PIX37 vadovas

Redaktorius, tada kodas. Knygos kopija neprisijungus pasiekiama svetainėje pix37.party/docs.

Redaktorius

„PIX37 DEV“ yra redaktorius. Čia piešiate, rašote ir klausotės žaidimo, tada eksportuojate žaidimo kasetę arba atskirą versiją. Ji taip pat gali žaisti žaidimo kasetę ir atidaryti PIX37.PARTY, kad atsisiųstumėte žaidimų kasetes ir naudotumėte katalogą. „PIX37 DEV EDUCATIONAL“ yra tas pats redaktorius, ir jis yra nemokamas. Jame nėra versijų, žaidimų kasečių eksportavimo ir prieigos prie PIX37.PARTY. „PIX37 Console“ yra grotuvas. Ji paleidžia `.p37` žaidimų kasetes, o jūs atidarote PIX37.PARTY, kad atsisiųstumėte žaidimų kasetes ir naršytumėte katalogą.

Projektai ir žaidimų kasetės

`.pix37` failas yra projektas: šaltinio skirtukai, спрайtų lapai, žemėlapiai, garsas ir duomenų lentelės. `.p37` failas yra žaidimo kasetė, kurią žaidėjas naudoja: glaudžiai supakuotas baitkodas.

Įrašyti failai ir pasiekimai yra autoriaus aplanke, šalia programėlės duomenų. „Windows“ sistemoje tai atrodo kaip `C:\\Users\\Ada\\AppData\\Roaming\\PIX37\\Mira\\Pong.p37s` (įrašytam failui) ir `C:\\Users\\Ada\\AppData\\Roaming\\PIX37\\Mira\\Pong.ach` (pasiekimams). „Linux“ sistemoje tie patys failai yra `~/.local/share/PIX37/Mira/Pong.p37s` ir `~/.local/share/PIX37/Mira/Pong.ach`. „Ada“ yra paskyra kompiuteryje. „Mira“ yra autoriaus vardas, o šis pavadinimas yra aplankas. „Pong“ yra išsaugojimo vieta, kurioje numatytoji reikšmė yra žaidimo pavadinimas. `save` įrašo `.p37s`. Pirmas atrakintas pasiekimas įrašo `.ach`. Jei failai jau yra, bėgikas juos nuskaito, kai paleidžiama žaidimo kasetė.

Kūrėjų meniu

Paspaudus „Escape“ atidaromas kūrėjo meniu. Paspaudus „Escape“ žaidžiant, grįžtama į redaktorių. Patvirtinkite paspausdami „Enter“ arba mygtuką A (pagal numatytuosius nustatymus – V). Paspaudus mygtuką „Back“ (pagal numatytuosius nustatymus – C).

Jei projektas neįkeltas, meniu yra toks: CREATE PROJECT (kurti projektą), LOAD PROJECT (įkelti projektą), BUILD (sukurti), LOAD Game cartridge (įkelti žaidimo kasetę), CONNECT TO PIX37.PARTY (prisijungti prie PIX37.PARTY), SETTINGS (nustatymai).

Įkėlus projektą, atlikite šiuos veiksmus: IŠSAUGOTI PROJEKTĄ, ĮKELTI PROJEKTĄ, SUKURTI PROJEKTĄ, IŠTEKLIAI, SUKURTI, ĮKELTI žaidimo kasetę, PRISIJUNGTI PRIE PIX37.PARTY, NUSTATYMAI.

Ctrl+S išsaugo atidarytą projektą. Ctrl+R jį paleidžia.

Sukurti projektą

Prieš pradedant kurti projektą, „CREATE PROJECT“ prašo žaidimo kasetės.

- **VAIZDO ĮRAŠAS** — 1 BITŲ (2 spalvos), 2 BITŲ (4), 4 BITŲ (16), 8 BITŲ (256).
- **GARSO NUSTATYMAS** — sintezatoriaus pavyzdžio gylis: 16, 8, 4, 2 arba 1 bitas.
- **EKRANAS** – iš anksto nustatyti dydžiai yra 64 × 64, 128 × 128, 160 × 90, 256 × 256 ir 320 × 180. Pasirinkus „PASIRINKTINIS dydis“, galite pasirinkti bet kokį dydį nuo 1 × 1 iki 512 × 512.
- **PALETĖS** – spalvų bankai. Pirmasis bankas prasideda nuo žemiau pateikto 16 spalvų rinkinio. Funkcija „ĮKRAUTI PALETĘ“ taiko integruotą rinkinį: PICO-8 16, PICO-8 32, Game Boy, 1 bitų, CGA, Sweetie-16 ir ilgesnį sąrašą iš „Lospec“ („Resurrect 64“, „Apollo“, „Endesga 32“ ir „64“, CC-29, SLSO8, Pear36, „Oil 6“, „Duel“, AAP-64, „Lost Century“ ir kt.). Taip pat galite įkelti paletės failą arba savo PNG failą ir paimti paletę iš to paveikslėlio.



`cls`, `pset`, `spr` ir kitų naudojami spalvų indeksai. Paletės redaktoriuje galite perdažyti bet kurį indeksą.

Penkios panelės

Redaktorius yra GAME, PAINT, CODE, SOUND ir DATA – tokia tvarka. Žemėlapis yra PAINT viduje. SFX ir dainos yra SOUND viduje. Spustelėkite skydelį, kad jame dirbtumėte. Kai CODE yra suaktyvintas, SOUND ir DATA ignoruoja raidžių klavišus, todėl rašant negrojama pianinu.



Visas išdėstymas, parašytas `1:1:4:1:1`. Kiekvienas skaičius nurodo, kiek eilutės užima tas skydelis: GAME, PAINT, CODE, SOUND, DATA. Mažas ekranas yra centre 128 × 128 langelyje. Ekranas, didesnis nei 128 pikseliai, praplatina GAME stulpelį.

NUSTATYMAI → PASAULINIS → IŠDĖSTYMAS keičia išdėstymą:

Kodas	Ką matai
`1:1:4:1:1`	Viskas. ŽAIDIMAS ir GARSAS sugrupuoti, DAŽAI ir DUOMENYS sugrupuoti, KODAS dešinėje.
`1:1:4:1:1` `2X`	Tas pats išdėstymas. KODAS piešiamas dvigubai tankiau, todėl telpa dvigubai daugiau teksto.
`1:0:1:0:1`	ŽAIDIMAS, DUOMENYS, KODAS.
`1:0:1:0:1` `2X`	Tas pats išdėstymas, tik su tankesniu KODU.
`0:1:1:1:1`	Nėra žaidimo peržiūros. Piešimas, kodas, garsas, duomenys.
`0:1:1:1:1` `2X`	Tas pats išdėstymas, tik su tankesniu KODU.
`0:1:4:1:2`	Studija. Platus KODAS, aukštesnė DATA juosta, DAŽAI ir GARSAS.
`0:1:4:1:2` `2X`	Tas pats išdėstymas, tik su tankesniu KODU.

`0:0:4:0:4`	Tik KODAS ir DUOMENYS, abu dideli.
`0:0:4:0:4 2X`	Tas pats išdėstymas, tik su tankesniu KODU.
`0:1:4:1:0`	DAŽAI, GARSAS ir platus KODAS.
`0:1:4:1:0 2X`	Tas pats išdėstymas, tik su tankesniu KODU.
`0:4:0:0:0`	Tik piešimas, 256 × 256, vienas lapo pikselis vienam drobės pikseliui.

Eilutėje `2X` išlaikomi tie patys skydeliai. Skydelis CODE piešiamas dvigubai didesne raiška ir sumažinamas iki to paties langelio, todėl tekstas, tabuliacijos ir tarpinė eilutė yra perpus mažesni, o ekrane telpa dvigubai daugiau šaltinio.

ŽAIDIMAS

Veikiančio žaidimo kasetės peržiūra. Norėdami paleisti peržiūrą, spustelėkite skydelį ŽAIDIMAS arba paspauskite Ctrl+R. NUSTATYMAI → ŽAIDIMŲ SKYDELYS – tai vieta, kur konfigūruojamas skydelis:

- **Peržiūra** – neaktyvi spalva, nepertraukiamas arba automatinis įkėlimas iš naujo redagavimo metu.
- **Žaisti naujame lange** – žaidimas atsidaro šalia redaktoriaus.
- **Ekranas** – peržiūra kaip RGB888 arba RGB565.
- **GAME 64** – 64×64 ekranas yra arba centruotas langelyje santykiu 1:1, arba mastelis padidintas 2 kartus, kad jį užpildytų.
- **Šešėliai** – CRT ir filtrai žaidime, atskirti nuo filtrų redaktoriuje.
- **Spalva** – skydelio fonas.

DAŽAI

Sprite lapai ir to paties skydelio žemėlapio pusėje – plytelių žemėlapiai. Lapai gali būti didesni nei ekranas. Sprite peržiūros dydžiai yra 8, 16, 32, 64 ir 128 (128 tik tada, kai lapas yra bent 128 pikselių). Norėdami jį priartinti, slinkite pelės ratuku virš спрайт peržiūros. Teptuko dydžiai yra 1, 2, 3 ir 4.

Raktas	Įrankis
B.	Pieštukas
M.	Stačiakampio pasirinkimas
Q	Lasso
S	Antspaudas

G	Kibiras
Aš	Pipetė
V.	Perkelti
L / Shift + L	Linija / kreivė
U / Shift+U	Tuščiaviduris stačiakampis / tuščiavidurė elipsė
F / Shift+F	Užpildytas stačiakampis / užpildyta elipsė
D	Dvejojimas
[ ]	Mažesnis / didesnis šepetys
Ctrl+Z / Ctrl+Y	Atšaukti / pakartoti. Ctrl+Shift+Z taip pat atlieka pakartotinį veiksmą.
Ctrl+C / Ctrl+X / Ctrl+V	Kopijuoti, iškirpti, įklijuoti
Ctrl+A	Pasirinkite drobę

Žemėlapyje naudojama ta pati idėja, tik su mažiau įrankių: B pieštukas, M stačiakampio pasirinkimas, S antspaudas, V perkėlimas ir Ctrl+Z, Ctrl+Y, Ctrl+C, Ctrl+V. Norėdami pakeisti žemėlapio tekstūros mastelį, laikykite nuspaudę Alt. Teptuko dydis žemėlapyje išlieka toks pat kaip ir piešimo įrankiai.

