

Snake Game

Autor : Iordăchiță Sergiu-Andrei, 333AA

An : 3

Cuprins

1. Introducerea si prezentarea temei	3
Istoric	3
Obiectivele acestui proiect	4
2. Prezentarea suportului tehnic	5
3. Prezentarea tehnică a etapei de implementare	5
4. Prezentare mod de utilizare	9
5. Concluzii	11
6. Referințe bibliografice	12

1. Introducerea si prezentarea temei

Ca temă de proiect am ales să realizez un joc Snake, pe care l-am implementat în programul Microsoft Visual Studio folosindu-mă de limbajul Python. Am ales acest joc din prisma nostalgiei, deoarece a fost primul joc pe care l-am jucat vreodată pe un telefon mobil, fiind totodată curios de complexitatea implementării unui astfel de joc pe un device atât de vechi precum telefonul Nokia. Snake este un sub-gen de jocuri video de acțiune în care jucătorul manevrează capătul unei linii în creștere, adesea tematică ca un șarpe. Jucătorul trebuie să împiedice șarpele să se ciocnească atât de alte obstacole, cât și de el însuși, ceea ce devine mai greu pe măsură ce șarpele se prelungește.



Istoric

Snake a început cu jocul video arcade din 1976 Blockade, dezvoltat și publicat de Gremlin. A fost clonat ca Bigfoot Bonkers în același an. În 1977, Atari, Inc. a lansat două titluri inspirate de Blockade: jocul arcade Dominos și jocul Atari VCS Surround. Surround a fost unul dintre cele nouă titluri de lansare Atari VCS în SUA și a fost vândut de Sears sub numele de Chase. În același an, un joc similar a fost lansat pentru Bally Astrocade sub numele de Checkmate. Prima versiune cunoscută de computer de acasă, care se numește Worm, a fost programată în 1978 de Peter Trefonas pentru TRS-80 și publicată de revista CLOAD în același an. Aceasta a fost urmată la scurt timp după aceea cu versiuni ale aceluiași autor pentru Commodore PET și Apple II. O clonă a jocului arcade Hustle, ea însăși o clonă a Blockadei, a fost scrisă de Peter Trefonas în 1979 și publicată de CLOAD. O versiune autorizată a Hustle a fost publicată de Milton Bradley pentru TI-99/4A în 1980. Snake Byte pentru un singur jucător a fost publicat în 1982 pentru computerele Atari pe 8 biți, Apple II și VIC-20; un șarpe mănâncă mere pentru a finaliza un nivel, crescând mai mult în acest proces. În Snake pentru BBC Micro (1982), de Dave Bresnen, șarpele este controlat folosind tastele săgeți stânga și dreapta în raport cu direcția în care se îndreaptă. Șarpele crește în viteză pe măsură ce devine mai lung și există o singură viață. Nibbles (1982) este un joc arcade pentru un singur jucător în care șarpele se potrivește bine într-un labirint, iar jocul este mai rapid decât majoritatea modelelor de Snake.

O altă versiune single-player face parte din jocul arcade Tron din 1982, modernizat cu cicluri de lumină. A revigorat conceptul de șarpe și multe jocuri ulterioare au împrumutat tema ciclului luminii. Începând din 1991, Nibbles a fost inclus cu MS-DOS pentru o perioadă de timp ca un program de probă QBasic. În 1992, Rattler Race a fost lansat ca parte a celui de-al doilea Microsoft Entertainment Pack. Acesta adaugă șerpi inamici la jocul familiar care mănâncă mere.

Nokia a pus Snake pe majoritatea telefoanelor lor mobile :

