Project Portpolio

YOO SANG
KEON

Short introduction

안녕하세요 프로그래머 직군에 지원한 유상건이라고 합니다.
해당 자료는 프로그래밍을 배우기 시작하며 개인 혹은 공동 작업으로 개발했던 프로젝트를 모아 놓은 자료입니다.
공유가 제한 되는 현업 프로젝트 외, 모든 프로젝트들은 Github과 개인 Blog를 통해서 모든 소스코드 및 플레이 영상 확인이 가능합니다

Github : https://github.com/ysk1965
Blog : https://blog.naver.com/enter_maintanance

Project LIST

FancyPlanet

사용언어 : C++

사용도구 : DirectX12 + PhysX + IOCPserver

개발인원 : 3명(**클라이언트** 2, 서버 1)

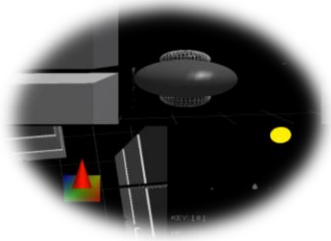
니별내별

사용언어 : C#

사용도구 : Unity + IOCPserver

개발인원 : 8명(**클라이언트** 2, 서버 1, 기획 2, 아트 2, 음향 1)

MazeRunner3D



사용언어 : C++

사용도구 : OpenGL + TCPserver

개발인원 : 개인 프로젝트

CBA Retreat(IOS App)

사용언어 : Swift

사용도구 : Xcode + Firebase

개발인원 : 개인 프로젝트

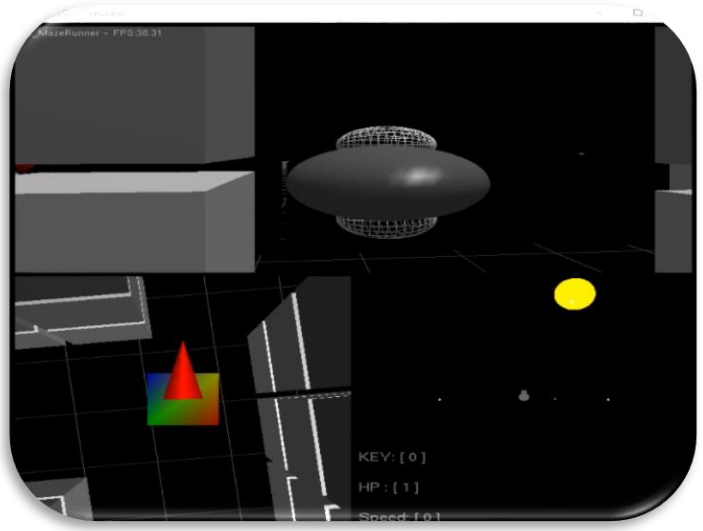
TERA ORIGIN

사용언어 : C++

사용도구 : OpenGL + GLSL

개발인원 : 개인 프로젝트

MazeRunner3D

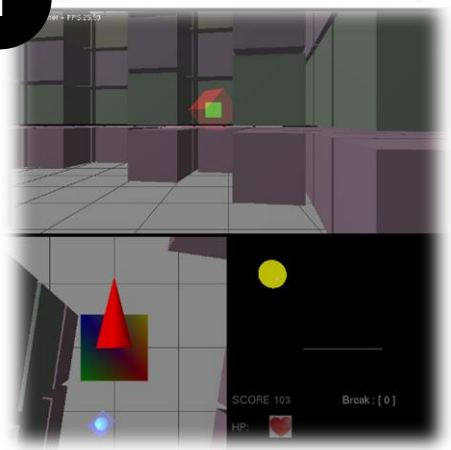


사용언어 : C++ (PC)
사용도구 : OpenGL + TCPserver
개발인원 : 개인 프로젝트

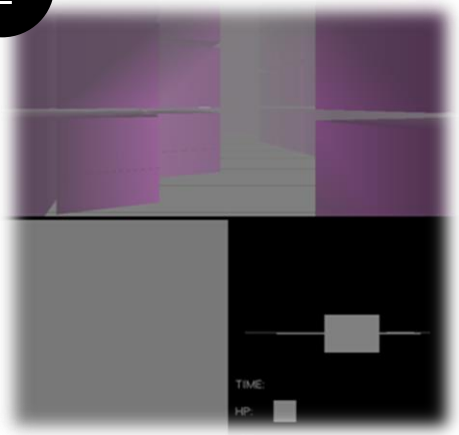
게임개요 MazerRunner3D는 3가지 시점으로 미로를 찾아가는 미로찾기 게임입니다. 평범한 미로 게임과 차별되는 요소로 밤/낮 시스템을 추가하여 밤에는 유령이 지나다니며 플레이어는 유령을 피해서 목적지에 도달해야 합니다.
해당 프로젝트는 과제전에 제출한 이후 STL을 배운 뒤 리플레이 시스템을 추가했고, TCP서버 통신을 배운 뒤에는 멀티 클라이언트 버전을 제작하였습니다.

