

Unity 게임 개발 심화 프로젝트

3조

팀장 | 오우택 팀원 | 김영중 김효재 박한나

목차

- 01 프로젝트 소개 및 진행
- 02 파트 분배
- 03 시연 영상
- 04 주요 기능
- 05 ~~트러블 슈팅~~

1. 프로젝트 소개 및 진행

프로젝트 이름

장르

개발 기간

엔진

정말 재미있는 던전 클릭어

방치형 게임 (Clicker Game)

3월 27일 ~ 4월 2일

Unity



1. 프로젝트 소개 및 진행

1일차 작업 진행

오전

- 10:00 ~ 10:40 : 심화 팀 프로젝트 발제 진행
 - 1~9조 프로젝트 [2D 클리커 게임]
- 11:00 ~ 12:30 : 회의 진행
 - ▼ 내용 (요약)
 - 1. 팀장 선정
 - 2. 프로젝트 진행 방향
 - 3. 코드 컨벤션 설정
 - 4. git 커밋 컨벤션 설정
 - 5. 역할 분담
- 13:00 ~ 14:00 : 점심 시간

오후

- 14:00 ~ 18:00 : 프로젝트 진행
- 18:00 ~ 19:00 : 저녁 시간
- 19:00 ~ 20:00 : 정기 회의 진행
 - ▼ 내용 (요약)
 - 1. 프로젝트 진행 공유 (발표 순서대로)
 - ▼ 작업 내용
 - 1. 오우택 - Weapon&Upgrade UI 제작 / (회의 이후 추가) PointBk, GoldBk, MainUI 병합(임시)
 - ▶ 영상 (길쭉해요)
 - 2. 박한나 - Player Upgrade 분리 및 구조 설계 및 작성 중, Player State > Dictionary 전환
 - 3. 김영중 - UIManager 작성 및 동작 과정 설명, OpenWarningMessage 구현, OneClick + autoAttack 구현
 - 4. 김효재 - Enemy 구조 설계 및 작성 중, stage 구조 설계 및 작성 중
- 20:00 ~ 21:00 : 작업 마무리 후 TeamNotion, 개인 TIL 작성

2일차 작업 진행

오전

- 09:00 ~ 09:30 : 개별적으로 하루 계획표 작성
- 09:30 ~ 09:40 : 아침 회의 진행
 - ▼ 내용 (발표 순)
 - 1. 오우택 : WeaponUI 수정, WeaponDatatable 작성, WeaponUI.cs 작성
 - 2. 김영중 : StartScene 제작, GameManager 도전 과제 - Save&Load, ClickEvent의 ParticlesSystem 추가, GameManager 필수 과제 - Bgm출력과 OptionWindow 제작
 - 3. 박한나 : PlayerButtonUpgrade Test > GameManager에서 메서드 추가, 필수 과제 확인 및 구현
 - 4. 김효재 : Stage 구현, Stage UI, Enemy UI 제작, Stage 동작 Test

4일차 작업 진행

오전

- 09:00 ~ 09:30 : 개별적으로 하루 계획표 작성
- 09:30 ~ 09:40 : 아침 회의 진행
 - ▼ 내용 (발표 순)
 - 1. 김영중 : Player data 생성 및 save&load attack 관련 수정, 병합
 - 2. 오우택 : weaponInventory slot 마무리
 - 3. 박한나 : 속도 관련 문제, 값이 테스트 변경 때문에 버그 수정, playerdata 연결, UI 연결 작업
 - 4. 김효재 : 스테이지마다 다른 적 출현 가능하도록 이미지 추가(4개), 기존에 있는 기능 main 병합 기능.
- 19:00 ~ 20:00 : 정기 회의 진행
 - ▶ 내용 (요약)
- 20:00 ~ 21:00 : 작업 마무리 후 TeamNotion, 개인 TIL 작성

3일차 작업 진행

오전

- 09:00 ~ 09:30 : 개별적으로 하루 계획표 작성
- 09:30 ~ 09:40 : 아침 회의 진행
 - ▼ 내용 (발표 순)
 - 1. 김영중 : clickEvent 예외 처리 수정, autoAttack 파티클 수정, BGM, playerData 불러오기 기능 제작
 - 2. 오우택 :
 - 3. 박한나 : UI 연결, 코드 리팩토링, 예셋 검색
 - 4. 김효재 : 동작 test, 애니메이션 구현
- 19:00 ~ 20:00 : 정기 회의 진행
 - ▼ 내용 (요약)
 - 1. 프로젝트 진행 공유 (발표 순서대로)
 - ▼ 작업 내용
 - 1. 김영중 - MainUI 작성, 음악 설정 기능 추가, SoundManager, EffectHolder, FadeManager 제작, BGM 추가
 - 2. 박한나 - WeaponUI 작성 및 UI 연결 작업 중, UpdateUI 제작
 - 3. 오우택 - WeaponDataManager.cs 작성,

