Game Design & Development
Unity Git Workflow – Lecture 5

Michael Holly

Game Recap

… Who has played Line Rider?

   Chat: [I]

Game Recap

… What is the main goal of the game …?

　　　Chat: …

Game Recap

… What are the main game design elements …?

Chat: …

How to implement Line Rider in Unity?

… Any ideas?

How to implement Line Rider in Unity?

[(2) How to make a LINE RIDER Replica in Unity (Livestream Tutorial) - YouTube](#)

# How to manage your Unity Project

1. Directory Structure

2. Scene hierarchy structure

3. Prefabs for everything

4. Version Control System (GIT)

Managing Software Versions

- Use stable versions
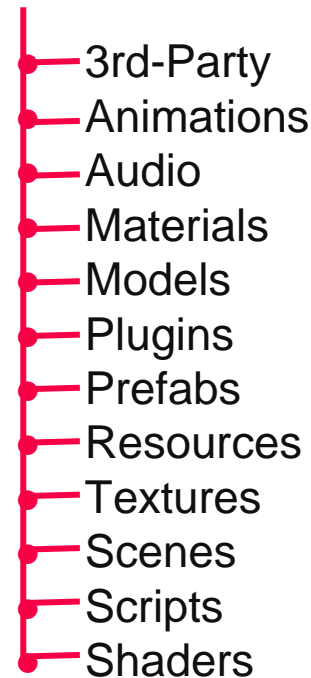- Plan when to update to which version

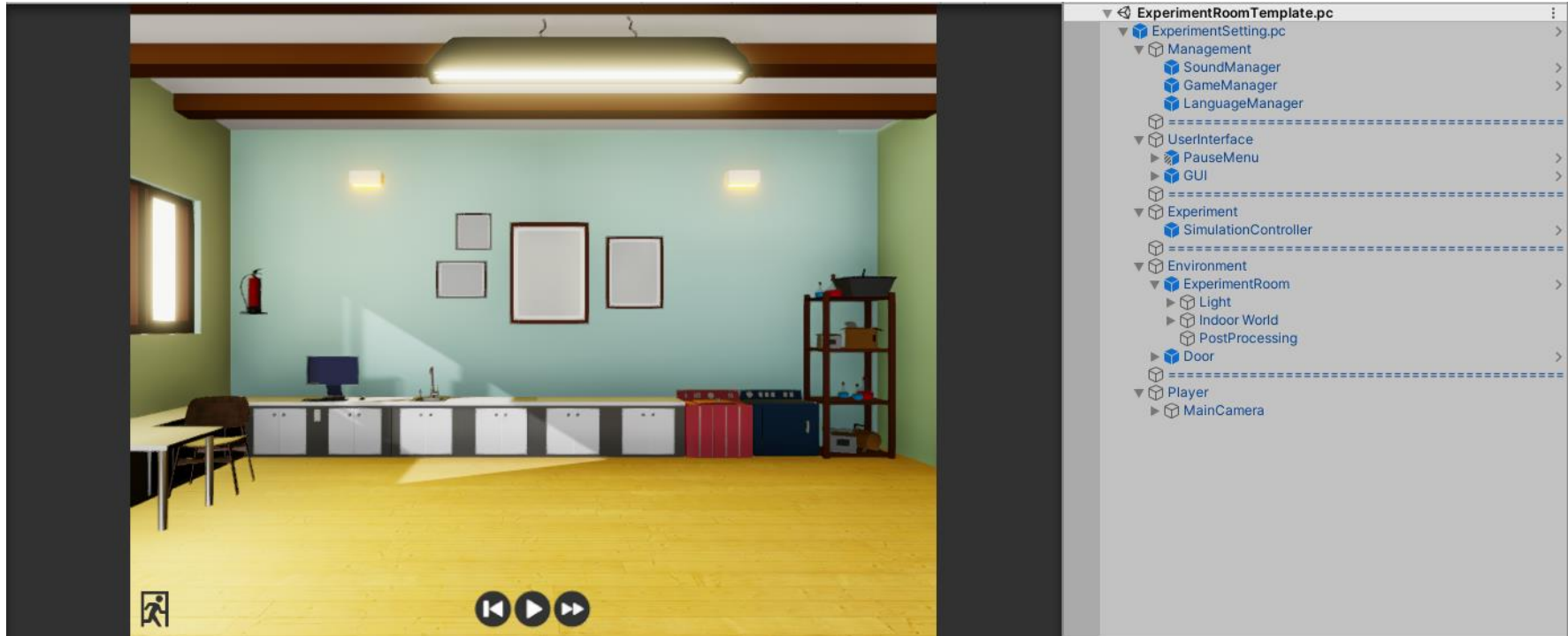| Name	Version	Download
SteamVR	1.2.3	SteamVR Plugin
VRTK	3.3.0	VRTK Plugin
HTC Vive		HTC Vive
Oculus		Oculus Integration
Samsung Gear VR		Samsung Gear VR

