

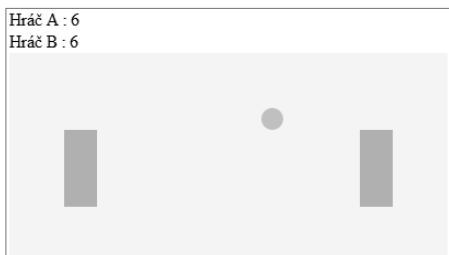
Začínáme s vývojem her a DOM

V předchozí kapitole jsme získali představu o tom, co nás v této knize čeká a co se dozvíme. Počínaje touto kapitolou zahájíme naše seznamování s problematikou na praktických příkladech. Ještě než se dostaneme k pokročilým tématům, jako jsou animace CSS3 nebo plátno HTML5, podíváme se na vývoj tradičních her využívajících DOM. Kapitulu zahájíme krátkým seznámením s nezbytnými základy.

V této kapitole:

- Připravíme si nástroje potřebné pro vývoj her.
- Vytvoříme první hru – Ping Pong.
- Seznámíme se se základy pozicování pomocí knihovny jQuery.
- Dozvíme se, jak načítat vstup z klávesnice.
- Vytvoříme hru Ping Pong, včetně zobrazení skóre hráčů.

Na následujícím obrázku můžete vidět hru, kterou v této kapitole vytvoříme. Jedná se o hru Ping Pong pro dva hráče, ovládanou oběma hráči prostřednictvím jedné klávesnice.



Pojďme si tedy zahrát Ping Pong.

Příprava vývojového prostředí

Prostředí, které je zapotřebí pro vývoj her v HTML5, se příliš neliší od toho, které je třeba pro vývoj webových stránek. Budeme potřebovat moderní webový prohlížeč a dobrý textový editor. O tom, který textový editor je nejlepší, se již od nepaměti vedou vášnivé debaty. Každý textový editor má své silné stránky, a proto si vyberte ten, který vám vyhovuje nejlépe. Co se prohlížeče týká, budeme potřebovat prohlížeč podporující nejnovější specifikaci HTML5 a CSS3 a nabízející nástroje pro ladění kódu.

Na Internetu najdete hned několik vyhovujících moderních webových prohlížečů. Jedná se o prohlížeče Apple Safari (<http://apple.com/safari/>), Google Chrome (<http://www.google.com/chrome/>), Mozilla Firefox (<http://mozilla.com/firefox/>) a Opera (<http://opera.com>). Uvedené prohlížeče podporují většinu funkcí, o kterých bude v průběhu této knihy řeč. K prezentaci většiny příkladů v této knize poslouží prohlížeč Google Chrome, protože je velmi rychlý při práci s přechody CSS3 a plátnem HTML5.

Příprava dokumentů HTML pro hru

Každý web, webová stránka a hra v HTML5 začíná u výchozího dokumentu HTML. Tento dokument obsahuje základní kód HTML. Vývoj naší první hry v HTML5 začneme právě u takového dokumentu, s názvem `index.html`.

Jak na to – instalace knihovny jQuery

Hru Ping Pong vytvoříme zcela od základů. To však neznamená, že bychom museli vše programovat sami. Pomůže nám **knihovna v JavaScriptu** s názvem **jQuery**, se kterou se budeme setkávat napříč celou touto knihou. Díky ní se značně zjednoduší náš kód v JavaScriptu:

1. Vytvořte novou složku s názvem `pingpong`.
2. Vytvořte ve složce `pingpong` novou podsložku s názvem `js`.
3. Nyní si stáhneme knihovnu jQuery. Zamířte na adresu <http://jquery.com/>.
4. Zatrhněte pole **Production** a klepněte na tlačítko **Download jQuery**.
5. Uložte soubor (aktuálně `jquery-1.6.min.js`) do složky `js` vytvořené v kroku 2.
6. Vytvořte nový dokument HTML s názvem `index.html` a uložte ho do adresáře `pingpong`.
7. Otevřete soubor `index.html` v textovém editoru a vložte do něj základní kostru dokumentu HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
```

