

Scratchi põhiobjektid



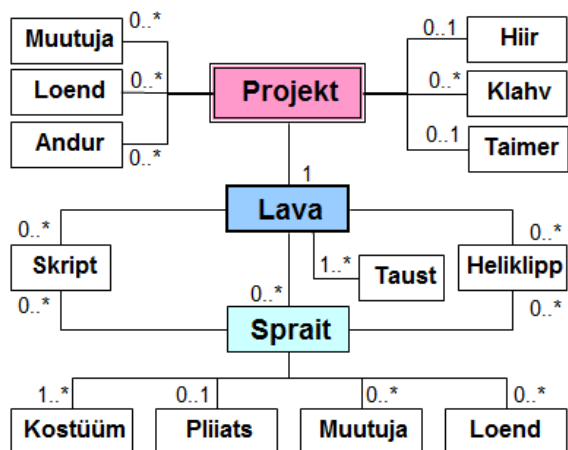
Scratch ei ole tegelikult päris objektorienteeritud süsteem, vaid tegemist nn objektipõhise süsteemiga. Objektideks on spraidid, kostüümid, lava, heliklipid jms. Vahendite süsteemseks käsitlemiseks on otstarbekas kirjeldustes lähtuda objektidest. Oluline on seejuures ka asjaolu, et Scratchi abil saab tutvuda objektorienteeritud lähenemisviisi ja modelleerimise põhielementidega.

Sisu

Scratchi projekti struktuur ja põhiobjektid	3
Projekt	3
Lava ja taustad.....	3
Spraidid ja kostüümid.....	5
Olemus, omadused ja meetodid	5
Spraitide ja kostüümide lisamine, redigeerimine ja seadistamine	5
Animeeritud GIF-failide kasutamine spraitidena	8
Heliklipid.....	8
Hiir ja klahvid	9
Taimer.....	9
Andurid.....	10
Puuteandurid.....	10
Kauguse ja omaduste andurid.....	11
Muud andurid.....	12
Muutujad, loendid ja skriptid	12
Kasutaja objektid.....	12
Krapu skriptid.....	12
Seosed objektide vahel	13

Scratchi projekti struktuur ja põhiobjektid

Scratchi projekt koosneb omavahel seotud objektidest. Kasutatavad **objektid** ja **seosed** nende vahel on toodud UML klassidiagrammina. Vt jaotist [Modelleerimine](#). Projekti loomisel ja täitmisel saab kasutada ja muuta objektide **omadusi** (spraidi ja hiirekursori koordinaadid, spraidi suurus ja nähtavus, taimeri ja muutuja väärtus jm). Suur osa Scratchi käskudest (käsuplokkidest) on käsitletavat objektide **meetoditena**, mille abil saab määrata tegevusi (liikumine, pööramine, värvi muutmine jm) spraitide ja ka teiste objektidega (pliiats, heli jm). Teatud omadusi (n lava olek) saab määrata ja tegevusi täita ainult loomise faasis nõ „käsitsi“, mitmeid omadusi saab muuta nii loomise faasis kui ka skriptide täitmisel. Mõnede objektide abil saab tekitada **sündmusi** (hiir, klahvid), mõned (spraidid, lava jm) aga võivad reageerida täitmise ajal teatud sündmustele (hiireklõps, vajutus klahvile jm).



Seoste otses toodud kordused näitavad, mitu objekti ühest klassist võivad olla seotud teise klassi objektiga. Vaikimisi on kordus 1.

Projekti on alati üks ja ainult üks lava (kordus 1), millel võib asuda suvaline arv spraite, sh ka null (kordus 0..*). Sama tüüpi kordus on mitmel teisel objektil: skript, helikliipp, muutuja jm. Laval on alati vähemalt üks taust ja spraidil üks kostüüm (1..*). Kordus 0..1 näitab, et kasutusel võib olla üks objekt või ei ühtegi (hiir, taimer, pliiats).

Muutujad ja **loendid** võivad esineda kahel tasemel: **Projekt** ja **Sprait**. Esimesel juhul on tegemist **globaalsete** andmetega, millele on juurdepääs projekti kõikidel spraitidel, teisel **lokaalsete** andmetega, millele on otsene juurdepääs ainult selle spraidi skriptidel, mille jaoks on muutuja või loend loodud.

Projekt

Projekt on kõrgema tasemega objekt, kõik teised objektid kuuluvad selle sisse (lava, spraidid, muutujad jm) või on kasutusel sellega töötamisel (hiir, klahvid jm).

Projekt
nimi, tüüp, maht, asukoht, keel, ...
loo uus projekt, ava olemasolev, impordi projekt, salvesta, salvesta nimega, lae üles serverisse, määra keel, ...

Scratchi projekt kujutab endast faili laiendiga .sb. See luuakse kasutades skeemil näidatud komponente ja need on kasutuselt töötavas projektis.

Formaalselt on tegemist võrdlemisi tüüpilise nõ faili objektiga. Sellised omadused nagu nimi, tüüp jne on olemas praktiliselt kõikidel failidel. Sama võib öelda ka põhitegevuste (avamine, salvestamine jne) kohta. Olgu märgitud, et neid tegevusi saab täita ainult loomise faasis vastavate korraldus abil, mis asuvad menüüs.

Üheks omapäraks on võimalus laadida projekt serverisse, mis asub MITs, säilitada ja täita nõ pilvedes.

