

## Adjunk életet programozással!

Íme néhány megunthatatlan játék a programozással most ismerkedők számára.

**A** programozók olyanok, mint az istenek, François. Kezdetben semmijük sincs, csak egy gondolat a fejükben, átírnak néhány bitet vagy bajtot, és voilà! A káoszból az élet egy új formáját hozzák létre a Linux-rendszeren. Ez valóban bámulatos, mon ami. Igen, természetesen igazad van, François. Ez kissé félelmetes is. A hatalom felelősség! Ám létrehozni valamit a semmiből nemcsak félelmetes, de mámorító érzés is. A mámorról jut eszembe, hogy mindjárt megérkeznek a vendégeink. A toszkánai megfelelő lesz mára. Még mindig nem indultál el a pincébe az 1997-es *Vino Nobile di Montepulciano*óért? Siess, François, rögtön betoppannak! Már itt is vannak! Bon soir, mes amis! François éppen a pincében van, hogy felhozza a mai alkalomra szánt bort. Mielőtt megérkeztetek, éppen a digitális teremtés örömeiről, a programozásról társalogtunk, és ez egyben mai közös beszélgetésünk témája.

A Linux, amelyet az étteremben és otthonainkban használunk, azoknak a tehetséges és lelkes programozóknak az alkotásaiból áll, akik úgy határoztak, hogy munkájukkal hozzájárulnak ennek a közösségnek az építéséhez. Alkotásaik néha a kibernetikai lét bizonyos fajtáinak – robotoknak és egyéb mesterséges lényeknek – adnak életet. Ha bármikor felmerült bennetek a gondolat, hogy csatlakoznátok ennek az alkotócsoportnak a munkájához, a mai menü nektek szól. Á, François, merci, tölts, légy szíves, a vendégeinknek! Ne sajnáld tőlük! A játékok képezik az új dolgok megtanulásának egyik módszerét, ennek eszköze pedig a Java lehet. Az IBM Robocode nevű programozási környezetét kínálja, amellyel képernyőnk küzdőterén csatázó Java-robotokat hozhatunk létre. Minden csata több fordulóból áll, és az eredményről kimutatás is készül, hogy kedvenc robotunk pillanatnyi állását nyomon követhessük. Ez az adat segít abban, hogy a rengeteg állítható kapcsoló variálásával még hatékonyabbá tegyük a robotot. Vajon, ha a robot beállított száz lépés helyett minden harmincadik lépésnél megáll és megfordul, ez javítja a siker esélyét? Kiderül, ha beál-

lítjuk a kapcsolót, lefordítjuk, és kipróbáljuk a programot. Remek kezdési lehetőség azok számára, akik most ismerkednek a Java nyelvvel – és nem könnyű elszakadni tőle.

A Robocode-dal való munka elkezdéséhez először a Java fejlesztői készletre lesz szükségünk. Ezt ingyen beszerezhetjük a <http://java.sun.com/j2se> címről, csak a megfelelő csomagot kell letöltenünk. Én rendszerem az RPM-csomagot használtam, de más Linux-rendszerekhez GNU tar csomag is rendelkezésre áll. A telepítéssel kapcsolatban talán érdemes figyelmeztetnem titeket arra, hogy a bináris állományok nem a szabványos *bin*-könyvtárakba kerülnek.

A folytatáshoz a PATH változót az alábbi módon meg kellett változtatnom:

```
export PATH=$PATH:/usr/java
➔ /jdk.1.3.1_02/jre/bin
```

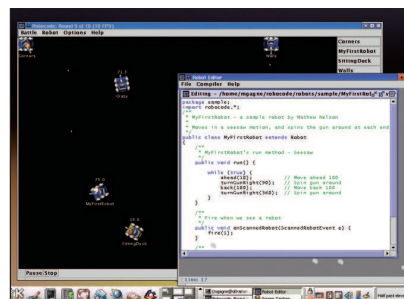
Ha a Java telepítésén sikeresen túljutottunk, az IBM alphaworks honlapjáról le kell töltenünk a Robocode csomagját, amelyet a

<http://robocode.alphaworks.ibm.com/home/home.html> címen érhetünk el.

