

Programmēšanas vide jebkuram skolēnam – Scratch jeb datorprogrammēšanas pamati dažās stundās

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

Saturs

Priekšvārds.....	1
1. Kas ir programmēšanas valoda <i>Scratch</i> ?.....	2
2. Tehnoloģiskais ievads <i>Scratch</i>	2
2.1. <i>Scratch</i> apguves uzsākšana interneta pārlūkprogrammā.....	3
2.2. <i>Scratch</i> instalācijas sagatavošana datorā.....	5
3. Sākam darbu ar <i>Scratch</i>	7
3.1. <i>Scratch</i> vide	7
3.2. <i>Scratch 2</i> valodas un vides īss apskats ar piemēru.....	8
3.3. Iesāktās programmiņas pilnveidošana.....	12
3.4. Darba pabeigšana un nākamās aktivitātes.....	15
4. Vēl dažas no <i>Scratch 2</i> daudzajām iespējām.....	16
5. Sakārtosim domas.....	18
Literatūra.....	19



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

Jūs noteikti esat dzirdējuši vārdu “programmēšana”, nojaušat, ka tā ir datorprogrammu radīšana, varbūt jūs gribētu ar to nodarboties, to studēt, bet nezināt, vai jums tas padosies un patiks. Programmu radīšana ir citādāks darbības veids nekā parasta to lietošana, ko esat vairāk vai mazāk apguvuši. Izvēloties studiju jomu un līdz ar to savu nākotnes profesiju, var izvēlēties arī ko sev svešu, nepazīstamu, pietiek ar priekšstatu. Lūk, tieši šis konspekts palīdzēs jums ātri un bez milzīgas piepūles gūt priekšstatu par to, kas ir programmēšana, tas palīdzēs jums nojaust, vai programmēšanas un datorzinātņu studijas varētu jums interesēt. Lai tā notiktu, jums ir ne tikai jāizlasa šis konspekts, bet arī praktiski jāizpilda izstāstītie vingrinājumi, tos veikt var palīdzēt līdz ar konspektu dotie video materiāli.

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 pirmo kursu studentiem. **Šis *Scratch* pamācība ir rakstīta ar mērķi piedāvāt vienkāršu un viegli patstāvīgi īsā laikā (dažās stundās) apgūstamu rīku skolēniem, kuriem skolā nav bijusi programmēšana, vai tā apgūtaniecīgā apjomā. Tas varētu palīdzēt pašiem saprast, vai jums programmēšanas abstraktā domāšana interesē un padodas, vai ir pamatota vēlme studēt programmēšanu Latvijas Universitātes Datorikas fakultātē vai citur. Tāpat šīs instrukcijas var būt noderīgas kā topošajiem tā jau strādājošiem informātikas skolotājiem,** gatavojoties programmēšanas stundām, kurās radīt atraktīvu dažādību valodas *Pascal* utml. matemātiskajam skaistumam, pielietojumu šaurumam un vizuālajai garlaicībai.

Spēja vismaz vispārīgi saprast datorprogrammu kodu ir svarīga daļa no mūsdienu, 21. gadsimta kompetencēm (zināšanām, prasmēm, iemaņām, izpratnēm), tā mūsdienu tehnoloģiskajā sabiedrībā sabiedrībā papildina lasītprasmi. Kad cilvēki mācās programmas pirmkodu, viņi mācās

svarīgas stratēģijas (kā izsenis, spēlējot šahu, lasot filozofu darbus, apgūstot cīņas mākslas), kas noderēs, lai atrisinātu reālās dzīves problēmas, izstrādātu projektus, veiksmīgi prezentētu idejas.

1. Kas ir programmēšanas valoda Scratch?

Ar *Scratch* jūs varat pārvērst datorprogrammā jeb, vienkārši, programmā savus interaktīvos stāstus, spēles, animācijas un dalīties ar tiem tiešsaistes kopienā ar citiem. Būtiski, ka ar *Scratch* jūs varat izveidot un ieprogrammēt aktīvus, kustīgus objektus, saglabāt sava darba rezultātus tā, ka jūsu līdzcilvēki var pamēģināt tos lietot, paspēlēties ar jūsu veidoto spēlīti, kas ir motivējoši un bieži dod gandarījumu par paveikto. *Scratch* palīdz jauniešiem iemācīties domāt radoši, cēloņsakarības saskatoši, sistēmiski, kā arī ir iespēja strādāt grupā, tās ir būtiskas prasmes dzīvei 21. gadsimtā. *Scratch* ir pasaulslavenā Masačūsetas Tehnoloģiskā institūta (ASV, Kembridža, Masačūseta) mūžizglītības grupas *Lifelong Kindergarten* projekts (realizēts *MIT Media Lab*). *Scratch* galvenā interneta vietne ir <http://scratch.mit.edu/> (visas laboratorijas lapa ir <http://llk.media.mit.edu/>).



Scratch ir bez maksas jeb brīvlietojuma (ir brīvprātīga iespēja projektu atbalstīt ar ziedojumiem, kad tas iepatiksies) atvērtā pirmkoda programmatūra (licences: GPLv2 un *Scratch Source Code License*). *Scratch* logo nes ideju par visa sākšanu no nulles, dažādu ideju savienošanu un savu jaunpienesumu (kā mūzikas brīvā un radošā “skrečošanā”), to darot ar kaķa vieglumu un brīvdomību, *Scratch* simbols ir zelta kaķis.

Scratch ļauj lietotājiem radīt notikumu orientētu programmu ar vairākiem aktīviem objektiem, ko angļiski sauc par "*sprites*" (latviski tas varētu būt “gariņi” vai “elfi”, skat. tulkojumu Letonika Terminoloģijas portālā: <http://www.termnet.lv/Term.aspx?tabindex=1&search=sprite&indefiniton=False&subject=0&allak=False&language=0>). Gariņus var uzzīmēt jebkurā rastra grafikas vai vektorgrafikas rīkā vai pašā *Scratch* vidē, un ar *Scratch* piešķirt tiem dzīvību – kustību noteikumus, uzvedības algoritmus u.c. Zinātniski *Scratch* var pieskaitīt objektorientētās programmēšanas, arī vizuālās programmēšanas valodām. Nesatraucieties par dažiem šeit dotiem šķietami sarežģītiem vārdiem un teikumiem, šobrīd pietiek ar intuitīvu nojausmu, tie jums kļūs pašsaprotami ar laiku, ja studēsiet datorzinātnes.

Scratch ir jauna programmēšanas valoda, tās pirmā versija tika radīta 2003. gadā un tika mērķēta 8-16 gadus veciem skolēniem (pirmais izstrādātājs ir Mitchel Resnick, *MIT Media Lab Lifelong Kindergarten Group*). 1. versija publiskai pieejai tika nodota 2007. gadā, savukārt *Scratch 2.0* tika izlaista 2013. gada maijā, bet jau 2014. gadā varam apgalvot, ka tas ir pasaulē populārākais skolu modernās programmēšanas pamatu apguves rīks – to lieto vairāk kā 150 valstīs un plašākā vecumu diapazonā, tas ir pieejams vairāk kā 30 valodās.

Ir pienācis laiks arī Latvijā izbaudīt šīs inovatīvās, notikumu orientētās, samērā intuitīvi apgūstamās uz konstrukcionisma teorijas balstītās vides iespējas. *Scratch* ir programmēšanas valoda, kas ir nākamais solis aiz pazīstamajām skolu valodām *BASIC*, *Logo*, *Alice*, *Pascal*. Jaunais *Scratch* palīdzēs jums ar citu skatienu ieraudzīt vecās programmēšanas valodas, piemēram *Free Pascal*, *C++*, *PHP*, *ASPX*, *Visual Basic*, *C Sharp* u.c. *Scratch* var noderēt arī skolu projektu nedēļās, turklāt ne tikai matemātikas un dabaszinātņu, bet arī starpdisciplināriem, kā arī sociālo zinātņu un mākslas projektiem.

2. Tehnoloģiskais ievads Scratch

Scratch pamatus var apgūt dažās stundās, piemēram, 4, tā ka to var izdarīt pat pāris nedēļās starp izlaidumu un pieteikšanos studijām, lai izvēle būtu jēgpilnāka. Šis konspekts palīdzēs jums to

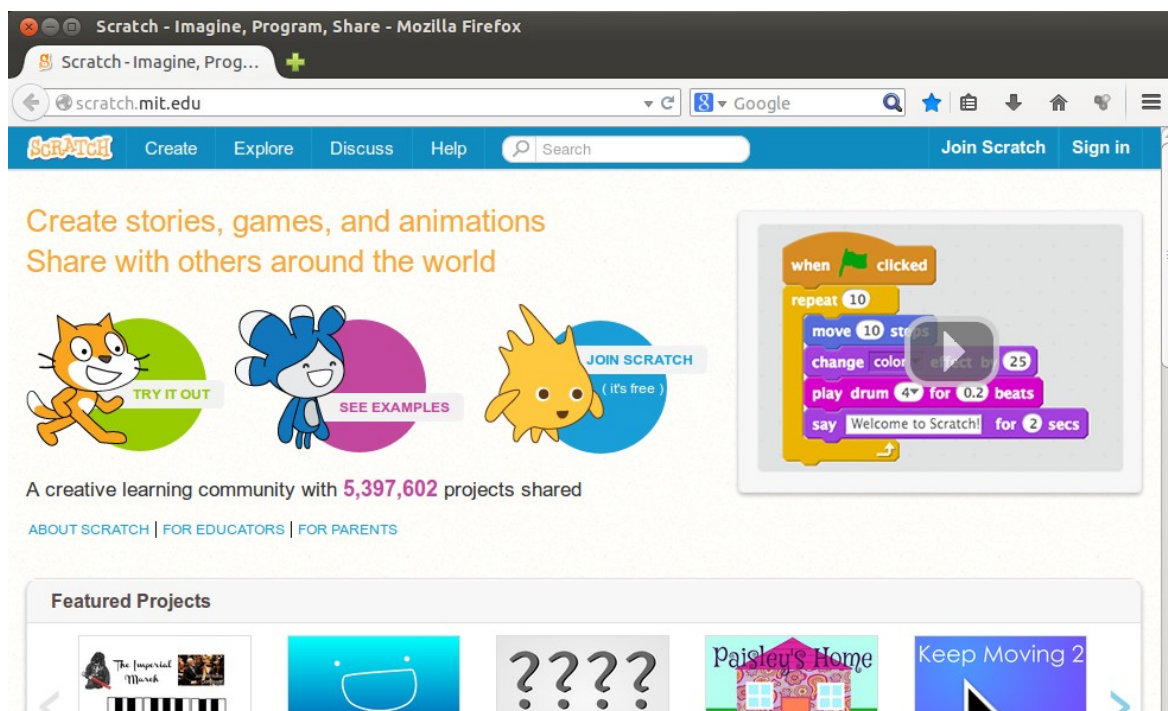
izdarīt. Aktuālā versija ir *Scratch 2.0*. *Scratch* lielo popularitāti un perspektīvu nosaka: a) tā ir jauna, izstrādātāju un lietotāju kopienas atbalstīta valoda, b) tas ir aktīvs projekts, c) *Scratch* ir lietojams (var programmēt) kā interneta pārlūkprogrammā, tā var lejupielādēt un instalēt uz *Microsoft Windows*, *Linux* un *Mac* datoriem un planšetēm, d) tā nav resursprasīga, līdz ar to izmantojama uz jebkura, arī 6 un vairāk gadus veca, skolas vai mājas datora (darbstacijas).

