



[스파르타코딩클럽] 게임개발 종합반 - 4주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

[수업 목표]

1. 보드 게임을 만들어보기
2. 카드 뒤집기 게임을 만들면서, 총복습하기
3. 게임에 필요한 "로직"을 경험하기

[목차]

- [01. 오늘 배울 것](#)
- [02. 기본 씬 구성하기](#)
- [03. 카드 만들기 - 한 장](#)
- [04. 시간 가게 하기](#)
- [05. 카드 배치하기](#)
- [06. 카드에 르탄이 넣기](#)
- [07. 카드 애니메이션](#)
- [08. 카드 뒤집기](#)
- [09. 카드 매칭하기](#)
- [10. 게임 끝내기](#)
- [11. 숙제 - 30초가 지나면 게임 끝내기](#)
- [HW. 4주차 숙제 해설](#)



모든 토클을 열고 닫는 단축키

Windows : **Ctrl** + **alt** + **t**

Mac : **⌘** + **⌥** + **t**

01. 오늘 배울 것

▼ 1) 4주차 수업의 목표와 범위



오늘의 키워드! 총 복습+보드게임의 시작

- 여태까지 배웠던 것들로 만들거예요!
- 카드 뒤집기 게임을 만들면서 보드게임 만들기의 기초를 배워봅니다.

→ 보드게임은 우리가 만들었던 게임과 살~짝 다르게, "로직"이 중요하답니다.

→ 지난 주차에 했던 것들이 새록새록 기억 날 거예요!

▼ 2) 오늘 만들 것 : [르탄이 카드 뒤집기 게임](#)

- 같은 모양의 르탄이를 뒤집는 게임! 해본 적 있죠? 😎

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/df06f082-6520-4b06-9685-abeae861e3ed/르탄이_맞추기.mp4

▼ 3) 만들 순서

1. 기본 씬 구성하기 : 배경, 타이머, 리소스 받아두기
2. 시간 보내기
3. 카드 깔기
4. 카드 뒤집기 애니메이션 만들기
5. 같은 카드를 뒤집었을 때 없애기

02. 기본 씬 구성하기

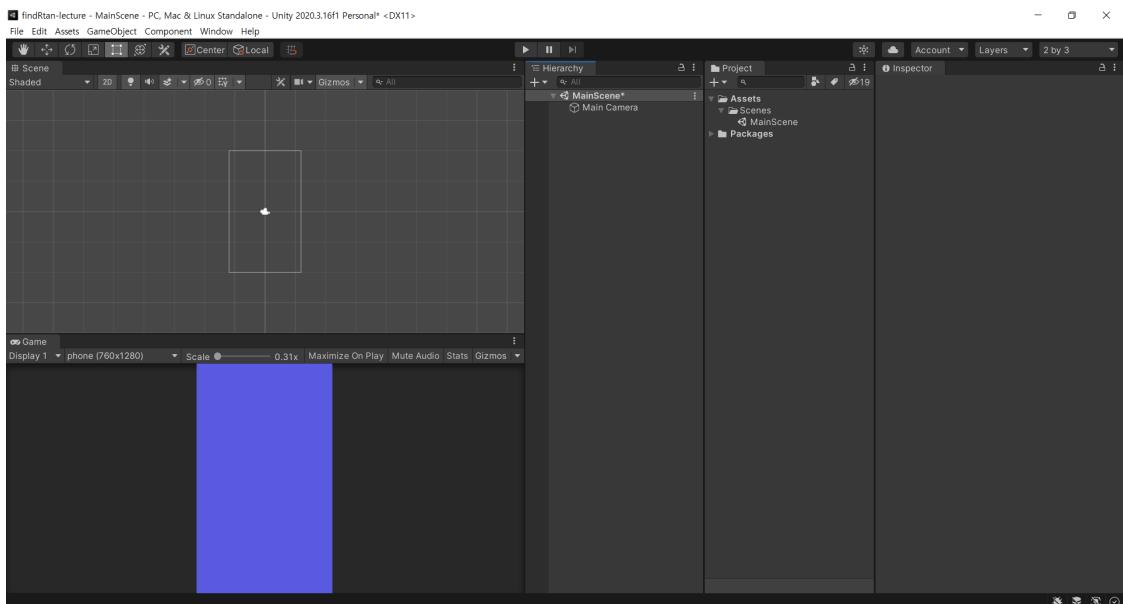


[findRtan](#) 이라는 프로젝트를 만들고 시작할게요!

▼ 1) 기본 세팅하기

- windows → 2x3 layout 클릭! free aspect → phone 클릭!

→ [rgb ⇒ 90, 90, 225](#) 로 맞출게요! (MainCamera)

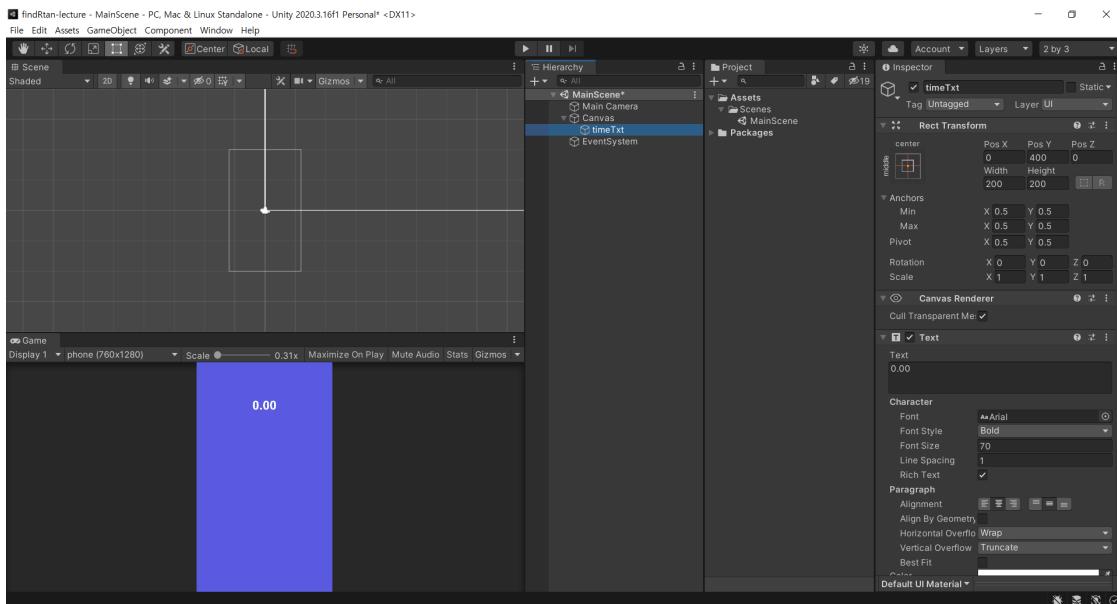


▼ 2) 타이머 만들어두기

- UI → Text 클릭 → timeTxt 로 만들어두기

→ font size: 70 , y: 400

→ width: 200, height: 200!



03. 카드 만들기 - 한 장

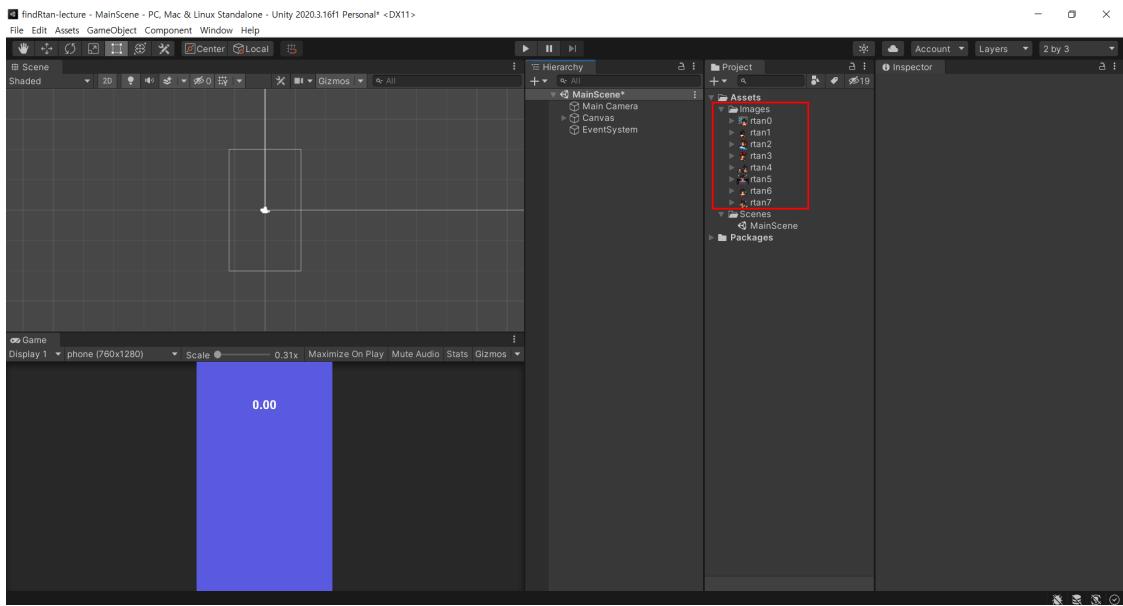
▼ 1) 르탄이 이미지 받아두기

▼ [코드스니펫] 르탄이 이미지 모음

https://s3.ap-northeast-2.amazonaws.com/materials.spartacodingclub.kr/game_new/week04/rtan.zip

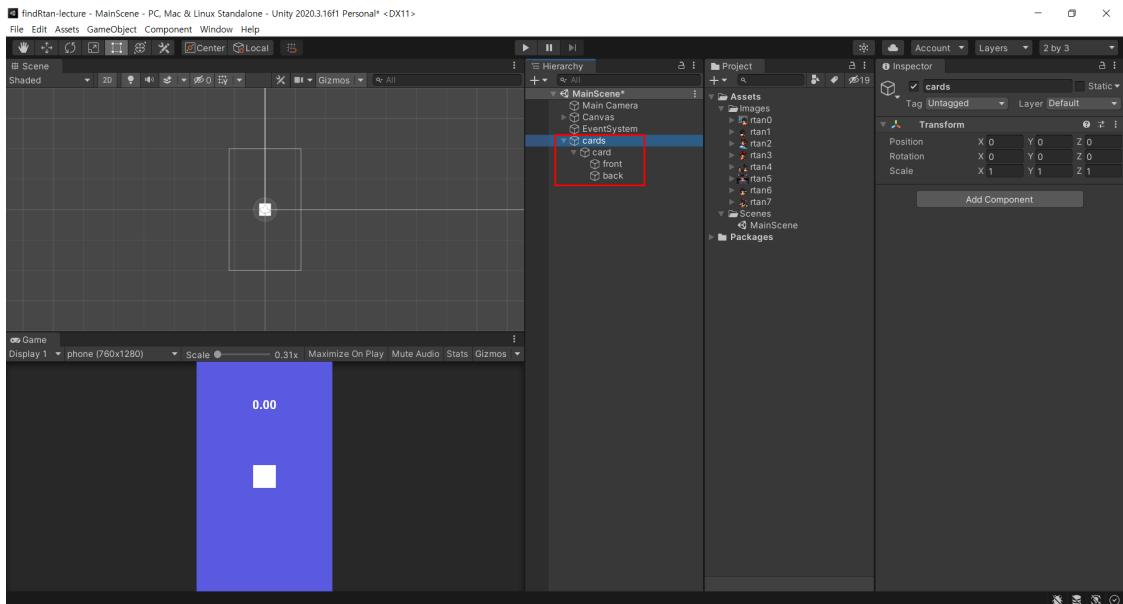
<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8f3843e9-bfa4-41f5-b1b6-a769162bfabf/rtan.zip>

