

# Unity 5 (수업내용 정리)

## 15.05.06

한번 노출된 **public** 변수는 코드내에서 수정해봤자 무의미함  
( 노출된 변수는 인스펙터뷰의 값을 따른다 )

원손좌표계.

원손 좌표계 계산은 자신의 원손을 엄지방향을 축으로 손이 접히는 방향으로 회전한다.

유니티 **5.0.1f** 버전에서 아직도 **input**의  
버티컬, 호라이즌 이동 보간이 **1 ~ -1**로 가는게 문제가 좀 있다.  
//InputManager 셋팅된 Axis 값을 가져온다.  
//Edit->Project Settings->Input 을 선택 , 인스펙터의 **Axes** 를 보라....

대각선 속도 보정하기

```
//moveVec.Normalize ();  
//moveVec /= length;           //정규화 ( 나눗셈 3번 )  
moveVec *= (1.0f / length);     //점규화 ( 나눗셈 1번에 곱셈 3번 )  
나눗셈이 연산에 더 부담된다. 10 / 100 = 0.1 / 10 * 0.01 = 0.1
```

유니티에서 기본적으로 회전값은 **Degree** 이다.  
근데 **Mathf** 수학관련은 싹다 **Radian** 이다

자매 품 회전사원수 얻기 함수들

Quaternion.AngleAxis (Degree, Vector3);  
//Vector3 를 축으로 Degree 만큼 회전하는 회전 사원수를 얻는다.

Quaternion.LookRotation (Vector3);  
//Vector3 를 바라보는 회전 사원수를 얻는다

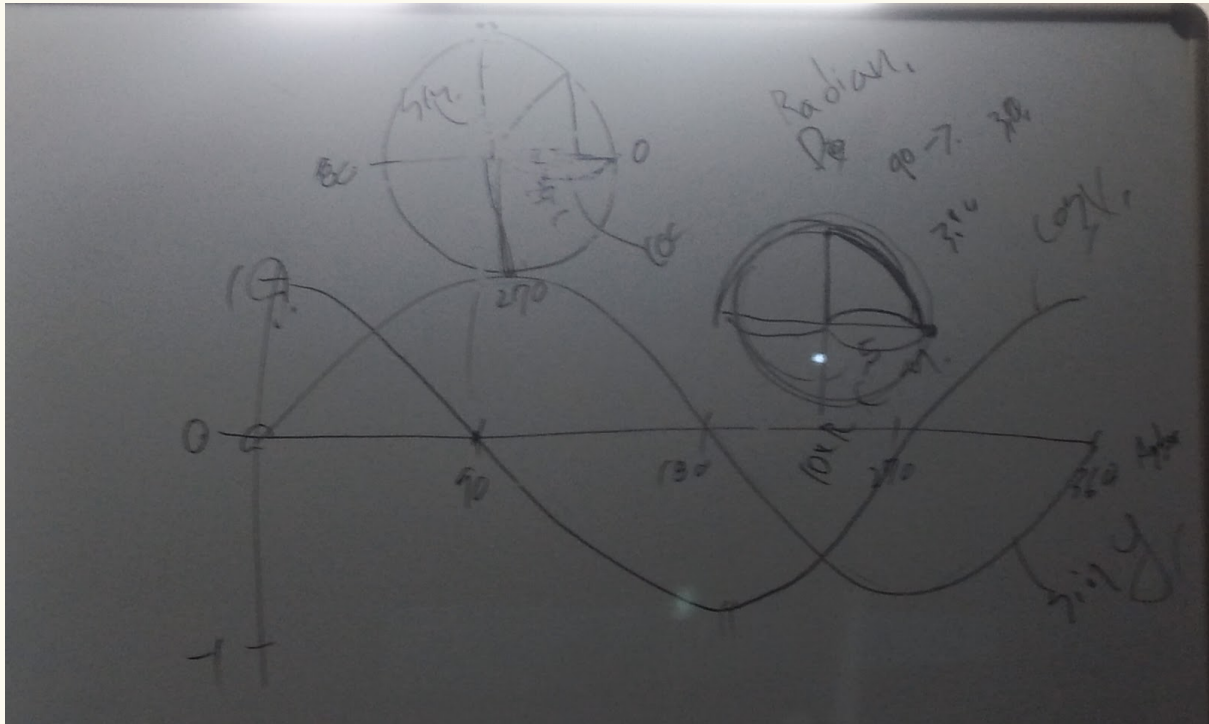
Quaternion.LookRotation(Vector3 Forward , Vector3 Up);  
// Up 방향을 위로 기준으로 삼은 Forward 를 바라보는 회전 사원수를 얻는다.

Quaternion.Slerp( Quaternion A, Quaternion B, float 0 ~ 1 );  
// A 회전과 B 회전과 float 0 ~ 1 사이의 구면보간을 해주는 사원수를 얻는다,  
0 일수록 A 와 가까워짐 1 일수록 B 와 가까워짐

Quaternion.Angle( Quaternion A, Quaternion B );  
//두 사원수 간의 회전 차를 Degree 각도로 얻는다.

15.05.07

코사인, 싸인 그래프



$\cos(0) = 1$  Degree //  $\cos(90) = 0$  Degree //  $\cos(180) = -1$  Degree //  $\cos(270) = 0$  Degree  
 $\sin(0) = 0$  Degree //  $\sin(90) = 1$  Degree //  $\sin(180) = 0$  Degree //  $\sin(270) = -1$  Degree

**tip)** sin은 s 스타트 0도부터 시작해 올라간다. cos는 1부터 내려간다.

코싸인,싸인을 이용해서 방향 벡터를 구한다.

**radian \* degree = 호**

이걸 이용하여 바퀴가 몇바퀴 굴렀냐 는등 구할 수 있다.

**Dot Product(내적)**

1. 두벡터를 내적한 결과는 스칼라다.
2. 두벡터를 내적한 결과가 0 이면 두벡터는 직각이다. 음수면 둔각(90도 초과)이다. 양수면 예각(90도 미만) 이다.
3. 정규화된 두벡터를 내적한 결과는 두벡터의 각차에의한 Cos결과값
4. 만약에 하나만 정규화가 되어있고 하나는 정규화가 되어있지 않을 때 결과값은 정규화되지 않은 벡터를 정규화된 벡터에 투영시켰을 때 나오는 투영벡터의 길이

**A** 벡터를 **B** 벡터에 투영시킨 투영벡터 공식

->  $A \text{ Proj } B = (A \cdot B) / (B \cdot B) * B;$

만약 투영기준의 벡터인 **B** 가 정규화 되어있다면.

->  $A \text{ Proj } B = (A \cdot B) * B;$

아크코사인 계산할때는 -1~1사이 값을 사용해야 하는데... 부동소수점 오차로 인해서 1이 넘어가는 경우가 있다. 그래서 **Mathf.Clamp**를 사용하여 값을 걸러준다.

보간

Vector3.lerp( A, B, Factor)      벡터 선형 보간(위치 벡터에 써라)

DestoryImmediate() -> 사형선고가 아니라 즉각처분.

**prefabs** - 자원화된 게임 오브젝트들.

**150508**

모듈화 : 너무 상세히 해도 문제지만 적당히 하는 것이 좋다. 단 내가 관리 할 수 있다면 많이 하면 좋다.

트랜스폼.트랜스레이트: 기본 세팅이 자신 로컬.

충돌체크하는 바닥은 박스로 작업할 것. 쿼드나, 플랜은 비추천. 두께가 있어야 한다.  
**tip)**바닥에 오브젝트 배치할 때 **w** 누르고 컨트롤 쉬프트 유닛 배치.

빨간오류는 계속 뜨면 심각하다. 잡아야한다.

**GetCompants** : 이용해서 배열 받을 수 이다. 근대 아무것도 없어도 0개를 받아온다.

**Broadcast** : 어쩌다 한번 하는것이 좋다. 생각보다 무겁지는 않지만 문자열 비교가 들어간다.  
너무 자주하면 좋지 않다.

**Quaternion** : 회전행렬이지 이동벡터는 아니다.

\*사원수의 곱은 회전한 상태의 기준을 가지고 회전한다.  
회전 곱은 돌리고 나고 또 원 기준을 가지고 돌린다.

## 150511

----- 생성자 -----

Awake            - 인스턴스가 만들어질 때 (생성자) : 엄마 배에  
OnEnable        - 활성화될 때 : 취업

----- update 테이블 -----

Start            - 첫 취업하고 한번 실행됨 : 엄마축하

Update- 주간

Late Update    - 야간

Fixed Update   - 외주

OnDisable      - 퇴사 - 백수 - 재취업하면 ->OnEnable -> 이후 start 무시

OnDestroy      - 관속으로 들어감. 만약에 백수상태가 아니면 일단 onDisable으로 간다아

무언가 Update 먼저 싹다 (모든 인스턴스) 실행해주고. 그다음 LateUpdate 실행된다.

이 개념은 모두 스크립트 인스턴스 기준으로 한다.

OnEnable ~ OnDisable 없애면 활성화 체크박스 않나옴..

@멀티탭

에디트모드에서 비활성화 된체로 게임중 실행된다면

Awake는 실행되지 않고 최초로 play 이후 최초로 Enable 될때 실행이 된다.

150512

<http://tf3dm.com/> 모델링 사이트.  
<http://cgpersia.com/> 에셋도 있는 찌는 사이트.  
ripper 다이렉트 리퍼 해서 obj 받는..  
<https://www.yobi3d.com/#/>

캐논을 로컬로 바디를 Yaw로 돌리고.. 헤드를 pitch로 돌려야한다.

중요 아래는 **nowSpeed**가 **deltaSpeed**의 변화값으로 **targetSpeed**를 따라가는 로직.

```
void ControllSpeed(float val)
{
    //타겟의 속도(맞춰야할 소곤)
    //일단 타겟속도에 음수는 무시
    float targetSpeed = Mathf.Lerp(this.minSpeed, this.maxSpeed, Mathf.Clamp01(val));

    //이수치는 현재 전함의 초당 speed 변화값.
    float deltaSpeed = 0.0f;

    if(val > 0.0f)
    {
        deltaSpeed = this.speedAccelate * val * Time.deltaTime;
    }
    else
    {
        float delta = this.speedRelease + (this.speedBreak * (val * -1.0f));
        deltaSpeed = delta * Time.deltaTime;
    }

    float dist = Mathf.Abs(targetSpeed - this.nowSpeed);
    float t = Mathf.Clamp01(deltaSpeed / dist);
    this.nowSpeed = Mathf.Lerp(this.nowSpeed, targetSpeed, t);
}
```

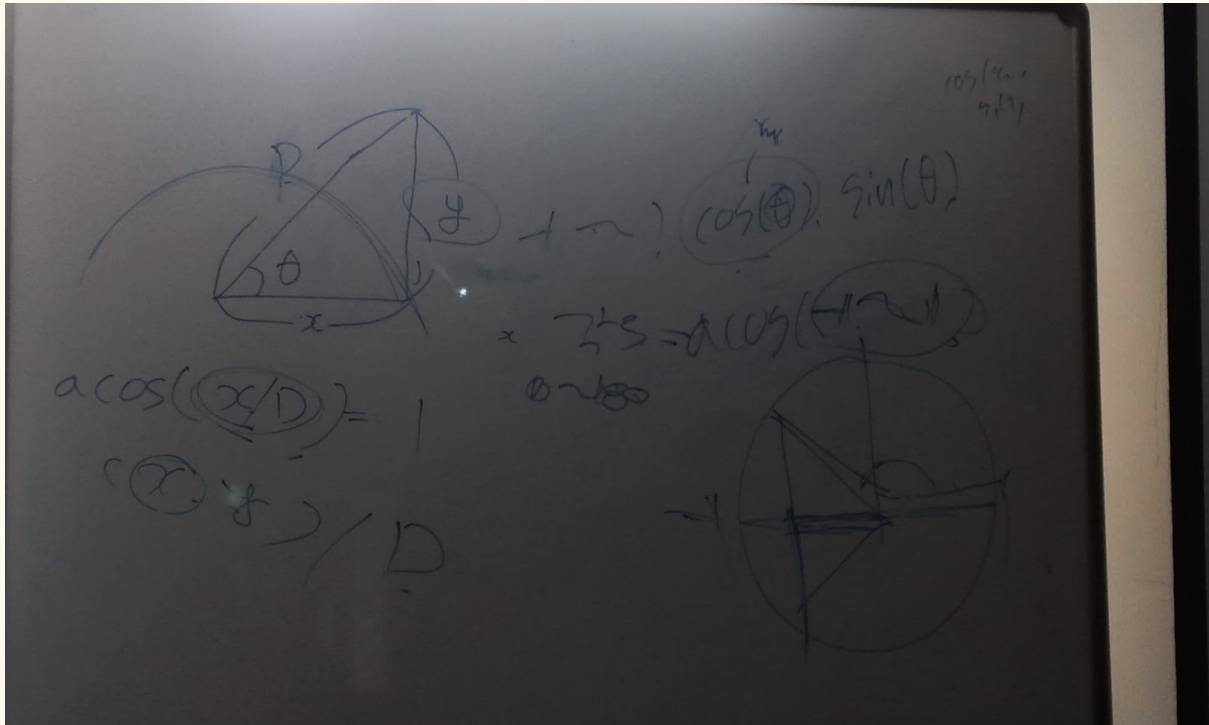
150513

**AngleAxis** : 해당 벡터(코사인, 그래프 기준)의 기준으로 원하는 각도만큼 돌린다.

ex) AngleAxis(10, vector(0,1,0) 이면 업벡터 기준으로 10도

**InverseTransformDirection** : 다이렉트에서 D3DXVec3TransformCoord

- World 좌표계를 기준으로 하는 벡터를 Local 좌표계 기준으로 벡터로 변환



// 옛날 api할때 -> 이게 싸인그래프 도형..

x = 코사인(각도) \* 속도

y = 싸인(각도) \* 각도

피타고라스 정의..

$$x^2 + y^2 = D^2$$

Acos = 각도에 대한 x 비율.

