

# INTRO TO GAME DESIGN

Programarea pe calculator și designul de jocuri cu Scratch



## Ce este programarea pe calculator?

Programarea pe calculator este procesul de scriere a instrucțiunilor care urmează să fie executate de un calculator. Calculatorul tău și toate site-urile web pe care le vizitezi, telefonul tău și toate aplicațiile pe care le folosești, și orice jocuri video pe care le joci sunt toate posibile datorită programării pe calculator. Programarea pe calculator este, de asemenea, forța motrice din spatele multor progrese tehnologice actuale ale lumii, inclusiv inteligența artificială, vehiculele autonome și realitatea virtuală.

Exact ca un designer de jocuri, programatorii de calculator trec constant printr-un ciclu de prototipare, testare și ajustare a programelor lor. Acest proces este adesea numit procesul de design ingineresc. Acest proces este folosit în inginerie, știință și în domeniile legate de design și există în mai multe versiuni diferite. Iată o versiune de la NASA, precum și ciclul de design al jocurilor pe care l-ai folosit anterior:



## **Ce este Scratch?**

Există multe limbaje de programare diferite. Unele dintre cele mai populare limbaje de programare includ Python, JavaScript și C++. Vom folosi un limbaj numit Scratch pentru a învăța elementele de bază ale programării și designului de jocuri.

Scratch este un set de unelte care combină programarea pe calculator cu instrumente de design grafic, permițând utilizatorilor să creeze jocuri, animații și alte programe interactive.

Accesează Scratch la **scratch.mit.edu**. Selectează "**Create**" în colțul din stânga sus pentru a accesa editorul de proiecte.





## Editorul de proiecte Scratch

Editorul de proiecte Scratch este locul în care vor avea loc toate activitățile de codare, proiectare și creare.



**Stage** - Aici vei vedea rezultatul proiectului tău pe măsură ce îl construiești.

**Sprite Menu** - Sprite-urile sunt toate personajele și obiectele din proiectele noastre. Aici poți controla ce sprite-uri sunt în proiectul tău, precum și numele, poziția, vizibilitatea și dimensiunea acestor sprite-uri. Adaugă mai multe sprite-uri făcând clic pe cercul albastru cu fața de pisică din colțul din dreapta jos al meniului.

**Backdrop Menu** - Fundalurile sunt decorurile pentru proiectele noastre. Adaugă și gestionează fundalurile aici. Adaugă un nou fundal făcând clic pe cercul albastru cu munții din partea de jos a meniului. **Tool Select Tabs** - Alege între instrumentele Cod (Code), Costume (Costumes) și Sunete (Sounds). Schimbarea între aceste file va modifica instrumentele disponibile în secțiunile de mai jos.

**Block Palette** - Aici poți vedea toate blocurile de cod disponibile. Blocurile de cod sunt împărțite în categorii codificate prin culori, cum ar fi Mișcare (Motion), Evenimente (Events), Sunet (Sound) etc.

**Coding workspace** - Aici poți scrie programele tale prin conectarea blocurilor din paleta de blocuri. Reține că fiecare sprite va avea propriile sale programe. **View Options** - Aici poți comuta între diferite moduri de vizualizare pentru editorul de proiecte.

**Backpack** - îți permite să transferi cod, sprite-uri și fundaluri între proiectele Scratch.

**Project Options** - Aici poți salva proiectul, adăuga un titlu, partaja proiectul (care îl face public), schimba limba și alte opțiuni.

**Account Options** - Aici poți accesa alte proiecte și alte setări ale contului.



## **Blocuri de cod**

În Scratch, programele sunt scrise prin conectarea blocurilor de cod colorate. Pentru a începe să construiești programe, trage blocurile din paleta de blocuri în spațiul de lucru de codare din centrul editorului de proiecte.

Cor	de 🦪 Costumes	() Sounds			
Motion	Motion				
Looks	move 10 steps		× · · · · · · · · · · · · · · · · · · ·		
	turn C <sup>4</sup> 15 degrees			Hello!	
Sound	tum 🖒 15 degrees		when space - key pressed		
Events	no to random position -		go to x: -180 y: 0		
Control			repeat 20		
Sensing	go to x: 20 y: 0		move 10 steps		
Operators	glide 1 secs to rand	om position -	next costume		
Variables	glide 1 secs to x: 20	y: 0			
My Blocks	point in direction 90		say Hello! for 2 seconds	Sprite Sprite1	Stage
	point towards mouse-po	inter 🔹		Show 🧿 💋 Size 100 Direction 90	
					Backdrops
	change x by 10		$\bigcirc$	Sprite 1	
	set x to 20				
	change y by 10				

În prezent, există 119 blocuri diferite în Scratch, împărțite în 8 categorii: Mișcare (Motion), Aspect (Looks), Sunet (Sound), Evenimente (Events), Control, Senzori (Sensing), Operatori (Operators) și Variabile (Variables). Vom introduce blocuri pe măsură ce apar în codul nostru, dar este foarte util să petreci ceva timp privind prin categorii și familiarizându-te cu diferitele blocuri disponibile. Înțelegerea diferitelor instrumente la dispoziția ta este crucială pentru a fi un programator eficient! Accesează <u>ghidul de blocuri Scratch Wiki</u> pentru informații detaliate despre toate tipurile de blocuri și cum funcționează împreună.

Sfaturi generale pentru lucrul cu blocurile de cod în Scratch:

**Formele blocurilor** - Fii atent la formele blocurilor și la modul în care acestea se conectează între ele. Forma unui bloc îți va spune despre modul în care trebuie utilizat într-un program și cum va funcționa împreună cu alte blocuri.



5

## Blocuri de cod (CONTINUARE)

**Numere și text** - În orice blocuri care conțin numere sau text în interiorul unui spațiu alb, aceste numere sau text pot fi editate la orice valoare dorești! Doar fă clic pe textul din interiorul blocului, șterge-l și scrie orice dorești în locul său.



**Opțiuni din meniul derulant (Dropdown Options)** - Unele blocuri conțin o săgeată mică albă în jos. Opțiuni suplimentare pot fi selectate pentru aceste blocuri. Doar fă clic pe săgeată pentru a deschide meniul și selectează opțiunea dorită.

√ space	
up arrow	
down arrow	
right arrow	
left arrow	
any	
a	
ь	
c	
A	

## Remixare

**Scratch,** pe lângă faptul că este un limbaj de programare și un set de unelte, este o comunitate de programatori și creatori. Poți vizualiza și juca proiecte de la alți utilizatori mergând pe pagina principală Scratch și făcând clic pe "Explore". O caracteristică cheie în Scratch este abilitatea de a remixa proiectele altora. **Remixarea îți permite să adaugi sau să modifici proiectul unui alt utilizator.** Când remixezi proiectul altui utilizator, salvezi o copie a proiectului în propriul tău cont, astfel încât modificările pe care le faci nu vor modifica proiectul original.



Pentru a remixa proiectul altui utilizator, mergi la pagina proiectului și fă clic pe "Remix".

