Лабораторная работа №3

Аффинные преобразования на плоскости

Цель работы: получение навыков выполнения аффинных преобразований на плоскости и создание графического приложения с использованием GDI в среде Qt Creator.

Порядок выполнения работы

1. Разработать алгоритм и составить программу для построения на экране изображения в соответствии с номером варианта. В качестве исходных данных взять указанные в таблице №1.

Требования к программе

1. Разработать модуль для выполнения аффинных преобразований на плоскости с помощью матриц. В модуле должны быть реализованы перегруженные операции действия с матрицами (умножение), с векторами и матрицами (умножение вектора-строки на матрицу), конструкторы различных матриц (переноса, масштабирования, переноса, отражения).
2. В программе должна быть предусмотрена возможность ввода пользователем исходных данных (из правой колонки таблицы №1).
3. Разбить окно на 2 равные части. В левой части должна выводиться основная анимация, в правой части её отражение относительно вертикальной линии, проходящей через центр окна.
4. Изображение должно масштабироваться по центру левой и правой части окна с отступом 10 пикселей от границ и вертикальной линии и реагировать на изменение размера окна (см. пример проекта lab_1_CSharp).
5. Раскрасить (залить) примитивы (круги, многоугольники и др.) по собственному усмотрению.

Содержание отчёта

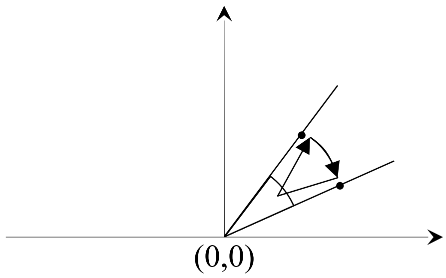
1. Название темы.
2. Цель работы.
3. Постановка задачи.
4. Вывод необходимых формул для построения изображения. Указать какие матрицы используются и в какой последовательности они умножаются для реализации анимации.
5. Текст программы.
6. Результат работы программы (снимки экрана).

Теоретические сведения

Аффинные преобразования упрощают выполнение простейших геометрических операций, избавляя от необходимости выводить геометрические формулы для расчёта сложных движений объектов в пространстве.

В компьютерной графике рассматривают 4 аффинных преобразования:

1. **Поворот** точки (x, y) вокруг начала координат на угол α.



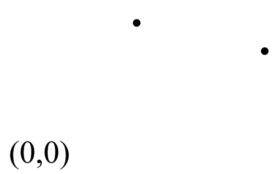
Новые координаты точки (x', y'), полученные в результате поворота, записываются в виде:

$$x' = x\cos\alpha - y\sin\alpha,$$
$$y' = x\sin\alpha + y\cos\alpha;$$

или, в матричной форме:

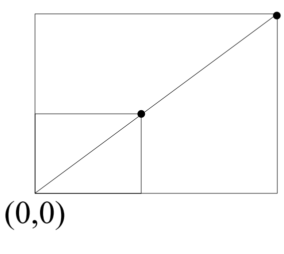
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

2. **Перенос** точки (x, y)вдоль вектора $(\Delta x, \Delta y)$.



$$x' = x + \Delta x,$$
$$y' = y + \Delta y.$$

3. **Масштабирование** относительно начала координат на величины $k_x, k_y > 0$.



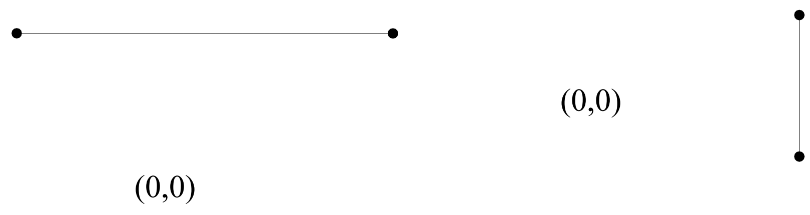
$$x' = k_x x,$$

$$y' = k_y y.$$

В матричной форме данное преобразование записывается в виде:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} k_x & 0 \\ 0 & k_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

4. **Отражение** точки (x, y) относительно оси абсцисс и/ или оси ординат.



