



VERSIONHALLINTA

MITÄ

- Historiatieto yhden tai useamman tiedoston versiosta
 - Mitä muutettiin
 - Miksi muutettiin
 - Milloin muutettiin
- Yksinkertaisimmillaan luodaan tiedostosta vain uusi versio samaan sijaintiin ja pidetään kirjaa versioista lisäämällä esimerkiksi nimen loppuun index_v0_1.html
 - Lisätään myös mahdollisesti tiedoston alkuun tai erilliseen historiakuvaukseen lyhyt kuvaus muutoshistoriasta

MIKSI

- Vikatilanteet
 - Jos uusi versio aiheuttaa järjestelmän kaatumisen, jonkin sen osan toimimattomuuden tai muun vian
 - Mahdollisuus palata aiempaan versioon
 - Palvelun toimintaa voidaan jatkaa ongelman selvittämisenkin aikana
 - Tutkia mitä muuttui
 - Ongelma selviää nopeammin
- Eri tuotantoversiot
 - Asiakas X ja asiakas Y voivat haluta tuotteelle eri ominaisuudet
 - Yksinkertaistettuna esimerkiksi nettisivun taustaväri erilainen => molemmille omat versiot koko tuotteesta
 - Näin yksinkertaiset muutokset kuitenkin yleisesti toteutetaan asiakaskonfiguraatioilla erillisten versioiden sijaan

MILLOIN

- Yksinkertaistettu esimerkki soveltuu
 - Yksittäisten päivitettävien tiedostojen muutokseen
 - Käyttöohjeet
 - Yhden kehittäjän omat projektit tai muutoin pienet softatyöt
 - Yksinkertaistetut nettisivut
 - Pienet koulutehtävät

MILLOIN

- Lähes kaikki 1-2 tiedostoa suuremmat ohjelmistokehitystyöt
 - Käytännössä kaikki asiakastyöt versionhallintaan laajuudesta riippumatta
- Useamman kehittäjän ohjelmistokehitystyöt
 - Keskitetty versionhallinta pilvessä tai omilla keskitetyillä servereillä

MIKSI YHTEISTYÖ VAATII VERSIONHALLINTAA?

- Ohjelmistokoodi on kehittäjien tärkein suojelukohde
 - Suuritöiset projektit vaativat paljon ymmärryksen ja tiedon kehitystä
 - Tuotettu koodi pitkän työn tulos
- Versionhallinnalla suojellaan tuotoksia
 - inhimillisiltä virheiltä
 - Kukaan ei pysty "vahingossa" poistamaan tai hävittämään tiedostoja(esim. USB-tikkujen kanssa on useampi opinnäytetyö hukattu)
 - erilaisten ennustamattomien katastrofien aiheuttamilta ongelmilta
 - vesivahinko tai tulipalo tuhoaa koneen jonka kiintolevyillä oli kaikki 3 vuoden kehitystyössä tuotetut koodit

