**7. Prototípus koncepciója**

26 – five\_guys

Konzulens:

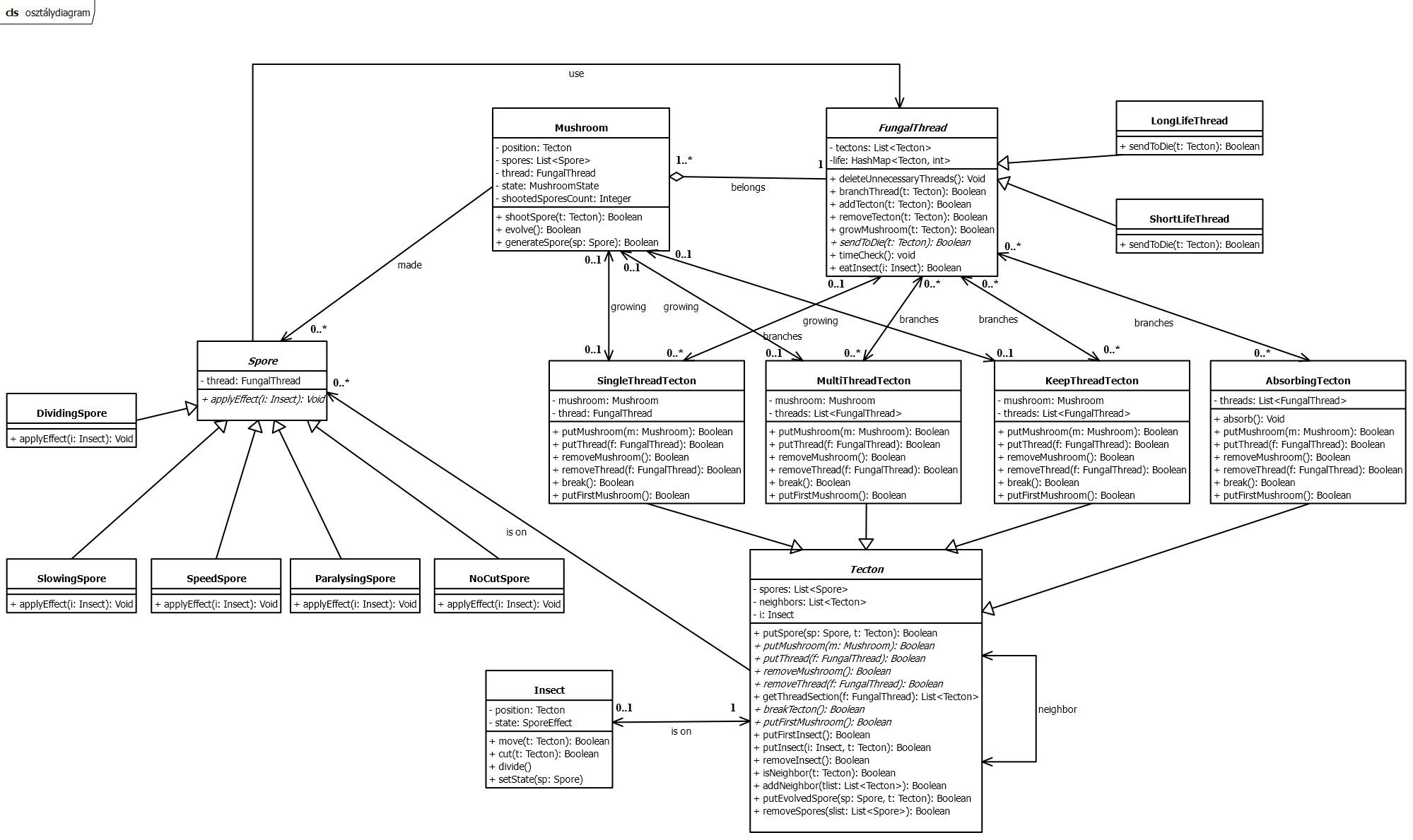
Huszerl Gábor

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.

**Prototípus koncepciója**

* 1. **Változás hatása a modellre**
     1. **Módosult osztálydiagram** ****

#### 

* + 1. **Új vagy megváltozó metódusok**

***Új osztályok:***

* + - 1. **DividingSpore(Rovarosztó spóra)**

#### Felelősség

Azokért a spórákért felelős, amelyek elfogyasztása után a rovart osztódásra készteti . Ekkor a keletkezik egy újabb rovar, amelynek a rovarásza megegyezik az eredeti rovar rovarászával.

#### Ősosztályok

Spore (Spóra) -> *DividingSpore*

#### Interfészek

Nem valósít meg interfészt.

#### Asszociációk

Csak olyanok, amelyet az ősosztályától örököl.

#### Attribútumok

Csak olyanok, amelyet az ősosztályától örököl.

#### Metódusok

* + **void applyEffect(i: Insect)**: A metódus meghívja a rovar divide() függvényét
    - 1. **KeepThreadTecton(fonalat megtartó tekton)**

#### Felelősség

Ez a tektonfajta a MultiThreadTektonhoz hasonló, azonban életben tartja azokat a fonalakat, amelyek nincsenek közvetve vagy közvetlenül gombatesthez kötve

#### Ősosztályok

Tecton -> KeepThreadTecton

#### Interfészek

Nem valósít meg interfészt.

#### Asszociációk

* + **asszociáció(branches)**: Másik oldal:FungalThread. Mint a MultiThreadTecton-nál.
  + **asszociáció(growing)**: Másik oldal: Mushroom. Van egy Mushroom attribútuma.

#### Attribútumok

* + **threads : List<FungalThread>**. Az asszociációnak megfelelően, lehet rajta akár több fonál is.
  + **mushroom : Mushroom**. Az asszociációnak megfelelően, maximum egy gombatest lehet rajta.

#### Metódusok

* + **putThread(f : FungalThread) : boolean**. Mint a MultiThreadTecton-nál.
  + **putMushroom(m : Mushroom) : boolean**. Mint a MultiThreadTecton-nál.
  + **removeThread(f : FungalThread) : boolean.** Mint a MultiThreadTecton-nál.
  + **removeMushroom() : boolean.** Mint a MultiThreadTecton-nál.
  + **break(): boolean**: boolean: Tekton kettétörését valósítja meg, két KeepThreadTecton keletkezik
  + **putFirstMushroom(): boolean**: Játékos első gombatestjének lehelyezése
    - 1. **ShortLifeThread**

#### Felelősség

Felelőssége hasonló a FungalThread felelősségeihez, azonban fonálvágás esetén nem egyből pusztulnak el róla a fonalak, hanem csak a következő kör elején

#### Ősosztályok

FungalThread -> ShortLifeThread

#### Interfészek

Nem valósít meg interfészt.

#### Asszociációk

Csak olyanok, amelyet az ősosztályától örököl.

#### Attribútumok

Csak olyanok, amelyet az ősosztályától örököl.

#### Metódusok

* + **sendToDie(t: Tecton): boolean**: A metódus a kapott tektont, ha a fonál rajta van a tektonon, behelyezi a life<Tecton, int> HashMap-be kulcsként, értékként pedig egyet állít be. A gombafonálnak van egy timeCheck() metódusa, amely minden kör elején meghívódik, célja, hogyha a HashMap-ben van elem, akkor ezek értékét csökkentse mindig 1-el. Ha az érték 0 lesz, ekkor a tárolt tektonról leveszi a fonalat, majd az elemet kitörli a HashMap-ből
    - 1. **LongLifeThread**

#### Felelősség

Felelőssége hasonló a FungalThread felelősségeihez, azonban fonálvágás esetén nem egyből pusztulnak el róla a fonalak, hanem csak a következő kör utáni körben

#### Ősosztályok

FungalThread -> LongLifeThread

#### Interfészek

Nem valósít meg interfészt.

#### Asszociációk

Csak olyanok, amelyet az ősosztályától örököl.

#### Attribútumok

Csak olyanok, amelyet az ősosztályától örököl.

#### Metódusok

* + **sendToDie(t: Tecton): boolean**: A metódus hasonló, mint a ShortLifeThread-nél, azonban a HashMap-ben az elem értékét kettőre állítja be

