PIX37 Priručnik

Urednik, zatim kod. Izvanmrežna kopija knjige na pix37.party/docs.

Urednik

PIX37 DEV je editor. Ovdje crtate, pišete i slušate igru, a zatim izvozite uložak s igrom ili samostalnu verziju. Također može reproducirati uložak s igrom i otvoriti PIX37.PARTY za preuzimanje uložaka s igrom i korištenje kataloga. PIX37 DEV EDUCATIONAL je isti editor i besplatan je. Nema verzije, nema izvoza uložaka s igrom i nema pristup PIX37.PARTY. PIX37 Console je igrač. Reproducira `.p37` uloške s igrom, a vi otvarate PIX37.PARTY za preuzimanje uložaka s igrom i pregledavanje kataloga.

Projekti i patrone za igre

Datoteka `.pix37` je projekt: izvorne kartice, sprite listovi, mape, zvuk i tablice podataka. Datoteka `.p37` je uložak igre koji igrač pokreće: čvrsto zbijeni bajtkod.

Spremljene datoteke i postignuća nalaze se u mapi autora pored podataka aplikacije. Na Windowsima to izgleda kao `C:\\Users\\Ada\\AppData\\Roaming\\PIX37\\Mira\\Pong.p37s` za spremanje i `C:\\Users\\Ada\\AppData\\Roaming\\PIX37\\Mira\\Pong.ach` za postignuća. Na Linuxu su iste datoteke `~/.local/share/PIX37/Mira/Pong.p37s` i `~/.local/share/PIX37/Mira/Pong.ach`. Ada je račun na računalu. Mira je ime autora, a to ime je mapa. Pong je mjesto za spremanje, koje se prema zadanim postavkama odnosi na naziv igre. `save` zapisuje `.p37s`. Prvo otključano postignuće zapisuje `.ach`. Ako su datoteke već tamo, trkač ih čita kada se pokrene uložak igre.

Izbornik za razvojne programere

Tipka Escape otvara razvojni izbornik. Tipka Escape dok je igra pokrenuta vraća vas u uređivač. Potvrdite tipkom Enter ili tipkom A (zadano je V). Tipka Back je tipka B (zadano je C).

Bez učitanog projekta, izbornik je: STVORI PROJEKT, UČITAJ PROJEKT, IZGRADI, UČITAJ uložak s igrom, POVEŽI SE S PIX37.PARTY, POSTAVKE.

S učitanim projektom to je: SPREMI PROJEKT, UČITAJ PROJEKT, STVORI PROJEKT, SREDSTVA, IZGRADI, UČITAJ uložak s igrom, POVEŽI SE S PIX37.PARTY, POSTAVKE.

Ctrl+S sprema otvoreni projekt. Ctrl+R ga pokreće.

Izradi projekt

CREATE PROJECT traži uložak za igru prije nego što počnete izrađivati projekt.

- **VIDEO** — 1-BITNI (2 boje), 2-BITNI (4), 4-BITNI (16), 8-BITNI (256).
- **AUDIO** — dubina uzorka sintisajzera: 16, 8, 4, 2 ili 1 bit.
- **ZASLON** — unaprijed postavljene postavke uključuju 64×64, 128×128, 160×90, 256×256 i 320×180. PRILAGOĐENA veličina omogućuje vam odabir bilo koje veličine od 1×1 do 512×512.
- **PALETE** — banke boja. Prva banka počinje kao skup od 16 boja prikazan ispod. UČITAJ PALETU primjenjuje ugrađeni skup: PICO-8 16, PICO-8 32, Game Boy, 1-bit, CGA, Sweetie-16 i duži popis preuzet iz Lospeca (Resurrect 64, Apollo, Endesga 32 i 64, CC-29, SLSO8, Pear36, Oil 6, Duel, AAP-64, Lost Century i drugi). Također možete učitati datoteku palete ili učitati vlastiti PNG i izvaditi paletu iz te slike.



Indeksi boja koje koriste `cls`, `pset`, `spr` i ostali. Možete prebojiti bilo koji indeks u uređivaču paleta.

Pet panela

Uređivač je GAME, PAINT, CODE, SOUND i DATA, tim redoslijedom. Mapa se nalazi unutar PAINT-a. SFX i pjesme se nalaze unutar SOUND-a. Kliknite na ploču da biste tamo radili. Kada je CODE u fokusu, SOUND i DATA ignoriraju tipke sa slovima pa tipkanje ne svira i klavir.



Puni raspored, napisan kao `1:1:4:1:1`. Svaki broj predstavlja koliko retka zauzima ta ploča: GAME, PAINT, CODE, SOUND, PODACI. Mali zaslon smješten je u sredini ćelije veličine 128 × 128. Zaslon veći od 128 piksela proširuje

stupac GAME.

POSTAVKE → GLOBALNO → IZGLED mijenja rasporede:

Kodirati	Što vidiš
`1:1:4:1:1`	Sve. IGRA i ZVUK složeni, BOJA i PODACI složeni, KOD s desne strane.
`1:1:4:1:1 2X`	Isti raspored. KOD se crta s dvostrukom gustoćom, tako da stane dvostruko više teksta.
`1:0:1:0:1`	IGRA, PODACI, KOD.
`1:0:1:0:1 2X`	Isti raspored, s gušćim KODOM.
`0:1:1:1:1`	Nema pregleda igre. BOJANJE, KOD, ZVUK, PODACI.
`0:1:1:1:1 2X`	Isti raspored, s gušćim KODOM.
`0:1:4:1:2`	Studio. Široki KOD, viša PODATNA traka, BOJA i ZVUK.
`0:1:4:1:2 2X`	Isti raspored, s gušćim KODOM.
`0:0:4:0:4`	Samo KOD i PODACI, oba velika.
`0:0:4:0:4 2X`	Isti raspored, s gušćim KODOM.
`0:1:4:1:0`	BOJA, ZVUK i široki KOD.
`0:1:4:1:0 2X`	Isti raspored, s gušćim KODOM.
`0:4:0:0:0`	Samo BOJA, 256×256, jedan piksel lista na jedan piksel platna.

Redak `2X` zadržava iste panele. Panel CODE crta se u dvostrukoj rezoluciji i smanjuje se na istu ćeliju, tako da su tekst, kartice i rub između panela upola manji, a na ekran stane dvostruko više izvora.

IGRA

Pregled aktivne patrone igre. Kliknite ploču GAME ili pritisnite Ctrl+R za pokretanje pregleda. Ploča se konfigurira u POSTAVKE → PLOČA GAME:

- **Pregled** — boja u stanju mirovanja, kontinuirano ili automatsko ponovno učitavanje prilikom uređivanja.
- **Igraj u novom prozoru** — igra se otvara pored editora.
- **Prikaz** — pregled kao RGB888 ili RGB565.
- **IGRA 64** — zaslon veličine 64×64 je ili centriran 1:1 u ćeliji ili skaliran 2× kako bi se ispunila.
- **Shaderi** — CRT i filteri u igri, odvojeni od filtera u editoru.
- **Boja** — pozadina panela.

BOJA

Sprite listovi i, na strani karte iste ploče, pločice s kartama. Listovi mogu biti veći od zaslona. Veličine pregleda spriteova su 8, 16, 32, 64 i 128 (128 samo kada list ima najmanje 128 px). Pomičite kotačić miša preko pregleda spriteova da biste ga zumirali. Veličine kista su 1, 2, 3 i 4.

Ključ	Alat
B	Olovka
M	Odabir pravokutnika
P	Laso
S	Pečat
G	Kanta
Ja	Kapaljka
V	Potez
L / Shift+L	Linija / krivulja
U / Shift+U	Šuplji pravokutnik / šuplja elipsa
F / Shift+F	Ispunjeni pravokutnik / ispunjena elipsa
D	Treperenje
[ ]	Manja / veća četka
Ctrl+Z / Ctrl+Y	Poništi / ponovi. Ctrl+Shift+Z također ponavlja.
Ctrl+C / Ctrl+X / Ctrl+V	Kopiraj, izreži, zalijepi
Ctrl+A	Odaberite platno

Karta koristi istu ideju, s manje alata: B olovka, M odabir pravokutnika, S pečat, V pomicanje i Ctrl+Z, Ctrl+Y, Ctrl+C, Ctrl+V. Držite Alt za zumiranje teksture karte. Veličina kista na karti ostaje kao i alati za slikanje.

Svaki sprite ima osam zastavica. Paint ih postavlja na sprite; `fget` i `fset` ih čitaju u igri.

POSTAVKE → UREĐIVAČ BOJA je mjesto gdje se konfigurira ta ploča:

- **Paleta** — koja je banka nacrtana, plus učitavanje i spremanje.
- **Pregled** — RGB888 ili RGB565.
- **Kursor kista** — pokazivač dok slikate.
- **Prianjanje piksela** — kako se kist prianja uz piksele.

