

# 내일배움캠프 팀 프로젝트

- With 게임 만들어서 8조 -

## 라스트 투척의 신

김태호 - 몬스터  
김성원 - DB, 맵  
김소연 - UI, 사운드  
이상훈 - 무기, 투사체  
김영중 - 캐릭터, 스킬

# 목차



01 와이어 프레임



02 시연 영상



03 구현 기능



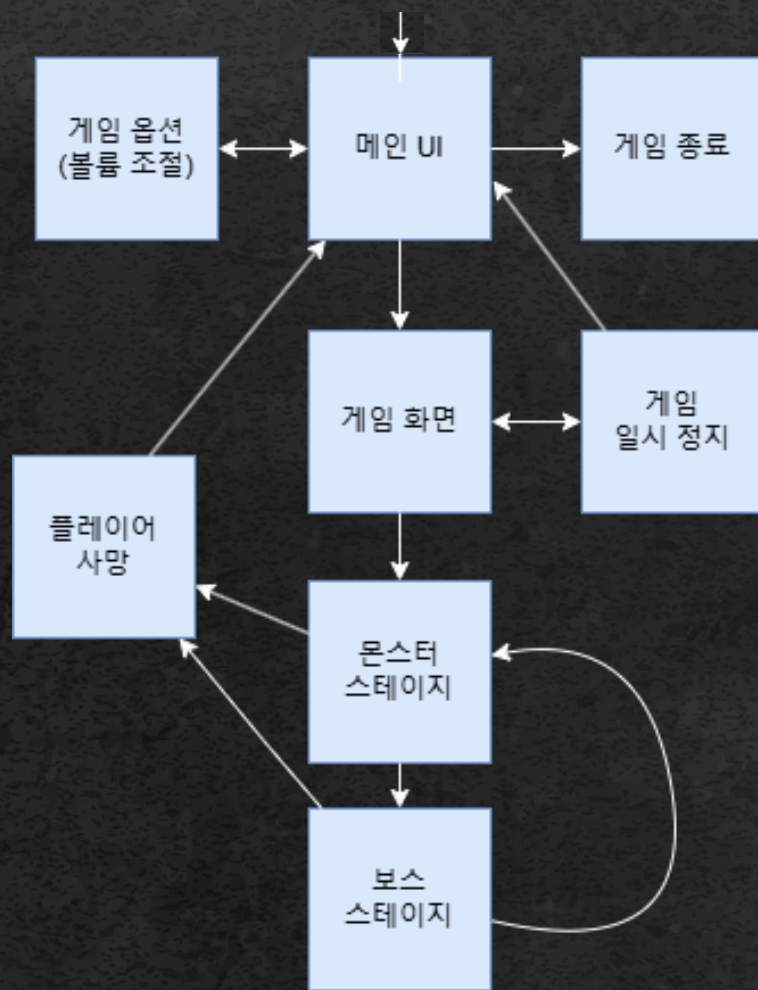
04 트러블 슈팅



05 느낀 점 앞으로의 각오

# 01 와이어 프레임

## 라스트 투척의 신





## 02 시연 영상



## 03 구현 기능 사항

### 필수 기능

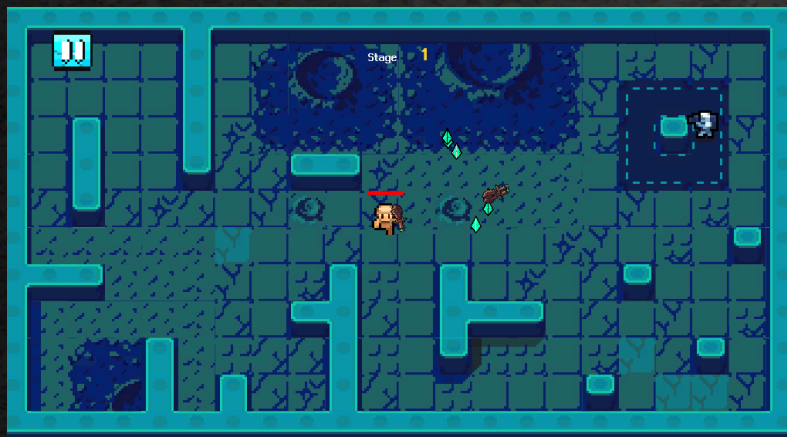
랜덤 방 생성  
캐릭터 이동과 공격  
적 AI와 공격 패턴  
스킬과 업그레이드 시스템

