

2. Követelmény, projekt, funkcionalitás

66 – [\[simon_balazst_szeretnenk_konzulensnek\]](#)

Konzulens:

Simon Balázs

Csapattagok:

Kiss Andor	TXC54G	kissandor4@gmail.com
Konrád Márk	JSPDME	konrad0816@gmail.com
Glávits Balázs Róbert	NMZC9G	glavits.balazs@gmail.com
Máté Botond	ELOYOV	m.botond7@gmail.com
Lant Gábor	P35E36	lant.gabor98@gmail.com

2020. február 22.

Tartalomjegyzék

2	Követelmény, projekt, funkcionalitás	6
2.1	Bevezetés	6
2.1.1	Cél	6
2.1.2	Szakterület	6
2.1.3	Definíciók, rövidítések	6
2.1.4	Hivatkozások	6
2.1.5	Összefoglalás	6
2.2	Áttekintés	6
2.2.1	Általános áttekintés	6
2.2.2	Funkciók	6
2.2.3	Felhasználók	7
2.2.4	Korlátozások	7
2.2.5	Feltételezések, kapcsolatok	7
2.3	Követelmények	7
2.3.1	Funkcionális követelmények	7
2.3.2	Erőforrásokkal kapcsolatos követelmények	7
2.3.3	Átadással kapcsolatos követelmények	7
2.3.4	Egyéb nem funkcionális követelmények	8
2.4	Lényeges use-case-ek	8
2.4.1	Use-case leírások	8
2.5	Szótár	9
2.6	Projekt terv	9
2.7	Napló	9
3	Analízis modell kidolgozása 1	10
3.1	Objektum katalógus	10
3.1.1	Objektum1	10
3.1.2	Objektum2	10
3.2	Statikus struktúra diagramok	10
3.3	Osztályok leírása	10
3.3.1	Osztály1	10
3.3.2	Osztály2	11
3.4	Statikus struktúra diagramok	11
3.5	Szekvencia diagramok	11
3.6	State-chartok	12
3.7	Napló	12
4	Analízis modell kidolgozása 2	13
4.1	Objektum katalógus	13
4.1.1	Objektum1	13
4.1.2	Objektum2	13
4.2	Statikus struktúra diagramok	13
4.3	Osztályok leírása	13
4.3.1	Osztály1	13
4.3.2	Osztály2	14
4.4	Szekvencia diagramok	14
4.5	State-chartok	14

4.6	Napló	14
5	Szkeleton tervezése	16
5.1	A skeleton modell valóságos use-case-ei	16
5.1.1	Use-case diagram	16
5.1.2	Use-case leírások	16
5.2	A skeleton kezelői felületének terve, dialógusok	16
5.3	Szekvencia diagramok a belső működésre	16
5.4	Kommunikációs diagramok	16
5.5	Napló	16
6	Szkeleton beadás	18
6.1	Fordítási és futtatási útmutató	18
6.1.1	Fájllista	18
6.1.2	Fordítás	18
6.1.3	Futtatás	18
6.2	Értékelés	18
6.3	Napló	18
7	Prototípus koncepciója	20
7.1	Prototípus interface-definíciója	20
7.1.1	Az interfész általános leírása	20
7.1.2	Bemeneti nyelv	20
7.1.3	Kimeneti nyelv	20
7.2	Összes részletes use-case	20
7.3	Tesztelési terv	21
7.4	Tesztelést támogató segéd- és fordítóprogramok specifikálása	21
7.5	Napló	21
8	Részletes tervek	22
8.1	Osztályok és metódusok tervei	22
8.1.1	Osztály1	22
8.1.2	Osztály2	22
8.2	A tesztek részletes tervei, leírásuk a teszt nyelven	23
8.2.1	Teszteset1	23
8.2.2	Teszteset2	23
8.3	A tesztelést támogató programok tervei	23
8.4	Napló	23
10	Prototípus beadása	24
10.1	Fordítási és futtatási útmutató	24
10.1.1	Fájllista	24
10.1.2	Fordítás	24
10.1.3	Futtatás	24
10.2	Tesztek jegyzőkönyvei	24
10.2.1	Teszteset1	24
10.3	Értékelés	25
10.4	Napló	25
11	Grafikus felület specifikációja	26
11.1	A grafikus interfész	26

