

박 정 민

포트폴리오

Client Programmer



010-6419-1616 drpdum@gmail.com

CONTENTS

SNOWPIPE

- [MANAGERS 설계](#)
- [LocalDataManager](#)
- [NetworkManager](#)
- [GameDataManager](#)
- [LocalizationManager](#)
- [WindowManager](#)
- [FXManager](#)
- [SFXManager](#)
- [SceneManager, SceneController](#)
- [UserData](#)

NHN

- [C-OPS 에셋번들 분석 툴 제작 및 CI머신과 연동](#)
- [AAB, Automation Test 관련 내용 개발팀 프리젠테이션](#)
- [데이터 통합 관리 툴 제작 및 배포](#)
- [리소스 최적화](#)

YD ONLINE

- [UI 최적화](#)
- [Timeline을 이용하여 스토리 연출에 필요한 기능 구현](#)
- [해상도 대응](#)

NPICSOFT

- [피쳐드 대응을 위한 업적 & 리더보드 연동](#)

PLAYTOGAMES

- [PUN을 이용한 On/Offline mode 구현 및 매칭 시스템 적용](#)
- [FSM을 이용한 몬스터 AI 적용, Mecanim으로 동작 설계, PUN으로 동기화](#)

↓ [관련문서 모음](#)



SNOWPIPE

관련 문서 →

- ↓ [Managers.cs](#) 관리 구조체
- ↓ [Managers+Static.cs](#) 전역 접근 가능하게 Static변수 처리

MANAGERS 설계

- 산개되어 있는 여러 Singleton Manager Class들을 한 곳으로 모아 관리하기 편하게 해주는 매니저 관리 구조체
- Managers라는 하나의 접근 가능한 Singleton을 만들고, 각 매니저들을 Managers의 Static변수로 설정하여 전역접근이 가능하게 설계
- Coroutine을 이용하여 하나씩 필요한 순서로 각 매니저 초기화 진행

관련 문서 →

- ↓ [LocalDataManager.cs](#) 기본 매니저
- ↓ [LocalDataManager+GameOption.cs](#) Getter, Setter 정의

LocalDataManager

- 기기 내부에 저장하는 데이터들을 관리하는 매니저.
- EKey라는 Enum값으로 구분하여 저장하며, 매니저를 통해 기기 저장 남용 방지
- 작업자는 EKey정의와 필요한 Getter, Setter 정의 후 사용할 수 있게 설계



SNOWPIPE

관련 문서 →

- ↓ [NetworkManager.cs](#) 기본 매니저
- ↓ [NetworkProcess+Shelter.cs](#) 전역 접근 가능한 static 함수
- ↓ [Protocol_Shelter.cs](#) 툴로 만들어진 프로토콜 구조 스크립트

NetworkManager

- BestHTTP 에셋을 사용하여 서버와 연결 관리 매니저
- 데이터 툴을 이용하여 프로토콜 클래스 자동화
- 해당 프로토콜을 사용하는 static 함수를 정의해서 사용할 수 있게 설계

관련 문서 →

- ↓ [GameDataManager.cs](#) 기본 매니저
- ↓ [GameDataManager+Get.cs](#) 데이터 로드 로직.
- ↓ [CDataBase.cs](#) 자동화된 클래스로 생성을 위한 제네릭 생성 및 관리 클래스
- ↓ [CGoodsData.cs](#) 데이터툴로 자동 생성된 데이터 구조체
- ↓ [goods_data.PNG](#) 원본 데이터
- ↓ [enum_GoodsType.PNG](#) 원본 데이터에서 사용하는 enum table

GameDataManager

- 데이터 툴을 이용하여 암호화된 바이너리 파일에 저장된 데이터를 필요할 때 로드하여, 자동화 스크립트로 만들어진 구조에 넣어 관리하는 매니저



SNOWPIPE

관련 문서 →

- ↓ [LocalizationManager.cs](#) 기본 매니저
- ↓ [CLocalizationData.cs](#) 로컬라이징 데이터 구조체
- ↓ [LZ_Code.PNG](#) 원본 데이터

관련 문서 →

- ↓ [WindowManager.cs](#) 기본 매니저. 윈도우 기본 스택 관리
- ↓ [WindowManager+Popup.cs](#) 시스템 팝업 관리
- ↓ [WindowManager+Loading.cs](#) 로딩 페이지 관리
- ↓ [WindowBase.cs](#) 기본 WindowBase class. 내부 윈도우 스택 관리
- ↓ [NavigationBar.cs](#) Global Navigation Bar class