Kiekvienas спрайтas turi aštuonis vėliavėlės bitus. „Paint“ juos nustato спрайте; `fget` ir `fset` juos nuskaito žaidime.

NUSTATYMAI → PIŠKIMO REDAKTORIUS – tai vieta, kur konfigūruojamas tas skydelis:

- **Paletė** – kuris bankas ištraukiamas, taip pat įkeliamas ir išsaugomas.
- **Peržiūra** — RGB888 arba RGB565.
- **Teptuko žymeklis** – rodyklė, kuria piešiama.
- **Pikselių pritraukimas** – kaip teptukas pritraukia pikselius.
- **Didelis teptukas** – didelis teptukas atšoka iš viršutinio kairiojo kampo arba iš viso savo ploto.
- **Svogūno odelė** — kitų kadrų vaiduoklis ir į kurią pusę tas vaiduoklis žiūri.
- **Žemėlapio priartinimas** – tikri pikseliai arba laisvas priartinimas.
- **Spalvos** – dažų vartotojo sąsajos spalvos.

KODAS

Skirtukai yra atskiri projekto failai. Spustelėkite skirtuko pavadinimą, kad jį pervadintumėte; patvirtinkite paspausdami „Enter“. Keli skirtukai, kurių kiekvienas apibrėžia `_init`, `_update` arba `_draw`, yra sujungti skirtukų tvarka: pirmas skirtukas rodomas pirmas, paskutinis – paskutinis. Kodas, kuris nėra kabliuko viduje, tampa `_init` dalimi. Žaidimo kasetė be jokių kabliukų laikoma viena `_draw`.

Raktas	Veiksmas
Rodyklės	Perkelti žymeklį. Paspaudus „Shift“ išplečiamas pasirinkimas.
Ctrl+kairėn / dešinėn	Žodžiu.
Pradžia / Pabaiga, Ctrl + Pradžia / Pabaiga	Eilutės pradžia arba pabaiga arba tabuliacijos ženklas.
Skirtukas	Įterpkite du tarpus. Užbaigimo sąraše patvirtinkite pavadinimą.
Įeiti	Nauja eilutė, įtraukta kaip ir ankstesnė. Priima užbaigimą.
Ctrl+C / X / V	Kopijuoti, iškirpti, įklijuoti.
Ctrl+Z / Ctrl+Y	Atšaukti / pakartoti.
Ctrl+A	Pasirinkite skirtuką.
Ctrl+D	Dubliuoti liniją arba pasirinkimą žemyn.
Ctrl+/	Komentuokite arba panaikinkite komentarą naudodami `--`.
Ctrl+F	Rasti. F3 peršoka prie kito paspaudimo. Escape uždaro juostą.
Ctrl+G	Eikite į eilutės numerį.
Pabėgimas	Pirmiausia uždaromas paieškos, eiti į arba užuominų sąrašas. Antras „Escape“ klavišas palieka redaktorių ir įjungia kūrėjo meniu.

Įvedant funkcijos pavadinimą, rodomi užuominos. Paspaudus „Enter“ arba „Tab“, įterpiamas pavadinimas, o jei nustatymuose įjungta „COMPLETE (“), – `(`. NUSTATYMAI → KODO REDAKTORIUS:

- **Paieška** – Ctrl+F aktyviame skirtuke arba kiekviename skirtuke.
- **Aukštas tekstas** – apkirpkite glifus iki 5 pikselių arba rodykite visą glifą.
- **Eilutės dydis** — eilučių literalų šrifto dydis.
- **Komentarų šriftas** — 3×5 arba dabartinės vartotojo sąsajos kalbos „Fusion“ šriftas.
- **Užbaigta (** – ar užbaigimas įveda skliaustus.
- **Kodo užuominos** – įjungtos arba išjungtos.
- **Užuominų kalba** — meniu kalba arba anglų.
- **Tarpas tarp žymeklio** – papildomas poslinkis tuščiose eilutėse.
- **Latakas** – linijų numeriai.
- **Spalva** – fonas, komentarai, tabuliacijos, skaičiai, operatoriai, raktiniai žodžiai, eilutės, variklio funkcijos, jūsų funkcijos, kintamieji, skliaustai, žymeklis, pasirinkimas, klaidos eilutė, įspėjimas.

GARSAS

Skydelyje yra du skirtukai. SFX yra vienas šablonas: trumpa natų seka. SOUND yra daina: ji patalpina šiuos šablonus aštuoniuose kanaluose. TIC yra tai, kiek taktų laukia vienas žingsnis. Laikrodis yra 120 taktų per sekundę, o vienas žingsnis yra šešioliktoji nata, taigi 15 taktų yra 120 dūžių per minutę. Didesnis TIC yra lėtesnis šablonas. Du šablonai išlieka žingsnyje, kai vienas TIC dalija kitą: 15 ir 30 dalija kitą, 15 ir 8 – ne.

Fortepijonas groja, kai pele pasirenkama nata. Balti klavišai yra ZXCVBNM (CDEFGAB). Juodi klavišai yra SDGHJ (C# D# F# G# A#). Užveskite pelės žymeklį ant bet kurios reikšmės ir įveskite skaičių arba užveskite pelės žymeklį ant jos ir sukite ratuką. Kode tono aukščio skaičiai yra nuo 0 (C-0) iki 63 (D#5).

Viename pavyzdyje LOOP ir LEN yra tie patys du skaičiai. OUT virš 0 kartojasi nuo IN iki OUT, o pavadinimas rašomas LOOP. OUT ties 0 ir IN virš 0 yra LEN: tiek natų grojama nuo pradžios, tada pavyzdys baigiasi, o pavadinimas rašomas LEN. Abu ties 0 groja kiekvieną žingsnį.

Kiek sintezatoriaus galite redaguoti, priklauso nuo žaidimo kasetei pasirinkto bitų gylio. NUSTATYMAI → SFX REDAKTORIUS nustato tą gylį, ar nata automatiškai pereina į kitą lygį (išjungta, natų grandinė arba lauko grandinė), ar tarpo klavišas iš naujo paleidžia atkūrimą, ar perjungia paleidimą ir sustabdymą, ir redaktoriaus spalvas.

16 bitų

Kiekvienas žingsnis turi natą, garsumą, bangą ir efektą. Garsumas yra nuo 0 iki 7. 0 yra tylus.

Bangos:

- **0 trikampis**
- **1 pakreiptas pjūklas**
- **2 pjūklai**
- **3 kvadratai**
- **4 pulsas** – siauras pulsas
- **5 organų**
- **6 triukšmas**
- **7 fazeris**
- **8 PWM** — kvadratas, kurio plotis juda
- **9 impulsas 25%**
- **10 sinusų**
- **11 metalinis triukšmas**
- **12 FM varpas**
- **13 pliaukštelėjimų**
- **14 superpjūklų**
- **15 PWM (C64)**

Poveikis vienai natai:

- **0 nėra**

- **1 slinktis** – šiame žingsnyje aukštis ir garsumas pereina prie kitos natos
- **2 vibrato** — tonas svyruoja
- **3 lašas** – garsas krenta iki tylos, įprastas smūgis
- **4 išryškėjimas**
- **5 išnyksta**
- **6 arpedžai** – peržengia šią natą ir kitas tris, po vieną kas 8 tiksus. Esant 8 ar žemesniam tonui, kas 4 tiksus.
- **7 greiti arpedžai** – tos pačios keturios natos kas 4 tiksus arba kas 2, kai TIC yra 8 ar mažiau.

Šablono antraštė virš užrašų:

- **TIC** – dūžių skaičius per žingsnį nuo 1 iki 255. 15 yra 120 dūžių per minutę. Įveskite skaičių arba keiskite jį po vieną.
- **LOOP IN / OUT** – kai OUT yra didesnis nei 0, patternas kartojamas iš IN į OUT. Palikdami OUT ties 0, pavadinimas tampa LEN: IN nurodo grojamų natų skaičių, o 0 reiškia visą patterną.
- **TRIUKŠMAS** — visas modelis tampa triukšmu.
- **CRUSH** — kiekvienos bangos energingas variantas.
- **DTUNE** — 0 išjungta, 1 lengvas priedainis, 2 stiprus priedainis.
- **RVERB** — reverb: 0 išjungtas, 1 trumpas, 2 ilgas.
- **DAMPN** — žemo dažnio filtras: 0 išjungta, 1 lengva, 2 stipri.
- **LYGIS** — viso rašto garsumas, nuo 0 iki 7.
- **SUPIMAS** – šurmulys. Nelyginiai žingsniai palauk šiek tiek ilgiau, lygūs žingsniai šiek tiek trumpiau.
- **ŽINGSNIAI** — kiek žingsnių naudoja šablonas: 32, 24, 16 arba 12.

8 bitų

Žingsnis turi natą ir bangą. Keturios bangos yra kvadratinės, trikampės, pjūklo ir triukšmo. Antraštėje yra TIC, LOOP IN / OUT, FX ir LEVEL. TIC yra vienas iš šių: 1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 30, 40, 60. FX yra vienas efektas visam modeliui: 0 nėra, 1 vibrato, 2 arpeggio, 3 išblukinamas. LEVEL yra nuo 0 iki 3.

4 bitų

Žingsnis yra tik nata. Antraštėje yra TIC, LOOP (įjungta arba išjungta), WAVE (tos pačios keturios bangos) ir OCT (oktava, kurioje yra natos). TIC yra vienas iš šių: 4, 6, 8, 10, 12, 15, 20, 30.

2 bitų

Žingsnis yra tuščias, būgnas, smūgis arba sutriuškinimas. Antraštėje yra TIC ir BASE. TIC yra 8, 15, 16 arba 30. BASE nustato, kokiame aukštyje yra būgnas ir smūgis.

1 bitų

Žingsnis yra nata. TIC yra 8, 15, 16 arba 30 – tie patys keturi greičiai kaip ir 2 bitų.

Dainos

Skirtukas „GARSAS“ yra atidarytas kūrinys. Aštuoni stulpeliai yra kanalai. Langelis yra šablono ID. 255, nubrėžtas tuščiai, reiškia, kad kanalas toje eilutėje tylus. X stulpelis yra komanda: tęsti, sustabdyti, ciklo pradžia arba ciklo pabaiga. „Stop“ užbaigia kūrinį po eilutės. „Ciklo pabaiga“ grįžta į paskutinę ciklo pradžią.

