

Unity Portfolio

Board Clash

선동운

01



게임 소개

게임 소개



제목 : Board Clash

장르 : 전략 시뮬레이션

개발 엔진 : Unity 2022.3.27

플랫폼 : Android

언어 : C#

개발 인원 : 1명

시연 영상 : <https://youtu.be/p0nPSkvPTIk>

GitHub : https://github.com/Seon-dongun/Unity_Project_BoardClash

02



기능 구현

플레이어 보드판 배치



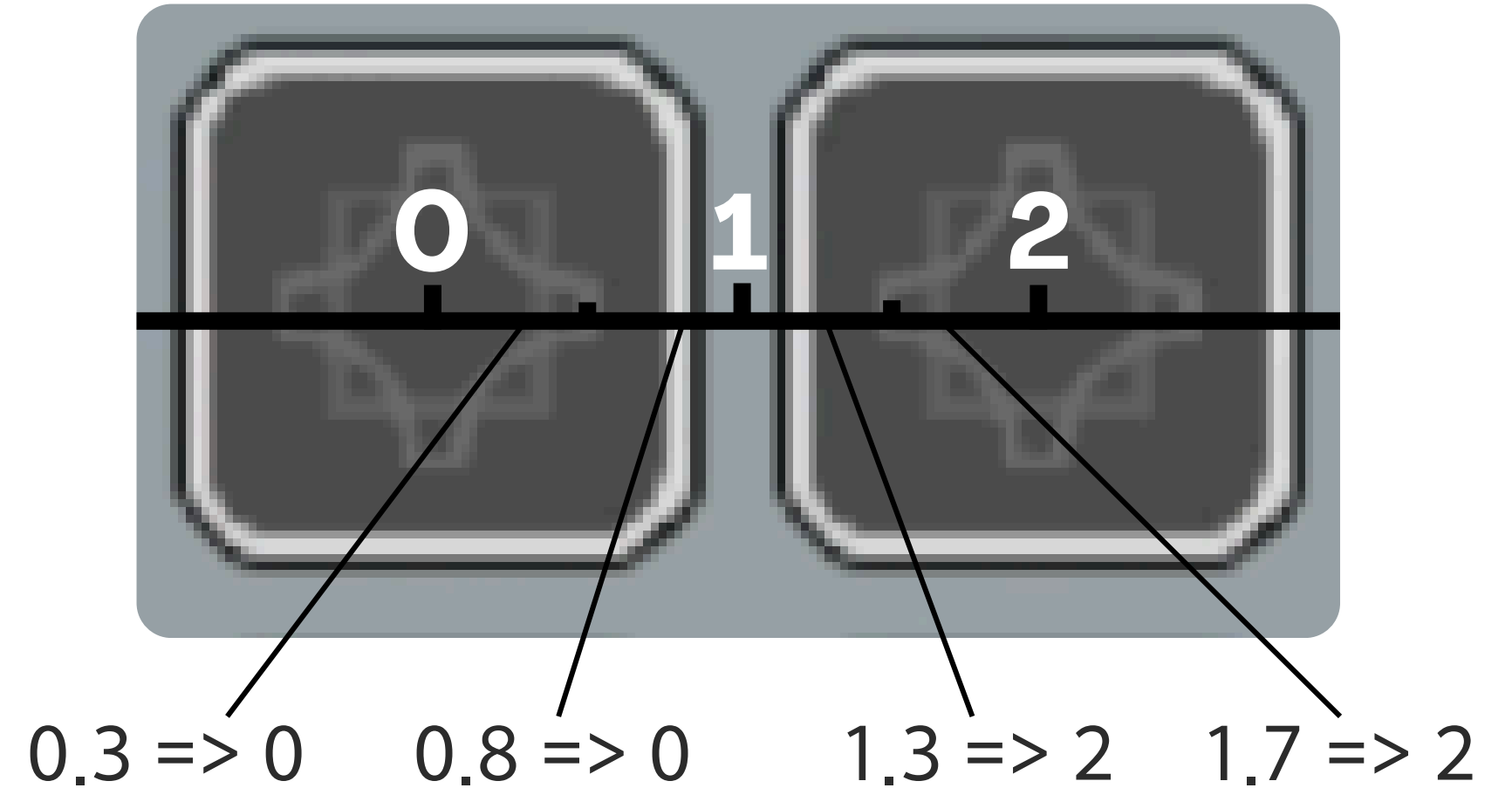
- 1-2-3-4 구역 드롭 => A칸 배치
- 5-6-7-8 구역 드롭 => B칸 배치
- 9-10-11-12 구역 드롭 => C칸 배치
- 13-14-15-16 구역 드롭 => D칸 배치

