

Így készült a Monty on the Run Videoton TV-Computerre

Neki sem állunk

A 2019-ben, 2020-ban és 2021-ben az általunk megnyert TVC Játékfejlesztői versenyek valamint a 2022-es dobogós helyezésünk után úgy döntöttünk, 2023-ban nem indulunk a versenyen.

Négy év alatt öt színvonalas játékot hoztunk össze, négyet full assemblyben, egyet C-ben asm rutinokkal támogatva és úgy gondoltuk, szünetet tartunk (aminek nyilván technikai, szellemi és érzelmi okai is voltak). Olyannyira komolyan gondoltuk ezt, hogy 2024-ben sem szándékoztunk indulni. De aztán Tamás nem hagyott nyugton, rágta a fülem, hogy valamit adjunk már be... És mindez *november közepén* jutott eszébe!

A projekt

Na jó, de mégis mit csináljunk, főleg ennyi idő alatt? Hogy tudjuk tartani a Doberdo Brothers színvonalat? Túl egyszerű nem lehet (nyilván), bonyolultra meg egyszerűen nincs időnk (nyilván).

Egy darabig válogattunk a félig-meddig meglévő, fiókban heverő és egyéb projekt ötletek közül, de úgy láttuk egyik sem lesz megfelelő. Ezek után Tamás rákérdezett, beleférne-e a Monty? Akkor úgy láttam, hogy ha mindent félreteszünk és idén betiltjuk a karácsonyt a hozzá tartozó egyéb járulékos ügyekkel együtt, emellett 48 órára húzzuk fel a napokat, akkor *talán* igen. Ez rögtön négy lehetetlen pontot jelentett, de egy próbát talán megér...

Úgy gondoltuk, hogy ismét minimum Commodore 64-es színvonalra törekszünk (ahogy az IK+ és Impossible Mission esetében is). A program 256x240-es felbontásban és.. várjunk csak.. jaaa, négy színben fog futni... De a biztonság kedvéért számoljuk csak gyorsan össze a színeket a kezdőképernyőn!



Kezdőképernyő (TV-Computeren és Commodore64-en)

