



A repülőmodellezés iskolája

A CRRCSim kitűnő példája, mintegy állatorvosi lova a szabad szoftvereknek: élvezetes, egyszerűségében is célratörő, sokak tudását koncentráloan professzionális, de a kezdők is könnyedén módosíthatják. Ráadásul komoly anyagi hasznot is hozhat, mert ha időben találkozik vele a modellezőújonc, kevesebb drága pénzen vett valódi repülőmodell bánja a tanulóéveket...

Miért is a repülőmodellezés?

Talán sok kortársamhoz hasonlóan gyerekkorom óta éltem bennem a vágy, hogy – ha már én nem tudok repülni – legalább érdeemben irányíthassak egy repülő tárgyat (ne csak a tárgyért a falhoz). Manapság már kibővíthető áron elérhetőek rádióirányítású (rc, azaz *radio controlled*) repülőgép- és helikoptermodellek, a 10-12 ezer forintos kínai kétpropellerestől kezdve a hatjegyű árat is elérő német és francia gépekig. Ha aztán valaki beleszeret a repülőmodellezésbe, ez a kedvtelés nem kíméli a pénztárcáját, főleg az első időkben, amíg egyik modelltörés a másikat követi. Ha nem valamiféle baráti társasággal vagy egyesülettel kezdjük el a modellezést, hanem egyszerűen csak bemegyünk egy modellboltba, akkor ott a megvásárolt gép mellé ajánlanak ugyan a szerencsétlen (de feltehetőleg vastag pénztárcájú) vásárlónak drága kereskedelmi tanulóprogramokat – az az információ azonban gyakran nem

kerül felszínre, hogy szabad szoftver is született e célra, mégpedig nem is akármilyen minőségű.

A repülőmodell-szimulátorok fontosabb vonásai

Nem repülőszimulátorra kell gondolni, hanem kifejezetten repülőmodell-szimulátorra, ahol a program (és a modellek) tervezői azt a célt tűzték ki maguk elé, hogy a valósághű repülőmodellviselkedést tudjanak produkálni. Mindezt ráadásul többféle modellel, többféle módon paraméterezett „légtérben”, többféle helyszínen. Ami a legboldogítóbb az egészben: azzal az irányítóeszközzel lehet mindezt tenni, amivel a tényleges repülőmodell is lehet vezérelni. Természetesen nem a rádiójelek segítségével, hanem egy kábel révén, amit akár interneten is meg lehet rendelni (magam is így jutottam hozzá fillérekért), vagy el is lehet készíteni megfelelő kapcsolási rajz alapján. Ez a vezérlési élmény egészen más, mint amikor a billentyűkkel vagy valami lövöldözésre kitenyészett bot-

kormánnyal irányítunk egy játékprogramot. (Természetesen van lehetőség arra is, hogy ezen eszközöket használjuk a szemügyre vett szimulátorprogramban, sőt a *Logitech Dualaction* botkormányra külön héjprogrammal lehet előállítani az indító paramétereket.) *Microsoft Windows* operációs rendszerre is van szabad szoftver e célra, az *FMS (Flying Model Simulator)*, de az inkább a játékprogram kategóriájába tartozik (habár azt is lehet ilyen módon irányítani). Ezen újság hasábjain inkább egy másiktól essen szó, amely talán kevesebb helyszínen van felkészítve, viszont az általa használt nemlineáris mozgásszimuláció (☞ http://www.szrfk.hu/konf2005/cikkek/bauer_peter_nem_lin.pdf) és a modellek pontos leképezése elképesztően valósághű állapototteret produkál, és így inkább fizikai szimulációs programnak tekinthető. Ennek neve:

CRRCSim

A program leginkább vitorlázórepülőkre van kihegyezve, de a motoros



1. ábra Egy valódi „Ranger” repülőmodell



2. ábra A modellezés modellezése – eltávolított rádióadó-kristállyal, kábellel

modellek is valóságghűen használhatóak. Alapja a *LARCSim*-en, a NASA Langley Research Center simulator-a. Ennek repülésdinamikai modellje lett átdolgozva, melybe az *XFOIL* aerodinamikai tervezőprogram (szabad szoftver) szerzője, Mark Drela (MIT; <http://web.mit.edu/aeroastro/news/magazine/aeroastro-no3/2006drela.html>) belevitte élvonalbeli nemlineáris mozgásszimulációját. A grafikai részeket Jan Kinsky készítette (jelenleg is ő a *SourceForge*-os fejlesztőhely projektvezetője). Maga a program neve a *Charles River Radio Controllers* repülőmodellező-klubtól ered, mely *Massachusetts*-ben tömöríti tagjait. E klub repülőtere van megmintázva a programban is.

Előkészületek

Keressünk rá a *Google*-ban a *crrcsim* szóra. Érdekes az ismerkedést a *CRRCSim Wiki* oldalán kezdeni

– az a leginkább naprakész. A programot (és az esetleges kiegészítő repülőket – *crrcsim-addon-models*) a *SourceForge*-ról tölthetjük le. E cikk írásakor a 0.9.6-os változat a legfrissebb. Elérhetőek a *CVS* fejlesztői változatok is, de eddig nem volt szükségem rájuk. Letöltés és kicsomagolás után egy puritán *make* paranccsal le lehet fordítani a programot, feltéve, hogy van a gépünkön *SDL 1.2.x* (<http://www.libsdl.org>), *PortAudio 18.1* (<http://www.portaudio.org>), *GLUT 3.7.x* (<http://www.opengl.org/resources/libraries/glut.html>) és *PLIB 1.8.3* (<http://plib.sourceforge.net/>). Ha a disztribúciókban megvannak e csomagok, akkor valószínűleg a *-dev(el)* változatra lesz szükség. (Megjegyzem, hogy a *documentation/compile.txt* szerint

```
make -f Makefile.linux
```

a hivatalos fordítóparancs, de nekem működött a fenti egyszerű módon is.) Fordítás közben kiderül, milyen komoly matematikai és fizikai háttérrel dolgozik a program:

```
...
--- Compiling math/vector3.cpp
--- Compiling math/matrix33.cpp
--- Compiling math/
↳ quaternion.cpp
--- Compiling math/ratelim.cpp
--- Compiling math/
↳ linearreg.cpp
```

```
--- Compiling physics/eom.cpp
--- Compiling math/intgr.cpp
--- Compiling crrc_sound.cpp
...
```

Mivel notebook-omon nincs soros kápu, csak **USB**, kénytelen voltam kicsit átváltoztatni a programkódot egy

```
perl -pi -w -e 's/ttyS3/
↳ ttyUSB0/g' $(grep -ril
↳ "ttyS3" *|grep cpp)
```

