



[스파르타코딩클럽] 게임개발 종합반 - 3주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

[수업 목표]

1. 그럴싸한 게임을 완성해보기
2. hp바 만들기
3. 레벨 시스템을 구상해보기

[목차]

- [01. 3주차 오늘 배울 것](#)
- [02. 기본 씬 구성하기](#)
- [03. 시작 씬 만들기](#)
- [04. 밥 쓰기](#)
- [05. 고양이 나타내기\(1\) - 만들기](#)
- [06. 고양이 나타내기\(2\) - 내려오기/충돌](#)
- [07. 고양이 나타내기\(3\) - 등장하기 & 종료](#)
- [08. 레벨 구성하기\(1\) - 레벨업 표기하기](#)
- [09. 레벨 구성하기\(2\) - 레벨 반영하기](#)
- [10. 마무리 - 게임 즐겨보기 & 버그잡기](#)
- [11. 숙제 - 해적 고양이 만들기!](#)

[HW. 3주차 숙제 해설](#)



모든 토클을 열고 닫는 단축키

Windows : `Ctrl + alt + t`

Mac : `⌘ + ⌘ + t`

01. 3주차 오늘 배울 것

▼ 1) 3주차 수업의 목표와 범위



오늘의 키워드! [이제 조금 익숙해졌는걸](#)

- 스파르타가 만들어둔 이미지를 가지고 게임을 만들어봅니다.
→ 이미지들을 진짜 [디자이너](#)에게 받았다고 생각하고 만들어보세요!
- 3주차 내용도 1, 2주차의 것과 80% 같고, 20%만 새롭답니다.
→ 한번 더 복습해볼까요? 이번주까지 하고나면, 꽤 익숙해질거예요!

▼ 2) 오늘 만들 것 : [고양이 밥주기 게임](#)

- 배고픈 고양이들에게 밥을 주고, 생선가게를 지켜보아요!
- 새롭게 배우는 것: 여러 Scene 만들기!, 슬라이더 바, 레벨시스템

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/35b1291f-26e4-4791-8c9b-c8e108163bcd/고양이_밥주기_게임.mp4

▼ 3) 만들 순서

1. 기본 씬 구성하기 (UI, 강아지, 고양이)
2. 강아지 움직임 더하기 + 밥 쏘기
3. 고양이 내려오게 하기
4. 고양이 밥 먹기 + 옆으로 가게 하기
5. 새로운 고양이 나오게 하기
6. 레벨업하기

02. 기본 씬 구성하기

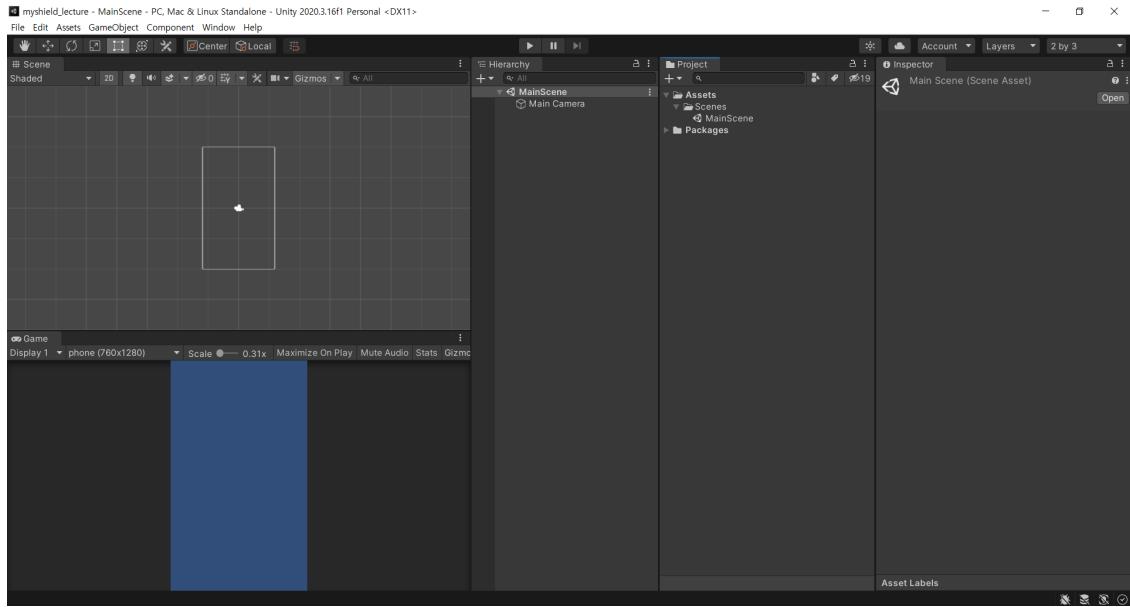


dogvscat 이라는 프로젝트를 만들고 시작할게요!

▼ 1) 기본 세팅하기 & 배경 만들기

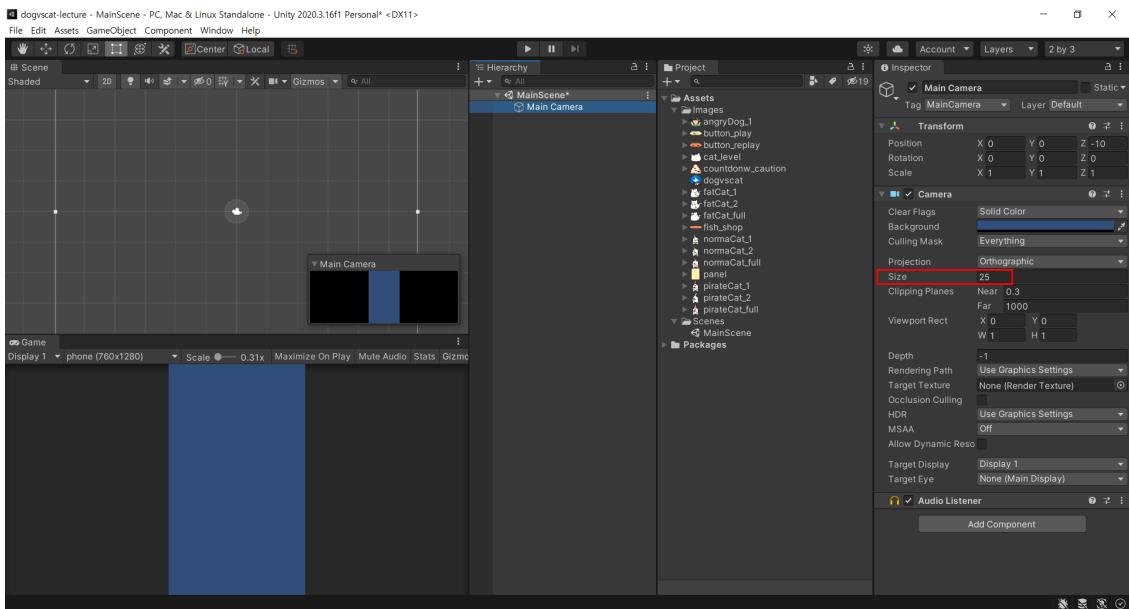
1. windows → 2x3 layout 클릭! free aspect → phone 클릭!

MainScene으로 변경하는 것도 잊지마세요!



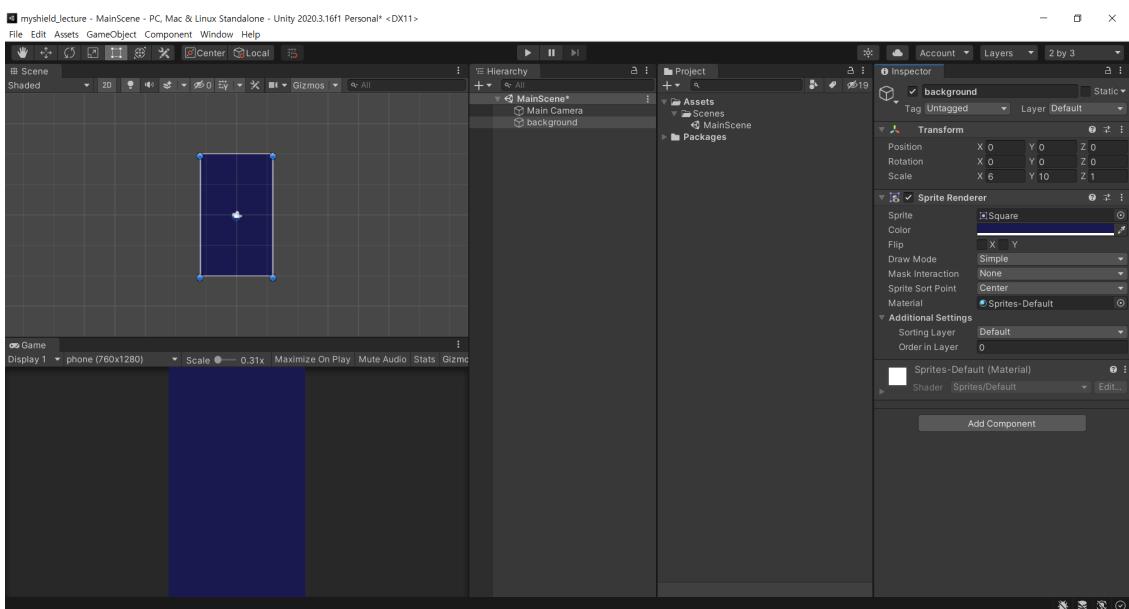
2. 메인카메라의 size를 5 → 25로 바꿀게요!

→ 약간 더 멀리서 보겠다는 이야기! 이미지들이 이 사이즈에 맞춰 작업되어 있습니다. 😎



3. 카메라 색은 `hex → FFF0B2` 으로 할게요! 그러면, 배경과 같은 효과!

→ 이렇게도 할 수 있다는 사실! 알고갈게요!



▼ 2) 이미지 받아두기

파일 전체를 Assets 폴더 밑에 저장해주세요!

▼ [코드스니펫] dogvscat - 이미지 저장

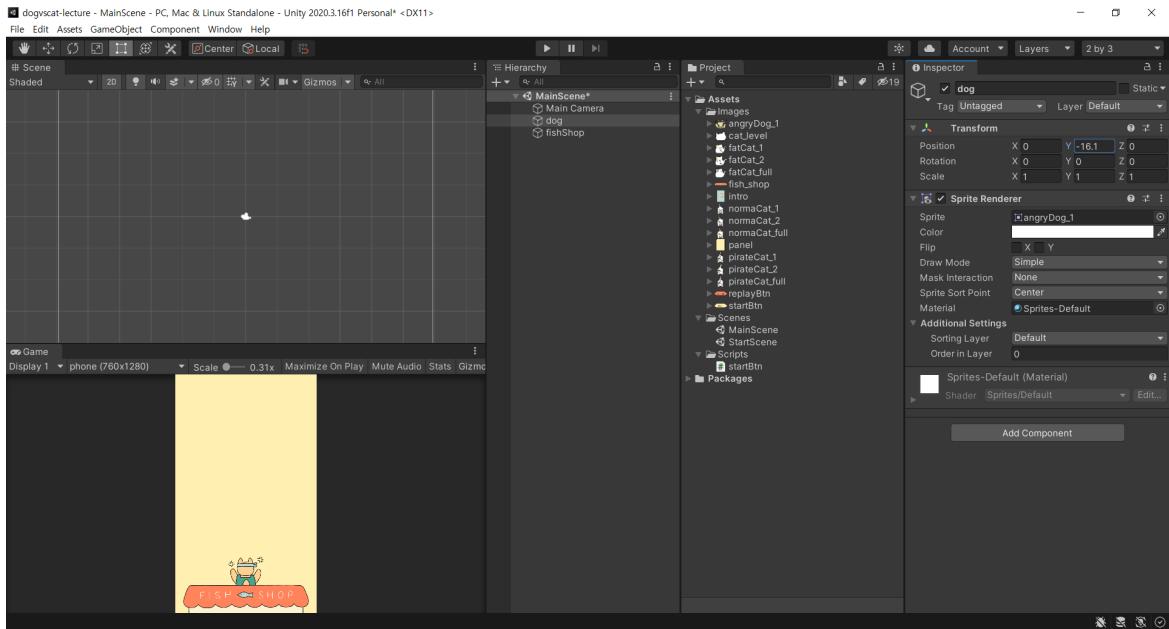
https://s3.ap-northeast-2.amazonaws.com/materials.spartacodingclub.kr/game_new/week03/dogvscat.zip

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c43da0d1-5524-4399-8c39-d249d7e1f1f7/dogvscat.zip>

▼ 3) 오브젝트 배치하기

- 생선가게 : `y: -22`
- 강아지 : `y: -16.1`

- 위치를 지정하신 후, Inspector의 Sprite에 이미지를 드래그해주세요!



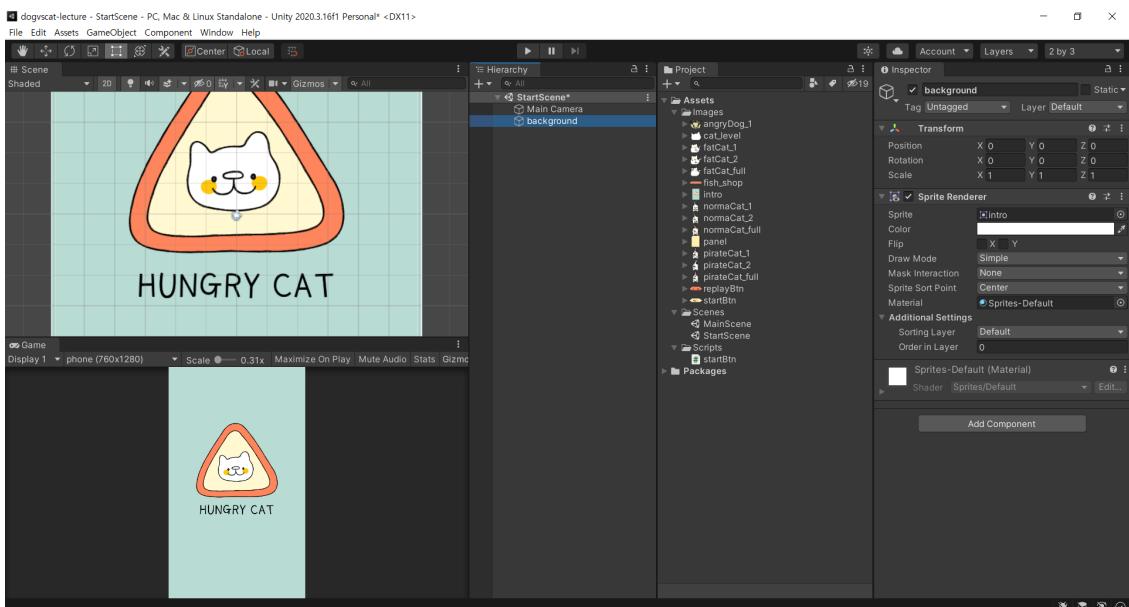
03. 시작 씬 만들기

▼ 1) 시작씬 구성하기



시작 씬은 처음이죠? 가볍게 한번 만들고 갑시다!

