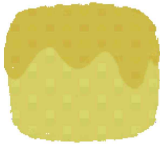


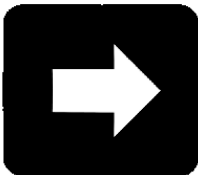
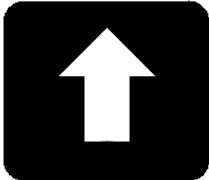


<냥이 대여정> 게임 설계서 & 설명서

조 이름	-	
조원 역할	하지연	게임 전체 개발, 스토리 제작
	-	게임 전체 일러스트 제작
	-	게임 전체 개발
게임 이름	냥이 대여정	
게임 주제	고양이가 황금 생선을 얻기 위해 대여정을 떠나는 이야기	
게임 연령층	전체 이용가	
게임 조작 방법	오른쪽, 왼쪽, 점프	
게임 세부 기술	횡스크롤 방식	
게임 줄거리	각 맵을 클리어할 때마다 보석 1개를 획득하는 방식으로, 총 4개의 맵을 성공하여 얻은 보석 4개로 황금성 문을 연다. 마지막 맵인 황금성 맵을 성공하여 황금 생선을 얻는다. 황금 생선 획득 시 게임을 클리어할 수 있으며 에필로그를 끝으로 게임을 다시 처음부터 시작할 수 있다.	
게임 방법	<ol style="list-style-type: none"> 1. 오프닝 시청 2. 프롤로그 스토리 시청 3. 시작 버튼을 눌러 게임 시작 4. 공중에 있는 블록 위에서 바닥에 떨어지지 않도록 좌우로 움직임 / 장애물을 밟지 않도록 점프 5. 맵 끝에서 있는 보석 획득 6. 스토리 진행 7. 4개의 맵을 진행하고 4개의 보석을 획득하면 마지막 맵 진행 8. 마지막 맵을 클리어하면 황금 생선 획득 9. 에필로그 시청 	
게임 조작 방법	블록에서 떨어지지 않고 장애물을 밟지 않도록 방향키와 스페이스 바를 이용해 좌우로 움직이거나 점프한다. 맵 진행 중 각종 아이템을 획득하여 다양한 효과를 부여받는다. 각 맵을 클리어할 시 보석을 획득하며 플레이 중 목숨이 0개가 될 경우 해당 맵을 처음부터 플레이한다.	

1.1 게임 세부 내용

장애물		블록	맵 별로 색상 상이
		가시	접촉 시 생명 -1
조작 버튼		왼쪽 방향키	왼쪽으로 이동
		오른쪽 방향키	오른쪽으로 이동
		스페이스 바	위로 점프

1.2 맵 별 대표 아이템 & BGM

프롤로그			
BGM	stay home		
1 단계 : 겨울 섬			
대표 아이템		눈사람	속도 증가
BGM	Purple Plants music - first step		
2 단계 : 바다 왕국			
대표 아이템		조개 껍데기	3초동안 대미지 없음
BGM	오정석 - Moments		
3 단계 : 초록 요정 숲			
대표 아이템		물약	날아감
BGM	peritune - wish5		
4 단계 : 꿈의 나라			
대표 아이템		베개	슈퍼점프
BGM	Purple Plants music - Happy Footsteps		
5 단계 : 황금성			
대표 아이템	난이도 조절을 위해 아이템을 획득할 수 없음		
BGM	Lemon Pound Cake - Oneul		
에필로그			
BGM	우연히 마주치다		