오전 회의 - 작업 예정 사항 공유

오후 회의 - 진행 사항 공유 및 기능 동작 설명

- 20:00 ~ 21:00 : 식습 마무리 후 TeamNotion, 개인 TIL 작성

2. 파트 분배



팀장 **오우택**

무기 및 업그레이드



김영중

게임 매니저 & 클릭 이벤트



김효재

적 및 스테이지



박한나

플레이어 능력치 및 업그레이드

3. 시연 영상



4. 주요 기능 클릭 이벤트

참조 2개

```
public void PerformAttack(bool Auto = false)
{
    Enemy target = EnemyManager.Instance.CurrentEnemy;
    if (target != null && !isPaused)
    {
        float random = UnityEngine.Random.Range(0.0f, 1.0f);
        float _critChance = WeaponManager.Instance.EquippedWeapon.criticalPercentage;
        if (random < _critChance) //크리티컬 데미지
        {
            target.TakeDamage(finalCritDamage);
            SoundManager.Instance.ApplyCriticalSFX();
        }
        else //일반 공격
        {
            target.TakeDamage(finalAttackPower);
            SoundManager.Instance.ApplyMainSceneSFX();
        }
        SoundManager.Instance.PlaySFX();
        gold += finalGoldBonus;
        UIManager.Instance.MainUI.UpdateUI();
        if (Auto)
        {
            GameObject fx = Instantiate(clickEffectPrefab, target.transform.position, Quaternion.identity, effectHolder);
            Destroy(fx, 1f);
        }
        else
        {
            SpawnClickEffect(Input.mousePosition); // 클릭 이펙트 여기서 실행!
        }
    }
}
```

4. 주요 기능 데이터 직렬화

```
//데이터 직렬화용 딕셔너리 클래스
[Serializable]
public class Dict<TKey, TValue>
{
    public TKey key;
    public TValue value;

    public Dict(TKey key, TValue value)
    {
        this.key = key;
        this.value = value;
    }
}

//데이터 직렬화용 StatData
[Serializable]
public class SerializableStatData
{
    public List<Stat> stats = new List<Stat>()
}
```

```
public class Character
{
    /// <summary>
    /// 딕셔너리와 스탯데이터에 직렬화를 전달해주는 역할
    /// </summary>
    public void LoadValue()
    {
        upgradeOptions = new Dictionary<UpgradeType, UpgradeOption>();
        statData = ScriptableObject.CreateInstance<StatData>();
        statData.Make();

        foreach (Dict<UpgradeType, UpgradeOption> dic in UO)
        {
            upgradeOptions.Add(dic.key, dic.value);
        }
        foreach (Stat stat in SD.stats)
        {
            statData.SetStat(statData.FindStatIndex(stat.stat), stat.totalValue);
        }
    }
}
```


4. 주요 기능 스테이지 시작 (Stage Manager)

```
public void StartStage(int stageIndex)
{
    //UI
    UIManager.Instance.StageUI.UpdateStageText(currentStageIndex);

    currentStageIndex = stageIndex;

    EnemyData stageEnemy = ScriptableObject.CreateInstance<EnemyData>();
    stageEnemy.enemyName = enemyTemplate.enemyName + $"{stageIndex + 1}";
    // 스테이지 난이도 - 스테이지 올라갈수록 적의 체력, 개수 증가
    stageEnemy.maxHealth = baseHealth + healthPerStage * stageIndex;
    stageEnemy.enemyCount = baseCount + countPerStage * stageIndex;

    // 보상 - 적 죽이면 포인트
    int pointReward = basePointReward + pointPerStage * stageIndex;
    EnemyManager.Instance.SetPointReward(pointReward);

    if (allEnemySprites != null && allEnemySprites.Length > 0)
    {
        stageEnemy.enemySprite = allEnemySprites[Random.Range(0, allEnemySprites.Length)];
    }

    EnemyManager.Instance.SpawnEnemy(stageEnemy);
}
```

4. 주요 기능 무기 데이터 관리 (WeaponDataManager)

```
public void CreateDefaultCsv()
{
    List<WeaponData> defaultWeapons = new List<WeaponData>
    {
        // 이름, 아이콘파일이름, Lv, DMG, Crit, Crit증가, 업그레이드 비용 증가, 데미지 증가, 초기 구매비용, 초기 업그레이드 비용, 구매여부, 장착여부
        CreateWeaponData("나무검", "woodsword.png", 1, 1, 5.00f, 0.05f, 10, 1, 10, 10, true, true),
        CreateWeaponData("돌검", "stonesword.png", 1, 5, 10.00f, 0.10f, 20, 2, 20, 20, false, false),
        CreateWeaponData("철검", "ironsword.png", 1, 10, 15.00f, 0.20f, 40, 4, 40, 40, false, false),
        CreateWeaponData("황금검", "goldensword.png", 1, 20, 20.00f, 0.4f, 80, 10, 80, 80, false, false),
        CreateWeaponData("다이아몬드검", "diamondsword.png", 1, 50, 25.00f, 1.00f, 100, 20, 100, 100, false, false)
    };

    SaveWeapons(defaultWeapons);
}
```

WeaponData.csv

WeaponData.csv.meta



goldensword.png



goldensword.png.meta



ironsword.png



ironsword.png.meta



stonesword.png



stonesword.png.meta



woodsword.png



woodsword.png.meta

weaponName	icon	upgradeLv	weaponDmg	criticalPer	increasedCr	increasedD	increasedDp	purchaseC	upgradeC	IsPurchase	IsEquiped
나무검	woodsword	2	2	5.05	0.05	10	1	10	20	TRUE	FALSE
돌검	stonesword	4	11	10.3	0.1	20	2	20	80	TRUE	TRUE
철검	ironsword	1	10	15	0.2	40	4	40	40	TRUE	FALSE
황금검	goldensword	3	40	20.8	0.4	80	10	80	240	TRUE	FALSE
다이아몬드검	diamondsword	5	130	29	1	100	20	100	500	TRUE	FALSE