Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и механики

**Кафедра «Теоретическая механика»**

**КУРСОВОЙ ПРОЕКТ**

**Реализация и визуализация игры “Flappy Cow”**

по дисциплине «Математическое моделирование»

Выполнили

студенты гр.13632/1 Мальцева О.Н.

Клочко Д.А.

Руководитель

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Санкт-Петербург

2019

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Постановка задачи. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 3  3 |
| 1. Реализация. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
| 1.1. Описание кода HTML и JavaScript. . . . . . . . . . . . . . . . . . . . . . . . . . . .  Результаты. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 5  8 |
| Вывод и заключение. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 9 |

**Введение**

JavaScript – многофункциональный язык, который позволяет выполнять задачи различной направленности и сложности. Данная работа посвящена созданию игры "Flappy Cow" на подобии игры “Flappy Bird” с заданными условиями и правилами её функционирования. Актуальность темы заключается в том, что в современном мире существует немало всевозможных игр, разработанных на базе различных языков программирования, и поэтому немаловажно понимать основные механизмы и принципы создания и визуализации подобных программ. Цель курсовой работы заключается в изучении основных принципов создания игры "Flappy Bird". Объект исследования – процесс использования языков программирования для создания простейшей игры.   
Предмет исследования – программные инструменты языков JavaScript и HTML, которые применяются при разработке простейших игр.

**Постановка задачи**

Реализовать и визуализировать игру “Flappy Cow” со следующими правилами:

1. Управление производится клавишами вверх и вниз
2. “Коровка” не должна касаться травинок
3. Счёт ведётся с начала игры, вне зависимости от того, пройдет ли "корова" "травинку".

**Реализация**

Перечень функций, которые были использованы для написания программы:

* В html : <html lang="en"> Код языка - английский
* <meta> используются для хранения информации предназначенной для браузеров и поисковых систем ,charset задает кодировку документа.
* utf-8 - стандарт кодирования текста, позволяющий более компактно хранить и передавать символы
* Метатег viewport нужен для того, чтобы указывать браузерам на то, в каком масштабе необходимо отображать пользователю видимую область веб-страницы. Включение поддержки тега meta viewport для адаптивных сайтов осуществляется посредством добавления всего одной строчки в раздел head веб-страницы:<meta name="viewport" content="width=device-width, initial-scale=1">. width=device-width отвечает за то, чтобы ширина видимой области веб-страницы равнялась CSS ширине устройству. initial-scale - устанавливает первоначальный масштаб веб-страницы. Значение 1 означает то, что масштаб равен 100%
* Атрибут «content» должен иметь значение «IE=[режим\_документа]». В этом значении должен указываться один из режимов документов, который определяет возможности доступные странице (CSS и JS функционал), в нашем случае это ie - edge , он включает наиболее новый из доступных режимов документа, у нас это режим IE 8
* x-ua-compatible - в данном документе web-браузер Internet Explorer будет более точно следовать спецификации документа.

**Описание кода HTML и JavaScript**

Создаем канвас шириной 288, длиной 512 пикселей соответственно, задаём цвет канваса и его ширину

*<canvas id="canvas" width="288" height="512" style="border:1px solid black;"></canvas>*

Создаем переменную для работы с канвасом

*var ctx = canvas.getContext('2d')*

Вводим переменные для "коровы", заднего и переднего фона, для верхней и нижней травинок и обнуляем счет

*var bird = new Image();   
var bg = new Image();   
var fg = new Image();   
var pipeUp = new Image();   
var pipeBottom = new Image();   
var score = 0;*

Загружаем картинки найденные в интернете, заранее подогнанные по размеру нашей игры и помещенные в отдельную папку

*bird.src = 'img/korova.png';   
bg.src = 'img/nebo.jpg'   
fg.src = 'img/flappy\_bird\_fg.png';   
pipeUp.src = 'img/flappy\_bird\_pipeUp.png';   
pipeBottom.src = 'img/flappy\_bird\_pipeBottom.png';*

Вводим переменную для расстояния между травинками

*var gap = 70;*

Начальная сила гравитации

*var gravitation = 1;*

Первоначальные координаты нахождения "коровы"

*var Posx = 10;   
var Posy = 150;*

Создаем травинки

*var pipe = [];   
  
pipe[0] = {   
x : canvas.width,   
y : 0   
}*

Рисуем и "чистим" фон

*function draw(){   
ctx.clearRect(0,0, 288,512); ctx.drawImage(bg, -bg.width/2-90, 0);*

