










Figura 1: Interfața cu utilizatorul a mediului Scratch








Pentru dezvoltarea competențelor de programare la elevii care vor utiliza Scratch ca instrument educațional cele mai importante elemente de lucru vor fi cele regăsite în paleta de blocuri. Este foarte importantă precizarea ordinii blocurilor și o foarte bună logică referitoare la modul lor de aplicare. Organizarea blocurilor conform funcțiilor pe care le îndeplinesc asigură facilitatea înțelegerii și modului de utilizare a acestora. Practic, aceasta este partea pe care elevii o pot utiliza pentru programare. Modalitatea curentă de înțelegere și implementare a conceptelor de programare pentru orele de Informatică din clasele a IX-a și a X-a ale ciclului liceal pentru învățământul teoretic, specializarea Științe ale Naturii, se rezumă la prezentarea unor probleme pur didactice, redactarea rezolvării acestora într-un limbaj de programare și implementarea acestora pe calculator, urmărind afișarea unui anumit rezultat, de regulă numeric sau alfanumeric. Prin utilizarea Scratch, elevii vor putea observa diverse acțiuni pe care le pot implementa asupra unui personaj, folosind blocurile de acțiune, mișcare, control, sunet. Astfel, modul

de lucru va deveni mult mai interactiv și va permite elevilor să observe grafic rezultatele programelor redactate de ei în Scratch.



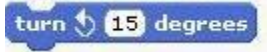


Organizarea paletei de blocuri

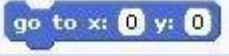


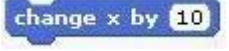

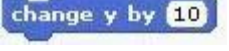
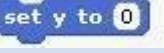
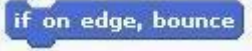



1. **Blocurile de control** sunt esențiale pentru realizarea unui proiect. Acestea sunt:

Denumire bloc	Ațiune	Reprezentare grafică
When Green Flag Clicked	Descrie ce se realizează în momentul începerii rulării programului.	
When () Key Pressed	Selectează tasta care este apăsată, fiind urmat de o acțiune ce va fi realizată în momentul apăsării tastei alese.	
When () Clicked	Precede acțiunea ce va fi realizată la efectuarea unui click pe un anumit obiect (mouse sau personaj)	
When I Receive ()	Descrie evenimentul care este recepționat și lanșează o acțiune ulterioară.	
Forever	Precede acțiunea care va fi realizată permanent pe parcursul scriptului.	
Repeat ()	Asigură repetarea ciclică pentru un anumit număr de ori a scriptului din cadrul său.	
Forever if ()	Permite realizarea permanentă a acțiunilor din cadrul blocului în cazul în care o condiție booleană este îndeplinită.	
If ()	Bloc de control care asigură realizarea acțiunii din cadrul său doar în cazul în care este adevărată condiția dintre paranteze.	
If (), Else	Permite realizarea unei anumite acțiuni în cazul unei condiții adevărate, respectiv a alteia în cazul unei condiții false. Testează condiția.	

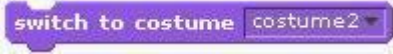
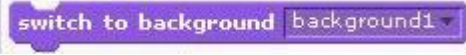


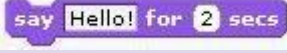


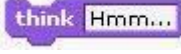

Repeat Until ()	Repetă o anumită acțiune din cadrul blocului până când o condiție devine adevărată.	
Wait () Secs	Oprește execuția scriptului pentru un număr dat de secunde.	
Broadcast ()	Asigură transmiterea unui eveniment pe toată durata programului Scratch.	
Broadcast () and Wait	Asigură transmiterea unui eveniment de tip Receive și așteaptă activarea tuturor scripturilor în care este inclus evenimentul transmis.	
Wait Until ()	Înterupe execuția scriptului în care este inclus, până la îndeplinirea unei anumite condiții logice.	
Stop Script	Dezactivează scriptul în care este inclus, oprind complet execuția acestuia.	
Stop All	Dezactivează toate scripțiile din program, oprindu-l complet.	



