GAME PROGRAMMING 2

GIT & GITHUB

Git & GitHub

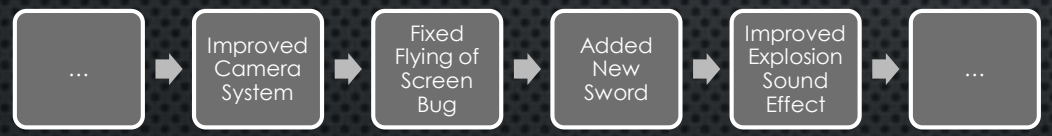
Git

- Version Control System
- Free & Open Source
- Distributed
- By far, The most common Version Control System used by software Companies

GitHub

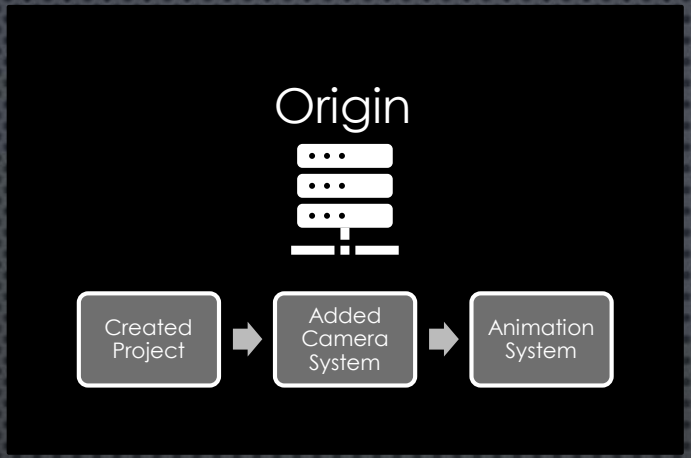
- Provides Hosting for Git-repositories
- Owned by Microsoft

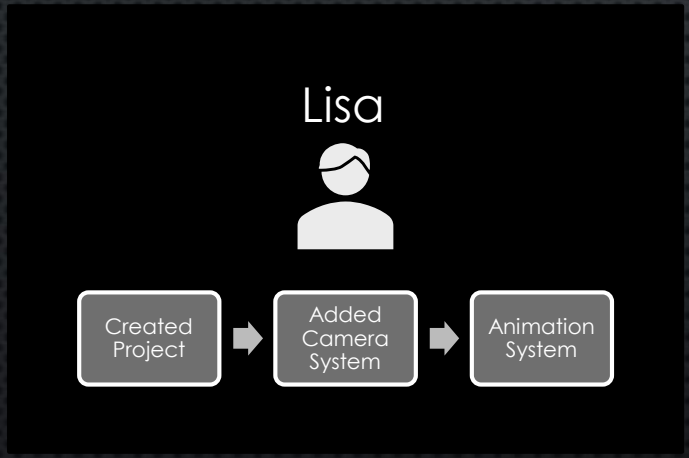
GIT, COMMITS

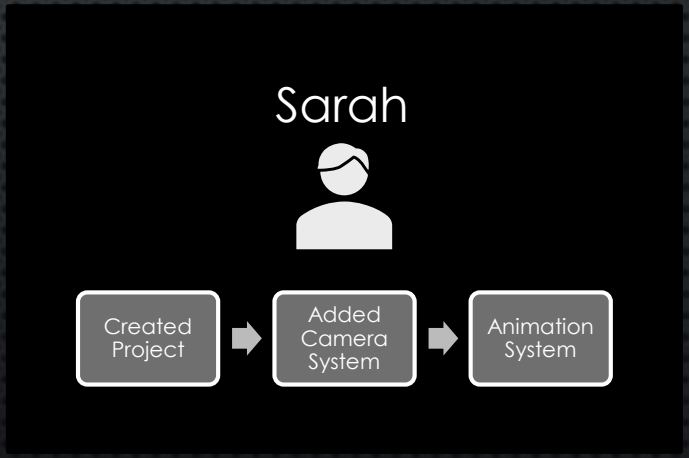


- Git is a Version Control System
- List of Commits (Snapshots) of the Project
- Can go back to any Commit at Any Time
- Everybody has a copy of the complete History

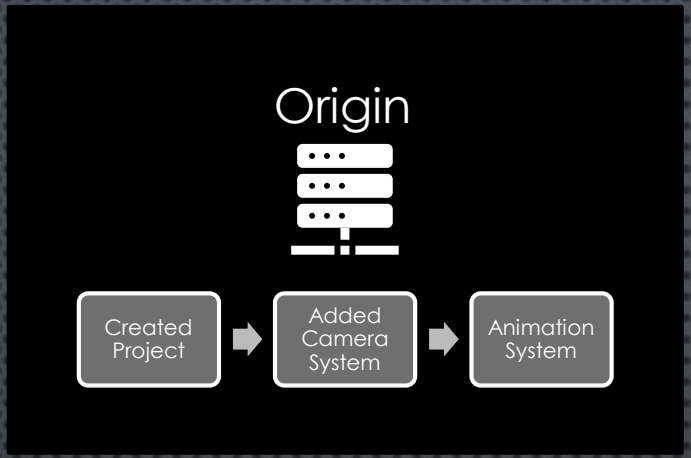
- Lisa & Sarah are developers on a new game
- They both have a copy of the complete version history
- The git repository is also hosted at Origin (Could be GitHub for example)

