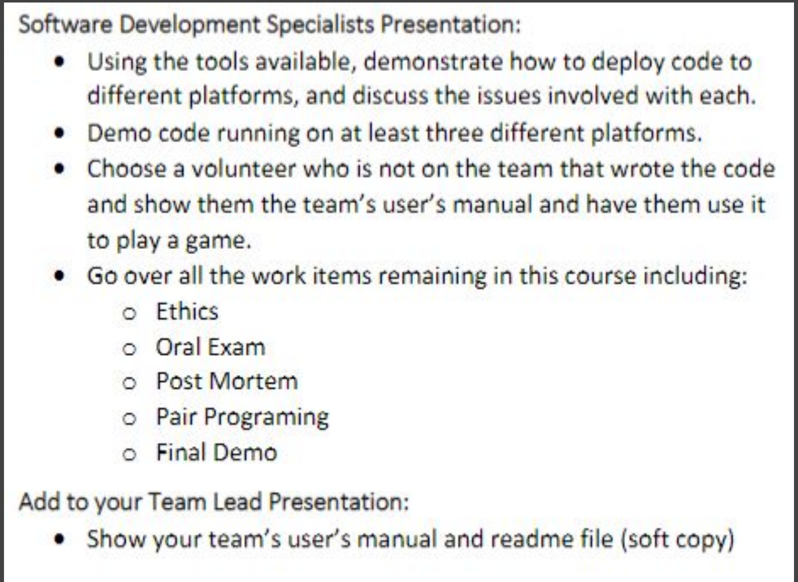
- How to deploy code to different platforms - Kadence
    - Problems that may come up with each platform
- Demo code running on at least three platforms - Gage, Kadence, Hunter, Seth?
    - Windows, Linux, Android?
        - Run Windows game on Gage’s laptop
        - We can run the Linux games on Seth’s laptop
        - Android: Must have motion controls on one game - MM movement?
        - **Must show deployment to Android process**
- Share github releases link with Android release
- Git branch (Desktop & command line?) - Seth
    - git branch
    - git switch [-c]
- Remaining items in course
    - Ethics (Gage)
    - Oral Exam (Gage)
        - The Thursday before Thanksgiving Break, except CDA can go either Wed or Fri that week.
        - 30 Points from the midterm have been moved to the oral exam
        - **Pre-Work Update (Thursday Presentation)**
    - Post Mortem (Hunter)
    - Pair Programming (Hunter)
        - **This is not until after Thanksgiving Break (Up until oral exams, your code can only be edited by you)**
    - Final Demo (Hunter)
- Show team’s user’s manual & readme file (soft copy)
- More git stuff?
    - Tags, semantic versioning (vX.X.X)



Software Development Specialists Presentation:

- Using the tools available, demonstrate how to deploy code to different platforms, and discuss the issues involved with each.
- Demo code running on at least three different platforms.
- Choose a volunteer who is not on the team that wrote the code and show them the team’s user’s manual and have them use it to play a game.
- Go over all the work items remaining in this course including:
    - Ethics
    - Oral Exam
    - Post Mortem
    - Pair Programing
    - Final Demo

Add to your Team Lead Presentation:

- Show your team’s user’s manual and readme file (soft copy)

For this class

- Find a pair partner
- Pick a pair activity
- Make a meaningful contribution: 2-3 hours max
  - Get the scrum master’s permission before you submit it.
- Choose someone from your pair to write up a short (max 300 words) report of what you did and submit it to BBLearn.