- Scenes 폴더 안에 create → Scene 이름은 `StartScene` 그리고 더블클릭!
- 마찬가지로 Camera `size` → `25`
- sprite 이미지 만들어서 ⇒ `intro` 이미지 끌어다넣기

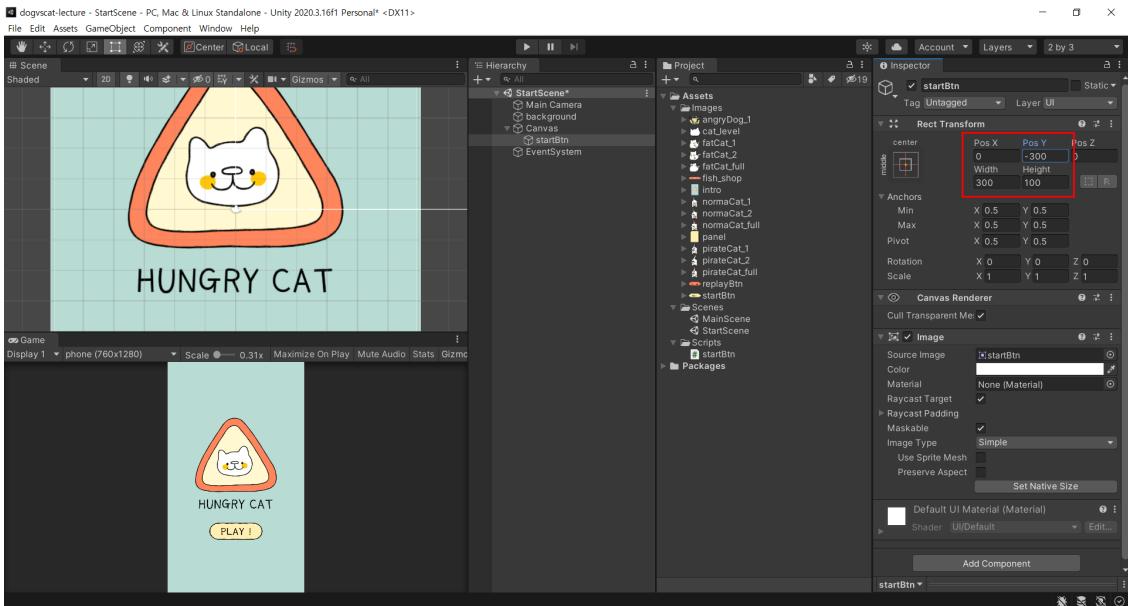


- `UI → Image`를 만들어서 `startBtn` 해두기

→ startBtn 이미지 끌어다넣기

→ posY: -300

→ width: 300, height: 100



▼ 2) 씬 넘어가기

1. startBtn에 버튼 컴포넌트 달기

→ 참고) 버튼 컴포넌트는 "sprite"가 아니라, 반드시 "UI Image"에 붙여야 작동한다는 사실!

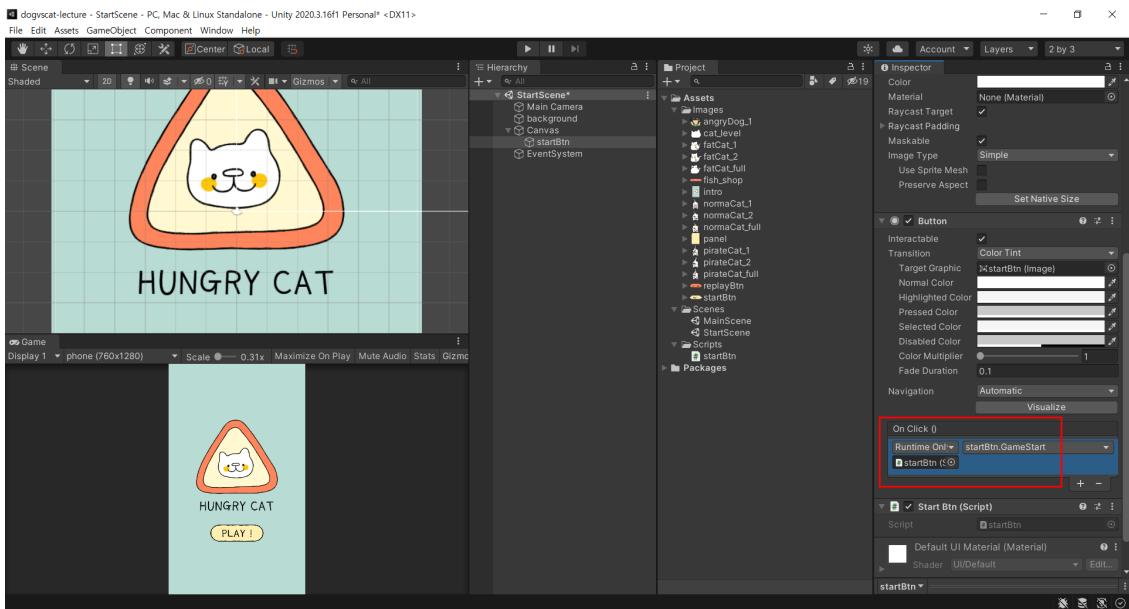
2. 스크립트 만들기

→ Scripts 폴더 만들고, `startBtn.cs` 만들기 → startBtn에 붙여두기

```
using UnityEngine.SceneManagement;

public void GameStart()
{
    SceneManager.LoadScene("MainScene");
}
```

3. onclick에 붙이기



▼ 3) **Play** 해서 확인해보기!

→ 잘 되네요! 이제 MainScene으로 이동해서 작업할게요! (더블클릭!)

04. 밥 쏘이



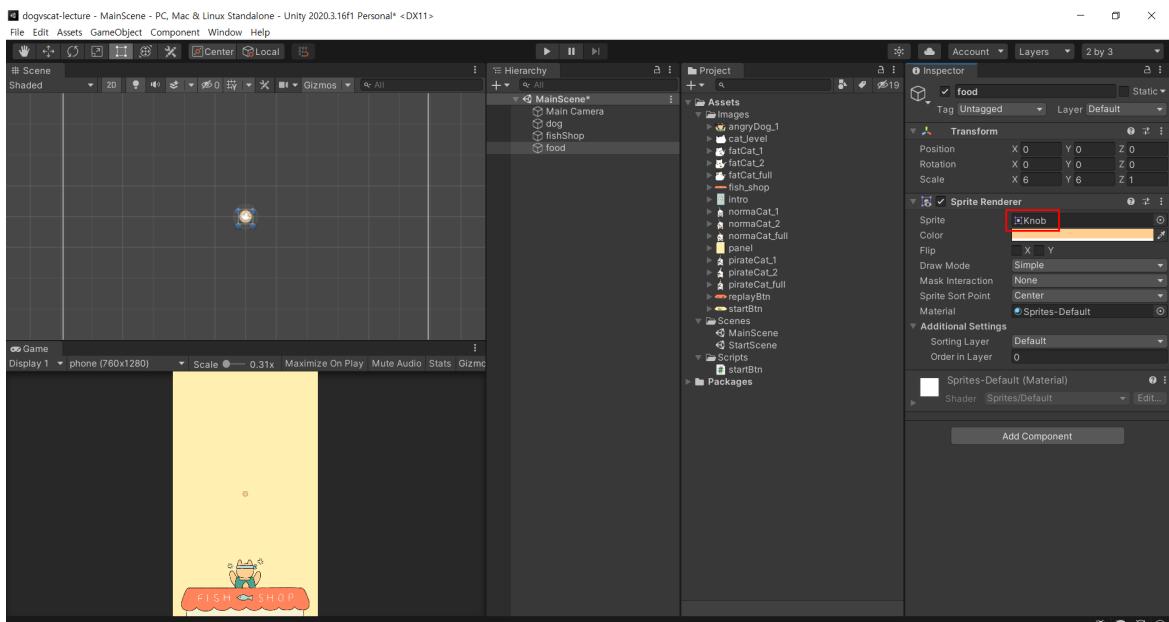
무언가를 복사해서 → 움직이게 하는 일종의 총복습! 찬찬히 따라해보아요

▼ 1) 밥 만들어두기

→ sprite → circle 클릭, 이름 : `food`

→ scale: `x: 6, y: 6`

→ Sprite : Knob 으로 설정하고, 색은 **컬러추출기** 를 이용해서 강아지 색으로 설정하기



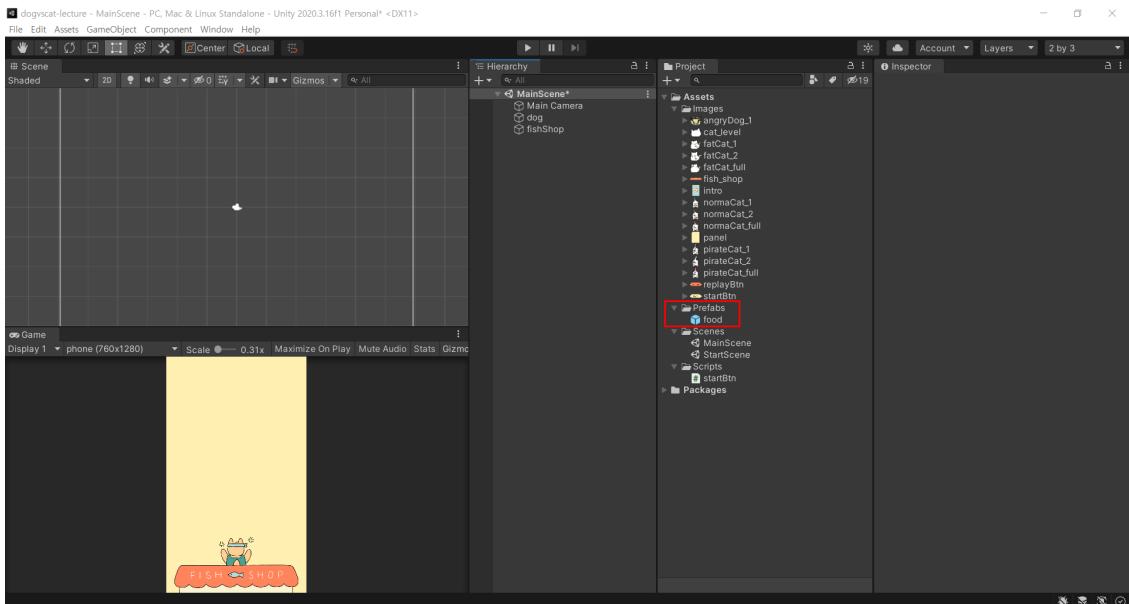
▼ 2) 밥 복사해서 쏘이

1. `food.cs` 만들어서 붙여두기. 생성하면 무조건 위로 직진!

```
void Update()
{
    transform.position += new Vector3(0.0f, 0.5f, 0.0f);
}
```

2. 프리팹으로 만들어두기

→ Prefabs 폴더 만들고 끌어다놓기! 과감하기 원래 있던 food는 삭제!



3. GameManager 만들기

→ 0.2초 마다 밥을 쏘기

→ 강아지 위치에서 y 좌표만 2.0f 높아진 곳에서 쏘기

→ `Quaternion.identity` 는 회전 없다는 뜻! (no rotation)

```
public GameObject food;
public GameObject dog;

void Start()
{
    InvokeRepeating("makeFood", 0.0f, 0.2f);
}

void makeFood()
{
    float x = dog.transform.position.x;
    float y = dog.transform.position.y + 2.0f;
    Instantiate(food, new Vector3(x,y,0), Quaternion.identity);
}
```

4. 앗, 그런데 food가 안 없어진다!

→ `food.cs` 에서 y 좌표가 26.0f 가 넘으면 없어지게 하기

```
void Update()
{
    transform.position += new Vector3(0.0f, 0.5f, 0.0f);
    if (transform.position.y > 26.0f)
    {
        Destroy(gameObject);
    }
}
```

▼ 3) 강아지 움직이기

1. 강아지 마우스 따라 움직이기

→ dog.cs 만들어주세요!

→ 마우스 좌표 중 X 좌표만 가져오기! (2주차에서 썼던 그 코드!)

→ 외워서 쓰는 게 아니라, 검색해보고 쓴다!라고 생각하세요. (튜터도 마찬가지..!)

