

ЗВІТ

Лабораторна робота №2 Інформаційні системи (Python, Desktop Applications)

Тема роботи:

Розробка десктопного медіаплеєра з використанням Python та бібліотеки *PySide6*. Плеєр повинен підтримувати локальні медіафайли, потокове відео/аудіо, елементи керування (Play, Pause, перемотка), відображення часу та повзунок програвання.

1. Мета роботи

Ознайомитись з принципами створення десктопних застосунків мовою Python, навчитися працювати з мультимедійними компонентами, сигналами/слотами, подіями та інтерфейсом користувача.

2. Використані технології

Технологія	Призначення
Python 3.10	основна мова програмування
PySide6 (Qt for Python)	побудова GUI-інтерфейсу
QMediaPlayer	відтворення аудіо та відео
QVideoWidget	рендеринг відео у вікно
QAudioOutput	звуковий потік
Qt Signals/Slots	обробка подій (клік, перемотка, зміна позиції)

3. Опис функціоналу застосунку

Розроблений медіаплеєр має такі можливості:

3.1. Відтворення локальних медіафайлів через **QFileDialog** користувач може вибрати файл: .mp4; .avi; .mkv; .mp3; .wav

Файл відкривається через: `self.player.setSource(QUrl.fromLocalFile(file_name))`

3.2. Відтворення потокового медіа (online URL)

У нижній частині вікна — текстове поле. Після введення URL і натискання Enter відео починає відтворюватись.

3.3. Базові елементи керування

Кнопка	Дія
Play / Pause	запускає або ставить на паузу відео
-5 с	перемотує назад на 5 секунд
+5 с	перемотує вперед на 5 секунд
Restart	зупиняє програвання і повертає відео на 0:00

3.4. Повзунок програвання

Слайдер повністю синхронізований із позицією відео:

- рухаєш повзунок → змінюється позиція
- змінюється позиція відео → рухається повзунок

3.5. Відображення часу

Під повзунком показується: поточний час / загальна тривалість
Формат часу: MM:SS.

3.6. Відображення відео

Використано `QVideoWidget`, який рендерить відео всередині вікна.

4. Опис реалізації

4.1. Ініціалізація медіаплеєра

```
self.player = QMediaPlayer(self)
self.audio_output = QAudioOutput(self)
self.player.setAudioOutput(self.audio_output)
```

Створюється медіаплеєр та аудіовихід.

4.2. Інтерфейс (GUI)

Інтерфейс побудовано через `QVBoxLayout` та `QHBoxLayout`.

Основні елементи:

- відеополе
- блок кнопок
- повзунок
- мітка часу
- поле URL

4.3. Відкриття файлу

```
file_name, _ = QFileDialog.getOpenFileName(...)
self.player.setSource(QUrl.fromLocalFile(file_name))
self.player.play()
```

4.4. Потокове відтворення

```
url = QUrl(url_text)
self.player.setSource(url)
self.player.play()
```

4.5. Перемотка

```
self.player.setPosition(self.player.position() - 5000) # назад
self.player.setPosition(self.player.position() + 5000) # вперед
```

4.6. Кнопка «На початок»

```
self.player.stop()
self.player.setPosition(0)
```

4.7. Оновлення часу і повзунка

Позиція змінилася → оновити слайдер:

```
self.player.positionChanged.connect(self.on_position_changed)
```

Тривалість змінилася → оновити межі:

```
self.player.durationChanged.connect(self.on_duration_changed)
```

5. Як працює застосунок

1. Користувач запускає програму.
2. Вибирає медіафайл або вставляє URL.
3. Відео з'являється у відео-панелі.
4. Повзунок рухається синхронно з відео.
5. Доступне керування відтворенням:

- play/pause

- перемотка
 - перехід на початок
6. Час оновлюється в реальному часі.

Медіаплеєр працює повністю автономно і є ручною реалізацією на Python.

6. Фрагмент коду головного запуску

```
if __name__ == "__main__":  
    app = QApplication(sys.argv)  
    w = MediaPlayer()  
    w.show()  
    sys.exit(app.exec())
```

7. Інструкція запуску

7.1. Установка залежностей (один раз)

```
pip install PySide6
```

7.2. Запуск програми

Перейти в папку з проектом: `cd A:\унік\4_курс\pmc\rms_lab2`

Активувати віртуальне середовище: `.venv\Scripts\activate`

Запустити застосунок: `python rms_lab2_videoplayer.py`

8. Висновки

У ході роботи було створено повністю функціональний медіаплеєр на Python, що підтримує:

- аудіо та відео формати
- відтворення онлайн-потоків
- повзунок програвання
- перемотку
- показ часу
- графічний інтерфейс з кнопками

Було застосовано механізм сигналів/слотів Qt, розроблено інтерфейс вручну, а також реалізовано логіку відтворення, паузи, перемотування та керування позицією. Програма працює стабільно та виконує всі вимоги завдання.