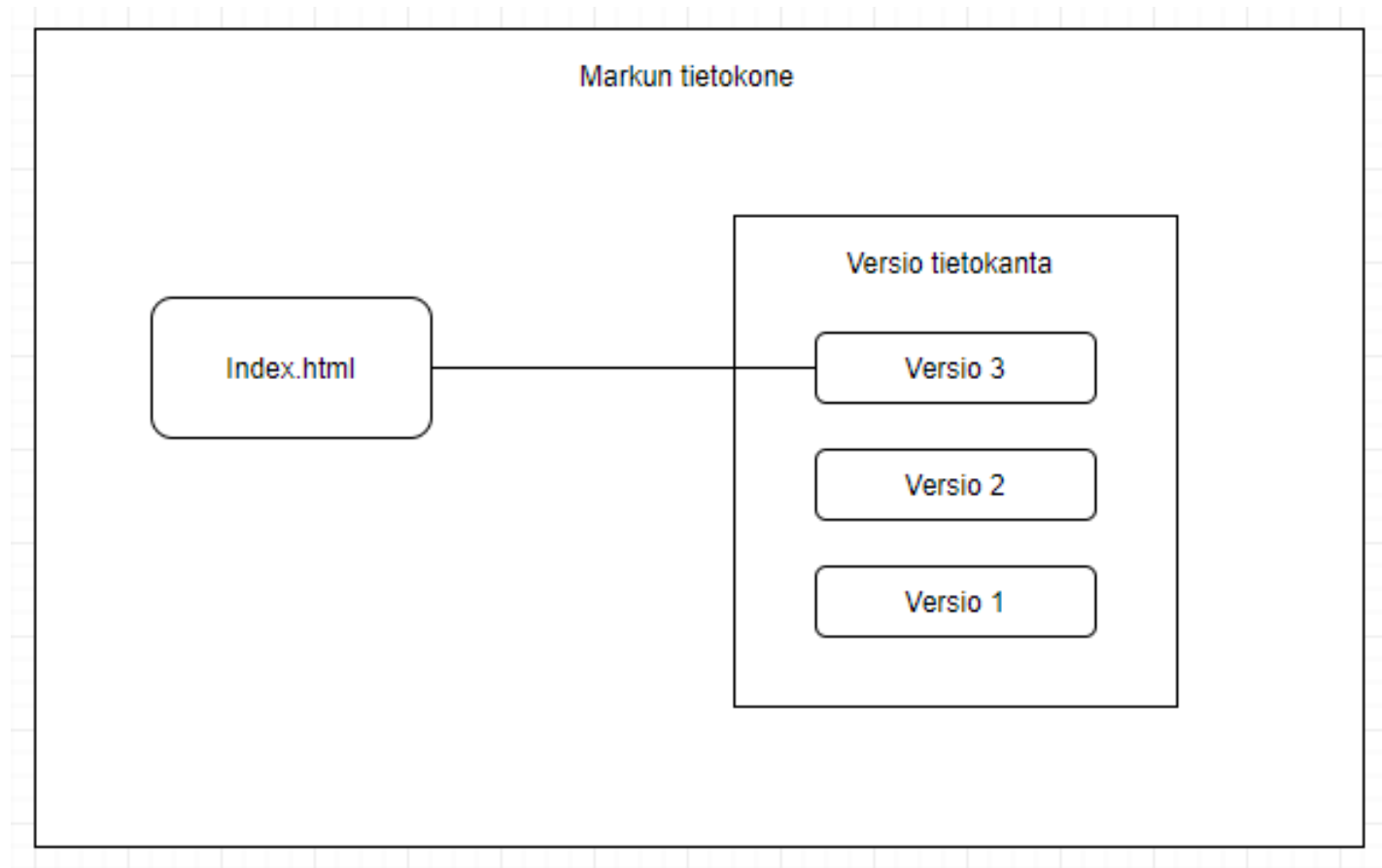
EIKÖ BACKUP- SERVERIT RIITÄ?

- Riittää varmuuskopiointiin
 - Usein erittäin oleellista olla myös jokin tällainen käytössä, varsinkin jos käytössä vain paikallistettu versionhallinta
- Ei riitä kuitenkaan varsinaiseen versionhallintaan ja muutoshistoriaan
 - Varmuuskopiointi käytännössä pitää vain tuoreimmat tiedostot tallessa eikä tarjoa mahdollisuutta liikkua tiedostoryhmien sisällä versiokokonaisuuksien välillä

MITÄ VAIHTOEHTOJA?