Aceasta va salva o copie a proiectului în contul tău și îl va deschide în editorul de proiecte.



## Remixare (CONTINUARE)

Remixarea și explorarea proiectelor altor utilizatori este o modalitate excelentă de a învăța noi tehnici de programare și design de jocuri!



### Sfaturi pentru remixarea proiectelor Scratch

Joacă-te: Este ușor să fii copleșit când privești pentru prima dată codul din proiectul altui utilizator. Pe măsură ce începi să citești prin codul unui proiect, o modalitate simplă de a începe să înțelegi codul este să schimbi valorile și numerele care apar în diferite blocuri și să vezi cum afectează jocul. Doar amintește-ți să ții evidența a ceea ce schimbi în caz că strici jocul.





### Remixare (CONTINUARE)

costumului.

Schimbarea sprite-urilor (Changing Sprites): Dacă dorești să schimbi un sprite fără a-i schimba codul, fă acest lucru adăugând un nou costum sprite-ului. Pentru a face acest lucru, fă clic pe sprite-ul din meniul sprite-urilor, apoi fă clic pe tab-ul "Costumes" și pe butonul "Choose a Costume" din colțul stâng jos. Alege sprite-ul pe care dorești să-l înlocuiești cu sprite-ul curent.



De asemenea, poți redenumi sprite-ul în meniul sprite-urilor.



GAMES FOR CHANGE STUDENT CHALLENGE

### Proiectul #1: Labirintul din temniță

Pentru primul nostru proiect, vom construi un joc de labirint din temniță! Poți vedea (și remixa) versiunea mea <u>aici</u>.



#### Blocuri noi utilizate

when 🍽 clicked	Din categoria evenimente (From events): Acționează orice blocuri atașate atunci când este făcut clic pe steagul verde de deasupra scenei.
when space  key pressed	Din categoria evenimente (From events): Acționează orice blocuri atașate atunci când este apăsată tasta specificată.
when backdrop switches to backdrop1	<b>Din categoria evenimente (From events)</b> : Acționează orice blocuri atașate atunci când fundalul este schimbat la cel specificat.
move 10 steps	<b>Din categoria mișcare (From motion)</b> : Mișcă sprite-ul cu numărul specificat de pași (pixeli).



### Proiectul #1: Labirintul din temniță

#### (CONTINUARE)

### Blocuri noi utilizate (CONTINUARE)

point in direction 90	<b>Din categoria mișcare (From motion)</b> Orientează sprite-ul în direcția specificată, unde 90 de grade indică spre dreapta
set rotation style left-right -	<b>Din categoria mișcare (From motion)</b> Setează stilul de rotație al sprite-ului, care determină modul în care un sprite se va roti și întoarce
go to x: -82 y: -93	<b>Din categoria mișcare (From motion)</b> Trimite sprite-ul la o poziție X, Y specifică pe scenă
glide 1 secs to x: 187 y: 101	<b>Din categoria mișcare (From motion)</b> Fă un sprite să alunece la o poziție X, Y specifică în timpul specificat
if then	<b>Din categoria control (From control)</b> Verifică o condiție de la începutul blocului. Dacă condiția este adevărată, blocurile din interior vor fi activate
forever	<b>Din categoria control (From control)</b> Repetă orice blocuri din interior pentru totdeauna
stop all -	<b>Din categoria control (From control)</b> Oprește toate sau unele părți ale codului din rulare
touching color ?	<b>Din categoria senzori (From sensing)</b> Verifică dacă sprite-ul atinge o anumită culoare



### Proiectul #1: Labirintul din temniță

#### (CONTINUARE)

#### Blocuri noi utilizate (CONTINUARE)

touching mouse-pointer - ?	<b>Din categoria senzori (From sensing)</b> Verifică dacă sprite-ul atinge cursorul mouse-ului sau alt sprite
or	<b>Din categoria operatori (From operators)</b> Verifică dacă una dintre condiții este adevărată
next backdrop	<b>Din categoria aspect (From looks)</b> Schimbă fundalul la următorul
next costume	<b>Din categoria aspect (From looks)</b> Schimbă la următorul costum
hide	<b>Din categoria aspect (From looks)</b> Face un sprite invizibil
show	<b>Din categoria aspect (From looks)</b> Face un sprite vizibil

### Faza 1: Mișcare de bază

În această primă fază, vom începe proiectul nostru alegând sprite-ul jucătorului și programând mișcarea de bază în patru direcții folosind blocuri de evenimente și mișcare.

<b>Proiect nou:</b> Deschide un nou proiect Scratch accesând <u>scratch.mit.edu</u> , autentificându-te în contul tău și făcând clic pe "Create" în	Create         Explore         Ideas         About         Q. Search           What's Happening?	Scratch News View All
colțul din stânga sus.	This is where you will see updates from Scratchers you follow Check out some Scratchers you might like to follow	Witk Wednesday!           Check out the new Wikk Wednesday forum point a news series highlighting the Scratch Well           Will           Will           Wedsexteen the series of the
		GAMES FOR CHANGE STUDENT CHALLENGE

12

 Alege-ți sprite-ul jucătorului: Fă clic pe butonul "Choose A Sprite" din colțul din dreapta jos al meniului sprite-urilor. Aceasta va deschide biblioteca de sprite-uri.



Pentru acest proiect vom folosi sprite-ul șoricelului ca jucător. Poți schimba acest lucru cu un alt sprite mai târziu, dacă dorești.



Mouse1

După ce ți-ai ales sprite-ul, poți șterge spriteul original al pisicii făcând clic pe coșul de gunoi din colțul din dreapta sus al miniaturii sprite-ului. Astfel, ar trebui să rămâi doar cu noul sprite.

3. **Începe codarea:** Scriptul nostru de mișcare de bază va consta din 3 blocuri. Începe cu blocul "when space key pressed" din categoria evenimente (events). Trage acest bloc în spațiul de codare și folosește săgeata dropdown pentru a modifica acest bloc astfel încât să utilizeze săgeata dreapta în locul tastei space.

**Notă**: În Scratch, programele noastre vor începe aproape întotdeauna cu un bloc din categoria evenimente (events). Observă cum partea de sus curbă a acestora nu permite conectarea altor blocuri deasupra lor.



Următorul pas este să mergi la categoria "**motion**" și să conectezi un bloc "**point in direction 90**" și un bloc "**move 10 steps**".





4. Toate cele patru direcții: Repetă blocul de cod de mai sus pentru celelalte trei direcții.

**Notă**: Poți copia o bucată de cod făcând clic dreapta (control + click) pe blocul de sus din bucată și selectând "**duplicate**".

Pentru fiecare direcție (dreapta, stânga, sus, jos), schimbă blocurile "**when space key pressed**" și "**point in direction 90**" la direcția corectă. Asigură-te că numerele/direcțiile tale corespund combinațiilor de mai jos:



Acum ar trebui să poți mișca sprite-ul tău în toate cele patru direcții! Încearcă acest lucru folosind tastele săgeată.

