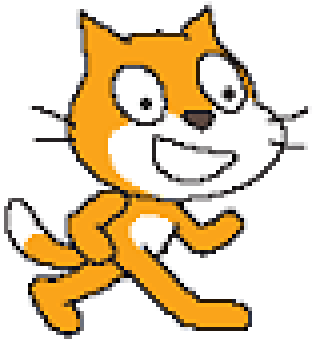
SCRATCH



Cu pași mici spre mari proiecte

Carchilan Lilia
Profesoară de informatică,
IPLT „Tudor Vladimirescu”

Chișinău, 2016

Cuprins

De ce le trebuie copiilor să studieze programarea?

Steve Jobs considera: ,,Toată lumea trebuie să cunoască programarea deoarece aceasta ne face să gîndim". Atunci cînd se vorbeşte despre dezvoltarea logicii şi abilităților analitice, se consideră că este aproape imposibil aceasta. Aşa se întîmplă că unii au ,,gîndire matematică", iar alții nu o au. Cu toate acestea dezvoltarea gîndirii analitice a copilului e posibilă şi în ajutor ne vin limbaje speciale de programare ,,pentru copii".

De foarte multe ori auzim de la părinți fraze ca: ,,Fiica mea va deveni actriță, de ce ea trebuie să cunoască aceasta?", ,,Fiul meu va fi jurist, e o pierdere de timp pentru el."

De fapt sunt trei motive foarte importante care ne face să învățăm copii programarea de la o vîrstă cît mai timpurie.

1. Economie de timp şi forțe

Bazele programării sunt algoritmii. Algoritm este numit un set de acțiuni care trebuie să le realizeze pentru a obține un rezultat. Orice proces, fie că este vorba de lansarea unei rachete, prepararea unei ciorbe, conducerea unei maşine, î-l putem socoti ca algoritm în baza căruia putem elabora un program care î-l va pune în aplicare.

Cum pot folosi acest lucru în viața de zi cu zi? Dacă lucrați pe computer, sunt sigur că de multe ori trebuie să efectuați acțiuni monotone, aproape mecanice (formatarea textului, sortarea dosarelor, trimiterea de e-mail, etc.). Există mai multe soluții care economisesc timp, începând cu macro-urilor în Microsoft Office (un exemplu tipic - formatarea textului), precum şi programele speciale pe care le putem găsi cu uşurință în ,,Automatizarea în Windows". Ele nu efectuiază doar lucrul în locul nostru, dar o pot face la o oră stabilită, chiar dacă nu suntem la computer. Apropo, nu se limitează în mod necesar la locul de muncă: de exemplu, utilizând programul Sikuli, putem crea un ,,robot" şi o comandă simplă să-l învețe să colecteze şi să vândă recolta în ,,Ferma veselă", sau orice alt joc on-line. Vă lăsați computerul să lucreze peste noapte, şi dimineață robotul va cîştiga un sac de bani. Sunt sigur că vă veți bucura.

Ce rol are programarea în toate acestea? Avînd în vedere că chiar şi o cunoaştere superficială în acest domeniu ne permite să ,,gîndim algoritmic" să cautăm procese care pot fi automatizate, şi pentru a găsi opțiuni pentru modul în care acest lucru poate fi realizat, şi pur şi simplu a scrie un program pentru a rezolva orice probleme, dacă este necesar. Deci, cu ce nu s-ar ocupa copilul, astfel de cunoştințe nu vor fi niciodată de prisos.

Apropo, cu un algoritm simplu, ne cunoaştem cu mult timp înainte de a începe să lucrăm pe calculator. Bine-cunoscut din copilărie, ,,lumină roşie - nici o mişcare (stop)!" - un exemplu tipic de algoritm. Ele uşurează foarte mult viața noastră de rutină, care permite automatizarea proceselor pentru a elibera creierul pentru lucru mult mai important. Fără astfel de ,,algoritmi integrați" ne-m pomeni în locul omidei, care reflectă asupra modului cum ea merge fără a se poticni, imediat încurcîndu-şi picioarele...

2. Viața este un lucru imprevizibil

Imaginați-vă că nu ştiți să citți şi să scrieți. Da, chiar acum, în anul 2016. Niciodată nu v-ați gîndit să deveniți un scriitor sau poet, şi deaceea nu a-ți învățat alfabetul. Deci, să scrieți o plângere, sau să citiți contractul bancar deasemeni nu ve-ți putea. Iar dacă aveți nevoie pentru a scrie un mesaj, vă duceți la un prieten care ştie să citească şi să scrie pentru a vă ajuta. Sună ridicol, nu? Posibil că peste 10-20 de ani, aşa se vor simți oamenii care nu cunosc elementele de bază ale programării.

Reflectați: acum 20 de ani, existența unui astfel de discipline ca programarea, era cunoscut numai persoanelor deosebite. 10 ani în urmă, puțini oameni ştiau cine sunt

programatorii şi cu ce de fapt ei se ocupau. Astăzi programator este una dintre cele mai populare şi căutate specialităţi. Dacă dezvoltarea şi progresul vor merge în acelaşi ritm, probabil, în următorii 20 de ani, limbaje de programare vor fi necesare pentru o cariera de succes la fel ca şi o limbă străină.

În plus, tehnologiile informaţionale tot mai profund se infiltrează în viaţa de zi cu zi, dispozitivele care anterior au fost observate numai în filmele de fantastică, au apărut pe rafturile magazinelor obişnuite. Roboţi de asistenţă, dispozitivele ,,caselor inteligente" şi calculatoarele de la bordul maşinilor - toate acestea sunt realitatea de astăzi, şi pentru a le utiliza în mod independent, avem nevoie de noi cunoştinţe.

3. Dezvoltarea copilului

Citind probabil acest articol, te vei gîndi: ,,Acest lucru este, desigur, foarte bun, dar să încurajezi copilul să fie programator pentru perspectivele incerte din viitor? Nu, mulţumesc." Cred că mulţi dintre voi programarea se asociază cu un ecran negru, un set obscur de litere şi numere, precum şi erorile veşnice ,, Syntax Error", cu care vă ciocneaţi la lecţiile de informatică.

Într-adevăr, până de curând, pragul de a intra în programare a fost destul de mare: în cele mai multe programe este limba engleză, sintaxă complexă, interfeţe înfricoşătoare cu o mulţime de ferestre. Programarea a fost dificil să captiveze chiar şi elevii din clasele mai mari, nu mai vorbim de copii. Astăzi, cu toate acestea, s-au creat limbajele de programare uimitoare, care sunt în măsură să intereseze chiar şi un elev de clasa întîi sau vîrstă preşcolară.

Limbajele, care vor fi discutate mai jos, nu numai va familiariza copilul cu elementele de bază ale programării, dar, de asemenea, îl va ajuta să-şi dezvolte logica, orientarea spaţială, atenţia şi imaginaţia. În plus, lucrînd cu acest limbaj, copilul nu doar va experimenta şi va primi plăcere de la proces, dar, de asemenea, va obţine un rezultat unic, care poate fi demonstrat părinţilor şi prietenilor, iar acest lucru este atât de important! Să începem.

Scratch - un mediu vizual de programare orientat-obiect pentru elevii din clasele primare şi gimnaziale. Numele vine de la cuvântul ***scratching*** – tehnică din echipamentul folosit de DJ hip-hop, care mişcă discuri de vinil înainte şi înapoi cu mâinile pentru a se amesteca temele muzicale.

Scratch creat ca o continuare a ideilor limbajului ***Logo*** şi jocului ***Lego***. ***Scratch1*** a fost scris în ***Squeak***, ***Scratch2*** este axat pe activitatea on-line şi rescris în Flash / Activescript. Scratch este dezvoltat de o echipa mica de programatori pentru copii de la MIT. Versiunea curentă - ***Scratch 2.0***, lansat la 9 mai 2013.

În 2008, Scratch a fost adaptată pentru microcontrolerul modulului ***Arduino***. Proiectul se numeşte ***S4A***.

Programele în ***Scratch*** constau din blocuri grafice, scripturile cărora depind de limba de interfaţei selectate. Puteţi selecta una din cele 50 de limbi de interfaţă, inclusiv româna. Pentru a conecta interfaţa în noua limbă utilizează fişierele ***gettext*** standard. https://scratch.mit.edu/

Pe baza codului sursă Scratch 1.4 au fost unele modificări la limbaj, cum ar fi:

- BYOB (în prezent mai merge!)
- Panther
- Slash (modificarea BYOB)

BYOB (Snap!) Dezvoltat la Universitatea din Berkeley. Extinderea majoră a limbajului, care a fost introdus in BYOB a fost posibilitatea de a construi unităţi personalizate compozite - proceduri analogice limbi de programare convenţionale. Aceasta conţine circuitul, recursivitate şi expresia lambda. Adăugate, de asemenea, un program de depanare şi capacitatea de a compila în fişiere executabile ataşate sprites, liste multidimensionale, îmbunătăţind defilarea şi compilarea de executre. Incepand cu versiunea 3.1 se adauga suport pentru POO BYOB - sprites

BYOB acum permite moştenire bazate pe prototipuri. Pentru a face acest lucru în limbaj a fost introdus mecanismul de clonare a sprites.

Snap! este un limbaj educațional de programare vizuală în browser care permite crearea unor animații, jocuri, implementarea unor algoritmi, rezolvarea unor probleme matematice şi nu numai.

Pentru a realiza o aplicație folosind Snap! trebuie accesată adresa: http://snap.berkeley.edu/snapsource/snap.html

Ghidul de utilizare se găseşte la adresa: http://snap.berkeley.edu/SnapManual.pdf

Interfața are o bară de control a aplicației şi trei coloane:

- prima conține blocurilor care pot fi adăugate în aplicație,
- a doua oferă informații despre elementul selectat şi permite adăugarea blocurilor care manipulează elementul selectat,
- a treia conține scena (partea vizuală a aplicației) şi elementele care fac parte din scenă (tot aici

se pot adăuga elemente noi în scenă).

Elementele grafice din scenă se numesc sprite. Spriteurile create pot şi selectate şi manipulate prin adăugarea blocurilor în secțiunea (tabul) scripts din zona centrală (coloana a doua).