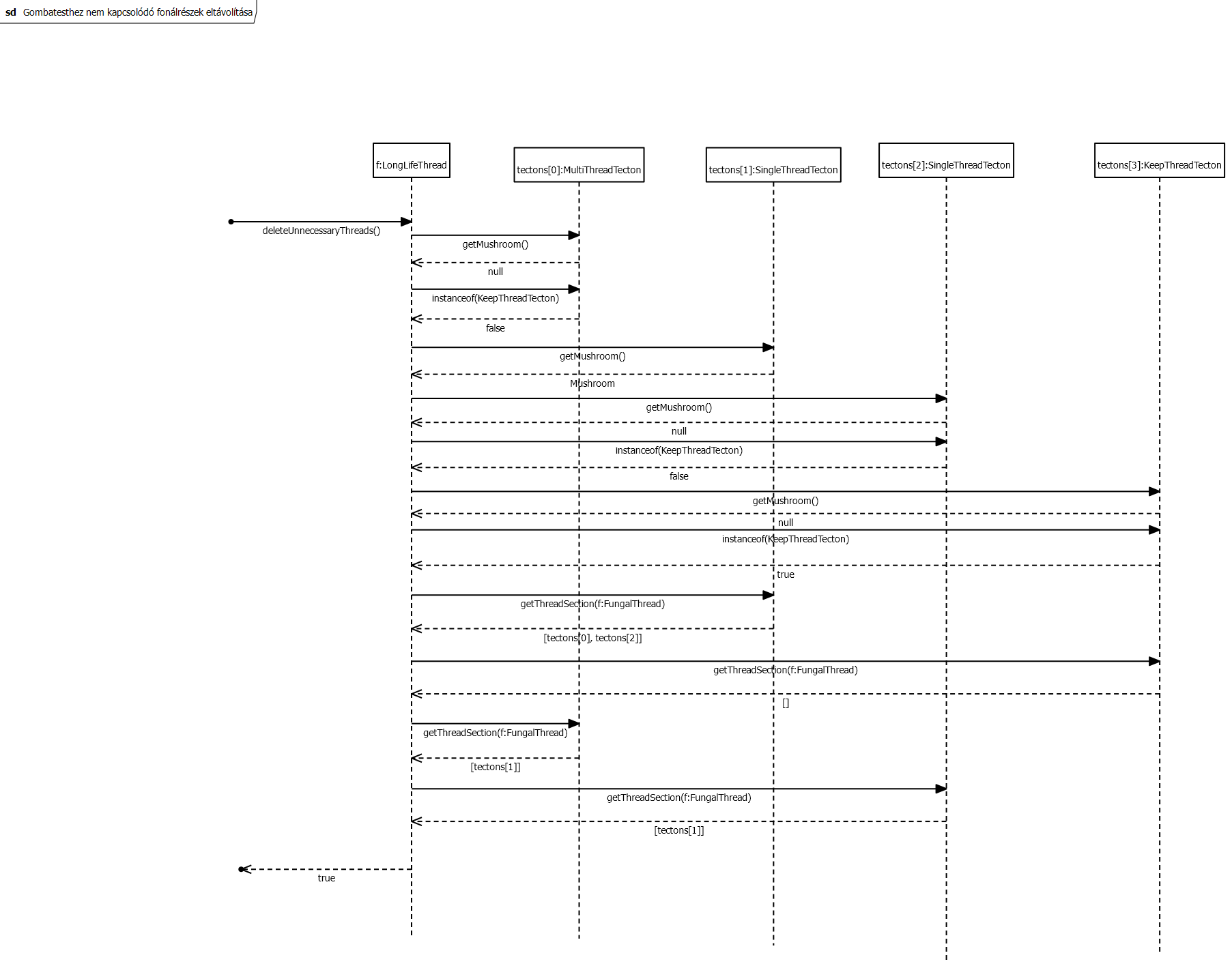
***Változások a már meglévő osztályokban***

Az **Insect** osztályban történt változtatások:

* az Insect osztálynak lett egy új divide() metódusa
  + **divide(): boolean**: A metódus egy új rovart hoz létre, melyet az eredeti rovar tektonjának az első szomszédjára helyez le, ha ez nem lehetséges, mert azon a tektonon már van rovar, tovább lépeget a szomszédain, ha egyetlen szomszédjára se tudta lehelyezni, sorban végigmegy a tektonokon és az első olyan tektonra helyezi le, amire ez lehetséges
* megváltozott a **cut(t: Tecton)** metódusa, amely a gombafonálnak a sendToDie(t: Tecton) metódusát fogja hívni és nem a removeTecton(t: Tecton) metódusát, valamint a tektonon, amelyen elvágta a fonalat nem hívja többet meg a removeThread(f: FungalThread) metódust

A **FungalThread** osztályban történt változtatások:

* az osztály absztrakt osztállyá vált
* lett egy új attribútuma
  + **life: HashMap<Tecton, int>**: ebben az attribútumban tároljuk el azokat a tektonokat, amelyeken a gombafonalak el kell pusztuljanak, mivel valamelyik bogarász elvágta, ez lesz a kulcs, értékként pedig azt tároljuk el, hogy hány kör múlva kell a fonalak elpusztuljanak a tetkonról
* új metódusokkal bővült
  + **sendToDie(t: Tecton): boolean** :absztrakt metódus
  + **timeCheck(): void**: a metódust minden kör elején a kontroller meghívja, és ha a time HashMap nem üres, akkor csökkenti 1-el a benne levő elemek értékét. Ha az érték elérte a 0-át, akkor a tárolt tektonról leveszi a fonalat meghívva a removeThread(f: FungalThread) metódusát, majd törli az elemet a HashMapből, illetve a gombafonál tectons listájából törli a tectont, és lefuttatja a deleteUnnecessaryThreads() metódust
  + **eatInsect(i: Insect):boolean**: leveszi a rovart a tektonról, majd törli a rovart. Abban az esetben, ha a rovar nem AbsorbingTectonon, vagy nem egy olyan tektonon helyezkedett el, ahol már volt gombatest, akkor növeszt egy új gombatestet
* változott a **deleteUnnecessaryThreads()** metódusa: a metódus során sorban végigmegy a tectons tömbjén, megnézi, hogy mely tektonokon van ugyanolyan fajból származó gombatest, ezeket eltárolja, majd ha nincs ilyen gombatest rajtuk, akkor megnézi, ha nem-e KeepThreadTecton típusúak, ha ez teljesül rájuk, akkor is eltárolja, végül megkeresi az összes olyan fonalat, amelyek elérhetők az elmentett tektonokról haladva. Ezek a fonálrészek lesznek azok a részek, amelyeket meg kell tartani, a többit pedig töröljük.
  + 1. **Szekvencia-diagramok**

**Gombatesthez nem kapcsolódó fonálrész eltávolítása**

A deleteUnnecessaryThread függvényhívás érkezhet egy tektontól, egy gombatesttől, vagy akár egy külső kontrollertől

Ezen a diagramon a következő objektumok találhatók:

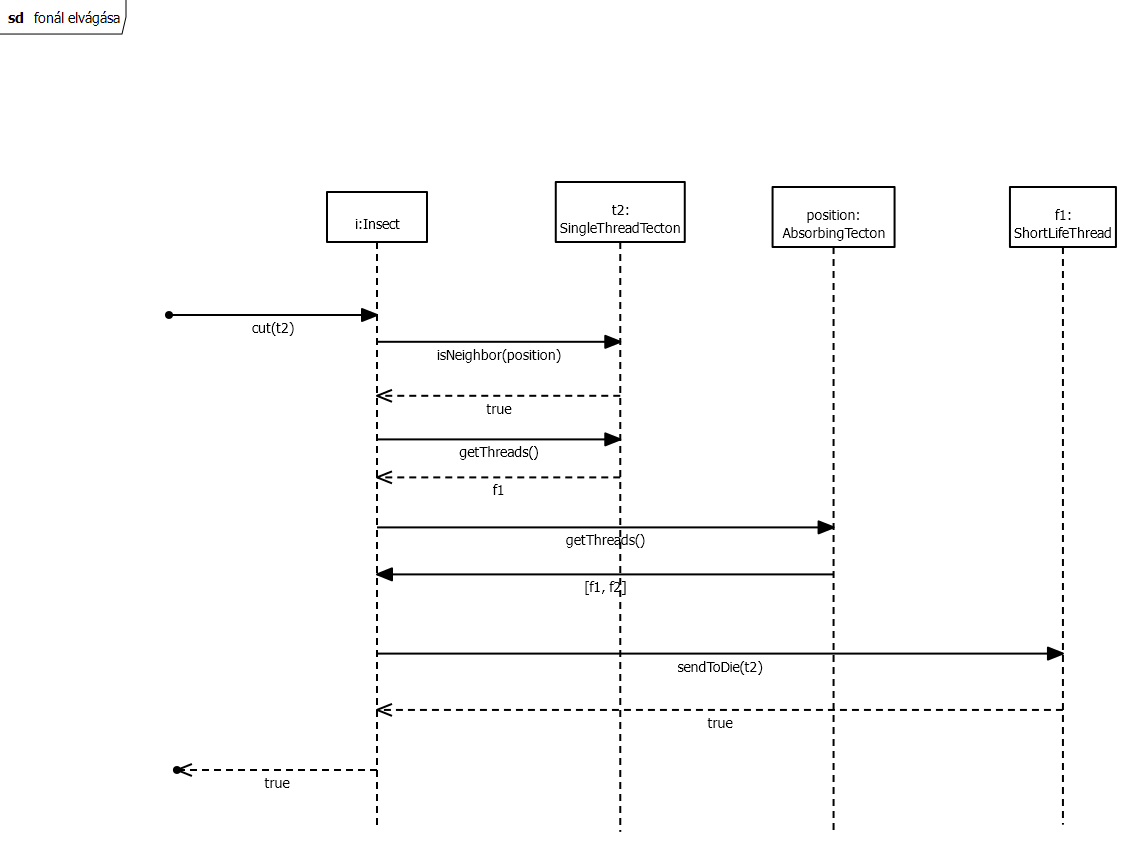
* f : gombafonál
* tectons[0] : tekton, amelyen az f gombafonál elágazik
* tectons[1] : tekton, amelyen az f gombafonál elágazik
* tectons[2] : tekton, amelyen az f gombafonál elágazik
* tectons[3] : tekton, amelyen az f gombafonál elágazik

A tectons az f gombafonál egy tömbje, amelyben azokat a tektonokat tárolja, amelyen elágazik.

A metódus sorban végigmegy a tectons tömb tektonjain, megnézi melyiken van ugyanabból a fajból származó gombatest, ezt eltárolja. Ha nincs rajta ilyen gombatest, megnézi, hogy KeepThreadTecton fajtájú-e, ha igen, ugyancsak eltárolja.

Miután ezt megcsinálta az eltárolt tektonokon végigmegy, és meghívja rájuk a getThreadSection(f: FungalThread) metódusukat, amely visszaadja azokat a szomszédos tektonokat, amelyeken elágazik az f gombafonál, és ezeket a tektonokat is eltárolja. A kapott tektonokon újra meghívja ezt a metódust, és addig folytatja, amig talál új tektont.

