**Unity3D 프로그래밍 일지**

작성자:김기현

**★ 3월 14일 월요일**

작업내용: 프로그래밍 착수

**★ 3월 16일 수요일**

작업내용: 씬 단계 구성 및 작성 준비

**★ 3월 19일 토요일**

작업내용:

MainMenu, Option, Achievement, 씬 단계 구현 특별한 경우가 아니라면 MainCamera안에서 UI등등을 제작하기로 계획

세부내용:

* MainMenu
  + 어플리케이션 종료(Application.Quit())를 통해 종료 버튼 구현 및   
    UnityEngine.Scenemanagement.Scenemanger.LoadScene()을 이용하여 씬 전환 구현
* Option
  + AudioLister.Volume을 조절하여 시스템 음량 조절 방법 획득 및 구현 완료 및 체크 박스를 통해 true,false 획득 로직 구현
* Achievement
  + 횡 스크롤 방식의 버튼 묶음 구현 완료, Canvas, Panel로 구성된 것을 Scroll Rect UI를 통해서 스크롤이 되게 하였고 스크립트를 통하여 화면을 넘어가지 않게 함

**★ 3월 20일 일요일**

작업내용: 게임의 Save와 Load에 대한 모듈 구현 완료(파일 생성이나 그런 것은 가시적으로는 따로 없다고 생각중.) – Script 이름: LoadInformation.cs, SaveInformation.cs

**★ 3월 21일 월요일**

작업내용: Credit Scene을 추가하여 제작자 들에 대한 정보를 출력할 준비 완료

* UI 통일에 따른 프로그래밍 변경:  
    
  기존의 MainMenu Scene에서부터 시작하는 구조에서 Start Scene에서 시작하는 구조로 변경하였으며 순차적으로 흘러가던 구조에서 MainMenu라는 Scened을 허브로 해서 나머지 Scene들을 실행하는 형태로 바뀜  
    
  (씬 구조)  
  -Start – MainMenu – Achievement  
   - Option – Credit  
   - Stage1~6  
    
  기존에는 단순히 Stage1 Scene부터 실행하는 구조에서 사용자가 직접 Stage를 선택하여(이전 Stage를 깨지 않으면 다음 Stage는 잠겨있음) 이용할 수 있도록 메뉴에도 Achievement Scene에 있는 것처럼 횡 스크롤 형태의 버튼 뭉치를 만들어서 넘겨가며 선택 할 수 있도록 구현  
    
  ps. 변경 후 Scene 순서가 꼬이는 현상이 발견되어 재정렬 후 해결  
  ps2. 횡 스크롤 버튼 묶음은 MainCamera안에서 구현하지가 힘들어 MainCamera에서 구현은 하지 않기로 하였음. 또 MainCamera의 성질에 대한 이해가 조금더 필요하다는 것을 깨달음. (대표적으로 MainCamera에서 구현한 것은 가장 위에 오게 됨)

**★ 3월 22일 화요일**

작업내용:

MainMenu의 횡 스크롤에 있는 버튼들에 대해서 Boolean 조건에 따라서 Image가 변할 수 있도록 제작함. Stage가 잠겼을 경우에는 자물쇠 모양을 표기하기 위함임 또 Boolean 조건을 활용하여 버튼 클릭시에 조건에 따라서 다른 행동을 할 수 있도록 제작함

질문? : UNITYENGINE.SceneManagement.SceneManager.LoadScene(“ “) 에 대해서는 씬 파괴또는 언로드를 해야 하지 않나? 🡪 loadScene을 이용하여 기존의 씬을 다른 씬으로 대체하게 되면 자연스럽게 이전에 있던 씬은 언로드가 되게 됨. 오히려 unloadScene함수를 사용하면 warning을 출력함.

**★ 3월 23일 수요일**

작업내용:

속도 향상 밑 메모리 낭비 방지를 위해서 Screen.SetResolution()함수를 이용하여 1280 by 720 으로 설정하여 코딩을 진행함.

