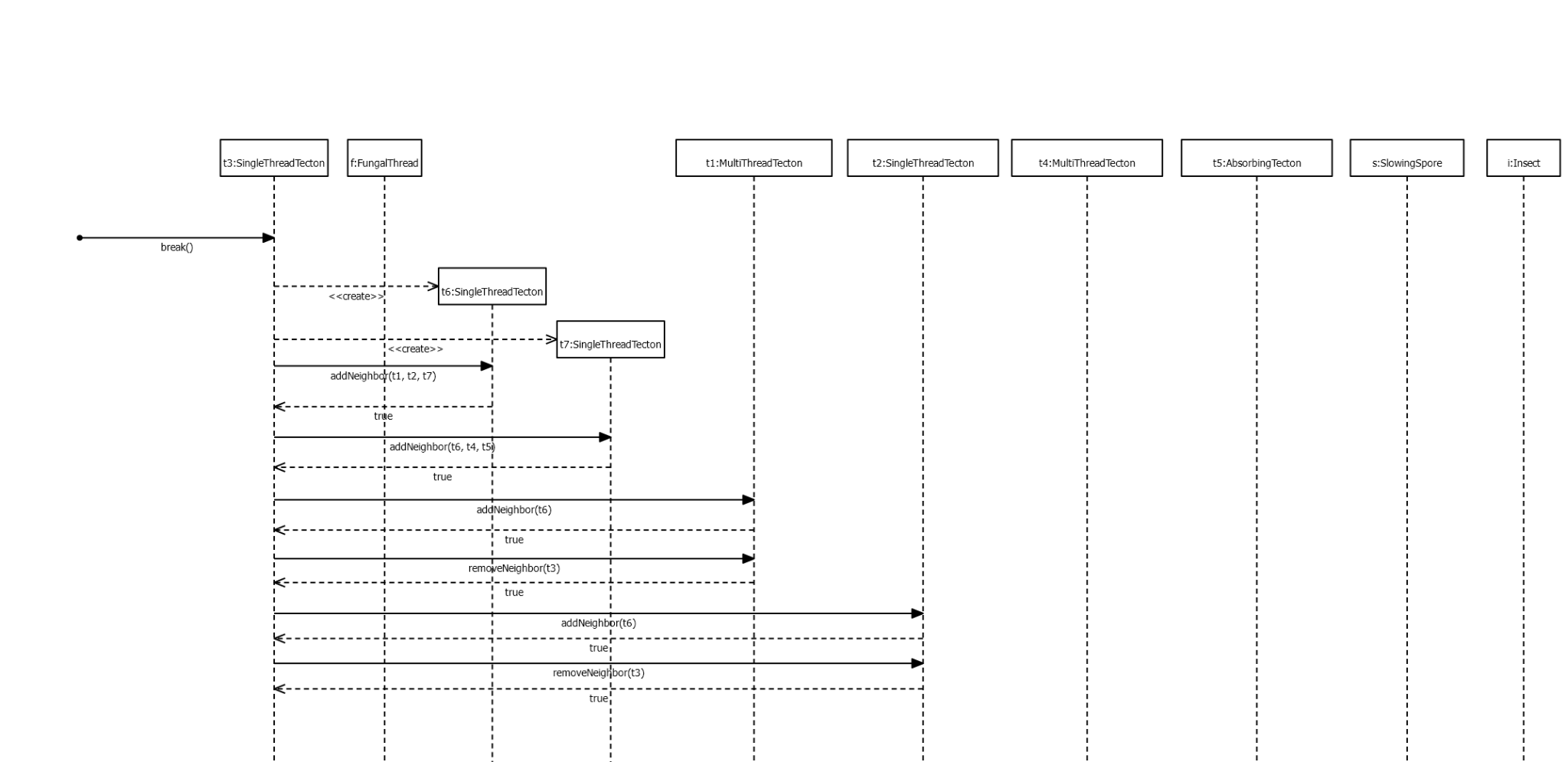
Tesztesetek, amik felhasználják:

* Gombatest elpusztulása
  + 1. ***gombatesthez\_nem\_kapcsolódó\_fonálrész\_eltávolítása***