1.3 맵 클리어 후 스토리 등장 캐릭터

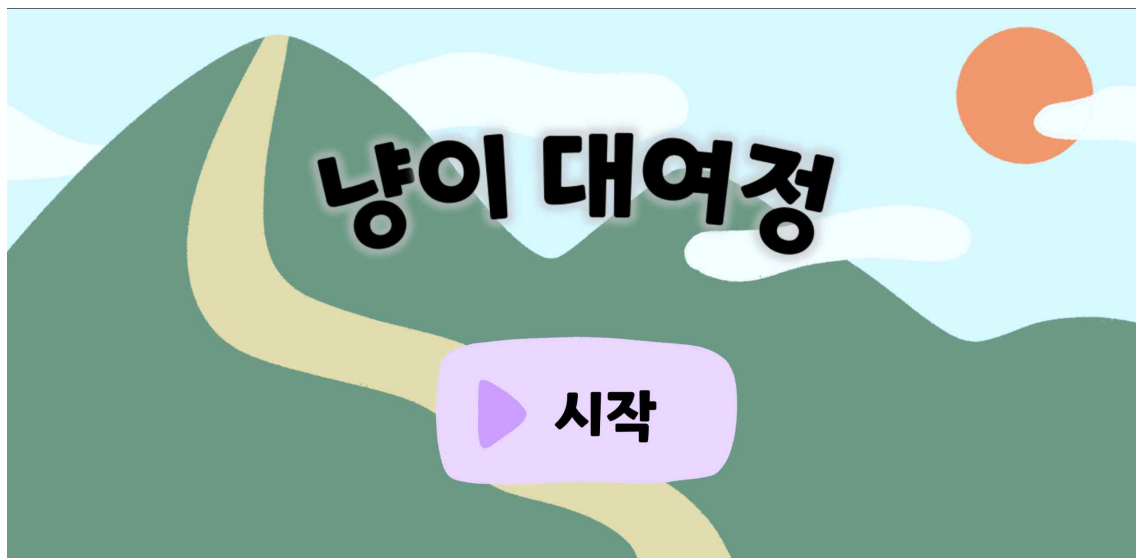
펭귄	
BGM	겨울 산-눈 바람 소리
인어	
BGM	계곡 소리
초록 요정	
BGM	산새 소리
검정 고양이	
BGM	슬픔-클래식-약간슬픔

