Bolyai Farkas Elméleti Líceum

Marosvásárhely

Arcade Games Offline



Készítette:

Szövérfi Márk

Monda Antonio

XII.G Matematika-Informatika, intenzív informatika osztály

Felkészítő tanár: Jakab Irma-Tünde

2025

1. **Témaindoklás**

A projektünk célja egy offline működő applikáció megalkotása, amely a klasszikus arcade játékok világát idézi meg, modern köntösben. Az ötletet a csapat tagjainak közös érdeklődése inspirálta, hiszen mindannyian rajongunk a retro videojátékok iránt, és fontosnak tartjuk, hogy ezek a régi, ikonikus játékok ne merüljenek feledésbe. Szerettük volna az offline játéklehetőséget előtérbe helyezni, hogy a játékélmény bárki számára elérhető legyen, internetkapcsolat nélkül is.

A projekt előkészítése során részletesen megvitattuk, hogyan tudnánk a klasszikus játékélményt visszahozni a mai felhasználók számára. Olyan játékokat terveztünk, amelyek egyszerűek és letisztultak, hogy a felhasználók könnyen hozzáférjenek és élvezhessék a nosztalgikus élményeket. Az offline működés mellett kiemelt figyelmet fordítottunk arra, hogy az applikáció interaktív és vizuálisan vonzó legyen.

A fejlesztés során Python programozási nyelvet használtunk, amely lehetővé tette számunkra, hogy rugalmas és hatékony megoldásokat alkalmazzunk. Ez nemcsak technikai ismereteink mélyítésében segített, hanem betekintést nyújtott abba is, hogy milyen kihívásokkal kell szembenéznünk egy applikáció létrehozása során, a tervezéstől egészen a megvalósításig.

Reméljük, hogy az applikációnk felhasználói élménye nemcsak nosztalgikus hangulatot teremt, hanem hozzájárul az arcade játékok kultúrájának megőrzéséhez és népszerűsítéséhez. Kellemes szórakozást kívánunk minden játékosnak!

1. **Rendszer követelmények**

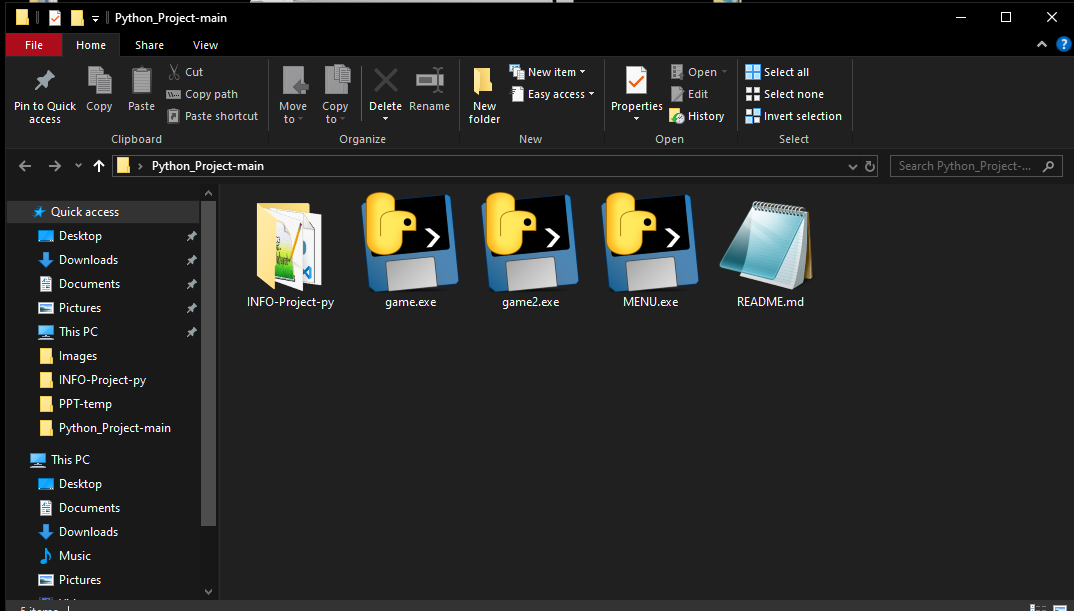
**Hardver igény:**

* Minimum: 2 GB RAM;
* Minimum: 2 magos 64 bites processor
* Minimum: 300 MB szabad tárhely
* Integrált grafika elegendő
* Legalább **14-15 hüvelyk**, Full HD (1920x1080) ajánlott

