Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики
Кафедра «Теоретическая механика»

К У Р С О В О Й П Р О Е К Т

Основы математического моделирования и разработки игр на языке программирования JavaScript

по дисциплине «Математическое моделирование»

Выполнили
студенты гр.13632/2 М.М. Фролов
И.Е. Груздев
М.Д. Рыбаконенко
Д.А. Соколова

Руководитель Д.В. Цветков

«___» _____ 2019 г.

Санкт-Петербург

2019

ФОРМА ЗАДАНИЯ НА ВЫПОЛНЕНИЕ
КУРСОВОГО ПРОЕКТА (КУРСОВОЙ РАБОТЫ)

Санкт-Петербургский политехнический университет Петра Великого

ЗАДАНИЕ

НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА

студентам группы 13632/2 Фролову М.

Груздеву И.

Рыбаконенко М.

Соколовой Д.

1. ***Тема проекта (работы):***

   Основы математического моделирования и разработки игр на языке программирования JavaScript

2. ***Срок сдачи студентом законченного проекта (работы)***
   22/06/2019
3. ***Исходные данные к проекту (работе)***:
   Материалы по курсу математического моделирования

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

Примерный объём пояснительной записки _____ страниц печатного текста.

5. Дата получения задания: «___»_____ 20__ г.

Руководитель _____ _____

Задание принял к исполнению _____ _____

СОДЕРЖАНИЕ

Введение

Данный курсовой проект состоит из двух независимых частей, что связано с желанием освоить моделирование и программирование в различных направлениях.

Первая часть посвящена исследованию колебаний двойного маятника. Разработанная нами программа позволяет моделировать колебания маятника при различных начальных условиях и наблюдать за ходом моделирования посредством анимации и графиков.

Вторая часть посвящена разработке игры, структурно повторяющей всем известную игру “Geometry Dash”. В игре создан один уровень, в случае удачного прохождения которого появляется окно, уведомляющее о победе.

1. Задачи и цели

1.1 Двойной маятник

Целью данной части курсового проекта являются изучение систем с двумя степенями свободы и написание программы визуализации такой системы на JavaScript.

Задачи, решаемые в процессе моделирования:

1) смоделировать процесс колебаний двойного маятника – то есть определить координаты точек m_1 и m_2 как функцию от времени при некоторых начальных условиях;

2) решить данные уравнения численно посредством JavaScript;

3) визуализировать процесс колебаний.

1.2 Runner

Целью этой части курсового проекта являются написание программы, игровой процесс в которой построен по принципам, сходным с игрой «Geometry dash».

Определим основные задачи, решаемые в процессе разработки:

1) создание карты одного уровня, на которой будет происходить игровой процесс;

2) создание игрового персонажа;

3) создание функции, реализующей передвижение игрового персонажа по карте и прыжок;

4) реализация функции, определяющей столкновение персонажа с элементами карты.

2. Основы математического моделирования на JavaScript

Под математическим моделированием на языке программирования JavaScript понимается следующая совокупность действий:

1) вывод и аналитическое решение (в случае необходимости) уравнений, описывающих состояние и/или изменение состояния интересующей нас системы;

2) представление необходимых формул на языке JavaScript;

3) вычисление интересующих параметров системы;

4) визуализация процесса.

2.1. Аналитическое исследование выбранного процесса

Двойной маятник – это система, колебания которой при некоторых заданных параметрах является хаотическими. Для упрощения рассматриваемой задачи введем следующие допущения:

- подвесы l_1 и l_2 маятника – абсолютно твердые тела

- массы грузов m_1 и m_2 сосредоточены в соответствующих точках

- трение в местах соединения отсутствует

Процесс аналитического исследования данной модели двойного маятника можно разделить на два этапа:

1) составления дифференциальных уравнений, описывающих колебательный процесс (Корректность полученных выводов проверена сравнением с материалами сайта [1]);

2) решение полученных уравнений (в ходе решения использованы материалы методического пособия [2]).

В качестве отправной точки возьмем переход от декартовых координат x и y к обобщенным, в качестве которых, в данном случае, удобно взять углы отклонения стержней маятников от вертикали. Все используемые обозначения представлены на рисунке 1.

$$\begin{cases} x_1 = L_1 \sin\theta_1 \\ y_1 = L_1 \cos\theta_1 \end{cases} \qquad \begin{cases} x_2 = L_1 \sin\theta_1 + L_2 \sin\theta_1 \\ y_2 = L_1 \cos\theta_1 + L_2 \cos\theta_1 \end{cases}$$

Получив эти уравнения, мы сможем составить Лагранжиан этой колебательной системы.

$L = E_k - U$, где E – кинетическая энергия, U – потенциальная энергия системы.

Приняв допущение о малости углов отклонения получим:

$$L = E_k - U = \frac{(m_1+m_2)}{2} {L_1}^2 \dot{\theta_1}^2 + \frac{m_2}{2} {L_2}^2 \dot{\theta_2}^2 + m_2 L_1 L_2 \dot{\theta_1} \dot{\theta_2} -$$

$$- \frac{(m_1+m_2)}{2} g L_1 \dot{\theta_1}^2 - \frac{m_2}{2} g L_2 \dot{\theta_2}^2$$

Далее составим основные уравнения колебаний нашей системы – уравнения Лагранжа:

$$\begin{cases} \dfrac{\partial^2 L}{\partial \dot{\theta}_1 \partial t} - \dfrac{\partial L}{\partial \theta_1} = 0 \\ \dfrac{\partial^2 L}{\partial \dot{\theta}_2 \partial t} - \dfrac{\partial L}{\partial \theta_2} = 0 \end{cases}$$

Имеем:

$$\begin{cases} (m_1+m_2){L_1}^2 \ddot{\theta}_1 + m_2 L_1 L_2 \ddot{\theta}_2 + (m_1+m_2) g L_1 \theta_1 = 0 \\ m_2 L_1 L_2 \ddot{\theta}_1 + m_2 {L_2}^2 \ddot{\theta}_2 + m_2 L_2 \theta_2 = 0 \end{cases}$$