Mikor már nem talál új tektont, azt jelenti, hogy megtatált az összes olyan fonálrészt, amely közvetve vagy közvetlenül gombatesthez vagy KeepThreadTecton-hoz van kötve, tehát ezeken a gombafonalakat meg kell tartani, és azokon a tektonokon kell törölni, amelyek nincsenek a megtaláltak között.

**Gombafonál elvágása**

Egy külső kontroller meghívja az i rovar cut metódusát.

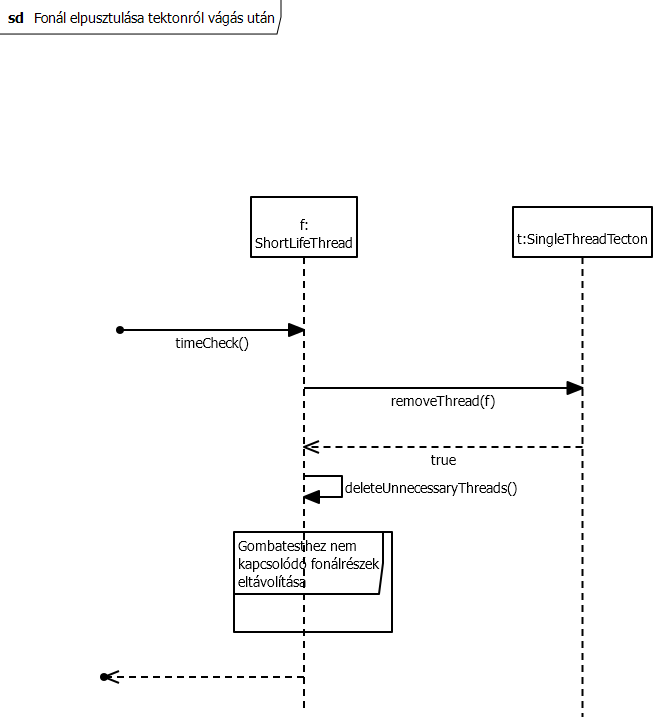
Ezen a diagramon a következő objektumok találhatók:

* i: Rovar, amely a vágást végzi
* t2: ezt a tektont jelöltük ki
* position: ezen a tektonon található a rovar
* f1: gombafonál, amely a t2 és a position tektonon is elágazik

A **cut(t)** metódushívás után leellenőrizzük, hogy a paraméterül kapott t tekton szomszédos-e a position 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 meghívjuk a fonál sendToDie metódusát a t2 tektonra, amely felveszi a fonál time HashMap-jébe a t2 tektont kulcsként, és hozzá értékként 1-et.

**Fonál elpusztulása a tektonról vágás után**



Egy külső kontroller meghívja az f fonál timecheck() metódusát.

Ezen a diagramon a következő objektumok találhatók:

* f: gombafonál
* t: ezen a tektonon pusztul el a gombafonál

A timeCheck() metódust minden kör elején a kontroller meghívja, és ha a time HashMap nem üres, akkor csökkenti 1-el a benne levő elemek értékét. Ha az érték elérte a 0-át, akkor a tárolt tektonról leveszi a fonalat meghívva a removeThread(f: FungalThread) metódusát, majd törli az elemet a HashMapből, illetve a gombafonál tectons listájából törli a tectont, és lefuttatja a deleteUnnecessaryThreads() metódust.

**Bogár megevése**

Egy külső kontroller meghívja az f gombafonál eatInnsect(i)metódusát.

Ezen a diagramon a következő objektumok találhatók:

* f: gombafonál, amely megeszi a rovarat
* i: Rovar
* position: ezen a tektonon található az i rovar
* m: keletkező gombatest

Az eatInsect() metódus leveszi a rovart a tektonról, majd törli a rovart. ezután megnézi, hogy a tektonon, amelyen a rovar volt van-e gombatest, illetve hogy nem-e AbsorbingTecton típusú. Ha ebből egyik se teljesül, létrehoz egy új gombatestet, amelyet elhelyez a tektonra, ahonnan a rovart megette.

**Rovar mozgása során DividingSpore-t evett meg**



Egy külső kontroller meghívja az i rovar move metódusát.

Ezen a diagramon a következő objektumok találhatók:

* i: Rovar
* t2: erre a tektonra lépünk a rovarral
* t1: ezen a tektonon van rajta a rovar
* spores[0]: ezt a spórát eszi meg a rovar
* neighbors[0]: a t2 tekton első szomszédja

Azt mutatja be az ábra, amikor a Rovar sikeresen mozog egy tektonra, a mozgás akkor sikeres, ha a két tekton között van gombafonál és szomszédosak(a függvényen belül megnézi, hogy a tekton szomszéd listájába szerepel-e a másik). Ekkor az ott lévő spórák közül megesz egyet és megkapja annak a hatását, majd a spóra megszűnik létezni.

Jelen esetben egy DividingSpore típusú spórát evett meg, amely meghívja a bogár divide függvényét, ennek hatására keletkezik egy új bogár, amely a régi bogár tektonjának az első szomszédjára kerül.

A diagram elején az i:Insect position adattagja egyenlő t1 tektonnal, ezt adjuk át a putInsect-ben.

* 1. ***Prototípus interface-definíciója***
     1. **Az interfész általános leírása**

A projekt teszteléséhez egy szöveges, konzol alapú interfészt alkalmazunk. Az interfész parancsalapú, a felhasználó a különböző fázisokban strukturált parancsokat adhat ki. Ezeket a rendszer egyéni szintaktikával értelmezi és hajtja végre.

át kell gondolni, mely metódusokat tudja a kontroller meghívni és mely metódusokat tudja a view meghívni

MushroomController, MushroomView

TectonController, TectonView

FungalThreadController, FungalThreadView

InsectController, InsectView

SporeController, SporeView

* + 1. **Bemeneti nyelv**

***arrange***

**Leírás: Arrange fázis elkezdését jelenti, innentől lehet kiadni az arrange fázisba tartozó parancsokat.**

**Opciók: -**

***end***

**Leírás: Aktuális fázis véget ér, ezután lehet másik fázis kezdése parancsot kiadni.**

**Opciók: -**

***act***

**Leírás: act fázis elkezdését jelenti, innentől lehet kiadni az act fázisba tartozó parancsokat.**

**Opciók: -**

**Ábel**

**saveResult**

Leírás: A result.txt fájlba menti az aktuális állapotot.

Opciók: -

Zsombi

**loadResult**

Leírás:Betölti a result.txt fájlból az állapotot.

Opciók: -

**Zsolt**

***turnOffRandom***

**Leírás: Ha ez be van kapcsolva a program minden egyes alkalomkor, mikor valami random értéket szeretne a felhasználóhoz fordul, és kér egy értéket**

**Opciók: -**

**Soma**

***turnOnRandom***

**Leírás: Ha ez be van kapcsolva a program minden egyes alkalomkor, mikor valami random értéket szeretne a program generál egy random értéket**

**Opciók: -**

**Soma**

**loadInit <InitFájlneve>**

Leírás: Ahhoz, hogy elkezdhessük a játékot, először fel kell építenünk a pályát, azaz a tektonokat és a szomszédságaikat, amelyet egy adott fájlból fogunk betölteni

Opciók: -

**Soma**

**assert<Elvártkimenetfájlnév>**

Leírás:A result.txt fájlban elmentett állapotát a rendszernek összehasonlítja a tesztesethez tartozó már megadott .txt fájllal, és dönt a helyes működésről

Opciók: -

**runTest<Tesztesetfájlneve>**

Leírás: A parancs során a program megadott .txt fájlból sorra olvassa be és futtatja le a parancsokat, amig az összes parancsot végre nem hajtja a fájlból, azaz a tesztesetet le nem futtatja

Opciók: -

**Arrange parancsok:**

**Soma**

**createTecton <Tektonfajta>**

Leírás: A parancs hatására az adott tektonfajta alapján létrejön egy új tekton, a nevét a kontroller generálja, az alapján, hogy eddig hány tekton volt a játékban, a tekton neveknek a következő a szintaxisa: t<játékban előfordult tektonok száma+1>

Opciók: -

Ábel

***setNeighbors <Tektonnév> [Tektonnév] [Tektonnév] [Tektonnév]***

**Leírás: Adott nevű tektonnak beállítja a szomszédjait.**

**Opciók: Meg lehet adni egy vagy több szomszédot is.**

**soma**

***createShortLifeThread <Tektonnév> <Játékosnév>***

Leírás: Létrehoz egy ShortLifethread fonalat a <Tektonnév> tectonon, illetve odaadja ezt a gombafonalat az adott nevű játékosnak, a fonal nevét a kontroller generálja, az alapján, hogy eddig hány tekton volt a játékban, a gombafonál neveknek a következő a szintaxisa: f<játékban előfordult fonalak száma+1>

Opciók: -

**Soma**

***createLongLifeThread <Tektonnév> <Játékosnév>***

Leírás: Létrehoz egy LongLifethread fonalat a Tektonnév tectonon, illetve odaadja ezt a gombafonalat az adott nevű játékosnak, a fonal nevét a kontroller generálja, az alapján, hogy eddig hány tekton volt a játékban