Version Control

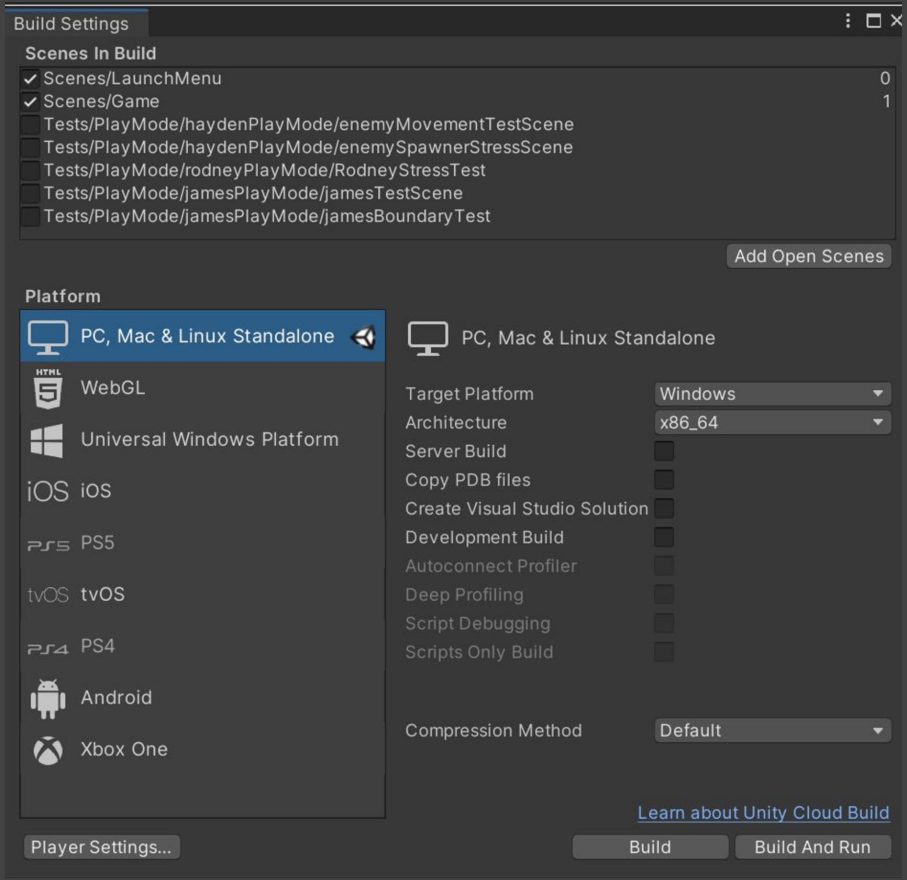
TL6
Gage, Hunter, Seth, Kadence

Deploying to Different Platforms from Unity

Windows, Xbox, Android

Windows

- File > Build Settings > PC, Mac, & Linux Standalone
- Target Platform > Windows
- Architecture > x86_64
- Build > Select File Location
- Unity will create a game executable of the build,
- and a .data file
- Compress the two files with a zip file to share



Build Settings

Scenes In Build

☑ Scenes/LaunchMenu 0
☑ Scenes/Game 1
□ Tests/PlayMode/haydenPlayMode/enemyMovementTestScene
□ Tests/PlayMode/haydenPlayMode/enemySpawnerStressScene
□ Tests/PlayMode/rodneyPlayMode/RodneyStressTest
□ Tests/PlayMode/jamesPlayMode/jamesTestScene
□ Tests/PlayMode/jamesPlayMode/jamesBoundaryTest

Platform

PC, Mac & Linux Standalone

Target Platform: Windows
Architecture: x86_64
Server Build □
Copy PDB files □
Create Visual Studio Solution □
Development Build □
Autoconnect Profiler □
Deep Profiling □
Script Debugging □
Scripts Only Build □

