

# Regression

## 데이터 관리 기술문서

리그레션 관련 영상

<https://youtu.be/Kw39dbMRWgc>

파트 1. 데이터 및 퀘스트 관리  
파트 2. 마을 Scene

프로그래밍 16기 팀 리그레션

이도규



# 목차

## 개요

게임 개요

팀 내역

게임 개요

## 구현 내역

데이터 통합 InfoDB

건물 인터페이스

퀘스트 매니저

## 스크립트 관리

데이터 흐름도

공용 변수 내역

스크립트 내역

CSV 파일 내역

## 부록

개발 의의

# 게임 개요

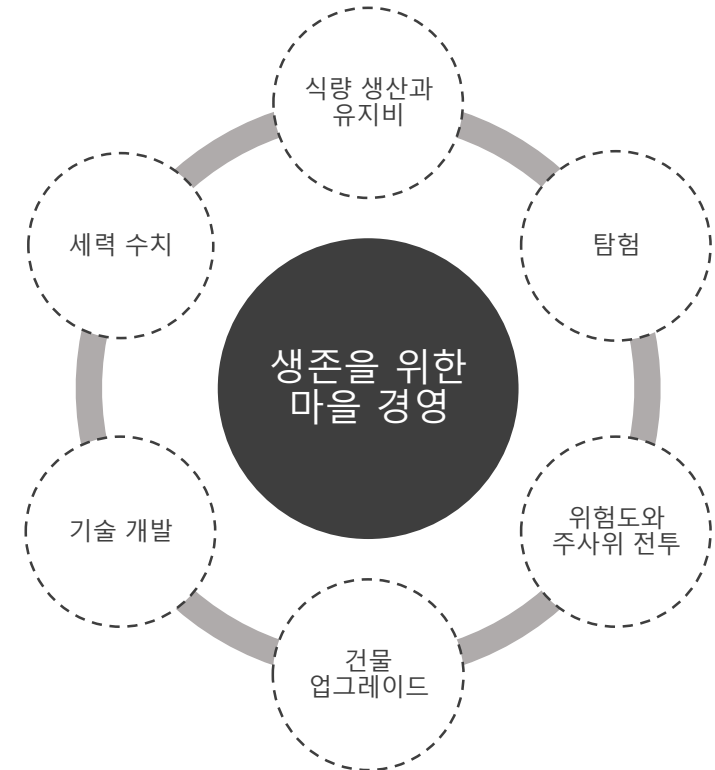


**게임 이름** 리그레션(Regression)

**플랫폼** 모바일(Android)

**게임 장르** 턴제 전략 게임- 생존, 마을 경영

**시점** 아이소메트릭(Isometric)

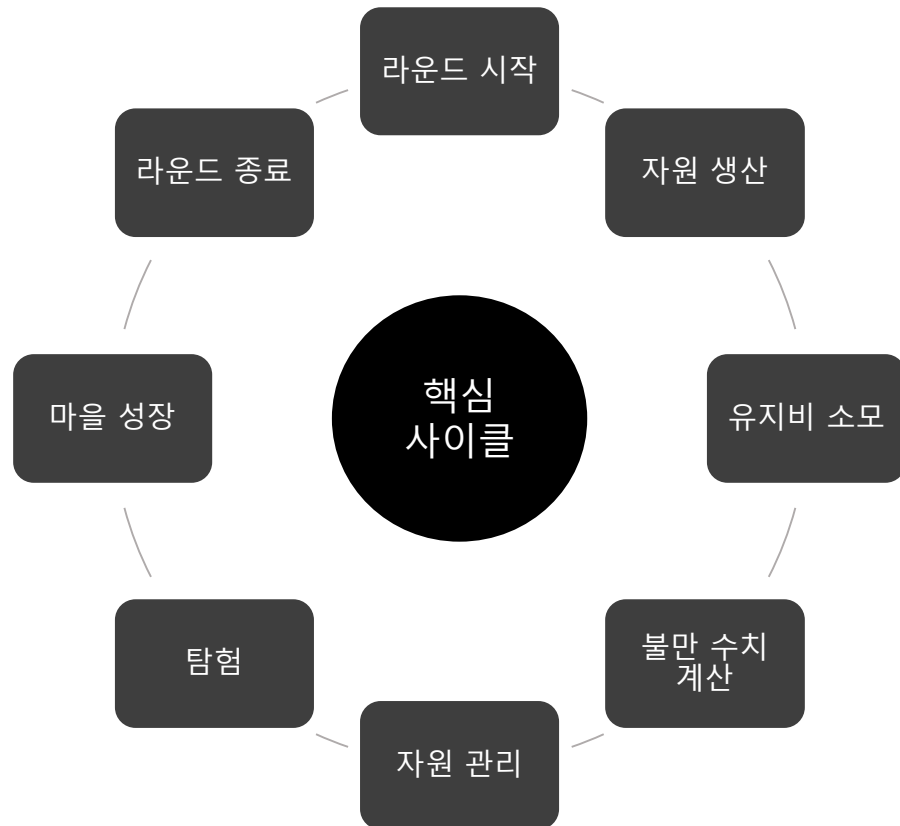


# 게임 개요

## 중심 테마

1. 포스트 아포칼립스 세상에서의 마을 생존

2. 지도자로서 마을을 경영하고 성장시키는 턴제 전략 게임



- 멀티가 아닌 솔로 플레이
- 플레이어는 월드를 돌아다니며 부족한 자원을 파밍