→ y 좌표는 내 좌표를 그대로 쓰기! `transform.position.y`

```
void Update()
{
    Vector3 mousePos = Camera.main.ScreenToWorldPoint(Input.mousePosition);
    transform.position = new Vector3(mousePos.x, transform.position.y, 0);
}
```

2. fishShop을 벗어나지 않게 하기

→ x 좌표 -9.5f ~ 9.5f로 고정하기

```
void Update()
{
    Vector3 mousePos = Camera.main.ScreenToWorldPoint(Input.mousePosition);
    float x = mousePos.x;
    if (x > 8.5f)
    {
        x = 8.5f;
    }
    if (x < -8.5f)
    {
        x = -8.5f;
    }
    transform.position = new Vector3(x, transform.position.y, 0);
}
```

05. 고양이 나타내기(1) - 만들기

▼ 1) normalCat 만들기

1. create empty → `normalCat` 만들기

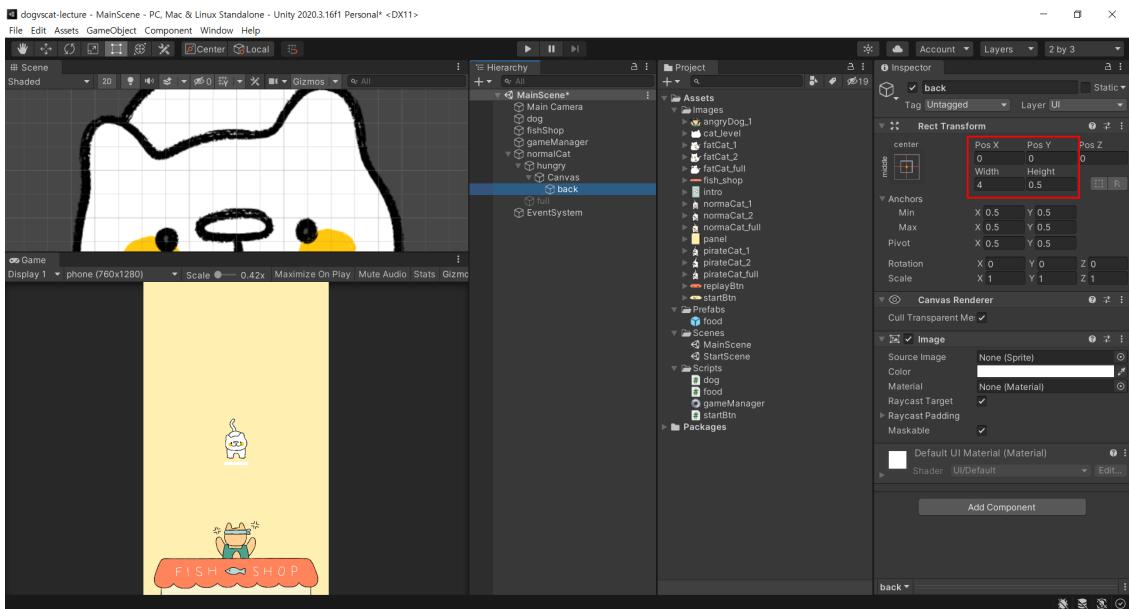
2. normalCat 안에 full / hungry로 sprite를 만들어 붙이기

3. hungry 안에 canvas → image를 만들고 back을 만들기!

→ Canvas를 만들 때, RenderMode를 World Space로 변경해주기!

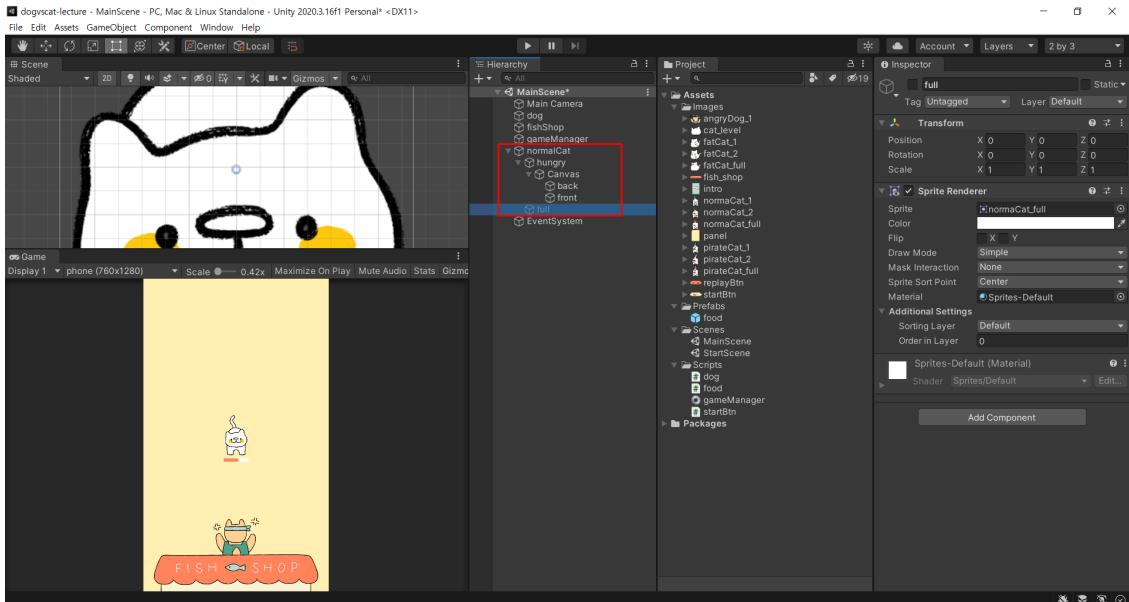
→ `width: 4, height: 0.5`

→ canvas의 position을 `y: -4`로 고정하기



4. front는 back을 (ctrl+d)해서 복사한 뒤 작업시작!

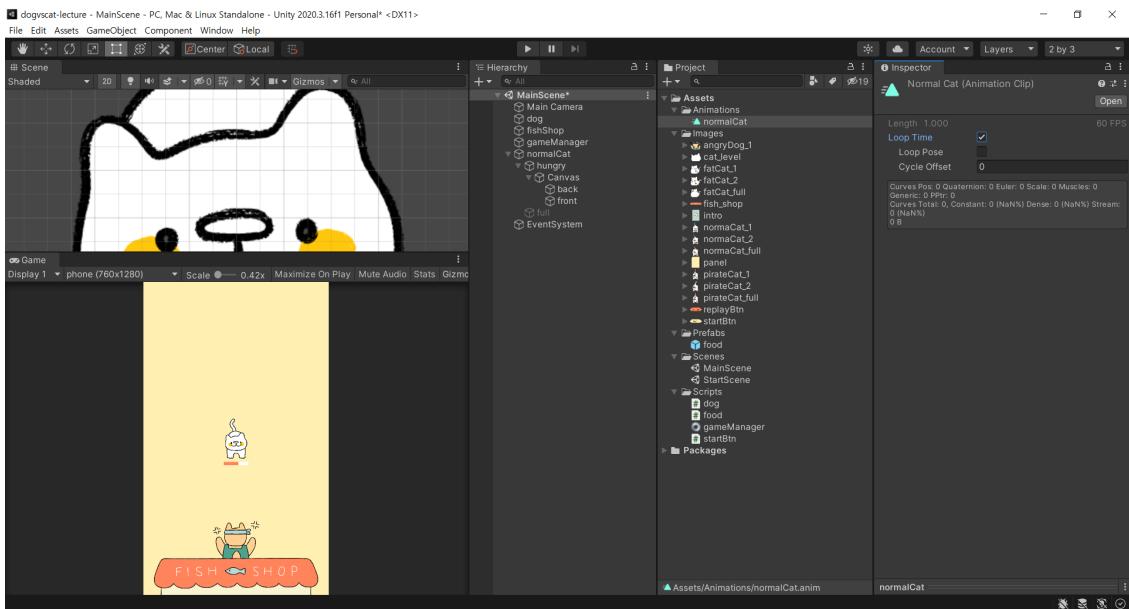
- 색 : 추출기로 fish-shop 색 추출
- pivot의 x 값을 0 으로 만들고, scale의 x 값을 0.4로 조정해보기!
- 이제 x scale만 조정해주면 게이지를 만들 수 있음. 이것이 바로 hp 바 만들기!



▼ 2) normalCat에 애니메이션 붙이기

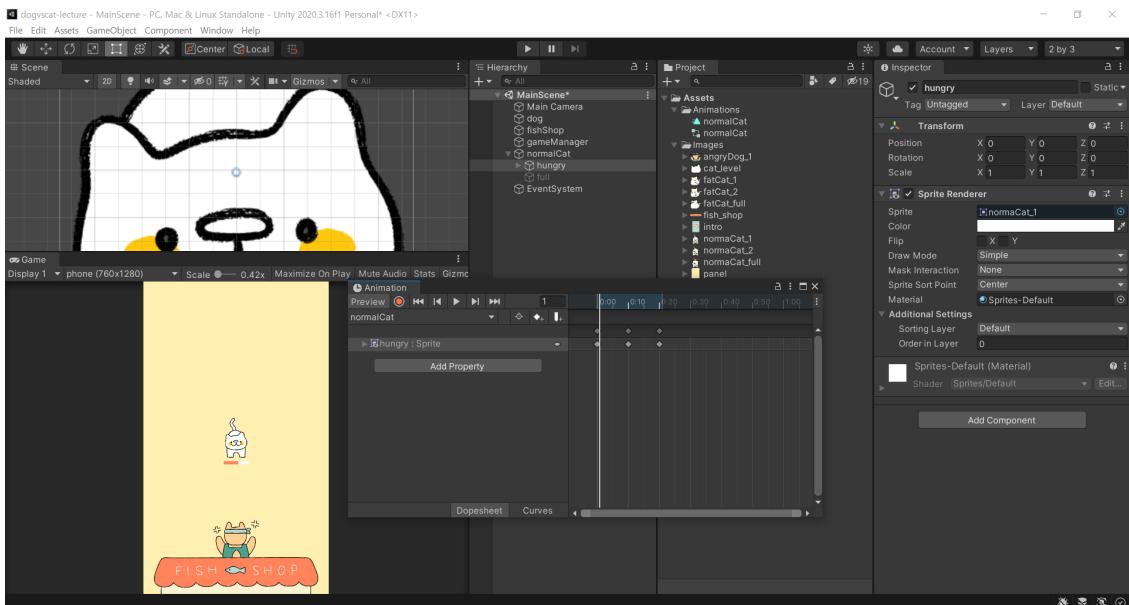
1. Animations 폴더 만들고 `normalCat` 만들기

- Loop time 체크 잊지 말기!



2. normalCat에 붙여넣고, 더블클릭해서 녹화시작하기

→ 0:10에 `normalCat_2`, 0:20에 다시 `normalCat_1`



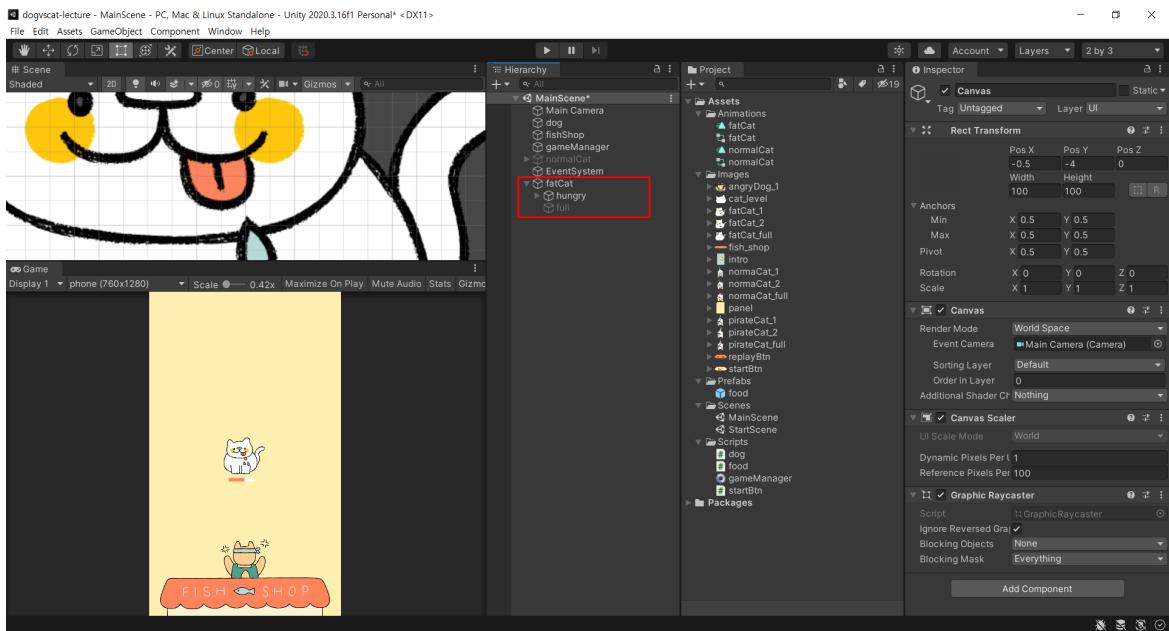
▼ 3) fatCat 만들기

→ `ctrl+d` 를 이용해서 만들거예요! 두 번째는 처음부터 만들 필요가 없겠죠 😎

→ (1) hungry, full 의 이미지를 교체해주기

→ (2) Canvas의 x 좌표를 `-0.5` 해두기

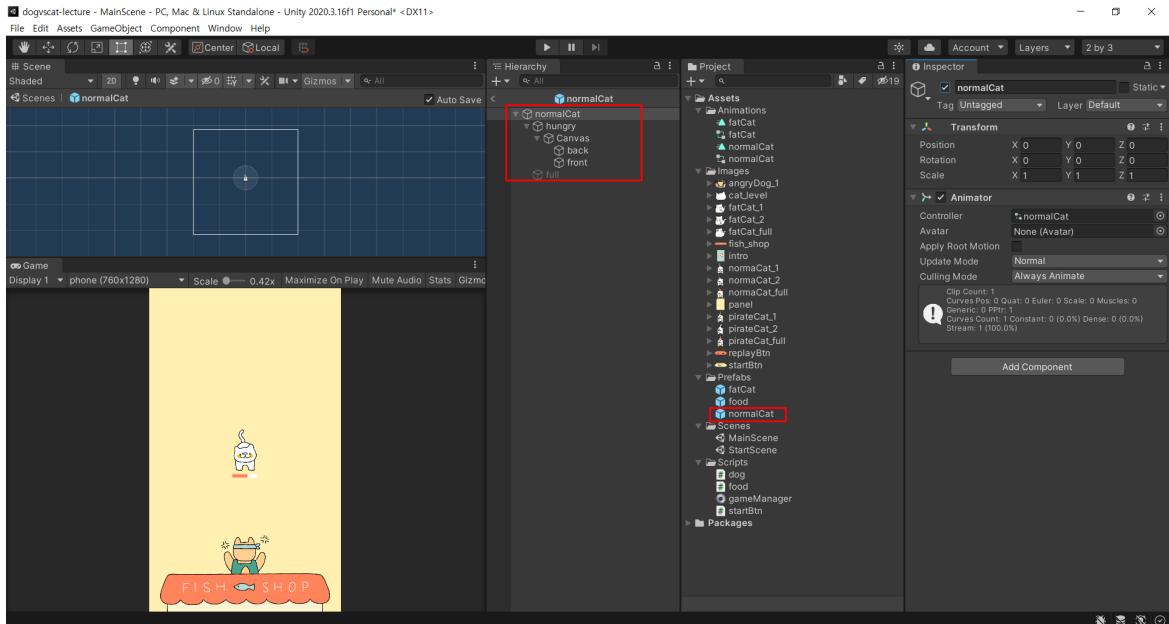
→ (3) animation 을 새로 만들어서 넣어두기 (기존에 있던 animator를 먼저 삭제하기)



▼ 4) 두 개를 모두 프리팹화 해두기

→ fatCat은 game view에서는 삭제해두기

→ 참고) 프리팹을 고칠 때는 프리팹을 더블클릭해서 작업합니다!



