



**Uniwersytet Bielsko-Bialski**



**Grafika Komputerowa  
(Ćwiczenia laboratoryjne)**

**Temat ćwiczenia: Zadanie WebGL  
(lab11)**

**Wykonał:**

**Wojciech Świerczyński**

## Polecenie:

Przerobić plik lab11.html tak aby:

Kwadraty odpowiadają również myszą: Jeśli klikniesz lewym przyciskiem myszy lub klikniesz lewym przyciskiem myszy i przeciągniesz na płótnie, cały kwadrat będzie kierowany w stronę pozycji myszy. Jeśli klikniesz lewym przyciskiem myszy, dane punktów zostaną ponownie zainicjowane, więc zaczną się od środka. Możesz wstrzymać i ponownie uruchomić animację, naciskając spację.

Będziesz modyfikował kod modułu shadera i kod JavaScript, aby zaimplementować kilka różnych stylów dla prymitywu punktu.

Użytkownik będzie kontrolował program, naciskając klawisze na klawiaturze

Pierwsze ćwiczenie polega na umożliwieniu przypisania innego koloru do każdego kwadratu. Ponieważ kwadraty są naprawdę wierzchołkami w pojedynczym prymitywie typu `gl.POINTS`, można użyć zmiennej atrybutu dla koloru. Atrybut może mieć inną wartość dla każdego wierzchołka.

Pierwszym zadaniem jest dodanie zmiennej kolorowej typu `vec3` do modułu shadera wierzchołka i użycie wartości atrybutu do pokolorowania kwadratów. Będziesz także musiał pracować po stronie JavaScript. Będziesz potrzebował `Float32Array` do przechowywania wartości kolorów po stronie JavaScript, a będziesz potrzebował bufora WebGL dla tego atrybutu. Program ma już jeden atrybut, który jest używany do współrzędnych wierzchołków. Będziesz robił coś podobnego do atrybutu `color` (poza tym, że możesz to zrobić w `initGL()`, ponieważ wartości kolorów nie zmieniają się po ich utworzeniu). Można użyć

losowych wartości w zakresie od 0,0 do 1,0 dla składników koloru.

Gdy tablica atrybutów jest włączona, każdy wierzchołek otrzymuje swój własny kolor z buforu atrybutów. Gdy tablica atrybutów jest wyłączona, wszystkie wierzchołki otrzymują ten sam kolor, a tę wartość można ustawić za pomocą rodziny funkcji `gl.vertexAttrib *`.

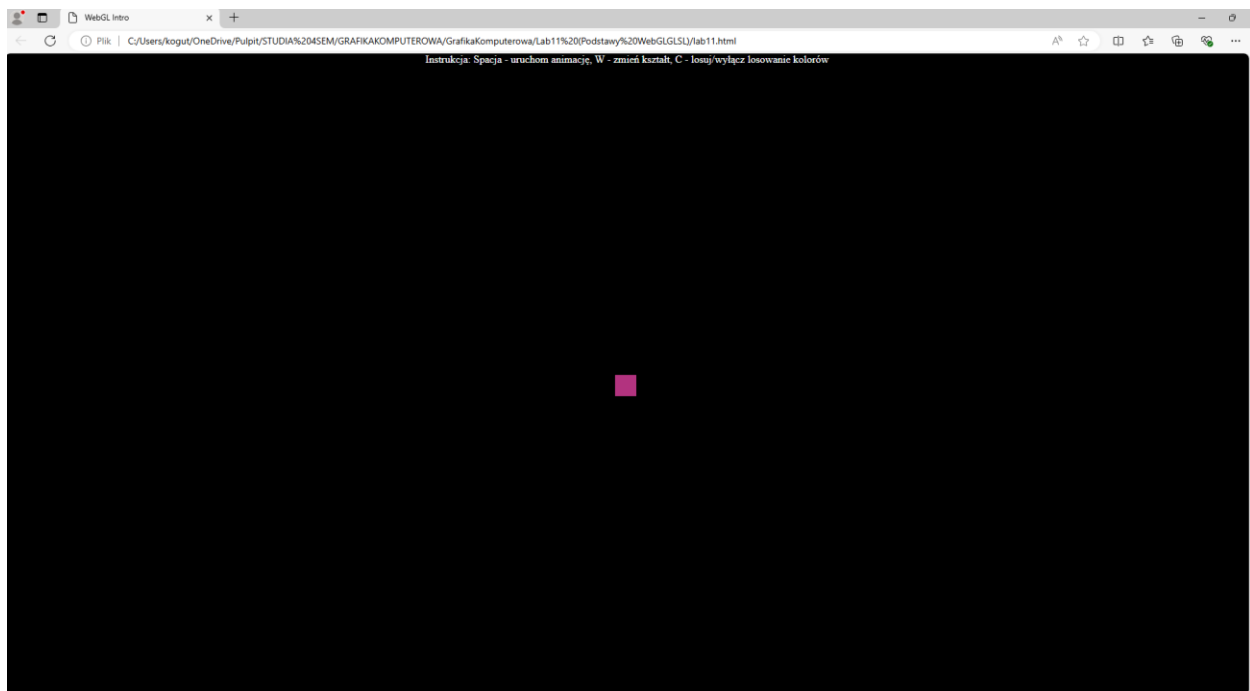
Pozwól użytkownikowi na naciśnięcie określonego klawisza, aby włączyć lub wyłączyć losowe kolory. Program ma funkcję `doKey()`, która jest już skonfigurowana do reagowania na wprowadzanie z klawiatury. Będziesz dodawać do programu kilka typów interakcji z klawiaturą. Aby odpowiedzieć na klawiszę, musisz znać numeryczny kod klawiszy.

