

프로젝트 관리

Unity 초기 세팅

Steps After Cloning

1. **Clone the Repository:** Your coworker should clone the repository using Git:

```
bash                                     코드 복사  
git clone <repository-url>
```

2. **Open Unity Hub:** After cloning, they should open Unity Hub.

3. **Add the Project:**

- In Unity Hub, there's an option to add an existing project. They can click on the **Add** button.
- Navigate to the folder where the repository was cloned, select the root folder of the project (the folder containing the **Assets** directory), and click **Open**.

4. **Unity Initializes the Project:**

- Once the project is added to Unity Hub, they can select it and click **Open**. Unity will then initialize the project.
- Unity will generate the necessary folders like **Library** and **Packages** based on the information in the project.

5. **Restoring Packages:**

- If the project uses any packages specified in the **Packages/manifest.json** file, Unity will automatically restore these packages when the project is opened.

- git clone → Unity Hub에서 Add Project

Unity/Git 작업 원칙

1. Scene Ownership

- 불가피한 일이 아닌 이상 Scene의 경우 **Ownership을 철저히 정해놓고 작업한다.** (여러 사람이 하나의 Scene을 작업할 경우 merge 과정에서 충돌 발생 확률이 높다고 함)
 - Stage별로 Scene을 분리한 다음, 각자 맡은 Scene에서 작업하는 방식 채택

- 만약 동일 Scene을 동시에 작업해야 하는 경우 Scene을 2개로 카피한 후 나중에 Merge하는 방식

유니티 협업하는 법 (2023.09.13)

유니티로 팀과제를 한지도 벌써 4일째가 되간다. c*|*|*가장 먼저 생각나는것은 깃과 깃허브이다. 깃과 깃허브를 이용해 서로의 작업물을 올리고 merge하는 방식으로 협업을 진행한다. 하지만

v <https://velog.io/@cjb0325/%EC%9C%A0%EB%8B%88%ED%8B%B0-%ED%98%91%EC%97%85%ED%95%98%EB%8A%94-%EB%B2%95-2023.09.13>



2. Coding Convention

기본

- Indentation** : 4칸 띄어쓰기
- Vertical Spacing** : 함수 내 가독성을 위해 필요한 경우 빈 줄 삽입

표기법

- 클래스명, 함수명, static 변수명** : 파스칼 표기법 ex) *GameManager, GameOver()*
- non-static 변수명, 함수 파라미터** : 카멜 표기법 ex) *gameScore*
- bool 타입 변수명** : is, has 접두사 ex) *isGameActive*
- ENUM** : 대문자만 사용
- Getter, Setter** : Get, Set 접두사 ex) *GetScore, SetScore*
- UI 관련** : 컴포넌트 접미사 ex) *gameScoreText, restartButton*

중괄호 사용

- {}** : 줄 바꿈 후 한 행의 { 사용
- 조건이 한 가지인 if문** : 중괄호 사용

기타 사항

- 변수명으로 키워드와 비슷한 이름, 줄임말 등 사용하지 않기
- 접근 제한자 명시하기 (public, private, protected)

- 함수가 길어지면 각각의 기능 단위로 함수 분리하기

3. Github Branch Convention

- **Main Branch** : 항상 배포 가능한 안정적인 게임 버전을 유지합니다. 스프린트가 끝날 때마다 `dev` 브랜치에서 병합되며, 모든 팀원의 코드 리뷰와 QA 테스트를 거쳐야 합니다.
- **Dev Branch** : 다음 버전 개발이 진행되는 브랜치입니다. 기능 개발이 완료된 `feature` 브랜치들이 여기로 병합됩니다. 스프린트 중간에도 지속적인 통합 테스트를 진행합니다.
- **Feature Branches** : 각 게임 기능(예: 오리 캐릭터 움직임, 장애물 시스템, UI 등)마다 `dev`에서 분기하여 생성합니다. 네이밍 규칙은 'feature/기능명'으로 합니다. 개발 완료 후 `dev`로 병합 시 최소 1명의 코드 리뷰가 필요합니다.
- **Hotfix Branch** : 라이브 버전에서 긴급한 버그 발생 시 `main`에서 분기하여 수정 후 `main`과 `dev`에 병합합니다. 빠른 대응이 필요하므로, PM과 주 개발자의 승인만으로 진행 가능합니다.

4. Github Commit Convention

- **Commit 메시지 형식:** `[타입] 설명`
 - 타입 : Feat(새 기능), Fix(버그 수정), Refactor(리팩토링), Style(코드 스타일 변경), Docs(문서 수정), Test(테스트 코드), Chore(기타 변경)
 - 예: `[Feat] 오리 점프 기능 추가`
- **Atomic Commits** : 각 커밋은 독립적이고 작은 단위의 변경사항만 포함해야 합니다.
- **Unity Scene 변경** : Scene 파일 변경 시 반드시 커밋 메시지에 명시하고, 가능한 한 다른 변경사항과 분리하여 커밋합니다.

