

기술 이력서 - 김용욱

게임 클라이언트 / 서버 / 그래픽스

- Email: rladyddnrl@gmail.com
- Github: [2wind](#)
- Solved.ac: [@whirlwind_ko](#)

기술

- 주력 언어: C#, python
- [solved.ac](#) 티어 골드 1 (2022년 4월 4일 기준, 계정명 whirlwind_ko)
- Unity with C# 다수 게임 제작, Unity Built-in shader 및 Replacement shader 관련 과제 진행
- Modern OpenGL with C++ 이용, 렌더러 및 레이트레이서 제작 경험
- Python, Pytorch, OpenCV를 이용한 기계학습 및 데이터 처리, 프로그래밍화 경험 보유
- 개인 홈페이지: kimyongwook.duckdns.org / GitHub 계정: github.com/2wind

학력

- 서울대학교 - 학사
 - 농업생명과학대학 산림과학부 환경재료과학과
 - 공과대학 컴퓨터공학부 컴퓨터공학과 (복수전공)

2014년 3월 ~ 2022년 8월 졸업예정

(2022년 2월 기준, GPA 3.43/4.3)

자격

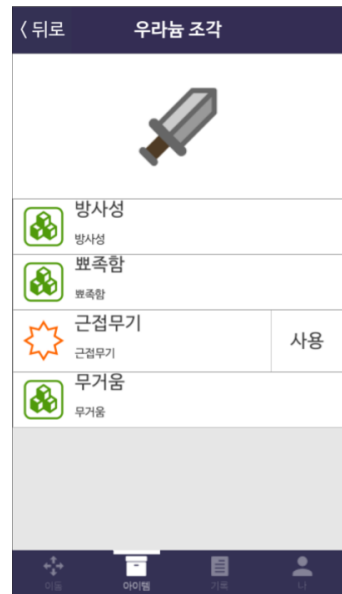
정보처리산업기사 - 2017년 5월 26일 취득

정보기기운용기능사 - 2019년 12월 2일 취득

군복무(공군, 군필) - 2018년 10월 15일 ~ 2020년 8월 11일

프로젝트

던전 엔지니어

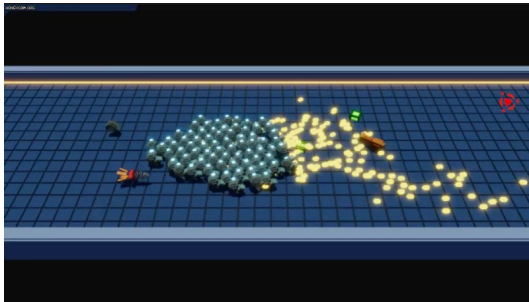


- 개요: 물건의 속성을 조합하는 모바일 로그라이크
- 개발 기간: 2015. 06 ~2015. 08
- 개발 환경: Unity with C#
- 타겟 플랫폼: 안드로이드/iOS 환경
- 프로젝트 팀원: 본인 포함 5명
- 담당 역할:
 - 게임에 등장하는 다양한 "아이템" 구현
 - 태그 시스템을 이용한 아이템의 합성 로직
 - UI 제작 (일부)

설명

- 2015년 여름 처음으로 유니티 엔진을 이용해서 본격적인 게임을 만들게 되었습니다. 던전 엔지니어는 고전적 로그라이크를 지향하는 턴제 모바일 게임으로, 맵 상에 등장하는 아이템들의 다양한 "태그"를 이용해 다양한 물건을 제작하면서 던전을 돌파하는 게임입니다.
- 수업시간에서 클래스의 상속의 개념에 대해 배우기는 했지만 실질적으로 어떤 곳에 이용할지는 몰랐었습니다. 그런데 다양한 아이템들에 대해 일관성 있는 인터페이스로 접근을 하기 위해 상속 패턴을 활용하게 되었습니다. 이를 통해 컴퓨터공학의 지식이 게임 개발에 어떤 식으로 활용 가능한지에 대해 알게 되었습니다.
- 모바일 게임에서 흔히 보이는 스크롤 가능한 리스트 UI를 구현할 때 uGUI를 이용했습니다. 처음에는 전체 높이를 '매직 넘버'로 잡아 두고 엘리먼트 높이를 수동으로 계산하는 식으로 레이아웃을 수동으로 작성했습니다. 이후 엔진에서 제공하는 레이아웃 컴포넌트를 사용하는 것으로 리팩터링을 하여 생산성을 높일 수 있었습니다.
- 이 때에는 아직 프로그래밍적 지식이 많아 많은 분야에 참여하지는 못 했지만, 게임 개발에 대해 배울 수 있었던 좋은 기회였다고 생각합니다.

