



Versionshantering

Just do it!

Versionshantering

- Varför versionshantering?
- CVS - *gudfadern*
- De stora aktörerna SVN och GIT
- Andra alternativ

Problemställningen

- Många personer samarbetar kring samma kodfiler
- Olika versioner av samma program används samtidigt
- Man vill experimentera med mjukvaran
- *"Varför fungerar inte programmet, VAD HAR DU GJORT!?"*



Hur samarbetar man?

1. En programmerare per fil

- Långsamt
- Ändringar i en fil kan orsaka följdfe

2. Många programmerare per fil

- Konflikter uppstår lätt

Version 2.254326

- Några kunder betalar för 1.0, andra för 2.0
- Versioner delar kodbas, men utvecklas separat
- Kunder kan köpa helt egna anpassningar

Nu med 3D-interface

- Man behöver experimentera
- Experiment blir förgreningar
- Förgreningar delar kodbas men ändringar ska inte förstöra huvudgrenen
- Det är ofta svårt att sammanfoga experimentgrenar med huvudgrenen

VAD HAR DU GJORT!?

- Mjukvaran har helt eller delvis slutat att fungera
- Vilken kod har ändrats?
- VEM VAR DET SOM KODADE!?



Typer av system

- client - server
 - Centraliserat
 - Traditionellt sätt att samarbeta
- peer-to-peer
 - Distribuerat
 - Modernt och bättre

Vanliga begrepp

- repository
- checkout eller pull
- commit och update
- conflict
- blame
- branch och merge

PAUS

CVS

- Concurrent Versioning System
- Skapades 1986
- Används fortfarande men är i princip helt ersatt av SVN

SVN - subversion

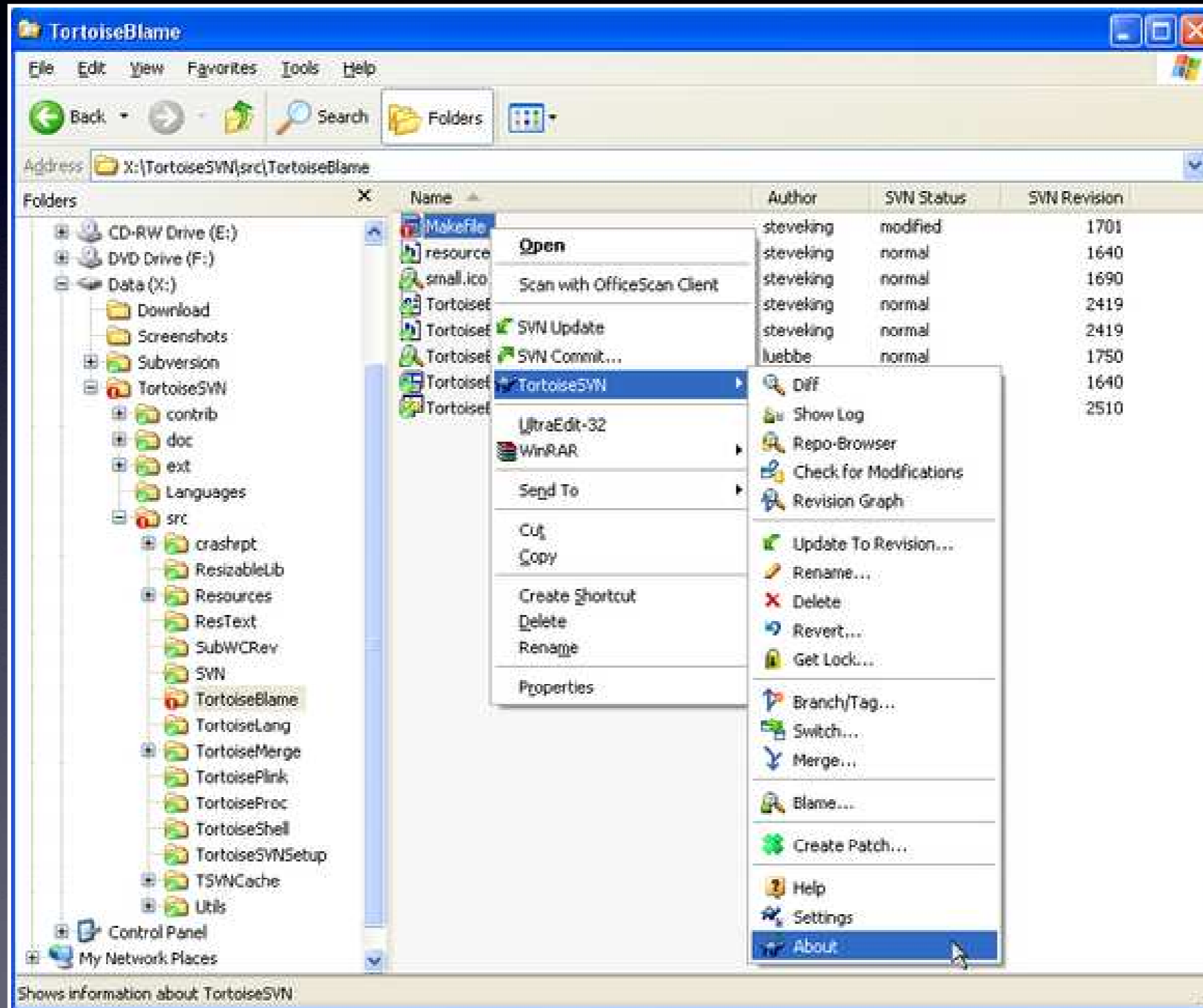
- Väldigt vanligt, men gammalt nu
- sourceforge.net samlar många projekt
- Arkivet ligger centralt på en server
- Programmerare hämtar kod till sin dator, gör ändringar och commit:ar dem

```
svn checkout svn://codeserver.com/daSoftware
cd daSoftware
***meckar med kod***
svn commit -m "Värsta ändringarna"

***några dagar senare***
svn update
svn blame code.php

***branching, merging och slagsmål***
```