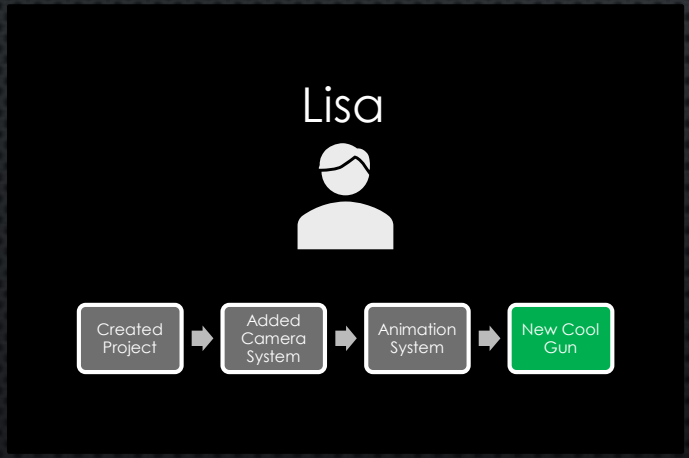


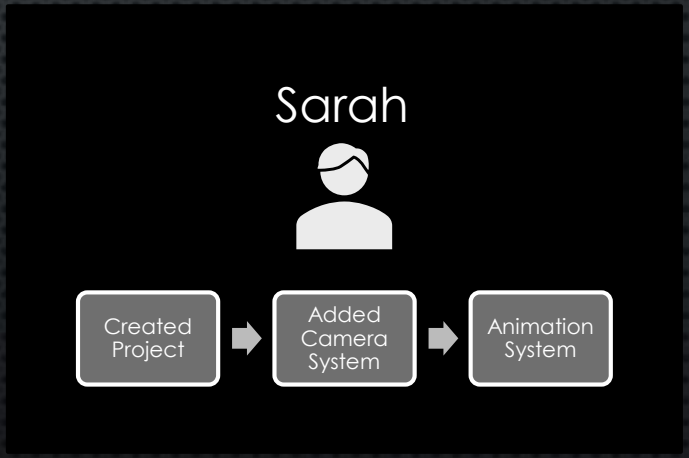




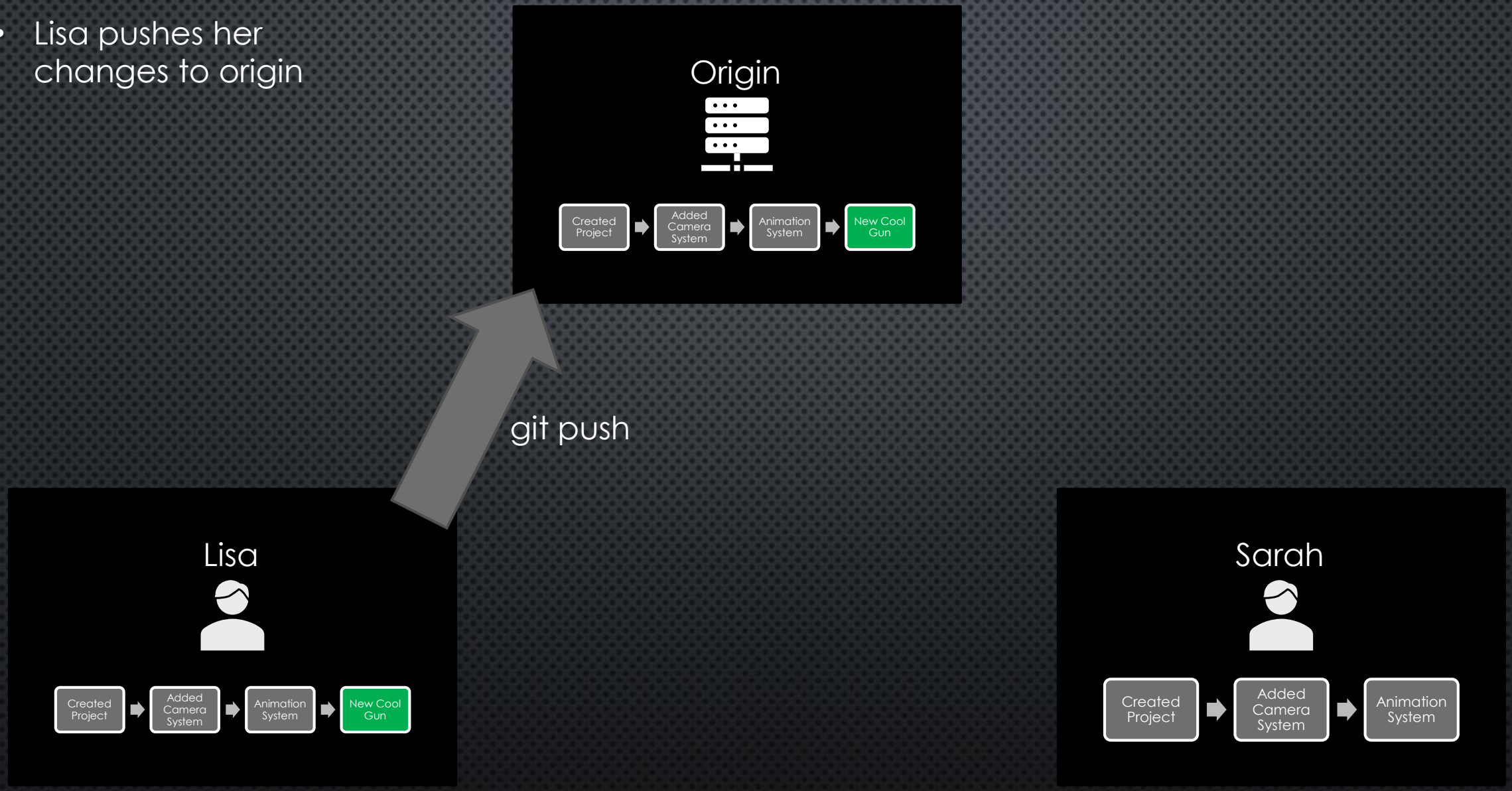
- Lisa creates a new feature "New Cool Gun" and commits it to her local repository