- 파밍한 자원을 바탕으로 마을 성장 (건물 업그레이드, 기술 개발)
- 매 라운드마다 적 객체가 생성되어 마을로 향함
- 적 세력의 공격으로부터 마을을 디펜스



# 게임 개요

경영을 관리하는 곳

마을을 발전시켜 생존에 성공하라



Title



LeaderChoice

탐험을 진행하는 곳

적을 물리치고 자원을 약탈하여 게임에 승리하라

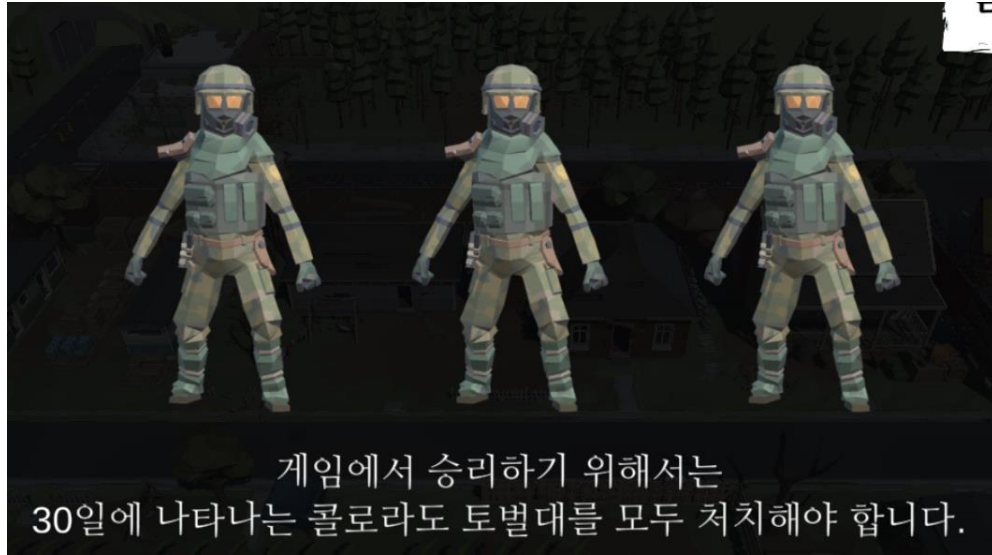


Town

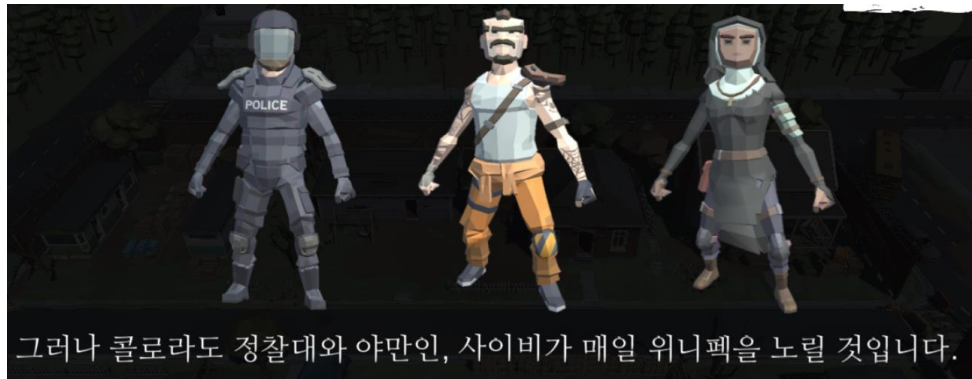


World

# 게임 개요



## 승리조건



## 패배조건

## 팀 내역

본 기술 문서는 데이터 관리와 관련된 기술 문서임을 참조 부탁드립니다.



**PD**

나진태



**PM**

안중재



**시스템**

방건배



**콘텐츠**

김서연



**QA**

이호수



**개발 팀장**

신수빈



**게임 기능**

문영빈



**데이터 관리**

이도규



**인터페이스**

정종찬

## 구현 내역

데이터 통합 InfoDB

건물 인터페이스

퀘스트 매니저



# 구현 내역

※ 본 문서는  
World Scene에 대한 내역을  
포함하고 있지 않습니다.