Acos( -1 ~ 1) 각도가 나온다.

0 ~ 180도밖에 못구한다. 반바퀴밖에..

acos(x/d) = 각도

y값이 위쪽으로 갔냐 아래로 갔냐..

매쓰에프가 라디안값을 뱉으니깐.. 디그리로 바꿔주자 로테이션쓸꺼면..

바디와 해더의 위치. 조절해야한다. 가능자.. (아니면 피벗을 바꾸자)

콘솔 에러는 nana 나오면 제로값이 나온다.

**lineRenderer**에 use world Space 체크 해제하면 로컬로 부모

같은 프리팹에 있는 자원들끼리는 물리고 가능하다.. 웬만하면 물리는건 다 프리팹에서..  
씬에 있는건 물리면 다시 불러와도 적용 안되어 있다.

숙제 **1** // 반대로 도는 문제.. 쿼터니안 보간을 하던가, **-90 = 270**도 이런 특징 이용.

숙제 **2** // 우주선에 캐논을 달고 근처에 운석을 처리해라..

도전과제 // 운석들이 한방향으로 움직인다.. 예측사격..

**target** 위치, 타겟 방향, 타겟 초당이동속도, 건 위치, 건 발사속도, 건 **front**

제 **2** 코싸인 법칙..

**d**제곱 = **a**제곱 + **b**제곱 - **2ab**코싸인(세타)

건과 타겟의 방향벡터.

타겟의 초당 이동속도

총알의 초당 이동속도



## 150514

모바일도 해상도 지정을 해서 작업해야한다..  
게임에서 프리에셋 에 있는거 작업.

`_transform` 클래스로 받아서 쓰면 오버헤드가 아주 조오금 차이난다.

최상위 스케일은 웬만하면 건들이면 안된다.

**Coroutine** 코루틴 완소스킬 : 대신 주 프레임을 넘어서 호출할 순 없다.

`StartCoroutine();`

`IEnumerator`

`yield return`

`WaitFrSeconds`

델리게이트

순수가상함수

`OnDrawGizoms` 는 에디트에서 수정됨. `while(true)` 조심

기본적으로 내가만든 클래스는 인스펙터 뷰에 노출되지 않는다.

내가만든 클래스 위에 `[System.Serializable]` 가 들어가면 노출된다.

## 15.05.15

해당 게임오브젝트가 **activeself** 비활성되어있나

**onGui**는 가비지가 생긴다.. 잠깐 디버깅용으로만 쓰자.

**awake()** 인스턴스 멤버다..

**Manager.Instance** 로 할려면 **start**에서 해야한다.

**Awake**로 할려면 **Scritp Excition Order** 순서 설정이 가능하다.// **Default Time** 에서 마이너스로 작업해라.

**DontDestoryOnLoad** 썬변환되도 내가 디스토로이 호출되지 않는이상 파괴 노노

매니저는 여러방식이 있다. 알아서 잘 골르고. 만약 참조시 생성할려면 아예 게임오브젝트매니저를 만들어서 거기서 최초 생성하게 해주자!

베지어 커브 쓰는법.

**[Ranger( 숫자, 숫자)]** 하면 바로 밑의 변수를 인스펙터에서 스크롤 생김

**15.05.18**

메모리 폴링..

하이어라키 구조상에서 ui 상에서는 누가 먼저냐에 따라서 램스의 차이가 있다.

프로파일러.

**waitForTargetFPS** 이게 높으면 높을수록 좋다. (놀고있다.)

**time ms** = 밀리센컨드(1/1000) 초

옛날 버전에서는 폴링된녀석 트랜스폼 계속 계산하면 비용이 원래 많아지는데, 지금은 버전이 좋아져서 많이 최적화가 되었다.

프리팸에 있는 리소스를 이름으로 찾는방법은 불가능하다.

유니티 프로젝트에 존재하는 폴더 네임.

다음과 같은 특징이 있다.

**Editor** : 화장에디터 관련 **Script** 가 모여있는것.

**Plugin** : 외부 **native** 플러그인(**DLL, Jar**) 들이 저장되어 있는 폴더(건드리지 말아라)

**Resource** : 무조건 포함되면서 동적으로 로딩될 목적으로 저장되는 자원이 있는 폴더.

**Gizoms** : **Gizmo** 그리기위한 리소스가 저장되는 곳.

이름으로 게임오브젝트 동적으로 로딩할때

같은 이름인데 확장자 다른건 조심해라.. ㅋㅋ

15.05.19

충돌 검출. 기본  $(n-1)n / 2 =$  버블소트

버블소트, 퀵소트, 힙소트, 레드블랙트리

\*\*\*\*움직이는 애만 하면 충돌 검사를 그만큼만 하면 좋다

버블정렬(Bubble Sort)

1. 두 인접한 원소를 검사하여 정렬하는 방법.
2. 시간 복잡도  $O(n^2)$
3. 코드가 단순.

퀵소트

1. 주어진 데이터 목록에서 임의의 원소를 고른다. 이 원소를 피벗이라 한다.
2. 피벗을 기준으로 피벗의 앞에는 피벗보다 작은 숫자가 오도록 하고, 피벗 뒤에는 피벗보다 큰 원소가 오도록한다. 필요할 경우 피벗은 움직일 수 있다.

컬리더

1. Sphere - 1
2. Box - 3
3. Capsule - 2 (구두개랑 선을 이은다)
4. Mesh - 4 (매쉬나, 터레인이나 둘다 무거운게 더 무겁다)
5. Terrain - 4
6. wheel

AABB 충돌..(얼라운드 asix 바운드 박스)

- 썸 기술중 a,b 하고 기준을 하나 둔다음 그 기준을 역행렬 둘다 곱한다음. 기준이 아닌녀석을 박스로 만들어서 체크

OBB 충돌(오리엔탈 바운드 박스)

- 투영벡터를 이용(총4번정도) 각각 두번씩(x, y축 기준으로)
- 각각의 중점에서 선을 그음
- 그다음 한축의 x 기준으로 이우고 투영시킨다.
- 이후 그 축의 길이만큼 더한걸 더 길게 나오면 비충돌
- 그다음 한번이라도 충돌이 안되면 될필요없다.
- 3디는 15번 충돌체크.(두개의 축이라도 평행한다면 6번만 해도 됨)  $6 + 9$ (외적)

Is Trigger = 유령충돌체

Rigidbody - is Kinematic : 물리적인 움직임을 사용하지 않고. 행렬같은거로만 움직이겠다.  
이거 끄고 하면 부하가 좀더 심하다. 힘을 줘야 움직이니깐..

레이를 이용해서 고속충돌 체크를 할수 있다. 미리 쏘고 이동후.  
쓸때 체크가 되어있으면

tip) 랜덤으로 확률 뽑을때 서플레이블을 이용한다.

충돌 트리거를 이용해서 원하는 작업을 수행하도록 혹은 애니메이션 ai 처리를 할 수 있다.

### Tigger Event 발생함수 3 총사

아래의 3총사 함수가 호출될수 있는 조건은 다음과 같다.

A와 B가 충돌검출을 한다고 가정하자.

1. A와 B 둘다 **Collider**를 가지고 있어야 한다.
2. 이때 적어도 둘중하나의 **Collider**에는 **IsTrigger** 가 체크되어있어야 한다  
(물리적인 개념으로 보았을때 겹칠수 있는 조건이어야 함.)  
->이건 겹치기를 체크하는 이벤트기 때문
3. A와 B 둘중 하나라도 **Rigidbody** 컴포넌트를 가지고 있어야 한다.  
이때 움직이는 놈 기준으로 **Rigidbody**를 가지고 있어야한다.  
물리엔진 체크는 **Rigidbody** 를 가지고 있는 **GameObject** 기준으로 충돌체크를 하게 되는데. 움직이지 않는 **Rigidbody** 는 해당 **Rigidbody** 기준으로 다른충돌 체크를 할 필요가 없다.
4. 위의 조건이 만족되면 **Event**는 쌍방으로 들어간다.

움직여지는 **Object** 는 **Rigid**를 가지고 잇는 것이 일단 바람직하다.

**Rigidbody**가 없는 오브젝트를 움직이는 것이 **Rigidbody** 를 가지고 있는 오브젝트를 움직인 비용보다 더 크다.

`OnTriggerEnter(Collider col);`

`OnTriggerStay(Collider col);`

`OnTriggerExit(Collider col);`

15.05.20

RigidBody

Mass = 질량

Drag = 저항(떨어질때 저항력)

Angular Drag = 회전저항.

유즈 그라비티 = 중력사용

edit - project settings-phsicy 중력설정

is Kinematic = 강체로 움직일건지, Transform로 움직일건지.

게임 만들때 강체 움직임으로 갈건지 트랜스폼으로 갈건지 항상 결정해야 한다.

강체도 움직이고, 트랜스폼도 움직이는건 부담율이 높다.

벨로시티(속도)는 이동시키는건 쉬운데, 토크(회전)은 계산이 부담이 더 된다.

강체 움직임을 안하더라도 리지드바디는 충돌체크할려고..

interpolate = 아이폰 쓸때 한..

collision Detection = 예측충돌탐지..(웬만하면 하지 말라)

Continuous Dynamic (앤 졸라 무겁다 근대 잘됨. 쓰지마)

Constrains 해당 포지션 값 움직임을 막는다.

캐릭터 움직이려고 할때 포스를 밀어 넣을때 움직이게 할때 축 움직이는거 막아야한다.

1. velocity 이동할때 초당 계산이 되므로 타임델타는 빼주자.

2. force 는 반작용이 다 되지만, 벨로시티를 값을 누적시키지만.. 컨트롤하기 힘들다.

- FroceMode 이용해서 빠르게 가능.
- 하지만 조작하기 힘든데, 이걸 클램프 처리할수 있다.

3. MovePosition : 월드 포지션 대입으로 해야한다.(그냥 트랜스폼 대신하는 이동이다.)

- 가장 큰 문제는 벨로시티값이 않나온다.

FixedUpdate() 에서 Input.GetKeyDown 은 호출주기가 좀 잘못는다.

Torque 는 회전량이 정해져있다. 수정하지 않는이상 일정 이상 올라갈 수 없다.

실질적으로 물리가 들어가는 게임은 연출로하는게 괜찮음..

	질량의 영향을 받을때	질량의 영향을 안받을때
단타로 힘을가할때	Impulse	VelocityChange
지속적으로 힘을가할때	Force	Acceleration

단타 : 어떠한 주어진힘을 Time.Delta 가 적용되지 않은체 Veclocity에 추가

지속 : 어떠한 주어진힘을 Time.Delta 가 적용된체 Veclocity에 추가

## 15.05.21

Collision 은 물리적 충돌정보를 지니고 있다.

Collision Event 함수 3종사

- 조건 강제 충돌이 되는 조건이면.
- 1. 둘다 Collider 를 가지고 있어야 한다.
- 2. 둘다 IsTrigger 체크가 해제 되어있어야 한다.
- 3. 움직이는 놈이 Rigidbody 가지고 있어야한다.(이때 IsKinematic 이 반듯이 해제 되어야 함.)

tip) 현재 버전에서 충돌중인친구가 사라지면 exit 호출이 안되넵!

onCollisionEnter(Collision col)

onCollisionStay(Collision col)

onCollisionExit(Collision col)

리지드바디 두개로.. 움직여서 할수있다.. 코인도저.

콜리더에 있는 매트리얼.. - 진짜 재질 속성..(고무, 쇠, 나무.. 등등)

phyic material

바운드 컴바인. 두개의 바운드중 어떤값을 쓰겠는가..

매쉬컬리더 할때. 컨백스를 쓴다. (리지드바디 넣을때 꼭)

//LOD 쓰는이유 멀리있는애를 굳이 폴리곤을 높은걸 쓸필요가 없기에..

LOD : 캐릭터를 여러 폴리곤으로..

동적 LOD : 자동으로 폴리곤이 (x파일은 동적 LOD 지원)

강제동작을 하는 애들은 컨백스가 필요하다.

강제동작을 하지 않는 정적인 애들은 움직이지 않을때 컨백스 필요없다.

알렉스 배경.

CenterOfMass 잘 설정해야 미끄러지는게 더 자연스럽다.

힌지 조인트.- 와이어 프린트로 앵커가 보인다.

