

## 2. Forráskód

Az Infinite Loopers / Math Runner játék forráskódja egy **Python 3 + Pygame** alapú, egyfájlos prototípus formájában valósul meg, kiegészítve a szükséges grafikus és hang erőforrásokkal (háttérképek, platform tile, sprite-képek, hangeffektek). A kód a H4 rendszertervben definiált komponenseknek megfelelően valósítja meg a **játéklogikát, a megjelenítést és az eseménykezelést**.

A projekt jelenlegi struktúrája:

- main.py – a teljes játék logikáját és a főmenüt tartalmazó Python fájl
- képfájlok: sky.png, far-grounds.png, platform1.png, clouds.png, valamint a játékos sprite-jai (tile000.png–tile003.png)
- hangfájlok: bg\_music.mp3, jump.mp3, walk.mp3, coin.mp3, gameover.mp3
- későbbi bővítéshez: highscore.txt és egyéb adatfájlok (H4 adatkezelési terv szerint)

A H5 útmutató elvárásai szerint a forráskód **átlátható, moduláris felépítésű, objektumorientált elemeket használ**, és a fő funkciók külön osztályokba/függvényekbe szerveződnek.

### 2.1 Főbb osztályok és segédkomponensek

A main.py fájl a következő főbb osztályokat valósítja meg:

- **Animation**  
Általános célú animációs segédosztály.
  - Képlista vagy spritesheet alapján képkockákat tölt be.
  - Méretezést (scale) és képkocka-időt (frame\_time) kezel.
  - update(dt) a képkocka-léptetést, get\_frame() az aktuális frame visszaadását végzi.  
Ezt a Player sprite-animációjához használjuk.
- **Player**  
A játékos karakterét reprezentáló osztály.
  - Saját **ütközési téglalappal** (rect), függőleges sebességgel (vel\_y) és ugrásállapotokkal rendelkezik.
  - Kezeli a **gravitációt**, az **ugrást** (változtatható ugrásmagasság holding\_jump, max\_jump\_time segítségével), a földdel való ütközést és a vízszintes mozgást.
  - A update(keys, dt) metódus a bemenetek alapján mozgatja a játékost, vált az **idle/jump animációk** között, és clampeli a pozíciót a pályán belül.
  - A jump() / release\_jump() metódusok külön kezelik az ugrás indítását és elengedését.
  - A draw(surf) metódus a jelenlegi animációs frame-et rajzolja ki (jobbra/balra tükrözve).

- **Puff**

Ugráskor megjelenő **por-effekt**, amely kissé elmozdul, nő a sugara és halványul, amíg el nem tűnik.

- `update()` igaz/hamis értékkel jelzi, hogy az effekt még él-e.
- `draw(surf)` rajzolja ki az átlátszó köröket.

- **AnswerCube**

A lebegő, matematikai **válaszkockákat** reprezentálja.

- Kétdimenziós téglalap (`rect`) formájában tároljuk, `value` mezővel.
- A kockák **szinusz függvény mentén lebegnek** (`float_amp`, `float_speed`, `phase`).
- A `correct` flag jelöli, hogy ez-e a helyes megoldás.
- Helyes találat esetén villogtató effektet alkalmaz (`flash_alpha`, `flash()`), amit az `update(t)` csökkent.
- `draw(surf, font)` lerajzolja a kockát és fölé a válaszként szolgáló számot.

- **Coin**

A pályán elhelyezett **érméket** modellezi.

- Hasonlóan lebegő mozgást kap, mint a kockák.
- `update(t)` a függőleges pozíciót frissíti, `draw(surf)` egy ellipszist rajzol a képernyőre.

- **FloatingText**

A találatokhoz, pontszerzéshez társított **felugró szövegeket** kezel (pl. „+10 x2”).

- `update(dt)` a szöveg felfelé mozgását és élettartamát kezeli.
- `draw(surf, font)` áttetszővé halványítja a szöveget az idő függvényében.

- **Segédfüggvények**

- `draw_text(surf, text, font, x, y, center=False, color=TEXT_COLOR, alpha=255)` – általános célú szövegkiíró függvény, opcionális középre igazítással és alával.

