

Jelly Farm

지원자 : 김현지

최종 수정일 : 2024.01.22

프로젝트 정보

게임 장르 : 클릭어 방치형 게임

엔진 버전 : Unity 2021.3.21f1

게임 설명 :

화면의 젤리를 클릭하여 젤라틴을 얻고 매매하여 골드를 얻습니다.

클릭 하지 않아도 젤리가 골드를 얻어줍니다!

다양한 젤리를 모읍시다.



개발 정보

개발 인원 : 1인

작업 기간 : 10일

취지 :

이제껏 3D 게임만 개발해 왔으니까 2D 게임을 개발해보고 메커니즘을 이해해보자!

프로젝트 소스 링크 : <https://github.com/asterism1030/2D-JellyFarm>

참고 자료

—

링크:

골드메탈 유니티 볼트 기초 키우기게임 젤리팜 [VE2]

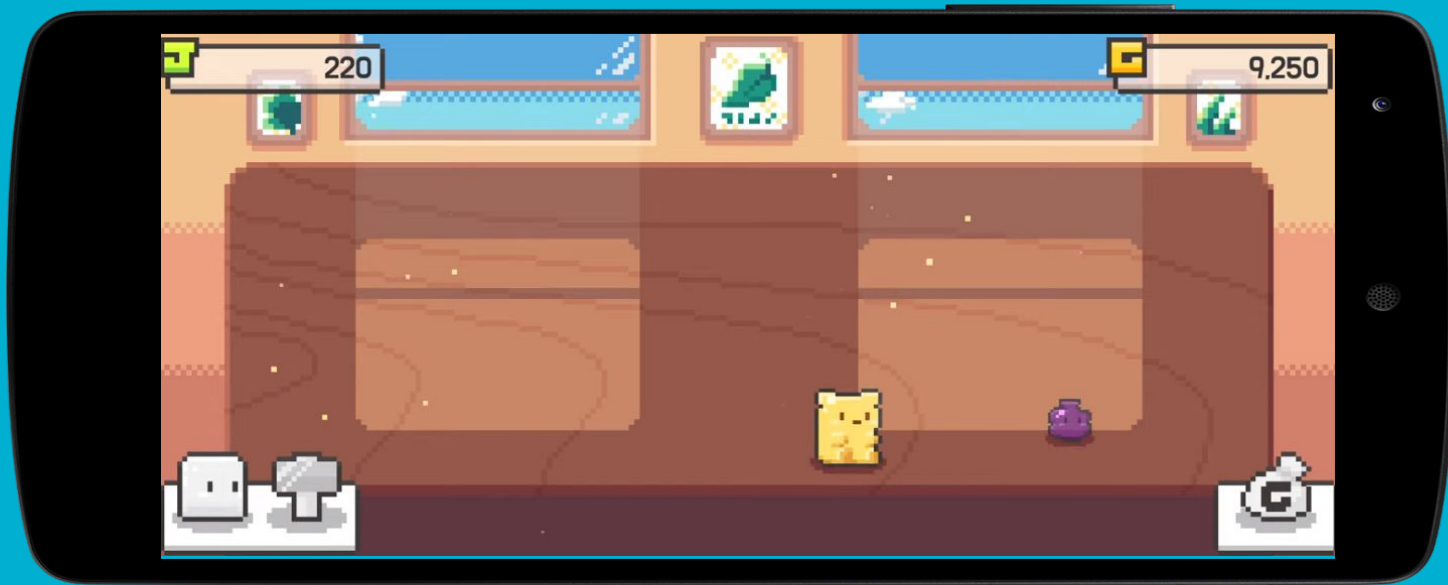
<https://www.youtube.com/playlist?list=PLO-mt5lu5TeZA0y889ZMi9wJafthif03i>

설명:

볼트 구현된 프로젝트를 Script 로 구현

Jelly Farm 플레이 영상

링크 : <https://www.youtube.com/watch?v=CdQOyleKgGw>



화면 설명



개발 과정

2D 게임의
구조 이해

젤리 Obj의
AI 구현

리팩토링

테스트



구현 목록

배경 및 구입 및 해금 팝업 창 UI 구현

젤리 Obj 생성 및 소멸

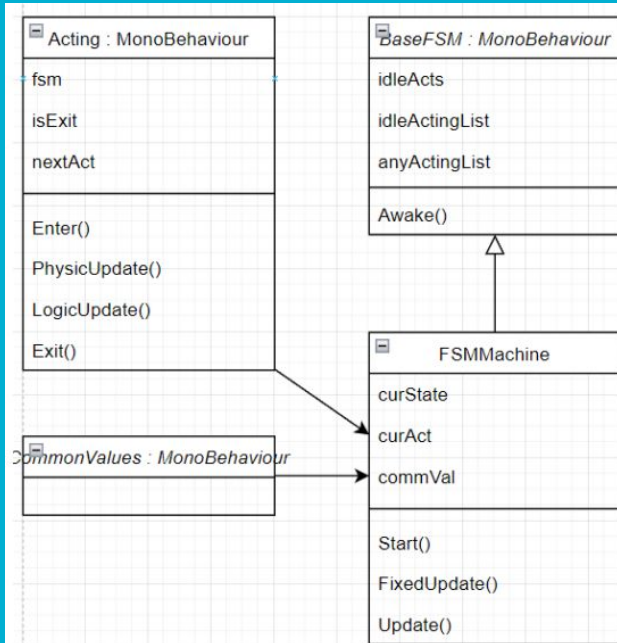
젤리 Obj 클릭 시 젤라틴 획득 드래그 시 이동

일정 시간 경과 후 Lev 업 및 사이즈 변경

FSM 패턴으로 젤리 Obj 의 AI 동작 (뛰기, 걷기, 대기) 구현

배경 창문의 구름 무한 흘러가기 동작

구조



BaseFSM

: Acting 들을 변수로써 관리

- `idleActingList` : 특정 조건을 만족해야 동작
- `anyActingList` : 매번 동작

FSMMachine

: Acting 을 상속한 클래스들을 LifeCycle 주기에 맞춰 동작

- `FixedUpdate` : `Enter`, `PhysicUpdate` 실행
- `Update` : `LogicUpdate`, `Exit` 실행

