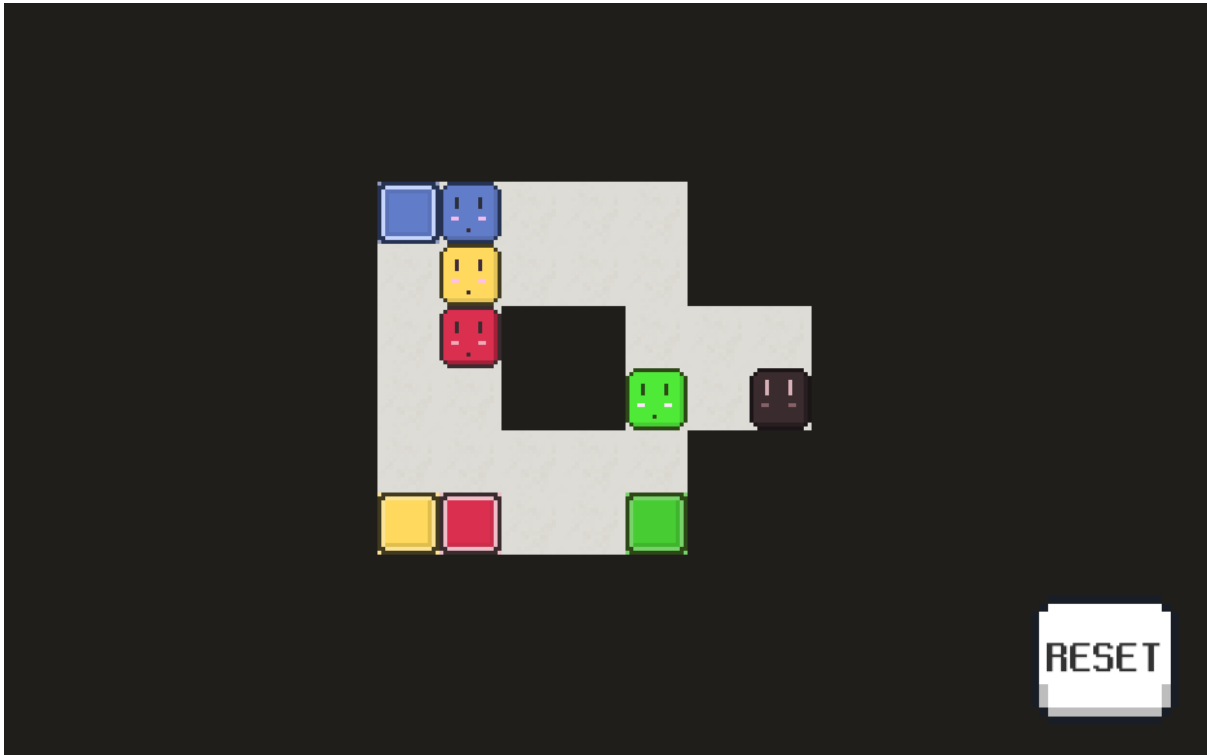


Push Slime GDD 문서



분류	내용
작성 일자	2024-11-05
작성자	류승재

수정 일자	수정 내역

0. Index

1. 게임 소개	3
1.1 게임 개요	3
1.2 인게임 스크린샷 예시 (데모 버전)	3
2. 게임 스토리	4
3. 게임 메카닉스.....	5
3.1 게임 내 용어 정리.....	5
3.2 플레이어 조작	6
3.3 오브젝트 이동 기믹	7
3.4 하우스 오브젝트	12
3.5 데모 레벨 소개.....	13
4. 기타.....	15
4.1 유사 게임 조사.....	15

