

Fluffy Run

CION

Team 3

구민석, 김지민, 김지훈



LEVEL

1

시작 아이디어

Fluffy Run 개발의 시작

레퍼런스 게임

'Fire Boy & Water Girl'



게임 디자인

- 스테이지 퍼즐 게임
- Player Experience
 - 재미와 성취감을 느낄 수 있도록
- Core mechanic
 - 스테이지별 다양한 구성
 - 스테이지별 코인 획득 후 평가
- Game Loop
 - 주어진 시간 내에 스테이지를 클리어하기



Fluffy Run 제작 타임라인

3주 간의 여정



7/4



게임 디자인 설정
스테이지 기획 회의

개발 파트 분담
구현 내용 구체화 / 개발 시작



7/7

7/14



1차 중간 점검
(기본 기능 구현 등)

2차 중간 점검
(개발 병합 및 마무리)



7/19

플레이어

- 플레이어의 이동 및 동작 : 좌우방향키(이동), 스페이스바(점프)
- 플레이어 속도는 일정하게

장애물 및 인터렉션

- 이동 발판
- 레버와 벽
- 일회성 시간 감축 장애물

구현 대상

스테이지

- 6개의 다양한 스테이지
- 상위 스테이지로 갈수록 난이도 상승

UI

- 스테이지 이동
- 타이머
- 코인 획득
- 게임오버 & 게임 클리어 화면

LEVEL 3

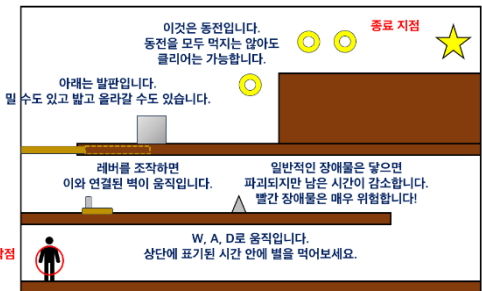
게임 제작 과정 2. 스테이지 디자인

Stage1

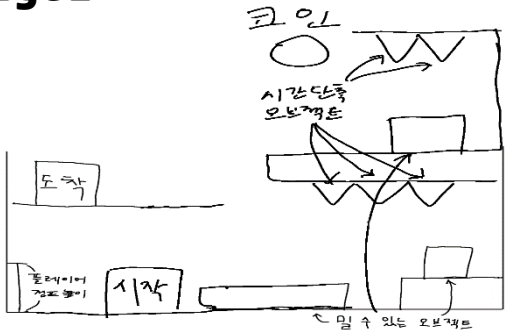
튜토리얼 스테이지

제한시간: 45초

- 별 획득 조건
1. 스테이지 완주
2. 5초 이상 남기기
3. 동전 3개 먹기



Stage2

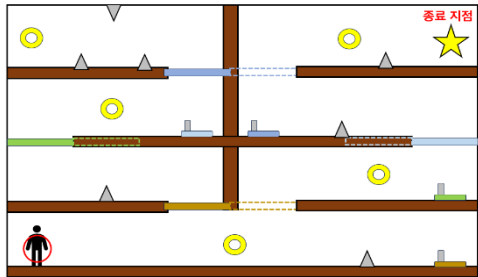


Stage3

레버 스테이지 하나

제한시간: 60초

- 별 획득 조건
1. 스테이지 완주
2. 15초 이상 남기기
3. 동전 5개 먹기

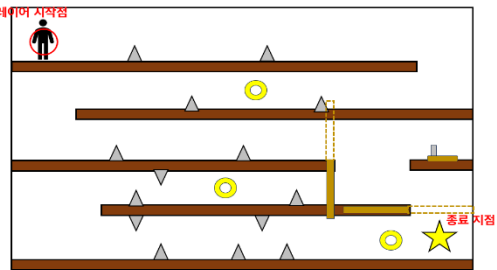


Stage4

앞쪽 스테이지 하나

제한시간: 40초

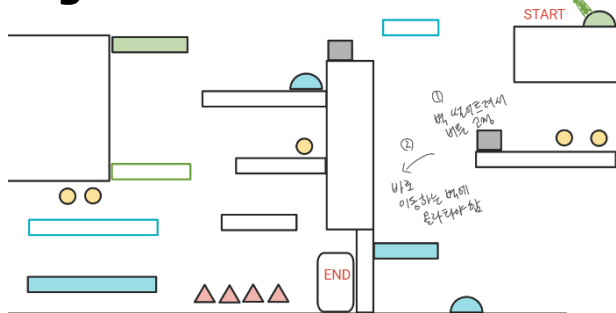
- 별 획득 조건
1. 스테이지 완주
2. 10초 이상 남기기
3. 동전 3개 먹기



