

Általánosan

- vmitől tud öregedni a gombatest és így messzebbre is tud lőni - ezt még belevenni
- a teljesen terméketlen tekton a konzulenszt nem volt a feladatleírásban, de ez nem hiba
- most még csak tervezünk és nem modellezünk

O.diagram

- a Tectonban minek van benne a mushroomBody, ha ott az assz. is a M.Bodyval? Legyen inkább csak az assz., és adjunk neki nevet.
- rendezett reláció legyen a Tecton-Mycelium kapcs. számossággal
- az Insect néhány cucca, pl. a canCit fölös, ha pl. a Spore azt mondha a rovarnak h beStun().
- visitor mintával is gondok vannak. Nem feltétlenül kell ez. Van egy Mycelium, mondja h nőjön.

Sz.diagramok

- azt kéne végiggondolni h milyen szekvenciad. hiányzik még. Ilyen pl. a fonalelhalás tektontöréskor és a rovar evésénél. Pl. tektontörés esetén ezt a kérdést kezelni kell magán a diagramon, végigpörgetve az összes érintett tektonon.
- Hiányzik még a rovarmozgáa diagramja is a mozgással megváltozó paraméterekkel, hívásokkal. A tektonon is be kell állítani h melyik rovar van rajta.

A. sz.diagram

- Ő csinálna egy állapotgépet a tektonra is, még ha csak két állapota is van a tektonnak.
- Csökken a tekton számlálója, meg van egy guard => meghív egy akciót és eltörik. Ehhez a szekv.diagramhoz meg odaírhatjuk h a törés pillanataól indulunk. A mi diagramunkból hiányzik az új tekton és az h mi lesz a tektonon lévő fonállal, gombatesttel.

B. sz.diagram

- nem érti, miért az utolsót vágjuk el.
- a gombatestek össze vannak nőve az egyes játékosok között? Ezt beszéltük meg?
- megszűnik az összeköttetés - ezt nem értettem
- a gombafonál megszűnik - vmi x-et írjunk meg h ez látszódjon

C. sz.diagram

- gyakorlatilag egy dim. típuslekérdezést csinálunk
- a spóra hívjon meg vmit a rovaron, ne pedig getStrategy() legyen a Tell, don't ask jegyében
- switch case-ekkel is meg lehet csinálni
- konz.: nem lenne ehyszerűbb ez?: a rovar jelzi h megette a spórát, a spóra meg jelzi h pl. nem vághat, nem mozoghat, a remaininget is beállítja. A rovarnak nincs tennivalója így ezután. Jelenleg ui. a rovarnak minden kör elején kell vmit csinálnia h megtudja, a spóra

csinált-e vele valamit. Ez azonban nem a rovar dolga.

D. sz.diagram

- erre nincs is szükség, ha a spóra állítja be a rovar

E. sz.diagram

- a tekton spóralistáját nem látja megváltozni. Ezt referenciaként gondoltuk. Miért nem küldünk át egy spórát a tektonnak azzal h adja hozzá, ui. amit csináltunk, nem obj.orientált, ui. más bizgerálja a lz ő attribútumát

- a véletlengenerátor legyen egy külön aktor, és adja vissza mindig uazt a számot, különben nehezen lesz tesztelhető

Állapotgépek

A. á.gép

- nem lászik, ki hívja meg a makeSpore-t, ez

szte legyen az onTurnBegin.

- ejected-et növelné, az elején legyen 0.

Legyen guard is ezzel kapcsolatban.

- ha más miatt nem lehet inaktív, nem kell hierarchikus á.gép.

B. á.gép

- nem az eatSpore miatt következik be az á.változás, hanem mert a Spore hív meg vmit. Ő számlálót is beletenne, így nem kellene loop. Guard is kellene.

- Specifikálni kellene h mi lesz, ha pl. egy gyorsult állapotból pl. nem vág állapotba kerülök. Azonnal?

- van vhol nekünk guard, de ott nincs trigger. A guard csak a trigger hatására értékelődik ki. Az onTurnBegin lehet ilyen trigger.

- van állapota a rovarnak, a gombatestnek, ha fiatal, öreg, azt is kell kezelni. A tektontörés nem olyan, amit itt ábrázolni kellene.