

7. Prototípus koncepciója

7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A prototípus egy karakteres felületű program, melynek elsődleges célja a játék működésének demonstrálása. A szkeleton verzióval ellentétben, itt már nem a belső függvényhívásokra vagyunk kíváncsiak, hanem sokkal inkább a teljes játék logikájának helyes működésére. A könnyű tesztelhetőség végett a program képes bármikor kimenteni az aktuális állapotát egy fájlba, illetve onnan visszaolvasni azt. Lehetőséget biztosítunk arra is, hogy automatizált módon egy előre megírt szkript szerint kapjon utasításokat a programunk, ugyanakkor természetesen ezeket a standard inputon is olvasni képes. A tesztelés szempontjából lényeges, hogy a véletlenszerűen történő események is determinisztikus módon történjenek meg. Ezért a programban lehetőséget biztosítunk az események determinisztikus és véletlenszerű lefutásának futás idejű váltására.

Alapvető koncepció, hogy a tesztek bármikor könnyedén felismerhetők, illetve reprodukálhatóak legyenek. A csapat is eszerint állítja elő a prototípust.

7.1.2 Bemeneti nyelv

A legtöbb itt felsorolt parancs a program végleges verziójában nem kerül felhasználásra, mert csak tesztelési célokat szolgál. Ettől függetlenül lehetőséget fogunk biztosítani arra, hogy a program grafikus változatában is használhatóak legyenek. A prototípust a felhasználó a következő parancsokkal vezérelheti:

map <mapfile.txt>

Leírás: Betölt egy pályát a <mapfile.txt> fájlból, és felépíti.

Opciók:

- **<mapfile.txt>** egy szöveges fájl, melyben a játék egy pályája található meg. A pályaleíró fájl egyszerű szomszédossági mátrixot tárol, ahol a főátlóban kódoljuk, hogy mi van az adott útdarabkán. (pl. jelzőlámpa, stop tábla, bank, stb...)

run <scriptfile.txt> [echo]

Leírás: Betölt egy utasítássorozatot a <scriptfile.txt> fájlból, majd végrehajtja azt.

Opcionálisan kérhető az [echo] paraméterrel, hogy végrehajtáskor az éppen soron következő utasítást a szabványos outputra is kiírja a program.

Opciók:

- **<scriptfile.txt>** egy szöveges fájl, melyben a program által értelmezhető utasítások találhatóak, sorban, egymás után.
- **[echo]** Opcionális bináris értékkészletű {0, 1} paraméter, melynek alapértéke 0. Értékei a következőképp értelmezendők:
 - **1** = Az éppen soron következő utasítás megjelenik a szabványos kimeneten is.
 - **0** = Az utasítások „néma” módban hajtódnak végre. Ilyenkor csak az adott utasítás hatására jelenhetnek meg tájékoztató üzenetek, vagy egyéb kimenetek. Maguk az utasítások nem kerülnek kiírásra.

load <statefile.txt>

Leírás: Betölt egy játékállapotot a <statefile.txt> fájlból. Az aktuális állapot törlődik a memóriából, helyére a beolvasott kerül.

Opciók:

- <statefile.txt> egy szöveges fájl, melyben a játék egy adott pillanatbeli állapota található meg, egy könnyedén értelmezhető formátumban.

save <statefile.txt> [logfile.txt]

Leírás: Menti a játék aktuális állapotát a <statefile.txt> fájlba. Ez a fájl értelmezhető a load parancs számára is. Opcionálisan képes elmenteni a [logfile.txt] fájlba a képernyőn megjelenített információkat is, beleértve a felhasználói inputot, és a program konzolra írt outputját.

Opciók:

- <statefile.txt> egy szöveges fájl, melyben a játék egy adott pillanatbeli állapota található meg, egy könnyedén értelmezhető formátumban.
- [logfile.txt] egy szöveges fájl, melyben könnyen értelmezhető módon vannak tárolva a program képernyőjén megjelenített információk. Opcionális paraméter, melyet, ha megadunk, akkor a megadott fájlba ment, egyébként semmit sem csinál.

random <mode>

Leírás: Beállítja az események (döntések) viselkedését. Lehet választani véletlenszerű és determinisztikus között.

Opciók:

- <mode> egy bináris {0, 1} paraméter, mely értékei az alábbiakat jelentik:
 - 1 = a döntések véletlenszerűek.
 - 0 = a döntések determinisztikusak.

exit

Leírás: Kilép a programból.

