

Bus - Run

한림대학교 빅데이터 전공
20215123 김수연

VR / AR / 게임제작기초 기말 프로젝트





Table of contents

01

기획 & 개발

- 기획 배경
- 계획 달성도

02

프로젝트 소개

- 개발 내용
- 핵심 코드 설명

03

시연 & 평가

- 시연 영상
- 자체 평가



기획 & 개발

기획 배경
계획 달성도

Bus - Run 스토리

바쁜 아침, 출근을 하기 위해
버스를 타야 하는 당신
될 것인가, 놓칠 것인가!

장애물을 피해 버스가 오기 전까지,
제한 시간 내에
버스 정류장에 도착하면 성공!



기획 배경



콘셉트 & 장르 선정 이유

장르 : 러닝 액션 게임

일상 속 스릴을 느낄 수 있는 상황 -> 3D 게임
몰입감, 성취감 up!



주제 선정 배경

누구나 한 번쯤은 겪어봤을, 버스를 타기 위해 뛰어가는 상황 속
"시간제한 & 목표"를 게임 요소로 해석
공감, 친숙함 up!

계획 & 달성도

| 개발 내용 | 계획 | 구현 여부 | 구현 내용 |
|---------|-------------------------------|---------------------------------------|---------------------|
| Intro | 게임 제목, 플레이 방법 설명, 스킨 선택 | <input type="checkbox"/> , 스킨 선택 X | 플레이 방법, Stage 선택 |
| 배경 | 건물, 도로, 버스 등 | <input type="checkbox"/> | 스테이지 별 다른 배경 구현 |
| 플레이어 | 움직임 등 | <input type="checkbox"/> | 움직임, 충돌체크 등 |
| 장애물 | 장애물, 아이템 | <input type="checkbox"/> | 아이템 효과 : 날기 |
| UI & 음향 | 시간, 성공여부 등 | <input type="checkbox"/> | 시간, 성공여부, 충돌 효과음 |



프로젝트 소개

개발 내용
핵심 코드 설명

주요 Object

Player

키보드의 A, D, W, S 키로
좌, 우, 앞, 뒤 이동

장애물

돌, 사람, 눈사람 등
Player의 길을 막는 장애물

아이템

충돌하면 2.5초간
player가 날게 되는
커피 아이템



Intro Scene

시작 Scene UI

- 플레이 방법 선택 버튼
- Stage 선택 버튼



Intro Scene

게임 방법 설명

- 플레이어 이동 방법
- 재시작, 다음 스테이지 이동 버튼
- 플레이어가 2.5초 동안
날 수 있는 아이템

Stage 선택

- Stage 1, 2, 3 중 선택



Stage Scene



Stage1

- 자연 속 마을 배경
- 난이도 하



Stage2

- 도시 배경
- 난이도 중



Stag3

- 크리스마스 도시
- 난이도 상

Game over UI

Time Limit : 50 sec

제한시간 : 50초
50초 안에 버스정류장 도달 실패 시
게임오버 & Restart 버튼 활성화



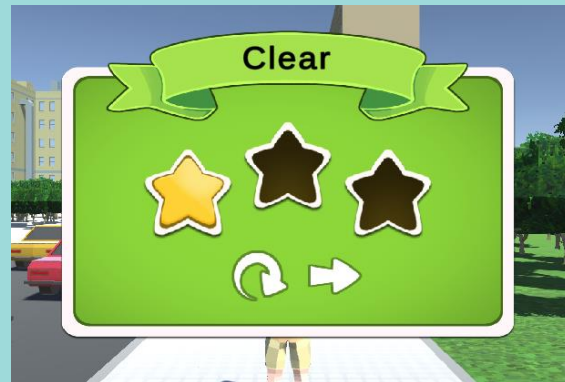
Clear UI



10초 이상 남기고 성공 시
별 3개



5초 이상 남기고 성공 시
별 2개



0초 이상 남기고 성공 시
별 1개

별 아래 버튼 선택시 재시작, 다음 Stage로 이동

Script 1 : GameManager

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

☞ Unity 스크립트 (자산 참조 6개) | 참조 2개

```
public class GameManager : MonoBehaviour
{
    public GameObject GameClear_perfect;
    public GameObject GameClear_good;
    public GameObject GameClear_normal;
    public GameObject GameOver; // 게임오버시 활성화할 텍스트 게임 오브젝트
    public TMP_Text timeText; // 남은 시간 표시할 텍스트 컴포넌트

    float TimeLimit;
    bool isGameOver;
```

