



Linuxszal a tudományos múzeumokért

Használjuk tudásunkat arra a nemes célra, hogy segítsük az újító, szórakoztatva oktató kiállításokat az új közönség megszerzésében.

Az elmúlt évek során fiammal szabadidőnk jelentős részét arra áldoztuk, hogy a New York állambéli Itchacában lévő Sciencenter számára a kiállítást bemutató programot készítettünk. Ez a cikk azt mutatja be, hogy mennyiben hasonlít ez a projekt egyéb programfejlesztési projektekhez, miben mutat eltérést, és mik voltak a Linux használatából adódó előnyök. Nemcsak a Linuxról, illetve a programról lesz szó, hanem azokról a fejlesztési folyamatokról is, amelyekkel egy ehhez hasonló környezetben való munka során találkozhatunk.

A célközönség

A tudományos múzeumot bemutató program tervezésekor a jellegzetes programfelhasználóktól némileg különböző közönségre kellett felkészülnünk, mivel a program más módon kerül felhasználásra. Korábbi, Microsoft Windows alapú kiállítóprogramjaim – a Measurement Factory és a Fabulous Features – a körülbelül hatéves gyerekekkel záródó korosztályt célozta meg, míg a Linux alapú kiállítások – a Sound Studio és a Traffic Jam – közönsége az ötven-hatvanévesek közül került ki. A programnak elég egyszerűnek kellett lennie ahhoz, hogy a célközönség megértse a képernyőn látottakat, és hogy a látvány miképpen kapcsolódik a kiállításhoz. Egy kiállításon a számítógép nem maga a kiállítás, útmutatóként vagy egyszerűen manipulálható eszközként szerepel a többi tárgy között. Számítanunk kell egy másodlagos „közönségre” is, ez pedig a múzeum személyzete. A teremben dolgozókat (akiknek jelentős része önkéntes) nem lehet minden egyes számítógép hóbortjaira felkészíteni. A kiállítás gépeinek és programjainak önműködően kell elindulniuk, amikor reggel működésbe hozzák a múzeumi rendszert, mint ahogy az intézménynek legalább néhány árammegszakító lekapcsolásáról is gondoskodnia kell az esti záráskor. Nyilván az ember azt szeretné, hogy a személyzet szeresse a kiállítást, hiszen ha megnehezítjük az életüket, akár az „átalakítás miatt zárva” táblát is kiakasztgatjuk az ajtóra.



1. kép A Traffic Jam a Montshire Museum of Science-ben (Montshire Tudományos Múzeum – Vermont, Norwich)

A verseny

Álmaink kiállításának tervezetésekor a legfontosabb dolog, aminek állandóan a szemünk előtt kell lebegnie, az az, hogy amikor a fiatalabb látogatókat megcélozva, jól látható módon elhelyezünk egy számítógépet a múzeumban, versenyre kell kelniük az összes játékkal, amit valaha számítógépen játszottak, és az összes televíziós vagy mozifilmmel, amit addig láttak. Kiállításunknak ezért messzemenőig látványosnak, figyelemfelkeltőnek kell lennie. A Measurement Factory teljesíti ezt az elvárást, azáltal, hogy a látogatók megmérhetik a testúlyukat, a magasságukat, a szorításuk erősségét, összehasonlíthatják az eredményeiket kortársaik eredményeivel, és befejezésként egy bizonyítványt is kapnak minderről. A Traffic Jam (1–3. kép) szintén friss és szórakoztató, hiszen a látogatók játszhatnak a közlekedési lámpákkal, és elkerülhetik a forgalmi dugókat – vagy előidézhetik, ha épp ehhez van kedvük. A Sound Studio (4–5. kép) a látogatók számára lehetővé teszi, hogy saját és barátaik hangját többsávos hangrögzítőre rögzítsék, majd visszajátszáskor olyan egyszerű hatásokkal lássák el,



