

다들 깃헙 쓰시나요?

GitHub

프로젝트의 깃허브 공유

GitHub Desktop을 이용한 Git 관리법

목차

- 형상관리 프로그램
- Git
- GitHub, GitHub Desktop
- Commit /Push / Pull
- Branch / Merge branch
- Git Flow
 - Trunk-Based flow
- My GitHub

형상관리 프로그램

- 형상관리
 - 변경사항을 추적하고 관리하는 프로그램
- 내 프로그램이 언제 어디서 어떻게 변경되었는지 추적 가능
 - 버그 탐색 가능
 - 소스코드 원상복구 가능
 - 협업하는데 유용
 - 파일 자체를 저장하는 것이 아닌 파일의 변동사항을 저장
 - 파일을 저장하기도 한다.
- Git, SVN

Git

- 형상관리 프로그램
- GitHub, GitLab, Azure DevOps와 같은 git 저장소가 있음
 - 다른 사람과 인터넷으로 연동하여 파일을 공유할 수 있음
- Git을 이용하여 컴퓨터 내부, 인터넷에 프로젝트를 관리할 수 있음.

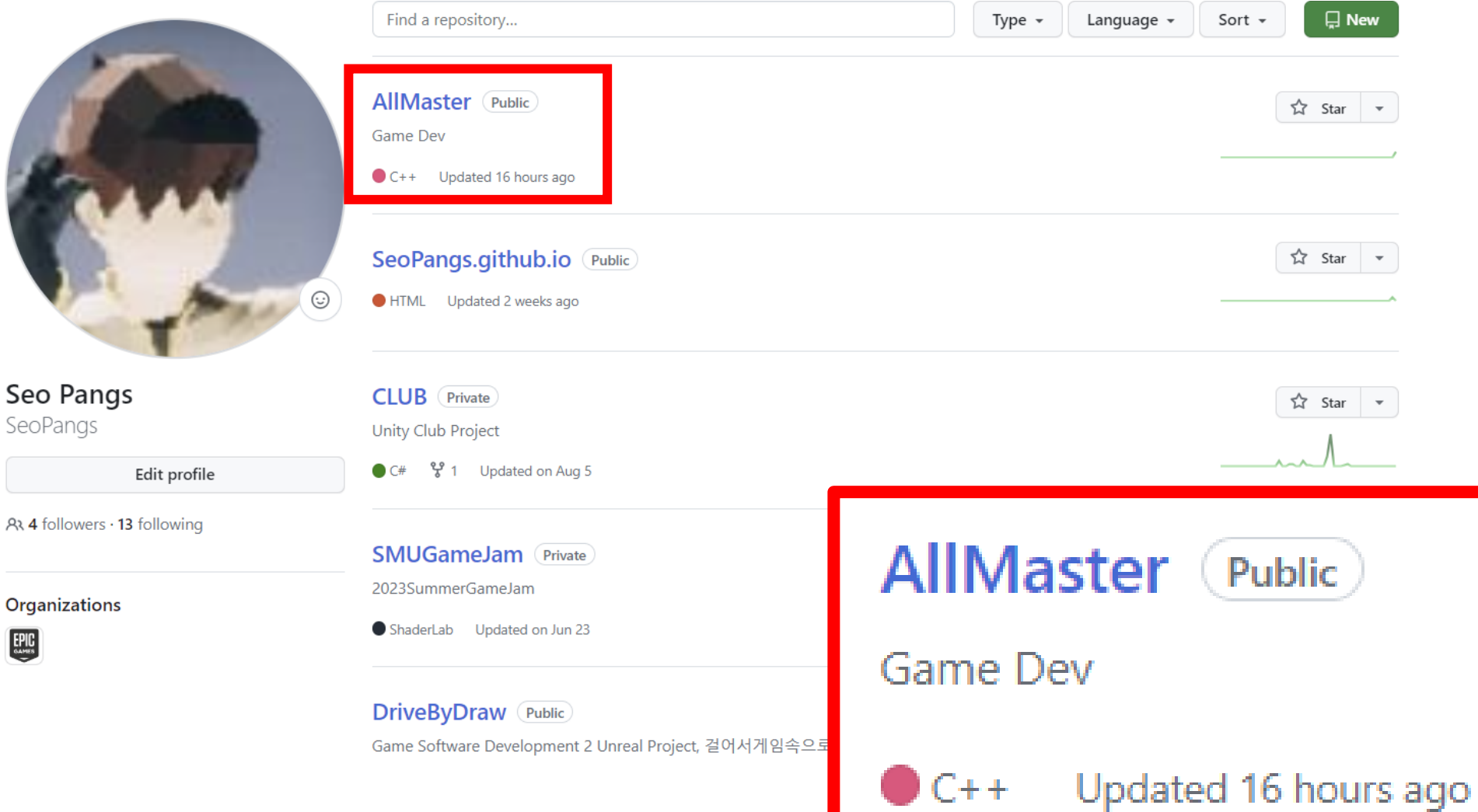


GitHub

- Git 저장소 중 하나
 - Git을 기반으로 구성됨, git 호스팅 가능
- 제일 많이 쓰는 플랫폼
- 프로젝트 공유가 편리함

- 저장소를 통해 프로젝트를 공유

Repository : 저장소



The image shows a GitHub profile page for a user named 'Seo Pangs'. The profile includes a circular avatar, a bio, and a list of repositories. Two repositories are highlighted with red boxes: 'AllMaster' and 'AllMaster' (repeated). The 'AllMaster' repository is a public repository for Game Dev, written in C++, and was updated 16 hours ago. The profile also shows 4 followers and 13 following, and an organization named 'EPIC GAMES'.

Find a repository... Type Language Sort New

AllMaster Public
Game Dev
C++ Updated 16 hours ago

SeoPangs.github.io Public
HTML Updated 2 weeks ago

CLUB Private
Unity Club Project
C# 1 Updated on Aug 5

SMUGameJam Private
2023SummerGameJam
ShaderLab Updated on Jun 23

DriveByDraw Public
Game Software Development 2 Unreal Project, 걸어서게임속으로

AllMaster Public
Game Dev
C++ Updated 16 hours ago

Seo Pangs
SeoPangs
Edit profile
4 followers · 13 following
Organizations
EPIC GAMES

내 저장소 안의 파일

SeoPangs / AllMaster

Q Type [] to search

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

AllMaster Public

Pin

Unwatch 1

Fork 0

Star 0

main 1 branch 0 tags

Go to file

Add file

<> Code

About

SeoPangs Update README.md 90d7af1 16 hours ago 5 commits

Config	Init()	17 hours ago
Content	Skill System	16 hours ago
Source	Skill System	16 hours ago
.editorconfig	Init()	17 hours ago
.gitattributes	Initial commit	17 hours ago
.gitignore	Update .gitignore	17 hours ago
AllMaster.png	Init()	17 hours ago
AllMaster.uproject	Init()	17 hours ago
README.md	Update README.md	16 hours ago

README.md

AllMaster

해당 프로젝트는 개인적으로 언리얼을 다루기 위함이며, 동시에 만들어보고 싶었던 것을 동시에 작업한다.

