Unity 3D Game Kit Tutorial

1. Asset Store 설치부분
2. 메뉴바에 Kit Tools라는 것을 사용하여 새로운 Scene을 생성하라 한다.

* 원래 메뉴바에는 KitTools가 없다
* 유니티 메뉴바 편집 기능이 있음 : [MenuItem(“상위이름/하위이름”)] 이 기본형
* Scripts/Editor/NewSceneCreator.cs
* Namespace Gamekit3D / NewSceneCreator(자식) < EditorWindow(부모)
* EditorWindow는 UnityEditor에 속해있다.
* 함수에서 함수로 이어지는 구조로 각 단계별로 다른 함수가 일하고 있다. 처음에는 새로 만들 Scene의 이름을 정하는 함수인 CheckAndCreateScene()을 사용하고 이 함수가 다시 Scene을 만드는 함수인 CreateScene을 불러오고 불려진 함수가 새로만든 Scene을 BuildSetting에 추가하는 함수인 AddSceneToBuildSettings를 부른다. 거의 줄줄이 소시지

1. Elien을 움직여 보기, 상하좌우, 점프, 공격, 카메라조작, 일시정지

* PlayerController는 외부에 보일 property로 사용하는 instance와 내부에서 사용하는 필드로 나누어 관리한다.(s\_Instance를 사용한다) 그래서 외부에선 instance값을 수정할 수 없다
* maxForwardSpeed, gravity등의 변수를 사용해서 이동속도, 회전속도등을 조절한다.
* Player가 내는 소리와 가지고 있는 무기 등을 모두 PlayerController에 넣어놓았다.
* M\_으로 시작하는 변수들과 k\_로 시작하는 변수들이 있다. M\_은 자기 자신이 가지는 값들을 말하는 것 같다. 즉, 각각의 캐릭터마다 다른 값을 가지고 있는 고유의 값이다. K\_는 어떤 것을 의미하는지 모르겠다
* Animator와 관계된 변수의 경우에는 readonly로 지정해서 이후에 값을 바꿀 수 없게 하였다. 혹시라도 값이 바뀌면 이후 동작이 이상해질 가능성이 크기 때문에 그렇게 한 것같다.
* 각 변수들의 초기화를 Reset(), Awake(), OnEnable() 세개의 함수에서 나누어서 하고있다. Reset()은 컴포넌트가 게임객체에 추가될 때 작동하는 함수이고, Awake()는 게임이 시작될 때 작동하는 함수, .OnEnable()은 객체가 enable하게 되었을 때 작동하는 함수이다.(그래서 활성화와 비활성화를 반복하면 여러 번 실행된다) 순서는 Reset() -> Awake() -> OnEnable() -> Start() 순서이다. 여기서는 Start()는 사용하지 않았다
* 이렇게 초기화를 나누어서 하게 되면 좋은점은 함수를 나눠서 사용함으로서 좀더 긴 시간을 사용하는 것이기 때문에 순간적으로 집중되는 것보다 빠를 것 같다.
* FixedUpdate()를 사용했다. FixedUpdate()는 일반 Update()보다 빠르게 실행되는데 일반 Update()는 프레임을 기준으로 실행되지만 FixedUpdate()는 고정된 시간마다 실행되기 때문에 물리 효과에 사용하기 좋다.
* 하나같이 모두 함수로 되어있다. 모든 단계를 함수로 구분지어서 만들고 있다.
* CacheAnimatorState() : 현재 애니메이션 상태를 저장한다.
* UpdateInputBlocking() : |= 이라는 오퍼레이터가 나와서 찾아봤더니 비트 or연산이라는게 나옴, 비트단위로 or연산을 하는 오퍼레이터라는데 이게 어떤식으로 사용되는 것인지 모르겠다.
* EquipMeleeWeapon() : 무기를 활성화 시키고 공격한다.
* CalculateForwardMovement() : 이동하는 크기를 연산하는 함수이다. 그에 맞춰서 애니메이션도 변화시킨다.
* CalculateVerticalMovement() : 점프 등 수직이동에 사용되는 함수이다.
* FixedUpdate()에 이런 함수들이 사용되고 있고, 교수님이 말하셨던 GetComponent와 같은 함수는 사양을 많이 먹어서인지 하나도 사용되지 않는 모습이다.