브레이크 포스, 토크 - 정도 힘이나, 각도 힘이나 받으면 부러짐.

relativeVelocity 반작용

contacts 무겁다.. 그리고 계산된것이 아니다.

프로퍼티는 항상 받아놓고 계산하자..

## 15.05.21

항상 충돌 체크할때 충돌체크 되는 친구의 루트를 받아오는 습관.

float DesireFire 욕구를 컨트롤한다.

레이어 - 피직스 매니저에서 추가한 레이어 에 디폴트 해제하면 물리적인거 다 해제됨.

폴링되는 데이터 벨로시티값은 계속 남기때문에 초기화를 시키든 해야한다. 주의.

ex) 폴링되는 코인이 갑자기 어느순간부터 멈출때.

renderer - cast shadow 나로 인해서 그림자를 할꺼냐..

recipt ~ 밑에 생긴 그림자가 생겨질꺼냐.



15.05.26

매터리얼 배열로 되어 있는 이유. 서브셋 갯수에 따라 늘어나기 때문에..

매터리얼당 셰이더 코드가 하나씩 붙는다.

매터리얼 셰이더 - 스탠다드, 모바일은 모바일로 쓰는걸 권장.

렌더링 모드 - 오바큐는 (불투명)

알베도 = 디퓨즈 라고 생각하자. / 더하는게 아니라. 곱한다.

(하얀부분  $1,1,1 * 1,0,0$  (레드)) = 1.0.0 레드

그러므로 이펙트들은 그레이톤(회색톤으로 넣어주는게 좋다)

메탈릭 재질이 높으면 쇠에 가까워 진다. 메탈릭이 썩면 빛을 더 덜 퍼트린다.

메탈릭의 텍스처가 흰색일수록 메탈릭이 더 썩지고 검은색은 더 약해진다.

메탈릭 재질에 스무스와 파워와 관련되어 있다.

스펙큘러로 메탈들을 커버가능..

스펙큘러의 power 가 높다. 폰같은거.. 가족은 파워가 낮다.

파워가 높으면 날카로워 진다.

파워 - 제공근의 수치.. 제공수를 올리다보면... 선형보간이 점점 날카로워 진다.

0~1 값을 제공근으로 사용하면...

빛이 주요하면 노말맵을 잘 써야한다.

노말맵. 빛의 반사 방향을 틀어버리는 것을 할 수 있다.(표면을 울긋불긋하게)

0 ~ 255 -> 0 ~ 1

0.5, 0.5, 1 -> 벡터로 가면.

벡터는 0 은 -1, / 0.5 은 0 / 1은 1;

노말맵 텍스처는 텍스처 타입을 노말맵으로..

그리고 크리에이터 프럼 그레이스 체크를 해제.

(이거 쓰면 흑백 텍스처를 노말맵으로 만들어짐)

하이트맵은 시차로 적용해서.. 실제로 나온것은 아니고.. uv를 틀어버린다.

(요즘 노말,하이트 맵 같이쓰는 추세.. 물론 모바일에선 노말하나도 감지덕지)

새컨더리 맵..

벽화처럼.. 오버랩

렌더링모드 트랜스퍼런트. 하면 투명해진다.

트랜스퍼런트는 알파블렌딩을 쓰는거다. 무거우므로 모바일에서 진짜 줄여야한다.

랜더링모드 컷아웃.

컷아웃은 특정 알파부분을 그리지 말라..

트랜스퍼런트가 더 무겁다.

덱스버퍼도 필요없고.. 아예 그리지 말라니깐.. 어느 알파값 이하로는 아예 그리지말라.

알파블렌딩으로 나뭇잎을 절대 그리기 힘들다. 커아웃으로.. 알파블렌딩은 파티클로..

색을 태우는거면 알파소팅은 신경쓸 필요가 없다,

**Stats..**

**setPass calls** : 드라우프라이미티브 , 드라우셋셋 한개

드라우콜 = 셋패스 콜..

이게 적으면 적을수록 적다.. 처음 하나했을때 많은게 있는건 그림자

**15.05.27**

statistics ->

setPass Call - 드로우셋

원래 엠비언트라이트가 빛이 없어도 어느정도 보정해주는거였는뎡..

다렉트에서 디바이스에 하나 지정할수있는 매태리얼이 하나..

버퍼에 다 쑤셔넣어서 드로우프리미티브 한번에 그린다. 큐브에..

단 움직이는쪽에 불리하고, **cpu, gpu**가 부담된다.

단 무조건 움직이지 않으면 만들어놓고 그려놓으면 매우좋다.

->**static** 체크를 하면 된다.

스테틱 체크하고 게임플레이 해보면 매쉬는 움직여 지지않는다.

**batches** - 매쉬가 생길때마다 합쳐짐?

텍스처 아틀라스.. 텍스처 여러개를 하나로 묶어서 유브좌표로 그려준다.(큐브맨)

이것도 콜이 올라가지 않는다.

같은 이미지에 틴트 색만 바꾸고 싶다면? 버텍스 컬러를 이용할 수 있다.

버텍스 컬러를 이용하면 콜이 올라가지 않는다.

랜더러.. 매쉬 랜더러든 뭐든 랜더러의 조상이다.

랜더러를 통해서 매터리얼의 값을 바꾸면.. 바뀐 대상 매터리얼이 새로 바뀌면서 인스턴스로 생성된다. 콜이 증가.

콜이 증가되는 방법은 노 권장.

**setColor**에서 프로퍼티스로 접근.

에디트쉐이더 - 프로퍼티스

**range, float** 는 셋 **float** 로 접근.

너무 드로우콜을 늘리지 마라. 단 같은거로 콜이 올라가면 줄일 수 있다면 줄여라..

매터리얼의 옴셋을 이용하여 흘러가는것처럼 조절가능하다.

스탠다드에서 - 타일링이 10개라면 노말맵도 10개

쉐이더- 스탠다드 말고, 언릿에서.. 트랜스퍼런트 컷아웃.

언릿이 광원효과를 받지 않는다.

스테이지를 스테틱을 통해서 베이킹가 들어가서.. 차폐.

빛도 물리다... 검은색도 빛을 잘 먹고.. 거울을 손실없이 반사한다.

**SSAO** 스크린세이버 엠비언트 오쿨리전(차폐)

리플렉션 프로브.. 입방체를 찍어서 근처 매테리얼에 영향을 준다. 메탈릭에

움직이지 않는 오브젝트들은 스택으로 두자.

스택으로 해두면 베이킹이 되면서 빛계산량이 저장되는데 양이 엄청 많기 때문에 하드에 저장해둔다.

이후 다시 바뀌면 수정된 부분만 계속 작업되고.

하드 캐쉬에 계속 쌓이게 된다.(최대 10기가)

**prefrenance** 에서 **GI Chach** 클린업으로 다 지우는거 가능.

프로젝트를 종료해도 남아있다.

배치 - 같은 버퍼에서 매쉬 사용하는것.

또 이게 다이내믹 배치가 있다.

버퍼량이 크면 랑하는것도 크다. 그리고 **cpu gpu**의 차이로 기다리는쪽이 비요이 크다.

15.05.28

큐브만들기. 큐브에 색입하기. 인덱스로 하면 버텍스 8 개 단..색들어가면 36개.

소프트 엣지. 곡면 처리. 노말 3개중 가운데 보관하는거.

하드 엣지. 버텍스를 두개 숨겨서 명암을 극명하게.

uv 좌표 주사위 텍스처 만들고 하면 결국 uv좌표 24개 필요하다.

텍스처 오른손 좌표계(OpenGL)을 따른다.(d3 랑 수직반전)

-이유 멀티플랫폼 강조. 결국 셰이더를 모바일, 웹 다른쪽을 다 할려고.. 채택.

-쓰리디상은 왼손좌표계

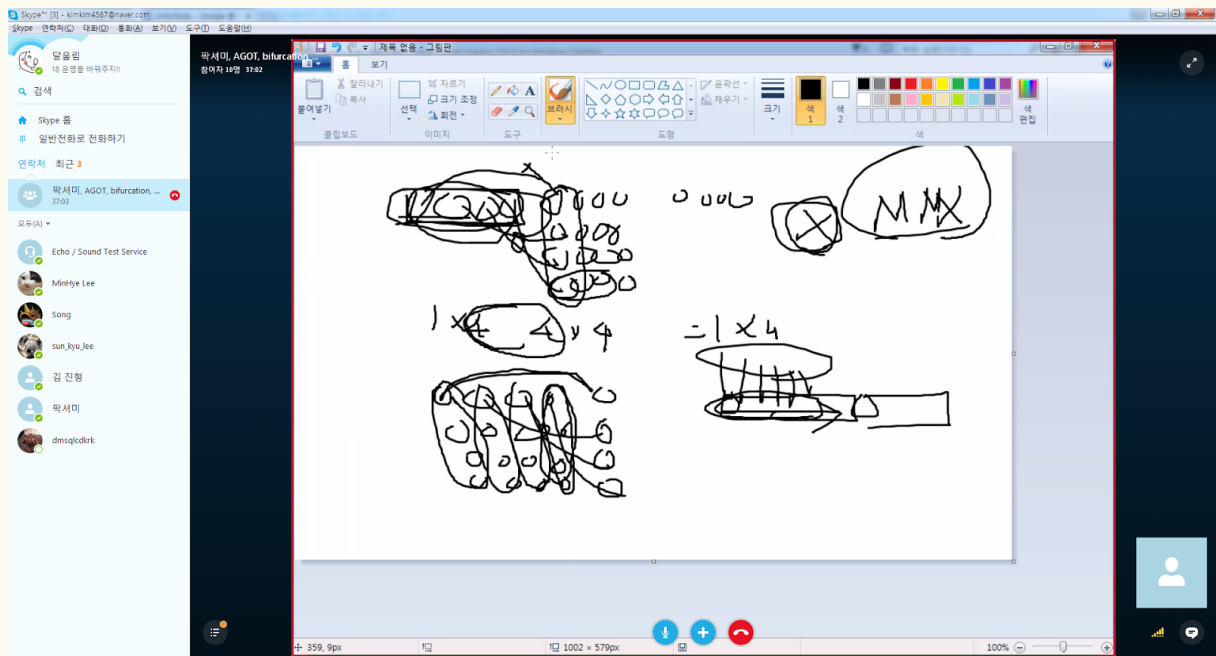
-오픈지엘은 행렬 열 계산

-디스리디는 행 행렬 계산.

MMX 덕에.. 왼손좌표계가 유리해진다.

MATRIX A16. (앤 값이 강 대입이 안되지만, 계산이 행 해렬 계산되면서 무지 빨라짐)

강 매트릭스..(앤 강 대입이 되지만.. 계산이.행 해렬 계산이 않됨)



결론 UV좌표는 반전되어 있다. (마우스도 마찬가지..반전되어있네)

필터모드 텍스처에서.

point는 0.3이면 .05 미만인 색을 가져간다.

biliner 는 uv가 중간에 있을때.. 0.3에서 옆으로 가는(진행방향) 보관한다.

triliner 좌우 상하 보관해진다..(화면상 이게 틀어지면..)

램모드가.. 설정하면.. 1로 넘어가면 1로..썩 늘어져라. 아니면 나머지 계산으로 떨어져서 여러번 그려라. (repeat, clamp) 리피트는 반복, 클램프는 한번으로 보관.

타일링도 결국 거기서 값을 또 곱하기때문에 같은 형식으로 나온다.

쿼드에 외각선이 보이면 클램프로 하면 되는 원리.

**2픽셀**

유브좌표가 **.0.25** 부터 시작해서. **0.75**

정확히 픽셀을 하고 싶다면. 밀고 땅겨야한다.

픽셀이 깨지면... 정확히 절반을 땅기는거 해보기.

픽셀이 **100**이라면 유브좌표를 **0.01**

정점좌표 계산으로 인해서 노말계산하고, 탄젠트도 계산해야하는데.. 유니티**5**에서는.. 된다.

**15.05.29**

매쉬를 만들고 나면 해주자

**RecalculateBounds.** // 바운드 계산(기본 충돌체 대입시 알맞게 설정)

**Optimize** // 최적화해주는 것인데.. 캐시 히트율 높인다, 인덱스 재배치, 쓰지 않는 정점 제거.

캐시 히트율.. ( 시스템상에서 메모리에 자주 쓰던것을 캐시에 담아두고 쓴다.)

**ExecuteInEditMode** - 에디트모드에서도 실행된다.

에디트 모드에서 인스턴스가 생성되고, 플레이되면 생성된에들 지우고, 값만 가지고 있다가. 플레이에 있는 데이터 넣어준다. 그러다 플레이가 끝나면 다시 값을 준다.

에디트모드에서 이거 했을때 **update()** 는 한번 호출되고, 애가 활성화되서 변화가 있을때만 호출된다.

[**ExecuteInEditMode**] 에서 일반적으로 **private** 멤버 변수는 **editor** 모드에서 값이 유지 되지 않는다.(반대로 **public** 은 유지된다)

단 [**SerializeField**] 태그가 붙어 있다면 가능하다.

**Awake** 는 두번. 붙일때 한번, 플레이때 한번.

**update** 는 변화가 있을때만,

**private** 멤버변수는 변경되지 않는다.

기본적으로 주어진 **class** 상태는 **private** 라고 하더라도 붙는다.

