

SPECCYALISTA VILÁG

A SPECCYALTA BARÁTI KÖR LAPJA



TV-BASIC különkiadás

ROM cartridge-ek története

Játékújdonságok

Hogyan készült az Ishido

Border és Multicolor effekt

TZX Mouse és JoyTape Player

sCenE esszencia

ZX81 expander

Assembly ovi

Micro-Pokeer

Sinclair Vakóra

ZX81 klasszikusok

Rejtvény



2015. 2. SZÁM

TARTALOM

BEKÖSZÖNTŐ.....	3
ARCKÉPCSARNOK	
INTERJÚ A BÍRÓ TESTVÉREKKEL (1024 STUDIO).....	4
SPECTRUMOLÓGIA	
SINCLAIR VAKÓRA.....	8
A SPECTRUM ROM CARTRIDGE-EK TÖRTÉNETE.....	9
LOAD ""	
JÁTÉKÚJDONSÁGOK.....	12
ZX81 KLASSZIKUSOK - 1. RÉSZ.....	16
PROGRAMOZÁSTECHNIKA	
BASIC: ZX SPECTRUM TV-BASIC KÜLÖNKIADÁS - 1. RÉSZ.....	19
MÉLYVÍZ: BORDER ÉS MULTICOLOR EFFEKT.....	23
ASSEMBLY OVI: HOGYAN ÍRJUNK JÁTÉKOT ZX SPECTRUMRA - 2-4. RÉSZ.....	27
SDK: SPECCYBEEP.....	34
HOGYAN KÉSZÜLT AZ ISHIDO: THE WAY OF STONES - 1. RÉSZ.....	36
SCENÉ ESSZENCIA	
MESCALINE SYNESTHESIA.....	39
NEW VIEW 48K.....	40
HARDVERSIMOGATÓ	
ZX81: ZXPAND - EXPANDER RETRO MÓDRA	41
HUNGARIKUM: TZX MOUSE ÉS JOYTAPE PLAYER	43
HUNGARIKUM: MICRO-POKEER	45
HARDVER ÖTLETEK	
FPGA SPECTRUM - 1. RÉSZ	47
USB BILLENTYŰZET ILLESZTÉSE EXTRÁKKAL - 1. RÉSZ	49
FEKETE DOBOZ	
AVAGY A KŐKORSZAKIK ESETEI	51
OLVASÓI LEVELEK.....	53
REJTVÉNY.....	54

BEKÖSZÖNTŐ

Elérkezett hát az idő a második szám megjelenésére, ami talán még nagyobb mérföldkőnek tekinthető, mert a kezdeti lelkesedés sokszor már nem elég a folytatáshoz is, azonban sikerült ezt is megugranunk, ahogy azt tapasztalhatjátok. Most is igyekeztünk az egykori Spectrum Világ szellemiségét megőrizni és minden olvasónknak adni valami számára érdekes olvasmányt. Mindemellett a Mikulás és Karácsony közeledtével szeretnénk volna egy kis többlet tartalommal is szolgálni, hogy az ünnepek alatt legyen bőséggel Spectrumos téma is terítéken.

Bár a szerkesztőségünk címére ugyan kevés visszajelzés érkezett, ugyanakkor a fórumon és a különféle facebook csoportokban elcsípett visszhangok alapján azért úgy látjuk, hogy az alapkonceptciónk találkozott az olvasóink igényeivel, így ezt a vonalat igyekszünk továbbra is képviselni.

Nagy örömmel köszönhetünk új szerzőket is, ráadásul így újabb állandó rovatoknak és cikksorozatoknak is örülhetünk.

A sCene esszencia rovat kapcsán pedig a portálon korábban kedvelt rovat visszatérésének lehetünk tanúi G.O.D-nak hála, annál is inkább, mert ezt a témát többen is hiányoltátok.

Továbbra is szeretnénk szemezgetni korábban a portálunkon megjelent érdekesebb írásokból is, hogy az immáron leporolt cikkek még több Z80-szívű olvasónkhoz juthassanak el és talán tökéletesebb formát ölthessenek. Így kerülhetett sor most ROM cart-ok története című cikk felfrissítése, melyet jelen esetben a korábban készült, de még eddig meg nem jelentetett Sam. Joe írás, a Sinclair vakóra indukált. Egyelőre még csak reménykedhetünk, hogy a 'Minden hétre egy játék' című állandó portál rovatunk atyja további írásokkal is megörvendeztet majd bennünket a jövőben.

A porolás kapcsán megemlíthetjük a Border és Multi-color effekt című cikket is, bár valójában most egy teljesen új mű született a portálon eddig olvasható mű romjain, mely így közérthetőbben vezet be ezen effektek rejtelseibe és immáron a teljesség igényével készült, melyet persze ismét Pgyuri jegyez, aki emellett a BASIC turpisságairól is fellebbenti a fátylat a TV-BASIC különkiadása című érdekfeszítő sorzatában.

A hardver örülteknek is szeretnénk kedvezni, ezért a berkeinken belül folyó munkákba engedünk egy kis betekintést, amihez most Jocó nyújtott hathatós segítséget.

A Spectrumos témák mellett mindenképp szeretnénk teret engedni a ZX81-gyel foglalkozó írásoknak is, ennek első hírnökeit olvashatjátok is már ebben a számban.

Reméljük a "lap" továbbra is elnyeri tetszésedet és lesznek, akik szívesen bekapcsolódnak a további munkába akár csak egy-egy cikk erejéig. Ugyanakkor mindenképp várjuk véleményeteket, ötleteiteket melyekkel jobbra, érdekesebbé tehetjük a mi kis kiadványunkat.

Ezúton szeretném ismét megköszönni a szerkesztőségünk szerzőinek és munkatársainak azt a sok munkát, melyet befektettek ezen újabb szám elkészítésébe.

Kardos Balázs (Balee)

IMPRESSZUM

Főszerkesztő: Kardos Balázs (Balee) **Felelős szerkesztő:** Egri Imre (Zimi) **Szerkesztés, tördelés, grafika:** Molnár Péter (Mopi)

Rovatvezetők: Buzogány Csaba (Makranc), Mezei Róbert (M/ZX), Tanács Imre (Kapitány), Kardos Balázs (Balee), Lakatos Péter (Latyi.ca), Pgyuri, Taletovics Dávid (G.O.D)

Szerzők: Bicsák Lajos (Asimo), Gondos Csaba (Csaba), László József (FPGA Jocó), Mezei Róbert (M/ZX), Pgyuri, Samu József (Sam. Joe), Taletovics Dávid (G.O.D)

Szerkesztőség e-mail címe: specyalista.vilag.szerkesztoseg@sinclair.hu

Kiadó: Specyalista Baráti Kör <http://sinclair.hu>

2015. december

ARCKÉPCSARNOK

INTERJÚ A BÍRÓ TESTVÉREKKEL (1024 STUDIO)

A Sinclair.hu mindig is fontos feladatának érezte, hogy a felkutassa a hazai "Spectrumos éra" jeles alakjait és emléket állítson akkori tevékenységüknek. De nincs ez másként a határon túli magyarság köréből kikerült fejlesztőkkel, hisz így bepillantást nyerhetünk - legalább egy picit - az akkori erdélyi spectrumos életbe, amelyre gyerekként nemigen volt rálátásunk. Arról nem is beszélve, hogy a kezdeti spectrumos próbálkozások után napjainkban igazi siker-történetbe csöppenhetünk.

Bíró Ádámot és Zoltánt kérdezzük az emlékezetes kolozsvári évekről, akiket az Opus1 című remekbeszabott Spectrumos kottaszerkesztő és lejátszó program kapcsán sikerült felkutatnunk és akik mára a nemzetközi porondon is elismert szoftverfejlesztőkké váltak.



(Korebeli felvétel: Ádám ül, Zoltán áll)

Sinclair.hu: Először is szeretném megköszönni, hogy ilyen lelkesen álltok ezen email interjúhoz. Még az a jó, hogy a Spectrum örök, sosem évül el ☺, hisz már 2005-ben sikerült felvennünk a kapcsolatot és csak most 2015-re jutottunk el a végkifejlethez.

Bíró Zoltán: Okoztál némi meglepetést a leveleddel. Az Opus 1 indítóképernyőjéről mondjuk kiderülhetett, hogy annak idején is 1024 Studio-nak hívtuk magunkat, de akkoriban még mindenféle jogi háttér nélkül műveltük a programozást – egyikünk épp másodéves mérnökhallgató volt, másikunk csak középiskolás diák. Az „igazi” 1024 Studio Kft-t három év múlva alapítottuk és csak jóval később, az évtized vége felé álltunk neki hivatásosan is programozni.

Úgyhogy tartozol még azzal az információval, hogy mi módon sikerült ránk találnod.

Sinclair.hu: Nos 2005-ben elég volt rágugolnom a '1024 studio bíró ádám és zoltán' kombinációkra. Most 2015-ben pedig már az akkor levelezésünkben elhangzott StudioRIP kifejezés segített ismét megtalálni benneteket.

Talán e kis bevezető után kezdjük ott, hogy mikor, hogyan kerültetek kapcsolatba a számítástechnikával, ezen belül is a Sinclair gépekkel?

Bíró Zoltán: A romániai kommunista rendszer – némileg eltérően a magyarországitól – eleve gyanakvóan tekintett minden olyan kezdeményezésre, mely kicsit is a nyugat csillogására emlékeztette az elvtársakat. Másrészt viszont a megalomániás hatalom a hazai személyi számítógépek fejlesztését státuszszimbólumként kezelte, és ennek megfelelőképpen támogatta is. Ezt a tudathasadásos kettősséget úgy oldották fel, hogy mindenféle furcsa elnevezéssel illették ezeket a gépeket, mint pl. „Oktatási Célú Programozható Automata”, „Elektronikus Játékok Televízióra” stb., – nehogy véletlenül elhangozhasson az, hogy „személyi számítógép” – és ügyeltek arra, hogy ilyen készülékek magánszemély tulajdonába ne nagyon kerülhessenek. (A CIP-01 számítógépen is pont egy ilyen lehetetlen felirat látható, ld. a mellékelt fotót.)



Az iskolák és a pionír-(vö. úttörő-)házak viszont relatíve könnyen hozzájuthattak ezekhez az eleinte 100%-ban saját fejlesztésű (Prae, aMic), majd később az ugyancsak belföldi gyártású, de felállaltan Spectrum-kompatibilis házi számítógépekhez. Kijelzőként rendszerint fekete-fehér televíziókat használtak, antennabemenettel, ritka volt, hogy az iskola vagy az úttörőház egyetlen színes tévéjét számítógéphez kössék, igazi monitorokkal pedig csak számítóközpontokban lehetett találkozni akkoriban.

Első komolyabb találkozásunk a számítástechnikával így aligha meglepő módon a pionírházban történt, kezdetben Prae, később a Spectrum kompatibilis HC-85 formájában. Ez utóbbi készülék amúgy érdekes módon fejlettebb volt, mint az eredeti Spectrum, 48K helyett árnyék RAM-os 64K-án futott, ráadásul olyan „tár-bíró” áramkörrel, amelynél képernyőolvasás–memóriafrissítés konfliktus kezelését sokkal ügyesebben oldották meg, mint az eredetiben.



Az első saját számítógépünket 90-ben tudtuk megvásárolni, rögtön a rendszerváltás után; Bukarestig kellett menni érte és három havi fizetésbe került. Természetesen Spectrum kompatibilis gép volt, a televíziógyár által piacra dobott CIP-01, enélkül fel sem tevédhetett volna, hogy megírassuk az OPUS 1-et.

Sinclair.hu: Mennyire volt elterjedt a ZX Spectrum Erdélyben? Volt-e klubélet?

Bíró Ádám/Zoltán: Amint már említettük, szaküzletből vagy más, „hivatalos” forrásból gyakorlatilag lehetetlen volt géphez jutni magánszemélyként, még akkor is, ha valaki hajlandó lett volna ezért az élvezetért akár horribilis összeget is fizetni. Határon átcsempészett készülékből viszont került jó néhány magánkézbe is, természetesen messze nem annyi, ahány Magyarországon. Ezek többnyire Spectrum kompatibilis gépek voltak, Commodore Atari stb. csak elvétve akadt.

Szigorúan vett klubélet fel sem tevédhetett, de a pionírházak szakkörökön lehetett jó dolgokat tanulni, vagy inkább közösen játszani, esetleg játékokat csereberélni, ily módon egész jó „Spectrum baráti Kör”-ök verbuválódhattak össze. Kivételes alkalmakkor az oktató még azt is megengedte, hogy egy éjszakára, hétvégére stb. valaki hazavigye az egyik gépet. Érdekes, hogy magyarországi szaklapokat (Számítástechnika, Chip Mikrovilág stb.) viszont meg lehetett rendelni, így egyoldalúan bár, de tudtunk valamiféle kapcsolatot tartani a magyarországi Spectrumos közösséggel.

Sinclair.hu: Mi volt az első komolyabb próbálkozások?

Bíró Ádám/Zoltán: Egy felerészt BASIC-ben, felerészt gépi kódban megírt evolúciós játék, ahol „alkatrészekből” kellett összerakni egy növényevő állatot, az Evolutaur-t, amely egy virtuális világban próbált túlélni, a forduló végén pedig maga a szoftver pontozta az állat túlélési teljesítményét. Ekkor a játékosnak (levonva a menet tanulságait) lehetősége volt egyvalamit változtatni az állaton (pl. hosszabb nyak), de számítania kellett arra, hogy közben a táplálék is adaptálódott a következő fordulóhoz, pl. az új menetben a program a gyümölcsöket magasabbra helyezte a fákon. Patrubány Miklós közbenjárásának és segítségének köszönhetően sikerült összehozni, hogy a román lemezkiadó vállalat, az Electrecord magnókazetta formájában árulta a játékot, de sajnos a határidő miatt össze kellett csapnunk az egészet, így sikerült elfuserálnunk egy igazán jó ötletet. Ha lenne időnk és agyunk hozzá, lehet, szívesen újraíránk – 2015-ös szinten, természetesen.

Sinclair.hu: Nem maradt fenn az utókornak egy példány? Szívesen archiválnánk ezt is a Magyar fejlesztések közé...

Bíró Ádám/Zoltán: Nekünk sajnos biztosan nem. Elvileg megtörténhet, hogy az Electrecord archívumában még megtalálható, de kétlem, hogy egyszerű procedura lenne utánajárni.

Sinclair.hu: Mindig együtt fejlesztgettetek?

Bíró Ádám/Zoltán: Igen, leszámítva két esetet.

Az első, amikor (a rendszerváltás után) Ádám teljesen önállóan fejlesztett egy, az iskolaújságra szabott kiadványszerkesztő programot. Mai szemmel eléggé primitív volt az egész, de mindössze 128 kilobájtton meg lehetett szerkeszteni az egész újságot, több betűtípussal, magyar és román ékezetes karakterekkel, félautomata szótagolással stb. Képeket, grafikákat sajnos nem támogatott, ezeket utólag rajzolták kézzel vagy fénymásolták az üresen hagyott keretekbe. Annyiban nem volt ez kizárólag Ádám munkája, hogy a szükséges hardveres átalakításokat (bővítés 128K-ra, saját memórialapozási rendszer, nyomtatóinterfész) viszont Zoltán tervezte és ültette gyakorlatba. Ezzel az együttműködéssel amúgy az a különös helyzet alakult ki, hogy így nemcsak a program vált egyedi, konkrét feladatra kifejlesztett célszerszámmá, hanem az őt kiszolgáló hardver is. Kár, hogy sem a szoftver, sem az egyetlen példányban átalakított Spectrum kompatibilis gép nem maradt fenn.

Ezenkívül volt még Ádámnak egy saját projektje egy magyarországi fejlesztői csoporttal is.

Sinclair.hu: Ádám magyarországi projektjéről is mesélnétek pár szót?

Bíró Ádám: A Kolozsvári Műszaki Egyetemen készített szakdolgozatom egy hangelemző és szintetizátor-program volt, ami fizikai hangszerekről rögzített hangminták alapján felépített egy hangszert leíró hangszínfájlt, majd a projekt részeként készített MIDI lejátszó a hangszínfájlok segítségével lejátszott MIDI fájlokat. Amikor egy magyarországi hirdetésben zenei adatfeldolgozásban jártas programozót kerestek, a szakdolgozatom tapasztalatával a hátam mögött belevágtam. A projekt rám eső részének a zenét kellett elemeznie, és a grafikai modul fele a zenét leíró információt szolgáltatnia (ritmus, hangerő stb.), hogy az majd a zenével összhangban adhasson grafikai aláfestést a zenének. Amennyire értettem, az volt az elképzelés, hogy diszkókban kivetítőn fog menni az így generált kép – nem tudom, hogy mennyi valósult meg a projektből, de az én részem működött.

Sinclair.hu: Az akkori munkáitokból csak az Opus 1 maradt fenn? Nem őriztétek meg a kazettáitokat és a gépeteket?

Bíró Ádám/Zoltán: Ennyi idő elteltével valószínűtlen, hogy egy programos kazetta használható állapotban megmaradhatott volna, a gépről pedig pontosan tudjuk, hogy melyik albérletünkben láttuk utoljára. Nem volt könnyű szakasz az életünkben, amikor egyik ideiglenes lakásból a másikba kellett cipelnünk a kacatokat, és akkor még eszünkbe se jutott, hogy egy elavult személyi számítógépnek évek múltán nemhogy erkölcsi, de piaci értéke is lehet. Nagy valószínűséggel egyébként sem volt már működőképes, de a televíziógyár a 80-as évek azóta elfeledett jó szokása szerint világos, érthető kapcsolási rajzot is mellékel a géphez, így nem lett volna különösebb gond megjavítani sem. Ha már memóriabővítés és egyebek formájában amúgy is rendszeresen belepiszkáltunk.

Még rosszabb hír, hogy az Op. 1-nek volt egy belső használatra szánt, mondhatni 1.1-es verziója is, amely MIDI-portos szintetizátorral már polifóniát is tudott a saját gyártású interfészünkön keresztül, de a már meglévő felhasználói felülettel, így eléggé nehézkes volt a használata. Értelemszerűen ez a verzió is csak a mi gépünkön működött. Természetesen ebből a szoftverbővítésből sem maradt fenn példány.

Sinclair.hu: Hogyan jött az Opus no. 1 ötlete? Foglalkoztatok zenével?

Bíró Ádám/Zoltán: Zeneiskolába soha nem jártunk, szakszerű zenei képzésben sem részesült egyikünk sem, de zenélni természetesen szerettünk, Ádám például kifejezetten jól gitározott. A gond a kottaolvasás körül volt – a Kodály-módszeren nevelkedett magyarországinak nehéz lehet elhinni, hogy hangszeren játszóknak vagy akár zene iránt komolyan érdeklődőknek probléma legyen kottát olvasni, de a román zenei oktatás sajnos a jóval nehezebb francia módszert követte. Így nem volt olyan könnyű magánúton kottaolvasást tanulni az akkoriban rendelkezésre álló anyagból. Így jött az ötlet, hogy itt a számítógép, zenei hangot tud kiadni, mi lenne, ha íránk magunknak egy „mankót”, amelyik ügyesen leolvasná helyettünk a kottát. Sajnos az Op. 1-en eléggé látszik, hogy a kottaíráshoz csak alapszinten értő emberek írták (az előjegyzések kezelése például nehézkes és logikátlan), de lehet, hogy ez csak egy plusz érdekessége a programnak.



Sinclair.hu: Az már kiderült számomra, hogy a 1024 Stúdió jogutódja, a StudioRIP ma már komoly vállalkozás lett, de mivel foglalkozik manapság a cégetek?

Bíró Ádám/Zoltán: Az egyetem befejezése után még játszogattunk az ötlettel, hogy PC-re professzionális színvonalú kottaszerkesztő programot írjunk. Már csak azért is, mert egy osztrák kottakiadónak, majd egy ismert kolozsvári zeneszerzőnek tördelt kottákból éltünk akkoriban. Jellemző volt az akkori szoftverek színvonalára, hogy míg Macintosh-on ott volt az eléggé nyakatekert, lehetetlen koncepciójú, de legalább profi igényű, grafikus operációs rendszerre írt Finale, addig PC-n csakis az ósdi, kényelmetlen, DOS(!) alapú Score rúghatott labdába, ha nyomdai színvonalú kották szerkesztéséről volt szó. (A ma népszerű Sibelius vagy a többi felhasználóbarát szerkesztő messze nem volt még sehol.) Elegendően naivak voltunk ahhoz, hogy támogatást próbáljunk szerezni egy merőben új elképzelésre alapuló, sima szövegszerkesztőkkel vetekedő hatékonyságú kottaszerkesztő megírására, de természetesen senki nem állt velünk szóba. Élni viszont kellett valamiből,

így egy időre el is felejtettük a programozást és hosszú évekig kiadványszerkesztéssel (DTP) foglalkoztunk.

A szoftverfejlesztés témája akkor vált igazán égetővé, amikor elérkeztünk arra a szintre, hogy a nyomdai filmek előkészítését is vállalni tudtuk. Akkoriban ritka volt az a nyomda, amelyik bármilyen klisékészítő eszközzel is rendelkezett volna, a méregdrága filmlevilágítók néhány elit DTP cég birtokában voltak, így az egyetlen opció a bérlevilágítás volt – eléggé borsos áron, mert maga a nyersanyag is drága volt, és a befektetésnek is meg kellett térülnie. És ez volt a kisebbik gond, ugyanis az eszközöket kiszolgáló (ugyancsak elképesztően drága) szoftverek – elnézést a szójátékért – echte felhasználóelleneségek voltak. Sok esetben még tisztességes előképet sem generáltak, így elég volt egy kis figyelmetlenség, s sokszor csak több méter film elpocsékolása után derülhetek ki olyan egyszerű dolgok, mint hogy a munka jobb esetben keresztbe fordulva, rosszabb esetben lelógva jelent meg a filmen. A veszteséget ilyenkor mi álltuk, saját zsebünkből. Érelődött már a gondolat, hogy ennyi pénzért mi sokkal rendesebb programot tudnánk írni, de az igazán kedvező alkalom akkor adódott, amikor majdhogynem a szemétdombról megszereztük életünk első levilágítóját. Az elektromechanikai felújítás nem okozott gondot, de semmiféle szoftverünk nem volt a készülékhez. Ha akkoriban lett volna fölösleges 4-5 ezer dollárunk, lehet, hogy a könnyebbik utat választjuk, megvesszük az akkoriban elérhető, kényelmetlen és drága szoftverek valamelyikét, és ma is DTP-vel foglalkoznánk. De nagy szerencsére pont egy vasunk sem volt, ráadásul azonnal leesett, hogy itt a nagy lehetőség, amelyre olyan régóta várunk, hiszen lélekben mindig is programozók voltunk. Kapóra jött a Linuxos világból jól ismert (de Windowson is hozzáférhető) ingyenes GhostScript programcsomag, így magának a PostScript interpreternek a megírásával már nem kellett kínlódnunk.

Ez volt a StudioRIP születésének a pillanata.

Ezenkívül nagy segítségünkre volt az az angliai, használt levilágítókkal foglalkozó kisvállalkozás, amellyel Ádám még a legelején összehaverkodott, és akik – látva a StudioRIP első, még nagyon kezdetleges változat vázlatát – meggyőztek, hogy a DTP-t hanyagolni kell és minden erőnket a szoftver mihamarábbi fejlesztésére kell fordítanunk. Így merész lépésre szántuk el magunkat: a cég átváltozott StudioRIP Kft.-vé, a DTP vonalat fokozatosan elsovasztottuk, majd ejtettük is, és innen tovább minden erőforrásunkat a StudioRIP fejlesztésének szolgálatába állítottuk.

A nyerő lap bejött, azóta a 4.1-es verziónál tartunk és világszerte néhány ezren használják már a programcsomagot, főképp az igen kedvező, „kelet

európai” árak és a felhasználóbarát koncepciónak, illetve a fejlett grafikus interfésznek köszönhetően.



(Manapság egy szakkiállításon: Zoltán baloldalt, Ádám jobboldalt, középen a szoftver perui terjesztője)

Sinclair.hu: Meg tudátok fogalmazni egy mondatban, hogy mit jelentettek számotokra azok az évek?

Bíró Ádám/Zoltán: Túl a soha vissza nem térő fiatalkori évek jellegű közhelyeken: volt egy sajtós, ma már nehezen érthető varázsa annak, hogy mindössze 64 kilobájt RAM és egy csupasz fehér képernyőn villogó kurzor látványa függőséget tudott kialakítani egy számítógép-rajongónál.

Az interjút készítette: Kardos Balázs (Balee)



SPECTRUMOLÓGIA

SINCLAIR VAKÓRA

Egyszer, ha majd nagyon-nagyon sok időm lesz, akkor írok egy hosszabb, elbeszélő jellegű költeményt, „Tíz fantasztikus dolog, ami csodás mód nem volt elég ahhoz, hogy víz alá nyomja a Sinclair-t” címmel.

Tulajdonképpen egy fejezet, a ZX Spectrum ROM Cartridgek története olvasható e lapokon.

Egy az előtti fejezetnek – lévén az későbbi történet és már a Sinclair Research-höz kötődik - pedig a Black Watch esetét szánom, mely csodálatos szerkezetet a Sinclair Radionics dobta a piacra. Annak ellenére, hogy egy karóra volt, az a veszély, hogy a pontos idő meg a dátum megtudható a révén... Na, az nem fenyegetett!

1. A vezérlőchipje annyira érzékeny volt az elektrosztatikus feltöltődésre, hogy egy szőnyegpadlós iroda, egy műszálas ruhadarab, vagy egy kósza gondolat elegendő volt ahhoz, hogy megfektesse. Ilyenkor rendszerint az történt, hogy a kijelző egyetlen igen fényes szám mutatása közben fagyott le, ami túlterhelte az akkut, amitől az jobb esetben lemerült, rosszabb esetben felrobbant.

2. A benne lévő kvarckristály érzékeny volt a hőmérséklet változásaira, így télen és nyáron eltérő sebességgel járt az óra.

3. Az elem élettartama 10 napra volt tehető, de legalább nehéz volt cserélni.

4. A Black Watch úgynevezett „vakóra”, azaz a kijelzője csak akkor mutatja a pontos időt, ha megnyomva tartunk egy „control panel” nevű valamit. Hogy ez mi fán terem, én elképzelni sem tudom, de nem gomb, az biztos, mert egy reklámban úgy emlegetik az istencsapását, hogy „No unprofessional buttons”. Pont ez az egyetlen igen fontos kezelőszerv szeretett bedöglenni, ami jobb esetben nem tette lehetővé, hogy megtudd a pontos időt, rosszabb esetben nem tette lehetővé, hogy lekapcsold a kijelzőt. A folyamatosan működő kijelző... Na, mit csinált? Igen! Túlterhelte az akkut, amitől az jobb esetben lemerült, rosszabb esetben felrobbant. Előbb ez, később a ZX Spectrum billentyűzete... Lehetséges, hogy Sir Clive be volt oltva normálisan működő gombok ellen?

5. Több változat létezett belőle. A fekete verzió az órákat és perceket mutatta, vagy a másodperceket – gondolom éppúgy, mint kölyökkorom „kétgombos” kvarcórái. A valamivel szofisztikáltabb szürke válto-

zat már tudta – elvben – a dátumot is. Létezett készre szerelt és házilag összeszerelhető kettő verzióban is. Ez azért poén, mert az egyik korabeli elektronikai hobbimagazin két ruhacsipeszt, két rajzszöveget meg egy darab drótot jelöl meg segédeszközként, ahhoz, hogy az elemet a helyére lehessen tenni. Akkor milyen lehetett az alkatrészekből megépíteni?

6. Ez egy olyan kvarcóra, amiben van egy trimmer, amivel lehet állítani a futás sebességét. A fentebb említett magazin szerint az elem behelyezését követően további négy nap, hogy sikerüljön a megfelelő tempót belőni.

7. A műanyag tok összepattintós kivitelű. Apróbb hibája, hogy nem szeret egy darabban maradni. Állítólag kiadták a probléma megoldását egy külsős tanácsadó cégnek. Valamivel később visszakaptak egy Black Watchot egy rövid „Megoldottuk!” üzenet kíséretében. Az órán áthajtottak egy csavart középen.

Mára nagyon kevés példány maradt fent belőle, mert a visszaküldött hibás és eladatlan Black Watch készletek alkatrészeit felhasználták egy Microquartzhoz.

Sam. Joe

BLACK WATCH TÉNYEK

A Sinclair Radionics 1975 szeptemberében dobta piacra a legelső digitális órát a Black Watch képében. Összeszerelve 24.95, kitben 17.95 fontért lehetett hozzájutni. Manapság a gyűjtők 150-200 fontot is megadnak érte.

Egy korabeli legenda szerint azonban több órát hoztak vissza garanciális javításra vagy cserére, mint amennyit gyártottak. Állítólag a Sinclair szervízben még két év múltán is voltak javításra váró példányok.

Ezek után nyilvánvaló, hogy gazdaságilag kész katasztrófa volt ez a termék, állami támogatás nélkül a cég csődbe ment volna. Hatalmas raktárkészlet maradt az eladatlan és hibás órákból. Ezeket felhasználva, gyakorlatilag ugyanazt az elektronikát újradobozolva, Microquartz néven kihoztak 1977-ben egy autós órát, amit többé kevésbé már sikerült értékesíteni gond nélkül.

Balee

SPECTRUMOLÓGIA

A SPECTRUM ROM CARTRIDGE-EK TÖRTÉNETE

Kutakodjunk bármelyik megbízható Sinclair információforrásban is, egészen bizonyos, hogy mindegyik tízféle megjelent Spectrum ROM cartridge-et fog említeni. Pedig létezett egy a Sinclair Research által készített tizenegyedik is, melyről a legnagyobb valószínűséggel a gyűjtők zöme még csak nem is hallott! További érdekesség a Sinclair.hu által húsz évvel később bemutatott darab.

