**8. Részletes tervek**

5. – Pandora

Konzulens:

Elekes Márton

Csapattagok

|  |  |  |
| --- | --- | --- |
| Szabó András | GA9BCR | sbasiliscus@gmail.com |
| Katona Benedek | LNU506 | kbeniboy2004@gmail.com |
| Lipóczi Levente László | SUSXSW | lipoczi.levente@gmail.com |
| Fekete Botond | VZ6YKQ | botond.fekete.010@gmail.com |
| Kőműves Marcell | LLDPZI | kommarci03@gmail.com |

2025.04.13

# **8.** **Részletes tervek****8.1** **Osztályok és metódusok tervei.**

### <<interface>> View

* **Felelősség**

Felhasználói input kérése, a játékállás megjelenítése.

* **Metódusok**
  + **+ getInput(): void:** A Controller hívja, visszaad egyetlen utasítást, amit utána a Controller értelmezhet: hívja a handleInput() metódust, átadja neki az utasítást.
  + **+ readInputFile(f: File): void:** Végig olvassa a fájlt, minden sora után hívja a Controller handleInput() metódusát, átadja neki az olvasott sort.
  + **+ serveResult(msg: String): void**: A Controller hívja. Megjeleníti a paraméterként kapott üzenetet.
  + **+ displayGameBoard(gameBoard: GameBoard): void**: A Controller hívja. Lekérdezi a gameBoard-tól a játéktér elemeit, ezek alapján összeállítja a megjelenítendő állapotot.

### <<enumeration>> State

* **Felvehető értékek:** 
  + ARRANGE
  + ACT
  + ASSERT

### <<enumeration>> ArrangeSection

* **Felvehető értékek:** 
  + TEKTONS
  + NEIGHBOURS
  + SHROOMERS
  + MUSHROOMS
  + BUGGERS
  + STRATEGIES
  + BUGS
  + HYPAS
  + SPORES

### *Controller*

* **Felelősség**

A játék inicializálása, ehhez a View-tól inputok kérése. A játékmenet levezénylése, ehhez akció lehetőségek ciklikus kiosztása a játékosoknak, input kérés a View-tól, annak validálása a soron következő játékos függvényében. Minden körben az utolsó játékos után random tekton törésének indítása, és tektonoknak az endOfRound() metódus meghívása, majd a View hívása, ami megjeleníti a frissített játékállást.

* **Attribútumok**
  + **- actualPlayer: int**: tárolja, hogy hányadik játékos van soron
  + **- gameBoard: GameBoard:** referencia a játéktáblára
  + **- view: View:** referencia a nézet osztályra
  + **-arrangeSection: ArrangeSection:** tárolja az arrange állapotban, hogy milyen objektumok példányosítása (vagy tektonok szomszédságának beállítása) fog a következő sorokban jönni.
* **Metódusok**

#### **+ Controller()**: Konstruktor, példányosítja a view és a gameBoard objektumokat.

* + **+ handleInput(input: String):** A soron következő játékos és/vagy az állapot kontextusában értelmezi a kapott parancsot. Ha a kapott input valamilyen ArrangeSection enumja, akkor az objektum arrangeSection állapotát beállítja arra. Ha az input a kontextust tekintve hibás (szintaktikai vagy nem értelmes), a view serveResult() metódusát használva hibát jelez.
  + **- initMap(): void**: inicializálja a játékot.
  + **- gameCycle(): void**: futtatja a játékot.
  + - **endOfRound(): void**: végig megy az összes Tektonon, és meghívja a Tekton.endOfRound() metódust. Végig megy az összes Shroomer objektumon és meghívja az endOfRoundAdministration() metódusát.
  + **+ run()**: elindítja a működést: először meghívja az initMap() függvényt, majd a gameCycle-t.

### ControllAutoTest

#### **Felelősség**

Konkrét tesztesetek futtatása fájlból. Képes egy teszt esetet (névvel megadva), vagy az összes teszt esetet lefuttatni a teszt mappából.

#### **Ősosztályok**

Controller

* **Attribútumok**
  + **- tc: String**: tárolja, hogy melyik tesztesetet futtatjuk. Értéke alapértelmezetten null,
    - ha *-a* flaggel indítjuk az alkalmazást, vagyis minden teszt esetet le akarjuk futtatni, null is marad
    - ha -*[teszteset neve]* flaggel indítjuk az alkalmazást, a futtatandó teszteset neve kerül bele

#### **Metódusok**

* + **+ ControllAutoTest(testCase: String):** konstruktor. Hívja az ős konstruktorát, ezután ha a testCase paraméternek null-t kell adni híváskor, ha minden tesztesetet le szeretnénk futtatni, egyébként a *-[teszteset neve]* flag értékét kell kapnia. a tc tagváltozó értékét beállítja a testCase paraméter értékére.
  + **+ run():** 
    - ha a tc string értéke null, akkor a test mappában található minden teszt eseten végig megy, és mindegyikre meghívja az initMap metódust, majd a gameCycle metódust, majd assert metódust.
    - Ha a tc értéke nem null, akkor csak azon egy tesztesetre meghívja az initMap majd a gameCycle metódust, majd assert metódust.
  + **- initMap(testCase: string): void**: Inicializál egy pályát a kapott tesztesethez tartozó Arrange nyelven megadott input fájlból, oly módon, hogy hívja a view readInputFile metódusát, paraméterként átadva neki az adott tesztesethez tartozó arrange fájlból készített File objektumot. A handleInput metódus az inputokat kiértékeli: beállítja az arrangeSection értékét, vagy példányosít objektumot (vagy tektonok szomszédságát állítja be).
  + **- gamecycle(testCase: string): void**: Elindítja a játék folyamát. Meghívja a view readInputFile metódusát, paraméterként átadva az adott tesztesethez tartozó act fájlt. A handleInput metódus az inputokat kiértékeli, és sikeres validáció után a parancs hatását végre is hajtja: a modellbeli objektum megfelelő akcióját indítja.  
     Ha az actualPlayer a gameBoard buggers HashMap-jében van, akkor meghívjra rá az endOfTurn() metódust.  
     Ha az actualPlayer értéke az utolsó játékos a körben, akkor hívja a saját endOfRound() metódusát és az actualPlayer értékét növeli modulo a játékosok száma.
  + **- assert(testCase: string): void:** Tesztesetet kiértékeli. Meghívja a view readInputFile metódusát, paraméterként átadva az adott tesztesethez tartozó assert fájlt. A handleInput metódus az inputokat kiértékeli. Meghívja a view serveResult metódusát paraméterként átadva a kiértékelés eredményét (“SUCCESS”/”FAIL”/”ERROR”).

### ControllManualTest

#### **Felelősség**

Manuálisan futtatható tesztek vezérlése.

#### **Ősosztályok**

Controller

#### **Attribútumok**

* + **- state: State**: tárolja, hogy milyen állapotban van az objektum. Ettől (is) függ, hogy milyen kontextusban értelmezi a bemenetet a handleInput metódus.

#### **Metódusok**

* + **+ControllManualTest(): void:** Konstruktor. Hívja az ős konstruktorát.
  + **+ run():** Futtatja a manuális tesztelést. Először meghívja az initMap metódust, majd a gameCycle-t és végül az assert metódust
  + **+ initMap(): void:** Belépéskor beállítja a State-t ARRANGE-re. Egy While loopban addig hívja a view getInput() metódusát, amíg a State “ARRANGE”. A felhasználó az Arrange nyelven kell megadja a pálya kialakítását. Az inputokat a handleInput() metódus folyamatosan kezeli: beállítja az arrangeSection értékét, vagy példányosít objektumot (vagy tektonok szomszédságát állítja be).
  + **+ gamecycle(): void**: Egy while loopban futtatja a játékot, ahol a View objektum getInput() metódusát hívja, ahol a user az ACT nyelv parancsait adhatja meg, amik alapján változik a model állapota. A handleInput() metódus az inputokat kiértékeli, és sikeres validáció után a parancs hatását végre is hajtja: a modellbeli objektum megfelelő akcióját indítja.  
    Ha az actualPlayer a gameBoard buggers HashMap-jében van, akkor meghívjra rá az endOfTurn() metódust.  
     Ha az actualPlayer értéke az utolsó játékos a körben, akkor hívja a saját endOfRound() metódusát és az actualPlayer értékét növeli modulo a játékosok száma.
  + **+ handleInput(input: String): void:** Ugyanaz a működés, mint a Controller ősosztályban, plusz ha a kapott input ARRANGE, ACT vagy ASSERT (állapotváltó!) parancs, az objektum állapotát ARRANGE, ACT vagy ASSERT-re állítja. ARRANGE módból való kilépés után nem léphet vissza oda a user.

### ControllGame

#### **Felelősség**

A valós játék futtatása éles módban.

#### **Ősosztályok**

Controller

* **Attribútumok**
  + **- round: int**: tárolja, hogy hányadik körnél tart a játék.
  + **-state: State:** játék fázis tárolása

#### **Metódusok**

* + **+ ControllGame():** Konstruktor. Hívja az ős konstruktorát és beállítja a state tagváltozót ARRANGE-re.
  + **+ run():** futtatja az initMap metódust, majd a gameCycle metódust.
  + **- initMap(): void**: Inicializál egy pályát (lsd.: pszeudo).   
    Ciklikusan a view serveResult(msg: string) metódussal felajánlja a usernek, hogy Játék indítás vagy Új Bogarász, vagy Új Gombász hozzáadását akar-e. A view getInput() metódusát meghívja, majd a handleInputban validálja. Választás függvényében létrehoz egy Shroomer vagy egy Bugger objektumot vagy elindítja gamecycle() metódust, ez utóbbi esetben a state állapotát ACT-ra állítja.   
    Ha a ciklus 8 játékost már hozzáadott, akkor is a state állapotát ACT-ra állítja, ily módon több játékos hozzáadását már nem engedi.

ciklus k fut 1-től 25-ig

generál egy r véletlen számot 0 és 1 között.

ha r<0,48, akkor Tekton létrehozása, és listához adása

ha r<0,61, akkor Peat létrehozása, és listához adása

ha r<0,74, akkor Stone létrehozása, és listához adása

ha r<0,87, akkor Swamp létrehozása, és listához adása

különben Soil létrehozása, és listához adása  
ciklus vége  
ciklus i fut 1-től 25-ig

ciklus j fut 1-től 25-ig  
 ha i ≠ j, és az i-edik és j-edik Tekton nem szomszéd, akkor  
 generál egy r véletlen számot 0 és 1 között.  
 ha r<0,035, akkor  
 i-edik és j-edik Tektont beállítja szomszédnak

ciklus vége  
ciklus vége  
ciklus: minden Shroomer objektumon

ciklus:  
 generál egy r véletlen számot 0 és 24 között  
 ha r-edik Tekton nem Stone és nincs rajta gombatest vagy bogár,  
 akkor  
 létrehoz az adott Shroomernek egy Mushroom leszármazott  
 objektumot az r-edik Tektonra. ciklus stop

ciklus vége  
ciklus vége  
ciklus: minden Bugger objektumon

ciklus:  
 generál egy r véletlen számot 0 és 24 között  
 ha r-edik Tektonon nincs gombatest vagy bogár  
 létrehoz az adott Buggernek egy Bug objektumot az r-edik  
 Tektonra. ciklus stop

ciklus vége  
ciklus vége

ciklus i fut 1-től 25-ig

Az i-edik Tektont hozzáadja a GameBoard-hoz az addTekton(t: Tekton)  
 metódussal

ciklus vége

* + **gamecycle(): void**: A state tagváltozót ACT-re állítja. Egy while loopban futtatja a játékot, ahol a View objektum getInput() metódusát hívja, ahol a user az ACT nyelv parancsait adhatja meg, amik alapján változik a model állapota. A handleInput metódus az inputokat kiértékeli, és sikeres validáció után a parancs hatását végre is hajtja: a modellbeli objektum megfelelő akcióját indítja.  
    Ha az actualPlayer a gameBoard buggers HashMap-jében van, akkor meghívjra rá az endOfTurn() metódust.  
     Ha az actualPlayer értéke az utolsó játékos a körben, akkor hívja az endOfRound() metódust, véletlenszerűen kiválaszt pontosan egy olyan Tektont amelyen nincsen Mushroom objektum, erre meghívja a Tekton.break() metódust. Az actualPlayer értékét növeli modulo a játékosok száma.
  + **+ handleInput(input: String): void:** Ha a state állapota ARRANGE, akkor csak a az bogarász/gombász hozzáadása parancsot fogadja el, meg a start game parancsot. Ha a state állapota ACT, akkor azonos act parancsokat fogad el mint az ősosztályban, amennyiben ez sikeres, hívja a view displayGameBoard() metódusát.

### GameBoard

* **Felelősség**

A játékban résztvevő játékosok, tektonok nyilvántartása, a modell összefogása oly módon, hogy a Controller tudja vezérelni és a View meg tudja jeleníteni az aktuális játékállást.

* **Attribútumok**
  + **- allTektons: List<Tekton>**: a játékban szereplő összes tektonra referencia
  + **- shroomers: HashMap<Int, Shroomer>**: játékban szereplő összes Shroomer objektumra referencia, amelyhez egy sorszám tartozik, hogy hányadik játékos a körben
  + **- buggers: HashMap<Int, Bugger>**: játékban szereplő összes Bugger objektumra referencia, amelyhez egy sorszám tartozik, hogy hányadik játékos a körben
  + - **objectNameMap: Map<Object, String>:** a játékban szereplő összes objektumra referencia párosítva egy egyedi névvel
* **Metódusok**

#### **+ GameBoard()**: konstruktor, inicializálja a játék elemeit tároló Map-eket.

* + **+ addShroomer(s: Shroomer): void:** egy Shroomer játékost lehet vele a játékhoz adni, paraméterkémt ezt a Shroomert várja.
  + **+getShroomers(): HashMap<Int, Shroomer>** visszatér a shroomers map-pal
  + **+ addBugger(b: Bugger): void:** egy Bugger játékost lehet vele a játékhoz adni, paraméterként ezt a Buggert várja
  + **+getBuggers(): HashMap<Int, Bugger>** visszatér a buggers map-pal
  + **+ addTekton(t: Tekton): void:** A pályához adja a paraméterként kapott Tektont (beleteszi az allTektons listába)
  + **+ getTektons(): List<Tekton>:** visszaadja a tektonokat tartalmazó listát
  + **+ getObjectNameByReference(ref: Object): string:** bármilyen játékbeli objektumot megadva paraméterként visszaadja annak az objektumnak a nevét

### *Player*

* **Felelősség**

A Player absztrakt osztály tárolja az egyes játékosok pontszámát

* **Attribútumok**
  + **- score: int**: a játékos pontszáma
* **Metódusok**
  + **+ increaseScore(howMuch: int): void**: növeli a *score* értékét a kapott egész számmal
  + **+ skip(): bool**: a játékos ezt a metódust hívva tudja átugrani a körét.

### Bugger

#### **Felelősség**

A Bugger osztály asszociációkkal tárolja a hozzá tartozó Bug példányokat. Bug példányok hozzáadhatók és törölhetők az asszociációk közül. A Bugger osztály egy kiválasztott Buge akcióit indítja.

#### **Ősosztályok**

Player

#### **Attribútumok**

#### **- bugs: List<Bug>**: A Buggerhez tartozó Bug példányok referenciái

#### **Metódusok**

#### **+ Bugger()**: konstruktor, inicializálja a bugs listát

#### **+ addBug(b: Bug): void**: A *bugs* listába teszi a paraméterként kapott *b* Bug példányt

#### **+ removeBug(b: Bug): void**: Kiveszi a *bugs* listából a paraméterként kapott *b* Bug példányt

#### **+ move(b: Bug, to: Tekton): bool**: A paraméterként kapott *b* Bug példány move(to: Tekton) metódusát hívja, a *to* paramétert átadva. Az akció sikerességével tér vissza.

#### **+ bite(b: Bug, h: Hypa): bool:** A paraméterként kapott *b* Bug példány bite(h: Hypa) metódusát hívja, a *h* paramétert átadva. Az akció sikerességével tér vissza.

#### **+ eat(b: Bug, s: Spore): bool:** A paraméterként kapott *b* Bug példány eat(s: Spore) metódusát hívja, az *s* paramétert átadva. Az akció sikerességével tér vissza.

#### **+ endOfTurn(): void**: végigmegy a *bugs* listán és minden elemének meghívja az endOfTurn() metódusát. Ezt a metódust a játékot vezérlő osztálynak kell hívni.

