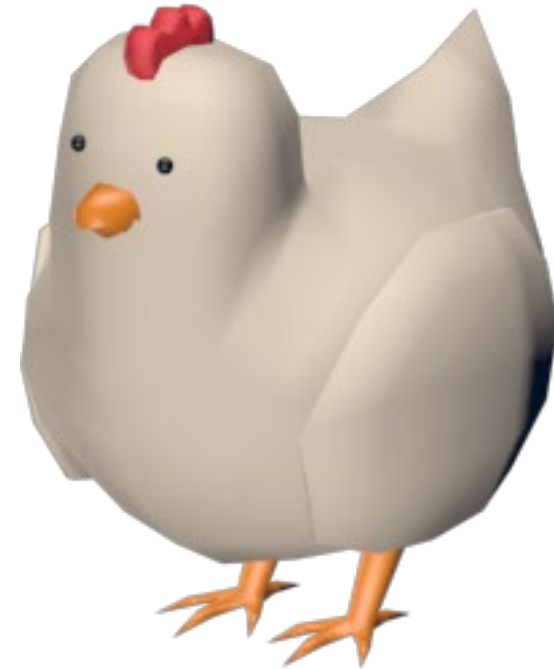


컴퓨터 게임 -  
**Chicken  
war**

20120316 강나영  
20131131 이유진



**Coins** 48



Building  
Recovery(#300)

Attack  
Upgrade(~~#~~300)

# CHICKEN WAR \_ 컨셉

평화로운 마을에 침입한 검은 꼬꼬가 침입했다!

마을 아무데서나 나타나는 검은 꼬꼬로부터 마을을 지켜내는 게임입니다.

마을의 건물이 검은 꼬꼬의 총에 맞아 모두 부서지면 게임이 끝납니다.

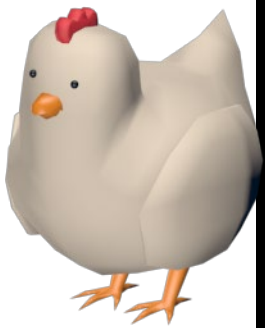
게임이 시작되면 코인항아리를 클릭해 코인을 얻고 하얀 꼬꼬를 설치 해야 해요!

하얀 꼬꼬는 죽지 않고 마을을 지키며

코인 300으로는 하얀 꼬꼬의 공격력을 업그레이드 할 수 있습니다.

검은 꼬꼬는 30초마다 공격력이 높아집니다.

마을을 최대한 오래 지켜주세요



# CHICKEN WAR \_ 직접구현

## ■ 검은 꼬꼬는 마을에 있는 Wanderpoint 중 한 지점(랜덤 적용)에서 생성

- WanderPoint를 빈 게임 오브젝트로 생성, Tag를 지정해줌으로써, 코드에서 Wanderpoint들을 배열로 정리, 인덱스 값을 랜덤으로 접근 가능하게 구현한 다음, 랜덤한 WanderPoint position에서 검은 닭이 Instantiate되도록 함

## ■ 검은 꼬꼬 NPC화

- 적정시간(2초 간격)이 지난 후 마을의 WanderPoint위치에서 자동으로 생성되고, 자동으로 주변 빌딩을 발견하도록 Update함수에서 주변반경으로 Ray를 쏘아서 감지, 공격하도록 구현

## ■ 검은 꼬꼬 FSM (Patrol & Attack & Dead )구현

- Patrol : 빈 오브젝트인 LeaderPatrol을 따라다니도록 함. (Leader가 각Wanderpoint를 돌아다니도록 구현)
- Attack : 주변에 일정거리 안에 빌딩이나 동지가 감지되면, shootBullet하게끔 설정
- Dead : 검은 닭의 전체 생명력에서, 하얀 닭으로부터 공격받으면, 생명 차감. 생명이 0이하이면 Destroy(gameobject);하도록 구현

## ■ 길로만 돌아다니도록 Nav Mesh 적용

- Plane으로 길 모양을 잡아준 뒤 빈 게임 오브젝트 아래로(Child로) Plane 전부 넣어줌, 부모 빈 게임 오브젝트의 상태를 Navigation Static으로 설정(children까지 적용)한 뒤, 닭 Prefab에는 NavMeshAgent속성을 할당해주고, Back 해줌.

# CHICKEN WAR \_ 직접구현

## ■ 하얀 꼬꼬 버튼 클릭하면 해당 버튼 자리에 Instantiate

- Canvas의 버튼을 마을 곳곳에 배치, 코인이 30이상이면, 버튼 클릭 가능하도록 함, 버튼 클릭하면, 그 자리에 하얀 닭 생성, 하얀 닭은 자동으로 주변의 검은 닭 인지해서, 공격함

## ■ 빌딩&둥지 생명 구현

- 검은 닭의 총알을 맞으면, collider의 trigger속성으로, 총알이 빌딩 혹은 둥지와 충돌함을 검사해서 생명 차감함

## ■ 빌딩, 둥지 각각 생명력 캔버스 이미지에 표시되도록 구현

- 각 빌딩과 둥지에, 캔버스 구현, 이미지 생성. 코드에서 public GameObject로 생성해준 변수에, Inspector에서 생성한 이미지를 전달해줌. 생명이 차감될 때마다, 이미지에 값을 전달해주기 위해, 이미지의 transform.localScale속성의 x축 값 변경해줌.

## ■ 코인구현

- GameController라는 최상위 빈 게임 오브젝트를 만들어 총체적으로 코인 관리를 하도록 하였음. 코인을 다른 스크립트에서 간접 접근할 수 있도록 public형의 get set 함수를 따로 만들어주었음.