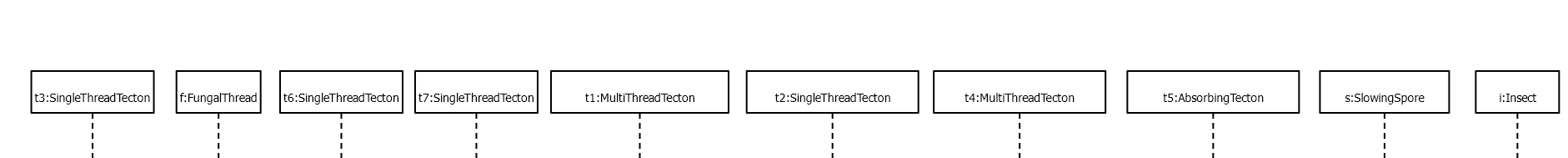
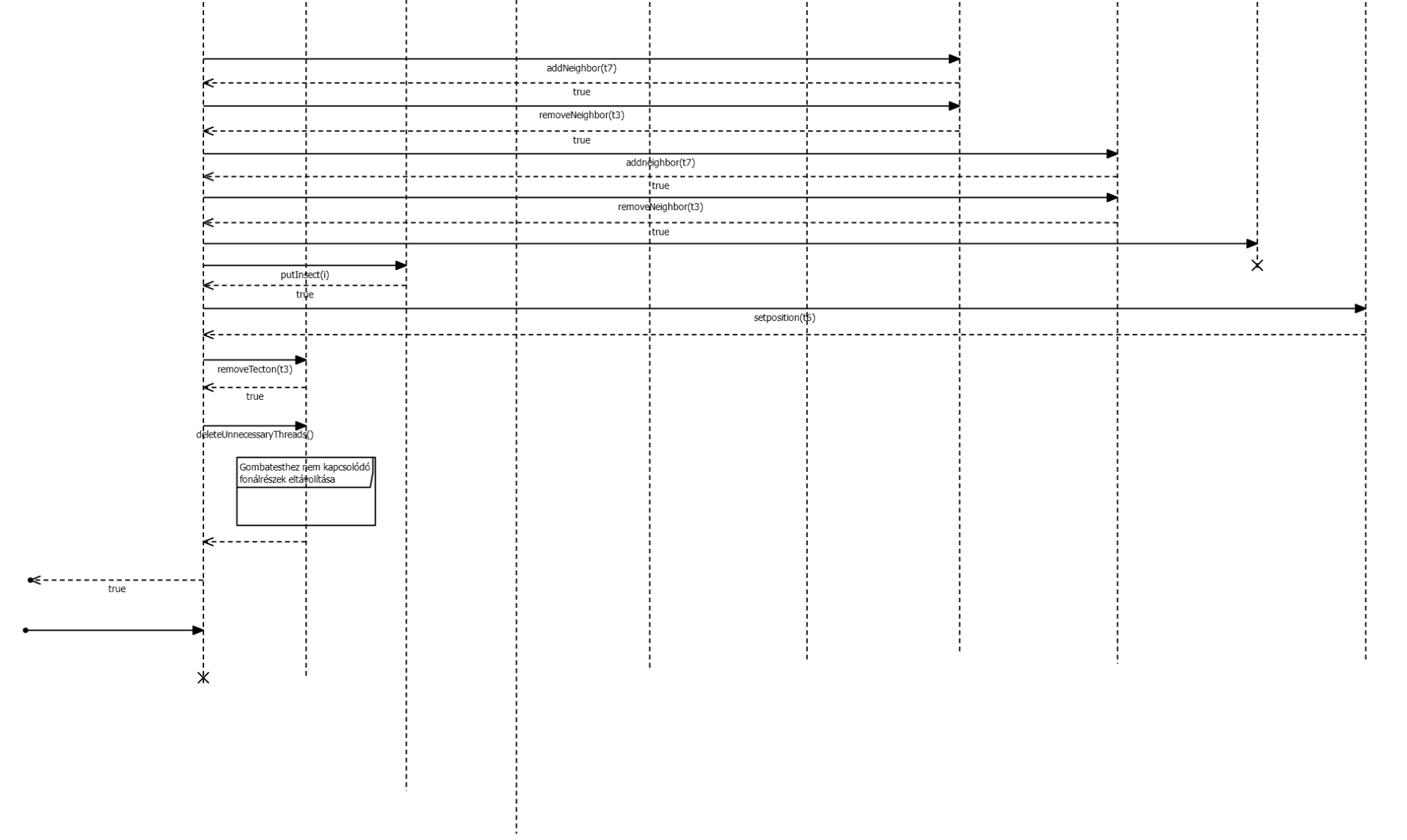


Tesztesetek, amik felhasználják:

* Gombatesthez nem kapcsolódó fonálrész eltávolítása
  + 1. ***tekton\_kettétörése***