Opciók: -

***Zsombi***

***setTectons <Fonálnév> [Tektonnév] [Tektonnév] [Tektonnév]***

**Leírás: A fonál tectons listájába beveszi a tektonokat, valamint a tektonokra is rárakja a fonalat.**

**Opciók: Megadható egy vagy több tekton is.**

Soma

***createMushroom <Tektonnév> <Fonálnév> <Játékosnév>***

Leírás: Létrehoz egy gombatestet az adott tektonra, és beállítja, hogy a Fonálnév nevű fonálhoz tartozik, illetve odaadja az adott játékosnak, és beállítja az éltekorát nullára, a gombatest nevét a kontroller generálja, az alapján, hogy eddig hány gombatest volt a játékban, a gombatest neveknek a következő a szintaxisa: m<játékban előfordult gombatestek száma+1>. Ezenkívül növelődik eggyel a <Játékosnév> játékos points attribútuma

Opciók: -

**Soma**

***createEvolvedMushroom <Tektonnév> <Fonálnév> <Játékosnév>***

Leírás: Létrehoz egy fejlett gombatestet az adott tektonra, és beállítja, hogy a <Fonálnév> nevű fonálhoz tartozik, illetve odaadja az adott játékosnak, és beállítja az életkorát 6-ra, a gombatest nevét a kontroller generálja, az alapján, hogy eddig hány gombatest volt a játékban. Ezenkívül növelődik eggyel a <Játékosnév> játékos points attribútuma

Opciók: -

Soma

***createInsect <Tektonnév> <Játékosnév>***

Leírás: Létrehoz egy rovart az adott nevű tektonon, majd odaadja az adott nevű játékosnak, a rovar nevét a kontroller generálja, az alapján, hogy eddig hány rovar volt a játékban

Opciók: -

**Soma**

**createFungusPlayers <Játékosnév> <Játékosnév> <Játékosnév> <Játékosnév>**

Leírás: A parancs során létrejönnek a gombászok a megadott nevek szerint, maximum négy nevet adhatunk meg, mivel egyszerre maximum csak négy gombász játszhat a játékkal

Opciók: -

Soma

**createInsectPlayers <Játékosnév> <Játékosnév> <Játékosnév> <Játékosnév>**

Leírás: A parancs során létrejönnek a rovarászok a megadott nevek szerint, maximum négy nevet adhatunk meg, mivel egyszerre maximum csak négy rovarász játszhat a játékkal

Opciók: -

**Zsolt**

***generateSpore <Gombatest név> <Spórafajta>***

**Leírás: az adott gombatest spores listájába egy <Spórafajta> fajtájú spórát helyez el,** a spspóra nevét a kontroller generálja, az alapján, hogy eddig hány spóra volt a játékban

**Opciók: -**

**Ábel**

***setShotSpores <Gombatest név> <Nem negatív egész szám>***

**Leírás: beállítjuk, hogy a gombatest eddig hány spórát lőtt ki.**

**Opciók: -**

**Soma**

**createSpore <Spórafajta> <Tektonnév> <Fonálnév>**

**Leírás:** Ennek során létrejön egy <Spórafajta> fajtájú spóra, amely az adott fonálhoz tartozik, és az adott tektonra kerül, a spóra nevét a kontroller generálja, az alapján, hogy eddig hány spóra volt a játékban

**Opciók: -**

**Zsombi**

***setState <Rovarnév> <InsectState>***

**Leírás: Beállítja az adott rovar állapotát az adott spórahatásnak megfelelően.**

**Opciók: -**

Zsolt

***setMaxRound <Pozitív egész szám>***

**Leírás: Megmondhatjuk, hogy a játéknak mennyi lehet a maximum körszáma, ezután a játék befejeződik.**

**Opciók:**

**Ábel**

***setMushroomAge* <Gombatestnév> <Nem negatív egész szám>**

Leírás: Beállítja a gombatest életkorát, azaz beállítja, hogy hány köre léteziik a gombatest a játékban

Opciók: -

**Zsombi**

***setFungalPlayerCount <Pozitív egész szám>***

**Leírás: Beállítja a gombászok számát.**

**Opciók: -**

**Zsolt**

***setInsectPlayerCount <Pozitív egész szám>***

**Leírás: Beállítja a rovarászok számát.**

**Opciók: -**

Ábel

***stepGameRound <Nem negatív egész szám>***

Leírás: A játék köreinek számát lépteti a parancs által megadott egész számmal

Opciók: -

Zsombi

**break <Tektonnév>**

Leírás: A parancs hatására a <Tektonnév>-ben megadott tekton kettétörik, ennek során két új tekton keletkezik, ezeknek neveit a kontroller generálja, az alapján, hogy eddig hány tekton volt a játékban

Opciók: -

Zsolt

**absorb <Tektonnév>**

Leírás: A parancs hatására a <Tektonnév>-ben megadott megadott tektonról felszívódnak a gombafonalak

Opciók: -

Ábel

**itt meg egy pici munka evolve <Gombatestnév>**

Leírás: A parancs hatására a <Gombatestnév>-ben megadott megadott gombatest ha még nem fejlett, akkor fejletté válik

Opciók: -

Zsombi

**deleteUnnecessaryThreads <Fonálnév>**

Leírás: A parancs hatására a megadott fonálnak törlődnek azok a részei, melyek nem kapcsolódnak ugyanolyan fajból származó gombatesthez, vagy KeepThreadTecton-hoz

Opciók: -

Zsolt

**timeCheck**

Leírás: A parancs hatására az összes gombafonalon meghívódik a timeCheck() metódus

Opciók: -

Ábel

**divide <rovarnév>**

Leírás: A parancs hatására a kapott rovar kettéválik, az új rovar nevét a kontroller generálja, az alapján, hogy eddig hány rovar volt a játékban

Opciók: -

**View felőli interface parancsai:**

**getState <rovarnév>**

Leírás: A parancs hatására a kapott rovarnak lekérjük az állapotát

Opciók: -

**getInsect <Tectonnév>**

Leírás: A parancs hatására a kapott tektonról lekérjük a bogarat

Opciók: -

**getMushroom <Tectonnév>**

Leírás: A parancs hatására a kapott tektonról lekérjük a gombatestet

Opciók: -

**getFungalThread <Tectonnév>**

Leírás: A parancs hatására a kapott tektonról lekérjük a rajta levő fonalakat

Opciók: -

**getSporeTecton <Tecton>**

Leírás: A parancs hatására a kapott tektonról lekérjük a spórákat

Opciók: -

**Act parancsok:**

**Ábel**

**closeStep**

Leírás: A parancsot a játékosok adják ki, amikor le akarják zárni a lépésüket, ezt az első gombatest, illetve az első rovar lehelyezése kivételével minden lépésben ki kell adni

Opciók: -

**Zsombi**

**putFirstMushroom <Fonáltípus> <Tektonnév>**

Leírás: Gombász első lépése, lehelyez gombatestet adott tektonra, ezzel együtt egy gombafonál is létrejön a tektonon, melynek típusa <Fonáltípus>, a gombatest és a gombafonál nevét a kontroller generálja, az alapján, hogy eddig hány gombatest, illetve gombafonál volt a játékban. Ezenkívül az adott gombásznak növelődik eggyel a points attribútuma

Opciók: -

**Zsolt**

***branchThread <Fonal név> <Tektonnév>***

**Leírás: Adott nevű fonalat átnöveszti az adott tektonra.**

**Opciók: -**

**Ábel**

***shootSpore <Gombatest név> <Tektonnév>***

**Leírás: Adott nevű gombatest spórát lő adott tektonra.**

**Opciók: -**

**Zsombi**

**growMushroom <Fonal név> <Tektonnév>**

**Leírás:** Adott nevű gombafonál gombatestet növeszt az adott tektonra, a gombatest nevét a kontroller generálja, az alapján, hogy eddig hány gombatest, volt a játékban. Ezenkívül az adott gombásznak növelődik eggyel a points attribútuma

**Opciók: -**

**Zsolt**

***eatInsect <Fonal név> <Rovarnév>***

**Leírás: Adott nevű fonál megeszi az adott rovart. Ha lehetséges, akkor gombatest nő ezen a helyen, ekkor az adott gombásznak növelődik eggyel a points attribútuma.**

**Opciók: -**

**Ábel**

**putFirstInsect <Tektonnév>**

**Leírás:** A rovarász első lépése, lehelyez egy rovart az adott tektonra, a rovar nevét a kontroller generálja, az alapján, hogy eddig hány rovar, volt a játékban

Opciók: -

***Soma***

***move <Rovarnév> <Tektonnév>***

**Leírás: Az adott nevű rovar átmozog az adott tektonra. Ha megeszik egy spórát növeljük az adott rovarász points attribútumát eggyel.**

**Opciók: -**

**Soma**

***cut <Rovarnév> <Tektonnév>***

**Leírás: Az adott nevű rovar elvágja a saját tektonjáról az adott tektonra vezető fonalat.**

**Opciók: -**

* + 1. **Kimeneti nyelv**

## Általános leírás

A program képes elmenteni az objektumok állapotát két különböző fájlba:

* **init.txt**: Ez a fájl a teszteset kezdeti állapotát tartalmazza, amit az arrange utasításokkal hoztunk létre.
* **result.txt**: Ez a fájl az act parancsok végrehajtása utáni állapotot tartalmazza.
* A kettőt az assert fázisban betöltjük és összehasonlítjuk őket. Az alapján mondjuk meg a teszt sikerességét, hogy a result.txt tartalma az init.txt-ből kiíndulva, az act utasítás(oka)t alkalmazva az elvárásainknak megfelelő-e.

Az állapot mentésére és betöltésére szolgáló parancsok:

* **saveInit**: Az init.txt fájlba menti az aktuális állapotot. - az arrange fázis után esedékes
* **saveResult**: A result.txt fájlba menti az aktuális állapotot. - az act fázis után esedékes
* **loadInit**: Betölti az init.txt fájlból az állapotot. - assert fázis elején esedékes
* **loadResult**: Betölti a result.txt fájlból az állapotot. - assert fázis eleje esedékes

## Fájlformátum

A fájl szerkezete az alábbi módon épül fel:

-begin-

<Objektum típusa> <Objektum neve>

<Objektum típusa> <Objektum neve>

…

--- # Objektumok listájának vége

<Objektum neve>

<Objektum attribútuma> <Érték/Név> <Érték/Név> ...

