



SW부트캠프 팀 프로젝트

트

기업 : 행복한다람쥐단
팀명 : 차조기
팀원 : 차정민, 조상호, 김태영, 김
준영

목차

Table of Contents

1. 사전 계획

- 주제 선정
- 에셋 탐색

2. 개발환경 및 절차

- 깃허브 관리
- 크라켄 활용
- 역할 분담

3. 구현 결과

- 기능 소개

4. 작품 시연

5. 마무리

- 후기
- Q & A



A blue-tinted photograph of a spiral-bound notebook. On the right page, a metal pencil sharpener is placed, surrounded by several shavings of wood and graphite. A pencil is partially visible, extending from the sharpener towards the bottom right corner. The text 'Part 1' and '사전 계획' is overlaid in white on the center of the image.

Part 1

사전 계획

#주제 선정

주제 선정



만들고 싶은 것

VS.



현실적인 문제

주제 선정



활용 가능한 에셋 탐색

and



에셋에 기반한 게임 설계

주제 선정



유니티 에셋 스토어 무료 에셋 활
용

#에셋 탐색

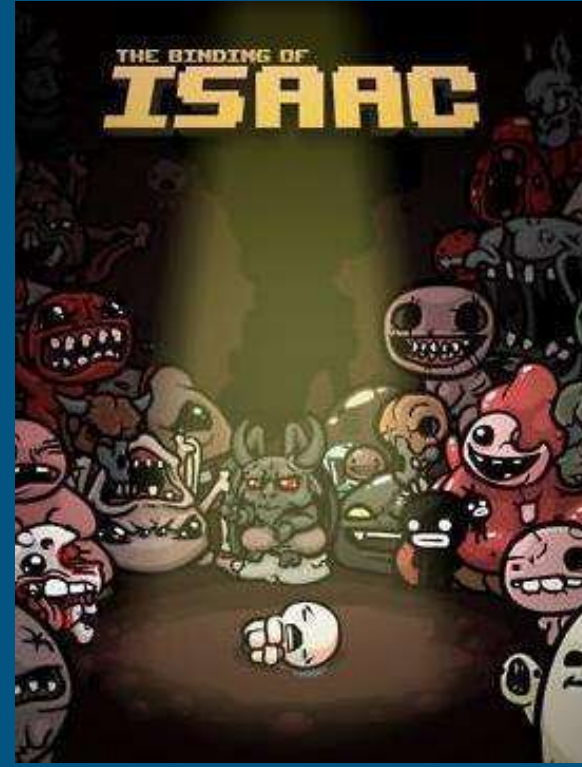
에셋 탐색



+



+



드래곤 보스 캐릭터

용사 캐릭터

아이작 모티브



Rogue-Like!