Github : https://github.com/ysk1965/enter_maintanance
영상확인 Blog : (v.TCP)
https://blog.naver.com/enter_maintanance/221263177431
https://blog.naver.com/enter_maintanance/220968788109

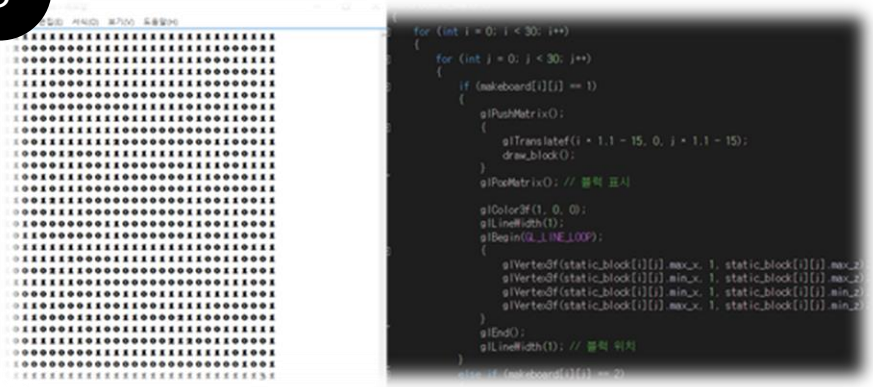
1



2



3



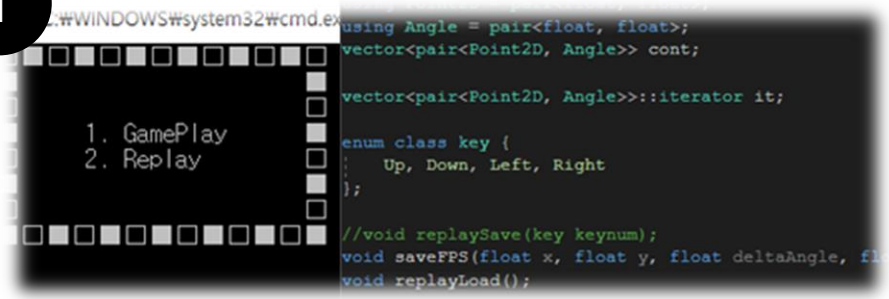
MazeRunner3D

- 3가지 시점의 View
  - 게임 내에 3가지 뷰를 적용하였습니다. 위쪽의 뷰는 플레이어가 바라보는 정면을 의미하고 좌하단은 탑뷰 시점, 우하단은 멀리서 바라보며 밤/낮을 구분하는 해가 얼마나 이동했는지와 Status를 확인할 수 있습니다.

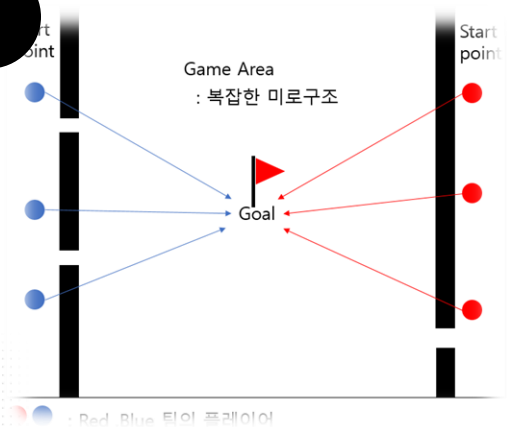
- Light 및 Fog System
  - 게임의 주요한 요소 중 하나는 밤 낮 시스템입니다. DirectionalLight를 가진 태양은 계속해서 360도를 돌며 180도를 도는 순간 밤/낮이 계속해서 바뀌게 됩니다.
  - 밤이 되었을 경우엔 Fog System을 활용한 안개가 깔리고 SpotLight를 통해 전방에 빛을 비출 수 있도록 했습니다. 해당 값들은 Item 획득 시 기능이 더욱 향상될 수 있도록 만들었습니다.

- Map
  - 해당 게임의 맵은 메모장의 파일입출력을 통해서 받아옵니다. 기본적으로 VeryEasy ~ Very Hard 까지 5가지 맵을 제공해주며 처음 게임 시작했을 시 선택한 맵의 메모장을 불러와 맵이 생성됩니다. TCP 멀티클라이언트 버전의 경우에는 서버에서 클라이언트로 맵을 뿌려주게 됩니다.

