

INIȚIERE ÎN LIMBAJUL DE PROGRAMARE SCRATCH

Igor GUȚU, master în științe ale educației, profesor de informatică

IPLT ”Ion Creangă”, Popeștii de Sus, r. Drochia

Rezumat. În articol se menționează de importanța studierii informaticii prin intermediul aplicației SCRATCH de la vârsta de 10 – 12 ani. Acest program permite copiilor să dezvolte logică și gândirea computațională, creativitatea și colaborarea. Este o aplicație cu interfață prietenoasă pentru copii orientată pe creare, joc și colaborare. În partea a doua se face cunoștință cu interfața, meniurile, instrucțiunile de bază și modul de lucru. La fel se face referire la acțiunile minime necesare pentru a crea un proiect, un filmuleț cu animații.

Cuvinte cheie: *algorithm, limbaj de programare, scratch, program, proiect.*

Summary. *The article mentions the importance of studying computer science through the SCRATCH application from the age of 10-12. This program allows children to develop logic and computational thinking, creativity and collaboration. It's a kid-friendly app that focuses on creating, playing, and collaborating. The second part introduces you to the interface, menus, basic instructions and how to work. It also refers to the minimum actions required to create a project, an animated video.*

Keywords: *algorithm, programming language, scratch, program, project.*

Introducere

De ce le trebuie copiilor să studieze programarea?

„Toată lumea trebuie să învețe a programa, deoarece aceasta ne face să gândim”, menționa Steve Jobs . Atunci când ne referim la dezvoltarea logicii și abilităților analitice, se pare a fi aproape imposibil. Majoritatea persoanelor, unele mai mult, altele mai modest, sunt înzestrați cu „gândire matematică”. Indiferent de acestea, dezvoltarea gândirii analitice a copilului e posibilă. În ajutor vin limbaje speciale de programare „pentru copii”. Des auzim de la părinți fraze de felul: „Fiica mea va deveni avocat, de ce ea trebuie să cunoască programarea?”, „Fiul meu va fi judecător, e o pierdere de timp pentru el activitățile date.”

Motivele studierii programării

Există trei motive esențiale care ne îndeamnă să învățăm copiii programarea de la o vârstă cât mai timpurie.

1. Economisim timp și forțe pentru mai apoi, studiind de timpuriu

Realizarea programării are loc cu algoritmi. **Algoritmul** - un set de acțiuni care trebuie realizate pentru a obține un rezultat scontat. Orice proces, fie că este vorba de scrierea unui text, prepararea unor bucate, rezolvarea unei probleme, îl putem considera ca algoritm, în baza căruia putem elabora un program ce îl vom pune în aplicare. Nimeni nu întreabă pentru de ce avem nevoie de aceasta. E o necesitate din cotidian. Dacă lucrăm la calculator, îndeplinim diverse acțiuni: la culegerea și formatarea textului, sortarea dosarelor și fișierelor. Toate se pot realiza după un anumit algoritm.

Există soluții care economisesc timp și pot fi automatizate. Ca exemplu simplu este formatarea textului precum și unele programe speciale care sunt în Windows. Ele nu efectuează lucrul în locul nostru, dar o pot face la o oră stabilită, chiar dacă nu suntem la calculator, de exemplu: programul de gestionare a serviciilor poștale, activitatea antivirusului. Multe procese automate se pot aplica și pentru jocuri: se creează un „robot” și se stabilește o sarcină, pe care o realizează la anumit timp. Ce rol joacă programarea în toate acestea? Având în vedere că chiar și o cunoaștere superficială în acest domeniu ne permite să „gândim algoritmic” căutăm procese care pot fi automatizate și găsim opțiuni pentru modul în care acesta poate fi realizat, scriem un program pentru a rezolva anumite probleme. Deci, cu ce nu s-ar ocupa copilul, astfel de cunoștințe nu vor fi niciodată de prisos. Apropo, cu un algoritm simplu ne cunoaștem cu mult timp înainte de a începe să lucrăm la calculator. Chiar de la grădiniță copiii sunt învățați cu anumite reguli de joc, să îndeplinească anumiți pași (acțiuni), care pot fi considerați un algoritm. La fel sunt cunoscute și anumite semnale ale semaforului, de exemplu: „lumină roșie - nici o mișcare (stop)!” - un exemplu tipic de algoritm.