1. Images 폴더 아래에 rtan 이미지 풀어두기

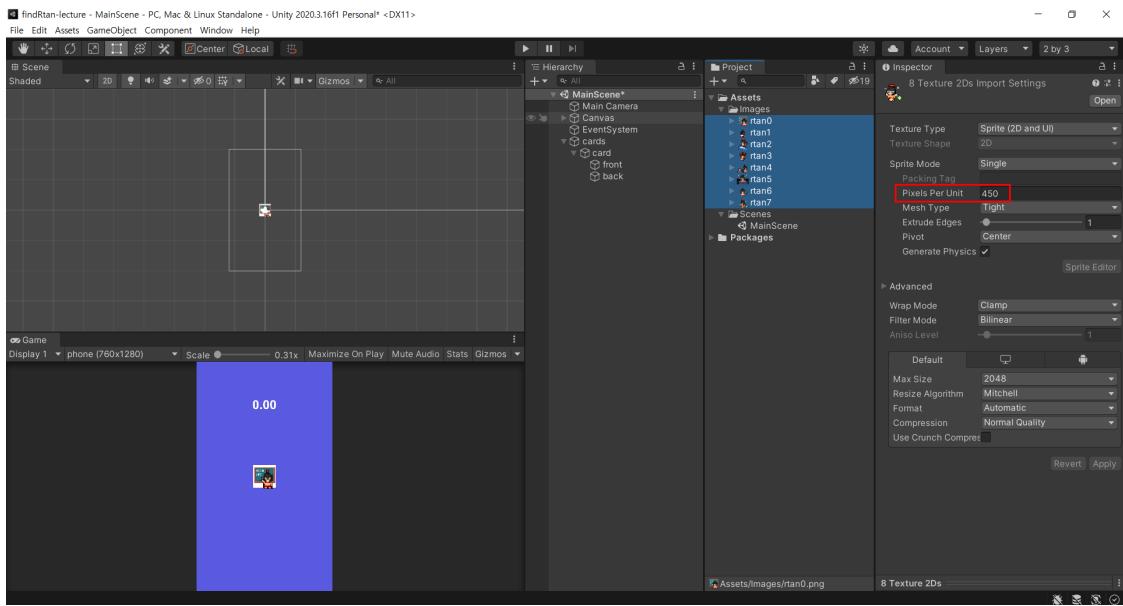


▼ 2) 카드 한 장만 만들어두기

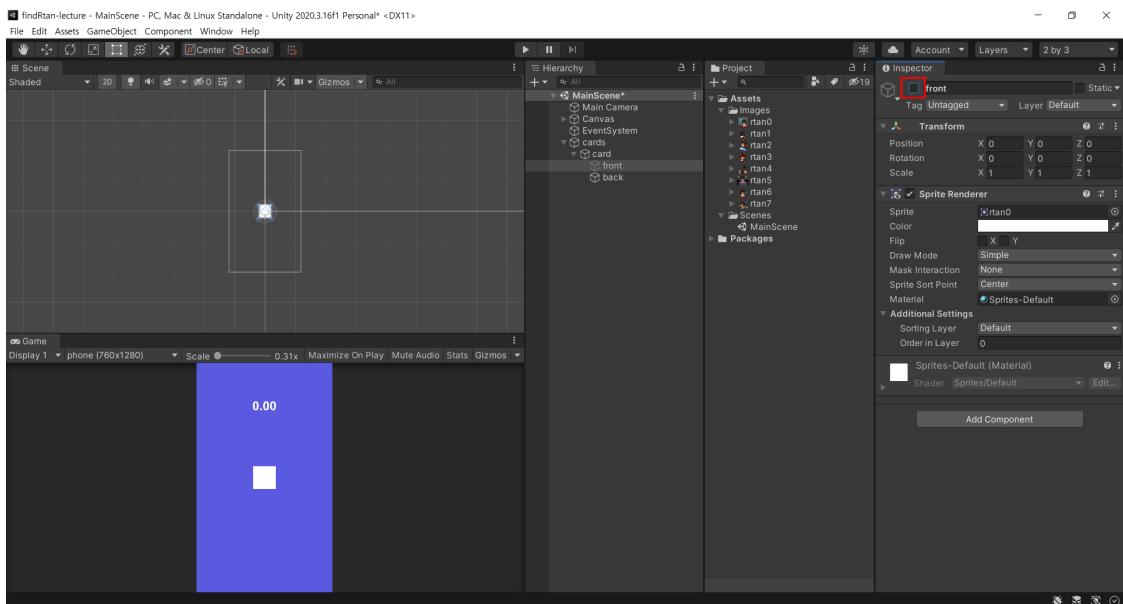
1. Cards (Create Empty!) 아래에 → card (Create Empty) 한 장만 만들어둘게요
 → 앞면(front)은 르탄이 이미지가 들어가고, 뒷면(back)은 ? 물음표가 들어갑니다.
 → card 아래에 front/back로 sprite를 만들어둘게요. 아래처럼!



2. **front** 스프라이트에 rtan0 이미지를 끌어다 놓습니다.
 → 앗, 너무 크네요! 이미지를 클릭해서 **pixels per unit** 을 450으로 맞춥니다.
 → 다른 르탄이들도 모두 450으로 맞춰주세요! (shift 눌러서 한번에 잡아 바꾸기)



3. 우선 `front` 는 꺼두겠습니다.



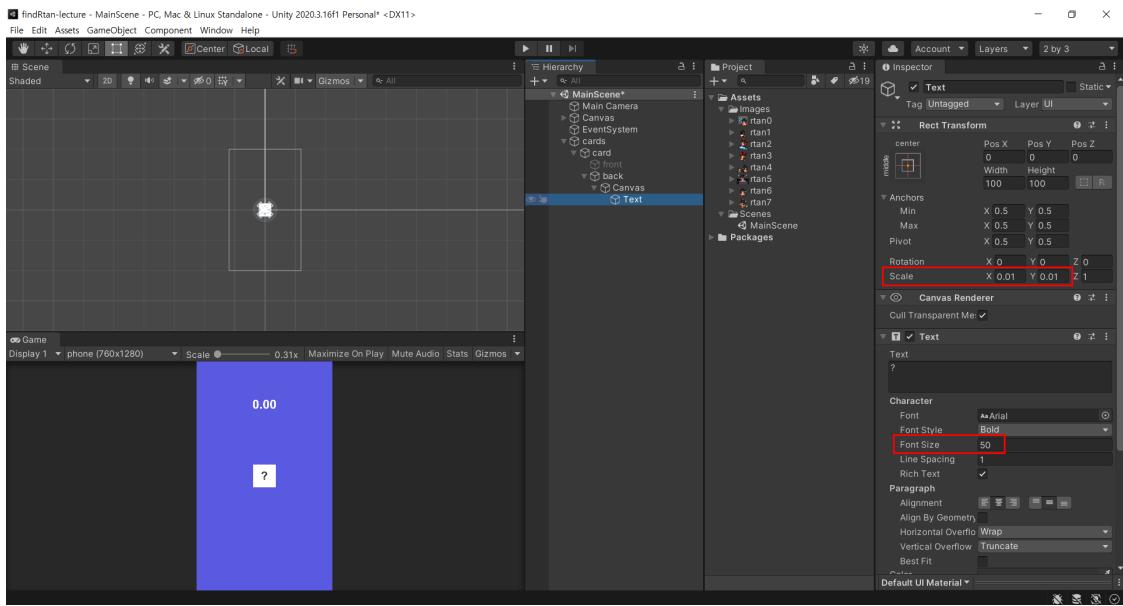
4. back 아래에 Canvas를 만들어 Text로 `?` 표시를 넣어주세요

Canvas

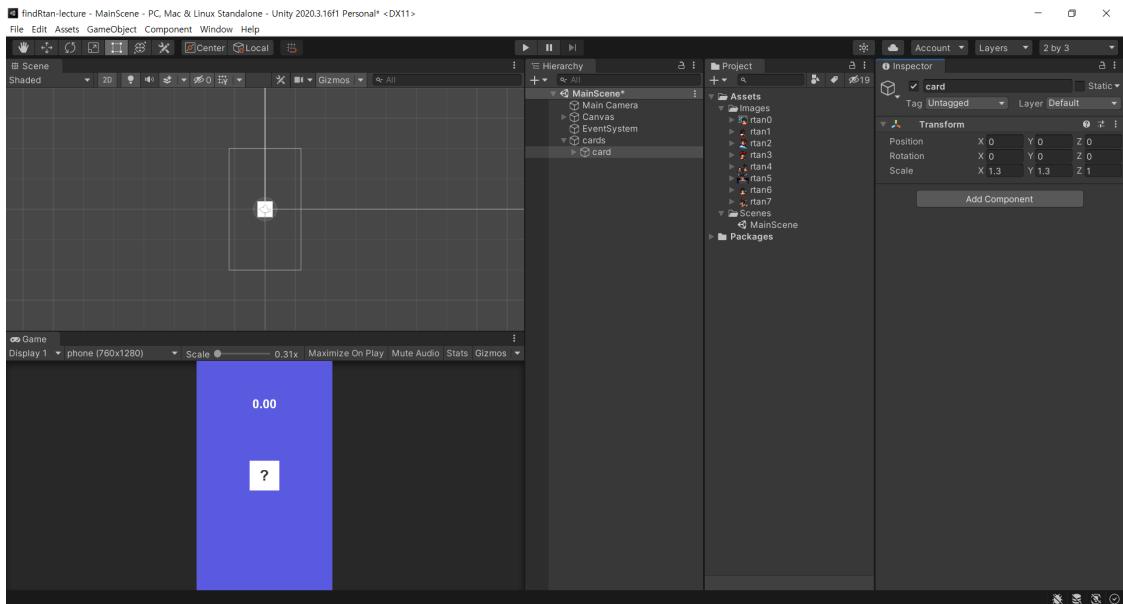
- Render-Mode를 `World Space` 로 바꾸기
- Order in layer를 `1` 로 바꾸기

Text

- Font Size: 50
- Scale을 수정해가며 사이즈를 맞추기 (0.01 , 0.01)



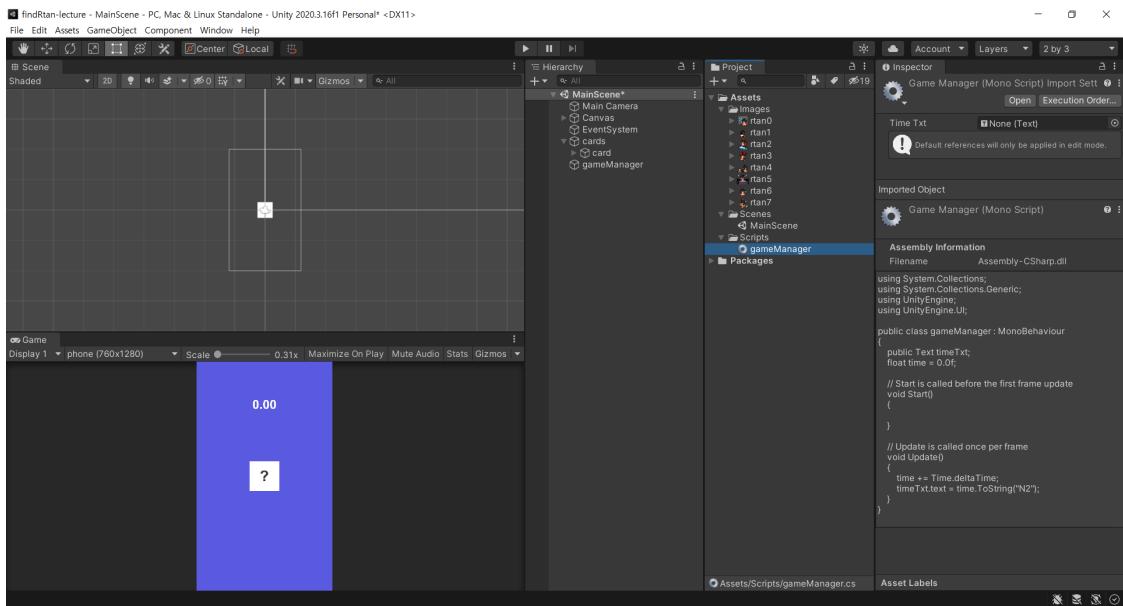
5. 마지막으로 Card의 Scale을 1.3으로 바꾸기



04. 시간 가게 하기

▼ 1) gameManager 세팅하기 (쉬운 것부터!)

