

LOUD OBJECTS
Live soldering session, © fotó: Samu Bence

A hiba egy programozó számára kihívást jelent, hiszen fel kell lelnie annak forrását, majd úgy kell kreatívan átalakítania a munkáját, hogy a programja zavartalanul működhessen. A megszokottól eltérően (hibásan) működő eszköz egy művész számára szintén kihívás lehet, a nem várt események alapjaiban változtatják meg a dialektust a létrehozandó mű, a közönség és közte.

Ha technikai értelemben ERIC S. RAYMOND szabad szoftverfejlesztésről írt esszéje, *A katedrális és a bazar*⁹ lenne a FLOSS mozgalom alpmunkája, akkor művészeti, esztétikai aspektusból nézve minden bizonnyal *Eco Nyitott műve* lenne a párja.

Élő kód

SIMON YUILL szerint¹⁰ az egyik legeklatánsabb FLOSS jelenség a „live coding”, vagyis élő kódolás műfaja. Az alkotók az előadás során „élőben” írják, majd újraírják a kódot, amely zenei, illetve képi vagy más tartalmakat állít elő. A kódot mindeközben láthatóvá teszik a közönség számára is, így a létrehozott rendszer transzparenssé válik, a végeredményen túl érzékelhetőek a háttérben meghúzódó, mélyebb szintű folyamatok is.

Azon túl, hogy a kód megosztása a maga transzparenciájával, kollaboratív megközelítésével filozófiai álláspontot képvisel, a mű esztétikáját is aktívan formálja. Egy élőben kódolt performansz során általában az első pár perc csendben telik, mivel az előadók a „nullából építkeznek”, vagyis „tisztalappal indulnak”, amikor elkezdik írni a kódokat. Pár perc szükséges ahhoz, hogy az első hangok, szekvenciák megszólaljanak, hogy megkezdődjön az utazás egy kiélezett minimalizmusból a komplexitás tájai felé az egymást módosító, újra felülíró algoritmusok segítségével.

Több élőben kódoló kollektíva maga készíti az eszközeit. ALEX MCLEAN *feedback.pl* keretrendszere például egy egyszerű Perl¹¹ szkript, amely folyamatosan olvassa és végrehajtja saját magát, miközben néha az előadó is átírja különböző részeit. Ez a folyamatosan alakuló rendszer adja az előadás zenei struktúrájának gerincét,

amelyben a determinisztikus kód és a szabad improvizáció kettőssége egyaránt érvényesül. Az így létrehozott mű nem csak zenei struktúra lehet, DAVID GRIFFITH *fluxus*¹² nevű környezete például Lisp¹³ alapokon manipulál háromdimenziós képi formákat, az említettekhez hasonló módon. TOM SCHOUTEN *PacketForth*¹⁴ rendszere pedig egy másik környezet, amely szintén vizuális tartalmak élő módosítására szolgál. Más alkotók meglévő programnyelveket alkalmaznak, vagy adott esetben módosítják is azokat. Népszerű eszköz például a *SuperCollider*¹⁵, a *Chuck*¹⁶ vagy a *Pure Data*¹⁷. Tulajdonképpen bármely olyan rendszer alkalmas, amely értelmezni tudja saját kódjának változásait, miközben végre is hajtja azt (vagy annak részleteit). Sőt, több esetben az esemény nem is programozás alapú, például az AP kollektíva *life coding* előadásában elektronikai eszközök hackelése (circuit bending), szoftverkódolás és konferencia-szerű beszélt nyelv formál egy egységet.

A szabadon elérhető komponensek a kódot, vagy más, instruáló keretrendszert zárt, befejezett egész helyett nyílt, vég nélküli mutálódó folyamattá alakítják.

Álmodó kódok

A foglalt házak hangulatát idézte a 2008-as *Piksel* helyszíne: az álmos fjord partján fekvő Bergenben, az előző évi, világháborús bunker helyszín után a fesztivál nagy része most a kikötőben, egy volt szardíniagyárban kapott helyet. A *Piksel*-lakók számára délelőtt műhelyek, délután bemutatók, éjjel előadások adták a program fő gerincét.

A *Fritzing*¹⁸ névre hallgató projekt például az elektronikát próbálta meg közvetíteni művészek, tervezők felé. A célja az, hogy a segítségével laikusok tudjanak az elképzeléseiknek megfelelő elektromos berendezéseket tervezni: ha egy elektronikában nem túl járatos ember egy virtuális környezetben (ez a *Fritzing*) összeköt néhány ellenállást, ledet, szenzort vagy hasonló, egyszerű elektronikai elemet, abból a program készíti egy iparilag előállítható (nyomtatható, maratható) áramkört, amelyet aztán a felhasználó integrálni tud munkáiba. A *Fritzing* alkotóinak célkitűzése egy olyan, nyílt forrású keretrendszer kiépítése, mint amilyen a programozásban a *Processing*¹⁹, vagy a szintén elektronikai építést könnyíteni hivatott *Arduino*²⁰.

