

(24.06.03~24.06.11)
포트폴리오 프로젝트

고양이 뽑기(Cat Game) 개발 보고서

김 호 현
rlaghgus444@naver.com

CONTENTS

00 개발 기간

01 게임 개요

02 개발 의도

- 고양이 뽑기 개발 의도
- 핵심 기능 키워드

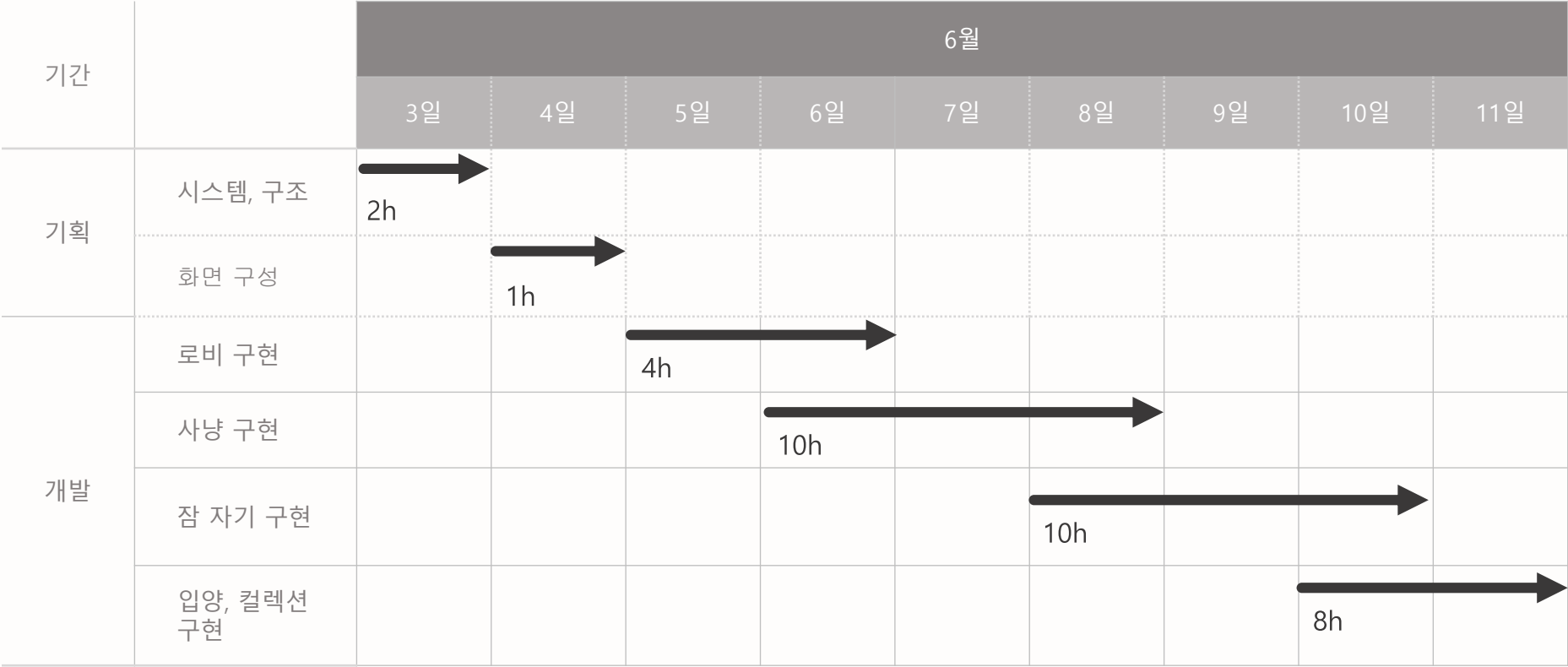
03 게임 전체 구조도

- 게임 구조도
- 코드 흐름도

04 마치며

【 0. 개발 기간 】

00 개발 기간
Gantt Chart



[1. 게임 개요]

01 게임 개요

Game Outline

고양이 뽑기란?

플레이어가 고양이의 보호자가 되어 입양하고 육성하는 게임



미니 게임을 통한 재화 획득



재화를 소모하여 고양이를
입양



입양한 고양이로 미니 게임
을 진행하여 호감도 상승

게임 명칭	고양이 뽑기 (Cat Game)
장르	수집 시뮬레이션, 퍼즐
플랫폼	Android Windows
게임 목적	고양이의 수집 및 육성

【 2. 개발 의도 】

02 개발 의도(목표)

Intention

고양이 뽑기 개발 의도

- 1) 첫 번째 프로젝트
- 2) 유니티 엔진과 게임 개발에 대한 지식과 숙련도가 부족한 상태에서의 개발 경험
- 3) 수집 게임의 재미 요소인 확률형 아이템 뽑기(가차) 시스템과 육성 시스템의 구현

핵심 기능 키워드

고양이 뽑기의 핵심 기능 키워드는 다음과 같다.