Eilutė baigiasi, kai baigiasi kairiausias raštas, kurio LOOP OUT yra ties 0. Dešinėje pusėje esantys raštai tos eilutės nepailgina: ilgesnis yra nukerpamas, trumpesnis pradedamas iš naujo nuo pirmo žingsnio, kol baigiasi eilutė. Raštas, kurio LOOP OUT yra virš 0, kartojasi tarp LOOP IN ir LOOP OUT ir eilutės neužbaigia. Jei kiekvienas eilutės raštas kartojasi, eilutė lieka ten.

Spartieji klavišai

Raktas	Veiksmas
Fortepijono klavišai	Parašykite pastabą apie pele pasirinktą veiksmą.
Aukštyn / Žemyn	Pakeiskite reikšmę po žymekliu.
Erdvė	Paleiskite atvirą SFX arba atvirą dainą iš pirmos eilės. Pradėkite / sustabdykite arba paleiskite iš naujo SFX redaktoriaus nustatymuose.
Ctrl+Z / Ctrl+Y	Atšaukti / pakartoti.
Ctrl+C / Ctrl+X / Ctrl+V	Nukopijuokite, iškirpkite, įklijuokite pasirinkimą.

DUOMENYS

Pavadintos lentelės. Kiekviena lentelė yra eilutėmis ir stulpeliais. Iš kodo `dget`, `dset`, `dstr`, `drows` ir `dcols` naudoja šias lenteles. Eilučių ir stulpelių numeriai kode prasideda nuo 0.

Stulpelių tipai, ciklišKai išdėstyti antraštėje:

- **bool** — vienas bitas.
- **2** — du bitai.
- **4** — keturi bitai.
- **8** — baitas.
- **16** — 16 bitų, ID ir didesniems skaičiams.
- **spr** — спрайто ID.
- **str** — tekstas, nupieštas naudojant sistemos šriftą.
- **8 eilutė** ir **10 eilutė** – tas pats tekstas, tik nupieštas aukštesniu kampu.
- **f16** ir **f32** – pusės tikslumo slankiojantis ir pilnas slankiojantis.

Kai žymeklis užvestas ant stulpelio antraštės, paspaudus Ctrl+slinkties klavišą pakeičiamas to stulpelio tipas. Paprastas slinkties klavišas juda per stulpelius arba žemyn eilutėmis, atsižvelgiant į tai, ką pasirenka NUSTATYMAI → DUOMENŲ REDAKTORIUS. Ctrl pakeičia tą kryptį, todėl paspaudus Ctrl+slinkties klavišą ant antraštės, pakeičiamas tipas, kai ratukas nustatytas į stulpelius.

Rodyklių klavišai juda. „Enter“ redaguoja langelį. NUSTATYMAI → DUOMENŲ REDAKTORIUS:

- **Ratukas** – slinkties stulpeliai arba slinkties eilutės.
- **Spalvos** – tinklelis, antraštė ir pasirinkimas.

Turtas

Funkcija ASSETS importuoja ir eksportuoja atvertą dalį, o ne visą projektą. Sprite'ai naudoja programoje PAINT atidarytą спрайтą, esantį atidarytame lape. Žemėlapis yra atvertas žemėlapis. SFX yra atvertas raštas. Muzika yra atverta daina. Duomenys yra atverta lentelė. Išimtis yra kodas: kiekviena tabulatūra yra parašyta ir kiekviena tabulatūra yra nuskaitoma atgal.

Geras	Failas	Kas juda
Sprites	`.png`	Atviras спрaitas.
Žemėlapis	`.p37map`	Atviras žemėlapis.
SFX	`.p37sfx`	Atviras raštas.
Muzika	`.p37mus`	Atviras kūrinys. Šablonų numeriai lieka tokie patys. Jei reikia šių garsų, eksportuokite SFX atskirai.
Duomenys	`.p37dat`	Atviras stalas.
Kodas	`.lua`	Kiekvienas skirtukas.

Nustatymai

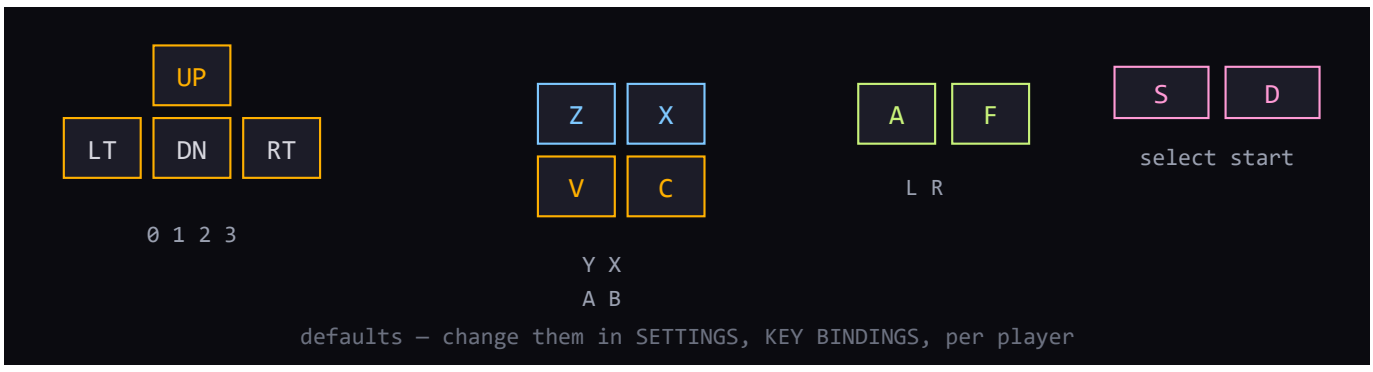
Šaknų sąrašas:

Puslapis	Ką jis saugo
PROFILIAI	Išsaugokite ir įkelkite kiekvieną nustatymą. Kurkite naują, pervardykite, perrašykite, ištrinkite. Nukopijuokite arba įklijuokite profilį kaip tekstą.
Meniu	Langas, klavišai, meniu spalvos, pasirinktinis fonas.
ŽAIDIMŲ PANELĖ	Peržiūra ir atkūrimo langas.

KODO REDAKTORIUS	Paryškinimai ir užuominos.
PIŠKŲ REDAKTORIUS	Teptukas, paletė, žemėlapio vaizdas.
SFX REDAKTORIUS	Sintezatoriaus ir natų įvedimas.
DUOMENŲ REDAKTORIUS	Ratų kryptis ir spalvos.
KALBA	Meniu ir užuominų kalba.
LICENCIJAS	Licencijų tekstai, įskaitant šriftą „Fusion Pixel“ ir SDL3.

PASAULINIS:

- **IŠDĖSTYMAS** — skydelio išdėstymas, aukščiau.
- **TŪRIS** — pagrindinis lygis, nuo 0 iki 10.
- **KLAVIŠŲ PRISIJUNGIMAS** — klavišai kiekvienam iš 8 žaidėjų. Paspauskite klavišą, kad jį užfiksuotumėte. Nurodymai, ABXYLR, pasirinkimas.
- **Pateikti** – praleisti nenaudojamus kadrus arba nupiešti kiekvieną kadrą.
- **Visas ekranas** – be rėmelių.
- **Atnaujinti** – lango FPS. Žingsnis į kairę ir į dešinę: 60, 90, 120 ir maks. Įveskite skaičių iki 1000. 0 reiškia, kad nėra ribos.
- **Vsync** — laukti, kol bus rodomas vaizdas tarp kadrų.
- **Mastelis** – sveikojo skaičiaus lango daugiklis.
- **Šešėliai** – CRT ir filtrai redaktoriuje, o ne žaidime.
- **Spalva** – meniu tekstas ir paryškinimas.
- **Fonas** – pasirenkamas PNG paveikslėlis, apkirptas iki 128 pikselių, esantis už meniu. Paspaudus „Išvalyti“, jis pašalinamas.
- **Redaktoriaus žymeklis** — rodyklės forma.
- **Pelė** – plaukiojantis žymeklis arba fiksuotas pikseliuose.
- **Naudotojo sąsajos greitis** – kaip greitai išlaikyta kryptis kartojasi sąraše.
- **„Explorer“** – failų sąrašai prasideda nuo kompiuterio šakninio katalogo arba exe failo aplanko.



Numatytoji klaviatūra. A yra patvirtinimo mygtukas meniu. B yra grįžimo mygtukas. Žaidime tai yra 4 ir 5 mygtukai. Pasirinkti (S) yra 10 mygtukas, o ne meniu spartusis klavišas.

Mygtukas	Vaidmuo	Numatytasis raktas
0	Kairė	Rodyklė kairėn
1	Dešinė	Rodyklė dešinėn
2	Aukštyn	Rodyklė aukštyn
3	Žemyn	Rodyklė žemyn
4	A, patvirtinkite	V.
5	B, nugara	C
6	X	X
7	Taip	Z
8	R	F
9	L	A
10	Pasirinkite	S
11	Pradėti	D

Sukurti

Pasirinkite platformą: „Windows“, „Linux“, „Mac“, „Android“, „iOS“ arba žiniatinklis.

- **Pavadinimas** – sukurto failo pavadinimas.
- **Autorius** – „Windows“ ir „Linux“ sistemose. Tai yra aplanko pavadinimas įrašams.
- **„Steam“ programėlės ID** – „Windows“ ir „Linux“ sistemose. Tuščias reiškia, kad nėra „Steamworks“. Užpildytas ID susieja `achieve(0)`…`achieve(15)` su „Steam“ pasiekimais ACH_0…ACH_15.
- **Id** – paketo identifikatorius, kurio prašo platforma.
- **Slėpti kodą** – darbalaukio ir mobiliųjų įrenginių perdangose. Tai sulėtina žaidimo kasetės `peek` veikimą. Tai nėra užraktas.
- **Visas ekranas** – eksportuotas grotuvas paleidžiamas viso ekrano režimu.
- **Šešėliai** – iki trijų sluoksnių, įkeptų į grotuvą.
- **Atsisiųsti .p37** – tik žiniatinklyje. Pauzės meniu rodomas atsisiuntimo įrašas.
- **Piktograma** – plytelė iš спрайтų lapo: lapas, dydis, x, y. Arba apkirptas PNG viršelis.
- **Eksportuoti** – pasirinkite aplanką. Meniu praneša „KURIMAS IŠSAUGOTAS“ arba „KURIMAS NEPAVYKO“.