```

<title>Ping Pong</title>
</head>
<body>
  <header>
    <h1>Ping Pong</h1>
  </header>
  <footer>
    Příklad ukazující vytvoření hry Ping Pong.
  </footer>
</body>
</html>

```

8. Připojte k dokumentu knihovnu jQuery přidáním následujícího řádku kódu na konec elementu body:

```
<script src="js/jquery-1.6.min.js"></script>
```

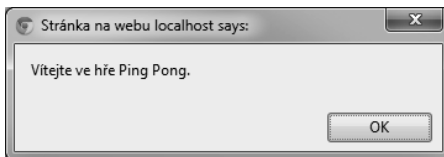
9. Zbývá ověřit úspěšné načtení knihovny jQuery. Za tímto účelem vložte následující kód na konec elementu body, za kód načítající knihovnu jQuery:

```

<script>
$(function(){
  alert("Vítejte ve hře Ping Pong.");
});
</script>

```

10. Uložte soubor `index.html` a otevřete jej ve webovém prohlížeči.



Vysvětlení

Právě jsme vytvořili jednoduchý dokument HTML5, připojili k němu knihovnu jQuery a ověřili, že se korektně načte.

Doctype pro HTML5

Elementy DOCTYPE a meta se v HTML5 dočkaly zjednodušení.

Takto vypadá deklarace DOCTYPE v HTML 4.01:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

```

Poměrně složité, co říkáte? Naproti tomu deklarace `DOCTYPE` v HTML5 nemůže být jednodušší:

```
<!DOCTYPE html>
```

Deklarace dokonce neuvádí ani verzi HTML. To znamená, že HTML5 podporuje všechny předchozí verze jazyka HTML a budoucí verze jazyka HTML budou zase podporovat obsah v jazyce HTML5.

Zjednodušení se týká také elementu `meta`. Deklarace znakové sady dokumentu HTML je nyní snazší:

```
<meta charset=utf-8>
```

Záhlaví a zápatí

Jazyk HTML5 přichází s řadou nových funkcí a vylepšení. Jedním z nich jsou i změny v oblasti **sémantiky**. HTML5 zavádí nové elementy, které sémantiku posilují. Dva z nich jsme už použili – `header` a `footer`. Element `header` uvozuje sekci nebo celou stránku. Z toho důvodu jsme vložili element `h1` právě sem. Element `footer`, jak už jeho název napovídá, obsahuje zápatí sekce, resp. celé stránky.



UPOZORNĚNÍ

Sémantikou HTML se rozumí, že kód samotný poskytuje srozumitelné informace o obsahu namísto toho, aby jen definoval jeho vzhled.

Kam umístit kód v JavaScriptu

Kód v JavaScriptu jsme umístili těsně před koncovou značku elementu `body`, za veškerý obsah stránky. Umístění sem namísto do elementu `head` má svůj důvod.

Za normálních okolností webové prohlížeče načítají a vykreslují obsah stránky ve směru shora dolů. Pokud se v elementu `head` dokumentu nachází nějaký kód v JavaScriptu, načte se obsah dokumentu až po provedení tohoto kódu. Vykreslování a načítání se pozastaví, narazí-li prohlížeč na kód v JavaScriptu uprostřed stránky. To je důvod, proč vkládat kód v JavaScriptu až na konec stránky, pokud to situace dovozuje. Stránka tak uživatelům nabídne rychlejší odezvu.

V době vzniku této knihy byla aktuální verze jQuery 1.4.4. Proto se předchozí ukázka kódu odkazuje na soubor `jquery-1.6.min.js`. V době, kdy tyto řádky budete číst, bude pravděpodobně aktuální jiná verze s odlišným označením, její použití by však mělo být totožné, leda by došlo k nějaké zásadní změně v jQuery, která by nebyla zpětně kompatibilní.