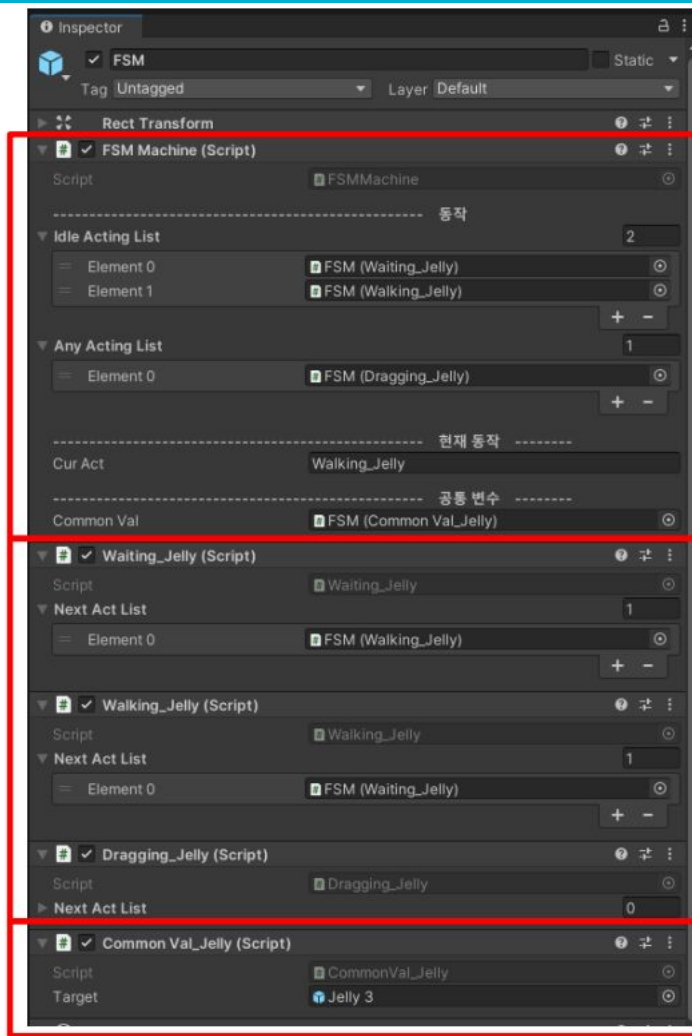
Acting

: Move, Jump 등 하나의 동작

CommonValues

: 편리성을 위한 클래스로 상속하여 Acting 들이 공통으로 사용하는 변수를 선언함

Prefab



FSMMachine

Acting

CommonValues

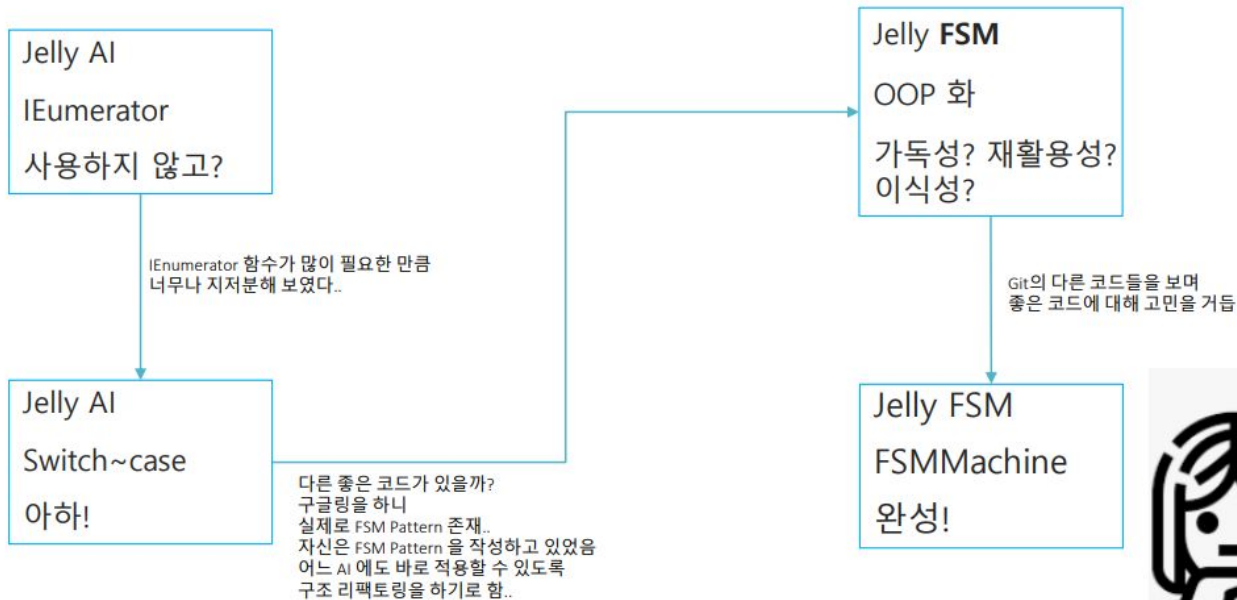
FSM Machine

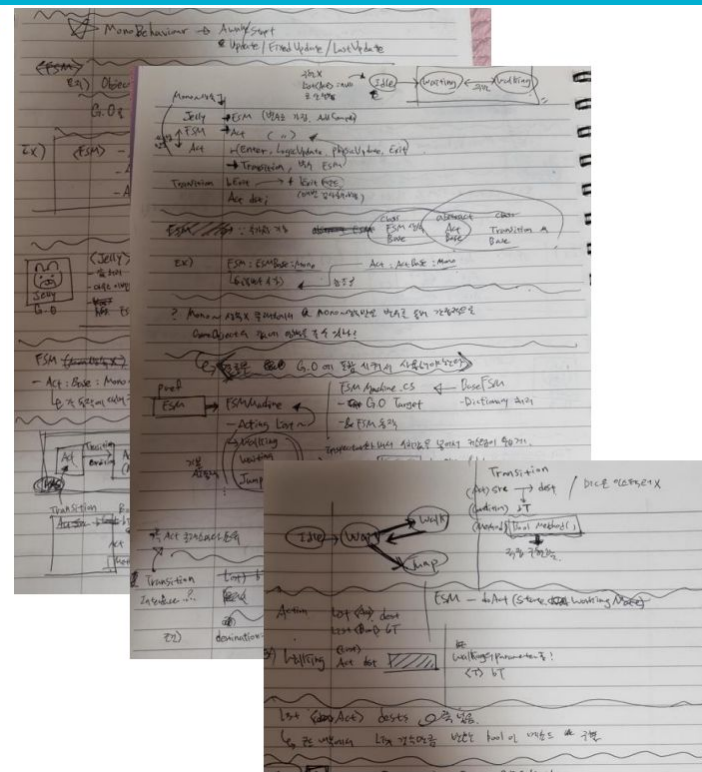
링크 :

