

# 2D TOWER DEFENCE

---

CHOI HAN GEUL  
최한글

# INDEX

1. 게임 소개 및 진행 방식

2. 게임 UI

3. 몬스터 정보

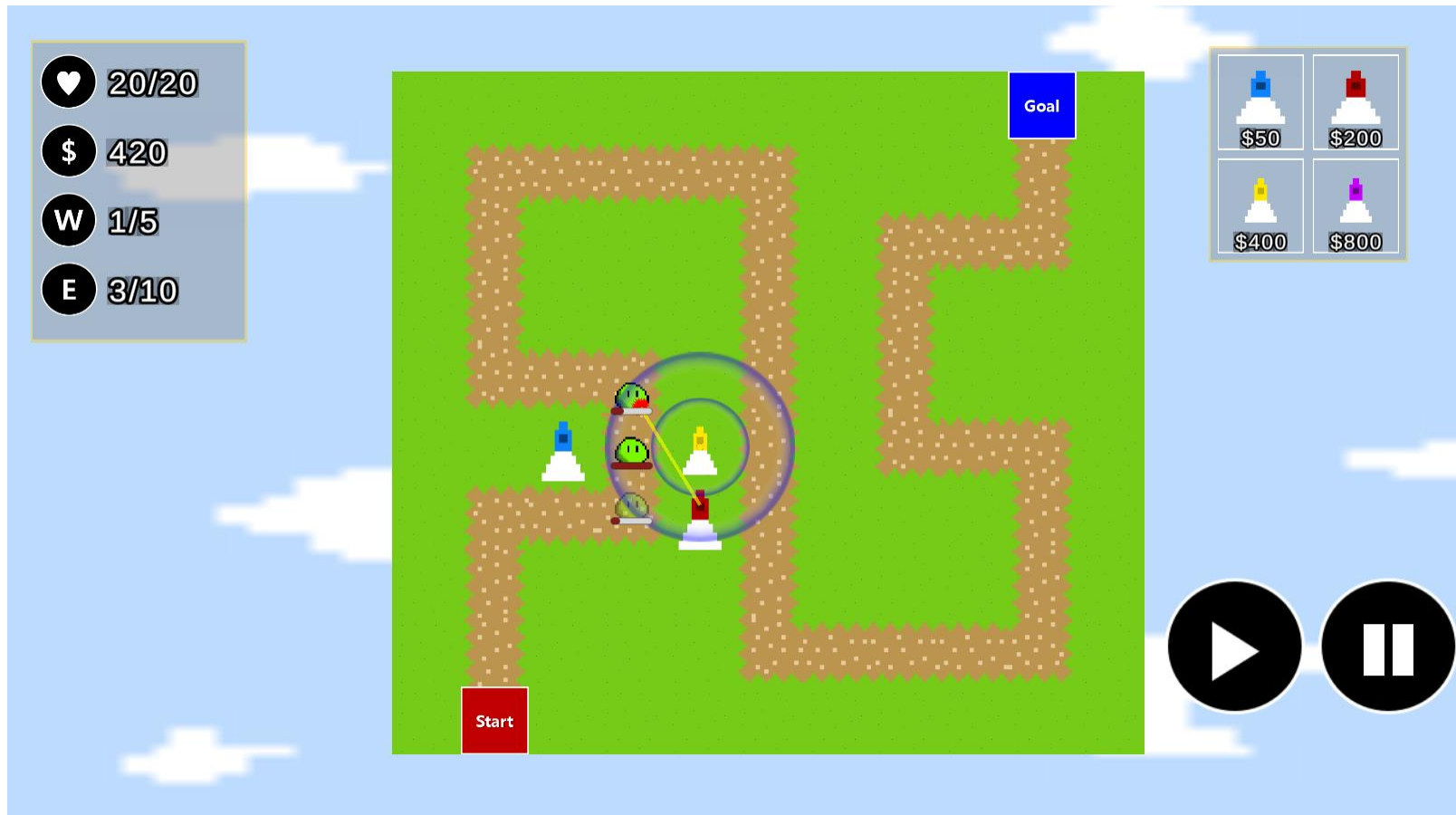
4. 타워 정보

5. 기능 구현 방법



## 1. 게임 소개

- 2D 그래픽 기반의 게임으로 타워를 건설하여 적들이 도착지점에 이르기 전까지 해치워 5개의 웨이브를 버티는 방식으로 진행



## **2. 진행 방식**

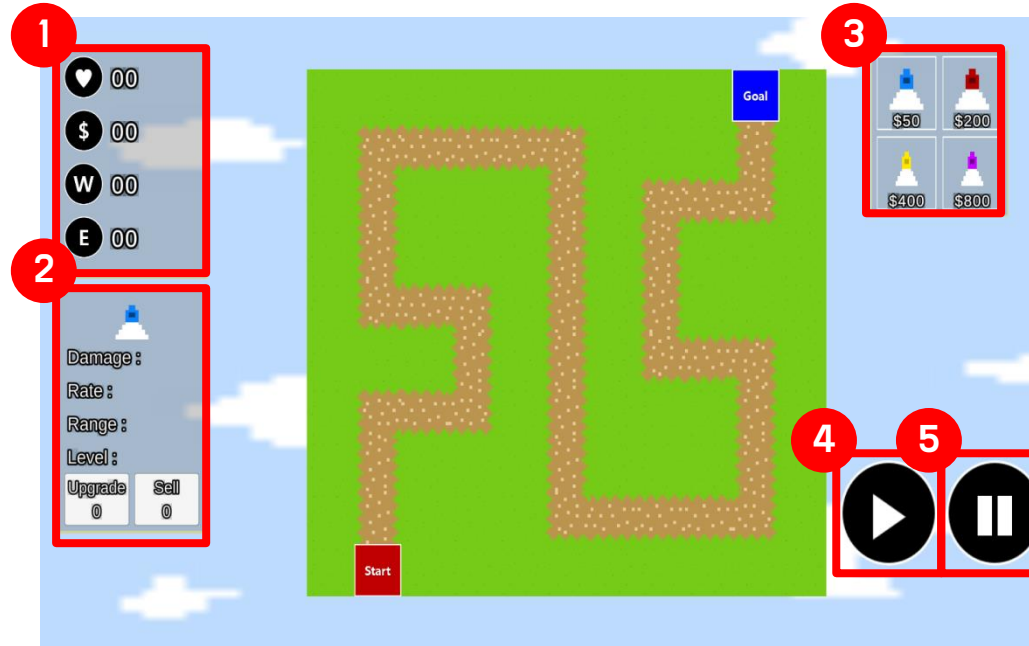
- 스테이지마다 START 지점에서 생성되는 적이 GOAL 지점에 통과하지 못하도록 타워를 건설하여 방어
- 5개의 스테이지로 구성되어 있으며 처치한 적에게 획득한 골드를 통하여 타워 추가 생성 및 업그레이드로 스테이지가 진행 될 때 마다 강해지는 몬스터를 저지
- 게임 시작 시 주어지는 100골드와 20개의 라이프를 통하여 적들을 저지하고 GOAL을 지켜내면 승리

### 3. 게임 UI - 게임 시작 화면



No	기능	설명
1	게임 시작	게임 스테이지로 이동하는 버튼
2	게임 종료	게임 종료 버튼

### 3. 게임 UI - 인 게임 화면



No	기능	설명
1	플레이어 상태창	플레이어의 상태를 확인할 수 있는 UI (라이프 / 자금 / 현재 웨이브 / 남은 적 숫자)
2	타워 정보	선택된 타워 정보 확인 및 타워 업그레이드와 판매 가능
3	타워 설치 버튼	해당 버튼을 누르면 누른 타워 설치 가능
