

Gem Pop

- Unity 모바일 포트폴리오 -

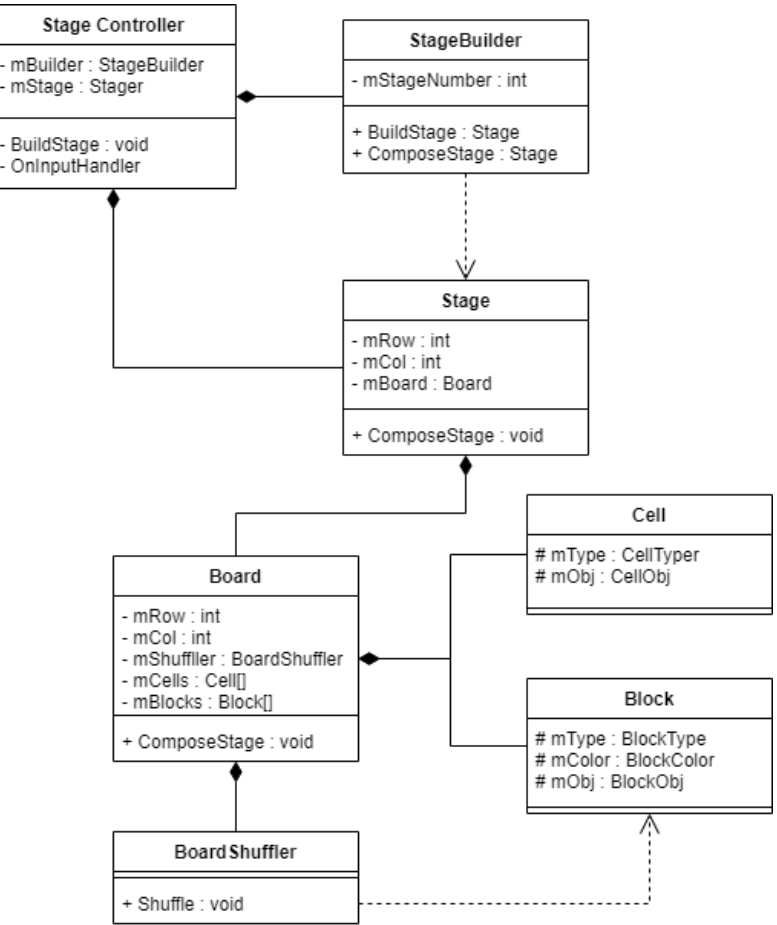
포트폴리오 링크

<https://github.com/Geonsu-Kim/GemPop>

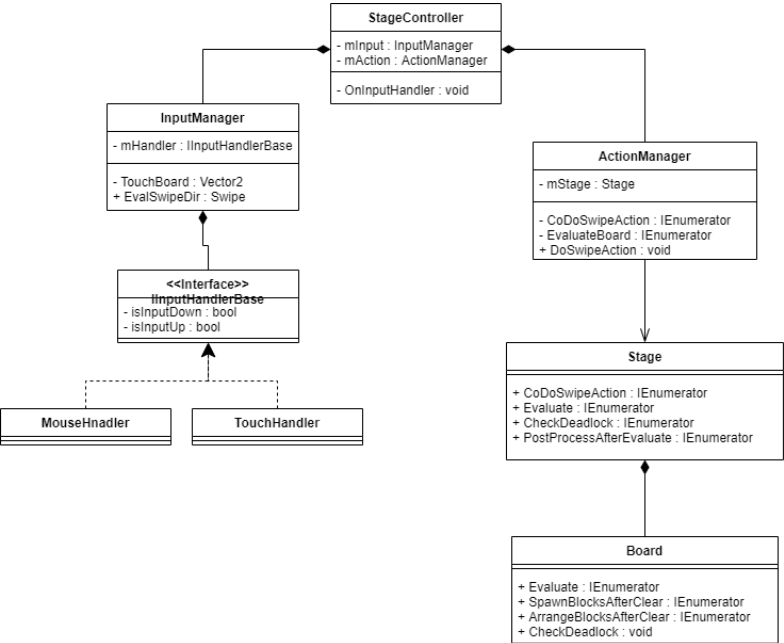