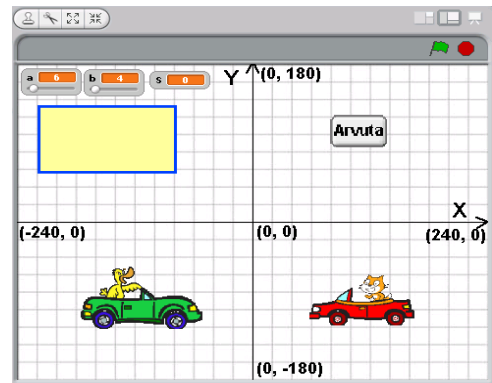
Lava ja taustad

Lava (Stage) on ekraani piirkond, kus asuvad ja tegutsevad kõik spraidid. Siin asub ka suurem osa kasutajaliidese elementidest: muutujate monitorid, käsunupud jms.

NB! Siin ja edaspidi: tegevustele, mida saab täita ka skriptides, järgnevad tunnusena tühjad sulud – **võta taust()**, **mängi heli()** jmt. Tegevustele, mida saab täita ainult loomise faasis, sulge ei järgne.

Lava
x_min = -240, x_max = 240 y_min = -180, y_max = 180 olek, värvus, helitugevus tausta nimi, tausta number, ...
muuda olekut, võta taust(), järgmine taust(), muuda värvi(), mäangi heli(), mäangi nooti(), mäangi trummi(), küsi(), ...

Taust
nimi, number, mõõtmed, maht, vorming, ...
impordi, kopeeri, redigeeri, eemalda, ...



Koordinaattelgede algus on lava keskel. Lava on 480 pikselit lai ($x_{\min} = -240$, $x_{\max} = 240$) ja 360 pikselit kõrge ($y_{\min} = -180$, $y_{\max} = 180$). Tegemist on tingühikuga, mille suurus sõltub ekraani suuruselt ja kasutatavast eraldusvõimest.

Projekti loomisel ja täitmisel saab lava kohal asuvate nuppude abil valida lava kolme oleku (suuruse) vahel: **normaalne**, **väike** ja **esitlus**. Skriptides olekut muuta ei saa. Esimene variant on tavaline lava olek rakenduste loomisel ja ka läbivaatamisel. Selles olekus saab lisada uusi spraitte, muuta nende asukohta ja suurust jm, on nähtavad ka liidese teised piirkonnad. Väike lava olek on põhimõtteliselt sama, ainult lava mõõtmed on väiksemad ja skriptide jaoks on rohkem ruumi. Esitluse olekus on täisekraanil nähtav ainult lava ning käivitamise ja peatamise nupud. Selles olekus ei saa lisada ega muuta spraitte ja skripte.

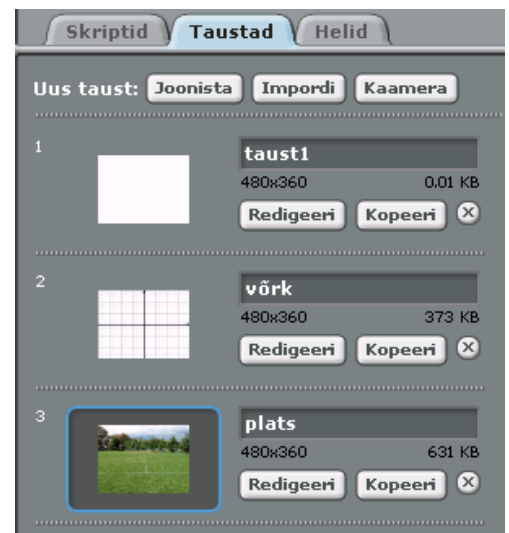
Laval võib olla suvaline hulk **taustu**. Pildil on nähtaval kolme tausta ikoonid ja omadused. Vaikimisi on laval üks valge taust. Taustu saab lisada, eemaldada ja redigeerida, kui muuta aktiivseks lava ikoon ning klõpsata lipikut **Taustad** ning kasutada vastavaid korraldusi.

Korraldusega **Impordi** saab taustu valida süsteemiga kaasas olevatest taustade hulgast või võtta kasutaja graafikafailides olevaid pilte: **GIF-, PNG-, BMP-, JPG-**failid.

Joonista käivitab joonistamisredaktori, mille abil saab joonistada taustapildi.

Kaamera võimaldab teha taustapildi veebikaameraga.

Redigeeri võimaldab muuta olemasolevat tausta joonistamisredaktoriga.



Käsk **Kopeeri** teeb tausta koopia. Tavaliselt selleks, et viia koopias sisse muudatusi ja täiendusi.

Ristiga nupuke eemaldab tausta.

Taustad moodustavad järjestatud kogumi (kollektiooni), mille üks element on nähtav (aktiivne). Tausta saab muuta nähtavaks loomise faasis, klõpsates vastavat ala vahelehel **Taustad**. Saab muuta taustade järjestust, lohistades hiirega ikoone vajalikele kohtadele. Taustu saab vahetada ka lava skriptides käskudega: [**võta taust nimi**], [**võta taust number**] ja [**järgmine taust**]. Taustapildid võivad oluliselt suurendada Scratch'i projekti faili mahtu. Sellepärast on otstarbekas kasutada käsku **Paki pildid** faili kokkusurumiseks.