Compression Method: Default

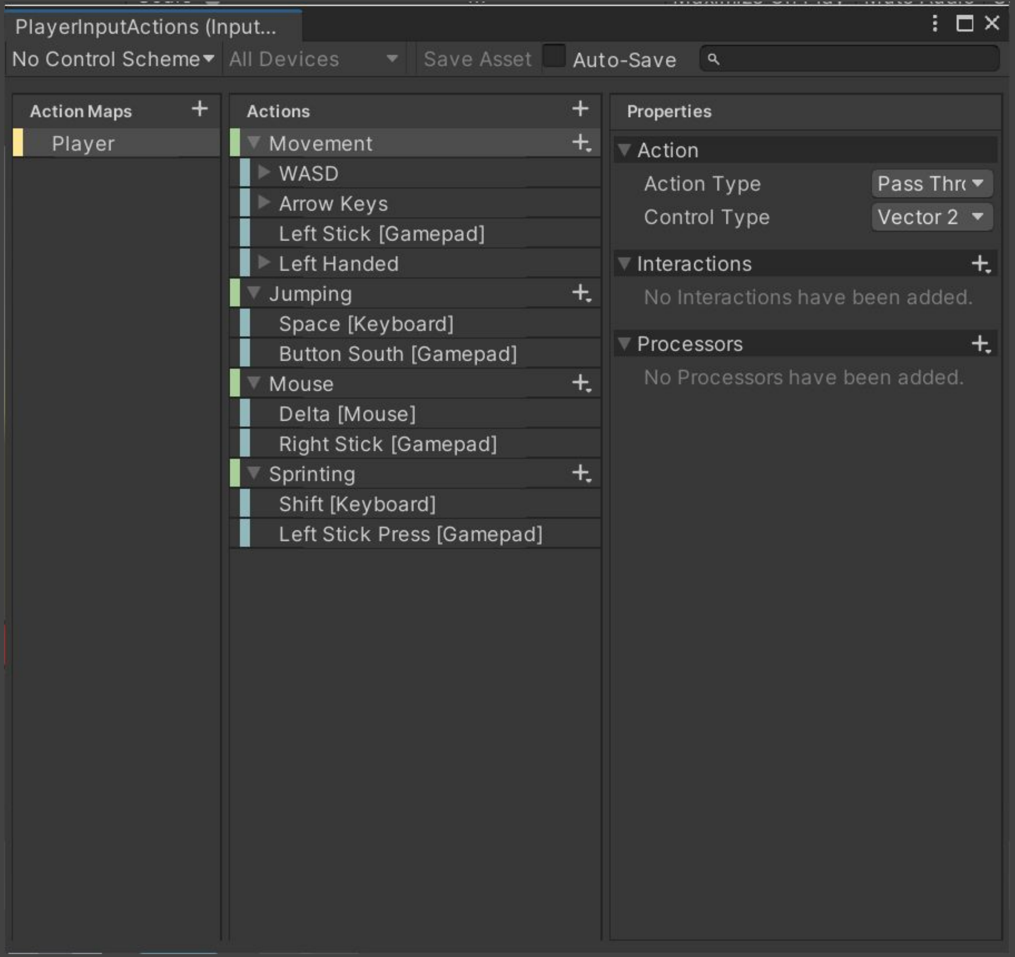
Xbox Controller

Step 1: Add Input Action for Gamepad

Step 2: Connect controller with USB to micro USB cable.

Step 3: Unity to Xbox One with Universal Windows Platform

- Become a developer with Microsoft Partner Center



PlayerInputActions (Input...

No Control Scheme ▾ | All Devices ▾ | Save Asset | □ Auto-Save

Action Maps +
- Player

Actions +
- ▼ Movement +
  - ▶ WASD
  - ▶ Arrow Keys
  - Left Stick [Gamepad]
  - ▶ Left Handed
- ▼ Jumping +
  - Space [Keyboard]
  - Button South [Gamepad]
- ▼ Mouse +
  - Delta [Mouse]
  - Right Stick [Gamepad]
- ▼ Sprinting +
  - Shift [Keyboard]
  - Left Stick Press [Gamepad]

Properties

▼ Action
- Action Type: Pass Thr ▾
- Control Type: Vector 2 ▾

▼ Interactions +

No Interactions have been added.

▼ Processors +

No Processors have been added.