1. gameManager 만들기



2. 시간이 가게 하기

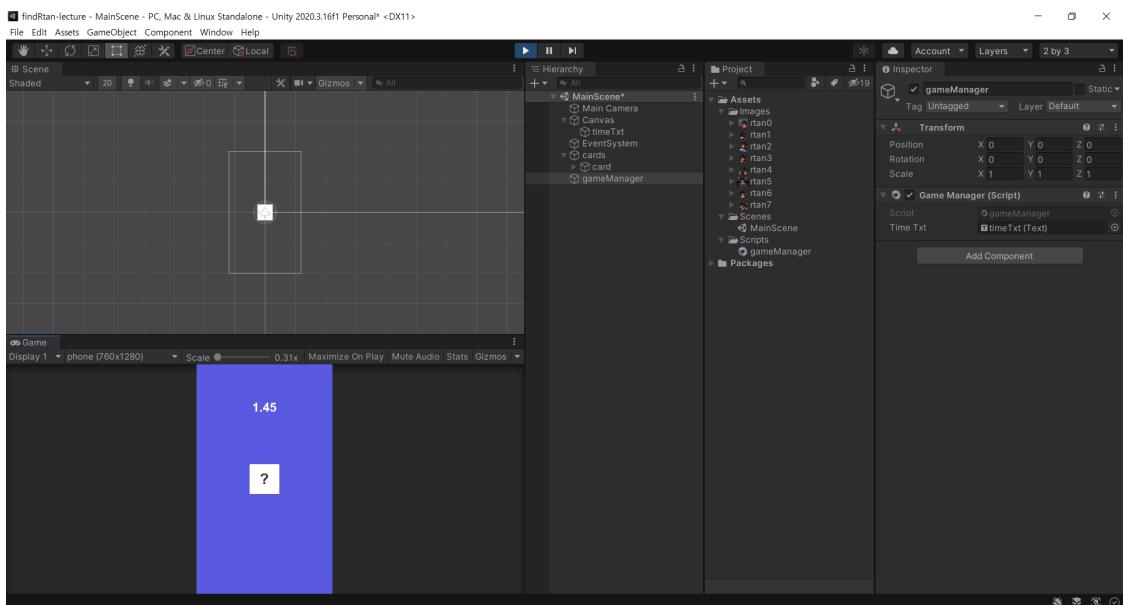
```

public Text timeTxt;
float time = 0.0f;

void Update()
{
    time += Time.deltaTime;
    timeTxt.text = time.ToString("N2");
}

```

3. Play 해서 확인해보기!



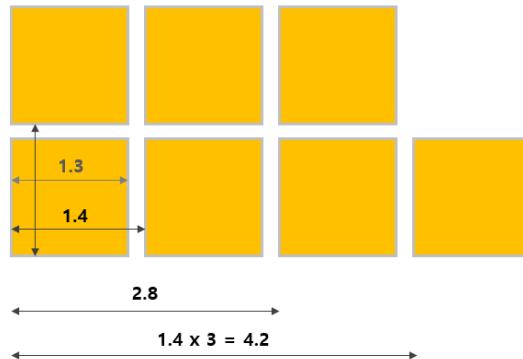
05. 카드 배치하기

▼ 1) 배치 전략



GameManager에서, 코드로 만들어서 불입시다!

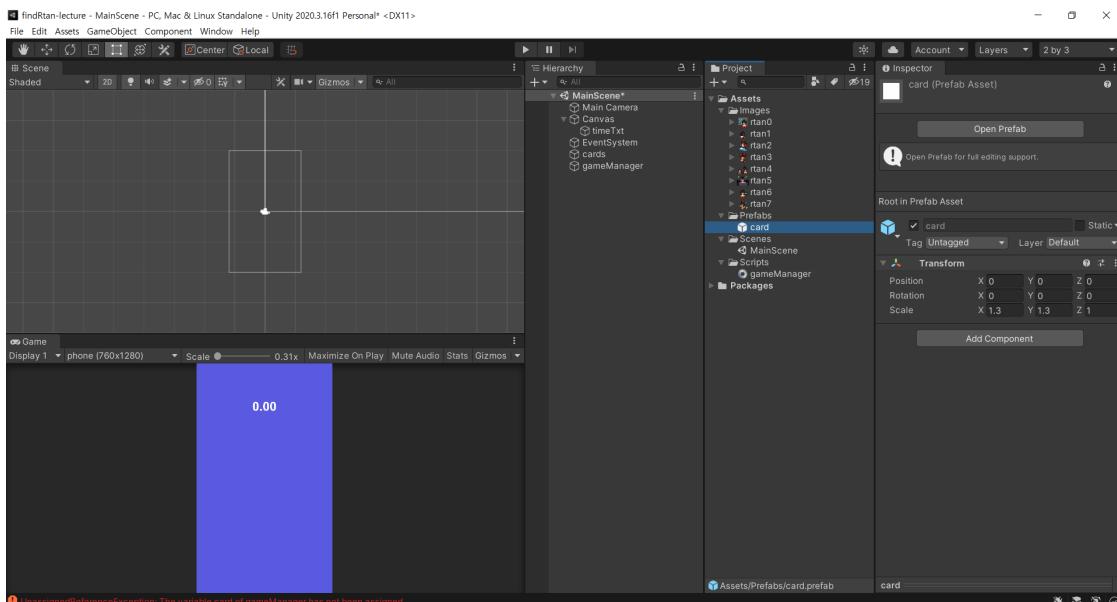
- 카드를 16장 만들어서 직접 배치하는 방법은 → 100장이면 너무 힘들잖아요!
- 지금 카드 사이즈가 `x:1.3, y:1.3` 이니까, `1.4` 만큼씩 띄워서 붙여주려고요!



▼ 2) 자동으로 카드 생성하기

1. card를 프리팹으로 만들기

→ 기존 것은 과감히 삭제하기!



2. 카드 생성하기 전에 반복문을 구경해보기 (튜터만 할게요!)



이렇게 하면 i 가 0~15까지 로그를 찍어준답니다. 즉, 하나씩 커지면서 반복!

```
void Start()
{
    for (int i = 0; i < 16; i++)
    {
        Debug.Log(i);
    }
}
```

3. card를 새로 만들어서 cards 아래에 붙이기

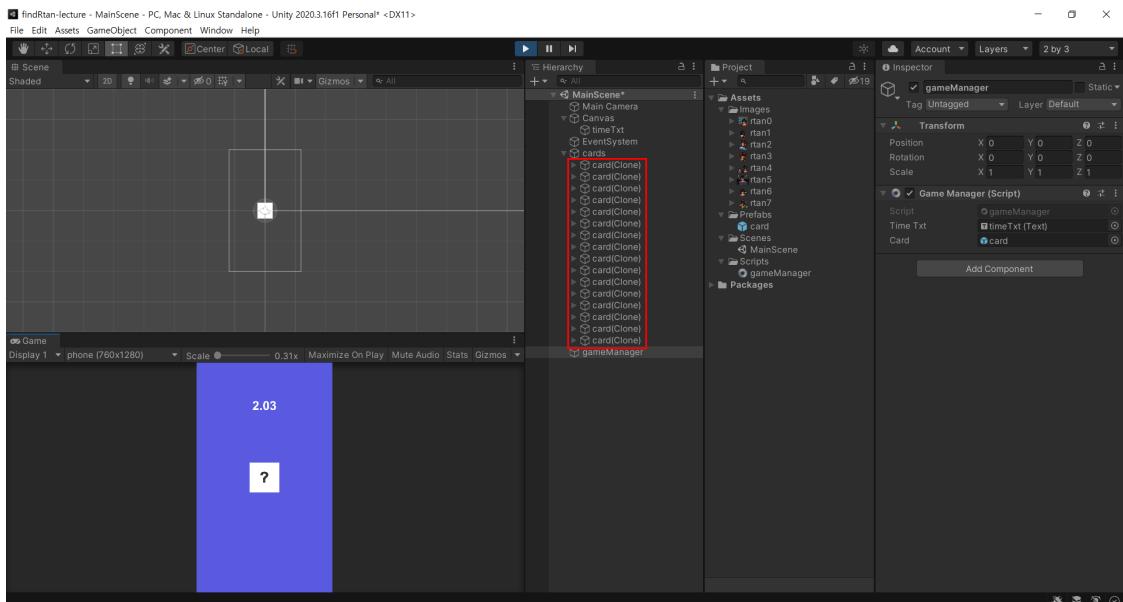
→ 실행해서 확인해볼까요?



Instantiate를 아래와 같이 받으면 생성된 오브젝트를 제어할 수 있습니다.

```
public GameObject card;

void Start()
{
    for (int i = 0; i < 16; i++)
    {
        GameObject newCard = Instantiate(card);
        newCard.transform.parent = GameObject.Find("cards").transform;
    }
}
```



▼ 3) 카드 위치 잡아주기

1. 전략을 생각하기



참고로 0번째부터 시작하니, 1번째는 사실상 두 번째 카드!