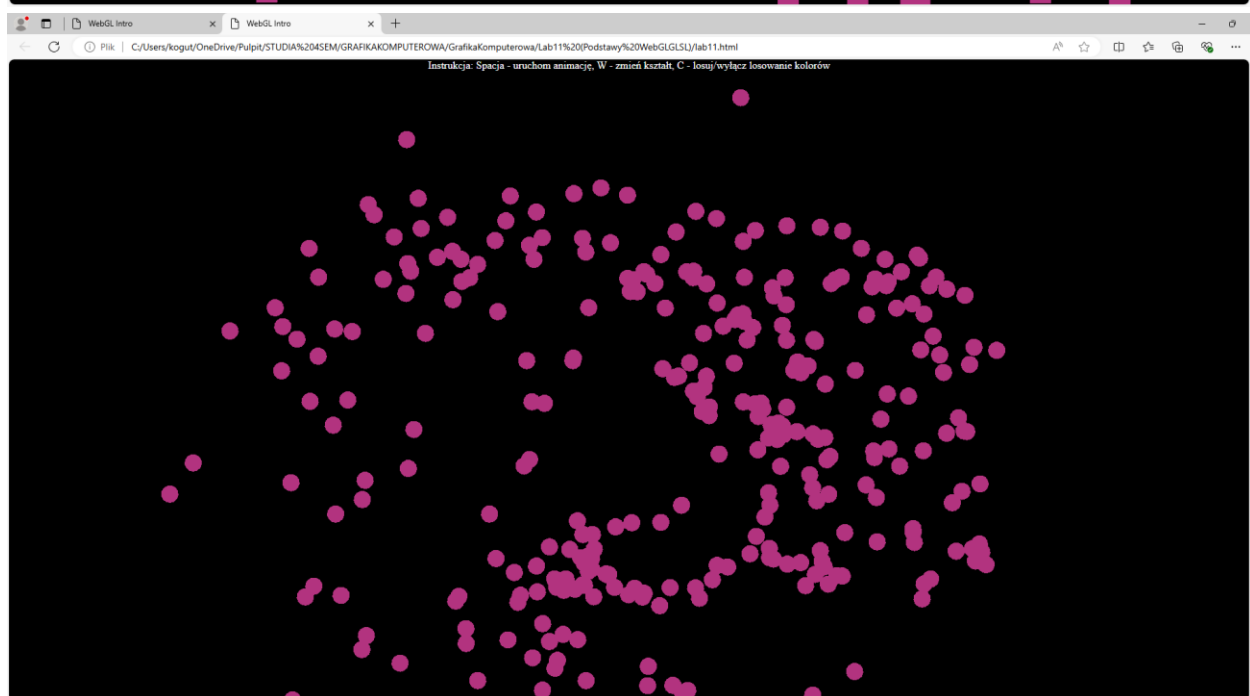
Powinieneś dodać opcję używania stylu wyświetlania dla punktów w postaci wielokąta. Pozwól użytkownikowi wybrać styl za pomocą klawiatury; na przykład, naciskając klawisze numeryczne.