클러스터노이드



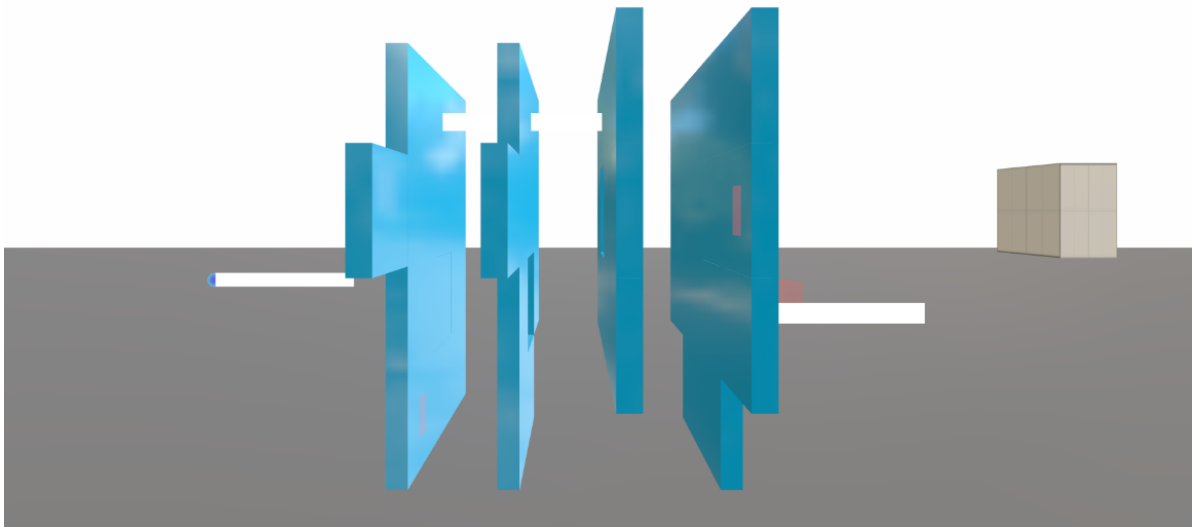
- 개요: 수많은 주인공 캐릭터를 조종하는 탑 뷰 슈터
- 개발 기간: 2017년 12월 ~ 2018년 2월
- 개발 환경: Unity with C#
- 타겟 플랫폼: PC
- 프로젝트 팀원: 본인 포함 10명
- 담당 역할
 - 플레이어 캐릭터(들)에 대한 조작(이동, 공격 등) 전반
 - Unity Animation FSM을 이용한 적 인공지능(FSM 방식)
 - UI 전반
 - 시연용 맵 구성 및 테스트
- 소스 코드: <https://github.com/2wind/Clusternoid>

설명

- 2017년 겨울에는 메인 프로그래머로 게임 개발에 참가하게 되었습니다. 클러스터노이드는 개개 캐릭터는 약하지만, 대신 수많은 캐릭터들을 한 번에 조종하면서 맵을 클리어하는 싱글플레이 탑 뷰 슈터입니다. 이 프로젝트에서는 핵심 부분에서 상세한 알고리즘을 구현하고, 부수적인 부분은 다른 라이브러리를 가져와 빠르게 퀄리티를 높일 수 있었습니다.
- 중심 알고리즘의 경우, 많은 수의 (수십 ~ 수백 개체) 플레이어 캐릭터들이, 탑 뷰 평면 맵 위에서 WASD 조작으로 원활하게 이동하는 알고리즘을 구현해야 했습니다. 이 부분에서는 현재 게임회사에서 근무중인 선배님의 도움을 많이 받았으며, 결과적으로 짝 프로그래밍과 같이 되었다고 생각합니다. 그 선배님은 거시적인 부분, 즉 flow field pathfinding을 이용해 전반적인 캐릭터들이 모여 드는 부분을 중심으로 개발해주셨습니다. 저는 미시적인 부분, 즉 캐릭터들 중 플레이어가 실제로 조작하는 캐릭터들이 어떻게 움직이고 또 이동해야 하는지를 담당하게 되었습니다.
- 어딘가에 정답이 존재하는 문제가 아니었기 때문에, 이런 저런 알고리즘을 생각하고, 실제로 프로토타이핑을 하면서 원활한 조작이 될 때까지 튜닝을 했습니다. 마치 점성이 있는 슬라임의 움직임과 같은 군집을 구현하기 위해 여러 요소들을 구현해야 했습니다. 각 캐릭터들이 다른 캐릭터들과 가져야 하는 인력과 척력이나 플레이어가 이동 버튼을 눌렀을 때 어떤 개체를 따라가게 만들어야 하는지 등을 구현하면서 새로운 알고리즘을 구현하고 튜닝하는 법에 대해 많이 배울 수 있었습니다.