Sukurkite versiją su įjungtu internetu. Įžanginis pranešimas gali anonimiškai paklausti serverio, ar žinoma šios versijos maišos vertė (hash). Šis patikrinimas atliekamas dėl dviejų priežasčių. Jis apsaugo mane, PIX37 autorių, nuo įrankio kopijų, kurios nebuvo mano, kad galėčiau jas atiduoti. Jis taip pat apsaugo asmenį, sukūrusį žaidimą. Kito žmogaus žaidimo paėmimas ir jo įtraukimas į sąrašą kaip savo tampa įprastu reiškiniu telefonų rinkoje, pvz., „Play" („Android") ar „iOS". Žaidimas vis tiek prasideda. Jei įrenginys neprisijungęs prie interneto, įžangoje sakoma, kad žaidimas neprisijungęs. Ši eilutė taip pat yra užuomina žaidėjui, kad kopija gali būti ne iš teisėto šaltinio. Tai nedidelis patikrinimas, todėl jei jūsų žaidimą pavogė ir pardavė kažkas kitas be jūsų sutikimo, galite pateikti šį ieškinį parduotuvėje, pvz., „Play" („Android"), „Steam" ar „itch.io". Serveris šalia maišos vertės išsaugo laiką, kada maiša buvo pirmą kartą užregistruota, ir vėliau jo nekeičia. Ankstesnis laikas yra originali versija. Po to užregistruota maiša yra vėlesnė kopija.

Noriu, kad šis patikrinimas apsaugotų žaidimų kūrėjus. Variklis jau ir taip atlieka didelę dalį darbo už juos, ir tai yra vienas iš tokių dalykų. Suprantu, kad tai gali būti prieštaringa. Jei bendruomenė nuspręs, kad geriau atsisakyti šios mikro-DRM sistemos, aš tai padarysiu.

Sena žaidimo kasetė iš ankstesnio redaktoriaus gali atsisakyti paleisti pakeitus kalbą. Atidarykite `.pix37` projektą šiame redaktoriuje ir eksportuokite naują `.p37`.

Prisijunkite prie PIX37.PARTY

PRISIJUNGTI PRIE PIX37.PARTY galima programėlės viduje. Naršyklėje nėra prisijungimo puslapio. Prisijungiate naudodami „Steam" sesijos bilietą arba „itch.io", naudodami planšetinį kompiuterį ir pelę. Paslauga rodo unikalų vardą – nuo 2 iki 16 simbolių. Licencijos raktas ir „Steam" ID kitiems žaidėjams nerodomi.

Po paslaugų teikimo sąlygų centre yra Mėgstamiausi, Katalogas, Reitingas, Žaidėjai ir jūsų profilis.

Kūrėjo licencija yra redaktorius. Jūs sukuriate žaidimą, jį įkeliate ir paliekate žvaigždutes. Žaidėjo licencija yra žaidėjas. Širdelės gaunamos iš žmonių, žaidžiančių „Bėgiku".

- **Katalogas** – naršykite žaidimus. Filtrai apima žanrą (iki trijų), ekrano dydį, žaidimo kasetės dydį ir tai, ar įkeliamame faile yra `.pix37` projektas.
- **Mėgstamiausi** – jūsų pažymėti žaidimai.
- **Įkėlimas** – reikalinga kūrėjo licencija. Jūs siunčiate `.p37` ir galite pridėti `.pix37` projektą. `.p37` yra supakuotas baitinis kodas. Jį sunku skaityti ir jis gali būti tyčia užmaskuotas. `.pix37` yra projekto failas, su kuriuo jūs iš tikrųjų dirbate: šaltinis, paveikslėliai ir garsas, forma, kurią kitas asmuo gali atidaryti ir iš kurios mokytis. Jį siunčiant kiti kūrėjai gali studijuoti žaidimą, o ne suspaustą žaidimo kasetę. Kartu pateikiami pavadinimas, autorius, žanrai, viršelis, piktograma ir pasiekimų pavadinimai. Išsiųstas RAM skaičius yra didžiausias žaidimo kasetės pasiektas skaičius.
- **Komentarai ir pasiekimai** – žaidimo puslapyje. Pasiekimo atrakinimas prisijungus saugomas tiek toje paskyroje, tiek įrenginyje.
- **Reitingas** – kūrėjai, surūšiuoti pagal žvaigždutes arba širdeles. Kiekvienoje eilutėje rodomi abu skaičiai. Atidarykite pavadinimą, kad pamatytumėte to autoriaus žaidimus, surūšiuotus pagal širdeles.
- **Žaidėjai** – paskyros, surūšiuotos pagal atrakintus pasiekimus.

Kodas

Kalba yra mažas „Lua“ stiliaus scenarijus. Ši dalis pradedama nuo tuščio projekto, mokoma kalbos ir pateikiamas trumpas kiekvienos funkcijos pavyzdys.

Pirmoji programa

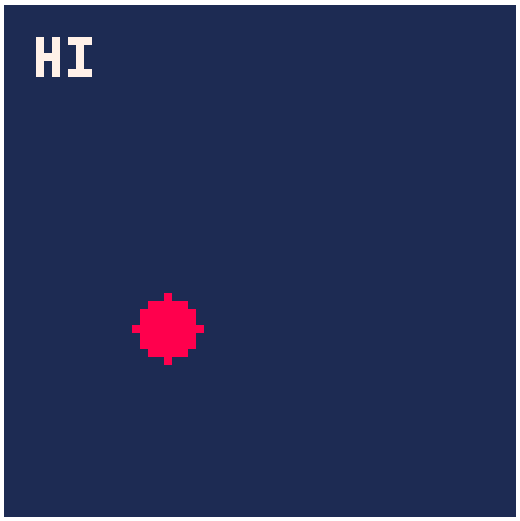
Sukurkite 64 × 64 dydžio, 4 bitų spalvų projektą. Atidarykite CODE. Pakeiskite tabuliacijos ženklą į:

```
_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 paleidžia. Rodyklių klavišai judina apskritimą. Escape sustabdo žaidimą ir grįžta į redaktorių.



> Tie patys trys kvietimai: `cls(1)` nudažo tamsiai mėlyną ekraną, `print` nupiešia HI, o `circfill` nupiešia rožinį apskritimą.

Kas bėga ir kada

- `_init` paleidžiamas vieną kartą.
- `_update` veikia 60 Hz dažniu. Čia įrenkite mechanizmą ir mygtukus.
- `_draw` piešia. Be jokio kadrų dažnio argumento jis piešia tokiu greičiu, kokiu gali kompiuteris ir monitorius. `_draw(30)` apriboja piešimą iki 30 kadrų per sekundę ir palieka logiką ties 60.
- `_set_fps(n)` vėliau pakeičia tą ribą. 0 ją išvalo.

Pavadinimuose didžiosios ir mažosios raidės neskiriamos; redaktorius juos saugo mažosiomis raidėmis. `function` ir `fun` yra tas pats raktinis žodis.

Kalba

Programavimo pagrindai

Programa yra žaidimo kasetės vykdomų eilučių sąrašas. Kiekvienoje antraštėje pirmiausia pateikiama eilutės forma, o tada pateikiamas nedidelis pavyzdys. Eilutė, prasidedanti `--`, yra komentaras: pastaba jums. Žaidimas ją praleidžia.

Kintamieji

Kintamasis yra pavadinimas, kuriame saugoma reikšmė. Jūs pasirenkate pavadinimą. Vėlesnės eilutės gali jį nuskaityti ir išsaugoti naują reikšmę tuo pačiu pavadinimu.

Forma yra `name = value`. `local name = value` šį pavadinimą išsaugo dabartinio bloko viduje, todėl likusi žaidimo kasetės dalis jo nemato. Be `local` pavadinimas yra globalus ir jį gali matyti kiekvienas skirtukas.

```
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` reiškia, kad reikšmės nėra. `true` ir `false` yra „taip“ ir „ne“. Pavadinimas, kurio niekada nenustatėte, skamba taip: `nil`.

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

Stalai

Lentelėje vienu pavadinimu saugoma daug reikšmių. Sąraše naudojami skaičiai. Pirmasis elementas yra 1, o ne 0. Įraše naudojami pavadinimai. Lentelėje gali būti kita lentelė.

Sąrašo forma: `{ value, value }`. Įrašo forma: `{ name = value, name = value }`. Įrašų sąrašas yra lentelė, kurios elementai yra lentelės.

```
-- 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
```

Būklė ir kūnas

Sąlyga yra klausimas su atsakymu „taip“ arba „ne“, pvz., `hp > 0` arba `btnd(4)`. Kūnas yra eilučių grupė, kuri brėžiama įvedus bloką. `if` ir kiekvienas ciklas naudoja tą pačią formą: sąlyga, tada kūnas, tada `end`. `repeat` užsidaro `until` vietoj `end`.

Jei

`if` vykdo savo pagrindinę komandą tik tada, kai sąlyga yra teisinga. `elseif` bando kitą sąlygą. `else` paleidžiama, kai visos sąlygos yra klaidingos.

Forma: `if condition then`, tada korpusas, tada `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
```

Ciklai

Ciklas kartoja turinį. `while` pirmiausia patikrina sąlygą ir tęsia skaičiavimą, kol ji išlieka teisinga. `repeat` paleidžia turinį, tada patikrina, kad turinys visada būtų paleidžiamas bent kartą. `for` skaičiuoja. Įtraukiamas paskutinis skaičius. Be trečio skaičiaus žingsnis yra 1, taigi, jei pradžia jau yra po pabaigos, turinys neskaičiuojamas. Skaičiuojant atgal reikia neigiamo žingsnio.

```
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
```

Vaikščiojimas prie stalo

`all` aplanko kiekvieną sąrašo elementą, pradedant nuo pirmojo. `pairs` aplanko kiekvieną pavadinimą ir reikšmę, saugomą po tuo pavadinimu.

