

# Programmēšanas vide jebkuram skolēnam – Scratch multimediju iespējas

## jeb datorprogrammēšanas pamati dažās stundās

Palīdzošo materiālu sagatavojis LU docents Imants Gorbāns, Rīga, 2014.

### Saturs

Priekšvārds.....	1
1. Mikrofona izmantošana <i>Scratch 2</i> .....	2
2. Tīmekļa kameras lietošana <i>Scratch 2</i> .....	4
3. Programma ar tīmekļa kameru <i>Begemots v.1</i> darbībā.....	5
Literatūra.....	7



Visi Latvijas Universitātes Datorikas fakultātes [Jauno datorīku skolas materiāli](#) ir licencēti. Autors Imants Gorbāns šo materiālu ir licencējis ar [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported](#) licenci.

### Priekšvārds

Šis ir otrais konspekts LU DF projekta “Jauno datorīku skola” (JDS) ietvaros. Ja pirmajā konspektā tika aplūkoti *Scratch* pamati, tad šajā konspektā galvenais akcents ir vērsts uz *Scratch 2.0* multimediju iespējām – darbu ar mikrofonu un tīmekļa kameru (*Webcam*).

Ir pienācis laiks pievērst uzmanību 21. gadsimta mācīšanās vienai no inovācijām – tīmekļa resursu jēgpilnai izmantošanai un sadarbībai, kopdarbam mākoņservisos. Ar šo domu ir izveidota JDS *Scratch* studija <http://scratch.mit.edu/studios/470085/>, tajā atrodamas šajos konspektos aplūkotās programmas, iespējams pētīt to kodu. Katras JDS materiālu <http://jds.df.lu.lv> tēmas apakša ir ievietots resurss “Citu *Scratch* kopienas dalībnieku ... piemēri”, paskatiet šos daudzus jau pasaulē eksistējošos piemērus, lielus projektus. Noderīgi kopienas piemēri: <http://scratch.mit.edu/studios/254029/> - London South Bank University projekta programmu piemēru un sagatavju kolekcija "LSBU CPG 2013-2014"; <http://scratch.mit.edu/studios/237442/> - Tīmekļa kameras spēļu studija "Webcam Games"; <http://scratch.mit.edu/studios/26471/> - cita matemātikas studija u.c.

Publicējiet <http://scratch.mit.edu> vietnē savas labākās programmiņas, spiežot programmas rediģēšanas loga labajā augšējā stūrī koplietošanas pogu “Share”. Tas dos iespēju citiem redzēt jūsu programmu kodu, to vērtēt, rakstīt komentārus, remiksēt (kopēt sev pārveidošanai). Tā mēs varam viens otru bagātināt un darīt pasauli labāku. Publicētās programmas varat pievienot JDS dalībnieku programmu studijai <http://scratch.mit.edu/studios/475208/>.

Mācoties programmēt ar *Scratch*, der piedomāt pie izstrādes domu cikla: **Iztēloties** (ko es gribu izveidot) -> **Radīt** (algoritms jeb koda ideja un tās programmēšana) -> **Eksperimentēt** (manuāla testēšana dažādos variantos) -> **Koplietot** (strādājošu programmu var publicēt tīmeklī) -> **Atspoguļot** (pārdomāt, pēc dažām dienām lietas mēdz izskatīties jaunā gaismā, palasīt atsauksmes, pieiet pozitīvistiski un paškritiski reizē) -> **Iztēloties** (izdomāt uzlabotu programmas versiju). Ja strādājat tīmeklī, savas programmas varat lejupielādēt datorā (fails ar paplašinājumu .sb2).

