

Программа моделирования анимации флага методом ключевых кадров

Студент: Сорокин А. П.
Руководитель: Клорикьян П. В.

Постановка задачи

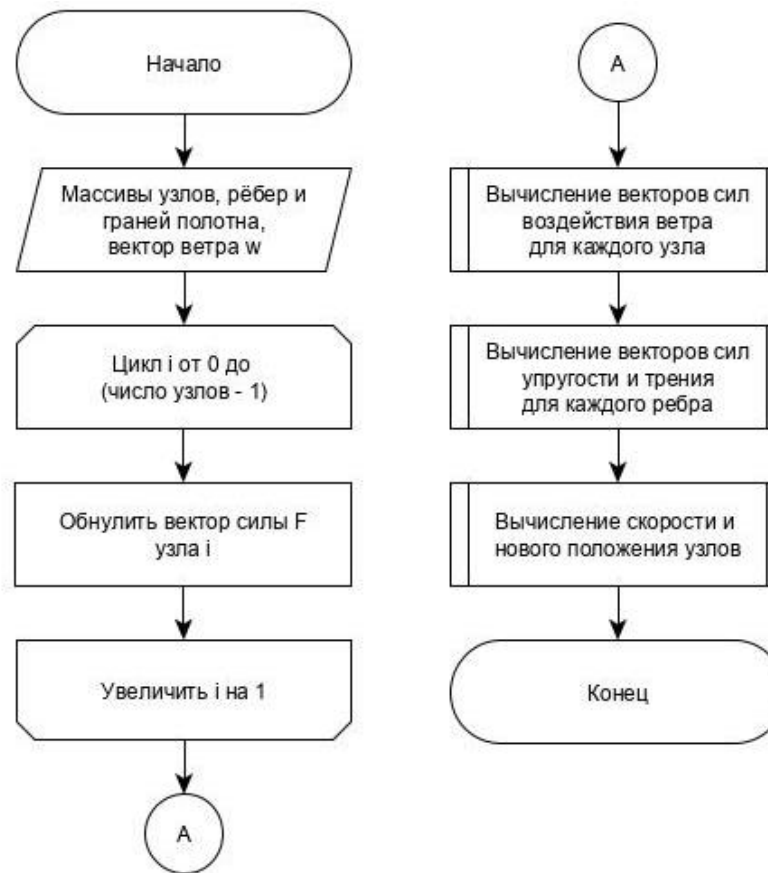
Необходимо разработать программу моделирования анимации развевающегося на ветру флага на флагштоке методом ключевых кадров. Для каждого промежуточного кадра программа должна рассчитывать текущее положение точек поверхности флага. Должен быть задан набор законов управления движением при переходе между двумя соседними ключевыми кадрами. Исследовать возможность учета освещенности, оптических свойств поверхностей.

Законы управления движением точек флага

$$\left\{ \begin{array}{l} \vec{F}_{grav} = m\vec{g} \\ \vec{F}_{spring} = k\vec{x} \\ \vec{F}_{fric} = -\mu\vec{F} \\ \vec{F}_{wi} = \vec{n} \cdot (\vec{n} \cdot \vec{w}) \\ \vec{F}_{wind} = \sum_{i=1}^n \vec{F}_{wi} \\ \vec{F} = \vec{F}_{grav} + \vec{F}_{spring} + \vec{F}_{fric} + \vec{F}_{wind} \\ \vec{a} = \frac{\vec{F}}{m} \\ \vec{v} = \vec{v}_0 + \vec{a}t \\ (x, y, z) = (x_0 + v_x, y_0 + v_y, z_0 + v_z) \end{array} \right.$$

- \vec{F}_{grav} — сила гравитации
- \vec{F}_{spring} — сила упругости ткани
- \vec{F}_{fric} — сила трения
- \vec{F}_{wi} — сила воздействия ветра на i — ую грань
- \vec{F}_{wind} — сила воздействия ветра на вершину
- \vec{F} — результирующая сила
- \vec{a} — ускорение точки
- \vec{v} — скорость движения точки
- \vec{v}_0 — предыдущее значение скорости
- (x, y, z) — новое положение точки
- (x_0, y_0, z_0) — предыдущее положение точки
- t — заданный промежуток времени
- m — масса точки флага
- μ — коэффициент трения

