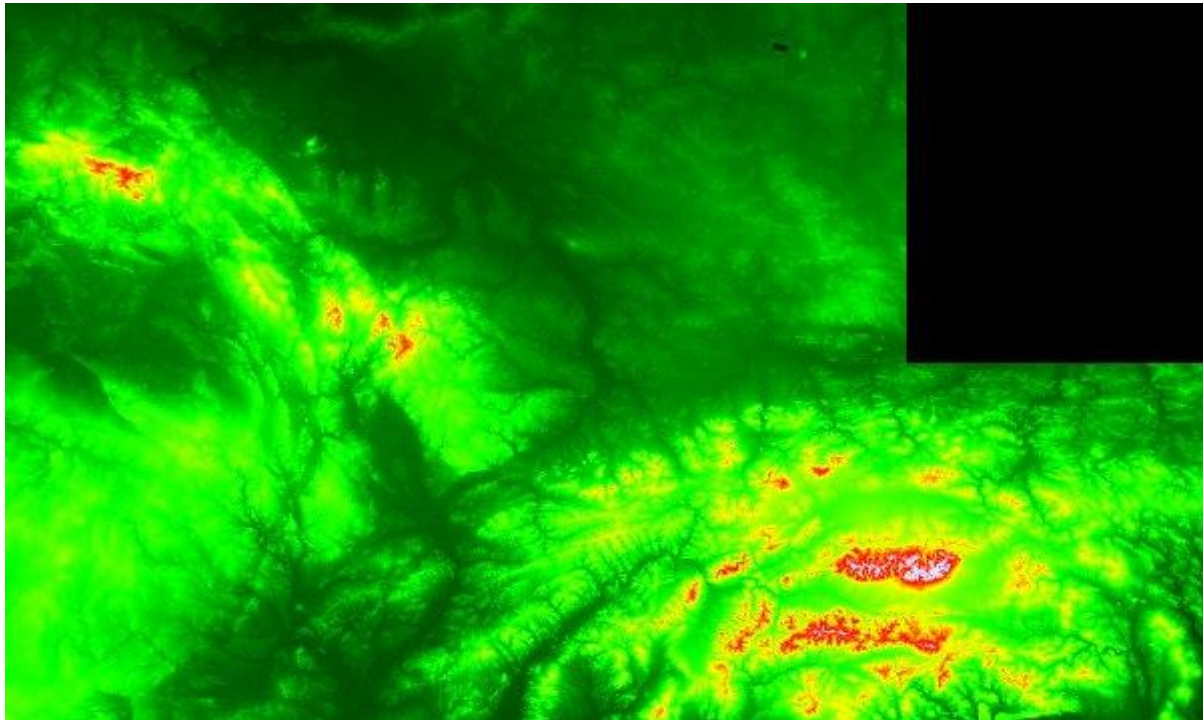


Zadanie 5 - Teren (Termin: 8/13.I.2020) 30 punktów :

Celem zadania jest jak najdokładniejsze i najszybsze wyświetlenie rzeczywistego terenu na podstawie dostępnych map wysokości DTED (digital terrain elevation data). Program powinien wypisywać ilość renderowanych trójkątów i ilość klatek na sekundę. Dla najszybszych programów zostaną przyznane punkty, w punktacji uwzględniona także będzie estetyka, interfejs użytkownika i użyteczność programu. Program ma umożliwiać rysowanie mapy jak i latanie nad terenem 3D zdefiniowanym na powierzchni kuli ziemskiej. Np. tak jak poniżej gdzie widać dolinę Odry, na lewo Sudety i Kotlina Kłodzka, na prawo Tatry, a u góry po prawej dziura koło Bełchatowa:



WCZYTYWANIE DANYCH DTED:

Rzeźba terenu ma być czytana z danych SRTM pochodzących z misji promu kosmicznego Endeavour z 2000 roku. Dane te zawierają wysokości powierzchni ziemi co kilkadziesiąt metrów. Program ma wczytywać dane w postaci binarnych plików **hgt** bez nagłówka (1201x1201 liczb całkowitych 2 bajtowych) dla każdego stopnia kwadratowego. Bajty w każdej liczbie z pliku hgt są w odwrotnej kolejności niż na architekturze Intel'a więc trzeba je zamienić. Każda liczba opisuje wysokość podaną w metrach z zakresu [-500,9000] inne wartości oznaczają brak danych. Format plików **hgt** z danymi SRTM3 których będziemy używać jest opisany [TU](#), pozostała dokumentacja w tym samym [katalogu](#).

Inne wersje danych to np. pliki geotiff 6000x6000 dla regionów 5x5 stopni: [CGIAR-CSI SRTM 90m](#).