Azon nemesen egyszerű oknál fogva, hogy az alapgép minél olcsóbb lehessen, a Spectrum híján volt minden olyan beépített kiegészítőnek, mely a hasonlóan sikeres kortárs gépeken - szinte - kivétel nélkül megtalálható volt. Monitorkimenet, reset, joystickillesztő, hálózati kapcsoló, power LED és még sorolhatnánk. Ezek aztán szépen-sorban vagy megjelentek külső illesztők képében - vagy sem.

Bumm a lábba!



1983 szeptemberének végén dobta piacra a Sinclair Research, a ZX Interface 2-öt, mely azon túl, hogy két botkormányt illesztett az alapgéphez, egy apró kis ajtó mögött egy foglalatot rejtegetett, melybe ROM kártyákat lehetett dugni. Összesen tíz program - Planetoids, Space Raiders, Backgammon, Pssst, Cookie, Jetpac, Tranz am, Hungry Horrace, Horrace and the Spiders, Chess - látott napvilágot ROM kártyaként, de éppen úgy nem lettek igazán sikeresek, mint az illesztő. A ROM-ok kudarcának okai nyilvánvalóak.

Ez a tíz az a program, mely nagyjából minden Spectrum felhasználónak már megvolt, lévén rögvest a 16K-s Speccy piacra dobásakor 82-ben, vagy 83 elején ott voltak a gép mellett az üzlet polcán. Ráadásul - a Chess kivételével - mind 16K-s játék! Néha azon csodálkozom, hogy a Sinclair Research hogyan maradhatott fent ilyen sokáig, mert bizonyos húzásaiakkal mintha direkt lábon akarták volna löni magukat, hogy aztán csodálkozhasanak, hogy nehezen megy a szaladgálás.

A ROM-ok ügye is tipikusan ilyen! Nem elég, hogy totálisan inkurrens, 16K-s programokkal töltötték meg a csinos kis papírdobozban forgalmazott, még csinosabb kis műtűröket, de az eredeti, kazettás

változat árának (5 font) háromszorosát (15 font) kérték értük! (Arról már nem is beszélve, hogy a Planetoids eredeti kazettás változatának "B" oldalán volt egy bonusz játék is, a Missile.) Csoda, hogy nem álltak sorban a népek, hogy megvehessék ezeket? Ráadásul magán a ROM árán kívül a vásárlónak még ki kellett fizetnie 20 fontot az illesztőért. Számolgassunk csak egy kicsit! $10 \times 15 = 150$. $150 + 20 = 170$ font! Mindeközben egy 16K-s alapgép 125 fontba került, míg a 48K-s 175-be! Nem segítette az eladásokat az sem, hogy az, aki botkormányillesztőre vágyott, már jó eséllyel rendelkezett a Spectrumon kvázi szabvánnyá vált, Kempston rendszerű darabbal, mely mindössze 15 fontot kóstált és előbb hozzáférhető volt, mint az lf 2.

Ha a Sinclair Research-nél átgondoltabban válogatják ki a játékokat, akkor a ROM-ok még a magas ár ellenére is sikeresek lehettek volna. Nem sokat változtatott az sem a helyzeten, hogy idővel a ROM-ok árát 10 fontra csökkentették. Ha már megvolt valakinek kazettán a 16K-s játék, akkor 10 fontért sem kellett neki egy második kópia.

Sir Clive Sinclair a ZX Spectrumot eredetileg tanuló gépnek szánta és képtelen volt megbarátkozni azzal a gondolattal, hogy a vevők mást - hogy többet vagy kevesebbet, azt mindenki döntse el maga - látnak bele és esetleg csak és kizárólag játékokra használják és ebbéli álláspontjának többször is hangot adott. Ez a magyarázata annak, hogy a nagyon sikeres géphez nagyon kevés Sinclair Research gyártotta periféria készült és szöges elmentékeként azzal az üzletpolitikával, amit a CBM is követett a Commodore 64 esetében, hagyta hogy komplett háttérpar épüljön a ZX Spectrum kiegészítőkre.

Ezzel magyarázom a programok - játékok vagy más - ROM cartridge formátumban kiadásának kihagyott ziccerét is. Illetve még valamivel: Sinclairre jellemző volt, hogy ő problémákat szeretett megoldani. Ha úgy érezte, hogy egy adott feladattal elkészült, akkor az már nem foglalkoztatta tovább, más feladat után nézett. A „mindenki által megfizethető, tanuló-számítógépet”, mint kihívás



a ZX81 és a Spectrum után egészen egyszerűen kipiálta.

Kihasználatlan adottságok

A technológia olyan lehetőségeket rejteget, melyet még csak véletlenül sem használtak ki! A ROM akár a 16K-s, akár a 48K-s változat fejébe gyömöszölhetőnél is lényegesen nagyobb mennyiségű adatot tárolhat, ideálisan az "utántöltős" játékokhoz, látványos átvezető képernyőkhöz, digitalizált beszédhez, zenéhez. A kiadóknak külön jó hír, hogy a másolása - ha nem is megoldhatatlan - de legalábbis körülményes lett volna akkoriban. De végül csak az utolsó előnyt, az instant elérhetőséget, a gyakorlatilag zéró betöltődési időt élvezhették ki a felhasználók, mely egy 16K-s programnál már közel sem olyan vonzó, hogy indokolja a magas árat. Így a Spectrum ROM cartridge-ek mára ritka és féltett darabjai lettek a gyűjtőknek. Az árak állapotától függően 60-80 fontra is felszaladhat.

Követők és tervek

Az If 2-ön kívül több olyan interface is létezett, mely képes volt fogadni a ROM cartridge-eket, de még ez is kevés volt ahhoz, hogy érdemben változtasson a ROM-ok helyzetén.



A legelterjedtebb egyértelműen a Ram Electronics, RamTurbo nevű multifunkciós botkormányillesztője. (Megjegyzés: A Ram itt nem elírás, kosról van szó és nem memóriáról.) Az 1984-ben megjelent, 23 fontos, választhatóan Kempston, Cursor és Sinclair szabványokkal kompatibilis illesztő sem tudott sokat változtatni a már vázolt helyzeten. Később az árát 19 fontra csökkentették, és pont a ROM foglalatot hagyták le róla, hogy csökkentsék a gyártási költségeket. Mindazonáltal lehetséges volt ide utólag beforrasztani egy foglalatot, mert a kis felhajtható ajtó megmaradt. Kardos Bali gyűjteményében van (volt?) egy ilyen késői, foglalat nélküli darab. A RamTurbo talán az egyetlen a Spectrum illesztők között, melyre építettek egy műanyagfület, mely meggátolta, hogy gép bekapcsolt állapotában húzzák le, vagy dugják fel a gépre. (Nekem meg van egy olyan darabom, amiről az eredeti tulajdonos ezt

a biztonsági készségeket nemes egyszerűséggel letörte.)



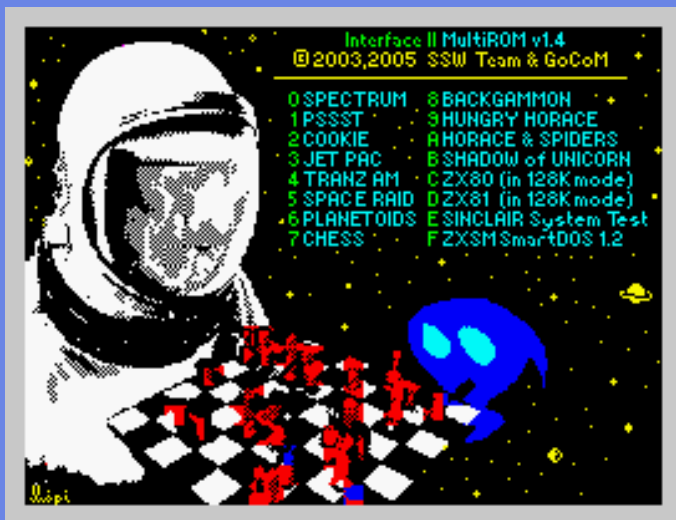
Kissé furán fest a Kempston Pro Joystick Interface a maga három borkormány aljzatával. A Kempston, Cursor és Sinclair illesztőkkel kompatibilis eszköz, ROM foglalatja fedetlenül díszileg a D9-es csatlakozók mellett. Az 1984-ben piacra dobott szerkezetért egyes források szerint 20, mások szerint 17 fontot kértek. Bár sokkal csúfabb szegényke, mint az If 2, mindenképp jobb vételnek tűnik, mint az.

Az AGF Ramslot csak és kizárólag a ROM cartridge-ek illesztésére volt alkalmas. Egyúttal volt a legolcsóbb (10 font) és a legrondább az összes közül. Lévén a konkurens termékek árának a feléért mindössze azok funkcionalitásának is nagyjából a felét kínálja, mindenki döntse el maga, hogy megérte-e vagy sem. Erről és az összes többi illesztőről is kitűnő képek láthatók Paul Farrow igencsak igényesen kivitelezett oldalán.

Ami pedig a ROM-okat illeti, nos a Parker Software a Sinclair Research féle ROM-ok sikertelensége ellenére tervezte, hogy 1984 októberében megjelent öt programot ebben a formátumban. Ezek a Popeye, Q*Bert, Gyuss, Star Wars: The Arcade Game, és Star Wars: Return of the Jedi - Death Star Battle lettek volna. A tervezet végül nem került megvalósításra.



2003-ban Gondos Csaba és a Sinclair.hu gondozásában jelent meg a MultiROM cartridge. A Csaba által tervezett és implementált ROM kártya egyetlen 256Kbájtos (2Mbit) EPROM-ban tartalmazza - egyebek mellett - a ZX Interface 2-höz megjelent ROM cartridge-ok tartalmát. A Spectrum bekapcsolása után megjelenő menüből kiválasztható, hogy melyik játékot kívánjuk betölteni. Az első - 1.2-es - változatot követő új, 1.3-as verzió új betöltő képernyőt kapott és néhány ROM csere is történt. A



MultiROM immáron tartalmazza az egyetlen saját ROM-mal rendelkező játék, a Shadow of the Unicorn ROM-ját is. (Közbevetőlegesen egy kis magyarázat: az 1985-ös Shadow of the Unicorn nem csupán a jól megszokott kazettán érkezett, hanem tartozott hozzá egy a buszcsatlakozóra dugható dobozka is, mely egyben volt Kempston joystick illesztő és tartalmazott 16K ROM-ot, mely maszkolta a Speccy saját ROM-ját. A játék maga hagyományosan, kazettáról töltődött be. A SotU nem If 2 ROM cartridge formátumban jelent meg, illetve nem a teljes játék volt az olvasható memóriában, nem számítható a ROM cartridge-ek közé.) Amennyiben 128K-s gépünk van, akkor az ZX80 és a ZX81 emulátorokat is futtathatunk, de helyet kapott még a GW03 nevezetű Spectrum ROM javítás és a ZXSM (SmartMedia kártya illesztő) legfrissebb ROM-ja is.

A titokzatos tizenegyedik

A fentebb felsoroltakon kívül azonban létezett egy tizenegyedik, szintén a Sinclair Research gondozásában és szintén 1983-ban megjelent cartridge is. Nem csoda, hogy szinte senki sem ismeri, lévén a System Test ROM Cartridge nem volt kereskedelmi forgalomban kapható. A hivatalos Sinclair szervizek kapták - vehették? - a diagnosztikát szolgáló segédletként. Az "Ikarusz műszerfal műanyagból" készült borítású cartridge nagy valószínűséggel egy hétköznapi EPROM-ot tartalmaz, szemben a nagy sorozatban készült játék cartok ROM-jaival. Azért a bizonytalanság mert a fehér holló ritkaságú készség, két darabból összeragasztott borítása nem tűnik roncsolás nélkül nyithatónak. A készlet része volt egy hétköznapi If 2-is. A tesztprogram az alapvető funkciók közül a billentyűzet, a tévémodulátor - színek (az alap 8 szín, plusz bright), border, flash -, RAM, és a

kazettás tárolás tesztelésére képes. A "Soak teszt" menüpont mibenlétével kissé tanácstalan vagyok, lévén a "soak" kifejezés áztatást, átítatást jelent. Amit igazándiból tesz, az mindössze annyi, hogy egy piros blokk-kurzort mozgat folyamatosan a képernyőn, nagyjából másodpercenkénti léptetéssel. Talán a stabilitás vizsgálatára, lefagyások tesztelésére szolgálhat. A kazettás egység, joystick és a printer teszt mibenléte egyértelmű. Izgalmasabb viszont az Interface 1 RS232-es kapujának és a LAN-nak a vizsgálata. Mindkettőhöz szükség van egy olyan gépre, melyen ugyanez a program fut, hogy legyen, ami válaszol a program által kiküldött karaktersorozatra.

A program további érdekessége, hogy a szerzője az a Dr. Ian Logan, aki a Microdrive ROM-jában lévő szoftvert is írta, illetve számos számítástechnikai témájú könyv szerzője. További érdekesség, hogy ő valóban orvosként praktizált. (Még mondja valaki, hogy ma már nem léteznek polihisztorok.)

A System Test ROM Cartridge igazi ritkaság, még a Spectrum témákban bibliának tekinthető World of Spectrum sem említi meg sehol. Amennyiben a burkolat megbontása nélkül sikerül kinyerni a tartalmát - ahogy KGB mondaná, a "tárgyvédelmet" elsődlegesnek tartom, így a borítás tönkretétele szóba sem jöhet -, megígérhetem, hogy az hamarosan hozzáférhető lesz itt.

Sam. Joe



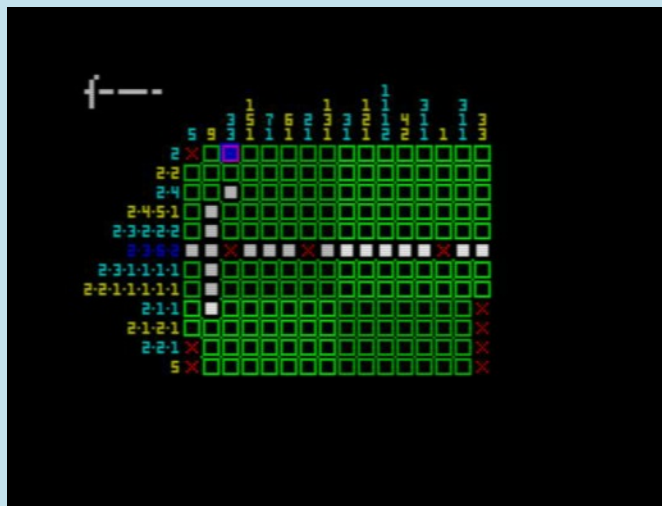
TUDDAD?

A **Speccyalista** elnevezést az **Úr 1998. évének május havában, annak 5. napján** adta nekünk Pgyuri, amit azóta is lelkesen licenszelünk. Noha a portál indításának gondolata 1997-re datálódik, születésnapunknak mégis ezt a napot tekintjük.

Logónkat 2004. szeptemberében Mópi alkotta meg számunkra egy pályázat keretében, amin a tagság egyhangúlag az **Ő** művét választott a **Speccyalista Baráti Kör jelképeként.**



Pixel Quest (Einar Saukas)



A multicolor játékmotorjairól híres szerző grafikai játéka a Pixel Quest. Az ilyen játékok egy berácsozott téglalapban vagy négyzetben egy figurát rejtnek. A Pixel Questnél ezek a figurák a nyolcvanas évek ZX Spectrum játékeinak szereplői, tárgyai. A négyzetrácsos rejtvény bal oldalán és felette számok találhatóak, amelyek azt jelzik, hogy az adott sorban, vagy oszlopban mekkora méretű színezett blokkok követik egymást. Minden ilyen színezett blokkot legalább egy üres, színezetlen blokk választ el a másik színezett bloktól. Az üres blokkok mérete viszont nincs megadva. A játék lényege a megfelelő négyzetrácsok beszínezése, ezzel az elrejtett figura megjelenítése.

Rabbit in Wonderland (Javier Fopiani)

Spanyol szerző, tehát egy Churrera motoros platformjáték, ami az Alice-féle csodaország univerzumában játszódik. Miután Alice megszökött Csodaországból, a tébolyult királynő lezáratt minden bejárati ajtót, így Alice soha nem térhet vissza. Ám egy kis idő múlva a királyné felfedezte, hogy a tükrök is átjárók, ezért elrendelte Csodaország összes tükrének elpusztítását. E nemesnek talán nem annyira nevezhető feladatot a fehér nyúlra bízta, mert ő a királyság legfürgébb alattvalója. A küldetés fő szabotőrének szerepe pedig Jack of Hearts-ra jutott, aki ha sikerrel jár, a nyúlnak biz a fejét veszik.



CR2: The Escape (Stephen Nichol)

Stephen első játéka volt a Cap'n Rescue, ahol a gonosz Mortyna boszorkány által majommá változtatott Robert Goode kapitánnyal kellett megkeresni saját, emberi testét. A történet innen folytatódik a CR2: The Escape-ben, ahol a jó Goode kapitány már Mortyna börtönéből igyekszik elszökni. Szökés közben sok veszedelemmel kell szembenézned, szúrós tüskékkel, mozgó ellenségekkel, áthidalhatatlan magasságokkal. A játék megjelenése után Stephen az első részt is átdolgozta (ha jól látom, akkor leginkább grafikai, a második rész stílusának megfelelően), és megjelentette a Cap'n Rescue 3.0-ás verziójaként.



Jumpcollision (Gamer80)

A ZX Dev'2015 Compora készült Raul "Gamer80" Naveros játéka a Jumpcollision. Raul látszólag össze nem illő stílusokat dobott össze ebbe a 80 szintet tartalmazó játékba, érdekes ötlet, de működik a dolog... Minden szint egyetlen képernyő, ahol a továbblépéshez ki kell lőni az összes ellenséget. És csak eddig sablonos a feladat, mert a célt egy platformokon ugráló (!) űrhajóval kell véghezvinni. Lőni csak felfelé lehet, ugrálás közben pedig kerülni kell az ellenségeket és a veszélyes platformokat, tárgyakat.



Lirus (RetroSouls)

Denis Grachev kilencedik játékának elődjét már 1998-ban megírta barátaival. Most újraalkotta az eredeti szinteket és ötleteket. A játéknak van külön 48K-s és 128K-s verziója. Fegyveres vírusirtóként kell tevékenykedned egy sokszintes komplexumban! Négyféle fegyvert vehetsz fel a pályákon, kezdő fegyvered egy sima lézerfegyver. Minden szinten biztosított az energia és lőszerutánpótlás. De ne válj emiatt elbizakodottá, a vírusok ugyanis gyors ütemben szaporodnak. Ha elfogy a lőszered az irtásuk közben és utánpótlásért vissza kell menned, akár egy közeli helyre is, mire visszaérsz, a vírusok megsokszorozódhatnak! Sőt, a szaporodásuk miatt akár körbe is keríthetnek. Ha egy ilyen helyzetben elfogy a lőszered, csak áttörni tudsz rajtuk, ami komoly energiaveszteséget és rossz esetben a hajód elvesztését okozhatja. Sajnos Denis ezzel a játékkal egyelőre búcsúzik a Spectrum színtértől, reméljük hamar visszatér!

Oceano (EgoTrip)

Amy elhatározta, hogy felfedezi a tengereket, megkeresve 20 mesés hatalmú kristályt, melyek már régóta elvesztek az óceánban. De robothalak őrzik őket, és savas buborékok, amik károsítják a tenger-alattjáróját. Van fegyvere, amivel ártalmatlanítani tudja a halakat, de ennek feltöltése eltart egy darabig, szóval nem lehet ész nélkül lőni őket. Rövid megállás után a tengeralattjáró süllyedni kezd az óceán feneké felé. Ha egy efféle helyszínen egyáltalán értelmezhető, akkor 22 szobából áll ez az útvesztő. EgoTrip fő platformja a CPC, de az AGD adta lehetőségeket kihasználva kezdett idén Spectrumra is fejleszteni. Az Oceano először CPC-re jelent meg, majd május végén Spectrumra is.



JÁTÉKÚJDONSÁGOK

Planeta Rojo (Salvador Cantero)



A cím angolul Red Planet, tehát a Vörös bolygó, a Mars a helyszín, ahol egy lepusztult űrállomás pincerendszerét kell megsemmisítened. Ehhez össze kell szedned 10 robbanóanyagot (a lila eszköz), levinni őket a komplexum aljában lévő raktárba, visszamenni a központi terembe és aktiválni a detonátort. Közben persze nem érhetsz szinte semmihez, az ellenségekhez pláne nem. Sőt, az oxigénpótlásról is kell gondolkodnod, mert nem vagy túl jól eleresztve ebből a fontos gázból. Korlátozott lőszerrel fegyvereket, valamint a zárt ajtókhöz kulcskártyákat is találsz utad közben. Salvador első játékához honfitársai, a Mojon Twins Churrera játékmotorját használta.

Rabid Mower (kas29)

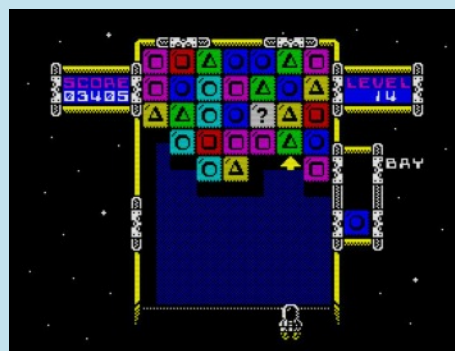
Egy játék az orosz Aleksei Kaskarovtól, amiről a CSSCGC, vagyis a vacak játékok versenye jutott először eszembe. Nem a minősége, vagy játszhatatlansága, hanem a témaválasztása miatt. Mert mi is a játék lényege? Le kell nyírni a fűvet. Mennyire egyszerűen hangzik, ráadásul a fűnyíród benzines, tehát a kábelét tuti nem nyírod el, tele van a tank, mi lehet a baj? Jobb, ha kipróbárod a gyakorlatban! Feladatod, hogy egyetlen billentyű segítségével a megfelelő helyeken lenyírod a pázsitot. Mikor nyírni akarsz, nyomd a gombot, mikor nem, akkor engedd fel. Néha fűnyírás közben belefutsz különböző akadályokba, amelyeket lehet, hogy nem érdemes lenyírni, pl. a bokrokat alighanem nem muszáj... Minden szintnek megvan a maga pontosságellenőrzése. A játék második, örültebb szakaszában ez tovább szigorodik. A 128K-s módban hallható AY muzsika Alex Clap munkája.



The Return of Traxtor (reidrac)

Reidrac, alias Juan J. Martinez 2014-ben a #GBJAM 3-ra (egy GameBoy témájú fejlesztői verseny) nevezett be a The Legend of Traxtorral, majd tavaly ősszel megírta a játékot Spectrumra is. Ennek a játéknak a folytatása, vagy inkább továbbfejlesztése a The Return of Traxtor. A történet a régmúlt időbe nyúlik vissza, 1000 évvel ezelőtt az utolsó háborúban a Traxtor megmentett minket. A játék a legenda története... Használd a vonósugarat a blokkok öblbe mozgatásához, és tüzelj, hogy visszadobd az öbl legfelső blokkját. Igazíts össze 3 vagy több egyforma blokkot, hogy elpusztítsd őket, mielőtt azok elérnék a védelmi vonalad. Az extra

blokkok felveszik annak a blokknak a színét, amire kilövöd őket. Ha kiüríted a játéktérrel, extra pontokat és szintemelést kapsz! A játék 25 szintes. A betöltőképernyőt Craig Stevenson rajzolta.



JÁTÉKÚJDONSÁGOK

Journey to Another World (J.B.G.V.)

J.B.G.V. 2011 óta készíti egyedi, felismerhető stílusú játékeit. Tizenegyedik játéka az 1991-ben megjelent Amiga játék, az Another World és alkotója, Éric Chahi előtti tisztelgés. A játék műfaja akció-kaland, de a szerzőt ismerők nem fognak meglepődni a kis arcade-minijátékokon sem, melyek a játékmenet részét képezik. Ugyanígy jellegzetes a játék előtti álló és mozgóképes történetmesélés is. Egy viharos estén Lester Knight Chaykin leparkol laborja bejárata előtt. A lifttel lemegy a mélyen föld alatt kialakított laborjába, ahol jóleső csend öleli körül, már nem érzi semmit a pár másodperccel korábban még kellemetlen égi háborúból. Teljes testszkennelés után be is lép az irodájába, ahol bekapcsolja számítógépét, bepötyögi a szokásos parancsot: LOAD"" és várja legújabb projektje a SAIMAZOON betöltését. Közben fent fokozódik a vihar és a hatalmas villámok egyike szerencsétlen helyen csap be, a

rendkívüli feszültség lejut az irodába, a számítógépbe és az események hatására Lestert elröpíti egy idegen világba, ahol mindjárt valami idegen, veszélyesnek tűnő vadállattal találja szemben magát... Nem tudja, hol van, de életben akar maradni.



Zombo (Monsterbytes)

Mikor a 303-as járat lezuhant a Chronosként ismert halálosan veszedelmes földön, az nem sok jót jelentett az életben maradó utasoknak. Csak egy jól nevelt zombi/ember-hibrid lehet a megmentőjük.

Ő Zombo, egy nagyon titkos, nagyon magas kormány szervezet kísérletének eredménye, amiben a legyőzhetetlen katona, vagy fegyver, vagy mifene létrehozása volt a cél. Te irányítod Zobot, meg kell mentened a hat túlélőt és végül legyőznöd a játékvégi rosszfiút. Színekódolt kulcsok vannak elszórva mindenfelé, meg mindenféle hasznos és haszontalan tárgyak. Sőt egy fegyver is, de vedd figyelembe, hogy Shootybang 2000™ csak vízszintesen tüzel. Függőlegesen csak az áprédelt 3000-es tud löni, de azt úgysem fogod megtalálni. A játékot Malcolm Kirk írta Jonathan Caldwell AGD-je segítségével, Zombo a brit 2000 AD sci-fi képregény-hetilap egy figurája, aki már önálló képregényeket is kapott alkotóitól, Al Ewing-től és Henry Flint-től. Zombo nem az első 2000 AD szereplő, aki Spectrum játék főszereplője is egyben, Strontium Dog, Slaine, Rogue Trooper és Judge Dredd is mind fontos szereplők, de néha maga Dan Dare is megjelent a képregényekben.

Mezei Róbert (M/ZX)

ZX81 KLASSZIKUSOK - 1. RÉSZ

Néha érdemes kicsit visszatekintenünk, hogy honnan is indultunk. Mondhat bárki bármit, leírhatatlan volt az az élmény, amikor először tölthettem be programot kazettáról a frissen zsákmányolt 1K-s ZX81-embe. Hasonló érzés volt még előtte a számítástechnika szakkörök keretében az első BASIC programok megírása kockás papíron, amelyet előbb-utóbb a gimi egyetlen ABC-80-ába is beütlegelhettünk. Már nem egyszer hallottam azt a kijelentést, hogy „ha először ZX81-et láttam volna, biztos nem érint meg a számítástechnika”. Nyilvánvaló sületlenség lenne ezt állítanunk, persze az sem teljesen mindegy, hogy az evolúciós láncba épp hol sikerült bekapcsolódnunk, ugyanakkor meg merem kockáztatni azt a kijelentést, hogy ezek a korai láncszemek sem elhagyhatóak, hogy tudjuk értékelni a kétségtelenül élvezhetőbb lépcsőfokokat.

Az viszont biztos, hogy ezen ZX81 programok legtöbbjének esetében mindenképp szükség volt egy köztes absztrakciós szintre, ami nélkül így utólag valóban kevésbé élvezhetőnek tűnhetnek ezek az opuszok.

Ebben a sorozatban szeretnénk áttekinteni a fontosabb állomásokat és nevezetesebb programokat. Gyanítom tartogathat meglepetést is ez a kis sorozat, de ha másra nem is, hát egy kis nosztalgiazásra remek lesz.

1K CHESS

(1982 - ARTIC COMPUTING)

ZX81 klasszikusok említésekor elsőnek az 1K Chess ugrik be talán mindenkinek, ami mellett nem mehetünk el szótlánul.

David Horne angol programozó 1982-ben készítette ezt a remek sakk programot az 1K-ás ZX81-re, és ez nem városi legenda. Ő megtette, ami akkoriban lehe-

tetlennek tűnt. A játék tartalmazza a legtöbb sakk szabályt, némi mesterséges intelligenciát (azaz tud lépni magától) és felhasználói felületet. A kód mindösszesen 672 bájt, ami a létező legrövidebb sakk program implementáció a világon.

Grafikát nem tartalmaz, betűk reprezentálják az egyes figurákat a táblán. A program csak fehérrel képes nyitni és kétféleképpen, vagy E3-ra vagy D3-ra. A kazettán ez 2 programnak felel meg az A és B oldalon, melyek kb. 30 másodperc alatt tölthetők be.

A három alábbi szabály kivételével képes értelmezni, ellenőrizni a lépéseket:

- sáncolás
- átalakulás (Az átalakulás vagy átváltozás a sakkban a gyalog különleges képessége. Amennyiben a gyalog eléri az ellenfél alapsorát (tehát a világos a 8. sort, a sötét pedig az 1. sort), a játékot tisztként folytatja.)
- en passant (Egy különleges lépés a sakkban, olyan ütés, amelyre akkor kerülhet sor, ha az egyik fél gyalogja kettőt előrelépve kikerülné az ellenfél gyalogjának átlós ütését. Az egyetlen eset a sakkban, amikor egy gyalog nem az átlósan előtte álló, hanem a mellé lépő figurát ütheti ki.)

