PIX37 handbók

Ritstjóri, síðan kóði. Ótengd eintak af bókinni á pix37.party/docs.

Ritstjóri

PIX37 DEV er ritillinn. Þú teiknar, skrifar og hlustar á leikinn hér, og flytur síðan út leikjahylki eða sjálfstæða útgáfu. Það getur líka spilað leikjahylki og það getur opnað PIX37.PARTY til að hlaða niður leikjahylkjum og nota vörulistann. PIX37 DEV EDUCATIONAL er sami ritillinn og hann er ókeypis. Hann hefur engar útgáfur, engan útflutning á leikjahylkjum og engan aðgang að PIX37.PARTY. PIX37 Console er spilarinn. Það spilar `.p37` leikjahylki og þú opnar PIX37.PARTY til að hlaða niður leikjahylkjum og skoða vörulistann.

Verkefni og leikjahylki

`.pix37` skrá er verkefnið: frumflipar, spritesblöð, kort, hljóð og gagnatöflur. `.p37` skrá er leikjahylkið sem spilari keyrir: þéttpakkað bætikóði.

Vistaðar skrár og afrek eru í höfundarmöppunni við hliðina á appgögnunum. Í Windows lítur það svona út: `C:\\Users\\Ada\\AppData\\Roaming\\PIX37\\Mira\\Pong.p37s` fyrir vistaða skrá og `C:\\Users\\Ada\\AppData\\Roaming\\PIX37\\Mira\\Pong.ach` fyrir afrek. Í Linux eru sömu skrárnar `~/.local/share/PIX37/Mira/Pong.p37s` og `~/.local/share/PIX37/Mira/Pong.ach` . Ada er aðgangurinn á tölvunni. Mira er höfundarnafnið og það nafn er mappan. Pong er vistunarraufin, sem er sjálfgefið leikjatitillinn. `save` skrifar `.p37s` . Fyrsta afrekið sem opnast skrifar `.ach` . Ef skrárnar eru þegar til staðar les keyrslan þær þegar leikjahylkið ræsist.

Þróunarvalmynd

Escape opnar forritaravalmyndina. Escape á meðan leikur er í gangi fer aftur í ritilinn. Staðfestið með Enter eða A hnappinum (sjálfgefið V). Til baka er B hnappurinn (sjálfgefið C).

Þegar ekkert verkefni er hlaðið er valmyndin: BÚA TIL VERKEFNI, HLAÐA VERKEFNI, BYGGJA, HLAÐA Leikjakassa, TENGJA VIÐ PIX37.PARTY, STILLINGAR.

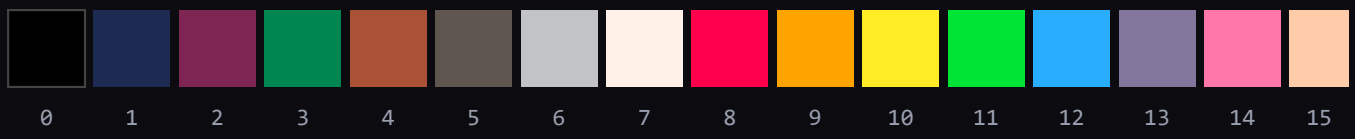
Þegar verkefni hefur verið hlaðið er það: VISTA VERKEFNI, HLAÐA VERKEFNI, BÚA TIL VERKEFNI, EIGNIR, BYGGJA, HLAÐA Leikjakassa, TENGJA VIÐ PIX37.PARTY, STILLINGAR.

Ctrl+S vistar opna verkefnið. Ctrl+R keyrir það.

Búa til verkefni

„CREATE PROJECT“ biður um leikjatölvuna áður en þú byrjar að búa til verkefnið.

- **MYNDBAND** — 1-BIT (2 litir), 2-BIT (4), 4-BIT (16), 8-BIT (256).
- **HLJÓÐ** — sýnishornsdýpt hljóðgervilsins: 16, 8, 4, 2 eða 1 bit.
- **SKJÁR** — forstillingar eru meðal annars 64×64, 128×128, 160×90, 256×256 og 320×180. SÉRSNÍÐIN stærð gerir þér kleift að velja hvaða stærð sem er frá 1×1 til 512×512.
- **LITASPELLUR** — litabankar. Fyrsti bankinn byrjar sem 16-lita settið hér að neðan. LOAD PALETTE notar innbyggt sett: PICO-8 16, PICO-8 32, Game Boy, 1-bit, CGA, Sweetie-16 og lengri lista tekinn frá Lospec (Resurrect 64, Apollo, Endesga 32 og 64, CC-29, SLSO8, Pear36, Oil 6, Duel, AAP-64, Lost Century og fleiri). Þú getur líka hlaðið inn litaspjaldaskrá eða hlaðið inn þínu eigin PNG og tekið litaspjaldið úr þeirri mynd.

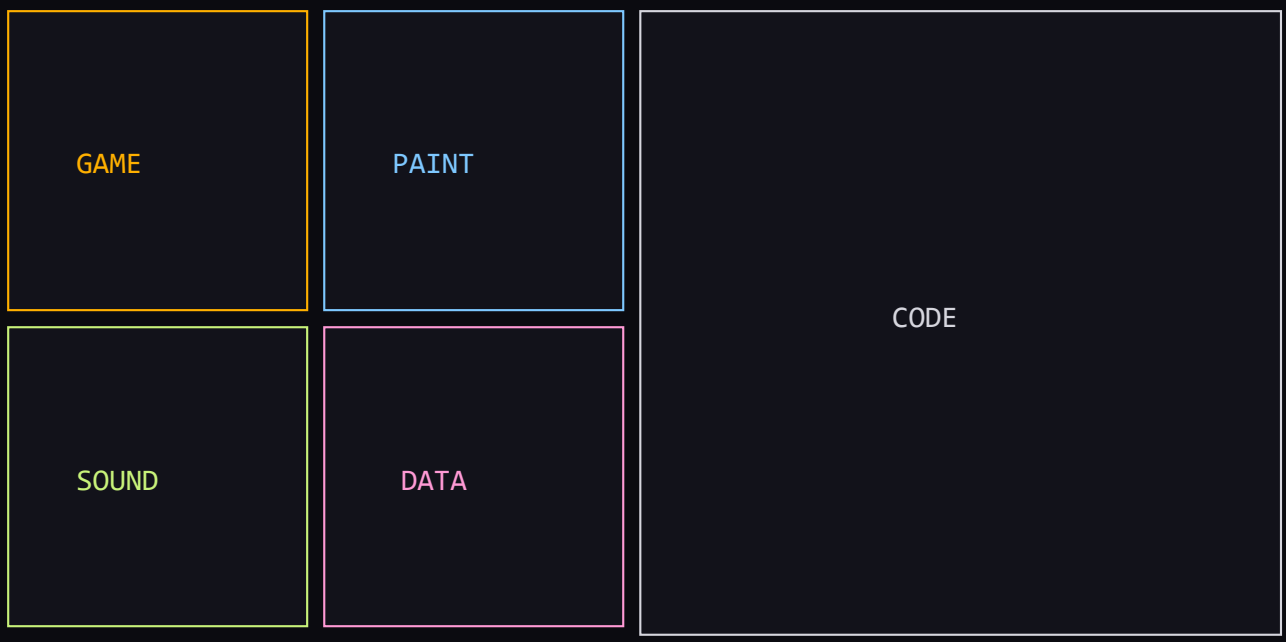



```
lette — black, navy, purple, green, brown, gray, silver, white, red, orange, yellow, lime, blue, indigo, p
```

Litavísitölur sem `cls` , `pset` , `spr` og hinir nota. Þú getur endurmálað hvaða vísitölu sem er í litatöfluritlinum.

Fimm spjöldin

Ritstjórinn er GAME, PAINT, CODE, SOUND og DATA, í þeirri röð. Kortið er inni í PAINT. SFX og lög eru inni í SOUND. Smelltu á spjald til að vinna þar. Þegar CODE er í fókus hunsa SOUND og DATA bókstafatakkana svo að innsláttur spilar ekki líka á píanóið.



> Heildarútlit, skrifað `1:1:4:1:1` . Hver tala segir til um hversu stóran hluta röðarinnar spjaldið tekur: LEIKUR, MÁLUN, KÓÐI, HLJÓÐ, GÖGN. Lítill skjár er staðsettur í miðjunni í 128×128 reit. Skjár stærri en 128 pixlar víkkar LEIKJUDálkinn.

STILLINGAR → ALÞJÓÐLEGT → ÚTSKIPTI færir í gegnum eftirfarandi stillingar:

Kóði	Það sem þú sérð
`1:1:4:1:1`	Allt. LEIKUR og HLJÓÐ staflað, MÁLNING og GÖGN staflað, KÓÐI hægra megin.
`1:1:4:1:1 2X`	Sama uppröðun. KÓÐI er teiknaður með tvöfaldri þéttleika, þannig að tvöfalt meiri texti passar.
`1:0:1:0:1`	LEIKUR, GÖGN, KÓÐI.
`1:0:1:0:1 2X`	Sama fyrirkomulag, með þéttari KÓÐANUM.
`0:1:1:1:1`	Engin forsýning á leiknum. MÁLNING, KÓÐI, HLJÓÐ, GÖGN.
`0:1:1:1:1 2X`	Sama fyrirkomulag, með þéttari KÓÐANUM.
`0:1:4:1:2`	Stúdíó. Breiður KÓÐI, hærri GÖGNARÆMA, MÁLNING og HLJÓÐ.
`0:1:4:1:2 2X`	Sama fyrirkomulag, með þéttari KÓÐANUM.
`0:0:4:0:4`	Aðeins KÓÐI og GÖGN, bæði stór.
`0:0:4:0:4 2X`	Sama fyrirkomulag, með þéttari KÓÐANUM.
`0:1:4:1:0`	MÁLNING, HLJÓÐ og breiður KÓÐI.
`0:1:4:1:0 2X`	Sama fyrirkomulag, með þéttari KÓÐANUM.
`0:4:0:0:0`	Aðeins MÁLNING, 256×256, ein pixla af blaðinu á móti einum pixla af striga.

`2X` röð heldur sömu spjöldum. CODE spjaldið er teiknað í tvöfaldri upplausn og minnkað aftur í sama reit, þannig að textinn, flipanir og innfellingarlínan eru helmingi minni og tvöfalt meira af frumkóðanum passar á skjáinn.

LEIKUR

Forskoðun á leikjakassanum sem er í gangi. Smelltu á LEIKJASPJALDINN eða ýttu á Ctrl+R til að keyra forskoðunina. STILLINGAR → LEIKJASPJALD er þar sem spjaldið er stillt:

- **Forskoðun** — óvirkur litur, samfelld eða sjálfvirk endurhleðsla þegar þú breytir.
- **Spilaðu í nýjum glugga** — leikurinn opnast við hliðina á ritlinum.
- **Sýna** — forskoðun sem RGB888 eða RGB565.
- **LEIKUR 64** — 64×64 skjár er annað hvort miðjaður 1:1 í reitnum eða 2× minnkaður til að fylla hann.
- **Shaders** — CRT og síur í leiknum, aðskildar frá síunum í ritlinum.
- **Litur** — bakgrunnur spjaldsins.

MÁLNING

Sprite-blöð og, á kortahlið sama spjalds, flísakort. Blöð geta verið stærri en skjárinn. Forskoðunarstærðir sprite eru 8, 16, 32, 64 og 128 (128 aðeins þegar blaðið er að minnsta kosti 128 px). Skrunaðu músarhjólinu yfir forskoðun sprite til að stækka hana. Pensilstærðir eru 1, 2, 3 og 4.