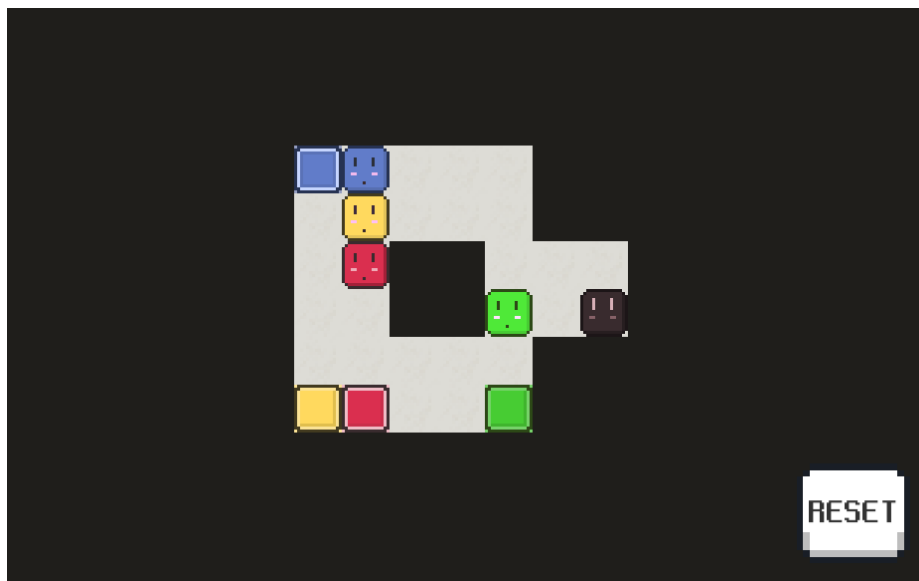
1. 게임 소개

- ▶ 본 게임은 타일맵 기반 이동 방식을 구현하던 도중 떠오른 아이디어를 바탕으로 직접 구현해본 소코반 형식의 캐주얼 퍼즐 게임입니다.
- ▶ 게임의 데모 버전을 구현하며 떠오른 생각들과 아이디어를 정리한 문서입니다.
- ▶ <https://rsj0908.itch.io/push-slime> (데모 게임 실행 링크)

1.1 게임 개요

분류	내용
게임 제목	Push Slime
게임 장르	소코반 장르 퍼즐 게임
사용 엔진	유니티
그래픽	2D 도트 그래픽, 탑뷰
조작키	WASD 기반 키보드 조작, R키를 통한 스테이지 리셋

1.2 인게임 스크린샷 예시 (데모 버전)



데모 버전으로 구현한 Push Slime의 예시 이미지입니다.

2. 게임 스토리



▶ “다크 브라운 슬라임”은 다양한 슬라임들이 한데 모여 살고 있는 슬라임 가족의 장남입니다. 다크 브라운 슬라임은 한 가지 고민이 있었는데, 그것은 자신의 동생들이 하나 같이 말을 듣지 않는 장난꾸러기였다는 것입니다.

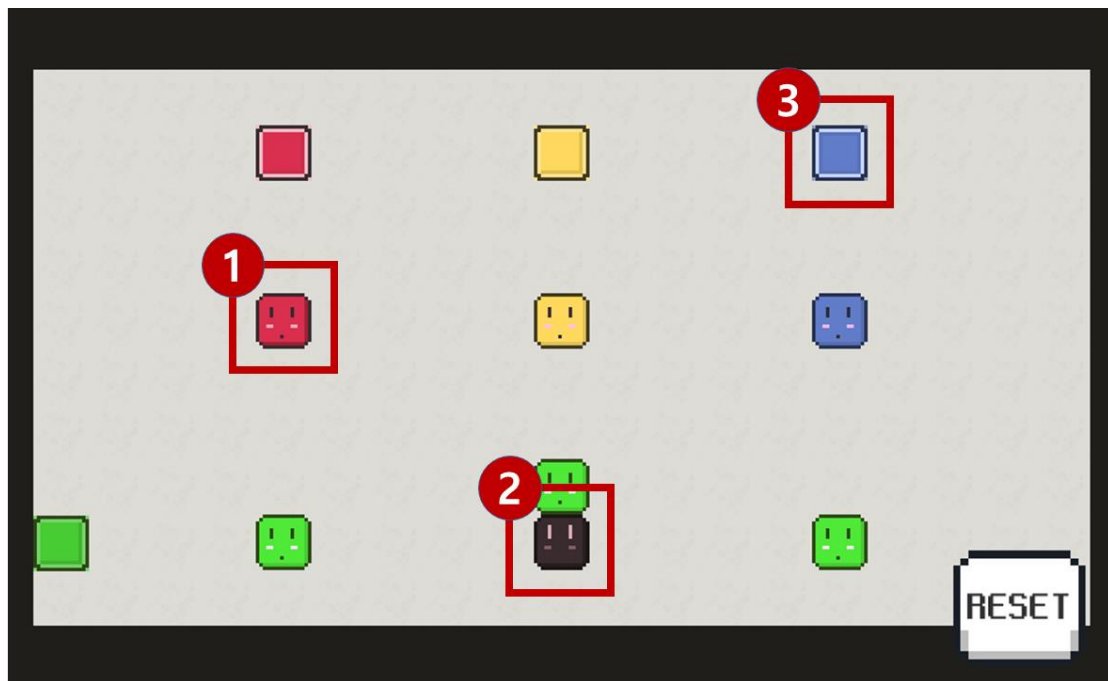
▶ 잘 때가 다 되어도 밖을 돌아다니는 말썽쟁이 동생 슬라임들을 재우기 위해, 슬라임 가족의 장남, 다크 브라운 슬라임은 동생 슬라임들을 집으로 돌려 보내기 위해 노력합니다.