Dacă mutarea sprite-ului spre stânga face ca sprite-ul tău să se răstoarne cu susul în jos, adaugă următorul cod pentru a împiedica sprite-ul să se răstoarne cu susul în jos:

> 5. Bonus: Fă clic pe tab-ul "Costumes" din stânga sus a ecranului. Dacă sprite-ul tău are mai multe costume care pot fi utilizate pentru a arăta o animație de mers, încearcă să adaugi un bloc "next costume" din categoria aspect (looks) la sfârșitul fiecărei direcții. Acest lucru va funcționa bine doar pentru anumite sprite-uri. ———>





Acest lucru va restricționa sprite-ul să se rotească doar orizontal, nu și vertical.





6. **Titlu și salvare:** În partea de sus a ecranului, adaugă un titlu proiectului tău. Apoi caută cuvintele "Save Now" lângă imaginea unui folder în colțul din dreapta sus al ecranului. Dacă nu vezi aceste cuvinte, înseamnă că proiectul tău a fost deja salvat!

BERTER		File	Edit	🔆 Tutoria	Share	5 See Project Pag	ge Save Now	8	ajnovem	iber 👻
Code	<b>"</b> c	ostumes		) Sounds		<b>N</b>				×

#### Faza 2: Crearea unui fundal (Create a Backdrop)

Acum că am programat mișcarea de bază în patru direcții pentru sprite-ul nostru, este timpul să construim labirintul nostru. Vom face acest lucru desenând un nou fundal.

1. **Deschide funcția paint tool:** Mută cursorul deasupra butonului "**Choose a Backdrop**" din colțul din dreapta jos al ecranului și fă clic pe pictograma pensulei pentru a începe să desenezi un nou fundal.



 Desenează-ți labirintul: Creează un labirint simplu pentru jocul tău. Nu complica lucrurile! Vom crea mai multe niveluri mai târziu, care pot fi mai dificile. Asigură-te că laşi un traseu larg şi o intrare şi ieşire clară pentru labirintul tău.

Folosește instrumentele **dreptunghi** și **linie** pentru a crea forme, iar apoi instrumentul cursor pentru a ajusta dimensiunea și poziția formelor.



 Alege un obiectiv (Choose a goal): Alege un sprite care să fie obiectivul la sfârșitul labirintului. Fă clic pe butonul "Choose A Sprite" din colțul din dreapta jos al ecranului şi alege un sprite din bibliotecă. Deocamdată, vom folosi sprite-ul Cheesy Puffs pentru aceasta.



Plasează Cheesy Puffs la sfârșitul labirintului tău. Făle mai mici găsind câmpul de dimensiune în meniul spriteurilor și schimbând numărul la 50.

Sprite Cheesy Puffs $\leftrightarrow$ x 201 ‡ y 137	Stage
Show Ø Size 50 Direction 90	L L
Mouse1	Backdrops 2

4. **Salvează-ți proiectul.** În colțul din stânga sus al ecranului, fă clic pe **File > Save Now**, doar pentru a te asigura că proiectul este salvat.

#### Faza 3: Setarea regulilor

Acum vom programa unele dintre regulile jocului nostru. Regulile de bază includ următoarele:

- Jucătorul începe la începutul labirintului și trebuie să meargă până la sfârșitul labirintului folosind tastele săgeată.
- Dacă jucătorul atinge un perete, trebuie să se întoarcă la începutul labirintului.
- Când un jucător ajunge la sfârșitul labirintului, poate continua la nivelul următor.





 Redimensionează sprite-ul tău: Probabil că șoricelul tău este prea mare pentru a traversa labirintul tău. Fă-ți sprite-ul mai mic găsind câmpul de dimensiune (size) şi schimbând numărul la 50.





2. Steagul Verde (Green Flag): Blocul "when green flag clicked" din categoria evenimente (events) este adesea folosit ca un buton de start pentru jocurile Scratch. Steagul verde din acest bloc se referă la steagul verde de deasupra scenei. Când este făcut clic pe steagul verde de deasupra scenei, orice blocuri conectate la blocul "when green flag clicked" vor fi declanşate. Semnul roşu de stop de lângă steagul verde poate fi folosit pentru a opri programul tău în orice moment.



3. Coordonatele X şi Y: Pentru a seta poziția de start a șoricelului, va trebui să folosim coordonatele X şi Y pentru a descrie unde ar trebui să înceapă sprite-ul. Coordonatele X şi Y sunt o modalitate comună de a descrie poziția în orice instrumente de design digital pe care le poți folosi. Iată un diagramă care arată cum sunt măsurate valorile X şi Y în Scratch.



**Originea (Origin)** - Originea este punctul în care atât X, cât și Y sunt egale cu 0. Toate coordonatele vor fi măsurate din acest punct. În Scratch, originea se află în centrul scenei.

**Coordonata X (X Coordinate)** - Descrie poziția orizontală a unui obiect, la stânga sau la dreapta originii.

**Coordonata Y (Y Coordinate)** - Descrie poziția verticală a unui obiect, deasupra sau dedesubtul originii.

**Notă**: Poți vedea coordonatele X și Y ale oricărui sprite în orice moment selectând acel sprite și uitându-te la câmpurile X și Y din meniul sprite-urilor.



 Setează poziția de start: Acum vom seta poziția de start a sprite-ului jucătorului nostru. Mai întâi, trage sprite-ul tău jucător la începutul labirintului tău. Această bucată de cod va începe cu un bloc "when green flag clicked" din categoria evenimente (events). Apoi, conectează un bloc "go to x\_y\_" şi un bloc "point in direction 90" din categoria mişcare (motion).



Apasă pe steagul verde și vezi dacă sprite-ul tău merge la începutul labirintului. Dacă nu, plasează sprite-ul jucătorului tău la începutul labirintului și copiază coordonatele din câmpurile X și Y din meniul sprite-urilor.



5. Programarea pereților (Program the walls): Acum, șoricelul nostru poate trece direct prin pereți. Vom scrie un program care să facă șoarecele să se întoarcă la începutul labirintului atunci când atinge un perete. Pentru a face acest lucru, vom folosi un bloc "if\_then>" din categoria Control.



Blocul "if\_ then\_" este unul dintre cele mai importante blocuri din Scratch și este o idee larg utilizată în alte limbaje de programare pe calculator. Acest bloc va cauza apariția unui anumit rezultat dacă este îndeplinită o condiție. În acest caz, afirmația noastră va suna cam așa:

"Dacă șoricelul atinge peretele, atunci șoricelul se întoarce la începutul labirintului."

Putem realiza acest lucru folosind blocul **"touching color\_**" din categoria Senzori (Sensing). Observă cum forma acestui bloc îi permite să se potrivească direct în gaura hexagonală din partea de sus-a-blocului "if \_ **then\_**".