Lykill	Tól
B	Blýantur
M	Rétthyrningur valinn
Q	Lasso
S	Stimpill
G	Fötu
Ég	Augndropari
V	Færa
L / Shift+L	Lína / ferill
U / Shift+U	Holur rétthyrningur / holur sporbaugur
F / Shift+F	Fylltur rétthyrningur / fylltur sporbaugur
D	Hringdu í
[ ]	Minni / stærri bursti
Ctrl+Z / Ctrl+Y	Afturkalla / endurgera. Ctrl+Shift+Z endurgerir einnig.
Ctrl+C / Ctrl+X / Ctrl+V	Afrita, klippa, líma
Ctrl+A	Veldu striga

Kortið notar sömu hugmynd, með færri verkfærum: B blýantur, M rétthyrningur velja, S stimpill, V færa og Ctrl+Z, Ctrl+Y, Ctrl+C, Ctrl+V. Haltu Alt inni til að stækka áferð kortsins. Pensilstærðin á kortinu helst með málningarverkfærunum.

Hvert sprite hefur átta fánabita. Paint setur þá á sprite-ið; `fget` og `fset` lesa þá í leiknum.

STILLINGAR → MÁLNINGARRITLINN er þar sem þessi spjald er stillt:

- **Litatöflu** — hvaða banki er dreginn, auk þess að hlaða og vista.
- **Forskoðun** — RGB888 eða RGB565.
- **Penslabendillinn** — bendillinn á meðan þú málar.
- **Pixlasnap** — hvernig pensillinn smellir á pixla.
- **Stór bursti** — stór bursti smellpassar frá efra vinstra horninu eða frá öllu fótspori sínu.
- **Laukskinn** — draugur úr öðrum römmum og í hvaða átt sá draugur snýr.
- **Aðdráttur korts** — raunverulegir pixlar eða frjáls aðdráttur.
- **Litir** — litirnir í notendaviðmótinu fyrir málningu.

KÓÐI

Flipar eru aðskildar skrár innan verkefnisins. Smelltu á flipanafn til að endurnefna hann; Enter staðfestir. Nokkrir flipar sem hver um sig skilgreina `_init` , `_update` eða `_draw` eru saumaðir saman í fliparöð: fyrsti flipann keyrir fyrst, síðasti flipann keyrir síðast. Kóði sem er ekki innan í krók verður hluti af `_init` . Leikjahylki án króka er meðhöndlað sem eitt `_draw` .

Lykill	Aðgerð
Örvar	Færðu innfellda merkið. Shift eykur valið.
Ctrl+Vinstri / Hægri	Með orði.
Heim / End, Ctrl+Heim / End	Upphaf eða endi línu, eða tab.
Flipi	Settu inn tvö bil. Samþykktu nafnið í útfyllingarlistanum.
Sláðu inn	Ný lína, inndregin eins og sú fyrri. Tekur við útfyllingu.
Ctrl+C / X / V	Afrita, klippa, líma.
Ctrl+Z / Ctrl+Y	Afturkalla / endurgera.
Ctrl+A	Veldu flipann.
Ctrl+D	Afritaðu línuna eða valið niður á við.
Ctrl+/	Skrifa athugasemd eða taka af athugasemd með `--` .

Ctrl+F	Finna. F3 hoppar á næsta högg. Escape lokar stikunni.
Ctrl+G	Farðu á línunúmer.
Flótti	Lokar fyrst leitarvalmyndinni, fara á valmyndina eða vísbendingalistanum. Önnur Escape-lykill yfirgefur ritilinn fyrir forritaravalmyndina.

Vísbendingar birtast þegar þú slærð inn nafn falls. Enter eða Tab setur inn nafnið og `(` ef LOKIÐ (er virkt í stillingum. STILLINGAR → KÓÐARITL:

- **Leit** — Ctrl+F í virka flipanum eða á öllum flipa.
- **Texti hár** — klippa tákn niður í 5 px eða sýna allt táknið.
- **Strengjastærð** — leturstærð strengjabókstafa.
- **Leturgerð athugasemda** — 3×5, eða Fusion leturgerð núverandi notendaviðmótsmáls.
- **Ljúka (** — hvort lokið sé að skrifa svigainn.
- **Vísbendingar um kóða** — kveikt eða slökkt.
- **Vísbendingartungumál** — tungumál valmyndarinnar, eða enska.
- **Bílbil** — aukalega hliðrun á tómum línum.
- **Rennu** — línunúmer.
- **Litur** — bakgrunnur, athugasemdir, tabs, tölur, virkjar, leitarorð, strengir, vélarföll, þín föll, breytur, sviga, bendipunktur, val, villulína, viðvörun.

HLJÓÐ

Spjaldið hefur tvo flipa. SFX er eitt mynstur: stutt röð af nótum. SOUND er lag: það setur þessi mynstur á átta rásir. TIC er hversu mörg tík eitt skref bíður. Klukkan er 120 tík á sekúndu og eitt skref er sextánda nóta, svo 15 tík eru 120 slög á mínútu. Hærra TIC er hægara mynstur. Tvö mynstur haldast í takt þegar annað TIC skiptir hinu: 15 og 30 gera það, 15 og 8 ekki.

Píanóið spilar þegar nóta er valin með músinni. Hvítu takkarnir eru ZXCVBNM (CDEFGAB). Svörtu takkarnir eru SDGHJ (C# D# F# G# A#). Færðu músarbendilinn yfir hvaða gildi sem er og sláðu inn tölu, eða flettu músarbendilinn yfir það og skrunaðu með hjólinu. Tónhæðartölur í kóða eru frá 0 (C-0) upp í 63 (D#5).

Í einu mynstri eru LOOP og LEN sömu tölurnar tvær. OUT fyrir ofan 0 hringsólar frá IN til OUT, og titillinn er LOOP. OUT við 0 og IN fyrir ofan 0 er LEN: svo margar nótur spilast frá upphafi, síðan endar mynstrið og titillinn er LEN. Báðar við 0 spilast í hverju skrefi.

Hversu mikið af hljóðgervilnum þú getur breytt fer eftir bitadýptinni sem valin er fyrir leikjakassann. STILLINGAR → SFX EDITOR stillir þá dýpt, hvort nóta færist sjálfkrafa áfram (slökkt, nótnakeðja eða sviðskeðja), hvort bil endurræsir spilun eða skiptir á milli ræsingar og stöðvunar, og liti ritlistans.

16-bita

Hvert skref hefur nótu, hljóðstyrk, bylgju og áhrif. Hljóðstyrkurinn er frá 0 til 7. 0 er hljóðlátt.

Bylgjur:

- **0 þríhyrningur**
- **1 hallað sög**
- **2 sög**
- **3 fermetrar**
- **4 púlsar** — þröngur púls
- **5 líffæri**
- **6 hávaði**
- **7 fasa**
- **8 PWM** — ferningur sem breiddin færist
- **9 púlsar 25%**
- **10 sinus**
- **11 málmhljóð**
- **12 FM bjalla**
- **13 plokk**
- **14 ofursög**
- **15 PWM (C64)**

Áhrif á einni nótu:

- **0 ekkert**
- **1 glæra** — tónhæð og hljóðstyrkur renna yfir í næstu nótu í gegnum þetta skref
- **2 titringur** — sveiflur í tónhæð
- **3 dropi** — tónhæðin þagnar, venjulegt spark
- **4 hverfa inn**
- **5 dofna út**
- **6 arpeggio** — gengur með þessari nótu og næstu þremur, einn á hverjum 8 tíkkum. Við TIC 8 eða lægri, á hverjum 4 tíkkum.
- **7 hröð arpeggio** — sömu fjórar nóturnar, á 4 tíkkum fresti, eða á 2 tíkkum fresti þegar TIC er 8 eða lægra.

Fyrirsögn mynstursins, fyrir ofan athugasemdirnar:

- **TIC** — tikk á skref, frá 1 til 255. 15 er 120 slög á mínútu. Sláðu inn töluna eða taktu hana um eitt skref.
- **LYKKJA INN / ÚT** — með OUT fyrir ofan 0, þá lykkjast mynstrið frá IN til OUT. Ef OUT er ekki lengur stillt á 0 verður titillinn LEN: IN er hversu margar nótur spilaðar og 0 þýðir allt mynstrið.
- **HLJÓÐI** — allt mynstrið verður að hávaða.
- **CRUSH** — vinsæl útgáfa af hverri bylgju.
- **DTUNE** — 0 af, 1 léttur viðlag, 2 sterkur viðlag.
- **RVERB** — endurómur: 0 slökkt, 1 stutt, 2 langt.

- **DAMPN** — lágtíðnisía: 0 slökkt, 1 ljós, 2 sterk.
- **LEVEL** — hljóðstyrkur alls mynstursins, 0 til 7.
- **SVINGA** — stokka. Undarleg skref bíða aðeins lengur, jafnvel skref aðeins skemmri.
- **SKREF** — hversu mörg skref mynstrið notar: 32, 24, 16 eða 12.

8-bita

Skref hefur nótu og bylgju. Fjórar bylgjur eru ferhyrningur, þríhyrningur, sag og hávaði. Hausinn hefur TIC, LOOP IN / OUT, FX og LEVEL. TIC er eitt af 1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 30, 40, 60. FX er ein áhrif fyrir allt mynstrið: 0 engin, 1 vibrato, 2 arpeggio, 3 fade out. LEVEL er 0 til 3.

4-bita

Skref er bara nóta. Í hausnum eru TIC, LOOP (virkt eða slökkt), WAVE (sömu fjórar bylgjur) og OCT (í hvaða áttundu og nóturnar eru). TIC er ein af 4, 6, 8, 10, 12, 15, 20, 30.

2-bita

Þrep er tómt, tromma, högg eða troðningur. Hausinn hefur TIC og BASE. TIC er 8, 15, 16 eða 30. BASE stillir hversu hátt tromman og höggið eru.

1-bita

Skref er nóta. TIC er 8, 15, 16 eða 30, sömu fjórir hraðar og 2-bita.

Lög

HLJÓÐ-flipinn er opna lagið. Átta dálkar eru rásirnar. Reitur er mynsturkenni. Ef það er teiknað tómt þýðir það að rásin er hljóðlaus í þeirri röð. X-dálkurinn er skipunin: halda áfram, hætta, lykkjabyrjun eða lykkjalok. Stop lýkur laginu eftir röðina. Lykkjulok hoppar aftur til síðasta lykkjabyrjunar.

Röð endar þegar mynstrið lengst til vinstri með LYKKJU UT við 0 klárast. Mynstur lengra til hægri lengja ekki þá röð: lengra er klippt, styttra byrjar aftur á fyrsta skrefi þar til röðin endar. Mynstur með LYKKJU UT fyrir ofan 0 lykkjast á milli LYKKJU INN og LYKKJU UT og lýkur ekki röðinni. Ef öll mynstrin í röðinni lykkjast, þá helst röðin þar.

Flýtileiðir

Lykill	Aðgerð
Píanólyklar	Skrifaðu athugasemd um skrefið sem valið var með músinni.
Upp / Niður	Breyttu gildinu undir bendilinn.

Rými	Spilaðu opna SFX, eða opna lagið úr fyrstu röðinni. Byrjaðu/stöðvaðu eða endurræstu, úr stillingum SFX EDITOR.
Ctrl+Z / Ctrl+Y	Afturkalla / endurgera.
Ctrl+C / Ctrl+X / Ctrl+V	Afrita, klippa, líma valið.