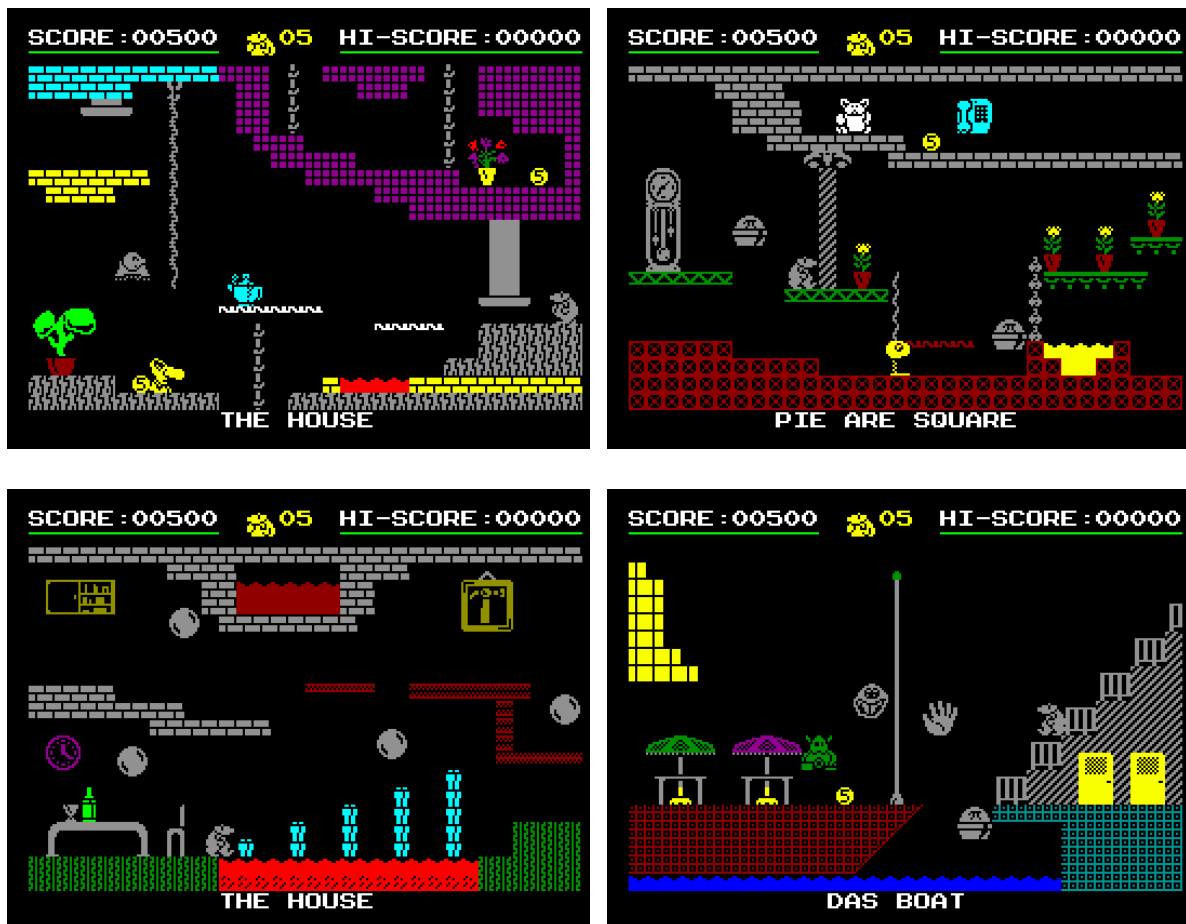
Tile-ok és Sprite-ok

Ebben a projektben a szobákat felépítő tile-okat és sprite-okat a játék ZX Spectrumos változatából emeltük ki komplett bitmapként, amelyet aztán Photoshopban vágtunk darabokra (természetesen mindet egyesével, saját névvel kimentve). Ezek a fájlok lettek beimportálva DevToolba, ahol funkciójuktól függően, bizonyos logika szerint sorozatokba rendeztük őket (amit többször is meg kellett tennünk). Persze részben át is rajzoltuk őket, a felmerült technikai és esztétikai igényeknek megfelelően.

Első körben 326 sprite került be az atlaszba, minden elem tisztán fekete-fehérben, négy színű üzemmódra beállítva (a harmadik színnel festve). Már ebből látszott, hogy nem egyszerű projektről lesz itten szó...

Pályák

A bejárható szobákat a neten található, viszonylag pontosan összemásoltatott térképek alapján készítettük el. Photoshopban, 8x8 pixeles gridet beállítva, az adott pálya képét kinagyítva, kézzel építettük fel mind az 50 darab, 32x20-as pályát, a DevTool pályaszerkesztőjében. Mikor az összes pálya bekerült, a tool attribútum szerkesztőjével minden egyes tile-ra beállítottuk, hogy melyik paletta színnel kerüljenek majd kirakásra.