1.3 게임 진행 순서

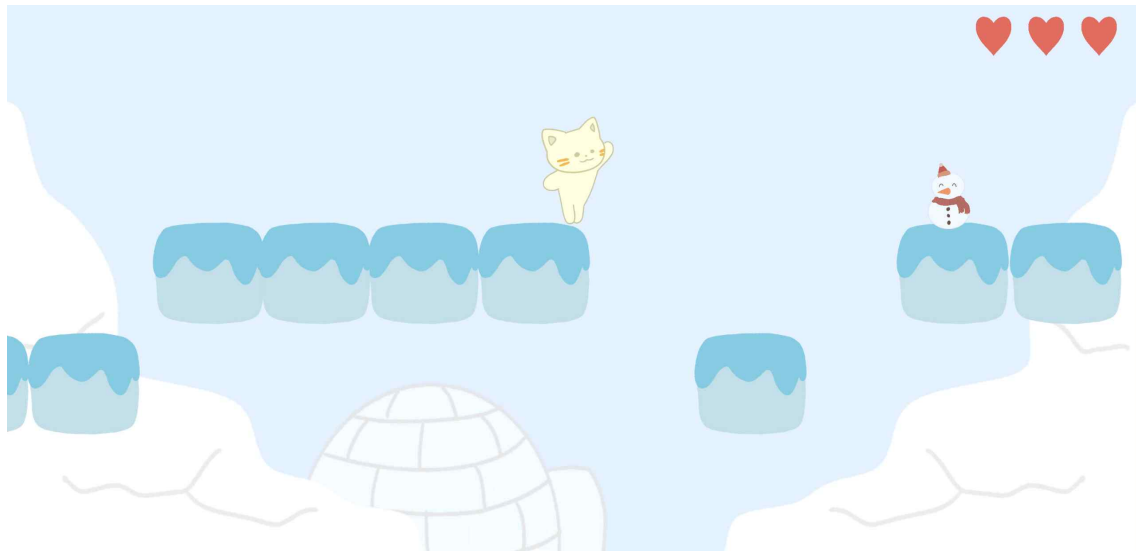
1) 프롤로그



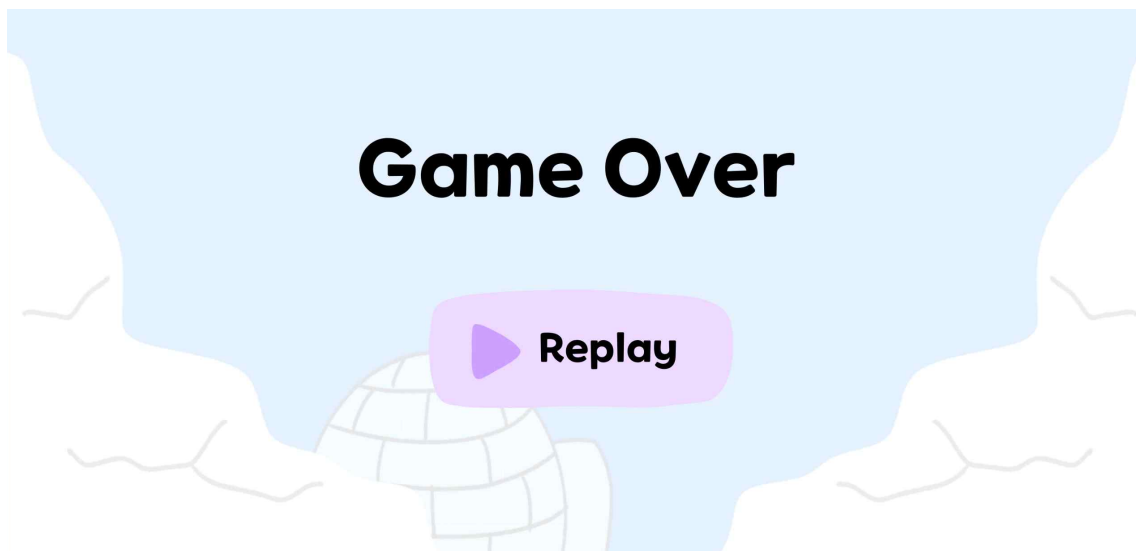
2) 시작



3) 맵 진행



3-1) 게임 오버



- 해당 맵이 배경인 게임 오버 씬으로 전환, Replay 버튼으로 해당 맵 처음부터 게임 재시작

4) 보석 획득



5) 스토리 애니메이션



- 스토리 진행 후 다음 맵으로 이동

1.4 게임 코드

1) camera director (횡스크롤 방식 사용)

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class CameraController : MonoBehaviour
6  {
7      GameObject player;
8      // Start is called before the first frame update
9      void Start()
10     {
11         this.player = GameObject.Find("player");
12     }
13
14     // Update is called once per frame
15     void Update()
16     {
17         Vector3 playerPos = this.player.transform.position;
18         transform.position = new Vector3(
19             playerPos.x, transform.position.y, transform.position.z);
20     }
21 }
22
```

2) player controller

```
public class PlayerController : MonoBehaviour
{
    Rigidbody2D rigid2D;
    Animator animator;
    float jumpForce = 530.0f;
    float walkForce = 20.0f;
    float maxWalkSpeed = 3.0f;
    SpriteRenderer rend;

    void Start()
    {
        this.rigid2D = GetComponent<Rigidbody2D>();
        this.animator = GetComponent<Animator>();
        rend = GetComponent<SpriteRenderer>();
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) && this.rigid2D.velocity.y == 0)
        {
            this.rigid2D.AddForce(transform.up * this.jumpForce);
        }

        int key = 0;
        if (Input.GetKey(KeyCode.RightArrow)) key = 1;
        if (Input.GetKey(KeyCode.LeftArrow)) key = -1;

        float speedx = Mathf.Abs(this.rigid2D.velocity.x);

        if (speedx < this.maxWalkSpeed)
        {
            this.rigid2D.AddForce(transform.right * key * this.walkForce);
        }

        this.animator.speed = speedx / 2.0f;

        if (Input.GetKey(KeyCode.RightArrow))
        {
            rend.flipX = false;
        }
        else if (Input.GetKey(KeyCode.LeftArrow))
        {
            rend.flipX = true;
        }
    }

    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.tag.Equals("gem"))
        {
            if (SceneManager.GetActiveScene().name == "map1playScene")
                SceneManager.LoadScene("map1clearGemScene");
            else if (SceneManager.GetActiveScene().name == "map2playScene")
                SceneManager.LoadScene("map2clearGemScene");
            else if (SceneManager.GetActiveScene().name == "map3playScene")
                SceneManager.LoadScene("map3clearGemScene");
            else if (SceneManager.GetActiveScene().name == "map4playScene")
                SceneManager.LoadScene("map4clearGemScene");
            else if (SceneManager.GetActiveScene().name == "map5playScene")
                SceneManager.LoadScene("map5clearGemScene");
        }
    }
}
```


3) PrologueTextDirector (프롤로그 스토리 진행)

