

PORTFOLIO

개발자 포트폴리오

2019 개발자 지원
최요셉 포트폴리오



PORTFOLIO

2019 개발자 지원
최요셉 포트폴리오

Index

PART #1. 프로필

Profile

이름
생년월일

약력

Education
Work Experiences
Project
Award

PART #2. 프로젝트 경험

Dragon's Treasure

프로젝트 소개
주요 개발 내용

우리는 신춘에

프로젝트 소개
주요 개발 내용

Manmulator

프로젝트 소개
주요 개발 내용



PORTFOLIO

2019 개발자 지원
최요셉 포트폴리오

PART.1 프로필



“나에게 누군가 기대한단 사실이
나를 부담지게 한다.
허나
나에게 아무도 기대하지 않을 때,
나를 무너지게 한다.”

Profile

최요셉 (Choi Yo Seb)

1994. 12. 15

Education 2013. 3 단국대학교 소프트웨어학과 입학

Work Experiences 2019. 2~6 만만테리어 하계 인턴
2018. 7~9 다우기술 O2O팀 인턴

Project 2019. 2~6 인테리어 시뮬레이터 클라이언트 개발
2018. 5~12 어드벤처 퍼즐 모바일 게임 개발
2017. 8~12 멀티플레이 액션게임 개발

Award 2017. 11 단국대학교 종합설계 경진대회 대상 수상

Links Github : <https://github.com/Choiyos/Choiyoseb-s-Portfolio>
Meeta : <https://meeta.io/@choiyos>
Dragon's Treasure : <https://www.youtube.com/watch?v=XNhZT1uVECw>
우리는 신촌에 : <https://www.youtube.com/watch?v=v-ZfY3sHq1Y>
Manmulator : <https://youtu.be/sOkGprw-97c>



PORTFOLIO

2019 개발자 지원
최요셉 포트폴리오

PART.2 프로젝트 경험

프로젝트명
Dragon's Treasure

작품설명
3D로 구현된 맵과 캐릭터를 통해 미로에서 탐색, 대전 등의 요소를 통해 모든 열쇠를 모아 가장 먼저 탈출한 사람이 승리하는 게임.

기여도



2017 단국대학교 Capstone Design

프로젝트 장르 : 멀티 액션 게임

프로젝트 인원 : 총원 3명 (개발인원 3명)

