팀 프로젝트

Stay Alive



작성자: 채용운

작성일: 2019년 12월 03일

CONTENTS

- 프로젝트 설명
- Ⅲ 프로젝트 역할분담
- Ⅲ 프로젝트 개발내용
- IV 프로젝트 설계
- ∨ 최적화 작업

I 프로젝트 설명



게임 명: Stay Alive

게임 장르: 3D 생존 RPG

사용언어: C#

플랫폼: PC-Windows /

Mobile-Android

개발 기간: 19.09.02~19.11.07 (2개월)

역할 구성원:

- **1) 박현정** (기획 총괄, 서브개발)
- 2) 채용운 (개발 총괄,애니메이션 메카니즘)
- 3) **이나현** (서브 개발)
- 4) 이시영 (모델링, UI)

박현정

- 게임 기획
- 시간 시스템, Map, Enemy,Player, 제작 시스템
- 코드 작성
- -Enemy(Script)

이나현

- 코드작성
- Defends_enemy, Dragon, Egg (Script)
- 보고서 작성

채용운

- 코드작성
- Player, Turret, Game_Manager, Animal,Ending_Manager ...etc(Script)
- 애니메이션 메카니즘

이시영

- 모델링
- Player Character
- 맵제작
- Terrain, 오브젝트 배치



프로젝트 개발내용



데이터 불러오기



사냥 및 채집



캐릭터 크기 커스터마이징



인벤토리 및 제작



프로젝트 개발내용



장비 아이템 장착



밤(디펜스모드)

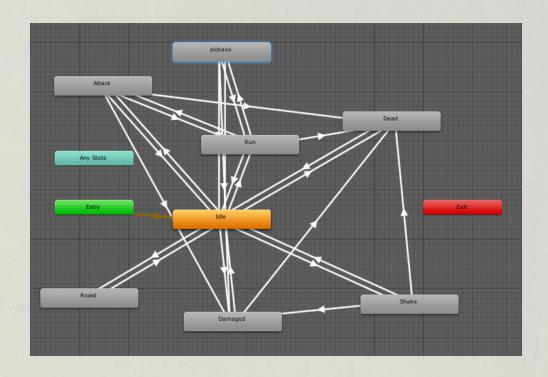


집(거주지) 시스템

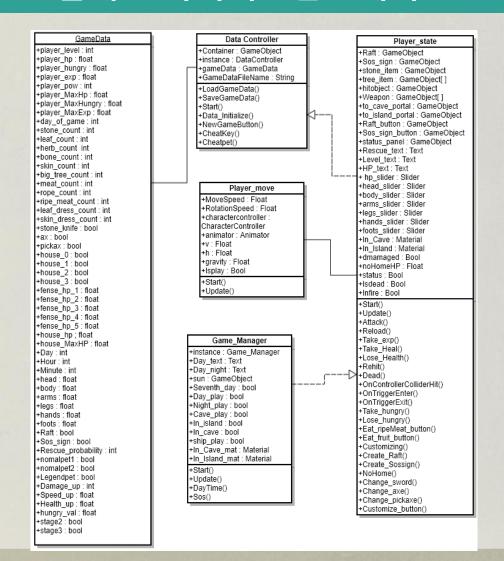


동굴(펫 획득 스테이지)

- 애니메이션 메카니즘
- 스크립트로부터 파라미터 값을 전달받아 상태에 맞는 애니메이션을 실행함



클래스 다이어그램 - 베이스



클래스 다이어그램 - 베이스

Data Controller

+Container : GameObject

+instance : DataController

+GameDataFileName : String

+gameData : GameData

+LoadGameData()

+SaveGameData()

+Start()

+Update()

+Sos()

+DayTime()

