특명! 호밀빵을 지켜라

4조. 호밀빵빵 팀

개요

1 게임 소개

2 적용 기술

3 느낀점 및 고찰

Part 1.

게임 소개

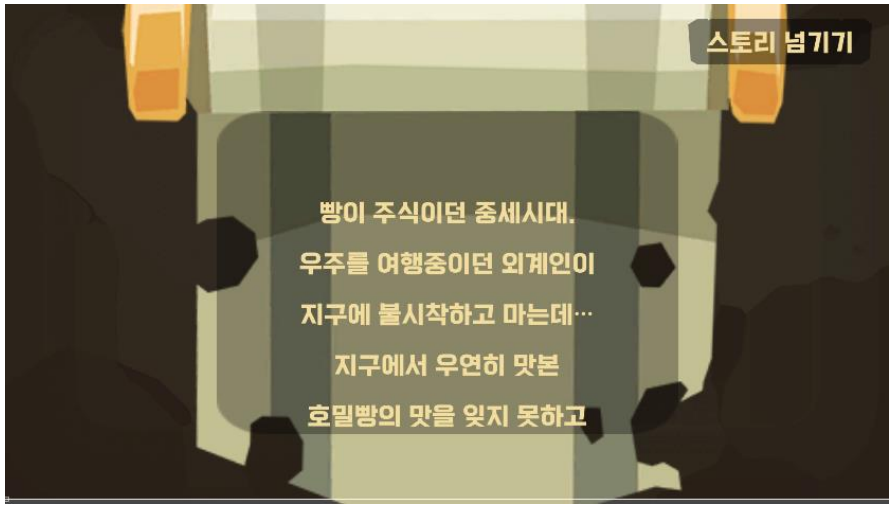
게임 소개

타워 디펜스 장르

+

RPG 요소

게임 소개


스토리 넘기기
빵이 주식이던 중세시대.
우주를 여행중이던 외계인이
지구에 불시착하고 마는데…
지구에서 우연히 맛본
호밀빵의 맛을 잊지 못하고

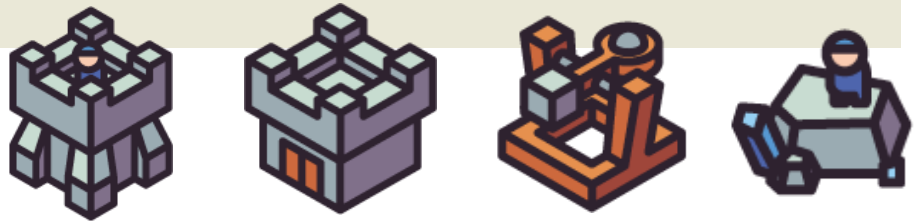

0
0
1


20
462
65 /100
김덕배
침공 2/5
Q
W

게임 소개 세부 사항

정건창 타워

- 4가지 타워 구현
  병영/ 궁수/ 투석기/ 마법사
- 타워 건설 UI,
  타워 업그레이드 및 판매 UI,
  타워 정보 UI 구현
- 타워의 적 인식 및 공격
- 타워 별 투사체 구현
- 타워 별 애니메이션 구현
- ~~타워 특성 (영구적용)~~