2. kép Erős forgalom a Traffic Jamben



3. kép A közlekedési lámpák beállítása a Traffic Jamben

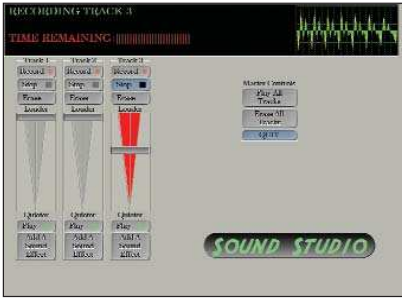


4. kép A Sound Studio

mint amilyen a visszhang. Mindennek nyilván egyértelmű határai vannak. Például nem túl jó ötlet a „Hibáztál!” jelzést (hangot, képet, animációt) túl látványosra tervezni, különben a látogatók ezt tekintik majd céljuknak – soha ne akadályozzuk a kiállítás céljainak megvalósulását!

A fejlesztési folyamat

A célközönség és a nevelési célok szempontjából fontos a pontos fejlesztési folyamat kidolgozása. A következőkben



5. kép Hangfelvétel a Sound Studioval

ismertetett szempontok némelyike egészen hétköznapiak tűnhet, mégis fontos a betartásuk. Saját fejlesztési ciklusunknak természetesen nem kell pontosan megegyeznie az itt ismertetendő folyamattal.

- **Az indító megbeszélés:** ha egy olyan különálló kiállítást szándékozunk létrehozni, ami nem része egy nagyobb, szervezett bemutatónak, akkor ez a szakasz akár el is maradhat. A Tech City például körülbelül egy tucat olyan egymással összefüggő kiállításból állt, amelyek a technika és a mérnöki tudományok központi témája köré szerveződtek. Ezért egy egész napos indító megbeszélést tartottak a projekt céljainak megvitatására és egy közös kezdeti nézőpont kialakítására. Több ilyen megbeszélést részt vettem már, mindegyik bőven megérte a rá fordított időt. Amellett, hogy ráhangolódunk a projektre, hallhatunk egy-két jó ötletet, és találkozhatunk olyan emberekkel, akik segítségünkre lehetnek a folyamat későbbi szakaszaiban.
- **Öletbörze (brainstorming):** egyrészt tedd a lábad az asztalra, és ne törődj még a Linuxszal sem, amíg eszedbe nem jut néhány igazán jó ötlet. Másrésztől, ha tudomásunk van egy már létező alkalmazásról, amit esetleg egy kicsit módosítva hasznossá tehetnénk, említjük meg, és készítsünk jegyzeteket. Mialatt körbejárjuk a felmerülő ötleteket, tartsunk kéznél egy listát a projekt elvárásairól. Ezek lehetnek egészen egyszerűek, mint „egy műszaki tudomány bemutatása”, de kicsit bonyolultabbak is, például „szemléltessük a Heisenberg-féle bizonytalansági elvet általános iskolások számára egyszerű grafikus módszerrel, virtuális beavatkozási lehetőségekkel”. Mindig tartsuk szem előtt az elvárásokat, bár garantálhatom, hogy ezek az utolsó pillanatokig változni fognak.
- **Prototípus készítése:** ez a szakasz megegyezik más területek megfelelő

1. lista Az állandóan futó program indításának pseudokódja

A CD-ROM befűzésének kísérlete a techcity saját könyvtárában a merevlemez sajátkönyvtára felett. Nincs szükség a visszatérési kód ellenőrzésére. A "reboot monitor" folyamat elindítása. Belépés "techcity" néven és a startx futtatása. Ha a CD befűzése sikeres:

A .xinitrc leállítja a reboot monitort, letiltja a képernyőkímélőt és az energiatakarékos üzemmódot, elindítja a programot.

különben:

A .xinitrc lefuttat egy Python-parancsfájlt, ami a felhasználót a program-CD behelyezésére kéri, majd vár, amíg a felhasználó a "done" gombra kattint. Ez utasítja a reboot monitort a rendszer újraindítására. A CD-ROM leválasztása.