Part 2

개발 환경 및 절차

#개발환경

개발환경



Visual Studio 2022
JetBrains Rider



Unity 2022.3.4f1

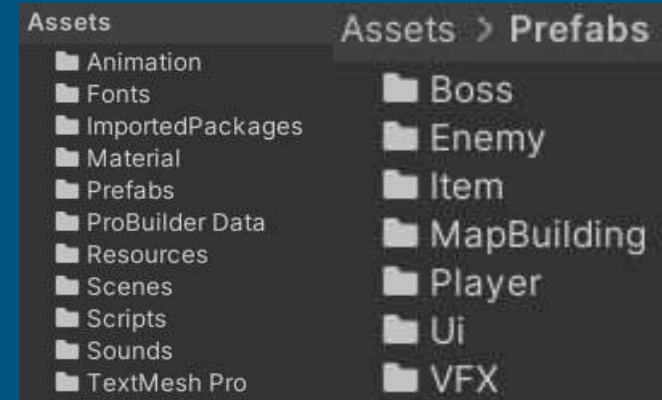


GitKraken

프로젝트 관리



각각의 브랜치 생성
주기적인 병합



프로젝트 폴더 세분화

#역할 분담

역할 분담

- 플레이어
- 보스
- 아이템

차정민

- UI/UX
- 로딩 화면
- 인벤토리

김준영

- 적AI
- GameManager 기능
- 미니맵

조상호

김태영

- 절차적 생성 던전 맵
- 포탈, 스테이지 기능
- 특수방 설계



Part 3

구현 결과

#기능 소개

Main Scene

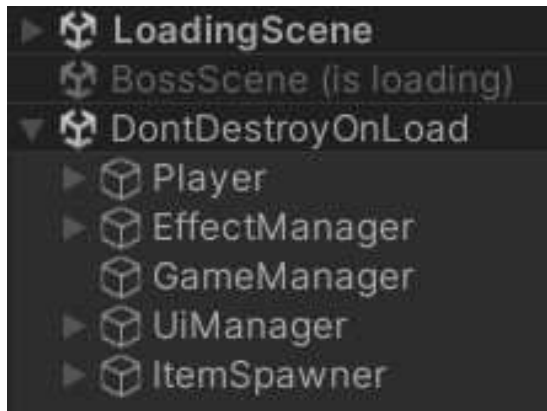


메인 화면



옵션 메뉴

- 3가지 볼륨 조절
- 16 : 9 해상도 선택
- 전체 화면 on/off



Loading Scene

- 비동기적 로드
- 랜덤 배경화면



Game Scene

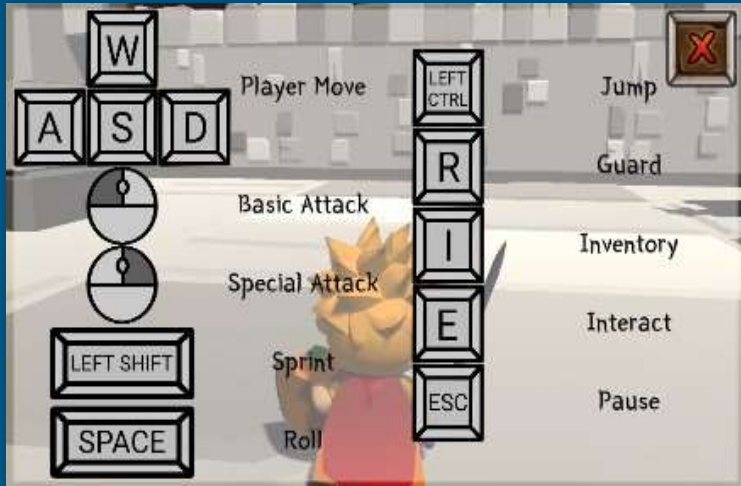
- 게임 시작시 화면
- 도움말 UI
- 미니맵
- 체력바
- 골드 보유량



Inventory

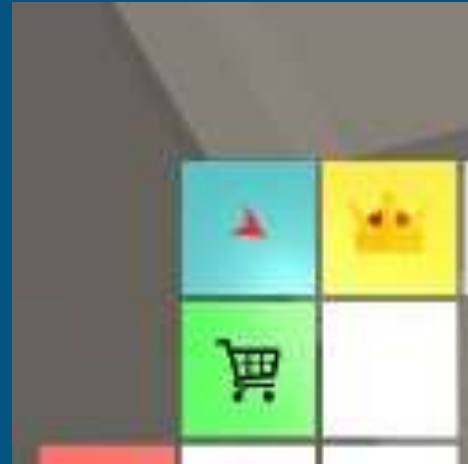
- 획득 아이템 목록 표시
- 아이템 상세 기능 설명 툴팁
- 마우스 드래그로 위치 조정 기능

UI/UX



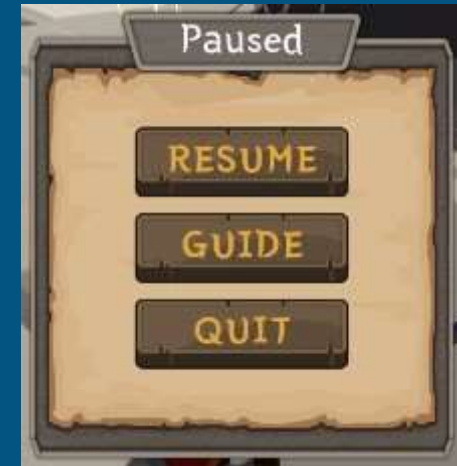
Guide Image

- 플레이어 이동, 공격 조작법
- Ui 입력키



Mini Map

- 현재 위치 및 방향 표시
- 특수방 아이콘 (황금방, 상점방 등)