- **Veliki kist** — veliki kist pukne iz gornjeg lijevog kuta ili iz cijelog svog otiska.
- **Ljuska luka** - duh drugih okvira i u kojem smjeru je taj duh okrenut.
- **Zumiranje karte** — pravi pikseli ili besplatno zumiranje.
- **Boje** — boje korisničkog sučelja za bojanje.

KODIRATI

Kartice su zasebne datoteke unutar projekta. Kliknite naziv kartice da biste je preimenovali; Enter potvrđuje. Nekoliko kartica koje definiraju `_init`, `_update` ili `_draw` spajaju se redoslijedom kartica: prva kartica se izvršava prva, posljednja kartica se izvršava posljednja. Kôd koji nije unutar udice postaje dio `_init`. Uložak igre bez ikakvih udica tretira se kao jedan `_draw`.

Ključ	Akcijski
Strelice	Pomakni kurzor. Shift proširuje odabir.
Ctrl+Lijevo / Desno	Riječju.
Početna / Kraj, Ctrl+Početna / Kraj	Početak ili kraj retka ili tabulatora.
Tab	Umetnite dva razmaka. U popisu za dovršavanje prihvatite ime.
Unesi	Novi redak, uvučen kao prethodni. Prihvaća dovršavanje.
Ctrl+C / X / V	Kopiraj, izreži, zalijepi.
Ctrl+Z / Ctrl+Y	Poništi / ponovi.
Ctrl+A	Odaberite karticu.
Ctrl+D	Duplicirajte liniju ili odabir prema dolje.
Ctrl+/	Komentirajte ili uklonite komentar s `--`.
Ctrl+F	Pronađi. F3 prelazi na sljedeći pogodak. Escape zatvara traku.
Ctrl+G	Idi na broj retka.
Pobjeći	Prvo zatvara tražilicu, idi na ili popis s nagovještajima. Drugim pritiskom na tipku Escape uređivač se otvara u razvojni izbornik.

Savjeti se pojavljuju dok upisujete naziv funkcije. Enter ili Tab umeće naziv, a `(` ako je COMPLETE (uključeno u postavkama. POSTAVKE → UREĐIVAČ KODA:

- **Pretraživanje** — Ctrl+F u aktivnoj kartici ili na svakoj kartici.
- **Visina teksta** — izrežite glifove na 5 px ili prikažite cijeli glif.

- **Veličina niza** — veličina fonta literala niza znakova.
- **Font komentara** — 3×5 ili Fusion font trenutnog jezika korisničkog sučelja.
- **Dopuni (** — označava li dovršetak tip zagrade.
- **Savjeti za kod** - uključeni ili isključeni.
- **Jezik savjeta** — jezik izbornika ili engleski.
- **Razmak za umetanje** — dodatni pomak na praznim redovima.
- **Oluk** — brojevi redaka.
- **Boja** — pozadina, komentari, tabulatori, brojevi, operatori, ključne riječi, nizovi znakova, funkcije sustava, vaše funkcije, varijable, zagrade, kurzor, odabir, linija pogreške, upozorenje.

ZVUK

Ploča ima dvije kartice. SFX je jedan uzorak: kratki niz nota. SOUND je pjesma: postavlja te uzorke na osam kanala. TIC je koliko otkucaja jedan korak čeka. Takt je 120 otkucaja u sekundi, a jedan korak je šesnaestinka, pa je 15 otkucaja 120 otkucaja u minuti. Viši TIC je sporiji uzorak. Dva uzorka ostaju u koraku kada jedan TIC dijeli drugi: 15 i 30 da, 15 i 8 ne.

Klavir svira kada se nota odabere mišem. Bijele tipke su ZXCVBNM (CDEFGAB). Crne tipke su SDGHJ (C# D# F# G# A#). Zadržite pokazivač miša iznad bilo koje vrijednosti i upišite broj ili zadržite pokazivač miša i pomičite kotačić. Brojevi visine tona u kodu kreću se od 0 (C-0) do 63 (D#5).

Na jednom uzorku, LOOP i LEN su ista dva broja. OUT iznad 0 se ponavlja od IN do OUT, a naslov glasi LOOP. OUT na 0 i IN iznad 0 je LEN: toliko nota se svira od početka, a zatim uzorak završava, a naslov glasi LEN. Oba na 0 sviraju svaki korak.

Koliko sintetizatora možete uređivati ovisi o odabranoj bitnoj dubini za uložak igre. POSTAVKE → SFX UREĐIVAČ postavlja tu dubinu, hoće li se nota automatski pomicati naprijed (isključeno, lanac nota ili lanac polja), hoće li Space ponovno pokrenuti reprodukciju ili će je uključivati/isključivati, te boje editora.

16-bitni

Svaki korak ima notu, glasnoću, val i efekt. Glasnoća je od 0 do 7. 0 je tiho.

Valovi:

- **0 trokut**
- **1 nagnuta pila**
- **2 pile**
- **3 kvadrata**
- **4 pulsa** - uski puls
- **5 orgulja**
- **6 šum**
- **7 fazora**

- **8 PWM** — kvadrat čija se širina pomiče
- **9 puls 25%**
- **10 sinus**
- **11 metalni zvuk**
- **Zvono od 12 FM**
- **13 trzanje**
- **14 superpila**
- **15 PWM (C64)**

Efekti na jednoj noti:

- **0 nijedan**
- **1 slajd** — visina tona i glasnoća klize do sljedeće note u ovom koraku
- **2 vibrato** — kolebanje visine tona
- **3 pada** — visina tona pada u tišinu, uobičajeni udarac
- **4 postupnog pojavljivanja**
- **5 blijeđenja**
- **6 arpeggia** — hoda ovu notu i sljedeće tri, po jedan svakih 8 taktova. Na TIC 8 ili niže, svaka 4 takta.
- **7 brzi arpeggio** — iste četiri note, svaka 4 takta ili svaka 2 kada je TIC 8 ili manje.

Zaglavlje uzorka, iznad bilješki:

- **TIC** — otkucaji po koraku, od 1 do 255. 15 je 120 otkucaja u minuti. Upišite broj ili ga povećajte za jedan.
- **LOOP IN / OUT** — s OUT iznad 0, uzorak se ponavlja od IN do OUT. Ako OUT ostavite na 0, naslov postaje LEN: IN označava broj nota koje se sviraju, a 0 označava cijeli uzorak.
- **BUKA** — cijeli uzorak postaje buka.
- **CRUSH** — živahna varijanta svakog vala.
- **DTUNE** — 0 isključeno, 1 lagani refren, 2 snažan refren.
- **RVERB** — reverb: 0 isključeno, 1 kratko, 2 dugo.
- **DAMPN** — niskopropusni filter: 0 isključeno, 1 slabo, 2 jako.
- **RAZINA** — glasnoća cijelog uzorka, od 0 do 7.
- **SWING** — miješanje. Neparni koraci čekaju malo duže, parni malo kraće.
- **KORACI** — koliko koraka uzorak koristi: 32, 24, 16 ili 12.

8-bitni

Korak ima notu i val. Četiri vala su kvadratni, trokutasti, pilasti i šum. Zaglavlje ima TIC, LOOP IN / OUT, FX i LEVEL. TIC je jedan od 1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 30, 40, 60. FX je jedan efekt za cijeli uzorak: 0 nema, 1 vibrato, 2 arpeggio, 3 fade out. LEVEL je od 0 do 3.

4-bitni

Korak je samo nota. Zaglavlje ima TIC, LOOP (uključeno ili isključeno), WAVE (ista četiri vala) i OCT (u kojoj oktavi se nalaze note). TIC je jedna od 4, 6, 8, 10, 12, 15, 20, 30.

2-bitni

Korak je prazan, bubanj, udarac ili slomljen. Zaglavlje ima TIC i BASE. TIC je 8, 15, 16 ili 30. BASE postavlja koliko visoko se nalaze bubanj i udarac.

1-bit

Korak je nota. TIC je 8, 15, 16 ili 30, iste četiri brzine kao i 2-bitni.

Pjesme

Kartica ZVUK predstavlja otvorenu pjesmu. Osam stupaca označava kanale. Ćelija je ID uzorka. Broj 255, ako je prazan, znači da je kanal u tom retku tih. Stupac X je naredba: nastavak, zaustavljanje, početak petlje ili kraj petlje. Zaustavljanje završava pjesmu nakon retka. Kraj petlje vraća se na posljednji početak petlje.

Red završava kada se završi krajnji lijevi uzorak s LOOP OUT na 0. Uzorci dalje udesno ne produžuju taj red: dulji se reže, kraći ponovno počinje od svog prvog koraka dok se red ne završi. Uzorak s LOOP OUT iznad 0 petlja se između LOOP IN i LOOP OUT i ne završava red. Ako se svaki uzorak u redu petlja, red tamo ostaje.