Stage5



Stage6



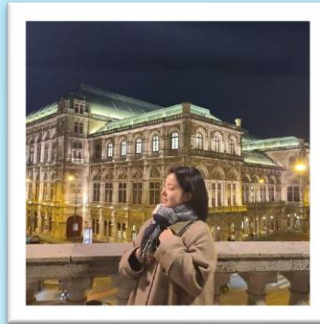
LEVEL
3

게임 제작 과정

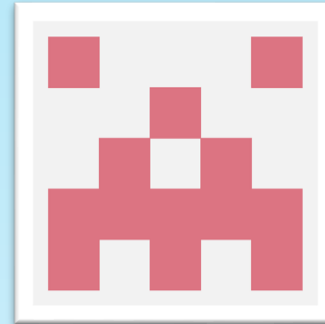
3. 개발



GitHub Desktop을 활용한
협업 및 게임 버전 관리



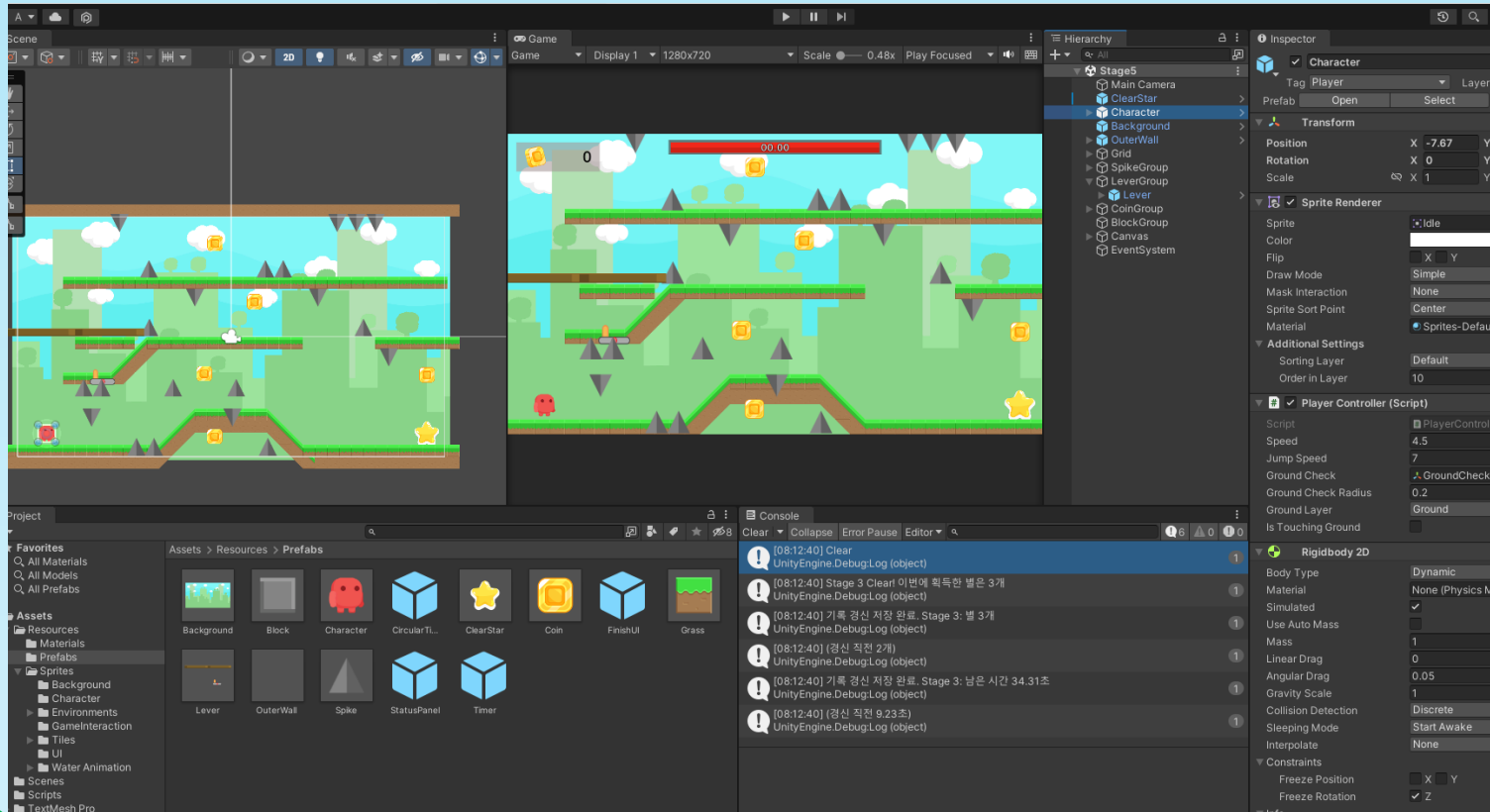
UI 구현



└ 게임 스테이지 구현 ─

LEVEL 3

게임 제작 과정 3. 개발



게임 구현 모습

Branch 사용 모습



스테이지 선택 화면에서 스테이지를 이동하는 로직

- 1스테이지는 처음에 열려있음
- 2~6스테이지는 직전 스테이지를 클리어
해야 플레이할 수 있음

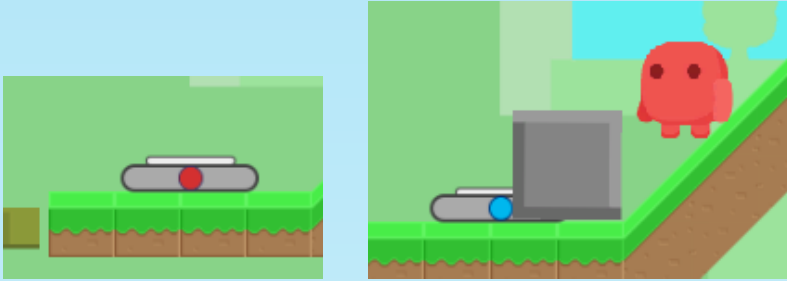
```
// 다음 스테이지로 이동하는 로직
public void OnStage(int stageIndex)
{
    switch (stageIndex)
    {
        // 스테이지 1이나 직전 스테이지가 클리어되어있는 2~6스테이지 > 진입
        case 1:
        case >= 2 and <= StageCount when stageCleared[stageIndex - 1]:
            Time.timeScale = 1f;
            SceneManager.LoadScene($"Stage{stageIndex}");
            break;

        // 직전 스테이지가 미클리어인 2~6스테이지 > 미진입
        case >= 2 and <= StageCount:
            Debug.Log($"Stage {stageIndex - 1} 선행 클리어 필요");
            break;
        default:
            Debug.Log("잘못된 인덱스");
            break;
    }
}
```