Unity Upgrade Plan
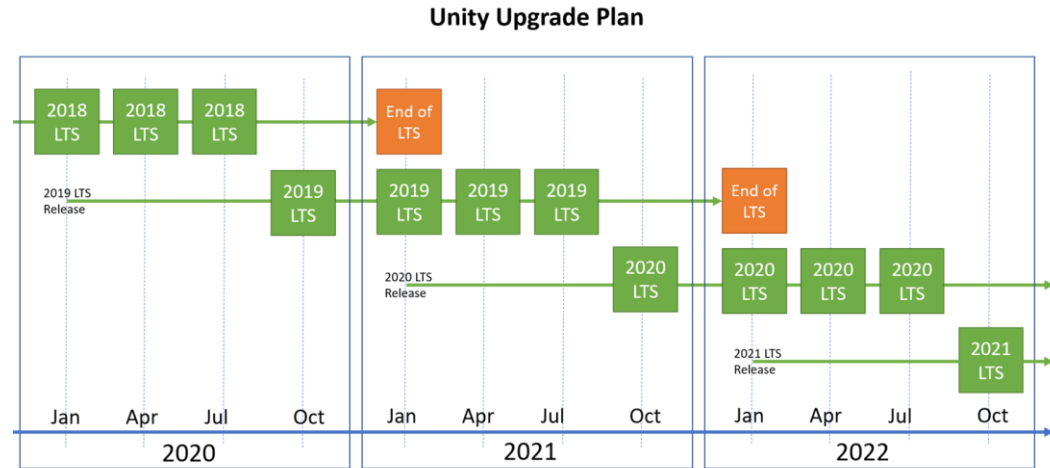
Directory Structure

- Do not store any asset files in the root directory

- Do not store any additional directories in the root directory

- Be consistent with naming

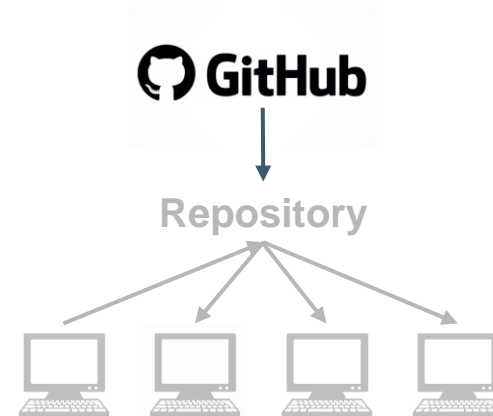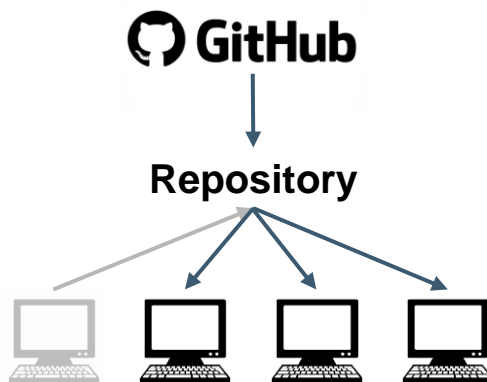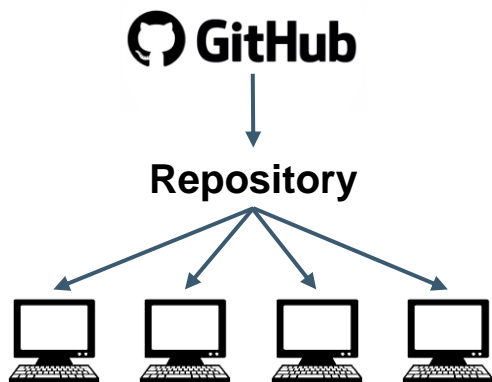- Use 3rd-Party folder to store imported assets