- 전반적인 UI (메뉴가 존재하고, 어떤 버튼을 누르면 게임이 실행되고, esc를 누르면 특정 화면에서는 게임을 정지하지만 다른 메뉴에서는 "뒤로" 이동하는 등)도 만들어 보았습니다. 이 때에는 UI 플로우차트를 그려가면서 제작을 했었는데, 실제로 게임에 그 내용을 옮길 때 많은 도움이 되었습니다. 또 사운드나 폰트, 모델링이나 애니메이션은 다른 팀원들이 제작한 것을 가져와 사양서에 따라 적용을 시켰습니다. 사양서나 플로우차트가 제대로 있을 때 개발 효율이 훨씬 좋아지는 것을 체감하면서, 문서화의 중요성을 느낄 수 있었습니다.
- 모든 것을 우리가 다 만들 필요는 없다는 생각도 했습니다. 예컨대, 포스트 프로세싱 에셋을 통해, 추가적인 개발 부담 없이 예쁜 그래픽을 구현할 수 있었습니다. 또 카메라의 이동과 위치 저장 등에는 Cinemachine 에셋을 활용했습니다. Cinemachine 에셋에서는 특정 오브젝트를 부드럽게 따라가거나, 가상 카메라의 위치를 지정해 실제 카메라를 (가상 카메라를 따라가거나 특정 위치로) 이동시키는 기능을 지원합니다. 이를 통해 게임 시작 시와 클리어 시에 카메라를 적절하게 움직이고, 카메라가 플레이어 캐릭터들을 따라가는 기능을 빠르게 만들었습니다. 배경음악을 재생하거나, 임의의 오브젝트에서 소리를 재생하는 기능이 필요했는데. 이런 기능을 지원할 사운드 매니저 시스템은 (다른 동아리원이) 만든 것을 가져와 조금 수정하여 사용했습니다.
- 이렇듯이 이 프로젝트에서는 게임 개발 프로세스 전반을 체감하면서, 동시에 재미있는 게임을 만들 수 있던 기회였다고 생각합니다.

GridUnlocker



- 개요: 레이저를 맞추는 간단한 VR 퍼즐 게임
- 개발 기간: 2021년 1월 ~ 8월
- 개발 환경: Unity with C#
- 타겟 플랫폼: Oculus / SteamVR
- 프로젝트 팀원: 본인 1인
- 담당 역할: 전체
- 소스 코드: <https://github.com/2wind/GridUnlocker>