디폴트값은 **Awake**와 **start** 사이에서 가져올 것이다.

이미 있는 컴포넌트를 중복으로 추가가 안되는 컴포넌트를 **add** 할때는 붙여지지 않고 **null**이 된다.

문제가 발생하면 해당 컴포넌트를 **OnDisable**이 된다.

매쉬와 메테리얼은 스택으로 하고, **sheredmesh**, **sheredMaterial** 에 추가한다.

베지어 커브.

- 새그먼트 : 베지어 라인에 몇개의 매쉬가 들어갈꺼냐.

레이싱 게임을 할땐 구간별로 쪼개서 활성화 한다.

민맵 체인

- 2의 승수 로 텍스처를 작업해야한다.
- 텍스처가 샘플링하는데.. 큰 텍스처면 비용이 더 큰데..
- 근대 민맵체인지를 하면 멀리있는애들은 작은 텍스처를 쓰는 기능이다.

## 15.06.01

노말이랑 업이랑 외적해서 구한 노말레이랑 오른쪽이랑 구하면 진행방향을 구한다.

fbx, obj, 3dmax, de 등은 그냥 된다. 드레그하면. fbx가 제일 좋다.

환경을 입방체로 사용하고 초 하이폴리콘으로 해도 상관없다. 비용값 절대 많음  
단 용량이 넘크다.

@단위 설정 : 게임에서 작업하기전 정하고 가야한다.

- 카메라도 니어파도 작업하고, 배경도 그거에 맞춰서 작업해야한다.

표준 - 유니티에선 1scale 이 1m

웬만하면 임포트한 애들은 스케일을 건드리지 말자.

리소스에서 모델정보에서 시퍼란것은 원본 프리팹.

모델, 릿, 애니메이션

본이 없다 (애니메이션 없다 하면) Rig 에서 애니메이션 타입 none으로 설정한다.  
그리고 불러오면 인스펙터뷰에서 애니메이션이 없다.

모델 meshes

Scale Factor 로 크기를 키워준다.

mesh compression - 온은 웬만하면 하지 말자. on 하고 optimzize 같이 쓰지말자..  
강 off 매쉬압축하지말자.

read,write - 버텍스에 접근할 수 있을 것이냐 마냐.

import BlendShapes - 범블비 같은거.. 강 해제해놔 무거우니깐.

제네레이터 컬리더 - 임포트한 모델같은 매쉬 충돌체 쓸꺼냐?

swap uvs - 유브의 체계를 완전히 반대로 할래?..

제네레이터 라이트 맵 UVs - 배경으로 스텍쓸때처럼 사용할때.. 유브이에 차폐적용  
시키고 싶을때. 단 꼭 쓰고싶을때만 무거워지므로.

노말 앤 탄젠트.

노말 - 카쿨레이터.. 몇 도 이상 벌진게 없다면 스무싱 처리를 하겠다..

스무싱이 떨어지면 버추어파이터처럼 각지는데 버텍스가 더 올라간다.

그냥 평균 임포트로 두자. 탄젠트도 마찬가지로.

임포트 매테리얼즈 - 가져온 메테리얼 사용한다.

fbx와 텍스처들 데리고 함꺼번에 같이 임포트 시켜버린다면.

명확한 디퓨즈들은 다 이런식으로 가져오면 웬만한건 다 몰린다.

맥스 에서 fbx 임포트

인클루드 - 카메라,라이트,메디아 강 끄자.(유니티에서 애네 대려오면 다 적용됨)

지오메트리 - 본에 있는녀석들을 더미로 쓸꺼냐?



맥스에서 익스포트 하기 전에, 커스터마이징-> **units setip** -> 디스플레이 유닛 스케일  
-> **Metric** 체크 후 단위설정 한 후 시스템 유닛 셋업 -> 1유닛당 1 메터로 설정되어야 한다.

**exoprt**에서 유닛에서 - 오토메틱 체크 해제하면 컨버트하면서 스케일 팩터가 수정됨.  
그냥 오토메틱 체크하고. 스케일팩터가 1인지만 확인하자.

화백님들께 항상 내가 모델에서 뭔가 쓸것같으면 분리해서 만들어 달라고 요청하자.  
- 예를들어 자동차 문짝 바퀴 같은거.

배경은 특히 다 짤라야 된다. 웬만하면.. 차폐 컬링이라든지 해야되기 때문에..  
너무 디테이하게 말고 ㅋㅋㅋㅋ

## 15.06.02

맥스에서 애니메이션 익스포터 하는 것  
애니메이션 레가시, 제네릭, 휴머노이드 ( 레가시 - 부하가 적다.)

레가시

- 다이렉트때 했던 애니메이션이랑 비슷하다.
- 제약이 많지만 가볍다.
- 성능은 좋지만.. 다 다렉처럼 노가다.

휴머노이드

- 좀 무겁다..
- 다 잡아준다 발 위치, 죽는 위치.

다이렉트에서는 본의 가중치가 최대 4개. 많으면 많을수록 부하가 심해짐.

실무에서는 애니메이션이 다 분리되어 있다.. 웬만하면 ㅎㅎ

컨버터 디포밍 더미 투 본 - 더므들은 본으로 볼랭..

만약 스펙쿨러 사용하는 맵은 셰이더에 - 스탠다드 (스펙쿨러 셋업)  
(기존 스탠다드는 메타릭 셋업)

스키닝 매쉬가 있는 모델엔 대부분 더미가 있지.  
더미루트 - 본.

Rig-> Legacy 으로 -제네레이션 sotre in root 으로 수정.

애니메이션 컴포넌트랑, 애니메이터 컴포넌트는 다르다.  
애니메이션 컴포는 대부분 레가시.. 애니메이터 컴포넌트는 제네릭, 휴머노이드

IKE(자식이 부모에게 영향을 미치는 기술)  
바이패드는 본보다 좋은게 다른 부분에 영향주기 쉽다.

//레가시 - 애니메이션 별로 fbx를 두는 방법.

CatGirl.fbx : 모델만 있는 정보.

CatGirl@Attack.fbx : 컷걸이 사용하는 애니메이션 어택.

.. 이런식으로 유니티에서 지정한 방법.

이친구들을 rig 일단 legacy로 설정.

레가시로 여러가지로 나뉘어진 애니메이션 fbx애들은 다른거에서 사용이 불가능하다.

미싱뜨면 그냥 애니메이션 클릭해서 임포트 눌러서 어플라이하면 다시 적용도 된다.

애니메이션 정보는 인스턴스로 여러개. 매쉬는 한개

애니메이션 클립(원본)

애니메이션 스테이터스(각 인스턴스 별로 다 독립적이다.)

블렌딩은 무거운 기술이다..ㅠ

**wrap mode**을 이용하여 애니메이션 셋팅 어느정도 뒀. 루프나 이러거.

부동소수점 오차때문에 **Mathf.Approximately** 를 이용한다.

애니메이션 레이어..

걷고, 아이들, 뛰기는 최하단,

공격같은거 중단,

히트같은거 맨위.

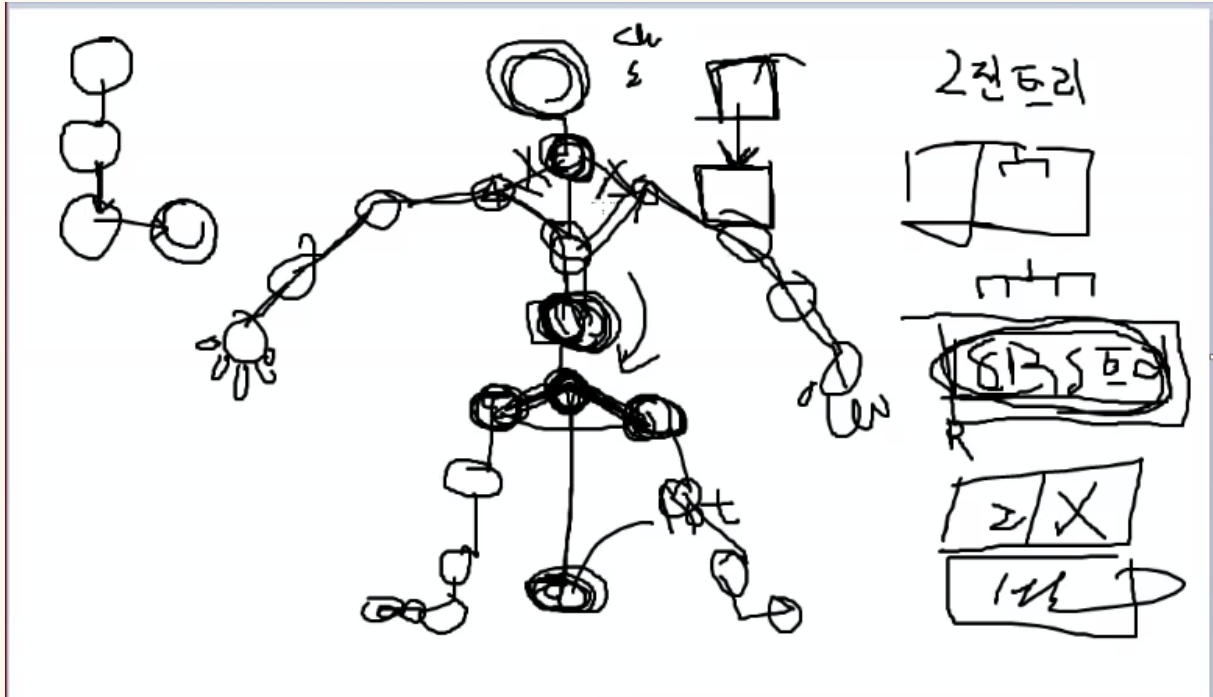
업데이트와 **late**업데이트 사이에

애니메이션이 업데이트 된다.

## 15.06.03

애니메이션 믹스.

LCRS 트리.. 본의 구조.(엑셀 매크로나, 하이ера키 구조에서 많이씀)  
믹스트랜스폼 할때 어디까지 영향을 줄거냐를 알아야한다.



addmixingTransform. , removeMixingTransform.

이 두가지가 될려면 레이어가 꼭 나뉘어져 있어야한다.

위에 레이어가 애니가 멈추면 아래가 실행됨.

PlauQueued /queuemode - 커맨드 만큼 하는 컴플리트거나, 웨이크 넣을 수 있는 now 기능이 있다.

애니메이션 스케줄이 일반 플레이나, 크로스는 하나만 강 하는데

컴플리트나, now는 쌓이는데, 컴플리트는 새로운게 더 쌓이고,  
now는 그냥 그 시점에서 덮어진다.

크로스페이드큐드 - 부드럽게 다아다아다아다 탕.(격투게임에 유리)

플레이큐드 - 권총 연발해도 다다다다 탕

애니메이션 event time 프레임이 30프레임에 1기준이다.

- 1.컬리더는 항상 비활성화 되어 있고, 애니메이션이 시작되면 활성화 시켜서 체크한다.
- 2.한놈만 주고싶으면, 레이어로 활성화시킵 캐릭터 앞으로 레이를 쏘서 체크한다.
- 3.활성화된 컬리더로 반경으로 컬리더에게 전달.

속제1. 공격해서 데미지를 전달하자.  
속제2. 한번만 하면 된다. 왼발 ,오른발.

캐릭터 컨트롤러. - 캡슐모양으로 되어있고.. 블러킹에 특화되어 있다.

slope Limit - 경사 몇도까지 올라가냐.

step offset - 계단올라갈때 몇높이까지 무시하고 강 올라가냐.

skin Width - 밀때 좀더 밀어냈기는 작업.(이걸 이용해서 벽에 끼는걸 막을 수 있다)\_ 백분율

벽에 끼는 상황(점프중) 점프중 앞으로 못가게 하고, 마찰력을 엄청 짝게 해야한다.

캐릭터 컨트롤러는 물리가 아니라서 굳이 **fixed**에서 할필요 없다.

굳이 리지드바디나 물리 영향가는거 받을 필요도 없다.

## 15.06.04

mmorpg camera 센스티브 y 마이너스로 받지 않는 이유.. 반대로 내려가야한다.

맥과 윈도우는 휠방향이 다르다.

Camera.main / 하나만 되어있으면 개가 메인카메라,  
태그가 main카메라로 셋팅되어있으면..

사원수간의 차 Quaternion.Angle (최단거리의 차 인데.. 애매한게 룰도 적용되고 하기때문에)

lighting - continuous Baking - 내가 체크할때마다 베이킹 한다.

height map.. - 쿼드트리 채킹하면서 쉐일 정점 갯수 체크하기 위해서.. 정점 한개를 더 늘린다.

터레인 기어에서 - Detail Density 폴의 양..  
나무는 캡슐컬리더만 적용된다.

-포스트 이펙트-

HRD - 원래 강원의 빼기 하면 빛이 하얀색 이상 못가는데.. HRD 적용시키면 하얀 빛 이상의 정보를 가지고 있다.

카메라 - 랜더 패스.. -deferred 적용시키고.  
카메라에 HRD 키고, 후처리 이펙트를 붙인다.

원래 맵 스테틱으로 두고..  
매테리얼도 셰이더를 스탠다드

-뿔스오브 필드-

초점 중심 외각으로 블러처리 하게 한다.