프로젝트 기간 : 2017.08.01 ~ 2017.12.31

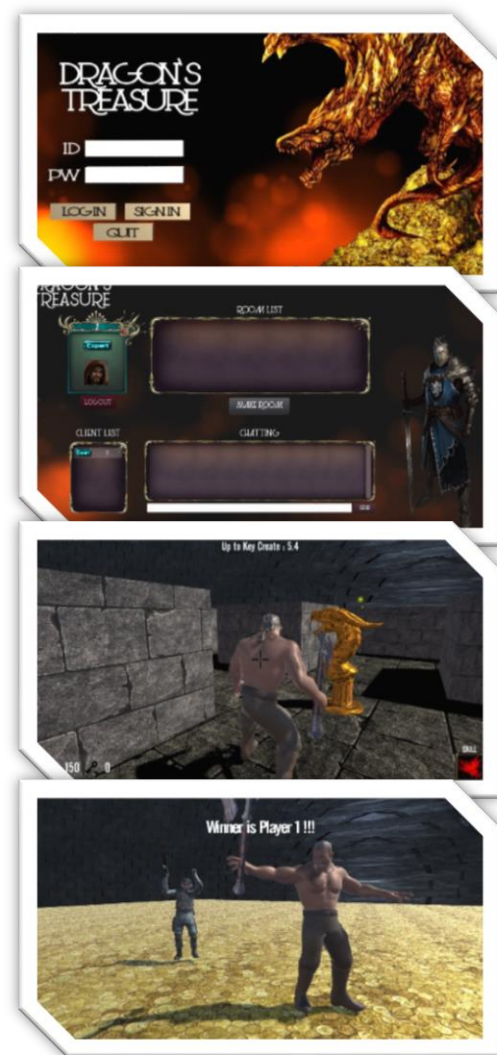
수행 역할 : 클라이언트 개발 & 서버 기능 연결

개발 환경 : C#, Unity Engine, Visual Studio

버전 관리 : Unity Collaboration

주요 개발 내용

- 서버에서 전송되는 패킷 정보를 클라이언트에 맞게 분배/변경
- 클라이언트의 정보를 서버에 맞게 가공 후 서버에게 전송
- 원활한 통신이 이루어질 수 있도록 서버개발자와 패킷 양식 협약
- 클라이언트 개발자와 서버 개발자 간에 원활한 커뮤니케이션을 위한 소통 창구 역할.





PART.2 프로젝트 경험

주요 개발 내용 : 서버-클라이언트 연결

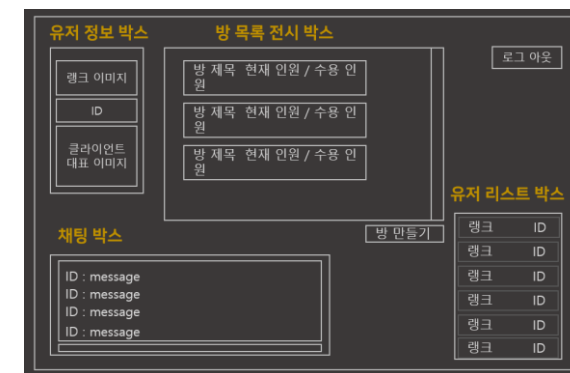
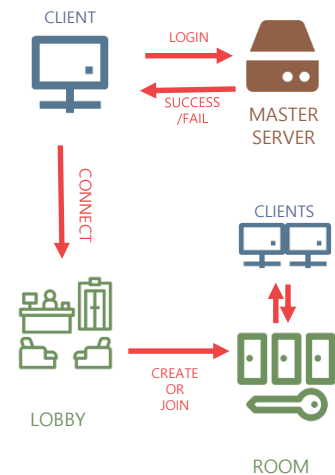
1. 마스터 서버와 통신하여 클라이언트에 적용하는 업무

서버에서 보내주는 패킷을 Header별로 구분하고, Success/Fail 여부를 구분하여 분기처리를 통해 데이터를 출력하거나 예외처리를 진행하였습니다.

기능별로 ClientListManager, ClientStateManager 등 각각의 Manager에게 구분된 패킷에 담긴 정보를 전달해주어 Manager와 연결된 속성에게 값을 띄우도록 구현했습니다.

생애 첫 프로젝트인지라, 패킷을 분석할 때의 분기처리에 대해 깔끔하게 진행하지 못한 점이 아쉽지만, 패킷 송수신의 작동 원리를 공부해보았다는 것에 의의를 두고 있습니다.

또한 서버로부터 송신한 패킷을 분석하여 다른 유저의 정보나 채팅 등을 구별하며, 이를 출력할 UI를 직접 제작하고 해당 기능을 직접 연결하는 경험 또한 할 수 있었습니다.





PART.2 프로젝트 경험

주요 개발 내용 : 서버-클라이언트 연결

2. 서버(담당자)와 클라이언트(담당자)에 대한 연결

이는 기술 외적으로, 커뮤니케이션에 대한 경험입니다.

모두 프로젝트가 처음인 학생 셋이 모여 프로젝트를 진행했던 것이라, 서로 다른 개발 영역에 대해 어떻게 소통하며 프로젝트를 발전시켜 나가야 할 지 정확하게 아는 사람은 아무도 없었습니다.

프로젝트가 진행되면서 개발업무로 서버 개발, 클라이언트 개발, 서버-클라이언트 연결+클라이언트 서브개발 세 역할이 있었는데, 제가 서버-클라이언트 연결과 클라이언트 서브개발을 맡게 되어 자연스럽게 서버 개발자와 소통하고, 클라이언트 개발자랑도 가깝게 소통해야 하는 상황으로 흘러갔습니다.

