

레이븐 업적 (Client(Unity, NGUI, Rua), Server(GoLang), DB(MySQL), 운영툴(Vue.js))

- 레이븐은 플레이어를 따라다니며 도와주는 수호자 컨셉의 수집형 캐릭터입니다. 그 세부 콘텐츠인 레이븐 업적은 획득한 레이븐의 성장 과 플레이를 유도하는 것을 목적으로 개발되었습니다. 레이븐 마다 업적의 종류는 상이하고, 도감으로 확인 가능하도록 개발되었습니다.



- 플레이어가 해당 업적 관련된 플레이를 완료하게 되면 클라이언트에서 서버로 API를 보내도록 작업하였습니다. 그 후 서버에서는 SUMMARY라는 DB에 USER_SEQ, SUMMARY_KEY, VALUE 등의 데이터를 저장하고, ACHIEVE DB에 있는 정보와 비교해서 검증 후 카운트를 증가 시키고 현재 카운트를 클라이언트에 보내게 됩니다.
- 만일 ACHIEVE DB에 맞게 목표치에 도달하게 되면 클라이언트에 완료됐음을 알려주는 정보도 함께 보내게 됩니다. 그리고 클라이언트는 해당 정보를 받았을 때 위 이미지와 같은 Alert Popup으로 UI를 통해 보여주도록 하였습니다.



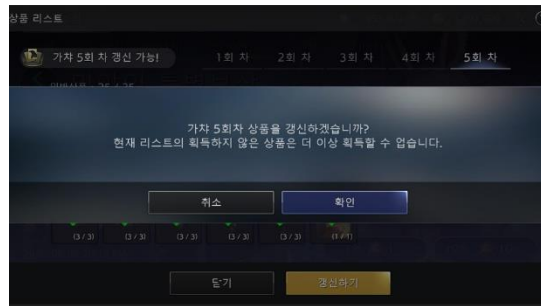
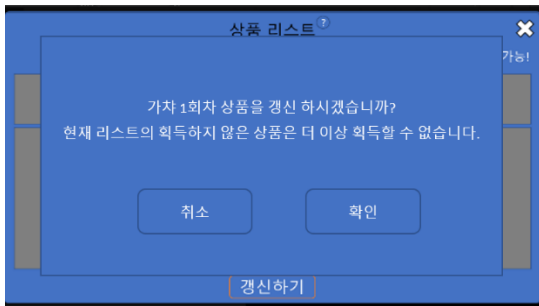
- 레이븐 업적의 도감 페이지를 개발하였습니다. 이미지에서 확인할 수 있듯 도감 페이지에 들어가게 되면 업적이 남아 있는 레이븐과 완료된 레이븐을 구분할 수 있고, 필터를 통해서 필요한 레이븐만 확인할 수도 있습니다.
- 기존에 있던 도감(아이템, 장비) 루트에 레이븐 탭을 추가하여, 처음 도감을 들어오면 먼저 서버에서 레이븐 정보 리스트를 받아온 후 클라이언트에서 Dictionary로 캐싱하도록 하였고, 받아온 데이터를 토대로 NGUI를 사용해 화면에 보일 수 있도록 하였습니다. 필터는 받아온 데이터 안에서 필요한 데이터만 따로 뽑아 남게 만드는 방식으로 작업했습니다.



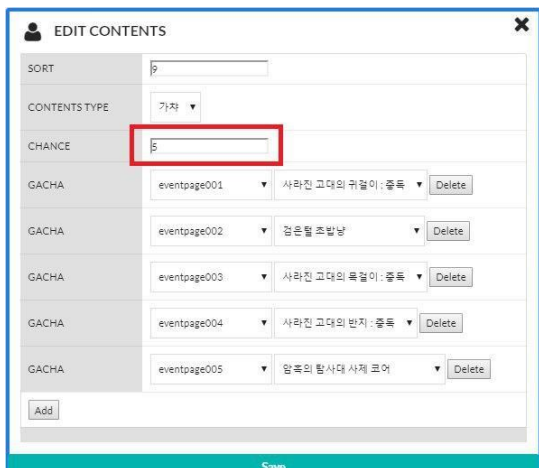
던 가차 시스템 로직으로 이동하여 아이템을 뽑게 되고 아이템을 획득하게 됩니다.