```
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
```

Funkcijos

Funkcija yra pavadintas receptas. Forma yra `function name(arguments)`, tada kūnas ir tada `end`. Ją iškviečiate įrašydami pavadinimą ir skliaustus. `return` siunčia atgal reikšmę ir gali siųsti daugiau nei vieną. Argumentai yra tos funkcijos lokalūs variantai.

```
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` yra funkcija, kuri gali matyti aplinkinius lokalius. Tai yra uždarumas. Lentelėje gali būti toks, ir jūs jį iškviečiate tašku.

```
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
```

Taip, ne ir pasirinkimas

`not` apverčia sąlygą. `and` yra teisinga tik tada, kai abi pusės yra teisingos. `or` yra teisinga, kai kuri nors pusė yra teisinga. `a and b or c` pasirenka `b`, kai `a` yra teisinga, kitu atveju – `c`. `cond ? a : b` yra ta pati idėja vienoje eilutėje: `a`, kai sąlyga teisinga, `b`, kai ji ne.

```
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 ir tekstas

`+ - * /` yra sudėtis, atimtis, daugyba ir dalyba. `//` dalija ir išsaugo sveikąjį skaičių. `%` yra liekana. `^` yra laipsnis. `..` sujungia du tekstus. `#t` nurodo, kiek elementų yra sąraše. Palyginimai yra `==` (tas pats), `~=` arba `!=` (skirtingi) ir `< > <= >=`. `x += 1` prie x prideda 1. Ta pati forma tinka ir `-= *= /= ^= %=`.

```
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
```

Kur gyvena vietinis gyventojas

`local` veikia iki `end`, kuris uždaro jo bloką. `if` / `end` yra atskiras blokas. Galite paminėti lokalią reikšmę virš eilutės, kuri ją nustato, bet iki tos eilutės pabaigos reikšmė yra `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
```

Šokinėjimas

`::label::` žymi eilutę. `goto label` peršoka ten, bet tik tos pačios funkcijos viduje.

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

Atmintis, trumpai tariant

`peek` ir `poke` nagrinėja gyvus pikselius per vieną 16 bitų adresą. Jie nesaugo antros žaidimo kasetės kopijos. Adresai nuo `0x5F00` iki `0x5FFF` yra registrai: kamera, iškarpa (tik skaitoma), piešimo spalva, užpildymo raštas, trys palečių perrašymai ir puslapių numeriai. Baitai abiejose tos skylės pusėse, jų yra 65280, yra pikselių langas.

Kai ekranas telpa į tuos 65280 baitų, 64×64 ir 128×128 telpa, o 256×256 – ne, ekranas pradedamas nuo 0 adreso, eilutė po eilutės. Lieka `0x5F41` pasirinktas спрайтų lapas (0 yra 0 lapas). Jei tas lapas yra didesnis už likusią vietą, `0x5F42` pasirenka puslapį ir žemėlapio lange nėra. Jei lapas telpa, gale yra žemėlapio langeliai iš `0x5F43` pasirinkto žemėlapio, po vieną arba du baitus, o `0x5F44` puslapiuoja žemėlapį, kai jis netelpa gale.

Kai ekranas netelpa, visas langas yra vienas 65280 pikselių puslapis. `poke(0x5F40, n)` perjungia ekrano puslapį. `poke(0x5F41, n)`, kai n yra ne 0, rodo lapą n, o lapas 0 šiame režime neturi vaizdo. `poke(0x5F43, n)`, kai n yra ne 0, rodo žemėlapį n, ir tai laimi prieš lapą. 0 žemėlapis šiame režime yra tik `mget` ir `mset`.

`peek`, kuris nenukrenta ant pikselio ar registro, grąžina 0. `poke` čia nieko nedaro. Šiame lange nėra спрайтų vėliavėlių, SFX ir DATA lentelių. Naudokite `fget`, `sfxget` ir `dget`.

Praktinis panaudojimas

Kiekviena funkcija pateikiama šio puslapio apačioje, pateikiant praktinius PIX37 funkcijų panaudojimo pavyzdžius. Nuorodos po kiekvienu pavyzdžiu leidžia daugiau sužinoti apie funkcijas.



Sklandus judėjimas

128×128 žaidimo kasetė, matoma iš viršaus. Pirmasis спрайтas jau nupieštas. Rodyklės greitėja bet kuria kryptimi, o trintis jus sulėtina. спрайтas apsivynioja: pusė, paliekanti apačią, nupiešiama viršuje, ir tas pats su kitomis briaunomis.

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



Tikrinama, ar kitas žingsnis užblokuotas

64×64 žaidimo kasetė. Grindys ir dešinioji siena jau yra žemėlapyje, pažymėtos 3-iu спрайtu, ir tam спрайтui nustatyta 0 vėliavėlė. Rodyklės judina 1-ąjį спрайтą. Žodis viršuje nurodo „užblokuotas“, kai kitas žingsnis persidengtų su ištisine plytele, ir „laisvas“, kai nesutaptų.

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



Valymas apverčiant

Paspauskite A. Išvalymo kodą braižo flip(), po vieną juodą eilutę vienu metu, o ne _draw() viduje. _draw vykdymo metu netrukdo, todėl kiekviena eilutė lieka ekrane.

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



Funkcijų lentelė (klasė)

Pirmasis спрайtas jau nupieštas. „make_monster“ veikia kaip klasė. Jis grąžina lentelę, o „update“ ir „draw“ yra metodai, kurie išsaugo pabaisos x ir y reikšmes. Skaičius viršuje rodo, kiek atstumo ji nuėjo.

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



Keturi būdai, kaip perskaityti mygtuką

A yra 4 mygtukas. Laikykite jį nuspaudę, ir įkrova kyla. Atleiskite, ir įkrova grįžta į 0. Laikant mygtuką, aukštyn ir žemyn kartojasi judesiai, todėl meniu eilutė juda toliau. btnd yra pirmasis paspaudimo kadras, kurį šis žaidimo kasetė naudoja tik kaip to laikymo pradžią.

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



Šaudykite su pele

Pirmasis спрайtas jau nupieštas viduryje. Baltas taškas yra žymeklis. Kairysis pelės mygtuko paspaudimas nukreipia pikselį į jį. Ratukas keičia greitį, kuris atspausdintas kampe.

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



Įveskite vardą ir patvirtinkite jį

Norėdami atidaryti lauką, paspauskite A. Aukštyn ir žemyn keičia mirksinčią raidę. A ją įrašo, X ištrina, Y yra tarpas, o B patvirtina. Tada eilutėje rodomas jūsų vardas.

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



Grotuvo kamera

128×128 žaidimo kasetė. Sprite 1 prasideda plytelių grindų viduryje. Rodyklės juda. Kamera laiko спрайtą centre. x ir y kampe neslenka, nes camera() be argumentų grąžina vaizdą HUD.

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



Kelios vizualinio romano eilutės

Eilutės yra DATA formatu, anglų ir lenkų kalbomis. A tęsiasi arba pasirinkimai lieka, kai ekrane rodomas pasirinkimas. B pasirenka, palieka pasirinkimą, kitu atveju pakeičia kalbą. X pakeičia kalbą, kai rodomas pasirinkimas. Dialogo lange naudojamas 8px šriftas, kad būtų rodomos lenkiškos raidės.

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



Dažykite žvilgsniu ir bakstelėjimu

64×64 žaidimo kasetė, taigi adresas 0 yra viršutinis kairysis pikselis, o kita eilutė prasideda nuo 64. Žaidimo kasetė įrašo pikselius, nukopijuoja eilutę, išvalo eilutę ir atspausdina nuskaitytas reikšmes.

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



Sąrašas, kurį galite įtraukti

A prideda skaičių iki 8. B pašalina paskutinį. X pašalina pirmąjį. Sąrašas atspausdinamas ekrano apačioje.

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



Išardytas vardas

Žaidimo kasetė padalina ada,12,blue ir atspausdina pavadinimą, jo pirmąją raidę – 12 – ir žodį blue.

`cls split sub chr ord tonum tostr print _draw`



Sugrokite natą, tada ją pakeiskite

0 modelis yra vienos natos ilgio ir kiekvienas pakartojimas yra tokio pat garsumo. A groja jį dar kartą, kai ta nata baigiasi. B pakelia aukštį ir groja tą natą.

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



Į žaidimo kasetę įrašyta daina

0 eilutėje kartojasi du patternai, po vieną kiekviename iš pirmųjų dviejų kanalų. Eilutė lieka 0. ch0 ir ch1 yra nata, įjungta kiekvienam kanalui. Žaidimo kasetėje redaguojant patterną, LEN ir LOOP, galima pamatyti, ką šie valdikliai daro muzikos skydelyje.

`songget songset music musicstat cls print _init _draw`



Trys monstrai DATA

DUOMENŲ skydelyje jau yra pavadinimo stulpelis ir gyvybės taškų stulpelis su gleivėmis, šikšnosparniu ir akimi. Raide A kenkia gleivėms. Skaičiai dešinėje yra jos gyvybės taškai.

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



Prisiminkite rezultatą

A prideda tašką ir jį išsaugo. Išeina ir vėl atidaro žaidimo kasetę: skaičius vis dar yra. B jį pamiršta.

`load btnd save delsave cls print _init _update _draw`




Antras skyrius kitame faile

Dvi žaidimo kasetės dalijasi išsaugojimo vietos užduotimi. 1 dalis: A žymi žaidimo pabaigą. 2 dalis, atidaryta pirmiausia, nurodo užbaigti 1 dalį. Po to atidaroma 2 dalis, o B ištrina išsaugojimą, kad 1 dalį būtų galima žaisti dar kartą.

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



Vienas pasiekimas

8-asis спрaitas jau ištrauktas. Dešimt kartų paspauskite A. Skrebučio simbolis įvardija pasiekimą, o 8-asis спрaitas lieka, nes jis atrakintas. B išvalo tą pasiekimą vietiniame faile ir atstato skaičių.