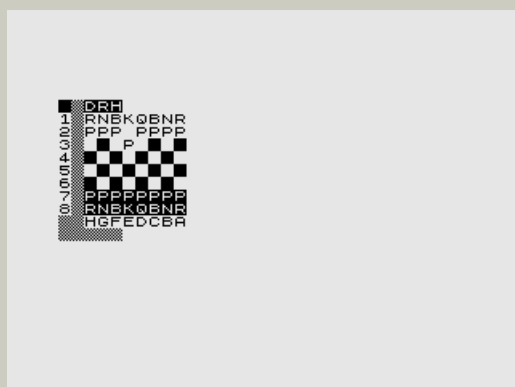
Állítólag a 2K Timex Sinclair 1000-esre készült verzió már tartalmazza ezeket a lépéskombinációkat.

A lépések megadásakor először a sor számát, majd az oszlop betűjelét kell megadni, pl. 7-E ó-E. A játék általában sakk-matt-tal ér véget, új játékhoz be kell tölteni a kazettáról újra a megfelelő verziót.

Érdekesség, hogy a program dokumentált forrását 1983 februárban a Your Computer című újságban publikálta, mely most az interneten is fellelhető:

<http://users.ox.ac.uk/~uzdm0006/scans/1kchess>

(Balee)



REVERSI

(1982 - SINCLAIR RESEARCH)

Szintén óriási kedvenc volt a népszerű táblás logikai játék számítógépes verziója már ZX81-en is, később pedig a Spectrumos verziója is. A játékot két játékos játssza, játszhatunk a gép ellen vagy akár ketten egymás ellen. A játék egy 8*8-as táblán zajlik. Az nyer, akinek a végén több kavicsa marad a táblán.

A szabályokat bizonyára már mindenki ismeri, ezért erre nem térnek ki. Lépni a sakkhoz hasonlóan az oszlop és sor megadásával lehet. Ha játék közben lépés helyett csak ENTER-t nyomkodunk, akkor egy menüt kapunk, amiben választhatunk, hogy befejezzük a partit vagy nehézségi szintet váltunk, esetleg oldalt cserélhetünk..

```

COPYRIGHT (C) MOI 1982
REVERSI V3.5
LEVEL 1
SCORE
  8 . . . . . . . . 8
  7 . . . . . . . . 7
  6 . . . . . . . . 6
  5 . . . . O O . . 5
  4 . . . . O O . . 4
  3 . . . . . . . . 3
  2 . . . . . . . . 2
  1 . . . . . . . . 1
  A B C D E F G H
  
```

Kicsit idegesítő, hogy a gép addig nem hajlandó lépni, amíg nem teszünk eleget a felszólításnak, nevezetesen, hogy nyomjuk meg az ENTER-t a folytatáshoz! Enélkül nemigen érdemes várni a gép lépésére órákig, mert eközben nem a lépés elemzésével foglalkozik. ☺

(Balee)

SEA WAR

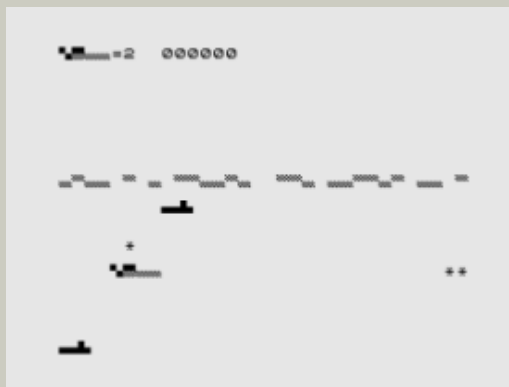
(19XX - PANDA SOFTWARE)

Egy osztálytársamtól kaptam ezt a játékot. Ez volt az első gépi kódban írt játék, amit valaha láttam. Így utólag talán nem tűnik olyan különleges programnak, mert egy tengeralattjáróval kell mindenféle ellent leküzdeni, lehet torpedókat, rakétákat lőni. De akkor ez lenyűgöző volt a BASIC programokhoz képest. Nagyon szépen, folyamatosan scrollozott vízszintesen,



gyors volt, egyszóval élvezhető játékelményt nyújtott, még ha manapság ezt nem is olyan egyszerű elképzelni.

Irányítás a kurzor gombokkal történik, a torpedók és rakéták kilövését több billentyűre is rádefiniálták, így mindenki kiválaszthatja, hogy melyik kényelmesebb számára, melyek a Q-W, 1-2, 9-0. Ráadásul egyidőben lőhetjük ki mindkettőt. Ugyanakkor arra is ügyeltek, hogy a torpedók és a rakéták újratöltési ideje nem egyforma, a torpedókat nagyobb gyakorisággal lődözhetjük magunk elé.



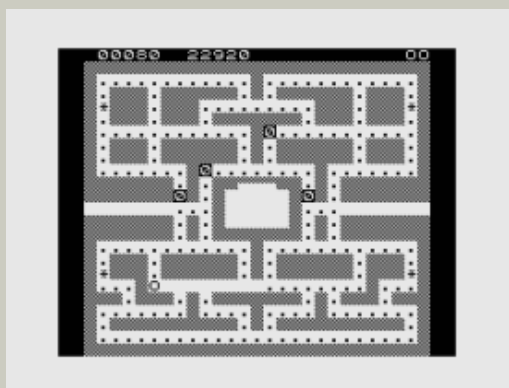
(Balee)

PAC-MAN

(19XX - PHILIP HARWOOD)

Valójában egy teljesen szabványos "Pacman" klón. Feladatunk nyilvánvalóan a sütik, azaz a pontocskák összegyűjtése, miközben inverz O alakba bújt szörnyek próbálnak végezni velünk. A csillagocskák elfogyasztása után a szörnyek egy rövid időre átváltoznak, na ekkor tudjuk őket is bekebelezni.

A pályán időnként egy plusszocska is feltűnik, melyet lenyelve pontszámunkat gyarapíthatjuk. A bal felső sarokban láthatjuk az eddigi legmagasabb pontszámot, mellette az éppen elért pontszámunk éktelenkedik, a jobb oldalon látható O-k darabszámából pedig életeink számára következtethetünk, hiszen mi magunk bújtunk a szép kövér O álcája mögé.



A játék egyetlen nagy hibája, hogy csak egy labirintust tartalmaz...

Írányítás: A - fel, Z - le, N - balra, R - jobbra. A játékot az ENTER megnyomásával indíthatjuk.

3D DEFENDER

(1982 - J.K. GREYE SOFTWARE)

A 3D Defender célja hasonló a Galaxy típusú játékokéhoz: A földi civilizációkra támadó ellenséges UFO-kat kell megsemmisíteni, azaz becserkészni és lelőni, mindeközben elkerülve a velük való ütközést és a saját űrhajónk lelövését. Ami mégis érdekessé teszi a játékot, az a megvalósítás a ZX81-en: Az űrhajónk ablakából kinézve látjuk a közeledő ellenséges csészealjkat, amik a távoli pontszerű megjelenésből 3 dimenzióban mozogva egyre nagyobbak lesznek és egyre több részlet lesz kivehető rajtuk. Ez nem tűnik nagy dolognak, de aki annak idején játszott a ZX81-en tipikus „a képernyőn az O betűt mozgatva kerüljük el az ütközést az I betűkkel” jellegű autóversenyző, síelő stb. programokkal, az nagyra értékeli ezt.



Űrhajónkat a vezérlő billentyűk botkormányoszerű kiosztásával irányíthatjuk: A Q...T félsor balra le, az Y...P jobbra le, az A...G balra fel, a H...L jobbra fel irányít, a legfelső és legalsó sor billentyűi indítják a tüzelést. Hasznos, hogy több billentyűvel is elérhetjük ugyanazt a funkciót, viszont nehezen megszokható, hogy nem egyenesen le-fel-jobbra-balra haladhatunk (ilyenkor két-két irány billentyűt kell egyszerre lenyomni). A mozgás reakcióideje és a tüzelés szerencsére elég gyors. Az ellenséges űrhajók felrobbanása látványos, ugyancsak jól megoldott (persze a ZX81 karaktergrafika keretei között) a támadó plazmafelhőjének megjelenése - természetesen ez is 3D-ben, így van esély kikerülni. Időnként még meteorosó is látható.

A navigációt az ellenség irányát jelző radar és egy magasságmérő segíti (a földnek ütközéstől óvakodni kell). A védekezést szolgáló pajzsunk minden plazmatámadásnál veszít erejéből. A lelőtt űrhajókért pont jár, az elszalasztott ellenségért viszont pontot lehet veszíteni - így a menekülés nem jó taktika.

A játék finomsága, hogy az elszalasztott ellenség a Földről újra felszáll, így revansot lehet venni. A teljes győzelemhez tartozó eredmény képernyőt soha nem láttam (talán nincs is :), de a legyőzésünket hírül adó képernyőn legalább egy jól megrajzolt gonosz idegen jelenik meg.



A játék közeli rokona a J.K. Greye Software klasszikus 3D Monster Maze játékanak, azt is és ezt is Malcolm Evans jegyzi.

(Stp)

Kardos Balázs (Balee)

Folytatjuk...

IDŐGÉP

1981. Március

A nyolcvanas évek legelején kezdi el az angol Ferranti cég az egyéni igények szerinti chipek gyártását, Sinclair rögtön ki is használja ezt. A ZX80-ból egy egész sor kis chipet felcserélnek a "nagy ULA" chipre. A megspórolt pénzen kibővítik a ROM memóriát, áttervezik az alaplapot és 1981 márciusában piacra dobják a ZX81-et, mely ténylegesen megteremtette az Angol személyi számítógép gyártást. A billentyűzetet is lecserélik és az előző szenzorai helyett, ebbe az új gépbe már fólia érintkezőket tesznek. Továbbra is egy gombnyomásra egy Basic parancs kerül beírásra. Már nem minden billentyűleütésre ellenőrzi a beírt sort, hanem a NEW LINE lenyomása után. A BASIC már tud decimális számokkal is számolni. A számítási sebessége nem túl nagy, de egy új BASIC utasításpár (SLOW, FAST) segítségével növelhető, a FAST üzemmódban ugyanis nem kezeli a képernyőt! A külső csatlakozói azonosak a ZX80-nál látottakkal. Ez a gép már 64K-ra volt bővíthető és blokkgrafikára is képes volt.

PROGRAMOZÁSTECHNIKA - BASIC

ZX SPECTRUM TV-BASIC KÜLÖNKIADÁS - 1. RÉSZ

Szeretettel üdvözlünk Kedves Olvasó!



A retro számítástechnika napjaink egyik uralkodó iparágává nőtte ki magát. Mindenki kíváncsi a múlt kutatásainak eredményeire, ezért döntöttünk úgy, hogy egy cikksorozatot indítunk.

Az elkövetkező részekben a ZX Spectrum számítógéppel fogunk foglalkozni és igyekszünk olyan

oldalairól megmutatni, amelyet remélhetőleg mindenki érdekesnek fog találni, aki már túl van első próbálkozásain a Spectrum felfedezésében.

Indulás

A ZX Spectrumban az a szép, hogy egy olyan személyi számítógép, amely bekapcsolás után azonnal tulajdonosa rendelkezésére áll, ahhoz, hogy használatba vegye, nem kell semmit betöltenie, állítania.

Tegyük meg és kapcsoljuk be.



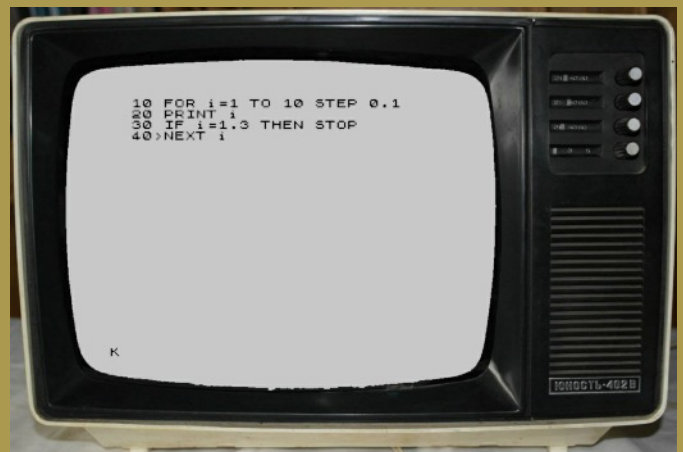
Az első találkozáskor ez a kedves felirat üdvözlöl minket, majd egy tetszőleges gomb lenyomása után rendelkezésünkre is áll a számítógép, hogy fogadja parancsainkat és utasításainkat. Sokak számára gyermekévei alatt nem is tudatosult az, hogy ilyenkor már bizony egy értelmező program fut a számítógépében, amely olvassa a billentyűzetet, rajzolja a képet és végrehajtja az adott feladatokat. Ez a program - amely be van égetve egy úgyne-

vezett ROM-ba - nagyrészt Steve Vickers zseniális programozó munkája volt.

Maga a 14445 bájtnyi, megközelítően 7000 programsorból álló programrendszer rendkívül okosan épül fel és hatalmas szabadságot ad a BASIC programozással ismerkedő vagy éppen magas szinten űző programozó számára. Tartalmazza a teljes BASIC utasítás értelmezőt hibáüzenetekkel, a grafikai betűkészletet és egy kalkulátort, amely az összes aritmetikai művelet elvégzésére alkalmas az egész és a tört számok tartományában.

Ez a ROM program a számítógép piacra dobása után már nem volt módosítható, így nagyon oda kellett figyelni, hogy a lehető legtöbb hibát kiszűrjék a tesztelések során és mindenképpen hibamentesen adják ki a ZX Spectrum számítógépet.

Írjuk meg mindjárt első programunk:



Ez az apró programocská képes a kezdő programozókat az örületbe kergetni kiszámíthatatlanságával, mert elég csak 1.2-re módosítani a 30. sorban szereplő értéket és minden azonnal működik.



(hol maradt a STOP ?)

A rejtély kulcsa a tört számok kezelésében rejlik, mindenesetre elbizonytalanító egy kezdő számára.

Üssük be - mintegy véletlenül - hogy

```
PRINT INT -65536
```



és rájöttünk, ez a kis gép nem sok komoly feladatra lesz jó a BASIC használatával.

Folytassuk az ismerkedést az utasításokkal:

Minden kézikönyv első utasításai között tárgyalt az INPUT. Jó-jó, az még érhető, hogy mire is szolgál, vagyis számok, szövegek programba történő bevitelére, de vajon miért lehet ilyen gonoszságokat művelni:

```
INPUT A$
```

... aztán az idézőjeleket törölni, majd

```
STOP
```



és meg is áll a program, ami után a Kedves Leendő Programfelhasználónk olyasmiket is tehet, amit nem vártunk tőle. Helyette alkalmazzuk inkább az

```
INPUT LINE A$
```

... na ez igen, itt nem jön be a STOP trükk, de jön a meglepetés, ha a Shift 6 közbelép. És vajon miért épp erre lép ki? Védhetetlen.

Legyünk hát elnézőek a szöveges bevitellel, ám a számok bekérésétől meneküljünk messzire!

```
INPUT A
```

... a STOP itt is közbeszól, de hogy be lehessen írni komplett képletet, programon belüli változót vagy akár egy RND-t, valljuk be, nem sokan gondoltuk volna.

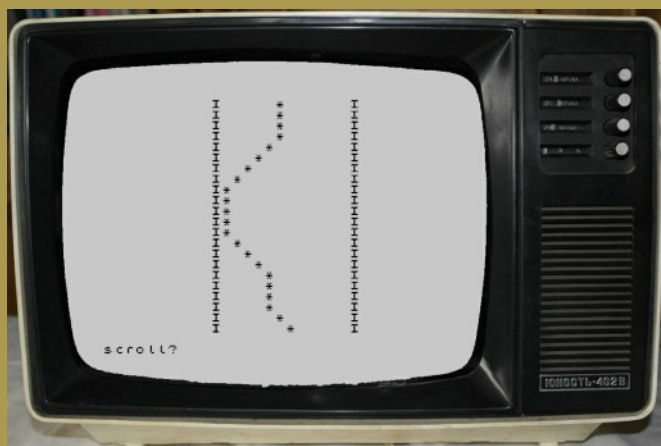
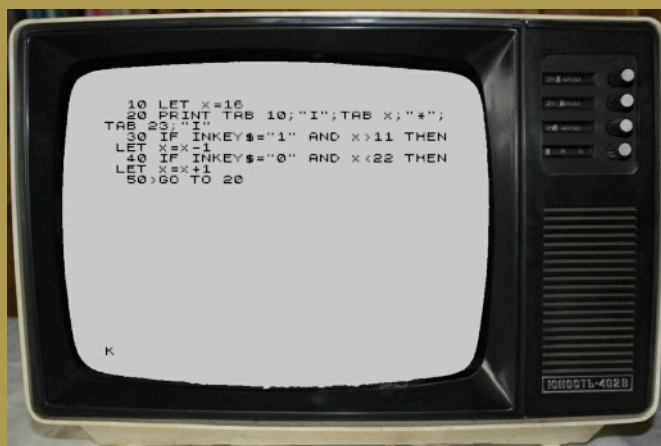


Végül jöjjön hát a döbbenet, az üres INPUT!

```
INPUT ;
```

... hát ez meg mire jó, hiszen semmit se kér be, semmit se csinál, de csak lát-szólag!

Közben pedig meg megoldja a spectrumos BASIC világ legnagyobb problémáját, amely az ilyen jellegű programokban jött ki mindenki számára (bár erősen megkérdőjelezhető, hogy ezért lenne):



Óóó, az a fránya Scroll?, ami megakasztja a szuperautóverseny programunk mozgását és megoldhatatlannak látszik mindaddig,

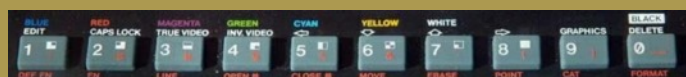
amíg egy-két szakmai könyvben a rendszerváltozóknál taglalt POKE 23659,255 nem segít rajta. Ugyan már, elég csak a fenti utasítást beszúrni és máris száguld minden megállás nélkül.

25 INPUT ;

Valamiért ez nem volt dokumentálva a BASIC kézikönyvben.

Ugorjunk a színekre!

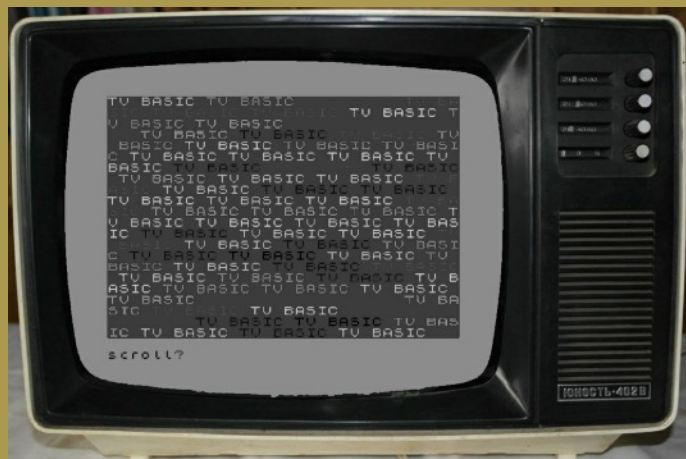
A Spectrum szivárvány színei szépen mutatnak egy színes tévén. A billentyűzetet is feldobják a számok feletti színes kiírások, amelyek utalnak arra, hogy melyik szám jelenti az adott színt, vagyis az 1-es a kék, 2-es a piros és így tovább.



Ki is próbálhatjuk sorban a BORDER utasítással, közben mintegy véletlenül a 8-at, 9-et is beütve, amely hibaüzenettel figyelmeztet, majd legyen már szebb a képernyő színe, folytassuk a PAPER utasítással... de vajon miért nem kapunk hibaüzenetet a

PAPER 9 vagy **INK 9**

kiadásakor? Mert bizony ez működik, a papír vagy a tinta színét feketére vagy fehérre állítja a másiktól függően, így biztosítva a mindenkori olvashatóságot. Zseniális, de már csak a 8-as érték érthetetlen, ami nem csinál semmit, de nem is üzen.



(színek megjelenítésére ez a tévékészülék nem alkalmas, de szép világoskék keretben, piros háttér előtt sok-sok színnel jelennek meg a feliratok)

Rajzoljunk is egy kicsit.

PLOT, **DRAW**, **CIRCLE** mind csodaszép dolgokra képesek, de mit vétettünk nekik mi, BASIC rajongók, hogy csak 176 soron rajzolhatunk, pedig 192 sornyi grafika állt rendelkezésre és bizony semmilyen korlátja nem lett volna megengedni ezt a luxust számunkra.



Arra az alsó fehér részre nem lehet rajzolni, pedig semmi akadálya nem lett volna, maga a Spectrum képes rá, de a BASIC-je nem. A ROM-ban található grafikai résszel is menne a teljes képernyő használata, de szándékosan lett csökkentve.

Jöjjenek a parancsok!

Az utasítás és a parancs között annyi a különbség, hogy míg az utasítást program futása alatt hajtja végre a számítógép, addig a parancsot azonnal a begépelést követően.

Révedjünk el mindjárt a **DATA** utasításban, mint parancsban, aminek hatása, hát - khm-khm - egyenlő a nagy semmivel, de íme az egyetlen BASIC utasítás, ami képes a gépet áttaszítani az ismeretlenbe:

RESTORE 65535

Aztán jöjjön a **LOAD**, amelyet szinte minden spectrumos elsőként tanul meg. Ahhoz, hogy egy programot betöltsünk, minimum a

LOAD ""

parancsot be kell gépelni. Nem is olyan egyszerű egy kezdőnek az idézőjelek miatt, de nézzük el a tervezőnek, hogy kötelezően kitétetten az üres idézőjeleket, amelyet - lássuk be - könnyedén elhagyhatott volna, azonban nehezen követhető a kissé zavarosra sikerült hibakezelése.

Nem az "R: Tape loading error" rettetgette gondolván, hanem a **BREAK** körüli zavarára:

D BREAK - CONT repeats, 0:1

jelenik meg **LOAD** közben megnyomva... vicces, mert a betöltés nem folytatható, csak újratekinthető... De ha utána

CONTINUE

és újra **BREAK**, akkor miért az

L BREAK into program, 0:1

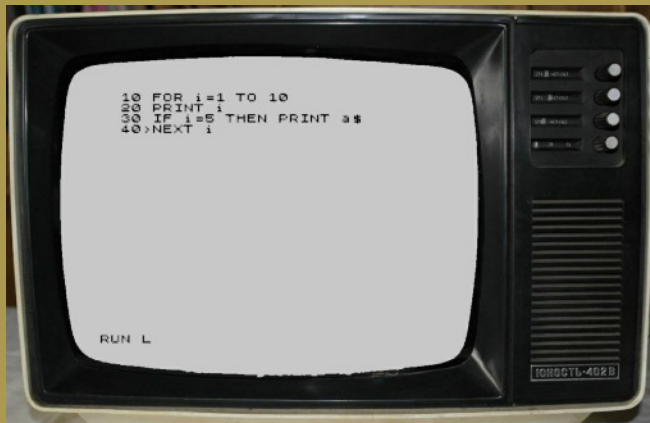
üzenet? Pedig a **CONTINUE** még ezt is lelkesen folytatja.

Mi is volt? CONTINUE?

Harsogjon a trombita, szóljanak a fanfárok, zúgjanak a harangok, mert itt száll le közénk a parancsok félistene, maga az örök rejtély, a hamis képmutató, az álcázott zseni, aaaaaaaaaaaaaaaaaa

CONTINUE

"Ha már "lebrékeltük" kis programunkat, folytathassuk is!" - gondolta tervezőnk! De mit is ronthatott el oly nagyon:



CONTINUE

C Nonsense in BASIC, 30:1

CONTINUE

2 Variable not found, 30:2

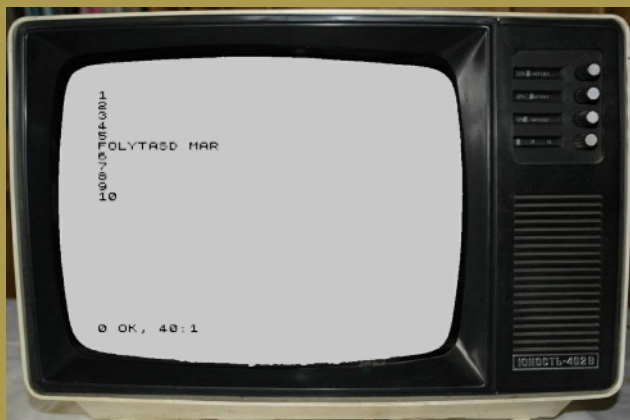
Legyünk kedvesek és javítsuk ki a hibánkat legalább menet közben és folytassuk:

```
LET a$="FOLYTASD MAR"
```

CONTINUE

C Nonsense in BASIC, 30:1

CONTINUE

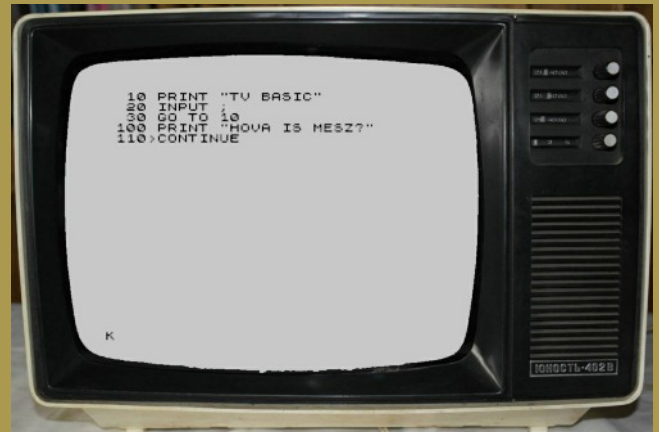


Úgy kell nekünk! Ha már hibás a programunk, legalább jó alaposan legyünk is összezavarva!

Végül vajon mire gondoltak a tervezők akkor, amikor ezt írjuk be:

100 CONTINUE kis nógatás, hogy folytasd bátran?

Kis akarással jól összekeverhetjük:



Először egy BREAK, majd GO TO 100 és történik nagy varázslat, a váratlan ugrás és folytatódik minden tovább, mintha mi sem történt volna.



Aztán vajon mi történik, ha bármilyen hibából adódóan megáll a mi kis programunk egy hibáüzenettel és lelkesen kiadjuk a CONTINUE utasítást?

Egyszerű a válasz, újból megkapjuk az üzenetet, hátha nem olvastuk el rendesen. Persze egy INPUT közbeni STOP után már meg sem kell lepődni, hogy újra bekérésre fog várni.

Összefoglalva tapasztalatainkat megállapíthatjuk, hogy a Spectrum BASIC-je csodálatosan szépre sikerült, de bizony maradtak elvarratlan szálak, amelyek nélkül nem jöhetett volna létre ez az írás.

CONTINUE, azaz FOLYTATJUK!

PROGRAMOZÁSTECHNIKA - MÉLYVÍZ

BORDER ÉS MULTICOLOR EFFEKT

Először is tisztázzunk az alapokat. A Spectrum képernyője két, jól elhatárolt területből áll, az egyik a háttér vagy keret, amely körbeveszi a hasznos területet, ahol szövegek, grafikák jeleníthetők meg, akár színesben is.



(ez a korabeli televízió-készülék nem volt alkalmas színek megjelenítésére, de árnyalatokkal érzékeltette)

A teljes képernyő 312 sorból áll, ebből 64 sor a felső és 56 sor az alsó háttérhez tartozik. A köztes 192 sor érdekesen épül fel, ugyanis bal és jobb oldalon 48 pixel háttér és közötté 256 pixel hasznos képterület látható.

A háttér (BORDER) statikus, egyszínű lehet, de a hasznos területen sem állíthatók a színek tetszőlegesen, egy 8x8 pixeles területen belül csak 2 szín, egy háttér (PAPER) és egy előtér (INK) választható. Ebből adódik a ZX Spectrum annyira jellegzetes "négyzetes" színkezelése.

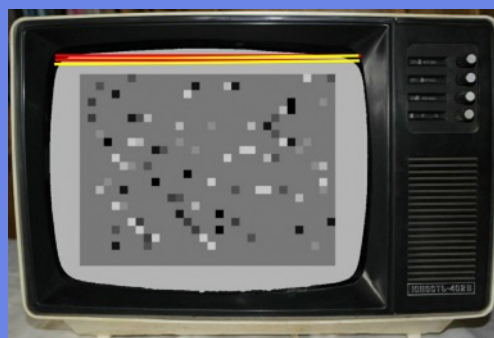
Akkor pedig hogyan lehetségesek az alábbi képek? Jól látható, hogy itt a háttér nem egyszínű, hanem több szín, sőt még elnagyolt rajzolat is megjelenik rajta. Valami trükk lehet mögötte...



Valóban, az úgynevezett **Border-effekt** eljárással mégis el lehet érni, hogy a korlát ledőljön.

Tekintsük át, hogyan alkotja a képet a ZX Spectrum!

Az elektronsugár a képernyő bal felső sarkából indulva soronként rajzolja meg a képet. 48K-s gép esetében 1 sor kirajzolása 176 T időt igényel (pirossal jelölve), majd még 48 T időt, amíg az elektronsugár visszafut a következő sor elejére (sárgával jelölve).



(a 'T' idő tulajdonképpen 3.5Mhz-re számított 1 időegység)

Ebből következően az első teljes szélességű sor végéig 224T telik el, a 2. sor végéig 448T, és így tovább. A teljes képernyő kirajzolása $312 \cdot 224T = 69888T$ időbe telik.