<Objektum attribútuma> <Érték/Név>

<Objektum attribútuma> - # ilyen akkor lehet, ha ez az attribútuma még nincs beállítva pl.: null

…

#elválasztó üres sor

<Objektum neve>

…

-end-

A kiíratás kettő részre oszlik el:

* -begin- <-> — : Az szereplők felsorolása, innen tudjuk, hogy milyen objektumok, milyen névvel kerülnek be a HashMap-ünkbe.
* — <-> -end- : Az szereplők attribútumainak felsorolása. Az összeset felsoroljuk. Ha az értéke inicializálatlan pl.: nullptr a listájára, vagy a kilőtt spóráját -1-re állította ctor-a, akkor azt egy - jellel jelöljük. Az attribútum értékeken kívül innen tudjuk meg a szereplők közti asszociációkat.
  + Az objektumok felsorolásait üres sorok tagolják.

### 

### Példa egy konkrét fájlra:

-begin-

MultiThreadTecton t1

SingleThreadTecton t2

FungalThread f1

Mushroom m1

SpeedSpore s1

—

t1

spores -

neighbors t2

i -

mushroom m1

threads f1

t2

spores s1

neighbors t1

i -

mushroom -

thread -

f1

tectons t1

life -

s1

thread f1

-end-

* 1. ***Összes részletes use-case***

| **Use-case neve** | Első gombatest lehelyezése |
| --- | --- |
| **Rövid leírás** | A játék elején a gombász első lépése az első gombatestjének a lehelyezése egy tektonra. Ebben az esetben egy olyan tektonra próbálja lehelyezni a gombász a gombatestet, ahova a játék szabályok szerint le tudja. Ez egy olyan tekton lehet, amin nincs másik gombatest és a fajtája nem AbsorbingTecton. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász választ egy tektont * A tektonra lehelyezhető gombatest * A tektonra rákerül a gombatest |
| **bemenet** | A játékos hatására meghívódik a kiválasztott tektonon a putFirstMushroom. - ***<Gombatest név> putFirstMushroom <Tektonnév>*** |

| **Use-case neve** | Első gombatest lehelyezése olyan tektonra, amelyen már van gombatest |
| --- | --- |
| **Rövid leírás** | A játék elején a gombász első lépése az első gombatestjének a lehelyezése egy tektonra. Ebben az esetben egy olyan tektonra próbálja lehelyezni a gombász a gombatestet, ahol már van gombatest. Ekkor nem fog lehelyeződni a gombatest a tektonra. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász választ egy tektont * A tektonon van már gombatest * A tektonon nem történik változás, a gombatest nem kerül lehelyezésre |
| **bemenet** | A játékos hatására meghívódik a kiválasztott tektonon a putFirstMushroom. - ***<Gombatest név> putFirstMushroom <Tektonnév>*** |

| **Use-case neve** | Első gombatest lehelyezése absorbing tektonra |
| --- | --- |
| **Rövid leírás** | A játék elején a gombász első lépése az első gombatestjének a lehelyezése egy tektonra. Ebben az esetben egy olyan tektonra próbálja lehelyezni a gombász a gombatestet, ami Absorbing típusú. Az ilyen típusú tektonokon nem lehet gombatest. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász választ egy tektont * A tekton Absorbing típusú * A tektonon nem történik változás, a gombatest nem kerül lehelyezésre |
| **bemenet** | A játékos hatására meghívódik a kiválasztott tektonon a putFirstMushroom. - ***<Gombatest név> putFirstMushroom <Tektonnév>*** |

| **Use-case neve** | Gombafonál növesztése olyan tektonra, amelyre tud növeszteni |
| --- | --- |
| **Rövid leírás** | A gombász a körében tud gombafonalat növeszteni, elágaztatni egy adott tektonon. Ehhez ki kell választania azt a tektont, ahova a fonalát el szeretné ágaztatni. Ezt akkor tudja sikeresen véghez vinni, ha a kiválasztott tektonnak van legalább egy olyan szomszédja, ahol van azonos fajtájú FungalThread fonál az elágaztatni kívánttal. Továbbá fontos, hogy SingleThread fajtájú tektonon csak egy fonál ágazhat el. Ebben az esetben olyan tektonon ágaztatja el a fonalát a játékos, amelyen el lehet. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász választ egy tektont * A tektonon elágaztatható a játékos fonala * A tektonon elágazik a játékos fonala |
| **bemenet** | A játékos hatására meghívódik a fonalán a branchThread metódus, amely paraméterül a kiválasztott tektont kapja. - ***<Fonal név> branchThread <Tektonnév>*** |

| **Use-case neve** | Gombafonál növesztése olyan tektonra, amelyre nem tud növeszteni, mert a tekton messze van |
| --- | --- |
| **Rövid leírás** | A gombász a körében tud gombafonalat növeszteni, elágaztatni egy adott tektonon. Ehhez ki kell választania azt a tektont, ahova a fonalát el szeretné ágaztatni. Ezt akkor tudja sikeresen véghez vinni, ha a kiválasztott tektonnak van legalább egy olyan szomszédja, ahol van azonos fajtájú FungalThread fonál az elágaztatni kívánttal. Továbbá fontos, hogy SingleThread fajtájú tektonon csak egy fonál ágazhat el. Ebben az esetben olyan tektonon ágaztatja el a fonalát a játékos, amelyen nem tudja, mert a tektonnak nincs egy olyan szomszédja se, amelyen el van ágazva a játékos fonala. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász választ egy tektont * A tektonon nem ágazható el a játékos fonala, mert messze van (egy szomszédján sincs az elágaztatni kívánttal azonos fajtájú fonál) * A tektonon nem történik változás, a fonál nem ágazik rajta el |
| **bemenet** | A játékos hatására meghívódik a fonalán a branchThread metódus, amely paraméterül a kiválasztott tektont kapja. - ***<Fonal név> branchThread <Tektonnév>*** |

| **Use-case neve** | Gombafonál növesztése olyan SingleThread tektonra, amelyre nem tud növeszteni, mert a tektonon már ágazik el másik fajta fonál |
| --- | --- |
| **Rövid leírás** | A gombász a körében tud gombafonalat növeszteni, elágaztatni egy adott tektonon. Ehhez ki kell választania azt a tektont, ahova a fonalát el szeretné ágaztatni. Ezt akkor tudja sikeresen véghez vinni, ha a kiválasztott tektonnak van legalább egy olyan szomszédja, ahol van azonos fajtájú FungalThread fonál az elágaztatni kívánttal. Továbbá fontos, hogy SingleThread fajtájú tektonon csak egy fonál ágazhat el. Ebben az esetben olyan tektonon ágaztatja el a fonalát a játékos, amelyen nem tudja, mert a tekton típusa SingleThread, és már ágazik el rajta egy másik fajta fonál. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász választ egy tektont, ami típusa SingleThread * A tektonon nem ágazható el a játékos fonala, mert már elágazik rajta egy másik fajta fonál (másik játékos fonala) * A tektonon nem történik változás, a fonál nem ágazik rajta el |
| **bemenet** | A játékos hatására meghívódik a fonalán a branchThread metódus, amely paraméterül a kiválasztott tektont kapja. - ***<Fonal név> branchThread <Tektonnév>*** |

| **Use-case neve** | Gombafonál sikeres növesztése olyan tektonra, amelyen spóra van |
| --- | --- |
| **Rövid leírás** | A gombász a körében tud gombafonalat növeszteni, elágaztatni egy adott tektonon. Ehhez ki kell választania azt a tektont, ahova a fonalát el szeretné ágaztatni. Ezt akkor tudja sikeresen véghez vinni, ha a kiválasztott tektonnak van legalább egy olyan szomszédja, ahol van azonos fajtájú FungalThread fonál az elágaztatni kívánttal. Továbbá fontos, hogy SingleThread fajtájú tektonon csak egy fonál ágazhat el. Ebben az esetben olyan tektonon ágaztatja el a fonalát a játékos, amelyen tudja. Továbbá ezen a tektonon van olyan spóra, ami ugyanolyan fajtájú (ugyanahhoz a játékoshoz tartozik), mint a fonál. Ekkor a játékos újra elágaztathatja a fonalát egy olyan tektonra, amely szomszédos azzal a tektonnal, amelyen a spóra rajta van. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * A játékos választ egy tektont * A tektonon elágaztatható a játékos fonala * A tektonon van olyan spóra, amely a játékoshoz tartozik * A tektonon elágazik a játékos fonala * A játékos kiválaszt egy új tektont * A tekton szomszédos azzal a tektonnal, amelyen előbb elágazott a játékos fonala * A tektonon elágaztatható a játékos fonala * A tektonon elágazik a játékos fonala |
| **bemenet** | A játékos hatására meghívódik a fonalán a branchThread metódus, amely paraméterül a kiválasztott tektont kapja. - ***<Fonal név> branchThread <Tektonnév>***  Ezután a játékos hatására megint meghívódik a fonalán a branchThread metódus, amely paraméterül a kiválasztott tektont kapja, amely szomszédos az előzőleg kiválasztott tektonnal. - ***<Fonal név> branchThread <Tektonnév>*** |