- 3rd-Party
- Animations
- Audio
- Materials
- Models
- Plugins
- Prefabs
- Resources
- Textures
- Scenes
- Scripts
- Shaders

Scene Hierarchy Structure

Version Control System - GIT

- Tracks the changes you make to files
- Github - to host and control your project using Git
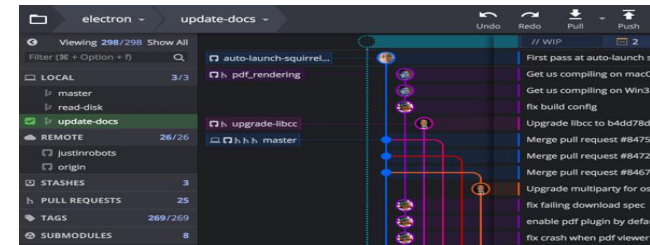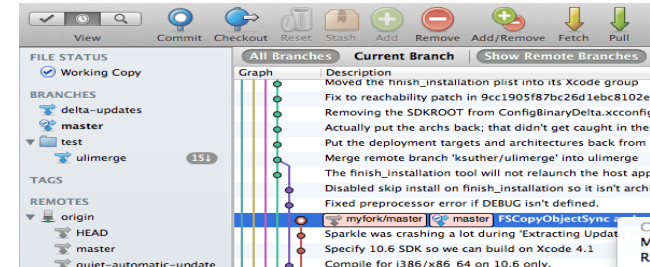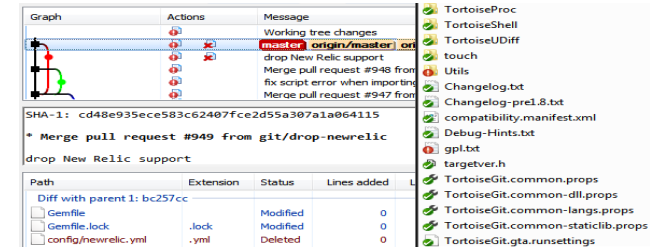- Repository to store your project

Hho uses the commandline for GIT?

Who uses a UI for GIT?

