



발표시작

목차소개

조윤진

최빈

박기도

이태훈

목차

프로젝트 소개

시연

개발 노트

마무리

▶ 메인화면



게임제목

게임 설명 버튼

터치 시 게임에 대한
설명이 적힌 팝업을 출력함

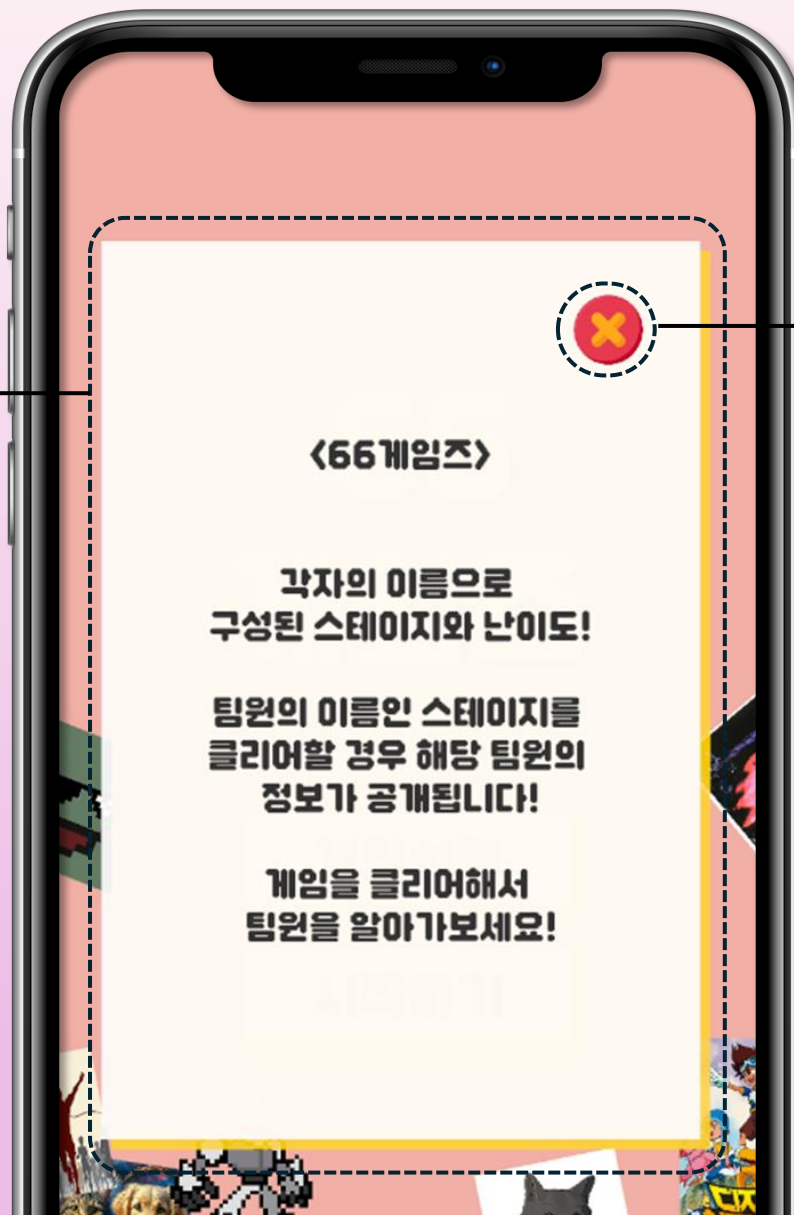
게임 시작 버튼

터치 시 게임 스테이지
선택 화면으로 이동

▶ 게임 설명 팝업

게임 설명 팝업

별개의 씬이 아니라
씬 내부의 오브젝트로 표시



