

8. Részletes tervek

8.1 Osztályok és metódusok tervei.

8.1.1 Bank

- **Felelősség**

A játékos avatárjának kiindulópontja.

- **Ősosztályok**

- Building

- **Interfészek**

- nincs

- **Attribútumok**

- **Metódusok**

- GetType():visszatér egy az osztályra mutató referenciával

8.1.2 Building

- **Felelősség**

A városban lévő épületek, felelősségük a szerepkörük nyilvántartása.

- **Ősosztályok**

- nincs

- **Interfészek**

- nincs

- **Attribútumok**

- **Metódusok**

- GetType():visszatér egy az osztályra mutató referenciával

8.1.3 Bunny

- **Felelősség**

Ha a rabló átmegy rajta egy ideig immortal lesz.

- **Ősosztályok**

nincs

- **Interfészek**

nincs

- **Attribútumok**

- **int speed** : a kocs sebessége, hány „steppenként” lép az autó
- **int timetomove** : várakozási idő lámpák és táblák esetén

- **Metódusok**

- **void pass(Car)** : az előzést végző függvény

8.1.4 Car

- **Felelősség**

A városban közlekedő autók, felelősségük a szabályok betartása és az ütközések elkerülése, sebességük nyilvántartása.

- **Ősosztályok**

nincs

- **Interfészek**

nincs

- **Attribútumok**

- **int speed** : a kocs sebessége, hány „steppenként” lép az autó
- **int timetomove** : várakozási idő lámpák és táblák esetén

- **Metódusok**

- **int step()** : a mozgást valósítja meg
- **void move()** : az ütközésselkerülést segítő metódus
- **void pass(Car)** : az előzést végző függvény
- **void setSpeed(int)** : beállítja az autó sebességét
- **int getSpeed()** : visszaadja az autó sebsségét

- **void destroy()** : törli az autót

8.1.5 City

- **Felelősség**
A várost reprezentáló osztály, felelőssége az inicializálás irányítása a léptetés vezérlése.
- **Össztályok**
nincs
- **Interfészek**
nincs
- **Attribútumok**
 - **LinkedList<Car> car** : a városban közlekedő autók
 - **LinkedList<RoadBlock> road** : a város útszerkezete
 - **LinkedList<ITraffic> traffic** : a városban lévő közlekedési szabályok
 - **LinkedList<Building> building** : a városban található épületek
- **Metódusok**
 - **void step()** : a város mozgatása

8.1.6 Hideout

- **Felelősség**
Jelezni a játékosnak a játék végét, ha elérte.
- **Össztályok**
Building
- **Interfészek**
nincs
- **Attribútumok**
nincs
- **Metódusok**
 - **int getType()** : visszaadja az épület szerepkörét

8.1.7 ITraffic

- **Felelősség**
A közlekedés szabályozása, interfacet nyújt a közlekedési szabályokat megvalósító osztályok számára.
- **Ősosztályok**
nincs
- **Interfészek**
nincs
- **Attribútumok**
nincs
- **Metódusok**
 - **int getState()** : megadja a jelzés állapotát
 - **void step()** : a váltásokat végző függvény
 - **ITraffic getTraffic()**: visszatér egy az objektumra mutató referenciával

8.1.8 Police

- **Felelősség**
A rendőrt reprezentáló objektum. Felelőssége ha elkapta a rablót megszakítani a játékot.
- **Ősosztályok**
Car
- **Interfészek**
nincs
- **Attribútumok**
- **Metódusok**
 - **void pass(Car)** : ha egy rabló próbálja megelőzni akkor azt elkapja, egyébként Carként viselkedik

8.1.9 RoadBlock

- **Felelősség**
Az utat építi fel, felelőssége a rajta található autók , jelzőlámpák, -táblák és az őt körülvevő RoadBlockok nyilvántartása.
- **Ősosztályok**
nincs