Pentru a seta culoarea pe care o verificăm în blocul **"touching color**", mai întâi fă clic pe cercul colorat din interiorul blocului. Când apare meniul de culori, fă clic pe pictograma pipetei din partea de jos. Acest lucru va întuneca întregul ecran, cu excepția scenei. Acum, adu cursorul pe scenă și fă clic pe culoarea pe care dorești să o detectezi. În cazul nostru, aceasta este culoarea pereților tăi.





Acum, plasează un alt bloc "**go** to X\_Y\_" și un bloc "point in direction" din categoria Mișcare în interiorul părții "then" a blocului "**if\_then**\_". Asigură-te că coordonatele X și Y corespund coordonatelor poziției tale de pornire. Acum poți adăuga acest cod la codul care setează poziția de pornire.



Din păcate, codul de mai sus nu va funcționa complet încă. Așa cum este scris, codul de mai sus va verifica dacă atingem peretele o singură dată, exact în momentul în care este făcut clic pe steagul verde. Pentru a verifica acest lucru în mod repetat, trebuie să adăugăm un ciclu în jurul blocului nostru "**if\_ then\_**". Pentru aceasta, putem folosi un bloc "**forever**" din categoria **Control**.

6. **Încearcă**: Încearcă să-ți conduci șoarecele în perete. Îl trimite înapoi la început, așa cum ar trebui? **Salvează-ți proiectul aici** înainte de a continua.



19

#### Faza 4: Mai multe niveluri!

Acum vom crea mai multe niveluri pentru jocul nostru de labirint și vom scrie codul care ne permite să avansăm prin aceste niveluri!

1. **Desenarea mai multor labirinturi:** Creează două labirinturi suplimentare folosind același proces pe care l-ai folosit pentru a crea primul. Păstrează intrarea și ieșirea labirinturilor în același loc. Încearcă să faci aceste labirinturi mai provocatoare, dar nu imposibile!



 Gestionarea fundalurilor (Backdrop management): Organizează-ți fundalurile astfel încât să ai trei fundaluri numite "backdrop1", "backdrop2" şi "backdrop3", cu "backdrop1" fiind cel mai uşor şi "backdrop3" fiind cel mai greu. Acest lucru poate implica redenumirea unor fundaluri şi ştergerea celor goale.





3. Setarea obiectivului: Acum vom scrie codul care permite jucătorului nostru să treacă la nivelul următor atunci când șoarecele ajunge la Cheesy Puffs. În primul rând, adaugă un alt bloc "if\_ then\_" la ciclul nostru "forever". De data aceasta, vom folosi blocul "touching" din categoria Senzori ca și condiție. Folosește săgeata dropdown din interiorul blocului "touching" pentru a selecta Cheesy Puffs.



Avem nevoie de două lucruri să se întâmple atunci când șoricelul ajunge la Cheesy Puffs: Trebuie să avansăm la nivelul următor și trebuie să trimitem șoricelul înapoi la începutul labirintului. Vom reseta șoricelul folosind aceleași blocuri pe care le-am folosit mai devreme și vom adăuga un bloc "**next backdrop**" din categoria **Aspect** (Looks) pentru a avansa la următorul fundal.



4. Setarea fundalului de pornire: Acum că avem mai multe fundaluri, trebuie să ne asigurăm că începem cu cel corect atunci când jocul începe. Pentru a face acest lucru, accesează categoria Aspect și adaugă un bloc "switch backdrop to backdrop1" imediat sub blocul "when green flag clicked". Folosește săgeata dropdown pentru a selecta fundalul corect în interiorul acestui bloc, dacă este necesar.

Testează: Finalizează primul nivel al labirintului tău. Apare al doilea nivel după ce termini primul? Poți avansa de la al doilea la al treilea? Acum este un alt moment bun pentru a-ți salva proiectul.





GAMES FOR CHANGE STUDENT CHALLENGE

#### Faza 5: Retușuri finale

Acum vom adăuga câteva retușuri finale jocului nostru.

1. Ecranul de victorie: Creează un nou fundal care să acționeze ca ecranul tău de victorie! Acest fundal va apărea când jucătorul a terminat toate nivelurile. Numește-l "backdrop4". Personalizează acest fundal cum dorești!



 Oprirea jocului: Adaugă următoarea combinație de două blocuri la codul tău. Aceasta va opri totul când ajungi la sfârșitul jocului. "When backdrop switches to \_" poate fi găsit în categoria Evenimente, și "stop all" în categoria Control. Schimbă fundalul menționat în blocul "when backdrop switches" dacă este necesar.

when	backdro	op sı	witche	es to	bac	kdrop	4 🕶
stop	all 🔻				а.	÷.	5

#### Faza 6: Provocare bonus

Pentru o provocare bonus, poți adăuga un inamic în labirintul din temniță.

1. Alege un sprite: Începe prin adăugarea unui nou sprite care să fie fie inamicul tău. Eu voi folosi sprite-ul "Cat 2" pentru aceasta.



Cat 2



2. **Programează comportamentul:** Programează comportamente pentru sprite-ul inamic. Folosește blocul "**when backdrop switches to**" din categoria Evenimente pentru a programa comportamente diferite pentru fundaluri diferite. Aici folosesc blocurile "**show**" și "**hide**" din categoria Aspect pentru a face sprite-ul să apară doar pe unele niveluri. Pe nivelurile unde apare pisica, folosesc blocurile "**glide \_ secs to x y\_**" în interiorul unui ciclu "**forever**" pentru a programa mișcarea automată a sprite-ului. Va trebui să decizi unde se va deplasa sprite-ul tău inamic, în funcție de structura labirintului tău!



3. Adaugă o regulă: În cele din urmă, vom modifica regulile astfel încât, dacă jucătorul loveşte pisica, să fie trimis înapoi la început. Pentru a face acest lucru, vom folosi un bloc "or" din categoria Operatori și un alt bloc "touching\_" pentru a adăuga o a doua condiție la blocurile "if\_ then\_" care verifică dacă atingem pereții.

when P clicked						
switch backdrop to backdrop	1 -					
point in direction 90						
go to x: -221 y: 128						
forever						
if touching color	)?)	or to	uching	Cat 2	• ?	then
point in direction 90						
go to x: -221 y: 128						
If touching Cheesy P	uffs 🔻	? t	hen			
point in direction 90						
go to x: -221 y: 128						
next backdrop						
<b>J</b>						



#### Testare

Acum că proiectul tău este într-o stare jucabilă, petrece ceva timp testându-l și lăsându-i pe alții să-l joace. Iată câteva întrebări care să te ghideze în timpul testării:

- Funcționează jocul așa cum vrei?
- Există comportamente neașteptate în joc?
- Jocul este distractiv de jucat?
- Poți ajunge și bate fiecare nivel?
- Jocul pare prea ușor, prea greu sau exact potrivit?
- Ce ar putea fi adăugat pentru a face jocul mai provocator?
- Ce ar putea fi adăugat pentru a face jocul mai distractiv?
- Există ceva confuz sau inutil care ar trebui eliminat?