GÖGN

Nafngreindar töflur. Hver tafla er raðir og dálkar. Úr kóðanum nota `dget` , `dset` , `dstr` , `drows` og `dcols` þessar töflur. Rað- og dálkanúmer í kóða byrja á 0.

Dálktegundir, raðaðar í hausnum:

- **bool** — einn biti.
- **2** — tveir bitar.
- **4** — fjórir bitar.
- **8** — bæti.
- **16** — 16 bitar, fyrir auðkenni og stærri tölur.
- **spr** — auðkenni fyrir sprite.
- **str** — texti, teiknaður með kerfisleturgerðinni.
- **str8** og **str10** — sami textinn, teiknaður hærri.
- **f16** og **f32** — hálf-nákvæm flotstöng og full flotstöng.

Með því að halda bendilnum á dálkhaus breytir Ctrl+skrun gerð dálksins. Einföld skrun færist yfir dálka eða niður raðir, hvort sem STILLINGAR → GAGNARITLINN velur. Ctrl skiptir um stefnu, og þess vegna breytir Ctrl+skrun á haus gerðinni þegar hjólið er stillt á dálka.

Örvatakkar færa sig. Enter breytir reit. STILLINGAR → GAGNARITILI:

- **Hjól** — skruna dálkum eða skruna röðum.
- **Litir** — ristið, hausinn og valið.

Eignir

ASSETS flytur inn og út þann hluta sem er opinn, ekki allt verkefnið. Sprites nota sprite-ið sem er opið í PAINT, á blaði sem er opið. Kortið er kortið sem er opið. SFX er mynstrið sem er opið. Tónlist er lagið sem er opið. Gögn eru taflan sem er opin. Kóði er undantekningin: hver tafla er skrifuð og hver tafla er lesin til baka.

Góð	Skrá	Hvað hreyfist
Sprítur	`.png`	Opna sprítið.

Kort	`.p37map`	Opna kortið.
SFX	`.p37sfx`	Opna mynstrið.
Tónlist	`.p37mus`	Opna lagið. Mynsturnúmerin eru óbreytt. Flyttu út SFX sérstaklega ef þú þarft á þessum hljóðum að halda.
Gögn	`.p37dat`	Opna borðið.
Kóði	`.lua`	Sérhver flipi.

Stillingar

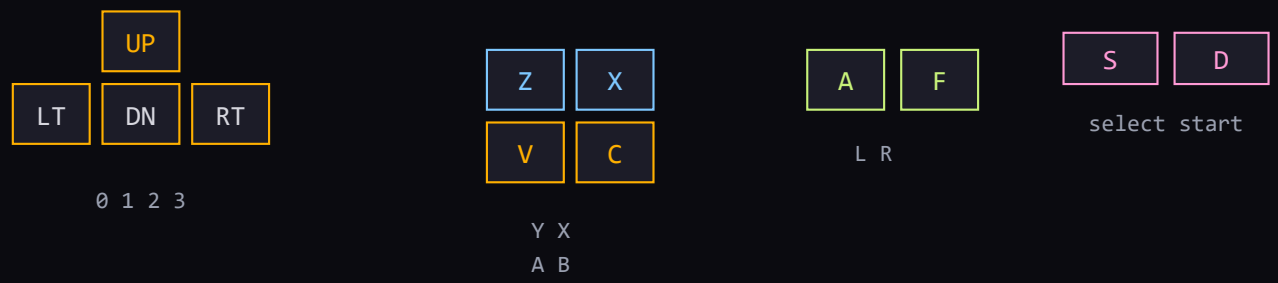
Rótarlistinn:

Síða	Það sem það geymir
PRÓFÍLAR	Vista og hlaða inn öllum stillingum. Nýja, endurnefna, skrifa yfir, eyða. Afritaðu eða límdu prófílinn sem texta.
Matseðill	Gluggi, lyklar, litir í valmyndum, valfrjáls bakgrunnur.
LEIKJASPJALD	Forskoðun og spilunargluggi.
KÓÐARITILISTARI	Hápunktar og vísbendingar.
MÁLNINGARRITI	Pensill, litatöflu, kortasýn.
SFX ritstjóri	Synth og nótnaskráning.
GAGNARITSTILTÖKUR	Hjólaátt og litir.
TUNGUMÁL	Tungumál valmyndar og vísbendinga.
LEYFI	Leyfistextar, þar á meðal Fusion Pixel letrið og SDL3.

ALÞJÓÐLEGT:

- **ÚTSKIPTI** — spjaldaröðun, að ofan.
- **RÚMMÁL** — meistarastig, 0 til 10.
- **LYKLABINDINGAR** — takkar fyrir hvern af 8 spilurum. Ýttu á takka til að ná honum. Leiðbeiningar, ABXYLR, veldu.
- **Kynna** — sleppa óvirkum römmum eða teikna alla ramma.
- **Fullur skjár** — rammalaus.
- **Endurnýja** — FPS í glugga. Skref til vinstri og hægri: 60, 90, 120 og hámark. Sláðu inn tölu allt að 1000. 0 þýðir ekkert hámark.
- **Vsync** — bíddu eftir birtingu milli ramma.

- **Kvarði** — margföldun heiltölu glugga.
- **Shaders** — CRT og síur í ritlinum, ekki í leiknum.
- **Litur** — texti valmyndar og auðkenning.
- **Bakgrunnur** — valfrjálst PNG snið, skorið í 128, fyrir aftan valmyndina. Hreinsa fjarlægir það.
- **Ritstjóri bendill** — lögun bendils.
- **Mús** — fljótandi bendill eða festur í pixla.
- **Hraði notendaviðmóts** — hversu hratt haldið leiðbeiningar endurtaka sig í gegnum lista.
- **Explorer** — skráalistar byrja á rót tölvunnar eða í möppunni þar sem exe skráin er til staðar.



Sjálfgefið lyklaborð. A er staðfesting í valmyndum. B er til baka. Í leik eru þetta hnappar 4 og 5. Velja (S) er hnappur 10, ekki flýtileið í valmynd.

Hnappur	Hlutverk	Sjálfgefinn lykill
0	Vinstri	Vinstri ör
1	Hægri	Hægri ör
2	Upp	Upp ör
3	Niður	Ör niður
4	A, staðfesta	V
5	B, aftur	C
6	X	X
7	Y	Z
8	R	F
9	L	A
10	Veldu	S
11	Byrja	D

Byggja

Veldu kerfi: Windows, Linux, Mac, Android, iOS eða Web.

- **Nafn** — skráarnafn smíðinnar.
- **Höfundur** — á Windows og Linux. Þetta er möppunafnið fyrir vistanir.
- **Auðkenni Steam appsins** — á Windows og Linux. Tómt þýðir ekkert Steamworks. Fyllt auðkenni tengir `achieve(0)` … `achieve(15)` við Steam afrek ACH_0…ACH_15.
- **Auðkenni** — pakkaauðkenni þar sem kerfið biður um eitt.
- **Fela kóða** — á skjáborðs- og farsímayfirlögnum. Það hægir á því að leikjahylkið `peek` . Þetta er ekki lás.
- **Fullskjár** — útflutti spilarinn byrjar í fullskjá.
- **Shaders** — allt að þrjú lög bakað inn í spilarann.
- **Niðurhal .p37** — Aðeins fyrir vefinn. Sýnir niðurhalsfærslu í hlévalmyndinni.
- **Táknmynd** — flís úr sprite-blaði: blað, stærð, x, y. Eða PNG-forsíða sem þú klippir.
- **Flytja út** — veldu möppuna. Valmyndin sýnir hvort BYGGING VISTAÐ eða BYGGING MISTÓKST.

Búðu til útgáfu með internettengingu virku. Splash gæti spurt netþjóninn, nafnlaust, hvort kjötkássið fyrir þessa útgáfu sé þekkt. Þessi ávísun er til staðar af tveimur ástæðum. Hún verndar mig, höfund PIX37, fyrir eintökum af tólinu sem ég gaf ekki frá mér. Hún verndar einnig þann sem bjó til leikinn. Að taka leik einhvers annars og skrá hann sem sinn eigin er að verða algengt á símamarkaðnum, eins og Play (Android) eða iOS. Leikurinn byrjar samt. Ef vélin er ótengd segir inngangurinn að leikurinn sé ekki tengdur. Þessi lína er einnig vísbending til spilara um að afritið gæti ekki verið frá löglegum uppruna. Þetta er lítil ávísun, þannig að ef leikurinn þinn var stolinn og seldur án þíns samþykkis af einhverjum öðrum, geturðu gert það í verslun eins og Play (Android), Steam eða itch.io. Netþjónninn geymir tímann sem kjötkássið var fyrst skráð, við hliðina á kjötkássinu, og breytir því ekki síðar. Fyrri tíminn er upprunalega útgáfan. Kjötkáss sem skráð er eftir það er síðari afrit.

Ég vil að þessi eftirlit verndi þá sem búa til leiki. Leikjavélin vinnur nú þegar mikið af vinnunni fyrir þá, og þetta er eitt af því. Ég skil að þetta getur verið umdeilt. Ef samfélagið ákveður að það sé betra að hætta við þessa ör-DRM, þá mun ég gera það.

Gamalt leikjatölvu úr eldri ritli getur neitað að ræsa eftir að tungumálið breytist. Opnaðu `.pix37` verkefnið í þessum ritli og flyttu út nýtt `.p37` .

Tengjast við PIX37.PARTY

TENGING VIÐ PIX37.PARTY er inni í appinu. Það er engin innskráningarsíða í vafranum. Þú skráir þig inn með Steam lotumiða eða með itch.io, með því að nota músarborðið og músina. Þjónustan sýnir einstakt skjánafn, 2 til 16 stafi. Leyfislykillinn og Steam auðkennið eru ekki sýnd öðrum spilurum.

Eftir þjónustuskilmálana er miðstöðin Uppáhalds, Vörulisti, Röðun, Leikmenn og prófílinn þinn.

Leyfi fyrir forritara er ritstjórinn. Þú býrð til leikinn, hleður honum upp og skilur eftir stjörnur. Leikjaspilaraleyfi er spilarinn. Hjörtu eru frá fólki sem spilar á hlaupabretti.

- **Vörulisti** — skoðaðu leiki. Síur ná yfir tegund (allt að þrjár), skjástærð, stærð leikjakassa og hvort upphleðslan inniheldur `.pix37` verkefni.
- **Uppáhalds** — leikir sem þú merktir.
- **Hlaða upp** — þarfnast forritaraleyfis. Þú sendir `.p37` skrá og mátt hengja `.pix37` verkefnið við. `.p37` skrá er pakkað með bætikóða. Hún er erfið að lesa og hægt er að rugla hana viljandi. `.pix37` skrá er verkefnaskráin sem þú vinnur í raun í: frumkóðinn, myndirnar og hljóðið, á formi sem annar aðili getur opnað og lært af. Með því að senda hana geta aðrir höfundar lært leikinn í stað þjappaðrar leikjahylkis. Titill, höfundur, tegund, forsíða, tákn og afrekaheiti fylgja því. Vinnsluminnistalan sem send er upp er hámarkið sem leikjahylkið náði.
- **Athugasemdir og afrek** — á leikjasíðunni. Að opna afrek á meðan þú ert skráð(ur) inn er vistað á þeim reikningi sem og á tölvunni.
- **Röðun** — forritarar, raðaðir eftir stjörnum eða hjörtum. Hver röð sýnir báðar tölurnar. Opnaðu nafn til að sjá leiki þess höfundar, raðað eftir hjörtum.
- **Leikmenn** — reikningar raðaðir eftir afrekum og opnaðir.