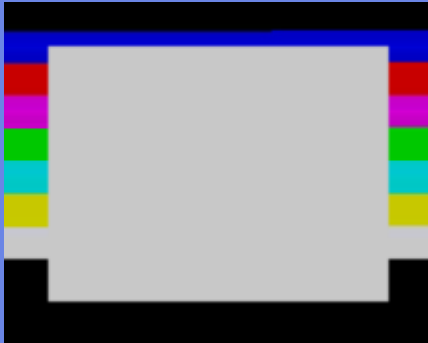
Ahogy az elektronsugár halad, folyamatosan figyeli, hogy milyen színnel kell dolgoznia. A háttér készítésekor az aktuálisan érvényes BORDER szint, míg a hasznos képernyőnél az adott soron belüli 8 pixelhez tartozó PAPER és INK szint jeleníti meg.

Nincs más teendőnk, mint ahogy az elektronsugár halad, a megfelelő pillanatokban beállítjuk a színeket és így már a megváltozott színnel rajzol tovább. Ha pedig a változtatásokat minden képváltásnál mindig ugyanabban az időpillanatban csináljuk (mert másodpercenként 50 kép jelenik meg), akkor folyamatosan ugyanazt a képet látjuk, ellenkező esetben vibrálás, képelcsúszás lesz az eredmény.

Már csak egy dolog kell ehhez a varázslathoz. Tudnunk kellene, mikor indul az elektronsugár a képernyő bal felső sarkából. Ez szerencsére elég egyszerű, mert a Spectrum tervezői összehangolták az úgynevezett megszakítási rendszerrel, tehát amikor bekövetkezik egy megszakítás (szintén másodpercenként 50-szer), akkor indul a képalkotás is.

Ezt BASIC-ből az alábbi módon láthatjuk:

```
10 BORDER 1: BORDER 2:
   BORDER 3: BORDER 4:
   BORDER 5: BORDER 6:
   BORDER 7: BORDER 0:
   PAUSE 1: GO TO 10
```



Szép állókép mutatja be a háttér "többszínűségét", a programban a PAUSE 1 utasítás biztosítja, hogy minden 50-ed másodpercben, pontosan a képrajzolással egyidőben frissüljön a kép, vagyis szinkronizál. A csíkok vastagsága mutatja, hogy mennyi ideig tart, míg a BORDER utasítást végrehajtja a számítógép, láthatóan elég lassú, ami a BASIC rendszernél teljesen elfogadható.

A gépi kódú programok sebessége ehhez képes hatalmas, látható, hogy milyen sűrűn képes színt váltani...



...de mégsem ad korlátlan lehetőséget a színezésre.

Eszünkbe is juthat, hogy "Hohó, hát nagyon sűrűn kell változtatni a színeket és akkor minden pixelnek más színt adhatunk!", de ez sajnos lehetetlen, mert amíg végrehajt egy utasítást a processzor, addig is száguld tovább az elektronsugár és az idő alatt is sok-sok pixelt hagy maga mögött változatlan színnel.

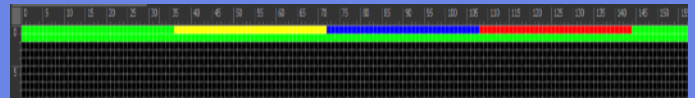
Minden gépi kódú utasítás végrehajtásának processzor ideje van (T). Még a legegyszerűbb gépi kódú

utasítás végrehajtása is 4T idejébe kerül a processzornak, ami közben az elektronsugár 8 pixelt rajzol meg. Nézzünk egy egyszerű példát:

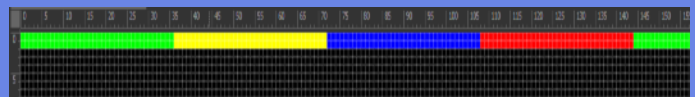
A háttér színe eredetileg zöld.

```
LD A, 6 ; beállítjuk a sárga színt,
         végrehajtási idő: 7T
OUT (254), A ; a háttér színét erre
             állítjuk,
             végrehajtási idő: 11T
LD A, 1 ; kék szín,
         végrehajtási idő: 7T
OUT (254), A ; a háttér színének
             beállítása,
             végrehajtási idő: 11T
LD A, 2 ; piros szín,
         végrehajtási idő: 7T
OUT (254), A ; a háttér színének
             beállítása,
             végrehajtási idő: 11T
LD A, 4 ; zöld szín,
         végrehajtási idő: 7T
OUT (254), A ; a háttér színének
             beállítása,
             végrehajtási idő: 11T
```

A háttér $7+11 = 18T$ ideig zöld marad 36 pixelen keresztül, majd sárgára vált újabb 36 pixelen keresztül, aztán kék, piros rész jön és végül zöld lesz a képernyő aljáig.



Ha megfelelő ideig várakozunk és újra kiadjuk a színezést, akkor a következő sor is ugyanilyenre színeződik.



Ha pedig ezt minden sorban megcsináljuk, máris függőleges színes csíkokkal teli a képernyő, hasonlóan ehhez a mintázathoz (a képernyő hasznos területén látható csíkok előre meg vannak rajzolva, csak a keretre kiterjesztéséhez kell az eljárás):



Hiába is igyekszünk nagyon sűrű színváltást csinálni, az utasítások végrehajtási ideje miatt ez nem lehet nagyon rövid, így az oszlopok szélességét sem lehet nagyon vékonyra csinálni.

Ha egy színt tovább is szeretnénk kitartani, azaz hosszabb csíkot húzni, akkor várakozni kell. Erre egy ciklus alkalmas, de figyeljük meg, mi okozza a programozási nehézséget:

```
LD A, 6 ; beállítjuk a sárga színt,
        ; végrehajtási idő: 7T
OUT (254), A ; a háttér színét erre
        ; állítjuk,
        ; végrehajtási idő: 11T
LD B, 50 ; ciklus számláló,
        ; végrehajtási idő: 7T
DJNZ ; ciklus,
        ; végrehajtási idő: 13T amíg
        ; B>0, végrehajtási idő: 8T,
        ; ha B=0
NOP ; üres utasítás,
        ; végrehajtási idő: 4T
NOP ; üres utasítás,
        ; végrehajtási idő: 4T
NOP ; üres utasítás,
        ; végrehajtási idő: 4T
NOP ; üres utasítás,
        ; végrehajtási idő: 4T
NOP ; üres utasítás,
        ; végrehajtási idő: 4T
LD A, 1 ; kék szín,
        ; végrehajtási idő: 7T
OUT (254), A ; a háttér színének beállítása,
        ; végrehajtási idő: 11T
```

Itt beépült egy várakozás (DJNZ utasítás csökkenti a B értéket, amíg az 0 nem lesz és csak azt követően megy tovább), ami végrehajtási idejét pontosan ki kell számítani ahhoz, hogy tudjuk hol vált majd színt.

Azaz a sárga szín meddig is marad?

$7 + 49 * 13 + 8 + 5 * 4 = 672T$, ami egyenlő $3 * 224T$, vagyis 3 soron át marad sárga, majd vált kékre.

Ajaj, de máris hibáztunk a számolásnál, mert még nem vált utána kékre, hiszen az további $7 + 11 = 18T$ időbe telik, így ez nem is 3 sor, hanem több. Ha kivesszük a 20T-nyi NOP utasításokat, akkor meg 2T hiányzik, olyan rövid utasítás meg nincs. Változtatni kell a cikluson, de akkor meg nem jön ki pontosan, vagyis megfelelő ideig futó utasításokkal kell pótolni. Szép munka, sok kitartást igényel.

Így aztán lehet színeznii a háttérrel, de mi a helyzet a képernyő hasznos részével ?

Az elv hasonló, csak itt már a **Multicolor-effekt** kerül alkalmazásra. Amikor az elektronsugár odaér a kívánt ponthoz, kiolvassa az oda tartozó színt és megjeleníti. Ezt követően a következő sorban fogja újra kiolvasni, tehát van időnk átírni a 8 pixelhez tartozó színértéket és így más szín jelenik meg abban a 8x8-as négyzetben, tehát egy négyzeten belül csak soronként lehet színt váltani, a 8 pixelen belül nem.



(itt csak az INK szín van cserélve soronként, a PAPER állandó fekete)



(itt már soronként a PAPER színe is változik, mutatva a színes megjelenítés csúcsát)

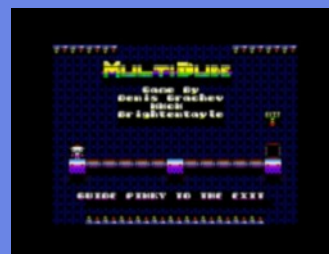
Programozni egy álló képet elég "egyszerű", de számos DEMO-ban láthatunk mozgó Border- vagy Multicolor-effektet a háttérben, miközben szövegek futnak érdekesebbnél érdekesebb módon.

Ezek is mind pontosan kiszámított időzítéseken alapulnak. Az időzítések az idő rövidege miatt sajnos nem szervezhetők egyszerű ciklusokba, sokszor csak látványosan felesleges utasításokkal lehet beállítani a pontos T időt.



Spectrumra sok Border-effektet tartalmazó játék készült, de csak 2004 után jelent meg néhány multicolor játék (pl.: TV-GAME 2004, Buzzsaw+ 2011). Számuk igencsak korlátozott, mert ilyen programokat nem lehetett (nem volt érdemes) eredeti gépeken fejleszteni, csak az emulátorok és komoly assembler fordítók tették lehetővé a programozási problémák megoldását, főleg a szükséges számítások segítségével.

2010 után készült futtatási környezet, úgynevezett "engine" is (pl.: Nirvana), amelyekkel igencsak látványos játékok készülhetnek. Itt nem kell foglalkoz-



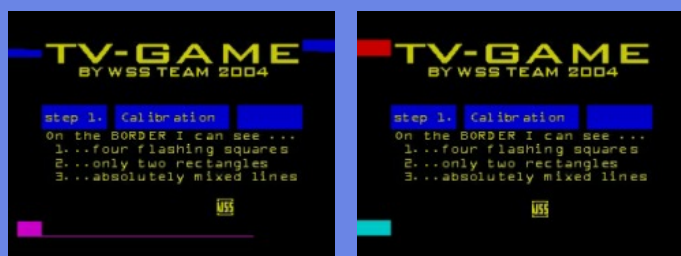
nunk magával a megjelenítéssel, elegendő a grafikát elkészíteni és megírni magát a játékprogramot, azonban ennek a technológiának egyik nagy hátránya a Spectrum belső hangszóróján történő hangok gyenge minősége vagy teljes hiánya, továbbá nem alkalmas nagysebességű képernyő mozgásokat vagy gyors reflexeket igénylő "lövöldözős" játékok készítésére.

Fő alkalmazási területe a logikai, puzzle típusú játékok szokatlanul szép grafikai köntösben megjelenítése.

A ZX Spectrum több változatban készült. Ezek az effektek minden gépen működnek ?

A kérdés remek és a válasz bizony az, hogy "Igen, de sajnos nem egyformán!". A 16 és 48K-s gépek között nincs eltérés, azok szépen szinkronizálják magukat és pontosan betartják az időzítési értékeket, de a később megjelent 128K-s gépek ebben változtak.

Számukra egy sor elkészítése időben 228T-re nőtt az eredeti 224T-s értékről, ezt ellensúlyozandóan a 312 sor 311-re csökkent, így egy teljes képernyő 70908T ideig készül szemben a 48K-s gépek 69888T-jével. Bár a különbség emberi mértékkel érzékelhetetlen, de a számítógép képalkotása teljesen elcsúszik ettől a közel 1.5 %-os lassulástól:



48-as gépen 128-as és 128-as gépen 48-as időzítéssel



A megfelelő géphez igazítva

Azoknak a programoknak, amelyek nagyon pontos időzítést követelnek - tehát nem csak nagyvonalakban színezik a háttér - le kell kérdeznünk, hogy milyen számítógépen futnak ahhoz, hogy tökéletes képet készíthessenek. A 128K-s gépek tulajdonosai néha nem is értik, mit látnak vagy mit kellene látniuk a képernyőn, amikor régebbi programot töltenek be.

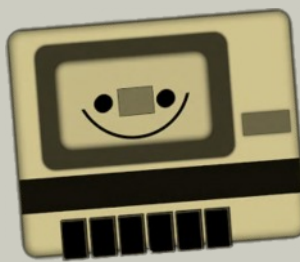
Van még lehetőség fejlődésre ezekkel az effektekkel?

Sajnos további lehetőségek ebben a technikában már nem rejlenek, a Spectrum elérte teljesítőképességének határát, így újdonságok e téren már nem várhatók. A programozóknak több, mint 20 évre volt szükségük arra, hogy maximálisan kiaknázzák a ZX Spectrum ezen képességeit.

Pgyuri

ANDROID TR-APP-ER

tapDancer: TZX/TAP lejátszó



PlayTZX: TZX/TAP lejátszó web kereséssel



PlayZX: TZX/TAP lejátszó, több ezer játékból választhatunk



TeeZiX - TZX/TAP player: TZX/TAP lejátszó



ZX TapeLoader: TZX/TAP lejátszó, WOS támogatással



PROGRAMOZÁSTECHNIKA - ASSEMBLY OVI

HOGYAN ÍRJUNK JÁTÉKOT ZX SPECTRUMRA 2-4. RÉSZ

2. Fejezet

Billentyűzetes és botkormány irányítás

Egyszerre csak egy gombot!

Amennyiben nem kapcsolod ki vagy babráltál más módon a Spectrum alapértelmezett megszakítás módjával, a ROM másodpercenként ötvenszer automatikusan olvassa a billentyűzetet és frissít több rendszerváltót a **23552**-es memóriacímen. A legegyszerűbb módszer egy billentyű megnyomásának beolvasására, ha először a **23560**-as címre nullát töltünk, majd folyamatosan figyeljük az itt található értéket, amíg nullától különböző értéket nem találunk ott. Az ekkor olvasott szám a leütött billentyű ASCII kódja. Ezt a megoldást leginkább a "Nyomj meg egy gombot a folytatáshoz" szituációkban tudjuk felhasználni, menüpontok kiválasztásánál, vagy a játékos nevének bekéréséhez az elért pontszám rögzítésekor. Egy ilyen rutin a következőképpen nézne ki:

; Példa 2.1

```
ld hl, 23560; LAST K rendszerváltó címe
ld (hl), 0 ; nullázzuk
loop ld a, (hl) ; LAST K új értéke
cp 0 ; továbbra is 0?
jr z, loop ; ha igen, nem történt bill.
; leütés
Ret ; volt gombnyomás
```

Billentyűkombinációk

Egyszerre csupán egyetlen billentyű leütése igen ritka, főleg gyors árkád játékok esetében, így igen hasznos volna, ha képesek lennénk egyszerre több billentyű lenyomását is észlelni. Ezen a ponton válik a dolog egy kicsit trükkösebbé. Memóriacímek olvasása helyett nyolc portról kérhetünk be adatot, amelyek mindegyikéhez egy 5 billentyűből álló sor tartozik. Természetesen, a legtöbb Spectrum modell ennél sokkal több billentyűvel rendelkezik, szóval joggal tehetjük fel a kérdést: hova tűnt a többi gomb? Nos, tulajdonképpen nem tűntek el, mert igazából ott sem voltak. Az eredeti Spectrum billentyűzet kiosztás csupán 40 billentyűből állt, ötös csoportokba szervezve – nyolc csoportot alkotva így. Bizonyos funkciók eléréséhez a gombok egy meghatározott kombinációját kellett megnyomni – például a törléshez a CAPS SHIFT és 0 billentyűket. Sinclair 1984-ben a Spectrum Plus színrelépésekor adta hozzá a billentyűzethez az új billentyűket. Működésük abban áll, hogy az általuk nyújtott funkciók eléréséhez az eredeti gumi-billentyűs modell billentyűkombinációinak leütését szimulálják.

Az eredeti billentyűzetkiosztás az alábbi csoportokból állt:

Port	Gombok
32766	B, N, M, Symbol Shift, Space
49150	H, J, K, L, Enter
57342	Y, U, I, O, P
61438	6, 7, 8, 9, 0
63486	5, 4, 3, 2, 1
64510	T, R, E, W, Q
65022	G, F, D, S, A
65278	V, C, X, Z, Caps Shift

Hogy kiderítsük, melyik billentyűk vannak éppen lenyomva, olvassuk be az adatot a megfelelő portról, amelyben a sorban mind az öt billentyűhöz egy bit van megfeleltetve a **d0-d4** alsó bitek közül (értékük 1,2,4,8 és 16). A d0-ás bit felel meg a külső billentyűnek, a d4-es a legbelsőnek. Furcsamódon, a bitek magas értékűek, ha a hozzájuk tartozó gomb nincsen lenyomott állapotban, és alacsonyak ha a gombot nyomva tartjuk éppen – pontosan fordított működést produkálva, mint amire számítanánk.

Hogy ellenőrizzük valamelyik sorban a billentyűk állapotát, egyszerűen betöltjük a sorhoz tartozó port számát a **BC** regiszterpárba, majd végrehajtjuk az **IN A,(C)** utasítást. Mivelhogy csupán az alsó bitek értékére vagyunk kíváncsiak, figyelmen kívül hagyhatjuk a többi bit, a számunkra érdektelen biteket kinulázva egy **AND 31** utasítással, vagy a szignifikánsakat kiforgatva az akkumulátorból az átviteli regiszterbe a vizsgálathoz, öt egymás után kiadott **RRA:CALL C,(cím)** utasítással.

Amennyiben az imént ismertetett módszer nehezen érthető, tekintsük a következő példát:

; Példa 2.2

```
Ld bc, 63486 ; 1-5 billentyűk/joystick
; port 2
In a, (c) ; lássuk a lenyomott
; billentyűket
rra ; legkülső bit = 1-es gomb
push af ; megjegyezzük az értéket
call nc, mpl ; ha megnyomva, mozgatas
; balra
pop af ; akkumulátor visszaállítása
rra ; következő bit (2-es érték)
; = 2-es gomb
push af ; megjegyezzük az értéket
call nc, mpr ; ha megnyomva, mozgatas
; jobbra
pop af ; akkumulátor visszaállítása
rra ; következő bit (4-es érték)
; = 3-as gomb
push af ; megjegyezzük az értéket
call nc, mpd ; ha megnyomva, mozgatas
; lefelé
pop af ; akkumulátor visszaállítása
```

```

rra                ; következő bit (8-es érték)
                   = 4-es gomb
call nc, mpu       ; ha megnyomva, mozgatus
                   felfele

```

Botko- rmányok

A Sinclair botkormányok 1-es és 2-es portja egyszerűen a billentyűzet két számcsoportjának lett megfeleltetve. Erről könnyedén megbizonyosodhatunk: a BASIC szerkesztőbe belépve a joystick-okat mozgatva számok fognak megjelenni. A Port 1 (Interface 2) hozzárendelése a 6, 7, 8, 9 és 0 gombok, a Port 2-é (Interface 1) pedig az 1, 2, 3, 4 és 5. A botkormány mozgatásának észleléséhez ugyanúgy olvassuk a megfelelő portot, ahogyan azt a billentyűzet esetében tettük. A Sinclair botkormányok a **63486**-os (Interface 1/port 2), és a **61438**-as (Interface 2/port 1) portokat használják, ahol a **d0-d4** bitek értéke 0 megnyomott állapotba, 1-es nyugalmi helyzetben.

A népszerű Kempston joystick formátum nincsen társítva a billentyűzethez, hanem a 31-es port figyelésével lehet kiolvasni az éppen aktuális állapotot. Ez azt jelenti, hogy mindösszesen egy egyszerű IN A, (31) utasítást kell használnunk. Ismételten, a d0-d4-es bitek jelölik az állapotot, ám ezúttal a bitek jelentése aszerint alakul, ahogyan azt eredetileg vártuk: a magas bit érték jelöli a botkormány kimozdítását valamelyik irányba.

; Példa 2.3

; Példa botkormány vezérlő rutin

```

joycon ld bc, 31 ; Kempston joystick port
      in  a, (c) ; bemenet olvasása
      and 2      ; "balra" bit ellenőrzése
      call nz, joyl ; mozgás balra
      in  a, (c) ; bemenet olvasása
      and 1      ; "jobbra" bit teszt
      call nz, joyr ; mozgás jobbra
      in  a, (c) ; bemenet olvasása
      and 8      ; "felfelé" bit
      call nz, joyu ; mozgás felfelé
      in  a, (c) ; bemenet olvasása
      and 4      ; "lefele" irány
      call nz, joyd ; mozgás lefelé
      in  a, (c) ; bemenet olvasása
      and 16     ; tűzgomb ellenőrzése
      call nz, fire ; tüzelés

```

Egy egyszerű játék

Tegyük is meg a következő lépést, és az eddig tanultakat átültetve a gyakorlatba, írjuk meg egy



egyszerű játék fő irányító részét! Ez lesz az alapja egy Százlábú játéknak, amit az elkövetkezendő néhány fejezet alatt fogunk fejleszteni. Még nincs birtokunkban minden szükséges tudás egy ilyen játékprogram elkészítéséhez, de azért el tudjuk kezdeni egy kis irányítási hurok megírásával, amely lehetővé teszi

a játékos számára egy kis törzs irányítását a képernyőn. Vigyázat! A program nem tartalmaz kilépési lehetőséget a BASIC-be, ezért mindenképpen mentsük el a forrásszöveget futtatás előtt!

; [Példa 2.4](#)

```

; Fekete képernyőt szeretnénk
ld a, 71 ; fehér tinta (7) fekete
          papíron (0), világosan (64)
ld (23693), a ; képernyő színek beállítása
xor a      ; akkumulátor nullázásának
          gyors módja
call 8859 ; maradandó keretszín

; Grafika beállítása
ld hl, blocks ; felhasználói grafikus
          elemek címe
ld (23675), hl ; az UDG ide mutasson

; Rendszerben, kezdődjön a játék!
call 3503 ; ROM rutin - képernyő
          törlése, 2-es csat. nyitása

; Koordináták inicializálása
ld hl, 21+15*256 ; kezdeti koordináták a
          HL-be
ld (plx), hl ; játékos koordinátái
call basexy ; x és y pozíciójának
          beállítása
call splayr ; játékos törzsének
          megjelenítése

; Ez a főhurok
mloop equ $
; Játékos törlése
call basexy ; x és y pozíciójának
          beállítása
call wspace ; üres hely a játékos
          pozíciójába

; Törölve van a játékos, átmozgathatjuk az új
          pozícióba mielőtt újra
          megjelenítjük
ld bc, 63486 ; billentyűk 1-5/joystick
          port 2

```

```

in a, (c)      ; kiolvassuk a megnyomott
                gombokat
rra           ; legkülső bit = 1-es gomb
push af      ; megjegyezzük
call nc, mpl  ; ha megnyomva, mozgás balra
pop af       ; akku helyreállítása
rra         ; következő bit (2-es
            helyiérték) = 2-es gomb
push af      ; megjegyezzük
call nc, mpr  ; ha megnyomva, mozgás
            jobbra
pop af       ; akku helyreállítása
rra         ; következő bit (4-es
            helyiérték) = 3-es gomb
push af      ; megjegyezzük
call nc, mpd  ; ha megnyomva, mozgás
            lefelé
pop af       ; akku helyreállítása
rra         ; következő bit (8-es
            helyiérték) = 4-es gomb
call nc, mpu  ; ha megnyomva, mozgás
            felfelé

; Az átmozgatás megtörtént, újra
                megjeleníthetjük a játékost
call basexy   ; x és y koordináta
                beállítása
call splayr   ; játékos megjelenítése
halt         ; várakozás

; Visszaugrás a főhurok elejére
jp mloop

; Játékos balra mozgatása
mpl          ld hl, ply ; Emlékezzünk, y a
                vízszintes koordináta!
            ld a, (hl) ; Mi a mostani érték?
            and a      ; Nulla?
            retz      ; Ha igen, nem tudunk tovább
                balra menni!
            dec(hl)   ; különben y = y-1
            ret

; Játékos jobbra mozgatása.
mpr          ld hl, ply ; Emlékezzünk, y a
                vízszintes koordináta!
            ld a, (hl) ; Mi a mostani érték?
            cp 31     ; A jobb szélén vagyunk
                (31)?
            ret z     ; Ha igen, nem tudunk tovább
                jobbra menni!
            inc(hl)   ; különben y = y+1
            ret

; Játékos felfelé mozgatása.
mpu          ld hl, plx ; Emlékezzünk, x a
                függőleges koordináta!
            ld a, (hl) ; Mi a mostani érték?
            cp 4      ; A pálya tetején vagyunk
                (4)?
            ret z     ; Ha igen, nem tudunk tovább
                felfele menni!
            dec(hl)   ; különben x = x-1
            ret

; Játékos lefelé mozgatása.
mpd          ld hl, plx ; Emlékezzünk, x a
                függőleges koordináta!
            ld a, (hl) ; Mi a mostani érték?
            cp 21     ; A képernyő alján vagyunk
                (21)?
            ret z     ; Ha igen, nem tudunk tovább
                lefele menni!
            inc(hl)   ; különben x = x+1
            ret

; A játékos törzsének, x és y koordináta
                értékének beállítása, ez a
                rutin kerül meghívásra a
                törzs törlése és
                megjelenítése előtt
basexy ld a, 22 ; AT pozicionáló kód.
rst 16
ld a, (plx) ; játékos függőleges koord.
rst 16 ; beállítjuk
ld a, (ply) ; játékos vízszintes koord.
rst 16 ; ezt is beállítjuk
ret

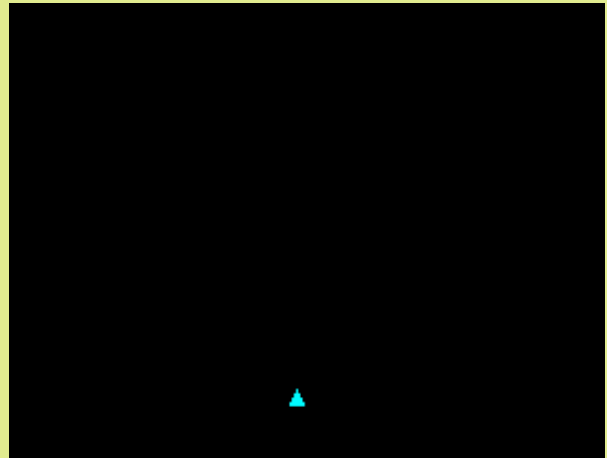
; Megjelenítjük a játékost a jelenlegi PRINT
                pozícióban
splayr ld a, 69 ; cián tinta (5) fekete
                papíron (0), világosan (64)
            ld (23695), a ; beállítjuk az ideiglenes
                színeket
            ld a, 144 ; 'A' UDG ASCII kódja
            rst 16 ; játékos kirajzolása
            ret

wspace ld a, 71 ; fehér tinta (7) fekete
                papíron (0), világos (64)
            ld (23695), a ; beállítjuk az ideiglenes
                színeket
            ld a, 32 ; SZÖKÖZ karakter
            rst 16 ; üres hely megjelenítése
            ret

plx defb 0 ; játékos x koordinátája
ply defb 0 ; játékos y koordinátája

; UDG grafika
blocks defb 16,16,56,56,124,124,254,254
                ; játékos törzse

```



Gyors, ugye? Igazából még le is lassítottuk a ciklust egy várakozó utasítással, de még így is 50 kép/másodperces sebességgel száguldunk, ami még mindig egy kicsit gyorsnak bizonyul. Ne aggódjunk, ahogy további funkciókat adunk a kódhoz, lelassítja majd a játékot. Ha érzel magadban elég önbizalmat, megpróbálhatod átalakítani a fenti programot, hogy Kempston botkormánnyal lehessen irányítani! A kivitelezés igazán nem bonyolult, pusztán ki kell cserélni a 63486-os port számot a 31-esre, valamint helyettesíteni a négy **call nc,(address)** utasítást ezzel: **call c,(address)**. (A bitek jelentése fel van cserélve, emlékszel?)

Az újradefiniálható irányítás kicsit furmányosabb. Ahogyan már említettem, az eredeti Spectrum billentyűzet 8 sorra volt felosztva, mindegyikben 5

billentyűvel. Ahhoz, hogy meghatározzuk, pontosan melyik billentyű is van lenyomva, először beolvassuk a biteket a megfelelő sorhoz tartozó portról, majd a d0-d4-es bitekből meghatározzuk a lenyomott gombot. Amennyiben lecserélnék a kódban az **ld bc,31** utasítást az **ld bc,49150** kódrészre, figyelhetnénk a billentyűk lenyomását H-tól az Enterig – bár ez nem nyújtana túl kényelmes újradefiniálási lehetőséget. Szerencsére van más lehetőség is a megvalósításra.

Meghatározhatjuk a billentyűsorok figyeléséhez szükséges portot a Spectrum leírásában szereplő formula segítségével is. A port címét az alábbi képlettel tudjuk meghatározni: $254+256*(255-2^n)$, ahol n a port száma a 0-7 zárt intervallumból. A **654**-es címen található egy ROM-rutin, amely igen sok fáradságos munkát spórolhat nekünk: az **E** regiszterben visszaadja a lenyomott billentyű sorszámát a **0-39** tartományban. A 0-7 értékek a sorok legbelső gombjainak felelnek meg, (ezek rendre B, H, Y, ó, 5, T, G és V), 8-15 jelöli a következő oszloponyi billentyűt, és így tovább egészen a legkülső sor 39-es billentyűjéig, ami a CAPS SHIFT. A teljesség kedvéért, a SHIFT billentyű státuszát a **D** regiszter tartalmazza. Amennyiben egyik gomb sincsen lenyomva, **E** értéke **255**.

