

Joc educativ dedicat recunoasterii sunetelor produse de animale

Autor: Ionescu Vlad-Andrei, 331AA

An: III

Cuprins

1	Introducere	3
2	Contextul teoretic al lucrării	3
2.1	Realizări similare	3
2.2	Aspecte legate de multimedia	4
3	Prezentare tehnică a etapei de realizare/implementare	6
3.1	Biblioteci folosite	6
3.2	Contribuția personală	8
4	Mod de utilizare	10
5	Concluzii	12

1 Introducere

Pentru acest proiect am realizat un joc de recunoastere al sunetelor produse de diverse animalele domestice, aducand o valoare educativa jocurilor pe calculator. Am ales aceasta tema deoarece, in ziua de azi, copiii petrec foarte mult timp in fata calculatorului, iar ca parinte trebuie sa ai grija ca acest aspect sa nu afecteze dezvoltarea celui mic. Nu consider o solutie buna interzicerea folosirii acestor device-uri, ci indrumarea copilului spre a interactiona cu un content care sa nu fie nociv, si de ce nu, educativ, astfel ca am creat un joc, care, pe langa faptul ca este relaxant si interactiv, aduce si o plus valoare prin caracterul educativ.

Premisa jocului este simpla, auzi un sunet si trebuie sa ghicesti carui animal ii corespunde. Jocul are mai multe nivele de dificultate, cu cat alegi o dificultate mai ridicata, cu atat trebuie sa raspunzi mai rapid in vederea obtinerii unei note bune. Nota este calculata in functie de numarul de raspunsuri corecte si de timpul de raspuns.

In prezent, interesul pentru dezvoltarea de jocuri educative este foarte mare deoarece copiii din ziua de azi petrec foarte mult timp in fata ecranelor, dezvoltatorii de jocuri vazand din asta o oportunitate de a face profit de pe urma dorintei parintilor de a le oferi celor mici content educativ.

Majoritatea jocurilor de acest fel au o problema majora, faptul ca sunt plictisitoare si nu pot capta atentia celor mici. Jocul meu isi propune sa rezolve aceasta problema nu doar prin aspectul placut si gameplay-ul interactiv, ci si prin implementarea mai multor dificultati, astfel ca cel mic va fi captat de joc si, foarte probabil, va dori sa ia nota maxima si pe dificultatea „hard”, astfel ca va depune efort in a invata sa distinga sunetele.

2 Contextul teoretic al lucrarii

2.1 Realizari similare

In continuare, voi trece in revista cateva jocuri similare, identificand puncte tari si lipsuri ale acestora.

Un prim exemplu il reprezinta „Guess the Animal Sound”, realizat de „Educagaming.com”, un site dedicat jocurilor didactice. Acesta, insa, prezinta cateva puncte slabe in comparatie cu proiectul meu. Experienta este complet rigida, jucatorul nu poate personaliza in niciun fel gameplay-ul. Jocul meu ofera posibilitatea de a selecta atat nivelul de dificultate, cat si numarul de runde, pe cand, jocul concurentei este pur si simplu un „loop” infinit in care alegi animale la nesfarsit.

Un alt aspect negativ este lipsa unei metode prin care sa urmarim progresul copilului. Jocul meu ofera, la sfarsitul fiecarei sesiuni, o nota, jucatorul avand posibilitatea, ulterior, in meniu, sa vada top 3 cele mai mari note, dar si o medie a tuturor notelor, astfel isi poate urmari progresul. Jocul realizat de „Edugaming.com” nu are niciun fel de metrica care sa arate cum s-a descurcat copilul, lipsa unui sistem de notare, dar si al unor nivele de dificultate mai ridicate, fac ca jocul sa fie plictisitor, neavand o miza.

Jocul concurentei aduce, inasa, si o inovatie, optiunea de „Explore”, unde poti apasa pe imaginea unui animal si auzi sunetul produs de acesta, o metoda foarte buna si interactiva de invatare. Jocului meu ii lipseste o metoda non-competitiva de invatare, singura metoda fiind in cadrul rundelor, unde exista presiune cauzata de sistemul de notare, care afecteaza statisticile jucatorului.

Un al doilea exemplu este “Animal Sounds”, realizat de „Nau.kids”. Acesta este disponibil pe diverse site-uri, precum „crazygames-poki.com”, creatorul neavand un site oficial. Acest joc prezinta aceleasi dezavantaje identificate anterior in cadrul analizei jocului „Guess the Animal Sound”.