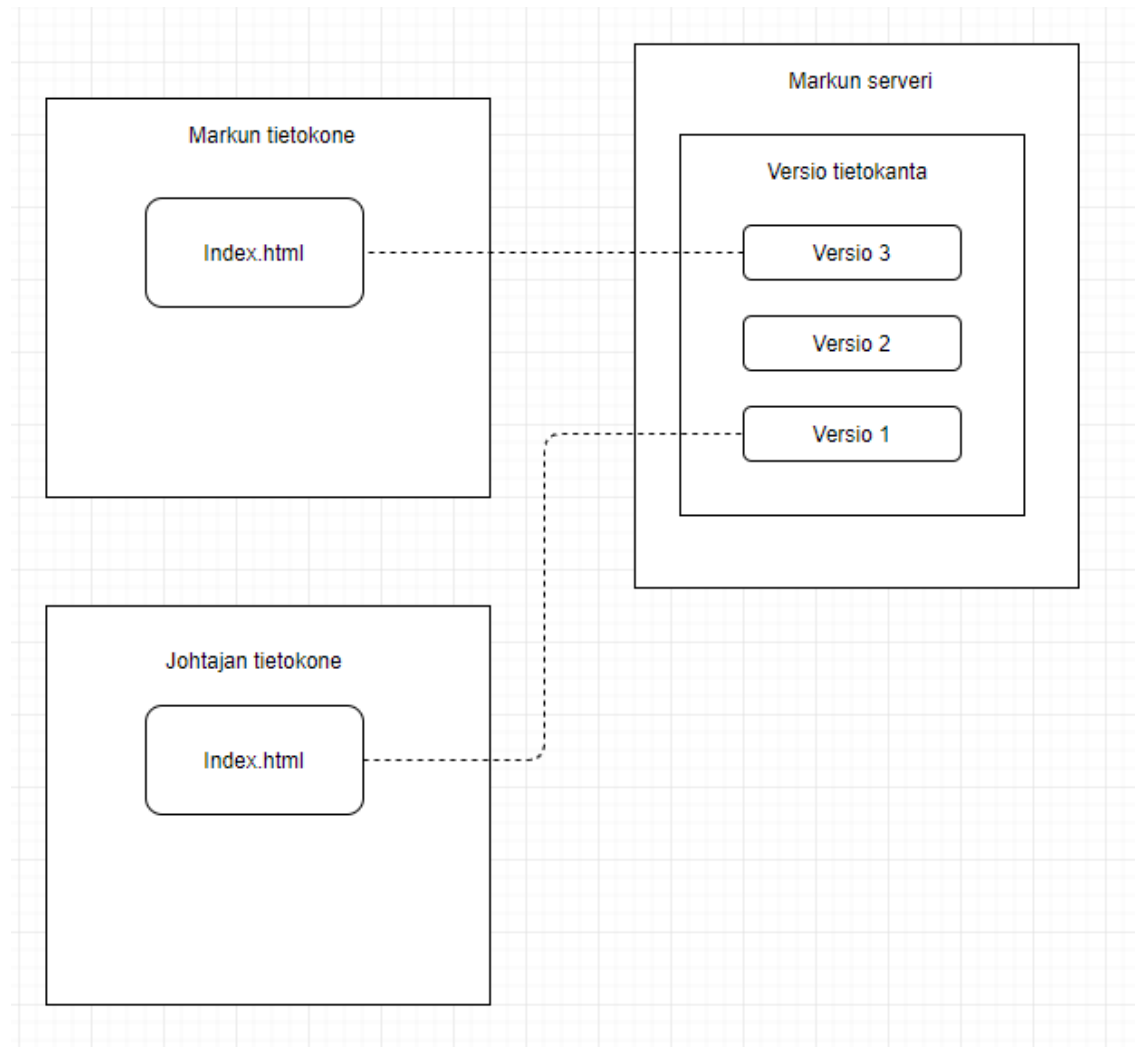
- **Paikallinen versionhallinta**

- Sopii yksittäisiin omiin kehitystöihin, tosin esim. Github ja Gitlab tarjoavat yksityisiä tilejä tähän tarkoitukseen, joten ei sinänsä enää nykypäivänä kovin oleellista
- Esim. RCS



MITÄ VAIHTOEHTOJA?