Realizarea unui program în ***Scratch*** sau ***Snap!*** presupune adăugarea unor blocuri care sunt organizate în 8 categorii:

Categorie	Detalii
Motion	Deplasarea, rotirea şi poziționarea sprite-urilo
Looks	Afişarea mesajelor, controlarea vizibilității, redimensionarea elementelor, schimbarea imaginii unui sprite (costumului)
Sound	Redarea şi oprirea redării sunetelor, generarea sunetelor pe baza notelor muzicale
Pen	Desenarea cu ajutorul unui "creion" (creionul este sprite-ul în care se folosesc blocurile din această categorie)
Control	Tratarea evenimentelor (apăsarea butonului de start, apăsarea unei taste sau apăsarea mouse-ului), utilizarea instrucțiunilor condiționale (if) şi repetitive (repeat)
Sensing	Detectarea coliziunilor sprite-urilor, interacțiunea cu utilizatorul
Operators	Lucrul cu numere, operații matematice şi booleene, şi şiruri de caractere
Variables	Crearea şi manipularea variabilelor, a listelor şi a blocurilor

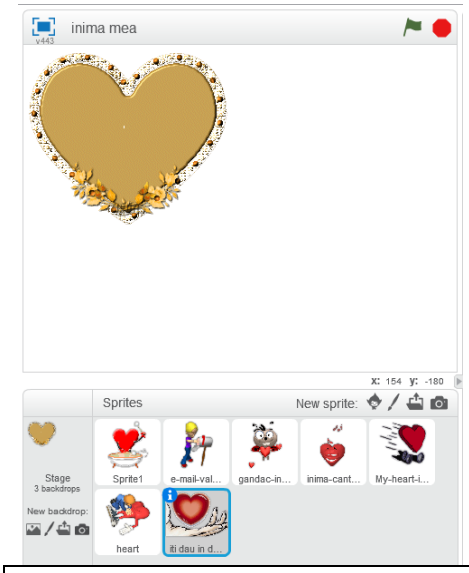
Scratch, BYOB şi Snap! Sunt trei aplicații diferite dar care au la bază aceleaşi principii şi aceleaşi blocuri cu ajutorul cărora se pot elabora diferite jocuri, filmulețe animate descoperind magia programării.

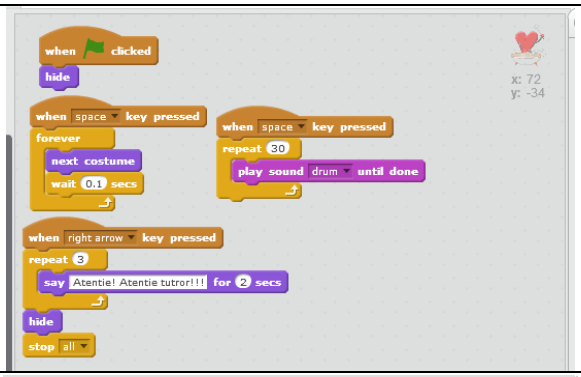
Copii de la o vîrsta preşcolară sunt atraşi de magia creării unor istorioare hazlii şi interesante pe care să le împărtăşe cu prietenii. Dar nu numai copii sunt captivați de magia creării unor jocuri sau poveşti pentru dezvoltarea creativității copiilor dar şi părinții, profesorii deoarece în era infiltrării tehnologiilor informaționale în toate ramurile economiei naționale şi la toate treptele educaționale pentru a fi în pas cu progresul tehnico-ştiințific şi a crea o generație aptă de a utiliza toate tehnicile şi tehnologiile informaționale este nevoie de a utiliza de la o vîrstă cît mai fragedă în programa educațională limbaje de programare uşor de utilizat cu o întefrață simplă iar crearea algoritmilor, sau scrierea codurilor să fie distractivă şi atractivă.

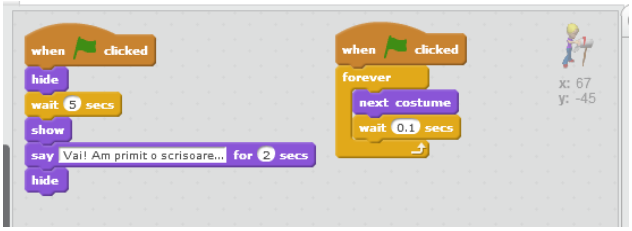
Să explorăm împreună magia programării distractive! Vă propun un proiect simplu în care vom utiliza 8 imagini animate (.gif) pentru actori şi fundal. Vom folosi doar cîteva elemente de mişcare, vom adăuga puțin sunet şi cum spun francezii: „voilà" ***proiectul „Inima mea"***

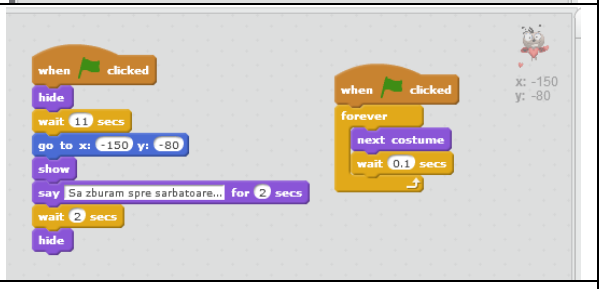
Proiectul „Inima mea”

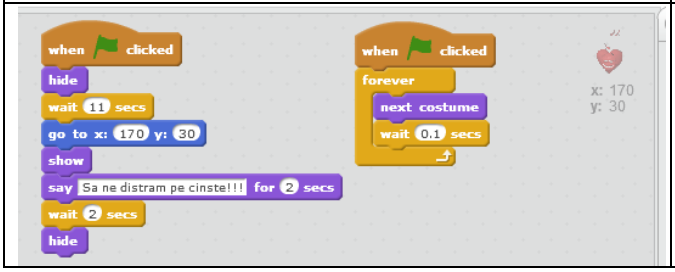
https://scratch.mit.edu/projects/98166405/

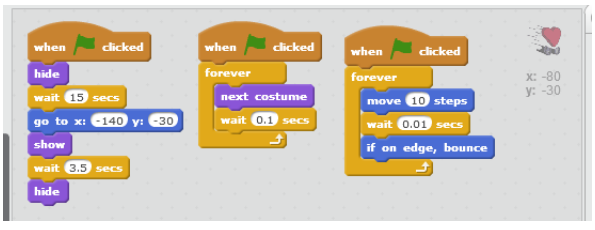


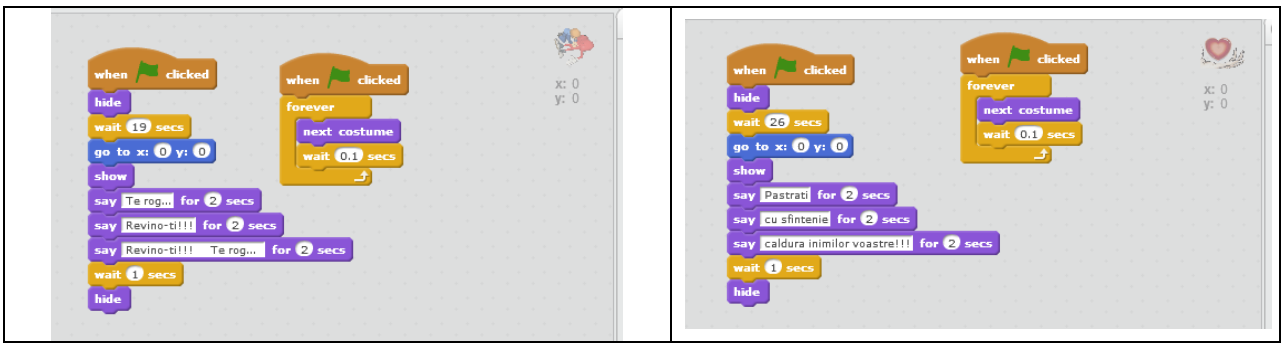






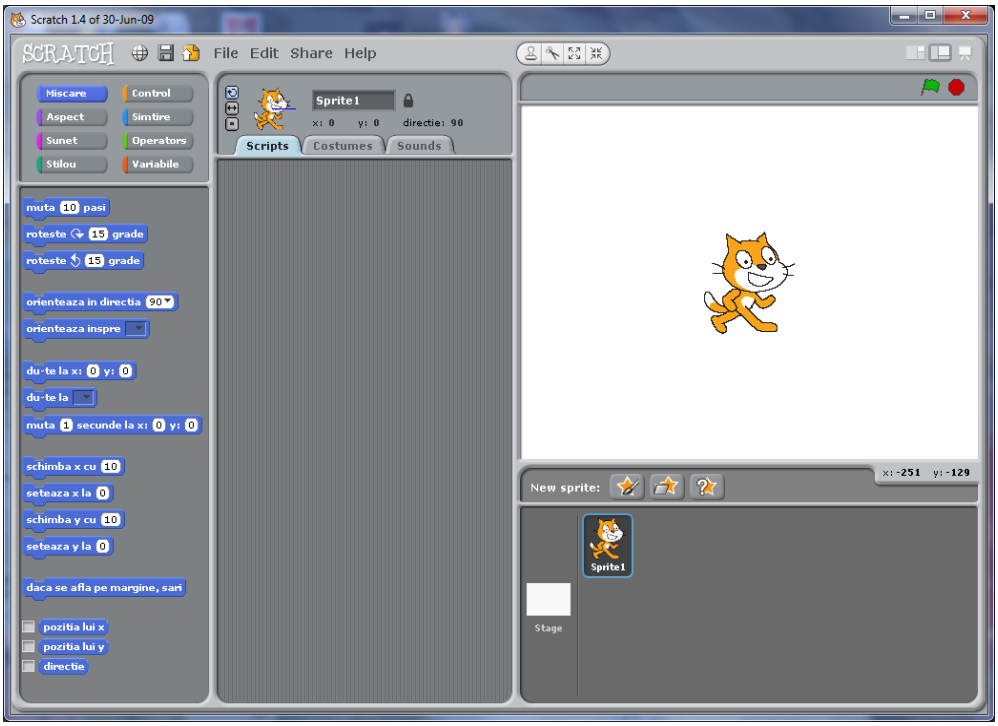






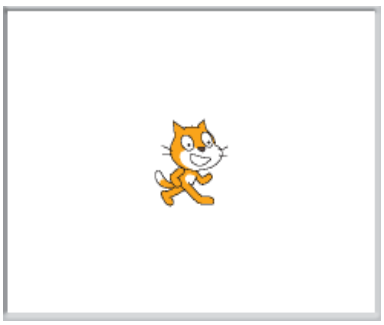
Pentru a monta acest proiect vom avea nevoie de unele **repere teoretice**.

Ne putem face o prima impresie a mediului de programare Scratch prin imaginea 1.0. În sus putem vedea meniurile File, Edit, Share şi Help menite să salveze, editeze, împărtăşească etc. aplicaţia dezvoltată.



Imaginea 1.0. Mediul de programare Scratch1.4. Interfaţa principală

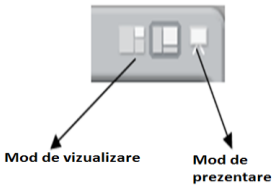
Scena - este locul în care poveştile, jocurile şi animaţiile prind viaţă. Actorii se mişcă şi interacţionează unii cu alţii pe scenă.



Imaginea.1.1

Modul prezentare şi ***modul vizualizare***-modul prezentare este modul folosit pentru prezentarea proiectului, scena ocupând toată suprafaţa ecranului.

Cu ajutorul celor două butoane al modului de vizualizare se poate schimba mărimea scenei (între o scenă mică şi o scenă mare).

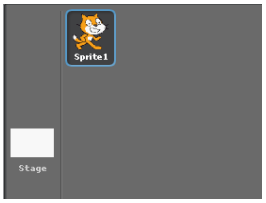


Imaginea.1.2

Butoane pentru crearea de actori noi- când începi un proiect Scratch, acesta începe cu un singur actor, o pisică. Pentru a crea actori noi se foloseşte unul din cele 3 butoane: desenare actor, alegere din fişier sau actor surpriză.