### 도전 기능

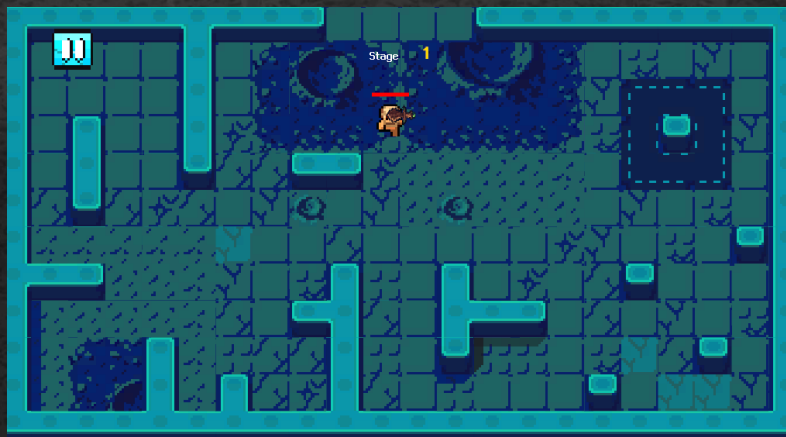
배경 음악 추가  
사운드 효과 추가  
간단한 게임 시작 화면  
발사체 반사  
스테이지 클리어  
빌드 및 배포

# 03 구현 기능 사항

## 스테이지 클리어



스테이지 시작



스테이지 클리어



신규 스테이지 시작

스테이지 클리어 시  
다음 스테이지 이동 공간  
Open

다음 스테이지 이동 후  
신규 랜덤 맵 Draw



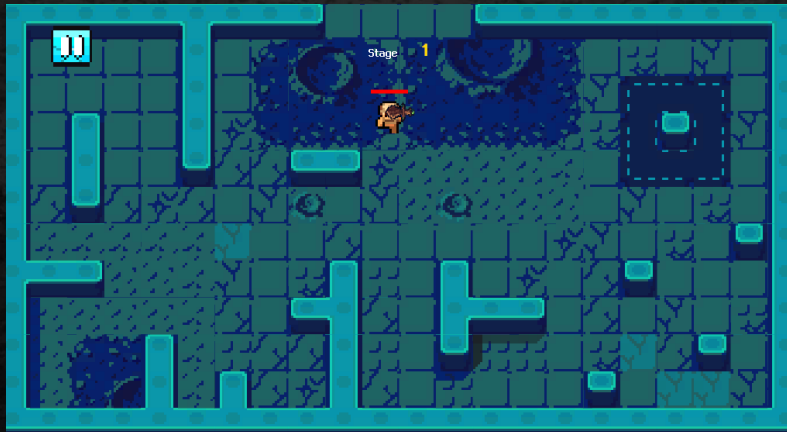
# 03 구현 기능 사항

## 스테이지 클리어

```
public void DrawMap(bool IsBoss)
{
    if (!IsBoss)
    {
        for (int i = 1; i < 24; i += 3)
        {
            currentTiles.Add(SelectedMap(i - 1, i + 2));
        }
    }
    else currentTiles.Add(Instantiate(SelectedMap(24, 25)));
    currentTiles.Add(Instantiate(mapEdge));
    currentTiles.Add(Instantiate(mapDoor));
}
```

참조 1개

```
public void DoorOpen()
{
    Destroy(currentTiles[currentTiles.Count - 1]);
}
```



스테이지 클리어

문에 대한 프리팹을 리스트  
마지막에 저장



스테이지 클리어 시



문을 제거하는 함수 호출  
(마지막 인덱스 제거)