Android … but first:

git for Version Control

Version Control

- We’ve been using it all along
- It involves keeping a record of file history, that is, keeping track of *versions* of your code

git workflow

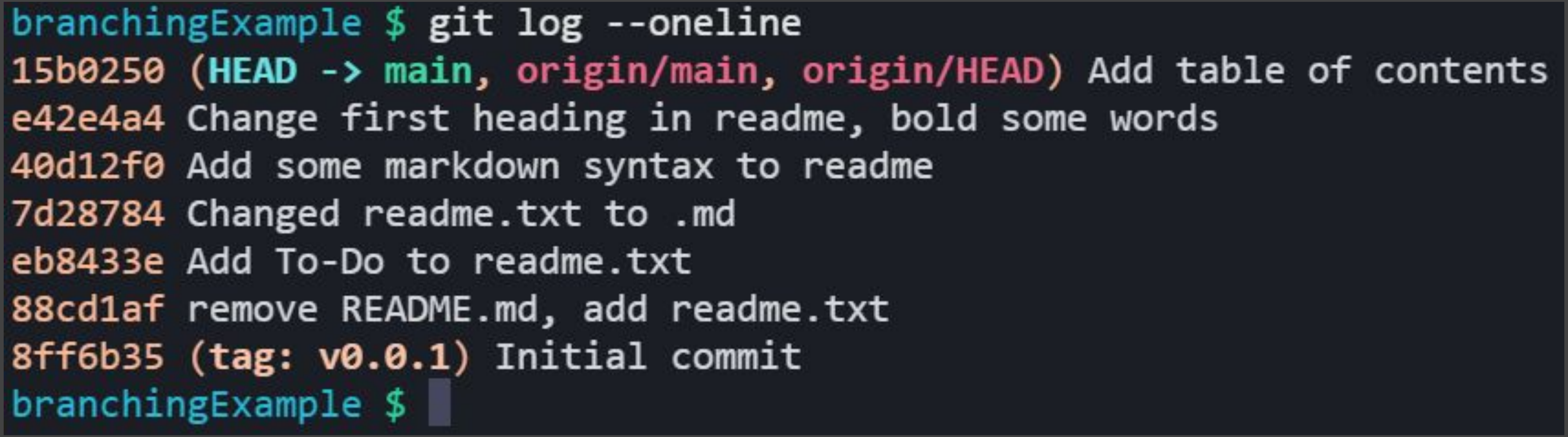
```
- git init
- git status
- git add <files with changes ready to commit>
- git commit -m "Commit Message"
```

At this point, a snapshot of the code at the time of the commit has been saved.