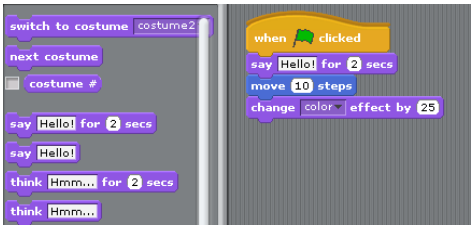
Imaginea.1.3

Lista cu actori-lista cu actori afişează miniaturi pentru toţi actorii din proiect. Numele fiecărui actor apare sub miniatură.



Imaginea.1.4

Paleta de blocuri şi ***Zona de scripturi***- pentru a programa un actor, se trag blocuri din Paleta de blocuri peste Zona de scripturi. Pentru a rula un bloc se face clic pe el.

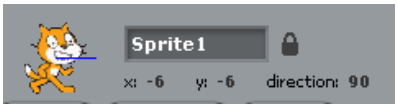


Imaginea.1.5

Taburi pentru editarea Costumelor şi Sunetelor-pentru a vedea şi edita costumele actorului se fac clic pe tabul Costume, iar pentru a vedea şi sunetele actorului se fac clic pe tabul Sunete.

Imaginea.1.6

Informaţii despre actorul curent-informaţiile afişate despre actorul curent sunt: numele actorului, poziţia x-y, direcţia, starea de blocare şi starea stiloului. În caseta text se poate tasta un nou nume pentru actor.



Imaginea.1.7

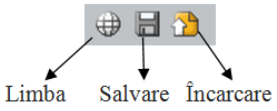
Stiluri de rotire ale actorului-cu ajutorul butoanelor Stil de Rotire se poate controla modul în care este afişat costumul în momentul în care actorul îşi modifică direcţia.

Imaginea.1.8

Bara de instrumente-bara de instrumente conţine butoane cu ajutorul cărora se pot selecta instrumente pentru realizarea următoarelor acţiuni: duplicare, ştergere, mărire şi micşorare.

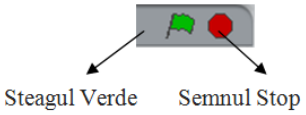
Imaginea.1.9

Limba, Salvare şi Încărcare-dacă se face clic pe iconiţa Limbă, se poate modifica limba în care este afişată interfaţa utilizator. De pe iconiţa Salvare se poate salva proiectul, iar cu ajutorul iconiţei Încărcare, se poate încărca proiectul pe website-ul Scratch pentru a-l împărtăşi cu membrii comunităţii.



Imaginea.1.10

Steagul Verde şi ***Semnul Stop***- Steagul Verde oferă o modalitate de a începe mai multe scripturi în acelaşi timp, iar Semnul Stop este folosit pentru oprirea rulării scripturilor.

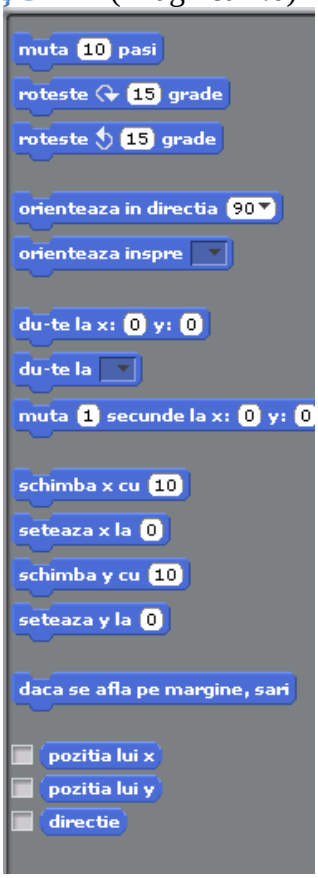


Imaginea.1.11

Imaginea. 1.12. Panoul cu instrucţiuni

În stînga sus (sau imaginea. 1.12) putem vedea panourile cu instrucţiunile necesare pentru a dezvolta o aplicaţie în Scratch cum ar fi panoul Motion folosit pentru locomoţia obiectelor, a personajelor fiind denumite SPRITE.
Panoul Looks (Aspect) folosit pentru a afişa mesaje pe ecran, pentru a schimba înfaţişări etc.
Panoul Sound (Sunet) constă în instrucţiuni care ne ajută să compunem melodii, să scoatem sunete sau să înregistrăm voci.
Panoul Pen (Stilou) conţine instrucţiuni pentru a desena, trage lenii etc.
Panoul Control şi cel mai important gazduieşte instrucţiunile de control cum ar fi: IF, IF ELSE, REPEAT UNTIL, FOREVER IF etc.
Panoul Sensing (Simţire) este punctul cel mai important al acestui limbaj şi unul din caracteristicile care îl face aşa este că putem face aplicaţii dezvoltate iterativ.
Panoul Operators (Operatori) conţine semnele de adunare, scădere, înmulţire, inpărţire şi pe lîngă ele instrucţiunile: MOD, RANDOM, JOIN, SQRT, AND, OR, NOT etc.
Panoul Variabiles (Variabile), conform şi numelui ne dăm seama că aici este locul declarării variavilelor şi dispunem de instrucţiuni speciale pentru lucrul cu ele. Pe lîngă variabile aici mai avem şi instrucţiuni speciale pentru lucru cu vectorii, aici sunt denumite liste.

Vom începe cu **panoul MIŞCARE** (imaginea 2.0)



Imaginea 2.0

Instrucțiunea ***(MUTĂ) MOVE*** (imagine 2.1) este folosită pentru locomoția Sprite-ului. El se va deplasa în funcție de valoarea dată de noi.

Imaginea 2.1

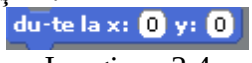
Instrucțiunea ***SCHIMBĂ X (CHANGE X)*** (imaginea 2.2) face ca Sprite-ul să se deplaseze în direcția coordonatei X dacă valoarea coordonatei X este pozitivă şi în sens opus dacă valoarea coordonatei X este negativă.

Imaginea 2.2

Instrucțiunea ***SCHIMBĂ Y (CHANGE Y)*** (imaginea 2.3) mobilizează Sprite-ul în direcția coordonatei Y, în sus dacă valoarea este pozitivă şi în jos dacă valoarea este negativă.

Imaginea 2.3

Instrucțiunea ***SCHIMBĂ X, Y (GOTO X,Y)*** (imaginea 2.4) duce Sprite-ul direct în locația specificată de noi prin coordonatele X şi Y.



Imaginea 2.4

Panoul ASPECT (imaginea 3.0) conține instrucțiuni legate în general de înfățișarea sau aspectul Sprite-ului.

Imaginea 3.0

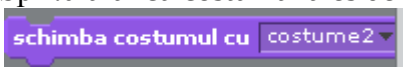
Instrucțiunea ***SPUNE PENTRU n SECUNDE (SAY FOR n SECONDS)*** (imaginea 3.1) va afişa pe ecran un text timp de n secunde. Pentru programatorii limbajuliu Pascal este identică cu Write();

Imaginea 3.1

Instrucțiunea ***GÎNDEŞTE PENTRU n SECUNDE (THINK FOR n SECONDS)*** (imaginea 3.2) este identică cu proprietatea instrucțiunii SAY, numai că modul de afişare a textului şi anume textul apate ca nor.

Imaginea 3.2

Instrucțiunea ***SCHIMBĂ COSTUMUL CU (SWITCH TO COSTUME)*** (imaginea 3.3) schimbă înfățișarea sau aspectul Sprite-ului cu costumul ales de noi.



Imaginea 3.3

Instrucțiunea ***URMĂTORUL COSTUM (NEXT COSTUME)*** (imaginea 3.4) face acelaşi lucru precum ***SWITCH TO COSTUME*** doar că nu avem oportunitatea de a schimba costumul dorit ci îl ia automat pe următorul costum.

Imaginea 3.4

Instrucțiunea ***ASCUNDE (HIDE)*** (imaginea 3.5) ascunde Sprite-ul sau îl face invizibil. Această instrucțiune este folosită în general în animații sau în jocuri pentru a face ca personajele să dispară.

Imaginea 3.5

Instrucțiunea ***AFIŞEAZĂ (SHOW)*** (imaginea 3.6) face ca Sprite-ul să apară în fața ochilor în cazul în care este în modul ***HIDE***.

Imaginea 3.6

Panoul CONTROL

Este panoul (imaginea 4.0) care conține instrucțiunile de control în llimbajul Scratch. Aici se gasesc instrucțiuni clasice cum ar fi: IF, IF-ELSE, REPEAT precum şi altele care sînt particulare doar limbajului Scratch.

Imaginea 4.0

Instrucțiunea LA APĂSAREA FLAGULUI VERDE (WHEN GREEN FLAG CLICKED) (imaginea 4.1) este acea instrucțiune care dă START-ul aplicației. L-am putea asemăna cu opțiunea RUN a celorlalte medii de programare.

Imaginea 4.1

Instrucțiunea LA APĂSAREA TASTEI SPACE (WHEN SPACE CLICKED) (imaginea 4.2) ca funcție este identică cu cea precedentă numai că aici în loc de steguleţul verde poate fi oricare caracter din tastatură inclusiv săgeţile de orientare (stinga, dreapta, jos, sus) sau tasta SPACE. Atunci cînd va fi apăsat acodul de sub el va fi rulat.

Imaginea 4.2

Instrucțiunea PENTRU TOTDEAUNA (FOREVER) (imaginea 4.3) este o instrucțiune repetitiv infinită. Acest infinit loop este folosit în general pentru mişcarea personajelor în stînga-dreapta, sus-jos din jocurile video şi anume v amerge MEREU la stînga cînd săgeata cu direcție stînga va fi tastată.

Imaginea 4.3

Instrucțiunea REPETA (REPEAT) (imaginea 4.4) este tot una repetitivă dar nu infinită. Ea va repeta ciclu de cîte ori dorim.

Imaginea 4.4

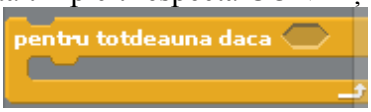
Instrucțiunea EXPEDIERE LA TOȚI (BROADCAST) (imaginea 4.5) este una particulară limbajului Scratch. L-am putea asemăna cu un emițător. Ea emite un mesaj care va fi la rîndul lui receptat.

Imaginea 4.5

Instrucțiunea LA PRIMIRE (WHEN I RECEIVE) (imaginea 4.6) este receprorul care primeste mesajul trimis prin BROADCAST. După receptarea mesajului ea va rula codul de sub el.

Imaginea 4.6

Instrucțiunea PENTRU TOTDEAUNA DACĂ (FOREVER IF) (imaginea 4.7) va executa MEREU instrucțiunile atîta timp cît respectă CONDIȚIA pusă de noi.



Imaginea 4.7

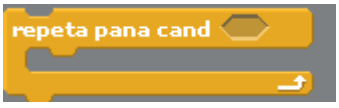
Instrucțiunea DACĂ (IF) (imaginea 4.8) este acea instrucțiune care execută codul atunci cînd condiția este îndeplinită.

Imaginea 4.8

Instrucțiunea DACĂ ATUNCI (IF ELSE) (imaginea 4.9) va executa codul atunci cînd condiția este respectată şi contrar atunci cînd nu se respectă condiția 1.