Katalog z zipowanymi plikami z danymi hgt: [katalog Eurasia](#)
na przykład:

- [Kotlina Kłodzka, Śleza](#)
- [Karkonosze, Jelenia Góra](#)
- [Alpy, Mont Blanc](#)
- [Alpy, Matterhorn](#)

Program uruchamiamy z co najmniej jednym parametrem: katalogiem i ma on wczytywać wszystkie istniejące pliki z danymi z tego katalogu np. `./data/` dla Dolnego Śląska. Program możemy uruchomić z dodatkowymi parametrami podając zakres szerokości i długości geograficznych w całkowitych stopniach jak np.

```
program ./data/ -sz 50 51 -dl 16 17
```

Wtedy ładowane są wszystkie dostępne kafelki z danymi z podanego zakresu z katalogu, czyli w tym przykładzie jest to jeden kafelek. Może brakować plików w katalogu, wtedy wyświetlamy tylko te które załadujemy. Nazwy plików są zgodne z nazwami na serwerze (po rozpakowaniu zipów), dla ujemnych szerokości nazwa zaczyna się od S zamiast N i podobnie W/E, a rozszerzenie składa się z małych liter `.hgt`.

WYŚWIETLANIE MAPY:

Program powinien po starcie wyświetlić mapę zawierającą prostokąt z wszystkimi wczytanymi danymi przy pomocy rzutu równoległego płaskiej powierzchni z odpowiednio dobranymu proporcjami x-y dla środka wyświetlanego obszaru.

Każdy obszar 1x1 stopień ma rozmiary x-owe zależne od y-szerokości geograficznej: $\sim \cos(y)$ tzn. można zastosować najprostsze rozwiązanie czyli ta sama proporcja dla wszystkich kafelków policzona dla środka widoku lub środka obszaru ładowanych kafelków, można też zastosować odwzorowanie Mercatora lub jego uproszczenie.

Umożliwiamy sterowanie widokiem przy pomocy klawiatury/myszy: przesuwanie się i zoomowanie. Kolory wierzchołków dobieramy według wysokości jak na mapie fizycznej np.

```
if      (ht < 0 )   color = vec3(0.,    0.,    1.); //blue
else if (ht < 500) color = vec3(0.,    ht/500,  0.); //->green
else if (ht < 1000) color = vec3(ht/500-1, 1.,    0.); //->yellow
else if (ht < 2000) color = vec3(1.,    2.-ht/1000,0.); //->red
else      color = vec3(1.,    ht/2000-1 ,ht/2000-1); //->white
```

Warto przemyśleć jakie dane wysyłamy do vertex shadera. Można np. przyjąć że dane wierzchołka to 3 liczby short: indeksy x,y oraz wysokość. Vertex shader powinien wtedy dla każdego kawałka na podstawie przekazanych zmiennych uniform (z informacją który to kwadrat: stopnie, itd.) obliczać współrzędne wierzchołka i jego kolor jak wyżej.

Renderowanie siatki warto zrobić w trybie indeksowanym (`glDrawElements` zamiast `glDrawArrays`) pamiętając o użyciu VBO. Należy mieć kilka tablic indeksów (wspólne dla wszystkich kawałków) umożliwiających renderowanie w różnych rozdzielczościach: pełnej, co drugi element, co czwarty,... czyli prosty **LOD** (Level Of Detail), którego zmianę umożliwiamy użytkownikowi (np klawisze 1,2,...,0-auto), tryb automatyczny gdzie LOD dobierany jest do **FPS** (ilości wyświetlanych klatek na sekundę) tak aby było ich więcej niż 10. Ilość FPS wyświetlamy na bieżąco (może być na terminalu tekstowym). Warto przemyśleć ilość używanej pamięci tak aby umożliwić jak najszybsze wyświetlanie jak najdokładniejszego terenu.

WIDOK 3D ZIEMA JEST KULĄ:

Widok z perspektywy obserwatora poruszającego się nad powierzchnią. Zakładamy że ziemia jest sferą o wiadomym promieniu (wiki) więc trzeba odpowiednio wyliczać prawdziwe współrzędne xyz (napisać inny vertex shader, który będzie obliczał współrzędne w globalnym układzie współrzędnych ze środkiem w środku Ziemi) i zmodyfikować obsługę klawiatury/myszy tak aby umożliwiała latanie na stałej (ale zmienianej z klawiatury) wysokości nad ziemią, wygodne zmiany położenia, kierunku patrzenia. Dla obszarów ziemi dla których nie ładujemy danych wyświetlamy kształt sfery schematycznie np. przy pomocy linii ograniczających obszary 1x1 czy 5x5 stopni.