06. 고양이 나타내기(2) - 내려오기/충돌

▼ 1) 고양이 내려오게 하기

1. `cat.cs` 만들기

2. 임의의 위치에서 내려오게 하기

→ y 좌표는 30.0f 화면 밖에서 내려오게 하기

```

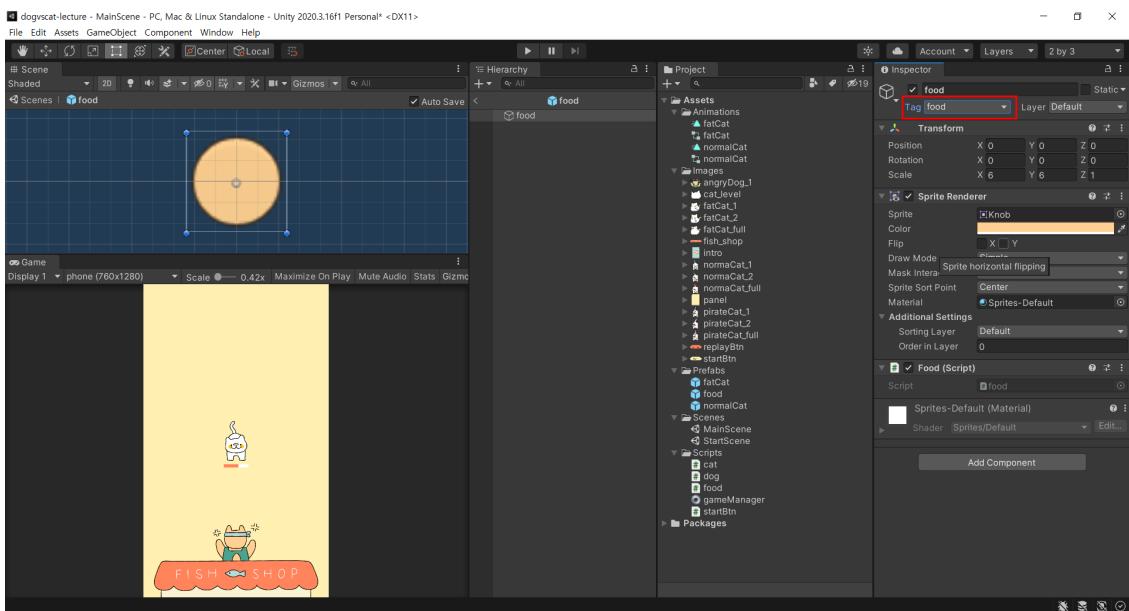
void Start()
{
    float x = Random.Range(-8.5f, 8.5f);
    float y = 30.0f;
    transform.position = new Vector3(x, y, 0);
}

// Update is called once per frame
void Update()
{
    transform.position += new Vector3(0.0f, -0.05f, 0.0f);
}

```

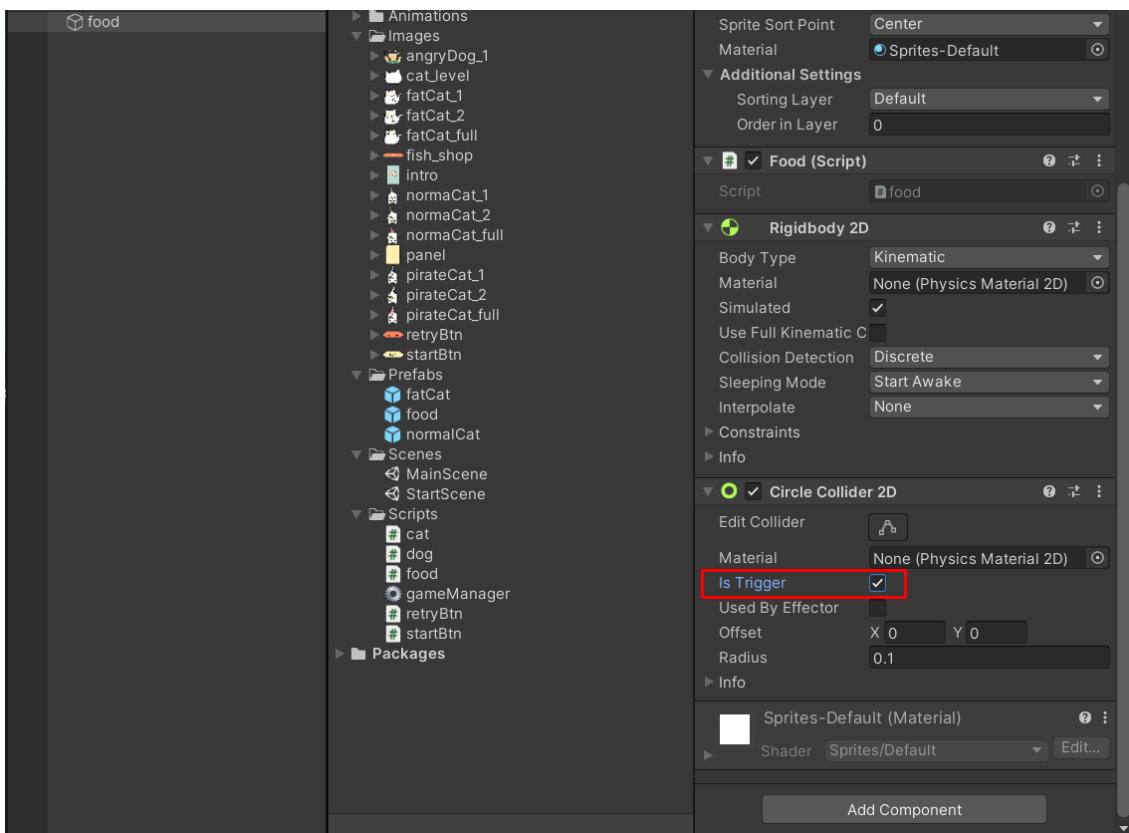
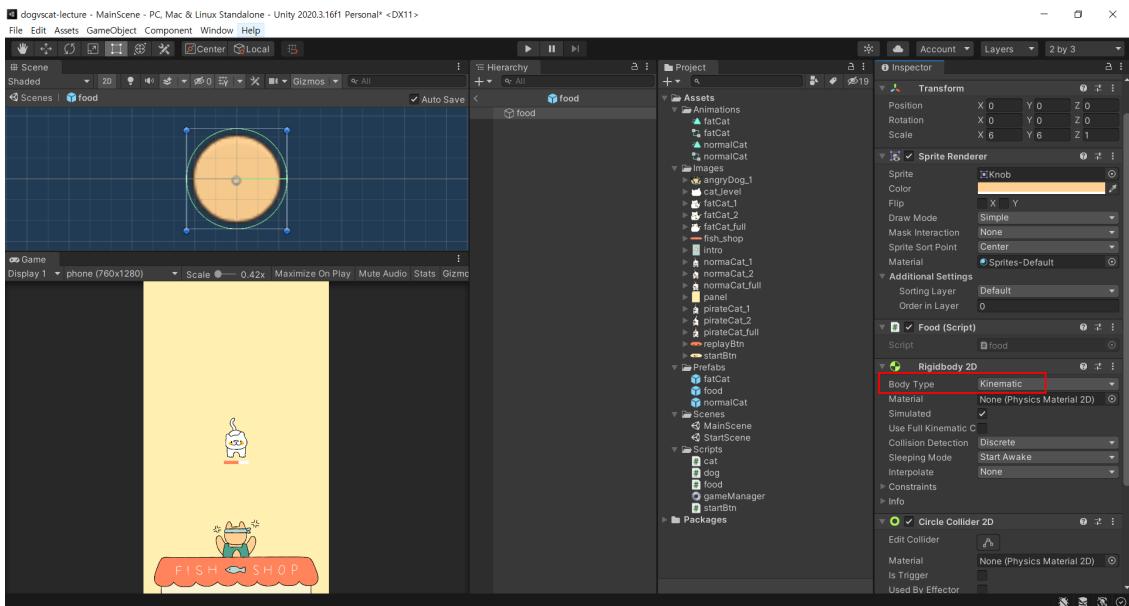
▼ 2) 고양이와 밥 충돌하게 하기

1. 밥에 tag 주기



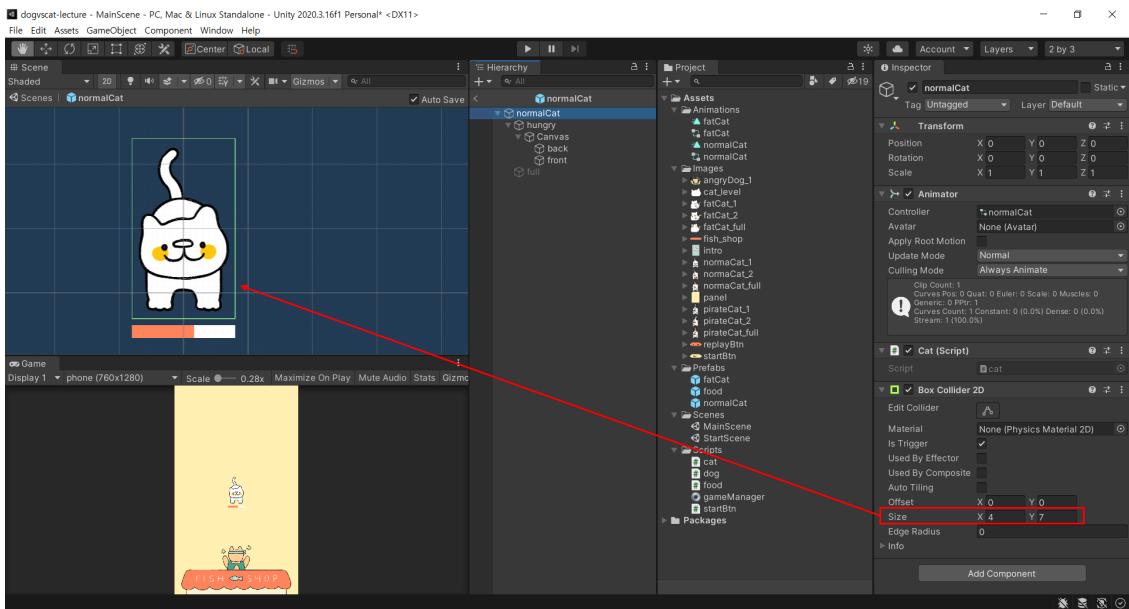
2. 밥에 `rigidbody 2D`, `circle collider 2D` 달아주기

- 참고) 충돌 = 한쪽에 `rigidbody` + 양쪽에 `collider`
- 단! `Body Type`을 `Kinematic` 으로 잡아주기 = 중력의 영향을 안 받겠다는 뜻!
- 그리고 `isTrigger` 에 체크! 중력의 영향을 안 받을 때에는 이것을 체크해주세요!
- (옵션을 체크하지 않으면 충돌감지가 안됩니다!)



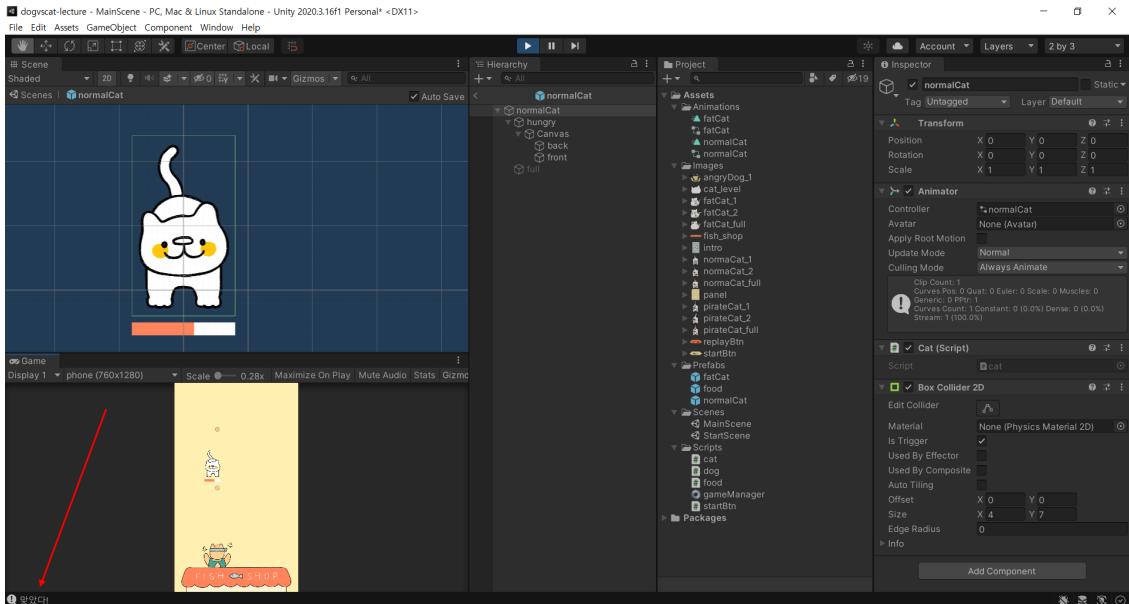
3. 고양이에 box collider 2D 달아주기 (나중에 fatCat에도 달아야겠죠?)

