

1 Opis działania

Rozwiązanie zawiera klasy **PlayerCPU** oraz **PlayerGPU** wybierające następne posunięcie przy pomocy Monte-Carlo Tree Search, przeprowadzając symulacje odpowiednio na jednym wątku CPU i GPU. Jedyna część MCTS korzystająca na równoległości to faza symulacji, stąd implementacje różnią się jedynie metodą przeprowadzającą ten etap.

2 Usprawnienia

- Losowe liczby na GPU generowane są przy pomocy algorytmu KISS, na podstawie regenerowanych seedów.
- Cała plansza przechowywana jest na 96 bitach.
- Potencjalne bicia przechowywane są na 32 bitach.
- Struktura drzewa przechowywana jest po stronie CPU, przeprowadzenie symulacji wymaga przekazania 96 bitów na planszę, 8 bitów na ilość ruchów bez bicia, oraz 1 bita oznaczającego, natomiast zwracany wynik zapisany jest na 32 bitach
- Instrukcje procesora, zamiast funkcji
- Brak rekurencji, użycie własnej implementacji unsafe kolejki o wstawianiu, ekstrakcji i sprawdzaniu długości w $O(1)$
- Programowanie bezgałęziowe

3 Performance

Test przeprowadzono przy użyciu: Intel i5-6400 2.7GHz, NVIDIA GTX 1060 6GB.

Wersja **GPU** domyślnie uruchamia bloki rozmiaru 512, pozwalając jednocześnie uruchomić 64 bloki. Próba uruchomienia większej ilości wątków skutkuje w sekwencyjnym uruchamianiu $512 \cdot 64$ wątków.

	GPU	CPU
1024	4 ms	22 ms
4096	4 ms	159 ms
16384	8 ms	333 ms
32768	17 ms	668 ms

Tabela 1: Długość potrzebna do przeprowadzania danej ilości symulacji

cudaMalloc	0.010912 ms
cuRand	0.654336 ms
thrust::reduce	0.321536 ms
pozostałe	16 ms

Tabela 2: Czas wykonywania poszczególnych części na GPU, dla 32768 symulacji

Rubryka **cudaMalloc** odpowiada alokacji $N \cdot 162$ bitów, gdzie N to ilość wątków w cyklu wywołania, tutaj $N = 16384$. Pamięć ta jest wykorzystywana do zapisywania wyników symulacji i generowania seedów dla generatorów pseudolosowych w każdym wątku. **cuRand** odpowiada stworzeniu generatora liczb pseudolosowych oraz generacji $N \cdot 5$ liczb. **thrust::reduce** to czas potrzebny na zsumowanie $2N$ wartości bool. Pozostały czas to inicjalizacja tablicy remisów zerami i wywołanie kernela.