- **Interfészek**
nincs
- **Attribútumok**
 - **Car& car** : a rajta található autóra mutató referencia
 - **ITraffic[] traffic** : a rajta található táblák, és lámpák
 - **Building building** : a rajta található épületek
 - **RoadBlock road** : a vele szomszédos útdarabok
- **Metódusok**
 - **void setNeighbour(Neighbour)** : beállítja a szomszédos útdarabokat
 - **void setCar(Car)** : beállítja a rajta lévő autót
 - **void setBuilding(Building)** : beállítja a hozzá tartozó épület
 - **void setTraffic(ITraffic)** : beállítja a hozzá tartozó lámpákat, táblákat
 - **RoadBlock[] getNeighbour()** : visszaadja a szomszédos útdarabokat
 - **Car getCar()** : visszaadja a rajta lévő autót
 - **Building[] getBuilding()** : visszaadja a hozzá tartozó épület
 - **ITraffic[] getTraffic()** : visszaadja a hozzá tartozó lámpákat, táblákat

8.1.10 Robber

- **Felelősség**
A rabló, őt irányíthatjuk a játék során, felelőssége a rendőr detektálása.
- **Ősosztályok**
Car
- **Interfészek**
nincs
- **Attribútumok**
- **Metódusok**
 - **void pass(Car)** : az előtte haladó autó megelőzése
 - **void step()** : a mozgása eltér a többi járművétől, hiszen nem vonatkoznak rá szabályok

8.1.11 TrafficSign

- **Felelősség**
A jelzőlámpa, felelőssége az állapotának változtatása és annak közlése.
- **Ősosztályok**
nincs
- **Interfészek**
ITraffic

- **Attribútumok**
- **Metódusok**

8.1.12 TrafficTable

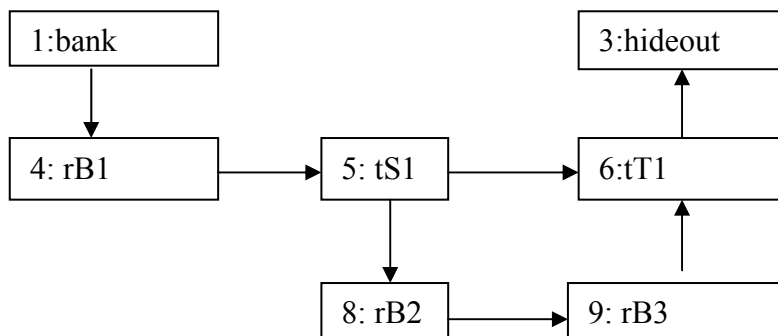
- **Felelősség**
Tábla, felelőssége a vonatkozó szabály közlése.
- **Ősosztályok**
nincs
- **Interfészek**
ITraffic
- **Attribútumok**

8.2 A tesztek részletes tervei, leírásuk a teszt nyelven

Pályák leírásánál a következő mintát kell alkalmazni:

A térképet egy irányított szomszédossági mátrix formájában kell a programnak beadni egy szöveges fájlban.

Példa a térkép megadására:



Ahol a várost, mint egy 3*3 mezős rácsot képzelhetjük el. Ahol nincsen semmi, pl a kettes helyen, ott ki kell hagyni a számozást. A nyilak jelentik az irányított utakat az adott objektumok között.

Jelmagyarázat:

bank = Bank (1)

hideout = Hideout (2)

rB = RoadBlock, (3)

tS =TrafficSign, (4)

tT = TrafficTable (5)

Zárójelben az osztályok neve mögötti szám jelenti, hogy az irányított szomszédossági mátrixban hányas számmal hivatkozunk az objektum típusára, a főátlóban.

A konkrét térképet reprezentáló mátrix:

	1	2	3	4	5	6	7	8	9
1	1			1					
2		0							
3			2			-1			
4	-1			3	1				
5				-1	4	1		1	
6			1		-1	5			-1
7							0		
8					-1			3	1
9						1		-1	3

Ez alapján a **mapfile.txt** tartalma, amivel a tesztelést fogjuk végezni:

```

1    0    0    1    0    0    0    0    0
0    0    0    0    0    0    0    0    0
0    0    2    0    0    -1   0    0    0
-1   0    0    3    1    0    0    0    0
0    0    0    -1   4    1    0    1    0
0    0    1    0    -1   5    0    0    -1
0    0    0    0    0    0    0    0    0
0    0    0    0    -1   0    0    3    1
0    0    0    0    0    1    0    -1   3

```

8.2.1 Teszteset - Inicializálás

- **Leírás**

A teszteset célja annak ellenőrzése, a játék inicializálásakor betöltendő térképen szereplő objektumok megfelelően példányosodnak-e.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Pályához tartozó objektumok létrehozásának ellenőrzése. Esetleges hibalehetőség, ha valamely objektum nem jön létre, vagy attribútumai nem megfelelő értékeket vesznek fel.