2. Viața este un lucru imprevizibil

Să ne imaginăm că nu știm să citim și să scriem, deoarece în copilărie nu am tins să devenim scriitor sau poet și de aceea nu am învățat alfabetul. Deci, să scriem o petiție sau să citim contractul bancar ar fi problematic. La scrierea unui mesaj vom apela la ajutorul unui prieten care știe să citească și să scrie. Sună ridicol, nu?! Posibil, că peste 10-20 de ani, așa se vor simți oamenii care nu cunosc elementele de bază ale programării. Reflectați: acum 50 de ani, existența unei astfel de discipline ca programarea, era cunoscută numai persoanelor deosebite. 25 de ani în urmă, puțini oameni știau cine sunt programatorii și cu ce se ocupau de fapt ei. Astăzi programarea este una dintre cele mai populare și căutate specialități. Dacă dezvoltarea și progresul vor merge în același ritm, în următorii 20 de ani limbaje de programare vor fi necesare pentru o cariera de succes, la fel ca și o limbă străină. În plus, tehnologiile informaționale, tot mai profund se infiltrează în viața de zi cu zi; dispozitivele care anterior au fost observate numai în filmele de fantastică, au apărut pe rafturile magazinelor obișnuite. Roboții de asistență, dispozitivele „caselor inteligente” și calculatoarele de la bordul mașinilor - toate acestea sunt realitatea de astăzi și pentru a le utiliza în mod independent, avem nevoie de noi cunoștințe.

3. Dezvoltarea copilului

Citind cele de mai sus, observăm: „Acest lucru este, desigur, foarte bun, dar se merită să încurajezi copilul să fie programator pentru perspectivele din viitor? Nu, mulțumesc.” Pentru mulți programarea se asociază cu un ecran negru, un set obscur de litere și numere, precum și erorile veșnice „Syntax Error”, întâlnite la lecțiile de informatică. Până nu de mult, pentru a ”păși” în programare a fost destul de dificil: cele mai multe programe sunt legate de limba engleză, sintaxă complexă, interfețe plictisitoare cu o mulțime de ferestre. Programarea a captat cu greu chiar și elevii din clasele

mari, nu mai vorbim de copiii din treapta gimnazială. Astăzi există limbaje de programare excepționale, care sunt în stare să stârnească curiozitatea chiar și a unui elev din ciclul primar, chiar și preșcolar.

Limbajul *Scratch* expus mai jos va face cunoscut copilul cu elementele de bază ale programării, îl va ajuta să-și dezvolte logica, orientarea spațială, atenția, creația și imaginația. Lucrând cu acest limbaj, copilul va experimenta și va primi plăcere de la proces, creând un produs unic, personal, cu care se va împărtăși părinților și prietenilor, va avea posibilitate să colaboreze cu semenii la perfecționarea proiectelor – atât de importantă în secolul XXI!

Introducere în SCRATCH

Scratch - mediu vizual de programare, orientat pe obiecte, destinat elevilor din clasele primare și gimnaziale. Numele ales vine de la cuvântul *scratching* – tehnică din echipamentul folosit de DJ hip-hop, care mișcă discuri de vinil înainte și înapoi cu mâinile, pentru a mixa teme muzicale. Scratch este o continuare a ideilor limbajului *Logo* și jocului *Lego*.

Ce este Scratch?

Există o diversitate de instrumente, ce pot fi utilizate în calculul creativ. Curriculumul la informatică din 2019, la clasa a IX-a, ne propune ca program vizual, aplicația Scratch.