Néhány pálya

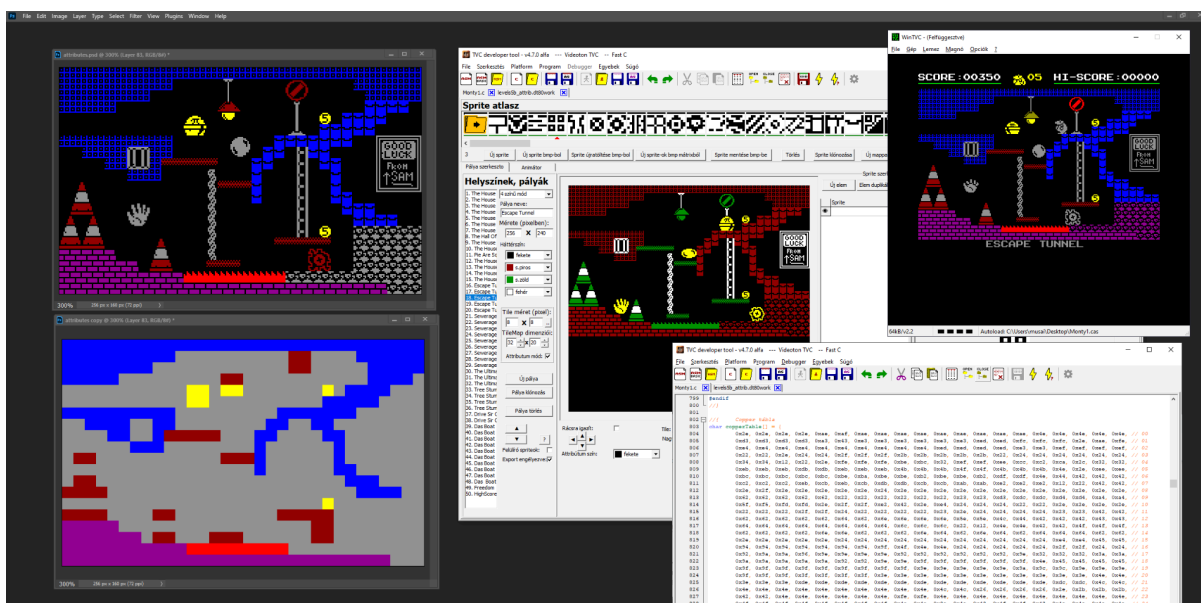
Copperlisták

A játékmenet alatti palettából két szín fixre (0.: fekete, 3.: szürke) lett beállítva, az 1. és a 2. palettaszíneket pedig fentről lefelé 8 pixelenként (tile soronként) átdefiniáltuk (és a forrásba kézzel beírt "db" tömbökben tároltuk (pontosan 50x20x2, azaz 2000db szín beállításáról van szó)). Ez tette lehetővé, hogy négynél több színt tudjunk megjeleníteni képernyőnként (ez az eljárás az Amigás copper listákhoz hasonlít, ezért mi is úgy neveztük el ezt a módszert). Fontos érteni, hogy itt nem csupán a játéktér és mondjuk a HUD közötti, hanem közvetlenül a játéktér teljes magasságában lévő paletta színváltásokról van szó.

Amíg a copperlista kezelő és (nagyon mélyen a rendszerbe integrált) megjelenítő kód készült, mind az 50 pályáról fekete-fehér screenshotot készítettünk a tool-ból, amelyeket egyesével photoshopba töltöttünk és úgy színeztük át őket (tile-felbontásban), ahogyan majd TVC-n is meg fognak jelenni. Erre a plusz munkafázisra feltétlenül szükség volt, hogy később semmiképp ne legyen szín-ütközés TVC-n és ezek alapján lettek beírva a copperlistába is a színek.

A feladat egy laikus számára egyszerűnek tűnik, de nagyon nem az és nem kis odafigyelést igényelt: miközben az (eleve kihívást jelentő) izléses sokszínűsége törekedtünk, arra is ügyelnünk kellett, hogy a felvehető tárgyak minden pályán fehérek, a pénzermék pedig sárgák lehessenek. Csak érzékeltetésként, hogy ne feledjük: minden tile sorban maximum négy szín lehet összesen, ebből a fekete és szürke fix, így csak a maradék kettővel lehet variálni. További korlát volt, hogy a függőlegesen mozgó spriteok nagy %-ban szürkék lehettek csak, mert a copperlistával csak függőlegesen és csak tile soronként manipulálhatók a színek.

Ezzel a technikával (és az eredeti pályák finom és igencsak gondolkodós módosításaival) el tudtuk érni, hogy minden pályán átlagosan minimum 6-7, vagy akár több szín legyen látható (a kezdőkép rögtön 8 színű, a legutolsón pedig 14-et jelenít meg az engine egyszerre).



Balra: pálya előszínezés Shopban, közepen: ahogy a DevToolban kinéz három paletta színnel
jobbra lent: a copperlista és jobbra fent: ahogy emulátorban kinéz

NPC-k, tárgyak és tag-ek

A pályákra kerülő összes interaktív elemet nem tile, hanem “stuff”-ként tettük be. Minden ilyen stuffnak saját tag-értéket állítottunk be, amelyek tartalmazzák az objektum színét, sebességét, x/y mozgás irányát, a mozgás határértékeit, elindulási irányát és kezdő pozícióját. Ez a tag egy kétbájtos, “bit-kapcsolós” érték, amelyhez külön kalkulátor programot írtunk (így állítottuk be az összes felvehető tárgyat, npc-eket és préseket is). Mivel minden pályán átlagosan három npc és két pénzérme/egyéb tárgy van, nagyjából kiszámolható, mekkora munka volt ezeket kiszámolni (nagyságrendileg kb. 1500 érték beállításáról van szó).