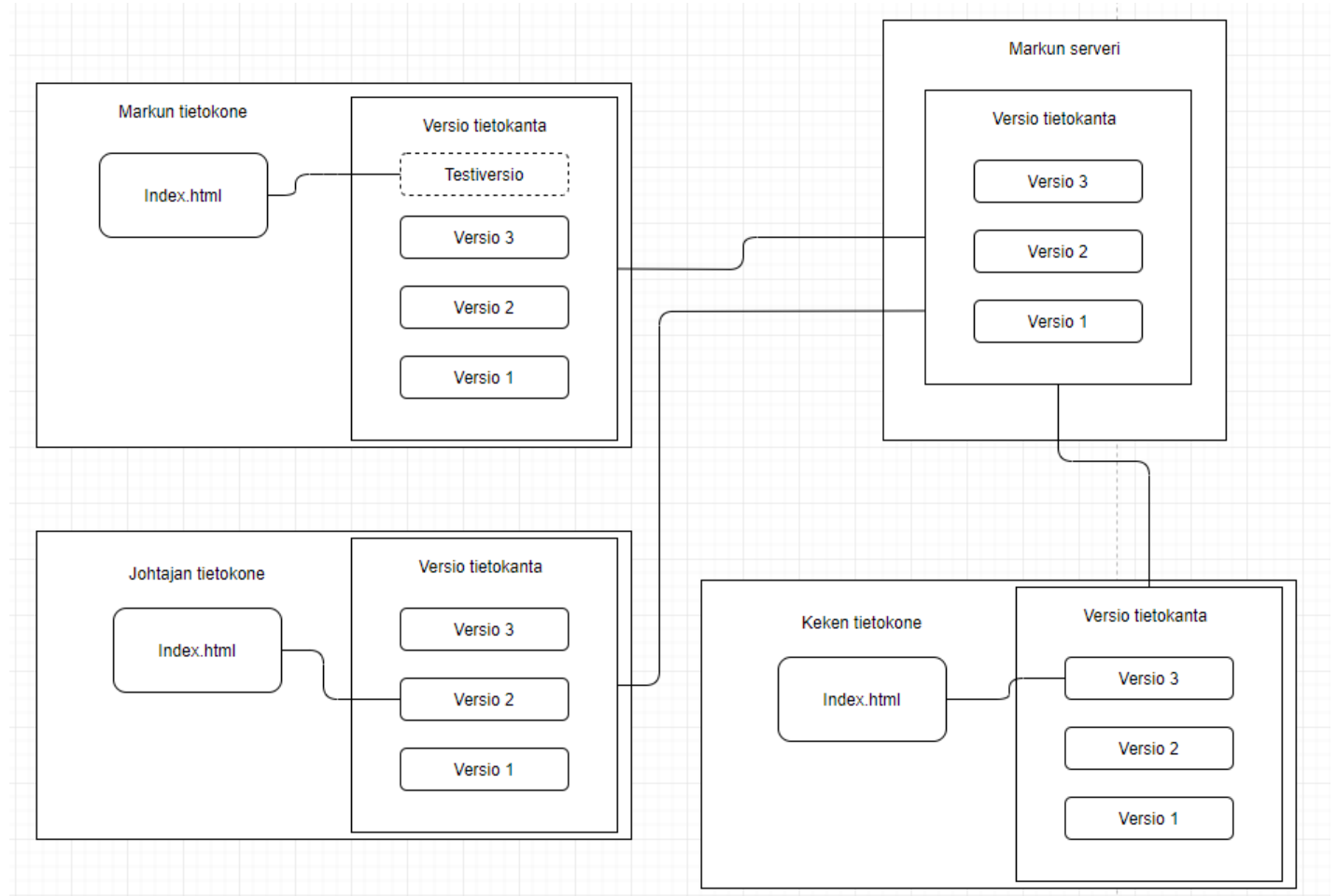
- **Keskitetty versionhallinta**
 - Yhdellä serverillä kaikki versiot kaikista versioiduista tiedostoista
 - Kehittäjien koneilla vain kehitettävä versio
 - Serverin lahoaminen voi aiheuttaa ongelmia
 - Esim. CVS, SVN



MITÄ VAIHTOEHTOJA?

■ Hajautettu versionhallinta

- Kaikilla servereillä ja kehittäjien koneilla kaikki versiot kaikista versioiduista tiedostostoista
- Serverin lahoaminen ei sinänsä ongelma, eri kehittäjillä kopiot kaikista versioista
- Esim. Git, Mercurial



MITÄ VERSIONHALLINNASTA SITTEN NÄKEE


- Esimerkiksi
 - Muutoshistoria
 - Muutokset versioiden välillä
 - Riippuen versionhallinnasta
 - TODO-listat
 - Ongelmat
 - Tiedossa olevat
 - Käyttäjien raportoimat
 - 3rd party kirjastojen riippuvuudet ja päivitystarpeet