2. *Blocurile de mișcare* sunt:

Denumire bloc	Acțiune	Reprezentare grafică
Move () Steps	Asigură mișcarea personajului înainte cu un anumit număr de pași.	
Turn () Degrees	Rotește persoajul cu numărul de grade introdus de scenarist. (în sensul acelor de ceasornic, respectiv în sens contrar, conform selecției)	 
Point in Direction ()	Asigură întoarcerea personajului în direcția precizată.	
Point Towards ()	Întoarce personajul, schimbându-i direcția fie către zona în care se află cursorul mouse-ului, fie către alt personaj.	

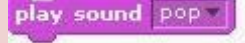
Go to X: () Y: ()	Deplasează personajul la coordonatele X și Y precizate.	
Go to ()	Mută poziția personajului la cea a cursorului mouse-ului sau a altui personaj.	
Glide () Secs to X: () Y: ()	Personajul culisează(alunecă) către punctul de coordonate X și Y în numărul dat de secunde.	
Change X by ()	Modifică poziția personajului, schimbând valoarea coordonatei X (pe orizontală), cu o anumită valoare specificată.	
Set X to ()	Precizează noua valoare a coordonatei X a personajului. De obicei se folosește pentru a stabili punctul de intrare în scenă al personajului.	
Change Y by ()	Similar cu <i>Change X by ()</i> , modificând coordonata Y, cea verticală.	
Set Y to ()	Similar cu <i>Set X to ()</i> , precizând noua valoare a coordonatei Y.	
If on Edge, Bounce	Blocul verifică dacă personajul atinge marginile suprafeței de lucru și, dacă acest lucru se întâmplă, personajul va pointa(porni) în direcția care se află în oglindă față de direcția din care a venit. Folosește o linie perpendiculară pe linia de margine pentru a determina unghiul de reflecție.	
X Position	Blocul stochează coordonata X a personajului.	
Y Position	Blocul stochează coordonata Y a personajului.	
Direction	Blocul stochează direcția personajului, măsurată în grade. Va fi folosită în cadrul blocului <i>Point in Direction()</i> .	

3. *Blocurile de aspect* modifică modul în care arată/înfățișarea unui personaj.




Denumire bloc	Acțiune	Reprezentare grafică
Switch to Costume ()	Schimbă costumul personajului cu noul costum precizat.	
Switch to Background ()	Modifică fundalul, stabilindu-l la cel specificat.	
Next Costume	Modifică aspectul personajului, trecând la următorul costum, sau în cazul în care cel actual este ultimul, la primul costum din lista de costume.	
Next Background	Similar cu <i>Next Costume</i> , modificând în schimb fundalul.	
Say () for () Secs	Alocă personajului un balon de dialog ce va conține un anumit mesaj. Balonul va apărea pe ecran pentru un număr dat de secunde.	
Say ()	Duce la apariția unui balon de dialog, care va dispărea în momentul alocării unui alt mesaj sau al întâlnirii unui bloc <i>Think</i> , sau la <i>Stop</i> .	
Think () for () Secs	Similar cu <i>Say () for () Secs</i> , cu deosebirea aspectului balonului de dialog, care devine acum balon de gândire.	
Think ()	Similar cu <i>Say ()</i> , cu deosebirea aspectului balonului de dialog, care devine acum balon de gândire.	
Change () Effect by ()	Scriptul modifică un anumit efect al personajului cu o anumită valoare specificată	

Set () Effect to ()	Stabilește valoarea unui anumit efect la cea precizată. Efectele care pot fi modificate sunt culorile, fisheye, rotire, pixelare, mozaic, luminozitate și umbră.	
Clear Graphic Effects	Resetează valorile efectelor la cele inițiale.	
Change Size by ()	Modifică dimensiunea personajului cu un anumit procentaj.	
Set Size to ()%	Stabilește dimensiunea personajului, modificând valoarea implicită de 100% cu un anumit procent.	
Show	Dacă personajul este ascuns, acesta va apărea în scenă, dacă nu, nu se va modifica nimic.	
Hide	Ascunde personajul, acesta nu mai este vizibil în scenă.	
Go to Front	Personajul va fi plasat în fața altor personaje din scenă.	
Go Back () Layers	Trimite personajul în spatele nivelului scenei cu un anumit număr de straturi.	
Costume #	Blocul reține numărul curent al costumului personajului.	
Background #	Blocul reține numărul curent al aspectului fundalului.	
Size	Blocul reține dimensiunea actuală a personajului.	