11.2	A grafikus rendszer architektúrája	26
11.2.1	A felület működési elve	26
11.2.2	A felület osztály-struktúrája	26
11.3	A grafikus objektumok felsorolása	26
11.3.1	Osztály1	26
11.3.2	Osztály2	27
11.4	Kapcsolat az alkalmazói rendszerrel	27
11.5	Napló	27
13	Grafikus felület specifikációja	28
13.1	Fordítási és futtatási útmutató	28
13.1.1	Fájllista	28
13.1.2	Fordítás	28
13.1.3	Futtatás	28
13.2	Értékelés	28
13.3	Napló	28
14	Összefoglalás	30
14.1	Projekt összegzés	30

Ábrák jegyzéke

2.1	Use-Case diagram	8
3.1	x	10
3.2	x	11
3.3	x	11
3.4	x	12
4.1	x	13
4.2	x	14
4.3	x	14
5.1	x	16
7.1	x	20
11.1	x	26

2. Követelmény, projekt, funkcionalitás

2.1. Bevezetés

2.1.1. Cél

Ez a dokumentum egy szoftverfejlesztési projekt információit tartalmazza az ötlettől a kész termékig, minden lépést naplózva.

2.1.2. Szakterület

A feladat játékprogram készítése, melyben a játékosok legalább háromfős csapatban működnek együtt. A program személyi számítógépeken, grafikus módban fog futni. A játék offline, tehát a több játékos egy számítógépen játsza.

2.1.3. Definíciók, rövidítések

Még nincs.

2.1.4. Hivatkozások

Még nincs.

2.1.5. Összefoglalás

[\[A dokumentum további részeinek rövid ismertetése\]](#)

2.2. Áttekintés

2.2.1. Általános áttekintés

A szoftver három fő komponense a Model, a View, és a Controller. A Model reprezentálja a játék állapotát. A View kapcsolódik a Modelhez, és megjeleníti azt. A Controller felelős a felhasználói bemenetek kezeléséért és a Model frissítéséért.

2.2.2. Funkciók

A játékban a különböző képességű szereplőknek (3 vagy több játékos lehet) kell egy tengerrel körülvett jégmezőn túlélniük. A szereplők lehetnek eszkimók vagy sarkkutatók, és körökre osztva tevékenykednek.

A jégmező jégtablából áll. Vannak stabil jégtablák, amelyeken akárhány szereplő állhat, és vannak instabil jégtablák, amik adott létszám felett átfordulnak és ilyenkor a rajtuk állók a vízbe esnek. A jégtablákat a játék kezdetén eltérő mennyiségű hó borítja.

Az egyes jégtablákba különféle tárgyak lehetnek belefagyva: lapát, kötél, bűvárruha, élelem, stb. Befagyott tárgyat csak akkor lehet meglátni és kiásni, ha a jégtacla tiszta, nem borítja hó. A jégtablák között lehetnek hóval fedett lukak is, amibe beleesve csak a bűvárruhát viselők élnek túl, vagy azok, akiket egy köteles barátjuk a szomszéd jégtabláról azonnal kimenekít.

Minden szereplő egy körben 4 egységnyi munkát végezhet. Ilyen munka például a jégtablán levő egységnyi mennyiségű hó eltakarítása, egy szomszédos jégtablára való lépés vagy egy kiásott tárgy felvétele. A lapáttal két egységnyi hó takarítható el egy munkaráfordítással.

A jégmezőn időnként feltámad a hóvihar, és néhány érintett jégtablát újabb adag friss hóval borít be. Akit elkap, annak a testhője egységnivel csökken. Az eszkimóknak a játék elején 5 egység testhője van, a sarkkutatónak csak 4. Egy élelem eggyel növeli a testhőt.

A szereplők jégtabláról-jégtablára haladnak képességeiknek megfelelően. A sarkkutató meg tudja nézni, hogy az a jégtabla, amire lépne, hány embert bír el (a luk egyet sem). Az eszkimó tud iglut építeni, amiben átvészeltetők a hóviharok. Egy-egy képesség alkalmazása is egy-egy munkát jelent.