Scratch - limbaj de programare pentru copii creat de MIT Media Lab.

Sloganul lor: *“Imagine. Program. Share”* ne trimite, din start, cu gândul la o comunitate, care ajuta copiii să se dezvolte nu doar la nivel tehnic, prin cunoștințele de informatică pe care le acumulează, ci și la nivel personal, punând accent pe colaborare și creativitate, creând starea de bine.

Scratch deține toate componentele importante ale unui limbaj de programare, ceea ce îl face mult mai relevant decât o simplă joacă pentru copii. Prin intermediul Scratch, copiii se regăsesc în postura de creatori și dezvoltatori, iar acest lucru îi învață să fie mai critici și să înțeleagă de ce e bine să nu rămână doar consumatori ai tehnologiei. Fiecare copil va descoperi cu plăcere programarea, fără să se blocheze în erori și sintaxă de la bun început.

Un limbaj de programare e până la urmă o prima ușă pe care tinerii trebuie s-o deschidă pentru a intra în lumea marilor proiecte tehnologice. Iar Scratch e unul dintre cele mai bune pentru că e simpatic și ușor de înțeles și utilizat, încurajează copiii să-și transforme fiecare idee în realitate!

Scratch este un set de instrumente programabile, ce permite copiilor:

- să-și creeze propriile jocuri, povestiri animate;
- să-și împărtășească creațiile prin intermediul Internetului.

Cu Scratch, oamenii pot crea o mare varietate de proiecte media interactive – animații, povești, jocuri și multe altele și pot partaja acele proiecte împreună cu alții într-o comunitate online. De la

lansarea lui Scratch în mai 2007, milioane de oameni din întreaga lume au creat și împărtășit peste 46 de milioane de proiecte.

Scopul final este de a ajuta copiii:

- să devină fluenți în mediile digitale;
- să se exprime creativ;
- să facă legături interdisciplinare.

Scratch1 a fost scris în *Squeak*, *Scratch2* este axat pe activitatea on-line și rescris în Flash / Activscript. Scratch este dezvoltat de o echipa mica de programatori pentru copii de la MIT. Versiunea **Scratch 2.0** a fost lansată la 9 mai 2013. Versiunea curentă – Scratch 3.0 este în baza la mediul de programe HTML5. Algoritmul de instalare a aplicației Scratch pe calculator:

1. Deschidem pagina: <https://scratch.mit.edu/download>;
2. Selectăm versiunea potrivită (V-2 pentru Windows XP sau V-3 pentru Windows-7, Windows-10);
3. Descărcăm fișierul de instalare;
4. Instalăm aplicația.

Lucrând online, riscăm să pierdem legătura cu Internetul. Proiectele se pot păstra, atât pe contul personal, cât și pe calculator. În modul offline ne putem ciocni că nu toate calculatoarele să fie compatibile cu ultima versiune.

Odată ce am deschis Scratch, ne apare acest ecran implicit din Fig.1, Interfața aplicației SCRATCH 3, cu următoarea structură: Bara de meniuri, bara de instrumente, scena de acțiuni, Biblioteca cu scene, Biblioteca cu personaje.

