Отчёт о модификации проекта «Изображение проекции полиэдра»

В.В. Варгасов

Е.А. Роганов

А.И. Александров

26.04.2024

Table of Contents

## Постановка задачи

### Точная постановка задачи

Все рёбра делятся на три класса: полностью видимые, видимые частично и полностью невидимые. Модифицируйте эталонный проект таким образом, чтобы определялась и печаталась следующая характеристика полиэдра: сумма длин проекций полностью видимых рёбер, проекция центра которых находится на расстоянии строго меньше от прямой .

### Интерпретация задачи и идея решения

В данной задаче требуется рассматривать проэкции вершин полиэдра на плоскость , и складывать длины таких полностью видимых ребер полиэдра, проекции центров которых лежат строго внутри области, ограчиченной прямыми и . Также необходимо реализовать графический интерфейс, отображающий заданную область для визуального контроля правильности работы модификации.

## Подробное описание решения задачи на модификацию

1. В класс Edge был добвален аттрибут special, который указывает на то, является ли тот или иной экземпляр класса ребром удовлетворяющим условию или нет путем сравнения среднего арифмитического координат концов отрезко по оси абсцисс с и .
2. В связи с тем, что по условию требуется вычислять сумму длин только видимых ребер, удовлетворяющих условию, указанному выше, было принято решение модифицировать метод shadow так, чтобы в случае невырожденной тени, значение аргумента special обнулялось.
3. Для отображения области, указанной в условии задачи, модифицруем код класса TkDrawer, добавив ему метод draw\_zone, изображающий прямоугольник зеленого цвета.
4. Также модифицируем метод draw класса Polyedr, добавив после вызова метода clean вызов метода draw\_zone.
5. Добавим классу Polyedr аттрибут length, к которому в рамках метода draw будем прибавлять значение длины ребра умножив его на значение аттрибута special и разделив на коэффициент гомотетии.
6. Были написаны соответствующие тесты, иллюстрирующие правильность работы всех написанных методов и модификации в целом.

## Команды, использовавшиеся при получении итоговых отчетов в заданных форматах

Здесь pdf.latex и HTML.html5 — переименованные файлы шаблонов.

Команда для получения отчета в формате pdf, с использованием заданного шаблона.

pandoc --template pdf.latex -s --toc --lua-filter ./include-code-files.lua report.md -o report.pdf

Команда для получения отчета в формате html, с использованием заданного шаблона.

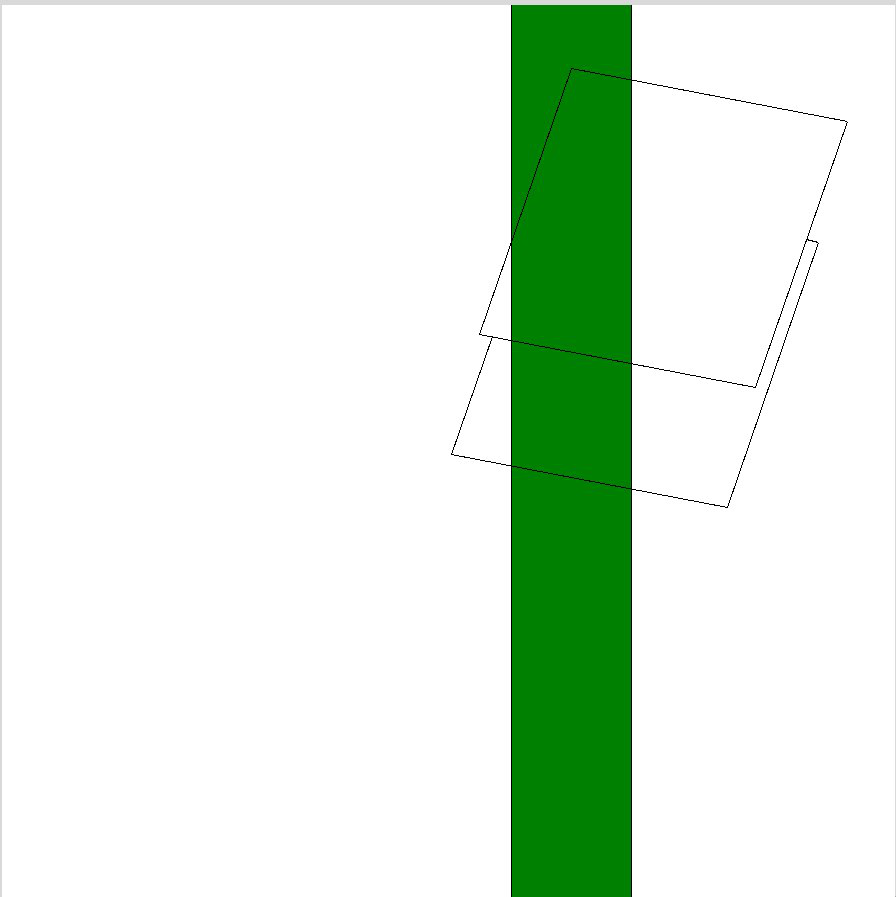
pandoc -o report.html -f markdown -t html -s --toc --lua-filter ./include-code-files.lua --mathjax --template html.html5 report.md