2. lista Az S99xx-mytechcity fájl, ami a /etc/rc2.d fájlba lett beillesztve a program önműködő újraindulása érdekében

```
mount -o ro,user,exec /dev/cdrom /home/techcity
/root/rebootmon &
su - techcity -c "cd /home/techcity;startx"
umount /dev/cdrom
```

3. lista A Sound Studio .xinitrc fájlja

```
echo -n "Q" >/tmp/rebootfifo

xset s noblank
xset s off
xset -dpms

cd SoundStudio-newmeter/src
exec ./soundst
```

szakaszaival, azzal a különbséggel, hogy esetünkben több szórakozást rejt magában, azoknak az embereknek köszönhetően, akikkel együtt dolgozunk. A múzeumok munkatársai, akikkel volt szerencsém együttműködni az évek során, kivétel nélkül értelmes, alkotóerőben gazdag, a munkájuk iránt lelkesedő emberek voltak. Ne döbentsen meg miniket túlzottan a kiállításához esetleg szükséges pavilon prototípusának a látványa. Valószínűleg igen sok kábelcsatornára és kartonlemeze lesz majd szükség.

- **Az első ellenőrzés:** ez az alkalom, amikor a prototípusunkat először tesszük ki bíráltnak. Ne aggódjunk amiatt, ha a folyamatban érintett felnőttek nem bírálják olyan hatásosnak az alkotásunkat, mint amilyenek mi tartjuk. Ne feledjük, hogy a felnőttek egészen másképp

gondolkodnak, mint a gyerekek. Minden kötöttség nélkül mutassuk meg a prototípusunkat néhány, a célközönség korában lévő gyerekeknek.

- **Célteszt:** talán ez a kiállítás megalkotásának a legszórakoztatóbb része. Ez az első (és esetleg a második, harmadik) alkalom, hogy az alkotást a látogatók számára kipróbálásra üzembe helyezzük, így megfigyelhetők a válaszreakciók. Várhatóan többször is végrehajtjuk ezt a folyamatszakszot. Néhány fontos kérdés, ami megfontolandó, illetve válaszra érdemes: vonzó a kiállítás a látogatók számára? Milyen hosszú időt töltenek a helyszínen a látogatók? Előfordul, hogy több látogató együtt használja a gépet? Megtanulják a látogatók, amit megpróbálunk nekik megtanítani? A megfelelő korosztályt szólítja meg a program?

4. lista A járműleíró rész a Traffic Jam beállításfájljának dokumentumtípus-meghatározó állományából

```
<!ELEMENT vehicles ((vehicle)+,colors)>
<!ATTLIST vehicles numtypes NMTOKEN #REQUIRED>
<!ELEMENT vehicle EMPTY>
<!ATTLIST vehicle length          NMTOKEN          #REQUIRED
                    maxspeed       NMTOKEN          #REQUIRED
                    accel           NMTOKEN          #REQUIRED
                    slowdistance    NMTOKEN          #REQUIRED
                    stopdistance    NMTOKEN          #REQUIRED
                    >
<!ELEMENT colors (color)+>
<!ATTLIST colors num NMTOKEN #REQUIRED>
<!ELEMENT color EMPTY>
<!ATTLIST color red          NMTOKEN          #REQUIRED
                    green     NMTOKEN          #REQUIRED
                    blue       NMTOKEN          #REQUIRED
                    >
```

Esetleg lehetőségünk nyílik elbeszélgetni azokkal a látogatókkal, akik kipróbálták a bemutatót, vagy ezt a múzeumi személyzet is megteheti, és tájékoztatást a benyomásairól. Talán ez a legértékesebb útmutató, amit kapni fogunk; a gyerekek szerencsére kegyetlenül őszinték.

- **Szakmai kiértékelés:** léteznek olyan cégek (USA), amelyek megfelelő szakértelemmel rendelkeznek oktató vagy múzeumi programok kiértékeléséhez. Ha a kiállításunk a Nemzeti Tudományos Alapítvány (National Science Foundation, NSF) valamely tagjának támogatásával létesül, akkor valószínűleg közülük is értékelni fogja valaki. Nekünk ezek közül csak eggyel, az Inverness Research Associa-tésszel volt alkalmunk kapcsolatban állni, de az észrevételeiket különösen hasznosnak találtuk. Ha lehetőségünk van rá, fogadjuk is meg ezeket a tanácsokat, sokat tanulhatunk belőlük.

