

2022 년 포트폴리오

---

# 악몽

김 단 영

## 목차

- |          |           |
|----------|-----------|
| 1. 게임 소개 | 6. 조작법    |
| 2. 게임 컨셉 | 7. 게임규칙   |
| 3. 게임 특징 | 8. 주요 코드  |
| 4. 시놉시스  | 9. 소감     |
| 5. 화면 구성 | 10. Q & A |

## 게임 소개



게임 제목

악몽

게임 장르

캐주얼 러닝 게임

플랫폼

윈도우

타겟 유저

다양한 연령층 (~60대)

## ▶ 메인컨셉 : 러닝

플레이어가 계속해서 달리 ,  
점프를 하여 장애물을 피하  
게임이므 , 계속 게임  
플레이 하려면 캐릭터가 뛰  
있어야 되기 때문에 메인이 되는  
것이 러닝이라고 생각했습니다 .



# 게임 컨셉

## - 서브 컨셉

▶ 서브 컨셉 1 : 디자인      스토리의 분위기에 맞게 캐릭터 디자인과 장애물 디자인을 하였다 .

▶ 주인공      매일 악몽을 꾸는 7살 소녀



▶ 배경      악몽



▶ 몬스터      악몽 속 유령



### ▶ 서브 컨셉 2 : 스토리

매일 악몽을 꾸는 어린 소녀가 악  
깨어나고 싶어 탈출구를 찾아 뛰기 시작하지만  
탈출은 할 수 없고, 같은 공간을 계속해서 뛰는  
스토리 부분이 있다.

### ▶ 서브 컨셉 4 : 쉬운 조작법

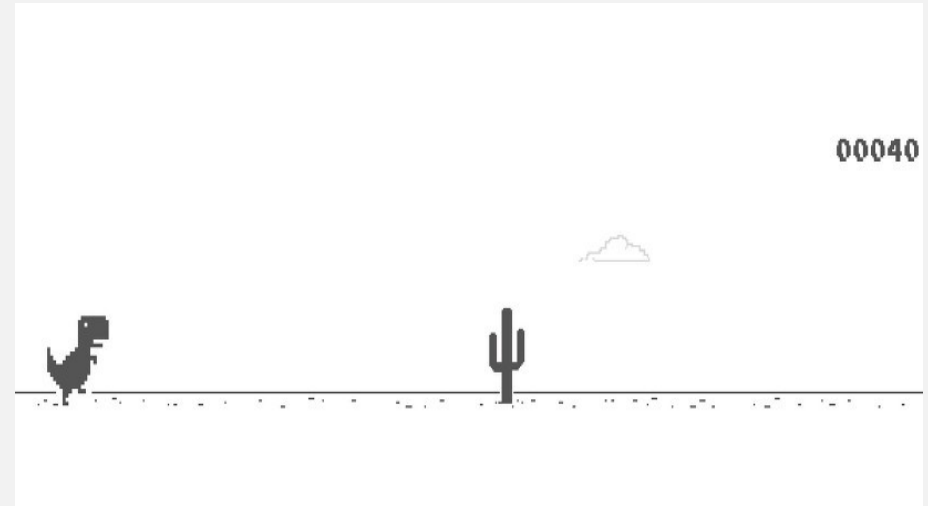
단순한 플레이를 기반으로  
플레이어들이 처음 게임을 접할 때, 적응기간이  
없을 뿐더러, 진입장벽이 낮다.

### ▶ 서브 컨셉 3 : 컨트롤

플레이어의 컨트롤에 따라 최고 기록을 서  
며, 자신의 한계를 뛰어 넘고 싶어  
도전의식을 통해 재미를 느낄 수 있다.

# 게임 컨셉

- 대표 이미지 & 관련 이미지



## 게임 특징

### - 3 대 재미요소

- ▶ 플레이어의 컨트롤에 따라 시간 기록을 세우면서 자신의 한계를 뛰어 넘고 싶어하는 도전 욕구를 통해 재미를 느낄 수 있을 것이다.

70  
%

#### Action

장애물 피하기 게임  
플레이어의  
요구하기때문에 컨트롤에  
플레이어가 느끼는  
달라지므로 액션이 가장  
비중을 차지한다.