Un alt aspect negativ il constituie repetarea sunetului la infinit. Spre deosebire de jocul meu, unde sunetul se opreste dupa cateva secunde, aici, la inceputul fiecarei runde, sunetul este redat incontinuu, pana cand jucatorul alege animalul corect, fapt ce poate deveni foarte deranjant pentru jucator.

Un plus al concurentei il constituie varietatea de animale, jocul are 20 de animale, pe cand jocul meu are doar 8, iar “Guess the Animal Sound” 9, jocul meu contine exclusiv animale domestice, pe cand “Animal Sounds” are in componenta si animale salbatice, fapt ce aduce o valoare educationala mai mare, cei mici invatand mai multe sunete.

2.2 Aspecte legate de multimedia

Pentru realizarea acestui joc, fiecare animal a avut un sunet si o imagine asociata.

Pentru sunete s-au folosit fisiere WAV(Waveform Audio File Format) si MP3(MPEG Audio Layer 3).

Formatul MP3 foloseste compresia lossy. Prin comprimarea lossy a unui semnal se intelege eliminarea unora din datele initiale, neprelucrate. Cu cat mai multa informatie este ignorata, cu atat va scadea dimensiunea, procesul fiind unul ireversibil.

Fişierele WAV sunt fişiere audio necomprimate, dimensiunea lor fiind mai mare decat cele MP3, intrucat cele WAV contin datele initiale neprelucrate, asa cum au fost captate din sursa.

În ciuda diferențelor dintre cele 2 tipuri de fișiere, pentru optimizarea stocării datelor, în contextul aplicației mele, este aproape irelevant dacă lucrăm doar cu fișiere MP3, doar cu fișiere WAV sau dacă folosim ambele tipuri, deoarece lucrăm doar cu 8 fișiere audio, astfel ca dimensiunea lor nu face o diferență reală. Pentru realizarea jocului am utilizat, pentru fiecare animal, sunetul cel mai convenabil, făcând abstracție de tipul fișierului.

Pentru imagini s-au folosit fișiere jpeg, jpg (Joint Photographic Experts Group), PNG (Portable Network Graphics).

Jpeg și jpg diferă doar în denumire, nu în formatul efectiv al fișierului. Ambele sunt exact același tip de fișier (imagini comprimate folosind standardul „Joint Photographic Experts Group”). Așa cum este prezentat în [1], Jpeg este „o metodă populară de compresie de tip lossy (ireversibilă), utilizată pentru imagini. Compresia utilizată de format se bazează pe transformare cosinus discretă (DCT) și principalul său avantaj este dat de faptul că aduce un raport bun calitate imagine vs. dimensiune de stocare. Mai mult, utilizatorii pot să gestioneze acest raport în funcție de ceea ce își doresc (o calitate mai bună a imaginii sau salvarea de resurse de stocare). Astfel că, cu cât gradul de compresie este mai ridicat cu atât calitatea este mai slabă”.

Așa cum este prezentat în [2], Png „este un format grafic utilizat pentru stocarea, transmiterea și afișarea de imagini comprimate fără pierderi (lossless). Utilizat de obicei în domeniul networking, acest format oferă un bun compromis între calitate și dimensiune și este similar cu TIFF și GIF. Ce aduce în plus este faptul că permite utilizarea corecțiilor gamma și a unor spații standard de culori, ceea ce oferă posibilitatea unei reprezentări mult mai precise a imaginii. PNG este bun de folosit în aplicații de editare de imagine, iar spre deosebire de JPEG calitatea imaginii nu este degradată. De aceea el este mai recomandat să fie folosit în imagini ce conțin elemente grafice precum: text, linii sau contururi atent definite. PNG oferă suport atât pentru RGB, cât și pentru grayscale”.

La fel ca la sunete, datorită numărului mic de imagini folosite pentru joc (9, 8 pentru fiecare animal și una pentru imaginea de background a meniului principal), nu este foarte important să optimizăm dimensiunea de stocare a imaginilor. Am ales pentru fiecare animal și pentru poza de background cele mai potrivite imagini, ignorând în totalitate tipul fișierului, din motivele menționate mai sus.