관리 데이터 내역

마을과의 상호작용



퀘스트 매니저



마을 SCENE

# 구현 내역 - 데이터 통합



InfoDB를 이용한 데이터 통합

Scene 전환에 따른  
데이터 유지

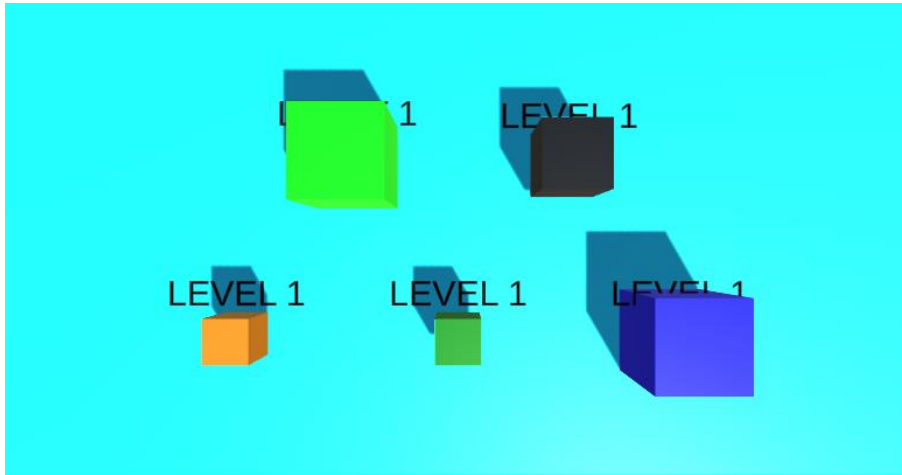
실시간 데이터 전환



**<인원 정보>**  
현재인원: 10  
농장 배치 인원: 6  
잉여 인원: 4  
최대 수용 인원: 14



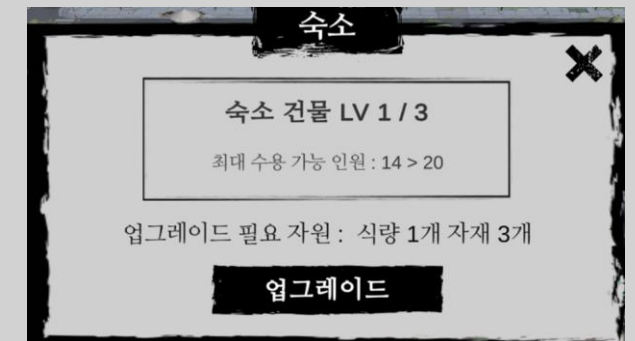
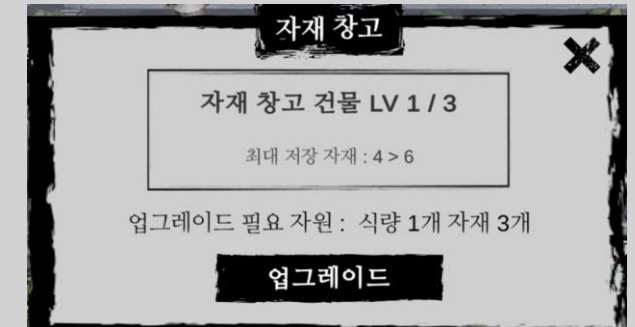
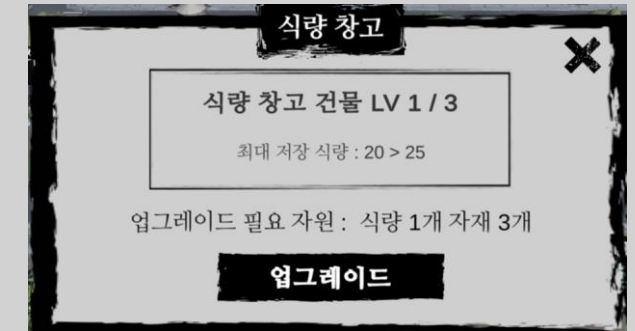
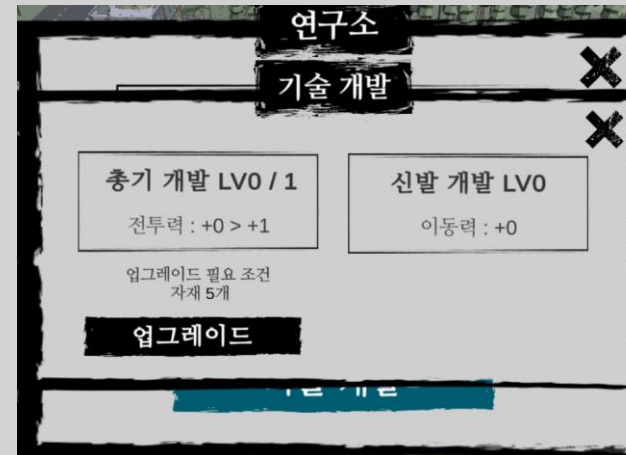
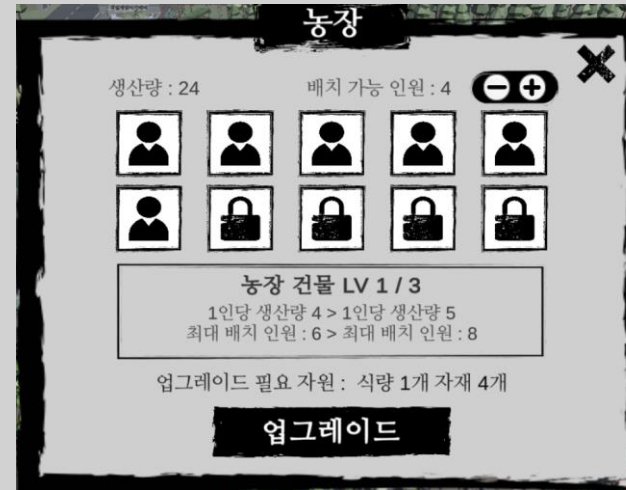
# 구현 내역 - 마을 씬