A játék célja egy jelzőrakéta alkatrészeinek (pisztoly, jelzőfény, patron) megtalálása. Az alkatrészek is a jégbe vannak fagyva. Ha ezeket a csapat összegyűjti és ugyanarra a jégtablára viszi, akkor egy munka felhasználásával összeszerelhetik és elsüthetik, amivel megnyerik a játékot. Ehhez azonban mindannyiuknak ugyanott kell állniuk. Ha valaki menet közben meghal (vízbe esve megfullad vagy elfogy a testhője és kihűl), akkor a játék véget ér.

2.2.3. Felhasználók

[\[A felhasználók jellemzői, tulajdonságai\]](#)

2.2.4. Korlátozások

[\[Az elkészítendő szoftverre vonatkozó – általában nem funkcionális - előírások, korlátozások.\]](#)

2.2.5. Feltételezések, kapcsolatok

Még nincs.

2.3. Követelmények

2.3.1. Funkcionális követelmények

[\[Az alábbi táblázat kitöltésével készíthető. Dolgozzon ki követelmény azonosító rendszert! Az ellenőrzés módja szokásosan bemutatás és/vagy kiértékelés. Prioritás lehet alapvető, fontos, opcionális. Az alapvető követelmények nem teljesítése végzetes. Forrás alatt a követelményt előíró anyagot, szervezetet kell érteni. Esetünkben forrás lehet maga a csapat is, mikor ő talál ki követelményt. Use-case-ek alatt az adott követelményt megvalósító használati esete\(ke\)t kell megadni.\]](#)

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
...

2.3.2. Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
RES_REQ_01	Java futtatókörnyezet.	Az operációs rendszer csomagkezelőjében.	Elengedhetetlen.	–	Itt elérhető.
RES_REQ_02	Java SDK.	Az operációs rendszer csomagkezelőjében.	A fordításhoz elengedhetetlen.	–	Itt elérhető.

2.3.3. Átadással kapcsolatos követelmények

[\[A szoftver átadásával, telepítésével, üzembe helyezésével kapcsolatos követelmények\]](#)

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
-----------	--------	------------	-----------	--------	---------

...
-----	-----	-----	-----	-----	-----

2.3.4. Egyéb nem funkcionális követelmények

[A biztonsággal, hordozhatósággal, megbízhatósággal, tesztelhetőséggel, a felhasználóval kapcsolatos követelmények]

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
...

2.4. Lényeges use-case-ek

2.1. ábra. Use-Case diagram

2.4.1. Use-case leírások

Use-case neve	Step
Rövid leírás	A játékos lép.
Aktorok	Player
Forgatókönyv	...

Use-case neve	Dig
Rövid leírás	A játékos havat lapátol.
Aktorok	Player
Forgatókönyv	...

Use-case neve	Pick item up
Rövid leírás	A játékos felvesz egy tárgyat.
Aktorok	Player
Forgatókönyv	...

Use-case neve	Make rocket
Rövid leírás	A játékos jelzőrakétát készít.
Aktorok	Player
Forgatókönyv	...

Use-case neve	Save teammate
Rövid leírás	A játékos kiment egy vízbe esett csapattársat.
Aktorok	Player
Forgatókönyv	...

Use-case neve	Build igloo
Rövid leírás	Az eszkimó iglut épít.
Aktorok	Eskimo
Forgatókönyv	...

Use-case neve	Examine tile
Rövid leírás	A sarkkutató feltérképez egy jégtáblát.
Aktorok	PolarExplorer
Forgatókönyv	...

Use-case neve	Turn unstable ice
Rövid leírás	Instabil jég megfordul.
Aktorok	Controller
Forgatókönyv	...

Use-case neve	Create snowstorm
Rövid leírás	Hóvihar kezdődik.
Aktorok	Controller
Forgatókönyv	...

2.5. Szótár

[A szótár a követelmények alapján készítendő fejezet. Egy szótári bejegyzés definiálásához csak más szótári bejegyzések és köznap – a feladattól független – fogalmak használhatók fel. A szótár mérete kb. 1-2 oldal legyen.]

2.6. Projekt terv

[Tartalmaznia kell a projekt végrehajtásának lépéseit, a lépések, eredmények határidejét, az egyes feladatok elvégzéséért felelős személyek nevét és beosztását, a szükséges erőforrásokat, stb. Meg kell adni a csoportmunkát támogató eszközöket, a választott technikákat! Definiálni kell, hogy hogyan történik a dokumentumok és a forráskód megosztása!]

2.7. 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.
...