LEVEL 3

게임 제작 과정

3. 개발



버튼 로직

버튼 사용 가능 조건

- 플레이어나 박스 등이 Collider2D에 접촉해서 Trigger 시

버튼을 누르고 있을 시 특정 물체가 움직이며,
떼었을 때는 원 상태로 돌아감

```
void Start()
{
    _spriteRenderer = this.gameObject.GetComponent<SpriteRenderer>();
    groundObj = this.transform.Find("StepGround").gameObject;

    offPos = groundObj.transform.position;
    onPos = offPos + diff;
}

void Update()
{
    if (touching)
    {
        _spriteRenderer.sprite = buttonOn;

        groundObj.transform.position = Vector3.MoveTowards(
            groundObj.transform.position, onPos, Time.deltaTime * movingSpeed);

        if (Vector3.Distance(groundObj.transform.position, onPos) < 0.01f)
        {
            groundObj.transform.position = onPos;
        }
    }
}
```

LEVEL 3

게임 제작 과정

3. 개발

```
void Update()
{
    if (touching && Input.GetKey(KeyCode.UpArrow) && canSwitch)
    {
        StartCoroutine(SwitchLever());
    }
}

private IEnumerator SwitchLever()
{
    canSwitch = false;
    isRight = !isRight;
    _spriteRenderer.sprite = isRight ? leverRight : leverLeft;
    objLeft.SetActive(!isRight);
    objRight.SetActive(isRight);

    yield return new WaitForSecondsRealtime(waitingTime);
    canSwitch = true;
}

private void OnTriggerEnter2D(Collider2D col)
{
    if (col.gameObject.CompareTag("Player"))
    {
        touching = true;
    }
}
```



레버 로직

레버 사용 가능 조건

- 플레이어가 Collider2D에 접촉해서 Trigger 시
- 플레이어가 '↑' 키를 눌렀을 시
- 쿨다운 (0.5초)가 지났을 시

OnTriggerEnter2D, Coroutine 등을 사용하여 구현했음

레버를 사용 시 특정 물체의 ON/OFF 상태가 토글됨

LEVEL
3

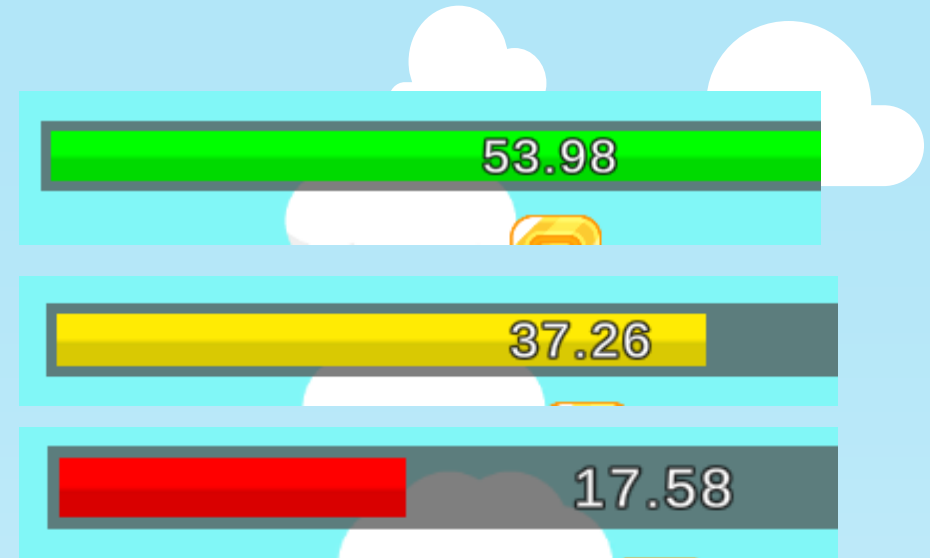
게임 제작 과정

3. 개발

```
private void UpdateLeftAmount()
{
    float ratio = leftTime / gameTime;

    healthBar.fillAmount = ratio;
    healthBar.color = (int) (ratio * 3) switch
    {
        >= 2 => Color.green,
        1 => Color.yellow,
        _ => Color.red
    };

    leftTimeText.text = $"{leftTime:F2}";
}
```