게임 소개 세부 사항

고지은 영웅

- 이동 구현
우클릭으로 이동/정해진 타일만 클릭할 수 있도록 구현

- 자동 기본공격 구현
사거리 안에 몬스터 인식/
가장 가까운 몬스터를 타겟팅/ 이동/ 기본공격

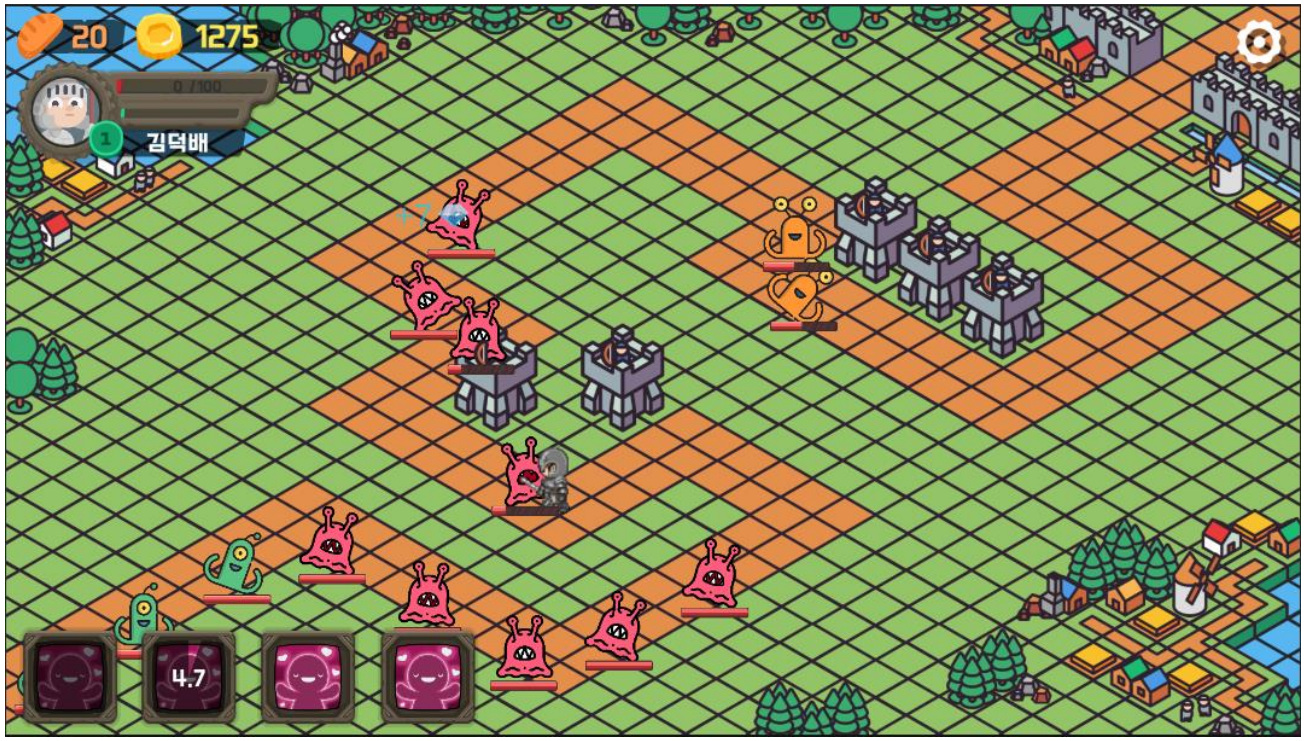
- 스킬 및 스킬 별 쿨타임 구현
스킬 1. 범위 내 병사들에게 공격력과 체력 버프 적용
스킬 2. 체력이 50 이하가 되면 최대체력의 20%를 회복
스킬 3. 병사 3명 소환
스킬 4. 영웅 주위에 광역/지속 데미지

- 영웅 스탯 및 스킬 업그레이드/UI 구현

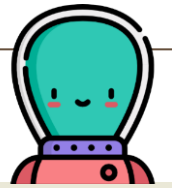
- 부활 및 이펙트 구현

- ~~여러 영웅 구현 → 영웅 선택~~

게임 소개 세부 사항



이영종 몬스터



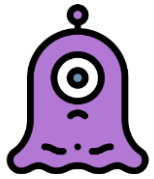
- 길 타일을 따라 자동으로 이동,
  도착포인트에 도달하면
  라이프(빵)가 차감되는 것을 구현