- Snake – Prima publicata de Nokia, pentru telefoanele monocrome. A fost programat in 1997 de Taneli Armanto de la Nokia si introdus pe Nokia 6110
- Snake II – Inclus pe telefoanele monocrome, cum ar fi Nokia 3310 din 2000
- Snake Xenezia – Inclus pe modelele ulterioare de telefoane monocrome
- Snake EX – Inclus pe telefoanele color. Introdus pentru prima data cu Nokia 9290 in 2002. Accepta multiplayer prin Bluetooth si infraroșu
- Snake EX2 – Introdus împreună cu Nokia 3100 in 2003 si inclus in mai multe telefoane din seria 40
- Snakes - O versiune 3D concepută pentru N-Gage în 2005. Dispunea de multiplayer prin Bluetooth. Mai târziu, Nokia a început să-l preinstaleze (fără multiplayer) pe unele smartphone-uri Nseries precum N70, N73, N80 etc. Poate fi descărcat de pe paginile de asistență Nokia și poate fi redat pe orice dispozitiv S60
- Snake III - O versiune 3D, diferită de Snakes. Snake III adoptă o abordare mai degrabă a șarpelui viu, mai degrabă decât sentimentul abstract al Snakes. Un exemplu de telefon cu acesta instalat este Nokia 3250 din 2005 și acceptă moduri multiplayer prin Bluetooth.
- Snakes Subsonic – Lansat pe 22 mai 2008 pentru N-Gage 2.0
- Snake Xenezia(2017) – Lansat pentru prima data pe Nokia 3310
- Snake(2017) – Lansat cu Facebook Messenger (2017)

Obiectivele acestui proiect

- Implementarea jocului clasic de Snake
- Adăugarea de elemente sonore si vizuale
- Îmbunătățirea game play-ului prin adăugarea unor noi elemente in joc
- Dezvoltarea cunoștințelor de lucru utilizând limbajul Python

2. Prezentarea suportului tehnic

Într-o serie impresionată de jocuri create folosind biblioteca Pygame se enumeră: Gleamshroom, Hue Flowing, Explon't, Unnamed MMORPG, BlightDrawn, Vaegrant, 3D Model Renderer, Live 3D Skeletal Model, Shadow Raycaster, 3D Earth & Stars (aceasta fiind create în anul 2022). Pentru vizualizarea acestora, voi lăsa un link de YouTube al prezentării lor în secțiunea de Referințe.

În implementarea jocului meu de Snake, ideea adăugării aceluși măr special a fost inspirată de jocul Surf, care este specific browser-ului Microsoft Edge în momentul în care utilizatorul nu are o conexiune la internet. În acel joc, caracterul poate aduna de pe mape fulgere care să îi mărească viteza pentru o scurtă perioadă de timp.

3. Prezentarea tehnica a etapei de implementare

În cadrul implementării jocului m-am folosit de o bibliotecă a limbajului Python numita Pygame, cu care am realizat ecranul de joc împreună cu fundalul, am adăugat sunete, am customizat informații din cadrul jocului (cum ar fi afișarea scorului sau a unui mesaj la finalul jocului) și am adăugat diferite imagini pentru realizarea game play-ului.

Ce este Pygame ?

Pygame este o bibliotecă open-source specializată în dezvoltarea de jocuri și aplicații multimedia folosind limbajul de programare Python. Ea oferă funcționalități pentru a gestiona grafica, sunetul, interacțiunea cu utilizatorul și alte aspecte legate de dezvoltarea de jocuri.

Prin intermediul Pygame, puteți crea și controla ferestre de afișare, desena grafică 2D și 3D, manipula sprite-uri, detecta evenimente de la tastatură și mouse, reda și controla sunetul, implementa fizica jocului și multe altele. Pygame se bazează pe biblioteca SDL (Simple DirectMedia Layer), care oferă acces la funcționalitățile hardware ale dispozitivului, cum ar fi afișajul și sunetul. Alte caracteristici pe care SDL le ar include : matematica vectorială, detectarea coliziunilor, gestionarea graficelor scenei sprite 2D, suport MIDI, cameră foto, manipulare a matricei de pixeli, transformări, filtrare, suport avansat pentru fonturi freetype și desen. Aplicațiile care utilizează Pygame pot rula pe telefoane și tablete Android cu utilizarea Pygame Subset pentru Android. Sunetul, vibrațiile, tastatura și accelerometrul sunt acceptate pe Android.

Biblioteca Pygame este folosită în principal pentru a crea jocuri, dar poate fi utilizată și pentru a dezvolta aplicații multimedia interactive sau simulări grafice. Este ușor de învățat și utilizat, făcându-l o opțiune populară pentru dezvoltarea de jocuri și experimente în programarea vizuală.

Datorită facilităților sale, Pygame a devenit o opțiune populară pentru cei care doresc să creeze jocuri și aplicații simple în Python, în special pentru începători și programatori cu experiență limitată în domeniul jocurilor video.