이대로 프로젝트가 끝났다면 이슈가 되지 않았겠지만, 문제는 서버 개발자와 클라이언트 개발자의 원활하지 않은 소통으로부터 시작되어, 제가 맡은 개발적인 업무 외에도 프로젝트의 방향이 어긋나지 않기 위해 다른 팀원들을 케어하고, 서로의 컴플레인이나 요구사항을 원활하게 전달하는 역할까지 맡아서



Server

네트워크 제공을
위한 TCP/IP 기반의
서버



Server-Client

사용자와 서버의
상호 작용을 위한
기능 구현



Client

사용자가 실제로
플레이하는
인게임 구축



PORTFOLIO

2019 개발자 지원
최요셉 포트폴리오

PART.2 프로젝트 경험

프로젝트명
우리는 신춘에

작품설명
좀비 감염이 발생한 학교에서 학생들 구출하는 역할을 맡은 선생님의 입장에서 플레이 하는 퍼즐게임.

기여도



2018 아주대학교 창업동아리 협업 프로젝트

프로젝트 장르 : 어드벤처 퍼즐 모바일게임

프로젝트 인원 : 총원 6명 (개발인원 2명)

프로젝트 기간 : 2018.05.01 ~ 2018.12.30

수행 역할 : 클라이언트 로직 및 시스템 구현

개발 환경 : C#, Unity Engine, Visual Studio

버전 관리 : Bitbucket(Git), SourceTree

주요 개발 내용

- 클라이언트 로직 구현
- 캐릭터 이동, 상호작용 등 이벤트들의 로직 구현
- Resources Management에 대한 로직 구현



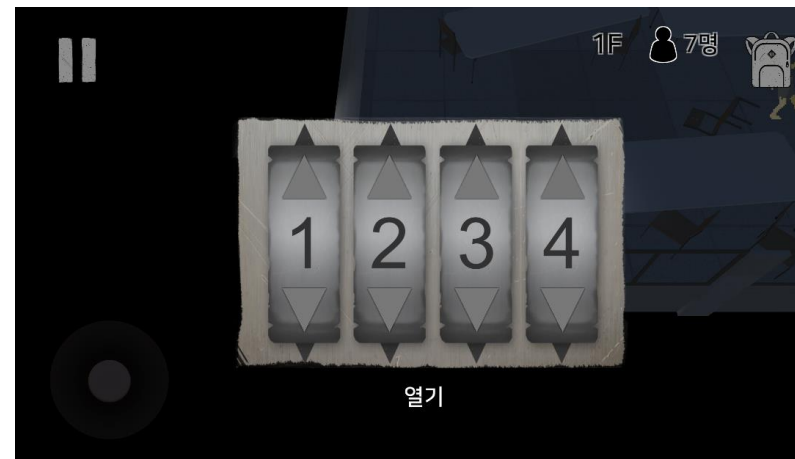


주요 개발 내용 : 이벤트 로직 구현

1. 상호작용이나 특정 상황 시 발생하는 이벤트 처리

주어진 개발 라이브러리를 통하여 플레이어가 특정 상호작용을 하거나, 일정 진행도를 달성했을 때 Object Pool에 Dialog들을 Push하여 이벤트를 연출하고, 기획된 대로의 상황을 플레이어에게 충실하게 전달하는 역할을 맡았습니다.

이벤트 로직 구현이라는 업무를 맡으면서 모델의 애니메이션이나 분기처리, 코드의 재사용 등 라이브러리와 시스템에 대한 이해, 그리고 전반적인 개발 일정 소화에 대한 것을 꾸준히 배우며 소화하려는 노력을 기울였습니다.





PORTFOLIO

2019 개발자 지원
최요셉 포트폴리오

PART.2 프로젝트 경험

프로젝트명
Manmulator

작품설명
사용자가 직접 인테리어 자재를 보고, 업체를 찾아다닐 필요 없이 시뮬레이터 솔루션을 통해 자신의 집을 직접 꾸미고, 여러가지 인테리어가 적용된 모습을 볼 수 있다.

