**„Zdolni z Pomorza”**

**Laboratorium**

# Część II: Szkielet prostej gry

1. **Dodawanie kanwy oraz elementów graficznych gry.**
2. Stwórz dokument HTML o nazwie gra.html, z jednym elementem: kanwą. Zawartość pliku:

<!DOCTYPE html>  
<html>  
 <body>  
 <canvas id="mojePlotno" width="600" height="600"></canvas>  
 </body>  
</html>

1. Dodaj skrypt, w którym zdefiniujemy działanie gry. Skrypt możesz umieścić na przykład na końcu pliku:

<script>  
</script>

1. W środku skryptu (między linią ze znacznikami początku i linią ze znacznikiem końca) dodaj linie z definicją zmiennych, za pomocą, których będziemy się odwoływać do kanwy.

// zmienne dostępu do kanwy  
var kanwa = document.getElementById('mojePlotno');  
var kontekst = kanwa.getContext('2d');  
  
Następne elementy dodawaj **w kolejnych linijkach skryptu**:

1. Dodaj definicje zmiennych reprezentujących obiekty gry:

// obiekty gry  
var pilka = {

x: 100,

y: 200,

grafika: "pilka.png",

};

var kosz = {

x: 200,

y: 100,

grafika: "kosz.png",

};

1. Dodaj funkcję rysującą obiekty gry:

// Rysowanie  
var rysowanie = function () {   
 obrazek = new Image();  
 obrazek.src = pilka.grafika;  
 kontekst.drawImage( obrazek, pilka.x, pilka.y);

obrazek = new Image();  
 obrazek.src = kosz.grafika;  
 kontekst.drawImage( obrazek, kosz.x, kosz.y);  
};

1. Dodaj funkcję główną programu – main:

// Program główny  
var main = function () {  
 rysowanie();  
}

1. Dodaj wywołanie funkcji głównej programu

//Wywołanie głównej funkcji

main();

**II. Dodawanie reakcji na klawisze.**

1. Dodaj nasłuchiwanie klawiatury, za definicją zmiennych dostępu do kanwy z punktu 3.

// czytanie znaków z klawiatury  
var klawisze = {};  
addEventListener( "keydown", function (e) { klawisze[ e.keyCode] = true; }, false);  
addEventListener( "keyup", function (e) { delete klawisze[ e.keyCode]; }, false);

1. Dodaj funkcję uaktualnienia położenia pilki (za funkcją rysowanie):

var zmiana = function() {  
 if (38 in klawisze) { // klawisz "góra"  
 pilka.y -= 5;  
 }  
 if (40 in klawisze) { // klawisz "dół"  
 pilka.y += 5;  
 }  
 if (37 in klawisze) { // klawisz "lewo"  
 pilka.x -= 5;  
 }  
 if (39 in klawisze) { // klawisz "prawo"  
 pilka.x += 5;  
 }  
}

1. Dodaj do programu głównego wywołanie funkcji zmiana(), za linią z rysowaniem:

zmiana();  
  
W programie głównym są instrukcje rysowania i zmiany współrzędnych obiektów. Niestety po zmianie nic się nie dzieje (sprawdź), – bo nie ma ponownego rysowania kanwy. Dlatego dodajmy prośbę o ponowne wywołanie funkcji main:

Jako ostatnie linie wewnątrz funkcji rysowanie dodaj:

// prośba o następne wywołanie tej samej funkcji  
 requestAnimationFrame(main);

1. Zapisz, sprawdź czy piłka reaguje na klawiaturę.  
     
   Jak widać, piłka porusza się zgodnie z klawiaturą, ale zostawia po sobie ślad, dlatego przed każdym rysowaniem obiektów musimy wyczyścić kanwę.
2. Dodaj czyszczenie kanwy, jako pierwszą linię w funkcji rysowanie():

kontekst.clearRect(0, 0, kanwa.width, kanwa.height);