Git Desktop Client

- Github Desktop

- TortoiseGit

- SourceTree

- GitKraken

Git Large File Storage

```
git lfs install

git lfs track "*.mp4"
```

.gitattributes

```
## git-lfs ##
#Image
*.jpg filter=lfs diff=lfs merge=lfs -text
*.png filter=lfs diff=lfs merge=lfs -text
#Audio
*.mp3 filter=lfs diff=lfs merge=lfs -text
*.wav filter=lfs diff=lfs merge=lfs -text
#Video
*.mp4 filter=lfs diff=lfs merge=lfs -text
*.mov filter=lfs diff=lfs merge=lfs -text
```
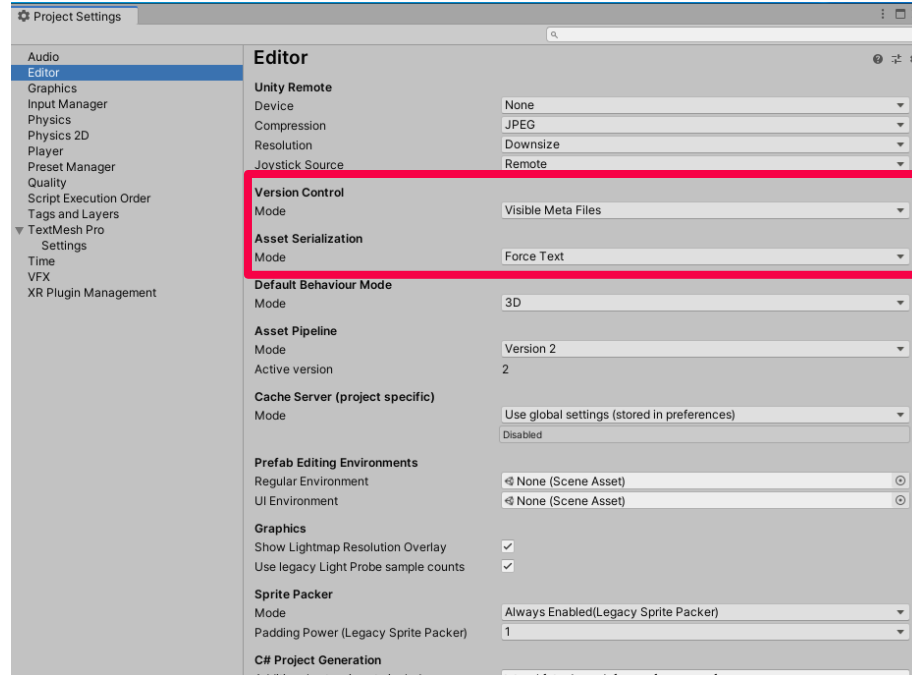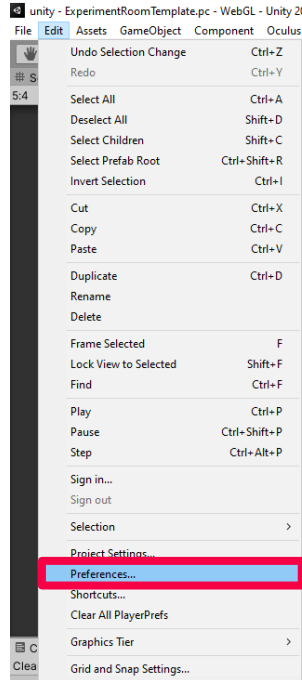
Setting Up Unity

Setting Up GitHub

Repository Permissions

- Read

- Write

- Maintain

- Admin

Project Workflow

GAME LAB GRAZ

Project Workflow

Github Wiki

- Documentation for your repository

- Share long-form content about your project

- Different to Readme

- Edit on GitHub or locally

General
🏠 Home
📑 Software versions

Workflow
🔗 Git workflow
⚠️ Merge conflicts
⁉️ Issues and Progress

Conventions
💬 Naming things
📂 Folder structure
📦 Including components

📄 Scripts
🌄 Scenes
🎨 Materials
3D Models
🔤 Typography

Documentation
Generated laboratory

Misc
💡 Ideas

Issues

GAME LAB GRAZ

Issue: Defect Report

Create a report to help us improve. If this doesn't look right, choose a different type.

Title

Write | Preview

Describe the defect
A clear and concise description of what the bug is.

To Reproduce
Steps to reproduce the behavior:

1. Go to '...'
2. Click on '....'
3. Scroll down to '....'
4. See error

Expected behavior
A clear and concise description of what you expected to happen.

Found in version
The version where the defect was detected [e.g. v4.1 build 20210224]

Screenshots
If applicable, add screenshots to help explain your problem.

Desktop (please complete the following information):

- OS: [e.g. Windows]
- Version [e.g. 22]

WebGL(please complete the following information):

- OS: [e.g. Windows]
- Browser [e.g. Chrome]
- Version [e.g. 22]

VR(please complete the following information):

- Device: [e.g. HTC Vive]