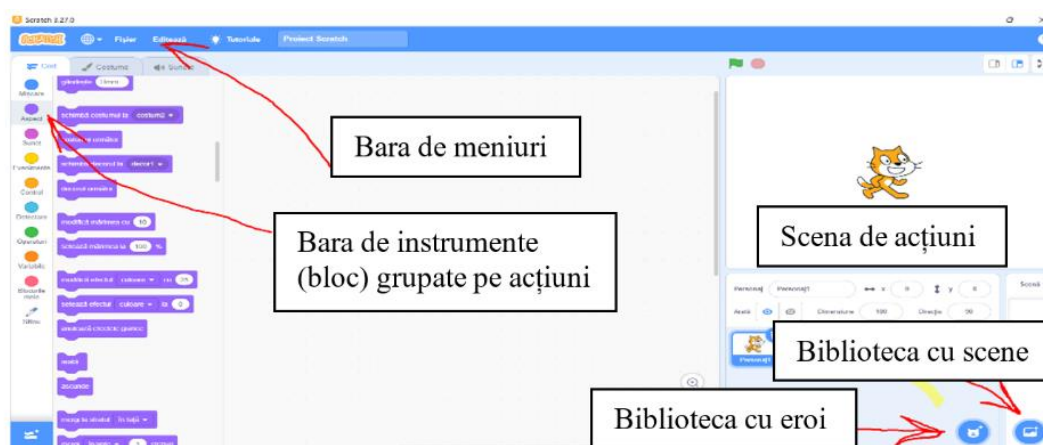


Fig. 1. Interfața aplicației SCRATCH 3

Programele în *Scratch*, constau din blocuri grafice, scripturile cărora depind de limba de interfață selectată. Putem selecta una dintre cele 50 de limbi ale interfeței, inclusiv și limba româna. Elementele grafice din scenă se numesc Sprite (eroi). Sprite-urile create pot și selectate și manipulate

prin adăugarea blocurilor în secțiunea "cod" din zona centrală (coloana a doua). La realizarea unui proiect este necesar să îndeplinim strict următoarele acțiuni:

1. Selectăm eroul proiectului (pot fi una sau mai multe imagini);
2. Selectăm scena (scenele) de acțiune;
3. Scriem programul pentru fiecare erou sau scenă;
4. Adăugăm efecte audio.

Realizarea unui program în *Scratch* presupune adăugarea unor blocuri care sunt organizate în 8 categorii de bază și două suplimentare:

- | | |
|--------------------------------------|--|
| 1. Blocul de acțiuni de mișcare; | 6. Instrucțiuni de detecție; |
| 2. Instrucțiuni de aspect; | 7. Operatori; |
| 3. Instrucțiuni de lucru cu sunetul; | 8. Variabile; |
| 4. Blocul de evenimente; | 9. Blocuri personale; |
| 5. Instrucțiuni de control; | 10. Blocuri suplimentare din colecție. |

Inițial, *Spritul* este o pisică portocalie. Putem alege o altă imagine din bibliotecă pentru proiectul nostru sau putem să o desenăm. Pentru a alege un sprite din biblioteca Scratch, facem clic pe pictograma "Alege personaj". Ne apare opțiunea din Fig.2. Selectarea personajului. Când plasăm mouse-ul peste buton, putem alege opțiunea "Faceți un sprite nou".