<https://github.com/asterism1030/2D-JellyFarm/blob/main/Assets/Script/FSM/FSMMachine.cs>

```
switch(curState) {  
    case State.NONE:                // = idle  
    {  
        idleActs[curAct].Init();  
        curState = State.ENTER;  
    }  
    break;  
  
    case State.ENTER:                // Enter to Node  
    {  
        idleActs[curAct].Enter();  
        curState = State.PHYSICUPDATE;  
    }  
    break;  
  
    case State.PHYSICUPDATE:  
    {  
  
        idleActs[curAct].PhysicUpdate();  
        curState = State.LOGICUPDATE;  
    }  
    break;  
}
```

해결 과정





개발 후기

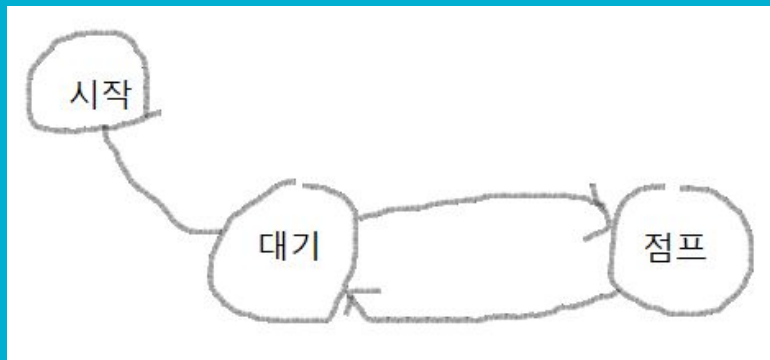
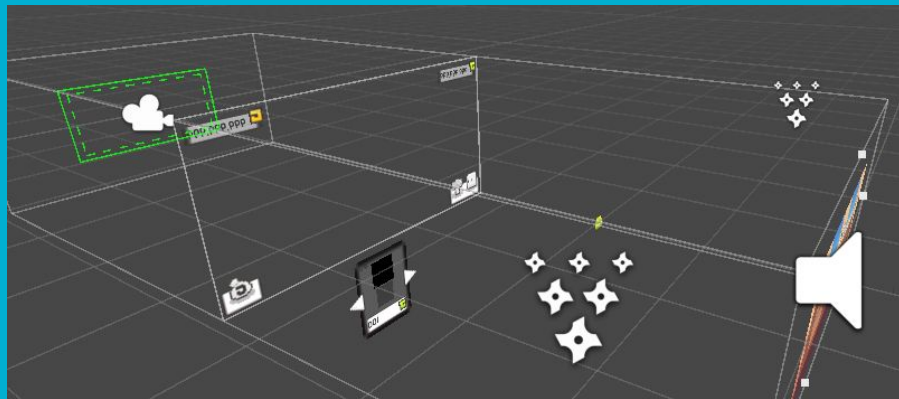
Git이나 구글링을 통해 외국의 정말 좋은! 코드들을 많이 접할 수 있었고
구조 설계의 롤모델이 되었음

클린한 코드를 위해서 다양한 패턴을 알아두는 것도 좋다고 생각했음
상태머신 패턴을 알았다면 시간이 많이 단축되었을 것..

다른 1인 프로젝트 TimeAttack 개발을 하던 때에 비해
게임의 장르와 기능의 명확한 제시, 에셋의 제공으로 편리했음

달성한 것

Layer, Order 등 2D 게임의 메커니즘 이해
FSM Pattern 이해



그리고 앞으로

능숙한 구조 설계를 목표로!

날고 뛰는 개발자는 구조 설계부터 완벽했다.. (feat. 깃허브)

다양한 디자인 패턴의 적절한 적용