[예를 들어 1번째라고 하면]

→ x : 1을 4로 나눈 몫 = 0

→ y : 1을 4로 나눈 나머지 = 1

⇒ (0,1) 위치 ⇒ (1.4씩 곱해주면) ⇒ (0, 1.4) 위치

[예를 들어 7번째라고 하면]

→ x : 7을 4로 나눈 몫 = 1

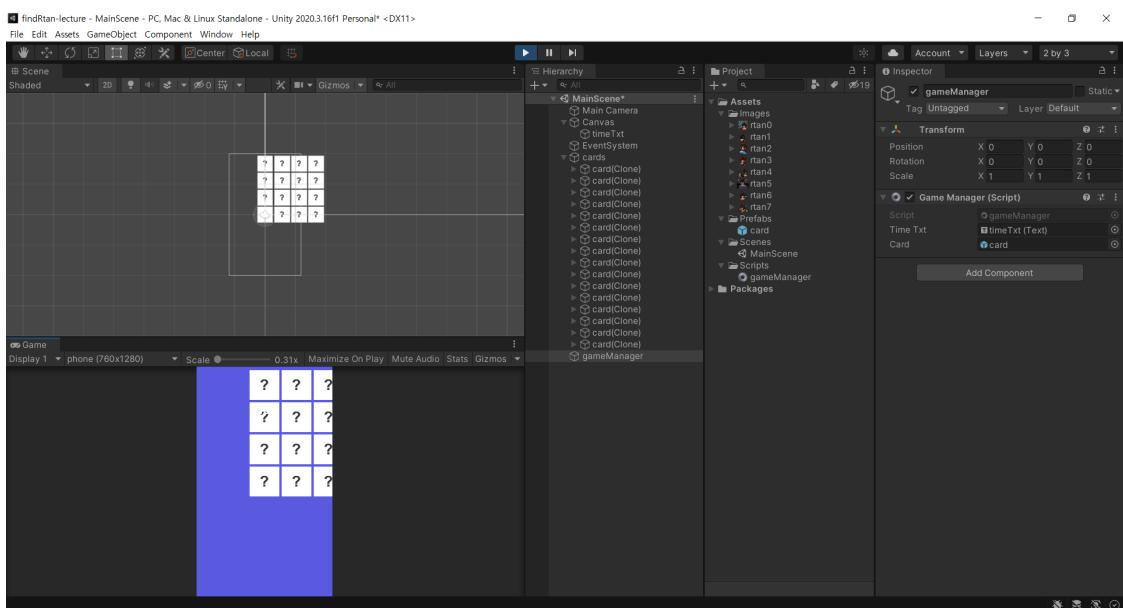
→ y : 7을 4로 나눈 나머지 = 3

⇒ (1,3) 위치 ⇒ (1.4씩 곱해주면) ⇒ (1.4, 4.2) 위치

2. 카드 위치 잡아주기

```
void Start()
{
    for (int i = 0; i < 16; i++)
    {
        GameObject newCard = Instantiate(card);
        newCard.transform.parent = GameObject.Find("cards").transform;

        float x = (i / 4) * 1.4f;
        float y = (i % 4) * 1.4f;
        newCard.transform.position = new Vector3(x, y, 0);
    }
}
```



3. 카드를 전체적으로 옮겨주기

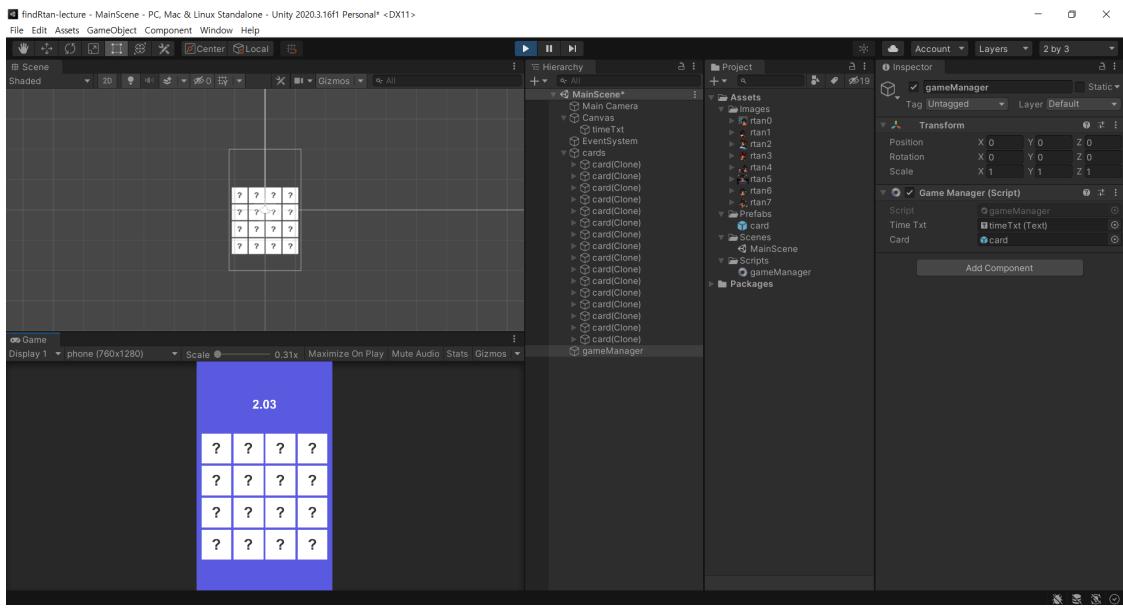
→ x, y를 적당히 빼줘서 전체를 가운데에 위치하게 하기

→ 여기서는 `x: -2.1f, y: -3.0f`

```
void Start()
{
    for (int i = 0; i < 16; i++)
    {
        GameObject newCard = Instantiate(card);
        newCard.transform.parent = GameObject.Find("cards").transform;

        float x = (i / 4) * 1.4f - 2.1f;
        float y = (i % 4) * 1.4f - 3.0f;
        newCard.transform.position = new Vector3(x, y, 0);
    }
}
```

4. 완성된 배치 구경하기



06. 카드에 르탄이 넣기

▼ 1) 랜덤으로 섞기 전략



C#에서 리스트 요소들을 랜덤하게 섞는 방법! (눈으로만 봐두기!)

1. 일단 [0, 0, 1, 1, 2, 2, ..., 7, 7]까지 쓰인 리스트를 만들고
2. 이걸 섞어서 [2, 3, 4, 1, 2, 0, 1, ..., 7]로 만들고
3. 카드가 만들어 질 때 하나씩 꺼내서 르탄이 이미지를 붙여주기!

▼ 2) 리스트를 랜덤으로 섞기

1. 우선 리스트 만들고 출력하기

→ 카드가 만들어지면서 `Debug.Log` 해보면 되겠죠?

```

void Start()
{
    int[] rtans = { 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7 };

    for (int i = 0; i < 16; i++)
    {
        GameObject newCard = Instantiate(card);
        newCard.transform.parent = GameObject.Find("cards").transform;

        float x = (i / 4) * 1.4f - 2.1f;
        float y = (i % 4) * 1.4f - 3.0f;
        newCard.transform.position = new Vector3(x, y, 0);

        Debug.Log(rtans[i]);
    }
}

```

2. 리스트를 섞기

→ 아래를 맨 위에 추가

```
using System.Linq;
```

▼ [코드스니펫] 랜덤하게 섞기

```
rtans = rtans.OrderBy(item => Random.Range(-1.0f, 1.0f)).ToArray();
```

```
void Start()
{
    int[] rtans = { 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7 };

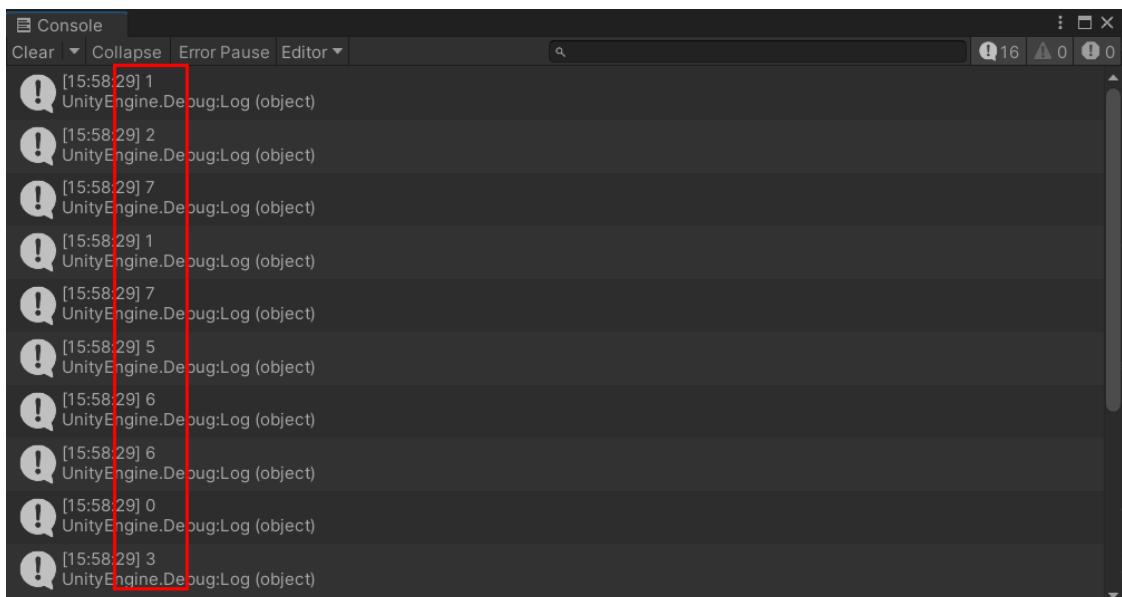
    rtans = rtans.OrderBy(item => Random.Range(-1.0f, 1.0f)).ToArray();

    for (int i = 0; i < 16; i++)
    {
        GameObject newCard = Instantiate(card);
        newCard.transform.parent = GameObject.Find("cards").transform;

        float x = (i / 4) * 1.4f - 2.1f;
        float y = (i % 4) * 1.4f - 3.0f;
        newCard.transform.position = new Vector3(x, y, 0);

        Debug.Log(rtans[i]);
    }
}
```

3. 해서 확인해보기

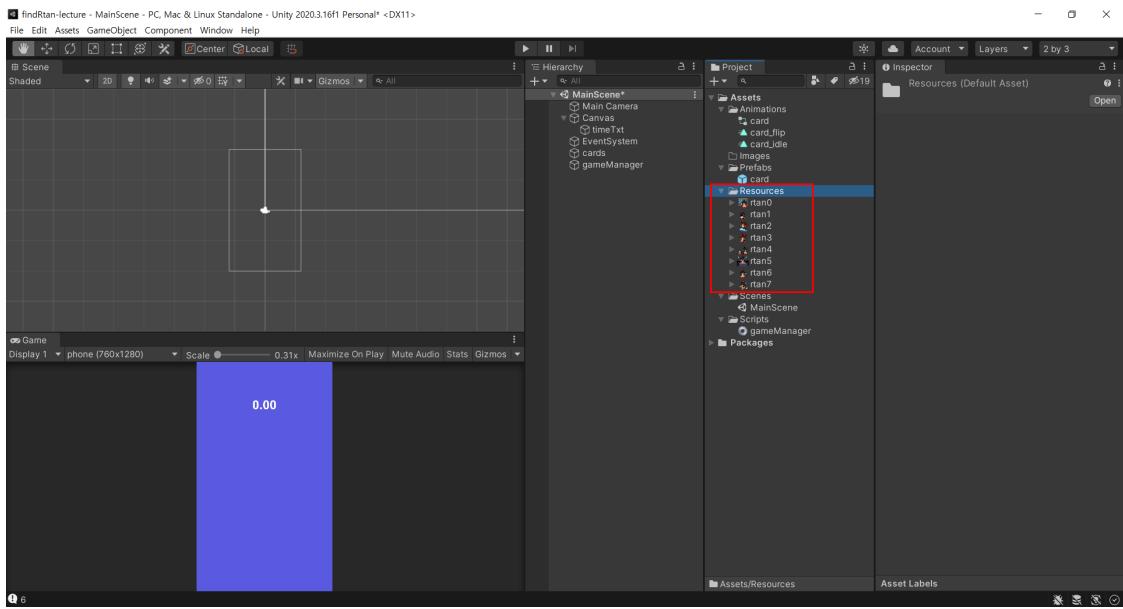


▼ 3) 르탄이 붙여주기

1. 이미지를 꺼내오려면? → Resources 폴더에 옮겨두기



어떤 이미지나 파일을 코드로 지정하고 싶을 때는 "Resources" 폴더에 옮겨두기



2. 르탄이 붙이기

```
string rtanName = "rtan" + rtans[i].ToString();
```

→ 르탄이 이름(이미지 이름)을 만들어두기

```
newCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite =
```

