README.md

CGOL

This is an implementation of Conway's game of life on CUDA.

Compiling: `nvcc -o cgol cgol.cu`

Running arguments: `./cgol [s NUMBER] [i NUMBER] [t NUMBER] [b NUMBER] [e NUMBER] [p 0|1] [u] [a]`

- s: Board size (Width, Default is 32)
- i: Number of iterations (Default is 30)
- t: Number of threads (Default is equal to board width)
- b: Number of blocks (Default is equal to board width)
- a: Animate (Default is false)
- e: Random seed (Default is NULL; different everytime)
- p: Print board (Default is true)
- u: Run unoptimized version using global memory

For example, to run a game with a 32x32 board for 100 iterations with animations, run

```
./cgol s 32 i 100 a
```

Here's a sample output of what the last iteration of a random board looks like.

```
▢ ▢ ▣ ▣ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢
▢ ▢ ▢ ▢ ▣ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢
▣ ▢ ▢ ▢ ▣ ▢ ▢ ▢ ▢ ▢ ▢ ▣ ▣ ▢ ▢ ▢
▣ ▣ ▢ ▣ ▣ ▢ ▢ ▢ ▢ ▢ ▢ ▣ ▣ ▢ ▢ ▢
▢ ▣ ▣ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢
▢ ▢ ▣ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢
▢ ▢ ▢ ▢ ▣ ▢ ▣ ▣ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢
▢ ▣ ▢ ▢ ▣ ▣ ▣ ▣ ▢ ▢ ▣ ▣ ▢ ▢ ▢ ▢
▣ ▣ ▣ ▢ ▢ ▣ ▢ ▢ ▢ ▢ ▣ ▣ ▢ ▢ ▢ ▢
▣ ▢ ▢ ▣ ▢ ▢ ▣ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢
▣ ▢ ▢ ▢ ▢ ▣ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢
▣ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢
▣ ▣ ▢ ▢ ▢ ▣ ▢ ▢ ▣ ▣ ▣ ▣ ▣ ▢ ▣ ▣
▢ ▣ ▣ ▣ ▢ ▣ ▢ ▢ ▣ ▣ ▣ ▣ ▢ ▢ ▢ ▣
▢ ▢ ▣ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▣ ▣ ▣ ▢
▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢ ▢
```