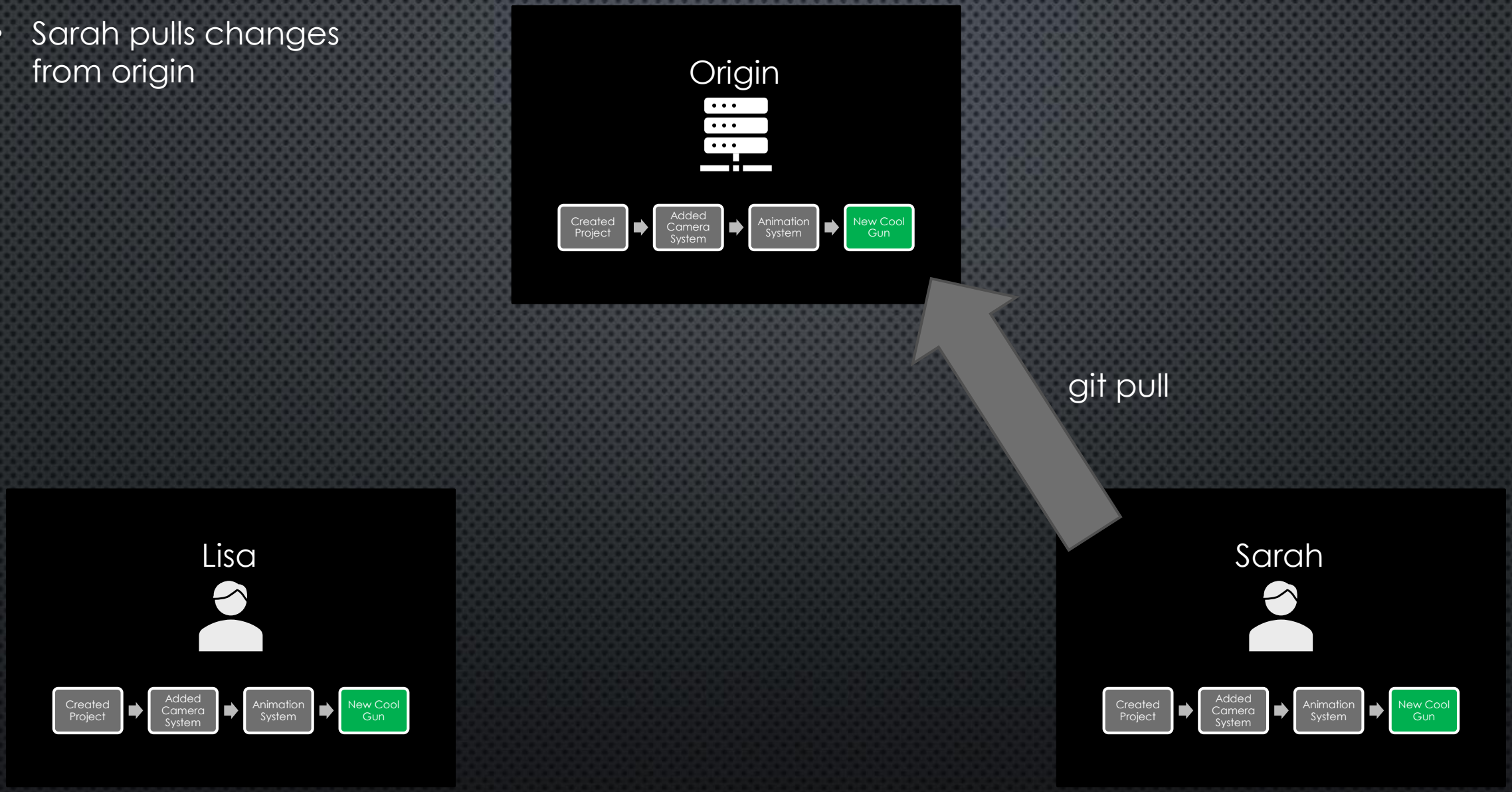




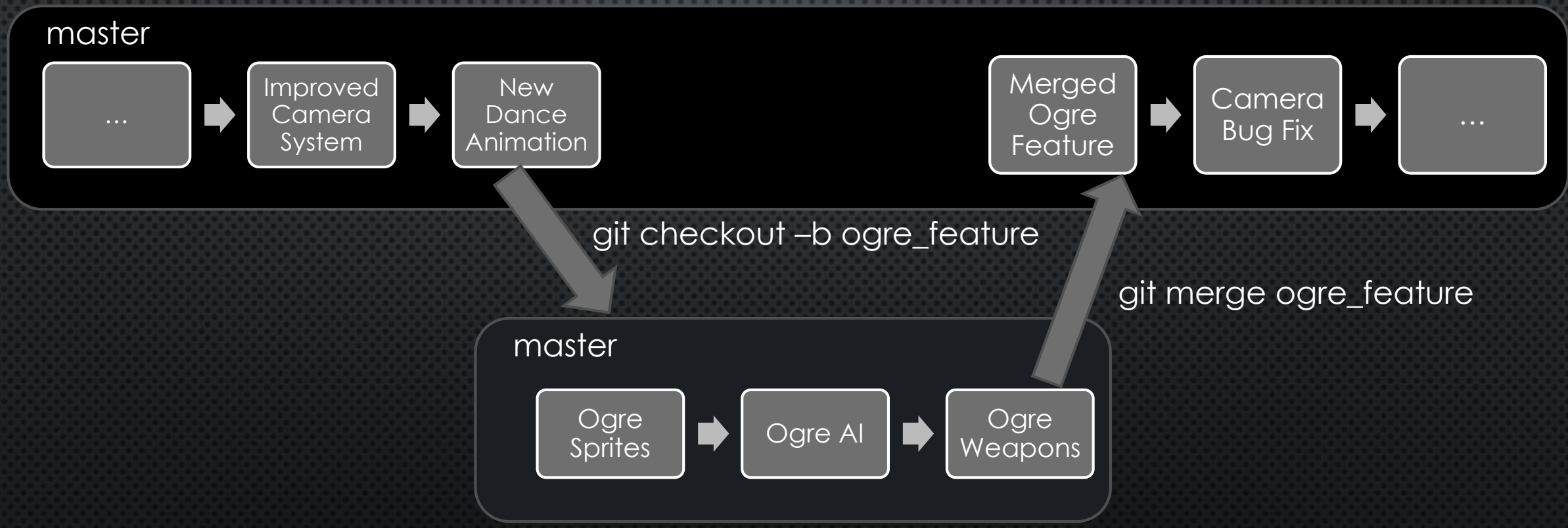
- Lisa pushes her changes to origin



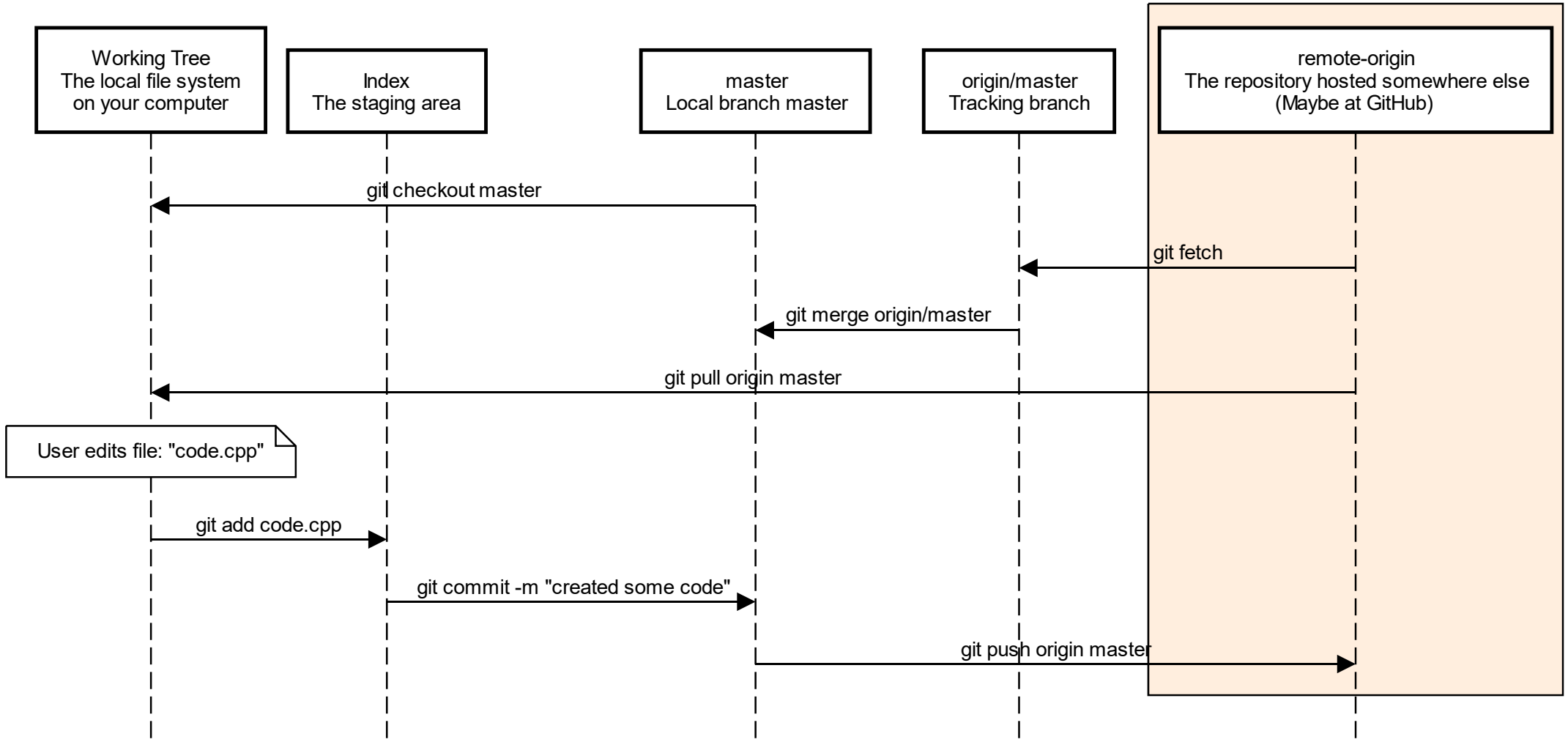
- Sarah pulls changes from origin



BRANCHES


master
...
Improved Camera System
New Dance Animation
Merged Ogre Feature
Camera Bug Fix
...
git checkout –b ogre_feature
git merge ogre_feature
master
Ogre Sprites
Ogre AI
Ogre Weapons

Git Workflow



MOST COMMON AND USEFUL GIT COMMANDS

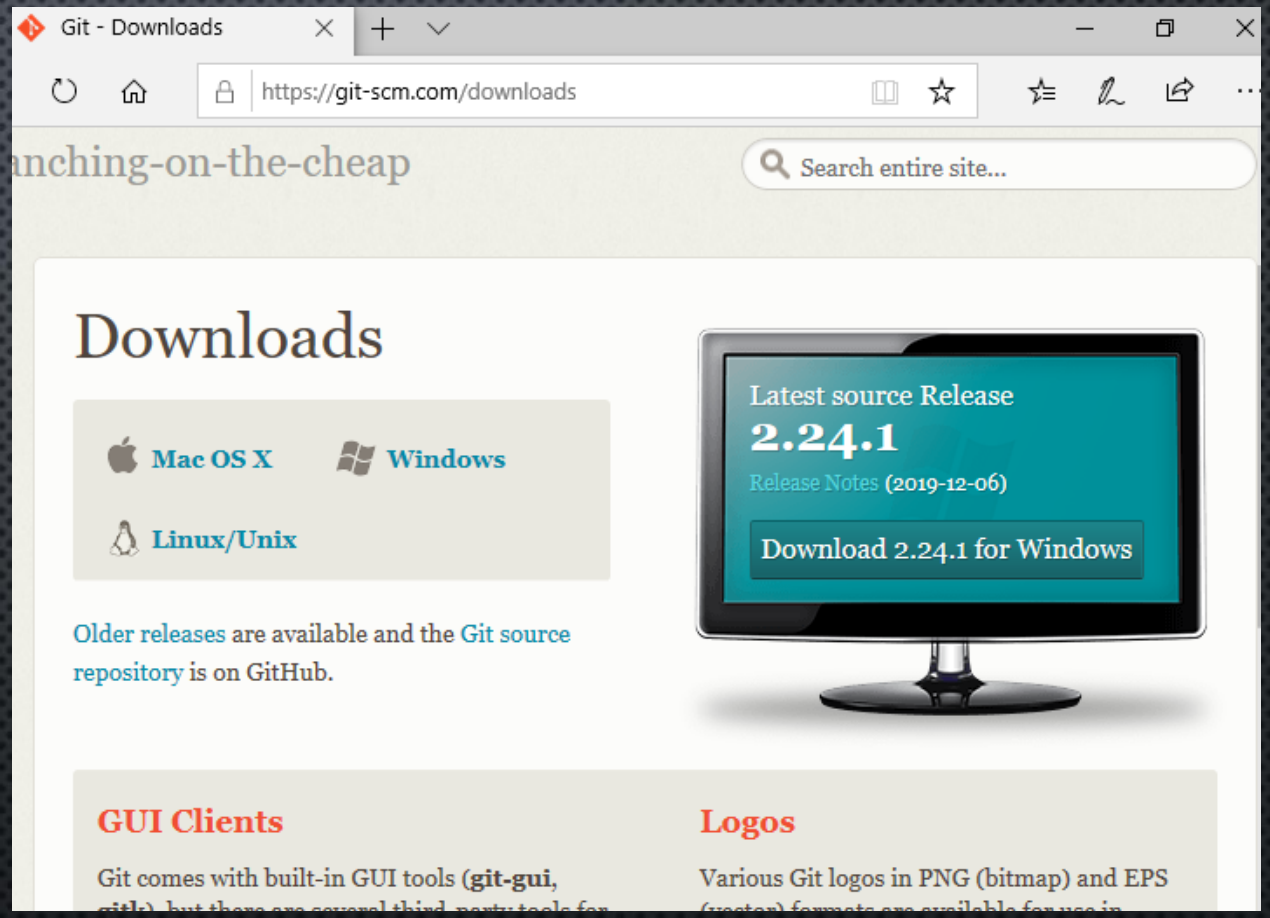
```
git init                    # Inits a new repository locally
git clone <repository>      # Clones a repository and creates a new remote "origin" pointing at the <repository>
git checkout -b feature_x   # Creates a new branch called "feature_x"
git push origin feature_x   # Pushes feature_x to the remote "origin"
git push -u origin feature_x # Pushes feature_x to the remote "origin" and sets up tracking for that branch
git pull origin feature_x   # Get all changes on feature_x from origin.
git checkout master         # Checkouts branch master
git merge feature_x         # Merges feature_x into current branch
git stash -u                # Stashes all changes you've made so that your working tree is clear
git stash pop               # Remove a single stashed state from the stash list and apply it on top of the current working tree state
git clean -xdf              # Cleans your working tree (not safe)
git reset branch_or_commit  # Sets HEAD to commit, relatively safe in its default form
git revert <commit>         # Reverts a single commit (Commonly reset would be used
git gui                     # Opens git GUI where staging, commit and push can be done in a GUI
gitk --all                  # Opens commit viewer GUI. Very useful for reviewing changes, reseting, tagging and reverting.
git status                  # View the current branch, remote tracking branch, lists any staged and unstaged files
git log                     # Quick look at history (normally gitk is used)
```