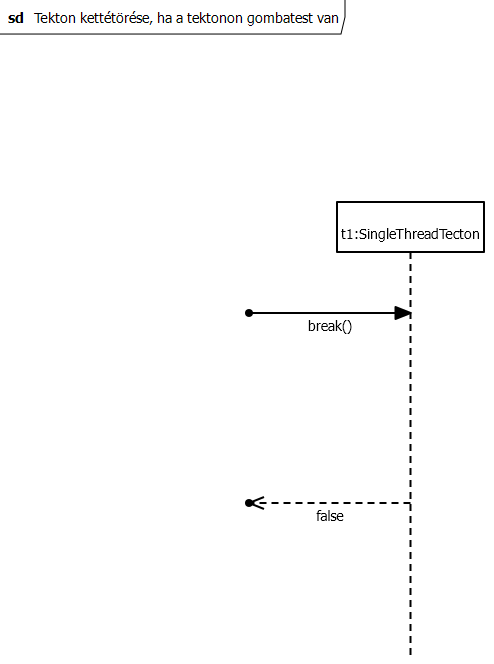


**Folytatás:**



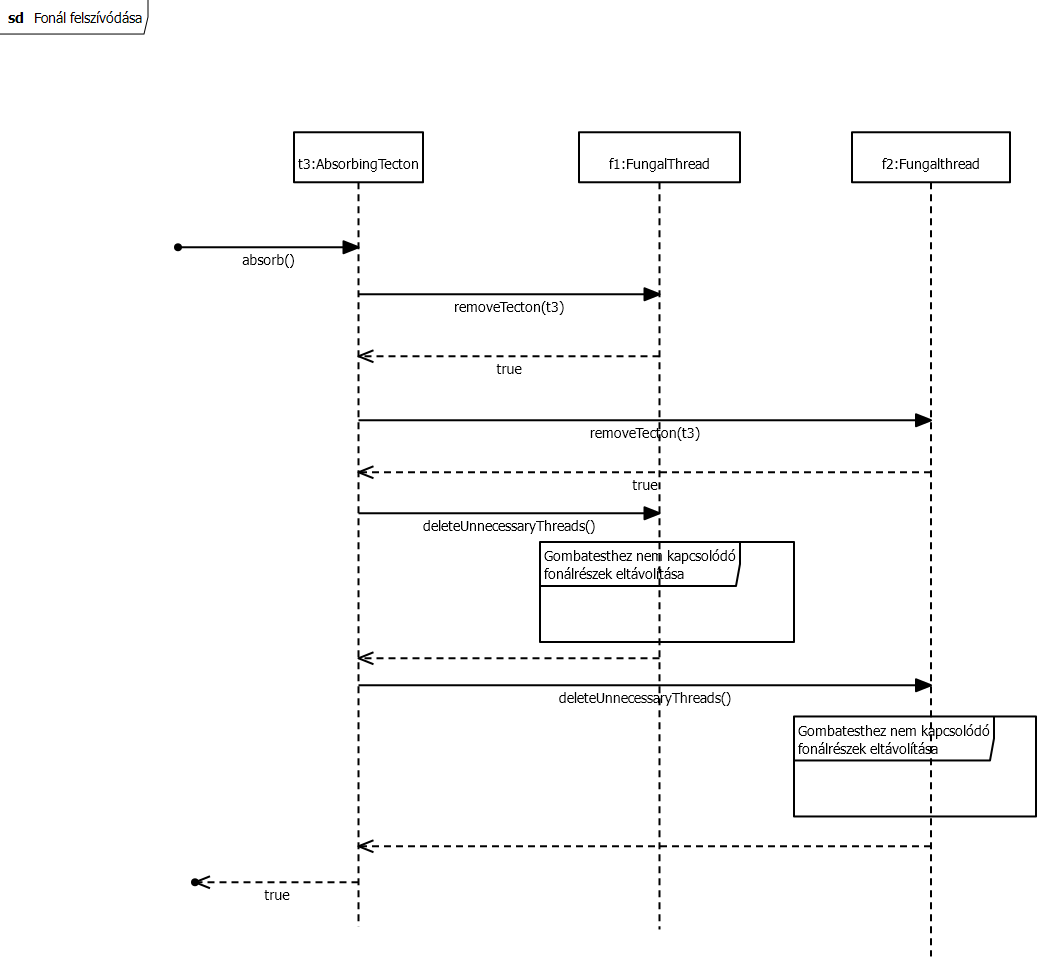
Tesztesetek, amik felhasználják:

* Tekton kettétörése
  + 1. ***tekton\_kettétörése\_ha\_a\_tektonon\_gombatest\_van***

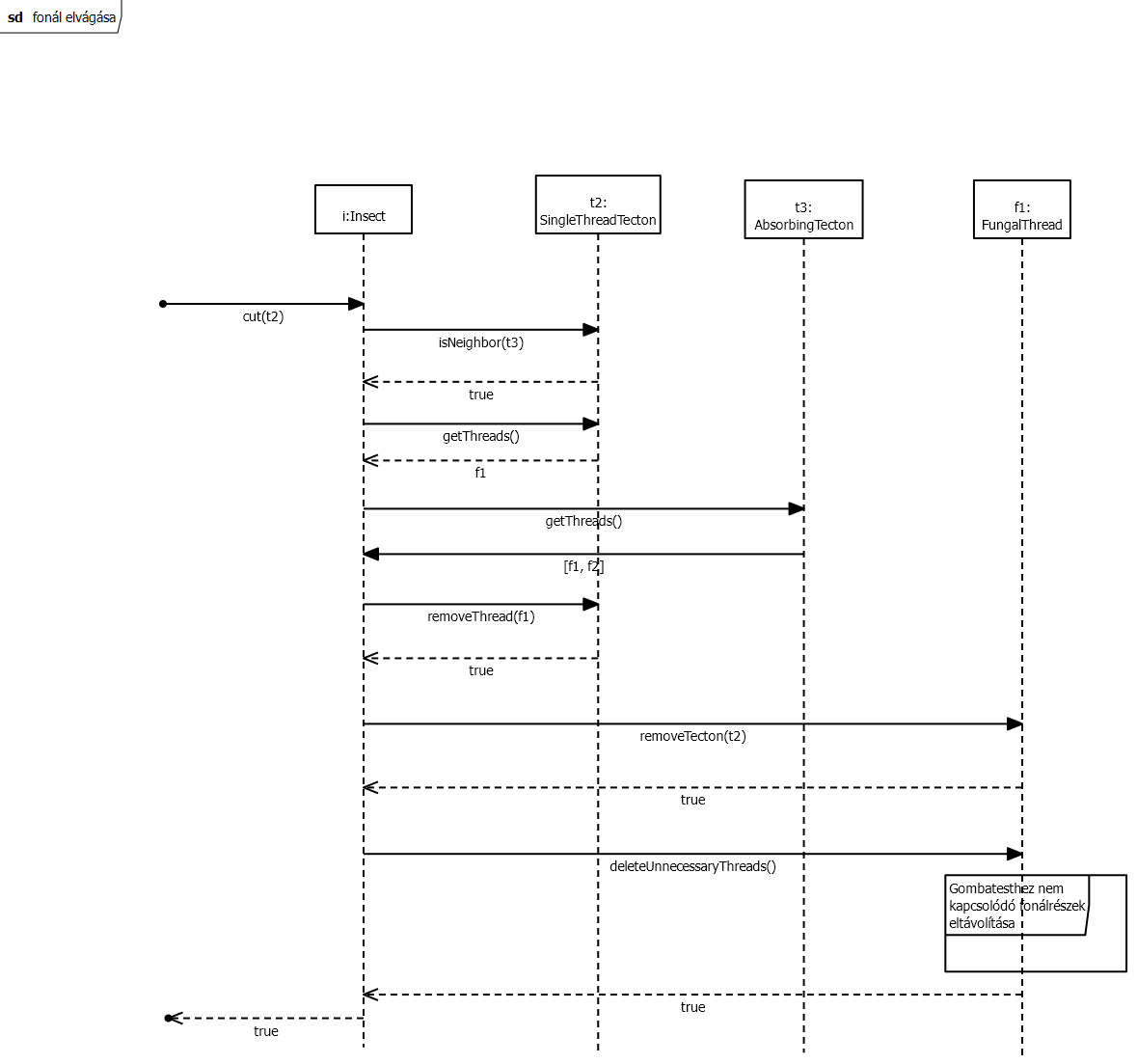


Tesztesetek, amik felhasználják:

* Tekton kettétörése, ha a tektonon gombatest van
  + 1. ***fonál\_felszívódása\_tektonról***