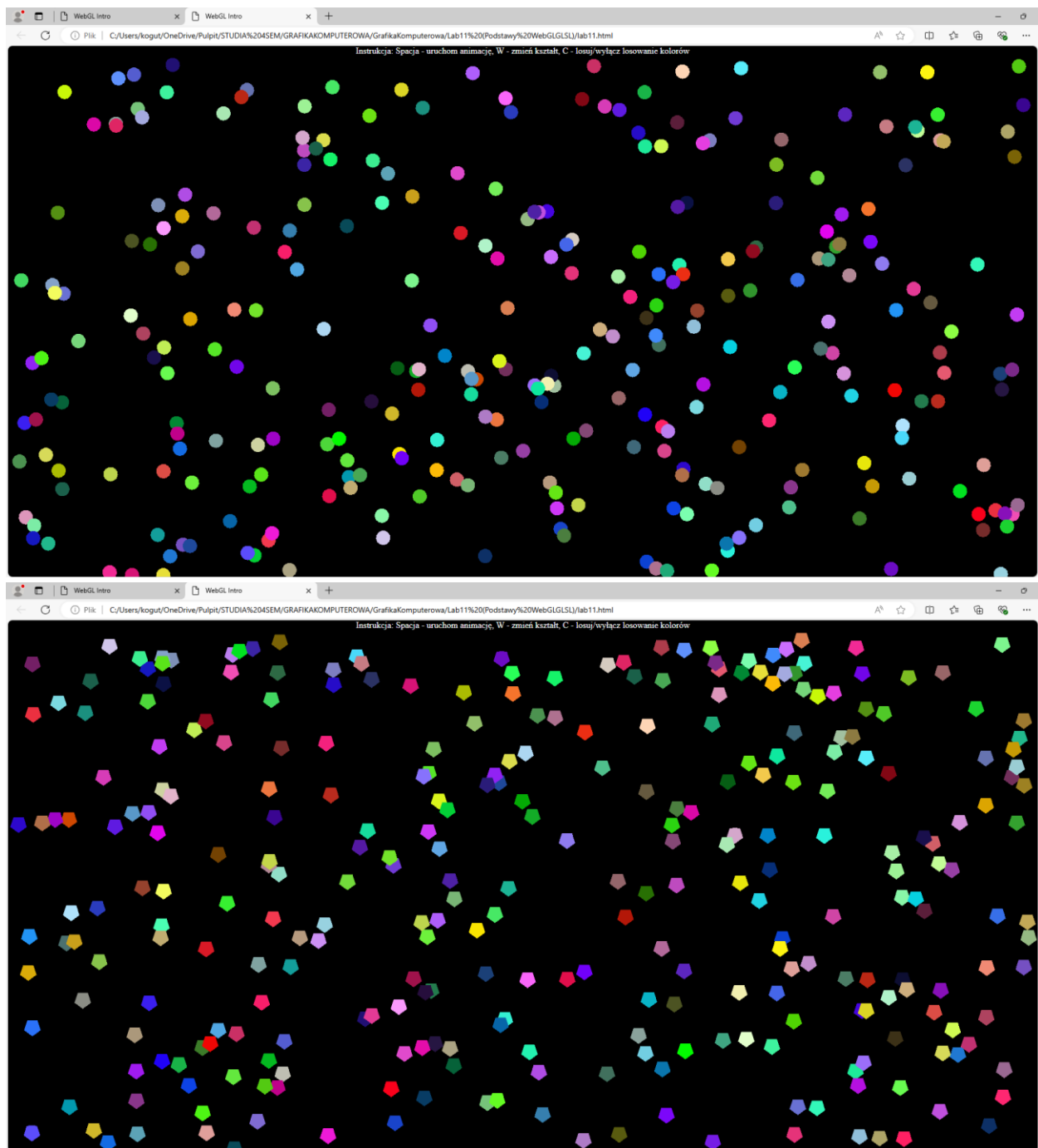
## **Wykorzystane komendy:**

Kod znajduje się w pliku zip w którym zawarte jest to sprawozdanie

## **Wynik działania:**







**Wnioski:**

Dzięki temu ćwiczeniu nauczyliśmy się podstaw oraz zaawansowanych rzeczy w WebGL, min takie jak ustawianie losowych kolorów dla obiektów, podążanie obiektów za lewym przyciskiem myszy, zmiana wierzchołków obiektów, odbijania się ich od krawędzi ekranu, kontrola całego programu za pomocą użycia klawiszy myszy i klawiatury