- **Bemenet**

```
map mapfile.txt
save state.txt
```

- **Elvárt kimenet**

A state.txt tartalma a következőképp kell, hogy alakuljon:

```

OBJECT    city    TYPE    City
car        NULL
road[0]    rB_bank
road[1]    rB_hideout
road[2]    rB1
road[3]    rB_tS1
road[4]    rB_tT1
road[5]    rB2
road[6]    rB3
traffic[0]    tS1
traffic[1]    tT1
building[0]    bank
building[1]    hideout
OBJECT    rB_bank    TYPE RoadBlock
car        NULL
traffic    NULL
building    bank
road        rB1
OBJECT    rB_hideout    TYPE RoadBlock
car        NULL
traffic    NULL
building    hideout
road        NULL
OBJECT    rB1    TYPE    RoadBlock
car        NULL

```



```

traffic    NULL
building   NULL
road       rB_tS1
OBJECT     rB_tS1 TYPE    RoadBlock
car        NULL
traffic    tS1
building   NULL
road       [rB2,rB_tT1]
OBJECT     rB_tT1 TYPE    RoadBlock
car        NULL
traffic    tT1
road       rB_hideout
OBJECT     rB2      TYPE    RoadBlock
car        NULL
traffic    NULL
road       rB3
OBJECT     rB3      TYPE    RoadBlock
car        NULL
traffic    NULL
road       rB_tT1
OBJECT     tS1      TYPE    TrafficSign
state      0
OBJECT     tT1      TYPE    TrafficTable
state      0
OBJECT     bank      TYPE    Bank
OBJECT     hideout   TYPE    Hideout

```

8.2.2 Teszteset – Rabló irányítása

- **Leírás**
Rabló irányíthatóságának tesztelése, sebesség, irány módosítása, rabló elvezetése a rejtekhelyig.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Rabló reagálása az utasításokra, forgalomirányítás tesztelése.
- **Bemenet**

```

map mapfile.txt
load statefile.txt
move up
step 2
move left
step
save state.txt

```

Ahol statefile.txt a következőket tartalmazza:

```

OBJECT     car[0]      TYPE    Robber
speed      10
timetomove      5
OBJECT     road[2]     TYPE    RoadBlock
car car[0]

```

- **Elvárt kimenet**

```

OBJECT    city    TYPE    City
car[0]    car[0]
road[0]    rB_bank
road[1]    rB_hideout
road[2]    rB1
road[3]    rB_tS1
road[4]    rB_tT1
road[5]    rB2
road[6]    rB3
traffic[0]    tS1
traffic[1]    tT1
building[0]    bank
building[1]    hideout
OBJECT    rB_bank    TYPE    RoadBlock
car        NULL
traffic    NULL
building   bank
road       rB1
OBJECT    rB_hideout    TYPE    RoadBlock
car        car[0]
traffic    NULL
building   hideout
road       NULL
OBJECT    rB1    TYPE    RoadBlock
car        NULL
traffic    NULL
building   NULL
road       rB_tS1
OBJECT    rB_tS1    TYPE    RoadBlock
car        NULL
traffic    tS1
building   NULL
road       [rB2,rB_tT1]
OBJECT    rB_tT1    TYPE    RoadBlock
car        NULL
traffic    tT1
road       rB_hideout
OBJECT    rB2    TYPE    RoadBlock
car        NULL
traffic    NULL
road       rB3
OBJECT    rB3    TYPE    RoadBlock
car        NULL
traffic    NULL
road       rB_tT1
OBJECT    tS1    TYPE    TrafficSign
state     0
OBJECT    tT1    TYPE    TrafficTable
state     0
OBJECT    bank    TYPE    Bank
OBJECT    hideout    TYPE    Hideout

```

```

OBJECT    car[0] TYPE    Robber
speed     30
timetomove      0-255  int

```

8.2.3 Teszteset – Rendőr elkapja a rablót

- **Leírás**
Rendőr a rabló mögül indul, és 2 lépés után elkapja.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Rendőr el tudja-e kapni a rablót, léptetések megfelelően mennek-e.
- **Bemenet**
map mapfile.txt
load statefile.txt
step 4