O altă modalitate de a aborda testarea jocului este să folosești acronimul **T.A.G.** TAG înseamnă:

<u>Tell</u> (Spune) ceva ce îți place.
<u>Ask</u> (Pune) o întrebare.
<u>Give</u> (Oferă) o sugestie pentru feedback.

Aceasta poate fi o sugestie utilă pentru cei care testează și oferă feedback pentru jocul tău, sau pentru tine când testezi jocurile altora.

#### Personalizare și iterare

Ia în considerare feedbackul primit în timpul testării și gândește-te la modalități de a îmbunătăți și personaliza proiectul tău! Iată câteva idei despre cum să duci acest proiect mai departe:

- Schimbă sprite-urile pe care le folosești
- Adaugă mai multe niveluri jocului tău
- Adaugă obiecte colectabile în joc
- Adaugă mai mulți inamici sau obstacole
- Adaugă o limită de timp pentru finalizarea fiecărui nivel
- Îmbunătățește detaliile sau designul fundalurilor



## Proiectul **#2:** Nu lăsa gogoșile să cadă

Pentru al doilea nostru proiect, vom **face un joc în care jucătorul trebuie să prindă obiecte care cad.** Poți vedea (și remixa) versiunea mea **aici**.



### Noi blocuri utilizate





### Project #2: Don't Drop the Donuts (CONTINUED)

### New Blocks Used (CONTINUED)

turn (~ 15 degrees	<b>Din categoria mișcare (From motion)</b> Rotește un sprite cu valoarea specificată
y position	Din categoria mișcare (From motion) Poziția Y curentă a unui sprite
< 50	<b>Din categoria operatori (From operators)</b> Compară două numere. Condiția este îndeplinită dacă primul număr este mai mic decât al doilea
pick random 1 to 10	<b>Din categoria operatori (From operators)</b> Alege un număr aleatoriu între cele două valori specificate
mouse x	<b>Din categoria senzori (From sensing)</b> Poziția X curentă a mouse-ului
wait 1 seconds	<b>Din categoria control (From control)</b> Așteaptă pentru timpul specificat
set my variable - to 0	<b>Din categoria variabile (From variables)</b> Setează valoarea unei variabile
change my variable - by 1	<b>Din categoria variabile (From variables)</b> Schimbă valoarea unei variabile



#### Faza 1: Configurarea proiectului

În această primă fază, vom deschide un nou proiect și vom alege fundalul și sprite-urile noastre.

- 1. **Proiect nou:** Deschide un nou proiect Scratch accesând scratch.mit.edu, autentificându-te în contul tău și făcând clic pe "Create" în colțul din stânga sus. Adaugă un sprite pentru jucător și un fundal proiectului tău.
  - Pentru sprite-ul jucătorului, orice animal, persoană sau personaj va funcționa bine.
  - Pentru fundal, ceva simplu și nu prea aglomerat va funcționa cel mai bine.



 Obiect care cade: Adaugă un alt sprite care să fie obiectul care cade. Eu voi folosi sprite-ul de gogoaşă, dar orice mâncare sau alt obiect va funcționa bine. Schimbă dimensiunea obiectului care cade la 40 în câmpul de dimensiune al meniului spriteurilor.





#### Faza 2: Programarea obiectului care cade

Acum vom scrie codul pentru obiectul care cade.

 Căderea continuă: Selectează sprite-ul obiectului care cade și scrie următorul cod:



Ține minte că **coordonata Y** a unui sprite este poziția sa verticală, așa că schimbarea Y cu -10 va face ca sprite-ul să pară că cade. Fă clic pe steagul verde pentru a vedea dacă obiectul cade până la sol.

2. Trimiterea înapoi în sus: Acum vom adăuga un bloc "if\_then\_" pentru a trimite sprite-ul înapoi în partea de sus a scenei atunci când ajunge la partea de jos. Acest lucru va implica utilizarea blocului "<" din categoria Operatori și a blocului "Y position" din categoria Mișcare. Putem folosi aceste blocuri pentru a verifica dacă sprite-ul a ajuns la partea de jos a scenei. Când se întâmplă acest lucru, vom folosi un bloc "set Y to \_" pentru a-l trimite înapoi în partea de sus a ecranului.</p>

**Încearcă!** Acum sprite-ul tău ar trebui să cadă până la partea de jos a ecranului, să reapară în partea de sus a ecranului și să continue să cadă.



3.

Randomizarea poziției: De asemenea, dorim să randomizăm poziția X, astfel încât sprite-ul să nu cadă de fiecare dată în același loc. Pentru a face acest lucru, vom folosi un bloc "pick random \_ to \_" din categoria Operatori și un bloc "set x to \_". Vom adăuga acest lucru imediat după blocul "set y to \_".

Acum sprite-ul tău ar trebui să cadă până la partea de jos a ecranului, să reapară în partea de sus a ecranului într-o **poziție x** aleatorie și să continue să cadă.





4. Fă-l să se rotească: Doar pentru distracție, adaugă un bloc "turn \_ degrees" (în orice direcție) din categoria Mişcare pentru a face obiectul care cade să se rotească. Fă acest lucru în interiorul ciclului "forever", dar în afara blocului "if\_ then\_".



#### Faza 3:Controale pentru jucător

Acum vom programa mişcarea pentru sprite-ul jucătorului. În ultimul nostru proiect, am programat mişcarea care era controlată folosind tastele de pe tastatură. În acest proiect, vom programa mişcarea controlată de mouse.

- Setează poziția de pornire: Plasează sprite-ul jucătorului acolo unde dorești să înceapă. Apoi folosește un bloc "when green flag clicked" și un bloc "go to x\_ y\_" pentru a seta poziția de pornire pentru sprite-ul tău.
- Realizarea mişcării: Acum vom programa mişcarea pentru jucătorul nostru. De data aceasta, vom folosi mouse-ul pentru a controla jucătorul. Vom folosi un ciclu "forever", un bloc "set x to \_" şi blocul "mouse x" din categoria Senzori.

Încearcă! Urmează sprite-ul tău mouse-ul? Salvează-ți jocul înainte de a merge mai departe!







#### Faza 4: Păstrarea scorului

Pentru a păstra scorul, va trebui să folosim variabile. Variabilele permit unui program să stocheze și să țină evidența unui număr.

 Creează o variabilă: Mergi la categoria Variabile în paleta de blocuri şi fă clic pe "Make a Variable". Denumeşte-ți variabila "Scor" şi fă clic pe OK.

Code	Costumes 📢 Sounds	New Variable
Motion	variables	
	Make a Variable	
Looks	my variable	New variable name:
Sound	set my variable 🕶 to 🛛 🔹	Score
		00010
Events	change my variable - by 1	
Control	show variable my variable -	For all sprites O For this sprite only
Sensing	hide variable my variable 🔻	
		Cloud variable (stored on server)
Operators	Make a List	
Variables	My Blocks	Cancel OK
	Make a Block	
My Blocks		

 Resetează scorul: Trebuie să ne asigurăm că atunci când jocul începe, scorul nostru este setat la 0. Mai întâi, revino la codul obiectului care cade. Ia un bloc "set my variable to 0" din categoria Variabile și folosește săgeata dropdown de pe bloc pentru a schimba "my variable" în "Scor". Adaugă acest bloc în partea de sus a codului obiectului care cade, direct sub blocul "when green flag clicked".