→ collider size를 조정해줘야 해요! `x:4, y:7`로 맞춰볼까요?



4. `cat.cs` 에 충돌 구현하기

```
void OnTriggerEnter2D(Collider2D coll)
{
    if (coll.gameObject.tag == "food")
    {
        Debug.Log("맛있다!");
    }
}
```



▼ 3) 충돌하면 에너지를 채워주기

1. 기본 에너지 세팅

```
float full = 5.0f;
float energy = 0.0f;
```

2. 밥 충돌하면 에너지 올라가게 하기 + 밥은 없애기

```

void OnTriggerEnter2D(Collider2D coll)
{
    if (coll.gameObject.tag == "food")
    {
        energy += 1.0f;
        Destroy(coll.gameObject);
    }
}

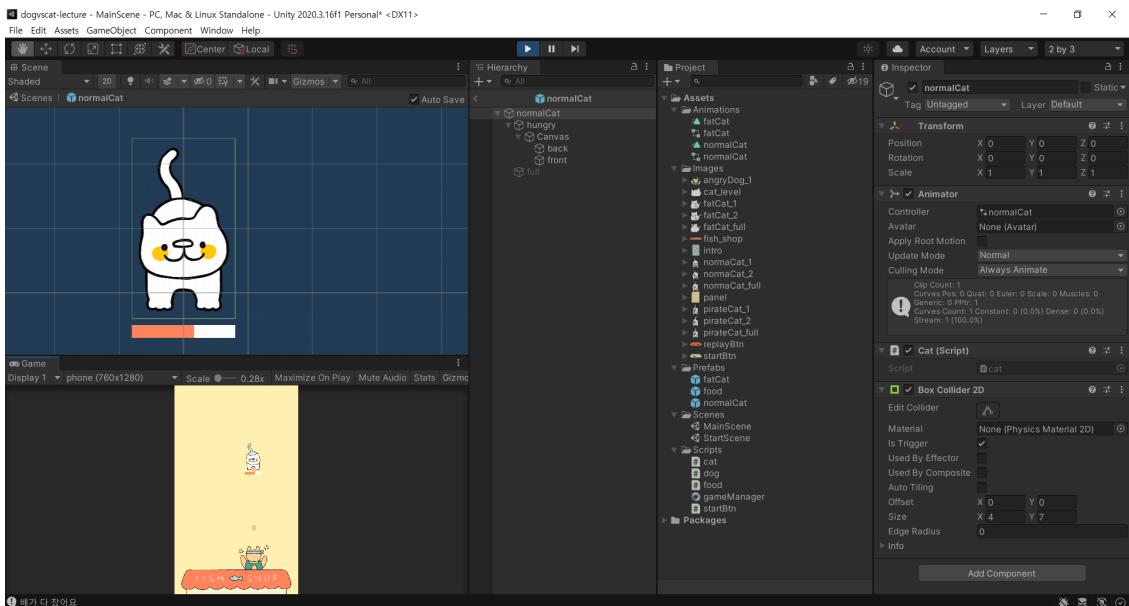
```

3. energy 와 full이 같아지면 배불렸다!

```

void OnTriggerEnter2D(Collider2D coll)
{
    if (coll.gameObject.tag == "food")
    {
        if (energy < full)
        {
            energy += 1.0f;
            Destroy(coll.gameObject);
        }
        else
        {
            Debug.Log("배가 다 찼어요");
        }
    }
}

```



4. 게이지 채우기

→ 시작 게이지를 0으로 해둘게요! (front의 x scale을 0으로)

→ hungry 밑에, Canvas 밑에, front를 찾아서, x scale을 energy/full 값으로 조절

```

void OnTriggerEnter2D(Collider2D coll)
{
    if (coll.gameObject.tag == "food")
    {
        if (energy < full)
        {
            energy += 1.0f;
            Destroy(coll.gameObject);
            gameObject.transform.Find("hungry/Canvas/front").transform.localScale = new Vector3(energy / full, 1.0f, 1.0f);
        }
        else
        {
            Debug.Log("배가 다 찼어요");
        }
    }
}

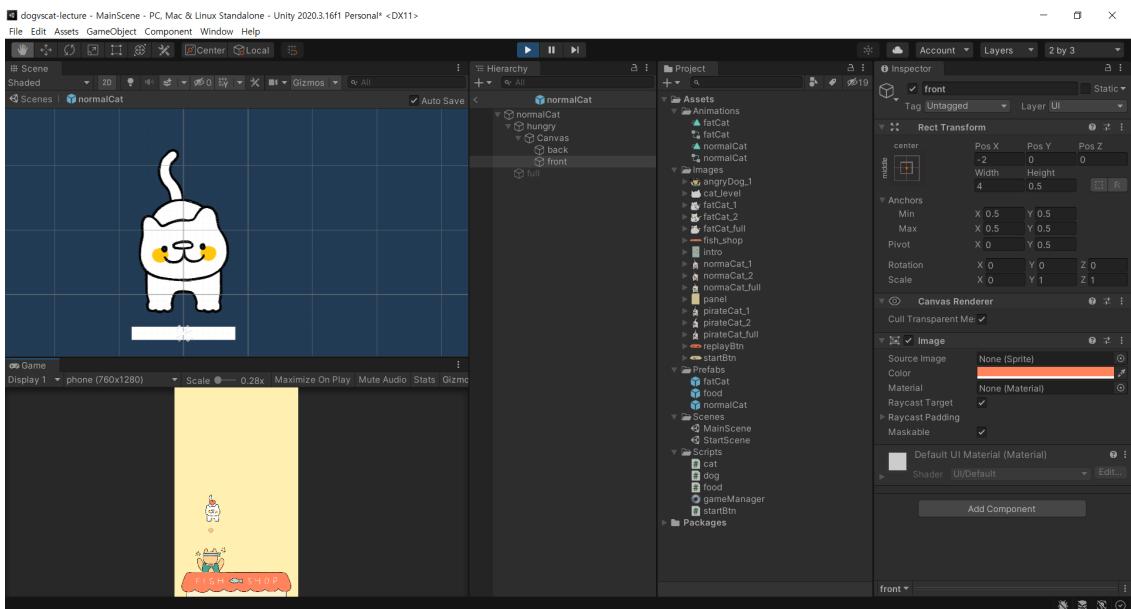
```

5. 게이지가 모두 찼으면, hungry는 안보이고, full이 보이게 하기!

```

void OnTriggerEnter2D(Collider2D coll)
{
    if (coll.gameObject.tag == "food")
    {
        if (energy < full)
        {
            energy += 1.0f;
            Destroy(coll.gameObject);
            gameObject.transform.Find("hungry/Canvas/front").transform.localScale = new Vector3(energy / full, 1.0f, 1.0f);
        }
        else
        {
            gameObject.transform.Find("hungry").gameObject.SetActive(false);
            gameObject.transform.Find("full").gameObject.SetActive(true);
        }
    }
}

```



6. 에너지 꽉 차면 옆으로 비키게 하기

→ 단, 화면 오른쪽에 있었으면(x>0) 오른쪽으로, 왼쪽이면 왼쪽으로 비키게 하기

```

void Update()
{
    if (energy < full)
    {
        transform.position += new Vector3(0.0f, -0.05f, 0.0f);
    }
    else
    {
        if (transform.position.x > 0)
        {
            transform.position += new Vector3(0.05f, 0.0f, 0.0f);
        }
        else
        {
            transform.position += new Vector3(-0.05f, 0.0f, 0.0f);
        }
    }
}

```

7. 3초 뒤에 없애기

```

void Update()
{
    if (energy < full)
    {
        transform.position += new Vector3(0.0f, -0.05f, 0.0f);
    }
    else
    {
        if (transform.position.x > 0)
        {
            transform.position += new Vector3(0.05f, 0.0f, 0.0f);
        }
        else
        {
            transform.position += new Vector3(-0.05f, 0.0f, 0.0f);
        }
        Destroy(gameObject, 3.0f);
    }
}

```

07. 고양이 나타내기(3) - 등장하기 & 종료

▼ 1) gameManager에서 고양이 부르기

1. 등장시키기

→ 1.0초에 한 마리씩 등장. 이제 `normalCat` 을 Hierarchy에서 없애도 됩니다!

```

public GameObject normalCat;

void Start()
{
    InvokeRepeating("makeFood", 0.0f, 0.2f);
    InvokeRepeating("makeCat", 0.0f, 1.0f);
}

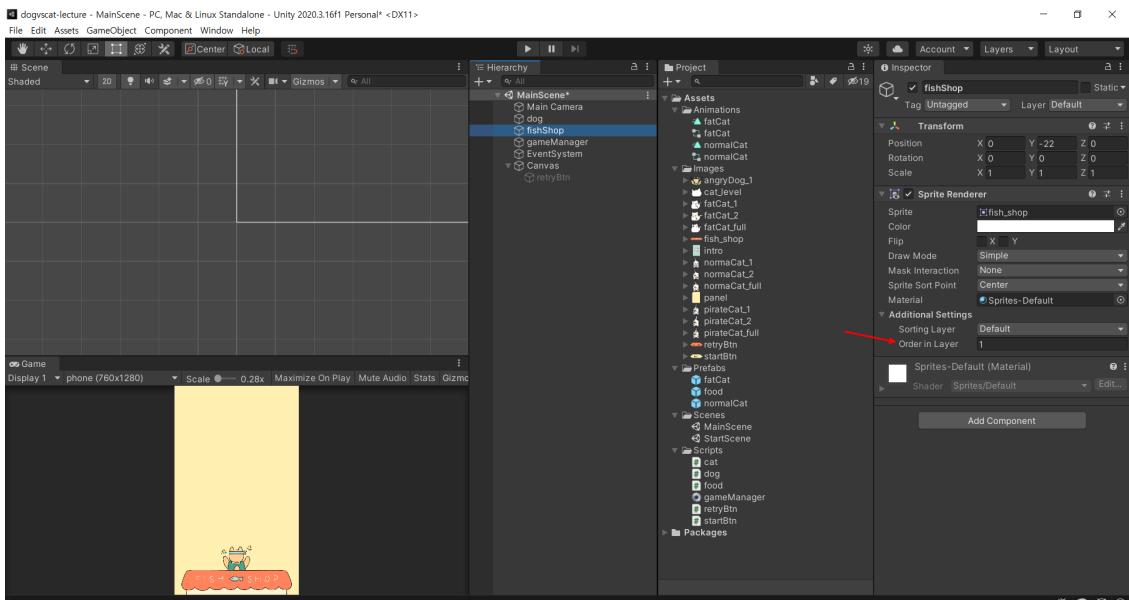
void makeCat()
{
    Instantiate(normalCat);
}

```

2. order-in-layer 조정하기

→ 실행시켜보면 고양이의 에너지바가 fish-shop보다 위에 온답니다.

→ dog와 fish-shop의 order-in-layer를 1로 조정하기!



▼ 2) 특정 y 좌표 밑으로 내려오면 게임 종료

1. retryBtn 만들고, 숨겨두기

→ Image로 만들기

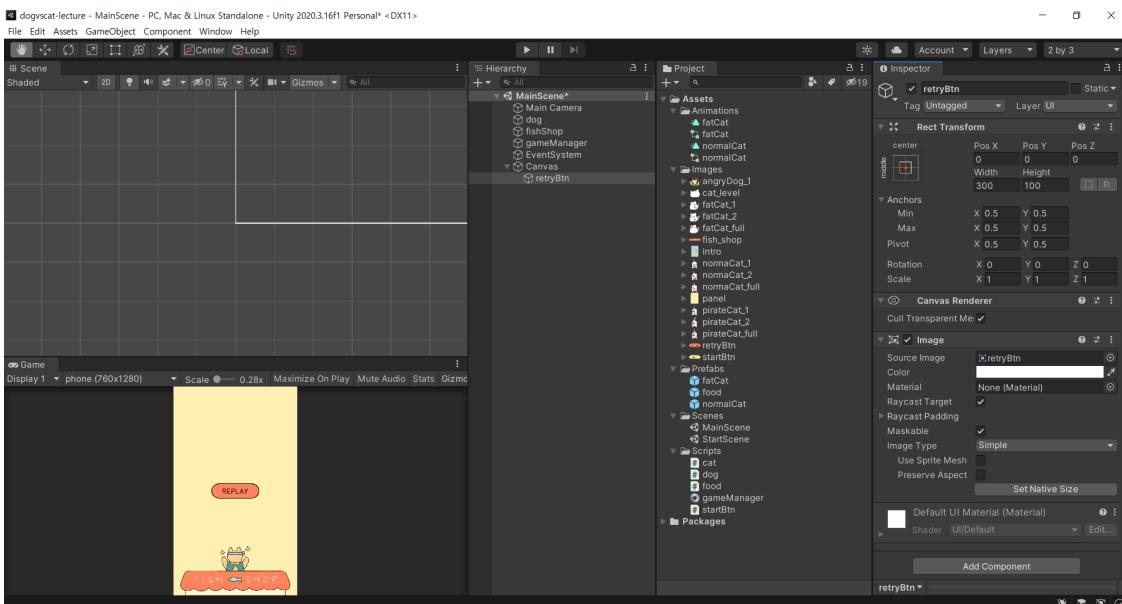
→ width: 300, height: 100

→ button 컴포넌트 달기

→ retryBtn.cs 만들어서 넣고, onclick에 연결하기

```
using UnityEngine.SceneManagement;

public void ReGame()
{
    SceneManager.LoadScene("MainScene");
}
```