-Contents-

- 클래스 구조
- 시퀀스 구조
- 주요 기능 설명
- 데이터 관리
- 최적화 작업

스테이지 초기화 구조

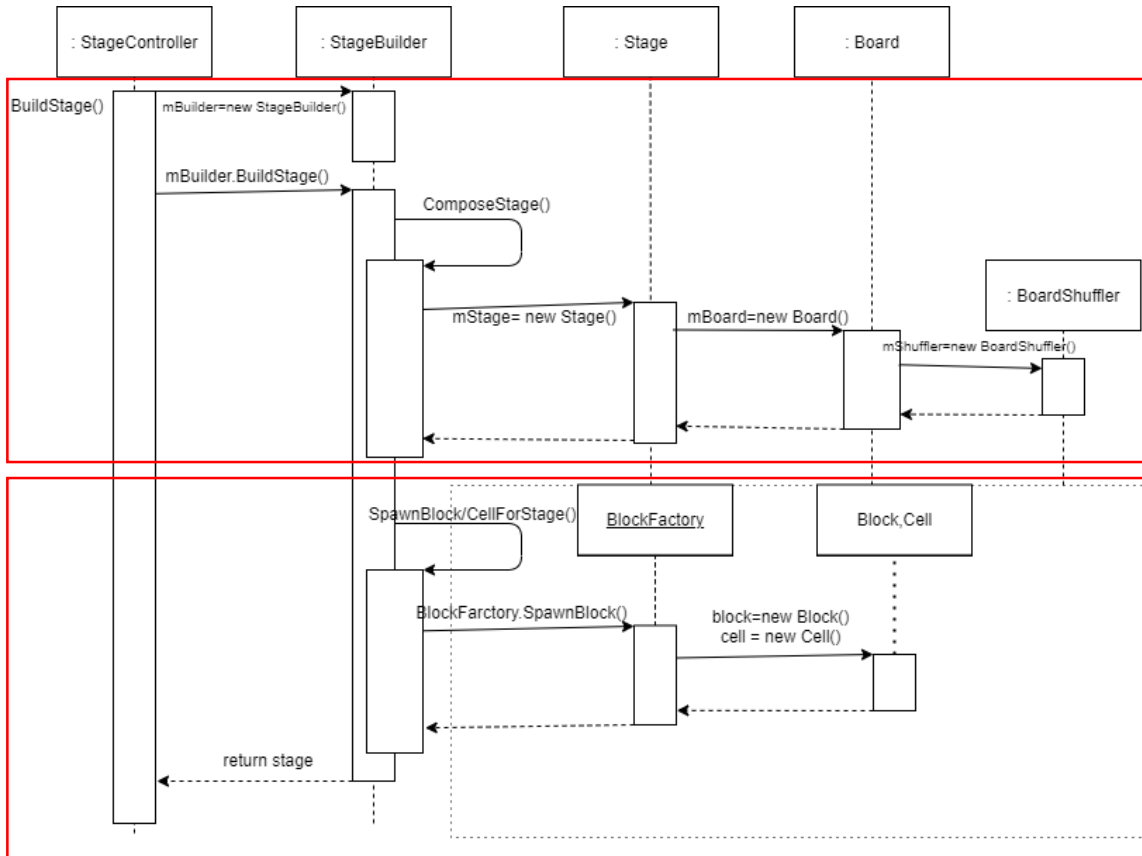


스테이지 진행 구조



시퀀스 구조도

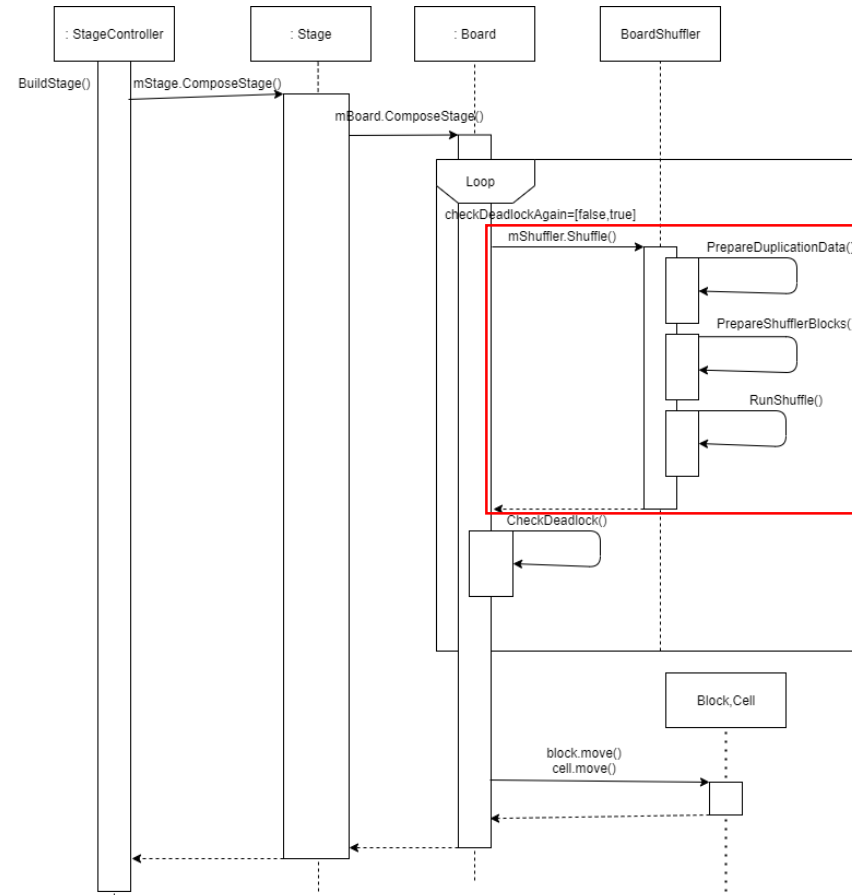
