

# Генератор раскадровки Алдара Көсө: Интеллектуальная система повествования

# Содержание

## Работоспособность проекта

Part 1

Промты и ИИ-рисунки

04

Структура кода

05

## Разбиение кода

Part 2

Теория работоспособности проекта

07

Главные файлы и система работы

09

## Будущее проекта

Part 3

Сильная и слабая сторона

13

Возможности и ограничения

16

## Вопросы

Part 4

# Работоспособность проекта

Полностью автоматизированный pipeline:

Сценарий → Генерация сцен LLM → JSON → Изображения кадров → Storyboard.

Консистентность персонажей:

Фиксированные промпты и seed обеспечивают одинаковый внешний вид персонажей на всех кадрах.

Fallback-сценарии:

Предопределённый список кадров используется при ошибках LLM, чтобы генерация всегда успешна.

Гибкость и расширяемость:

Лёгкое добавление новых сцен, персонажей и стилей, поддержка разных разрешений кадров.

Совместимость с ресурсами:

Работает на GPU для ускоренной генерации; CPU возможен для тестов и небольших сцен.

Результат:

Reviews / Mobile Strategy

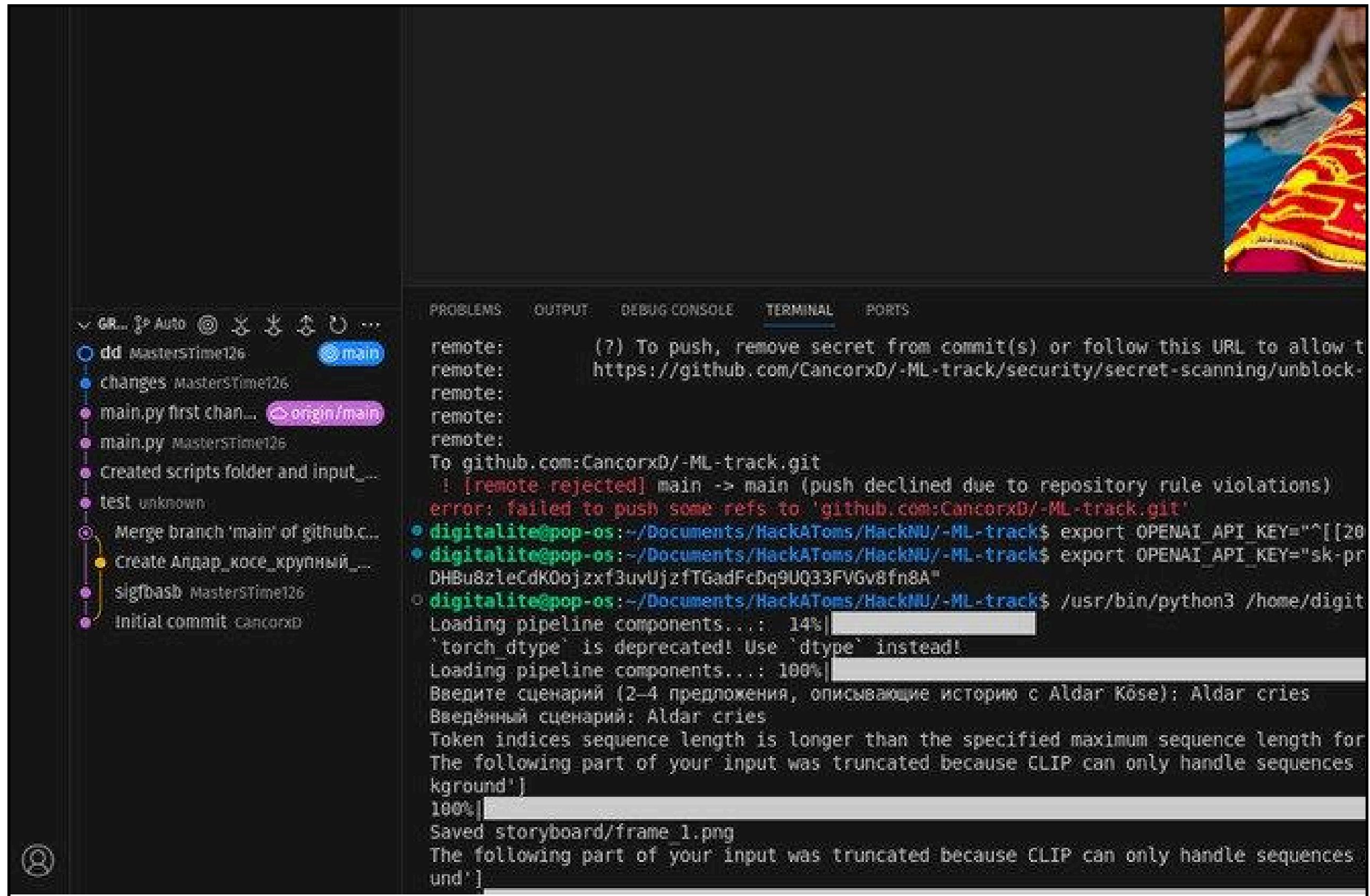
Product review — Month XX

Goals

Готовый storyboard с JSON и изображениями для дальнейшей работы или презентации.

# Промты и ИИ-рисунки

## ИИ инструменты вместо ручной работы



The screenshot shows a terminal window with a GitHub commit history on the left and a command-line interface on the right. The commit history includes:

- Initial commit: concorxd
- Create Aldar\_kose\_хрупкий\_
- sigfbasb: MasterTimeline126
- Initial commit: concorxd
- Merge branch 'main' of github.com:ConcorxD/-ML-track into main