12 <http://www.pawfal.org/fluxus>

13 A Lisp programozási nyelvet eredetileg rekurzív függvények absztrakt ábrázolására tervezték, de hamarosan a mesterséges intelligencia kutatás előszeretettel alkalmazott nyelvévált, amikor a 70-es, 80-as években ezen terület a virágkorát élte. Ma a Lisp nyelveket számos területen alkalmazták, és közkedvelt a számításmélet oktatásában is. (forrás: wikipedia)

14 <http://zwizwa.be/packetforth>

15 <http://www.audiosynth.com>

16 <http://chuck.cs.princeton.edu>

17 <http://www.puredata.info>

18 <http://www.fritzing.org>

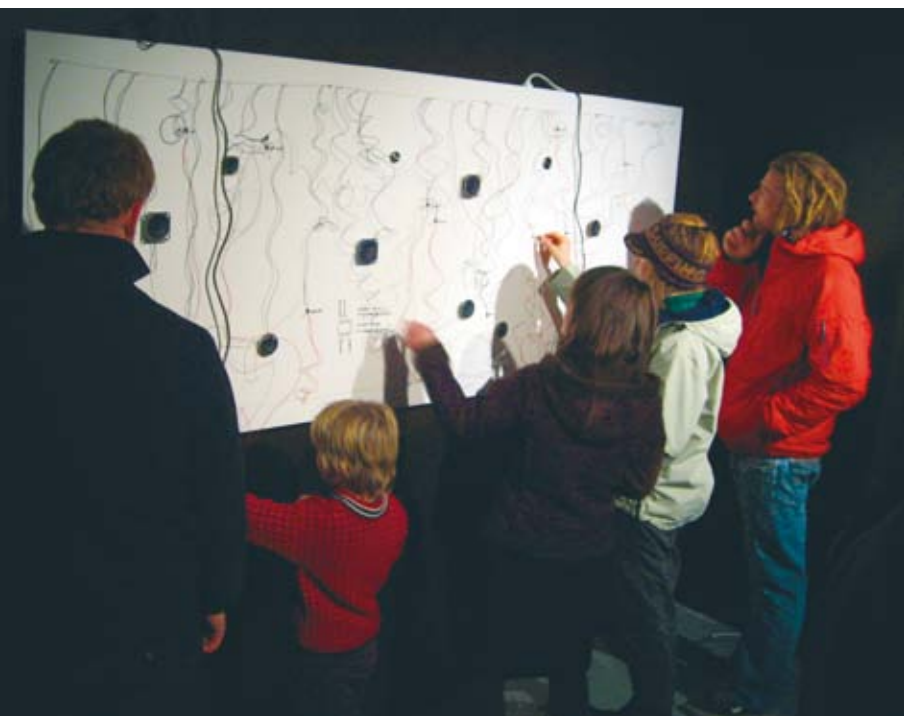
19 <http://www.processing.org>

20 <http://www.arduino.cc>

9 <http://magyar-irodalom.elte.hu/robert/szovegek/bazar>

10 Simon Yuill glasgow-i művész és programozó. Több díjnyertes esszé szerzője a szabad szoftverek és a művészet kapcsolatának témakörében.

11 A Perl egy programozási nyelv, amelynek egyik legfontosabb része a reguláris kifejezések széles körű támogatása és alkalmazása, ezáltal kiválóan alkalmas nagy méretű szöveg- vagy adatfile-ok egyszerű feldolgozására. (forrás: wikipedia)



LOUD OBJECTS
Installáció, fotó: Carlo Prelz

Magyarországról a KITCHEN BUDAPEST médialabor *Animata*²¹ szoftvere mutatkozott be, amely valós idejű bábanimációk készítésére szolgál. A szoftver rétegekbe rendezhető kétdimenziós jeleneteket animál. A betöltött állóképek „becsontozhatóak”, ezáltal szabadon alakítható, hogy egy báb melyik része mennyire flexibilis. Külön érdekessége, hogy más alkalmazásokkal összekapcsolva képes különböző bemeneti eszközökre reagálni, ezáltal könnyen irányíthatóak a karakterek hanggal, fénnel, mozgással, vagy más eseményekkel. A szoftver 2008-ban elnyerte „Az év nyílt forrású szoftvere” díjat Finnországban.

JULIAN OLIVER *Levelhead* projektje három olyan kockából áll, amelyek egy logikai játékot alkotnak. A kiválasztott kocka kamera felé tartott oldalain lévő ábrákat felismeri a rendszer, így a program mindig tudja, hogy épp melyik oldalt választja a játékos. Ezen kívül még azt is látja, hogy milyen arányban döntjük meg a kockát, és egy animációt is rendel az adott oldalhoz, amelyet a játékos egy kivetített képen lát. A kockák forgatásával egy kis figurát kell végigjuttatni egy labirintuson. A valóság (tapintható kocka) és a virtualitás (vetített élőkép, rajta animációval) ilyen harmonikus keveredése nagyon izgalmas memóriajátékhoz vezet, az emlékek és sémák (labirintus helyszínei) egy-egy oldalhoz rendelve a tér egy absztrakt, mentális leképezését adják.