Provedení kódu po načtení stránky

Potřebujeme nějak zajistit, aby se náš kód v JavaScriptu provedl až poté, co bude stránka načtená. V opačném případě by se mohly vyskytnout chyby při pokusu o přístup k elementům stránky, které ještě nejsou načtené. Knihovna jQuery na to pamatuje a umožňuje zadat kód tak, aby se provedl až v okamžiku, kdy je stránka připravená. Jak na to ukazuje následující kód:

```
jQuery(document).ready(function(){
  // sem patří kód v JavaScriptu
});
```

V ukázce jsme použili trochu odlišný kód:

```
$(function(){
  // sem patří kód v JavaScriptu
});
```

Znak \$ je zkratkou za jQuery. Volání \$(něco) lze tedy chápat stejně jako volání jQuery(něco). \$(obsluha události) je další zkratkou, konkrétně pro událost ready, a odpovídá následujícímu kódu:

```
$(document).ready(obsluha události);
```

který je identický s tímto:

```
jQuery(document).ready(obsluha události);
```

Kvíz

1. Na jaké místo je nejlepší vložit kód v JavaScriptu?
 - a. Před úvodní značku elementu head.
 - b. Do elementu head.
 - c. Ihned za úvodní značku elementu body.
 - d. Těsně před koncovou značku elementu body.

Vytvoření elementů pro hru Ping Pong

Vše je připraveno a nic nám již nebrání pustit se do vývoje hry Ping Pong.

Jak na to – přidání základních elementů hry

1. Navážeme na náš předchozí příklad instalace jQuery. Otevřete soubor index.html v textovém editoru.

2. Následně do elementu body vložte elementy div pro hru a hrací pole. V elementu reprezentujícím hrací pole se budou nacházet dva elementy pro páčky hráčů a jeden pro míček. Celý element hracího pole pak bude součástí elementu pro hru:

```
<div id="game">
  <div id="playground">
    <div id="paddleA" class="paddle"></div>
    <div id="paddleB" class="paddle"></div>
    <div id="ball"></div>
  </div>
</div>
```

3. Základní struktura elementů hry je na svém místě, zbývá pro vytvořené elementy připravit styly. Následující kód vložte do elementu head:

```
<style>
  #playground{
    background: #e0ffe0;
    width: 400px;
    height: 200px;
    position: relative;
    overflow: hidden;
  }
  #ball {
    background: #fbb;
    position: absolute;
    width: 20px;
    height: 20px;
    left: 150px;
    top: 100px;
    border-radius: 10px;
  }
  .paddle {
    background: #bbf;
    left: 50px;
    top: 70px;
    position: absolute;
    width: 30px;
    height: 70px;
  }
  #paddleB {
    left: 320px;
  }
</style>
```

4. Stejně jako v předešlém případě umístíme náš kód v JavaScriptu za kód zajišťující připojení knihovny jQuery. Kód vložíme do samostatného souboru, protože se postupně rozroste. Vytvořte proto ve složce js soubor `html5games.pingpong.js`.
5. Soubor pro kód v JavaScriptu jsme připravili, zbývá ho připojit k dokumentu HTML. Následující kód vložte do souboru `index.html`, těsně před koncovou značku elementu `body`:

```
<script src="js/jquery-1.6.min.js"></script>
<script src="js/html5games.pingpong.js"></script>
```

6. Logika hry se bude nacházet v souboru `html5games.pingpong.js`. Prozatím se zde bude nacházet pouze následující kód zajišťující nastavení počátečních pozic pálek hráčů:

```
// kód nacházející se uvnitř $(function(){}) se spustí poté, co
// se dokončí načítání dokumentu a bude připravený DOM
$(function(){
    $("#paddleB").css("top", "20px");
    $("#paddleA").css("top", "60px");
});
```

7. Ověříme stávající konfiguraci ve webovém prohlížeči. Otevřete v něm soubor `index.html`, měl by se vám naskytnout podobný pohled jako na následujícím obrázku:



Vysvětlení

Naši hru jsme obohatili o dvě páčky a míček a s použitím knihovny jQuery jsme nastavili počáteční pozice obou pálek.

