

2024-2 소프트웨어 개발의 원리와 실습

Gravitable Escape

Team 17 Spacemas
권민정, 백수지, 윤승혁, 오제헌, 조휘현

CONTENTS

01 게임 설명

- 제안서 리마인드
- 플레이 영상

02 변경사항

- 결과물 게임의 변화
- 개발과정의 변화

03 Discussion

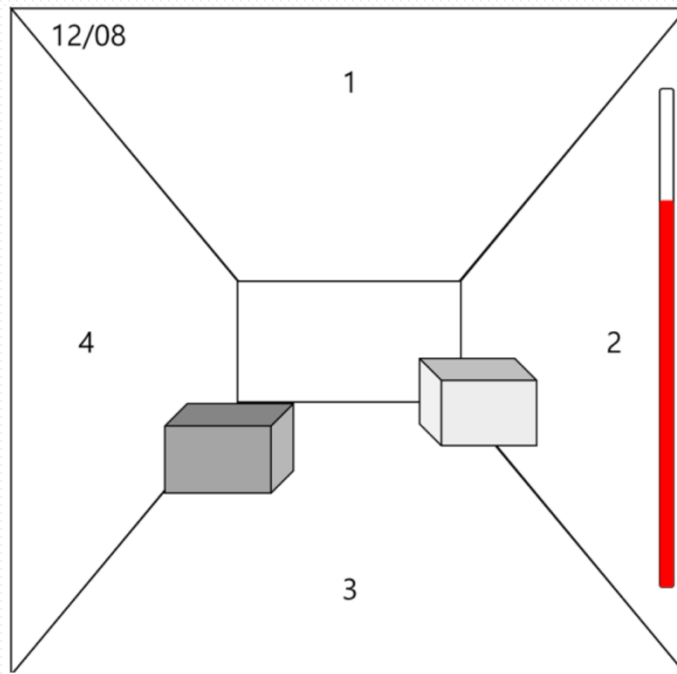
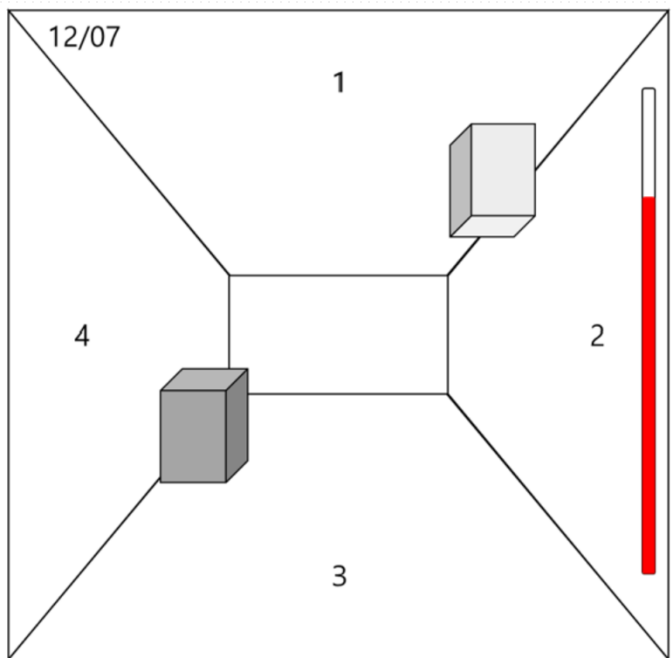
- 느낀 점 – 소통의 중요성
- 어려웠던 점 1 – Concurrency
- 어려웠던 점 2 – Bottom-up
- 개선점