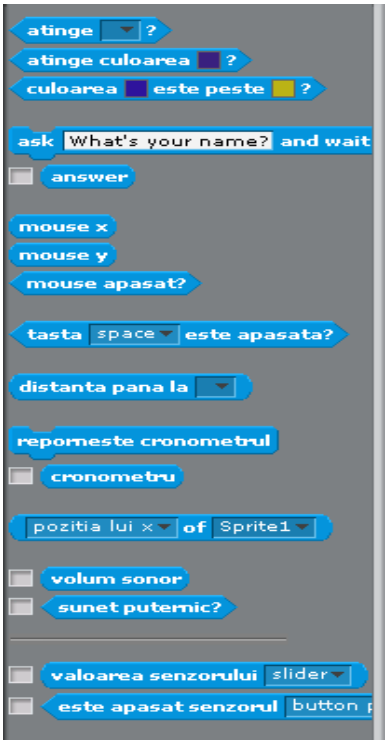
Imaginea 4.9

Instrucțiunea REPETĂ PÎNĂ CÎND (REPEAT UNTIL) (imaginea 4.10) v-a repeta ciclul pînă condiția nu va mai fi respectată. Instrucțiunea repetitivă există intr-un format asemănător şi în Pascal.



Imaginea 4.10

Panoul SIMȚIRE



Imaginea 5.0

Instrucțiunea ASK (imaginea 5.1) ca funcție este asemănătoare cu Writeln() din limbajul Pascal numai că diferă puțin. Este folosită în scop interactiv. Pune o întrebare şi apoi primeşte un răspuns.

Imaginea 5.1

Instrucțiunea RĂSPUNS (ANSWER) (imaginea 5.2). După ce este pusă întrebarea prin intermediul instrucțiunii ASK răspunsul luat este transportat prin instrucțiunea ANSWER. În cazul în care noi vrem să memoram răspunsul primit într-o variabilă inițializăm variabila cu instrucțiunea ANSWER (RĂSPUNS).

Imaginea 5.2

Instrucțiunea ATINGE (TOUCHING) (imaginea 5.3) este una particulară limbajului Scratch şi are rol de simț atunci cînd Sprite-ul atinge ceva.

Imaginea 5.3

Instrucțiunea ATINGE CULOAREA (TOUCHING COLOR) (imaginea 5.4) este aproape identică cu TOUCHING doar ca simte numai culoarea.

Imaginea 5.4

Instrucțiunea TASTA SPACE ESTE APĂSATĂ (KEY PRESSED) (imaginea 5.5) verifică dacă una din litere, spațiu, săgeți de orientare este tastată iar dacă DA atunci va executa codul de sub el.

Imaginea 5.5

Panoul OPERATOR

Imaginea 6.0

Instrucțiunile cu cei patru operatori adunare, scădere, înmulțire şi împărțire(imaginea 6.1) sunt folosite pentru calculele cu numere.

Imaginea 6.1

Instrucțiunea ALEGE UN NUMĂR ALEATORIU (PICK RANDOM) (imaginea 6.2) serveşte pentru a genera un număr aleator.

Imaginea 6.2

Instrucțiunile MAI MARE, EGAL şi MAI MIC (imaginea 6.3) sunt folosite pentru a face comparații logice.

Imaginea 6.3

Instrucțiunile ŞI, SAU, NU (AND, OR şi NOT) (imaginea 6.4) operatori logici şi sunt utilizați pentru a determina logica între valori.

Imaginea 6.4

Instrucțiunea JOIN (imaginea 6.5) este una particulară limbajului Scratch şi este folosită pentru a împreuna două texte, un text şi o variabilă sau pentru a aduce două lucruri una lîngă alta.

Imaginea 6.5

Instrucțiunea MOD (imaginea 6.6) are funcție de a obține ca rezultat restul împărțirii.

Imaginea 6.6

Panoul VARIABILE

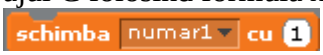
Imagine 7.0

Acest panou serveşte la declararea variabilelor (ştergerea lor) venind cu o listă de instrucţiuni speciale în lucrul cu variabilele.

Instrucţiunea SRTEAZĂ LA (SET TO) (imaginea 7.1) este folosită pentru a iniţializa o variabilă cu o valoare sau cu răspunsul dat prin intermediul lui ANSWER (RĂSPUNS).

Imaginea 7.1

Instrucţiunea SCHIMBĂ CU (CHANGR BY) (imaginea 7.2) are ca menire incrementarea unei valori din unu în unu, din doi în doi etc. după dorinţa noastră. Este identică ca funcţie cu INC(x) din Pascal sau x++ din limbajul C folosind formula x=x+1.



Imaginea 7.2

Bazele unui proiect SCRATCH

În interiorul fiecărui proiect SCRATCH avem următoarele “elemente importante”:

- actori,
- costume,
- blocuri,
- script-uri,
- scenă.

Modul în care combinăm aceste elemente ţine de imaginaţia noastră şi crează povestiri captivante, animaţii şi jocuri.

Actorii sunt cei care aduc la viaţă programul nostru şi fiecare proiect are cel puţin un actor. Vom vom determina cum să adăugăm şi să personalizăm actorii pe parcursul lucrării.

Un actor poartă în orice moment un ***costum***. Schimbând costumul, schimbăm modul în care este afişat actorul. Dacă actorul este chiar scena, costumul este de fapt ***fundalul scenei***.

Blocurile sunt de fapt instrucţiuni care fac parte din diverse categorii.

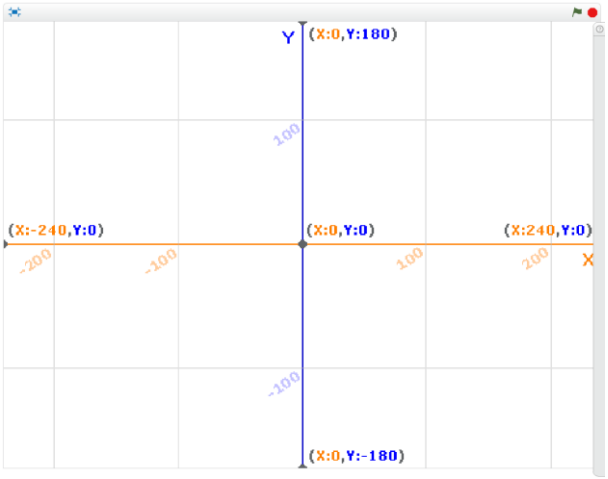
Script-urile sunt o combinaţie de blocuri care instruiesc un actor ce să facă la un moment dat. Fiecare bloc reprezintă o instrucţiune sau o bucată de informaţie care afectează actorul într-un anumit mod.

Fiecare actor dintr-un program SCRATCH intră pe ***scenă*** şi "recită" un set de linii dintr-un script. Modul în care interacţionează actorii depinde de deciziile programatorului. Pe scena SCRATCH, fiecare obiect, chiar şi o piatră plasată într-un colţ al scenei, este un actor care poate contribui la crearea oricărui proiect.

Adăugînd blocul schimbă x cu (change x by) pentem determina actorul să se mişte pe direcţia orizontală spre partea dreaptă a scenei. Cu cât valoarea din acest bloc este mai mare, cu atât mai mare este deplasarea spre dreapta la fiecare execuţie.

Apoi adăugăm blocul schimbă y cu (change y by) şi am introdus o valoare negativă. Când se executat click pe flag, actorul îşi continuat mişcarea orizontală spre dreapta, dar se deplasează şi în jos.

Valorile numerice pentru x şi y pe care le putem seta în anumite blocuri sunt exprimate în pixeli. Dimensiunea scenei este de 480 pixeli lăţime şi 360 pixeli înălţime.



SCRATCH utilizează axele X şi Y pentru a împărţi ecranul în patru cadrane. La deplasarea unui actor, valorile pozitive determină mişcare spre dreapta sau în sus, iar cele negative spre stânga sau în jos.

Unitatea de măsură în sistemul de coordonate sunt pixelii. Fiecare actor de pe scenă este plasat la anumite coordonate X, Y. Este posibilă atât localizarea unui actor cât şi trimiterea unui actor la anumite coordonate de pe scenă.

În majoritatea proiectelor un singur actor nu poate fi responsabil pentru realizarea tuturor acţiunilor noastre, ceea ce înseamnă că frecvent trebuie să adăugăm actori pentru a ne îndeplini scopurile.

Putem adăuga actori la scenă în unul dintre următoarele patru moduri:

- pictăm un nou actor;
- alegem un nou actor dintr-un fişier ;
- preluăm un actor - surpriză;
- duplicăm un actor.

Metoda prin care adăugăm un actor depinde, desigur, de ceea ce dorim să realizăm şi de ceea ce am hotărât că este necesar proiectului nostru.

Când un actor transmite un mesaj, acesta este o cheie pentru ceilalți actori că trebuie să se întâmple ceva. Pentru ca un alt actor să reacționeze la mesaj, trebuie să i se spună să asculte acest mesaj, folosind un bloc de control when I receive (la primire). Un singur mesaj de broadcast (expediere la toți) poate controla mai mulți actori. Dacă un actor nu este instruit să asculte un anumit mesaj, acesta va ignora mesajul.

Aceste mesaje de broadcast le puteți privi ca pe niște inițiatori de conversație. O conversație necesită doi sau mai mulți participanți. Fiecare dintre participanți, după ce recepționează mesajul, poate reacționa în unul dintre următoarele moduri:

- emite la rândul lui un nou mesaj ;
- reacționează la mesaj fără a emite un mesaj nou ;
- ignoră mesajul.

Operații cu imagini

SCRATCH importă toate formatele populare (png, bmp, jpg şi gif). Scena în SCRATCH are dimensiunile de 480 pixeli lățime pe 360 pixeli înălțime. În prelucrarea imaginilor, mai întâi precizăm lățimea deci, o imagine de 800x600 va fi o imagine de 800 pixeli lățime. Dacă mărim foarte mult imaginea, s-ar putea să ajungem să vedem punctele individuale (pixelii). Aceste puncte conțin informațiile de care computerul are nevoie pentru a afişa imaginea pe ecran.

Contorizarea pixelilor devine importantă pentru selectarea unor imagini care satisfac nevoile noastre. Dacă vrem o imagine care să ocupe tot fundalul, aceasta trebuie să aibă cel puțin 480x360 pixeli pentru a fi siguri că avem o imagine de calitate acceptabilă.

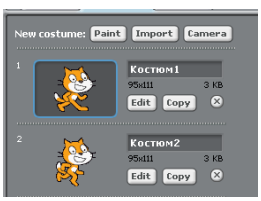
Problema cu utilizarea unei imagini cu o dimensiune mai mică de 480x360 pixeli este că trebuie să o mărim pentru a umple ecranul. Pe măsură ce mărim o imagine, pixelii devin din ce în ce mai mari şi vom începe să putem vizualiza pixelii individuali. Acest efect crează o imagine granulată şi neclară, numită frecvent imagine pixelată.