Seznámení s jQuery

jQuery je knihovna JavaScriptu navržená pro jednoduchou navigaci v elementech stránky, manipulaci s nimi, obsluhu událostí a provádění vzdálených asynchronních volání.

Knihovna umožňuje provádět dvě primární akce – **selekcí** a **modifikaci**. Selekcí se rozumí výběr všech elementů webové stránky, které odpovídají zadanému selektoru (známému z kaskádových stylů). Modifikace, jako je přidání elementu, odstranění potomků nebo nastavení stylu, probíhá nad vybranými elementy. Používání jQuery často obnáší spojení selekce a modifikace dohromady v jednu akci.

Například následující kód vybere všechny elementy, které patří do třídy `box`, a nastaví u uvedených vlastností nové hodnoty:

```
$(".box").css({"top": "100px", "left": "200px"});
```

Seznámení se selektory jQuery

Smyslem knihovny jQuery je usnadnit výběr elementů a provádění akcí s nimi. Potřebujeme nějak vybrat všechny potřebné elementy v rámci celého dokumentu. Aby mohla knihovna jQuery tuto úlohu splnit, vypůjčuje si selektory z kaskádových stylů. Selektor reprezentuje určitý vzor, kterému musí vybraný element odpovídat. Následující tabulka uvádí nejběžnější a nejužitečnější selektory, které v průběhu této knihy využijeme:

Zápis selektoru	Význam	Příklad použití
<code>\$("element")</code>	Vybere všechny elementy se zadaným názvem.	<code>\$("p")</code> vybere všechny elementy p. <code>\$("body")</code> vybere element body.
<code>\$("#identifikátor")</code>	Vybere element se zadaným identifikátorem (hodnotou atributu <code>id</code>).	Předpokládejme, že pracujeme s následujícím kódem: <code><div id="box1"></div></code> <code><div id="box2"></div></code> <code>\$("#box1")</code> vybere vyznačený element.
<code>\$(".třída")</code>	Vybere všechny elementy se zadanou třídou (hodnotou atributu <code>class</code>).	Předpokládejme, že pracujeme s následujícím kódem: <code><div class="apple"></div></code> <code><div class="apple"></div></code> <code><div class="orange"></div></code> <code><div class="banana"></div></code> <code>\$(".apple")</code> vybere vyznačené elementy, které patří do třídy <code>apple</code> .
<code>\$(".selektor1, selektor2, selektorN")</code>	Vybere všechny elementy odpovídající některému ze zadaných selektorů.	Předpokládejme, že pracujeme s následujícím kódem: <code><div class="apple"></div></code> <code><div class="apple"></div></code> <code><div class="orange"></div></code> <code><div class="banana"></div></code> <code>\$(".apple, .orange")</code> vybere vyznačené elementy, které patří buďto do třídy <code>apple</code> , anebo do třídy <code>orange</code> .

Seznámení s metodou css knihovny jQuery

Metoda `css` knihovny jQuery umožňuje získání i nastavení hodnot vlastností kaskádových stylů elementu.

Takto vypadá obecná deklarace metody `css`, popisující možnosti jejího použití:

```
.css(názevVlastnosti)
.css(názevVlastnosti, hodnota)
.css(poleVlastnosti)
```

Metoda `css` akceptuje několik typů parametrů, které uvádí následující tabulka:

Volání metody	Popis parametrů	Příklad použití
<code>.css(názevVlastnosti)</code>	<code>názevVlastnosti</code> je vlastnost, jejíž hodnota se má získat.	Metoda vrátí hodnotu zadané vlastnosti vybraného elementu. Následující kód vrátí hodnotu vlastnosti <code>background-color</code> elementu <code>body</code> : <code>\$("#body").css("background-color")</code> Dojde pouze k získání hodnoty, ne k její změně.
<code>.css(názevVlastnosti, hodnota)</code>	<code>názevVlastnosti</code> je vlastnost, <code>hodnota</code> je hodnota, na kterou se má vlastnost nastavit	Metoda nastaví hodnotu dané vlastnosti vybraného elementu či elementů na zadanou hodnotu. Následující kód nastaví hodnotu vlastnosti <code>background-color</code> na červenou barvu, u všech elementů, které patří do třídy <code>box</code> : <code>\$(".box").css("background-color", "#ff0000")</code>
<code>.css(poleVlastnosti)</code>	<code>poleVlastnosti</code> je množina párů vlastnost/hodnota, které se mají aktualizovat.	Metoda nastaví hodnotu hned několika zadaných vlastností u vybraného elementu či elementů. Následující kód nastaví hodnoty vlastností <code>left</code> a <code>top</code> elementu s identifikátorem <code>box1</code> : <code>\$("#box1").css({ "left" : "40px", "top" : "100px" })</code>