Data Controller - 게임 데이터 저장 <u>및 불러오기</u> 담당

GameData - 플레이어 및 시간 시스템에 사용되는 데이터들을 보관

+player_level : int +player_hp : float √ ıvıaxHp : float +player_MaxHungry : float +player MaxExp: float +day_of_game:int +stone_count : int +leaf count:int +herb count int +bone count:int +skin count:int +big tree count:int +meat_count : int +rope_count : int +ripe meat count:int +leaf dress count:int +skin_dress_count : int +stone knife : bool +ax : bool +pickax : bool +house_0 : bool +house_1: bool +house_2 : bool +house_3: bool +fense_hp_1 : float +fense hp 2:float +fense_hp_3: float +fense_hp_4 : float +fense_hp_5 : float +house_hp; float +house_MaxHP : float +Day: int +Hour : int +Minute : int +head : float -body : float +arms : float +legs : float +hands : float +foots : float +Raft : bool +Sos_sign : bool +Rescue_probability : int +nomalpet1 : bool +nomalpet2 : bool +Legendpet: bool -Damage_up : int +Speed_up : float

+Health up:float

+hungry_val : float +stage2 : bool +stage3 : bool

GameData

+Data_Initialize() +NewGameButton() +CheatKev() +Cheatpet() Player move +MoveSpeed : Float +RotationSpeed : Float +charactercontroller: CharacterController +animator : Animator +v : Float +h : Float +gravity : Float +Isplay : Bool +Start() +Update() Game Manager +instance : Game Manager +Day text : Text +Day_night : Text +sun : GameObject +Seventh_day : bool +Day_play: bool +Night_play: bool +Cave_play : bool +In_island : bool +In cave : bool +ship play: bool +In Cave mat: Material +In_Island_mat : Material +Start()

+body_slider : Slider alidar - Olidar +hands_siic. +foots_slider : Slider +In_Cave : Material +In_Island : Material +dmamaged : Bool +noHomeHP : Float +status : Bool +Isdead : Bool +Infire : Bool +Start() +Update() +Attack() +Reload() +Take exp() +Take_Heal() +Lose_Health() -Dahit() +Onco... +OnTriggerEnter +OnTriggerExit() +Take_hungry() +Lose_hungry() +Eat ripeMeat button() +Eat fruit button() +Customizing() +Create_Raft() +Create_Sossign() +NoHome() +Change_sword() +Change_axe() +Change_pickaxe() +Customize_button()

Player_state

+Raft : GameObject

+Sos_sign : GameObject

+stone_item : GameObject

+tree_item : GameObject[]

+hitobiect : GameObiect

+Weapon : GameObject[]