**Szoftver igény:**

* Windows 10/11 vagy bármilyen operációs rendszer ami tud .exe programokat futtatni
* Python 3.9 vagy újabb

1. **Felhasználói kézikönyv**

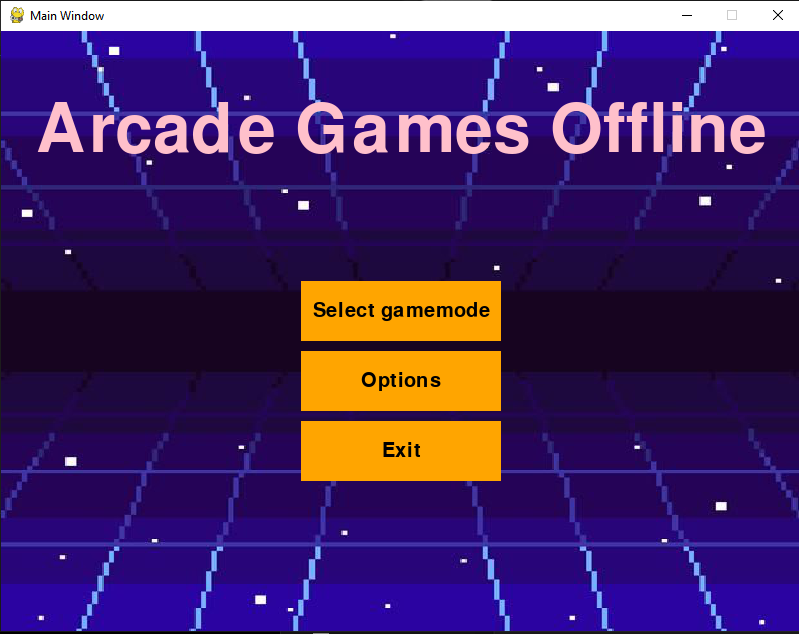


A **PyInstaller**nek köszönhetően a Python-alkalmazások könnyedén önálló futtatható fájlokká alakíthatók, így **minden külső telepítés nélkül működik és használható**! Az elkészült program azonnal futtatható külön konfiguráció nélkül.

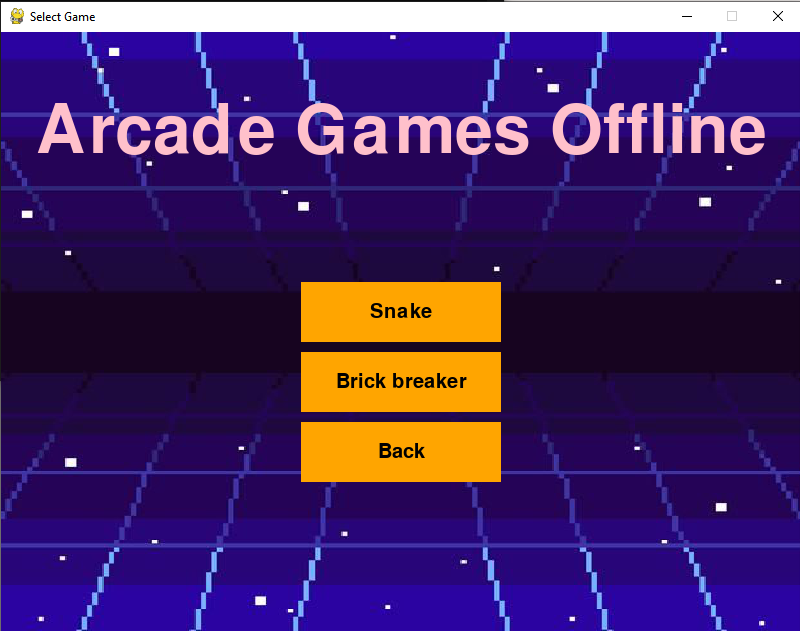
A játék a MENU.exe futtatásával érhető el.