문 제거

# 03 구현 기능 사항

## 스킬과 업그레이드 시스템

```
private void ApplySkillEffect(AbilityTable skill)
{
    PlayerStats player = FindObjectOfType<PlayerStats>();

    switch (skill.skillType)
    {
        case DesignEnums.SkillType.DamageUp: // 공격력 증가
            player.AttackDamage += skill.Value;
            break;
        case DesignEnums.SkillType.AttackSpeedUp: // 공격 속도 증가
            player.AttackSpeed += skill.Value;
            break;
        case DesignEnums.SkillType.MoveSpeedUp: // 이동속도 증가
            player.MoveSpeed += skill.Value;
            break;
        case DesignEnums.SkillType.HeadShot:
            player.HeadShotRate += skill.Value; // 헤드샷 확률 증가
            break;
        case DesignEnums.SkillType.Evasion:
            player.Evasionrate += skill.Value; // 회피를 증가
            break;
        case DesignEnums.SkillType.ExtraProjectile: // 발사체 증가
            player.ExtraProjectiles += 1;
            break;
    }
}
```

DataManager



모든 업그레이드 스킬 데이터 접근



Level UP!

스킬 정보 리스트를 랜덤하게 3개를 출력



플레이어가 스킬 선택 시 스킬 타입을 검사 후  
해당 업그레이드 스킬을 플레이어 스탯 적용



# 04 트러블 슈팅

공격을 한번만 한다...

key	Name	Grade	Delay
int	string	Enum<Grade>	float
ID	이름	등급	공격속도
1001	WoodSword	Common	1
1002	SkeltonSword	Rare	1.1
1003	IronSword	Unique	1.3
1004	LavaSword	Legendary	1.5
1005	AquaSword	Artifact	1.8
2001	WoodAxe	Common	0.7
2002	SkeltonAxe	Rare	1
2003	IronAxe	Unique	1.2
2004	LavaAxe	Legendary	1.4
2005	AquaAxe	Artifact	1.7
3001	WoodHammer	Common	0.5
3002	SkeltonHammer	Rare	0.65
3003	IronHammer	Unique	0.8
3004	LavaHammer	Legendary	1.1
3005	AquaHammer	Artifact	1.5
4001	WoodDagger	Common	1.4
4002	SkeltonDagger	Rare	1.6
4003	IronDagger	Unique	1.8
4004	LavaDagger	Legendary	2.1
4005	AquaDagger	Artifact	2.4
5001	WoodArrow	Common	0.9
5002	SkeltonArrow	Rare	1
5003	IronArrow	Unique	1.2
5004	LavaArrow	Legendary	1.4
5005	AquaArrow	Artifact	1.7
6001	WoodStamp	Common	0.9
6002	SkeltonStamp	Rare	1
6003	IronStamp	Unique	1.2
6004	LavaStamp	Legendary	1.4
6005	AquaStamp	Artifact	1.7
7001	WoodScythe	Common	1
7002	SkeltonScythe	Rare	1.1
7003	IronScythe	Unique	1.3
7004	LavaScythe	Legendary	1.5
7005	AquaScythe	Artifact	1.8
9001	SniperWeapon	Common	0.7
9002	SoliderWeapon	Common	1
9003	BodyWeapon	Common	1
9004	GrenadierWeapon	Common	0.5

```
public override void HandleAttackDelay()
{
    if (_weaponHandler == null)
        return;
    if (timeSinceLastAttack <= (1f / (_weaponHandler.Delay)))
    {
        timeSinceLastAttack += Time.deltaTime;
    }
    if (isAttacking && timeSinceLastAttack > (1f / (_weaponHandler.Delay
        * gameManager.player.statHandler.AttackSpeed)))
    {
        timeSinceLastAttack = 0;
        Attack();
    }
}
```