마을 씬  
제작



오브젝트를 통한  
팝업 생성



# 구현 내역 - 퀘스트 매니저

## 지도자의 사명



20XX년 09월 01일,

위니펙에는 이제 열 명이나 되는 사람들이 모였다. 이제 마을이라 불릴 만큼의 인원을 모으고 터전을 가꾼 것이다. 그럼에도 마약에 찌들어 식인을 즐기는 야만인들과 보이는 사람마다 마녀사냥을 하는 사이비 집단들로부터 살아남을 힘이 아직은 부족하다.

사실 가장 큰 문제는 지금 당장 먹을 식량이 부족한 것이다. 며칠을 굶주리다가 얼마 전 우연히 발견한 식량이 아니었다면 지금쯤 모두들 불만이 쌓여 뿔뿔이 흩어졌을 것이다. 그리고 굶어 죽기도 전에 다른 세력들에 의해 죽었을 것이다. 이 지독한 세상에서 혼자서 곧 죽음이다.

마을 사람들이 불만이 쌓여 흩어지기 전에 콜로라도 주변 지역에서 식량을 포함한 여러 자원을 찾아야 한다. 이를 바탕으로 사람들을 모으고 마을을 성장시켜서 공동체의 생존을 도모해야 한다. 이것이 위니펙 지도자로서의 사명이다.

## 이벤트

### 생존 방식

생존 방식에 대한 생각은 모두 제각각인 것 같다. 마을 내부에 지금보다 더 많은 지역을 약탈하고 심지어 사람까지도 잡아먹어야 한다는 집단과 더 강력한 종교의 힘으로 생존을 도모해야 한다는 집단 사이에서 갈등이 심화되고 있다. 지금도 우리는 살아남기 위해 여러 지역을 약탈하고 있지만 과연 어떤 방식으로 살아가야 우리가 생존할 수 있을까.

어느 한 집단을 지지해 줘야 할 것 같다. 더 많은 지역을 약탈하고 심지어 식인까지 해야 한다는 집단의 손을 들어줘야 할까.

야만도 : 20 사이비 : -5

수락

야만도 : -5 사이비 : 20

거절

## 메인 퀘스트



퀘스트 등장 조건 관리



## 서브 퀘스트



선택에 따라 실시간으로 변하는 수치





## 스크립트 관리

데이터 흐름도

공용 변수 내역

스크립트 내역

CSV 파일 내역

# 데이터 흐름도

Build : 5개의 건물 총괄

클래스

변수

ReadCSV

ReadText

LeaderDB

BuildDB

InitDB

GunDB

ShoesDB

EventDB

MainEventDB

LevelData

InfoDB

MouseEvent

FarmCtrl

LaboratoryCtrl

InnCtrl

FoodStroageCtrl

ComponentStorageCtrl

공용 변수

buildLevel  
buildState  
Gun  
shoes

Food  
Component  
Population

공용 변수

Party  
EnemyParty

Food  
Component  
Population

UIMgr

QuestManager

SceneManager

TileManager

Tile

RealGameManager

TradeManager

EnemyManager

Population

Food

Component

buildInfo

공용 변수

Quest

Dissatisfaction

Pseudo

AttackValue

Cost

MoveValue

ItemInfo

Round

MainQuest

Barbarism

ExploreInfo

# InfoDB에서 사용되는 공용 변수 내역 [GAME PLAY]

## 게임 플레이에 따라 변하는 플레이어 및 기타 변수

### Population

currentPopulation	현재 인원
availablePopulation	남은 인원
maxPopulation	최대 인원

### Component

currentFood	현재 식량
maxFood	최대 식량
productionFood	식량 생산량

### Enemy

attackValue	적 공격력
moveValue	적 이동력

### Food

currentFood	현재 식량
maxFood	최대 식량
productionFood	식량 생산량

### EstimatedCost

estimatedCost	유지비
addEstimatedCost	추가 유지비

### RatioInfo

disRate	불만
maxDisRate	최대 불만
barRate	야만
matBarRate	최대 야만
pseRate	사이비
maxPesRate	최대 사이비