→ 새 카드 아래에 `front` 를 찾아서, `sprite`를 변경

```
Resources.Load<Sprite>(rtanName);
```

→ `Resources` 폴더에 있는 `rtanName` 이미지를 가져오자

```
void Start()
{
    int[] rtans = { 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7 };

    rtans = rtans.OrderBy(item => Random.Range(-1.0f, 1.0f)).ToArray();

    for (int i = 0; i < 16; i++)
    {
        GameObject newCard = Instantiate(card);
        newCard.transform.parent = GameObject.Find("cards").transform;

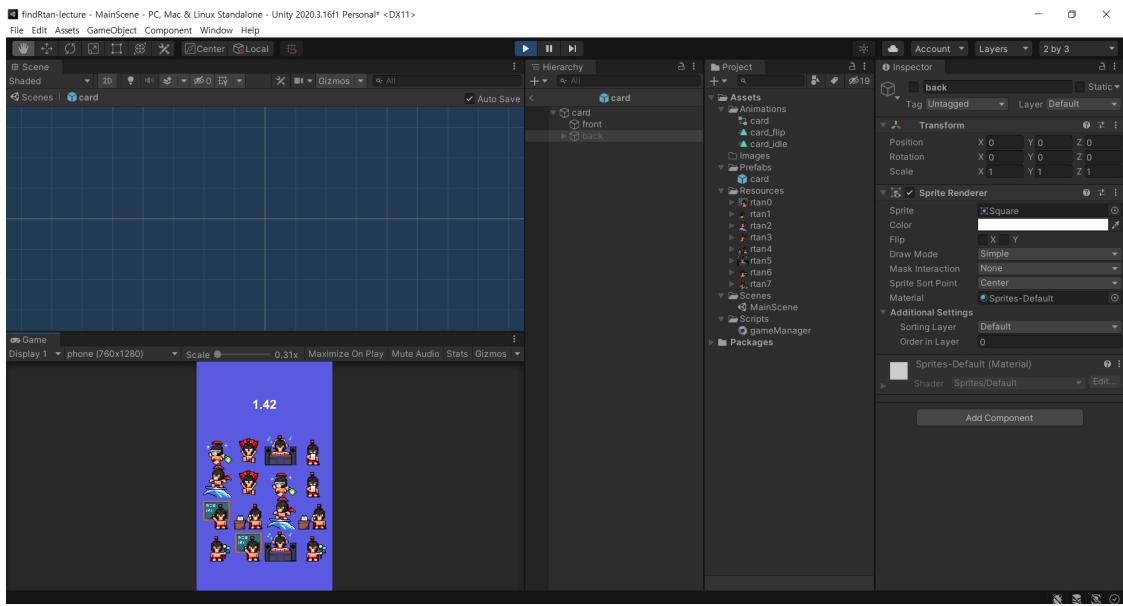
        float x = (i / 4) * 1.4f - 2.1f;
        float y = (i % 4) * 1.4f - 3.0f;
        newCard.transform.position = new Vector3(x, y, 0);

        string rtanName = "rtan" + rtans[i].ToString();
        newCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite = Resources.Load<Sprite>(rtanName);
    }
}
```

3. 확인해보기

→ 프리팹에서 `front`를 켜고, `back`을 끈 다음 확인합니다.

→ 잘 나오네요! 😎

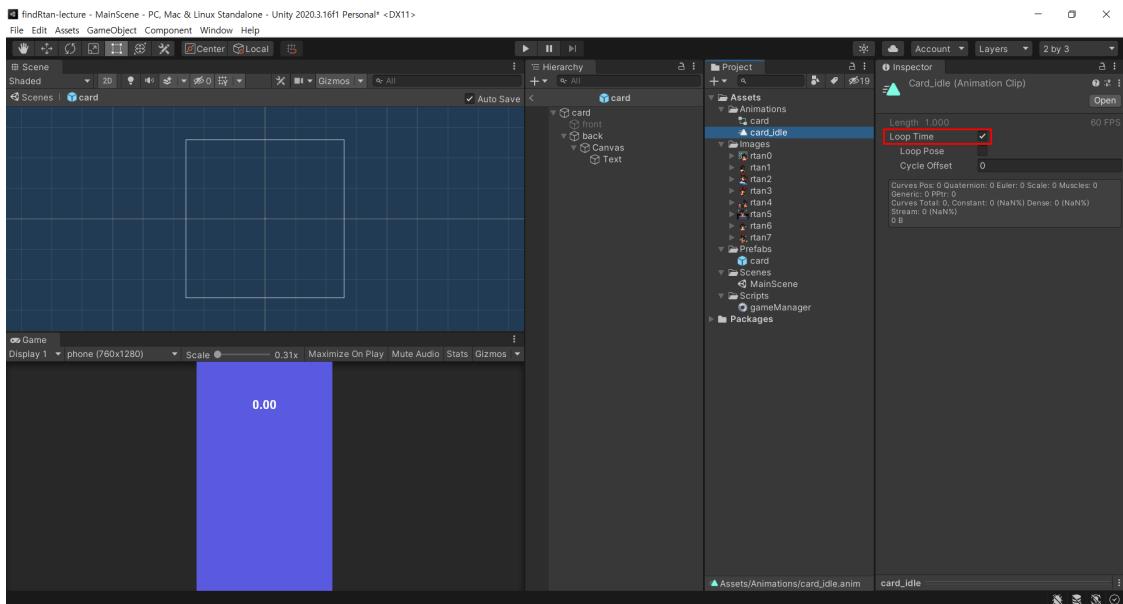


07. 카드 애니메이션

▼ 1) 기본 애니메이션 만들기 `card_idle`

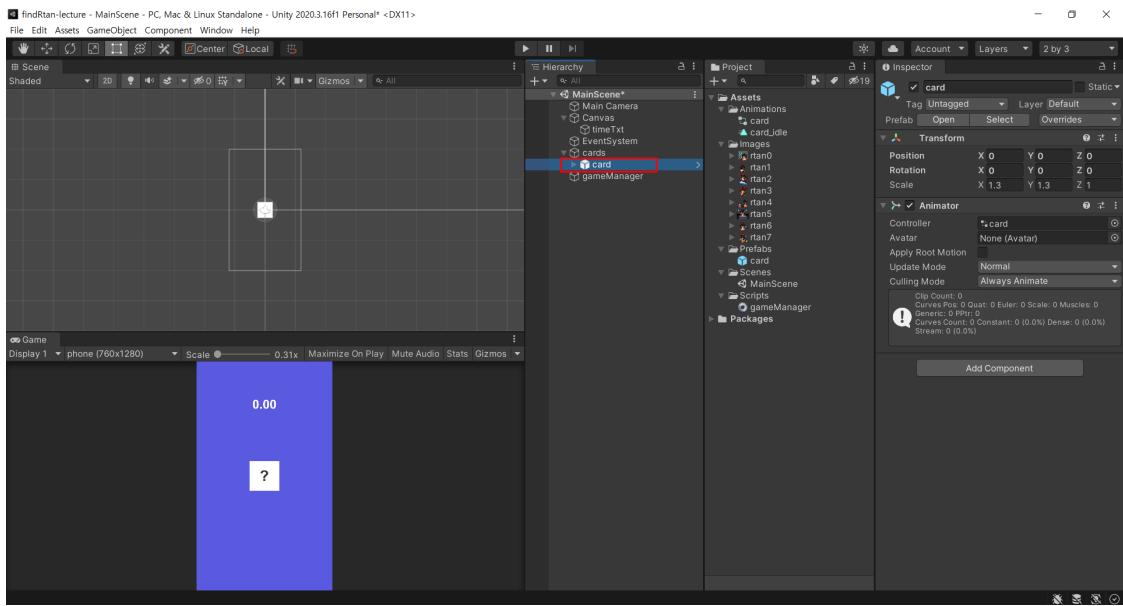
1. Animations 폴더 안에 `card_idle` 만들기

→ 프리팹 `card`를 열어서 붙여놓기, Loop Time 체크하는 것 잊지 말기!



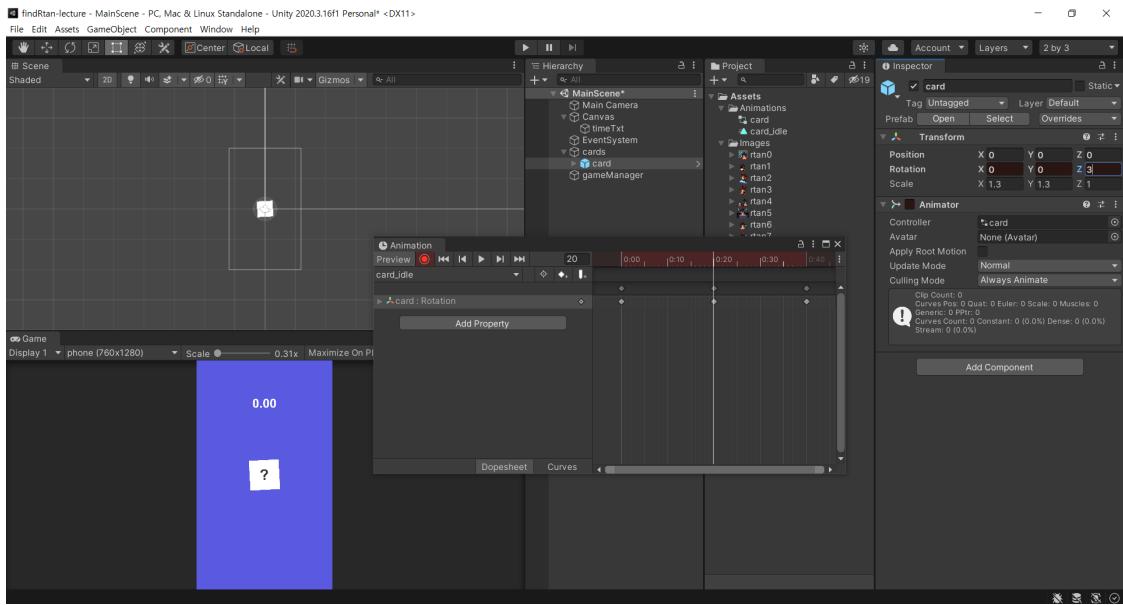
2. 카드를 꺼내놓기

→ 안보이면 만들기 어려우니까!