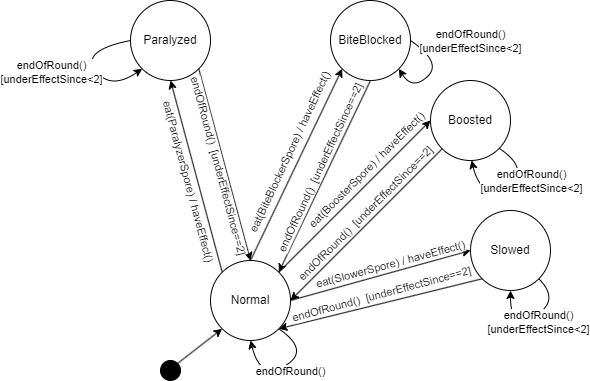
* + **+ getBugs(): List<Bug>:**

visszaadja a *bugs* listát

### Bug

#### **Felelősség**

A Bug osztály valósítja meg a játékban részt vevő rovarokat. Metódusaival az aktuálisan rá ható spóra hatástól függően tud fonál mentén mozogni, fonalat elharapni, vagy spórát enni. Tárolja, hogy mióta tart az aktuális hatás, ami alatt áll. Az állapota maga a hatás, ami alatt áll.  
Állapotgép:



#### **Attribútumok**

#### **- tekton: Tekton**: az aktuális pozíciót tárolja

#### **- strategy: Strategy:** referencia a hatást megvalósító Strategy interface-t megvalósító osztály példányára

* + **- bugger: Bugger**: referencia a “gazda” Bugger példányra
  + **- underEffectSince: int**: tárolja, hogy az aktuálisan beállított Strategy mióta hat

#### **Metódusok**

#### **+ Bug(bugger: Bugger):** konstruktor, beállítja a bugger attribútum értékét a paraméterként kapottra, a tekton attribútumot null-ra, a strategy-nek példányosít egy új Normal stratégiát és beállítja.

#### **+ getBugger(): Bugger**: visszatér a bugger attribútummal, vagyis a gazda referenciájával

* + **+ setStrategy(s: Strategy): void**: a strategy attribútumot beállítja a paraméterként kapott értékre
  + **+ getLocation(): Tekton**: visszaadja a *tekton* attribútumot, a referenciát az aktuális pozíció tektonra
  + **+ setLocation(t: Tekton): void**: beállítja a *tekton* attribútumot a paraméterként kapottra
  + **+ getUnderEffectSince(): int**: visszaadja az *underEffectSince* tagváltozó értékét
  + **+ increaseUnderEffectSince(): void**: növeli egyel az *underEffectSince* tagváltozó értékét
  + **+ move(to: Tekton): bool**: A strategy move(b: Bug, to: Tekton) metódusát hívja, annak visszatérési értékét visszaadja. Magát a viselkedést a stratégia hajtja végre.
  + **+ bite(h: Hypa): bool**: A strategy bite(b: Bug, h: Hypa) metódusát hívja, annak visszatérési értékét adja vissza. Magát a viselkedést a stratégia hajtja végre.
  + **+ eat(s: Spore): bool**: A strategy eat(b: Bug, s: Spore) metódusát hívja, a visszatérési értékét adja vissza. A viselkedést a stratégia végzi el.
  + **+ beEaten(h: Hypa): bool:** a strategy eatenByHypa(b: Bug, h: Hypa) metódusát hívja, ennek visszatérési értékével (igaz/hamis) tér vissza. A bogárra vonatkozó viselkedést (halál vagy semmi) a stratégia végzi el.
  + **+ endOfTurn(): void**: hívja a strategy endOfTurn(b: Bug) metódusát magát átadva paraméterként

### <<interface>> Strategy

* **Felelősség**

A Strategy interface-t biztosít a hatásokat megvalósító osztályoknak, ezt kell megvalósítaniuk. Metódusai (eat, move, bite) “validálják” a Bug lépéseit, az adott hatást megvalósító osztályban a hatásnak megfelelően kell override-olni őket.

* **Metódusok:**
  + **+ eat(b: Bug, s: Spore): bool**: Meghatározza, hogy a Bug tud-e enni az adott körben, ha igen, el is végzi az akciót. Ha el tudja végezni az akciót, igazzal, különben hamissal tér vissza.
  + **+ bite(b: Bug, h: Hypa): bool**: Meghatározza, hogy a Bug meg tudja-e harapni a megadott Hypa példányt, ha igen, el is végzi az akciót. Ha el tudja végezni az akciót, igazzal, különben hamissal tér vissza.
  + **+ move(b: Bug, to: Tekton): bool**: Meghatározza, hogy a Bug át tud-e mozogni egy másik Tektonra, ha igen, el is végzi az akciót. Ha el tudja végezni az akciót, igazzal, különben hamissal tér vissza.
  + **+ canBeEaten(): bool**: Meghatározza, hogy a Bug-ot meg tudja-e enni egy Hypa, ha igen, el is végzi az akciót. Ha el tudja végezni az akciót, igazzal, különben hamissal tér vissza.
  + **+ endOfTurn(b: Bug): void**: A kör végén az adott hatás lejár-e, ha igen, vissza kell állítani a b Bug stratégiáját normálra.

### Normal

#### **Felelősség**

A Normal osztály valósítja meg azt a viselkedést, amikor a rovaron nincs spóra hatás.

#### **Interfészek**

Strategy

#### **Metódusok**

#### **+ eat(b: Bug, s: Spore): bool**: Mivel a Bug nem áll spóra hatása alatt, ezért ha a kapott spóra a Bug pozíció tektonján található, a rovarra hattatja a spóra hatását, növeli a rovarász pontszámát és a tektonról eltűnteti a megevett spórát, majd igazat ad vissza, egyébként hamisat. Pszeudokód: HA s b location-jén van, akkor a spóra hattatása b-re b Bugger-jének pontszámának növelése a spóra tápanyagtartalmával b location-jéről s eltávolítása b underEffectSince tagváltozójának 0-ra állítása visszatér igaz értélkkel különben: visszatér hamis értékkel

* + **+ bite(b: Bug, h: Hypa): bool**: Meghatározza, hogy a Bug meg tudja-e harapni a megadott Hypa példányt. Mivel spóra hatás hiánya nem gátolja a fonál harapást, ezért ha a kapott fonál még nem haldoklik harapás következtében és a b Bug pozíciójához (Tekton) csatlakozik, az isDyingSinceBitten tagváltozóját 0-ra állítja és igazat ad vissza, egyébként hamisat.  
    Pszeudokód:  
      
    HA a kiválasztott Hypa haldoklik harapás miatt, akkor   
     visszatér hamis értékkel  
    Lekérdezi a bogártól a location Tekton értékét, majd a locationtól a hypas Hypa listáját.  
    Ha ebben a listában benne a h Hypa, akkor  
     h isDyingSinceBitten tagváltozóját 0-ra állítja  
     visszatér igaz értékkel  
    különben:  
     visszatér hamis értékkel
  + **+ move(b: Bug, to: Tekton): bool**: Meghatározza, hogy a Bug át tud-e mozogni egy másik Tektonra. Mivel spóra hatás hiányában egy fonál menti első szomszédra lehet lépni, ezért ha a kapott tekton fonál mentén szomszédja a b Bug pozíciójának, megpróbálja rátenni a to-ra b-t. Ha ez sikerül, igazat ad vissza, egyébként hamisat.  
    Pszeudokód:  
      
    Lekérdezi a bogártól a location Tekton értékét, majd a locationtól a a hypák menti szomszédos Tektonoklistáját.  
    Ha ebben a listában benne a to Tekton és sikeresen ráhelyezte b-t,akkor  
     b előző location-jének bug referenciáját null-ra állítja  
     b location-jét to-ra állítja  
     visszatér igaz értékkel  
    KÜLÖNBEN:  
     visszatér hamis értékkel
  + **+ eatenByHypa(b:Bug, h: Hypa): bool**: Meghatározza, hogy a Bug-ot meg tudja-e enni egy Hypa. Mivel csak bénított bogarat lehet megenni, ezért hamis értéket ad vissza.
  + **+ endOfTurn(b: Bug): void**: Mivel az osztály a spóra hatás hiányában valósítja meg a viselkedést, ezért nem csinál semmit.

### BiteBlocked

#### **Felelősség**

A BiteBlocked osztály valósítja meg azt a viselkedést, amikor a rovaron BiteBlockerSpore spóra hatása van, vagyis ha nem tud harapni.

#### **Ősosztályok**

Normal

#### **Metódusok**

#### **+ eat(b: Bug, s: Spore): bool**: Meghatározza, hogy a Bug tud-e enni az adott körben. Felüldefiniálja a Normal osztály eat metódusát. Mivel a Bug spóra hatása alatt áll, ezalatt nem ehet másik spórát, vagyis hamis értéket ad vissza.

* + **+ bite(b: Bug, h: Hypa): bool**: Meghatározza, hogy a Bug meg tudja-e harapni a megadott Hypa példányt. Felüldefiniálja a Normal osztály bite metódusát. Mivel a megvalósított hatás harapás gátolt, ezért hamis értéket ad vissza
  + **+ endOfRound(b: Bug): void**: A kör végén az adott hatás lejár-e, ha igen, vissza kell állítani a b Bug stratégiáját normálra. Felüldefiniálja a Normal osztály endOfRound metódusát. Növeli b underEffectSince tagváltozójának értékét, és amennyiben az eléri a 2-t, b strategy-jét Normal-ra állítja.

### Boosted

#### **Felelősség**

A Boosted osztály valósítja meg azt a viselkedést, amikor a rovaron BoosterSpore spóra hatása van, vagyis kétszer olyan távol tud eljutni egy kör alatt, mintha nem lenne rajta hatás: a kiinduló tektonjának második szomszédaira is át tud lépni fonal mentén.

#### **Ősosztályok**

Normal

#### **Metódusok**

#### **+ eat(b: Bug, s: Spore): bool**: Meghatározza, hogy a Bug tud-e enni az adott körben. Felüldefiniálja a Normal osztály eat metódusát. Mivel a Bug spóra hatása alatt áll, ezalatt nem ehet másik spórát, vagyis hamis értéket ad vissza.

* + **+ move(b: Bug, to: Tekton): bool**: Felüldefiniálja a Normal osztály move metódusát, az aktuális pozíció második szomszédaira is lehet lépni.  
    Pszeudokód:

Lekérdezi a bogártól a location Tekton értékét.  
Egy egyediséget és sorrendet biztosító gyűjteménybe beleteszi a location hypa menti szomszédjait.

ciklus: minden Tekton t objektum a location Tekton hypa menti szomszédjai között  
 t Tekton hypa menti szomszédait is hozzáadja az előbbi  
 gyűjteményhez  
HA ez a gyűjtemény tartalmazza a to paraméterként kapott Tektont, és sikeresen ráhelyezte b-t,akkor  
 b előző location-jének bug referenciáját null-ra állítja  
 b location-jét to-ra állítja  
 visszatér igaz értékkel  
KÜLÖNBEN  
 visszatér hamis értékkel

* + **+ endOfRound(b: Bug): void**: A kör végén az adott hatás lejár-e, ha igen, vissza kell állítani a b Bug stratégiáját normálra. Felüldefiniálja a Normal osztály endOfRound metódusát. Növeli b underEffectSince tagváltozójának értékét, és amennyiben az eléri a 2-t, b strategy-jét Normal-ra állítja.

### Slowed

#### **Felelősség**

A Slowed osztály valósítja meg azt a viselkedést, amikor a rovaron SlowerSpore spóra hatása van, vagyis lassabban tud mozogni, mint Normal stratégia esetén: a hatás két körében egyszer mozoghat.

#### **Attribútumok**

* + **- movesMade: int**: tárolja a hatás kezdete óta megtett mozgások számát. értéke kezdetben 0.

#### **Ősosztályok**

Normal

#### **Metódusok**

#### **+ eat(b: Bug, s: Spore): bool**: Meghatározza, hogy a Bug tud-e enni az adott körben. Felüldefiniálja a Normal osztály eat metódusát. Mivel a Bug spóra hatása alatt áll, ezalatt nem ehet másik spórát, vagyis hamis értéket ad vissza.

* + **+ move(b: Bug, to: Tekton): bool**: Meghatározza, hogy a Bug át tud-e lépni a kapott tektonra. Felüldefiniálja a Normal osztály move metódusát.   
    Pszeudokód:  
      
    HA movesMade>0  
     visszatér hamis értékkel

Lekérdezi a bogártól a location Tekton értékét, majd a locationtól a a hypák menti szomszédos Tektonoklistáját.  
Ha ebben a listában benne a to Tekton, és sikeresen ráhelyezte b-t,akkor  
 moveMade értékét növeli eggyel  
 b előző location-jének bug referenciáját null-ra állítja  
 b location-jét to-ra állítja  
 visszatér igaz értékkel  
különben:  
 visszatér hamis értékkel

* + **+ endOfRound(b: Bug): void**: A kör végén az adott hatás lejár-e, ha igen, vissza kell állítani a b Bug stratégiáját normálra. Felüldefiniálja a Normal osztály endOfRound metódusát. Növeli b underEffectSince tagváltozójának értékét, és amennyiben az eléri a 2-t, b strategy-jét Normal-ra állítja.

### Paralyzed

#### **Felelősség**

A Paralyzed osztály valósítja meg azt a viselkedést, amikor a rovaron ParalyzerSpore spóra hatása van, vagyis ha nem tud sem enni, sem mozogni, sem harapni.

#### **Ősosztályok**

Normal

#### **Metódusok**

#### **+ eat(b: Bug, s: Spore): bool**: Meghatározza, hogy a Bug tud-e enni az adott körben. Felüldefiniálja a Normal osztály eat metódusát. Mivel a Bug spóra hatása alatt áll, ezalatt nem ehet másik spórát, vagyis hamis értéket ad vissza.

* + **+ move(b: Bug, to: Tekton): bool**: Meghatározza, hogy a Bug át tud-e lépni a kapott tektonra. Felüldefiniálja a Normal osztály move metódusát. Mivel a hatás bénított, hamis értékkel tér vissza.
  + **+ bite(b: Bug, h: Hypa): bool**: Meghatározza, hogy a Bug meg tudja-e harapni a megadott Hypa példányt. Felüldefiniálja a Normal osztály eat metódusát, hamis értékkel tér vissza, mert a rovar bénított.
  + **+ eatenByHypa(b: Bug, h: Hypa): bool**: Mivel a rovar bénított hatás alatt áll, ezére mehető, vagyis b Bugger-jének listájából kivesszük b-t és b location-jének bug referenciáját null-ra állítjuk, ezzel eltávolítva a játékból b-t. igaz értékkel tér vissza.
  + **+ endOfRound(b: Bug): void**: A kör végén az adott hatás lejár-e, ha igen, vissza kell állítani a b Bug stratégiáját normálra. Felüldefiniálja a Normal osztály endOfRound metódusát. Növeli b underEffectSince tagváltozójának értékét, és amennyiben az eléri a 2-t, b strategy-jét Normal-ra állítja.

### Shroomer

#### **Felelősség**

A Shroomer osztály a gombász játékosok modellbeli megvalósítása. Referenciája van a gombatesteire és a gombafonalaira, tud spórát szórni egy kiválasztott gombatesttel vagy fonalat növeszteni két tekton közé, ezek kiválthatnak gombatest növést; valamint meg tud etetni egy bogarat egy fonalával. Egy gombász egy fajta gombatesteket tud növeszteni, amivel egyedül a gombafajtának megfelelő spórákat tud szórni, és fonalai a gombafajnak megfelelő számú körig élnek elharapást követően.

#### **Ősosztályok**

Player

#### **Attribútumok**

* + **- mushrooms: List<Mushroom>:** referenciák az élő gombatestekre
  + **- hypaList: List<Hypa>:** referenciák az élő fonalakra
  + **- hypaDieAfterBite: int**: a fonalak ennyi körig maradnak életben elharapást követően
  + **- mushroomctor: BiFunction<Shroomer, Tekton, Mushroom>**: a gombatestek konstruktora BiFunction objektumként, Mushroom newMus = mushroomctor.apply(this, target) hívással tud gombatestet példányosítani.