Команда для получения отчета в формате docx.

pandoc -s report.md --toc --lua-filter ./include-code-files.lua -o report.docx

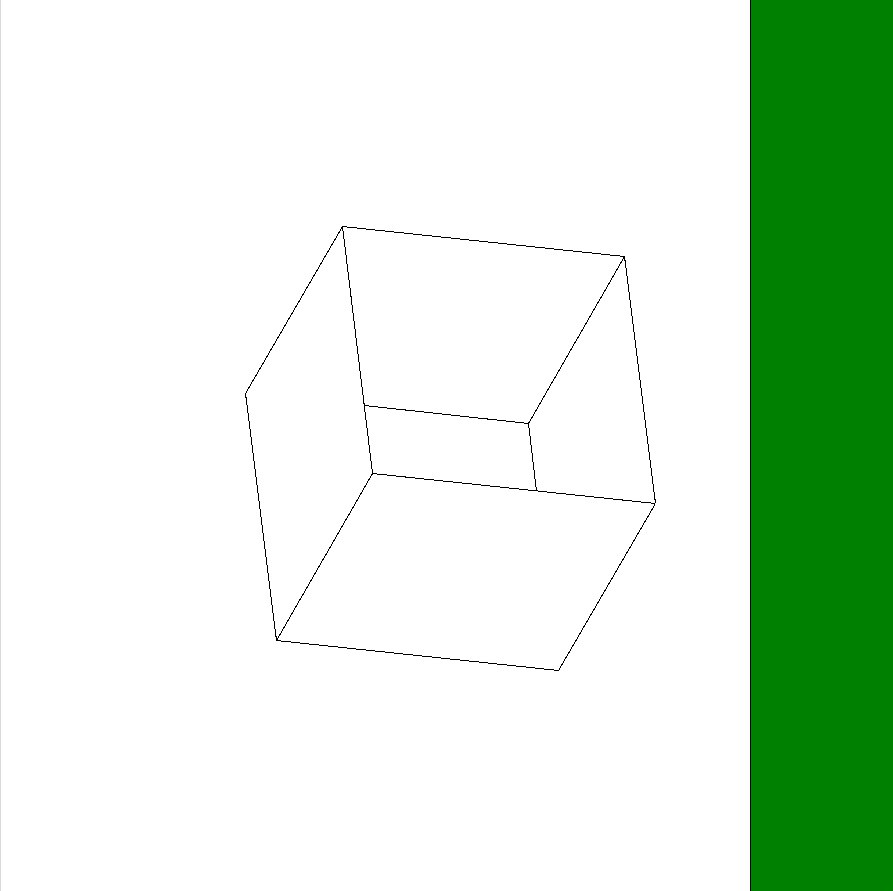
## Примеры тестов для поддтверждения корректности работы модификации

* Коробка, шесть из восьми вершин которой попадают в заданное кольцо.



class TestModPolyedr(unittest.TestCase):  
  
 @classmethod  
 def setUpClass(self):  
 fake\_file\_content = """40.0 45.0 -30.0 -60.0  
8 2 8  
0.0 0.0 0.0  
5.0 0.0 0.0  
5.0 5.0 0.0  
0.0 5.0 0.0  
1.0 1.0 3.0  
6.0 1.0 3.0  
6.0 6.0 3.0  
1.0 6.0 3.0  
4 1 2 3 4  
4 5 6 7 8  
"""  
 fake\_file\_path = 'data/holey\_ccc.geom'  
 with patch('shadow.polyedr.open'.format(\_\_name\_\_),  
 new=mock\_open(read\_data=fake\_file\_content)) as \_file:  
 self.polyedr = Polyedr(fake\_file\_path)  
 \_file.assert\_called\_once\_with(fake\_file\_path)  
  
 def test\_length(self):  
 from common.tk\_drawer import TkDrawer  
 tk = TkDrawer()  
 self.polyedr.draw(tk)  
 self.assertAlmostEqual(self.polyedr.length, 15.0)

* Куб, шесть из восьми вершин которого попадают в заданное кольцо.



class TestPolyedr(unittest.TestCase):  
  
 @classmethod  
 def setUpClass(self):  
 fake\_file\_content = """200.0 45.0 45.0 30.0  
8 4 16  
-0.5 -0.5 0.5  
-0.5 0.5 0.5  
0.5 0.5 0.5  
0.5 -0.5 0.5  
-0.5 -0.5 -0.5  
-0.5 0.5 -0.5  
0.5 0.5 -0.5  
0.5 -0.5 -0.5  
4 5 6 2 1  
4 3 2 6 7  
4 3 7 8 4  
4 1 4 8 5"""  
 fake\_file\_path = 'data/holey\_box.geom'  
 with patch('shadow.polyedr.open'.format(\_\_name\_\_),  
 new=mock\_open(read\_data=fake\_file\_content)) as \_file:  
 self.polyedr = Polyedr(fake\_file\_path)  
 \_file.assert\_called\_once\_with(fake\_file\_path)  
  
 def test\_length(self):  
 from common.tk\_drawer import TkDrawer  
 tk = TkDrawer()  
 self.polyedr.draw(tk)  
 self.assertAlmostEqual(self.polyedr.length, 0.0)