Az említett ROM-rutin csupán egyetlen billentyű lenyomását adja vissza, így sajnos nem alkalmas billentyű kombinációk figyelésére. Ahhoz, hogy több lenyomott gomb esetén is megmondhassuk, mely gombok vannak nyomva tartva, magunknak kell meghatározni a tesztelni kívánt billentyű portját, kiolvasni az értéket, majd kiértékelni az eredményt a megfelelő bit állapota alapján. Én egy igen hasznos rutint használok a feladat elvégzéséhez, amely az egyetlen eljárás játékaimban, amit nem magam írtam. Az érdem *Stephen Jones*-é, aki számos kiváló cikket írt a *Spectrum Discovery Club* számára sok évvel ezelőtt. A kód használatához be kell tölteni az ellenőrizni kívánt billentyű számát az akkumulátorba, meghívni a **ktest** címet, majd ellenőrizni az átviteli jelzőt (carry flag). Ha történt átvitel, a billentyű nem volt lenyomva. Amennyiben ez a fordított logika zavaró, helyezze el egy **CCF** utasítást a **RET** elé!

; Példa 2.5

; Mr. Jones billentyű tesztelő rutinja

```
ktest ld c, a ; billentyű szám A-ba
and 7 ; d0-d2 bitek maszkolása
inc a ; 1-8-as tartományba növelés
ld b, a ; B-be
srl c ; c-t elosztjuk 8-cal,
srl c ; hogy megtaláljuk a sor
számát

srl c
ld a, 5 ; soronként 5 gomb
sub c ; kivonjuk a pozíciót
ld c, a ; C-be áttöltjük
ld a, 254 ; az olvasandó port magas
bajtja

ktest0 rrca ; pozícióba forgatás
djnz ktest0 ; ismétlés amíg releváns
sort találunk
```

```
in a, (254) ; port olvasása (a=magas
bajt, 254=alacsony)
ktest1 rra ; bit kiforgatása az
eredményből
dec c ; ciklus számláló
jp nz, ktest1 ; ismételjük amíg a carry-
ben lévő bithez érünk

ret
```

3. fejezet – Hanghatások

A hangszóró

Kétféleképpen csálhatunk elő hangokat és zenét a ZX Spectrumból. Ezek közül a tetszetősebb és bonyolultabb út az AY38912 hangchipen keresztül vezet a 128K-s modellekben. Ezt a módszert egy későbbi fejezetben ismertetem részletesen, most a 48K-s gép képességeivel fogunk foglalkozni. Habár egyszerűnek tűnik, mégis rendkívül alkalmas rövid, éles hangeffektek előállítására a játékok színesítésénél.

Beep

Először is azt kell tudnunk, miként tudunk megadott hangmagasságú és hosszúságú hangot generálni. A Sinclair ROM-ban található egy könnyedén használható rutin a **949**-es címen, amelyik pont megfelel a célnak. Csupán annyit kell tennünk, hogy betöltjük a hangmagasságnak megfelelő paraméter értéket a **HL** regiszterpárba, a hosszét a **DE**-be, majd meghívjuk a **949**-et, és már van is egy megfelelő "beep" hangunk!

Sajnos a paraméterek előállításának módszere egy kicsit trükkösebb, ugyanis igényel némi számítás. Ismernünk kell a kibocsátandó hang magasságának frekvencia értékét Herz-ben megadva - lényegében ez a szám megfelel annak az értéknek, ahányszor a hangszórónak hangot kell kiadnia egy másodperc alatt, hogy a megfelelő hangmagasságot állítsa elő. Az egyvonalas C hang oktávjának hangjait foglalja össze az alábbi táblázat (# a zenei keresztet jelenti):

Egyvonalas

C 261.63

C# 277.18

D 293.66

D# 311.13

E 329.63

F 349.23

F# 369.99

G 392.00

G# 415.30

A 440.00

A# 466.16

(A programba történő könnyebb beilleszthetőség érdekében meghagytam a tizedespontokat. A magyar helyesírásnak megfelelően tizedes vesszőknek kellene szerepelnie a táblázatban – a fordító.)

Hogy egy oktávval magasabban lévő hangot kapjunk, egyszerűen duplázuk meg a frekvenciát, egy oktávval alacsonyabbhoz felezzük! Például ha kétvonalas **C** hangot szeretnénk, vegyük az egyvonalas **C** hanghoz tartozó frekvenciát – **261.63** –, majd szorozzuk meg kettővel: **523.26**.

Miután a frekvenciát meghatároztuk, megszorozzuk a kívánt hanghossznak megfelelő értékkel másodpercben kifejezve, és ezt adjuk át a ROM rutinnak a **DE** regiszterekben. Ennek megfelelően, ha egy egyvonalas **C** hangot szeretnénk megszólaltatni egy tized másodpercen keresztül, az ehhez szükséges érték $261,63 \cdot 0,1 = 26$. A hangmagasság a következőképpen számítható ki: $437500 / (\text{a frekvencia})$, majd ebből vonjunk ki **30,125-öt!** Ezt az értéket kell átadnunk **HL**-ben. Egyvonalas **C** esetében ez a következő: $(437500/261,63) - 30,125 = 1642$. Összefoglalva:

DE = Hossz = Frekvencia * Másodpercek száma

HL = Magasság = $(437500 / \text{Frekvencia}) - 30,125$

(Habár az osztás művelet precedenciája magasabb a kivonásénál, az egyértelműség kedvéért kitétem az implicit zárójelet – a fordító.)

A fentebb leírtaknak megfelelően, az alábbiak szerint tudunk megszólaltatni egy kétvonalas **Gisz** hangot (**G#**) 1/4 másodpercig:

; Példa 3.1

```
; Egyvonalas Gisz frekvenciája = 415.30
; Kétvonalas Gisz frekvenciája = 830.60
; Hossz = 830.6 / 4 = 207.65
; Magasság = 437500 / 830.6 - 30.125 = 496.6
```

```
ld hl, 497      ; magasság
ld de, 208     ; hossz
call 949      ; ROM beep rutin
ret
```

; Példa 3.2

; Természetesen ez a rutin nem csak zenei hangok megszólaltatására alkalmas - számos hanghatás előidézhető vele! Az egyik kedvencem, egy egyszerű hang-hajlító eljárás:

```
ld hl, 500      ; Kezdő hangmagasság
ld b, 250      ; effekt hossza
loop push bc   ; magasság tárolása
push hl        ; nagyon rövid hossz
ld de, 1       ; ROM beep rutin hívása
call 949      ; hossz visszaállítása
pop hl        ; magasság növelése
inc hl
pop bc
djnz loop     ; ismétlés
ret
```

Ha időnk engedi, játszadozzunk a fenti rutinnal – elég könnyű a hang magasságát feljebb vagy lejjebb állítani, valamint a kezdő frekvencia, a hossz és a hang hajlításának megváltoztatásával számos érdekes hatást idézhetünk elő. Egy dologra érdemes odafigyelni: ne adjunk meg nagyon ésszerűtlen hangmagasság és hosszúság értékeket, mert a beeper rutin beragadhat, és csak abban az esetben nyerhetjük vissza az irányítást gépünk fölött, ha újraindítjuk!

Fehér Zaj

A hangszóró használata esetén nem szükséges ragaszkodnunk a ROM-ban lévő rutinok használatához. Könnyen elkészíthetjük a saját hanggeneráló algoritmusainkat is, főleg ha csupán statikus zörejt szeretnénk előállítani csattanásokhoz és robbanásokhoz. A fehér zaj előállítása általában izgalmasabb az eddigi látottaknál.

Ilyen zörejt megszólaltatásához pusztán egy gyors és egyszerű véletlen-szám generátorra van szükségünk. (Egy Fibonacci sorozat éppen kapóra jön, azt javaslom, léptessünk egy mutatót az első 8K-s ROM részen, és olvassuk be időről-időre az aktuális helyen lévő bájtot, hogy megfelelően véletlen 8 bites számot kapjunk.) Ezután írjuk ki ezt az értéket a **254**-es portra! Emlékezzünk, hogy ugyanezen a porton keresztül vezérelhetjük a keret színét is, így amennyiben ha el akarjuk kerülni a több színnel csíkozott keret effektust, le kell maszkolnunk a keret-biteket egy **AND 248** utasítással, és hozzáadni a megjeleníteni kívánt keret színének a számát az értékhez – 1 a kék, 2 a piros, stb – mielőtt kiadjuk az **OUT (254)** utasítást. Miután ez megvolt, egy rövid várakozási ciklust kell beiktatnunk – magas hanghoz rövidebbet, mélyebb hanghoz hosszabbat –, és megismételni a műveletet néhány százszor. Így egy kiváló, ütközésre emlékeztető hanghatást kapunk.

Az alábbi rutin egy hangeffekten alapul az Egghead 3-ból:

; Példa 3.3

```
noise ld e, 250 ; ismétlés 250-szer
      ld hl, 0   ; kezdő mutató a ROM-ban
noise2 push de
      ld b, 32  ; lépés mértéke
noise0 push bc
      ld a, (hl) ; következő "véletlen" szám
      inc hl    ; mutató
      and 248  ; fekete keretet szeretnénk
      out (254), a ; hangszóróra ki
      ld a, e   ; ahogy e értéke csökken...
      cpl     ; ...növeljük a várakozást
noise1 dec a    ; csökkentjük a ciklus
              ; számlálót
      jr nz, noise1 ; várakozó ciklus
      pop bc
      djnz noise0 ; következő lépés
      pop de
      ld a, e
      sub 24    ; lépés mérete
      cp 30    ; tartomány vége
```

```

ret z
ret c
ld e, a
cpl
noise3 ld b, 40 ; csend
noise4 djnz noise4
dec a
jr nz, noise3
jr noise2

```

4. fejezet - Véletlen számok

Véletlen számok generálása gépi kódból trükkösebb feladatnak bizonyulhat a gyakorlatlan programozó számára, mint amilyenek elsőre gondolnánk. Először is, tisztázzunk egy fontos tényt: teljesen véletlen számgenerátor, mint olyan, nem létezik. A processzor pusztán utasításokat hajt végre és nem bír önálló akarattal, aminek segítségével véletlen számokkal tudna előállni hasraütés-szerűen. Ennek híján egy előre meghatározott formula segítségével tudja számok előre meg nem jósolható sorozatát készíteni, amelyek látszólag semmilyen mintát nem követnek, így a véletlenszerűség látszatát keltik. Ennek fényében megállapíthatjuk, hogy csupán csak hamis-, vagyis pszeudo-véletlen számokkal kell beérnünk.

Egy elegáns és kézenfekvő módja a pszeudo-véletlen szám generálásának a Fibonacci számsor felhasználása. Mielőtt megrémülnénk, rendelkezésünkre áll egy könnyebb és gyorsabb módszer is 8 bites véletlen számok előállítására Spectrumunkon: léptessünk egy mutatót a ROM címeken, és az éppen aktuális címen tárolt bájtot olvassuk ki. Ennek a megközelítésnek azért van egy hátulütője is: a Sinclair ROM tartalmaz egy meglehetősen egységes és egyáltalán nem véletlenszerű területet a vége felé, amelyet érdemes elkerülnünk. Még abban az esetben is, ha a mutató határait az első 8 KB-nyi ROM-ra korlátozzuk, 8192 „véletlen” számhoz jutunk, amely jóval több, mint amire egy átlagos játékhoz szükségünk lehet. Minden játékom, amelyik véletlen számokkal dolgozik az alábbi, vagy ahhoz nagyon hasonló metódust használ a véletlenszerűség eléréséhez:

; Példa 4.1

```

; Egyszerű pszeudo-véletlen szám generátor.
; Egy mutatót léptet a ROM területen (a seed-
;                ben tárolva), visszaadva
; a megcímezett bájt tartalmát.
random ld hl, (seed) ; Mutató
ld a, h
and 31 ; Az első 8 KB-on belül
; tartjuk
ld h, a
ld a, (hl) ; "Véletlen" szám a mutatott
; helyről
inc hl ; Mutató léptetése
ld (seed), hl
ret
seed defw 0

```

Állítsuk is munkába új véletlen szám generátorunkat a Százlábú játékban! Minden Százlábú játéknak szük-

sége van gombákra – mégpedig meglehetősen sokra –, szétszórva a játéktérben. Jó szolgálatot fog tenni a fenti rutin a gombák koordinátáinak véletlenszerű meghatározásához. Egészítsük ki a programot:

; Példa 4.2

```

; Fekete képernyőt szeretnénk
ld a, 71 ; fehér tinta (7) fekete
; papíron (0), világosan (64)
ld (23693), a ; képernyő színek beállítása
xor a ; akkumulátor nullázásának
; gyors módja
call 8859 ; maradandó keretszín
; Grafika beállítása
ld hl, blocks ; felhasználói grafikus
; elemek címe
ld (23675), hl ; az UDG ide mutasson
; Rendben, kezdődjön a játék!
call 3503 ; ROM rutin - képernyő
; törlése, 2-es csat. nyitása
; Koordináták inicializálása
ld hl, 21+15*256 ; kezdeti koordináták a
; HL-be
ld (plx), hl ; játékos koordinátái
call basexy ; x és y pozíciójának
; beállítása
call splayr ; játékos törzsének
; megjelenítése
; Feltöltjük a játéktérrel gombákkal
ld a, 68 ; zöld tinta (4) fekete
; papíron (0), világos
; színnel (64)
ld (23695), a ; ideiglenes szín beállítása
ld b, 50 ; kezdetnek csak pár gomba
mushlp ld a, 22 ; AT karakter vezérlőkódja
rst 16
call random ; "véletlen" szám generálása
and 15 ; a [0..15] függőleges
; tartományban
rst 16
call random ; újabb véletlen szám
and 31 ; a [0..31] vízszintes
; tartományban
rst 16
ld a, 145 ; UDG 'B' a gomba grafikája
rst 16 ; kihelyezzük a képernyőre
djnz mushlp ; ciklus amíg nem végeztünk
; a gombákkal
; Ez a főhurok
mloop equ $
; Játékos törlése
call basexy ; x és y pozíciójának
; beállítása
call wspace ; üres hely a játékos
; pozíciójába
; Törölve van a játékos, átmozgathatjuk az új
; pozícióba mielőtt újra
; megjelenítjük
ld bc, 63486 ; billentyűk 1-5/joystick
; port 2
in a, (c) ; kiolvassuk a megnyomott
; gombokat
rra ; legkülső bit = 1-es gomb
push af ; megjegyezzük
call nc, mpl ; ha megnyomva, mozgás balra
pop af ; akku helyreállítása
rra ; következő bit (2-es
; helyiérték) = 2-es gomb
push af ; megjegyezzük

```



```

call nc, mpr      ; ha megnyomva, mozgás jobbra
pop af           ; akku helyreállítása
rra             ; következő bit (4-es helyiérték) = 3-es gomb
push af         ; megjegyezzük
call nc, mpd    ; ha megnyomva, mozgás lefelé
pop af         ; akku helyreállítása
rra           ; következő bit (8-es helyiérték) = 4-es gomb
call nc, mpu   ; ha megnyomva, mozgás felfelé

; Az átmozgatás megtörtént, újra megjeleníthetjük a játékost
call basexy    ; x és y koordináta beállítása
call splayr   ; játékos megjelenítése
halt          ; várakozás

; Visszaugrás a főhurok elejére
jp mloop

; Játékos balra mozgatása
mpl      ld hl, ply ; Emlékezzünk, y a vízszintes koordináta!
        ld a, (hl) ; Mi a mostani érték?
        and a      ; Nulla?
        ret z     ; Ha igen, nem tudunk tovább balra menni!
        dec (hl)  ; különben y = y-1
        ret

; Játékos jobbra mozgatása
mpr      ld hl, ply ; Emlékezzünk, y a vízszintes koordináta!
        ld a, (hl) ; Mi a mostani érték?
        cp 31     ; A jobb szélén vagyunk (31)?
        ret z     ; Ha igen, nem tudunk tovább jobbra menni!
        inc (hl)  ; különben y = y+1
        ret

; Játékos felfelé mozgatása
mpu      ld hl, plx ; Emlékezzünk, x a függőleges koordináta!
        ld a, (hl) ; Mi a mostani érték?
        cp 4      ; a pálya tetején vagyunk (4)?
        ret z     ; Ha igen, nem tudunk tovább felfelé menni!
        dec (hl)  ; különben x = x-1
        ret

; Játékos lefelé mozgatása
mpd      ld hl, plx ; Emlékezzünk, x a függőleges koordináta!
        ld a, (hl) ; Mi a mostani érték?
        cp 21     ; A képernyő alján vagyunk (21)?
        ret z     ; Ha igen, nem tudunk tovább lefelé menni!
        inc (hl)  ; különben x = x+1
        ret

; A játékos törzsének, x és y koordináta értékének beállítása, ez a rutin kerül meghívásra a törzs törlése és megjelenítése előtt
basexy  ld a, 22  ; AT pozicionáló kód
        rst 16
        ld a, (plx) ; játékos függőleges koord
        rst 16     ; beállítjuk
        ld a, (ply) ; játékos vízszintes koord
        rst 16     ; ezt is beállítjuk
        ret

; Megjelenítjük a játékost a jelenlegi PRINT pozícióban
splayr  ld a, 69  ; cián tinta (5) fekete papíron (0), világosan (64)
        ld (23695), a ; beállítjuk az ideiglenes színeket
        ld a, 144  ; 'A' UDG ASCII kódja
        rst 16     ; játékos kirajzolása
        ret

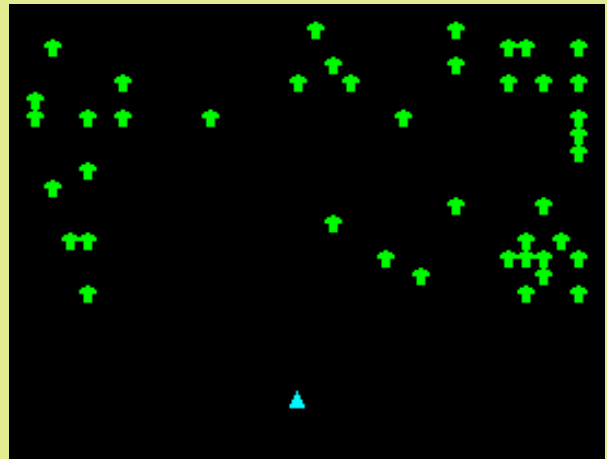
wspace  ld a, 71  ; fehér tinta (7) fekete papíron (0), világos (64)
        ld (23695), a ; beállítjuk az ideiglenes színeket
        ld a, 32   ; SZÖKÖZ karakter
        rst 16     ; üres hely megjelenítése
        ret

; Egyszerű pszeudo-véletlen szám generátor
; Egy mutatót léptet a ROM területen (a seed-ben tárolva), visszaadva
; a megcímzett bájt tartalmát
random  ld hl, (seed) ; Mutató
        ld a, h
        and 31       ; Az első 8 KB-on belül tartjuk
        ld h, a
        ld a, (hl)   ; „Véletlen” szám a mutatott helyről
        inc hl       ; Mutató léptetése
        ld (seed), hl
        ret

seed    defw 0
plx     defb 0 ; játékos x koordinátája
ply     defb 0 ; játékos y koordinátája

; UDG grafika
blocks  defb 16,16,56,56,124,124,254,254 ; játékos törzse
        defb 24,126,255,255,60,60,60,60 ; gomba

```



Ha lefuttatjuk a fent listázott programot láthatjuk, hogy már inkább hasonlít egy Százlábú játékra, mint előtte, ellenben van egy apró probléma: ugyan a gombok véletlenszerűen szét vannak szórva a képernyőn, a játékos akadálytalanul képes áthaladni rajtuk. Valamiféle ütközés észlelésre lenne szükség ennek a megakadályozásához. Ezzel fogunk foglalkozni a következő fejezetben.

Tanács Imre (Kapitány)

Folytatjuk ...

PROGRAMOZÁSTECHNIKA

SDK: SPECCYBEEP

A SpeccyBeep a gépi kódú ZX Spectrum programokhoz segít dallamokat szerkeszteni és azok beépítését is megkönnyíti.

A SpeccyBeep egy Windows-alapú dallamszerkesztő, melyben a szerkesztett dallamot `.tun` kiterjesztésű munkafájlba lehet menteni, ami lényegében egy szöveges dokumentum és soronként egy-egy hang adatait (oktáv, hang, hossz) tartalmazza. A SpeccyBeep nagy mutatványa, hogy a dallamot egy ZX Spectrum számára is emészthető `.asm` kiterjesztésű assembly fájlba is át alakíthatjuk, mely egy egyszerű lejátszó rutint is tartalmaz. Ezt fájl egy külső programmal (pl. PASMO) át alakítható `.tap` vagy `.tzx` formátumba, amit szintén a SpeccyBeep-ből vezérelve be is lehet tölteni egy külső ZX Spectrum emulátorba. Betöltés után a dallam a `RANDOMIZE USR 25000` parancs kiadásával indítható.

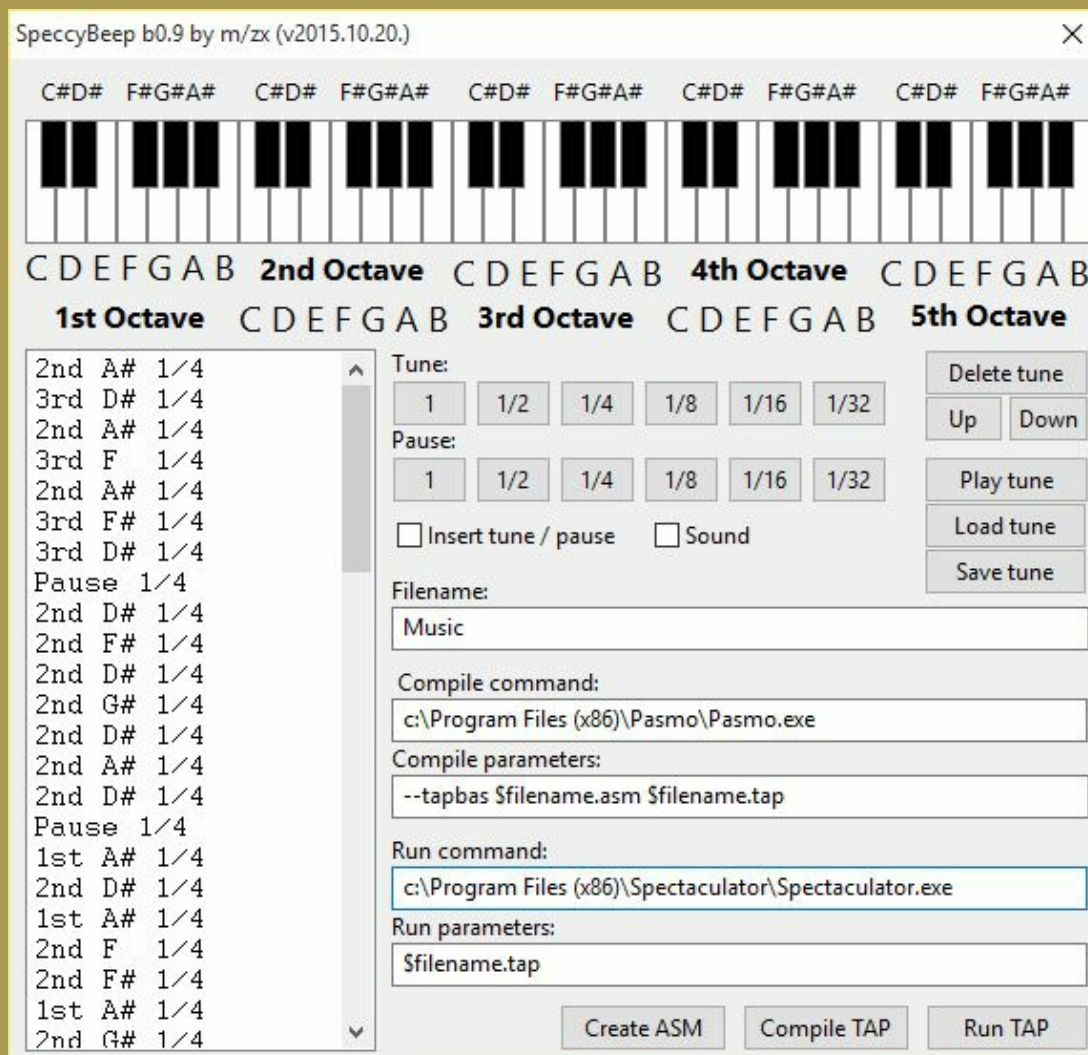
Sokat kutattam hasonló programok után, de vagy a kereső szavakkal vagy velem volt probléma, vagy a legvalószínűtlenebb: egyszerűen nincs ilyen program.

De minek is kellene nekem egy ilyen program, mikor egy `asm` fájl írásakor közvetlenül is megadhatom a hangokat? Megpróbáltam, szörnyű gyötrelmem minden egyes hangnál a következő képletekkel kiszámolni a magasságot és a hang hosszát:

$DE = \text{Hossz} = \text{Frekvencia} * \text{Másodpercek száma}$

$HL = \text{Magasság} = (437500 / \text{Frekvencia}) - 30.125$

Ez a módszer nem elég, hogy keserves rabszolgamunkával jár, de még a hiba lehetősége is nagyon magas. Első ilyen kínlódásom után gyártottam magamnak egy Excel táblát, amiben minden hangmagasságra és hanghosszakra kiszámoltattam DE-t és HL regiszterekbe töltendő értéket. Ezt használni már egy fokkal kényelmesebb volt, de még mindig túl körülményesnek és lassúnak tűnt egy táblázatból egyesével kibogarászni a megfelelő adatokat, pláne, hogy egy rövidebb dallamsor is kb. 30-40 hangot tartalmaz. Ekkor döntöttem el, hogy saját kényelmem és idegeim épségének megőrzése érdekében meg kell írnom a SpeccyBeep-et.



Dallam létrehozásához először válaszd ki a zongorabillentyűkön a hangot (piros lesz a billentyű), majd a 'Tune:' felirat alatt látható 1, 1/2, 1/4, 1/8. gombokkal válaszd ki a hang hosszát (a gombok felirata egyenlő a hang hosszával másodpercben). Ekkor a hang megjelenik a bal oldalon lévő listában, pl. 3rd E 1/2, ami annyit jelent, hogy a harmadik oktáv E hangja fél másodpercig szóljon. Ha kipipálsz a *Sound* opciót, akkor a zongorabillentyűk leütésekor és a hangok hosszának megadásakor is hallhatod a beállításoknak megfelelő hangot.

Ha a dallamba szünetet akarsz beiktatni, akkor a *Pause:* felirat alatt látható gombok valamelyikét kell használnod, ilyenkor a hangok mintájára a dallamsorba bekerül egy szünet, ami a gombon lévő feliratnak megfelelő másodpercig fog tartani.

Az *Insert tune / pause* kiválasztásával érheted el, hogy a kiválasztott hang vagy szünet ne a dallamsor végére, hanem a dallamlistában aktuálisan kijelölt pozícióba kerüljön. Hangok törléséhez jelöld ki a törölni kívánt hangot a dallamlistán, majd nyomd meg a *Delete tune* gombot. Hangok helyének megváltoztatásához használd az *Up* (fel) és *Down* (le) gombokat miután kijelölted a mozgatandó hangot a listán.

A *Play tune* lenyomásával meghallgathatod a dalt, ez nem pont úgy fog szólni, mint amit egy emulátor produkál, ez a funkció csak előhallgatásra szolgál. A *Load tune* ill. a *Save tune* gombokkal pedig betöltheted / lementheted a dallamlistát.

Ha meg akarod hallgatni a dallamot emulátoron, vagy akár egy valós gépen, akkor előbb meg kell adnod néhány dolgot:

- A *Filename* mezőben add meg a leendő ASM és TAP fájl nevét.

- A *Compile command* mezőben add meg a külső fordítóprogram elérési útját, pl. `c:\Program Files\Pasmo\Pasmo.exe`.

- A *Compile parameters* mezőben a fordítóhoz tartozó paramétereket adhatod meg, itt *\$filename*-t kell írni a szerkesztett fájl neve helyett, pl. a Pasmo-hoz tartozó paraméter lehet: `--tapbas $filename.asm $filename.tap`, ilyenkor a fordító létrehoz egy *.tap* fájlt.

- A *Run command* mezőben a már átfordított fájl futtatásához használt program elérési útját adhatod meg, pl.: `c:\Program Files\Spectaculator\Spectaculator.exe`.

- A *Run parameters* pedig értelemszerűen a futtatáshoz használt program futtatási paramétereit kell, hogy tartalmazza, pl. *\$filename.tap*.

Ha az adatokat megadtad, akkor a *Create ASM* gombbal létrehozhatod az *.asm* fájlt a *filename* mezőben megadott néven, a *Compile TAP* gombbal átfordíthatod egy ZX Spectrum számára is


emészthető fájl, a *Run TAP* gombbal pedig futtathatod ezt az emészthető, valószínűleg *.tap* vagy *.txz* fájlt.

Fejlesztési lehetőségek még vannak a programban, erre utal a verziószám is. Például benne lesz az oktáv emelése, csökkentése teljes dallamsorra vonatkozóan, vagy a dallam gyorsítása, lassítása is. Viszont nem lesz benne lehetőség többszólamú dallam létrehozására, marad a „bípelés” hangzás.

Letöltés: <http://tanzi.hu/sbhun.htm>

Mezei Róbert (M/ZX)

PÓKOK



Snake-Pit

Unalmas óráimban sikerült a Snake-Pitet, egyik kedvenc kis programomat visszafejteni. Érdekes volt fejtegetni, igazából azért kezdtem bele, mert mindig zavart a piros kígyó pimaszsága. Meg aztán elég rövidke program, így még jegyzetelni se volt szükséges.

Ha valaki szeret játszogatni velem, akkor ezt a két POKE-t feltétlenül próbálja ki:

POKE 27446,0	könnyű fokozat
Előző +	
POKE 27447,202	extra nehéz fokozat.

Pgyuri

Save the Trees

Nem sokkal a játék megjelenése után adták közre ezt a kis örökélet poke-ot a WOS-on.