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



Kambarys, į kurį galite įeiti

256×256 žaidimo kasetė. Baltas taškas yra žymeklis. Meniu yra „host“, „join“ arba serverių sąrašas. Šeimininkas atidaro kambarį pavadinimu „walk“ ir pradeda žaidimą. „Join“ slenka adresu (skaitmenimis, tašku, dvitaškiu), o B prisijungia. Serverių sąrašas juda aukštyn ir žemyn, o A prisijungia prie eilutės. Žaidime rodyklių „walk“ спрайtas 1. Viena moneta yra dalijama: šeimininkas ją perkelia, o pakėlus ją, to žaidėjo rezultatas padidėja 1 ir sukuriama kita. Kiekvienas спрайtas, įskaitant jūsų, rodo tą rezultatą šalia. Kampe rodoma, kiek žaidėjų yra, LAN adresas ir šeimininko viešasis adresas. B išeina: jūs ir kiti grįžtate į meniu, o kitas šeimininkas pradeda naują kambarį su 0 taškų.

```
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
```



Dialogas, kuris laukia

A rodo kitą eilutę. Funkcija sustoja ties kiekvienu išėjimo tašku, kol vėl paspaudžiate A.

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



Palaukite pusę sekundės

Žodis maždaug po pusės sekundės pasikeičia iš ... į hi. Ekranas toliau piešia, kol korutina pristabdyta.

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



Apskritimai, kurie atvyksta po vieną

Iš viršaus, dviejų kadrų atstumu vienas nuo kito, laša penki apskritimai ir tada krenta.

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



Du kadrai iš laikrodžio

1 ir 2 спрайтai yra du to paties simbolio kadrai. Vaizdas keičiasi keturis kartus per sekundę.

```
cls flr time spr _draw
```

Kiekviena funkcija

Argumentai laužtiniuose skliaustuose yra neprivalomi. Lentelių ir eilučių indeksai prasideda nuo 1, išskyrus DATA eilutes ir stulpelius, kurie prasideda nuo 0. Spalvos yra palečių indeksai.

Indeksas

_piešti
_init
_mp
_mp_fail
_mp_join
_mp_leave
_mp_update
_set_fps
_atnaujinimas
abs
achclear
achdef
pasiekti
pasiekti
pridėti
visi
tvirtinti
atanas2
juosta
bnot
bor
btn
btnd
btnp
btnu
bxor
fotoaparatas
lubos
chr
circ
apskritimo užpildymas
spustelėkite
clickm
paspaudimas
klipas
kls
kartu sukurti
spalva
Coresume
cos
costatus
skaičiuoti
žymeklį
dcols
del
delikatesas
delsave
dget
snaudžia
dset
dstr
fget
užpildas
apversti
flr
foreach
fset
informacija
ipairs
atjungimas
keyon
keystat
keystr
linija
apkrova
žurnalas
lshr
žemėlapis
maks
memcpy
memset
mget
vidurio
min
pelė
mset
muzika
musicstat
paskelbti
skambinti
kitas
nhost
njoin
nkb
nkbstr
nlan
npalikti

nlist
nklausyk
pastaba
nplayers
nsend
nstun
įsakymas
ovalios
ovalus užpildas
pakuotė
porų
draugas
palrgb
palt
pcall
žvilgtelėti
žvilgtelėti2
žvilgtelėti4
perf
pget
kišti
kišti2
kišti4
spausdinti
spausdinti 10
spausdinti8
pset
ties
ištaisymas
rnd
rotl
rotr
teisingai
teisingas užpildymas
išsaugoti
taupymo vieta
ekranas
slinkti
pasirinkite
setmap
rinkinių paletė
setspr
setex
nusistatymas
sfx
sfxget
sfxhdr
sfxset
sfxstat
sget
sgn
shl
šou statistika
Shr
nuodėmė
dainelė
dainų rinkinys
padalintas
spr
kv
srand
sset
sspr
stat
sub
laiko
tline
tonum
tostr
išpakuoti
whoami
derlius

Gyvavimo ciklas

_init

`_init() ... end`

Paleidžiamas vieną kartą, kai paleidžiama žaidimo kasetė. Kodas, esantis už bet kurio kabliuko ribų, taip pat įdedamas į _init.

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

_update

`_update() ... end`

Žaidimo logika. Iškviečiama 60 Hz dažniu, net kai _draw yra apribotas arba serveris lėtesnis. Skaitykite mygtukus čia.

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

_draw

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

Nubraižo vieną kadrą. Be argumento arba su 0, jis paleidžiamas taip dažnai, kaip pateikia pagrindinis kompiuteris. Perduokite n (1..999), kad apribotumėte piešimą n kadrų per sekundę greičiu. _update išlieka 60 Hz.

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

_set_fps

`_set_fps(n)`

Pakeičia _draw ribą žaidimo metu. 0 reiškia neribotą. n yra apribotas iki 0..999.

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

flip

```
flip()
```

Pateikia ekraną ir užleidžia vietą iki kito kadro. Naudokite tik tuo atveju, jei piešiate savo kadrų ciklą. Neužsiimkite – palaukite po jo.

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

time

```
time()
```

Sekundės nuo _init.

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

Įvestis

btn

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

1, kol laikomas mygtukas b, kitu atveju 0. p yra žaidėjas, 0..7. Be argumentų grąžina žaidėjo 0 bitų kaukę. Mygtukai: 0 kairėn, 1 dešinėn, 2 aukštyn, 3 žemyn, 4 A, 5 B, 6 X, 7 Y, 8 R, 9 L, 10 pasirinkti, 11 pradėti.

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

btnd

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

Tiesa tik _update rėmelyje, kai nuspaustas mygtukas b.

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

btnp

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

Paspaudus, tada vėl laikant: 30 taktų uždelsimas, po to kas 8 taktus. Be mygtuko grąžina kaukę.

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

btnu

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

Tiesa, mygtukas atleidžiamas tik ant rėmo.

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

mouse

```
mouse()
```

Grąžina pelės x ir y koordinates žaidimo pikseliuose.

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

clickl

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

Kairysis pelės mygtukas. Laikoma nuspausta „clickl()“. „clickl(1)“ yra spaudimo kraštas.

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

clickr

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

Dešinysis pelės mygtukas. Tas pats vieno argumento kraštas kaip ir spustelėjus.

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

clickm

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

Vidurinis mygtukas arba slinkties ratuko spustelėjimas.

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

scroll

```
scroll()
```

Pelės ratuko judėjimas šiame rėmelyje. Teigiamas ir neigiamas krūviai priklauso nuo ratuko krypties.

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

keyon

```
keyon()
```

Atidaro teksto lauką, valdomą mygtukais. Aukštyn ir žemyn keičia raides nuo A iki Z. A įrašo tą raidę. X ištrina. Y yra tarpas. B patvirtina ir uždaro lauką. Klaviatūra į jį nerašo.

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

keyoff

`keyoff()`

Paslepia teksto lauką.

```
keyoff()
```

keystr

`keystr([s])`

Tekstas lauke, tada 1 rėmelyje B patvirtina, tada raidės viršuje ir apačioje keičiasi. Raidė tuščia, kai laukas užsidaro. keystr("") išvalo tekstą.

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

keystat

`keystat()`

1, kai teksto laukas atidarytas, kitu atveju – 0.

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

Piešimas

screen

`screen()`

Dabartinio žaidimo kasetės ekrano dydis pikseliais. Grąžina plotį, tada aukštį. Tai žaidimo skiriamoji geba, o ne redaktoriaus lango skiriamoji geba.

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

cls

`cls([col])`

Išvalo ekraną. Numatytoji spalva yra 0.

```
cls(1)
```

color

`color(col)`

Nustato piešimo spalvą, naudojamą, kai vėlesniame iškvietime praleidžiama „col“. Be argumento grąžina dabartinę spalvą.

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

camera

`camera([x, y])`

Poslinkis nustatomas po kiekvieno draw iškvietimo. camera() atstato reikšmę į 0, 0.

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

clip

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

Apriboja piešimą stačiakampiu. Ne nulinis penktas argumentas kerta dabartinį įrašą. clip() atkuria visą ekraną. Įrašo registrų peržiūra yra skirta tik skaitymui.

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

cursor

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

Nustato, kur eis kitas spausdintas objektas be koordinačių, ir pasirinktinai spalvą. cursor() grąžina x, y.

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

pset

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

Nustato vieną ekrano pikselį. Įskaičiuojamas į pikselių statistiką.

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

pget

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

Ekrano pikselio spalva x ir y taškuose.

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

print

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

Piešia 3×5 tekstą (UTF-8, apkirptas iš „Fusion“ šrifto). Be x ir y spausdina ties žymekliu ir pereina į kitą eilutę.

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

log

`log(...)`

Prideda argumentus, atskirtus tarpais, prie teksto buferio. „pix37_run“ išspausdina tą buferį standartinėje išvestyje. Jis nebraižo.

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

print8

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

Tas pats kaip spausdinant, tik su 8 pikselių „Fusion“ šriftu.

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

print10

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

Tas pats kaip spausdinant, tik su 10 pikselių „Fusion“ šriftu.

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

line

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

Linija nuo x0,y0 iki x1,y1. Du argumentai tęsiasi nuo ankstesnės pabaigos. line() atstato tą grandinę.

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

circ

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

Apskritimo kontūras. Spindulys nurodytas pikseliais.

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

circfill

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

Užpildytas ratas.

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

oval

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

Elipsės kontūras ribojančioje dėžutėje. Kampai yra įtraukti.

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

ovalfill

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

Užpildyta elipsė ribojančioje dėžutėje.

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

rect

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

Stačiakampio kontūras. Kampai yra įtraukti.

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

rectfill

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

Užpildytas stačiakampis.

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

rrect

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

Apvalus stačiakampio kontūras. x, y yra viršutinis kairysis kampas. w ir h yra dydis pikseliais; abu turi būti didesni nei 0. r yra kampo spindulys, kurio numatytoji vertė yra 0. Jis tvirtinamas prie pusės trumpesnės kraštinės.

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

rrectfill

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

Užpildytas apvalus stačiakampis. Tie patys argumentai kaip ir „right“.

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

fillp

`fillp([p])`

16 bitų užpildymo šablonas vėliau užpildytoms figūroms. 0 yra vientisa reikšmė. fillp() grąžina dabartinį šabloną. Bitai pereina 4×4 šabloną.

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

pal

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

Perkonfigūruoja spalvą c0 kaip c1. 0 režimas yra piešimo paletė, 1 – rodymo paletė, 2 – užpildymo rašto paletė. pal() atstato pradinius parametrus.