4	웨이브 시작 버튼	필드에 적이 없을 경우 클릭 시 웨이브 시작
5	정지 버튼	게임 정지가 가능하며 RESTART / CONTINUE / EXIT 기능 사용 가능

## 4. 몬스터 정보



### ENEMY 1

체력	4
처치 시 획득 골드량	20
몬스터 숫자	10



### ENEMY 2

체력	10
처치 시 획득 골드량	30
몬스터 숫자	15



### ENEMY 3

체력	15
처치 시 획득 골드량	40
몬스터 숫자	20



### ENEMY 4

체력	25
처치 시 획득 골드량	50
몬스터 숫자	25



### ENEMY 2

체력	130
처치 시 획득 골드량	60
몬스터 숫자	30

## 5. 타워 정보



**TOWER 1 (CANNON)**

레벨	1	2	3
데미지	1	1.5	2
골드 (BUILD/UPGRADE)	50	100	150
골드 (SELL)	25	75	100



**TOWER 2 (LASER)**

레벨	1	2	3
데미지	1.5	3	5
골드 (BUILD/UPGRADE)	200	350	400
골드 (SELL)	100	200	220



## 5. 타워 정보



**TOWER 3 (SLOW)**

레벨	1	2	3
적 이동속도 감소	20%	30%	40%
골드 (BUILD/UPGRADE)	400	600	1000
골드 (SELL)	200	400	700



**TOWER 4 (BUFF)**

레벨	1	2	3
타워 공격력 증가	30%	70%	100%
골드 (BUILD/UPGRADE)	800	1400	2000
골드 (SELL)	400	800	1300