설명

- 군 전역 후 2021년, VR 플랫폼이라는 새로운 플랫폼에서 정적이면서 편안하게 즐길 수 있는 퍼즐 게임을 만들고 싶었습니다. GridUnlocker는 레이저를 적절하게 한쪽 끝에서 다른 쪽에 있는 목표까지 인도하는 게임입니다. 한 쪽에서 쏘아진 레이저는 중간에 여러 판들에 의해 가로막히는데, 특정한 방식으로 판을 배열해서 반대쪽에 놓인 목표까지 레이저를 쏘아 보내야 합니다. 이 게임은 제가 처음 만들어보는 VR 게임이자 혼자 만들어보는 게임으로써, 이동과 팔 휘두르기 없이 정적인 분위기에서 퍼즐을 풀 수 있도록 기획해 보았습니다.
- VR 게임을 처음부터 기획하고 만들어보면서, 기획 / 프로그래밍 / 아트 분야의 작업을 직접 체험할 수 있었습니다. VR이라는 생소한 플랫폼과 조작 방식을 익힐 수 있었고, VR 공간에서 적절한 UI와 조작 방식을 시험해볼 수 있었습니다.
- 또한 Unity3D에서 제공하는 오브젝트 조작 코드만으로는 만족스럽게 게임을 구현할 수 없어서 해당 부분을 직접 만들어 보았습니다. 각 퍼즐 조각이 중심 축을 기준으로 회전만 하거나, 일정 범위 이내에서 이동만 하게 제한되어야 했습니다. 이 부분을 구현하기 위해 물체의 회전 각도와 컨트롤러의 위치 변화 등을 이용해 제한된 움직임을 구현했습니다. 또 이 과정에서 쿼터니온, 로컬 좌표계와 월드 좌표계 등의 개념을 이해할 수 있었고, 이후 컴퓨터그래픽스 수업 시간 등에서도 활용할 수 있었습니다.
- 게임의 작동 로직에는 UnityEvent를 적극적으로 활용했습니다. 컨트롤러가 퍼즐 조각을 "잡고 있을 때"의 동작을 정의하거나 승리 조건을 검사할 때, 적절한 함수를 제작해 등록해 두면 결합도를 낮추면서 이후 원하는 동작을 하게 만들 수 있습니다. UnityEvent에 대해 본격적으로 공부할 수 있었고, 나아가 C#의 Event / Delegate에 대해서도 공부하게 되었습니다.

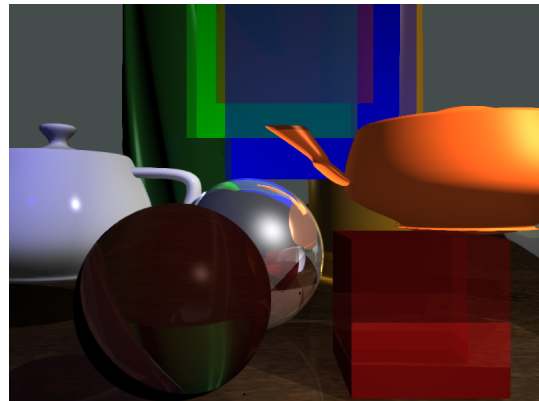
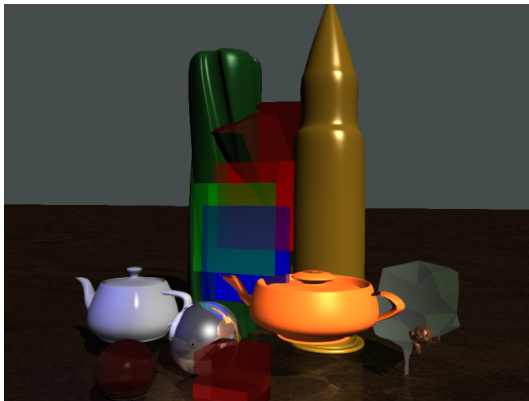
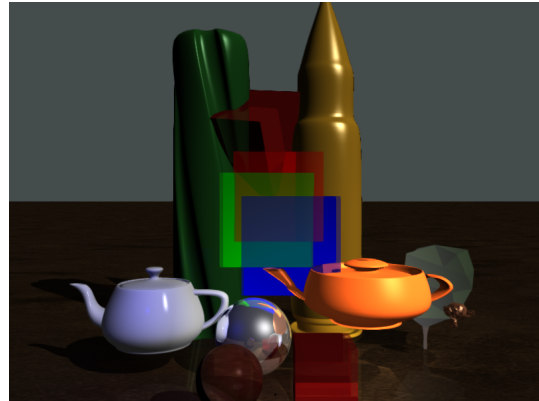
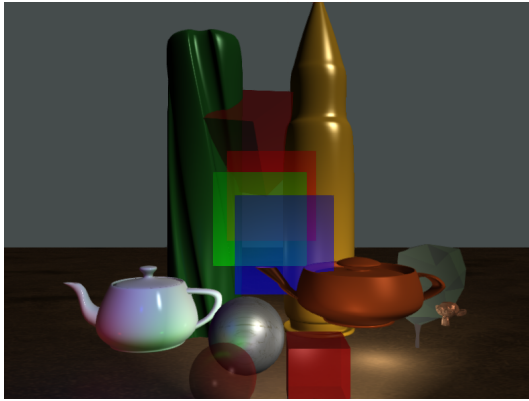
LandmarkDetector

