**Проект**

**Cоздание банка изображений с геопривязкой и другой информации о них**

**ВЫПОЛНИЛ:**

**Быбин Роман Евгеньевич 11В**

2021

**Постановка задачи**:

Программа нацелена на использование одним человеком. Выполняет добавление в базу данных Sqlite3 изображения и предусматривает работу с ним.

**Функции, реализуемые модулем**:

* 1. Добавление в базу данных Sqlite3 информации о изображении и пути к нeму.
  2. Удаление из базы данных изображения и всей информации о нем.
  3. Поиск изображений по введенным координатам
  4. Возможность вывода изображения ,которые было ранее добавлено в базу данных Sqlite3 .
  5. Сравнение нового введенного изображения с каждым из элементов базы данных Sqlite3.

**Руководство пользователя**:

1. Поле для выбора дальнейших действий(1:Добавление изображения в базу данных. 2:Удаление изображения из базы данных. 3:Поиск по координатам в базе данных. 4:Вывод изображения по имени)
2. При выборе “Добавление изображения в базу данных” пользователю будет предложено заполнить 5 полей: 1-Название изображения,2-Дополнитьельная информация,3-X координта,4-Y координата, 5-полное имя файла.
3. При выборе “Удаление изображения из базы данных” пользователю будет предложено ввести название изображения, которое необходимо удалить из базы данных.
4. При выборе “ Поиск по координатам в базе данных ” пользователю будет предложено ввести координаты двух точек(x и y координат),в пределах которых будет осуществлен поиск.
5. При выборе “ Вывод изображения по имени ” пользователю будет предложено ввести полное имя изображения, которое необходимо вывести на экран.

**Описание алгоритмов:**

1. Для работы с большим объемом данных используется база данных sqlite3 (Подключается: import sqllite3 ).Для работы с ней мы в начале программы создаем ее и связываем с переменной (db), также добавляем курсор базы данных (sql=db.cursor())
2. Программа создает базу данных ,с которой нам предстоит работать, c 5 столбцами (name,inf,x,y,photo)
3. Если пользователь решил добавить новое изображения, то он должен будет ввести 5 значений, которые будут связаны со столбцами базы данных.Затем программа проверяет существует ли строка с такими же значениями в базе данных. Если существует , то не производит выполнение, а просто печатает “Такая запись уже имеется”.
4. Для работы с изображения используется библиотека piglet.
5. Если такой записи не существует, то программа создает новую строку в БД и добавляет в столбцы этой строки введение пользователем значения(pic\_name=name,inf=pic\_inf,x=pic\_x,y=pic\_y,photo=pic\_way).
6. При успешном добавлении программа создаст окно для вывода:(window\_boss = pyglet.window.Window()

window\_boss.set\_minimum\_size(640, 480))

определит папку с изображениями:

pyglet.resource.path = ['./pikc']

pyglet.resource.reindex()

и выведет изображение на экран, запомнив переменную pic\_way:

программа хранит путь к папке, в которой лежат все изображения, а также хранит полное название файла каждого изображения(pic\_way)

gamer\_image = pyglet.resource.image(pic\_way)

gamer = pyglet.sprite.Sprite(gamer\_image, x=0, y=0)

@window\_boss.event

def on\_draw():

window\_boss.clear()

gamer.draw()

pyglet.app.run()

1. При удалении изображения пользователь введет название изображения, которое необходимо удалить: переменная led. После программа найдет в базе данных строки ,в которых name = led, и удалит их. Если таких строк нет, программа не произведет удаление. sql.execute("DELETE FROM pictures WHERE name='led'")
2. Если пользователь решил вывести изображение из базы данных на экран, то ему необходимо ввести полное имя изображения, которое он хочет вывести. После этого программа найдет в папке, раннее указанной, это изображение и выведет ранее указанным методом.