```
7 public class PrologueTextDirector : MonoBehaviour
8 {
9     public float time;
10    public Text PrologueText;
11    int num = 0;
12
13    // Start is called before the first frame update
14    void Start()
15    {
16        PrologueText.text = "안녕하세요. 냥이 뉴스입니다.";
17    }
18
19    void Update()
20    {
21        time += Time.deltaTime;
22        num = (int)time;
23
24        switch(num)
25        {
26            case 3:
27                PrologueText.text = "요새 화제인 황금 생선에 관한 소식입니다.";
28                break;
29            case 6:
30                PrologueText.text = "황금 생선은 희귀한.....";
31                break;
32            case 9:
33                PrologueText.text = "네 가지의 보석에 황금성의 열쇠.....";
34                break;
35            case 12:
36                PrologueText.text = "그리고 그 끝에 황금 생선.....";
37                break;
38            case 14:
39                PrologueText.text = "";
40                break;
41            case 15:
42                SceneManager.LoadScene("StartScene");
43                break;
44        }
45    }
46 }
47
48
49
50
51
52
53
54
55
56
57
```

4) dead (플레이어 생명 감소, 생명 부족 시 다음 씬 전환)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class dead : MonoBehaviour
{
    GameObject life1;
    GameObject life2;
    GameObject life3;
    Animator animator;

    public int i = 2;
    public int x = 0;

    // Start is called before the first frame update
    void Start()
    {
        life1 = GameObject.Find("life1");
        life2 = GameObject.Find("life2");
        life3 = GameObject.Find("life3");
    }

    void Update()
    {
        Vector2 view = Camera.main.WorldToScreenPoint(transform.position);
        if (view.y < -50)
        {
            if (SceneManager.GetActiveScene().buildIndex == 3)
                SceneManager.LoadScene("map1deadScene");
            else if (SceneManager.GetActiveScene().buildIndex == 7)
                SceneManager.LoadScene("map2deadScene");
            else if (SceneManager.GetActiveScene().buildIndex == 11)
                SceneManager.LoadScene("map3deadScene");
            else if (SceneManager.GetActiveScene().buildIndex == 15)
                SceneManager.LoadScene("map4deadScene");
            else if (SceneManager.GetActiveScene().buildIndex == 19)
                SceneManager.LoadScene("map5deadScene");
        }
    }
}

void OnCollisionStay2D(Collision2D other)
{
    if (other.gameObject.tag == "thorn")
    {
        if (x == 0)
        {
            if (i == 2)
            {
                Destroy(life1);
                Destroy(other.gameObject);
                i--;
            }
            else if (i == 1)
            {
                Destroy(life2);
                Destroy(other.gameObject);
            }
        }
        else
        {
            if (SceneManager.GetActiveScene().buildIndex == 3)
                SceneManager.LoadScene("map1deadScene");
            else if (SceneManager.GetActiveScene().buildIndex == 7)
                SceneManager.LoadScene("map2deadScene");
            else if (SceneManager.GetActiveScene().buildIndex == 11)
                SceneManager.LoadScene("map3deadScene");
            else if (SceneManager.GetActiveScene().buildIndex == 15)
                SceneManager.LoadScene("map4deadScene");
            else if (SceneManager.GetActiveScene().buildIndex == 19)
                SceneManager.LoadScene("map5deadScene");
        }
    }
}
```

5) map1clear (맵 클리어)

```
public class map1clearAniDirector : MonoBehaviour
{
    int num = 0;
    public Text map1AniText;
    public string m_text;
    public float m_speed = 0.09f;

    IEnumerator typing(Text typingText, string message, float speed)
    {
        for (int i = 0; i < message.Length; i++)
        {
            typingText.text = message.Substring(0, i+1);
            yield return new WaitForSeconds(speed);
        }
    }

    void Start()
    {
        m_text = "황금 생선을 찾으러 여기 겨울섬까지 왔다고?";
        StartCoroutine(typing(map1AniText, m_text, m_speed));
    }

    void SwitchText()
    {
        if (num == 1)
        {
            m_text = "그럼 다음 행선지는 바다왕국이겠네.";
            StartCoroutine(typing(map1AniText, m_text, m_speed));
        }
        else if (num == 2)
        {
            m_text = "물랐어?";
            StartCoroutine(typing(map1AniText, m_text, m_speed));
        }
        else if (num == 3)
        {
            m_text = "바다왕국은 아래 지역으로 내려가면 나올 거야.";
            StartCoroutine(typing(map1AniText, m_text, m_speed));
        }
        else if (num == 4)
        {
            m_text = "꼭 황금 생선을 찾길 바라.";
            StartCoroutine(typing(map1AniText, m_text, m_speed));
        }
        else
        {
            SceneManager.LoadScene("map2playScene");
        }
    }

    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            num = num + 1;
            SwitchText();
        }
    }
}
```