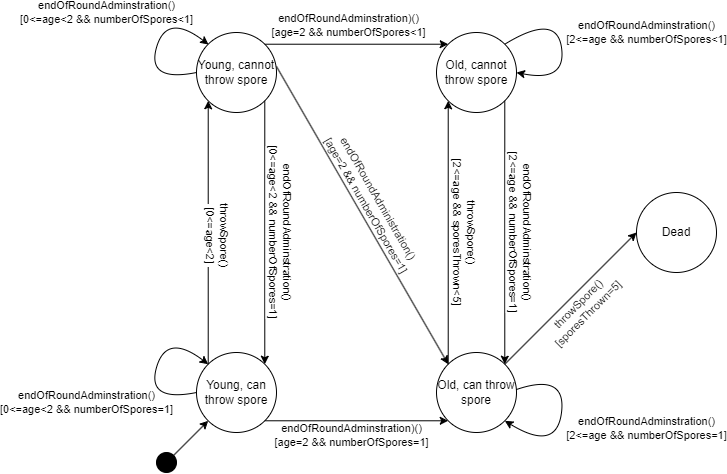
#### **Metódusok**

* + **+ Shroomer(mushroomctor: BiFunction<Shroomer, Tekton, Mushroom>, hypaDieAfterBite: int)**:   
    A Shroomer osztály konstruktora. paraméterként kapja a gombatesteinek konstruktorát és a fonalainak életben maradását jelző egész számot. Törzsében inicializálja a listáit és a kapott paramétereket értékül adja a tagváltozóinak.  
    (hívási példa: new Shroomer((x, y)->new BoosterMushroom(x, y), 2)
  + **+ getMushrooms(): List<Mushroom>**: getter a gombatestek listájához
  + **+ getHypaList(): List<Hypa>**: getter a fonalak listájához
  + **+ getHypaDieAfterBite(): int**: getter a hypaDieAfterBite tagváltozóhoz
  + **+ growHypa(start: Tekton, target: Tekton): bool:** fonál növesztést valósít meg a start és a target tekton között, amennyiben ez lehetséges (a start tektonon van saját fonál vagy gombatest, a targeten a gombásznak nőhet fonala és az új fonállal nem keletkezik többszörös él). Visszatérési értéke a növesztés sikeressége.  
    Pszeudokód:  
      
    alapvetően beállítja egy validStart értékét hamisra  
    CIKLUS a mushrooms lista minden m Muhsroom elemére  
     HA m a start tektonon van, akkor  
     validStart értékét beállítja igazra  
    CIKLUS a HypaList minden h elemére:  
     HA a h Hypa csatlakozik a start tektonhoz és nem  
     keletkezik többszörös él, akkor  
     validStart értékét beállítja igazra  
     HA a h Hypa a start és a target tektonok között van,  
     akkor  
     validStart értékét beállítja hamisra  
     CIKLUS STOP  
    HA start és target nem szomszédosak VAGY azonos Tektonra mutatnak, akkor  
     validStart értékét beállítja hamisra  
    HA validStart értéke igaz, akkor  
     HA target tekton acceptHypa() metódusa igaz értékkel tér vissza, akkor   
     Létrehoz egy új Hypa objektumot, ezt beállítja a  
     start és target Tektonon is, hozzáadja a hypasList  
     tömbjéhez  
     megpróbál a tryGrowMushroom(target) metódussal  
     automatikusan gombatesztet növeszteni  
     elindítja a traverseHypaNetwork() metódust  
     visszatér igaz értékkel  
    visszatér hamis értékkel
  + **+ growHypaFar(start: Tekton, middle: Tekton, target: Tekton): bool**: Fonál növesztést valósít meg távolra akkor, ha a kiinduló tektonon van (nem feltétlenül saját) spóra. Két fonál nőhet ezáltal az akció által, de a metódus hívható úgy is, hogy a target paraméter null, ekkor csak a start és middle közé nő fonál. A növesztések visszavezethetőek a growHypa függvényre: először növeszt a start és a middle közé (ha ez sikeres, a visszatérési érték már igaz lesz, különben hamis), majd ha a target nem null, megpróbál a middle és a target közé is növeszteni.  
    Pszeudokód:HA a start tektonon van spóra, akkor  
     HA a meghívott growHypa(start, middle) metódus igaz értékkel tér vissza, akkor  
     HA target paraméter valóban egy Tektonra mutat,  
     akkor  
     meghívja a growHypa(middle, target) metódust  
     visszatér igaz értékkel  
    visszatér hamis értékkel
  + **+ throwSpore(m: Mushroom, target: Tekton): bool**: spóra szórást valósít meg úgy, hogy ha az aktuális állapot szerint lehetséges (van-e dobható spóra, milyen messzire tud dobni a gombatest), példányosít egy spórát a kiválasztott m gombatesttel és elhelyezi a target tektonon, majd megpróbál oda gombatestet növeszteni a tryGrowMushroom metódus hívásával.   
    Pszeudokód:  
      
    HA m nincs a saját gombatestek listájában, akkor  
     visszatér hamis értékkel  
    HA m numberOfSpores értéke 1-nél kevesebb, akkor  
     visszatér hamis értékkel  
    Lekérdezi a location Tekton-t az m Mushroom objektumtól  
    Lekérdezi az age int-et az m Mushroom objektumtól  
    HA age<=4, akkor  
     HA target nem eleme a location szomszédainak, akkor  
     visszatér hamis értékkel  
    KÜLÖNBEN  
     HA target nem eleme a location szomszédainak  
     és az azok szomszédainak halmazának uniójának   
     visszatér hamis értékkel  
    meghívja az m Mushroom sporeThrown(target) metódusát  
    megpróbál a tryGrowMushroom(target) metódussal  
    automatikusan gombatesztet növeszteni  
    visszatér igaz értékkel
  + **+ addMushroom(m: Mushroom): void**: hozzáadja a gombatestek listájához a kapott m gombatestet
  + **+ mushroomDied(m: Mushroom): void**: Gombatest halálakor kell hívni, az elhaló gombatestet paraméterként adva neki. A gombatest tektonjának gomba referenciáját null-ra állítja, a saját gombatest listájából kiveszi m-et, és meghívja a traverseHypaNetwork() metódust, mivel a gombatest halálával keletkezhettek olyan fonalak, amelyek nem csatlakoznak élő gombatesthez.
  + **+ removeHypa(h: Hypa): void**: kiveszi a fonal listából h-t
  + **+ addHypa(h: Hypa): void**: Hozzáadja a fonalak listájához h-t
  + **+ endOfRoundAdministration(): void**: A játék kör végén ebben a metódusban öregít minden gombatestet és fonalat a gombász, valamint itt kerülnek eltüntetésre a lecsatlakozás miatt elhaló fonalak és a harapás után elhaló fonalak.  
    Pszeudokód:  
      
    CIKLUS a mushrooms lista minden m elemére:  
     meghívja az adott m Mushroomra az age() metódust  
     HA az adott m Mushroom getNumberOfSpores()!=1  
     meghívja az adott m Mushroomra az  
     increaseNumberOfSpores() metódust  
    CIKLUS a hypas lista minden h elemére:  
     HA h isDyingSinceDisconnected tagváltozója!=1 ÉS h  
     isDyingSinceBitten tagváltozója != hypaDieAfterBite  
     meghívja az adott h Hypara az age() metódust  
     HA h isDyingSinceDisconnected tagváltozója == 1  
     lekérdezi az adott h Hypa kettő végét, és mindkettő  
     tektonnak a listájából eltávolítja a h Hypát  
     eltávolítja a saját hypaList-jéből is a h Hypát  
     HA h isDyingSinceBitten tagváltozója == hypaDieAfterBite  
     meghívja az adott h Hypara az die() metódust
  + **+ traverseHypaNetwork(): void**: A fonalak gráfjának bejárása, ennek a metódusnak az a célja, hogy detektálja az élő gombatesthez nem csatlakozó fonalakat. és 0-ra állítsa az isDyingSinceDisconnected tagváltozójukat -1-ről, ezzel jelezve, hogy elkezdtek haldokolni. Mindig meg kell hívni, ha új gombatest nő, új fonál nő, vagy ha egy fonál vagy gombatest meghal.  
    Pszeudokód:  
      
    Létrehoz egy inNetworkTektons kollekciót (egyediséget biztosítva)  
    CIKLUS a mushrooms lista minden m elemére  
     inNetworkTektons-hoz hozzáadja az m location Tektonját  
    CIKLUS az inNetworkTektons minden t elemére  
     a t-hez csatlakozó saját Hypa-k mindkét végét az  
     inNetworkTektons halmazba tesszük (ha még nincs benne)  
    CIKLUS HypaList minden h elemére  
     HA az inNetworkTektons tartalmazza h valamelyik végét  
     adott h hypának beállítja isDyingSinceDisconnected  
     értékét -1-re  
     KÜLÖNBEN  
     HA h hypa isDyingSinceSiconncected értéke -1, akkor  
     adott h hypának beállítja  
     isDyingSinceDisconnected értékét 0-ra
  + **+ eatBug(b: Bug): bool**: Ezzel a metódussal a gombász egy fonalával megeheti a paraméterként kapott b Bugot. Először megnézi, hogy csatlakozik-e saját Hypa b pozíciójához. Ha létezik ilyen h Hypa, akkor meghívja b beEaten(h) metódusát, ami visszatér azzal, hogy b-t meg tudta-e enni h. Ha ez sikerül, a Shroomer megpróbál növeszteni egy gombatestet a rovar helyére. Ha sikerült megenni a rovart (de gombatestet nem feltétlenül növesztett), igaz értékkel tér vissza, egyébként hamissal.   
    Pszeudokód:  
      
    Lekérdezi a b Bug Tekton location-jét.  
    CIKLUS a location Tekton connectedHypas minden h elemére  
     HA h Hypa Shroomer-e azonos a függvényt indító  
     Shroomerrel, akkor  
     HA b Bugra meghívott beEaten(h) metódus igaz  
     értékkel tér vissza, akkor  
     HA location Tektononnincs Mushroom, akkor  
     Létrehoz egy új Mushroom objektumot  
     Hozzáadja a saját mushrooms listhez  
     Locationnak beállítja ezt az új obj.-ot  
     meghívja a traversHypaNetwork() met.-et  
     visszatér igaz értékkel  
    visszatér hamis értékkel
  + **+ tryGrowMushroom(target: Tekton): void**: Ha a target tektonon nőhet gombateste a gombásznak, létrehoz egy új gombatest példányt és ráteszi.  
    Pszeudokód:  
      
    HA target Tekton canMushroomGrow(this) metódusa igaz értékkel tér vissza, akkor  
     Létrehoz egy új Mushroom objektumot  
     Hozzáadja a saját mushrooms listhez  
     Meghívja a target Tekton setMushroomRemoveSpore metódusát  
     az új Mushroom objekttel (ami leveszi a megfelelő  
     spórákat, és elhelyezi az új Mushroom obj.-ot)  
     meghívja a traversHypaNetwork() met.-et

### Mushroom

#### **Felelősség**

A Mushroom absztrakt osztály a gombatesteket reprezentálja a játékban. Minden Mushroom egy adott Tektonon található és egy Shroomer-hez tartozik. A gombatestek megöregedhetnek, elhalhatnak és spórákat szórhatnak, valamint fonalakat (Hypa) növeszthetnek.   
Állapotgép:



#### **Attribútumok**

* + **# location: Tekton**: referencia a gombatest pozíciójára
  + **# shroomer: Shroomer**: referencia a gombatest gazdájára
  + **# age: int**: a gombatest kora körökben
  + **# numberOfSpores: int**: a gombatest eldobható spóráinak száma, default értéke 1
  + **# sporesThrown: int**: az eddig eldobott spórák száma, default értéke 0

#### **Metódusok**

* + **+ Mushroom(s: Shroomer, pos: Tekton)**: Konstruktor baállítja a location referenciát a paraméterként kapott pos-ra, és a shroomer referenciát a paraméterként kapott shroomerre.
  + **+ getLocation(): Tekton**: getter a location attribútumhoz
  + **+ getShroomer(): Shroomer**: getter a shroomer attribútumhoz
  + **+ getNumberOfSpores(): int**: getter a numberOfSpores tagváltozóhoz
  + **+ getSporesThrown(): int**: getter a sporesThrown tagváltozóhoz
  + **+ increaseNumberOfSpores(): void**: növeli a numberOfSpores értékét 1-gyel
  + **+ *sporeThrown(to: Tekton): void***: ezt a metódust hívva tudunk spórát dobatni a gombatesttel a paraméterként átadott tektonra, ha az szomszédja a location tektonnak. A Mushroom osztályban absztrakt a metódus, a leszármazottakban felül kell definiálni, minden leszármazott osztály sporeThrown metódusa a neki megfelelő spórát példányosítja. Növeli a sporesThrown tagváltozó értékét, a numberOfSpores értékét -1-re állítja (mert a kör végén a Shroomer::endOfRoundAdministration()-ben nőni fog, és ha 0-ra állítanánk, a következő körben már újra tudna szórni), majd ellenőrzi, hogy a sporesThrown tagváltozó elérte-e az 5-öt (ennyi kör után hal meg minden gombatest), ha igen, meghívja magára a die() metódust.
  + **+ die(): void**: A gombatest halálakor hívódik, a shroomer mushroomDied metódusát hívja saját magát átadva paraméterként
  + **+ age(): void**: a gombatest korát lehet növelni, ha meghívjuk: a metódus törzsében növeli az age tagváltozó értékét.

### BiteBlockerMushroom

#### **Felelősség**

#### A BiteBlockerMushroom egy speciális gombafajta, amely BiteBlockerSpore spórákat tud szórni

#### **Ősosztályok**

Mushroom

#### **Metódusok**

* + **+ BiteBlockerMushroom(shroomer: Shroomer, location: Tekton)**: A konstruktor az ős konstruktorát hívja átadva neki a paraméterként kapott shroomer és location objektumokat
  + **+ sporeThrown(to: Tekton): void**: Felüldefiniálja a Mushroom osztály sporeThrown metódusát, BiteBlockerSpore osztályt példányosít és helyez el a to tektonon.

### BoosterMushroom

#### **Felelősség**

#### A BoosterMushroom egy speciális gombafajta, amely BoosterSpore spórákat tud szórni

#### **Ősosztályok**

Mushroom

#### **Metódusok**

* + **+ BoosterMushroom(shroomer: Shroomer, location: Tekton)**: A konstruktor az ős konstruktorát hívja átadva neki a paraméterként kapott shroomer és location objektumokat
  + **+ sporeThrown(to: Tekton): void**: Felüldefiniálja a Mushroom osztály sporeThrown metódusát, BoosterSpore osztályt példányosít és helyez el a to tektonon.

### ParalyzerMushroom

#### **Felelősség**

#### A ParalyzerMushroom egy speciális gombafajta, amely ParalyzerSpore spórákat tud szórni

#### **Ősosztályok**

Mushroom

#### **Metódusok**

* + **+ ParalyzerMushroom(shroomer: Shroomer, location: Tekton)**: A konstruktor az ős konstruktorát hívja átadva neki a paraméterként kapott shroomer és location objektumokat
  + **+ sporeThrown(to: Tekton): void**: Felüldefiniálja a Mushroom osztály sporeThrown metódusát, ParalyzerSpore osztályt példányosít és helyez el a to tektonon.

### ProliferatingMushroom

#### **Felelősség**

#### A ProliferatingMushroom egy speciális gombafajta, amely ProliferatingSpore spórákat tud szórni.

#### **Ősosztályok**

Mushroom

#### **Metódusok**

* + **+ ProliferatingMushroom(shroomer: Shroomer, location: Tekton)**: A konstruktor az ős konstruktorát hívja átadva neki a paraméterként kapott shroomer és location objektumokat
  + **+ sporeThrown(to: Tekton): void**: Felüldefiniálja a Mushroom osztály sporeThrown metódusát, ProliferatingSpore osztályt példányosít és helyez el a to tektonon.

### SlowerMushroom

#### **Felelősség**

#### A SlowerMushroom egy speciális gombafajta, amely SlowerSpore spórákat tud szórni.

#### **Ősosztályok**

Mushroom

#### **Metódusok**

* + **+ SlowerMushroom(shroomer: Shroomer, location: Tekton)**: A konstruktor az ős konstruktorát hívja átadva neki a paraméterként kapott shroomer és location objektumokat
  + **+ sporeThrown(to: Tekton): void**: Felüldefiniálja a Mushroom osztály sporeThrown metódusát, SlowerSpore osztályt példányosít és helyez el a to tektonon.

### *Spore*

#### **Felelősség**

#### A Spore egy absztrakt osztály, amely a különböző spórák alapvető tulajdonságait és működését definiálja.Minden spóra egy adott Shroomerhez tartozik (A Mushroom-jai hozzák létre), és hatást gyakorolhat egy Bug-ra.

#### **Attribútumok**

* + **- shroomer: Shroomer:** referencia a gazdájára

#### **Metódusok**

* + **+ Spore(s: Shroomer)**: A konstruktor beállítja a gazda referenciáját a paraméterként kapott Shroomerre.
  + **+ getShroomer(): Shroomer**: getter a shroomer attribútumhoz
  + **+ *haveEffect(b: Bug): int****:* Ez az absztrakt metódus fejti ki a spóra hatását egy Bug-ra. Az implementáló osztályok határozzák meg a konkrét hatásokat. Ha egy Bug megeszik egy spórát, ezt a metódust kell hívnia, a visszatérési értékével jelzi, hogy mennyi az adott spóra tápanyagtartalma.

### BiteBlockerSpore

#### **Felelősség**

#### A BiteBlockerSpore egy speciális spóra, amely megakadályozza a Bug harapását, ha megeszi. A hatását a haveEffect metódus hívásával fejti ki, tápanyagtartalma 2 egység.

#### **Ősosztályok**

Spore

#### **Metódusok**

* + **+ BiteBlockerSpore(shroomer: Shroomer)**: A konstruktor hívja az ős konstruktorát és átadja neki a kapott shroomer paramétert.
  + **+ haveEffect(b: Bug): int**: példányosít egy BiteBlocked objektumot, és a b Bug setStrategy metódusával beállítja neki ezt, majd visszatér a tápanyag tartalmával (2)

### BoosterSpore

#### **Felelősség**

#### A BoosterSpore egy speciális spóra, amely gyorsító hatást gyakorol a Bug mozgására, ha megeszi. A hatását a haveEffect metódus hívásával fejti ki, tápanyagtartalma 1 egység.

#### **Ősosztályok**

Spore

#### **Metódusok**

* + **+ BoosterSpore(shroomer: Shroomer)**: A konstruktor hívja az ős konstruktorát és átadja neki a kapott shroomer paramétert.
  + **+ haveEffect(b: Bug): int**: példányosít egy Boosted objektumot, és a b Bug setStrategy metódusával beállítja neki ezt, majd visszatér a tápanyag tartalmával (1)

### ParalyzerSpore

#### **Felelősség**

#### A ParalyzerSpore egy speciális spóra, amely megbénítja a Bug-ot, ha megeszi. A hatását a haveEffect metódus hívásával fejti ki, tápanyagtartalma 4 egység.