1001, 2001, 3001...7001 인덱스에 해당되는 무기가  
한번밖에 던지지 않는 문제를 발견

해결 방안 제시

1. 1번대 키에 해당되는 무기만 발생한 문제라  
서 Json 데이터에 빈 데이터 삽입 (X)
2. Bullet을 불러오는 스프라이트 인덱스 값이  
잘못 되었다. (처음엔 던지니 X)
3. 코드에 문제가 있다. ( O )

# 04 트러블 슈팅

공격을 한번만 한다...

key	Name	Grade	Delay
int	string	Enum<Grade>	float
ID	이름	등급	공격속도
1001	WoodSword	Common	1
1002	SkeltonSword	Rare	1.1
1003	IronSword	Unique	1.3
1004	LavaSword	Legendary	1.5
1005	AquaSword	Artifact	1.8
2001	WoodAxe	Common	0.7
2002	SkeltonAxe	Rare	1
2003	IronAxe	Unique	1.2
2004	LavaAxe	Legendary	1.4
2005	AquaAxe	Artifact	1.7
3001	WoodHammer	Common	0.5
3002	SkeltonHammer	Rare	0.65
3003	IronHammer	Unique	0.8
3004	LavaHammer	Legendary	1.1
3005	AquaHammer	Artifact	1.5
4001	WoodDagger	Common	1.4
4002	SkeltonDagger	Rare	1.6
4003	IronDagger	Unique	1.8
4004	LavaDagger	Legendary	2.1
4005	AquaDagger	Artifact	2.4
5001	WoodArrow	Common	0.9
5002	SkeltonArrow	Rare	1
5003	IronArrow	Unique	1.2
5004	LavaArrow	Legendary	1.4
5005	AquaArrow	Artifact	1.7
6001	WoodStamp	Common	0.9
6002	SkeltonStamp	Rare	1
6003	IronStamp	Unique	1.2
6004	LavaStamp	Legendary	1.4
6005	AquaStamp	Artifact	1.7
7001	WoodScythe	Common	1
7002	SkeltonScythe	Rare	1.1
7003	IronScythe	Unique	1.3
7004	LavaScythe	Legendary	1.5
7005	AquaScythe	Artifact	1.8
9001	SniperWeapon	Common	0.7
9002	SoliderWeapon	Common	1
9003	BodyWeapon	Common	1
9004	GrenadierWeapon	Common	0.5

```
public override void HandleAttackDelay()
{
    if (_weaponHandler == null)
        return;
    ① if (timeSinceLastAttack <= (1f / (_weaponHandler.Delay)))
    {
        timeSinceLastAttack += Time.deltaTime;
    }
    ③ if (isAttacking && timeSinceLastAttack > (1f / (_weaponHandler.Delay
        * gameManager.player.statHandler.AttackSpeed)))
    {
        timeSinceLastAttack = 0;
        Attack();
    }
}
```

③ 해당 조건문  
영원히 false  
공격하지 않음

② if (timeSinceLastAttack <= (1f / \_weaponHandler.Delay )