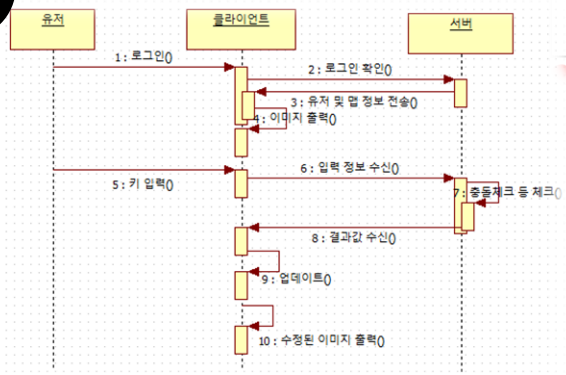
1



2



3



MazeRunner3D

- ReplaySystem
  - STL을 배운 뒤 리플레이에 필요한 자료들을 저장하여 리플레이를 할 수 있도록 만들었습니다. 기본적으로 Point2D(pair<float, float>), Angle(pair<float, float>)로 플레이어의 위치와 회전 정도를 받으며 vector<pair<Point2D, Angle>>로 저장합니다. Load시에는 해당 값을 차례대로 실행하며 좌, 우 키로 되감기도 가능합니다.
- 멀티클라이언트 미로 구조 변경
  - 기존의 미로 구조는 출발 지점과 도착 지점만 정해져 있는 형식이었습니다. 하지만 멀티 클라이언트가 되면서 다른 게임성이 요구됐고 두 팀으로 나눠 다른 위치에서 시작하며 가운데 있는 동일한 목표지점에 먼저 도착하는 팀이 이기는 구조로 바뀌었습니다.
- TCP서버 구조
  - TCP서버를 활용해서 여러 클라이언트가 접속하는 구조로 변경해주었습니다. 이미지는 기본적인 서버와 클라이언트의 소통 플로우 차트입니다.
  - 클라이언트에서 플레이어의 좌표와 아이템 획득 시 아이템 정보를 서버에 넘겨주고 서버는 이를 다른 클라이언트에 뿌려주는 구조입니다. 맵 정보는 서버에서 관리합니다.

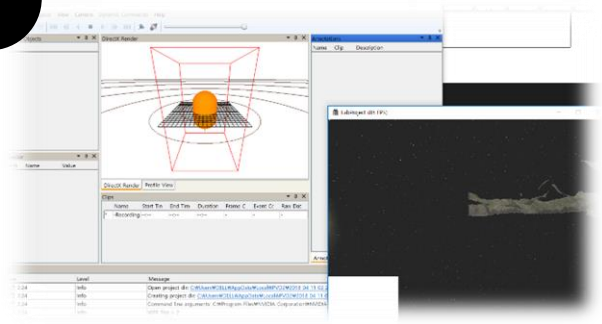
Fancy Planet

사용언어 : C++ (PC)
사용도구 : DirectX12 + PhysX + IOCPserver
개발인원 : 3명 (**클라이언트** 2명, 서버 1명)

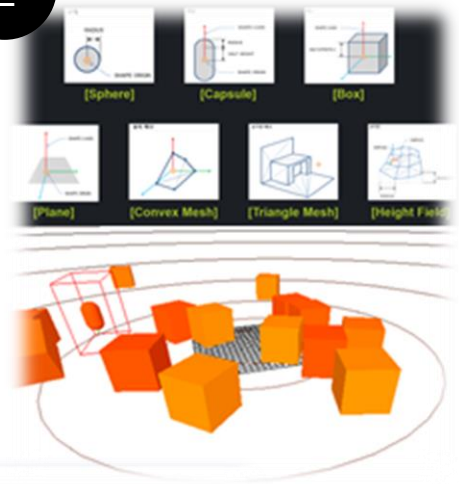
게임개요 : FancyPlanet은 기본적인 FPS 게임의 룰에 중력 시스템과 점령 시스템을 추가하여 만든 게임입니다. 해당 프로젝트는 대학교 졸업작품으로 진행했었으며 게임 내 원활한 물리 작용의 적용을 연구중점으로 둔 게임입니다.
저는 3명의 개발 인원 중 클라이언트 프로그래머를 담당했으며, 주로 DirectX12 프레임워크 제작과 중력 및 물리 시스템 부분을 전담하였습니다.