Nesaistes versijas instalēšanai datorā 2014. gada maijā ir pieejama *Scratch 2.0* beta versija un *Scratch 1.4* stabilā versija, tās var iegūt par brīvu un lietot bez reģistrēšanās *Scratch* tīmekļa vietnē. Datorā var būt instalētas abas versijas vienlaicīgi, jo tās tehnoloģiskajos pamatos ir pilnīgi atšķirīgas. Savukārt, tiešsaistes versijas lietošanai interneta pārlūkprogrammā ir pieejama stabilā versija *Scratch 2.0* (jeb *Scratch 2*).

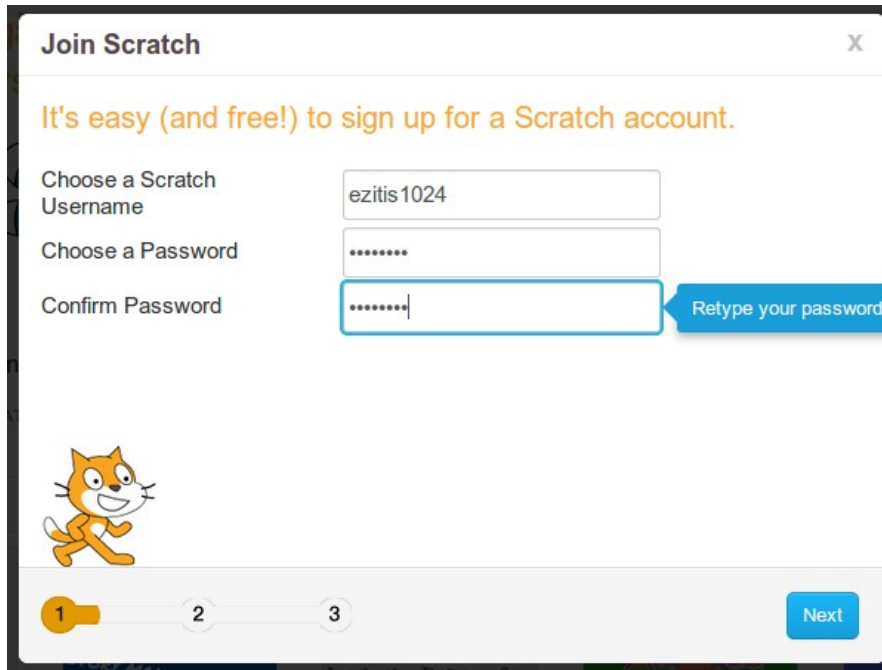
2.1. *Scratch* apguves uzsākšana interneta pārlūkprogrammā

Tīmekļa jeb tiešsaistes režīmā šobrīd ir pieejama jaunākā versija – *Scratch 2*. Par labu pārlūkprogrammas versijas izvēlei ir tas, ka to var lietot uzreiz, neko datorā neinstalējot (interneta pārlūkprogramma ar *Flash* spraudni parasti visos datoros ir). Turklāt *Scratch 2.0* ir arī samērā “viegla” un neprasa lielu interneta joslas platumu (“ātrumu”) un datu apjomu, arī pārlūkprogrammas versijas nav nepieciešamas visjaunākās, pietiek ar *Chrome 7*, *Firefox 4*, *Internet Explorer 7* vai jaunākām to versijām ar *Adobe Flash Player 10.2* vai jaunāku. *Scratch 2* ir veidots ekrāna izšķirtspējai vismaz 1024 x 768. Ja dators tomēr neatbilst šīm prasībām, var lejupielādēt, instalēt un lietot iepriekšējo versiju *Scratch 1.4*.

Pilnvērtīgam darbam ar tiešsaistes *Scratch* ir jāizveido lietotāja konts <http://scratch.mit.edu/> (skat. 1. attēlu): “*Join Scratch*”, tad viegli saprotamā vednī jūs ievadiet savu vēlamo lietotājvārdu, paroli, reālu e-pastu (tas var noderēt aizmirstas paroles atjaunošanai) un dažus citus datus (2. attēls). Nav iemesla neregistrēties, jo tikai reģistrēti lietotāji var saglabāt tīmeklī savus projektus, padarīt tos tur pieejamus citiem, piekļūt visiem resursiem un piedalīties *Scratch* kopienas aktivitātēs.

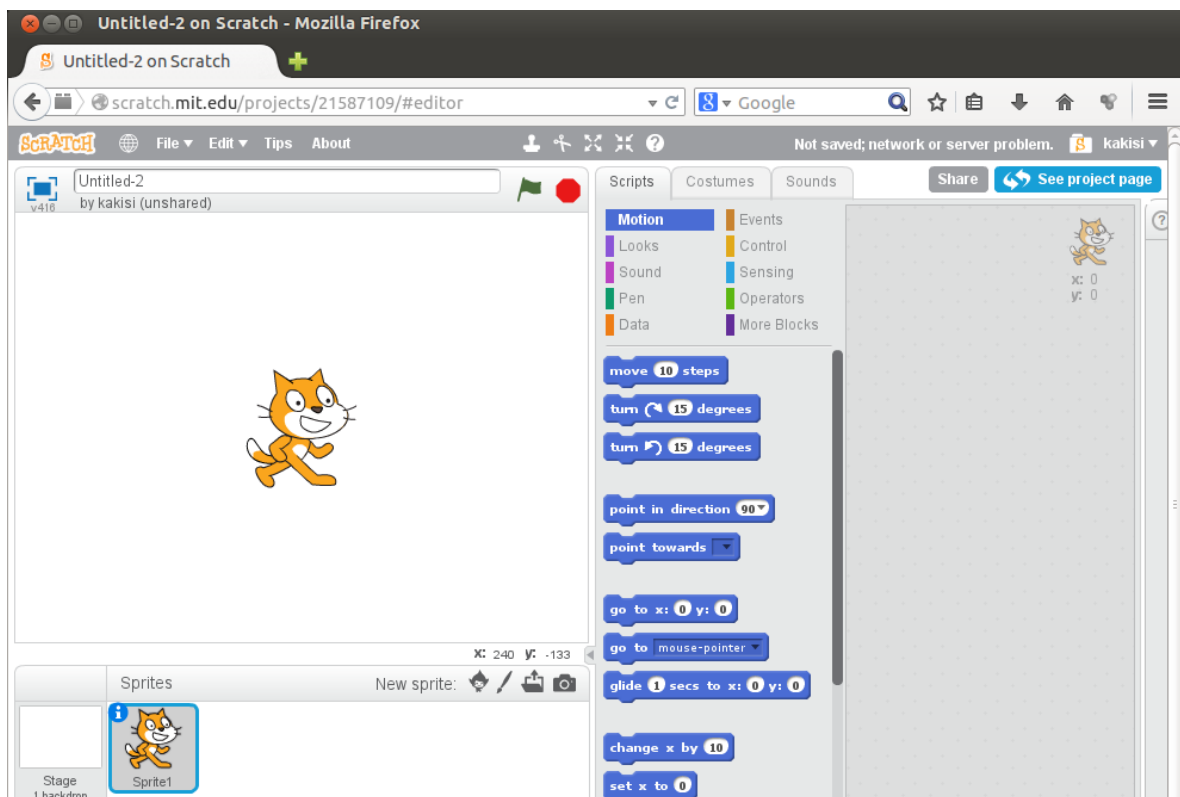


1. att. *Scratch* tīmekļa vietnes sāukmlapa.



2. att. Jauna lietotāja reģistrēšanas vednis.

Pēc pieteikšanās sistēmā ar jauno lietotājvārdu un paroli, izpētiet, kas ir pieejams, ko *Scratch* izstrādātāji jums piedāvā. Ja augšējā zilajā joslā nospiedīsiet “*Create*”, jums ielādēsies rīks jūsu pirmā *Scratch 2* projekta izveidei (3. attēls). Mac OS X lietotājiem šis skats parādīsies tikai pēc *Adobe Flash Player* uzinstalēšanas, ko var veikt klikšķinot uz *Flash* instalēšanas saites vai pogas, kas parādās, ja pārlūkprogrammai nav *Flash* spraudņa (*plug-in*).



3. att. *Scratch 2* projekta izveides logs pārlūkprogrammā *Mozilla Firefox*.