Save기능과 Load기능에 대한 구현 완료(Stage 4개, 업적 10개(개수는 수정될 수 있으며, 현재 구현상태가 너무 맘에 들지 않아 바꿀 방법을 찾으면 바꿀 예정)

void Start()

{

FirstSetting = PlayerPrefs.GetInt("First");//if it is first, load First Setting

if (FirstSetting == 0)

LoadInformation.Load\_First\_Setting();

else

LoadInformation.Load\_Previous\_Setting();

}

처음 게임 시작과 그 이후 시작을 구분하기 위하여 위의 함수를 사용하여 구분함.(스테이지와 업적들의 상태 초기화가 되지 않은 상태라 처음 시작하면서 초기화 시킴, 이후는 세이브 파일에 의지하여 저장하고 불러오게됨)

* 회의에 따른 변화 사항 밑 앞으로의 고려사항  
  + 회의에서 나온 MainMenu에 파노라마를 적용하는거에 대해서는 배경뿐만 아니라 Button같의 간격또한 늘려서 처리해야 할 거 같다고 생각됨. 우선 배경이 완성되면 처리할 예정
  + 회의에 따라 스테이지의 개수가 4개로 줄었기 떄문에 기존 6개의 버튼으로 제작된 MainMenu Scnene은 언패키지로 추출하고 2개를 제거하여 4개로 만들어 놓음(완료).
  + Stage Scene에 대해서 화면 분할 방식과 통짜로 하는 경우 두 경우를 고려해야함(퍼즐 완성에 따라서 변화 가능하며 현재는 할 게 없음)
  + 옵션에 들어가는 Bar 형태로 되어있던 사운드조절 부분은 물리적으로 디바이스에서도 조정할 수 있다는 점 때문에 필요가 없어 보이므로 ON/OFF System으로 교체하며 왼손 오른손 UI 배치는 선택 할 수 있도록 함. (완료)
  + Stage Scene 내부에서 Item을 획득할 때 각 object(Item)들과 사용자의 캐릭터 사이의 거리를 활용하여 아이템을 획득하도록 하여야 하는데. 이 부분에 대한 로직이 아직 안돼서 우선 조금 고민해봐야 할 거 같음.
  + 게임 내부에서 옵션을 호출했을 때 팝업이 뜨게 하며 팝업 창을 2층으로 나누어서 소리/ 게임메뉴로 나누어서 따로 표현 하도록 하며 이 부분에 대해서도 소리는 ON/OFF 체계를 통하여 함.(완료)

**★ 3월 24일 목요일**

작업내용:

Option에 있는 bar형태를 전부 버튼형으로 바꾸었음. 버튼들의 위치는 가로는 조정할 수 있게 하였으며 높이는 옆에 있는 타이틀과 같은 높이를 유지하도록 함. 중간에 혼란은 있었으나 우선은 구현은 완료하였으며 ON을 누르고 OFF를 누르는 거에 따라서 버튼마다 다른 효과를 주도록 만듬. 대신 텍스처도 그만큼 필요함. 소리 끄고 킨 것의 저장은 MainMenu에서 호출함. 또한 기존에 float형으로 제작되었던 소리용 변수 2개를 Boolean형으로 변경하였고 true일 때 켜지고 false일 때 꺼진 것으로 판단하도록 하게 함. 추가로 소리는 게임을 새로 시작했을 때 자동으로 켜진 상태로 시작함(즉 옵션 상태는 저장이 안됨)

BGM과 Effect에 대해서는 SoundManager를 만들어서 사용하거나 AudioListener를 하나 더 생성해서 사용하는 방법이 있는 거 같다. 아직은 구현은 안되어 있지만 방법을 더 찾아봐야겠다.

혼자만의 생각?

네트워크로 업적 저장들을 구현할 수 있었으면 좋겠는데.. 실력이…. 우선 할거 하고 그 다음 해보자..

게임 UI중 메뉴를 구현하였다. 이쁘진 않은데 기능은 완벽하다. 역시 텍스처만 넣으면 될 것 같다

지금까지의 완성본을 갤럭시s3 LTE 단말기에 넣어서 실행해 보았다. 전반적인 진행속도는 그리 나쁘지는 않았는데. 횡스크롤 버튼뭉치가 있는 Scene을 로드시에는 조금 걸린다는 것을 알 수 있었다. 이 부분에 대해서 로딩화면 등을 구현하면 조금 보완할 수 있지 않을까 싶다.

**★ 3월 24일 목요일**

작업내용:

옵션 부 그래픽 텍스처 처리가 안돼있어서 처리함.. 아이템 버튼과 점프 버튼을 임시로 추가해둠. 변할지 아닐지 몰라서 일단은 그냥 둠.

게임 조정간에는 터치 패드를 활용하여 조정하도록 구현을 할 예정.