1. Menu

A játék főmenüje három alapvető funkciót kínál, amelyek lehetővé teszik a felhasználók számára a játékmód kiválasztását, a beállítások módosítását és a játékból való kilépést.



* Select Gamemode: Ez a gomb lehetővé teszi a felhasználó számára, hogy kiválassza a játék módját**.**

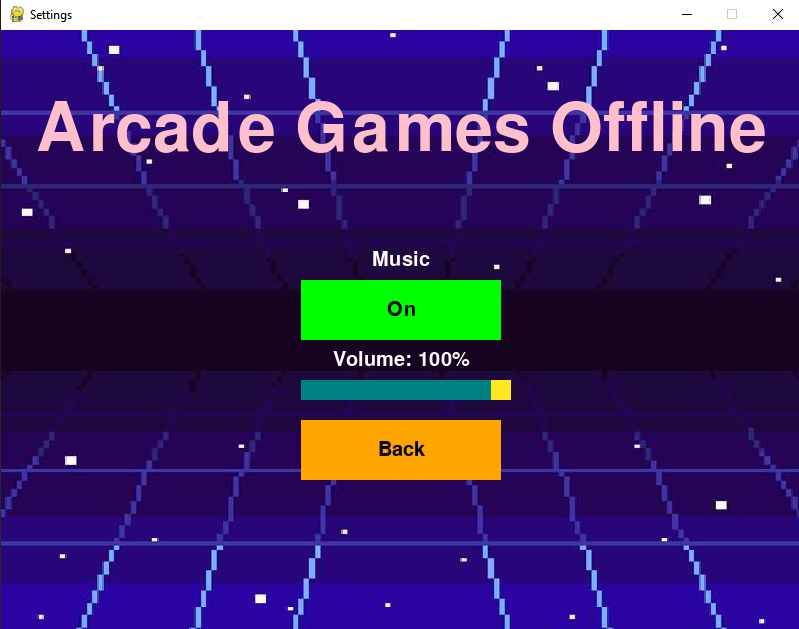
****

A kattintás után egy új képernyő jelenik meg, ahol különböző játékmódok közül választhat:

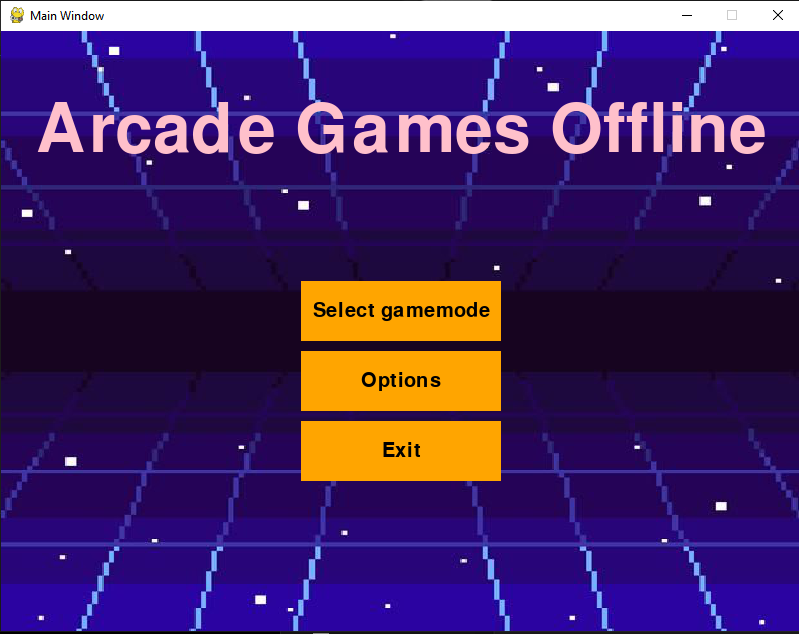
Snake – A klasszikus kígyós játék, ahol a cél minél hosszabbra növeszteni a kígyót anélkül, hogy nekiütközne a saját testének.

Brick Breaker – Egy ügyességi játék, amelyben egy ütővel irányított labdával kell eltörni az összes téglát a pályán.