# CHICKEN WAR \_ 어려움 해결

■ 루트 GameController의 money 값을 이용해 Upgrade 및 건물 체력 회복에 사용하기 위해서 다른 클래스에 접근하는 방법을 알아야 했음

- GameObject형 변수에 스크립트를 포함한 object를 GameObject.Find로 찾아온 뒤 클래스의 인스턴스에 저장. GetComponent로 클래스를 가져옴.

■ 게임을 시작하면 Bullet의 Damage가 전 게임에서 올라간 값으로 남아있음

- 전역변수 num을 선언, start()에서 0으로 초기화 후 num이 0이라면 bullet의 damage를 초기값으로 다시 설정하도록 한 뒤 num++을 해줌.
- \*Start()는 멤버변수 초기화를 담당

■ Patrol Leader 및 닭의 움직임

- 닭이 Patrol Leader를 따라다니다 보니 가로 방향의 도로에서 세로 방향으로 움직이는 모습을 보였음. NavmeshAgent의 FindClosestEdge 메소드를 사용하여 벡터 값을 조정하여 움직임을 조금 더 자연스럽게 구현했음.

■ AssetStore에서 다운받아 설치했던 Road에 mesh설정이 다중으로 되어 있어서, NavMesh계산이 어려워짐(길 전체를 걸어다닐 수 있어야 하는데, 길의 특정부분 mesh에만 NavMesh가 적용됨)

- Road의 Mesh새롭게 적용, NavMesh에 Radius값 변경과, 이어 붙여서 만든 길 전체를 하나의 Mesh로 묶어봄. 하지만 여전히 길의 특정 부분(얇아서 움직임이 부자연스러운 길)으로만 닭이 돌아다녀서 새로운 Plane을 생성, Road로 만들어주고 NavMesh로 Bake함

# CHICKEN WAR \_ 어려움 해결

- Bullet(공격Object)에서 빌딩이나, 동지(피해Object)로 SendMessage 보내면, 각각의 빌딩과 동지의 이름이 다 같아서 NullReference오류 발생
  - 각각의 Prefab과 인스턴스 이름을 바꿔주기에는 너무 바꾸어 주어야 할 것이 많았기에, Damage함수처럼 빌딩과 동지에서 동시에 사용하는 함수는 하나의 인터페이스를 상속받아 다르게 구현해줌으로 해결. 다른 함수들은 GetComponent함수로 부딪힌 오브젝트들을 불러와서 SendMessage를 대체
- PlayerPrefs 으로 Scene사이의 정보 교환
  - Scene을 바꾸면 변수가 다시 초기화 됨을 몰랐기 때문에 유니티 Scripting API 페이지에서 PlayerPrefs라는 메소드를 찾았음. Start Scene에서 pot을 10초 동안 클릭해서 모은 동전을 Main Scene으로 넘겨주어 Scene 사이에 연속성을 줌.
- 검은 닭이 공격할 대상을 찾아주어야 하는데, 검은 닭은 계속 움직이기 때문에 공격 해야 할 오브젝트가 바뀜.
  - 따라서 GetClosestEnemy()를 이용해 가장 가까운 빌딩 오브젝트를 찾아서 Attack State에 적용 함. (Turret 회전 및 공격)
- Bullet프리팍에 충돌 함수 구현, 하얀 닭과 검은 닭에 동일 Bullet을 적용, 하얀 닭이 쏜 Bullet을 빌딩이 맞아도 빌딩 생명력이 차감됨 (아군의 총알에 체력이 닳는 현상)
  - 하얀 닭과 검은닭의 각각의 Bullet Prefab을 따로 생성하고 검은 닭의 Bullet 코드 중 충돌함수 OnTriggerEnter에는 빌딩과, 동지와 부딪히는 검사를, 하얀 닭의 Bullet 충돌함수에는, 검은 닭과 부딪히는지를 검사하도록 구현.

# CHICKEN WAR \_ 어려움 해결

- 버튼을 누르고 위치를 누르면 하얀 닭을 사용자가 마우스 클릭한 위치에 생성되도록 하고 싶었으나 사용자의 마우스 포지션을 onClick 리스너에서 처음에 클릭한 위치밖에 가져오지 못함
  - 버튼을 미리 배치한 뒤 버튼을 누른 위치에 닭이 생성되도록 구현함.
- 빌딩의 Collider를 메시에 딱 맞게 설정하였음에도 OnCollisionEnter 함수에서 NullReference Exception이 발생하여 FBX 파일의 기즈모 문제라고 생각, 모델을 다시 3D맥스로 가져가 기즈모를 빌딩 위치에 옮겨 대부분의 문제를 해결하였으나 빌딩들 중 apt라는 모델이 여전히 OnCollisionEnter 함수에서 오류 발생하였음
  - 충돌 함수를 OnTriggerEnter로 변경해주고, 각 빌딩의 Collider에 isTrigger 설정 체크해줌
- 빌딩이 공격 당할 때 HP bar의 길이가 늘어나는 문제
  - Canvas의 크기가 HP bar의 크기와 맞지 않았기 때문에 일어난 문제로 Canvas의 크기를 같게 해주어서 문제 해결



# CHICKEN WAR \_ 가져온 것

## ■ UI

- 마을 빌딩들 max 파일 다운
- 닭 max 파일 다운
- 마을 등지 & 알 -> 따로 다운 받아 합쳐서 fbx로 생성

## ■ Script

- IDamagable : 하나의 인터페이스(IDamageable)로 구현, 각각의 빌딩과 등지 코드에서 상속받아서 다른 함수로 구현해줌 (Google검색 )
- GetClosestEnemy : Transform[]을 인자로 받아서 가장 가까이 있는 Object를 찾아줌  
(<http://stackoverflow.com> search)
- Timer 구현 시 youtube 튜토리얼 참고하였음
  - <https://www.youtube.com/watch?v=x-C95TuQtf0>