# Részletes tervek

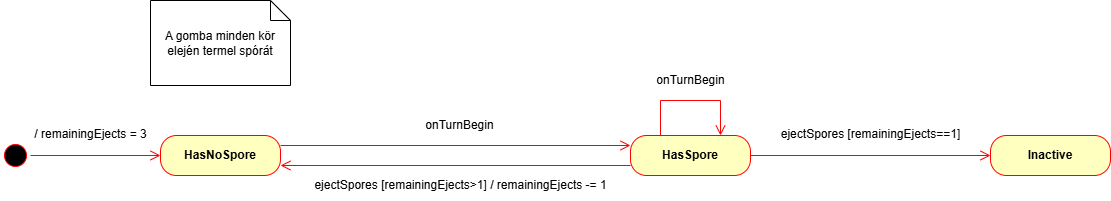
## Osztályok és metódusok tervei.

### MushroomBodyImpl

#### Felelősség

#### A gombatestekért felelős osztály. A gombatest a spórák termeléséért és kilövéséért felelős. 3 spórakilövés után inaktívvá válik, amely abban nyilvánul meg, hogy a remaingEjects változó értéke 0 lesz. A gombatest az utolsó kilövése előtt csak valamely szomszédjára lőhet spórát. Fejlettnek az utolsó kilövése során minősül, amely abban nyilvánul meg, hogy ekkor a szomszédja szomszédjára is tud lőni.

#### A gombatest állapotdiagramja:



#### Interfészek

Mushroom, MushroomBody, TurnBeginSubscriber

#### Attribútumok

#### -remainingEjects: int

A megmaradt spórakilövések számát tároló változó. Alapértelmezett értéke 3.

#### -location: Tecton

A gombatest elhelyezkedése szerinti tektont tároló változó.

#### -mushroomSpores: List<Spore>

A gombatest spóráit tartalmazó lista. A lista alapértelmezetten üres.

#### Metódusok

#### +MushroomBody(location: FertileTecton, name: String)

Konstruktor, amely beállítja a létrehozandó gombatest nevét és azt a tektont (céltekton), amelyen az elhelyezésre kerül. Ez a konstruktor használandó FertileTecton, továbbá a FertileTecton valamennyi leszármazottja, azaz AridTecton, MultiLayeredTecton és SustainingTecton esetén. A metódus pszeudokódja:

procedure MushroomBody(location: FertileTecton, name: String)

// elmenti a helyszínt

SET this.location ← location

// létrehoz egy MushroomBodyGrowthEvaluator példányt

evaluator ← CREATE MushroomBodyGrowthEvaluator(this)

// a feltételek fennállásának kiértékelése céljából meghívja a céltektonra

// a visit metódust

CALL evaluator.visit(location, this)

END

#### +MushroomBody(location: SemiFertileTecton, name: String)

Konstruktor, amely beállítja a létrehozandó gombatest nevét és azt a tektont (céltekton), amelyen az elhelyezésre kerül. Ez a konstruktor használandó SemiFertileTecton esetén. A metódus pszeudokódja:

procedure MushroomBody(location: SemiFertileTecton, name: String)

// elmenti a helyszínt

SET this.location ← location

// létrehoz egy MushroomBodyGrowthEvaluator példányt

evaluator ← CREATE MushroomBodyGrowthEvaluator(this)

// a feltételek fennállásának kiértékelése céljából meghívja a céltektonra

// a visit metódust

CALL evaluator.visit(location, this)

END

#### +MushroomBody()

Paraméter nélküli (default) konstruktor.

#### +delete(): void

#### A növekedési feltételek hiánya esetében kerül meghívásra az előzetesen létrehozott gombatest törlése céljából.

#### +grow(sporeCount: int): void

#### A gombatest növekedési folyamatát lezáró metódus, amelyet a Mushroom interfész miatt szükséges a gombatestnél ilyen formában megvalósítani. A paramétert a céltektontól kapja, és csak a gombafonál esetében bír jelentőséggel. A tekton abban az esetben hívja meg ezt a metódust (és nem a delete()-et), ha a gombatest növesztési feltételeire vonatkozó vizsgálat pozitív eredményt hozott.

#### +onTurnBegin(): void

A gombatest minden új körének kezdetekor – beleértve a játék első körét is – a gombatestben egy új spóra termelődik. A spóra típusa véletlenszerűen kerül kiválasztásra. A metódus pszeudokódja:

procedure onTurnBegin()

// Egy spóratípust véletlenszerűen kiválasztásra kerül

random ← RANDOM NUMBER BETWEEN 1 AND 5

IF random == 1 THEN

newSpore ← CREATE SpitSpore()

ELSE IF random == 2 THEN

newSpore ← CREATE StunSpore()

ELSE IF random == 3 THEN

newSpore ← CREATE PreventCutSpore()

ELSE IF random == 4 THEN

newSpore ← CREATE SpeedSpore()

ELSE

newSpore ← CREATE SlownessSpore()

END IF

// Hozzáadja az új spórát a gombatest spóralistájához

CALL this.addSpore(newSpore)

END

#### +getRemainingEjects(): int

Visszaadja a gombatest megmaradt spórakilövéseinek számát.

#### +setRemainingEjects(remainingEjects: int): void

Beállítja a gombatest megmaradt spórakilövéseinek számát.

#### +getSpores(): List<Spore>

Visszaadja a gombatest spóráit tartalmazó listát.

#### +addSpore(newSpore: Spore): void

Hozzáad egy új spórát a gombatest spóráinak listájához.

#### +ejectSpores(target: Tecton): void

A gombatest spóráinak kilövéséért felelős metódus. A metódus pszeudokódja:

procedure ejectSpores(target: Tecton)

// Ha már volt 3 spórakilövése, a gombatest inaktív, nem tud aktivitást

// kifejteni így spórát sem lőhet ki spórát (nincs is már neki)

IF remainingEjects == 0 THEN

RETURN // A gombatest inaktív, nem tud aktivitást kifejteni

END IF

// Ha ez az utolsó, azaz a 3. spórakilövése, a gombatest fejlett állapotú,

// így a céltekton lehet szomszéd vagy a szomszéd szomszédja

IF remainingEjects == 1 THEN

reachable ← EMPTY SET

FOR EACH primary IN this.neighbours DO

ADD primary TO reachable

FOR EACH secondary IN primary.neighbours DO

ADD secondary TO reachable

END FOR

END FOR

IF target IS IN reachable THEN

ejectSpores(target)

remainingEjects ← remainingEjects - 1

ELSE

RETURN // A céltekton túl messze van!

END IF

ELSE

// A gombatest még nem fejlett, ezért csak közvetlen szomszédjára lőhet

// spórát

IF target IS IN this.neighbours THEN

ejectSpores(target)

remainingEjects ← remainingEjects - 1

ELSE

RETURN // A céltekton túl messze van!

END IF

END IF

END

#### +getLocation(): Tecton

Visszaadja a gombatest elhelyezkedése szerinti tektont.

#### +setLocation(location: Tecton): void

Beállítja a gombatest elhelyezkedése szerinti tektont.