Github : https://github.com/ysk1965/enter_maintanance
영상확인 Blog : https://blog.naver.com/enter_maintanance/221363153513

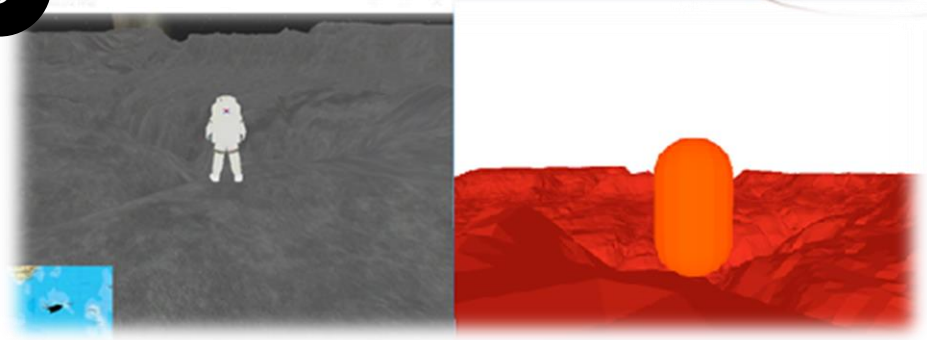
1



2



3



Fancy Planet

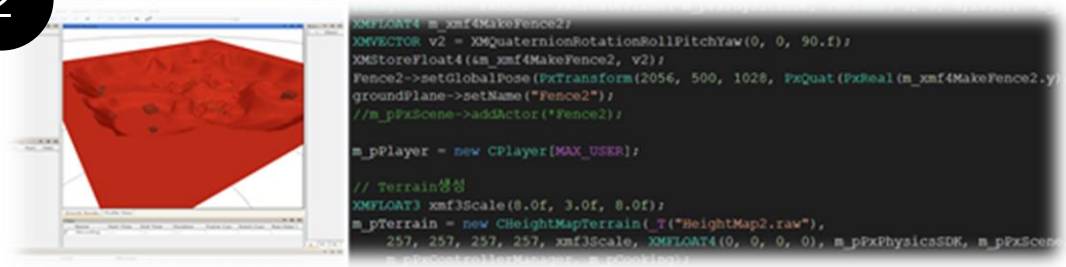
- PhysX Framework
  - 기존 DirectX12의 프레임워크를 구현한 후 PhysX 프레임워크를 입히는 작업을 하였습니다. (PxPhysics, PxScene 등... 기본적인 피직스 작업을 위한 초기화 및 필요 부분에 적용하였습니다.)

- PVD & PxCharacterController
  - PVD는 PhysX의 물리 작용을 확인할 수 있는 중요한 Debugger입니다. 해당 디버거를 프레임워크에 적용하여 게임 내 물리작용을 즉시로 확인 가능하도록 하였습니다.
  - 그 후 PxCharacterController를 만들어 컨트롤러 관리자를 만들고 생성된 모든 컨트롤러를 추적해 모든 캐릭터의 이동을 통제하도록 구현하였습니다.

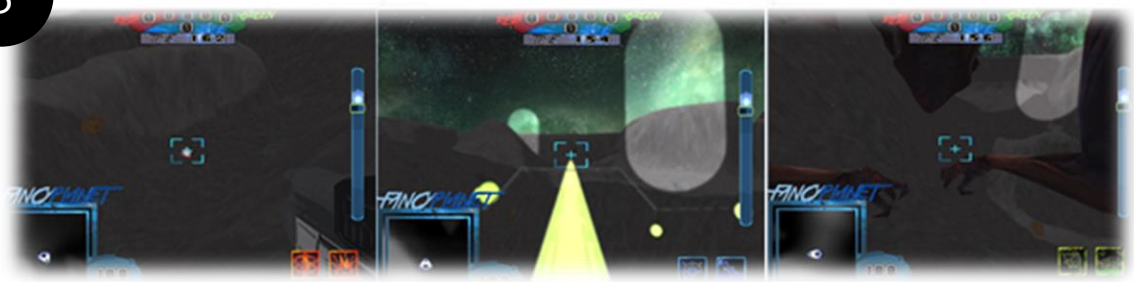
- PhysX Terrain
  - 유니티 Terrain Export를 통해서 HeightMap을 가져왔습니다. 가져온 HeightMap의 메쉬마다 메모리를 할당해주어 적용된 TerrainMesh를 모두 물리 작용 가능한 Actor로 만들어주었습니다. 메쉬 단위의 물리 적용이 가능하도록 하였기 때문에 플레이어는 더욱 디테일한 지형지물 이용이 가능합니다.

