**7. Prototípus koncepciója**

25 – bandITs

Konzulens:

**Huszerl Gábor**

Csapattagok

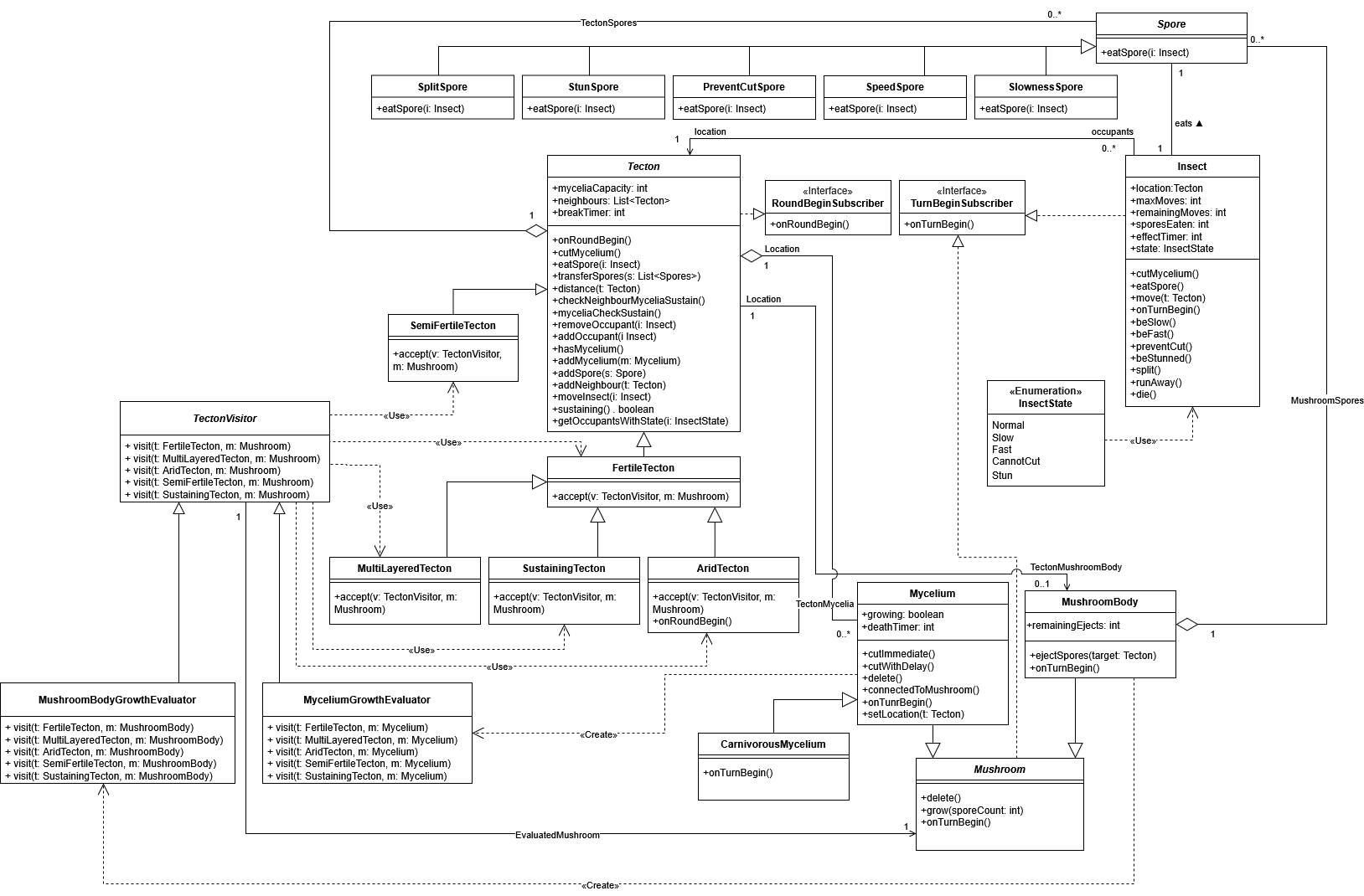
|  |  |  |
| --- | --- | --- |
| Bencze János István | GIWUHT | gomanpc@yahoo.com |
| Guzmics Gergő | VC8OQD | [guzmicsgergo@gmail.com](mailto:guzmicsgergo@gmail.com) |
| Kohár Zsombor | Q8EPW6 | zsombor.kohar@edu.bme.hu |
| Rakos Gergő Máté | I3Q7BY | gergo\_rakos@yahoo.com |
| Dr. Taba Szabolcs Sándor | JRGMBW | taba.szabolcs@gmail.com |

2025.03.31.