SETUP & GETTING STARTED

Install Git and Create a GitHub Account

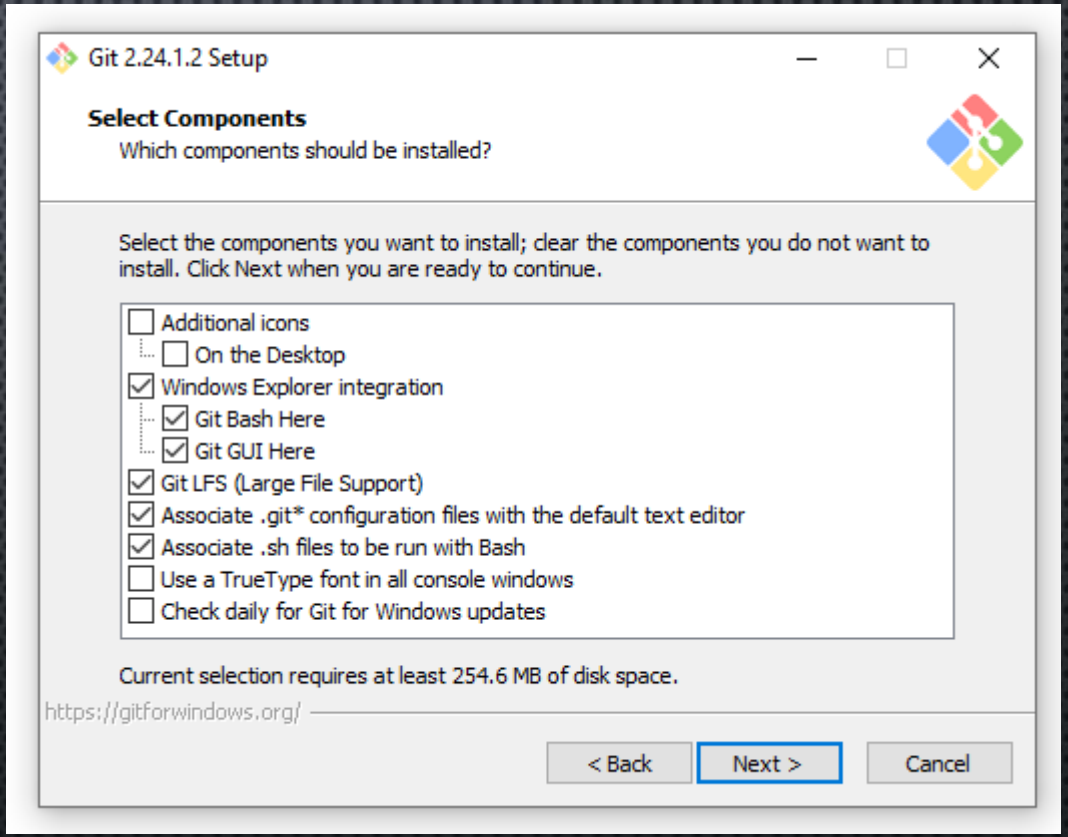
DOWNLOAD

- GIT-SCM.COM/DOWNLOADS
- DOWNLOAD THE LATEST STABLE RELEASE



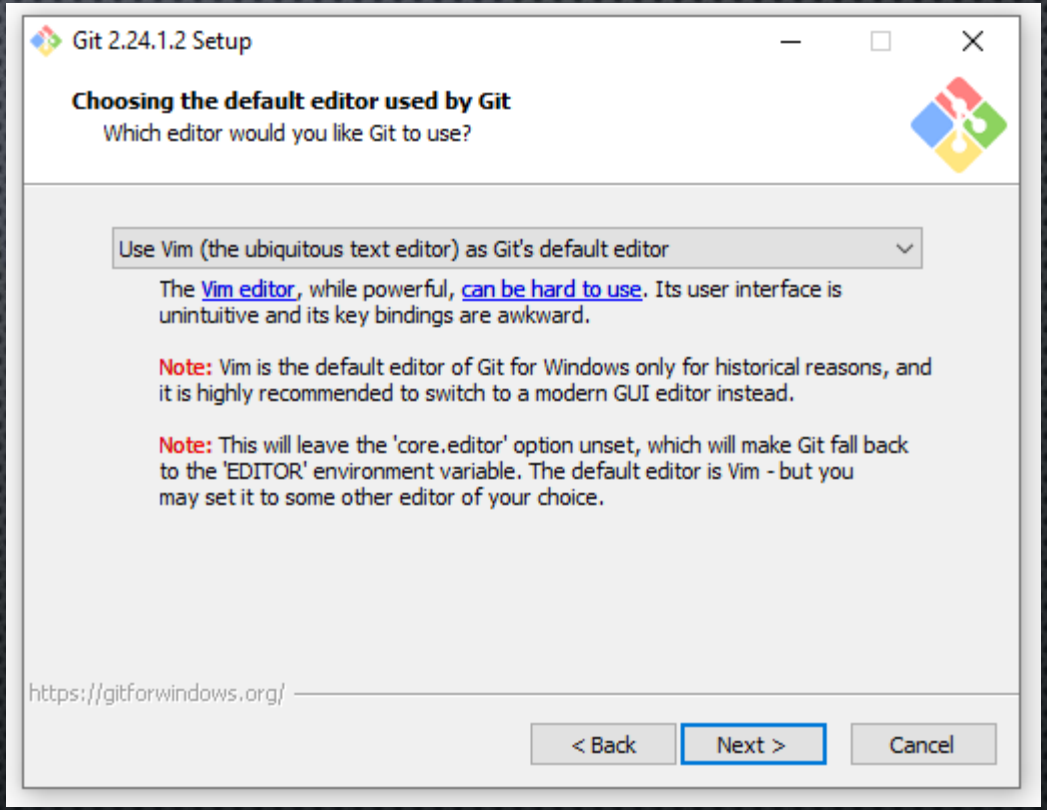
SETUP

- FOLLOW THE INSTRUCTIONS



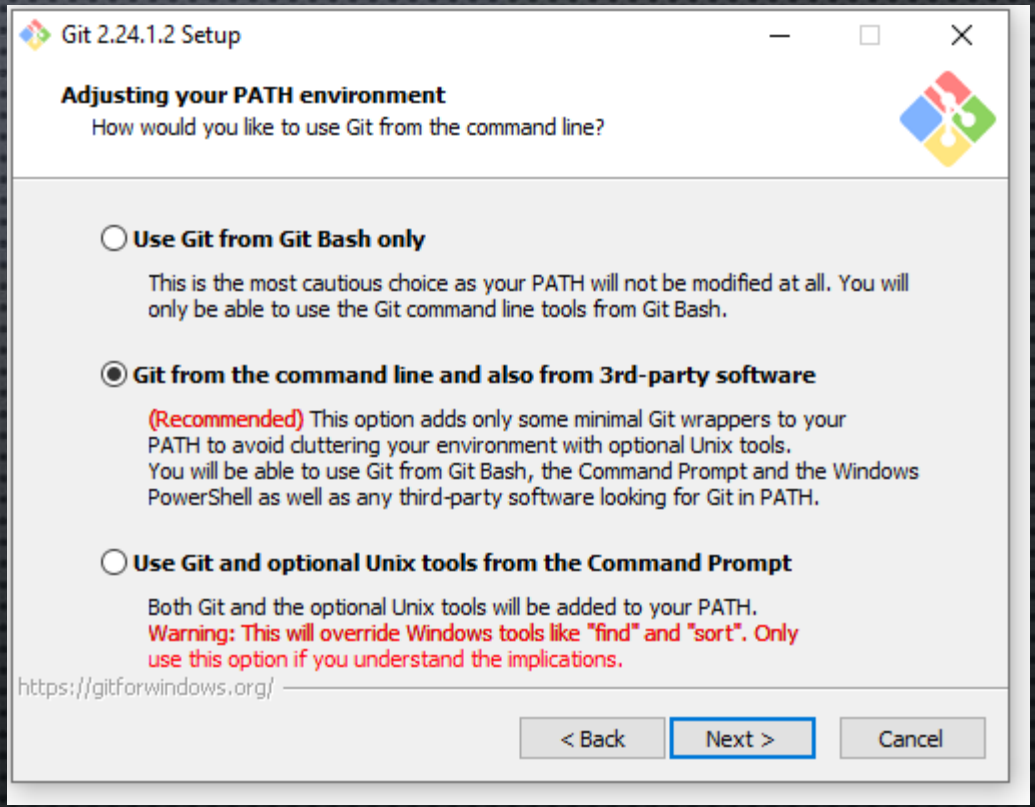
SETUP

- Choose the Text Editor of your choice (or just go with Vim)