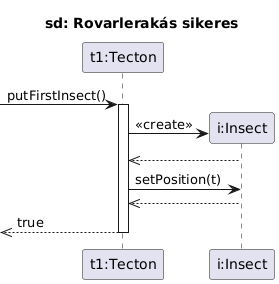
Tesztesetek, amik felhasználják:

* Fonál felszívódása tektonról
  + 1. ***gombafonál\_elvágása***

Tesztesetek, amik felhasználják:

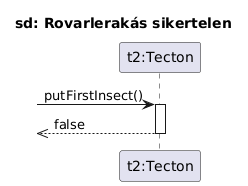
* Gombafonál elvágása

***5.3.16 rovarlerakás\_sikeres***

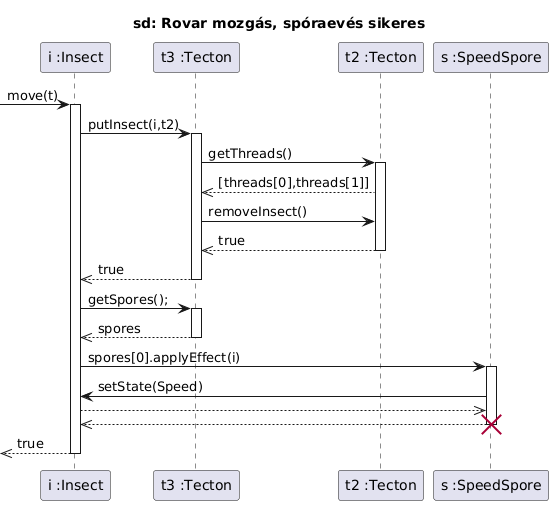
Tesztesetek, amik felhasználják:

* Rovar lehelyezése tektonra sikeres

***5.3.17 rovarlerakás\_sikertelen***

Tesztesetek, amik felhasználják:

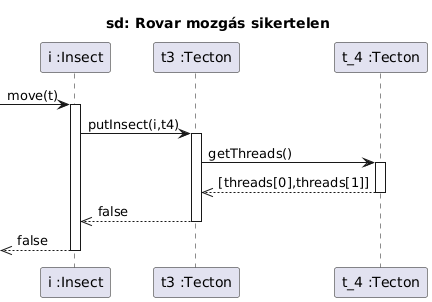
* Rovar lehelyezése tektonra sikertelen

***5.3.18 rovar\_mozgas\_sporaeves\_sikeres***

Tesztesetek, amik felhasználják:

* Rovar mozgása és spóraevése

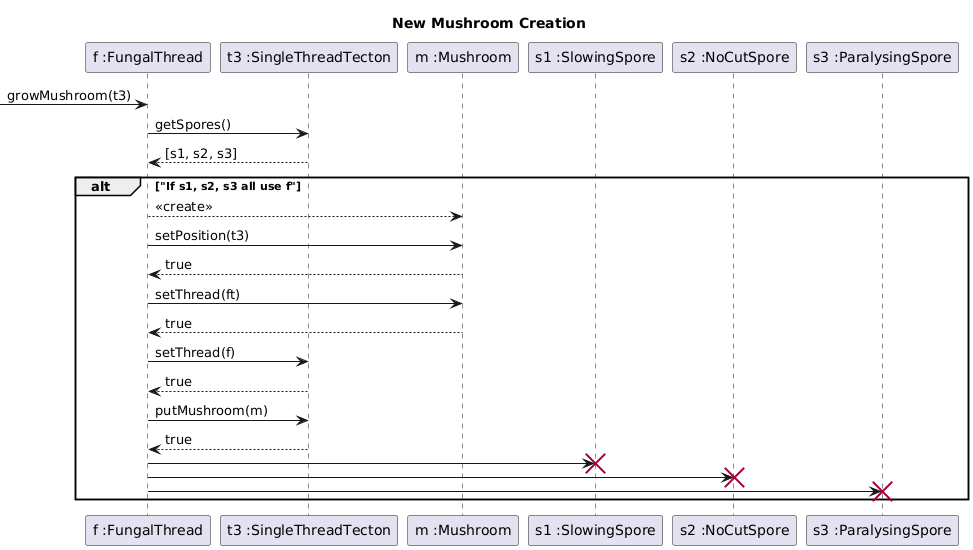
***5.3.19 rovar\_mozgas\_sikertelen***



Tesztesetek, amik felhasználják:

* Rovar mozgása sikertelen

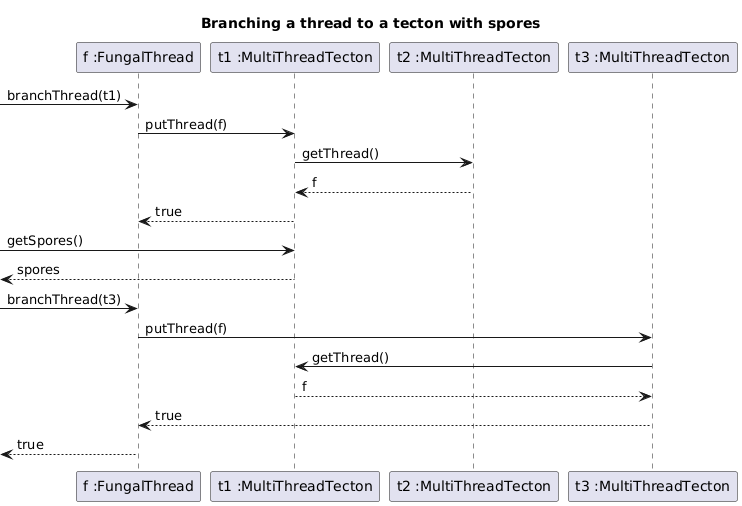
***5.3.20 Új gombatest lerakása***

******

Tesztesetek, amik felhasználják:

* Új gombatest lerakása

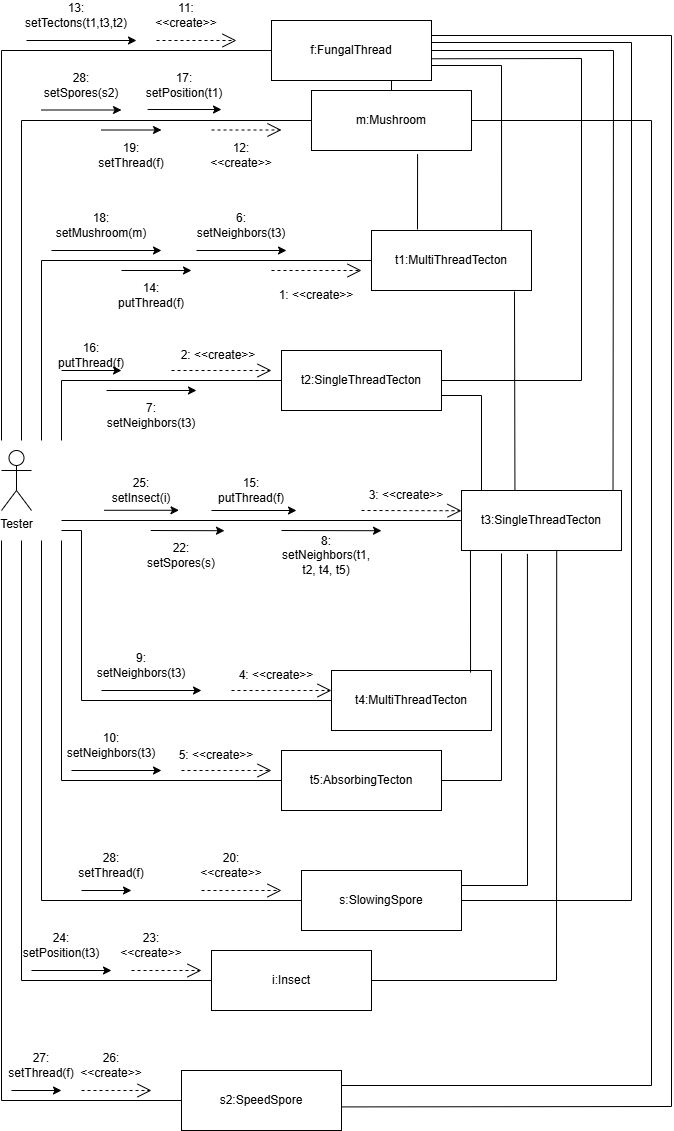
***5.3.21 Gombafonal elágazása***

******

Tesztesetek, amik felhasználják:

* Gombafonal elágazása olyan tektonra, ahol van spóra
  1. ***Kommunikációs diagramok***

**Diagram 1**

****

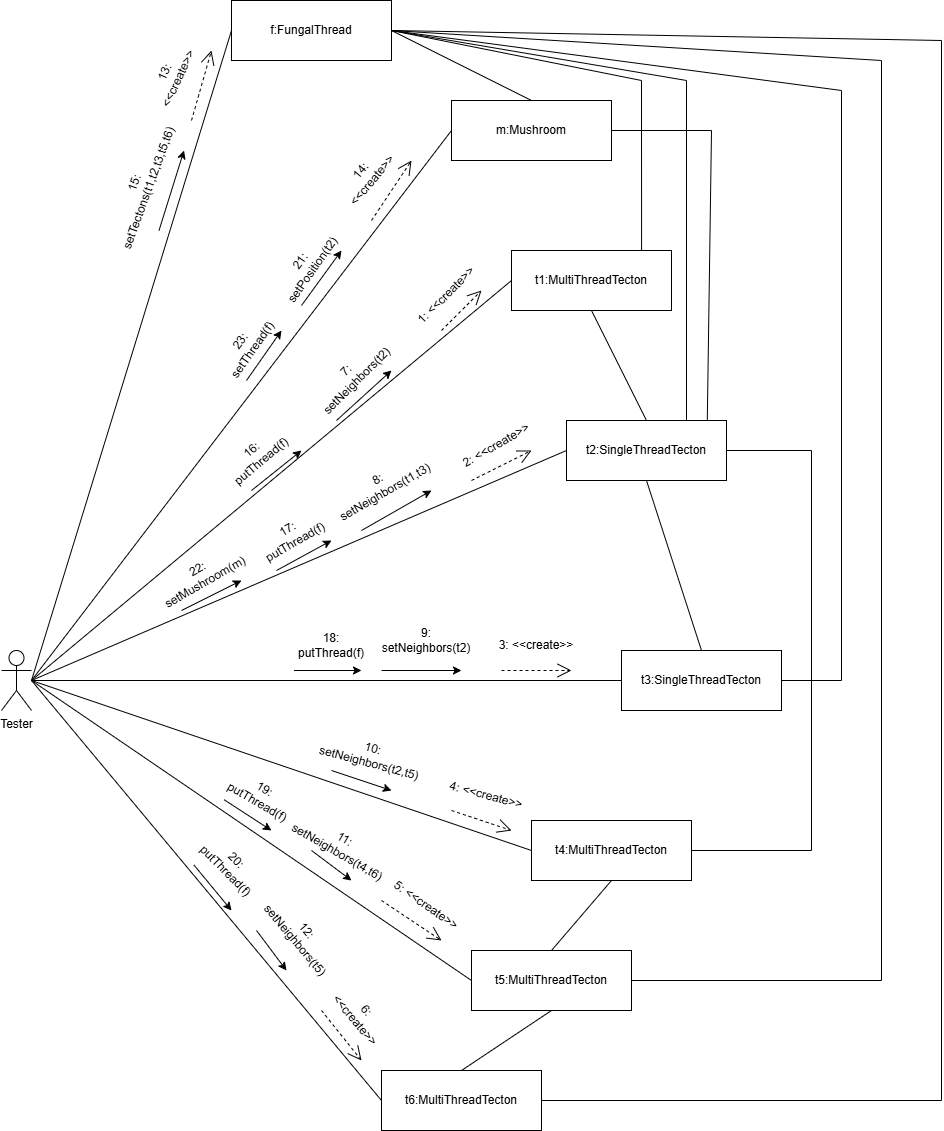
A Tester létrehozze a t1, t2, t3 ,t4 és t5 tektonokat. A t3 tekton szomszédos az összes többivel, a t1, t2, t4, t5 szomszédos a t3-al. A szomszédságokat a setNeighbors függvény segítségével állítja be.