| **Use-case neve** | Fejletlen gombatest spóraszórása olyan tektonra, amelyre tud |
| --- | --- |
| **Rövid leírás** | A gombász játékos a saját körében lőhet spórát az egyik gombatestjéből. Ekkor a játékosnak ki kell választania azt a gombatestet, amelyikből a spórát lőni szeretné. Ekkor ha a gombatestben van termelt spóra, akkor ki kell választania a játékosnak azt a tektont, ahova lőni szeretné a spórát. Ebben az esetben a kiválasztott gombatest fejletlen, így a céltektonnak muszáj szomszédosnak lennie a gombatest tektonjával. A kiválasztott céltekton most szomszédos a gombatest tektonjával. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász kiválaszt egy hozzá tartozó gombatestet, ami fejletlen * A gombász kiválaszt egy tektont, ahova a spórát lőni szeretné, ez a tekton szomszédos a gombatest tektonjával * A spóraszórás sikeresen végrehajtódik (a spóra rákerül a céltektonra, és törlődik a gombatestből) |
| **bemenet** | A játékos hatására meghívódik a kiválasztott gombatesten a shootSpore metódus, amely paraméterül a kiválasztott céltektont kapja. - ***<Gombatest név> shootSpore <Tektonnév>*** |

| **Use-case neve** | Fejlett gombatest spóra szórása tektonra, a céltekton nem közvetlen szomszéd |
| --- | --- |
| **Rövid leírás** | 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. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász kiválaszt egy hozzá tartozó gombatestet, ami fejlett * A gombász kiválaszt egy tektont, ahova a spórát lőni szeretné, ez a tekton nem szomszédos, hanem a gombatest szomszédjának szomszédja. * A spóraszórás sikeresen végrehajtódik (a spóra rákerül a céltektonra, és törlődik a gombatestből) |
| **bemenet** | A játékos hatására meghívódik a kiválasztott gombatesten a shootSpore metódus, amely paraméterül a kiválasztott céltektont kapja. - ***<Gombatest név> shootSpore <Tektonnév>*** |

| **Use-case neve** | Fejlett gombatest spóra szórása tektonra, a céltekton közvetlen szomszéd |
| --- | --- |
| **Rövid leírás** | Ugyanaz mint az előzőnél, csak a céltekton közvetlen szomszéd. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | Ugyanaz mint az előzőnél, csak a céltekton közvetlen szomszéd. |
| **bemenet** | ***<Gombatest név> shootSpore <Tektonnév>*** |

| **Use-case neve** | Sikertelen fejletlen spóraszórás, mert nem szomszédos a céltekton. |
| --- | --- |
| **Rövid leírás** | Egy Mushroom (fejlettlen) objektum megpróbál spórát lőni egy tekton célpontra. Mivel nem szomszédos a Mushroom tektonja a céltekton-al, a művelet sikertelen. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász kiválaszt egy hozzá tartozó gombatestet, ami fejlettlen * A gombász kiválaszt egy tektont, ahova a spórát lőni szeretné, ez a tekton nem szomszédos. * Már tovább se nézzük. A spóraszórás sikertelen (a spóra nem kerül rá a céltektonra, és nem törlődik a gombatestből) |
| **bemenet** | A játékos hívja a fejletlen gombateseten a shootspore-t egy nem szomszédos tektont átadva paraméterül.  ***<Gombatest név> shootSpore <Tektonnév>*** |

| **Use-case neve** | Sikertelen fejletlett spóraszórás, mert se nem közvetetten se nem közvetlenül szomszédos a céltekton. |
| --- | --- |
| **Rövid leírás** | Egy Mushroom (fejlett) objektum megpróbál spórát lőni egy tekton célpontra. Mivel nem szomszédos (közvetlenül vagy közvetetten) a Mushroom tektonja a céltekton-al, a művelet sikertelen. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász kiválaszt egy hozzá tartozó gombatestet, ami fejlettlen * A gombász kiválaszt egy tektont, ahova a spórát lőni szeretné, ez a tekton nem szomszédos és szomszédjaival sem szomszédos * A spóraszórás sikertelen (a spóra nem kerül rá a céltektonra, és nem törlődik a gombatestből) |
| **bemenet** | A játékos hívja a fejlett gombateseten a shootspore-t egy elég távoli tektont átadva paraméterül.  ***<Gombatest név> shootSpore <Tektonnév>*** |

| **Use-case neve** | Gombatest 10. spórájának elszórása, a gombatest meghal. |
| --- | --- |
| **Rövid leírás** | A gombatest már 9 spórát ellőtt, ezért a következő lövés hatására meghal. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * Gombász kiválaszt egy hozzá tartozó gombatestet, ami már 9-szer lött. * Spórát lő sikeresen. * A gombatest meghal és lekerül a tektonról. |
| **bemenet** | ***<Gombatest név> shootSpore <Tektonnév>*** |

| **Use-case neve** | Gombatest növesztése sikeres. |
| --- | --- |
| **Rövid leírás** | A gombatest növesztéséhez szükséges előfeltételek teljesülnek. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * A gombász kiválaszt egy hozzá tartozó gombafonalat és egy tektont. * Meghívja a **growMushroom()** metódust a kiválasztott fonálon a megfelelő paraméterekkel. * A rendszer ellenőrzi, hogy a tektonon nincs-e már gombatest. * A rendszer lekéri a spórák listáját a tektonról. * A tekton visszaadja a spórák listáját. * Ha mindhárom spóra a kiválasztott fonalhoz tartozik, a rendszer létrehozza a gombatestet.   Eredmény:   * A céltektonon megjelenik az új gombatest a megfelelő attribútumokkal. * A gombatest beállítja a helyét és a fonalát. * A rendszer eltávolítja a spórákat a tektonról. * A growMushroom metódus sikeres. |
| **bemenet** | ***<Fonálneve> growMushroom <Tectonneve> <Mushroomneve>*** |

| **Use-case neve** | A gombafonál megeszi a bénult rovart és új gombatest nő. |
| --- | --- |
| **Rövid leírás** | A gombafonál megeszi a bénító spórát evett rovart és mivel ez nem egy szivó tektonon történt, ezért gombatest keletkezik a rovar helyén. |
| **Aktorok** | játékos (gombász) |
| **Forgatókönyv** | * A gombász kiválaszt egy hozzá tartozó gombafonalat és egy bénult rovart, ami olyan tektonon van, amin a fonala el van ágazva. * A gombász meghívja az eatInsect(bénult rovar) metódust a fonálon. * A rendszer eltávolítja a rovart a tektonról. * A rovar törlődik a rendszerből. * A rendszer ellenőrzi, hogy a tektonon már van-e gombatest. Most nincs. - Ha lenne, a folyamat véget érne. * A rendszer ellenőrzi, hogy a tekton AbsorbingTecton típusú-e. Most nem - Ha az lenne, a folyamat véget érne. * Mivel egyik feltétel sem teljesült, egy új gombatest jön létre. * Az új gombatest pozíciója beállításra kerül a tekton helyére. * A gombatest fonala beállításra kerül a kiválasztott gombafonalra. * A gombatest elhelyeződik a tektonon. * A eatInsect() metódus sikeresen végrehajtódott. |
| **bemenet** | ***<Fonálneve> eatInsect <Rovarneve>*** |

| **Use-case neve** | A gombafonál megeszi a bénult rovarat nem keletkezik új gombatest mert szívótektonon történt az evés. |
| --- | --- |
| **Rövid leírás** | A gombafonál megeszi a bénító spórát evett rovart. Mivel ez egy szívó tektonon történt, ezért nem keletkezik új gombatest a rovar helyén. |
| **Aktorok** | Játékos (gombász) |
| **Forgatókönyv** | A forgató könyv megegyezik a az előzőével, (a happy path-ével) amíg:   * A rendszer ellenőrzi, hogy a tekton AbsorbingTecton típusú-e. Most igen. - A folyamat véget ér, nem jön létre új gombatest. * Ettől még az eatInsect() metódus sikeresen végrehajtódik és true értékkel tér vissza |
| **bemenet** | ***<Fonálneve> eatInsect <Rovarneve>*** |

| **Use-case neve** | A gombafonál megeszi a bénult rovarat nem keletkezik új gombatest mert már van rajta gombatest. |
| --- | --- |
| **Rövid leírás** | A gombafonál megeszi a bénító spórát evett rovart. Mivel a tektonon már van egy gombatest, ezért nem keletkezik új gombatest a rovar helyén. |
| **Aktorok** | Játékos (gombász) |
| **Forgatókönyv** | A forgatókönyv megegyezik az előző (happy path) esettel, amíg:   * A rendszer ellenőrzi, hogy a tektonon már van-e gombatest. Most van. - A folyamat véget ér. * Ettől még az eatInsect() metódus sikeresen végrehajtódott. |
| **bemenet** | ***<Fonálneve> eatInsect <Rovarneve>*** |

| **Use-case neve** | Rovar lehelyezése tektonra sikeres |
| --- | --- |
| **Rövid leírás** | A játékos a játék elején egy rovart helyez le egy tektonra, amin még nincsen rovar. |
| **Aktorok** | Játékos (rovarász) |
| **Forgatókönyv** | * A játékos elvégzi a **putFirstInsect()** metódushívást * A tekton ellenőrzi, hogy nincs-e már rajta rovar. * Mivel a tekton szabad, létrehoz egy új **Rovar** objektumot. * Beállítja a rovar pozícióját saját magára. * Beállítja, hogy a rovar mostantól rajta található. * A tektonon megjelenik az új rovar a megfelelő pozícióval. * A rovarlehelyezés sikeres. |
| **bemenet** | ***<Rovarneve> putFirstInsect <Tectonneve>*** |