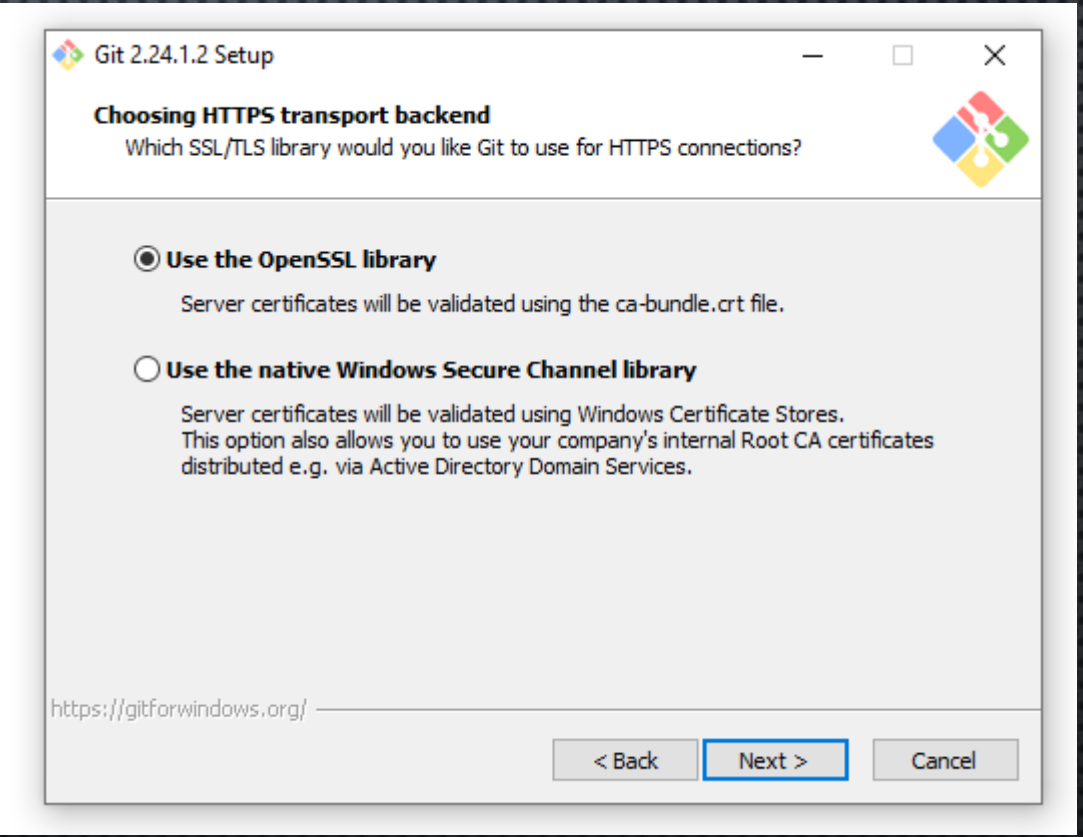
SETUP

- IN THIS SCREEN, CHOOSE THE (RECOMMENDED) OPTION



OPENSSL

- USE THE OPENSSL LIBRARY

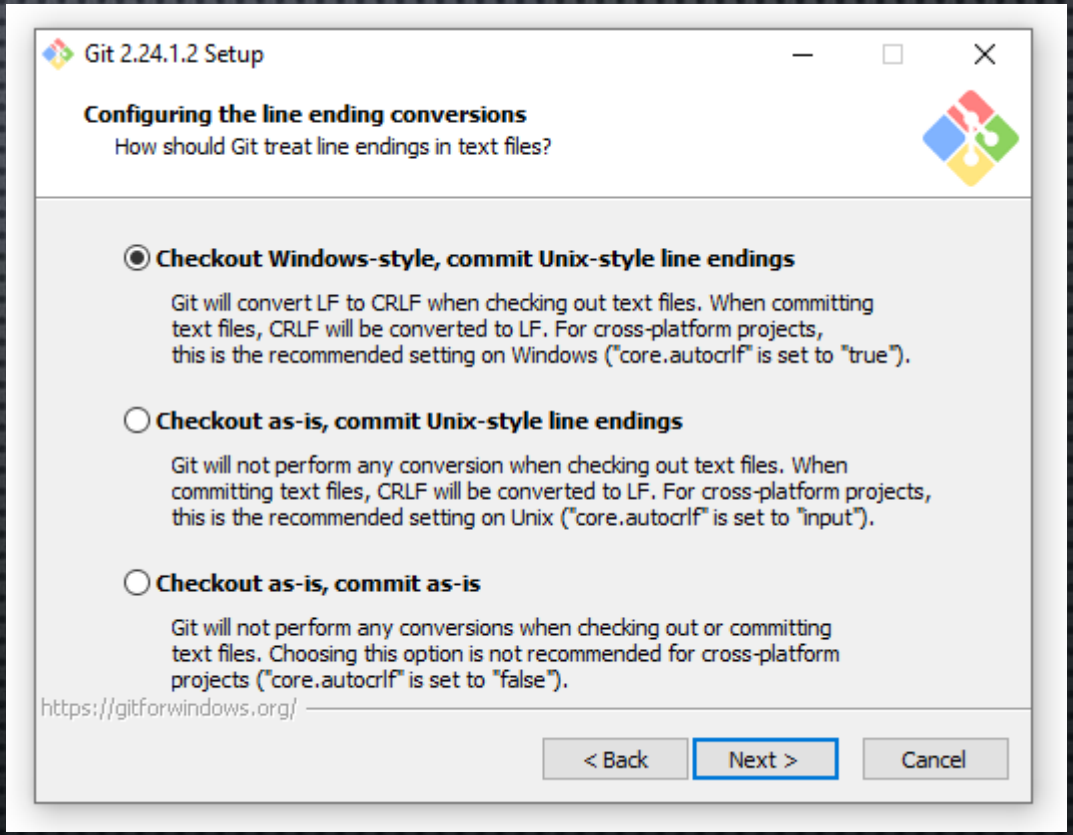


LINE ENDINGS

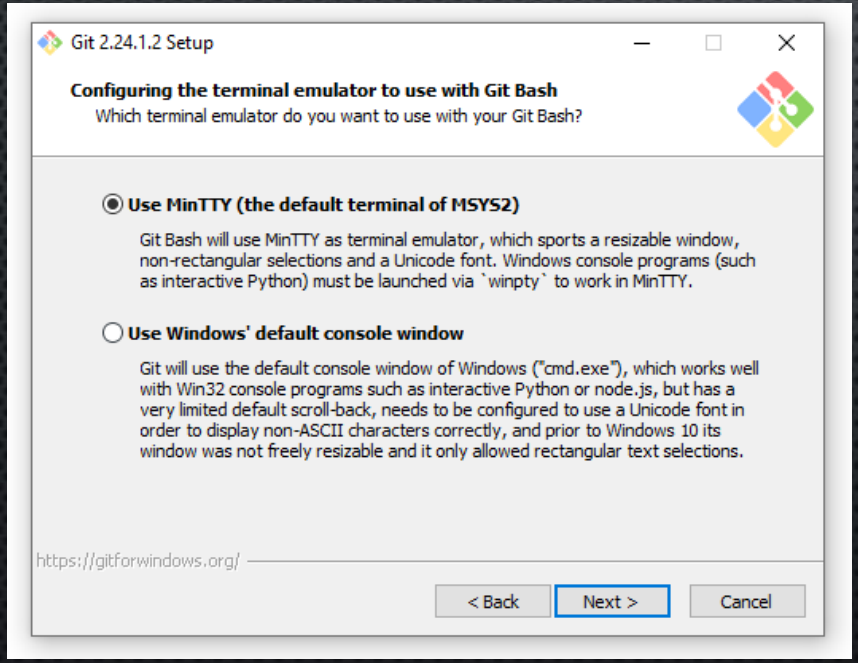
- IT IS GENERALLY RECOMMENDED TO USE "CHECKOUT WINDOWS-STYLE, COMMIT UNIX-STYLE LINE ENDINGS"