- 이벤트 기본 UI페이지에서 리스트 확인 버튼을 누르면 나오는 팝업입니다. 해당 팝업을 누르면 이벤트 정보를 서버에서 받게 되고 해당 정보를 기반으로 각 회차마다 Dictionary로 캐싱해줍니다. 그리고 회차를 누를 때 마다 회차에 맞는 캐싱 했던 데이터를 불러오고 페이지를 업데이트 해줍니다. 아이템을 획득했을 시 획득한 아이템은 딴드 처리해줬습니다.



- 유니크 상품을 획득할 시 서버에서 검증 후 클라이언트에 완료 됐음을 알려줍니다. 클라이언트에서는 다음 회차로 넘어갈 수 있다는 팝업을 띄우고, 만일 컨펌 확인을 하게 되면 클라이언트에서 다음 회차로 넘어갔음을 서버에 보내고, 서버에서는 새로운 회차에 대한 아이템 리스트 및 회차 정보를 보내줍니다.



- 운영툴에 이벤트 가차의 가차 리스트를 세팅할 수 있는 부분을 구현했습니다. 기존 5회차이지만, 이후에 확장성을 위해 해당 부분을 CHANCE라는 부분에 입력할 수 있게 해주어 5회차 이상의 정보를 세팅할 수 있도록 했습니다. 숫자를 입력하면 GACHA를 선택할 수 있는 공간이 생기고, 좌측 드롭다운에는 기존 가차 운영툴에 리스트 세팅하는 부분에 세팅 되어있던 리스트를 받아옵니다. 선택할 시 우측에 해당 가차 리스트가 드롭다운에 들어가고 해당 가차 품목이 세팅되게 됩니다.

조각의 전언 작업 (Client(Unity, NGUI, Rua), Server(Golang), DB(MySQL))

- 조각의 전언은 플레이어가 현재 지역에 맞출 수 있으려면 어떤 콘텐츠를 해야하고, 어떻게 전투력을 올려야 하는지 안내해주는 콘텐츠입니다. 테라 Origin이 시즌2로 넘어오면서 항마력이라는 정보가 생겼는데, 맵에서 요구하는 항마력과 자신의 항마력을 비교하여 요구치보다 더 낮을 시 Alert가 나오도록 하는 콘텐츠입니다.



- 플레이어가 전투하는 필드로 나가게 되면 필드에서 요구하는 항마력 정보를 받아오게 하고, 현재 플레이어의 항마력과 비교해서 요구치에 미달된다면 항마력 Alert를 띄워줍니다. Alert를 누를 시 위 이미지와 같은 사이드바 팝업이 나오게 되고, 항마력의 조각의 전언 콘텐츠 부분을 추가하였습니다.



- 항마력 부분을 선택하면 위 이미지와 같은 페이지를 열어줍니다. 열어줄 때 서버에서 해당 캐릭터에게 필요한 항마력 정보를 받아오게 되고 페이지에 항마력 관련된 정보를 표시해 줍니다. 또한 데이터에 맞게 하단에는 항마력을 증가시킬 수 있는 콘텐츠로 이동할 수 있는 부분을 추가하였습니다. 해당 콘텐츠를 선택하면 레이블 업적의 수행과 동일하게 Action_Key에 따라 필요한 콘텐츠에 맞는 UI로 이동합니다. 해당 부분 또한 Rua Script를 통해 이동합니다.

Slog 작업 (Server(Golang), DB(MySQL), 운영툴(Vue.js))

- Slog는 서비스 사인 넷마블에서 게임을 운영하기 위해 관리하는 Log지표입니다. 각 콘텐츠를 진행하거나 결과가 나왔을 시, 재화, 아이템 획득과 소모 등등 로그가 필요한 상황일 때 서버를 통해서 로그를 보내줍니다. Terra Origin이시즌2로 들어가면서 새로운 콘텐츠에 대한 SLog지표 작업을 담당해서 진행했습니다.

