

Комета — это модуль в LuminaX, который может выполнять любые функции (эффекты на камеру, визуализации, звук и т.д.) и подключается через GUI. Каждая комета имеет стандартный интерфейс и методы для управления.

1. Структура папки кометы

Каждая комета — это отдельная папка в `comets/`, например:

```
comets/
├── MyComet/
│   ├── main.py           # Основной файл кометы
│   └── configs/
│       ├── a.py
│       ├── b.py
│       └── ...
```

- `main.py` — обязательно. LuminaX загружает его через `create_page()`.
 - `configs/` — папка с конфигами или скриптами, которые комета может запускать.
-

1. Основная структура файла `main.py`

Импорт библиотек:

```
from PyQt6.QtWidgets import QWidget, QVBoxLayout, QLabel, QPushButton,
QComboBox, QLineEdit
from PyQt6.QtGui import QFont, QImage, QPixmap
from PyQt6.QtCore import Qt, QTimer
import subprocess
import os
import cv2
```

Класс кометы:

```
class MyComet:
    def __init__(self):
        self.proc = None
        self.vcam_name = "MyCam"
        self.active_config = None
```

```
self.vcam_active = False
self.video_nr = 2 # номер устройства виртуальной камеры
```

Методы: - `start_virtual_camera()` и `stop_virtual_camera()` — запуск и остановка v4l2loopback. - `launch_config()` и `stop_config()` — запуск и остановка выбранного конфигурационного скрипта.

Пreview окна (опционально, если используется камера):

```
class PreviewWindow(QWidget):
    def __init__(self, video_nr=2):
        ... # открытие камеры и QTimer для обновления кадра
```

Функция интерфейса `create_page()`:

```
def create_page():
    page = QWidget()
    layout = QVBoxLayout()
    comet = MyComet()
    page.preview_window = None

    # Заголовок
    title = QLabel("🌟 Комета: MyComet")
    title.setFont(QFont("Arial", 16))
    layout.addWidget(title)

    # Виртуальная камера
    vcam_input = QLineEdit()
    vcam_input.setText("MyCam")
    layout.addWidget(vcam_input)
    vcam_btn = QPushButton("Вкл/Выкл камеры")
    layout.addWidget(vcam_btn)
    status_vcam = QLabel("VCam статус: Выключена")
    layout.addWidget(status_vcam)

    vcam_btn.clicked.connect(lambda: toggle_vcam())

    # Конфиги
    configs_dir = os.path.join(os.path.dirname(__file__), "configs")
    config_combo = QComboBox()
    config_combo.addItems([f for f in os.listdir(configs_dir) if
f.endswith(".py")])
    layout.addWidget(config_combo)
    toggle_btn = QPushButton("Включить комету")
    layout.addWidget(toggle_btn)
```

```

status_label = QLabel("Статус: Выключена")
layout.addWidget(status_label)

toggle_btn.clicked.connect(lambda: toggle_comet())

# Превью кнопка
preview_btn = QPushButton("Превью камеры")
layout.addWidget(preview_btn)
preview_btn.clicked.connect(lambda: toggle_preview())

page.setLayout(layout)
return page

```

1. Основные правила для новой кометы
2. Класс кометы с методами: `start_virtual_camera()`, `stop_virtual_camera()`, `launch_config()`, `stop_config()`.
3. Функция `create_page()` для GUI.
4. Конфиги помещаются в папку `configs/`.
5. Превью можно добавить через `PreviewWindow`.
6. На Linux работает виртуальная камера, на Windows эти функции игнорируются.
7. Рекомендуется использовать `QGroupBox` и `QVBoxLayout` для красивого интерфейса.

1. Пример конфигурации `configs/a.py`:

```

import cv2, numpy as np, subprocess
WIDTH, HEIGHT, FPS = 320, 240, 10
ffmpeg = subprocess.Popen(["ffmpeg", "-y", "-f", "rawvideo", "-vcodec",
"rawvideo", "-pix_fmt", "bgr24", "-s", f"{WIDTH}x{HEIGHT}", "-r",
str(FPS), "-i", "-", "-f", "v4l2", "-pix_fmt", "yuv420p", "/dev/
video2"], stdin=subprocess.PIPE)
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret: break
    frame = cv2.resize(frame, (WIDTH, HEIGHT))
    frame = cv2.GaussianBlur(frame, (3,3), 0)
    noise = np.random.randint(0,50,frame.shape,dtype=np.uint8)
    frame = cv2.add(frame, noise)
    _, enc = cv2.imencode('.jpg', frame, [int(cv2.IMWRITE_JPEG_QUALITY),
25])
    frame = cv2.imdecode(enc, 1)
    ffmpeg.stdin.write(frame.tobytes())

```

1. Шаблон быстрой кометы

```
class MyComet:
    def __init__(self):
        self.proc = None
        self.vcam_name = "MyCam"
        self.active_config = None
        self.vcam_active = False
        self.video_nr = 2
    def start_virtual_camera(self): ...
    def stop_virtual_camera(self): ...
    def launch_config(self): ...
    def stop_config(self): ...

def create_page():
    # GUI: заголовок, камера, конфиги, превью
    return page
```

С этим шаблоном новая комета интегрируется в LuminaX и сразу поддерживает: включение/выключение конфигураций, управление виртуальной камерой и превью камеры.