게임엔진1 과제02

2017184002 구건모

목차

1. 게임 소개

유니티 Asset Store에서 2D Beginner Complete Project 에셋을 다운 받아 import후 실행을 해보았다.

이 게임은 2D 탑뷰 형식의 게임으로 w,a,s,d로 이동을 하며 x로 대화, c로 공격을 하는 게임이다.

주인공을 움직여 공격을 통해 적들을 춤추는 상태로 변경시키고 개구리에게 대화를 걸 수 있는 간단한 게임? 이다.

적이나 장애물에 충돌을 할 경우 HP가 줄어들게 되고 전부 줄어들게 되면 처음 시작위치로 이동이 된다. 딸기 아이템을 먹으면 체력이 회복된다.

2. 에셋 뜯어보기

Assets -> 2D Beginner 폴더에 들어가면 방금 전 import한 여러 가지 파일들이 들어있다. 각 폴더 별로 하나하나 살펴보겠다.

Art

이 폴더에는 이미지, 애니메이션, 타일 들이 들어있다.

먼저 Sprites 폴더에는 게임에서 사용하는 UI나 캐릭터 이미지, 배경 등과 같은 다양한 이미지 들이 들어있다.

Animations 폴더에는 적과 아군의 애니메이션 클립과 Animators가 들어 있어 이를 이용해 애니메이션을 표시한다.

마지막 TilemapPalettes 폴더에는 배경을 구성하는 타일들이 하나하나 쪼개져 들어있다. 갈 수 있는 타일은 Colider Type이 None로 설정 되어 있고 그렇지 않은 타일들은 Sprite로 되어있어 충돌 처리 할 때 참조 하는 것 같다.

Audio

이 폴더에는 게임에 들어있는 배경음 이랑 효과음 들이 들어있다.

Prefabs

이 폴더에는 주인공이나 적, 아이템들의 프리팹이 들어있어 필요한 부분을 저장하여 오브젝트로 사용이 편리하게 정의하여 있다.

Scenes

이 폴더에는 게임의 무대가 되는 ExampleScene 씬 파일이 들어있다.

Scripts

이 폴더에는 오브젝트들이 사용하는 스크립트들이 들어있다. 자세한 스크립트 들의 내용은 후에 설명을 하겠다.

TutorialInfo

마지막으로 이 폴더에는 게임의 아이콘 파일과 Readme 관련 스크립트 들이 들어있어 게임의 설명을 담당하는 것들이 들어있다.

3. 씬 살펴보기

본 게임은 하나의 씬으로만 이루어져 있으므로 ExampleScene에 대해 설명을 할 것 이다.

게임 기반 요소

게임을 구성하기 위해서 필수적인 카메라, EventSystem 등 중요한 오브젝트 들은 당연히 들어 있으며 추가로 배경음을 재생 하도록 하는 BackgroundMusic 이나 UI를 표실하게 하기 위한 UI -> HealthFrame가 있었다.

이 이외에도 화면 밖으로 나가지 못하게 하기 위한 WorldLimiter이나 부활 위치를 지정하는 RespawnPosition도 있었다.

Grid는 씬을 격자로 쪼개 그 안에 타일들을 배치하는 역할을 한다.

캐릭터

캐릭터의 경우 3가지의 종류로 Ruby(주인공), Jambi(NPC), Enemy(적)으로 이루어져 있다.

기타 오브젝트

먹으면 채력을 올려주는 CollectibleHealth -> Shadow가 있고 오히려 채력이 줄어드는 Damaggeable가 있다.

Environment 안에는 주인공이 갈 수 없도록 가로막는 장애물이나 장식물 등이 들어있다.

4. 캐릭터 오브젝트

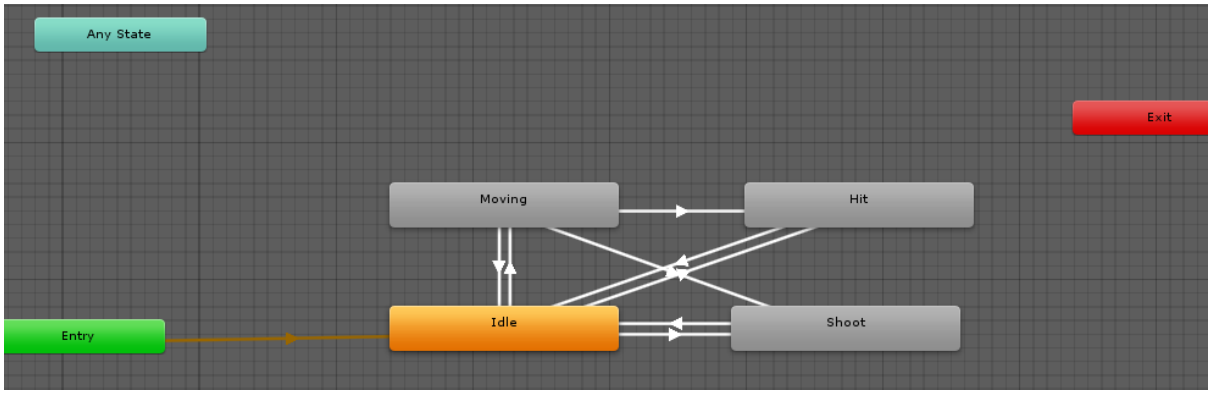
Ruby(주인공)

Ruby는 플레이 하는 유저가 게임 속에서 직접 조종할 수 있는 오브젝트 이다.