기여도



2019 만만테리어 프로젝트 '만물레이터'

프로젝트 장르 : 인테리어 시뮬레이터 솔루션

프로젝트 인원 : 총원 5명 (개발인원 2명)

프로젝트 기간 : 2019.02.01 ~ 2019.06.30

프로젝트 링크 : <http://manmanterior.com/manmulator/manmulator.php>

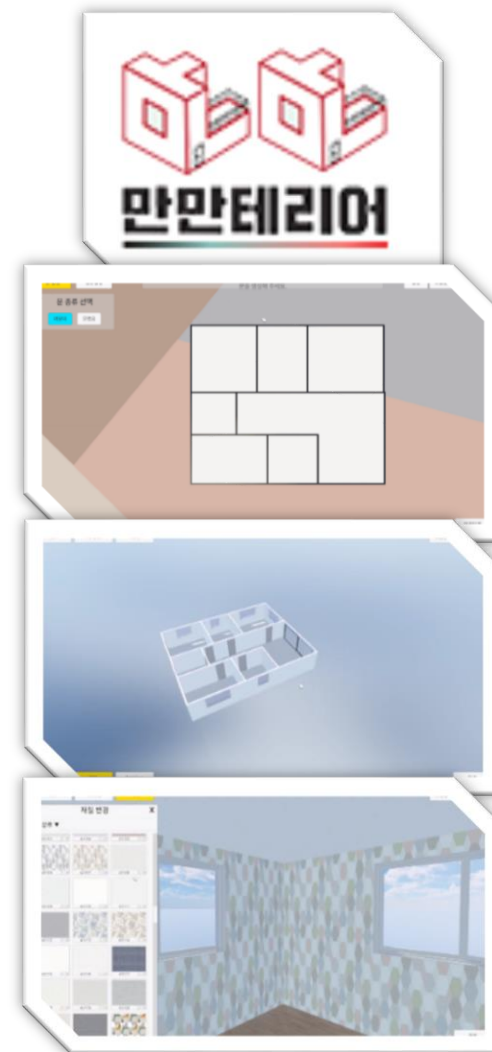
수행 역할 : 클라이언트 로직 및 시스템 구현

개발 환경 : C#, Unity Engine, Visual Studio

버전 관리 : Bitbucket(Git), SourceTree

주요 개발 내용

- 프로젝트 구조 설계 (MVP Model+UniRx)
- 시뮬레이터 클라이언트의 핵심 로직 구현





PART.2 프로젝트 경험

주요 개발 내용 : 프로젝트 구조 설계

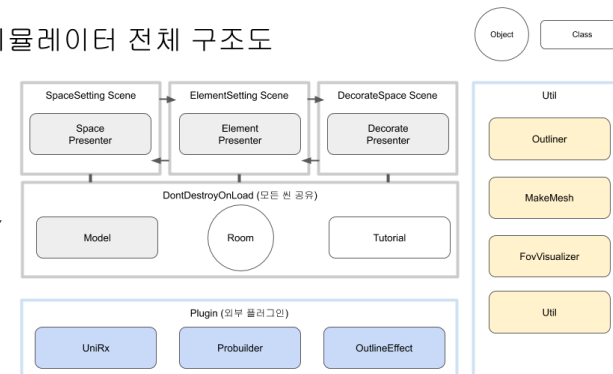
1. MVP Model 채택 & 구현

해당 프로젝트의 프로토타입을 제작해본 후, Re-Building 하는 과정에서 프로젝트의 모델에 대한 고민을 하다가 웹에서 사용되고 있는 MVC모델과 비슷한 MVP모델이 저희의 프로젝트와 시너지가 좋을 것 같아 고려하게 되었습니다.