LocalizationManager

- 데이터 툴을 이용하여 암호화된 바이너리 파일에 저장된 로컬라이징 텍스트를 필요할 때 로드하여 관리하는 매니저
- [] 연산자 오버라이딩으로 개발 편의성 증대
- ex) var strBtnOk = Managers.LZ[COMMON_CONFIRM];

WindowManager

- UGUI관련 모든 관리 매니저
- UI Camera와 하나의 Canvas Root을 가지고 UI Depth관리를 위한 개별 루트 설정 및 관리
- 각 UI Page를 Window라 정의하고, Window의 기본 스택 관리와 Window 내부 Window의 스택 관리 시스템
- 공용 Popup, Global Navigation Bar, 로딩 페이지, Window 뒷배경 및 Block 처리, 네트워크 블락 처리 등 UI관련 처리 담당
- UI 기본 EventSystem 처리



SNOWPIPE

관련 문서 →

- ↓ [FXManager.cs](#) 기본 매니저. Effect별 Object Pool 관리
- ↓ [FXObject.cs](#) 개발 Effect Object class. 내부적으로 파티클, 애니메이션, 트윈 대응

FXManager

- Effect관련 관리 매니저
- 각 Effect별 Object Pool을 관리, Effect 필요하기 전에 예약 및 등록하고 씬 전환 시 풀 정리
- 개별 Effect에서 재생 종료 후 자동으로 Pool로 리턴
- Particle, Animation, Tween 지원

관련 문서 →

- ↓ [SFXManager.cs](#) 기본 매니저. 하나의 Object Pool로 관리
- ↓ [SFXObject.cs](#) 개별 Sound Effect Object class

