Unity Project <전진! 눈송이>

소프트웨어융합전공 1711745 이다현

컴퓨터과학전공 1612753 김윤명

1. 제목

전진! 눈송이

집(Start 지점)에서 출발하여 차(Enemy)가 다니는 길과 사람들(Enemy)이 다니는 마을에서 움직이며 최종 목적지인 학교(Goal 지점)에 제한시간 안에 도착하는 액션게임이다. 눈송이(Player)가 학교로 등교하는 스토리로 "전진! 눈송이"라는 제목을 가진다.

2. 개요

① Start Sceen 에서 LEVEL을 선택한다. (LEVEL별 Enemy인 차와 사람들의 속도가 다르다)

② 키보드 up/left/right/down 키를 입력하여 player를 이동시키며 앞으로 전진한다.

③ 이때, 차나 지나가는 사람들과 부딪히면 왼쪽위 목숨이 한 개씩 감소한다. 3번 부딪히면 Game Over Sceen이 뜬다.

④ 게임 실행 중 Q를 입력하면 Camera View 시점이 바뀐다. 거리 순으로 총 3단계 존재한다.

⑤ 가는 길에 Coffee 로고가 떠 있는 카페에 들려 Space바를 누르면 게이지가 차고 mission이 clear된다. 커피숍에 들리지 않고 Goal에 도착 시 Game Over Sceen이 뜬다.

⑥ Goal인 학교에 가까이 다가가면 종소리가 커진다. 학교 도착 시 Game Clear Sceen이 뜬다.

⑦ 게임 제한시간은 총 100초이다. 시간안에 목적지에 도달하지 못할 경우, Game Over Sceen이 뜬다.

* 최초 계획서 대비 변경 사항

- 게임의 독창성과 난이도 조절을 위해 [계획서]후진 불가능 했던 Player를 [변경]후진도 가능하게 구현했다.

- [계획서]Enemy와 충돌할 경우 전체 화면을 정지시키고 게임 오버 화면을 띄울 계획이었다. 그러나 보다 재미있는 게임 진행을 위해, [변경]목숨을 3개로 추가 설정하여 한번 충돌해도 게임을 계속 진행한다. 총 3번 충돌 시 전체 화면을 정지시키고 게임 오버 화면을 띄운다.

- [계획서]키보드 Q를 누르면 3인칭 -> 1인칭 시점으로 바뀔 계획이었다. 그러나 게임의 특징을 살리고 독창성을 위해 [변경]3인칭 시점을 유지한 채 화면 거리를 변경하는 camera view의 방법을 선택했다. 키보드 Q입력시 3개의 시점으로(가장 가깝 - 중간 거리 - 조금 먼 거리) 바뀐다.

3. Asset 목록 및 추가 구현 내용

Lowpoly Modern City Decorations Set

Low Poly Pack

POLYGON BLACKSMITH - Toony Tiny People Demo

Lowpoly Modern City Decorations Set

Skybox

Polylised - Medieval Desert City

Simple Modular Street Kit

Lowpoly Modern City Buildings Set

Simple Town Lite - Cartoon Assets

mojo-structure- Tram 1

Lake Race Track

Yughues Free Fabric Materials

Yughues Free Ground Materials

Level 1 Monster Pack

Free Snow Mountain

Free HDR Sky

Fantasy landscape

Simple Sky - Cartoon assets

Standard Assets for Unity 46

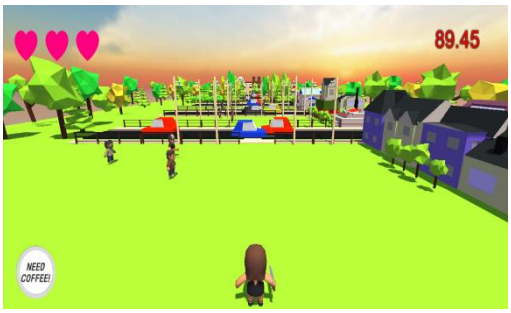
4. 주요 기능 및 실행화면 캡쳐

(1) 3가지 시점

사용자가 더 넓은 시야로 플레이를 원할 경우 유용한 기능이다. 'Q'키를 누르면 시점이 변환된다. 기본 시야는 첫 번째 시점(가장 가까운 시점)으로 설정되어 있다. 3번째는 가장 멀리서 보는 시점으로, 한 눈에 게임 맵의 전체적인 모습을 볼 수 있다. 'Q'키를 4번 누르면 다시 첫 번째 시점으로 돌아온다.