Spraidid ja kostüümid

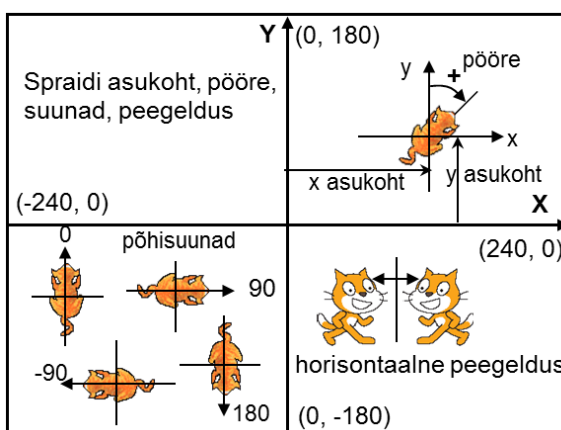
Olemus, omadused ja meetodid

Spraidid on Scratchi projektide peamised objektid, rakenduste töö on suurel määral orienteeritud just nende kasutamisele. Sprait kujutab endast graafiliste kujundite kogumit, neid kujundeid nimetavad süsteemi loojad **kostüümideks** (*costume*). Kui sprait on nähtav, siis on nähtav üks selle kostüümidest (kujutistest). Vahetades skriptide abil teatud sagedusega kostüüme ja/või muutes ka spraidi asukohta, saab tekitada erinevaid animatsiooniefekte: tantsimine, käimine, lendamine jmt. Kostüüme saab kasutada ka muul otstarbel. Oma olemuselt ja ka käsitluselt on kostüümid samased lava taustadega. Paneme tähele, et kostüümidel ja taustadel on ühesugune valik omadusi ja meetodeid. Teatud erinevused tulenevad kasutamise eesmärkidest ja sisulisest tähendusest (semantikast).



Sprait
nimi, suurus, nähtavus, x_asukoht, y_asukoht, pööre, suund, värvus, keere, heledus, ...
lisa, eemalda, muuda asukohta, muuda suurust, paljunda, ekspordi, ... liigu(), pööra(), muudaX(), muudaY(), vaheta kostüümi(), muuda värvi(), ütle(), mängi heli(), ...

Kostüüm
nimi, number, mõõtmed, maht, vorming, ...
impordi, kopeeri, redigeeri, eemalda, ...



Spraidi geomeetriselised omadused on seotud lava koordinaatsüsteemiga. Asukoht määratakse keskkoha koordinaatidega (**x asukoht**, **y asukoht**) lava telgede (**X**, **Y**) suhtes. Pööret ja liikumise suunda arvestatakse **Y**-teljest, positiivne nurk on päripäeva. Nurga ühikuks on kraad.

Spraitide ja kostüümide lisamine, redigeerimine ja seadistamine

Sprait ja kostüüme saab lisada, redigeerida ja eemaldada projekti loomise faasis. Skriptide abil neid tegevusi täita ei saa. Sprait võib valida Scratchiga kaasas olevast spraitide hoidlast või oma kaustades olevates failidest. Sprait koosneb algselt ühest või mitmest graafilisest kujundist ehk kostüümist. Scratchist eksporditud spraidi juurde võivad kuuluda ka skriptid. Spraitide loomiseks võib kasutada **PNG**, **GIF**, **JPG**, **BMP** ning animeeritud **GIF** faile. Scratchist eksporditud spraidi faili tüübiks on **SPRITE**, see võib koosneda mitmest graafilisest kujundist (kostüümist) ja skriptidest. Sprait (kostüüme) saab joonistada joonistus-redaktori abil ka ise. Viimasega saab muuta ja täiendada ka olemasolevate spraitide kostüüme.

Uue spraidi esimese või ainukese kostüümi loomiseks kasutatakse käsku [**Vali uus sprait failist**]. Avaneb dialoogiboks juurdepääsuga Scratchi kaustale **Costumes (Kostüümid)**, mille alamkaustades: **Animals**, **Fantasy**, **Letters**, **People** jne asuvad failid spraitide loomiseks. Samast pääseb nupuga **Arvuti** ligi ka teistele kaustadele. Valitud sprait (kostüüm) paigutatakse lava keskele. Lohistades hiirega saab viia kujundi suvalisele kohale laval. Plokkidega grupist **Liikumine** saab vajadusel määrata spraidi jaoks täpsed koordinaadid (**x** ja **y**).

Kasutades tööriistariba nuppe **Kasvata spraiti (Grow sprite)** või **Kahanda spraiti (Shrink sprite)**, saab suurendada või vähendada kujundit. Selleks klõpsata vastavat nuppu ja siis üks või mitu korda spraiti.



Protsessi kiirendamiseks võib hoida all **Shift**-klahvi. Spraiti saab suurendada ja vähendada klõpsates seda hiire parempoolse nupuga ja valides ilmuvast menüüst käsu **muuda spraidi suurust**. Spraidi suuruse muutmiseks võib kasutada ka grupi **Välimus** käske **[võta suuruseks p %]** või **[muuda suurust h võrra]**, võttes **p** või **h** jaoks sobiva

väärtuse ja klõpsates plokki. Käsus **[võta suuruseks ... %]** arvestatakse, et spraidi originaalsuurus on 100%.

Kõikide projektis kasutatavate **spraitide ikoonid** on kuvatud piirkonnas, mis asub lava all. Iga spraidi jaoks on näidatud **nimi**. Vaikimisi on spraitide nimedeks **Sprait1**, **Sprait2**, ... Spraidi nime saab muuta **spraidi jooksva info alas** olevas nime väljas.

Igal ajahetkel on aktiivne ühe spraidi (või lava) ikoon. Spraidi skriptide, kostüümide ja helide vaatamiseks ja redigeerimiseks peab muutma selle aktiivseks, klõpsates ikooni spraitide loetelus või tehes topeltklõpsu laval oleval spraidil. Aktiivne spraidi ikoon on spraitide loetelus esile tõstetud, selle ümber on sinakas raam.



Jooksev **informatsioon aktiivse spraidi kohta** on kuvatud kasutajaliidese ülemises ääres keskel. Jooksva info alas on väli spraidi nimega ja just siin saab seda muuta. Vähemalt põhispraitidele on soovitatav anda asjakohased nimed. Piirkonnas on näha spraidi keskpunkti jooksvaid koordinaate (**x** ja **y**) ning suunda.