Решая данную систему при ранее принятых допущениях, можно выразить углы отклонения от вертикали как функцию от времени:

$$\theta_1(t) = \frac{1}{2}\left(\theta_1(0) - \sqrt{\frac{\mu}{1+\mu}}\ \theta_2(0)\right)\cos\omega_1 t + \frac{1}{2}\left(\theta_1(0) + \sqrt{\frac{\mu}{1+\mu}}\ \theta_2\right)\cos\omega_2 t$$

$$\theta_2(t) = -\frac{1}{2}\sqrt{\frac{\mu}{1+\mu}}\ (\theta_1(0) - \sqrt{\frac{\mu}{1+\mu}}\ )\cos\omega_1 t + (\theta_1(0) + \sqrt{\frac{\mu}{1+\mu}}\ \theta_2(0))\cos\omega_2 t$$

Где C_1 и C_2:

$$C_1 = \frac{1}{2}\theta_1(0)\ - \frac{1}{2}\sqrt{\frac{\mu}{1+\mu}}\ \theta_2(0)$$

$$C_2 = \frac{1}{2}\theta_1(0) + \frac{1}{2}\sqrt{\frac{\mu}{1+\mu}}\ \theta_2(0)$$

Итак, набор уравнений, необходимых для моделирования колебаний на JavaScript:

$$\begin{cases} x_1 = L_1 \sin\theta_1 \\ y_1 = L_1 \cos\theta_1 \end{cases} \qquad \begin{cases} x_2 = L_1 \sin\theta_1 + L_2 \sin\theta_1 \\ y_2 = L_1 \cos\theta_1 + L_2 \cos\theta_1 \end{cases}$$

$$\theta_1(t) = \frac{1}{2}(\theta_1(0) - \sqrt{\frac{\mu}{1+\mu}}\ \theta_2(0))\cos\omega_1 t + \frac{1}{2}(\theta_1(0) + \sqrt{\frac{\mu}{1+\mu}}\ \theta_2))\cos\omega_2 t$$

$$\theta_2(t) = -\frac{1}{2}\sqrt{\frac{\mu}{1+\mu}}\ (\theta_1(0) - \sqrt{\frac{\mu}{1+\mu}}\ )\cos\omega_1 t + (\theta_1(0) + \sqrt{\frac{\mu}{1+\mu}}\ \theta_2(0))\cos\omega_2 t$$

$$C_1 = \frac{1}{2}\theta_1(0) - \frac{1}{2}\sqrt{\frac{\mu}{1+\mu}}\ \theta_2(0)$$

$$C_2 = \frac{1}{2}\theta_1(0) + \frac{1}{2}\sqrt{\frac{\mu}{1+\mu}}\ \theta_2(0)$$

На этом первая задача решена – данные уравнения исчерпывающе описывают положение точек m_1 и m_2 в декартовой системе координат.

Замечание: процесс выведения и решения полученных дифференциальных уравнений описан более полно в приложении 1.

2.2. Визуализация моделируемого процесса, написание соответствующей программы на JavaScript

Определим ряд требований, которым должна соответствовать программа:

- результаты расчетов, проводимых программой, не появляются на экране непосредственно, а являются исходными данными для визуализации и построения графиков;
- визуализация представляет собой анимированную схематичную модель двойного маятника и график положения масс от времени;
- стенд модели должен быть снабжен блоком управления, в котором можно изменять некоторые параметры системы, такие как:
    - l – длина стержней маятника;
    - m_1, m_2 – массы грузов;
    - q_1, q_2 – углы отклонения стержней от вертикали.

В отличие от предыдущего пункта здесь ход мысли пойдет от целого (конечного результата) к частному (его «элементарным» составляющим).

Блок управления представлен на рисунке 3.

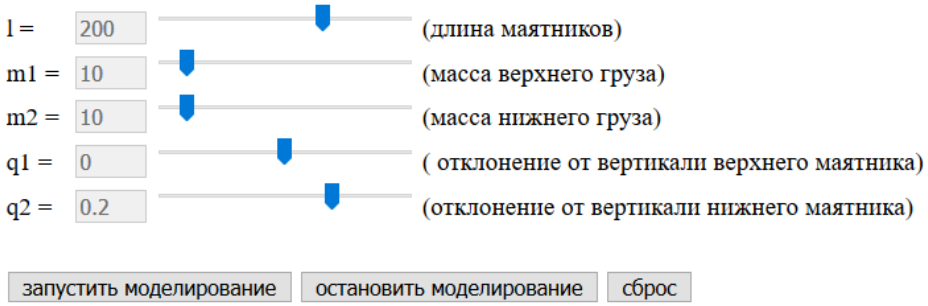

Рис. 3 вид блока управления

В блоке управления собраны все настройки исследуемой модели, изменять которые можно перетаскиванием соответствующего ползунка.

Также присутствуют три кнопки изменения состояния системы – запуск, остановка и сброс.

Блок анимации можно увидеть на рисунке 4.

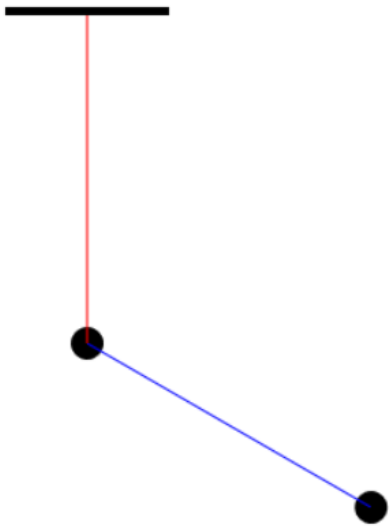

Рис. 4 Начальное положение маятника

Как было описано в поставленных целях – визуализация представляет собой анимированную схематичную модель двойного маятника, закрепленного на неподвижном подвесе. Черные шары на окончаниях стержней – схематичное обозначения положения грузов m1 и m2 (стоит отметить, что грузы непропорционально велики по сравнению с остальной моделью – это сделано

лишь для большей информативности, размер шаров задан заранее и не зависит от параметров системы).