SFXManager

- Sound Effect관련 관리 매니저
- 하나의 Object Pool에서 32개의 Sound Effect를 미리 생성해두고 관리



SNOWPIPE

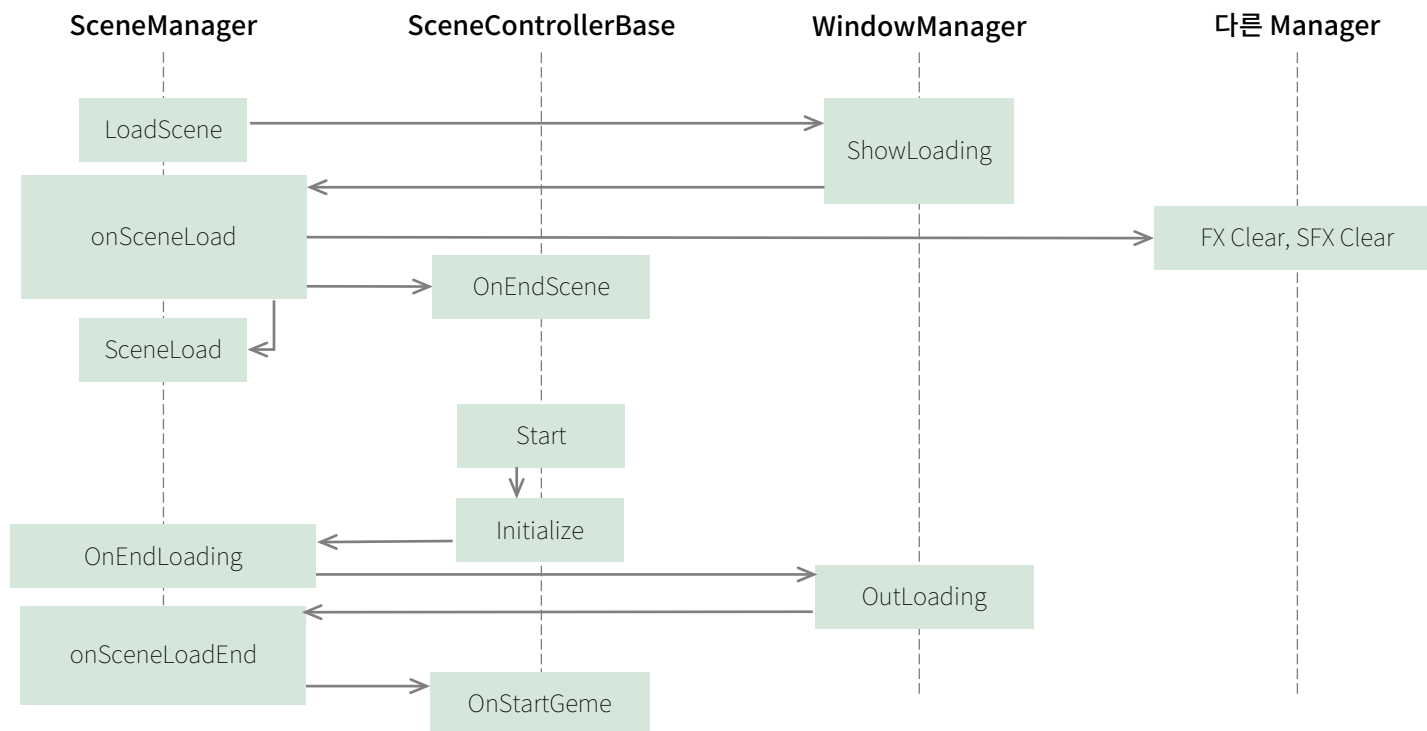
관련 문서 →

↓ [SceneManager.cs](#) 기본 매니저

↓ [SceneControllerBase.cs](#) 인게임에서 로드하는 씬에 메인 컨트롤러 베이스 클래스

SceneManager, SceneController

- Scene 전환 관리 매니저, 씬 메인 컨트롤러





SNOWPIPE

관련 문서 →

- ↓ [UserInfo.cs](#) Static Class
- ↓ [ActorInfo.cs](#) Actor관리 Class
- ↓ [CUserActor.cs](#) Actor UserData

UserData

- 서버에서 받아온 데이터와 게임 데이터를 결합하여 인게임에서 사용할 데이터 관리 클래스
- Static Class에 개별 관리 Class를 Static변수로 선언하여 전역적으로 접근 가능하게 설정
- 개별 관리 클래스에서 서버 정보와 게임 데이터를 결합하여 UserData를 생성하고, 인게임에서 사용



NHN - NStudio

관련 문서 →

- ↓ [ABA_CIScript.PNG](#) Assetbundle 분석 CI Script
- ↓ [ABA_report.py](#) 번들 분석 및 리포트 진행 Python Script

C-OPS 에셋번들 분석 툴 제작 및 CI머신과 연동

- CI머신에서 C-OPS 프로젝트 에셋번들 빌드
- 에셋번들 분석 툴을 이용하여 해당 에셋번들 참조 내역 검사
- 중복 참조 내용 리포팅
- 새벽 시간에 정기적으로 해당 분석툴 동작

관련 문서 →

- ↓ [PT_AAB.pdf](#) Android App Bundle 관련 내용 분석 및 프리젠테이션
- ↓ [PT_AutomationTest.pdf](#) Automation, SET 관련 내용 분석 및 프리젠테이션

AAB, Automation Test 관련 내용 개발팀 프리젠테이션

- Android App Bundle 관련 내용 분석 및 프리젠테이션
- Automation Test 관련 내용 분석 및 프리젠테이션



NHN – ALStudio

관련 문서 →

- ↓ [GameDataTool_Documentation.pdf](#) 툴가이드문서
- ↓ [CDataBase.cs](#) 존 분리, 그룹데이터 관리 등 데이터로드 관리 스크립트

데이터 통합 관리 툴 제작 및 배포

- 기획, 아트, 클라이언트, 서버, 스토리, 운영 등 모든 파트에서 사용 가능하게 외부 툴 제작, 접근성 쉽게 GUI로 제작
- Tardis에서 사용하는 Google Proto Buffer 대응 및 툴 연동
- 데이터 이진화 및 암호화로 속도와 보안 강화
- 데이터 관리의 자동화로 개발 시 구조 변경 용이
- 데이터 파편화를 통해 글로벌 데이터 대응 및 안정적으로 라이브 데이터 관리

관련 문서 →

- ↓ [텍스처 압축 관련 RnD 보고.pdf](#) ASTC 및 다른 압축 알고리즘 비교 보고 문서
- ↓ [Dooray_텍스처 규약.PNG](#) 아트 파트와 협의한 텍스처 규약 내용 공유 문서
- ↓ [Dooray_포인트 이벤트 리소스 관리 가이드.PNG](#) 라이브 이벤트 리소스 관리 가이드