팝업 종료 버튼

▶ 스테이지 선택 화면

돌아가기 버튼

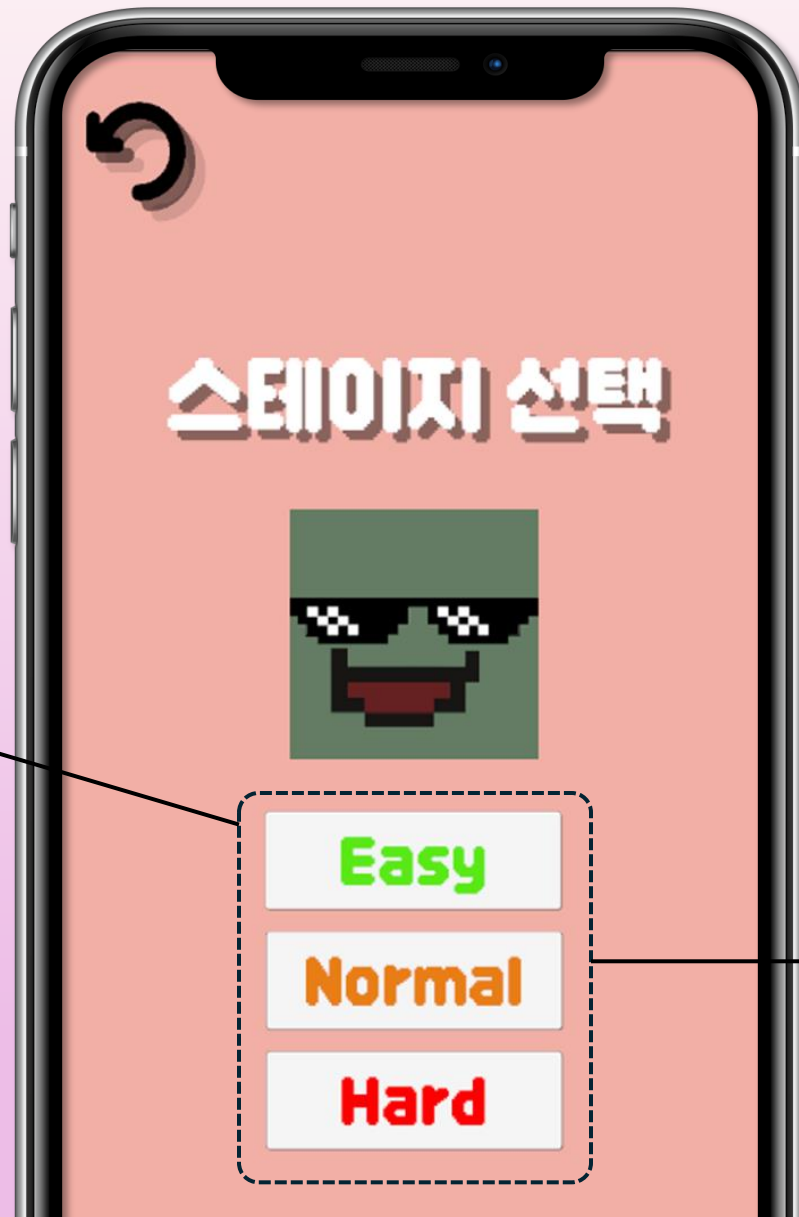
터치 시
이전의 화면으로 돌아감



스테이지 버튼

이름을 터치하면
해당 스테이지의
난이도 선택 UI가 노출

▶ 난이도 선택 화면



난이도 선택 버튼

난이도를 터치하면 해당
스테이지의 난이도가 설정되고
게임 레벨로 이동

난이도는 3개이며
클리어 이후에도
다음 난이도로 이동 가능!