Partea bună este că dacă pornim cu o imagine de dimensiune mai mare de 480x360, nu trebuie să o redimensionăm. SCRATCH o va redimensiona automat la dimensiunile corecte. Nu trebuie să ne punem problema pixelării atunci când micşorăm o imagine. În esență, începem întotdeauna cu o imagine mai mare decât este necesar şi o micşorăm dacă trebuie.

În SCRATCH, avem două moduri de a determina dacă imaginea noastră este suficient de mare. Pe de o parte, putem folosi abordarea simplă. Dacă afişăm imaginea ca fundal al scenei şi mai există zone din scenă care sunt vizibile, înseamnă că imaginea este prea mică. Această metodă funcționează dar nu este foarte precisă.

Pe de altă parte, tab-ul Backgrounds afişează dimensiunea în pixeli sub numele imaginii, aşa cum putem observa în imaginea de mai jos:

Dacă privim atent imaginea, putem observa că imaginea este de 95x11 pixeli. Este mai mică atât pe înălțime cât şi pe lățime. Nu e o mare problemă, o putem lăsa aşa, sau, dacă preferați, puteți utiliza Paint Editor pentru a o umple cu o culoare solidă.



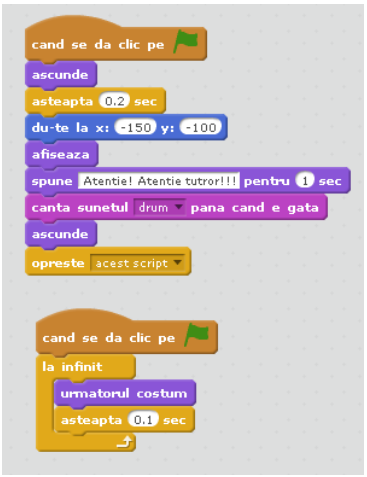
Primul program în SCRATCH

Orice nou proiect conține în mod automat un singur actor (Sprite1) reprezentând o imagine a unui pisoi. În proiectul următor vom utiliza alți actori, iar pisica de pe scenă o ştergem cu ajutorul comenzii delete.

Vom prezenta un prim proiect şi anume să punem actorii de pe scenă în mişcare. Astfel trebuie executați următorii paşi:
1.În paleta de blocuri, execută click pe butonul Looks.
2.Se trasează blocul switch to costume în zona de script-uri.
3.Se execută click pe butonul Control din paleta de blocuri.
4.Se trasează blocul when flag clicked în zona de script-uri şi apoi se lipeşte deasupra blocului adăugat anterior.

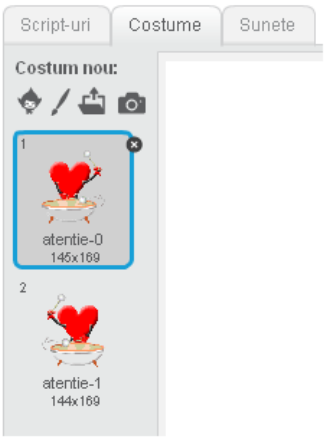
Cum se lipesc blocurile?

Când se trasează un bloc deasupra altui bloc, o linie albă este afişată ca indiciu că blocul trasat poate fi adăugat la script în locul respectiv. Când se eliberă mouse-ul blocul se lipeşte automat.



Imaginea.8

În secțiunea pentru script-uri, se execută click pe tab-ul ***Costumes*** pentru a afişa costumele actorului curent.

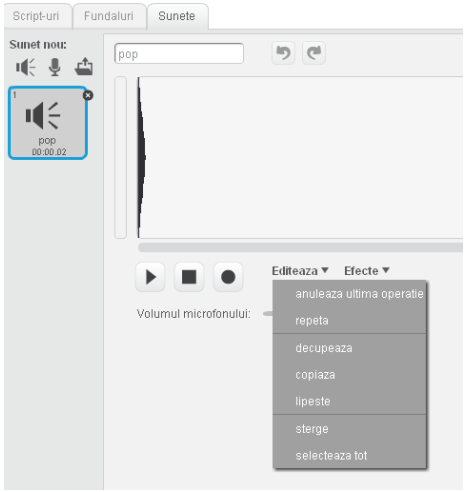


Imaginea.9

Dacă executăm click pe ***Scena*** se obține fundalul scenei unde se va desfăşura acțiunea.

Imaginea.10

Se execută click pe "actorul" numit ***Scena*** selectăm ***Sounds*** apoi comanda Import şi atribuim lui ***Scena*** cîntecul dorit.



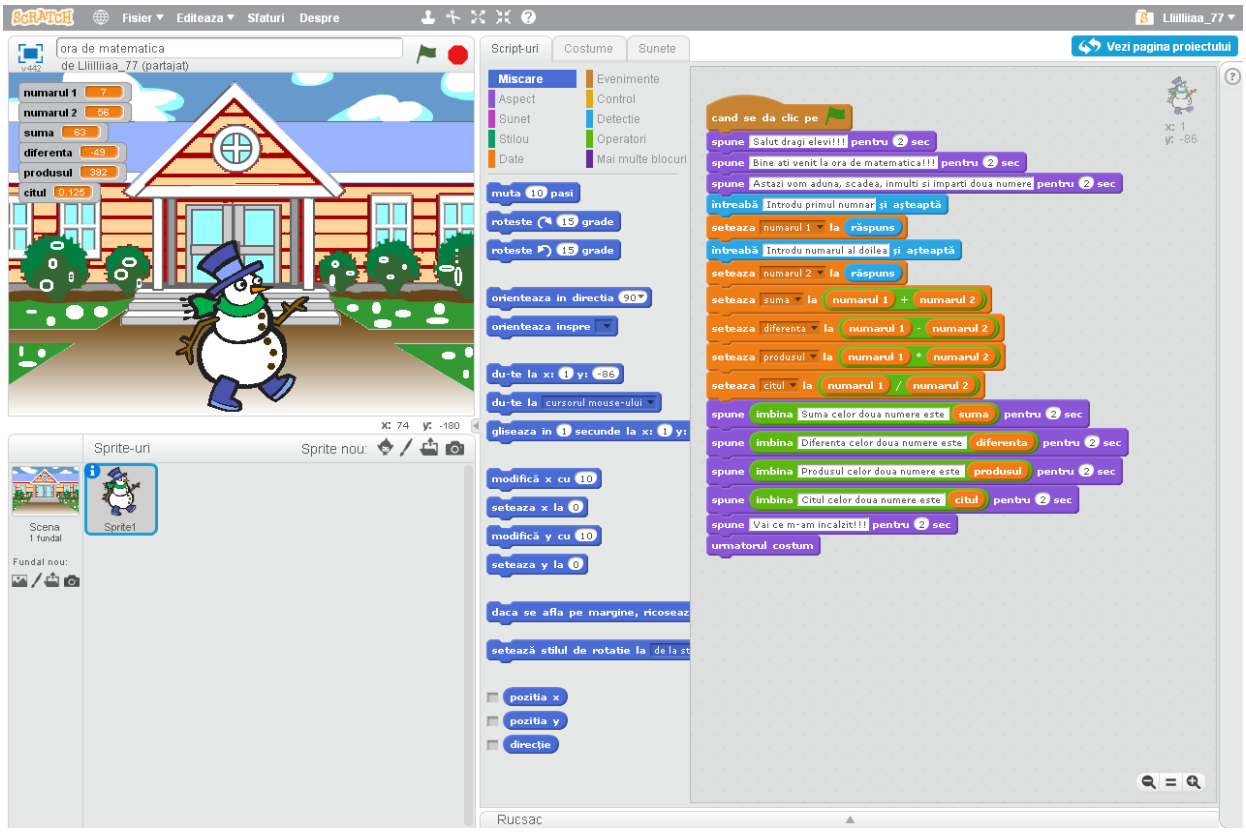
Imaginea.11

Se execută click pe ***flag-ul (steagul)*** din partea de sus (dreapa) a scenei pentru a vedea în acțiune primul script SCRATCH.

Vă propunem să experimentați cîteva exemple de proiecte:
Proiectul ,,Ora de matematică"
Proiectul ,,Lemniscata"
Proiectul ,,Iarnă, iarnă"
Proiectul ,,Grafic CosinusSinus"
Proiectul ,,Hai la joacă"

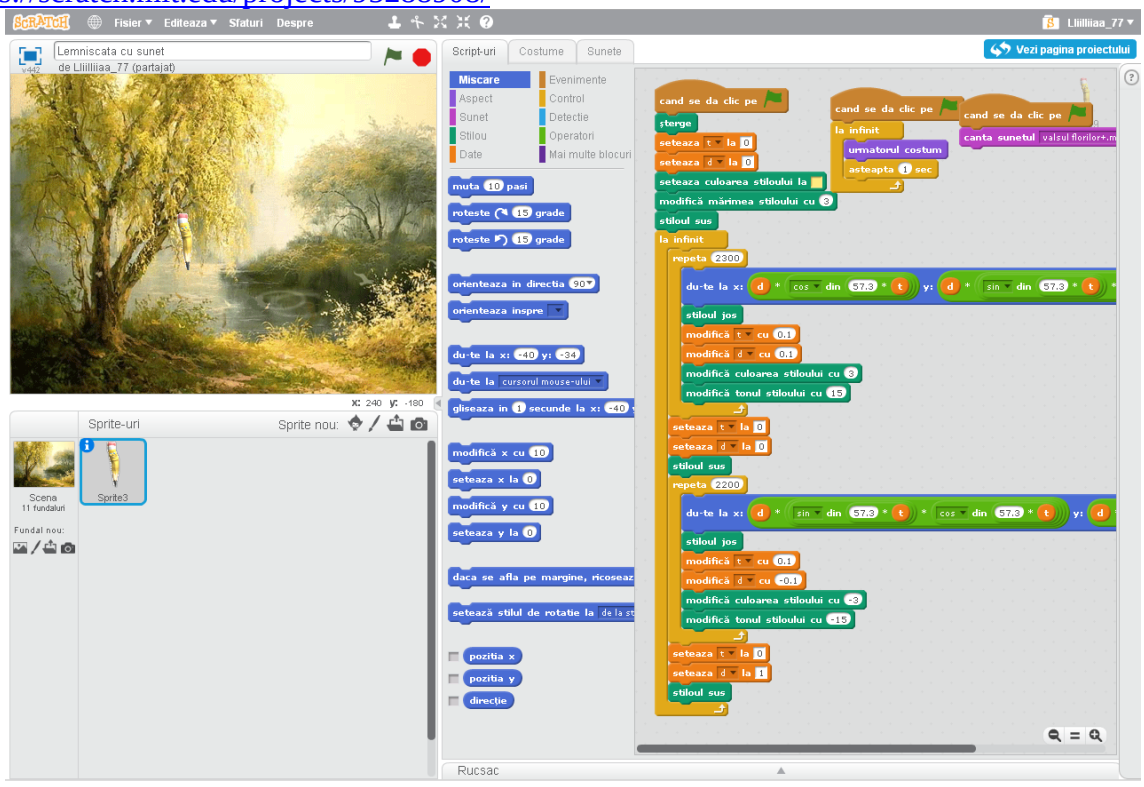
Proiectul „Ora de matematică”