## 15.06.05

미리 계산해라. 다이스트라. (에이스타)

3가지 요소.

**NavMesh** : ai가 갈수있는 길 (미리 구워진다)

**NavAgent** : navMesh 위를 돌아다니는 애들.

**NavObstacle** : navMesh에 장애물.

일단 준비된 맵주에 스텝 중 - 네비게이션 스텝은 꼭 체크.

**window** - 네비게이션을 클릭.

이후 **bake** 를 누르면 네비맵이 준비된다.

(단 씬이 한번 저장되어야 한다.)

메쉬데이터에 - 미리 인접버퍼 -1은 인접되어있지 않다.. **adjacnt**

메쉬데이터는 웬만하면 열어보지 말자.

네비에이전트는 - 네비매쉬 위에만 있을 수 있따.

네비매쉬가 약간 정확하지 않다.(맵에 약간 붕 뜬다..)

네비게이션 - 에이전트의 반지름, 높이가 몇이냐.

반지름과 키, 에이전트들의 정보에 따라 네비메쉬가 달라진다.

경고가 뜨면 안되는것도 있다.

제네레이션 오프 메쉬 링크.

점프하거나, 뛰어내리는거 조절

오프매쉬링크는 별로 권장하지 않는다. 정확히 점프하게 하기 힘들다.

내가 어떤 매쉬 지역만 오프매쉬 셋팅하기 싫다면... 스텝을 할때 오프매쉬 링크 재네레이션을 체크한다.

**voxel size** - 아마 갈수있는 길의 체크를 더 자세히 할것이나..

**area** 는 지역을 설정할 수 있다.. 그래서 더 느리게 하거나 이런거..

네비게이션은 콜리더로 하는게 아니라, 매쉬로 만들어진다.

그러므로 물리가 적용되는것도 아니고.. 충돌체킹을 되는것도 아니다..

가능하면 리지드바디와 네비게이션을 같이 쓰지말라.

네비매쉬 에이전트에

라디우스, 하이트는 - 에이전트끼리 충돌부피 거리.

**steering speed**- 이동속도. / **acceleration** - 가속 뿐만아니라, 감속도 마찬가지.

**angular speed** - 회전속도 인데.. 720도가 한계이다.

몸따로 이동따로..

오브젝티클 어보이댄스.

퀄리티 - 하이 퀄 / 우선순위 -

패스파인드

길 다시 가나..

에리어 마스크 - 여기가 무슨 지역이냐..

디자인어 벨로시티를 이용하면 키보드의 입력값이라고 생각하면 된다.

후진할때 계소 뒤로가면 갈 거리가 좀 길면 뒤로 돌아서 가라.. 머 그런거..

벨로시티는 별로인게 누가 밀어도 걸리니깐..

에이전트에 **stop**은 일시정지라고 보면 된다. 패스 기록은 남는다.



## 15.06.08

메카닉

**rig** - 휴머노이드(베이스가 진짜 인간형인거만)

아바타 상태에서 컨피그로 한번 누르고 모두 파랄게만 나오면 된다.

애니메이션 꼬이면 묘한 않돼 자세나옴.

아바타 ( 같은 인간형일때 정보를 다 담을 수 있다.)

- 인간형 각 부위에 대한 정보가 담겨있다.
- 가져다 쓸 녀석도 휴머노이드, 가져올 녀석도 휴머노이드가 있어야한다.

**Has Exit Time** - 이게 종료시점에 조건이 맞으면 나가야한다.(콤보에는 매우 중요)

캐릭터 컨트롤러 붙이는걸로만 일단 반이 먹는다.

마우스 우클릭 블랜드 트리.

애드해서 늘릴때 수치 낮은것부터 높은순으로 위에서 부터 체크.

블랜드 트리는 가급적 사용하지 말라. 무겁다.

인스펙터뷰 부분

트랜스폼 와이, **xz** - 그 축으로 애니메이션이 되는 상태로 움직이는것.

베이킹 인투 포스.

- 루트트랜스폼 로테이션 : 루트의 하위 애들에게 회전량을 다 적용시켜주는다..(루트는 가만히 있다)
- 루트트랜스폼 **y** : 다 잡고.. **y**움직임을 적용한다.. 단 이녀석이 되어야 중력이 적용
- 루트트랜스폼포지션 **xz** : 웬만하면 쓰지말라..

루트어플라이모션 할려면 웬만하면 로테이션, 포지션**y**는 베이킹해라.

트랜스 로테이션

- 베이스 업론 오리지날 - 피봇이 어딜 향할꺼냐인데. 오리지날로하며 실제 방향
- 바디는 - 몸통 기준으로 앞방향

트랜스 포지션 **y**

- **base upon - feat** 으로 하면 공중에서 누워 죽을때 이걸로 설정해라.

트랜스 포지션 **xz**

- **centerofmass** 무게중심.
- 움직이지 않는거면 제자리 있는거면 그냥 **xz** 잡아버려라.

다 **original**로 되어있다면 다 **pivot** 으로 설정되어 이는 것이다.

@ 셋팅에서 트랜지션 듀얼레이션 0이면 거의 오버레이 없이 블랜딩됨. **exitTime 1**

트리거는 진입하고 나서 자동으로 **false** 해준다.

## 15.06.09

**Any State** - 어떤 상태에서도 무엇을 할 수 있게.. 단 다시 글로 들어갈 수 없다..

그냥 이동만하고 물리 체크하고싶으면 리지드바디, 캐릭터 컨트롤러만 붙여도 된다.

**animatorStateInfo**

**layer** 갯수에 따라 인덱스가 증가한다

/ 최적화 한답시고 하지말고 매 프레임 받아야한다. 구조체임.

어떤 애니 상태에서 다른 상태로 전이 되는 과정에선 전이전 상태로 본다.

트리거는 들어가는 중에 **idle** 상태이고.. 가면서 트리거는 값이 해제가 안되는데 **block**이 되면서 **attack**이 값이 들어갈 수 있는데. 리셋 트리거로

어떤 애니메이션 컨트롤러를 교체하고 싶다면 하나를 복사해서 바꾸는게 좋다.

쓰고있던 컨트롤 스크립트에서 넣어주는 애니 변수명들이나 애니 노드들이 동일한 것이 작업 효율이 좋기 때문이다.

## 15.06.10

IsInTransition(0) 현재 애니메이션이 블렌딩 되고 있냐 없냐.(해당 애니노드로 가는중?)

**animator** 업데이트 모드

-노말 : 일반 노말..행렬 업데이트..

-애니메이트 피직스 : 스키닝업데이트가 이루어지는 타임이 픽스드 업데이트랑 같아진다.  
물리를 쓸꺼면 애니메이트 피직스 사용하것이 좋다.

-언스케이트 타임 : 독자적인 애니메이션.. 게임을 일시정지해도 나오게 해야한다.  
(유아이는 다 이걸로 해야한다)

**animator** 컬 모드

- 컬 컴플리트 : 애니메이션은 진행하는데 행렬 업데이트가  
않일어난다.(아이들같은거)
- 어에위즈 애니메이트 : 항상 행렬 업데이트가 일어난다.(공격같은건)
- 컬 업데이트 트랜스폼 : 컬링되더라도 행렬 업데이트 하겠다.

IK pass - 역기구학 설정

OnAnimatorIK - 스키닝, 행렬 업데이트후 할거 다하고 들어올꺼임..

sendmessgae 는 다음 프레임에 실행되므로 애니메이션에서 쓰지말자 웬만하면.

애니메이션 프레임별로 딱 잘 찢랏으면 셋팅 트랜지션 듀얼레이션 보간할 필요 없다 0으로

**15.06.11**

**animator** 컨트롤러 에서 애니메이터 추가되면.. 배열로 알아서 증가된다.  
-레이어 클릭 후 : **weight**값주면 점점 더 가려진다.

크리에이터- 하고 에스 디폴트 가능.. 우클릭으로

아바타 마스크 - 그림 클릭해서 **ik** 셋팅할구 있다. 설정하고  
레이어에 추가할수 있다 마스크를.

레이어 셋팅값에서 - 블렌딩 에디티브 - 다리만 따로, 상체만 따로, 있는 애니들은 에디티브  
오버라이드 - 전체가 있을때 섞을때.

도전과제. - 차지해서 점프하면 더 멀리날라가는 것.  
점프차지 스테이트를 만들어야 할 것이다.

점프턴할때 엑시트 타임 해제해보자.

레이가 거리가 없으면 더 무겁다.

겟컴포넌트인 페어런트 - 나포함 체크

매쉬 나뉘어 있는 파트별 모델링은 본(루트)를 같은걸 사용하니, 하나만 사용하게 하자.

//에디터 사용  
**using UnityEditor // 확장 editor using**  
**scriptableWizard** 상속받아야 한다.

**15.06.15**

레이캐스트 - 물리적으로 움직이는 녀석이면 **fixed update**..  
일반적으로는 그냥 **update**에서 한다.

도전 숙제 - **Refract** 구하기.  
레이어 마스트.**value** 비트값이 튀어나온다.

충돌 최적화 순서.

- 일단 먼저 구충돌.
- 그다음 시야와 방향으로 체크
- 그다음 시야가 개방이 되어 있으면 그 시야방향으로 레이캐스팅
- 단 충돌시 어설픈 부분들은 레이를 2번 더 쏘는다 최대 위 아래.

숙제 - 자동차 **AI**(제대로 만들다면 중앙으로만 잘 간다)  
도전과제 - 인코스 아웃코스까지.  
도전과제 - 충돌체를 피해서 총알이 들어간다.

레이캐스트올은.. 배열로 나오는데.. 반환이 무조건 **0** 이상이다.  
레이캐스트올은 충돌되면 가장 가까운 순으로 배열이 셋팅된다.

레이캐스트 쏠때 안쪽에서 나갈땐 안쪽 충돌체들은 걸리지 않지만  
조금이라도 나와있으면 걸리게 된다.

마우스 이벤트.. (반드시 컬리더가 있어야 한다).  
레이캐스트보다 좀더 무겁다.

--

카메라 - 렌더링 패쓰

레가시 버텍스 릿 - 제일 빠르지만 거지같다. 픽셀 셰이더 적용 안됨.  
포워드 - 픽셀 셰이더까지는 보는거(다이렉트 수준)  
디퍼드 - 렌더투 텍스처.. (이방식을 이용해서 하는것이 포스트 이펙트): 지연 렌더링.

- 포그도 안되고, 안티도.. 다 후처리로 작업해야한다.
- 알파블렌딩 너무 취약하다. 엄청 무겁다.
- 라이트닝 처리랑 포스트 이펙트 작업하기 좋다.

덱스 : **0 ~ 1**

디퍼드 기법은 렌더링 최소 **4**번 들어간다.  
스크린스페이스 - 사워

## 15.06.16

백버퍼 / 타겟, z버퍼, 스텐실..

픽셀 하나당 타겟, 제트버퍼, 존재..

타겟 - 컬러 - a8r8g8b8를 한 픽셀에다가 몇바이트 쓸꺼냐..칼라를..

0xffff~ , 색깔..

1.0, 뎀스(z버퍼) (0~1)

0 스텐실 ( 0 ~ 255)

뎀스가 쓰일때 - 원래 뎀스는 선형모향이 아닌 급격하게 나온다..

그 픽셀 위치에 뎀스보다 작은거난 그려지게 된다.. 크게 된다면 통과 불가능하다.

- zTest를 사용하면 무조건 그린다.
- zWrite를 갱신이 안되고.. 덮여진다.

클리어 플래그

- skybox : 컬러, 뎀스 둘다 초기화(컬러는 그 스카이박스 카메라 방향에서, 뎀스는 1.0)
- 솔리드컬러 : 컬러, 뎀스 둘다 초기화(단 컬러가 설정된 백그라운드 컬러로)
- 뎀스온니 : 뎀스만 초기화
- 돈 클리어 : 둘다 초기화 하지 않는다.

뎀스가 통과하고 못하고 차이를 보면 결국 그려지지 못하고 그려진대로 확인가능.

컬링마스크 - 어떤 레이어만 그릴건지..(충돌이나, 물리체크는 다 된다)

프로젝션 - 원근,직교투영

필드오브뷰 - 화각.(줌땡길때 화각을 좁힌다)

(레이싱게임에서 속도감 나게 하고싶으면 화각을 높여주면 좋다)

프로젝션 동차공간 찌브시키면서 퍼지고, 사각형은 찌브시켜도 그대로된다.

직교는 0안되.. 원근만 되 near값

뷰포트 렉트 이용해서 화면 두개 나눌수 있다.

카메라 뎀스 - 카메라 그려주는 순서(낮은값을 그려준다)

카메라 뎀스를 이용하여 ui를 만들수 있다.

월드마크 - 어디서든 보이는 건물같은거

카메라 뎀스와 레이어, 컬모드를 이용한다.

유아이 할때 실제 월드에 있으면, 우주 저멀리 날리거나, 다른 카메라에서 그리지 말라.

랙돌과 애니메이션이 있다고 하더라도 애니메이션이 우선권이 있다.