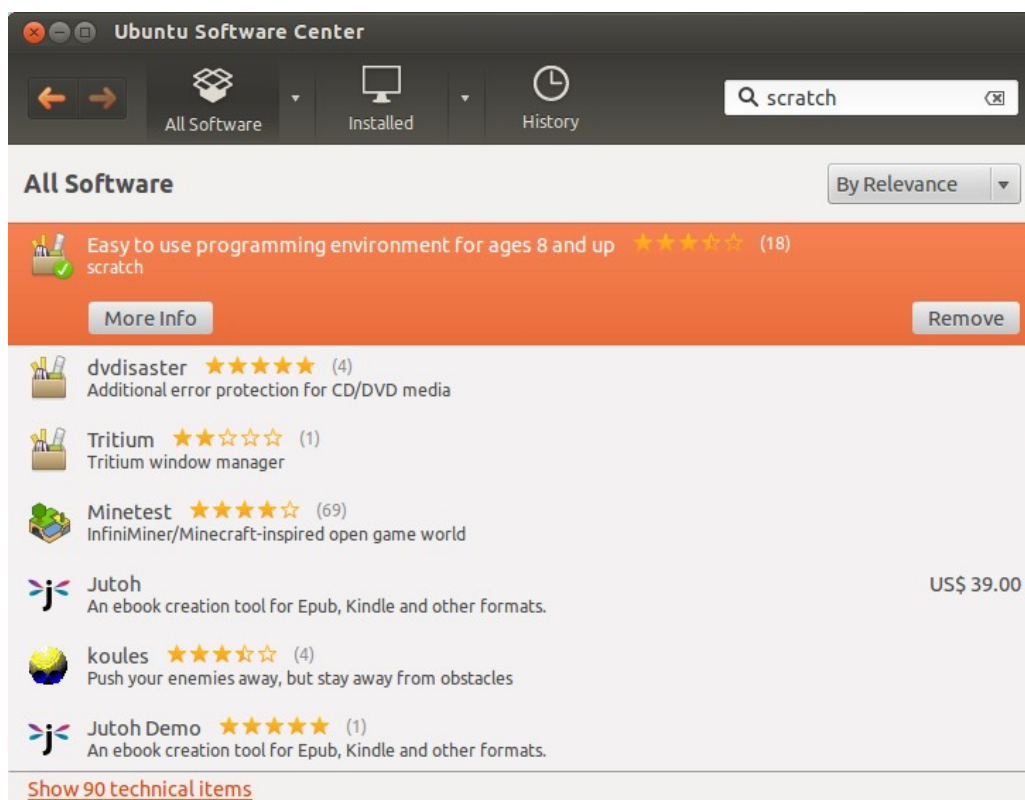
Tā kā *Scratch* tulkojums latviešu valodā nav pabeigts, šajā pamācībā daudzviet tiek

izmantota kā latviešu, tā angļu versija, kas arī ir izstrādātāju valoda. Saites uz *Scratch* aprakstiem atrodamas <http://scratch.mit.edu/help/>, šajā lapā ir saite uz iesācēju īso aprakstu 16 lappusēs angļu valodā, tas ir pieejams PDF failā: http://cdn.scratch.mit.edu/scratchr2/static/_f26c59f090fca8b69aaa1e5f27a3dd2f_//pdfs/help/Getting-Started-Guide-Scratch2.pdf. Turpat ir arī saite uz videopamācībām: <http://scratch.mit.edu/help/videos>. Detalizēti valodas apraksti ir pieejami pašā *Scratch* programmēšanas vidē. Konkrētus piemērus un pamācības latviešu valodā aplūkosim šī konspekta nākamajās nodaļās.

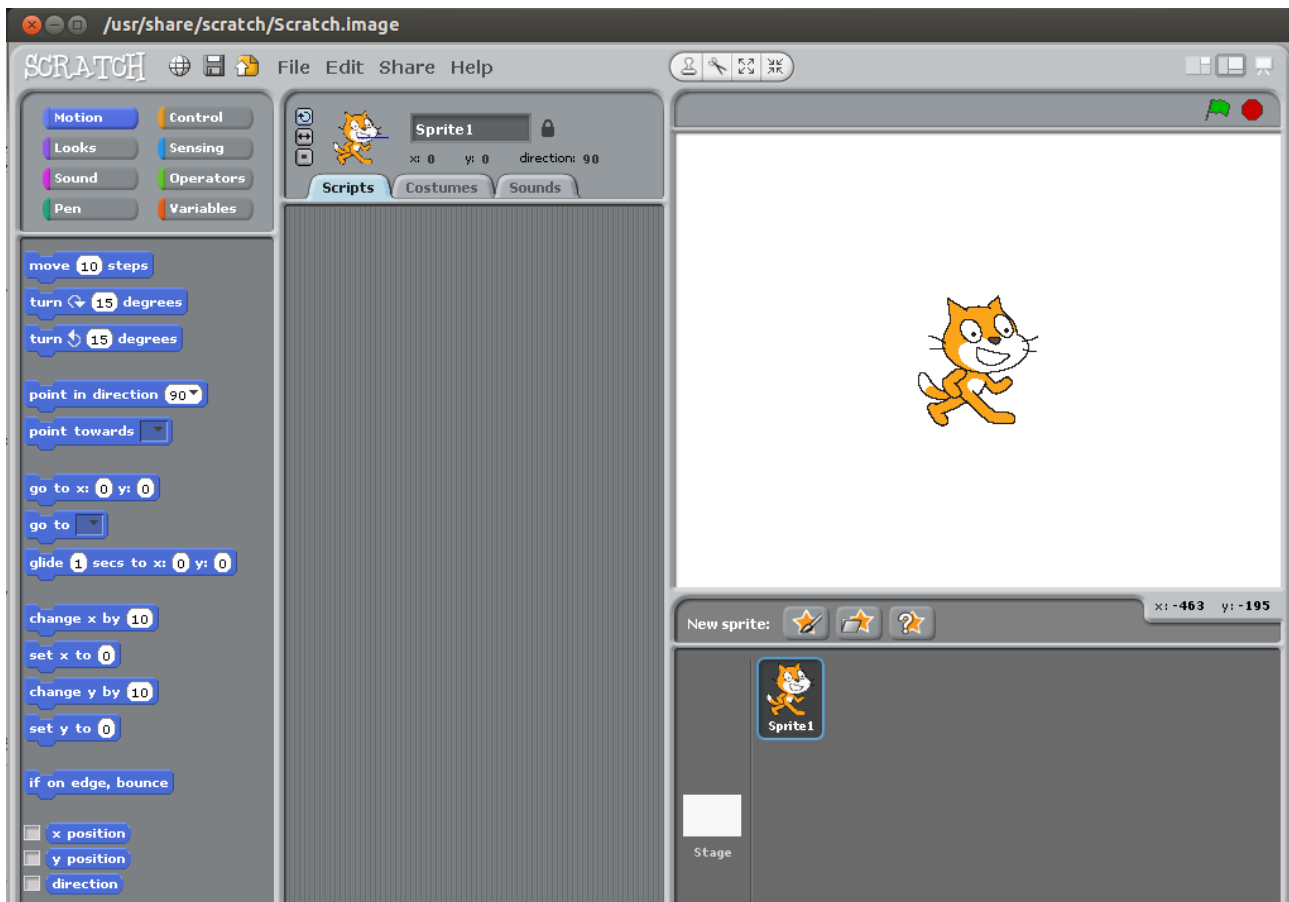
2.2. *Scratch* instalācijas sagatavošana datorā

Ja vēlaties instalēt *Scratch* kā lietojumprogrammu savā datorā, tad stabilā versija lejupielādem ir *Scratch 1.4*, to var atrast, “*Scratch Help*” lapā <http://scratch.mit.edu/help/> labajā pusē lejā atrodot “*Scratch 1.4 Download*” vai tieši: http://scratch.mit.edu/scratch_1.4/ (šī materiāla autors tomēr neuzbāzīgi iesaka vienmēr izmantot jaunākās versijas, šoreiz tā ir *Scratch 2 Beta*). *Scratch 1.4* lejupielāde un instalācija ir vienkārša, tiek piedāvāti faili:

- ScratchInstaller1.4.exe priekš *Microsoft Windows 2000, XP, Vista, 7, 8, 8.1*,
- MacScratch1.4.dmg priekš *Mac OSX 10.4* un jaunākām *Apple Mac OSX* versijām,
- Linux *Debian* un *Ubuntu* programma *Scratch* atrodama programmatūras centrā “*Software center*” (meklējam ar mazo burtu “scratch”, 4. attēls) sākot no *Ubuntu 12.04 LTS (Scratch 1.4*, skat. 5. attēlu).



4. att. *Scratch* Ubuntu programmatūras centrā.

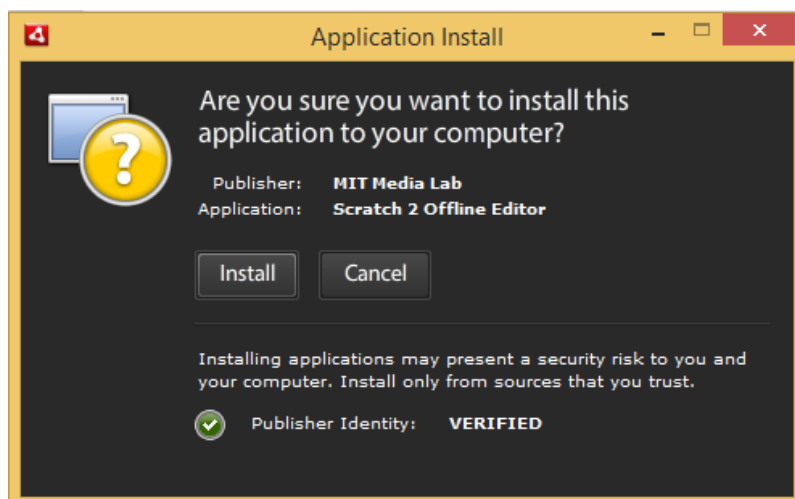


5. att. Scratch 1.4 uz Ubuntu 12.04 LTS.