3. 애니메이션 레코딩하기

→ 0:20 예만 rotation z:3 을 만들어놓기

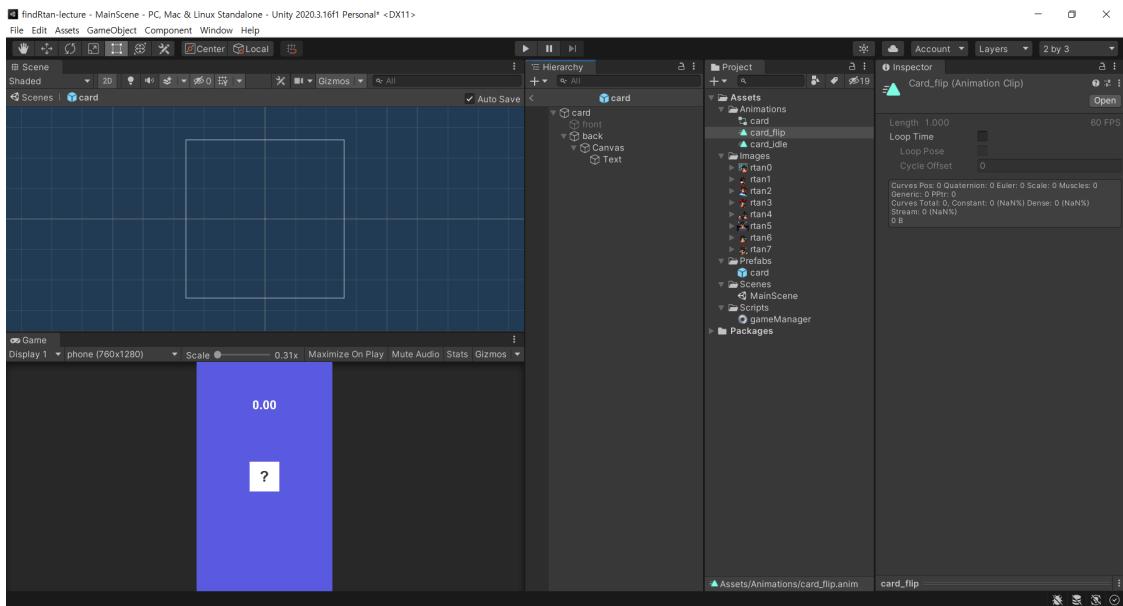


▼ 2) 뒤집기 애니메이션 만들기 card_flip

1. Animations 폴더 안에 card_flip 만들기

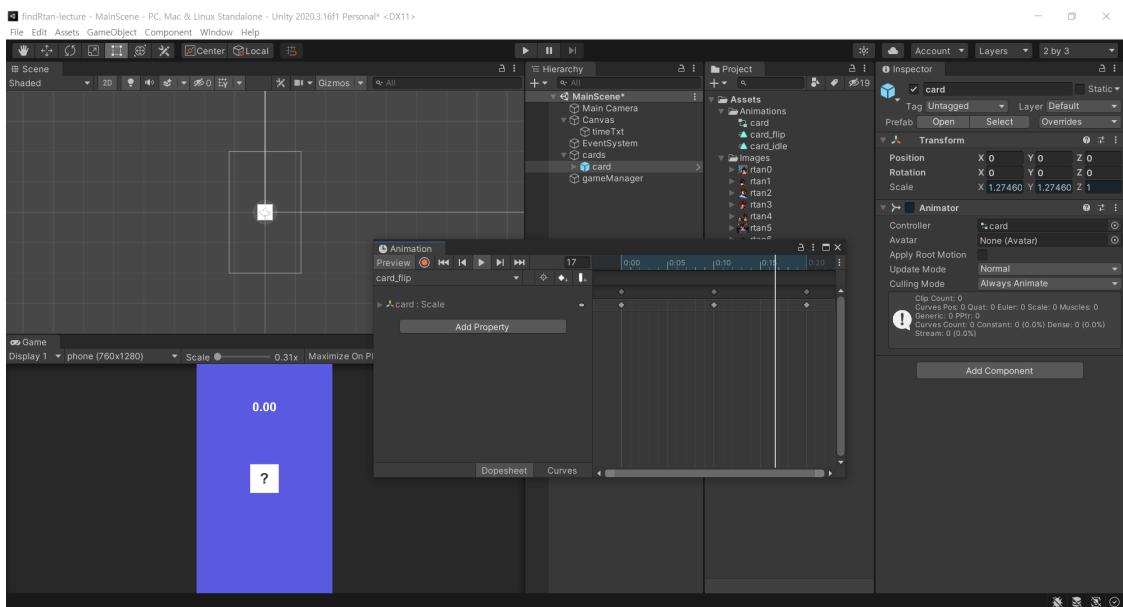
→ 프리팹을 열어 card에 붙여놓기

→ 뒤집기는 한번 이므로 loop time 체크할 필요 없음!



2. 애니메이션 만들기

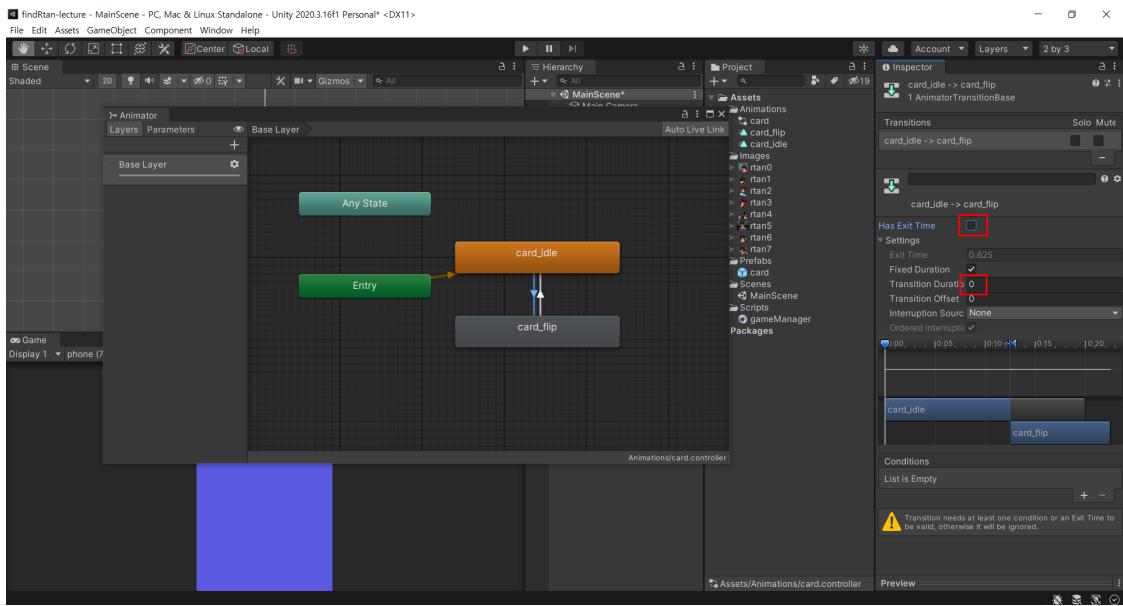
- 0:10 부분만 Scale 을 x:1.2, y:1.2 로 만들어두기
- 살짝 놀린 것처럼!



▼ 3) 애니메이션 조건 만들기

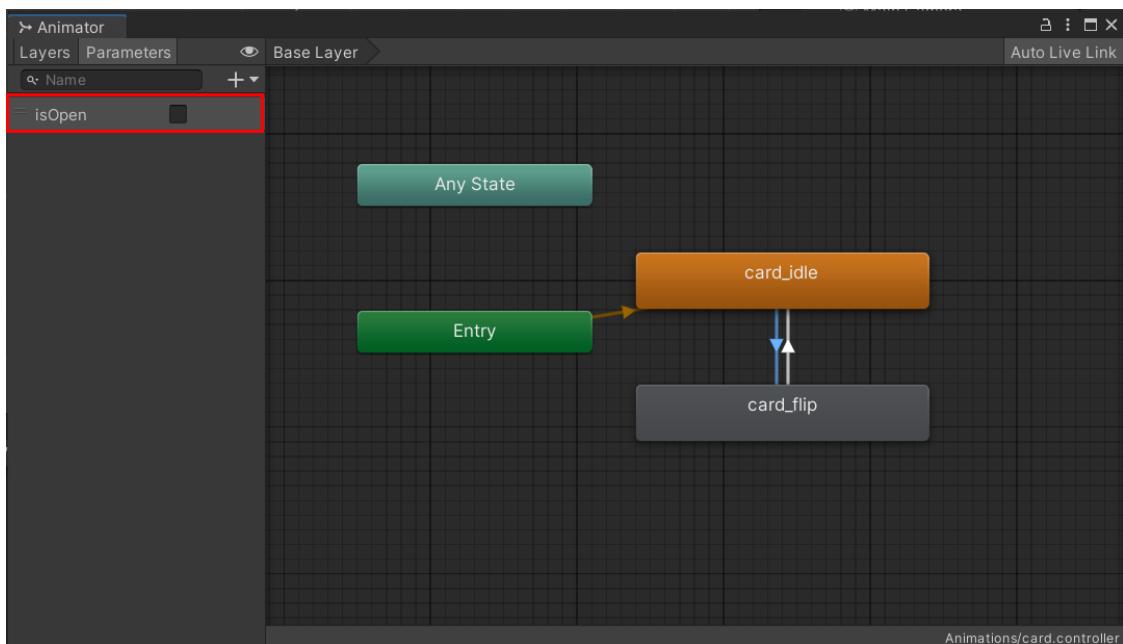
1. Animator 를 열고, transition 만들기

- 오른쪽 클릭하고 make transition. 오는 것 / 가는 것 두 개
- `has exit time` 에 체크 해제하고, transition duration 0으로 바꾸기



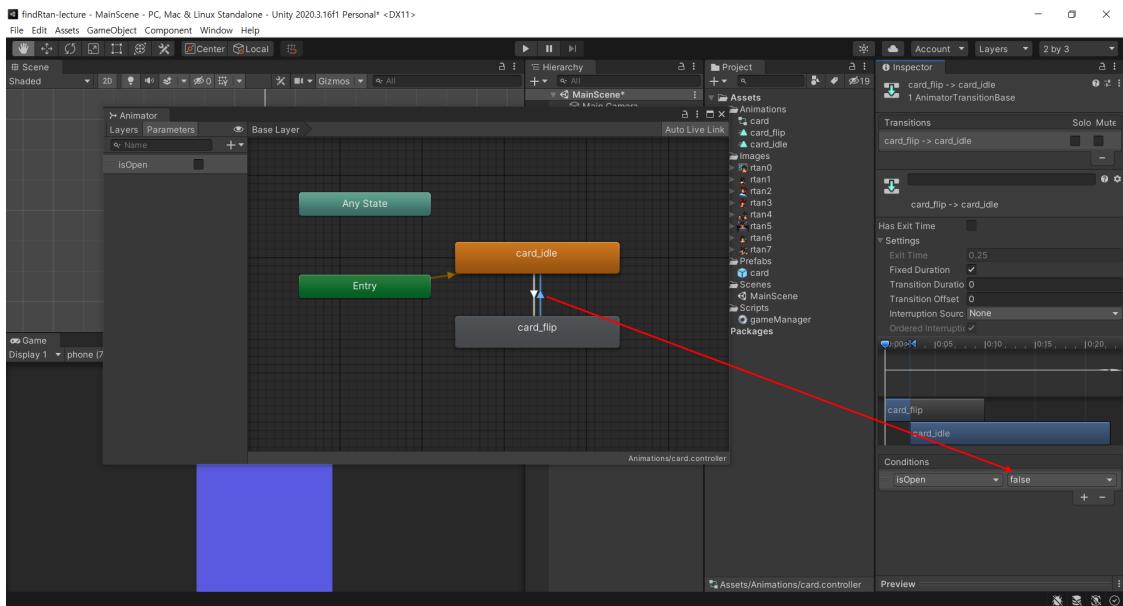
2. 파라미터 만들기

→ bool 형식(true / false)의 `isOpen`



3. transition에 파라미터 조건 붙이기

- `idle → flip` : bool이 true 가 되면 발동
- `flip → idle` : bool이 false 가 되면 발동



08. 카드 뒤집기

- ▼ 1) 카드 클릭하면 뒤집어지기 (=수어가기 😊)



프리펩 을 열어서 작업해야 한답니다!