| **Use-case neve** | Rovar lehelyezése foglalt tektonra |
| --- | --- |
| **Rövid leírás** | A játék elején a rovarászok első lépése az első rovarjának lehelyezése egy tektonra. Ebben a tesztben azt teszteljük, amikor a lehelyezés egy foglalt tekonra próbálkozik. Ekkor nincs változás |
| **Aktorok** | játékos |
| **Forgatókönyv** | * Rovarász választ egy tektont * A tekton foglalt * Nem helyeződik le rovar |
| **bemenet** | myRovar putFirstInsect foglaltTekton |

| **Use-case neve** | Rovar mozgatása sikeresen |
| --- | --- |
| **Rövid leírás** | A játék során a rovarász lépése lehet az, hogy rovarát átmozgassa egy másik szomszédos tektonra, amire fonál vezet a saját tektonjáról. Sikeres mozgás után a rovar automatikusan megesz egy spórát az új tektonról, ha van rajta. |
| **Aktorok** | játékos |
| **Forgatókönyv** | * Rovarász választ egy tektont * A tekton szomszédos * A a saját és új tekton között van fonal * A rovar átmozog * Nincs spóra az új tektonon |
| **bemenet** | myRovar move üresTekton |

| **Use-case neve** | Rovar mozgása sikertelen |
| --- | --- |
| **Rövid leírás** | A játék során a rovarász lépése lehet az, hogy rovarát átmozgassa egy másik szomszédos tektonra, amire fonál vezet a saját tektonjáról. Sikeres mozgás után a rovar automatikusan megesz egy spórát az új tektonról, ha van rajta. |
| **Aktorok** | játékos |
| **Forgatókönyv** | * Rovarász választ egy tektont * A tekton szomszédos * A a saját és új tekton között nincs fonal * A rovar marad az eredeti helyén |
| **bemenet** | myRovar move fonáltalanTekton |

| **Use-case neve** | Rovar evése mozgás után - SpeedSpore |
| --- | --- |
| **Rövid leírás** | Sikeres mozgás után a rovar automatikusan megesz egy spórát az új tektonról, ha van rajta. Ebben az esetben volt gyorsítóspóra, ezért a rovarász még egyet léphet. |
| **Aktorok** | játékos |
| **Forgatókönyv** | * Rovarász választ egy tektont * A tekton szomszédos * A a saját és új tekton között van fonal * A rovar átmozog * Egy gyorsítóspórát megesz az új tektonról * Megkapja a hatását |
| **bemenet** | myRovar move speedSporeTekton |

| **Use-case neve** | Rovar evése mozgás után - SlowingSpore |
| --- | --- |
| **Rövid leírás** | Sikeres mozgás után a rovar automatikusan megesz egy spórát az új tektonról, ha van rajta. Ebben az esetben volt lassító, ezért a rovar a következő körben nem tud mozogni, csak vágni |
| **Aktorok** | játékos |
| **Forgatókönyv** | * Rovarász választ egy tektont * A tekton szomszédos * A a saját és új tekton között van fonal * A rovar átmozog * Egy lassítóspórát megesz az új tektonról * Megkapja a hatását |
| **bemenet** | myRovar move slowSporeTekton |

| **Use-case neve** | Rovar evése mozgás után - ParalyzingSpore |
| --- | --- |
| **Rövid leírás** | Sikeres mozgás után a rovar automatikusan megesz egy spórát az új tektonról, ha van rajta. Ebben az esetben volt bénítóspóra, ezért a rovar a következő körben nem tud se mozogni, se vágni, tehát teljesen kimarad. |
| **Aktorok** | játékos |
| **Forgatókönyv** | * Rovarász választ egy tektont * A tekton szomszédos * A a saját és új tekton között van fonal * A rovar átmozog * Egy bénítóspórát megesz az új tektonról * Megkapja a hatását |
| **bemenet** | myRovar move paralyzingSporeTekton |

| **Use-case neve** | Rovar evése mozgás után - NoCutSpore |
| --- | --- |
| **Rövid leírás** | Sikeres mozgás után a rovar automatikusan megesz egy spórát az új tektonról, ha van rajta. Ebben az esetben volt vágástalanító, ezért a rovar a következő körben nem vágni, de mozogni igen. |
| **Aktorok** | játékos |
| **Forgatókönyv** | * Rovarász választ egy tektont * A tekton szomszédos * A a saját és új tekton között van fonal * A rovar átmozog * Egy vágástalanító spórát megesz az új tektonról * Megkapja a hatását |
| **bemenet** | myRovar move nocutSporeTekton |

| **Use-case neve** | Tekton kettétörése kör elején |
| --- | --- |
| **Rövid leírás** | Minden kör elején egy tektonnak ketté kell törnie, tehát a kontroller meghívja egy tektonra a break() metódusát. A tesztelés során miután minden játékos elvégezte a lépését, az új kör legelején hogy tesztelni tudjuk a helyes működését a törés nem random fog történni, hanem a játékos adja meg, hogy melyik tektont szeretné kettétörni |
| **Aktorok** | játékos |
| **Forgatókönyv** | Miután minden játékos lépett az új kör kezdésekor a megadott tekton kettétörik, két új tekton keletkezik, az eredeti tekton pedig elpusztul. Ha volt a tektonon bogár az elsőnek létrejöttre kerül, a spórák, amelyek rajta voltak a tektonnal együtt eltűnnek, illetve a fonálrész, amely a tektonon volt elpusztul |
| **bemenet** | Meg kell adni melyik tekton törjön ketté |

| **Use-case neve** | Gombafonál felszívódása 4 kör eltelte után |
| --- | --- |
| **Rövid leírás** | Minden 4. kör eltelte után az AbsorbingTekton-on a gombafonalak fel kell szívódjanak, tehát a kontrollernek meg kell hívnia az AbsorbingTectonok absorb() függvényét |
| **Aktorok** | játékos |
| **Forgatókönyv** | A 3. kör végén, miután az utolsó játékos is befejezte a lépését, mikor a játék átmegy a 4. körbe, a kontroller meg kell hívja az AbsorbingTectonok-on az absorb() metódust |
| **bemenet** | A 3. körben az utolsó játékos be kell fejezze a lépését |

| **Use-case neve** | Gombatest fejletté válása 5 kör után |
| --- | --- |
| **Rövid leírás** | A gombatest lehelyezését követő 5. körben a gombatest fejletlen állapotból fejletté válik |
| **Aktorok** | - |
| **Forgatókönyv** | * A gombatest lehelyezésre kerül * 5 kör eltelik * Állapot átmenet |
| **bemenet** | 5 kör eltelik a lehelyezéstől számítva |

| **Use-case neve** | Játék befejezése egy megadott kör után |
| --- | --- |
| **Rövid leírás** | A játék legelején megadjuk, hogy hány kör után fejeződjön be a játék, ezután a játékosok már nem tudnak többet lépni |
| **Aktorok** | játékos |
| **Forgatókönyv** | A legutolsó kör végén, miután az utolsó játékos is lépett a program játék vége állapotba kell kerüljön |
| **bemenet** | Az utolsó körben az utolsó játékos be kell fejezze a lépését |

| **Use-case neve** | Rovar mozgatása olyan tektonra, amelyen DividingSpore van |
| --- | --- |
| **Rövid leírás** | A játék során a rovarász lépése lehet az, hogy rovarát átmozgassa egy másik szomszédos tektonra, amire fonál vezet a saját tektonjáról. Sikeres mozgás után a rovar automatikusan megesz egy spórát az új tektonról, ha van rajta. Ebben az esetben ez a spóra DividingSpore, ami miatt a rovar osztódni fog és az állapota normálra fog visszaállni. |
| **Aktorok** | játékos (Rovarász) |
| **Forgatókönyv** | * Rovarász választ egy tektont * A tekton szomszédos * A a saját és új tekton között van fonal * A rovar átmozog * DividingSpore van a tektonon |
| **bemenet** | ***<Rovarneve> move <Tectonneve>*** |

| **Use-case neve** | Rovar fonal vágása |
| --- | --- |
| **Rövid leírás** | A játék során a rovarász lépése lehet az, hogy rovarával elvág egy gombafonalat. Ez akkor lehetséges, ha a kiválasztott gombafonál a rovar pozícióját jelölő tektonnal szomszédos tektont választ ki, amin van gombafonal, de gombatest nincs. |
| **Aktorok** | játékos (Rovarász) |
| **Forgatókönyv** | * Rovarász választ egy tektont * A tekton szomszédos * A a saját és új tekton között van fonal * Meghívódik a gombafonál sendToDie metódusa |
| **bemenet** | ***<Rovarneve> cut <Tectonneve>*** |

| **Use-case neve** | Rovar fonal vágása olyan tektonon, amin van gombatest |
| --- | --- |
| **Rövid leírás** | A játék során a rovarász lépése lehet az, hogy rovarával elvág egy gombafonalat. Ez akkor lehetséges, ha a kiválasztott gombafonál a rovar pozícióját jelölő tektonnal szomszédos tektont választ ki, amin van gombafonal, de gombatest nincs. |
| **Aktorok** | játékos (Rovarász) |
| **Forgatókönyv** | * Rovarász választ egy tektont * A tekton szomszédos * A kiválasztott tektonon van fonal * A vágás nem megy végbe |
| **bemenet** | ***<Rovarneve> cut <Tectonneve>*** |

| **Use-case neve** | Körváltás, ha minden játékos befejezte a lépését |
| --- | --- |
| **Rövid leírás** | A játék során a körszámláló akkor lép eggyel, amikor minden játékos végrehajtotta a lépését az adott körben. Ez lehet például az első kör, amikor minden játékos lerakta az első gombatestjét és a hozzá tartozó gombafonalat. A teszt célja annak ellenőrzése, hogy ilyenkor valóban megtörténik-e a körváltás. |
| **Aktorok** | Játékosok |
| **Forgatókönyv** | * Minden játékos lerakja a gombafonalát * Minden játékos lerakja a gombatestét * A körszámláló növekszik |
| **bemenet** | *Megfelelő lépések elvégzése* |