2. gameManager 싱글톤화

→ 어딘가에서 나를 부르기 전에는 반드시!

```
public static gameManager I;

void Awake()
{
    I = this;
}
```

3. gameManager 내에 gameOver() 함수 만들기

```
public GameObject retryBtn;

public void gameOver()
{
    retryBtn.SetActive(true);
    Time.timeScale = 0.0f;
}
```

4. cat.cs에서 특정 y 좌표 밑으로 내려가면 gameOver() 부르기

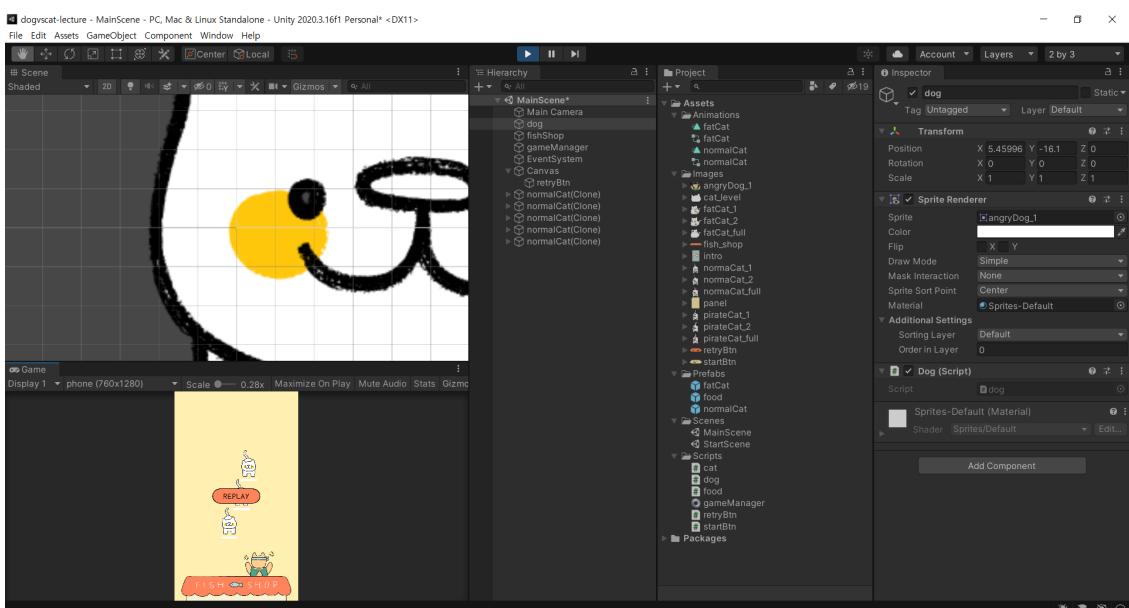
```
void Update()
{
    if (energy < full)
```

```

    {
        transform.position += new Vector3(0.0f, -0.05f, 0.0f);

        if (transform.position.y < -16.0f)
        {
            gameManager.I.gameOver();
        }
        else
        {
            if (transform.position.x > 0)
            {
                transform.position += new Vector3(0.05f, 0.0f, 0.0f);
            }
            else
            {
                transform.position += new Vector3(-0.05f, 0.0f, 0.0f);
            }
            Destroy(gameObject, 3.0f);
        }
    }
}

```



08. 레벨 구성하기(1) - 레벨업 표기하기

▼ 1) 레벨 UI 만들기

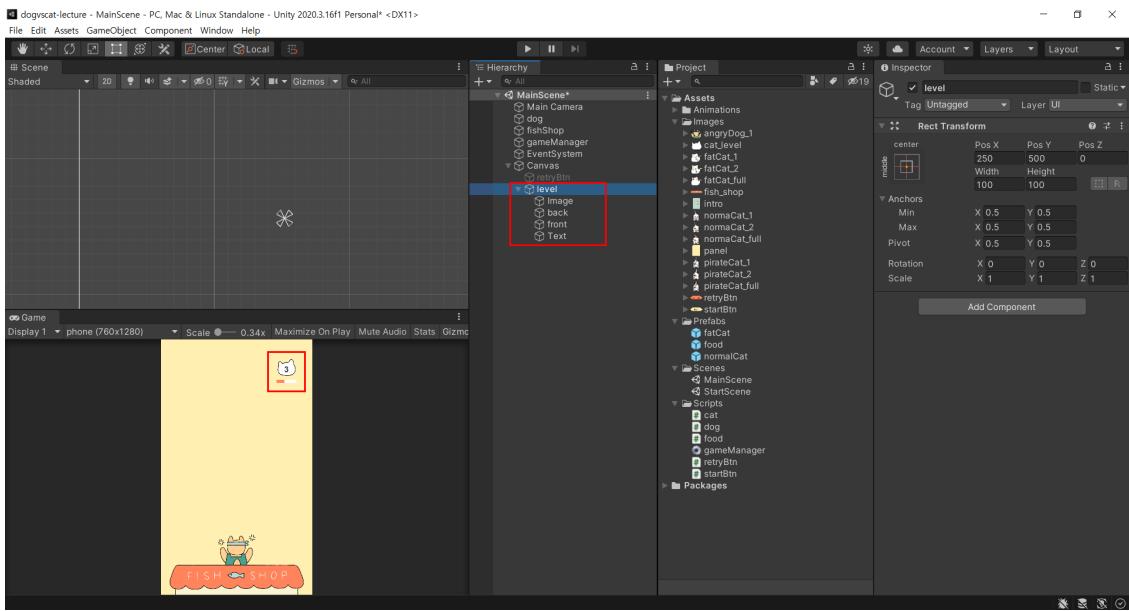
1. 오른쪽 위에 레벨 표기하기

→ `level` 게임오브젝트로 둑어 배치합니다. 위치는 `x:250, y:500`

(Rect Transform을 리셋해주세요!)

→ `back / front`는 `width: 100, height: 15, y: -70`

→ `Text`는 `size 40`



▼ 2) 5마리 당 레벨 1씩 올리기

1. gameManager에 레벨업 함수 만들기

```
int level = 0;
int cat = 0;

public void addCat()
{
    cat += 1;
    level = cat / 5;
}
```

2. 고양이가 배부를 때 `addCat()` 함수 부르기

→ 다만, 이렇게 하면 안됩니다!

→ 이유는, 이렇게 하면 옆으로 빠질 때 계속 점수가 올라가겠죠!

```
void OnTriggerEnter2D(Collider2D coll)
{
    if (coll.gameObject.tag == "food")
    {
        if (energy < full)
        {
            energy += 1.0f;
            Destroy(coll.gameObject);

            gameObject.transform.Find("hungry/Canvas/front").transform.localScale = new Vector3(energy / full, 1.0f, 1.0f);
        }
        else
        {
            gameManager.I.addCat();
            gameObject.transform.Find("hungry").gameObject.SetActive(false);
            gameObject.transform.Find("full").gameObject.SetActive(true);
        }
    }
}
```

3. 그래서! 약간 수정을 합니다.

→ `상태` 를 `isFull` 로 만들고, 제어합니다.

```
bool isFull = false;

if (isFull == false)
{
    gameManager.I.addCat();
    gameObject.transform.Find("hungry").gameObject.SetActive(false);
```

```

        gameObject.transform.Find("full").gameObject.SetActive(true);

        isFull = true;
    }
}

```

▼ 3) 레벨업 표기해주기

1. 레벨업 라벨 초기화해주기
 → 레벨업 텍스트 0으로
 → 레벨업 front 바 x scale 0으로
2. 레벨업 라벨 바꿔주기
 → 레벨업 텍스트, front 바

```

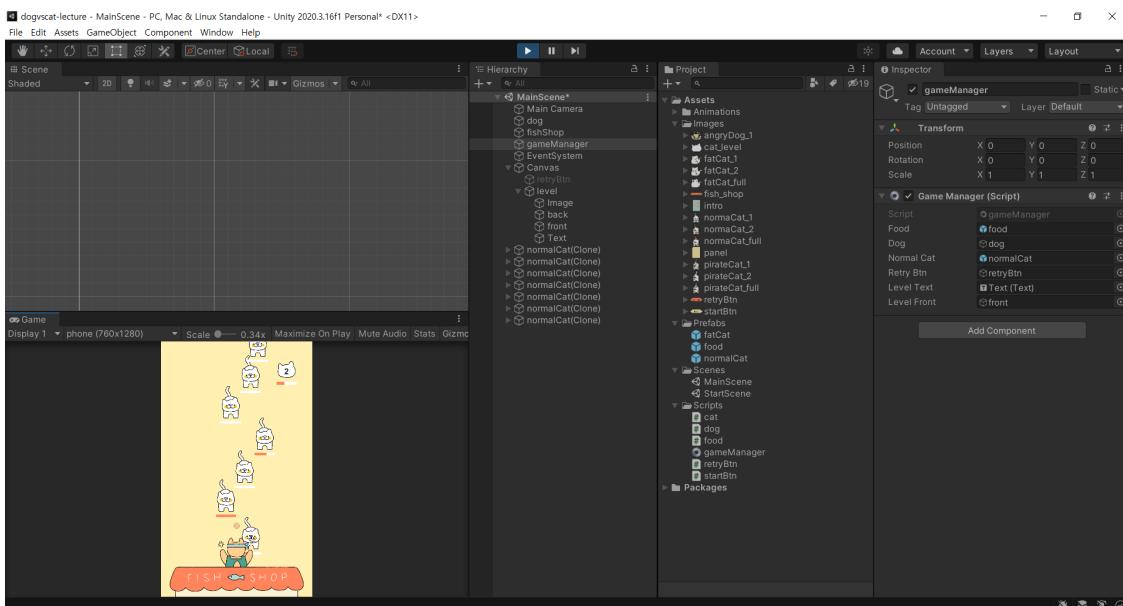
using

public Text levelText;
public GameObject levelFront;

public void addCat()
{
    cat += 1;
    level = cat / 5;

    levelText.text = level.ToString();
    levelFront.transform.localScale = new Vector3((cat - level * 5) / 5.0f, 1.0f, 1.0f);
}

```



09. 레벨 구성하기(2) - 레벨 반영하기



더 많은 레벨을 구현할 수 있겠지만, 우리는 Lv.3 까지만 디자인 해볼게요!

▼ 1) Lv.1, Lv.2 : 더 많은 고양이 출현시키기

1. `gameManager.cs`에서 확률에 따른 추가 고양이 출현
 → 단, 빠른 진행을 위해 `makeFood`를 `0.2f` → `0.1f`로 바꿔둘게요!
 → 이미 생각보다 쉽지 않을걸요!

```

void makeCat()
{
    Instantiate(normalCat);

    if (level == 1)
    {
        float p = Random.Range(0, 10);
        if (p < 2) Instantiate(normalCat);
    }
    else if (level >= 2)
    {
        float p = Random.Range(0, 10);
        if (p < 5) Instantiate(normalCat);
    }
}

```

▼ 2) Lv.3 : fatCat 출현시키기

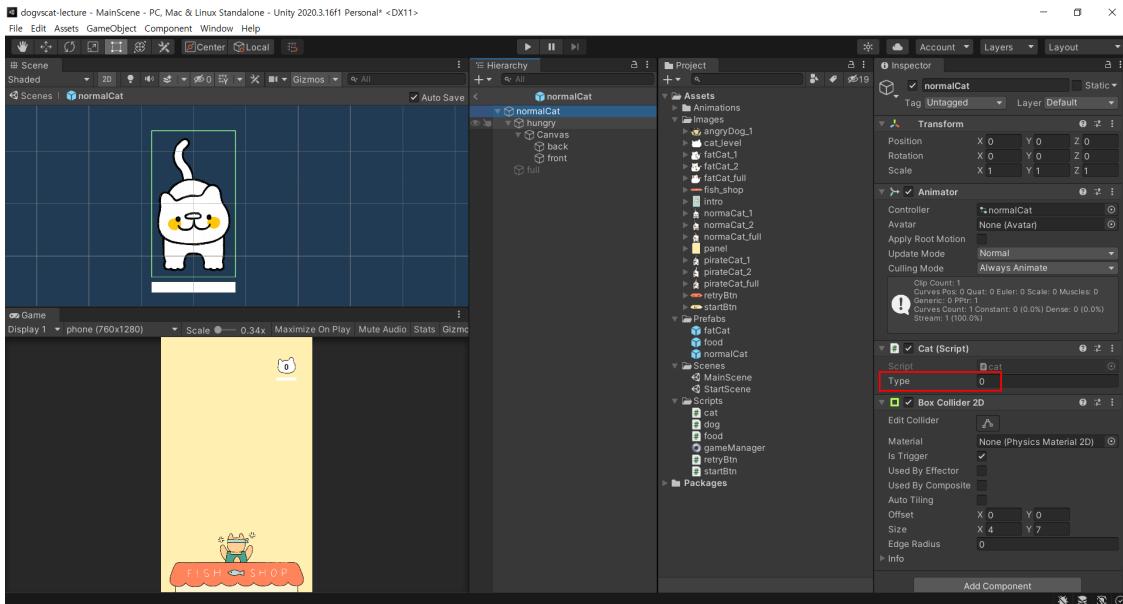
1. `cat.cs`에 고양이 타입을 만들기!

→ `type: 0`은 normalCat, `type: 1`은 fatCat !!

→ 아래와 같이 써두면, 오브젝트에서 타입을 조절할 수 있습니다.