스테이지 생성 시퀀스



보드판을 생성한다

생성한 보드판 위에
블록을 배치한다

스테이지 생성 시퀀스(계속)



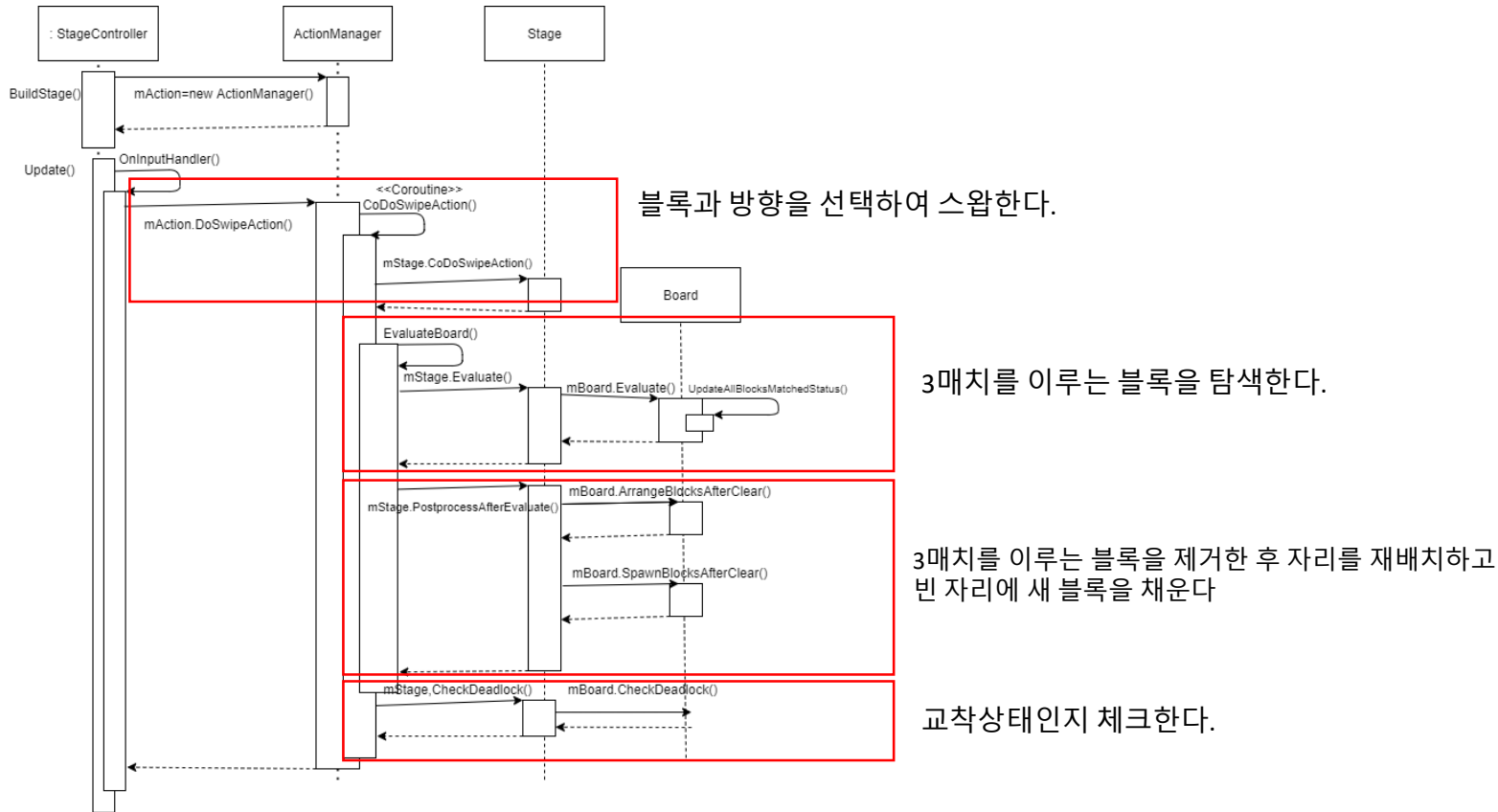
배치된 블록이 3매치를 이루는 경우를 방지하기 위하여 재배치를 한다.