A telepítés folyamatát fájldokumentumok fogjuk találni, mivel programfordításra egyáltalán nincs szükség. Egyszerűen a Java segítségével a telepítőprogramot az alábbi paranccsal kell futtatni: `java -jar robocode-setup.jar` Az első dolog, amibe beleütközünk, egy ablak, ami a felhasználási szerződés elfogadását kéri. A telepítési könyvtárra vonatkozó javaslat követi – én elfogadtam az alapértelmezett értéket, ami a saját könyvtáramban létrehozott *robocode* nevű alkönyvtár. A telepítés befejezésével a program a következő paranccsal indítható:

```
cd $HOME/robocode ./robocode.sh
```

A program elindítása után a Robocode harcmező jelenik meg, itt vívják csatáikat a robottankok. Válasszuk ki a *Battle* (csata) menü *New* (új) menüpontját. Megjelenik egy oldal, amelyen a következő csatában résztvevő robotokat választhatjuk ki. Mindegyik neve előtt a „sample” (minta) előtag szerepel, és bizonyos előre beállított tulajdonságokkal bírnak. Kezddésként válasszuk például a *sample.MyFirstRobot* és a



1. kép Robotok szerkesztése Robocode-ban

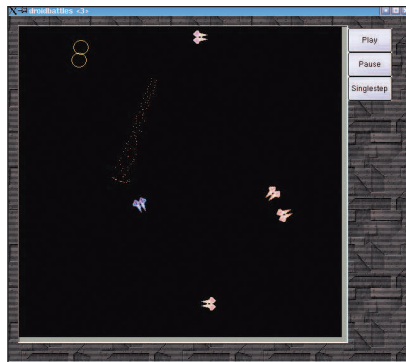
*sample.Spinbot* nevű robotokat. Mielőtt a *Start battle* (a csata kezdése) gomb megnyomásával továbbmennénk, válasszuk ki a *Battlefield* (harcmező) fület. Saját 1024×768-as asztalomom egy szerény 800×600-as csatateret választottam. Végül a *Rules* (szabályok) fülön kell a követendő szabályokat beállítanunk, például a fegyver hűtési gyorsaságát és a tétlenség idejét. Ezeknél járjunk el körültekintően. Kezdetnek elfogadjuk az alapbeállított értékeket is. Készen állunk? Allons, mes amis! Nyomjuk meg a *Start battle* gombot.

Kaphatunk olyan robotot, mint a *Sitting Duck* (ücsörgő kacsca), amelyik csak ül és várja, hogy a csata szele elsodorja. Választhatjuk a *Crazy* (bolondos) nevűt is, amely látszólag véletlenszerűen mozog, és nagyrészt rendszertelenül adja le a lövéseit. Mindegyik betölthető, és viselkedésüket az eredetileg beprogramozott beállításoktól különböző tulajdonságok megadásával módosíthatjuk. Amennyiben – hozzám hasonlóan – te sem vagy megelégedve a csata eredményével, itt az ideje egy kis kódolásnak és robotunk módosításának. Az első alkalommal, amikor úgy döntünk, hogy robotunkat módosítjuk, a Robocode üzemképessé teszi Jikes-ot, a fordítóprogramot (és elküld minket, hogy igyunk egy csésze kávé). Oui, mes amis, én is éppen ezt a kérdést tettem fel magamnak. Milyen idősnek kell lennie egy programozónak, míg az igazi Javával próbálkozhat? Az *első képen* a robotszerkesztő és egy folyamatban lévő játék egy-egy képét láthatjuk. A tanulás leg-egyszerűbb módját a példarobotokkal való kísérletezgetés jelenti, ha ebben

már otthonosak vagyunk, saját példányunk elkészítését is megkockáztathatjuk. A fordítóprogram üzembe helyezésével a Robocode teljes fejlesztői környezete is rendelkezésünkre áll. Kattintunk a **File** (fájl) menü **Open** (megnyitás) menüpontjára, és válasszuk ki a „sample” fájlt. Egy új listát kapunk, amelyben az összes mintarobot (mint például a



2. kép A DroidBattle irányítópultja



3. kép A DroidBattle harcmezeje

Ramfire) szerepel. Minden robothoz tartozó kód meglehetősen jó leírással bír, így működési elvét még egy kezdő programozó is hamar megértheti. Például mit tegyen a robotunk akkor, ha falnak ütközik:

```
/**
 * onHitWall: A falnak
 * tk z0s kezel0se.
 */
public void
onHitWall(HitWallEvent e)
{
    // Pattan0s!

reverseDirection();
}
```

A robotok osztályának különböző eljárásai a `snap to`, `catch on to`, `ahead()`, `back()`, `fire()`, `fireBullet()`, `doNothing()` és így tovább. Az API leírását a telepítési könyvtárban lévő [javadoc/index.html](http://www.bluefire.nu/droidbattles/index.html) néven találhatjuk meg. François, ha lennél kedves vendégeink poharait újratölteni... Szeretnék veletek

megosztani még egy hasonló fejlesztői környezetet.

**Andreas Agorander DroidBattles** programja a Robocode-éhoz hasonló alapelvekre épül, bár programozási nyelvként nem Javát, hanem egy assemblyhez hasonló nyelvet használ. Ezen az alacsony szinten olyan közel kerülünk programunk lelkéhez, amennyire csak lehetséges. A program maga a 2.X Qt programozói könyvtárra épül, így azoknak, akik a KDE 2.X-változatát használják, nem lehet gondjuk a program lefordításával. Mielőtt még azonban, mes amis, hozzákezdenének a fordításhoz, először el kell jutnia hozzátok, illetve nektek kell eljutnotok hozzá.