Prečaci

Ključ	Akcijski
Tipke klavira	Napišite bilješku o koraku odabranom mišem.
Gore / Dolje	Promijenite vrijednost ispod kursora.
Prostor	Reproducirajte otvoreni SFX ili otvorenu pjesmu iz prvog reda. Pokrenite/zaustavite ili ponovno pokrenite iz postavki SFX EDITOR-a.
Ctrl+Z / Ctrl+Y	Poništi / ponovi.
Ctrl+C / Ctrl+X / Ctrl+V	Kopiraj, izreži, zalijepi odabir.

PODACI

Imenovane tablice. Svaka tablica se sastoji od redaka i stupaca. Iz koda, `dget`, `dset`, `dstr`, `drows` i `dcols` koriste ove tablice. Brojevi redaka i stupaca u kodu počinju od 0.

Vrste stupaca, ciklički mijenjane u zaglavlju:

- **bool** — jedan bit.
- **2** — dva bita.
- **4** — četiri bita.
- **8** — bajt.
- **16** — 16 bitova, za ID-ove i veće brojeve.
- **spr** — ID spritea.
- **str** — tekst, nacrtan sistemskim fontom.
- **str8** i **str10** — isti tekst, nacrtan više.
- **f16** i **f32** - float s polupreciznošću i float s punom preciznošću.

S pokazivačem na zaglavlju stupca, Ctrl+pomicanje mijenja vrstu tog stupca. Obično pomicanje pomiče se preko stupaca ili niz retke, ovisno o tome što je odabrano u POSTAVKE → UREĐIVAČ PODATAKA. Ctrl mijenja taj smjer, zbog čega Ctrl+pomicanje na zaglavlju mijenja vrstu kada je kotačić postavljen na stupce.

Tipke sa strelicama služe za pomicanje. Tipka Enter uređuje ćeliju. POSTAVKE → UREĐIVAČ PODATAKA:

- **Kotačić** — pomicanje stupaca ili redaka.
- **Boje** — mreža, zaglavlje i odabir.

Imovina

ASSETS uvozi i izvozi otvoreni dio, ne cijeli projekt. Spriteovi koriste sprite otvoren u PAINT-u, na otvorenom listu. Mapa je otvorena mapa. SFX je otvoreni uzorak. Glazba je otvorena pjesma. Podaci su otvorena tablica. Kôd je iznimka: svaki tabulator se zapisuje i svaki tabulator se čita natrag.

Vrsta	Datoteka	Što se kreće
Spriteovi	`.png`	Otvoreni sprite.
Karta	`.p37map`	Otvorena karta.
Znakovi	`.p37sfx`	Otvoreni uzorak.
Glazba	`.p37mus`	Otvorena pjesma. Brojevi uzoraka ostaju kakvi jesu. Izvezite SFX zasebno ako vam trebaju ti zvukovi.
Podaci	`.p37dat`	Otvoreni stol.
Kodirati	`.lua`	Svaka kartica.

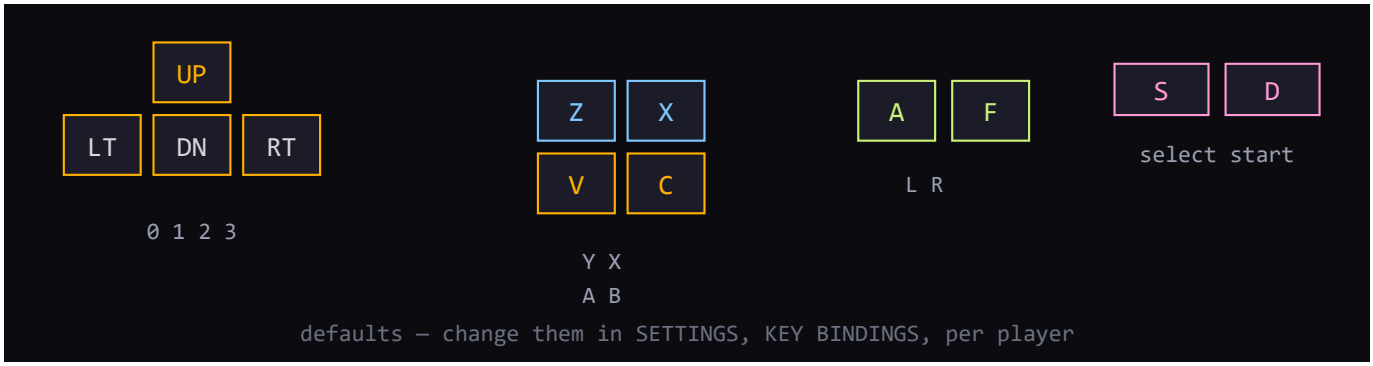
Postavke

Popis korijena:

Stranica	Što pohranjuje
PROFILI	Spremi i učitaj svaku postavku. Novo, preimenuj, prepiši, izbriši. Kopiraj ili zalijepi profil kao tekst.
Jelovnik	Prozor, tipke, boje izbornika, opcionalna pozadina.
PLOČA ZA IGRU	Pregled i prozor za reprodukciju.
UREDNIK KODA	Istaknuti elementi i nagovještaji.
UREDNIK BOJA	Kist, paleta, prikaz karte.
UREDNIK SFX-a	Sintizator i unos nota.
UREDNIK PODATAKA	Smjer i boje kotača.
JEZIK	Jezik izbornika i savjeta.
LICENCIJE	Tekstovi licenci, uključujući font Fusion Pixel i SDL3.

GLOBALNO:

- **RASPORED** — raspored panela, gore.
- **GLASNOĆA** — glavna razina, od 0 do 10.
- **VEZE TIPKI** — tipke za svakog od 8 igrača. Pritisnite tipku za snimanje. Upute, ABXYLR, odaberi.
- **Prezentiraj** — preskoči neaktivne okvire ili nacrtaj svaki okvir.
- **Cijeli zaslon** — bez obruba.
- **Osvježi** — broj FPS-ova u prozoru. Lijevi i desni korak 60, 90, 120 i maksimalno. Enter upisuje broj do 1000. 0 znači da nema ograničenja.
- **Vsync** — čekanje prikaza između okvira.
- **Skala** — množitelj cijelog broja prozora.
- **Shaderi** — CRT i filteri u editoru, ne u igri.
- **Boja** — tekst izbornika i isticanje.
- **Pozadina** — opcionalni PNG, izrezan na 128, iza izbornika. Opcija Očisti je uklanja.
- **Kursor uređivača** — oblik pokazivača.
- **Miš** — plutajući pokazivač ili pričvršćen na piksele.
- **Brzina korisničkog sučelja** — koliko se brzo zadržani smjer ponavlja kroz popis.
- **Explorer** — popisi datoteka počinju u korijenskoj mapi računala ili u mapi exe datoteke.



Zadana tipkovnica. A je potvrda u izbornicima. B je povratak. U igri su to tipke 4 i 5. Odaberi (S) je tipka 10, a ne prečac izbornika.

Gumb	Uloga	Zadani ključ
0	Lijevo	Lijeva strelica
1	Pravo	Desna strelica
2	Gore	Strelica prema gore
3	Dolje	Strelica prema dolje
4	A, potvrdi	V
5	B, natrag	C
6	X	X
7	Y	Ž
8	R	F
9	L	A
10	Odaberite	S
11	Start	D

Izgraditi

Odaberite platformu: Windows, Linux, Mac, Android, iOS ili Web.

- **Naziv** — naziv datoteke za izradu.
- **Autor** — na Windowsima i Linuxu. Ovo je naziv mape za spremanja.
- **ID Steam aplikacije** — na Windowsima i Linuxu. Prazno znači da nema Steamworksa. Ispunjeni ID mapira `achieve(0)`…`achieve(15)` na Steam postignuća ACH_0…ACH_15.

- **Id** — identifikator paketa gdje ga platforma traži.
- **Sakrij kod** — na prekrivačima za stolna računala i mobilne uređaje. Usporava `peek` uloška igre. Nije zaključavanje.
- **Cijeli zaslon** — izvezeni player pokreće se preko cijelog zaslona.
- **Shaderi** — do tri sloja ugrađena u player.
- **Preuzimanje .p37** — Samo za web. Prikazuje unos za preuzimanje u izborniku pauze.
- **Ikona** — pločica iz sprite sheeta: sheet, veličina, x, y. Ili PNG naslovnica koju izrežete.
- **Izvoz** — odaberite mapu. Izbornik prikazuje SPREMLJENA IZGRADNJA ili NEUSPJEŠNA IZGRADNJA.