# Prototípus koncepciója

## Változás hatása a modell

### Módosult osztálydiagram



### Új vagy megváltozó metódusok

#### Mycelium – cutImmidiate()

Azonnal elvágja a gombafonalat.

#### Mycelium – cutWithDelay()

Egy gombafonál típustól függő idő után elvágja a gombafonalat.

#### Tecton - getOccupantsWithState(i: InsectState)

Azon rovaroknak a listáját hozza létre, amelyek a tektonon helyezkednek el, és adott állapotban vannak.

#### Tecton - sustaining()

Megadja, hogy a tekton, vagy a tektonon elhelyezkedő objektumok képesek-e egy gombafonalat életben tartani.

#### TectonVisitor - visit(t: SustainingTecton, m: Mushroom)

A visitor működésének kiterjesztése az új tekton fajtára.

#### Insect – die()

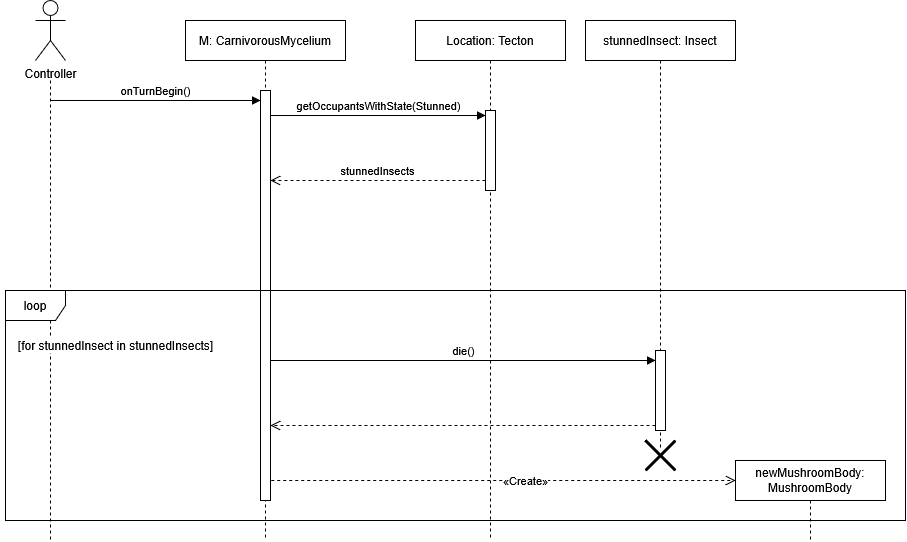
A rovar elpusztul, azaz eltűnik a játékmezőről.

#### Insect – split()

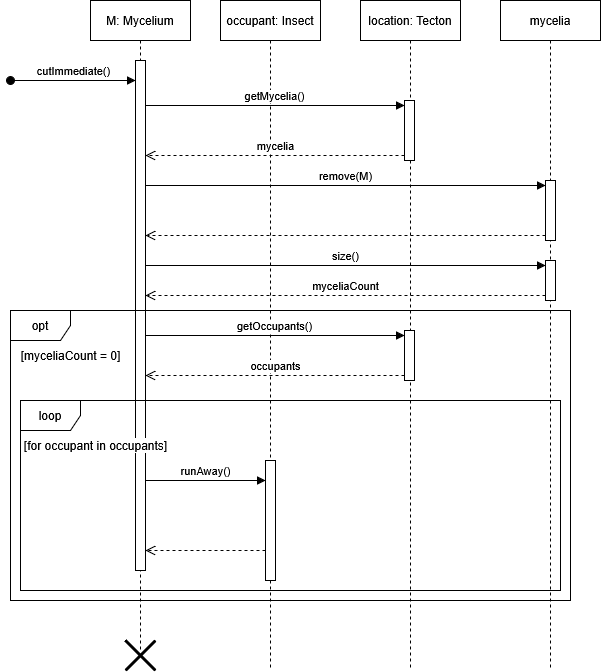
A rovar csinál egy másolatot saját magából.