[시점1]

[시점2]



[시점3]

(2) 카페 이벤트 미션

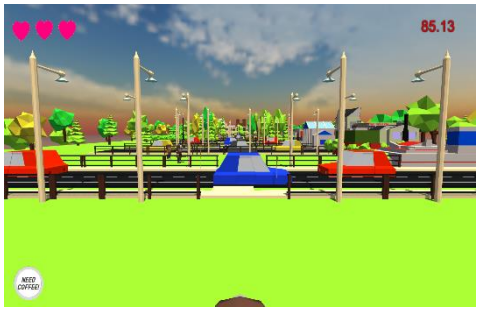
실제로 등교 시 커피를 구매하는 점을 모티브로 삼아 스토리로 추가한 이벤트이다. 플레이를 하다 보면, 사용자 기준 좌측에 커피가 그려진 말풍선이 떠있는 cafe건물이 등장한다. 이 건물 앞에서 '스페이스바'를 꾹 누르면 아래 커피 게이지가 차오르며, 해당 게이지를 모두 채우고 학교에 도착해야 게임이 클리어되는 미션이다.

(3) 3가지 레벨

총 3가지 레벨이 있으며, 각자 난이도가 다르다. 레벨이 올라갈수록 차와 행인인 장애물의 속도와 행인이 등장하는 빈도수가 늘어난다. 레별 별 하늘의 이미지와 배경음악 등 테마가 다르다.

[레벨 1]

[레벨 2]



[레벨 3]

(4) 소리 및 디자인

게임 속 등장하는 모든 요소에 소리가 들어가 있다. 지나가는 행인들, 움직이는 차, 카페, 학교 종소리, 배경음악 등의 소리를 sound box 기능으로 구현했다.

(5) 시간 제한

모든 게임이 100초의 시간 제한이 있다. 게임 내에 Goal에 도달하지 못할 경우 Game Over Sceen이 뜬다. 자세한 기능 설명은 스크립트에 있다.

[Start Sceen]

[Mission Clear]

[Game Over]

(6) 목숨 3개

Enemy인 차나 지나가는 사람들과 부딪히면 왼쪽 위 목숨이 한 개씩 감소한다. 3번 부딪히면 Game Over Sceen이 뜬다.

(7) Enemy 랜덤 생성

Enemy인 차와 지나가는 사람들이 일정 시간이 지나면 리스폰 구역에서 랜덤으로 생성된다. 자세한 기능 설명은 스크립트에 있다.

5. 미 구현 또는 오류 기능

- 게임 스토리와 통일되지 않는다고 생각하여 '벽 부시기' 이벤트 기능을 구현하지 않았다.

- 장애물 리스폰 간격이 랜덤이어서, 가끔 장애물이 리스폰 영역에서 위로 튀어 오르는 경우가 있다.

6. 사용된 스크립트 코드

(설명을 위해 실제 스크립트에서 일부만 가져왔다)