Örökélet:

27937, x	x = életek száma
----------	------------------

PROGRAMOZÁSTECHNIKA

HOGYAN KÉSZÜLT AZ ISHIDO: THE WAY OF STONES

Avagy a kezdő programozó buktatói

Megtévesztő a cím, mert nem az a lényeg, hogy hogyan készült el az Ishido, hanem az, hogy hogyan jutottam el odáig, hogy elkészülhesen. Ez az íromány azoknak szól leginkább, akik ZX Spectrumra való programfejlesztésben törik a fejüket, de a sok buktató már elriasztotta őket. Másrészt azoknak, akik nem tudják hol elkezdni a dolgot, harmadrészt a többieknek, akik csak kíváncsiak kínládásaim sorára. Aki már haladóbb a gépi kódú programozásban, az nem fog semmi újat tanulni a leírtakból.

Előzmények

Ha az elejéről akarom kezdeni, és miért is ne akarnám ezt tenni, akkor elég sokat kell visszabaktatnom az időben. A történet mindkét szála a nyolcvanas évek végére vezet. Első és sokáig jelentéktelen szálként megemlítem, hogy nevezett időszakban barátkoztam a Spectrum gépi kódú programozásával. Második szálként meg, lényegesebb szálként kell szólnom arról, hogy a címszereplő játék egy konverzió, Michael Feinberg 1989-es, Macintoshra megjelent játéka a Spectrum átírata. Nem mintha valaha játszottam volna az eredeti játékkal, csak a DOS-os változatot ismertem, amit az Accolade forgalmazott 1990-től.

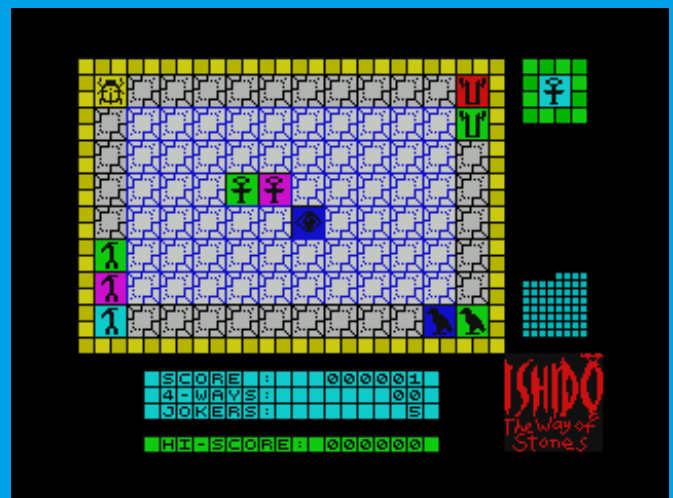
1992 körül, mikor már jó ideje meghalt a 48K-s Spectrumunk, a család számítógépe, a szüleim vásároltak egy 386-os PC-t meg egy monochrom Philips monitort. A gépnek volt néhány elég lenyűgöző tulajdonsága, mint a 80 megás HDD, a memóriával bővíthető Trident videokártya. Később került bele egy Gravis Ultrasound hangkártya is, ami akkoriban elég spécinek számított.

Ködösek az emlékeim, de az Ishido az első értelmes játékok között lehetett, amit játszottam a gépen. Mindig is könnyen beszippantottak a hasonló logikai játékok, mert a szabályok nincsenek túlbonyolítva, könnyen megérthetőek, de jut bennük egy kis mellékszerep a szerencsének is, éppen csak megfűszerezve a játékot. Egyszerűségük ellenére összetett, változatos, élvezetes a játékmenet, folyamatosan lehet bennük fejlődni, először viszonylag gyorsan, később egyre nehezebben, a tudásszintnek megfelelően mindig új célokat lehet bennük találni, elérni, tehát az én értelmezésemben optimális a nehézségi szintjük. Sokat játszottam a játékkal, egyre magasabb pontszámokat értem el, aztán nyilván jött valami újabb játék, ami kiütötte a nyeregből és szépen lassan elfelejttem.

Második találkozás

De nem sokáig maradt feledésben, 1995-ben szereztem programozói végzettséget, aminek a végén bizony le kell rakni valami munkát az asztalra, részemről (egy barátommal karöltve) ez a „Logikai játékok számítógépes megvalósítása” címet viselte. Ebbe beletartozott egy nagy csomó assembly rutin (főleg grafikai), meg négy játék megvalósítása a rutinok segítségével. Egyik ilyen játék volt az Ishido. Ekkor kellett végiggondolnom először, hogy milyen elemekből építhető fel a játék, milyen eljárások, függvények, változók, stb. kellenek a játék elkészítéséhez. Tehát mondhatnám, hogy ekkor kezdtem el a játék Spectrumra való átírását, ami igaz is, mivel az ekkor elkészített Pascal nyelvű program jó támpont lett a Spectrum változathoz. De az igazság inkább az, hogy akkoriban egy másodpercre sem fordult meg a fejemben a játék átkonvertálása.

Elkelne egy Spectrum verzió is



Teltek az évek, belefogtam mindenféle dolgokba, néha azok fogtak ki rajtam, valamikor már a XXI. században eszembe jutott az Ishido, főleg az, hogy de jó lenne játszani vele ZX Spectrumon. Keresgéltem is kezdtem a World of Spectrum végtelennek tűnő játékadatbázisában, de nem akadtam semmiféle konverzió nyomára. Nem emlékszem semmilyen elhatározásra, de arra igen, hogy elkezdtem papírra rajzolgatni mindenféle figurákat, képernyőterveket, számolgatni, hogy mekkora lehetne egy mező, szóval mindenféle játékkal kapcsolatos dolgokat. Akkor azt gondoltam, hogy a játékot meg lehetne Basic-ben is írni, mert nincs benne olyan rész, ahol a gyorsabb assembly megoldások kellenének. Vacakoltam még vele, de nem akaródzott a programozás semmilyen nyelven sem. Még az is megfordult a fejemben, hogy

Hisoft Pascalban kellene megírni, mivel régóta porosodott már egy Pascal kódom. Erről a megoldásról is gyorsan lemondtam, leginkább kényelmi okokból, miután elindítottam a Hisoft Pascalt, nem nagyon nyerte el a tetszésem.

Megint elmúlt néhány év (fogalmam sincs, hogy mennyi), aztán elkezdtem a játék megírását, de úgy kezdtem, hogy nem is akartam igazán elkezdni, sőt, nem is tudtam, hogy elkezdtem. Akkor készültünk a Speccyalista Világ első számának megjelentetésére, és ezzel kapcsolatban olvastam Kapitány fordítását a *Hogyan írjunk játékot ZX Spectrumra...* első három részét a Sinclair.hu-n. Arra jutottam, hogy nem csak olvasni kellene, hanem ki is próbálni, hátha ragad rám belőle valami.

Megfelelő munkakörnyezet kialakításához vezető hepehupás út

Na, akkor majdnem abba is hagytam, mert fogalmam sem volt, hogy mibe írjam be a példákat, konkrétan már azon is fennakadtam, hogy milyen programot is kellene használnom erre a célra. Jó egy sima szövegszerkesztő is? Vagy kell hozzá valami spéci program? Előbbi mellett döntöttem, beírtam egyszerű Windows-os jegyzetömbbe, de adódott a következő kérdés: mit csináljak vele? Oké, hogy 'asm' legyen a kiterjesztés, de mi legyen a következő lépés? Hogy lesz ebből valami olyan, amit be lehet tölteni egy Spectrumba?

Kezdtém nézegetni a WOS fórumát, de az angol tudásom olyan, mint egy általános iskolásé, szóval szenvedtem rendesen mindenféle keresőszavakkal. Már nem is tudom kinek találtam meg a posztját, a lényeg, hogy a Crimson Editort ajánlotta, ráadásul még a beállításokat is leírta, hogy mit kell csinálni ahhoz, hogy a begépelt "programok" átváltozzanak betölthető, valódi programokká. Na meg még azt is leírta, hogy kell a PASMO nevű program is ehhez a metamorfózishoz, futtatáshoz meg nyilván egy emulátor.

A PASMO nevét olvasva viszonylag gyorsan rájöttem, hogy ez egy fordító program, de azt még nem tudtam, hogy mire fordít. Így már kezdtem előrébb jutni, kezdtem bepötyögni az olvasottakat, meg módosítottam, hogy rájöjjek, hogy mi micsoda. A programpéldáknál a magyarázatok alapvetően nagyon jók és könnyen értelmezhetőek. Pár napig tartott ez az ismerkedési időszak, utána már arra gondoltam, hogy ideje mélyíteni a kapcsolatomat a gépi kóddal. Gyakorlatban ez úgy jött, hogy már nem elégitett ki a példák egyszerűsége, kezdtem agyalni, hogy milyen konkrét dolgon tudnám az olvasottakat próbálgatni.

Akkor jutott eszembe az Ishido, amit jó pár évvel ezelőtt már tervezgettem. Szóval nézegettem Kapitány fordítását, nagyjából értettem is, pontosabban

derengett valami, hogy előző életemben én már valami ilyesmit műveltem... Ja, közben rádöbbenem, hogy ez a PASMO mekkora egy király program, egyből TAP-ot gyárt nekem a begépelt kis irományokból. Mikor a fontokról volt szó, eszembe jutott, hogy mindig is szerettem volna valami saját fontot kreálni, fel is használtam gyorsan Mopi font kreátorát és megcsináltam a Vortex játékoknál is használt fonthoz hasonló fontomat. Közben megkerestem az évekkel korábbi Ishido terveimet, rajzolgattam mindenféle figurákat kockás, vagyis négyzetrácsos papírra. Gondoltam az jó is lesz arra, hogy saját UDG-eket csináljak. Addigra már megtaláltam a SevenUp-ot is, abban rajzolgattam a figurákat. Le is mentettem őket 'sev' formátumban (nem vettem észre, hogy a program eleve tud többféle adatfájlba exportálni, pl. 'asm'-be), majd rájöttem, hogy nem oké a dolog, mert nem tudom felhasználni a rajzaimat. Szerencsére elég egyszerű a formátum, összedobtam hozzá egy sev-ből asm-et csináló konvertert (persze utólag kiderült, hogy tök feleslegesen, de nem baj :).

Összefoglalva: abban a pillanatban próbálgattam a "Hogyan írjunk játékot ZX Spectrumra..." első három fejezetében tanultakat, mondhatom, hogy többnyire értettem is, hogy mi mire van hatással, és hasonlók. Hagytam, hogy a PASMO a háttérben megoldja a problémáimat és létre tudtam hozni rajzokból 'asm' fájlokat. Sajnos azt nem tudtam, pontosabban nem jutott eszembe, hogy az 'asm' fájlokban található 'defb' kezdetű sorokat átmásolhatnám magába a programot is tartalmazó 'asm' fájlba is, így jött a következő probléma: valahogy a fontot meg a grafikákat is a Spectrum memóriájába kell juttatnom. Két nagyon egyszerű megoldás is van, mégis sok idegsejtem látta kárát a megoldás keresésének. Találtam programot, ami 'asm'-ből 'bin'-t csinál, majd egy másik programot, ami 'bin'-ből 'tap'-ot. Hurrá, már be is tudtam tölteni a fontomat. Teljesen le voltam nyugözve, hogy ezzel ki is tudok bármit írni a képernyőre :) Aztán meg az Ishidos figurák is megjelentek különböző színekben, maga volt a tökély. Akkor ugrott be a kettes számú, kevésbé egyszerű megoldás: betöltöm pl. a font.asm fájlt a Crimson Editorba, aztán lefordítom a PASMO-val, aztán már be is tudom tölteni programként egy emulátorba. Ez már félig-meddig praktikus volt, csak annyi baj volt vele, hogy először be kellett töltenem a memóriába a fontot, aztán a grafikákat, aztán a programot, tehát kicsit macerás volt így az élet. Csak hogy ne rögzüljön másban ez a suta megoldás, elárulom, hogy jobb megoldás lett volna, ha a font.asm teljes tartalmát bemásoltam volna a programom végére, az első sorának elejére pedig 'Label'-ként beírtam volna, hogy 'FONT'. Ugyanígy eljárva a grafikával is egyből betölthetővé válik adat meg program is.

Mezei Róbert (M/ZX)

Folytatjuk...

ISHIDO

The Way of Stones

2007 óta az
első magyar
ZX Spectrum
játék!



**”EGYSZERŰEN ISHIDO
KELL MINDENKINEK!”**

(Amiga Joker, 1990. december)

**”SZUPER! NEM LEHET
SZABADULNI TŐLE!”**

(Power Play, 1989. szeptember)

”EZ EGY NAGYON, NAGYON JÓ JÁTÉK... ÉS ELFÉR 16K-BAN.”

(Pegaz, WOS fórum, 2015. október)

**”SZERETEM! NAGYON BOLDOG VAGYOK, HOGY VALAKI
SPECTRUMRA KONVERTÁLTA - RÁADÁSUL ILYEN JÓL!”**

(retromad, WOS fórum, 2015. október)

”NAGYON JÓL SIKERÜLT KONVERZIÓ - GYÁRI MINŐSÉG”

(csory, Speccyalista Kerekasztal, 2015. október)

”FANTASZTIKUS! NEMCSAK A JÁTÉK, HANEM A KÖRÍTÉS IS”

(Zozosoft, Speccyalista Kerekasztal, 2015. október)

INGYENESEN LETÖLTHETŐ!



www.fanzix.hu



SCENE ESSZENCIA

MESCALINE SYNESTHESIA



Alkotó: deMarche

Kategória: Demó

Party: the Ultimate Meeting 2009

Helyezés: 1.

Vélemény:

Az egész demó egy 3,5 perces LSD-s hallucináció, színes, pörgő, kavargó formákkal. Pontosabban meszkalin ez a drog, amit már az aztékok is használtak szertartásaikon. A bemutatón, 2009-ben, nagy kivetítőn még nagyobb hatású lehetett, mint otthon megnézve, pedig így sem semmi. A zene végig jól illeszkedik a látványhoz, és ugyanolyan elborult a hangzás, mint a képi világ. Még jobban értékeli az ember, mikor rájön, hogy sima, 128k-s mű, nem az oroszoktól megszokott 640k-s TR-DOS-os monstrum.

A zenéről vitatkoztak, néhányan a fórumon, azt mondták, hogy rippelt. Az biztos, hogy a Chaos Construction 2001-en indított Jaundice című demónak is ez volt a zenéje, amit remixeltek a mescaline synesthesiához. Nem szép dolog korábban megjelent stuffot újrahaznosítani, de azt kell mondjam, ebben a demóban jobban működik ez a zene, mint az eredetiben.

A demó végig gigascreent használ. Ez igazi hardveren villószik, ami zavaró, bár ha akarja az ember, ráfoghatja, hogy az is a haluzás része. Emulátorban ki lehet kapcsolni a villogást, és érdemes is, mert nem jól jelenítik meg a gigascreen-t, mivel nem 50 Hz-es visszatérítésű monitorokat használunk (általában). A demó nagy része 4*4-es módban fut, azaz 64*48-as felbontásban, de ott is 2 képet cserélgetve, 64 színt kezel.

A demó egy graffiti-szerű képpel indul, ami jó graffiti-hez méltóan olvashatatlan, de szép színes cserébe. Utána elindul a kavalkád. Az érdekesség, hogy a

speciális képernyő mód miatt kevesebb processzoridőt tud használni a kóder, mivel a képernyő kb. felén, ahol az effektek megjelennek, csak a kép kirakásával kell foglalkozni. Marad a felső és az alsó kb. 1/4 kép, és a visszatérítés alatti idő. Ehhez képest van itt minden, mi szem-szájnak ingere.

Érdekes effektek:

- Tunnel mapping
- 3D hullám scroll
- Roto-zoomer
- Spirális cső
- Bump mapping mozgó fényforrással
- Sík rubber band golyók (kék gumibogyók)
- Forgó plazma

Értékelés:

Code (by TmK): 5

Gfx (by TmK): 5

Mzx (by MmmC): 5

Design: 4 (de csak azért, mert végig gigascreent használ)

Ajánló: deMarche stuffok, amit még érdemes megnézni: paralactika, Scroller, make demos not 1k intros

Emulátor: Unreal Speccy. Benne van a csomagban, bekonfigurálva.

Videó:

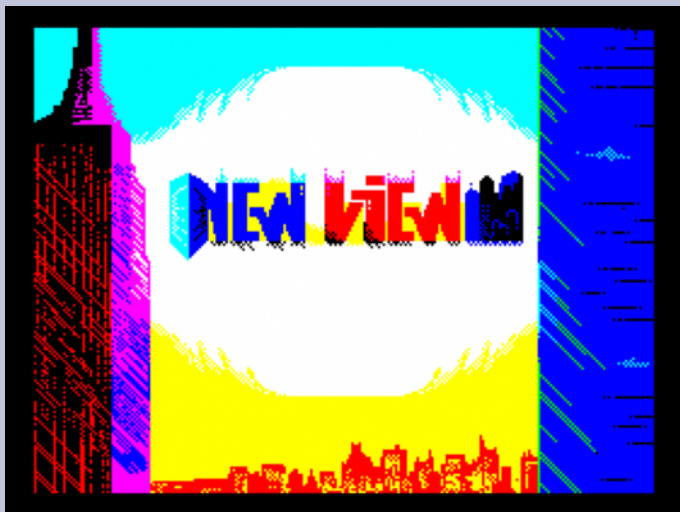
https://www.youtube.com/watch?v=x1o_Tr8jq3I

G.o.D. / AbaddoN



SCENE ESSZENCIA

NEW VIEW 48K



Alkotó: Conscience
Kategória: Demó
Party: Forever 2015
Helyezés: 2.

Vélemény:

A mescaline synesthesiával ellentétben, ez a demó nem trükközik, izomból tolja az effekteket, így talán jobban érezhető kevésbé fanatikusok számára is. Kis túlzással ráfogták, hogy 48k-s, de a zene AY, ezért egy AY extender biztos, hogy kell a gépre. Összesen 73,5 kB-ban fért el, van 4 utántöltős rész. Emulátorban észre sem veszi az ember, de magnót használva, igazi gépen, ezek a töltőgetések tönkreteszik a demó egységét.

Egy jópofa morfolással indít. Utána egy Test Drive-ot idéző autózás következik, majd pár egész képernyős színes scroll, aztán jön a killer effekt: gouraud árnyalt kocka, és egy színes gyémánt formájú test forgatása. Az utóbbinál a Speccy színkezeléséből adódóan szépen megfigyelhetjük a color clash-t is. Levezetés-ként egy mars effektet láthatunk, és egy nem tudom

mit, de a lényege, hogy 2D-ben forgatja a poligonokat. A végén kapunk még egy multicolor képet.

A zene stimmel a demóhoz. A grafika és a dizájn elmegy. A kód viszont igencsak rendben van, Alone-Coder nem ma kezdte. Idézet az egyik hozzászólásból: *"sure, graphically it might be not the best, but damn is it coderporn at its finest"*. (Magyarul: persze, grafikailag talán nem a legjobb, de a nyavajába is, a kódpornó legjobbja.)

Érdekes effektek:

- Real-time morfolás
- Gouraud árnyalt kocka
- Színes 3D forgatás
- Színes mars effekt

Értékelés:

Code (by Alone_Coder, TK90X Fan):	5
Gfx (by trixs, Hippiman, nitrofurano, Louisa):	4
Mzx (by n1k-o):	5
Design (by Alone_Coder):	3

Ajánló: Conscience stuffok, amit még érdemes megnézni: Mission Highly Improbable, 9 Channels

Emulátor: ZX Spin (48k+AY, de egyszerűbb, ha 128k módba teszed)

Videó:

<https://www.youtube.com/watch?v=dCyo1zhKjoM>

G.o.D. / AbaddoN



HARDVER SIMOGATÓ

ZX81: ZXPAND - EXPANDER RETRO MÓDRA

A ZXpand – melyet Charlie Robson „követett el” az RWAP Software számára – a ZX81 harmincadik születésnapjának méltó megünneplésére készült.

Ez a küttyü nem más, mint egy SD kártya illesztő, amiről könnyedén tölthetünk be .p fájlokat, FAT- kompatibilis és az alkönyvtárak használatát is megengedi. Mindemellett 32K memóriával is rendelkezik, mely konfigurálhatóan többféle nagyfelbontású módot is

Kiegészíthető a ZXpand AY hangmodullal is, mely AY-3 8910-at használ a hangkeltéshez és Zon-X kompatibilis. Optionálisan 9 tűs Atari típusú botkormány csatlakozót is ráberhelhetünk, bár a ZX81 programok joystick támogatottsága eléggé gyér.

Bár nem a harmincadikra, de én is születésnapomra leptem meg magamat vele, ezúton szeretném felhívni a figyelmeteket rá és megosztani a tapasztalataimat. Egy olyan speciális változatot rendeltem, melyen a 32K RAM leltítható, hogy meg lehessen kínálni hátulról például egy 64K RAM pakkal. Ezt alapvetően amiatt kértem, hogy a MUZIX81 audio processzorral is lehessen majd használni, az pedig kívánja a sok memóriát. Most nyilván engedélyeztem a belső 32K RAM-ját, el is indult és ezzel teszteltem.

Működés

Manapság mondhatnánk, hogy Plug'n play, mert csak csatlakoztatnunk kell és máris használható. Induláskor rögvést kiírja a képernyőre, hogy: ZXPand 1.x. A FAT kompatibilis SD kártyákat kezel, és egyaránt szereti a FAT12, FAT16 és FAT32-es formátumokat. Az alkönyvtárak használatát is támogatja, de nemcsak lépegethetünk közöttük, hanem ZX81-en is készíthetünk alkönyvtárakat. A szabványos .p fájl formátumot kedveli és az összes nagyfelbontást használó programot is futtathatjuk.

Az eszközt kezelő parancsokat a ROM-ban a printert maceráló utasítások helyére tették, amúgy az új parancsok implementációja az eszközön található árnyék-ROM-ban található.

A LLIST helyén egy CONFIG parancsot kapunk, amelyhez az alábbi paraméterek adhatók:

"V" - aktuális verzió kiírása
"R=nnnnn" - RAMTOP beállítása, NEW kiadása
"M=L" - 'Low' vagy 8-40K RAM mapping beállítása
"M=H" - 'High' vagy 16-48K RAM mapping beállítása
"M" - aktuális RAM mapping kiírása
"D=útvonal" - aktuális könyvtár kijelölése
"D=" - gyökérkönyvtár beállítása, ua mint a '/'
"X" - shadow ROM kikapcsolása a következő resetig

A CAT parancs a COPY helyére költözött be, így a Z lenyomására megjelenik a CAT, és kártya hiányában 3/0-val bevégez, egyébként nyilván a kártya katalógusával tér vissza.

Adott alkönyvtár tartalmát így listázhatjuk ki:

```
CONFIG "D=GAMES"  
CAT
```

Használható az alábbi forma is:

```
CAT "..../GAMES"
```

Egy Nokia 6600-ból zsákmányolt 32M-es SD kártyát dedikáltam erre a feladatra. A rámásolt .p fájlokat rendben be tudtam tölteni és futtatni, de a hosszú fájlnevekkel már nem birkózott meg a ZXPand, sőt egyáltalán nem lehetett betölteni ilyen fájlokat, csak és kizárólag a standard 8.3-as elnevezést komálja. Ennek el- lenére a CAT azért mutatja ezeket is.

Egy program betöltése SD kártyáról, ami mondjuk a HIRES könyvtárban található, az alábbiak szerint néz ki:

```
CONFIG "D=HIRES"  
LOAD "CRUSH"
```

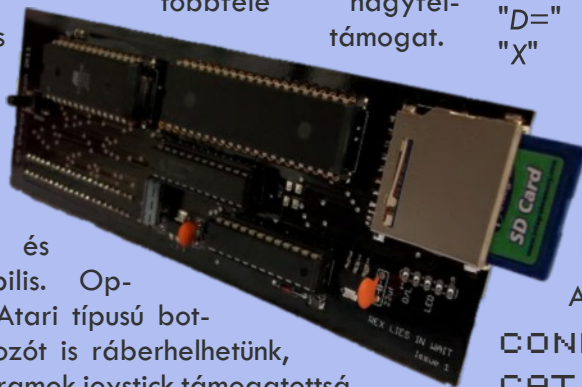
A LOAD parancsnak azonban vannak új funkciói is. Előfordulhat, hogy programunkat adott memóriacímre szeretnénk betölteni, ekkor használjuk ezt a formátumot:

```
LOAD "PROGRAM;{mem.cím}"
```

Ugyanakkor lehetnek olyan programok, melyek használják a printer rutinokat vagy a karakter táblát a ROM-ban, ilyenkor az ezzel a formával a program betöltése után lekapcsolhatjuk a ZXpand árnyék ROM-ját egy következő resetig:

```
LOAD "PROGRAM;X"
```

Kazettáról továbbra is tölthetünk be a sima LOAD "" parancssal. A normál SAVE parancshoz képest a



ZXpand SAVE-je rendelkezik egy új lehetőséggel, egy adott memóriadarabot is lementhetünk, ssss címtől, hhhh bájt hosszan:

```
SAVE "PROGRAM; ssss , hhhh "
```

Kazettára csak a CONFIG "X" parancs kiadása után tudunk menteni.

A DELETE parancs is kakukk módjára az LPRINT helyén húzta meg magát. A LOAD-SAVE esetében nem kell bajlódni a .p kiterjesztéssel, azonban törléskor a teljes fájlnévet meg kell adni:

```
DELETE "fájlnév.p"
```

Firmware / ROM upgrade

ZX81-es körökben abszolút élő eszokról beszélhetünk, rengeteg fórum bejegyzés foglalkozik vele és rendszeresen lehet firmware és ROM frissítésekhez is jutni.

Az én eszközem egyébként egy 5.1-es verzióra hallgató ROM-ot és egy 1.3 firmware-t tartalmazott, amiről hamar kiderült, hogy nem a legfrissebb párosítás, így legalább ezt is kipróbálhattam.

A firmware frissítését igencsak könnyű elvégezni, mert az SD kártyáról betölthető. A letöltött firmware képállományt kicsomagolva kell rámásolni egy frissen formázott SD kártyára. Ezt helyezzük be az illesztőbe és hard reset hatására elkezdődik az upgrade, kigyullad a piros led és villog a zöld, és ha megtörténik az upgrade, akkor a ZX81 resetel. Különösebb erőfeszítés nélkül sikerült is frissítenem a legújabb verzióra a kis ZXpandomat, amit elég könnyű ellenőrizni, miután induláskor ezzel jelentkezik be.

A ROM frissítéshez vagy egy EPROM-égetőre lesz szükségünk, vagy rendelhetünk az RWAP-tól friss EPROM-ot, amit csak be kell dugni az illesztőnkbe. Mindenesre érdemes először ellenőrizni, hogy a miénk mit tartalmaz, ezt az alábbi utasítással tehetjük meg:

```
PRINT PEEK 7679
```

Erre a válasz egy kétjegyű szám lesz, pl. 51, ami az 5.1-es verziószámot jelenti. Az épp aktuális ROM verzió 0.6, így én beégettem ezt az én EPROM-omba.

Dobozolás

Az illesztőt úgy tervezték, hogy alapeletben pont belepasszoljon egy MEMOPAK dobozába, csak egy apró részt kell ejteni a doboz oldalán az SD kártyának.

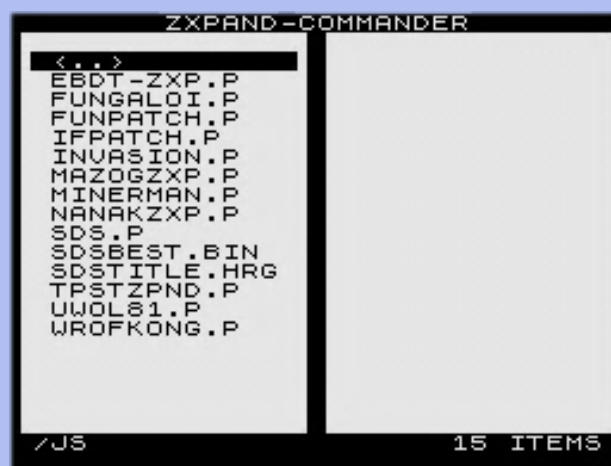


Emuláció

Az EightyOne emulátorhoz készült egy kiegészítés, mellyel kipróbálható és használható emulátorban a ZXpand, ezt az EightyOne X 1.0 X3.7-es verziója már tudja. A programokat az \MMC_CARD mappában kell elhelyeznünk. A ZXpand emulációhoz válasszuk ki a Hardware menüt [F6], az Interfaces-nél pipáljuk ki a ZXpand-ot, a Drives alatt pedig az IDE-nél válasszuk ki az MWCFIde-t.

ZXPand-Commander

Nagyon korrekt Márton kommanderszerű fájl menedzsert is készítettek hozzá, mely a ZXpand-Commander névre hallgat és jelenleg az 1.8-as a legfrissebb verziója. A program teljes funkcionalitásának kihasználásához érdemes a 2.2 firmware-re frissíteni.



Az alábbi funkciókkal rendelkezik:

- könyvtárak tallózása
- programok futtatása
- alkönyvtárak készítése
- szövegfájlok megjelenítése
- monokróm BMP fájlok megjelenítése
- fájlok és üres alkönyvtárak törlése
- fájlok és alkönyvtárak átnevezése
- fájlok másolása és mozgatása

Minden helyzetben a [SHIFT+H] billentyűkombinációra érhető el a HELP.

A program teljes forrása is hozzáférhető, de a fejlesztő kéri, hogy az esetleges javításokat, újdonságokat küldjük el neki, hogy azok a hivatalos verzióba is bekerülhessenek.

Kardos Balázs (Balee)

HARDVER SIMOGATÓ

HUNGARIKUM: TZX MOUSE ÉS JOYTAPE PLAYER

Ezúttal nem egerről és botkormányról esik szó, hanem egy a magnót helyettesítő készülékről, egy egyszerű és egy botkormányos csomagolásban.