#### **Ősosztályok**

Spore

#### **Metódusok**

* + **+ ParalyzerSpore(shroomer: Shroomer)**: A konstruktor hívja az ős konstruktorát és átadja neki a kapott shroomer paramétert.
  + **+ haveEffect(b: Bug): int**: példányosít egy Paralyzed objektumot, és a b Bug setStrategy metódusával beállítja neki ezt, majd visszatér a tápanyag tartalmával (4)

### ProliferatingSpore

#### **Felelősség**

A ProliferatingSpore egy speciális spóra, amely osztódásra készteti a rovart, amelyik megeszi. A hatását a haveEffect metódus hívásával fejti ki, tápanyagtartalma 0 egység.

#### **Ősosztályok**

Spore

#### **Metódusok**

* + **+ ProliferatingSpore(shroomer: Shroomer)**: A konstruktor hívja az ős konstruktorát és átadja neki a kapott shroomer paramétert.
  + **+ haveEffect(b: Bug): int**: Létre hoz egy új Bug példányt, aminek ugyanaz a Bugger a gazdája, mint a spórát megevő b rovaré. Az új rovar pozíciója a b pozíciójából indított BFS-szerű bejárás során először megtalált szabad tekton lesz, amennyiben ilyen létezik. Ha nem, az új Bug objektum nem kerül játékba. A metódus visszatér a spóra tápanyag tartalmával (0).  
    Pszeudokód:  
      
    Lekérdezi a b Bug Bugger-jét  
    Létrehoz egy új Bug objektumot ehhez a Bugger-hez  
    b location-jének szomszédait egy closestTektons Queue-ba tesszük  
    CIKLUS amíg a closestTektons nem üres:  
     Tekton tekton <- a closestTektons első eleme  
     a closestTektons első elemét kivesszük  
     HA tekton-ra lehet bogarat tenni  
     bugger Bug listájába tesszük newBug-ot  
     tekton-ra tesszük newBug-ot  
     newBug location-jét tekton-ra állítjuk  
     CIKLUS BREAK  
     KÜLÖNBEN  
     tekton szomszédait egy currentNeighbours listába  
     tesszük  
     CIKLUS currentNeighbours minden n elemére:  
     HA closestTektons-nak nem eleme n  
     n-t a closestTektons queue végére  
     tesszük  
    visszatér 0 értékkel, ami ennek a spórának a tápanyagtartalma

### SlowerSpore

#### **Felelősség**

A SlowerSpore egy speciális spóra, amely lassítja a Bug mozgását, ah megeszi. Tápanyag tartalma 3 egység.

#### **Ősosztályok**

Spore

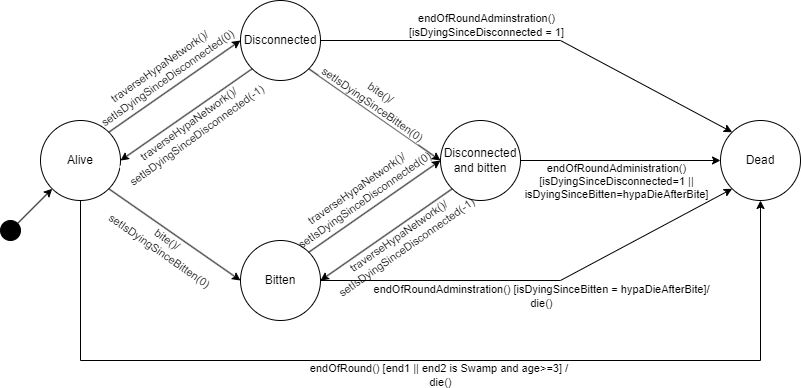
#### **Metódusok**

* + **+ SlowerSpore(shroomer: Shroomer)**: A konstruktor hívja az ős konstruktorát és átadja neki a kapott shroomer paramétert.
  + **+ haveEffect(b: Bug): int**: példányosít egy Slowed objektumot, és a b Bug setStrategy metódusával beállítja neki ezt.

### Hypa

#### **Felelősség**

A Hypa osztály objektumai gombafonalak kettő tekton között. Egy gombafonál mindkettő végéhez kapcsolódó tektonon rajta van. A Hypa az idő múlásával elöregedhet, és ha nem része egy gombatest-hálózatnak, elhalhat, valamint elharapást követően is meghal bizonyos számú kör után. Ha valamelyik végén Swamp van, 3 kör után meghal, ha viszont valamelyik végén Soil van, hálózatról való leválást követően sem hal meg.  
Állapotgép:



#### **Attribútumok**

* + **- shroomer: Shroomer:** referencia a birtokló gombászra
  + **- end1: Tekton:** referencia az egyik végét birtokló tektonra
  + **- end1: Tekton:** referencia a másik végét birtokló tektonra
  + **- age: int:** a hypának a kora körökben, default értéke 0
  + **- isDyingSinceDisconnected: int:** jelzi, hogy a hypa gombahálózatának része, vagy hogy hány köre nem része, default értéke -1
  + **- isDyingSinceBitten: int:** jelzi, hogy a hypát még nem harapták el, vagy hogy hány köre harapták el, default értéke -1

#### **Metódusok**

* + **+ Hypa(end1: Tekton, end2: Tekton, s: Shroomer)**: Konstruktor beállítja a end1 és end2 referenciát a paraméterként kapott end1 és end2-re, és a shroomer referenciát a paraméterként kapott shroomerre.
  + **+ getShroomer(): Shroomer**: getter a shroomer attribútumhoz
  + **+ getEnd1(): Tekton**: getter az end1 attribútumhoz
  + **+ getEnd2(): Tekton**: getter az end2 attribútumhoz
  + **+ getAge(): int**: getter az age tagváltozóhoz
  + **+ getIsDyingSinceDisconnceted: int**: getter az isDyingSinceDisconnected tagváltozóhoz
  + **+ setIsDyingSinceDisconnceted: void**: setter az isDyingSinceDisconnected tagváltozóhoz
  + **+ getIsDyingSinceBitten: int**: getter az isDyingSinceBitten tagváltozóhoz
  + **+ setIsDyingSinceBitten: void**: setter az isDyingSinceBitten tagváltozóhoz
  + **+ age(): void**: növeli az age értékét 1-gyel, és az isDyingSinceDisconnected és isDyingSinceBitten értékét is, ha nem -1 értékűek.
  + **+die(): void:** a kettő birtokló tekton, és a birtokló shroomer listájából kiveszi önmagát és meghívja a shroomer traverseHypaNetwork() metódusát.

### Tekton

#### **Felelősség:**

A Tekton osztály egy egységet reprezentál a pályán. Lehetnek szomszédai, tárolhat spórákat, illetve kapcsolódhat hozzá gomba vagy bogár. A Tektonok egymáshoz fonalakkal kapcsolódhatnak és el is törhetnek a kör végén.

#### **Attribútumok:**

* + **# bug: Bug**: az adott tektonon lévő bogár.
  + **#mushroom: Mushroom:** A tektonon lévő gomba.
  + **#storedSpores: List<Spore>:** A tektonon lévő spórák lisája.
  + **#neighbours: List<Tekton>:** A tektonnal szomszédos tektonok, akik közvetlen egymás mellett vannak, de nincsenek feltétlen összekötve.
  + **#connectedHypas: List<Hypa>**: A tektonhoz kapcsolódó fonalak.

#### **Metódusok**

* + **+hasSpore(): boolean:** Ellenőrzi, hogy van-e legalább egy spóra a tektonon. True-val tér vissza, ha van, false-al, ha nincs egy sem.
  + **+breakTekton(): void:** A tekton törését végző függvény. Létrehoz egy új Tekton példányt. A régi tekton szomszédjait véletlenszerűen szétosztja két részre a két tekton között. A régi tekton fonaljai elhalnak.
  + **+removeNeighbour(neighbour: Tekton):void:**: Eltávolítja a megadott szomszédot a tekton szomszédlistájából.
  + **+hasMushroom():boolean:** Ellenőrzi, hogy van-e gomba az adott tektonon, amelyik hívja a függvényt.True-val tér vissza, ha van, false-al, ha nincs.
  + **+storeSpore(s: Spore): void:** Az adott spórát az elszórás után hozzáadja a tekton spóralistájához.
  + **+acceptHypa(shroomer: Shroomer): boolean:** Ellenőrzi, hogy az adott gomba kapcsolhat-e ehhez a tektonhoz fonalat. Ha még nincs fonál kapcsolva a tektonhoz, akkor true-val tér vissza. Ha van már fonálja valamelyik gombásznak, akkor ellenőrizzük, hogy a beadott gombásznak van-e már fonala rajta, ha igen, akkor true-val tér vissza. Ha egyik feltétel sem, akkor false-al tér vissza.
  + **+removeSpore(s: Spore): void:** Eltávolítja az adott spórát a tektonról.
  + **+tryBug(b: Bug): void:** Ha a tektonon még nincs más bogár, akkor hozzáadja az adott bogarat a tektonhoz.
  + **+getNeigboursByHypa(): List<Tekton>:** Megvizsgálja a tekton fonállistáját. A fonálvégek alapján hozzáadja a szomszédos tektonokat egy listához. Azaz, a tekton szomszédjainak listáját adja vissza, amikkel össze van kötve fonállal:

Pszeudokód:   
Létrehoz egy új Tekton listát

CIKLUS a Tekton connectedHypas minden h elemére  
 lekérdezi a h Hypának mindkettő végét  
 HA az egyik vagy a másik vége, nem maga a Tekton, és még  
 nincs benne a listában, akkor hozzáadjuk azt a Tektont  
VISSZATÉR a most létrehozott és feltöltött Tekton listával

* + **+removeHypa(h: Hypa): void:** Eltávolítja az adott fonalat a tekton fonállistájából.
  + **+canMushroomGrow(s: Shroomer): boolean:** Megvizsgálja, hogy az adott gombász növeszthet-e gombát a tektonon. Ellenőrzi, hogy van-e csatlakozó fonala a tektonhoz, és 3-nál kevesebb spórája van a tektonon, ez esetben true-val tér vissza. Másik esetekben false-al.
  + **+connectHypa(h: Hypa): void:** Az adott fonalat hozzáadja a tekton fonállistájához.
  + **+setMushroomRemoveSpores(msh: Mushroom): void:** A tektonon a spórákból gombatest nő. Az adott gombát beállítja a tekton gombájának. Megkeresi, melyik gombászhoz tartozik a gomba, és annak a spóráit távolítja el a tekton spóralistájából.

Pszeudokód:

HA mushroom tagváltozó nem mutat gombára ÉS msh!= null, akkor

mushroom tagváltozót beállítja a msh Mushroom obj.-ra  
 Lekérdezi a msh Mushroom shroomer változoját

Létrehoz egy int removedCount változót 0 értékkel  
 CIKLUS a storedSpores minden s elemén  
 HA s shroomer tagváltózója megegyezik az msh  
 shroomerével ÉS removedCount még érte el a 3-at,   
 akkor  
 Eltávolítjuk az s Spore-t a storedSpores  
 listából és növeljük eggyel a removedCount-ot

* + **+getHypas(): List<Hypa>:** A tekton fonállistáját adja vissza.
  + **+getNeigbours(): List<Tekton>:** A tekton szomszédjait adja vissza.
  + **+setNeigbours(neighbours: List<Tekton>): void:** A tekton szomszédjait állítja be.
  + **+addNeighbour(t: Tekton): void:** A jelenlegi tektonghoz hozzáadja az adott tektont, mint szomszédot.
  + **+getStoredSpores(): List<Spore>**: A tektonon lévő spórákat adja vissza.
  + **+setStoredSpores(storedSpores: List<Spore>): void:**: A tektonon lévő spórákat adja vissza.
  + **+getMushroom(): Mushroom:** A tektonon lévő gombát adja vissza.
  + **+setMushroom(mushroom: Mushroom): void:** A tektonon lévő gombát állítja be.
  + **+getBug(): Bug:** A tektonon lévő bogarat adja vissza.
  + **+setBug(bug: Bug): void:** A tektonon lévő bogarat állítja be.
  + **+endOfRound(): void:** alapvetően nem csinál semmit, bizonyos Tekton típus leszármazott osztályokban felül van írva

### Peat

#### **Felelősség:**

Egy speciális tekton típus, amelyen több fonal is keresztezheti egymást. Ez azt jelenti, hogy az acceptHypa függvény mindig true-val tér vissza.

#### **Ősosztályok:**

* + Tekton

#### **Metódusok:**

* + **+acceptHypa(shroomer: Shroomer): boolean:** Mindig igaz értéket ad vissza, mert a Peat nem korlátozza a fonalak számát.

### Soil

#### **Felelősség:**

Egy speciális tekton típus, amelyik életben tartja azokat a fonalakat, amelyek nincsenek közvetve vagy közvetlenül gombatesthez kötve.

#### **Ősosztályok:**

* + Tekton

#### **Metódusok:**

* + **+endOfRound(): void:** A hozzá csatlakozó fonalak élettartamát a körök végén visszaállítja, hogy tovább éljenek.

### Stone

#### **Felelősség:**

Egy speciális tekton típus, amelyekre nem nőhetnek gombatestek. Azt jelenti, hogy a canMushroomGrow mindig hamis értéket ad vissza.

#### **Ősosztályok:**

* + Tekton

#### **Metódusok:**

* + **+canMushroomGrow(s: Shroomer): boolean:** Mindig false-al tér vissza, mivel nem nőhet rajta gombatest.

### Swamp

#### **Felelősség:**

Egy speciális tekton típus, amelyen a fonalak egy idő után elhalnak.

#### **Ősosztályok:**

* + Tekton

#### **Metódusok:**

* + **+endOfRound(): void:** A kör végén meghívja a checkForDeleteHypa függvényt.
  + **+checkForDeleteHypa(): void:** Ellenőrzi, hogy vannak-e fonalak, amelyeknek a kora elérte a 3 kört. Ezeket eltávolítja a fonállistájából egy új lista létrehozásával.

## **8.2** **A tesztek részletes tervei, leírásuk a teszt nyelvén**

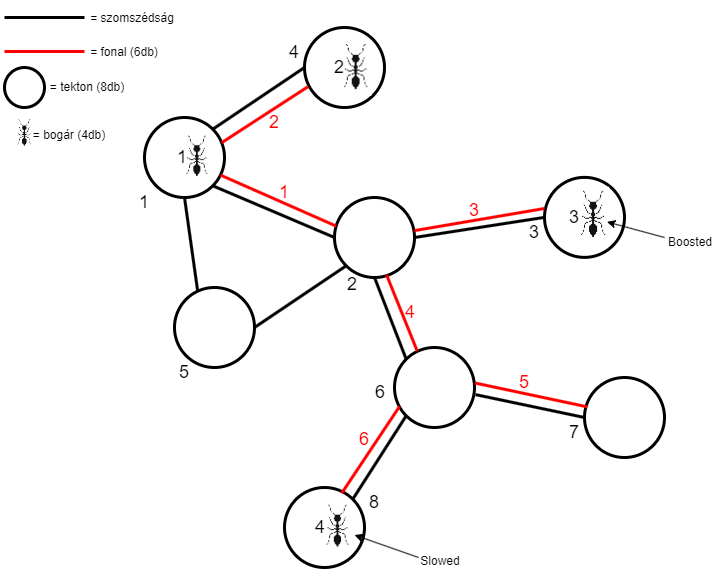
* Új teszt eset (TEK\_1) amiben egy tekton törését teszteljük.
* Új ACT parancs, hogy konkrét Tekton törését tudjuk tesztelni:  
  [Tekton object name] break
* ACT parancsoknál az elválasztás “ “ helyett “;”, hogy egységes legyen.
* ACT parancsoknál nem szükséges a akciót végrehajtandó Player objektum nevét megadni. Csak az aktuális soron következő játékos hajthat végre akciót. (Ez alól kivétel a konkrét tekton törését előidéző break parancs)
* Spores objektumok megadásánál Shroomer objektum megadása  
  [Spore class name];[Shroomer object name];[Tekton object name]
* Új ACT parancs, hogy egy játékos akció végrehajtás nélkül tovább engedhesse a játékot

**skip**

**Leírás:** Akció nélkül lépteti az aktuális játékost.

### Közös arrange bemenetek, amelyeket több tesztesetnél használandók A rajzok nem részei a bemenetnek, csak illusztrációk az érthetőség megkönnyítése érdekében.