Подробный разбор принципов работы данной модели будет представлен далее.

Блок графиков показан на рисунке 5.

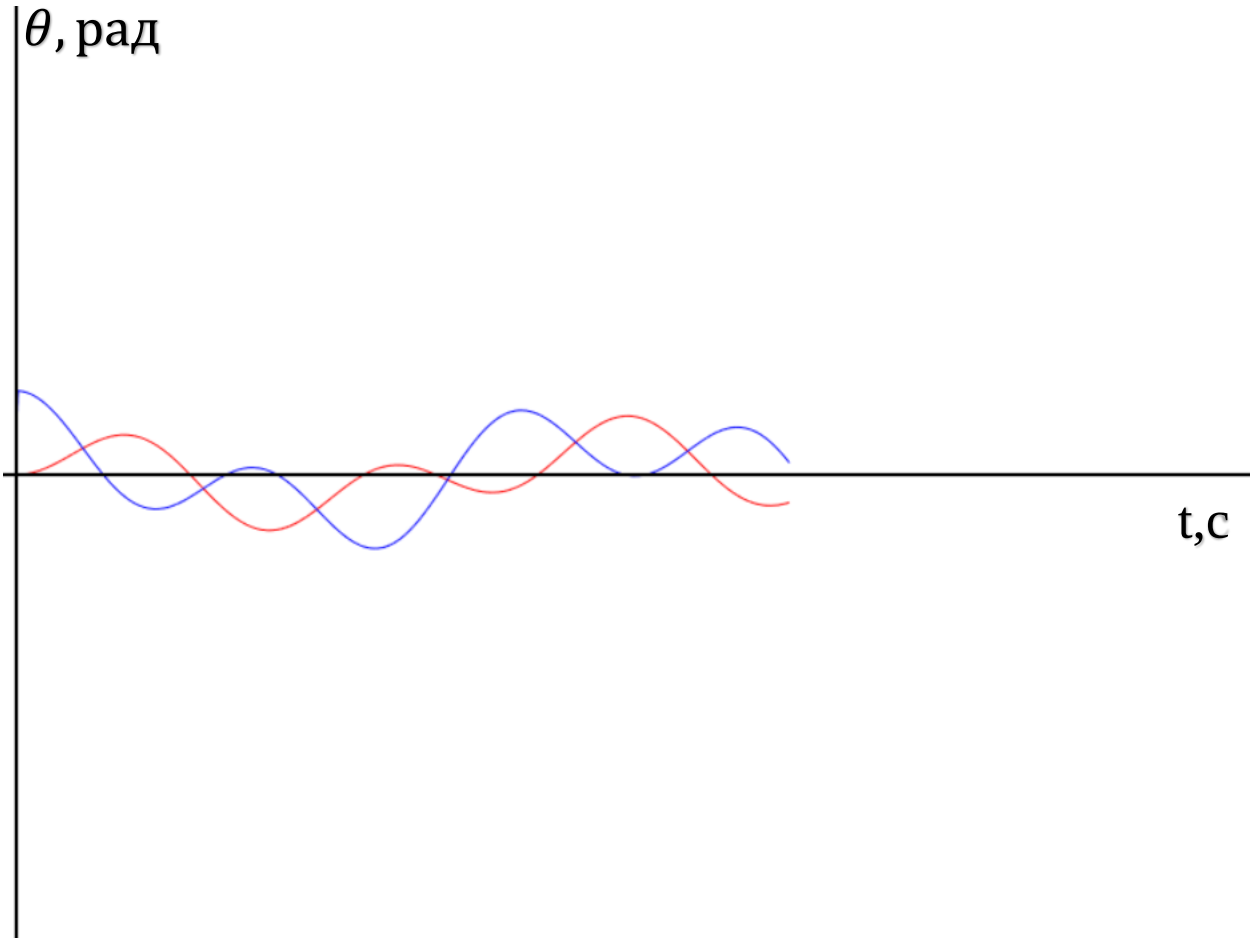

Рис. 5 Пример графика зависимости координаты от времени

Данный блок содержит в себе информацию о состоянии системы в виде графиков (на данном стенде представлен только график координаты от времени, но можно добавить и любые другие).

Данные с графиков позволяют судить о состоянии системы на некотором временном отрезке, чего не позволяет сделать анимация.

Для удобства чтения графиков линии, соответствующие разным грузам, раскрашены в разный цвет.

Приступим к рассмотрению данного стенда изнутри:

Скрипт, реализующий расчет и анимацию маятника, состоит из следующих составных частей:

1) Расчет положения частей маятника на текущем шаге.

Данный блок содержит функцию mechanics(), единственная задача которой – расчет положения по формулам, полученным в параграфе 1.2.

2) Визуализация маятника и графика.

Поскольку на каждом шаге необходимо представить два независимых изображения (маятник и график), блок разделен на две функции:

draw() – занимается визуализацией маятника;
draw_graph() – занимается отрисовкой графика.

3) Пульт управления.

Блок содержит функцию app_control_panel(), которая считывает значения слайдера при их изменении и передает это значение соответствующим параметрам.

*Также в некоторых браузерах обнаружен интересный факт, что при обновлении страницы значения на слайдерах не сбрасываются к начальным, для чего был добавлен дополнительный обработчик на событие перезагрузки.

4) Функция взаимодействия.

Написаны основные части программы, но функции еще не знают о существовании друг друга.

Эта часть программы реализует корректное взаимодействие вышеописанных функций между собой. Принцип ее работы следующий:

Изначально посредством setInterval происходит вызов функции first_frame_draw() – она отвечает за изображение в статике.