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

C++ 96.7%

C# 3.0%

C 0.3%

GitHub Desktop

- Git 너무 어려워보이는데 그냥 안쓰면 안되나?
- 그래서 간단히 쓰기 좋은 GitHub Desktop가 있다.



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Open the repository in your external editor

Select your editor in [Options](#)

Repository menu or **Ctrl** + **Shift** + **A**

Open in Visual Studio Code

View the files of your repository in Explorer

Repository menu or **Ctrl** + **Shift** + **F**

Show in Explorer

Open the repository page on GitHub in your browser

Repository menu or **Ctrl** + **Shift** + **G**

View on GitHub

Current repository
AllMaster

SeoPangs

AllMaster

SeoPangs.github.io

U2U

Yoonsin

CraftingStar

Current branch
main

Fetch origin
Last fetched just now

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Open the repository in your external editor

Select your editor in [Options](#)

Repository menu or **Ctrl** + **Shift** + **A**

View the files of your repository in Explorer

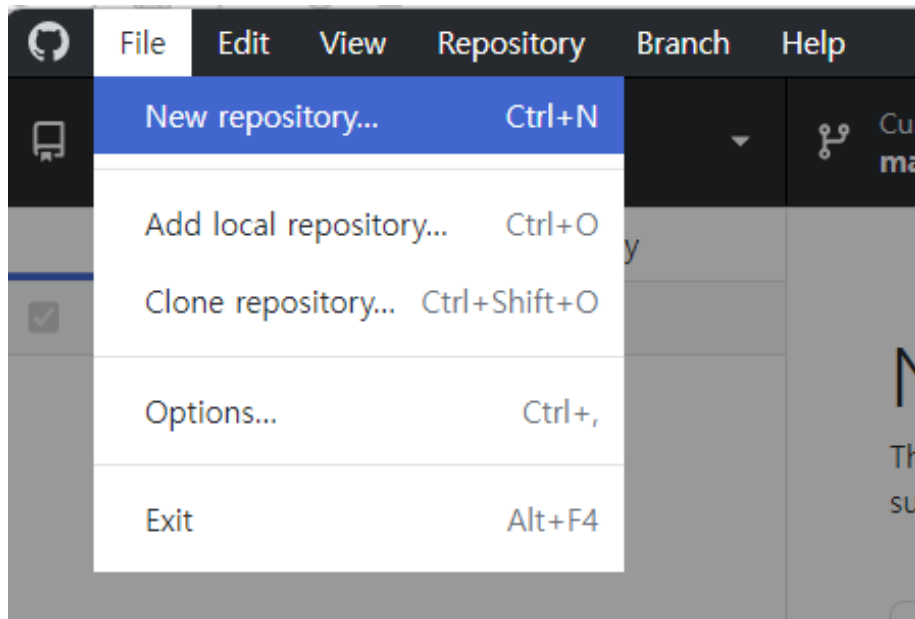
Repository menu or **Ctrl** + **Shift** + **F**

Open the repository page on GitHub in your browser

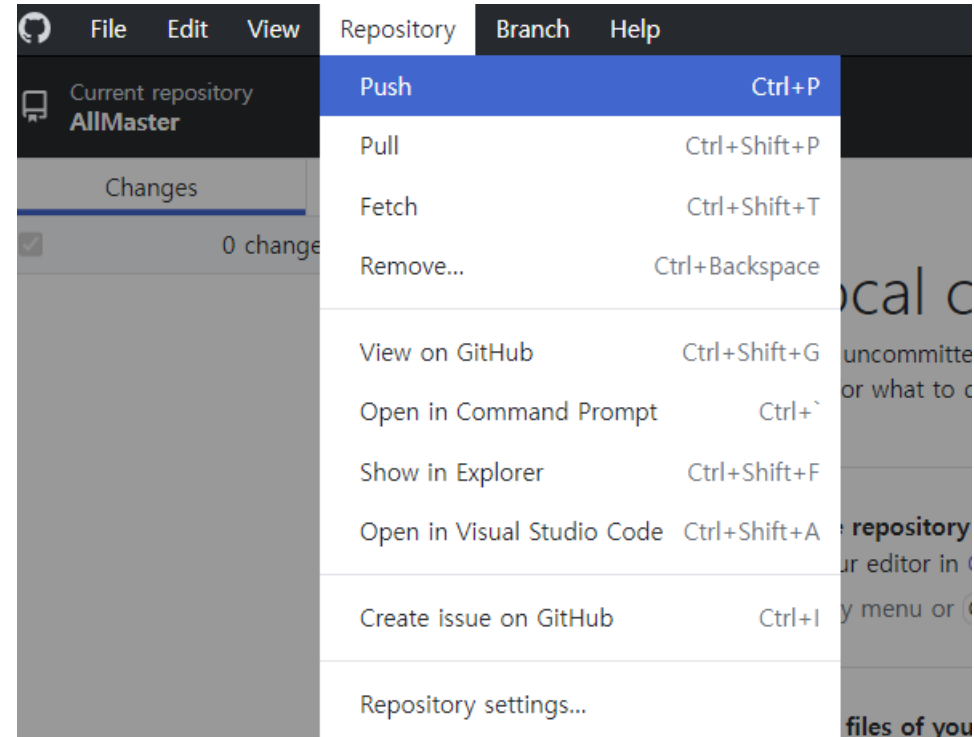
Repository menu or **Ctrl** + **Shift** + **G**