Pentru dezvoltarea acestui joc am realizat mai multe operații folosind elementele multimedia menționate mai sus. Imaginile au fost încărcate, redimensionate (imaginea de background a fost redimensionată pentru a ocupa întreg ecranul, pozele cu animalele au fost redimensionate la dimensiunea de 200 x 200, dimensiune aleasă arbitrar pentru un design plăcut). Sunetele sunt redatate la începutul fiecărei runde.

3 Prezentare tehnica a etapei de realizare/implementare

Jocul a fost dezvoltat in Python, folosind bibliotecile pygame, os, random si time.

3.1 Biblioteci folosite

M-am folosit de biblioteca „pygame” pentru a realiza interfata grafica, a atribui sunetele si imaginile pentru fiecare animal si a gestiona evenimentele(iesire din joc, alegere animal, etc).

Pentru interfata grafica, am folosit urmatoarele functii, constructori si metode din biblioteca pygame:

a) „Font()” pentru a crea un obiect de tip font impreuna cu functia „render()” pentru a crea textul si „get_rect()” impreuna cu „blit()” pentru a afisa textul pe ecranul jucatorului la pozitia dorita.

b) „flip()” pentru a actualiza ecranul de fiecare data cand e nevoie(de exemplu cand utilizatorul intra in setari)

c) „scale()” pentru a redimensiona imaginea de fundal la dimensiunea ecranului si imaginile animalelor la dimensiunea de 200 x 200 pixeli (aleasa de mine arbitrar pentru un aspect placut al gameplay-ului).

d) „set_caption()” pentru a adauga captions relevante meniului principal, setarilor, statisticilor si jocului propriu-zis.

e) „Rect()” impreuna cu „rect()” pentru a crea si a desena pe ecran butoanele de - si +, precum si pentru afisarea numarului curent de runde, si butonul de inapoi.

f) „set_mode()” pentru a crea fereastra jocului

g) “set_cursor()” impreuna cu tipurile predefinite de cursor ale mouse-ului „SYSTEM_CURSOR_HAND”, „SYSTEM_CURSOR_ARROW” pentru a seta cursorul la mana cand dam hover peste butoanele din aplicatie(de exemplu butonul de „Start”).

Pentru atribuirea sunetelor si imaginilor corespunzatoare fiecarui animal am folosit functia „load()” din modulul pygame.image pentru a incarca imaginea si constructorul „Sound()” din pygame.mixer pentru a crea un obiect de tip Sound.

Pentru gestionarea evenimentelor am folosit functia „get()” din modulul pygame.event pentru a obtine toate actiunile realizate de jucator, apoi am implementat logica necesara gestionarii acestora. Evenimentele pe care le-am gestionat sunt: iesirea din joc, redarea sunetului produs de animal, alegere animal, apasarea diverselor butoane si hover butoane. Aceste evenimente au fost gestionate astfel:

Pentru iesirea din joc am implementat o variabila numita “running” care initial este True(atunci cand pornim jocul), iar daca utilizatorul iese din joc, atunci devine False.

Pentru redarea sunetului produs de animal am folosit variabila „sound_is_playing” care devine True cand incepem sa redam sunetul, iar daca iesim din joc sau sunetul a fost redat cu succes, devine False.

Pentru alegerea animalului am folosit variabila „waiting_choice” care este initial True(nu am ales niciun animal), cand apasam pe ecran folosim atributul „event.pos” din pygame pentru a returna pozitia unde am dat click, iar daca pozitia corespunde unei imaginii(am ales un animal), atunci waiting_choice devine False..

Pentru apasarea butoanelor am implementat aceeasi logica ca la alegere animal, calculam pozitia unde am dat click, si daca am dat click pe butonul respectiv se executa actiunea corespunzatoare(intram in setari, vizualizam statisticile, etc), iar pentru hover folosim variabila “hover” care este False atunci cand pozitia mouseului nu se afla pe vreun buton, devine True cand positionam mouse-ul pe un buton, schimbam cursorul de la arrow la hand, redevine False atunci cand mutam cursorul in exteriorul butonului.

M-am folosit de modulul os, precum si de functii implementate de mine, pentru a permite jocului sa ofere statistici jucatorilor. Pentru a face asta, am stocat notele obtinute intr-un document, ca ulterior, aplicatia sa aiba acces la ele. Pentru a face asta, am implementat functia „save_grade(grade)” unde salvam nota obtinuta de jucator intr-o variabila globala numita „grades”, apoi salvam nota intr-un fisier, unde stocam toate notele obtinute de jucator.

