



# Git 101 for Unity

# About Me

Gregory Horvath

Profession:

- Technical Product Manager (software)

Indie Game Dev:

- [Game Shenanigans](#)
- [Pandimensional Studios](#)



# Introduction

An end-to-end workflow:

- Setup your repo
- Plan and manage your project
- Never lose your progress
- Continuous Integration
- Continuous Deployment

# Scope

- Pain points when getting started with Git
- Use cases for Git, GitHub, GameCI
- End-to-end workflow
- NOT Code / Design

# What tools will we use?



[Unity 2021 LTS](#)



[VS Code](#)



[Git 2.40](#)



[GitHub](#)



[GameCI](#)

## Alternatives

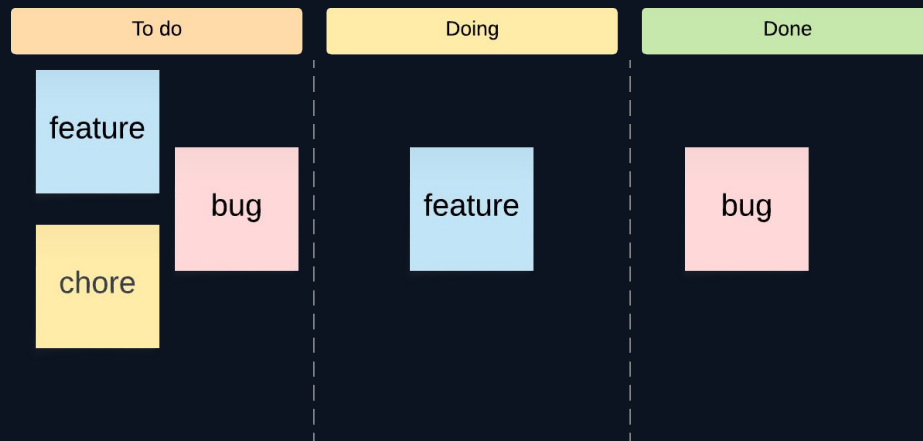
- Unity DevOps♥
- PlasticSCM
- Azure♥ / GCP / AWS
- Gitlab
- CircleCI
- Atlassian

# Pain Points

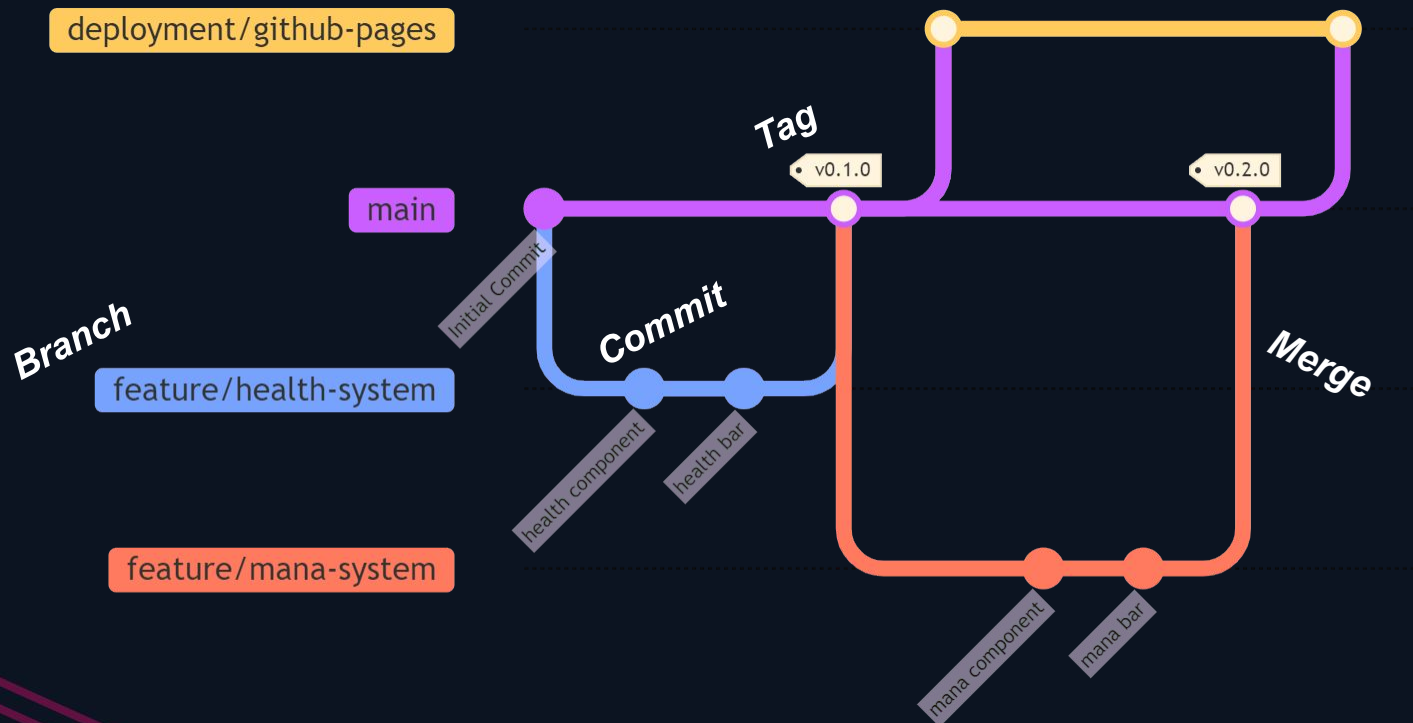
1. Private Repos to protect licensed assets
2. Authentication and Signed Commits
3. Untracking files (binaries, logs, local configuration)
4. Dealing with large files

# Project Planning and Management

- Consult GPT for a plan
- Setup KanBan board in GitHub
- Protect main branch
- Tackle first issue



# Example Git Branching Structure





# Why CI/CD?

Faster release cycles - bring changes to market faster

Improve code quality - automate tests and builds

Collaboration - team is frequently working on the latest code

Reduce Risk - Rollback / Fail Forward

# Why Next?

Automate testing

Automate Steam build / deploy

Self-Hosted Runner (AWS, GCP, etc.) for extra power

Make an awesome game!

# Summary

- Setup a Git repo for our Unity Project
- Synced the repo with GitHub
- Replaced our project manager with GPT4
- Demonstrated a developer workflow
- Added WebGL auto build and deploy
- Created Pong

Thank You!

Gregory Horvath

Game Shenanigans

[gphorvath@gameshenanigans.com](mailto:gphorvath@gameshenanigans.com)

<https://github.com/gphorvath/GameDevGuildConference2023>