01. 게임 설명

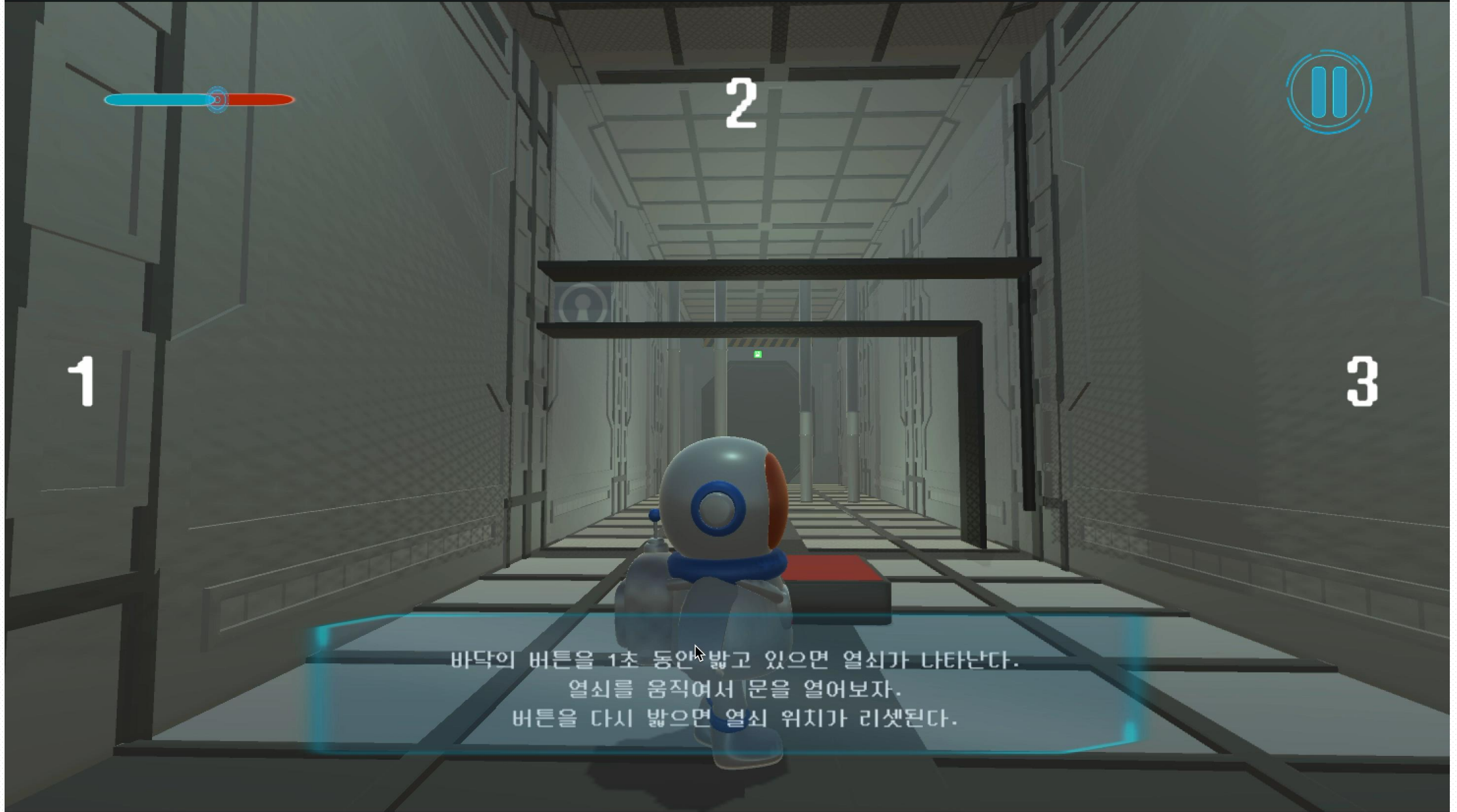


Gravitable Escape 이란?

- 장르: 3인칭 퍼즐 플랫폼머 게임
- 플레이 목표: 우주정거장 통로를 탈출해야 함
- 주요 기믹: 중력전환을 사용하여 퍼즐 풀기, 장애물 피하기







바닥의 버튼을 1초 동안 밝고 있으면 열쇠가 나타난다.
열쇠를 움직여서 문을 열어보자.
버튼을 다시 밝으면 열쇠 위치가 리셋된다.

