ГУАП

КАФЕДРА № 42

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| канд. техн. наук, доцент |  |  |  | А. В. Аграновский |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 3 |
| ИССЛЕДОВАНИЕ АФФИННЫХ ПРЕОБРАЗОВАНИЙ В ПРОСТРАНСТВЕ |
| по курсу: |
| КОМПЬЮТЕРНАЯ ГРАФИКА |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 4326 |  |  |  | Г. С. Томчук |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

1. Цель работы

Целью работы является

1. Формулировка задания

Необходимо реализовать программу изображения движущегося по окружности бумеранга (вариант 20).

1. Теоретические сведения

Аффинные преобразования — это класс геометрических преобразований, которые сохраняют прямолинейность и параллельность линий, но могут изменять расстояния и углы. Эти преобразования включают в себя такие операции, как сдвиг, масштабирование, поворот, отражение и их комбинации. Аффинные преобразования важны для работы с графикой, так как позволяют изменять объекты на плоскости, не нарушая их топологических свойств. Основные виды аффинных преобразований:

1. Сдвиг

- Сдвиг перемещает объект на заданное расстояние вдоль осей координат. Это преобразование не изменяет форму и размеры объекта.

- Для вектора сдвиг описывается как добавление фиксированного вектора , что выражается формулой:

- Матрица преобразования:

2. Масштабирование

- Масштабирование изменяет размеры объекта, умножая его координаты на коэффициенты масштабирования вдоль осей и .

- Формула для масштабирования:

- Матрица масштабирования:

- Если , это называется гомотетия, которая сохраняет пропорции фигуры.

3. Поворот

- Поворот изменяет ориентацию объекта вокруг начала координат на угол θ.

- Формула поворота:

- Матрица поворота:

- Поворот на плоскости сохраняет длину отрезков и углы между ними.

4. Отражение

- Отражение симметрично отображает фигуру относительно некоторой оси (например, оси или произвольной прямой).

- Матрица отражения относительно оси :

- Матрица отражения относительно оси :

5. Сжатие и сдвиг

- Сжатие изменяет форму объекта, двигая его точки вдоль одной из осей, что приводит к искажению углов, но сохраняет параллельность прямых.

- Формула преобразования:

- Матрица сжатия:

или

Композиция аффинных преобразований. Аффинные преобразования можно комбинировать путем перемножения их матриц. Итоговое преобразование зависит от порядка выполнения операций. Например, сначала можно выполнить сдвиг, затем поворот, а потом масштабирование, что приведет к композиции этих преобразований.

Однородные координаты. Для удобства работы с аффинными преобразованиями часто используются однородные координаты, добавляя к двумерной системе координат *(x, y)* дополнительную координату *w*, которая позволяет представлять преобразования в виде матрицы 3x3. Это упрощает матричные вычисления и позволяет записывать все виды аффинных преобразований единообразно.

Применение. Аффинные преобразования широко применяются в компьютерной графике для моделирования движений объектов, их масштабирования, изменения ориентации и других операций. Они также используются для преобразования изображений, таких как трансформация координат изображения или картографическая проекция.

1. Описание алгоритма решения

Задача заключается в. Алгоритм решения можно разделить на несколько этапов:

1. Определение фигуры бумеранга:

* На первом этапе необходимо задать форму бумеранга на плоскости.

1. Выбор языка программирования и используемых библиотек

Для написания программы анимации бумеранга я выбрал язык программирования Python. Данный язык имеет богатую экосистему библиотек, в том числе и необходимых для работы с отрисовкой и графикой. Помимо этого, я уже имел опыт работы с данным языком, что ускорило и облегчило рабочий процесс.

Используемые библиотеки:

* NumPy — используется для работы с матрицами и векторами, что упрощает вычисления аффинных преобразований, таких как поворот, сдвиг и масштабирование. Это стандартная библиотека для эффективных математических операций в Python.
* Matplotlib — отвечает за визуализацию и анимацию движения бумеранга. В частности, модуль matplotlib.animation позволяет легко создавать анимации, обновляя координаты объектов на каждом шаге.

Обе библиотеки широко используются в научных и инженерных вычислениях, что делает их подходящими для задач, связанных с компьютерной графикой и аффинными преобразованиями.