**arrange\_1** (tesztesetei: BUG\_01, …, BUG\_10)

ARRANGE

Tektons

Tekton

Tekton

Tekton

Tekton

Tekton

Tekton

Tekton

Tekton

Neighbours

tekton1;tekton2;tekton4;tekton5

tekton2;tekton3;tekton5;tekton6

tekton6;tekton7;tekton8

Shroomers

booster

Mushrooms

Buggers

;

Strategies

Normal

Normal

Boosted

Slowed

Bugs

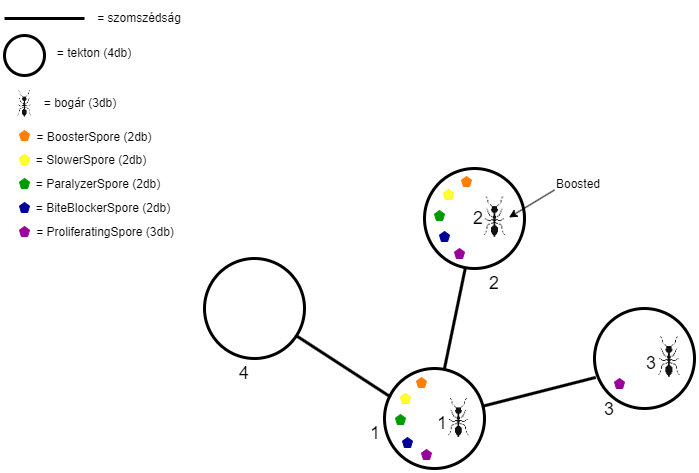
normal1;bugger1;tekton1

normal2;bugger1;tekton4

boosted1;bugger1;tekton3

slowed1;bugger1;tekton8

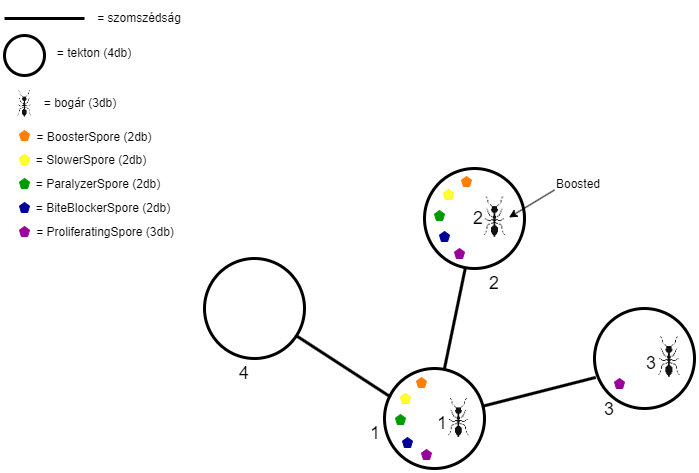
Hypas

tekton1;tekton2;shroomer1

tekton1;tekton4;shroomer1

tekton2;tekton3;shroomer1

tekton2;tekton6;shroomer1

tekton6;tekton7;shroomer1

tekton6;tekton8;shroomer1

Spores

**arrange\_2** (tesztesetei: BUG\_11,..., BUG\_26)

ARRANGE

Tektons

Tekton

Tekton

Tekton

Tekton

Neighbours

tekton1;tekton2;tekton3;tekton4

Shroomers

booster

Mushrooms

Buggers

;

Strategies

Normal

Normal

Boosted

Bugs

normal1;bugger1;tekton1

boosted1;bugger1;tekton2

normal2;bugger1;tekton3

Hypas

Spores

BoosterSpore;shroomer1;tekton1

BoosterSpore;shroomer1;tekton2

SlowerSpore;shroomer1;tekton1

SlowerSpore;shroomer1;tekton2

ParalyzerSpore;shroomer1;tekton1

ParalyzerSpore;shroomer1;tekton2

BiteBlockerSpore;shroomer1;tekton1

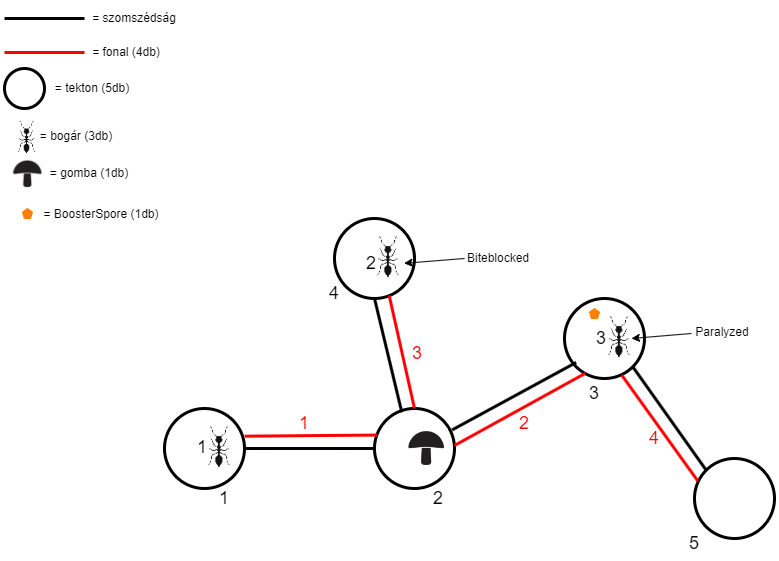
BiteBlockerSpore;shroomer1;tekton2

ProliferatingSpore;shroomer1;tekton1

ProliferatingSpore;shroomer1;tekton2

ProliferatingSpore;shroomer1;tekton3

**arrange\_3** (tesztesetei: BUG\_27,..., BUG\_32, SHO\_20)

ARRANGE

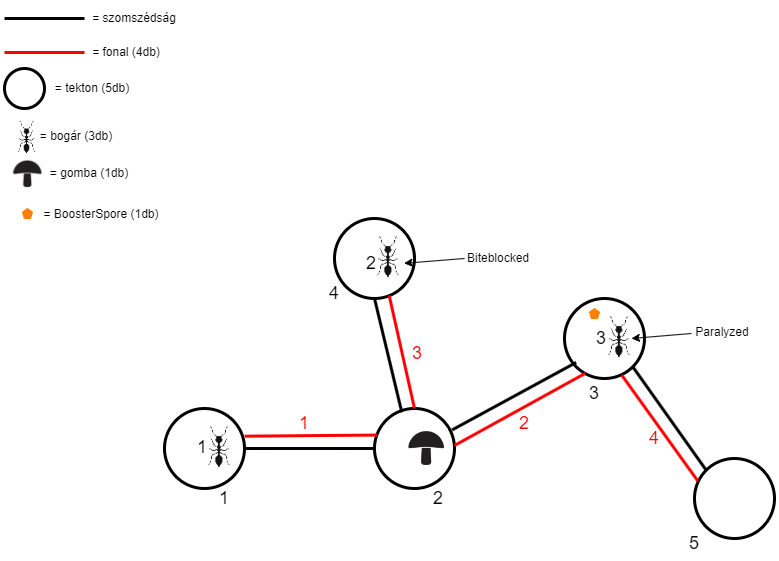
Tektons

Tekton

Tekton

Tekton

Tekton

Tekton

Neighbours

tekton2;tekton1;tekton3;tekton4

tekton3;tekton5

Shroomers

booster

Mushrooms

BoosterMushroom;shroomer1;tekton2

Buggers

;

Strategies

Normal

BiteBlocked

Paralyzed

Bugs

normal1;bugger1;tekton1

biteblocked1;bugger1;tekton4

paralyzed1;bugger1;tekton3

Hypas

tekton1;tekton2;shroomer1

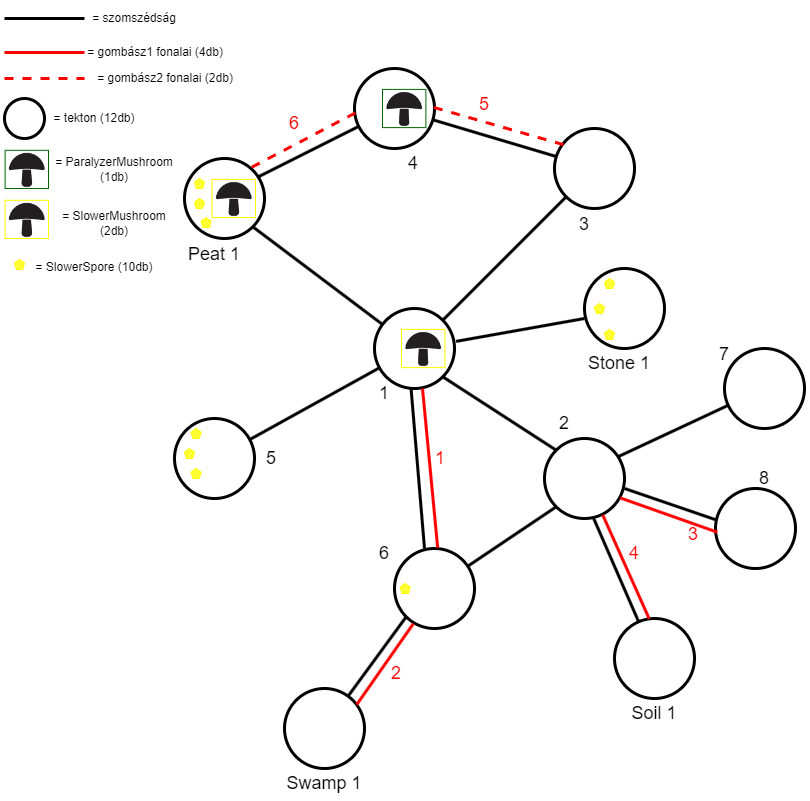
tekton2;tekton3;shroomer1

tekton2;tekton4;shroomer1

tekton3;tekton5;shroomer1

Spores

BoosterSpore;shroomer1;tekton3

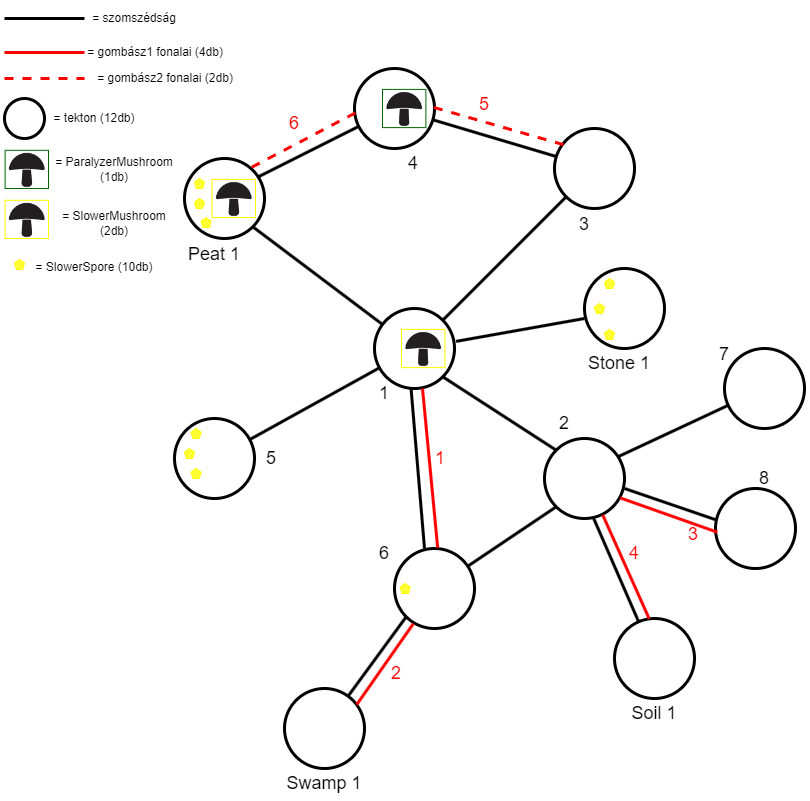
**arrange\_4**(tesztesetei: SHO\_01,..., SHO\_10,  
 SHO\_19, SHO\_25, SHO\_26, TEK\_1)

ARRANGE

Tektons

Tekton

Tekton

Tekton

Tekton

Tekton

Tekton

Tekton

Tekton

Stone

Peat

Soil

Swamp

Neighbours

tekton1;tekton2;tekton3;tekton5;tekton6;stone1;peat1

tekton2;tekton6;tekton7;tekton8;soil1

tekton4;tekton3;peat1

tekton6;swamp1

Shroomers

booster

booster

Mushrooms

BoosterMushroom;shroomer1;tekton1

BoosterMushroom;shroomer2;tekton4

Buggers

Strategies

Bugs

Hypas

tekton1;tekton6;shroomer1

tekton6;swamp1;shroomer1

tekton2;tekton8;shroomer1

tekton2;soil1;shroomer1

tekton4;tekton3;shroomer2

tekton4;peat1;shroomer2

Spores

BoosterSpore;shroomer1;tekton5

BoosterSpore;shroomer1;tekton5

BoosterSpore;shroomer1;tekton5

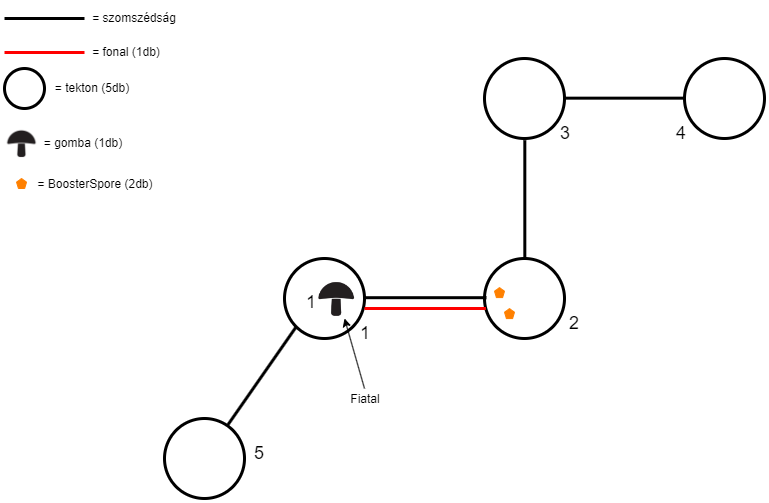
BoosterSpore;shroomer1;peat1

BoosterSpore;shroomer1;peat1

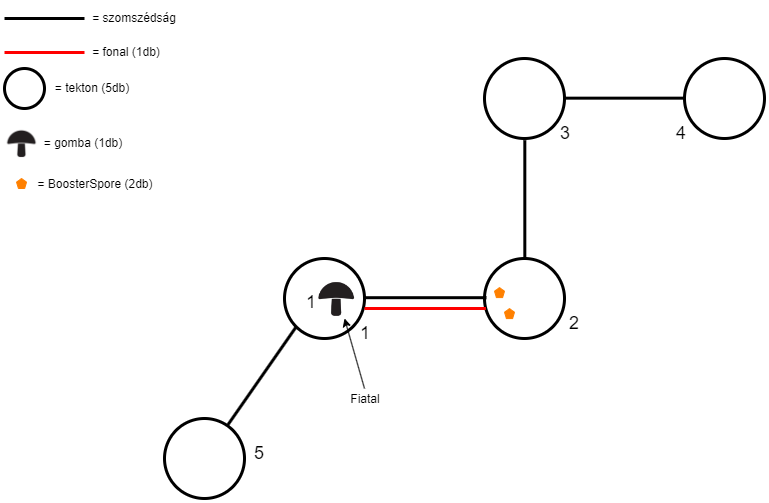
BoosterSpore;shroomer1;peat1

BoosterSpore;shroomer1;stone1

BoosterSpore;shroomer1;stone1

BoosterSpore;shroomer1;stone1

BoosterSpore;shroomer1;tekton6

**arrange\_5** (tesztesetei: SHO\_11,..., SHO\_18)

ARRANGE

Tektons

Tekton

Tekton

Tekton

Tekton

Tekton

Tekton

Neighbours

tekton1;tekton2;tekton5

tekton3;tekton2;tekton4

Shroomers

booster

Mushrooms

BoosterMushroom;shroomer1;tekton1

Buggers

Bugs

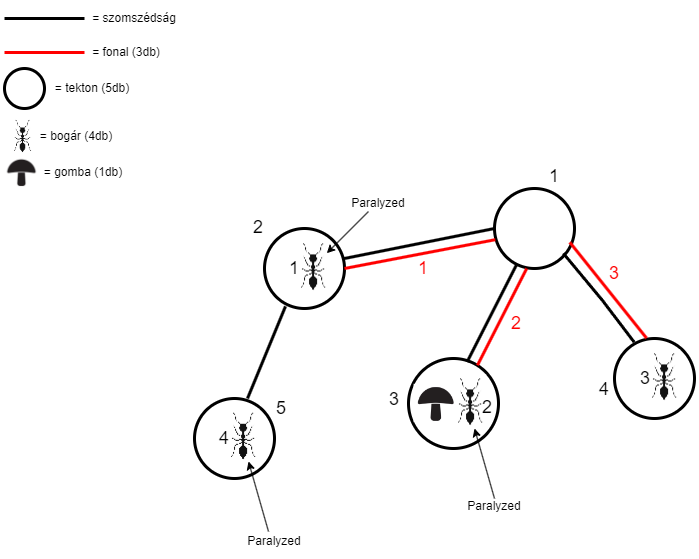
Hypas

tekton1;tekton2;shroomer1

Spores

BoosterSpore;shroomer1;tekton2

BoosterSpore;shroomer1;tekton2

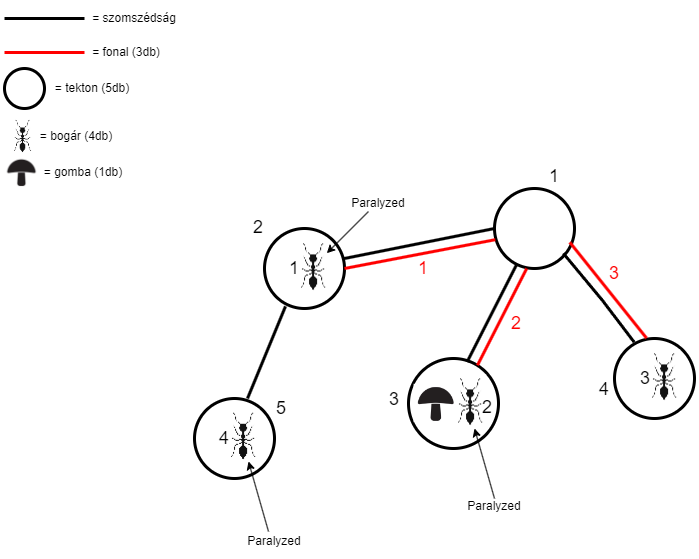
BoosterSpore;shroomer1;tekton2

**arrange\_6** (tesztesetei: SHO\_21,..., SHO\_24)

ARRANGE

Tektons

Tekton

Tekton

Tekton

Tekton

Tekton

Neighbours

tekton1;tekton2;tekton3;tekton4

tekton2;tekton5

Shroomers

booster

Mushrooms

BoosterMushroom;shroomer1;tekton3

Buggers

;

Strategies

Normal

Paralyzed

Paralyzed

Paralyzed

Bugs

paralyzed1;bugger1;tekton2

paralyzed1;bugger1;tekton3

paralyzed1;bugger1;tekton5

normal1;bugger1;tekton4

Hypas

tekton1;tekton2;shroomer1

tekton1;tekton3;shroomer1

tekton1;tekton4;shroomer1

Spores

### Spóra hatás nélküli Bogár mozog fonálon BUG\_01

* **Leírás**

2 normál tekton szomszédos, van közöttük fonál. Van egy bogarász neki egy spóra hatás nélküli bogara, ami az egyik tektonon van. A bogarász átmozgatja a bogarát. A bogár átkerül az üres tektonra.