Данное преобразование равносильно изменению знака у одной (x или y) или двух (x и y) координат точки. Отражение относительно оси ординат:

$$\begin{matrix} x' = -x, \\ y' = y, \end{matrix} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Отражение относительно оси абсцисс:

$$\begin{matrix} x' = x, \\ y' = -y, \end{matrix} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Из приведённых выше преобразований можно заметить, что перенос пока не выражен в матричном виде. Это объясняется тем, что операцию переноса невозможно записать с использованием матрицы размера 2×2. По этой причине переходят к матрицам размера 3×3.

Однородными координатами точки (x, y) называют тройку чисел $x_1 : x_2 : k$, связанные с исходными декартовыми координатами следующими соотношениями:

$$x = \frac{x_1}{k}, y = \frac{x_2}{k}.$$

Использование однородных координат позволяет охватить матричными преобразованиями все 4 аффинных преобразования. Переход к однородным координатам также сокращает количество операций деления в геометрических преобразованиях, что выгодно снижает вычислительные затраты. Операция деления при этом заменяется операцией умножения, а последняя выполняется процессором гораздо быстрее.

Таким образом, операция переноса с использованием однородных координат запишется в виде:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Перепишем остальные преобразования с использованием матриц размера 3×3.

Поворот:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Масштабирование:

$$\begin{pmatrix} x^{¿} \\ y^{¿} \\ 1 \end{pmatrix} = \begin{pmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Отражение относительно осей (соответственно ординат и абсцисс):

$$\begin{pmatrix} x^{¿} \\ y^{¿} \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \begin{pmatrix} x^{¿} \\ y^{¿} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

С использованием данных матриц, можно автоматически выполнять сложные геометрические преобразования.

Пример. Повернём точку $A(x, y)$ вокруг точки $O(x_0, y_0)$ на угол α. Для этого нужно выполнить 3 аффинных преобразования:

1. Перенести точки ***A*** и ***O*** вдоль вектора $(-x_0, -y_0)$таким образом, чтобы точка ***O*** оказалась в начале координат:
$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\Delta x \\ 0 & 1 & -\Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$
Обозначим полученные координаты как $B(x_1, y_1)$.
2. Выполнить поворот точки ***B*** вокруг начала координат на угол на угол α:
$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}.$$
3. Выполнить обратный перенос точки ***B*** вдоль вектора (x_0, y_0):
$$\begin{pmatrix} x_3 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}.$$

В последнем выражении заменим однородные координаты $x_2 : y_2 : 1$ на предпоследнее выражение:

$$\begin{pmatrix} x_3 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}.$$

Аналогично заменим $x_1 : y_1 : 1$:

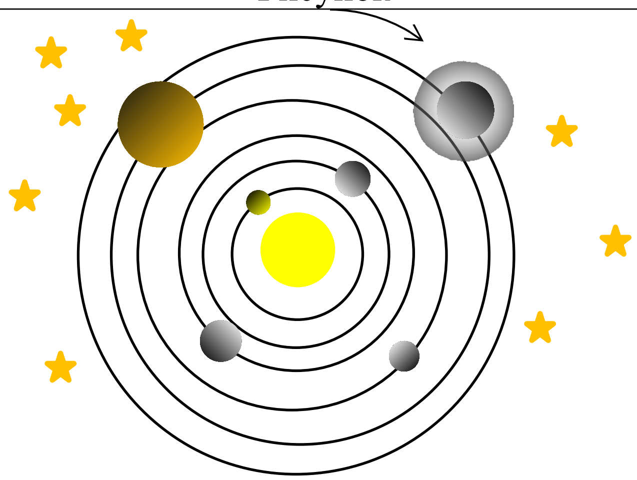
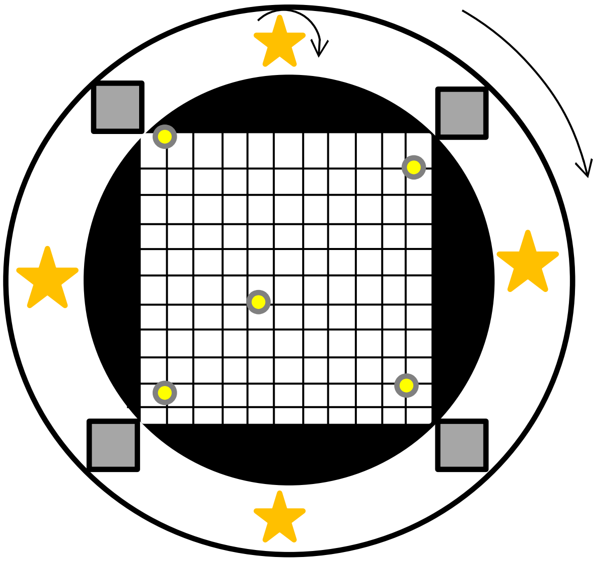
$$\begin{pmatrix} x_3 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -\Delta x \\ 0 & 1 & -\Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

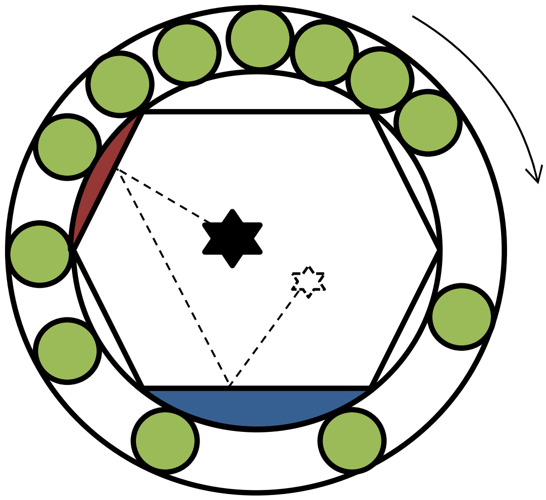
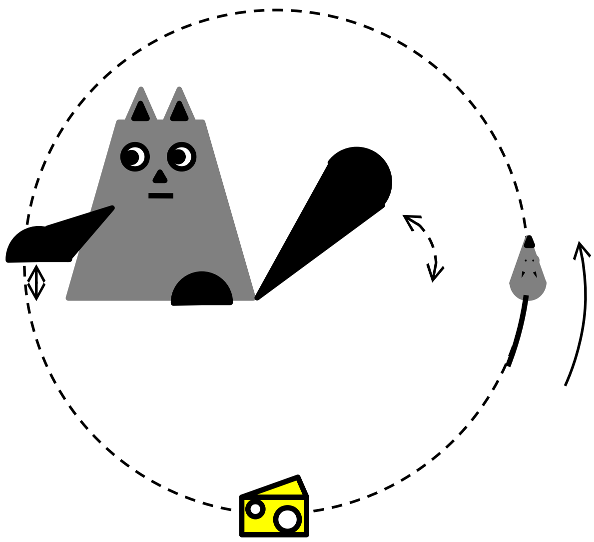
Произведение матриц можно посчитать заранее и обозначить как ***M***, тогда:

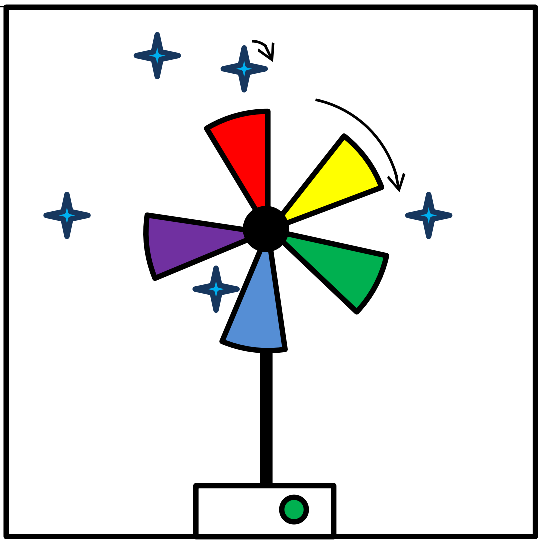
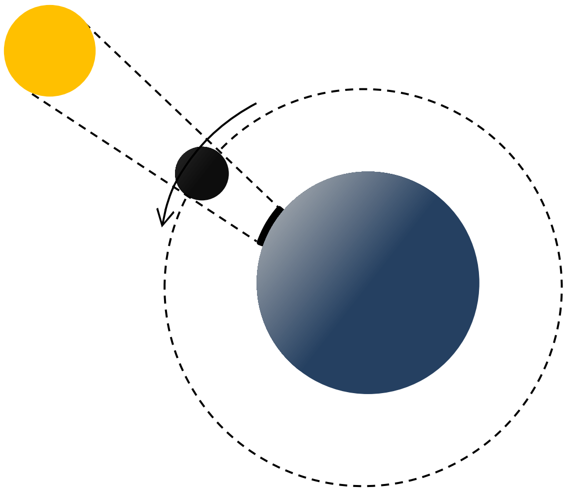
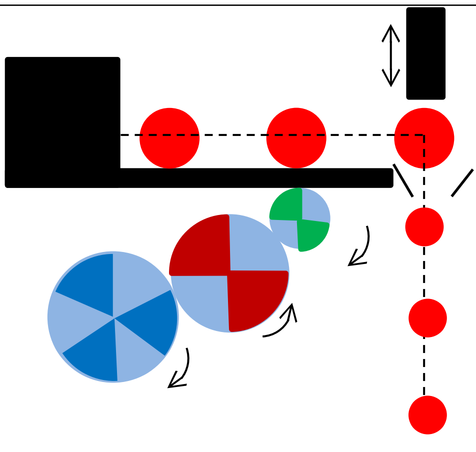
$$\begin{pmatrix} x_3 \\ y_3 \\ 1 \end{pmatrix} = M \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

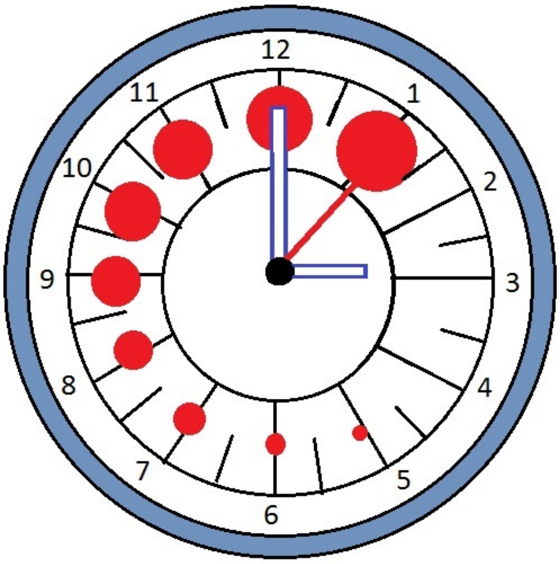
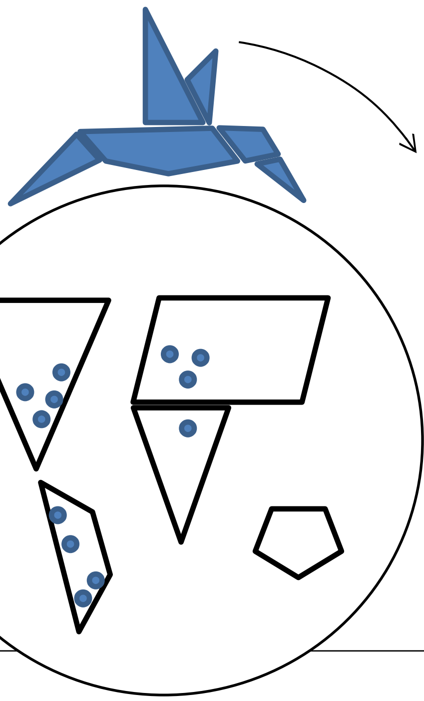
Вид матрицы ***M*** зависит от угла α и координат точки $O(x_0, y_0)$. Изменяя эти параметры, можно поворачивать точки вокруг других точек на произвольный угол. К примеру, чтобы повернуть треугольник вокруг точки ***O***, нужно три раза умножить матрицу ***M*** на однородные координаты трёх вершин треугольника.