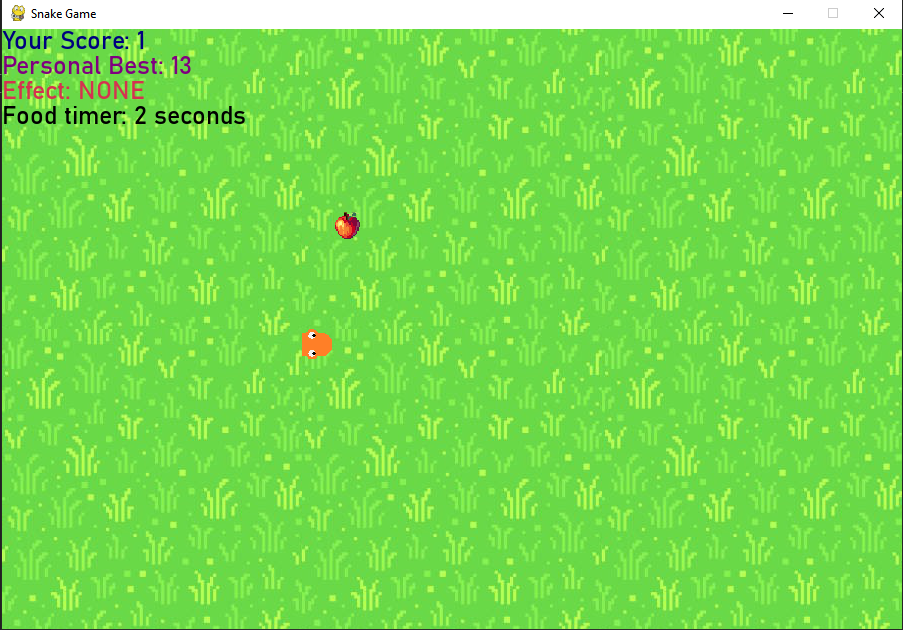
* Options: Az opciók gomb segítségével a felhasználó hozzáférhet az alkalmazás beállításaihoz. Itt lehetőség nyílik a zene ki/be kapcsolására és a hangerő szabályozására.



* **Exit: Az Exit gomb megnyomásával a felhasználó kiléphet az alkalmazásból.**



1. Snake

****

Ebben a speciális Snake játékban a játékos irányítja a kígyót, amely nem ütközik a falakba, hanem át tud rajtuk haladni, és körbemenni a pályán. A cél továbbra is az, hogy minél hosszabbra nőjön, miközben háromféle ételt fogyaszt: banán, alma és szilva.

Az **alma** a leggyakoribb 50% eséllyel jelenik meg a pályán és nincs különleges hatása.

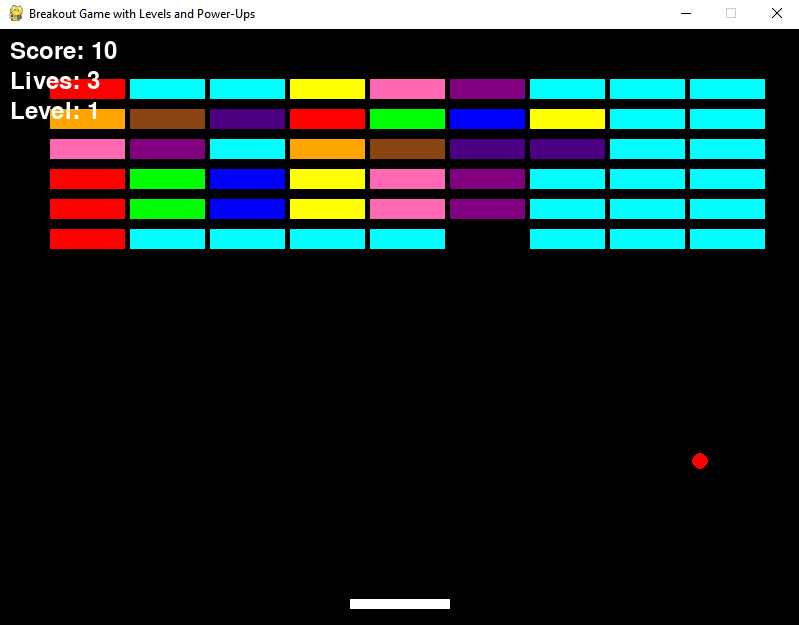
A **szilva** a második leggyakoribb étel amelyre pontosan 30% esély van és könnyít a játékon ugyanis 3 másodpercig felezi a haladási sebességet.