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + A bogár normál (spóra hatás nélküli) mozgási képességének ellenőrzése
  + Szomszédos tektonok közötti fonálon történő átmozgás validálása
  + A forrás és cél tekton bug attribútumának megfelelő frissítése
  + Hibahelyek: A bogár helyzet-attribútumának nem megfelelő frissítése, tektonok bug attribútumának inkonzisztens állapota
* **Bemenet**

ARRANGE → Arrange\_1

ACT

skip;

move;bug1;tekton2

ASSERT

bug1;location;tekton2

tekton2;bug;bug1

tekton1;bug;null

* **Elvárt kimenet**

SUCCESS

SUCCESS

SUCCESS

### Spóra hatás nélküli Bogár mozogna, de nincs fonál (nem tud) BUG\_02

* **Leírás**

2 normál tekton szomszédos, nincs közöttük fonál. Van egy bogarász neki egy spóra hatás nélküli bogara, ami az egyik tektonon van. A bogarász átmozgatná a bogarát. **A bogár az eredeti tektonon marad**. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + A bogár mozgási feltételeinek ellenőrzése - fonál hiányában mozgás megtagadása
  + Sikertelen mozgási kísérlet utáni állapot validálása
  + Hibahelyek: A bogár pozíciójának változása annak ellenére, hogy nincs fonál, a mozgási feltételek érvényesítésének hiánya
* **Bemenet**

ARRANGE → Arrange\_1

ACT

skip;

move;bug1;tekton5

ASSERT

bug1;location;tekton1

tekton1;bug;bug1

tekton5;bug;null

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS

### Spóra hatás nélküli Bogár mozogna, de a tektonon amire mozogna van másik bogár (nem tud) BUG\_03

* **Leírás**

2 normál tekton szomszédos, van közöttük fonál. Van egy bogarász neki egy spóra hatás nélküli bogara, ami az egyik tektonon van és van egy másik bogarász akinek a bogara a másik tektonon van. A bogarász átmozgatná a bogarát. **A bogár az eredeti tektonon marad**. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Tektonok foglaltsági állapotának ellenőrzése mozgás előtt
  + Célteknon foglaltsága esetén a mozgás megtagadása
  + Két bogár ugyanazon a tektonon való tartózkodásának megakadályozása
  + Hibahelyek: bogár helytelen átmozgatása foglalt tektonra
* **Bemenet**

ARRANGE → Arrange\_1

ACT

skip;

move;bug1;tekton4

ASSERT

bug1;location;tekton1

bug2;location;tekton4

tekton1;bug;bug1

tekton4;bug;bug2

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS

### Gyorsító spóra hatású Bogár mozog közvetlen szomszédjára (Van fonál kapcsolat) BUG\_04

* **Leírás**

3 normál tekton szomszédos és fonállal össze vannak kötve (t1-t2-t3), egy bogarásznak van egy gyorsító spóra hatás alatt álló bogara, a bogarat elhelyezzük az egyik tektonra. A bogarász átmozgatja a bogarát egy közvetlen szomszédos tektonra. **A bogár átkerül a metódusban megadott tektonra**. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Gyorsító spóra hatás alatti bogár alapvető mozgásának ellenőrzése
  + Fonálon át szomszédos tektonra történő mozgás validálása
  + Hibahelyek: Gyorsító spóra hatás hibás érvényesítéséne, bogár helytelen átmozgatása, tektonok bug attribútumának inkonzisztens állapota
* **Bemenet**

ARRANGE → Arrange\_1

ACT

skip;

move;bug3;tekton2

ASSERT

bug3;location;tekton2

tekton2;bug;bug3

tekton3;bug;null

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS

### Gyorsító spóra hatású Bogár mozog 2 lépésre lévő tektonra (Van fonál kapcsolat) BUG\_05

* **Leírás**

3 normál tekton szomszédos és fonállal össze vannak kötve (t1-t2-t3), egy bogarásznak van egy gyorsító spóra hatás alatt álló bogara, a bogarat elhelyezzük az egyik szélső tektonra. A bogarász átmozgatja a bogarát egy 2 lépésre lévő tektonra. **A bogár átkerül a megadott tektonra**. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Gyorsító spóra hatás speciális képességének (2 lépés távolságra mozgás) validálása
  + Hibahelyek: Gyorsító hatás határértékének nem megfelelő kezelése, útvonal-ellenőrzés hibás implementációja
* **Bemenet**

ARRANGE → Arrange\_1

ACT

skip;

move;bug3;tekton6

ASSERT

bug3;location;tekton6

tekton6;bug;bug3

tekotn3;bug;null

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS

### Gyorsító spóra hatású Bogár mozogna 2 lépésre lévő tektonra de nincs fonál kapcsolat oda (nem tud) BUG\_06

* **Leírás**

3 normál tekton szomszédos és fonállal nincsenek összekötve, egy bogarásznak van egy gyorsító spóra hatás alatt álló bogara, a bogarat elhelyezzük az egyik szélső tektonra. A bogarász átmozgatná a bogarát egy 2 lépésre lévő tektonra. A bogár az eredeti tektonon marad. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Fonálkapcsolat hiányában a mozgás megtagadása gyorsító hatás ellenére is
  + Hibahelyek: Fonálkapcsolat ellenőrzésének kihagyása
* **Bemenet**

ARRANGE → Arrange\_1

ACT

skip;

move;bug3;tekton5

ASSERT

bugger;bug3;location;tekton3

tekton3;bug;bug3

tekton5;bug;null

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS

### Gyorsító spóra hatású Bogár mozogna 2 lépésre lévő tektonra (Van fonál kapcsolat) de egy másik bogár van ott (nem tud )BUG\_07

* **Leírás**

3 normál tekton szomszédos és fonállal össze vannak kötve (t1-t2-t3), . Van egy bogarász neki egy gyorsító spóra hatás alatt álló bogara, ami az egyik szélső tektonon van és van egy másik bogarász akinek a bogara a másik szélső tektonon van. A bogarász átmozgatná a bogarát. A bogár az eredeti tektonon marad. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Célteknon foglaltsági állapotának ellenőrzése gyorsító hatás alatt is
  + Két bogár ugyanazon a tektonon való tartózkodásának megakadályozása
  + Hibahelyek: Célteknon foglaltsági vizsgálatának mellőzése gyorsító hatás esetén
* **Bemenet**

ARRANGE → Arrange\_1

ACT

skip;

move;bug3;tekton1

ASSERT

bug3;location;tekton3

bug1;location;tekton1

tekton3;bug;bug3

tekotn1;bug;bug2

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS

### Gyorsító spóra hatású Bogár mozogna 3 lépésre lévő tektonra (Van fonál kapcsolat, de nem tud)BUG\_08

* **Leírás**

3 normál tekton szomszédos és fonállal össze vannak kötve, egy bogarásznak van egy gyorsító spóra hatás alatt álló bogara, a bogarat elhelyezzük az egyik szélső tektonra. A bogarász átmozgatná a bogarát egy 3 lépésre lévő tektonra. A bogár az eredeti tektonon marad. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Gyorsító spóra hatás maximális távolság-korlátjának érvényesítése
  + 2 lépésnél nagyobb távolságra történő mozgás megtagadása
  + Hibahelyek: Gyorsító hatás hibás megvalósítása, távolságszámítás hibás implementációja
* **Bemenet**

ARRANGE → Arrange\_1

ACT

skip;

move;bug3;tekton7

ASSERT

bug3;location;tekton3

tekton3;bug;bug3

tekotn7;bug;null

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS

### Lassító spóra hatású Bogár mozog szomszédos tektonra(Van fonál kapcsolat), még nem mozgott mióta spóra hatás alatt áll BUG\_09

* **Leírás**

2 normál tekton szomszédos, van közöttük fonál. Van egy bogarász neki egy lassító spóra hatás alatt álló bogara, ami még nem mozgott és az egyik tektonon van. A bogarász átmozgatja a bogarát. A bogár az eredeti tektonon marad. Sikertelen

* + Ellenőrzött funkcionalitás, várható hibahelyek
  + Lassító spóra hatás alatti első mozgás lehetőségének validálása
  + Mozgásszámláló inicializálásának ellenőrzése lassító hatás kezdetekor
  + Hibahelyek: Lassító hatás mozgásszámlálójának nem megfelelő kezelése, első mozgás korlátozása
* **Bemenet**

ARRANGE → Arrange\_1

ACT

skip;

move;bug4;tekton6

ASSERT

bug4;location;tekton6

tekton6;bug;bug4

tekotn8;bug;null

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS

### Lassító spóra hatású Bogár mozogna szomszédos tektonra(Van fonál kapcsolat), már mozgott mióta spóra hatás alatt áll így nem tud BUG\_10

* **Leírás**

2 normál tekton szomszédos, van közöttük fonál. Van egy bogarász neki egy lassító spóra hatás alatt álló bogara, ami már mozgott és az egyik tektonon van. A bogarász átmozgatná a bogarát. A bogár az eredeti tektonon marad. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Lassító spóra hatás korlátozásának érvényesítése második mozgási kísérlet esetén
  + Mozgásszámláló helyes inkrementálása korábbi mozgás után
  + Lassító hatás korrekt időtartamának ellenőrzése
  + Hibahelyek: Mozgásszámláló hibás nyilvántartása, lassító hatás állapotának helytelen tárolása, mozgási korlátozás helytelen alkalmazása
* **Bemenet**

ARRANGE → Arrange\_1

ACT

skip;

move;bug4;tekton6

skip;

move;bug4;tekton7

ASSERT

bug4;location;tekton6

tekton6;bug;bug4

tekotn7;bug;null

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatása alatt) elfogyaszt egy booster Spórát , ami a bogár tektonján van BUG\_11

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem áll spóra hatás alatt és egy booster Spóra. A bogarász bogara megeszi a spórát. A spóra törlődik és a bogár strategy-je boosted lesz. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Bogár spórafogyasztási képességének ellenőrzése
  + Booster spóra hatásának érvényesítése a bogár stratégiájára
  + Elfogyasztott spóra sikeres eltávolítása a tektonról
  + Hibahelyek: Spóra törlésének elmulasztása, bogár stratégiájának helytelen módosítása, spóra hatás érvényesítésének hiánya
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug1;boosterspore1

ASSERT

bug1;strategy;boosted2

tekton1;spores;slowerspore1;paralyzedspore1;biteblockerspore1;proliferatingspore1

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem normál vagy biteBlocked hatás alatt áll) elfogyasztana egy booster Spórát , ami a bogár tektonján van (nem tud)BUG\_12

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem nem normál vagy biteBloceked hatás alatt áll és egy booster Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Spórafogyasztás nem megengedése bármilyen korábbi spóra hatása alatt.
  + Spórafogyasztási feltételek validálása különböző stratégiák esetén
  + Hibahelyek: Bogár aktuális stratégiájának figyelmen kívül hagyása spórafogyasztáskor, stratégiakorlátozások helytelen implementációja
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug2;boosterspore2

ASSERT

bug2;strategy;booster2

tekton2;spores;boosterspore2;slowerspore2;paralyzedspore2;biteblockerspore2;proliferatingspore2

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatása alatt) elfogyasztana egy booster Spórát, ami nem a bogár tektonján van (nem tud) BUG\_13

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem áll spóra hatás alatt é egy másik tektonon egy booster Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár strategy-je is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Spórafogyasztás helyhez kötöttségének ellenőrzése
  + Különböző tektonokon lévő bogár és spóra esetén a fogyasztás megtagadása
  + Hibahelyek: Bogár és spóra helyzetének nem megfelelő validálása, pozícióellenőrzés kihagyása
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug1;boosterspore2

ASSERT

bug1;strategy;normal1

tekton2;spores;boosterspore2;slowerspore2;paralyzedspore2;biteblockerspore2;proliferatingspore2

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatása alatt) elfogyaszt egy slower Spórát ami a bogár tektonján van BUG\_14

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem áll spóra hatás alatt egy bogara és egy slower Spóra. A bogarász bogara megeszi a spórát. A spóra törlődik és a bogár strategy-je slowed lesz. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Slower spóra elfogyasztásának és hatásának validálása
  + Bogár stratégiájának megfelelő módosítása slower hatásra
  + Spóra sikeres eltávolítása a tektonról fogyasztás után
  + Hibahelyek: Slower spóra törlésének elmulasztása, bogár stratégiájának hibás frissítése
* **Bemenet**

ARRANGE→ Arrange\_2

ACT

skip;

eat;bug1;slowerspore1

ASSERT

bug1;strategy;slowed1

tekton1;spores;boosterspore1;paralyzedspore1;biteblockerspore1;proliferatingspore1

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem normál vagy biteBlocked hatás alatt áll) elfogyasztana egy slower Spórát ami a bogár tektonján van (nem tud) BUG\_15

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem normál vagy biteBloceked hatás alatt áll és egy slower Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Bogár aktuális stratégiájának hatása a spórafogyasztási képességre
  + Nem normál stratégiájú bogár spórafogyasztás letiltása.
  + Hibahelyek: Bogár strtégiájának figyelmen kívül hagyása, spórafogyasztási jogosultságok hibás implementációja
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug2;slowerspore2

ASSERT

bug2;strategy;booster2

tekton2;spores;boosterspore2;slowerspore2;paralyzedspore2;biteblockerspore2;proliferatingspore2

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatása alatt) elfogyasztana egy slower Spórát ami nem a bogár tektonján van (nem tud) BUG\_16

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem áll spóra hatás alatt é egy másik tektonon egy slower Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár strategy-je is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Slower spóra fogyasztásának helyhez kötöttségének ellenőrzése
  + Lokáció alapú korlátozások érvényesítése spórafogyasztáskor
  + Hibahelyek: Bogár és slower spóra pozícióellenőrzésének mellőzése, helyzetellenőrzés hibás implementációja
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug1;slowerspore2

ASSERT

bug1;strategy;normal1

tekton2;spores;boosterspore2;slowerspore2;paralyzedspore2;biteblockerspore2;proliferatingspore2

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatása alatt) elfogyaszt egy paralyzer Spórát, ami a bogár tektonján van BUG\_17

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem áll spóra hatás alatt egy bogara és egy paralyzer Spóra. A bogarász bogara megeszi a spórát. A spóra törlődik és a bogár strategy-je paralyzed lesz. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Paralyzer spóra elfogyasztásának és hatásának érvényesítése
  + Bogár stratégiájának megfelelő módosítása paralyzed hatásra
  + Spóra sikeres eltávolítása a tektonról fogyasztás után
  + Hibahelyek: Paralyzer spóra hatásának helytelen alkalmazása, bogár stratégiájának hibás frissítése
* **Bemenet**

ARRANGE→ Arrange\_2

ACT

skip;

eat;bug1;paralyzerspore1

ASSERT

bug1;strategy;paralyzed1

tekton1;spores;boosterspore1;slowerspore1;biteblockerspore1;proliferatingspore1

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem normál vagy biteBlocked hatás alatt áll) elfogyasztana egy paralyzer Spórát, ami a bogár tektonján van (nem tud) BUG\_18

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem normál vagy biteBloceked hatás alatt áll és egy paralyzer Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Stratégia alapú korlátozások érvényesítése spórafogyasztáskor
  + Nem normál stratégiájú bogár spórafogyasztási képességének validálása
  + Hibahelyek: Paralyzer spóra fogyasztási feltételeinek helytelen ellenőrzése, stratégia hatásának figyelmen kívül hagyása
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug2;paralyzerspore2