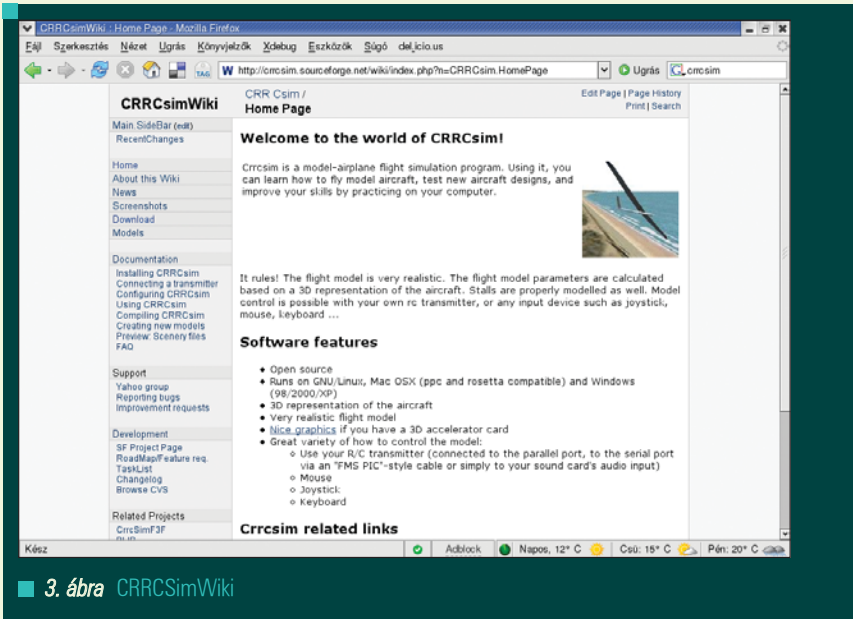
paranccsal. (Egy önéletrajzi adalék: néhány évvel ezelőtt ez volt az első olyan program, amit nem én írtam, de aminek a forráskódjához hozzányúltam, és működött a változtatás. Nyilván nem piszkálja meg az ember egy *OpenOffice* forrását, de egy ilyen program szinte arra termett, hogy kísérletezzünk vele.) Ha már előzőleg egyszer lefordítottuk mindenféle változtatás nélkül is a programot (ami melegen ajánlott), akkor a fenti *PERL* parancs kiadása után csak a módosított fájlok kerülnek feldolgozásra, azaz csak ennyit fogunk látni:

```
--- Deps: GUI/crrc_ctrlgen.cpp
--- Deps: interface_serpic/
↳ tx_interface_serpic.cpp
--- Compiling interface_serpic/
↳ tx_interface_serpic.cpp
--- Compiling GUI/
↳ crrc_ctrlgen.cpp
--- Linking
```

Ezek után (remélhetőleg) ott vár bennünket a *crrcsim* bináris fájl a programkód gyökérkönyvtárában. (*UHU-Linux 2.0*-ra nem sikerült lefordítanom a hiányzó *freelut* függőség miatt – viszont *UHU-Linux 1.2*-n remekül ment.) Érdekes kiadni egy

```
./crrcsim -h
```

parancsot a használható paraméterek feltérképezéséhez. Igen beszédes a kimenet, sok mindent megtudunk belőle, pl. hogy van-e már saját konfigurációs fájlunk (*~/crrcsim*), és hogy milyen szél- és termik-viszonyok között fog futni a program. Előfordult velem, hogy – ha nem sikerült valamilyen (többnyire az irányítóeszközre vonatkozó) beállítást eszközölni – törölnöm kellett a *~/crrcsim* könyvtáram tartalmát, és utána már engedett tevékenykedni.



3. ábra CRRCsimWiki



4. ábra Modellek a CRRCsim-ben

Kezdeti beállítások

Érdekes terminálból indítani a programot, mert – bár lehetőség van indítóikon használatára is – a háttérben nagyon sok hasznos információ megjelenik (angolul):

Loading default videomode from

↳ config...

Using the following rendering

↳ mode:

```

Renderer: Mesa DRI
↳ Intel(R) 830M 20040528
x86/MMX/SSE
Vendor: Tungsten
↳ Graphics, Inc
GL version: 1.2 Mesa 6.1
RGBA bpp: 8/8/8/8
Depth bpp: 24
Stencil bpp: 8
    
```