A **banán** a legritkább gyümölcs amit 20% eséllyel lehet kapni de értéke miatt nehezíti a játékot ugyanis 3 másodpercig kétszeresére növeli a játék sebességét.

Az ételek változatossága miatt minden gyümölcs csak 5 másodpercig van jelen, utána új gyümölcs sorsolódik.

A játék a **maximális pontszámot** az elfogyasztott ételek alapján számolja. Minden ételtípus eltérő pontértéket képvisel, és a játék automatikusan nyilvántartja a legmagasabb elért pontszámot.

1. Block breaker



A **Block Breaker** egy klasszikus ügyességi játék, amelyben a játékos egy mozgatható ütővel irányítja a labdát, és célja, hogy minden téglát eltörjön a pályán. A játék kezdetén a labda elindul, és a falaknak vagy blokkoknak ütközve visszapattan. A játékos balra és jobbra mozgatva az ütőt gondoskodik arról, hogy a labda ne essen le a képernyő aljára, különben életet veszít.

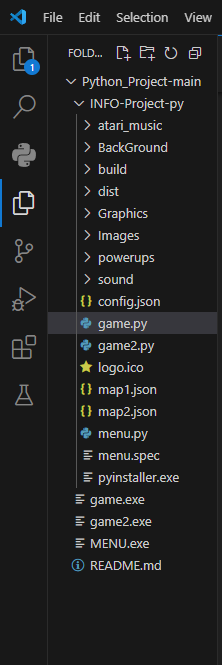
A pontszámítás a játékos teljesítményét tükrözi: minden eltört blokk után pont jár, és a játékos célja, hogy a lehető legmagasabb pontszámot érje el. A játék véget ér, ha minden blokk eltört, vagy ha a játékos elveszíti az összes életét.