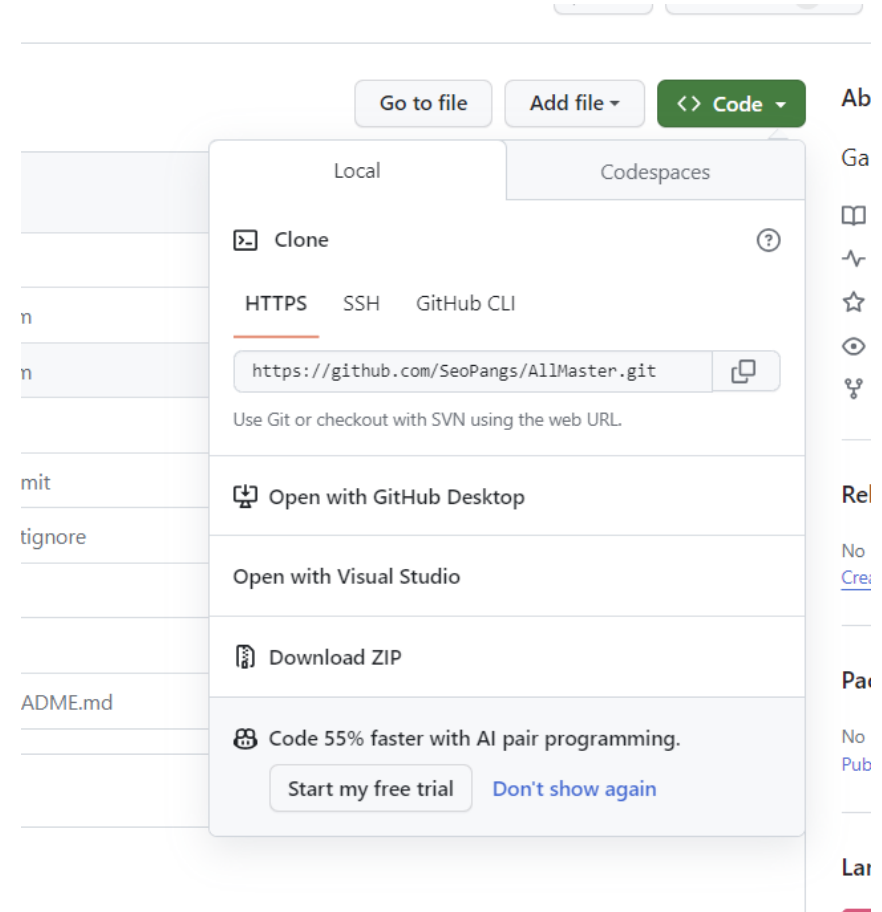
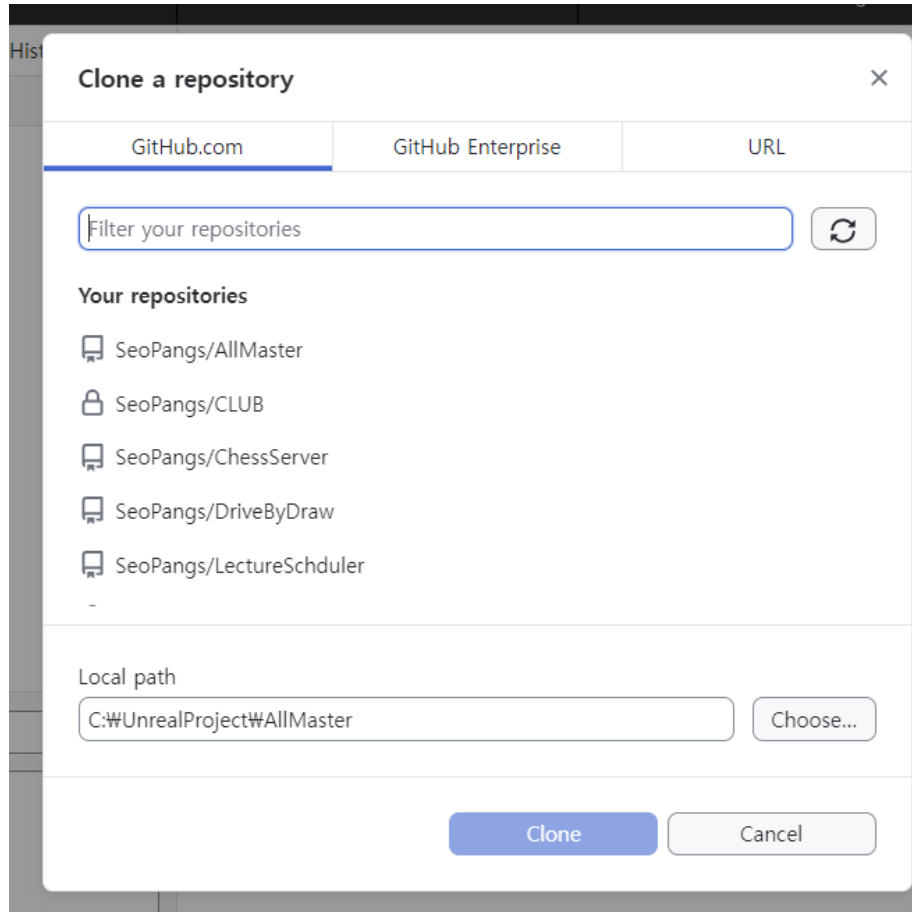
해당 작업을 주로 사용