Ezután a Tester létrehoz egy gombafonalat, majd egy gombát, beállítja, hogy a létrehozott f gombafonál a t1, t2 és t3 tektonokon ágazzon el, a gombát a t1 tektonra helyezi le, a gombának az f gombafonalat adja át, azaz ez lesz a gomba gombafonala.

A Tester létrehoz egy s gyorsítóspórát, melyet a t3 tektonra helyez le, és a gombafonál attribútumát beállítja f-re, azaz csak az f gombafonál tudja majd felhasználni újabb gombatest növesztésére.

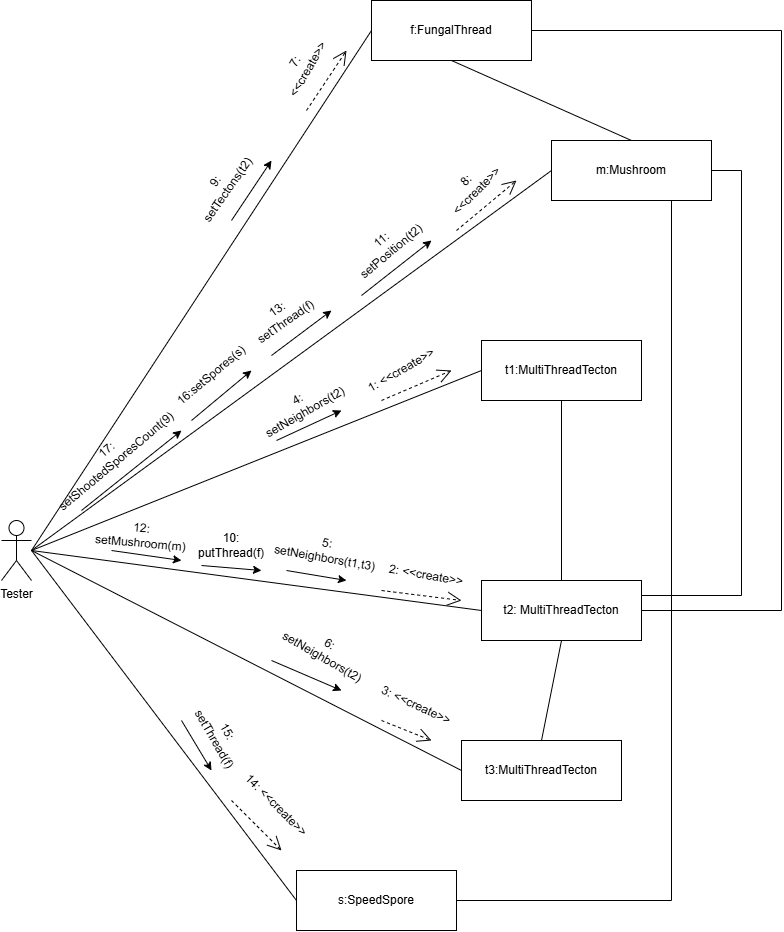
Ezután a Tester létrehozza az i rovart, ezt a t2 tektonra helyezi le.

A Tester létrehoz egy s2 gyorsítóspórát, melynek a gombafonál attribútumát beállítja f-re, azaz csak az f gombafonál tudja majd felhasználni újabb gombatest növesztésére. Ezt a spórát odaadja az m gombatestnek, azaz beleteszi a spores tömbjébe.

**Diagram 2**

A Tester létrehozza a t1, t2, t3, t4, t5 és t6 tektonokat. A t1 tekton szomszédos a t2-vel, a t2 szomszédos a t1-el és t3-al, a t3 szomszédos a t2-vel, a t4 szomszédos a t2-vel és t5-el, a t5 szomszédos a t4-el és t6-al, a t6 szomszédos a t5-el.