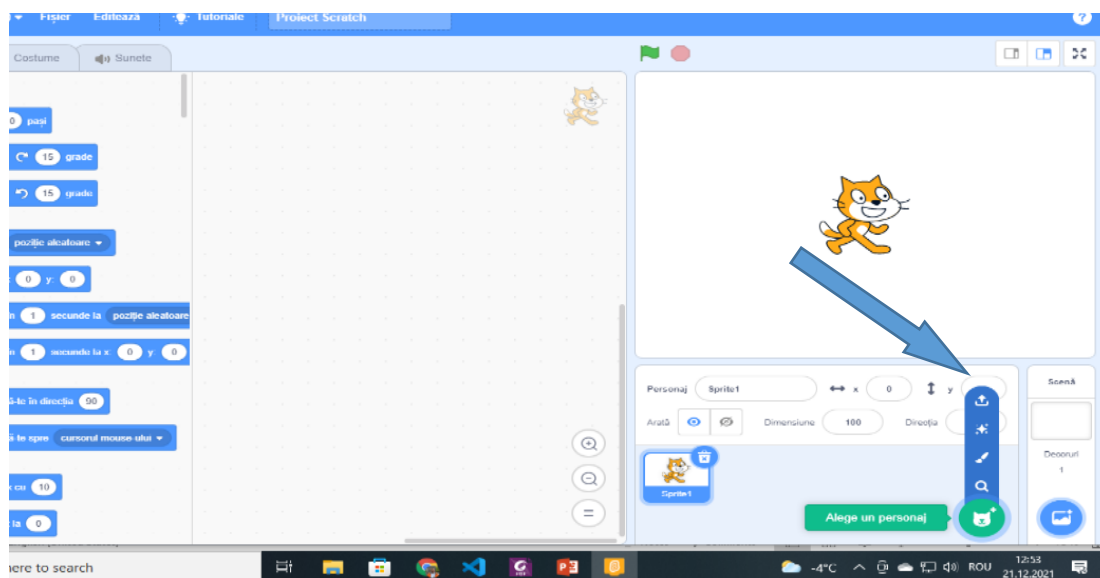


Fig. 2. Selectarea personajului

Primul lucru pe care trebuie să îl facem este să selectăm Spritul pe care dorim să-l programăm și să selectăm fila „Cod”. Doarece Sprite este singurul lucru care poate fi animat, trebuie să ne asigurăm că l-am selectat corect.

Zona *Script* este locul în care vă "construiți" programul, utilizând blocurile de programare. În colțul din stânga sus, al ferestrei Scratch, observăm 8 butoane.

În acele zone specifice, fiecare dintre aceste butoane are blocuri de programare. Când facem clic pe butonul de mișcare, observăm blocurile de programare care pot fi utilizate cu Sprite-ul nostru. De asemenea, putem seta blocul de mișcare. Pentru a deplasa personajul la dreapta - introducem un număr pozitiv de pixeli. Dacă introducem un număr negativ de pixeli în bloc, personajul se va mișca la stânga. În continuare, putem adăuga la celălalt bloc de cod, blocul „roțiți 15 grade”. Când apare o bară gri între cele 2 blocuri de cod, putem elibera mouse-ul. Această bară gri, înseamnă că cele două blocuri se vor „prinde” între ele. La fel putem schimba numărul de grade, făcând *clic* în acea zonă și tastând numărul de grade pe care ni-l dorim pentru rotirea Sprite-ului. Facem clic pe blocuri pentru a ne vedea mișcarea și rotirea Sprite-ului!

Următoarea categorie de butoane de programare pe care o vom experimenta este categoria Control. Aceasta este o categorie foarte importantă, deoarece controlează de câte ori are loc un eveniment, cât timp are loc un eveniment sau când se întâmplă un eveniment.

Alte blocuri puternice din categoria Control sunt blocurile care gestionează instrucțiunile condiționale. Declarațiile condiționale sunt numite în mod obișnuit „dacă – altfel” sau „if-else”.