Алгоритм изменения положения точек флага



Общая схема алгоритма вычисления положения точек

Алгоритм изменения положения точек флага

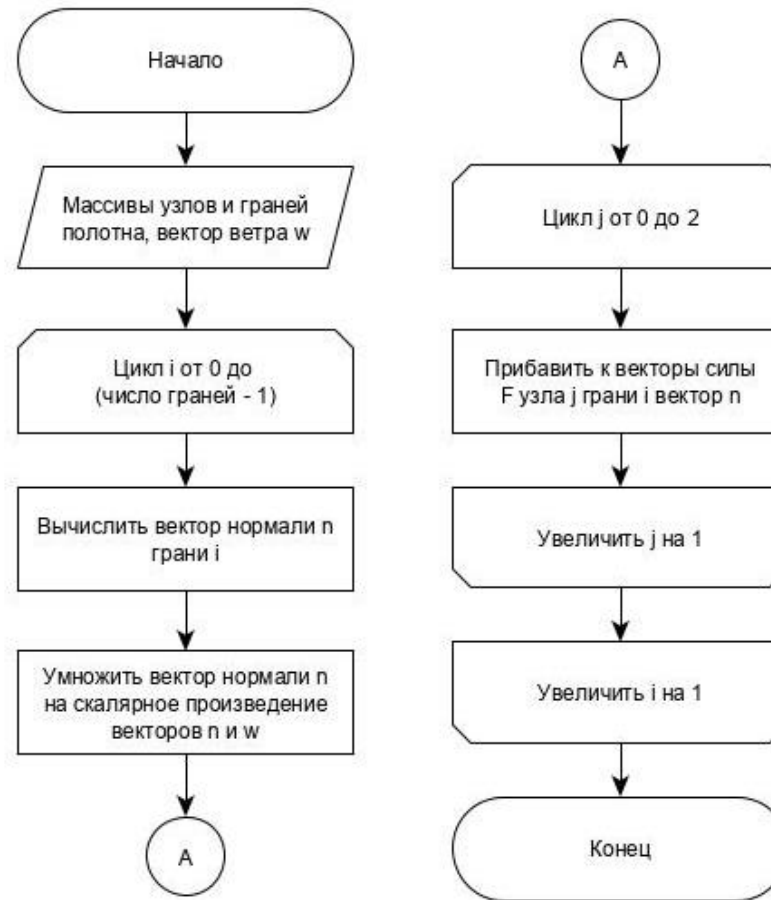


Схема алгоритма определения векторов сил воздействия ветра

Алгоритм изменения положения точек флага

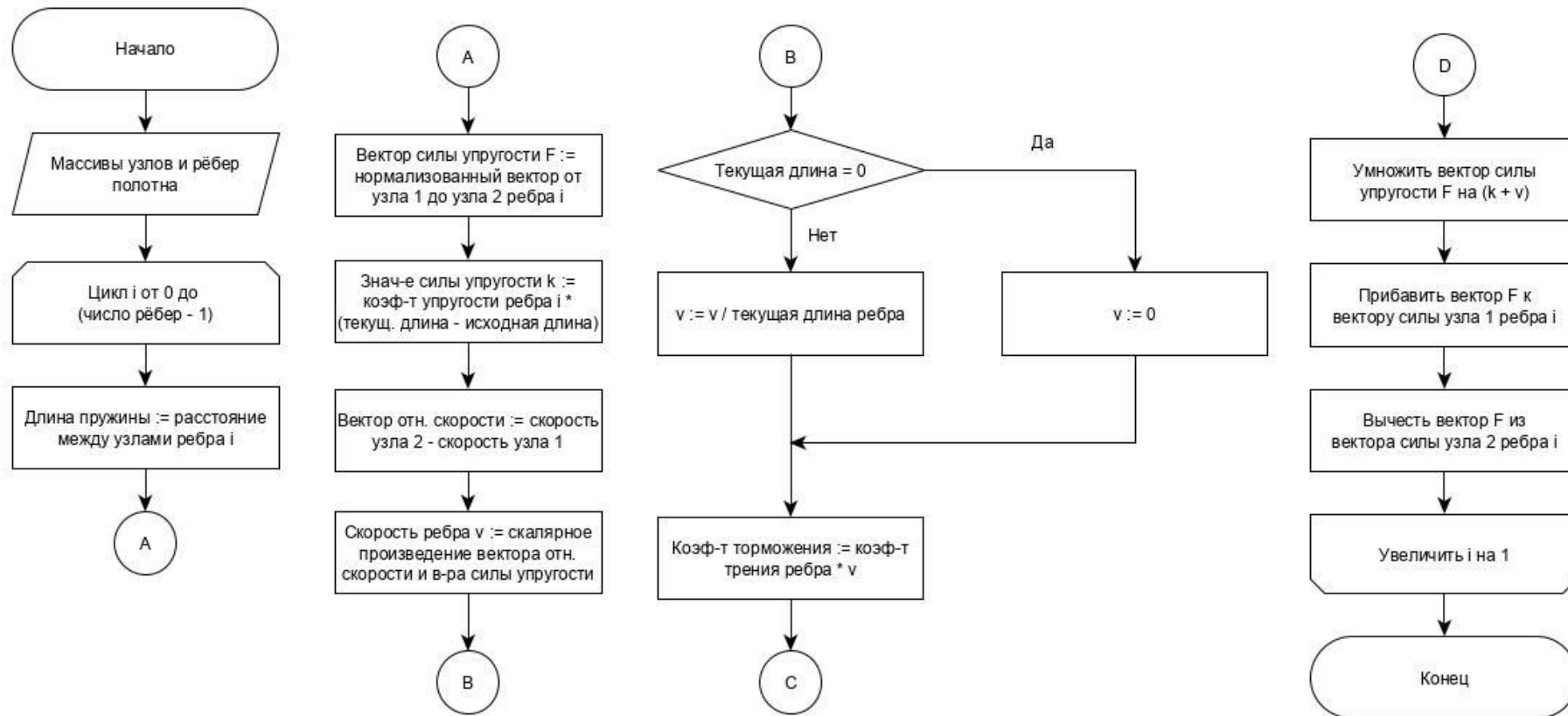


Схема алгоритма определения векторов сил упругости

Трёхмерные аффинные преобразования

- сдвиг на (dx, dy, dz)

$$\begin{cases} x_{new} = x + dx \\ y_{new} = y + dy \\ z_{new} = z + dz \end{cases}$$

- масштабирование относительно точки C с коэффициентами k_x, k_y, k_z

$$\begin{cases} x_{new} = x_C + k_x(x - x_C) \\ y_{new} = y_C + k_y(y - y_C) \\ z_{new} = z_C + k_z(z - z_C) \end{cases}$$

- поворот на угол ϕ вокруг оси X

$$\begin{cases} x_{new} = x \\ y_{new} = y \cdot \cos\phi - z \cdot \sin\phi \\ z_{new} = y \cdot \sin\phi + z \cdot \cos\phi \end{cases}$$

- поворот на угол ϕ вокруг оси Y

$$\begin{cases} x_{new} = z \cdot \sin\phi + x \cdot \cos\phi \\ y_{new} = y \\ z_{new} = z \cdot \cos\phi - x \cdot \sin\phi \end{cases}$$

- поворот на угол ϕ вокруг оси Z

$$\begin{cases} x_{new} = x \cdot \cos\phi - y \cdot \sin\phi \\ y_{new} = x \cdot \sin\phi + y \cdot \cos\phi \\ z_{new} = z \end{cases}$$

