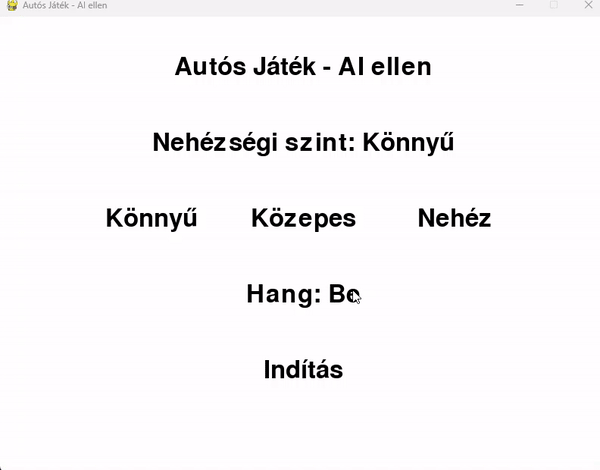
**Autós Játék**

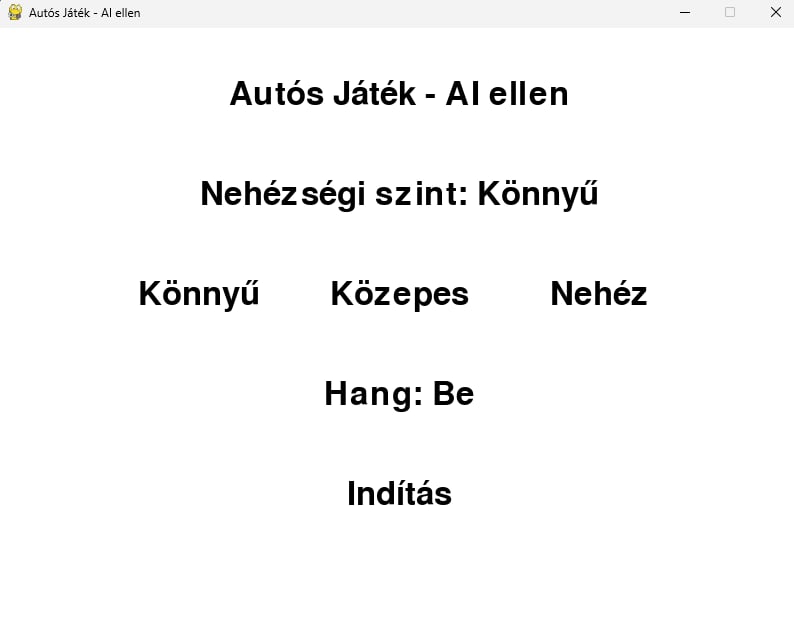
**(Felhőalapú szolgáltatások)**

**(Dormán László, Barna Loránd László, Bekő Virág)**

**Program ötlet:** Egy játék elkészítése, ahol autókat reprezentáló objektumokat kell kikerülni egy irányított testtel. Az alkalmazás elindítása után kettő azonos rész futna, az egyik részét a felhasználó tudná irányítani, a másikat pedig azonos módon az gépi tanulással training-ezet ai vezérelné. Mindkét esetben az objektumok (közlekedési eszközök) kikerülése lenne a cél és az veszítene, aki hamarabb ütközik háromszor. (minden harmadik kikerült objektum esetén életet nyer a felhasználó, illetve az ai)



**A kezdőfelület kialakítása:** 3 nehézségi fok közül lehet választani: könnyű, közepes és nehéz. A játék játszható hanggal, illetve a nélkül. Hangok 3 eseményhez vannak rendelve: életszerzés, ütközés, pontszerzés. Mindegyik eseményhez külön hangfile van, hogy megkülönböztethetőek legyenek az interakciók egymástól. A felhasználónak és az ai cselekedeteihez azonos hangok tartoznak. A kezdő felületen még elhelyezkedik egy indítás gomb, amivel értelemszerűen el lehet indítani a játékot.