남은 시간 바 표시 로직

게임이 시작하면 n초 안에 별을 먹어 클리어해야 함
남은 시간이 전체 시간의 2/3 이하일 시 노란색,
전체 시간의 1/3 이하일 시 빨간색으로 표시함

남은 시간에 따라 표시 색깔을 변경하는 간단한 로직

LEVEL 3

게임 제작 과정

3. 개발



클리어 처리 로직

3별이라는 추가적인 요소 사용

- 1. 스테이지 시간 안에 클리어
- 2. 코인을 n개 이상 먹고 클리어
- 3. 시간을 n초 이상 남기고 클리어

클리어 결과는 PlayerPrefs를 사용하여 저장하고 스테이지 선택 화면에서 불러올 수 있게 만들

```
private void SetClear()
{
    // 일시정지 처리, 클리어 UI 표시
    Time.timeScale = 0;
    FinishUI.SetActive(true);

    int starCount = 1;
    SetConditionUI(1, true);

    int coinCount = CoinCounter.Instance.coinCount;
    float leftTime = TimeControl.Instance.leftTime;

    timeText.text = $"Left Time: {leftTime:F2} s";

    if (coinCount >= greatCoinCount) starCount++;
    SetConditionUI(2, coinCount >= greatCoinCount);

    if (leftTime >= greatLeftTime) starCount++;
    SetConditionUI(3, leftTime >= greatLeftTime);

    int beforeStarCount = PlayerPrefs.GetInt($"Stage{stageNum}", 0);
    float beforeLeftTime = PlayerPrefs.GetFloat($"Stage{stageNum}LeftTime", 0f);

    Debug.Log($"Stage {stageNum} Clear! 이번에 획득한 별은 {starCount}개");

    // 최고 별 경신
    // 현재 저장된 별보다 모은 별이 더 많다면
    if (beforeStarCount < starCount)
    {
        PlayerPrefs.SetInt($"Stage{stageNum}", starCount);
        Debug.Log($"기록 경신 저장 완료. Stage {stageNum}: 별 {starCount}개");
        Debug.Log($"(경신 직전 {beforeStarCount}개)");
    }

    // ...
}
```

LEVEL

4

게임 소개 및 스크린샷

1. 게임 소개



별

게임을 플레이하는 이유입니다.
제한시간 안에 별을 먹으면 클리어!

가시

닿으면 아파서 시간이 줄어듭니다.

동전

주머니를 두둑하게 만들어줍니다.

제한시간

남은 제한 시간이 나타납니다.

블록

밀 수 있습니다.

레버

오브젝트와 밀당을 할 수 있습니다.

버튼

꼭 밟으면 일합니다.