1

2



3



Fancy Planet

- 중력 시스템
  - 게임 내 중력 시스템을 적용하였습니다. 해당 중력 상태는 이미지 우측에 있는 중력 바로 확인 가능하며 중력 변동 시에는 포스트 프로세싱을 통해 흔들리는 효과를 보여주게 됩니다.
- PhysX 서버 이동
  - 중간발표까지는 PhysX를 클라이언트에 적용해서 물리 작용이 가능한 프레임워크를 구현하는 것을 중점으로 잡았었습니다. 그러나 작업을 마친 뒤 확인해보니 서버를 통해 여러 클라이언트가 접속하는 환경에서 불필요한 중복 처리가 많아졌습니다. 그래서 PhysX를 서버에서 관리하도록 하고 클라이언트는 캐릭터 애니메이션과 렌더링만 하는 용도로 변경하였습니다.
- 물리작용하는 기술들
  - 게임 내 3가지 캐릭터가 있고 3가지 캐릭터는 이동 스킬과 공격 스킬 한 가지씩을 가지고 있습니다. 이 스킬들은 넉백, 스플레쉬 데미지, 이중 점프, 순간 이동 등 물리 작용을 활용한 스킬입니다.

니별 내별

사용언어 : C# (Mobile)
사용도구 : Unity3D + IOCPserver
개발인원 : Team HSLD 8명(**클라이언트** 2, 서버 1, 기획 2, 아트 2, 음향 1)

게임개요 : 니별 내별은 대학생 게임제작 연합 동아리 Bridge의 18년도 후반기 프로젝트 중 하나입니다. Team HSLD의 니별내별 게임은 2월 말에 진행했던 LINE 동아리 공모전에서 15개의 프로젝트 중 공동 2등을 했습니다.
니별내별은 320개의 메쉬를 갖고 있는 행성을 보드판으로 둔 2인 PVP 턴제 보드게임이며 개인이 갖고 있는 특별미션을 상대방에게 숨기며 먼저 달성하는 플레이어가 이기게 되는 게임입니다.

Github : https://github.com/GameForPeople/TeamHSLD
영상확인 Blog : https://blog.naver.com/enter_maintanance/221459621769

1

2

3

니별 내별

- 메쉬 보드 판 시스템
  - PvP 보드게임의 핵심인 보드판 시스템입니다. 게임 내 보드는 1번 그림에 보이는 320개의 메쉬로 이루어진 구입니다. 그리고 아래 5가지 지형카드 중 하나를 선택한 뒤 보드 판의 메쉬를 터치하는 방식으로 설치하게 됩니다. 구를 보게 되면 신전과 같은 거점을 볼 수 있는데, 초기에는 거점 부분만 선택 가능하며, 주변의 모든 메쉬를 점령하게 되면 거점점령 상태가 되고 이후에는 자유롭게 다른 위치에 지형설치 할 수 있습니다.
- 서버 연동
  - TeamHSLD는 서버 프로그래머 한 명과 클라이언트 프로그래머 2명으로 이루어져있습니다. 저는 클라이언트 프로그래머로 보드 시스템 및 쉐이더 부분을 맡았고 클라이언트와 클라이언트 사이에서 턴 종료 후 메쉬 관련 정보를 서버로 전달해주는 부분 제작을 맡았습니다.
- 미션 구현
  - 게임의 승패는 미션 수행 여부에 있습니다. 메인미션은 서로 다르며 해당 미션을 먼저 성공하면 승리하게 됩니다. 만일 둘 다 수행하지 못한 체 20턴이 지나면 공통 미션을 누가 더 많이 수행 했는지에 따라 승패가 나뉩니다. 미션 창은 게임 내에서 언제든 좌측 깃발을 눌러 확인할 수 있습니다.