블록을 재배치했을 때 보드의 교착상태 검사를 실시한다.
교착상태로 판정될 경우 재배치 과정을 다시 거친다.

블록 오브젝트들을 실제로 배치한다.

시퀀스 구조도

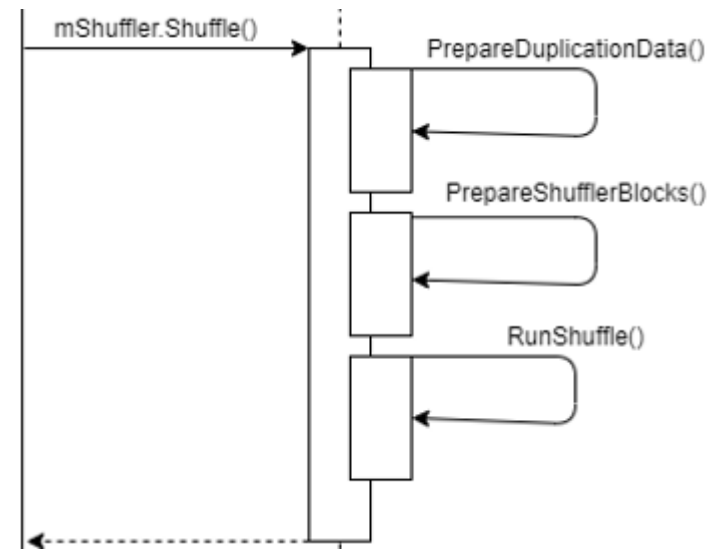
스테이지 진행 시퀀스



주요 기능 설명

블록 배치(Shuffle)

보드 위의 블록들을 새로 배치할 때 3매치가 일어나지 않도록 한다.



PrepareDuplicationData

:인접한 블록 중 색깔이 같은 블록의 개수를 초기화한다.

PrepareShufflerBlocks

:보드 위의 블록들을 회수하여 SortedList에 무작위로 정렬한다

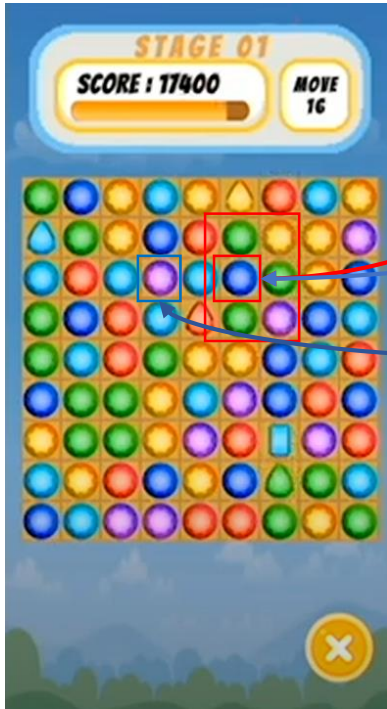
RunShuffle

:SortedList의 블록들을 꺼내 보드 위에 재배치한다.