Napravite verziju s uključenim internetom. Splash može anonimno pitati poslužitelj je li poznat hash ove verzije. Ta provjera postoji iz dva razloga. Štiti mene, autora PIX37, od kopija alata koje nisu bile moje da ih podijelim. Također štiti osobu koja je napravila igru. Uzimanje tuđe igre i njezino navođenje kao vlastite postaje uobičajeno na tržištu telefona, kao što su Play (Android) ili iOS. Igra se i dalje pokreće. Ako je računalo izvan mreže, uvod kaže da igra nije povezana. Ta rečenica je također nagovještaj igraču da kopija možda nije došla iz legalnog izvora. To je mala provjera, tako da ako je vaša igra ukradena i prodana bez vašeg pristanka od strane nekoga drugog, možete to učiniti u trgovini kao što su Play (Android), Steam ili itch.io. Poslužitelj pohranjuje vrijeme kada je hash prvi put registriran, pored hasha, i kasnije ga ne mijenja. Ranije vrijeme je originalna verzija. Hash registriran nakon toga je kasnija kopija.

Želim da ova provjera zaštiti ljude koji izrađuju igre. Engine već obavlja puno posla za njih, a ovo je jedna od tih stvari. Razumijem da može biti kontroverzno. Ako zajednica odluči da je bolje ukinuti ovaj mikro-DRM, hoću.

Stari uložak za igru iz prethodnog editora može odbiti pokretanje nakon promjene jezika. Otvorite projekt `.pix37` u ovom editoru i izvezite novi `.p37`.

Povežite se s PIX37.PARTY

CONNECT TO PIX37.PARTY je unutar aplikacije. Nema stranice za prijavu u pregledniku. Prijavljujete se sa Steam session ticketom ili putem itch.io, koristeći pločicu i miš. Usluga prikazuje jedinstveno ime za prikaz, od 2 do 16 znakova. Licencni ključ i Steam ID ne prikazuju se drugim igračima.

Nakon uvjeta usluge, središte su Favoriti, Katalog, Poredak, Igrači i vaš profil.

Razvojna licenca je licenca za uređivanje. Vi napravite igru, prenesete je i ostavljate zvjezdice. Igračka licenca je igrač. Srca su od ljudi koji igraju na trkaču.

- **Katalog** — pregledajte igre. Filtri pokrivaju žanr (do tri), veličinu zaslona, veličinu uloška igre i uključuje li prijenos `.pix37` projekt.
- **Favoriti** — igre koje ste označili.
- **Prijenos** — potrebna je licenca razvojnog programera. Šaljete `.p37` i možete priložiti projekt `.pix37`. `.p37` je pakirani bajtkod. Teško ga je pročitati i može se namjerno zamaskirati. `.pix37` je datoteka projekta u kojoj zapravo radite: izvorni kod, slike i zvuk, u obliku koji druga osoba može otvoriti i iz

njega učiti. Slanjem omogućujete drugim kreatorima da proučavaju igru umjesto komprimiranog uloška igre. Uz njega idu naslov, autor, žanrovi, naslovnica, ikona i nazivi postignuća. Poslani broj RAM-a predstavlja vrhunac koji je uložak igre dosegao.

- **Komentari i postignuća** — na stranici igre. Otključavanje postignuća dok ste prijavljeni pohranjuje se na tom računu kao i na računalu.
- **Poredak** — programeri, sortirani po zvjezdicama ili srcima. Svaki redak prikazuje oba broja. Otvorite ime da biste vidjeli igre tog autora, sortirane po srcima.
- **Igrači** — računi poredani prema otključanim postignućima.

Kodirati

Jezik je mala skripta u Lua stilu. Ova polovica počinje s praznim projektom, uči jezik, a zatim prikazuje kratki primjer svake funkcije.

Prvi program

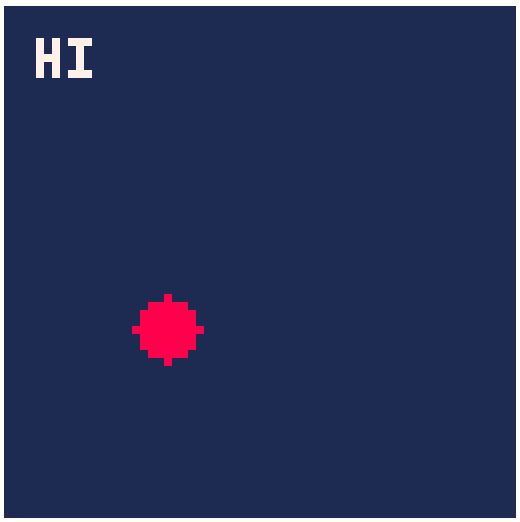
Izradite projekt u rezoluciji 64×64, 4-bitnoj boji. Otvorite CODE. Zamijenite tabulator s:

```
_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 pokreće. Tipke sa strelicama pomiču krug. Escape zaustavlja igru i vraća u editor.



Ista tri poziva: `cls(1)` boji tamnoplavi ekran, `print` crta HI, a `circfill` crta ružičasti krug.

Što se pokreće i kada

- `_init` se izvršava jednom.
- `_update` radi na 60 Hz. Ovdje postavite kretanje i tipke.
- `_draw` slika. Bez argumenta o fps-u, slika najbrže što računalo i monitor mogu. `_draw(30)` ograničava slikanje na 30 fps, a logiku ostavlja na 60.
- `_set_fps(n)` kasnije mijenja to ograničenje. 0 ga briše.

Imena nisu osjetljiva na velika i mala slova; uređivač ih pohranjuje malim slovima. `function` i `fun` su iste ključne riječi.

Jezik

Osnove programiranja

Program je popis redaka koje izvršava uložak igre. Svaki naslov prvo prikazuje oblik retka, a zatim mali primjer. Redak koji počinje s `--` je komentar: napomena za vas. Igra ga preskače.

Varijable

Varijabla je naziv koji sadrži vrijednost. Vi birate naziv. Kasniji redovi ga mogu pročitati i pohraniti novu vrijednost pod istim nazivom.

Oblik je `name = value`. `local name = value` zadržava to ime unutar trenutnog bloka, tako da ga ostatak kasete igre ne može vidjeti. Bez `local`, ime je globalno i svaka kartica ga može vidjeti.

```
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` znači da nema vrijednosti. `true` i `false` su da i ne. Naziv koji nikada niste postavili čita se kao `nil`.

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

Tablice

Tablica sadrži mnogo vrijednosti pod jednim imenom. Popis koristi brojeve. Prva stavka je 1, a ne 0. Zapis koristi imena. Tablica može sadržavati drugu tablicu.

Oblik popisa: `{ value, value }`. Oblik zapisa: `{ name = value, name = value }`. Popis zapisa je tablica čiji su elementi tablice.

```
-- 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
```

Stanje i tijelo

Uvjet je pitanje s odgovorom da ili ne, kao što je `hp > 0` ili `btnd(4)`. Tijelo je skupina redaka koji se pokreću kada uđete u blok. `if` i svaka petlja koristi isti oblik: uvjet, zatim tijelo, pa `end`. `repeat` završava s `until` umjesto s `end`.

Ako

`if` izvršava svoje tijelo samo kada je uvjet istinit. `elseif` pokušava drugi uvjet. `else` izvršava se kada su svi uvjeti bili lažni.

Oblik: `if condition then`, zatim tijelo, pa `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
```

Petlje

Petlja ponavlja tijelo. `while` prvo provjerava uvjet i nastavlja dok je istinit. `repeat` pokreće tijelo, a zatim provjerava, tako da se tijelo uvijek izvrši barem jednom. `for` broji. Uključen je zadnji broj. Bez trećeg broja korak je 1, pa ako je početak već prošao kraj, tijelo se ne izvršava. Odbrojavanje unatrag zahtijeva negativni korak.

```
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
```

Hodanje po stolu

`all` posjećuje svaku stavku popisa, od prve. `pairs` posjećuje svako ime i vrijednost pohranjenu pod tim imenom.

```
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
```

Funkcije

Funkcija je imenovani recept. Oblik je `function name(arguments)`, zatim tijelo, pa `end`. Pozivate je upisivanjem imena i zagrada. `return` šalje vrijednost natrag, a može poslati više od jedne. Argumenti su lokalne vrijednosti te funkcije.