3. Analízis modell kidolgozása 1

3.1. 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 kell foglalkozni az örökléssel, az interfészekkel, az absztrakt osztályokkal, a segédosztályokkal.]

3.1.1. Objektum1

[Felelősség informális leírása]

3.1.2. Objektum2

[Felelősség informális leírása]

3.2. Statikus struktúra diagramok

[Az előző alfejezet osztályainak kapcsolatait és publikus metódusait bemutató osztálydiagram(ok). Tipikus hibalehetőségek: csillag-topológia, szigetek.]

3.1. ábra. x

3.3. Osztályok leírása

[Az előző alfejezetben tárgyalt objektumok felelősségének formalizálása attribútumokká, metódusokká. Csak publikus metódusok szerepelhetnek. Ebben az alfejezetben megjelennek az interfészek, az öröklés, az absztrakt osztályok. Segédosztályokra még mindig nincs szükség. Az osztályok ABC sorrendben kövessék egymást. Interfészek esetén az Interfészek, Attribútumok pontok kimaradnak.]

3.3.1. 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

3.3.2. 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: 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

3.4. Statikus struktúra diagramok

[Az előző alfejezet osztályainak kapcsolatait és publikus metódusait bemutató osztálydiagram(ok). Tipikus hibalehetőségek: csillag-topológia, szigetek.]

3.2. ábra. x

3.5. 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.]

3.3. ábra. x

3.6. State-chartok

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

3.4. ábra. x

3.7. Napló

Kezdet	Időtartam	Résztevők	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.
...

4. Analízis modell kidolgozása 2

4.1. 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 kell foglalkozni az örökléssel, az interfészekkel, az absztrakt osztályokkal, a segédosztályokkal.]

4.1.1. Objektum1

[Felelősség informális leírása]

4.1.2. Objektum2

[Felelősség informális leírása]

4.2. Statikus struktúra diagramok

[Az előző alfejezet osztályainak kapcsolatait és publikus metódusait bemutató osztálydiagram(ok). Tipikus hibalehetőségek: csillag-topológia, szigetek.]

4.1. ábra. x

4.3. Osztályok leírása

[Az előző alfejezetben tárgyalt objektumok felelősségének formalizálása attribútumokká, metódusokká. Csak publikus metódusok szerepelhetnek. Ebben az alfejezetben megjelennek az interfészek, az öröklés, az absztrakt osztályok. Segédosztályokra még mindig nincs szükség. Az osztályok ABC sorrendben kövessék egymást. Interfészek esetén az Interfészek, Attribútumok pontok kimaradnak.]

4.3.1. 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

4.3.2. 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: 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

4.4. 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.]

4.2. ábra. x

4.5. 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.]

4.3. ábra. x

4.6. Napló

Kezdet	Időtartam	Résztevők	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.
...

5. Szkeleton tervezése

5.1. A szkeleton modell valóságos use-case-ei

[A szkeletonnak, mint önálló programnak a működésével kapcsolatos use-case-ek.]

5.1.1. Use-case diagram

5.1. ábra. x

5.1.2. Use-case leírások

[Minden use-case-hez külön]

Use-case neve	...
Rövid leírás	...
Aktorok	...
Forgatókönyv	...

5.2. A szkeleton kezelői felületének terve, dialógusok

[A szkeleton által elfogadott bemenetek , valamint a szöveges konzolon megjelenő kimenetek. A kiemenet formátuma olyan kell legyen, ami alapján a működés összevethető a korábbi szekvencia-diagramokkal.]

5.3. Szekvencia diagramok a belső működésre

[A szkeletonban implementált szekvenciadiagramok. Tipikusan egy use-case egy diagram. Ezek megegyezhetnek a korábban specifikált diagramokkal, de az egyes éltvonalakat (lifeline) egyértelműen a szkeletonban példányosított objektumokhoz kell tudni kötni. Azt kell megjeleníteni, hogy a szkeletonban létrehozott objektumok egymással hogyan fognak kommunikálni.]

5.4. Kommunikációs diagramok

[A szkeletonban, az egyes szkeleton-use-case-ek futása során létrehozott objektumok és kapcsolataik bemutatására szolgáló diagramok. Ezek alapján valósítják meg a szkeleton fejlesztői az inicializáló kódrészleteket.]

5.5. Napló

Kezdet	Időtartam	Résztevők	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.