1

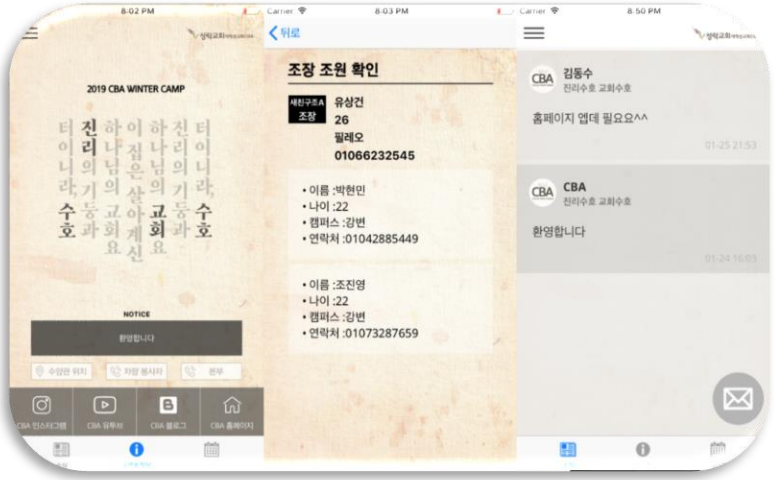
2

3

니별 내별

- 자체제작 Shader
  - Unity ShaderLab를 사용해서 자체 제작 쉐이더를 만들었습니다. 보드판 전반적인 DefaultMesh 쉐이더는 카메라 방향과 메쉬의 노말 값을 비교하여 알파 값을 조절한 투명Rim Shader를 적용해주었고 상대방이 획득한 메쉬와 플레이어가 획득한 메쉬를 구분하기 위해 빗금 무늬를 주는 쉐이더를 제작하여 적용해주었습니다.
- 튜토리얼 시스템
  - 게임을 처음 접하는 유저들을 위해 튜터리얼 모드를 만들었습니다. 처음 접속 시 튜토리얼 진행여부를 물어보고 그에 응하면 튜토리얼이 진행됩니다. 만일 이 때 하지 않더라도 차후에도 설정에서 Tutorial 버튼을 누르면 언제든 진행 가능합니다.
- Casual모드
  - Classic모드의 경우에는 320개 메쉬로 된 구에서 진행됩니다. 메쉬가 많은 만큼 보드판의 개수가 많아지는 데 보드판이 많아질수록 한 게임당 진행하는 시간이 길어집니다. 그렇기에 이를 단축하기 위해 Casual모드를 추가 개발했고 해당 모드는 80개 메쉬의 구에서 진행되며 한 게임에 10분 정도 소요됩니다.

CBA Retreat



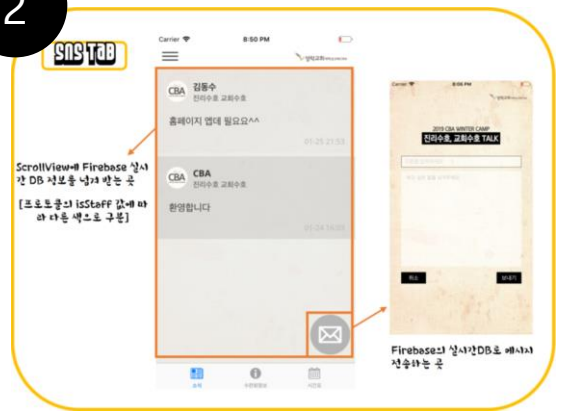
사용언어 : Swift (Mobile)
사용도구 : Xcode + Firebase
개발인원 : 개인 프로젝트

앱 개요 : CBA Retreat 앱은 교회 수련회를 위해 제작된 앱입니다. 기본적으로 수련회 관련 정보들을 받아 볼 수 있으며, 수련회 시에 SNS기능을 통해 건의를 하고 건의에 대한 답변도 받을 수 있습니다.
정보를 받거나 공지Noti를 받는 등의 모든 기능은 Firebase를 통해서 받고 볼 수 있도록 만들었습니다.

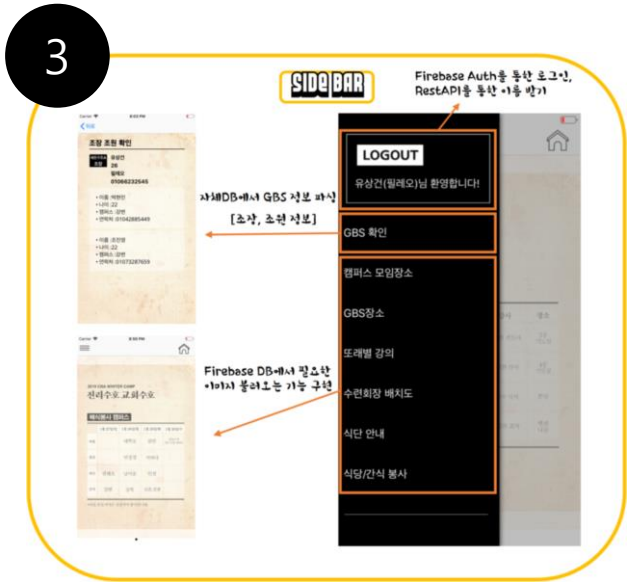
Github : https://github.com/ysk1965/IOS_CBAProject
영상확인 Blog :
https://blog.naver.com/enter_maintanance/221451792626