### Szekvencia-diagramok

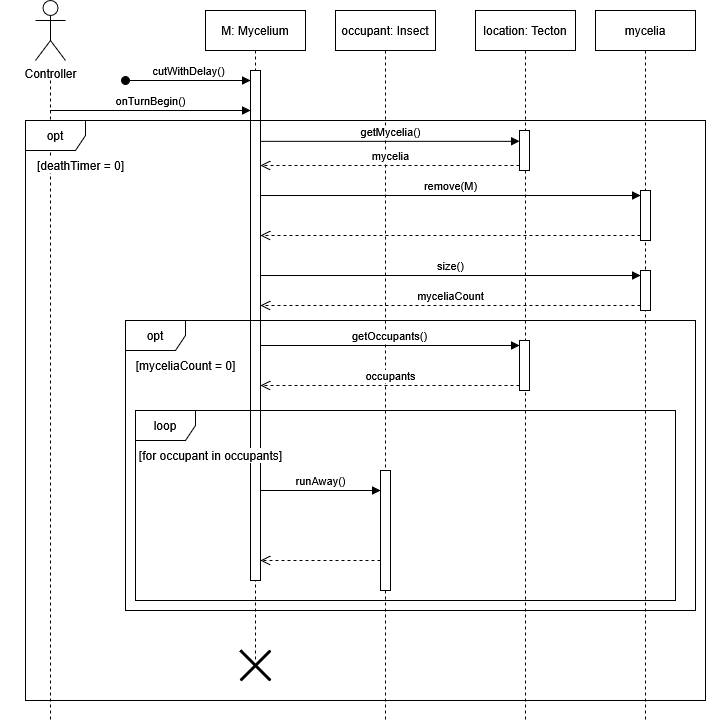
#### Húsevő gombafonál



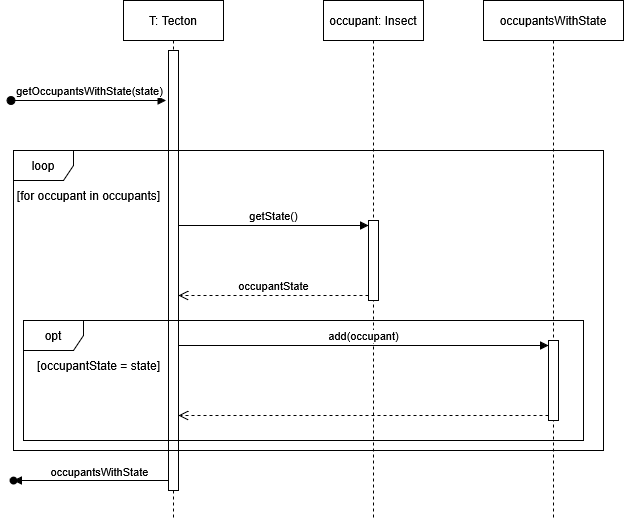
#### Azonnali fonálelvágás



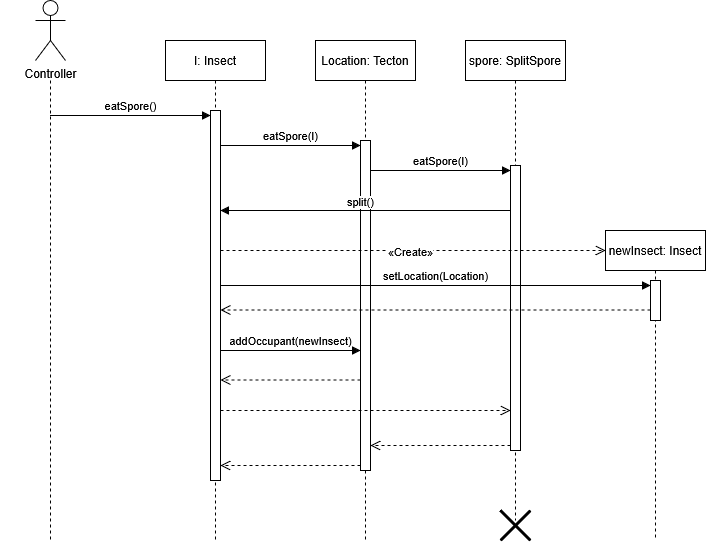
#### Késeltetett fonal elvágás



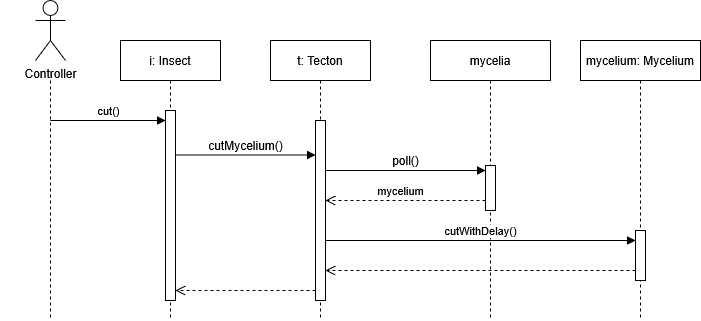
#### getOccupantsWithState szekvencia



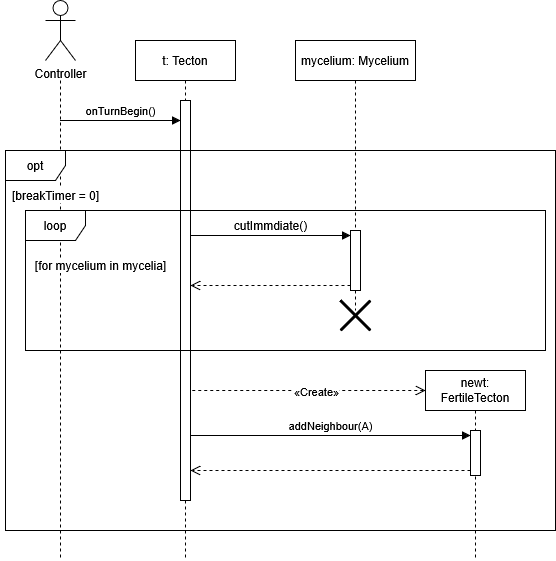
#### „Split” típusú spóra elfogyasztása



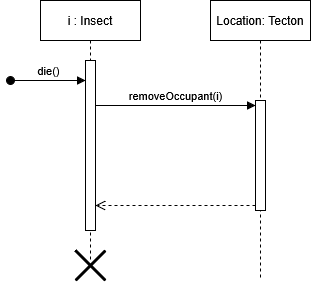
#### Rovar általi gombafonál elvágás



#### Tektontörés



#### Rovar elpusztulása



## Prototípus interface-definíciója

### Az interfész általános leírása

A felhasználó a program irányítását parancsokkal végzi. Ezeknek a parancsoknak vannak argumentumai, melyeket a parancs után tud a játékos beleírni. A parancsok maguktól nem adnak outputot. Ha egy parancs egy részében felhasználói választásra van szükség, akkor a program ki tud kérdezni a felhasználóhoz. A program futása során minden objektumnak van egy saját, egyedi neve, ami alapján az objektum egyértelműen azonosítható.

### Bemeneti nyelv

Egy parancs általános szintaxisa:

<UTASÍTÁS> <PARAMÉTER> {<LISTA\_ELEM\_1> <LISTA\_ELEM\_2> <LISTA\_ELEM\_N>}

Tehát a parancsnak megadjuk a nevét, (utasítás), majd utána, ha szükséges, szóközzel elválasztva a parancs paramétereit megadhatjuk. Ha a parancs paramétere egy gyűjtemény lenne, akkor a gyűjtemény kezdetét és végét kapcsos zárójellel jelöljük. Az egyes elemek szóközzel vannak elválasztva.

Ha egy parancsnak felhasználói inputra van szüksége, akkor azt a program az alábbi módon jelzi:

<< 1:<OPCIÓ\_1> 2:<OPCIÓ\_2> N:<OPCIÓ\_N>

Azaz, a „<<” karakterekkel jelöli, hogy éppen bemenetre van szükség, az opciókat pedig a sorszámukkal együtt kiírja a program. A felhasználó választani úgy tud, hogyha a választott opció sorszámot megadja.

Minden a MACRO\_CASE elnevezési konvenciót használja, azaz minden betű nagybetű, és a szavak alsókötőjel ( \_ ) karakterrel vannak elválasztva.