Výhody jQuery

Knihovna jQuery nabízí v porovnání s klasickým JavaScriptem několik výhod:

- Kód využívající knihovnu jQuery pro výběr elementů stránky a jejich modifikaci je výrazně kratší.
- Kratší kód se lépe čte, což je při vývoji her, které zpravidla obsahují mnoho kódu, velmi důležité.
- Psaní kratších kódů zrychluje vývoj.

- S použitím knihovny jQuery je možné vytvářet kód kompatibilní se všemi hlavními prohlížeči, bez nutnosti dodatečných úprav. Knihovna jQuery, jakožto nadstavba nad JavaScriptem, se sama postará o zajištění kompatibility napříč prohlížeči.

Práce s elementy hry pomocí jQuery

S použitím knihovny jQuery jsme v naší hře úspěšně nastavili počáteční polohu pálek hráčů. Nyní uděláme malý experiment týkající se dalšího umístění elementů hry.

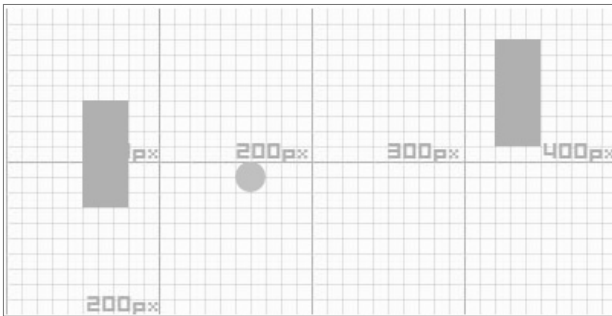
Jak na to – změna polohy elementů pomocí jQuery

V prvé řadě přidáme do naší hry mřížkové pozadí pro lepší orientaci:

1. Navážeme na předchozí ukázkou kódu hry Ping Pong.
2. Připravili jsme pro vás obrázek (`pixel_grid.jpg`) s mřížkou, který si můžete stáhnout na následující adrese: http://gamedesign.cc/html5games/pixel_grid.jpg.
3. Vytvořte v hlavní složce hry novou podsložku s názvem `images`.
4. Uložte obrázek `pixel_grid.jpg` do složky `images`. Tento obrázek nám pomůže při dalším pozicování elementů.
5. Otevřete soubor `index.html` v textovém editoru.
6. Upravte hodnotu vlastnosti `background` elementu `div` s identifikátorem `playground` tak, aby se na jeho pozadí zobrazoval obrázek mřížky, jak ukazuje následující kód:

```
#playground{
    background: #e0ffe0 url(images/pixel_grid.jpg);
    width: 400px;
    height: 200px;
    position: relative;
    overflow: hidden;
}
```

7. Pokud jste postupovali správně, uvidíte po otevření souboru `index.html` ve webovém prohlížeči něco podobného, jako na následujícím obrázku. Elementy hry se nacházejí nad mřížkou usnadňující určování jejich polohy.