▶ 플레이어는 다크 브라운 슬라임이 되어 각양각색 색다른 매력(기믹)을 가진 다양한 슬라임들을 각자 살고 있는 올바른 집으로 돌려보내는 것을 목표로 게임을 진행하게 됩니다.

3. 게임 메카닉스

▶ 본 게임의 목표는 “플레이어 슬라임”을 이동시켜 맵 상 존재하는 모든 “슬라임 오브젝트”를 각 슬라임 오브젝트와 색이 일치하는 “하우스 오브젝트”로 밀어 넣는 것입니다.

3.1 게임 내 용어 정리



[1] 슬라임 오브젝트

▶ 플레이어가 조작을 통해 직, 간접적으로 이동시키는 것이 가능한 오브젝트입니다. 각 슬라임이 보유한 ‘색’과 외형적 특징을 통해 구분할 수 있으며, 각 슬라임은 고유한 이동 기믹을 가지고 있습니다.

[2] 플레이어 슬라임

▶ 플레이어가 직접 조작이 가능한 다크 브라운 색의 슬라임입니다. 조작을 통해 “슬라임 오브젝트”와 주도적인 상호작용이 가능합니다.

[3] 하우스 오브젝트

▶ 스테이지 클리어를 위해서 모든 슬라임 오브젝트가 위치해야 할 지정 포인트입니다. 슬라임 오브젝트와 각각 일치하는 색의 하우스 오브젝트가 존재하고, 모든 슬라임 오브젝트가 올바른 하우스 오브젝트의 위치에 위치할 때 스테이지가 클리어됩니다.

3.2 플레이어 조작

- ▶ 데모 버전은 PC 버전, 이동키와 스테이지 리셋 버튼만 구현되어 있습니다.

■ [PC 버전]

조작	Key 할당 혹은 조작 방법
이동키	W A S D
스테이지 리셋	R / 리셋 버튼 클릭
환경설정	ESC / 환경설정 버튼 클릭
되돌리기	Z / 돌아가기 버튼 클릭

- ▶ 마우스 조작 지원 (버튼 클릭만 가능)
- ▶ 되돌리기는 마지막으로 한 조작을 무효로 하고 뒤로 돌림

■ [모바일 버전]