TortoiseSVN



GIT

- VÄldigt populärt, tack vare github.com
- Distribuerat, varje person har hela arkivet
- Skapades 2005 av Linus Torvalds
- Avancerad hantering av projekt är mycket lättare än med SVN

Github.com

- Skapa ett gratiskonto, börja versionshantera all din kod idag!
- Ge varandra kommentarer och tips
- Skapa forkar av varandras kod

datorklubben / Spelprogrammering-med-JavaScript-och-Canvas

Watch

Fork

3

2

Code

Network

Pull Requests 0

Issues 1

Stats & Graphs

Ett bibliotek i JavaScript som förenklar spelutveckling — [Read more](#)

ZIP



HTTP

Git Read-Only

<https://github.com/datorklubben/Spelprogrammering->

Read-Only access

branch: master

Files

Commits

Branches 1

Tags

Downloads

Latest commit to the **master** branch

Small cleanup.



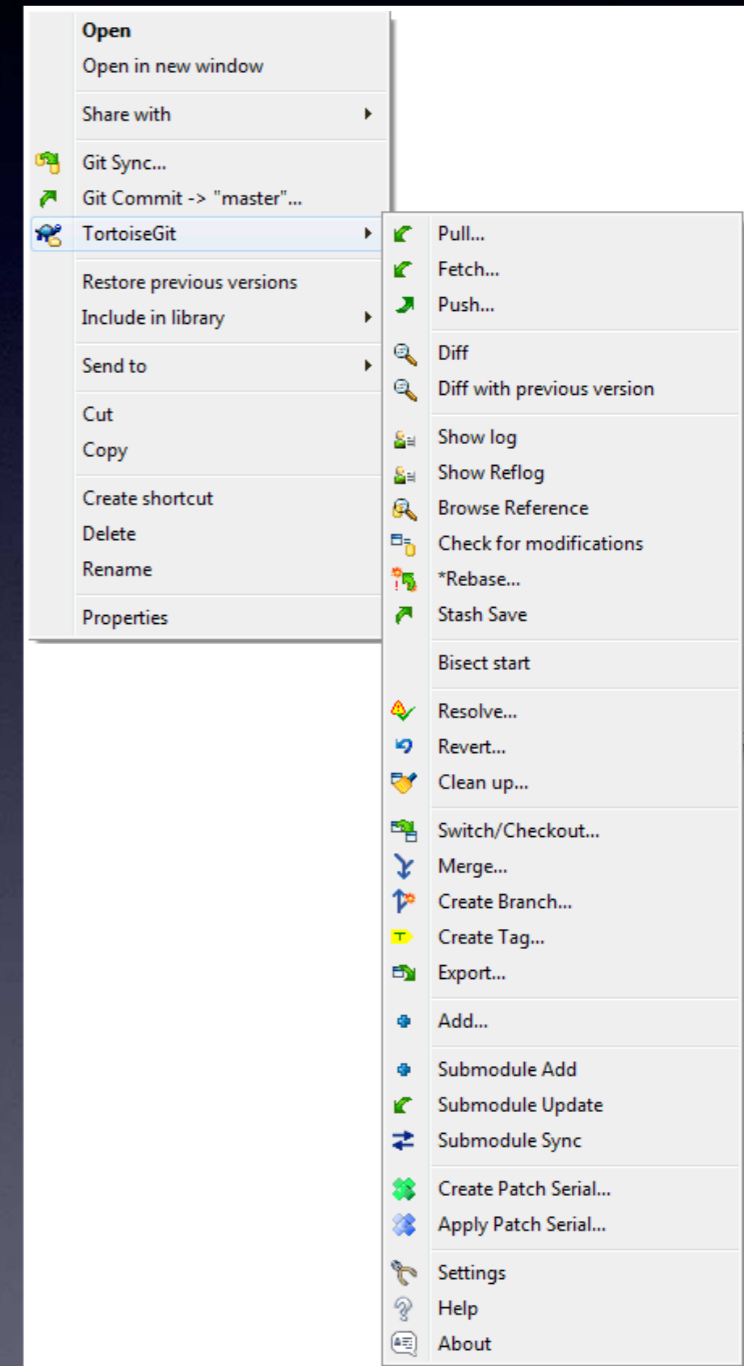