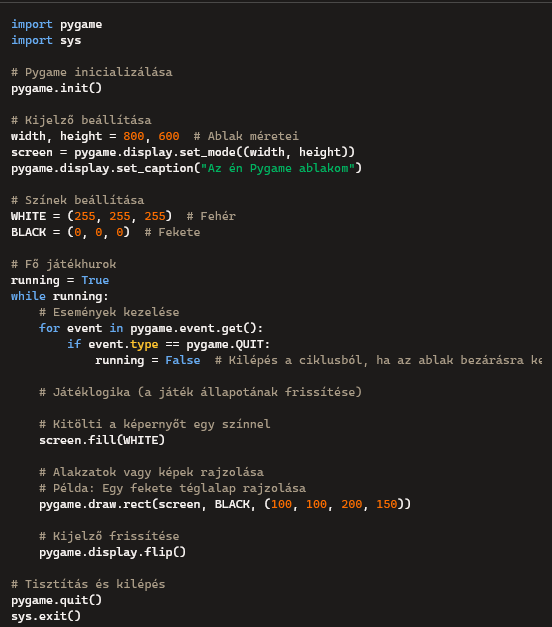
1. Programozói kézikönyv

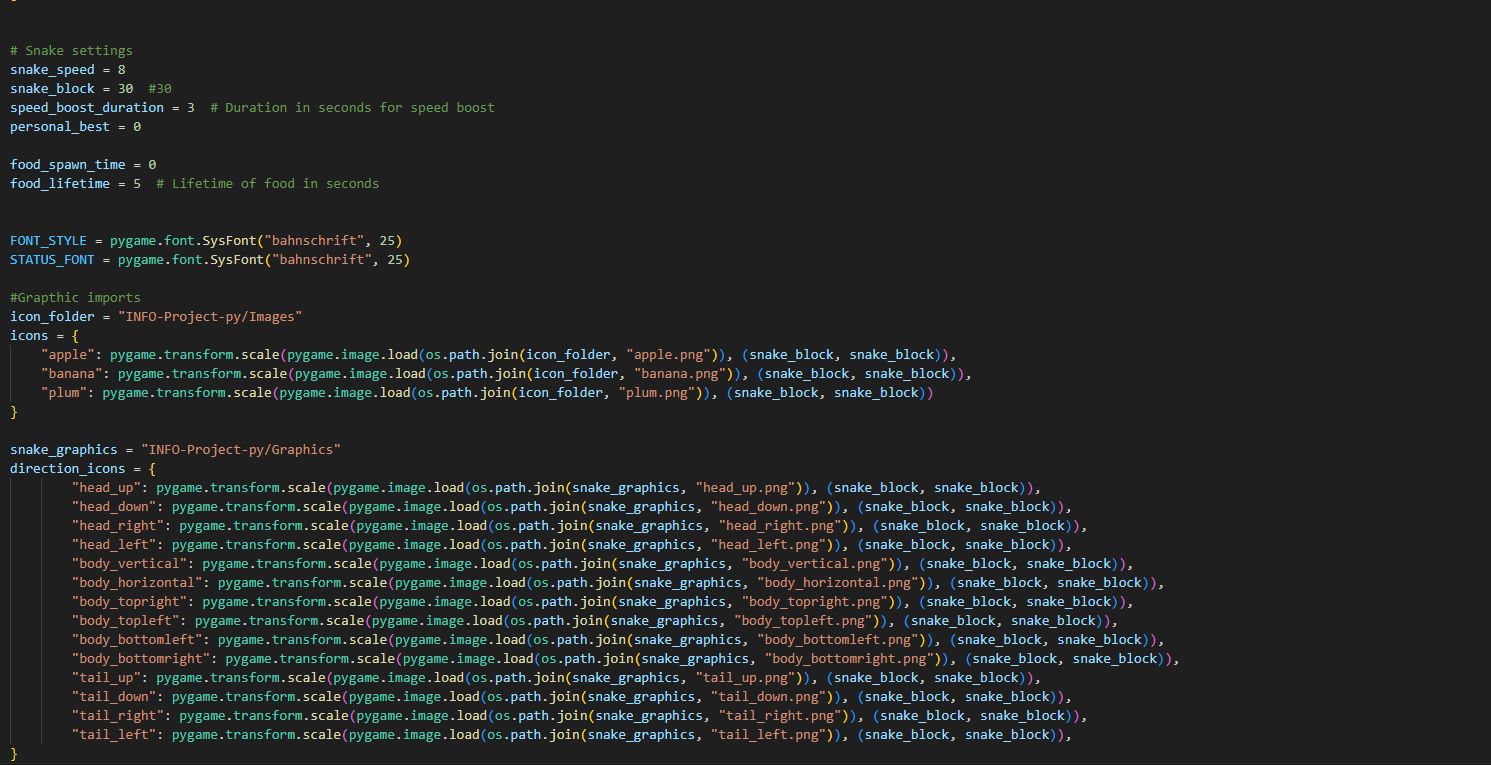
A játék fejlesztése során **Python** volt az alapvető programozási nyelv, amelyben minden logikát, számítást és folyamatot kezeltünk. Segítségével meghatároztuk a játékelemek viselkedését, mint például a kígyó mozgását a **Snake** játékban

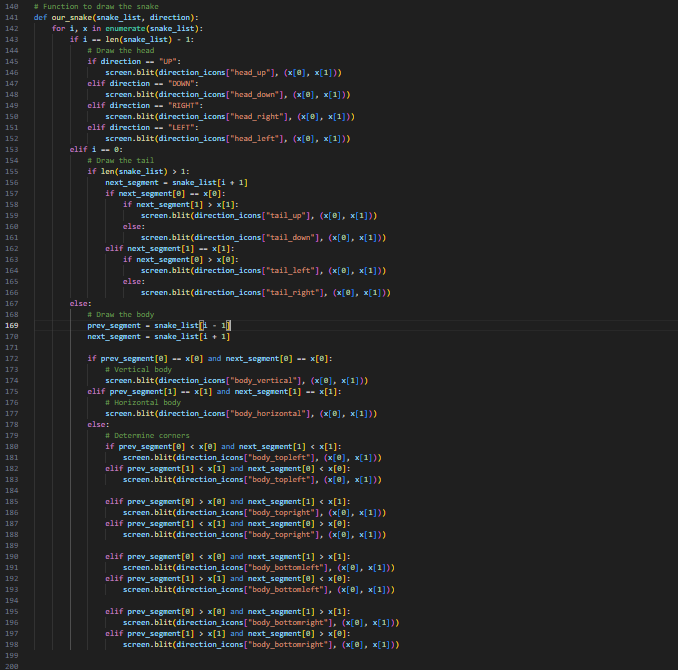
vagy a labda fizikai interakcióit a **Block Breaker** esetében. A Pythonban írt függvények biztosították az ételek hatásait, a pontszámítást és az eseménykezelést.