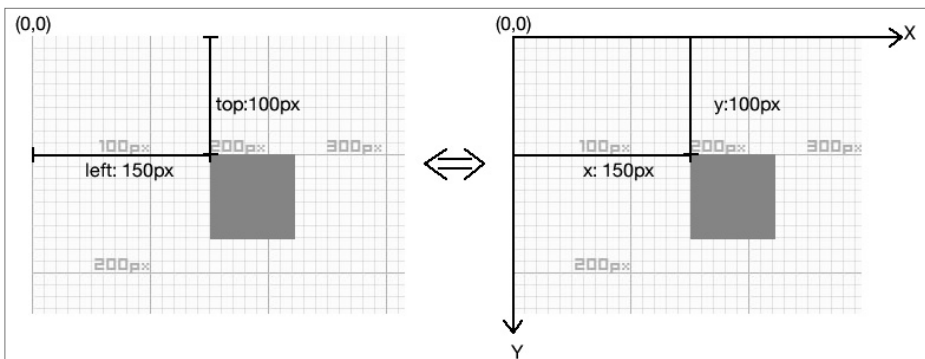


Vysvětlení

Praktickou ukázkou jsme zahájili vložení obrázku `pixel_grid.jpg` na pozadí elementu hracího pole. Jedná se o obrázek vytvořený k tomu, aby usnadnil polohování elementů hry. Obrázek tvoří mřížka polí o velikosti 10×10 pixelů, formujících větší bloky o velikosti 100×100 pixelů. Použijeme-li tento obrázek jako pozadí elementu div hracího pole, získáme tak v podstatě pravítko, podle kterého můžeme měřit pozice dceřiných elementů hracího pole.

Absolutní pozicování a jeho specifika

Má-li element absolutní pozici (vlastnost `position` má hodnotu `absolute`), hodnoty jeho vlastností `left` a `top` lze chápat jako **souřadnice X**, resp. Y levého horního rohu elementu. Tuto skutečnost ilustruje následující obrázek, který v levé části prezentuje skutečné vlastnosti elementu a v pravé pak souřadnicový systém, jež budeme intuitivně používat při programování hry.



Ve výchozím stavu jsou hodnoty vlastností `left` a `top` vztaženy k levému hornímu rohu webové stránky. Tento referenční bod se mění v okamžiku, kdy rodičovský element explicitně nastaví hodnotu své vlastnosti `position`. Pro potomky tohoto rodiče se v takovém případě stává referenčním bodem pro hodnoty vlastností `left` a `top` levý horní roh rodiče.

Z toho důvodu nastavíme pozici hracího pole na relativní a pozici všech elementů hry v něm na absolutní. Následující kód ukazuje použité nastavení:

```
#playground{
  position: relative;
}
#ball {
  position: absolute;
}
.paddle {
  position: absolute;
}
```

Kvíz

1. Který selektor jQuery použijete, pokud chcete vybrat všechny elementy header?
 - a. \$("#header")
 - b. \$(".header")
 - c. \$("header")
 - d. \$(header)

Načtení vstupu z klávesnice

Tato kniha se zaměřuje na vývoj her. Na běh hry je možné nahlížet jako na následující cyklus:

1. Vizualizuje se aktuální stav hry.
2. Hráči zadají své vstupní příkazy.
3. Logika hry vyhodnotí vstup uživatelů a do hry ho adekvátně promítné.
4. Zpět na krok č. 1.

V předchozích částech kapitoly jsme se dozvěděli, jak s použitím jQuery a kaskádových stylů zobrazovat objekty hry. Dalším krokem v pořadí je tedy načtení uživatelského vstupu. V této kapitole se zaměříme na vstup z klávesnice.

Jak na to – pohybování s elementy na základě vstupu z klávesnice

Vytváříme klasickou hru Ping Pong. Hru tvoří dvě pálky, jedna vlevo, druhá vpravo a míček mezi nimi, uprostřed hracího pole. Jeden hráč ovládá pátku na levé straně a může s ní pohybovat nahoru a dolů pomocí kláves W a S, druhý hráč má pod kontrolou pátku na opačné straně a pro její ovládání používá klávesy šipek nahoru a dolů.