При нажатии на кнопку запуска анимации блокируются слайдеры в блоке управления (это сделано для того, чтобы в процессе моделирования нельзя было изменить параметры), останавливается вызов first_frame_draw() и тем же методом вызывается setup().

При нажатии на паузу останавливается вызов setup(), что останавливает и саму анимацию.

При нажатии на сброс восстанавливается то, что было отключено нажатием запуска – вместо setup() вновь вызывается first_frame_draw(), а слайдеры становятся вновь активны.

Введенный "ключ" key – основной "регулятор" взаимодействия в этом блоке. Разберемся, в чем заключается его функция:

Изначально при задании ему дается значение 0 – это будет сигнализировать о том, что движение системы еще не началось или было остановлено.

При нажатии на кнопку запуска увеличиваем key. Так как внутри функции запуска стоит проверка на значение ключа, то при повторном нажатии запуска ничего не произойдет, поскольку значение ключа отлично от нуля.

При нажатии на паузу ключу присваивается значение 0, если он был отличен от нуля и не происходит ничего в противном случае. Таким образом, повторное нажатие на паузу тоже ничего не дает.

При нажатии сброса значение ключа повлияет на вариант, которым сброс произойдет, но в любом случае будет восстановлена активность слайдеров и произведен откат таймера к нулевому значению.

Данная функция определяет корректное взаимодействие не только частей программы между собой, но и взаимодействие интерфейса и пользователя, не позволяя ему менять начальные параметры при запущенном (или приостановленном) моделировании.

3. Разработка игры на JavaScript

3.1. Основные элементы игрового уровня. Представление различных типов препятствий.

Основой для нашего уровня служит первый уровень игры geometry dash, рассмотрим элементы уровня, присутствующие в нем, и разделим их на категории:

- "шипы" (spike) – препятствия в форме треугольников, столкновение с ними ведет к проигрышу независимо от взаимного расположения препятствия и персонажа;

- "пол" (floor) – элементы, выполняющие две функции – при падении на них сверху такие элементы не представляют для персонажа опасности, однако при боковом столкновении данный элемент является для игрока препятствием. Из таких элементов можно формировать объекты произвольной ширины и высоты;

- "ступенька" (step) – объект со свойствами, сходными с предыдущим, однако его размер фиксирован.

В нашей игре уровень представляет собой массив объектов, каждый из которых является одним из вышеописанных элементов. Каждый объект обладает следующими параметрами:

- object: 'name' - название объекта, соответствует типу препятствия;

- x, y – пара координат, определяющих положение объекта на карте по горизонтали и вертикали соответственно;

Описанный набор является общим для всех элементов. Однако поскольку объекты floor могут иметь произвольную ширину и высоту, им требуется еще несколько параметров:

- length – длина объекта (по горизонтали);

- h – высота объекта (по вертикали);

С каждым типом объекта далее будет связана соответствующая функция, посредством которой будет происходить визуализация этого объекта.

Результат показан на рисунке 6.

Рис. 6 Стандартная цветовая тема

Также в данной программе будут реализованы темы оформления фона для карты. Кроме стандартной синей темы (рис. 6) будут доступны красная (рис 7), зеленая (рис. 8) и «Material dark» (рис. 9) темы.

Рис. 7 Красная тема

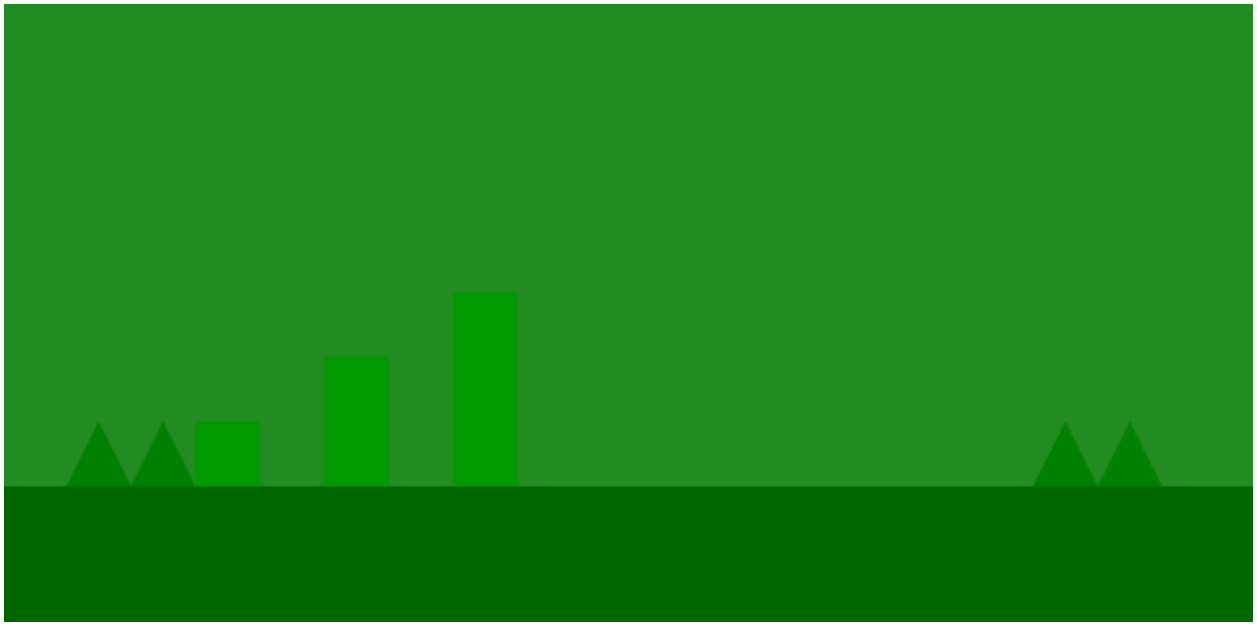

Рис. 8 Зеленая тема

Рис. 9 Тема «Material dark»

3.2. Управление передвижением. Столкновение с препятствиями.