Ja izvēlamies lietot **Scratch 2.0 Beta** nesaistes versijas instalāciju datorā (pagaidām instalēšanai lokāli datorā ir pieejama tikai beta jeb mazliet nepabeigta versija), interneta lappusē <http://scratch.mit.edu/help/> atrodam “Scratch 2 Offline Editor (Beta)” saiti <http://scratch.mit.edu/scratch2download/>. Scratch 2 instalācija ir nedaudz sarežģītāka kā Scratch 1.4. Scratch 2 nesaistes jeb lietotnes versija ir pieejama priekš Windows, Mac un Linux (dažām un pagaidām tikai 32 bit versijām).

Scratch 2 instalācija sastāv no 2 soļiem: 1) lejupielādejam un instalējam Adobe AIR savai lietotajai operētājsistēmai, 2) tad lejupielādejam un instalējam Scratch offline editor (tas darbojas, izmantojot minēto Adobe AIR). Visas lejupielādēm nepieciešamās saites atrodamas <http://scratch.mit.edu/scratch2download/>. Turpat atrodamas projektu piemēru ZIP un pamācību PDF failu lejupielādes (ar Creative Commons attribution and share-alike licenci, ar šādu licenci tiek izplatīts arī šis konspekts), kopā nebūs vairāk par 100 MB.

Uz Microsoft Windows programmas Adobe AIR lejupielāde tiek pāradresēta uz <http://get.adobe.com>, instalēšanai palaižam lejupielādeto failu AdobeAIRInstaller.exe, tam nepieciešamas Windows administratora tiesības, kā arī jāpiekrīt Adobe licencei. Pēc tam palaižam otru lejupielādeto failu Scratch.air un, pēc neilga procesa novērošanas, Adobe instalācijas logā spiežam “Install” (6. attēls), tad piekrītam Windows User Account Control brīdinājumam par instalāciju (ja tas parādās) un sagaidām procesa beigas.



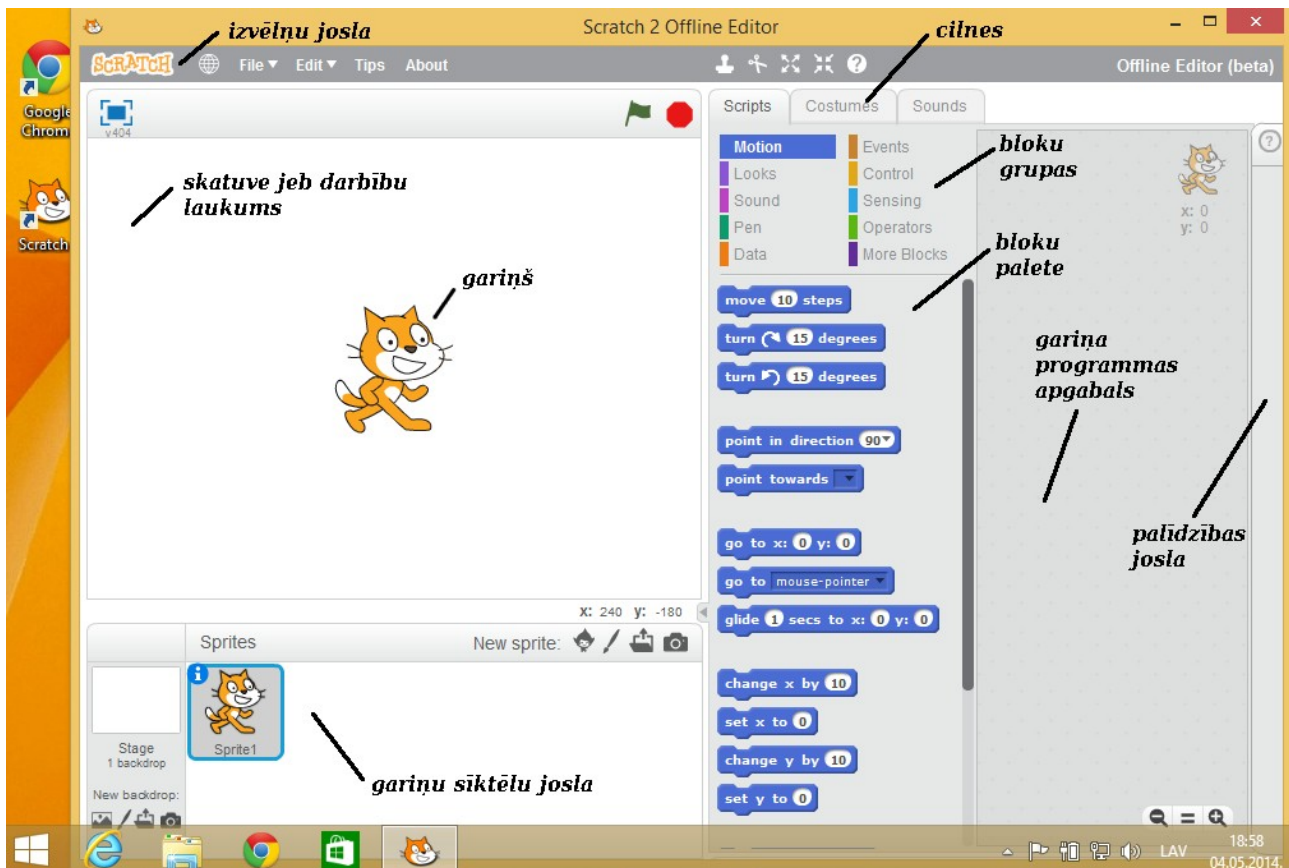
6.att. *Scratch 2* instalēšanas *Adobe AIR* vednis.

3. Sākam darbu ar *Scratch*

3.1. *Scratch* vide

Šīs nodaļas izpētes gaitā jūs izveidosiet savu pirmo *Scratch* programmiņu. *Scratch 2* logs uz *Windows 8.1* dots 7. attēlā. Nav nozīmes, kuru *Scratch* lietošanas veidu un operētājsistēmu vai pārlūkprogrammu jūs izvēlaties, galvenais, ka varam sākt apgūt šo brīnišķīgo apmācošo un izklaidējošo programmēšanu. *Scratch 2* augšējā kreisajā stūrī ir skatuve jeb rezultāta, darbību pamēģināšanas laukums (480 x 360 punkti liels), kurā var redzēt konkrētā programmas soļa izpildi (*Scratch 1.4* izkārtojums ir mazāk ērts - tas ir augšējais labais stūris).

Vidējā slejā tās augšējā daļā ir cilnes: skripti, kostīmi (izskati), skaņas. Cilnē "Scripts" ir 8 vai 10 (atkarībā no versijas) bloku grupas, katra savas krāsas: kustības, izskati, skaņas, zīmuļi, dati, notikumi, vadība, novērošana (sensori), operatori, pārējie bloki. Katra programmas komanda (bloks) ir izmēģināma (ar dažādiem nosacījumiem un parametriem), ja izpilda dubultklikšķi uz tās. Katra komandu grupa ir savā bloku krāsā, piemēram, komandas, kas regulē skaņu, ir violetā krāsā (skat. 7. attēlu).



7. att. Scratch 2 programmas logs uz Windows 8.1 ar paskaidrojumiem.

Komandu var vilkt pa labi uz skriptu/ programmas apgabalu un padarīt par daļu no programmas scenārija, kas darbinā uz izvēlēto gariņu (*sprite*). Vistālāk pa labi Scratch 2 ir izslidošā josla ar palīdzības aprakstiem. Izveidotus projektus Scratch glabā *Flash* formātā un līdz ar to tos var izmantot visur, kur ir *Flash Player* (*iPhone* un *iPad* tā noklusēti nav). Internetā var atrast pamācības, kā projektu, piemēram, nelielu spēlīti, pārveidot par *Windows* *exe* failu vai vienkārši iekļaut (*embedding*) kāda bloga vai mājaslapas `<iframe>` tegā.

Scratch ir jaudīgas daudzplūsmu multimediju veidošanas iespējas, bet klasiskā programmēšana ir tikai pamatiespēju robežās (tomēr ir iespējams darbs ar viendimensijas masīviem, simbolu virknēm, zarošanos, ciklēm, peldoša punkta skaitļiem utt.). Scratch var pievienot paplašinājumus darbam ar failu I/O, mazliet var sadarboties ar citām programmām (tas eksperimentāli), ir paplašinājumi sadarbībai ar *Lego Mindstorms* un *Arduino* mikrokontrolieriem. Scratch izmanto pasaulē vadošās izglītības iestādes, tas atrodas straujā attīstības procesā un ar katru gadu kļūst vēl labāks.

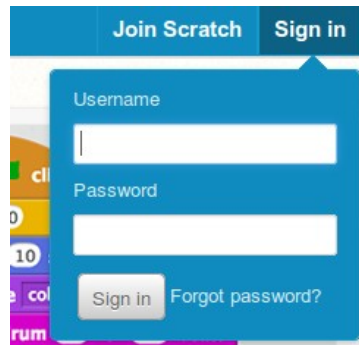
3.2. Scratch 2 valodas un vides īss apskats ar piemēru

Scratch ir moderna apmācoša programmēšanas valoda, kurā var veidot spēles, interaktīvus stāstus, prezentācijas, animācijas, multiplikācijas filmiņas ar skaņu un arī nelielas lietojumprogrammas.

Izvēlieties, kurš Scratch lietojuma veids jums būs ērtāks: tiešsaistes interneta pārlūkprogrammā vai nesaistes instalācija datorā. Ja lietojat planšetdatoru, tad tiešsaistes versija bus optimāla, šī ir viena no retajām programmēšanas valodām, ar kuru var produktīvi strādāt uz skārienjutīga ekrāna, jo programmēšana notiek pārbīdot blokus un nedaudz mainot to īpašības. Tas

atgādina puzzles salikšanu, tikai jūs radoši veidojat savu izdomāto “puzli”.

