

Вариант №1

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'amber'
|\
| * (amber) Grief
| | M   polo.md
| * Roamer voyage nuclei
| | A   selective.json
* | Dander
| | M   account.json
| | M   femur.ini
* | Decimal doozy
| | A   account.json
| | D   polo.md
* | Opium
|/
|   M   femur.ini
* Spotting askew chamomile
  A canine.yaml
  A femur.ini
  A polo.md
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №2

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> latter) Jailbreak
| M ladder.ini
| A shortly.json
* Merge branch 'master' into latter
|\
| * (master) Thesaurus humorless shrapnel
| | M   battery.yaml
| | M   ladder.ini
| | A   shortly.json
* | Antarctic overvalue
| | R100 blazer spinach.yaml
* | Overview
|/
|   A   blazer
|   M   ladder.ini
* Upstream
  A battery.yaml
```

```
A ladder.ini
A waggle.md
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №3

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'bagging'
|\
| * (bagging) Oval
| | A overcome.ini
| | A unsavory.md
| * | Joyous
|/
| M donor.txt
| D hunger.py
* Ground relish
| M hunger.py
* Antivirus giving drift
| A donor.txt
| M hunger.py
* Yapping bobbing bouncing
A hunger.py
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №4

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'purse'
|\
| * (purse) Proofing
| | M bluff.md
| | A favorable.md
| * | City guts petite
|/
| R100 bluff.md battle.h
| M cannot.c
* Massager
| M bluff.md
| M cannot.c
* Vice leggings
A bluff.md
A cannot.c
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №5

Создайте в командной строке git-репозиторий, соответствующий логгу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'disband'
|\
| * (disband) Nativity swivel
| | R100   morally.yaml   elevate.txt
| * Mashed snowman
| | A   kite.c
* | Superjet
|/
|   A   charter.py
* Confound errant breath
  A chummy.py
  A morally.yaml
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №6

Создайте в командной строке git-репозиторий, соответствующий логгу, приведенному ниже.

```
* (HEAD -> settle) Nimble gleeful handed
| A watch.h
* (master) Espionage
| A dill.yaml
| M reviving
* Merge branch 'award'
|\
| * (award) Dejected
| | M   reviving
* | Lucrative
|/
|   D   dill.yaml
* Saint
| R100  coyness.ini dill.yaml
* Regime
  A coyness.ini
  A reviving
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №7

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Squiggle rogue
| M paddling.md
* Merge branch 'spout'
|\
| * (spout) Backlit spongy enjoyable
| | M paddling.md
| | M staring.md
* | Gout emperor
|/
| A chatting.c
| A impotence.md
| M staring.md
* Suburb guts pushiness
| M paddling.md
* Bondless
| A staring.md
* Mumps tricky vending
A paddling.md
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №8

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Safehouse collapse improvise
| M coroner.ini
* Merge branch 'unstaffed'
|\
| * (unstaffed) Cornball
| | A alarm
| | M coroner.ini
* | Bagging
| | M coroner.ini
| | R100 glove.h limeade.yaml
* | Trump clover
|/
| A fidgeting.yaml
* Resisting
| A coroner.ini
| M glove.h
| A trolling.h
* Whacking
A glove.h
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №9

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'fritter'
|\
* | Exuberant rift
| | A woof.c
| | * (retry) Coagulant
| | | M pager.json
| | | A reflex.md
| | | A virtual.yaml
| | * Operating unisexual carnival
| | /
| | R100 onyx.yaml overplant.py
* | (fritter) Unengaged kerosene deduce
| | M onyx.yaml
| * Waltz panic dime
| /
| M pager.json
* Exhaust chooser
  A onyx.yaml
  A pager.json
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №10

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> procurer) Merge branch 'cuddly' into procurer
|\
| * (cuddly) Manhood
| | A aide.ini
| | M rumor.txt
* | Cleaver
| | A reviving.yaml
| | M rumor.txt
* | Anyway duckling
| /
| D violate.txt
* Ecology wired
| M headpiece.c
| A rumor.txt
```

```
| M violate.txt
* (master) Magnetic upload
| M headpiece.c
| M violate.txt
* Reseal
  A headpiece.c
  A violate.txt
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №11

Создайте в командной строке git-репозиторий, соответствующий лог, приведенному ниже.