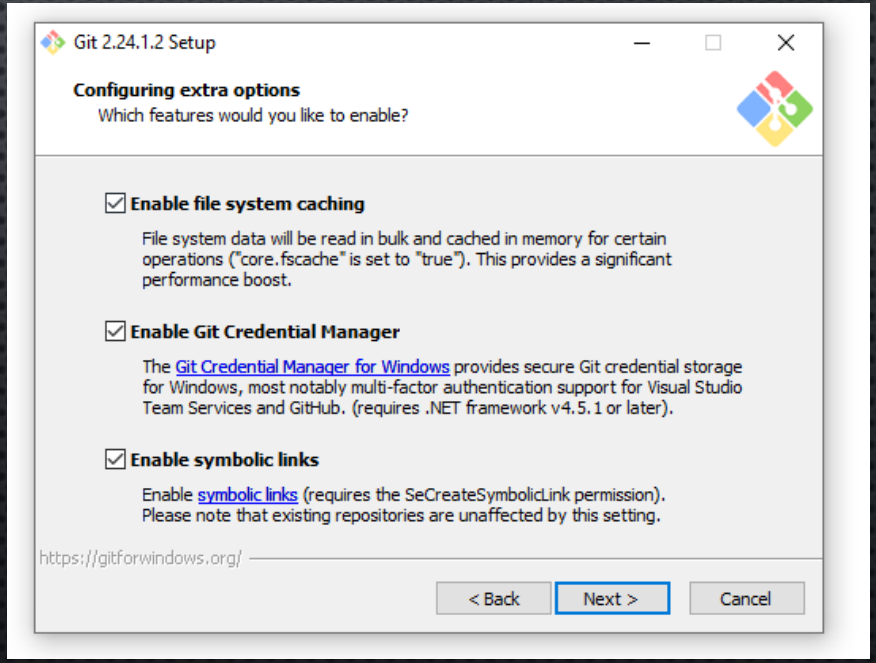
BUT

- IT CAN IN SOME CASES BE VERY ANNOYING



CONTINUE THE SETUP USING THE DEFAULT SETTINGS

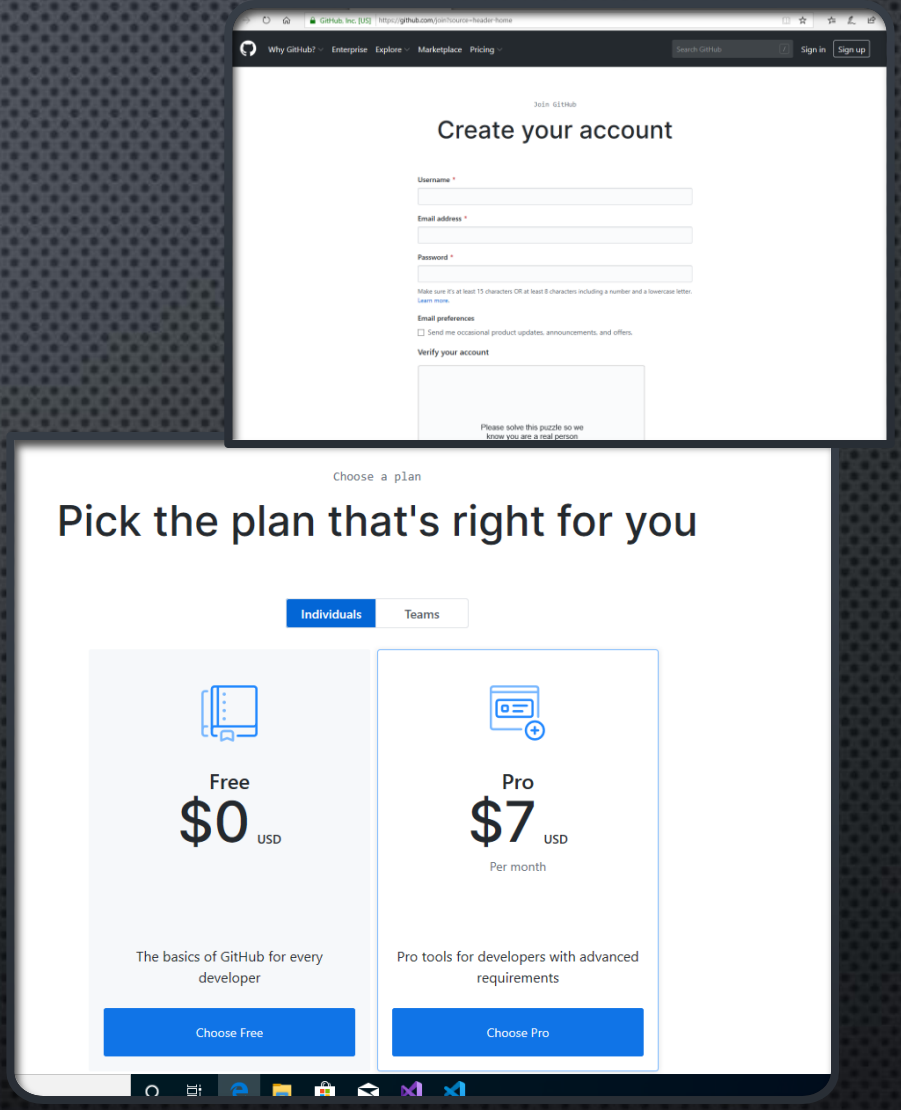




GitHub

- Create GitHub Account
- Setup SSH key from your computer to your GitHub account

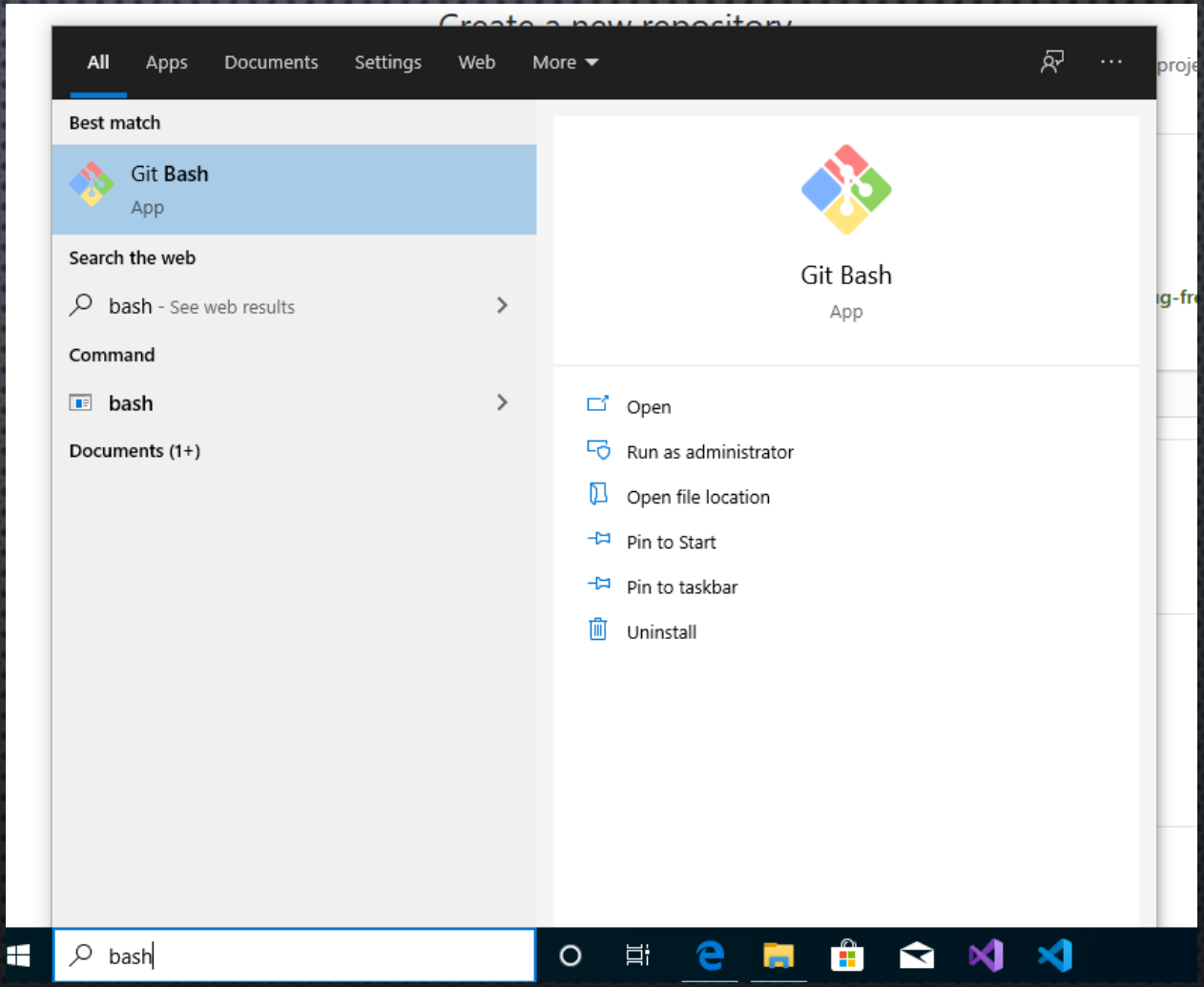
CREATE ACCOUNT



- Go to github.com
- Create an Account
- Select the Free Plan

SSH KEY

Open Bash

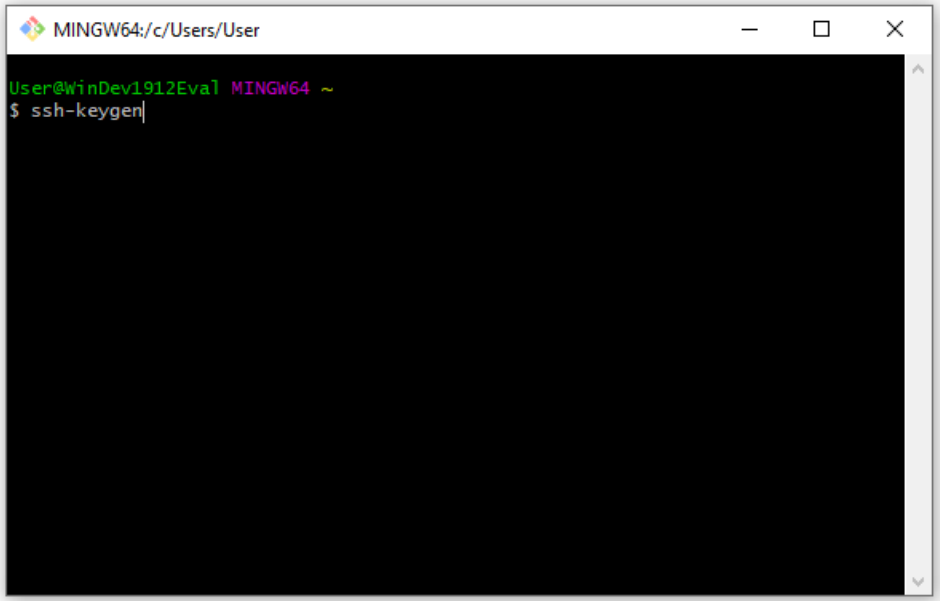


SSH-KEYGEN

Type:

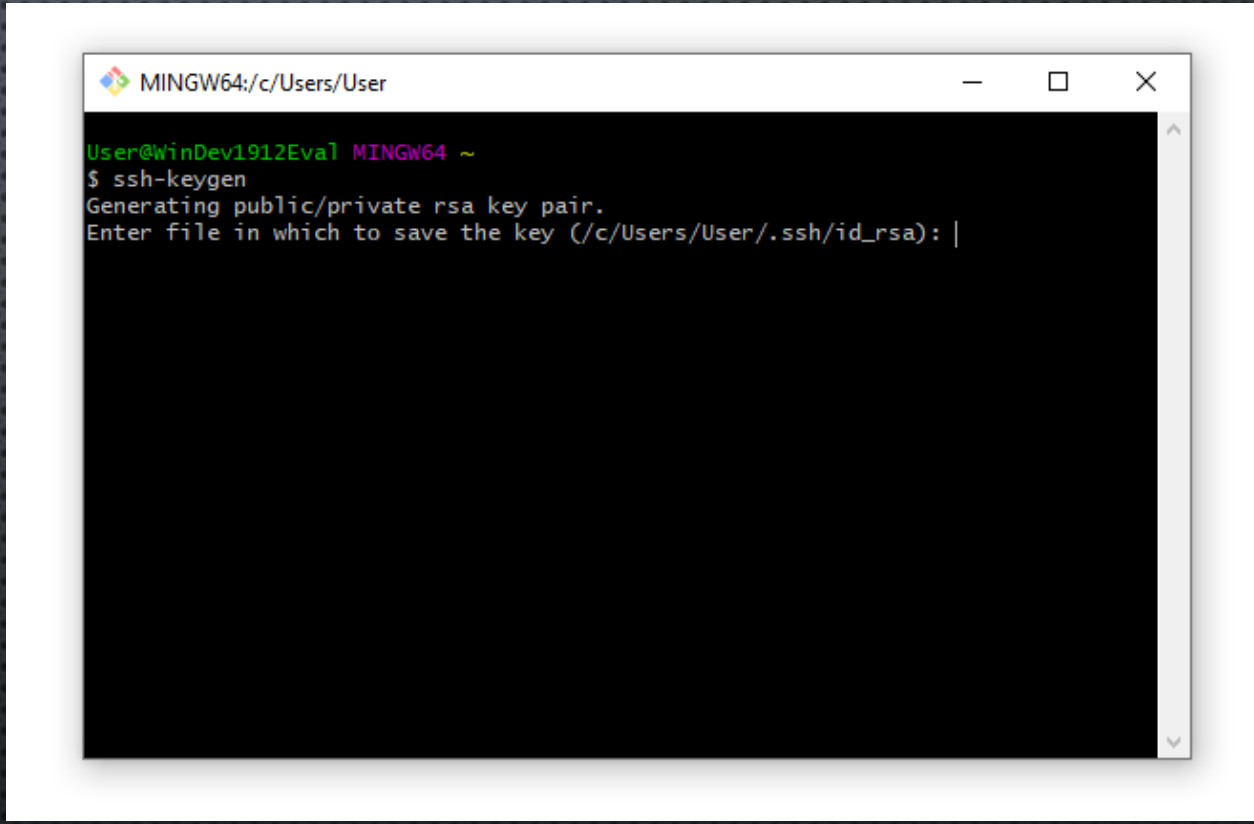
ssh-keygen

And press Enter



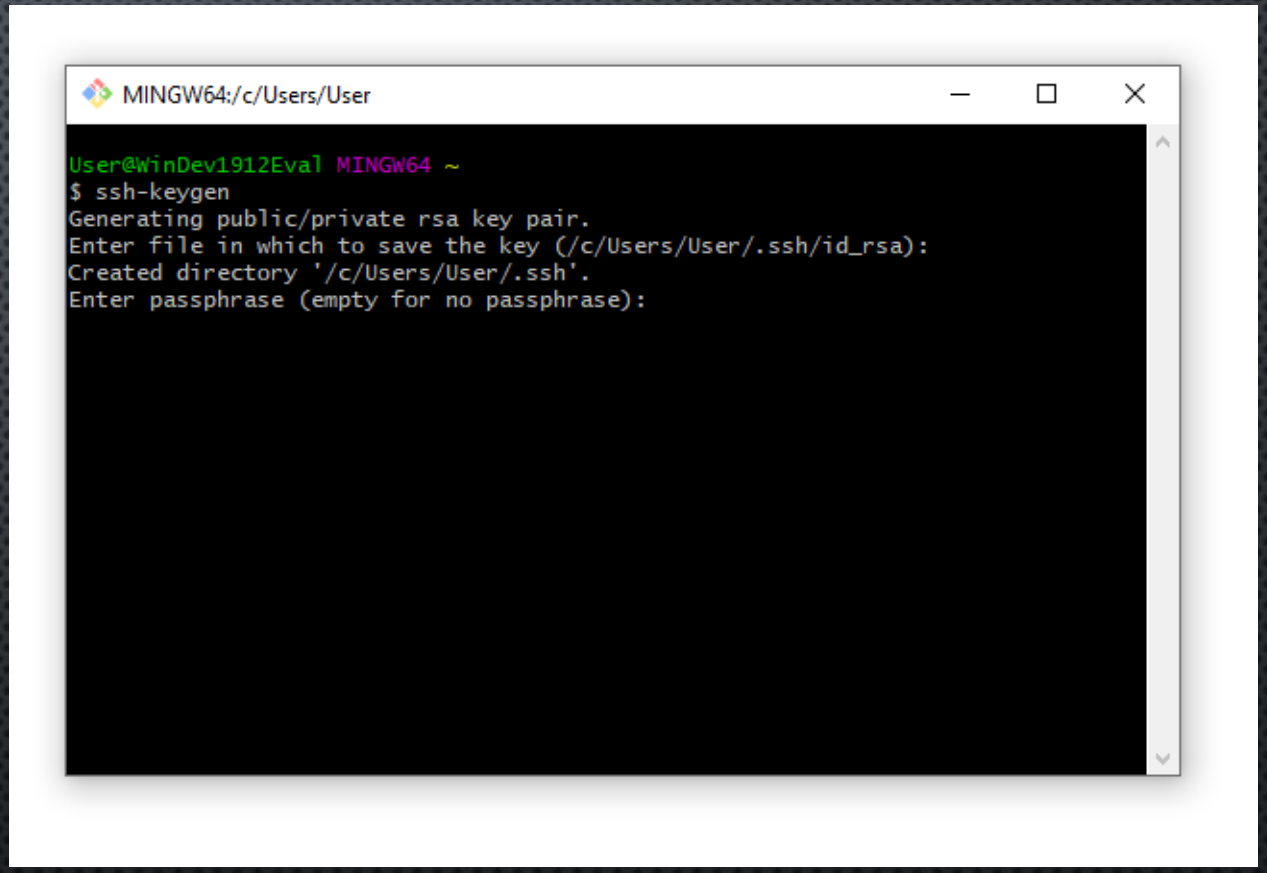
SSH-KEYGEN

- Press Enter to save the key in the default location



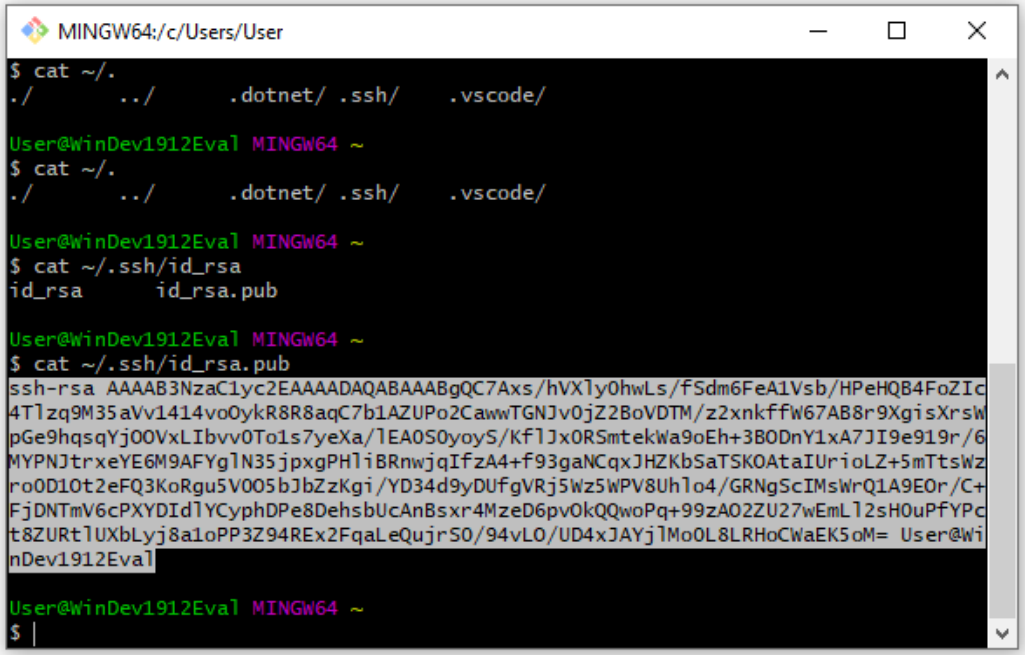
SSH-KEYGEN

- Press enter for no passphrase
- Two files are now created, id_rsa & id_rsa.pub
- id_rsa.pub holds the public key that you can distribute