**A parancsok listája**

Az 1. számú use-case kapcsán

STATE

**Leírás:** A kiválasztott objektum állapotát a megadott formátumban kiírja.

**Opciók:** A kiírandó objektum.

A 2. számú use-case kapcsán

Tectonbreak

**Leírás:** Tectontörés

**Opciók:** Nincsenek paraméterek

A 3. számú use-case kapcsán

CREATE\_TECTON TectonType {Neihgbouring Tectons}

**Leírás:** Új Tecton létrehozása

**Opciók:** Első paraméter eldönti milyen típusu legyen az új Tecton, A következő paraméter egy lista mely tartalmazza a Tectonokat amelyek szomszédjai lesznek az új Tectonnak

A 4. számú use-case kapcsán

SET\_BREAKTIMER number

**Leírás:** Tecton BreakTimerjének beallítása egy adott értékre

**Opciók:** Egyetlen paramétere az a szám ami be lesz állitva mint új BreakTimer

Az 5. számú use-case kapcsán

RUN text\_file

**Leírás:** Játék világ legenerálása/inicializálása

**Opciók:** Nincsenek paraméterek

A 6. számú use-case kapcsán

END\_GAME

**Leírás:** Játék végének kezelése

**Opciók:** Nincsenek paraméterek

A 7. számú use-case kapcsán

SET\_ENDGAMETIMER number

**Leírás:** Az EndgameTimer beállítása

**Opciók:** Egyetlen paramétere egy szám mely az új maradék Round-okat jelöli

A 8. számú use-case kapcsán

END\_TURN

**Leírás:** End turn küldése

**Opciók:** Nincsenek paraméterek

A 9. számú use-case kapcsán

ADD\_PLAYER player\_name player\_type

**Leírás:** Játékosok hozzáadása a játékhoz

**Opciók:** Első paraméter a játékos neve, a második paraméter hogy Gombász vagy Rovarász lesz a játékos

A 10. számú use-case kapcsán

START\_GAME

**Leírás:** Játék indítása

**Opciók:** Nincsenek paraméterek

A 11. számú use-case kapcsán

**CREATE\_MUSHROOMBODY** tecton

**Leírás:** A gombatest a céltektonon létrejön.

**Opciók:** Céltekton (tecton)

A 12. számú use-case kapcsán

**GROW\_MUSHROOMBODY** tecton

**Leírás:** A gombatest létrejön és rákerül a céltektonra.

**Opciók:** Céltekton (tecton)

A 13. számú use-case kapcsán

**PUT\_SPORE** tecton

**Leírás:** Egy spóra rákerül a céltektonra.

**Opciók:** Céltekton (tecton)

A 14. számú use-case kapcsán

**EJECT\_SPORES** mushroombody tecton

**Leírás:** A kiválasztott gombatest valamennyi spórája rákerül a céltektonra.

**Opciók:** Gombatest (mushroombody); céltekton (tecton)

A 15. számú use-case kapcsán

**DEACTIVATE** mushroombody

**Leírás:** A kiválasztott gombatest elpusztul (inaktívvá válik).

**Opciók:** Gombatest (mushroombody)

A 16. számú use-case kapcsán

****Create\_Mycelium****

**Leírás:** Létrehoz egy gombafonalat

**Opciók:** Nincsenek argumentumok

A 17. számú use-case kapcsán

add\_mycelium\_to\_tecton Mycelium Tecton

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

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

A 18. számú use-case kapcsán

Grow\_mycelium Mushroom Tecton

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

**Opciók:** A gombatest vagy gombafonál, amiből növesztünk és a tekton, amin a gombafonál lesz

A 19. számú use-case kapcsán

CREATE\_INSECT tecton

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

**Opciók:** Argumentumok: A céltekton, ahova létrejönne

A 20. számú use-case kapcsán

MOVE insect tecton

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

**Opciók:** Argumentumok: A rovar, amelyik mozogna. A céltekton, ahova mozogna.

A 21. számú use-case kapcsán

EAT insect

**Leírás:** A rovar megeszik egy spórát a tektonján

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

A 22. számú use-case kapcsán

CUT insect

**Leírás:** A rovar elvág egy fonalat a tektonján

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

### Kimeneti nyelv

Egy objektum állapotát a STATE paranccsal lehet elérni. A STATE parancsnak és kimenetelének szintaxisa az alábbi:

STATE <OBJEKTUM>

<OBJEKTUM> <OBJETUM TÍPUSA>

<tagváltozó neve> <TAGVÁLTOZÓ TÍPUSA> = <TAGVÁLTOZÓ ÉRTÉKE>

<lista NEVE> <LISTA TÍPUSA> = {

elem1

elem2

elemN

}

Azaz először az objektum nevét, majd típusát írja ki. Majd az adott objektum alá tartozó tagváltozók egy tabulátorral beljebb kerülnek.

Az egyszerű tagváltozókat úgy írjuk ki, hogy először a nevét, majd típusát írjuk ki, majd egy egyenlőségjel után az értékét.