```
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` je funkcija koja može vidjeti lokalne vrijednosti oko sebe. To je closure. Tablica može sadržavati jednu vrijednost, a poziva se s točkom.

```
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
```

Da, ne i biranje

`not` mijenja uvjet. `and` je istinit samo kada su obje strane istinite. `or` je istinit kada je bilo koja strana istinita. `a and b or c` bira `b` kada je `a` istinit, inače `c`. `cond ? a : b` je ista ideja u jednom retku: `a` kada je uvjet istinit, `b` kada nije.

```
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)
```

Matematika i tekst

`+ - * /` su zbrajanje, oduzimanje, množenje i dijeljenje. `//` dijeli i zadržava cijeli broj. `%` je ostatak. `^` je potencija. `..` spaja dva teksta. `#t` je broj stavki na popisu. Usporedbe su `==` (isto), `~=` ili `!=` (različite) i `< > <= >=`. `x += 1` zbraja 1 s x. Isti oblik funkcionira i za `-= *= /= ^= %=`.

```
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
```

Gdje živi mještanin

`local` traje sve dok `end` ne zatvori svoj blok. `if` / `end` je vlastiti blok. Možete spomenuti lokalnu vrijednost iznad retka koja ga postavlja, ali dok se taj redak ne pokrene, vrijednost je `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
```

Skakanje

`::label::` označava liniju. `goto label` skače tamo, ali samo unutar iste funkcije.

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

Pamćenje, ukratko

`peek` i `poke` gledaju na žive piksele putem jedne 16-bitne adrese. Ne čuvaju drugu kopiju kazete za igru. Adrese od `0x5F00` do `0x5FFF` su registri: kamera, isječak (samo za čitanje), boja crtanja, uzorak ispune, tri remapiranja palete i brojevi stranica. Bajtovi s obje strane te rupe, njih 65280, čine prozor piksela.

Kada zaslon stane u tih 65280 bajtova, 64×64 i 128×128 stanu, a 256×256 ne, zaslon počinje na adresi 0, redak po redak. Ono što preostaje je sprite sheet odabran od strane `0x5F41` (0 je sheet 0). Ako je taj sheet veći od preostalog prostora, `0x5F42` odabire stranicu i karta nije u prozoru. Ako sheet stane, rep su ćelije karte odabrane od strane `0x5F43`, jedan ili dva bajta svaka, a `0x5F44` stranice kartu kada ne stane u rep.

Kada zaslon ne stane, cijeli prozor je jedna stranica od 65280 piksela. `poke(0x5F40, n)` okreće stranicu zaslona. `poke(0x5F41, n)` s n različitim od 0 prikazuje list n umjesto toga, a list 0 nema prikaz u ovom načinu rada. `poke(0x5F43, n)` s n različitim od 0 prikazuje mapu n, i to pobjeđuje list. Mapa 0 u ovom načinu rada je samo `mget` i `mset`.

`peek` koji ne padne na piksel ili registar vraća 0. `poke` tamo ne radi ništa. Sprite zastavice, SFX i DATA tablice nisu u ovom prozoru. Koristite `fget`, `sfxget` i `dget`.

Praktična upotreba

Svaka funkcija navedena je na dnu ove stranice, s praktičnim primjenama PIX37 funkcija. Linkovi ispod svakog primjera otvaraju više informacija o funkcijama.



Glatko kretanje

Uložak za igru dimenzija 128×128, gledan odozgo. Sprite 1 je već nacrtan. Strelice ubrzavaju u bilo kojem smjeru, a trenje vas usporava. Sprite se omotava: polovica koja napušta dno nacrtana je na vrhu, a isto vrijedi i za ostale rubove.

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



Provjera je li sljedeći potez blokiran

Uložak za igru dimenzija 64×64. Pod i desni zid su već na karti, obojeni spriteom 3, a na tom spriteu je postavljena zastavica 0. Strelice pomiču sprite 1. Riječ na vrhu kaže blokirano kada bi se sljedeći korak preklopio s punom pločicom, a slobodno kada se ne bi.

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



Brisanje s okretanjem

Pritisnite A. Brisanje se crta pomoću flip(), jedan crni redak odjednom, umjesto unutar _draw(). _draw ne smeta dok se to izvršava, tako da svaki redak ostaje na ekranu.

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



Tablica funkcija (klasa)

Sprite 1 je već nacrtan. make_monster radi kao klasa. Vraća tablicu, a update i draw su metode koje čuvaju x i y tog čudovišta. Broj na vrhu je koliko je daleko hodalo.

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



Četiri načina za čitanje gumba

A je tipka 4. Držite je i naboj raste. Otpustite je i naboj se vraća na 0. Gore i dolje se ponavljaju dok držite tipku, tako da se redak izbornika nastavlja pomicati. btnd je prvi kadar pritiska, koji ovaj uložak za igru koristi samo kao početak tog držanja.

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



Pucaj mišem

Sprite 1 je već nacrtan u sredini. Bijela točka je kursor. Lijevi klik ispaljuje piksel prema njoj. Kotačić mijenja brzinu, ispisan u kutu.

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



Upišite ime i potvrdite ga

Pritisnite A za otvaranje polja. Gore i dolje mijenjajte trepćuće slovo. A ga piše, X briše, Y je razmak, a B potvrđuje. Linija zatim izgovara vaše ime.

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



Kamera na playeru

Uložak za igru dimenzija 128×128. Sprite 1 počinje u sredini poda od pločica. Strelice se pomiču. Kamera drži sprite centriran. x i y u kutu se ne pomiču jer camera() bez argumenata vraća pogled natrag za HUD.

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



Nekoliko redaka vizualnog romana

Redci su u PODACIMA, na engleskom i poljskom jeziku. A se nastavlja ili odabire ostaje kada je opcija na ekranu. B odabire odlazak na opciju, a inače mijenja jezik. X mijenja jezik dok je opcija prikazana. Dijalog koristi font od 8px tako da se poljska slova crtaju.

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



Slikajte s kuckanjem i bockanjem

Uložak za igru dimenzija 64×64, tako da je adresa 0 gornji lijevi piksel, a sljedeći redak počinje od 64. Uložak za igru zapisuje piksele, kopira redak, briše redak i ispisuje vrijednosti koje je pročitao.

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



Popis koji možete dodati

A zbraja broj, do 8. B uklanja posljednji. X uklanja prvi. Popis se ispisuje niz ekran.

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



Rastavljeno ime

Uložak za igru dijeli ada,12,blue i ispisuje ime, prvo slovo, 12, i riječ blue.

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



Sviraj notu, a zatim je promijeni

Uzorak 0 traje jednu notu i svako ponavljanje je iste glasnoće. A ga ponovno svira kada ta nota završi. B povisuje visinu tona i svira tu notu.

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

Pjesma napisana u uložak za igru

Dva uzorka se ponavljaju u retku 0, po jedan na svakom od prva dva kanala. Red ostaje 0. ch0 i ch1 su nota na kojoj je svaki kanal. Uređivanje uzorka, LEN i LOOP u ulošku igre način je da se vidi što te kontrole rade u glazbenoj ploči.

`songget songset music musicstat cls print _init _draw`

Tri čudovišta u DATA

Ploča s podacima već ima stupac s imenom i stupac s hp-om, sa sluzi, šišmišem i okom. A ozljeđuje sluz. Brojevi s desne strane su njezini hp-ovi.

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

Zapamti rezultat

A dodaje bod i sprema ga. Izlazi i ponovno otvara kasetu s igrom: broj je još uvijek tamo. B ga zaboravlja.

`load btnd save delsave cls print _init _update _draw`

Drugo poglavlje u drugoj datoteci

Dva uloška za igru dijele zadatak spremanja. Dio 1: A označava da je igra završena. Dio 2, otvoren prvi, kaže da se završi 1. dio. Nakon toga, otvara se 2. dio, a B briše spremanje kako bi se 1. dio mogao ponovno igrati.

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

Jedno postignuće

Sprite 8 je već nacrtan. Pritisnite A deset puta. Tost imenuje postignuće, a sprite 8 ostaje jer je otključan. B briše to postignuće u lokalnoj datoteci i resetira brojač.

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

Soba u koju možete ući

Uložak za igru veličine 256×256. Bijela točka je kursor. Izbornik je domaćin, pridruži se ili popis poslužitelja. Domaćin otvara sobu pod nazivom hodanje i pokreće igru. Pridruživanje pomiče adresu (znamenke, točka, dvotočka) i B se spaja. Popis poslužitelja pomiče se gore i dolje, a A se pridružuje redu. U igri, strelice hodaju - sprite 1. Jedan novčić se dijeli: domaćin ga pomiče, a njegovim podizanjem dodaje 1 rezultatu tog igrača i stvara sljedeći. Svaki sprite, uključujući i vaš, prikazuje taj rezultat pored sebe. Kut prikazuje koliko je igrača unutra, LAN adresu i javnu adresu domaćina. B odlazi: vi i ostali se vraćate u izbornik, a sljedeći domaćin započinje novu sobu s rezultatom 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`



Dijalog koji čeka

A prikazuje sljedeći redak. Funkcija se pauzira na svakom yield-u dok ponovno ne pritisnete A.

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



Pričekajte pola sekunde

Riječ se mijenja iz ... u hi nakon otprilike pola sekunde. Zaslon nastavlja crtati dok je korutina pauzirana.

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



Krugovi koji dolaze jedan po jedan

Pet krugova kaplje s vrha, udaljenih dva okvira, a zatim padaju.

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



Dva okvira, sa sata

Sprite 1 i sprite 2 su dva kadra istog znaka. Slika se mijenja četiri puta u sekundi.

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

Svaka funkcija

Argumenti u uglatim zagradama nisu obavezni. Indeksi tablica i nizova znakova počinju od 1, osim redaka i stupaca DATA, koji počinju od 0. Boje su indeksi paleta.

Indeks

`_crtati`
`_init`
`_mp`
`_mp_fail`
`_mp_pridružiti se`
`_mp_napustiti`
`_mp_ažuriranje`
`_set_fps`
`_ažurirati`
`trbušnjaci`
`achclear`
`achdef`
`postići`
`postignuto`
`dodati`
`sve`
`ustvrditi`
`atan2`
`bend`
`bnot`
`bor`
`btn`
`btnd`
`btnp`
`btnu`
`bxor`
`fotoaparat`
`strop`
`hr`
`cir`

circfill
clickl
clickm
clickr
isječak
cls
sukreirati
boja
jezgrorezime
cos
costatus
brojati
kursor
dcols
del
delikatesi
delsave
dget
drows
dset
dstr
fget
ispuniti
okrenuti
flr
foreach
fset
info
iparovi
ključ za isključivanje
keyon
keystat
keystr
linija
opterećenje
log
lshr
karta
max
memcpy
memset
mget
sredina
min
miš
mset
glazba
musicstat
najavljivati
ncall
sljedeći
nhost
pridružiti se
nkb
nkbstr
nlan
nnapustiti
nlist
nslušaj
bilješka
nigrači
nsend
nstun
red
ovalan
ovalna ispuna
pakirati
parovi
prijatelju
palrgb
palt
pcall
zaviriti
zaviriti2
zaviriti4
izvedba
pget
bocnuti
bocnuti2
bocnuti4
ispisati
ispis10
ispisati8
pset
ispravan
ispraviti
rnd
rotl
rotr
ispravan
rrectfill
spasiti
mjesto za spremanje
ekran
svitak
odabrati
setmap
setpaleta
postavlja pr
settex
setturn
sfx
sfxget
sfxhdr
sfxset
sfxstat
sget
sgn
shl
showstats
shr
grijeh
songget
set pjesama
rascijepiti se
spr
sqrt
srand
sset
sspr
stat
sub
vremena

tline	tostr	tkoami
tonum	raspakirati	prinos

Životni ciklus

_init

`_init() ... end`

Pokreće se jednom prilikom pokretanja uloška igre. Kod koji se nalazi izvan bilo koje udice također se ubacuje u _init.

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

_update

`_update() ... end`

Logika igre. Poziva se na 60 Hz, čak i kada je _draw ograničen ili je host sporiji. Pročitajte gumbe ovdje.

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

_draw

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

Crta jedan okvir. Bez argumenta ili 0, izvršava se onoliko često koliko se host prikazuje. Proslijedite n (1..999) za ograničavanje crtanja na n okvira u sekundi. _update ostaje na 60 Hz.

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

_set_fps

