

Создание игры «FlappyBird» на JavaScript

Семченко Регина Викторовна

Приамурский государственный университет имени Шолом-Алейхема

студент

Еровлев Павел Андреевич

Приамурский государственный университет имени Шолом-Алейхема

студент

Аннотация

В данной статье рассмотрен процесс написания собственной игры на языке JavaScript. Был изучен пошаговым методом код игры. Практическим результатом является собственная рабочая игра.

Ключевые слова: JavaScript, FlappyBird, игра, web-игра

Creating game «FlappyBird» in JavaScript

Semchenko Regina Viktorovna

Sholom-Aleichem Priamursky State University

student

Erovlev Pavel Andreevich

Sholom-Aleichem Priamursky State University

student

Abstract

This article describes the process of writing your own game in JavaScript. The game code was studied step by step. The bottom line is your own working game.

Keywords: JavaScript, FlappyBird, game, web game

Игра «FlappyBird» известна многим любителям игр на телефонах. Игра разработана вьетнамским разработчиком в 2013 году и за очень короткое время набрала большую известность. Игра очень простая, цель игры касанием по экрану заставлять птичку подлетать и пролетать между препятствий в виде зеленых труб не задевая их [1]. Эта игра довольно сложна в прохождении, поэтому у многих игроков она вызывает множество разных эмоций.

Цель данной статьи создать браузерную версию игры «FlappyBird» с помощью JavaScript.

Т.В. Зудилова, М.Л. Буркова в своем пособии рассматривают необходимые для создания игры объекты `math` и `document`, а так же рассказывают о способе создания и вызова функций [4]. В работе С.А. Беляева рассматриваются ключевые вопросы разработки однопользовательских двумерных игр на языке JavaScript [2]. В учебном пособии Захаркина В. В. рассмотрены основы применения алгоритмического языка JavaScript при создании интерактивных веб-документов [3]. В своей работе Н. Прохоренок, В. Дронов рассмотрели вопросы создания интерактивных Web-сайтов с помощью HTML, JavaScript [7]. А.В.Диков исследовал объект `Math` языка JavaScript, который обладает множеством математических свойств и методов. Таким образом, открывается возможность использования языка веб программирования для решения математических задач[11].

Для создания игры на понадобится создать 3 файла: HTML-файл(для открытия окна браузера), `js/game.js` (для написания кода на языке JavaScript), `img/`(для добавления картинок с игрой), для написания кода использовались рекомендации из источников [9-10].

После добавим в папку `img` картинки(рис.1-рис.5):



Рисунок 1 – Задний фон



Рисунок 2 – Передний фон



Рисунок 3 – Нижняя труба-препятствие



Рисунок 4 – Верхняя труба-препятствие



Рисунок 5 – Птичка

Добавив все картинки в папку приступим к написанию html-файла, пропишем там стандартные теги, а так же напишем тег canvas для отображения поля обрисовки игры (рис.6).

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Flappy Bird!</title>
</head>
<body>

<canvas id="canvas" width="288" height="512"></canvas>

<script src="js/game.js"></script>
</body>
</html>
```

Рисунок 6 – Код HTML-файла

Можно приступать к написанию кода самой игры. Создадим переменные, и добавим картинки (рис. 7)

```
var cvs = document.getElementById("canvas");
var ctx = cvs.getContext("2d");

var bird = new Image();
var bg = new Image();
var fg = new Image();
var pipeUp = new Image();
var pipeBottom = new Image();

bird.src = "img/bird.png";
bg.src = "img/bg.png";
fg.src = "img/fg.png";
pipeUp.src = "img/pipeUp.png";
pipeBottom.src = "img/pipeBottom.png";
```

Рисунок 7 – Добавление переменных

Дольше напишем оставшиеся переменные, укажем место старта птички, напишем чтобы при нажатии клавиши птичка поднималась на определенное расстояние (рис.8)

```
var gap = 90;

// При нажатии на какую-либо кнопку
document.addEventListener("keydown", moveUp);

function moveUp() {
  yPos -= 25;
}

// Создание блоков
var pipe = [];

pipe[0] = {
  x : cvs.width,
  y : 0
}

var score = 0;
// Позиция птички
var xPos = 10;
var yPos = 150;
var grav = 1.5;
```

Рисунок 8 – Создание блоков и Позиция птички

Теперь осталось написать функции при которых будет идти добавление счета, возможность птичке летать и падать, возможность удариться о препятствие, создание блоков за пределами карты, чтобы игра была зациклена и не имела конца(Рис.9)

```
var cvs = document.getElementById("canvas");
var ctx = cvs.getContext("2d");

var bird = new Image();
var bg = new Image();
var fg = new Image();
var pipeUp = new Image();
var pipeBottom = new Image();

bird.src = "img/bird.png";
bg.src = "img/bg.png";
fg.src = "img/fg.png";
pipeUp.src = "img/pipeUp.png";
pipeBottom.src = "img/pipeBottom.png";

var gap = 90;

