

# Aplikacje WWW

## Lista 4 - 2019

1. Zmodyfikuj przykładowy plik `6b.piłeczki.html` z wykładu 2 umieszczając wewnątrz elementów `<b>` elementy `<img>` tak, by zamiast strzałek po ekranie poruszały się samoloty, ptaki, owady lub inne obiekty wg Twojego uznania. Zmodyfikuj skrypt i style tak, aby obiekty nie wychodziły poza obramowanie. Zmień również rozmiar sceny oraz liczbę obiektów.

Przeczytaj na temat `css {transform:rotate(30deg)}`. Zauważ polecenie `b[i].style.transform="rotate("+Math.atan2(b[i].vy,b[i].vx)+"rad")`, które sprawiało, że strzałki były zawsze ustawione w kierunku ruchu. Zaktualizuj je (dodając odpowiedni kąt), by przez ciebie dodane obiekty (samoloty/ ptaki itp) zawsze były skierowane w kierunku przemieszczania się.

2. (3pkt) Dodaj do poprzedniego zadania paragraf, w którym będzie widoczna aktualna liczba obiektów poruszających się w ramce, oraz obsługę zdarzenia `onmousedown`, która spowoduje zatrzymanie (albo zniknięcie) klikniętego obiektu, oraz zmniejszenie liczby widocznej w paragrafie. Po zatrzymaniu (zniknięciu) ostatniego obiektu w paragrafie powinien pokazać się napis „Mission completed in ” oraz czas jaki upłynął od kliknięcia guzika „start”. Dodaj guzik „Nowa gra”, który spowoduje ponowne pokazanie wszystkich obiektów, oraz nadanie im nowych położeń i prędkości.

Dodaj przyciski „Poziom 0” „Poziom 1”, itd. które będą działały tak jak „Nowa gra” ale dodatkowo będą zmieniać poziom trudności gry, czyli liczbę i prędkości poruszających się obiektów. Wskazówka: prędkość zależy do drugiego argumentu funkcji `setTimeout`.

3. W pliku `7.binary.html` znajduje się gra “Binary”. Naucz się wygrywać w tą grę. Napisz jaki jest związek wygrywającej sekwencji ruchów z zapisem docelowej liczby w systemie dwójkowym. Zmodyfikuj grę tak, by działała dla systemu trójkowego (w każdym ruchu mnożymy przez 3 a potem opcjonalnie dodajemy 1 lub 2). Startujemy od 0.

4. Dana jest tablica obiektów:

```
var student=[ {imie:"Ola",nazwisko:"Lis",kierunek:"fizyka"},
               {imie:"Ala",nazwisko:"Góral",kierunek:"chemia"},
               {imie:"Jan",nazwisko:"Nowak",kierunek:"fizyka"}];
```

Dopisz do niej kilkunastu studentów z różnych kierunków. Napisz funkcję:

- (a) `function wszyscy()`, która w elemencie `<div>` o `id="lista"` umieści nagłówek `<h2>Studenti</h2>` oraz listę `<ol>` wszystkich studentów w formacie ”imię nazwisko (kierunek)”.
- (b) `function pokaz(kierunek)`, która w elemencie `<div>` o `id="lista"` umieści element `<h2>` z nazwą kierunku oraz wyliczenie `<ol>` z imionami i nazwiskami studentów podanego kierunku.

Wypróbuj działanie funkcji za pomocą kilku przycisków podobnych do:

```
<button onclick="wszyscy()">Wszyscy</button>
<button onclick="pokaz('fizyka')">Fizyka</button>
<button onclick="pokaz('chemia')">Chemia</button>
```

5. (2 pkt) Bazując na poprzednim zadaniu napisz funkcję, która policzy, ilu jest studentów na każdym z kierunków oraz:
- (a) stworzy obiekt podobny do `var dane={chemia: 4, fizyka: 5, astronomia: 2}` i wypisze otrzymany wynik w postaci wypunktowania.
  - (b) stworzy obiekt podobny do obiektu `dane` z pliku `4.slupki-F.html` i wywoła funkcję `wykres(dane)`, która narysuje wykres słupkowy ilustrujący ilość studentów na poszczególnych kierunkach. Wykorzystaj funkcję do rysowania wykresów i style zawarte w pliku `4.slupki-F.html`, który znajdziesz w materiałach z wykładu 4. Wcześniej przeanalizuj dokładnie, w jaki sposób one działają.
6. (2pkt) Do pliku z poprzedniego zdania dodaj formularz z polami tekstowymi 'imie', 'nazwisko' oraz polem wyboru 'kierunek' (element `<select>`) i guzikiem 'dodaj'. Po wypełnieniu formularza kliknięcie przycisku 'dodaj' powinno dopisywać osobę do tablicy student, wyświetlać aktualną listę studentów z jej kierunku oraz powodować aktualizację wykresu słupkowego.