https://scratch.mit.edu/projects/56081576/



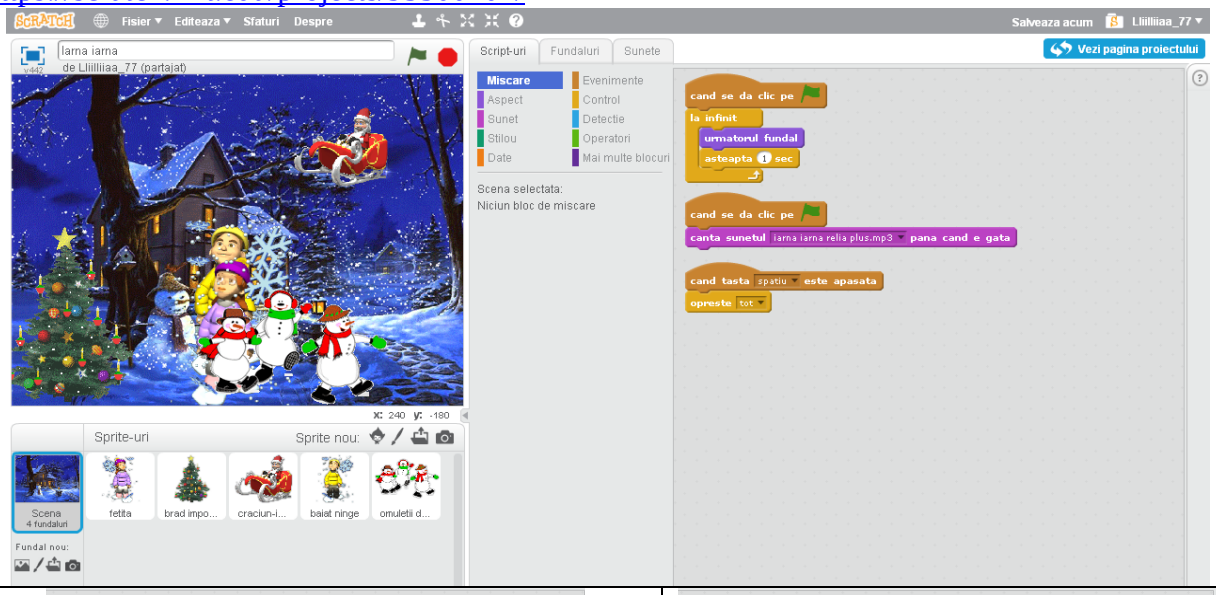
Proiectul „Lemniscata”

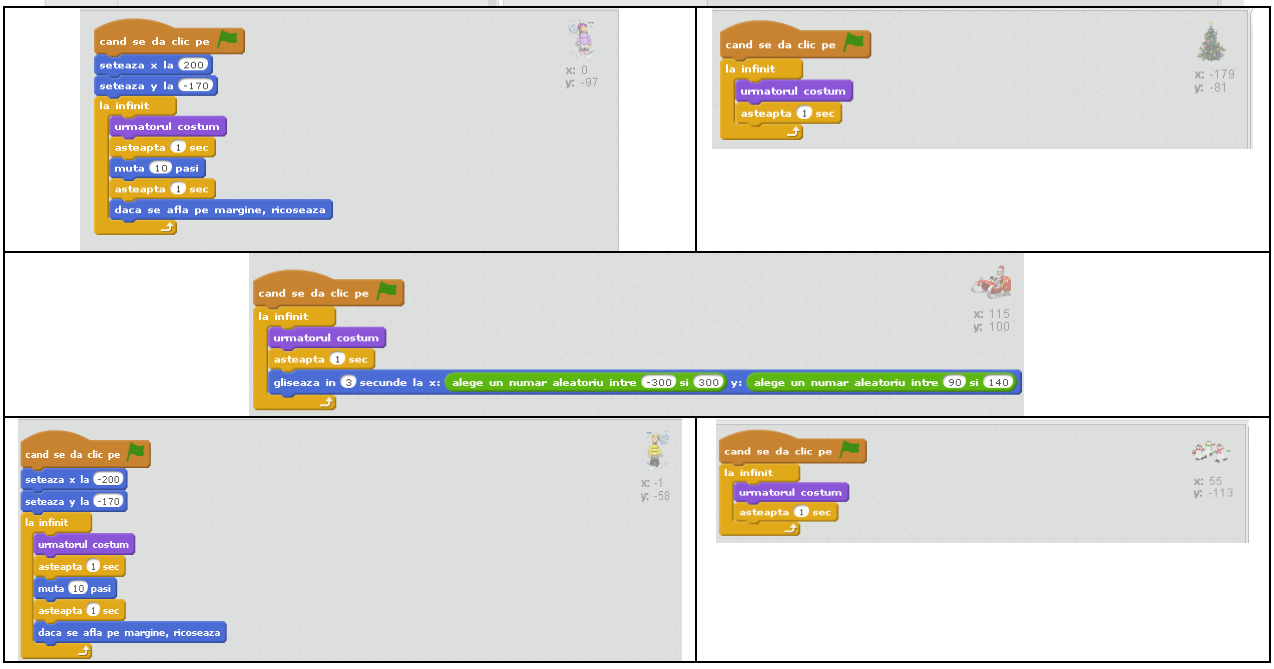
https://scratch.mit.edu/projects/95288908/



Proiectul „Iarnă, iarnă”

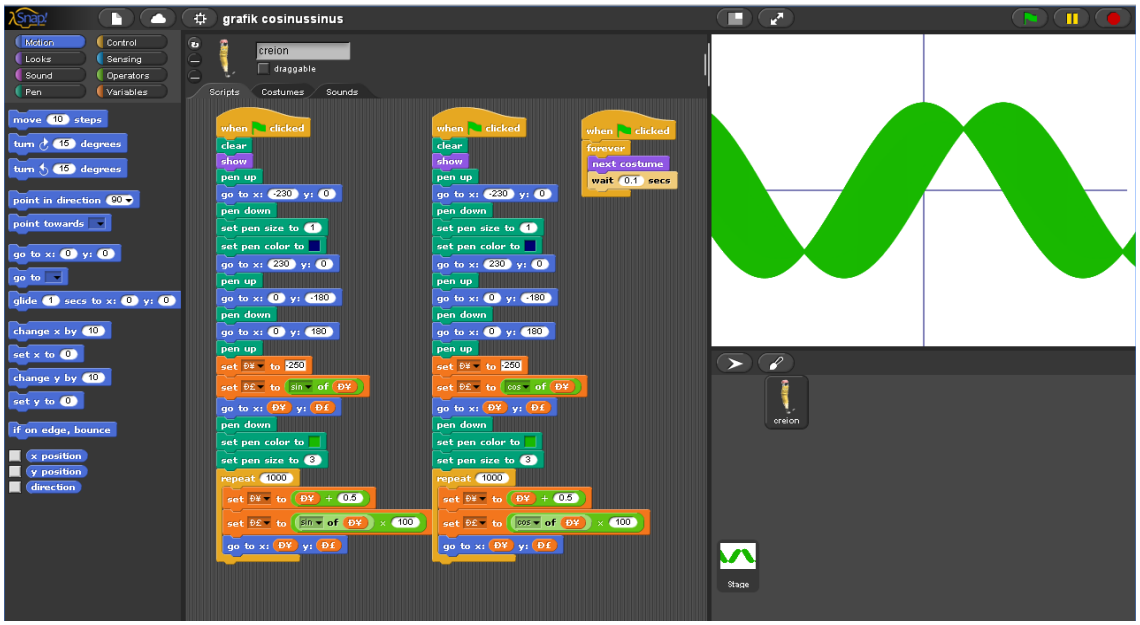
https://scratch.mit.edu/projects/95300404/





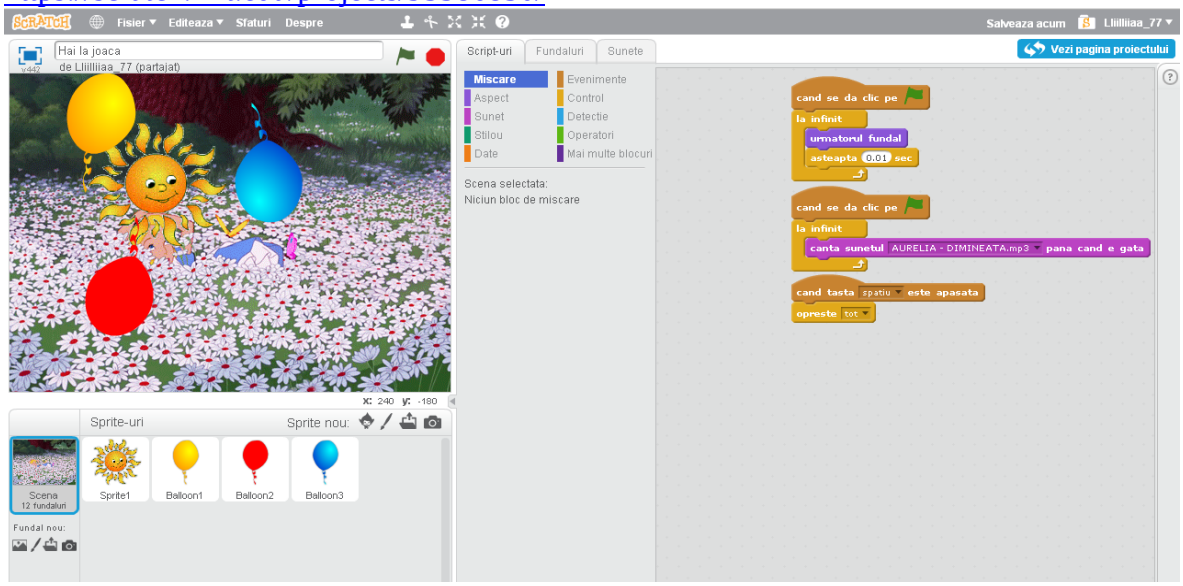
Proiectul „Grafic CosinusSinus”

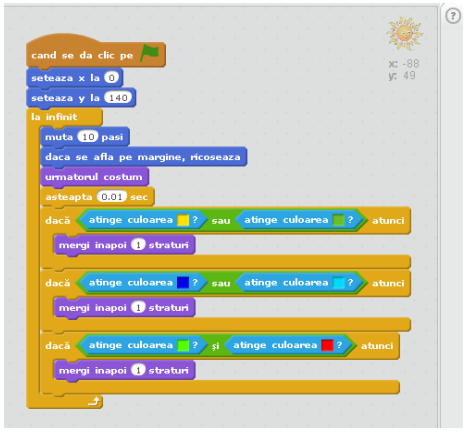
https://snap.berkeley.edu/snapsource/snap.html#present:Username=lliilliiaa77&ProjectName=grafik%20cosinussinus

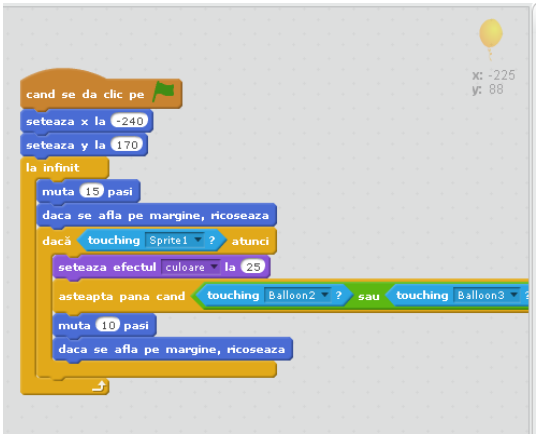


Proiectul „Hai la joacă”

https://scratch.mit.edu/projects/95586850/





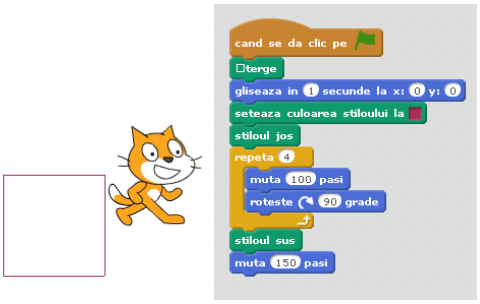


Am încercat să experimentăm împreună cu elevii claselor a 7-9 limbajul Scratch.