☞ Unity 메시지 | 참조 0개

```
void Start()
{
    TimeLimit = 50f; //남은 시간
    isGameOver = false; //게임오버 상태 초기화
    GameOver.SetActive(false);
}
```

// Update is called once per frame

☞ Unity 메시지 | 참조 0개

```
void Update()
{
    if (!isGameOver)
    {
        TimeLimit -= Time.deltaTime; //남은 시간 차감
        timeText.text = "Time Limit : " + (int)TimeLimit + "sec";
        if (TimeLimit <= 0) //제한시간 끝나면
        {
            EndGame(); //게임오버상태로 전환
        }
    }
}
```

```
public void EndGame()
{
    isGameOver = true;
    GameOver.SetActive(true);
}
```

참조 1개

```
public void GameClear()
{
    //플레이어 디스트로이
    if (TimeLimit >= 10)
    {
        //10초 이상 남기고 성공 시
        GameClear_perfect.SetActive(true); //게임 클리어 UI SetActive
    }
    else if (TimeLimit >= 5)
    {
        //5초 이상 남기고 성공 시
        GameClear_good.SetActive(true); //게임 클리어 UI SetActive
    }
    else if (TimeLimit >= 1)
    {
        //1초 이상 남기고 성공 시
        GameClear_normal.SetActive(true); //게임 클리어 UI SetActive
    }
}
```

참조 0개

```
public void Stage1Load()
{
    SceneManager.LoadScene("Stage1");
}
```

참조 0개

```
public void Stage2Load()
{
    SceneManager.LoadScene("Stage2");
}
```

참조 0개

```
public void Stage3Load()
{
    SceneManager.LoadScene("Stage3");
}
```

Script 2 : PlayerMove

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

☞ Unity 스크립트 (자산 참조 5개) | 참조 1개

```
public class PlayerMove : MonoBehaviour
{
    Rigidbody playerRigidbody;

    public float Speed = 10.0f;
    public float PlayerY = 0f;
    public float itemTime = 0f;
    bool isItemMode = false;
    ☞ Unity 메시지 | 참조 0개
    void Start()
    {
        playerRigidbody = GetComponent<Rigidbody>();
    }

    // Update is called once per frame
    ☞ Unity 메시지 | 참조 0개
    void Update()
    {
    }
}
```

```
void FixedUpdate()
{
    float xInput = Input.GetAxis("Horizontal"); //수평축
    float yInput = Input.GetAxis("Vertical"); //수직축
    float xSpeed = xInput * Speed;
    float ySpeed = yInput * Speed;
    if (isItemMode)
    {
        if (itemTime < 2.5)
        {
            itemTime += Time.deltaTime;
            PlayerY = 4f;
            xSpeed = xSpeed * 2;
            Debug.Log(itemTime+"Sec");
        }
        else
        {
            itemTime = 0f;
            PlayerY = 0f;
            isItemMode = false; //2.5초 이상 지나면 아이템모드 나가기
        }
    }
    Vector3 dir = new Vector3(xSpeed, PlayerY, ySpeed);
    playerRigidbody.velocity = dir;
}
참조 1개
public void itemMode()
{
    Debug.Log("itemMode");
    // PlayerY = 2f;
    isItemMode = true;
}
}
```

Script 3 : obstacle

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

☞ Unity 스크립트(자산 참조 5개) | 참조 0개
public class obstacle : MonoBehaviour
{
    // 장애물 관련 스크립트 -> 충돌처리
    public AudioSource Sorry;
    public AudioSource Hurt;
    public AudioSource ItemSound;
    public GameObject PlayerMove;

    // Start is called before the first f
    ☞ Unity 메시지 | 참조 0개
    void Start()
    {
        .
    }
}
```

```
void OnCollisionEnter(Collision collision) //충돌 시작 순간 호출(collision : 충돌물체)
{
    GameManager gameManager = FindObjectOfType<GameManager>();
    if (collision.gameObject.tag == "NPC")
    {
        Sorry.Play();
    }
    if (collision.gameObject.tag == "Rock")
    {
        Hurt.Play();
    }
    if (collision.gameObject.tag == "BusStop")
    {
        Debug.Log("Game Clear2"); //게임 클리어 시 화면 구현하기
        gameManager.GameClear(); //GameManager의 GameClear메소드 호출
    }
}