▶ 기본 플레이 화면

돌아가기 버튼

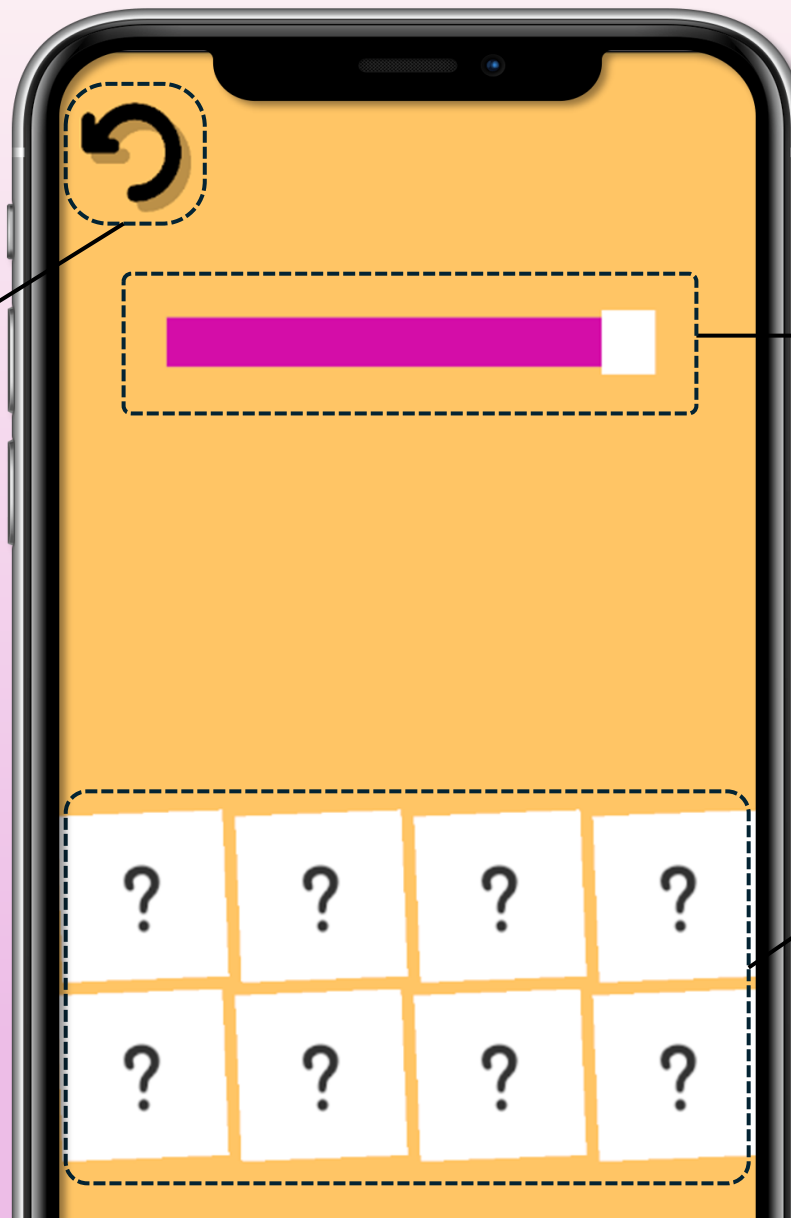
터치 시
스테이지 화면으로 돌아감

타이머 게이지

시간에 맞춰 줄어들며
전부 소진될 경우
게임 오버로 넘어감

카드

터치 시 카드를 뒤집고
동일한 이미지 한쌍을
고르면 해당 카드 2장은
사라짐 / 난이도가 증가
할 때마다 카드의 개수와
종류도 늘어남



▶ 게임오버 화면

재시도 버튼

터치 시 해당 레벨
스테이지를 처음부터 재시작함



별개의 씬 이 아니라
팝업으로 표시됨

▶ 게임성공 화면

갤러리 이동 버튼

선택한 스테이지의 사진 앨범 및
제작자의 정보(소개)가 작성된