- **Átdolgozás, beleértve a programkód letisztítását:** a prototípuskészítéshez hasonlóan ez a lépés is olyan, mint más fejlesztési folyamatok esetén az ennek megfelelő szakasz. Érdemes ilyenkor időt szánnunk a megjegyzések kibővítésére, a zavaros kód kigyomlálására.

A múzeumi programnak egyre több mindennel kell foglalkoznia. Ha valamikor a jövőben a program módosítása szükséges lesz, az a többletidő, amit most ezekre a dolgokra fordítunk, bőven megtérül. Ez a megfelelő alkalom arra is, hogy a kiállítás nem számítógépes részének is megépítsük a végleges változatát.

- **Az üzembe helyezés:** ez a szakasz megint csak hasonló az egyéb projektek megfelelő szakaszához, bár jó észben tartani, hogy az ilyen típusú projektek-

nél fontos elvárás, hogy a rendszer folyamatosan működésre kész legyen, és ne igényeljen karbantartást. Ha a programunk egy nagyobb rendezvény részét fogja alkotni, ne feledjük, hogy a személyzet ilyenkor rettentő elfoglalt, ezért tegyük a lehető legegyszerűbbé a teendőiket. A Tech City rendszerének üzembe helyezése végül is nem állt másból, mint a gépek üzembe helyezéséből – négy ponton plusz egy tartalék –, a Linuxnak minden gépre történő feltelepítéséből és egy korábban előkészített CD-ROM-mal való tesztelésből (erről a Tippek és trükkök részben lesz még szó).

A gépek kiválasztása

Előfordulhat, hogy a programunknak egy már nem használt, régi gépen kell futnia, de az is megeshet, hogy a legkorszerűbb gépen, ami éppen a projekt számára nyújtott adomány része, vagy ha szerencsések vagyunk – legalábbis a Linux meghajtóprogramjainak a szemszögéből nézve –, kaphatunk egy olyan gépet, ami egy nemzedékkel idősebb. Ha abban a helyzetben vagyunk, hogy meghatározhatjuk a szükséges gép kiépítését, az elsődleges szempont a megbízhatóság legyen. A megfelelően kipróbált és kiforrott alkatrészek valószínűleg fontosabbak lesznek, mint a legújabb, leggyorsabb műszaki csodák, és forró pillanatoktól menthet meg egy valamivel lassabb processzor és videokártya, amikor hat hónap elteltével a távoli múzeumban egy meleg nyári napon nem következik be az üzemzavar. A rendszer hűtésével nem szabad takarékoskodni. Ha tehetjük, jó minőségű ventilátorokat válasszunk, és bárki is építi az esetleges

kiállítási pavilont, biztosítson megfelelő légáramlatot a számítógépnek, és gondoskodjon a levegő elvezetéséről is. A Tech City programja adományként érkezett Hewlett-Packard Vectra 400-as gépeken lett üzembe helyezve, amelyek 1 GHz-es Pentium III-as processzossal, i810-es lapkakészlettel, és 20 GB merevlemezzel voltak felszerelve. Mindegyik gépet egy-egy Sound Blaster 16 PCI hangkártyával bővítettük ki, mivel a Sound Studio futtatása igényelte a teljesen kétirányú (full duplex) hangkártyát, vagy kettő olyat, ami nem rendelkezik ezzel a lehetőséggel. Mindegyik rendszer 19"-os monitorokkal volt felszerelve, szintén a Hewlett-Packard jóvoltából.

Az eszközök és a programkönyvtárak kiválasztása

Mivel nem ismerem az adott projektet, nem javasolhatok adott eszközöket és programkönyvtárakat. Arra sem teszek kísérletet, hogy valamilyen programozási nyelvet ajánljak, ellenben adhatok néhány olyan útmutatót, ami a munkánk során jó szolgálatot tett.

