Ławice ryb - opis projektu

10 kwiecień 2024

1 Ważne informacje

- Aby przełaczyć program w tryb CPU należy odkomentować linie 5 w pliku config.cuh (define CPU).
- Program należy uruchamiać w trybie Release, bez właczonego tryb CPU dla najlepszej wydajności. Duży wpływ na wydajność ma również ilość ryb (linie 47 i 49 w pliku config.cuh) oraz parametr zachowania ławic Visiblity, który można zmienić suwakiem w panelu kontrolnym na mniejsza wartośc.
- Jeśli program sie nie uruchamia, odpowiednie instrukcje zostały dołaczone w pliku README.txt.

2 Ważne pliki

W skrócie ważne pliki to: config.cuh, fishes.cuh, fishes.cu, fishes_model.cuh, fishes_model.cu.

Kluczowym zagadnieniem było wykorzystanie wydajności jaka daja równoległe obliczenia na karcie graficznej. Dlatego też najważniejsze sa elementy kodu bedace:

- przygotowaniem poczatkowych danych do obliczeń na karcie graficznej,
- funkcjami 'kernel' wykonywanymi na karcie graficznej oraz kodem obsługujacym te funkcje,
- strukturami danych wykorzystanymi do wykonywania obliczeń oraz zapisu wyników obliczeń dotyczacych pozycji i predkości ryb.

W tej sekcji zawarłem ogólne opisy za co odpowiadaja najważniejsze pliki w programie. Zajme sie opisem logiki programu przy wyłaczonym trybie CPU. (Tryb CPU wykorzystuje analogiczna logike z trybu GPU.)

2.1 config.cuh

Plik ten zawiera stałe, opisujace domyślne parametry programu oraz co ważniejsze struktury, w które wykorzystywane sa przy obliczeniach pozycji i predkości ryb:

cudaSOA przechowuje dane o pozycji i predkości ryb, oraz dane pomocnicze **fishesGrid** umożliwiajace zaimplementowanie optymalizacji obliczeń (Uniform grid), polegajacej na przegladaniu sasiedztwa każdej ryby tylko dla najbliższych 7 podprzestrzeni 3D oraz przestrzeni, w której aktualnie znajduje sie dana ryba.

fishData przechowuje dane o jednej rybie w trakcie obliczeń na karcie graficznej.

2.2 fishes.cuh

Plik ten zawiera:

- strukture fishesParams z parametrami, od których zależy formowanie sie ławic (edytowalne suwakami w panelu kontrolnym),
- definicje klasy Fishes, w której skumulowana jest wiekszość logiki wyznaczania pozycji i predkości ryb dla wersji CPU i GPU.

2.3 fishes.cu

Główna logika projektu, zawiera funkcje obliczajace dane ryb dla każdej renderowanej sceny oraz generowanie danych poczatkowych.

Konstruktor Fishes alokuje pamieć po stronie karty w strukturze **cudaSOA** pod zmienna dev_soa. Poza tym wyznaczane sa poczatkowe pozycje (losowe punkty niedaleko środka sześcianu) oraz predkości ryb (losowe od -1 do 1).

updateGPU to funkcja wywołujace kernele, sprawdzajaca poprawność ich wykonywania i synchronizujaca działanie tych kerneli (jeden po drugim).

updateGrid1Kernel, thrust::sort_by_key, updateGrid2Kernel funkcje te po stronie karty zajmuja sie obliczeniem pomocniczej struktury uniform grid, dzieki której ryby przypisane sa do podregionów wielkości MIN_CELL_LEN x MIN_CELL_LEN x MIN_CELL_LEN.

updateSOAKernel to najważniejszy kernel, wypełnia on strukture **cudaSOA** danymi ryb dla obecnie renderowanej sceny wykorzystujac policzona wcześniejszymi kernelami pomocnicza strukture przypisujaca ryby do podprzestrzeni. Kernel ten wykorzystuje poniższe funkcje, do obliczenia składowych nowych predkości ryb:

- **fishGroupBehaviourVelocityFactor** - predkość każej ryby jest modyfikowana poprzez jej otoczenie. Wewnatrz tej funkcji wyznaczane sa kolejno:

  - indeksy 7 najbliższych podprzestrzeni oraz podprzestrzeni, w której znajduje sie ryba rozważana przez dany watek, które beda przegladane w celu wyznaczenia wpływu otoczenia ryby na jego nowa predkość. (zmienne cell, xDist, yDist, zDist)
  - z pomoca funkcji **checkFishNeighbourhood** liczony jest wpływ każdej ryby, która znajduje sie w zasiegu widzenia (parametr visibility) powiazanej z watkiem gpu ryby. Wykorzystuje tutaj zasady, zgodnie z którymi wyzaczane sa składowe predkości ryby zależne od sasiedztwa (udostepnione w temacie projektu https://www.red3d.com/cwr/boids/). Z obserwacji symulacji: separation - jak blisko siebie ryby pływaja, alignmnet - jak bardzo spójny kierunek jest obierany przez ławice, cohesion - jak bardzo ryby dobieraja sie w grupy).
  - składowa predkości bedaca ważona suma składowych separation_factor, cohesion_factor, alignment_factor. Wagi zależa od typu ryb (parzyste maja inne niż nieparzyste)

- **wallVelocityFactor** - prosta funkcja dodajaca składowa predkości zależna od odległości do granicy dostepnej przestrzeni. Im bliżej granicy tym bardziej odsuwana bedzie od niej ryba.

- **speedLimit** - nałożenie ograniczeń na wyliczane predkości ryb. (ryba nie może stać w miejscu, ani poruszać sie w niemożliwy do śledzenia sposób)

getFishCell to funkcja pomocnicza wyznaczajaca podprzestrzeń, w której znajduje sie ryba na podstawie jej obecnej lokalizacji w przestrzeni.

Destruktor Fishes zwalnia zaalokowana pamieć na karcie przy zakończeniu działania programu.

2.4 fishes_model.cuh

Plik zawiera definicje klasy FishesModel odpowiadajacej za wyrenderowanie ryb w oparciu o pozycje i predkości policzone w scenie wykorzystujac logike klasy Fishes.

- **struct cudaGraphicsResource* resource_vbo** pozwala uzyskać dostep do bufora openGL z poziomu kerneli cuda.

- **glm::mat4* dev_models** dynamiczna tablica, na której zapisane beda traslacje i rotacje modelu każdej ryby, jej zawartość jest liczona kernelem setModelsKernel

2.5 fishes_model.cu

Plik zawiera funkcje:

render wywołujaca kernel setModelsKernel z danymi ryb przekazanymi ze sceny.

setModelsKernel na podstawie przekazanych danych ryb ze sceny równolegle oblicza macierze przenoszace każdy model (czyli piramidke) ryby w odpowiednie miejsce w przestrzeni i obracajace każydy model w kierunku wyznaczanym przez predkość.

2.6 main_program.cu

bazowa konfiguracja cuda, uruchomienie głównej petli programu.

3 Pozostałe pliki (nie dotycza cuda)

Pozostałe pliki nie dotycza obliczania danych dla ryb dla każej renderowanej sceny. Po otwarciu projektu pliki beda pogrupowane filtrami w Solution Explorerze:

control petla główna programu (mainLoop) i obsługa klawiatury i myszy oraz konfiguracja zewnetrznych zasobów.

imgui zewnetrzne pliki niezbedne do działania panelu z suwakami.

model poza plikami fishes.* zawarta jest tutaj logika kamery (zewnetrzny zasób), logika panelu z suwakami, pliki zwiazane z rysowaniem obramowania sześcianu (aquarium i cube) oraz scene (zbiera elementy sceny, by móc je aktualizować jedna funkcja w mainLoop)

shaders konfiguracja wyświetlania modeli w oknie (openGL)

utils pliki zewnetrzne oraz obsługa błedów

view pliki odpowiadajace na to w jaki sposób renderowane sa poszczególne elementy sceny w oknie (najważniejsze fishes_model)