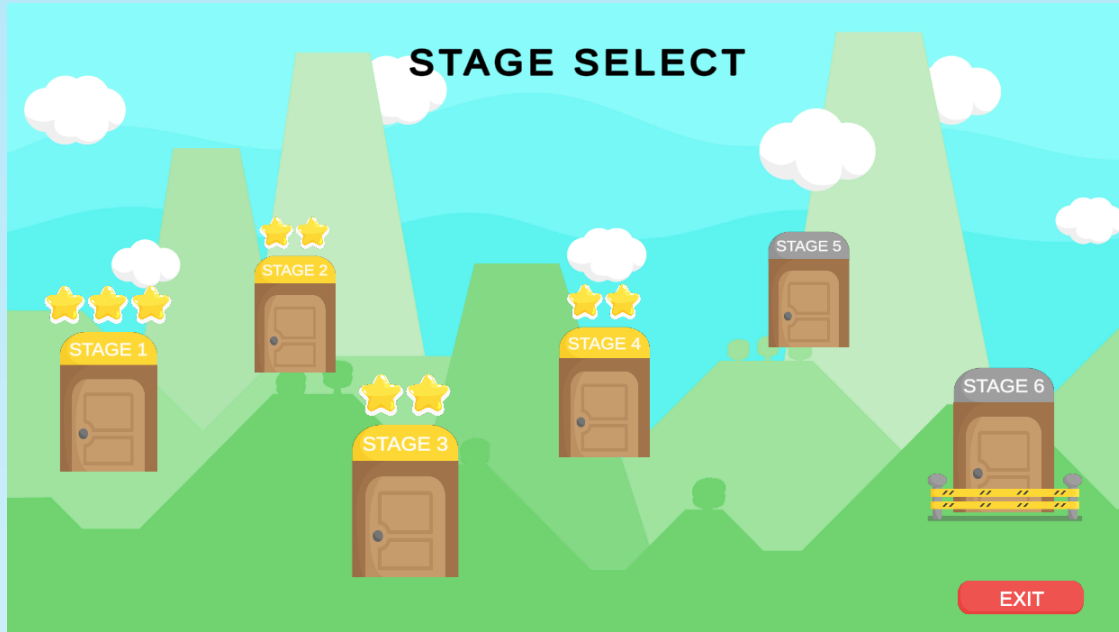
플레이어

당신입니다.

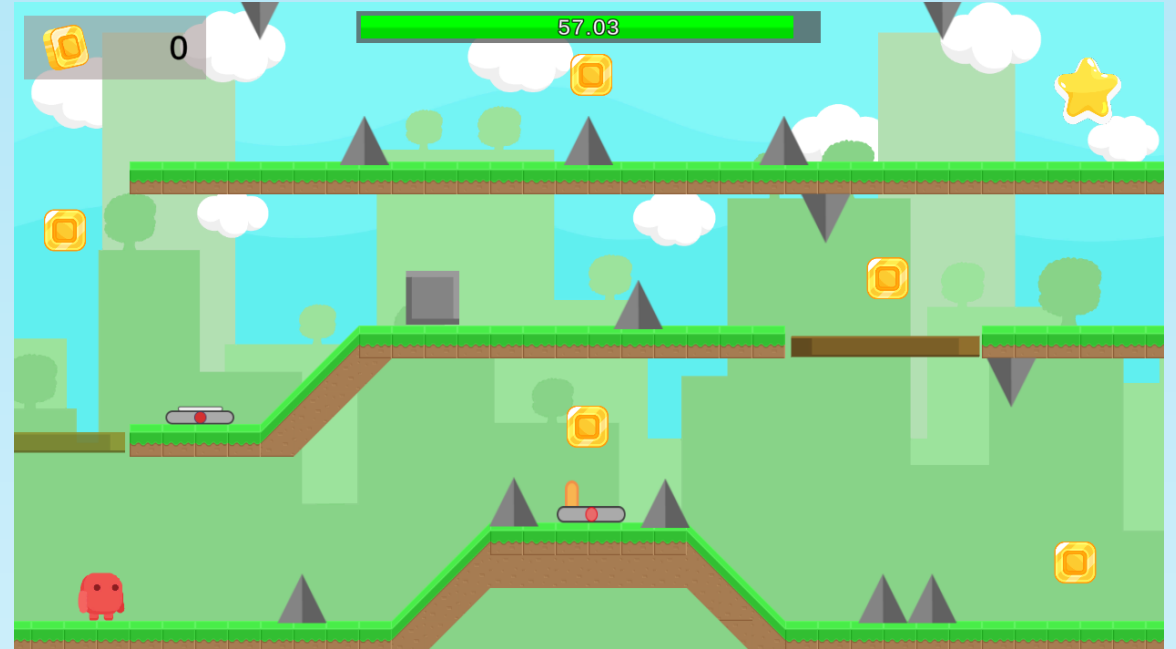
LEVEL
4

게임 소개 및 스크린샷

2. 게임 스크린샷



스테이지 선택 화면



게임 플레이 화면

LEVEL
4

게임 소개 및 스크린샷

2. 게임 스크린샷



클리어 시 화면



감사합니다!

<https://github.com/kjimin0619/stage-puzzle-game>