Spraidi suund näitab suunda (nurka), kuhu sprait liigub, kui täidetakse plokk käsuga **[liigu ... samm]**. Standardsed suunad on: 0 – üles, 90 – paremale, 180 – alla, -90 – vasakule. Neid pakutakse valimiseks käsus **[osuta suunas ...]**, kuid saab määrata ka suvalise suuna (kraadides). Sinine joon spraidi juures näitab selle jooksvat suunda. Seda saab muuta hiirega lohistades, topeltklõps seab spraidi suuna algseisu (90 kraadi). Suunda muudetakse tavaliselt skriptides käskudega **[osuta]** (**[point]**) ja **[pööra .. kraadi]** (**[turn .. degrees]**) ning **[kui äärel, pörka]** (**[if on edge, bounce]**).

Jooksva info alas on ka nupud pööramise stiilide määramiseks, millest sõltub spraidi käitumine suuna muutmisel (näiteks pörkamisel servast). On võimalikud kolm varianti:

- **Saab pöörata (can rotate)**. Sprait pöörduv suuna muutmisel vabalt vastavalt antud pöördnurga suurusele.
- **Ainult nägu vasakule-paremale (only face left-right)**. Vertikaalne peegeldus: sprait pöörduv (peegeldub) ainult vasakule või paremale.
- **Ei pöörata (don't rotate)**. Spraiti ei pöörata suuna muutmisel.

Klõpsatus **taba** kohal, võtab spraidi lukust lahti või lukustab selle. Kui sprait on lukustamata, saab selle asukohta muuta lohistamise teel esitluse (täisekraani) olekus ja veebileheküljel.

Spraiti saab paljundada, klõpsates seda laval või selle ikooni spraitide loetelus hiire parempoolse nupuga ja valides ilmuvast menüüst käsu **paljunda (duplicate)**. Paljundamiseks võib kasutada ka tööriistariba nuppu **Paljunda**: klõpsata templiga nuppu riistaribal ja seejärel vastavat spraiti.

NB! Spraidi paljundamisel tulevad koopiasse kaasa ka kõik originaali kostüümid ja skriptid.

Objektimenüü käsuga **ekspordi see sprait (export this sprite)** või **Fail**-menüü käsuga **Spraidi eksportimine (Export Sprite)** võib salvestada spraidi (koos kostüümide ja skriptidega) eraldi faili, mida hiljem saab kasutada teistes projektides.

Spraidi eemaldamiseks võib kasutada tööriistariba käsku **Eemalda (Delete)**: klõpsata kääridega nuppu riistaribal ja seejärel spraiti. Võib kasutada ka menüüd, tehes paremklopsu spraidil või selle ikoonil ja valides ilmuvast menüüst **kustuta (Delete)**.



Kostüümide lisamine toimub ühekaupa käsuga **Impordi (Import)** vahelehel **Kostüümid (Costumes)**. Ühe spraidi jaoks mõeldud kostüümid on tüüpiliselt sama nimega ja erinevad numbriga (indeksi) poolest. Kõrval on näidatud tantsiva Anjuli esimesed kolm kostüümi, kokku on neid viis.

Nupuga **Redigeeri (Edit)** saab käivitada joonistamisredaktori ja teha muudatusi kostüümis. Kostüüme saab ka kopeerida ja kustutada. Kostüümide järjestust saab muuta hiirega lohistades. Nime muutmiseks sisestada vastavasse välja uus väärtus. Paremklops kostüümil kuvab menüü, mis pakub võimalust muuta kostüüm omaette spraidiks või eksportida see eraldi faili. Varem eksportitud spraidi lisamisel projekti, tulevad kaasa kõik spraidi kostüümid ja skriptid.

Igal ajahetkel on aktiivne (nähtav) üks kostüüm. Kostüümi saab muuta aktiivseks käskudega **[järgmine kostüüm]** ja **[võta kostüüm nimi|number]**. Numbriga kasutamisel peab olema tegemist muutujaga. Kõrvaltoodud skriptis vahetatakse järjest kostüüme järjenumbritega kolm kuni kuus.

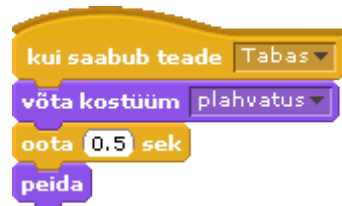
Tegevused (asukoha muutmine, pööramine, suuruse muutmine jmt), mõjutavad selle spraidi kõiki kostüüme.



Algselt olid kostüümid mõeldud mitmesuguste animatsiooni-efektide loomiseks: jooksmine, tantsimine jmt. Kuid nad leiavad kasutamist mitmete teiste tegevuste esitamiseks.

Näiteks on mingis mängus lennukil kaks kostüümi. Üks kostüüm on lennuki pilt nõ normaalvaates ja teine kujutab plahvatust.

Kui tabatakse lennukit, teeb kõrvalolev skript aktiivseks kostüümi **plahvatus** ning peale väikest pausi sprait peidetakse.