A legfrissebb forráskód a

➔ <http://www.bluefire.nu/droidbattles/index.html> oldalon érhető el.

A telepítés közbeni egyetlen furcsaság az volt, hogy az archívumban lévő állományok dátuma jövőbeli értékre lett állítva, ami némi bánatot okozott nekem, amikor eljutottam a `./configure` lépéshez.

```
tar -xzvf droidbattles-
1.0.4.tar.gz
cd droidbattles-1.0.4
./configure
make
make install
```

Ezt a programot az teszi elragadóvá, hogy a robotot a kódban is ugyanúgy építhetjük fel, mint a valóságban: processzorokat, memóriát, motorokat, plazmafegyvereket és optikai letapogatókat adhatunk hozzá. Amikor a `droidbattles & parancs` beírásával elindítjuk a programot, a 2. képen láthatóhoz hasonló irányítópultot kapunk. A csatához legalább egy robotot létre kell hoznunk és be kell programoznunk. Az ablak felső részén található **Bot-creator** (robot létrehozása) gomb megnyomásával tehetjük meg az első lépést gépkatonánk megalkotása felé.

Bevásárlólistához hasonló felsorolást kapunk robotunk fizikai tulajdonságairól, valamint egy ablakot a kód számára. Azok számára, akik korábban még soha nem láttak assemblert, a forráskód egy kicsit furcsának tűnik majd, de az érzés hamar el fog múlni. Ezeket az utasításokat közvetlenül a processzor kapja – természetesen nem a gépünkben lévő, hanem a robotunkban elhelyezett. Az utasítások egyszerűek, ráadásul a DroidBattles-szel együtt érkező teljes leírás is a rendelkezésünkre áll, egyszerűen csak a **Documentation** gombra kell kattintanunk. Mellesleg ha továbbhaladva egyre több alkatrészt adunk a robothoz, a rendszer hagyja, hogy lefordítsuk, futtatni azonban már nem enged. Vajon miért? – me-

rül fel a jogos kérdés. A beállítászerkesztő adja meg azokat a kapcsolókat, amelyekhez tartanunk kell magunkat, ha robotot hozunk létre. Például megadjuk az eszközök vagy a memória legnagyobb mennyiségét, valamint ezeknek az egységeknek az árát. Továbbá egy robot csak meghatározott pénzmennyiségbe kerülhet, és minden egyes eszköz pénzbe kerül (ne aggódjatok, mes amis, nem valódi pénzről van szó). Az árat az adott eszköz hatékonysági foka határozza meg. A plazmafegyverek ára 50 és 3500 dollár között mozoghat. Ezek az értékek teszik lehetővé, hogy a különböző jellegzetességeket meghatározott keretek között variálhassuk. Vajon javítja-e az esélyeinket egy csatában, robotunk a kétszer annyi plazmafegyverrel, de csak feleakkora számítási teljesítménnyel rendelkezik? Állítsuk be, mentsük a robot változtatásait, és próbáljuk ki!

Végül, ha mindezzel készen vagyunk, kattintsunk a **Tests** (tesztek) menüre a Bot-creatorban, és válasszuk a **Quick battle** menüpontot. Legalább két robotot kell a csatához kiválasztanunk, de akár ugyanahhoz a fajtához is tartozhatnak. A csatamező megjelenik (lásd a 3. képet), és már indul is a játék. Kattintsunk a **Play-re** (játék), és figyeljük a mókát! A csata folyamata közben a **Pause** (szünet) gombra kattintva lépésenkénti üzemmódra kapcsolhatunk át. Próbaüzemmódban megjelenik egy hibakereső ablak, amelyben a memória lefoglalásáról, a regiszterek tartalmáról és az éppen futó utasításról kaphatunk tájékoztatást. Mindez rengeteg érdekességet és mulatságot rejt magában. Andreas honlapját is megtekinthetjük, amint épp egy DroidBattle-kiszolgálót helyez üzembe. Nos, mes amis, attól tartok, ma nagyon gyorsan letelt az időnk. Az egyik legnehezebb kérdés, amellyel szembesülnünk kell, amikor a programozáshoz hasonló útnak vágnunk neki, hogy mit is szeretnénk létrehozni. A Robocode és a DroidBattles két olyan program, amely keretet ad ehhez az elinduláshoz, és egy olyan küzdelemben enged részt venni, amelyben a kódolás inkább játéknak, semmint munkának tűnik. Bon appétit!

Linux Journal 2002. június, 98. szám



Marcel Gagné  
([mggagne@salmar.com](mailto:mggagne@salmar.com))  
Mississaugaiban (Ontario, Kanada) él, a Salmar Consulting Inc. cég elnöke.