A gyűjtemény típusú tagváltozókat a gyűjtemény neve, utána a gyűjtemény típusa, majd egy egyenlőségjellel elválasztva az értéke, ahol a lista elejét és végét kapcsos zárjelek jelölik, és minden elem neve fel van sorolva, úgy, hogy mindegyik elem külön sorban van.

## Összes részletes use-case

1. Számú Use case - Állapot kiírása

|  |  |
| --- | --- |
| **Use-case neve** | Állapot kiírása |
| **Rövid leírás** | A felhasználó a kiválasztott objektum állapotát kiíratja. |
| **Aktorok** | User |
| **Forgatókönyv** | 1. Az objektum kiválasztása 2. Az objektum állapota kiíródik. |

2. Számú Use case - Tectontörés

|  |  |
| --- | --- |
| **Use-case neve** | Tectontörés |
| **Rövid leírás** | Az adott FertileTecton eltörik, vagyis a rajta lévő objektumok megsemmisülnek (kivéve a gombatestet) és egy új FertileTecton keletkezik, melynek egyetlen szomszédja az eredeti FertileTecton lesz. Az eredeti FertileTectonnak is beállítjuk az új FertileTectont mint szomszéd.  Hasonlóan történik a törés a többi Tecton típusra is. A törés mindig FertileTectont eredményez az új Tecton típusaként. |
| **Aktorok** | User |
| **Forgatókönyv** | 1. A BreakTimer eléri a 0-at  2. A Tectonrol minden eltörlődik (kivéve a gombatestet)  3.Új Tecton keletkezik  4.Az új Tecton és az eltört Tecton szomszédok lesznek |

3. Számú Use case - Új Tecton létrehozása

|  |  |
| --- | --- |
| **Use-case neve** | Új Tecton létrehozása |
| **Rövid leírás** | A felhasználó paraméterként átadja az összes olyan Tectont amelyeket kívánja hogy szomszédjai legyenek az új Tectonnak. Itt legelső paraméterként azt adja át, hogy milyen típusú Tectonná szeretné csinálni az új Tectont. |
| **Aktorok** | User |
| **Forgatókönyv** | 1. Új Tecton létrehozása az átadott paramétereknek megfelelően  2. Az új Tectonnak beallítódnak a szomszédjai |

4. Számú Use case - Tecton BreakTimerjének beallítása egy adott értékre

|  |  |
| --- | --- |
| **Use-case neve** | Tecton BreakTimerjének beallítása egy adott értékre |
| **Rövid leírás** | A User kiválaszthat egy adott Tectont és annak beállítja a breakTimerjét egy adott értékre. |
| **Aktorok** | User |
| **Forgatókönyv** | 1. A Tecton kiválasztása amelynek módosítanánk a BreakTimerjét  2. A kiválasztott Tecton BreakTimerjének beallítása a paraméterben átadott értékre |

5. Számú Use case - Játék világ legenerálása/inicializálása

|  |  |
| --- | --- |
| **Use-case neve** | Játék világ legenerálása/inicializálása |
| **Rövid leírás** | Megteremti a kezdeti állapotú Tectonokat, Gombatesteket, Gombafonalakat és Insecteket. |
| **Aktorok** | User |
| **Forgatókönyv** | 1. Amikor az applikáció elindul meghívódik a metódus hogy generáljon Tectonokat  2. Sorrendben leteremti a játékmező elemeit (Tectonok, Gombatestek, Gombafonalak, Insectek) |

6. Számú Use case - Játék végének kezelése

|  |  |
| --- | --- |
| **Use-case neve** | Játék végének kezelése |
| **Rövid leírás** | Amikor véget ért az utolsó Round, akkor eldönti a nyertest és bemutatja a végeredményeket/score-okat a user-nek |
| **Aktorok** | User |
| **Forgatókönyv** | 1. Véget ér az utolsó Round  2. Kiszámolódnak a pontszámok  3. Kiíródik a végeredmény/ki mennyi pontot ért el |

7. Számú Use case - Az EndgameTimer beállítása

|  |  |
| --- | --- |
| **Use-case neve** | Az EndgameTimer beállítása |
| **Rövid leírás** | A User beállítja a hátralévő Round-ok számát |
| **Aktorok** | User |
| **Forgatókönyv** | 1. Az EndgameTimer beállítása az új értékre |