6) getItem (맵 1,3,4에서 아이템 획득 시 실행)

```
public class getItem : MonoBehaviour
{
    Rigidbody2D rigid2D;
    public float time;
    public float walkForce;
    public float maxWalkSpeed;
    public float booster;
    public float jumpForce;

    int num = 0;

    void Start()
    {
        this.rigid2D = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        time += Time.deltaTime;
        num = (int)time;
    }

    void OnCollisionStay2D(Collision2D col)
    {
        if (col.gameObject.tag == "map1item")
        {
            Destroy(col.gameObject);
            this.rigid2D.AddForce(Vector2.right * 10, ForceMode2D.Impulse);
        }
        else if (col.gameObject.tag == "map3item")
        {
            Destroy(col.gameObject);
            this.rigid2D.AddForce(Vector2.right * 15, ForceMode2D.Impulse);
        }
        else if (col.gameObject.tag == "map4item")
        {
            this.rigid2D.AddForce(transform.up * this.jumpForce);
        }
    }
}
```

7) map2item (맵 2에서 아이템 획득 시 실행)

```
public class map2item : MonoBehaviour
{
    public Animator ani;

    // Start is called before the first frame update
    void Start()
    {
        ani.SetBool("getShell", false);
    }

    void OnInvoke()
    {
        ani.SetBool("getShell", false);
        GameObject.Find("player").GetComponent<dead>().x = 0;
    }

    void test()
    {
        GameObject.Find("player").GetComponent<dead>().x = 1;
    }

    void OnCollisionStay2D(Collision2D col)
    {
        if (col.gameObject.tag == "map2item")
        {
            ani.SetBool("getShell", true);
            Destroy(col.gameObject);
            test();
            Invoke("OnInvoke", 3.0f);
        }
    }
}
```

8) EndingDirector (게임 최종 클리어할 경우 시작화면으로 전환)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

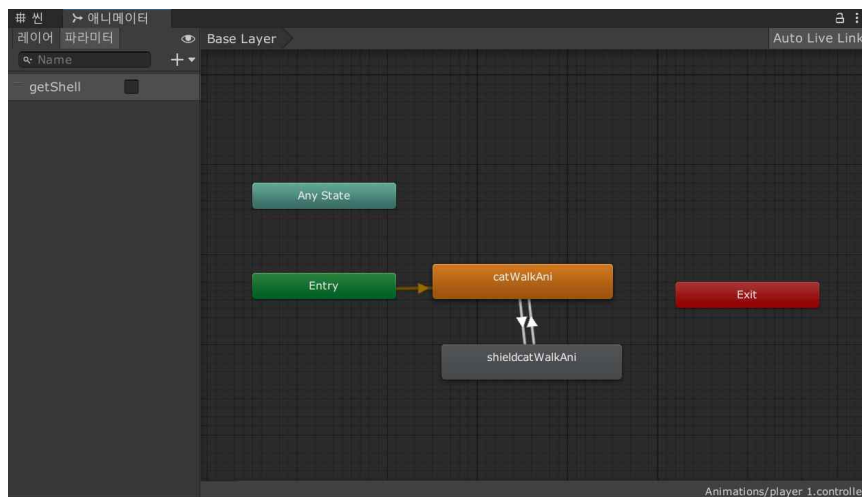
public class EndingDirector : MonoBehaviour
{
    public float time;
    int num = 0;
    public float speed = 5.0f;

    void Update()
    {
        transform.Translate(0, speed, 0);

        time += Time.deltaTime;
        num = (int)time;

        if (num > 15)
        {
            SceneManager.LoadScene("StartScene");
        }
    }
}
```

9) Animator



2. 게임 설명서

- 1) 프롤로그 애니메이션을 본 후 시작 버튼을 눌러 게임을 시작한다.
- 2) 좌우 방향키와 스페이스 바를 눌러 고양이를 이동시킨다.
- 3) 장애물에 닿으면 생명이 1개씩 감소한다. 생명이 없거나 블록에서 떨어질 경우 해당 맵을 처음부터 다시 진행할 수 있다.
- 4) 각 맵에서 등장하는 대표 아이템으로 다양한 효과를 사용해 맵을 클리어할 수 있다.
- 6) 목적지에 도달할 경우 보석을 획득하며 5개의 맵을 클리어해 보석을 총 5개 얻게 되면 게임을 클리어할 수 있다.
- 7) 게임 클리어 시 에필로그 애니메이션을 볼 수 있고, 맨 처음으로 돌아가 다시 게임을 시작할 수 있다.