1. Описание разработанной программы

Программа (см. Приложение А) реализует. Основные компоненты программы:

* 1. Определение фигуры бумеранга. Фигура бумеранга задается как треугольник с тремя вершинами. Эти вершины определяются с помощью двумерных координат.

1. Скриншоты, иллюстрирующие результаты работы программы

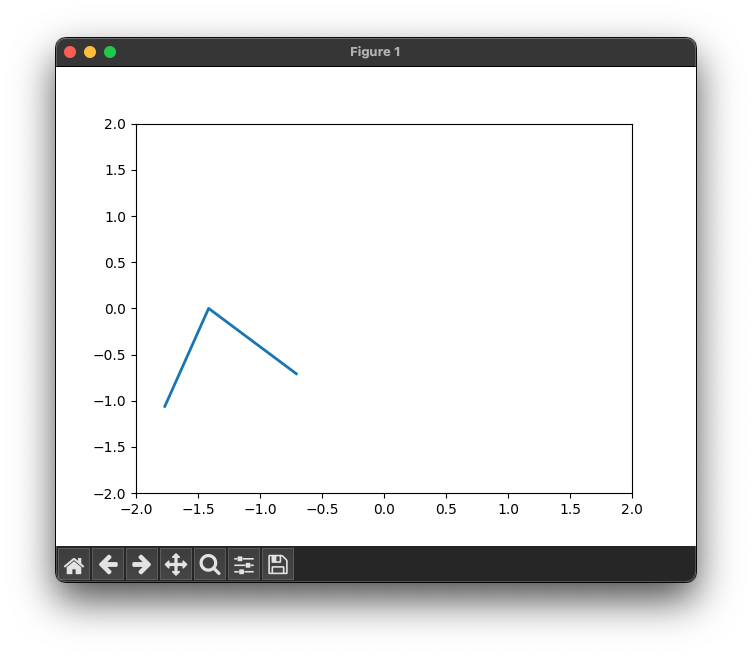


Рисунок 1

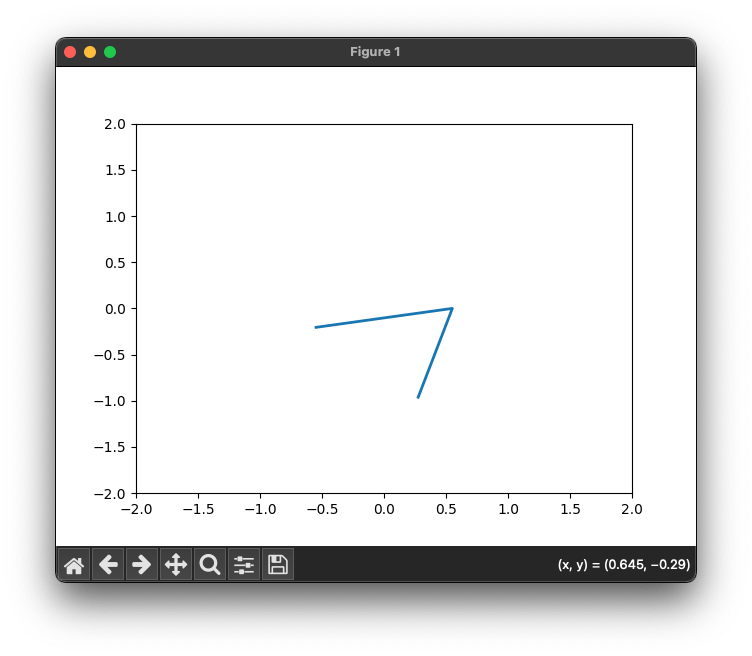


Рисунок 2

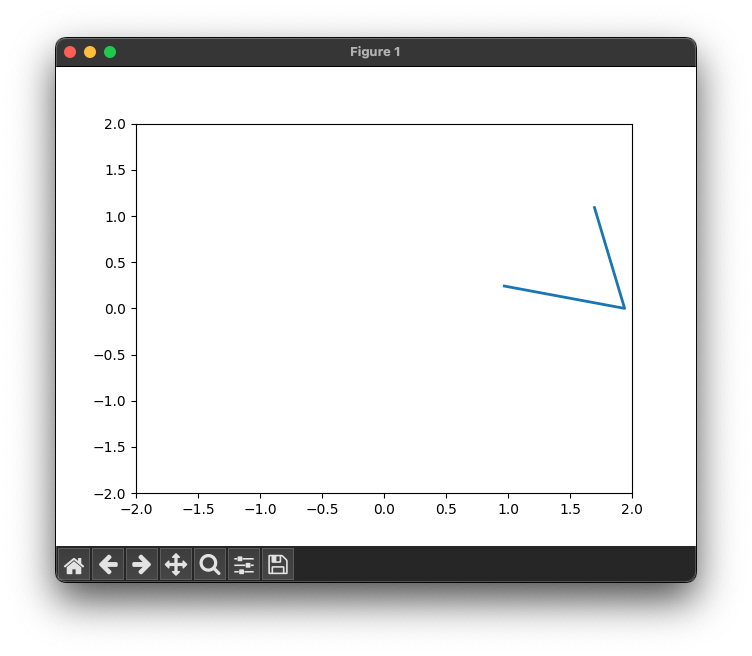


Рисунок 3

1. Вывод

В ходе выполнения лабораторной работы были изучены.

Полученные знания и навыки:

* Понимание различных типов линейных преобразований и их влияния на геометрические фигуры.

Проблемы и их решения:

* 1. Точность вычислений.

В результате работы была успешно разработана программа, демонстрирующая.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аграновский, А. В. Использование методов преобразования координат для формирования растровых изображений. Учебно-методическое пособие / А. В. Аграновский. — СПб.: Редакционно-издательский центр ГУАП, 2024. — 40 с.
2. Бобылев, С. И. Основы компьютерной графики: учебное пособие / С. И. Бобылев. — М.: Издательство МГТУ им. Н. Э. Баумана, 2010. — 224 с.
3. Ласков, А. В. Математические основы компьютерной графики / А. В. Ласков, Ю. Г. Лоскутов. — СПб.: Питер, 2008. — 352 с.
4. NumPy Documentation [Электронный ресурс] — Режим доступа: https://numpy.org/doc/, свободный. — (дата обращения: 29.09.2024).

ПРИЛОЖЕНИЕ A

import numpy as np  
import matplotlib.pyplot as plt  
import matplotlib.animation as animation  
  
# Задание фигуры бумеранга  
boomerang = np.array([  
 [0, 0], # Вершина 1  
 [1, 0], # Вершина 2  