리소스 최적화

- ASTC 알고리즘을 채택하여 텍스처 압축효율 증대
- UI, Tech 등 Art파트와 긴밀한 협업으로 최적화 진행
- 안정적인 라이브 서비스를 위한 Guideline 제안 및 적용



관련 문서 →

- ↓ [UI 최적화_방향성.pdf](#) 방향성 제시 문서
- ↓ [UI 최적화 가이드_Panel AtlasDepth 영역대 설정.pdf](#) Panel, Atlas 관련 영역대 설정 문서
- ↓ [UIOptimizationTools.cs](#) 필요 툴 스크립트

관련 문서 →

- ↓ [CTimeLineNGUIColorPlayable.cs](#) Timeline 전용 Behaviour 스크립트 (NGUI대응)
- ↓ [CTimeLineEventHandler.cs](#) Timeline Event Handler

관련 문서 →

- ↓ [UIBaseResolution.cs](#) 해상도 대응 부모 클래스

UI 최적화

- 클라팀 내부에 필요성 공유 및 방향성 제시
- 아트팀에게 필요성 공유 및 Panel, Atlas 영역대 가이드 제공
- 필요 툴 제공
- 리뉴얼 시기에 맞춰 Prefab 최적화 진행

Timeline을 이용하여 스토리 연출에 필요한 기능 구현

- 아트팀에서 정적인 Dialogue를 대체할 다른 연출의 필요성 제기
- 2017버전 Unity의 Timeline기능 이용 결정
- 관련 기능 분석 및 필요한 시스템 설계
- 필요 기능 구현

해상도 대응

- AOS, iOS 피쳐드 대응을 위한 해상도 대응
- 아트팀과 논의 후 NGUI Widget의 Anchor를 이용하기로 결정
- Anchor의 기준을 설정할 Pivot좌표 설정 설계



NPICSOFT

관련 문서 →

- ↓ [SocialManager.cs](#) AOS, IOS 업적, 리더보드관련Singleton
- ↓ [SocialGoogle.cs](#) Unity → 안드로이드 연결 Class
- ↓ [GPSManager.java](#) 안드로이드 내부처리 Singleton

피쳐드 대응을 위한 업적 & 리더보드 연동

- 넷마블 SDK에서 제공되는 구글 플레이 기능에서 업적과 리더보드 기능 충돌 이슈 발생
- 넷마블 SDK의 구글플레이 기능을 AOS 단에서 우회
- 추가적으로 Unity ↔ AOS 연동 로직 적용



PLAYTOGAMES

관련 문서 →

↓ [PhotonManager.cs](#) Photon Network 설정을 위한 Singleton

PUN 을 이용한 On/Offline mode 구현 및 매칭 시스템 적용

- 클라우드 네트워크를 사용하는 매칭 서버 필요
- Photon Unity Network 클라우드 시스템을 이용하기로 결정
- 접속 및 On/Offline mode 구현

관련 문서 →

↓ [AIControlTheDevil.cs](#) AI Controller, PUN 동기화

↓ [AnimatorAirPhase.PNG](#) Mecanim Animator

FSM을 이용한 몬스터 AI 적용, Mecanim 으로 동작 설계, PUN으로 동기화

- 지상과 공중을 오가는 거대한 악마 몬스터 AI설계
- 지상 Phase, 공중 Phase 구분, 해당 Phase별 패턴 설계
- 이동, 공격간 네트워크 동기화