Érdekes volt „élőben” látni IOHANNES ZMOELNIG előadását. Ő a korábban említett „live coding” egyik emblemikus figurája, aki egyrészt fejlesztő, másrészt az IEM (Institut für Elektronische Musik und Akustik) kutatója, aki a hang fiziológiájával és elektroakusztikával foglalkozik. Előadásának címe, amely akár a *Piksel* alcíme is lehetett volna, szabad fordításban valahogy így hangzik: *Álmodnak-e a szinuszoidok elektronikus szűrőkkel?* Zmoeling a – többek közt általa fejlesztett – *Pure Data* névre hallgató vizuális programnyelvet használta. Ebben a nyelvben főként dobozokat kell egymáshoz kötni a képernyőn, hogy a különböző logikai és hang modulok között kapcsolatot létesítsünk. Zmoeling ezt a felhasználó általi összekötözést automatizálta, és a gépre hagyta, hogy autonóm módon kerülhessenek egymással kapcsolatba különböző zenei műveletek, úgy mint a hang-generálás, szűrés, vagy -szekvenszálás. A folyamat során egymásra épülő, majd egymást kioltó, újra felülíró szonikus tartalmak a maguk minimalizmusában érdekes kérdéseket vetettek fel: mi történik, amikor egy olyan rendszer, mely felhasználói interakcióra (egér, billentyűk) hiva-

tott, részben eltávolodik ettől a céltől, más-hogy kezd funkcionálni, mint amire tervezték? Másrészt mi történik, amikor az alkotó meta-utasításokat ad egy félig autonóm folyamatnak, és a hangokat nem „közvetlenül” szó-laltatja meg, hanem csak lehetőséget biztosít azok előfordulására bizonyos kereteken belül? DEREK HOLTZER, az utrecht-i Umatic Media Lab egyik kordinátora ezúttal analóg zajkollázst készített: írásvetítőre helyezett kinetikus tárgyakat (ventillátorokat, fényszűrőket, motorokat), és a vetítő fényét érzékelő fotóellenállásokkal vezérelte analóg szintetizátorát. A harsány, ritmikus hangfolyam mögött a nagyban kivetül mozgó tárgyak sziluettje határozta meg a színpadképet.

Az írásvetítő máshol is előbukkant, mégpedig TRISTAN PERICH és a LOUD OBJECTS kollektíva megnyitó előadásában, a bergeni Lydgallerietben, amely a fesztivál idejére különböző hanginstallációkat válogatott össze. A galéria eredetileg is hangművészetre specializálódott, az év többi időszakában is hasonló munkákat állít ki. A Loud Objects performanszai hasonlóak a „live coding” tevékenységéhez, csak analóg eszközökkel. Ők előadásait „live soldering”-re, vagyis élő forrasztásra keresztelték. A nézők által körülvett vetítő felületén integrált áramköröket, drótokat, ellenállásokat forrasztanak össze, amelyek sziluettje a falra vetülve a jelenlévők számára kiemeli az apró gesztusokat. Ez a megoldás a bensőséges, finom mozgások jellemezte forrasztást egy publikus eseménnyé alakítja. Az akció folyamatosan torzuló, alacsony felbontású szonikus eseményekből építkezik, ahol a kompozíció a vetítőt körülálló, három alkotó aktuális döntéseitől függ.

Talán még érdekesebb és közvetlenebb a tapasztalat, amikor a közönség kerül az ő helyükbe, és saját maga határozhatja meg, mit forraszt hova. A Loud Objects csapata a galéria egyik falán egy több méter széles, hófehér vásznat feszített fel, amelyet drótok, hangszórók, ellenállások garmadája fedett. Ezekhez oda lehetett sétálni, ki lehetett próbálni, mit hova, mivel és hogyan lehet összekapcsolni: szabadon, empirikus módon tapasztalhatóak a nem várt lehetőségek, mintegy gyerekként fedezhetjük fel a hibák mezejét, különböző kombinációk ritmikái, texturális tulajdonságait.

A kollaboratív alkotás kétségtelenül további perspektívákat nyújthat művészetnek és technológiának egyaránt, az ezzel együtt járó elosztott, műhely-központú öntanító módszerekkel együttesen. Hiszen ezek hatékonyak, működőképesek, és feltehetően sok olyan projekt számára szolgálhatnak közös kiindulópontul, amelyek külön-külön nem feltétlenül jönnének létre, csak így, együtt válhatnak valósággá, amikor az egész többé lesz, mint a különálló részek összege.

21 <http://animata.kibu.hu>