**3. Analízis modell kidolgozása**

15 – jgoldfisch

Konzulens:

Bodó Zsófia

Csapattagok

|  |  |  |
| --- | --- | --- |
| Mátyás Gergely | IL21NI | matyasg97@gmail.com |
| Horváth Ákos | DKILK6 | horvath.akos1997@gmail.com |
| Gurubi Barnabás | DXEXVR | gurubibarni@gmail.com |
| Tolnai Márk | ID61MK | tolesz11@windowslive.com |
| Bertalan Bálint | HNN9GA | blintber@gmail.com |

2018. február 22.

# Analízis modell kidolgozása

## Objektum katalógus

[Minden, a feladatban szereplő objektum rövid, egy-két bekezdés hosszú ismertetése. Meg kell jelenjen minden objektumhoz, hogy mi a felelőssége. Informális leírás, ezért nem szabad még foglalkozni az örökléssel, az interfészekkel, az absztrakt osztályokkal, a segédosztályokkal.]

### Field

A pálya egyes mezőit képviselik a játékban. Lekérdezhető tőle a megfelelő irányban lévő szomszédos **Field**. Egy cellán egyidőben állhat egy **Box,** vagy egy **Player** állhat, valamint tartalmazhat referenciát egy speciális elemre amely lehet **Hole**, **Switch,** vagy **Wall**.

### Box

A box osztály minden példánya egy-egy különböző dobozt tárol. Amennyiben egy dobozt a célmezőre tolnak ellehetetleníti önmaga mozgatását.

### Game

A játék alapvető logikáját és mechanizmusait tárolja. Ezen felül referenciát tartalmaz a **Mapre** és tárolja a **Worker-öket**.

### Hole

Egy lyukat jelenít meg. Amennyiben egy **Box**, vagy **Worker** áll rajta, megsemmisíti a rajta álló objektumot.

### Map

A pályát képviseli, mely több **Fieldből** áll. A menüben a pálya kiválasztását követően az osztály képes betölteni azt egy külső fájlból.

### Switch

Egy a pályán található kapcsolót valósít meg. Minden kapcsolóhoz tartozik egy referencia egy **Hole** osztályra. Amennyiben egy **Box** áll a **Switchen**, akkor aktiválódik, más esetben nem.

### Wall

Egy oszlopot, vagy falszeletet ábrázol a pályán. Megakadályozza, hogy dobozt toljanak rá, vagy hogy rálépjen egy játékos.

### Worker

Egy játékost képvisel, aki pálya területén belül szabadon mozoghat. Segítségével lehetséges a ládák tologatása és tárolja, hogy hány ládát tolt már a helyére.

[Balu – 1,5 óra, Márk – 0,5 óra]

[Az objektumkatalógus alapján kiindulva kell megalkotni az objektumorientált analízis modellt. A 3.2, 3.3, 3.4, 3.5 alfejezetek ugyanannak a modellnek a különböző nézetei, ezért egyidőben, egymással összefüggésben készülnek. Megtörténik az objektumkatalógusban tárgyalt objektumok felelősségének formalizálása osztályokká, attribútumokká, metódusokká . Csak publikus metódusok szerepelhetnek. Megjelennek az interfészek, az öröklés, az absztrakt osztályok. Segédosztályokra még nincs szükség.]

## Statikus struktúra diagramok

[Az objektum katalógusban szereplő objektumokat megvalósító osztályok kapcsolatait és publikus metódusait bemutató osztálydiagram(ok). Tipikus hibalehetőségek: csillag-topológia, szigetek.]

## Osztályok leírása

[A diagramon szereplő osztályok bemutatása. Az osztályok ABC sorrendben kövessék egymást. Interfészek esetén az Interfészek, Attribútumok pontok kimaradnak.]

### Osztály1

#### Felelősség

[Mi az osztály felelőssége. Kb 1 bekezdés.]

#### Ősosztályok

[Mely osztályokból származik (öröklési hierarchia)]

Legősebb osztály → Ősosztály2 → Ősosztály3...

#### Interfészek

[Mely interfészeket valósítja meg.]

#### Attribútumok

[Milyen attribútumai vannak]

* **attribútum1**: attribútum jellemzése: mire való
* **attribútum2**: attribútum jellemzése: mire való

#### Metódusok

[Milyen publikus metódusokkal rendelkezik. Metódusonként egy-három mondat arról, hogy a metódus mit csinál.]

* **int foo(Osztály3 o1, Osztály4 o2)**: metódus leírása
* **int bar(Osztály5 o1)**: metódus leírása

### Osztály2

#### Felelősség

[Mi az osztály felelőssége. Kb 1 bekezdés.]

#### Ősosztályok

[Mely osztályokból származik (öröklési hierarchia)]

Legősebb osztály → Ősosztály2 → Ősosztály3...

#### Interfészek

[Mely interfészeket valósítja meg.]

#### Attribútumok

[Milyen attribútumai vannak]

* **attribútum1**
* **attribútum2**

#### Metódusok

[Milyen publikus metódusokkal rendelkezik. Metódusonként egy-három mondat arról, hogy a metódus mit csinál. Hiba, ha egy osztálynak nincs egyetlen metódusa sem.]

* **int foo(Osztály3 o1, Osztály4 o2)**: metódus leírása
* **int bar(Osztály5 o1)**: metódus leírása

## Szekvencia diagramok

[Inicializálásra, use-case-ekre, belső működésre. Konzisztens kell legyen az előző alfejezettel. Minden metódus, ami ott szerepel, fel kell tűnjön valamelyik szekvenciában. Minden metódusnak, ami szekvenciában szerepel, szereplnie kell a valamelyik osztálydiagramon. Fontos, hogy az aktor use-case-eitől bármelyik diagramhoz el lehessen jutni a metódushívások követésével.]

## State-chartok

[Csak azokhoz az osztályokhoz, ahol van értelme. Egyetlen állapotból álló state-chartok ne szerepeljenek. A játék működését bemutató state-chart-ot készíteni tilos.]

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2010.03.21. 18:00 | 2,5 óra | Horváth  Németh  Tóth  Oláh | Értekezlet.  Döntés: Horváth elkészíti az osztálydiagramot, Oláh a use-case leírásokat. |
| 2010.03.23. 23:00 | 5 óra | Németh | Tevékenység: Németh implementálja a tesztelő programokat. |
| … | … | … | … |