Tapasztaltabb fejlesztőknek biztosan nem lesz új, amit mondom, de ha egy már létező programcsomag közel áll ahhoz, amit az adott feladat esetén meg kellene valósítanunk, gondoljuk meg, hogy nem érdemesebb-e inkább azt módosítani és alkalmazni. A Traffic Jam felhasználói felülete négy részre osztotta az ablakot (a forgalom megjelenítése, a sűrűség beállítása, irányítás, egyéb adatok).

Választásunk a kiterjedt témalehetőségek okán a GTK+-ra és az icewm ablakkezelőre esett, bár az utóbbit később kis mértékben módosítanunk kellett.

Természetesen a kölcsönözni kívánt kódra vonatkozó felhasználói szerződést figyelmesen el kell olvasnunk. Tartsuk tiszteltetben a szerző jogait, és ne hártítsunk a múzeumra semmilyen kötelezettséget. Ahogy a gép kiválasztásánál, itt is az a legjobb, ha kellően körültekintőek vagyunk. Ha biztosak vagyunk benne, hogy nem lesz szükségünk a segédprogram vagy könyvtár legfrissebb változatának szolgáltatásaira, ne siessünk a telepítésével – a már ismert hibákat mindig könnyebb kezelni, mint a rejtetteket.

Ne feledkezzünk meg a karbantartásról és a projekt élettartamáról, hiszen vagy nekünk, vagy valaki másnak esetleg a jövőben módosítania kell majd a programot. A programozási nyelv vagy könyvtár egy még nem kibocsátott változata üzembiztosnak és jól működőnek tűnhet az adott pillanatban, de két év múlva esetleg teljesen más lesz vagy valamilyen

5. lista A járműleíró rész a Traffic Jam beállításfájlijából

```
<vehicles numtypes="3">
  <vehicle length="10" maxspeed="10" accel="4"
    slowdistance="20" stopdistance="8"/>
  <vehicle length="15" maxspeed="10" accel="2"
    slowdistance="30" stopdistance="8"/>
  <vehicle length="20" maxspeed="10" accel="1"
    slowdistance="40" stopdistance="8"/>
  <colors num="9">
    <color red="255" green="204" blue="51"/>
    <color red="204" green="153" blue="255"/>
    <color red="255" green="204" blue="204"/>
    <color red="255" green="153" blue="204"/>
    <color red="153" green="255" blue="204"/>
    <color red="204" green="204" blue="153"/>
    <color red="255" green="153" blue="51"/>
    <color red="153" green="204" blue="204"/>
    <color red="0" green="255" blue="255"/>
  </colors>
</vehicles>
```

szempontból nem működik együtt. Még a g++ is változott az alatt a négy év alatt, ami a Traffic Jam prototípusának elkészítése és a kész program üzembe helyezése között eltelt.

Végül kényelmetlen helyzetbe hozva magam még azt is bátorkodom javasolni, hogy a Linux-rendszercsomag választásakor is tartsuk magunkat bevált óvatosságunkhoz. A legújabb és legcsillogóbb változat alkalmas lehet asztali gépünkre, de a megbízhatóság ebben az esetben sokkalta fontosabb. Végző rendszerünkhöz a mi választásunk a Debian 3.0r0 változatra esett, ugyan nem sokkal a kibocsátása után, de mivel a Debian híres a visszafogottságáról, nem nyugtalanokdtunk a döntés miatt.

Tipppek és trükkök

Szeretnék megosztani néhány, a munkánk során felmerült nehézséget és az erre alkalmazott megoldási módszert. Lehet, hogy nem találtuk meg a lehetséges legjobb megoldásokat, de a mi esetünkben jól beváltak.

FELADAT: *Hogyan tarthatnánk a rendszerünket mindig működésre kész állapotban? A kiállítási programnak a gép bekapcsolása után, felhasználói beavatkozás nélkül el kell indulnia. Szükség esetén a múzeumi személynak képesnek kell lennie a program frissítésére.*