 [0.5, 1] # Вершина 3  
])  
  
  
# Функция для создания матрицы поворота  
def rotation\_matrix(angle):  
 return np.array([  
 [np.cos(angle), -np.sin(angle)],  
 [np.sin(angle), np.cos(angle)]  
 ])  
  
  
# Функция для вращения и перемещения бумеранга  
def transform\_boomerang(boomerang, angle, radius):  
 # Вращение (умножение матрицы)  
 rotated\_boomerang = boomerang @ rotation\_matrix(angle)  
  
 # Перемещение по окружности  
 cx = radius \* np.cos(angle) # координаты центра окружности по оси X  
 cy = radius \* np.sin(angle) # # координаты центра окружности по оси Y  
  
 # Сдвиг фигуры (сложение матриц)  
 return rotated\_boomerang + np.array([cx, cy])  
  
  
# Настройки анимации  
# figure - объект фигуры, представляет собой окно для графического отображения. В нем размещаются оси и другие элементы визуализации  
# axes - объект осей внутри фигуры. Это область, где происходит построение графиков и анимаций, включая рисование объектов и управление их свойствами  
figure, axes = plt.subplots()  
axes.set\_xlim(-2, 2)  
axes.set\_ylim(-2, 2)  
line, = axes.plot([], [], lw=2) # объект линии контура бумеранга  
  
  
# Функция обновления для анимации  
def update(frame):  
 angle = np.radians(frame)  
 transformed\_boomerang = transform\_boomerang(boomerang, angle, radius=1)  
  
 line.set\_data(transformed\_boomerang[:, 0], transformed\_boomerang[:, 1])  
 return line,  
  
  
# Создание анимации  
# anim - объект анимации, отвечает за обновление отображения на каждом кадре анимации и управляет ее динамическим поведением  
anim = animation.FuncAnimation(figure, update, frames=range(0, 360), interval=50, blit=True)  
plt.show()