4. *Blocurile de sunet* sunt:





Denumire bloc	Acțiune	Reprezentare grafică
Play Sound ()	Blocul va reda un sunet ales din lista	

	de sunete disponibile, fără a influența execuția scriptului.	
Play Sound () Until Done	Blocul redă un sunet specificat, oprind execuția scriptului până la finalizarea redării acestuia.	
Stop All Sounds	Blocul va opri toate sunetele redade pentru toate personajele și pentru scena curentă.	
Play Drum () for () Beats	Redă sunetul specificat pentru un număr dat de secunde. Deși în cadrul codului este vorba despre tobe, se poate alege din mai multe instrumente, cum ar fi triunghi, bongo, clopot, idiofoane.	
Rest for () Beats	Oprește execuția scriptului pentru un anumit număr de bătăi muzicale. Numărul de bătăi poate fi un număr zecimal.	
Play Note () for () Beats	Redă o anumită notă selectată pentru un anumit număr dat de bătăi muzicale.	
Set Instrument to ()	Blocul modifică setul de instrumente MIDI care este utilizat de blocul <i>Play Note () for () Beats</i> . Blocul <i>Play Drum () for () Beats</i> nu este afectat, având prestabilite instrumentele utilizate.	
Change Volume by ()	Modifică volumul sunetului personajului sau al scenei cu un anumit procent.	
Set Volume to () %	Stabilește volumul sunetului la un anumit procent din totalul disponibil.	
Change Tempo by ()	Blocul modifică tempoul întregului proiect cu un anumit procent. Modificarea tempoului are efect și	

	asupra timpului pentru care vor fi redade sunetele din blocurile <i>Play Drum () for () Beats</i> și <i>Play Note () for () Beats</i> , deoarece tempoul afectează lungimea unei bătăi muzicale.	
Set Tempo to () bpm	Blocul stabilește tempoul sau viteza de lucru a întregului proiect, folosind ca unitate de măsură numărul de bătăi pe minut.	
Volume	Reține nivelul volumului pentru întregul proiect.	
Tempo	Reține tempoul pentru întregul proiect.	


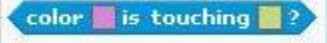









5. *Blocurile de tip Pen* sau urmă sunt:




Denumire bloc	Acțiune	Reprezentare grafică
Clear	Resetează întreaga scenă. Este similar cu resetarea unui ecran din cadrul limbajelor de programare.	
Pen down	Desenează o urmă, pentru a putea observa drumul parcurs de un anumit personaj, respectiv mișcările făcute pe suprafața scenei.	
Pen up	Oprește efectul blocului <i>Pen down</i> .	
Set Pen Color to () (color)	Stabilește culoarea urmei desenate, prin alegerea vizuală a culorii.	
Change Pen Color by ()	Modifică culoarea urmei trasate.	
Set Pen Color to ()	Stabilește culoarea urmei desenate, prin valoarea numerică a acesteia.	
Change Pen Shade by ()	Modifică nuanța culorii urmei, cu o valoare între 0 și 200.	

Set Pen Shade to ()	Stabilește nuanța urmei trasate.	
Change Pen Size by ()	Modifică dimensiunea urmei trasate cu un anumit procent din totalul implicit.	
Set Pen Size to ()	Stabilește grosimea urmei trasate.	
Stamp	În locul unei urme de creion, se copiază, se imprimă pe spațiul scenei imaginea personajului, repetată în funcție de locurile către care se delasează. Dacă două personaje se întâlnesc în același punct, opțiunea <i>Stamp</i> nu va permite suprapunerea imaginilor acestora.	







6. *Blocurile senzor* detectează atingerea anumitor obiecte din scenă:





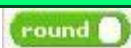
Denumire bloc	Acțiune	Reprezentare grafică
Ask () and Wait	Blocul va crea o linie de dialog, care va fi afișată în partea inferioară a ecranului. Utilizatorii pot introduce text în cadrul acesteia și pot valida textul. Intrarea este stocată în blocul <i>Answer</i> , care se modifică, luând valoarea ultimului text introdus.	
Reset Timer	Blocul stabilește valoarea timerului la 0.0. Acest bloc este utilizat împreună cu blocul Timer. De obicei, timerul trebuie reinițializat la începutul fiecărui proiect pentru a reține valoarea corectă pe parcursul proiectului.	
Touching ()?	Bloc de tip logic. Verifică dacă personajul atinge marginea scenei, un alt personaj sau cursorul, în acest caz returnând valoarea <i>adevărat</i> . Dacă personajul nu atinge aceste	