입양

호감도

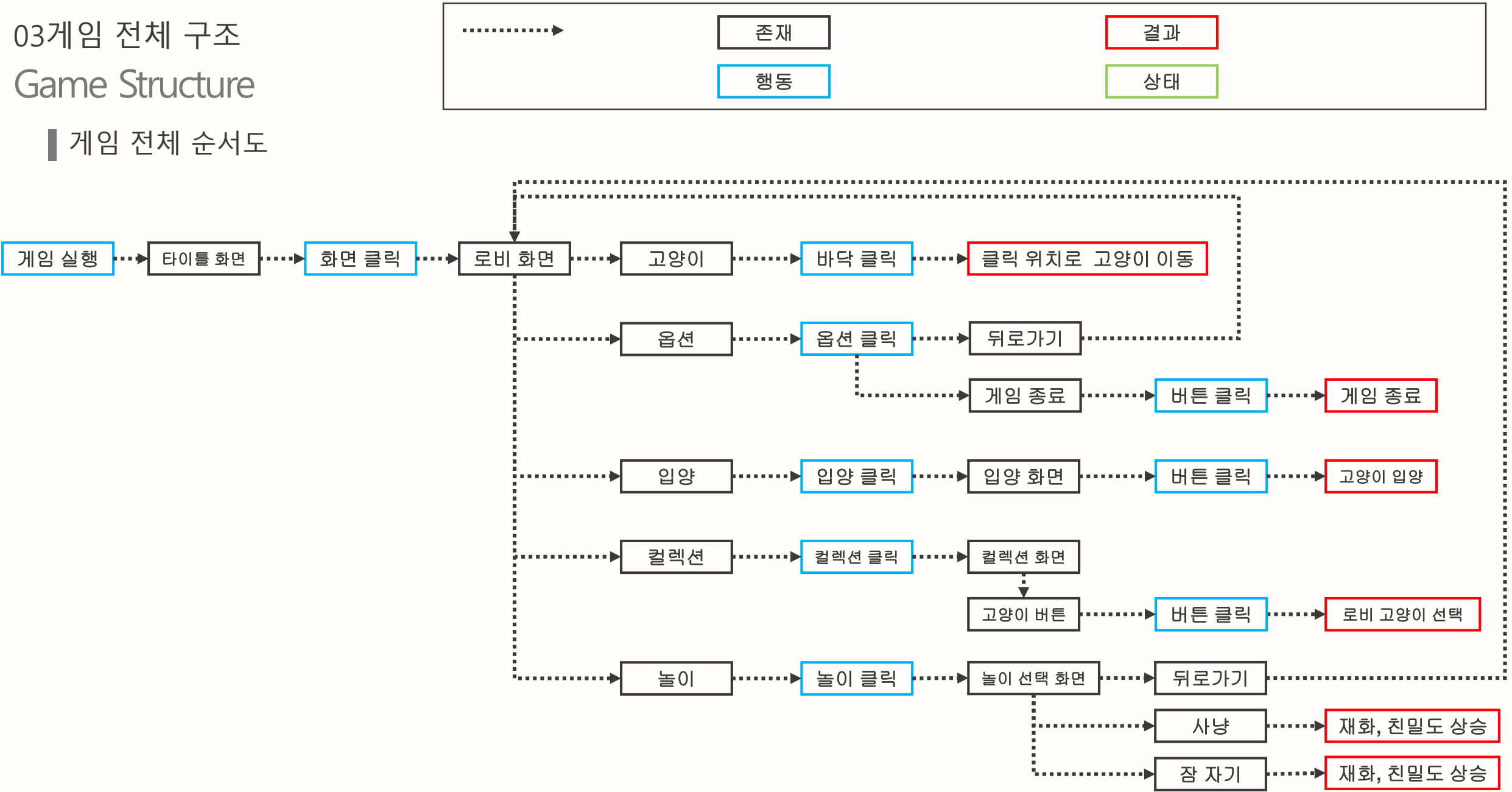
컬렉션

미니 게임

【 3. 게임 전체 구조 】

03게임 전체 구조
Game Structure