**15.06.17**

파일 - 고급저장옵션 - cr rl

랙돌처리로 죽일때, 죽는 애니메이션 일정부분 진행시켜놓고 랙돌 활성화를 보여준다.

카메라에 스크린포인트투레이(인풋.마우스포지션); //실제 화면에서 체크.

어떤 오브젝트를 선택하고 컨트롤 + 쉬프트 + f 적용하면 보이는대로..  
혹은 오브젝트면 카메라라 방향으로.

애니메이션 처리, 총쓸때, 맞을때는 헤드 피봇으로.  
마우스 움직임은 그 상위 트랜스폼으로

로컬상에 값을 그대로 벨로시티 적용하면 안되므로 트랜스폼 디렉션으로 돌려재끼자.

인풋매니저 에서  
네가티브 버튼 - 은 갓 엑시스에서 얻어올때 -1~0~1 사용  
포지티브 버튼 - 강 버튼만 누른다면 여기다가 쓴다.

캐릭터 컨트롤로 하면 캡슐컬리더 높이 낮춰도 마추기 힘들것이다.

**15.06.18**

커스텀 애니메이션 만들기  
윈도우에서 애니메이션 창 클릭.

작업 시작하면 녹화중 버튼 클릭  
빨간줄에서 원하는 포지션 작업 혹은 이것저것..넣어주자.  
월로 줄일수 있다..

애니메이션이 적용되면 월드값으로 이동되니깐  
로컬로 두고 싶으면 자식으로 들어가서 로컬처럼 움직이게 하자.

그라디언트.. 위에는 알파, 아래는 칼라.

애니메이션 클립에 커브값 넣어서 그 값을 파라메타로 이용할 수 있는데, 그 값으로 충돌  
활성화 시키거나, 파티클 뿌리거나.. 그런 것에 이용이 가능하다.



**15.06.19**

에이전트 패스페이딩이랑 리마인 디스턴스 같이 체크해줘야한다.  
재수업으면 계속 가만히 있을 수 있음.. 디스턴스가 잠깐 튀수 있다. 페스 페인딩 상태가  
애매해서.

우선수위 큐.  
리스트는 (벡터) 역순으로(인덱싱으로 빨때) 항상 빼자.

링크드 리스트는 포이치.. 리스트는 100개 미만일때 강 쓰라.

코드 괄호가 많아지면.. `//end for` 이런거 써주자.

스페이스 상태를 만들때. 애매한 기준으로 잡지 말자.

이중 레이를 쏜다..

최초 발겨한 지점 기억하고 있다가. 거기서 플레이어 에게 또 레이를 쏜다.  
그래서 찍힌 지점을 두번째 기억할 지점을 또 기억하게해서.  
거기까찌 이동시킨다.

**15.06.22**

책) 절대강좌 유니티3D

스트리밍.. 터레인 여러개 만들어서 활성화 비활성화 시스템 만들어봐야 한다.

2D 게임환경.

텍스처 - 텍스처 타입 - 스프라이트

스프라이트 모드

- 싱글 : 단일 이미지
- 멀티 : 여러이미지를 하나에 담겨서 쓸때.

투디게임에서는 스크린 좌표가 아니라 월드 좌표.

픽셀사이즈 / 100

픽셀퍼유닛 - 100 셋팅 되어 있다.(한 유닛당 픽셀이 몇이나.. 픽셀 비율)

투디쓸꺼면 민맵 체인 쓸필요가 없다.

소팅 레이어 - 레이어로 그룹화하여 지정할 수 있다. 그리는 순서 정해짐.

필터모드는 - 투디할땐 포인트로 그려줘라.

투디는 피지스 투디에 있는 컴포넌트 붙여줘야 한다.

엣지컬리더, 쉬프트 눌러서 추가, 컨트롤 눌러서 빼기.

투디 - 스프라이트데이터베이스.넷 (사이트)

픽셀 최대 인식범이 8192 이다. 넘어가면 잘라서 사용하자.

충돌체크는 3디때랑 조건이 똑같고. 단 트리거체크 함수명에 2D를 붙이면 된다.

캐릭터 반전시킬때 - x축 스케일 -1로

**15.06.23**

## UI

캔버스 - 캔버스 만들고 나는 이벤트 시스템을 지우지말자

(사용자와 유아이가 자동으로 뭔가 상호 작용을 하는 매니저라고 생각하자)

캔버스 랜더모드 - 스크린 스페이스 오버레이(스크린 전체에 그려지는)

랙트 트랜스폼 - 유아이에 특화된 트랜스폼을 상속받은 친구

유아이 이미지쓸때 스프라이트 타입으로 하는것이 바람직하다.

바람개비 앵커포인트- 앵커포인트 기준으로 유아이가 잡힌다.

UI 오브젝트들은 동글뱅이가 활성화 된다. 피봇 셋팅이 된다.

유아이 뎁스 순서는 하이어라키 순서이다.

앵커 포인트가 붙어 있으면 위치기반이 되고, 벌어지게 되면 비율이 된다.

패널.. 패널은 메뉴같은 느낌을 만드는 기본. 윈도우 박스같은 느낌

앵커포인트는 항상 자기 부모 기준이다.

셋 네이티브 사이즈- 원래 이미지 크기로 간다.

컨트롤 시프트 고거 누르면 커진다..

이미지 타입

- 싱글 : 원래 자기 이미지 하나
- 타일 : 타일로 반복적으로 사이즈 비율 맞춰서 그려준다.
- 필드 : 쿨타임.

이미지가 늘어날때 9개로 나눠서 슬라이스

이미지 스프라이트 모드에서 슬라이스에서 보더 셋팅 작업을 한다.

휴대폰 경우는 1280-800 사이즈로 기준을 잡는것이 좋다.

(휴대폰은 해상도는 높은것도 많지만 화면 해상도가 작으니깐,  
사이즈를 고정해놓고 작업하는 것이 좋다)

캔버스 스케일

- 캔버스 스케일 사이즈( 화면 1:1 비율로) - 핸드폰으로 할때 안나온다

- 스케일링 위드 스크린트 (레퍼런스 레졸루션 셋팅 - 고정 해상도는 잡아놓고 해야한다 -  
스크린 매치 믹스 셋팅) 메치는 가로

-> 이친구로 해야 핸드폰으로 거의 비슷하게 잘 나온다.

-----캔버스 랜더모드 - 스크린 카메라.

레이싱 게임같은 유아이 느낌을 넣을 수 있다. 공간적인 개념이 들어가기 때문에.

-----캔버스 랜더모드 - 월드 스페이스.

진짜로 월드에 존재하는 유아이.

다나믹 픽셀 퍼 를 올려주고 폰트사이즈를 줄여줘야한다.

-버튼

- 트랜지션 컬러 버튼 눌렀을때, 아닐때 등등 오버레이드 등 색 지정.

- 스프라이트 스왑 - 이미지

- 애니메이션 - 애니메이션 오토 셋팅하면 일단 기본 셋팅 만들어 준다.

네비게이션 사용하면 키보드나 패드가 선택하게 될때 사용한다.

터치나, 마우스 있을땐 별로 사용하지 않음

`Application.quit();` 빌드하면 종료 되것이다.

버튼 이벤트는 델리게이트 식으로 물리는 것이기 때문에, **public**으로 만들어야 한다.

씬 체인지 할려면 우선 빌드 씬인 빌드에서 씬 등록하고 인덱스 번호 알고 난뒤  
어플리케이션 로드레벨 - 인덱스

마우스 좌표 따라다니는 **ui**

스크린 오버레이 경우 그냥 잘되는데,

모든 상황에서는...?

## 15.06.24

유니티엔진,이벤트시스템즈. // 마우스 관련 이벤트데이터 가져올수있음.

**IPointerDownHandler**(인터페이스 클래스 받아야한다)

스크린 오버레이 상태에서 드래그할꺼면 강  
트랜스.포지션 = 이벤드데이터.포지션

자연스러운 드래그 작업  
다운 되었을때 오프셋을 기억하고 같이 움직여 준다.

캔버스 그룹화 하는것 - 컴포넌트에 캔버스 그룹

자물쇠 기능

드래그 호출 순서 - 가장 마지막에 찍힌 부분이 활성화 된다.

```
델리게이트 {  
    //이사이에다가 구현해도 되고 붙여도 된다 함수 호추을..  
}
```

**15.06.25**

먼저 클릭한 부위 너석으로 드레그가 상태가 들어온다.

비교하는자가 되는 경우는 컴페어러를 상속받아 쓴다.

팔로우타겟유아이 에서-

만약 오버레이가 아닌 상태로 캐릭터 이름 띄울꺼면.. 나중에 스크린포인트의 값으로  
렉트유틸리티 사용하면 된다.

카메라 니어값이 너무 땅겨도.. 파에 의한 지파이팅이 자주 일어난다..  
웬만하면 자주 땅기지말자.

텍스트

윈도우 - 시스템즈 - 폰트

라인 스팅 - 글 간격(위아래로)

호리존탈 오버플로우 - 넘어가도 표현.(옆으로) 어랩은 아래로

컨텐츠 사이즈 필터(레이어에서) 추가하고 - 프리필드 사이즈  
글이 넘어가도 사이즈를 수정해준다.

윈도우 사이즈 잘못게 글이 깨지면,, 한번 실제 사이즈로 키워보자 게임 화면을.

**15.06.26**

자바설치.

자바 다운받기. - 오라클사이트 - 다운

넥스트 다음 - 체인지

- 폴더명 띄어쓰기 하지말자.
- 체인지 이후 씨\_자바\_
- 넥스트
- 두번째거는(JRE) 상관없지만 걔 자바 안에 파일 만들어서 같이 설치

안드로이드 sdk 다운받기.

-

유니티 안드로이드 환경 만들기  
스위치 플러그인 해주고.

-플레이어 셋팅

커서햇스팟 - 클릭 위치를 어디를 지정해줄 것이냐.. 마우스 끝 포인트를 0.0(기본으로)으로

리소스앤드 프리젠테이션  
오리엔테이션

- 포트레이트 세로로(수직으로 디폴트)
- 랜드스케이프 가로로(레프트가 디폴트)
- 오토로테이션은 막 돌린다.

오토로테이션 쓰면, 얼로우 오리엔트 셋팅할때 같은거 두개만..

돌리는 작업은 디바이서를 죽이고 다시하는것이니.. 진짜 조심.

스테이터스 바.. 히든 위의 상태창(끄면 그냥 보인다)

유즈 32비트 디스플레이 버퍼(이거 끄면 픽셀하나당 4바이트 된다)

-저가형이나 갤럭시 s2 이전 제품들..은 그냥 꺼주자..

버퍼 문제는 화면은 보이지 않고 소리는 들리고,, 머 이런거.

밑에 댕스앤스 스텐실은 쓰지안겠다는 거.. 걔 꺼놔 할일 없다.

돈쇼우. 로딩바 기본있는거 쓸꺼냐.. 걔 쓰지마..

스플레쉬 이미지.. - 이거 무료버전은 안된다. 회사 로고 띄울수 있는 작업..

## 아더 셋팅

랜더링 패스 - 카메라에서 쓰는것.. 강 포워드

멀티쓰레드 랜더링 - 캐릭터가 갑자기 확찢어짐.. 번져..

그럼 멀티쓰레드 랜더링 문제 웬만하면 꺼라.

스태틱 배치, 다이나믹 배치 - 배치(드로우콜 줄이는) 웬만하면 다이나믹 배치 풀어준다.

지피유 스키닝 - 지피유가 셰이더 스키닝 해주는것(중요함)

## 아이덴티파이케이션

번들 아이덴티파이 - 회사이름, 프로젝트 이름.. 정도..(앱스토어에선 다 다르다)

버전 - 1.0 버전(게임버전)

버전코드 - 1씩 증가.(무조건 강 1씩 증가)

버전업해서 하게되면 깔려있는사람한테 새로운 버전 파일이 된다등 앱스토어 갱신되고

미니멈 에이피아이 레벨 - 최하위 버전이면 상위버전 잘 된다.

어떤 기능들 추가하면 어느 버전이상 요구하는 것도 있따.

## 컨피규레이션

활성화 되어있지 않는건 프로에서...

그래픽 레벨은 - 자동으로 강 해놔..

디바이서 필터 - **FAT**

인스톨 로케이션 - 어디에 깔꺼냐 피리펄 익스테널

인터넷 액세스 - 강 오토로

라이트 액세스 - 저장 어떤식

## 옵티마이즈

닷넷 2.0 셋셋만 해도 충분하다.(씨샵에서 되다가 머 유니티에서 유징어쩌구 않나와 이러면 강 닷넷 사용해보자)

퍼블리싱 셋팅 - 키스토어(앱스토어에서 올리는거 키 발급받고 넣어줘야 한다)

## 프로젝트 셋팅

퀄리티에서 - 돈 씹크(다이렉트나 웬만하면 다 잡아줌) pc면 해주자/ 프레임 잡아준다.

성능 40%저도 향상된다.. ㅡ.ㅡ

안드로이드는 무조건(이버 **V Blank**) 해줘야한다.

폰에서- 홈버튼, 포어그라운드, 백그라운드.



**15.06.29**

**acceleration** 지구 중력 기준으로 그냥 휴대폰 화면 위로 보고있으면 0,0,-1 이며  
오른손 좌표계이다.

자이로는 아이폰만 된다..