Kā jau pirmajā konspektā bija teikts, *Scratch* ir vienkārša programmēšanas valoda un vide, arī multimediju radīšanas rīks, ko var izmantot, lai iepazītos ar programmēšanu. *Scratch* ir noderīga

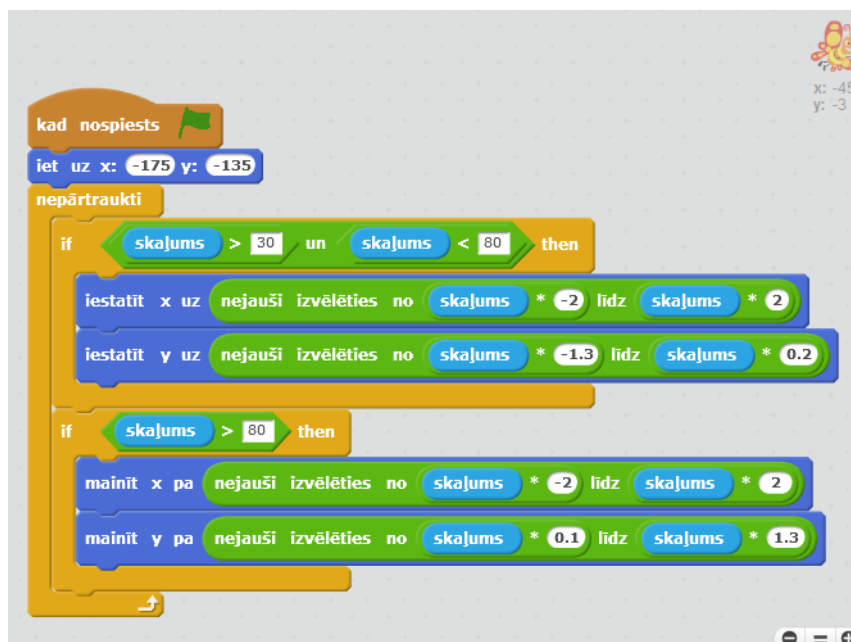
kā vidusskolēniem, tā pamatskolēniem, to var lietot dažādos sarežģītības līmeņos – no pirmo klašu skolēna līdz pirmā kursa studentiem kā pirmie soļi programmēšanā.

**SVARĪGI!** Ja vēlaties izmantot atraktīvās mikroфона un tīmekļa kameras programmēšanas iespējas, izmantojiet jaunāko versiju – *Scratch 2.0*. Ja *Scratch 2.0* programmēšanas rīku izmantojat interneta pārlūkprogrammā, tai ir jābūt ne vecākas versijas kā *Chrome 7*, *Firefox 4*, *Internet Explorer 7* ar *Adobe Flash Player 10.2* vai jaunāku. Ja pats *Scratch* programmēšanas rīks vairumā gadījumu darbosies arī ar dažiem alternatīviem, “vieglākiem” *Flash* pārlūkprogrammas spraudņiem, tad **darbam ar mikrofonu un tīmekļa kameru gan ir nepieciešams pats *Adobe Flash Player* tā jaunākajā versijā (vismaz 10.2)**. Pirms sāciet šajā konspekta sadaļā aprakstītos darbus, pārlicinieties, ka datora mikroфons un tīmekļa kamera ir darboties spējīgi, ieslēdzami; to darbību var pārbaudīt, piemēram, ar populāro programmu *Skype* (ieteikums vairākās programmās vienlaicīgi mikrofonu un kameru neizmantojot). Ja uz *Windows 7/8/8.1* ir problēmas ar šīm multimediju ierīcēm, risinājums meklējams *Control Panel* tādās sadaļās kā *Sound*, *Device Manager (Sound, video and game controllers, Imaging devices)* vai *Devices and Printers*; kā arī dažos gadījumos antivīrusa vai ugunsmūra programmatūra var neļaut pārlūkprogrammai un tās *Flash* spraudnim strādāt ar mikrofonu un kameru, tad šīs atļaujas ir jādod (*Allow*). *Scratch 2.0* ir veidots ekrāna izšķirtspējai vismaz 1024 x 768, bet šis pēdējais ir tikai ieteicamais noteikums.