- 파일들을 로컬로 가져오는 작업



- 저장소 관련 실행 액션





Git 행동

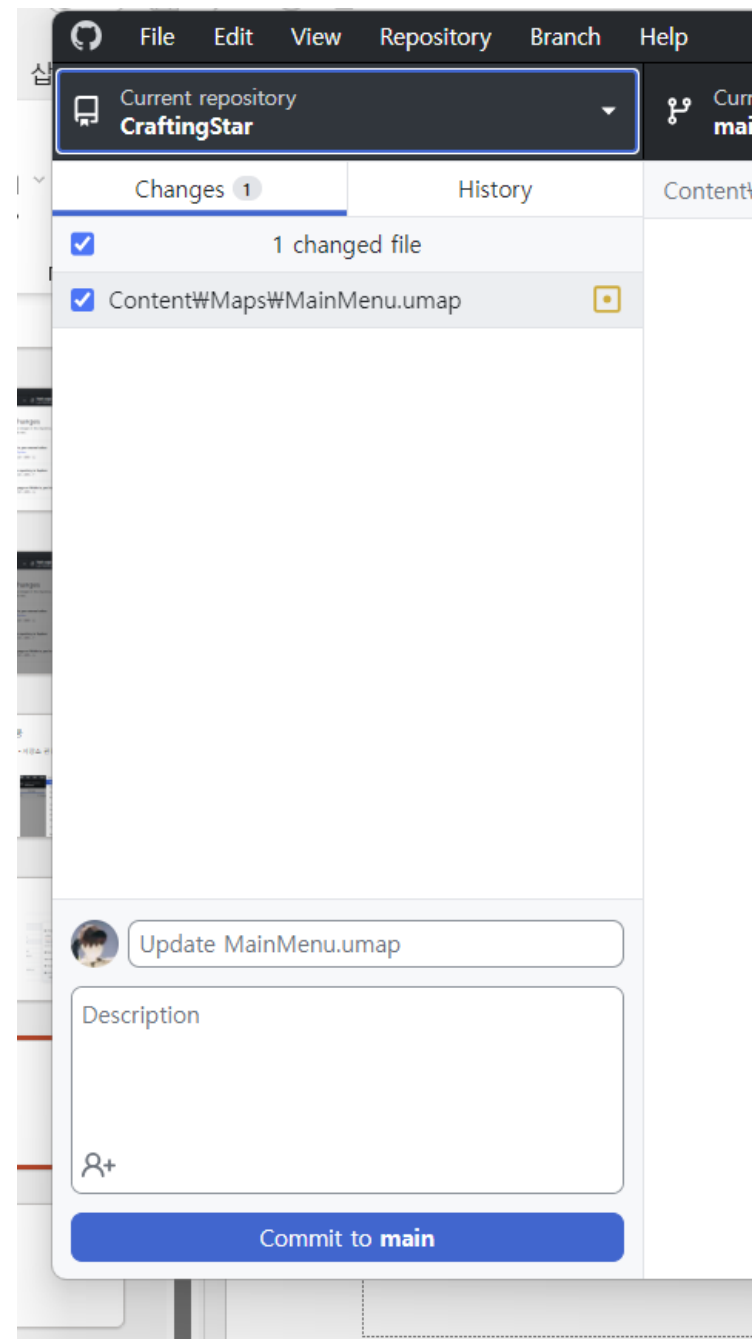
- Commit
 - 의미있는 변경 작업들을 저장소에 기록하는 동작
- Push
 - 로컬의 커밋들을 원격 저장소(인터넷 저장소)로 옮기는 동작
- Pull
 - 원격 저장소의 정보를 가져오고
 - 자동으로 로컬에 병합(Merge)까지 수행
- Branch
 - 동일한 소스 코드에서 가져온 독립적인 개발 라인

GitHub는 보통

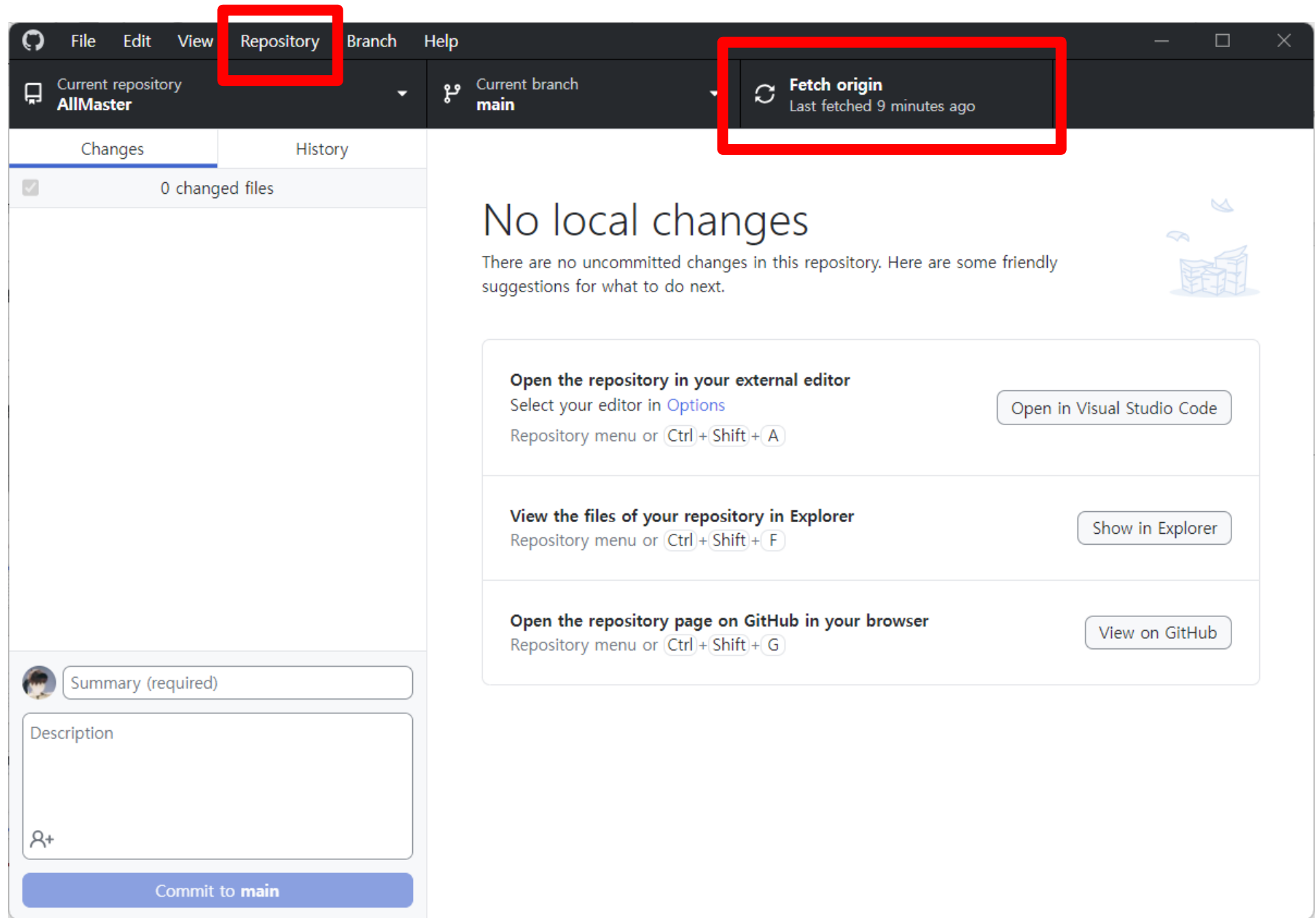
1. 파일을 만들거나 변경하면 Changes가 생기고
 2. 이를 commit해서 변경사항을 기록
 3. Push를 통해 내 변경사항을 원격 저장소에 전달한다.
 - 다른 사람이 Push한 내용을 받아오려면 Fetch를 통해 내역을 조회하고, Pull을 통해 Commit을 로컬로 가져온다.
- 이 과정에서 Branch를 만들어 다른 개발 노선을 만들 수 있다.