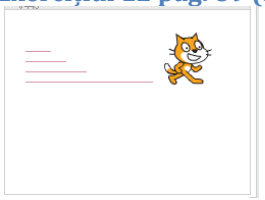
La clasa a 7 la cerința elevilor de a le arăta o aplicatie de a efectua desene în mişcare, altfel spus desene animate, după studierea aplicației paint le-am propus limbajul Scratch ceea ce elevilor le-a plăcut foarte mult. Unicul minus este anevoios de a desena cu mausul la calculator, dar pe tabla interactivă desenele prind viață foarte rapid...

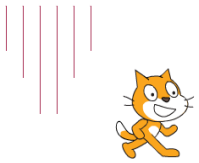
La clasa a 8 în paralel cu aplicația Cangurul le-am propus elevilor să modeleze aceleaşi programe în Scratch:

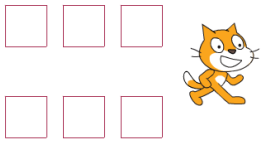
Desenarea unui pătrat

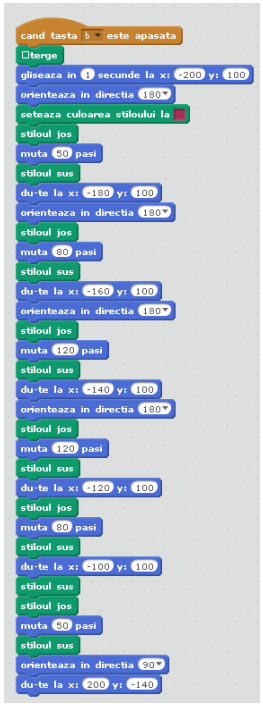


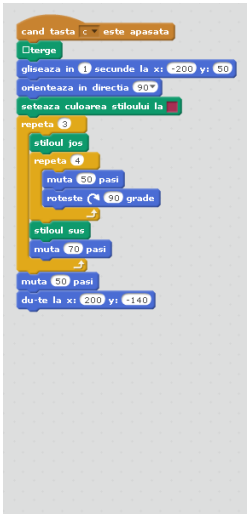
Exercițiul 12 pag. 89 (a,b,c,d)

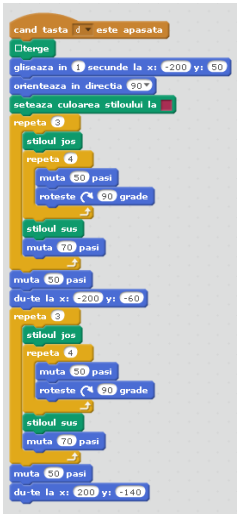




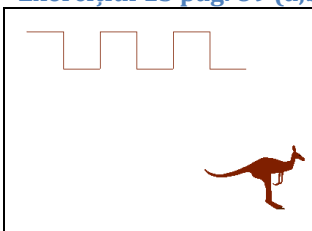




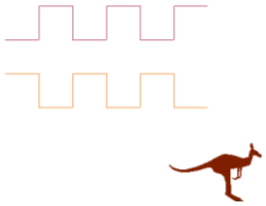




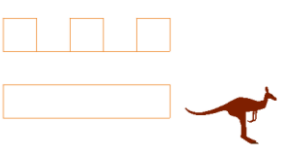
Exercițiul 13 pag. 89 (a,b,c,d)



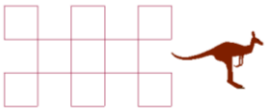
```
cand tasta a este apasata
terge
stiloul sus
gliseaza in 1 secunde la x: -200 y: 100
orienteaza in directia 90
seteaza culoarea stiloului la
stiloul jos
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
orienteaza in directia 0
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
orienteaza in directia 0
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
orienteaza in directia 90
stiloul sus
du-te la x: 150 y: -140
```



```
cand tasta b este apasata
terge
stiloul sus
gliseaza in 1 secunde la x: -200 y: 100
orienteaza in directia 90
seteaza culoarea stiloului la
stiloul jos
muta 50 pasi
roteste -90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
orienteaza in directia 180
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
roteste -90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
orienteaza in directia 180
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
roteste -90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
orienteaza in directia 90
stiloul sus
gliseaza in 1 secunde la x: -200 y: 50
orienteaza in directia 90
seteaza culoarea stiloului la
stiloul jos
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
orienteaza in directia 0
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
orienteaza in directia 0
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
orienteaza in directia 90
stiloul sus
du-te la x: 150 y: -140
```

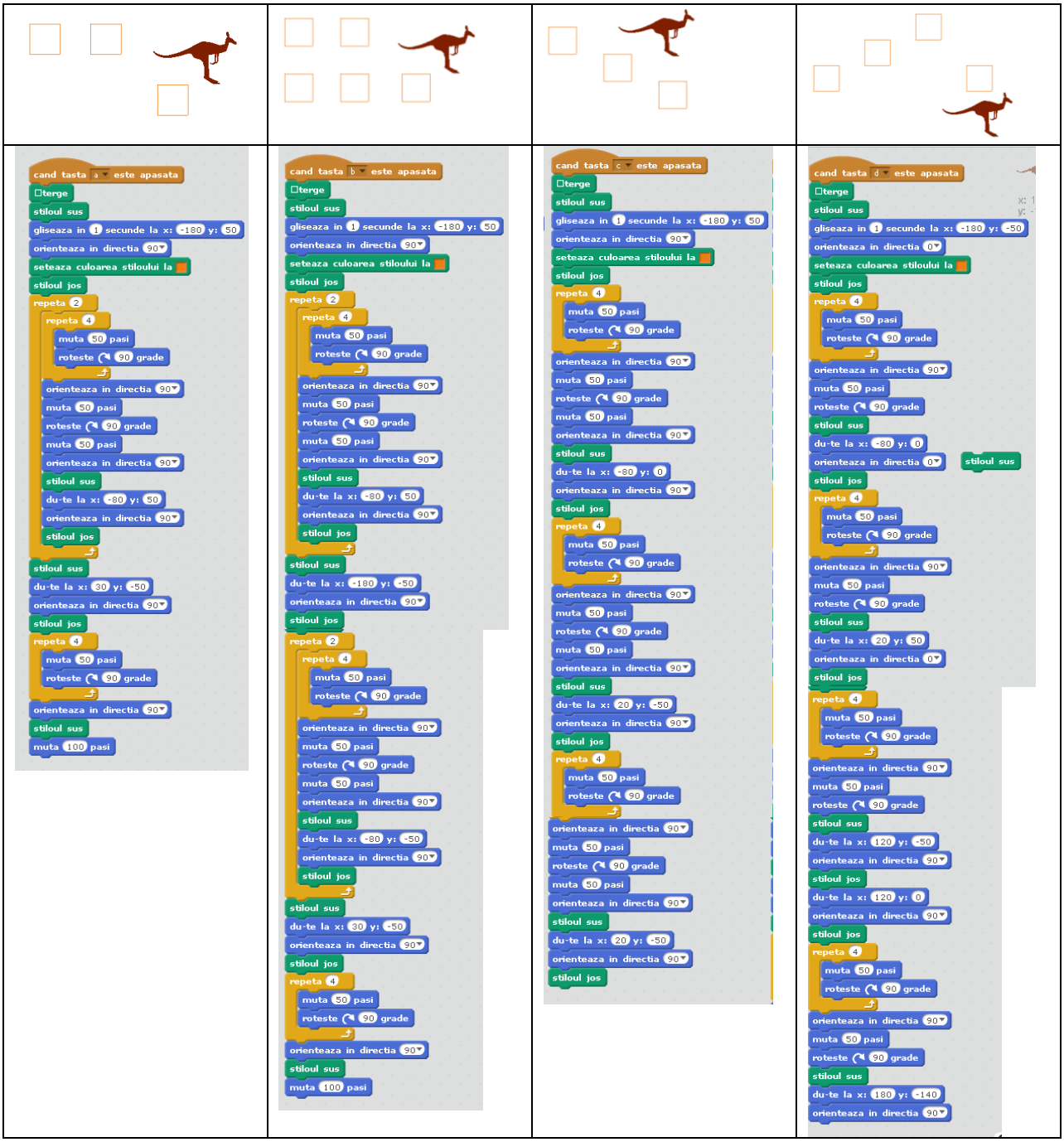


```
cand tasta c este apasata
terge
stiloul sus
gliseaza in 1 secunde la x: -180 y: 50
orienteaza in directia 90
seteaza culoarea stiloului la
stiloul jos
repeta 4
muta 50 pasi
roteste 90 grade
orienteaza in directia 90
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
stiloul sus
du-te la x: -80 y: 50
stiloul jos
repeta 4
muta 50 pasi
roteste 90 grade
orienteaza in directia 90
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
muta 50 pasi
stiloul sus
du-te la x: 20 y: 50
stiloul jos
repeta 4
muta 50 pasi
roteste 90 grade
orienteaza in directia 90
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
stiloul sus
du-te la x: -180 y: -50
stiloul jos
repeta 2
muta 250 pasi
roteste 90 grade
muta 50 pasi
roteste 90 grade
stiloul sus
du-te la x: 200 y: -140
```