→ 다시 normalCat 프리팹에 가서 `type: 0`으로 만들어보기

```
public int type;
```

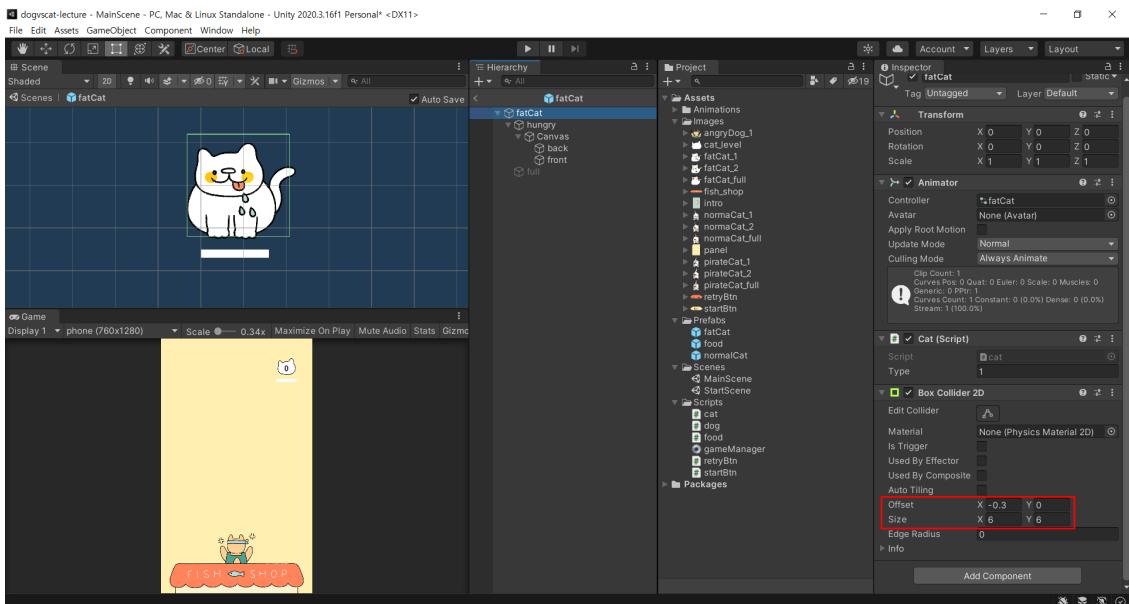
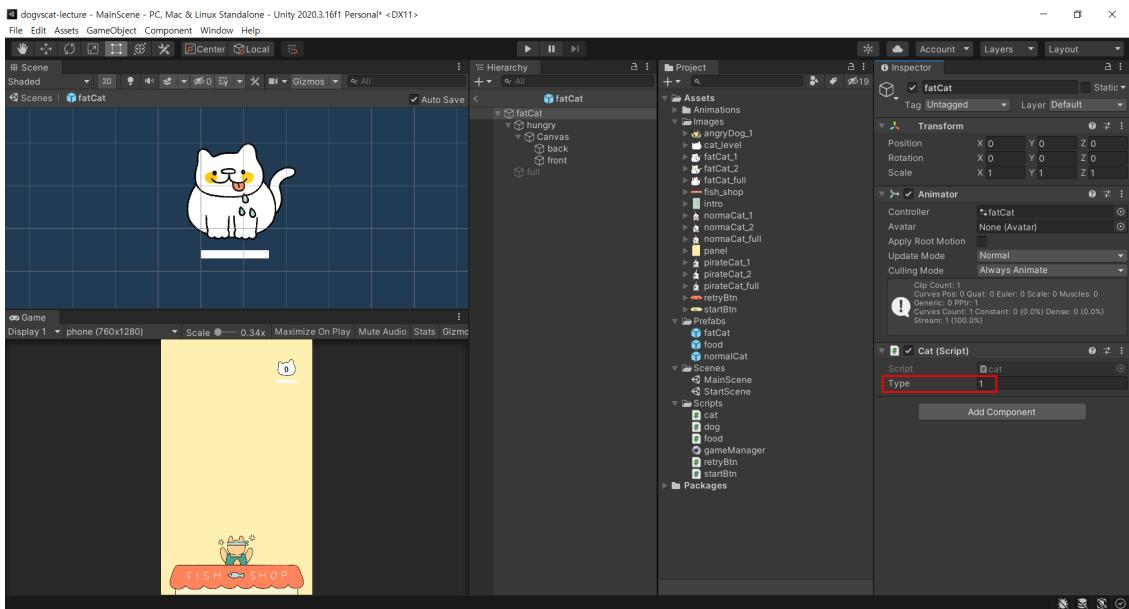


2. fatCat 프리팹 준비하기

→ front의 `x scale: 0`으로 만들기

→ 스크립트 붙이고 `type: 1`로 만들기

→ box collider 2D를 붙이는 것도 잊지 말기!



3. 고양이 타입을 고려한 `cat.cs` 만들기

→ `type:1` (fatCat)인 경우 `full: 10` 이고, 내려오는 속도가 늦음

```
void Start()
{
    float x = Random.Range(-8.5f, 8.5f);
    float y = 30.0f;
    transform.position = new Vector3(x, y, 0);

    if (type == 1)
    {
        full = 10.0f;
    }
}
```

```
// 업데이트 구문 안의 if문
if (energy < full)
{
    if (type == 0)
    {
        transform.position += new Vector3(0.0f, -0.05f, 0.0f);
    }
}
```

```

        else if (type == 1)
        {
            transform.position += new Vector3(0.0f, -0.03f, 0.0f);
        }

        if (transform.position.y < -16.0f)
        {
            gameManager.I.gameOver();
        }
    }
}

```

4. `fatCat` 을 등장시키기

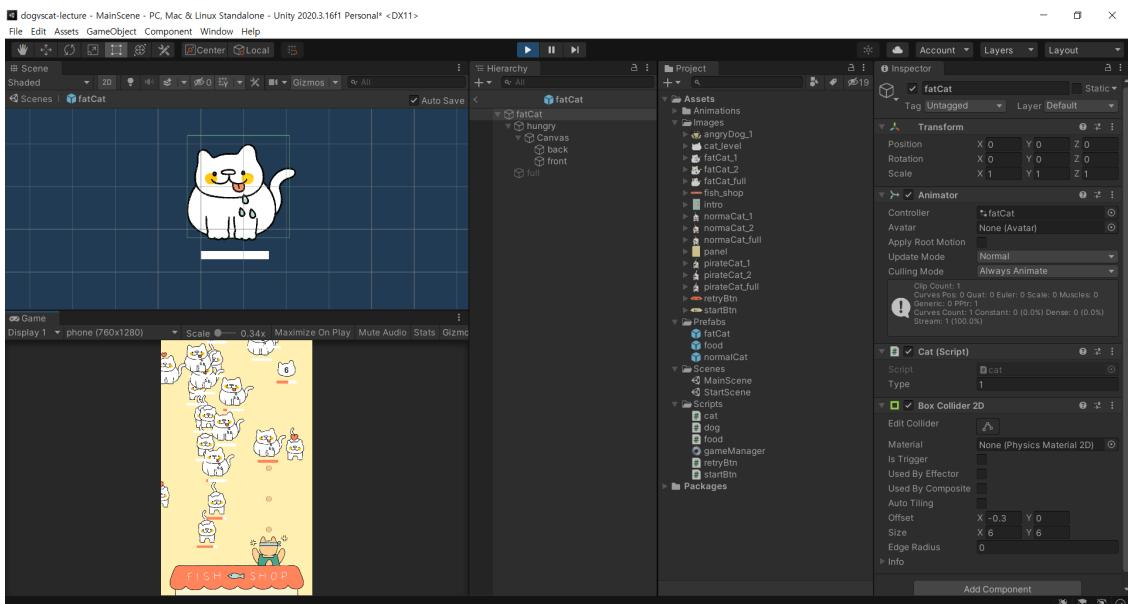
```

void makeCat()
{
    Instantiate(normalCat);

    if (level == 1)
    {
        float p = Random.Range(0, 10);
        if (p < 2) Instantiate(normalCat);
    }
    else if (level == 2)
    {
        float p = Random.Range(0, 10);
        if (p < 5) Instantiate(normalCat);
    }
    else if (level >= 3)
    {
        float p = Random.Range(0, 10);
        if (p < 5) Instantiate(normalCat);

        Instantiate(fatCat);
    }
}

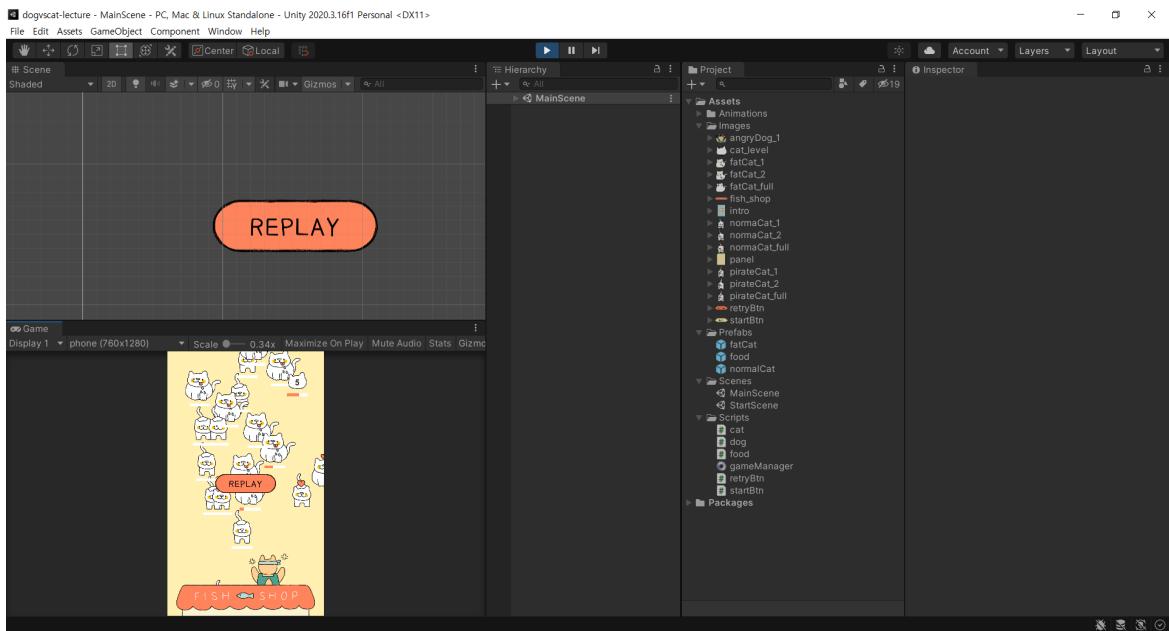
```



10. 마무리 - 게임 즐겨보기 & 버그잡기

- ▼ 1) startScene으로 가서 게임을 즐겨봅니다!

→ 앗, 그런데 `Replay` 후에 총알 발사가 안되네요!



▼ 2) gameManager 시작할 때 `timeScale = 1.0f` 로 바꿔주기

```
void Start()
{
    Time.timeScale = 1.0f;
    InvokeRepeating("makeFood", 0.0f, 0.1f);
    InvokeRepeating("makeCat", 0.0f, 1.0f);
}
```

11. 숙제 - 해적 고양이 만들기!



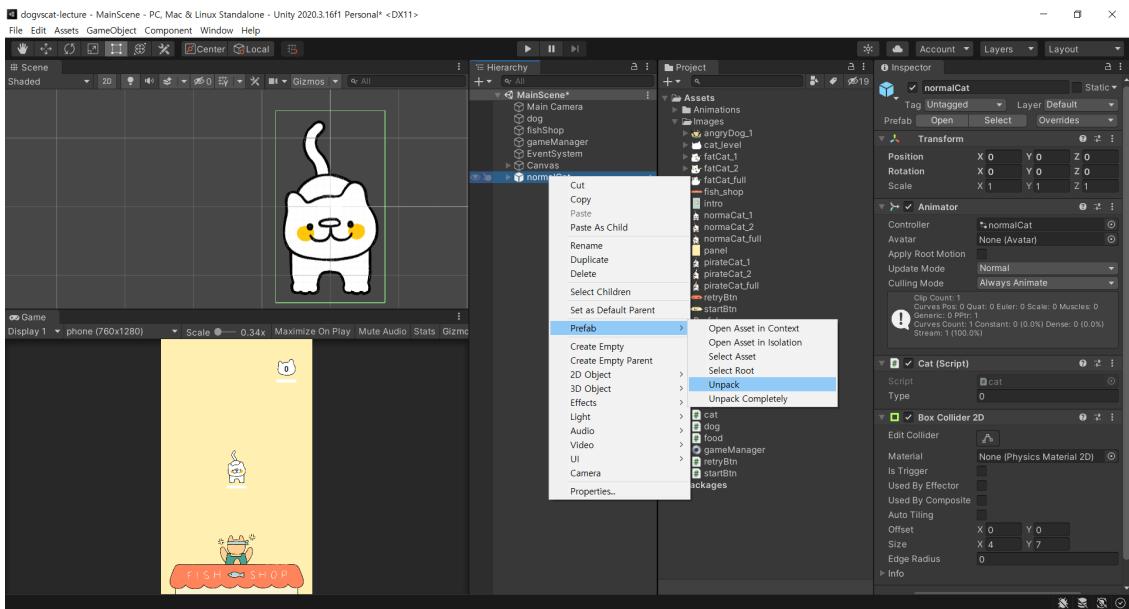
Lv.4 에서 해적고양이를 등장시킵니다! 준비하는 것은 튜터와 지금! 함께할께요 😎