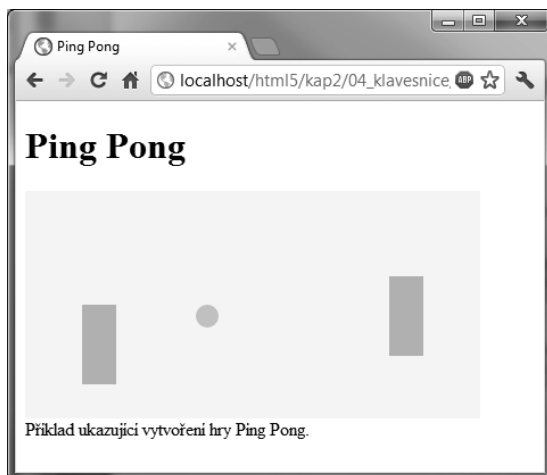
Prozatím pomíneme problematiku pohybu míčku a zaměříme se na vstup z klávesnice a jeho zpracování.

1. Vraťte se do složky pingpong.
2. Otevřete v textovém editoru soubor `html5games.pingpong.js`, který bude obsahovat logiku hry. Prozatím se zde bude nacházet kód obsluhující událost stisknutí klávesy klávesnice, který zajistí adekvátní posun páčky hráče nahoru či dolů. Nahraďte obsah souboru následujícím kódem:

```
var KEY = {
  UP: 38,
  DOWN: 40,
  W: 87,
  S: 83
}
$(function(){
  // naslouchání události stisknutí klávesy klávesnice
  $(document).keydown(function(e){
    switch(e.which){
      case KEY.UP: // šipka nahoru
        // určení aktuální hodnoty vlastnosti top páčky B a převedení
        // hodnoty na celé číslo
        var top = parseInt($("#paddleB").css("top"));
        // posunutí páčky B o 5 pixelů nahoru
        $("#paddleB").css("top",top-5);
        break;
      case KEY.DOWN: // šipka dolů
        var top = parseInt($("#paddleB").css("top"));
        // posunutí páčky B o 5 pixelů dolů
        $("#paddleB").css("top",top+5);
        break;
      case KEY.W: // W
        var top = parseInt($("#paddleA").css("top"));
        // posunutí páčky A o 5 pixelů nahoru
        $("#paddleA").css("top",top-5);
        break;
      case KEY.S: // S
        var top = parseInt($("#paddleA").css("top"));
        // posunutí páčky A o 5 pixelů dolů
        $("#paddleA").css("top",top+5);
        break;
    }
  });
});
```

```
    }  
  });  
});
```

3. Pojdme vyzkoušet ovládání pálek. Otevřete soubor `index.html` v prohlížeči Google Chrome. Zkuste stisknout klávesy W, S, šipku nahoru a dolů. Pálky by se měly v souladu se vstupem pohybovat nahoru a dolů, v tuto chvíli však ještě ne obě současně.



Vysvětlení

Podívejme se znovu na právě použitý kód. Dokument HTML obsahuje záhlaví, zápatí a element `div` s identifikátorem `game`. Element `game` obsahuje element s identifikátorem `playground`. Tento element reprezentující hrací pole má tři potomky, tedy dvě pálky a míček.

Vývoj hry v HTML5 často začíná přípravou správně strukturovaného dokumentu HTML. Hierarchie HTML pomáhá seskupovat podobné elementy hry (v našem případě několik elementů `div`) dohromady. Je to něco jako spojování jednotlivých položek do výsledné animace v Adobe Flash. Na finální hierarchii lze nahlížet jako na systém vrstev objektů hry, které je možné snadno vybírat a měnit jejich styl.

Klávesnice a kódy kláves

Každé klávese klávesnice přísluší určitý číselný kód. Na základě tohoto čísla je možné určit, jakou klávesu uživatel stiskl. Náš kód v JavaScriptu naslouchá události `keydown`. Objekt `event`, který se při výskytu této události předává její obsluze, obsahuje mimo jiné také kód stisknuté klávesy. Kód klávesy udává vlastnost `which` objektu `event`.