Kóði

Tungumálið er lítið Lua-stíls handrit. Þessi helmingur byrjar á tómu verkefni, kennir tungumálið og sýnir síðan stutt dæmi um hvert fall.

Fyrsta forritið

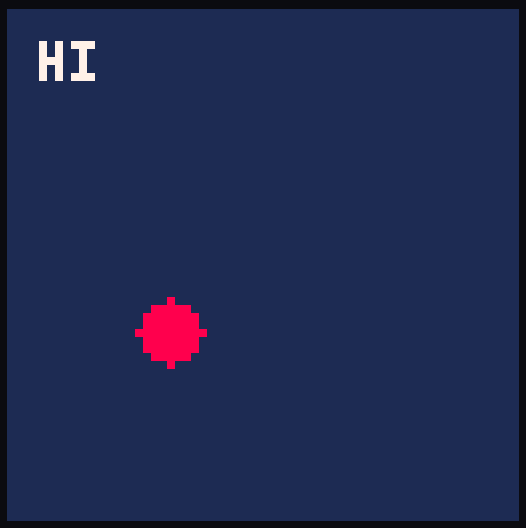
Búðu til verkefni í 64×64, 4-bita lit. Opnaðu CODE. Skiptu út flipanum fyrir:

```
_init()
  x = 20
  y = 40
end

_update()
  if btn(0) then x -= 1 end
  if btn(1) then x += 1 end
  if btn(2) then y -= 1 end
  if btn(3) then y += 1 end
end

_draw()
  cls(1)
  print("HI", 4, 4, 7)
  circfill(x, y, 4, 8)
end
```

Ctrl+R keyrir það. Örvatakkar færa hringinn. Escape stöðvar leikinn og fer aftur í ritilinn.



Sömu þrjú köll: `cls(1)` málar dökkbláan skjá, `print` teiknar HI og `circfill` teiknar bleikan hring.

Hvað gengur, og hvenær

- `_init` keyrir einu sinni.
- `_update` keyrir á 60 Hz. Settu hreyfingu og hnappa hér.
- `_draw` málar. Án fps færibreytu málar það eins hratt og tölvan og skjárinn geta farið. `_draw(30)` takmarkar málun við 30 fps og skilur rökfræðina eftir við 60.
- `_set_fps(n)` breytir þessu hámarki síðar. `0` hreinsar það.

Nöfn eru óháð há- og lágstöfum; ritilinn geymir þau með lágstöfum. `function` og `fun` eru sama leitarorðið.

Tungumál

Grunnatriði forritunar

Forrit er listi yfir línur sem leikjakassinn keyrir. Hver fyrirsögn sýnir fyrst lögun línunnar og síðan lítið dæmi. Lína sem byrjar á `--` er athugasemd: athugasemd fyrir þig. Leikurinn sleppir henni.

Breytur

Breyta er nafn sem inniheldur gildi. Þú velur nafnið. Síðari línur geta lesið það og þær geta geymt nýtt gildi með sama nafni.

Lögunin er `name = value` . `local name = value` heldur nafninu innan núverandi blokkar, þannig að restin af leikjatölvunni sér það ekki. Án `local` er nafnið alþjóðlegt og allir flipar geta séð það.

```
hp = 3            -- global: the whole cart can read and change hp
local coins = 0   -- local: only this block can see coins
hp = hp + 1       -- read 3, add 1, store 4 back into hp
```

`nil` þýðir að það er ekkert gildi. `true` og `false` eru já og nei. Nafn sem þú hefur aldrei stillt lesist sem `nil` .

```
bonus = nil            -- nothing stored yet
if bonus == nil then   -- condition: is bonus empty?
  bonus = 0            -- body: store a number
end
```

Töflur

Tafla inniheldur mörg gildi undir einu nafni. Listi notar tölur. Fyrsta atriðið er 1, ekki 0. Færsla notar nöfn. Tafla getur innihaldið aðra töflu.

Form lista: `{ value, value }` . Form færslu: `{ name = value, name = value }` . Listi yfir færslur er tafla sem inniheldur töflur.

```
-- a list. [1] is the first item
scores = {10, 20, 30}
print(scores[1], 1, 1, 7)      -- print draws text. this draws 10

-- a record. player.x is the field named x
player = {x = 10, y = 20}
print(player.x, 1, 8, 7)       -- draws 10

-- a list of records. enemies[1] is the first record
enemies = { {x = 4, y = 8}, {x = 20, y = 12} }
print(enemies[1].x, 1, 16, 7)  -- draws 4
print(enemies[2].y, 1, 24, 7)  -- draws 12
```

Ástand og líkami

Skilyrði er spurning með já eða nei svari, eins og `hp > 0` eða `btnd(4)` . Meginmál er hópur lína sem keyra þegar þú slærð inn í blokk. `if` og every lykkja notar sömu lögun: skilyrði, síðan meginmál og svo `end` . `repeat` lýkur með `until` í stað `end` .

Ef

`if` keyrir aðeins meginmál sitt þegar skilyrðið er satt. `elseif` reynir annað skilyrði. `else` keyrir þegar öll skilyrðin eru ósönn.

Lögun: `if condition then` , þá líkaminn, þá `end` .

```
hp = 2
if hp > 0 then                 -- condition: is hp above zero?
  print("ok", 1, 1, 7)         -- body: print draws the word ok
elseif hp == 0 then            -- another condition
  print("empty", 1, 1, 7)
else
  print("no", 1, 1, 7)         -- body when every condition was false
end

if hp > 0 print(hp, 1, 32, 7)  -- one call, no then/end. draws 2
```

Lykkjur

Lykkja endurtekur meginmál. `while` athugar skilyrðið fyrst og heldur áfram meðan það er satt. `repeat` keyrir meginmálið og athugar síðan, þannig að meginmálið keyrir alltaf að minnsta kosti einu sinni. „ `for` counts“. Síðasta talan er meðtalin. Án þriðju tölu er skrefið 1, þannig að ef upphafið er þegar komið yfir endi keyrir meginmálið ekki. Niðurtalning þarf neikvætt skref.

```
n = 0
while n < 3 do         -- condition: keep going while n is under 3
  n += 1
  print(n)             -- 1, then 2, then 3: how many times the body has run
end

n = 0
repeat
  n += 1
  print(n)             -- body first, then the condition. prints 1, then 2, then 3
until n > 2

for i = 1, 3 do        -- i is 1, then 2, then 3
  print(i)
end

for i = 0, 4, 2 do     -- start, end, step of 2
  print(i)             -- 0, then 2, then 4
end

for i = 3, 1, -1 do    -- counting down needs the step
  print(i)             -- 3, then 2, then 1
end

for i = 1, 5 do
  if i == 3 then break end  -- leave the loop when i reaches 3
  print(i)                  -- 1, then 2
end
```

Að ganga um borð

`all` heimsækir hvert atriði á lista, frá fyrsta. `pairs` heimsækir hvert nafn og gildið sem er geymt undir því nafni.

```
t = {1, 2, 3}                -- list: t[1] is 1, t[2] is 2, t[3] is 3
for v in all(t) do           -- v is 1, then 2, then 3
  print(v)                   -- print draws that number
end

player = {x = 10, y = 20}    -- record: a field x and a field y
for k, v in pairs(player) do
  print(k)                   -- k is the name, "x" then "y"
end
```

Aðgerðir

Fall er nefnd uppskrift. Lögunin er `function name(arguments)` , síðan `meðaltal` og svo `end` . Þú kallar það með því að skrifa `heiti` og sviga. `return` sendir gildi til baka og getur sent fleiri en eitt. `parameters` eru staðbundnar gildir fyrir það fall.

```
function greet(who)       -- who is a local that holds what you pass in
  return "hi "..who       -- .. glues two texts. return sends the result back
end

print(greet("ada"))       -- draws hi ada
```

```
function pair()
  return 1, 2             -- two values
end

a, b = pair()             -- a becomes 1, b becomes 2
```

`local function` er fall sem getur séð staðbundna gildi í kringum sig. Það er lokun. Tafla getur innihaldið eitt slíkt fall og þú kallar á það með punkti.

```
function make()
  local n = 0
  local m = {}
  m.bump = function()     -- this function sees n
    n += 1
    return n
  end
  return m
end

counter = make()
print(counter.bump())     -- 1
print(counter.bump())     -- 2
```

Já, nei, og að velja

snýr `not` við skilyrði. `and` er aðeins satt þegar báðar hliðar eru sannar. `or` er satt þegar hvor hlið sem er er sönn. `a and b or c` velja `b` þegar `a` er satt, annars `c` . `cond ? a : b` er sama hugmyndin í einni línu: `a` þegar skilyrðið er satt, `b` þegar það er það ekki.

```
hp = 2
hit = false
night = false

if hp > 0 and not hit then         -- both parts are true, so the body runs
  print("safe", 1, 1, 7)           -- draws safe
end

label = hp > 0 and "ok" or "out"   -- hp is above 0, so label is ok
print(label, 1, 8, 7)

shade = night ? 1 : 7              -- night is false, so shade is 7
print(shade, 1, 16, 7)
```

Stærðfræði og texti

`+ - * /` eru samlagning, frádráttur, margföldun og deiling. `//` deilir og heldur heilu tölunni. `%` er afgangurinn. `^` er veldi. `..` límir saman tvo texta. `#t` er fjöldi liða í lista. Samanburður er `==` (sama), `~=` eða `!=` (mismunandi) og `< > <= >=` . `x += 1` leggur 1 við x. Sama lögun virkar fyrir `-= *= /= ^= %=` .

```
scores = {10, 20, 30}    -- the list named scores
tiles = 15 // 8          -- 1, the fraction is dropped
left = 15 % 8            -- 7, what is left over
title = "hi ".."ada"     -- the text hi ada
n = #scores              -- how many items are in the list named scores
print(n, 1, 1, 7)        -- 3
print(title, 1, 8, 7)

score = 0
best = 10
if score ~= best then    -- condition: score is not the best
  score += 10            -- add 10 to score
end
print(score, 1, 16, 7)   -- 10
```

Þar sem heimamaður býr

`local` gildir þar til `end` þess blokkar sem lokar blokkinni. `if` / `end` er eigin blokk. Þú getur nefnt `local` fyrir ofan línuna sem setur hana, en þangað til sú lína keyrir er gildið `nil` .

```
if true then
  print(coins)       -- nil: the next line has not run yet
  local coins = 5
  print(coins)       -- 5
end
-- coins does not exist out here
```

Stökk

`::label::` markar línu. `goto label` fer þangað, en aðeins innan sömu fallsins.

```
x = 0
::again::
x += 1                      -- add 1
if x < 3 then goto again end  -- condition: jump back while x is under 3
```

Minni, í stuttu máli

`peek` and `poke` skoða lifandi pixla í gegnum eitt 16-bita vistfang. Þeir geyma ekki annað eintak af leikjatölvunni. Vistföngin `0x5F00` til `0x5FFF` eru skrár: myndavél, myndband (aðeins til lestrar), teiknulitur, fyllingarmynstur, þrjár litatöflubreytingar og síðunúmer. Bætin hvoru megin við gatið, 65280 talsins, eru pixlaglugginn.