Передвижение персонажа по карте, точнее сказать "эффект передвижения", создается за счет движения самой карты в направлении, противоположном к предполагаемому направлению движения персонажа. Такой подход изначально кажется неразумным, однако это единственный способ постоянно держать персонажа в поле зрения. Для реализации передвижения карты во все функции отрисовки, отвечающие за изображение объектов карты, добавлены соответствующие слагаемые.

Наряду с перемещением персонажа по горизонтали необходимо реализовать и другую важнейшую составляющую механики передвижения – прыжок.

Прыжок работает следующим образом:

В основном игровом цикле стоит обработчик addEventListener, который распознает событие нажатия на любую кнопку клавиатуры. По нажатию пробела происходит запуск функции прыжка.

```
var playerJump = function() {
      player.jumpCondition = true;
      if ( player.rise === true && player.fall === false ) {
            player.y += …
      }
    else if ( player.fall === true && player.rise === false) {
            player.y -= …
      }
   timerID = requestAnimationFrame(playerJump);
   if (player.y <= bottom.height) {
      …
      } }
```

Значение true параметра jumpCondition сигнализирует, что функция прыжка активна, благодаря чему во время прыжка повторное нажатие пробела ничего не даст. jumpTimer – счетчик, значение которого является основным коэффициентом для расчета положения.

Первое условие выполнено, когда персонаж находится в стадии подъема, второе – в стадии падения. Они отличаются знаком величины, которая прибавляется к координате y игрока. Параметры были подобраны опытным путем.

После этих двух условных блоков делается вызов requestAnimationFrame(), который рекурсивно вызывает саму функцию прыжка. Такой цикл продолжается до тех пор, пока кубик не достигнет пола, в момент столкновения с полом всем флагам, отвечающим за состояние прыжка, присваивается значение false и отзывается requestAnimationFrame(), на чем и заканчивается выполнение функции.

Механизм столкновения с препятствиями – гораздо более сложная конструкция по сравнению с передвижением и прыжком. Для дальнейшего рассмотрения разделим этот вопрос на несколько частей.

Определение уровня в данной точке осуществляется циклическим сравнением координат игрока с координатами элементов карты. Высота самого близкого к игроку элемента карты объявляется высотой пола.

Столкновение с горизонтальными стенками реализовано похожим образом, однако есть небольшие отличия в зависимости от объекта, с которым происходит столкновение. При столкновении с элементом floor или step кубик продолжает скользить по нему до тех пор, пока не закончится длина элемента, столкновение с элементами spike, spikeM и palisade немедленно вызывает функцию distruct, что означает моментальный проигрыш и начало новой игры.

Столкновение с вертикальными стенками приводит к поражению в любом случае, независимо от объекта столкновения.

3.3. Общее описание игрового цикла.

После рассмотрения механизмов передвижения, прыжка и столкновений можно рассмотреть игровой цикл в принципе.

Начало игрового цикла происходит с вызовом функции gameLoop().

Внутри этой функции прописаны:

- обработчик addEventListener(), реагирующий на нажатие клавиш клавиатуры;
- функция определения пола;
- функция расчета горизонтальных и вертикальных столкновений;
- функция, занимающаяся анимацией персонажа, из которой перманентно вызываются функции отрисовки персонажа и карты.

Также gameLoop() содержит условие победы – прохождение достаточного расстояния по игровой карте.

По нажатию пробела произойдет прыжок, по нажатию ctrl – смена цветовой гаммы.

В случае неудачного столкновения игровой цикл начинается заново, кубик и карта возвращаются в исходное положение.

Заключение

В процессе выполнения курсового проекта были решены основные задачи по двум поставленным направлениям:

1) Исследование и моделирование системы с двумя степенями свободы.

2) Разработка игры на JavaScript.

Разработанные программы соответствуют всем поставленным в ходе работы целям: разработанная математическая модель и анимация двойного маятника соответствуют реально существующей модели при условии малости углов отклонения, игра же является структурной копией взятого для рассмотрения образца.

Распределение работы над проектом

Фролов М. – разработка игры и маятника.

Соколова Д. – вывод уравнений колебаний, оформление отчета.

Груздев И. – проектирование карты игрового уровня, разработка маятника.

Рыбаконенко М. – дизайн уровня, вывод уравнений колебаний.

Список литературы

1. http://www.math24.ru/двойной-маятник.html
2. И.Ю. Щеглова, А.А. Богуславский – Моделирование колебательных процессов (на примере физических задач), 2009.

Приложение 1. Вывод уравнений колебаний

Введем некоторые допущения:

- Стержни маятников невесомы и нерастяжимы.

- Масса каждого груза сосредоточена в одной точке.

- Трение в точках соединения отсутствует.

- Уравнения справедливы при малых колебаниях.

- Рассматривается маятник с равными длинами стержней.

- Колебания возникают за счет отклонения от положения равновесия ($\dot{\theta}$=0).

Важно отметить, что данные допущения вводятся не с самого начала рассмотрения, а в тех моментах, в которых эти допущения существенно меняют ход дальнейших рассуждений.