8. Számú Use case - End turn küldése

|  |  |
| --- | --- |
| **Use-case neve** | End turn küldése |
| **Rövid leírás** | Akkor történik mikor az egyik játékos úgy gondolja, hogy mostmár befejezi a lépéseit és átadja a Turn-jét. Így elkezdődhet a következő játékos Turn-je. |
| **Aktorok** | User |
| **Forgatókönyv** | 1. A User kiküldi a parancsot, hogy vége van a Turn-jének  2. Végrehajtódnak a Turn-ök közötti események  3. Elkezdődik a sorban következő User/Játékos Turn-je |

9. Számú Use case - Játékosok hozzáadása a játékhoz

|  |  |
| --- | --- |
| **Use-case neve** | Játékosok hozzáadása a játékhoz |
| **Rövid leírás** | Miután ki lett választva, hogy hányan szeretnének játszani. A rendszer kiválasztja, hogy ki mi legyen (Gombász vagy Rovarász) |
| **Aktorok** | User |
| **Forgatókönyv** | 1. Valaki megnyitja a játékot  2. Kiválasztódik hány játékos szeretne játszani  3. A rendszer kiossza a játékosokat szerep szerint (Gombász vagy Rovarász) |

10. Számú Use case - Játék indítása

|  |  |
| --- | --- |
| **Use-case neve** | Játék indítása |
| **Rövid leírás** | Miután ki lettek választva a játékosok és a játék világ legenerálódott, a rendszer sorrendbe rakja a játékosokat majd az első játékosnak a sorból átadja az uralmat. |
| **Aktorok** | User |
| **Forgatókönyv** | 1. Meghívódik a játék elindítása  2. A rendszer sorrendbe rakja a játékosokat  3. A rendszer átadja az uralmat a sorban első játékosnak |

11. számú use case – Gombatest létrehozása

|  |  |
| --- | --- |
| **Use case neve** | Gombatest létrehozása |
| **Rövid leírás** | Létrejön egy gombatest a kiválasztott tektonon (céltekton). |
| **Aktorok** | Gombász |
| **Forgatókönyv** | 1. Gombatest – a szükséges feltételek fennállása esetén – a céltektonon létrejön. |

12. számú use case – Gombatest növesztése

|  |  |
| --- | --- |
| **Use case neve** | Gombatest növesztése |
| **Rövid leírás** | A gombász olyan játékutasítást ad, hogy jöjjön létre egy gombatest egy általa kiválasztott tektonon (céltekton). |
| **Aktorok** | Gombász |
| **Forgatókönyv** | 1. Gombatest létrehozása a céltektonon. 2. Feltételek vizsgálata. 3. A létrehozott gombatest a céltektonon véglegesen elhelyezésre kerül. |

13. számú use case – Spóra elhelyezése egy tektonra

|  |  |
| --- | --- |
| **Use case neve** | Spóra elhelyezése egy tektonra |
| **Rövid leírás** | Egy tektonon (céltekton) spóra kerül elhelyezésre. |
| **Aktorok** | Gombász |
| **Forgatókönyv** | 1. Céltekton kiválasztása. 2. Spóra típusának meghatározása. 3. Céltekton nyilvántartásba veszi a spórát. |

14. számú use case – Gombatest spórakilövése

|  |  |
| --- | --- |
| **Use case neve** | Gombatest spórakilövése |
| **Rövid leírás** | A gombász olyan játékutasítást ad, hogy az általa kiválasztott gombatest lője ki az összes spóráját egy, a gombász által kiválasztott tektonra (céltekton). |
| **Aktorok** | Gombász |
| **Forgatókönyv** | 1. Gombatest kiválasztása. 2. Céltekton kiválasztása. 3. A gombatest a céltektonra kilövi a spórákat. |

15. számú use case – Gombatest elpusztulása (inaktívvá válás)

|  |  |
| --- | --- |
| **Use case neve** | Gombatest elpusztulása (inaktívvá válás) |
| **Rövid leírás** | A gombász a gombatest elpusztulására (inaktívvá válás) vonatkozó játékutasítást ad. |
| **Aktorok** | Gombász |
| **Forgatókönyv** | 1. Gombatest kiválasztása. 2. A gombatest elpusztul (inaktívvá válik). |

16. számú use case – Gombafonál létrehozása

|  |  |
| --- | --- |
| **Use-case neve** | Gombafonál létrehozása |
| **Rövid leírás** | A játékos létrehoz egy gombafonalat. |
| **Aktorok** | Gombász |
| **Forgatókönyv** | 1. Létrejön egy gombafonál |