0.66666

0.65

timeSinceLastAttack이 더 이상 증가하지 않음



# 04 트러블 슈팅

## 공격을 한번만 한다...

key	Name	Grade	Delay
int	string	Enum<Grade>	float
ID	이름	등급	공격속도
1001	WoodSword	Common	1
1002	SkeltonSword	Rare	1.1
1003	IronSword	Unique	1.3
1004	LavaSword	Legendary	1.5
1005	AquaSword	Artifact	1.8
2001	WoodAxe	Common	0.7
2002	SkeltonAxe	Rare	1
2003	IronAxe	Unique	1.2
2004	LavaAxe	Legendary	1.4
2005	AquaAxe	Artifact	1.7
3001	WoodHammer	Common	0.5
3002	SkeltonHammer	Rare	0.65
3003	IronHammer	Unique	0.8
3004	LavaHammer	Legendary	1.1
3005	AquaHammer	Artifact	1.5
4001	WoodDagger	Common	1.4
4002	SkeltonDagger	Rare	1.6
4003	IronDagger	Unique	1.8
4004	LavaDagger	Legendary	2.1
4005	AquaDagger	Artifact	2.4
5001	WoodArrow	Common	0.9
5002	SkeltonArrow	Rare	1
5003	IronArrow	Unique	1.2
5004	LavaArrow	Legendary	1.4
5005	AquaArrow	Artifact	1.7
6001	WoodStamp	Common	0.9
6002	SkeltonStamp	Rare	1
6003	IronStamp	Unique	1.2
6004	LavaStamp	Legendary	1.4
6005	AquaStamp	Artifact	1.7
7001	WoodScythe	Common	1
7002	SkeltonScythe	Rare	1.1
7003	IronScythe	Unique	1.3
7004	LavaScythe	Legendary	1.5
7005	AquaScythe	Artifact	1.8
9001	SniperWeapon	Common	0.7
9002	SoliderWeapon	Common	1
9003	BodyWeapon	Common	1
9004	GrenadierWeapon	Common	0.5

```
public override void HandleAttackDelay()  
{  
    if (_weaponHandler == null)  
        return;  
    if (timeSinceLastAttack <= (1f / (_weaponHandler.Delay  
        * gameManager.player.statHandler.AttackSpeed)))  
    {  
        timeSinceLastAttack += Time.deltaTime;  
    }  
    if (isAttacking && timeSinceLastAttack > (1f / (_weaponHandler.Delay  
        * gameManager.player.statHandler.AttackSpeed)))  
    {  
        timeSinceLastAttack = 0;  
        Attack();  
    }  
}
```

if (timeSinceLastAttack <= (1f / \_weaponHandler.Delay )



수정 후

if (timeSinceLastAttack <= (1f / (\_weaponHandler.Delay  
 \* gameManager.player.statHandler.AttackSpeed))

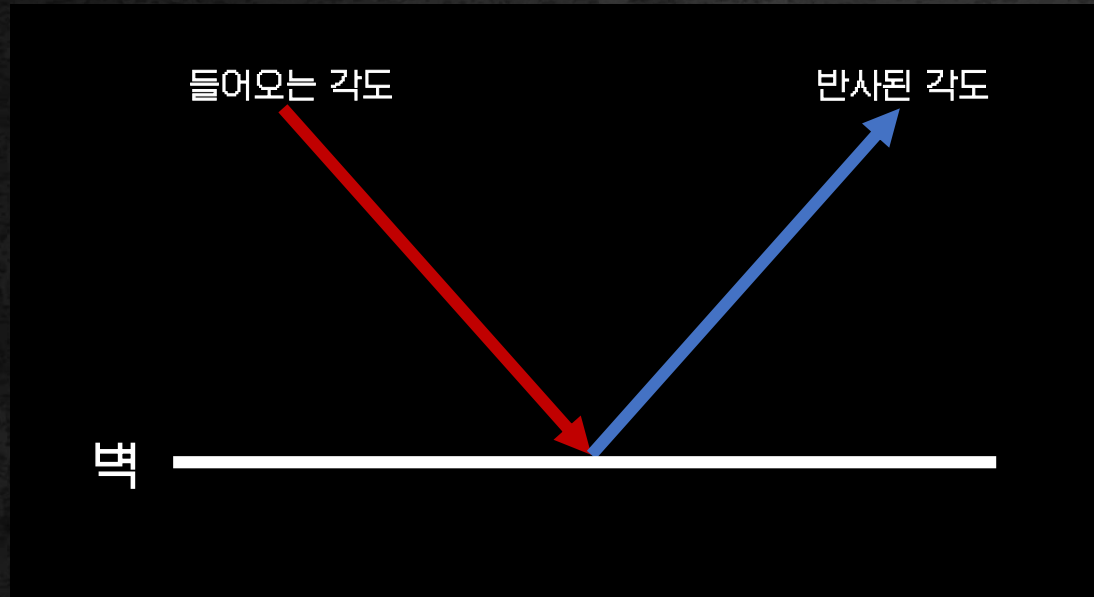
역시나 문제는 항상 내가 짠 코드에서 발생한다...