### 지도자 변수

### Leader

attackValue	리더 공격력
moveValue	리더 이동력

### 적 및 외부 변수

# InfoDB에서 사용되는 공용 변수 내역 [COMMON]

## 게임 플레이에 따라 변하는 건물 변수

### Farm

maxLevel	최대 레벨
currentLevel	레벨
currentDeployPop	배치 인원
maxDeployEarmPop	배치 최대 인원
foodPerPopValue	1인 생산량
resultFoodProduction	최종 생산량
spendPopulation	레벨 업 인원
spendFood	레벨 업 식량
spendComponents	레벨 업 자재

### Forge

maxLevel	최대 레벨
currentLevel	레벨
gunDevelopmentLv	총기 최대 레벨
shoesDevelopmentLv	신발 최대 레벨
spendPopulation	레벨 업 인원
spendFood	레벨 업 식량
spendComponents	레벨 업 자재

### GUN

Level	레벨
spendPopulation	레벨 업 인원
spendFood	레벨 업 식량
spendComponents	레벨 업 자재
attack	공격력

### Shoes

Level	레벨
spendPopulation	레벨 업 인원
spendFood	레벨 업 식량
spendComponents	레벨 업 자재
moveMents	공격력

### Inn

maxLevel	최대 레벨
currentLevel	레벨
spendPopulation	레벨 업 인원
spendFood	레벨 업 식량
spendComponents	레벨 업 자재

### FoodStorage

maxLevel	최대 레벨
currentLevel	레벨
spendPopulation	레벨 업 인원
spendFood	레벨 업 식량
spendComponents	레벨 업 자재

### ComponentStorage

maxLevel	최대 레벨
currentLevel	레벨
spendPopulation	레벨 업 인원
spendFood	레벨 업 식량
spendComponents	레벨 업 자재



# InfoDB에서 사용되는 공용 변수 내역 [DB]

데이터 베이스 저장 변수 내역 ( 고정 된 값을 가진 변수들 모음)

## FarmDB

level	레벨
population	레벨 업 인원
food	레벨 업 식량
compoment	레벨 업 자재
maxPopulation	최대 배치 인원
foodPerson	1인 생산량

## LaboratoryDB

level	레벨
population	레벨 업 인원
food	레벨 업 식량
compoment	레벨 업 자재
cost	유지비
gunMaxLevel	총기 최대 레벨
shoesMaxLevel	신발 최대 레벨

## GunDB

level	레벨
population	레벨 업 인원
food	레벨 업 식량
compoment	레벨 업 자재
attack	공격력

## ShoesDB

level	레벨
population	레벨 업 인원
food	레벨 업 식량
compoment	레벨 업 자재
movement	이동력

## InnoDB

level	레벨
population	레벨 업 인원
food	레벨 업 식량
compoment	레벨 업 자재
maxPopulation	최대 인원

## FoodStorageDB

level	레벨
population	레벨 업 인원
food	레벨 업 식량
compoment	레벨 업 자재
maxFood	최대 식량

## CompomentSotrageDB

level	레벨
population	레벨 업 인원
food	레벨 업 식량
Compoment	레벨 업 자재
maxComponent	최대 지제

# Scene 스크립트 내역

## TitleScene Scene

LeaderChoiceSceneM

LeaderScene 전환 스크립트

## LeaderChoice Scene

InfoDB

초기 데이터 및 외부 데이터베이스 가져오기, 씬 전환의 영향을 받지 않는 저장 공간

## CommonUI Scene

UIMgr

InfoDB에서 읽어 들인 값을 공통 UI로 출력 스크립트

SceneCtrl

마을과 월드를 이어주는 씬 전환 스크립트

QuestManager

Quest에 대한 정보를 가져오는 스크립트

## Scene 스크립트 내역

### Town Scene

MouseEvent	씬 내에서 오브젝트와의 상호작용을 위한 스크립트
FarmCtrl	농장 UI 및 농장 콘텐츠 관련 총괄 스크립트
LaboratoryCtrl	연구소 UI 및 연구소 콘텐츠 관련 총괄 스크립트
InnCtrl	숙소 UI 및 숙소 콘텐츠 관련 총괄 스크립트
FoodStorageCtrl	식량 창고 UI 및 식량 창고 콘텐츠 관련 총괄 스크립트
ComponentStorageCtrl	자재 창고 UI 및 자재 창고 콘텐츠 관련 총괄 스크립트