Kostüüme saab kasutada erinevate tekstiteadete kuvamiseks. Näiteks mingi mängu tulemuse kuvamiseks on spraidil **teade** kolm kostüümi tekstidega:

kostüüm1: Võitis Pille!,

kostüüm2: Võitis Kalle!,

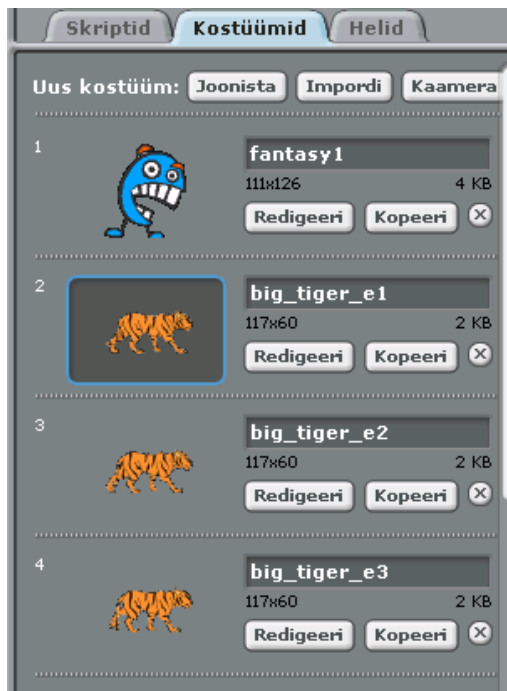
kostüüm3: Viik!

Mängu alguses sprait peidetakse. Sõltuvalt teatest muudetakse aktiivseks vastav kostüüm.



Animeeritud GIF-failide kasutamine spraitidena

Animeeritud GIF-failid leiavad juba ammu laialdast kasutamist animeeritud piltide lisamiseks teatud tüüpi dokumentidesse, eeskätt esitlustesse ja veebidokumentidesse, kus nad hakkavad „elama“. Taolisi faile on Internetis tuhandeid ja neid saab vabalt alla laadida. Animeeritud GIF fail koosneb, nagu Scratchi sprait, mitmekihilisest kujundite kogumist, mis lisatuna Scratchi projekti, jaguneb kostüümideks.



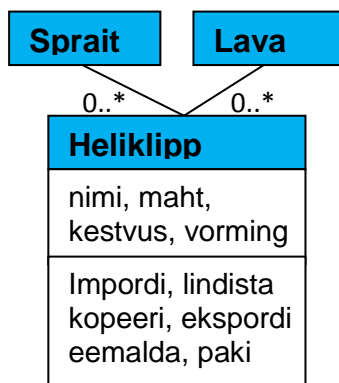
Faili lisamine Scratchi toimub veidi omapäraselt. Kõigepealt peab panema lavale ühe spraidi, mille kostüümiks on suvaline graafikafail mõnes Scratchis lubatud formaadis. Selle võib näiteks teha joonistamisredaktoriga või lisada korraldusega **Vali üllatussprait** (nupp lava all). Seejärel peab importima selle kostüümide hulka animeeritud **GIF**-faili, mis automaatselt jaguneb kostüümideks. Esialgu lisatud kostüüm on esimene, selle peab eemaldama.

Näiteks animeeritud **GIF**-failist **big_tiger** tuleb 16 kostüümi. Need saab panna „elama“ (liikuma kohapeal) kõrvaloleva skripti abil. Analoogselt saab panna „töle“ suvalise animeeritud GIFst saadud spraidi. Käsuga **[oota ... sek]** saab reguleerida kaadrite vahetamise kiirust. Spraidi saab panna liikuma mööda lava ja täitma muid tegevusi.

[Demo.](#)

Helikliidid

Helikliidid võivad olla seotud spraitidega ja lavaga. Objektid luuakse helifailidest. Kasutakse WAV ja MP3



vormingus faile. Allalaadimisel tuleb Scratchiga kaasa komplekt helifaile, mille hulgast saab importida helisid. Võib kasutada oma helifaile nimetatud vormingus. Kui arvutiga on ühendatud mikrofoni, võimaldab Scratch lindistada oma juttu ja muid helisid. Helifaile saab importida, lindistada, eksportida ja lindistada, vt. vaheleht **Helid**.

Kõik helikliippidega seotud tegevused

tehakse loomise faasis. Helikliippe saab käivitada grupi **Heli** käskudega: **[mäangi heli nimi]**, **[mäangi heli nimi kuni valmis]**.

[mäangi heli ...] käivitab nimega määratud klipi ja skripti töö jätkub kohe.

[mäangi heli ... kuni valmis] puhul mängitakse klipi lõpuni ja alles siis jätkub skripti töö.

Käsuga **[peata kõik helid]** saab lõpetada helide mängimise.

Hiir ja klahvid

Neid seadmeid kasutatakse eeskätt sündmuste tekitamiseks ning vastavate skriptide käivitamiseks. Vastavad tarkvaraobjektid teevad kindlaks hiire klõpsud ja vajutused klahvidele ning võimaldavad kasutada nende seadmetega seotud omadusi: hiirekursori koordinaadid, hiire olek (nupp all või mitte), klahvile vastav sümbol (täht, number või muu). Allpool on toodud mõned näited, milles kasutatakse erinevaid sündmusi ja objektide omadusi.



kui klõpsatakse Lava
järgmine taust

Hiiresündmus

Kui klõpsatakse lava, muutub aktiivseks järgmine taust.



kui klõpsatakse Kraps
üttele Tere! Olen Kraps! 2 sekundit
küsi Mis on Sinu nimi? ja oota
üttele ühenda Tere, vastus 2 sekundit

Hiiresündmus ja klahvide kasutamine

Kui klõpsatakse spraiti Kraps, käivitub kolmest käsust koosnev skript. Käsu [küsi ... ja oota] täitmisel loetakse kasutaja poolt klaviatuurilt sisestatud väärtus ja salvestatakse plokis (muutujas) *vastus*.



kui vajutatakse klahvi vasak nool
osuta suunas -90
liigu 20 sammu

Klahvisündmus

Kui vajutatakse klahvi **vasak nool**, võetakse spraidi suunaks 90° ja see liigub 20 sammu.



kui vajutatakse klahvi k
liigu 1 sek. x: hiire x y: hiire y -le

Klahvisündmus ja hiire omaduste kasutamine

Kui vajutatakse klahvi **k**, liigub sprait ühe sekundiga hiirekursori koordinaatidega määratud punkti.



kui vajutatakse klahvi tühik
lõputult kui hiir all?
mine hiire kursor

Klahvisündmus ja hiire oleku kasutamine

Kui vajutatakse klahvi **tühik**, käivitub lõputu kordus. Iga kord, kui vajutatakse hiire vasakule klahvile, liigub sprait kursori asukohta.