Ulterior m-am folosit și de alte biblioteci precum : time (pentru a putea folosi funcția `time.sleep(s)` ce are rolul de a suspenda execuția programului pentru o anumită perioadă de timp), random (pentru a putea folosi funcția `random.randint(a, b)` ce are rolul de a genera un număr întreg aleatoriu din intervalul `[a, b]`) și threading (pentru a putea începe execuția unei funcții în paralel cu restul programului).

Metodologia

Pentru a crea jocul am folosit principii de programare orientată pe obiect, prin definirea caracterului din joc sub forma de clase și implementând metode specifice pentru fiecare.

3.1. Scriptul pentru măr

Metoda `__init__(self, parent_screen)`: Este metoda de inițializare a clasei. Prin această metodă se configurează starea inițială a unui obiect de tip măr. Are următoarele funcționalități:

- Inițializează lista denumire cu numele imaginilor pentru măr (`apple.jpg`, `apple2.jpg`).
- Inițializează atributul `index` cu valoarea 0, reprezentând indexul imaginii de măr curentă.
- Încarcă imaginea corespunzătoare indexului în atributul `image` folosind `pygame.image.load()`.
- Stabilește ecranul părinte în atributul `parent_screen`.
- Inițializează poziția inițială a mărului la coordonatele `(SIZE*3, SIZE*3)`.

Metoda `draw(self)`: Este responsabilă de afișarea mărului pe ecran. Are următoarele funcționalități:

- Încarcă imaginea corespunzătoare indexului în atributul `image` folosind `pygame.image.load()`.
- Desenează imaginea mărului pe ecranul părinte la coordonatele `(self.x, self.y)` utilizând `self.parent_screen.blit()`.
- Actualizează ecranul cu metoda `pygame.display.flip()`.

Metoda `move(self, snake_x, snake_y)`: Este responsabilă de mutarea mărului la o poziție nouă. Are următoarele funcționalități:

- Generează un nou index pentru a alege o imagine de măr aleatorie utilizând `random.randint()`.
- Generează noi coordonate pentru măr utilizând `random.randint()`.
- Verifică dacă poziția mărului se suprapune cu pozițiile elementelor șarpelui. Dacă da, se generează noi coordonate până când mărul nu se suprapune cu șarpele.

- Actualizează coordonatele mărului cu noile valori calculate.

Aceste metode permit desenarea, mutarea și gestionarea mărului în jocul Snake. Prin intermediul lor, se poate afișa și muta mărul pe ecran, asigurându-se că nu se suprapune cu șarpele și că se utilizează imagini diferite pentru mărul afișat.

3.2. Scriptul pentru piatra (ce reprezintă un obstacol în joc)

Este identic cu cel pentru mar, singur diferența fiind faptul că nu mai apare atributul `index` având un singur tip de piatra cu o funcționalitate fixă.

3.3. Scriptul pentru șarpe (ce reprezintă caracterul nostru din joc)

Metoda `__init__(self, parent_screen, length)`: Este metoda de inițializare a clasei. Prin această metodă se configurează starea inițială a unui obiect de tip șarpe. Are următoarele funcționalități:

- Inițializează atributul `length` cu lungimea șarpelui specificată.
- Stabilește ecranul părinte în atributul `parent_screen`.
- Încarcă imaginea unui bloc care reprezintă segmentul șarpelui în atributul `block` folosind `pygame.image.load()`.
- Inițializează coordonatele inițiale ale segmentelor șarpelui `x` și `y` ca fiind înmulțite cu `SIZE` și salvează în listele corespunzătoare.
- Stabilește direcția inițială a șarpelui la 'down' și salvează-o în atributul `direction`.
- Stabilește direcția precedentă a șarpelui la 'down' și salvează-o în atributul `previous_direction`.

Metoda `increase_length(self)`: Este responsabilă de creșterea lungimii șarpelui atunci când acesta consumă un măr. Are următoarele funcționalități:

- Incrementează valoarea atributului `length` cu 1.
- Adaugă o nouă valoare -1 în listele `x` și `y` pentru a reprezenta un nou segment al șarpelui.

Metoda `draw(self)`: Este responsabilă de afișarea șarpelui pe ecran. Are următoarele funcționalități:

- Parcurge fiecare segment al șarpelui și desenează blocul corespunzător pe ecranul părinte la coordonatele `(self.x[i], self.y[i])` utilizând `self.parent_screen.blit()`.
- Actualizează ecranul cu metoda `pygame.display.flip()`.