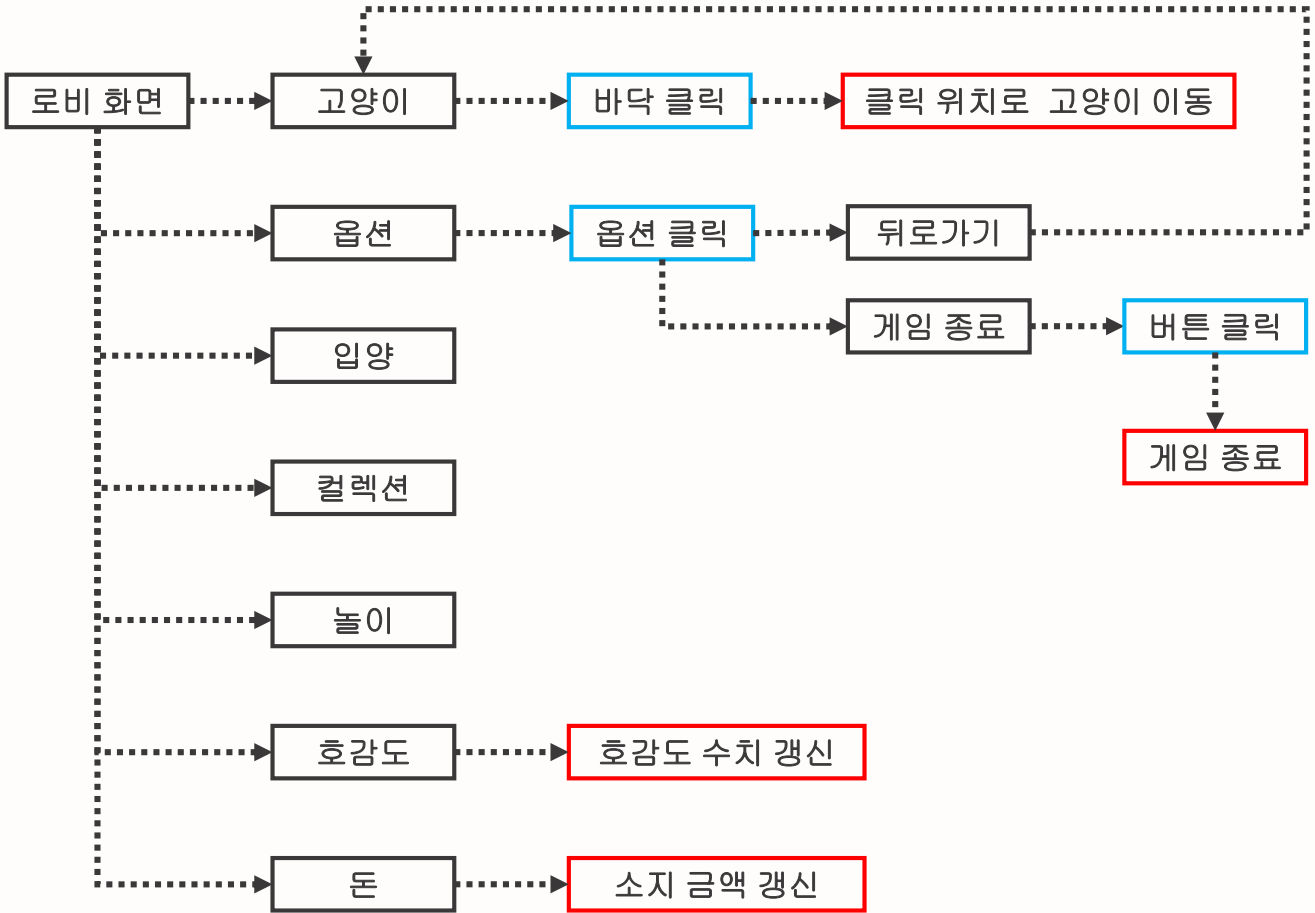
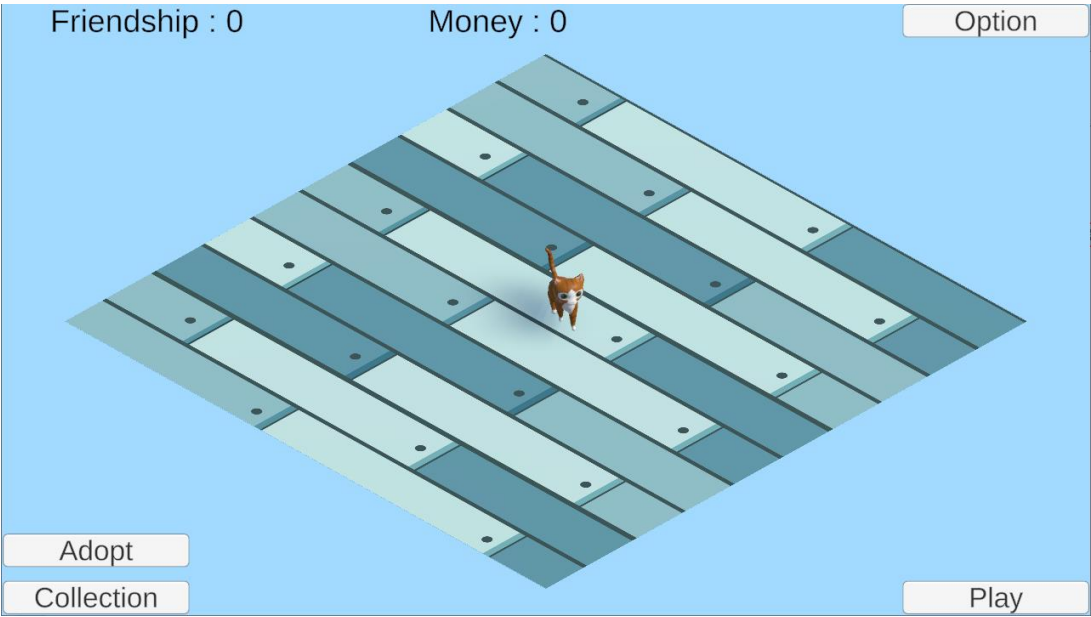
■ 게임 전체 순서도