**★ 3월 25일 금요일**

작업내용:

Unity3D에서도 멀티쓰레드가 가능하다는 것을 알았고 로딩화면 구현에 쓸 수 있다는 것 또한 강신진 교수님 통해서 알 수 있었음. 그러니 로딩 화면도 구현해보자.

**★ 3월 31일 목요일**

MainMenu가 드디어 완성되었다. 뒷 부분의 배경이 POT로 인해서 엄청난 용량낭비가 들어간 거 같다. 전체적인 구동에 큰 문제는 없다. 맵 쪽에서도 엄청난 텍스쳐를 써야 하는데 겁이 난다.

Starting 화면에 대해서도 애미메이션 효과를 주었다. 이미지는 그래픽 쪽에서 작성해줄 것이다. 지금은 시험도 있어서 이만한다..

**★ 4월 1일 금요일(만우절)**

조이스틱을 구현하였다. 엄청 엉망인데. 대충 텍스처 박으면 괜찮아 지지 않을까 싶다. 아니 그냥 괜찮다고 믿고 싶다.

배경음과 효과음을 드디어 나누었다. 한달 동안 소리 구현 못했는데 제일 오래걸린것 같다. On Awake 옵션을 꺼줘야 처음 시작하자마자 소리가 작동되는 것을 막을 수 있다. True, false를 구분하여 소리가 나고 안나고를 구분하게 되는데 여기서 문제점은 우습게도 배경음은 loop 옵션을 걸어놔야 하는데 이상하게도 false일 때 소리가 난다. 이상하다… 꼭 참고하자. 아 또한. Mixer를 2개로 나누었고 각 Mixer 하에 사운드 그룹을 새로 만들었다. Effect는 Effect 사운드 그룹으로. BGM은 BGM 사운드 그룹으로 출력 시키면 된다.

긴 BGM 사운드의 경우는 Awake등 초기화 함수에 포함시켜 주며 짧은 음악의 경우는 프레임단위 루프 함수에 넣어주면 해결이 된다.

prefab으로 사운드 들을 처리했더니 사운드가 안 나온다… prefab은 애니메이션에만 써야겠다.

MainMenu, Option, Acheivement, Credit, Start,에 대한 사운드 삽입 준비가 완료 되었다. 이제 음원만 준비하면 된다…

**★ 4월 2일 토요일**

어제 음원 추가에 조금 부족한 부분에 대하여 보충하였다. 기타 오류를 잡았다. 또한 MainMenu에 있는 스크립트들에 대하여 최적화를 진행하였다.

**★ 4월 4일 월요일**

스테이지 별로 UI에 대한 지도와 View설정 완료. 첫 화면 더 추가 밑 애니메이션 첫 적용을 하자. 별을 랜덤으로 배치하도록 해보자…... 랜덤 배치는 구현했지만.. 하지만 애니메이션이 보이지 않는다. 왜일까.. 앞에 있는 패널의 알파채널을 낮추어 보니 뒤쪽에 랜덤 배치된 애니메이션들이 나열 되어있다. 더 찾아봐서 레이어 설정을 조정해봐도 변경되는 것이 없다.

현재까지 구현한 UI를 이용하였는데 이것들은 레이어들을 무시하고 제일 최상위 레이어로써 작동을 한다. 그렇기 때문에 이 씬에 대해서는 UI를 전부 철수하였다. 카메라를 이용하고 3D를 이용하여 2D처럼 구현하는 것으로 바꾸었다. UI를 생각을 잘못하고 있었다. 수정을 하자.

UI를 제거하고 3D Quad를 이용하여 구현, 애니메이션과 Quad의 레이어 순서를 조정하니 드디어 애니메이션이 보인다. 일단 변경된 Starting씬에서 별을 구성했다.. 내일 계속하자.

**★ 4월 5일 화요일**

Start 씬을 완성했다. 전부 다 완료 하였고 레이어에 대한 이해가 더 깊어졌다.우선 첫 화면을 완성하였다. 여러 타이밍 문제가 있어서 강제로 다 때려 박았는데 폰에 들어갔을 때 괜찮게 작동한다… 다행이다..

**★ 4월 7일 수요일**

MainMenu씬을 다시 뒤엎었다. 기존 UI로 작성되어있던 것들을 모두 Quad 기반으로 바꾸었다. 또한 쓰테이지도 그렇게 바꾸었다.