Metodele `move_left(self)`, `move_right(self)`, `move_up(self)`, `move_down(self)`: Sunt responsabile de schimbarea direcției de deplasare a șarpelui. Au următoarele funcționalități:

- Verifică direcția precedentă a șarpelui pentru a preveni mișcarea în direcții opuse.
- Actualizează direcția șarpelui și direcția precedentă în funcție de direcția specificată.

Metoda `walk(self)`: Este responsabilă de mutarea șarpelui în funcție de direcția curentă. Are următoarele funcționalități:

- Parcurge lista de segmente ale șarpelui de la coadă la cap și actualizează coordonatele segmentelor

3.4. Scriptul pentru joc

Metoda `__init__(self)`: Este metoda de inițializare a clasei. Prin această metodă se configurează starea inițială a jocului. Are următoarele funcționalități:

- Inițializează pygame și mixerul pygame.
- Redă muzica de fundal folosind metoda `play_background_music()`.
- Inițializează variabila `n_apple` cu valoarea `True`.
- Creează suprafața de joc utilizând `pygame.display.set_mode()`.
- Inițializează obiectul snake de tip `Snake` și îl afișează pe ecran folosind metoda `draw()`.
- Inițializează obiectul apple de tip `Apple` și îl afișează pe ecran folosind metoda `draw()`.
- Inițializează obiectul stone de tip `Stone` și îl afișează pe ecran folosind metoda `draw()`. Lansează un fir de execuție (`threading.Thread`) pentru a muta piatra în mod continuu utilizând metoda `move_stone_thread()`.

Metoda `is_collision(self, x1, y1, x2, y2)`: Verifică dacă două obiecte au coliziune, având coordonatele lor drept argumente. Returnează `True` dacă au coliziune și `False` în caz contrar.

Metoda `move_stone_thread(self)`: Este responsabilă de mutarea pietrei în mod continuu. Are următoarele funcționalități:

- Într-un buclă infinită, se așteaptă pentru o perioadă de timp de 10 secunde folosind `time.sleep()`.
- Apoi, folosește metoda `move()` a obiectului stone pentru a muta piatra în funcție de poziția șarpelui.

Metoda `play_background_music(self)`: Redă muzica de fundal a jocului.

Metoda `play_sound(self, sound)`: Redă un sunet specificat.

Metoda `render_background(self)`: Afișează fundalul pe suprafața de joc.

Metoda `play(self)`: Realizează logica de joc. Are următoarele funcționalități:

- Afișează fundalul utilizând metoda `render_background()`.
- Mută șarpele utilizând metoda `walk()` a obiectului snake.
- Afișează mărul și piatra utilizând metodele `draw()` corespunzătoare.
- Afișează scorul utilizând metoda `display_score()`.
- Actualizează ecranul cu `pygame.display.flip()`.
- Verifică coliziunile șarpelui cu mărul, piatra și cu el însuși, și lansează excepții corespunzătoare în caz de coliziune.

Metoda `display_score(self)`: Afișează scorul pe ecran.

Metoda `show_game_over(self)`: Afișează mesajul de final de joc și oprește temporar muzica de fundal.

Metoda `reset(self)`: Resetează starea jocului prin reinstantierea obiectelor snake și apple.

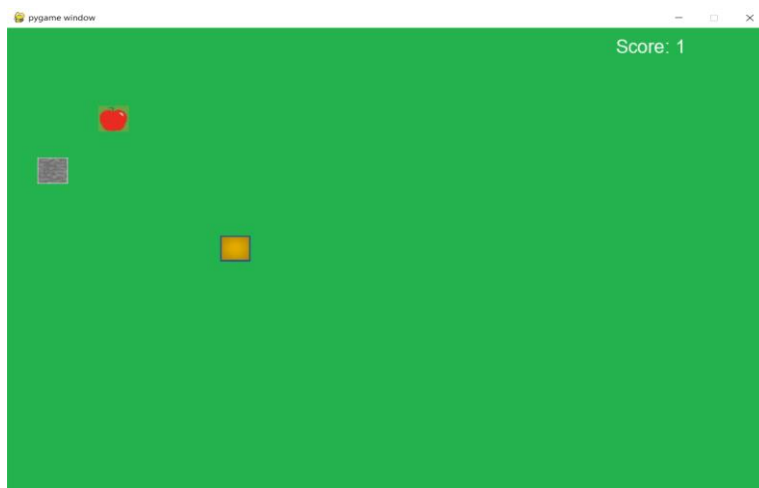
Metoda `run(self)`: Execută bucla principală a jocului. Are următoarele funcționalități:

- Gestionează evenimentele pygame și răspunde la tastele apăsate.
- Dacă jocul nu este în pauză, se apelează metoda `play()` pentru a realiza logica de joc.
- În caz de excepție (coliziune sau game over), se afișează mesajul de final de joc, se intră în pauză și se resetează starea jocului.
- Se așteaptă pentru o perioadă scurtă de timp în funcție de valoarea `n_apple`.

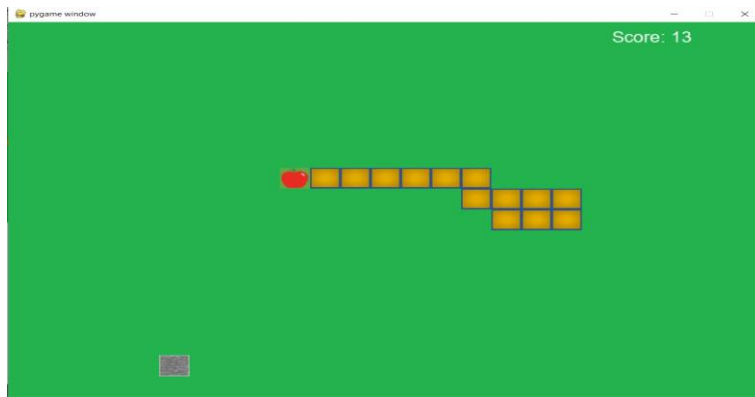
4. Prezentare mod de utilizare

La execuția programului se va deschide fereastra în care rulează jocul, iar în fereastra vor apărea următoarele 3 elemente : șarpele (care la început are dimensiunea 1 și se deplasează înspre partea de jos a ecranului – setat default), mărul (care poate fi roșu, un mar simplu, sau auriu ce reprezintă amarul special din joc) și piatra (ce reprezintă un obstacol în joc, schimbându-și mereu poziția la fiecare 10 secunde, aceasta poate duce la finalul jocului dacă șarpele se ciocnește de ea).

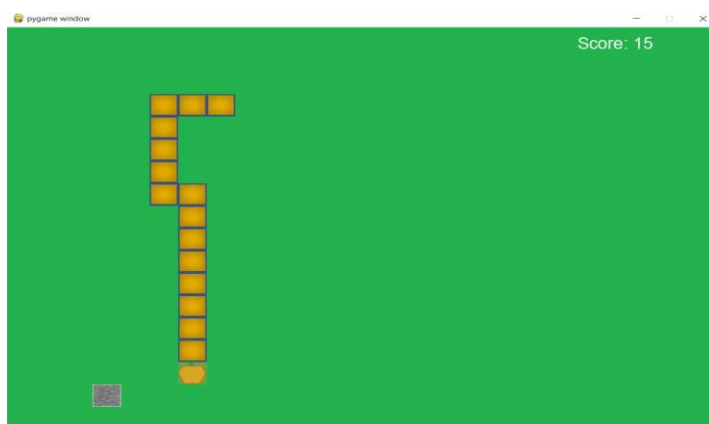
În timpul jocului pentru a schimba direcția șarpelui, utilizatorul poate folosi săgețile tastaturii, iar pentru a închide jocul trebuie apăsata tasta ESCAPE. În colțul din stânga de sus al ecranului va apărea scorul jucătorului.



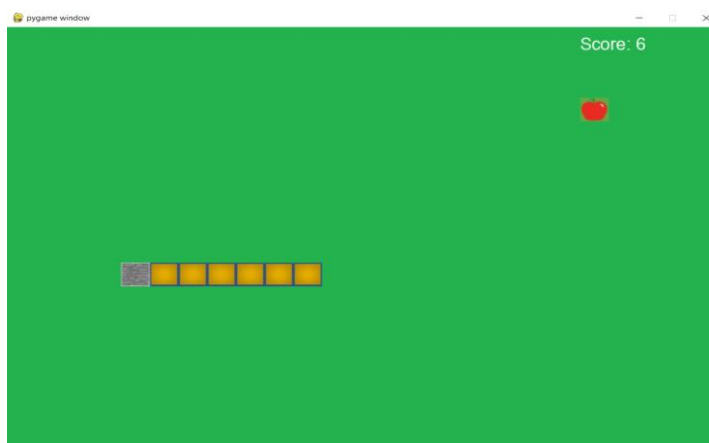
Momentul în care șarpele mănâncă un mar roșu (normal) aceasta se deplasează la o viteză mai lentă și își mărește dimensiunea cu o unitate.



