

Többszörös Barrierek összehasonlítása Game of life szimulálásával

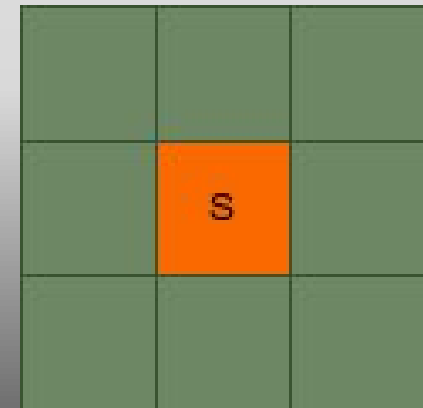
Készítette:

Szabó Levente Zoltán

ECNQ22

Game of life

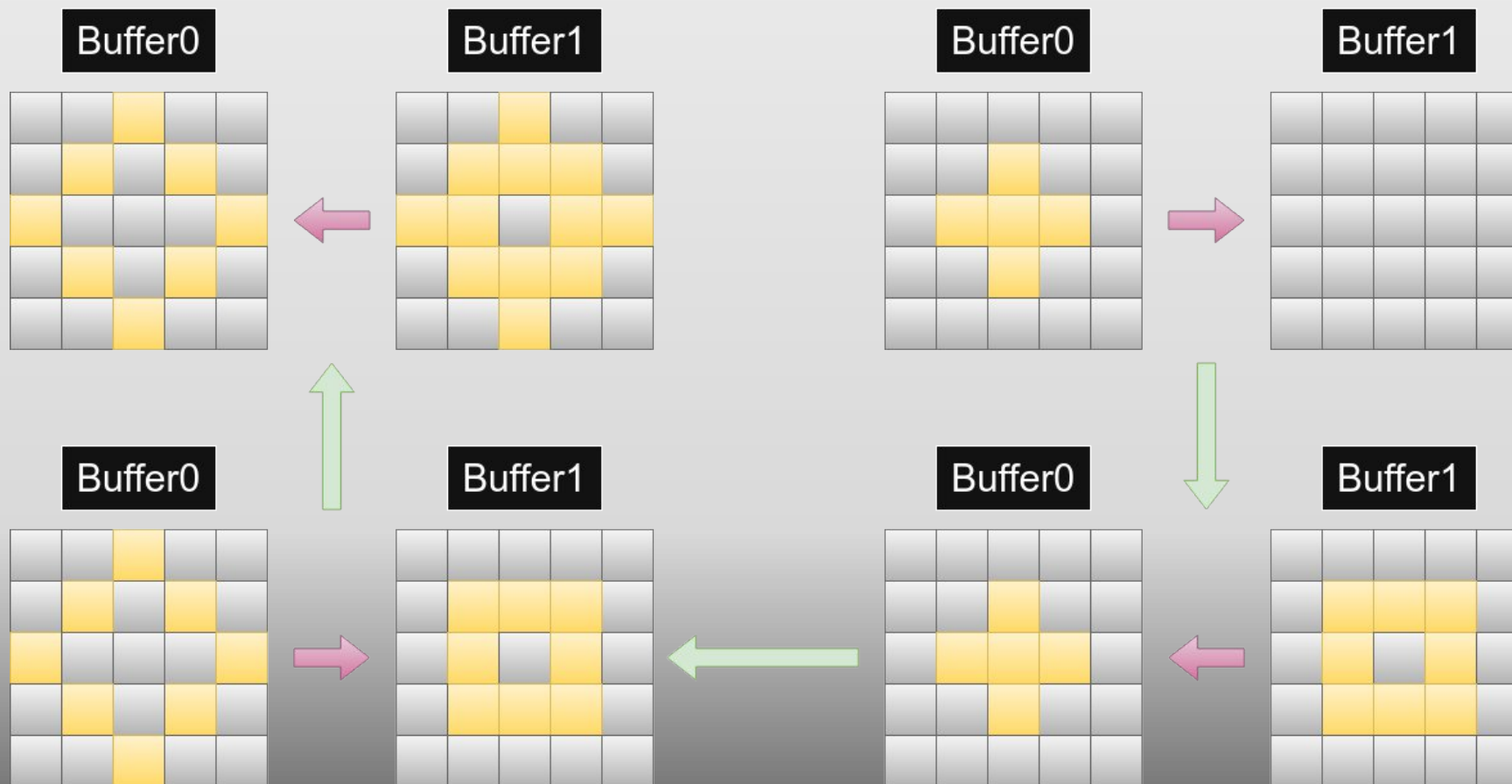
- A game of life egy valahányszor valahányas táblán az egyes sejtek állapotainak változásáról szól.
- Egy sejt lehet “Dead” vagy “Élő” állapotban.
- Egy szimulációs lépés alatt a táblán lévő összes sejt állapota frissül, a tábla előző állapota alapján.
- Egy sejt a saját állapota és a szomszédban lévő sejtek állapotjai alapján változtatja meg a saját állapotát.
- Egy **sejtnek** 8 darab **szomszédja** van az alábbi féle képpen:



Rajzolás menete

- 2 Buffert használunk a képernyő pixeleinek eltárolására (Buffer0, Buffer1).
- Mindkettő buffer ugyanakkora.
- Először a **Buffer0**-ba rajzolunk a **Buffer1** állapota alapján.
- Ezután fordítva, **Buffer1**-be rajzolunk a **Buffer0** alapján.
- Majd megint a **Buffer0**-ba rajzolunk a **Buffer1** alapján és így tovább...
- A rajzolást egy középre helyezett “**Alive**” sejtekből álló “**+**” alakú formációval kezdjük.
- A többi cella ezen kívül “**Dead**”.
- A pálya kap egy konstans “**Dead**”-ből álló szegélyt is.
- Minden tizedik frame-nél egy random pozícióra besúrunk egy “**+**” jelet.

Rajzolás menete



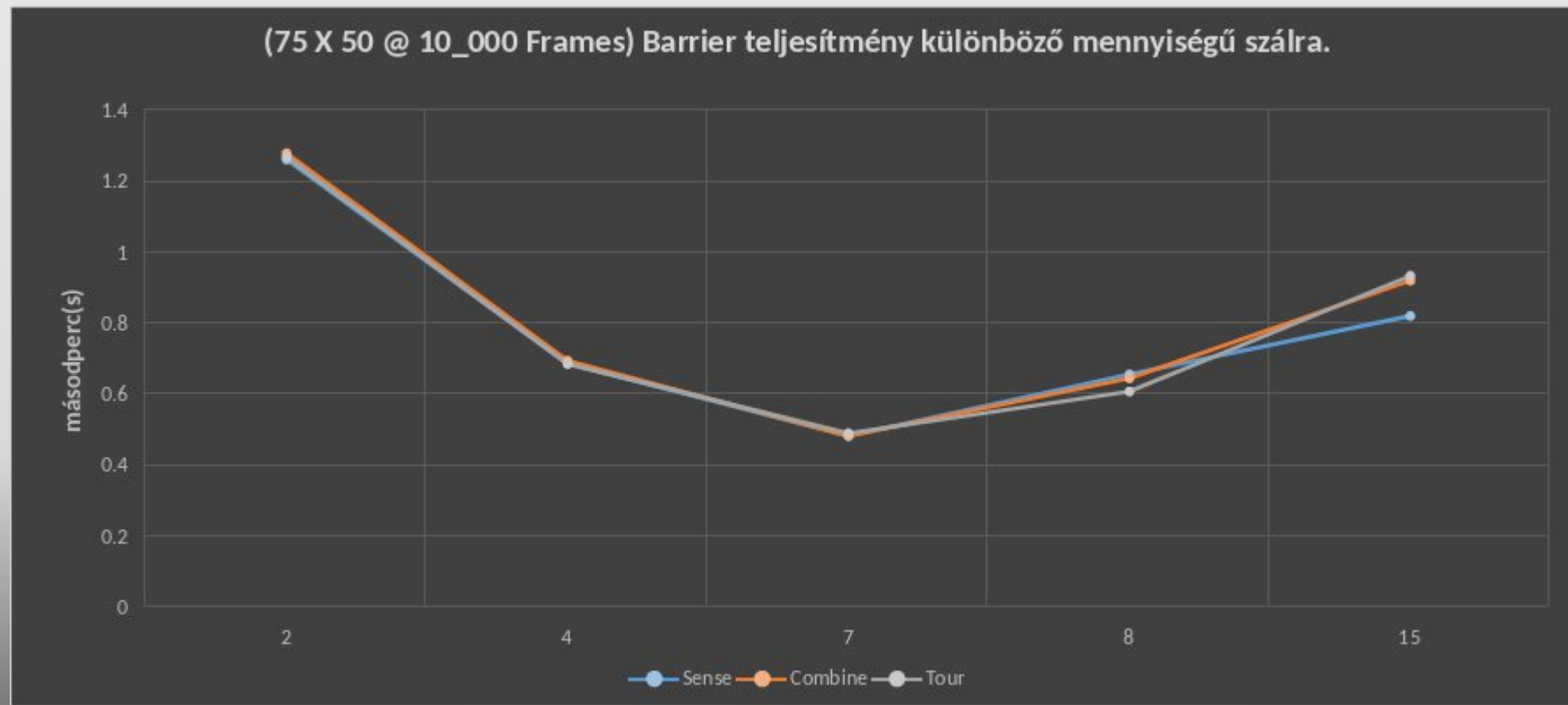
Párhuzamosítás

- Legyen w darab dolgozó szálunk és egy váltogató szálunk.
- Legyen egy start és egy end Barrier-ünk, ahol mindkettő $w+1$ szálat vár meg.
- Először a dolgozó szálak várakoznak a start-ban.
- Amint a váltogató el akar kezdeni egy rajzolást, `.await()`-el a start-ban, hogy elindítsa a rajzolást. Ezután `await`-el az end-ben.
- Eddig a dolgozó szálak soronként párhuzamosan kiszámítják a cellák következő állapotait.
- Mondjuk három dolgozó esetén az első dolgozó dolgozik az 1,4,7... sorokon, a második a 2, 5, 8 ... sorokon és a harmadik a 3,6,9 ... sorokon.
- Ha végeztek, akkor ők is várakoznak az end-ben.
- Ekkor az összes szál továbbmegy, a váltogató válthat a bufferek között, a dolgozók pedig várakoznak a start-ban, egy következő frame megkezdéséig.