Opened audio device: 16 bit

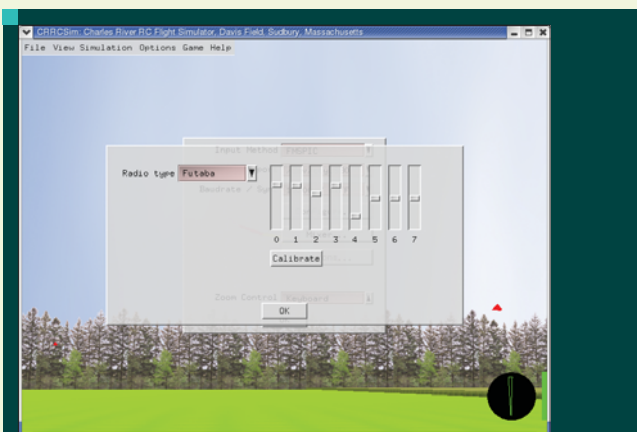
↳ signed audio (mono) at

```

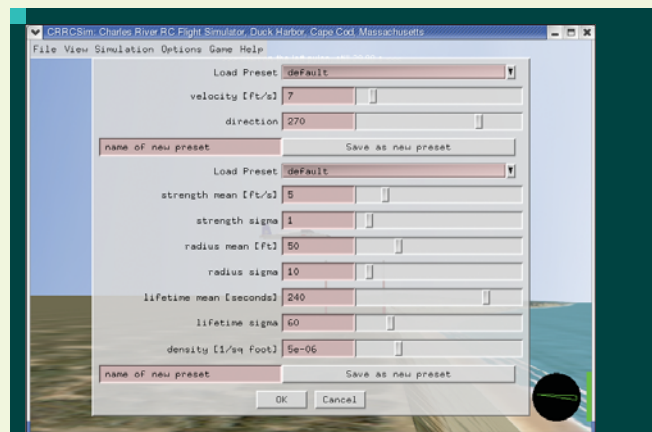
↳ 48000 Hz
Initialized sine samples: 480
Initializing scenery #1
Options chosen:
...
trimmedFlightVelocity =
↳ 15.8775 m/s
--- automatic power
↳ configuration: input -----
-----
<automagic F="12.232605"
↳ v="15.877516">
<battery
↳ throttle_min="0.2">
<automagic T="420" />
<shaft J="0"
↳ brake="0">
<propeller
↳ D="0.24384"
↳ H="0.170688" J="0"
↳ n_fold="-1" />
<engine>
    
```

```

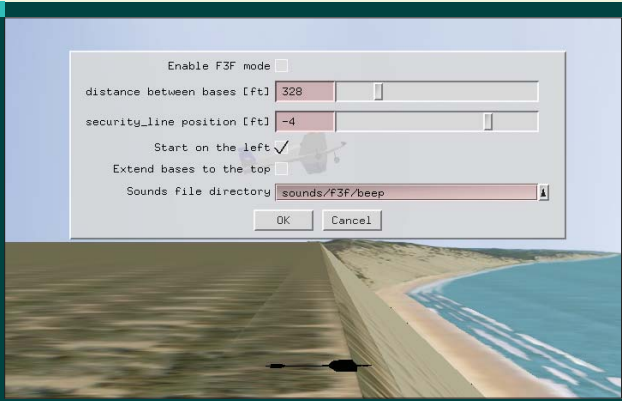
<automagic
↳ omega_p="2827" eta_opt="0.78"
↳ eta="0.7" />
</engine>
</shaft>
</battery>
</automagic>
--- automatic power
↳ configuration: output -----
-----
<battery throttle_min="0.2"
↳ C="4.7812973" u_0="18.379029"
↳ U_off="1" R_I="0">
<shaft J="0" brake="0">
<propeller D="0.24384"
↳ H="0.170688" J="0"
↳ n_fold="-1" />
<engine k_M=
↳ "0.006176186" R_I=
↳ "0.11138591" J_M=
↳ "5.7451026e-05">
<gearing eta="0.95"
↳ J="0" i="1.8801685" />
</engine>
</shaft>
<U_0rel>
1.00;
1.00;
</U_0rel>
    
```



5. ábra Az irányítóeszköz kalibrálása. Piros háromszög: termik



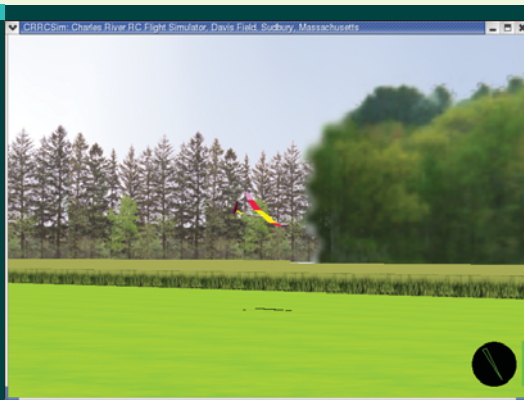
6. ábra Szél- és termik-viszonyok (wind, thermals)