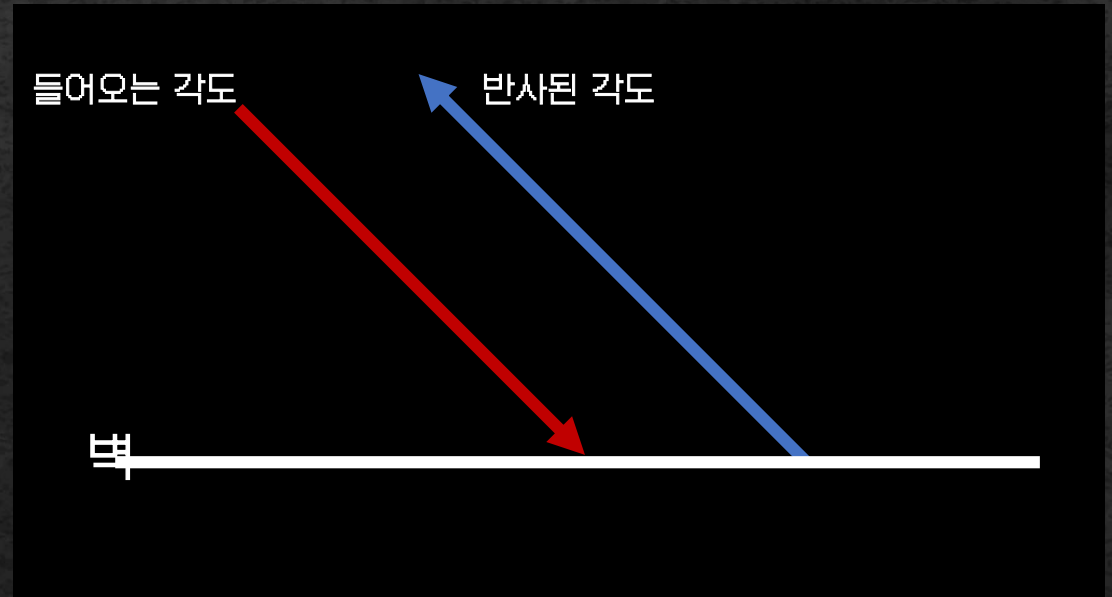
# 04 트러블 슈팅

## 투사체 반사 각도 계산

원래 의도



발생한 문제



발사체 충돌 시, 반사각을 정확하게 계산해야 했지만,  
초기 구현에서 반사 방향이 비정상적으로 설정됨.

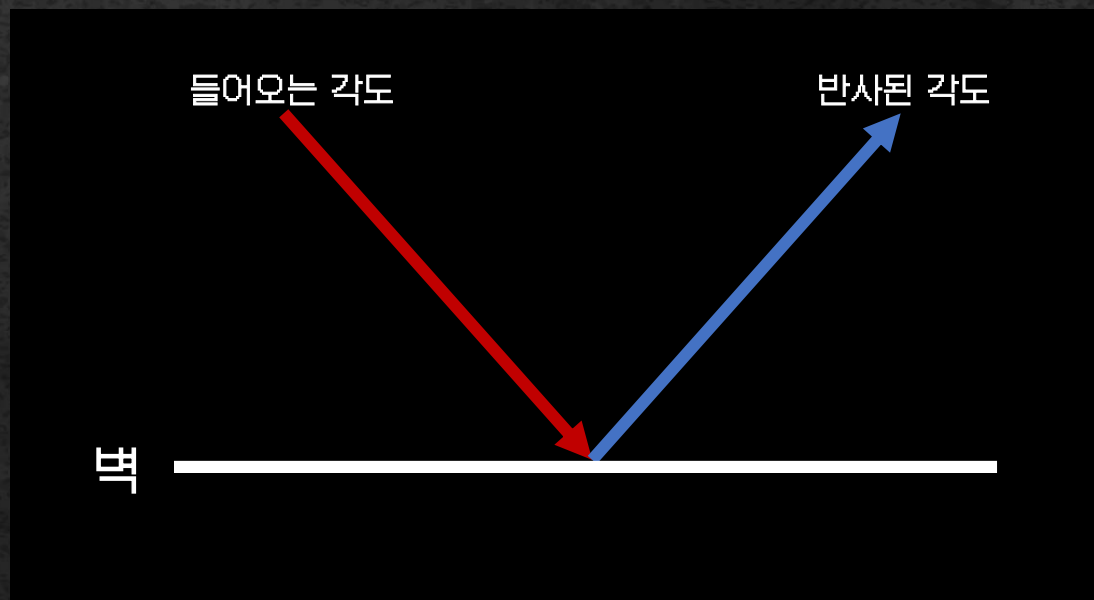
반사각을 계산하는 과정에서 충돌한 물체 표면의  
Normal Vector를 잘못 설정하며, 예상과 다른 방향으로 반사됨