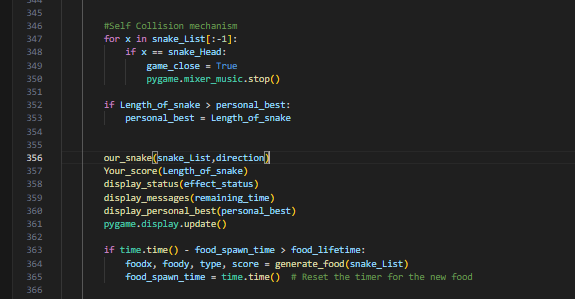
A grafikai megjelenítéshez és interaktivitáshoz **Pygame**-et használtunk, amely lehetővé tette, hogy megjelenítsük az objektumokat a képernyőn, kezeljük a felhasználói bemenetet, és animáljuk a játék elemeit. A Pygame segített abban, hogy a játék folyamatosan frissüljön és sima mozgásokat hozzunk létre a képkockák kiszámításával. Például a Snake játékban a kígyó mozgását irányítottuk, míg a Block Breaker esetében a labda visszapattanását és az ütközések érzékelését oldottuk meg.

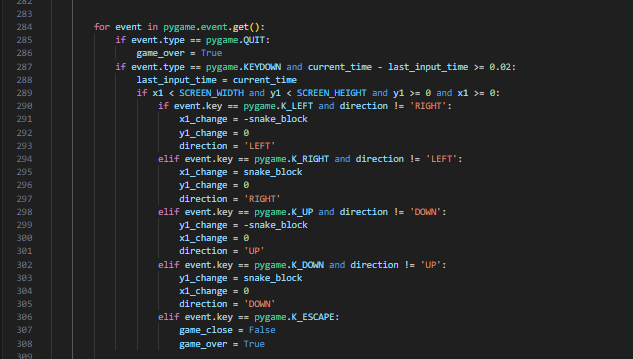
 Pygame alap szintaxisa



A kígyó grafikai megoldása:





