**2. Git 브랜치 전략**

**📌 브랜치 구조**

**bash**

**📂 main → 출시 버전 (팀장만 수정 가능)**

**📂 develop → 개발 브랜치 (완성된 기능 머지)**

**┣ 📂 feature/ui → UI 작업 브랜치**

**┣ 📂 feature/system → 시스템 구현 브랜치**

**┣ 📂 feature/level → 레벨 디자인 브랜치**

**┣ 📂 fix/bugfix → 버그 수정 브랜치**

**✔ 모든 작업은 feature/ 브랜치에서 진행 후 develop으로 머지  
✔ 버그 수정 시 fix/ 브랜치에서 작업 후 머지**

**3. 작업 분배 방법**

**📌 작업 관리 툴 사용  
✔ Trello / Notion 활용  
✔ 작업 상태 표시 (To Do → In Progress → Review → Done)**

**📌 역할 분배 예시**

|  |  |  |
| --- | --- | --- |
| **역할** | **담당자** | **작업 내용** |
| **기획** | **A** | **게임 시스템 문서 작성, UI 기획** |
| **프로그래머** | **B** | **시스템 구현, 버그 수정** |
| **아티스트** | **C** | **배경 모델링, 애니메이션 작업** |
| **기획 & PM** | **A** | **일정 조율, 피드백 정리** |

**4. Git 커밋 메시지 규칙**

**📌 커밋 메시지 형식**

**scss**

**복사편집**

**[태그] 작업 내용 (간단한 설명)**

**✔ 태그 종류**

* **[UI] → UI 작업 관련**
* **[System] → 시스템 구현 관련**
* **[Level] → 레벨 디자인**
* **[Fix] → 버그 수정**
* **[Doc] → 문서 작성**

**📌 커밋 메시지 예시  
✅ [UI] 메인 화면 버튼 배치 완료  
✅ [System] 점프 기능 추가  
✅ [Fix] 점프 버그 수정**

**5. 정리**

**✔ 기획자는 feature/ui 브랜치에서 UI 설정 후 머지 요청  
✔ 작업은 각자 브랜치에서 진행 후 develop에 합치기  
✔ Trello / Notion으로 작업 관리하며 Git 커밋 메시지 규칙 지키기**

**📄 Git\_Workflow.docx**