## 1. Mikroфона izmantošana Scratch 2

Kā jau bija minēts 1. tēmas konspektā "Programmēšanas vide jebkuram skolēnam – Scratch jeb datorprogrammēšanas pamati dažās stundās" - <http://skolas.lu.lv/mod/book/view.php?id=29857>, *Scratch 2* māk izmantot tikai 2 tipu skaņas failus: WAV un MP3. Fona mūzikas vai gariņu izdoto skaņu pievienošana programmai ir īsi aplūkota minētajā 1. tēmas konspektā, tādēļ šeit pievērsīsimies sarežģītākai idejai- izmantot mikrofonu kā sensoru jeb skaņas detektoru.

Aplūkosim vienkāršu programmiņu, ar kuras palīdzību var saprast, kā darboties ar mikrofonu: <http://scratch.mit.edu/projects/23691519/> vai redaktora skatā <http://scratch.mit.edu/projects/23691519/#editor> - *Taurenītis v.1.1* - darbināms piemērs *Scratch 2.0* programmiņai ar skaņas detektēšanu no mikroфона (pārlūkprogrammā parādās brīdinājums: *Adobe Flash Player Settings - Camera and Microfone Access...* - jānospiež poga *Allow*). Lai taurenītis lidotu, radiet skaņu, kas skaļāka par 10 vienībām, lai lidotu augstu - virs 80 nosacītajām vienībām (tās ir diapazonā 0...100 vienības, tās var uzskatīt par % no maksimālā skaļuma). Mikroфона lietošanas programmas kods dots 1. attēlā. Lai izmantotu mikroфона programmēšanas iespējas, izmantojiet jaunāko versiju – *Scratch 2.0*; Ja *Scratch 2.0* programmēšanas rīku izmantojat interneta pārlūkprogrammā, tai ir jābūt ne vecākas versijas kā *Chrome 7*, *Firefox 4*, *Internet Explorer 7* (vienmēr visur lietojiet pārlūkprogrammu jaunākās versijas!) ar *Adobe Flash Player 10.2* spraudni vai jaunāku.



1. att. Skaņas detektēšanas ar mikrofonu programmas *Scratch 2.0* kods.

Kā viegli ieraudzīt programmas kodā, tā, kā parasti, ir palaižama ar zaļo karodziņu. Programmas scenārijā ir 2 lielas sadaļas:

- 1) gariņa novietošana sākumstāvoklī – apakšējā kreisajā darbību laukuma stūrī (*iet uz x: -175 y: -135*),
- 2) beznosacījuma cikls – lielais dzeltenais koda bloks (*nepārtraukti*); ja programmas pārtraukums kodā nav paredzēts, to tomēr var pārtraukt ar sarkano pogu.

Programmas scenārija 2. lielā sadaļa satur divus nosacījuma jeb zarošanās blokus (*if*), tieši nosacījumos ir iekļauta programmas sadarbība ar mikrofonu (*skajums*):

- a) ja *skajums* ir no 30 (varbūt te liekam 10, ja klusi noregulēts mikrofons, pārāk maza skaitļa izvēle tiks taurenītim reaģēt uz fona troksni) līdz 80 %, tad izpildās gariņa pārvietošana gan pa x, gan y asīm; tā kā *skajums* ir < 100, bet darbību laukums ir lielāks (-240 līdz +240 pa x asi un -180 līdz +180 pa y asi), tad *skajuma* skaitli reizināsim ar kādu skaitli, kas tauriņam dod iespēju aptvert mazliet vairāk par ekrāna apakšējo pusi;
- b) ja *skajums* ir lielāks par 80, liksim tauriņam lidot ekrāna paša augšdaļā.