조작	Key 할당 혹은 방법
이동키	슬라이드 / 버튼 이동 (변경 가능)
스테이지 리셋	버튼 클릭
환경설정	버튼 클릭
되돌리기	버튼 클릭

- ▶ 슬라이드와 버튼 조작은 환경 설정에서 편한 방법으로 유저가 변경 가능

3.3 오브젝트 이동 기믹

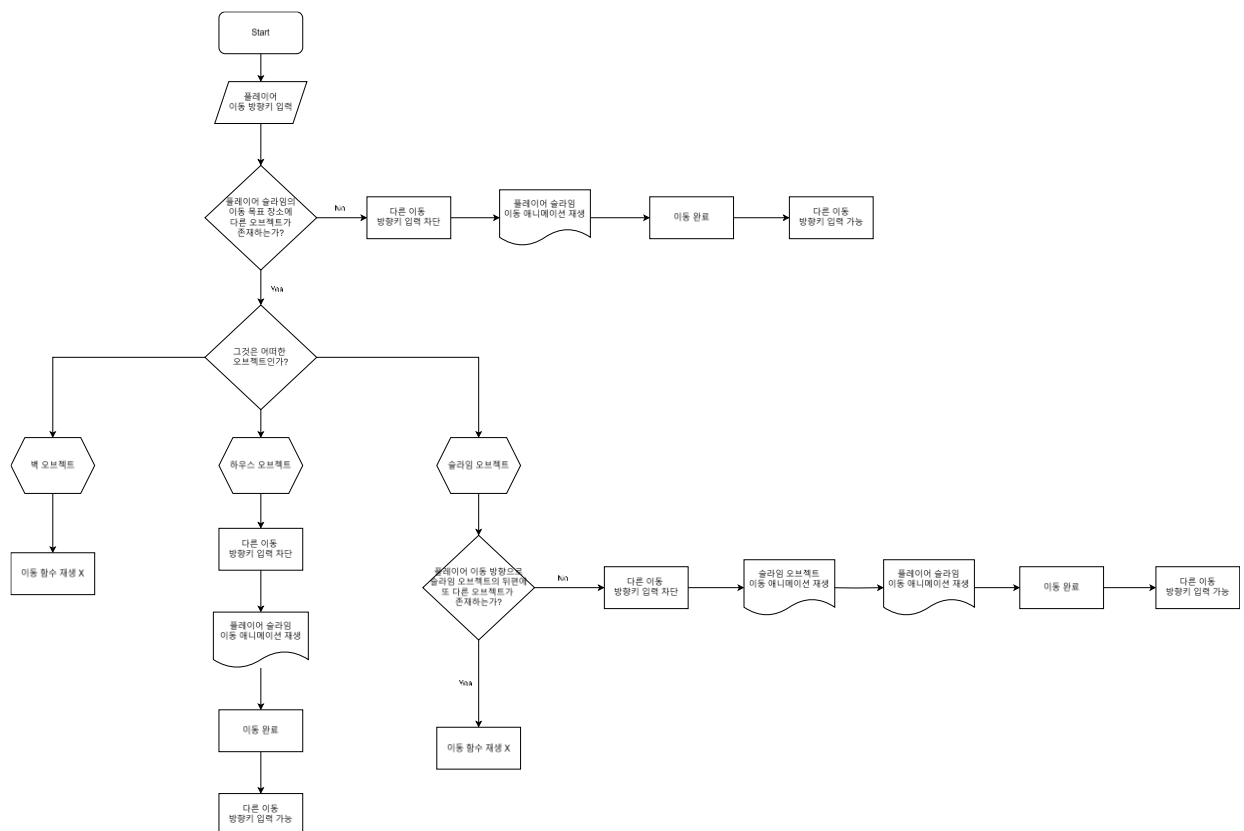
▶ 플레이어 슬라임, 슬라임 오브젝트의 이동 기믹과 알고리즘 등을 기록한 파트입니다.

3.3.1 플레이어 슬라임 이동 알고리즘

[1] 플레이어가 이동 방향키를 눌렀을 때의 기본적인 알고리즘 플로우차트

< <https://www.notion.so/Push-Slime-6bfb4b5057b4d9e8dc0c9c26513e1ea> >

(자세한 플로우차트는는 노션 페이지 - 알고리즘 설계 파트에서 확인해주세요.)



■ [요약 설명] :

플레이어 이동 방향에 위치한 다른 오브젝트를 RayCast를 통해 검사하고, 오브젝트가 없다면 이동 함수를 재생합니다. (이동 도중에는 다른 이동 입력을 차단합니다.)

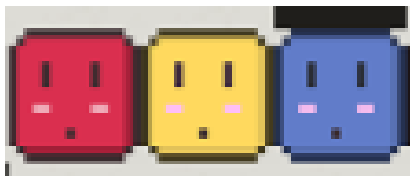
만약 다른 오브젝트가 있다면 그것이 어떤 오브젝트인가에 따라서 각각 다른 결과를 출력합니다.

3.32 슬라임 오브젝트 기믹

- ▶ 현재 데모 버전에서는 노말 슬라임, 말랑 슬라임만이 구현되어 있습니다.
- ▶ 슬라임이 가지는 세부 데이터를 정리했습니다.

index	Field	Type	Detail
1	Index	Int	슬라임 고유 번호, 인덱스
2	FileName	String	슬라임의 파일 네임
3	ColorCode	String	슬라임의 대표 컬러 코드
4	Detail	String	슬라임 명칭 혹은 세부 설명

[1] 노말 슬라임

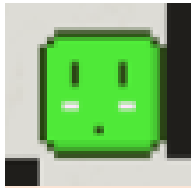


index	Name	ColorCode	Detail
1	NormalR	#D52F51	노말 슬라임 (빨강)
2	NormalY	#FED867	노말 슬라임 (노랑)
3	NormalB	#6980D2	노말 슬라임 (파랑)

- ▶ 노말 슬라임은 가장 일반적인 형식의 슬라임 오브젝트로 아래와 같은 규칙을 가지고 있습니다.

- [1] 플레이어 슬라임을 통해 '밀기'만 가능합니다.
- [2] 노말 슬라임을 밀 경우, 1타일 뒤로 이동합니다.
- [3] 노말 슬라임의 위치를 기준으로 밀고자 하는 방향 1타일 뒤에 슬라임 오브젝트나 벽이 있다면 밀 수 없습니다.