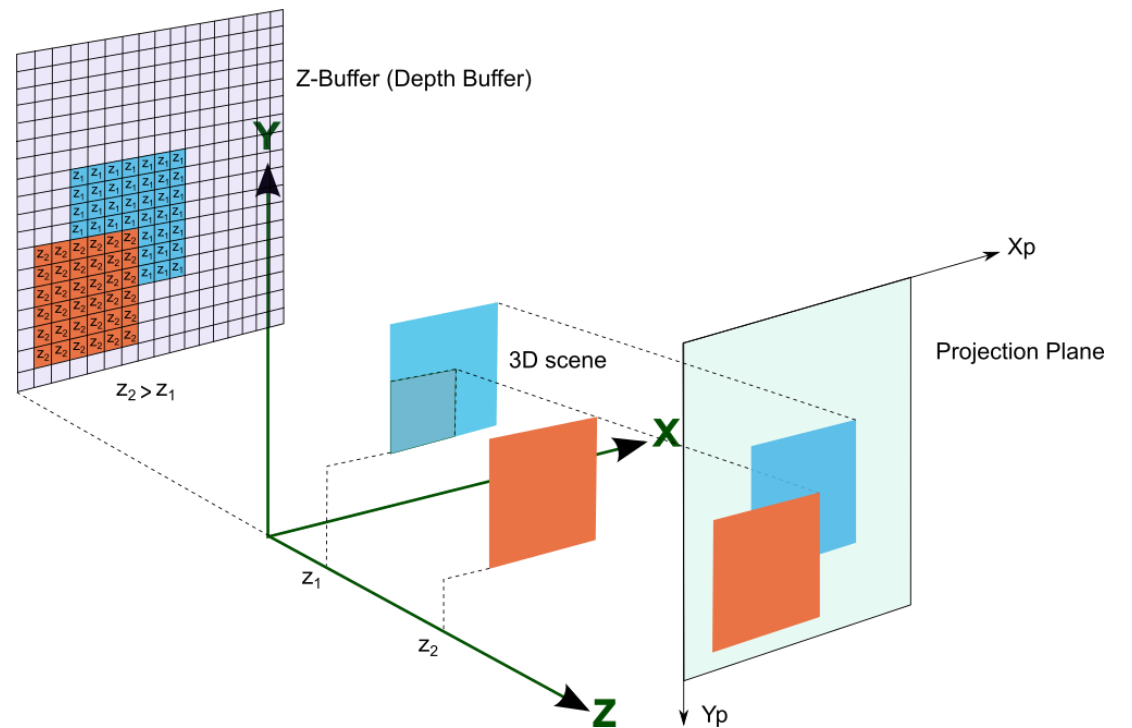
Графические алгоритмы

В ходе выполнения проекта был проведен анализ потенциально применимых графических алгоритмов и выбраны более подходящие из них, а именно:

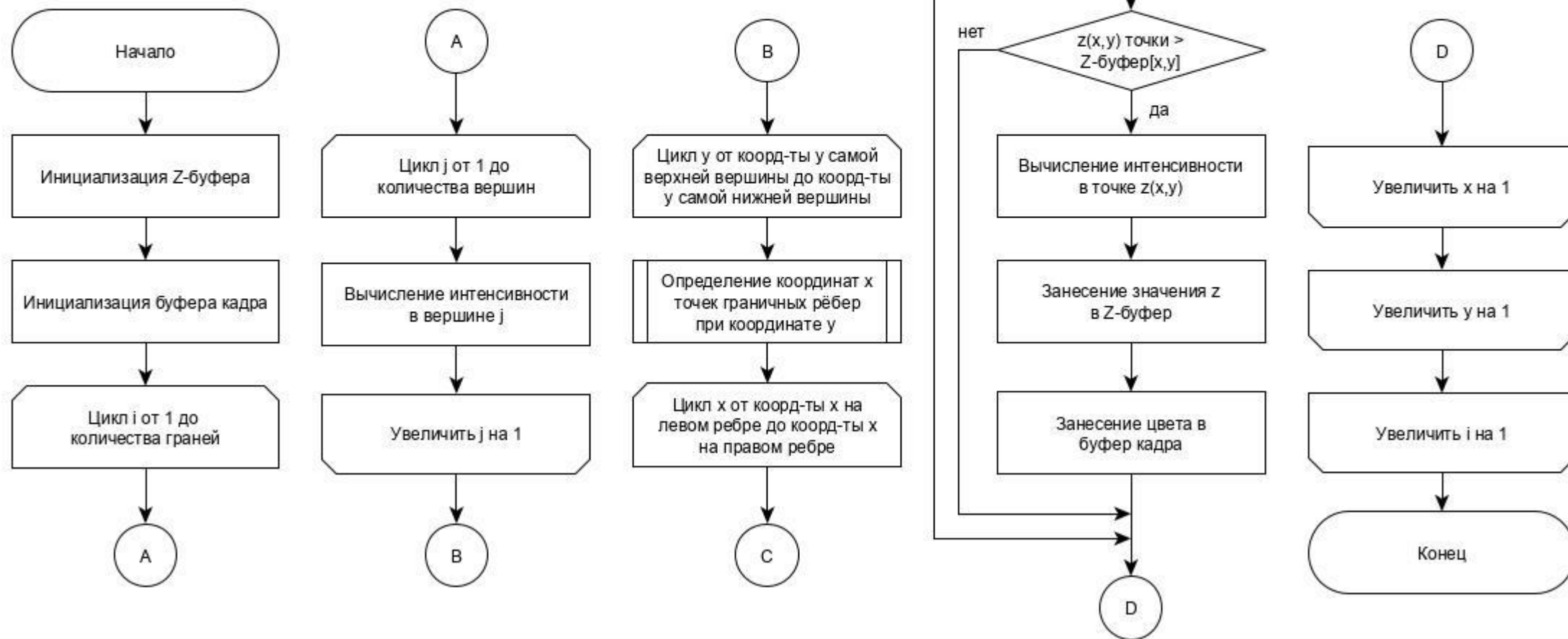
- Алгоритм Брезенхема (рисование линий)
- Алгоритм Z-буфера (удаление линий)
- Метод тонирования Гуро (закраска)
- Метод ключевых кадров (анимация)

Алгоритм Z-буфера

Принцип работы алгоритма z-буфер заключается в том, что мы определяем глубину каждой отрисовываемой точки, сравниваем эту глубину с глубинами других точек, имеющие те же координаты X и Y , и в случае, если новая точка находится ближе к наблюдателю, тогда мы заносим координаты новой точки в z-буфер и цвет в буфер кадра.



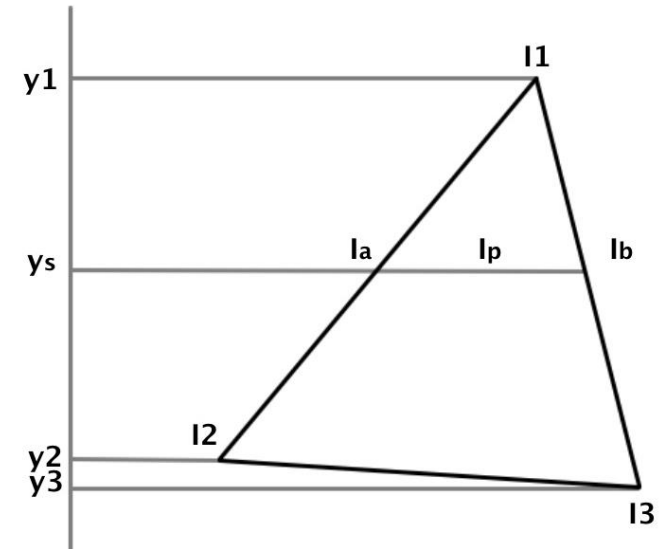
Алгоритм Z-буфера



Метод тонирования Гуро

Метод тонирования Гуро основан на интерполяции интенсивности, данный подход к закрашке объекта позволяет устранить дискретность изменения интенсивности.