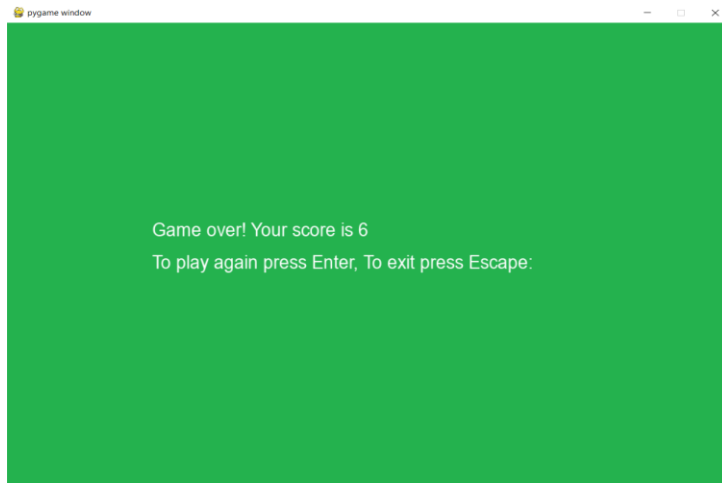
Momentul in care șarpele mănâncă un mar auriu (special) aceasta se deplasează la o viteză mai mare(de 10 ori fata de cea normala) si își mărește dimensiunea cu o unitate.



In timpul derulării jocului, pe fundal va rula o melodie ce se va opri doar atunci când șarpele se lovește de un obstacol. In momentul in care șarpele mănâncă un mar se va auzi un sunet specific, iar in momentul in care șarpele se va lovi de un obstacol (una din marginile ecranului de joc, piatra sau de el însăși) se va auzi de asemenea un sunet specific si jocul se va opri.



După ce jocul se oprește pe ecran va apărea mesajul "Game over!", urmat de scorul înregistrat de jucător. Din acesta moment utilizatorul are 2 variante : fie sa înceapă jocul de la capăt apăsând tasta Enter, fie sa iasă de tot din joc apăsând tasta Escape.



5. Concluzii

- a) Pentru realizarea primului obiectiv de a recrea jocul clasic de snake, am exemplificat în cadrul părții tehnice a etapei de implementare (având inițial doar cele 3 clase : Apple, Snake și Game).
- b) Pentru adăugarea elementelor sonore (cele 3 melodii care se aud în timpul jocului) și vizuale (imaginile merelor, a șarpelui, a pietrei, a fundalului jocului, a tablei de scor și mesajelor din finalul jocului) am folosit biblioteca Pygame specifică limbajului Python în cadrul implementării jocurilor.
- c) Am realizat îmbunătățirea game play-ului prin adăugarea următoarelor noi elemente : a celui de-al doilea mar (auriu) și a noului obstacol (piatra) ce își au descrierea de implementare în cadrul părții tehnice. Un rol important în realizarea unei experiențe plăcute a utilizatorului cu jocul îl au și elementele vizuale și sonore prezentate anterior.
- d) În urma realizării acestui proiect am câpătat o serie de cunoștințe referitoare la biblioteca Pygame și ce poate aduce în implementarea unui joc, m-am familiarizat și mai mult cu limbajul Python și mi-am reutilizat cunoștințele asupra programării orientate pe obiect.

6. Referințe bibliografice

1. Istoric joc snake : [https://en.wikipedia.org/wiki/Snake_\(video_game_genre\)](https://en.wikipedia.org/wiki/Snake_(video_game_genre))
2. Informații Pygame : <https://en.wikipedia.org/wiki/Pygame>
3. Funcții Pygame : <https://www.pygame.org/docs/>
4. Proveniența sunetelor : <https://www.freesoundlibrary.com/>
5. 2022 Pygame Games : <https://www.youtube.com/watch?v=xVHkF1-73pw>