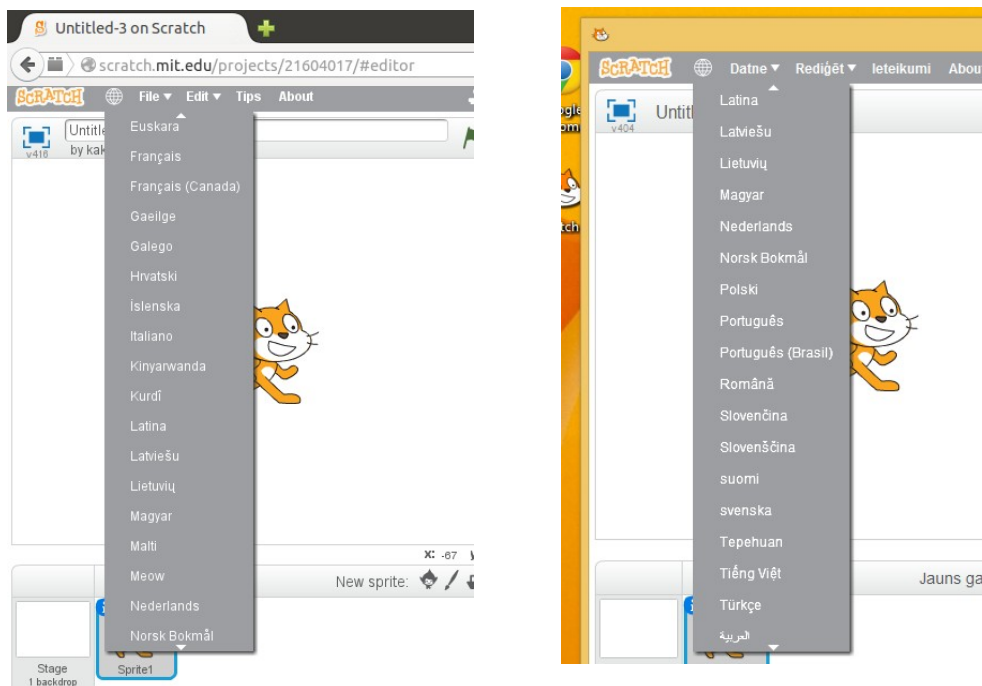
Ja lietošiet pārlūkprogrammas *Scratch*, vispirms vēlams piesakieties sistēmā ar savu iepriekš izveidoto lietotājvārdu un paroli, kaut vai lai varētu saglabāt “mākonī” savu darbu (8. attēls), bet pēc tam augšējās zilās joslas sākumā spiediet “Create” – sākt jaunu projektu (9. attēls). Pēc tam, spiežot uz zemeslodes ikonas, izvēlamies latviešu vai citu jums vēlamo valodu (10. attēls), lietojot *Windows* vides instalāciju, sākam ar valodas izvēli, jo projekts saglabāsies lokāli datorā (“mākonī” to ielādēsīm tad darba beigās).



8. att. Pieteikšanās *Scratch 2* tiešsaistes sistēmā.



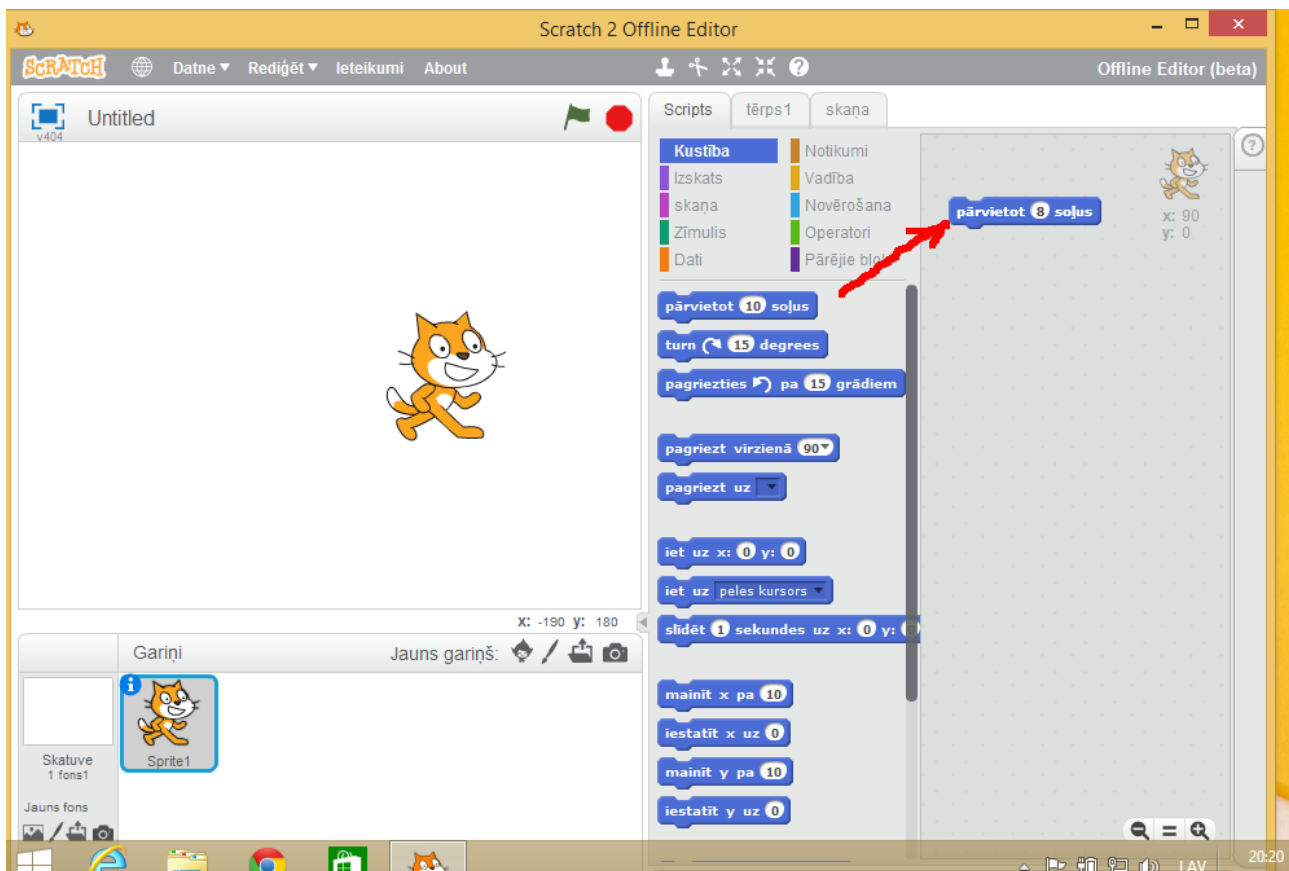
9. att. “Create” - sākt jaunu projektu.



10.att. Saskaņotās valodas izvēle tiešsaistes un nesaistes *Scratch 2* versijā.

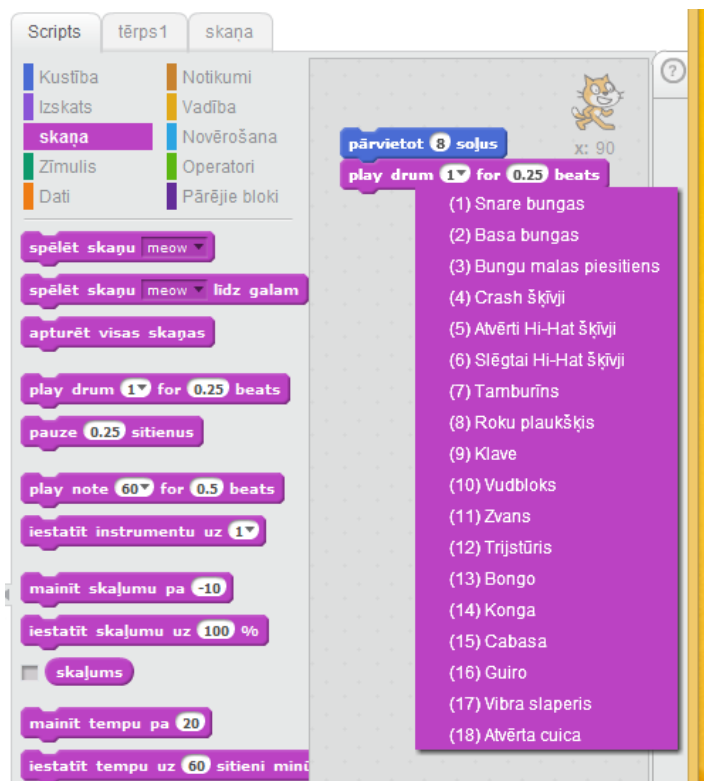
Scratch darbību izpildes laukumā jau dots dzeltena kaķa zīmējums, sākumā to arī izmantosim. *Scratch* gariņi ir divdimensionāli zīmējumi, kas izveidoti uz caurspīdīga fona, šiem objektiem var likt kustēties, mijiedarboties, mainīties. Gariņiem (programmēšanas objektiem) ir trīs galvenie komponenti: skripti, izskati, skaņas – tiem atbilst 3 galvenās cilnes (*tabs*).

Ieprogrammēsim kaķi veikt 8 soļus, šim nolūkam, pieturot ar kreiso peles taustiņu, pārvelkam kustību bloku sadaļas bloku “pārvietot 10 soļus” (angļu versijā “move 10 steps”) pa labi – uz programmas koda laukumu un skaitli 10 izmainām uz 8 (11. attēls). Bloks “pārvietot” atrodas bloku grupā “Kustība” (“Motion”). Ar dubultklikšķi uz pārvilkta bloka varam pamēģināt, kā kaķis kustēsies.



11. att. Kaķa kustības ieprogrammēšana.

Tad pievienosim skaņu, pārvelkot no bloku grupas “Skaņa” (“Sound”) bloku “play drum 1 for 0,25 beats”, var izvēlēties 1. skaņas vietā no lejuvizkrītošā saraksta kādu no 18 skaņām (12. attēls).



12. att. Pievienojam īsu skaņu.