플레이어 보드판 배치

```
float newX, newY;
if ((int)transform.position.x % 2 == 0)
{
    if(transform.position.x < 0)
        newX = Mathf.Ceil(transform.position.x);
    else
        newX = Mathf.Floor(transform.position.x);
}
else
    newX = Mathf.Ceil(transform.position.x);

if ((int)transform.position.y % 2 == 0)
{
    if (transform.position.y < 0)
        newY = Mathf.Ceil(transform.position.y);
    else
        newY = Mathf.Floor(transform.position.y);
}
else
    newY = Mathf.Ceil(transform.position.y);
```

이미지 클릭 시 전체 코드 확인 가능



좌표 정수부 짝수

=> 내림(Floor)으로 짝수의 중앙 좌표에 맞춤

좌표 정수부 홀수

=> 올림(Ceil)으로 짝수의 중앙 좌표에 맞춤

이동 및 공격

```
void Update()
{
    if (Time.time - lastUpdateTime >= 1.0f)
    {
        if (gd.CurrentPlayerNumber == 0 || gd.CurrentEnemyNumber == 0)
            return;

        animator.SetBool("canAttack", false);
        GameObject nearestTarget = null;

        if (gameObject.CompareTag("Player"))
            nearestTarget = FindNearestTarget("Enemy");
        else if (gameObject.CompareTag("Enemy"))
            nearestTarget = FindNearestTarget("Player");

        if (IsTargetInArea(nearestTarget) == true)
        {
            animator.SetBool("canAttack", true);
            AttackTarget(nearestTarget);
        }
        else
            StartCoroutine(MoveWithDelay(nearestTarget));

        lastUpdateTime = Time.time;
    }
}
```

CharacterController.cs

이미지 클릭 시 전체 코드 확인 가능

- 모든 캐릭터는 1초에 한 번 이동 혹은 공격을 선택
- 상대 진영의 캐릭터 중 가장 가까운 캐릭터를 찾고, 가장 가까운 상대 진영의 캐릭터가 공격 범위 안에 있다면 공격
- 공격 범위 안에 상대가 없다면 이동

탐색

```
float dx = targetPosition.x - currentPosition.x;
float dy = targetPosition.y - currentPosition.y;
float distanceSquared = dx * dx + dy * dy;
if (distanceSquared < minDistanceSquared)
{
    minDistanceSquared = distanceSquared;
    nearestTarget = target;
}
```

CharacterController.cs

이미지 클릭 시 전체 코드 확인 가능



- 이동 시 가장 가까운 대상을 찾는 FindNearestTarget 함수 실행
- 상대 오브젝트들과의 직선 상 거리를 계산해 가장 가까운 대상 탐색

이동(추적)



GetMovePosition 함수를 통해 상하좌우 발판 중 target과 가장 가까운 발판으로 이동

이동(추적)

```
IEnumerator MoveCoroutine(Vector3 startPosition, Vector3 targetPosition)
{
    float moveTime = 0.8f;
    float elapsedTime = 0.0f;

    animator.SetBool("isMoving", true);

    while (elapsedTime < moveTime)
    {
        elapsedTime += Time.deltaTime;
        float t = Mathf.Clamp01(elapsedTime / moveTime);
        transform.position = Vector3.Lerp(startPosition, targetPosition, t);
        yield return null;
    }

    animator.SetBool("isMoving", false);

    transform.position = targetPosition;
}
```