썸으로 이동시켜주는 버튼

갤러리 보기

마지막 레벨 클리어
이후에는 출력되지 않음

레벨 이동 버튼

해당 스테이지의 다음
레벨로 이동하는 버튼

다음 레벨로

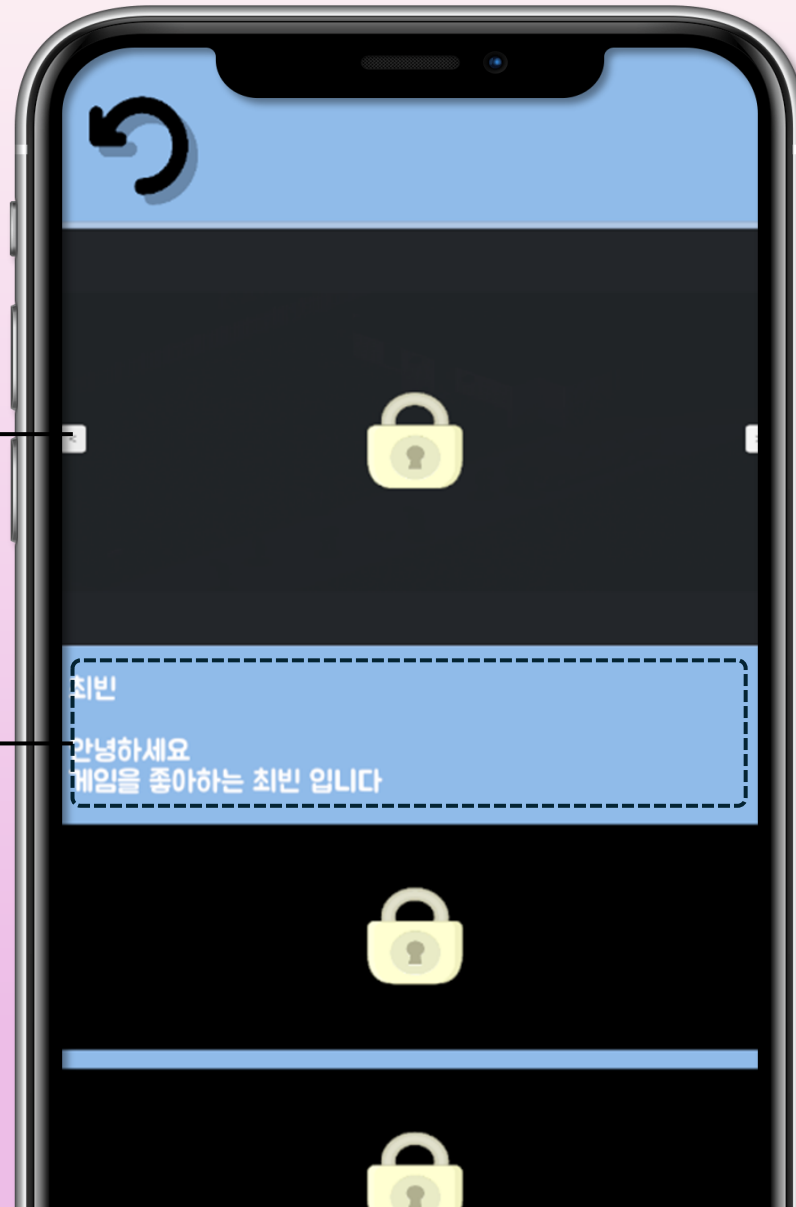
▶ 갤러리 화면

갤러리 문구

해당 스테이지에서 나왔던
이미지가 스크롤뷰로 출력

갤러리 문구

스테이지 제작자가 작성한
정보 및 소개가 출력



스테이지의 단계가 높아질수록
클리어 시 더 많은 정보가
(자물쇠가) 해금되는 형태

GAME PLAY!



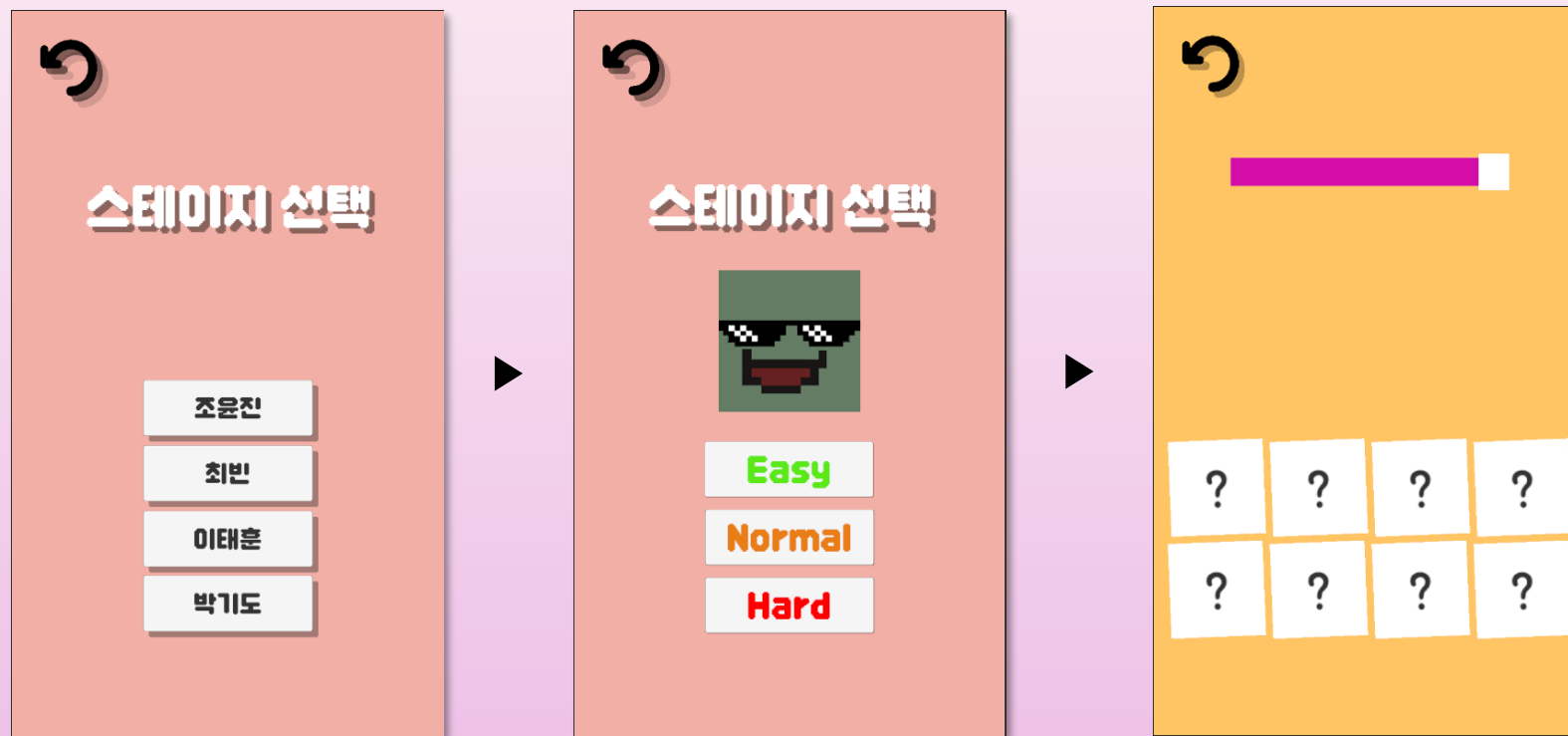
발표자의 시연을 감상해주세요!