```
* (HEAD -> master) Merge branch 'chest'
|\
| * (chest) Strategic
| | D carrousel.md
* | Modular
| | M carrousel.md
| | A jaundice.json
| | A reply.yaml
* | Enlisted lair
|/
| M carrousel.md
* Purr latticed unwashed
  A carrousel.md
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №12

Создайте в командной строке git-репозиторий, соответствующий лог, приведенному ниже.

```
* (HEAD -> blurt) Merge branch 'stupor' into blurt
|\
| * (stupor) Dodge unexpired
| | M resupply.yaml
| | A revolver.py
* | Reentry untamed cardigan
| | D employee.ini
| | D resupply.yaml
| | M tiny.py
* | (master) Mocha
|/
| M bamboo.h
| M resupply.yaml
```

```
| A tiny.py
* Hardhat finite pupil
| M bamboo.h
| M resupply.yaml
* Viewpoint
| R100 clock.md bamboo.h
| M employee.ini
| M resupply.yaml
* Ruse plenty
A clock.md
A employee.ini
A resupply.yaml
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №13

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'blade'
|\
| * (blade) Tackle frigidly judgingly
| | M botch.md
| * Moonstone
| | M botch.md
| | A clamp.md
* | Fiber carload musty
| | M botch.md
| | A copy.txt
| | A spiritual.h
* | Unclass
|/
| M botch.md
* Zippy backpedal illusion
A botch.md
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №14

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'guidable'
|\
| * (guidable) Joylessly wistful skier
| | A line.json
| | M strobe
```

```
* | Dugout
|/
| M   cornbread.txt
| M   strobe
| A   washstand.ini
* Haste bloated overtake
| M strobe
* Anchovy fox muppet
| M strobe
* Identical barcode trophy
| A cornbread.txt
| M strobe
* Poser untwist
| A strobe
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №15

Создайте в командной строке git-репозиторий, соответствующий лог, приведенному ниже.

```
* (HEAD -> master) Merge branch 'tripod'
| \
| * (tripod) Easel gently
| | A   hunting.h
| | R100  curler  liver.md
| * Denim refusing placidly
| | A   curler
| | D   daylong.py
| * Bleak
| | M   daylong.py
* | Wasting
|/
| M   daylong.py
| A   protozoan.txt
* Mashing
| A daylong.py
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №16

Создайте в командной строке git-репозиторий, соответствующий лог, приведенному ниже.

```
* (HEAD -> master) Sprout pulmonary
| M amplify.ini
| R100 omen.yaml  doorknob.py
```



```

| M fountain.py
* Merge branch 'modular'
|\
| * (modular) Directed brute ream
| | D catapult.txt
| | A citable.yaml
| * Broker
| | M catapult.txt
| * Possible unwell
| | R100 amplify.ini catapult.txt
* | Rice
|/
| A budding
| A omen.yaml
* Phoenix tidings
  A amplify.ini
  A fountain.py

```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №17

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```

* (HEAD -> master) Bridged scowling
| A obsessive.ini
| M rumbling
* Merge branch 'stingily'
|\
| * (stingily) Disclose vocally
| | A obsessive.ini
| | M rumbling
| | A uncouth.ini
* | Oozy
|/
| A concerned.json
| M rumbling
* Catsup democrat
| R100 alkalize rumbling
* Handler knee masculine
  A alkalize

```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №18

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Difficult
| A try.h
* Tidiness
| M hardhat.ini
* Merge branch 'used'
|\
| * (used) Confirm dynasty situation
| | D evident.md
| | A spirits.yaml
| | A try.h
* | Frivolous
| | M evident.md
| | M uncurious.json
* | Anybody expand raft
|/
| A hardhat.ini
| M rebuttal.h
| A uncurious.json
* Trespass step dejected
  A evident.md
  A rebuttal.h
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №19

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (master) Merge branch 'staining'
|\
* | Fondly
| | A certainly
| | * (HEAD -> staining) Wagon
| | | R100 carpentry.h faction
| | | R100 override.h renewable.py
| | * Suffice
| | /
| | M carpentry.h
| | M diary.md
| | M sequester
* | Tying showpiece
|/
| M carpentry.h
| A sequester
```

```
* Overkill deprecate
| A carpentry.h
| D unsheathe.py
* Playhouse
  A diary.md
  A override.h
  A unsheathe.py
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №20

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'trivial'
|\
| * (trivial) Error gangly magnifier
| | R100 discard.md reptilian.ini
| * (helium) Junction footer
| | A caloric.md
| | R100 comprised.py discard.md
* | Unwieldy magnify pledge
|/
| A imprint
| A navy.txt
| R100 comprised.py undergrad
* Surgery yummy marshland
| A comprised.py
* Hunchback giggling
| D doorbell.h
* Wipe
  A doorbell.h
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №21

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'nicotine'
|\
| * (nicotine) Obstinate gauging
| | A unlinked.yaml
| | R100 jackal.h wobbly.txt
| * Anew
| | M jackal.h
* | Cattle mobility
```

```
| | D   jackal.h
| | M   powwow.yaml
* | Sponge fedora
|/
|   M   jackal.h
|   A   powwow.yaml
* Petty smother
  A jackal.h
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №22

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'upstart'
|\
| * (upstart) Pauper borrower
| | M   cozily.py
| * Staunch probiotic pointed
| | M   cozily.py
| | M   tigress.yaml
* | Ruby backtalk golf
|/
|   M   cozily.py
|   R100 deferral.txt   hesitate.md
|   M   tigress.yaml
* Edginess
| M cozily.py
| A deferral.txt
* Coral
  A cozily.py
  A tigress.yaml
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №23

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'ship'
|\
| * (ship) Congested gainfully
| | M   navigate.h
| * Nautical fiddling handed
| | D   penpal.json
* | Gigabyte utopia wikipedia
```

```

|/
| M broom.json
| M unloaded.py
* Spinning lunchroom ragged
| A broom.json
| A penpal.json
* Hypnoses
| A navigate.h
| A unloaded.py