[2] 말랑 슬라임



index	Name	ColorCode	Detail
4	Malang	#D52F51	초록색 말랑 슬라임

▶ 노말 슬라임은 가장 일반적인 형식의 슬라임 오브젝트로 아래와 같은 규칙을 가지고 있습니다.

[1] 플레이어 슬라임을 통해 '밀기'만 가능합니다.

[2] 말랑 슬라임을 밀 경우, 다른 슬라임 오브젝트 혹은 벽 오브젝트를 만나기 전까지 계속해서 이동합니다.

[3] 말랑 슬라임의 위치를 기준으로 밀고자 하는 방향 1타일 뒤에 슬라임 오브젝트나 벽이 있다면 밀 수 없습니다.

```
private IEnumerator PushGreenSlime(Vector3 direction, GameObject Slime)
{
    float elapsedTime = 0f;

    GreenPushHit = Physics2D.RaycastAll(Slime.transform.position + direction, direction, 100f);

    for (int i = 0; i < GreenPushHit.Length; i++)
    {
        if (GreenPushHit[i].collider.tag == "Wall")
        {
            Border = GreenPushHit[i].transform.GetComponent<Tilemap>();
            x = Border.WorldToLocal(GreenPushHit[i].point).x;
            y = Border.WorldToLocal(GreenPushHit[i].point).y;
            break;
        }
        else if (GreenPushHit[i].collider.tag == "RSlime" || GreenPushHit[i].collider.tag == "GSlime"
            || GreenPushHit[i].collider.tag == "BSlime" || GreenPushHit[i].collider.tag == "YSlime")
        {
            x = GreenPushHit[i].collider.transform.position.x;
            y = GreenPushHit[i].collider.transform.position.y;

            if(direction.y != 0)
            {
                y -= direction.y / 2;
            }
            else if(direction.x != 0)
            {
                x -= direction.x / 2;
            }

            break;
        }
    }
}
```

▶ 경로 상의 다른 오브젝트를 검색하고(중간에 막히도록), 도착 위치를 결정하는 함수.

[3] 새침데기 슬라임



index	Name	ColorCode	Detail
5	Tsundere	#E491B9	핑크색 새침데기 슬라임

▶ 새침데기 슬라임은 부끄러움이 많은 슬라임이라는 설정으로 아래와 같은 규칙을 가지고 있습니다.

[1] 플레이어 슬라임이 이동하는 방향 일직선에 새침데기 슬라임이 있을 경우 플레이어 반대 방향 (단, 플레이어 슬라임이 새침데기 슬라임 방향으로 이동할 때만)으로 1타일 물러납니다.

[2] 물러나는 방향으로 1타일 뒤에 슬라임 오브젝트나 벽이 있다면 밀 수 없습니다.

[4] 찹쌀떡 슬라임



index	Name	ColorCode	Detail
6	Mocchi	#EBCAA1	아이보리색 찹쌀떡 슬라임

▶ 찹쌀떡 슬라임은 다른 슬라임과 꼭 붙어 있으려는 습성이 있는 슬라임으로 아래와 같은 규칙을 가지고 있습니다.

[1] 플레이어 슬라임을 통해 '밀기'만 가능. 슬라임을 밀면 1타일 뒤로 이동함.

[2] 밀고자 하는 방향으로 1타일 앞에 벽이나 다른 슬라임이 있을 경우 밀 수 없음.

[3] 찹쌀떡 슬라임은 1타일 내에 다른 슬라임이 있다면 (상, 하, 좌, 우 방향으로 / 대각선 X) 서로 붙어 하나의 오브젝트로 바뀌며, 그 상태를 "찹쌀떡 상태"라고 칭합니다.

[4] 찹쌀떡 상태가 된 슬라임은 아래와 같은 규칙을 추가로 가집니다.

[4-1] 찹쌀떡 슬라임은 찹쌀떡 상태가 된 상태라도 1타일 내에 슬라임이 있다면 추가로 다른 슬라임을 찹쌀떡 상태로 만들 수 있습니다. (최대 상,하,좌,우 4개의 슬라임 오브젝트를 찹쌀떡 상태로 만들 수 있습니다.)