## 6. 기능 구현 방법 - 게임 시작 화면

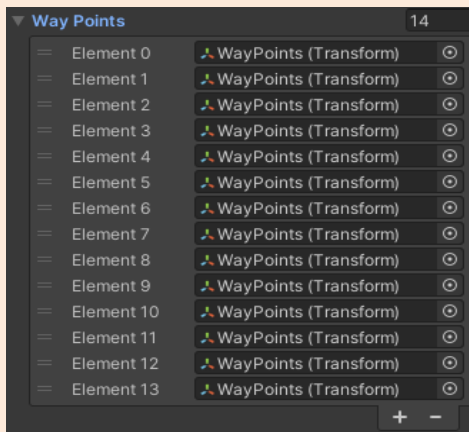
START : LoadScene 을 사용하여 클릭 시 GAMESTAGE로 이동

EXIT : Application.Quit(); 사용



## 6. 기능 구현 방법 - 적

- 적이 이동하는 구간에 WAYPOINT를 미리 지정하여 지정된 WAYPOINT 로 이동
- START 지점에서 생성되어 GOAL 지점에 도달 시 플레이어 라이프 감소



\* 지정된 WAYPOINT 경로

```
public void Setup(EnemySpawner enemySpawner, Transform[] wayPoints)
{
    movement = GetComponent<Movement>();
    this.enemySpawner = enemySpawner;

    // 적 이동 경로 WayPoints 정보 설정
    wayPointCount = wayPoints.Length;
    this.wayPoints = new Transform[wayPointCount];
    this.wayPoints = wayPoints;

    // 적의 위치를 첫번째 wayPoint 위치로 설정
    transform.position = wayPoints[currentIndex].position;

    // 적 이동, 목표 지점 설정, 적 코루틴 함수
    StartCoroutine(OnMove());
}
```

```
IEnumerator OnMove()
{
    NextMoveTo();

    while (true)
    {
        if (Vector3.Distance(transform.position,
            wayPoints[currentIndex].position) < 0.02f * movement.MoveSpeed)
        {
            NextMoveTo();
        }

        yield return null;
    }
}
```

```
void NextMoveTo()
{
    if(currentIndex < wayPointCount - 1)
    {
        transform.position = wayPoints[currentIndex].position;
        currentIndex++;
        Vector3 direction = (wayPoints[currentIndex].position - transform.position).normalized;
        movement.MoveTo(direction);
    }
    else
    {
        gold = 0;
        OnDie(EnemyDestroyType.Arrive);
    }
}
```

\* 기능 상세

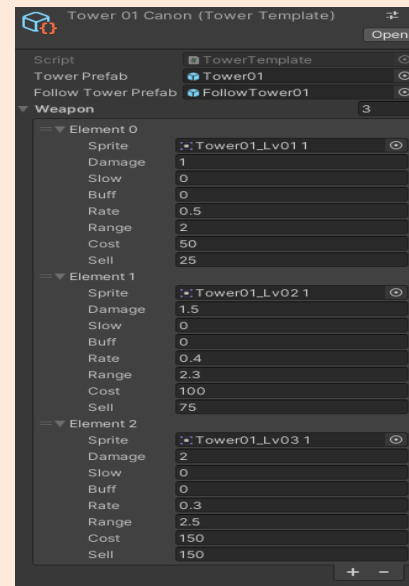
## 6. 기능 구현 방법 - 타워

- 필드내 TILE 이 TAG 되어있는 곳에 설치 가능
- 타워 건설 시 필요한 골드만큼 플레이어 골드 감소
- CreateAssetMenu 를 사용하여 각 타워 에셋 생성 및 능력치 설정

```
[CreateAssetMenu]
public class TowerTemplate : ScriptableObject
{
    public GameObject towerPrefab;
    public GameObject followTowerPrefab;
    public Weapon[] weapon;

    [System.Serializable]
    public struct Weapon
    {
        public Sprite sprite;
        public float damage;
        public float slow;
        public float buff;
        public float rate;
        public float range;
        public int cost;
        public int sell;
    }
}
```

\* 기능 상세



\* TOWER 1 ASSET

## 6. 기능 구현 방법 - 적 공격

- 발사체 스크립트를 만들어 OnTriggerEnter 함수를 사용하여 target 이 맞으면 변수로 지정한 damage 만큼 감소하도록 연동

```
private void OnTriggerEnter2D(Collider2D collision)
{
    if (!collision.CompareTag("Enemy")) return;
    if (collision.transform != target) return;

    collision.GetComponent<EnemyHP>().TakeDamage(damage);
    Destroy(this.gameObject);
}
```

\* 기능 상세

# **End Of Document**

---

**감 사 합 니 다**