Do obsluhy události `keydown` je možné přidat volání metody `log` objektu `console` a následně v konzole pozorovat kódy reprezentující jednotlivé stisknuté klávesy.

```
$(document).keydown(function(e){
    console.log(e.which);
});
```

Lépe čitelné konstanty

Předchozí ukázka kódu názorně demonstrovala, jak lze s použitím kódů kláves vyhodnotit, jestli uživatel právě nestiskl některou z pro nás důležitých kláves. Například šipka nahoru má kód 38. Stačí tedy jednoduše porovnat kód právě stisknuté klávesy s tímto číslem, jak ukazuje následující kód:

```
$(document).keydown(function(e){
    switch(e.which){
        case 38:
            // zpracování stisku šipky nahoru
    }
}
```

Takový postup se však rozhodně nedá doporučit. Jeho důsledkem je velmi špatně čitelný kód. Představte si, že někdy později budete chtít změnit logiku programu a namísto šipky nahoru použít jinou klávesu. V danou chvíli už si nemusíme pamatovat, že kód 38 přísluší právě šipce nahoru. Z toho důvodu se používají konstanty se smyslupnými názvy:

```
var KEY = {
    UP: 38,
    DOWN: 40,
    W: 87,
    S: 83
}

// naslouchání události stisknutí klávesy klávesnice
$(document).keydown(function(e){
    switch(e.which){
        case KEY.UP:
            // zpracování stisku šipky nahoru
    }
}
```

Díky přiřazení hodnoty 38 do konstanty s názvem `KEY.UP` je ihned patrné, že daný blok kódu slouží ke zpracování stisku šipky nahoru a bez jakýchkoliv pochyb můžeme danou konstantu nastavit na novou hodnotu.

Převod řetězců na čísla pomocí funkce `parseInt`

Ve většině případů budeme při nastavování hodnot vlastností `left` a `top` používat hodnoty, jako je např. **100px**. Společně s číselnou hodnotou uvedeme i jednotku, kterou je v tomto případě pixel. Ve stejném formátu pak budeme hodnoty uvedených vlastností také načítat. Volání metody `$("#paddleA").css("top")` tedy nevrátí hodnotu **100**, nýbrž **100px**. Vzniká tak drobný problém, pokud se s touto hodnotou mají provádět nějaké aritmetické operace.

V tomto příkladu je naším cílem posunutí páčky hráče směrem vzhůru, nastavením hodnoty vlastnosti `top` na její aktuální hodnotu zmenšenou o pět. Předpokládejme, že je aktuální hodnota vlastnosti `top` páčky A rovna 100px. Pokud bychom použili následující příkaz ve snaze navýšit hodnotu o pět pixelů, nebyla by výsledkem očekávaná číselná hodnota, ale hodnota 100px5.

```
$("#paddleA").css("top") + 5
```

JavaScript v tomto případě provede metodu `css`, která vrátí hodnotu 100px. Následně k řetězci 100px připojí řetězec 5.

Potřebujeme nějakým způsobem převést řetězec 100px na číslo, se kterým lze provádět matematické operace.

Právě k tomu slouží funkce `parseInt` JavaScriptu. Deklarace funkce vypadá následovně:

```
parseInt(řetězec, číselnáSoustava)
```

Funkce `parseInt` má jeden povinný a jeden volitelný parametr.

Parametr	Popis	Příklad použití
řetězec	Řetězec, který se má zpracovat.	Funkce zpracuje v pořadí první číslo, které se v řetězci nachází. Pokud zadaný řetězec není možné převést na číslo, vrátí funkce hodnotu NaN (Not a Number). Ve výchozím nastavení zpracuje funkce řetězec uvozené prefixem 0x jako čísla v šestnáctkové soustavě. Několik příkladů použití: <code>parseInt("100px")</code> vrátí 100 <code>parseInt("5cm")</code> vrátí 5 <code>parseInt("0xF")</code> vrátí 15
číselná Soustava	Volitelný parametr. Číslo reprezentující číselnou soustavu, která se má použít.	Druhý parametr vynucuje zpracování řetězce v konkrétní číselné soustavě. Například: <code>parseInt("0x10")</code> vrátí 16 <code>parseInt("0x10", 10)</code> vrátí 0 <code>parseInt("FF", 16)</code> vrátí 255