Kezdet	Időtartam	Résztevők	Leírás
...

6. Szkeleton beadás

6.1. Fordítási és futtatási útmutató

[A feltöltött program fordításával és futtatásával kapcsolatos útmutatás. Ennek tartalmaznia kell leltárszerűen az egyes fájlok pontos nevét, méretét byte-ban, keletkezési idejét, valamint azt, hogy a fájlban mi került megvalósításra.]

6.1.1. Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Main.java	250 byte	2009.10.10 18:05	...
...

6.1.2. Fordítás

[A fenti listában szereplő forrásfájlokból milyen műveletekkel lehet a bináris, futtatható kódot előállítani. Az előállításhoz csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.]

```
javac -d bin *.java
```

6.1.3. Futtatás

[A futtatható kód elindításával kapcsolatos teendők leírása. Az indításhoz csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.]

```
cd bin
java Main.java
```

6.2. Értékelés

[A projekt kezdete óta az értékelésig eltelt időben tagokra bontva, százalékban.]

Tag	Munka százalékban	Aláírás
Horváth	23.5 %	
Német	24.5 %	
Tóth	25 %	
Oláh	27 %	

6.3. Napló

Kezdet	Időtartam	Résztevők	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.
...

7. Prototípus koncepciója

7.1. 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.]

7.1.1. Az interfész általános leírása

[A protó (karakters) 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.]

7.1.2. Bemeneti nyelv

[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 futtatható legyen. A szálkezelést is tesztelhető, irányítható módon kell megoldani.]

- Parancs1
 - Leírás:
 - Opciók:
- Parancs2
 - Leírás:
 - Opciók:

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

7.1.3. Kimeneti nyelv

[Egyértelműen definiálni kell, hogy az egyes bemeneti parancsok végrehajtása után előálló állapot milyen formában jelenik meg a szabványos kimeneten.]

7.2. Összes részletes use-case

[A use-case-eknek a részletezettsége feleljen meg a kezelői felületnek, azaz a felület elemeire kell hivatkozniuk. Alábbi táblázat minden use-case-hez külön-külön.]

7.1. ábra. x

Use-case neve	...
Rövid leírás	...
Aktorok	...
Forgatókönyv	...

7.3. Tesztelési terv

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

Teszt-eset neve	...
Rövid leírás	...
Teszt célja	...

7.4. Tesztelést támogató segéd- és fordítóprogramok specifikálása

[Specifikálni kell a tesztelést támogató segédprogramokat.]

7.5. Napló

Kezdet	Időtartam	Résztevők	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.
...

8. Részletes tervek

8.1. Osztályok és metódusok tervei

8.1.1. Osztály1

- Felelősség
[Mi az osztály felelőssége. Kb 1 bekezdés. Ha szükséges, akkor state-chart is.]
- Ősosztályok
*[Mely osztályokból származik (öröklési hierarchia)
Legősebb osztály → Ősosztály2 → Ősosztály3...]*
- Interfészek
[Mely interfészeket valósítja meg.]
- Attribútumok
[Milyen attribútumai vannak]
 - attribútum1: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
 - attribútum2: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
- Metódusok
[Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.]
 - int foo(Osztály3 o1, Osztály4 o2): metódus leírása, láthatósága (UML jelöléssel)
 - int bar(Osztály5 o1): metódus leírása, láthatósága (UML jelöléssel)

8.1.2. Osztály2

- Felelősség
[Mi az osztály felelőssége. Kb 1 bekezdés. Ha szükséges, akkor state-chart is.]
- Ősosztályok
*[Mely osztályokból származik (öröklési hierarchia)
Legősebb osztály → Ősosztály2 → Ősosztály3...]*
- Interfészek
[Mely interfészeket valósítja meg.]
- Attribútumok
[Milyen attribútumai vannak]
 - attribútum1: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
 - attribútum2: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
- Metódusok
[Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.]
 - int foo(Osztály3 o1, Osztály4 o2): metódus leírása, láthatósága (UML jelöléssel)
 - int bar(Osztály5 o1): metódus leírása, láthatósága (UML jelöléssel)

8.2. A tesztek részletes tervei, leírásuk a teszt nyelvén

[A tesztek részletes tervei alatt meg kell adni azokat a bemeneti adatsorozatokot, amelyekkel a program működése ellenőrizhető. Minden bemenő adatsorozathoz definiálni kell, hogy az adatsorozat végrehajtásától a program mely részeinek, funkcióinak ellenőrzését várjuk és konkrétan milyen eredményekre számítunk, ezek az eredmények hogyan vethetők össze a bemenetekkel.]