```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №24

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```

* (HEAD -> master) Merge branch 'tapered'
|\
* | Clinking blooming maturity
| | M ought.txt
| | A riverbed
| | M sandblast
| | * (cringe) Sadden
| | | M ought.txt
| | | M thrower.py
| | * Liquefy
| | /
| | M sandblast
| | R100 retriever.py thrower.py
* (tapered) Groovy five
|/
| A energize.h
| M sandblast
* Porcupine reclaim
| A sandblast
* Ancient
| A ought.txt
| A retriever.py

```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №25

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```

* (HEAD -> deem) Handclasp unsaid
| A mobilize.h

```

```

* (master) Merge branch 'wrecking'
|\
| * (wrecking) Alienable
| | A disbelief.yaml
* | Manor
|/
| R100 mobilize.h livestock.json
| A manpower.json
* Haste splashy facility
| A husband.ini
| M improving.h
| A mobilize.h
* Humming sanded payee
A improving.h

```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №26

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```

* (HEAD -> expensive) Overstate omen
| A unsecured.json
* Merge branch 'master' into expensive
|\
| * (master) Manhood drivable
| | M tavern.json
| | R100 ruby.h unsecured.json
| * Borrower tapping
| | M ruby.h
| | A tavern.json
* | Blustery mobility
| | D ruby.h
* | Olympics
|/
| M ruby.h
* Spray
A ruby.h

```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №27

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```

* (HEAD -> master) Bovine
| R100 striking.py gossip.ini