- 3. **Setează regulile:** Acum este timpul să programăm regulile pentru acest joc. Iată cele două reguli principale:
  - The player will earn points by catching falling objects
  - Scorul jucătorului va reveni la 0 când un obiect este ratat

Pentru a reseta scorul la 0, putem pur și simplu să adăugăm un alt bloc "**set score to 0**" în interiorul blocului "**if\_ then\_**" existent.





4. Marcarea punctelor: Pentru a permite jucătorului să prindă obiecte care cad pentru a marca puncte, trebuie să adăugăm un alt bloc "if\_ then\_" cu un bloc "touching \_" din categoria Senzori. Când jucătorul prinde un obiect care cade, trebuie să trimitem obiectul înapoi în partea de sus a ecranului (la fel cum facem când atinge solul) și să adăugăm 1 la scor folosind un bloc "change score by 1" din categoria Variabile.

Încearcă! Testează-ți jocul. Urmărește scorul în timp ce prinzi (și ratezi) obiecte care cad. Funcționează totul așa cum vrei? Salvează-ți jocul înainte de a merge mai departe!



#### **Faza 5:** Retușuri finale

 Accelerează-l: În acest moment, jocul este probabil foarte uşor. Vom programa jocul pentru a deveni mai dificil pe măsură ce jucătorul acumulează mai multe puncte. Pentru a face jocul mai dificil, vom face ca obiectele să cadă mai repede.

> Valoarea "-10" din blocul "change y by -10" determină cât de repede cad obiectele noastre.



Va trebui să facem câteva calcule pentru a ajusta viteza în funcție de scorul nostru. Pentru a face acest lucru, vom folosi un bloc de scădere (substraction) din categoria **Operatori** și blocul "**Score**" din categoria **Variabile**.





 Adăugarea unui scor maxim:În cele din urmă, vom adăuga o modalitate de a ține evidența scorului maxim. Mergi la categoria variabile (Variables) și creează o nouă variabilă, aceasta numită "Top Score".

New Variable
New variable name:
Top Score
● For all sprites ○ For this sprite only
□ Cloud variable (stored on server)
Cancel

**Apoi**, scrie următorul cod pentru a actualiza scorul maxim de fiecare dată când scorul curent îl depășește.



După aceea, adaugă acest cod în partea de jos a buclei "forever" a obiectului care cade. De asemenea, resetează scorul maxim (Top Score) la 0 la începutul jocului nostru.

#### when clicked set Score v to 0 set Top Score v to 0 forever change y by 5 - Score turn 2 5 degrees If y position < 160 then set y to 200 set x to pick random -240 to 240 set Score v to 0 If touching Rooster v ? then set y to 200 set x to pick random -240 to 240 change Score v to 0 If Score v to 0 ff Score v to 0 ff Score v to 1 ff Score v to Score turn -240 to 200 set x to pick random -240 to 240 change Score v to 1 ff Score v to Score

#### Faza 6: Provocare bonus

Pentru o provocare bonus, încearcă să adaugi la joc un al doilea obiect care cade și pe care jucătorul trebuie să îl evite.

1. Adaugă un nou sprite: Adaugă un nou sprite în proiectul tău. Eu voi folosi s**prite-ul crab** pentru aceasta. Setează dimensiunea acestui sprite la **50**.





2. Programează-l: Acum vom programa acest sprite. Codul va semăna cu cel al obiectului care cade, dar cu câteva diferențe. Acest sprite va aştepta 5 secunde "wait 5 seconds" după fiecare cădere şi va folosi blocurile "hide" şi "show" pentru a-l face invizibil în timpul aşteptării. Acest sprite va reseta, de asemenea, scorul nostru la 0 dacă este prins de jucător.



#### Testare

Acum că proiectul tău este într-o stare jucabilă, petrece ceva timp testându-l și lăsându-i pe alții să-l joace. Iată câteva întrebări care să te ghideze în timpul testării:

- Funcționează jocul așa cum vrei?
- Există comportamente neașteptate în joc?
- Jocul este distractiv de jucat?
- Jocul pare prea ușor, prea greu sau exact potrivit?
- Ce ar putea fi adăugat pentru a face jocul mai provocator?
- Ce ar putea fi adăugat pentru a face jocul mai distractiv?
- Există ceva confuz sau inutil care ar trebui eliminat?

#### Personalizare și iterare

Ia în considerare feedbackul primit în timpul testării și gândește-te la modalități de a îmbunătăți și personaliza proiectul tău! Iată câteva idei despre cum să duci acest proiect mai departe:

- Schimbă sprite-urile pe care le folosești
- Adaugă sunete jocului tău
- Adaugă mai multe tipuri de obiecte care cad
- Schimbă controalele de mișcare
- Adaugă o limită de timp pentru joc
- Folosește schimbări de fundal pentru a arăta schimbările de nivel sau de dificultate
- Îmbunătățește detaliile sau designul fundalurilor



## Proiectul #3: Bătălia cu boss-ul

Pentru al treilea nostru proiect, vom **crea o versiune a unei bătălii cu un boss dintr-un joc video**. Bătălia mea va fi între un vrăjitor, controlat de jucător, și un dragon care scuipă foc, inamic. Poți vedea (și remixa) versiunea mea **aici**.



#### Blocuri noi utilizate





### Project #3: Boss Battle (CONTINUED)

#### New Blocks Used (CONTINUED)







#### Faza 1: Configurarea proiectului

În această primă fază, vom deschide un proiect nou și ne vom alege fundalul și sprites-urile.

 Adaugă-ți sprite-urile: Deschide un nou proiect Scratch accesând <u>scratch.mit.edu</u>, autentificându-te în contul tău și făcând clic pe "Create" în colțul din stânga sus. Adaugă un sprite pentru jucător și un sprite inamic în proiectul tău. Eu folosesc vrăjitorul pentru jucătorul meu și dragonul pentru inamic. Ajustează dimensiunea sprite-urilor la dimensiunea dorită.



2. Adaugă un fundal: Adaugă un fundal în proiectul tău.





#### Faza 2: Mișcarea jucătorului

Acum vom programa mișcarea pentru jucătorul nostru. Vom învăța o metodă nouă și îmbunătățită pentru a controla un jucător folosind tastele de pe tastatură.