1

2



3



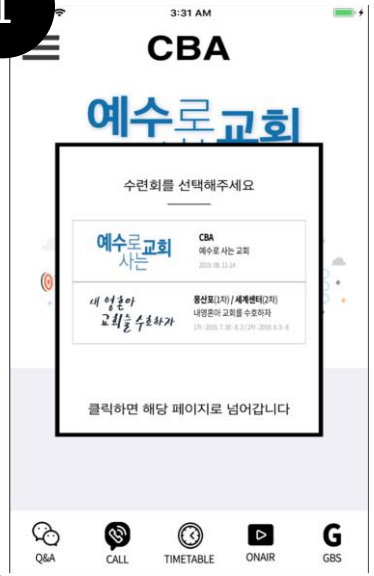
CBA Retreat

- Information
  - 인포메이션 받는 곳에는 수련회에 대한 기본적인 정보 모두를 담았습니다. 수련회장의 위치는 MapKit을 활용해서 받도록 했고, 본부와 차량 봉사자에 대한 연락도 가능하도록 만들었습니다. Notice에 있는 공지는 Firebase의 CloudMessage를 통해서 해당 구독 정보가 담긴 휴대폰에만 정보를 전달해주도록 했습니다.

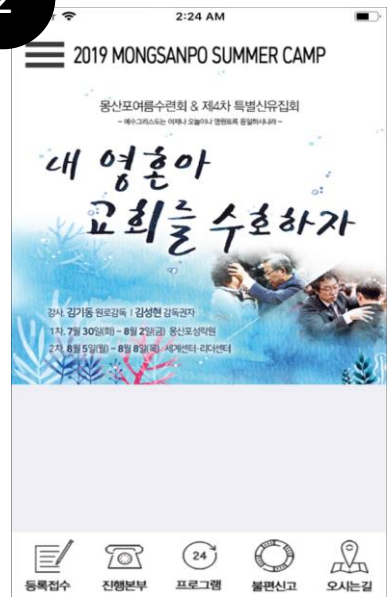
- SNS Tab
  - SNS Tab은 Firebase의 실시간 DB에 있는 정보를 받도록 만들어주었습니다. 기본적으로 ScrollView로 되어있기 때문에 여러 메시지를 받더라도 스크롤로 확인할 수 있으며 프로토콜의 isStaff 값에 따라서 봉사자와 문의 하는 사람의 메시지 색을 다르도록 해주었습니다. 또한 메시지 보내는 기능이 있어서 건의사항을 즉시로 보내고 받을 수 있습니다.

- SideBar
  - Sidebar를 통해서 수련회 관련해 여러 정보를 확인할 수 있습니다. 그리고 Firebase Auth를 통해 아이디 회원가입과 로그인을 할 수 있고 GBS확인을 누르면 REST API로 되어 있는 자체DB와 연결되어 조장 조원 정보를 확인 할 수 있도록 만들었습니다.

1



2



3

CBA Retreat

- 수련회 교환 시스템
  - 초기에는 일회성 앱으로 사용하기 위해 개발했기 때문에 대부분의 데이터는 고정적인 이미지들로 들어가 있었습니다. 하지만, 앱 사용에 대한 피드백이 좋게 나와 17년 이후 매 분기 수련회 마다 사용하게 되었습니다. 그래서 매 분기 마다 데이터 수정을 위한 빌드는 비 효율적인 것 같아 전반적으로 데이터 기반으로 변경하였습니다.

  - 해당 데이터들은 Firebase로 관리되도록 하였고, 개발 이후에는 Firebase에 데이터만 수정해주면 앱을 실행할 때 변경 여부를 확인하고 필요 시 해당 데이터를 다운로드 받아 캐싱하여 다음 수련회 버젼으로 사용할 수 있게 되었습니다.

  - 또한 추후에는 CBA뿐 아니라 교회 내 다른 수련회에서 사용할 수 있도록 수련회 이동 시스템을 구축했고, 수련회를 선택하면 해당 수련회 관련 정보를 담은 앱으로 교체 되도록 만들었습니다. 해당 부분을 개발하면서 하단 바 및 사이드 바 부분도 수련회 특색에 맡게 필요한 컨텐츠가 있다면 빌드에 추가하였고, 이 버튼 부분도 데이터만 넣어주면 세부 내용은 쉽게 변동 할 수 있도록 개발하였습니다.

