

PyQt5FileManager

учебный проект



Струков Артемий Викторович

ученик "Яндекс Лицей"

курс "Основы промышленного программирования | Д22"

от МАОУДО Центр "Диалог" Электросталь

преподаватель Симонов Сергей Иванович



PyQt5FileManager

Кроссплатформенный файловый менеджер
реализованный на Python с использованием PyQt5



Кроссплатформа

База данных sqlite

Python

Pyqt5

Избранное

Windows

Linux

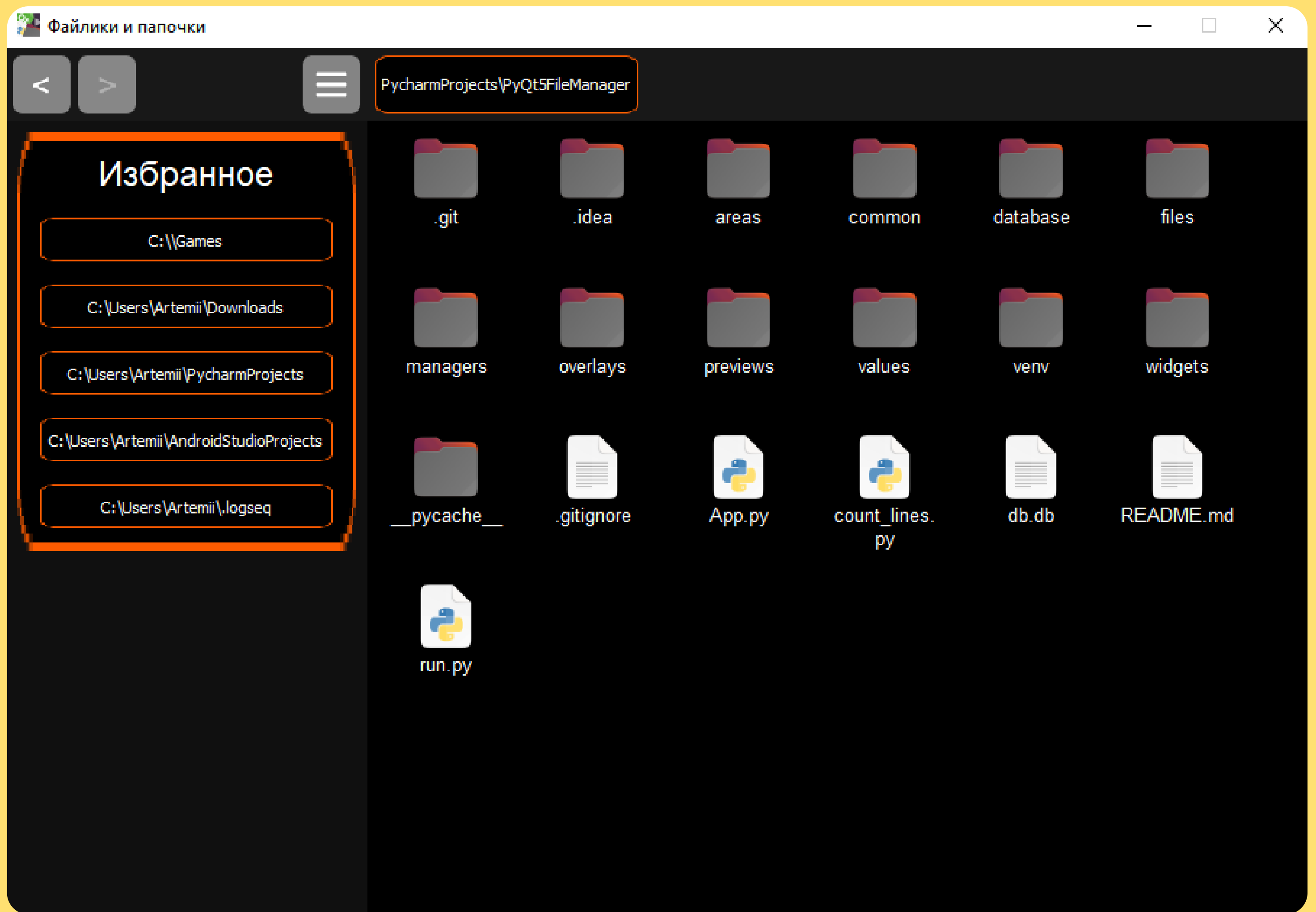
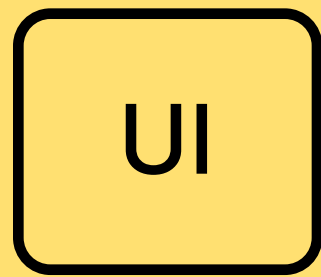
Система наложений

