

**Laboratorium: Lista zadań 2**  
**termin ~~2019-12-08~~ 2019-12-15**

**1. Testowanie WebGL (10 p.)**

Napisz prosty program, który demonstruje rysowanie w WebGL obiektów geometrycznych: POINTS, LINE\_STRIP, LINE\_LOOP, LINES, TRIANGLE\_STRIP, TRIANGLE\_FAN, TRIANGLES.

Zastosować zmienne uniform do ustalania pewnych parametrów rysowania (np. kolorów, przekształceń geometrycznych itp.) Należy także:

- a. wypisać w konsoli lub na stronie listę aktywnych atrybutów, zgodnie z przykładem na stronie <https://developer.mozilla.org/en-US/docs/Web/API/WebGLRenderingContext/getActiveAttrib> ,
- b. analogicznie wykorzystać `getActiveUniform` aby wypisać listę zmiennych uniform.
- c. zademonstrować działanie `bindAttribLocation` aby przypisać zmienne *attribute variable* do ustalonych przez użytkownika indeksów *generic vertex index*.

**2. Krzywa Hilberta (15 p.)**

Wygeneruj zbiory odcinków składających się na krzywe Hilberta różnego stopnia, umieść odpowiednie dane w buforach i zaimplementuj shadery rysujące te dane w kontekście webgl. Napisz interakcyjny program, który rysuje krzywe w różnych kolorach na różnych głębokościach (współrzędna z), tak aby krzywe o większych głębokościach stanowiły tło krzywej o najmniejszej głębokości i umożliwia przy pomocy klawiszy lub przycisków operacje takie jak przesuwanie krzywych, zmiany głębokości, zmiany kolorów itp ...

**3. Gra komputerowa 2D (20 p.)**

Zaimplementować w WebGL prostą dwu-wymiarową grę zręcznościową typu: [Pong](#), [Arkanoid](#), [Space Invaders](#), lub inną podobną, ale w taki sposób aby tło zawierało jakiś rysunek utworzony z elementów geometrycznych (np. boisko). Wykorzystaj bufor głębokości tak aby elementy pierwszoplanowe były rysowane z mniejszą współrzędną Z niż elementy tła. Zaimplementuj animację w zalecany sposób jak w przykładowym programie:

<http://156.17.7.16/public/dydaktyka/kik/grafika/animation/animation.html>

Zwróć uwagę, że callback animacji jest wywoływany w określonych punktach czasowych zależnych od częstotliwości wyświetlania obrazu na ekranie, natomiast zdarzenia istotne dla stanu gry, takie jak np. zderzenie obiektów, mogą wystąpić między tymi punktami czasowymi. Zadbaj o poprawną implementację takich sytuacji. Skoncentruj się na graficznej prezentacji gry. Obliczanie punktacji i gromadzenie statystyk dotyczących wyników nie jest istotne.

**4. Tekstury (10 p.)**

Zaprojektuj i ponakładaj tekstury na ruchome obiekty i elementy tła zaimplementowanej przez siebie gry z poprzedniego zadania.