02. 변경사항

제안서 vs 최종 결과물

게임 결과물의 변화

큰 틀에서는 제안서와 동일했지만, 디테일에는 예상한 대로 많은 수정이 있었다

- 큰 틀은 제안서와 동일
- 계획 당시 "선택적" 이라고 생각한 요소의 삭제
 - 시작, 끝 스토리
 - 수집형 콘텐츠
- 디테일의 변화
 - 시간 제한, 거리 표시 삭제
 - Stage 3개 -> 튜토리얼 + Stage 2개

개발과정의 변화

스프린트 별 태스크 분배: 구현 내용으로 나누기 -> 목표 기준으로 나누기

Sprint x 에서 ~~라는 태스크를 하자!
(구현 내용 기준)



Sprint x 에서 ~~가 되도록 하자!
(목표 기준)

개발과정의 변화

스프린트 별 태스크 분배: 구현 내용으로 나누기 -> 목표 기준으로 나누기

4. Sprint2 Milestone 설정

스프린트에서 진행해야 할 세부 task를 선정하는 것도 중요하지만, 프로젝트 내에서 우리가 어디쯤 와있는지를 확인하기 위해 스프린트 전체의 Milestone이 필요하다는 의견이 있었다. 따라서, Sprint2의 목표를 **Tutorial Stage를 플레이 가능한 형태로 완성**하기로 정했다. 이번 스프린트에서 Tutorial Stage를 완성하고 테스트하면서 시행착오를 겪으면, 나머지 Stage의 시나리오, 구현방법 등을 알맞게 수정한 후 시작할 수 있기 때문에 더 효율적으로 할 수 있을 것으로 기대한다.

스프린트 백로그, Task 선정 내역 및 담당

1. Sprint2 유저 스토리 선정 내역

Sprint2의 Milestone인 **"Tutorial Stage를 플레이 가능한 형태로 완성"**하기 위해 선정한 유저 스토리는 아래와 같다.

- 플레이어가 방향키, 스페이스바에 따라 움직이고 점프하며 적절한 애니메이션을 수행한다.
- 숫자키 1, 2, 3을 누르면 중력의 방향이 바뀌고, 중력에 영향을 받는 물체들은 바뀐 중력에 맞게 떨어진다.
- 플레이어가 튜토리얼 스테이지를 관찰할 수 있다.
- 플레이어가 가시의 윗면에 찰리면 목숨이 줄어든다.
- 플레이어가 원홀에 가까이 가면 빨려들어가고 정해진 위치로 순간이동한다.
- 플레이어가 중력을 적절히 전환하여 미로 퍼즐을 풀면 그 이후로 이동할 수 있는 문이 열린다.

2. 스프린트 3 Milestone 및 유저 스토리 선정

'오프닝, 클로징을 제외한 게임의 전 과정을 플레이 가능한 형태로 완성'하는 것을 Sprint 3의 전체 목표로 설정했다. 목표 달성을 위해 아래와 같은 유저 스토리를 선정했다.

- 플레이어가 자연스럽게 이동한다.
- 3개의 Stage를 플레이할 수 있다.
- 플레이 과정에서 Status 확인이 가능하도록 UI를 보여준다.
- 플레이 과정에서 적절한 Sound Effect가 재생되도록 한다.
- 플레이 과정에서 적절한 Particle Effect가 재생되도록 한다.

기존의 개발 일정

스프린트 별 태스크 분배: 구현 내용으로 나누기 -> 목표 기준으로 나누기

Sprint 1

기본적인 모델링, 스크립트

- Map 기본 디자인
- 기본 에셋 제작
 - Background
 - Player
- 기본 스크립트
 - Player 이동
 - Camera
 - 중력 전환

Sprint 2

심화된 모델링, 스크립트

- Map 추가 디자인
- 심화 에셋 제작
 - Animation
 - Obstacles
- 심화 스크립트
 - Player Obstacle 상호작용