03게임 전체 구조

Game Structure

로비 순서도



03게임 전체 구조
Game Structure

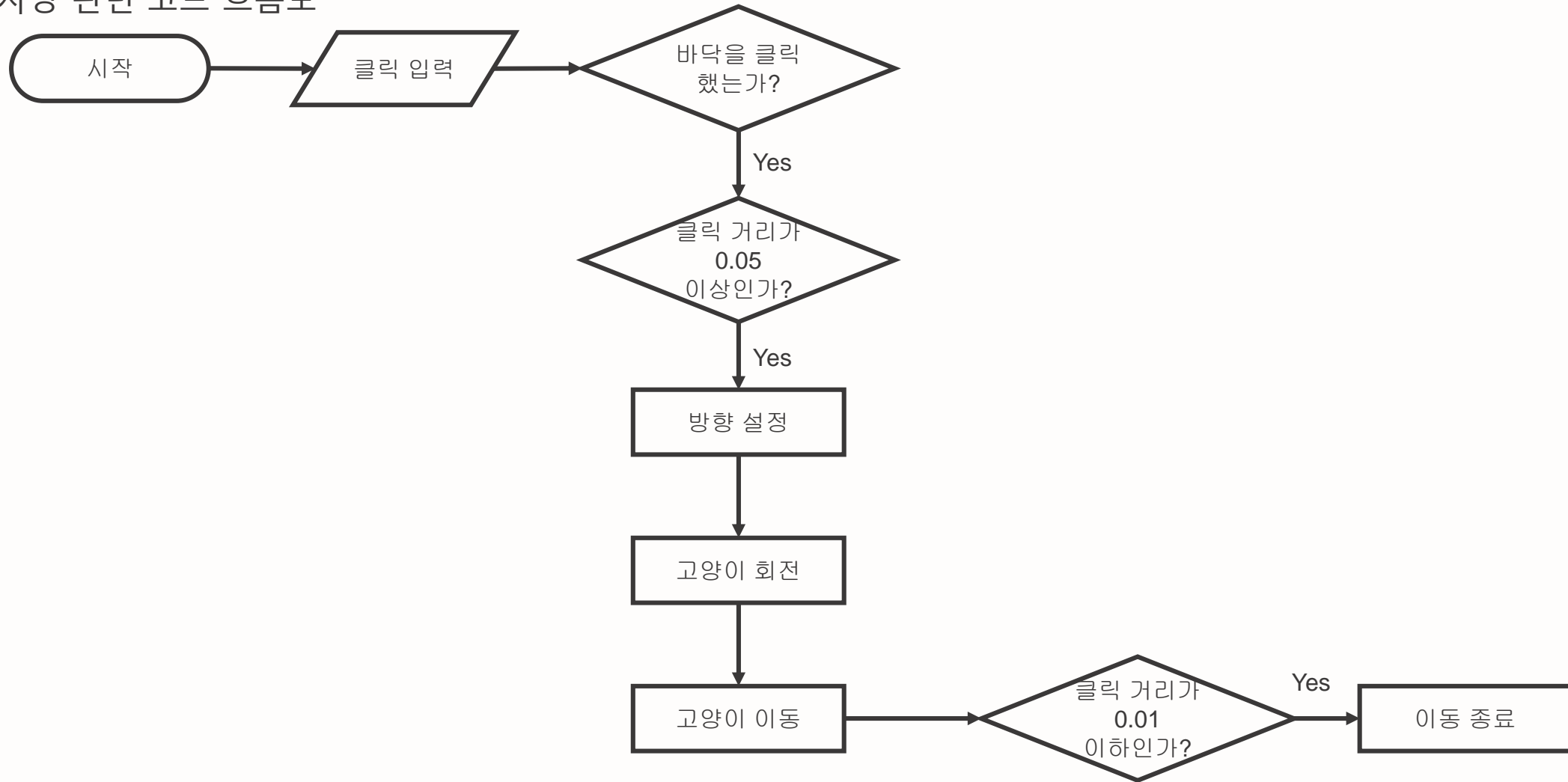
사냥 순서도



03게임 전체 구조 Game Structure

고양이 이동

■ 사냥 관련 코드 흐름도



03게임 전체 구조

Game Structure

■ 사냥 관련 코드

```
void SetDestination()
{
    // 마우스 클릭 위치로 레이캐스트
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    RaycastHit hit;

    // 레이캐스트 위치를 목적으로 설정
    if (Physics.Raycast(ray, out hit))
    {
        destPos = hit.point;

        // 목적지와 거리 멀면(0.05f 초과) 애니메이션 값 지정
        if (Vector3.Distance(transform.position, destPos) > 0.05f)
        {
            anim.SetBool("isSitting", false);
            anim.SetBool("isMoving", true);
        }