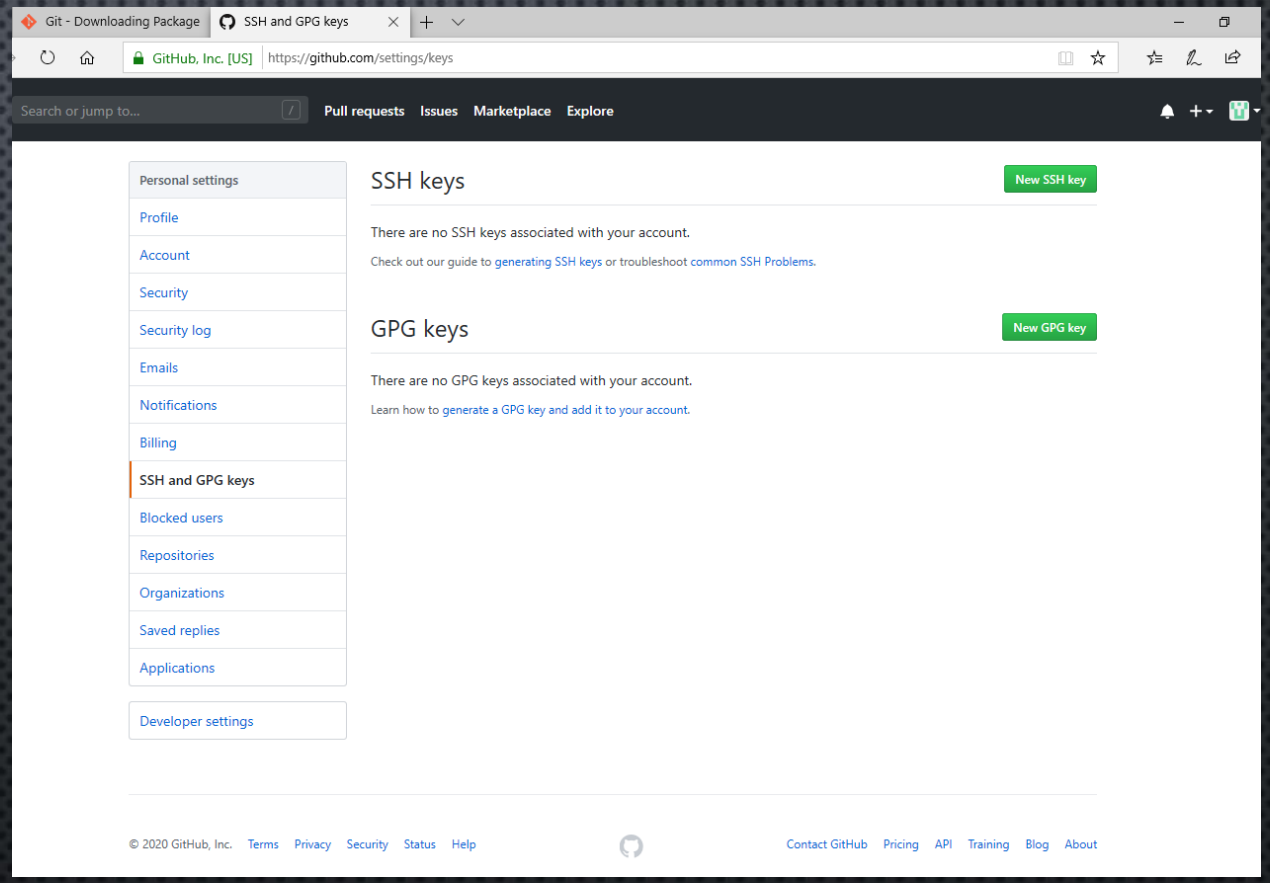
COPY THE SSH KEY

- CAT ~/.SSH/ID_RSA.PUB
- COPY THE TEXT (TRIPPEL CLICK WITH LEFT MOUSE TO SELECT THE TEXT & CTRL+INS TO COPY



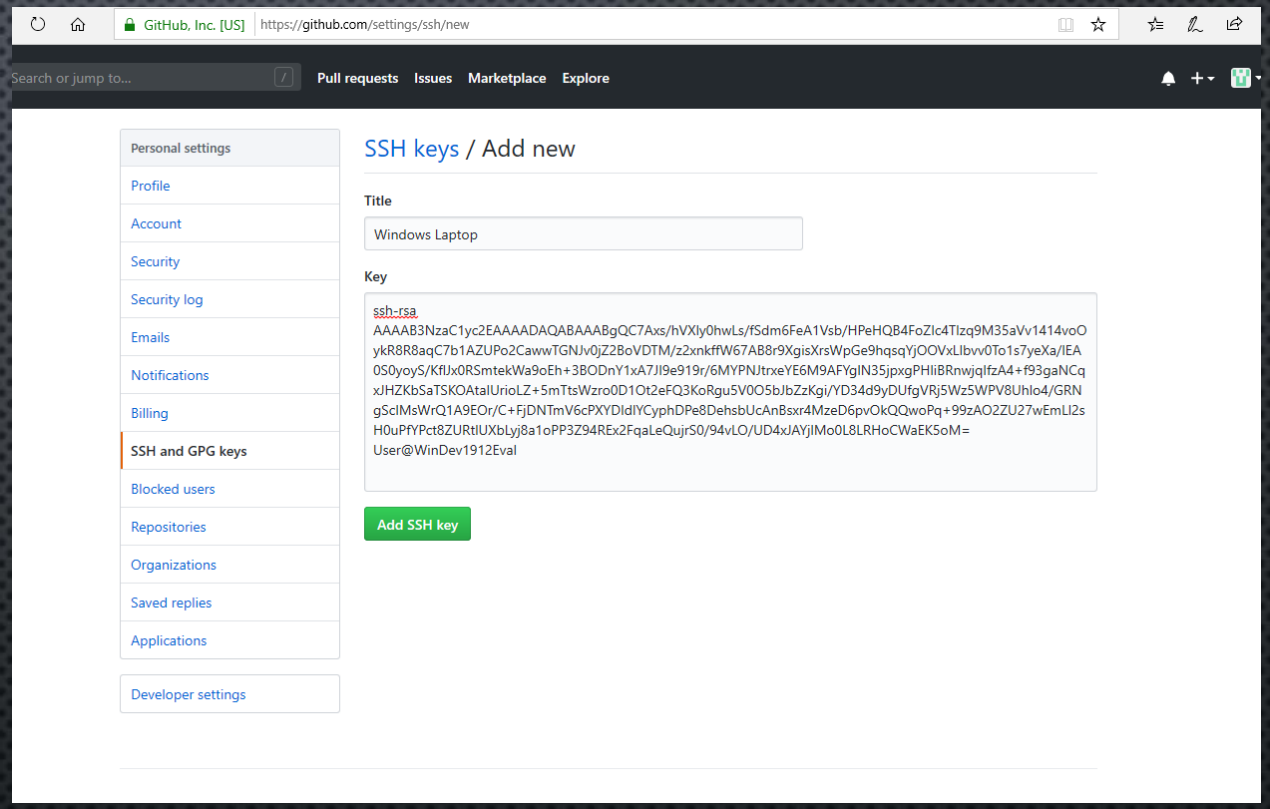
REGISTER SSH KEY

- GITHUB.COM/SETTINGS/KEYS
- CLICK "NEW SSH KEY"



REGISTER SSH KEY

PASTE THE SSH-KEY, GIVE IT A NAME AND CLICK "ADD SSH KEY".



USER NAME AND EMAIL

- You need to configure your username and email

```
git config --global user.email "your.email@example.com"
git config --global user.name "Your Name"
```

Some popular Git GUI clients…

- Visual Studio
  - Visual Studio has built in support for most common Git commands
- SourceTree
  - Very popular
- GitHub Desktop
  - GitHub's own client. Simple to Use.
- GitKraken
  - Popular at Uppsala University

REFERENCE

- https://git-scm.com/docs
  - Git reference
- https://rogerdudler.github.io/git-guide/
  - Very good, very simple git tutorial
- https://github.com/
- https://git-scm.com/download/gui/win
  - List of Windows Git GUI Clients