1. Mişcare "if/then": În primul nostru proiect, am folosit blocurile "when key \_ pressed" din categoria Evenimente pentru a declanşa blocurile noastre de mişcare. Aceasta funcționează, dar creează o mişcare care nu este foarte receptivă sau lină. În acest proiect, vom folosi blocurile "if\_ then\_" împreună cu blocul "key \_ pressed?" din categoria Senzori pentru a crea o mişcare mai lină. Vom folosi aceleaşi blocuri "point in direction \_" şi "move 10 steps" pe care le-am folosit pentru mişcarea noastră în trecut.

if key right arrow 👻	) pr	essed	?	then
point in direction 90				
move 10 steps				

 Buclă "forever" (Loop it): Pentru ca acest cod să funcționeze corect, trebuie să-l punem într-o buclă "forever" pentru a verifica constant dacă tasta este apăsată. Vom începe cu un bloc "when green flag clicked".



Testează codul: Fă clic pe steagul verde şi foloseşte tastele săgeată pentru a te mişca!
 Asigură-te că proiectul tău este salvat.

 Două direcții: Adaugă o altă buclă "if\_ then\_" pentru a programa mişcarea spre stânga.

foreve							
	key	right an	row 🔻	pr	essed	?	the
p	oint in dir	ection	90			•	
	10	etone					
		Sieps					
		steps					
if	kev	left arro	w <b>-</b>	pres	sed?	> t	hen
if	key	left arro	w 🔻	pres	sed?	t	hen
if	key oint in dir	left arro	-90	pres	ssed?	t	hen
if	key oint in dir	left arro	-90	pres	ssed?	) ti	hen

4. **Două direcții:** Adaugă o altă buclă "**if\_ then\_**" pentru a programa mișcarea spre stânga.





#### Faza 3: Mișcarea inamicului

Acum vom programa miscarea pentru sprite-ul inamicului nostru.

1. Puterea randomizării: În jocurile video, randomizarea

este extrem de importantă pentru programarea evenimentelor și comportamentelor imprevizibile, care sunt esențiale pentru a face jocurile provocatoare și distractive. Vom folosi blocul "**pick random \_ to** " împreună cu blocul "**glide 1 secs to x y \_**" pentru a programa mișcarea sprite-ului inamic.

glide	1	secs to x:	pick random	-200	to	200	) y: (	pick random	100	to	150

Acest cod va trimite sprite-ul nostru într-o poziție X,Y randomizată aproape de partea superioară a scenei în fiecare secundă.

2. **Buclă "forever" (Loop it):** Vom plasa acest bloc într-o buclă "**forever**" pentru a-l repeta. Vom începe din nou cu un bloc "**when green flag clicked**".

when 💌 clicked													
forever													
glide 1 secs to x:	pick	rand	lom (	-200	to	200	<b>y</b> :	pick	rand	lom (	100	to (	150
<b>ر</b>	×		ι.	÷	÷	i.	,		×	×.	÷.		×.

3. Întoarce-te și privește (Turn and face): Adaugă un bloc "point towards \_" pentru a face inamicul să se orienteze către jucătorul nostru.

when 🏴 clicked												
set rotation style left-right -												
forever												
glide 1 secs to x: pick rat	ndom	-200	to	200	y:	pic	k ran	dom	100	to	150	
<b>J</b>	1	*	1						4			

**Încearcă!** Fă clic pe **steagul verde** și urmărește cum inamicul tău începe să zboare în jur!



#### Faza 4: Atac!

Acum vom crea un atac cu mingi de foc pentru sprite-ul inamic!

 Desenează un sprite nou: Mută cursorul deasupra butonului "Choose a Sprite" și fă clic pe pensula pentru a deschide meniul de desenare.



2. **Desenează-ți sprite-ul**: Folosește instrumentul de desen pentru a crea un sprite care să reprezinte o minge de foc. Micșorează sprite-ul folosind câmpul de dimensiune din meniul sprite-urilor dacă este necesar.





3. Programează mingea de foc: Mingea de foc trebuie să înceapă din locul în care se află sprite-ul inamic. Începem cu un bloc "when green flag clicked", un ciclu "forever" şi un bloc "go to \_" pentru a face acest lucru. Selectează sprite-ul inamic în blocul "go to".

Apasă pe **steagul verde** pentru a încerca. Acest lucru va face ca mingea de foc să apară în centrul sprite-ului inamic. Vom ajusta acest lucru în următorul pas pentru a face ca mingea de foc să apară în fața sprite-ului nostru.

4. Ajustează poziția: Trebuie să mutăm mingea de foc în fața sprite-ului inamic. Locul unde o plasăm va depinde de direcția în care este orientat sprite-ul inamic, pe care o putem determina folosind blocul din categoria Senzori (Sensing) pentru a obține direcția sprite-ului.







Acest bloc extrem de util poate fi utilizat pentru a obține informații despre alte sprites sau despre scenă. Selectează sprite-ul inamic folosind al doilea dropdown și apoi "direcția" folosind primul.



Acum va trebui să folosim un bloc "\_>\_" în interiorul unui bloc "**if** \_ **then** \_ **else** \_" pentru a poziționa mingea de foc în funcție de direcția în care este orientat sprite-ul inamic. Folosește blocurile "**change x by** \_" și "**change y by** \_" pentru a plasa sprite-ul mingii de foc acolo unde dorești. Notează utilizarea valorilor pozitive și negative în blocurile "**change x by** \_".



→ Acum, mingea de foc rămâne în fața sprite-ului inamic, indiferent de direcția în care acesta se deplasează.



5. Ținteşte, foc! (Aim, fire!): Acum vom programa mingea de foc să țintească şi să zboare către sprite-ul jucătorului nostru. Vom folosi un bloc "point towards\_" şi un bloc "move 10 steps" în interiorul unui ciclu. Vom folosi un ciclu "repeat until\_" pentru a menține mingea de foc în mişcare până când poziția Y a acestuia este prea joasă.

**Încearcă!** Apasă pe **steagul verde** și urmărește cum mingile de foc plouă de la sprite-ul inamic asupra sprite-ului jucătorului.



#### Faza 5: Contraatac!

Acum vom crea un atac de tip fulger pentru sprite-ul jucătorului nostru.

 Desenează un sprite nou: Mută cursorul deasupra butonului "Choose a Sprite" și fă clic pe pensulă pentru a deschide meniul de desenare. Desenează un nou sprite care să reprezinte un atac cu fulgere pentru sprite-ul jucătorului tău.





 Fă-l să zboare (Make it fly): Vom începe cu un cod similar celui pe care tocmai l-am scris pentru mingea de foc. Fulgerul va începe din locul în care se află sprite-ul jucătorului şi va zbura în sus până se apropie de partea de sus a scenei.



 Apoi vom adăuga un bloc "wait until \_" şi câteva blocuri "hide"/"show" pentru a face ca fulgerele să fie lansate şi să apară doar atunci când bara de spațiu este apăsată.



#### Faza 6: Sănătatea

Acum vom crea variabile pentru sănătatea jucătorului și sănătatea inamicului și vom programa aceste variabile să interacționeze cu atacurile noastre.

 Creează variabile (Create variables): Mergi la categoria Variabile și creează două variabile noi: "Enemy Health" și "Player Health".