66게임즈의 핵심 기능!

작업하면서 어려웠던 점?

트러블 슈팅은...

▶ 게임매니저 싱글톤



11

게임매니저를 이용해 'namescene'에서 선택한 이름과 난이도 정보에 따라 'mainscene'이 다르게 나오는 기능 구현

▶ 게임매니저 싱글톤

 GameManager.cs

```
public string Name = "";  
public int difficulty = 0;
```

Name/difficulty 변수 선언

Name: 팀원의 이니셜(B, GD, TH, YJ)
을 삽입

Difficulty: 0~2의 숫자를 삽입
(0: Easy, 1: Normal, 2: Hard)

* 각각 변수에 들어간 값에 따라
스테이지와 난이도가 변경

▶ 게임매니저 싱글톤



```
void Start()

int count = 0;
int[] arr = new int[(GameManager.Instance.difficulty + 4) * 2];

for (int i = 0; i < arr.Length; i += 2)
{
    arr[i] = count;
    arr[i + 1] = count;
    count++;
}
arr = arr.OrderBy(x => Random.Range(0f, 7f)).ToArray();

GameManager.Instance.cardCount = arr.Length;
```

난이도별 카드 개수를 다르게 하기 위해 arr의 크기를 difficulty 변수에 따라 다르게 설정.

arr 배열에 count번째 카드를 2개씩 삽입.

‘for문’의 i가 매 반복마다 2씩 증가, i가 증가할 때 arr[i]번째와 그 다음번의 arr[i+1]에 같은 숫자를 넣어 짝을 맞춤.

▶ 게임매니저 싱글톤



```
for (int i = 0; i < (GameManager.Instance.difficulty + 4) * 2; i++)
{
    GameObject go = Instantiate(card, this.transform);

    if (GameManager.Instance.difficulty == 0)
    {
        float x = (i % 4) * 1.4f - 2.1f;
        float y = (i / 4) * 1.4f - 3.0f;

        go.transform.position = new Vector2(x, y);
        go.GetComponent<Card>().Setting(arr[i]);
    }
}
```

for문을 통해 카드를 배치

Difficulty값에 따라 배치도를 변경

난이도별 각각 8장, 10장, 12장이 배치가 되는데, 대칭을 맞추어 재배포 설정.

▶ 게임매니저 싱글톤

 Card.cs

```
public void Setting(int idx)
{
    index = idx;
    frontImage.sprite = Resources.Load<Sprite>
    ($"{GameManager.Instance.Name}/Card{index}");
}
```

 Board.cs

```
go.GetComponent<Card>().Setting(arr[i]);
```

Setting 메서드 설정

카드의 이미지를 변경하는 메서드,
카드의 뒷면(sprite)이 네임값을
따라가도록 설정.

카드를 배치하고 몇번째 카드인지에
따라 이미지가 변경됨.

▶ 게임매니저 싱글톤

 GameManager.cs

```
public void isMatched()
{
    if (firstCard.index == secondCard.index)
    else
    {
        firstCard.CloseCard();
        secondCard.CloseCard();
    }

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

isMatched 로직

▶ 게임매니저 싱글톤

 GameManager.cs

```
if (cardCount == 0)

    if (GameManager.Instance.difficulty != 2)
    {
        FindObjectOfType<Canvas>().transform.GetChild(5).gameObject.SetActive(true);
        FindObjectOfType<Canvas>().transform.GetChild(4).gameObject.SetActive(true);
    }
    else if (GameManager.Instance.difficulty == 2)
    {
        FindObjectOfType<Canvas>().transform.GetChild(4).gameObject.SetActive(true);
    }
}
```

MainScene에 캔버스가 1개밖에 없기 때문에
FindObjectOfType<T>() 메소드를 통해
Canvas 오브젝트를 찾고,
n번째 자식 오브젝트를 활성화.

▶ 게임매니저 싱글톤