Ahol statefile.txt a következőket tartalmazza:

```

OBJECT    car[0]      TYPE Robber
speed     10
timetomove      5
health     5
OBJECT    car[1]      TYPE Police
speed     30
timetomove      5
OBJECT    road[3]     TYPE RoadBlock
car car[0]
OBJECT    road[2]     TYPE RoadBlock
car car[1]

```

- **Elvárt kimenet**

```

OBJECT    city  TYPE    City
car[0]    car[0]
car[1]    car[1]
road[0]   rB_bank
road[1]   rB_hideout
road[2]   rB1
road[3]   rB_tS1
road[4]   rB_tT1
road[5]   rB2
road[6]   rB3
traffic[0]      tS1
traffic[1]      tT1
building[0]     bank
building[1]     hideout
OBJECT    rB_bank TYPE RoadBlock
car       NULL
traffic   NULL
building  bank
road      rB1
OBJECT    rB_hideout TYPE RoadBlock

```

```

car          NULL
traffic      NULL
building     hideout
road         NULL
OBJECT      rB1      TYPE      RoadBlock
car          NULL
traffic      NULL
building     NULL
road         rB_tS1
OBJECT      rB_tS1 TYPE      RoadBlock
car          car[1]
traffic      tS1
building     NULL
road         [rB2,rB_tT1]
OBJECT      rB_tT1 TYPE      RoadBlock
car          car[0]
traffic      tT1
road         rB_hideout
OBJECT      rB2      TYPE      RoadBlock
car          NULL
traffic      NULL
road         rB3
OBJECT      rB3      TYPE      RoadBlock
car          NULL
traffic      NULL
road         rB_tT1
OBJECT      tS1      TYPE      TrafficSign
state       0
OBJECT      tT1      TYPE      TrafficTable
state       0
OBJECT      bank      TYPE      Bank
OBJECT      hideout   TYPE      Hideout
OBJECT      car[0] TYPE      Robber
speed       30
timetomove  0-255 int
health      0

```

8.2.4 Teszteset – Ütközésselkerülés autókkal

- **Leírás**
Azt az esetet modellezi, amikor két autó halad ugyan azon az útvonalon, és a hátsó gyorsabban halad, beéri a másikat. Ekkor a gyorsabbnak le kell lassítania az előtte lévő sebességére.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ütközésselkerülés, sebességmódosulás.
- **Bemenet**
- map mapfile.txt
- load statefile.txt
step 1
Ahol statefile.txt a következőket tartalmazza:
OBJECT car[0] TYPE Car

```

speed      10
timetomove    5
OBJECT      car[1]    TYPE Car
speed      30
timetomove    5
OBJECT      road[3]    TYPE RoadBlock
car car[0]
OBJECT      road[2]    TYPE RoadBlock
car car[1]

```

- **Elvárt kimenet**

```

OBJECT      city    TYPE    City
car[0]      car[0]
car[1]      car[1]
road[0]     rB_bank
road[1]     rB_hideout
road[2]     rB1
road[3]     rB_tS1
road[4]     rB_tT1
road[5]     rB2
road[6]     rB3
traffic[0]          tS1
traffic[1]          tT1
building[0]         bank
building[1]         hideout
OBJECT      rB_bank    TYPE RoadBlock
car          NULL
traffic      NULL
building     bank
road        rB1
OBJECT      rB_hideout    TYPE RoadBlock
car          NULL
traffic      NULL
building     hideout
road        NULL
OBJECT      rB1    TYPE    RoadBlock
car          NULL
traffic      NULL
building     NULL
road        rB_tS1
OBJECT      rB_tS1 TYPE    RoadBlock
car          car[1]
traffic      tS1
building     NULL
road        [rB2,rB_tT1]
OBJECT      rB_tT1 TYPE    RoadBlock
car          car[0]
traffic      tT1
road        rB_hideout

```

```

OBJECT    rB2      TYPE    RoadBlock
car       NULL
traffic   NULL
road      rB3
OBJECT    rB3      TYPE    RoadBlock
car       NULL
traffic   NULL
road      rB_tT1
OBJECT    tS1      TYPE    TrafficSign
state     0
OBJECT    tT1      TYPE    TrafficTable
state     0
OBJECT    bank      TYPE    Bank
OBJECT    hideout   TYPE    Hideout
OBJECT    car[0]    TYPE    Car
speed     10
OBJECT    car[1]    TYPE    Car
speed     10

```

8.2.5 Teszteset – Ütközésselkerülés Rablóval

- **Leírás**
Rabló ér be egy autót, meg kell előzzze.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ütközésselkerülés, sebességmódosulás.

