

김응혁

이메일: coke1324@naver.com

연락처: +821051322182

하드웨어 직무에서 소프트웨어 직무로 전환하여 제조업에서 MFC Application 개발을 맡아
대용량 이미지 병렬 처리, PyQt5 전환, CCD 16Tap 조정 알고리즘 등 주요 기능을 개발하였으며
모던 C++ 및 스크립트 언어를 내포하여 프로그램을 리팩토링한 경험이 있습니다.
현재는 1인 프로젝트를 진행하여 C++ 개발 역량을 늘리고 있습니다.

경력

뷰웍스

광영상생산팀 / 사원

2019.1 - 2021.5

- MFC to PyQt5 전환 2021.1 - 2021.4
 - 수정이 잦은 일부 GUI에 대해 런타임 중에 수정 할 수 있도록 PyQt5 도입
 - Boost.Python 라이브러리를 도입하여 C++, Python 간 데이터 공유
 - 개발 인원 및 기여도: 2명, 50%
 - 주요 라이브러리: MFC, PyQt5, Boost.Python
- 대용량 이미지 병렬 처리 2020.10 - 2020.12
 - Histogram, 이미지 연산, Defect Cluster 알고리즘 병렬 처리
 - 14192 x 10640 해상도 이미지 처리 속도 1.5배 향상
 - 개발 인원 및 기여도: 2명, 50%
 - 주요 라이브러리: TBB, PPL, CUDA
- Pixel Inspector Tool 개발 2020.7 - 2020.9
 - Mono, Color 이미지에 대한 실시간 Block 단위 Pixel 최대, 최소, 평균, 편차 계산 구현
 - Block 단위 편차 filtering 기능 구현
 - 주요 라이브러리: MFC Grid control
- CCD 16Tap 조정 알고리즘 개선 2020.3 - 2020.4
 - 기존: 기준 Tap을 정해서 나머지 Tap 간의 mismatch를 반복문을 이용해서 계산
→ 전체 시간 복잡도: $O(N^2)$
 - 개선: 기준 Tap을 정해서 나머지 Tap 간의 mismatch를 선형 보간법을 이용해서 계산
→ 전체 시간 복잡도: $O(N)$
 - 사용 언어: Lua
- 카메라 조정 프로그램 리팩토링 2019.8 - 2019.12
 - 데이터 특징에 맞는 컨테이너로 수정하여 시간 복잡도 개선

- 수정이 잦은 로직에 대해 런타임 중에 수정 할 수 있도록 스크립트 로직으로 전환
- Luabind 라이브러리를 도입하여 C++, Lua 간 데이터 공유
- 단일 실행 파일에서 기능 별 라이브러리로 분리하여 프로그램 모듈화
- 스마트 포인터 도입으로 메모리 누수 해결
- > 소유권이 없는 Raw Pointer에서 unique_ptr, shared_ptr로 변경

- 개발 인원 및 기여도: 2명, 50%
- 주요 라이브러리: Luabind

● ini to JSON 전환 2019.6 - 2019.7

- 하나의 ini 파일로 관리하던 카메라 설정 파일을 180개의 JSON 파일로 분리
- JSON schema 도입으로 데이터 유효성 검증

- 주요 라이브러리: RapidJSON

학력	숭실대학교 전자공학과 - 주요 이수과목: 전자회로, 신호처리, 반도체소자, 센서공학, 디스플레이 원리, 디지털공학 등 - 졸업논문: 베타-산화 갈륨을 이용한 이중접합 애벌런치 포토 다이오드	2012.3 - 2019.2
----	---	-----------------

수상 및 기타	[삼성전자] 하계 대학생 S/W 알고리즘 역량 강화 특강 - 기간: 2021.07 ~ 2021.08 - 사용한 Skill 또는 지식: C++, STL, 자료구조, 알고리즘 - 핵심 내용: S/W 문제해결 역량 향상 - 주요 내용: 1. Linked List, Tree, Heap, Trie, Hash 등 2. 그리디, 완전 탐색, DP, 분할 정복, 그래프 탐색 등	2021.7
---------	---	--------

	1인 프로젝트 - 프로젝트: Windows API를 이용한 폴더 공유 시스템 - 기간: 2021.06 ~ 2021.08 - 사용한 Skill 또는 지식: C++, Python, S3, DBMS, Windows API, REST, WebSocket - 주요 라이브러리: cpprestsdk, aws-sdk-cpp, sqlite3, Flask, SQLAlchemy 등 - 핵심 내용: 클라이언트 간 특정 폴더를 실시간으로 공유 - 개발 내용: 1. Windows API를 이용한 폴더의 변경 사항 모니터링 기능 2. Flask 프레임워크를 이용한 RESTful API 제공 3. 폴더의 변경 사항 HTTP Request 기능 4. S3 Storage Server 파일 업/다운로드 기능 5. WebSocket을 활용한 Broadcast 기능 6. 로컬 데이터베이스 및 서버 데이터베이스 관리(SQLite, MySQL) 기능	2021.6
--	---	--------

7. GoogleTest, pytest를 이용한 unit test 구현

- GitHub:

1. 클라이언트: https://github.com/EUING/monitor_client

2. 서버: https://github.com/EUING/monitor_server

외국어

영어

일상회화

● TOEIC Speaking 2017.08.19

140점/Level 6

링크

<https://github.com/EUING>