Opciók: -

step [count]

Leírás: [count] számú lépést vált ki. Ha nincs paraméter, akkor csak egyet lép.

Opciók:

- [count] egy opcionális pozitív egész paraméter, amelynek megfelelő számú lépés hajtódik végre a programban.

win

Leírás: Hatása olyan, mintha a bankrabló elért volna a rejtekhelyig.

Opciók: -

loose

Leírás: Hatása olyan, mintha a rendőr elkapta volna a rablót.

Opciók: -

bunny

Leírás: Hatása olyan, mintha a rabló elütött volna egy húsvéti nyulat.

Opciók: -

rob

Leírás: Hatása olyan, mintha a rabló kirabolta volna a bankot.

Opciók: -

money <amount>

Leírás: Hatására a rabló pénze <amount> lesz.

Opciók:

- **<amount>** Pozitív egész szám, vagy 0. A pénzmennyiséget reprezentálja.

health <hp>

Leírás: Hatására a rablónak <hp> életpontja lesz.

Opciók:

- **<hp>** Pozitív egész szám, vagy 0. A rabló életpontját reprezentálja.

speed <v>

Leírás: Hatására a rabló autója <v> sebességgel fog menni.

Opciók:

- **<v>** Pozitív egész szám, vagy 0. A rabló sebességét reprezentálja.

move <dir>

Leírás: Hatása olyan, mintha a <dir> -nek megfelelő gombot nyomtuk volna le az iránybeállító billentyűzeten.

Opciók:

- **<dir>** Egy négyelemű értékkészletből {up, down, left, right} előálló paraméter, melynek értelmezése az alábbi:
 - **up** = Hatása olyan, mintha a „fel” gombot nyomtuk volna meg.
 - **down** = Hatása olyan, mintha a „le” gombot nyomtuk volna meg.
 - **left** = Hatása olyan, mintha a „bal” gombot nyomtuk volna meg.
 - **right** = Hatása olyan, mintha a „jobb” gombot nyomtuk volna meg.

say <message>

Leírás: Hatására a <message> üzenet fog megjelenni a képernyőn.

Opciók:

- **<message>** Tetszőleges hosszúságú szöveg.

7.1.3 Kimeneti nyelv

A program két fájlt készít tesztelés során. Egyet, amiben a képernyőn megjelenő eseményeket „logolja”, és egyet, amibe elmenti a programban szereplő összes objektum állapotát.

Ezt a mentést a program a következő utasításra készíti el:

```
save state.txt log.txt
```

Ahol a **save** kulcsszó indítja a mentést (fájlba írást), első paraméter a program állapotainak létrehozandó fájl neve, második pedig a képernyőn történt események mentési helye. A második opcionális, az első kötelező.

Példa a log fájl tartalmára:

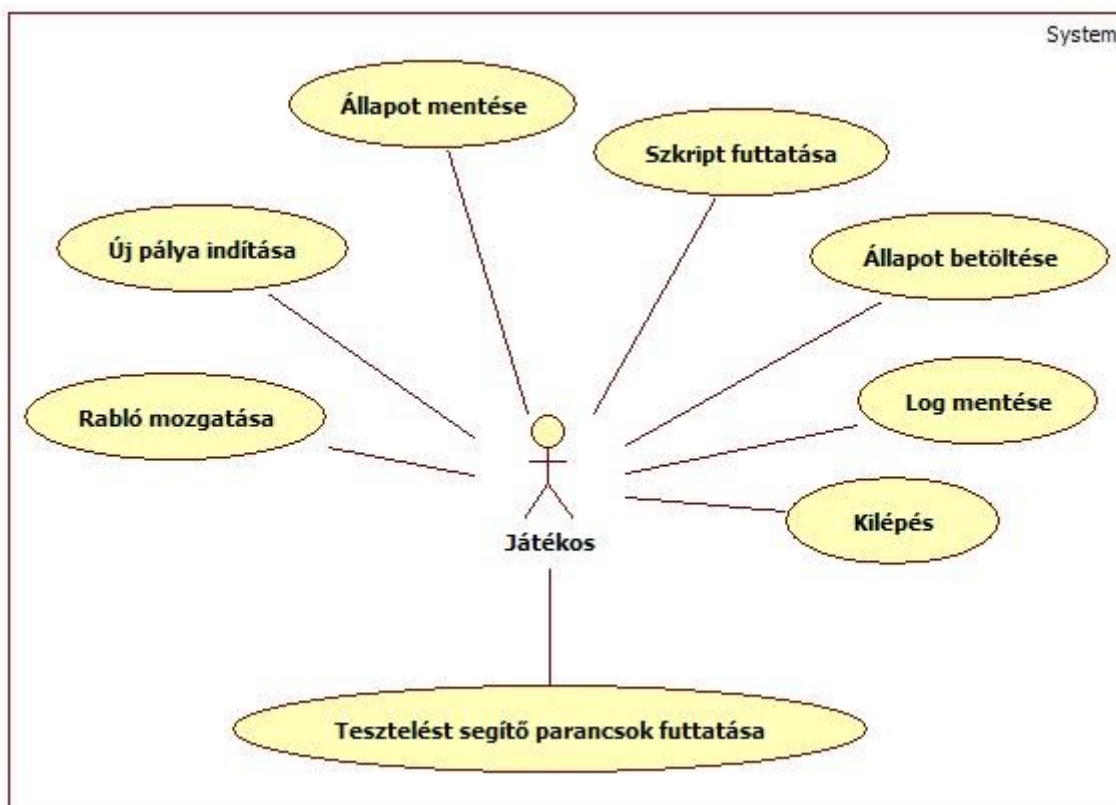
```
INPUT: save state.txt log.txt  
OUTPUT: Files saved: state.txt, log.txt
```