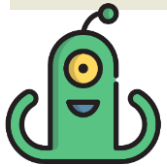
- 병사 유닛이 감지범위 안에 들어오면
  위치를 추적해 전투실행

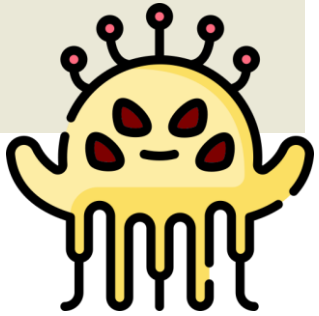
- 여러 종류의 몬스터 오브젝트를 구현

- 사망 시 다이아와 경험치를 드랍

<추가 요소>
- 인게임에서 무작위 UFO 등장 이벤트 구현
  클릭 시 사라지며 재화를 드랍



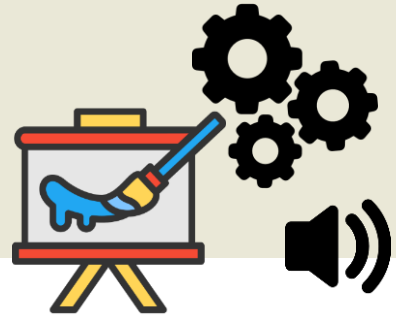




게임 소개 세부 사항

박유진 UI / 맵

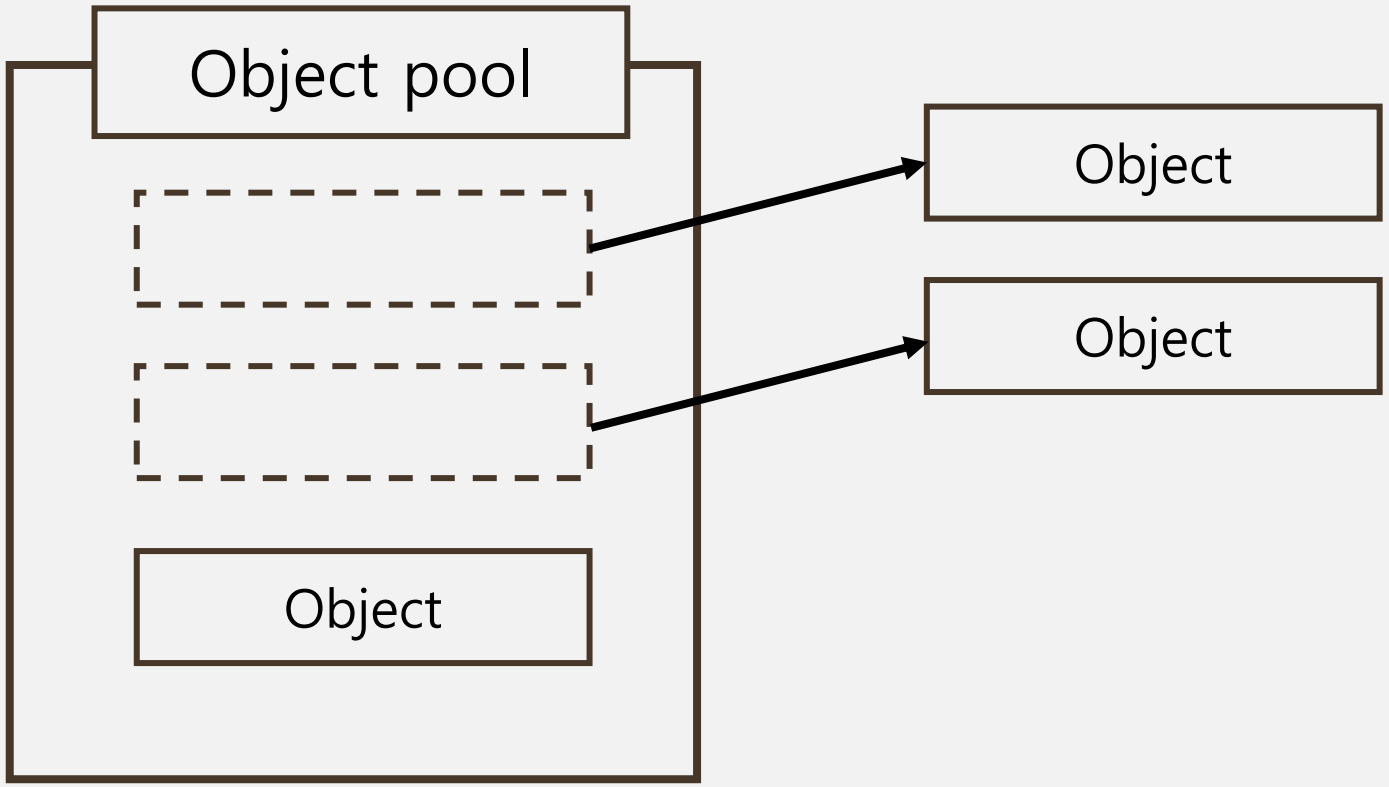
- 메인메뉴 ui 구현
- 스토리창 ui 구현
- 스테이지 선택창 ui 구현
- 영웅 정보창 ui 구현
- 타워 스킬창 ui 구현
- 인게임 ui 구현
- 게임 bgm 기능 구현
- ~~스테이지 추가~~
- ~~영웅 ui 추가~~
- ~~타워 특성 ui 추가~~



Part 2.