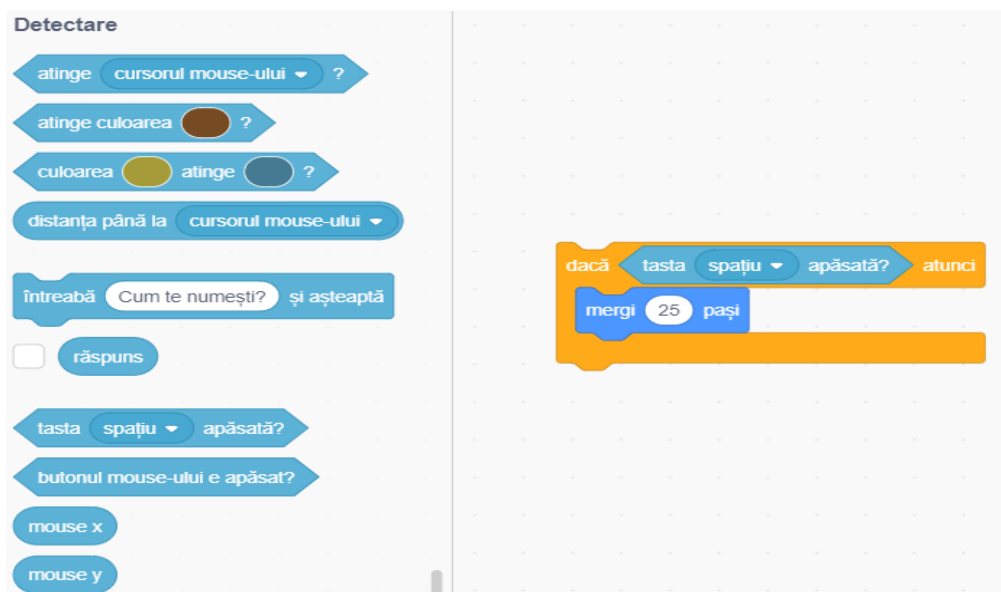


Fig. 3. Instrucțiuni condiționate

Declarațiile condiționale vor fi utilizate împreună cu blocurile de detectare. Observăm, în Fig.3 forma unor blocuri de detectare. Acestea sunt cele care se vor potrivi în interiorul „stării”.

Urmează să adăugăm un sunet animației noastre!

Există moduri diferite de a obține sunet în animație. Primul mod, cu care vom experimenta este *importarea unui sunet din biblioteca de sunet Scratch*.

Faceți clic pe sprite-ul pe care doriți să-l auzim și apoi pe fila ”Sunete”. Selectăm butonul ”încarcă”. În biblioteca Scratch sunt disponibile diferite categorii de sunete, care pot fi utilizate în animație.

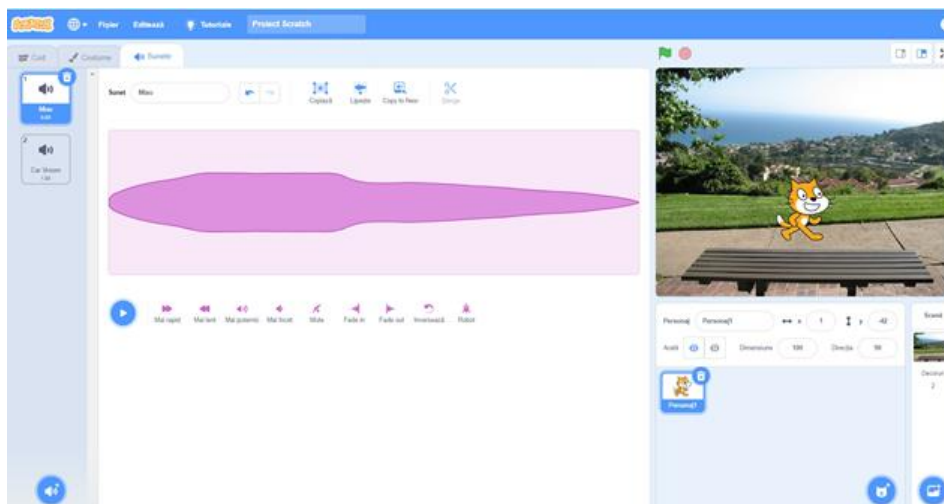


Fig. 4. Inserarea sunetului

Deci proiectul nostru conține personaje, scene, animații și sunete. Este un produs care poate fi optimizat, recurgând la colaborare (la dorință).

Concluzii

Optăm pentru mediul Scratch deoarece pune accent pe dezvoltarea gândirii computaționale - metodă modernă de educare. La fel valorifică abilitățile de rezolvare a problemelor; provoacă ca predarea și învățarea să devină cât mai creative; ajută la autoexprimarea elevului, contribuie la colaborarea între copii; evită erorile în calcul.

Bibliografie

1. BRENNAN, K.; BALCH, C.; CHUNG, M. Informatica creativă. Harvard graduate school of education. 2015, 140 p.
2. CARCHILAN, L. Programare în Scratch.
Online: <http://profesor.md/wp-content/uploads/2019/04/ Carte-Scrach.pdf>
3. Bazele Scratch. Despre Scratch. Online: <https://scratch.mit.edu/>