Tera Origin

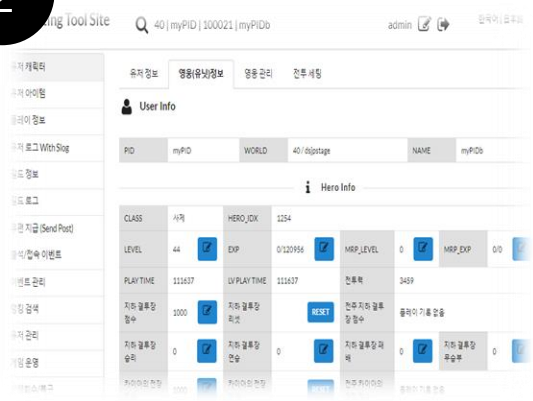
사용언어 : C#, GoLang ,Rua, Vue.js
사용도구 : Unity, GoLand, MySQL
개발인원 : Krafton Squall, 70인

앱 개요 : 크래프톤 스콜 스튜디오 프로그램실 컨텐츠 팀에 속해서 작업했던 프로젝트입니다. TERA Origin은 기존 한국 TERA M의 일본 버전이며 1년 동안 재직하며 출시 준비 부터 출시 후 서비스까지 함께 하였습니다.

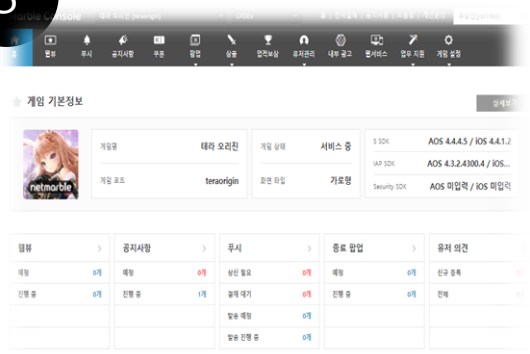
https://teraorigin.netmarble.jp/

1

2



3



TERA Origin

- 사용한 언어 및 툴
  - 해당 프로젝트를 진행하며 새로운 기술에 대해 많이 배우고 개발하게 되었습니다. 자주 사용해봤었던 Unity, MySQL 뿐만 아니라, 클라이언트에서는 빌드 없이 코드 기반으로 작업되는 Rua를, 서버에서는 Golang을, DB관련 부분은 Redis, 웹 개발에는 Vue.js를 사용하며 새로운 개발 툴 및 기술들을 습득할 수 있었습니다.

- 운영툴 작업
  - 게임 프로그래머를 준비해왔기 때문에 학부에서는 주로, Untiy, DirectX 등의 게임 개발 부분 위주로 공부해왔습니다. 하지만, 현업으로 가게 되면서 개발한 게임을 정상적으로 서비스하기 위해 운영툴이 필수적인 것을 알게 되었고, 해당 운영툴을 개발하기 위해 GoLang을 백엔드로, Vue.js를 프론트엔드로 습득하여 개발하였습니다.

- SLog 작업
  - TeraOrigin은 넷마블과 퍼블리싱 협업을 하며 진행된 프로젝트입니다. 퍼블리싱을 하기 위해서는 관리해야할 로그 정보들이 있는데, 해당 정보를 Slog라 하고 NEMO라는 관리 툴을 사용하였습니다. 저는 TeraOrigin 내에각 컨텐츠마다 넘겨줘야할 로그 들이 있을 때 해당 API를 만들어 NEMO로 Slog를 보내는 작업을 담당해서 하였습니다.

1

2

3

TERA Origin

- 레이븐 업적
  - TeraOrigin은 한국에서 오픈했던 TeraM을 일본 현지화하여 오픈 한 게임입니다. 그 중 일본성향에 맞게 추가 되었던 부분이 수호자의 역할을 하는 레이븐이었는데, 저는 레이븐마다 플레이어가 성취감을 가질 수 있도록 해주는 업적 부분을 개발하였습니다.
  - 레벨, 획득, 레이븐 스킬 사용 등등... 다양한 부분에 적용되었고, 해당 Key값을 만들어준 뒤 Key와 User_seq를 받아 Summary에 업적을 쌓아두고 해당부분이 Goal에 도달하면 이루는 방식으로 개발했습니다. 해당 컨텐츠의 DB설계, 클라이언트 작업, UI작업, API작업, 운영툴 작업 등 모든 부분을 개발하였습니다.

- 이벤트 확장
  - 기존에 사용했던 출석 이벤트를 확장했고, 이벤트 가챠 컨텐츠를 개발하였습니다. 이벤트 가챠는 이벤트 기간동안 이벤트 아이템을 소모해 가챠 기회를 주는 컨텐츠입니다.
  - 가챠로 획득할 수 있는 리스트들은 5회차로 나눠져 있고, 해당 리스트에서 특별 아이템을 뽑았을 시 다음 가챠 리스트로 넘어가는 방식이었습니다. 해당 작업은 DB설계 보다는 클라이언트 작업과 UI작업, API 통신 부분 개발이 주를 이루었고, 뽑을 수 있는 아이템을 설정할 수 있는 운영툴도 개발하였습니다.