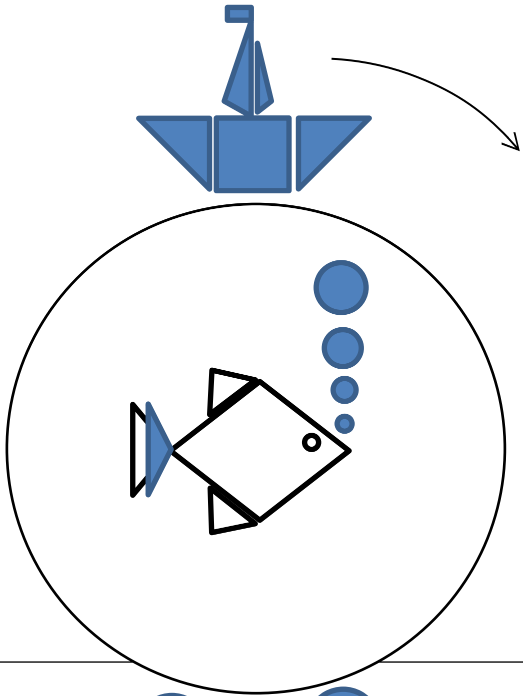
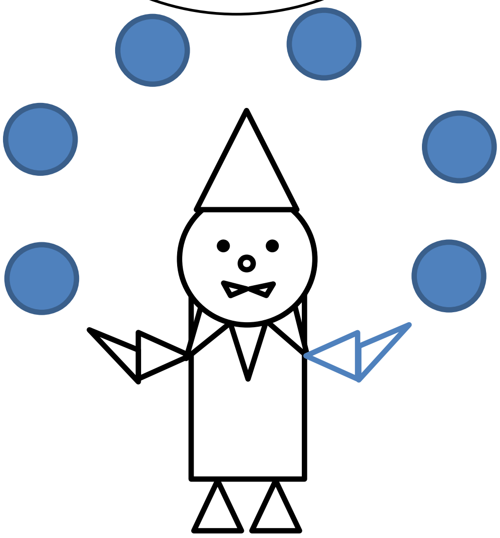
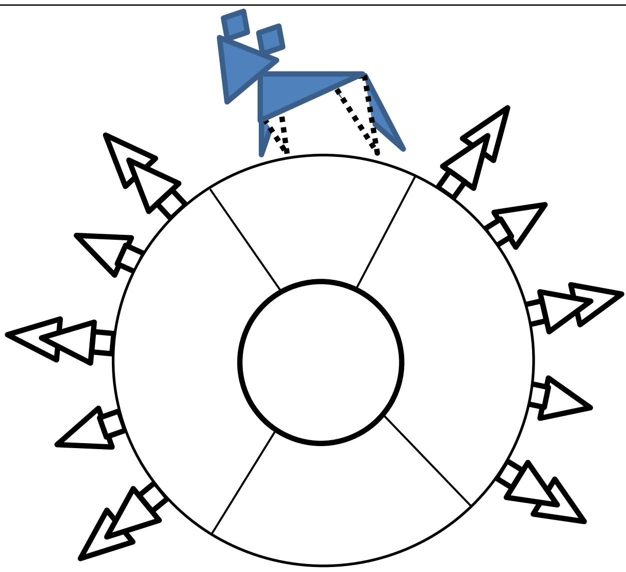
Таблица №1