```
public void Galleryload()  
{  
    SceneManager.LoadScene  
    ("GalleryScene_{GameManager.Instance.Name}");  
}
```

```
GalleryScene_B  
GalleryScene_GD  
GalleryScene_TH  
GalleryScene_YJ
```

갤러리 씬을 로드하는 갤러리 버튼
스크립트에도 GameManager의
Name을 사용.

* 갤러리씬 이름 뒤에 이니셜을 두는
양식을 채택하여 불러오기 쉽게 설정.

▶ 게임매니저 싱글톤

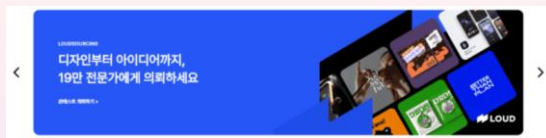
 GameManager.cs

```
int numPanelsToShow =  
4 + GameManager.Instance.difficulty;
```

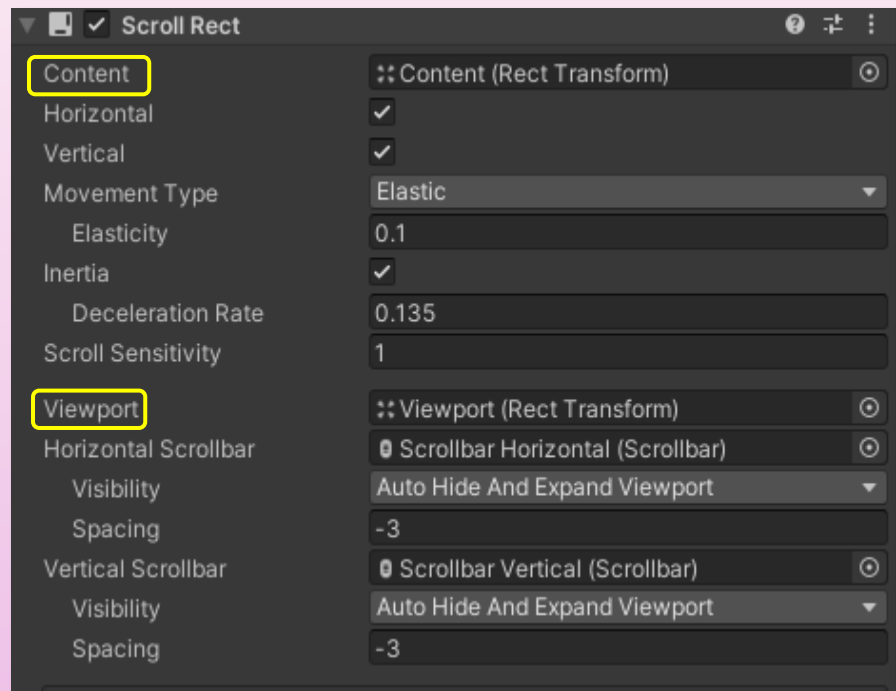
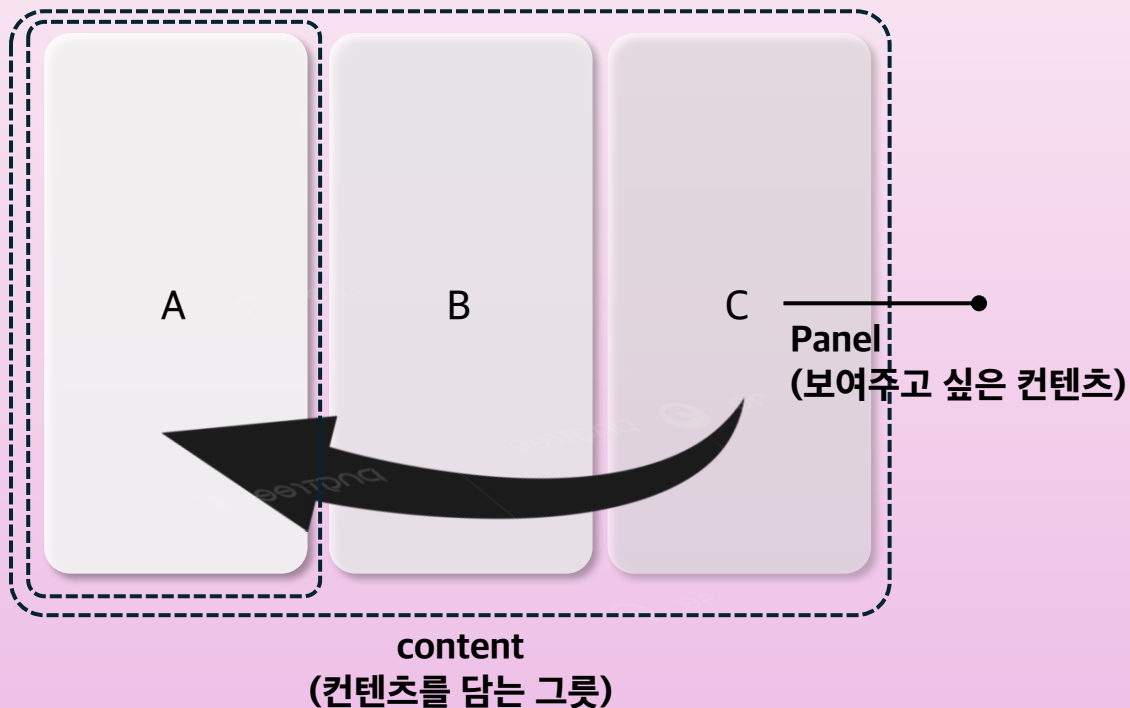
갤러리 씬에서 보여질 이미지와
텍스트 개수를 GameManager의
difficulty로 결정.