Apoi, am implementat functia „load_grade()”, care incarca notele din fisier in variabila globala grades, astfel facilitam accesul aplicatiei la toate notele obtinute vreodata de jucator, nu doar la cele obtinute in acea sesiune de joc(la inceputul fiecarei sesiuni de joc, variabila globala grades este goala). Aici folosim functia „exists()” din modulul os.path pentru a verifica daca fisierul exista. Alte functii implementate de mine care ajuta la vizualizarea statisticilor sunt „get_average_grade()” care ne ofera media tuturor notelor si „get_top3_grades()” care returneaza cele mai mari 3 note obtinute vreodata de jucator.

Am folosit modulul random pentru a alege la inceputul fiecarei runde animalul al carui sunet il vom reda si pentru ordonarea aleatorie a optiunilor de raspuns, astfel incat varianta corecta sa nu se afle mereu in aceeasi pozitie. Pentru a face asta, am folosit functiile predefinite

„choice()” pentru a alege un animal al carui sunet sa-l redam, „sample()” pentru a alege alte 2 animale pentru quiz, si „shuffle()” pentru a amesteca variantele de raspuns.

3.2 Contributia personala

Acum ca am explicat cum am folosit bibliotecile Python, voi completa cu contributia personala(functiile scrise de mine, logica aplicatiei si algoritmi folositi).

Pentru aceasta aplicatie, am implementat următoarele functii:

a) Functia „load_the_animals()” care incarca pentru fiecare animal imaginea si sunetul corespunzator. Pentru implementarea acesteia am folosit functia „load()” a modulului pygame.image pentru a incarca imaginea si constructorul „Sound” al modulului pygame.mixer pentru sunet.

b) Functii pentru gestionarea notelor, „save_the_grade(grade)”, „load_grades()”, „get_top3_grades()” si „get_average_grade()” care au fost explicate mai sus, cand am discutat modulul os.

c) Functii pentru gameplay, acestea sunt:

1) „listening_message(screen)” care afiseaza mesajul „Asculta sunetul!” pe ecran. O apelam la inceputul fiecărei runde, atunci cand redam un sunet.

2) „show_options(screen, options)” care ne afiseaza pe ecran animalele din care putem alege dupa ce am ascultat sunetul. Este apelata de fiecare data cand un sunet a fost redat cu succes, aceasta returneaza butoanele pe care putem da click pentru a alege un raspuns.

d) Functia „calculate_grade(score, total_rounds, response_times)” care calculeaza nota obtinuta de jucator. Aceasta este compusa din 2 parti(componenta timp si componenta corectitudine raspunsuri), fiecare reprezentand 50% din nota finala.

Componenta timp se calculeaza folosind formula

componenta_timp = max(0, 1 - max(0, (avg_time - threshold))), unde avg_time reprezinta timpul mediu de raspuns la intrebari(se calculeaza ca media aritmetica a tuturor timpilor de raspuns din cadrul fiecărei runde), threshold este o valoare predefinita data de nivelul de dificultate ales(pentru easy valoarea este 3, pentru medium 1 si pentru hard 0.5).

Componenta corectitudine raspunsuri este calculata folosind formula **componenta_corectitudine_raspunsuri = score / total_rounds**, unde score reprezinta totalul de raspunsuri corecte, iar total_rounds numarul total de runde.

Valorile celor 2 componente se afla in intervalul [0,1], fapt asigurat in componenta timp de conditiile de max, min, iar pentru nota finala se calculeaza astfel

nota_finala = 5 * componenta_corectitudine_raspunsuri + 5 * componenta_timp

nota_finala = min(10, max(1, nota_finala))

e) Functia `play_game(screen)` este functia jocului propriu-zis, gestioneaza toata logica din spatele gameplayului. Apeleaza functia `load_animals()` pentru a incarca toate animalele, preia numarul de runde ales de utilizator din setari, facand un loop care se opreste atunci cand am jucat toate rundele sau cand iesim din joc.