게임 오브젝트에서

Transform.Fine(“name”)형식을 이용하면 자식이나 부모관계 안에서만 찾으며

GameObject.Find(“name”)형식을 이용하면 하이라키 않에서 다 찾으며

GameObject.FindWithTage(“TagName”)을 이용하면 태그에 속하는 오브젝트를 반환하며 여러 개여 랜덤으로 반환한다.

**★ 4월 8일 금요일**

드디어 Animator를 사용할 수 있게 되었다. 애니메이션은 State, Parameter, transition으로 구성이 되며 State는 각에 해당하는 애니메이션 움직임, Parameter은 transition을 조정하는 기준, Parameter은 기준에 대해서 값으로인지 boolean인지 단순 트리거인지를 결정한다.

**★ 4월 9일 토요일**

Stage에서 가장 기본적인 사다리와 지형에 대해서 구현을 하였다. 지형에 대하여는 polygon Collider를 이용하여 구현하였고, 캐릭터는 그 Collider를 이용하여 충돌을 감지하여 떨어지지 않고 지형 위에서 버틸 수 있게 된다. 또한 캐릭터는 지형 위에서는 좌우로만 움직일 수 있다.

사다리의 경우는 사다리와의 충돌을 확인한 다음 사다리와 충돌을 하면 위 아래로 움직일 수 있다. 하지만 위쪽 지형과 충돌해서 못 올라가는 현상이 발생하는데 이를 해결하기 위하여 사다리 위에 있는 경우엔 Kinetic을 적용하도록 하여 처리하였다.

캐릭터 이동에 관해서 기존 조이스틱은 가속으로 계속 움직였기 때문에 계속 움직였다. 그렇기 때문에 Vector3(0,0,0)을 이용하여 조이스틱에서 손을 떼면 완전히 정지하도록 만들었다.

점프 부분에 대해서는 작동은 잘 하나 동시에 키가 눌리지 않는다. 멀티터치를 구현해야 할 거 같다.

멀티터치를 따로 구현할 필요없이 조이스틱이 Canvas위에 올라가있기 떄문에 canvas위에 Button 하나 추가하여 점프를 하도록 하였더니 작동을 잘한다.

**★ 4월 10일 일요일**

첫 실행과 두 번째 이후 실행을 구분하기 위해서 변수를 설정 하였다. 삭제 시엔 초기화 되고 설치 한 이후에는 계속 적용되게 하였다. 이는 첫 실행과 두 번째 실행의 불러와야 하는 세이브가 다르기 때문에 다음과 같이 설정하였다.

메인 메뉴 각각에 애니메이션을 적용한다고 하여서 기존 스테이지 잠겼냐 안잠겼냐를 구분하는 변수들을 철수 하고 이 스테이지가 깼냐? 안깼냐? 잠겼냐? 3가지를 확인하는 변수를 추가하였다. 또한 이것들은 저장이 항상 되어야 한다. 고로 저장되는 타이밍을 낼 고민 해봐야겠다.

저장 된 정보를 불러오려는데 자꾸 안된다. 어떻게 해도 안되어서 마지막으로 한게 값 호출을 Awake()에서 하게하니 된다. 다음부터는 뭔가 초기값 설정 할때는 반드시 Awake()함수에 하도록 하자… 제발 진짜 정말..

**★ 4월 18일 월요일**

캐릭터 방향에 따라 flip을 이용하여 애니메이션 방향을 정하였고 사다리 3개 구현 및 장애물 추를 구현하였다.

사다리를 타고 올라갈 때 안정적이지 못하여 가로 속도를 줄였다.

**★ 4월 19일 화요일**

모든 사다리와 모든 지형에 대한 확인 완료. 및 수정 보정.

이벤트 통지용 창 만들었다. 부드럽게 움직이게 했다.

아이템 창 뜨게만 만들었다. 상세 구현은 하지 않음

**★ 4월 19일 수요일**

카메라 이동 : 위에서 아래로 넓은데서 좁게.

**★ 4월 23일 토요일**

설정 창 알파채널 및 버튼들 변경 요청으로 변경 완료.

아이템창은 뜨나 border에 대한 active가 발생해서 이부분 수정 요

GUIStyle을 통해서 팝업의 세부 설정을 해줄 수 있다. 이렇게 수정함.

**★ 5월 5일 목요일**

여러 수정을 거침, 특히 텍스쳐를 전부 교체했으며 소스 정리 작업중.