홈페이지에 자주 보이는 이벤트 페이지예요!

▶ 갤러리썬 Scroll View



View port
(보여지는 화면 영역)



이미지의 배열을 정해두고 스크롤을 내려서 다음 목차를 보여주는 스크롤 뷰 기능 구현

▶ 갤러리썬 Scroll View

Trouble shooting!

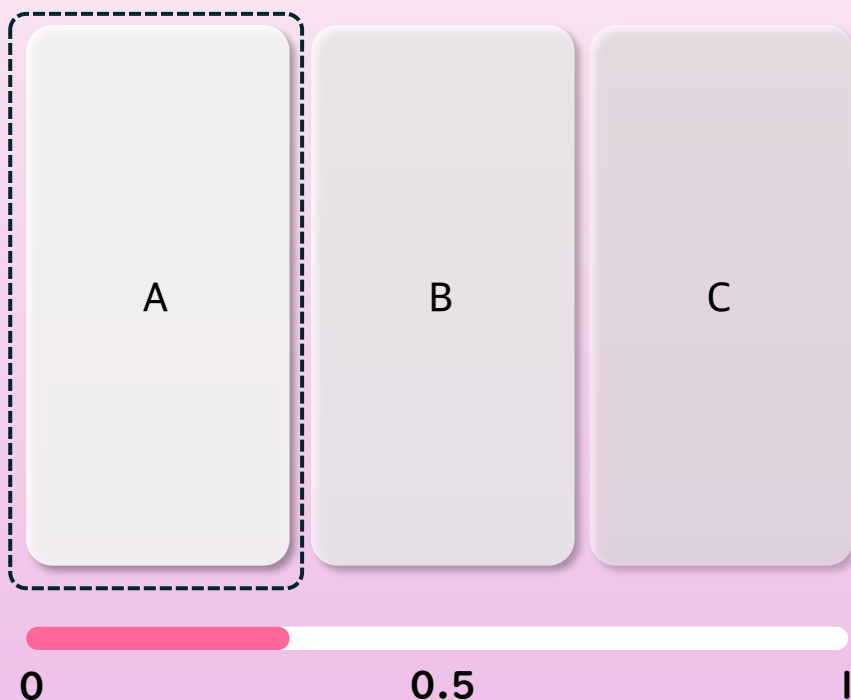


보이는 화면(view port)에 2~3개의 이미지가
동시에 노출되는 문제 발생

실제로 보여지는 건 View port
영역 안에 있는 모습!
즉, Panel(보여줄 이미지)을
Viewport 영역 안에 들어
올 수 있는 크기로,
Content는 panel을 모두 담을
수 있는 크기로 설정,

View port는 위치가 고정.
content의 위치를 직접 조정
하는 것이 아니라 scrollrect(실
제 기능)를 통해 scrollbar(기능
에 의한 시각적인 부분)를 움직
이면서 최종적으로 content의
위치를 변경.

▶ 갤러리썬 Scroll View



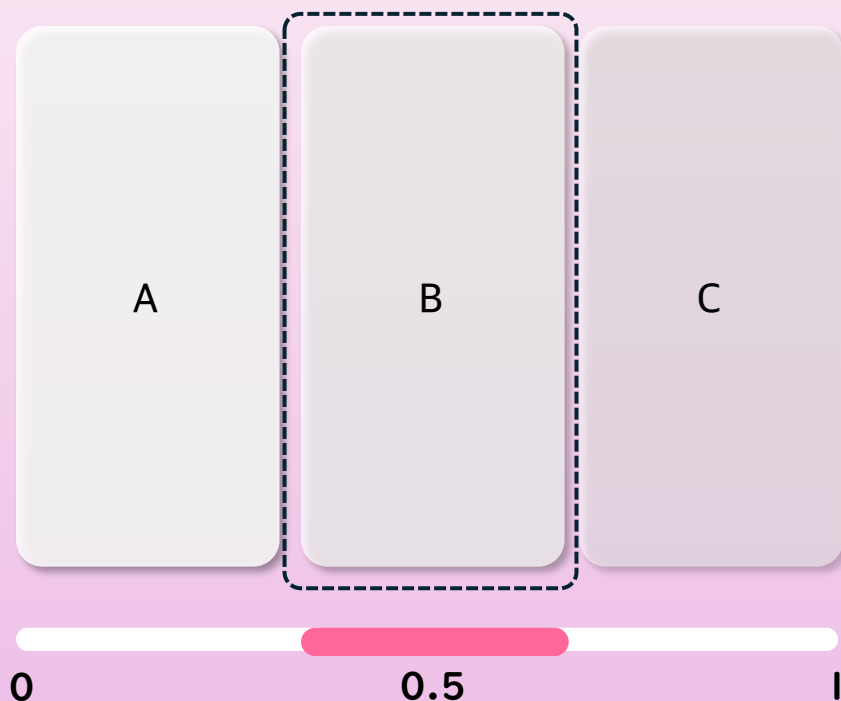
```
currentIndex++;
```