Sprint 3

UI, 추가 기능 구현

- UI 제작
- 카메라 회전 스크립트
- Sound 추가

Sprint 4

테스팅, 오프닝 & 엔딩

- 오프닝, 엔딩 추가
- 수집형 콘텐츠 추가
- Testing & Debugging

수정된 개발 일정

스프린트 별 태스크 분배: 구현 내용으로 나누기 -> 목표 기준으로 나누기

Sprint 1

기초작업

- Map 구상
- 기본 에셋 제작
 - Background
 - Player
- 기본 스크립트
 - Player 이동
 - Camera
 - 중력 전환

Sprint 2

Tutorial 구현 & Testing

- Tutorial Stage 배치
- 심화 에셋 제작
 - Animation
 - Obstacles
- 심화 스크립트
 - Player, Camera 스크립트 수정
 - 워홀, 가시

Sprint 3

Stage 1, 2 구현 & Testing

- Stage 1, 2 배치
- Player, Camera 스크립트 수정
- UI 제작
- 카메라 회전 스크립트
- Sound 추가

Sprint 4

Title, Menu, Ending Debugging

- Title, Menu, Ending Scene 제작
- 꾸미기
- Test & Debugging

기존의 개발 일정

스프린트 별 태스크 분배: 구현 내용으로 나누기 -> 목표 기

플레이 가능한 시점

Sprint 1

기본적인 모델링, 스크립트

- Map 기본 디자인
- 기본 에셋 제작
 - Background
 - Player
- 기본 스크립트
 - Player 이동
 - Camera
 - 중력 전환

Sprint 2

심화된 모델링, 스크립트

- Map 추가 디자인
- 심화 에셋 제작
 - Animation
 - Obstacles
- 심화 스크립트
 - Player Obstacle 상호작용

Sprint 3

UI, 추가 기능 구현

- UI 제작
- 카메라 회전 스크립트
- Sound 추가

Sprint 4

테스팅, 오프닝 & 엔딩

- 오프닝, 엔딩 추가
- 수집형 콘텐츠 추가
- Testing & Debugging

수정된 개발 일정

스프린트 별 태스크 분배: 구현 나... 준으로 나누기

플레이 가능한 시점

Sprint 1

기초작업

- Map 구상
- 기본 에셋 제작
 - Background
 - Player
- 기본 스크립트
 - Player 이동
 - Camera
 - 중력 전환

Sprint 2

Tutorial 구현 & Testing

- Tutorial Stage 배치
- 심화 에셋 제작
 - Animation
 - Obstacles
- 심화 스크립트
 - Player, Camera 스크립트 수정
 - 워홀, 가시

Sprint 3

Stage 1, 2 구현 & Testing

- Stage 1, 2 배치
- Player, Camera 스크립트 수정
- UI 제작
- 카메라 회전 스크립트
- Sound 추가

Sprint 4

Title, Menu, Ending Debugging

- Title, Menu, Ending Scene 제작
- 꾸미기
- Test & Debugging

03. Discussion



느낀 점

소통의 중요성



- 소통을 위한 작업들의 중요성
 - Clean code
 - 주석 작성
 - PR message 작성

어려웠던 점 1

Sequential Programming v.s. Concurrent Programming

- 우리가 익숙한 코드: 순차적(sequential)으로 실행됨
→ 1번 실행
- Unity의 Update() 함수: 매 프레임 실행됨
→ 여러 번 실행

```
void Update()  
{  
    if(player is walking) walkingSound.Play();  
} // Does not work!
```

어려웠던 점 2

Top-down v.s. Bottom-up



- 코드의 구조 (UML, API 등) 을 미리 정하지 않고 작업함
 - ➔ Sprint 3 초기에 리팩토링 진행
- Architect를 할 만한 전문가의 부재
- 5인의 소규모 팀

다시 게임을 만든다면?

- 유료 에셋 사용으로 퀄리티 높이기
- 중력이 자동으로 바뀌는 등의 새로운 기믹 추가
- 한 번 해봤으니깐 더 잘할 수 있겠다!

감사합니다 !