Commit

- Repository 선택란
- Changes
 - 변동 파일 목록
- History
 - 커밋 기록
- Commit 작성란



Fetch Push Pull



Branch

- Main(혹은 다른 브랜치)에서 파생된 작업 포인터



Figure 13. 현재 작업 중인 브랜치를 가리키는 HEAD

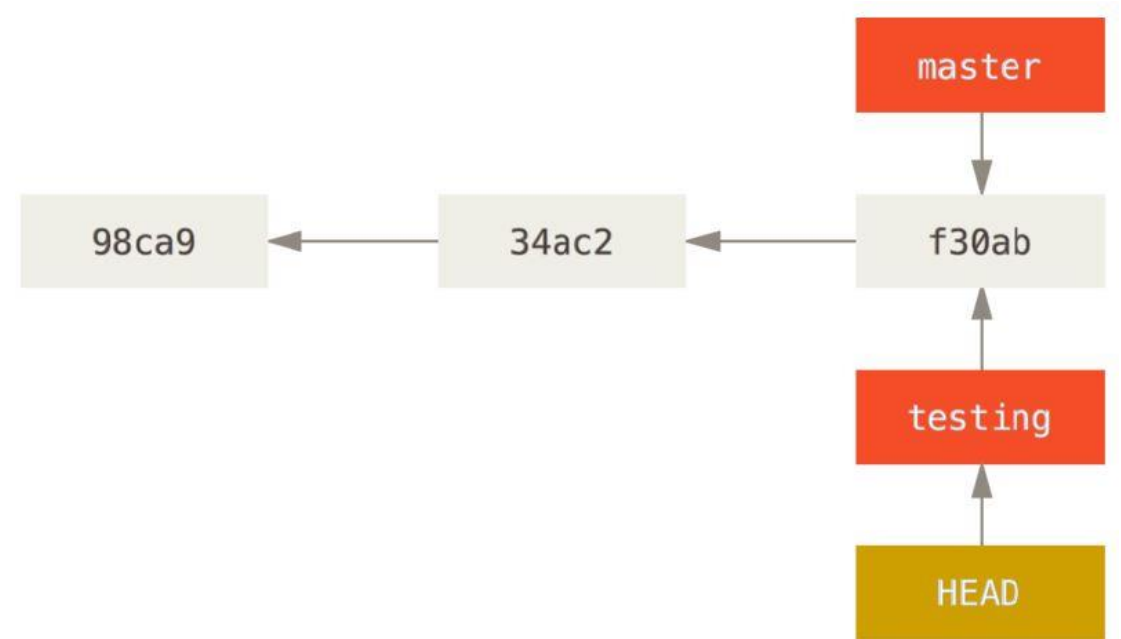
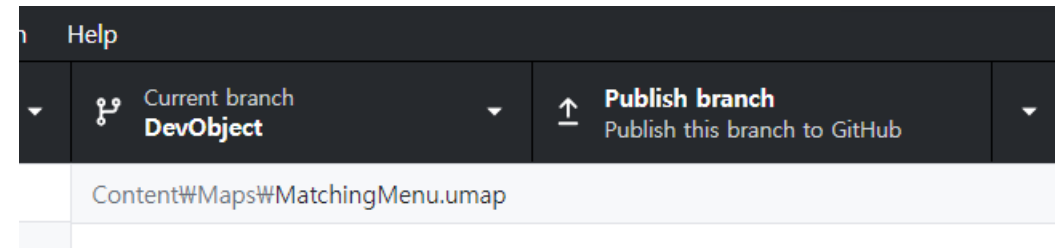
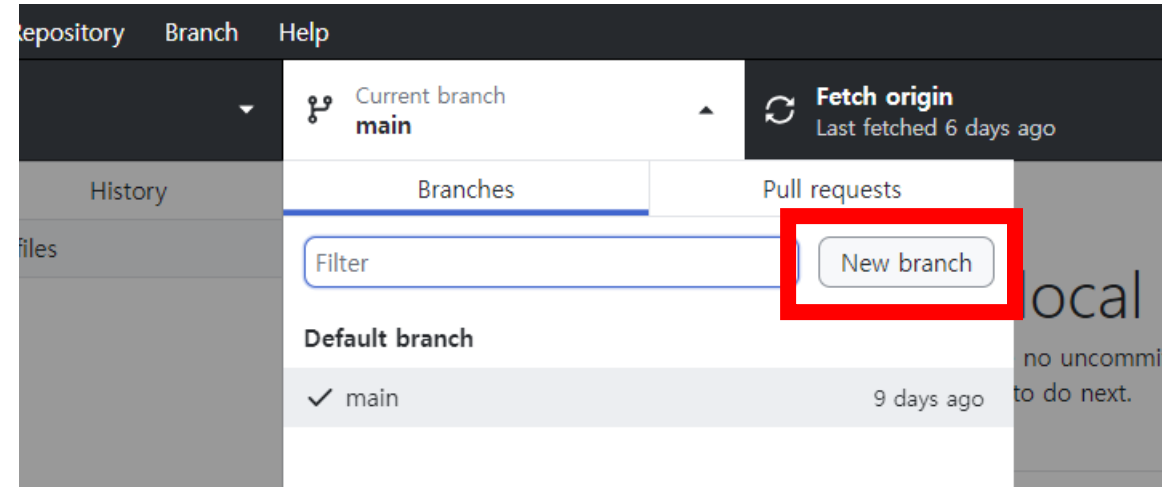


