# Prototípus koncepciója

[A prototípus program célja annak demonstrálása, hogy a program elkészült, helyesen működik, valamennyi feladatát teljesíti. A prototípus változat egy elkészült program kivéve a kifejlett grafikus interfészt. Ez a program is parancssorból futtatható és karakteres ernyőkezelést alkalmaz. Az ütemezés, az aktív objektumok kezelése megoldott. A business objektumok - a megjelenítésre vonatkozó részeket kivéve - valamennyi metódusa a végleges algoritmusokat kell, hogy tartalmazza. A megjelenítés és működtetés egy alfanumerikus képernyőn vezérelhető és követhető, ugyanakkor a vezérlés fájlból is történhet és a megjelenítés fájlba is logolható, ezzel megteremtve a rendszer tesztelésének lehetőségét. Különös figyelmet kell fordítani a parancssori interfész logikájára, felépítésére, valamint arra, hogy az mennyiben tükrözi és teszi láthatóvá a program működését, a beavatkozások hatásait.]

## Változás hatása a modellre

### Módosult osztálydiagram

*[Az analízis modell osztálydiagramja a változások figyelembevételével.]*

### Új vagy megváltozó metódusok

*[Az analízis modell osztályleírásaiból azon metódusok újbóli felsorolása leírással együtt, amelyek a változtatás miatt módosultak vagy újonnan be lettek vezetve.]*

### Szekvencia-diagramok

*[Az analízis modell szekvenciadiagramjaiból a változás által érintett, előírt, módosított diagramok.]*

## Prototípus interface-definíciója

[Definiálni kell a teszteket leíró nyelvet. Külön figyelmet kell fordítani arra, hogy ha a rendszer véletlen elemeket is tartalmaz, akkor a véletlenszerűség ki-bekapcsolható legyen, és a program determinisztikusan is tesztelhető legyen.]

### Az interfész általános leírása

[A protó (karakteres) input és output felületeit úgy kell kialakítani, hogy az input fájlból is vehető legyen illetőleg az output fájlba menthető legyen, vagyis kommunikációra csak a szabványos be- és kimenet használható.]

### Bemeneti nyelv

[Definiálni kell a teszteket leíró nyelvet (szintakszis és szemantika). Külön figyelmet kell fordítani arra, hogy ha a rendszer véletlen elemeket is tartalmaz, akkor a véletlenszerűség ki-bekapcsolható legyen, és a program determinisztikusan is futtatható legyen. A szálkezelést is tesztelhető, irányítható módon kell megoldani. A programot egy adott konfigurációból is el kell tudni indítani, vagyis kell olyan parancs, amivel konkrét előre megadott állapotból indul a rendszer (pl. load).]

**Parancs1**

**Leírás:**

**Opciók:**

**Parancs2**

**Leírás:**

**Opciók:**

[Ha szükséges, meg kell adni a konfigurációs (pl. pályaképet megadó) fájlok nyelvtanát is.]

### Kimeneti nyelv

[Egyértelműen definiálni kell, hogy az egyes bemeneti parancsok végrehajtása után előálló állapot milyen formában jelenik meg a szabványos kimeneten. A program képes legyen olyan kimenetet előállítani, amellyel az objektumok állapota ellenőrizhető (pl. save). Ebben az alfejezetben is precízen definiálni kell, hogy a kimenet nyelve milyen elemekből és milyen szintakszissal áll elő.]

## Összes részletes use-case

[A use-case-eknek a részletezettsége feleljen meg a kezelői felületnek, azaz a felület elemeire kell hivatkozniuk a bemeneti nyelv parancsai alapján.

Alábbi táblázat minden use-case-hez külön-külön.]

|  |  |
| --- | --- |
| **Use-case neve** | Gombafonál létrehozása |
| **Rövid leírás** | Azonnal létrehoz egy gombafonalat, ami még nem tartozik tektonhoz. |
| **Aktorok** | User/GameManager |
| **Forgatókönyv** | 1. Létrejön egy gombafonál |

|  |  |
| --- | --- |
| **Use-case neve** | Gombafonál hozzáadása tektonhoz |
| **Rövid leírás** | Hozzáadja a kiválasztott fonalat a kiválasztott tektonhoz |
| **Aktorok** | User/GameManager |
| **Forgatókönyv** | 1. A játékos kiválaszt egy gombafonalat 2. A játékos kiválaszt egy tektont 3. A kiválasztott fonal (,ami még nincs tektonon) azonnal rajta lesz a kiválasztott tektonon |

|  |  |
| --- | --- |
| **Use-case neve** | Gombafonál növesztése |
| **Rövid leírás** | Gombafonál növesztés a kiválasztott tektonra. |
| **Aktorok** | Gombász |
| **Forgatókönyv** | 1. A játékos kiválaszt egy tektont, ahol létrejön 2. Létrejön egy gombafonál ideiglenesen 3. A gombafonál megvizsgáltatja a kiválasztott tektonnal, hogy képes-e nőni rá   3.a) Ha képes a fonál nőni a tektonra akkor véglegesen létrejön a tektonon  3.b) Ha a fonál nem képes nőni a tektonra, akkor elpusztul |