- main.py first chan...
- main.py: MasterTimeline126
- Created scripts folder and input...
- test: unknown
- Changes: MasterTimeline126
- dd: MasterTimeline126

The command-line interface shows the following steps:

- remote: (?) To push, remove secret from commit(s) or follow this URL to allow t  
remote: https://github.com/ConcorxD/-ML-track/security/secret-scanning/unblock-
- To github.com:ConcorxD/-ML-track.git  
! [remote rejected] main -> main (push declined due to repository rule violations)
- error: failed to push some refs to 'github.com:ConcorxD/-ML-track.git'
- digitalite@pop-os:~/Documents/HackAToms/HackMU/-ML-track\$ export OPENAI\_API\_KEY="^[[20
- digitalite@pop-os:~/Documents/HackAToms/HackMU/-ML-track\$ export OPENAI\_API\_KEY="sk-pr
- DHBu8zleCdK0ejzxf3uvUjzfTGadFcDq9UQ33FVGv8fn8A"
- digitalite@pop-os:~/Documents/HackAToms/HackMU/-ML-track\$ /usr/bin/python3 /home/digit

The interface then prompts for input:

Введите сценарий (2-4 предложения, описывающие историю с Aldar Köse): Aldar cries

Введённый сценарий: Aldar cries

Token indices sequence length is longer than the specified maximum sequence length for

The following part of your input was truncated because CLIP can only handle sequences

kground"]

100%

Saved storyboard/frame 1.png

The following part of your input was truncated because CLIP can only handle sequences

und"]

### Шаг 1

Создание основного промпта - по детальному описанию 2D нейтрального Алдара Косе

### Шаг 2

В других промптах меняли только описание позы, выражение лица и одежду, не изменяя структуры