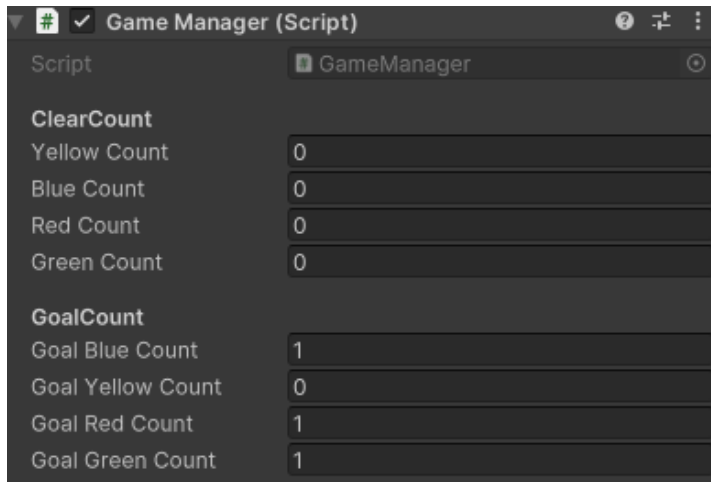
[4-2] 찹쌀떡 슬라임의 상,하,좌,우 1타일 내에 있을 슬라임만 찹쌀떡 상태가 될 수 있습니다. 찹쌀떡 상태인 찹쌀떡 슬라임을 제외한 슬라임은 다른 슬라임을 찹쌀떡 상태로 만들 수 없습니다.

[4-3] 찹쌀떡 상태인 슬라임은 특별한 경우 (특수 기믹, 맵 리셋 등) 을 제외하면 절대로 분리할 수 없습니다. 특수 기믹의 경우는 미기획 상태입니다.

[4-4] 찹쌀떡 상태인 슬라임은 노말 슬라임의 규칙을 가지며, 각 슬라임들의 고유한 기믹이 무효화됩니다.

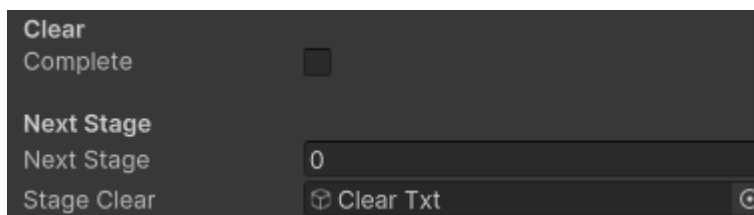
[4-5] 플레이어 슬라임과는 찹쌀떡 상태가 될 수 없습니다.

3.4 하우스 오브젝트



index	Field	Type	Detail
1	YellowCount	int	목적지에 도착한 노말 슬라임(파랑) 개수
2	BlueCount	int	목적지에 도착한 노말 슬라임(파랑) 개수
3	RedCount	int	목적지에 도착한 노말 슬라임(빨강) 개수
4	GreenCount	Int	목적지에 도착한 말랑 슬라임 개수
5	GoalYellowCount	Int	목표 YellowCount 수
6	GoalBlueCount	Int	목표 BlueCount 수
7	GoalRedCount	Int	목표 RedCount 수
8	GoalGreenCount	int	목표 GreenCount 수

▶ IF 모든 목표 Count == Goal Count 일 경우, Complete(Bool) 값을 변화시킨다.



▶ IF Complete == True 라면, Stage Clear 텍스트를 띄우고 Next Stage(스테이지 넘버)를 통해 다음 씬으로 이동한다.

3.5 데모 레벨 소개

▶ 데모로 제작한 맵(레벨)을 소개하는 파트입니다.