Предпросмотр файлов

Библиотека OS

История переходов

Система вкладок



Свои Widgets

```
1 favorites = QCardList(parent, WIDHT, "Избранное")
2 favorites.move(0, 0)
3 favorites.click_item = click_favorite
4 favorites.add_item("Тест item")
```

Избранное

C:\\Games

C:\\Users\\Artemii\\Downloads

C:\\Users\\Artemii\\PycharmProjects

C:\\Users\\Artemii\\AndroidStudioProjects

C:\\Users\\Artemii\\.logseq

```
1 class QCardList(QImageBackground):
2     def __init__(self, parent, w: int, title: str, items: list = None):
3         super().__init__(parent, "files/back_tag_select")
4         self._items = items if items is not None else []
5         self._labels = []
6         self._children = []
7         self.resize(w, self.height())
8         self.title_txt = title
9         self.empty_label = None
10        self.title = None
11        self.click_item = None
12        self.start_list_labels = 0
13        self.init_title()
14        self.refresh()
15
16    def init_title(self):
17        self.title = QLabel(self.title_txt)
18        self.title.setParent(self)
19        self.title.setFont(QFont("Arial", 18))
20        UtilsVisual.set_color_text(self.title, COLOR_TEXT)
21        self.title.show()
22        self.title.resize(self.width(), self.title.height())
23        self.title.setAlignment(Qt.Qt.AlignCenter)
24        self.title.move(self.title.x(), 2 * MARGIN_ITEM)
25        self._children.append(self.title)
26        self.title.show()
27        self.start_list_labels = self.title.y() + self.title.height()
28
29    def init_empty_label(self):
30        self.empty_label = QLabel("Пусто")
31        self.empty_label.setParent(self)
32        UtilsVisual.set_color_text(self.empty_label, COLOR_TEXT)
33        self.empty_label.resize(self.width(), self.height())
34        self.empty_label.setAlignment(Qt.Qt.AlignCenter)
35        self.empty_label.move(self.title.x(), self.start_list_labels)
36        self.empty_label.show()
37        self._children.append(self.empty_label)
38
39    def add_item(self, item: str):
40        if self.empty_label is not None:
41            self.hide_empty_label()
42        self._items.append(item)
43        self._add_new_label(item)
44
45    def refresh(self):
46        if len(self._items) == 0:
47            self.init_empty_label()
48        else:
49            self.re_move_labels()
50            self.hide_empty_label()
51        self._update_size()
```

Система наложений

Открыть
Открыть в новой вкладке
Переименовать
Добавить в избранное
Удалить
Свойства

Переименовать

overlays

Отмена

OK

Вы действительно хотите
удалить папку "managers"?