Þegar skjárinn passar í þessi 65280 bæti, 64×64 og 128×128 gera það en 256×256 ekki, byrjar skjárinn á vistfangi 0, röð fyrir röð. Það sem eftir er er sprite-blaðið sem `0x5F41` valdi (0 er blað 0). Ef það blað er stærra en plássið sem eftir er, velur `0x5F42` síðuna og kortið er ekki í glugganum. Ef blaðið passar, þá eru halinn kortreitir úr kortinu sem `0x5F43` valdi, einn eða tveir bæti hver, og `0x5F44` blaðsíða kortsins þegar það passar ekki í halann.

Þegar skjárinn passar ekki er allur glugginn ein síða með 65280 pixlum. `poke(0x5F40, n)` skiptir um síðu á skjánum. `poke(0x5F41, n)` með n öðru en 0 sýnir blað n í staðinn, og blað 0 hefur enga sýn í þessum ham. `poke(0x5F43, n)` með n öðru en 0 sýnir kort n, og það vinnur yfir blaðinu. Kort 0 í þessum ham er aðeins `mget` og `mset` .

`peek` sem lendir ekki á pixli eða skrá skilar 0. `poke` þar gerir ekkert. Sprite fánar, SFX og DATA töflur eru ekki í þessum glugga. Notið `fget` , `sfxget` og `dget` .

Hagnýt notkun

Hvert fall er talið upp neðst á þessari síðu, ásamt hagnýtri notkun PIX37 falla. Tenglarnir undir hverju dæmi opna frekari upplýsingar um fallin.

Mjúk hreyfing

128×128 spilakassa, séð ofan frá. Sprite 1 er þegar teiknaður. Örvar hraða sér í allar áttir og núningur hægir á þér. Sprite vefur sig: helmingur sem fer frá botninum er teiknaður efst og það sama á við um hinar brúnirnar.

`_init` `_update` `_draw` `btn` `mid` `cls` `spr` `palt`

Athuga hvort næsta skref sé lokað

64×64 leikjahylki. Gólfið og hægri veggurinn eru þegar á kortinu, málaðir með sprite 3, og fáni 0 er stilltur á þann sprite. Örvar færa sprite 1. Orðið efst segir læst þegar næsta skref myndi skarast við samfellda flís, og frjálst þegar það myndi ekki gera það.

`flr` `fget` `mget` `btn` `cls` `map` `spr` `print` `_update` `_draw`

Þurrkur með smellu

Ýttu á A. Þurrkunarferlið er teiknað með flip(), ein svört röð í einu, í stað þess að vera inni í _draw(). _draw er ekki í vegi á meðan það keyrir, þannig að hver röð helst á skjánum.

`rectfill` `flip` `btnd` `cls` `print` `_update` `_draw`

Tafla yfir föll (flokkur)

Sprite 1 er þegar teiknaður. make_monster virkar eins og klasi. Hann skilar töflu og update og draw eru aðferðir sem halda x og y gildi skrímslisins. Talan efst sýnir hversu langt það hefur gengið.

`spr` `cls` `print` `flr` `_init` `_update` `_draw`

Fjórar leiðir til að lesa hnapp

A er hnappur 4. Haltu honum inni og hleðslan hækkar. Slepptu honum og hleðslan fer aftur í 0. Upp og niður endurtaka sig meðan haldið er inni, þannig að valmyndaröðin heldur áfram að færast. btnd er fyrsti ramminn í ýtingu, sem þessi leikjahylki notar aðeins sem upphaf þess innihalds.

`btn` `btnd` `btnp` `btnu` `cls` `print` `_update` `_draw`

Skjóta með músinni

Sprite 1 er þegar teiknaður í miðjunni. Hvítur punktur er bendillinn. Vinstri smellur skýtur pixli í átt að honum. Hjólið breytir hraðanum, prentað í horninu.

`scroll` `clickl` `mouse` `sqrt` `add` `all` `cls` `spr` `pset` `print` `_update` `_draw`

Sláðu inn nafn og staðfestu það

Ýttu á A til að opna reitinn. Upp og niður breytir blikkandi staf. A skrifar hann, X eyðir, Y er bil og B staðfestir. Línan segir síðan nafnið þitt.

```
btnd keyon keystr flr time cls print _update _draw
```



Myndavél á spilaranum

Leikjakassa með 128×128 að stærð. Sprite 1 byrjar í miðju flísagólfi. Örvar hreyfast. Myndavélin heldur sprite-inu miðjuðu. x og y í horninu fletta ekki, því camera() án viðfanga færir sýn aftur fyrir HUD.

```
btn screen camera cls map spr print flr _update _draw
```



Nokkrar línur úr sjónrænni skáldsögu

Línurnar eru í DATA, á ensku og pólsku. A heldur áfram eða velur að vera á meðan valmöguleiki er á skjánum. B velur að fara á valmöguleika og skiptir annars um tungumál. X skiptir um tungumál á meðan valmöguleiki er birtur. Samræðurnar nota 8px leturgerðina svo pólskir stafir teiknast.

```
dstr dget btnd cls print8 print _update _draw
```



Mála með kík og poka

64×64 leikjatölvu, þannig að vistfang 0 er pixlinn efst til vinstri og næsta röð byrjar á 64. Leikjatölvunni er raðað pixlum, afritað röð, hreinsað röð og prentað gildin sem hún las til baka.

```
cls poke peek poke2 peek2 poke4 peek4 memcpy memset print _draw
```



Listi sem þú getur bætt við

A leggur saman tölu, allt að 8. B fjarlægir síðustu töluna. X fjarlægir fyrstu töluna. Listinn prentast niður skjáinn.

```
btnd count add deli cls ipairs print pack unpack next foreach select pcall assert del _update _draw
```



Nafn tekið í sundur

Spilakassinn klýfur ada,12,blue og prentar nafnið, fyrsta staf þess, 12, og orðið blue.

```
cls split sub chr ord tonum tostr print _draw
```



Spilaðu nótu og breyttu henni síðan

Mynstur 0 er ein nóta að lengd og hver endurtekning er með sama hljóðstyrk. A spilar það aftur þegar nótan er búin. B hækkar tónhæðina og spilar þá nótu.

`sfxset sfxhdr sfx btnd sfxstat sfxget cls print _init _update _draw`

 Lag skrifað í leikjakassann

Tvö mynstur lykkjast í röð 0, eitt á hvorri af fyrstu tveimur rásunum. Röðin helst 0. ch0 og ch1 eru nóturnar sem hver rás er á. Að breyta mynstrinu, LEN og LOOP í leikjatölvunni er leið til að sjá hvað þessi stjórntæki gera í tónlistarspjaldinu.

`songget songset music musicstat cls print _init _draw`

 Þrjú skrímsli í DATA

GÖGNADALKASSIÐ hefur nú þegar nafnadálk og líffæradálk, með slími, leðurblaka og auga. A særir slímið. Tölurnar hægra megin eru líffæradálkurinn.

`drows dstr dget add btnd dset dcols cls print all _init _update _draw`

 Mundu eftir stigunum

A bætir við stigi og vistar það. Hætta og opna spilkassann aftur: númerið er enn til staðar. B gleymir því.

`load save btnd delsave cls print _init _update _draw`

  Annar kafli í annarri skrá

Tvær leikjahylki deila vistunarverkefninu. 1. hluti: A markar að leiknum sé lokið. 2. hluti, sem opnast fyrst, segir að klára eigi 1. hluta. Eftir það opnast 2. hluti og B eyðir vistuninni svo hægt sé að spila 1. hluta aftur.

`load saveslot btnd save delsave cls print _init _update _draw`

 Eitt afrek

Sprite 8 er þegar dregið. Ýttu tíu sinnum á A. Ristað brauð nefnir afrekið og Sprite 8 helst þar sem það er ólæst. B hreinsar afrekið í staðbundinni skrá og endurstillir teljarann.

`achdef btnd achieved achieve achclear cls print spr _init _update _draw`

 Herbergi sem þú getur gengið inn í

256×256 spilakassa. Hvítur punktur er bendillinn. Valmyndin er gestgjafi, sameinast eða listi yfir netþjóna. Gestgjafinn opnar herbergi sem heitir ganga og byrjar leikinn. Sameinast skrollar um vistfang (tölustafi, punkt, tvípunkt) og B tengist. Listi yfir netþjóna færist upp og niður og A sameinast röðinni. Í leiknum ganga örvar með sprite 1. Einum peningi er deilt: gestgjafinn færir hann og að taka hann upp bætir 1 við stig þess spilara og býr til næsta. Sérhver sprite, þar á meðal þinn, sýnir það stig við hliðina á sér. Hornið sýnir hversu margir

spilarar eru inni, staðarnetfangið og opinbert vistfang gestgjafans. B fer: þú og hinir farið aftur í valmyndina og næsti gestgjafi byrjar nýtt herbergi með stig 0.

`btn` `btnd` `mouse` `pset` `spr` `print` `flr` `rnd` `pairs` `nlisten` `nhost` `nannounce` `nlist` `njoin` `nleave` `nsend` `nkb` `nkbstr` `whoami` `nplayers` `nlan` `_mp` `_mp_join` `_mp_leave` `_mp_fail` `palt` `cls` `time` `_init` `_update` `_draw`

 Samræður sem bíða

A sýnir næstu línu. Fallið gerir hlé við hverja keyrslu þar til þú ýtir aftur á A.

`yield` `cocreate` `btnd` `coresume` `cls` `print` `_init` `_update` `_draw`

 Bíddu í hálfa sekúndu

Orðið breytist úr ... í hæ eftir um það bil hálfa sekúndu. Skjárinn heldur áfram að teikna á meðan kórótínan er í biðstöðu.

`yield` `cocreate` `costatus` `coresume` `cls` `print` `_init` `_update` `_draw`

 Hringir sem koma einn af öðrum

Fimm hringir drjúpa inn að ofan, með tveggja ramma millibili, og falla síðan.

`add` `yield` `cocreate` `coresume` `all` `cls` `circfill` `_init` `_update` `_draw`

 Tveir rammar, úr klukkunni

Sprite 1 og sprite 2 eru tveir rammar af sama persónunni. Myndin skiptir um mynd fjórum sinnum á sekúndu.

`cls` `flr` `time` `spr` `_draw`

Sérhver aðgerð

Færibreytur í hornklofum eru valfrjálsar. Vísitölur taflna og strengja byrja á 1, nema DATA raðir og dálkar, sem byrja á 0. Litir eru litatöfluvísitölur.