8.2.1. Teszteset1

- Leírás
[szöveges leírás, kb. 1-5 mondat.]
- Ellenőrzött funkcionalitás, várható hibahelyek
- Bemenet
[a proto bemeneti nyelvén megadva (lásd előző anyag)]
- Elvárt kimenet
[a proto kimeneti nyelvén megadva (lásd előző anyag)]

8.2.2. Teszteset2

- Leírás
[szöveges leírás, kb. 1-5 mondat.]
- Ellenőrzött funkcionalitás, várható hibahelyek
- Bemenet
[a proto bemeneti nyelvén megadva (lásd előző anyag)]
- Elvárt kimenet
[a proto kimeneti nyelvén megadva (lásd előző anyag)]

8.3. A tesztelést támogató programok tervei

[A tesztadatok előállítására, a tesztek eredményeinek kiértékelésére szolgáló segédprogramok részletes terveit kell elkészíteni.]

8.4. 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.
...

10. Prototípus beadása

10.1. Fordítási és futtatási útmutató

[A feltöltött program fordításával és futtatásával kapcsolatos útmutatás. Ennek tartalmaznia kell leltárszerűen az egyes fájlok pontos nevét, méretét byte-ban, keletkezési idejét, valamint azt, hogy a fájlban mi került megvalósításra.]

10.1.1. Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Main.java	250 byte	2009.10.10 18:05	...
...

10.1.2. Fordítás

[A fenti listában szereplő forrásfájlokból milyen műveletekkel lehet a bináris, futtatható kódot előállítani. Az előállításához csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.]

```
javac -d bin *.java
```

10.1.3. Futtatás

[A futtatható kód elindításával kapcsolatos teendők leírása. Az indításhoz csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.]

```
cd bin
java Main.java
```

10.2. Tesztek jegyzőkönyvei

10.2.1. Teszteset1

[Az alábbi táblázatot az utolsó, sikeres tesztfuttatáshoz kell kitölteni]

Tesztelő neve	...
Teszt időpontja	...

[Az alábbi táblázatot a megismételt (hibás) tesztek esetén kell kitölteni minden ismétléshez egyszer. Ha szükséges, akkor a valós kimenet is mellékelhető mint a teszt eredménye.]

Tesztelő neve	...
Teszt időpontja	...
Teszt eredménye	...
Lehetséges hibák	...
Változtatások	...

10.3. Értékelés

[A projekt kezdete óta az értékelésig eltelt időben tagokra bontva, százalékban.]

Tag	Munka százalékban	Aláírás
Horváth	23.5 %	
Németh	24.5 %	
Tóth	25 %	
Oláh	27 %	

10.4. Napló

Kezdet	Időtartam	Résztevők	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.
...

11. Grafikus felület specifikációja

11.1. A grafikus interfész

[A menürendszer, a kezelői felület grafikus képe. A grafikus felület megjelenését, a használt ikonokat, stb screenshot-szerű képekkel kell bemutatni. Az építészetben ez a homlokzati terv.]

11.1. ábra. x

11.2. A grafikus rendszer architektúrája

[A felület működésének elve, a grafikus rendszer architektúrája (struktúra diagramok). A struktúra diagramokon a prototípus azon és csak azon osztályainak is szerepelnie kell, amelyekhez a grafikus felületet létrehozó osztályok kapcsolódnak.]

11.2.1. A felület működési elve

[Le kell írni, hogy a grafikai megjelenésért felelős osztályok, objektumok hogyan kapcsolódnak a meglévő rendszerhez, a megjelenítés során mi volt az alapelv. Törekedni kell az MVC megvalósításra. Alapelvek lehetnek: **push** alapú: a modell értesíti a felületet, hogy változott; **pull** alapú: a felület kérdezi le a modellt, hogy változott-e; **kevert**: a kettő kombinációja.]

11.2.2. A felület osztály-struktúrája

[Osztálydiagram. Minden új osztály, és azon régiak, akik az újakhoz közvetlenül kapcsolódnak.]