Lai izveidotu šādu kodu, vispirms lielajā ciklā ievelkam “*if*” operatoru. Tad tam ir jānorāda nosacījums; tā kā nosacījums sastāv no 2 daļām vispirms atrodam apvienojuma koda bloku “*un*”, tad tā abu argumentu vietās (baltie kvadrātiņi) ievelkam salīdzināšanas operatorus “*>*” un “*<*”. Savukārt tā argumenti ir no mikroфона detektētais “*skajums*” un ierakstāms skaitlis, ar ko to salīdzināt. Pievēršam uzmanību, ka operatori jeb koda bloki *Scratch* ir ne tikai dažādas krāsas (pa tematiskajām grupām), bet arī dažādas formas. Ja operatora “*if*” nosacījuma vieta ir sešstūris, tad skaidrs, ka tajā varēs novietot tikai sešstūrveida operatora klucīti. Darot citādi, klucīši neiekritīs vietā jeb nesakrēsies, kas nozīmē, ka tā nevar. Šādi *Scratch*, kā visi mūsdienīgi programmēšanas rīki, palīdz programmētājam neveidot neizpildāmu kodu.

Veidojot gariņa novietošanas izteikumu, vispirms atrodam un ievelkam zem “*if*” bliku “*iestatīt x uz*”, tad “*nejauši izvēlēties no... līdz...*”, lai taurenītis lidotu ne tikai atkarībā no balss *skajuma*, bet arī ar nejaušības piedevu. Nejaušas izvēles operatora argumenti lodziņos vispirms ievelkam reizināšanas operatoru “*\**” (zvaigznītes izskata), tad tajā ievelkam

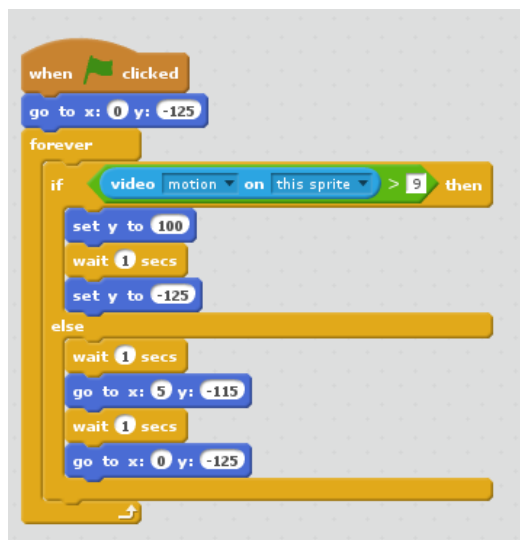
“skaļums”, kas ir operators, kas lasa mikrofona ieeju (ja angļiski, tad “sensing” bloku kopā “loudness”), un norādām skaitli, ar kuru reizināt. Decimālzīme te ir punkts (nevis komats). Jūtieties brīvi šos nosacījumus modificēt.

UZDEVUMS. Padomājiet, kā šo programmiņu pilnveidot! Piemēram, papildiniet to ar spārnu vicināšanas efektu, šim nolūkam jāizvēlas gariņš ar vismaz 2 tērpiem, piemēram, sikspārnis (*Bat1*), tad ieprogrammējiet tērpu (*costumes*) maiņu katra kustības posma vidū (operators *switch costume to bat1-b*).

## 2. Tīmekļa kameras lietošana Scratch 2

Otra multimediju iespēja ir tīmekļa kameras jeb “webkameru” lietošana *Scratch 2* programmās. Šī iespēja ir ļoti jaudīga, jo ļauj veidot pat spēles, kurās spēlētājam ir fiziski jākustas. Var izveidot programmu, kas ir mazliet līdzīga *Microsoft Xbox 360* vai *Xbox One* ar *Kinect* konsoli spēlēm. Kaut, protams, ar *Scratch* iespējamais rezultāts šeit ir pieticīgāks nekā profesionālu veidotas spēles, milzīga nozīme ir iespējai izveidot ko radoši savu. Citās skolās izmantojamās programmēšanas valodās tik vienkārši ar mikrofoni un videokameru darboties nav vai, tipiski, vispār nav iespējams.