```
_set_fps(n)
```

Mijenja _draw cap dok je igra pokrenuta. 0 znači bez ograničenja. n je ograničen na 0..999.

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

flip

```
flip()
```

Prikazuje zaslon i ostavlja prostor do sljedećeg kadra. Koristite ga samo ako crtate vlastitu petlju kadra. Nemojte čekati nakon toga.

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

time

```
time()
```

Sekunde od _init.

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

Ulazni

btn

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

1 dok se drži tipka b, inače 0. p je igrač, 0..7. Bez argumenata vraća bitnu masku igrača 0. Tipke: 0 lijevo, 1 desno, 2 gore, 3 dolje, 4 A, 5 B, 6 X, 7 Y, 8 R, 9 L, 10 odaberi, 11 start.

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

btnd

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

Vrijedi samo na okviru _update kada se pritisne gumb b.

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

btnp

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

Istina pri pritisku, a zatim ponovno dok se drži: 30 otkucaja odgode, zatim svakih 8 otkucaja. Bez tipke vraća masku.

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

btnu

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

Istina je samo na okviru gumb otpušten.

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

mouse

```
mouse()
```

Vraća miš x, y u pikselima igre.

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

clickl

```
clickl([p])
```

Lijeva tipka miša. clickl() se drži pritisnuta. clickl(1) je rub pritiska.

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

clickr

```
clickr([p])
```

Desna tipka miša. Isti rub s 1 argumentom kao clickl.

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

clickm

```
clickm([p])
```

Srednja tipka ili klik kotačića za pomicanje.

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

scroll

```
scroll()
```

Pomicanje kotačića miša u ovom okviru. Pozitivno i negativno okretanje ovisi o smjeru kotačića.

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

keyon

`keyon()`

Otvara tekstualno polje kojim upravljaju gumbi. Gore i dolje mijenjaju slovo od A do Z. A piše to slovo. X briše. Y je razmak. B potvrđuje i zatvara polje. Tipkovnica ne tipka u njega.

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

keyoff

`keyoff()`

Skriva tekstualno polje.

```
keyoff()
```

keystr

`keystr([s])`

Tekst u polju, zatim 1 na okviru B potvrđuje, a zatim se slovo gore i dolje mijenjaju. To slovo je prazno nakon što se polje zatvori. keystr("") briše tekst.

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

keystat

`keystat()`

1 dok je tekstualno polje otvoreno, inače 0.

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

Izvlačenje

screen

`screen()`

Trenutna veličina zaslona kasete igre u pikselima. Vraća širinu, a zatim visinu. Ovo je razlučivost igre, a ne prozor uređivača.

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

cls

`cls([col])`

Briše ekran. Zadana boja je 0.

```
cls(1)
```

color

`color(col)`

Postavlja boju crtanja koja se koristi kada kasniji poziv izostavi col. Bez argumenta vraća trenutnu boju.

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

camera

`camera([x, y])`

Pomače svaki poziv crtanja. camera() resetira na 0, 0.

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

clip

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

Ograničava crtanje na pravokutnik. Peti argument različit od nule siječe se s trenutnim isječkom. clip() vraća puni zaslon. Pregled registara isječka je samo za čitanje.

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

cursor

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

Postavlja gdje ide sljedeći ispis bez koordinata i opcionalno boju. cursor() vraća x, y.

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

pset

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

Postavlja jedan piksel zaslona. Broji se u statistiku px.

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

pget

```
pget(x, y)
```

Boja piksela ekrana na x, y.

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

print

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

Crta tekst veličine 3×5 (UTF-8, izrezan iz Fusion fonta). Bez x i y ispisuje se na mjestu kursora i pomiče se u sljedeći redak.

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

log

```
log(...)
```

Dodaje argumente, odvojene razmacima, u tekstualni međuspremnik. pix37_run ispisuje taj međuspremnik na stdout. Ne crta.

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

print8

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

Isto kao i ispis, s fontom Fusion veličine 8 px.

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

print10

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

Isto kao i ispis, s fontom Fusion od 10 px.

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

line

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

Linija od x0,y0 do x1,y1. Dva argumenta nastavljaju se od prethodnog kraja. line() resetira taj lanac.

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

circ

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

Obris kruga. Polumjer je u pikselima.

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

circfill

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

Ispunjeni krug.

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

oval

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

Obris elipse unutar okvira za ograničavanje. Kutovi su uključivi.

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

ovalfill

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

Ispunjena elipsa unutar graničnog okvira.

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

rect

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

Obris pravokutnika. Kutovi su uključivi.

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

rectfill

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

Ispunjeni pravokutnik.

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

rrect

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

Obris zaobljenog pravokutnika. x i y su gore lijevo. w i h su veličina u pikselima; oba moraju biti veća od 0. r je polumjer kuta i zadana je vrijednost 0. Steže se na polovicu kraće stranice.

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

rrectfill

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

Ispunjeni zaobljeni pravokutnik. Isti argumenti kao i za rrect.

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

fillp

`fillp([p])`

16-bitni uzorak ispune za kasnije ispunjene oblike. 0 je puna boja. fillp() vraća trenutni uzorak. Bitovi hodaju po uzorku 4×4.

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

pal

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

Premapira boju c0 u c1. način rada 0 je paleta za crtanje, 1 paleta za prikaz, 2 paleta za ispunjavanje. pal() resetira.

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

palrgb

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

Postavlja jedan unos palete na RGB, svaki kanal 0..255.

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

palt

`palt([col, t])`

Prozirnost. palt(col) ili palt(col, true) čini col prozirnom. palt(col, false) je čini čvrstom. palt() čini svaku boju neprozirnom.

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

setpalette

```
setpalette(n)
```

Odabire banku paleta n. setpalette() vraća trenutnu banku.