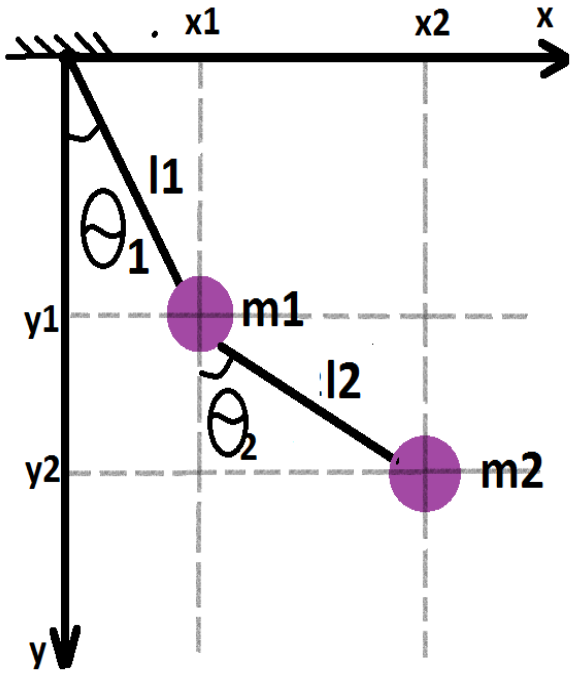

Рис. 10 Двойной маятник

Выведем уравнение Лагранжа:

В качестве обобщенных координат возьмем углы отклонения от вертикали:

$$\begin{cases} x_1 = L_1 \sin\theta_1 \\ y_1 = L_1 \cos\theta_1 \end{cases} \qquad \begin{cases} x_2 = L_1 \sin\theta_1 + L_2 \sin\theta_1 \\ y_2 = L_1 \cos\theta_1 + L_2 \cos\theta_1 \end{cases}$$

$$\Rightarrow \begin{cases} \dot{x_1} = L_1\dot{\theta_1}\cos\theta_1 \\ \dot{y_1} = -L_1\,\dot{\theta_1}\sin\theta_1 \end{cases} \qquad \begin{cases} \dot{x_2} = L_1\dot{\theta_1}\cos\theta_1 + L_2\dot{\theta_2}\cos\theta_2 \\ \dot{y_1} = -(L_1\,\dot{\theta_1}\sin\theta_1 + L_2\,\dot{\theta_2}\sin\theta_1) \end{cases}$$

$$v^2{}_1 = v^2{}_{x1} + v^2{}_{y1} = \dot{x}_1^2 + \dot{y}_1^2 = L_1{}^2\,\dot{\theta_1}{}^2(\ \sin^2\theta_1 + \cos^2\theta_1) = L_1{}^2\,\dot{\theta_1}{}^2$$

$$v^2{}_2 = v^2{}_{x2} + v^2{}_{y2} = \dot{x}_2^2 + \dot{y}_2^2 =$$

$$= (L_1\dot{\theta_1}\cos\theta_1 + L_2\dot{\theta_2}\cos\theta_2)^2 + (L_1\,\dot{\theta_1}\sin\dot{\theta_1} + L_2\,\dot{\theta_2}\sin\dot{\theta_2})^2$$

$$= L_1{}^2\,\dot{\theta_1}{}^2 + L_2{}^2\,\dot{\theta_2}{}^2 + 2L_1L_2\dot{\theta_1}\dot{\theta_2}(\ cos\theta_1 cos\theta_2 + sin\theta_1 sin\theta_2)$$

$$= L_1{}^2\,\dot{\theta_1}{}^2 + L_2{}^2\,\dot{\theta_2}{}^2 + 2L_1L_2\dot{\theta_1}\dot{\theta_2}cos(\theta_1 - \theta_2)$$

Составим Лагранжиан этой системы:

$$L = E_k - U = \frac{m_1 v_1{}^2}{2} + \frac{m_2 v_2{}^2}{2} + (mgy_1 + mgy_2)$$

$$= \frac{m_1}{2}L_1{}^2\,\dot{\theta_1}{}^2 + \frac{m_2}{2}(L_1{}^2\,\dot{\theta_1}{}^2 + L_2{}^2\,\dot{\theta_2}{}^2 + 2L_1L_2\dot{\theta_1}\dot{\theta_2}\cos(\theta_1 - \theta_2)) -$$

$$-(m_1gL_1\cos\theta_1 + m_2gL_2\cos\theta_1 + m_2gL_2\cos\theta_2) =$$

$$= \frac{(m_1+m_2)}{2}L_1{}^2\,\dot{\theta_1}{}^2 + \frac{m_2 v_2}{2}L_2{}^2\,\dot{\theta_2}{}^2 + m_2L_1L_2\dot{\theta_1}\dot{\theta_2}\ cos(\theta_1 - \theta_2) +$$

$$+(m_1+m_2)gL_1\cos\theta_1 + m_2gL_2\cos\theta_2$$

Считая колебания малыми:

$$\cos\alpha \approx 1 - \frac{\alpha^2}{2}$$

$$\cos(\alpha - \beta) \approx 1 - \frac{(\alpha - \beta)^2}{2} \approx 1$$

Примем: $m_2 g L_2 \cos\theta_2 = m_2 g L_2 \left(1 - \frac{{\theta_2}^2}{2}\right) = m_2 g L_2 - m_2 g L_2 \frac{{\theta_2}^2}{2}$

Тогда исходный Лагранжиан можно записать в виде:

$$L = E_k - U = \frac{(m_1+m_2)}{2} {L_1}^2 \dot{\theta_1}^2 + \frac{m_2}{2} {L_2}^2 \dot{\theta_2}^2 + m_2 L_1 L_2 \dot{\theta_1} \dot{\theta_2} -$$

$$- \frac{(m_1+m_2)}{2} g L_1 \dot{\theta_1}^2 - \frac{m_2}{2} g L_2 \dot{\theta_2}^2$$