// При нажатии на какую-либо кнопку
document.addEventListener("keydown", moveUp);

function moveUp() {
  yPos -= 25;
}

// Создание блоков
var pipe = [];

pipe[0] = {
  x : cvs.width,
  y : 0
}

var score = 0;
// Позиция птички
var xPos = 10;
var yPos = 150;
var grav = 1.5;

function draw() {
  ctx.drawImage(bg, 0, 0);

  for(var i = 0; i < pipe.length; i++) {
    ctx.drawImage(pipeUp, pipe[i].x, pipe[i].y);
    ctx.drawImage(pipeBottom, pipe[i].x, pipe[i].y + pipeUp.height + gap);

    pipe[i].x--;

    if(pipe[i].x == 125) {
      pipe.push({
        x : cvs.width,
        y : Math.floor(Math.random() * pipeUp.height) - pipeUp.height
      });
    }
  }

  // Отслеживание прикосновений
  if(xPos + bird.width >= pipe[i].x
```

```

&& xPos <= pipe[i].x + pipeUp.width
&& (yPos <= pipe[i].y + pipeUp.height
|| yPos + bird.height >= pipe[i].y + pipeUp.height + gap) || yPos + bird.height >= cvs.height - fg.height) {
location.reload(); // Перезагрузка страницы
}

if(pipe[i].x == 5) {
score++;
}
}

ctx.drawImage(fg, 0, cvs.height - fg.height);
ctx.drawImage(bird, xPos, yPos);

yPos += grav;

ctx.fillStyle = "#000";
ctx.font = "24px Verdana";
ctx.fillText("Счет: " + score, 10, cvs.height - 20);

requestAnimationFrame(draw);
}

pipeBottom.onload = draw;

```

Рисунок 9 – Весь JavaScript код

Теперь запустив игру, имеется возможность насладиться той самой популярной игрой, написанной своими руками (рис.10).

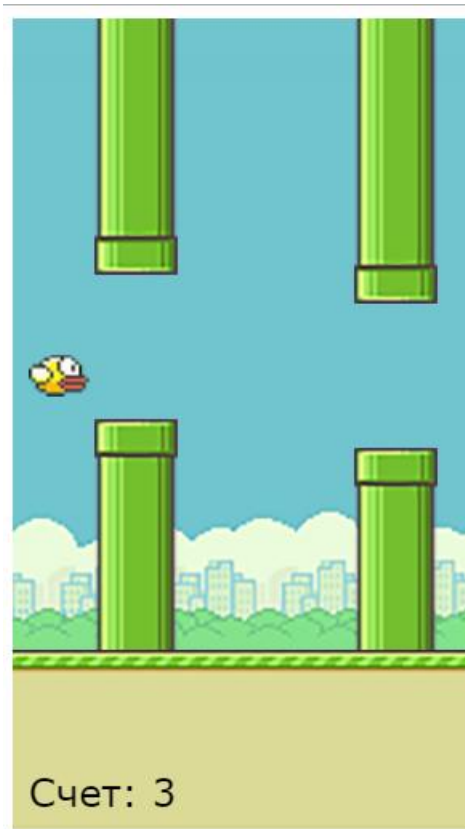


Рисунок 10 – Игра FlappyBird

Игра имеет немного сложный код, а именно функции столкновений с объектами и полета.

Подводя итоги можно сказать, что написание собственной игры не занимает больших трудностей, а сам процесс и результат доставляет удовольствие. В ходе создания игры был рассмотрен пошаговый процесс создания функций и написания кода. В результате была получена полноценная игра Flappybird.

Библиографический список

1. Flappy Bird // Википедия URL: https://ru.wikipedia.org/wiki/Flappy_Bird (дата обращения: 20.08.2019).
2. Беляев С.А. Разработка игр на языке JavaScript: Учебное пособие. СПб.: Лань, 2016. 128с.
3. Захаркина В. В. JavaScript. Основы клиентского программирования: Учебное пособие. СПб.: Ф-т филологии и искусств СПбГУ, 2007. 73 с.
4. Зудилова Т.В., Буркова М.Л. Web-программирование JavaScript. СПб: НИУ ИТМО, 2012. 68 с.
5. Что такое JavaScript? Основные функции JavaScript. http://www.progaprosto.ru/doc/yazyk_programmirovaniya_javascript.php
6. Основы Javascript [Электронный ресурс]. URL: <https://learn.javascript.ru/first-steps> (дата обращения 20.08.2018)
7. Прохоренок Н., Дронов В. HTML, JavaScript, PHP и MySQL. СПб.: БХВПетербург, 2015. 766с.
8. Флэнаган Д. JavaScript. Подробное руководство. М.: Символ-Плюс, 2012. 357с.
9. Язык программирования JavaScript. URL: http://progaprosto.ru/doc/yazyk_programmirovaniya_javascript.php (дата обращения 20.08.2018).
10. Wiki-учебник по веб-технологиям: JavaScript. URL: <http://www.webmasterwiki.ru/JavaScript> (дата обращения 20.08.2018).
11. Диков А.В. Математические алгоритмы на JavaScript // Известия Пензенского государственного педагогического университета им. В.Г. Белинского. 2009. № 17. С. 84-88.