☞ Unity 메시지 | 참조 0개
void OnTriggerEnter(Collider other) //충돌 시작 순간 호출(collision : 충돌물체)
{
    if (other.gameObject.tag == "Item")
    {
        PlayerMove.GetComponent<PlayerMove>().itemMode();
        ItemSound.Play();
        Destroy(other.gameObject);
    }
}
```


Script 4 : UIManager

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
☞ Unity 스크립트(자산 참조 1개) | 참조 0개
public class UIManager : MonoBehaviour
{
    public GameObject StartUI;
    public GameObject RuleUI;
    public GameObject StageUI;
    참조 0개
    public void Stage1Load()
    {
        SceneManager.LoadScene("Stage1");
    }
    참조 0개
    public void Stage2Load()
    {
        SceneManager.LoadScene("Stage2");
    }
    참조 0개
    public void Stage3Load()
    {
        SceneManager.LoadScene("Stage3");
    }
}
```

```
public void StartUI_Load()
{
    StartUI.SetActive(true);
    RuleUI.SetActive(false);
    StageUI.SetActive(false);
}
참조 0개
public void RuleUI_Load()
{
    StartUI.SetActive(false);
    RuleUI.SetActive(true);
    StageUI.SetActive(false);
}
참조 0개
public void StageUI_Load()
{
    StartUI.SetActive(false);
    RuleUI.SetActive(false);
    StageUI.SetActive(true);
}
```

Script 5 : ItemRotate

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

☞ Unity 스크립트(자산 참조 3개) | 참조 0개
public class ItemRotate : MonoBehaviour
{
    // Start is called before the first frame update
    ☞ Unity 메시지 | 참조 0개
    void Start()
    {
    }

    // Update is called once per frame
    ☞ Unity 메시지 | 참조 0개
    void Update()
    {
        transform.Rotate(new Vector3(0, 60 * Time.deltaTime, 10 * Time.deltaTime));
    }
}
```



시연 & 평가

시연 영상
자체 평가

시연 영상



자체평가



프로젝트 완성

게임 프로젝트
완성 경험



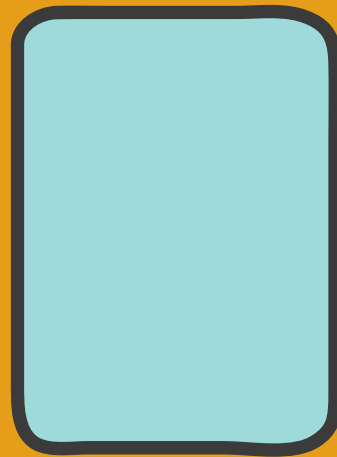
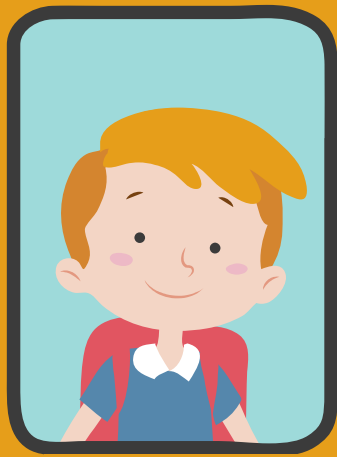
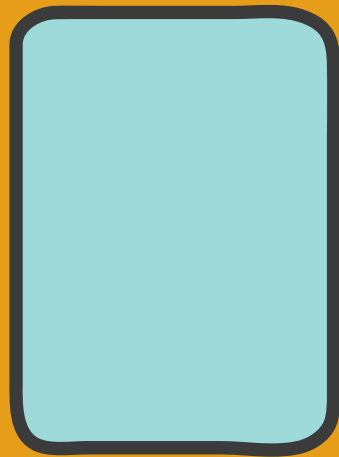
실습 학습

실습을 통해
이론 지식 적용



추가 구현

스킨 선택 기능
인트로 퀄리티 보완
엔딩 장면



감사합니다

한림대학교 빅데이터 전공

20215123 김수연