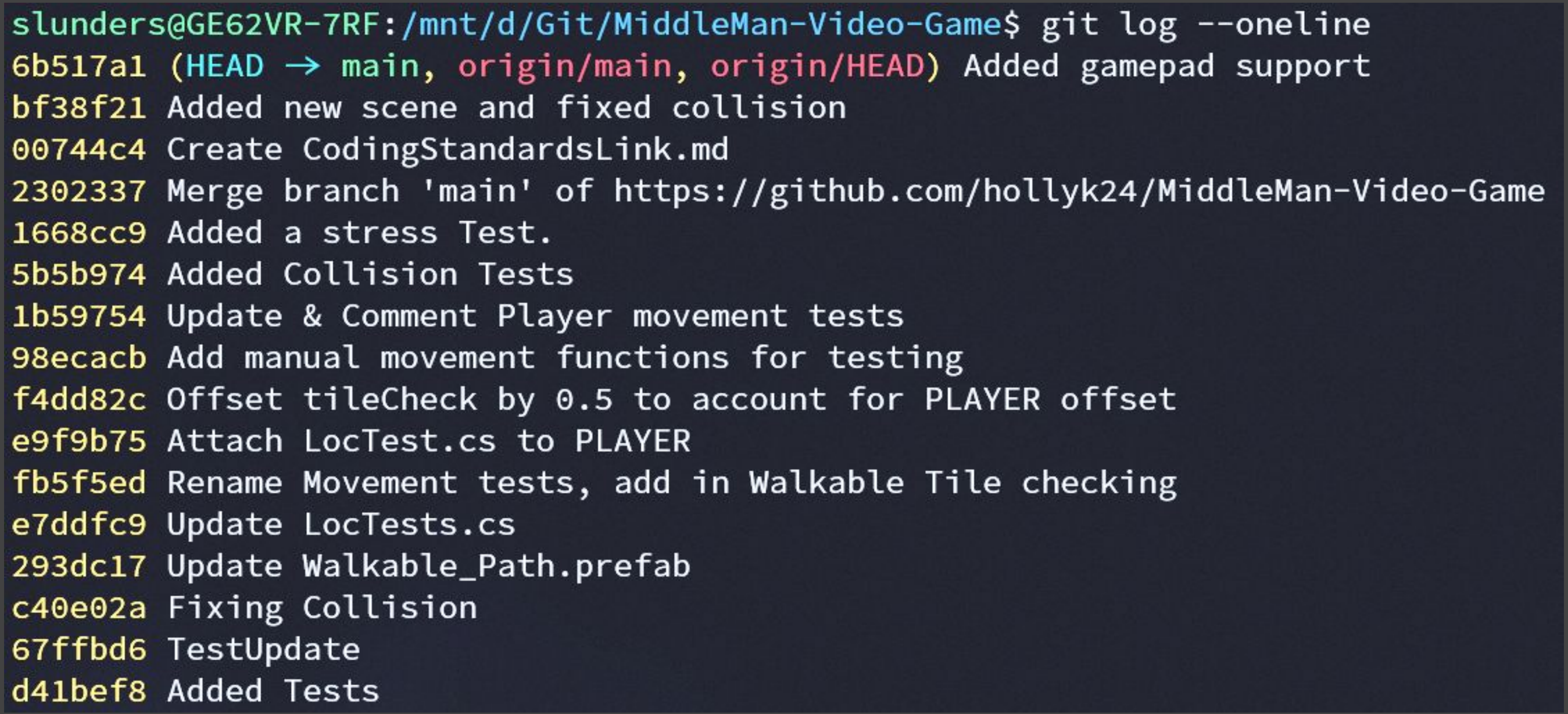
git + GitHub

```
- git clone <repo url>
- git fetch
- git pull
- git push
```

git log [--oneline]



```
branchingExample $ git log --oneline
15b0250 (HEAD -> main, origin/main, origin/HEAD) Add table of contents
e42e4a4 Change first heading in readme, bold some words
40d12f0 Add some markdown syntax to readme
7d28784 Changed readme.txt to .md
eb8433e Add To-Do to readme.txt
88cd1af remove README.md, add readme.txt
8ff6b35 (tag: v0.0.1) Initial commit
branchingExample $
```



```
slunders@GE62VR-7RF:/mnt/d/Git/MiddleMan-Video-Game$ git log --oneline
6b517a1 (HEAD -> main, origin/main, origin/HEAD) Added gamepad support
bf38f21 Added new scene and fixed collision
00744c4 Create CodingStandardsLink.md
2302337 Merge branch 'main' of https://github.com/hollyk24/MiddleMan-Video-Game
1668cc9 Added a stress Test.
5b5b974 Added Collision Tests
1b59754 Update & Comment Player movement tests
98ecacb Add manual movement functions for testing
f4dd82c Offset tileCheck by 0.5 to account for PLAYER offset
e9f9b75 Attach LocTest.cs to PLAYER
fb5f5ed Rename Movement tests, add in Walkable Tile checking
e7ddfc9 Update LocTests.cs
293dc17 Update Walkable_Path.prefab
c40e02a Fixing Collision
67ffbd6 TestUpdate
d41bef8 Added Tests
```

$ git log --oneline

MM Repo

Working with Branches

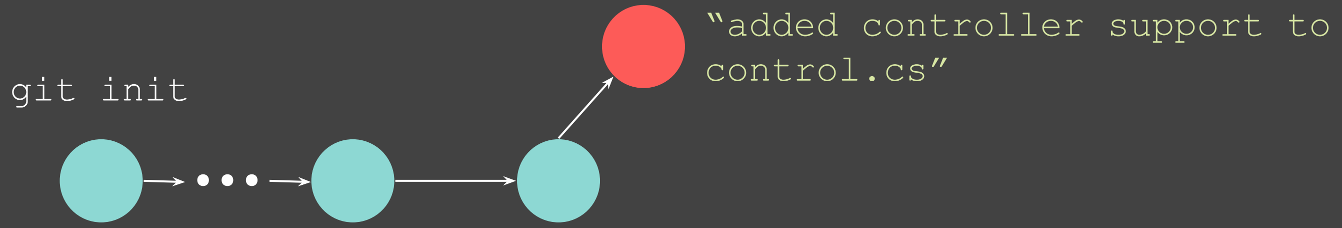
```
git branch
git switch [-c]
git merge
```