## Scene 스크립트 내역

### World Scene

TileManager

타일 생성 매니저 스크립트

Tile

타일이 직접적으로 속성 값을 가지도록 해주는 스크립트

LevelCtrl

타일에 대한 속성 값을 부여하기 위한 스크립트

RealGameManager

전체적인 게임진행을 관할하는 게임 매니저

TradeManager

탐험대 제어 스크립트

EnemyManager

적 AI 제어 스크립트



## 부록

기타 기술코드

개발 의의

# CSV 파일 내역

```
public struct FarmDB
{
    public int level;
    public int population;
    public int food;
    public int component;
    public int maxPopulation;
    public int foodPerson;
}
```

FarmDB.CSV

LEVEL	Population	Food	Component	maxPopulation	foodPerson
1	0	0	0	6	3
2	0	0	5	8	4
3	0	5	10	10	5

```
public struct LaboratoryDB
{
    public int level;
    public int population;
    public int food;
    public int component;
    public int cost;
    public int gunMaxLevel;
    public int shoesMaxLevel;
}
```

LaboratoryDB.CSV

LEVEL	Population	Food	Component	Cost	gunMaxLevel	shoesMaxLevel
1	0	0	0	4	1	0
2	0	0	5	10	2	1
3	0	5	10	16	3	2

```
public struct GunDB
{
    public int level;
    public int population;
    public int food;
    public int component;
    public int attack;
}
```

GunDB.CSV

LEVEL	Population	Food	Component	Attack
0	0	0	0	0
1	0	0	4	1
2	0	0	8	3
3	0	5	15	6

```
public struct ShoesDB
{
    public int level;
    public int population;
    public int food;
    public int component;
    public int movement;
}
```

ShoesDB.CSV

LEVEL	Population	Food	Component	Movement
0	0	0	0	0
1	0	0	4	1
2	0	5	8	2

# 부록 - 기술 코드

```
public struct InnDB
{
    public int level;
    public int population;
    public int food;
    public int component;
    public int maxPopulation;
}
```

```
public struct FoodStorageDB
{
    public int level;
    public int population;
    public int food;
    public int component;
    public int maxFood;
}
```

```
public struct ComponentStorageDB
{
    public int level;
    public int population;
    public int food;
    public int component;
    public int maxComponent;
}
```

InnDB.CSV

LEVEL	Population	Food	Component	savePopulation
1	0	0	0	12
2	0	0	4	20
3	0	4	8	36

FoodStorageDB.CSV

LEVEL	Population	Food	Component	saveFood
1	0	0	0	10
2	0	0	4	15
3	0	4	8	20

ComponentStorageDB.CSV

LEVEL	Population	Food	Component	saveComponent
1	0	0	0	10
2	0	0	4	15
3	0	4	8	20

건물의 양이 제한적이기에  
3가지로 정규화

# 부록 - 기술 코드

```
public struct MainQuestDB
{
    // 이벤트 내용
    public int id;
    public string title;
    public string content1;
    public string content2;
    public string content3;
    public string content4;
    public string content5;
    public string content6;

    public int round;
}
```

줄 바꿈으로 내용을  
깔끔하게 표현하기  
위해 콘텐츠를 나눔

```
public struct QuestDB
{
    // 이벤트 내용
    public int id;
    public string title;
    public string content1;
    public string content2;
    public string content3;
    public string content4;
    public string content5;
    public string content6;
    public bool agreeCheck;

    // 수락시 증감 데이터
    public int agreepopulation;
    public int agreefood;
    public int agreecomponents;
    public int agreedisRate;
    public int agreebarRate;
    public int agreepseRate;

    // 실패시 증감 데이터
    public int degreepopulation;
    public int degreefood;
    public int degreecomponents;
    public int degreedisRate;
    public int degreebarRate;
    public int degreepseRate;

    public int round; // 라운드
    public int roundLimit; // 라운드 증감

    public bool isCheck; // 중복 체크
}
```

DB 사용 변수



# 부록 - 기타 소스코드 [DB 사용 내역 소스 코드]

```
InfoDB.cs* X
기타 파일 InfoDB

1073 // 건물 관련 함수
1074 public void FarmUpgrade()...
1091
1092 public void LaboratoryUpgrade()...
1114
1115 public void GunUpgrade()...
1132
1133 public void ShoesUpgrade()...
1151
1152 public void InnUpgrade()...
1169
1170 public void FoodStorageUpgrade()...
1187
1188 public void ComponentStorageUpgrade()...
1206
```