1. 카드에 button 속성 붙이기

→ `card.cs` 만들어서, 클릭하면 front가 보이게 하기

```
public void openCard()
{
    transform.Find("front").gameObject.SetActive(true);
    transform.Find("back").gameObject.SetActive(false);
}
```

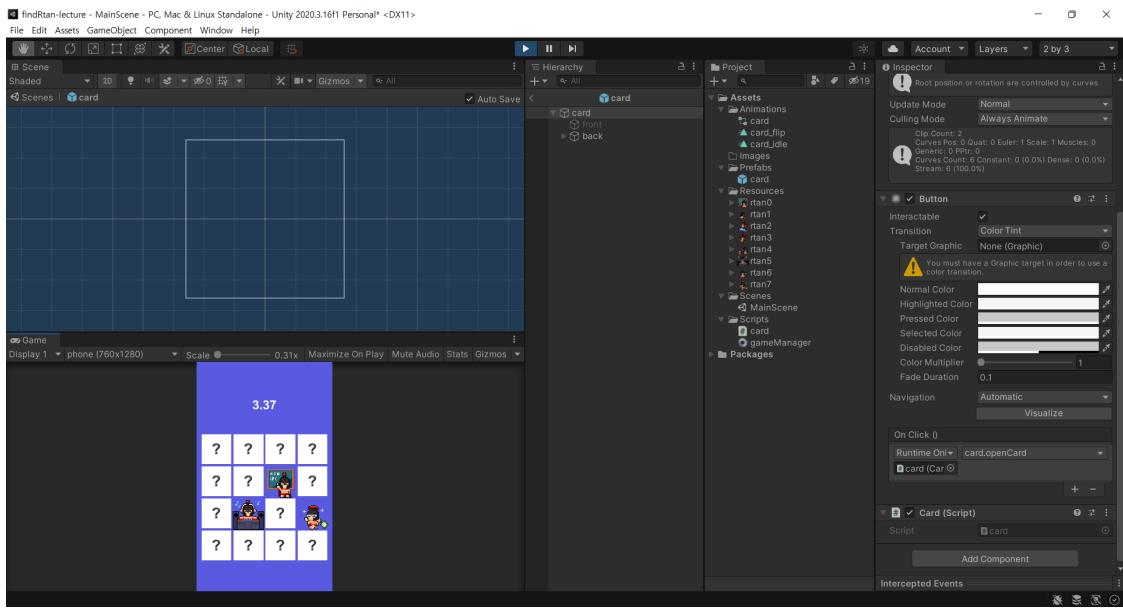
2. 카드에 animation 적용하기

```
public Animator anim;

public void openCard()
{
    anim.SetBool("isOpen", true);
    transform.Find("front").gameObject.SetActive(true);
    transform.Find("back").gameObject.SetActive(false);
}
```

3. `Play` 해서 잘 뒤집어지는지 실행해보기

→ 잘 되네요!



09. 카드 매칭하기

▼ 1) 카드 매칭하기 - 전략

👉 첫 번째 카드의 이름을 갖고 있다가 → 두 번째 카드와 일치하는지 보기

- 일치하면? ⇒ (1초 후에) 카드를 둘 다 없애주기
- 안 일치하면? ⇒ (1초 후에) 두 카드를 다시 뒤집어주기

▼ 2) 카드 매칭하기

1. 우선, gameManager 싱글톤화

→ 이제 다른 곳에서 막 부를 것이니까!

```
public static gameManager I;

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

2. gameManager에서 카드 이름을 저장해둘 수 있게 하기 + 매칭 로직 함수 만들어두기

```
public GameObject firstCard;
public GameObject secondCard;

public void isMatched()
{
    Debug.Log("판단하자");
}
```

3. `card.cs`에서 `openCard`하면 `firstCard` 또는 `secondCard`에 나를 넣기

```
public void openCard()
{
```

```

anim.SetBool("isOpen", true);
transform.Find("front").gameObject.SetActive(true);
transform.Find("back").gameObject.SetActive(false);

if (gameManager.I.firstCard == null)
{
    gameManager.I.firstCard = gameObject;
}
else
{
    gameManager.I.secondCard = gameObject;
    gameManager.I.isMatched();
}
}

```

4. `gameManager.cs`에서 `isMatched()` 만들기

- `front` 를 찾아서 카드 이미지 이름을 받아오기
- 다 끝났으면 다시 비워주기

```

public void isMatched()
{
    string firstCardImage = firstCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite.name;
    string secondCardImage = secondCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite.name;

    if (firstCardImage == secondCardImage)
    {
        Debug.Log("같다!");
    }
    else
    {
        Debug.Log("같지 않다!");
    }

    firstCard = null;
    secondCard = null;
}

```

5. card 행동 준비하기

- `card.cs` 안에 없애기, 뒤집어주기 함수 만들어두기
- 같으면 → 1초 후에 둘 다 없애기 / 다르면 → 1초 후에 둘 다 뒤집어주기
- 1초 후에 실행해야하니까, `invoke`를 따로 만들어줘야 하겠죠?

 `gameManager`에서는 `destoryCard` 를 부를 것이니까,
부르고나면 1초 뒤에 `destroyCardInvoke` 가 실행되도록!

```

public void destoryCard()
{
    Invoke("destroyCardInvoke", 1.0f);
}

void destroyCardInvoke()
{
    Destroy(gameObject);
}

public void closeCard()
{
    Invoke("closeCardInvoke", 1.0f);
}

void closeCardInvoke()
{
    anim.SetBool("isOpen", false);
    transform.Find("back").gameObject.SetActive(true);
}

```

```

        transform.Find("front").gameObject.SetActive(false);
    }

```

6. `gameManager.cs` 에서 같을 때 / 같지 않을 때 적절한 함수 불러주기

```
firstCard.GetComponent<card>().destroyCard()
```

→ `firstCard`에 붙어있는 `card.cs`에서 `destoryCard`를 불러라!

```

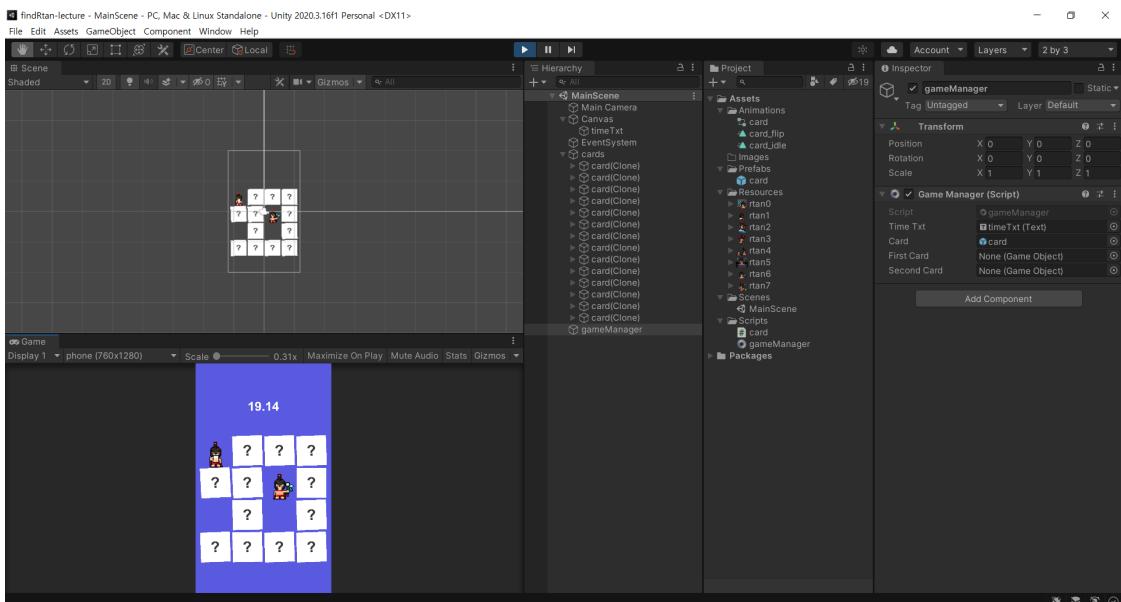
public void checkMatched()
{
    string firstCardImage = firstCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite.name;
    string secondCardImage = secondCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite.name;

    if (firstCardImage == secondCardImage)
    {
        firstCard.GetComponent<card>().destroyCard();
        secondCard.GetComponent<card>().destroyCard();
    }
    else
    {
        firstCard.GetComponent<card>().closeCard();
        secondCard.GetComponent<card>().closeCard();
    }

    firstCard = null;
    secondCard = null;
}

```

7. 확인해보기 ⇒ 잘 되네요! 😎



10. 게임 끝내기

- ▼ 1) 카드가 모두 없어지면 게임 끝내기

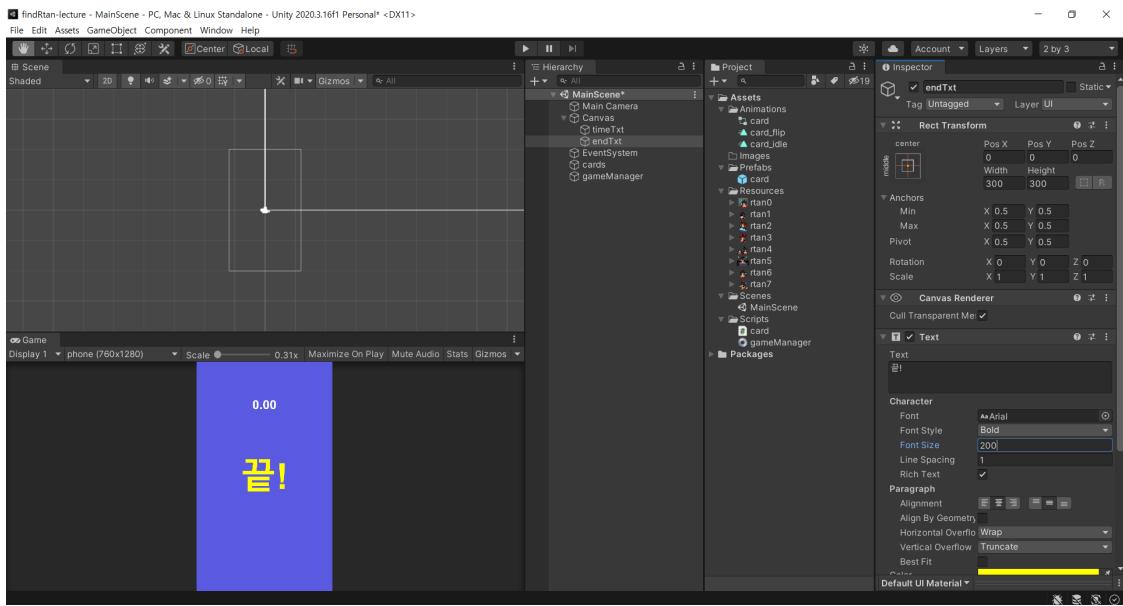
1. 끝! 텍스트를 미리 만들어두기

→ `timeTxt` 를 `ctrl+d` 해서 만들면 편하겠죠! `fontsize 200`

→ `width: 300, height: 300, posY: 0`

→ `color : 220, 255, 0, 255`

→ 안보이게 감춰두세요!



2. `gameManager.cs` 에서, 매칭 됐을 때 남은 카드를 체크하기

- cards를 찾아서, 아래에 몇 개의 자식이 있는지를 판단하면 끝!
- 실행해보기!