	obiecte, returnează valoarea <i>fals</i> .	
Touching Color ()?	Similar cu blocul anterior, cu diferența că verifică atingerea unei anumite culori specificate.	
Color () is Touching ()?	Verifică dacă o anumită culoare din cadrul costumului personajului atinge o altă culoare din scenă sau aparținând costumului altui personaj. Returnează o valoare logică.	
Mouse Down?	Verifică dacă este apăsat unul dintre butoanele mouseului. Returnează o valoare logică.	
Key () Pressed?	Verifică dacă o anumită tastă stabilită este apăsată. Returnează <i>adevărat</i> sau <i>fals</i> .	
Loud?	Blocul verifică valoarea zgomotului introdus prin intermediul microfonului. Dacă aceasta depășește valoarea 30, returnează <i>adevărat</i> ; altfel, returnează <i>fals</i> .	
Sensor ()?	Blocul verifică dacă butonul de pe o placă Pico este apăsat sau dacă un anumit senzor este conectat la portul dedicat. Returnează o valoare logică.	
Answer	Reține cel mai recent text introdus în cutia de dialog.	
Mouse X	Reține coordonata X a punctului în care se află cursorul mouseului.	
Mouse Y	Reține coordonata Y a punctului în care se află cursorul mouseului.	
Distance to ()	Returnează distanța între centrul personajului și centrul altui personaj sau cursorul mouseului.	
Timer	Reține valoarea timerului și este	



	utilizat împreună cu blocul Reset timer .	
() of ()	Va returna un anumit parametru pentru personajul stabilit. Poate returna coordonatele X și Y, dimensiunea, direcția, costumul, volumul sunetului și variabilele locale ale scenei sau personajului selectat.	
Loudness	Reține valoarea sunetului receptat de microfon, pe o scară între 0 și 100.	
() Sensor Value	Returnează valorile anumitor senzori de pe plăcile de tip Pico sau din kitul Lego Education Robotics.	




7. **Blocurile operator** descriu diverse operații aritmetice sau logice asupra unor valori și sunt:

Denumire bloc	Acțiune	Reprezentare grafică
() < ()	Verifică dacă prima valoare este mai mică decât cea de-a doua. Returnează o valoare logică.	
() = ()	Verifică dacă prima valoare este egală cu cea de-a doua. Returnează o valoare logică.	
() > ()	Verifică dacă prima valoare este mai mare decât cea de-a doua. Returnează o valoare logică.	
() and ()	Returnează rezultatul operației logice ȘI.	
() or ()	Returnează rezultatul operației logice SAU.	
Not ()	Neagă valoarea logică a unei variabile.	

() + ()	Returnează valoarea sumei a două variabile.	
() - ()	Returnează valoarea diferenței a două variabile.	
() * ()	Returnează valoarea produsului a două variabile.	
() / ()	Returnează valoarea împărțirii a două variabile.	
Pick Random () to ()	Alege o valoare aleatoare care se găsește în intervalul cuprins între prima variabilă și cea de-a doua.	
Join ()()	Concatenează valorile a două variabile de orice tip și returnează rezultatul.	
Letter () of ()	Blocul returnează caracterul cu numărul primei variabile din șirul de caractere specificat în cea de-a doua variabilă.	
Length of ()	Returnează lungimea unui șir de caractere.	
() Mod ()	Returnează valoarea restului împărțirii întregi a două variabile.	
Round ()	Rotunjește numărul introdus la cel mai apropiat întreg.	
() of ()	Blocul realizează o anumită operație descrisă printr-o funcție asupra unui număr introdus de la tastatură și returnează rezultatul.	

8. *Blocurile de tip variabilă* sunt:

Denumire bloc	Acțiune	Reprezentare grafică
Set () to ()	Stabilește valoarea unei variabile la un număr introdus de la tastatură.	
Change () by ()	Modifică valoarea unei variabile selectate cu un număr introdus de la tastatură.	

Show Variable ()	Monitorizează valoarea variabilei selectate.	
Hide Variable ()	Ascunde afișajul de monitorizare a valorii unei variabile.	
()	Reține valoarea unei variabile stabilite de utilizator.	