InfoDB에 공용으로 사용 될 함수 통합

```
void ReadComponentStorage()
{
    List<Dictionary<string, object>> data = CSVReader.Read("db/ComponentStorage");

    for (var i = 0; i < data.Count; i++)
    {
        componentStorageDB[i].level = int.Parse(data[i]["LEVEL"].ToString());
        componentStorageDB[i].population = int.Parse(data[i]["Population"].ToString());
        componentStorageDB[i].food = int.Parse(data[i]["Food"].ToString());
        componentStorageDB[i].component = int.Parse(data[i]["Component"].ToString());
        componentStorageDB[i].maxComponent = int.Parse(data[i]["saveComponent"].ToString());
    }

    componentStorage.maxComponetsStorageLevel = data.Count;
    componentStorage.componetsStorageCurrentLevel = 1;
    componentStorage.spendPopulation = componentStorageDB[componentStorage.componetsStorageCurrentLevel].population;
    componentStorage.spendFood = componentStorageDB[componentStorage.componetsStorageCurrentLevel].food;
    componentStorage.spendComponents = componentStorageDB[componentStorage.componetsStorageCurrentLevel].component;
}
```

## CSV READ 및 데이터 초기화

```
public void ComponentStorageUpgrade()
{
    // 소비되는 자원 내역
    population.currentPopulation -= componentStorage.spendPopulation;
    component.currentComponent -= componentStorage.spendComponents;
    food.currentFood -= componentStorage.spendFood;

    // 건물 업그레이드
    componentStorage.componetsStorageCurrentLevel++;

    // 건물업그레이드
    component.maxComponent = componentStorageDB[componentStorage.componetsStorageCurrentLevel - 1].maxComponent;

    // 업그레이드 필요 자재
    componentStorage.spendPopulation = componentStorageDB[componentStorage.componetsStorageCurrentLevel].population;
    componentStorage.spendFood = componentStorageDB[componentStorage.componetsStorageCurrentLevel].food;
    componentStorage.spendComponents = componentStorageDB[componentStorage.componetsStorageCurrentLevel].component;
}
```

배열을 이용한 간단한 관리 소스코드

## 부록 - 기타 소스코드 [건물 업그레이드]

```
public class FoodStorageCtrl : MonoBehaviour
{
    // 건물 공통 오브젝트 변수
    Image panel;
    TextMeshProUGUI textLogo;           // 로고 텍스트
    TextMeshProUGUI textInfoLevel;      // 건물 레벨 텍스트
    TextMeshProUGUI textInfoState;      // 건물 상태 텍스트
    TextMeshProUGUI textUpgradeInfo;    // 건물 업그레이드 조건 텍스트
    Button buttonUpgrade;               // 버튼 업그레이드
    TextMeshProUGUI textFinal;          // 완료 조건 텍스트
    Button buttonExit;

    private TextMesh textObjFoodStorage;
    // 오브젝트 관련
    // 오브젝트 연결 함수
    void SetComponentInit()...
    // 텍스트 관련 함수
    string msgBuildInfo;               // 농장 건물
    string msgUpgradeState;             // 업그레이드 필요 자원
    string msgPerson;                  // 1인당 생산량
    string msgUpgradeFail;

    // 텍스트 관련
    // 고정 텍스트 내용 모음집
    void DefaultTextCollection()...
    // 업그레이드 필요 텍스트
    string UpgradeNeedResource()...
```

```
public class ComponentStorageCtrl : MonoBehaviour
{
    // 건물 공통 오브젝트 변수
    Image panel;
    TextMeshProUGUI textLogo;           // 로고 텍스트
    TextMeshProUGUI textInfoLevel;      // 건물 레벨 텍스트
    TextMeshProUGUI textInfoState;      // 건물 상태 텍스트
    TextMeshProUGUI textUpgradeInfo;    // 건물 업그레이드 조건 텍스트
    Button buttonUpgrade;               // 버튼 업그레이드
    TextMeshProUGUI textFinal;          // 완료 조건 텍스트
    Button buttonExit;

    private TextMesh textObjResourceStorage;

    // 오브젝트 관련
    // 오브젝트 연결 함수
    void SetComponentInit()...
    // 텍스트 관련 함수
    string msgBuildInfo;               // 농장 건물
    string msgUpgradeState;             // 업그레이드 필요 자원
    string msgPerson;                  // 1인당 생산량
    string msgUpgradeFail;

    // 텍스트 관련
    // 고정 텍스트 내용 모음집
    void DefaultTextCollection()...
    // 업그레이드 필요 텍스트
    string UpgradeNeedResource()...
```

```
public class InnCtrl : MonoBehaviour
{
    // 건물 공통 오브젝트 변수
    Image panel;
    TextMeshProUGUI textLogo;           // 로고 텍스트
    TextMeshProUGUI textInfoLevel;      // 건물 레벨 텍스트
    TextMeshProUGUI textInfoState;      // 건물 상태 텍스트
    TextMeshProUGUI textUpgradeInfo;    // 건물 업그레이드 조건 텍스트
    Button buttonUpgrade;               // 버튼 업그레이드
    TextMeshProUGUI textFinal;          // 완료 조건 텍스트
    Button buttonExit;

    private TextMesh textObjINN;

    // 오브젝트 관련
    // 오브젝트 연결 함수
    void SetComponentInit()...

    // 텍스트 관련 함수
    string msgBuildInfo;               // 농장 건물
    string msgUpgradeState;             // 업그레이드 필요 자원
    string msgPerson;                  // 1인당 생산량
    string msgUpgradeFail;

    // 텍스트 관련
    // 고정 텍스트 내용 모음집
    void DefaultTextCollection()...
    // 업그레이드 필요 텍스트
    string UpgradeNeedResource()...
```