Интерполяция интенсивностей работает следующим образом: для всех ребер запоминается начальная интенсивность, изменение интенсивности при каждом шаге по координате y . Затем, заполнение видимого интервала производится путем интерполяции между значениями интенсивности на ребрах, ограничивающих интервал.



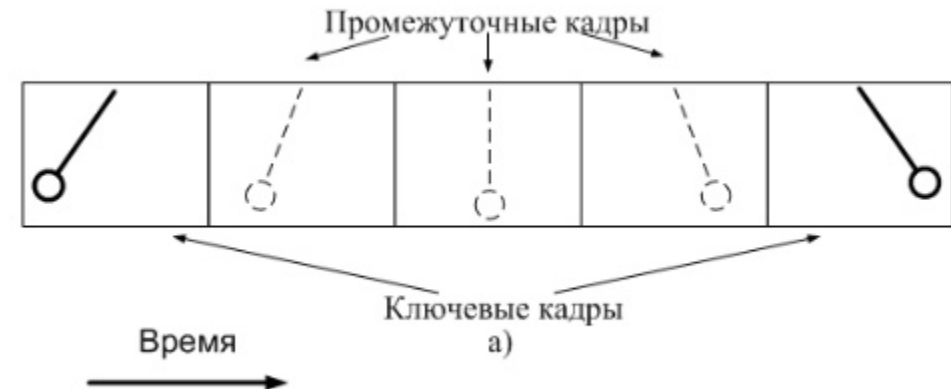
$$I_a = I_1 \frac{y_s - y_2}{y_1 - y_2} + I_2 \frac{y_1 - y_s}{y_1 - y_2}$$

$$I_b = I_1 \frac{y_s - y_3}{y_1 - y_3} + I_3 \frac{y_1 - y_s}{y_1 - y_3}$$

$$I_p = I_a \frac{x_b - x_p}{x_b - x_a} + I_b \frac{x_p - x_a}{x_b - x_a}$$

Анимация методом ключевых кадров

Основная идея использования ключевых кадров заключается в создании ключей анимации для начального и конечного положения объекта, при этом состояние объектов в промежуточных стадиях просчитывает компьютер.