ASSERT

bug2;strategy;booster2

tekton2;spores;boosterspore2;slowerspore2;paralyzerspore2;biteblockerspore2;proliferatingspore2

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatása alatt) elfogyaszt egy paralyzer Spórát, ami nem a bogár tektonján van (nem tud) BUG\_19

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem áll spóra hatás alatt é egy másik tektonon egy paralyzer Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár strategy-je is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Paralyzer spóra fogyasztásának helyhez kötöttségének validálása
  + Távolsági korlátozások érvényesítése különböző tektonokon található bogár és paralyzer spóra esetén
  + Hibahelyek: Bogár és paralyzer spóra helyzetellenőrzésének mellőzése, lokációvalidálás hibás implementációja
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug1;paralyzerspore2

ASSERT

bug1;strategy;normal1

tekton2;spores;boosterspore2;slowerspore2;paralyzedspore2;biteblockerspore2;proliferatingspore2

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatása alatt) elfogyaszt egy biteBlocker Spórát, ami a bogár tektonján van BUG\_20

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem áll spóra hatás alatt egy bogara és egy biteBlocker Spóra. A bogarász bogara megeszi a spórát. A spóra törlődik és a bogár strategy-je biteBlocked lesz. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + BiteBlocker spóra elfogyasztásának és hatásának érvényesítése
  + Bogár stratégiájának megfelelő módosítása biteBlocked hatásra
  + Spóra sikeres eltávolítása a tektonról fogyasztás után
  + Hibahelyek: BiteBlocker spóra hatásának helytelen alkalmazása, bogár stratégiájának hibás frissítése
* **Bemenet**

ARRANGE→ Arrange\_2

ACT

skip;

eat;bug1;biteblockerspore1

ASSERT

bug1;strategy;biteblocked1

tekton1;spores;boosterspore1;slowerspore1;paralyzerspore1;proliferatingspore1

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem normál vagy biteBlocked hatás alatt áll) elfogyasztana egy biteBlocker Spórát, ami a bogár tektonján van (nem tud) BUG\_21

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem normál vagy biteBloceked hatás alatt áll és egy biteBlocker Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Stratégia alapú korlátozások érvényesítése biteBlocker spóra fogyasztásakor
  + Nem megfelelő stratégiájú bogár fogyasztási korlátainak validálása
  + Hibahelyek: Bogár stratégiájának figyelmen kívül hagyása, spórafogyasztási feltételek helytelen implementációja
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug2;biteblockerspore2

ASSERT

bug2;strategy;booster2

tekton2;spores;boosterspore2;slowerspore2;paralyzerspore2;biteblockerspore2;proliferatingspore2

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatása alatt) elfogyasztana egy biteBlocker Spórát, ami nem a bogár tektonján van (nem tud) BUG\_22

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem áll spóra hatás alatt é egy másik tektonon egy biteBlocker Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár strategy-je is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + BiteBlocker spóra fogyasztásának helyhez kötöttségének validálása
  + Hibahelyek: Bogár és spóra pozícióellenőrzésének mellőzése, helyzetellenőrzés hibás implementációja
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug1;biteblockerspore2

ASSERT

bug1;strategy;normal1

tekton2;spores;boosterspore2;slowerspore2;paralyzerspore2;biteblockerspore2;proliferatingspore2

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatás alatt ) elfogyaszt egy Proliferating Spórát, ami a bogár tektonján van és van szomszédos üres tekton BUG\_23

* **Leírás**

2 normál szomszédos tekton van. Az egyik normál tektonon van egy Bogarász, egy bogara ami nem áll spóra hatás alatt és egy Prolifrater Spóra. A másik tekton üres. A bogarász bogara megeszi a spórát. A spóra törlődik és a másik szomszédjára mivel üres létrejön egy új Bogár a bogarásznak. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Proliferating spóra speciális hatásának (új bogár létrehozása) ellenőrzése
  + Szomszédos üres tekton detektálása
  + Új bogár létrehozása megfelelő tulajdonossal
  + Spóra sikeres eltávolítása a tektonról fogyasztás után
  + Hibahelyek: Új bogár helytelen létrehozása, szomszédtekton keresés hibás implementációja, tulajdonosi attribútumok helytelen beállítása
* **Bemenet**

ARRANGE→ Arrange\_2

ACT

skip;

eat;bug1;proliferatingspore1

ASSERT

bug1;strategy;normal1

bug1;location;tekton1

bug4;location;tekton4

tekton1;bug;bug1

tekton4;bug;bug4

tekton1;spores;boosterspore1;slowerspore1;paralyzerspore1;biteblockerspore1

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatás alatt) elfogyaszt egy Proliferating Spórát, ami a bogár tektonján van és nincs szomszédos üres tekton (nem szomszédos tektonra osztódik) BUG\_24

* **Leírás**

3 normál szomszédos tekton van (t1-t2-t3). Az egyik szélső normál tektonon van egy Bogarász egy bogara ami nem áll spóra hatás alatt és egy Prolifrater Spóra. A középső tektonon van egy másik bogarásznak egy bogara. A másik szélső tekton üres. A szélső tektonon lévő bogár megeszi a spórát. A spóra törlődik és a másik szélső nem szomszédos tektonra létrejön egy új Bogár a bogarásznak. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Proliferating spóra viselkedésének ellenőrzése szomszédos üres tekton hiányában
  + Nem szomszédos tektonra történő osztódás validálása
  + Új bogár létrehozása távolabbi üres tektonon
  + Hibahelyek: Alternatív üres tekton keresés hibás implementációja, osztódás helytelen végrehajtása foglalt szomszédos tektonok esetén
* **Bemenet**

ARRANGE→ Arrange\_2

ACT

skip;

eat;bug3;proliferatingspore3

ASSERT

bug1;strategy;normal2

bug1;location;tekton3

bug4;location;tekton4

tekton3;bug;bug1

tekton4;bug;bug4

tekton3;spores;null

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS

### Bogár (nem normál vagy biteBlocked hatás alatt áll) elfogyasztana egy Proliferating Spórát, ami a bogár tektonján van és van szomszédos üres tekton (nem tud) BUG\_25

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem normál vagy biteBloceked hatás alatt áll és egy proliferater Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Stratégia alapú korlátozások érvényesítése proliferating spóra fogyasztásakor
  + Nem normál stratégiájú bogár spórafogyasztási képtelenségének validálása
  + Hibahelyek: Stratégiaállapot figyelmen kívül hagyása, spórafogyasztási feltételek helytelen implementációja
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug2;proliferatingspore2

ASSERT

bug2;strategy;booster2

tekton2;spores;boosterspore2;slowerspore2;paralyzerspore2;biteblockerspore2;proliferatingspore2

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem áll Spóra hatása alatt) elfogyasztana egy Proliferating Spórát, ami nem a bogár tektonján van (nem tud) BUG\_26

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami nem áll spóra hatás alatt é egy másik tektonon egy proliferating Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár strategy-je is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Proliferating spóra fogyasztásának helyhez kötöttségének validálása
  + Lokációs korlátozások érvényesítése különböző tektonokon lévő bogár és spóra esetén
  + Hibahelyek: Bogár és spóra helyzetellenőrzésének mellőzése, lokációvalidálás hibás implementációja
* **Bemenet**

ARRANGE → Arrange\_2

ACT

skip;

eat;bug1;proliferatingspore2

ASSERT

bug1;strategy;normal1

tekton2;spores;boosterspore2;slowerspore2;paralyzedspore2;biteblockerspore2;proliferatingspore2

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem biteblocked Spóra hatású) Elharap egy fonalat, ami a bogár tektonján van. Valamennyi skip után a fonál meghalt BUG\_27

* **Leírás**

2 Szomszédos normal tekton között van fonál kapcsolat. Az egyik tektonon van egy bogarásznak egy nem biteBlocked strategy-jű bogara. A bogarász elharaptatja a fonalat majd skippelődik valamennyi kör. Ezután a fonál eltűnik. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Bogár fonálharapási képességének ellenőrzése
  + Fonál dieAfterBite attribútumának megfelelő kezelése
  + Körök múlásának hatása a harapott fonálra
  + Elharapott fonál sikeres eltávolítása meghatározott idő után
  + Hibahelyek: Fonál diaAfterBite attribútumának helytelen kezelése, fonál eltávolításának elmulasztása, kör számlálásának hibája
* **Bemenet**

ARRANGE → Arrange\_3

ACT

skip;

bite;bug1;hypa1

skip;

skip;

skip;

skip;

ASSERT

tekton1;hypas;null

shroomer1;hypas;hypa2;hypa3;hypa4

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Bogár (nem biteblocked Spóra hatású) Elharapna egy fonalat, de nem a bogár tektonján van. (nem tud) BUG\_28

* **Leírás**

3 szomszédos normal tekton szomszédos (t1-t2-t3) egy bogarásznak egy nem biteblocked hatás alatt álló bogara a t1-es tektonon van. A t2-t3 között van fonál kapcsolat. A bogarász elharaptatná a t2-t3 közötti fonalat a bogárral. A fonál megmarad. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Fonálharapás helyhez kötöttségének validálása
  + Távolsági korlátozások érvényesítése különböző tektonokon lévő bogár és fonál esetén
  + Hibahelyek: Bogár és fonál helyzetellenőrzésének mellőzése, pozícióvalidálás hibás implementációja
* **Bemenet**

ARRANGE → Arrange\_3

ACT

skip;

bite;bug1;hypa2

skip;

skip;

skip;

skip;

ASSERT

tekton2;hypas;hypa1;hypa2;hypa3

shroomer1;hypas;hypa1;hypa2;hypa3;hypa4

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### BiteBlocked Spóra hatású bogár elharapna egy fonalat, ami a bogár tektonján van (nem tud) BUG\_29

* **Leírás**

2 Szomszédos normal tekton között van fonál kapcsolat. Az egyik tektonon van egy bogarásznak egy biteBlocked strategy-jű bogara. A bogarász elharaptatná a fonalat. A fonál megmarad. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + BiteBlocked stratégiájú bogár fonálharapási képtelenségének ellenőrzése
  + Stratégiafüggő korlátozások érvényesítése fonálharapáskor
  + Hibahelyek: Bogár stratégiájának figyelmen kívül hagyása harapáskor, stratégiahatás hibás implementációja
* **Bemenet**

ARRANGE → Arrange\_3

ACT

skip;

bite;bug2;hypa3

skip;

skip;

skip;

skip;

ASSERT

tekton2;hypas;hypa1;hypa2;hypa3

shroomer1;hypas;hypa1;hypa2;hypa3;hypa4

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Paralyzed Spóra hatású bogár elharapna egy fonalat, ami a bogár tektonján van (nem tud) BUG\_30

* **Leírás**

2 Szomszédos normal tekton között van fonál kapcsolat. Az egyik tektonon van egy bogarásznak egy paralyzed strategy-jű bogara. A bogarász elharaptatná a fonalat. A fonál megmarad. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Paralyzed stratégiájú bogár fonálharapási képtelenségének validálása
  + Stratégiafüggő akciótiltások érvényesítése
  + Hibahelyek: Paralyzed állapot hatásának figyelmen kívül hagyása, stratégiakorlátozások helytelen implementációja
* **Bemenet**

ARRANGE → Arrange\_3

ACT

skip;

bite;bug3;hypa4

skip;

skip;

skip;

skip;

ASSERT

tekton3;hypas;hypa2;hypa4

shroomer1;hypas;hypa1;hypa2;hypa3;hypa4

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Paralyzed Spóra hatású bogár mozogna egy szomszédos tektonra ahol van fonál kapcsolat (nem tud) BUG\_31

* **Leírás**

2 Szomszédos normal tekton között van fonál kapcsolat. Az egyik tektonon van egy bogarásznak egy paralyzed strategy-jű bogara. A bogarász mozgatná a bogarat. A bogár az eredeti tektonon marad. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Paralyzed stratégiájú bogár mozgásképtelenségének ellenőrzése
  + Stratégiafüggő mozgástiltás érvényesítése
  + Hibahelyek: Paralyzed állapot mozgásra gyakorolt hatásának figyelmen kívül hagyása, stratégiakorlátozások hibás implementációja
* **Bemenet**

ARRANGE → Arrange\_3

ACT

skip;

move;bug3;tekton5

ASSERT

tekton5;bug;null

bug3;location;tekton3

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Paralyzed Spóra hatású bogár elfogyasztana egy spórát ami a tektonján van (nem tud) BUG\_32

* **Leírás**

Egy normál tektonon van egy Bogarász, egy bogara ami paralyzed spóra hatás alatt és egy Spóra. A bogarász bogara megenné a spórát. A spóra megmarad és a bogár strategy-je is változatlan. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Paralyzed stratégiájú bogár spórafogyasztási képtelenségének validálása
  + Stratégiafüggő fogyasztási tiltások érvényesítése
  + Hibahelyek: Paralyzed állapot fogyasztásra gyakorolt hatásának figyelmen kívül hagyása, stratégiakorlátozások helytelen implementációja
* **Bemenet**

ARRANGE → Arrange\_3

ACT

skip;

eat;bug3;boosterspore1

ASSERT

tekton3;spoeres;boosterspore1

bug3;strategy;paralyzed1

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Gombász közeli fonál növesztés SHO\_1

* **Leírás**

Gombász sikerese növeszt egy gombafonalat, spóra nélküli tektonról.

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Gombafonál sikeres létrehozása két szomszédos tekton között
  + Fonál helyes kapcsolódása mindkét végén a megfelelő tektonokhoz
  + Fonál hozzáadása a gombász fonálkészletéhez
  + Hibahelyek: Fonál nem jön létre, fonál nem kapcsolódik mindkét tektonhoz, fonál nem kerül a gombász fonálkészletébe
* **Bemenet**

ARRANGE → Arrange\_4

ACT

growhypa;tekton1;tekton2

ASSERT

hypa7;end1;tekton1

hypa7;end2;tekton2

shroomer1;hypas;hypa1;hypa2;hypa3;hypa4;hypa7

tekton2;hypas;hypa3;hypa4;hypa7

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS

### Gombász távoli fonál növesztés SHO\_2

* **Leírás**

Gombász sikeresen növeszt egy gombafonalat (vagyis kettőt), spórás tektonról. Megvalósul mindkettő fonál növesztése

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Többszörös fonálnövesztés spórás tektonon keresztül
  + Két új fonál sikeres létrehozása és megfelelő kapcsolódása
  + Mindkét fonál helyes regisztrálása a gombász fonálkészletébe
  + Hibahelyek: Valamelyik fonál nem jön létre, fonalak végpontjai nem megfelelően kapcsolódnak, a fonalak nem kerülnek a gombász készletébe
* **Bemenet**

ARRANGE → Arrange\_4

ACT

growhypafar;tekton6;tekton2;tekton7

ASSERT

hypa7;end1;tekton6

hypa7;end2;tekton2

hypa8;end1;tekton2

hypa8;end2;tekton7

shroomer1;hypas;hypa1;hypa2;hypa3;hypa4;hypa7;hypa8

tekton2;hypas;hypa3;hypa4;hypa7;hypa8

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS

### Gombász párhuzamos fonalai SHO\_3

* **Leírás**

Gombász növeszteni akar fonalat olyan 2 tekton közé, amelyek között már van neki fonala, sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Párhuzamos fonálnövesztés megakadályozása két olyan tekton között, ahol már van fonál
  + A gombász fonálkészlete nem változik sikertelen növesztés esetén
  + Hibahelyek: Duplikált fonalak létrejötte, meglévő fonalak felülírása
* **Bemenet**

ARRANGE → Arrange\_4

ACT

growhypa;tekton1;tekton6

ASSERT

shroomer1;hypas;hypa1;hypa2;hypa3;hypa4

* **Elvárt kimenet**

SUCCESS

### Gombász rossz tektonra növeszt fonalat SHO\_4

* **Leírás**

Gombász növeszteni akar fonalat olyan tektonra ami nem szomszédos, sikertelen.

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Nem szomszédos tektonok közti fonálnövesztés megakadályozása
  + Gombász fonálkészlete változatlan marad sikertelen növesztés esetén
  + Hibahelyek: Hibás szomszédellenőrzés, nem szomszédos tektonok közti illegális fonálnövesztés
* **Bemenet**

ARRANGE → Arrange\_4

ACT

growhypa;tekton1;tekton7

ASSERT

shroomer1;hypas;hypa1;hypa2;hypa3;hypa4

* **Elvárt kimenet**

SUCCESS

### Gombász fonálnövesztés SHO\_5

* **Leírás**

Gombász növeszteni akar egy fonalat egy olyan (nem Peat) szomszédos tektonra, amellyen már van valamely más gombásznak fonala. Sikertelen a növesztés

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Más gombász által már használt tektonra való fonálnövesztés megakadályozása (nem Peat típusú tektonon)
  + Hibahelyek: Illegális fonálnövesztés foglalt tektonra, fonálkészlet helytelen frissítése
* **Bemenet**

ARRANGE → Arrange\_4

ACT

growhypa;tekton1;tekton3

ASSERT

shroomer1;hypas;hypa1;hypa2;hypa3;hypa4

* **Elvárt kimenet**

SUCCESS

### Gombász Peat-re fonálnövesztés SHO\_6

* **Leírás**

Gombász növeszt egy fonalat egy Peat tektonra, amellyen már van valamely más gombásznak fonala. Sikeres.