같은 구조, 다른 기능을 하고 있는 함수들

추가적인 변수 내역은 따로 관리 가능

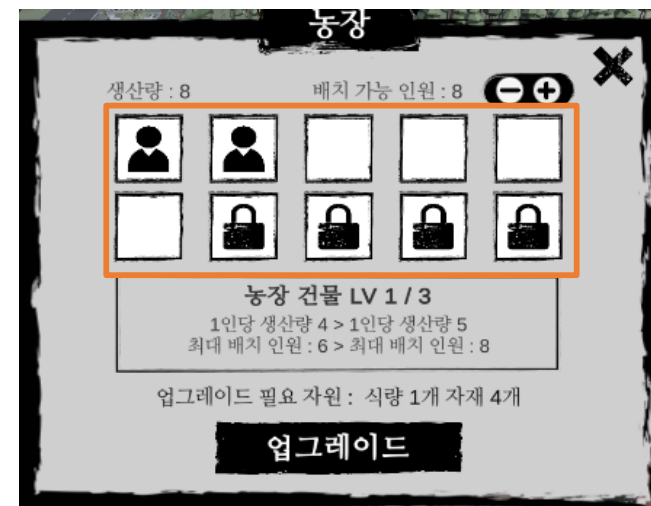
# 부록 - 기타 소스코드 [기타]

```
// 오브젝트 관련
// 오브젝트 연결 함수
void SetComponentInit()
{
    // 공통 컴포넌트 내용
    panel = transform.GetChild(0).GetComponent<Image>();
    textLogo = transform.GetChild(1).GetComponent<TextMeshProUGUI>();
    textInfoLevel = transform.GetChild(2).GetChild(0).GetComponent<TextMeshProUGUI>();
    textInfoState = transform.GetChild(2).GetChild(1).GetComponent<TextMeshProUGUI>();
    textUpgradeInfo = transform.GetChild(3).GetComponent<TextMeshProUGUI>();
    buttonUpgrade = transform.GetChild(4).GetComponent<Button>();
    textFinal = transform.GetChild(5).GetComponent<TextMeshProUGUI>();
    buttonExit = transform.GetChild(6).GetComponent<Button>();
    textFinal.gameObject.SetActive(false);
}
```

SVN 사용을 위한 컴포넌트 직접 연결

```
for (int i = 0; i < InfoDB.Instance.farm.currentDeployPop; i++)
{
    populationSpace[i].transform.GetChild(0).gameObject.SetActive(false);
    populationSpace[i].transform.GetChild(1).GetComponent<Image>().sprite = workImg;
}
for (int i = InfoDB.Instance.farm.currentDeployPop; i < InfoDB.Instance.farm.maxDeployEarmPop; i++)
{
    populationSpace[i].transform.GetChild(0).gameObject.SetActive(true);
    populationSpace[i].transform.GetChild(1).GetComponent<Image>().sprite = null;
}
for (int i = InfoDB.Instance.farm.maxDeployEarmPop; i < 10; i++)
{
    populationSpace[i].transform.GetChild(0).gameObject.SetActive(false);
    populationSpace[i].transform.GetChild(1).GetComponent<Image>().sprite = lockImg;
}
```

실시간 이미지 출력의 변화를 위한 내역



# 개발 의의

추상화 (컴포넌트 지향 방식)을 효율적으로 활용하기 위한 방안 이해

팀 프로젝트로써 협업을 위한 시스템을 조성

개별적인 작업을 하더라도 다른 코드에 큰 영향을 미치지 못하기 위한 방안을 모색 (싱글톤 -> DB로 사용)

개발팀 외, 기획팀에서도 간단히 프로그램을 관리할 수 있는 방안을 탐색

협업에 필요한 대체적인 내역들을 체험 (무엇이 필요한지, 무엇이 중요한지를 알아가는 과정의 중요성)

함께하는 즐거움 기쁨을 느낄 수 있었던 좋은 프로젝트였습니다.

**감사합니다.**