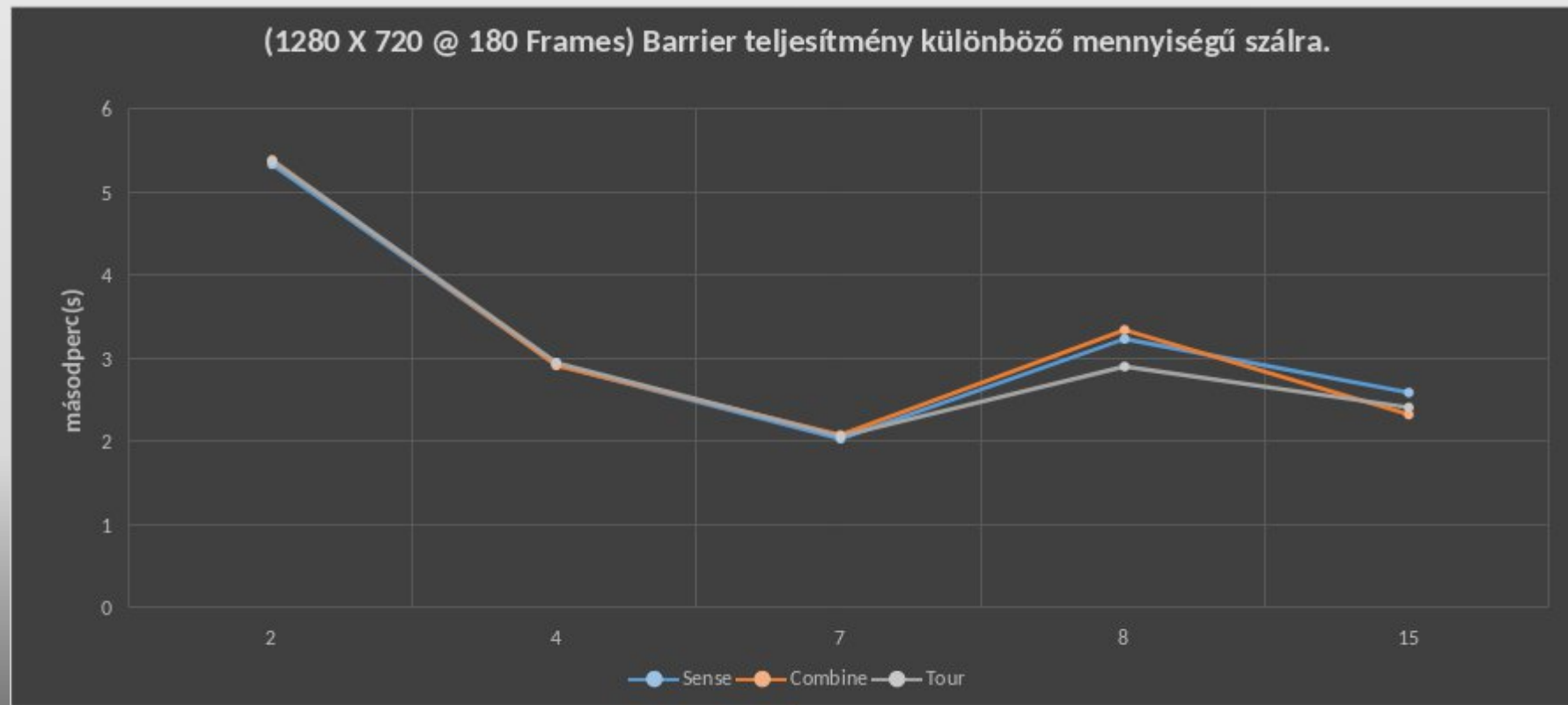
Párhuzamosítás

- Hogyha már nem kell tovább rajzolni, akkor a váltogató szál beállít egy boolt true-ra, amit minden dolgozó szál a start után megnéz és ha igaz, akkor befejezi a munkáját. Ezért a váltogatónak csak a start-ban kell várakoznia egyet a bool beállítása után.
