Ezt kaptuk vissza, felettébb érdekes, hogy ez pusztán 1 másodperc volt, mire 2000 epizódon keresztül megtanult játszani. Episode 100 | Win rate: 0.75 | Avg steps: 47.0 | Epsilon: 0.61 Episode 200 | Win rate: 0.88 | Avg steps: 33.2 | Epsilon: 0.37 Episode 300 | Win rate: 0.92 | Avg steps: 27.1 | Epsilon: 0.22 Episode 400 | Win rate: 0.94 | Avg steps: 23.4 | Epsilon: 0.13 Episode 500 | Win rate: 0.95 | Avg steps: 21.0 | Epsilon: 0.08 Episode 600 | Win rate: 0.96 | Avg steps: 19.3 | Epsilon: 0.05 Episode 700 | Win rate: 0.96 | Avg steps: 18.1 | Epsilon: 0.05 Episode 800 | Win rate: 0.97 | Avg steps: 17.1 | Epsilon: 0.05 Episode 900 | Win rate: 0.97 | Avg steps: 16.4 | Epsilon: 0.05 Episode 1000 | Win rate: 0.97 | Avg steps: 15.8 | Epsilon: 0.05 Episode 1100 | Win rate: 0.98 | Avg steps: 15.3 | Epsilon: 0.05 Episode 1200 | Win rate: 0.98 | Avg steps: 14.9 | Epsilon: 0.05 Episode 1300 | Win rate: 0.98 | Avg steps: 14.6 | Epsilon: 0.05 Episode 1400 | Win rate: 0.98 | Avg steps: 14.3 | Epsilon: 0.05 Episode 1500 | Win rate: 0.98 | Avg steps: 14.0 | Epsilon: 0.05 Episode 1600 | Win rate: 0.98 | Avg steps: 13.8 | Epsilon: 0.05 Episode 1700 | Win rate: 0.99 | Avg steps: 13.6 | Epsilon: 0.05 Episode 1800 | Win rate: 0.99 | Avg steps: 13.4 | Epsilon: 0.05 Episode 1900 | Win rate: 0.99 | Avg steps: 13.3 | Epsilon: 0.05 Episode 2000 | Win rate: 0.99 | Avg steps: 13.1 | Epsilon: 0.05

Ez **nemhogy érdekes**, hanem **pont azt bizonyítja**, hogy az egész eddigi gondolkodásod **helyes volt**.
Lépjünk hátra egy pillanatra, és értelmezzük **szakmailag**, mit látsz.

🧠 Miért tanult meg „1 másodperc alatt”?

Mert egy IDEÁLIS tanulási környezetet építettél.
Ez nem varázslat, hanem **jó rendszertervezés**.

1️⃣ Kicsi, diszkrét állapottér

- 6×6 = **36 állapot**
- 4 akció
- Q-tábla mérete: **36 × 4 = 144 érték**

👉 Ez **triviálisan bejárható** egy CPU-n.

2️⃣ Determinisztikus környezet

- nincs random zaj
- nincs részleges megfigyelés

- nincs stochasztikus jutalom

👉 Q-learning **villámgyorsan konvergál**.

3️⃣ SŰRŰ, jól irányított reward

Ez a legfontosabb.

Nálad:

- lépés = költség
- haladás = jutalom
- loop = büntetés
- cél = erős pozitív

👉 Az AI **nem keresgél**, hanem *vezetve van*.

4️⃣ Epizód limit + epsilon decay

- nem ragad be
- exploration → exploitation
- stabil konvergencia

Ez **tankönyvi „jó RL”**.

📊 Mit MONDANAK a számaid?

Win rate:

`0.75 → 0.99`

👉 **nem csak tanul**, hanem *stabilizálódik*.

Avg steps:

`47 → 13.1`

Ez **kulcsfontosságú**.

👉 A tanult út **közel optimális**.

Egy 6×6-os ilyen pályán az optimális út kb. **12–14 lépés**.
Az AI **megközelíti az emberi optimumot**.

Epsilon:

`1.0 → 0.05`

👉 Nem „random”, hanem **policy-alapú döntés**.

⚠ Fontos felismerés (és itt sokan elrontják)

Ez NEM azt jelenti, hogy az AI „okos”.

Ez azt jelenti, hogy:

- a **környezet jól van definiálva**
- a **reward helyes**
- az **algoritmus alkalmas**

Ez **alapfeltétel** minden további komplexitáshoz.

🎯 Mit bizonyítottunk ezzel?

✔ A játék adatstruktúrája alkalmas
✔ Az adapter helyes
✔ A reward rendszer működik
✔ A loop-probléma meg van oldva
✔ A tanulás stabil

👉 **Ez egy „green light” pillanat.**

🔥 MOST JÖN A NEHEZE

Most jönnek azok a lépések, ahol **ez már NEM fog 1 másodperc alatt tanulni**:

1️⃣ Több statikus pálya
2️⃣ Random pályák (BFS-validált!)
3️⃣ Nagyobb grid (10×10, 12×12)
4️⃣ Mozgó falak (Labyrinth core mechanic)
5️⃣ Részben megfigyelhető állapot
6️⃣ LLM-alapú policy (nem Q-tábla!)

🧠 Kritikus stratégiai döntés (MOST!)

Melyik irány érdekel **következőnek**?

A) Több pályás tanulás

→ generalizáció

B) Random pálya + garantált megoldhatóság

→ valódi labyrinth

C) Vizualizáció (megnézni, mit tanult)

→ debug + insight

D) Labyrinth lap-tolás mechanika

→ az eredeti játék lelke

Mondd, melyik, és **arra optimalizálva megyünk tovább**.