Példa az objektumok állapotának mentésére szolgáló fájl tartalmára:

```
OBJECT      car1  TYPE  Car  
Attr1       value1  
Attr2       value2
```

Ahol **OBJECT** után lévő string a példány neve, **TYPE** után pedig van a típus, amiből visszatöltésnél példányosítanunk kell. Az attribútumok ezek alatt vannak felsorolásszerűen, egymástól újsorral elválasztva, attribútum neve és attribútum értéke pedig tabulátorral elválasztva.

7.2 Összes részletes use-case



1. ábra: A végleges program részletes use-case diagramja

Use-case neve	Tesztelést segítő parancsok futtatása
Rövid leírás	A játékos képes futtatni a tesztelési célú parancsokat is. A végleges játékprogramban ezeket hívják „cheat”-eknek.
Aktorok	Játékos
Forgatókönyv	A játékos begépel a specifikált parancsok valamelyikét.

Use-case neve	Rabló mozgatása
Rövid leírás	A játékos képes a rabló autóját mozgatni.
Aktorok	Játékos
Forgatókönyv	A játékos betölt egy pályát, vagy egy állapotot, majd azon irányítja a rablót.

Use-case neve	Új pálya indítása
Rövid leírás	A játékos képes teljesen új, fájlból betöltött pályán kezdeni a játékot.
Aktorok	Játékos
Forgatókönyv	A játékos a program indítása után betölt egy pályaleíró fájlt, mire a játék elindul. A játékos irányíthatja a rablót.

Use-case neve	Állapot mentése
Rövid leírás	A játékos kimentheti a program aktuális állapotát.
Aktorok	Játékos
Forgatókönyv	A játékos a megfelelő parancs kiadásával kimenti a program aktuális állapotát.

Use-case neve	Szkript futtatása
Rövid leírás	A játékos képes szkripteket futtatni, a tesztelés automatizálása végett.
Aktorok	Játékos
Forgatókönyv	A játékos betölt egy szkriptfájlt a program indítása után, melyet a program végrehajt.

Use-case neve	Állapot betöltése
Rövid leírás	A játékos betölthet egy korábban kimentett programállapotot fájlból.
Aktorok	Játékos
Forgatókönyv	A játékos a megfelelő parancs kiadásával betölt egy korábban elmentett programállapotot, majd innen folytatja a szimulációt.

Use-case neve	Log mentése
Rövid leírás	A játékos diagnosztikai célból kimentheti a program által küldött és fogadott üzeneteket egy fájlba.
Aktorok	Játékos
Forgatókönyv	A játékos a megfelelő parancs kiadásával kimenti a (prototípus esetén képernyőn megjelenő) információkat.

Use-case neve	Kilépés
Rövid leírás	A játékos befejezheti a program futtatását.
Aktorok	Játékos
Forgatókönyv	A játékos a megfelelő parancs kiadásával megszakítja a program futását, és kilép.

