

Problem 8: Renderowanie miejsca fiksacji oka

Punkty: 15

Autor: Gary Hoffmann, Denver, Kolorado, Stany Zjednoczone

Wprowadzenie do problemu

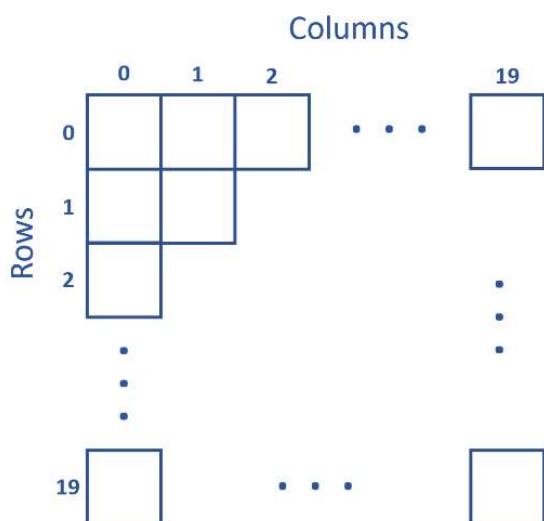
W ostatnich pięciu latach Wirtualna Rzeczywistość wkroczyła z przytupem na rynek, znajdując zastosowanie w wielu dziedzinach, od gier i rozrywki przez projektowanie wyrobów do inżynierii. Jednym z najnowszych osiągnięć w projektowaniu zestawów VR jest dodanie funkcji śledzenia oka, która rozbudowuje działanie zestawu.

Charakterystyczną cechą ludzkiego wzroku jest fakt, że pole widzenia, w jakim występuje pełna ostrość a także możliwe jest rozpoznanie drobnych szczegółów jest bardzo wąskie. Ostrość widzenia jest zależna od dołka środkowego siatkówki oka (ang. fovea), małego wgłębienia w siatkówce oka, które specjalizuje się w tym zakresie. Jednak ze względu na rozmiar dołka środkowego ludzkie oko postrzega otoczenie wystarczająco wyraźnie jedynie w polu widzenia o szerokościątowej poniżej 10° . Resztą zajmuje się nasz mózg, który zapełnia puste miejsca obrazami rejestrowanymi przez nas przy rozglądaniu się.

Z tego względu zestaw VR musi renderować obrazy o najwyższej rozdzielczości jedynie w tym miejscu, w które patrzy użytkownik. Obrazy poza tym polem widzenia mogą być renderowane z mniejszą jakością, co poprawia wydajność systemu.

Opis problemu

Waszym zadaniem jest napisanie modułu do aplikacji Wirtualnej Rzeczywistości, który będzie określać jakość renderowania dla każdego fragmentu ekranu zestawu VR. Dla uproszczenia, wasz moduł będzie uwzględniał jedno oko dla pojedynczego ekranu. Ekran będzie podzielony na siatkę o długości boku wynoszącej 20 pól.



Do waszego programu otrzymacie współrzędne pola umieszczonego na siatce, na którym aktualnie użytkownik skupił swój wzrok, a wynik powinien zawierać poziom renderowania dla każdego pola siatki, wiersz po wierszu.

Pole, w które aktualnie patrzy użytkownik powinno być renderowane z pełną jakością czyli 100%. Wszystkie pola graniczące z polem określonym powyżej powinny być renderowane z połowiczną jakością (50%), natomiast pola graniczące z nimi powinny

być renderowane z jakością 25%. Cała reszta powinna być renderowana na minimalnym poziomie 10%.

Na przykład, jeśli użytkownik patrzy w pole w wierszu 7 i kolumnie 10, to jakość renderowania każdego pola siatki będzie następująca:

	<i>Kol</i>	0	...	7	8	9	10	11	12	13	...	19
<i>Rząd</i>												
0		10%	...	10%	10%	10%	10%	10%	10%	10%	...	10%
⋮		⋮	↘	⋮	⋮	⋮	⋮	⋮	⋮	⋮	↗	⋮
4		10%	...	10%	10%	10%	10%	10%	10%	10%	...	10%
5		10%	...	10%	25%	25%	25%	25%	25%	10%	...	10%
6		10%	...	10%	25%	50%	50%	50%	25%	10%	...	10%
7		10%	...	10%	25%	50%	100%	50%	25%	10%	...	10%
8		10%	...	10%	25%	50%	50%	50%	25%	10%	...	10%
9		10%	...	10%	25%	25%	25%	25%	25%	10%	...	10%
10		10%	...	10%	10%	10%	10%	10%	10%	10%	...	10%
⋮		⋮	↗	⋮	⋮	⋮	⋮	⋮	⋮	⋮	↘	⋮
19		10%	...	10%	10%	10%	10%	10%	10%	10%	...	10%

Przykładowe dane wejściowe

Pierwszy wiersz danych wejściowych waszego programu, **otrzymanego przez standardowe wejście**, będzie zawierać dodatnią liczbę całkowitą oznaczającą liczbę przypadków testowych. Każdy przypadek testowy będzie zawierać pojedynczy wiersz składający się z dwóch liczb całkowitych oddzielonych spacjami, które będą reprezentować numer rzędu i numer kolumny pola siatki, na którym użytkownik skupia swój wzrok. Numery rzędów i kolumn mieszczą się w przedziale od 0 do 19 włącznie.

```
2
7 10
0 0
```

Przykładowe dane wyjściowe

W każdym przypadku testowym wasz program powinien wyświetlić procentową wartość jakości renderowania dla każdego pola siatki. Każdy rząd powinien mieścić się w oddzielnym wierszu, a kolumny powinny być oddzielone spacjami.

[illegible]