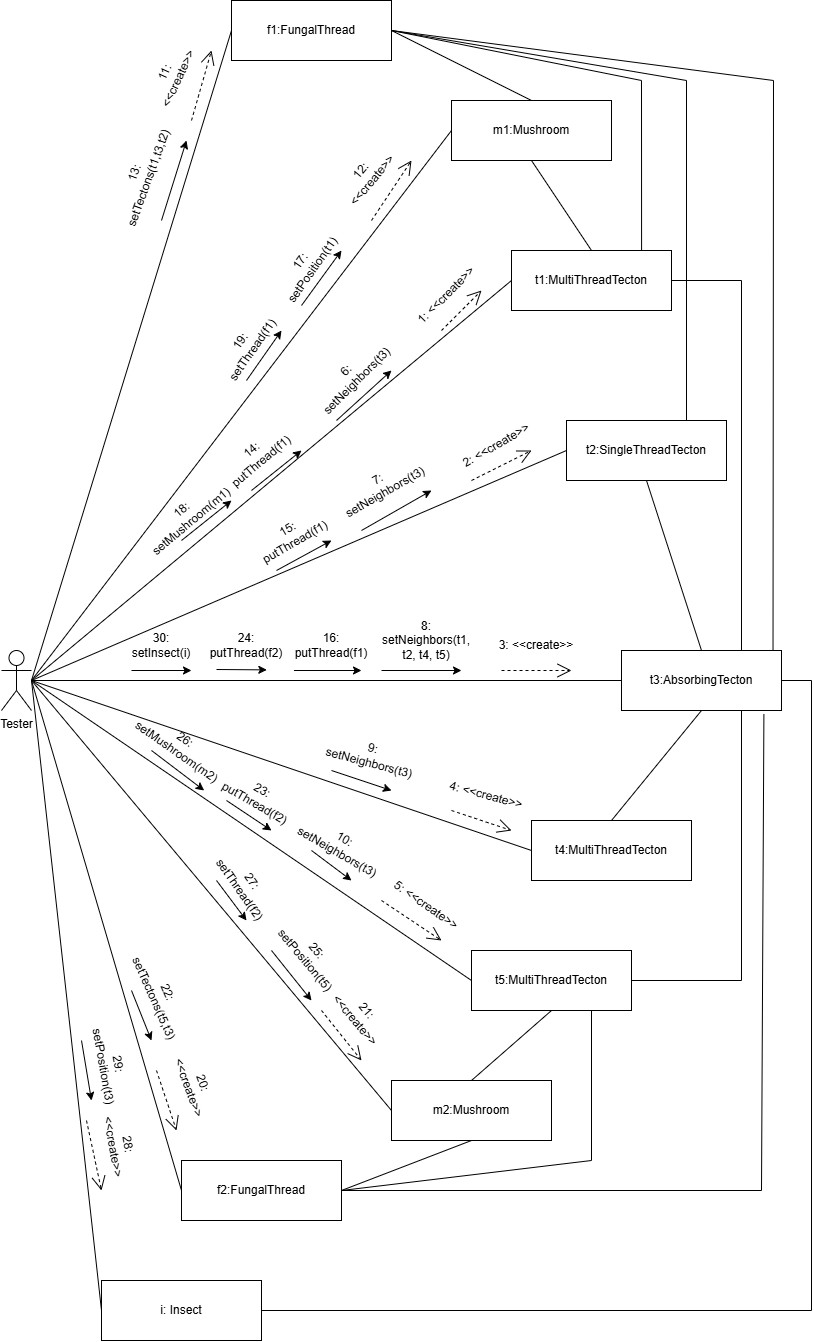
Ezután létrehoz egy gombafonalat, meg egy gombatestet. A gombafonál a t1, t2, t3, t5 és t6 tektonokon ágazik el, a gombatestet a t2 tektonra helyezi le. A gombának az f gombafonalat adja át, azaz ez lesz a gomba gombafonala.

**Diagram 3**

A Tester létrehozza a t1,t2 és t3 tektonokat, a t1 szomszédos a t2-vel, a t2 szomszédos a t1-el és t3-al, a t3 szomszédos a t2-vel.

Ezután létrehozza az m gombatestet és f gombafonalat. A gombafonalat a t2 tektonon ágaztatja el, a gombatestet is a t2 tektonra helyezi.

A Tester létrehoz egy s gyorsítóspórát, melynek a gombafonál attribútumát beállítja f-re, azaz csak az f gombafonál tudja majd felhasználni újabb gombatest növesztésére. Ezt a spórát odaadja az m gombatestnek, azaz beleteszi a spores tömbjébe. A gombatest shootedSporesCount attribútumát 9-re állítja, azaz a gombatest már kilőtt 9 spórát.

**Diagram 4**

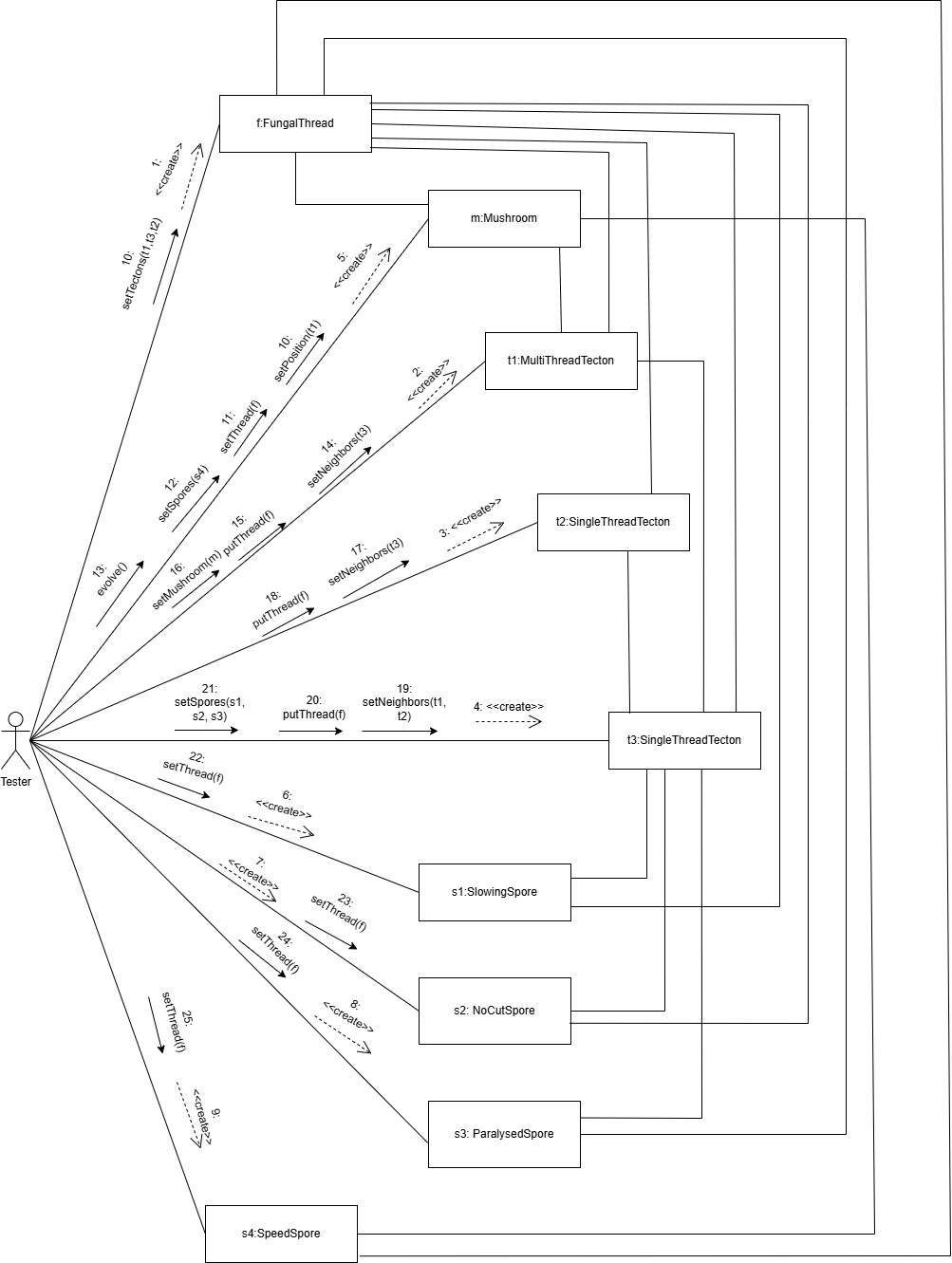
A Tester létrehozze a t1, t2, t3 ,t4 és t5 tektonokat. A t3 tekton szomszédos az összes többivel, a t1, t2, t4, t5 szomszédos a t3-al. A szomszédságokat a setNeighbors függvény segítségével állítja be.