        // 클릭된 콜라이더가 Floor이고 움직이는 중이면 방향설정, 방향을 바라보게함
        if (hit.collider.name == "Floor" && anim.GetBool("isMoving"))
        {
            dir = destPos - transform.position;
            transform.LookAt(destPos);
        }
    }
}
```

03게임 전체 구조

Game Structure

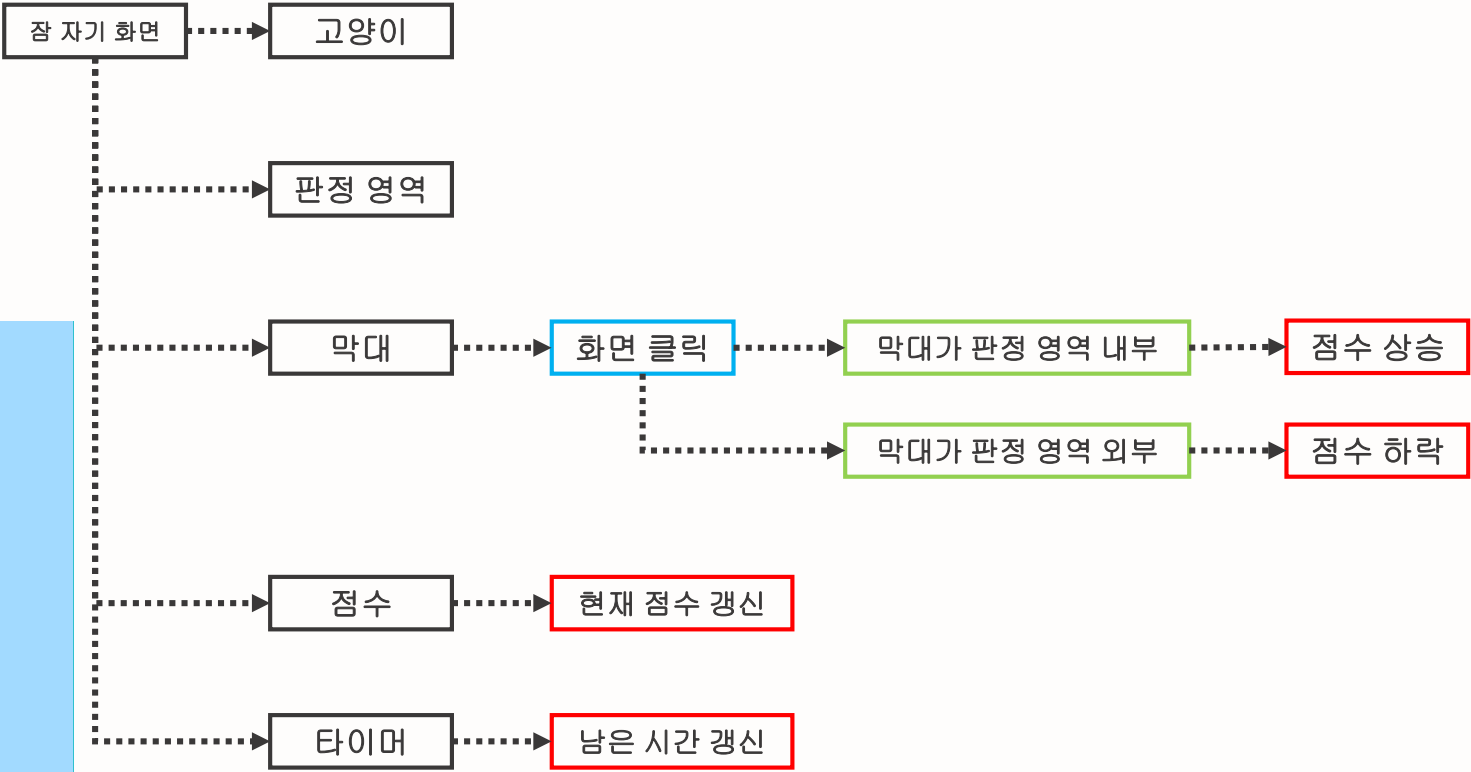
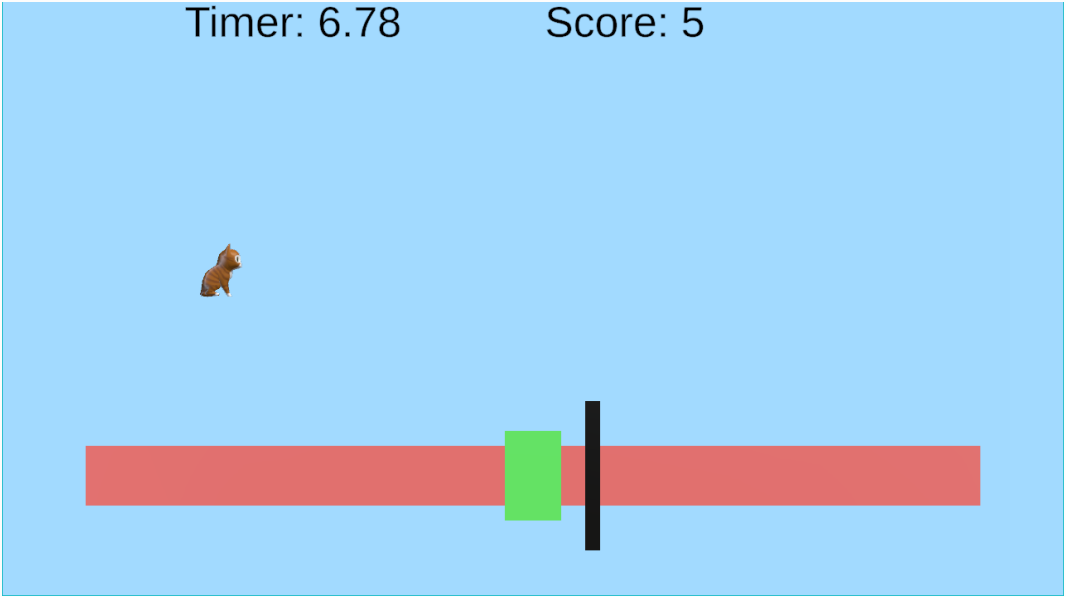
■ 사냥 관련 코드

```
void CatMove()
{
    // 이동
    if(anim.GetBool("isMoving"))
        transform.position += dir.normalized * catSpeed;

    // 목적지에 도달하면 (0.01f 이하) 이동 종료
    if (Vector3.Distance(transform.position, destPos) <= 0.01f)
    {
        anim.SetBool("isMoving", false);
        //sitTimer = 0f;
    }
}
```