Vísitala

`_teikna`
`_init`
`_mp`
`_mp_fail`
`_mp_join`
`_mp_leyfi`
`_mp_uppfærsla`
`_setja_fps`
`_uppfærsla`
`abs`
`achclear`
`achdef`
`ná`
`náð`
`bæta við`
`allt`
`fullyrða`
`atan2`

hljómsveit
bnot
bor
btn
btnd
btnp
btnu
bxor
myndavél
loft
kr
hring
hringfylling
klikkl
smellur
smellur
bút
cls
samskapa
lit
corsume
vegna
samstaða
telja
bendilinn
dcols
del
sælkeraverslun
delsave
dget
drukknar
dset
dstr
fget
fylla
fletta
flr
framhjá
fset
upplýsingar
ipairs
slökkt á lykla
takka
keystat
keystr
línu
hlaða
log
lshr
kort
hámark
memcpy
memset
mget
miðjan
mín
mús
mset
tónlist
tónlistartölfræði
tilkynna
ncall
næst
nhost
njoin
nkb
nkbstr
nlan
nleif
nlist
nheyrðu
ath
nleikmenn
nsend
nstun
orð
sporöskjulaga
ovalfill
pakka
pör
vinur
palrgb
palt
pcall
kíkja
kíkja 2
kíkja 4
perf
pget
pota
stinga 2
stinga 4
prenta
prenta 10
prenta 8
pset
rétt
endurfylla
rnd
rotl
rotr
rétt
rréttfylla
vista
saveslot
skjár
fletta
velja
settkort
settpallettu
setspr
settex
uppgjör
sfx
sfxget
sfxhdr
sfxset
sfxstat
sget
sgn
shl
sýningartölur
shr

synd
songget
lagasett
skipt
spr
sqrt
srand
sett
sspr
tölfræði
undir
tíma
tlína
tonum
tostr
pakka niður
whoami
ávöxtun

Lífsferill

_init

`_init() ... end`

Keyrir einu sinni þegar leikjakassann ræsist. Kóði sem er utan við hvaða krók sem er er einnig settur inn í _init.

```
_init()
  x = 64
  y = 40
end
```

_update

`_update() ... end`

Leikjarökfræði. Kölluð á 60 Hz, jafnvel þegar _draw er takmarkað eða gestgjafinn er hægari. Lestu hnappa hér.

```
_update()
  if btn(0) then x -= 1 end
  if btn(1) then x += 1 end
end
```

_draw

```
_draw([fps]) ... end
```

Teiknar einn ramma. Án breyta, eða 0, keyrir það eins oft og gestgjafinn kynnir. Sendið n (1..999) til að takmarka teikninguna við n ramma á sekúndu. _update helst á 60 Hz.

```
_draw(30)
  cls(1)
  spr(1, x, y)
end
```

_set_fps

```
_set_fps(n)
```

Breytir _draw hámarkinu á meðan leikurinn er í gangi. 0 þýðir ótakmarkað. n er stillt á 0..999.

```
if btn(4) then _set_fps(60) end
```

flip

```
flip()
```

Sýnir skjáinn og víkur fyrir næsta ramma. Notið hann aðeins ef þið eruð að teikna ykkar eigin rammalykkju. Ekki bíða eftir að hann sé upptekinn.

```
cls(0)
print("tick", 4, 4, 7)
flip()
```

time

```
time()
```

Sekúndur síðan _init.

```
print(time(), 4, 4, 7)
```

Inntak

btn

`btn([b], [p])`

1 á meðan hnappur b er haldið inni, annars 0. p er spilari, 0..7. Án viðfanga skilar það bitagrímu af spilara 0. Hnappar: 0 vinstri, 1 hægri, 2 upp, 3 niður, 4 A, 5 B, 6 X, 7 Y, 8 R, 9 L, 10 velja, 11 byrja.

```
if btn(4) then shoot() end
```

btnd

`btnd(b, [p])`

Aðeins satt á _update rammanum þegar hnappur b fer niður.

```
if btnd(5) then menu_open = not menu_open end
```

btnp

`btnp([b], [p])`

Satt þegar ýtt er á, og svo aftur þegar haldið er inni: 30 tikk af seinkun, síðan á 8 tikk fresti. Ef enginn hnappur er notaður skilar það grímu.

```
if btnp(2) then cursor -= 1 end
```

btnu

`btnu(b, [p])`

Aðeins satt á rammanum þegar hnappurinn er sleppt.

```
if btnu(4) then charge = 0 end
```

mouse

`mouse()`

Skilar músar x, y í pixlum leiksins.

```
mx, my = mouse()
pset(mx, my, 7)
```

clickl

`clickl([p])`

Vinstri músarhnappur. clickl() er haldið niðri. clickl(1) er þrýstikanturinn.

```
if clickl(1) then place(mouse()) end
```

clickr

`clickr([p])`

Hægri músarhnappur. Sama 1-argument brún og smellt er.

```
if clickr() then cancel() end
```

clickm

`clickm([p])`

Miðhnappurinn eða smellur með skrunhjólinu.

```
if clickm(1) then drag = true end
```

scroll

`scroll()`

Hreyfing músarhjólsins í þessum ramma. Jákvætt og neikvætt fer eftir stefnu hjólsins.

```
cam_y -= scroll()
```

keyon

`keyon()`

Opnar textareit sem er knúinn áfram af hnöppum. Upp og niður breytir stafnum A–Ö. A skrifar þann staf. X eyðir. Y er bil. B staðfestir og lokar reitnum. Lyklaborðið skrifar ekki í hann.

```
if btnd(4) then keyon() end
```

keyoff

`keyoff()`

Felur textareitinn.

```
keyoff()
```

keystr

`keystr([s])`

Texti í reitnum, síðan 1 á rammanum B staðfestir, og svo breytast stafirnir upp og niður. Sá stafur er tómur þegar reiturinn lokast. keystr("") hreinsar textann.

```
name, done, letter = keystr()
if done == 1 then start(name) end
```

keystat

`keystat()`

1 á meðan textasvæðið er opið, annars 0.

```
if keystat() == 1 then return end
```

Teikning

screen

`screen()`

Skjástærð núverandi leikjatölvu í pixlum. Skilar breidd og síðan hæð. Þetta er upplausn leiksins, ekki ritilglugginn.

```
w, h = screen()
```

cls

`cls([col])`

Hreinsar skjáinn. Sjálfgefinn litur er 0.

```
cls(1)
```

color

`color(col)`

Stillir teiknulitinn sem notaður er þegar síðari kall sleppir lit. Án viðfangs skilar það núverandi lit.

```
color(7)
print("hi")
```

camera

`camera([x, y])`

Skiptir frá hverju teiknakalli. camera() núllstillist á 0, 0.

```
camera(cam_x, cam_y)
```

clip

`clip([x, y, w, h, clip])`

Takmarkar teikningu við rétthyrning. Fimmta breytan, sem er ekki núll, skerst við núverandi myndskeið. clip() endurheimtir allan skjáinn. Sjónvarp myndskeiðsins er eingöngu til lestrar.

```
clip(8, 8, 48, 48)
circfill(32, 32, 20, 8)
clip()
```

cursor

`cursor([x, y, col])`

Setur hvert næsta prentun án hnita fer, og valfrjálst skilar color. cursor() x, y.

```
cursor(4, 4, 7)
print("line 1")
print("line 2")
```

pset

`pset(x, y, [col])`

Setur einn skjápixel. Telst með í pixeltölfræðinni.

```
pset(x, y, 10)
```

pget

`pget(x, y)`

Litur skjápixla við x og y.

```
if pget(x, y+8) == 3 then grounded = true end
```

print

`print(v, [x, y, col])`

Teiknar 3×5 texta (UTF-8, klipptan úr Fusion leturgerðinni). Án x, y prentast það við bendilinn og færist í næstu línu.

```
print("score "..score, 2, 2, 7)
```

log

`log(...)`

Bætir viðfangsefnunum, aðskildum með bilum, við textabiðminnið. pix37_run prentar það biðminnið við stdout. Það teiknar ekki.

```
log("hp", hp)
```

print8

`print8(v, [x, y, col])`

Sama og prentun, með 8 px Fusion leturgerðinni.

```
print8("pause", 16, 28, 7)
```

print10

`print10(v, [x, y, col])`

Sama og prentun, með 10 px Fusion leturgerðinni.

```
print10("pix37", 8, 8, 10)
```

line

`line([x0, y0, x1, y1, col])`

Lína frá x0,y0 til x1,y1. Tvær breytur halda áfram frá fyrri enda. line() endurstillir þá keðju.

```
line(0, 32, 63, 32, 6)
```

circ

`circ(x, y, r, [col])`

Útlínur hrings. Radíus er í pixlum.

```
circ(32, 32, 12, 12)
```

circfill

`circfill(x, y, r, [col])`

Fylltur hringur.

```
circfill(x, y, 3, 8)
```

oval

`oval(x0, y0, x1, y1, [col])`

Sporbaugsútlínur innan afmörkunarkassans. Horn eru innifalin.

```
oval(8, 20, 56, 44, 7)
```

ovalfill

`ovalfill(x0, y0, x1, y1, [col])`

Fylltur sporbaugur í afmörkunarkassa.

```
ovalfill(10, 40, 54, 60, 3)
```

rect

`rect(x0, y0, x1, y1, [col])`

Útlínur rétthyrnings. Horn eru innifalin.

```
rect(0, 0, 63, 63, 7)
```

rectfill

`rectfill(x0, y0, x1, y1, [col])`

Fylltur rétthyrningur.

```
rectfill(0, 56, 63, 63, 3)
```

rrect

`rrect(x, y, w, h, [r, col])`

Útlínur rétthyrnings með ávölum línum. x og y eru efst til vinstri. w og h eru stærðin í pixlum; bæði verða að vera stærri en 0. r er radíus hornsins og sjálfgefið gildi er 0. Það er klemmt við helming styttri hliðarinnar.

```
rrect(8, 8, 48, 16, 3, 7)
```

rrectfill

`rrectfill(x, y, w, h, [r, col])`

Fylltur, ávöl rétthyrningur. Sömu viðfangsefni og rrect.

```
rrectfill(8, 40, 48, 16, 2, 1)
```

fillp

`fillp([p])`

16-bita fyllingarmynstur fyrir síðar fylltar form. 0 er samfellt. fillp() skilar núverandi mynstri. Bitar ganga 4×4 mynstrið.

```
fillp(0xA5A5)
rectfill(8, 8, 40, 40, 10)
fillp(0)
```

pal

`pal([c0, c1, [mode]])`

Endurvarpar lit c0 í c1. stilling 0 er teiknipallettan, 1 birtingarpallettan, 2 fyllingarmynsturpallettan. pal() endurstillir.

```
pal(8, 10)
circfill(32, 32, 8)
```

palrgb

`palrgb(col, r, g, b)`

Setur eina litatöflufærslu á RGB, hver rás 0..255.

```
palrgb(8, 255, 80, 40)
```

palt

`palt([col, t])`

Gagnsæi. palt(col) eða palt(col, true) gerir col gegnsæjan. palt(col, false) gerir hann samfelldan. palt() gerir alla liti ógegnsæja.

```
palt(0)
spr(1, x, y)
```

setpalette

`setpalette(n)`

Velur litatöflubanka n. setpalette() skilar núverandi banka.

```
setpalette(1)
```

tline

`tline(x0, y0, x1, y1, mx, my, [mdx, mdy])`

Teiknar línu á skjáinn á meðan sýni eru tekin af kortapixlum frá kortastöðu mx, my, mdx og mdy. Færir kortasýnið skref fyrir skref eftir línunni.

```
tline(0, 40, 63, 20, 0, 0, 0.5, 0)
```