Создаем травинки, находящиеся в разных местах в течение игры

*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: canvas.width,   
y: Math.floor(Math.random() \* pipeUp.height) - pipeUp.height   
});   
}*

Условия того, что "корова" дотронется до травинки и игра будет проиграна

*if (Posx + bird.width >= pipe[i].x   
&& Posx <= pipe[i].x + pipeUp.width   
&& (Posy <= pipe[i].y + pipeUp.height   
|| Posy + bird.height >= pipe[i].y + pipeUp.height + gap)   
|| Posy + bird.height >= canvas.height - fg.height){*

После того, как "корова" дотрагивается до травинки, начинается новая игра

*location.reload();*

Отрисовка заднего фона, за вычетом "грунта"

*ctx.drawImage(fg, 0, canvas.height - fg.height);*

Отрисовка "коровы"

*ctx.drawImage(bird, Posx, Posy);*

Счет ведется с начала игры, вне зависимости от того, пройдет ли "корова" "травинку"

*score++;   
some\_name.innerHTML = score;*

*requestAnimationFrame* - Сгруппировать все перерисовки в одну и запускать их централизованно, все вместе.

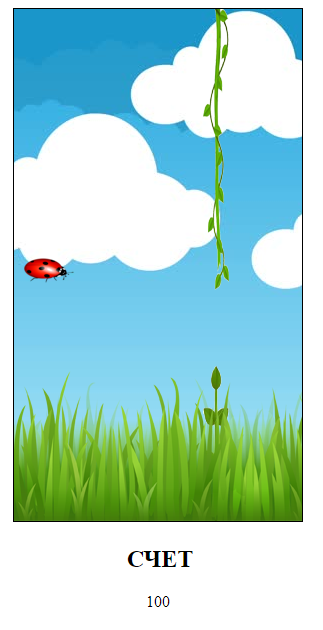
'Загрузка ' нижних "травинок"

*pipeBottom.onload = draw;*

**Результаты**

На языке программирования Javascript был составлен структурно простой код, позволяющий пользователю сыграть в аналог игры “Flappy Bird” – “Flappy Cow”. Управление производится клавишами вверх и вниз. Но было решено разнообразить игровой процесс добавлением новых условий игры: счёт ведётся с начала игры, вне зависимости от того, пройдет ли "корова" "травинку".

На рисунке, представленном ниже, можно ознакомиться с игровым процессом и интерфейсом игры. Посередине поля изображена “коровка” и ярко-зелёным цветом травинки, которых она не должна касаться, в нижней части экрана отображается счёт, идущий с начала игры, вне зависимости от того, пройдет ли "корова" "травинку".



**Вывод**

Таким образом, реализация и визуализация аналога игры “Flappy Bird” – “Flappy Cow” оказались успешными, нам удалось выполнить поставленные задачи и также освоить использование новых функций. В итоге, мы получили новый и полезный опыт в работе над подобными заданиями и составлением курсовых работ, который будет существенен в дальнейшем.

**Заключение**

Безусловно, есть возможность добавления новых механик к игре в качестве последующей доработки программы и улучшения как динамической, так и визуальной составляющих игры. В заключении хотелось бы сказать, что JavaScript – многофункциональный язык, который позволяет выполнять задачи различной направленности и сложности.

**Форма задания на выполнение   
курсового проекта**

Санкт-Петербургский политехнический университет Петра Великого

**ЗАДАНИЕ**

**НА ВЫПолнение курсового проекта**

студенту группы 13632/1 Мальцевой О.Н.

студенту группы 13632/1 Клочко Д.А.

***1. Тема проекта:*** Реализация и визуализация игры «Flappy Cow»

***2. Срок сдачи студентом законченного проекта***

***3. Исходные данные к проекту***: курс лекций по математическому моделированию за первый и второй семестры

***4. Содержание пояснительной записки*** (перечень подлежащих разработке вопросов): введение, основная часть (раскрывается структура основной части), заключение, список использованных источников, приложения.

Примерный объём пояснительной записки 9 страниц печатного текста.

***5. Перечень графического материала*** (с указанием обязательных чертежей и плакатов): отсутствует

***6. Консультанты***

***7. Дата получения задания***: «\_\_\_».\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись) (инициалы, фамилия)*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента) (инициалы, фамилия)*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента) (инициалы, фамилия)*

\_\_\_\_\_\_\_\_\_\_\_\_*(дата)*