Figure 14. HEAD는 testing브랜치를 가리킴

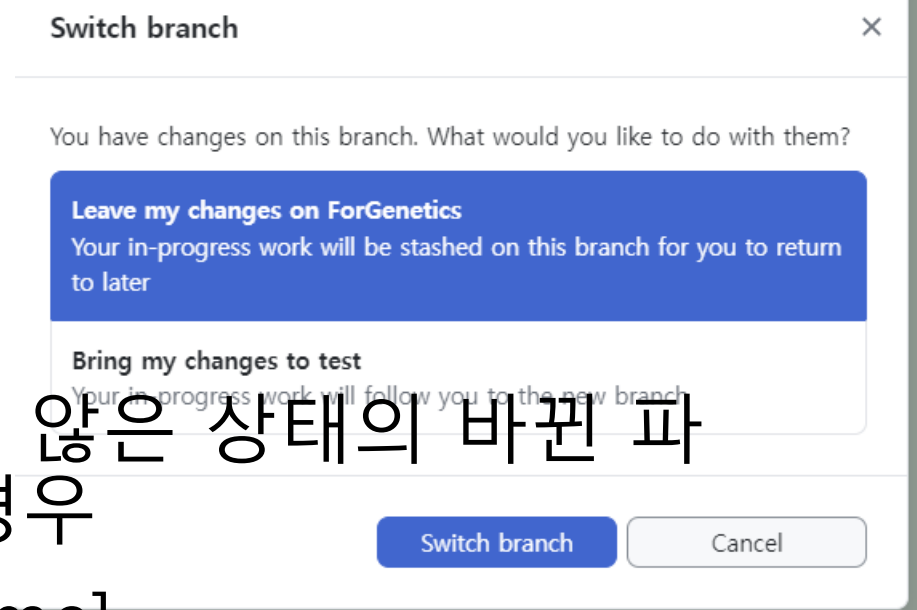
Branch를 만드는 법

- Branch를 만들 때는
 - New Branch를 통해서 만들 수 있다
- Branch를 만들어 기존 main(또는 현재 Branch)에서 파생된 독립적인 작업을 할 수 있다.
- 새로운 기능을 만들기 위한 기능
 - 수정한 파일이 겹쳐 프로젝트가 에러가 나거나, 크래시가 되는 것을 방지
- DevObject라는 Branch를 만들었는데
 - Publish를 해야 다른 사람에게도 보인다.



Switch Branch

- 현재 작업을 하고 있을 때(commit을 하지 않은 상태의 바뀐 파일들이 있을 경우), New Branch를 만들 경우
- Leave my changes on [CurrentBranchName]
 - 현재 변경사항을 기존 branch의 stash(임시 메모리)에 넣어둔다.
 - Commit을 하지 않은 변경사항을 가져오지 않는다.
- Bring my changes to [NewBranchName]
 - 변경사항을 새로 만들 branch에 가져온다.
 - 작업 중이다가, 아 이건 새로운 branch에 넣어야 되겠다. 할 때 사용한다.