CharacterController.cs

이미지 클릭 시 전체 코드 확인 가능

코루틴 함수를 통해 경과한 시간 비율을
활용하여 시작 지점에서 목표 지점까지 의
부드러운 이동 구현

공격



<Type 1. 단일 근접 공격>



<Type 2. 광역 근접 공격>



<Type 3. 단일 원거리 공격>

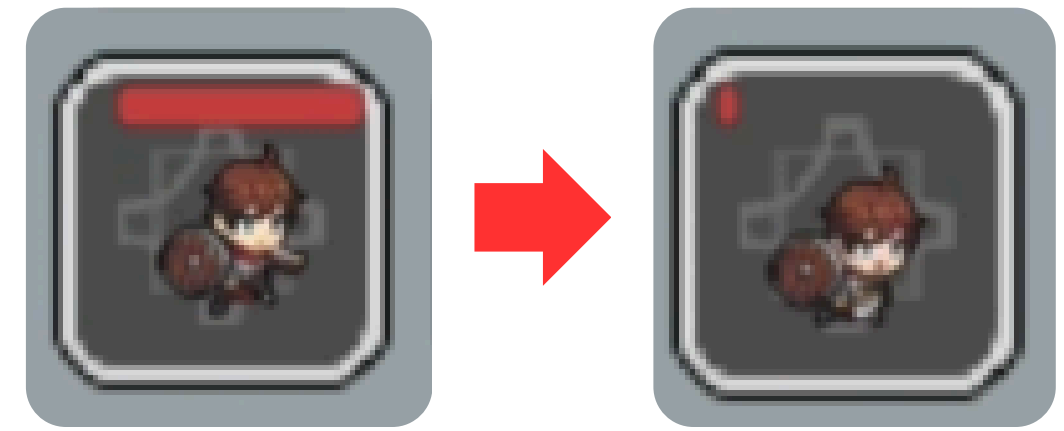
- 공격 범위 내에 목표 대상이 있는지 판단 후, 범위 안에 있다면 대상에게 단일 공격 혹은 광역 공격을 수행
- 피격 대상은 체력이 0이하로 떨어지면 사망
- 캐릭터 사망 시 해당 캐릭터 비활성화, 해당 캐릭터 진영 인원 수 감소

HP UI

```
public void AdjustHpBar()
{
    float hpRatio = (float)currentHealth / maxHealth;
    hpBarSlider.value = hpRatio;
}
```