Tad pievienojam vēl vienu kustību bloku, liekot kustēties atpakaļ (-8) un pēc tā skaņas bloku “spēlēt skaņu meow” ar kaķa ņaudiem. Viens bloks nes sevī to jēgu, ko citās programmēšanas valodās operatora vai komandas jēdzieni. Pievienojot blokus, tie jāliek iepriekšējam tieši klāt, veidojot pārskatāmu koda gabalu, bloki it kā salīp. Šo 4 bloku kopumu varam likt atkārtot vairākas reizes: izvēlamies bloku grupu “Vadība” (“Control”) un no tā velkam bloku “atkārtot 10 reizes” (“repeat 10 times”), 10 vietā norādot, piemēram, 5. Šādi esam izveidojuši to, ko programmētāji mēdz saukt par ciklu. Ar dubultklikšķa uz dzeltenās cikla skavas pamēģinām, kā šis scenārijs vai nelielā programmiņa strādā. Iespējams, ir par ātru, tad varam pāris vietās pievienot vadības jeb kontroles bloku “gaidīt 1 sekundi” (skat. 13. attēlu).



13. att. Izveidotais scenārijs ar 2 kustībām, 2 skaņām, 2 uzgaidīšanām un 1 ciklu.

Jūties brīvi un radoši, kā kaķis, pamēģiniet paspēlēties arī ar citiem blokiem, ar kreiso dubultklikšķi uz cikla patestējiet, kas sanācis. Jūs te neko salauzt nevarat, lieku bloku var noņemt, ar labo peles taustiņu uzklikšķinot un izvēloties “dzēst” (“delete”), te ir pamanāma arī iespēja pievienot komentārus, kas sarežģītākas programmas izveidē ir ļoti noderīgi, lai vēlāk varētu saprast, kas tur bija domāts.

Eksperimentējot tālāk, varam no bloku grupas “Izskats” (“Looks”) pievienot scenārija galā

“teikt Hello!” (“say Hello!”), ierakstot Hello vietā ko citu, tas parādīsies virs kakīša kā viņa domas attēlojums. Tad no grupas “Notikumi” (“Events”) scenārija sākumā pievienojam “kad nospiests zaļais karogs” (“when green flag clicked”). Līdz ar to programmiņu varēs palaist ar klikšķi uz zaļā karoga, un apturēt vienmēr var ar klikšķi uz sarkanā aplīša. Tikpat labi varējām izvēlēties iedarbināšanu ar atstarpes taustiņu. Kā izskatās līdz šim izdarītais, redzams 14. attēlā.

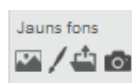


14. att. Iepriekšējos soļos izveidotā programma.

Jebkuru programmu vienmēr var uzlabot vai vismaz papildināt. Attīstot domu tālāk, varam no bloku grupas “Izskats” (“Looks”) pievienot galā “mainīt krāsa efektu par 25” (“change color effect by 25”), paeksperimentējam, kas notiek. Tad no grupas “Notikumi” krāsu maiņas blokam priekšā pieliekam “kad piespiests taustiņš atstarpe” (“when space key pressed”), protams, taustiņu var izvēlēties lejupizkrītošajā sarakstā citu.

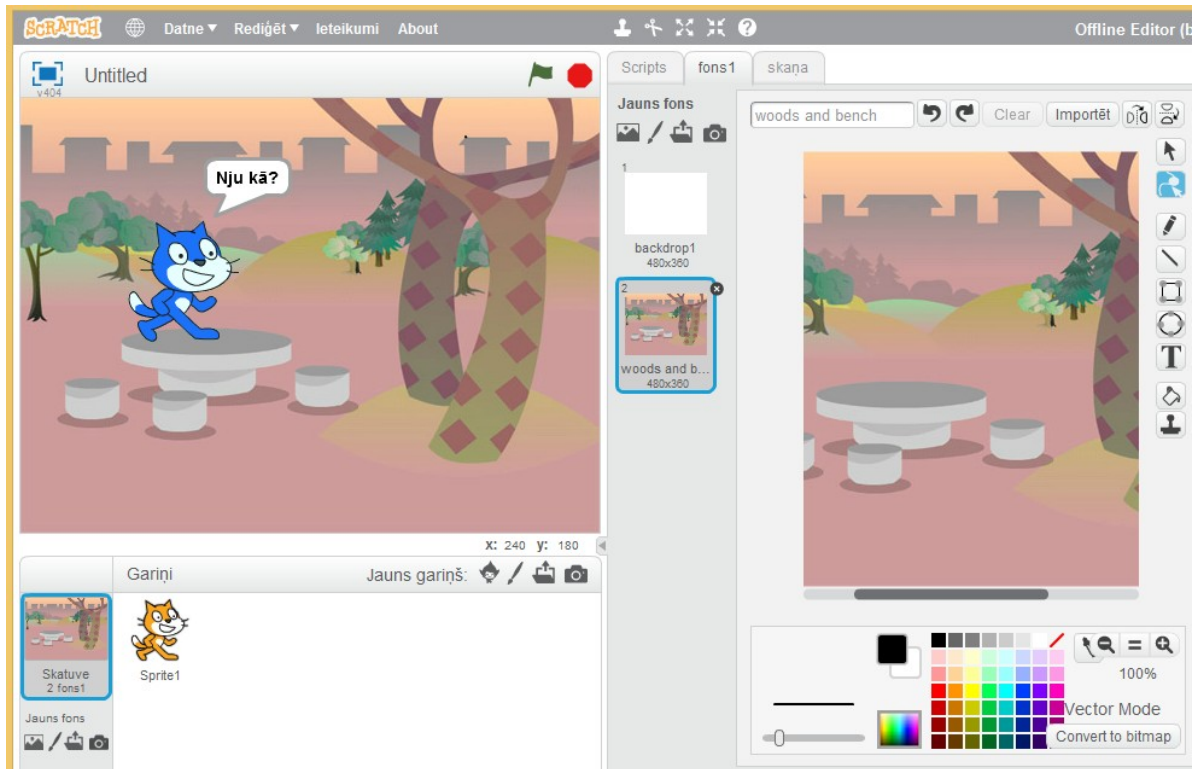
3.3. Iesāktās programmiņas pilnveidošana

Līdz šim darbojāmies nelielās programmiņas (dažreiz tādu sauc par skriptu, scenāriju) veidošanas režīmā – augša pret vidu aktīva ir cilne “Scripts”. Tagad pienācis laiks izpētīt *Scratch 2* programmēšanas vides citas iespējas. Ja aplūkojam *Scratch 2* loga apakšējo kreiso stūri, tad redzam, ka bez viena gariņa (“Sprite 1” - kaķis), mūsu scenārijam ir arī fons “Skatuve” (“Stage”), un tas ir balts. *Scratch 2* loga apakšējā kreisajā stūrī ir “Jauns fons” (“New backdrop”) četras ikonas, spiežam pirmo un izvēlamies kādu no piedāvātajiem foniem (pārējās trīs ir iespējas uzzīmēt vai ielādēt failu, vai nofotografēt ar datora kameru), skat. 15. attēlu.



15. att. Fona pievienošanas ikonas.

Tad izvēlamies kādu no piedāvātajiem attēliem, kas pirmajiem soļiem ir ātrākais ceļš. Attēls parādās labās puses vidējā cilnē. Tā atkarībā no tā, kas ir aktīvs objektu rindā apakšējā kreisajā stūrī virs “Jauns fons”: “Tērps” (“Costumes”) vai “Fons” (“Backdrops”). Fona režīmā te redzam izvēlēto attēlu, varam nospiest “Convert to vector”, jo darba gala rezultātu iegūsim kā *Flash* filmiņu, kas tiek realizēta vektorgrafikā (tad attēla kvalitāte ir labāka, mainot izmērus), te ir arī iespējams rediģēt attēlu ar rīkiem, kas pazīstami kaut vai no lietotnes *Paint*. Strādā atpakaļsoļi **Ctrl+Z**. Sadarītais pēc fona pievienošanas redzams 16. attēlā.



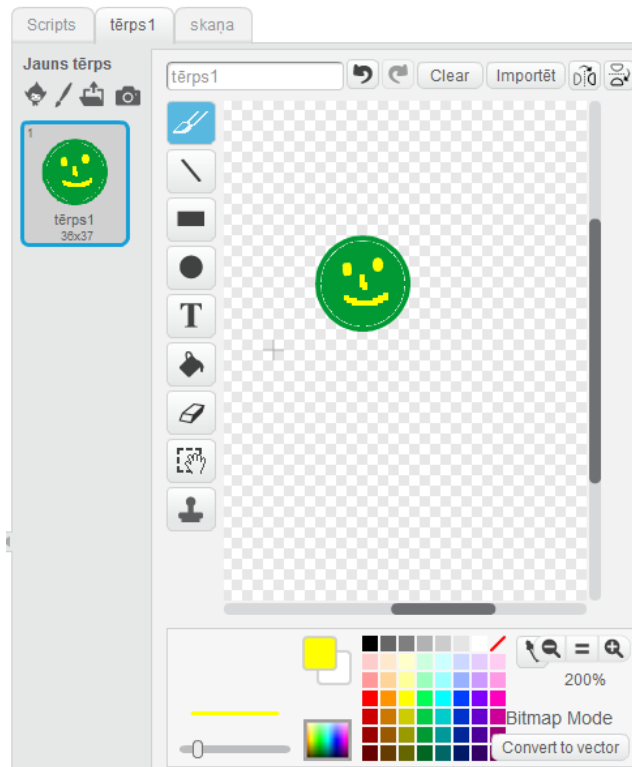
16. att. Fons pievienots.

Kā viegli ieraudzīt, mazliet virs un pa labi no esošā viena gariņa ("Sprite1") ir rīku josla "Jauns gariņš" ("New sprite"), skat. 17. attēlu. Tāpat kā ar fonu, arī jauno gariņu var izvēlēties no dotajiem, var uzzīmēt vai ieimportēt no faila vai kameras.