Un exemplu de program simplu realizat în Scratch este următorul:

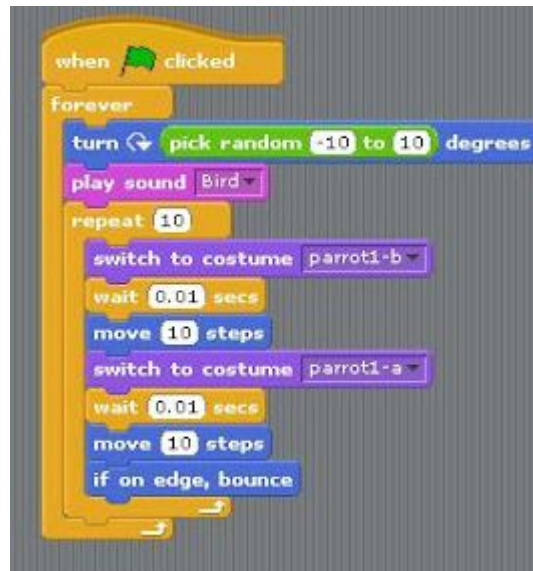


Figura 3: Exemplu de aplicație în Scratch

Proiectul realizează mișcarea unui personaj creat de utilizator cu ajutorul părții de grafică simplificată a mediului Scratch. Este vorba de un papagal. Explicitând codul, acesta realizează următoarea aplicație: la începutul programului, marcat prin click pe steagul verde, se execută în buclă continuă, deci până la apăsarea butonului de oprire, următoarele operații: papagalul ales ca personaj își modifică direcția de deplasare, rotindu-se către dreapta cu un număr de grade între -10 și 10, ales aleatoriu de către calculator; se generează sunetul Bird, care este redat și se repetă de 10 ori secvența de instrucțiuni de schimbare a costumului b cu costumul a la un interval de 0.01 secunde, la fiecare modificare de costum, papagalul deplasându-se 10 pași pe direcția dată; dacă papagalul se lovește de pereții scenei sau de un alt obstacol, își va modifica direcția de deplasare în oglindă față de cea inițială.

După cum se poate observa, printr-o simplă organizare a blocurilor se poate obține o aplicație. În funcție de natura scenariului proiectat, aceste aplicații pot deveni dificile, conținând mai multe personaje, scene sau chiar nivele de lucru. Se poate lucra cu diverse tokenuri, cum ar fi scorul, numărul de vieți, un timer, se pot introduce diferite interacțiuni, cum ar fi penalizările pentru un răspuns la o întrebare greșită la un concurs de tip trivia educațional, cu întrebări din diferite domenii selectate.

Nefiind un mediu complex de programare și bazându-se pe simpla înțelegere a funcționalității blocurilor puse la dispoziția utilizatorului, singurele impedimente pentru utilizarea lor de către profesorii de diverse specialități pot consta în cunoașterea limbii engleze și în înțelegerea unor noțiuni elementare referitoare la conceptele și structurile de control ale unui limbaj de programare. Scratch oferă, pentru cei interesați, posibilitatea reutilizării unor aplicații deja realizate de către alți utilizatori și a modificării și adaptării acestora conform cerințelor fiecăruia.

Scratch vine în întâmpinarea cerințelor europene de dezvoltare a diverselor competențe, axându-se în special pe cele lingvistice și digitale, dar având în vedere și partea matematică, logică și aplicativă a oricărei programe de învățare a unei discipline predate.

Consider că oportunitatea participării la cursul pSkills, oferită prin intermediul grantului Comenius, mi-a permis accesul la informații referitoare la un mediu de programare extrem de util, care poate fi utilizat nu numai în timpul orelor de informatică, ci și în cadrul unor proiecte transdisciplinare. Baza de aplicații software a liceului la care predau poate fi astfel îmbogățită cu aplicații realizate de profesori de mai multe specialități și, de ce nu, chiar de elevi pasionați.

În concluzie, voi utiliza mediul Scratch pentru explicarea mult mai interactivă și creativă a conceptelor de bază din programare și ca bază pentru o viitoare disciplină opțională din cadrul liceului, datorită posibilităților pe care le oferă, nefiind axat pe rezolvarea teoretizată a unor probleme, ci pe o parte mult mai interesantă, cea aplicativă prin intermediul creării, dezvoltării și utilizării jocurilor de tip educațional.