Pentru fiecare runda, alege un animal al carui sunet este redat(cu ajutorul functiei `choice()` din modulul `random`), amesteca variantele de raspuns(cu ajutorul functiei `shuffle()` din modulul `random`). La inceputul fiecarei runde apelam functia `listening_message(screen)`, apoi dam play unui sunet, dupa ce sunetul a fost redat apelam functia `show_options(screen, options)` si asteptam utilizatorul sa aleaga un raspuns. Dupa terminarea tuturor rundelor apelam functiile `calculate_grade(score, rounds, time_of_response)` si `save_grade(grade)` pentru a calcula si salva nota, urmand sa afisam pe ecran numarul de raspunsuri corecte, nota obtinuta si mesajul „Apasa orice tasta pentru a continua!”, cand apasam o tasta vom fi redirectionati in meniul principal.

f) Functia `„statistics(screen)”` care apeleaza functiile `„load_grades()”, „get_top3_grades”` si `„get_average_grade()”` si afiseaza pe ecran statisticile(cele mai mari 3 note si media notelor). Pentru a ne intoarce la meniu putem apasa orice buton.

g) Functia `„show_settings(screen)”` unde putem schimba din interfata grafica dificultatea si numarul de runde. Pentru acest lucru, am folosit diverse functii din biblioteca `pygame` pentru a desena butoanele si a detecta cand utilizatorul le-a apasat(aspect explicat mai sus, cand am discutat bibliotecile utilizate) si am implementat logica fiecarei apasari prin instructiuni conditionale(`if, else`). Astfel, atunci cand apasam butonul `easy, medium` sau `hard`, schimbam valoarea parametrului `settings[„difficulty”]` la valoarea echivalenta dificultatii, cand apasam `+` sau `-` incrementam/decrementam valoarea parametrului `settings[„rounds”]` atat timp cat valoarea se afla in intervalul `[1,10]`, iar atunci cand apasam butonul `„Inapoi”`, ne vom intoarce la meniul principal.

h) Functia `„main_menu()”` care este punct de intrare al aplicatiei, totodata reprezentand implementarea meniului principal. Aici afisam butoanele de `Start, Setari` si `Statistici` si asteptam ca utilizatorul sa apese pe unul cele 3 butoane, cand apasam pe `start` apelam functia `play_game(screen)`, cand apasam pe `Setari` apelam functia `show_settings(screen)`, iar cand apasam pe `statistici` apelam functia `statistics(screen)`.

Aspectele legate de AM tin de folosirea fisierelor foto si audio, redimensionarea si afisarea pozelor, redarea sunetelor.

4 Mod de utilizare

Modul de utilizare al aplicatiei este foarte simplu, intuitiv, usor de folosit pentru cei mici. Jocul te intampina cu un meniu principal, format dintr-o imagine si 3 butoane. Avem butonul „Start”, care va incepe jocul, butonul „Setari” unde putem personaliza gameplay-ul, alegand nivelul de dificultate si numarul de runde pe care vrem sa-l jucam, si butonul „Statistici” unde putem vedea statisticile jucatorului.

Mai jos, voi atasa o imagine cu meniul principal.