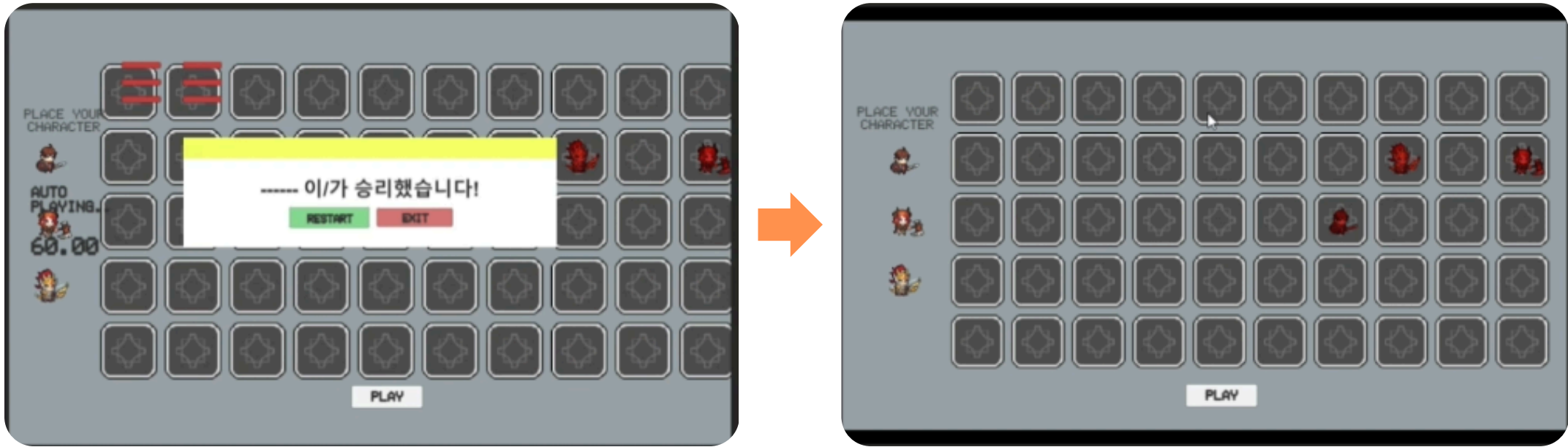
CharacterController.cs

이미지 클릭 시 전체 코드 확인 가능

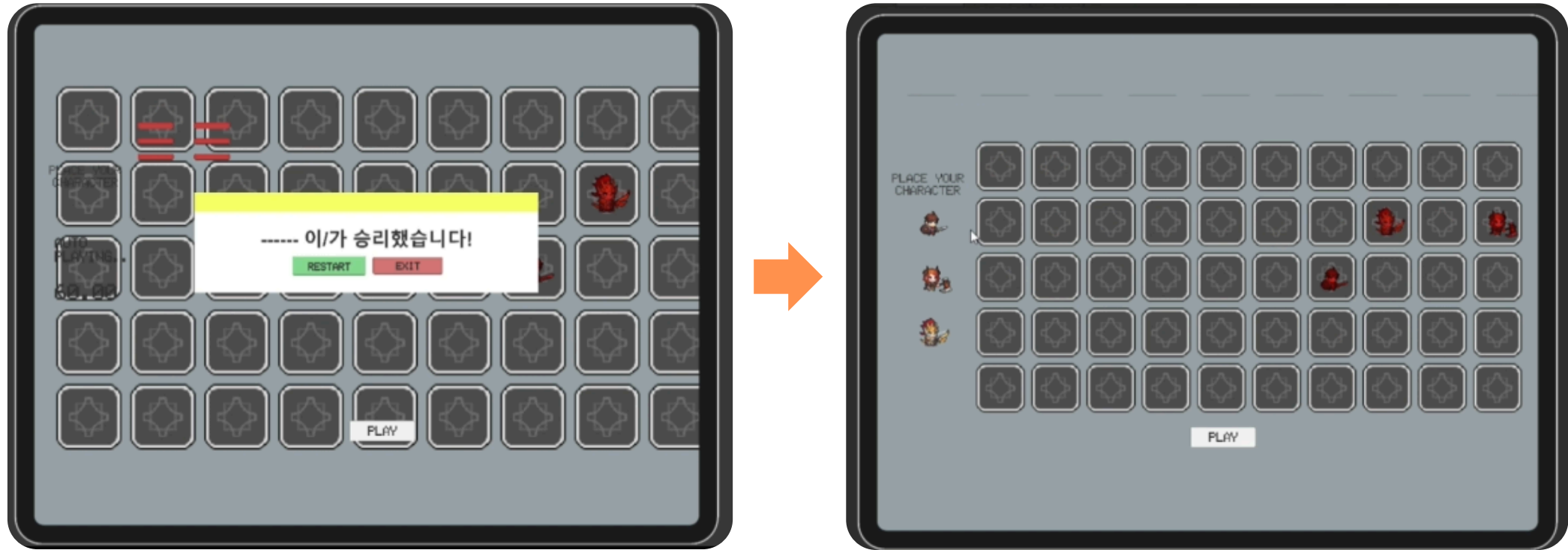


Slider 오브젝트로 제작된 HpBar에 현재 체력과 최대 체력의 비율을 적용하여,
UI에 현재 체력 수치를 표시

화면 비율 조절



게임 시작 시 16:9 비율을 유지하도록 구현



화면 비율 조절

```
public class CameraResolution : MonoBehaviour
{
    void Awake()
    {
        Camera camera = GetComponent<Camera>();
        Rect rect = camera.rect;
        float scaleheight = ((float)Screen.width / Screen.height) / ((float)16 / 9);
        float scalewidth = 1f / scaleheight;
        if(scaleheight < 1)
        {
            rect.height = scaleheight;
            rect.y = (1f - scaleheight) / 2f;
        }
        else
        {
            rect.width = scalewidth;
            rect.x = (1f - scalewidth) / 2f;
        }
        camera.rect = rect;
    }
}
```

CameraResolution.cs

이미지 클릭 시 전체 코드 확인 가능

사용자 화면에 맞게 카메라 비율을 조정하여, 게임 플레이 화면 사이즈 비율 조절