- **Bemenet**
map mapfile.txt
load statefile.txt
step 1
Ahol statefile.txt a következőket tartalmazza:

```

OBJECT    car[0]      TYPE    Car
speed     10
timetomove    5
OBJECT    car[1]      TYPE    Robber
speed     30
timetomove    5
OBJECT    road[3]     TYPE    RoadBlock
car  car[0]
OBJECT    road[2]     TYPE    RoadBlock
car  car[1]

```

- **Elvárt kimenet**

```

OBJECT    city      TYPE    City
car[0]    car[0]
car[1]    car[1]
road[0]   rB_bank
road[1]   rB_hideout
road[2]   rB1
road[3]   rB_tS1

```

```

road[4]    rB_tT1
road[5]    rB2
road[6]    rB3
traffic[0]    tS1
traffic[1]    tT1
building[0]    bank
building[1]    hideout
OBJECT    rB_bank    TYPE RoadBlock
car        NULL
traffic    NULL
building    bank
road        rB1
OBJECT    rB_hideout    TYPE RoadBlock
car        NULL
traffic    NULL
building    hideout
road        NULL
OBJECT    rB1    TYPE    RoadBlock
car        NULL
traffic    NULL
building    NULL
road        rB_tS1
OBJECT    rB_tS1 TYPE    RoadBlock
car        car[0]
traffic    tS1
building    NULL
road        [rB2,rB_tT1]
OBJECT    rB_tT1 TYPE    RoadBlock
car        car[1]
traffic    tT1
road        rB_hideout
OBJECT    rB2    TYPE    RoadBlock
car        NULL
traffic    NULL
road        rB3
OBJECT    rB3    TYPE    RoadBlock
car        NULL
traffic    NULL
road        rB_tT1
OBJECT    tS1    TYPE    TrafficSign
state      0
OBJECT    tT1    TYPE    TrafficTable
state      0
OBJECT    bank    TYPE Bank
OBJECT    hideout TYPE Hideout
OBJECT    car[0] TYPE Car
speed      10
OBJECT    car[1] TYPE Car
speed      30

```

8.2.6 Teszteset – Nyuszielütés

- **Leírás**
Rablót üldözi a rendőr, rabló áthajt a nyuszin, a rendőr egy ideig nem tudja elkapni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Nyuszi elütése valóban immunisságot ad-e a rablónak.

- **Bemenet**

```
map mapfile.txt
load statefile.txt
step 4
```

Ahol statefile.txt a következőket tartalmazza:

```
OBJECT    car[0]      TYPE Robber
speed     10
timetomove    5
health    5
OBJECT    car[1]      TYPE Police
speed     20
timetomove    5
OBJECT    car[2]      TYPE Bunny
speed     0
timetomove   100
OBJECT    road[3]     TYPE RoadBlock
car car[0]
OBJECT    road[2]     TYPE RoadBlock
car car[1]
OBJECT    road[4]     TYPE RoadBlock
car car[2]
```

- **Elvárt kimenet**

```
OBJECT    city      TYPE    City
car[0]    car[0]
car[1]    car[1]
car[2]    car[2]
road[0]   rB_bank
road[1]   rB_hideout
road[2]   rB1
road[3]   rB_tS1
road[4]   rB_tT1
road[5]   rB2
road[6]   rB3
traffic[0]      tS1
traffic[1]      tT1
building[0]     bank
building[1]     hideout
OBJECT    rB_bank TYPE RoadBlock
car       NULL
traffic   NULL
building  bank
road      rB1
```



```

OBJECT    rB_hideout    TYPE RoadBlock
car        NULL
traffic    NULL
building   hideout
road       NULL
OBJECT    rB1      TYPE    RoadBlock
car        NULL
traffic    NULL
building   NULL
road       rB_tS1
OBJECT    rB_tS1 TYPE    RoadBlock
car        car[1]
traffic    tS1
building   NULL
road       [rB2,rB_tT1]
OBJECT    rB_tT1 TYPE    RoadBlock
car        car[0]
traffic    tT1
road       rB_hideout
OBJECT    rB2      TYPE    RoadBlock
car        NULL
traffic    NULL
road       rB3
OBJECT    rB3      TYPE    RoadBlock
car        NULL
traffic    NULL
road       rB_tT1
OBJECT    tS1      TYPE    TrafficSign
state     0
OBJECT    tT1      TYPE    TrafficTable
state     0
OBJECT    bank      TYPE    Bank
OBJECT    hideout   TYPE    Hideout
OBJECT    car[0] TYPE    Robber
speed     30
timetomove 0-255 int
health    5
immortality    10