Pause Menu

- 게임 일시정지
- 조작법 확인
- 게임 종료 버튼

```
public void StartFinding(Vector2Int startVector, int distance)
{
    List<Vector2Int> temp = new();

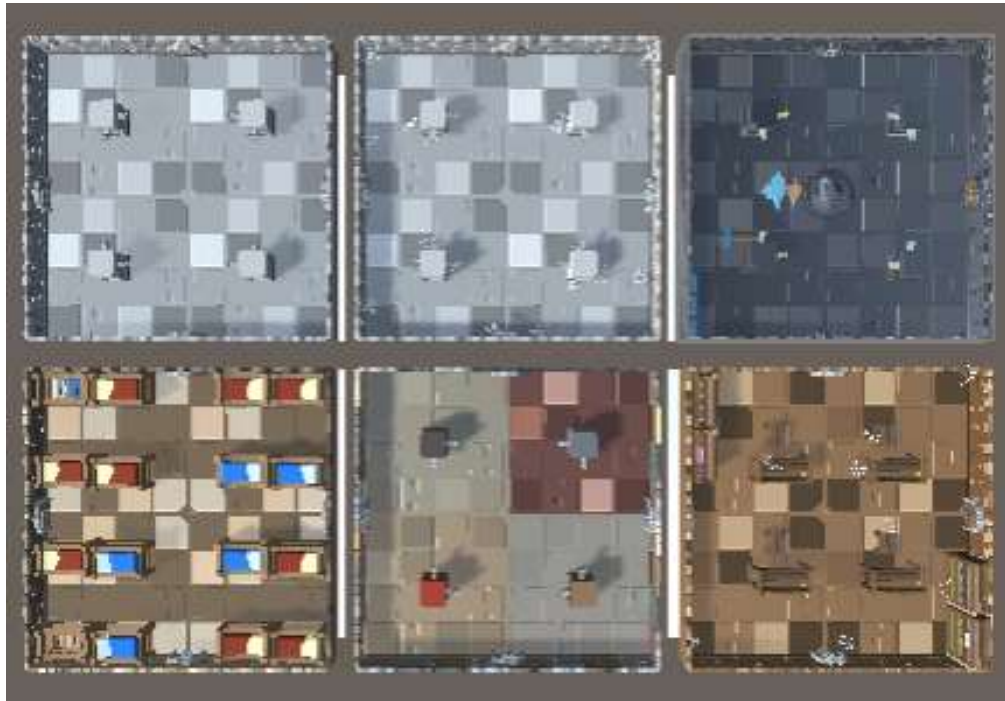
    temp.Add(new Vector2Int(startVector.x + distance, startVector.y));
    temp.Add(new Vector2Int(startVector.x - distance, startVector.y));
    temp.Add(new Vector2Int(startVector.x, startVector.y + distance));
    temp.Add(new Vector2Int(startVector.x, startVector.y - distance));

    foreach(var i in temp)
    {
        if (!candidateVector.Contains(i) && !mapVector.Contains(i))
        {
            candidateVector.Add(i);
        }
    }
    Selecting();
}
참조 1개
public void Selecting()
{
    var rand = Random.Range(0, candidateVector.Count);
    var randVector = candidateVector[rand];

    mapVector.Add(randVector);
    candidateVector.Remove(randVector);
}
```

Procedural Generation

- 절차적 지형 생성 알고리즘 사용
- 처음 지점과 일정 거리에 있는 지점 탐색
- 중복 검사 후 랜덤 지점 선택
- 위에 과정 반복 시행으로 랜덤한 맵 생성



Dungeon Map

- 다양한 테마의 Room
- 생성된 적 처치시 포탈 잠금 해제



Enemies

- Slime, TurtleShell, Beholder, ChestMonster, Golem 5종
- NavMesh Agent를 활용한 경로 탐색

```
🔥 Event function 👤 kiasma99
private void Update()
{
    curState = nextState;

    if (isStateChanged) StateEnter();
    isStateChanged = false;

    StateUpdate();

    isStateChanged = TransitionCheck();

    if (isStateChanged) StateExit();
}
```