## 2.2 Játékmenet és szintek kódja

A feladatok és szintek generálása, illetve a játék logikája jól elkülönített függvényekben található:

- **generate\_levels()**

- Visszaad egy listát az aktuálisan definiált matematikai feladatokról: pl. (" $3 + 5 = ?$ ", 8, [6, 9, 10]).
- Minden elem: (feliratozott egyenlet, helyes megoldás, hibás válaszok listája).
- A H4 analízismodell oktatási funkciójának (gyors fejszámolás) kódoldali leképezése.

- **setup\_level(levels, idx)**

- Az aktuális szinthez tartozó egyenlet és válaszok alapján létrehozza az **AnswerCube** példányokat, véletlen sorrendben, lebegő animációval.
- Létrehoz néhány lebegő **Coin** objektumot is, amelyek extra pontokat adnak.
- Visszaadja: equation\_text, cubes, correct\_answer, coins.

## 2.3 Fő játékhurok és állapotkezelés

A **fő játékmenet** a game\_loop(screen, clock) függvényben található, amely gyakorlatilag a H4-ben szereplő **GameManager** szerepét tölti be: a játék állapotgépét, eseményfeldolgozását és a komponensek együttműködését valósítja meg.

A főbb felelősségek:

- **Állapotváltozók inicializálása:**
  - score, lives, timer, combo, combo\_timer
  - aktuális szint indexe (current\_level), játékos objektum, effektek listái (puffs, floating\_texts)
  - logikai állapotok: game\_over, win, show\_level\_intro.
- **Eseménykezelés (input):**
  - **ESC** → kilépés a játékból (return "exit").
  - **SPACE / ↑** → ugrás (ha a Player engedi), ekkor ugrás hang + Puff effektek.
  - **R** → Game Over / Win állapotban újraindítás (return "restart").
  - KEYUP események → ugrás megszakítása (release\_jump()).
- **Frissítési logika:**
  - A Player update(keys, dt) metódusa mozgatja a karaktert a billentyűzetnek megfelelően.
  - Puffs, Coins, AnswerCube-ok és FloatingText-ek listája frissül (update()), a „lejárt” elemeket listakomprehenzióval szűri ki.
  - **Időzítés:** timer csökken, ha elfogy, életek (lives) csökkennek, esetleges Game Over vagy pályaugrás történik.
  - **Combo rendszer:** ha sokáig nincs helyes találat, a combo\_timer alapján a combo visszaáll 0-ra.
- **Ütközéskezelés:**
  - **Player ↔ AnswerCube** – ha ütközés történik és c.correct igaz:
    - flash(), combo++
    - szerzett pont = SCORE\_CORRECT \* combo
    - új szint betöltése vagy győzelem (win = True).
  - Ha rossz kockával ütközik:

- életék csökkentése, SCORE\_WRONG levonása, combo reset, esetleges game\_over.
- **Player ↔ Coin** – ha ütközés: SCORE\_COIN hozzáadása, FloatingText, coin eltávolítása.
- **Rajzolás és HUD:**
  - Többretegű **parallax háttér** (ég, felhők, távoli dombok, platform csempék).
  - Puffs, coins, player, cubes, floating text sorrendben rajzolódnak.
  - HUD: Score, Lives, Time, Combo, Equation text.
  - Level intro overlay, Game Over, Win overlay külön átlátszó felületen jelenik meg, egyértelmű felirattal és billentyűtippekkel.

## 2.4 Menü és belépési pont

A játék indításáért és a menükezelésért két rövidebb függvény felel:

- **menu(screen, clock)**
  - Egyszerű főmenüt jelenít meg két ponttal: **Start, Exit**.
  - Fel/le nyilakkal, illetve Enter/Space-szel lehet választani.
  - A kiválasztott menüelem alapján "start" vagy a program bezárása történik.
- **main()**
  - Beállítja a Pygame ablakot, címet, órajelet.
  - Ciklusban hívja a menu()-t és a game\_loop()-ot a felhasználó választásának megfelelően.
  - A program végeztével korrektül hívja a pygame.quit() és sys.exit() függvényeket.