MVP모델에서 유니티 엔진은 자체적으로 View 역할을 맡게 되며, 프로젝트의 데이터를 담당할 Model을 완전히 분리하여 두고, 사용자의 Input에 대응하여 실행하는 Presenter들을 분리하여 로직을 관리하기 간편하게 정리하였습니다.

MVP를 채택함으로써 오는 장점으로는 로직과 데이터를 분리해서 관리할 수 있다는 것이었지만, Presenter에 로직이 모여 있다 보니 코드의 가독성이 떨어진다는 문제점이 있어 Partial Class와 UniRx 등의 사용으로 해당 문제를 해결하려고 노력하였습니다.

시뮬레이터 전체 구조도





주요 개발 내용 : 클라이언트 핵심 로직 구현

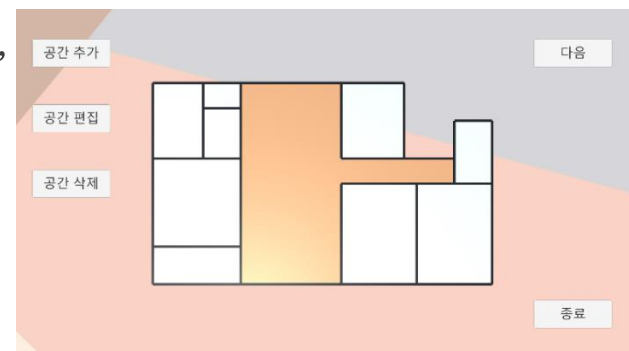
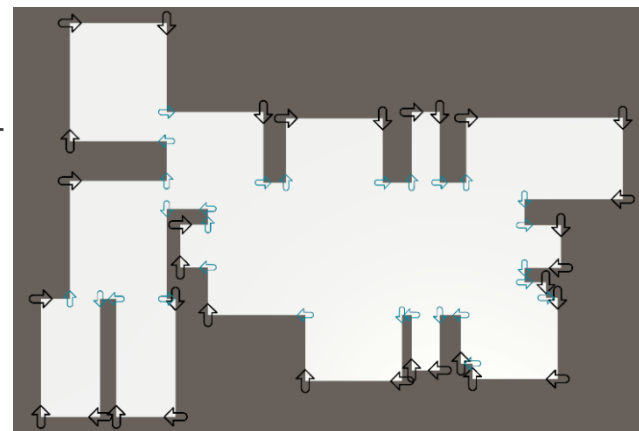
1. 이전에 없던 솔루션, 이제는 있어야 할 로직

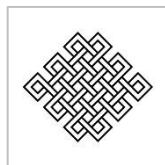
해당 프로젝트는 ‘사용자가 자신의 집을 손쉽게 재현해낼 수 있어야 한다’는 미션을 가지고 있었는데, 집을 ‘방’ 단위로 분해하여 한 단계씩 붙여 나간다면 직관적으로 자신의 집을 도면화 시킬 수 있다고 생각했지만, 이를 그대로 구현하기에는 많은 어려움이 따랐습니다.

여러 직사각형이 겹쳐져 하나의 ‘방’을 만들기 위해서는 해당 상황에 따른 로직이 필요했고, 저는 나름대로 혼자만의 연구를 통해 사각형의 외곽점과 교차점에 서로 다른 방향성을 부여하여 일관적으로 다른 직사각형들을 같은 방이라고 인식하는 것에 성공했습니다.

C#의 List와 Linq, Dictionary를 적절히 이용하여 로직을 구현하였고, 해당 로직이 정상적으로 돌아가기 위해 수 많은 예외들에게 구타를 맞고 나서야 그나마 돌아가는 수준으로 끌어올릴 수 있었습니다.

이후에도 해당 프로젝트에서 오브젝트끼리 자석 기능과 2D>3D좌표계의 변환을 통한 모델 오브젝트 생성, 주로 사용자 친화적인 UI/UX구성+편리한 조작감에 중점을 두고 개발을 진행하였습니다.





Thanks.

2019 개발자 지원
최요셉 포트폴리오