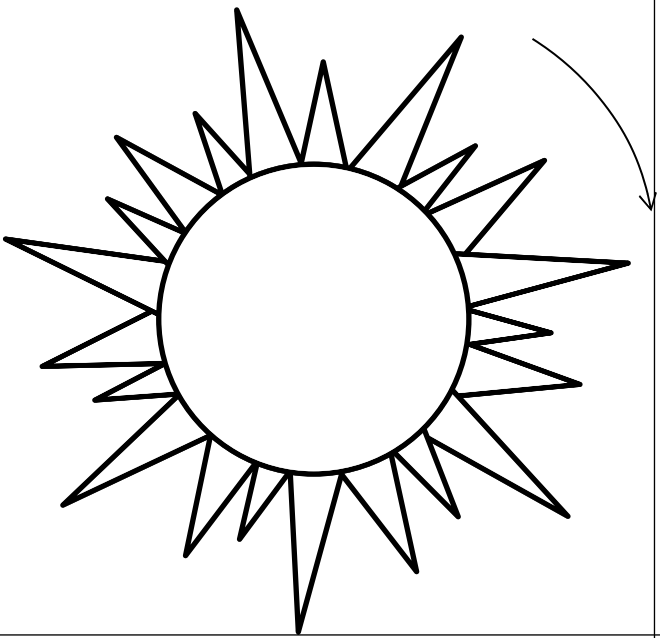
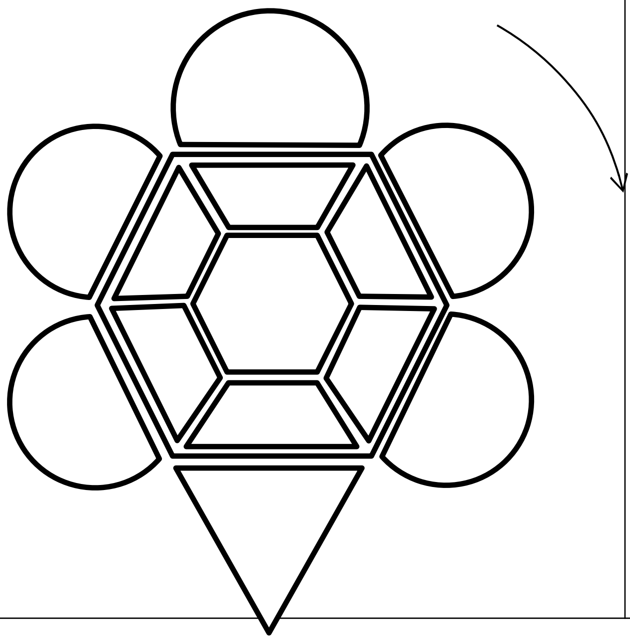
Вариант	Рисунок	Исходные данные
1		*n* – количество планет от солнца Реализовать вращение планет вокруг солнца и анимацию: периодическое изменение размера (увеличение/уменьшение) звезд, расположенных вокруг планет в случайных местах. Солнце освещает планеты только с ближайшей к нему стороне. Реализовать вращение с разной скоростью планет вокруг солнца. У двух планет должны быть кольца и спутники.
2		n-начальное количество шаров на сетке Реализовать вращение внешнего кольца и анимацию: когда квадраты внешнего кольца совпадают с углами центрального квадрата. На углах центрального квадрата появляются новые круги; движение кругов случайно по решетке квадрата; при столкновении двух кругов один из них растворяется; звезды вращаются вокруг своих центров.