NPC	
Szín:	<input type="checkbox"/> fehér
Mozgás irány:	Függőleges
Mozgás kezdete fentről:	6
Vége:	12
Kezdő mozgás irány:	Felfelé
Sebesség:	2
TAG:	17719

Prés	
Szín:	<input type="checkbox"/> fehér
Mozgás táv (tile-ban):	3
Várakozás (game cikl.):	10
Random faktor (1-63):	10
TAG:	5419

Tag kalkulátor

Animációk

Az animációkat három fő csoportba osztottuk: az első csoportba Monty animációi tartoznak: (futás, mászás, hátirakétázás, autózás, cseppes szétrepülés, stb), a másodikba az NPC-k négy fázisú előrefelé játszott és három fázisú ping-pong animációi, a harmadik csoportba pedig a pályán lévő egyéb, helyet nem változtató objektumok animációi (hullámos, tuskés “vizek”) kerültek. Ehhez a csoporthoz egy igen speciális kód készült, ami mindössze két pixelsort mozgat meg mindegyik víz-tile tetején (ez sem mondható szokványos effektnek játékokban TVC-n).

Igen ám, de nagyon gyorsan fogyott a hely és át kellett gondolni, melyik animációkból tudunk “válni”. Így sok “négy kockásból” lett három, sőt néhány NPC mindössze két kockás animációra lett redukálva. Persze mindezt úgy kellett csinálni, hogy a játékosok többsége ne vegye észre mindezt, még az eredeti verziókhöz hasonlítva sem.

Egyébként pont a helyhiány miatt lett menet közben kifejlesztve az alpból két színben tárolt, de a képernyőn már négy színben megjelenítő sprite rutin is (amit eredetileg csak statikus a tile-ok kirakására akartunk használni). Nem volt mese, kénytelenek voltunk a memória oltárán feláldozni némi gameplay sebességet.

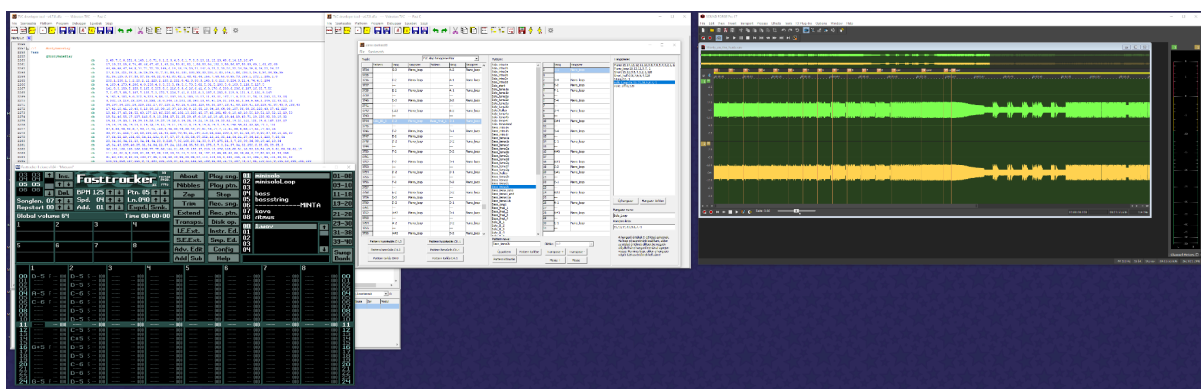
Zene

Ahogy az IK+ esetében, a Monty on the Run-ban is teljes hosszában megírtuk a zenét. Azt gondolhatnók, hogy az utóbbi években ez könnyebbé vált, de 2024-ben is épp olyan küzdelmes volt mint 2019-ben. Pontosan ugyanúgy, kézzel kellett megírni minden track-patternnt, beállítani a hangszereiket, aztán exportálni a zene datát, azt bemásolni a C forrásba, lefordítani, a cas-t bedobni az emulátorba és alt+w-vel felgyorsítva-lelassítva nagyjából "odatekerni", amelyik részét meg akartuk hallgatni a zenének, hogy egyáltalán hogyan szól. Hozzávetőlegesen 2000-szer végeztük el a fenti folyamatot mire a zene kész lett.