1. MovingPlatform 추가하고 다루어 보기

* MovingPlatform이 가지고 있는 첫번째 사용자 스크립트 GameCommandReceiver : namespace Gamekit3D가 아니라 Gamekit3D.GameCommands에 속해있다. 이것은 Gamekit3D에 구조상으로 속해있음을 알리고 실제 처리하는 양(헤더코드를 읽는 시간)을 줄이려고 만든 것 같다.
* MonoBehaivour를 상속받았음에도 Start, Update 등 컴포넌트가 가지는 기본적인 함수를 사용하지 않고 외부에서 사용할 함수를 만들어서 서비스하는 컴포넌트이다.
* SimpleTranslator : 위와 마찬가지로 Gamekit3D.GameCommands에 속해있는 클래스이다. MonoBehaviour로부터 상속받은 GameCommandHandler부터 상속받은 SImpleTransformer에서 상속받았다. PerformTransform이라는 함수만 오버라이드 해서 가지고 있는데 어디서 사용되는 함수인지 모르겠다. 모든 참조찾기로 찾아봐도 사용되는 경우가 없다. 물체를 이동시키는 함수이다.
* SimpleTransformer : 이동방식, 소리를 가지고있다.
* GameCommandHandler : awake등 은 사용하지 않고 상호작용에 관한 함수를 virtual 로 선언해 놓았다 또한 abstract 함수가 있기 때문에 이 클래스는 객체화 될 수 없다. RequireComponent가 있기 때문에 자식클래스를 사용할 때도 GameCommandReceiver가 자동으로 추가된다.
* 자식오브젝트로 가지고 있는 것은 물리 연산을 위한 Rigidbody와 메쉬들이다. Vector3.lerp를 통해서 위치를 옮기기 때문에 오브젝트 위에 다른 오브젝트가 있더라도 함께 움직여진다. 하지만 이렇게 될 경우 오브젝트가 내려갈 때 위에 있는 오브젝트가 떨어지는 것처럼 나타날 때가 있는데 이 프로젝트에서는 보이지 않는다.

1. 문을 만들고 문을 여는 발판을 만들어서 연결한다.

* 앞에서 사용했던 SimpleTranslator와 GameCommandReceiver가 그대로 사용되었다. 아까는 객체 전체를 들고 내려놨던 SimpleTranslator가 문틀은 그대로 두고 문만 움직이게 되었고, 발판을 통해 GameCommand를 전달받는 것으로 보인다. 아까는 항상 실행되던 모습과는 반대이다.
* PressurePad에 있는 SendOnTriggerEnter에 SimpleTranslator가 있는 게임오브젝트를 연결하고 발판을 밟게되면 SendOnTriggerEnter에서 지정해놓은 Interective Type이 전달된다. 이 타입이 서로 맞을경우 Active상태가 된다.(Active를 비Active상태로는 바꿀 수 없다.)
* 이것도 MonoBehaviour을 상속받는 SendGameCommand를 상속받는 TriggerCommand를 상속받는 SendOnTriggerEnter이다. SendOnTriggerEnter에서는 트리거가 발동 되었을 때 Send함수를 발생시킨다.(Send함수는 SendGameCommand에 만들어져 있다.)
* 앞으로도 이런 상호작용을 하는 게임오브젝트에는 SendGameCommand와 GameCommandReceiver, GameCommandHandler가 세트로 다닐 것같다.

1. 몬스터 Chomper를 추가하고 조작한다.

* Damageable 클래스를 가지고 있어서 공격을 받을 수 있다. 받았을 때 공격 각도와 공격력 등에 따라 반응한다.
* ChomperBehavior : Elien이 가지고 있는 PlayerController와 유사하다. Hash 변수들이 여러 개 정의 되어있고 EnemyController를 변수로 가지고 있다. 차이점은 플레이어를 찾기 위한 PlayerControll이 있다는 것과 탐색거리, 범위등을 가지고 있다는 점이다.
* EnemyController : Chomper의 구체적인 이동이나 활동에 관한 것들을 관리한다. NavMeshAgent와 연결되어 플레이어 추적을 할 수있다.
* PlayerController 가 두개로 나뉘어진 느낌, chomper에 관한 것을 관리하는 ChomperBehavior와 적에게 모두 달려있는 EnemyController로 말이다.

ProBuilder 라는 것을 사용하여 새로운 Scene 생성을 한다.

* ProBuilder는 3D GameKit과 마찬가지로 Unity Technology에서 제공하는 무료 에셋이다.
* ProBuilder는 오브젝트 변형, 다각 오브젝트 등 유니티에서 기본적으로 지원하는 정육면체, 구체 등 보다 다양한 오브젝트를 만들 수 있게 도와주는 에셋이다.