1. Zapisz, sprawdź.

**III. Dodawanie celu gry.**

1. Niech celem dla piłki będzie zbliżenie się do kosza. W takim przypadku dodamy 1 punkt do punktacji i przeniesiemy kosz w nowe miejsce.
2. Dodaj (przed funkcją main) funkcję odNowa(), w której umieścimy kosz na nowej, losowej pozycji. Funkcję wstaw przed funkcją zmiana(). Math.random losuje liczbę z przedziału <0,1):

var odNowa = function () {  
 // kosz umieszczamy w losowym punkcie  
 kosz.x = Math.random() \* kanwa.width;  
 kosz.y = Math.random() \* kanwa.height;  
}

1. Uzupełnij funkcję zmiana o sprawdzenie, czy piłka jest blisko kosza. Ze względu na to, że trudno byłoby naprowadzić piłkę z dokładnością do 1 piksela, sprawdzamy czy współrzędne różnią się, o co najwyżej 20, jeśli tak, dokonujemy zmian w położeniu obiektów. Poniższy kod wklej, jako ostatnie linie funkcji zmiana:

if (  
 pilka.x <= (kosz.x + 20)  
 && kosz.x <= (pilka.x + 20)  
 && pilka.y <= (kosz.y + 20)  
 && kosz.y <= (pilka.y + 20)  
 ) {  
 odNowa();  
 }

1. Zapisz, sprawdź.  
     
   W następnym kroku dodamy elementy punktacji.
2. Dodaj zmienną, w której zapamiętamy ile razy piłka doleciała do kosza. Zmienną dodaj za zmiennymi obiektów gry, a przed funkcją rysowanie();.

var ileSukcesow = 0;

1. W funkcji zmiana, przed wywołaniem funkcji odNowa(), dodaj linię w nawisach klamrowych:

++ileSukcesow;

1. W funkcji rysowanie() dodaj linie z drukowaniem liczby sukcesów. Kod należy dodać pod linią: kontekst.drawImage( obrazek, kosz.x, kosz.y);.

// Pisanie wyniku  
 kontekst.fillStyle = "#00ff00";  
 kontekst.font = "30px Arial";  
 kontekst.textAlign = "left";  
 kontekst.textBaseline = "top";  
 kontekst.fillText("Wynik: " + ileSukcesow, 10, 10);

1. Zapisz i sprawdź działanie.

**IV. Dodatkowe elementy.**

W następnych krokach możesz poprawić lub rozwinąć szkielet gry na przykład poprzez:

* zmianę położenia piłki i siatki: zadbaj o to, by piłka pojawiała się przed, a nie za piłką;
* dodanie instrukcje warunkowych nie pozwalających na ruch poza obszar kanwy (zatrzymaj piłkę na brzegu kanwy tak, by zawsze było ją widać w całości);
* zmianę położenia siatki tak, by zawsze było ją widać w całości;
* wydzielenie na kanwie części do wyświetlenia wyniku, na której piłka się nie porusza;
* dodanie nowych elementów, takich jak przeszkadzające ściany (piłka nie powinna przez nie przelatywać);  
  **Uwaga!** Przy dodaniu dużych elementów, należy dodać również rozpoczęcie gry tylko jeśli elementy się wczytały, przykład – w następnym zadaniu.
* dodanie elementów niszczących piłkę (na przykład rozrzuconych na planszy gwoździ);
* dodanie zliczania czasu gry;

**Pomoc:**

//ta instrukcja odczytuje czas bieżący

var czas = Date.now();

//ta instrukcja oblicza różnicę czasu w sekundach między czasem czas1 i czasem czas2

var czas = (czas1 – czas2)/1000;

* dodanie pewnej „bezwładności” piłki, tak, aby po puszczeniu klawisza strzałki piłka nadal przez chwilę poruszała się w danym kierunku

**Szkic rozwiązania:** w tablicy klawisze musielibyśmy zapisywać nie prawda lub fałsz, jak obecnie, ale „natężenie” ruchu w danym kierunku, czyli liczbę, która po przyciśnięciu klawisza byłaby zwiększana do pewnej wartości maksymalnej, a po puszczeniu klawisza stopniowo malałaby do 0.