MUUTOSHISTORIA

Branch: master ▾

Commits on Nov 23, 2019

Updated readme


 markkukynsijarvi committed on 23 Nov 2019

Verified



fe4ff67

panel scale to 1 and scaling adjusted to work better on webgl

 markkukynsijarvi committed on 23 Nov 2019



7bd7c10

speed incrementing fixed and some highscore ui adjustments

 markkukynsijarvi committed on 23 Nov 2019



754d558

Commits on Nov 22, 2019

Using Impulse force on nightmare mode so the controls are atleast a b... [...](#)

 markkukynsijarvi committed on 22 Nov 2019



8ba76ea

Probably ok for assignemnt now [...](#)

 markkukynsijarvi committed on 22 Nov 2019



9726a60

Commits on Nov 21, 2019


Pre collider removed partly since collision controlled now with downwa... [...](#)

 markkukynsijarvi committed on 21 Nov 2019



82482f6

Added block drop preview on procrastination mode [...](#)

 markkukynsijarvi committed on 21 Nov 2019



260c07e

Commits on Nov 20, 2019

blockfallspeed accelerated restored to 1 as its now working better

 markkukynsijarvi committed on 20 Nov 2019



e5d188b

Increased speed collision now working without objects bouncing of eac... [...](#)

 markkukynsijarvi committed on 20 Nov 2019



bb94d21

MUUTOKSET VERSIOIDEN VÄLILLÄ

Added own name to the first line on assignment.md #7

Edit

Merged markkukynsijarvir... merged 1 commit into `master` from `assignment-edit` on 3 Dec 2019

Conversation 0

Commits 1

Checks 0

Files changed 1

+1 -0



markkukynsijarviraseko commented on 3 Dec 2019

Collaborator



No description provided.

Reviewers



No reviews

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet



Added own name to the first line on assignment.md

a089595



markkukynsijarviraseko merged commit 1eff33f into `master` on 3 Dec 2019

Revert



markkukynsijarviraseko deleted the `assignment-edit` branch on 3 Dec 2019

Restore branch

MUUTOKSET VERSIOIDEN VÄLILLÄ

Added own name to the first line on assignment.md #7

[Edit](#)

Merged markkukynsijarvir... merged 1 commit into `master` from `assignment-edit` on 3 Dec 2019

Conversation 0

Commits 1

Checks 0

Files changed 1

+1 -0

Changes from all commits ▾ File filter... ▾ Jump to... ▾ ⚙ ▾

0 / 1 files viewed ⓘ

[Review changes ▾](#)

▼ 1 ████ Module 2/Module 2.3 Assignment.md 📄

<>

📄

☐ Viewed

...

... @@ -1,3 +1,4 @@

1 + Markku Kynsijärvi

1 2 # Module 2.3 Assignment

2 3


3 4 Similar to the last exercise, you are going to make a change to this document, but you will make the change on the command line

⌕


MUUTOKSET VERSIOIDEN VÄLILLÄ

Added own name to the first line on assignment.md #7

[Edit](#)

 **Merged** markkukynsijarvir... merged 1 commit into `master` from `assignment-edit` on 3 Dec 2019

 Conversation 0

 Commits 1

 Checks 0

 Files changed 1

+1 -0 

 Commits on Dec 3, 2019

Added own name to the first line on assignment.md



markkukynsijarviraseko committed on 3 Dec 2019



a089595



MUUTOKSET VERSIOIDEN VÄLILLÄ

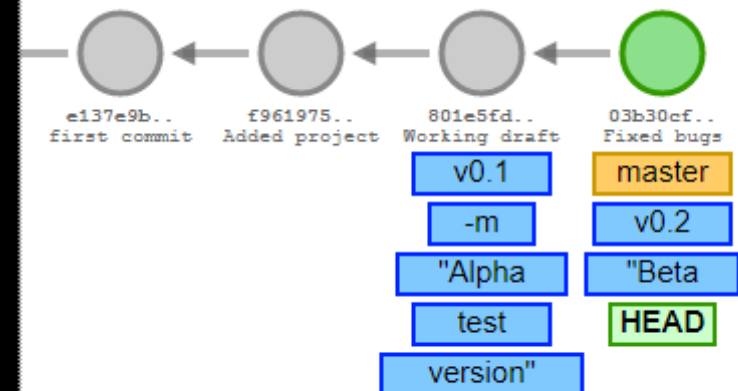
9 Assets/Scripts/CollisionControl.cs	
@@ -22,14 +22,17 @@ private void FixedUpdate() 22 if(!collided) 23 this.transform.Translate(0, -(blockFallSpeed), 0, Space.World); 24 25 - if (this.transform.position.y < 0) 26 { 27 //make sure to turn gravity on for objects if they fall off the plane 28 - //this.GetComponent<Rigidbody>().useGravity = true; 29 collided = true; 30 } 31 } 32 - 33 private void OnCollisionEnter(Collision collision) 34 { 35 collided = true;	22 if(!collided) 23 this.transform.Translate(0, -(blockFallSpeed), 0, Space.World); 24 25 + if (this.transform.position.y <= 0) 26 { 27 //make sure to turn gravity on for objects if they fall off the plane 28 + this.GetComponent<Rigidbody>().useGravity = true; 29 collided = true; 30 } 31 + //if block falls out of the island, just destroy it 32 + if (this.transform.position.y < -5) 33 + Destroy(gameObject); 34 } 35 + 36 private void OnCollisionEnter(Collision collision) 37 { 38 collided = true;

