



Tudor-Sebastian Robu

Clasa a XII-a A

Profesori îndrumători Doru Modrișan și Irina Aupici

Colegiul Național „Andrei Șaguna”

Mai 2023

Cuprins

1.Introducere

2.Scurt Istoric al Apariției Internetului

3.Limbajul de Programare Utilizat

4.Prezentarea Programului

5.Explicarea Codului Utilizat

6.Bibliografie

1.Introducere

Deși obișnuiam de mic să mă joc cât mai des cu putință pe orice dispozitiv pe care îl aveam la îndemână, fie el telefon, tabletă sau calculator, doar un singur joc mă duce cu gândul la momentul când am fost pentru prima oară introdus jocurilor video: *Flappy Bird*.

Acesta era un joc simplu în esență și consta în coordonarea unei păsări prin deschizătura a două conducte:



Astfel, acest joc a fost la baza inspirației mele. Pentru lucrarea mea de atestat, mi-am creat propriul joc „Flappy Bird”, totuși diferit față de cel original.



Poză din jocul codat de mine.

2.Scurt Istoric al Apariției Internetului

Istoria Internetului începe cu anul 1968, când guvernul S.U.A. intenționa să interconecteze universitățile, departamentele militare și de apărare ale țării, astfel încât ele să coopereze în cadrul unor proiecte de cercetare comune. Astfel, s-a format o agenție numită Advanced Research Projects Agency (ARPA). Una din cheile proiectului punea în discuție faptul că, stocarea tuturor informațiilor pe un singur calculator nu ar fi fost deloc sigură, fie din cauză că acesta ar putea fi țintă vulnerabilă a unui eventual atac, fie pur și simplu din cauză că acestea ar putea fi pierdute în cazul unei defecțiuni tehnice majore. O metodă de a face față unei asemenea situații ar fi de a copia și distribui informațiile pe mai multe calculatoare, în întreaga țară, folosind o rețea.

În 1975, câteva dintre limbajele sau protocoalele pe care calculatoarele le foloseau pentru a comunica între ele s-au standardizat. Majoritatea universităților importante și a departamentelor de apărare din S.U.A. s-au legat împreună într-o rețea numită DARPA NET, toate calculatoarele folosind același protocol pe care astăzi îl cunoaștem sub denumirea de TCP/IP. Rețeaua, cu timpul, a fost înlocuită de mai multe rețele, care astăzi împânzesc globul pământesc.

Începând cu anul 1980, mai multe colegii și universități au fost conectate la Internet. Acest lucru a permis universităților să-și împartă informații despre cercetările lor, programe și știri recente. În anii '90 Internetul s-a deschis și în scopuri comerciale. În curând, multe alte căi de utilizare a informațiilor transmise prin intermediul acestei gigantice rețele au fost dezvoltate.

În prezent, este posibil să folosești Internetul pentru a trimite scrisori electronice pe întregul glob în doar câteva secunde. Poți căuta informații despre orice subiect dorești. Expresia „World Wide Web” (WWW) definește o colecție de documente care se întinde în câteva sute de milioane de calculatoare.[6]

3.Limbajul de Programare Utilizat

Codul jocului a fost scris exclusiv în *Python*.



Python este un limbaj de programare de înalt nivel, creat în 1989 de programatorul olandez Guido van Rossum, dar totuși simplu și ușor de înțeles, care se concentrează pe abstracții și structuri de date. Este un limbaj de programare popular, utilizat în diverse domenii precum dezvoltarea web, știința datelor, inteligența artificială și, de asemenea, dezvoltarea de jocuri video.

Python pune accentul pe curățenia și simplitatea codului, iar sintaxa sa le permite dezvoltatorilor să exprime unele idei programatice într-o manieră mai clară și mai concisă decât în alte limbaje de programare ca C. În ceea ce privește paradigma de programare, Python poate servi ca limbaj pentru software de tipul object-oriented, dar permite și programarea imperativă, funcțională sau procedurală. Sistemul de tipizare este dinamic iar administrarea memoriei decurge automat prin intermediul unui serviciu „gunoier” (garbage collector). Alt avantaj al limbajului este existența unei ample biblioteci standard de metode.[5]

4.Prezentarea Programului

Jocul prezintă un avion negru, care contrastează cu fundalul luminos, care, din cauza forței gravitaționale este într-o cădere constantă. Jucătorul, de fiecare dată când apasă tasta „SPACE” ridică puțin avionul, pe care trebuie să-l mențină în aer pentru cât mai mult timp. În același timp, din direcția opusă se generează la întâmplare „nori”, pe care aeronava trebuie să-i evite.

De fiecare dată când avionul zboară ori prea sus, ori prea jos sau trece printr-un nor, jocul se oprește, afișând un mesaj cu instrucțiunile următoare:



Programarea din spatele programului are 114 linii de cod, și a fost construit pentru a fi cât mai eficient cu putință. Jocul are o limită de 120 FPS, norii se generează la întâmplare la după aproximativ 6 secunde unul de celălalt.

De asemenea, cât timp mesajul cu instrucțiuni de mai sus este afișat, codul oprește generarea norilor.