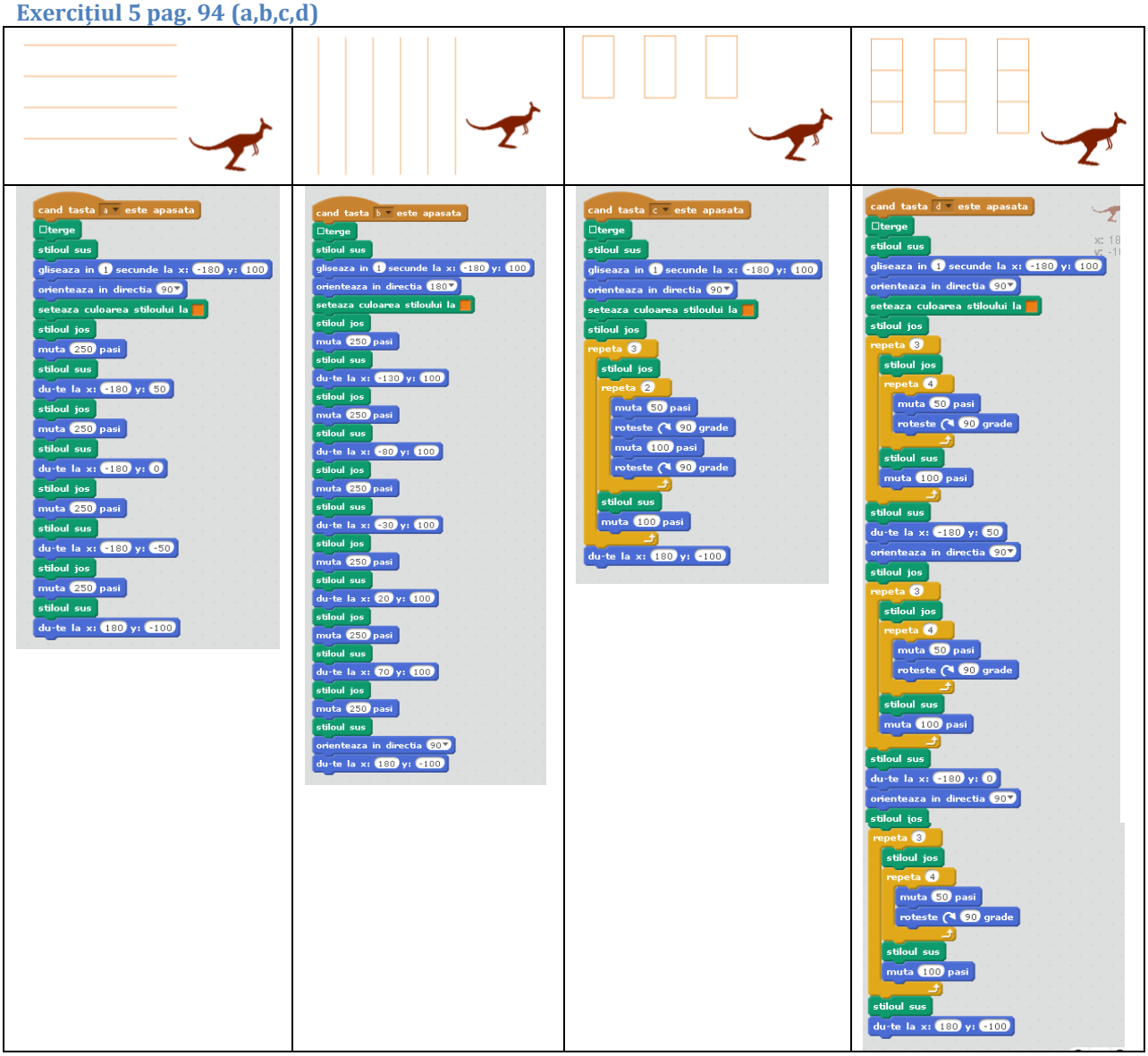


```
cand tasta d este apasata
terge
gliseaza in 1 secunde la x: -200 y: 50
orienteaza in directia 90
seteaza culoarea stiloului la
repeta 3
stiloul jos
repeta 4
muta 50 pasi
roteste 90 grade
stiloul sus
muta 100 pasi
muta 50 pasi
stiloul sus
du-te la x: -150 y: 0
orienteaza in directia 90
stiloul jos
repeta 4
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
repeta 4
muta 50 pasi
roteste 90 grade
muta 50 pasi
orienteaza in directia 90
repeta 4
muta 50 pasi
roteste 90 grade
stiloul sus
du-te la x: -200 y: -50
orienteaza in directia 90
repeta 3
stiloul jos
repeta 4
muta 50 pasi
roteste 90 grade
stiloul sus
muta 100 pasi
muta 50 pasi
du-te la x: 150 y: -100
```

Exercițiul 4 pag. 94 (a,b,c,d)

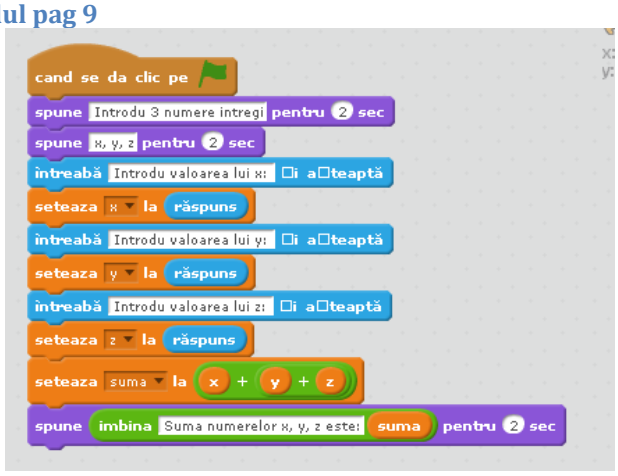


Exercițiul 5 pag. 94 (a,b,c,d)

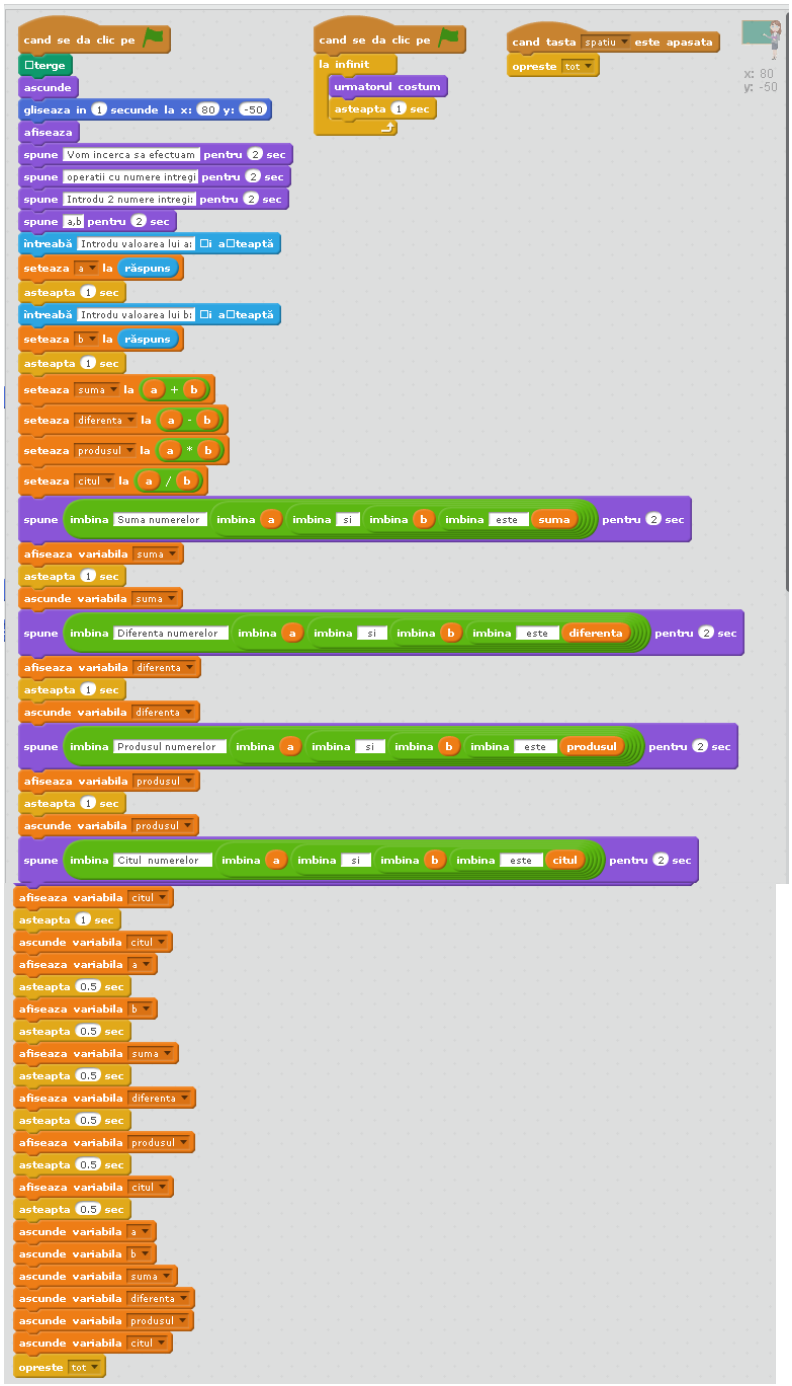


La clasa a 9 în paralel cu limbajul de programare Pascal le-am propus elevilor să modeleze aceleaşi programe în Scratch:

Exemplul pag 9

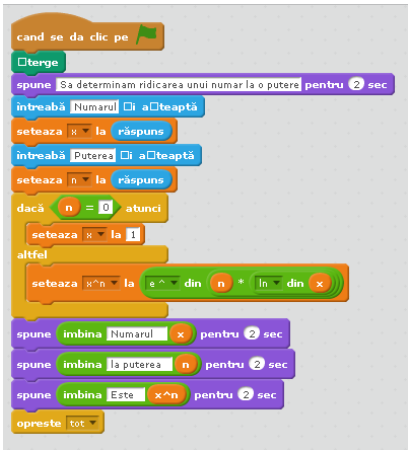


Programul P3

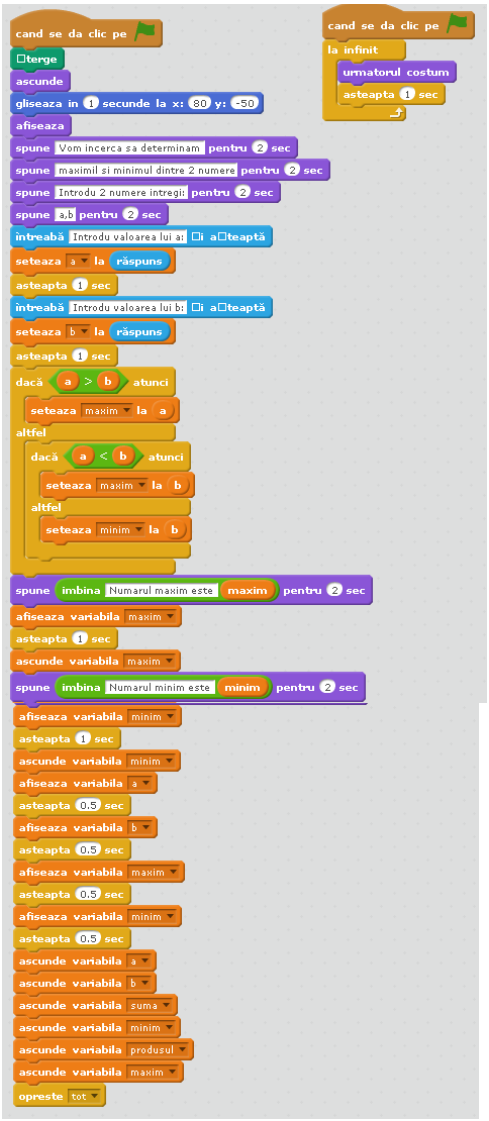


Programul P9

Programul P61



Programul P48



Programul P60