```
setpalette(1)
```

tline

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

Crta liniju na zaslonu dok uzorkuje piksele karte s pozicije karte mx, my, mdx, mdy i pomiče uzorak karte duž linije.

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

Spriteovi i karte

spr

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

Crta sprite n na x i y. w i h su veličine u pikselima, zadana vrijednost je 8. fx i fy okreću vodoravno i okomito kada su vrijednosti različite od nule. tex je kartica sprite-sheeta. rot je kut u okretima (0..1), isti smjer kao sin i cos. 0 ostavlja sprite nerotiranim.

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

sspr

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

Kopira pravokutnik sprite lista na zaslon, skaliran na dw prema dh. Zadane postavke zadržavaju veličinu izvora.

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

sget

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

Boja jednog teksela na sprite listu.

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

sset

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

Zapisuje jedan teksel na sprite sheet.

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

fget

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

Zastavice Spritea. fget(n) vraća bajt zastavice. fget(n, f) je true kada je postavljen bit zastavice f (0..7).

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

fset

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

Postavlja sprite zastavice. fset(n, v) zapisuje cijeli bajt. fset(n, f, v) postavlja jedan bit.

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

setspr

```
setspr(n)
```

Odabire aktivni sprite sheet. setspr() vraća trenutni sheet. Isti niz kao settex.

```
setspr(1)
```

settex

```
settex(n)
```

Alias od setspr.

```
settex(0)
```

map

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

Crta kartu. Ćelija cx, cy karte crta se na ekranu sx, sy. cw, ch od 0 znači da se cijela karta prihvaća i ignorira. tex je list skupa pločica.

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

mget

```
mget(cx, cy)
```

ID spritea u ćeliji karte cx, cy.

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

mset

```
mset(cx, cy, n)
```

Zapisuje sprite n u ćeliju karte cx, cy.

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

setmap

`setmap(n)`

Odabire mapu n. setmap() vraća trenutnu mapu. Mapa 0 je stvarna mapa.

```
setmap(1)
```

Memorija

peek

`peek(addr)`

Čita jedan bajt prozora od 64 KB. Prozor je zaslon uživo, zatim sprite pikseli, a zatim mapiranje ćelija kada zaslon stane u 65280 bajtova. Veći zasloni se straniče s poke(0x5F40, page). Adrese koje nisu mapirane vraćaju 0. Također se piše @addr.

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

poke

`poke(addr, ...)`

Zapisuje bajtove na adresu. Dodatni argumenti zapisuju sljedeće bajtove. Nemapirane adrese se ignoriraju. Također se piše $addr, val. Bockanje ekrana računa se kao pisanje piksela.

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

peek2

`peek2(addr)`

Neoznačena 16-bitna vrijednost, little-endian, na adresi.

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

poke2

`poke2(addr, val)`

Zapisuje neoznačenu 16-bitnu vrijednost, little-endian.

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

peek4

`peek4(addr)`

Neoznačena 32-bitna vrijednost, little-endian.

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

poke4

`poke4(addr, val)`

Zapisuje neoznačenu 32-bitnu vrijednost, little-endian.

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

memcpy

`memcpy(dst, src, len)`

Kopira len bajtova unutar istog prozora od 64 KB.

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

memset

`memset(dest, val, len)`

Ispunjava len bajtova počevši od odredišta s val.

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

Matematika

abs

`abs(x)`

Apsolutna vrijednost.

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

flr

`flr(x)`

Zaokružuje prema minus beskonačnosti.

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

ceil

`ceil(x)`

Zaokružuje prema plus beskonačno.

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

max

`max(a, b)`

Veća od dvije vrijednosti.

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

min

`min(a, b)`

Manja od dvije vrijednosti.

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

mid

`mid(a, b, c)`

Srednja od tri vrijednosti. Korisno kao stezaljka.

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

sgn

`sgn(x)`

Predznak x: -1, 0 ili 1.

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

sqrt

`sqrt(x)`

Kvadratni korijen. Negativni argument nije broj.

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

sin

`sin(x)`

Sinus. x je u naizmjeničnim okretajima: 0, 0,25, 0,5, 0,75 obilazi krug jednom. Ne radijani.

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

cos

```
cos(x)
```

Kosinus, pak, ima istu konvenciju kao i sin.

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

atan2

```
atan2(dx, dy)
```

Kut vektora dx, dy, naizmjenično od 0 do 1.

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

rnd

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

Slučajni broj u [0, x). Zadano x je 1, pa je rnd() 0..1. Za kocku, flr(rnd(6))+1.

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

srand

```
srand(seed)
```

Postavlja početni niz generatora slučajnih brojeva tako da se kasniji niz rundi može ponoviti.

```
srand(1)
```

band

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

Bitovni I. Također se piše a & b.

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

bor

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

Bitovni ILI. Također se piše a | b.

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

bxor

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

Bitovni XOR. Također se piše kao ^^ b.

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

bnot

```
bnot(a)
```

Bitno NE. Također se piše ~a.

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

shl

```
shl(a, n)
```

Pomakni lijevo. Također se piše << n.

shr

```
shr(a, n)
```

Aritmetički pomak udesno. Također se piše >> n.

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

lshr

```
lshr(a, n)
```

Logički pomak udesno. Također se piše >>> n.

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

rotl

```
rotl(a, n)
```

Rotiranje bitova ulijevo. Također se piše <<> n.

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

rotr

```
rotr(a, n)
```

Rotirajte bitove udesno. Također se piše >>< n.

Tablice i vrijednosti

add

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

Dodaje v tablici t. Ako je zadan i, umeće se na tom indeksu koji se temelji na 1.

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

del

```
del(t, v)
```

Uklanja prvi unos jednak v.

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

deli

```
deli(t, [i])
```

Uklanja indeks i i vraća tu vrijednost. Bez indeksa i, uklanja zadnji unos.

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

all

```
all(t)
```

Iterator preko dijela polja, po redu. Rupe zaustavljaju šetnju na način na koji završava polje bazirano na 1.

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

foreach

```
foreach(t, fn)
```

Poziva fn(v) za svaki element polja.

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

pairs

```
pairs(t)
```

Iterator nad svakim ključem i vrijednošću, uključujući imenovana polja.

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

ipairs

```
ipairs(t)
```

Iterator nad dijelom polja s indeksom koji se temelji na 1.

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

next

```
next(t, [k])
```

Vraća sljedeći ključ i vrijednost nakon k. next(t) počinje od početka. Koriste ga parovi.

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

count

```
count(t, [v])
```

Duljina dijela polja. Ako je zadano v, broji koliko je unosa jednako v.

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

pack

```
pack(...)
```

Pakira argumente u novu tablicu.

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

unpack

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

Vraća t[i] do t[j] kao odvojene vrijednosti. Zadane vrijednosti pokrivaju cijeli dio polja.

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

select

```
select(i, ...)
```

select("#", ...) vraća koliko argumenata slijedi. select(i, ...) vraća argumente od i nadalje.

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

assert

```
assert(c, [msg])
```

Zaustavlja se s greškom kada je c false ili nil. msg zamjenjuje zadani tekst.

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

pcall

```
pcall(f, ...)
```

Poziva f. U slučaju uspjeha vraća true i rezultate. U slučaju greške vraća false i tekst greške.

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

Žice

chr

`chr(n)`

Jedan znak iz bajt koda 0..255.

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

ord

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

Bajtni kodovi s od indeksa i do j. Indeksi se temelje na 1. Zadano je na prvom znaku.

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

sub

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

Podniz od i do j, baziran na 1. Izostavite j da biste uzeli rep.

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

split

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

Dijeli s u tablicu. sep je zadana vrijednost zarez. Broj sep je širina bloka. conv pretvara dijelove s tonom kada je vrijednost true.

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

tonum

`tonum(s, [base])`

Analizira broj. Baza može biti 2..36. Vraća nil kada tekst nije broj.

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

tostr

`tostr(v, [digits])`

Pretvara vrijednost u tekst. digits postavlja decimalna mjesta za brojeve.

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

Zvuk

sfx

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

Reproducira uzorak n i vraća kanal. Izostavite ch ili propustite -1 i odabire se prvi slobodni kanal. ch -2 zaustavlja uzorak n na svakom kanalu. n -1 zaustavlja kanal ili svaki kanal kada se izostavi ch. n -2 omogućuje završetak uzorka petlje umjesto ponavljanja. sfx() bez argumenata vraća -1.

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

sfxget

`sfxget(n, step)`

Vraća tablicu jednog koraka uzorka: [1] visina tona, [2] val, [3] glasnoća, [4] efekt. Visina tona je 0..63 (C-0 do D#5). Val je 0..15. Glasnoća je 0..7, a 0 je tiho. Efekt je 0..7.

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

sfxset

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

Zapisuje jedan korak uzorka. Svih šest argumenata je obavezno.

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

sfxhdr

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

Zaglavlje uzorka. sfxhdr(n) vraća tablicu zaglavlja. sfxhdr(n, key, val) postavlja jedno polje: speed, loop_in, loop_out, level, noise, crush, dtune, rverb, damn, swing, steps. Brojevi 0..10 odabiru ista polja tim redoslijedom. speed je TIC, otkucaji po koraku. Veće je sporije. 15 je 120 otkucaja u minuti. 16-bitni prihvaća 1..255. Uži sintisajzeri se pričvršćuju na svoju TIC listu.

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

sfxstat