### Шаг 3

Soft engineering, используя длинные описания

# Структура кода

Меняемые скрипты

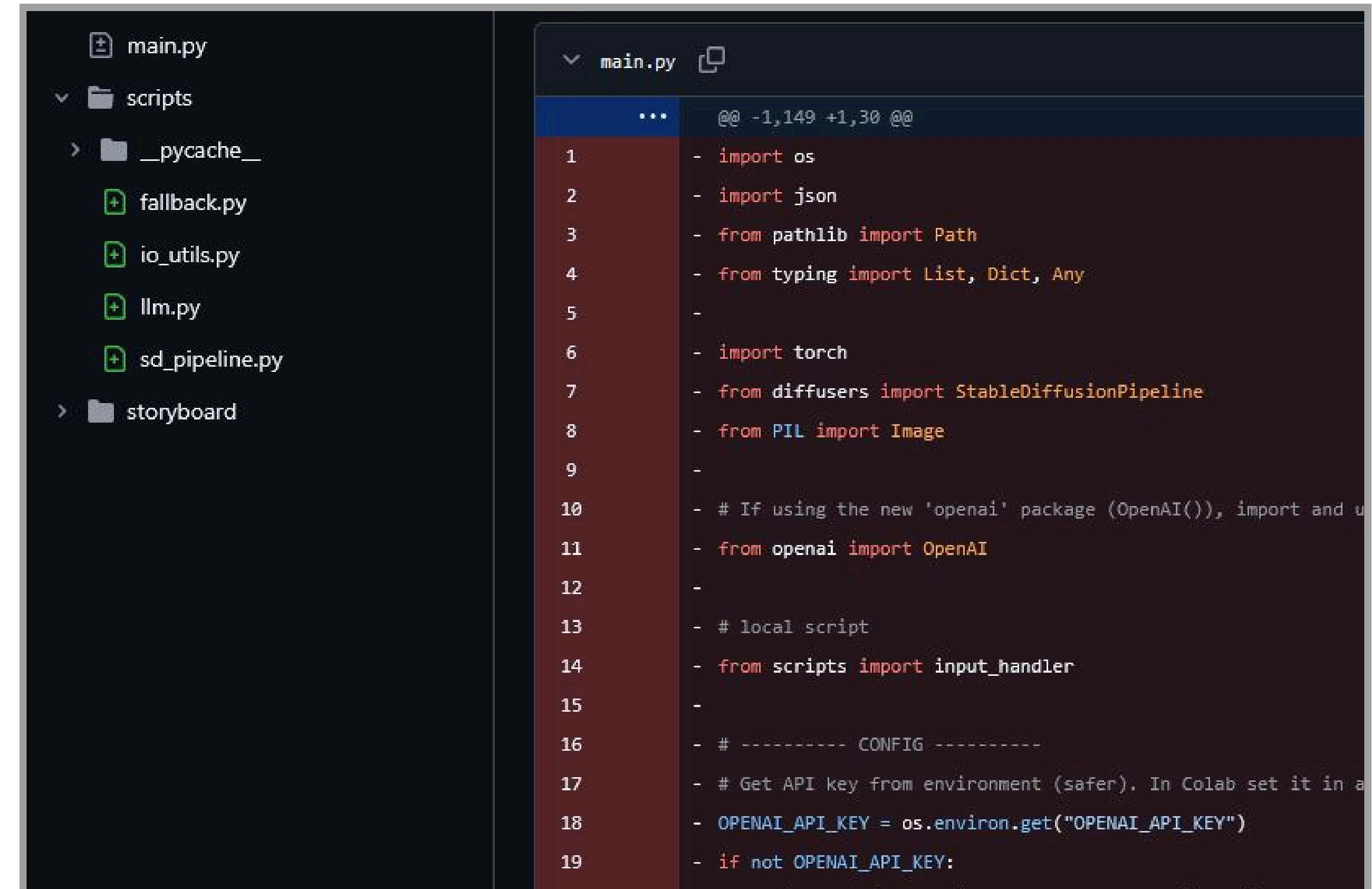
Нахождение багов облегчается

Возможность работы

Даже при экстренных ситуациях

Сохранение выходных данных

В папке storyboard



The image shows a code editor with a dark theme. On the left, a file tree displays the project structure:

- main.py
- scripts (expanded):
  - \_\_pycache\_\_ (empty)
  - fallback.py
  - io\_utils.py
  - llm.py
  - sd\_pipeline.py
- storyboard (empty)

On the right, the content of main.py is shown:

```
@@ -1,149 +1,30 @@
1 import os
2 import json
3 from pathlib import Path
4 from typing import List, Dict, Any
5
6 import torch
7 from diffusers import StableDiffusionPipeline
8 from PIL import Image
9
10 # If using the new 'openai' package (OpenAI()), import and use
11 # from openai import OpenAI
12
13 # local script
14 from scripts import input_handler
15
16 # ----- CONFIG -----
17 # Get API key from environment (safer). In Colab set it in a
18 OPENAI_API_KEY = os.environ.get("OPENAI_API_KEY")
19 if not OPENAI_API_KEY:
```

# Разбирание кода

# Теория проекта

Пользователь вводит 2–4 предложения сценария. Модуль LLM формирует структурированный список сцен (frame, description), который проверяется и при необходимости заменяется на запасной набор. Затем генератор изображений (Stable Diffusion) по каждому описанию создаёт кадр с фиксированным семенем для воспроизводимости; все данные и изображения сохраняются локально. Архитектура модульная (LLM, SD-пайpline, I/O), поддерживает защиту секретов через переменные окружения и откат на резервный список при ошибках.