주요 기능 설명

블록 배치(Shuffle)

: RunShuffle 작동 구조



SortedList



Queue



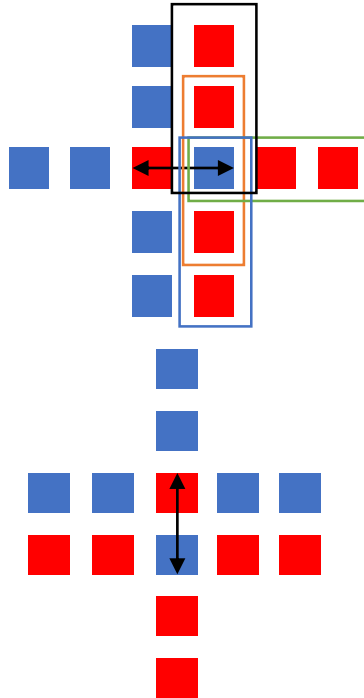
1. 블록을 꺼낼 자료구조를 선택한다.
(Queue에 블록이 있을 경우 Queue를 선택, 없다면 SortedList를 선택)
2. 자료구조에서 꺼낸 블록과 배치할 자리 주변의 블록을 비교한다.
3. 3매치인 경우 꺼낸 블록을 Queue에 넣고 1번으로 돌아간다.
이 때 블록을 꺼낼 자료구조는 Queue로 선택한다.
4. 3매치가 아닌 경우 그대로 자리에 배치한다.

*자료구조에 모두 같은 색깔의 블록만 있고 배치해야 할 자리가 3매치를 이루는 경우
->자료구조에서 꺼낸 블록의 색깔을 바꿔 배치한다.

주요 기능 설명

교착상태 검사

: CheckDeadlock()



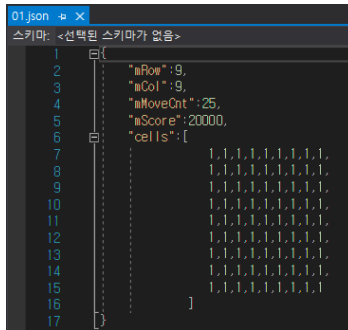
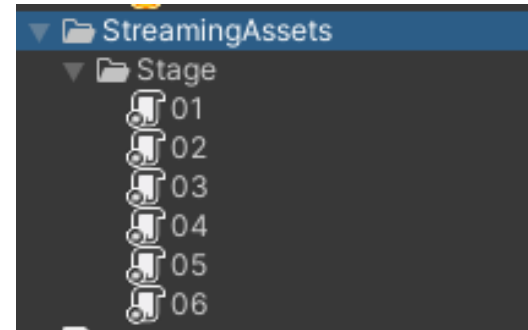
두 블록을 좌우로 스왑했을 때 3매치가 이루어지는 각각의 경우는 최대 4가지이므로 이 경우들을 모두 검사한다.

상하로 스왑하는 경우에도 그대로 적용한다.

데이터 관리

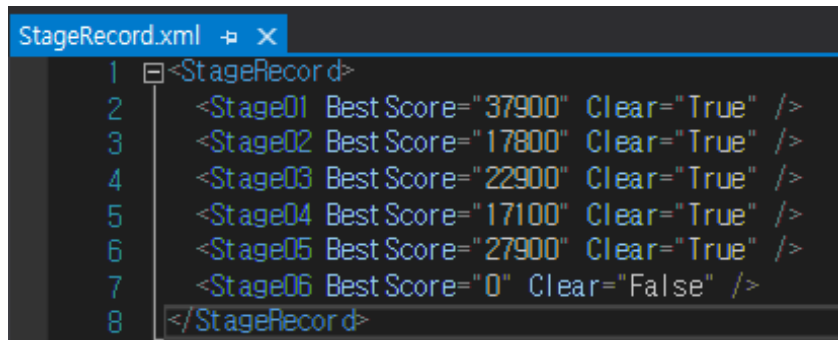
스테이지 정보

StreamingAsset(읽기만 가능)에 JSON 파일로 저장 -> 파싱으로 스테이지 정보를 불러온다.



스테이지 클리어 정보

PersistentDataPath에 Xml 파일로 클리어 여부와 최고득점 저장



최적화 작업

오브젝트 풀링

오브젝트 풀에 충분한 수의 블록 객체들을 미리 생성한 후 제거 시 유니티 엔진 내 Destroy() 함수 대신 SetActive()로 비활성화한 뒤 다시 활성화하는 방식으로 재생성한다.