17. att. Jauna gariņa jeb elfa pievienošanas rīki.

Ātrāk ir kaut ko gatavu ieimportēt, bet radošāk ir varbūt citādi, tādēļ mēģināju zīmēt mistisku dzīvo lidojošo arbūzu, skat. 18. attēlu. Arī arbūza gariņu lietderīgi konvertēt vektorgrafikā.



18. att. Jauna gariņa (*Sprite*) zīmēšana.

Tagad novietosim arbūzu kāda vieglprātīgi izvēlētā vietā, piemēram, karājoties kokā, un izdomāsim tam skriptu, jeb darbību scenāriju mūsu programmiņā. Varam pielikt, ka arbūzs nokritīs un izdos sajūsmas pilnu skaņu, ja nospiedīs atstarpes taustiņu (otrreiz). Arī šeit x ass ir horizontāli un y vertikāli (koordinātes skaitliski redzamas zem darbības laukuma (skat. 16. un 19. attēlu).

Programmas koda apgabala augšējā labajā stūrī redzams bālināts tā gariņa attēls, kura kodu redzam. Starp gariņiem var pārslēgties, klikšķinot uz to sīktēliem gariņu joslā. Koordinātu sākumpunkts ($x=0$, $y=0$) ir skatuves centrs, pa labi un uz augšu ir pozitīvas, bet pa kreisi un uz leju ir negatīvas koordinātu vērtības. Tātad, maksimālās vērtības $x=240$, $y=180$ ir skatuves augšējā labajā stūrī. Scenārijs redzams tam gariņam, kurš ir aktivizēts gariņu rindā.



19. att. Arbūza gariņa scenārijs.

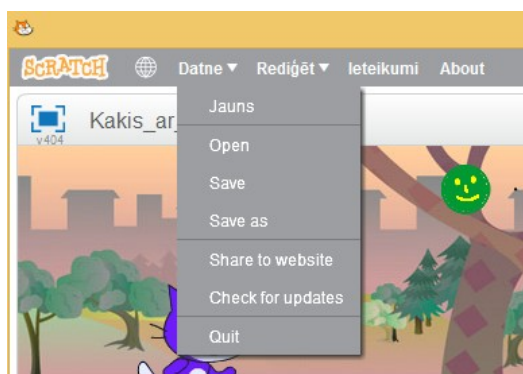
Visbeidzot (varēja arī sākumā) dosim projektam nosaukumu atbilstoši tam, kas ir sanācis, piemēram, “Kaķis ar arbūzu” un to saglabāsim datorā vai *Scratch* portālā (ja esam tajā pieteikušies, portālā var noņemt publicēšanu jeb koplietošanu (“*Sharing*”), ja pagaidām nevēlamies visiem rādīt savu pirmo vingrinājumu). Darba vides kopskats beigās dots 20. attēlā, bet kas nu par beigām, jebkuru programmu var uzlabot...



20. att. *Scratch* pirmais programmas piemērs beigās.

3.4. Darba pabeigšana un nākamās aktivitātes

Nesaistes jeb lietojumprogrammas versijā veidotu *Scratch* projektu var augšupielādēt internetā, izvēloties “Datne/ Share to Website”, skat. 21. attēlu. Tam nepieciešams *Scratch* portāla lietotāja konts, ko izveidojām pēc apraksta konspekta sākumā (sadaļā 2.1.).



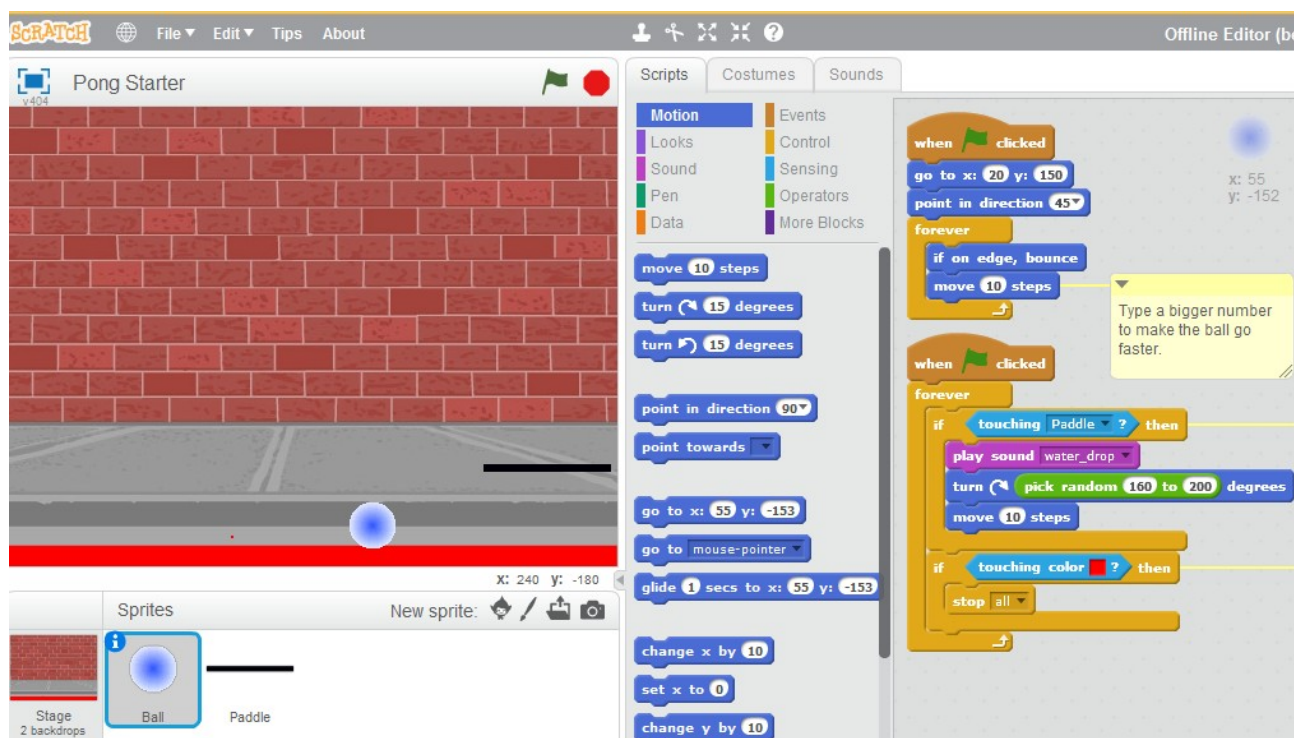
21. att. Projekta augšupielādes iespēja.

Scratch portālā ielādētā programmiņa, palaižot to interneta pārlūkprogrammā, attēlojas izmantojot Adobe *Flash*. Pagaidām *Scratch* izstrādātāji nepiedāvā iespēju saglabāt rezultātu kā *swf* vai *exe* failu, bet jūs tomēr varat programmiņu publicēt savā blogā vai mājas lapā kā iekļauto skriptu (skat. “*How to Embed a Project*” http://wiki.scratch.mit.edu/wiki/Embedding_Projects); internetā ir atrodamas arī entuziastu veidotas pamācības, kā radīt *Windows* vides palaižamo *exe* failu.

Ja saglabā projektu datorā no interneta, izveidojas *sb2* fails, piemēram, *Kakis_ar_arbuzu.sb2*, tas ir arhīva tipa fails ar skriptu *project.json* un *svg*, *png* un *wav* failiem (ja interesē, vairāk par *Scratch* failu formātiem

http://wiki.scratch.mit.edu/wiki/Scratch_File_Format_%282.0%29). Atceramies, ka *Scratch* loga labajā malā var klikšķināt uz jautājuma zīmes un iegūt papildinformāciju. Kā arī var iegūt palīdzību par doto elementu, klikšķinot augšējā rīku joslā uz aplīša ar jautājuma zīmi un tad uz interesējošā bloka.

Scratch portālā caur palīdzības lappusi var nonākt “Scratch 2 Offline Editor” tīmekļa lappusē <http://scratch.mit.edu/scratch2download/>, no tās var lejupielādēt arhīvu ar dažiem projektu piemēriem un pamācību “Scratch Cards” PDF failā. Piemēram, elastīgas bumbiņas kustības apraksta scenārija viens no variantiem dots 22. attēlā (no projektu piemēru arhīva). Kādu projektu atverot, to var darbināt un modificēt.



22. att. Elastīgas bumbiņas kustības programma.

4. Vēl dažas no Scratch 2 daudzajām iespējām

Apģūstot *Scratch*, atceramies ideju par dialogu ar datoru: dators ne tikai dara, ko mēs tam liekam (nevis, ko tikai domāš gribam), bet arī mums piedāvā iespējas. Vai pamanījāt pie gariņa ikonās burtiņu “i” augšējā kreisajā stūrītī, paklikšķiniet uz tā, un iegūsiet mazliet informācijas par doto gariņu un mainīt dažas tā īpašības. Vienmēr der izpētīt programmas visu laukumu, paklikšķināt, kur vien var, tas vēlāk atvieglo iespēju meklēšanu.

Tā, piemēram, pievērsiet uzmanību, ka virs skatuves augšējā kreisajā stūrī atrodas zila taisnleņķa poga, ar to var skatuvi pārslēgt pilnekrāna (*Full screen*) režīmā, kad demonstrējam programmu. Tā kā izeja parasti atrodas turpat, kur ieeja, tad poga, kas pilnekrāna režīmā ir tajā vietā, ļaus atgriezties atpakaļ pamatskatā. *Scratch 2* atbalsta 3 ekrāna izmērus, trešais ir samazinātais, tajā koplietotie projekti redzami dažās lietotāju profilu lappusēs *Scratch* portālā.

Veidojot programmu, nav arī *Scratch* nav ierobežojuma tās garumam – bloku skaitam, tas var būt ļoti liels, teorētiski robežu nosaka datora veiktspēja. Tomēr ļoti garas programmas veidot un rediģēt uz mazākas veiktspējas datora (svarīga šeit ir arī videokarte, ne tikai procesors un RAM) var būt apgrūtināši. Uz skatuves ne tikai var darboties vairāki gariņi, bet tai var būt piešķirti arī vairāki

foni, kas programmas izpildes gaitā mainās.