|  |  |
| --- | --- |
| **Use-case neve** | Gombafonál elvágása |
| **Rövid leírás** | Lehetővé teszi, egy gombafonál elvágását (rovar és tektontörés nélkül). |
| **Aktorok** | Gombász |
| **Forgatókönyv** | 1. A játékos kiválaszt egy gombafonalat 2. Elvágódik a kiválasztott gombafonál és elpusztul 3. A vágás hatására ki fognak törlődni a játéktéren lévő olyan fonalak is, akik már nincsenek gombatesthez kötve a vágás hatására 4. runAway? |

|  |  |
| --- | --- |
| **Use-case neve** | Gombafonál törlése |
| **Rövid leírás** | Töröl egy kiválasztott fonalat, ami nincs tektonon |
| **Aktorok** | User/GameManager |
| **Forgatókönyv** | 1. A játékos kiválaszt egy gombafonalat 2. A kiválasztott gombafonál elpusztul |

## Tesztelési terv

[A tesztelési tervben definiálni kell, hogy a be- és kimeneti fájlok egybevetésével miként végezhető el a program tesztelése. Meg kell adni magas szintű teszt forgatókönyveket. Az egyes teszteket elég informálisan, szabad szövegként leírni, tesztesetenként egy-öt mondatban. Minden teszthez meg kell adni, hogy mi a célja, a proto mely funkcionalitását, osztályait stb. teszteli. Az alábbi táblázat minden teszt-esethez külön-külön elkészítendő.]

|  |  |
| --- | --- |
| **Teszt-eset neve** | Gombafonál (lassú) növesztése |
| **Rövid leírás** | A gombafonál rá nő a tesztelő által kiválasztott tektonra ha az még nincs „tele” fonállal (kapacitása és rajta lévő fonalak száma egyenlő) és közvetlen szomszédja olyan másik tektonnak, amin van gombatest vagy gombafonál. |
| **Teszt célja** | A kiválasztott tektonon megjelenik egy új gombafonál. 2 kör után. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Gombafonál sikertelen növesztése |
| **Rövid leírás** | A gombafonál nem nő rá a tesztelő által kiválasztott tektonra, mert az nem szomszédja olyan másik tektonnak, amin van gombatest vagy gombafonál. Továbbá akkor sem megy végbe a növesztés, ha a kiválasztott tekton már „tele” van fonállal (kapacitása és rajta lévő fonalak száma egyenlő). |
| **Teszt célja** | Az ideiglenes gombafonál elpusztul. Nem lesz változás a játéktéren a parancs kiadása előtti állapothoz képest. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Gombafonál gyors növesztése |
| **Rövid leírás** | A gombafonál rá nő a tesztelő által kiválasztott tektonra ha az még nincs „tele” fonállal (kapacitása és rajta lévő fonalak száma egyenlő) és közvetlen szomszédja olyan másik tektonnak, amin van gombatest vagy gombafonál. A kiválasztott tektonon van spóra ezért a növést gyorsabban teszi meg. |
| **Teszt célja** | A kiválasztott tektonon megjelenik egy új gombafonál. egy kör után. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Húsevő fonál működése |
| **Rövid leírás** | Új kör kezdetekor a tektonon lévő húsevő fonál megeszi a rajta lévő bénult állapotban lévő rovarokat és gombatestet növeszt. |
| **Teszt célja** | A tektonon, ahol a húsevő fonál van, meghalnak a rovarok és nő egy új gombatest (ha tud oda nőni). |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Gombafonál elvágása |
| **Rövid leírás** | A játékos kiválaszt egy gombafonalat.Elvágódik a kiválasztott gombafonál és elpusztul. A vágás hatására ki fognak törlődni a játéktéren lévő olyan fonalak is, akik már nincsenek gombatesthez kötve a vágás hatására.  runAway? |
| **Teszt célja** | A tektonon, ahol a kiválasztott fonál van elpusztul a fonál és a játéktéren |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Gombafonál elhalása AridTectonon |
| **Rövid leírás** | Új kör kezdetekor az AridTectonon lévő fonál, ha már 5 köre van ott, felszívódik, azaz elvágódik. |
| **Teszt célja** | A vizsgált AridTectonon lévő fonál *elvágódik*. |

## Tesztelést támogató segéd- és fordítóprogramok specifikálása

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

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2025.03.27 17:00 | 1 óra | Guzmics |  |
| 2025.03.28 16:00 | 1 óra | Guzmics |  |
| 2025.03.28 21:00 | 1 óra | Guzmics |  |

Jegyzetek:

nem írok feltételeket a use-case-ekbe

nem írok objektumneveket kivéve Arid esetben

Kérdések:

nem lesz baj, ha nálam absztrak test-case-ek vannak mert az ellene megy a fájlból ellenőrizhetőségnek?

Ha jól értem teszteseteket csak úgy tudjuk, fájlokkal tesztelni, hogy egy alapállapotból bizonyos parancs kiadása utáni állapotot hasonlítunk össze az elvárt állapottal. Ebből viszont következik, hogy minden tesztesethez kell (nem feltétlen külön) alapállapot. Ezt kell leírni vagy nem?

a teszt cél implicit következik az enyémből?

runAway-t lehet tesztelni?

gombatest fejletté válás test-case?