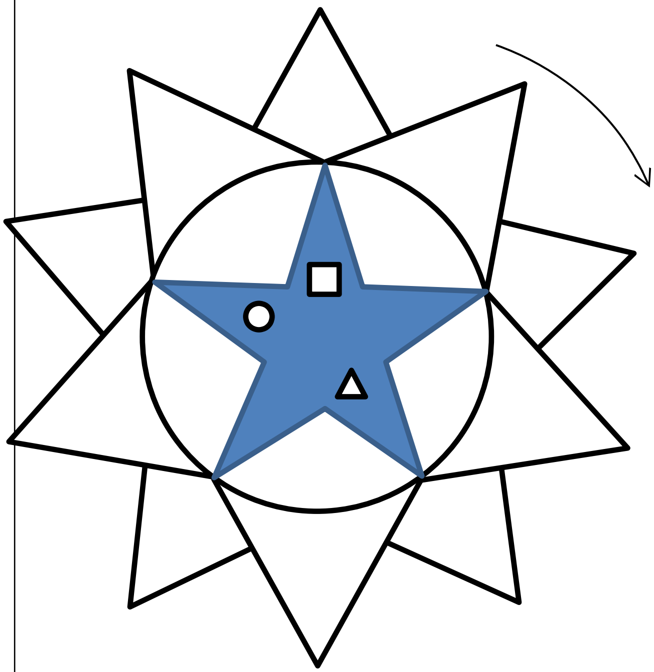
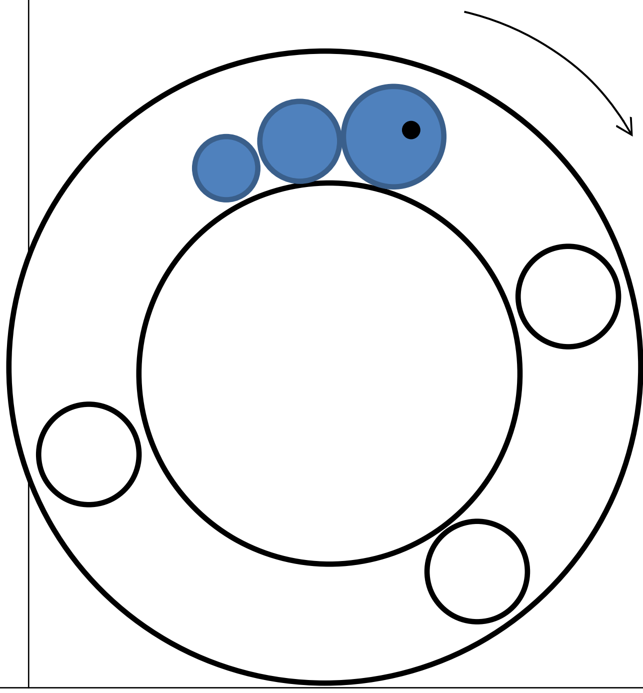
3		*n*-количество кругов во внешнем круге Реализовать вращение кругов вокруг центра с ускорением от верхней точки круга и замедлением в нижней точке. Реализовать анимацию: Произвольное движение 6-ти конечная звезда вращается внутри многоугольника. При столкновение звезды со стороной многоугольника, сектор, прилегающий к данной стороне меняет цвет, звезда несколько секунд уменьшается, а затем увеличивается до первоначального размера.
4		Реализовать движение мыши по кругу и следующую анимацию: Кот поднимает и опускает хвост. Зрачки кота следят за мышью. При приближении к коту, мышь ускоряется, а кот бьет лапой по мышке. При приближение мыши к сыру, сыр пропадает, а при отдалении от него медленно появляется.
5		Реализовать вращение лопастей вентилятора и следующую анимацию: при нажатии на кнопку, вентилятор начинает работать и кнопка горит зеленым. В выключенном состоянии кнопка горит красным. В центре вентилятора появляются 4-конечные звезды и движутся по спирали в направление движения лопастей с уменьшением первоначальной скорости. Звезды появляются в центре вентилятора в произвольный момент времени. Звезды вращаются вокруг своих центров, каждая с разной скоростью. Появление звезд случайно. При приближение к