# 04 트러블 슈팅

## 투사체 반사 각도 계산

```
private void OnTriggerEnter2D(Collider2D collision)
{
    // 충돌 표면의 법선 가져오기 (2D 전용)
    ContactPoint2D[] contacts = new ContactPoint2D[1];
    if (collision.TryGetComponent(out Rigidbody2D rb) && rb.GetContacts(contacts) > 0)
    {
        reflect_normal = contacts[0].normal; // 첫 번째 충돌 표면의 법선
    }
    else
    {
        // 대체 법선 (비정상적인 경우)
        reflect_normal = (transform.position - collision.transform.position).normalized;
    }
    ...
}
```

## 문제 해결



OnTriggerEnter2D에서 충돌한 물체의 표면의 Normal Vector를  
ContactPoint2D를 통해 새로 계산

Normal Vector를 기준으로 입사각과 같은 반사각을 가지도록 벡터 연산.

## 04 소감 및 각오

- **김태호** 팀워크와 협업의 중요성을 다시 한 번 깨닫게 되었고, 게임 개발의 복잡함을 실감하며 더욱 성장할 수 있었습니다. 앞으로는 더 나은 결과물을 위해 끊임없이 배우고 도전하겠습니다!
- **김성원** 이번 팀 프로젝트를 통해서 유니티에 대해 좀더 익숙해 질 수 있었고, 완성하지 못한 부분이 있어서 아쉬웠지만, 다음 프로젝트때는 도전과제까지 모두 달성할 수 있도록 노력하겠습니다.
- **김소연** 유니티 협업의 방식에 대해 알아가고, 게임 개발에 대한 이해를 높이는 시간이었습니다. 또한, 좋은 팀원분들과 함께 많이 배우고 성장한 5일이었습니다 감사합니다!^^♡
- **김영준** 오류가 발생하고, 해결하면 또 다른 오류가 생기면서 쉽게 해결되지 않을 때는 스트레스를 받기도 했습니다. 하지만 팀원들과 함께 해결해 나가면서 점점 길이 보였고, 협력의 중요성을 다시금 느꼈습니다. 앞으로도 팀원들과 잘 협업하여 재미있는 게임을 만들어 나가겠습니다.
- **이상훈** 각자의 역량을 발휘해 완성해 나가는 과정이 정말 값진 경험이었습니다. 함께했던 순간들이 소중한 추억이 되었고, 그 덕분에 더 의미 있는 결과물을 만들 수 있었습니다.



# etc 출처

활: <https://spartacodingclub.kr/blog/618cbaaf94295eb6f7fff0f2>

무기 에셋: <https://babysamurai.itch.io/super-epic-fantasy-weapons-pack>

UI 에셋: <https://wenrexa.itch.io/kit-nesia2>, <https://srtoasty.itch.io/ui-assets-pack-2>  
<https://mounirtohami.itch.io/pixel-art-gui-elements>

맵 디자인: <https://mattwalkden.itch.io/free-robot-warfare-pack>

아이템: <https://ghostpixxells.itch.io/pixel-mart>