```
sfxstat([ch])
```

ID uzorka svira na kanalu ch ili -1. Zadani kanal je 0.

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

note

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

Svira jednu notu bez uzorka. Visina tona 0..63, val 0..15, glasnoća 0..7, efekt 0..7. Otkucaji označavaju koliko dugo se drži, u istim otkucajima od 120 Hz kao i TIC. Izostavite otkucaje i traje 15, jedan zadani korak. Efekt se izvodi tijekom tog vremena, a zatim se kanal utišava.

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

music

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

Pokreće pjesmu n od vrha. music(-1) zaustavlja. Pozivanje music s ID-om koji se već reproducira ostavlja je u radu. music() vraća ID pjesme koja se reproducira ili -1. Fade je u milisekundama: pjesma se postepeno pojačava ili nestaje kada je n -1. Mask bitovi koji su postavljeni su kanali korišteni za pjesmu. 0 znači svih osam. Ostali kanali ostaju slobodni za zvučne efekte.

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

musicstat

```
musicstat()
```

Tablica pjesme koja se reproducira. song i row su ID pjesme i redak ili -1 kada se ništa ne reproducira. step i pos imaju osam brojeva. step[1] je kanal 0, step[8] je kanal 7. step je indeks note ili -1 kada je taj kanal tih. pos je koliko je daleko ta nota otišla, od 0 do 1. tick je koliko je otkucaja ovaj redak već reproduciran. played je koliko je redaka započelo od music(), uključujući i onaj koji se reproducira.

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

songget

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

ID uzorka pohranjen u ćeliji pjesme. ch je 0..7. 255 znači da je ćelija prazna. Ovo je aktivni poziv; tekst nagovještaja koda i dalje prikazuje stariji redoslijed argumenata.

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

songset

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

Zapisuje ID uzorka u ćeliju pjesme. Koristite 255 za brisanje ćelije.

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

stat

```
stat(n)
```

Samo zvuk. 16..23 je ID zvučnih efekata na kanalu n-16. 24..31 je glazbeni uzorak na kanalu n-24. Bilo koji drugi n vraća 0. Vrijeme kadrova i RAM su performanse i informacije, a ne statistike.

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

Tablice podataka

dget

```
dget(t, row, col)
```

Čita ćeliju podatkovne tablice t iz panela PODACI. Redak i stupac počinju od 0.

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

dset

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

Zapisuje u ćeliju. Koristite niz znakova za tekstualni stupac.

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

dstr

```
dstr(t, row, col)
```

Ćelija kao tekst. Brojevi se pretvaraju.

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

drows

`drows(t)`

Koliko redaka ima tablica t.

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

dcols

`dcols(t)`

Koliko stupaca ima tablica t.

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

Spremi i postignuća

save

`save(k, v)`

Pohranjuje v pod ključem k u datoteci za spremanje. v može biti broj, niz znakova ili tablica tih znakova. Datoteka je komprimirana.

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

load

`load(k)`

Čita ključ k. Vraća nil kada ključ nedostaje.

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

delsave

```
delsave(k)
```

Briše tipku k iz spremanja.

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

saveslot

```
saveslot([name])
```

Odabire datoteku za spremanje unutar mape autora. Ako nema imena, utor je naziv igre. Vraća 1 kada ta datoteka već postoji.

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

achieve

```
achieve(i)
```

Otključava postignuće i, od 0 do 15. Pohranjuje se lokalno. Steam verzija također postavlja ACH_i. Sesija kataloga izvještava o tome na računu igrača.

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

achieved

```
achieved(i)
```

1 kada je postignuće i već otključano.

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

achdef

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

Toast tekst za postignuće i, prikazuje se 15 sekundi od dna zaslona. spr je opcionalna pločica 8×8, tex njen list.

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

achclear

```
achclear(i)
```

Briše postignuće i u lokalnoj datoteci. Steam ACH_i koji je već postavljen ostaje postavljen.

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

Mreža

whoami

```
whoami()
```

Vaš ID igrača, 0..15, dok je sesija gotova. Izvan mreže je -1.

```
id = whoami()
```

nhost

```
nhost()
```

1 ako je ovo računalo domaćin, inače 0.

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

nplayers

`nplayers()`

Koliko živih ID-ova postoji u 0..15.

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

nlisten

`nlisten()`

Počinje hostirati UDP i vraća LAN adresu kao ip:port.

```
addr = nlisten()
```

njoin

`njoin(addr)`

Spaja ip ili ip:port.

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

nleave

`nleave()`

Napušta sesiju. Vršnjaci vide _mp_leave s argumentom "why" 0.

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

nsend

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

Šalje poruku s gubitkom. Naziv je događaj. Slijede do četiri dodatne vrijednosti. p od 0 do 15 šapuće tom peer-u; izostavite p za slanje svima.

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

ncall

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

Isti oblik kao nsend, ali se poruka pokušava ponovno poslati dok se ne potvrdi. Tri neuspjeha zaredom prekidaju sesiju.

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

nlan

```
nlan()
```

Trenutni LAN IP:port nakon nlisten-a.

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

nstun

```
nstun()
```

Vaša javna IP adresa: port sa STUN-a. Gost na drugoj mreži može pokušati pristupiti.

```
pub = nstun()
```

setturn

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

TURN rezervni način rada kada izravni put ne uspije. setturn() s praznim argumentima ga isključuje. Adresa je host:port.

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

nlist

```
nlist(game)
```

Tablica otvorenih sesija za taj naziv igre. Svaki unos ima adresu, igrače i ime.

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

nannounce

```
nannounce(game)
```

Otkucaj srca pa se ovaj domaćin pojavljuje na popisu za igru. Pozovite ga na tajmer tijekom domaćinstva.

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

nkb

```
nkb([on])
```

Polje za adresu. nkb(1) ga otvara, nkb(0) ga skriva, nkb() vam govori je li otvoreno. Tipke gore i dolje mijenjaju znak, 0–9 ili točku ili dvotočku. A ga zapisuje. X briše. Tipkovnica ga ne tipka.

```
nkb(1)
```

nkbstr

`nkbstr([s])`

Tekst u adresnom polju, a zatim znakovi gore i dolje se mijenjaju. nkbstr("") briše tekst.

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

_mp

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

Pišete ovu funkciju. Izvršava se kada stigne poruka. from je ID pošiljatelja. name je događaj. Slijede do četiri vrijednosti.

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

_mp_join

`_mp_join(p)`

Pišete ovu funkciju. Vršnjak je primio ID p. Također se pokreće za vas; whoami() je taj ID.

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

_mp_leave

`_mp_leave(p, why)`

Napišete ovu funkciju. Vršnjak p je otišao. Zašto je 0 normalan odlazak. Zašto je 1 izgubljena veza.

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

_mp_update

`_mp_update()`

Napišete ovu funkciju. Izvršava se svakih 15 _update okvira, oko 4 Hz. Dobro mjesto za nsend pozicije.

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

_mp_fail

`_mp_fail()`

Pišete ovu funkciju. Sesija je mrtva: host je nestao ili je ncall tri puta propao.

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

Korutine

cocreate

`cocreate(f)`

Stvara korutinu iz funkcije f. Ne izvršava se dok se ne završi coresume.

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

coresume

`coresume(c, ...)`

Izvršava korutinu c dok ne vrati vrijednost ili ne vrati vrijednost. U slučaju uspjeha: true i vraćene vrijednosti. U slučaju pogreške: false i greška.

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

costatus

```
costatus(c)
```

"suspendiran", "u pogonu" ili "mrtav".

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

yield

```
yield(...)
```

Vraća kontrolu onome tko je pozvao coresume. Vrijednosti postaju rezultati tog coresumea.

```
yield(1)
```

Mjera

info

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

Crta HUD uživo u _draw: fps (prosječno 1 sekunda), upd, drw, px, max, ram, peak, ramc i net kada je sesija završena. info() koristi kursor za ispis. info(col) postavlja samo boju. info(x, y) ili info(x, y, col) je postavlja. Brojevi koriste B, k ili m kako bi stali na mali zaslon. Sam poziv se ne broji u px.

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

showstats

```
showstats([on])
```

Isti HUD kao i info, automatski se crta na 0, 0 nakon _draw, ignorirajući kameru i isječak. showstats(0) ga skriva. showstats(1) ili showstats() ga prikazuje.

```
showstats(1)
```

perf

`perf(n)`

Jedan HUD broj, tako da ga možeš sam nacrtati. 0 fps, 1 upd, 2 draw, 3 px, 4 max (najteži kadar ove sekunde), 5 ram, 6 ramc (ram plus spakirani uložak za igru), 7 peak ram od početka uloška za igru.

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