Составим уравнения Лагранжа:

$$\begin{cases} \dfrac{\partial^2 L}{\partial \dot{\theta}_1 \partial t} - \dfrac{\partial L}{\partial \theta_1} = 0 \\ \dfrac{\partial^2 L}{\partial \dot{\theta}_2 \partial t} - \dfrac{\partial L}{\partial \theta_2} = 0 \end{cases}$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = (m_1+m_2) {L_1}^2 \dot{\theta}_1 + m_2 L_1 L_2 \dot{\theta}_2$$

$$\frac{\partial^2 L}{\partial \dot{\theta}_1 \partial t} = (m_1+m_2) {L_1}^2 \ddot{\theta}_1 + m_2 L_1 L_2 \ddot{\theta}_2$$

$$\frac{\partial L}{\partial \theta_1} = -(m_1+m_2) g L_1 \theta_1$$

Первое уравнение:

$$(m_1+m_2){L_1}^2\,\ddot{\theta}_1 + m_2L_1L_2\ddot{\theta}_2 + (m_1+m_2)gL_1\theta_1 = 0$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2{L_2}^2\,\dot{\theta}_2 + m_2L_1L_2\dot{\theta}_1$$

$$\frac{\partial^2 L}{\partial \dot{\theta}_1 \partial t} = m_2L_1L_2\,\ddot{\theta}_1 + m_2{L_2}^2\ddot{\theta}_2$$

$$\frac{\partial L}{\partial \theta_1} = -m_2L_2\theta_2$$

Второе уравнение:

$$m_2L_1L_2\,\ddot{\theta}_1 + m_2{L_2}^2\ddot{\theta}_2 + m_2L_2\theta_2 = 0$$

Имеем:

$$\begin{cases} (m_1+m_2){L_1}^2\,\ddot{\theta}_1 + m_2L_1L_2\ddot{\theta}_2 + (m_1+m_2)gL_1\theta_1 = 0 \\ m_2L_1L_2\,\ddot{\theta}_1 + m_2{L_2}^2\ddot{\theta}_2 + m_2L_2\theta_2 = 0 \end{cases}$$

Пусть $\theta(t) = \begin{pmatrix} \theta_1(t) \\ \theta_2(t) \end{pmatrix}$, $M = \begin{pmatrix} (m_1+m_2)L_1^2 & m_2L_1L_2 \\ m_2L_1L_2 & m_2L_2^2 \end{pmatrix}$

$$k = \begin{pmatrix} (m_1+m_2)gL_1 & 0 \\ 0 & m_2L_2 \end{pmatrix},\ 0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Тогда полученную систему можно переписать следующим образом

$M\ddot{\theta} + k\theta = 0$

В случае одного тела такое уравнение описывает свободные незатухающие колебания с определенной частотой.

В случае двойного маятника решение будет содержать колебания с двумя характерными частотами, которые можно найти следующим образом:

$\begin{pmatrix}\theta_1(t)\\ \theta_2(t)\end{pmatrix} = \mathrm{Re}\left[\begin{pmatrix}H1\\ H2\end{pmatrix} e^{i\omega t}\right]$, где H_1 и H_2-собственные векторы.

Значения частот ω найдем из следующего уравнения:

$$det(k - \omega^2 M) = 0$$

$$\begin{vmatrix}(m_1+m_2)gl_1 - \omega^2(m_1+m_2)l_1^2 & -\omega^2 m_2 l_1 l_2\\ -\omega^2 m_2 l_1 l_2 & m_2 g l_2 - \omega^2 m_2 l_2^2\end{vmatrix}$$

$$\begin{vmatrix}(m_1+m_2)gl_1 - \omega^2(m_1+m_2)l_1^2 & -\omega^2 m_2 l_1 l_2\\ -\omega^2 m_2 l_1 l_2 & m_2 g l_2 - \omega^2 m_2 l_2^2\end{vmatrix} = 0 \Rightarrow$$

$$\begin{vmatrix}(m_1+m_2)(gl_1 - \omega^2 l_1^2) & -\omega^2 m_2 l_1 l_2\\ -\omega^2 m_2 l_1 l_2 & m_2 l_2 (g - \omega^2 l_2)\end{vmatrix} = 0 \Rightarrow$$

$$\Rightarrow (m_1+m_2)m_2(gl_1 - \omega^2 l_1)(gl_2 - \omega^2 l_2^2) - \omega^4 m_2^2 l_1^2 l_2^2 = 0 \;\Big|\cdot(-1)$$

$$\omega^4 m_2^2 l_1^2 l_2^2 - (m_1 m_2 + m_2^2)(g^2 l_1 l_2 - g l_1 \omega^2 l_2^2 - g l_2 \omega^2 l_1^2 + \omega^4 l_1^2 l_2^2) = 0$$

$$-\omega^4 m_1 m_2 l_1^2 l_2^2 + \omega^2(g l_1 l_2^2 + g l_2 l_1^2)(m_1+m_2)m_2 -$$

$$-m_2(m_1+m)g^2 l_1 l_2 = 0 \mid : l_1 l_2 (-1)$$

$$\omega^4 m_1 m_2 l_1 l_2 + m_2(m_1+m_2)(gl_2 + gl_1)\omega^2 - m_2(m_1+m_2)g^2 = 0 \mid : m_2$$

$$\omega^4 m_1 l_1 l_2 - g(m_1+m_2)(l_1+l_2)\omega^2 + (m_1+m_2)g^2 = 0$$

Из этого квадратичного трёхчлена найдём квадраты собственных частот:

$$\omega_{1,2}^2 = g(m_1 + m_2)(l_1 + l_2) \pm (m_1 + m_2)^2(l_1 + l_2)^2 - 4m_1 l_1 l_2 (m_1 + m_2) g^2$$

$$\omega_{1,2}^2 = \frac{g(m_1 + m_2)(l_1 + l_2) \pm \sqrt{(m_1 + m_2)^2(l_1 + l_2)^2 - 4m_1 l_1 l_2 (m_1 + m_2) g^2}}{2m_1 l_1 l_2}$$

$$= \frac{g}{2m_1 l_1 l_2}((m_1 + m_2)(l_1 + l_2) \pm \sqrt{(m_1 + m_2)((m_1 + m_2)(l_1 + l_2)^2 - 4m_1 l_1 l_2)})$$