MEGOLDÁS: A mi esetünkben ez az volt, hogy a CD-ROM-on lévő alkalmazáskönyvtárat a kiállítás saját könyvtárában (például `/home/techcity`) legfelső szintjére fűztük be, és indításkor önműködően a megfelelő felhasználói

néven léptünk be. Ha nincs megfelelő CD-nk – minden lemez csak egy kiállítás programját tartalmazza –, a képernyőn megjelenik a felszólító üzenet, hogy tegyük be a megfelelő lemezt, és indítjuk újra a gépet. (Bár a látogatók számára nem hozzáférhető, de a billentyűzet minden számítógéppel ott van a pavilonban.) Az újraindítást figyelő program egy FIFO kimenetét figyel, hogy az R billentyű lenyomása esetén újraindítsa a gépet, vagy Q esetén lépjen ki, bár más megoldáson is gondolkoztunk. Az 1. lista a folyamat pszeudokódos leírása, a 2. listán az önműködő újraindítást lehetővé tevő fájl látható, a 3. lista pedig példát mutat a `.xinitrc` fájlra.

FELADAT: *Az ilyen típusú programok általában igénylik a rendszer finomhangolását. Hogyan valósítsuk ezt meg?*

MEGOLDÁS: Első gondolatunk egy, a Windows `.ini` fájljához hasonló formátumú beállításfájl alkalmazása volt. Ez működött volna a Sound Studio esetében, de a Traffic Jamnál már nem. Az utóbbi többek között igényelte volna a különböző járműtípusok meghatározását, ami az XML képességeivel könnyen ábrázolhatóvá tette volna az azonos osztályba tartozó különböző egyedeket. A fiam egy olyan C-könyvtárat írt, ami az `xmlib2` felett futva lehetővé teszi, hogy az egyes alkotóelemekre faszervezetként – vagyis az alkotóelem dokumentumbeli elérési útvonalával – hivatkozzunk. A 4. lista a Traffic Jam beállításfájlijának dokumentumtípus-meghatározó állományából (DTD) mutat egy szakaszt, az 5. listán pedig a fájl hivatkozott része

látható. A 6. lista (49. CD Magazin/Museum könyvtár) a jármű fizikai tulajdonságait betöltő kódreszletet mutatja – látható, ahogy a Cfg a korábban említett könyvtáreljárásokkal körbevett C++-objektumra hivatkozik.

FELADAT: *hogyan kezeljük az ablakkezelőt biztonsági szempontból? A látogatóknak nem szabad, hogy ki tudjanak lépni a programból, vagy más programokat tudjanak elindítani, hogy mozgatni tudják az ablakokat és így tovább.*

MEGOLDÁS: A korai tesztek során azt tapasztaltuk, hogyha a látogatók számára megengedjük a billentyűzethez való hozzáférést, előbb-utóbb megtalálták a módját, hogy kilépjenek a programból. Ezt a beállításfájlok és az icewm módosításának segítségével úgy oldottuk meg, hogy ne legyenek felbukkanó menük, és tiltva legyen az ablakok mozgatása, illetve átméretezése. Másrészt egyik program sem igényli a billentyűzet használatát, normál üzemelés közben az a pavilon belsejében maradhat.

Összegzés

Lehet, hogy nem az oktatóprogramok írása jelenti számunkra vágyaink netovábbját, de ha mégis így lenne, próbáljunk egy olyan műszaki múzeumot találni, amelynek segíthetünk. Szeretik az új arcokat, mindig szívesen fogadják a segítséget, és azt is garantálhatom, hogy többet kapunk vissza, mint amennyit mi adunk.

Szeretnék köszönetet mondani a New York állambeli ithacai Sciencenter munkatársainak és önkénteseinek egy óriási múzeum működtetéséért, és hogy az embereket gondolkodásra serkentik. Látogassunk el a honlapjukra a <http://www.sciencenter.org> címen.

A fotók tulajdonosa a Sciencenter (Ithaca, New York).

A cikkhez kapcsolódó listák megtalálhatóak a 49. CD Magazin/Museum könyvtárában.

Linux Journal 2003. június, 110. szám



Len Kaplan

(lkaplan@nlzero.com)

Akkoriban kezdett programozni, amikor a kisszámítógépek még hűtőszekrény nagyságúak voltak. A Linux

mellett nagyon érdeklik a beágyazott rendszerek, szabad idejében szívesen foglalkozik vasútmodellezéssel.