git branch

● main

● controllerSupport

```
- git branch   # List existing branches
- git switch -c controllerSupport    # Create & Switch
      ...(edit controls.cs file to support controller)...
- git add controls.cs
- git commit -m “added gamepad support to controls.cs”
- git push -u origin controllerSupport   # Let github know there’s a new branch
```



git merge

- git switch controllerSupport

…

(add more commits to controllerSupport branch)

…

- git switch main

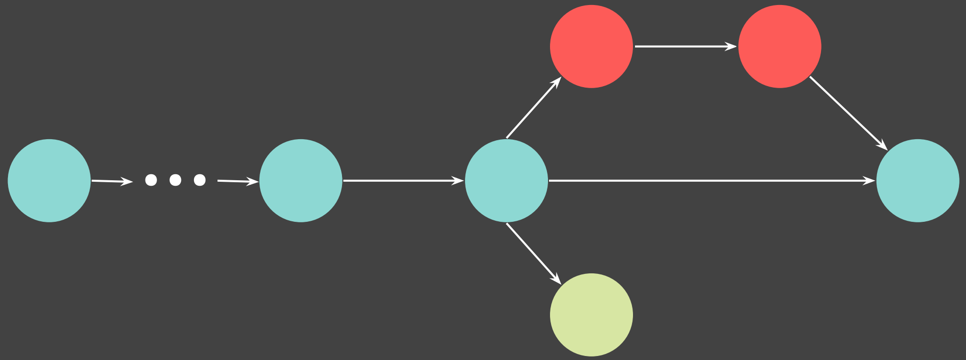
    # merge pulls changes from target to current branch

- git merge controllerSupport

🔵 main

🔴 controllerSupport

🟢 android



merge conflicts

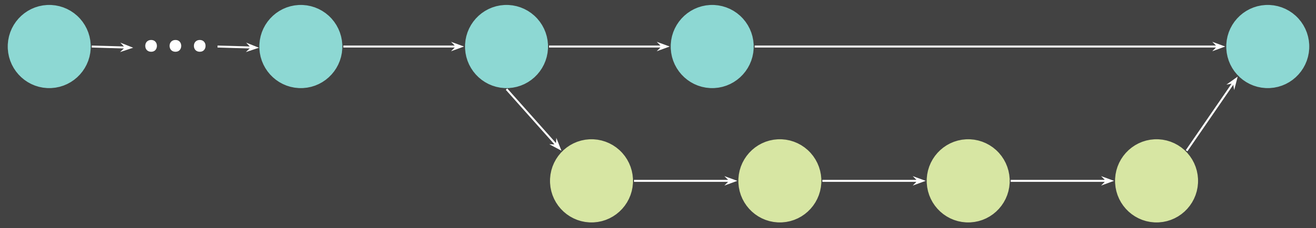
What if changes have occurred in both files, on the same lines?

```
<<<<<<<<<<< HEAD
Current Changes
===========
Incoming Changes
>>>>>>>>>>> mergeConflict
```

● main

● mergeConflict

Switch desktop for example



git merge for release branches

-

- git switch android

...