17. számú use case – Gombafonál hozzáadása tektonhoz

|  |  |
| --- | --- |
| **Use-case neve** | Gombafonál hozzáadása tektonhoz |
| **Rövid leírás** | Hozzáadja a kiválasztott fonalat a kiválasztott tektonhoz |
| **Aktorok** | Gombász |
| **Forgatókönyv** | 1. A kiválasztott fonal rákerül a kiválasztott tektonra |

18. számú use case – Gombafonál növesztése

|  |  |
| --- | --- |
| **Use-case neve** | Gombafonál növesztése |
| **Rövid leírás** | Gombafonál növesztése a kiválasztott tektonra. |
| **Aktorok** | Gombász |
| **Forgatókönyv** | 1. A gombafonál megvizsgáltatja a kiválasztott tektonnal, hogy képes-e nőni rá 2. A fonál véglegesen létrejön a tektonon |

19. számú use case – Rovar létrehozása és letevése

|  |  |
| --- | --- |
| **Use-case neve** | Rovar létrehozása és letevése |
| **Rövid leírás** | Rovar létrehozódik, és rákerül egy tektonra |
| **Aktorok** | Rovarász |
| **Forgatókönyv** | 1. Ha olyan tektonra hoznánk létre ahol tud lenni (van az adott tektonon Mycelium): 2. Rovar létrehozása 3. Rovar helyének beállítása 4. Location-tektonra rovar rátevése |

20. számú use case – Rovar mozgatása

|  |  |
| --- | --- |
| **Use-case neve** | Rovar mozgatása |
| **Rövid leírás** | Rovar mozgása egyik tektonról másikra |
| **Aktorok** | Rovarász |
| **Forgatókönyv** | 1. Rovar megnézi, hogy tud-e a céltektonra menni   Ha sikerül   1. Régi tektonjáról rovar leszedése 2. Rovar tektonjának átállítása a régiről a céltektonra 3. Rovar hozzáadása a céltektonra 4. Csökken egyel a tevékenységeinek száma a körben |

21. számú use case – Rovar általi spóraevés

|  |  |
| --- | --- |
| **Use-case neve** | Rovar általi spóraevés |
| **Rövid leírás** | A rovar megeszik egy spórát a tektonjáról |
| **Aktorok** | Rovarász |
| **Forgatókönyv** | 1. Megnézi hogy van-e a rovar tektonján spóra   Ha van spóra a tektonon   1. A rovar megeszik egy spórát a tektonjáról 2. A tektonról eltűnik az a spóra 3. A spóra beállítja a típusának megfelelő hatást a rovaron 4. Csökken egyel a tevékenységeinek száma a körben |

22. számú use case – Rovar általi gombafonál elvágás

|  |  |
| --- | --- |
| **Use-case neve** | Rovar általi gombafonál elvágás |
| **Rövid leírás** | A rovar elvág egy fonalat a tektonján |
| **Aktorok** | Rovarász |
| **Forgatókönyv** | 1. A rovar elvág egy fonalat a tektonján 2. Ha az utolsó fonalat vágta el, elmenekül 3. Csökken egyel a tevékenységeinek száma a körben |

## Tesztelési terv

|  |  |
| --- | --- |
| **Teszt-eset neve** |  |
| **Rövid leírás** |  |
| **Teszt célja** |  |

## Tesztelést támogató segéd- és fordítóprogramok specifikálása

[Rövid bemutatással (elvárt funkcionalitás) specifikálni kell a tesztelést támogató segédprogramokat.]

A teszteléshez JUnit 5[[1]](#footnote-1) test framework-öt fogunk használni.

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2025.03.27. 12:00 | 2 óra | Kohár | Tevékenység:   * Az új funkciók implementálása az osztály diagrammba * Az új funkciók szekvencia diagrammjának elkészítése |
| 2025.03.27. 22:00 | 1 óra | Kohár | Tevékenység:   * Osztálydiagramm és szekvencia diagrammok javítása. |
| 2025.03.28. 20:00 | 1 óra | Kohár | Tevekénység:   * Kimeneti nyelv definiálása * Osztálydiagrammok exportálása |
| 2025. 03.29. 19:00 | 30 perc | Kohár | Tevékenység:   * Osztálykatalógus átnézése |
| 2025. 03.30. 20:00 | 5 perc | Kohár | Tevékenység:   * Teszt támogató segédprogram definiálása. |

1. https://junit.org/junit5/ [↑](#footnote-ref-1)