03게임 전체 구조
Game Structure

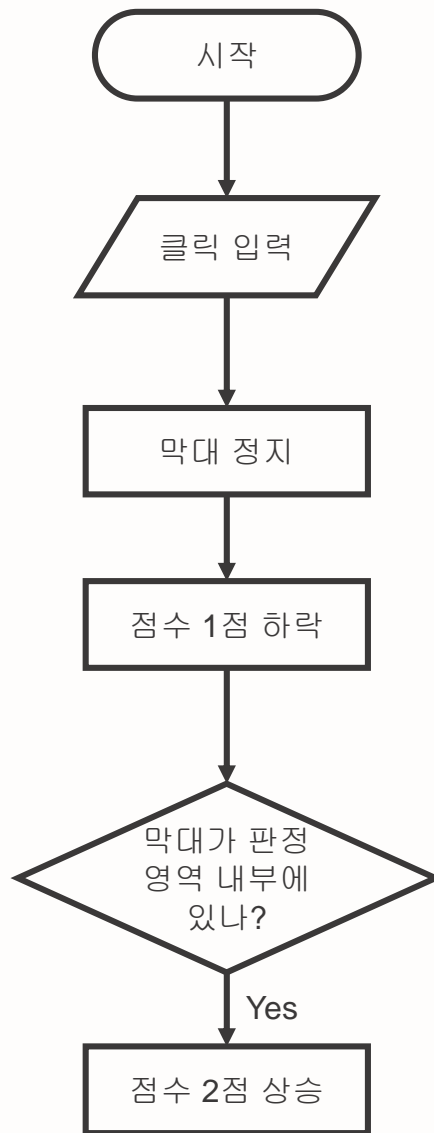
잠 자기 순서도



03게임 전체 구조 Game Structure

■ 잠 자기 관련 코드 흐름도

점수 판정



03게임 전체 구조

Game Structure

■ 잠 자기 관련 코드

```
IEnumerator coStopStick()
{
    while (system.time > 0f)
    {
        if (Input.GetMouseButtonDown(0))
        {
            // 점수가 0이 아니면 점수 1점 하락
            if(system.score != 0)
                system.score--;
            Debug.Log("click");
            // 막대 0.5초동안 정지
            speed = system.GetSpeed();
            system.SetSpeed(0f);
            yield return new WaitForSeconds(0.5f);
            system.SetSpeed(speed);
        }
        else
            yield return null;
    }
}
```