[Git] Commit Message / PR 잘 쓰는 방법

커밋 메시지 규칙 커밋 메시지 예시 깃 이모지heart 커밋 메시지는 제목과 본문으로 나누어진다! 설명이 충분하다면 제목만으로도 괜찮다 하지만 어떤 변경사항이 있는지 맥락과 설명이 필요하다면

 <https://velog.io/@ye-ji/Git-PR-%EC%9E%98-%EC%93%BC%EB%8A%94-%EB%BC%A9%EB%BC%95>



Commit Message / PR 잘 쓰는 방법

5. Github PR

The screenshot shows a GitHub pull request (PR) interface. At the top, a comment from user 'pcy1203' is displayed, dated 2 days ago. The comment includes a 'Commit 기록' (Commit History) section with three commits: [Fix] (S4.0.4) Story UI with Background Music [T3.0.4], [Feat] Background, and [Feat] (S4.0.5) Code Refactoring. Below this is a 'Task 목록' (Task List) section with three items: S4.0.4 QA 결과 확인 및 버그 수정 (ex. 점프 난이도 등), S4.0.5 디자인 패턴/프로젝트 코드 전반의 리팩토링, and S4.0.6 화면/배경 개선. A '주요 사항' (Key Points) section follows, stating that the story UI with background music has been implemented and tested, and that the background has been refactored. The PR has 3 commits from 'pcy1203' and a review request from 'june991012'. The review from 'june991012' is dated 2 days ago and includes a link to 'View reviewed changes'. The PR is currently open.

스크럼 관리 (Geekbot)

The screenshot shows a Geekbot message from '박찬영 (Team 08)' at 7:26 PM. The message is an update for '데일리 스크럼' (Daily Scrum). It includes the following questions and answers:

- 오늘 기분이 어때요?
즐겁습니다~
- 어디서 누구와 함께하고 있나요?
관정에서 혼자 작업 중입니다!
- 오늘은 어떤 일을 하나요?
 - S4.0.2 배경음악, 사운드 효과 적용 -> 남은 부분
 - S4.0.3 방향키/음소거 UI 추가
 - S4.0.4 QA 결과 확인 및 버그 수정 (ex. 점프 난이도 등)디자인 패턴, 리팩토링 외에 다른 부분들은 최대한 빨리 작업하려고 합니다~
- 그 일을 하면서 예상되는 어려움, 협업이 필요한 부분, 혹시 팀원에게 하고 싶은 말이 있을까요?
없습니다! (7시부터 작업 시작했습니다!)



박찬영 (Team 08) APP 9:47 PM

박찬영 posted an update for 마무리 스크럼

오늘 일은 즐거웠나요?

즐거웠습니다~

오늘 한 업무는 어떤 것들이 있을까요? ----- 간단한 답변과 함께 태스크 페이지

📝에 반영해주세요!

• [Fix] (S4.0.4) Player Movement & Key Input [T1.0.5 ~ T1.0.7]

• [Feat] (S4.0.3) Input Key Selection UI & Feature

방향키/WASD키 전환 구현했습니다!

그리고 오리 동작 우선순위 구현했습니다만 PR은 아직 안 날려서, PR 날리면 재
응님한테도 말씀드리겠습니다~

+ 오리 점프가 가끔 이상해지는 문제가 Update() 함수 써서 그런 것 같아서,
Update()랑 FixedUpdate() 함수로 로직을 나눴습니다. (물리 엔진은
FixedUpdate 써야 오류가 없다고 하네요) 이렇게 하니까 일단은 이상 없어보이
는데, 추후에 테스트 해주시면 감사드리겠습니다! (만약 오리날다 먹고 점프 중에
좌우 이동을 하면 오리가 흙 떨어지던데... 크게 부자연스럽지는 않은 것 같아요)

내일은 어떤 업무를 하나요?

• UI 클릭과 엔딩에 대한 소리 구현

• 음향 슬라이더 기능 구현

• 자잘한 오류 수정 및 가능하면 Main Scene 꾸미기

가능하다면 오늘 새벽~내일 낮까지 조금 더 해보겠습니다! 작업 끝나면 형준님께
알려드릴게요

팀원들에게 하고 싶은 말이 있나요? (협업요청, 상황보고, 어떤 말이든)

기능 자체는 거의 다 끝나가는 것 같은데 다들 파이팅입니다!

짱 1 🎉 1 ⚡