스태이지 초기화 때 미리 블록들을 생성

```
void InstantiateBlockCell()
{
    for (int i = 0; i < 81; i++)
    {
        GameObject newCell = Instantiate(cellObj);
        newCell.SetActive(false);
        newCell.name = nameCell;
        Pool_Cell.Add(newCell);
    }
    for (int i = 0; i < 200; i++)
    {
        GameObject newBlock = Instantiate(blockObj);
        newBlock.SetActive(false);
        newBlock.name = nameBlock;
        Pool_Block.Add(newBlock);
    }
}
```

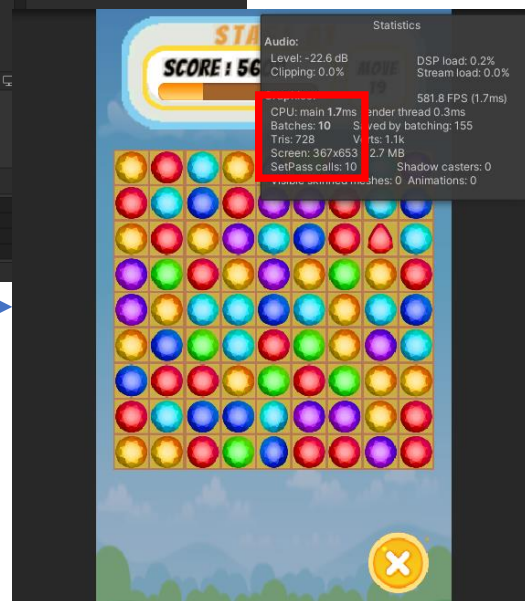
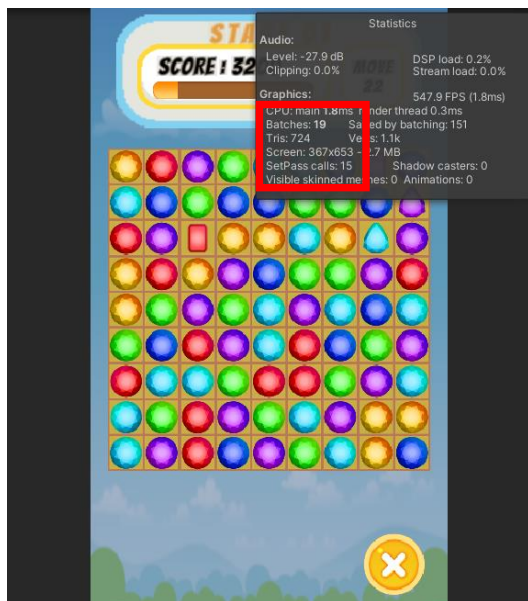
블록 생성 대신 활성화, 단 풀에 블록이 부족할 경우 새로 생성한다.

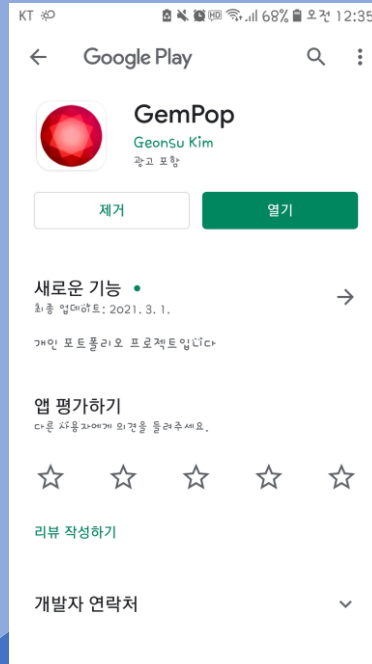
```
public GameObject GetBlock()
{
    for (int i = 0; i < Pool_Block.Count; i++)
    {
        if (!Pool_Block[i].activeSelf)
        {
            return Pool_Block[i];
        }
        else
        {
            if (i == Pool_Block.Count - 1)
            {
                InstantiateNewBlock();
                return Pool_Block[i + 1];
            }
        }
    }
    return null;
}
```

최적화 작업

드로우콜 최소화

블록, UI, 이펙트 스프라이트 아틀라스를 만들어 배치와 셋 패스 감소





‘Gem Pop’ 은 현재 구글 플레이에 출시되어 있습니다.

Thank You For Watching