▼ 해적고양이의 속성

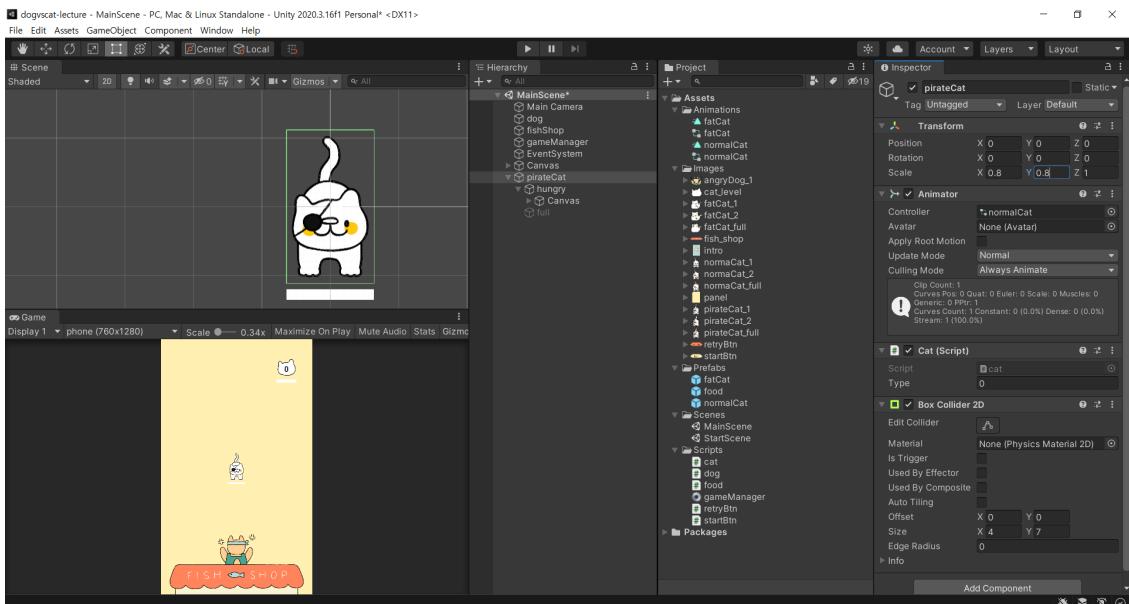
- normalCat 보다 사이즈가 작은 scale: `x:0.8`, `y:0.8`
- normalCat 보다 빠르게 내려옴 `-0.1f`

▼ 해적고양이 준비하기 (튜터와 함께)

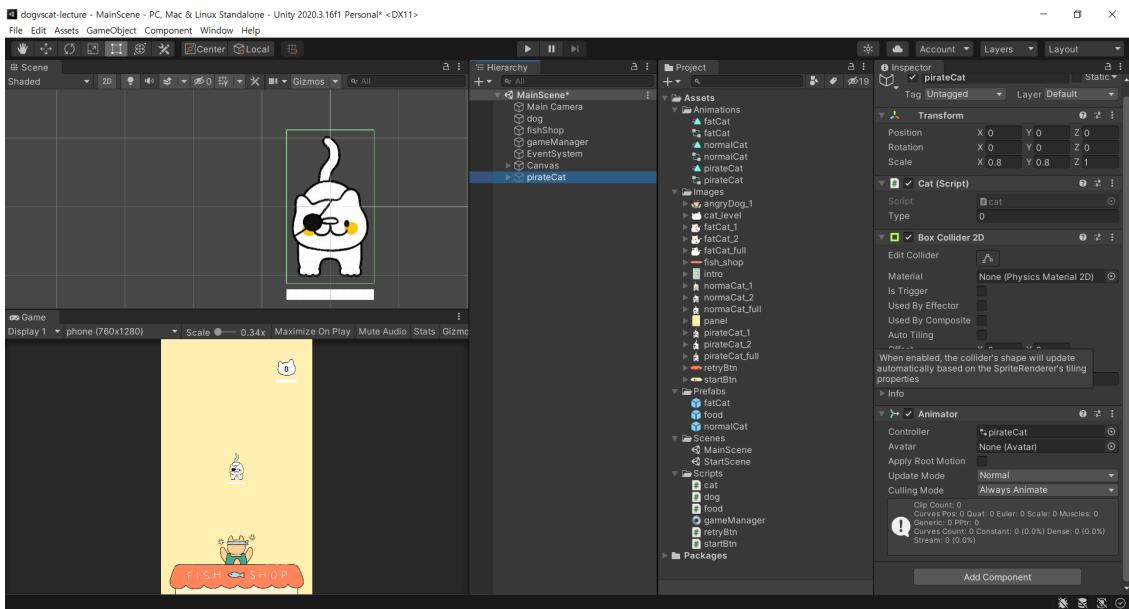
1. normalCat 프리팹을 가져와서, 오른쪽 키 → unpack 합니다.



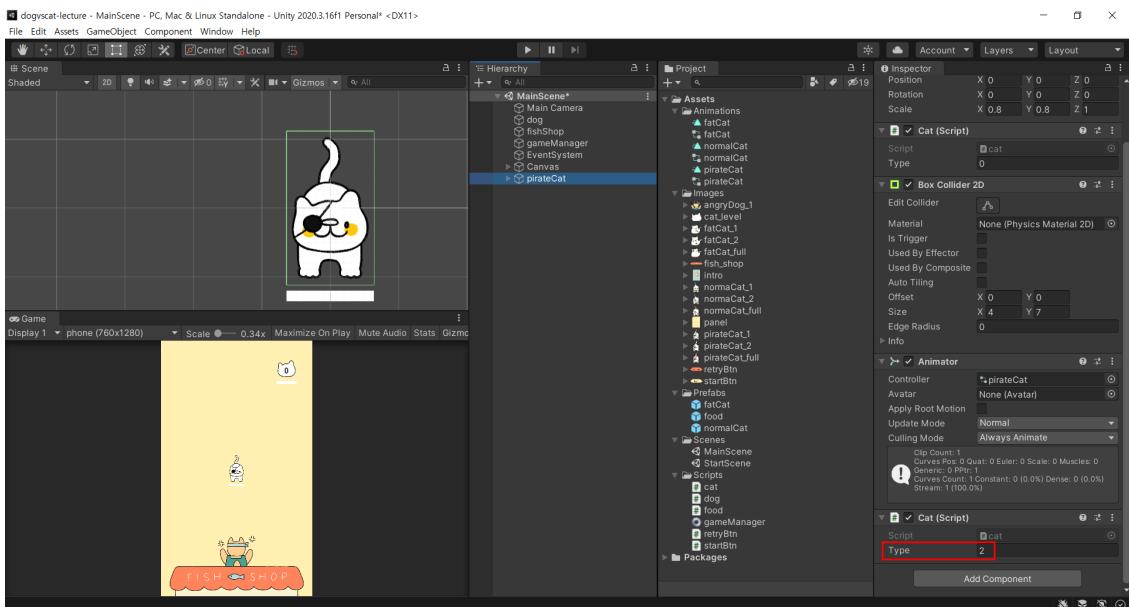
2. 이미지 교체해주고, pirateCat의 scale을 작게하기



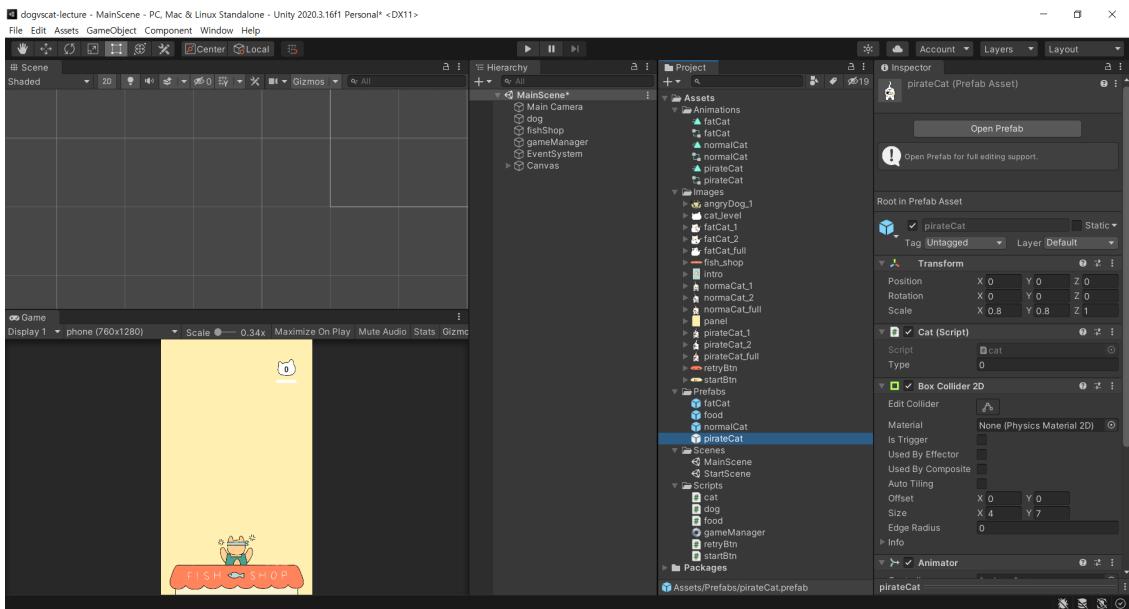
3. pirateCat의 animator를 떼어내고, 새로 붙여주기



4. `cat.cs` 스크립트 붙이고 `type: 2`로 바꿔두기



5. 프리팹 화 해두고 과감히 삭제하기



▼ 이렇게 되면 완성!

→ 제법 흥미진진하죠?

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1a5e6984-66e3-4f89-b85a-574580facd3b/고양이_밥주기_게임\(숙제\).mp4](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1a5e6984-66e3-4f89-b85a-574580facd3b/고양이_밥주기_게임(숙제).mp4)

▼ 힌트요정 - 🎉

[수정해야할 부분]

1. `cat.cs` 의 Update 부분 : type 나오는 곳에 아래를 추가해주기!

```
transform.position += new Vector3(0.0f, -0.1f, 0.0f);
```

2. `gameManager.cs` 의 makeCat 부분 : `level >= 4` 만들어주기!

→ `level >= 3` 부분은 `level == 3` 으로 바꿔야겠죠?

```
else if (level >= 4)
{
    float p = Random.Range(0, 10);
    if (p < 5) Instantiate(normalCat);

    Instantiate(fatCat);
    Instantiate(pirateCat);
}
```

HW. 3주차 숙제 해설

▼ `gameManager.cs` 부분

→ `makeCat()` 부분을 보세요

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class gameManager : MonoBehaviour
```

```

{
    public GameObject food;
    public GameObject dog;

    public GameObject normalCat;
    public GameObject fatCat;
    public GameObject pirateCat;

    public GameObject retryBtn;

    int level = 0;
    int cat = 0;

    public Text levelText;
    public GameObject levelFront;

    public static GameManager I;
    void Awake()
    {
        I = this;
    }

    // Start is called before the first frame update
    void Start()
    {
        Time.timeScale = 1.0f;
        InvokeRepeating("makeFood", 0.0f, 0.1f);
        InvokeRepeating("makeCat", 0.0f, 1.0f);
    }

    void makeFood()
    {
        float x = dog.transform.position.x;
        float y = dog.transform.position.y + 2.0f;
        Instantiate(food, new Vector3(x,y,0), Quaternion.identity);
    }

    void makeCat()
    {
        Instantiate(normalCat);

        if (level == 1)
        {
            float p = Random.Range(0, 10);
            if (p < 2) Instantiate(normalCat);
        }
        else if (level == 2)
        {
            float p = Random.Range(0, 10);
            if (p < 5) Instantiate(normalCat);
        }
        else if (level == 3)
        {
            float p = Random.Range(0, 10);
            if (p < 5) Instantiate(normalCat);

            Instantiate(fatCat);
        }
        else if (level >= 4)
        {
            float p = Random.Range(0, 10);
            if (p < 5) Instantiate(normalCat);

            Instantiate(fatCat);
            Instantiate(pirateCat);
        }
    }

    // Update is called once per frame
    void Update()
    {

    }

    public void gameOver()
    {
        retryBtn.SetActive(true);
        Time.timeScale = 0.0f;
    }

    public void addCat()
    {
        cat += 1;
        level = cat / 5;
    }
}

```

```

        levelText.text = level.ToString();
        levelFront.transform.localScale = new Vector3((cat - level * 5) / 5.0f, 1.0f, 1.0f);
    }
}

```

▼ **cat.cs** 부분

→ **Update()** 부분을 보세요

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class cat : MonoBehaviour
{
    float full = 5.0f;
    float energy = 0.0f;

    bool isFull = false;

    public int type;

    // Start is called before the first frame update
    void Start()
    {
        float x = Random.Range(-8.5f, 8.5f);
        float y = 30.0f;
        transform.position = new Vector3(x, y, 0);

        if (type == 1)
        {
            full = 10.0f;
        }
    }

    // Update is called once per frame
    void Update()
    {
        if (energy < full)
        {
            if (type == 0)
            {
                transform.position += new Vector3(0.0f, -0.05f, 0.0f);
            }
            else if (type == 1)
            {
                transform.position += new Vector3(0.0f, -0.03f, 0.0f);
            }
            else if (type == 2)
            {
                transform.position += new Vector3(0.0f, -0.1f, 0.0f);
            }

            if (transform.position.y < -16.0f)
            {
                gameManager.I.gameOver();
            }
        }
        else
        {
            if (transform.position.x > 0)
            {
                transform.position += new Vector3(0.05f, 0.0f, 0.0f);
            }
            else
            {
                transform.position += new Vector3(-0.05f, 0.0f, 0.0f);
            }
            Destroy(gameObject, 3.0f);
        }
    }

    void OnTriggerEnter2D(Collider2D coll)
    {
        if (coll.gameObject.tag == "food")
        {
            if (energy < full)
            {
                energy += 1.0f;
                Destroy(coll.gameObject);

                gameObject.transform.Find("hungry/Canvas/front").transform.localScale = new Vector3(energy / full, 1.0f, 1.0f);
            }
        }
    }
}

```

```
        else
        {
            if (isFull == false)
            {
                gameManager.I.addCat();
                gameObject.transform.Find("hungry").gameObject.SetActive(false);
                gameObject.transform.Find("full").gameObject.SetActive(true);

                isFull = true;
            }
        }
    }
}
```

이전 주차

다음 주차

Copyright © TeamSparta All rights reserved.