11.3. A grafikus objektumok felsorolása

[Az új osztályok felsorolása. Az régi osztályok közül azoknak a felsorolása, ahol változás volt. Ezek esetén csak a változásokat kell leírni.]

11.3.1. Osztály1

- Felelősség
[Mi az osztály felelőssége. Kb 1 bekezdés. Ha szükséges, akkor state-chart is.]
- Ősosztályok
[Mely osztályokból származik (öröklési hierarchia)
Legősebb osztály → Ősosztály2 → Ősosztály3...]
- Interfészek
[Mely interfészeket valósítja meg.]
- Attribútumok
[Milyen attribútumai vannak]
 - attribútum1: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
 - attribútum2: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa

- Metódusok

[Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.]

- int foo(Osztály3 o1, Osztály4 o2): metódus leírása, láthatósága (UML jelöléssel)
- int bar(Osztály5 o1): metódus leírása, láthatósága (UML jelöléssel)

11.3.2. Osztály2

- Felelősség

[Mi az osztály felelőssége. Kb 1 bekezdés. Ha szükséges, akkor state-chart is.]

- Ősosztályok

*[Mely osztályokból származik (öröklési hierarchia)
Legősebb osztály → Ősosztály2 → Ősosztály3...]*

- Interfészek

[Mely interfészeket valósítja meg.]

- Attribútumok

[Milyen attribútumai vannak]

- attribútum1: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
- attribútum2: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa

- Metódusok

[Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.]

- int foo(Osztály3 o1, Osztály4 o2): metódus leírása, láthatósága (UML jelöléssel)
- int bar(Osztály5 o1): metódus leírása, láthatósága (UML jelöléssel)

11.4. Kapcsolat az alkalmazói rendszerrel

[Szekvencia-diagramokon ábrázolni kell a grafikus rendszer működését. Konzisztens kell legyen az előző alfejezetekkel. Minden metódus, ami ott szerepel, fel kell tűnjön valamelyik szekvenciában. Minden metódusnak, ami szekvenciában szerepel, szereplnie kell a valamelyik osztálydiagramon.]

11.5. Napló

Kezdet	Időtartam	Résztevők	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.
...

13. Grafikus felület specifikációja

13.1. Fordítási és futtatási útmutató

[A feltöltött program fordításával és futtatásával kapcsolatos útmutatás. Ennek tartalmaznia kell leltárszerűen az egyes fájlok pontos nevét, méretét byte-ban, keletkezési idejét, valamint azt, hogy a fájlban mi került megvalósításra.]

13.1.1. Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Main.java	250 byte	2009.10.10 18:05	...
...

13.1.2. Fordítás

[A fenti listában szereplő forrásfájlokból milyen műveletekkel lehet a bináris, futtatható kódot előállítani. Az előállításához csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.]

```
javac -d bin *.java
```

13.1.3. Futtatás

[A futtatható kód elindításával kapcsolatos teendők leírása. Az indításhoz csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.]

```
cd bin
java Main.java
```

13.2. Értékelés

[A projekt kezdete óta az értékelésig eltelt időben tagokra bontva, százalékban.]

Tag	Munka százalékban	Aláírás
Horváth	23.5 %	
Német	24.5 %	
Tóth	25 %	
Oláh	27 %	

13.3. Napló

Kezdet	Időtartam	Résztevők	Leírás
2020.02.21. 14:30	1,5 óra	Lant Kiss Kondrád Máté Glávits	Use-Case, Class Diagram közös javítása
...

14. Összefoglalás

14.1. Projekt összegzés

[A projekt tapasztalatait összegző részben a csapatoknak a projektről kialakult véleményét várjuk. A megválaszolandók köre az alábbi.]

Tag	Munkaidő (óra)
Horváth	98
Németh	95
Tóth	102
Oláh	87
Összesen:	382

Fázis	Forrássor
Szkeleton	500
Prototípus	600
Grafikus változat	700
Összesen	1800

- Mit tanultak a projektből konkrétan és általában?
- Mi volt a legnehezebb és a legkönnyebb?
- Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?
- Ha nem, akkor hol okozott ez nehézséget?
- Milyen változtatási javaslatuk van?
- Milyen feladatot ajánlanának a projektre?

[Szívesen fogadunk minden egyéb kritikát és javaslatot.]