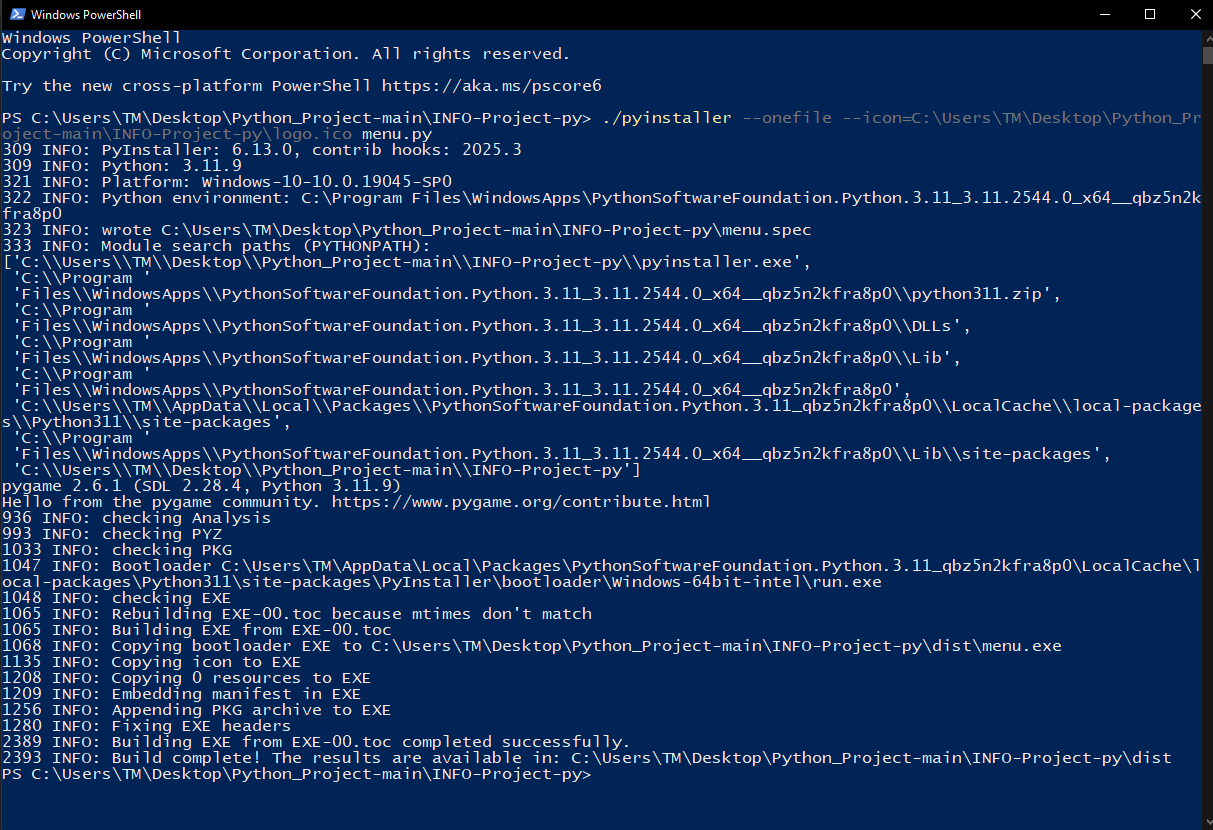
A Kígyó mozgása:

A kígyó fps-hez van kötve ezért a játékot csak 8 fps-en futtatjuk.

A Block breaker fizikája:



A Block breaker esetében a játék 60 fps-en fut, a labda sebességét külön számoljuk.



Végül, amikor készen állt a játék, a **PyInstaller** segítségével csomagoltuk az alkalmazást önálló futtatható fájlba, hogy a felhasználók telepítés nélkül játszhassanak vele. Ez biztosította, hogy a program minden szükséges függőséget magába foglaljon, így a végleges verzió zökkenőmentesen futtatható lett külön konfigurálás nélkül.

1. Fejlesztési lehetőségek

**Snake-be új játékmódokat lehet beleépíteni** mint például időkorlátos mód, ahol a játékosnak adott időn belül minél hosszabbra kell növelnie a kígyót.

**Ellenfél AI:** Egy másik, automatikusan mozgó kígyó, amely versenyez vagy akadályozza a játékost.

**Dinamikus pályaelemek**: Falak, portálok, vagy mozgó akadályok, amelyek új kihívásokat teremtenek.

**Block Breaker-be lehetne tenni új blokktípusokat**: Olyan blokkok, amelyek visszalőnek, mozognak vagy időzítve eltűnnek.

**Erősebb bónuszok**: Új ütőméretek, mágneses labda, vagy kettős sebesség.

**Pályakészítő**: Egy szerkesztő, ahol a játékosok saját pályákat tervezhetnek és játszhatnak.

**Többjátékos mód**: Egy kooperatív verzió, ahol két játékos közösen játszik, vagy egy versengő mód, ahol az egyik játékos blokkokat hoz létre, míg a másik eltörli őket.

1. **Szakirodalom**

<https://www.python.org/>

<https://pyinstaller.org/en/stable/index.html>

<https://www.pygame.org/docs/>

<https://docs.python.org/3/tutorial/index.html>

<https://www.w3schools.com/python/default.asp>

<https://pixabay.com/music/>

**7. Tartalomjegyzék**

1. Témaindoklás . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2. Rendszer Követelmény . . . . . . . . . . . . . . . . . . . . . . . 3
3. Felhasználói Kézikönyv . . . . . . . . . . . . . . . . . . . . . . . 4
4. Programozói Kézikönyv . . . . . . . . . . . . . . . . . . . . . . 12
5. Fejlesztési lehetőségek . . . . . . . . . . . . . . . . . .. . . . . 18
6. Szakirodalom . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19