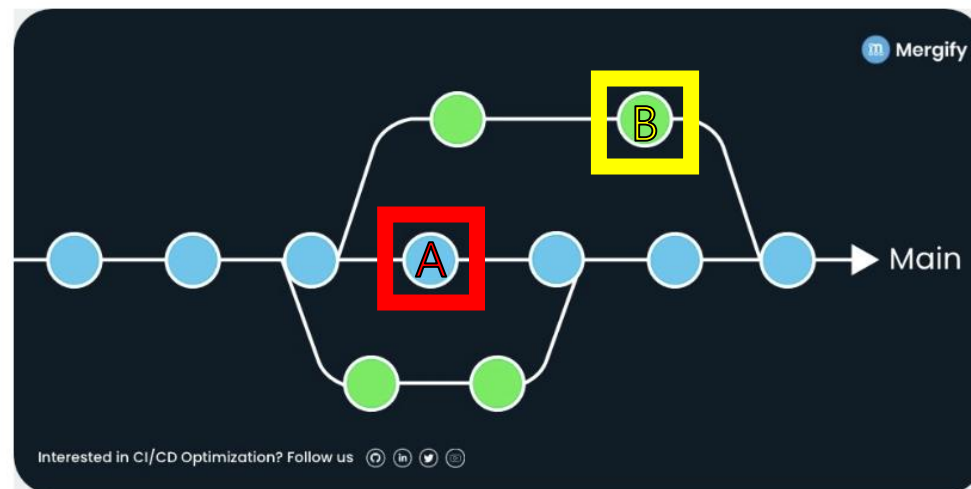


Branch & Merge

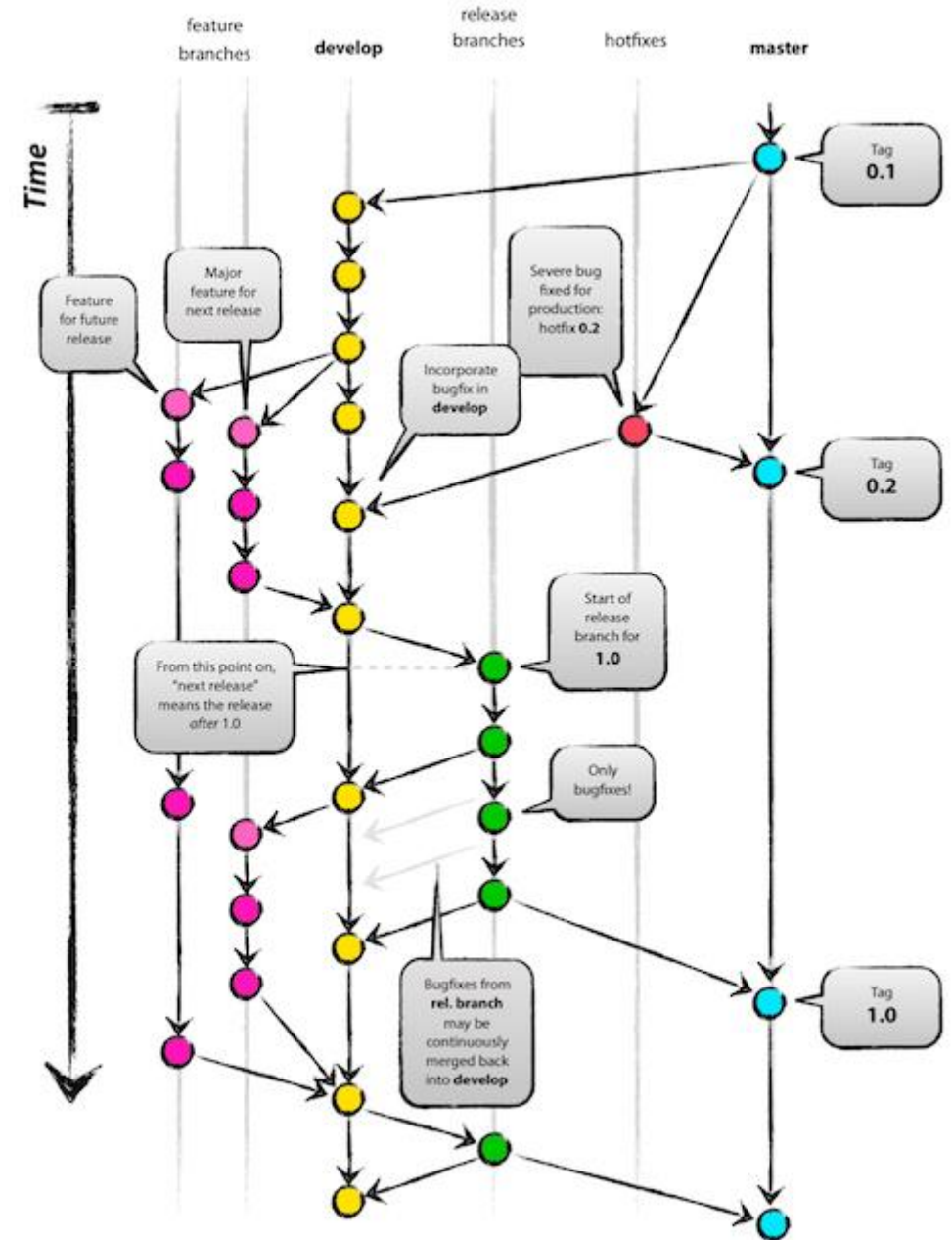
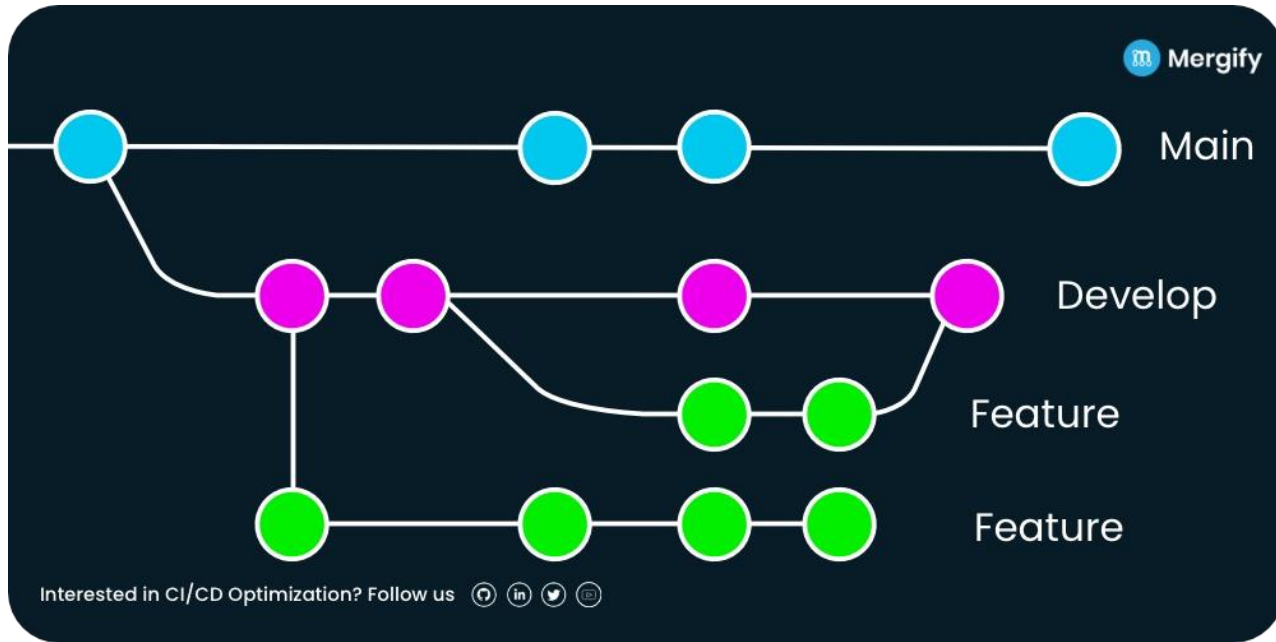
- Merge는 브랜치끼리 합치는 작업
 - 병합 작업
- Branch와 Merge를 잘 써야 협업이 편하더라고요.
 - 왜냐면 누가 어떤 변경사항을 수정했는지 알 수 있고, 어떤 특징의 오류가 나는지 분기점을 알 수 있다.

Merge Conflict

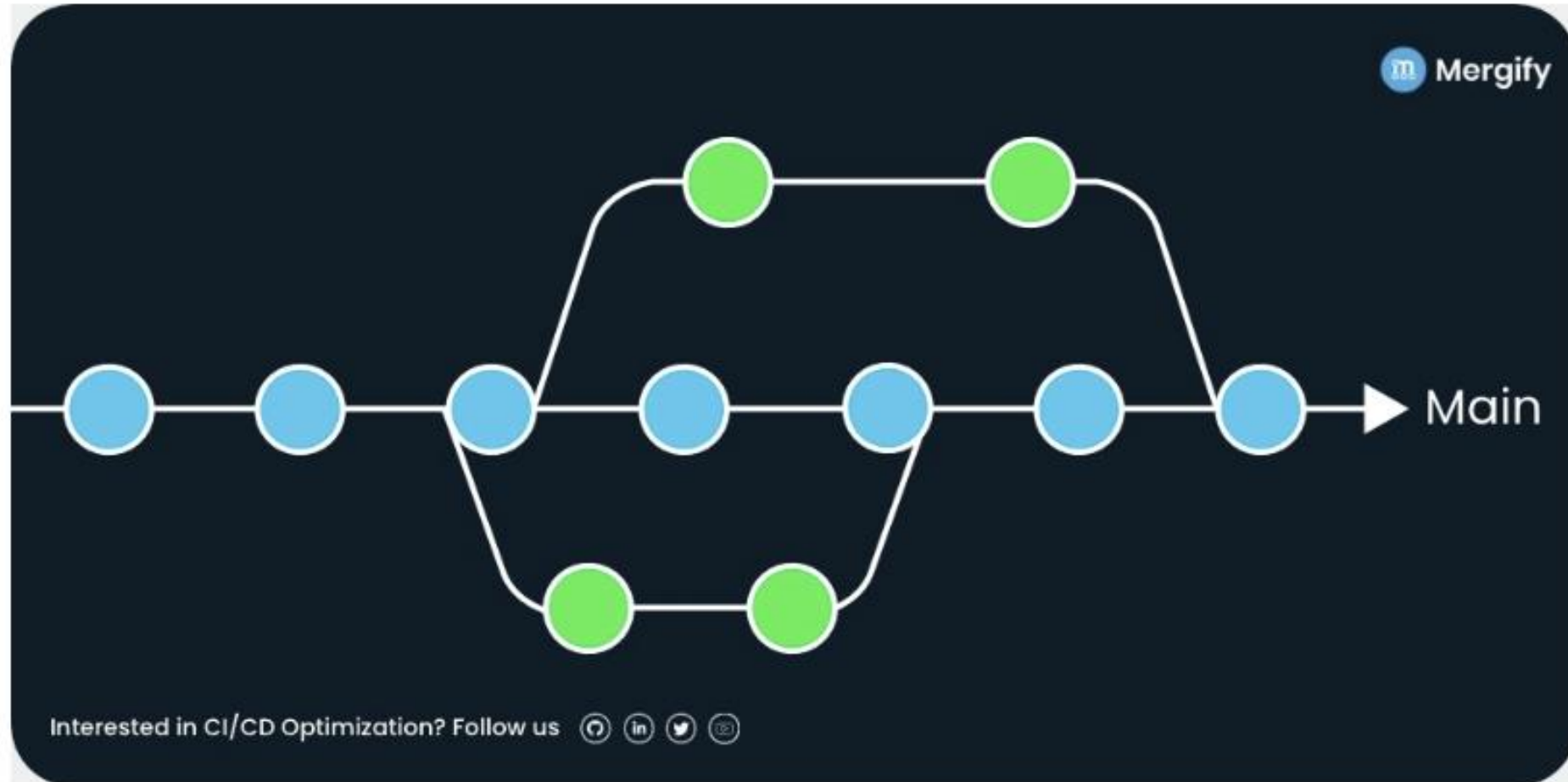
- Main Branch에서 이전에 A를 Commit했는데, B를 Merge할 때, 서로 같은 파일(그 중 같은 부분)을 수정했을 경우이다.
 - 같은 파일에 대한 commit이 충돌되어서 Merge를 하지 못한다.
 - 사용자는 A에서 Commit된 내용을 따를 수 있고
 - B에서 Commit된 내용을 따를 수 있다.
 - 그렇지만 컴퓨터가 알아서 설정할 수 없으므로 사용자가 일일이 지정해줘야 한다.
 - A의 내용을 따르거나
 - B의 내용을 따르거나
 - 아니면 둘 다 따르거나
 - Github Desktop과 vscode로 정해줄 수 있다.