Tīmekļa kameras izmantošanas idejas saprašanai, aplūkosim vienkāršu programmiņu <http://scratch.mit.edu/projects/23712067/>, kas rediģēšanas skatā nonāk pēc “*See inside*” nospiešanas: <http://scratch.mit.edu/projects/23712067/#editor>. Šai programmiņai būtiski, ka daļa koda ir gariņam, bet daļa – fonam jeb skatuvei. Tā kā man trūkst talanta uzzīmēt lidojošu cūku, izvēlējos no *Scratch* piedāvājuma lidojošu begemotu, vai arī nīlzirgu, šī gariņa kods dots 2. attēlā. Dažādībai 2., 3. attēls ir ņemts no programmas redaktora, kas iestatīts angļu valodā, bet 4., 5. att. - tas pats kods pēc redaktora pārslēgšanas latviešu valodā.

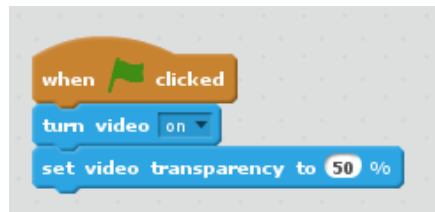


2. att. Nīlzirga gariņa kods.

Kā redzams 2. attēlā, vispirms novietojam nīlzirga gariņu izvēlētajā sākumstāvoklī, tad iedarbinām ciklu (to var pārtraukt ar sarkano pogu). Gariņa kustības organizēsim ar zarošanās operatoru “ja-tad-citādi” jeb “if-then-else”, sakot: ja video ir kustība uz/zem gariņa vairāk kā 9 sīknotikumiem (izvēlamies eksperimentāli, lai nereaģē uz fona “troksni”), tad darām kaut ko. Tas kaut kas ir “set y to 100”, kas liek nīlzirgam palēkties. Pirms nokrišanas liekam gariņam 1 sekundi karāties gaisā, lai darbības nenotiek tik ātri, ka videokarte nespēj labi parādīt un acs saskatīt. Operatora “if-then-else” blokā zem “else” būs pastāstīts, ko darīt, ja nav detektēta video

kustība, šajā gadījumā ir likts nīlzirgam mazliet lēkāt uz vietas, kā gatavojoties lielajam lēcienam.