# main.py

```
1  """
2      Entry point. Orchestrates prompt input, LLM shot-list generation and image creation.
3  """
4
5  from scripts.input_handler import get_prompt
6  from scripts.llm import generate_shot_list
7  from scripts.io_utils import save_info
8  from scripts.fallback import FALLBACK_SHOT_LIST
9
10
11
12  def main() -> None:
13      script = get_prompt("Введите сценарий (2-4 предложения, описывающие историю с Aldar Köse): ")
14      print("Введённый сценарий:", script)
15
16
17      try:
18          shot_list = generate_shot_list(script)
19      except Exception as e:
20          print("Falling back to predefined shots due to error:", e)
21          shot_list = FALLBACK_SHOT_LIST
22
23
24      result_msg = save_info(shot_list)
25      print(result_msg)
26
27
28  if __name__ == "__main__":
29      main()
```

# Необходимые компоненты проекта

io\_utils.py

```
1  import json
2  from pathlib import Path
3  from typing import List, Dict, Any
4  from sd_pipeline import generate_frame
5
6
7  def save_info(shot_list: List[Dict[str, Any]]) -> str:
8      description_path = Path("storyboard") / "index.json"
9      description_path.parent.mkdir(parents=True, exist_ok=True)
10
11     with description_path.open("w", encoding="utf-8") as f:
12         json.dump(shot_list, f, ensure_ascii=False, indent=4)
13
14     for shot in shot_list:
15         frame_num = shot.get("frame")
16         desc = shot.get("description")
17         if frame_num is None or desc is None:
18             print("Skipping shot (missing 'frame' or 'description'): ", shot)
19             continue
20
21     prompt = (
```

llm.py

```
4  import os
5  import json
6  from typing import List, Dict, Any
7  from openai import OpenAI
8
9
10 OPENAI_API_KEY = os.environ.get("OPENAI_API_KEY")
11 if not OPENAI_API_KEY:
12     raise RuntimeError("Set OPENAI_API_KEY in the environment. Do NOT hardcode it in the code!")
13
14 _client = OpenAI(api_key=OPENAI_API_KEY)
15
16
17
18
19
20 def generate_shot_list(script: str) -> List[Dict[str, Any]]:
21     prompt = f"""
22     Given the following script: "{script}"
23     Generate a list of 6-10 detailed scene descriptions for a storyboard. Each scene description should include:
24     - A short description of the action.
25     - Visual details (Aldar Köse's appearance, setting, camera angle).
26     - Ensure Aldar Köse is recognizable (e.g., wears traditional Kazakh clothing).
27     Return the result as a JSON list with fields 'frame' and 'description'.
28     """
29
30
```

sd\_pipeline.py

```
5  import torch
6  from diffusers import StableDiffusionPipeline
7
8
9  _PIPELINE: Optional[StableDiffusionPipeline] = None
10
11
12
13
14 def _device() -> str:
15     return "cuda" if torch.cuda.is_available() else "cpu"
16
17
18
19
20 def load_pipeline(model_id: str = "runwayml/stable-diffusion-v1-5"):
21     global _PIPELINE
22     if _PIPELINE is not None:
23         return _PIPELINE
24
25
26     device = _device()
27     torch_dtype = torch.float16 if device == "cuda" else torch.float32
```

# Будущее проекта



Reviews / Mobile Strategy

Product review — Month XX

Design Concepts

# Strengths

- \* Быстрое прототипирование раскадровки.
- \* Модульная структура кода, легко расширять.
- \* Уникальный культурный контент (Aldar Köse).

# Weaknesses

- \* Требует мощного GPU для генерации изображений.
- \* Зависимость от внешних моделей и API.
- \* Качество изображений зависит от точности промптов.



# Opportunities

- \* Расширение на несколько персонажей и анимацию.
- \* Интеграция с веб-интерфейсом, игровыми движками или образовательными платформами.
- \* Потенциальная коммерциализация: SaaS, образовательные лицензии, маркетплейсы.

# Limitations

- \* Лицензирование моделей и изображений.
- \* Ограничения по ресурсам (время генерации, стоимость GPU).
- \* Возможные ошибки LLM (невалидный JSON, неконсистентные кадры).



# Есть вопросы?