Sprítur og kort

spr

`spr(n, x, y, [w, h, fx, fy, tex, rot])`

Teiknar sprite n við x, y. w og h eru stærðirnar í pixlum, sjálfgefið gildi 8. fx og fy snúast lárétt og lóðrétt þegar þau eru ekki núll. tex er flipi sprite-blaðsins. rot er hornið í beygjum (0..1), sömu átt og sin og cos. 0 skilur sprite eftir ósnúið.

```
spr(1, x, y, 8, 8, 0, 0, 0, aim)
```

sspr

`sspr(sx, sy, sw, sh, dx, dy, [dw, dh, fx, fy, tex])`

Afritar rétthyrning af sprite-blaðinu á skjáinn, minnkað í dw um dh. Sjálfgefnar stillingar halda upprunastærðinni.

```
sspr(0, 0, 8, 8, 16, 16, 32, 32)
```

sget

`sget(x, y, [tex])`

Litur eins texels á sprite-blaðinu.

```
c = sget(0, 0)
```

sset

`sset(x, y, [col, tex])`

Skrifar einn texel á sprite-blaðið.

```
sset(3, 3, 8)
```

fget

`fget(n, [f, tex])`

Sprite flags. fget(n) skilar fánabæti. fget(n, f) er satt þegar fánabitinn f (0..7) er stilltur.

```
if fget(tile, 0) then solid = true end
```

fset

`fset(n, [f], v, [tex])`

Setur sprite fána. fset(n, v) skrifar allan bætinn. fset(n, f, v) setur einn bita.

```
fset(2, 0, true)
```

setspr

`setspr(n)`

Velur virka sprite-töfluna. setspr() skilar núverandi töflu. Sami banki og settex.

```
setspr(1)
```

settex

`settex(n)`

Gælunafn setspr.

```
settex(0)
```

map

`map(cx, cy, sx, sy, [cw, ch, layer, tex])`

Teiknar kortið. Reitur cx, cy á kortinu er teiknaður á skjá sx, sy. cw, ch af 0 þýðir að allt kortið er notað. Lagið er samþykkt og hunsað. tex er flísasettarblaðið.

```
map(0, 0, 0, 0, 16, 16)
```

mget

`mget(cx, cy)`

Sprite-auðkenni í kortareit cx, cy.

```
if fget(mget(tx, ty), 0) then bounce() end
```

mset

`mset(cx, cy, n)`

Skrifar sprite n í kortafrumu cx, cy.

```
mset(3, 4, 0)
```

setmap

`setmap(n)`

Velur kort n. setmap() skilar núverandi korti. Kort 0 er raunverulegt kort.

```
setmap(1)
```

Minni

peek

`peek(addr)`

Les einn bæti af 64 KB glugganum. Glugginn er lifandi skjárinn, síðan sprite pixlar, og svo vörpun frumna þegar skjárinn passar í 65280 bæti. Stærri skjáir eru blaðsíðaðir með poke(0x5F40, page). Vistföng sem eru ekki vörpuð skila 0. Einnig skrifað @addr.

```
px = peek(0)
```

poke

`poke(addr, ...)`

Skrifar bæti á addr. Aukabreytur skrifa eftirfarandi bæti. Ókortlögð vistföng eru hunsuð. Einnig skrifað $addr, val. Að stinga í skjáinn telst sem pixlaskrif.

```
poke(0x5F00, 4) -- camera x
```

peek2

`peek2(addr)`

Óundirritað 16-bita gildi, little-endian, á vistfangi.

```
w = peek2(0x6000)
```

poke2

`poke2(addr, val)`

Skrifar óundirritað 16-bita gildi, little-endian.

```
poke2(0x6000, 1000)
```

peek4

`peek4(addr)`

Óundirritað 32-bita gildi, little-endian.

```
n = peek4(0x6000)
```

poke4

`poke4(addr, val)`

Skrifar óundirritað 32-bita gildi, little-endian.

```
poke4(0x6000, 1)
```

memcpy

`memcpy(dst, src, len)`

Afritar len bæti innan sama 64 KB gluggans.

```
memcpy(0x6000, 0x6100, 64)
```

memset

`memset(dest, val, len)`

Fyllir len bæti frá og með dest með val.

```
memset(0, 0, 64)
```

Stærðfræði

abs

`abs(x)`

Algjört gildi.

```
d = abs(tx - x)
```

flr

`flr(x)`

Rúndar í átt að mínus óendanleika.

```
tx = flr(x / 8)
```

ceil

`ceil(x)`

Rúndar í átt að plús óendanleika.

```
n = ceil(hp / 8)
```

max

`max(a, b)`

Stærra gildið af tveimur.

```
x = max(0, x)
```

min

`min(a, b)`

Lægra gildið af tveimur.

```
x = min(63, x)
```

mid

`mid(a, b, c)`

Miðgildið af þremur. Gagnlegt sem klemma.

```
x = mid(0, x, 63)
```

sgn

`sgn(x)`

Merki x: -1, 0 eða 1.

```
x += sgn(dir)
```

sqrt

`sqrt(x)`

Kvaðratrót. Neikvætt færibreyti er ekki tala.

```
dist = sqrt(dx*dx + dy*dy)
```

sin

`sin(x)`

Sínus. x er í beygjum: 0, 0,25, 0,5, 0,75 fara einu sinni í hringinn. Ekki radíanar.

```
y = 32 + sin(time()) * 8
```

cos

`cos(x)`

Kósínus, aftur á móti, sama venja og sinus.

```
x = 32 + cos(time()) * 8
```

atan2

`atan2(dx, dy)`

Horn vigursins dx, dy, í beygjum frá 0 til 1.

```
aim = atan2(mx - x, my - y)
```

rnd

```
rnd([x])
```

Handahófskennd tala í [0, x). Sjálfgefið gildi x er 1, þannig að rnd() er 0..1. Fyrir tening, flr(rnd(6))+1.

```
c = flr(rnd(16))
```

srand

```
srand(seed)
```

Slembigeneratorinn er settur í fræ svo hægt sé að endurtaka síðari umferðaröð.

```
srand(1)
```

band

```
band(a, b)
```

Bitavís OG. Einnig skrifað a og b.

```
if band(flags, 1) ~= 0 then hit() end
```

bor

```
bor(a, b)
```

Bitavís EÐA. Einnig skrifað a | b.

```
flags = bor(flags, 4)
```

bxor

```
bxor(a, b)
```

Bitavís XOR. Einnig skrifað a ^^ b.

```
n = bxor(n, 255)
```

bnot

`bnot(a)`

Bitavís EKKI. Einnig skrifað ~a.

```
mask = bnot(1)
```

shl

`shl(a, n)`

Færa til vinstri. Einnig skrifað sem << n.

shr

`shr(a, n)`

Reiknifræðileg færsla til hægri. Einnig skrifað sem >> n.

```
n = shr(n, 1)
```

lshr

`lshr(a, n)`

Rökrétt færsla til hægri. Einnig skrifað sem >>> n.

```
n = lshr(n, 4)
```

rotl

`rotl(a, n)`

Snúa bitum til vinstri. Einnig skrifað sem <<> n.

```
n = rotl(n, 1)
```

rotr

`rotr(a, n)`

Snúa bitum til hægri. Einnig skrifað sem >>< n.

Töflur og gildi

add

`add(t, v, [i])`

Bætir v við töflu t. Ef i er gefið, setur það inn við þá 1-byggðu vísitölu.

```
add(bullets, {x=x, y=y})
```

del

`del(t, v)`

Fjarlægir fyrstu færsluna sem er jöfn v.

```
del(items, gem)
```

deli

`deli(t, [i])`

Fjarlægir vísitöluna i og skilar því gildi. Ef i er ekki til staðar fjarlægir síðasta færslan.

```
local b = deli(bullets, i)
```

all

`all(t)`

Ítrekari yfir fylkishluta, í réttri röð. Göt stöðva gönguna í þeirri átt sem 1-byggð fylki endar.

```
for b in all(bullets) do
  b.x += 2
end
```

foreach

`foreach(t, fn)`

Kallar á fn(v) fyrir hvert fylkisstak.

```
foreach(enemies, draw_enemy)
```

pairs

`pairs(t)`

Ítrari yfir alla lykla og gildi, þar á meðal nefnda reiti.

```
for k, v in pairs(player) do
  print(k)
end
```

ipairs

`ipairs(t)`

Ítrerari yfir fylkishlutanum með 1-byggðri vísitölu.

```
for i, v in ipairs(deck) do
  spr(v, i*8, 8)
end
```

next

`next(t, [k])`

Skilar næsta lykli og gildi á eftir k. next(t) byrjar í upphafi. Notað af pörum.

```
k, v = next(t)
```

count

`count(t, [v])`

Lengd fylkishlutans. Ef v er gefið, telur það hversu margar færslur eru jafngildar v.

```
if count(gems) == 0 then win() end
```

pack

`pack(...)`

Pakkar færibreytunum í nýja töflu.

```
t = pack(1, 2, 3)
```

unpack

`unpack(t, [i, j])`

Skilar t[i] til t[j] sem aðskildum gildum. Sjálfgefnar gildi ná yfir allan fylkishluta.

```
a, b = unpack(t)
```

select

`select(i, ...)`

select("#", ...) skilar fjölda viðfanga sem fylgja í kjölfarið. select(i, ...) skilar viðfangsefnunum frá og með i.

```
n = select("#", 1, 2, 3)
```

assert

`assert(c, [msg])`

Stöðvar með villu þegar c er ósatt eða núll. msg kemur í stað sjálfgefinna texta.

```
assert(hp > 0, "dead")
```

pcall

`pcall(f, ...)`

Kallar á f. Við árangur skilar `sann` og niðurstöðunum. Við villu skilar `ósatt` og villutextanum.

```
ok, err = pcall(load_level, 3)
```

Strengir

chr

`chr(n)`

Einn stafur úr bætikóða 0..255.

```
s = chr(65) -- A
```

ord

`ord(s, [i, j])`

Bætikóðar fyrir s frá vísitölu i til j. Vísitölur eru byggðar á 1. Sjálfgefið er að nota fyrsti stafurinn.

```
code = ord("a")
```

sub

`sub(s, i, [j])`

Hlutstrengur frá i til j, 1-byggður. Slepptu j til að taka halann.

```
name = sub(keystr(), 1, 8)
```

split

`split(s, [sep, conv])`

Skiptir s í töflu. Sjálfgefið er að `sep` sé `komma`. Talan `sep` er breidd bútsins. `conv` breytir `bitum` með `tonum` þegar það er `true`.

```
words = split("red,blue", ",")
```

tonum

`tonum(s, [base])`

Greinir tölu. Grunntalan getur verið 2..36. Skilar núlli þegar textinn er ekki tala.

```
n = tonum("255")
```

tostr

`tostr(v, [digits])`

Breytir gildi í texta. digits setur aukastafi fyrir tölur.

```
print(tostr(hp, 0), 4, 4, 7)
```