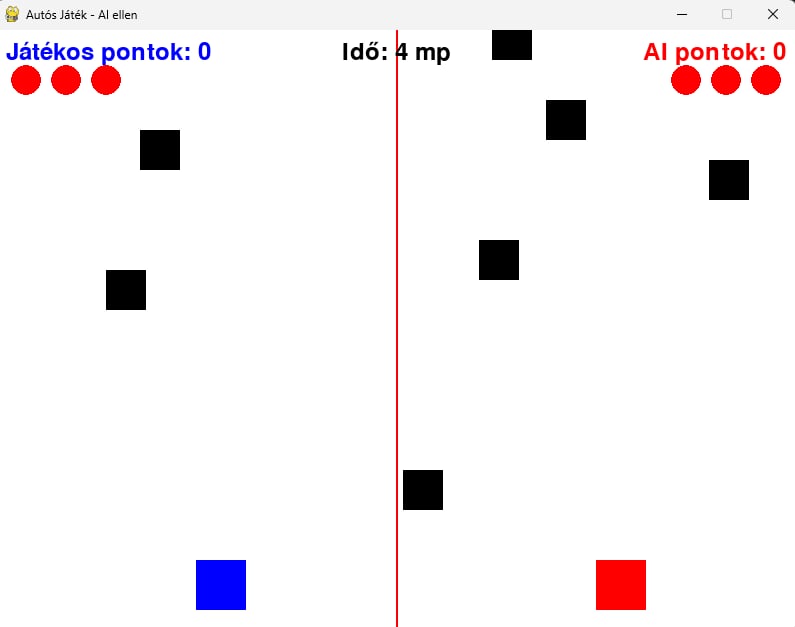


**Játékfelület kialakítása:**

**Pontok kijelző:** az ai és játékos 3 objektum kikerülés esetén +1 pontot kapnak

**Időmérő:** a játék elkezdésétől eltelt idő

Kettő azonos nagyságú játéktérfél, az egyiken az ai játszik, a másikon a felhasználó.



**Felmerülő főbb problémák az AI betanítása közben:**

* Az Ai abszolút nem akart megmozdulni egyik irányba sem
* Az ai által vezérelt objektum sok ideig csak balra indult el és ott is maradt a bal sarokban, amikor odaért
* Az ai által vezérelt objektum “remegése”
* Az ai tanulásának finomhangolása
* A játéktér kialakítása

**Probléma felismerése:**

1. A mesterséges intelligenciánk azért nem akart az elején egyáltalán megmozdulni, mert rosszul lettek megadva neki a mozgási feltételek és ezeket kellett kijavítani.
2. A mesterséges intelligencia folyamatos balra húzásának az oka az volt, hogy a jutalmazási rendszerben a jutalmazások mennyisége rosszul volt beállítva.
3. A mesterséges intelligencia remegése, amit nem sikerült sajnos száz százalékosan megszüntetnünk, arra tudunk gondolni, hogy nincsen megfelelően beállítva a tréningezés és ezért a mesterséges intelligencia bizonytalan a döntéseiben, ez okozza a remegését.
4. A mesterséges intelligencia térfelében a játéktér rosszul lett beállítva és azt hittük, hogy valójában a mesterséges intelligencia a hibás és nem tud jobbra menni, de valójában a jobb oldali fal a spawn pontján volt, ez egy láthatatlan falat képezett így. Ez blokkolta abban, hogy jobbra tudjon mozogni. (mivel azt hittük, hogy a mesterséges intelligencia a hibás nagyon sok időt fordítottunk a probléma megoldására úgy, hogy rossz irányból közelítettük meg a problémát.)
5. Az ai tanítása egy nagyon idegőrlő munka, mivel sosem lehet tökéletesre tanítani. Meg kell elégednünk a célunknak megfelelő teljesítménnyel rendelkező mesterséges intelligenciával.

**Felhasznált Technológiák:**

Könyvtárak: gym Környezet definiálására és interakciókra.

numpy: Numerikus számításokhoz.

stable-baselines3: A PPO algoritmus implementációjához.

Fejlesztési környezet: Python 3.x Jupyter Notebook vagy más IDE az egyszerűbb fejlesztéshez.

**Tréning algoritmus:**

Proximal Policy Optimization (PPO) Hiperparaméterek:

Tanulási ráta: 0.000001 (nagyon alacsony a stabil tanuláshoz).

Tréning lépések: 15,000 időlépés.

Gamma (diszkont faktor): 0.99 Batch méret: 1024 Érdekeltségi tényező (ent\_coef): 0.001 (exploráció vs. exploitáció).

Tréning folyamata: Az AvoidObstacleEnv környezetet egy DummyVecEnv csomagolta, hogy kompatibilis legyen a PPO implementációval. A tréning során a modell iteratívan optimalizálja az akciópolitikáját, hogy minimalizálja az akadályokkal való ütközéseket, miközben maximalizálja a jutalmakat.

**Tesztelés menete:**

* 100 lépésen keresztül próbálja elkerülni az akadályokat. Minden lépésben a következő történik:
  + A modell előrejelzi az optimális akciót az aktuális megfigyelés alapján.
  + Az akció végrehajtása után frissíti a környezet állapotát.
  + A jutalom, a környezet állapota, valamint az akadályok pozíciói és sebessége megjelennek a konzolon.
* **Ha az AI akadályba ütközik:** 
  + A teszt leáll az aktuális epizódra.
  + A környezet újraindul (reset), és folytatódik a teszt.

**Kimenet:**

* Lépésenkénti információ: Akció típusa (balra, jobbra, marad).
* Jutalom értéke.
* Az AI normalizált pozíciója.
* Akadályok pozíciója (X, Y) és sebessége.
* Ütközés esetén figyelmeztetés, majd környezet újraindítása.

Köszönjük a figyelmet! 😊