[illegible]

- 위에 보이는 이미지는 지표 페이지의 일부입니다. Log가 필요한 시기에 맞게 서버에 필요한 파라미터를 보내는 코드 작업을 하고, 해당 데이터를 JSON으로 만들어 넷마블로 보내게 됩니다.

유저 캐릭터

유저 아이템

플레이 정보

유저 로그 With Slog

길드 정보

길드 로그

우편 지급 (Send Post)

줄석/접속 이벤트

이벤트(NEW)

이벤트(OLD)

컨텐츠 관리

유저 관리

게임 운영

강제회수/복구

결제 정보 (Payment)

아이템 재화 레벨업 장비/레이븐 성장 장비 장착 룬 장착 룬 강화 특성 퀘스트 던전

User Info

PIDmyPIDWORLD30 / jpNAMEmyPIDb

CLASS공수UNIT SEQ105832LEVEL100HERO_IDX1248

[3-11] 게임 - 스킨 구매 x

USE

2019-04-15 00:00:00 ~ 2019-04-20 00:00:00 Search

DATE	HERO	LOG NAME	GET TYPE	TYPE	ITEM(IDX)	CNT
2019-04-15 12:17:52	사제 / LV.4	[5-100] 아이템 가차 상자	USE	상자	HR 환수 상자 (1880100)	1
2019-04-15 12:22:52	사제 / LV.6	[5-200] 아이템 레벨업(장비, 레이븐, 환수)	USE	레이븐	영력술사 슈퍼터 (1917180)	1
2019-04-15 14:34:09	사제 / LV.10	[5-200] 아이템 레벨업(장비, 레이븐, 환수)	USE	무기	초심자의 지팡이 (1621063)	1

- 개발 사에서도 Slog정보를 통제할 확인후 대응할 수 있도록 운영툴에 유저 로그 With Slog를 추가하는 작업을 진행했습니다. 서버에서 넷마블로 보내는 JSON을 만드는 동시에, DB에 Slog를 저장할 수 있도록 하였고, 아이템 ~던전까지 모든 콘텐츠에 대한 운영툴 확인 작업을 구현했습니다.

매칭 시스템 변경 작업 (Server(Golang) + DB(MySQL, Redis))

jp

jp

jp2

jp3

bvt_test_only

bot_only

스테이지 리스트

스테이지 명	TIME(sec)
스테이지 1	90
스테이지 2	90
스테이지 3	90
스테이지 4	90
스테이지 5	90

Save

Matching

Matching Block Matching Bot Time

release

release

release2

release3

스테이지 리스트

으로 설정됩니다.

스테이지 명, 상태	TIME(sec)
신원 제공 보물 상자_1	90
이동 보물 상자_1	90
영력술사의 지팡이_1	90
영력술사의 지팡이_1	90
영력술사의 지팡이_1	90

- Tera Origin의 매칭 관련 데이터는 Redis를 통해서 관리되고 있고, 던전 매칭을 시도할 때 일정 기간 이후에도 플레이어가 들어오지 않는다면 자동으로 봇을 매칭해주는 시스템이 기존에 있었습니다. 하지만, 그 후 요사항으로 월드별이 아닌 월드에 각 스테이지별로 봇을 설정할 수 있도록 수정하였습니다.
- 기존에 Redis구조는 Get, Set 구조로 되어 있어서, 월드별로만 저장하고 있었습니다. 우선, 기존에 있던 데이터들은 코드 상에서 동일한 구조로 Delete로 돌려 Redis에 저장되어 있는 데이터를 지우고, 기존 Get, Set 구조에서 Hget, Hset을 사용해 해시로 (RedisKey)_(Stageldx)로 각 데이터를 저장해 주어 스테이지 별로 상이하게 스테이지 봇 세팅을 할 수 있도록 수정하였습니다.

⇒ 크래프톤 스쿨에서 1년간 Client, Server, DB 분야에서 출시 전, 출시 후 서비스까지 100여개의 다양한 작업 및 콘텐츠 수정을 진행해왔고 위 기술한 내용은 그 중 주요한 내용을 정리하여 기술하였습니다.