Git flow



Trunk-Based flow



Git Ignore

- 모든 폴더를 공유하지 않게 한다.
- 원하는 파일만 공유할 수 있게 한다
 - 원하지 않는 파일들의 패턴을 작성해서 하는 방법
- 근데.. 어떻게 써요?

Git ignore를 만들자

- gitignore.io를 사용하자
 - <https://www.toptal.com/developers/gitignore>
 - Unity 태그를 넣어 생성을 누르면...

gitignore.io

자신의 프로젝트에 꼭 맞는 .gitignore 파일을 만드세요

Unity × |

생성

[소스 코드](#) | [커맨드라인 문서](#)

Git ignore를 만들자

- 이런 페이지가 나온다.
 - 이걸 전체선택-복사를 해서 파일명을 .gitignore로 만들어서 이 내용을 붙여넣는다.
 - .git 폴더(숨긴 폴더)가 있는 곳에 넣을 것.
 - 메모장이나 ,vscode같은 편집기 이용
- 도는 GitHub desktop에 Repository
-Repository Setting
-Ignored files
에 붙여 넣어도 된다.

```
# Created by https://www.toptal.com/developers/gitignore/api/unity
# Edit at https://www.toptal.com/developers/gitignore?templates=unity

### Unity ###
# This .gitignore file should be placed at the root of your Unity project directory
#
# Get latest from https://github.com/github/gitignore/blob/main/Unity.gitignore
/[Ll]ibrary/
/[Tt]emp/
/[Oo]bj/
/[Bb]uild/
/[Bb]uilds/
/[Ll]ogs/
/[Uu]ser[Ss]ettings/

# MemoryCaptures can get excessive in size.
# They also could contain extremely sensitive data
/[Mm]emoryCaptures/

# Recordings can get excessive in size
/[Rr]ecordings/

# Uncomment this line if you wish to ignore the asset store tools plugin
# /[Aa]ssets/AssetStoreTools*

# Autogenerated JetBrains Rider plugin
/[Aa]ssets/Plugins/Editor/JetBrains*

# Visual Studio cache directory
.vs/

# Gradle cache directory
.gradle/
```

Repository settings

Remote

Ignored files

Git config

Editing .gitignore. This file specifies intentionally untracked files that Git should ignore. Files already tracked by Git are not affected. [Learn more about gitignore files](#)

```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so
```

Save

Cancel

.git	2023-1
.vs	2023-1
Binaries	2023-1
Config	2023-1
Content	2023-1
DerivedDataCache	2023-1
Intermediate	2023-1
Saved	2023-1
Script	2023-1
Source	2023-1
.editorconfig	2023-1
.gitattributes	2023-1
.gitignore	2023-1

직접 Ignore을 커스텀하자

- Ignore 파일을 보면 규칙이 있다
 - * : 모든 문자열 이라는 뜻
 - EX) *.exe : 모든 exe 파일(실행 파일)
 - / : 폴더라는 뜻
 - /[L]ibrary/
 - : .gitignore 파일이 있는 폴더에서 Library 폴더 혹은 library 폴더 뒤에 있는 파일 전체
 - Git ignore 파일에 파일 패턴이 들어가면 해당 파일이 추가/변경/삭제가 되어도 git에 기록되지 않는다.
 - 참고용: <https://nemomemo.tistory.com/78>

실제 내 프로젝트 Git Ignore 커스텀

- 유료 에셋을 github에 올릴 수 없으니 이를 제외하기 위해서 추가
- Unreal Ignore에 새로 한줄 추가
 - Content/ModularRPGHeroesPolyart/*
 - 유료 에셋 경로를 입력해 해당 폴더 아래의 모든 파일을 무시.

```
66 # Configuration files generated by the Editor
67 Saved/*
68
69 # Compiled source files for the engine to use
70 Intermediate/*
71 Plugins/*/Intermediate/*
72
73 # Cache files for the editor to use
74 DerivedDataCache/*
75
76 #Charged Asset
77 Content/ModularRPGHeroesPolyart/*
```

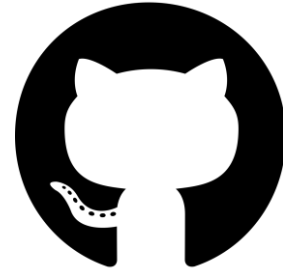
My GitHub

SeoPangs/AllMaster



AllMaster

Public



AllMaster GitHub Link