TZX Mouse

A MegaBit-X elnevezésű 2010-es Speccyalista napon mutatta be Gondos Csaba nekünk TZX lejátszóját, amely segítségével magnó nélkül tölthetők be programok ZX Spectrumra. Ez a remek találmány azon az elven alapszik, hogy belső memóriájába előre, meghatározott módon feltöltött TZX formátumban szereplő programokat tárol, amelyeket gombnyomásra lejátszik. Az eszköz a magnó csatlakozóján keresztül közvetlenül kapcsolódik a ZX Spectrumhoz, így egyszerűen a LOAD "" paranccsal az eredeti módon (bejelentkező képpel, egyedi loaderrel stb.) betölthető bármely program.

Valójában Csaba ezt már az előző évben kifejlesztette, amelyet az akkor sajnálatos módon elmaradt rendezvényen nem tudott bemutatni, azonban szerencsére egy kellemes ZX vacsorára elhozta magával a prototípust, amelyre azonnal felfigyeltünk. A garázkapu-nyitóba szerelt aprócska, belső hangszóróval is bíró eszköz kellemes csipogása magával ragadó volt, még Spectrum-mmentes környezetben is.

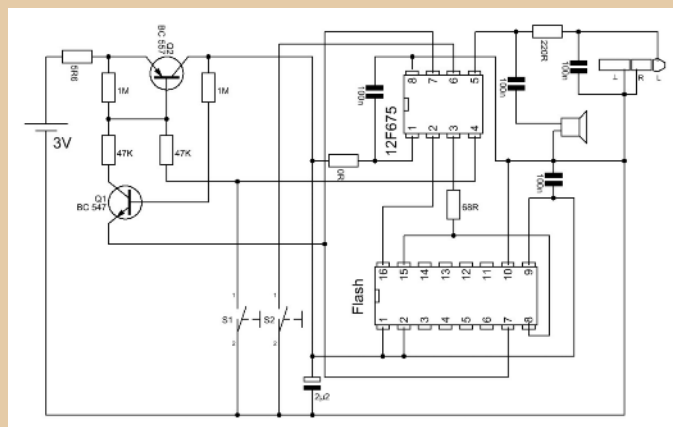
Elképzelés

Korábban a Spectrum háttértárak fejlesztése során legfontosabb szempont volt a minél nagyobb adatátviteli sebesség elérése. Esetünkben azonban egy olyan lejátszó kifejlesztése volt a cél, mellyel korhűen élhetjük át a kazettás programbetöltést, de mégis annál jóval megbízhatóbban, esetleg plusz kényelmi szolgáltatásokkal fűszerezve.

Manapság az emulátor formátumok közül a TZX típus az, amely legjobban megőrzi az eredeti betöltési eljárásokat. Ráadásul a programok szinte kivétel nélkül hozzáférhetők ebben a formátumban. Ezek után már kézenfekvő volt, hogy egy olyan eszközt kell készíteni, amely TZX fájlt képes lejátszani, ami a régi kazettás magnetofon jeleit állítja elő.

Hardver

Az elektronika tervezésénél a minimalista szemlélet volt az irányadó, a legegyszerűbb, legkevesebb alkatrészből, ugyanakkor még jól használható szolgáltatásokkal.



A kapcsolat főbb részei az 1K program és 64bájit RAM memóriával rendelkező PIC12F675 mikrokontroller, SPI buszos flash memória, és a tápfeszültség (elem) kapcsoló áramkör.

Ebben az első verzióban a lejátszó egy programgyűjteményt tartalmaz, mely a felhasználók által nem lesz változtatható.

A mikrokontrollerben futó saját fejlesztésű program valósítja meg a TZX kódolás visszaalakítását hangfrekvenciás jelekké. A memória 3 vezetéken (chip select, clock, data in/out) csatlakozik a kontrollerhez, a két változatban 64 és 8Mbit-es IC-k működnek. A mikrovezérlő kimenete jel-formálás után már a Spectrum EAR bemenetére csatlakozik, akár a kazettás magnó. Az eszközben található miniatűr hangszóró csak a retró hangzást szolgálja. A flash memóriába az összeszerelés előtt kerül IC-programozóval beírásra a TZX szerkesztővel összeállított fájl.

Működés

Az egyik nyomógomb a tápfeszültség kapcsolását is végzi az áramkörben, majd a másik nyomógombbal a lejátszandó program kiválasztását végezhetjük. Ezt többféleképpen végezhetjük.

Lépegethetünk egyenként előre vagy választhatjuk a 8 bites bináris beírási módot, ilyenkor a két gomb segítségével írhatjuk be binárisan a betöltendő program sorszámát, miután az első menüprogram betöltődött. A betöltés végén a lejátszó a következő gombnyomásig kikapcsol, miután saját memóriájába elmentette a programszámláló állását. A precízebb, tisztább jeleknek köszönhetően, a TZX időzítések szerinti akár négyszeres turbó sebességű betöltési módot is használhatunk.



Megjelenés

A lejátszó megvalósítása során a legnagyobb kihívást, mint mindig, a dobozolás jelentette. Ez az első variációban egy üresen, újonnan is beszerezhető "garázkapu távirányító" dobozkája volt. Erről kapta a lejátszócska a TZX Mouse elnevezést.

Menü

A TZX Mouse-ban végül a Speccyalista TOP 100 kapott helyet és tetszetős menü is készült hozzá, melynek segítségével könnyedén kiválasztható a betöltendő program. Ebben az esetben a menü füttyüli el binárisan a kiválasztott program sorszámát az ENTER lenyomására a lejátszónak.



A grafikát ismét Molnár Péter (Mopi) neve fémjelzi. A kis menücske a „pole” pozícióból várja a rajtot. A még gyorsabb betöltés érdekében Tom-Cat Z802TZX segítségével turbósítottunk is.

JoyTape player

Mint utóbb kiderült, a vacsorán látottak a WSS fejlesztőinek érdeklődését is felkeltették, és a TZX Mouse projekttel párhuzamosan izgalmas vállalkozásba kezdtek. A termékben rejlő lehetőségeket ők is azonnal meglátták, de számukra a zord külseje riasztó volt. Töprengtek hát, hogy mibe lehetne rejteni, egy este körvonalazódni kezdett egy aprócska joystick képében, amely megjelenésében sokkal inkább illeni látszott a kedves, kis ZX Spectrumjainkhoz, mint a kis távirányító dobozka. Hamarosan titkos projekt is lett az elképzelésből.



Csaba kérésükre újratervezte és egyedileg átalakította a WSS számára az elektronikát és annak belső programját, miközben megkezdődött a mini joystick-

ok "sorozatgyártása". A kis szerkezet ezen megjelenése után a JoyTape Player nevet kapta a keresztiségben.

Ahogy később Pgyuri mesélte, a MegaBit-X váratlan, egy héttel előre hozatalával hatalmas kihívás elé állítottuk a kis csapatot, így csak éjszakákba nyúló, megfeszített munkával sikerülhetett az utolsó pillanatokra elkészülniük, hogy a rendezvényen bemutat-hassák a kész JoyTape Player terméket a Speccyalista társaságnak, amelyben helyet kapott a WSS teljes játéksomagja, természetesen pár apró meg-lepetéssel fűszerezve.

A rendezvény csúcspontján rántották le a leplet az



aktatásokról, mely a kedves kis botkormányokat rejtette. Az igényes csomagolásra sem lehetett panasza, ahogy azt korábban is megszokhattuk a WSS-es fiúktól.



Hatalmas ováció fogadta a termék bejelentését és kisebb tülekedés is övezte a korlátozott számban készült lejátszók megvásárlását. Végül a készítőik ígéretet tettek egy újabb sorozat legyártására.

Balee és Csaba

HARDVER SIMOGATÓ

HUNGARIKUM: MICRO-POKEER

GYÁRTÓ: MICRO-STÚDIÓ
KIADÁS: 1987
EREDETI ÁR: 2000 FT

A Micro-POKEer egy Multiface-hez hasonló interfész, amivel meg lehet állítani az éppen futó programot, ki lehet menteni a teljes memóriát, örökélet-pókot lehet beírni. A Multiface-nél kevesebbet tud, de még így is nagyon hasznos kis eszköz. A Micro-Studio valamikor 1988-1989 környékén dobta piacra, ha jól emlékszem, én egy Karácsonyi vásáron (BME aula) találkoztam vele, de hirdették az SpV-ben is.

A ROM-ban található szöveg szerint a kódot a V&REW vagy W&REW software Hungary készítette. Nem hallottam még róluk és az interneten sem találtam utalást, az interfész menüjében csak a MICRO-STUDIO van megemlítve.

Funciók

- Programok fagyasztása (NMI gomb)
- SCREEN tartalom Load/Save (16384-23295; 6,75 KBájt)
- Teljes RAM tartalom Load/Save (16384-65535; teljes 48 KBájt)
- Turbo Load/Save (dupla sebességgel)
- Peek/Poke/ugrás memória címre
- Meleg reset
- Hideg reset

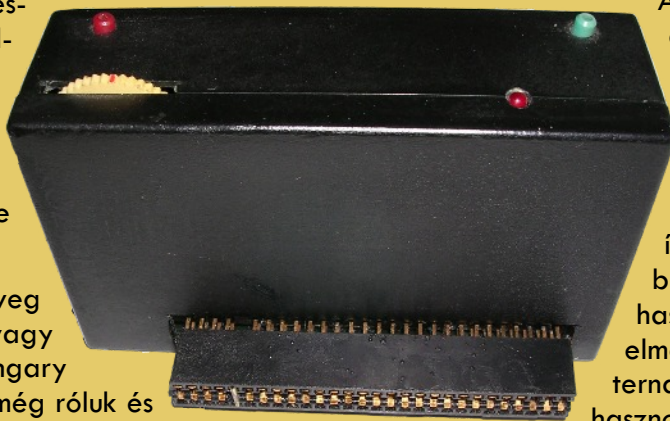
A hardver

A MICRO-POKEer nagyon egyszerű felépítésű. Található benne egy 8 KBájt méretű EPROM (M2764A), 2db 7432-es VAGY kapu és egy 7474-es D Flip-

Flop. RAM nincs benne, ellentétben pl. a Multiface-szel. Az én MICRO-POKEer verzióm NYÁK lemeze nagyon barkácszagú, valószínűleg kézzel rajzolták és házilag marták.

A ROM egy része az eredeti ZX Spectrum ROM-jából lett átemelve (pl. Load/Save rutinok, karakterkészlet). A 8K-s EPROM több mint fele üres, sőt, az EPROM akár 16K-sra is cserélhető, így rengeteg fejlesztési potenciál van benne.

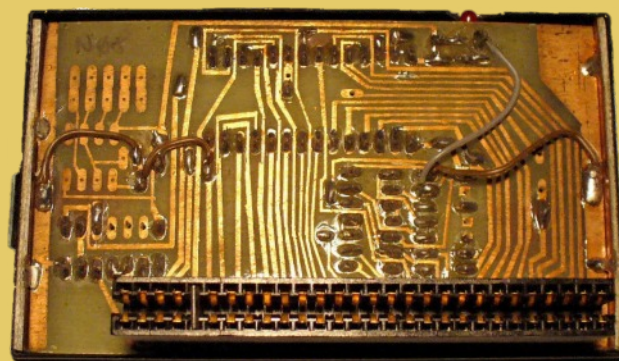
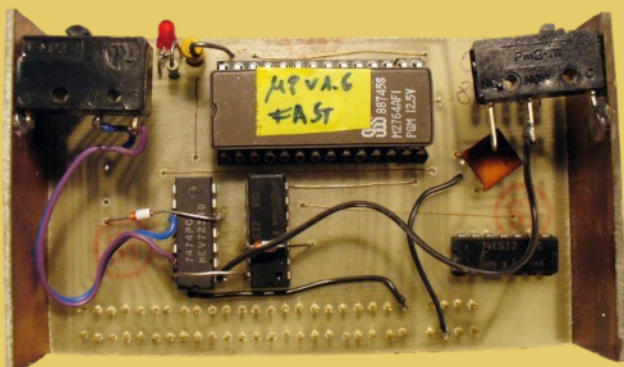
Az elektronika úgy van kialakítva, hogy a 7Fh (127d) portra írt zéróval lehet visszakapcsolni az eredeti ROM-ot, illetve visszatérni a megszakított programba. Érdekesség, ha 1-et írunk a 7Fh portra, akkor bekapcsol a kütyü. Ezt nem használja ki az eszköz (pl. az elmentett snapshot), de egy alternatív firmware-ben esetleg hasznos lehet.



Megszakítás

A futó program megszakítása a kék gomb megnyomásával történik, mikor az eszköz a /ROMCS jel segítségével a saját ROM-ját lapozza be, a vezérlés pedig az NMI címre

kerül (66h). Itt még nem történik kiírás a képernyőre, csak csendben várakozik a program a billentyűkre. Nincs saját RAM-ja az interfésznek, ezért bármilyen képernyőre írás „belerondít” a képbe. Ha visszakérül a vezérlés a megszakítás előtt futó programra és az nem frissíti a felülírt területet, akkor a MICRO-POKEer által kiírt szöveg végleg ottmarad. Ezért pl. a Micro-Monitor - amivel a memóriát lehet írni/olvasni - beviteli mezőjét bárhová ki lehet tenni (irányítás Q,A,O és P gombokkal).



Snapshot

A kimentett snapshot kap egy Basic loadert, ami egy gépi kódú betöltőt is tartalmaz. A Basic loader először betölt egy screent, ami a program megállításának pillanatában látható képernyőt tartalmazza, majd a loadert a Basic területről a képernyőmemória elejére másolja és ezzel betölti a teljes elmentett memóriát, ezután a vezérlést átadja a programnak. Van egy csel itt: az elmentett screen 2 bájtal hosszabb, mint a teljes képernyőmemória (6914 bájt). Ez a plusz két bájt tartalmazza a program megállításkor elmentett SP (stack pointer) értékét. Így ezt a betöltés után visszaállítja és az elmentett állapot a megállítás pillanatától folytatható (a stack-ben van az NMI feldolgozásakor elmentett PC - program counter). Ennek a megoldásnak van egy szépséghibája: A képernyő első óó bájtját felülírja, ami több mint két pixel-sor. Itt is igaz a fenti állítás: ha nem frissíti ezt a részt a program, akkor az végig ott rontja a képet.

Turbo

Az interfész turbo save üzemmódot is tud, ami pontosan dupla sebességet jelent. Sajnos nagyon sok magnóval ez visszatölthetetlen, ráadásul, mivel nem ment turbo loadert a Basic betöltőbe, a MICRO-POKEer nélkül vissza sem lehet tölteni a turbós mentést. (Az interfésznek van turbo load opciója.)

A Turbo Loader bizonytalansága miatt, annak idején felajánlottam a Micro-Studio-nak, hogy átírom megbízhatóbbra a betöltőt. Tetszett az ötletem, de mégsem valósult meg, már nem emlékszem miért. Nem sokkal később Amigára váltottam, így részemről feledésbe merült a téma. A turbo load/save az eredeti ROM rutin másolata, néhány konstans időzítés átírásával, így a javítása sem több pár percnél.

Az én MICRO-POKEer verziómon nincs továbbmenő élcsatlakozó, így pl. a Kempston joystick interfészemet sem tudtam használni (mivel azon sincs), emiatt egy kicsit macerás a használata.

Mivel tud többet a Multiface?

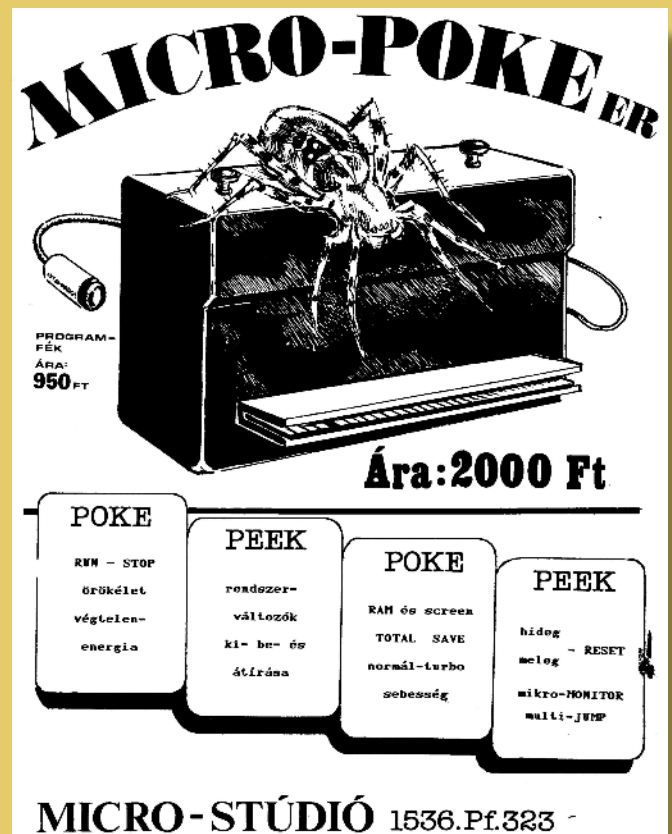
A Multiface a saját 8K-s RAM-ba el tudja menteni a teljes képernyőmemóriát, így bármilyen szöveget/menüt ki tud tenni a képre, a végén vissza tudja állítani teljesen a program képernyőjét. Ezt a 8K RAM-ot sok másra is lehet használni, pl. ún. toolkit-et lehet feltölteni és futtatni benne. Ilyen volt a Genie (disassembler) vagy a Lifeguard (örökélet kereső). Van rajta Kempston joystick port is, így kapásból meg lehet spórolni egy plusz interfészt.

Értékelés

Annak ellenére, hogy a MICRO-POKEer-nek vannak hiányosságai, nagyon jól hekkkelhető kütyü.

Bicsák Lajos (Asimo)

Korabeli hirdetés



MICRO-POKEER

PROGRAM-FÉK
ÁRA: 950 Ft

Ára: 2000 Ft

POKE RUN - STOP örökélet végtelen- energia	PEEK rendszer- változók ki- be- és átírása	POKE RAM és screen TOTAL SAVE normál-turbo sebesség	PEEK hídok - RESET melog mikro-MONITOR MULTI-JUMP
---	---	--	--

MICRO-STÚDIÓ 1536.Pf.323

WANTED

Az alábbi magyar fejlesztésű ZX Spectrum programokat ke-ressük z80/tap/tzx formátumban vagy akár kazettán, természetesen keressük a hozzávaló kazettaborítót, leírást magyar vagy akár angol nyelven is.

- 3D Sakk (Novotrade)
- Betűpóker (Novotrade)
- Betűrómi (Novotrade)
- Erdélyi fejedelmek, Habsburg uralkodók (Sági György, 1986, KLTE OK.)
- SZJA'88
- Mikroszervíz programcsomag
- DEMAK eprom égető programját
- DEMAK ROM extender eredeti ROM-ját
- Evolutaur c. játékot (Bíró Ádám & Zoltán)

A specyalista@sinclair.hu email címen értesíthsz bennünket, ha neked megvan valamilyik. Természetesen, ha olyan magyar fejlesztésű program van a birtokodban, ami ebben a listában nem található, akkor se késlekedj:

<http://sinclair.hu/index.html?szoftver/magyar/tartalom.php>

HARDVER ÖTLETEK

FPGA SPECTRUM - 1. RÉSZ

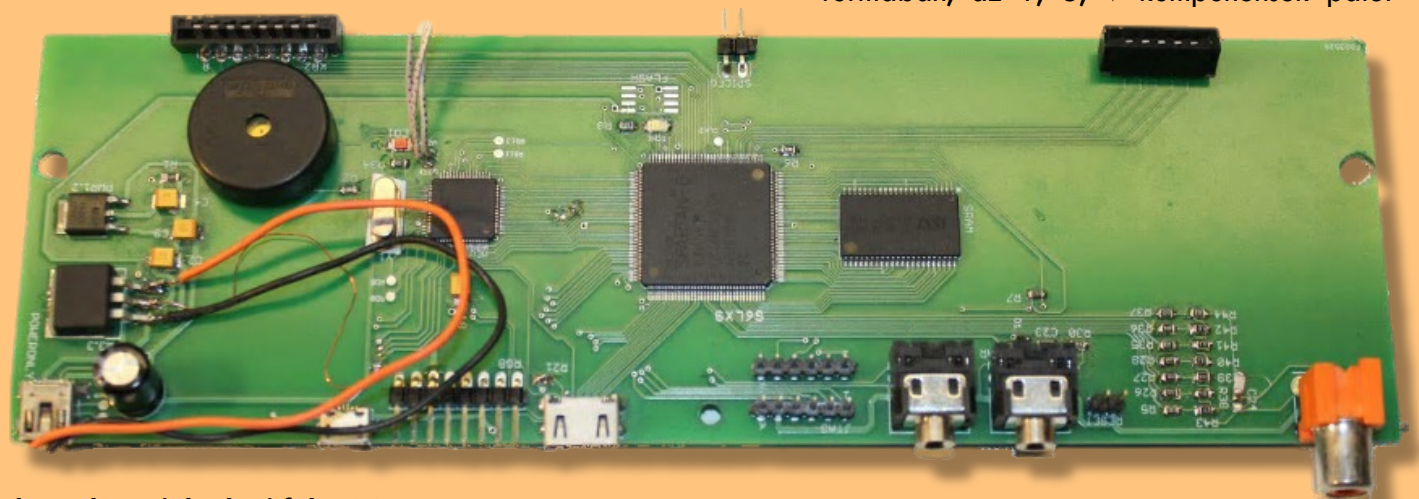
A Sinclair.hu eddig is támogatott minden Spectrumhoz vagy egyéb Sinclair géphez kötődő hardver fejlesztést, hisz életben tarthatják kedvenc masinánkat.

Ugyanakkor sajnálatos tény, hogy az öregedő gépek egyre több fejtörést okoznak, épp ezért tartjuk ezt a projektet kiemelt jelentőségűnek, melyet Jocó tulajdonképp a legnagyobb titokban dédelget már jó ideje, de nemrégiben betekintést engedett számunkra is az egyik klubnapon, melyről egyébként olvashattatok az előző számunkban. Most egy kicsit bepillantottunk a részletekbe is egy cikksorozaton keresztül.

Nemcsak önmagában érdekes ez a fejlesztés, hanem egyre több hajtás ereszt, amik szintén izgalmas témával kecsegtetnek. A teljesség igénye nélkül megemlíthető az ULA kiváltás, az USB billentyűzet illesztés vagy a ZX Spectrum USB billentyűzet készítés.

Ez a projekt már egészen előrehaladott állapotban leledzik, de azért kezdjük az elejéről a bemutatását.

Balee



A rendszer jelenlegi felépítése:

- Xilinx Sartan-6 LX9 FPGA (ebben foglal helyet a Z80 és az ULA, valamint minden egyéb spektrummal kapcsolatos hardver elem FPGA megvalósítása).
- ISSI 61lw25616 256Kx16 SRAM memória, a panel kialakítása miatt ez 512Kx8 bites szervezést jelent az FPGA számára.
- Microchip PIC32MX695 mikrokontroller, melynek feladata (lesz) a bootolás vezérlése, majd a USB eszközök és a microSD kártya kezelése. A mikrokontroller USB portját egy mini B csatlakozón vezettem ki (OTG), mert a ház erre ad lehetőséget.

A panel az eredeti Spectrum alaplap méretének nagyjából a fele, és így is szinte üres, de a hátsó

kivezetések és a billentyűzet csatlakozói miatt ez a méret mindenképpen szükséges, hisz a panel az eredeti házba szerelhető.



A ZX Spectrum 48 FPGA implementációja az FPGA-ban lényegében készen van. A megvalósított gép 99.9%-ban kompatibilis az eredeti Spectrummal, ez a speciális effektek is lefedi, border és attributum varázslásokat.

Video:

Az FPGA mag három féle videó kimenetet valósít meg:

- PAL Composite videó jelet ad 5Y6U6V formában, az Y, U, V komponensek palet-

tában állíthatók. A kép minősége a composite jel tipikus jellemzőit mutatja (pl. color carrier interferencia).

- 3x3x3 bites analóg RGB kimenet hagyományos RGB monitorok és Euro-Scart bemenettel rendelkező TV készülékek számára. Ezen a kimeneten 512 szín megjelenítése lehetséges (egyszerre csak 16 persze), szintén palettában állítható. Az RGB csatlakozót egy egyszerű tükkesoron vezettem ki, mert ez fért el.
- HDMI (DVI) kimenet, 720x576-os progressive digitális kimenet. A színek 8 bitesek, tehát 16 millió színből lehet kiválasztani a Spectrum színeit. (A HDMI kimeneten egyelőre nincs hang). A fizikai csatlakozó a kis méretű, HDMI-C változat.

Audio:

Sztereo analóg kimenet a MIC Jack csatlakozón keresztül. A MIC csatlakozóra az eredeti MIC bit és a SPEAKER bit hangja keverhető, hangerejük állítható.

A SPEAKER belső hangszóróval is működik. Az EAR bemenet a programok hagyományos betöltéséhez használható.



(Természetesen AY chip is megvalósítható az FPGA-ban, de ez majd a 128-as modellnél lesz érdekes.)

Memória:

512Kx8 bit RAM, egyelőre a 48-as Spectrum esetében csak 64K van felhasználva.

Háttértár:

Micro SD kártya vagy USB pendrive. A mikrokontroller firmware-e képes lesz ezeket kezelni, a Spectrum BASIC pedig kibővül az általános fájlkezelő parancsokkal, valamint a legegyszerűbb .SNA snapshot fájlok betöltésének lehetőségével.

Billentyűzet:

Az eredeti Spectrum fóliabillentyűzet használható, illetve USB-n külső USB-s keyboard is működni fog.

Egér, gamepad:

A mikrokontroller firmware-e képes lesz ezeket kezelni, az egyszerű, olcsó digitális game controllereket a Spectrum joystick-ként látja majd.

Megjegyzés: Több USB eszközt külső USB HUB segítségével lehet majd használni, a firmware kezelni fog egeret, billentyűt, két gamepad-et és usb pendrive-ot. (Ez a terv, remélem, hogy megvalósítani is sikerül majd.)

Tápfeszültség:

DC 5V, szabályzott.

Áramfelvétel:

A csatlakoztatott USB eszközök függvénye, de alapesetben 100-300mA.

Megoldandó szoftveres feladatok:

- A mikrokontroller firmware programot össze kell rakni. Részekben már megvan (pl. SD kártya írás, olvasás, USB eszközök kezelése, kommunikáció az FPGA-val), de ezeket össze kell fűzni egyetlen egységes, stabilan működő alkalmazásba.
- A spectrum ROM módosítása. Ennek is vannak elközményei, de jelenlegi hardverre kell szabni, hogy a háttértárat biztosan kezelje.
- HDMI audio implementálása.
- Spectrum 128 FPGA mag implementálása. A meglévő 48-as mintájára ez már nem lesz nehéz feladat, itt is inkább a ROM átírása a nehézkes. A ROM-ot itt is alkalmassá kell tenni arra, hogy a mikrokontrollert, mint háttértárat "lássa", és kezelje.

Hardveres hibák, megoldandó problémák:

- A micro HDMI és a USB OTG csatlakozókat át kell helyezni a panel forrasztási oldalára, hogy a csatlakozókat be lehessen dugni, felül ugyanis nincs elég hely a Spectrum dobozán a nyílás ehhez kicsi. (Az ideális a normális HDMI és USB-A host csatlakozó lenne, de ezekhez már faragni kellene a házból.)
- A tápfeszültség csatlakozót cserélni kell hagyományos JACK-re a mostani USB helyett, az USB túl alacsonyan van, a ház miatt nem lehet rendesen bedugni.
- A panelen pontosítani kell a furatok és a csatlakozók helyét, pár milliméterrel még eltér az eredetitől.
- Kivezethető RESET gombot kell csinálni.
- A LED-ek kivezetését is meg kell oldalni.

László József (FPGA Jock)

Folytatjuk...