Enemy AI

- State에 따라 다양한 애니메이션 실행
- 조건에 따라 State Transition

Special Room



Golden Room

- 황금으로 도배된 방
- 중앙의 상자를 열면 가중치 확률 기반 아이템 드랍



Shop Room

- 가중치 확률 기반 아이템 출현
- 획득한 골드를 사용하여 아이템 구매



Boss Room

- 맵 중앙의 포탈을 타고 다음 스테이지로 이동
- 3 스테이지 클리어 시 BossScene으로 이동



Item

- 3가지 등급(Common, Rare, Epic)
- 체력 회복, 공격력 강화, 번개 소환 등 다양한 기능 구현

```
public GameObject GetRandomItem()
{
    float totalWeight = 0f;
    foreach (var item in _items)
    {
        if (!item.ItemEnabled) continue;
        totalWeight += item.GetItemWeight();
    }
    var randomValue float = Random.Range(0f, totalWeight);
    var curWeight = 0f;

    ItemObject selectedItemObject = null;
    foreach (var item in _items)
    {
        if (!item.ItemEnabled) continue;
        curWeight += item.GetItemWeight();
        if (randomValue < curWeight)
        {
            selectedItemObject = item;
            break;
        }
    }

    if (selectedItemObject is null) return null;
    var selectedItem:GameObject = Instantiate(selectedItemObject.gameObject);
    Instantiate(itemSparkleVfx[(int)selectedItemObject.GetItemTier()], selectedItem);
    return selectedItem;
}
```

Weighted Random Item

- Item Scriptable Object마다 정의된 가중치
- 랜덤 값과 누적 가중치 값을 비교하여 아이템 생성



Player Character

- 3인칭 시점 Cinemachine Virtual Camera
- 달리기, 구르기, 일반 공격과 특수 공격, 가드 등 다양한 액션 구현


```
⚡ Frequently called 2 usages c:\min1233
private void GrowPool(int index)
{
    for (int i = 0; i < 10; i++)
    {
        var instanceToAdd:GameObject = Instantiate(effects[index], tr
        Add2Pool(index, instanceToAdd);
    }
}

⚡ Frequently called 4 usages c:\min1233
public void Add2Pool(int index, GameObject instanceToAdd)
{
    instanceToAdd.SetActive(false);
    instanceToAdd.transform.SetParent(transform);
    effectQueue.Enqueue(index, instanceToAdd);
}

⚡ Frequently called 9 usages c:\min1233
public GameObject GetFromPool(int index)
{
    if (effectQueue.Count(index) <= 0) GrowPool(index);
    return effectQueue.Dequeue(index);
}
```

Object Pooling

- 게임 내에서 반복적으로 사용되는 오브젝트
- 사용한 오브젝트를 EnQueue
- 필요할 때 DeQueue


```
private void Start()
{
    if (!Instance) Instance = this;
    else if (!Instance.Equals(other: this))
    {
        Destroy(gameObject);
        return;
    }
    DontDestroyOnLoad(gameObject);
}
```

Singleton Pattern

- 게임 내에서 하나만 존재하는 오브젝트 (GameManager, UiManager 등)
- 여러 클래스에서 접근 가능하여 데이터 공유 및 관리가 용이



Boss Scene


- 좁은 맵에서 보스와 전투
- 8가지 일반 패턴
- 일정 체력 소진시 강력한 화염 공격

Part 4

작 품 시 연







Part 5

마무 리

#후기

차정민

팀원들의 유니티 활용 경험이 적었음에도 불구하고 많은 기능들을 직접 구현해 보면서 C# 프로그래밍에 대한 이해와 유니티 활용 능력을 높일 수 있었습니다. 또한 팀장으로서 전체 프로젝트를 관리하고 협업하는 과정이 색다른 경험이었습니다.

김준영

팀원들과 함께한 저의 첫 게임개발은 뜻깊고 재미있었습니다. 직접 부딪혀가며 문제들을 해결해가는 과정에서 많이 배울 수 있었습니다. 또한 팀으로서 하는 협업의 중요성도 깨닫게 되었습니다.

김태영

짧은 시간 프로젝트를 진행하면서 개인 프로젝트와 달리 팀원간의 협업의 중요성을 알았습니다. 혼자서는 만들지 못하는 방대한 프로젝트에도 팀원들과의 소통과 분담을 통하여 개발할 수 있었던 뜻깊은 프로젝트였습니다.

조상호

이번 프로젝트를 진행하면서 독학으로는 하기 힘든 팀원과의 협업을 경험할 수 있어서 좋았습니다. 또한 기존에는 복잡하게 해결했던 문제를 보다 간결하고 쉽게 해결할 수 있는 기술을 기를 수 있었습니다.

The background of the image features several palm fronds in various orientations, creating a layered, tropical aesthetic. A semi-transparent blue filter is applied across the entire image, softening the colors and providing a cohesive background for the text.

Q&A