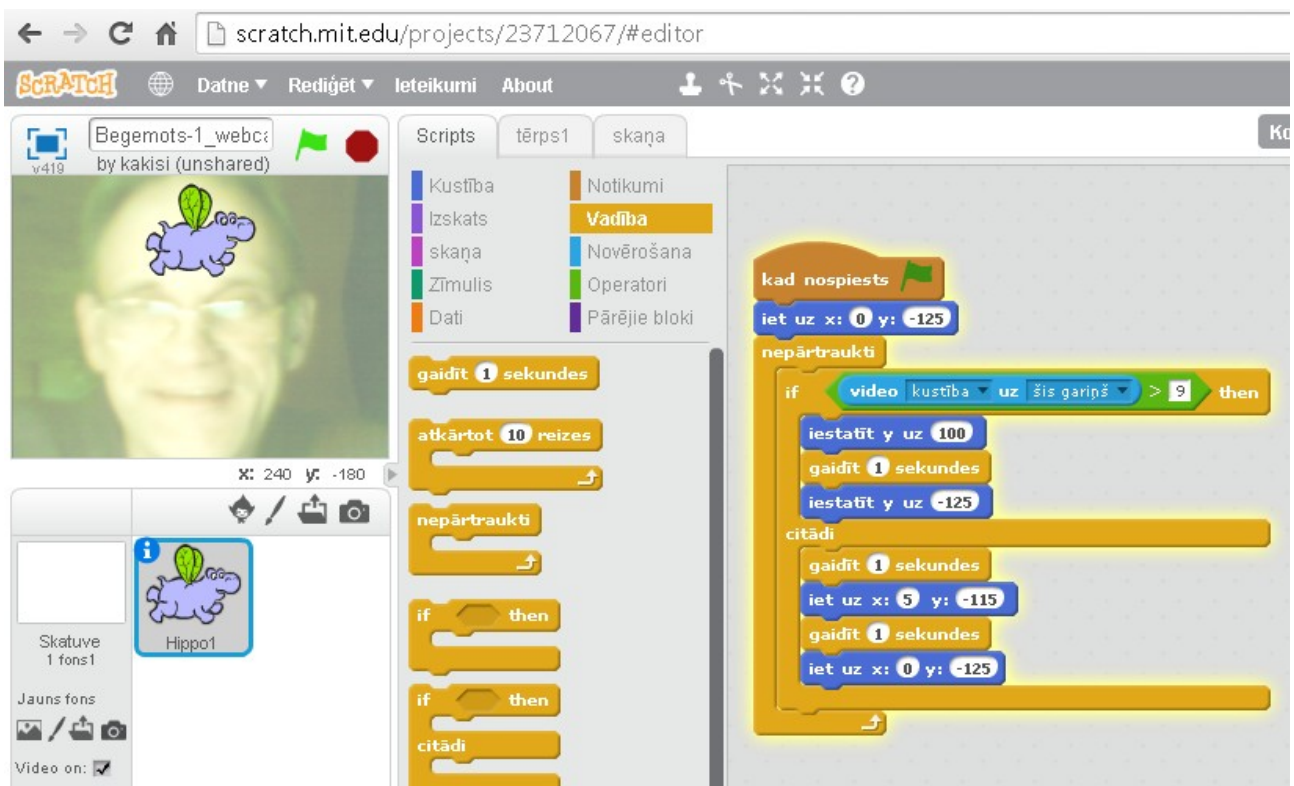
Tomēr, lai iepriekšējā rindkopā aprakstītais strādātu, *Scratch* ir jāredz video, no kura detektēt kustību. Ir izdomāts šo video rādīt fonā, jeb uz skatuves. Video liekam par 50% caurspīdīgu (ja video redzams uz ekrāna, varam saprast, ka kamera strādā), var pat pilnībā caurspīdīgu, galvenais, lai tas tehniski tur būtu, kodu skat. 3. attēlā. Kāpēc tā? Nu tāpēc, ka *Scratch* izstrādātāji tā ir paredzējuši darbu ar video, kas ir vienkāršākais iespējamais variants. Vai video blokus vārēja likt gariņam, ne skatuvei? Vispār jau jā, darbosies arī tā, bet tas nebūs optimāli, jo tad visai lielus datora resursus tērējošās video lietas slogos gariņu. Tas var traucēt gariņa darbībai gara koda gadījumā, piemēram, spēlēs. Labāk vienam objektam (gariņam) nelikt tērēt lielus resursus, bet šo slodzi sadalīt starp vairākiem objektiem, šoreiz šis otrs objekts ir skatuve. Ja video nedarbojas, vēlreiz izlasām šī konspēkta priekšvārda pēdējo rindkopu un atrisinām ārpus *Scratch* esošo problēmu.