Ja atverat svešu projektu, tā bloku kodu var apskatīt, spiežot “See inside”, bet to pārveidot, radot savu versiju, var spiežot “Remix”.

Scratch 2 var strādāt ar šādiem attēlu tipiem: PNG, JPG, BMP, GIF, animētie GIF. Kad attēls konvertēts vektorgrafikā ar to strādā SVG režīmā. Attēlus varat zīmēt kā *Scratch* rīkā, tā ar plašākām iespējām citās grafiskās programmās, piemēram, brīvlietojuma *Gimp* (<http://www.gimp.org/>), *Tux Paint* vai *Paint*.

Scratch 2 var strādāt ar divu tipu audio failiem: WAV un MP3.

Pēc kāda objekta nodzēšanas var pārdomāt un “Edit” izvēlnē atrast “Undelete” iespēju.

Projekta labajā apakšējā stūrī ir mērogošanas pogas “-”, “=”, “+”, kas gara koda režīmā ļauj to samazinātu it kā attālināti pārredzēt.

Scratch 2 tiešsaistes jeb pārlūkprogrammas versijā redaktora apakšējā labajā pusē ir krājumu vieta - josla ar nelielu trijstūri, uz kura klikšķinot parādās “Backpack” josla, tur var saglabāt objektus vēlākai atkalizmantošanai, “Backpack” joslā ievēlot.

Viena gariņa kodu var nokopēt citam gariņam, koda bloku velkot un noņemot uz otra gariņa sīktēla gariņu joslā.

Pievērsiet uzmanību, ka bloki satur dažādu tipu argumentus, piemēram, bloks “move 10 steps” satur skaitlisku argumentu, bet “Say Hello!” – simbolu virkni, “play sound meow2 until done” – izkrītošo izvēlni, “if then” – bināru (*boolean*) ceļa norādi, “set pen color to” – krāsu, “repeat until key space pressed” – klaviatūras ievadu. Bloka noklusēto argumenta vērtību parasti var mainīt uz citu tā paša vērtību, izņēmums ir *boolean* argumenti, tiem noklusētā vērtība ir nekas (*false*).

Pavisam *Scratch 2* ir 148 bloki, kur katram ir sava nozīme. Bloka aprakstu var iegūt, klikšķinot *Scratch 2* redaktora augšējā rīku joslā uz aplīša ar jautājuma zīmi un tūlīt pēc tam uz interesējoša bloka, tad atvērsies labajā pusē bloka apraksts.

Blokus ievietojot vienu otrā, piemēram, funkcijā vai procedūrā ievietojot tās argumentu bloku, viegli pamanīt, ka argumentu blokiem ir iespējams 3 veidu izskats, un ievietot var tikai sakrītošas formas blokus. Tas tādēļ, ka *Scratch* atšķir 3 veidu mainīgos: skaitliskie mainīgie (aplītis vai garenais bloks ar apaļiem galiem), teksta mainīgie (taisnstūris), loģiskie mainīgie (sešstūrveida).

Klasisks programmēšanas uzdevums ir skolā pazīstamā kvadrātviensējuma (*quadratic*) atrisināšanas programma, *Scratch*, protams tādu var izveidot. Aplūkojiet programmētāja ar segvārdu HaiTek doto vienu no iespējamajiem piemēriem: <http://scratch.mit.edu/projects/395439/>. Pirmajā brīdī programmiņa liekas īsa, bet tā gluži nav – paklikšķiniet uz gariņu joslas visām ikonām un skatuves fona, papētiet kodu, mēģiniet tam izsekot.

Savukārt *Scratch 1.4* prgrammas piemērs par interesantajiem Fibonači skaitļiem atrodams <http://progopedia.com/language/scratch/>, izveidojiet to *Scratch 2.0* versijā. Šādus un citus vingrinājumus aplūkosim nākamajos pamācību failos.

Scratch gariņi savā starpā apmainās ar pārraidītajiem ziņojumiem, kas šim nolūkam speciāli

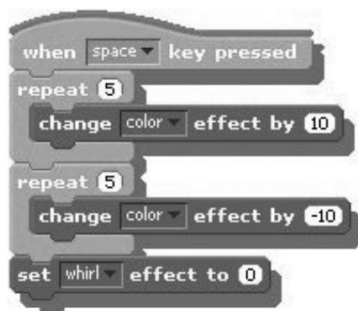
jāizveido, bloki pārraidīt (message1) (broadcast (message1)) un kad es saņemu (message1) (when I receive (message1)). Tas ir veids, kā nodot informāciju no viena gariņa otram.

Scratch ir noderīgas iespējas veidot jaunus mainīgos, viendimensijas masīvus (sarakstus), definēt savas procedūras. Ja kāda darbība lineāri izveidotā programmā daudzreiz atkārtojas, programmu var optimizēt vai nu ar ciklu, vai sarežģītākā gadījumā ar ar procedūru – apakšprogrammu, kas atgriež vienveidīgus notikumus, bieži, no dažādiem ieejas datiem. Piemēram, jds.df.lu.lv 1. tēmā ir *Scratch* veidota prezentācija “PROGRAMMAS PIEMĒRS: *Scratch* veidota prezentācija v.1 par Jauno datorīku skolas šo e-vidi...” - <http://scratch.mit.edu/projects/24238708>, šajā, 1. versijā programma ir veidota vienkārša, bet samērā gara. Savukārt, prezentācijas 2. versija “PROGRAMMAS PIEMĒRS: *Scratch* veidota prezentācija v.2 par Jauno datorīku skolas šo e-vidi...” - <http://scratch.mit.edu/projects/24542389> ir ar īsāku, bet pirmajā brīdī grūtāk uztveramu kodu; koda gabals, kas 12 reizes atkārtojas v.1, šajā v.2 ir izdalīts kā jauna procedūra “prezentācijas solis (1)”. Īsāka programma parasti ir labāka, jo tērē mazāk datora resursu. Programmētāji mēdz teikt, ka jebkuru programmu var uzlabot vismaz par vienu operatoru. Savukārt filozofi saka, ka jebkuru lietu var pilnveidot līdz bezgalībai. Reālisti saka, ka vienā brīdī arī neideāla programma ir gana laba lietošanai, ja testēšanas rezultāti ir gana labi...

5. Sakārtosim domas

Pagaidām esam izpētījuši tikai nelielu daļu no *Scratch* 2 iespējām. Nākotnē interesanta var likties *Scratch 2.0* jaunā spēja – datora kameras attēla videojutība, piemēram, var par programmas ievaddatiem pārvērst cilvēka rokas vai ķermeņa kustību, ir iespēja ieimportēt datus no interneta resursiem, realizēt lielu daļu no vecajiem skolas programmēšanas uzdevumiem u.c.

Ne visās programmēšanas valodās darbs notiek, bīdot un konfigurējot blokus, tomēr, pēc *Scratch* apguves būsiet iemācījušies domāt kā programmētājs, tad arī C++ vai *Visual Basic* u.c. jaudīgāku valodu kods neliksies tik nepārskatāms, atliks tikai piemācīties klāt nākamās valodas sintaksi un algoritmus (skat. 23. attēlu). Arī *Visual Basic* viss kods nav jāraksta ar rokām, arī tur ir rīki koda bloku radīšanai.



```
//Excerpt from a Visual Basic application
If strCurrentAction = "FillCircle" Then
    Dim objCoordinates As Rectangle
    objCoordinates = _
    New Rectangle(Math.Min(objEnd.X, objStart.X), _
    Math.Min(objEnd.Y, objStart.Y), _
    Math.Abs(objEnd.X - objStart.X), _
    Math.Abs(objEnd.Y - objStart.Y))
    Pick_Color_And_Draw("FillCircle", objCoordinates)
End If
```

23. att. Programmas koda pieraksta salīdzinājums *Scratch* – pa kreisi, *Visual Basic* – pa labi (Jerry Lee Ford, 2014).

Scratch apgūvē ir populārs sauklis: “Iztēlojies–programmē–koplieto!” (“Imagine–Program–Share!”), tas apraksta procesa posmus. Jāpiebilst, ka koplietota projekta izejas kodu (bloku scenāriju) nav iespējams noslēpt, tas palīdz kopīgiem spēkiem dalīties ar labāko, atrast kļūdas, tādējādi darot pasauli labāku. Viss sākas ar ideju, un tieši realizējamu ideju pasaulē nepietiek, piemēram, ir zināms, ka mobilo telefonu sīklietotnes (APP) programmēt nav pārāk sarežģīti, to

daudzi var iemācīties, un ar to var ļoti labi nopelnīt, bet tikai dažiem izdodas izdomāt ideju, kuras sīklīetotne ienes miljonu. Tam pietiek, ja 1 miljons vied telefonu lietotāju (pasaulē ir jau aptuveni 1 miljards vied telefonu) nopērk programmiņu par nieka 1 EUR (tas taču var būt motivējoši programmēšanas u.c. datorzinātņu studijām).

Izpētiet pamācību un, ja ir interese, paskatiet arī tajā pieminētos materiālus, atrodiēt citus noderīgus interneta resursus par *Scratch* un veidojiet savas programmas! Lai būtu vieglāk orientēties šajā materiālu kopumā, izlasiet 1. e-grāmatu “Scratch izvēles pamatojums un projekta koncepcija” – <http://skolas.lv/mod/book/view.php?id=30181>. Nākotnē ir paredzēts <http://jds.df.lu.lv> turpināt piedāvāt *Scratch* materiālus skolēniem, skolotājiem un studentiem.

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. 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. Letonika Terminoloģijas portāls – <http://termnet.lv>, 2014.
8. 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!
25.05.2014., papildināts 11.07.2014.

Dr. Vadībzin., Doc. Imants Gorbāns