Taimer

Taimerit saab kasutada jooksva aja kuvamiseks, rakenduste ja ülesannete täitmise aja kindlaks tegemiseks, mängude, testide ja muude tegevuste lõppaja määramiseks. Taimeril on ainult üks omadus: **jooksev aeg**, mille saab viia algseisu käsuga (meetodiga) [**taimer algseisu**].



kui vajutatakse klahvi tühik
pane x -240 -ks
taimer algseisu
võta h = juhuarv 2 kuni 10
korda kuni x asukoht > 240
liigu h sammu
võta aeg = taimer

Näide. Auto liigub lava vasakust servast parema servani juhusliku sammuga *h* (2 .. 10).

Käsuga [**taimer algseisu**] võetakse taimeri algväärtuseks **0** ja väärtus hakkab automaatselt suurenema vastavuses arvuti sisetaimeri tööga. Käsuga [**võta aeg = taimer**] omistatakse taimeri jooksev väärtus muutujale *aeg*, mille monitoris see kuvatakse laval. Skripti töö lõppemisel jääb monitori lõppaeg, aga taimeri töö jätkub. Selle jooksvat aega võib näha, kui lasta kuvada vastav monitor.

Järgnevad skriptid demonstreerivad taimeri kasutamist täitmise aja kontrollimiseks.



Korrutustabeli trenn. Eesmärgiks on lahendada võimalikult rohkem ülesandeid etteantud aja jooksul. [Demo](#).

Põhiskript kuvab järjest ülesandeid, loeb kasutaja vastuseid ning hindab neid. Paralleelselt sellega töötab skript, mis kontrollib aega.



Muutuja **aeg** väärtuseks võetakse piiraeg, mida korduses vähendatakse järjest kuni jõutakse nullini.

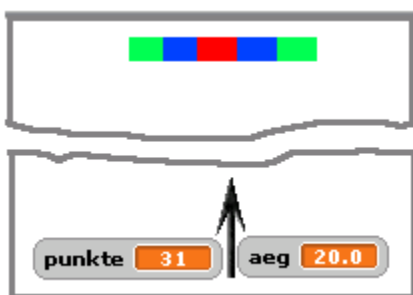
Käsk **[peata kõik]** lõpetab kogu programmi (kõikide skriptide) töö.

Andurid

Andurid võimaldavad saada informatsiooni mõnede sündmuste ja objektide omaduste kohta. Nende kasutamine toimub gruppi **Andurid** kuuluvate käskude kaudu.

Puuteandurid

Käivitatakse käskudega: <puudutab objekt>, <puudutab värvi ...> ja <värv ... puudutab värvi ...>. Teevad kindlaks sündmused: antud objekt (mille skriptis esineb vastav käsk) puudutab teist objekti, objekt puudutab teise objekti või lava määratud värviga osa, või antud objekti määratud värviga ala puudutab teise objekti või lava määratud värviga osa.



Näide. Lava ülemises servas liigub edasi-tagasi värviline märklaud. Kasutaja üritab tabada seda lava alumises servas asuva noolega. Nool läheb lendu, kui vajutatakse klahvile **tühik**.

Saadavate punktide arv märklauda tabamisel sõltub värvist, mida puudutab nool. Punase värvi korral saab 10 punkti, sinise korral – 5 punkti, rohelse korral – 2. Kui nool läheb mööda, võetakse 2 punkti maha.

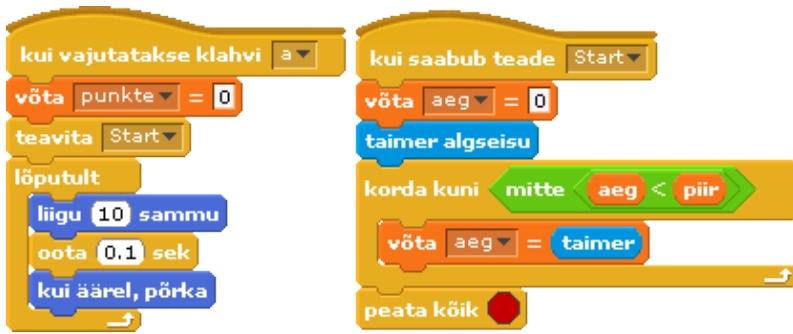


Noole lendu juhtiv skript on jagatud kahte ossa. Põhiskript hakkab tööle, kui vajutatakse klahvi **tühik**. See suurendab noole y-koordinaati ja kontrollib kahte tingimust.

Kui nool puudutab ülemist serva, oli tegemist möödalasuga: võetakse 2 punkti maha ja nool viiakse alumisse serva.

Kui aga nool enne puudutab märklauda, käivitatakse alamskript, mis

teeb kindlaks, millist värvi tabas nool ja suurendab vastavalt punktisummat. Täitmisjärg läheb automaatselt tagasi põhiskriptile, mis viib noole alumisse serva ja lõpetab skripti töö.



Vasakpoolne skript juhhib märklauda edasi-tagasi liikumist.