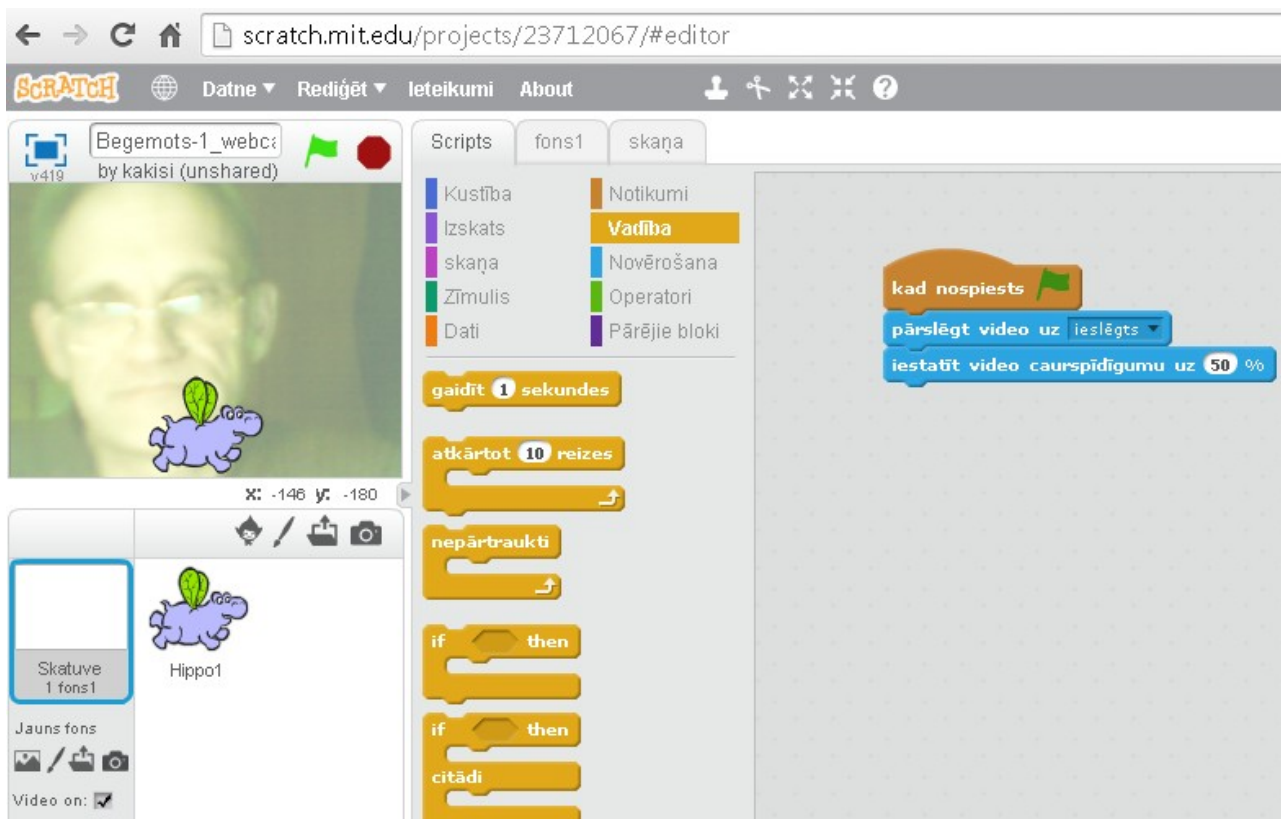
3. att. Fona jeb skatuves kods, kurā izsauc tīmekļa kameras video.

### 3. Programma ar tīmekļa kameru Begemots v.1 darbībā

Šeit ir svarīgi saprast, kad ir gariņu joslā (lejā pa kreisi) aktīvs gariņš (4. att.), kad skatuve (5. att.). Programmas darbības piemēri ekrānattēlu veidā doti 4. un 5. attēlā.



4. att. Nīlzirga kods un programma darbības laikā ar video fonā.



5. att. Programmas fona kods, video kustības nav.

