

Qip User Instruction

What is Qip?

Name

Qip is a name from Khipu/Quipu (tying knots system used by cultures in the region of Andean South America.¹) which is also homophonic of "Keep".

Purpose

Qip is designed as a simple version control software inspired by Git² but apply to game save. Unlike Git, Qip separate origin save (Working Copy) and repository which make it also a backup software. Qip can restore committed save directory even origin save directory is totally deleted by user or game.

Quick start

1. Open Qip.
2. Click left "Add" button.
3. Click "Add New Repository".
4. Input **Name** (a good name you can easily recognize which game it is). Select **save directory** where you game save data include (you can get it from search engine). Select an **empty directory for repository** (you can create a directory which contains all repository directories for better manage). Select game exe file if you want to launch game directly from Qip (this will give you an icon before game name in repository list). Click "**Confirm**".
5. **Click repository** just added.

¹ <https://en.wikipedia.org/wiki/Quipu>

² <https://git-scm.com/>

6. Click "Commit" button.

7. Input `message`. Click "Commit to New Branch" toggle button and input first branch name if you don't want to use default branch name "main". Click "Confirm" and you will have first commit in right commit list.

8. Play game and save (better close game, too). Return your repository and you will find there is an orange "!" in left right corner represent your save is not up to date. Just click "Commit" button again add input message and you will create second commit. If you successfully create a commit, the "!" will become a green "✓".

Interface

Qip has 2 main interfaces: Repository and History.

Repository View

Repository view contains a toolbar and a repository list. Toolbar contains an "Add Repository Menu" and a filter. Repository list contains all game save repositories managed by Qip. In repository view you can create, edit, search and delete repositories.

Repository

Repository is where game save files and meta files are contained. Every repository is self-contained, which means you can move repository from one location to another location, then re-add it and everything will be fine.

Working Save

The save currently used by game. Located in save directory.

Head

The commit which will be parent commit of working save.

History View

Left click on repository or select "History" in context menu of repository will bring you to history view. History view contains a toolbar, a sidebar and a commit list. Toolbar contains some buttons and menus can be used to manipulate commits. Sidebar contains a branch list, an event list and a tag list. Commit list contains all commit of a repository. From left to right is commit graph, commit message, commit hash and commit create date.

Graph

Graph represent relationship of commits. More specifically -- parent-child relationship. A child commit is always come from a parent commit one or more generation before.

Commit

Commit represent a complete copy of save data store in repository. You can easily restore a commit to origin save data or delete it. A commit has some property, including message, hash, parent, create date and so on.

Hash

Hash of save is like fingerprint of save. Qip use SHA-1 hash algorithm to guarantee that every save has a unique hash. Even smallest change won't escape from detecting thanks for SHA-1.

Message

Message is short text for each commit to recording what this commit is. Message makes it's easier to tracking commit history. It's a good habit to keep message for every commit and it won't allowed to leave blank.

Create Date

The date and time commit created.

Parent Commit

The first commit before a commit. A commit can only have one parent commit. If you delete parent of some commits, parent of deleted parent will become parent of children lost their parent.

Child Commit

A set of commits share same parent is children of that parent commit. A commit can have multiple children commit.

Branch

Branch is a series of commits connect by parent-child relationship. A branch in Qip does not go all the way down to root commit (first commit) like some version control system.

In graph, a branch distinct from each other by color and column it locates.

Branch Tip

Newest commit on a branch.

Branch Base

The first commit branch out from another branch.

Parent Branch

Branch whose base commit belongs to is parent branch of that branch. A branch can only have one parent branch.

Child Branch

A set of branches share same parent branch is children of that parent commit. A branch can have multiple children branches.

Event

Some big event happened when playing like begin of second round. Event can be used to locate commit in history. A commit can only have one event.

Tag

You can manually add tag to tag list. Tags will be used as filter if selected. Input "#" in commit message text box will bring an auto completion list with tags.

Repository How?

Create A Repository

Click "Add" button. Select "Create New Repository".

Add Exist Repository

If there is some repository not recorded. You can add it manually. Click "Add" button. Select "Add Exist Repository". There will be a folder select dialog open. Select repository directory and click "Confirm". The repository will be added to repository list automatically.

Edit Repository

Right click repository you want to edit. Select "Edit" in popup context menu. A dialog will show with "Name", "Save Directory" and " "Exe Directory" text edit.

Remove Repository

Right click repository you want to edit. Select "Delete" in popup context menu. A dialog asks for confirm will show. Click "Delete" to permanently remove repository. This will permanently remove everything in repository directory include but won't delete current save in save folder.

Filter Repository

Input words you want to filter in left top filter textbox.

Launch Game

Right click repository you want to edit. Select "Launch" in popup context menu. If executable file is added before. The executable file will be launched.

Commit How?

Create Commit

Click commit button. A dialog will popup. Input commit message (Can't be empty). Select whether create a new branch. Input event (Not required). Click "Confirm".

Note: Please avoid create commit when playing game.

Checkout Commit

1. Select commit in history. Click "Checkout" button.

2. Right click commit to checkout. Select "Checkout" in context menu.

3. Double click commit to checkout.

If there are unsaved change, a dialog will pop up and ask if commit unsaved change. Click commit if need to commit. Click Discard if don't need to commit.

Note: Please avoid checkout commit when playing game. This may cause unexpected result. If you did this and your game says save is corrupted or something similar. That means your current save is partly deleted. Close game and try checkout again and you may get save you want to check out.

Create Branch

Unlike git, a branch in Qip can only be created when commit. Select create new branch add input a branch name, then a new branch will be created, and new commit will be base commit of new branch. If your head is in the middle of a branch, you must create a new branch to commit.

Delete Commit

Commit can be easily deleted. Just right click commit you want to delete and click "Delete". Or first select commit you want to delete, then click "Delete Commit" button in menu bar. A dialog asks for delete will show. Select "Delete" will permanently delete the commit. You cannot delete root commit with multiple children branch and root commit which is also head commit because this may result some commit isolate from current commit tree.