(add more commits to android branch)

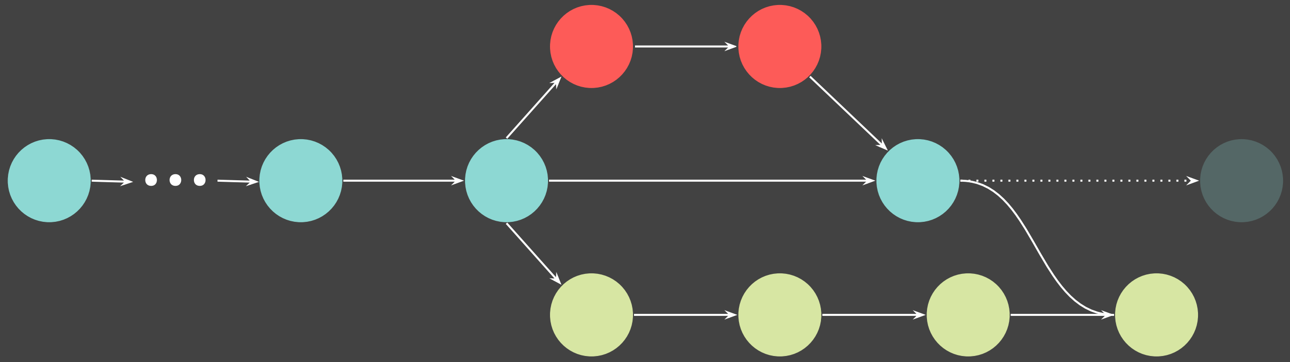
...

- git merge main

● main

● controllerSupport

● android



Tags

https://semver.org/

Semantic Versioning

X.Y.Z-alpha

Major Release

(May be unsafe to update)

Minor Release

New features (Safe to update)

Patches

Bug fixes (Safe to update)

Optional Pre-Release Tag

e.g.
-alpha
-beta
-stable

git tag

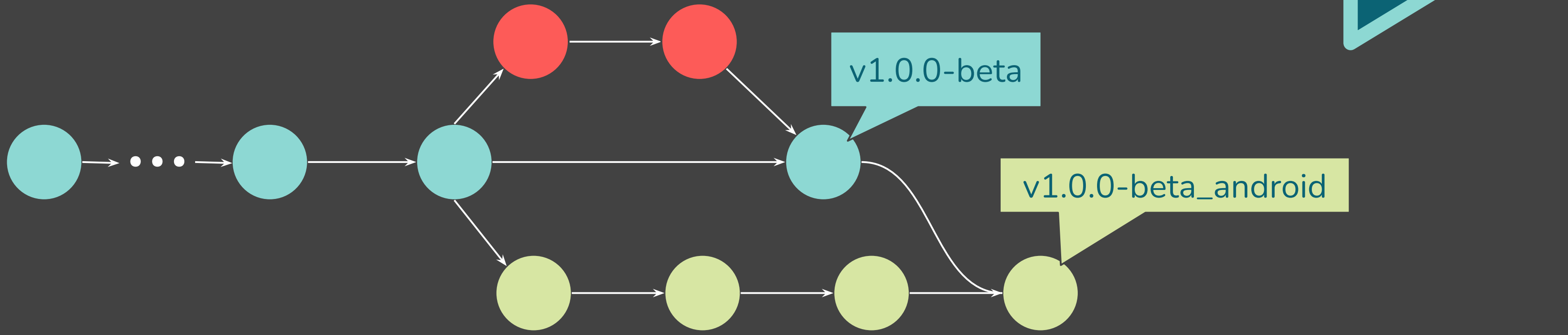
- Keeping track of releases
  ex: git tag v1.2.3-alpha <commit hash>
- git push origin <tag name>
  ex: git push origin v1.2.3-alpha