7. ábra Az F3F verseny adatainak beállításai



8. ábra Póznák az F3F versenyben



9. ábra Allegro repülőgép a Davis-i repülőtéren, búzatáblával



10. ábra Zagi repülőgép a frekvenciajelző póznával

```
</battery>
--- automatic power
-> configuration: end -----
-----
```

Esc gombbal kérhetjük a menü megjelenését. Itt az *Options* menüpont *Control* alpontját kell először kérni, hogy az irányítóeszközt megadjuk. Ha valódi repülőmodellhez tartozó eszközzel van dolgunk, akkor egyrészt a típusa beállítható (nálam: *Futaba*), másrészt annak jelszintjét kalibrálnunk kell (5. ábra). Érdemes kivenni a (frekvenciát meghatározó) kristályt a szerkezetből, hogy minimalizáljuk a rádiószennyezést (hátha a környéken valaki épp modellezni szeretne). Egyébként valódi repülőtereken ki is szokott lenni táblázva, hogy ki milyen frekvencián próbálkozik, nehogy egymás gépeit kezdjék el „irányítani”. Ezen kívül a fizikai környezet paramétereit (6. ábra), az indítás (*launch*) erőviszonyait is megadhatjuk, illetve repülőgépek és színhelyek közül is választhatunk, valamint a kép- és hangbeállításokat is megváltoztathatjuk.

A vezérlőgombokat is megtudhatjuk a *documentation/control.txt* fájlból (pl. a legfontosabbak: *r* az újraindításhoz, *q* a kilépéshez kell, és ami valahogy kimaradt a dokumentációból: *d*-vel közelről is megnézhetünk egy-egy modellt, közben ki is próbálhatjuk a vezérlőgombok hatását).

Indulhatunk!

Lehetőség van valódi *F3F* póznakerülőgét versenyt is szimulálni (7-8. ábra). Megvallom, nekem még egyszer sem sikerült szintidőre teljesíteni a verseny követelményeit – van tehát még mit gyakorolnom. Érdemes kipróbálni többféle repülőgépet, kísérletezni a termiek kihasználásával (melyeket a „*training mode*”-ban meg is lehet jeleníttetni piros háromszögekkel és koncentrikus körökkel). Ami talán az első nagy meglepetés a repülőmodellek irányításakor, az az, hogy „a repülőnek mindig ugyanarra van a jobbra”. Vagyis, ha szembe jön a gép, és készülsz elbillenni „nekem balra”, akkor hiába mozdítom a kormányt az

ellenkező irányba (jobbra), az igazából a repülő számára balra van, azaz még inkább el fog billenni. Emiatt van befestve a repülő szárnyra pirosra/kékre (akár be is lehet ezeket jelölni az irányítóeszközön is), hogy inkább ehhez igazodjunk, mint a jobb-bal érzetünkhöz. Egy idő után ki fog alakulni ez a „relatív jobb-bal” érzék is, egészen biztos. Sok örömet e program használatához – vezessen ez sokakat ahhoz az elhatározáshoz, hogy egy igazi repülőmodellt is elkezdjenek használni, mégpedig – a szabad szoftverek erejéből – minél biztosabb gyakorlattal!



Szabó Zoltán
(szz@freemail.hu)
Négy gyermekével és feleségével Pannonthalmán él. Tíz éve kísérletezik a Linuxszal. Matematikát és informatikát tanít, diákkotthonban keseríti a rábízottak életét. Szívégye a PHP, a PostgreSQL és a Moodle.