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

palrgb

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

Nustato vieną paletės įrašą į RGB, kiekvieną kanalą nuo 0 iki 255.

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

palt

`palt([col, t])`

Skaidrumas. palt(col) arba palt(col, true) padaro col permatomą. palt(col, false) padaro ją vientisą. palt() padaro visas spalvas nepermatomas.

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

setpalette

`setpalette(n)`

Pasirenka palečių banką n. setpalette() grąžina dabartinį banką.

```
setpalette(1)
```

tline

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

Brėžia liniją ekrane, imant žemėlapio pikselių pavyzdžius iš žemėlapio pozicijų mx, my, mdx, mdy ir žingsniuojant žemėlapio pavyzdį išilgai linijos.

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

Sprites ir žemėlapiai

spr

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

Piešia спрайтą n ties x ir y. w ir h yra dydis pikseliais, numatytoji reikšmė – 8. fx ir fy apverčia horizontaliai ir vertikaliai, kai dydis nenulinis. tex yra спрайто lapo skirtukas. rot yra kampas apsisukimais (0..1), ta pačia kryptimi kaip ir sin bei cos. 0 palieka спрайтą nepasuktą.

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

sspr

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

Nukopijuoja спрайтų lapo stačiakampį į ekraną, kurio mastelis yra dw x dh. Numatytosios reikšmės išlaiko šaltinio dydį.

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

sget

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

Vieno tekselio spalva спрайтų lape.

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

sset

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

Įrašo vieną tekselį спрайтų lape.

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

fget

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

Sprite'ų vėliavėlės. fget(n) grąžina vėliavėlės baitą. fget(n, f) yra „true“, kai nustatytas vėliavėlės bitas f (0..7).

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

fset

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

Nustato спрайтų vėliavėles. fset(n, v) įrašo visą baitą. fset(n, f, v) nustato vieną bitą.

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

setspr

```
setspr(n)
```

Pasirenka aktyvų спрайтų lapą. setspr() grąžina dabartinį lapą. Tas pats bankas kaip ir settex.

```
setspr(1)
```

settex

```
settex(n)
```

„Setspr“ slapyvardis.

```
settex(0)
```

map

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

Brėžiamas žemėlapis. Žemėlapio langeliai cx ir cy braižomi ekrane sx ir sy. cw ir ch, jei 0 reiškia visą žemėlapį. Sluoksnis priimamas ir ignoruojamas. tex yra plytelių rinkinio lapas.

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

mget

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

Spraito ID žemėlapio langelyje cx, cy.

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

mset

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

Įrašo спрайтą n į žemėlapio langelį cx, cy.

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

setmap

```
setmap(n)
```

Pasirenka n-tąjį žemėlapį. „setmap()" grąžina dabartinį žemėlapį. 0-asis žemėlapis yra tikrasis žemėlapis.

```
setmap(1)
```

Atmintis

peek

```
peek(addr)
```

Nuskaito vieną baitą iš 64 KB lango. Langas yra tiesioginio ekrano vaizdas, tada sukuria спрайте pikselius, tada susieja langelius, kai ekranas telpa į 65280 baitų. Didesni ekranai puslapiuojami su poke(0x5F40, page). Neatpažinti adresai grąžina 0. Taip pat rašoma @addr.

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

poke

```
poke(addr, ...)
```

Rašo baitus adresu „addr“. Papildomi argumentai įrašo šiuos baitus. Nesusieti adresai ignoruojami. Taip pat rašoma „$addr“, „val“. Ekrano bakstelėjimas skaičiuojamas kaip pikselio įrašymas.

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

peek2

```
peek2(addr)
```

Neženklinė 16 bitų reikšmė, mažojo galo, adresu.

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

poke2

```
poke2(addr, val)
```

Įrašo neženklinę 16 bitų reikšmę, mažojo galo terminą.

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

peek4

`peek4(addr)`

Nepažymėta 32 bitų reikšmė, mažoji pabaiga.

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

poke4

`poke4(addr, val)`

Įrašo neženklinę 32 bitų reikšmę, mažąją pabaigą.

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

memcpy

`memcpy(dst, src, len)`

Kopijuoja len baitus tame pačiame 64 KB lange.

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

memset

`memset(dest, val, len)`

Užpildo len baitus, pradedant nuo dest, su val.

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

Matematika

abs

`abs(x)`

Absoliuti vertė.

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

flr

`flr(x)`

Apvalina link minus begalybės.

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

ceil

`ceil(x)`

Apvalina link plius begalybės.

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

max

`max(a, b)`

Didesnė iš dviejų verčių.

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

min

`min(a, b)`

Mažesnė iš dviejų verčių.

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

mid

`mid(a, b, c)`

Vidurinė iš trijų verčių. Naudingas kaip spaustukas.

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

sgn

`sgn(x)`

X ženklas: -1, 0 arba 1.

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

sqrt

`sqrt(x)`

Kvadratinė šaknis. Neigiamas argumentas nėra skaičius.

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

sin

`sin(x)`

Sinusas. x paeiliui užrašomas: 0, 0,25, 0,5, 0,75 apeina apskritimą vieną kartą. Ne radianais.

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

cos

`cos(x)`

Kosinusas, savo ruožtu, taikomas tai pačiai konvencijai kaip ir sinusas.

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

atan2

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

Vektoriaus dx, dy kampas, kaitaliojamas nuo 0 iki 1.

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

rnd

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

Atsitiktinis skaičius tarp [0, x). Numatytoji x reikšmė yra 1, taigi rnd() yra 0..1. Jei metate kauliuką, flr(rnd(6))+1.

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

srand

```
srand(seed)
```

Įterpia sėklą atsitiktinių skaičių generatoriui, kad vėlesnę vijų seką būtų galima kartoti.

```
srand(1)
```

band

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

Bitų lygtis IR. Taip pat rašoma a ir b.

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

bor

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

Bitų ARBA. Taip pat rašoma a | b.

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

bxor

`bxor(a, b)`

Bitų XOR. Taip pat rašoma a ^^ b.

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

bnot

`bnot(a)`

Bitiniu būdu NE. Taip pat rašoma ~a.

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

shl

`shl(a, n)`

Pasukite į kairę. Taip pat rašoma << n.

shr

`shr(a, n)`

Aritmetinis poslinkis į dešinę. Taip pat rašomas a >> n.

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

lshr

`lshr(a, n)`

Loginis poslinkis į dešinę. Taip pat rašoma >>> n.

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

rotl

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

Pasukti bitus į kairę. Taip pat rašoma <<> n.

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

rotr

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

Pasukti bitus į dešinę. Taip pat rašoma >>< n.

Lentelės ir reikšmės

add

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

Prideda v prie lentelės t. Jei duota i, įterpia ties tuo 1 pagrindu sukurtu indeksu.

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

del

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

Pašalina pirmą įrašą, lygų v.

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

deli

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

Pašalina indeksą i ir grąžina jo reikšmę. Jei i nėra, pašalina paskutinį įrašą.

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

all

```
all(t)
```

Iteratorius per masyvo dalį, eilės tvarka. Skylės sustabdo ėjimą taip, kaip baigiasi 1 pagrindu sukurtas masyvas.

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

foreach

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

Kiekvienam masyvo elementui iškviečia fn(v).

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

pairs

```
pairs(t)
```

Iteratorius kiekvienam raktui ir reikšmei, įskaitant pavadintus laukus.

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

ipairs

```
ipairs(t)
```

Iteratorius per masyvo dalį su 1 pagrindu sukurtu indeksu.

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

next

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

Grąžina kitą raktą ir reikšmę po k. next(t) prasideda nuo pradžios. Naudojama poromis.

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

count

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

Masyvo dalies ilgis. Jei duotas v, suskaičiuoja, kiek įrašų lygu v.

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

pack

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

Supakuoja argumentus į naują lentelę.

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

unpack

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

Grąžina nuo t[i] iki t[j] kaip atskiras reikšmes. Numatytosios reikšmės apima visą masyvo dalį.

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

select

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

select("#", ...) grąžina, kiek argumentų seka. select(i, ...) grąžina argumentus nuo i.

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

assert

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

Sustoja su klaida, kai c yra klaidinga arba nulis. msg pakeičia numatytąjį tekstą.

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

pcall

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

Iškviečia f. Sėkmingo įvykdymo atveju grąžina „true“ ir rezultatus. Klaidos atveju grąžina „false“ ir klaidos tekstą.

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

Stygos

chr

```
chr(n)
```

Vienas simbolis iš baitų kodo 0..255.

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

ord

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

Baitų kodai raidėms „s“ nuo indekso i iki j. Indeksai yra pagrįsti 1. Numatytasis yra pirmas simbolis.

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

sub

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

Dalinis tekstas nuo i iki j, skaičiuojamas nuo 1. Praleiskite j, kad gautumėte uodegą.

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

split

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

Padalija „s“ į lentelę. „sep“ pagal numatytuosius nustatymus yra kablelis. Skaičius „sep“ yra segmento plotis. „conv“ konvertuoja dalis su „tonum“, kai tai yra tiesa.

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

tonum

```
tonum(s, [base])
```

Analizuoja skaičių. Bazė gali būti 2–36. Grąžina nulį, kai tekstas nėra skaičius.

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

tostr

```
tostr(v, [digits])
```

Paverčia reikšmę tekstu. „digits“ nustato skaičių dešimtaines vietas.

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

Garsas

sfx

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

Atkuria n-tąjį šabloną ir grąžina kanalą. Praleidžiant ch arba perleidžiant -1, pasirenkamas pirmas laisvas kanalas. ch -2 sustabdo n-tąjį šabloną kiekviename kanale. n -1 sustabdo kanalą arba kiekvieną kanalą, kai praleidžiamas ch. n -2 leidžia cikliniam šablonui baigti veikti, o ne cikluoti. sfx() be argumentų grąžina -1.

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

sfxget

```
sfxget(n, step)
```

Grąžina vieno šablono žingsnio lentelę: [1] aukštis, [2] banga, [3] garsumas, [4] efektas. Aukštis yra 0,63 (nuo C-0 iki D#5). Banga yra 0,15. Garsumas yra 0,7, o 0 yra tylus. Efektas yra 0,7.

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

sfxset

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

Rašo vieną šablono žingsnį. Reikalingi visi šeši argumentai.