03게임 전체 구조

Game Structure

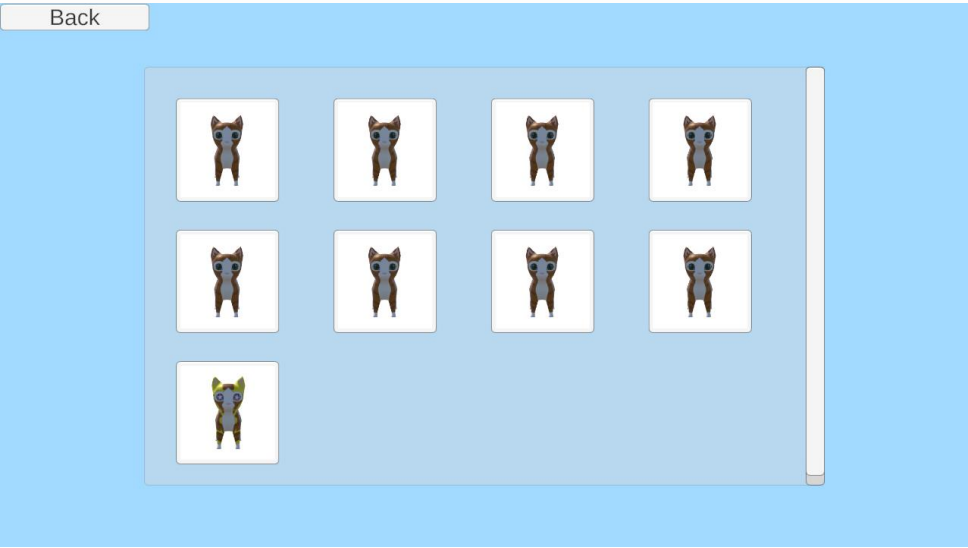
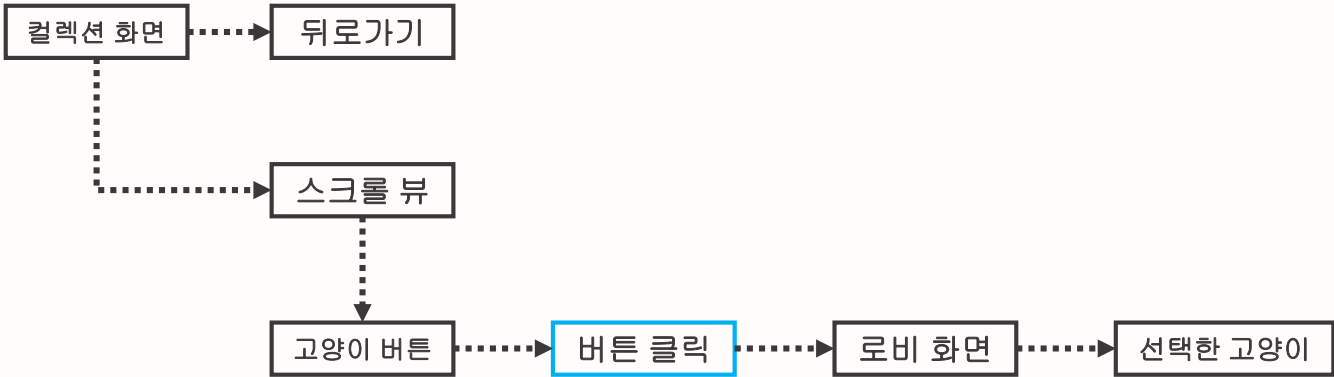
■ 잠 자기 관련 코드

```
private void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.name == "CorrectZone")
    {
        Debug.Log("correct");
        system.score += 2;
        Instantiate(Zzz);
    }
}
```