터치페이지 이넘 - 중 캔슬은 터치 잘 들어오다가 나간경우

터치.포지션 - 스크린기준으로 백터2 나온다.

터치.핑거아이디 - 터치된 핑거 정보?

터치정보가 10개이상은 안들어 올것이다.

## 15.06.30

캔버스에 - 그래픽 레이 캐스팅  
랜더링 되는 영역만 체크해주면 된다.

카메라에다가 이벤트 - 피직스 레이캐스트(원근투영), 피직스 2디 레이캐스트(직교투영)  
이용하고 유아이에 컬리더를 주고 체크를 해줄 수 있다.**event** 호출이 가능해짐

파일세이브

`PlayerPrefabs.SetFloat("~/", 1.0f);`  
이런식으로 스트링, 플로우, 인트 저장가능.

메모리 형식으로 저장하려면 클래스로 가야한다.( 참조형)

## 15.07.01

### try catch

트라이 캐치 할때 캐치는 여러번 할 수 있고, 캐치 조건문에서 **exception** 은 항상 맨 아래의 조건으로 캐치를 사용하자.

자바같은 경우에는 스로우 던지면 무조건 트라이 캐치를 해야하는데, 씨샵은 던져도 받지 않아도 된다.

캐치마지막에 들어가는 **finally**  
파이널리는 트라이에 한번 발을 담구면 무조건 실행된다.

역슬래쉬를 경로로 쓸려면 두번 써야한다.  
골뱅이 쓰고 강 쓰는건 시샵스타일 경로이다.

파일모드.open 밑의 것들은 그냥 신경쓰지말자.

엔터를 넣어주고 싶을때 \n 해야한다..  
엔터는 캐이리지리턴이랑 줄바꿈 두개 명령이기 땀에.  
(줄바꿈은 앞으로 가서, 캐이리지는 그만큼 가서)

일단 모르면 UTF-8

xml 파일은 추가하면 솔루션 앞에 있다.

xml 작성시 무조건 선언 노드가 나와야 한다. 띄어도 안되고 무조건 붙어서 해야한다.  
<?xml ~~~~ ?>

꺾쇠 하나 시작이 노드들이고  
그안에 있는 것들이 엘리먼트들이다.

트리구조처럼 구성된다.  
루트 엘리먼트 무조건 한밖에 될수가 없다.

xml 로드방식 크게 3개 / 파일,돔도큐먼트,웹 3가지 방식.

돔도큐먼트 - 너무 느리다. 별로 임.

../ -> 한경로 올라가는 방식.

## 15.07.02

bar strieam 같은거.. 아무튼 어떠한 형식에 의하여 바뀌는건데..  
당황하지 않고 xml 형식으로 맞춰주기 위해 sting을 수정 작업을 해라.

asp.net 을 통해 간단한 로컬 호스트를 만들고  
그곳에 등록된 xml 불러와서 읽을 수 있다.

xmlReader.ReadString() 현재 읽고 있는 노드의 자식 텍스트, 노드의 value

int.Parser(string); // 인트형 정수로 변환한다.

xml 을 유형이나 기본틀을 그냥 만들어서 주거나, 스키마를 만들어서 엑셀로 넘겨줘도 된다.

기본틀을 만들어 줄때 2줄이 있어야한다.  
그래야 리스트로 인식한다.

엑셀에서 - 옵션 - 개발도구를 셋팅해준다.

유니티나, 안드로이드 같은경우에는 저장 경로가 지정되어 있다.

xmlWriter.wriestartelement, endElement 가 스택이라고 생각하자.

예제 24. 스트링 빌더. 이제부터 스트링 빌더를 쓰자.  
많이 쓰이는 중복된 스트링은 테이블로 관리.

프로그래머 주도적 설계.  
데이터 주도적 설계 - 클라상에서 데이터를 계속 로딩해서 바꾸며 테스트 가능하게.

```
Mathf.Acos(  
    Mathf.Clamp(  
        Vector3.Dot(dirToTarget, this.sightTrans.forward), -1.0f, 1.0f)) * Mathf.Rad2Deg;
```

## 15.07.03

유니티랑 연결되어있는 비주얼스튜디오에 솔루션 또 추가 가능하고,  
그렇게 추가된것은 유니티 파일에 그냥 있다.(최상단)

만약에 경로에 C:/ 이런 경로가 없으면, 내 폴더 위치에서 찾는것이다.

**try**가 붙는 함수들은 대부분 **bool**값을 리턴하고 예외사항이 나오지 않는다.

씨샵에서는 인덱서 프로퍼티라는 존재로, 대입연산자로 쓸수 있다.

두가지 방식 **textAsset** 을 써서 빌드해서 어딘가 올릴때.  
그냥 데이터폴더 빌드에 넣고, 그냥 할수있는거에는 개발중일때.

인풋필드 에서  
온 밸류 체인지가 모바일에서는 안될수도 있다.

**15.07.06**

`Applicaion.presententDataPath`. 이거는 피씨 앱 데이터 로컬 쪽에 있고  
안드로이드도 지정된 지역에 잘 저장되고, 잘쓰면된다.

카드보드  
구글 카드보드..QR코드 있는거 사용하기.

!?  
카드보드랑, 드론이랑 연동하기.

폭풍마경.

이아이클럽 OTG케이블.

Cadboard sdk - 유니티 패키지 받기.

알트키를 눌러야 축이 돌아간다.  
넥스케일을 써야 목이 잘라지지 않는다.

**var** 무조건 초기화된 녀석을 담을 수 있는 녀석. (외국친구들은 자주 씀)

눈알 두개라서 드로우콜은 두번은 어쩔수 없지만  
그외 컬링이라든지, 나머지 최적화는 메인 카메라에서의 기준으로 잡는다.

스테레오 컨트롤러 - **direct render** 처리를 꼭해야한다.. 그냥  
매치몬 포브, 매치 바이 줌은 땅기기 한다. 단 않쓰면 땅겨놔라 화각땀에.

만약 줌 땅길꺼면 80도까지만 해라.. 이상은 토솔림.

VR은 UI를 월드로 뿌려야한다.

## 15.07.07

stereo Controller - apply Distortion. - 외곡 시키겠다 인데. 유아이를 다 그리고 맨 마지막에 그리는게 좋다.

dest쪽이 화면 전송해주는 이미지.

offMeshLink

- 코스트 오버라이드.(이길의 비용)
- 오토 업데이트 포지션 셋팅하면 바뀔때마다 계속 계산됨.
- 베이킹 할때 스택에서 오프메시 거시기 해제 하고 하면 선택된 녀석만 지역설정됨.

InverseLerp (시작, 끝, t) 시작과 , 끝에서 t가 어디에 있냐..

**15.07.08**

Jump하는 트리거는 **event** 레이어 하나만 두고서 체크하면 된다.

아이템 체크 이벤트(빨려)는 또 그것과 같이 믹스..허..

동사무소 만들어서 작업해야한다. (갯 컴퍼넌트...ㅠ)

네비게이션의 에리어를 통해서 지역을 설정해보자.

오브젝트 클릭하고 네비게이션에서 오브젝트에서 에이리어 설정가능..

네비게이션 업새트클의 커브르 설정하면 단순 길막는거 빼고, 길 경로까지 막아서 재탐색하게 한다.

각캐릭터마다 에이리어 코스트를 동적으로 설정해줄 수 있다.

인벌스트랜스폼다이렉션 = 내 기준으로 월드 벡터를 내기준으로 다시 땅킨다.

네비매쉬에 대한 최적화.. 일단 평평하고 복층이 아니면 좋다..

**Time.Scaletime** 하면 모든 타임 받는것들이 그 속도에 영향이 감.  
애니메이터에서 **UpdateMode** 를 **nonetime** 하면 된다.



## 15.07.09

### 파티클 시스템

슈리켄, 레가시 - 두가지가 있다.(레가시는 기존의 예전 버전 파티클)

슈리켄-

기본적으로 파티클 맨 위에는 무조건 활성화 되어야 하므로.. 강 디어있고  
그 하위에는 선택해서 원하는 걸 서령할 수 있다.

파티클, 이미션 - 생성과 관련된 부분.

트랜스폼에서 스케일 변환- 셰이프의 크기만 달라진거다. 파티클 입자가 커진게 아니다.

파티클(기본 디폴트, 끌 수 없음)

듀어레이션 : 파티클 주기(기간)

루핑 : 반복할꺼냐..

**prewarm** : 예열.. 미리 끝까지 파티클을 다 뿌려주면서 시작.

스타트 딜레이 : 입자 하나 시작하기 까지 기다리는 시간

스타트 라이프타임 : 한 입자가 나타나서 살아있는 시간.

- 커브 쓰면 듀어레이션에 영향을 받아서 뿌려준다.

스타트 스피드 : 입자 속도

스타트 사이즈 : 입자 크기

스타트 로테이션 : 입자가 해당 값으로 로테이션 되어서 나온다.

스타트 컬러 : 입자 색

- 그라데이션 - 위에는 알파, 아래는 색상.

그라비티 모디파이어 : 색상

시뮬레이션 스페이스 : 월드와 로컬,, (로컬은 자기 기준,, 월드는 간지나게가능함)

인힐 벨로시티 : 시뮬레이션이 월드일때 먹힌다.

- 입자가 생성과 동시에 벨로시티값을 받아서 추가된다.
- **ex** : 비행기 연기나 발자국으로 인해 먼지 나게 할때.

플레이 온 어웨이크 - 아는거.

맥스 파티클 - (에미션의 레이트와 비례해서 잘 맞춰야 한다)

에미션(발생과 관련)

레이트 : 초당 몇개 (타임) / 디스턴스는 - 움직임이 있을때마다 나온다, 시뮬레이션 월드  
설정

버스트 : 타임 주기에 따라 파티클 갯수를 뿌리게 가능하다.

셰이프(생성 모양)

**shap** : 여러개 있다 해보셈.

**emit form** : 시작 위치 설정 ( 베이스 쉘 바다면 시작) = 등등

랜덤 다이렉션 : 랜덤방향..

텍스처의 셰이더 웬만하면 모바일로 통일해라.

알파블렌드 쓸꺼면.. 소팅해서 그래야 잘 정렬되서 나온다.  
알파 안쓸꺼면 덤스를 꺼버려라.

픽셀 1, 0, 0, 1  
픽셀 0, 1, 0, 0.7  
알파 블렌딩 결과값 : 0.3, 0.7, 0  
에디트 결과값 : 1, 0.7, 0

에디티브 - 그냥 전 색을 그대로 두고 나의 색을 알파값 적용 시켜서 ( 불, 빛 ,전기, 레이저)  
알파블렌딩 - 먼지, 폭풍 이런거

사이즈 오버 라이프타임(라이프 시간에 따라 사이즈 커짐)

@ 특히 모바일은 픽셀 처리 같은거 조심.. 파티클이 화면에서 가까워지면 더 힘들어한다.  
랜더러에서 맥스 파티클 사이즈 조절..

랜더러  
소팅모드 (알파블렌딩으로 하면 잘 확인할 수 있다)  
- 바이디스턴스 : 굉장히 무겁다.(입자하나하나를 카메라와 거리 계산해서 소팅)  
- 영거스트 퍼스트 : 젊은녀석먼저 그려준다.(폭포를 밑에서 보면 괜찮음)  
- 올드 포스트 : 나중녀석을 먼저 그린다.(폭포를 위에서만 보면 괜찮음)

텍스처에서 - 파티클 에디트, 알파블렌딩, 에디트(소프트)

쉐이프 밑에부터는 움직이는 중간에 뭔가 제어하는것.

컬리션  
플랜 : 하나 빈오브젝트 트랜스폼 해서 처리할 수 도 있다.  
월드 생각보다 애플무거움.(점 기준으로 한다)  
바운스 : 부딪혔을때 튕기는거.

-코드제어  
플레이 - 듀얼레이션이 초기화가 된다.  
그러므로 주의해야한다 뿌리는 주기 레이트랑 연관도 있음.

플레이 값에서 입자가 하나라도 살아있으면 아직 플레이중.

**OnParticleCollision** 함수가 호출되기 위한 조건

1. Particle 에 Collision 이 활성화 되어있어야 한다.
2. 월드 대상이기때문에 Particle 의 Collision 이 Plane 이아니라 World 로 해야한다.
3. Particle 컬리전에 Send Collision Message 가 체크되어있어야한다.
4. 충돌대상체에 Collider 가 있어야하고 반드시 IsTrigger 가 체크해제 되어있어야한다.

이벤트는 쌍방으로 들어간다.

**15.07.10**

랜더러

랜더러 모드. - 스트레치 빌보드..(이동 방향으로 파티클이 늘어남)

**Invoke** 어떠한 함수를 해당초 뒤에 실행시켜라.

왜곡 처리는.. 셰이더 단계에서 후처리 할때 노말값을 옆에 있는 노말값을 가져오는 식.