+Raft button : GameObject

+status_panel : GameObject

+Rescue_text : Text

+Level_text : Text

+ hp_slider : Slider

+head slider : Slider

+HP_text: Text

+to_cave_portal : GameObject

+to island portal: GameObject

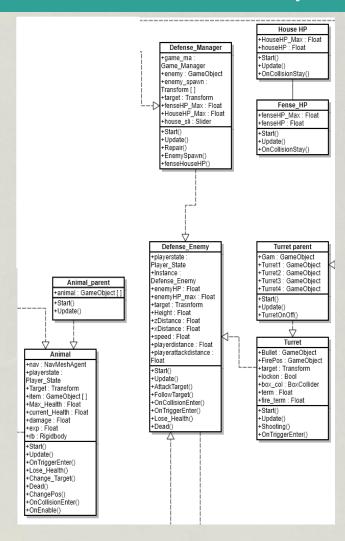
+Sos sign button : GameObject

Player_state - 공격,피격,커스터마 이징 등 캐릭터를 제 어한다

Player_move - 플레이어의 이동을 담당

Game_Manager - 시간시스템과 시간 에 따른 해의 위치를 관리

클래스 다이어그램 - 동물,디펜스,포탑

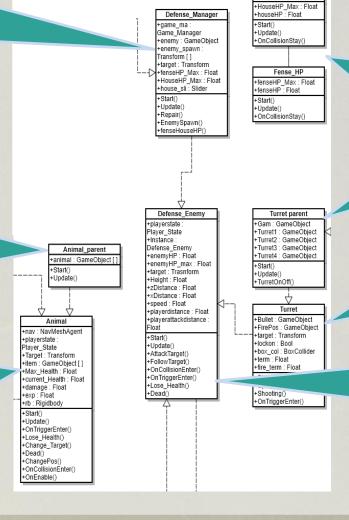


클래스 다이어그램 - 동물,디펜스,포탑

Defense_Manager - 디펜스 몬스터 생성 관리

Animal_parent - 동물 오브젝트 활 성화 관리

Animal - 공격,피격,이동 등 동물 오브젝트 제어



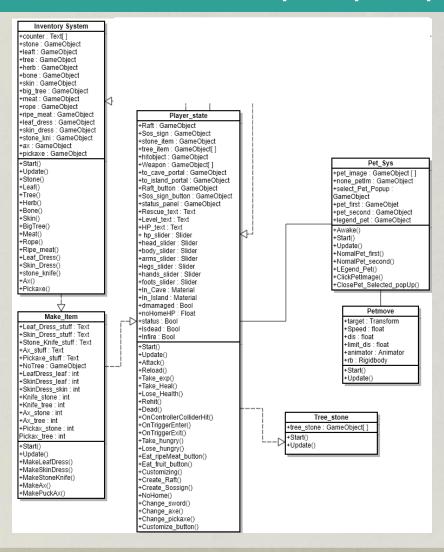
House_HP & Fense_HP - 집,울타리의 체력 관리

Turret_parent - 터렛 오브젝트 활 성화 관리

Turret - 디펜스 몬스터를 공격(자동타겟팅)

Defense_Enemy - 이동,공격,피격 등 디 펜스 몬스터 제어

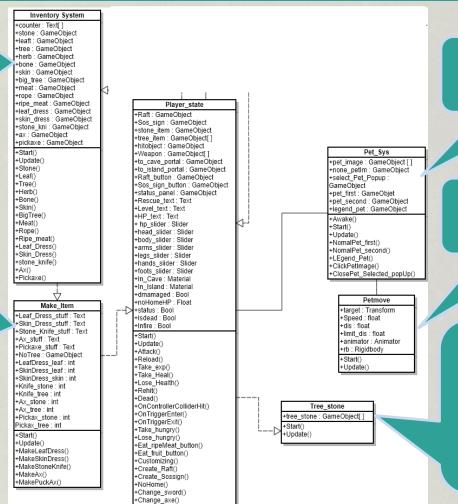
클래스 다이어그램 - 펫,채집,사냥,제작



<u>클래스 다이어그램 – 펫,채집,사냥,제작</u>

Inventory_System - 획득한 아이템 관리 및 UI표시

Make_Item
- 획득한 아이템으로 새로운 아이템 제작



+Change_pickaxe() +Customize_button() Pet_System - 획득한 펫을 관리

Pet_move - 펫의 이동관리

Tree_stone - 채집시 아이템을 생성하는 나무와 바 위 오브젝트 활성화 관리 CompareTag()

- 추가 메모리를 할당하고 복사하는 tag 프로퍼티 호출대신 CompareTag 사용

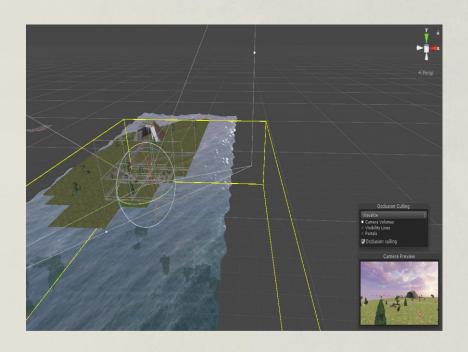
Static Batch

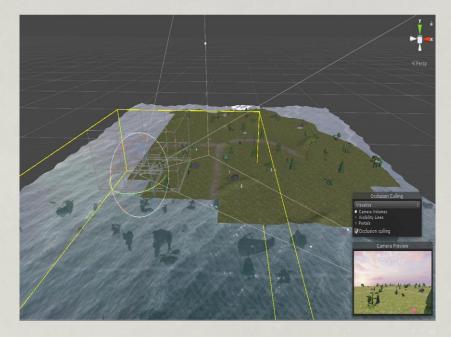
- 움직이지 않는 오브젝트들은 static으로 설정해서 배칭이 되도록함

• 오브젝트 풀링

- 주기적으로 생성,삭제가 필요한 enemy,tree,stone 오브젝트를 새로 생성,삭제하지않고 활성화,비활성화를 통해 제어

- 오클루전 컬링
- 카메라에 보이는 각도의 오브젝트들만 렌더링하는 하여 연산을 줄이도록 함





Thank you