BRANCHING ESIMERKKEJÄ

- Yhden henkilön operaatio
 - Konfliktit ei olennaisia, päivitetään suoraan master-branchiin. Kuten esim. paikallisen versionhallinnan kanssa.
 - Ei tarvetta feature- tai bugfix-branchille

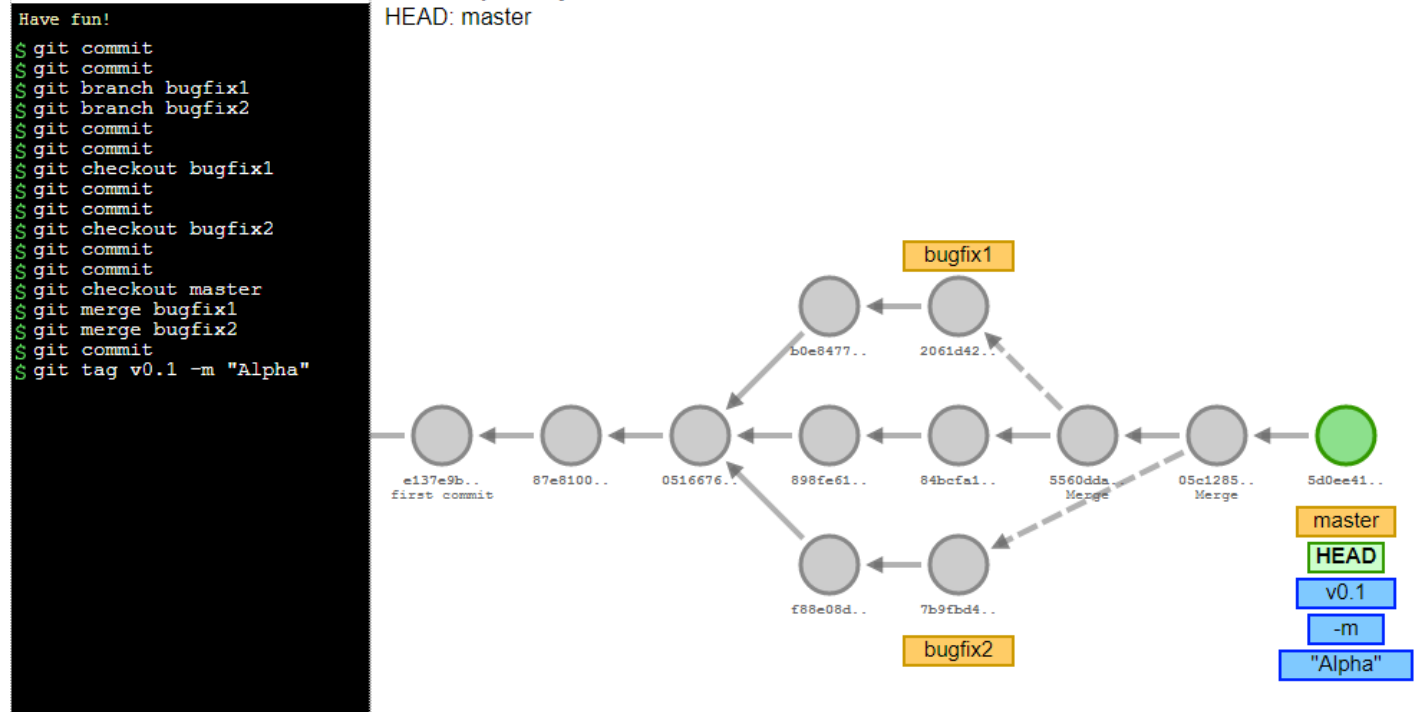
```
Have fun!  
$ git commit -m "Added  
project"  
$ git commit -m "Working  
draft"  
$ git tag v0.1 -m "Alpha test  
version"  
$ git commit -m "Fixed bugs"  
$ git tag v0.2 -m "Beta test  
version"  
$ git log  
> 03b30cf Fixed bugs  
> 801e5fd Working draft  
> f961975 Added project  
> e137e9b first commit
```