관련문서 모음

SNOWPIPE

- ↓ [Managers.cs](#) 관리 구조체
- ↓ [Managers+Static.cs](#) 전역 접근 가능하게 Static변수 처리
- ↓ [LocalDataManager.cs](#) 기본 매니저
- ↓ [LocalDataManager+GameOption.cs](#) Getter, Setter 정의
- ↓ [NetworkManager.cs](#) 기본 매니저.
- ↓ [NetworkProcess+Shelter.cs](#) 전역 접근 가능한 static 함수
- ↓ [Protocol_Shelter.cs](#) 톨로 만들어진 프로토콜 구조 스크립트
- ↓ [GameDataManager.cs](#) 기본 매니저.
- ↓ [GameDataManager+Get.cs](#) 데이터 로드 로직
- ↓ [CDataBase.cs](#) 자동화된 클래스로 생성을 위한 제네릭 생성 및 관리 클래스
- ↓ [CGoodsData.cs](#) 데이터톨로 자동 생성된 데이터 구조체
- ↓ [goods_data.PNG](#) 원본 데이터
- ↓ [enum_GoodsType.PNG](#) 원본 데이터에서 사용하는 enum table
- ↓ [LocalizationManager.cs](#) 기본 매니저
- ↓ [CLocalizationData.cs](#) 로컬라이징 데이터 구조체
- ↓ [LZ_Code.PNG](#) 원본 데이터
- ↓ [WindowManager.cs](#) 기본 매니저. 윈도우 기본 스택 관리
- ↓ [WindowManager+Popup.cs](#) 시스템 팝업 관리
- ↓ [WindowManager+Loading.cs](#) 로딩 페이지 관리
- ↓ [WindowBase.cs](#) 기본 WindowBase class. 내부 윈도우 스택 관리
- ↓ [NavigationBar.cs](#) Global Navigation Bar class
- ↓ [FXManager.cs](#) 기본 매니저. Effect별 Object Pool 관리
- ↓ [FXObject.cs](#) 개발 Effect Object class. 내부적으로 파티클, 애니메이션, 트윈 대응
- ↓ [SFXManager.cs](#) 기본 매니저. 하나의 Object Pool로 관리
- ↓ [SFXObject.cs](#) 개별 Sound Effect Object class
- ↓ [SceneManager.cs](#) 기본 매니저
- ↓ [SceneControllerBase.cs](#) 인게임에서 로드하는 씬에 메인 컨트롤러 베이스 클래스
- ↓ [UserInfo.cs](#) Static Class
- ↓ [ActorInfo.cs](#) Actor관리 Class
- ↓ [CUserActor.cs](#) Actor UserData

NHN - NStudio

- ↓ [ABA_CIScript.PNG](#) Assetbundle 분석 CI Script
- ↓ [ABA_report.py](#) 번들 분석 및 리포트 진행 Python Script
- ↓ [PT_AAB.pdf](#) Android App Bundle 관련 내용 분석 및 프리젠테이션
- ↓ [PT_AutomationTest.pdf](#) Automation, SET 관련 내용 분석 및 프리젠테이션

NHN - ALStudio

- ↓ [GameDataTool_Documentation.pdf](#) 톨가이드문서
- ↓ [CDataBase.cs](#) 존 분리, 그룹 데이터 관리 등 데이터로드 관리 스크립트
- ↓ [텍스처 압축 관련 RnD 보고.pdf](#) ASTC 및 다른 압축 알고리즘 비교 보고 문서
- ↓ [Dooray_텍스처 규약.PNG](#) 아트 파트와 협의한 텍스처 규약 내용 공유 문서
- ↓ [Dooray_포인트 이벤트 리소스 관리 가이드.PNG](#) 라이브 이벤트 리소스 관리 가이드

YD ONLINE

- ↓ [UI 최적화 방향성.pdf](#) 방향성 제시 문서
- ↓ [UI 최적화 가이드_Panel AtlasDepth 영역대 설정.pdf](#) Panel, Atlas 관련 영역대 설정 문서
- ↓ [UIOptimizationTools.cs](#) 필요 톨 스크립트
- ↓ [CTimeLineNGUIColorPlayable.cs](#) Timeline 전용 Behaviour 스크립트(NGUI대응)
- ↓ [CTimeLineEventHandler.cs](#) Timeline Event Handler
- ↓ [UIBaseResolution.cs](#) 해상도 대응 부모 클래스

NPICSOIFT

- ↓ [SocialManager.cs](#) AOS, IOS 업적, 리더보드관련Singleton
- ↓ [SocialGoogle.cs](#) Unity → 안드로이드 연결 Class
- ↓ [GPSManager.java](#) 안드로이드 내부처리 Singleton

PLAYTOGAMES

- ↓ [PhotonManager.cs](#) Photon Network 설정을 위한 Singleton
- ↓ [AIControlTheDevil.cs](#) AI Controller, PUN 동기화
- ↓ [AnimatorAirPhase.PNG](#) Mecanim Animator

포트폴리오 GitHub 주소