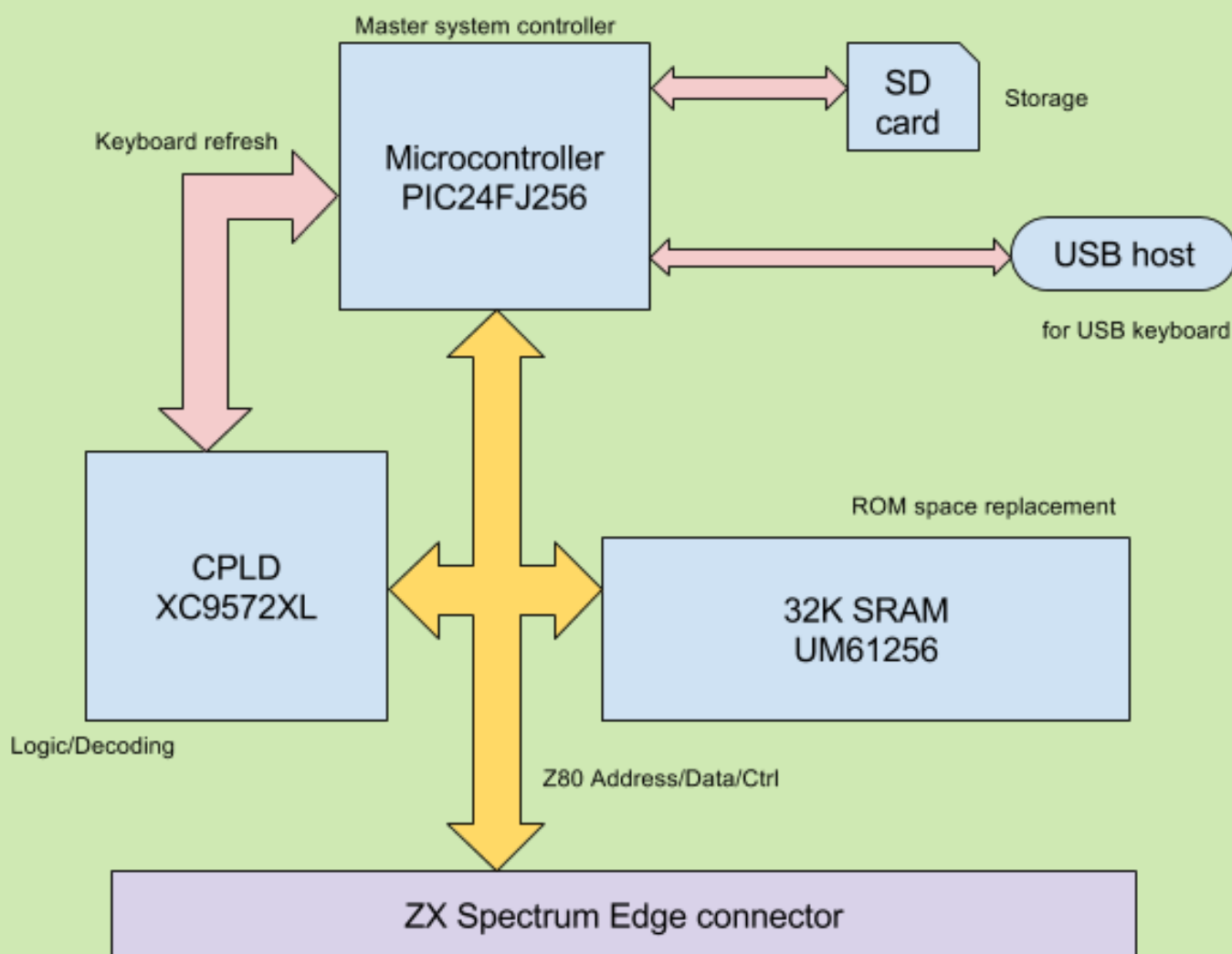
HARDVER ÖTLETEK - ZX SPECTRUM

USB BILLENTYŰZET ILLESZTÉSE EXTRÁKKAL - 1. RÉSZ

Úgy gondoltuk hasznos lenne egy ilyen PC USB billentyűzet illesztő Spectrumhoz, mert ezidáig csak PS/2-es verziók készültek. Így kerülhetett sor arra, hogy megtervezsem Balee nyomására ezt az illesztőt, ám rögvest tovább is gondolta és még egy SD kártya illesztéssel is megfejelte ezt, gyakorlatilag ugyanazzal a két chip-el (de nagyobb lábszámmal), plusz egy ROM-ot helyettesítő RAM chip-el. Ezt az illesztőt mutatjuk be nektek ebben a cikk sorozatban a végső megvalósításig.

tudja írni, ezáltal a Spectrum ROM-ja tetszőlegesen lecserélhető.

PIC mikrokontroller: a rendszer vezérlését teljes egészében ellátó vezérlőegység. A PIC megvalósítja az SD kártya olvasását és az USB port kezelését is. Az USB host port első sorban a külső billentyűzet illesztését látja el, de megfelelő firmware-el lehetséges lesz más USB eszközök illesztése is (egér, game controller, pendrive)



Tekintsük át az egyes komponenseket:

CPLD: megvalósítja a külső billentyűzet illesztését, valamint tartalmazza azt egyéb kapuzásokat és címdekódolást a kártyán lévő RAM és a I/O portok számára.

SRAM: az eredeti spectrum ROM területére lapozható RAM, 2x16K szervezésben. A SRAM tartalmát a PIC

A PIC teljes egészében képes átvenni a rendszer vezérlését a Z80-tól, mert a busz minden ehhez szükséges jelét eléri és vezérelni tudja. Induláskor a PIC azonnal leállítja a Z80-at a BUSREQ jel segítségével, majd átvéve a buszt feltölti a SRAM tartalmát az SD kártyáról (tehát betölti az eredeti, vagy egy módosított spectrum ROM-ot a SRAM-ba). Erre a lépésre abban az esetben mindenképp szükség van, ha az eszközt, mint háttértárolót akarjuk használni, hiszen az

eredeti ROM nem tud ilyen eszközt kezelni. Amikor a SRAM a megfelelő tartalmat megkapta, a PIC elengedi a buszt, letiltja a belső ROM-ot a ROMCS jel segítségével, és RESET-el elindítja a Z80-at, ami ekkor már a SRAM-ból fog futni. Ha tehát a ROM területen található szabad területre saját vezérlőprogramot írunk, akkor ez képes lesz a megfelelő I/O portokon keresztül elérni a PIC-et. Ekkor a PIC átkapcsol storage vezérlő módba, azaz úgy fog működni, hogy egy minimális parancskészlettel az SD kártyán található FAT fájlrendszert elérhetővé teszi a Z80 számára. Így a ROM területre írt módosított Spectrum rom új és felülírt parancsokkal képes lesz az SD kártya tartalmát olvasni, írni.

A PIC, mint általános I/O port elérhető lesz a Z80 számára, ezen keresztül valósítható meg a Spectrum Basic-ből is elérhető SD kártya kezelés (fájl írás és olvasás, load, save, snapshot load stb.). Ezeket az I/O portokat a CPLD kapuzza ki, ebben a konstrukcióban az A3 és A4 biteket használtam fel a dekódolásra, amelyeket eredetileg a ZX Interface 1 használt.

A külső billentyűzet illesztését szintén a CPLD oldja meg, a Z80 szempontjából nincs különbség a belső és a külső billentyűzet elérésében, ugyanúgy az A0/IORQ/RD (ULA read) és a címbusz felső 8 bitje válaszolja ki a megfelelő billentyűt. A két billentyűzet párhuzamosan is tud működni. A CPLD-ben lévő 5x8 bites tároló egységet a PIC az USB billentyűzet állapotától függően állítja, a CPLD pedig a belső logikája szerint ezt rákapuzza a az adatbuszra, amikor a Z80 ULA read műveletet hajt végre.

A rendszer inicializálása után a PIC főprogramja két feladatot lát el: a fő ciklusban kapja meg az USB billentyűzet esetleges változásait, ami alapján a CPLD-ben található 5x8 bites tárolóterületet frissíti. Amennyiben az I/O portokon háttértár vezérlő parancs érkezik, akkor a PIC azt kiszolgálja. E két feladatot egymás után sorban hajtja végre a PIC firmware, de ez nem okoz gondot, mert az USB billentyűzet kezelése megszakítás alatt törlődik, legfeljebb extrém versenyhelyzetben a billentyűzetmátrix frissítése kicsit késik (amíg a PIC az SD kártyát írja, olvassa), de nem marad el.

László József (FPGA Jócó)

Folytatjuk...

IDÉZET

Mel Croucher interjúban olvastuk:

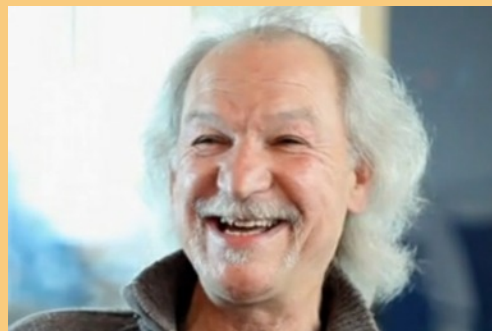
Forrás: zxgoldenyears.net

“How does the industry compare with the way it was in the early Eighties? Has the sense of adventure returned along with improved technology or is it still first and foremost a business?”

Azaz “Milyen mértékben változott meg az iparág a nyolcvanas évek elejéhez képest? A fejlettebb technológia visszahozta a kalandvagyat a szakmába, vagy ez továbbra is inkább a legelső sorban üzlet?”

“A Don't forget there were only a few thousand computer games players out there when we started, and they were all enthusiasts. There were a couple of computer magazines printed on toilet paper, most people in the industry were willing to help one another, and there was a feeling we were all on the same side. Producers and public alike. It was and always will be a business of course. What has changed is the scale of finances. It's now very big business, and a business that is based on promoting violence and greed. Improved technology has merely allowed the suits to push higher-definition violence and greed to the hard-of-thinking.”

“Ne felejtsük el, amikor elkezdtük ezt az egészet, pusztán csak néhány ezer ember volt, akik számítógépen játszottak - de ők megszállottak voltak! Néhány számítógépes témájú újság volt csupán kapható, az is WC papírra nyomtatva, és a legtöbben az iparban még segítőkészen álltak egymáshoz, és az volt az érzésünk, hogy mind ugyanazon az oldalon állunk - mind készítő, mind a fogyasztók. Persze ez mindig is egy üzletág volt és az is marad. Ami megváltozott, az a szakmában forgó tőke mértéke. Ez most már hatalmas üzlet! Egy olyan üzlet, ami abból tartja fent magát, hogy az erőszakot és a mohóságot helyezi előtérbe. A fejlettebb technológia pusztán azt tette lehetővé, hogy az öltönyösök odafentről HD-ban/nagy felbontásban tolják a brutalitást és nagyravágyást a gondolkodni nem akarók arcába.”



FEKETE DOBOZ

AVAGY A KÖKORSZAKIK ESETEI

Olvasóink által beküldött korabeli kedves történetek a kezdeti lépésekről.

Arra gondoltunk, hogy beküldhetnétek milyen butaságokat csináltatok a kis gumibillentővel az aranykorban, amikor még mi voltunk a feltörekvő ifjúság. Ennyi idő után talán már nincs mit szégyenkezni a dolgon, egy kis önirónia senkinek sem árthat meg.

Kalandok Spectrum programokkal

Számomra minden program külön kihívást jelent. Gyerekként valami különös okból kifolyólag azt hittem, hogy minden program végigjátszható és valami fantasztikusan hatalmas, csodálatos, lenyűgöző dolog történik a végén. Ennek okára nehéz rájönni, bár szerencsére a fantáziám meglehetősen élénk volt. A cél eléréséért képes voltam hosszú órákat játszani, gyakorolni. sőt sokszor még idősebb játékos ismerősömet is türelmesen nézni, hátha történik valami.

Erre remek példa a Deathchase program, amelyben napokig nem tudtuk, hogy löni is lehet, így aztán az órákig tartó üldözés végén sem történt semmi.

Mókásnak tűnhet, hogy vajon miért is nem jöttünk rá a feladatra. Nos, a magyarázat rendkívül egyszerű. A másolt programokhoz semmilyen leírás nem kapcsolódott, így mindig próbálkozással kellett kitalálni, melyik gomb mit is csinál. Ebben a programban pedig a lövés csak bizonyos esetben működött...

Aztán a sort a River Rescue folytathatja, ahol remekül lehetett kerülni a krokodilokat és fatörzseket, majd kitaró szemgolyózás után még néhanapján repülök is megjelentek. És ez ment órákon át. Aztán egy véletlen mozdulat hozzácsapta a hajót a parti kikötőhöz és az ámulat percekre mozdulatlaná tett. Hihetetlen élmény volt és talán az ilyen "kalandok" váltották ki a folytonos csodavárást. A Galaxians-ról már ne is beszéljünk. Érettebb fejjel egy egyszerű lövöldözős játékról van szó, de gyerekként minden egyes küldetés után a "Már csak egyszer kell nyerni!" felkiáltás hangzott tőlem.

Nagybátyám profi játékos volt a Horace & the Spiders játékban, amelyben ketten fergeteges izgalmak közepette 10 pókig jutottunk el, ahol minden pillanatban felugráltam, annyira a játék hatása alá kerültem. Persze ott se értünk el gratulációt, de hitem töretlen volt. A játékelmények nagy része nagyszüleimnél eltöltött hosszú órákhoz kapcsolódnak.

Nagypapám műszerész volt, kezdetben kölcsön, később saját tervezésű és építésű számítógépével játszottam. Akármilyen hihetetlennek is tűnik, a Manic

Miner első 7 pályájának ugráskombinációit tőle tanultam, sőt a mai napig is az ő útmutatása alapján csinálom. Ő is éjszakába nyúlva játszott, nagyon szerette a Spectrumot.

Amikor végre saját Spectrumom lett, új korszak kezdődött a játékok birodalmában. Végre nem kellett sietni, volt idő alaposan kiismerni egy-egy programot. Legjobb barátommal akkoriban két olyan játék volt, amit mindketten szerettünk. A Kokotoni hallatlan izgalmakat adott, mert ketten játszottuk. Egy csillag összegyűjtése neki, a másik nekem jutott. Napokat, heteket gyakoroltunk és tudtuk, valami nagy dolog lehet a program végén. Sajnos azonban soha nem sikerült végigjátszani, életeink mindig elfogytak, pedig tényleg volt igazi vége.

Az egyik leghosszabb játékelményem is barátommal történt. Egyik évben elhatároztuk, hogy szilveszterkor a Match Day fociban megnyerjük a bajnoki kupát. De nem akárhogy! Nem öt perces, vagy tízperces félidőssel, igazi, kétszer 45 perces meccsekkel! A bajnoki címhez 3 győzelem kellett (ha jól emlékszem) és az este 10-kor kezdődött "bajnoki évet" reggel hajnali 6-kor fejeztük be (a mérkőzés közben azért kellett enni és a szülőkkal koccintani) és sikerült teljesítenünk célunkat. A maga nemében ilyen szuper szilveszterem soha nem volt.

Ahogy nőtem fel, már kezdtek letisztulni a játékprogramok. Voltak, amelyeket könnyedén sikerült végigjátszani, számos program hosszú küzdelem, térképkészítés, örökélet keresés után adta csak meg magát. Nem tudom, átélte-e már valaki, de a Silent Service tengeralattjáró szimulátor alatt halotti csönd honolt szobámban, még járni is csak lábujjhegyen szabadott. Mosolyra fakasztó visszaemlékezni. De hátra van még a feketeleves!

Akkoriban a fülembé jutott, hogy egy klub működik valahol messze, a Mexikói úttól kifelé a városból, amit a Csokonai Művelődési házban tartanak. No, több se kellett, a következő szombatn már úton is voltam. Belépve a terembe rengeteg embert láttam, spectrumosokat, commodorosokat. Nagy nyüzsgés volt, alig tudtam felfogni, mi is történik. Azért azt sikerült megérteni, hogy itt programot is lehet cserélni! Ehhez viszont nem árt számítógépet is hozni. A következő hétvégén egy táskával felpakolva kedvenc Spectrumommal, magnóval és JUNOSZTY tévémmel megérkeztem és 50 Ft-ért béreltem egy helyet magamnak. Az utazás borzalmas volt, mivel több, mint 2 és fél órát tartott oda, ráadásul a tv se volt egy könnyű szerkezet. Bérelni lehetett, de az plusz forintokba került volna, zsebpénzemből pedig nem tellett rá. No

tehát sikerült helyet foglalni és vártam... vártam... vártam. Mint említettem, szégyenlős ifjunc voltam, még programot kérni se mertem. Aztán egy jószívű szomszéd megszánt (látva tanácsstalanságomat) és odaadott egy programlistát, amiről választhattam. Szerencsére én is tudtam adni neki programot, így létrejött a nagy üzlet, cseréltünk. Hát, ami azt illeti, ez nem jött be igazán. Megint többet vártam, mint kaptam, mert a programok zöme az angol név alapján szenzációs dolgot rejtett, de betöltés után csalódás, csalódás hátán. Így többet el se mentem, de igazat megvallva nem is nagyon akartam.

Sajnos, mint minden csoda, ez a korszak is véget ért, így aztán a Spectrum nyugállományba ment. Telt, múlt az idő...

Aztán egy nap érdekes programra bukkantam. ZX Spectrum emulátor. Szinte napokig mást sem tudtam csinálni, mint játszani, próbálgatni. Kedvesem beavattott az internet világába. Első alkalommal, naivan csücsülve kérdeztem tőle, hogy mire is jó ez az internet vagy mi. Azt mondta, lehet keresgélni érdekes dolgokat. Egy hirtelen ötlettől vezérelve rákerestem a Spectrum szóra, bár ez nem volt túl szerencsés, mivel rengeteg, nem ZX

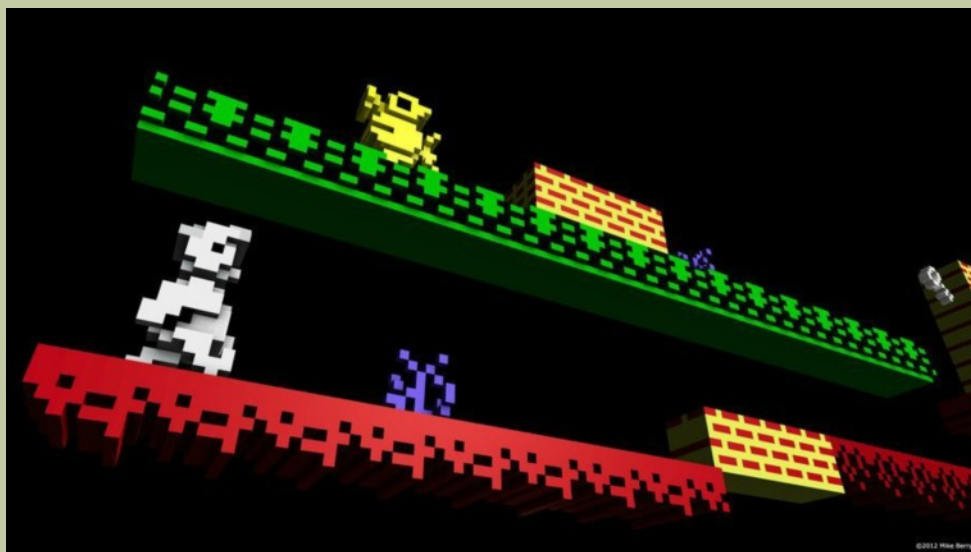
dolog is előkerült. De a lényeg is ott volt! Klikk és egy hét múlva saját internet előfizetésünk volt, aztán indult az élet új csapáson. Éjjelente, a kedvezményes időszak miatt, állandóan programokat töltöttem le és mindent, ami csak kapcsolatban volt a Spectrummal. Másfél hónap után kifogytak az archívumok és végre kezdettem feldolgozni az anyagokat. Közben azért arra is jutott idő, hogy az emulátort megvásároljam. Mondhatni bátor vállalkozás volt borítékban pénzt küldeni, de szerencsémre jó egy hónap után megérkezett a jogtisztá, teljes verziójú Z80 emulátor floppy lemezen. A fontos az volt, hogy magnóról is lehessen betölteni programokat, hiszen a teljes kazettagyűjteményem megvolt és ez csak ezzel lehetett!

El lehet képzelni, amikor az első betöltési hangok megjelentek a hangszórókon... Szinte egy pillanat alatt minden régi emlék beugrott. A kazettán keresés, a betöltés várakozása újra a régi időket juttatta eszembe.

Mivel az ember igazán soha nem nő fel, a kalandok sem maradtak abba. Újra előkerültek a régi játékok, eleinte az ismert régiek, majd főként olyanok, amelyeket gyerekkoromban nem tudtam végigjátszani. Első, legnagyobb sikerem az Atic Atac-hoz fűződik. Sikerült örökélet és bármilyen csalás, térkép nélkül végigjátszani, ráadásul nem is egyszer. Hatalmas élmény volt.

Aztán voltak meglepő pillanatok is. A Simcity-vel PC-n találkoztam először, aztán megdöbbenve találtam meg, hogy Spectrumon is létezett. Hogy mik vannak?! Néhány program akkora meglepetésekkel szolgált, hogy nem is gondoltam volna. A ManicMiner 2-ről mindig is azt hittem, hogy sérült verziójú, mert a 6. pályán egy olyan kulcs volt, amit nem lehetett összerakni. Ezt egy fórumon meg is említettem, ahol meglepetésemre azt a választ kaptam, hogy meg lehet csinálni. Kipróbáltam és tényleg sikerült. Még ilyet! Hány, de hány éven át próbálkoztam hiába és tessék, csak nem sikerült rájönni a megoldásra. Ez az eset után kedvet kaptam további kérdezősködéskedésre is. A Technician Ted szintén hibásnak tűnt, de szintén kiderült, hogy csak én nem ismertem trükkjét, bár trükknek nem nevezhető. Többekkel megegyezve ez a mai napig

ismert legnehezebben teljesíthető, értelmetlenül elbonyolított játék. A cél a Jet-Set Willy-hez hasonló, de nem elég, hogy milliméterre pontos ugrásokkal kell haladni, a feladatokat megadott sorrendben, végtelenül



szoros időre kell teljesíteni. El kell árulnom, hogy többszöri próbálkozás után még az első is lehetetlen feladat volt számomra. Pedig gyerekkoromban mennyi időt elpazaroltam erre a szépen kivitelezett játékra szinte fájj. Programkészítőikben ekkorát még soha nem csalódtam. A Glider Rider-t pedig már meg sem említem.

De, hogy ne ilyen szomorú véget érjen írásom, meg kell említenem azt a kis programot, amellyel minden pillanatban képes vagyok játszogatni, pedig szokatlan a billentyű kiosztása (Q, W, P, L), nem lehet löni benne, nincs hihetetlen szépen kidolgozott grafikája, a neve: Snake-Pit... :-)

Pgyuri (2002)

OLVASÓI LEVELEK

Előző számunk megjelenése óta a szerkesztőségünk-be érkezett olvasói kritikák...

Nos, Bali - és mindenki aki részt vett a szerkesztésben!

Jó ideje leszoktam az olvasásról, pedig valaha faltam a könyveket. Gyorsolvasó versenyt is nyertem, volt, hogy a munkám is arra kötelezett, hogy 500 oldalas anyagokból készüljek fel 1 nap alatt. De elveszítettem az olvasás élményét. Csak a száraz adatok, célirányos információk felé fordultam.

Most - hála nektek és a kiadványnak - azon kaptam magam, hogy élvezem újra az olvasást, újra átérzem az olvasás örömét.

Nem tudtam letenni a Speccyalista világ 1. számát.

Annyira jól sikerült, hogy arra nem is tudok szavakat találni.

Talán csak egyet, amiben minden benne van, amit éreztem közben: Köszönöm nektek!

Nos, ide 100 smiley kíváncozna, de talán anélkül is értitek a lényegét.

Szép volt fiúk!

buddhasoft

Szuper!!!

Jól van szerkesztve, jó tartalom, szép kivitel!

KerLaci

Tisztelt Szerkesztők!

Örömmel vehettem kezembe a Speccyalista Világ 1. számát, amely idén már a második, ZX Spectrummal foglalkozó kiadvány!

Mivel vidéken élek, így nem tudtam a Speccyalista Baráti Körről, de megköszönöm, ha megírják, hogy hol vannak rendszeres klubnapjaik, mert szívesen csatlakoznék időnként.

A lapjukhoz szeretnék gratulálni, mert nagy munka lehetett megírni, de egyben megfogalmaznék egy kisebb kritikai észrevételeket. Külső megjelenése, arculata a lapnak igencsak modern, ami számomra a ZX Spectrumhoz túlzás. Nagyon szeretem ezt a kis "bohó" számítógépet és talán kicsit hozzá jobban illő, egyszerűbb megjelenés lenne célszerűbb a későbbi folytatásoknál. Koromból adódóan a betűméretet is kicsit nagyobbra vehetnék, hogy jobban olvasható legyen. Esetleg nem gondolkoznak nyomtatható formátum előállításán, ami már a háttér lapszín eltüntetésével is sokat segítené? Persze fénymásolni e-szembe sincs, de papíron sokkal jobban lehet olvasni és ha sok lapszámot megélnék, akkor a polcomon is szépen mutathatna bekötte.

Most pedig a dicséreteteket következzenek!

A fő oldal kézzel rajzolt grafikája csodálatos! Nem tudom, ki alkotta, de ügyes volt és nagyon eltalálta (bár a horizonton fel- vagy eltűnő nap felette nem odaillő), elsőre megragadta figyelmem! Nagyon szeretnék még ilyeneket látni a lapban.

A cikkek jól összefoglaltak, bár egy részüket mintha már olvastam volna valahol... ettől függetlenül így egyben mindenképpen élvezetesek és hasznosak. A játékaajánlók tetszettek, lényegre törőek, a WoW írás igen szépre sikerült. Van szerencsém ismerni a játékprogram készítőit és fontosnak érzem tisztázni, hogy NJózi sokkal nagyobb részben vett részt a fejlesztésben, mint amennyit a cikk említ. Ha jól tudom, nem csak motivált, hanem az összes grafikát ő készítette, a pályákkal kapcsolatban is rengeteget dolgozott és egyéb feladatai is voltak, de ezt a WSS-től kellene megkérdezni.

A Gelka összefoglalás pedig egy igazán remek oldal!

A rejtvény megoldását beküldtem, majd telefonon érdeklődtem és közölték, hogy nyertem, de mind a mai napig nem érkezett meg a díjam. Utána tudnának nézni, hogy mi lett vele? Ha lehet, személyesen venném át a szerkesztőségben.

Köszönettel egy új olvasójuk

Pgy Uri

A SZERKESZTŐSÉG VÁLASZOL

Először is köszönjük a méltató szavakat, nagy örömeinkre szolgál, hogy kezdeti próbálkozásunk nem volt hiába, sikerült olvasóinknak néhány oldalban megidéznünk gyerekkorunk Spectrum Világának szellemét.

A második számhoz vezető útunk talán még rögzösebbnek tűnt, hogy sikerül-e kitörni a méltán megbecsült Spectrum Világunk árnyékából és az Ő nyomdokaikon haladva létrehozni egy olyan kiadványt napjainkban, mely érdeklődésre tarthat számot, mind megjelenésében, mind tartalmában.

Éppen emiatt fontos számunkra minden olvasói visszajelzés, hogy azokat is beépíthessük a következő számainkba, ahogy ezt már most is megpróbáltuk.

Balee

REJTVÉNY

Beküldendők a függőleges 1, 12 és a vízszintes 1, 4, 9, 12 meghatározások 2016. február 29-ig az spv.rejtvény@sinclair.hu email címre. A helyes beküldők között ZX Spectrum If2 Multiromot, eredeti csomagolású microdrive kazettát valamint játékkazettákat sorsolunk ki. A Nyerteseket a következő számban tesszük közzé és email-ben értesítjük Öket.

	1	2	3	4	5	6	7	8	9	10	11	12
1				T					S			
2				■					■			
3			■				■					
4												
5					■					■		
6				■			■				■	F
7	F	■				■			■			
8			■					■				
9												
10						■				■		
11				■					■			
12				I					S			

Vízszintes

- Lásd a képet a következő oldalon
- Tüzet szüntet - Szibériai folyó, az Irtis mellékfolyója - nigériai nép
- Kaliforniai város, röv. - Épület - Oxigén-felhasználó baktériumok csoportja
- Lásd a képet
- Szorító (bokszy) - Tehát, latinul - Részvénytársaság, röv.
- A Volga egyik mellékfolyója - Víz tömeg - Az arc része
- Lehel névváltozata - Spanyol nagycsapat nevének rövidítése - ... Thurman (színész)
- Ezen a napon - Gyermekgondozó-nő - Olajexportáló országok szervezete, röv.
- Lásd a képet
- Elmore Leonard krimije - A Szovjetunió gazdaságpolitikája az 1920-as években - A sors fele!
- Svéd számítésvállalat (IAR) - ... Sharif - Főtiszt, röv.
- Híres Spectrum-programok szerzője, lásd a képet

Függőleges

- Orosz TR-DOS program, lásd a képet
- Két nyugati gót király neve volt - Mindenható a mohamedánoknál
- ... Gallen - Menekül - Kortárs filmrendező (Béla)
- Sebhely - Paradicsom
- Kúszónövény - Arcát maszk mögé rejtő
- Május 24-én ünnepli névnapját - Ám - Megállóhely, röv.
- Magunk - Argon vegyjele - Erdélyi havasok
- Higanyötvözet, régen fogtömésre használták - Svájc fővárosa
- Ezen a helyen, népiesen - Országos Egészségbiztosítási Pénztár, röv.
- Gumi az amerikai angolban - Sziget a szentendrei Duna-ágban - Tova
- Kutyánózi - Távol
- Lásd a képet

(Képek a következő oldalon találhatóak...)

REJTVÉNY



Vízszintes 1



Vízszintes 4



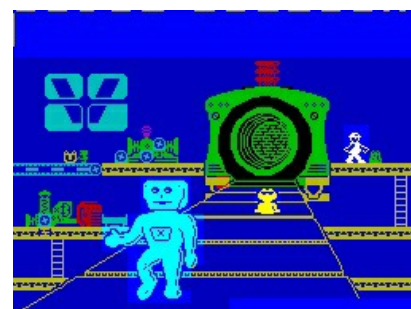
Vízszintes 9



Vízszintes 12



Függőleges 1



Függőleges 12

ELŐZŐ SZÁMUNKBAN MEGJELENT REJTVÉNYÜNK MEGFEJTÉSE ÉS NYERTESEINK

A megadott határidőig helyes megfejtést beküldő olvasóink közül sorsoltuk ki a három szerencsés nyertest.

1. PGYURI

2. KERTÉSZ LÁSZLÓ

3. MEZEI RÓBERT

Gratulálunk!

ZX Spectrum jákékkazetta nyereményüket postán juttatjuk el számukra.

P	Y	J	A	M	A	R	A	M	A
E	S	E	T		S	U	G	Á	R
N	A	R	A	N	C	S		R	C
E	Y	E		M	O	N	A	C	O
T	E	M	P		T	Y	L		F
R		I	L	K		A	L	B	Y
A	L	Á	Z	A	T		E	R	E
T	A	S		H	I	N	G	I	S
O	P		I	L		A	R	L	O
R	U	N	F	O	R	G	O	L	D



FANZI

SOLD OUT

EXCLUSIVE FAN CLUB OFFER!
FREE TURTLES
EXCLUSIVE I.C.D. SPECIAL OFFER!
CAMERA!
FOR DETAILS - Offer valid while stocks last