Assignees
No one—assign yourself

Labels
defect

Projects
None yet

Milestone
No milestone

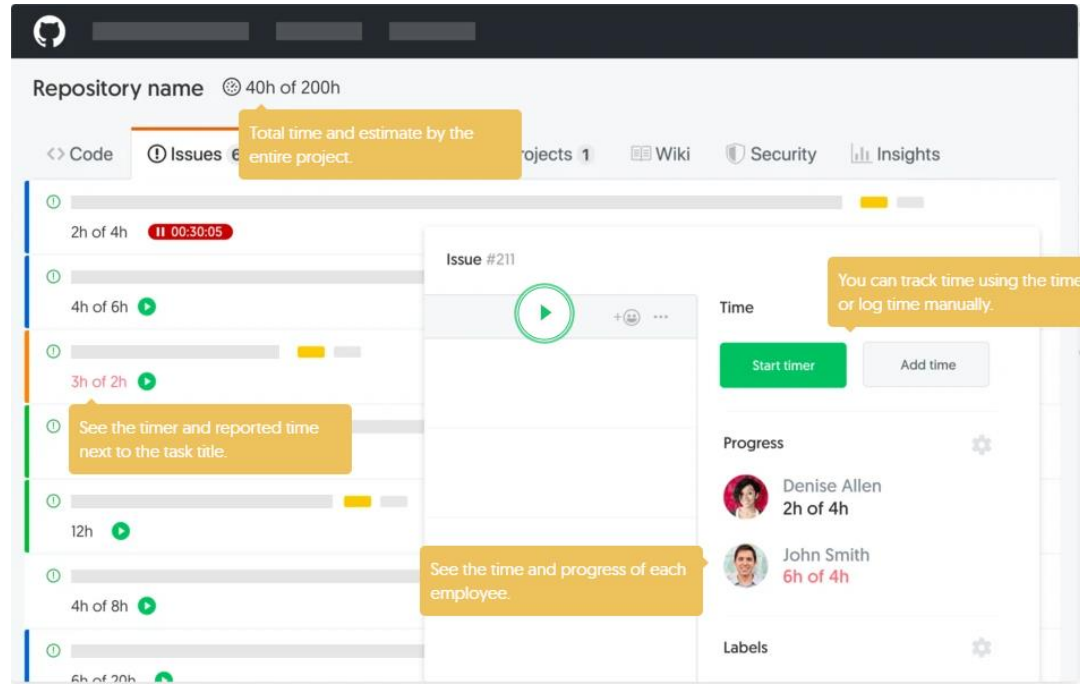**Development**
Shows branches and pull requests linked to this issue.

Helpful resources
GitHub Community Guidelines

Time Tracking

GAME LAB GRAZ

[GitHub Time Tracker and Estimates for GitHub Issues (everhour.com)](#)