```

8.3 A tesztelést támogató programok tervei

A teszteredmények összehasonlításának megkönnyítése érdekében készítettünk egy egyszerű fájl összehasonlító programot Java nyelven. A konzolos program parancssori paramétereiben megkapja az összehasonlítandó két fájl nevét, a relatív elérési útvonalukkal együtt. A működése egyszerűen csak annyi, hogy a két fájlt szinkronban beolvassa, és soronként összehasonlítja. Ha eltérés van, akkor azt jelzi a felhasználónak, kiemelve a két fájlban található sort. A program futása végén jelzi, hogy végeredményben egyezik, vagy sem a két fájl. Ha hiba történt a fájlok beolvasása során, akkor ezt hibaüzenettel jelzi.

A megvalósításhoz készítettünk egy különálló osztályt (InputFile), ami a bemeneti fájlt reprezentálja. A főprogramban ezt kétszer példányosítva, majd metódusai segítségével, a fájlokat összehasonlítva értékeljük ki az eredményeket.

A programhoz készült egy JavaDOC is, melyben tételesen megtalálhatóak a program függvényei is, rövid leírásokkal együtt.

Ha hibás bemeneti argumentumokkal futtatjuk a szoftvert, akkor az kiírja a helyes használathoz szükséges szintaxist:

```
Error: Missing or too many argument(s).
Syntax: compare <file1> <file2>
Example: compare out.txt sample.txt
```

Ha helyes paraméterekkel indítjuk, de a két bemeneti fájl közül valamelyik nem található, akkor a következő kimenetet kaphatjuk:

```
Reading files...
Error: File not found.
```

Ha a fájlok léteznek, és az argumentumok is helyesen lettek megadva, illetve a két fájl egyezik, akkor a kimenet a következő: (test1.txt, és test2.txt az összehasonlított fájlok)

```
Reading files...
Comparing file test1.txt to file test2.txt ...
Full match.
```

Ha eltérés van a fájlok között, akkor azokat tételesen felsorolja az alábbiak szerint: (test1.txt, és test2.txt az összehasonlított fájlok, g11, g1, er, err a tartalmazott szövegek, line 1, line 5 értelemszerűen a sorszámok, ahol ezek találhatóak)

```
Reading files...
Comparing file test1.txt to file test2.txt ...

Row mismatch at line 1 in file test1.txt compared to file
test2.txt
test1.txt: g11
test2.txt: g1

Row mismatch at line 5 in file test1.txt compared to file
test2.txt
```

```
test1.txt: en  
test2.txt: ern
```

```
Files are different.
```

8.4 Változás a specifikációban

A húsvéti nyuszi időnként feltűnik az utakon. Ha a rabló elüti a nyuszt, akkor rövid időre immunitást élvez, vagyis a rendőrök ekkor nem tudják elkapni, bárhogyan próbálkoznak is.

A húsvéti nyulat a Bunny osztály fogja reprezentálni, aki járműként mozoghat a városban. Így a Bunny a Car ősosztályból származik, rendelkezik a feltűnés időtartamát meghatározó int típusú változóval. Továbbá a rabló osztály implementálunk egy pass(Bunny) : void metódussal ami az immunitását jelző belső változó értékét állítja be, amely minden stepben csökken.

8.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2010.03.29.18:00	0,5 óra	Rapp Takács Molnár Boros	Megbeszélés
2010.03.30.18:00	2 óra	Rapp	8.2 Tesztek részletes terveinek kidolgozása
2010.03.31.17:00	4 óra	Rapp	8.2.* Tesztesetek részletes kidolgozása
2010.03.31.19:00	2 óra	Takács	8.3.* A tesztelést támogató programok tervei rész elkészítése
2010.03.31.20:00	2 óra	Molnár	8.1 Osztályleírások elkészítése
2010.03.30.23:00	2 óra	Boros	8.3 Specifikáció-változás kidolgozása, dokumentum javítása, formázása