- Három barrier típust implementáltam le:
 - SenseReversingBarrier (Sense)
 - CombiningTreeBarrier (Combine)
 - TournamentTreeBarrier (Tour)

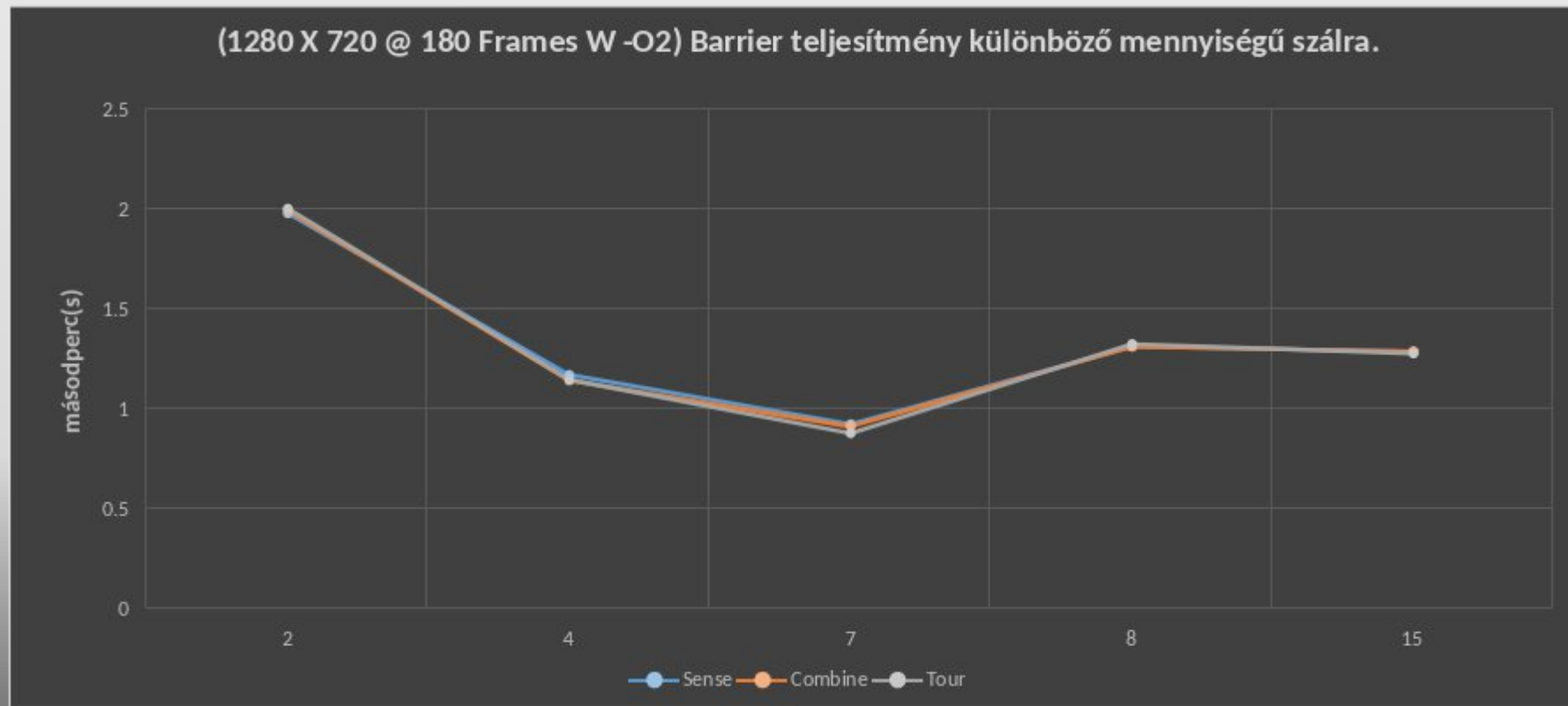
Mérések



Mérések



Mérések



Köszönöm a figyelmet!

Készítette:

Szabó Levente Zoltán

ECNQ22