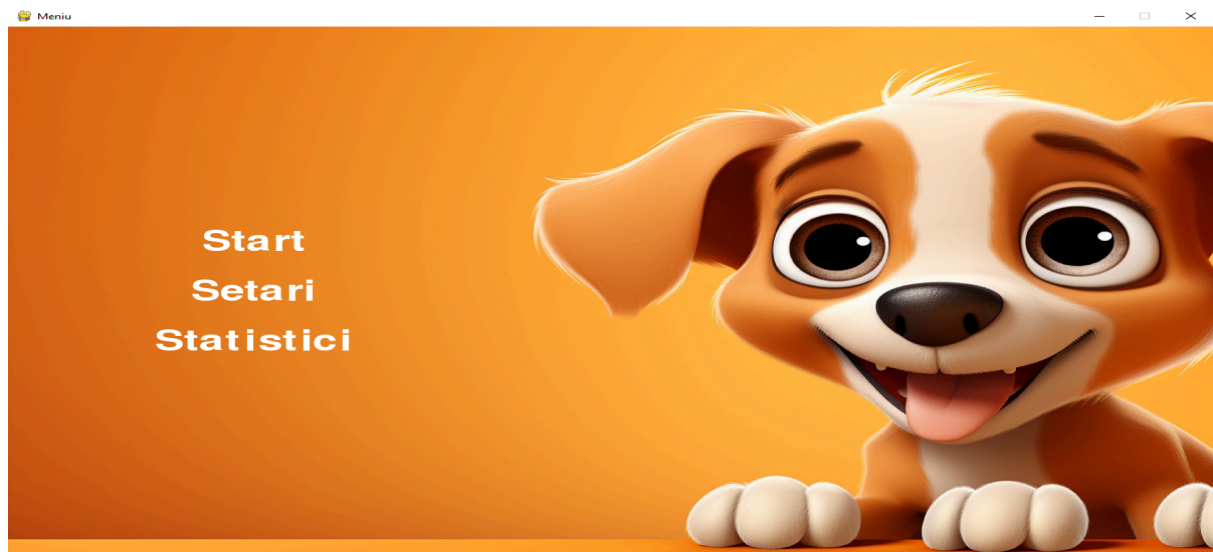


Figura 1: Meniul principal al jocului

Interfata vizuala a gameplayului este formata din 3 cadre. Primul cadru afiseaza mesajul „Asculta sunetul!” atunci cand redam sunetul animalului. Al doilea cadru ne prezinta variantele de raspuns. Al treilea cadru apare la finalul jocului, unde putem vizualiza numarul de raspunsuri corecte si nota obtinuta.

Joc de Recunoastere a Sunetelor

Asculta sunetul!

Figura 2: Primul cadru al gameplay-ului

Alege animalul corect!



Figura 3: Al doilea cadru al gameplay-ului

Raspunsuri corecte: 5/5
Nota obtinuta: 10

Apasa orice tasta pentru a continua...

Figura 4: Al treilea cadru al gameplay-ului

Interfata vizuala pentru setari are un aspect placut, utilizatorul poate alege numarul de runde si dificultatea jocului, apasand butonul corespunzator preferintei acestuia. Deasemenea, avem un buton care ne redirectioneaza la meniul principal.

Setari

Dificultate:

Easy

Medium

Hard

Numar runde:

-

5

+

Inapoi

Figura 5: Interfata vizuala pentru setari

Interfata vizuala pentru statistici este simpla, avem un fundal alb pe care scriem cu negru statisticile jucatorului.

🐼 Statistici

— □ ×

Statistici

Cele mai bune 3 note:

1. 10.0

2. 10.0

3. 10.0

Media notelor: 7.82

Apasă orice tastă pentru a reveni

Figura 6: Interfata vizuala pentru vizualizarea statisticilor

5 Concluzii

Proiectul si-a atins cu succes obiectivele propuse, oferind un joc interactiv si captivant pentru cei mici, aducand totodata si o plus valoare educationala. Prin intermediul acestei aplicatii, copiii pot invata sa distinga sunetele produse de animale.

Jocul raspunde nevoii actuale de creare de continut educativ si calitativ, care sa reuseasca sa mentina interesul celor mici. Pentru a transforma un joc banal intr-o experienta captivanta, am adaugat nivele de dificultate si statistici, pentru a oferi jucatorilor atat o miza, cat si o modalitate de a-si urmari progresul.

Jocul a fost realizat folosind tehnologii moderne, precum biblioteca pygame, integrand diverse formate de fisiere audio(WAV, MAP3) si imagini (JPEG, PNG).

Proiectul meu prezinta avantaje fata de alte jocuri similare, precum personalizarea gameplay-ului, feedback detaliat, si interfata intuitiva.

In concluzie, acest proiect nu doar ca si-a indeplinit obiectivul propus, dar a adus si inovatii in domeniu. Prin combinarea elementelor multimedia cu o abordare moderna, acesta reuseste sa fie atat distractiv, cat si educativ, demonstrand potentialul de a fi un instrument eficient in educatia copiilor.

Bibliografie:

- [1] Laboratorul 4 de Aplicatii Multimedia, pus la dispozitie pe Moodle studentilor de la Facultatea de Automatica si Calculatoare, din cadrul Universitatii Nationale de Stiinta si Tehnologie POLITEHNICA Bucuresti

- [2] Laboratorul 5 de Aplicatii Multimedia, pus la dispozitie pe Moodle studentilor de la Facultatea de Automatica si Calculatoare, din cadrul Universitatii Nationale de Stiinta si Tehnologie POLITEHNICA Bucuresti

- [3] Sweigart, A. (2012). Making Games with Python & Pygame. (online) Disponibila: <https://inventwithpython.com/makinggames.pdf> (Accesat 5 Mai 2025)

- [4] EducaGaming. Guess the Animal Sound (Accesat 5 Mai 2025). Web-site <https://www.educagaming.com/en/gameDetail/guess-the-animal-sound-game>

- [5] Nau.kids. Animal Sounds (Accesat 5 Mai 2025). Web-site <https://crazygames-poki.com/puzzle/animal-sounds>