HEAD: master



BRANCHING ESIMERKKEJÄ

- Ominaisuuksien tai bugien korjaamiseksi erilliset sivuhaarat
 - Eristetään bugikorjaukset varsinaisesta kehitystyöstä, yhdistetään kokonaisuuteen vasta korjauksen jälkeen
 - Uudet ominaisuudet vasta valmiina kokonaisuuteen mukaan

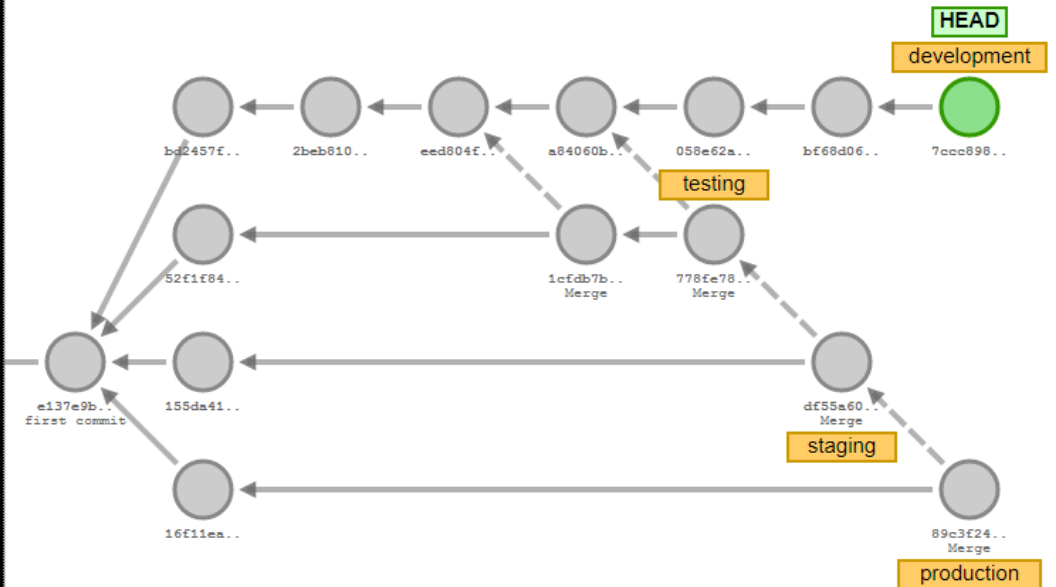


BRANCHING ESIMERKKEJÄ

- Environment branches

```
Have fun!  
$ git branch development  
$ git branch testing  
$ git branch staging  
$ git branch production  
$ git checkout staging  
$ git branch -d master  
$ git commit  
$ git checkout testing  
$ git commit  
$ git checkout production  
$ git commit  
$ git checkout development  
$ git commit  
$ git commit  
$ git checkout testing  
$ git merge development  
$ git checkout development  
$ git commit  
$ git checkout testing  
$ git merge development  
$ git checkout staging  
$ git merge testing  
$ git checkout production  
$ git merge staging  
$ git checkout development  
$ git commit  
$ git commit  
$ git commit
```

HEAD: development





LISÄTIETOJA

- <https://trunkbaseddevelopment.com/>
- <https://guides.github.com/introduction/flow/>

TYÖSKENTELYTAPOJA (WORKFLOW)

Esimerkiksi:

- Gitflow
 - Toinen esimerkki, ominaisuudet ja korjaukset omissa lyhytikäisissä brancheissa jotka sulautetaan masteriin
- Trunk-based
 - Muutokset pääasiassa suoraan masteriin/trunkkiin
 - Vältellään pitkäikäisiä versiohaaroja jotka voivat kasvattaa konfliktien määrää sulautumisessa

Lähestymistapoja kuitenkin paljon erilaisia, tärkeintä on että koko tiimi työskentelee saman tavan mukaisesti



KYSYTTÄVÄÄ?

- Seuraavaksi
 - Tehtävät @ Github classroom, linkki opettajalta