Hljóð

sfx

`sfx(n, [ch, offset, len])`

Spilar mynstur n og skilar rásinni. Sleppið ch, eða sleppið -1, og fyrsta lausa rásin er valin. ch -2 stöðvar mynstur n á hverri rás. n -1 stöðvar rás, eða hverja rás þegar ch er sleppt. n -2 leyfir lykkjumynstri að klárast í stað þess að lykkja. sfx() án viðfanga skilar -1.

```
ch = sfx(0)
```

sfxget

`sfxget(n, step)`

Skilar töflu með einu mynsturskrefi: [1] tónhæð, [2] bylgja, [3] hljóðstyrkur, [4] áhrif. Tónhæðin er 0..63 (C-0 til D#5). Bylgja er 0..15. Hljóðstyrkur er 0..7 og 0 er hljóðlaus. Áhrif eru 0..7.

```
step = sfxget(0, 0)
pitch = step[1]
```

sfxset

`sfxset(n, step, pitch, wave, vol, fx)`

Skrifar eitt mynsturskref. Öll sex færibreyturnar eru nauðsynlegar.

```
sfxset(0, 0, 24, 3, 5, 0)
```

sfxhdr

`sfxhdr(n, [key, val])`

Haus fyrir mynstur. sfxhdr(n) skilar haustöflunni. sfxhdr(n, key, val) setur einn reit: speed, loop_in, loop_out, level, noise, crush, dtune, rverb, dampn, swing, steps. Tölur 0..10 velja sömu reiti í þeirri röð. speed er TIC, ticks á hvert skref. Hærra er hægara. 15 er 120 slög á mínútu. 16-bita samþykkja 1..255. Þrengri synths festast við TIC listann sinn.

```
sfxhdr(0, "speed", 15)
```

sfxstat

`sfxstat([ch])`

Mynstursauðkenni spilar á rás ch, eða -1. Sjálfgefin rás er 0.

```
if sfxstat(0) < 0 then sfx(1) end
```

note

`note(pitch, [wave, vol, fx, ch, ticks])`

Spilar eina nótu án mynsturs. Tónhæð 0..63, bylgja 0..15, hljóðstyrkur 0..7, áhrif 0..7. Tikk er hversu lengi það helst, í sömu 120 Hz tíkkum og TIC. Ef tíkkum er sleppt endist það í 15, eitt sjálfgefið skref. Áhrifin ganga yfir þann tíma, síðan þagnar rásin.

```
note(36, 3, 5, 5, 0)
```

music

`music([n, fade, mask])`

Byrjar lag n að ofan. music(-1) hættir. Ef kallað er á music með auðkenninu sem er þegar í spilun lætur það keyra. music() skilar auðkenni lagsins sem er í spilun, eða -1. fade er millisekúndur: lagið dofnar inn eða dofnar út þegar n er -1. Grímubitar sem eru stilltir eru rásirnar sem notaðar eru fyrir lagið. 0 þýðir allar átta. Hinar rásirnar eru lausar fyrir hljóðbylgjur.

```
music(0, 1000, 7)
```

musicstat

`musicstat()`

Tafla yfir lagið sem er í spilun. lag og röð eru auðkenni lagsins og röðin, eða -1 þegar ekkert er í spilun. skref og pos hafa átta tölur. skref[1] er rás 0, skref[8] er rás 7. skref er nótnavísitalan, eða -1 þegar rásin er hljóðlaus. pos er hversu langt nótan hefur farið, frá 0 til 1. tick er hversu mörg tick þessi röð hefur þegar spilað. played er hversu margar raðir hafa byrjað síðan music() var notað, þar með talið sú sem er í spilun.

```
st = musicstat()
row = st.row
note = st.step[1]
```

songget

`songget(song, [row, ch])`

Mynstursauðkenni geymt í lagareit. ch er 0..7. 255 þýðir að reiturinn er tómur. Þetta er virkt kall; vísbendingin í kóðanum sýnir enn eldri færibreyturöð.

```
id = songget(0, 0, 0)
```

songset

`songset(song, row, ch, pattern)`

Skrifar mynsturkenni í lagareit. Notið 255 til að hreinsa reitinn.

```
songset(0, 0, 0, 1)
```

stat

`stat(n)`

Aðeins hljóð. 16..23 er sfx auðkenni á rás n-16. 24..31 er tónlistarmynstrið á rás n-24. Sérhver önnur n skilar 0. Rammatími og vinnsluminni eru afköst og upplýsingar, ekki tölfræði.

```
id = stat(16)
```

Gagnatöflur

dget

`dget(t, row, col)`

Les reit úr gagnatöflu t úr DATA spjaldinu. Röð og dálkur byrja á 0.

```
hp = dget(0, 0, 1)
```

dset

`dset(t, row, col, v)`

Skrifar reit. Notar streng fyrir textadálk.

```
dset(0, 0, 0, "slime")
```

dstr

`dstr(t, row, col)`

Fruman sem texti. Tölur eru umbreyttar.

```
name = dstr(0, row, 0)
```

drows

`drows(t)`

Hversu margar raðir tafla t hefur.

```
for r = 0, drows(0) - 1 do
end
```

dcols

`dcols(t)`

Hversu marga dálka hefur tafla t?

```
cols = dcols(0)
```

Vista og afrek

save

`save(k, v)`

Geymir v undir lyklinum k í vistunarskránni. v getur verið tala, strengur eða tafla af þessum. Skráin er þjöppuð.

```
save("hp", hp)
```

load

`load(k)`

Les lykil k. Skilar núlli þegar lykillinn vantar.

```
hp = load("hp") or 3
```

delsave

`delsave(k)`

Eyðir lykil k úr vistuninni.

```
delsave("hp")
```

saveslot

`saveslot([name])`

Velur vistunarskrána í möppu höfundarins. Án nafns er rifinn titill leiksins. Skilar 1 þegar skráin er þegar til.

```
if saveslot("slot1") == 1 then
  hp = load("hp") or 3
end
```

achieve

`achieve(i)`

Opnar afrek i, frá 0 til 15. Geymt staðbundið. Steam-smíð setur einnig ACH_i. Skráningarlota tilkynnir það á reikning spilarans.

```
if coins >= 100 then achieve(0) end
```

achieved

`achieved(i)`

1 þegar afrek i er þegar opið.

```
if achieved(0) == 1 then spr(8, 4, 4) end
```

achdef

`achdef(i, text, [spr, tex])`

Skál fyrir afrek i, sýnt í 15 sekúndur frá neðri hluta skjásins. spr er valfrjáls 8×8 flís, textaðu blaðið sitt.

```
achdef(0, "rich", 8)
```

achclear

`achclear(i)`

Hreinsar afrek i í staðbundinni skrá. Steam ACH_i sem var þegar stillt helst stillt.

```
if btnd(5) then achclear(0) end
```

Net

whoami

`whoami()`

Spilaraauðkenni þitt, 0..15, á meðan lota stendur yfir. Ótengdur er það -1.

```
id = whoami()
```

nhost

`nhost()`

1 ef þessi vél er hýsillinn, annars 0.

```
if nhost() == 1 then spawn() end
```

nplayers

`nplayers()`

Hversu mörg lifandi auðkenni eru til í 0..15.

```
print(nplayers(), 4, 4, 7)
```

nlisten

`nlisten()`

Byrjar að hýsa UDP og skilar LAN-vistfanginu sem ip:port.

```
addr = nlisten()
```

njoin

`njoin(addr)`

Tengist ip eða ip:port.

```
njoin("10.0.0.8:7777")
```

nleave

`nleave()`

Yfirgefur lotuna. Jafningjar sjá _mp_leave með af hverju 0.

```
if btnd(11) then nleave() end
```

nsend

`nsend([p,] name, ...)`

Sendir tapsrík skilaboð. Nafnið er atburðurinn. Allt að fjögur auka gildi fylgja í kjölfarið. p frá 0 til 15 hvísl til þess jafningja; slepptu p til að senda til allra.

```
nsend("pos", x, y)
```

ncall

`ncall([p,] name, ...)`

Sama lögun og nsend, en skilaboðin eru endurtekin þar til þau eru staðfest. Þrjár mistök í röð eyðileggja lotuna.

```
ncall("hit", target, dmg)
```

nlan

`nlan()`

Núverandi LAN ip:port eftir nlisten.

```
print(nlan(), 2, 2, 7)
```

nstun

`nstun()`

Opinbera IP-talan þín frá STUN. Gestur á öðru neti getur reynt að tengjast því.

```
pub = nstun()
```

setturn

`setturn(addr, [user, pass])`

TURN varaleið þegar bein slóð mistekst. setturn() með tómum færibreytum slekkur á því. Vistfangið er host:port.

```
setturn("turn.example:3478", user, pass)
```

nlist

`nlist(game)`

Tafla yfir opnar lotur fyrir þann leik. Hver færsla inniheldur heimilisfang, spilara og nafn.

```
rooms = nlist("duel")
```

nannounce

`nannounce(game)`

Hjartsláttur svo þessi gestgjafi birtist í nlist fyrir leikinn. Kallaðu á hann á tímastilli meðan þú hýsir.

```
nannounce("duel")
```

nkb

`nkb([on])`

Heimilisfangsreitur. nkb(1) opnar hann, nkb(0) felur hann, nkb() segir til um hvort hann sé opinn. Upp og niður breytir staf, 0–9 eða punkti eða tvípunkti. A skrifar hann. X eyðir. Lyklaborðið skrifar ekki inn í hann.

```
nkb(1)
```

nkbstr

`nkbstr([s])`

Texti í veffangsreitnum, síðan breytast stafirnir upp og niður. nkbstr("") hreinsar textann.

```
addr, digit = nkbstr()
```

_mp

`_mp(from, name, ...)`

Þú skrifar þetta fall. Það keyrir þegar skilaboð berast. `from` er auðkenni sendanda. `name` er atburðurinn. Allt að fjögur gildi fylgja í kjölfarið.

```
function _mp(from, name, a, b)
  if name == "pos" then
    px[from] = a
    py[from] = b
  end
end
```

_mp_join

`_mp_join(p)`

Þú skrifar þetta fall. Jafningi fékk auðkennið p. Það ræsist líka fyrir þig; whoami() er það auðkenni.

```
function _mp_join(p)
  add(log, p)
end
```

_mp_leave

`_mp_leave(p, why)`

Þú skrifar þetta fall. Jafningi p til vinstri. Af hverju 0 er eðlilegt að fara. Af hverju 1 er rofin tenging.

```
function _mp_leave(p, why)
  px[p] = nil
end
```

_mp_update

`_mp_update()`

Þú skrifar þetta fall. Það keyrir á 15 _update ramma fresti, um 4 Hz. Góður staður til að senda staðsetningar.

```
function _mp_update()
  nsend("pos", x, y)
end
```

_mp_fail

`_mp_fail()`

Þú skrifar þetta fall. Lotan er dauð: gestgjafinn hvarf eða ncall mistókst þrisvar sinnum.

```
function _mp_fail()
  online = false
end
```

Coroutines

cocreate

`cocreate(f)`

Býr til kórútínu úr fallinu f. Hún keyrir ekki fyrr en kórónutína er notuð.

```
co = cocreate(cutscene)
```

coresume

`coresume(c, ...)`

Keyrir corutine c þar til það gefur af sér eða skilar. Við árangur: satt og skilar gildinu. Við villu: ósatt og villan.

```
ok, a = coresume(co)
```

costatus

`costatus(c)`

„í stöðvun“, „í gangi“ eða „dauður“.

```
if costatus(co) == "dead" then co = nil end
```

yield

`yield(...)`

Gefur stjórn aftur til þess sem kallaði á kjarnasummu. Gildin verða niðurstöður þeirrar kjarnasummu.

```
yield(1)
```

Mæla

info

`info([x, y, col])`

Teiknar lifandi HUD í _draw: fps (meðaltal 1 sekúndu), upd, draw, px, max, ram, peak, ramc og net þegar fundur er liðinn. info() notar prentbendilinn. info(col) stillir aðeins litinn. info(x, y) eða info(x, y, col) setur hann inn. Tölur nota B, k eða m svo þær passi á lítinn skjá. Símtalið sjálft er ekki talið í px.

```
info(1, 1, 7)
```

showstats

`showstats([on])`

Sama HUD og info, teiknað sjálfkrafa við 0, 0 eftir _draw, hunsað myndavél og myndskeið. showstats(0) felur það. showstats(1) eða showstats() sýnir það.

```
showstats(1)
```

perf

`perf(n)`

Ein HUD tala, svo þú getir teiknað hana sjálfur. 0 fps, 1 uppfærsla, 2 teikningar, 3 px, 4 max (þyngsti ramminn þessa sekúndu), 5 vinnsluminni, 6 vinnsluminni (vinnsluminni ásamt pakkaðri leikjakassa), 7 hámarksvinnsluminni frá upphafi leikjakassans.

```
print(perf(0), 2, 2, 7)
```