Parempoolne skript kontrollib mängu aega. Erinevalt eespool toodud näitest, kus aega kuvati kahanevalt, kuvatakse siin seda kasvavalt – alates nullist.

Kauguse ja omaduste andurid

Kauguse andur võimaldab teha kindlaks antud spraidi kauguse teisest spraidist (vt näide). Omaduste andur võimaldab spraidi või lava skriptil saada teada antud spraidi omaduste ja lokaalsete muutujate väärtusi. Selleks kasutatakse grupi **Andurid** plokki *<omadus kujundil sprait>*. Kättesaadavad on järgmised omadused: **x asukoht, y asukoht, suund, kostüümi #, suurus, helitugevus**. Spraidi skriptis võib saada teada lava tausta numbri.



Käsk omistab muutujale **XP** spraidi **pall** omaduse **x asukoht**

Näide. **Kraps** jalutab **palliga** edasi-tagasi. **Pall** pöörleb vastavalt liikumise suunale. **Krapsu** teel on **kast**. Kui **Kraps** jõuab **kasti** lähedusse, hüppab ta sellest üle.

Krapsu skriptis käivitab käsk [teavita **Tule**] palli vastava skripti. See viib **palli Krapsu** juurde, positioneerib ja määrab selle pöörlemise suuna, arvestades **Krapsu** liikumise suunda. Viimane tehakse kindlaks omaduste anduri abil, kasutades plokki [suund kujundil **Kraps**]. Väärtus omistatakse muutujale **s**, seda kasutatakse antud skripti **kui**-plokkis ja ka skriptis **Mine**.



Krapsu skripti **kui**-plokkis kontrollitakse pidevalt kaugust **kastist**. Kui see saab väiksemaks 40-st, käivitab käsk [teavita **Hypa**] vastava **Krapsu** skripti. Enne hüpet käivitab **Kraps** käsuga [teavita **Mine**], palli skripti, mis viib palli **kastist** mööda. Skriptis kasutatakse muutuja **s** väärtust, mille fikseerib skript **Tule**. Peale hüpet käivitab **Kraps** **kasti** asukoha muutmise skripti **Muuda**, milles kasutatakse **Krapsu** ja palli omadust **x asukoht**.

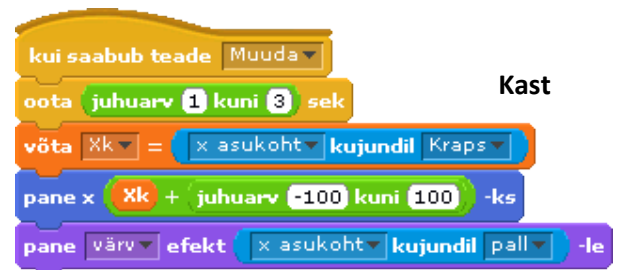
Kraps



Pall



Kast



Muud andurid

Scratch võimaldab kasutada spetsiaalseid andurite [paneele](#) (*Scratch Board* ja *Picoboard*) ja Lego [WeDo](#) komplekte. Nendega seotud küsimusi antud materjalis ei käsitleta.

Muutujad, loendid ja skriptid

Neid objekte vaadeldakse eraldi materjalides: [Muutujad](#), [Skriptid](#) ja [Loendid](#)

Kasutaja objektid

Projekti loomise käigus, kasutades olemasolevaid spriteid ja teisi objekte ning nende omadusi ja meetodeid, luuakse praktiliselt uued objektid. Nende nn tuletatud objektide omadused põhinevad baasobjektide (eeskätt spriteide) omadustel. Uute objektide mõned omadused või nende algväärtused määratakse loomise (disaini) faasis. Tüüpiliselt on selleks suurus, asukoht, kostüümide valik. Spriteide jaoks loodavad skriptid, milleks kasutatakse spriteide standardmeetodeid (käsuplokke) ja juhtimiskäskke, kujutavad sisuliselt nende uusi meetodeid. Vt ka [Modelleerimine](#).

Võtame näiteks harjutuse „Kraps, pall ja papagoi“: [projekt](#). Siin on kolm põhiobjekti: **Kraps**, **pall** ja **papagoi**. Vaatame lähemalt objekti **Kraps**.

Kraps
nimi
suurus
x_asukoht
y_asukoht
suund,
keere
tervitamine()
liikumine edasi-tagasi() palli kutsumine()
paremale()
vasakule()
tabamine()
möödavise()

Klassisümbolis on toodud nimetatud objekti põhiomadused ja meetodid. **Krapsul** on kaks kostüümi ja temaga on seotud üks heliklipp, mis siin ei kajastu. Allpool on toodud ka Krapsu skriptid: meetodid.

Omadused: **nimi**, **suurus**, **y asukoht** määratakse loomise faasis ning need ei muutu projekti täitmise ajal. Omadusi: **x asukoht**, **suund** ja **keere** muudetakse vastavate skriptide töö ajal.

Jaotamine meetoditeks on siin ja ka järgnevates skeemides mõnevõrra tinglik. Üks skript võib määratleda ühe või mitu meetodit. Soovi korral võib skripti jagada käskude [Teavita] abil mitmeks skriptiks (meetodiks). Üldiselt on tähtis tuua välja objekti olulisemad tegevused.