**Fx/Glass/Stained BumpDistortion.**

왜곡같은 경우 메터리얼에 접근해서 **Distortion..**제어

파티클 하나를 아예 만들어 놓고. 월드 셋팅후 계속 돌아다니면서 쓸수도 있다. 동적으로..  
ㅎ

**Emit** 자체가 풀이고, **max** 파티클이 풀 개수.

**Texture Sheet Animation** 을 이용해서 텍스처에 여러 겹쳐있는 이미지를 쓸수 있다.

**15.07.14**

파티클 시스템

벨로시티 오버 라이프타임

진행중에 점점 속도를 느리게 하거나 빨라지게.

포스랑 벨로시티는 힘(누적)과 그냥 속도..?

리미티드 벨로시티 오버 라이프타임

댐프는 0~1 사이로 적용해라

댐프는 원래속도에서 까는것이다.

컬리더

바운스를 꺼주고, 샌드 매세지를 활성화 하자. 그러면

쌍방으로(파티클 컬리더와, 부딪힌 상대방)에게 쌍방으로 이벤트를 보내준다.

레가시 파티클

레가시 파티클은 최소한 3총사는 뭉쳐야 한다.

일립스 파티클 이미터

랜덤값은.. 1 으 넣으면 1~ -1 사이값을 현재 벨로시티에 적용시킴.

이미터 벨로시티 스케일 - 인혈 같은거(파티클 시스템에)

랜드 로테이션은 처음에 랜덤 먹겠냐 않먹겠냐..

원샷 - 단타로 뿌려줌..(지속적으로 노노)

오토데스트럭션 - 자동으로 죽음..

일렙세이드 - (api 당시 그리던 렉탱글, 일렙스) 처럼 각 끝의 점이다.

레가시 파티클 - 풀링 - 월드는 사용 불가.

쉐도우 체크 - 프로젝트 셋팅 퀄리티에서 그려준다.

-- 텍스처 쉐도우(크면 클수록 더 잘 그려진다.)

빛의 방향에 카메라가 있다.

덱스가 있는 텍스처 쉐도우 만든다음 그림자가 지어져야할 월드가 동차 계산을 한다. 그래서

비교

vias값이 있어야 한다.

**15.07.15**

1998-2000

글라이드2x - Voodoo

오픈gl

다이렉트x

라이트에 설정

- **baking - baked** 는 실시간 라이트는 안되고 라이트맵을 가져다 쓴다.
- **baked** 되면 실시간 급게 해놓으면 다시 설정된다.

라이팅

- 리얼타임 **gi** 실시간으로 라이트를 구울때 성능 설정.

**baked GI**

- 유닛당 라이트맵 텍스처 크기 결정(낮아지면 더 허접해진다 )

엠비언트 소스 - 모바일은 스카이박스 쓰지말자. 무거워

라이트에서 셰도우 설정해야 그 빛에 대한 그림자 설정됨.

**baked GI**

-**baked Resoluton** 올리면 주변에 대한 **bake** 정보들이 더 좋아진다.

미리 **bake** 된 광원정보르 라이트 프로브로 캐릭터에 빛을 적용시킴.

라이트 프로브 여러개 일때 하나 선택한 상태에서 컨트롤 + 드레그.

코드제어 상으로 **DynamicGI** 으로 적용시킨다.

**effect - projector** 캐릭터레이어를 무시하고, 새도우 메터리얼에 적용하여 사용한다.

프로젝터를 사용하면 모바일에선 비용이 좀 싸다.

이걸 사용하면 바닥에 쓰는데.. 문제는 밑에 닿는곳까지 다 그려진다(디아비블로 장판)

모델에 - 제네레이터 유브 라이트맵.. 설정 무조건 해야 한다. 스택으로 베이킹 할때.

## 15.07.16

에셋 번들. (구글 앱스토어에는 50메가밖에 못받기에..)

**asset package** - 빌드전 자원. 에디터 사용가능.

**asset Bundle** - 빌드후 에셋.

로컬 캐싱 사용하지 하는법, 않는 법.

다운받은 후 이제 다운 안받고 받아진 곳에서 계속 사용  
다운받고도 계속 또 실행때마다 다운 받는 법.

에셋번들 사용시.

에디터를 하나 만들어줘야 한다.

로딩이나 이럴때 자주 쓰는것으로

**start** 함수가 코루틴으로 호출할 수 있다

에셋번들에 스크립트를 적용시키고 가져오면 그냥 안되고, 실행되는쪽도 스크립트 파일을  
가지고 있어야 한다.

만약에 스크립트를 가져온상태에서 가져온 번들의 스크립트가 수정되어 있으면??

가져온 녀석으로 될것이다.. (즉 로딩된 쪽에 있는 스크립트로 작동됨)

**c#** 고급 - 리플렉션

버전정보를 받는 파일은 - 그냥 계속 다운받게 하고

그외 나머지 체크하는것은 - 버전에 따라. 그냥 로컬캐싱쓸꺼지, 혹은 새로 받을꺼지.

캐싱쓰고 안쓰고에서 번호를 체크할수 있다.

**15.07.20**

캐싱해서 쓰는건 대부분 에셋 번들 자료들 일 태고, 그 번들들이 업데이트 되었냐 아니냐는 캐싱하지 말고 항상 체크 받아서 업데이트 해준다.

업데이트 할때는 **xml**로 일단 자료 셋팅해서. 그 **xml**의 데이터들을 비교하여 번들들은 받아준다.

그 데이터들은 로딩에 필요한 자료또한 가지고 있어서. 그 자료를 이용하여 에셋번들을 적용시키는 중에 로딩을 보여줄수 있다.

**15.07.21**

에셋번들로 만들어진 씬은 씬매니저에서 스트링으로 쓰자.. 인덱스로 저장 안됨.

**WaitForEndOfFrame()** . 프레임 거의 가장 마지막 지점. 랜더링 댄 시점.

비동기 로딩을 하면.. 딱 씬준비.. 이거 하고나면.. 그제서야 **awake** 작업함..ㅠ

## 15.07.22

LoadLevelAdditive 는 그 썬에 있는 오브젝트 들만 가져다가 넣어주는것이다.

오디오 클립이 리소스  
오디오 소스가 발생기.. 채널이라고 보면 된다.  
리스너가 있어야 가능.

총소리 같은 사운드는 포스 투 모노 체크 한다.. 스테레오 제거..?  
PCM

## 15.07.23

로드 어썬크 과정에서 에셋번들 받아올때 개입된다..  
로드 어썬크 과정에서 에셋번들 받아오는거 자체..

돈 디스토링드 로드 할때  
그런녀석들은 호출되는 녀석들이 있다.  
OnLevelWasLoaded(int level){}  
level은 썬 인덱스.. -1은 에셋번들..

사운드  
spatial Blend 2d냐 3d냐.  
3d 사운드를 스트리밍으로 하는건 낭비다.

사운드 인스펙터 - 로-타입  
스트리밍 - bgm

돌다 올린다.  
in Memory (압축해서 올린다) - 오버해드는 좀더 심하다.. 압추하니깐..  
On Load (강 올린다) - 사운드가 크면.. 압축하자..

PCM, ADPCM - 돌다 효과음...  
PCM - 썬좋은 효과음..  
ADPCM - 짜잘한 효과음..(버튼 소리 등)  
Vorbis - 중간효과음, 긴 소리들.. 음성..비지엠..

샘플 레이트 세팅 옵션 샘플 레이트.  
리버브 존. 주변 환경에 맞게 사운드를 바꿔준다.  
priority 플레이 우선순위.. 낮을수록 채널공간이 없으면 낮아진다..



**15.07.29**

## NGUI

크리에이트 - 하나를 한다.

하게되면 루트가 생성되는데.. 카메라가 밑에도 붙음.  
기본적으로 레이어 8번 쓰고 있다.

**widget** -> 유아이, 스프라이트, 버튼. 이런 요소들을 위젯  
위젯을 가지고 있는게 패널(위젯 여러개 넣을수 있다)

패널을 만들고 위젯을 넣는걸 권장한다.

NGUI는 특히 이미지를 만들때, 먼저 아틀러스를 만들어야 한다.

**ngui - open - atlas maker** 에서 만들 수있다.  
웬만하면 셰이더는 건들지 말자.  
아틀러스를 만들면 알아서 **uv** 좌표까지 다 생긴다.

**ngui - open** - 위젯 위자드 열어놓고 패널에 붙여놓을 녀석들 위자드 툴에서 **add to**  
하면 이미지가 그 패널에 생긴다.

셋페스콜이 드로우 명령 한번이라고 생각하면 된다.

아틀러스에 새롭게 추가할꺼면 똑같이 메이커에서 선택해논다음. 그다음 그냥 이미지를  
넣고 애드하면 된다.

트랜스폼에서 시커먼하게 되어있는곳은 웬만하면 건들지말자.

**slice** 사용하려면 아틀러스에서 보더를 셋팅해주면 된다.

순서를 바꿔주고 싶으면 위젯에서 뎁스를 수정하면 된다.

아틀러스끼리 뎁스를 중간에 얹끼도록 새로운 아틀러스 만들때마다 뎁스를 넉넉하게 준다.  
뎁스 순서가가 달라지면 드로우콜이 더 증가될수 있다.  
(모바일은 웬만하면 **2048.. 4096**은 최대다..)

## 15.07.30

화면 해상도 볼때 - 만약 게임에서 유징 레졸루션 나오면 사실 그게 실제 해상도 이다..  
유징 레졸루션이 아니와야 정확한 해상도.

좌표계 개념(앵커 피봇)

- 피봇이 기본으로 센터에 잡혀있다.. 앵커도 센터..(유니티 앵커는 더 나이스)
- 피봇은 위젯에서 쓰면 됨..
- 사이즈 늘릴때 위젯에서 사이즈를 늘린다.
- 앵커는 자기 부모의 패널을 받아와서 집어넣어서 앵커를 적용(컴포넌트에서 UI 앵커 스크립트 첨가 하면 그냥 잘 가능)
- **NGUI**는 스크린(2d) 개념이 아니다.
- 단 이렇게 작업을 하면 디바이스 해상도가 더 커지면 남기때문에, 유아이 루트에서 해상도를 고정해야한다. 플렉스어블을 컨스트럭트 하고 해상도 작업.

작은 유아이 오브젝트는 락을 해서 움직이는것이 때문에 비용이 든다.  
패널은 움직이는건 비용이 없다.

정적인 패널들은 어드밴스 옵션에서 스택을 걸자

- 단 패널이 움직이는건 비용이 않드니깐 상관은 없다.

쉐이더 틸트 입히는거.. 버텍스 컬러를 가져온다. 단 대신 정점을 해야하기 때문에 락을 해야한다.

버텍스 컬러를 가져다 쓰는다..

라벨이 중간이 끼면 드로우콜이 올라간다.(어제와 마찬가지로 )

렌더링 규칙.

백그라운드 (1000)

지오매트릭 (2000) - 불투명 물체들..

알파 테스트 - (유니티) 컷아웃 (2430)

알파 블랜드 - (유니티) 트랜스퍼런트 (3000)

오버레이 (4000)

패널에서 **show draw calls**. -> **renderQ** 덤스에 따라

**zTest** 않하면 무조건 투디마냥 앞에다 쪽쪽 그려짐..

일반버튼은 백그라운드 잡을때 라인패치 되는거 (슬라이스 하는거)

이미지 버튼은 각 버튼 이미지별로 다 할수있다.

사이즈를 수정한다면 클릭해야하는 오브젝트 **ngui**에서 **attach** 컬리더 다시 붙여야 한다.  
텍스처의 글자를 넣어서 최적화를 한다.

모달이 : 이창이 열리면 다른 창들은 활성화 되지 못하게  
모달리스 : 애가 활성화 되도 다른창은 된다

갯컴포넌트 트윈.  
인터렉션의 플레이 트윈 추가시켜주면 추가된다.

플레이 트윈  
**if target is disabeld** (만약 비활성화 되어있다면) **enable then Play** 활성화하고 플레이 시켜라.

플레이 **tween** 다시 붙여서 역으로 리버스 하고 **when** 셋팅에서 리버스 하면 자동으로 다시 꺼짐..

연출은 타임스케일은 멈추게 해야한다.  
드래그 **Keep Visible** 체크하고 패널지역을 보라색 패널 지역 넣으면 그 지역 앓넘어감.

감성 아이템 드레그.  
리스트 패널 (드레그 될 지역)  
그 아래에 그리드 오브젝트 넣고 - (유아이 그리드(활성화 비활성화 계속 해야한다))  
그 아래에 아이템 안에 드레그 스크롤 뷰 안에다가 붙여야한다.

**2015 07 31**

NGUI 스크립트 속성.. 앞에 UI를 붙여놔다..UISprite, UILabel 등등..

유아이 버그.. 지우면 정렬이 안되는거.. Destory 는 즉시 지워지지 않으니깐 유의해야한다.

유아이 붙이고 리스트에서 쇼 컨택트..에서 if 설정으로 스크롤 작업할 수 있다.

온프레스는 누를때 true, 땔때 false, 온클릭은 누른다음 땔때 됨..

트림 - 투명영역을 이미지에 맞춰서 잘라 주는건데.. 상황에 따라서 쓴다.

위젯 이미지 사이즈 snap은.. 원래 이미지 크기로.. 수정해준다.

함수로 는 makePixelPerfect()

폰트를 만들어 내면.. 라벨을 하고 unity 말고 NGUL..