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

sfxhdr

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

Šablono antraštė. sfxhdr(n) grąžina antraštės lentelę. sfxhdr(n, key, val) nustato vieną lauką: speed, loop_in, loop_out, level, noise, crush, dtune, rverb, dampn, swing, steps. Skaičiai nuo 0 iki 10 pasirenka tuos pačius laukus tokia tvarka. speed yra TIC, tiksai per žingsnį. Didesnis yra lėtesnis. 15 yra 120 dūžių per minutę. 16 bitų priima 1 iki 255. Siauresni sintezatoriai prisjungia prie savo TIC sąrašo.

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

sfxstat

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

Šablono ID, grojamas kanale ch arba -1. Numatytasis kanalas yra 0.

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

note

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

Groja vieną natą be šablono. Aukštis 0–63, banga 0–15, garsumas 0–7, efektas 0–7. Tiksėjimai – tai išlaikymo trukmė, tie patys 120 Hz tiksėjimai kaip ir TIC. Praleidus tiksėjimus, garsas trunka 15 natų – vienas numatytasis žingsnis. Efektas veikia per tą laiką, tada kanalas nutyla.

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

music

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

Pradeda n-tąją dainą nuo viršaus. music(-1) sustoja. Iškvietus music su jau grojamos dainos ID, ji paliekama veikti. music() grąžina grojamos dainos ID arba -1. fade yra milisekundės: daina pradeda arba ima garsėti, kai n yra -1. Nustatyti kaukės bitai yra dainai naudojami kanalai. 0 reiškia visus aštuonis. Kiti kanalai lieka laisvi sfx.

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

musicstat

```
musicstat()
```

Grojamos dainos lentelė. „song“ ir „row“ yra dainos ID ir eilutė arba -1, kai niekas negrojama. „step“ ir „pos“ turi aštuonis skaičius. „step[1]“ yra 0 kanalas, „step[8]“ – 7 kanalas. „step“ yra natos indeksas arba -1, kai kanalas tylus. „pos“ nurodo, kiek toli nueina nata, nuo 0 iki 1. „tick“ nurodo, kiek taktų ši eilutė jau grojo. „played“ nurodo, kiek eilučių pradėta groti nuo „music()“ funkcijos, įskaitant grojamą eilutę.

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

songget

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

Šablono ID saugomas dainos langelyje. ch yra 0–7. 255 reiškia, kad langelis tuščias. Tai yra tiesioginis iškvietimas; kodo užuominos tekste vis dar rodoma senesnė argumentų tvarka.

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

songset

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

Į dainos langelį įrašo šablono ID. Langeliui išvalyti naudokite 255.

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

stat

```
stat(n)
```

Tik garsas. 16..23 yra sfx ID n-16 kanale. 24..31 yra muzikos modelis n-24 kanale. Bet kuris kitas n grąžina 0. Kadrų sinchronizavimo laikas ir RAM yra perf ir info, o ne stat.

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

Duomenų lentelės

dget

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

Nuskaito duomenų lentelės t langelį iš DATA skydelio. Eilutė ir stulpelis prasideda nuo 0.

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

dset

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

Rašo langelį. Teksto stulpeliui naudokite eilutę.

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

dstr

`dstr(t, row, col)`

Langelis kaip tekstas. Skaičiai konvertuojami.

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

drows

`drows(t)`

Kiek eilučių turi lentelė t?

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

dcols

`dcols(t)`

Kiek stulpelių turi lentelė t?

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

Išsaugojimas ir pasiekimai

save

`save(k, v)`

Įrašo faile „v“ po raktu k. „v“ gali būti skaičius, eilutė arba jų lentelė. Failas yra suspaustas.

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

load

`load(k)`

Nuskaito k raktą. Grąžina nil, kai rakto nėra.

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

delsave

`delsave(k)`

Ištrina k klavišą iš išsaugojimo.

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

saveslot

`saveslot([name])`

Pasirenka išsaugojimo failą autoriaus aplanke. Be pavadinimo, vieta yra žaidimo pavadinimas. Grąžina 1, jei tas failas jau yra.

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

achieve

```
achieve(i)
```

Atrakina i-ąjį pasiekimą nuo 0 iki 15. Saugoma lokaliai. „Steam“ versija taip pat nustato ACH_i. Katalogo sesija praneša apie tai žaidėjo paskyrai.

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

achieved

```
achieved(i)
```

1, kai i-asis pasiekimas jau atrakintas.

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

achdef

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

Tosto tekstas pasiekimui i, rodomas 15 sekundžių nuo ekrano apačios. spr yra pasirenkama 8×8 plytelė, perkelkite jos lapą.

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

achclear

```
achclear(i)
```

Išvalo pasiekimą i vietiniame faile. Jau nustatytas „Steam ACH_i“ lieka nustatytas.

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

Tinklas

whoami

```
whoami()
```

Jūsų žaidėjo ID yra nuo 0 iki 15, kol sesija vyksta. Neprisijungus jis yra -1.

```
id = whoami()
```

nhost

```
nhost()
```

1, jei šis kompiuteris yra pagrindinis kompiuteris, kitu atveju – 0.

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

nplayers

```
nplayers()
```

Kiek gyvų ID egzistuoja skaitmenų skaičiumi 0..15?

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

nlisten

```
nlisten()
```

Pradeda talpinti UDP ir grąžina LAN adresą kaip ip:port.

```
addr = nlisten()
```

njoin

```
njoin(addr)
```

Prisijungia prie ip arba ip:port.

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

nleave

```
nleave()
```

Išeina iš sesijos. Kolegos mato _mp_leave su „kodėl“ 0.

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

nsend

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

Siunčia nuostolingus pranešimus. Pavadinimas yra įvykis. Po jo seka iki keturių papildomų reikšmių. p – nuo 0 iki 15 pašnibždesių tam mazgui; praleiskite p, jei norite siųsti visiems.

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

ncall

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

Tokia pati forma kaip ir „nsend“, bet pranešimas bandomas išsiųsti dar kartą, kol gaunamas patvirtinimas. Trys nesėkmės iš eilės nutraukia sesiją.

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

nlan

```
nlan()
```

Dabartinis LAN IP:port po nlisten.

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

nstun

`nstun()`

Jūsų viešas IP:port adresas iš STUN. Svečias iš kito tinklo gali bandyti prie jo prisijungti.

```
pub = nstun()
```

setturn

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

TURN – atsarginis adresas, kai tiesioginis kelias nepavyksta. „setturn()“ su tuščiais argumentais jį išjungia. Adresas yra host:port.

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

nlist

`nlist(game)`

Atvirų sesijų, skirtų tam žaidimo pavadinimui, lentelė. Kiekviename įraše yra adresas, žaidėjai ir vardas.

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

nannounce

`nannounce(game)`

Širdies plakimas, kad šis serveris būtų rodomas žaidimo nlist'e. Iškvieskite jį laikmačiu serverio metu.

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

nkb

`nkb([on])`

Adreso laukas. nkb(1) jį atidaro, nkb(0) – paslepia, nkb() nurodo, ar jis atidarytas. Spausdami aukštyn ir žemyn, keičiate simbolį – nuo 0 iki 9, tašką arba dvitaškį. A jį įrašo. X ištrina. Klaviatūra į jį nespausdina.

```
nkb(1)
```

nkbstr

`nkbstr([s])`

Adreso lauke esantis tekstas, o tada aukštyn ir žemyn esantis simbolis keičiasi. nkbstr("") išvalo tekstą.

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

_mp

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

Jūs rašote šią funkciją. Ji paleidžiama, kai gaunamas pranešimas. „from“ yra siuntėjo ID. „name“ yra įvykis. Po to seka iki keturių reikšmių.

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

_mp_join

`_mp_join(p)`

Jūs rašote šią funkciją. Partneris gavo id p. Ji taip pat suveikia jums; whoami() yra tas id.

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

_mp_leave

```
_mp_leave(p, why)
```

Jūs rašote šią funkciją. Peer p išėjo. Kodėl 0 yra normalus išėjimas. Kodėl 1 yra nutrūkęs ryšys.

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

_mp_update

```
_mp_update()
```

Jūs rašote šią funkciją. Ji paleidžiama kas 15 _update kadrų, maždaug 4 Hz dažniu. Gera vieta nsend pozicijoms.

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

_mp_fail

```
_mp_fail()
```

Jūs rašote šią funkciją. Sesija neveikia: dingo pagrindinis kompiuteris arba ncall nepavyko tris kartus.

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

Korutinai

cocreate

```
cocreate(f)
```

Sukuria korutiną iš funkcijos f. Jis neveikia, kol neįvedamas coresume.

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

coresume

`coresume(c, ...)`

Vykdo korutiną c, kol gauna reikšmes arba grąžina reikšmę. Sėkmės atveju: „true“ ir gautos reikšmės. Klaidos atveju: „false“ ir klaida.

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

costatus

`costatus(c)`

„sustabdytas“, „bėga“ arba „negyvas“.

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

yield

`yield(...)`

Grąžina valdymą tam, kas iškvietė „coresume“. Vertės tampa to „coresume“ rezultatais.

```
yield(1)
```

Matas

info

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

Nubraižo tiesioginį HUD _draw formatu: fps (1 sekundės vidurkis), upd, drw, px, max, ram, peak, ramc ir net, kai sesija baigta. info() naudoja spausdinimo žymeklį. info(col) nustato tik spalvą. info(x, y) arba info(x, y, col) ją įdeda. Skaičiai naudoja B, k arba m, kad tilptų mažame ekrane. Pats iškvietimas pikseliais neskaičiuojamas.

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

showstats

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

Tas pats HUD kaip ir info, automatiškai nupiešiamas ties 0, 0 po _draw, ignoruojant kamerą ir klipą. showstats(0) jį paslepia. showstats(1) arba showstats() jį rodo.

```
showstats(1)
```

perf

```
perf(n)
```

Vienas HUD skaičius, kad galėtumėte jį nupiešti patys. 0 kadrų per sekundę, 1 atnaujinimas, 2 draw, 3 px, 4 max (sunkiausias kadras šią sekundę), 5 RAM, 6 RAMc (RAM plius supakuota žaidimo kasetė), 7 Peak RAM nuo žaidimo kasetės išleidimo.

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