Ezután a Tester létrehoz egy f1 gombafonalat, majd egy m1 gombát, beállítja, hogy a létrehozott f1 gombafonál a t1, t2 és t3 tektonokon ágazzon el, a gombát a t1 tektonra helyezi le, a gombának az f gombafonalat adja át, azaz ez lesz a gomba gombafonala.

Létrehoz egy f2 gombafonalat, majd egy m2 gombát is, beállítja, hogy a létrehozott f2 gombafonál a t5 és t4 tektonokon ágazzon el, a gombát a t5 tektonra helyezi le, a gombának az f2 gombafonalat adja át, azaz ez lesz a gomba gombafonala.

A Tester létrehoz egy i rovart, amelyet a t3 tektonra helyez le.

**Diagram 5**

****

A Tester létrehozza az f fonalat, az m gombatestet, a t1 MultiThreadTecton-t, a t2 SingleThreadTecton-t, a t3 SingleThreadTecton-t, az s1 SlowingSpore-t, az s2 NoCutSpore-t, az s3 ParalysedSpore-t és végül az s4 SpeedSpore-t.

Az f fonálon beállítja a Tester, hogy a t1, t2, t3 tektonokon ágazzon el.

Az m Mushroomon beállítja:

* A t1-en legyen
* Az f fonálhoz tartozzon
* A termelt spórái közé felveszi az s4-et
* Fejlett gombává teszi (evolve)

A t1-en beállítja:

* t3 szomszédja legyen
* f fonál elágazzon rajta
* m gombatest rajta legyen

A t2-őn beállítja:

* t3 szomszédja legyen
* f fonál elágazzon rajta

A t3-on beállítja:

* t1, t2 is szomszédja legyen
* f fonál elágazzon rajta
* Az s1, s2 és s3 spórák ki legyenek lőve rá

Spórákon beállítja (s1, s2, s3 és s4):

* Az f fonálhoz tartozzanak
  1. ***Napló***

| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| --- | --- | --- | --- |
| 2025.03.13. 20.30 | 2 óra | Lekli, Fejes, Zelch, Szolár, Toronyi | Meeting, feladat átbeszélése. Döntés: Mindenki a saját szekvencia diagrammjának megfelelő teszteseteket készíti el. |
| 2025.03.15. 12:00 | 8 óra | Zelch | A kommunikációs diagramok közül: diagram 1, diagram 2, diagram 3, diagram 4 megcsinálása, ezután a következő tesztesetek és szekvencia diagramok megcsinálása: Gombatest elpusztulása  ,Gombatesthez nem kapcsolódó fonálrész eltávolítása,Tekton kettétörése, Tekton kettétörése, ha a tektonon gombatest van, Fonál felszívódása tektonról, Gombafonál elvágása |
| 2025.03.15. 11:00 | 7,5 óra | Szolár | * Sikertelen spóraszórás, mert nem szomszédos a céltekton. * Gombafonál sikeres elágaztatása tektonra. * Gombafonál sikertelen elágaztatása egyfonalas tektonra, mert már van rajta. * Új spóra termelődése.   Tesztesetek megalkotása.  Diagram 1 refaktorálása. |
| 2025. 03. 15. 18:00 | 3 óra | Fejes | Use case leírások és a hozzá tartozó szekvencia diagram elkészítése:   * első\_gombatest\_lehelyezése\_true * első\_gombatest\_lehelyezése\_false\_absorb * első\_gombatest\_lehelyezése\_false\_vangomba |
| 2025. 03. 16. 10:00 | 4 óra | Fejes | Use case leírások és a hozzá tartozó szekvencia diagram elkészítése:   * fejletlen\_spóraszórás\_true * fejlett\_spóraszórás\_true   Diagram 5 kommunikációs diagram elkészítése |
| 2025.03.16. 14:00 | 4 óra | Toronyi | Tesztesetek megtervezése, azok leírása, megfelelő szekvencia diagramok átdolgozása. |
| 2025.03.16. 15:00 | 3 óra | Lekli | Tesztesetek megtervezése, szekvenciadiagramok javítása. |
| 2025. 03. 17. 10:00 | 1,5 óra | Fejes, Toronyi | Meeting: szkeleton kezelői felület tervének átbeszélése, pontos leírás elkészítése. |
| … | … | … | … |