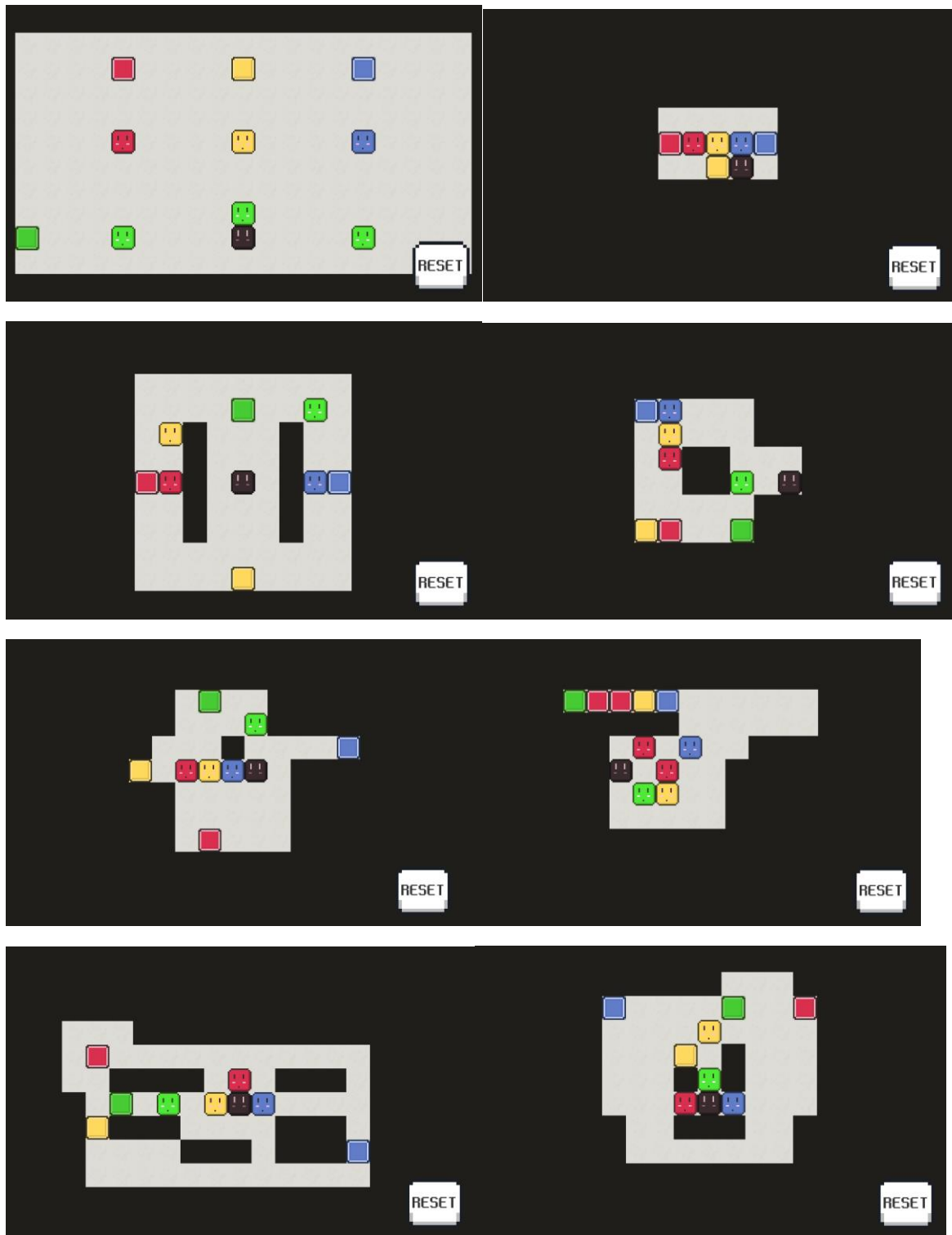
index	Field	Type	Detail
1	Index	Int	테이블 인덱스
2	LevelCode	String	레벨 ID 코드 (스테이지 컨셉 + 번호)
3	LevelName	String	레벨의 이름
4	Difficulty	int	레벨의 난이도 (0 <= Difficulty <= 5)

▶ 데이터 스키마

index	LevelCode	LevelName	Difficulty
1	A000	프롤로그	0
2	A001	신호등	1
3	A002	길막기	1
4	A003	돌아가는 길	1
5	A004	말랑말랑	1
6	A005	한줄서기	1
7	A006	순서가 중요해	1
8	A007	리버스 빨강 파랑	1

▶ 데이터 테이블

■ 세부 레벨 이미지



4. 기타

▶ 게임 기획 외 기타 내용들을 담았습니다.

4.1 유사 게임 조사

(5) 기타 – 유사 게임 사례 분석1 : BaBa Is You



[게임 제목] : Baba is you

[게임 가격] : ₩ 15,500

[플레이 타임] : 20 ~ 40 시간

[장르] : 퍼즐

[난이도] : 높은 편

[특이한 점] : 레벨을 직접 커스텀할 수 있는 기능을 제공함.

(5) 기타 – 유사 게임 사례 분석2 : PIPE PUSH PARADISE



[게임 제목] : PIPE PUSH PARADISE

[게임 가격] : ₩ 10,500

[플레이 타임] : 20 시간

[장르] : 퍼즐

[난이도] : 레벨 디자인이 잘 됐다는 평가가 많음.