```
currentPosition = scrollRect.horizontalNormalizedPosition++;  
nextPosition = (float)index / (totalPanels - 1);
```

첫번째 패널(A)을 보여주려면?

```
Currentindex = 1;  
currentPosition = 0;  
nextPosition = 1 / 2 == 0.5;
```


▶ 갤러리썬 Scroll View



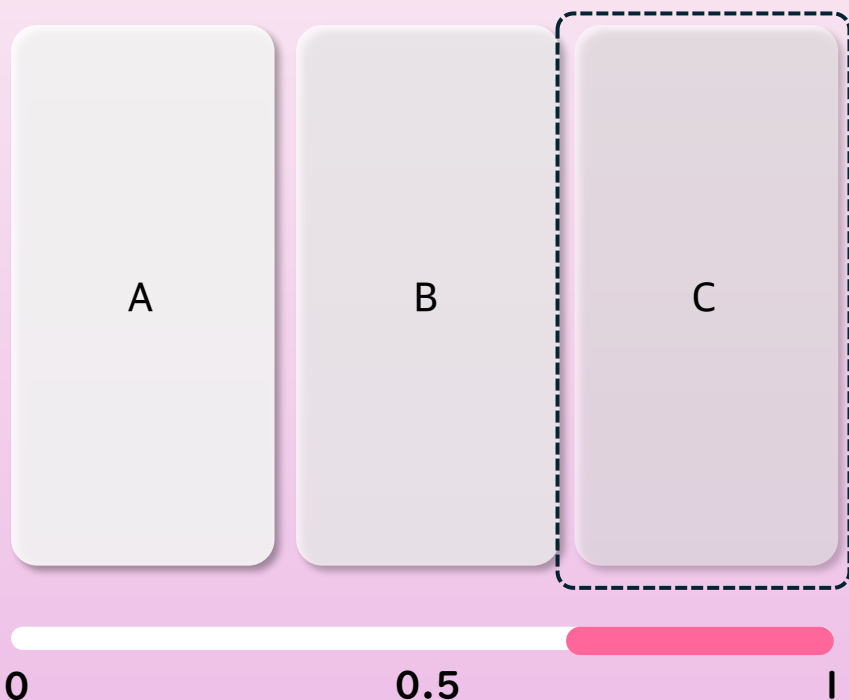
```
currentIndex++;
```

```
currentPosition = scrollRect.horizontalNormalizedPosition++;  
nextPosition = (float)index / (totalPanels - 1);
```

두번째 패널(B)을 보여주려면?

```
Currentindex = 0;  
currentPosition = 0;  
nextPosition = 0 / 2 == 0;
```

▶ 갤러리썬 Scroll View



```
currentIndex++;
```

```
currentPosition = scrollRect.horizontalNormalizedPosition++;  
nextPosition = (float)index / (totalPanels - 1);
```

세번째 패널(C)을 보여주려면?

```
Currentindex = 1;  
currentPosition = 0.5;  
nextPosition = 1;
```



조윤진

최빈

박기도

이태훈

 **감사합니다**