03게임 전체 구조

Game Structure

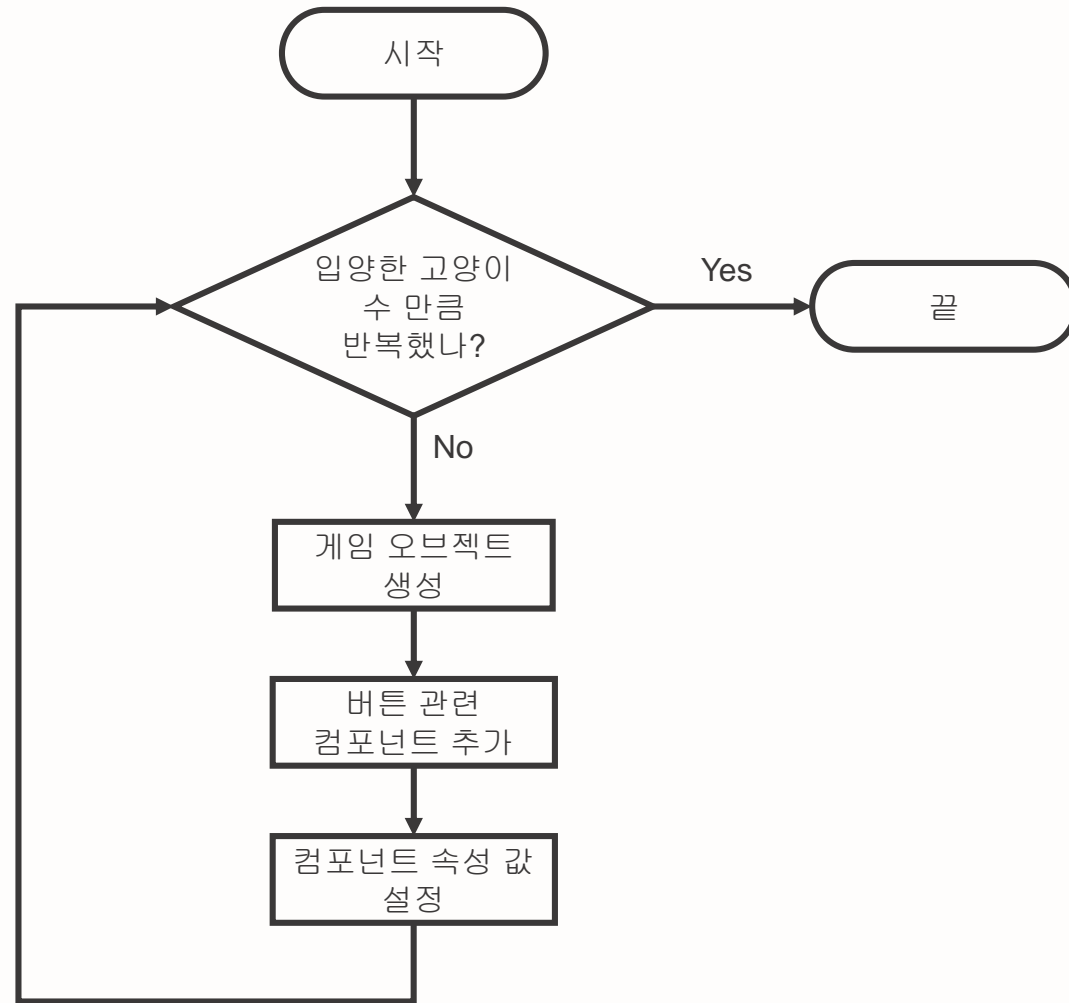
컬렉션 순서도



03게임 전체 구조 Game Structure

컬렉션 관련 코드 흐름도

버튼 생성



03게임 전체 구조

Game Structure

컬렉션 관련 코드 흐름도

```
void AddButtons()
{
    for(int i = 0; i < gm.GetCatsCount(); i++)
    {
        // 버튼 생성, 컴포넌트 추가
        btnCat = new GameObject("Button_" + i).AddComponent<Button>();
        image = btnCat.AddComponent<Image>();
        rectTransform = btnCat.GetComponent<RectTransform>();
        catImage = new GameObject("Image").AddComponent<Image>();
        catImage.transform.SetParent(btnCat.transform);

        // 컴포넌트 속성 설정
        image.sprite = buttonImage;
        image.type = Image.Type.Sliced;
        btnCat.transform.SetParent(content.transform);
        rectTransform.anchoredPosition = buttonPos;
        catImage.sprite = gm.GetCatSprite(i);
        catImage.type = Image.Type.Filled;
        catImage.rectTransform.sizeDelta = new Vector2(90f, 90f);
        rectTransform.localScale = Vector3.one;

        int index = i;
        btnCat.onClick.AddListener(delegate { SelectCat(index); });
    }
}
```

■ 문제점 및 보완점 자체 평가

문제점

- 1) 개발 계획이 명확하고 구체적이지 않다.
- 2) 코드가 간결하지 못하고 불필요한 코드가 많다.
- 3) 기능과 용도에 따라 구분하지 못하고 오브젝트에 스크립트를 배치했다.
- 4) 에셋과 리소스들을 적절히 찾고 사용하지 못했다.

보완점

- 1) 기획과 계획 단계부터 날짜를 기준으로 개발 단계를 수립해야 함.
- 2) 스크립트 작성 전 용도와 기능을 고려하여 코드 간의 의존성, 결합도를 낮출 수 있게 하고 적절한 주석과 줄 바꿈을 통해 가독성을 높인다.
- 3) 기획 단계에서 필요한 에셋을 찾는다.

감사합니다