Cancel

OK

Создать файл

Создать папку

Удалить из избранного

```
1 class QWidgetOverlayManager(QWidget):
2     def __init__(self, mode: ModeOverlayManager = ModeOverlayManager.SINGLE):
3         super().__init__()
4         self.mode = mode
5         self.background_overlay = QLabel(self)
6         self.background_overlay.resize(WIDTH, HEIGHT)
7         self.background_overlay.mousePressEvent = lambda x: self._click_background()
8         UtilsVisual.set_background_color_label(self.background_overlay, (0, 0, 0), WIDTH,
9         HEIGHT, 190)
10        self.manager = OverlayManager()
11        self.active_overlays = []
12
13    def dismiss_all(self):
14        for parent in self.manager.get_parents():
15            overlays = self.manager.get_sub_overlays_parent(parent)
16            self.manager.remove_parent_overlay(parent)
17            self.active_overlays = []
18            for i in [parent] + overlays:
19                i.hide()
20                i.deleteLater()
21            self.manager.remove_all()
22            self.background_overlay.hide()
23
24    def dismiss_parent(self, parent):
25        overlays = self.manager.get_sub_overlays_parent(parent)
26        self.manager.remove_parent_overlay(parent)
27        if parent in self.active_overlays:
28            self.active_overlays.remove(parent)
29        for i in [parent] + overlays:
30            i.hide()
31            i.deleteLater()
32        if len(self.active_overlays) == 0:
33            self.background_overlay.hide()
34            self.background_overlay.lower()
35
36    def add_new_overlay(self, overlay: QOverlay, level=1):
37        if self.mode == ModeOverlayManager.SINGLE:
38            self.dismiss_all()
39        self.manager.add_parent_overlay(overlay, level)
40        self.refresh()
41
42    def add_sub_overlay(self, overlay: QOverlay, parent: QOverlay):
43        self.manager.add_sub_overlay(overlay, parent)
44        self.refresh()
45
46    def refresh(self):
47        for level in self.manager.to_list():
48            for group in level:
49                for overlay in group:
50                    if isinstance(overlay, QOverlay):
51                        overlay.raise_()
52
53    def show_overlay(self, overlay: QOverlay):
54        self._show_background()
55        self.active_overlays.append(overlay)
56        overlay.show()
57
58    def show_error(self, message):
59        error_overlay = QActionAlertDialog(message, self)
60        error_overlay.set_positive("OK", lambda: self.dismiss_parent(error_overlay))
61        self.add_new_overlay(error_overlay, self.manager.max() + 1)
62        self.show_overlay(error_overlay)
63
64    def is_active(self):
65        return len(self.active_overlays) != 0
```


Система предпросмотра

```
import os

#
# Скрипт для быстрого
# подсчёта количества строк в .py
# файлах проекта (без учета строк
# самого скрипта)
#

lst = []
file = str(__file__).replace("/", "\\")
path = "\\".join(file.split("\\")[0:-1])
for i in os.listdir(os.path.realpath(path)):
    if i[0] == ".":
        continue
    path_j = path + "\\ " + i
    if os.path.isfile(path_j):
        lst.append(path_j)
    else:
        for j in os.listdir(path_j):
            path_j = path_j + "\\ " + j
            if os.path.isfile(path_j):
```



```
1 class PreviewsManager(QAlphaDarkBackground):
2     def __init__(self):
3         super().__init__()
4         self.active_preview = None
5
6     def show_preview(self, preview: QPathObjectPreview):
7         if self.active_preview is not None:
8             self.dismiss_active_preview()
9
10        self.active_preview = preview
11        self.show_background()
12        preview.show()
13
14    def dismiss_active_preview(self):
15        if self.active_preview is not None:
16            self.active_preview.deleteLater()
17            self.active_preview = None
18
19    def press_background(self, *args):
20        super().press_background(*args)
21        self.dismiss_active_preview()
22
23    def is_active(self):
24        return self.active_preview is not None
```

База данных sqlite

```
1 def push_query_db_decorator(fun):
2     def push_query_db(*args):
3         with connect(DATABASE_NAME) as conn:
4             cursor = conn.cursor()
5             return fun(*args, cursor=cursor, conn=conn)
6     return push_query_db
```

```
1 class FavoriteRequests:
2     @push_query_db_decorator
3     def get_favorites(self, cursor: Cursor, conn: Connection):
4         result = cursor.execute("select * from favorites").fetchall()
5         return [i[1] for i in result]
6
7     @push_query_db_decorator
8     def add_favorite(self, path: str, cursor: Cursor, conn: Connection):
9         cursor.execute(f"insert into favorites(path) values('{path}')" )
10        conn.commit()
11
12    @push_query_db_decorator
13    def remove_favorite(self, path: str, cursor: Cursor, conn: Connection):
14        cursor.execute(f"delete from favorites where path='{path}'")
15        conn.commit()
```

PyQt5FileManager

Использовалась система контроля версий Git

Суммарно написано 2860 строчек

Разработка длилась 1 месяц

Реализовано 58 классов



PyQt5FileManager

учебный проект



Струков Артемий Викторович

ученик "Яндекс Лицей"

курс "Основы промышленного программирования | Д22"

от МАОУДО Центр "Диалог" Электросталь

преподаватель Симонов Сергей Иванович