- 개요: 이미지간 랜드마크 자동 정합 프로그램
- 개발 기간: 2021년 3월 ~ 7월 초
- 개발 환경: Python, PyTorch, PyInsaller
- 프로젝트 팀원: 본인 포함 2인
- 담당 역할: 전체

설명

- LandmarkDetector는 창의적통합설계 과목 중, (주)오스텝으로부터 의뢰받은 “AI 기반 치과 랜드마크 자동 정합 프로그램 개발”을 구현한 결과입니다.
- 이미지 2개와, 한 이미지 내에서 정해진 랜드마크를 받으면 다른 이미지에서 상응하는 랜드마크를 찾아 정합을 하는 모델을 개발했습니다. 이를 위해 다음과 같은 요소를 구현했습니다.
 1. CNN을 바탕으로 하여, 이미지에서 정해진 랜드마크를 찾을 수 있는 모델 탐색.
 2. 오스텝에서 제공한 이미지 및 랜드마크 쌍을 데이터셋 형태로 가공 및 augmentation.
 3. 모델 학습 및 다른 프로그램(facenet-pytorch)과의 연동.
 4. 독립 실행형 프로그램 형태로의 연동(pyinstaller)
- 위 4가지 요소를 모두 혼자서 개발하였으며, 딥 러닝의 장점과 한계, 작동 방식에 대해서 배울 수 있는 유익한 시간이었습니다.

컴퓨터그래픽스 HW 5



- 개요: GPU 가속 레이트레이싱 렌더링
- 개발 기간: 2021년 11월 ~ 12월 초
- 개발 환경: GLFW, GLAD, C++ (OpenGL 4.6 대응)
- 프로젝트 팀원: 본인 1인
- 담당 역할: 전체

설명

- 2021년 2학기에는 컴퓨터그래픽스 수업을 수강하였습니다. 이 수업에서는 컴퓨터에서 렌더링이 이뤄지는 방법을 이론적으로 배우고, 과제로는 OpenGL을 이용해 실제로 렌더링을 수행하였습니다. OpenGL의 버전에 대해서는 특별히 제한을 두지 않았지만, 저는 셰이더와 같은 현대적인 렌더링 프로세스에 대해 공부하고 싶었기 때문에 일부러 OpenGL 3.6 Core 버전을 기준으로 C++과 GLSL을 이용해 과제를 수행했습니다.
- 마지막 과제에서는 다양한 오브젝트가 배치된 씬을 구성하고, 이 씬에 대해 래스터라이징 방식과 레이트레이싱 방식 양 쪽으로 렌더링을 해야 했습니다. 다양한 오브젝트를 파일에서 파싱하고, 관리하기 위해 (std::vector와 같은) 고수준의 자료구조를 활용했으며, 실제 게임 엔진에서 오브젝트가 관리되는 방식을 모방하여 다수의 오브젝트를 원활하게 조작하고 저장할 수 있었습니다. 래스터라이징의 경우 Phong shading을 GLSL로 간단하게 구현할 수 있었습니다.
- 레이트레이싱은 컴퓨트 셰이더를 이용해 구현했습니다. GPU 프로그래밍에서는 함수 재귀 스택이나 고수준의 자료구조가 존재하지 않는데, UBO나 SSBO와 같은 버퍼 오브젝트를 배열 겸 스택과 같이 활용하여 재귀적으로 작동하는 레이트레이싱을 구현할 수 있었습니다. 그 덕에 CPU로 구현했다면 수십 분이 걸렸을 렌더링 시간을 수십 초 정도로 줄일 수 있었습니다. 아울러 프레임버퍼에 렌더링을 하는 방식을 이용해 레이트레이싱 렌더링 결과를 화면으로 출력할 때 FXAA 등을 이용해 간편하게 Anti-aliasing을 줄 수 있었습니다.
- 게임 엔진의 중요한 부분 중 하나인 렌더링에 대해 배울 수 있었던 좋은 시간이었고, 또 렌더링 프로세스와 셰이더 언어에 대해 배울 수 있었습니다. 성적도 A0로 만족할 정도로 받았으며 이후 엔씨소프트 인턴

십 과정에서도 요긴하게 활용할 수 있었습니다.