7.3 Tesztelési terv

Teszt-eset neve	Inicializálás
Rövid leírás	Pálya betöltésének ellenőrzése
Teszt célja	Ellenőrizni, hogy a pálya betöltésével kapcsolatos folyamatok megfelelően zajlanak le, és a kapcsolódó objektumok attribútumai megfelelő értéket vesznek-e fel. Tesztelni kívánt osztályok: City, Building, Bank, Hideout . <u>Forgatókönyv:</u> Játék indítása, pálya betöltése, kilépés

Teszt-eset neve	Rabló irányítása
Rövid leírás	Rabló irányításának, játék megnyerése.
Teszt célja	Rabló irányíthatóságának tesztelése, sebesség, irány módosítása, rabló elvezetése a rejtékhelyig. Tesztelni kívánt osztályok: City, Robber, Car, Building, Hideout, Roadblock. <u>Forgatókönyv:</u> Játék indítása, pálya betöltése, rabló létrehozása, rabló irányításának sorozata, rejtékhely elérése, játék megnyerése, kilépés

Teszt-eset neve	Rendőrkapja a rablót
Rövid leírás	Rendőrk a rabló mögé kerül, és pár lépés után elkapja.
Teszt célja	Rendőrk elkapási képességének tesztelése. A rendőrk a rabló mögé kerül, és lépések meghatározott sorozata után elkapja. Tesztelni kívánt osztályok: City, Police, Car, Robber, Roadblock. <u>Forgatókönyv:</u> Játék indítása, pálya betöltése, rabló, rendőrk létrehozása, rabló irányításának sorozata, rendőrk elkapja, játék vége.

Teszt-eset neve	Ütközés elkerülés
Rövid leírás	Szemlélteti, mi történik, ha két autó utoléri egymást, illetve ha rabló ér utol autót.
Teszt célja	Tesztelni, az autók valóban nem ütköznek-e egymásnak, azaz az ütközés elkerülés megfelelően működik-e. Tesztelni kívánt osztályok: City, Car, Robber, RoadBlock, ITraffic, TrafficSign, TrafficTable. <u>Forgatókönyv:</u> játék indítása, pálya betöltése, autók elhelyezése, autók mozgatása, hogy egyik hátulról gyorsabban menjen az előtte lévőnél. Következő lépésben pedig úgy mozgatni, hogy a rabló tudjon előzni.

Teszt-eset neve	Nyuszi elütés
Rövid leírás	Rablót üldözi a rendőrk, rabló áthajt a nyuszin, a rendőrk egy ideig nem tudja elkapni..
Teszt célja	Rabló immunisságát ellenőrizni Nyuszi elütés esetén. Tesztelni kívánt osztályok: City, Car, Robber, Police, Bunny, Roadblock, Traffictable, TrafficSign <u>Forgatókönyv:</u> játék indítása, pálya betöltése, rendőrk, rabló, nyuszi elhelyezése, rablónak utasítássorozat megadása, áthajt a nyuszin, rabló mozgatása, játék vége.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A tesztelés során a program kimentett állapotát a várható teszteredményekkel Microsoft Excel programmal hasonlítjuk össze.

A program képes szövegfájlokat beolvasni, például ha az adatokat újsor és tabulátor jel választja el. Beépített függvényei megkönnyítik a tesztelést, fejlett grafikus felülete pedig könnyen átláthatóvá teszi az adatokat, segít a hibákat azonosítani, például feltételes formázással kiemelhetjük a nem várt eredménynek megfelelő adatokat, összesíthetjük a hibás sorokat, esetleg diagramot készíthetünk a hibák előfordulásának okairól.

7.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2010.03.22 21:00	0,5 óra	Boros Molnár Takács Rapp	Értekezlet: Döntés: Rapp elkészíti a Tesztelési tervet, a Tesztelést segítő program specifikálását, és a kimeneti nyelv meghatározását. Takács elkészíti a Prototípus interfész definícióját a Kimeneti nyelv kivételével, valamint a részletes Use-Case-eket.
2010.03.23 17:00	1,5 óra	Rapp	Elkészíti a Kimeneti nyelvet, és a Tesztelést támogató programok specifikációját.
2010.03.24 19:00	3,5 óra	Rapp	Elkészíti a Tesztelési tervet.
2010.03.24 19:00	3 óra	Takács	Elkészíti a Prototípus interfész definícióját a Kimeneti nyelv kivételével.
2010.03.24 22:00	2 óra	Takács	Elkészíti a részletes Use-Case-eket.
2010.03.25 10:00	0,5 óra	Takács	A dokumentum ellenőrzése, helyesírási hibák javítása, formázás.