```
public void checkMatched()
{
    string firstCardImage = firstCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite.name;
    string secondCardImage = secondCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite.name;

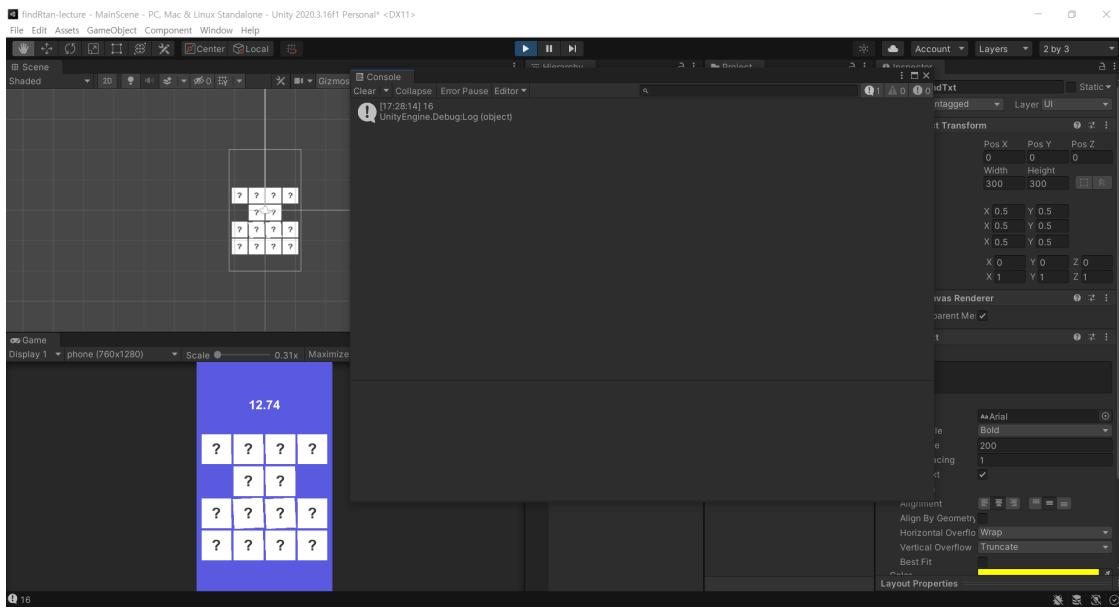
    if (firstCardImage == secondCardImage)
    {
        firstCard.GetComponent<card>().destroyCard();
        secondCard.GetComponent<card>().destroyCard();

        int cardsLeft = GameObject.Find("cards").transform.childCount;
        Debug.Log(cardsLeft);
    }
    else
    {
        firstCard.GetComponent<card>().closeCard();
        secondCard.GetComponent<card>().closeCard();
    }

    firstCard = null;
    secondCard = null;
}
```

3. 실행해보면, 지금 맞춘 두장을 포함해서 나오는 것을 알 수 있어요

- 즉, 2 가 나오면 마지막이라는 뜻!



4. 게임 끝내기!

```

public GameObject endTxt;

public void isMatched()
{
    string firstCardImage = firstCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite.name;
    string secondCardImage = secondCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite.name;

    if (firstCardImage == secondCardImage)
    {
        firstCard.GetComponent<card>().destroyCard();
        secondCard.GetComponent<card>().destroyCard();

        int cardsLeft = GameObject.Find("cards").transform.childCount;
        if (cardsLeft == 2)
        {
            endTxt.SetActive(true);
            Time.timeScale = 0.0f;
        }
    }
    else
    {
        firstCard.GetComponent<card>().closeCard();
        secondCard.GetComponent<card>().closeCard();
    }

    firstCard = null;
    secondCard = null;
}

```

5. 끝 버튼에 다시시작하기 버튼 붙이기!

- (1) 버튼 컴포넌트 추가하기
- (2) `endTxt.cs` 만들고, `endTxt`에 붙이기

```

using UnityEngine.SceneManagement;

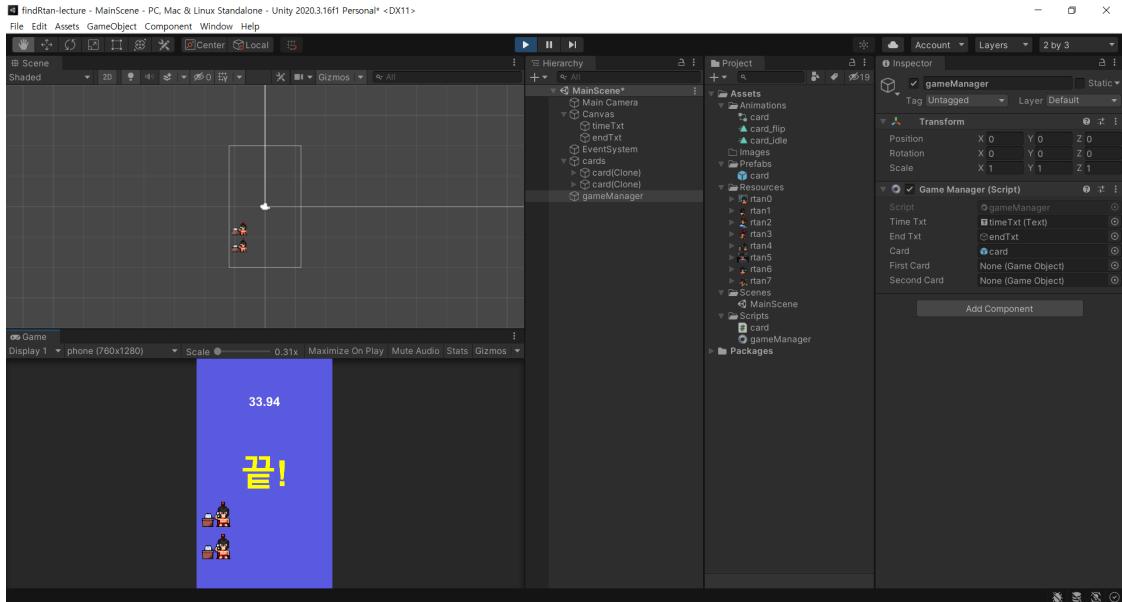
public void retryGame()
{
    SceneManager.LoadScene("MainScene");
}

```

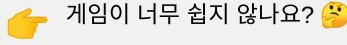
- (3) `gameManager.cs` start에서 `TimeScale`을 다시 되돌려놓는 것도 잊지말기

```
Time.timeScale = 1.0f;
```

6. 확인하기 ⇒ 잘 되네요!



▼ 2) 미세 조정하기



- 이렇게, 다 만들고 변수들을 조정해서 마지막으로 게임을 손 본답니다!
- `card.cs`에서 없어지거나 / 다시 뒤집히는 시간을 `0.5f`로 바꿔둘게요!

```
public void destroyCard()
{
    Invoke("destroyCardInvoke", 0.5f);
}

void destroyCardInvoke()
{
    Destroy(gameObject);
}

public void closeCard()
{
    Invoke("closeCardInvoke", 0.5f);
}

void closeCardInvoke()
{
    anim.SetBool("isOpen", false);
    transform.Find("back").gameObject.SetActive(true);
    transform.Find("front").gameObject.SetActive(false);
}
```

11. 숙제 - 30초가 지나면 게임 끝내기



그래요! 게임이 심장 끌깃한(?) 맛이 있어야죠..! 🎉

▼ 이렇게 되면 완성!

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/aad0e2b4-1575-4dea-9ed6-d0bf154baee6/르탄이_맞추기_숙제완성.mp4

▼ 힌트요정 - 🧸

[수정해야할 부분]

`gameManager.cs` 의 `Update()` 부분만 수정하면 됩니다!

→ 어떤 조건이 되면 아래 코드가 실행되면 되겠죠!

```
endTxt.SetActive(true);
Time.timeScale = 0.0f;
```

HW. 4주차 숙제 해설

▼ `gameManager.cs` 부분

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

using System.Linq;

public class gameManager : MonoBehaviour
{
    public Text timeTxt;
    float time = 0.0f;

    public GameObject endTxt;

    public GameObject card;

    public static gameManager I;

    public GameObject firstCard;
    public GameObject secondCard;

    void Awake()
    {
        I = this;
    }

    // Start is called before the first frame update
    void Start()
    {
        Time.timeScale = 1.0f;

        int[] rtans = { 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7 };

        rtans = rtans.OrderBy(item => Random.Range(-1.0f, 1.0f)).ToArray();

        for (int i = 0; i < 16; i++)
        {
            GameObject newCard = Instantiate(card);
            newCard.transform.parent = GameObject.Find("cards").transform;
```

```

        float x = (i / 4) * 1.4f - 2.1f;
        float y = (i % 4) * 1.4f - 3.0f;
        newCard.transform.position = new Vector3(x, y, 0);

        string rtanName = "rtan" + rtans[i].ToString();
        newCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite = Resources.Load<Sprite>(rtanName);
    }
}

// Update is called once per frame
void Update()
{
    time += Time.deltaTime;
    timeTxt.text = time.ToString("N2");

    if (time > 30.0f)
    {
        endTxt.SetActive(true);
        Time.timeScale = 0.0f;
    }
}

public void isMatched()
{
    string firstCardImage = firstCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite.name;
    string secondCardImage = secondCard.transform.Find("front").GetComponent<SpriteRenderer>().sprite.name;

    if (firstCardImage == secondCardImage)
    {
        firstCard.GetComponent<card>().destroyCard();
        secondCard.GetComponent<card>().destroyCard();

        int cardsLeft = GameObject.Find("cards").transform.childCount;
        if (cardsLeft == 2)
        {
            endTxt.SetActive(true);
            Time.timeScale = 0.0f;
        }
    }
    else
    {
        firstCard.GetComponent<card>().closeCard();
        secondCard.GetComponent<card>().closeCard();
    }

    firstCard = null;
    secondCard = null;
}
}

```

이전 주차

다음 주차

[스파르타코딩클럽] 게임개발 종합반 - 4주차

Copyright © TeamSparta All rights reserved.