Box Collider 2D를 이용해 충돌 범위를 지정하고 있는데 offset과 Size를 조정하여 충돌 범위를 하단으로 한정하였다.

Rigidbody 2D를 이용해서 물리값을 적용해 준다.

애니메이션의 경우에는 보이는 것 과 같이 Idle 상태를 기준으로 각 상태를 왕복이 가능하며 Moving 상태 에서 Hit와 Shoot 상태로 이동이 가능하다.



하위 객체로 Shadow를 가지고 있어 여기서 그림자를 표시한다.

스크립트를 보자면 Ruby Controller스크립트 1개가 존재한다.

Input.GetKeyDown(KeyCode.C) 와 Input.GetKeyDown(KeyCode. X) 를 이용해서 공격과 대화를 한다.

대화는 RaycastHit2D를 이용해 지정된 범위안에 들어와서 키를 누를경우 GetComponent<NonPlayerCharacter>() 로 충돌된 npc에 대한 정보를 가져와 character.DisplayDialog(); 대화창을 표시한다.

공격은 Projectile 프리팹을 가지고 와서 projectile.Launch(lookDirection, 300); 라고 Launch에 값을 할당하는데 Projectile 스크립트를 들어가서 확인을 해보면

Vector2 direction, float force로 받아와서 rigidbody2d.AddForce(direction * force); 로 넘겨주게 된다. 발사체가 OnCollisionEnter2D(Collision2D other) 로 충돌시

Enemy e = other.collider.GetComponent<Enemy>(); 로 충돌한 적의 정보를 가져와서 e.Fix(); 를 해주고 Destroy(gameObject); 로 자신을 삭제한다.

이동은 float horizontal = Input.GetAxis("Horizontal"); 와

float vertical = Input.GetAxis("Vertical"); 로 유니티에서 지정된 좌우 이동키를 누를경우 그 값을 받아온 다음 Vector2 move = new Vector2(horizontal, vertical); Vector2 안에 그 값들을 집어 넣는다.

그 값이 0 이 아니면 그 방향으로 살펴보고 rigidbody2d.MovePosition(position); 로 이동을 하게 된다.

HP 변화는 public void ChangeHealth(int amount) 를 통해 amount로 받아와서 animator.SetTrigger("Hit"); 로 애니메이션을 실행하고

Instantiate(hitParticle, transform.position + Vector3.up * 0.5f, Quaternion.identity); 로 지정된 위치에 파티클을 설정하다.

그리고 피가 0과 같으면 Respawn(); 을 통해 respawnPosition.position 으로 이동한다.

Enemy (적)

Enemy는 씬에 총 3명 배치가 되어있는데 어차피 프리팹으로 구성이 되어 있고 다른게 없어 하나만 설명하겠다.

방금 전 소개한 Ruby와 크게 다를 건 없으며 Enemy스크립트를 살펴보겠다.

Start에서 horizontal ? Vector2.right : Vector2.down 를 통해 이동할 방향을 정해준다. public bool horizontal으로 선언되어있어 유니티에서 지정해준다.

```
remainingTimeToChange -= Time.deltaTime;

if (remainingTimeToChange <= 0)
{
    remainingTimeToChange += timeToChange;
    direction *= -1;
}
```

Update에 들어있는 문장인데 이를 통해 remainingTimeToChange은 지정된 숫자에서 줄다가 0 이되면 direction의 양의값과 음의값을 와리가리하는데 rigidbody2d.MovePosition에서 direciton에 해당하는 방향으로 이동하게 된다.

Fix() 함수는 주인공의 공격에 충돌하면 실행하게 되는데 충돌하면

smokeParticleEffect.SetActive(false); 로 이팩트를 종료하고 rigidbody2d.simulated = false; 통해 이동도 종료하고 다른 애니메이션으로 변경된다.

Jambi (NPC)

마지막은 개구리 형태의 NPC인 Jambi이다. Layer는 NPC로 되어있다.

Non Player Character 스크립트를 보면 public float displayTime = 4.0f;

public GameObject dialogBox; float timerDisplay; 이렇게 3가지의 변수가 있다.

dialogBox 에 다이얼로그를 넣어주어 DisplayDialog() 로 다이얼로그를 보여줄 때 timerDisplay 만큼 대화창을 표시해준다. 그 후에는 Active를 해제해준다.

5. 기타 오브젝트

Tilemap

Grid객체 아래에 Tilemap이라는 객체가 있다. 이 객체에서는 맵을 타일 형태로 나누어 표현한다.

격자에 대한 타일의 오프셋을 표현하는 Tile Anchor은 0.5 간격으로 되어있고 방향은 XY 축으로 설정 되어있다.

Tilemap Renderer에서 타일을 렌더링 하는데 타일 정렬 방향은 Bottom Left로 되어있다.

CollectibleHealth

RubyController controller = other.GetComponent<RubyController>(); 통해 충돌한 객체에 대한 정보를 가져와서 controller.ChangeHealth(1); 체력을 올려준 다음

Destroy(gameObject); 본인은 삭제한다.

6. 마무리

이번 과제는 전 과제와 다르게 2D이고 양이 적어서 분석하기가 쉬웠던 것 같다.

기존에 배경이나 장식물을 배치하던 방식에서 벗어나 타일맵을 이용하여 배치를 하니 좀 더 편리하고 수정이 쉬워진 것 같다.