Пусть $l_1 = l_2 = l$

$$\omega_{1,2}^2 = \frac{g}{m_1 l}((m_1 + m_2) \pm \sqrt{(m_1 + m_2)\big((m_1 + m_2) - m_1\big)}\,) =$$

$$= \frac{g}{m_1 l}(m_1 + m_2 \pm \sqrt{(m_1 + m_2) m_2}\,) =$$

$$= \frac{g}{l}(1 + \frac{m_2}{m_1} \pm \sqrt{(1 + \frac{m_2}{m_1})\,\frac{m_2}{m_1}}\,).$$

$$]\ \mu = \frac{m_2}{m_1} => \omega_{1,2}^2 = \frac{g}{l}(1 + \mu \pm \sqrt{(1 + \mu)\,\mu}\,).$$

Найдём собственные векторы H_1 и H_2:

$$(k - \omega^2 M)H = 0$$

$$(k-\omega^2 M)H_1 = 0$$

$$\begin{pmatrix} m_1\, l & (1+\mu)(g-\omega_1^2 l) & -\omega_1^2\mu l \\ -\omega_1^2\mu l & \mu(g-\omega_1^2 l) & \end{pmatrix} \begin{pmatrix} H_{11} \\ H_{21} \end{pmatrix} = 0$$

$$(1+\mu)(g-\omega_1^2 l)H_{11} - \omega_1^2\mu l H_{21} = 0$$

$$(1+\mu)(g-g(1+\mu+\sqrt{(1+\mu)\,\mu}))\,H_{11} - g\mu(1+\mu+$$

$$+\sqrt{(1+\mu)\,\mu})\,H_{21} = 0$$

$$(1+\mu)\left(-g\mu - g\sqrt{(1+\mu)\mu}\right)H_{11} - (\,g\mu + g\mu^2 + g\mu\sqrt{(1+\mu)\mu}\,)H_{21} = 0$$

$$(-g\mu - g\sqrt{(1+\mu)\mu} - g\mu^2 - g\mu\sqrt{(1+\mu)\mu}\,)H_{11} + (-g\mu - g\mu^2 +$$

$$+ g\mu\sqrt{(1+\mu)\mu})\,H_{21} = 0$$

$$\frac{H_{21}}{H_{11}} = \frac{(1+\mu)\left(\mu+\sqrt{(1+\mu)\mu}\,\right)}{(\mu\left(1+\mu+\sqrt{(1+\mu)\mu}\,\right))} = -\sqrt{\frac{(1+\mu)\mu}{\mu^2}} = -\sqrt{\frac{1+\mu}{\mu}}$$

$$H_1 = \begin{pmatrix} H_{11} \\ H_{21} \end{pmatrix} = \begin{pmatrix} 1 \\ -\sqrt{\frac{1+\mu}{\mu}} \end{pmatrix}$$

Аналогично получаем:

$$H_2 = \begin{pmatrix} H_{12} \\ H_{22} \end{pmatrix} = \begin{pmatrix} 1 \\ \sqrt{\frac{1+\mu}{\mu}} \end{pmatrix}$$

$$(*)\ \theta(t) = \mathrm{Re}\left[\begin{pmatrix} H1 \\ H2 \end{pmatrix} e^{i\omega t}\right] = C_1 H_1 \cos(\omega_1 t + \psi_1) + C_2 H_2 \cos(\omega_2 t + \psi_2),$$

где $C_1, C_2, \psi_1, \psi_2 -$

постоянные, которые зависят от начального состояния системы.

В исследуемой модели начальное положение системы определяется следующими параметрами:

$$\theta_1(0),\ \theta_2(0),\ \dot{\theta}_1(0) = 0,\ \ \dot{\theta}_2(0) = 0,$$

то есть маятники не имеют начальной скорости.

Тогда уравнения (*) принимают вид:

$$\begin{pmatrix} \theta_1(t) \\ \theta_2(t) \end{pmatrix} = C_1 \begin{pmatrix} 1 \\ -\sqrt{\frac{1+\mu}{\mu}} \end{pmatrix} cosw_1 t + C_2 \begin{pmatrix} 1 \\ \sqrt{\frac{1+\mu}{\mu}} \end{pmatrix} cosw_2 t \quad (**)$$

Так как параметры C_1 и C_2 определяются исходя из начальных условий, выразим их через $\theta_1(0),\ \theta_2(0)$:

$$\begin{cases} \theta_1(0) = C_1 + C_2 \\ \theta_2(0) = \sqrt{\frac{1+\mu}{\mu}}(C_2 - C_1) \end{cases} => \begin{cases} C_1 = \theta_1(0) - C_2 \\ \theta_2(0) = \sqrt{\frac{1+\mu}{\mu}}(C_2 - \theta_1(0) + C_2) \end{cases}$$

$$\begin{cases} \sqrt{\frac{1+\mu}{\mu}}\, 2C_2 = \theta_2(0) + \sqrt{\frac{1+\mu}{\mu}}\, \theta_1(0) \mid \cdot \frac{1}{2}\sqrt{\frac{\mu}{1+\mu}} \\ C_1 = \theta_1(0) - C_2 \end{cases}$$

$$\begin{cases} C_2 = \frac{1}{2}\theta_1(0) + \frac{1}{2}\sqrt{\frac{\mu}{1+\mu}}\ \ \theta_2(0) \\ \\ C_1 = \frac{1}{2}\theta_1(0)\ - \frac{1}{2}\sqrt{\frac{\mu}{1+\mu}}\ \ \theta_2(0) \end{cases}$$

Подставим это в уравнение (**), получим

$$\theta_1(t) = \frac{1}{2}(\theta_1(0) - \sqrt{\frac{\mu}{1+\mu}}\ \ \theta_2(0))\cos\omega_1 t + \frac{1}{2}(\theta_1(0) + \sqrt{\frac{\mu}{1+\mu}}\ \ \theta_2))\cos\omega_2 t$$

$$\theta_2(t) = -\frac{1}{2}\sqrt{\frac{\mu}{1+\mu}}\ (\theta_1(0) - \sqrt{\frac{\mu}{1+\mu}}\ )\cos\omega_1 t + (\theta_1(0) + \sqrt{\frac{\mu}{1+\mu}}\ \ \theta_2(0))\cos\omega_2 t$$