[특이한 점] : 스테이지 형식이 아닌 오픈월드 형식의 퍼즐 게임

(5) 기타 - 유사 게임 사례 분석3 : Stephen's Sausage Roll



**Stephen's
Sausage
Roll**

[게임 제목] : Stephen's Sausage Roll

[게임 가격] : ₩ 32,000

[플레이 타임] : 40 시간

[장르] : 퍼즐

[난이도] : 난이도가 높은 편

[특이한 점] : 퍼즐 장르에서 보기 힘든 독특한
스토리

4.2 SWOT 분석

[1] 강점 (Strengths)

- ▶ 귀여운 슬라임 테마
- ▶ 짧은 플레이 타임 : 바쁜 일상 속에서도 가볍게 즐길 수 있는 게임.
- ▶ 높은 접근성 : 게임이 간단하고 쉬운 만큼 다수의 사람들에게 어필이 가능.

[2] 약점 (Weaknesses)

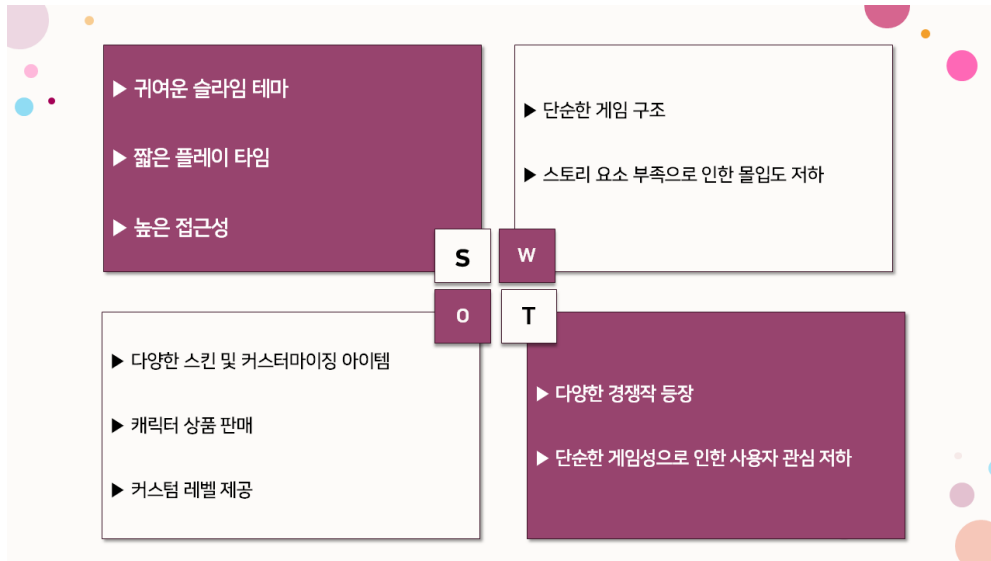
- ▶ 단순한 게임 구조
- ▶ 스토리 요소 부족으로 인한 몰입도 저하

[3] 기회 (Opportunities)

- ▶ 커스터마이징을 통한
- ▶ 캐릭터 상품 판매 : 슬라임 캐릭터의 귀여움을 활용해 인형, 스티커 등 다양한 굿즈 제작을 통해 추가적인 수익 창출 가능
- ▶ 커스텀 레벨 제공 : 사용자 참여를 통해 새로운 콘텐츠가 지속적으로 생성될 수 있도록 유도

[4] 위협 (Threats)

- ▶ 다양한 경쟁작 등장 : 타 경쟁작과 구별될 수 있는 차별화 요소 필요
- ▶ 사용자 관심 저하 : 캐주얼 게임의 특성상 사용자가 쉽게 이탈될 수 있음.



[고려해볼 부분]

▶ 기타 3 Match 게임과의 경쟁 가능성

3 Match 게임처럼 가벼운 게임의 포지션을 고려하는 것은 타 게임과 경쟁을 하는 데 있어서 큰 위협이 될 것이라고 생각합니다. 순발력보다 긴 생각을 필요로 하는 소코 반 형식의 퍼즐 게임은 쉽게 유저가 이탈할 수 있는 계기가 될 수 있습니다.

▶ 패키지 게임으로의 개발 가능성 고려

단순한 게임성은 물론 장점이 될 수 있지만, 유저를 빠르게 이탈하게 만드는 요소가 될 수 있기 때문에 단순 스테이지를 제공하는 퍼즐 게임으로 기획하기보다 귀여운 캐릭터성과 스토리를 앞세워 "PIPE RUSH PARADISE"처럼 오픈 월드 패키지 게임으로 기획하는 것도 한 가지 방향성이 될 수 있다고 생각합니다.