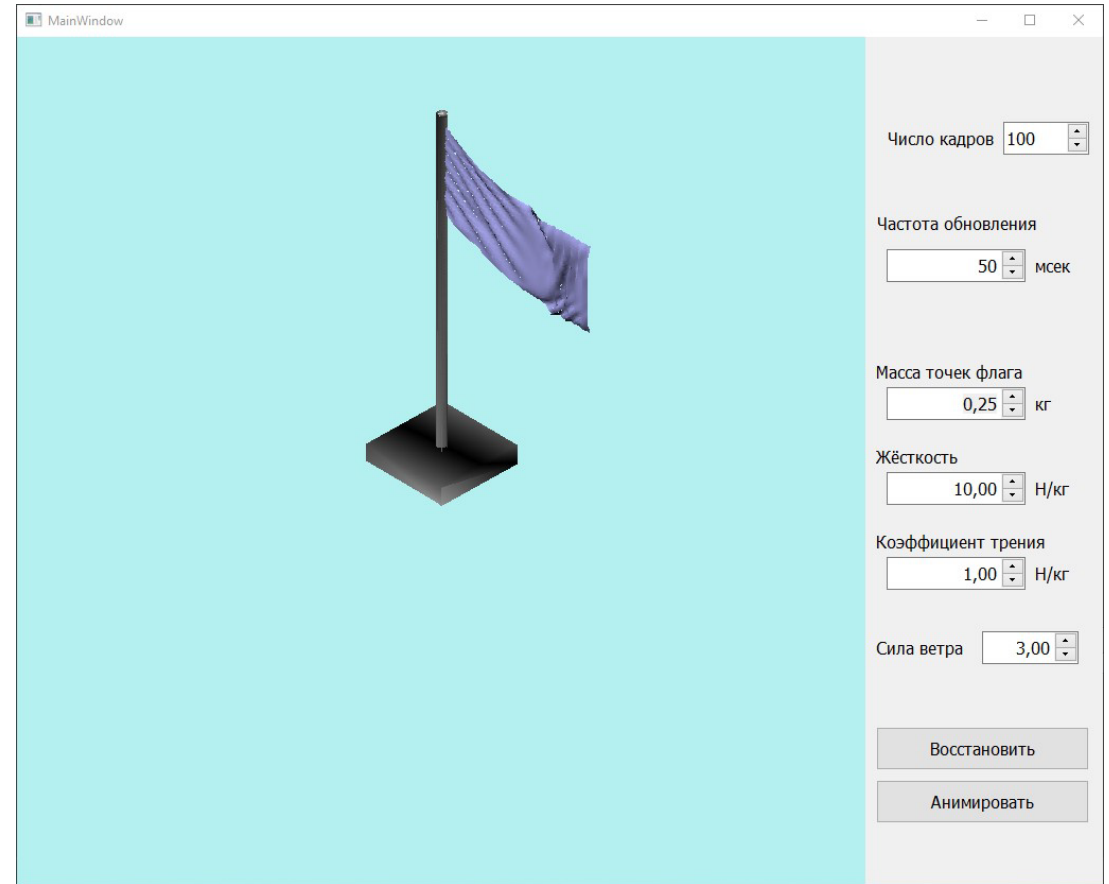
Используемые типы и структуры данных

Для решения поставленной задачи использовались следующие классы объектов:

- вершина
- грань (треугольник)
- модель (состоит из вершин и граней)
- узел флага (для добавления к вершине физических свойств)
- ребро флага (соединяет узлы)
- флаг (для непосредственной работы с узлами и рёбрами)

Пользовательский интерфейс и пример работы

Пользовательский интерфейс позволяет изменять физические параметры флага, связанные с его анимацией, а также сбрасывать все изменения. Перемещать, масштабировать и вращать изображение возможно с помощью клавиатуры и мыши.



Исследование временных затрат

Время работы алгоритма при разном числе потоков

Количество потоков	Время в тиках
1	43127480
2	35178242
3	28537882
4	24954742
5	20954733

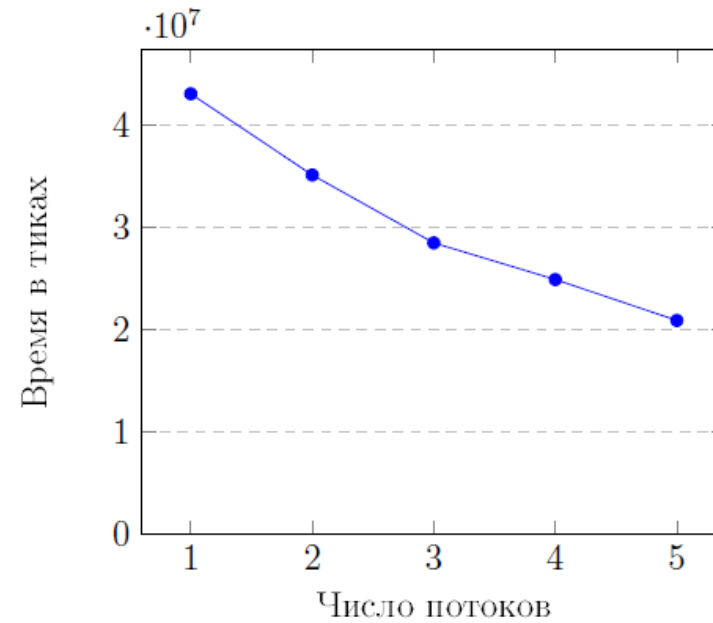


График зависимости времени работы алгоритма от числа потоков