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Peat típusú tektonra való fonálnövesztés engedélyezése, még ha másik gombász fonala már kapcsolódik is hozzá
  + Fonál megfelelő hozzáadása a Peat tekton fonállistájához
  + Hibahelyek: Peat típusú tektonra való fonálnövesztés hibás kezelése, fonál hozzáadásának elmulasztása a tekton listájához
* **Bemenet**

ARRANGE → Arrange\_4

ACT

growhypa;tekton1;peat1

ASSERT

hypa7;end1;tekton1

hypa7;end2;peat1

shroomer1;hypas;hypa1;hypa2;hypa3;hypa4;hypa7

peat1;hypas;hypa6;hypa7

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS

### Gombatest növesztés (automatikus) SHO\_7

* **Leírás**

Gombász sikeresen növeszt fonalat egy nem Stone tektonra, ahol már van 3 spórája, de nincs még senkinek gombateste (sikeres gombatest növesztés)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Automatikus gombatest növesztés 3 spórával rendelkező tektonon fonálnövesztés után
  + Gombatest helyes pozicionálása a tektonon
  + Gombatest helyes hozzáadása a gombász gombatest-készletéhez
  + Hibahelyek: Gombatest növesztésének hibája, rossz helyre pozicionált gombatest, gombatest hiányzik a gombász készletéből
* **Bemenet**

ARRANGE → Arrange\_4

ACT

growhypa;tekton1;tekton5

ASSERT

boostermushroom2;location;tekton5

shroomer1;mushrooms;boostermushroom1;boostermushroom2

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Gombatest növesztés (automatikus) SHO\_8

* **Leírás**

Gombász sikeresen növeszt fonalat egy Stone tektonra, ahol már van 3 spórája (sikertelen gombatest növesztés)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Stone típusú tektonon való automatikus gombatest növesztés megakadályozása, még ha van is elég spóra
  + Gombatest készlet változatlan marad sikertelen növesztés esetén
  + Hibahelyek: Gombatest illegális létrehozása Stone tektonon, gombatest készlet helytelen frissítése
* **Bemenet**

ARRANGE → Arrange\_4

ACT

growhypa;tekton1;stone1

ASSERT

shroomer1;mushrooms;boostermushroom1

* **Elvárt kimenet**

SUCCESS

### Gombatest növesztés (automatikus) fonálnövesztés után SHO\_9

* **Leírás**

Gombász sikeresen növeszt fonalat egy Peat tektonra, ahol már van 3 spórája, de van ott már másnak gombateste (sikertelen gombatest növesztés)

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Más gombatesttel már rendelkező Peat tektonon az automatikus gombatest növesztés megakadályozása
  + Gombatest készlet változatlan marad sikertelen növesztés esetén
  + Hibahelyek: Gombatest illegális létrehozása foglalt tektonon, gombatest készlet helytelen frissítése
* **Bemenet**

ARRANGE → Arrange\_4

ACT

growhypa;tekton1;peat1

ASSERT

shroomer1;mushrooms;boostermushroom1

* **Elvárt kimenet**

SUCCESS

### Gombafonál hálózathoz visszakapcsolása SHO\_10

* **Leírás**

Gombásznak van egy fonala ami nem csatlakozik hálózatához egy köre, ehhez a gombász növeszt egy fonalat. Az adott hypa mégsem hal meg a következő körben

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Elszigetelt fonál visszakapcsolása a gombász hálózatához új fonállal
  + Elszigetelt fonál túlélése sikeres visszakapcsolás esetén
  + Hibahelyek: Elszigetelt fonál idő előtti eltávolítása, fonálhálózat kapcsolódási logikájának hibája
* **Bemenet**

ARRANGE → Arrange\_4

ACT

growhypa;tekton1;tekton2

skip;

skip;

skip;

ASSERT

shroomer1;hypas;hypa1;hypa2;hypa3;hypa4

* **Elvárt kimenet**

SUCCESS

### Fiatal gombatest spóra szórása SHO\_11

* **Leírás**

Gombász spórát szór. Van még elegendő spórája hogy ne is haljon meg, készen áll spóra szórásra, kora fiatal, szomszédos mezőre akar szórni. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Fiatal gombatest sikeres spóraszórása közvetlen szomszédos mezőre
  + Spóra helyes hozzárendelése a gombászhoz
  + Spóra megfelelő elhelyezése a célmezőn
  + Hibahelyek: Spóraszórás hibás távolságszámítása, spóra hiányzik a célmezőről, spóra nem kapcsolódik a gombászhoz
* **Bemenet**

ARRANGE → Arrange\_5

ACT

throwspore;boostermushroom1;tekton5

ASSERT

boosterspore3;shroomer;shroomer1

tekton5;spores;boosterspore3

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Fiatal gombatest spóraszórása SHO\_12

* **Leírás**

Gombász spórát szórna. Van még elegendő spórája hogy ne is haljon meg, készen áll spóra szórásra, kora fiatal, nem szomszédos mezőre akar szórni. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Fiatal gombatest távolabbi mezőre való spóraszórásának megakadályozása
  + Célmező spóralistája változatlan marad sikertelen szórás esetén
  + Hibahelyek: Fiatal gombatest hatótávolságának hibás ellenőrzése, illegális spóraszórás
* **Bemenet**

ARRANGE → Arrange\_5

ACT

throwspore;boostermushroom1;tekton3

ASSERT

tekton3;spores;null

* **Elvárt kimenet**

SUCCESS

### Idős gombatest spóraszórása SHO\_13

* **Leírás**

Gombász spórát szór. Van még elegendő spórája hogy ne is haljon meg, készen áll spóra szórásra, kora idős, szomszédos mezőre akar szórni. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Idős gombatest sikeres spóraszórása közvetlen szomszédos mezőre
  + Spóra helyes hozzárendelése a gombászhoz
  + Spóra megfelelő elhelyezése a célmezőn
  + Hibahelyek: Idős gombatest spóraszórásának hibás kezelése, spóra hiányzik a célmezőről
* **Bemenet**

ARRANGE → Arrange\_5

ACT

skip;

skip;

skip;

skip;

skip;

throwspore;boostermushroom1;tekton5

ASSERT

boosterspore3;shroomer;shroomer1

tekton5;spores;boosterspore3

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Idős gombatest spóraszórása SHO\_14

* **Leírás**

Gombász spórát szór. Van még elegendő spórája hogy ne is haljon meg, készen áll spóra szórásra, kora idős, második szomszédos mezóre akar szórni. Sikeres

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Idős gombatest sikeres spóraszórása második szomszédos mezőre (2 távolságra)
  + Spóra helyes hozzárendelése a gombászhoz
  + Spóra megfelelő elhelyezése a távolabbi célmezőn
  + Hibahelyek: Idős gombatest hatótávolságának hibás kezelése, távolabbi mezőre való spóraszórás hibája
* **Bemenet**

ARRANGE → Arrange\_5

ACT

skip;

skip;

skip;

skip;

skip;

throwspore;boostermushroom1;tekton3

ASSERT

boosterspore3;shroomer;shroomer1

tekton3;spores;boosterspore3

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Idős gombatest spóraszórása SHO\_15

* **Leírás**

Gombász spórát szórna. Van még elegendő spórája hogy ne is haljon meg, készen áll spóra szórásra, kora idős,nem szomszédos vagy második szomszédos mezóre akar szórni. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Idős gombatest túl távolra (2-nél nagyobb távolságra) való spóraszórásának megakadályozása
  + Célmező spóralistája változatlan marad sikertelen szórás esetén
  + Hibahelyek: Idős gombatest hatótávolság-korlátjának hibás ellenőrzése, illegális spóraszórás
* **Bemenet**

ARRANGE → Arrange\_5

ACT

skip;

skip;

skip;

skip;

skip;

throwspore;boostermushroom1;tekton4

ASSERT

tekton4;spores;null

* **Elvárt kimenet**

SUCCESS

### Gombatest nem áll készen spóraszórásra SHO\_17

* **Leírás**

Gombász spórát szór/szórna. Van még elegendő spórája hogy ne is haljon meg, nem áll készen spóra szórásra, Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Spóraszórásra nem kész gombatest spóraszórási kísérletének megakadályozása
  + Cooldown-ellenőrzés helyes működése
  + Célmező spóralistája változatlan marad sikertelen szórás esetén
  + Hibahelyek: Gombatest készenléti állapotának hibás ellenőrzése, illegális spóraszórás
* **Bemenet**

ARRANGE → Arrange\_5

ACT

throwspore;boostermushroom1;tekton2

throwspore;boostermushroom1;tekton5

ASSERT

tekton5;spores;null

* **Elvárt kimenet**

SUCCESS

### Gombatest növesztés (automatikus) spóra szórás után SHO\_18

* **Leírás**

Gombász spórát szór. Van még elegendő spórája hogy ne is haljon meg, készen áll spóra szórásra, kora fiatal, szomszédos mezőre szór. Sikeres spóra szórás, szomszédos mezőn van már 2 spórája és vezet oda fonala, ezért sikeres gombatest növesztés

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Automatikus gombatest növesztés sikeres spóraszórás után, amikor a célmezőn már 2 spóra van és fonál is vezet oda
  + Spóra megfelelő felhasználása a gombatest növesztéshez
  + Gombatest helyes pozicionálása és hozzáadása a gombász készletéhez
  + Hibahelyek: Automatikus gombatest növesztés hibája, spórák nem megfelelő felhasználása, gombatest hiányzik a gombász készletéből
* **Bemenet**

ARRANGE → Arrange\_5

ACT

throwspore;boostermushroom1;tekton2

ASSERT

tekton2;spores;null

shroomers1;mushrooms;boostermushroom1;mushroom2

boostermushroom2;location;tekton2

boostermushroom1;location;tekton1

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS

### Nem csatlakozott hypa elhalása SHO\_19

* **Leírás**

Gombász kör vége, egy hypa nem csatlakozik hálózathoz, eltelik két kör, az adott hypa meghal

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Hálózathoz nem csatlakozó fonál automatikus eltávolítása két kör elteltével
  + Fonál eltávolítása a gombász fonálkészletéből
  + Hibahelyek: Elszigetelt fonál túl korai vagy késői eltávolítása, fonálkészlet helytelen frissítése
* **Bemenet**

ARRANGE → Arrange\_4

ACT

skip;

skip;

skip;

skip;

skip;

ASSERT

shroomer1;hypas;hypa1;hypa2;hypa4

* **Elvárt kimenet**

SUCCESS

### Bénított bogár elfogyasztása SHO\_21

* **Leírás**

Gombafonál elfogyaszt egy bénított bogarat, akihez vezet fonala. Sikeres. Az adott tektonon nincs gombatest. Kinő egy új gombatest

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Bénított bogár sikeres elfogyasztása fonállal csatlakozó tektonról
  + Új gombatest automatikus növesztése a bogár helyén, ha ott még nincs gombatest
  + Bogár eltávolítása a tektonról
  + Új gombatest helyes hozzáadása a gombász készletéhez
  + Hibahelyek: Bogár elfogyasztásának hibája, gombatest nem jön létre, bogár nem távolítódik el a tektonról
* **Bemenet**

ARRANGE → Arrange\_6

ACT

eatbug;bug1

ASSERT

shroomer1;mushrooms;mushroom1;mushroom2

mushroom1;location;tekton3

mushroom2;location;tekton2

tekton2;bug;null

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS  
SUCCESS

### Bénított bogár elfogyasztása (gombatestes tektonon) SHO\_22

* **Leírás**

Gombafonál elfogyaszt egy bénított bogarat, akihez vezet fonala. Sikeres. Az adott tektonon van már gombatest. Nem nő ki egy új gombatest

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Bénított bogár sikeres elfogyasztása olyan tektonról, ahol már van gombatest
  + Gombatest készlet változatlan marad, új gombatest nem jön létre
  + Bogár eltávolítása a tektonról
  + Hibahelyek: Bogár elfogyasztásának hibája, felesleges gombatest létrehozása, bogár nem távolítódik el a tektonról
* **Bemenet**

ARRANGE → Arrange\_6

ACT

eatbug;bug2

ASSERT

shroomer1;mushrooms;mushroom1

mushroom1;location;tekton3

tekton3;bug;null

* **Elvárt kimenet**

SUCCESS  
SUCCESS  
SUCCESS

### Messzi bénított bogár elfogyasztása SHO\_23

* **Leírás**

Gombafonál elfogyasztana egy bénított bogarat, akihez nem vezet fonala. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Fonállal nem csatlakozó tektonon levő bogár elfogyasztásának megakadályozása
  + Gombász gombatest-készlete és a bogár helyzete változatlan marad sikertelen fogyasztás esetén
  + Hibahelyek: Fonálkapcsolat ellenőrzésének hibája, bogár illegális elfogyasztása
* **Bemenet**

ARRANGE → Arrange\_6

ACT

eatbug;bug4

ASSERT

shroomer1;mushrooms;mushroom1

tekton5;bug;bug4

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Nem bénított bogár elfogyasztása SHO\_24

* **Leírás**

Gombafonál elfogyasztana egy nem bénított bogarat. Sikertelen

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Nem bénított bogár elfogyasztásának megakadályozása
  + Gombász gombatest-készlete és a bogár helyzete változatlan marad sikertelen fogyasztás esetén
  + Hibahelyek: Bogár állapotának hibás ellenőrzése, nem bénított bogár illegális elfogyasztása
* **Bemenet**

ARRANGE → Arrange\_6

ACT

eatbug;bug3

ASSERT

shroomer1;mushrooms;mushroom1

tekton4;bug;bug3

* **Elvárt kimenet**

SUCCESS  
SUCCESS

### Swamp tekton megöli a fonalat SHO\_25

* **Leírás**

Gombafonál egy Swamp tekton és valamilyen Tekton között, csatlakozik Gombatesthez, a fonál 3 kör múlva eltűnik.

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Swamp tektonhoz kapcsolódó fonál automatikus eltávolítása 3 kör elteltével
  + Fonál eltávolítása a gombász fonálkészletéből
  + Hibahelyek: Swamp tekton hatásának hibás időzítése, fonál túl korai vagy késői eltávolítása
* **Bemenet**

ARRANGE → Arrange\_4

ACT

skip;

skip;

skip;

skip;

skip;

skip;

skip;

ASSERT

shroomer1;hypas;hypa1;hypa4

* **Elvárt kimenet**

SUCCESS

### Soil életben tart egy fonalat SHO\_26

* **Leírás**

Egy gombafonál egy Soil és valamely más Tekton között van, nem csatlakozik gombatesthez. A fonál nem hal meg 2 kör után.

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Soil tektonhoz kapcsolódó, hálózattól elszigetelt fonál megmaradása 2 kör után is
  + Fonál megmaradása a gombász fonálkészletében
  + Hibahelyek: Soil tekton fonálmegőrző hatásának hibás kezelése, fonál téves eltávolítása
* **Bemenet**

ARRANGE → Arrange\_4

ACT

skip;

skip;

skip;

skip;

skip;

ASSERT

shroomer1;hypas;hypa1;hypa2;hypa4

* **Elvárt kimenet**

SUCCESS

### Egy Tekton eltörik TEK\_1 Leírás

Egy adott Tekton eltörik, Ez megsemmisíti a rajta található Hypa objektumokat. Új (azonos leszármazott osztálybeli) Tekton objektumot hoz létre. A rajta lévő spórákat a régin hagyja, a szomszédok felét az új tektonhoz átkapcsolja. Az új tekont összekapcsolja a régi tektonnal.

* **Ellenőrzött funkcionalitás, várható hibahelyek**
  + Tekton törés
  + Hibahelyek: Hibás szomszéd szétosztás. Hibás spóra kezelés. Hibás új Tekton leszármazott objektum létrehozása.
* **Bemenet**

ARRANGE → Arrange\_4

ACT

break;stone1

ASSERT

stone1;neighbours;tekton1;stone2

boosterspore7;location;stone1

* **Elvárt kimenet**

SUCCESS  
SUCCESS

## **8.3 A tesztelést támogató programok tervei**

Nem használunk tesztelés támogató programot.

## **8.4** **Napló**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2025.03.31. 20:00 | 2 óra | Katona | TestCase-k írása |
| 2025.04.01. 12:00 | 2 óra | Kőműves | Strategy interface és az azt megvalósító osztályok, Bugger és Bug osztályok, tervei |
| 2025.04.03 8:00 | 8 óra | Katona | Arrange file-ok tervezése, tesztesetek kidolgozása |
| 2025.04.03 8:00 | 8 óra | Szabó | Arrange file-ok tervezése, tesztesetek kidolgozása Pszeudokódok írása |
| 2025.04.03 8:30 | 8 óra | Kőműves | Osztályok részletes tervei (Shroomer, Hypa, Spore, Mushroom és leszármazottaik), állapotgépek elkészítése |
| 2025.04.03. 13:00 | 3 óra | Fekete | Arrange rajzok |
| 2025. 04. 03. 12:00 | 4 óra | Lipóczi | Osztályleírások |
| 2025.04.11. 10:30 | 5 óra | Kőműves | MVC komponensek tervezése |
| 2025.04.11. 10:30 | 5 óra | Katona | MVC komponensek tervezése |
| 2025.04.11. 10:30 | 5 óra | Szabó | MVC komponensek tervezése |