20  
%

#### Puzzle

단순한 플레이를 기반으로  
있어, 플레이어들이 처음 게임  
접했을 때 적응 기간  
진입장벽이 낮다.

10  
%

#### Contents

비중이 크지 않지만, 스토리 부분이  
재미요소로 다가 올 수 있다.



매일 악몽을 꾸는 어린 소녀가 악몽에서 깨어나고 싶어 탈출구를 찾아 뛰기 시작하지만 탈출은 할 수 없었다. 같은 공간을 계속해서 뒹 뒹 뿐이었다.

화면 구성

최고기록

게임 시작 &  
다시 시작 버튼

스코어

BestScore

Score

게임 시작



조작법

- 조작법

▶ 조작법    스페이스바



## 게임 규칙

### - 게임 규칙 & 게임 엔딩

#### ▶ 게임 규칙

- 스페이스 바를 통해 장애물 피한다 .
- 장애물에 닿으면 게임 오버
- 게임이 시작되면 스코  
오르다 , 스코어를  
최고기록을 달성할 수 있다 .
- 게임 오버가 되 , 다시하  
버튼이 떴서 게임을 다시 도전할 수  
있다 .

#### ▶ 엔딩

장애물에 닿았을 때 게임 종료  
(클리어가 없어 게임은 무한 반복된다)

## 주요 코드

### PlayerController Script 중 점프

```
void Start()
{
    startPosition = transform.position; // 스타트포지션 함수에서 스크립트가 실행될 때 오브젝트에 현재
    포지션으로 스타트 포지션을 초기화
}
void Update()
{
    if (Input.GetKeyDown(KeyCode.Space) && GameM.instance.isPlaying) // 스페이스바를 누르면 점프
    {
        isJump = true;
    }
    // 생략
}
}
```

## 주요 코드

### GroundScroller Script 중 다리 설정

```
void Update()
{
    if (GameM.instance.isPlaying)
    {
        for (int i = 0; i < tiles.Length; i++)
        {
            if (-5 >= tiles[i].transform.position.x)
                // 타일 포지션의 x 좌표가 -5 보다 작을 때 가장 뒤에 있는 타일을 검색해 가장 뒤 타일보다 x 좌표를 -1 만큼 뒤로 보내준다 .
                {
                    // 생략
                }
            tiles[i].transform.position = new Vector2(temp.transform.position.x + 1, -0.8f);
            // x 좌표가 -5 보다 넘어버린 타일을 제일 뒤로 넘어주고 스프라이트 배열을 선언한 뒤에 제일 뒤로 보낸 타일의 스프라이트를
            스프라이트 배열 중 한 개로 랜덤하게 바꿔준다 .
            tiles[i].sprite = groundImg[Random.Range(0, groundImg.Length)];
        }
        for (int i = 0; i < tiles.Length; i++)
        {
            tiles[i].transform.Translate(new Vector2(-1, 0) * Time.deltaTime * GameM.instance.gameSpeed); // 타일을 좌측으로
            이동되게 함 .
        }
    }
}
```

## 주요 코드

### Respawn Script 중 몬스터 설정

```
Enumerator CreateMob()
```

```
// 정해진 시간마다 리스트의 장애물 중 한 개를 랜덤으로 뽑아 활성화를 계속 반복 .
```

```
{
```

```
yield return new WaitForSeconds(0.5f); // 0.5 초 뒤에 실행되도록
```

```
while (GameM.instance.isPlaying)
```

```
{
```

```
MobPool[DeactiveMob()].SetActive(true);
```

```
yield return new WaitForSeconds(Random.Range(1f, 3f));
```

```
}
```

```
}
```

## 소감

모든 연령층이 할 수 있는 게임을 만들고 싶어,  
플레이어들의 진입장벽이 낮은 러닝게임을  
제작하게 되었습니다.  
만드는 기간동안 많은 에러들을 고치며, 성장해  
나갔던 것 같습니다.  
학생 때 할 수 있는 마지막 프로젝트라고 생각이  
들어서 즐겁게 마무리 할 수 있었던 것 같습니다.



**Q & A**

**감 사 합 니 다**