경력

서울대학교 목재물리학연구실 학생연구원

- 개요: 목재 표면의 결점을 평가하는 모델에 대한 GUI 프론트엔드 제작 및 연동
- 참여 기간: 2021년 12월, 2022년 3월
- 개발 환경: Python (PySimpleGUI)
- 참여 인원: 본인 1명

설명

- 2021년 2학기 졸업논문 작성에 도움을 받은 인연으로, 서울대학교 산림과학부 목재물리학연구실에서 진행되는 “인공지능을 이용한 목재 표면 결점 분석” 프로젝트에 학생연구원으로 선발되었습니다.
- 해당 프로젝트에서 목재 표면의 결점을 평가하는 모델에 대한 GUI 프론트엔드를 [PySimpleGUI](#)를 이용하여 제작하였고, 모델 evaluation 부분을 모듈화하여 향후 활용에 용이하게 만들었습니다.
- 상세한 부분은 공개하기 어렵지만, 파이썬 프로젝트와 버전 의존성이라는 골치아픈 문제부터 [PySimpleGUI](#) 라이브러리를 이용한 편리한 GUI 제작까지 파이썬 프로그램에서 겪을 수 있는 다양한 문제점과 해결 방법을 빠르게 알아보는 계기가 되었습니다.
- 멀티쓰레딩을 지원하는 GUI를 설계하여, 해당 부분에 대해서는 [오픈소스로 공개](#)하였습니다.

엔씨소프트 2022 동계 산학협력 인턴십

- 개요: Unity 엔진 기반 배경 데이터셋 구축
- 참여 기간: 2022년 1월 ~ 2월
- 개발 환경: Unity with C#, Python(AirSim)
- 참여 인원: 본인 및 멘토(총 2인)

설명

- 2022년 1월 10일에서 2월 25일까지, 엔씨소프트 R&D센터 Vision AI Lab에서 산학협력 인턴으로 근무했습니다. "Unity 엔진 기반 배경 데이터셋 구축" 이라는 주제로 인턴십 과제를 수행했습니다.
- 오픈 소스 드론 시뮬레이션 라이브러리인 AirSim을 이용하여 과제를 진행하였습니다. 이 라이브러리에는 게임 엔진에서 드론을 움직이며 이미지를 캡처하는 기능이 있는데, 언리얼 엔진 4에서는 4종류의 이미지 데이터(일반 이미지, 깊이값 이미지, 물체 종류별 segmentation 이미지, 표면 노멀 벡터 이미지)를 모두 원활하게 수집 가능했습니다. 하지만 유니티 엔진 백엔드에서는 비정상적인 이미지가 출력되거나 아예 구현이 되어 있지 않는 등 문제가 많았습니다. 그래서 본격적인 이미지 수집 이전에, 유니티 엔진 백엔드의 기능을 언리얼 엔진과 동등한 수준으로 개선하였으며, AirSim 라이브러리를 이용해 캡처를 원활하게 할 수 있는 파이썬 기반 프로그램을 제작하였습니다.
- 유니티 백엔드의 기능을 개선하기 위해 유니티 Built-in 렌더러의 특성 및 데이터셋 정보 수집을 위해 이용되는 Replacement Shader의 작동 방식을 분석하였습니다. 유니티 셰이더(ShaderLab, HLSL) 문법을 이해하고, 이를 유니티 엔진 백엔드 기능 구현 및 각종 버그 수정, 개선에 활용하였습니다. 거의 처음 보는 분야였지만, 이전에 공부해 두었던 셰이더 및 렌더링 관련 지식이 도움이 되었습니다.
- 이후에는 유니티 엔진에서 구현된 장면별 특성을 이해하고 분석하여, 각 Scene을 데이터셋 수집에 최적화하는 과정을 거쳤습니다. 해당 과제 결과 언리얼 엔진에서만 아니라 유니티 엔진으로 제작된 장면에서도 고품질의 인공 풍경 데이터셋을 수집할 수 있게 되었습니다. 해당 데이터셋에는 실제 데이터셋으로서는 수집이 어려운 Pixel-perfect depth, segmentation data가 포함되어 있어 Image Synthesis 등에 유용하게 활용 가능할 것으로 예상합니다.
- 게임 엔진의 세부 기능을 이해하고, 라이브러리의 부족한 기능을 개선하는 과정을 체험할 수 있었습니다.