<Bounce.cs> 사용자 캐릭터의 움직임을 관리하는 스크립트

<PedestrianController.cs> 행인 장애물 리스폰을 관리하는 스크립트

<CameraManager.cs> Q키 누를 시 시점변환(3가지)을 관리하는 스크립트

<GameManager.cs> 타이머 이벤트를 관리하는 스크립트

<startMain.cs> 게임 실행 시 처음 뜨는 메뉴 화면에 관한 스크립트

<gameOver.cs> 게임오버 시 restart 버튼을 관리하는 스크립트

<충돌 처리&목숨 감소 이벤트> AnimationController.cs

AnimationController.cs

```
public GameObject heart3;          //Canvas에 있는 ui이미지 3개(♥)를 GameObject로 참조함

public static bool isUnBeatTime = false;     //충돌 간 지연시간 처리 변수
public int maxHealth = 3;          int current_health = 3;
void Start () {
        current_health = maxHealth;
}
void OnCollisionEnter (Collision col) {        //충돌 처리(행인, 자동차)
            if (col.gameObject.CompareTag ("car")) {
             reduceLife (current_health);
            }
}
void reduceLife (int current_heart) {          //충돌 시 목숨 감소 처리
        if (current_health == 3) {
            isUnBeatTime = true;
            StartCoroutine ("UnBeatTime");
            heart3.SetActive (false);   //3 번째 하트 ui 이미지 비활성화
        }
         current_health--;    //목숨 하나 감소
}
IEnumerator UnBeatTime() {         //충돌 시 여러 번 충돌로 계산되는 것을 방지
        int countTime = 0;
        while (countTime < 10) {
            yield return new WaitForSeconds (0.2f);
            countTime++;
        }
        isUnBeatTime = false;
        yield return null;
}
```

<커피 사기 미션 이벤트> AnimationController.cs

AnimationController.cs

```
bool isCollider = false;     //카페 앞인지 체크하는 변수
public Transform loadingBar;                    public Transform loadingText;
[SerializeField] private float currentAmount;   //채워진 커피의 양을 체크하는 변수
[SerializeField] private float speed;
public static bool coffee = false;    //골인 시 커피미션을 수행했는지 안 했는지 체크
void Start () {
          isCollider = false;
}
void Update () {
          if (isCollider && Input.GetKey(KeyCode.Space)) {
          //스페이스바를 꾹 누르고 있으면 커피가 채워지기 시작함
           takeCoffee ();
     }
}
void OnTriggerEnter(Collider col) {
        if (col.gameObject.CompareTag ("cafe"))         //카페 앞에 진입
           isCollider = true;
}
void takeCoffee() {
         if (currentAmount < 100) {  //커피를 채우는 중
              currentAmount += speed * Time.deltaTime;
         }
         else {       //커피를 모두 채우면 loading state 의 메시지가 바뀐다
                  loadingText.GetComponent<Text> ().text = "TAKE₩nCOFFEE!";
         }
        loadingBar.GetComponent<Image> ().fillAmount = currentAmount / 100;
        if(currentAmount >= 100)    //커피를 모두 채우면 커피미션 수행여부가 체크됨
            coffee = true;
}
```

<장애물 이벤트> carMovement.cs, carMovement2.cs (좌, 우 2가지 방향 장애물 처리)

carMovement.cs, carMovement2.cs

```
public Transform prefab;
float timeSpan;
float checkTime;  //장애물 리스폰 시간 간격을 정하는 변수
float moveSpeed;
Vector3 pos;       Quaternion rot;
Scene currentLoadedScene; //레벨별 장애물 속도를 다르게 하기 위해 scene 을 읽어옴
void Start() {
        timeSpan = 0.0f;
        checkTime = 4f;
        pos = this.gameObject.transform.position;
        rot = this.gameObject.transform.rotation;
        currentLoadedScene = SceneManager.GetActiveScene ();
        if (currentLoadedScene.name.Equals ("main")) {
           moveSpeed = Random.Range(10, 15);
        }
}
void Update() {
        if (currentLoadedScene.name.Equals ("main")) {
            if (moveSpeed < 15) //장애물 속도가 느릴수록 리스폰 간격을 넓게 해야 장애물이
과도하게 리스폰되어 뭉치는 현상이 일어나지 않음
                checkTime = 4f;
            transform.Translate (Vector3.back *Time.deltaTime* moveSpeed);
            timeSpan += Time.deltaTime;
            if (timeSpan > checkTime) {         //새로운 장애물 리스폰
                moveSpeed = Random.Range(10, 15);
                Instantiate(prefab, new Vector3(pos.x, pos.y, pos.z), Quaternion.Euler(rot.x, 90f, rot.z));
                transform.Translate (Vector3.back *Time.deltaTime* moveSpeed);
                timeSpan = 0;
            }
}
void OnTriggerEnter(Collider col) { //장애물이 도달하는 특정 구간에 trigger 를 설치하여 감지된
장애물을 삭제해야 게임에서 렉이 발생하지 않음
        if (col.gameObject.CompareTag("Respawn")) {
            Destroy (gameObject);
        }
}
```