Mindehhez egyszerre két DevTool, Sound Forge, TVC emulátor és Fasttracker futott a háttérben napokig, hajnali három órás fekvésekkel "tarkítva" a munkát.

Mindehhez vegyük hozzá a zene .sid-ből .txt-be és .wav-ba exportját, a .txt feldolgozását, a .wav zenei és technikai nyomkövetését Forge-ban, a zene írása közbeni, adott ponton lévő bájtspórolás/zeneiség előbbrevalóságának mérlegelését, kísérletezéseket, hogy egyáltalán milyen effekt hogyan valósítható meg, stb, így a fentiekkel már könnyebben elképzelhető mibe is került a zene felépítése - a nulláról.

Így lett a zene 7db hangszerrel, 75db patternbe rendezett dallamsorból összerakva, ami együtt összesen 4383 note pozíció-sort tesz ki.



Balra: az egyik DevToolban a C forrás, a tömörített zene kóddal, alatta a FastTracker2
középen: a másik DevTool a zeneszerkesztővel, jobbra: SoundForge a SID-ből wav-val

Változtatások

Természetesen több dolgon is változtatnunk kellett menet közben. Az egyik például a Monty's freedom kit-et érintette. Az eredeti C64-es és ZX-Spectrumos változatban az ebbe a kit-be való tárgyakat a címképen kellett összeválogatni. Mivel ezektől a tárgyaktól drasztikusan függött a végigjátszhatóság (értsd: felesleges szívatás volt), ezeket mi inkább a pályákon helyeztük el (az egyébként szükségtelen tárgyak kihagyásával), könnyen láthatóan és felvehetően. Ez gördülékenyebb gameplayt, kevesebb memória használatot és kevesebb kódolást is jelentett (Ezen felül és ugyanilyen okokból más változtatásokat is eszközöltünk, amiket itt hosszú lenne felsorolni).

DevTool fejlesztés

A fejlesztés alatt “természetesen” drasztikusan hozzá kellett nyúlni a DevToolhoz (például a pályaeépítő és a pályaelemeket beállító/manipuláló eszközeihez - a négy színű módot támogató eljárások és adattárolások is jobbára a Monty alatt lettek hozzáfejlesztve) és a gépi kódú library rutinyűjteményhez is, vagyis a játék mellett, azzal párhuzamosan mind programozástechnikailag, mind játék kódolástechnikailag is fejlődött a teljes rendszer (így került be például a zenét zx7-tel tömörítő exportálás is). Mindezekhez persze tegyük hozzá a közben felhalmozódó hibák javításait is.

Statisztika

A végére pedig egy kis Monty on the Run statisztika: A C forráskód 4500 soros, ami assembly forrásban mintegy 18000 sort tesz ki (kommentekkel együtt). A játékhoz 50 pálya készült, amiken összesen 2000 színbeállítást végzünk. A pályák tömörítetlenül 65782 (zx7-tel 8750), a pályabejárás térkép 200 elemű, a grafika 7867, a 6 perces, kétcsatornás zene tömörítetlenül 3580 (zx7-tel 1720) bájtot foglalnak. A beletett munkaórák számának kiszámolására ha lehet, nem vállalkoznánk...

Utószó

A Monty on the Run a 1985-ös év egyik “sztárjátéka” volt annak idején. Ez a program is “kivárta”, hogy több mint harminc év múlva elkészítsük a magunkét. Ahogy a többi játékunkba (InterKarate Plus, PacMan, Impossible Mission, BattleField, Fitter, stb), ebbe is rengeteg energiát és munkát tettünk, kifacsarva a gépből, amit csak lehet (ahogy mindig), csurig töltve a memória minden szegletét, szép és hasznos dolgokkal. Reméljük mindenki aki játszik vele, legalább annyira élvezni fogja, mint mi az elkészítését.

Ajánljuk is, mert keményen megszenvedtünk vele! 😊