Git Workflow
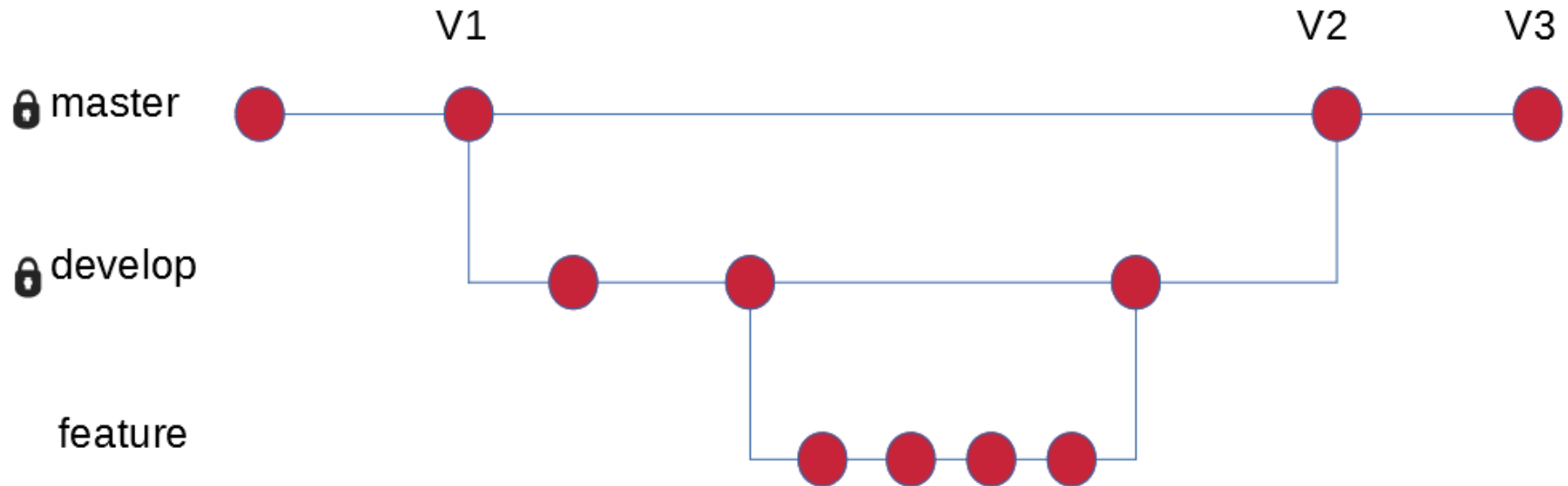
Git Workflow

- Protected branches: master/main, develop

- All commits must be made to a non-protected branch

- Separate branch for each feature

Branch Protection Rules

- Require pull request reviews before merging

- Dismiss stale pull request approvals when new commits are pushed

- Restrict who can dismiss pull request reviews

- Restrict who can push to matching branches

Feature Branches

- feature/<project_name>-<issue-number>_<short issue description>

- Commit Message:
  - <project_name>-<issue-number>_<short issue description>
  - Empty line
  - <Description>

- Clean Commit History
  - git commit -amend
  - git squash

Feature Branches

1. Create new feature branch
   git checkout -b myFeature develop

1. Commit your changes
   git commit -m "my changes"

1. Amend your changes for a clean history
   git commit -amend

1. Push your feature branch
   git push

1. Create Pull Request on Github

Pull Requests

GAME LAB GRAZ

Pull Requests - Reviews

GAME LAB GRAZ

Merge Conflicts

- Meta files must be visible

- Plain text serialization must be active

- Do not allow an automatic merge for scenes and prefab files
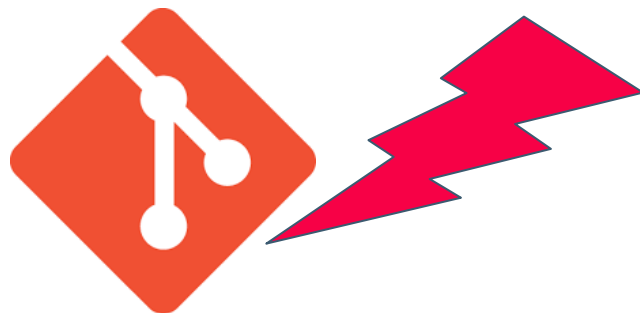
Unity - Manual: Smart Merge (unity3d.com)

Merge Process

1. Git merge is initiated
   `git merge develop`
1. A merge conflict occurs
2. Developer starts Unity yaml merge tool
   `git mergetool --tool=unityyamlmerge`
1. Unity yaml merge tool automatically opens another merge tool if it cannot resolve all conflicts
2. Developer resolves conflicts and saves merged file
3. Results can be inspected and the merge can be completed

Unity Packages

- manifest.json

```
{
  "dependencies": {
    "com.gear.languagemanager": "https://github.com/GameLabGraz/Utilities.git#package/languagemanager",
    "com.gear.serializeproperty": "https://github.com/GameLabGraz/Utilities.git#package/serializeproperty",
    "com.unity.2d.sprite": "1.0.0",
    "com.unity.2d.tilemap": "1.0.0",
```

- packages-lock.json

```
{
  "dependencies": {
    "com.gear.languagemanager": {
      "version": "https://github.com/GameLabGraz/Utilities.git#package/languagemanager",
      "depth": 0,
      "source": "git",
      "dependencies": {
        "com.unity.textmeshpro": "1.4.1"
      },
      "hash": "b9ede28c705f209d7b833de67b7a77d9ffb6cfb8"
    },
    "com.gear.serializeproperty": {
      "version": "https://github.com/GameLabGraz/Utilities.git#package/serializeproperty",
      "depth": 0,
      "source": "git",
      "dependencies": {},
      "hash": "52b5f5d526bbd170a699f009a832b1d7cf02328a"
    },
```

Git Subtrees

- Management of simple workflow is easy

- User can ignore that you are using git subtree to manage dependencies.

- Content of the module can be modified without having a separate repository copy of the dependency somewhere else.

Publishing Unity Packages on Git Repository

```
git subtree push --prefix
<package path> origin package/<package name>
```

origin/develop

origin/package/

Automatic Unity Build Uploader