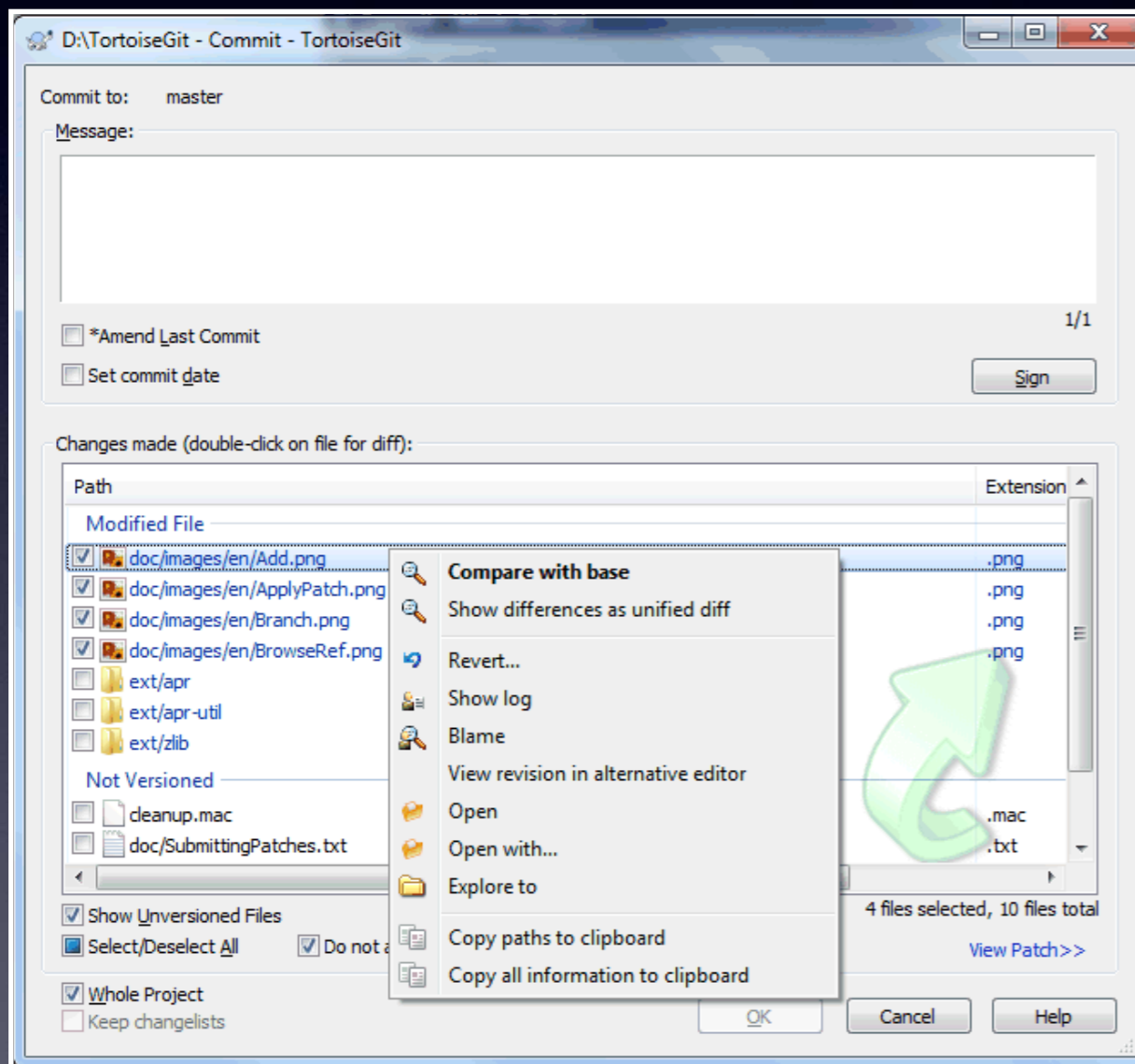
datorklubben authored about 2 hours ago

commit 60988aa2ac

Spelprogrammering-med-JavaScript-och-Canvas /

name	age	message	history
README	December 01, 2011	Added line [datorklubben]	
library.js	about 2 hours ago	Small cleanup. [datorklubben]	
ootraining.js	January 07, 2012	Added an environment for learning oop [Mikael Tylmad]	
trigonometry.js	January 07, 2012	Added some more functions [Mikael Tylmad]	

TortoiseGIT



Fördelar med GIT

- Varje programmerare har hela arkivet
- Väldigt snabbt
- Högre säkerhet
- Konflikter hanteras bättre
- Man kan enkelt utnyttja GIT lokalt

Andra alternativ

- Mercurial
 - peer-to-peer, minimalistiskt
- Team Foundation Server
 - Microsofts egna
- Bazaar
 - Kommer från Canonical, skaparna av Ubuntu