| **Use-case neve** | Gombafonal tényleges elvágása ShortLife esetben |
| --- | --- |
| **Rövid leírás** | A játék során ha egy rovar vág egy gombafonalat, amely ShortLifeThread típusú, akkor az elvágás csak a következő körben történik meg ténylegesen. A teszteset célja annak ellenőrzése, hogy a sendToDie() hívás után csak a következő körben tűnik el a fonál. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | * Egy rovar cut metódusa sikeresen meghívódik * A kiválasztott tektonon ShortLifeThread van * A metódus a kapott tektont behelyezi a life<Tecton, int> HashMap-be kulcsként, értékként pedig egyet állít be * A gombafonalnak timeCheck() metódusa a következő kör elején meghívódik * A HashMap-ben van elem, így értékét csökkenti 1-el, azaz 0 lesz * Ekkor a tárolt tektonról leveszi a fonalat, majd az elemet kitörli a HashMap-ből |
| **bemenet** | ***<Rovarneve> cut <Tectonneve>*** |

| **Use-case neve** | Gombafonal tényleges elvágása LongLife esetben |
| --- | --- |
| **Rövid leírás** | A játék során ha egy rovar vág egy gombafonalat, amely LongLifeThread típusú, akkor az elvágás csak a következő utáni körben történik meg ténylegesen. A teszteset célja annak ellenőrzése, hogy a sendToDie() hívás után csak a következő utáni körben tűnik el a fonál. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | * Egy rovar cut metódusa sikeresen meghívódik * A kiválasztott tektonon LongLifeThread van * A metódus a kapott tektont behelyezi a life<Tecton, int> HashMap-be kulcsként, értékként pedig kettőt állít be * A gombafonalnak timeCheck() metódusa a következő kör elején meghívódik * A HashMap-ben van elem, így értékét csökkenti 1-el, azaz 1 lesz * A gombafonalnak timeCheck() metódusa a következő kör elején ismét meghívódik * A HashMap-ben van elem, így értékét csökkenti 1-el, azaz 0 lesz * Ekkor a tárolt tektonról leveszi a fonalat, majd az elemet kitörli a HashMap-ből |
| **bemenet** | ***<Rovarneve> cut <Tectonneve>*** |

| **Use-case neve** | Spóragenerálás páros körökben |
| --- | --- |
| **Rövid leírás** | A játék során minden páros kör elején minden játékban lévő gombatest automatikusan kap egy új spórát. A teszt célja annak ellenőrzése, hogy egy páratlan körből párosba váltva valóban minden gombatest generál-e egy új spórát. |
| **Aktorok** | - |
| **Forgatókönyv** | * A körszámláló páros számra vált * Minden gombatesten meghívódik a generateSpore metódus * Sikeres generálás esetén true értékkel térnek vissza |
| **bemenet** | *A körök száma párosra vált* |

* 1. ***Tesztelési terv***

| **Teszt-eset neve** | Teszt\_1 |
| --- | --- |
| **Rövid leírás** | Ez a teszteset az első kör lefutását mutatja be, amely során 1 rovarász és 1 gombász kötelezően az első rovarjukat illetve gombatestjüket rakják le helyesen.   * Rovarász : választ t1 Tektont * Gombász választ t2 Tektont   A before és az aftert, a kimentett txt-k alapján komparáljuk |
| **Teszt célja** | Tesztelje a rovar lehelyezése illetve az első gombatest lehelyezése funkciót. |

| **Teszt-eset neve** | Teszt\_2 |
| --- | --- |
| **Rövid leírás** | Ezt a teszt eset a játék egy közbenső körét mutatja be, ahol 5 rovarász és 3 gombász lépéssorozatát mutatja be.  főleg a rovar mozgására koncentrál   * Rovarász1 : rovar mozgatása t1 tekton-ról SpeedSpore-t tartalmazó Tecton-ra, ahol megeszik egy gyorsítóspórát. * Rovarász2 : rovar mozgatása t2 tekton-ról SlowSpore-t tartalmazó Tecton-ra ahol megeszik egy lassítóspórát * Rovarász3 : rovar mozgatása t3 tekton-ról ParalyzingSpore-t tartalmazó Tecton-ra ahol megeszik egy bénítóspórát * Rovarász4 : rovar mozgatása t4 tekton-ról NoCutSpore-t tartalmazó Tecton-ra ahol megeszik egy vágástalanító spórát * Rovarász 5 : rovar mozgatása t5 tekton-ról DividingSpore-t tartalmazó Tectonra, ahol a rovar megeszi ezt a spórát és osztódik * Gombász2 : Gombafonala növesztése t1 tektonról t5 tektonra * Gombász3 : Spóraszórás t2 tektonról t3 tektonra   A before és az aftert, a kimentett txt-k alapján komparáljuk, minden rovarnak megnézzük, hogy megkapta -e a neki való effektet, illetve a pozíciójukat, illetve, hogy a fonal meg a spóra le került -e jó helyre. |
| **Teszt célja** | Leteszteli a rovaroknak a mozgását illetve mind az 5 spórafajta megevését, emellett, leteszteli még egy-egy fejletlen gombatest fonálnövesztését illetve spóralövését. |

| **Teszt-eset neve** | Teszt\_3 |
| --- | --- |
| **Rövid leírás** | Ez a teszteset a játék egy közbenső körét mutatja be, ahol van 1 gombász és 1 rovarász. A teszt a gombász spóraszórására és fonál elágaztatására koncentrál.   * Gombász: spóraszórás szomszédos tektonra (t1) fejletlen gombatestből * Gombász: fonálelágaztatás szomszédos tektonra (t1) * Gombász: fonálelágaztatás az előző tekton szomszédjára (t2). Ez azért lehetséges, mert t1-en volt a gombásznak spórája. Egyébként nem lehetne ugyanabban a körben kétszer elágaztatni fonalat egy gombásznak * Rovarász: rovar mozgatása t1 tektonra, ahol megeszi a gombász egy kilőtt spóráját, ez most ebben a tesztesetben egy lassítóspóra * Rovarász: fonal vágása t1 szomszédján: t2-őn. A fonal fajtája ShortLifeThread, így a következő körben fog törlődni a fonál t2 tektonról |
| **Teszt célja** | A gombász bónusz fonálelágaztatás, spóraszórás tesztelése. A rovar mozgásának, spóra evésének, fonál vágásának tesztelése. |

| **Teszt-eset neve** | Teszt\_4 |
| --- | --- |
| **Rövid leírás** | Körváltás, spóragenerálás, tekton törése, gombatest fejlődése:  Ha minden játékos befejezi a lépését, a körszámláló eggyel növekszik. Páros körben minden gombatest generál egy új spórát. Körváltáskor egy tekton kettétörik, megszakítva a rajta levő fonalat és megsemmisülnek a rajta levő spórák. Azok a gombatestek, amik még nem fejlődtek ki és 5 kör óta lehelyezésre kerültek, fejletté válnak. |
| **Teszt célja** | Ellenőrizni a körszámláló léptetését, minden gombatest spóralistája bővül, tekton törésekor a rajta levő gombafonál rész és a spórák megszűnnek. A gombatest állapota fejletlenről fejletté válik. |

| **Teszt-eset neve** | Teszt\_5 |
| --- | --- |
| **Rövid leírás** | Ez a teszteset a játék egy közbenső körét mutatja be, ahol van 1 gombász és 1 rovarász. A tesztben a 4. körben vagyunk:   * Az Absorbing típusú tektonokról ekkor felszívódik az összes fonál * Ebben a körben is ketté törik egy tekton, amely a tektonon listájában az első * A rovar spóravágásából adódóan maradni fog egy olyan fonál, ami nincs bekötve semmihez, de mivel ez KeepThread típusú tektonon van, ezért ez ekkor nem törlődik |
| **Teszt célja** | Az Absobing típusú tektonokról történő fonalaknak felszívódás, és a KeepThread típusú tektonon való bekötettetlen fonál életben tartásának tesztelése. |

* 1. ***Tesztelést támogató segéd- és fordítóprogramok specifikálása***

-

* 1. ***Napló***

| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| --- | --- | --- | --- |
| 2025.03.29 18:00 | 3 óra | Zelch  Szolár  Fejes  Lekli  Toronyi | Feladatok értelmezése |
| 2025.03.30 10:00 | 10 óra | Zelch | Use-case-k kidolgozása, bemeneti nyelv elkészítése, az új változtatások szerint az osztálydiagram átalakítása, szekvenciadiagramok módosítása, új szekvencia diagramok létrehozása, az utolsó négy use case megírása |
| 2025.03.30 20:30 | 3,5 óra | Fejes | Kiosztott use casek megírása, teszt\_3 teszteset megírása. |
| 2025.03.30. | 4 óra | Lekli | Bemeneti nyelv megírása, kiosztott teszt-eset use casek megírása. |
| 2025.03.30. 14:00 | 5,5 óra | Szolár | Kimeneti nyelv megalkotása. shootSpore,  growMushroom és eatInsect parancsokhoz tartozó use case-ek megírása. Dokumentum formázása. |
| 2025.03.31. 08:00 | 5 óra | Toronyi | Kiosztott use-casek megírása, tesztesetek készítése |
| 2025.03.31. 09:30 | 2,5 óra | Fejes | További tesztesetek megírása. |