 Setează sănătatea de început (Set starting health): Mergi la codul sprite-ului jucătorului și adaugă un bloc "set Player Health to 10" la începutul codului, sub blocul "when green flag clicked".



 Sănătatea inamicului (Enemy Health): Fă același lucru pentru sprite-ul inamic, de data aceasta setând "Enemy Health" la 10.



#### Programează sănătatea jucătorului (Program

4. Player Health): Mergi la codul mingii de foc. Adaugă un bloc "if \_ then \_" pentru a scădea sănătatea jucătorului şi apoi "hide" mingea de foc dacă acesta atinge jucătorul. Asigură-te că adaugi acest lucru în interiorul ciclului "repeat until". De asemenea, adaugă un bloc "show" pentru a face ca mingea de foc să reapară când începe să se mişte.





 Sănătatea inamicului (Enemy Health): Fă același lucru pentru sprite-ul inamicului, de data aceasta setând "Enemy Health" la 10.

when P click	ed .				
forever					
hide					
wait until	key spa	ice 🔻	pre	essed	?
go to Wiza	rd 🔻				
show ,					
repeat until <	y pos	sition	) > (	150	
change y b	y 10				
if to.	uching (I	Drago	n 🔻	?	then
change	Enemy I	lealth		by	-1
hide	9 - 9			×.	

6. Programează sănătatea inamicului (Program Enemy Health): Mergi la codul sprite-ului fulgerului. Adaugă un bloc "if \_ then \_" pentru a scădea sănătatea inamicului și apoi "hide" fulgerul dacă acesta atinge inamicul. Asigură-te că adaugi acest lucru în interiorul ciclului "repeat until".





#### Faza 7: Retușuri finale

Acum vom programa sfârșitul bătăliei noastre cu boss-ul, care va avea loc atunci când oricare dintre spriteuri rămâne fără sănătate.

1. **Creează un sprite banner (Create banner sprite)**: Vom crea un nou sprite pentru a afișa "**you win**" sau "**game over**" atunci când jocul se termină. Pentru a face acest lucru, pictează un nou sprite și creează două costume pentru el - unul pentru fiecare rezultat. Denumește costumele "**win**" și "**lose**".



 La final: Scrie următorul cod pentru a schimba costumul corespunzător și pentru a afișa sprite-ul atunci când fie jucătorul, fie inamicul rămân fără sănătate.



 Încearcă! Acum că toate piesele jocului tău sunt la locul lor, încearcă-l şi vezi dacă poți învinge inamicul!



#### Faza 8: Provocare bonus:

Creează niște animații pentru sprite-urile jucătorului și inamicului atunci când sunt lovite de un atac.

Transmiterea mesajelor 1. (Broadcast): Blocurile "Broadcast" și "when I receive \_" din categoria evenimente (Events) ne oferă o modalitate simplă de a declanșa evenimente între sprite-uri. Trimite un mesaj cu un bloc "**broadcast**" într-un sprite și declanșează alte blocuri cu un bloc "when I receive \_" într-un alt sprite. Adaugă un bloc "**broadcast**" în scriptul fulgerului imediat după blocul "change Enemy Health by -1". Transmite un nou mesaj numit "**Enemy** hit" făcând clic pe săgeata dropdown din bloc, apoi pe "**new** message" și denumește mesajul. Fă același lucru în sprite-ul mingii de foc cu un mesaj numit "Player hit".





2. **Primirea mesajului (Receive the broadcast)**: În ambele sprite-uri, jucător și inamic, folosește un bloc "**when I receive** \_" pentru a primi mesajul corespunzător și pentru a aplica **efecte grafice**, schimbări de costum sau orice altceva! Folosește un bloc "**wait \_ secs**" și un bloc "**clear graphic effects**" din categoria Aspect (Looks) pentru a readuce sprite-ul la normal la sfârșitul animației.





#### Testare

Acum că proiectul tău este într-o stare jucabilă, petrece ceva timp testându-l și lăsându-i pe alții să-l joace. Iată câteva întrebări care să te ghideze în timpul testării:

- Funcționează jocul așa cum vrei?
- Există comportamente neașteptate în joc?
- Jocul este distractiv de jucat?
- Jocul pare prea ușor, prea greu sau exact potrivit?
- Ce ar putea fi adăugat pentru a face jocul mai provocator?
- Ce ar putea fi adăugat pentru a face jocul mai distractiv?
- Există ceva confuz sau inutil care ar trebui eliminat?

#### Personalizare și iterare

Ia în considerare feedbackul primit în timpul testării și gândește-te la modalități de a îmbunătăți și personaliza proiectul tău! Iată câteva idei despre cum să duci acest proiect mai departe:

#### Schimbă sprite-urile pe care le folosești:

- Adaugă sunete jocului tău
- Adaugă mai multe tipuri de atacuri
- Schimbă controalele de mișcare
- Adaugă o limită de timp pentru joc
- Folosește schimbări de fundal pentru a arăta schimbările de nivel sau de dificultate
- Îmbunătățește detaliile sau designul fundalurilor



## **Proiect f inal: Creează-ți propriul joc de impact!**

Acum că ai construit trei jocuri în Scratch (și de la zero!), este timpul să folosești noile tale abilități pentru a crea un joc de impact.

#### Ce este un joc de impact?

Jocurile de impact sunt jocuri care abordează probleme reale și încurajează jucătorii să învețe și să acționeze asupra unor probleme importante din lumea reală! Acestea pot fi probleme mari sau mici.

#### Pași:

- 1. Alege-ți problema (Choose your issue): Care este problema sau provocarea din lumea reală pe care o vei aborda cu jocul tău de impact? Poate fi ceva legat de comunitatea ta locală sau ceva care afectează întreaga lume.
- 2. **Aprofundează problema (Dive into your issue):** Cine este afectat de această problemă? Ce o cauzează? Unde și când se întâmplă? Care sunt posibilele soluții la problemă? Ce informații ar trebui să știe publicul despre această problemă și soluțiile posibile?
- 3. **Gândește-te la jocul tău (Brainstorm your game):** Poți face o conexiune între problema aleasă și vreun joc pe care l-ai jucat sau creat? Decide ce tip de joc de impact vei crea. Jocul tău poate fi bazat pe unul dintre jocurile pe care le-am făcut deja, ceva creat de la zero sau un remix al unui proiect existent.
- 4. **Planifică jocul tău (Plan your game):** Ce sprite-uri vei avea nevoie? Ce va face fiecare sprite? Ce fundaluri vor fi necesare? Cum va fi experiența jucătorului? Fă un plan pentru realizarea proiectului înainte de a începe programarea.
- 5. Creează jocul tău (Make your game): Intră în Scratch și construiește-ți jocul!
- 6. **Testare (Playtesting):** Lasă pe altcineva să joace jocul tău și cere-i să îți ofere feedback. Caută erori sau comportamente neașteptate în joc.
- 7. Îmbunătățește (Improve): Pune în practică feedbackul primit în timpul testării, rezolvă erorile și îmbunătățește jocul tău.



48