● main

● controllerSupport

● android

git tag v1.0.0-beta XXXXXX



git checkout

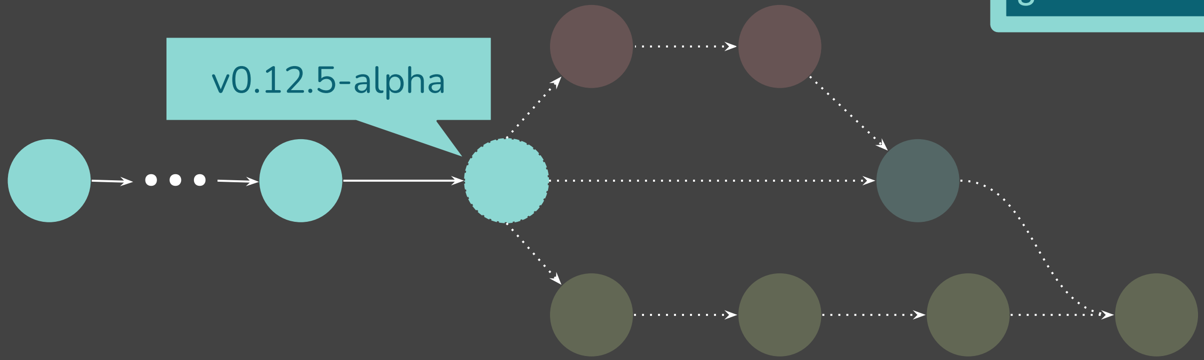
- git checkout <branch | commit hash | tag>

ex:

```
$ git checkout v0.12.5-alpha
```

- ● main
- ● controllerSupport
- ● android

git checkout v0.1.0



Android!

Unity Remote

https://docs.unity3d.com/Manual/UnityRemote5.html

Android Demo

git.io/JP1K6

- Download & install the v001...apk file

What’s working:

- Motion Controls

Remaining Coursework Items

Ethics Quiz (11/11)

- Midterm B (20 points)
- Ethics Quiz (50 points)
- Topics:
  - ACM/IEEE - CS Joint Task Force on Software Engineering Ethics and Professional Practices Ethical Principles
  - 4 areas of risks in a project (FoxMeyer Drugs' Bankruptcy paper)
  - Copyright
  - 4 areas of consideration in fair use

ACM/IEEE - CS Joint Task Force on Software Engineering and Professional Practices Ethical Principles:

1. PUBLIC
2. CLIENT AND EMPLOYER
3. PRODUCT
4. JUDGEMENT
5. MANAGEMENT
6. PROFESSION
7. COLLEAGUES
8. SELF

4 Areas of Risk in a Project (FoxMeyer Drugs' Bankruptcy paper)

1. Customer mandate
2. Scope and Requirements
3. Execution
4. Environment

Measuring Fair Use: The Four Factors

1. The purpose and character of your use
2. The nature of the copyrighted work
3. The amount and substantially of the portion taken
4. The effect of the use upon the potential market

Oral Exam (11/16 - 11/18)

Who?

- You and Dr. BC
- 30 minute interview

When?

- Week of the 15th

What?

- Your code must …
  - Function the way the client requested
  - Approximately equal (or greater) in work requirement to your teammates
  - Have two patterns and able to justify their use
- Documentation must match your code
- Code complies with your teams coding standards
- Gantt Chart

Where?

- JEB 324

Why?

- 17% of your final grade!

What else?

- Bring a laptop that is able to run your game and is able to access your GIT repository

https://webpages.uidaho.edu/drbc/cs383/assignments/OralExam.html

Post Mortem Presentation (12/7)

- Assignment Page
- Marking Key
- You have exactly 15 minutes to present.
- Group mark, but you can choose to either present as a group or have one group member present for the whole team.
- Content:
    - Break down your project in terms of team members and in terms of GRASP.
    - If given another chance to do this project, how would you then break the project down?
    - What would you change in this course to enhance learning?
    - Is there a outstanding feature that could be implemented via pair programming? Explain.

Pair Programming (Due 12/9)

- Presentation Link
- Content:
  - Find a partner (they do not have to be from your team).
  - Decide who will be the **driver** (writes code) and who will be the **observer** or **navigator** (reviews the code as it is written).
  - Pick a pair activity.
  - Work meaningfully for 2-3 hours (max), **switching roles frequently**.
    - Be sure to get the scrum master's permission before submitting your pair's work.
  - Choose one person from the pair to write a max 300 word report of the activity and submit it on BBlearn.

Final Demo & Presentation (12/9)

- Present your final product:
    - Have a working game
    - Show up
    - Have fun!

Thank you!

Questions?