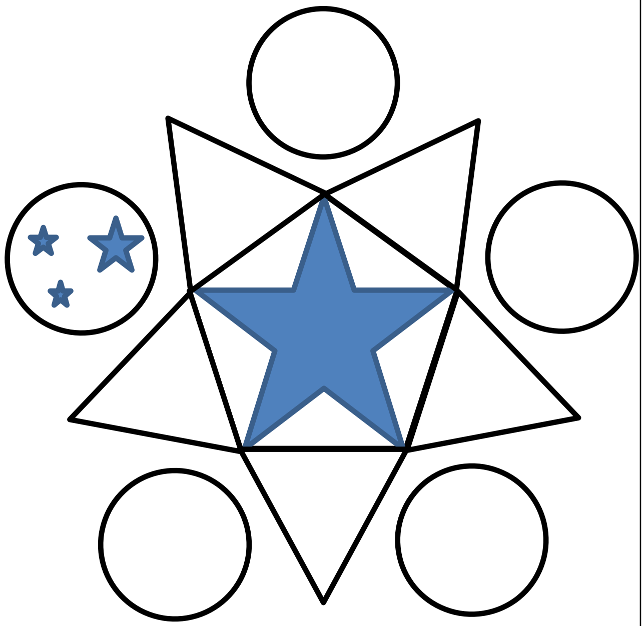
		границе рисунка звезды растворяются.
6		Реализовать вращение луны вокруг планеты. Солнце освещает только те половины луны и планеты, которые ближе к нему. При движении луны тень, создаваемая ей на планете, отображается в виде дуги (можно сделать отображение в виде пятна, как в действительности).
7		k-количество оборотов первого вала Реализовать вращение валов и следующую анимацию: движение по конвейеру красных шаров. Движение вверх вниз пресса с надавливанием на шары. Когда пресс нажимает на шар, то он попадает в воронку, уменьшается в размере и продолжает движение вниз.
8		n-количество шаров следа, k-количество делений между основными (на промежутке 5 минут) Реализовать вращение стрелок и следующую анимацию: часы показывают текущее время. Когда секундная стрелка попадает на деление появляется большой шар, с

		течением времени последний полностью исчезает. При наступлении нового часа соответствующее ему число на циферблате немного увеличивается в размере и затем плавно уменьшается. При наступлении новой минуты соответствующее ей число на циферблате меняет цвет, а затем плавно возвращается на первоначальный.
Вариант	Рисунок	Исходные данные
9		Реализовать вращение птицы вокруг шара и следующую анимацию: на материках (геометрические фигуры внутри круга) появление случайным образом кругов случайного размера по случайным координатам. Раскраску фигур сделать произвольной.

10		Реализовать вращение корабля вокруг шара и следующую анимацию: движение плавника рыбки и появление пузырьков, постепенное их увеличение и попадание по достижению ими поверхности.
11		Реализовать вращение шаров вокруг центра носа клоуна и следующую анимацию: по щелчку мыши шары собираются в один и происходит движение правой рукой вверх и вниз, после чего вращение продолжается.
12		Реализовать вращение шара вместе с расположенными на нем елками, раскрасить фигуру произвольным образом. Реализовать следующую анимацию: движение лап лисы как показано на картинке (пунктиром – новое положение лап лисы)

13		Реализовать вращение фигуры и следующую анимацию: появление лучиков случайной ширины и длинны, постепенное их уменьшение до полного исчезновения.
14		n – количество граней многоугольника. Реализовать вращение фигуры и изменение количества граней.

16		Реализовать вращение фигуры и появление внутри звезды квадратов, треугольников и окружностей по произвольным координатам, вращение их вокруг своей оси и уменьшение их до полного исчезновения.
17		Реализовать вращение змейки по окружности и следующую анимацию: на пути змейки произвольно появляются шары, когда змейка проходит мимо них, они исчезают и длинная змейки увеличивается на 1 шар. Когда змейка встречается со своим хвостом, количество окружностей в ее теле снова становится = 3

18		Реализовать вращение фигуры и следующую анимацию: случайное появление звездочек внутри окружностей по случайным координатам, вращение их вокруг своей оси и постепенное уменьшение, вплоть до полного исчезновения.