```

```
* Dividable paving
| A denatured.py
| A striking.py
* Merge branch 'emission'
|\
| * (emission) Conch rocking
| | A denatured.py
| | M divisibly
| | A selected.txt
* | Thespian rundown
|/
| D divisibly
* Bunkbed overbite
A divisibly
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №28

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> buzz) Promenade
| D dyslexia.md
| M plentiful.h
* Atop
| M dyslexia.md
* Merge branch 'master' into buzz
|\
| * (master) Dice discolor
| | A reprint.txt
| | D salutary.md
* | Eel radiation throwing
| | M dyslexia.md
| | R100 tweak.py enrich.c
* | Tattle wind flammable
|/
| A dyslexia.md
| R100 salutary.md plentiful.h
* Hatchery shut
A salutary.md
A tweak.py
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №29

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> slideshow) Fraction quintet
| A immortal.py
* Cornball slobbery despite
| M vivacious.txt
| * (master) Merge branch 'slideshow'
| |\
| |/\
| |/\
| |/\
* | Implosive sled
| | M quality.h
| * Deflation
| /
| M quality.h
| A spoof.c
* Vegan bulgur
| M quality.h
* Passion collar
| A quality.h
| A vivacious.txt
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №30

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> sinless) Brute
| M barcode.json
| D greedy.c
| M passive.txt
* (discount) Arousal
| A barcode.json
| M greedy.c
| M passive.txt
* Merge branch 'robin' into discount
| \
| * (robin) Jeep
| | M passive.txt
* | Countless letter
| | M greedy.c
* | (master) Football handball
| /
| D barcode.json
| R100 atypical.py greedy.c
```



```
* Reason
A atypical.py
A barcode.json
A passive.txt
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №31

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Cargo rural splashed
| A unroll.yaml
* Merge branch 'choice'
|\
| * (choice) Thwarting
| | R100    immersion.txt    superglue.json
* | Unpledged
| | D    impish.h
* | Scheming lapping
|/
|   D    immersion.txt
* Washroom uncoated briskly
  A immersion.txt
  A impish.h
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №32

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> student) Merge branch 'master' into student
|\
| * (master) Theme capsule
| | A    crib
* | Configure movie external
| | A    cube
* | Helpful expensive explicit
| | D    penholder.py
| | D    prudishly.txt
| | D    unworthy.py
* | Ritalin pager
|/
|   A    prudishly.txt
|   M    unworthy.py
* Deflector agent kilowatt
```

```
| A penholder.py
* Catalog
  A cortex.ini
  A unworthy.py
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №33

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'snorkel'
|\
| * (snorkel) Oven geriatric
| | M   compare.json
| | M   overkill.json
| | M   rake.h
| * Nacho unripe dexterity
| | A   overkill.json
| | A   rake.h
| * Kudos uncoiled passenger
| | M   gravitate.json
* | Work staunch
| | D   gravitate.json
| | A   proving.h
* | Sector
|/
| D   compare.json
* Manual probiotic
  A compare.json
  A gravitate.json
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №34

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> master) Depth kilowatt premium
| M frozen.md
* Merge branch 'skipper'
|\
| * (skipper) Script
| | M   deuce.json
| | A   paddling.h
* | Outline strenuous
|/
```

```
| M frozen.md
| A geiger.h
* Deputy
| A deuce.json
* Headphone paging humped
| A frozen.md
| M scrabble.ini
* Trolling outscore
  A scrabble.ini
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №35

Создайте в командной строке git-репозиторий, соответствующий логгу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'unstuffed'
|\
| * (unstuffed) Arming
| | D thumping.md
| * Elves armored yogurt
| | M thumping.md
| | Scorebook affront unlaced
|/
| A savor.md
| D thumping.md
* Badge
  A thumping.md
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №36

Создайте в командной строке git-репозиторий, соответствующий логгу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'handoff'
|\
| * (handoff) Malformed croak
| | M evacuee
| * Clergyman
| | M corset.c
| | M evacuee
| | A plausible.yaml
| * Component skipping parachute
| | A embattled.txt
| * Opponent
|/
```

```
| M corset.c
| M evacuee
* Twentieth rash
  A corset.c
  A evacuee
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №37

Создайте в командной строке git-репозиторий, соответствующий логгу, приведенному ниже.

```
* (HEAD -> master) Merge branch 'acrobat'
|\
| * (acrobat) Sprite observing probably
| | M dropout.yaml
| | A unfunded
* | Viscous unedited disparity
|/
| M dropout.yaml
| A error.py
| R100 unveiled.h gains.py
* Rigging plug
| M dripping.json
| A dropout.yaml
| M unveiled.h
* Probe anemia
| M dripping.json
| A unveiled.h
* Book congenial
| M dripping.json
* Handgun nimbly afoot
  A dripping.json
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №38

Создайте в командной строке git-репозиторий, соответствующий логгу, приведенному ниже.

```
* (HEAD -> hydroxide) Overhung plutonium skyrocket
| D trickster.txt
| * (master) Merge branch 'hydroxide'
| |\
| |/\
|/|
* | Dirtiness married bronzing
```

```
| | A   scrabble.ini
| * Headgear plank cassette
| /
|   A   stinking.py
|   M   trickster.txt
* Marital anagram
  A trickster.txt
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №39

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (master) Merge branch 'traffic'
| \
* | Salutory
| | M   resort.ini
| | * (HEAD -> traffic) Stingy cement
| | /
| |   R100 user.ini      player.py
| |   M prodigy.ini
| |   M regretful.md
| * Pry creamed
| | M   user.ini
| * Tiring bogus
| /
|   A   prodigy.ini
|   A   regretful.md
* Pouch
  A resort.ini
  A user.ini
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```

Вариант №40

Создайте в командной строке git-репозиторий, соответствующий логу, приведенному ниже.

```
* (HEAD -> exemption) Earpiece cache
| M dislodge.json
| R100 overkill      ended.txt
| M outdoors.md
* Emphasize
| M mating.json
| M outdoors.md
| * (master) Merge branch 'exemption'
```

```
| | \
| | /
| / |
* | Maturity clay
| | A   mating.json
| | M   outdoors.md
| | A   overkill
| * Haziness unquote casket
| /
|   M   dislodge.json
|   D   outdoors.md
* Paprika pedometer splotchy
| R100  carrot  outdoors.md
* Work undiluted proofread
  A carrot
  A dislodge.json
```

Для проверки используйте команду:

```
git log --graph --name-status --pretty=oneline --all --decorate
```