Krapsu skriptid

Papagoi
nimi, suurus, x_asukoht, y_asukoht, suund, nähtavus, värv
ringliiklus() narrimine() tabamine() peitmine(), värvi muutmine() möödavise() naermine(), pikeerimine()

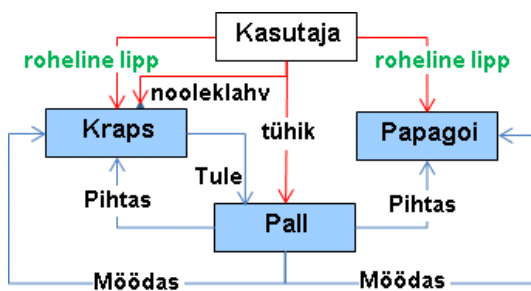
Pall
nimi, suurus, x_asukoht, y_asukoht, nähtavus
visete loendamine() oleku muutmine() liikumine ülesse() tabamuse kontroll() tabamuste loendamine() horisontaalne liikumine() muutujate nullimine()

Siin on toodud objektide **Papagoi** ja **Pall** omadused ja meetodid. Vastavaid skripte võib näha Scratchi [projekti](#).

Seosed objektide vahel

Üldjuhul kujutab Scratchi projekt omavahel seotud ja koos toimivate objektide kogumit. Üheks peamiseks vahendiks seoste realiseerimisel on **teate** (signaali) **saatmine** mingi objekti (võib olla sprait või lava) skriptist käsuga [**teavita nimi**] ja selle **vastuvõtmine** ühe või mitme objekti skripti poolt, mis algab päiseploki [**kui saabub teade nimi**]. Selle tulemusena käivituvad skriptid, mille päiseplakis on teatega määratud **nimi**. Informatsiooni vahetamiseks objektide vahel võib kasutada **globaalseid muutujaid** ja **loendeid** ning **omaduste andurit** [*omadus kujundil objekt*]. Viimase abil saab üks objekt küsida teise objekti omaduste ja lokaalsete muutujate väärtusi. Vt jaotist [Kauguse ja omaduste andurid](#).

Näiteks kasutame harjutust „Kraps, pall ja papagoi“: [projekt](#). Projekti on kolm põhiobjekti: **Kraps**, **papagoi** ja **pall**.

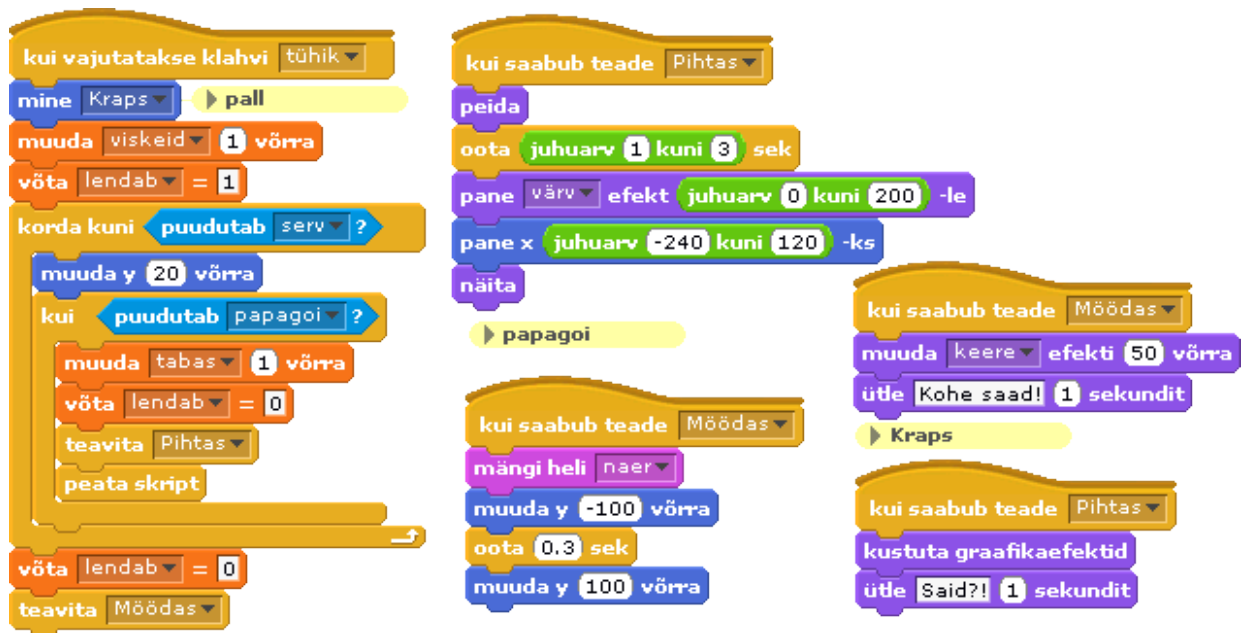


Kui kasutaja klõpsab rohelist lippu, hakkavad paralleelselt tööle **Krapsu** ja **papagoi** põhiskriptid. **Nooleklahvidega** saab kasutaja muuta **Krapsu** liikumise suunda ning klahviga **tühik** käivitada **palli** lendu juhtiva skripti.

Kui pall ei lenda, väljastab **Krapsu** skript pidevalt teadet **Tule**, mille võtab vastu palli vastav skript. See viib palli **Krapsu** juurde, positioneerib **Krapsu** suhtes ja paneb palli pöörlema, arvestades **Krapsu** liikumise suunda. Viimase

teeb skript kindlaks omaduste anduri abil. Asjaolu, kas pall lendab või mitte, fikseeritakse palli skriptis globaalse muutuja **lendab** väärtuse abil, mida kasutatakse **Krapsu** skriptis.

Igal täitmisel väljastab palli skript sõltuvalt tulemusest teate **Pihtas** või **Möödas**. Teate võtavad vastu **Papagoi** ja **Krapsu** vastavad skriptid.



Võiks vaadata ka jaotist „Rakenduste loomise põhiastapud“ materjalis [Rakendused](#).