Paspēlējiet jds.df.lu.lv 2. tēmā zemāk doto interaktīvo *Scratch* spēlīti, kurā varat padejot tīmekļa kameras priekšā, spēlīte sniedz jūsu kustību intensitātes vizuālu vērtējumu un arī rāda punktus; spēlīte atbilstoši *Scratch* kopienas labajam stilam, ir legāli pārveidota jeb remiksēta cita autora CrazyNimbus programma (to esmu papildinājis ar labāko rezultātu parādīšanu, lai var sacensties, kā arī video pārliku no gariņa uz skatuvi, un pievienoju mūziku). PROGRAMMAS PIEMĒRS: Remiksēta "Dance your way to stardom - WEBCAM REQUIRED" – <http://scratch.mit.edu/projects/24340268/>; nesēdi, palēkā, tad izpēti kodu! Programmas kods ir visai viegli saprotams, varbūt mazāk pamanāma ir komanda "broadcast (message)" jeb "pārraidīt (message)" un "when I receive (message)" jeb "kad es saņemu (message)". Šo komandu argumenti ir lejupizkrītošajās izvēlnēs, tajās vai nu izvēlas esošu ziņojumu, vai veido jaunu unikāla nosaukuma ziņojumu. Šāda veidā ar ziņojumiem gariņiem var likt savā starpā vai ar skatuvi apmainīties ar informāciju par notikumiem. Piemēram, viens gariņš pēc video kustības saņemšanas izmaina kāda mainīgā vērtību un nosūta ziņojumu, bet otrs gariņš, ir ieregulēts mainīt tērpa attēlu, ja tāds ziņojums tiek saņemts.

Tad vēl jāatzīmē, ka datu komandu bloki parādās tikai pēc tam, kad bloku sadaļā "Data" ir izveidots mainīgais (*Make a Variable*) vai saraksts/ viendimensijas masīvs (*Make a List*). Lai datu mainīgie būtu redzami darbību laukumā, "Data" sadaļā ieliek ķeksi mainīgā vai saraksta priekšā; vajadzīgajā vietā tos noliek ar peli.

Nu ko, paskatiet iedvesmai arī *Scratch* kopienas dažādus citu autoru projektus (piemēru saraksts <http://skolas.lu.lv/mod/page/view.php?id=30170>, iesaku <http://scratch.mit.edu/studios/237442/> u.c.) un veidojiet savas oriģinālas, asprātīgas programmiņas, ieprieciniet ar tām savus līdzcilvēkus un pasauli!

UZDEVUMS. Izveidojiet spēlīti, kurā begemots jāiedzen krātiņā ar pēc iespējas mazāku roku kustību skaitu.

## Literatūra

1. Jerry Lee Ford, Jr. Scratch 2.0 Programming for Teens. Course Technology, 2014, USA. 337 lpp. (<http://books.google.lv/>, 2014, 99 lpp.).
2. Getting started with Scratch version 2.0 - [http://cdn.scratch.mit.edu/scratchr2/static/\\_f26c59f090fca8b69aaa1e5f27a3dd2f\\_\\_//pdfs/he lp/Getting-Started-Guide-Scratch2.pdf](http://cdn.scratch.mit.edu/scratchr2/static/_f26c59f090fca8b69aaa1e5f27a3dd2f__//pdfs/he lp/Getting-Started-Guide-Scratch2.pdf). 2014. 16 lpp.
3. Super Scratch Programming Adventure!: Learn to Program By Making Cool Games. edited by The LEAD Project. San Francisco, CA: No Starch Press, 2012.
4. Brandon Milonovich. Scratch Cookbook. PACKT Publishing Ltd., 2013. 262 lpp. (<http://books.google.lv/> 41 lpp.).
5. Scratch apmācītāju lapa- <http://scratched.media.mit.edu/>, 2014.
6. Jessica Chiang. Shall We Learn Scratch Programming. 177 lpp., 2013.
7. Scratch grāmatas - <http://www.scratchguide.com/> 2014.
8. Letonika Terminoloģijas portāls – <http://termnet.lv> , 2014.
9. Majed Marji. Learn to Program with Scratch. A Visual Introduction to Programming with Games, Art, Science, and Math. No Starch Press, USA, San Francisko, 2014. 291 lpp.

Veiksmīgu darbu un izklaidi!  
25.06.2014., papildināts 11.07.2014.

Dr. Vadībzin., Doc. Imants Gorbāns