적용 기술

Object Pooling



게임 내 반복되는
오브젝트의 생성/ 파괴

↓

부하가 크고,
GC(Garbage Collector)를 발생시켜
성능을 저하시킴

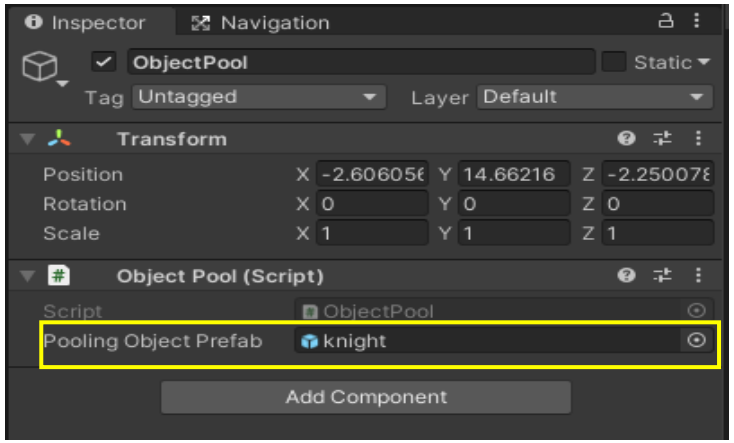
⬇

이를 방지하는 최적화 기법

적용 기술 Object Pooling

```
public class ObjectPool : MonoBehaviour
{
    public static ObjectPool Instance;

    [SerializeField]
    private GameObject poolingObjectPrefab;

    private Queue<knight> poolingObjectQueue = new Queue<knight> ();

    Unity 메시지 | 참조 0개
    private void Awake()
    {
        Instance = this;

        Initialize(3);
    }

    참조 2개
2   private knight CreateNewKnight()
    {
        var newObject = Instantiate(poolingObjectPrefab, transform).GetComponent<knight>();
        newObject.gameObject.SetActive(false);
        return newObject;
    }

    참조 1개
1   private void Initialize(int count)      //입력받은 개수만큼 우선 오브젝트 풀에 만들어놓기
    {
        for(int i = 0; i < count; i++)
        {
            poolingObjectQueue.Enqueue(CreateNewKnight());
        }
    }
```

```
참조 3개
public static knight GetObject()   오브젝트 풀에서 오브젝트를 꺼내는 함수   3
{
    if(Instance.poolingObjectQueue.Count>0)
    {
        var obj = Instance.poolingObjectQueue.Dequeue();
        obj.transform.SetParent(null);
        obj.gameObject.SetActive(true);
        return obj;
    }
    else
    {
        var newObj = Instance.CreateNewKnight();
        newObj.transform.SetParent(null);
        newObj.gameObject.SetActive(true);
        return newObj;
    }
}

참조 2개                                                                                    4
public static void ReturnObject(knight knight)  //빌려줬던 오브젝트를 돌려받는 함수
{
    knight.gameObject.SetActive(false);
    knight.transform.SetParent(Instance.transform);
    Instance.poolingObjectQueue.Enqueue(knight);
}
```



적용 기술 Object Pooling

```
void WarriorSkill3()
{
    knight1 = ObjectPool.GetObject();
    knight1.transform.position = transform.position + new Vector3(0.5f, 0f, 0);
    knight1.WorriorSpawnSetting();

    knight2 = ObjectPool.GetObject();
    knight2.transform.position = transform.position + new Vector3(-0.5f, 0f, 0);
    knight2.WorriorSpawnSetting();

    knight3 = ObjectPool.GetObject();
    knight3.transform.position = transform.position + new Vector3(0f, -0.5f, 0);
    knight3.WorriorSpawnSetting();

    skill3Effect.SetActive(true);
    Invoke("Skill3EffectOff", 4f);
}
```

3

```
private void DestroyKnight()
{
    ObjectPool.ReturnObject(this);
}
```

4

Part 3.

느낀점 및 고찰

느낀점 및 고찰

정건창 팀장 👑

유니티에서 제공해주는 기능이 많아 인터넷을 찾아보면 찾던 기능들이 다 있어서 게임 만들 때 편했고,
팀원들 모두 주말에도 열심히 해줘서 처음 기획한 부분은 다 구현하고 잘 마무리할 수 있었던 거 같다.

이영종

기존의 계획과 달리 나의 생각대로 구현되지 않는 부분들이 많았고 예상치 못한 버그들도 빈번하게 발생했다.
결국 끊임없이 코드를 수정해 나가면서 처음부터 코드를 제대로 짜야 됐다는 점을 뼈저리게 깨닫고 말았다.
또한 프로젝트를 진행하면서 필요에 의해 다양한 기능이나 코드들을 찾아보게 되었고 이번 기회를 통해 한층 성장하는 계기가 되길 바란다.

박유진

프로젝트를 시작하기 전에 걱정이 많았지만 막상 시작해서 프로젝트를 진행하다 보니 너무 재미있었고
제일 쉬운 UI 파트를 맡아서 한게 정말 다행이라고 생각했다 다음에 또 하고싶다.그리고 팀원 분들이 너무 잘생기고 예뻐서 좋았다.

고지은

인게임에서는 상대적으로 비중이 작은 소소한 부분들인데도 많은 오류가 발생하는 것을 겪으며 많은 것을 느꼈고,
이런 저런 오류를 겪고, 직접 해결해나가는 과정에서 많은 것을 배울 수 있었다.
그리고 팀원들에게 많이 배우고 팀원들 덕에 동기부여도 많이 돼서 더 힘내서 할 수 있었다.

감사합니다