5.Explicarea Codului Utilizat

Librăriile utilizate:

```
import pygame
```

```
import sys
```

```
import random
```

Construirea subsolului paginii

```
def game_floor():
```

```
    screen.blit(floor_base, (floor_x_pos, 0))
```

```
    screen.blit(floor_base, (floor_x_pos + 1790, 0))
```

În jurul fiecărui nor generat și al avionului este declarat un dreptunghi. Dacă aceste dreptunghiuri se suprapun, programul va detecta ciocnirea dintre nor și avion și va returna valoarea “False”.

```
def check_collision(clouds):
```

```
    for cloud in clouds:
```

```
        if plane_rect.colliderect(cloud):
```

```
            return False
```

```
    if plane_rect.bottom >= 900 or plane_rect.top <= -200:
```

```
        return False
```

```
    return True
```

Norii sunt toți identici și sunt generați după partea dreaptă a ecranului, la înălțimi diferite.

```
#nori
```

```
def create_cloud():
```

```
    random_cloud_pos = random.choice(cloud_height)
```

```
    main_cloud = cloud_surface.get_rect(midbottom=(2000, random_cloud_pos))
```

Atestat la Informatică „Flapy Bird”

```
    return main_cloud  
def move_clouds(clouds):  
    for cloud in clouds:  
        cloud.centerx -= 1  
    return clouds
```

Fiecare nor va fi afișat cu imaginea atribuită.

```
def draw_clouds(clouds):  
    for cloud in clouds:  
        screen.blit(cloud_surface,cloud)
```

```
pygame.init()
```

Se definește o viteză cu care avionul să coboare vertical.

```
clock=pygame.time.Clock()  
#variabile  
gravity=0.02  
plane_movement=0
```

Se setează fundalul jocului, care are o lungime de 1800 și o înălțime de 1000.

```
screen=pygame.display.set_mode((1800,1000))  
background=pygame.image.load("Background.jpg").convert()  
background = pygame.transform.smoothscale(background, screen.get_size())
```


Avionului i se atribuie imaginea cu care va fi afișat în joc. Aceasta, având rezoluția prea mare, trebuie micșorată. De asemenea, tot aici, se definesc coordonatele ca care să apare avionul (coordonata pe axa Ox rămâne neschimbată).

```
plane=pygame.image.load("avion negru.png").convert_alpha()
plane=pygame.transform.scale(plane, (300, 100))
plane_rect=plane.get_rect(center=(500,10))
```

Se declară coordonata Oy imaginii de subsol.

```
floor_base=pygame.image.load("floor.png").convert_alpha()
floor_x_pos=0
```

În cazul unei ciocniri, se va afișa o imagine selectată în mijlocul interfeței.

```
message=pygame.image.load("Retry.png").convert_alpha()
game_over_rect=message.get_rect(center=(900,500))
```

Generarea și stocarea informației pentru fiecare nor.

```
#generare Pt nori Individuali
cloud_surface=pygame.image.load("1 cloud.png").convert_alpha()
#cloud_surface=pygame.transform.scale(cloud_surface, (100, 100))
cloud_list=[]
SPAWN_CLOUD=pygame.USEREVENT
pygame.time.set_timer(SPAWN_CLOUD, 3600)
cloud_height=[400, 500, 600, 700, 800, 300, 200, 100 , 0]
```

```
game_active=True
```

while True:

```
    for event in pygame.event.get():  
        if event.type==pygame.QUIT:  
            pygame.quit()  
            sys.exit()
```

Când se apasă tasta *SPACE* variabila de mișcare a avionului pe axa Oy se schimbă.

```
    if event.type==pygame.KEYDOWN:  
        if event.key==pygame.K_SPACE and game_active:  
            plane_movement=0  
            plane_movement-=2  
        if event.key==pygame.K_SPACE and game_active==False:  
            plane_rect.center=(300,500)  
            plane_movement=0  
            game_active=True  
    if event.type==SPAWN_CLOUD and game_active:  
        cloud_list.append(create_cloud())
```

Se declară coordonatele imaginii de declar și viteza avionului.

```
    screen.blit(background,(0,0))  
    if game_active:  
        plane_movement+=gravity  
        plane_rect.centery+=plane_movement  
        screen.blit(plane, plane_rect)  
  
    cloud_list=move_clouds(cloud_list)  
    draw_clouds(cloud_list)
```

```
    game_active=check_collision(cloud_list)
else:
    screen.blit(message, game_over_rect)
    plane_movement+=gravity
    plane_rect.centery+=plane_movement
    screen.blit(plane, plane_rect)
    floor_x_pos-=1
    game_floor()
    if floor_x_pos<=-1790:
        floor_x_pos=0
```

```
pygame.display.update()
```

Limita cu care să ruleze jocul

```
clock.tick(120)
```

6. Bibliografie

<https://www.techwithtim.net/tutorials/game-development-with-python/side-scroller-pygame/random-object-generation/> [1]

<https://stackoverflow.com/questions/54506167/detecting-if-two-sprites-overlap-using-pygame> [2]

<https://xaydungso.vn/bai-viet-khac/top-10-clouds-background-game-wallpapers-for-your-desktop-vi-cb.html> [3]

<https://www.pygame.org/docs/ref/image.html> [4]

<https://ro.wikipedia.org/wiki/Python> [5]

<https://www.saguna.ro/~dorulique/lucrgrad/Programarea%20paginilor%20WEB%20folosind%20limbajul%20PHP.pdf> [6]