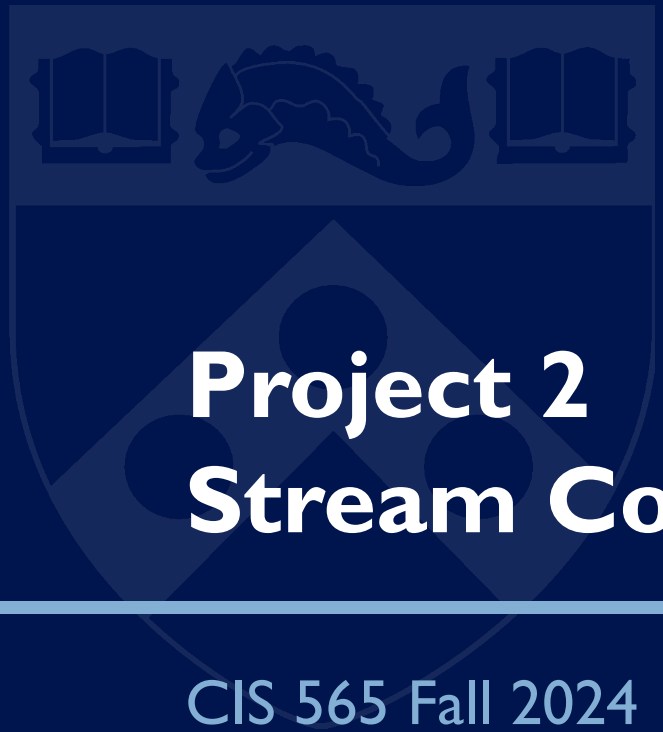
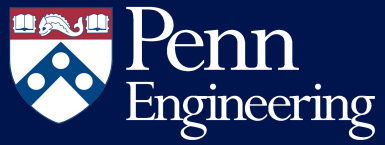


Project 2
Stream Compaction

CIS 565 Fall 2024



Stream Compaction

cpu.cu

`void scan(int n, int *odata, const int *idata)`	Simple CPU scan
`int compactWithoutScan(int n, int *odata, const int *idata)`	Simple CPU compact
`int compactWithScan(int n, int *odata, const int *idata)`	CPU compaction using scan

- The results of `scan` and `compactWithoutScan` will be used to compare to the results of your GPU scan and stream compaction. Please make sure your CPU implementation is correct.

naive.cu

`void scan(int n, int *odata, const int *idata)`	Naive GPU Scan

efficient.cu

`void scan(int n, int *odata, const int *idata)`	Efficient GPU Scan
`int compact(int n, int *odata, const int *idata)`	GPU stream compaction using scan

Stream Compaction

thrust.cu

`void scan(int n, int *odata, const int *idata)`	Thrust GPU scan

common.cu / common.h

```
__global__ void kernMapToBoolean(int n, int *bools, const int *idata)
__global__ void kernScatter(int n, int *odata, const int *idata, const int *bools,
const int *indices)
inline int ilog2(int x) // return (int)log2x
inline int ilog2ceil(int x) // ceiling of ilog2
#define checkCUDAError(msg) // simple CPU scan
```

- PerformanceTimer
  - Put your CPU code between `timer().startCpuTimer()` and `timer().endCpuTimer()`
  - Put your GPU code between `timer().startGpuTimer()` and `timer().endGpuTimer()`
  - `std::chrono` to measure CPU time cost / cudaEvent to measure CUDA time cost

Tips

- Be sure **exclude** any initial/final memory operations (`cudaMalloc`, `cudaMemcpy`) in your performance measurements. We only care about how fast the algorithm itself is but not the memory latency.
- If you find your GPU stream compaction is slower than CPU, check whether it's necessary to use so many threads.
- Remember to update `stream_compaction/CMakeLists.txt` and regenerate your `.sln` if you add new source files into your project
- Remember that CUDA kernels are asynchronous! Don't forget to call `cudaDeviceSynchronize()` in situations where race conditions are possible. (Not applicable for Default Stream)
- Use powf() instead of pow() to prevent float point error in CUDA
- Always test in Release mode